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
  5 级深度堆栈寄存器
  直接，间接以及相对寻址模式
- **非易失程序与数据存储空间**
  1024x14 一次可编程**(OTP)**程序存储器
  64 字节内部 **SRAM**
  全新的程序加密算法，保证用户代码安全
- **外设控制器**
  1x 10bit 定时器**(TMR0)**
    - **10** 位比较寄存器、**8** 位预分频器
    - 可编程内部**/**外部时钟源

  1x 16bit 定时器**(TMR1)**
    - 支持外部门控输入
    - 支持内部**/**外部时钟输入模式

  增强型俘获**/PWM** 控制器**(ECP)**
    - 独立外部事件输入俘获
    - 支持最多 **4** 路 **PWM** 输出
    - 增强型 **PWM** 模式，支持自动关闭、死区控制

  1x 多路输入模拟比较器**(ACM)**
    - 出厂失调校准
    - **0.58V** 内部参考
    - **32** 级分压参考**(VCC/n)**

  1x **I²C** 控制器，支持主**/**从工作模式
    - 内置 **I²C** 接口 **E2PROM** 控制器

  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**
  SWD 双线量产接口
  外部中断源与 **I/O** 电平变化中断支持
  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与晶振启动计时器
  内置**±1%**可校准 **8MHz RC** 振荡器
  支持外部晶振输入
  多种低功耗休眠模式，支持外部中断以及电平变化唤醒
- **I/O 与封装：SOP14/8L**
- **工作环境**
  工作电压： **1.8V ~ 3.6V**
  工作温度： **-40C ~ +85C**
  待机功耗： **1uA@3.3V**

8-bit MIC8S

Microcontroller with
1024 Bytes In-System
OTP Memory

LGT8P663A

Overview
Version 1.0.4

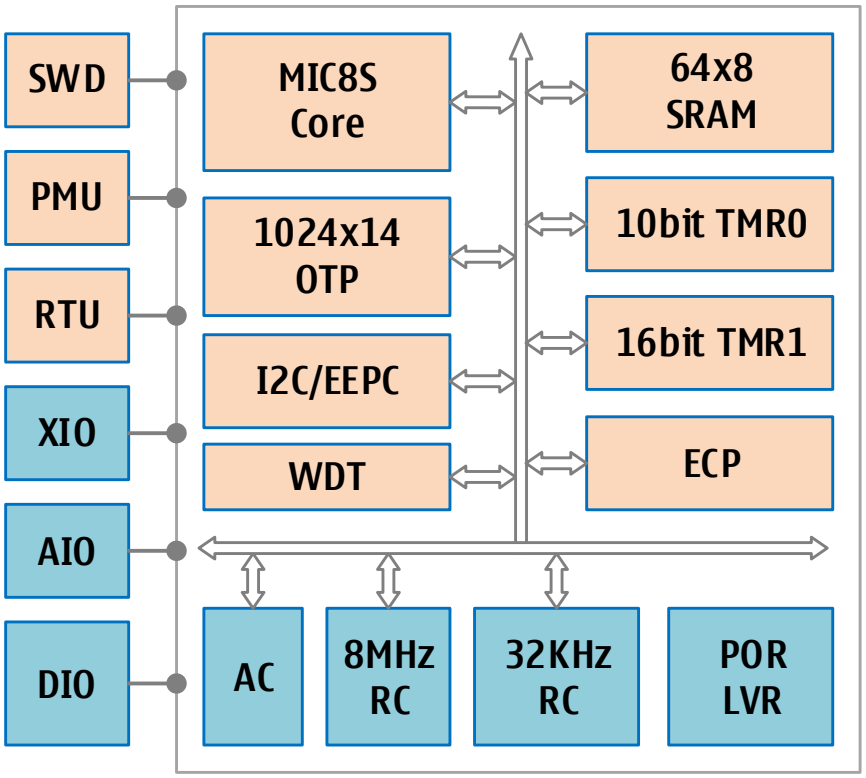
应用领域

小家电

智能控制电路

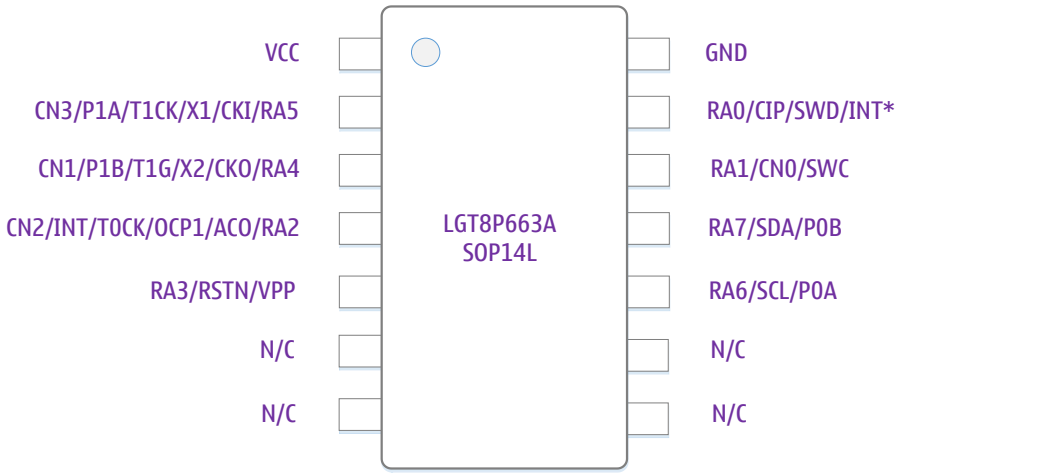
- 电源管理
- 电动产品
- 智能玩具
- 密码锁
- 报警器
- 照明控制与驱动

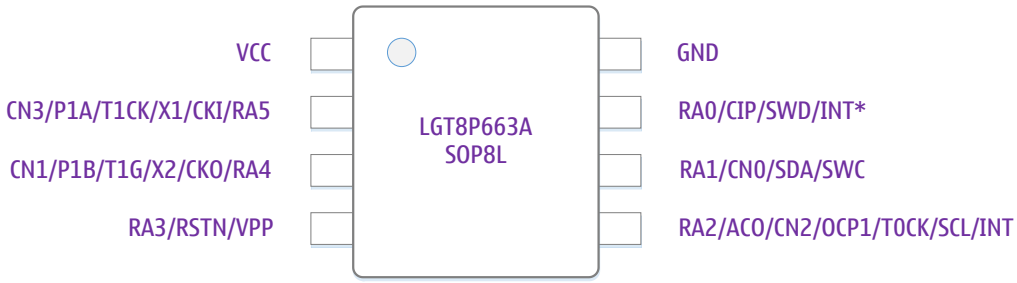
系统框架



模块名称	模块功能
MIC8S	MIC8S 8 位微处理器内核
SWD	SWDISP 编程器接口
PMU	功耗管理单元
RTU	复位控制单元
XIO	晶振 I/O
AIO	模拟 I/O
DIO	数字 I/O
AC	多路输入模拟比较器
I²C/EEPC	I²C 主从控制器 E²PROM 控制器
TMR0	10 位定时器 0
TMR1	16 位定时器 1
ECP	增强型俘获/PWM 控制器
WDT	看门狗定时器
POR/LVR	上电复位 低电压复位

封装定义





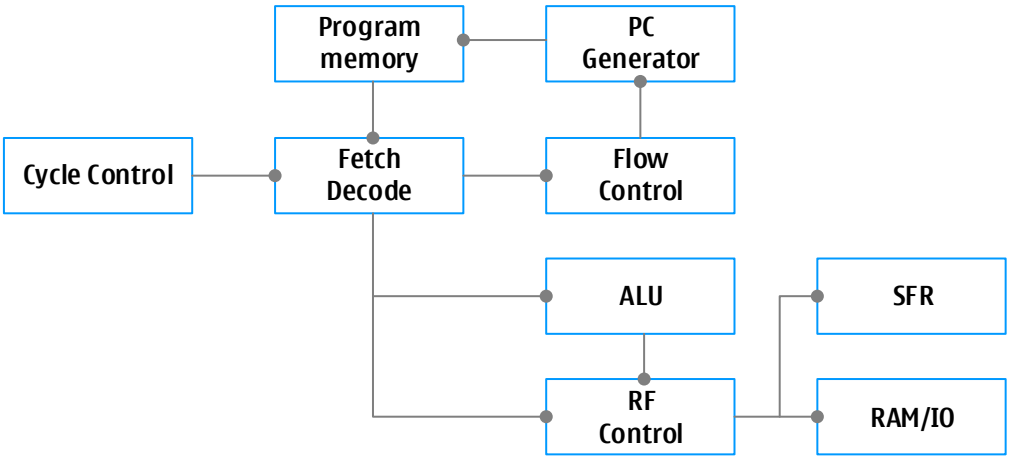
引脚说明

Pin.	GPIO	AC	I2C	Timer	ECP/PWM	IRQ	Power	MISC
							VDD	
	RA0	CIP				IOC/INT*		
	RA1	CN0	SDA*			IOC		
	RA2	ACO/CN2	SCL*	T0CK	OCP1	IOC/INT		
	RA3					IOC	VPP	RESETN
	RA4	CN1		T1G	P1B	IOC		X2/CLKO
	RA5	CN3		T1CK	P1A	IOC		X1/CLKI
	RA6		SCL		P0A	IOC		
	RA7		SDA		P0B	IOC		
							GND	

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：

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；

LGT8PE663A 寄存器空间为分页映射模式，实现两页地址空间的映射，可以通过间接寻址方式访问到最大 256 字节(0x00~0xFF)的地址范围。包括 64 字节内部数据空闲，定时/计数器 1 以及模拟比较器等等。

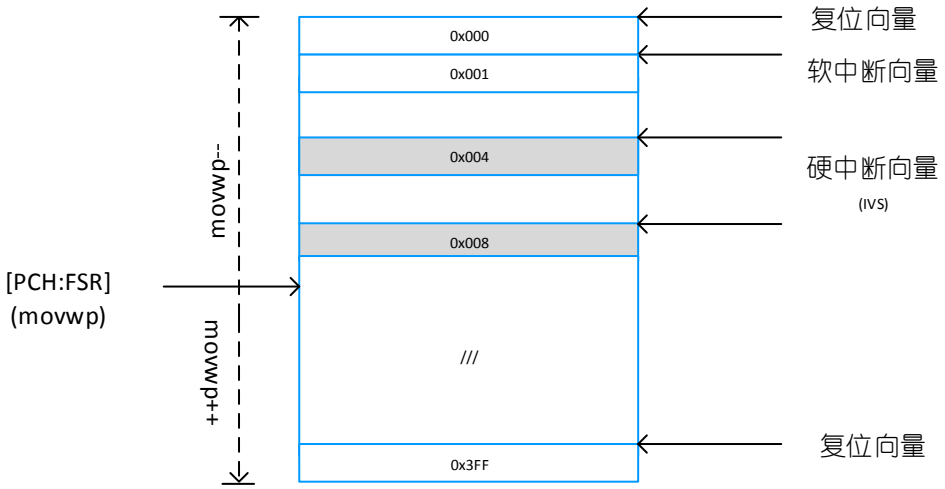
存储系统

程序存储空间

MIC8S 指令可直接寻址最大 8K 指令字(14b)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。

MIC8S 系统的复位向量地址为 0x000。

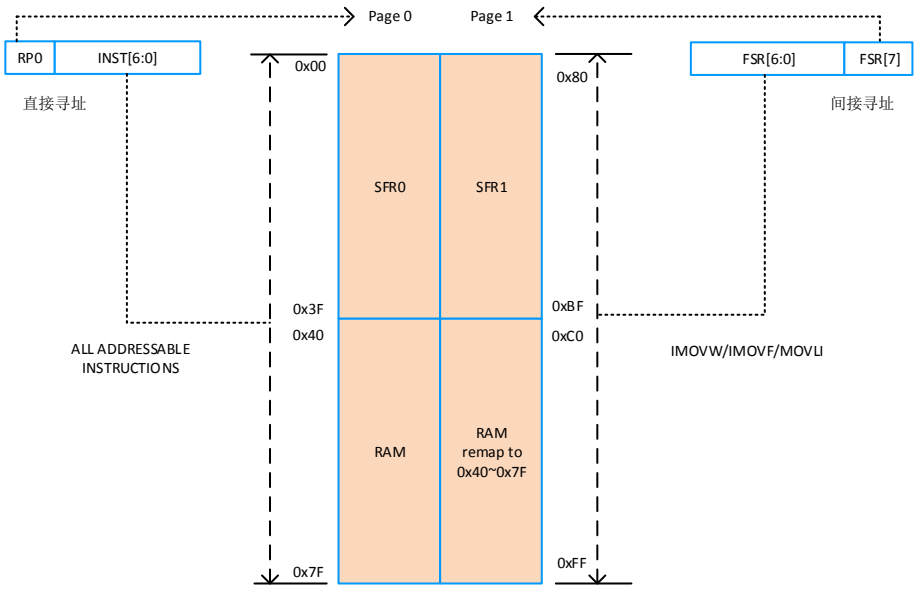
MIC8S 支持软件触发的中断（INT 指令）和一个硬件中断入口，软中断向量地址为 0x001，硬件中断向量地址可配置为 0x004 或 0x008。



数据存储空间

MIC8S 支持分页映射模式，MIC8S 最大支持 4 页的分页空间；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。

数据存储空间地址映射图：



对于连续映射模式，地址空间 0x00~0x3F 能够被指令直接寻址。MIC8S 大部分指令都可以直接操作在这个范围内的寄存器或者 RAM。不同的是涉及到 RAM 空间读操作的指令，需要两个指令周期才能完成。对于 0x40~0xFF 地址范围的访问，需要使用间接寻址模式，FSR 将作为访问他们的间接地址寄存器。MIC8S 提供了 IMOVW/IMOVF/MOVLI 等专用指令，方便对这部分空间的访问。

对于分页模式中的寄存器空间，如果没有实现，默认将会直接映射到第 0 页对应地址。

在分页模式下， 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。

5 级深度堆栈寄存器

LGT8PE663A 实现了一个 5 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 5 级，当连续执行 5 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。

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	WKPF	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
FSR	0x04	FSR[7:0] 间接寻址地址寄存器							
PORT	0x05	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0

	0x06 ~ 0x09 Unimlemented								
PCHBUF	0x0A	-	-	-	-	-	-	PCH[9]	PCH[8]
INTCON	0x0B	GIE	PEIE	T0IE	INTIE	GPIE	T0IF	INTF	GPIF
PIR1	0x0C	TWIF	-	ECPIF	-	CMIF	-	-	TMR1IF
	0x0D	Unimplemented							
TMR1L	0x0E	16 位 Timer1 计数器低 8 位							
TMR1H	0x0F	16 位 Timer1 计数器高 8 位							
T1CON	0x10	T1GNV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
DC0AL	0x11	TMR0 占空比寄存器 A 低 8 位							
DC0BL	0x12	TMR0 占空比寄存器 B 低 8 位							
DC1AL	0x13	TMR1 占空比寄存器 A 低 8 位							
DC1AH	0x14	TMR1 占空比寄存器 A 高 8 位							
ECP0CON	0x15	P0AOEN	P0BOEN	DC0AH[1:0]		DC0BH[1:0]		P0APOL	P0BPOL
PWM0CON	0x16	PWM0M	PWM0ADB[2:0]			PWM0BDB[3:0]			
ECP0AS	0x17	ECP0ASE	PSR0EN	CMPS0E	INTS0E	PSS0A[1:0]		PSS0B[1:0]	
PWM1CON	0x18	PWM1M	PWM1ADB[2:0]			PWM1BDB[3:0]			
VRCON	0x19	CMVREN	-	VRR	FVREN	VR3	VR2	VR1	VR0
CMCON0	0x1A	CMON	COUT	CMOE	CMPOL	-	CMR	CMCH1	CMCH0
ECP1CON	0x1B	P1AOEN	P1BOEN	CAPEN	CAPM	BUF1AE	BUF1BE	P1APOL	P1BPOL
CMCON1	0x1C	-	CFEN1	CFEN0	T1ACS	CMHYS	-	T1GSS	CMSYNC
ECP1AS	0x1D	ECP1ASE	PRS1EN	CMPS1EN	INTS1EN	PSS1A[1:0]		PSS1B[1:0]	
0x1D~0x1F Unimplemented									
0x20~0x2F：配置空间+用户自定义配置空间									

