

# AT8B62F1

---

**14 I/O + 12 通道 ADC 8 位 EPROM-Based**

**单片机**

**Version 1.2**

**Nov. 28, 2025**

---

ATW TECHNOLOGY CO. reserves the right to change this document without prior notice. Information provided by ATW is believed to be accurate and reliable. However, ATW makes no warranty for any errors which may appear in this document. Contact ATW to obtain the latest version of device specifications before placing your orders. No responsibility is assumed by ATW for any infringement of patent or other rights of third parties which may result from its use. In addition, ATW products are not authorized for use as critical components in life support devices/systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of ATW.

## 改版记录

版本	日期	内容描述	修正页
1.0	2025/05/22	新发布。	-
1.1	2025/09/04	<ol style="list-style-type: none"><li>增加 2 组外部中断 (INT0/INT1)。</li><li>增加 PA4 高电平保持 1M 规格。</li><li>更新 6.2 直流特性。</li></ol>	<p>10 8, 113 112</p>
1.2	2025/11/28	<ol style="list-style-type: none"><li>更新第 3.4.28 和 29 章节的 PWM5 设置。</li><li>更新比较器/ LVD 特性</li><li>更新 LVR 推荐电压。</li></ol>	<p>49, 50 114 119</p>

## 目 录

1. 概述	8
1.1 功能	8
1.2 AT8B62F 与 AT8B62F1 的主要差异	10
1.3 方框图	11
1.4 引脚图	12
1.5 引脚说明	13
2. 内存结构	15
2.1 程序存储器	15
2.2 数据存储器	16
3. 功能概述	19
3.1 R-page 特殊功能寄存器	19
3.1.1 INDF (间接寻址寄存器)	19
3.1.2 TMR0 (定时器 0 寄存器)	19
3.1.3 PCL (程序计数器低字节)	19
3.1.4 STATUS (状态寄存器)	20
3.1.5 FSR (数据指针寄存器)	20
3.1.6 PortA (PortA 数据寄存器)	21
3.1.7 PortB (PortB 数据寄存器)	21
3.1.8 PCON (Power 寄存器)	21
3.1.9 BWUCON (PortB 唤醒控制寄存器)	22
3.1.10 PCHBUF (程序计数器高字节)	22
3.1.11 ABPLCON (PortA/PortB 下拉电阻控制寄存器)	22
3.1.12 BPHCON (PortB 上拉电阻控制寄存器)	23
3.1.13 INTE (中断使能寄存器)	23
3.1.14 INTF (中断标志寄存器)	24
3.1.15 ADMD (ADC 模式寄存器)	25
3.1.16 ADR (ADC 时钟, ADC 中断标志位与 ADC 低四位输出寄存器)	26
3.1.17 ADD (ADC 输出数据寄存器)	26
3.1.18 ADVREFH (ADC 参考电压寄存器)	26
3.1.19 ADCR (ADC 采样时间与 ADC 位数寄存器)	27
3.1.20 AWUCON (PortA 唤醒控制寄存器)	27

3.1.21	PACON (ADC模拟引脚寄存器) .....	28
3.1.22	ADJMD (ADC调校模式) .....	28
3.1.23	INTEDG (外部中断控制寄存器) .....	28
3.1.24	TMRH (定时器高字节寄存器) .....	29
3.1.25	ANAEN (比较器使能寄存器) .....	29
3.1.26	RFC (电阻频率转换控制寄存器) .....	30
3.1.27	TM3RH (定时器3高字节寄存器) .....	30
3.1.28	INTE2 (第2个中断使能和标志寄存器) .....	31
3.2	T0MD (定时器0控制寄存器) .....	31
3.3	F-page 特殊功能寄存器 .....	32
3.3.1	IOSTA (PortA I/O控制寄存器) .....	32
3.3.2	IOSTB (PortB I/O控制寄存器) .....	32
3.3.3	APHCON (PortA上拉电阻控制寄存器) .....	33
3.3.4	PS0CV (预分频器0寄存器) .....	33
3.3.5	BODCON (PortB开漏控制寄存器) .....	33
3.3.6	CMPCR (比较器控制寄存器) .....	33
3.3.7	PCON1 (Power控制寄存器1) .....	34
3.4	S-page 特殊功能寄存器 .....	35
3.4.1	TMR1 (定时器1寄存器) .....	35
3.4.2	T1CR1 (定时器1控制寄存器1) .....	35
3.4.3	T1CR2 (定时器1控制寄存器2) .....	36
3.4.4	PWM1DUTY (PWM1占空比寄存器) .....	37
3.4.5	PS1CV (预分频器1寄存器) .....	37
3.4.6	BZ1CR (蜂鸣器1控制寄存器) .....	38
3.4.7	IRCR (IR控制寄存器) .....	38
3.4.8	TBHP (表格指针高字节寄存器) .....	39
3.4.9	TBHD (表格数据高字节寄存器) .....	39
3.4.10	TMR2 (定时器2寄存器) .....	40
3.4.11	T2CR1 (定时器2控制寄存器1) .....	40
3.4.12	T2CR2 (定时器2控制寄存器2) .....	41
3.4.13	PWM2DUTY (PWM2占空比寄存器) .....	42
3.4.14	PS2CV (预分频器2计数值寄存器) .....	42
3.4.15	BZ2CR (蜂鸣器2控制寄存器) .....	42
3.4.16	OSCCR (振荡器控制寄存器) .....	43
3.4.17	TMR3 (定时器3寄存器) .....	44
3.4.18	T3CR1 (定时器3控制寄存器1) .....	44

3.4.19 T3CR2 (定时器3控制寄存器2) .....	45
3.4.20 PWM3DUTY (PWM3 占空比寄存器) .....	46
3.4.21 PS3CV (预分频器3计数值寄存器) .....	46
3.4.22 BZ3CR (蜂鸣器3控制寄存器) .....	46
3.4.23 P4CR1 (PWM4 控制寄存器1) .....	47
3.4.24 PWM4DUTY (PWM4 占空比寄存器) .....	47
3.4.25 CCPCON (CCP控制寄存器) .....	48
3.4.26 PWMDB (死区控制寄存器) .....	49
3.4.27 P5CR1 (PWM5 控制寄存器1) .....	49
3.4.28 PWM5DUTY (PWM5 占空比寄存器) .....	49
3.4.29 PWM5RH (MSB[9:8]的PWM5 占空比寄存器) .....	50
3.5 I/O Port .....	50
3.5.1 IO引脚结构框图 .....	52
3.6 Timer0 .....	63
3.7 Timer1 / PWM1 / Buzzer1 .....	64
3.8 Timer2 / PWM2 / Buzzer2 .....	66
3.9 Timer3 / PWM3 / Buzzer3 / PWM4 / PWM5 .....	69
3.10 CCP模式 .....	72
3.10.1 捕捉模式 .....	73
3.10.2 比较模式 .....	74
3.10.3 CCP PWM 模式 .....	74
3.11 RFC模式 .....	76
3.12 IR (红外) 载波 .....	77
3.13 低电压检测 (LVD) .....	77
3.14 电压比较器 .....	78
3.15 模拟-数字转换器 (ADC) .....	81
3.15.1 ADC参考电压 .....	81
3.15.2 ADC模拟输入通道 .....	82
3.15.3 ADC时钟 (ADCLK)、采样时钟 (SHCLK) 和位数选择 .....	83
3.15.4 ADC操作过程 .....	84
3.16 看门狗定时器 (WDT) .....	84
3.17 中断 .....	84
3.17.1 Timer0 上溢中断 .....	85
3.17.2 Timer1 下溢中断 .....	85

3.17.3 Timer2 下溢中断或CCP中断	85
3.17.4 Timer3 下溢中断	85
3.17.5 WDT超时中断	85
3.17.6 PA/PB输入状态改变中断	85
3.17.7 外部中断0输入	85
3.17.8 外部中断1输入	86
3.17.9 低电压侦测中断	86
3.17.10 比较器输出翻转中断	86
3.17.11 模数转换完成中断	86
3.18 振荡器配置	86
3.19 工作模式	88
3.19.1 正常模式	90
3.19.2 慢速模式	90
3.19.3 待机模式	90
3.19.4 睡眠模式	91
3.19.5 唤醒稳定时间	91
3.19.6 工作模式概述	92
3.20 复位	92
4. 指令设置	94
5. 配置表	110
6. 电气特性	112
6.1 最大绝对值	112
6.2 直流电气特性	112
6.3 OSC电气特性	114
6.4 比较器/LVD电气特性	114
6.5 ADC电气特性	114
6.6 特性曲线图	115
6.6.1 高速RC振荡频率(I_HRC)及低速RC振荡频率(I_LRC)与电源电压(VDD)曲线图	115
6.6.2 高速RC振荡频率(I_HRC)与温度曲线图	115
6.6.3 低速RC振荡频率(I_LRC)与温度曲线图	116
6.6.4 内部参考电压LDO与温度曲线图	116
6.6.5 内部上拉电阻与电源电压(VDD)曲线图	117
6.6.6 VIH/VIL与电源电压(VDD)曲线图	117

---

6.7 建议工作电压.....	119
6.8 LVR电压与温度曲线图 .....	119
6.9 LVD电压与温度曲线图 .....	120
6.10 LDO与温度曲线图 .....	120
<b>7. 封装尺寸.....</b>	<b>121</b>
7.1 8 引脚SOP (150 mil).....	121
7.2 14 引脚SOP (150 mil).....	121
7.3 16 引脚SOP (150 mil).....	122
<b>8. 订购信息.....</b>	<b>122</b>

## 1. 概述

AT8B62F1 是以EPROM作为记忆体的 8 位微控制器，专为家电或量测等等的I/O应用而设计，采用CMOS制程并同时提供客户低成本、高性能及高性价比等显著优势。AT8B62F1 核心建立在RISC精简指令集架构可以很容易地做编辑和控制，共有 55 条指令。除了少数指令需要 2 个时序，大多数指令都是 1 个时序即能完成，可以让用户轻松地以程式控制完成不同的应用。因此非常适合各种中低记忆容量但又复杂的应用。

AT8B62F1 内建高精度十一加一通道十二位模-数转换器，与高精度电压比较器，足以应付各种模拟界面的侦测与量测。

在I/O的资源方面，AT8B62F1 有 14 根弹性的双向I/O脚，每个I/O脚都有单独的寄存器控制为输入或输出脚。而且每一个I/O脚位都有附加的程式控制功能如上拉或下拉电阻或开漏极(Open-Drain) 输出。此外针对红外线遥控的产品方面，AT8B62F1 内建了可选择频率的红外载波发射口。

AT8B62F1 有四组计时器，可用系统频率当作一般的计时的应用或者从外部讯号触发来计数。另外AT8B62F1 提供 5 组 10 位解析度的PWM驱动LED或蜂鸣器等等。

AT8B62F1 采用双时钟机制，高速振荡或者低速振荡都可以分别选择内部RC振荡或外部Crystal输入。在双时钟机制下，AT8B62F1 可选择多种工作模式如正常模式(Normal)、慢速模式(Slow mode)、待机模式(Standby mode) 与睡眠模式(Halt mode)可节省电力消耗延长电池寿命。并且微控制器在使用内部RC高速振荡时，低速振荡可以同时使用外部精准的Crystal计时。可以维持高速处理同时又能精准计算真实时间。

在省电的模式下如待机模式(Standby mode)与睡眠模式(Halt mode)中，有多种事件可以触发中断唤醒AT8B62F1 进入正常操作模式(Normal) 或 慢速模式(Slow mode) 来处理突发事件。

AT8B62F1 的PWM1/PWM2/PWM3/PWM4/PWM5 其计数器来源可不经除级。

### 1.1 功能

- 宽广的工作电压：
  - 3.0V ~ 5.5V @20MHz/2T
  - 1.6V ~ 5.5V @4MHz/4T
- 宽广的工作温度：-40°C ~ 85°C。
- 2Kx14 bits EPROM。
- 128 bytes SRAM。
- 14 根可分别单独控制输入输出方向的I/O脚(GPIO)、PA[7:0]、PB[5:0]。
- PA[5, 3:0] 及 PB[3:0] 可选择输入时使用内建下拉电阻。
- PA4 可选择High Level Hold 1M 高阻态功能。
- PA[7:0] 及 PB[5:0] 可选择输入时使用上拉电阻。
- PB[5:0] 可选择开漏极输出(Open-Drain)。
- PA[5] 可选择当作输入或开漏极输出(Open-Drain)。

- 所有I/O 脚输出可选择一般灌电流(Normal Sink Current)或大灌电流(Large Sink Current)，除PA5 外。
- 8 层程序堆栈(Stack)。
- 存取数据有直接或间接定址模式。
- 一组 8 位上数计时器(Timer0)包含可程式化的频率预除线路。
- 三组 10 位下数计时器(Timer1, 2, 3)可选重复载入或连续下数计时。
- 五个 10 位脉冲宽度调变(PWM1, 2, 3, 4, 5)。
- 三个蜂鸣器输出(BZ1, 2, 3)。
- 38/57KHz红外线载波频率可供选择，同时载波的级性也可以根据数据作选择。
- 内建准确的低电压侦测电路(LVD)。
- 内建十一加一通道 12 位模数转换器(Analog to Digital Converter)。
- 内建准确的电压比较器(Voltage Comparator)。
- 内建上电复位电路(POR)。
- 内建低压复位功能(LVR)。
- 内建看门狗计时(WDT)，可由程序控制开关。
- 内建电阻频率转换器(RFC)功能。
- 双时钟机制，系统可以随时切换高速振荡或者低速振荡。
  - 高速振荡: E\_HXT (超过 6MHz外部高速石英振荡)  
E\_XT (455K~6MHz外部石英振荡)  
I\_HRC (1~20MHz内部高速RC振荡)
  - 低速振荡: E\_LXT (32KHz外部低速石英振荡)  
I\_LRC (内部 32KHz低速RC振荡)
- 四种工作模式可随系统需求调整电流消耗：正常模式(Normal)、慢速模式(Slow mode)、待机模式(Standby mode) 与 睡眠模式(Halt mode)。
- 十二种硬件中断：
  - Timer0 上溢中断。
  - Timer1 下溢中断。
  - Timer2 下溢中断或CCP中断。
  - Timer3 下溢中断。
  - WDT超时中断。
  - PA/PB 输入状态改变中断。
  - 两组外部中断(INT0/INT1)。
  - 低电压侦测中断。
  - 比较器输出翻转中断。
  - 模-数转换完成中断。

- AT8B62F1 在待机模式(Standby mode)下的十二种唤醒中断:

- Timer0 上溢中断。
- Timer1 下溢中断。
- Timer2 下溢中断或CCP中断。
- Timer3 下溢中断。
- WDT超时中断。
- PA/PB输入状态改变中断。
- 两组外部中断(INT0/INT1)。
- 低电压侦测中断。
- 比较器输出翻转中断。
- 模-数转换完成中断。

- AT8B62F1 在睡眠模式(Halt mode)下的四种唤醒中断:

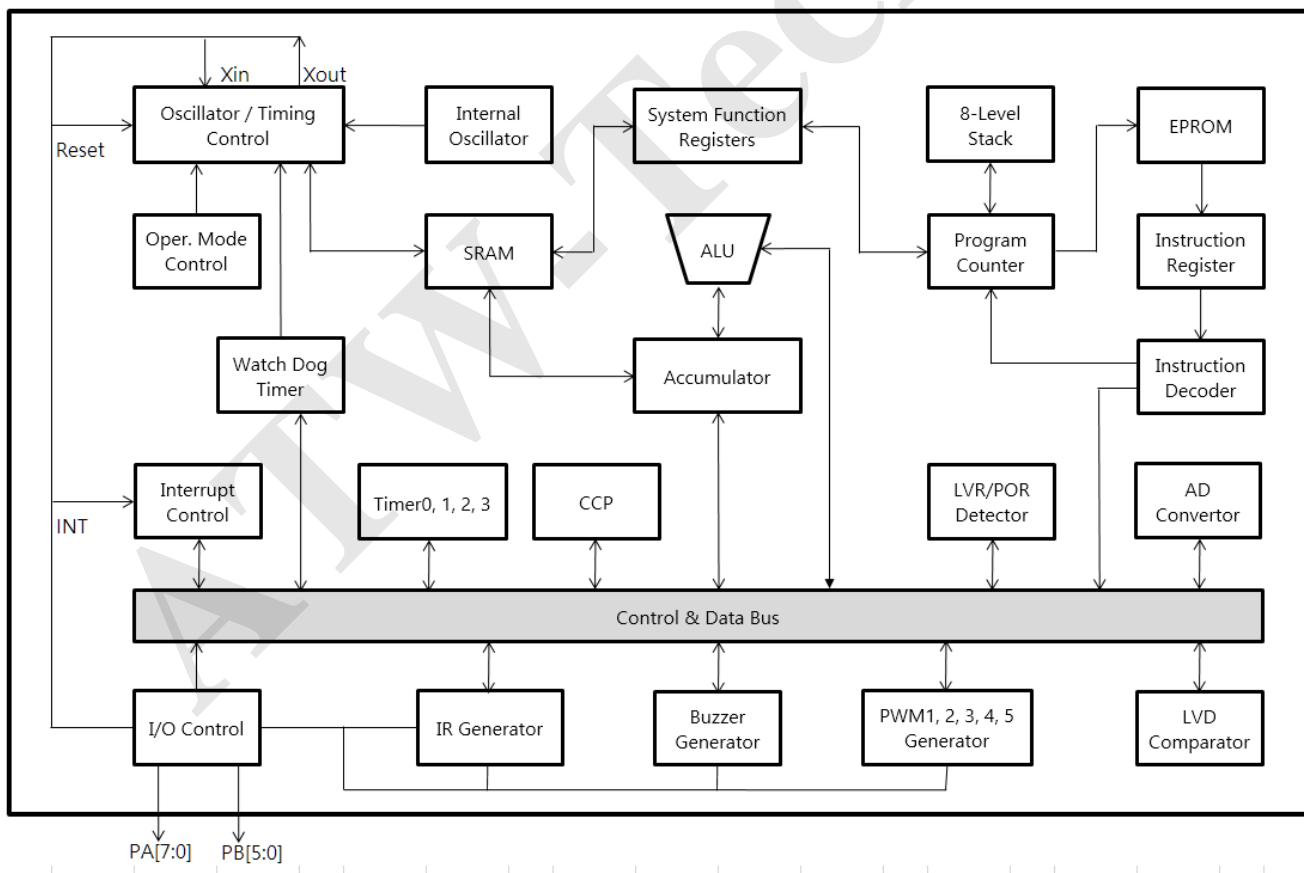
- WDT超时中断。
- PA/PB输入状态改变中断。
- 两组外部中断(INT0/INT1)。

## 1.2 AT8B62F 与 AT8B62F1 的主要差异

项目	功能	AT8B62F	AT8B62F1
1	ADC 补偿校准	Yes	Yes
2	ADC 解析度 DNL: +/-1; INL: +/- 1 Lsb	V	V
3	CCP	X	V
4	PWM 通道	1, 2, 3, 4	1, 2, 3, 4, 5
5	死区	X	V
6	PA4 保持 1M 上拉电阻	X	V
7	PWM 不经除级 I_HRC	X	V
8	PA5 输出高	X	V

(有关 ADC 零点校准流程, 可参考 NYIDE 范例程序)

## 1.3 方框图



## 1.4 引脚图

AT8B62F1 提供 3 种封装类型：SOP16, SOP14 和SOP8。

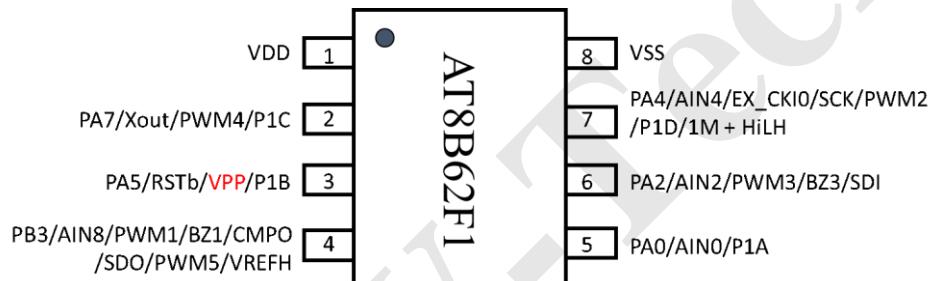
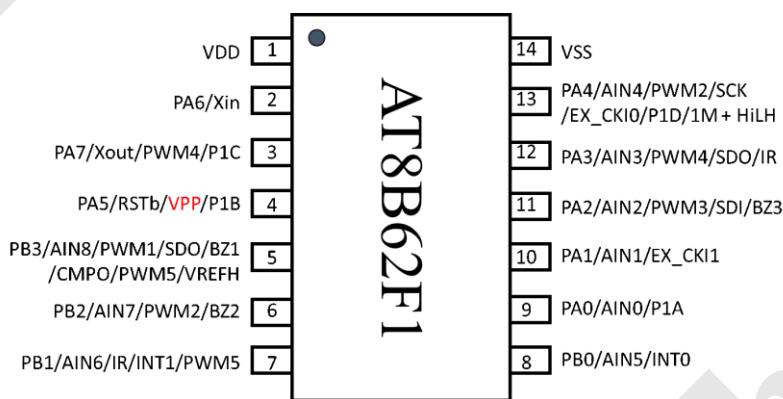
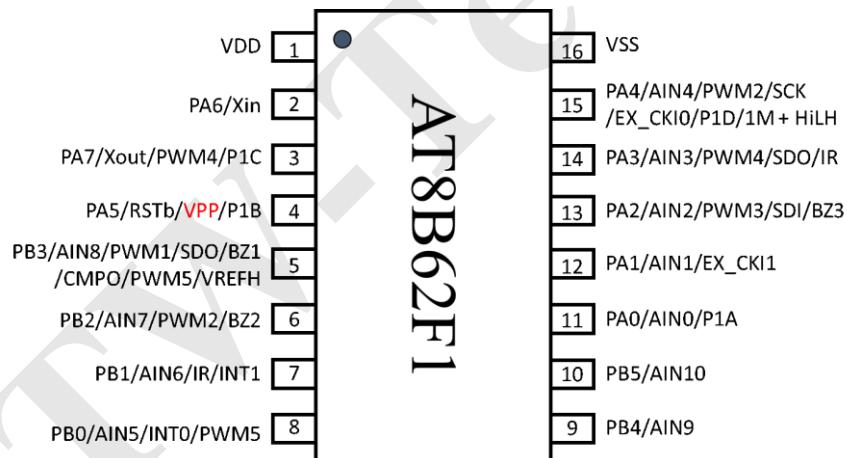


图 1 封装引脚图

## 1.5 引脚说明

引脚名	I/O	描述
PA0 AIN0 P1A	I/O	PA0 是一个双向I/O引脚，也可当作比较器输入引脚。 PA0 可作ADC的模拟输入引脚AN0。 PA0 可作CCP模式的P1A输出。
PA1 AIN1 EX_CK11	I/O	PA1 是一个双向I/O引脚，也可当作比较器输入引脚。 PA1 可作ADC的模拟输入引脚AN1。 PA1 可作定时器 2/3 外部时钟来源EX_CK11。
PA2 AIN2 PWM3/BZ3 SDI	I/O	PA2 是一个双向I/O引脚，也可当作比较器输入引脚。 PA2 可作ADC的模拟输入引脚AN2。 PA2 可输出PWM3 或BZ3。 PA2 也是编程数据输入SDI。
PA3 AIN3 PWM4 IR SDO	I/O	PA3 是一个双向I/O引脚，也可当作比较器输入引脚。 PA3 可作ADC的模拟输入引脚AN3。 PA3 可输出PWM4。 如果开启IR模式，此脚可作IR载波输出。 PA3 也是编程数据输出SDO。
PA4 AIN4 PWM2 EX_CK10 SCK P1D	I/O	PA4 是一个双向I/O引脚，并且有 $1\text{M}\Omega$ 上拉电阻。 PA4 可作ADC的模拟输入引脚AN4。 PA4 可输出PWM2。 PA4 可作定时器 0 / 1 外部时钟来源EX_CK10。 PA4 也是编程时钟输入SCK。 PA4 可作CCP模式的P1D输出。
PA5 RSTb Vpp P1B	I/O	PA5 是一个输入引脚或开漏极输出引脚。 PA5 可当作复位引脚RSTb。 如果Vpp电压高于 7.75V，IC会进入EPROM编程模式。 PA5 可作CCP模式的P1B输出。
PA6 Xin	I/O	PA6 是一个双向I/O引脚。 PA6 可当作晶振输入引脚Xin。
PA7 PWM4 Xout P1C	I/O	PA7 是一个双向I/O引脚。 PA7 可输出PWM4。 PA7 可当作晶振输出引脚Xout。 PA7 可作CCP模式的P1C输出。
PB0 AIN5 INT0 PWM5	I/O	PB0 是一个双向I/O引脚。 PB0 可作ADC的模拟输入引脚AN5。 PB0 可当作外部中断 0 的输入引脚INT0。 PB0 可输出PWM5。
PB1 AIN6 IR INT1	I/O	PB1 是一个双向I/O引脚。 PB1 可作ADC的模拟输入引脚AN6。 如果开启IR模式，此脚可作IR载波输出。 PB1 可当作外部中断 1 的输入引脚INT1。
PB2 AIN7 PWM2/BZ2	I/O	PB2 是一个双向I/O引脚。 PB2 可作ADC的模拟输入引脚AN7。 PB2 可输出PWM2 或BZ2。

引脚名	I/O	描述
PB3 AIN8 PWM1/BZ1 CMPO SDO PWM5 VREFH	I/O	PB3 是一个双向I/O引脚。 PB3 可作ADC的模拟输入引脚AN8。 PB3 可输出PWM1 或BZ1。 PB3 可作比较器输出脚。 PB3 也是编程数据输出SDO。 PB3 可输出PWM5。 PB3 可作ADC外部参考电压输入引脚VREFH。
PB4 AIN9	I/O	PB4 是一个双向I/O引脚。 PB4 可作ADC的模拟输入引脚AN9。
PB5 AIN10	I/O	PB5 是一个双向I/O引脚。 PB5 可作ADC的模拟输入引脚AN10。
VDD	-	电源正端。
VSS	-	电源负端。

## 2. 内存结构

AT8B62F1 存储器分为两类：分别是程序存储器和数据存储器。

### 2.1 程序存储器

AT8B62F1 程序存储器空间是  $2K \times 14$  位。因此，11 位宽的程序计数器 (PC) 可以访问程序存储器的任何地址。

程序存储器的一些位置保留给中断入口。上电复位地址位于 0x000，软件中断地址位于 0x001，内部和外部硬件中断地址位于 0x008。

AT8B62F1 提供 GOTOA 和 CALLA 等指令去访问程序空间的 256 个地址。还提供 LCALL 和 LGOTO 指令访问程序空间的任何地址。

当发生子程序调用或中断情况时，下一个 ROM 地址写入堆栈的顶部。而当执行 RET、RETIA 或 RETIE 指令，堆栈顶部的数据会被读取并加载到程序计数器。

AT8B62F1 程序存储器地址 0x7FE~0x7FF 是保留地址。如果用户在这些地址写入程序可能会发生无法预期的程序执行错误。

AT8B62F1 程序存储器地址 0x00E~0x00F 是 Preset Rolling Code 地址。如果用户在不设置滚码时可当作程序区使用。

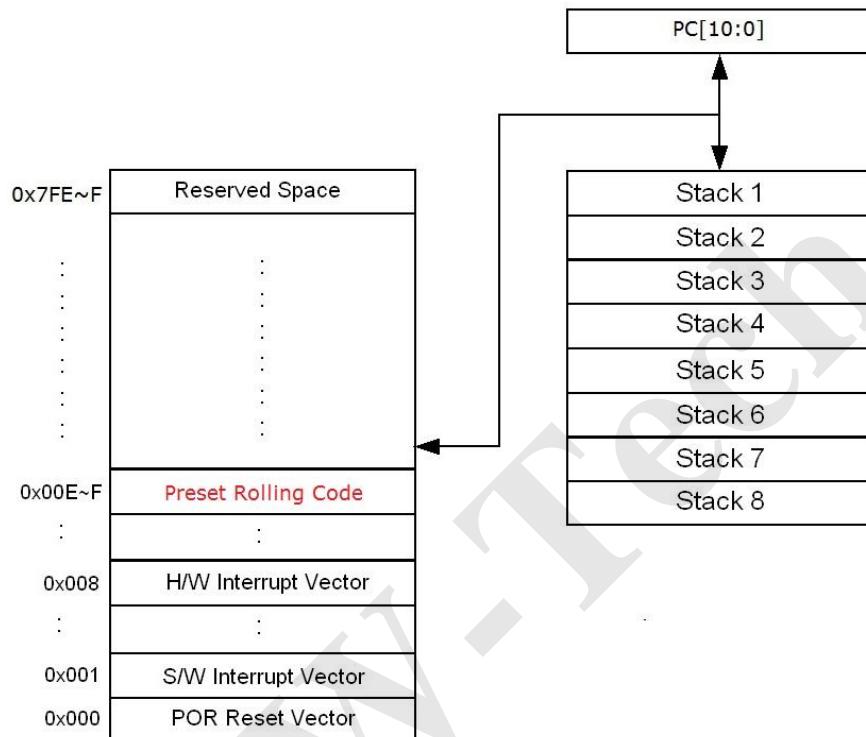


图 2 程序存储器对应地址

## 2.2 数据存储器

根据用于存取数据存储器的指令，数据存储器可分为三类：R-page特殊功能寄存器（SFR）和通用寄存器（GPR）、F-page特殊功能寄存器、S-page特殊功能寄存器。GPR是由SRAM组成，用户可以使用它们来存储变量或计算结果。

R-page特殊功能寄存器和数据存储器分为四组Bank，可通过数据指针寄存器（FSR）来切换Bank。寄存器BK[1:0]为STATUS[7:6]，可从四个Bank中选择其中一个。

R-page特殊功能寄存器和数据存储器可用直接寻址方式和间接寻址方式进行存取。

数据存储器使用间接寻址方式如下图所描述，这种间接寻址方式包含使用INDF寄存器。Bank选择是由STATUS[7:6]决定，地址选择则是由FSR[6:0]而定。

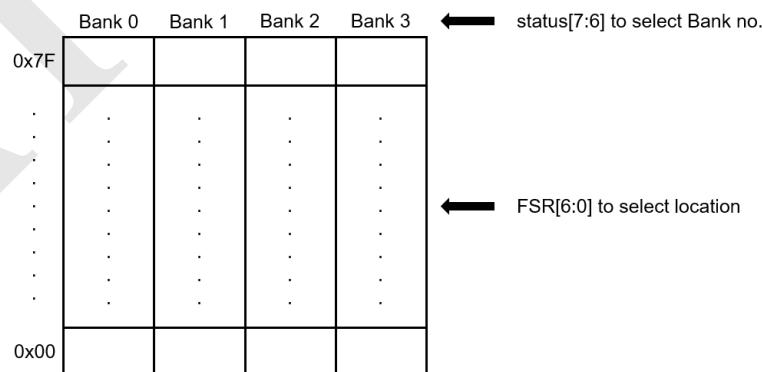


图3 间接寻址方式存取数据存储器

下面描述了数据存储器使用的直接寻址方式。Bank选择是由寄存器STATUS[7:6]决定，而地址选择则是由指令码OP-Code[6:0]直接决定。

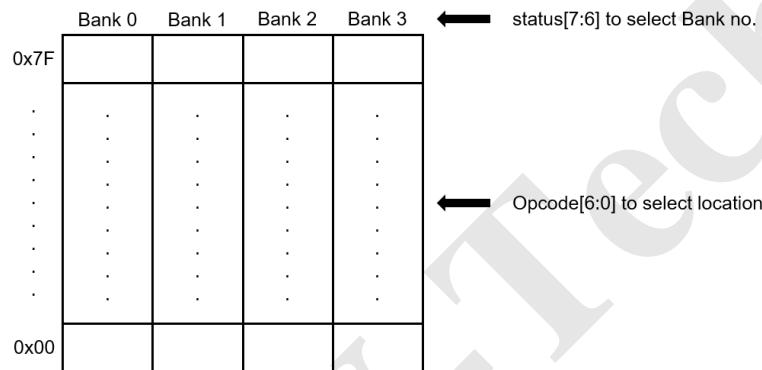


图4 直接寻址方式存取数据存储器

R-page特殊功能寄存器可以通过一般的指令存取，如算术指令和数据搬移指令。R-page特殊功能寄存器占用了从Bank 0 的 0x0 到 0x1F。然而，Bank 1、Bank 2 和Bank 3 的相同地址会映射到Bank 0。换句话说，R-page特殊功能寄存器只存在于Bank 0。GPR占用了Bank 0 数据存储器的 0x20 到 0x7F与Bank 1 数据存储器的 0x20 到 0x3F，另一个bank从 0x20 到 0x7F映射如表 1 所示。