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	FSR 间接寻址数据							
OPTION	0x81	$\overline{GPPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
PCL	0x82	PC[7:0]							
STATUS	0x83	WKPF	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
FSR	0x84	FSR[7:0] 间接寻址地址寄存器							
TRISIO	0x85	TRIS7	TRIS6	TRIS5	TRIS4	-	TRIS2	TRIS1	TRIS0
0x86~0x87 Unimplemented									
PR1L	0x88	TMR1 计数周期寄存器低字节							
PR1H	0x89	TMR1 计数周期寄存器高字节							
PCHBUF	0x8A	-	-	-	-	-	-	PCH[9]	PCH[8]
INTCON	0x8B	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF
PIE1	0x8C	TWIE		ECPIE		CMIE			TMR1IE
TWDR	0x8D	I2C 数据接收/发送寄存器							
PCON	0x8E	WDTE	CWOK	LVRE	DPSM1	DPSM0	SWDD	POR	LVR
0x8F~0x90 Unimplemented									

OSCTUNE	0x90	TUN7	TUN6	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
T0CON	0x91	T0D	-	T0H1	T0H0	-	-	PR0H1	PR0H0
PR0L	0x92	TMR0 计数周期寄存器低字节							
DC1BL	0x93	TMR1 占空比寄存器 B 低 8 位							
DC1BH	0x94	TMR1 占空比寄存器 B 高 8 位							
PUCR	0x95	PUC7	PUC6	PUC5	PUC4	PUC3	PUC2	PUC1	PUC0
IOCR	0x96	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
PDCR	0x97	PDC7	PDC6	PDC5	PDC4	-	PDC2	PDC1	PDC0
TWSR	0x98	TXP	RXK	TXD	TXS	RXP	TXK	RXD	RXS
0x99 Unimplemented									
EEPDR	0x9A	E2PROM 数据寄存器							
EEPAR	0x9B	E2PROM 地址寄存器							
EEPCR	0x9C	EEPEN	EERST	EEPMD	EEPBR	EEPER	EEPWE	EEPPE	EEPRE
TWCR	0x9E	TWEN	TWMST	-	-	TACK	RACK	CKPS1	CKPS0
ANSEL	0x9F	-	-	-	ANS4	ANS3	ANS2	ANS1	ANS0
0xA0 ~ 0xBF：用户自定义配置空间									

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，$\overline{TO}$/$\overline{PD}$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/MOVWF 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	RP1	RP0	$\overline{TO}$	$\overline{PD}$	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	RP1/0	通用寄存器位；在分页地址映射模式下 RP[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	$\overline{PD}$	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 LVD 复位有效		

OPTION – 外设配置寄存器

OPTIO 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够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。

在 8P53A 模式下，OPTION 寄存器只能用 OPTION 指令访问；OPTION 指令将工作寄存器 W 中的数据更新到 OTPION 寄存器中。在 8P609A 模式下，OPTION 寄存器被映射到第 1 页 0x81 地址，除了可以通过 OPTION 指令访问，也可以通过其他指令寻址。

OPTION – 外设配置寄存器								
~~8P53A 地址：仅通过 OPTION 指令访问~~					默认值: 1111_1111			
8P609A 地址: 0x81								
Bit	7	6	5	4	3	2	1	0
OPTION	$\overline{GPPU}$	INTEDG	T0CS	T0SE	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	$\overline{GPPU}$	GPIO 全局上拉禁止控制 1：全局禁用上拉 0：全局上拉使能						
6	INTEDG	外部中断/唤醒边沿选择 1：INT 的上升沿产生中断 0：INT 的下降沿产生中断						
5	T0CS	Timer0 时钟源选择 1：选择 T0CKI 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	T0SE	Timer0 时钟触发沿选择 1：T0CKI 的下降沿 0：T0CKI 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		
2:0	PS[2:0]	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。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 LGT8PE663A 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。

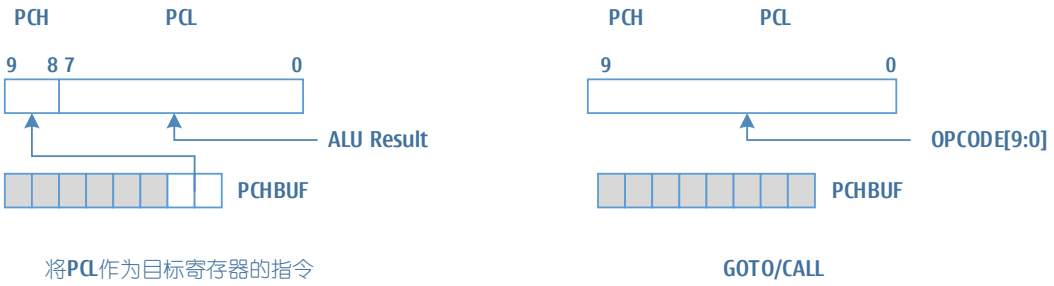
下面我们通过一个实例程序说明如何利用间接寻址进行高效的数据访问；实例程序使用间接寻址方式，清零 0x40 到 0x7F 之间的 RAM 区域：

```
        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							
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；						

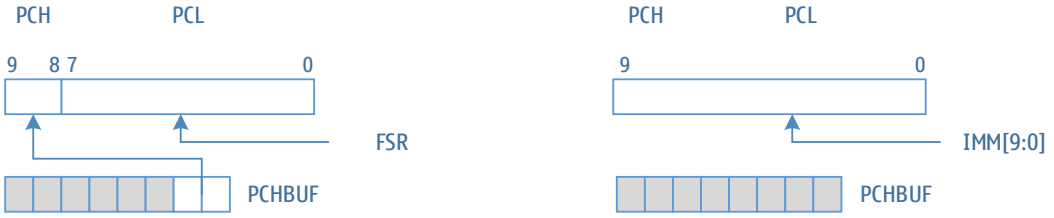
PCL – PC 低字节寄存器

LGT8PE663A 实现了 1K 指令字的程序空间，程序计数器(PC)为 10 位宽。PC 的低字节来自 PCL 寄存器。PCL 寄存器可读/写，PC 的高 2 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 MOVLC/MOVLT/MOVLI 这三条双指令字指令，MOVLC 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：

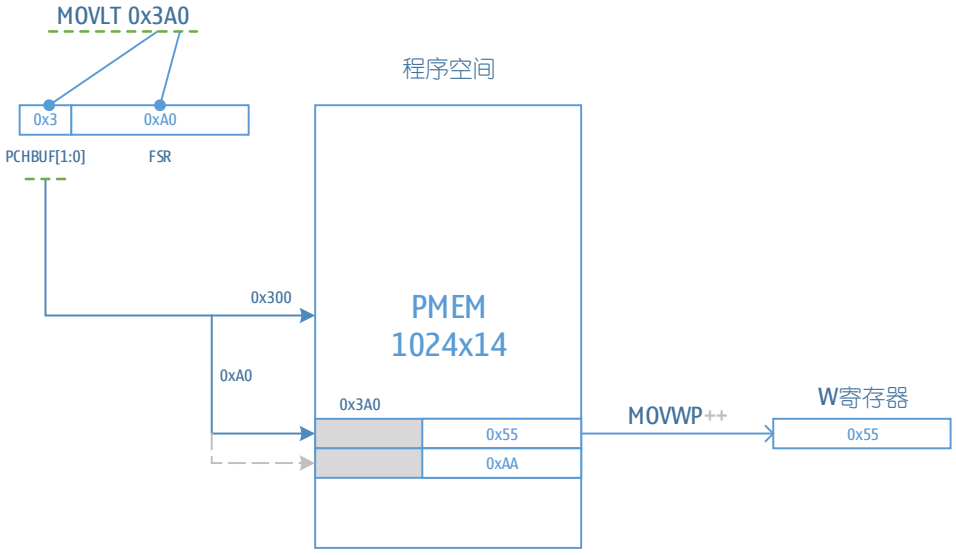
- 间接寻址数据/程序空间; (MOVWP/IMOVW/IMOVF)
- 自动 FSR 递增/递减的数据/程序空间访问; (MOVWP++/ --, IMOVW/F++/--)
- 带 4 位偏移量的数据空间访问; (IMOVW/F +q/-q)

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：

MIC8S 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[1:0]以及 FSR 共同决定，可以实现对 LGT8FP653A 整个 1K 指令字程序空间的访问。

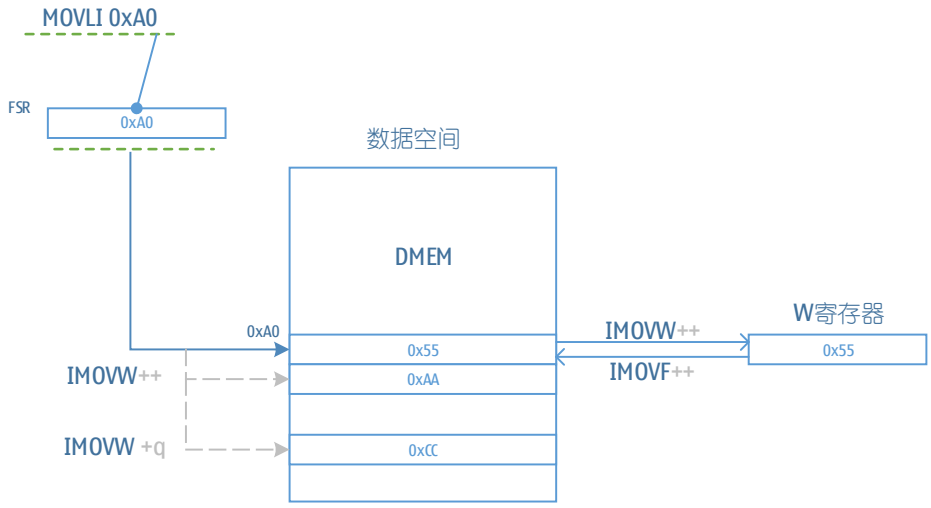
下图简单说明了 MOVWP 寻址程序空间的过程以及目标地址的构成方法：



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，MIC8S 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。MIC8S 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 LGT8PE663A 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

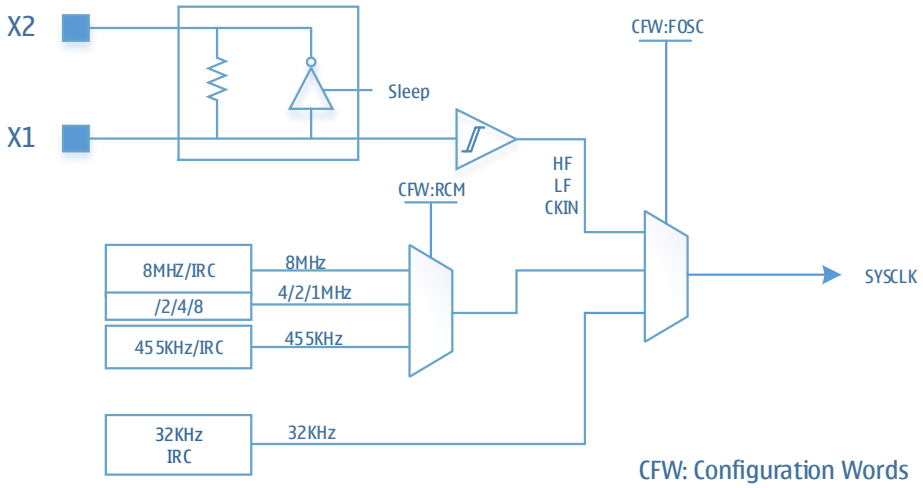
备注：LOOP 指令执行时，先判断 LCR 的值，如果为 0，则直接运行下一条指令；否则 LCR 的值减 1，程序跳转到 LOOP 指定的目标地址执行。LCR 寄存不寻址范围内，只能通过 MOVLC 访问

系统时钟与复位

- 可校准 8MHz 内部 RC 振荡器
  - 可配置为 8/4/2/1MHz 系统输出
- 可校准 455KHz 低速 RC 振荡器
- 4T/2T/1T 可配置指令周期
- 32KHz 低功耗 RC 振荡器
- 外部最高 20MHz 高速晶振
- 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

LGT8PE663A 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：



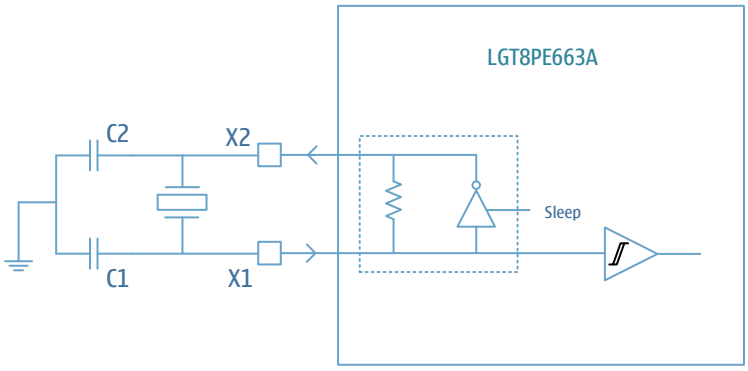
系统时钟源可通过配置字设置为来自外部的高频或者低频晶振，外部直接时钟输入以及来自内部多种振荡器时钟源。下面的表格列出了系统支持的全部时钟配置模式。配置字相关的详细定义，请参考本手册“系统配置位”章节。

时钟模式	配置描述	其他说明
RCM	系统时钟来自内部高速 RC 振荡器	可以通过 CFW:RCM 进一步配置系统时钟为 8M/4M/2M/1M 或者 455KHz
RCK	系统时钟来自内部 32KHz IRC	
HFOSC	系统时钟来自外部高速晶振	对于 10MH~20MHz 外部晶振，建议系统工作在 2T 或者 4T 指令周期模式
CLKIN	直接外部时钟源输入(RA5)	

外部时钟源

时钟模式 HFOSC 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 10MHz，建议用户同时将系统的指令周期配置为 2T 或者 4T。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。



外部晶振与电容的选择：

外部晶振	外接电容
2MHz~8MHz	C1/C2：16pF
8MHz~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
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，RA4 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， LGT8PE663A 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 8MHz RC 或者内部 455KHz 可校准时钟作为主时钟源。在 RCM 模式下，用户可以使用配置位选择系统使用 8MHz 的 1/2/4/8 分频作为系统时钟，或者使用 455KHz 的内部 RC 作为系统时钟。内部 8MHz 以及 455KHz 时钟源均为可校准时钟。一般如果客户没有特殊要求，芯片出厂后仅为 8MHz RC 振荡器校准。晶振校准后，当系统供电电压在 2.3V 到 5.5V 的工作范围内变化时，可以保证±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[2:0]	System Clock Source
010	455KHz IRC
101	8MHz/8
100	8MHz/4
111	8MHz/2 (default)
110	8MHz/1
Others	Reserved

复位控制

LGT8PE663A 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。

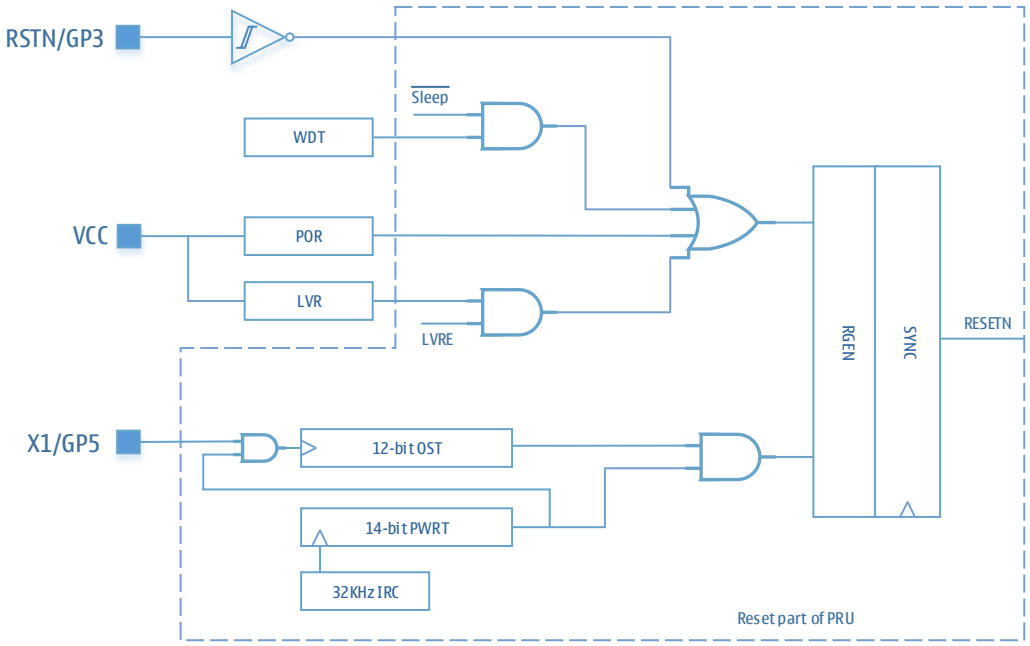
在系统默认状态下，RSTN/RA3 引脚为外部复位输入。用户可以通过配置字改变 RA3 的默认状态。关闭 RSTN/RA3 引脚的外部复位功能后， RSTN/RA3 可做为一个开漏输出的 I/O 使用。

低电压检测复位（LVR）是一个阀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 配置为适当的监控阀值，当系统供电（VCC）电压低于预设阀值时，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 的检测阀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：



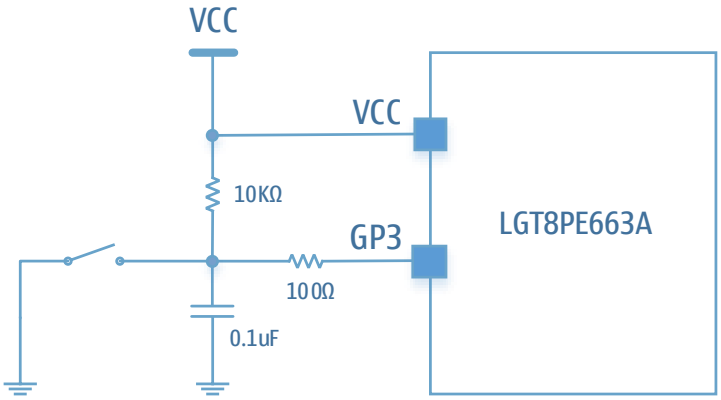
上电复位（POR）

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 阀值以下时，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，也同时会受到 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）

LGT8PE663A 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 OTP 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阀值时，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 并设置复位阀值。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)

LGT8PE663A 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	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
GPIO	05h/06h	XX00_0000	XXU0_U000
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
VRCON	19h	0X00_0000	0X00_0000
CMCON0	1Ah	0000_X0X0	0000_X0X0
CMCON1	1Ch	XXX0_0X10	XXX0_0X10
OPTION_REG	81h	1111_1111	1111_1111
TRISIO	85h	XX11_1111	XX11_1111
PIE1	8Ch	XXXX_0XX0	XXXX_0XX0
PCON	8Eh	0X00_0001	UXUU_UUUU
WPU	95h	XX11_1111	XX11_1111
IOC	96h	XX00_0000	XX00_0000
ANSEL	9Fh	XXXX_1X11	XXXX_1X11

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 OTP 烧写接口(SWD)。LGT8PE663A 的 OTP 烧写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：

$\overline{POR}$	$\overline{LVR}$	$\overline{TO}$	$\overline{PD}$	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								
8P53A 地址: 0x08					默认值: 0X01_1000			
8P609A 地址: 0x8E								
Bit	7	6	5	4	3	2	1	0
PCON	WDTE	CFOK	LVRE	DPSM1	DPSM0	SWDD	$\overline{POR}$	$\overline{BOR}$
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

Bit	Name	描述
7	WDTE	看门狗使能控制器，1：使能，0：禁止
6	CFOK	配置加载状态位，1：加载成功；0：加载失败
5	LVRE	低压检测复位使能控制，1：使能， 0：禁止
4:3	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，只能通过外部中断或外部复位唤醒
2	SWDD	置 1 将关闭 OTP 烧写端口
1	$\overline{POR}$	上电复位标志位
0	$\overline{BOR}$	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
- 引脚电平变化唤醒
- 最低 1uA@3.3V 待机功耗

综述

LGT8PE663A 实现了一个简单高效的功耗管理模块(PRU)，实现对时钟源的自动功耗控制。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。LGT8PE663A 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。

LGT8PE663A 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，比如模拟比较器，定时/计数器等。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 PMOD 位控制，功耗管理模块可以根据 PMOD 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。PMOD 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	AC	TMR1	唤醒源
00	X	(1)	(2)	(2)	√	√	(2)	(2)	外部复位 所有中断(2)
01	X	(1)	(2)	(2)	X	√	(2)	(2)	外部复位 所有中断(3)
1X	X	(1)	X	X	X	X	X	X	外部复位(4) 引脚变化中断 外部 INT(5)

(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 PMOD 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 PMOD=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)：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)：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；

(4)：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；

(5)：系统支持 INT 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，LGT8PE663A 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。此模式只能通过外部中断引脚(INT)，引脚电平变化或者外部复位唤醒。

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。LGT8PE663A 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，避免产生漏电通路。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。

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 INTEDGE 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：LGT8PE663A 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, 4         ; enable deep sleep mode (DPSM[1]=1)
BCF         TRISIO, 4       ; RA4 as output (toggle indicator)
BCF         OPTION_REG, 6   ; INTEDG=0, low level wakup on INT
```

```
        MOVLW   0x3F            ;
        MOVWF   WPU             ; enable all pullup
        BCF     OPTION_REG, 7   ; global pullup enable

        BANSEL  GPIO            ; select bank 0
        CLRF    GPIO            ; clear I/O status
        MOVLW   0x10            ; use to toggle RA4

LOOP    XORWF   GPIO            ; toggle RA4
        BTFSS   GPIO, 5         ;
        SLEEP                   ; goto sleep if RA5 = 0
        GOTO    LOOP
END                             ; support by MPASM assembly
```

输入输出

- 可编程内部上/下拉电阻
- 可编程开漏输出控制
- 引脚变化中断

综述

LGT8PE663A 最多支持 8 路可编程输入/输出端口(SOP10L 封装)。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。

除 RA3/RSTN 引脚外，其他的 I/O 都支持可编程的上下拉电阻以及开漏输出控制。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 TRISIO 寄存器控制。当设置 TRISIO 某位为 1，对应的 I/O 为输入模式（默认），清零 TRISIO 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。

TRISIO 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，TRISIO 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     GPIO        ; select bank 0
CLRF        GPIO        ; init GPIO
BANKSEL     ANSEL       ; select bank 1
CLR         ANSEL       ; disable analog function

MOVLW       0x3C        ;
MOVWF       TRISIO      ; set GP[1:0] as output
```

端口附加功能

LGT8PE663A 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

LGT8PE663A 内部集成一个多输入的模拟比较器，可接受来自外部端口的模拟信号输入。ANSEL 寄存器用于控制比较器相关输入端口的状态。RA1/4 可作为比较器的负端输入，RA0 可作为比较器的正端输入。由于 RA0/1 同时复用为 OTP 的编程控制接口(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。一旦配置为模拟功能 I/O，SWD 接口将被关闭。RA4 默认状态下是作为模拟端口。

比较器相关 I/O 被设置为模拟端口功能后，通过 GPIO 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 TRIS 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。

上拉/下拉电阻

除 RA3 以外，其他的 I/O 都具有可编程的内部上拉/下拉电阻。下拉电阻可以通过 PDCR 寄存器控制。每个 I/O 的上拉电阻可以通过 PUC 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。所有引脚(RA3 除外)的上拉被全局上拉控制位(GDPU)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，内部上拉可以通过寄存器控制。RA6/7 两个 I/O 内部为 6K 左右强上拉，其他 I/O 的上拉为 25K 左右弱上拉。

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

每个 I/O 都可以通过 IOCR 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/INTEN 寄存器的 GPIE 位控制。软件可以通过 INTCON/INTFR 寄存器 GPIF 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。

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 OTP 中的数据。烧写 OTP 还需要通过 RA3/VPP 提供一个高压源。OTP 烧写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 OTP 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 OTP 的 VPP 端口。

RA0/2 在不同内核模式下分别可作为外部中断输入端口。当系统配置为 8P53A 模式时，RA0 作为外部中断的输入，当配置为 8P609A 模式，RA2 将作为外部中断输入。

RA6/7 在 SOP10L 封装下可作为独立的 I/O 访问。 对于 SOP8L 封装，RA7 与 RA1 同时被绑定到 SOP8 的第 6 脚；RA6 与 RA2 同时被绑定到 SOP8 的第 5 脚。使用时请特别注意，需要根据所使用的功能，正确的配置。比如，在 SOP8L 封装下使用 SCL/SDA 的内部上拉， 需要通过 PUC6/7 进行控制，使用 PUC1/2 只能打开 RA1/2 内部的弱上拉电阻。

寄存器定义

GPIO/PORT – 端口数据寄存器

GPIO – 端口数据寄存器								
8P53A 地址: 0x06					默认值: XXXX_XXXX			
8P609A 地址: 0x05								
Bit	7	6	5	4	3	2	1	0
GPIO	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						

TRISIO – 端口方向控制寄存器

GPIO – 端口方向控制寄存器								
8P53A 地址: TRIS 指令访问					默认值: 1111_1111			
8P609A 地址: 0x85								
Bit	7	6	5	4	3	2	1	0
TRISIO	TRIS7	TRIS6	TRIS5	TRIS4	TRIS3	TRIS2	TRIS1	TRIS0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	TRIS7..0	设置端口的输入/输出方向 1：端口为输入 0：端口为输入/输出						

PDCR – 端口下拉控制寄存器

PDCR – 端口下拉控制寄存器								
8P53A 地址: 0x0B					默认值: 0000_X000			
8P609A 地址: 0x97								
Bit	7	6	5	4	3	2	1	0
PDCR	PDC7	PDC6	PDC5	PDC4	-	PDC2	PDC1	PDC0
R/W	W/R	W/R	W/R	W/R	-	W/R	W/R	W/R
Bit	Name	描述						
7:0	PDC7..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						

PUCR – 端口上拉控制寄存器

PUCR/WPU- 端口上拉控制寄存器								
8P53A 地址: 0x0D					默认值: 1111_1111			
8P609A 地址: 0x95								
Bit	7	6	5	4	3	2	1	0
PUCR	PUC7	PUC6	PUC5	PUC4	PUC3	PUC2	PUC1	PUC0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	PUC7..0	端口上拉控制 1：使能端口上拉电阻 0：禁止端口上拉电阻						

IOCR – 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								
8P53A 地址: 0x09					默认值: 0000_X000			
8P609A 地址: 0x96								
Bit	7	6	5	4	3	2	1	0
IOCR	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	IOC7..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						

ODCR – 端口开漏输出控制寄存器

ODCR – 端口开漏输出控制寄存器								
8P53A 地址: 0x0C					默认值: XX00_X000			
8P609A 地址: N/C								
Bit	7	6	5	4	3	2	1	0
IOCR	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	ODC7..0	端口开漏输出控制 1：使能端口开漏输出功能 0：禁止端口开漏输出功能						

中断控制

- 软中断(INT 指令支持)
- 外部中断(RA2/INT*)
- 引脚电平变化中断
- 比较器中断
- Timer0/1 溢出中断
- ECP 中断
- TWI/EEP 控制器中断

综述

LGT8PE663A 最多支持 7 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为。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

LGT8PE663A 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。

根据 LGT8PE663A 的运行模式，中断向量分配如下：

中断类型	8P53A 模式	8P609A 模式
软件中断(INT 指令)	0x001	0x001
硬件中断(外设请求)	0x008	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 LGT8PE663A/8P609A 模式的中断功能：