AT8B62F1 寄存器名称和R-page特殊功能寄存器的映射地址说明如下表。

地址	Status [7:6]	00 (Bank 0)	01 (Bank 1)	10 (Bank 2)	11 (Bank 3)
0x0		INDF			
0x1		TMR0			
0x2		PCL			
0x3		STATUS			
0x4		FSR			
0x5		PORTA			
0x6		PORTB			
0x7		-			
0x8		PCON			
0x9		BWUCON			
0xA		PCHBUF			
0xB		ABPLCON			
0xC		BPHCON			
0xD		-			映射至Bank 0
0xE		INTE			
0xF		INTF			
0x10		ADMD			
0x11		ADR			
0x12		ADD			
0x13		ADVREFH			
0x14		ADCR			
0x15		AWUCON			
0x16		PACON			
0x17		ADJMD			
0x18		INTEDG			
0x19		TMRH			
0x1A		ANAEN			
0x1B		RFC			映射至Bank 0
0x1C		TM3RH			
0x1D ~ 0x1E		-			-
0x1F		INTE2			映射至Bank 0
0x20 ~ 0x3F		通用寄存器	通用寄存器	映射至Bank 0	映射至Bank 1
0x40 ~ 0x7F		通用寄存器	映射至Bank 0	映射至Bank 0	映射至Bank 0

表 1 R-page特殊功能寄存器地址映射表

F-page特殊功能寄存器只能被指令IOST和IOSTR存取，S-page特殊功能寄存器只能被指令SFUN和SFUNR存取。当F-page和S-page寄存器被存取时，STATUS[7:6]选择位会被忽略。寄存器名称和F-page、S-page的地址说明如下表。

SFR Category 地址	F-page SFR	S-page SFR
0x0	-	TMR1
0x1	-	T1CR1
0x2	-	T1CR2
0x3	-	PWM1DUTY
0x4	-	PS1CV
0x5	IOSTA	BZ1CR
0x6	IOSTB	IRCR
0x7	-	TBHP
0x8	-	TBHD
0x9	APHCON	TMR2
0xA	PS0CV	T2CR1
0xB	-	T2CR2
0xC	BODCON	PWM2DUTY
0xD	-	PS2CV
0xE	CMPCR	BZ2CR
0xF	PCON1	OSCCR
0X10	-	TMR3
0X11	-	T3CR1
0X12	-	T3CR2
0X13	-	PWM3DUTY
0X14	-	PS3CV
0X15	-	BZ3CR
0X16	-	P4CR1
0X17	-	-
0X18	-	PWM4DUTY
0x19		CCPCON
0X1A		PWMDB
0X1B		P5CR1
0X1D		PWM5DUTY
0X1F		PWM5RH

表 2 F-page特殊功能寄存器和S-page特殊功能寄存器地址表

### 3. 功能概述

本章节将详细描述AT8B62F1 的操作方式。

#### 3.1 R-page 特殊功能寄存器

##### 3.1.1 INDF (间接寻址寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INDF	R	0x0								INDF[7:0]
			读/写属性							读/写
			初始值							xxxxxxxx

间接寻址寄存器并不是真的存在，而是以间接寻址模式来使用。任何指令访问间接寻址寄存器时，实际上是访问数据指针寄存器FSR所选择的寄存器。

##### 3.1.2 TMR0 (定时器 0 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR0	R	0x1								TMR0[7:0]
			读/写属性							读/写
			初始值							xxxxxxxx

当读取TMR0 寄存器时，会得到定时器 0 目前计数数值。

当写入TMR0 寄存器时，会更新定时器 0 目前计数数值。

藉由设置T0MD与配置字 (Configuration Word)，定时器 0 时钟源可以从指令时钟 $F_{INST}$ 、外部时钟 $EX\_CKIO$ 或低频振荡器 $I\_LRC/E\_LXT$ 中择一。

##### 3.1.3 PCL (程序计数器低字节)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCL	R	0x2								PCL[7:0]
			读/写属性							读/写
			初始值							0x00

程序计数器 (PC) 是一个 11 位寄存器，分高 3 位和低 8 位。当程序执行了一个指令，同时PC数值会增加，除了某些指令会直接更改PC数值。PCL寄存器可存取PC低字节 (PC[7:0])，PC高字节 (PC[10:8]) 并不能直接存取，必须藉由PCHBUF寄存器完成存取。

LGOTO指令的PC[10:0]是从指令码取得。

LCALL指令的PC[10:0]是从指令码取得，下一个PC地址 (PC+1)，将被存到堆栈的顶部。

### 3.1.4 STATUS (状态寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	R	0x3	BK[1]	BK[0]	GP5	/TO	/PD	Z	DC	C
读/写属性	读/写	读/写	读/写	读/写	读/写 <sup>*2</sup>	读/写 <sup>*1</sup>	读/写	读/写	读/写	读/写
初始值	0	0	0	1	1	X	X	X	X	X

寄存器STATUS包含算术指令的结果和引起复位的原因。

**Bit0 C:** 进位/借位标志位。

C=1 时, 加法运算有进位或减法运算无借位。

C=0 时, 加法运算无进位或减法运算有借位。

**Bit1 DC:** 半进位/半借位标志位。

DC=1 时, 加法运算低四位有进位或减法运算时没有向高四位借位。

DC=0 时, 加法运算低四位无进位或减法运算时有向高四位借位。

**Bit2 Z:** 零位。

Z=1 时, 算术或逻辑运算的结果是零。

Z=0 时, 算术或逻辑运算的结果不为零。

**Bit3 /PD:** 睡眠模式标志位。

/PD=1 时, 上电或执行CLRWDT指令后。

/PD=0 时, 执行SLEEP指令后。

**Bit4 /TO:** 看门狗超时标志位。

/TO=1 时, 上电或执行CLRWDT或SLEEP指令后。

/TO=0 时, 发生WDT超时。

**Bit5 GP5:** 通用寄存器读/写位。

**Bit[7:6] BK[1:0]:** Bank选择位, BK[1:0]=00b选择Bank0, BK[1:0]=01b选择Bank1, BK[1:0]=10b选择Bank2。

BK[1:0]=11b, 选择Bank3。

(\*1) 可以被SLEEP指令清除。

(\*2) 可以由CLRWDT指令设定

### 3.1.5 FSR (数据指针寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FSR	R	0x4	GP7	FSR[6:0]						
读/写属性			读/写							
初始值			0	X	X	X	X	X	X	X

**Bit7[6:0] ]FSR[6:0]:** 从指定Bank数据存储器的 128 个寄存器中选择一个。

**Bit7 GP7:** 通用寄存器数据位。

### 3.1.6 PortA (PortA 数据寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PortA	R	0x5	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
读/写属性			读/写							
初始值			数据锁存器值是 xxxxxxxx, 读取值则是 xxxxxxxx, 端口值(PA7~PA0)							

读取PortA时, 若特定脚位被配置为输入脚, 将得到该脚位输入状态。然而, 若该脚位被配置为输出脚, 依据配置字选项RD\_OPT, 得到该脚位的状态或相对应的输出数据锁存值。当写入PortA时, 数据是被写入PortA的输出数据锁存器中。

### 3.1.7 PortB (PortB 数据寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PortB	R	0x6	GP7	GP6	PB5	PB4	PB3	PB2	PB1	PB0
读/写属性			读/写							
初始值			数据锁存器值是 xxxxxxxx, 读取值则是 xxxxxxxx, 端口值(PB5~PB0)							

读取PortB时, 若特定脚位被配置为输入脚, 将得到该脚位输入状态。然而, 若该脚位被配置为输出脚, 依据配置字选项RD\_OPT, 得到该脚位的状态或相对应的输出数据锁存值。当写入PortB时, 数据是被写入PortB的输出数据锁存器中。

Bit[7:6] **GP7~6**: 通用寄存器数据位。

### 3.1.8 PCON (Power 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	R	0x8	WDTEN	GP6	LVDEN	/PHPA5	LVREN	GP2	GP1	GP0
读/写属性			读/写							
初始值			1	0	0	1	1	0	0	0

Bit[2:0] **GP2, GP1, GP0**: 通用寄存器数据位。

Bit3 **LVREN**: 开启/关闭 LVR。

LVREN=1 时, 开启LVR。

LVREN=0 时, 关闭LVR。

Bit4 **/PHPA5**: 关闭/开启PA5 上拉电阻。

/PHPA5=1 时, 关闭PA5 上拉电阻。

/PHPA5=0 时, 开启 PA5 上拉电阻。

Bit5 **LVDEN**: 开启/关闭 LVD。

LVDEN=1 时, 开启LVD。

LVDEN=0 时, 关闭LVD。

Bit6 **GP6**: 通用寄存器数据位。

Bit7 **WDTEN**: 开启/关闭 WDT。

WDTEN=1 时, 开启WDT。

WDTEN=0 时, 关闭WDT。

### 3.1.9 BWUCON (PortB 唤醒控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BWUCON	R	0x9	-	-	WUPB5	WUPB4	WUPB3	WUPB2	WUPB1	WUPB0
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

Bit[5:0] **WUPBx**: 开启/关闭PBx唤醒功能,  $0 \leq x \leq 5$ 。

WUPBx=1 时, 开启PBx唤醒功能。

WUPBx=0 时, 关闭PBx唤醒功能。

### 3.1.10 PCHBUF (程序计数器高字节)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCHBUF	R	0xA	-	XSPD_STP	-	-	-	-	-	PCHBUF[2:0]
读/写属性			-	写	-	-	-	-	-	读/写
初始值			X	0	X	X	X	X	X	000

Bit[2:0] **PCHBUF[2:0]**: 程序计数器PC的第十个位到第八个位。

Bit6 **XSPD\_STP**: 写 1 停止外部晶振 32.768KHz起振强化功能。

### 3.1.11 ABPLCON (PortA/PortB 下拉电阻控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ABPLCON	R	0xB	/PLPB3	/PLPB2	/PLPB1	/PLPB0	/PLPA3	/PLPA2	/PLPA1	/PLPA0
读/写属性	读/写									
初始值			1	1	1	1	1	1	1	1

Bit[3:0] **/PLPAx**: 关闭/开启PAx下拉电阻,  $0 \leq x \leq 3$ 。

/PLPAx=1, 关闭PAx下拉电阻。

/PLPAx=0, 开启PAx下拉电阻。

Bit[7:4] **/PLPBx**: 关闭/开启PBx下拉电阻,  $0 \leq x \leq 3$ 。

/PLPBx=1, 关闭PBx下拉电阻。

/PLPBx=0, 开启PBx下拉电阻。

### 3.1.12 BPHCON (PortB 上拉电阻控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BPHCON	R	0xC	-	-	/PHPB5	/PHPB4	/PHPB3	/PHPB2	/PHPB1	/PHPB0
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

Bit[5:0] **/PHPBx**: 关闭/开启PBx上拉电阻,  $0 \leq x \leq 5$ 。

/PHPBx=1 时, 关闭PBx上拉电阻。

/PHPBx=0 时, 开启PBx上拉电阻。

### 3.1.13 INT0 (中断使能寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INT0	R	0xE	INT0IE	WDTIE	T2IE/CCPIE	LVDIE	T1IE	INT0IE	PABIE	T0IE
读/写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	0	0	0	0	0	0	0

Bit0 **T0IE**: 定时器 0 上溢 (overflow) 中断使能位。

T0IE=1 时, 开启定时器 0 上溢中断。

T0IE=0 时, 关闭定时器 0 上溢中断。

Bit1 **PABIE**: PortA / PortB输入状态变化中断使能位。

PABIE=1 时, 开启PortA/ PortB输入状态变化中断。

PABIE=0 时, 关闭PortA/ PortB输入状态变化中断。

Bit2 **INT0IE**: 外部中断 0 使能位。

INT0IE=1 时, 开启外部中断 0。

INT0IE=0 时, 关闭外部中断 0。

Bit3 **T1IE**: 定时器 1 下溢 (underflow) 中断使能位。

T1IE=1 时, 开启定时器 1 下溢中断。

T1IE=0 时, 关闭定时器 1 下溢中断。

Bit4 **LVDIE**: 低电压侦测中断使能位。

LVDIE=1 时, 开启低电压侦测中断。

LVDIE=0 时, 关闭低电压侦测中断。

Bit5 **T2IE/CCPIE**: 当CCP捕捉或比较模式使能时, 此中断为CCP中断使能位, 否则为Timer2 下溢中断使能位。

T2IE=1 时, 开启中断。

T2IE=0 时, 关闭中断。

Bit6 **WDTIE**: WDT超时中断使能位。

WDTIE=1 时, 开启WDT超时中断。

WDTIE=0 时, 关闭WDT超时中断。

Bit7 **INT1IE**: 外部中断 1 使能位。

INT1IE=1 时, 开启外部中断 1。

INT1IE=0 时, 关闭外部中断 1。

### 3.1.14 INTF (中断标志寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF	R	0xF	INT1IF	WDTIF	T2IF/CCPIF	LVDIF	T1IF	INT0IF	PABIF	T0IF
读/写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值(note*)			0	0	0	0	0	0	0	0

Bit0 **T0IF**: 定时器 0 上溢中断标志位。

T0IF=1 时, 发生定时器 0 上溢中断。

T0IF=0 必须由程序清零。

Bit1 **PABIF**: PortA / PortB输入状态变化中断标志位。

PABIF=1 时, 发生PortA / PortB输入状态变化中断。

PABIF=0 必须由程序清零。

Bit2 **INT0IF**: 外部中断 0 标志位。

INT0IF=1 时, 发生外部 0 中断。

INT0IF=0 必须由程序清零。

Bit3 **T1IF**: 定时器 1 下溢中断标志位。

T1IF=1 时, 发生定时器 1 下溢中断。

T1IF=0 必须由程序清零。

Bit4 **LVDIF**: 低电压侦测中断标志位。

LVDIF=1, 发生低电压侦测中断。

LVDIF=0 必须由程序清零。

Bit5 **T2IF/CCPIF**: 当CCP捕捉或比较模式使能时, 此中断为CCP中断标志位, 否则为T2 下溢中断标志位。

T2IF=1 时, 发生中断。

T2IF=0 必须由程序清零。

Bit6 **WDTIF**: WDT超时标志位。

WDTIF=1 时, 发生WDT超时中断。

WDTIF=0 必须由程序清零。

Bit7 **INT1IF**: 外部中断 1 标志位。

INT1IF=1 时, 发生外部 1 中断。

INT1IF=0 必须由程序清零。

**注意: 当对应的INTE寄存器控制位未使能, 读取中断标志是0。**

## 3.1.15 ADMD (ADC 模式寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADMD	R	0x10	ADEN	START	EOC	GCHS	CHS3	CHS2	CHS1	CHS0
读/写属性			读/写	写	读	读/写	读/写	读/写	读/写	读/写
初始值			0	0	1	0	0	0	0	0

Bit[3:0] **CHS3~0**: ADC模拟输入通道选择位。

- 0000: 选择PA0 引脚为模拟输入通道。
- 0001: 选择PA1 引脚为模拟输入通道。
- 0010: 选择PA2 引脚为模拟输入通道。
- 0011: 选择PA3 引脚为模拟输入通道。
- 0100: 选择PA4 引脚为模拟输入通道。
- 0101: 选择PB0 引脚为模拟输入通道。
- 0110: 选择PB1 引脚为模拟输入通道。
- 0111: 选择PB2 引脚为模拟输入通道。
- 1000: 选择PB3 引脚为模拟输入通道。
- 1001: 选择PB4 引脚为模拟输入通道。
- 1010: 选择PB5 引脚为模拟输入通道。
- 1011: 选择内部 1/4 VDD为模拟输入通道。
- 1100: 选择GND为模拟输入通道。

Bit4 **GCHS**: 开启/关闭ADC总通道。

GCHS=0 时, 关闭所有ADC模拟输入通道。

GCHS=1 时, 开启ADC模拟输入通道。

Bit5 **EOC**: ADC状态位, 只读。

EOC=1 时, ADC转换完成。可由ADR与ADD读取转换结果数据。

EOC=0 时, ADC转换中。

Bit6 **START**: ADC转换启动位。

写 1 开始执行ADC转换。此位只能写, 读取此位将得到 0。

Bit7 **ADEN**: 开启/关闭ADC功能。

ADEN=1 时, 开启ADC功能。

### 3.1.16 ADR (ADC 时钟, ADC 中断标志位与 ADC 低四位输出寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADR	R	0x11	ADIF	ADIE	ADCK1	ADCK0	AD3	AD2	AD1	AD0
读/写属性			读/写	读/写	读/写	读/写	读	读	读	读
初始值			0	0	0	0	X	X	X	X

Bit[3:0] **AD3~0**: ADC转换结果低四位数据缓存器。

Bit[5:4] **ADCK1~0**: ADC时钟选择位。

00: ADC 时钟= $F_{INST}/16$ 。

01: ADC 时钟= $F_{INST}/8$ 。

10: ADC 时钟= $F_{INST}/1$ 。

11: ADC 时钟= $F_{INST}/2$ 。

Bit6 **ADIE**: ADC中断使能位。

ADIE=1 时, 开启ADC中断。

ADIE=0 时, 关闭ADC中断。

Bit7 **ADIF**: ADC中断标志位。

ADIF=1 时, 发生ADC转换完成中断。

ADIF=0 必须由程序清零。

### 3.1.17 ADD (ADC 输出数据寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADD	R	0x12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4
读/写属性			读	读	读	读	读	读	读	读
初始值			0	0	0	0	0	0	0	0

Bit[7:0] **AD11~4**: ADC高八位数据缓存器。

### 3.1.18 ADVREFH (ADC 参考电压寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADVREFH	R	0x13	EVHENB	-	-	-	-	-	VHS1	VHS0
读/写属性			读/写	-	-	-	-	-	读/写	读/写
初始值			0	X	X	X	X	X	1	1

Bit7 **EVHENB**: ADC参考电压 (VREFH) 选择控制位。

EVHENB=0 时, ADC参考电压由内部产生, 参考电压水平由VHS1~0 决定。

EVHENB=1 时, ADC参考电压由引脚PA0 提供。

Bit [1:0] **VHS1~0**: ADC内部参考电压选择位。

- 11: VREFH=VDD。
- 10: VREFH=4V。
- 01: VREFH=3V。
- 00: VREFH=2V。

### 3.1.19 ADCR (ADC 采样时间与 ADC 位数寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCR	R	0x14	-	PBCON5	PBCON4	PBCON3	SHCK1	SHCK0	ADCR1	ADCR0
读/写属性			-	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	0	0	0	1	0	1	0

Bit[1:0] **ADCR1~0**: ADC位数选择位。

- 00: 8 位ADC。
- 01: 10 位ADC。
- 1x: 12 位ADC。

Bit[3:2] **SHCK1~0**: ADC采样时间选择位。

- 00: 1 个ADC时钟。
- 01: 2 个ADC时钟。
- 10: 4 个ADC时钟。
- 11: 8 个ADC时钟。

Bit[6:4] **PBCONx**: PB模拟引脚选择位,  $3 \leq x \leq 5$ 。

- 0=PBx 作为ADC模拟输入引脚或数字IO引脚。
- 1=PBx 仅作为ADC模拟输入引脚, 可省电。

### 3.1.20 AWUCON (PortA 唤醒控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AWUCON	R	0x15	WUPA7	WUPA6	WUPA5	WUPA4	WUPA3	WUPA2	WUPA1	WUPA0
读/写属性			读/写							
初始值			1	1	1	1	1	1	1	1

Bit[7:0] **WUPAx**: 开启/关闭PAx唤醒功能,  $0 \leq x \leq 7$ 。

- WUPAx=1 时, 开启PAx唤醒功能。
- WUPAx=0 时, 关闭PAx唤醒功能。

### 3.1.21 PACON (ADC 模拟引脚寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PACON	R	0x16	PBCON2	PBCON1	PBCON0	PACON4	PACON3	PACON2	PACON1	PACON0
读/写属性			读/写							
初始值			0	0	0	0	0	0	0	0

Bit[4:0] **PACONx**: PA引脚选择位,  $0 \leq x \leq 4$ 。

0=PAx 作为ADC模拟输入引脚或数字IO引脚。

1=PAx 仅作为ADC模拟输入引脚, 可省电。

Bit[7:5] **PBCONx**: PB引脚选择位,  $0 \leq x \leq 2$ 。

0=PBx 作为ADC模拟输入引脚或数字IO引脚。

1=PBx 仅作为ADC模拟输入引脚, 可省电。

### 3.1.22 ADJMD (ADC 调校模式)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADJMD	R	0x17	-	-	ADJ_SIGN	ADJ[4]	ADJ[3]	ADJ[2]	ADJ[1]	ADJ[0]
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	0	0	0	0	0	0

Bit[4:0] **ADJ[x]**: 调校位选择,  $0 \leq x \leq 4$ 。

00000: 偏移 0mV。

11111: 偏移 12.5mV。

Bit5 **ADJ\_SIGN**: 调校标志位。

0: ADC数据递减。

1: ADC数据递增。

**注意: 应用时, 请参考NYIDE范例 “ADC\_Interrupt\_AutoK”。**

### 3.1.23 INTEDG (外部中断控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTEDG	R	0x18	-	-	EIS1	EIS0	INT1G1	INT1G0	INT0G1	INT0G0
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	0	0	0	1	0	1

**INT0G1~0**: INT0 边沿触发选择位。

00: 保留。

01: 上升沿触发。

10: 下降沿触发。

11: 上升/下降沿触发。

Bit[3:2] **INT0G1~0**: INT0 边沿触发选择位。

- 00: 保留。
- 01: 上升沿触发。
- 10: 下降沿触发。
- 11: 上升/下降沿触发。

Bit4 **EIS0**: 外部中断 0 引脚选择位。

- EIS0=1 时, PBO 选择为外部中断 0 引脚。
- EIS0=0 时, PBO 选择为GPIO。

Bit5 **EIS1**: 外部中断 1 引脚选择位。

- EIS1=1 时, PB1 选择为外部中断 1 引脚。
- EIS1=0 时, PB1 选择为GPIO。

### 3.1.24 TMRH (定时器高字节寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMRH	R	0x19	TMR29	TMR28	TMR19	TMR18	PWM2 DUTY9	PWM2 DUTY8	PWM1 DUTY9	PWM1 DUTY8
读/写属性			读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	X	X	X	X	X	X

Bit[1:0] **PWM1DUTY9~8**: PWM1 占空比的高 2 位。

Bit[3:2] **PWM2DUTY9~8**: PWM2 占空比的高 2 位。

Bit[5:4] **TMR19~8**: 定时器 1 高 2 位。写这 2 位将覆写定时器 1 第 9 位与第 8 位重载值。

读取这 2 位将得到定时器 1 第 9 位与第 8 位目前计数值。

Bit[7:6] **TMR29~8**: 定时器 2 高 2 位。写这 2 位将覆写定时器 2 第 9 位与第 8 位重载值。

读取这 2 位将得到定时器 2 第 9 位与第 8 位目前计数值。

### 3.1.25 ANAEN (比较器使能寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANAEN	R	0x1A	CMPEN	-	-	-	-	-	-	-
读/写属性			读/写	-	-	-	-	-	-	-
初始值			0	X	X	X	X	X	X	X

Bit7 **CMPEN**: 开启/关闭电压比较器。

CMPEN=1 时, 开启电压比较器。

CMPEN=0 时, 关闭电压比较器。

### 3.1.26 RFC (电阻频率转换控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RFC	R	0x1B	RFCEN	-	-	-	PSEL[3:0]			
读/写属性			读/写	-	-	-	读/写			
初始值			0	X	X	X	0			

Bit[3:0] **RFCEN**: 关闭/开启RFC功能。

RFCEN=1, 开启RFC功能。

RFCEN=0, 关闭RFC功能。

Bit7 **PSEL[3:0]**: 选择RFC输入引脚。

PSEL[3:0]	RFC PAD
0000	PA0
0001	PA1
0010	PA2
0011	PA3
0100	PA4
0101	PA5
0110	PA6
0111	PA7
1000	PB0
1001	PB1
1010	PB2
1011	PB3
1100	PB4
1101	PB5

表 3 选择RFC输入引脚

### 3.1.27 TM3RH (定时器 3 高字节寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM3RH	R	0x1C	-	-	TMR39	TMR38	PWM4 DUTY9	PWM4 DUTY8	PWM3 DUTY9	PWM3 DUTY8
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			-	-	X	X	X	X	X	X

Bit[1:0] **PWM3DUTY9~8**: PWM3 占空比高 2 位。

Bit[3:2] **PWM4DUTY9~8**: PWM4 占空比高 2 位。

Bit[5:4] **TMR39~38**: 定时器 3 高 2 位。写这 2 位将覆写定时器 3 第 9 位与第 8 位重载值。

读取这 2 位将得到定时器 3 第 9 位与第 8 位目前计数值。

### 3.1.28 INTE2 (第 2 个中断使能和标志寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE2	R	0x1F	-	-	-	T3IF	-	-	-	T3IE
读/写属性			-	-	-	读/写	-	-	-	读/写
初始值			-	-	-	0	-	-	-	0

Bit0 **T3IE**: 定时器 3 下溢 (underflow) 中断使能位。

T3IE=1 时, 开启定时器 3 下溢中断。

T3IE=0 时, 关闭定时器 3 下溢中断。

Bit4 **T3IF**: 定时器 3 下溢中断标志位。

T3IF=1 时, 发生定时器 3 下溢中断。

T3IF=0 必须由程序清零。

## 3.2 T0MD (定时器 0 控制寄存器)

T0MD 是可读/写寄存器, 但只能由指令 T0MD / T0MDR 存取。

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T0MD	-	-	LCKTM0	GP6	T0CS	T0CE	PS0WDT	PS0SEL[2:0]		
读/写属性										
初始值(注意*)			0	0	1	1	1	111		

Bit[2:0] **PS0SEL[2:0]**: 选择预分频器 0 的预分频比 (Dividing Rate)。预分频器 0 根据 PS0WDT 控制位决定分配给

定时器 0 或 WDT。当预分频器 0 被分配给 WDT, 预分频比取决于选择哪种计数机制 (WDT 复位或 WDT 中断)。

PS0SEL[2:0]	预分频比		
	PS0WDT=0 (Timer0)	PS0WDT=1 (WDT Reset)	PS0WDT=1 (WDT Interrupt)
000	1:2	1:1	1:2
001	1:4	1:2	1:4
010	1:8	1:4	1:8
011	1:16	1:8	1:16
100	1:32	1:16	1:32
101	1:64	1:32	1:64
110	1:128	1:64	1:128
111	1:256	1:128	1:256

表 4 预分频器 0 的预分频比选项

Bit3 **PS0WDT**: 预分频器 0 分配选择。

PS0WDT=1 时, 预分频器 0 被分配到 WDT。

PS0WDT=0 时, 预分频器 0 被分配到定时器 0。

**注意：在开启看门狗或定时器中断前，要先设定PS0WDT和PS0SEL[2:0]，否则复位或中断可能导致错误触发。**

Bit4 **T0CE**: 定时器 0 外部时钟源触发沿选择。

T0CE=1 时，EX\_CKIO 有上升沿时定时器 0 加一。

T0CE=0 时，EX\_CKIO 有下降沿时定时器 0 加一。

**注意：T0CE也适用于低振荡器频率作为Timer0 时钟源条件。**

Bit5 **T0CS**: 定时器 0 时钟源选择。

T0CS=1 时，选择EX\_CKIO 脚或低频振荡I\_LRC/E\_LXT。

T0CS=0 时，选择指令时钟F<sub>INST</sub>。

Bit6 **GP6**: 通用读写器寄存器位。

Bit7 **LCKTM0**: 当T0CS=1 时，定时器 0 可被随意选择作为低频振荡器。

T0CS=0 时，指令时钟F<sub>INST</sub>被选作定时器 0 时钟源。

T0CS=1 时，LCKTM0=0 时，外部EX\_CKIO 脚被选择当作定时器 0 时钟源。

T0CS=1 时，LCKTM0=1 时，低频振荡（I\_LRC或E\_LXT由配置字决定）输出代替引脚EX\_CKIO 作为定时器 0 时钟源。

**注意：有关定时器 0 时钟源选择的详细说明，请参考定时器 0 章节。**

### 3.3 F-page 特殊功能寄存器

#### 3.3.1 IOSTA (PortA I/O 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTA	F	0x5	IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0
	读/写属性		读/写							
	初始值		1	1	1	1	1	1	1	1

Bit[7:0] **IOPAx**: PAx I/O模式选择， $0 \leq x \leq 7$ 。

IOPAx=1 时，PAx设为输入口。

IOPAx=0 时，PAx设为输出口。

#### 3.3.2 IOSTB (PortB I/O 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOSTB	F	0x6	-	-	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0
	读/写属性		-	-	读/写	读/写	读/写	读/写	读/写	读/写
	初始值		X	X	1	1	1	1	1	1

Bit[5:0] **IOPBx**: PBx I/O模式选择， $0 \leq x \leq 5$ 。

IOPBx=1 时，PBx设为输入口。

IOPBx=0 时，PBx设为输出口。

### 3.3.3 APHCON (PortA 上拉电阻控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
APHCON	F	0x9	/PHPA7	/PHPA6	/PLPA5	/PHPA4	/PHPA3	/PHPA2	/PHPA1	/PHPA0
读/写属性			读/写							
初始值			1	1	1	1	1	1	1	1

/PHPAx: 开启/关闭 PAx上拉电阻, x=0~4, 6~7。

/PHPAx=1 时, 关闭 PAx上拉电阻。

/PHPAx=0 时, 开启 PAx上拉电阻。

Bit5 /PLPA5: 开启/关闭 PA5 下拉电阻。

/PLPA5=1 时, 关闭 PA5 下拉电阻。

/PLPA5=0 时, 开启 PA5 下拉电阻。

**注意: PA6 与 PA7 作为晶振引脚时应关闭内置上拉电阻, 否则振荡可能失败。**

### 3.3.4 PS0CV (预分频器 0 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS0CV	F	0xA	PS0CV[7:0]							
读/写属性			读							
初始值			1	1	1	1	1	1	1	1

读取PS0CV时, 会得到预分频器 0 寄存器的目前计数值。

### 3.3.5 BODCON (PortB 开漏控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BODCON	F	0xC	-	-	ODPB5	ODPB4	ODPB3	ODPB2	ODPB1	ODPB0
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	0	0	0	0	0	0

Bit[5:0] ODPBx: 开启/关闭PBx的开漏,  $0 \leq x \leq 5$ 。

ODPBx=1 时, 开启PBx的开漏。

ODPBx=0 时, 关闭PBx的开漏。

### 3.3.6 CMPCR (比较器控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	F	0xE	PS3	PS2	PS1	PS0	VS3	VS2	VS1	VS0
读/写属性			读/写							
初始值			0	0	0	0	1	1	0	0

VS[3:0], PS[3:0]: 当VS[3:0]=0 时, 为P2P模式。VS[3:0]不为 0 时为P2V模式。

当比较器处于P2V模式时, VS[3:0]从 15 个参考电压中选择一个作为比较器的负输入源。PS[3:0]决定 11 个引脚中的一个作为比较器的正输入源。

当比较器处于P2P模式时, VS[3:0]为固定 0, PS[3:0]从 4 个组合中选择 2 个脚位作为比较器的负输入源和正输入源。有关P2P模式的详细信息, 请参阅功能描述比较器部分。

VS[3:0]	比较器参考电压	PS[3:0]	选择引脚
0000	P2P mode	0000	PA0
0001	1 / 16 V <sub>DD</sub>	0001	PA1
0010	2 / 16 V <sub>DD</sub>	0010	PA2
0011	3 / 16 V <sub>DD</sub>	0011	PA3
0100	4 / 16 V <sub>DD</sub>	0100	-
0101	5 / 16 V <sub>DD</sub>	0101	-
0110	6 / 16 V <sub>DD</sub>	0110	-
0111	7 / 16 V <sub>DD</sub>	0111	-
1000	8 / 16 V <sub>DD</sub>	1000	-
1001	9 / 16 V <sub>DD</sub>	1001	-
1010	10 / 16 V <sub>DD</sub>	1010	-
1011	11 / 16 V <sub>DD</sub>	1011	-
1100	12 / 16 V <sub>DD</sub>	1100	-
1101	13 / 16 V <sub>DD</sub>	1101	-
1110	14 / 16 V <sub>DD</sub>	1110	-
1111	15 / 16 V <sub>DD</sub>	1111	-

表 5 P2V 模式

PS[3:0]	比较器正输入源	比较器负输入源
0000	PA0	PA1
0001	PA1	PA0
0010	PA2	PA3
0011	PA3	PA2

表 6 P2P模式 (VS[3:0] = 4'b0000)

### 3.3.7 PCON1 (Power 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON1	F	0xF	GIE	LVDOU	GP5	LVDS2	LVDS1	LVDS0	GP1	T0EN
读/写属性			读/写 <sup>*1</sup>	读	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	X	0	1	1	1	0	1