```
        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
```

RA02/INT 外部中断

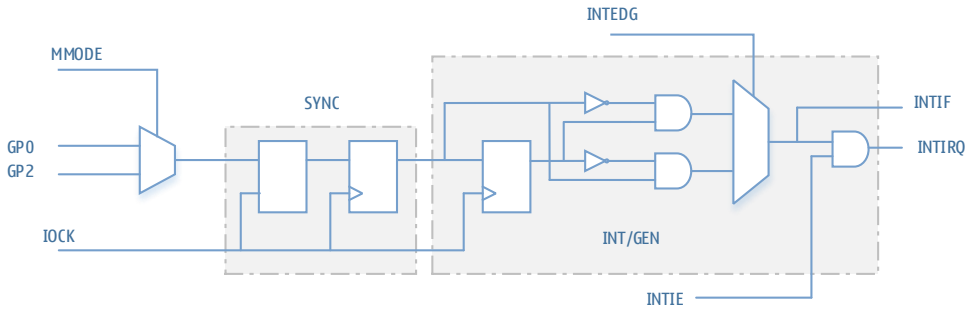
外部中断 INT 为边沿触发，触发边沿通过 OPTION 寄存器的 INTEDG 位设置。当 RA02 上发生有效的边沿的变化，INTCON 寄存器的 INTIF 位被置位。RA02/INT 上的中断功能可以通过 INTCON 寄存器的 INTIE 位禁止。INTIF 位必须在再次使能 GIE 位之前清除，否则会产生重复的中断。

RA02/INT 可用于休眠唤醒。特别是在深休眠模式下，只能通过 RA02/INT 或者外部复位唤醒。作为唤醒功能，RA02/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。RA02/INT 的唤醒功能和中断无关，即使没有使能 INTIE，RA02/INT 仍可作为唤醒功能。

LGT8PE663A 的外部中断输入在不同系统模式下被分配到不同的 I/O 上，在 8P53A 模式下，外部中断 INT 从 RA0 输入；在 8P609A 模式下，外部中断 INT 从 RA2 输入。

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：

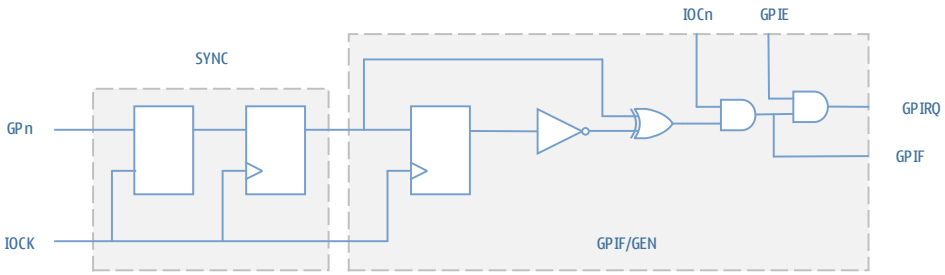


端口电平变化中断

LGT8PE663A 的 GPIO 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/INTEN 寄存器的 GPIE 位独立控制。单个引脚的电平变化功能也可以通过 IOCR 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

当定时器 0 计数器(TMR0))计数到 PR0(L/H)时，将会置位 INTCON 寄存器的 T0IF 中断标记位。如果此时使能了 TMR0 溢出中断(INTCON:T0IE)，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 T0IE 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。

定时器 1 溢出中断

当定时器 1 计数器(TMR1)计数到 PR1(L/H)时，将会置位 INTCON 寄存器的 T1IF 中断标记位。如果此时使能了 TMR1 溢出中断(INTCON:T1IE)，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 1 的溢出中断可以通过置位/清零 INTCON 寄存器的 T1IE 位实现使能/禁止控制。定时器 1 的相关细节，请查看本手册相关章节。

ECP 控制器中断

增强俘获/PWM 控制器可用于俘获来自外部 CCP1 引脚的电平变化事件。当控制器俘获到目标事件，将会触发置位 ECPIF 中断标志位。事件触发时的定时器结果被保存到 DC1H/L 中。增强 PWM 发生器支持自动关闭功能。用户可以根据应用需求，选择合适的自动关闭触发源。当自动关闭事件发生后，ECP 控制器将关闭 PWM 输出，同时置位 ECPIF 中断标志位。如果软件同时使能了 ECPIE 中断使能位，外设中断使能位(PEIE)基于全局中断使能 GIE，系统将在执行完当前指令后响应 ECP 中断请求。用户需要在中断服务程序中清零 ECPIF 标志位。

TWI/EEP 控制器中断

TWI 控制器与 EEP 控制器共享 TWIF 中断标志位。EEP 为 E2PROM 接口控制器，可用于访问支持 I2C 接口的内部或外部 E2PROM 存储芯片。EEP 控制器构建在 TWI 接口之上，TWI 作为 EEP 模块的底层通讯接口。EEP 模块只负责处理 E2PROM 的读写控制。TWI 可作为独立的 I2C 接口控制器使用。可配置位主机或者从机设备。当控制器检测到总线上发生的有效传输后，将会触发 TWIF 中断标志位。用户可以使用 TWIF 作为驱

动实现通讯的链路控制。当软件使能了 EEP 功能模块后，TWIF 标志位实际为 EEP 模块控制，用于指示 EEP 模块的工作状态。具体请参考 TWI 控制器以及 EEP 控制章节。

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 8P609A 模式为分页寻址，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。LGT8PE663A 将 64 字节的 SRAM 映射到每个月面的 0x40~0x7F 的区域，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 空间最后 16 字节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 (8P609A)
HISR        MOVWF   W_TEMP      ; copy W to TEMP regiser
            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/INTEN – 中断控制寄存器

INTCON/INTEN– 中断控制寄存器								
8P53A 地址: 0x0E					默认值: 0000_0000			
8P609A 地址: 0x0B/0x8B								
Bit	7	6	5	4	3	2	1	0
INTCON	GIE	-	T0IE	INTIE	GPIE	T0IF	INTIF	GPIF
R/W	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	-	Unimplemented
5	T0IE	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	GPIE	引脚电平变化中断使能控制位
2	T0IF	定时器 0 溢出中断标记位
1	INTIF	外部中断标记位
0	GPIF	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**								
8P53A 地址: N/C					默认值: XXXX_0XX0			
8P609A 地址: 0x8C								
Bit	7	6	5	4	3	2	1	0
PIE1	TWIE	-	ECPIF	-	CMIE	-	-	T1IE
R/W	W/R	-	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	TWIE	I2C/EEM 控制器中断使能						
6	-	Unimplemented						
5	ECPIE	ECP 控制器中断使能						
3	CMIE	模拟比较器中断使能						
2	-	Unimplemented						
1	-	Unimplemented						
0	T1IE	定时器 1 溢出中断使能						

PIR1 – 外设中断标记寄存器

PIR1 **– 外设中断标记寄存器**								
8P53A 地址: N/C					默认值: XXXX_0XX0			
8P609A 地址: 0x0C								
Bit	7	6	5	4	3	2	1	0
PIR1	-	-	-	-	CMIF	-	-	T1IF
R/W	W/R	-	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	TWIF	I2C 控制器中断标志位						
6	-	Unimplemented						
5	ECPIF	ECP 中断标志位						
4	-	Unimplemented						
3	CMIF	比较器中断标志位						
2:1	-	Unimplemented						
0	T1IF	定时器 1 溢出中断标记位						

看门狗定时器

- 独立内部 32KHz IRC
- 8 位预分频器(与 TMR0 共享)
- 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。即便在休眠模式下，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 PCON 寄存器的 WDTE 位进行控制。即使在配置字禁用 WDT 的情况下，仍然可以通过 PCON 的 WDTE 寄存器将其使能。

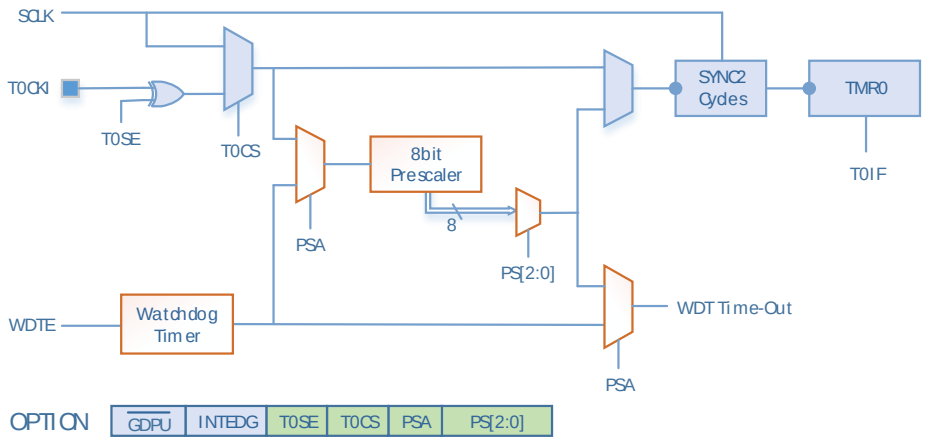
看门狗定时器正常模式下超时时间为 18MS。超时时间与温度以及系统供电电压也有关系。WDT 超时时间与电压变化的关系，请参考本手册电气特性部分。

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 (2.3s)。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。

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$\overline{TO}$位被清零。软件可以通过 STATUS 寄存器的$\overline{TO}$/$\overline{PD}$位判定系统复位是否来自 WDT 超时。

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

看门狗定时器结构图：



寄存器定义

OPTION – 外设配置寄存器

OPTION– 外设配置寄存器								
8P53A 地址: 仅通过 OPTION 指令访问					默认值: 1111_1111			
8P609A 地址: 0x81								
Bit	7	6	5	4	3	2	1	0
OPTION	$\overline{GPPU}$	INTEDG	T0CS	T0SE	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	$\overline{GPPU}$	GPIO 全局上拉禁止控制 1：全局禁用上拉 0：全局上拉使能
6	INTEDG	外部中断/唤醒边沿选择 1: 上升沿触发中断 0：下降沿触发中断
5	T0CS	Timer0 时钟源选择
4	T0SE	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

10 位定时计数器 0

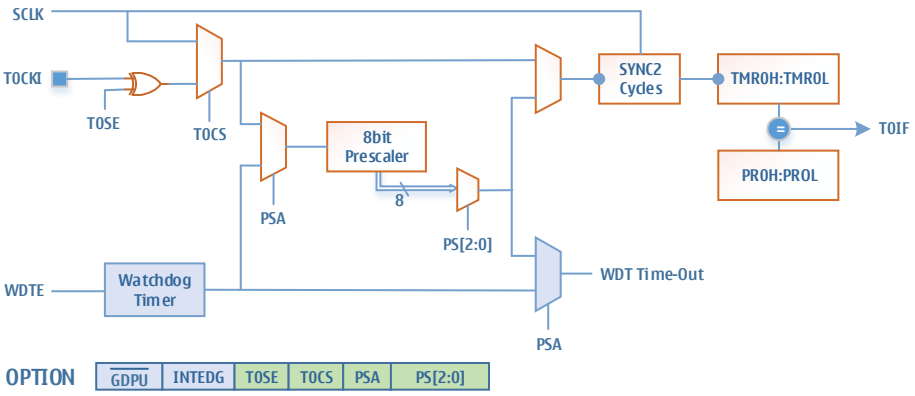
- 10 位定时/计数寄存器(TMR0H:TMR0L)
- 10 位计数周期寄存器器(PR0H:PR0L)
- 8 位预分频器(与 WDT 共享)
- 可编程内部/外部时钟源
- 可编程外部时钟边沿选择
- 溢出中断

综述

TMR0 模块支持两种工作模式：10 位定时器/10 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 T0IF 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 T0CS 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 T0CS 位置 1，TMR0 工作于计数器模式。在计数器模式下，TMR0 寄存器在 T0CKI 的边沿驱动下递增计数。T0CKI 的边沿通过 OPTION 寄存器的 T0SE 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 TMR0 后，任何对 TMR0 寄存器的写操作都将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 TMR0 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 TMR0 之前正确安全的切换预分频器。

预分频器从 TMR0 切换至 WDT:

```
BANKSEL     TMR0            ; select bank 0
CLRWDT                      ; Clear WDT
CLRF        TMR0            ; 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 TMR0:

```
CLRWDT                      ; clear WDT and prescaler

BANKSEL     OPTION_REG      ;
MOVLW       B'1111_0000     ; mask TMR0 select and
ANDWF       OPTION_REG, W   ; prescaler bits
IORLW       b'0000_0011     ; set perscaler to 1:16
MOVWF       OPTION_REG      ;
```

TMR0 中断处理

TMR0 在计数与(PR0H:PR0L)相等时，T0IF 为在溢出中断发生时。无论系统有没有使能 T0IE，此中断标记位仍然会被置位为 1。在休眠模式下，TMR0 工作时钟被关闭，无法产生用于休眠唤醒中断的信号。

16 位定时计数器 1

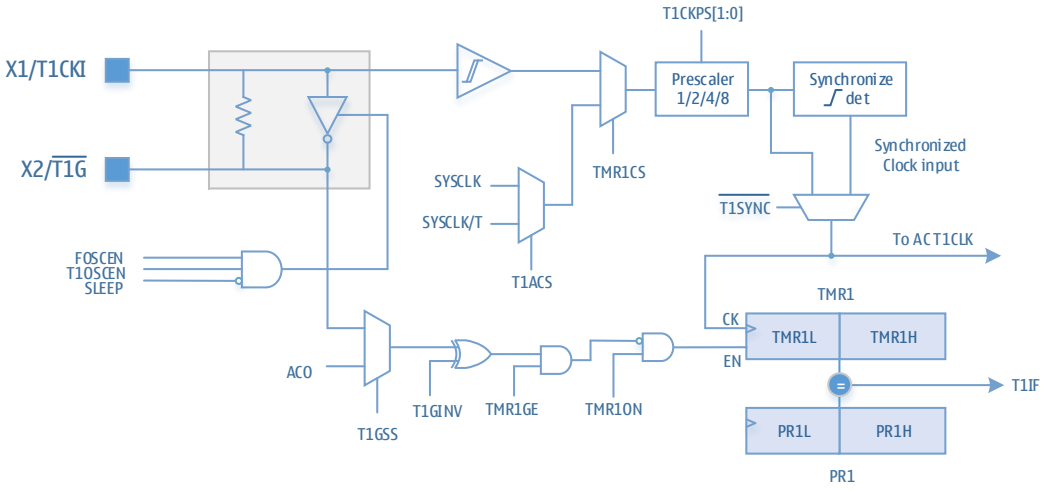
- 16 位定时/计数寄存器(TMR1H:TMR1L)
- 可编程内部或者外部时钟源
- 3 位独立预分频
- 可选外部低频晶振输入(LFOSC)
- 同步/异步模式
- TMR1 计数门控支持：比较器或/T1G 引脚
- 计数溢出中断
- 溢出唤醒(外部时钟模式或异步模式)
- 比较器输出与 TMR1 时钟同步

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)同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保持运行。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 LGT8PE663A 相关应用手册。

溢出中断

当 TMR1 计数到与 PR1H:PR1L 相等时，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一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								
8P53A 地址: N/C					默认值: 0000_0000			
8P609A 地址: 0x10								
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

PR1H – TMR1 定时周期寄存器高字节

PR1H **– TMR1 周期寄存器高字节**								
8P53A 地址: N/C					默认值: 0000_0000			
8P609A 地址: 0x89								
Bit	7	6	5	4	3	2	1	0
PR1H	PR1H[7:0]							
R/W	R/W							
Bit	Name	描述						
7:0	PR1H	PR1 定时周期寄存器高字节						

PR1L – TMR1 周期寄存器低字节

PR1L **– TMR1 周期寄存器低字节**								
8P53A 地址: N/C					默认值: 0000_0000			
8P609A 地址: 0x88								
Bit	7	6	5	4	3	2	1	0
PR1L	PR1L[7:0]							
R/W	R/W							
Bit	Name	描述						
7:0	PR1L	TMR1 定时周期寄存器低字节						

TMR1H – TMR1 计数寄存器高字节

TMR1H **– TMR1 计数寄存器高字节**								
8P53A 地址: N/C					默认值: 0000_0000			
8P609A 地址: 0x0F								
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								
8P53A 地址: N/C					默认值: 0000_0000			
8P609A 地址: 0x0E								
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