Bit0 **T0EN**: 开启/关闭定时器 0。

T0EN=1 时, 开启定时器 0。

T0EN=0 时, 关闭定时器 0。

Bit1 **GP1**: 通用寄存器数据位。

Bit[4:2] **LVDS2~0**: 从 8 个LVD电压中选择一个。

LVDS[2:0]	电压
000	2.0V
001	2.2V
010	2.4V
011	2.7V
100	3.0V
101	3.3V
110	3.6V
111	4.3V

表 7 LVD 电压选择

Bit5 **GP5**: 通用寄存器数据位。

Bit6 **LVDOUT**: 低电压检测输出, 只读。

Bit7 **GIE**: 总中断使能位。

**GIE=1** 时, 开启总中断。

**GIE=0** 时, 关闭总中断。

(1\*) : 由指令 ENI 设置 1、指令 DISI 清除、指令 IOSTR 所读取。

### 3.4 S-page 特殊功能寄存器

#### 3.4.1 TMR1 (定时器 1 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
TMR1	S	0x0	TMR1[7:0]									
读/写属性			读/写									
初始值			XXXXXXXX									

当读取TMR1 寄存器时, 会得到 10 位定时器 1 中的低字节目前计数值。写TMR1 时, 会将TMRH[5:4]和 TMR1[9:0]一起写到定时器 1 重载寄存器中。

#### 3.4.2 T1CR1 (定时器 1 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR1	S	0x1	PWM1OEN	PWM1OAL	-	-	TM1_HRC	T1OS	T1RL	T1EN
读/写属性			读/写	读/写	-	-	读/写	读/写	读/写	读/写
初始值			0	0	X	X	0	0	0	0

此寄存器用于配置定时器 1 功能。

Bit0 **T1EN**: 开启/关闭定时器 1。

T1EN=1 时, 开启定时器 1。

T1EN=0 时, 关闭定时器 1。

Bit1 **T1RL**: 当连续模式被选择 (T1OS=0), 选择定时器 1 下数方式。

T1RL=1 时, 当下溢发生, 定时器 1 初始值从 TMR1[9:0] 寄存器被重新加载。

T1RL=0 时, 当下溢发生, 定时器 1 继续从 0x3FF 下数。

Bit2 **T1OS**: 当下溢发生, 设置定时器 1 操作模式。

T1OS=1 时, 单次计数模式 (One-Shot mode)。定时器 1 会从初始值到 0x00 计数一次。

T1OS=0 时, 连续计数模式 (Non-Stop mode)。下溢后, 定时器 1 会持续下数。

T1OS		T1RL		定时器 1 计数选项					
0	0	定时器 1 从重载的数值下数到 0x00。 当下溢发生, 0x3FF 被重载至定时器 1 并继续下数。							
0	1	定时器 1 从重载的数值下数到 0x00。 当下溢发生, 重载值重新加载, 并继续下数。							
1	x	定时器 1 从初始值下数到 0x00。 当下溢发生, 定时器 1 停止下数。							

表 8 定时器 1 功能

Bit3 **TM1\_HRC**: Timer1 时钟源来自内部高频 (I\_HRCx1)。

TM1\_HRC = 1 开启。Timer1 时钟源来自内部高频 (I\_HRCx1)

TM1\_HRC = 0 关闭。Timer1 时钟源来自 CPU 运行时钟。

**注意: 当定时器通过合格的预分频器时 (TMx\_HRC=1) 时, 其最小工作 LVR 电压会更高。**

Bit6 **PWM1OAL**: 定义 PWM1 输出有效状态。

PWM1OAL=1 时, PWM1 输出低电平有效。

PWM1OAL=0 时, PWM1 输出高电平有效。

Bit7 **PWM1OEN**: 开启/关闭 PWM1 输出。

PWM1OEN=1 时, PB3 输出 PWM1。

PWM1OEN=0 时, PB3 为 GPIO。

### 3.4.3 T1CR2 (定时器 1 控制寄存器 2)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR2	S	0x2	-	-	T1CS	T1CE	/PS1EN	PS1SEL[2:0]		
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

这个寄存器用于配置 Timer1 功能。

Bit[2:0] **PS1SEL[2:0]**: 预分频器 1 预分频比选项。

PS1SEL[2:0]	预分频比选项
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

表 9 预分频器 1 预分频比选项

**注意：在PS1EN=1 时须设定PS1SEL[2 : 0]，否则可能会误触发中断。**

Bit3 /PS1EN：关闭/开启预分频器 1。

/PS1EN=1 时，关闭预分频器 1。

/PS1EN=0 时，开启预分频器 1。

Bit4 T1CE：定时器 1 外部时钟触发沿选项。

T1CE=1 时，EX\_CKIO 脚有下降沿时定时器 1 减一。

T1CE=0 时，EX\_CKIO 脚有上升沿时定时器 1 减一。

Bit5 T1CS：定时器 1 时钟源选项。

T1CS=1 时，选择EX\_CKIO 脚作为外部时钟输入。

T1CS=0 时，选择指令时钟F<sub>INST</sub>。

#### 3.4.4 PWM1DUTY (PWM1 占空比寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DUTY	S	0x3	PWM1DUTY[7:0]							
读/写属性			写							
初始值			XXXXXXXX							

定时器 1 重新加载的数值储存在TMRH[5:4]与TMR1[7:0]寄存器，以用来定义 PWM1 帧率，TMRH[1:0]与 PWM1DUTY[7:0]寄存器用于定义 PWM1 的占空比。

#### 3.4.5 PS1CV (预分频器 1 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS1CV	S	0x4	PS1CV[7:0]							
读/写属性			读							
初始值			1	1	1	1	1	1	1	1

读取PS1CV时，将会得到预分频器 1 的目前数值。

## 3.4.6 BZ1CR (蜂鸣器 1 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BZ1CR	S	0x5	BZ1EN	-	-	-	BZ1FSEL[3:0]			
读/写属性			写	-	-	-	写			
初始值			0	X	X	X	1	1	1	1

Bit[3:0] BZ1FSEL[3:0]: BZ1 输出频率选项。

BZ1FSEL[3:0]	BZ1 频率选项	
	时钟源	预分频比
0000	预分频器 1 输出	1:2
0001		1:4
0010		1:8
0011		1:16
0100		1:32
0101		1:64
0110		1:128
0111		1:256
1000	定时器 1 输出	Timer1 bit 0
1001		Timer1 bit 1
1010		Timer1 bit 2
1011		Timer1 bit 3
1100		Timer1 bit 4
1101		Timer1 bit 5
1110		Timer1 bit 6
1111		Timer1 bit 7

表 10 蜂鸣器BZ1 输出频率选项

Bit7 BZ1EN: 开启/关闭蜂鸣器 1 输出。

BZ1EN=1 时, 开启蜂鸣器 1。

BZ1EN=0 时, 关闭蜂鸣器 1。

## 3.4.7 IRCR (IR 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCR	S	0x6	IROSC358M	-	-	-	-	IRCSEL	IRF57K	IREN
读/写属性			写	-	-	-	-	写	写	写
初始值			0	X	X	X	X	0	0	0

Bit0 IREN: 开启/关闭IR载波输出。

IREN=1 时, 开启IR载波输出。

IREN=0 时, 关闭IR载波输出。

Bit1 **IRF57K**: IR载波频率选择。

IRF57K=1 时, IR载波频率是 57KHz。

IRF57K=0 时, IR载波频率是 38KHz。

Bit2 **IRCSEL**: IR载波极性选择。

IRCSEL=0 且I/O脚数据是 1 时, IR载波会被产生。

IRCSEL=1 且I/O脚数据是 0 时, IR载波会被产生。

Bit3 **IROSC358M**: 选择使用的外部晶振频率类型。

若选择 I\_HRC此位将被忽略。

IROSC358M=1, 外部晶振频率请用 3.58MHz。

IROSC358M=0, 外部晶振频率请用 455KHz。

**注意:**

1.仅有高速振荡时钟  $F_{Hosc}$  (详见章节 3.17) 可以当作IR时钟源。

2.不同振荡类型的分频比。

OSC. Type	57KHz	38KHz	条件
High IRC(4MHz)	64	96	HIRC 模式 (不论系统时钟频率是多少, IR 模块的输入时钟都设定为 4MHz)
Xtal 3.58MHz	64	96	Xtal 模式 & IROSC358M=1
Xtal 455KHz	8	12	Xtal 模式 & IROSC358M=0

表 11 不同振荡类型的分频比

### 3.4.8 TBHP (表格指针高字节寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHP	S	0x7	-	-	-	-	-	TBHP2	TBHP1	TBHP0
读/写属性			-	-	-	-	-	读/写	读/写	读/写
初始值			X	X	X	X	X	X	X	X

当指令CALLA、GOTOA或TABLEA被执行时, 程序计数寄存器会指向欲寻址的 11 位ROM地址, 此目标地址是由TBHP[2:0]与ACC组成。ACC是PC[10:0]的低字节, TBHP[2:0]是PC[10:0]的高字节。

### 3.4.9 TBHD (表格数据高字节寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHD	S	0x8	-	-	TBHD5	TBHD4	TBHD3	TBHD2	TBHD1	TBHD0
读/写属性			-	-	读	读	读	读	读	读
初始值			X	X	X	X	X	X	X	X

当指令TABLEA被执行时, 寻址ROM的高字节内容被加载到TBHD[5:0]寄存器。寻址ROM的低字节内容则被加载到ACC。

### 3.4.10 TMR2 (定时器 2 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR2	S	0x9								TMR2[7:0]
读/写属性			读/写							
初始值			XXXXXXXX							

当读取TMR2 寄存器时, 会得到 10 位定时器 2 中的低字节目前计数值。写TMR2 时, 会将TMRH[7:6]和TMR2[7:0]一起写到定时器 2 重载寄存器中。

### 3.4.11 T2CR1 (定时器 2 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
T2CR1	S	0xA	PWM2OEN	PWM2OAL	-	-	TM2_HRC	T2OS	T2RL	T2EN		
读/写属性			读/写									
初始值			0	0	X	X	0	0	0	0		

此寄存器用于配置定时器 2 功能。

Bit0 **T2EN**: 开启/关闭定时器 2。

T2EN=1 时, 开启定时器 2。

T2EN=0 时, 关闭定时器 2。

Bit1 **T2RL**: 当连续模式被选择 (T2OS=0), 选择定时器 2 下数方式。

T2RL=1 时, 当下溢发生, 定时器 2 初始值从TMR2 寄存器被重新加载。

T2RL=0 时, 当下溢发生, 定时器 2 继续从 0x3FF 下数。

Bit2 **T2OS**: 当下溢发生, 设置定时器 2 操作模式。

T2OS=1 时, 单次计数模式 (One-Shot mode)。定时器 2 会从初始值到 0x00 计数一次。

T2OS=0 时, 连续计数模式 (Non-Stop mode)。下溢后, 定时器 2 会持续下数。

Bit3 **TM2\_HRC**: Timer2 时钟源来自内部高频 (I\_HRCx1)。

TM2\_HRC = 1 开启, Timer2 时钟源来自内部高频 (I\_HRCx1)

TM1\_HRC = 0 关闭, Timer2 时钟源来自CPU运行时钟。

**注意: 当定时器通过合格的预分频器时 (TMx\_HRC=1) 时, 其最小工作 LVR 电压会更高。**

T2OS	T2RL	定时器 2 计数选项
0	0	定时器 2 从重载的数值下数到 0x00。 当下溢发生, 0x3FF被重载至定时器 2 并继续下数。
0	1	定时器 2 从重载的数值下数到 0x00。 当下溢发生, 重载值重新加载, 并继续下数。
1	x	定时器 2 从初始值下数到 0x00。 当下溢发生, 定时器 2 停止下数。

表 12 定时器 2 功能

Bit6 **PWM2OAL**: 定义 PWM2 输出有效状态。

PWM2OAL=1 时, PWM2 输出低电平有效。

PWM2OAL=0 时, PWM2 输出高电平有效。

Bit7 **PWM2OEN**: 开启/关闭 PWM2 输出。

PWM2OEN=1, PB2 输出 PWM2。

PWM2OEN=0, PB2 为 GPIO。

### 3.4.12 T2CR2 (定时器 2 控制寄存器 2)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CR2	S	0xB	-	-	T2CS	T2CE	/PS2EN	PS2SEL[2:0]		
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

该寄存器用于配置定时器 2 功能。

Bit[2:0] **PS2SEL[2:0]**: 预分频器 2 的预分频比选项。

PS2SEL[2:0]	预分频比
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

表 13 预分频器 2 预分频比选项

**注意: 在 $\text{PS2EN}=1$  时须先设定 $\text{PS2SEL}[2:0]$ , 否则可能会误触发中断。**

Bit3 **/PS2EN**: 关闭/开启预分频器 2。

/PS2EN=1 时, 关闭预分频器 2。

/PS2EN=0 时, 开启预分频器 2。

Bit4 **T2CE**: 定时器 2 外部时钟触发沿选项。

T2CE=1 时, EX\_CK11 脚有下降沿时定时器 2 减一。

T2CE=0 时, EX\_CK11 脚有上升沿时定时器 2 减一。

Bit5 **T2CS**: 定时器 2 时钟源选项。

T2CS=1 时, 选择 EX\_CK11 脚作为外部时钟输入。

T2CS=0 时, 选择指令时钟  $F_{INST}$ 。

## 3.4.13 PWM2DUTY (PWM2 占空比寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DUTY	S	0xC								PWM2DUTY[7:0]
读/写属性			读/写							

定时器 2 重新加载的数值储存在TMRH[7:6]与TMR2[7:0]寄存器, 以用来定义PWM2 帧率, TMRH[3:2]与PWM2DUTY[7:0]寄存器用于定义PWM2 的占空比。

## 3.4.14 PS2CV (预分频器 2 计数值寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS2CV	S	0xD								PS2CV[7:0]
读/写属性			读							
初始值			1	1	1	1	1	1	1	1

读取PS2CV时, 将会得到预分频器 2 计数器的目前数值。

## 3.4.15 BZ2CR (蜂鸣器 2 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BZ2CR	S	0xE	BZ2EN	-	-	-				BZ2FSEL[3:0]
读/写属性			写	-	-	-	写			
初始值			0	X	X	X	1	1	1	1

Bit[3:0] BZ2FSEL[3:0]: BZ2 输出频率选项。

BZ2FSEL[3:0]	BZ2 频率选项	
	时钟源	预分频比
0000	预分频器 2 输出	1:2
0001		1:4
0010		1:8
0011		1:16
0100		1:32
0101		1:64
0110		1:128
0111		1:256
1000		Timer2 bit 0
1001		Timer2 bit 1
1010	定时器 2 输出	Timer2 bit 2
1011		Timer2 bit 3
1100		Timer2 bit 4

BZ2FSEL[3:0]	BZ2 频率选项	
	时钟源	预分频比
1101		Timer2 bit 5
1110		Timer2 bit 6
1111		Timer2 bit 7

表 14 蜂鸣器BZ2 输出频率选项

Bit7 **BZ2EN**: 开启/关闭蜂鸣器 2 输出。

BZ2EN=1 时, 开启蜂鸣器 2。

BZ2EN=0 时, 关闭蜂鸣器 2。

### 3.4.16 OSCCR (振荡器控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCR	S	0xF	CMPOUT	CMPOE	CMPIF	CMPIE	OPMD[1:0]	STPHOSC	SELHOSC	
读/写属性			读	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	0	0	0	00	0	0	1

Bit0 **SELHOSC**: 系统振荡器选择 (Fosc)。

SELHOSC=1 时, Fosc是高频振荡器 (F<sub>HOSC</sub>)。

SELHOSC=0 时, Fosc是低频振荡器 (F<sub>LOSC</sub>)。

Bit1 **STPHOSC**: 关闭/开启高频振荡器 (F<sub>HOSC</sub>)。

STPHOSC=1 时, F<sub>HOSC</sub>会停止振荡并被关闭。

STPHOSC=0 时, F<sub>HOSC</sub>保持振荡。

Bit[3:2] **OPMD[1:0]**: 选择操作模式。

OPMD[1:0]	操作模式
00	正常模式
01	睡眠模式
10	待机模式
11	保留

表 15 通过OPMD[1:0]选择操作模式

Bit4 **CMPIE**: 开启/关闭比较器中断。

CMPIE=1 时, 开启比较器中断。

CMPIE=0 时, 关闭比较器中断。

Bit5 **CMPIF**: 比较器输出翻转中断标志位。

CMPIF=1, 发生比较器输出翻转中断。

CMPIF必须由程序清零。

Bit6 **CMPOE**: 开启/关闭比较器输出到PB3 引脚。

CMPOE=1 时, 开启比较器输出到PB3 引脚。

CMPOE=0 时, 关闭比较器输出到PB3 引脚。

**注意: 比较器输出到 PB3 引脚优先于 PWM1/BUZZER1。**

Bit7 **CMPOUT**: 比较器输出状态位, 只读。

**注意: STPHOSC 不能与SELHOSC 或OPMD 同时更改。在SELHOSC=1 时, STPHOSC 不能与OPMD 同时更改。**

### 3.4.17 TMR3 (定时器 3 寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMR3	S	0x10								TMR3[7:0]
读/写属性			读/写							
初始值			XXXXXXXX							

当读取TMR3 寄存器时, 会得到 10 位定时器 3 中的低字节目前计数值。写TMR3 时, 会将TM3RH[5:4]和TMR3[7:0]一起写到定时器 3 重载寄存器中。

### 3.4.18 T3CR1 (定时器 3 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
T3CR1	S	0x11	PWM3OEN	PWM3OAL	-	-	TM3_HRC	T3OS	T3RL	T3EN		
读/写属性			读/写									
初始值			0	0	X	X	0	0	0	0		

此寄存器用于配置定时器 3 功能。

Bit0 **T3EN**: 开启/关闭定时器 3。

T3EN=1 时, 开启定时器 3。

T3EN=0 时, 关闭定时器 3。

Bit1 **T3RL**: 当连续模式被选择 (T3OS=0), 选择定时器 3 下数方式。

T3RL=1 时, 当下溢发生, 定时器 3 初始值从TMR3[9:0]寄存器被重新加载。

T3RL=0 时, 当下溢发生, 定时器 3 继续从 0x3FF下数。

Bit2 **T3OS**: 当下溢发生, 设置定时器 3 操作模式。

T3OS=1 时, 单次计数模式 (One-Shot mode)。定时器 3 会从初始值到 0x00 计数一次。

T3OS=0 时, 连续计数模式 (Non-Stop mode)。下溢后, 定时器 3 会持续下数。

T3OS	T3RL	定时器 3 计数选项
0	0	定时器 3 从重载的数值下数到 0x00。 当下溢发生, 0x3FF被重载至定时器 3 并继续下数。
0	1	定时器 3 从重载的数值下数到 0x00。 当下溢发生, 重载值重新加载, 并继续下数。
1	x	定时器 3 从初始值下数到 0x00。 当下溢发生, 定时器 3 停止下数。

表 16 定时器 3 功能

Bit3 **TM3\_HRC**: Timer3 时钟源来自内部高频 (I\_HRCx1)。

TM1\_HRC = 1 开启, Timer3 时钟源来自内部高频 (I\_HRCx1)

TM1\_HRC = 0 关闭, Timer3 时钟源来自CPU运行时钟。

Bit6 **PWM3OAL**: 定义 PWM3 输出有效状态。

PWM3OAL=1 时, PWM3 输出低电平有效。

PWM3OAL=0 时, PWM3 输出高电平有效。

Bit7 **PWM3OEN**: 开启/关闭 PWM3 输出。

PWM3OEN=1, PA2 输出 PWM3。

PWM3OEN=0, PA2 为 GPIO。

### 3.4.19 T3CR2 (定时器 3 控制寄存器 2)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3CR2	S	0x12	-	-	T3CS	T3CE	/PS3EN	PS3SEL[2:0]		
读/写属性			-	-	读/写	读/写	读/写	读/写	读/写	读/写
初始值			X	X	1	1	1	1	1	1

该寄存器用于配置定时器 3 功能。

Bit[2:0] **PS3SEL[2:0]**: 预分频器 3 预分频比选项。

PS3SEL[2:0]	预分频比
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

表 17 预分频器 3 预分频比选项

**注意: 在PS3EN=1 时须设定PS3SEL[2:0], 否则可能会误触发中断。**

Bit3 **/PS3EN**: 关闭/开启预分频器 3。

/PS3EN=1 时, 关闭预分频器 3。

/PS3EN=0 时, 开启预分频器 3。

Bit4 **T3CE**: 定时器 3 外部时钟触发沿选项。

T3CE=1 时, EX\_CK1 脚有下降沿时定时器 3 减一。

T3CE=0 时, EX\_CK1 脚有上升沿时定时器 3 减一。

Bit5 **T3CS**: 定时器 3 时钟源选项。

T3CS=1 时, 选择 EX\_CK1 脚作为外部时钟输入。

T3CS=0 时, 选择指令时钟  $F_{INST}$ 。

#### 3.4.20 PWM3DUTY (PWM3 占空比寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3DUTY	S	0x13	PWM3DUTY[7:0]							
读/写属性			写							
初始值			XXXXXXXX							

定时器 3 重新加载的数值储存在 TM3RH[5:4]与 TMR3[7:0]寄存器, 以用来定义 PWM3 帧率, TM3RH[1:0]与 PWM3DUTY[7:0]寄存器用于定义 PWM3 的占空比。

#### 3.4.21 PS3CV (预分频器 3 计数值寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS3CV	S	0x14	PS3CV[7:0]							
读/写属性			读							
初始值			1	1	1	1	1	1	1	1

读取PS3CV时, 将会得到预分频器 3 的目前数值。

#### 3.4.22 BZ3CR (蜂鸣器 3 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
BZ3CR	S	0x15	BZ3EN	-	-	-	BZ3FSEL[3:0]				
读/写属性			写	-	-	-	写				
初始值			0	X	X	X	1	1	1	1	

Bit[3:0] **BZ3FSEL[3:0]**: BZ3 输出频率选项。

BZ3FSEL[3:0]	BZ3 频率选项	
	时钟源	预分频比
0000	预分频器 3 输出	1:2
		1:4
		1:8
		1:16

BZ3FSEL[3:0]	BZ3 频率选项	
	时钟源	预分频比
0100	定时器 3 输出	1:32
0101		1:64
0110		1:128
0111		1:256
1000		Timer3 bit 0
1001		Timer3 bit 1
1010		Timer3 bit 2
1011		Timer3 bit 3
1100		Timer3 bit 4
1101		Timer3 bit 5
1110		Timer3 bit 6
1111		Timer3 bit 7

表 18 蜂鸣器BZ3 输出频率选项

Bit7 **BZ3EN**: 开启/关闭蜂鸣器 3 输出。

BZ3EN=1 时, 开启蜂鸣器 3。

BZ3EN=0 时, 关闭蜂鸣器 3。

### 3.4.23 P4CR1 (PWM4 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P4CR1	S	0x16	PWM4OEN	PWM4OAL	-	-	-	-	-	-
读/写属性			读/写	读/写	-	-	-	-	-	-
初始值			0	0	X	X	X	X	X	X

Bit6 **PWM4OAL**: 定义 PWM4 输出有效状态。

PWM4OAL=1 时, PWM4 输出低电平有效。

PWM4OAL=0 时, PWM4 输出高电平有效。

Bit7 **PWM4OEN**: 开启/关闭 PWM4 输出。

PWM4OEN=1, PA3 或PA7 输出PWM4。

PWM4OEN=0, PA3 或PA7 为GPIO。

**注意: PWM4 输出由 NYIDE 配置块设置定义。**

### 3.4.24 PWM4DUTY (PWM4 占空比寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
PWM4DUTY	S	0x18	PWM4DUTY[7:0]									
读/写属性			写									
初始值			XXXXXXXX									

用寄存器 TM3RH[5:4]和TMR3[7:0]上存储的 10 位 Timer3 的重载值来定义 PWM4 的帧率, 用寄存器 TM3RH[3:2]和PWM4DUTY[7:0]来定义 PWM4 的占空比。

### 3.4.25 CCPCON (CCP 控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCPCON	S	0x19	PWM2M1	PWM2M0	FBCH1	FBCH0	CCPM3	CCPM2	CCPM1	CCPM0
读/写属性	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值	0	0	0	0	0	0	0	0	0	0

CCPM[3:2]=00/01/10: 捕捉或比较模式。PB2 是捕捉输入或比较输出。

CCPM[3:2]=11:

PWM2M1~0 = 00 → PWM单个输出。

PWM2M1~0 = 01 → PWM全桥正向输出。

PWM2M1~0 = 10 → PWM半桥输出。

PWM2M1~0 = 11 → PWM全桥反向输出。

T2IF/CCPIF=1, 发生中断。

**注意: T2IF/CCPIF必须由程序清零。**

CCPM[3:0]:

0000 = CCP 关闭。

0010 = 比较模式, 在匹配时切换输出。

0100 = 捕捉模式, 捕捉每个下降沿。

0101 = 捕捉模式, 捕捉每个上升沿。

0110 = 捕捉模式, 每 4 个上升沿捕捉一次。

0111 = 捕捉模式, 每 16 个上升沿捕捉一次。

1000 = 比较模式, 在匹配时设置输出和中断。

1001 = 比较模式, 在匹配时清除输出和中断。

1010 = 比较模式, 在匹配时只有中断。

1011 = 比较模式, 在匹配时触发ADC和中断。

1100 = PWM模式, P1A/P1C高有效, P1D/P1B高有效。

1101 = PWM模式, P1A/P1C高有效, P1D/P1B低有效。

1110 = PWM模式, P1A/P1C低有效, P1D/P1B高有效。

1111 = PWM模式, P1A/P1C低有效, P1D/P1B低有效。

注意: P1A是PA0, P1B是PA5, P1C是PA7, P1D是PA4。

FBCH1~0: 全带变向间隙。

00 = 1 个CPU周期。

01 = 4 个CPU周期。

1x = 16 个CPU周期。

注意：比较/捕捉模式步骤：

- a. 设置Timer2 工作模式：设置T1OS/T1RL，设置T2EN = PWM2OEN = 0。
- b. 设置Timer1 时钟源、Timer1 预分频器。
- c. 设置Timer1 / Timer2 重载/初始值。
- d. 设置PWM1DUTY / PWM2DUTY。
- e. 开启比较/捕捉模式。
- f. 在更新PWM / Timer数据前关闭比较/捕捉模式。

#### 3.4.26 PWMDB (死区控制寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDB	S	0x1A	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
读/写属性			读/写							
初始值			0	0	0	0	0	0	0	0

DB[7:0]: 定义 CCP PWM 模式的死区宽度。死区单位在 CPU 周期内。

$$td = F_{INST} * (DB[7:0])$$

#### 3.4.27 P5CR1 (PWM5 控制寄存器 1)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5CR1	S	0x1B	PWM5OEN	PWM5OAL	-	-	-	-	-	-
读/写属性			读/写	读/写	-	-	-	-	-	-
初始值			0	0	X	X	X	X	X	X

Bit6 **PWM5OAL**: 定义 PWM5 输出有效状态。

PWM5OAL=1 时，PWM5 输出低电平有效。

PWM5OAL=0 时，PWM5 输出高电平有效。

Bit7 **PWM5OEN**: 开启/关闭 PWM5 输出。

PWM5OEN=1，PB3 或PB0 输出PWM5。

PWM5OEN=0，PB3 或PB0 为GPIO。

**注意：PWM5 输出由 NYIDE 配置块设置定义。**

#### 3.4.28 PWM5DUTY (PWM5 占空比寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
PWM5DUTY	S	0x1D	PWM5DUTY[7:0]									
读/写属性			写									
初始值			XXXXXXXX									

用寄存器TM3RH[5:4]和TMR3[7:0]上存储的 10 位 Timer3 的重载值来定义 PWM5 的帧率，用寄存器 PWM5RH[1:0]和PWM5DUTY[7:0]来定义 PWM5 的占空比。

### 3.4.29 PWM5RH (MSB[9:8]的 PWM5 占空比寄存器)

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5RH	S	0x1F	-	-	-	-	-	-	PWM5 DUTY9	PWM5 DUTY8
读/写属性			-	-	-	-	-	-	读/写	读/写
初始值			X	X	X	X	X	X	X	X

Bit[1:0] **PWM5 duty[9:8]**: 用来定义 PWM5 的占空比周期MSB。

## 3.5 I/O Port

AT8B62F1 提供 14 个 I/O 口 (PA[7:0] 和 PB[5:0]), 用户可以由寄存器 PORTA 和 PORTB 读写这些脚位。每个 I/O 脚位都有一个对应的寄存器控制位以定义该脚位是输入或输出口, 寄存器 IOSTA[7:0] 定义 PA[7:0] 为输入口或输出口, 寄存器 IOSTB[5:0] 定义 PB[5:0] 为输入口或输出口。

当一个 I/O 脚位被配置为输入口, 它可以由寄存器开启或关闭内部上拉/下拉电阻。寄存器 APHCON[7:6,4:0] 用于开启或关闭 PA[7:6,4:0] 的内部上拉电阻。寄存器 APHCON[5] 和 ABPLCON[3:0] 用于开启或关闭 PA[5, 3:0] 的内部下拉电阻。寄存器 BPHCON[5:0] 用于开启或关闭 PB[5:0] 的内部上拉电阻。寄存器 ABPLCON[7:4] 用于开启或关闭 PB[3:0] 的内部下拉电阻。 PCON[4] 用于开启或关闭 PA[5] 的内部上拉电阻。

当 PortB 的一个 I/O 脚位被配置为输出口时, 有一个对应的且单独的寄存器来选择作为开漏输出脚。寄存器 BODCON[5:0] 决定 PB[5:0] 是否为开漏输出脚。

I/O 口功能摘要如下表:

功能		PA[3:0]	PA[7:6]&PA[4]	PA[5]	PB[3:0]	PB[5:4]
输入	上拉电阻	V	V	V	V	V
	下拉电阻	V	X	V	V	X
输出	开漏	X	X	V	V	V

表 19 I/O 端口功能摘要

在 PA 和 PB 的每个 I/O 脚都有输入状态改变产生中断功能。寄存器 AWUCON[7:0] 和 BWUCON[5:0] 将选择 PA 和 PB 的一个 I/O 引脚产生此中断。只要 AWUCON 和 BWUCON 对应到的任一 PA 和 PB 脚位被置为 1 时, 且在此输入脚位有电平变化时, 寄存器 PABIF (INTF[1]) 就会被设为 1。如果寄存器 PABIE (INTE[1]) 与 GIE (PCON1[7]) 同时设定为 1, 将发生中断要求并执行中断服务程序。

AT8B62F1 提供 2 个外部中断, 当寄存器 EIS0 (INTEDG[4]) 设定为 1, PB0 则被当作外部中断 0 的输入脚。当寄存器 EIS1 (INTEDG[5]) 设定为 1, PB1 则被当作外部中断 1 的输入脚。

**注意: 当 PB0 或 PB1 同时设定成输入状态改变触发脚与外部中断脚, 外部中断有较高的优先权, 而 PB0 或 PB1 输入状态改变触发脚则会被关闭, 但其它输入状态改变触发脚不会被影响。**

AT8B62F1 提供红外线 IR 载波生成器。当 IREN=1 时, PB1 会输出 IR 载波。当 IREN=0 时, 不产生 IR 载波。

由配置字决定PA5 可否当作外部复位输入RSTb。当PA5 为低电平时将导致AT8B62F1 发生复位。

当配置字设置外置晶振 (E\_HXT, E\_XT 或 E\_LXT) 用于高速振荡时钟或低速振荡时钟时, PA6 当作晶振输入引脚 (Xin), PA7 当作晶振输出引脚 (Xout)。

当配置字设置I\_HRC 或 I\_LRC 用于高速振荡时钟或低速振荡时钟时, 用户可以在PA7 输出指令时钟F<sub>INST</sub>。

此外, 当T0MD T0CS=1 和LCK\_TM0=0 时, PA4 可以是Timer0 外部时钟源EX\_CK10。当T1CS=1 时, PA4 为Timer1 外部时钟源EX\_CK10。当T2CS/T3CS=1 时, PA1 为Timer2/Timer3 外部时钟源EX\_CK11。

CMPOE=1 时, PB3 可作为比较器输出。如果T1CR1[7] PWM1OEN=1, PB3 可以输出PWM1。当BZ1CR[7] BZ1EN=1 时, PB3 可作为Buzzer1 输出。PB3 的输出优先级为比较器输出> PWM1 输出> Buzzer1 输出。

如果T2CR1[7] PWM2OEN=1, PB2 可以输出PWM2。当BZ2CR[7] BZ2EN=1 时, PB2 可作为Buzzer2 输出。PB2 的输出优先级为PWM2>Buzzer2。

当IO配置为输出口, 每个引脚可在配置字设置为一般灌电流 (19mA@VDD=3V), 大灌电流 (28mA@VDD=3V)。如下表所示。

配置字	一般灌电流	大灌电流
PXcurrent	0	1
PXcsc	0	0

表 20 灌电流模式选择 (X=A, B)