PIE1 – 外设中断使能控制寄存器

T1IE 定义请参考本手册中断控制|寄存器定义部分

PIR1 – 外设中断标记寄存器

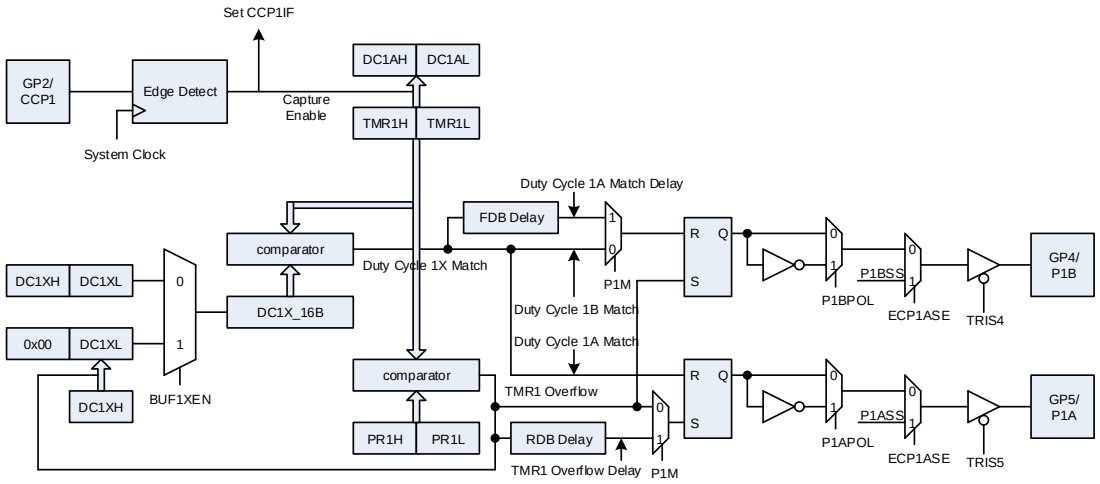
T1IF 定义请参考本手册中断控制|寄存器定义部分

增强俘获/PWM 模块

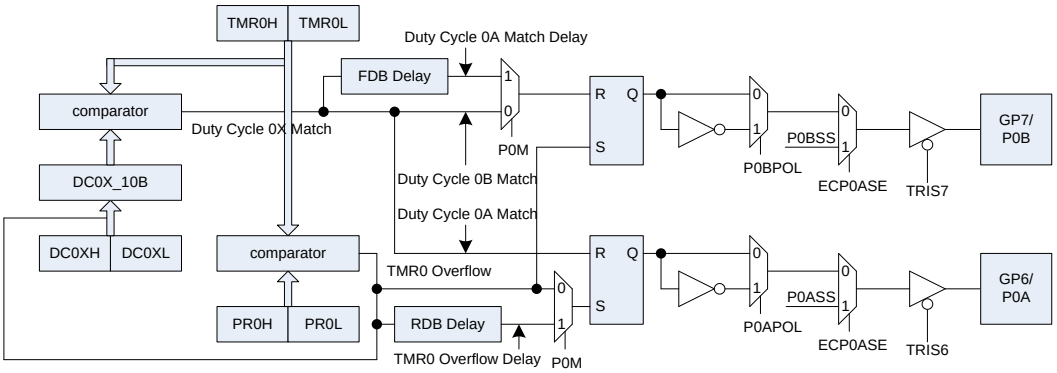
- 外部事件俘获模式
- 可编程 PWM 输出
- 增强 PWM 模式：支持半桥驱动以及自动关闭

综述

增强俘获 PWM 控制(ECP)用于实现对多种外部事件的俘获和控制。在俘获模式下，可实现对外部事件周期的计数。PWM 模式可实现可编程频率以及占空比的脉宽调制信号。增强 PWM 模式可直接产生用于驱动半桥电路的 PWM 输出，并支持死区控制以及多种硬件事件触发的自动关闭功能。



基于 TMR1 的俘获和 PWM1 结构图



基于 TMR0 的 PWM0 结构图

俘获模式

在俘获模式下，当发生在 CCP1 引脚的目标事件被俘获后，DC1AH:DC1AL 保存 16 位的计数结果。目标事件的类型可以通过 ECP1CON 寄存器的 CAPM 位在以下两种模式中选择:

- 任意 CCP1 上升沿事件
- 任意 CCP1 下降沿事件

为实现正确的俘获来自 CCP1 引脚的外部事件，需要首先通过 TRISI 寄存器将 CCP1 所在的 IO 设置为输入模式。

俘获发生后，俘获中断标志位 ECPIF 被置位。中断标志位必须通过软件清零。如果在本次俘获发生后，没有及时读取 DC1AH:DC1AL 的计数结果，此结果将会被接下来产生的俘获事件取代。在改变俘获模式时，可能会产生一个无效的俘获事件，用户需要在设置俘获中断时特别注意，避免系统响应配置过程产生的无效事件。同时，用户需要在每次响应俘获事件后及时的清零 ECPIF 中断标志位。

PWM 模式

ECP 模块可以实现最多 4 路 PWM 输出。PWM 输出被分配到 P0A/P0B/P1A/P1B 引脚。P0A/B 输出基于 TMR0 的计数器产生；P1A/B 基于 TMR1 的计数器产生。

为简化描述，下文中的'n'表示定时器 0/1 其中之一；'x'用于表示 PWM 输出通道 A/B 之一。PWM 输出引脚与通用 I/O 端口以及其他功能引脚复用，因此为正确的输出 PWM 信号，需要通过 TRISIO 寄存器将对应 IO 配置为输出模式，同时使能 ECPnCON 寄存器的 PnxOEN 位。如需产生一路单独的 PWM 输出，可将 PWMnCON 寄存器的 PWMnM 位清零。

PWM 输出相关的周期，占空比以及分辨率由以下寄存器决定:

- PRn (PRnH:PRnL)
- DCnA (DCnAH:DCnAL)
- DCnB (DCnBH:DCnBL)

PWM 周期通过 PRn 寄存器控制，可以通过如下公式计算:

PWM 周期 = [(PRn) + 1] * Tosc

当定时计数器(TMRn)计数达到 PRn 时，将会在下一个计数周期发生如下事件:

- TMRn 寄存器被清零
- PWM 输出高
- 如果使能了缓存模式，PWM1x 占空比将会从 DC1AH/DC1BH 加载到 DC1AL/DC1BL

PWM 占空比由 DCnA/DCnB 寄存器指定。DCnA/DCnB 寄存器可以在任意时刻更新。P0A/B 的 PWM 占空比由 DC0A/B 直接确定。相比之下，P1A/B 支持缓存模式，当设置了 ECP1CON 寄存器的 BUF1AE/BUF1BE 位，将使能 PWM1 的缓存模式。此模式下 DC1AL/DC1BL 的值仅在完成一个计数周期后才会从 DC1AH/DC1BH 中加载。缓存模式仅适用于 PWM 分辨率低于 8 位的应用中，因为占空比的高 8 位将会被固定保持为 0。当禁用了 PWM1 的缓存模式，PWM1 的占空比将直接由 DC1A/B 寄存器确定。

PWM 输出脉冲宽度可以由如下公式计算:

PWM 脉宽 = (DCnx) * Tosc

PWM 输出占空比可以由如下公式计算:

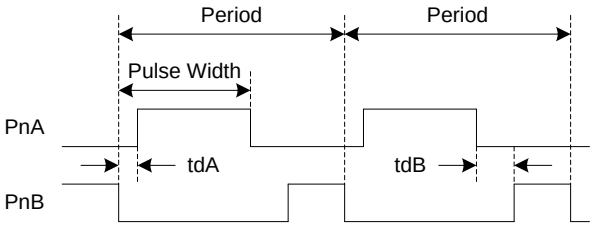
PWM 占空比 = (DCnx) / [(PRn) + 1]

增强 PWM 模式

增强 PWM 模式可产生直接驱动半桥电路的脉宽输出。可通过置位 PWMnCON 寄存器的 PWMnM 位使能增强 PWM 模式。 PWM 信号输出在 PnA 引脚，另外一路互补的 PWM 输出在 PnB 引脚。PWM 输出引脚与通用 I/O 端口以及其他功能引脚复用，因此为正确的输出 PWM 信号，需要通过 TRISIO 寄存器将对应 IO 配置为输出模式，同时使能 ECPnCON 寄存器的 PnxOEN 位。如需产生一路单独的 PWM 输出，可将 PWMnCON 寄存器的 PWMnM 位清零。

半桥驱动模式下， 可以通过可编程的死区时间控制互补 PWM 输出的相位关系，以避免驱动器件的开关延迟造成短路。PWMnCON 寄存器的 PWMnxDB 位用于设置在驱动 PWM 输出前插入的系统指令周期数。如果设置的死时间值大于占空比的设置，对应的输出将会在整个周期内变为无效。

下图为半桥模式下的 PWM 输出。输出信号为高电平有效：



PWM 周期通过 PRn 寄存器控制，可以通过如下公式计算:

PWM 周期 = [(PRn) + 1] * Tosc

PWM 输出 A 路的脉冲宽度可以由如下公式计算:

PnA 脉宽 = (DCnA) * Tosc

PWM 输出 B 路的脉冲宽度可以由如下公式计算:

PnB 脉宽 = PWM 周期 – PnA 脉宽 – TdA – TdB

TdA 于 TdB 为设置的死时间。TdA 为 PnA 输出从非有效状态到有效状态的死时间，由 PWMnCON 寄存器的 PWMnADB 位设定。TdB 为 PnB 输出从非有效状态到有效状态的死时间，由 PWMnCON 寄存器的 PWMnBDB 位设定。死时间可以通过如下公式计算:

Tdx = (PWMnxDB) * Tosc

自动关闭与重启

PWM 模式支持自动关闭模式。自动关闭由可选的外部事件触发。PWM 自动关闭后，输出处于输入高阻模式。这种模式可用于保护 PWM 驱动的外部电路。

自动关闭模式可以通过 ECPnAS 寄存器的 CMPnSEN 与 INTnSEN 位选择控制。可选的关闭事件为:

- INT 引脚的低电
- 比较器输出翻转
- 软件通过 ECPnASE 位触发

PWM 的工作状态可以通过 ECPnAS 寄存器的 ECPnASE 位获得。如果此位为 0，表示 PWM 输出正常；如果此位为 1，表示 PWM 已处于关闭状态。.

当一个自动关闭事件发生时，会同时产生以及影响:

1) ECPnASE 位置 1。ECPnASE 为将会保持为 1，直到使用软件清零或者发生自动重启；.
2) 中断标志位 ECPIF 被置位。中断标志位必须通过软件清零；
3) PWM 输出处于关闭状态。此时 PWM 引脚的状态可以通过 ECPnAS 寄存器的 PSSnx 位进行控制，可被设置为以下三种状态:
   - 驱动为高电平
   - 驱动位低电平
   - 输入高阻

增强 PWM 模式下，当自动关闭时间清除后，可自动恢复（重启）。此功能可以通过 PWMnCON 寄存器的 PSRnEN 位设置。

使能了自动重启功能后，如果自动关闭条件仍然有效，ECPnASE 位将会保持为 1。当自动关闭条件消失后，ECPnASE 位将会被硬件自动清零，并恢复正常工作。

寄存器定义

DC0AL – PWM0 通道 A 占空比控制寄存器低 8 位

DC0AL		地址:0x11	初始值:0x00
Bit	Name	描述	
7:0	DC0AL	PWM0 通道 A 占空比寄存器低 8 位	

DC0BL – PWM0 通道 B 占空比控制寄存器低 8 位

DC0BL		地址:0x12	初始值:0x00
Bit	Name	描述	
7:0	DC0BL	PWM0 通道 B 占空比寄存器低 8 位	

ECP0CON – ECP0 控制寄存器

ECP0CON		地址:0x15	初始值:0x00
Bit	Name	描述	
7	P0AOEN	P0A 输出使能。1 = 使能 P0A 输出 PWM	
6	P0BOEN	P0B 输出使能。1 = 使能 P0B 输出 PWM	
5:4	DC0AH	TMR0 通道 A 占空比最高 2 位	
3:2	DC0BH	TMR0 通道 B 占空比最高 2 位	
1	P0APOL	P0A 输出极性选择。 1 = 低有效；0= 高有效	
0	P0BPOL	P0B 输出极性选择。 1 = 低有效；0 = 高有效	

PWM0CON – PWM0 控制寄存器

PWM0CON		地址:0x16	初始值:0x00
Bit	Name	描述	
7	PWM0M	P0A/B 输出模式选择位： 1 = 半桥驱动模式(P0A/B) 0 = 单路输出模式(P0A)	
6:4	PWM0ADB	P0A 死时间寄存器 P0A 进入到有效输出状态插入的死时间	
3:0	PWM0BDB	P0B 死时间寄存器 P0B 进入到有效输出状态插入的死时间	

ECP0AS – PWM0 自动关闭控制寄存器

ECP0AS		地址:0x17	初始值:0x00
Bit	Name	描述	
7	ECP0ASE	PWM0 自动关闭状态位 1 = 已关闭状态。P0A/B 输出处于关闭状态 0 = P0A/B 处于正常工作状态	
6	PRS0EN	PWM0 自动重启使能位 1 = 使能自动重启。待关闭条件清楚后，ECP0ASE 位自动清零。 0 = 关闭自动重启。ECP0ASE 位必须由软件清除。	
5	CMPS0E	使能比较器输出作为自动关闭条件 1 = 比较器输出翻转作为自动关闭条件 0 = 禁用比较器输出作为自动关闭条件	
4	INTS0EN	外部中断输入作为自动关闭条件 1 = INT 输入低电平作为自动关闭条件 0 = 禁用 INT 作为自动关闭条件	
3:2	PSS0A	P0A 关闭状态控制位: 1x = P0A 输入高阻 01 = 驱动 P0A 为高电平 00 = 驱动 P0A 为低电平	
1:0	PSS0B	P0B 关闭条件控制位: 1x = P0B 输入高阻 01 = 驱动 P0B 为高电平 00 = 驱动 P0B 为低电平	

DC1AL – PWM1 通道 A 占空比控制寄存器低 8 位

DC1AL		地址:0x13	初始值:0x00
Bit	Name	描述	
7:0	DC1AL	PWM1 通道 A 占空比寄存器低 8 位	

DC1AH – PWM1 通道 A 占空比控制寄存器高 8 位

DC1AH		地址:0x14	初始值:0x00
Bit	Name	描述	
7:0	DC1AH	PWM1 通道 A 占空比寄存器高 8 位	

DC1BL – PWM1 通道 B 占空比控制寄存器低 8 位

DC1BL		地址:0x93	初始值:0x00
Bit	Name	描述	
7:0	DC1BL	PWM1 通道 B 占空比寄存器低 8 位	

DC1BH – PWM1 通道 B 占空比控制寄存器高 8 位

DC1BH		地址:0x94	初始值:0x00
Bit	Name	描述	
7:0	DC1BH	PWM1 通道 B 占空比寄存器高 8 位	

ECP1CON – ECP1 控制寄存器

ECP1CON		地址:0x1B	初始值:0x00
Bit	Name	描述	
7	P1AOEN	P1A 输出使能。1 = 使能 P1A 输出	
6	P1BOEN	P1B 输出使能。1 = 使能 P1B 输出	
5	CAPEN	俘获功能控制位。 1 = 使能俘获功能；DC1AH:DC1AL 工作于俘获模式 0 = 禁用俘获功能；DC1AH:DC1AL 工作于 PWM 模式	
4	CAPM	俘获模式选择位： 1 = CCP1 上升沿 0 = CCP1 下降沿	
3	BUF1AE	DC1AH 缓存功能使能位 1 = DC1AH 作为 DC1AL 的缓存，TMR1 通道 A 占空比高 8 位为 0 0 = DC1AH 作为 TMR1 通道 A 占空比的高 8 位	
2	BUF1BE	DC1BH 缓存功能使能位 1 = DC1BH 作为 DC1BL 的缓存，TMR1 通道 B 占空比高 8 位位 0 0 = DC1BH 作为 TMR1 通道 B 占空比的高 8 位	
1	P1APOL	P1A 输出极性选择位： 1 = 低为有效输出 0 = 高为有效输出	
0	P1BPOL	P1B 输出极性选择位： 1 = 低为有效输出 0 = 高为有效输出	

PWM1CON – PWM1 控制寄存器

PWM1CON		地址:0x18	初始值:0x00
Bit	Name	描述	
7	PWM1M	P1A/B 输出模式选择位: 1 = 半桥驱动模式(P1A/B) 0 = 单路输出模式(P1A)	
6:4	PWM1ADB	P1A 死时间寄存器 P1A 进入到有效输出状态插入的死时间	
3:0	PWM1BDB	P1B 死时间寄存器 P1B 进入到有效输出状态插入的死时间	

ECP1AS – PWM1 自动关闭控制寄存器

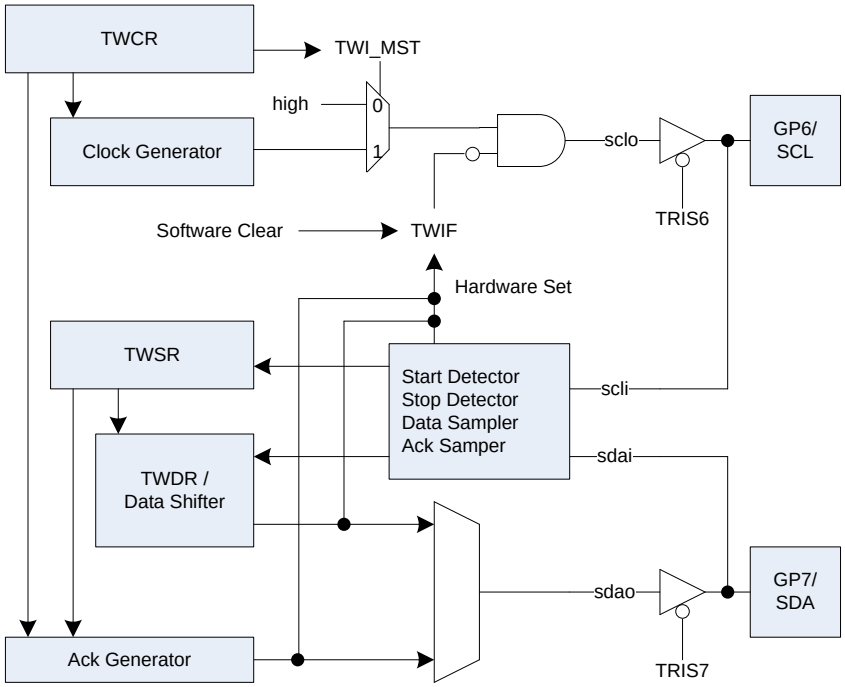
ECP0AS		地址:0x1D	初始值:0x00
Bit	Name	描述	
7	ECP1ASE	PWM1 自动关闭状态位 1 = 已关闭状态。P1A/B 输出处于关闭状态 0 = P1A/B 处于正常工作状态	
6	PRS1EN	PWM1 自动重启使能位 1 = 使能自动重启。待关闭条件清楚后，ECP1ASE 位自动清零。 0 = 关闭自动重启。ECP1ASE 位必须由软件清除。	
5	CMPS1E	使能比较器输出作为自动关闭条件 1 = 比较器输出翻转作为自动关闭条件 0 = 禁用比较器输出作为自动关闭条件	
4	INTS1EN	外部中断输入作为自动关闭条件 1 = INT 输入低电平作为自动关闭条件 0 = 禁用 INT 作为自动关闭条件	
3:2	PSS1A	P1A 关闭状态控制位: 1x = P1A 输入高阻 01 = 驱动 P1A 为高电平 00 = 驱动 P1A 为低电平	
1:0	PSS1B	P1B 关闭条件控制位: 1x = P1B 输入高阻 01 = 驱动 P1B 为高电平 00 = 驱动 P1B 为低电平	

TWI 主从控制器（I2C）

- 支持主机/从机模式
- 支持传输速率设置

综述

TWI 控制器是一种双线通讯接口控制器。协议实现兼容 I2C 总线。可用于与其他使用 I2C 接口的外设通讯。



TWI 结构框图

工作模式

如需要 TWI 控制器工作为从机模式，需要置位 TWCR 寄存器的 TWEN。如需要工作为 I2C 主机模式，需要同时设置 TWEN 位与 TWMST 位，然后通过 TWCR 寄存器的 CKPS 位设置产生通讯所需的 SCL 时钟信号。

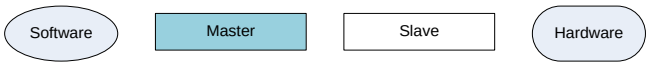
当接收到一个开始条件，重启位，一个停止位，8 位数据或一个响应位，控制器将会置位 TWSR 寄存器的对应状态信息位。软件可以通过读取 TWSR 寄存器获得接收信息，更新状态以及相关操作，然后通过写 0 清除相关的标记位。

当需要发生一个 8 位数据，响应位或者主机发生一个开始或者停止条件，软件需要设置 TWSR 寄存器中相应的标志位，之后控制器会做出相应的动作并清除标志位。当主机发送一个开始位或者停止位时，控制位也会同时接收到这个开始或者停止位。

当接收到一个开始位或一个停止位或 8 位数据或 1 位响应或者是发送一个响应位，控制器会同时置位 PIR1 寄存器中的 TWIF 中断标志位。TWIF 被置位后，控制器将 SCL 保持为低电平直到 TWIF 会被清零。

操作模式

TWI 控制可工作于 4 种主要模式。这些模式包括：主机发送(MT)，主机接收(MR)，从机发送送(ST)，从机接收(SR)。下面的章节将会分别描述这些模式。在下面的图示中，"Software"表示软件相关操作；"Hardware"表示硬件相关操作；"Master"表示主机操作；"Slave"表示从机操作；"SLA"表示从机地址；"W"表示写操作(SDA 低电平)；"R"表示读操作(SDA 高电平)。"ACK"表示响应位；"NACK"表示无响应。



主机发送模式

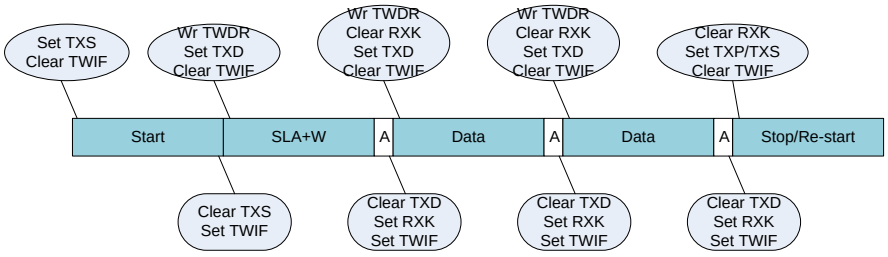
如需工作在主机发送模式，首先，TWI 需要配置为主机工作模式。

通过置位 TWSR 寄存器的 TXS 位并清零 TWIF 中断状态，控制器将会产生在总线空闲时产生一个传输开始条件(START)。开始条件传输完成后，TWIF 状态位被置位，同时 TXS 位被清零。

接下来， 需要发送一个 SLA+W。这个操作可以通过向 TWDR 寄存器中写入 SLA+W 完成，设置 TWSR 寄存器的 TXD 位，清零 TWIF。当 SLA+W 发送完成并接收到一个响应位后，TWIF 位再次被置位，TXD 位清零，RXK 位设置以表明当前已收到正确的响应信息。

SLA+W 成功传输后，接下来可以发送数据。将需要发送的数据写入 TWDR 寄存器，清零 RXK 位，置位 TXD 并清零 TWIF。当数据发送完成并接收到正确的响应信息后，TWIF 位被重新置位，TXD 被清零，RXK 被置位。

接下来重复数据发送流程，直到最后一个字节的数据发送完成。然后发送一个停止位停止数据传输或者重复一个开始位，继续传输其他的数据。停止位通过写 TWSR 寄存器的 TXP 位，清零 RXK 位与 TWIF 位实现。重复开始条件可以通过置位 TXS 位，清零 RXK 以及 TWIF 位实现。停止位发送结束后，TXP 位被清零，但 TWIF 此时不会被置位。当重复开始发送结束后，TXS 被清零，同时 TWIF 标志位被置位。



主机发送模式数据流图

主机接收模式

首先，需要将 TWI 配置为 I2C 主机模式。

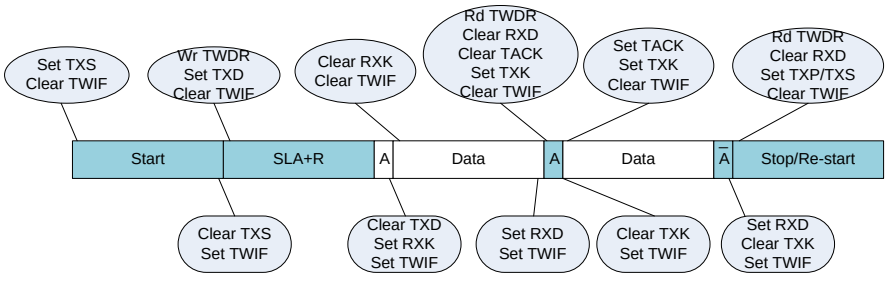
通过置位 TWSR 寄存器的 TXS 位，清零 TWIF 标志位，启动传输流程。TWI 控制器将会在总线空闲后发起一个开始位。开始位发送完成后，TWIF 标志位置位，同时 TXS 位清零。

进入 MR 模式前，需要先发送 SLA+R。将 SLA+R 写入 TWDR 寄存器，置位 TXD，清零 TWIF 将会启动 SLA+R 传输。在 SLA+R 传输完成并成功接收到响应信号后，TWIF 被再次置位，TXD 位被清零，同时 RXK 位被置位。

SLA+R 成功传输后，控制器可以进入数据接收状态。清零 RXK 以及 TWIF 标志位，控制器将会在 SCL 产生 8 个时钟周期脉冲，并接收完成一个自己的数据。数据接收完成后，TWSR 寄存器的 RXD 被置位。接收的数据可以通过读取 TWDR 寄存器获得。读完数据后需清零 RXD 标志位。数据接收后，需发送一个响应给从机。置位 TXK 标志位并清零 TWIF 标志位，控制器发送的是响应(ACK)还是非响应(NACK)取决于 TWCR 寄存器的 TACK 位。如果 TACK 为 0，主机将返回一个 ACK，否则返回一个 NACK。响应信号发送完成后，TXK 位被清零，同时 TWIF 位被再次置位。

数据接收过程将会重复直到接收到最后一个字节。数据接收的过程也可以被进一步简化。简化接收流程的操作通过预先设定响应位并置位 TXK 位实现。TXK 置位后，控制器在接收到一个字节的数据后，并不会触发 TWIF 标志位。待到响应位成功发送后，TWIF 置位。接收的数据可以从 TWDR 中读取。

传输由一个停止位或者一个重复开始位结束。通过置位 TXP 位并清零 RXK 位以及 TWIF 标志位，可以向总线发送一个停止位。重复开始条件可以通过置位 TXS 位，清零 RXK 位以及 TWIF 标志位实现。停止位发送后，TXP 位清零，TWIF 标志位将不会被置位。当重复开启位发送后，TXS 位被清零，同时 TWIF 位被置位。



Master Received Mode Data Flow Diagram

从机发送模式

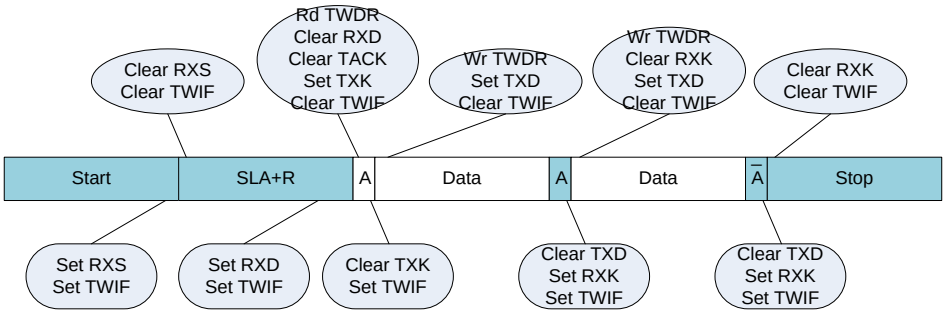
首先，需要设置 TWI 控制为 I2C 从机工作模式.

将控制器在总线上检测到一个传输开始位，TWSR 寄存器的 RXS 位被置位，同时 TWIF 位也被置位。这种状态说明总线此时已被其他的主机占用。软件需要清零 RXS 位以及 TWIF 位准备接收接下来主机发送的地址以及读写控制位。

当接收到 SLA+W/R 字节后，RXD 标志位被置位，同时 TWIF 位也被置位。软件响应 TWIF 中断，读取 SLA+W/R 数据并根据其中的地址信息判断是否被寻址。如果本机是访问目标，清零 TACK 标志位，否则置位 TACK 标志位。如果本机被寻址，软件需要更加 W/R 控制位确定接下来的操作。如果是主机发起的读操作，控制器将会进入到从机发送模式，软件可以通过清零 RXD，设置 TXK 标志位以及清零 TWIF 标志位发送响应信息。

响应位发送后，TXK 位被清零，TWIF 位被置位。此后，可以发送数据部分。将待发送的数据写入 TWDR 寄存器，置位 TXD 标志位并清零 TWIF 标志位，启动数据发送。当数据被成功发送并接收到来自主机的响应位后，TXD 位被清零，RXK 以及 TWIF 位再次被置位。

数据传输流程将会反复循环直到发送完成最后一个数据字节或接收到 NACK，接收到一个停止位，或者接收到一个重复开始条件。



从机发送模式数据流程图

从机接收模式

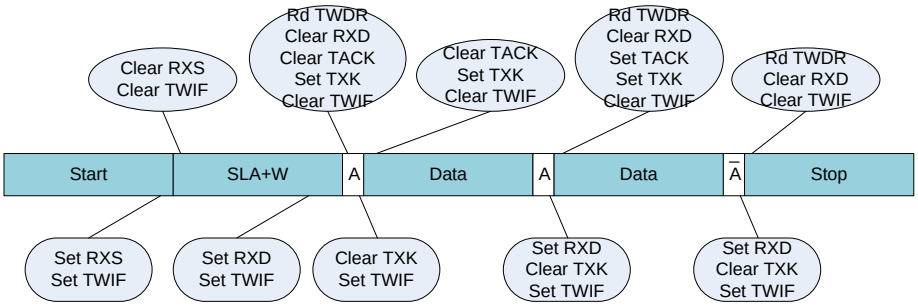
进入到从机接收模式前，首先需要将 TWI 控制配置为 I2C 从机模式.