### 3.5.1 IO 引脚结构框图

- IO\_SEL:** 设定引脚为输入或输出口。
- WRITE\_EN:** 将数据写入引脚。
- READ\_EN:** 读取引脚状态。
- WUA:** 开启PA口唤醒功能。
- PULLUP\_ENB:** 开启内部上拉电阻。
- PULLDOWN\_EN:** 开启内部下拉电阻。
- VPEN:** 开启为比较器正输入引脚。
- VNEN:** 开启为比较器负输入引脚。
- CMPVP, CMPVN:** 比较器正输入与负输入引脚。
- RD\_TYPE:** 选择读取脚位或数据锁存器。
- SET\_PAIF:** PA口唤醒标志。
- PAEN:** 开启PA口功能。
  - PA0:** 开启CCP P1A功能。
  - PA2:** 开启PWM3/BZ3 功能。
  - PA3:** 开启PWM4/SDO功能。
- PADT:** PA口数据。
  - PA0DT:** CCP P1A 数据。
  - PA2DT:** PWM3/BZ3 数据。
  - PA3DT:** PWM4/SDO 数据。

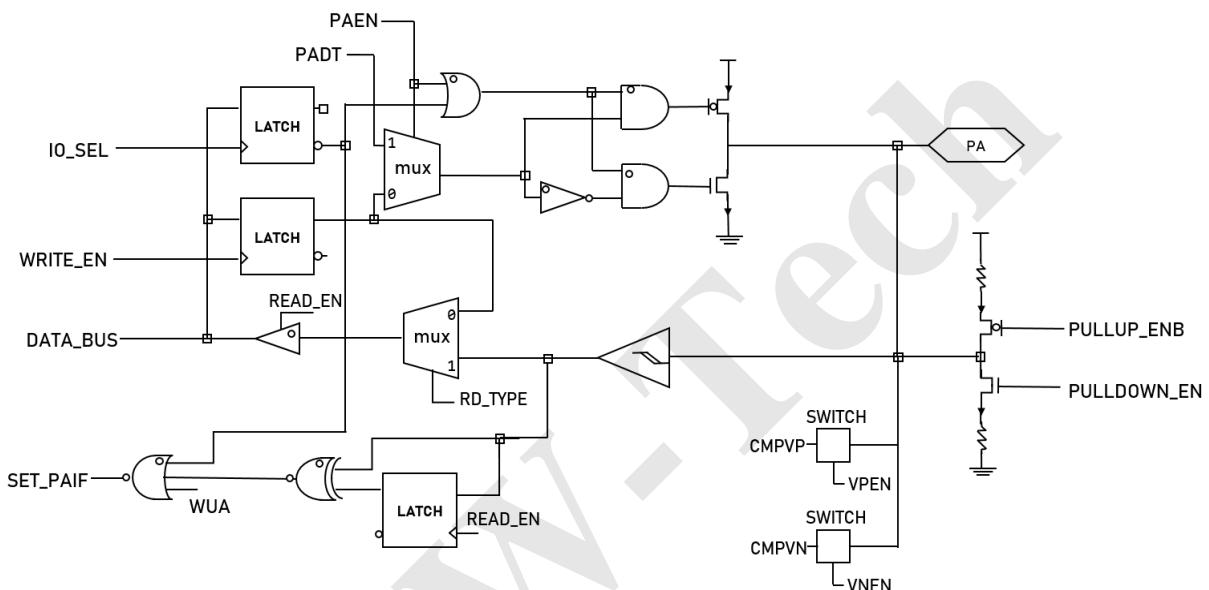


图 5 PA[3:2], PA0 引脚结构框图

IO\_SEL: 设定引脚为输入或输出口。  
WRITE\_EN: 将数据写入引脚。  
READ\_EN: 读取引脚状态。  
WUA: 开启PA口唤醒功能。  
PULLUP\_ENB: 开启内部上拉电阻。  
PULLDOWN\_EN: 开启内部下拉电阻。  
VPEN: 开启为比较器正输入引脚。  
VNEN: 开启为比较器负输入引脚。  
CMPVP, CMPVN: 比较器正输入与负输入引脚。  
RD\_TYPE: 选择读取脚位或数据锁存器。  
SET\_PAIF: PA口唤醒标志。  
EX\_CK1: Timer 2/3 外部时钟输入。

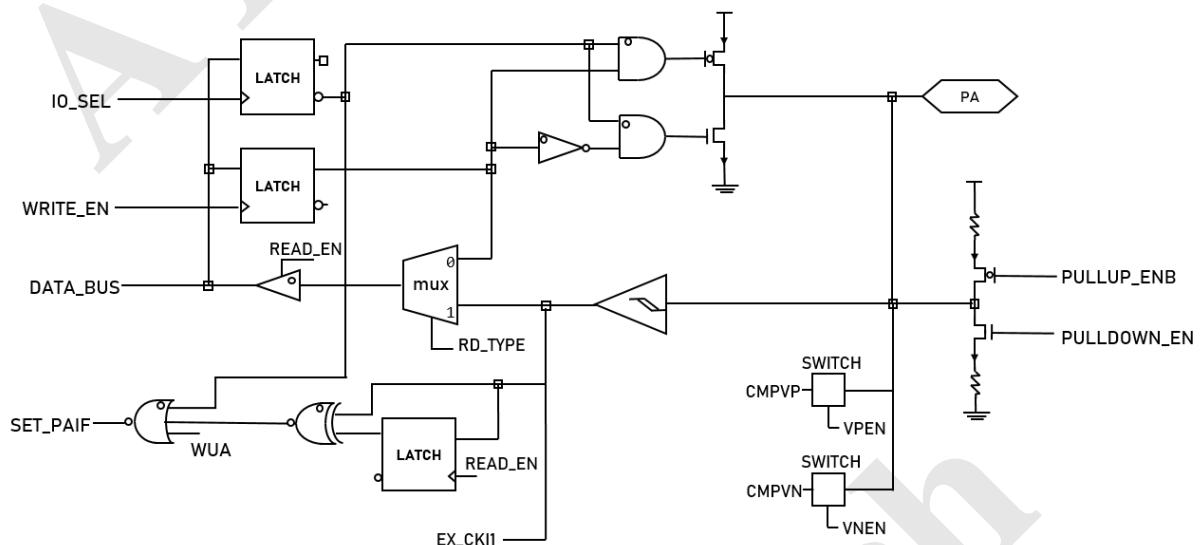


图 6 PA1 引脚结构框图

- IO\_SEL: 设定引脚为输入或输出口。
- WRITE\_EN: 将数据写入引脚。
- READ\_EN: 读取引脚状态。
- WUA: 开启PA口唤醒功能。
- PULLUP\_ENB: 开启内部上拉电阻。
- RD\_TYPE: 选择读取脚位或数据锁存器。
- SET\_PAIF: PA口唤醒标志。
- PAEN: 开启PWM2/CCP P1D功能。
- PADT: PWM2/CCP P1D数据。
- EX\_CKIO: Timer0/1 外部时钟输入。

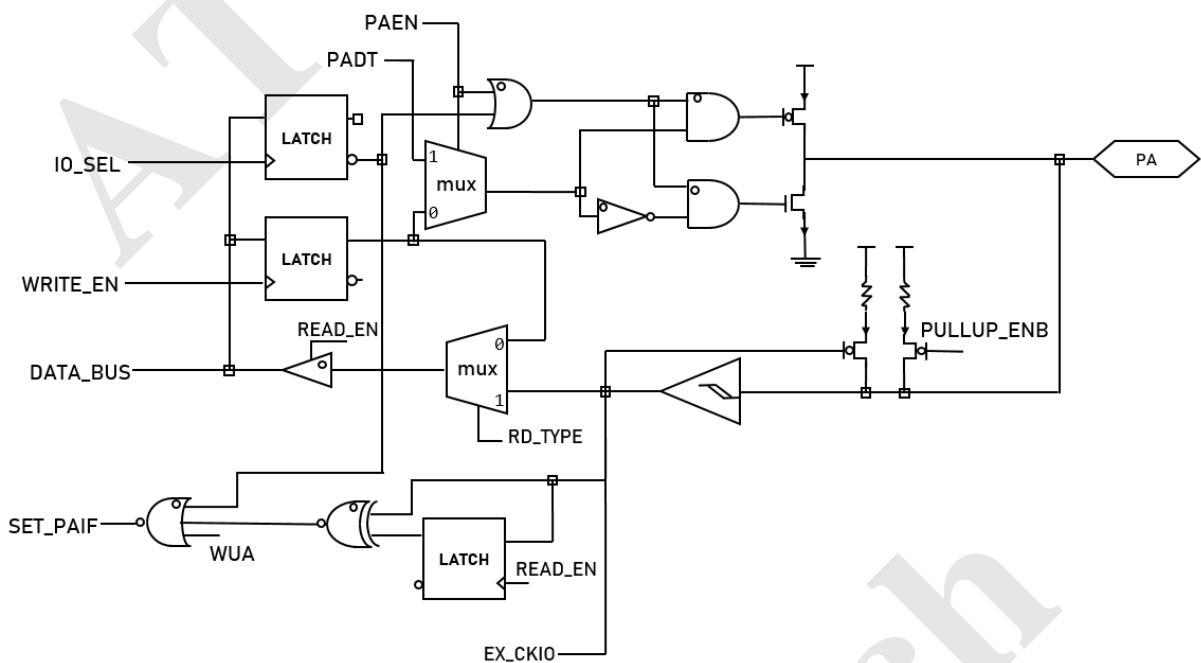


图 7 PA4 引脚结构框图

RSTPAD\_EN: 开启PA5 为复位引脚。

RSTB\_IN: 复位信号输入。

IO\_SEL: 设定引脚为输入或输出口。

WRITE\_EN: 将数据写入引脚。

READ\_EN: 读取引脚状态。

WUA: 开启PA口唤醒功能。

OD\_EN: 开启开漏。

PULLUP\_ENB: 开启内部上拉电阻。

PULLDOWN\_EN: 开启内部下拉电阻。

RD\_TYPE: 选择读取脚位或数据锁存器。

SET\_PAIF: PA口唤醒标志。

PAEN: 开启CCP P1B 功能。

PADT: CCP P1B 数据。

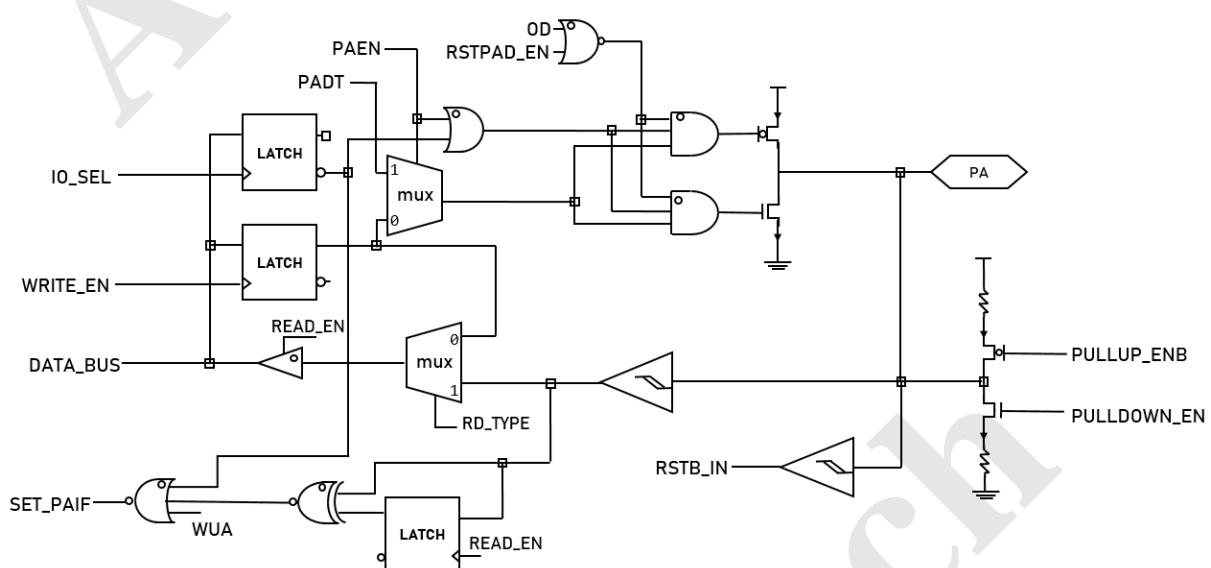


图 8 PA5 引脚结构框图

XTL\_EN: 开启外部晶振模式。

IO\_SEL: 设定引脚为输入或输出口。

WRITE\_EN: 将数据写入引脚。

READ\_EN: 读取引脚状态。

WUA: 开启PA口唤醒功能。

PULLUP\_ENB: 开启内部上拉电阻。

RD\_TYPE: 选择读取脚位或数据锁存器。

SET\_PAIF: PA口唤醒标志。

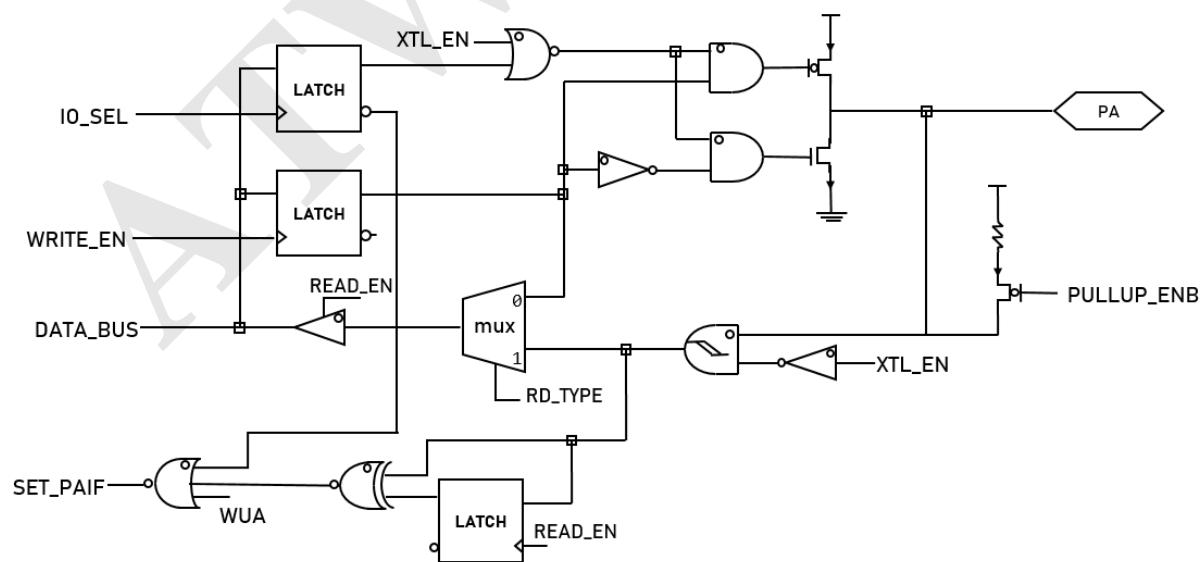


图 9 PA6 引脚结构框图

**XTL\_EN:** 开启外部晶振模式。

**IO\_SEL:** 设定引脚为输入或输出。

**WRITE\_EN:** 将数据写入引脚。

READ\_EN: 读取引脚状态。

**PULLUP\_ENB:** 开启内部上拉电阻。

RD\_TYPE: 选择读取脚位或数据锁存器。

WUA: 开启PA口唤醒功能。

SET PAIF: PA口唤醒标志。

PAEN: 开启PWM4/CCP P1C功能。

PADT: PWM4/CCP P1C数据。

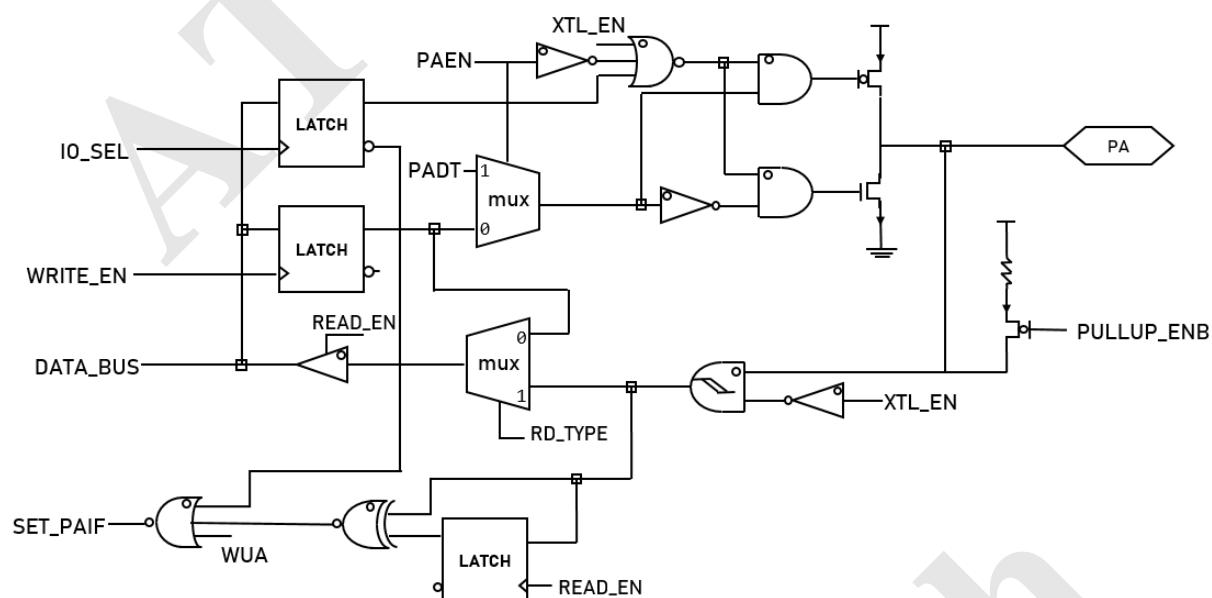


图 10 PA7 引脚结构框图

IO\_SEL: 设定引脚为输入或输出口。  
WRITE\_EN: 将数据写入引脚。  
READ\_EN: 读取引脚状态。  
OD\_EN: 开启开漏。  
PULLUP\_ENB: 开启内部上拉电阻。  
PULLDOWN\_EN: 开启内部下拉电阻。  
RD\_TYPE: 选择读取脚位或数据锁存器。  
EISO: 开启外部中断功能。  
EX\_INTO: 外部中断信号。  
WUB: 开启PB口唤醒功能。  
SET\_PAIF: PB口唤醒标志。  
PBEN: 开启PWM5 功能。  
PBDT: PWM5 数据。

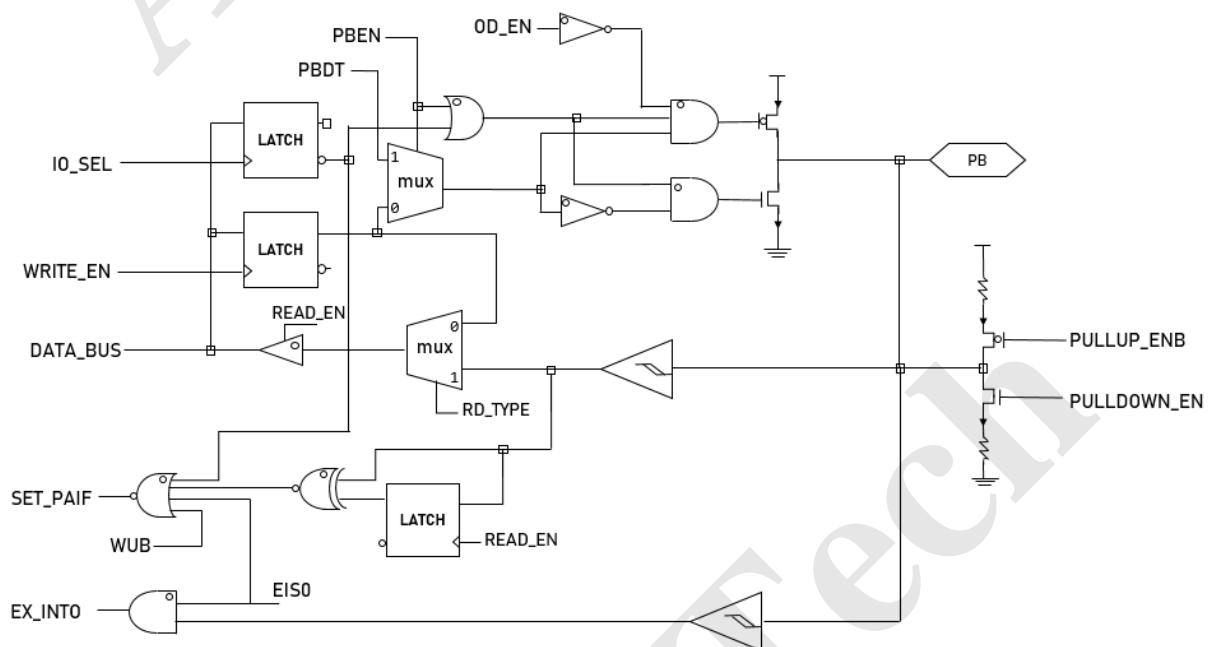


图 10 PB0 引脚结构框图

IO\_SEL: 设定引脚为输入或输出口。  
WRITE\_EN: 将数据写入引脚。  
READ\_EN: 读取引脚状态。  
OD\_EN: 开启开漏。  
PULLUP\_ENB: 开启内部上拉电阻。  
PULLDOWN\_EN: 开启内部下拉电阻。  
RD\_TYPE: 选择读取脚位或数据锁存器。  
EIS1: 开启外部中断功能。  
EX\_INT1: 外部中断信号。  
WUB: 开启PB口唤醒功能。  
SET\_PBI: PB口唤醒标志。  
PBEN: 开启IR功能。  
PBDT: IR数据。

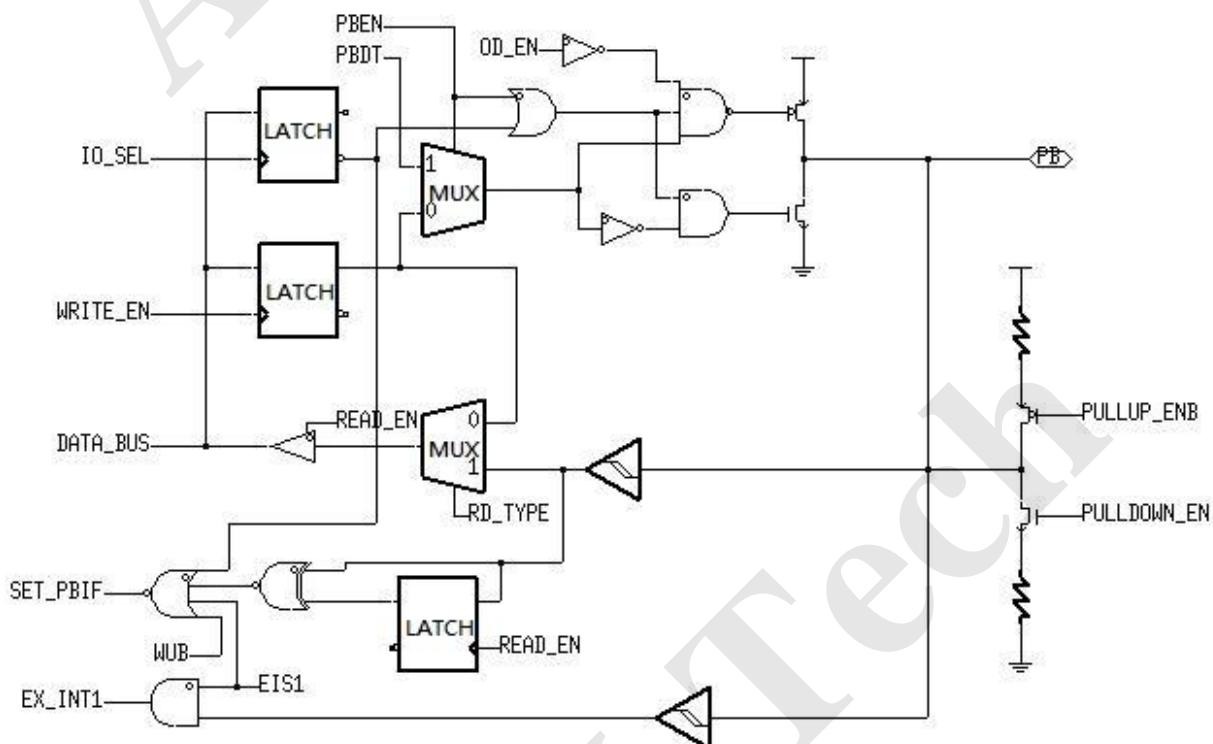


图 11 PB1 引脚结构框图

IO\_SEL: 设定引脚为输入或输出口。  
WRITE\_EN: 将数据写入引脚。  
READ\_EN: 读取引脚状态。  
OD\_EN: 开启开漏。  
PULLUP\_ENB: 开启内部上拉电阻。  
PULLDOWN\_EN: 开启内部下拉电阻。  
RD\_TYPE: 选择读取脚位或数据锁存器。  
WUB: 开启PB口唤醒功能。  
SET\_PBIFF: PB口唤醒标志。  
PBEN: 开启PB口功能。  
PB2: 开启PWM2/BZ2 功能。  
PB3: 开启PWM1/PWM5/SDO/BZ1/CMPO功能。  
PBDT: 数据  
PB2: PWM2/BZ2 数据。  
PB3: PWM1/PWM5/SDO/BZ1/CMPO数据。

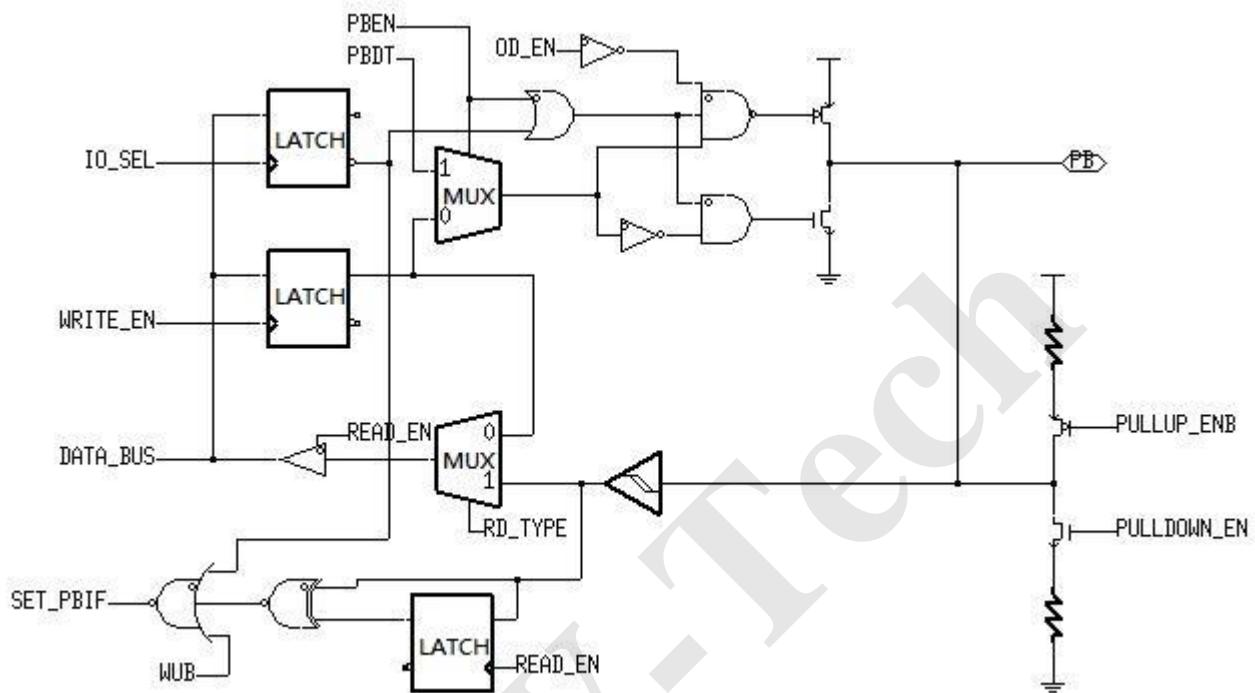


图 12 PB2, PB3 引脚结构框图

- IO\_SEL:** 设定引脚为输入或输出口。
- WRITE\_EN:** 将数据写入引脚。
- READ\_EN:** 读取引脚状态。
- OD\_EN:** 开启开漏。
- PULLUP\_ENB:** 开启内部上拉电阻。
- PULLDOWN\_EN:** 开启内部下拉电阻。
- RD\_TYPE:** 选择读取脚位或数据锁存器。
- WUB:** 开启PB口唤醒功能。
- SET\_PBI:** PB口唤醒标志。

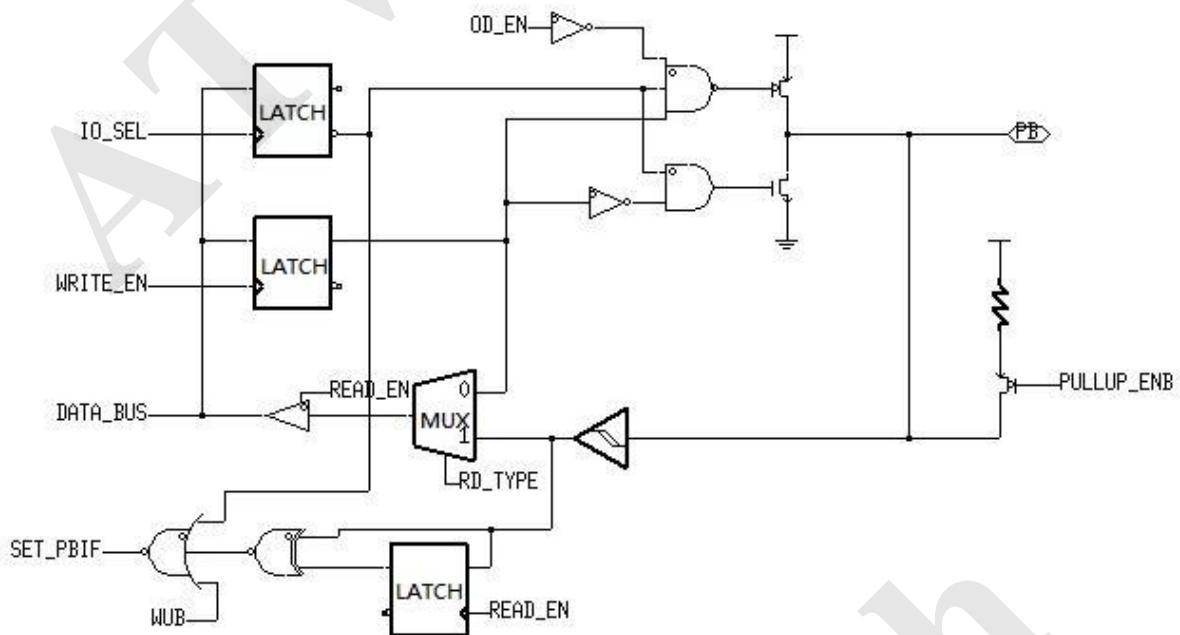


图 13 PB3 引脚结构框图

IO\_SEL: 设定引脚为输入或输出口。  
WRITE\_EN: 将数据写入引脚。  
READ\_EN: 读取引脚状态。  
OD\_EN: 开启开漏。  
PULLUP\_ENB: 开启内部上拉电阻。  
RD\_TYPE: 选择读取脚位或数据锁存器。  
WUB: 开启PB口唤醒功能。  
SET\_PBIFF: PB口唤醒标志。

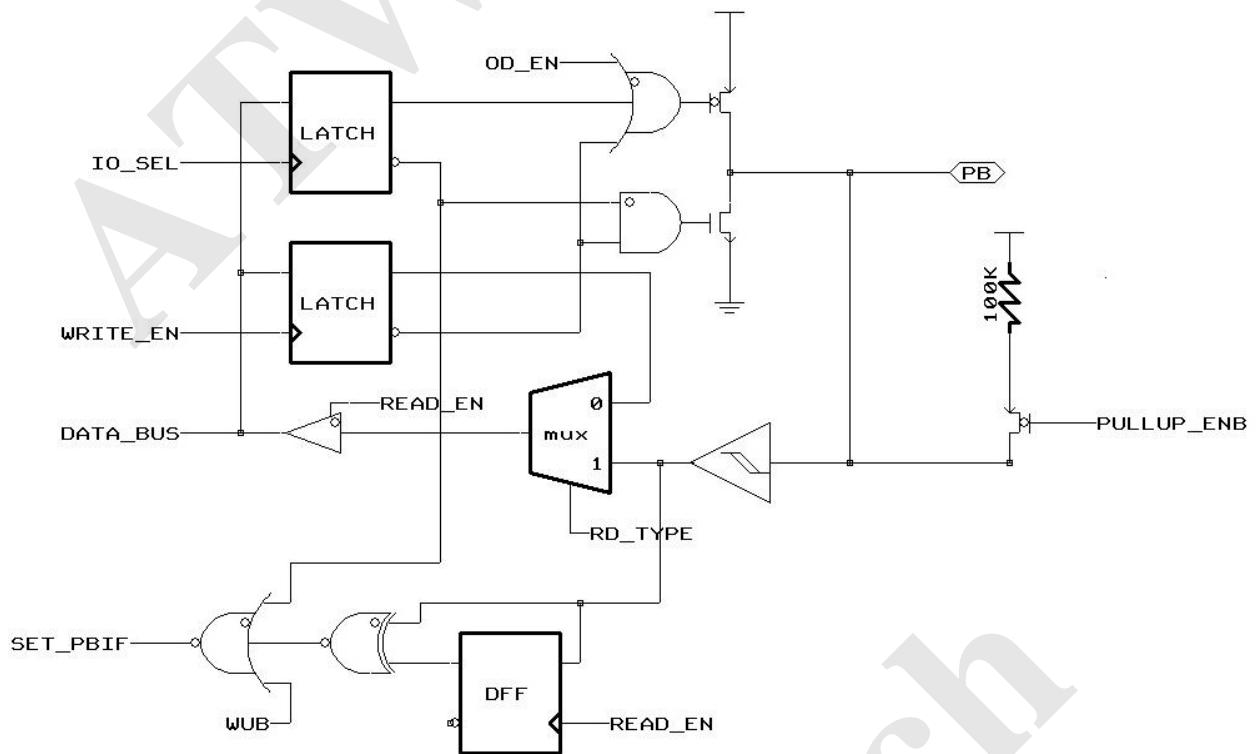


图 14 PB4, 5 引脚结构框图

### 3.6 Timer0

定时器 0 是 8 位上数定时器，由寄存器T0EN (PCON1[0]) 开启/关闭。写入定时器 0 将会设定其初始值，读取定时器 0 时则会显示目前的计数数值。