当控制器在总线上检测到开始传输条件，RXS 位置位，同时 TWIF 也被置位。此时的状态说明总线已被其他主机占用。 软件需清零 RXS 以及 TWIF 位以接收从机地址控制数据。

SLA+W/R 字节接收完成后，RXD 位以及 TWIF 位同时被置位。软件通过 TWDR 寄存器获得接收的数据并根据接收的地址信息判断是否被寻址。如果被主机寻址，软件需要清零 TACK 位，否则置位 TACK。被寻址的控制器还需要根据读写控制位确定接下来需要进入的工作模式。如果是接收到一个写操作，控制器进入从机接收模式。控制器通过清零 RXD 位，置位 TXK 位以及清零 TWIF 位向主机发送响应信息。

响应信息发送完成后，TXK 被清零，TWIF 再次置位。此时可以接收数据。与主机接收模式相同，从机的数据接收流程也可以被简化。通过预定义响应位(TACK 位)，设置 TXK 位以及清零 TWIF 位，可使控制器工作于简化数据接收模式。控制器接收完一个字节的数据并发送完响应信息后，TXK 位清零，TWIF 位被置位。接收的数据被保存到 TWDR 寄存器。

同样，数据接收流程将会被循环直到接收完最后一个字节并完成 NACK 的发送。



从机接收模式流程图

寄存器定义

TWCR – TWI 控制寄存器

TWCR		地址:0x9E	初始值:0x08
Bit	Name	描述	
7	TWEN	TWI 使能控制位，高有效。	
6	TWMST	TWI 主机模式控制位： 1 = 主机模式 0 = 从机模式	
5:4	-	-	
3	TACK	发送总线响应控制位 0 = ACK 1 = NACK	
2	RACK	接收总线响应标志位 0 = ACK 1 = NACK	
1:0	CKPS	I2C 总线时钟分频控制： 00: F_{SCL} = Fsys/10 01: F_{SCL} = Fsys/20 10: F_{SCL} = Fsys/40 11: F_{SCL} = Fsys/64	

TWSR – TWI 状态存器

TWSR		地址:0x98	初始值:0x00
Bit	Name	描述	
7	TXP	软件写 1 发送停止位 硬件清零，表示已接收到停止位	
6	RXK	接收到总线响应。硬件位置，软件清零	
5	TXD	软件写 1 发送数据 硬件清零，表示数据发送完成	
4	TXS	软件写 1 发送开始位 硬件清零，表示开始位已发送完成	

3	RXP	接收到停止位。硬件置位，软件清零
2	TXK	软件写 1 发送响应位 硬件清零，表示停止位已发送完成
1	RXD	数据接收完成。硬件置位，软件清零
0	RXS	接收到开始位。硬件置位，软件清零

TWDR – TWI 数据接收/发送寄存器

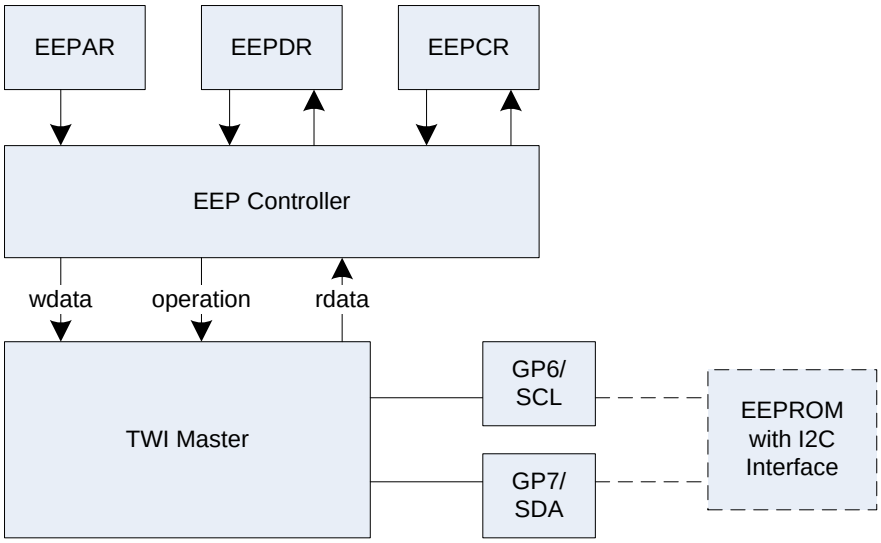
TWDR		地址:0x8D	初始值:0x00
Bit	Name	描述	
7:0	TWDR	TWI 接收/发送数据寄存器	

E2PROM 控制器 (EEP)

- 支持最大 256 字节 E2PROM
- 支持 I2C 接口 E2PROM
- 支持字节写，页写，字节读以及顺序读等模式

综述

EEP 控制器专为访问 I2C 接口的 E2PROM 设计。实现包括一个 8 位的地址寄存器和一个 8 位的数据寄存器。可用于访问内部(LGT8PE663A)或者外部 I2C 接口 E2PROM 存储设备。.



EEP 控制结构图

工作模式

EEP 主要支持两种工作模式：读操作、写操作.

写操作

首先，软件将目标地址写入 EEPAR 寄存器，将待写的数据写入 EEDPR 寄存器。置位 EEPCR 寄存器的 EEPWE 位，使能 EEP 控制器的写功能模块。然后，软件通过置位 EEPPE 位启动写操作流程。写操作完成后，EEP 控制器清零 EEPPE 位，并同时置位 EEPIF 中断标志位。

当需要连续写入多个字节，可以使用多字节写操作模式。通过置位 EEPCR 寄存器的 EEPMD 位，使能多字节操作模式。同样，通过置位 EEPPE 位开始写操作流程。当控制器完成一个字节的写操作后，将置位 EEPCR 寄存器的 EEPBR 位，软件可以据此更新 EEEPDR 为下一个字节数据。控制器将自动递增 EEPAR 寄存器。软件清零 EEPBR 位完成更新数据的写操作。当进入到最后一个字节，软件需要首先清零 EEPMD 位，进入单字节写入模式，更新 EEPDR，清零 EEPBR 完成最后一个字节的写操作。操作完成后，EEPE 位将会被硬件清零，同时 EEPIF 中断标志位被置位。

读操作

读 EPROM 数据前，软件需要将目标地址写入到 EEPAR 寄存器。然后置位 EEPCR 寄存器的 EEPRE 位开始一个读操作流程。读完成后，控制器将会清零 EEPRE 位。同时 EEPIF 中断标志位被置位。所读数据被更新到 EEPDR 寄存器中。

以上为单字节读操作。当需要连续读多个字节数据时，可以使用 EEP 控制器的连续读模式。通过使能 EEPCR 寄存器的 EEPMD 位，开启连续地址操作模式。同样，软件需要将需要访问连续数据的起始地址写入到 EEPAR 寄存器，并置位 EEPRE 位开始读流程。控制器读取到一个字节的数据后，置位 EEPBR 位。软件需据此读走 EEPDR 中的数据，然后清零 EEPBR 位继续下一个字节的读取。当来到最后一个字节的读取时，软件需要首先清零 EEPMD 位，切换到单字节工作模式，然后清零 EEPBR 位，完整最后一个字节的读取。最后一个字节完成后，硬件清零 EEPRE 位，并同时置位 EEPIF 中断标志位。

异常处理

如果在读写操作过程中出现错误，EEP 控制器将会置位 EEPCR 寄存器的 EEPER 位。并同时置位 EEPIF 中断标志位。读/写操作完成后，用户需要通过检查 EEPER 位判断操作是否成功。

中断请求

当控制器完成读/写操作，或者在操作过程中出现异常状况，EEP 控制器将会位置 EEPIF 中断标志位。此时如果使能了 EEP 中断使能位 EEPIE 以及外设中断使能位 PEIE，EEP 将会向内核发送一个中断请求。如果系统使能了全局中断控制位 GIE，内核将会在执行完当前指令后响应这个中断请求。用户需要在响应 EEPIF 中断后清除该标志位。

寄存器定义

EEPDR – EEP 数据寄存器

EEPDR		地址:0x9A	初始值:0x00
Bit	Name	描述	
7:0	EEPDR	EEP 读/写数据寄存器	

EEPAR – EEP 地址寄存器

EEPAR		地址:0x9B	初始值:0x00
Bit	Name	描述	
7:0	EEPAR	EEP 读/写目标地址寄存器	

EEPCR – EEP 控制寄存器

EEPCR		地址:0x9C	初始值:0x00
Bit	Name	描述	
7	EEPEN	EEP 模块使能控制位，高有效。	
6	EERST	EEP 模块复位寄存器，高有效。	

5	EEPMD	EEP 操作模式选择位： 1 = 连续地址操作模式 0 = 单字节操作模式
4	EEPBR	EEP 连续地址模式下的字节就绪状态位 1 = 连续模式下的单此操作完成 软件清零继续下一个字节的操作
3	EEPER	EEP 异常标志位。硬件置位，软件清零
2	EEPWE	EEP 写操作使能位 1 = 使能 EEP 的写操作功能 0 = 禁止 EEP 模块的写操作功能
1	EEPPE	EEP 写操作控制位，写 1 启动写操作流程
0	EEPRE	EEP 读操作控制位，写 1 启动读操作流程

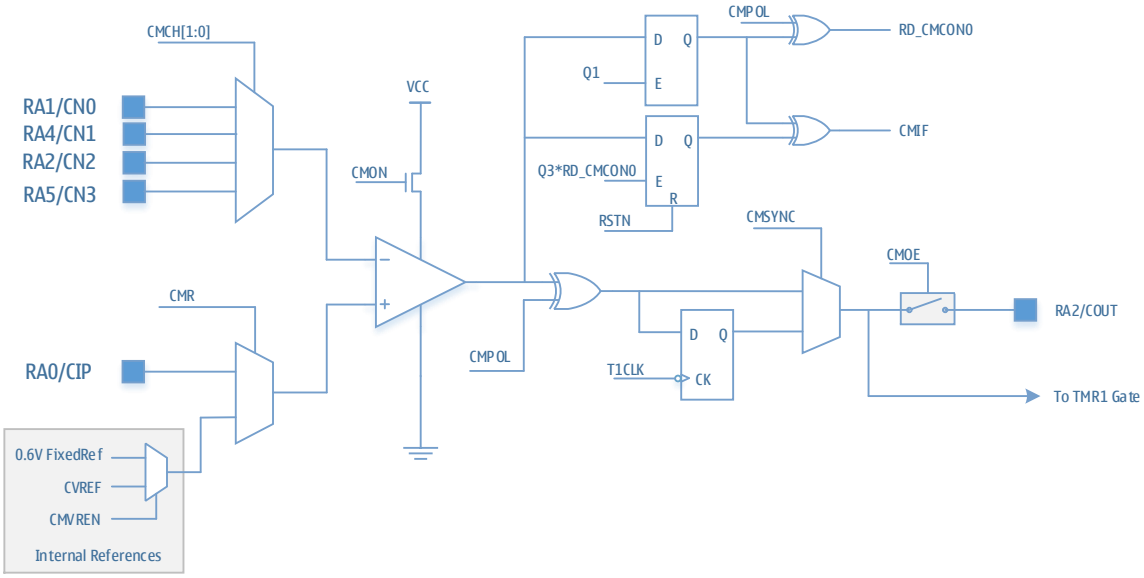
模拟比较器

- 可编程多路输入选择
- 比较器输出到端口(RA2)
- 可配置输出极性控制
- 比较器输出变化中断
- 支持休眠唤醒
- 可作为 TMR1 门控
- 内部数字滤波
- 可编程内部参考电压
- 输出迟滞控制

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震荡，为避免这种情况，一般比较器输出都有一个迟滞电路，通过使能比较器输出的迟滞电路，可以获得一个稳定的比较结果。

比较器模块结构图：



比较器控制

比较器由两个控制寄存器：CMCON0/CMCON1。CMCON1 用于控制与 TMR1 的互动操作以及读取比较器的输出。CMCOM0 寄存器包含如下控制和状态位：

- 比较器使能控制
- 输入选择
- 内部参考选择
- 输出选择
- 输出极性选择

将 CMCON0 寄存器的 CMON 位设置为 1，比较器正常工作。清零 CMON 位，比较器进入待机状态。当不需要比较器工作时，应该设置比较器为待机模式以节省功耗。

CMCON0 寄存器的 CMCH 位用于选择比较器负端输入。为了使用 ACIN0/RA1 或者 ACIN1P/RA4 作为比较器外部模拟输入，需要通过 ANSEL 寄存器将设置为 1，同时设置 TRISIO 寄存器对应位，关闭该段的输出驱动。

CMCON0 的 CMR 位选择比较器的正端输入。比较器的正端输入可选为来自外部 ACINP/RA0 或来自内部点参考电压。内部参考相关细节请参考本章节"参考电压"部分。

比较器的输出状态，可以通过 CMCON0 寄存器的 COUT 位获得。比较器的输出也可以直接连接到外部端口(RA2)。通过置位 CMCON0 寄存器的 CMOE 位，并清零 TRISIO[2]将 RA2 设置为输出端口，比较器的输出即可直接与 RA2 连接，RA2 将直接反应比较器的实时变化。

CMCON0 的 CMPOL 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	CMPOL	COUT
ACIN > ACIP	0	0
ACIN < ACIP	0	1
ACIN > ACIP	1	1
ACIN < ACIP	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。

内部参考电压

LGT8PE663A 内部提供了多种参考电压，这些参考电压都可通过 CMR 位配置为比较器的正端输入。比较器内部参考电压包括以下功能：

- 独立于比较器工作
- 32 级电压范围
- 输出 VSS 电平
- 内部固定参考(0.58V)

比较器参考电压独立于比较器，VRCON 寄存器的 VREN 位用于控制参考电压的使能与禁止。

可配置参考电压 (CVREF)

CVREF 参考电压支持两种可变模式，每种模式支持 16 级电压输出。电压模式由 VRCON 寄存器的 VRR 位控制。16 级电压选择通过 VRCON 寄存器的 VR[3:0]选择。

CVREF 输出电压可以通过以下公式计算：

VRR = 1 (low range)
CVREF = ((VR[3:0]+1)/24) x VDD
VRR = 0 (hign range)
CVREF = (VDD/4) + ((VR[3:0]+1)/32) x VDD

说明：因此参考电压的电路结构，内部参考无法实现 VSS 到 VDD 的全范围覆盖

CVREF 可以通过设置 VRCON 寄存器的 FVREN 为零将参考电压输出钳位到 VSS，此时比较器参考不会消耗多余的电量，这种模式可以在最低功耗条件下实现输入电压的过零检测。

固定参考电压 (0.58V ± 2%)

固定参考电压独立于 VDD 的变化。正常条件下，固定电压输出为 0.58V。固定参考电压通过设置 VRCON 寄存器的 FVRREN 为 1 使能。当使能内部低电压检测(LVR)模块，内部 0.58V 参考也会同时处于工作状态，为 LVR 电路提供一个内部参考。

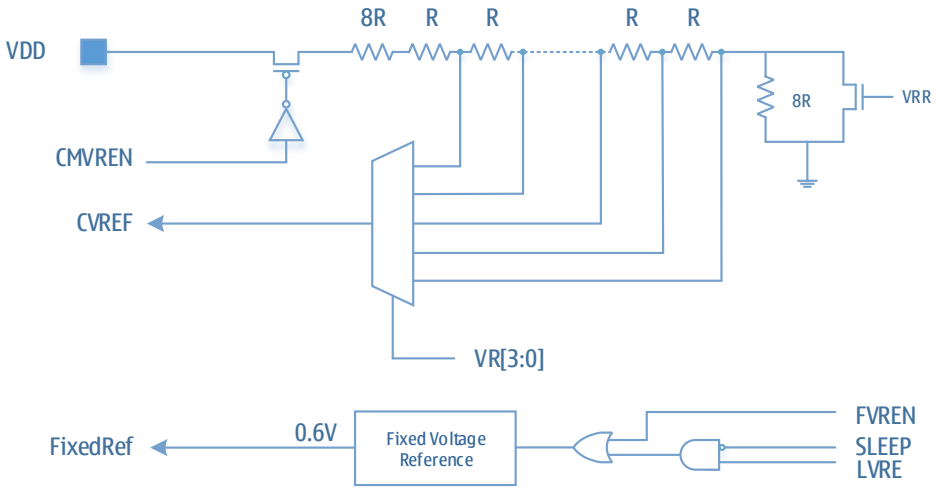
固定参考电压使能后，需要一个稳定时间。用户需要在使能固定参考电压后，增加一个软件延时。相关电气信息，请参考本手册电气特性部分。

参考电压选择

可变参考电压(VREF)与固定参考电压通过一个多路选择器，用户可以通过 VRCON 寄存器的相关控制位选择其中之一作为比较器正端的输入。

置位 VRCON 寄存器的 CMVREN 位，选择可变参考 CVREF 作为比较器输入；清零 CMVREN 位，将固定参考电压作为比较器输入。当 CMVREN 清零，可变参考 CVREF 的分压电路与 VDD 断开，分压电路不会消耗多余的电量。