定时器 0 的时钟源可由寄存器T0CS (T0MD[5]) 与LCK\_TM0 (T0MD[7]) 所决定，可以从指令时钟 $F_{INST}$ 、外部时钟输入脚EX\_CKIO 或低频振荡I\_LRC/E\_LXT中择一。当T0CS为 0，指令时钟会被选择当作定时器 0 时钟源。当T0CS为 1 且LCK\_TM0 为 0，EX\_CKIO 会被当作定时器 0 时钟源。当T0CS是 1 且LCK\_TM0 为 1，会选择低频振荡I\_LRC/E\_LXT当作定时器 0 时钟源。汇总成表格如下。(也请参考图 15)

定时器 0 时钟源	T0CS	LCKTM0	定时器 0 来源	低频振荡
Instruction clock	0	X	X	X
EX_CKIO	1	0	X	X
		X	0	
E_LXT	1	1	1	1
I_LRC	1	1	1	0

表 21 定时器 0 时钟源摘要

寄存器T0CE (T0MD[4]) 可决定EX\_CKIO 或I\_LRC/E\_LXT的时钟触发沿选择。当T0CE是 1，EX\_CKIO 或I\_LRC/E\_LXT的下降沿将让定时器 0 计数加一。当T0CE是 0，EX\_CKIO 或I\_LRC/E\_LXT的上升沿将让定时器 0 计数加一。当使用I\_LRC/E\_LXT作为定时器 0 的时钟源时，建议使用预分频器 0 (见下文)，并将比率设置为 4 以上，否则可能会出现漏计数。

在为定时器 0 提供时钟源之前，如果寄存器位PS0WDT (T0MD[3]) 清零，则可以被预分频器 0 分频。当用指令将 0 写入PS0WDT时，将预分频器 0 赋值给定时器 0，执行完该指令后，将清除预分频器 0。寄存器PS0SEL[2:0]决定预分频器 0 的预分频比，其数值从 1:2 到 1:256。

当定时器 0 上溢时，寄存器TOIF(INTF[0])将会设定为 1，以表明定时器 0 发生上溢中断。如果寄存器TOIE(INTE[0])与GIE都设定为 1，会发生中断的请求并执行中断服务程序。直到程序写入 0 到TOIF，TOIF才会被清除。

定时器 0 与WDT的结构框图如下图：

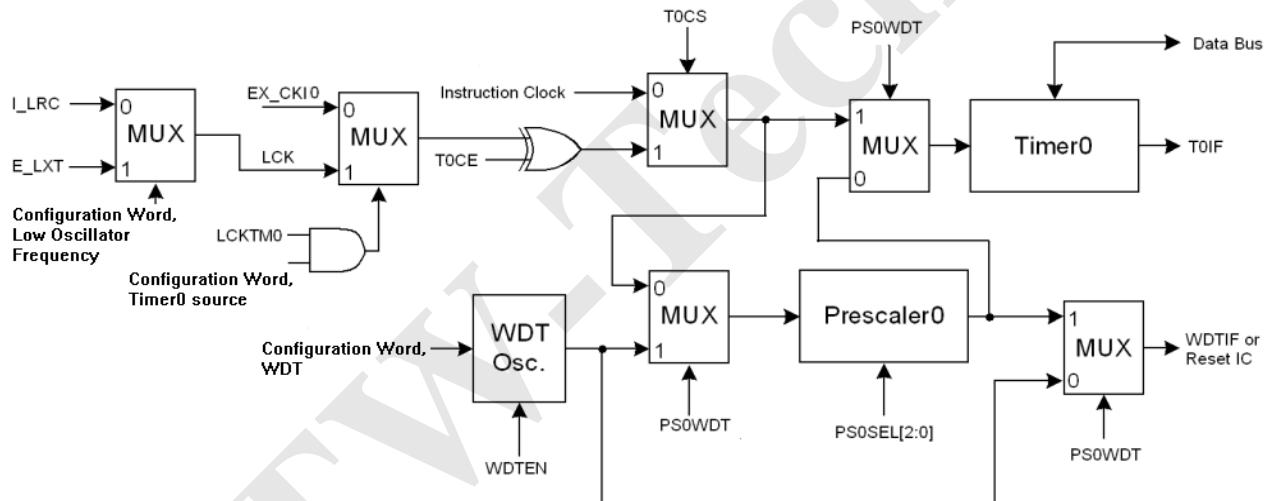


图 15 定时器 0 与 WDT 结构框图

### 3.7 Timer1 / PWM1 / Buzzer1

定时器 1 是具有预分频器 1 的 10 位下数定时器，其预分频比是可编程的。定时器 1 的输出可以被用于产生 PWM1 输出与蜂鸣器 1 输出。定时器 1 内置自动加载功能，定时器 1 加载寄存器用双缓冲区存储加载数据。当用户写定时器 1 重载寄存器时，先写定时器 1 MSB 2 位 (TMRH[5:4])，再写 TMR1，当 T1EN=1 时发生定时器 1 下溢后，定时器 1 重载寄存器更新为定时器 1 计数器。如果 T1EN=0，写 TMR1 后立即将定时器 1 重载寄存器更新为定时器 1 计数器。读取定时器 1 会显示定时器 1 目前计数数值的内容。

定时器 1 的结构框图如下图所示：

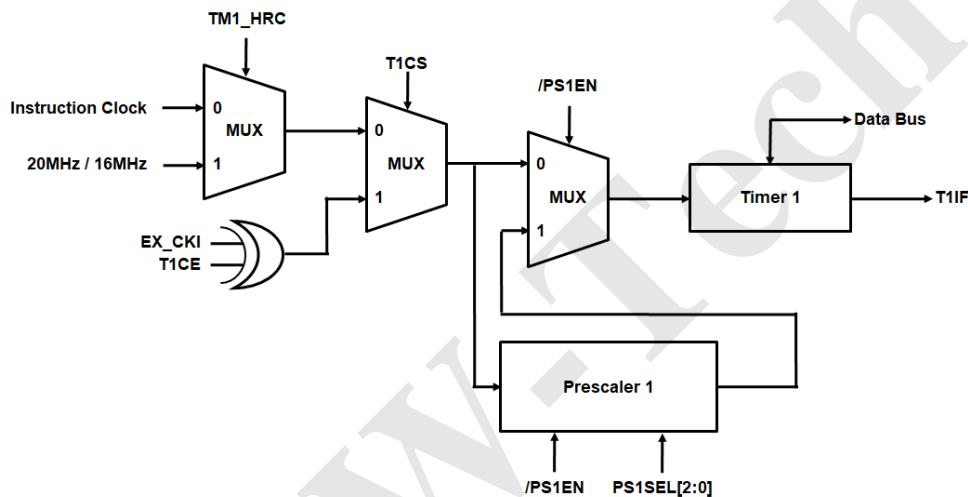


图 16 定时器 1 结构框图

定时器 1 的操作可以由寄存器 T1EN (T1CR1[0]) 开启或关闭。开启定时器 1 后，寄存器 T1CS (T1CR2[5]) 可决定定时钟源是指令时钟  $F_{INST}$  或外部时钟 EX\_CKIO。当 T1CS 为 1，EX\_CKIO 被选择当作时钟源。当 T1CS 为 0，指令时钟会被选择当作时钟源。当 EX\_CKIO 被选取，寄存器控制位 T1CE (T1CR2[4]) 可决定 EX\_CKIO 的时钟触发沿

使定时器 1 减一。当 T1CE 是 1 时, EX\_CK10 的下降沿将让定时器 1 计数减一。当 T1CE 是 0 时, EX\_CK10 的上升沿将让定时器 1 计数减一。所选时钟源在应用于定时器 1 之前, 可以通过预分频器 1 进一步分频。寄存器/PS1EN (T1CR2[3]) 为 0, 可开启预分频器 1。寄存器PS1SEL[2:0] (T1CR2[2:0]) 可以决定其预分频比从 1:2 到 1:256。预分频器 1 的目前数值可以由读取寄存器PS1CV取得。

定时器 1 提供两种计数模式: 单次计数模式与连续计数模式。当寄存器T1OS (T1CR1[2]) 为 1 时, 选择单次计数模式。从寄存器TMR1[9:0]到 0x00, 定时器 1 将倒数一次, 即发生了下溢。当寄存器T1OS (T1CR1[2]) 为 0 时, 选择连续计数模式。当发生下溢时, 有两个选择开始下一个下数计数, 这是由寄存器位T1RL (T1CR1[1]) 决定的。当T1RL为 1, 定时器 1 从寄存器TMR1[9:0]重新载入数值作为初始值并继续下数。当T1RL为 0, 定时器 1 以 0x3FF 作为初始值并继续下数。

当定时器 1 下溢, 寄存器T1IF (INTF[3]) 会被设定为 1, 表明定时器 1 发生下溢中断。如果寄存器T1IE (INTE[3]) 与GIE同时设定为 1, 会发生中断请求且执行中断服务程序。直到程序写入 0 到T1IF, T1IF才会被清除。

定时器 1 时序图如下图所示:

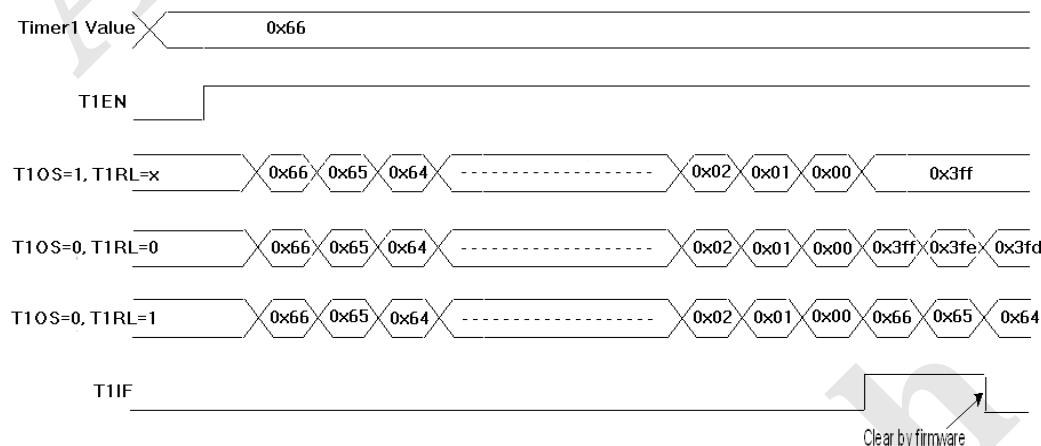


图 17 定时器 1 时序图

当寄存器PWM1OEN (T1CR1[7]) 设定为 1, PB3 为 PWM1 输出。当PWM1OEN为 1, PB3 会自动成为输出脚。PWM1 输出的有效状态是由寄存器PWM1OAL (T1CR1[6]) 决定。当PWM1OAL为 1, PWM1 为低电平有效输出; PWM1OAL为 0, PWM1 为高电平有效输出。

PWM1 的占空比与帧率皆可编程的。占空比是由寄存器TMRH[1:0]和PWM1DUTY[7:0]决定。当PWM1DUTY为 0, PWM1 输出无效。当PWM1DUTY为 0x3FF, PWM1 将输出 1023/1024 的占空比。帧率是由TMRH[5:4] + TMR1[7:0] 初始值所决定。因此, PWM1DUTY数值必须小于或等于TMRH[5:4] + TMR1[7:0]。当用户写PWM1DUTY时, 先写PWM1DUTY[9:8] MSB 2 位 (TMRH[1:0]), 再写PWM1DUTY[7:0], 定时器 1 下溢后, PWM1DUTY寄存器将会被更新。

PWM1 的结构框图如下:

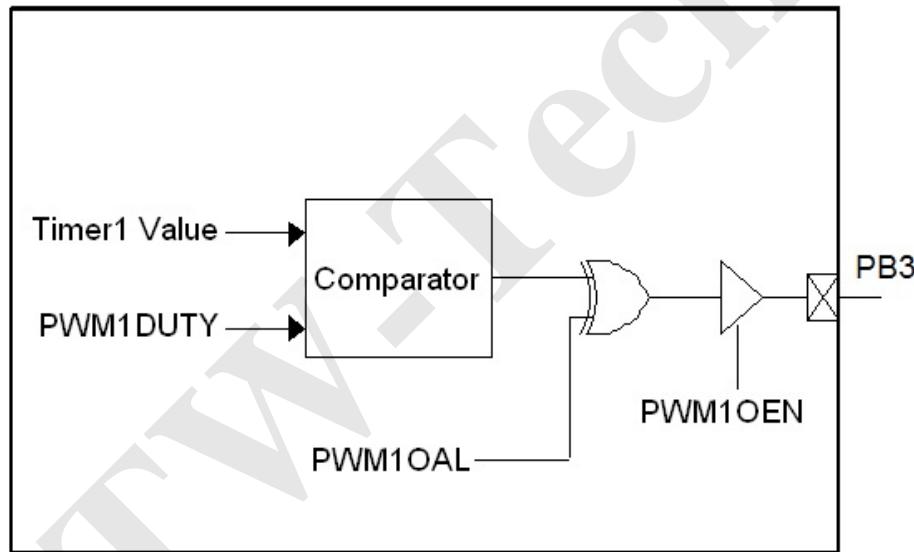


图 18 PWM1 结构框图

当寄存器BZ1EN (BZ1CR1[7]) 设定成 1 且使能配置字, PB3 为蜂鸣器 1 输出。当BZ1EN设定为 1, PB3 会自动成为输出脚。BZ1 的频率是由寄存器BZ1FSEL[3:0] (BZ1CR[3:0]) 决定, 可以选择从定时器 1 输出或预分频器 1 输出。当BZ1FSEL[3]为 0, 预分频器 1 输出被选择来产生BZ1 输出。当BZ1FSEL[3]为 1, 定时器 1 输出被选来产生BZ1 输出。为了产生各种频率, 预分频比可以从 1:2 到 1:256。蜂鸣器 1 结构框图如下所示:

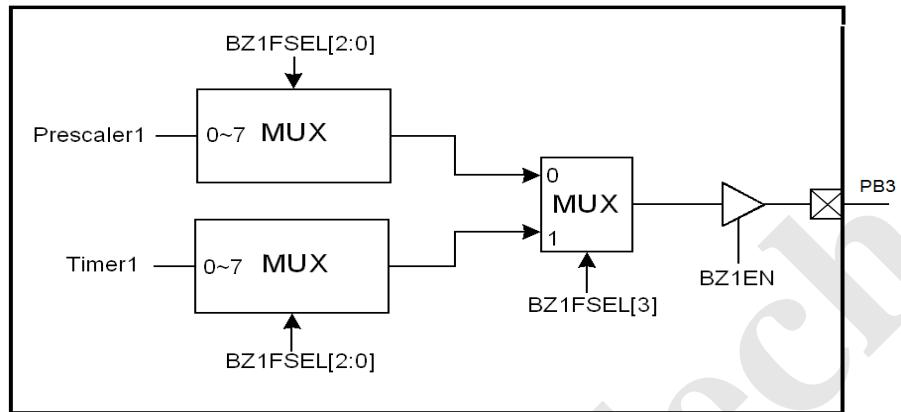


图 19 蜂鸣器 1 结构框图

**注意: PB3 复用引脚输出优先级为 PWM1 输出 > 蜂鸣器 1 输出。**

### 3.8 Timer2 / PWM2 / Buzzer2

定时器 2 是具有预分频器 2 的 10 位下数定时器, 其预分频比是可编程的。定时器 2 的输出可以被用于产生PWM2 输出与蜂鸣器 2 输出。定时器 2 内置自动加载功能, 定时器 2 加载寄存器用双缓冲区存储加载数据。当用户写定时器 2 重载寄存器时, 先写定时器 2 MSB 2 位 (TMRH[7:6]), 再写TMR2, 当T2EN=1 时发生定时器 2 下溢后, 定时器 2 重载寄存器更新为定时器 2 计数器。如果T2EN=0, 写TMR2 后立即将定时器 2 重载寄存器更新为定时器 2 计数器。读取定时器 2 会显示定时器 2 目前计数数值的内容。

定时器 2 的结构框图如下图所示:

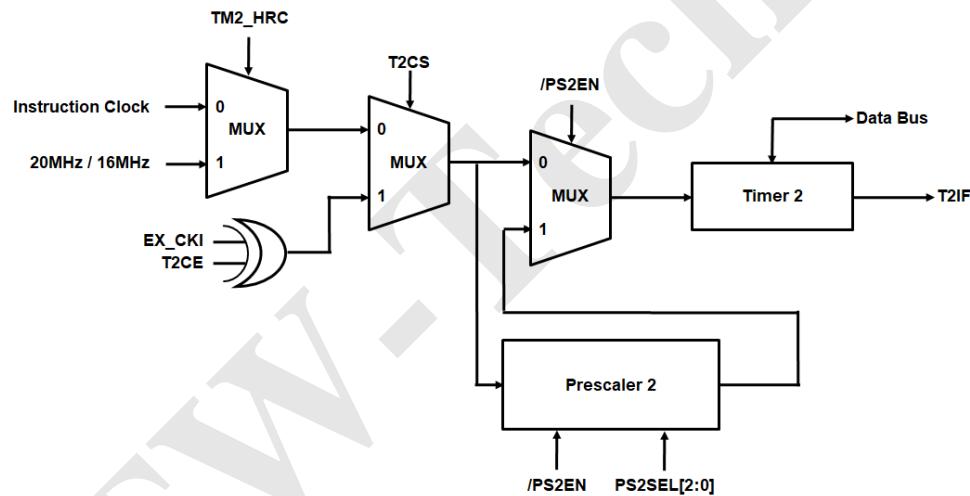


图 20 定时器 2 结构框图

定时器 2 的操作可以由寄存器 T2EN (T2CR1[0]) 开启或关闭。开启定时器 2 后，寄存器 T2CS (T2CR2[5]) 可决定定时钟源是指令时钟  $F_{INST}$  或外部时钟 EX\_CK1。当 T2CS 为 1，EX\_CK1 被选择当作时钟源。当 T2CS 为 0，指令时钟会被选择当作时钟源。当 EX\_CK1 被选取，寄存器控制位 T2CE (T2CR2[4]) 可决定 EX\_CK1 的时钟触发沿使定时器 2 减一。当 T2CE 是 1 时，EX\_CK1 的下降沿将让定时器 2 计数减一。当 T2CE 是 0 时，EX\_CK1 的上升沿将让定时器 2 计数减一。

所选时钟源在应用于定时器 2 之前，可以通过预分频器 2 进一步分频。寄存器 /PS2EN (T2CR2[3]) 为 0，可开启预分频器 2。寄存器 PS2SEL[2:0] (T2CR2[2:0]) 可以决定其预分频比从 1:2 到 1:256。预分频器 2 的目前数值可以由读取寄存器 PS2CV 取得。

定时器 2 提供两种计数模式：单次计数模式与连续计数模式。当寄存器 T2OS (T2CR1[2]) 为 1 时，选择单次计数模式。从寄存器 TMR2[9:0] 到 0x00，定时器 2 将下数一次，即发生了下溢。当寄存器 T2OS (T2CR1[2]) 为 0 时，选择连续计数模式。当发生下溢时，有两个选择开始下一个下数计数，这是由寄存器位 T2RL (T2CR1[1]) 决定的。当 T2RL 为 1，定时器 2 从寄存器 TMR2[9:0] 重新载入数值作为初始值并继续下数。当 T2RL 为 0，定时器 2 以 0x3FF 作为初始值并继续下数。

当定时器 2 下溢，寄存器 T2IF (INTF[5]) 会被设定为 1，表明定时器 2 发生下溢中断。如果寄存器 T2IE (INTE[5]) 与 GIE 同时设定为 1，会发生中断请求且执行中断服务程序。直到程序写入 0 到 T2IF，T2IF 才会被清除。

定时器 2 时序图如下图所示：

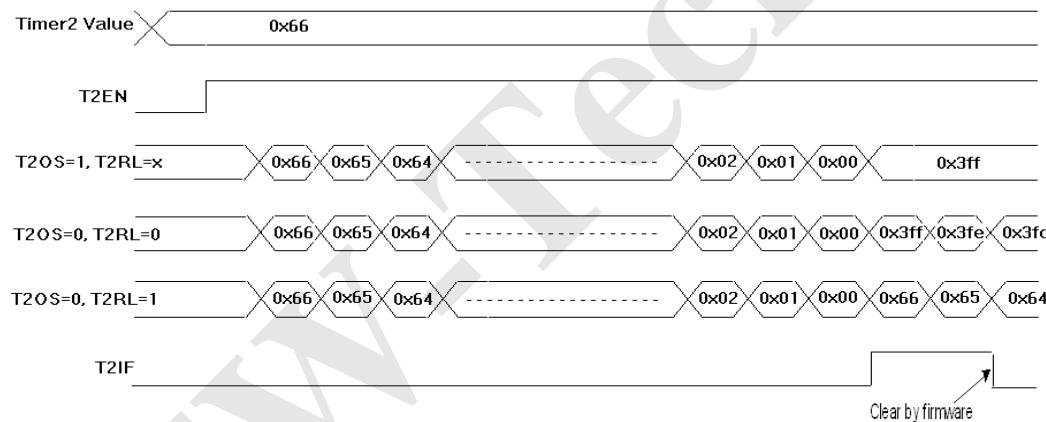


图 21 定时器 2 时序图

当寄存器 PWM2OEN (T2CR1[7]) 设定为 1, PA4 或 PB2 为 PWM2 输出。当 PWM2OEN 为 1, PA4 或 PB2 会自动成为输出脚。PWM2 输出的有效状态是由寄存器 PWM2OAL (T2CR1[6]) 决定。当 PWM2OAL 为 1, PWM2 为低电平有效输出; PWM2OAL 为 0, PWM2 为高电平有效输出。

PWM2 的占空比与帧率皆可编程的。占空比是由寄存器 TMRH[3:2] 和 PWM2DUTY[7:0] 决定。当 PWM2DUTY 为 0, PWM2 输出无效。当 PWM2DUTY 为 0x3FF, PWM2 将输出 1023/1024 的占空比。帧率是由 TMRH[7:6] + TMR2[7:0] 初始值所决定。因此, PWM2DUTY 数值必须小于或等于 TMR2[9:0]。当用户写 PWM2DUTY 时, 先写 PWM2DUTY[9:8] MSB 2 位 (TMRH[3:2]), 再写 PWM2DUTY[7:0], 定时器 2 下溢后, PWM2DUTY 寄存器将会被更新。

PWM2 的结构框图如下:

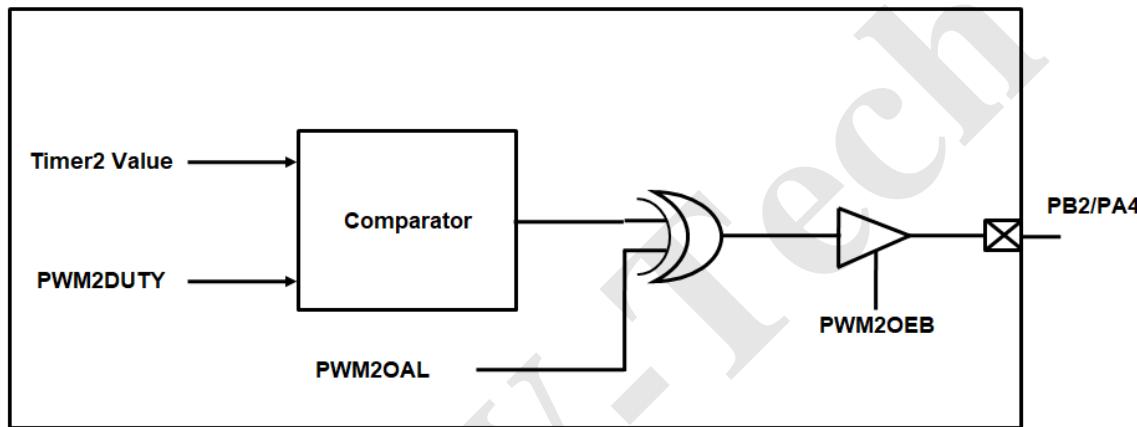


图 22 PWM2 结构框图

当寄存器 BZ2EN (BZ2CR1[7]) 设定成 1 且使能配置字, PB2 为蜂鸣器 2 输出。而且, PB2 会自动成为输出脚。BZ2 的频率是由寄存器 BZ2FSEL[3:0] (BZ2CR[3:0]) 决定, 可以选择从定时器 2 输出或预分频器 2 输出。当 BZ2FSEL[3] 为 0, 预分频器 2 输出被选择来产生 BZ2 输出。当 BZ2FSEL[3] 为 1, 定时器 2 输出被选来产生 BZ2 输出。为了产生各种频率, 预分频比可以从 1:2 到 1:256。蜂鸣器 2 结构框图如下所示:

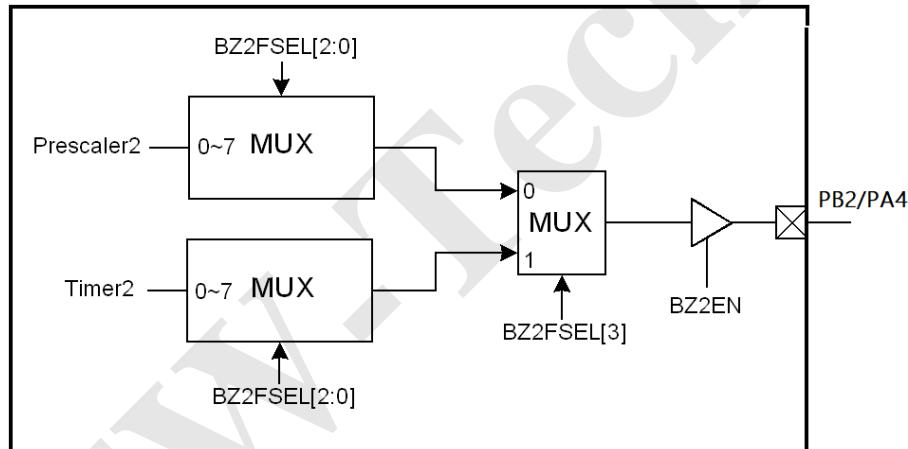


图 23 蜂鸣器 2 结构框图

### 3.9 Timer3 / PWM3 / Buzzer3 / PWM4 / PWM5

定时器 3 是具有预分频器 3 的 10 位下数定时器，其预分频比是可编程的。定时器 3 的输出可以被用于产生 PWM3 输出 3。定时器 3 内置自动加载功能，定时器 3 加载寄存器用双缓冲区存储加载数据。当用户写定时器 3 重载寄存器时，先写定时器 3 MSB 2 位 (TM3RH[5:4])，再写 TMR3，当 T3EN=1 时发生定时器 3 下溢后，定时器 3 重载寄存器更新为定时器 3 计数器。如果 T3EN=0，写 TMR3 后立即将定时器 3 重载寄存器更新为定时器 3 计数器。读取定时器 3 会显示定时器 3 目前计数数值的内容。

定时器 3 的结构框图如下图所示：

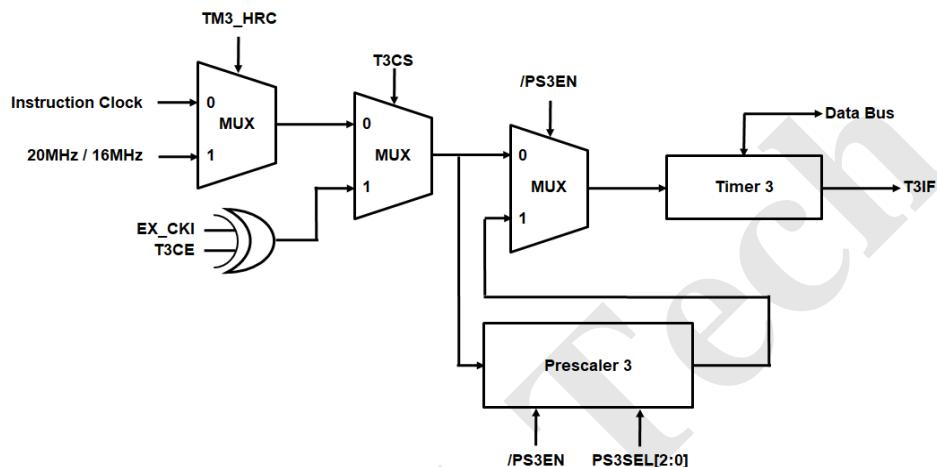


图 24 定时器 3 结构框图

定时器 3 的操作可以由寄存器 T3EN (T3CR1[0]) 开启或关闭。开启定时器 3 后，寄存器 T3CS (T3CR2[5]) 可决定定时钟源是指令时钟  $F_{INST}$  或外部时钟 EX\_CK1。当 T3CS 为 1，EX\_CK1 被选择当作时钟源。当 T3CS 为 0，指令时钟会被选择当作时钟源。当 EX\_CK1 被选取，寄存器控制位 T3CE (T3CR2[4]) 可决定 EX\_CK1 的时钟触发沿使定时器 3 减一。当 T3CE 是 1 时，EX\_CK1 的下降沿将让定时器 3 计数减一。当 T3CE 是 0 时，EX\_CK1 的上升沿将让定时器 3 计数减一。

所选时钟源在应用于定时器 3 之前，可以通过预分频器 3 进一步分频。寄存器/PS3EN (T3CR2[3]) 为 0，可开启预分频器 3。寄存器PS3SEL[2:0] (T3CR2[2:0]) 可以决定其预分频比从 1:2 到 1:256。预分频器 3 的目前数值可以由读取寄存器PS3CV取得。

定时器 3 提供两种计数模式：单次计数模式与连续计数模式。当寄存器T3OS (T3CR1[2]) 为 1 时，选择单次计数模式。从寄存器TMR3[9:0]到 0x00，定时器 3 将下数一次，即发生了下溢。当寄存器T3OS (T3CR1[2]) 为 0 时，选择连续计数模式。当发生下溢时，有两个选择开始下一个下数计数，这是由寄存器位T3RL (T3CR1[1]) 决定的。当T3RL为 1，定时器 3 从寄存器TMR3[9:0]重新载入数值作为初始值并继续下数。当T3RL为 0，定时器 3 以 0x3FF 作为初始值并继续下数。

当定时器 3 下溢，寄存器T3IF (INTE2[4]) 会被设定为 1，表明定时器 3 发生下溢中断。如果寄存器T3IE (INTE2[0]) 与GIE同时设定为 1，会发生中断请求且执行中断服务程序。直到程序写入 0 到T3IF，T3IF才会被清除。

定时器 3 时序图如下图所示：

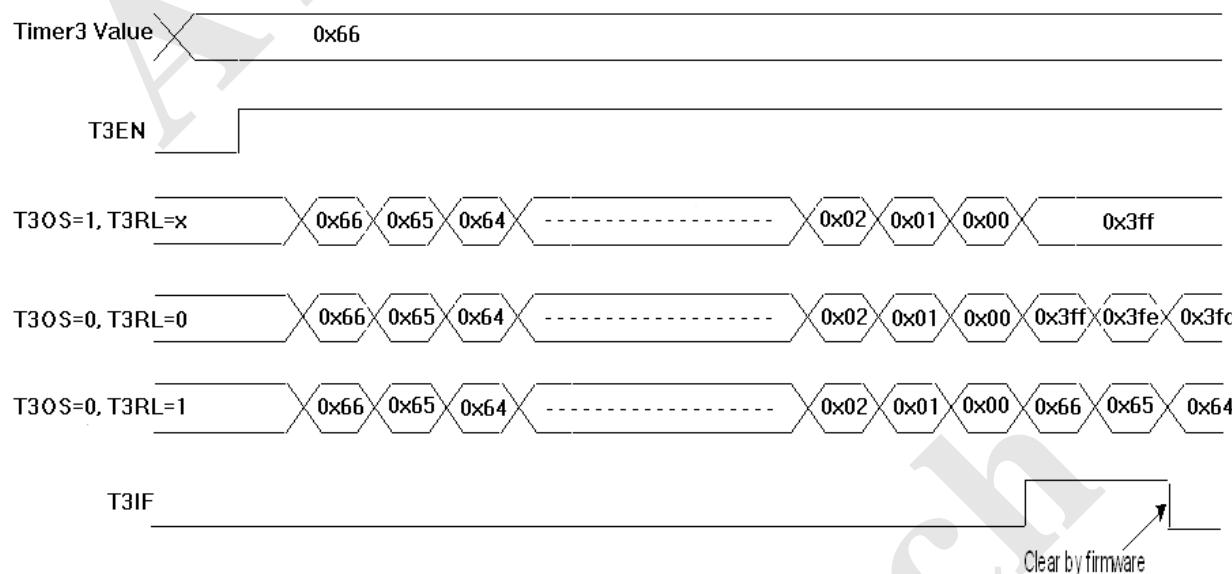


图 25 定时器 3 结构框图

当寄存器PWM3OEN (T3CR1[7]) 设定为 1，PA2 为 PWM3 输出。并且，PA2 会自动成为输出脚。PWM3 输出的有效状态是由寄存器PWM3OAL (T3CR1[6]) 决定。当PWM3OAL为 1，PWM3 为低电平有效输出；PWM3OAL 为 0，PWM3 为高电平有效输出。

PWM3 的占空比与帧率皆可编程的。占空比是由寄存器TM3RH[1:0]和PWM3DUTY[7:0]决定。当PWM3DUTY为 0，PWM3 输出无效。当PWM3DUTY为 0x3FF，PWM3 将输出 1023/1024 的占空比。帧率是由TM3RH[5:4]+TMR3[7:0]初始值所决定。因此，PWM3DUTY 数值必须小于或等于 TMR3[9:0]。当用户写 PWM3DUTY 时，先写 PWM3DUTY[9:8] MSB 2 位 (TM3RH[1:0])，再写 PWM3DUTY[7:0]，定时器 3 下溢后，PWM3DUTY 寄存器将会被更新。

PWM3 的结构框图如下：

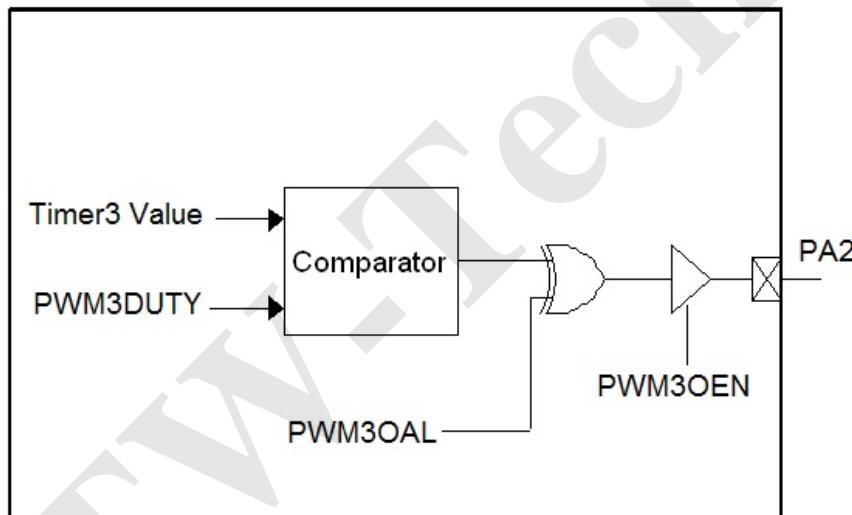


图 26 PWM3 结构框图

当寄存器BZ3EN (BZ3CR1[7]) 设定成 1 且使能配置字, PA2 为蜂鸣器 3 输出。而且, PA2 会自动成为输出脚。BZ3 的频率是由寄存器BZ3FSEL[3:0] (BZ3CR[3:0]) 决定, 可以选择从定时器 3 输出或预分频器 3 输出。当 BZ3FSEL[3]为 0, 预分频器 3 输出被选择来产生BZ3 输出。当BZ3FSEL[3]为 1, 定时器 3 输出被选来产生BZ3 输出。为了产生各种频率, 预分频比可以从 1:2 到 1:256。蜂鸣器 3 结构框图如下所示:

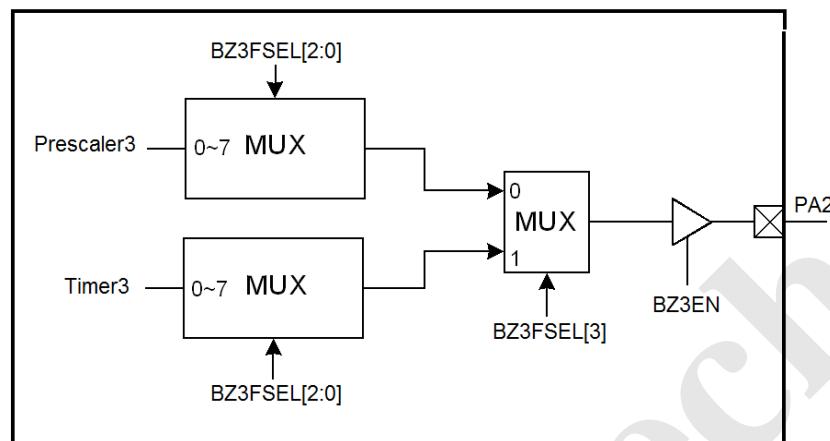


图 27 蜂鸣器 3 结构框图

当寄存器PWM4OEN (P4CR1[7]) 设定为 1, PA3 或PA7 为PWM4 输出。并且, PA3 或PA7 会自动成为输出脚。PWM4 输出的有效状态是由寄存器PWM4OAL ((P4CR1[6])) 决定。当PWM4OAL为 1, PWM4 为低电平有效输出; PWM4OAL为 0, PWM4 为高电平有效输出。

PWM4 的占空比与帧率皆可编程的。占空比是由寄存器TM3RH[3:2]和PWM4DUTY[7:0]决定。当PWM4DUTY为 0, PWM4 输出无效。当PWM4DUTY为 0x3FF, PWM4 将输出 1023/1024 的占空比。帧率是由TM3RH[5:4]+TMR3[7:0]初始值所决定。因此, PWM4DUTY 数值必须小于或等于 TMR3[9:0]。当用户写 PWM4DUTY 时, 先写 PWM4DUTY[9:8] MSB 2 位 (TM3RH[3:2]), 再写PWM4DUTY[7:0], 定时器 3 下溢后, PWM4DUTY寄存器将会被更新。

PWM4 的结构框图如下:

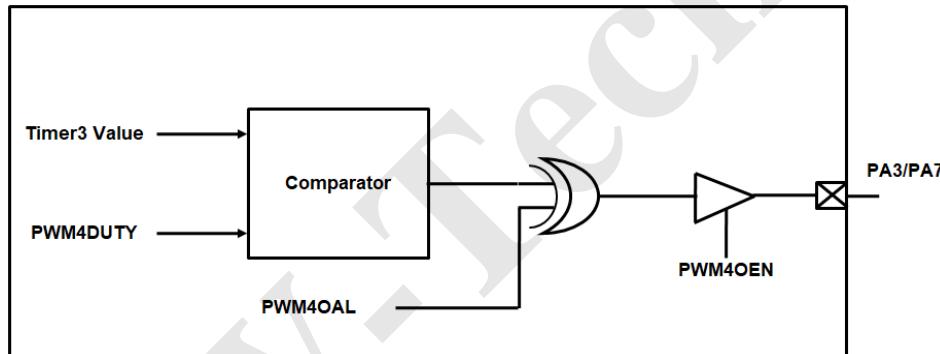


图 28 PWM4 结构框图

当寄存器 PWM5OEN (P5CR1[7]) 设定为 1, PB0 或 PB3 为 PWM5 输出。并且, PB0 或 PB3 会自动成为输出脚。 PWM5 输出的有效状态是由寄存器 PWM5OAL (P5CR1[6]) 决定。当 PWM5OAL 为 1, PWM5 为低电平有效输出; PWM5OAL 为 0, PWM5 为高电平有效输出。

PWM5 的占空比与帧率皆可编程的。占空比是由寄存器 TM3RH[3:2] 和 PWM5DUTY[7:0] 决定。当 PWM5DUTY 为 0, PWM5 输出无效。当 PWM5DUTY 为 0x3FF, PWM5 将输出 1023/1024 的占空比。帧率是由 TM3RH[5:4]+TMR3[7:0] 初始值所决定。因此, PWM5DUTY 数值必须小于或等于 TMR3[9:0]。当用户写 PWM5DUTY 时, 先写 PWM5DUTY[9:8] MSB 2 位 (PWM5RH[1:0]), 再写 PWM5DUTY[7:0], 定时器 3 下溢后, PWM5DUTY 寄存器将被更新。

PWM5 的结构框图如下:

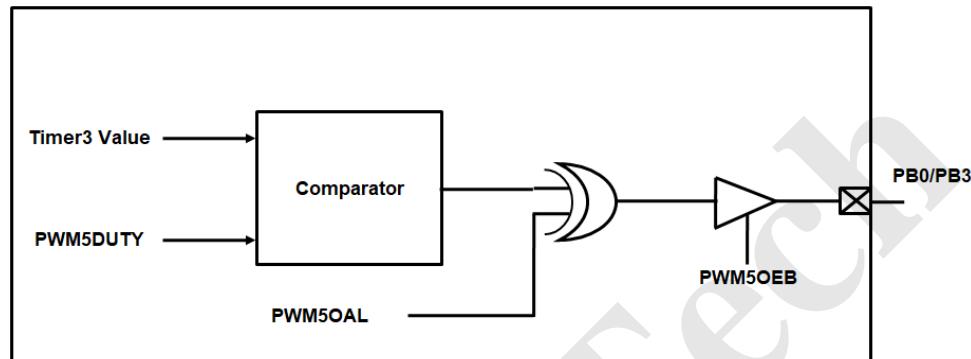


图 29 PWM5 结构框图

### 3.10 CCP模式

CCP (捕捉/比较/脉宽调变) 寄存器 (CCPR) 由 2 个 8-bit 寄存器组成。CCPRL (低字节) 和 CCPRH (高字节)。CCPCON 和 PWMDB 寄存器控制 CCP 操作。捕捉和比较模式使用 16 位定时器, PWM 模式使用 10 位定时器。CCP 定时器和寄存器利用现有的定时器和寄存器。以下表格显示的是与 CCP 共享的定时器和寄存器。注意当 AT8B62F1 处于 CCP 模式时, 相关定时器/PWM 功能被禁用。

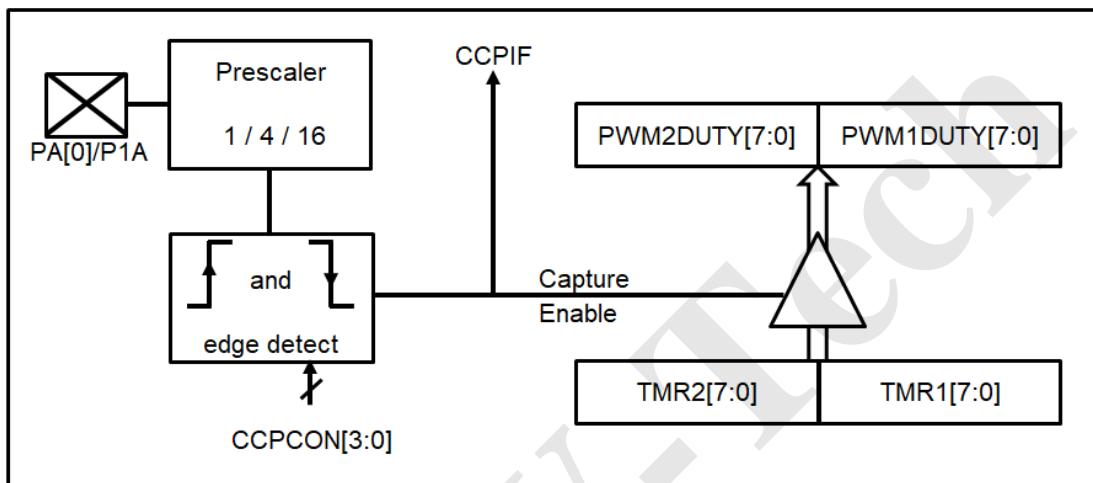
CCP模式	CCP寄存器/定时器	共享的定时器	共享的寄存器
捕捉/比较	CCPL	-	PWM1DUTY[7:0]
捕捉/比较	CCPH	-	PWM2DUTY[7:0]
PWM	PWMDUTY	-	PWM2DUTY[9:0]
PWM	PWMDB	-	-
捕捉/比较	捕捉/比较定时器	Timer2[7:0] (MSB) + Timer1[7:0] (LSB)	-
PWM	PWM timer	Timer2[9:0]	-

### 3.10.1 捕捉模式

在捕捉模式下, 当P1A (PA0)发生一个事件时, CCPRH:CCPRL ( PWM2DUTY[7:0]:PWM1DUTY[7:0] ) 捕捉16位定时器寄存器, 该事件定义为:

- 每一次的下降沿
- 每一次的上升沿
- 每 4 次的上升沿
- 每 16 次的上升沿

捕捉模式的结构框图如下:



一次事件由CCPM[3:0]控制。当捕捉一次时, 设置中断请求标志位CCPIF。

在捕捉模式下, P1A(PA0)应该被配置为输入脚, 如果它是输出脚, 要写入IO口来创造一个捕捉条件。

在捕捉模式下, 捕捉定时器必须在CPU时钟同步模式下运行, 否则捕捉操作不能进行。

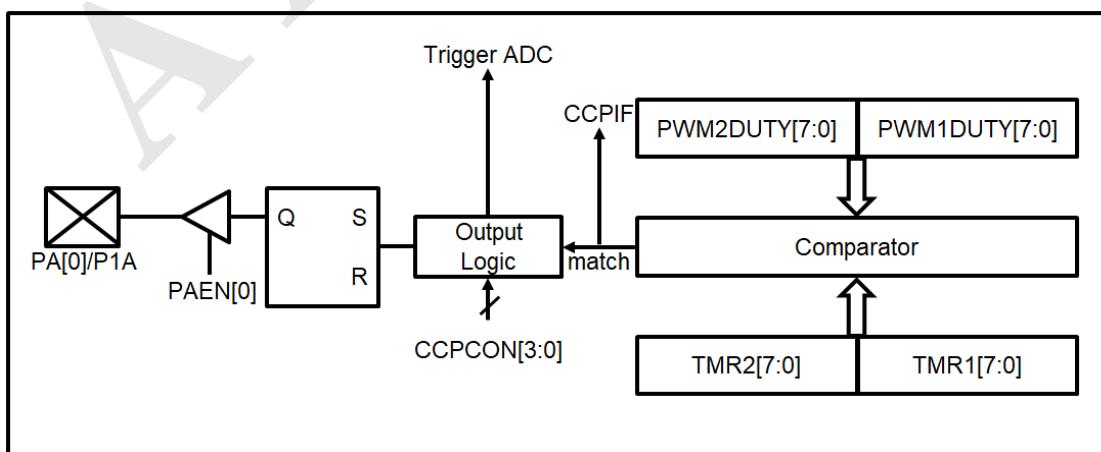
当不处于捕捉模式时, 捕捉事件计数器(或预分频器)将被清除。从一个捕捉预分频器切换到另一个可能会产生中断。此外, 预分频器计数器将不会被清除。

### 3.10.2 比较模式

在比较模式下，16位比较数据值（ $\text{PWM2DUTY}[7:0]$ :  $\text{PWM1DUTY}[7:0]$ ）不断与所选定时器的寄存器的值进行比较。当匹配发生时，CCP1（PA0）引脚可以为

- 高驱动
- 低驱动
- 切换输出
- 保持不变（仅中断）
- 如果ADC开启则触发ADC

比较模式的结构框图如下：



引脚上的动作是基于控制位CCPM[3:0]的值。当比较匹配发生时，中断标志位CCPIF被设置。

在比较模式下，用户必须配置CCP（PA0）引脚作为输出。

在比较模式下，比较定时器必须在CPU时钟同步模式下运行，否则比较操作可能无法工作。

当CCP模块开启时，该模块不会自动配置引脚作为输出。

若要用于比较功能，所选计时器必须在计时器模式或同步计数器模式下运行。在异步计数器模式下，比较操作可能无法工作。注意，CCP定时器以下数模式运行。

### 3.10.3 CCP PWM 模式

在CCP PWM模式下，CCP模块产生高达10位分辨率的PWM输出。PWM脚位为P1A（PA0）、P1B（PA5）、P1C（PA7）和P1D（PA4）。PWM周期和占空由Timer2[9:0]和PWM2DUTY[9:0]寄存器指定。

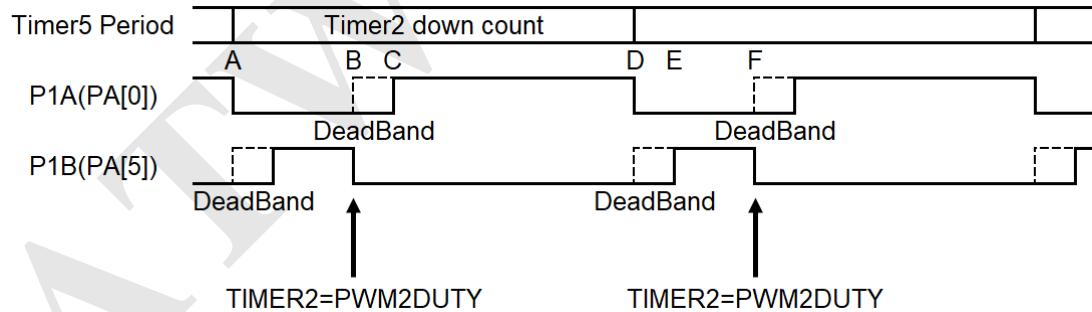
CCPCON寄存器中的PWM2M[1:0]允许以下配置之一：

- 单路输出：P1A输出。P1B, P1C, P1D被分配为正常I/O。

- 半桥输出：P1A、P1B调制，带死区控制。P1C, P1D被分配为正常I/O。
- 全桥输出，正向模式：P1D调制，P1A有效。P1B, P1C无效。
- 全桥输出，反向模式：P1B调制，P1C有效。P1A, P1D无效。

在单路输出模式下，P1A (PA0) 引脚用作PWM输出。PA0 必须设置为输出。

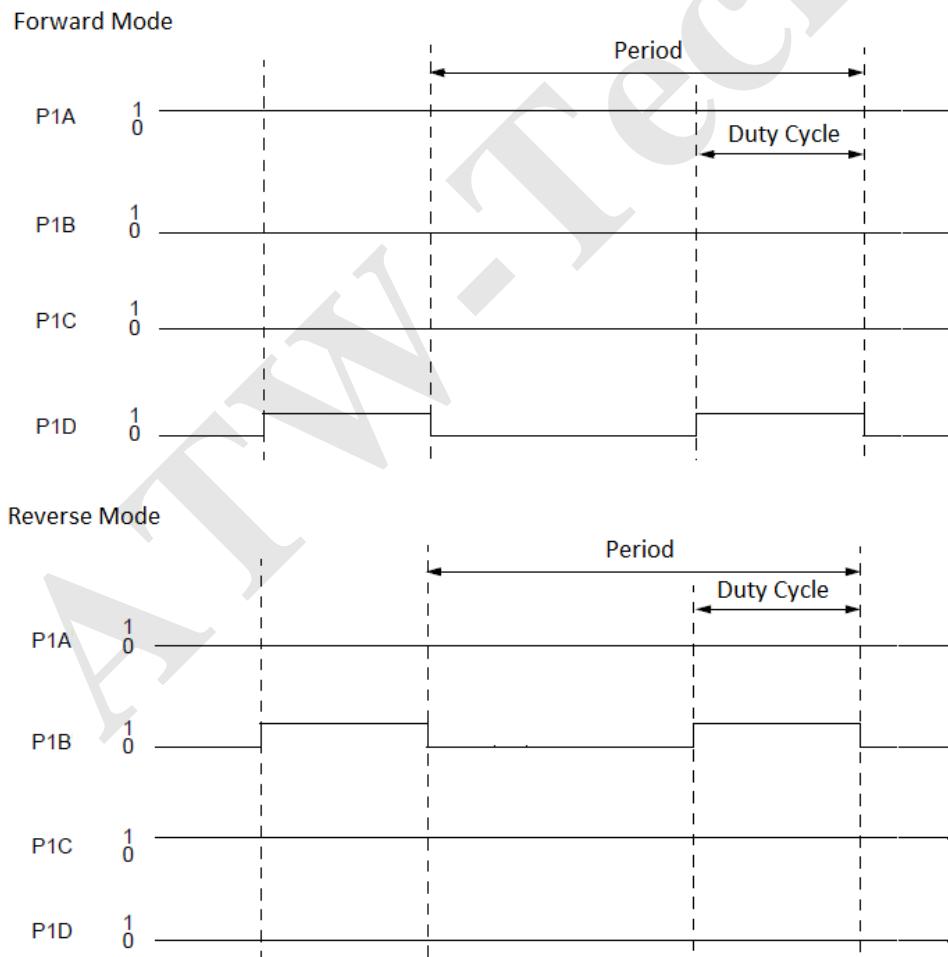
在半桥模式下，P1A (PA0) 引脚具有PWM输出信号，P1B (PA5) 引脚具有互补的PWM输出信号。在这种模式下，PA0 和PA5 必须设置为输出。



在半桥输出模式下，数字可编程死区延迟可用于避免击穿电流破坏桥电源开关。PWMDB[7:0]是半桥模式的死区延迟计数，延迟单位在CPU周期。

如上时序所示，Timer2 为 10 位下数计数器，P1A为PWM2 输出，P1B为PWM2 互补输出。在B点，Timer2 的值等于PWM2DUTY。在D点，Timer2 计数为零并重新加载值。如果没有死区控制，从B点到D点P1A输出应该是有效的。从D点到F点P1B输出应该是有效的。非零死区会延迟P1A从B点到C点的上升点，延迟P1B从D点到E点的上升点。死区范围是指从B点到C点和从D点到E点的时间。

在全桥模式下，四个引脚用于输出；但是，一次只有两个输出是有效的。在正向模式下，P1A (PA0) 持续有效，P1D (PA4) 被调制。在反向模式下，P1C (PA7) 持续有效，P1B (PA5) 被调制。在此模式下，PA0、PA5、PA7 和PA4 必须设置为输出。下面的时序图显示了全桥正向和反向的情况。



### 3.11 RFC模式

AT8B62F1 内置RFC模式。一旦启用RFC模式，所选的输入脚状态将控制Timer1 计数。当所选输入脚被识别为 0 状态（输入脚电压小于 $V_{IL}$ ）时，Timer1 继续计数。当所选引脚被识别为 1 时（输入脚电压大于 $V_{IH}$ ），Timer1 停止计数。RFC模式的工作原理如下图所示：PSEL3~0 用于从 14 个AT8B62F1 脚位中选择一个RFC输入脚。RFCEN 用于在正常使能信号T1EN和RFC选择输入状态之间切换Timer1 使能信号。

RFC模式的一个应用是测量电容-电阻充电时间，如图所示，当PSEL3~0=0x01 时，选择PA1 作为RFC输入脚。首先将PA1 设置为输出低电平（PA1 的电压放电到 0）。下一步，清除Timer1 内容，设置PA1 为输入，开启RFC模式。然后Timer1 开始计数，RC 电路开始对PA1 充电。当PA1 被充电到 $V_{IH}$ 电压时，由于PA1 输入是高电平，Timer1 计数停止。Timer1 内容将显示RC 电路充电时间。（注意：Timer1 是下数计数。）

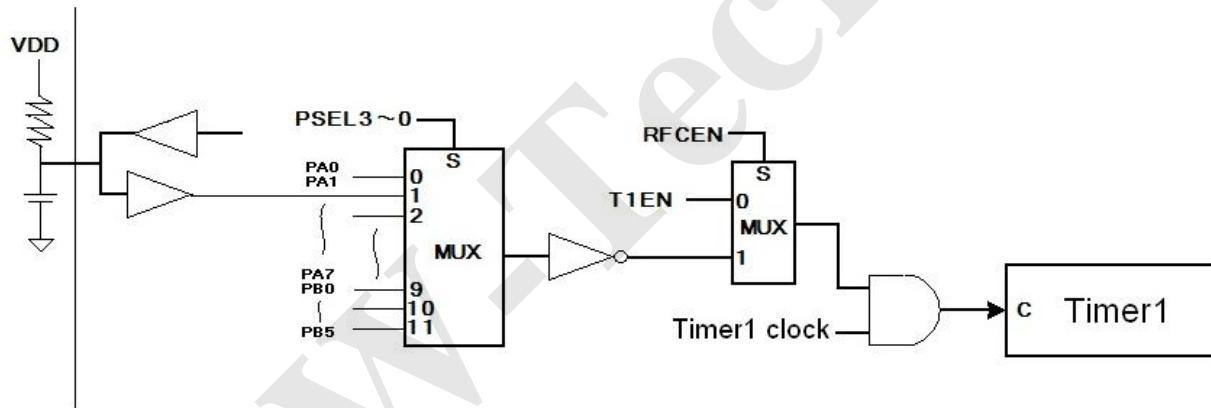


图 28 RFC结构框图

### 3.12 IR (红外) 载波

寄存器IREN (IRCR[0]) 设置为 1 后IR载波将会产生。并且，PB1 引脚自动成为输出脚。当IREN清除为 0 时，PB1 将成为配置时的通用I/O引脚。

红外载波频率由寄存器IRF57K (IRCR[1]) 选择。当IRF57K为 1 时，IR载波频率为 57KHz。当IRF57K为 0 时，IR 载波频率为 38KHz。由于红外载波频率来源于高频系统振荡 $F_{HOSC}$ ，所以在使用外部晶振时，有必要指定系统振荡的频率。寄存器IROSC358M (IRCR[7]) 用于向AT8B62F1 提供此信息。当IROSC358M为 1 时，外部晶振频率为 3.58MHz，当IROSC358M为 0 时，外部晶振频率为 455KHz。当采用内部高频振荡时，该寄存器将被忽略，并为红外模块提供 4MHz时钟。

根据红外载波引脚输出数据可选择红外载波的有效状态（极性）。当寄存器位IRCSEL (IRCR[2]) 为 1，且红外引脚输出数据为 0 时，该引脚产生红外载波。当寄存器位IRCSEL (IRCR[2]) 为 0 时，且红外引脚输出数据为 1 时，该引脚产生红外载波。红外载波的极性如下图所示。

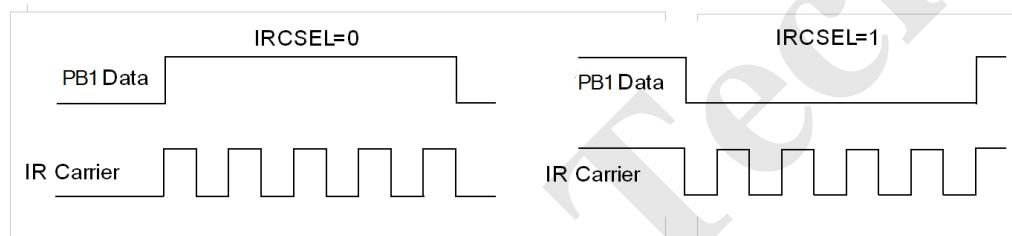


图 29 红外载波极性与输出数据

### 3.13 低电压检测 (LVD)

AT8B62F1 低电压检测 (LVD) 内置精确带隙基准，用于精确检测VDD电平。如果LVDEN (寄存器PCON[5]) =1，且VDD电压值低于LVDS[2:0]选择的LVD电压，如下表所示，则LVD输出变低。如果LVD中断被启用，LVD中断标志将是高，如果GIE=1，它将强制程序执行中断服务程序。此外，LVD的实际状态输出可以通过寄存器PCON1[6] 轮询。LVD框图如下：

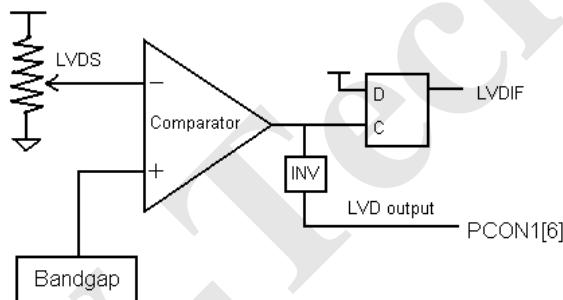


图 30 LVD 结构框图

以下是LVD电压选择表:

LVDS[2:0]	电压
000	2.0V
001	2.2V
010	2.4V
011	2.7V
100	3.0V
101	3.3V
110	3.6V
111	4.3V

表 22 LVD 电压选择

### 3.14 电压比较器

AT8B62F1 提供 1 套具有各种模拟比较模式的电压比较器和内部参考电压。比较器正相输入和反相输入可与 GPIO 共享。内部参考电压只能按路径发送到比较器的反相输入。

CMPEN (寄存器ANAEN[7]) 用于启用和禁用比较器。当 CMPEN=0 (默认) 时, 比较器被禁用。当 CMPEN=1 时, 启用比较器。在睡眠模式下, 比较器被自动禁用。

AT8B62F1 比较器有两种工作模式, P2V模式和P2P模式。这两种模式由VS[3:0] (寄存器CMPCR[3:0]) 决定。当 VS[3:0]=0 时, 比较器处于P2P模式。当 VS[3:0]=1~15 时, 为P2V模式。在配置字“比较器输入”引脚选择中, 将比较器中使用的引用设置为模拟引脚。

P2V模式具有在指定模拟引脚和指定基准之间比较电压的功能。P2V模式的结构如下图所示:

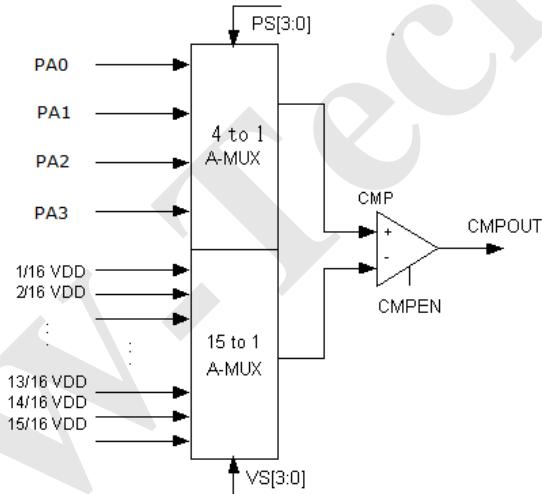


图 31 比较器P2V模式结构框图

在P2V模式下，比较器的反相输入由VS[3:0]决定。VS[3:0]用于从15个参考电压中选择一个，即1/16 VDD至15/16 VDD，如下表所示。

VS[3:0]	参考电压
0000	P2P mode
0001	1/16 V <sub>DD</sub>
0010	2/16 V <sub>DD</sub>
0011	3/16 V <sub>DD</sub>
0100	4/16 V <sub>DD</sub>
0101	5/16 V <sub>DD</sub>
0110	6/16 V <sub>DD</sub>
0111	7/16 V <sub>DD</sub>
1000	8/16 V <sub>DD</sub>
1001	9/16 V <sub>DD</sub>
1010	10/16 V <sub>DD</sub>
1011	11/16 V <sub>DD</sub>
1100	12/16 V <sub>DD</sub>
1101	13/16 V <sub>DD</sub>
1110	14/16 V <sub>DD</sub>
1111	15/16 V <sub>DD</sub>

表 23 P2V模式参考电压选择

在P2V模式下，比较器的正相输入由PS[3:0]（寄存器CMPCR[7:4]）决定。PS[3:0]从4个引脚PA0~3中选择一个作为比较器的正相输入。表格如下所示。

PS[3:0]	选择引脚
0000	PA0
0001	PA1
0010	PA2
0011	PA3
0100 ~ 1111	-

表 24 P2V 模式引脚选择

P2P模式具有在两个模拟引脚之间比较电压的功能。在此模式下VS[3:0]=0, PS[3:0]从4个模拟引脚中选择2个作为比较器的正相输入和反相输入。选择表如下：

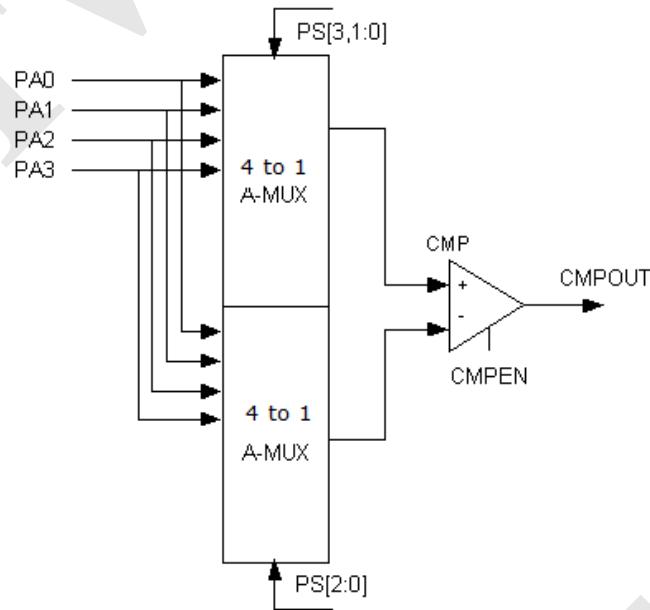


图 32 比较器P2P模式结构框图

PS[3:0]	正相输入	反相输入
0000	PA0	PA1
0001	PA1	PA0
0010	PA2	PA3
0011	PA3	PA2
0100 ~ 1111	-	-