内部参考电路结构



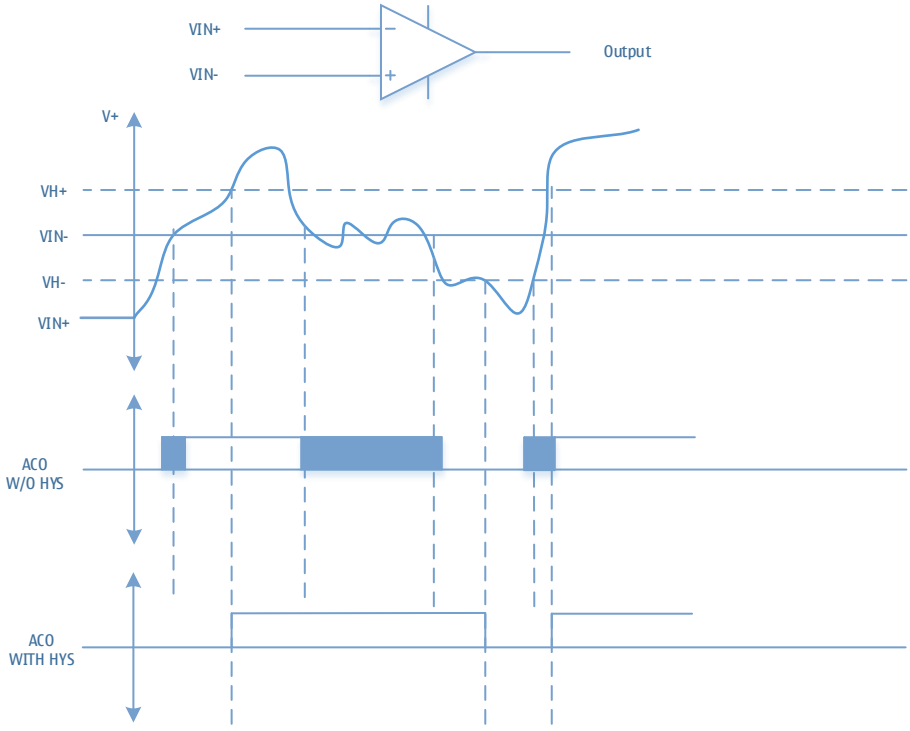
输出迟滞滤波

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。

如下图所示，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 COUT 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。

比较器迟滞电压与比较器输出关系图：



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								
8P53A 地址: N/C					默认值: 0000_0000			
8P609A 地址: 0x1A								
Bit	7	6	5	4	3	2	1	0
CMCON0	CMON	COUT	CMOE	CMPOL	-	CMR	CMCH1	CMCH0
R/W	R/W	R/W	R/W	R/W	-	W/R	W/R	W/R

Bit	Name	描述
7	CMON	比较器使能控制位，1：使能， 0：禁止
6	COUT	比较器输出位
5	CMOE	比较器输出使能 1：比较器输出到外部端口(RA2)
4	CMPOL	比较器输出极性选择位 1：COUT 输出反向
3	-	Unimplemented
2	CMR	比较器正端内部参考选择位 1：正端接 CMVREF 内部参考 0：正端接外部输入(RA0/CIP)
1:0	CMCH	比较器负端通道选择 00：负端接 CN0/RA1 01：负端接 CN1/RA4 10：负端接 CN2/RA2 11：负端接 CN3/RA5

CMCON1 – 比较器控制寄存器 1

CMCON1– 比较器控制寄存器 1								
8P53A 地址: N/C					默认值: XXX0_0X00			
8P609A 地址: 0x1C								
Bit	7	6	5	4	3	2	1	0
CMCON1	-	-	-	T1ACS	CMHYS	-	T1GSS	CMSYNC
R/W	-	-	-	R/W	W/	-	W/R	W/R

Bit	Name	描述
7	-	Unimplemented
6:5	CFEN	比较器输出滤波控制位 00 = 关闭输出数字滤波 01 = 32us 滤波宽度 10 = 64us 滤波宽度 11 = 96us 滤波宽度
4	T1ACS	TMR1 内部时钟源选择

		1：TMR1 时钟来自系统时钟(SYSCLK) 0：TMR1 时钟来自指令周期(SYSCLK/T)
3	CMHYS	比较器输出迟滞使能控制。1：使能，0：禁用
2	-	Unimplemented
1	T1GSS	TMR1 门控输入选择位 1：门控输入源来自外部端口(/T1G) 0：门控输入源来自比较器输出
0	CMSYNC	比较器输出同步控制位 1：比较器输出同步于 TMR1 时钟的下降沿 0：比较器输出异步模式

VRCON – 参考电压控制寄存器

VRCON– 参考电压控制寄存器								
8P53A 地址: N/C					默认值: 0X0_0000			
8P609A 地址: 0x19								
Bit	7	6	5	4	3	2	1	0
VRCON	CMVREN	-	VRR	FVREN	VR3	VR2	VR1	VR0
R/W	R/W	-	R/W	R/W	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	CMVREN	内部参考选择位 1：可变参考(CVREF)接比较器输入 0：0.58V 固定参考接比较器输入						
6	-	Unimplemented						
5	VRR	可变参考模式选择位 1：低压范围， CVREF = ((VR[3:0]+1)/24) x VDD 0：高压范围，CVREF = VDD/4 + ((VR[3:0]+1)/32) x VDD						
4	FVREN	0.58V 固定参考使能位 1：使能 0.58V 固定参考电压源 0：禁用 0.58V 固定参考电压源						
3:0	VR	可变参考电压设置 VRR = 1： CVREF = ((R[3:0]+1)/24) x VDD VRR = 0：CVREF = VDD/4 + ((VR[3:0]+1)/32) x VDD						

ANSEL – 端口模拟功能控制寄存器

ANSEL– 端口模拟功能控制寄存器								
8P53A 地址: N/C					默认值: 0000_0000			
8P609A 地址: 0x9F								
Bit	7	6	5	4	3	2	1	0
ANSEL	-	-	-	ANS4	ANS3	ANS2	ANS1	ANS0
R/W	-	-	-	R/W	W/R	W/R	W/R	W/R
Bit	Name	描述						

7:5	-	Unimplemented
4	ANS4	RA5 模拟功能控制，1：使能，0：禁用
3	ANS3	RA4 模拟功能控制，1：使能，0：禁用
2	ANS2	RA2 模拟功能控制，1：使能，0：禁用
1	ANS1	RA1 模拟功能控制，1：使能，0：禁用
0	ANS0	RA0 模拟功能控制，1：使能，0：禁用

系统配置位

LGT8PE663A 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 OTP 的编程信息，请参考相关资料。

配置字 1：系统时钟相关配置

Bit13 Bit0

Bit[13:11]	DPSM[1:0]	OSCO	RCM[2:0]	SUT[1:0]	FOSC[2:0]
位定义					
FOSC[2:0]	系统时钟模式配置字 000/111: RCM 模式，8/4/2/1MHz 或者 455KHz 内部 RC 模式 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				
SUT[1:0]	启动时间配置字 11: 264ms 10: 63ms 01: 16ms 00: 2ms				
RCM[2:0]	内部 RCM 时钟模式配置字，选择 RCM 时钟输出频率 111: 4MHz 101: 1MHz 100: 2MHz 110: 8MHz 0XX: 455KHz				
CLKOE	系统时钟输出控制配置字 1: RA4 输出系统时钟 0: RA4 不输出系统时钟				
DPSM[1:0]	休眠模式配置字 00: 待机模式 01: 休眠模式 1x: 掉电模式				

配置字 2：系统运行相关配置

Bit13 Bit0

Bit[13:11]	PINRM	LVRSM	LVDT[3:0]	WDTE	RSTE	TCYC[1:0]	MMODE

位定义	
MMODE	内核运行模式配置字 1: 8P609A 模式 ~~0: 8P53A 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
RSTE	外部复位功能配置字 1: 使能 RA3 的外部复位功能 0: 禁用 RA3 的外部复位功能
WDTE	看门狗定时器使能配置字 1: 使能看门狗定时器 0: 禁用看门狗定时器
LVDT[3:0]	LVR 复位阀值配置字 1111: 关闭 LVR 0000: 1.5V 0001: 1.7V 0010: 1.9V 0011: 2.1V 0100: 2.3V 0101: 2.5V 0110: 2.7V 0111: 2.9V 1000: 3.1V Others: Unimplemented
LVRSM	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	读端口状态模式配置字，主要影响 BSF/BCF 指令 1: 读数据来自端口寄存器 0: 读数据来自外部端口状态
Bit[13:11]	Unimplemented

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] = 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) ^ 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 ^ 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
BTFSC	BTFSC F, b	IF(!F[b]) PC = PC + 2	1	-	1/2
BTFSS	BTFSS 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

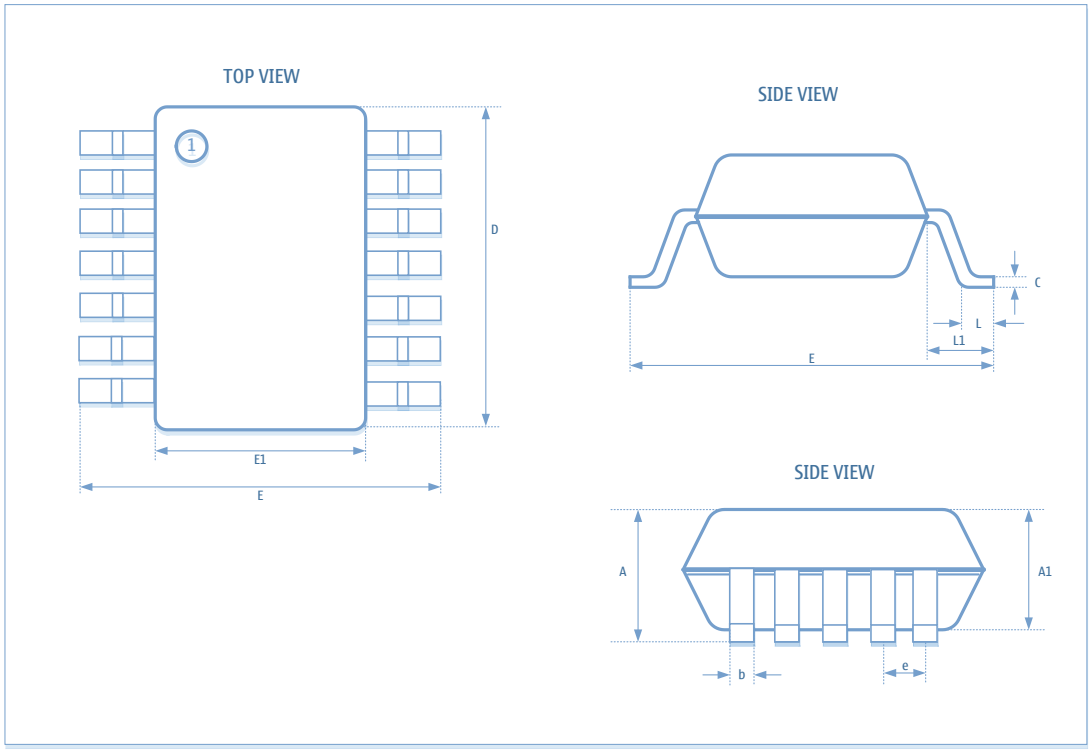
RETLW	RETLW k	W = k, PC <- STACK	1	-	2
基本数据传输指令					
MOVWF	MOVWF F	(F) = W	1	-	1
MOVF	MOVF F, d	[W/F] = (F)	1	Z	1
MOVLW	MOVLW k	W = k	1	-	1
其他辅助指令					
NOP	NOP	NO operation	1	-	1
OPTION	OPTION	OPTION = W	1	-	1
TRIS	TRIS F	IOSTA/B = W, (F=5/6)	1	-	1
SLEEP	SLEEP	Sleep mode	1	TO/PD	1
CLRWDT	CLRWDT	Clear 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

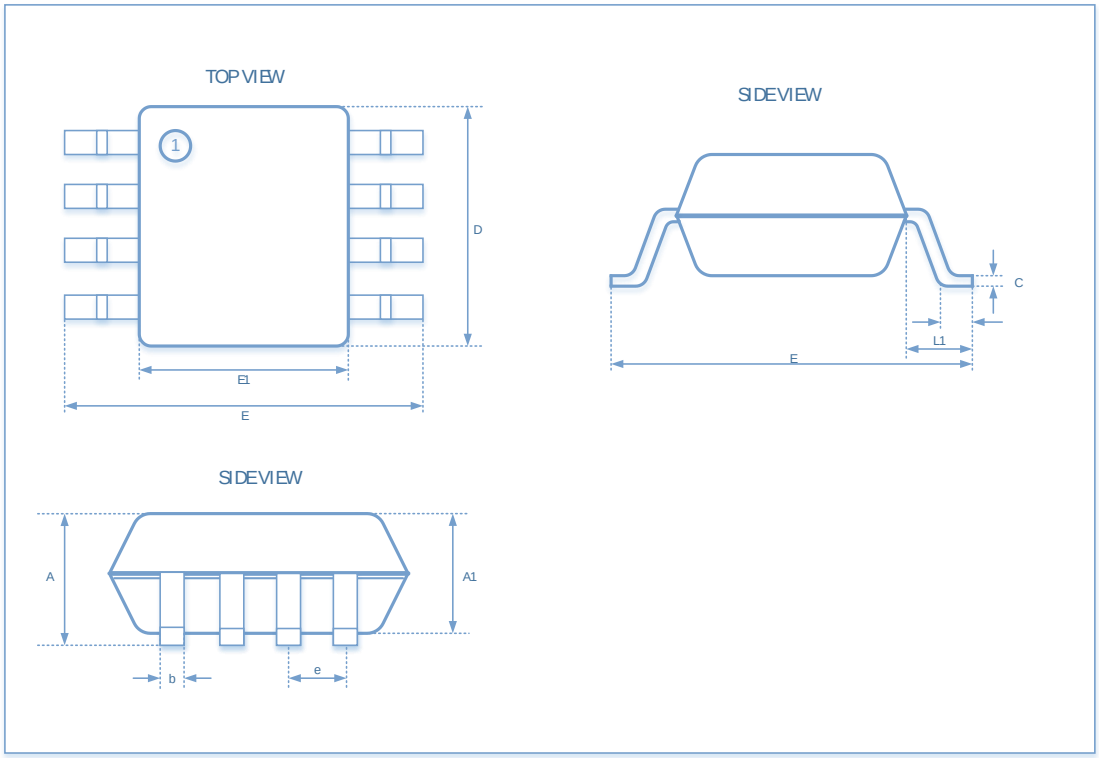
封装参数

SOP14 封装



Item	Min.	Typ.	Max.	Unit
A	1.35	-	1.75	mm
A1		1.25	-	mm
b	0.33	-	0.51	mm
e	-	1.27BSC	-	mm
E	5.80	-	6.20	mm
E1	3.80	-	4.00	mm
D	8.55	-	8.75	mm
C	0.19	-	0.25	mm
L	0.40	-	1.27	mm
L1	-	0.95BSC	-	mm

SOP8 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

版本历史

V1.0.4 2016/8/31	更新了对 OPTION 寄存器的位定义
V1.0.3 2016/8/12	改正了 SOP8 封装中对 RA6/7 复用的说明错误 增加了 SOP14 封装信息
V1.0.2 2016/03/20	更新内部参考电压为 0.58V
V1.0.1 2016/03/20	更新了部分寄存器描述
V1.0.0 2016/3/9	初始版本