表 25 P2P 模式引脚选择

比较器输出结果的获取有三种方式：一种是通过中断机制，一种是通过寄存器轮询，另一种是通过探测输出引脚。要使用比较器中断功能，设置 CMPEN=1 和 CMPIE=1，然后读入寄存器 OSCCR，结束比较器输出与已登记的比较器输出不匹配的情况，然后清除中断标志 CMPIF。当比较器输出改变状态时，CMPIF 设为 1，进入中断服务程序。

比较器输出可以通过 CMPOUT（寄存器 OSCCR[7]）轮询。

要在输出引脚探测比较器输出，将 CMPOE（寄存器 OSCCR[6]）设置为 1，则 PB3 将是比较器输出的实时状态。需要注意的是，当 CMPOE=1 时，如果 PWM1 功能启用，则该功能将被禁用。

### 3.15 模拟-数字转换器 (ADC)

AT8B62F1 提供 11+1 通道 12 位SAR ADC，将模拟信号转换为 12 位数字信号。ADC高参考电压是可选的。它们可以是来自PA0 或内部产生的电压VDD, 4V, 3V或 2V。模拟输入可选择从模拟信号输入引脚PA0~PA4、PB0~PB5 或从内部生成  $1/4 * VDD$ 。ADC时钟ADCLK可选择  $F_{INST}/1$ ,  $F_{INST}/2$ ,  $F_{INST}/8$  或  $F_{INST}/16$ 。采样脉冲宽度可以选择为ADCLK\*1, ADCLK\*2, ADCLK\*4 或ADCLK\*8。ADC运行前设置ADEN=1。然后设置START=1, ADC开始将模拟信号转换为数字信号。EOC=0 表示ADC正在处理中。EOC=1 表示ADC转换结束。如果ADIE=1 并且全局中断被启用，则ADC中断将在EOC低转高后发出。结构框图如下图所示。

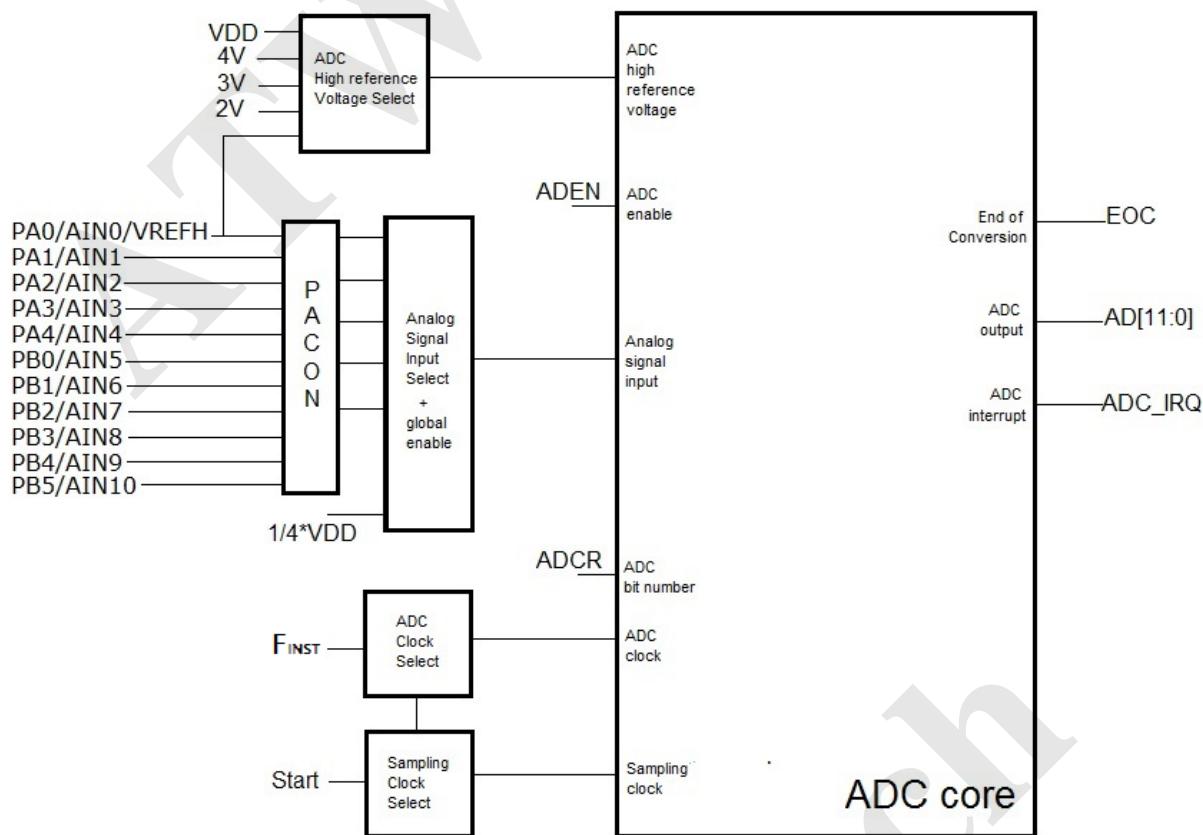


图 33 ADC结构框图

#### 3.15.1 ADC 参考电压

ADC内置 5 个高参考电压源，由ADVREFH寄存器控制。这些高参考电压源是外部电压源（PA0）和四个内部电压源（VDD, 4V, 3V, 2V）。当EVHENB位为“1”时，ADC参考电压为来自外部电压源PA0。在这种模式下，PA0 必须是介于VDD和 2V之间的一个电压。如果EVHENB位为 0，则ADC参考电压来自VHS[1:0]选择的内部电压源。如果VHS[1:0]为“11”，则ADC参考电压为VDD。如果VHS[1:0]为“10”，则ADC参考电压为 4V。如果VHS[1:0]为“01”，则ADC参考电压为 3V。如果VHS[1:0]为“00”，则ADC参考电压为 2V。内部参考电压应用的限制是VDD 不能低于内部电压的每一个电平，或者该电平等于VDD。ADC采样电压范围受高/低参考电压的限制。ADC低参考电压为VSS，不可改变。ADC高参考电压包括内部VDD/4V/3V/2V和来自PA0 引脚的外部参考电压源。ADC 参考电压范围限制为 (ADC高参考电压-低参考电压)  $\geq 2V$ 。ADC低参考电压为VSS=0V。因此ADC的高参考电压范围为 2V ~ VDD。

ADC模拟输入信号电压必须从ADC低参考电压到ADC高参考电压。如果ADC模拟输入信号电压超过此范围，则ADC转换结果不可预料（满量程或零）。

EVHENB	VHS[1:0]	参考电压
1	xx	PA0
0	11	VDD
0	10	4V
0	01	3V
0	00	2V

表 26 ADC参考电压选择

### 3.15.2 ADC 模拟输入通道

ADC采用CHS[3:0]和GCHS选择模拟输入源。GCHS是全局通道选择。即在选择和转换任何模拟输入源之前，GCHS必须为1。

GCHS	CHS[3:0]	ADC模拟输入源
0	xxxx	x
1	0000	PA0
1	0001	PA1
1	0010	PA2
1	0011	PA3
1	0100	PA4
1	0101	PB0
1	0110	PB1
1	0111	PB2
1	1000	PB3
1	1001	PB4
1	1010	PB5
1	1011	1 / 4 * VDD
1	11xx	N.C.

表 27 ADC模拟输入源选择

ADC输入引脚与数字I/O引脚共享。将模拟信号连接到这些引脚可能会导致I/O引脚的额外电流泄漏。在断电模式下，上述漏电流将是一个大问题。PACON[0:4]是PA[0:4]配置寄存器，PACON[5:7]是PB[0:2]配置寄存器，ADCR[4:6]是PB[3:5]配置寄存器来解决上述问题。将“1”写入PACON和ADCR[4:6]，将相关PA/PB引脚配置为纯模拟输入引脚，避免漏电流，它不能作为普通I/O使用。

除了设置PxCON和ADCR[4:6]寄存器位外，所选的模拟输入引脚必须设置为输入模式，并且必须关闭内部上拉/下拉，否则可能影响模拟输入电平。

### 3.15.3 ADC 时钟 (ADCLK)、采样时钟 (SHCLK) 和位数选择

ADC时钟 (ADCLK)、采样脉宽 (SHCLK) 和转换位数的选择影响转换速度和转换精度。ADCLK是ADC的基准时钟。在SAR ADC的工作过程中，位操作与ADCLK同步。SHCLK是模拟信号采样时间的持续时间，较大的SHCLK会更接近模拟信号的原始电平，但会减慢ADC的转换速度。亦然。ADC可以根据ADCR[1:0]寄存器位选择不同的转换位数。有 2 bit 数字可供选择，分别是 12-bit、10-bit 和 8-bit。转换位数越少，ADC的转换速率越快，但有效的ADC位数就越少。转换位数越多，转换速率越慢，但转换精度越高。

ADC时钟来自I\_HRC，可从ADCK[1:0]中选择。

ADCK[1:0]	ADC 时钟
0 0	$F_{INST}/16$
0 1	$F_{INST}/8$
1 0	$F_{INST}/1$
1 1	$F_{INST}/2$

表 28 ADC时钟选择

采样时钟宽度来自ADCLK，可从SHCK[1:0]中选择。

SHCK[1:0]	采样时钟
0 0	1 ADCLK
0 1	2 ADCLK
1 0	4 ADCLK
1 1	8 ADCLK

表 29 ADC采样时钟选择

ADC位数选择来自ADCR[1:0]。

ADCR[1:0]	转换位数
0 0	8-bit
0 1	10-bit
1 x	12-bit

表 30 转换位数选择

ADC转换时间从START (开始ADC转换) 到EOC=1 (ADC转换结束)。持续时间取决于ADC分辨率、ADC时钟速率和采样时钟宽度。

**ADC转换时间≈采样时钟宽度+ (ADC 位数+ 2)\*ADCLK宽度**

下表是ADC转换时间和转换速率的一些示例。

Bit 数.	ADC 时钟	SHCLK	转换时间 (ADCLK No.)	$F_{INST}=2MHz$		$F_{INST}=250K$	
				时间	速率	时间	速率
12	$F_{INST}/16$	8 ADCLK	22	176us	5.68kHz	1408us	710Hz
12	$F_{INST}/1$	1 ADCLK	15	7.5us	133.3kHz	60us	16.7kHz
10	$F_{INST}/1$	1 ADCLK	13	6.5us	153.8kHz	52us	19.2kHz
8	$F_{INST}/1$	1 ADCLK	11	5.5us	181.8kHz	44us	22.7kHz

表 31 ADC转换时间

### 3.15.4 ADC 操作过程

设置ADC时钟（ADCLK）、采样时钟宽度（SHCLK）、转换位数（ADCR）、ADC高参考电压（ADVREFH）、选择输入通道和PACON相关位。然后设置ADEN=1。

设置ADEN=1后，ADC工作之前至少要等待256us（ADC内部偏置稳定时间）。将START写入1以启动ADC转换会话。ADC处理期间EOC=0。轮询EOC=1或在ADC转换结束时等待ADC中断。

### 3.16 看门狗定时器（WDT）

AT8B62F1中有一个自由运行的振荡器被WDT所使用。由于该振荡器与其它振荡电路无关，故在待机模式和睡眠模式中WDT仍能继续工作。

WDT能被配置字开启或关闭。当WDT被配置字开启时，在程序运行过程中，WDT仍然可以通过WDTEN位（寄存器PCON[7]）来控制。此外，WDT超时后可由另一个配置字决定复位AT8B62F1或发出的中断请求。同时，在WDT超时后，寄存器/TO（STATUS[4]）位将被清除为0。

WDT超时的时基可以是3.5毫秒、15毫秒、60毫秒或250毫秒，由两个配置字决定。如果将预分频器0分配给WDT，则可以延长超时周期。通过将1写入寄存器PS0WDT位，预分频器0将分配给WDT。预分频器0对WDT的分频比由寄存器PS0SEL[2:0]位决定，而且取决于WDT的超时机制。如果WDT超时将复位AT8B62F1，分频比从1:1到1:128。如果选为WDT中断时，则分频比从1:2到1:256。

当预分频器0分配给WDT时，执行CLRWDTR指令将清除WDT、预分频器0。并设置/TO标志位为1。

如果用户选择WDT中断机制，在WDT超时后，寄存器WDTIF（INTF[6]）位将设置为1。如果寄存器WDTIE（INTE[6]）位和GIE位都设置为1，则可能产生中断请求。直到程序将0写入WDTIF，WDTIF才会被清除为0。

### 3.17 中断

AT8B62F1提供二种中断：一种是软件中断，另一种是硬件中断。软件中断由执行指令INT来产生。硬件中断则有以下十一种：

- Timer0上溢中断。
- Timer1下溢中断。
- Timer2下溢中断或CCP中断。
- Timer3下溢中断。
- WDT中断。
- PA/PB输入状态改变中断。
- 外部中断0输入。
- 外部中断1输入。
- 低电压侦测中断。
- 比较器输出翻转中断。
- ADC模数转换完成中断。

**GIE**是总中断屏蔽位，必须为 1 才能开启硬件中断功能。**GIE**可以通过**ENI**指令设置 1，通过**DISI**指令清除为 0。

执行完指令**INT**后，无论**GIE**是置 1 还是清除为零，下一条指令都将从地址 0x001 读取。同时，**GIE**将由AT8B62F1自动清除为零。当发生硬件中断时，相应的中断标志位将被设置为 1。该位在程序将 0 写入该位之前不会清除为零。因此，用户可以通过轮询相应的中断标志位得知哪个硬件引发中断。需注意只有当相应的中断使能位设置为 1 时，才能正确地读取相应的中断标志。如果相应的中断使能位设置为 1，**GIE**也为 1，将发生硬件中断，下一条指令将从 0x008 执行。同时，AT8B62F1 将自动清除寄存器**GIE**位为零。需要注意的是，**ENI**指令不能放在**RETIE**指令的正前方，因为中断服务程序中的**ENI**指令会触发嵌套中断，而**RETIE**会在跳出**ISR**后清除内部中断处理，因此有可能出现中断标志被错误清除的情况。

### 3.17.1 Timer0 上溢中断

Timer0 上溢（从 0x00 到 0xFF）将设置寄存器位**T0IF**。如果**T0IE**和**GIE**设置为 1，这个中断请求将被处理。

### 3.17.2 Timer1 下溢中断

Timer1 下溢（从 0x3FF 到 0x00）将设置寄存器位**T1IF**。如果**T1IE**和**GIE**设置为 1，这个中断请求将被处理。

### 3.17.3 Timer2 下溢中断或 CCP 中断

Timer2 下溢（从 0x3FF 到 0x00）将设置寄存器位**T2IF/CCPIF**。如果**T2IE/CCPIE**和**GIE**设置为 1，这个中断请求将被处理。

### 3.17.4 Timer3 下溢中断

Timer3 下溢（从 0x3FF 到 0x00）将设置寄存器位**T3IF**。如果**T3IE**和**GIE**设置为 1，这个中断请求将被处理。

### 3.17.5 WDT 超时中断

当WDT超时时，配置字选择WDT超时将产生中断请求，它将设置寄存器位**WDTIF**。如果**WDTIE**和**GIE**设置为 1，这个中断请求将被处理。

### 3.17.6 PA/PB 输入状态改变中断

当PAx ( $0 \leq x \leq 7$ )，PBy ( $0 \leq y \leq 5$ ) 设置为输入口且相应的寄存器WUPAx、WUPBx位设置为 1，且如果**PABIE**和**GIE**设置为 1，当这些选定输入口上的状态变化时，寄存器**PABIF**位将被硬件设为 1 并处理此中断请求。需注意当PB0、PB1 同时设置为状态变化中断和外部中断时，设置**EIS0=1** 或 **EIS1=1** 将禁止PB0、PB1状态变化中断。

### 3.17.7 外部中断 0 输入

根据**EIS0=1** 和寄存器**INTEDG**的配置，如果**INT0IE**和**GIE**设置为 1，PB0 引脚上的有效边沿触发会让寄存器**INT0IF**位将被硬件设为 1 并处理此中断请求。

### 3.17.8 外部中断 1 输入

根据EIS1=1 和寄存器INTEDG的配置, 如果INT1IE和GIE设置为 1, PB1 引脚上的有效边沿触发会让寄存器INT1IF位将被硬件设为 1 并处理此中断请求。

### 3.17.9 低电压侦测中断

当VDD电压低于LVD电压时, LVD标志由高变低, 并设置寄存器位LVDIF=1。如果LVDIE和GIE设置为 1, 这个中断请求将被处理。

### 3.17.10 比较器输出翻转中断

当比较器输出状态发生变化时, 就会触发比较器中断。如果CMPIE和GIE设置为 1, 该中断请求将被处理。注意, 在比较器中断发生之前, 需要读取寄存器OSCCR来清除先前的比较器输出状态差异。

### 3.17.11 模数转换完成中断

当ADC转换结束信号发出时, 就会触发ADC中断。如果ADIE和GIE被设置为 1, 这个中断请求将被处理。

## 3.18 振荡器配置

因为AT8B62F1 是双时钟IC, 有高振荡时钟 ( $F_{Hosc}$ ) 和低振荡时钟 ( $F_{Losc}$ ) 可选择作为系统振荡时钟 ( $F_{osc}$ )。可用作 $F_{Hosc}$ 的振荡器有内部高速RC振荡器 ( $I\_HRC$ )、外部高速晶体振荡器 ( $E\_HXT$ ) 与外部晶体振荡器 ( $E\_XT$ )。可用作 $F_{Losc}$ 的振荡器是内部低速RC振荡器 ( $I\_LRC$ ) 与外部低速晶体振荡器 ( $E\_LXT$ )。

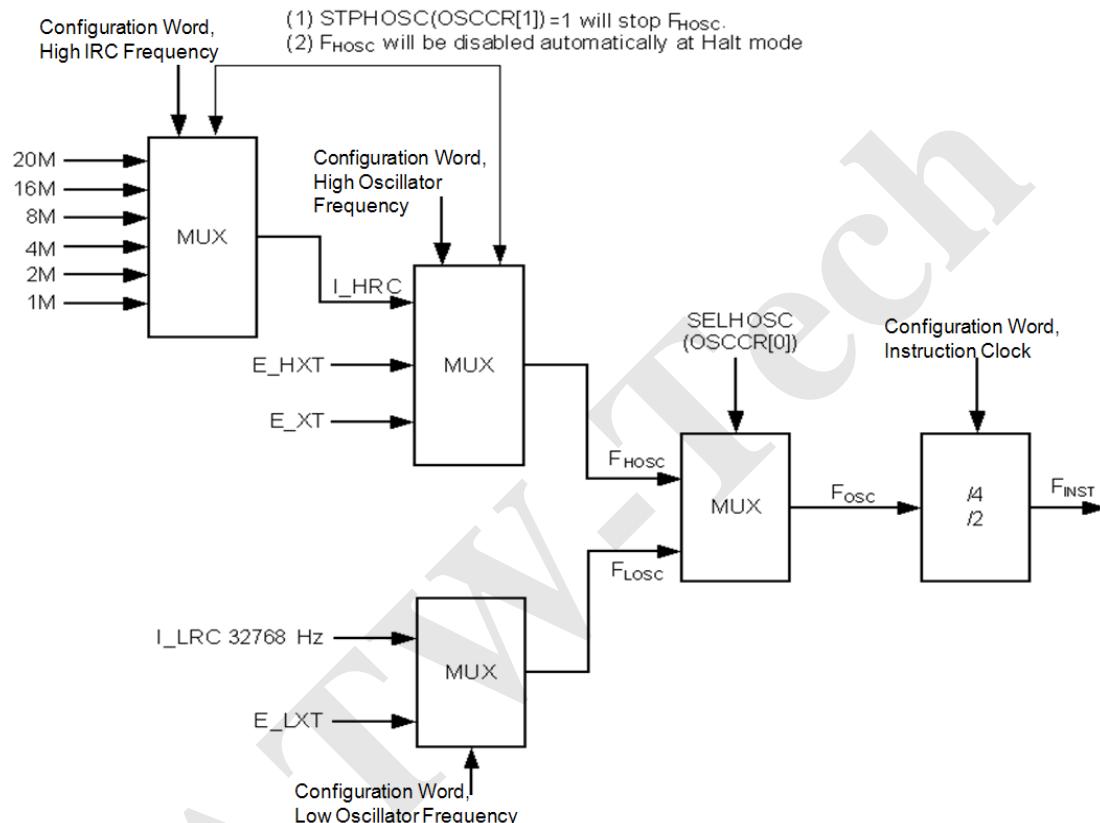


图 34 AT8B62F1 振荡配置结构图

有两个配置字来确定哪个振荡器将被用作  $F_{Hosc}$ 。当  $I_{HRC}$  被选为  $F_{Hosc}$  时,  $I_{HRC}$  输出频率由三个配置字决定, 它可以是 1M、2M、4M、8M、16M 或 20MHz。此外, 外部晶体振荡器引脚 PA6 和 PA7 可以用作 I/O 引脚。另一方面, 根据一个配置字的设置, PA7 可以是指令时钟的输出脚。如果  $F_{Hosc}$  需要外部晶体, 其频率范围从 8MHz 到 20MHz, 建议使用 E\_HXT。如果  $F_{Hosc}$  需要外部晶体, 其频率范围从 455KHz 到 6MHz, 推荐 E\_XT。当使用 E\_HXT 或 E\_XT 时, PA6 / PA7 不能被用作 I/O 引脚。它们必须用作晶体输入脚和输出脚。PA7 是晶体输出脚(Xout), PA6 是晶体输入脚(Xin)。

有一个配置字来确定哪个振荡器将被用作  $F_{Losc}$ 。当选择  $I_{LRC}$  时, 它的频率集中在 32768Hz。如果需要  $F_{Losc}$  外部晶体, 则选择 E\_LXT, 只允许 32768Hz 晶体。E\_LXT 时, 不能使用 PA6/PA7 作为 I/O 引脚。它们必须用作晶体输出引脚和输入引脚。PA7 为晶体输出引脚(Xout), PA6 为晶体输入引脚(Xin)。下面列出了  $F_{Hosc}$  和  $F_{Losc}$  的双时钟组合。

No.	$F_{Hosc}$	$F_{Losc}$
1	$I_{HRC}$	$I_{LRC}$
2	E_HXT or E_XT	$I_{LRC}$
3	$I_{HRC}$	E_LXT

表 32 双时钟组合

当 E\_HXT, E\_XT 或 E\_LXT 作为其中一个振荡时, 晶体或谐振器被连接到 Xin 和 Xout 以提供振荡。此外, 为了提供可靠的振荡, 建议按下图连接一个电阻和两个电容器。参考晶体或谐振器的规格选用合适的 C1 或 C2 值。C1 和 C2 的推荐值如下表所示。

振荡模式	振荡频率(Hz)	C1, C2 (pF)
E_HXT	16M	5 ~ 10
	10M	5 ~ 30
	8M	5 ~ 20
E_XT	4M	5 ~ 30
	1M	5 ~ 30
	455K	10 ~ 100
E_LXT	32768	5 ~ 30

表 33 不同类型的外部晶体振荡器频率所推荐的 C1 和 C2 电容器数值

对于 2 个时钟 CPU 周期模式下的 20MHz 谐振器, C2 必须接 18pF 电容。

为得到精准且稳定的 32.768kHz 频率, 选择正确的 C1 和 C2 电容器数值是相当重要的。每家晶振厂商数据手册中都有记载低速晶体振荡器的负载电容值 (CL), 外接 C1 和 C2 电容器数值的计算如下公式:

$$C1=C2=2*CL-Cbt$$

其中 Cbt 是 AT8B62F1 内部电容值, 大约为 5pF。假设低速晶体振荡器的负载电容值 CL=12.5pF, 依公式算出的 C1=C2=20pF。

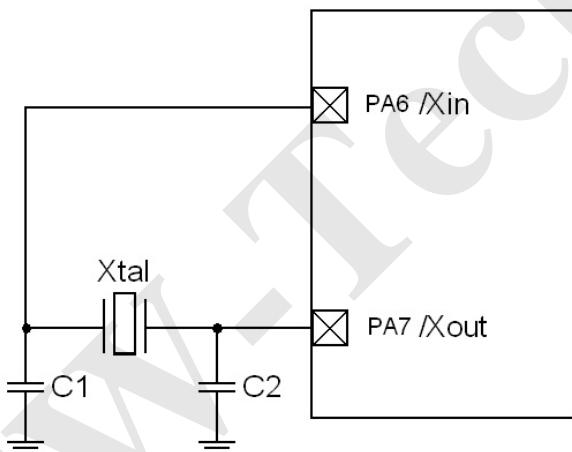


图 35 外部晶体振荡器的硬件连接图

根据寄存器SELHOSC (OSCCR [0]) 位的值, 可以选择 $F_{Hosc}$ 或 $F_{Losc}$ 作为系统振荡时钟 $F_{osc}$ 。当SELHOSC为1时, 选择 $F_{Hosc}$ 作为 $F_{osc}$ 。当SELHOSC为0时, 选择 $F_{Losc}$ 作为 $F_{osc}$ 。一旦确定 $F_{osc}$ , 根据配置字设置, 指令时钟可以选择为 $F_{osc}/2$ 或 $F_{osc}/4$ 。

### 3.19 工作模式

AT8B62F1 提供了四种操作方式来定制各种应用和节省电力消耗, 分别是正常模式、慢速模式、待机模式和睡眠模式。正常模式被指定为高速运行模式, 慢速模式被指定为低速模式, 以节省功耗。在待机模式下, AT8B62F1 将停止几乎所有的运作, 可由定时器 0/1/2/3、看门狗来唤醒。在睡眠模式下, AT8B62F1 将睡眠, 直到外部事件或看门狗定时器来唤醒。

四种工作模式如下图所示。

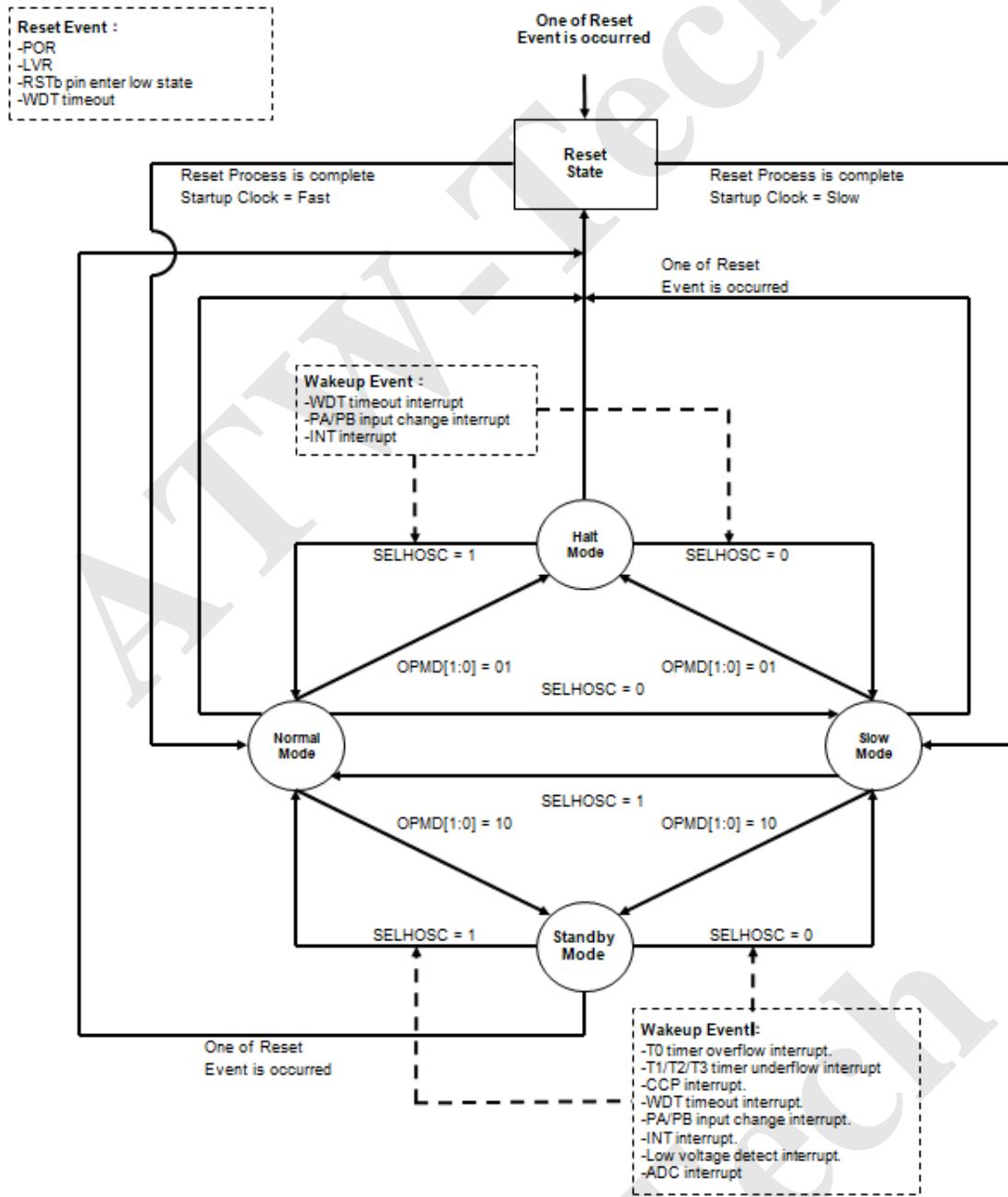


图 36 四种工作模式

### 3.19.1 正常模式

发生任何复位事件并且复位过程完成后, AT8B62F1 将在正常模式或慢速模式下开始执行程序。重置过程后选择的模式由启动时钟配置字决定。如果启动时钟=fast, AT8B62F1 将进入正常模式, 如果启动时钟=Slow, AT8B62F1 将进入慢速模式。在正常模式下, 为提供最高性能而以  $F_{Hosc}$  作为系统振荡时钟, 其功耗在四种操作模式中将是最大的。在上电或任何重置触发器被释放后, 待复位程序完成, AT8B62F1 将进入正常模式。

- 指令的执行是基于  $F_{Losc}$  且所有硬件功能可以根据相应的硬件使能位来开启/关闭。
- $F_{Losc}$  仍运行。
- IC 可由写 0 至寄存器 SELHOSC (OSCCR[0]) 位切换为慢速模式。
- IC 可通过寄存器 OPMD[1:0] (OSCCR[3:2]) 位切换为待机或睡眠模式。
- 对于实时时钟的应用, AT8B62F1 在运行正常模式时可同时将低频振荡时钟设为 Timer0 的时钟源, 这是通过设置 LCKTMO 为 1 和配置字中 Timer0 时钟源来实现。

### 3.19.2 慢速模式

通过写 0 至寄存器 SELHOSC 位, AT8B62F1 将进入慢速模式。在慢速模式下, 为节省功耗,  $F_{Losc}$  被选为系统振荡时钟。然而,  $F_{Hosc}$  将不会自动被 AT8B62F1 关闭。因此在慢速模式下, 用户可写 1 至寄存器 STPHOSC (OSCCR[1]) 位来停止  $F_{Hosc}$  进一步降低功耗。但需注意的是, 禁止进入慢速模式同时停止  $F_{Hosc}$ , 必须先进入慢速模式, 然后关闭  $F_{Hosc}$ 。

- 指令执行是基于  $F_{Losc}$  且所有硬件功能可以根据相应的硬件使能位来开启/关闭。
- 通过写 1 至寄存器 STPHOSC 位,  $F_{Hosc}$  可以被停止。
- IC 可通过寄存器 OPMD[1:0] 位切换为待机模式或睡眠模式。
- IC 可通过写 1 至寄存器 SELHOSC 切换到正常模式。

### 3.19.3 待机模式

通过写入 10b 至寄存器 OPMD[1:0], AT8B62F1 将进入待机模式。然而, 在待机模式下,  $F_{Hosc}$  不会自动被 AT8B62F1 关闭, 用户必须先进入慢速模式后写入 1 至寄存器 STPHOSC 位, 以停止  $F_{Hosc}$ 。部分 AT8B62F1 的硬件功能会被关闭, 但 T0EN / T1EN / T2EN / T3EN 位被设置为 1 时定时器仍可运作。因此 Timer0 / Timer1 / Timer2 / Timer3 失效后 AT8B62F1 会被唤醒。失效时间由寄存器 TMR0/TMR1[9:0]/TMR2[9:0]/TMR3[9:0]、 $F_{INST}$  和 Timer0/Timer1/Timer2/Timer3 的其他配置决定。

- 停止执行指令且一些硬件功能可以根据相应的硬件使能位来开启/关闭。
- 由写入 1 至寄存器 STPHOSC 位  $F_{Hosc}$  可以被关闭。
- $F_{Losc}$  仍保持运作。
- 如遇以下任一状况 IC 便能从待机模式唤醒:
  - (a) Timer0/Timer1/Timer2/Timer3 (上溢/下溢) 中断
  - (b) 看门狗超时中断
  - (c) PA/PB 输入状态改变中断
  - (d) INT 外部中断。
- 在从待机模式唤醒后, 如 SELHOSC=1, IC 将回到正常模式, 如 SELHOSC=0 则 IC 将回到慢速模式。
- 不建议在改变振荡模式的同时进入待机模式 (正常到慢速/慢速到正常)。

### 3.19.4 睡眠模式

通过执行SLEEP指令或写入 01b至寄存器OPMD[1:0]位, AT8B62F1 将进入睡眠模式。在进入睡眠模式后, 寄存器/PD (STATUS[3]) 位将清除为 0, 寄存器/TO (STATUS[4]) 位将设置为 1 且清除WDT并保持运作。

在睡眠模式下, 所有硬件功能是被关闭的, 停止指令执行且AT8B62F1 只能通过一些特殊事件唤醒。因此, 睡眠模式是AT8B62F1 最省电的模式。

- 指令执行停止, 所有硬件功能关闭。
- $F_{Hosc}$ 和 $F_{Losc}$ 两者都自动关闭。
- 如遇以下任一状况IC便能从睡眠模式中唤醒:
  - (a)看门狗超时中断 (b)PA/PB输入状态改变中断 (c)INT或外部中断。
- 从睡眠模式唤醒后, 如SELHOSC=1, IC将回到正常模式, 如SELHOSC=0 则IC将回到慢速模式。  
**注意: 用户可以在同一指令中更改STPHOSC并进入睡眠模式。**
- 不建议在改变振荡模式 (正常到慢速/慢速到正常) 的同时进入待机模式。

### 3.19.5 唤醒稳定时间

睡眠模式的唤醒稳定时间由配置字决定: 高速振荡器频率或低速振荡器频率。如果选择E\_HXT、E\_XT和E\_LXT 中的一个, 则唤醒周期为  $512*F_{osc}$ 。如果不选择XT模式, 则将  $16*F_{osc}$ 设置为唤醒周期。另一方面, 待机模式不需要唤醒稳定时间, 因为 $F_{Hosc}$ 或 $F_{Losc}$ 仍在待机模式下运行。

在AT8B62F1 进入待机模式或睡眠模式之前, 用户可以执行ENI指令。在这种情况下, AT8B62F1 将分支到地址 0x008, 以便在唤醒后执行中断服务程序。如果指令DISI在进入待机模式或睡眠模式之前执行, 则唤醒后将执行下一条指令。在“从睡眠中唤醒”的“SLEEP”指令后, 用户必须要NOP\*2。

```
;choice one way to enter Halt mode
;sleep
    movia  C_Halt_Mode | C_FHOSC_Sel      ; 1. Execute instruction to enters Halt mode (from Normal mode)
    sfun   OSCCR                         ; 2. Set OSCCR register to enters Halt mode (from Normal mode)
    ; Pr_SYS_Ctrl

;-----[Red Box]-----;
    nop
    nop                                ; for Interrupt latency time

    bcr    PORTB,3                      ; while wakeup from Halt mode then set PB3 outputs low
    movia  ~C_INT_PABKey
    movar  INTF                          ; Clear PABIF(PortB input change interrupt flag bit)
```

### 3.19.6 工作模式概述

四种工作模式概述如下：

模式	正常模式	慢速模式	待机模式	睡眠模式
$F_{HOSC}$	使能	STPHOSC	STPHOSC	关闭
$F_{LOSC}$	使能	使能	使能	关闭
Instruction Execution	执行	执行	停止	停止
Timer0/1/2/3	TxEN	TxEN	TxEN	关闭
WDT	配置和WDTEN	配置和WDTEN	配置和WDTEN	配置和WDTEN
其它硬件	硬件使能位	硬件使能位	硬件使能位	全部关闭
唤醒源	-	-	<ul style="list-style-type: none"> <li>- Timer0 上溢</li> <li>- Timer1/2/3 下溢</li> <li>- CCP 中断</li> <li>- WDT 超时</li> <li>- PA/PB 输入状态改变</li> <li>- 外部中断 0/1</li> <li>- LVD 中断</li> <li>- 比较器中断</li> <li>- ADC 中断</li> </ul>	<ul style="list-style-type: none"> <li>- WDT超时</li> <li>- PA/PB输入状态改变</li> <li>- 外部中断 0/1</li> </ul>

表 34 工作模式概述

### 3.20 复位

当以下任一复位事件发生时，AT8B62F1 将会进入复位状态并开始复位动作：

- 当VDD检测到上升沿时为上电复位。
- 当VDD电压低于预设的LVR电压时，为LVR复位。
- RSTb引脚为低电平。
- WDT超时复位。

此外，所有寄存器如果初始值未知时，寄存器将会被初始化为初始值或保持不变。状态位/TO和/PD可以根据复位事件来初始化。/TO和/PD的值及其相关的事件概述如下：

事件	/TO	/PD
POR, LVR	1	1
非睡眠模式时发生RSTb复位	不变	不变
睡眠模式时发生RSTb复位	1	1
非睡眠模式时发生WDT复位	0	1
睡眠模式时发生WDT复位	0	0
执行SLEEP指令	1	0
执行CLRWDAT指令	1	1

表 35 /TO和/PD值和相关事件概述

复位事件发生后, AT8B62F1 将会开始复位进程。无论采用什么样的振荡器, 它将等待一定的周期使振荡稳定。这个周期被称为上电复位时间, 它由三位配置字决定, 这个时间可能是 140us, 4.5ms, 18ms, 72ms或 288 ms。

在启动复位时间后, AT8B62F1 将等待进一步的振荡器启动时间(OST), 然后开始执行程序。如果之前的启动时间为 140us, 那么  $F_{osc}$  的时钟周期为  $OST=1$  个时钟周期, 如果之前的启动时间为 4.5ms、18ms、72ms或 288 毫秒,  $OST = 16$  个时钟周期。

振荡器稳定后, AT8B62F1 将等待  $F_{osc}$  的 16 个时钟周期 (OST, 振荡器启动时间) 后完成复位。若上电复位时间设为 140us时, 将等待  $F_{osc}$  的 1 个时钟周期后完成复位。若上电复位时间设为 4.5ms, 18ms, 72ms或 288 ms 时, 将等待  $F_{osc}$  的 16 个时钟周期后完成复位。

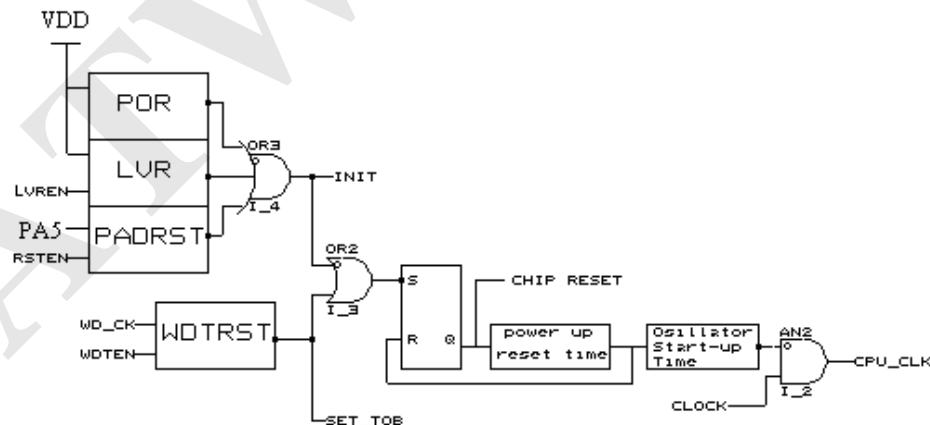


图 37 芯片复位电路框图

对于VDD上电缓慢的情况, 建议使用RSTb复位, 如下图所示。

- 建议R阻值不大于 40KΩ。
- R1 值= 100Ω ~ 1KΩ时, 将阻止过大电流, ESD或电气过载信号进入复位引脚。
- 该二极管有助于在断电时快速放电。

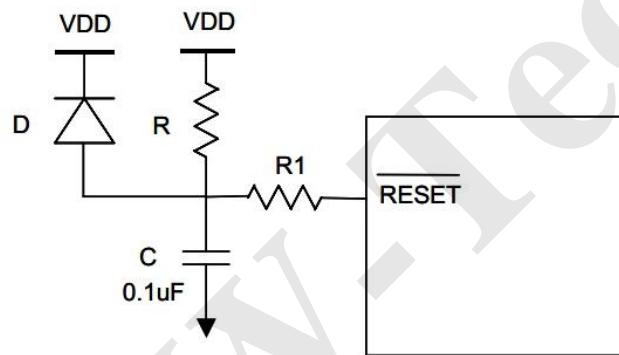


图 38 复位应用框图

#### 4. 指令设置

AT8B62F1 为各种应用程序提供了 55 个强大的指令。

指令	助记符		说明	周期数	影响标志
	1	2			
<b>Arithmetic Instructions</b>					
ANDAR	R	d	dest = ACC & R	1	Z
IORAR	R	d	dest = ACC   R	1	Z
XORAR	R	d	dest = ACC $\oplus$ R	1	Z
ANDIA	i		ACC = ACC & i	1	Z
IORIA	i		ACC = ACC   i	1	Z
XORIA	i		ACC = ACC $\oplus$ i	1	Z
RRR	R	d	Rotate right R	1	C
RLR	R	d	Rotate left R	1	C
BSR	R	bit	Set bit in R	1	-
BCR	R	bit	Clear bit in R	1	-
INCR	R	d	Increase R	1	Z
DECR	R	d	Decrease R	1	Z
COMR	R	d	dest = ~R	1	Z
<b>Conditional Instructions</b>					
BTRSC	R	bit	Test bit in R, skip if clear	1 or 2	-
BTRSS	R	bit	Test bit in R, skip if set	1 or 2	-
INCRSZ	R	d	Increase R, skip if 0	1 or 2	-
DECRSZ	R	d	Decrease R, skip if 0	1 or 2	-
<b>Data Transfer Instructions</b>					
MOVAR	R		Move ACC to R	1	-
MOVR	R	d	Move R	1	Z
MOVIA	i		Move immediate to ACC	1	-
SWAPR	R	d	Swap halves R	1	-
IOST	F		Load ACC to F-page SFR	1	-
IOSTR	F		Move F-page SFR to ACC	1	-
SFUN	S		Load ACC to S-page SFR	1	-
SFUNR	S		Move S-page SFR to ACC	1	-
T0MD			Load ACC to T0MD	1	-
T0MDR			Move T0MD to ACC	1	-
TABLEA			Read ROM	2	-

指令	助记符		说明	周期数	影响标志
	1	2			
<b>Arithmetic Instructions</b>					
ADDAR	R	d	dest = R + ACC	1	Z, DC, C
SUBAR	R	d	dest = R + (~ACC)	1	Z, DC, C
ADCAR	R	d	dest = R + ACC + C	1	Z, DC, C
SBCAR	R	d	dest = R + (~ACC) + C	1	Z, DC, C
ADDIA	i		ACC = i + ACC	1	Z, DC, C
SUBIA	i		ACC = i + (~ACC)	1	Z, DC, C
ADCIA	i		ACC = i + ACC + C	1	Z, DC, C
SBCIA	i		ACC = i + (~ACC) + C	1	Z, DC, C
DAA			Decimal adjust for ACC	1	C
CMPAR	R		Compare R with ACC	1	Z, C
CLRA			Clear ACC	1	Z
CLRR			Clear R	1	Z
<b>Other Instructions</b>					
NOP			No operation	1	-
SLEEP			Go into Halt mode	1	/TO, /PD
CLRWDT			Clear Watch-Dog Timer	1	/TO, /PD
ENI			Enable interrupt	1	-
DISI			Disable interrupt	1	-
INT			Software Interrupt	3	-
RET			Return from subroutine	2	-
RETIE			Return from interrupt and enable interrupt	2	-
RETIA	i		Return, place immediate in ACC	2	-
CALLA			Call subroutine by ACC	2	-
GOTOA			unconditional branch by ACC	2	-
LCALL	adr		Call subroutine	2	-
LGOTO	adr		unconditional branch	2	-

表 36 指令设置

**ACC:** 累加器。

**adr:** 地址。

**bit:** R-page中 8 位寄存器的位地址。

**C:** 进位/借位。

C=1, 加法指令有进位, 减法指令无借位。

C=0, 加法指令无进位, 减法指令有借位。

**d:** 目标。

若d=“0” , 结果存入ACC。

若d=“1” , 结果存入R寄存器。

**DC:** 半字节进位/借位标记。

**dest:** 目标。

**F:** F 页面特殊功能寄存器, F 值为 0x5~0xF。

**i:** 8 位立即数。

**PC:** 程序计数器。

**PCHBUF:** 程序计数器的高字节。

**/PD:** 睡眠标志位。

/PD=1, 上电或CLRWDT指令执行后。

/PD=0, SLEEP指令执行后。

**Prescaler:** 预分频器。

**R:** R 页面特殊功能寄存器, R 值为 0x0~0x7F。

**S:** S 页面特殊功能寄存器, S 值为 0x0~0x15。

**T0MD:** T0MD寄存器。

**TBHP:** 表格指针高字节寄存器。

**TBHD:** 表格数据高字节寄存器。

**/TO:** 看门狗超时标志位。

/TO=1, 上电或执行 CLRWDT 或 SLEEP 指令后。

/TO=0, 看门狗超时。

**WDT:** 看门狗计时器。

**Z:** 清零标志。

ADCAR	Add ACC and R with Carry	ADDR	Add ACC and R
语法	ADCAR R, d	语法	ADDR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$	操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R + ACC + C \rightarrow dest$	操作	$ACC + R \rightarrow dest$
状态影响	Z, DC, C	状态影响	Z, DC, C
说明	ACC和R带进位加法: 若d="0", 结果存入ACC; 若d="1", 结果存入"R"。	说明	ACC和R加法: 若d="0", 结果存入ACC; 若d="1", 结果存入"R"。
周期	1	周期	1
举例	ADCAR R, d 执行指令前: ACC=0x12, R=0x34, C=1, d=1, 执行指令后: R=0x47, ACC=0x12, C=0.	举例	ADDR R, d 执行指令前: ACC=0x12, R=0x34, C=1, d=1, 执行指令后: R=0x46, ACC=0x12, C=0.

ADCI A	Add ACC and Immediate with Carry	ADDIA	Add ACC and Immediate
语法	ADCI A i	语法	ADDIA i
操作数	$0 \leq i < 255$	操作数	$0 \leq i < 255$
操作	$ACC + i + C \rightarrow ACC$	操作	$ACC + i \rightarrow ACC$
状态影响	Z, DC, C	状态影响	Z, DC, C
说明	ACC和8位立即数带进位加法, 结果存入ACC。	说明	ACC和8位立即数加法, 结果存入ACC。
周期	1	周期	1
举例	ADCI A i 执行指令前: ACC=0x12, i=0x34, C=1, 执行指令后: ACC=0x47, C=0.	ADDIA i 执行指令前: ACC=0x12, i=0x34, C=1, 执行指令后: ACC=0x46, C=0.	

ANDAR	AND ACC and R	BCR	Clear Bit in R
语法	ANDAR R, d	语法	BCR R, bit
操作数	$0 \leq R \leq 127$ $d = 0, 1.$	操作数	$0 \leq R \leq 127$ $0 \leq bit \leq 7$
操作	$ACC \& R \rightarrow dest$	操作	$0 \rightarrow R[bit]$
状态影响	Z	状态影响	--
说明	ACC和R做“AND”运算；若d=“0”，结果存入ACC；若d=“1”，结果存入“R”	说明	将R寄存器的bit位(0~7) 清 0。 .
周期	1	周期	1
举例	ANDAR R, d 执行指令前： ACC=0x5A, R=0xAF, d=1. 执行指令后： R=0x0A, ACC=0x5A, Z=0.	举例	BCR R, B2 执行指令前： R=0x5A, B2=0x3, 执行指令后： R=0x52.

ANDIA	AND Immediate with ACC	BSR	Set Bit in R
语法	ANDIA i	语法	BSR R, bit
操作数	$0 \leq i < 255$	操作数	$0 \leq R \leq 127$ $0 \leq bit \leq 7$
操作	$ACC \& i \rightarrow ACC$	操作	$1 \rightarrow R[bit]$
状态影响	Z	状态影响	--
说明	ACC和8位立即数 i 做“AND”运算， 结果放入ACC。	说明	将R寄存器的bit位(0~7) 置 1。 .
周期	1	周期	1
举例	ANDIA i 执行指令前： ACC=0x5A, i=0xAF, 执行指令后： ACC=0x0A, Z=0.	举例	BSR R, B2 执行指令前： R=0x5A, B2=0x2, 执行指令后： R=0x5E.

BTRSC	Test Bit in R and Skip if Clear	CALLA	Call Subroutine
语法	BTRSC R, bit	语法	CALLA
操作数	$0 \leq R \leq 127$ $0 \leq \text{bit} \leq 7$	操作数	--
操作	Skip next instruction, if $R[\text{bit}] = 0$ .	操作	$PC + 1 \rightarrow \text{Top of Stack}$ $\{TBHP, ACC\} \rightarrow PC$
状态影响	--	状态影响	--
说明	位判断指令, 为“0”则跳过下一条指令。	说明	子程序调用。首先将返回地址PC+1压入栈顶, 然后将TBHP[2:0]赋值给PC[10:8], 将ACC赋值给PC[7:0]。

周期	1 or 2(跳过)	周期	2
举例	BTRSC R, B2 指令 1 指令 2 执行指令前: $R=0x5A, B2=0x2$ , 执行指令后: 由于 $R[B2]=0$ , 则指令 1 不执行, 程序直接从指令 2 开始执行。	举例	CALLA 执行指令前: $TBHP=0x02, ACC=0x34$ . $PC=A0$ . Stack pointer=1. 执行指令后: $PC=0x234, Stack[1]=A0+1$ , Stack pointer=2

BTRSS	Test Bit in R and Skip if Set	CLRA	Clear ACC
语法	BTRSS R, bit	语法	CLRA
操作数	$0 \leq R \leq 127$ $0 \leq \text{bit} \leq 7$	操作数	--
操作	Skip next instruction, if $R[\text{bit}] = 1$ .	操作	$00h \rightarrow ACC$ $1 \rightarrow Z$
状态影响	--	状态影响	Z
说明	位判断指令, 为“1”则跳过下一条指令。	说明	ACC清零, Z标志位置“1”。
周期	1 or 2(跳过)	周期	1
举例	BTRSS R, B2 指令 2 指令 3 执行指令前: $R=0x5A, B2=0x3$ , 执行指令后: 由于 $R[B2]=1$ , 则指令 2 不执行, 直接从指令 3 开始执行。	举例	CLRA 执行指令前: $ACC=0x55, Z=0$ . 执行指令后: $ACC=0x00, Z=1$ .

CLRR	Clear R	COMR	Complement R
语法	CLRR R	语法	COMR R, d
操作数	$0 \leq R \leq 127$	操作数	$0 \leq R \leq 127$
操作	$00h \rightarrow R$ $1 \rightarrow Z$	操作	$\sim R \rightarrow dest$
状态影响	Z	状态影响	Z
说明	寄存器R清零, Z标志位置“1”。	说明	R寄存器取反, 结果存入d; d=“0”, 结果存入ACC; d=“1”, 结果存入R。
周期	1	周期	1
举例	CLRR R 执行指令前: R=0x55, Z=0. 执行指令后: R=0x00, Z=1.	举例	COMR, d 执行指令前: R=0xA6, d=1, Z=0. 执行指令后: R=0x59, Z=0.

CLRWDT	Clear Watch-Dog Timer	CMPAR	Compare ACC and R
语法	CLRWDT	语法	CMPAR R
操作数	--	操作数	$0 \leq R \leq 127$
操作	$00h \rightarrow WDT$ , $00h \rightarrow WDT prescaler$ $1 \rightarrow /TO$ $1 \rightarrow /PD$	操作	$R - ACC \rightarrow (No restore)$
状态影响	/TO, /PD	状态影响	Z, C
说明	清WDT计数器和预分频器; /TO和/PD标志位置“1”。	说明	ACC和R比较: 执行R-ACC, 不改变ACC和R的值, 只改变Z和C标志位。
周期	1	周期	1
举例	CLRWDT 执行指令前: /TO=0 执行指令后: /TO=1	举例	CMPAR R 执行指令前: R=0x34, ACC=12, Z=0, C=0. 执行指令后: R=0x34, ACC=12, Z=0, C=1.

DAA	Convert ACC Data Format from Hexadecimal to Decimal	DECRSZ	Decrease R, Skip if 0
语法	DAA	语法	DECRSZ R, d
操作数	--	操作数	$0 \leq R \leq 127$
操作	ACC(hex) $\rightarrow$ ACC(decimal)	操作	$d = 0, 1$ .
状态影响	C	状态影响	$R - 1 \rightarrow \text{dest}$ ,
说明	将累加器中的 16 进制数调整为十进制数，该指令必须紧跟在加法指令后。	说明	Skip if result = 0
周期	1	周期	--
举例	DISI ADDAR R,d DAA ENI 执行指令前： ACC=0x28, R=0x25, d=0. 执行指令后： ACC=0x53, C=0.	举例	R 先减 1, 若 d="0", 结果存入 ACC; 若 d="1", 结果存入 R, 若结果为 "0" 则跳过下一条指令, 改为执行 NOP 指令, 因此结果为 "0" 时要执行两个周期。

DECR	Decrease R	DISI	Disable Interrupt Globally
语法	DECR R, d	语法	DISI
操作数	$0 \leq R \leq 127$ $d = 0, 1$ .	操作数	--
操作	$R - 1 \rightarrow \text{dest}$	操作	Disable Interrupt, $0 \rightarrow \text{GIE}$
状态影响	Z	状态影响	--
说明	R 减 1, 若 d="0", 结果存入 ACC; 若 d="1", 结果存入 R。	说明	GIE 设置为 0, 关闭总中断。
周期	1	周期	1
举例	DECR R, d 执行指令前： R=0x01, d=1, Z=0. 执行指令后： R=0x00, Z=1.	举例	DISI 执行指令前： GIE=1, 执行指令后： GIE=0.

ENI	Enable Interrupt Globally	INCR	Increase R
语法	ENI	语法	INCR R, d
操作数	--	操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	Enable Interrupt, $1 \rightarrow GIE$	操作	$R + 1 \rightarrow dest.$
状态影响	--	状态影响	Z
说明	GIE设置为 1, 开启总中断。	说明	R加1, 若d="0", 结果存入ACC; 若d="1", 结果存入R。
周期	1	周期	1
举例	ENI 执行指令前: GIE=0, 执行指令后: GIE=1.	举例	INCR R, d 执行指令前: R=0xFF, d=1, Z=0. 执行指令后: R=0x00, Z=1.

GOTOA	Unconditional Branch	INCRSZ	Increase R, Skip if 0
语法	GOTOA	语法	INCRSZ R, d
操作数	--	操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	{TBHP, ACC} $\rightarrow$ PC	操作	$R + 1 \rightarrow dest,$ Skip if result = 0
状态影响	--	状态影响	--
说明	无条件跳转指令, ACC值写入 PC[7:0] ; TBHP[2:0] 值写入 PC[10:8]。	说明	R先加 1, 若d="0", 结果存入ACC; 若d="1", 结果存入R。若结果为"0" 则跳过下一条指令 (执行NOP指令)
周期	2	周期	1 or 2(跳过)
举例	GOTOA 执行指令前: PC=A0. TBHP=0x02, ACC=0x34. 执行指令后: PC=0x234	举例	INCRSZ R, d 指令 2, 指令 3. 执行指令前: R=0xFF, d=1, Z=0. 执行指令后: R=0x00, Z=1, 因结果为 0, 程序 跳过指令 2。

INT	Software Interrupt	IORIA	OR Immediate with ACC
语法	INT	语法	IORIA i
操作数	--	操作数	$0 \leq i < 255$
操作	PC + 1 → Top of Stack, 001h → PC	操作	ACC   i → ACC
状态影响	--	状态影响	Z
说明	软中断指令。首先将返回地址 (PC+1) 压入栈顶, 然后将 001H 的地址装入PC[10:0]。	说明	ACC和 8 位立即数做“OR”运算, 结 果存入ACC。
周期	3	周期	1
举例	<p>INT</p> <p>执行指令前: PC=address of INT code</p> <p>执行指令后: PC=0x01</p>	举例	<p>IORIA i</p> <p>执行指令前: i=0x50, ACC=0xAA, Z=0.</p> <p>执行指令后: ACC=0xFA, Z=0.</p>
IORAR	OR ACC with R	IOST	Load F-page SFR from ACC
语法	IORAR R, d	语法	IOST F
操作数	$0 \leq R \leq 127$ $d = 0, 1.$	操作数	$5 \leq F \leq 15$
操作	ACC   R → dest	操作	ACC → F-page SFR
状态影响	Z	状态影响	--
说明	ACC和R做“OR”运算, 若d=“0”, 结 果存入ACC; 若d=“1”, 结果存入R。	说明	将ACC的值赋给F-page特殊寄存器
周期	1	周期	1
举例	<p>IORAR R, d</p> <p>执行指令前: R=0x50, ACC=0xAA, d=1, Z=0.</p> <p>执行指令后: R=0xFA, ACC=0xAA, Z=0.</p>	举例	<p>IOST F</p> <p>执行指令前: F=0x55, ACC=0xAA.</p> <p>执行指令后: F=0xAA, ACC=0xAA.</p>

IOSTR	Move F-page SFR to ACC	LGOTO	Unconditional Branch
语法	IOSTR F	语法	LGOTO adr
操作数	$5 \leq F \leq 15$	操作数	$0 \leq adr \leq 2047$
操作	F-page SFR $\rightarrow$ ACC	操作	$adr \rightarrow PC[10:0]$ .
状态影响	--	状态影响	--
说明	将F-page特殊寄存器数值给ACC。	说明	无条件长跳转，11位立即数写入PC[10:0]。
周期	1	周期	2
举例	IOSTR F 执行指令前: F=0x55, ACC=0xAA. 执行指令后: F=0x55, ACC=0x55.	举例 LGOTO Level 执行指令前: PC=A0. 执行指令后: PC=address of Level.	

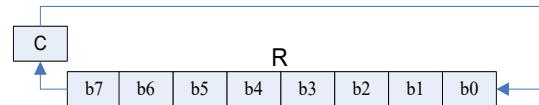
LCALL	Call Subroutine	MOVAR	Move ACC to R
语法	LCALL adr	语法	MOVAR R
操作数	$0 \leq adr \leq 2047$	操作数	$0 \leq R \leq 127$
操作	$PC + 1 \rightarrow$ Top of Stack, $adr \rightarrow PC[10:0]$	操作	$ACC \rightarrow R$
状态影响	--	状态影响	--
说明	长调用子程序。首先将PC+1压入栈顶，然后将11位立即数载入PC[10:0]。	说明	ACC赋值给R-page寄存器
周期	2	周期	1
举例	LCALL SUB 执行指令前: PC=A0. Stack level=1 执行指令后: PC=address of SUB, Stack[1]= A0+1, Stack pointer =2.	举例 MOVAR R 执行指令前: R=0x55, ACC=0xAA. 执行指令后: R=0xAA, ACC=0xAA.	

MOVIA	Move Immediate to ACC	NOP	No Operation
语法	MOVIA i	语法	NOP
操作数	$0 \leq i < 255$	操作数	--
操作	$i \rightarrow ACC$	操作	No operation.
状态影响	--	状态影响	--
说明	8位立即数赋值给ACC。	说明	空操作。
周期	1	周期	1
举例	MOVIA i 执行指令前: $i=0x55, ACC=0xAA.$ 执行指令后: $ACC=0x55.$	举例 执行指令前: $PC=A0$ 执行指令后: $PC=A0+1$	
MOVR	Move R to ACC or R	RETIE	Return from Interrupt and Enable Interrupt Globally
语法	MOVR R, d	语法	RETIE
操作数	$0 \leq R \leq 127$	操作数	--
操作	$d = 0, 1.$	操作	Top of Stack $\rightarrow PC$
状态影响	$R \rightarrow dest$	状态影响	$1 \rightarrow GIE$
说明	Z	说明	--
	$R-page$ 寄存器赋值给d, 若d="0", 结果存入ACC; 若d="1", 结果存入寄存器R。指令执行后, 通过状态标志Z检查R是否为0。		中断返回, 栈顶地址载入PC同时使能中断。
周期	1	周期	2
举例	MOVR R, d 执行指令前: $R=0x0, ACC=0xAA, Z=0, d=0.$ 执行指令后: $R=0x0, ACC=0x00, Z=1.$	举例 执行指令前: $GIE=0, Stack\ level=2.$ 执行指令后: $GIE=1, PC=Stack[2], Stack\ pointer=1.$	

RETIA	Return with Data in ACC
语法	RETIA i
操作数	$0 \leq i < 255$
操作	$i \rightarrow ACC$ , Top of Stack $\rightarrow PC$
状态影响	--
说明	带参数返回：8位立即数赋值给ACC，栈顶地址载入PC。
周期	2
举例	<p>RETIA i</p> <p>执行指令前： Stack pointer = 2, i=0x55, ACC=0xAA.</p> <p>执行指令后： PC=Stack[2], Stack pointer = 1. ACC=0x55.</p>

RLR	Rotate Left R Through Carry
语法	RLR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$C \rightarrow dest[0]$ , $R[7] \rightarrow C$ , $R[6:0] \rightarrow dest[7:1]$
状态影响	C
说明	带进位R循环左移：若d=“0”，结果存入ACC；若d=“1”，结果存入R。
周期	1
举例	<p>RLR R, d</p> <p>执行指令前： R=0xA5, d=1, C=0.</p> <p>执行指令后： R=0x4A, C=1.</p>

RET	Return from Subroutine	RRR	Rotate Right R Through Carry
语法	RET	语法	RRR R, d
操作数	--	操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	Top of Stack $\rightarrow PC$	操作	$C \rightarrow dest[7]$ , $R[7:1] \rightarrow dest[6:0]$ , $R[0] \rightarrow C$
状态影响	--	状态影响	C
说明	子程序返回，栈顶载入PC。	说明	带进位R循环右移：若d=“0”，结果存入ACC；若d=“1”，结果存入R。
周期	2	周期	1
举例	<p>RET</p> <p>执行指令前： Stack level=2.</p> <p>执行指令后： PC=Stack[2], Stack level=1.</p>	<p>RRR R, d</p> <p>执行指令前： R=0xA5, d=1, C=0.</p> <p>执行指令后： R=0x52, C=1.</p>	



SBCAR	Subtract ACC and Carry from R	SBCIA	Subtract ACC and Carry from Immediate
语法	SBCAR R, d	语法	SBCIA i
操作数	$0 \leq R \leq 127$ $d = 0, 1.$	操作数	$0 \leq i < 255$
操作	$R + (\sim ACC) + C \rightarrow dest$	操作	$i + (\sim ACC) + C \rightarrow dest$
状态影响	Z, DC, C	状态影响	Z, DC, C
说明	R和ACC带借位减法, 若d="0", 结果存入ACC; 若d="1", 结果存入R。	说明	常数和ACC带借位减法, 结果存入ACC。

周期	1	周期	1
举例	<p>SBCAR R, d</p> <p>(a) 执行指令前: R=0x05, ACC=0x06, d=1, C=0,</p> <p>执行指令后: R=0xFE, C=0. (-2)</p> <p>(b) 执行指令前: R=0x05, ACC=0x06, d=1, C=1,</p> <p>执行指令后: R=0xFF, C=0. (-1)</p> <p>(c) 执行指令前: R=0x06, ACC=0x05, d=1, C=0,</p> <p>执行指令后: R=0x00, C=1. (-0), Z=1.</p> <p>(d) 执行指令前: R=0x06, ACC=0x05, d=1, C=1,</p> <p>执行指令后: R=0x1, C=1. (+1)</p>	<p>SBCIA i</p> <p>(a) 执行指令前: i=0x05, ACC=0x06, C=0, 执行指令后: ACC=0xFE, C=0. (-2)</p> <p>(b) 执行指令前: i=0x05, ACC=0x06, C=1, 执行指令后: ACC=0xFF, C=0. (-1)</p> <p>(c) 执行指令前: i=0x06, ACC=0x05, C=0, 执行指令后: ACC=0x00, C=1. (-0), Z=1.</p> <p>(d) 执行指令前: i=0x06, ACC=0x05, C=1, 执行指令后: ACC=0x1, C=1. (+1)</p>	<p>SBCIA i</p> <p>(a) 执行指令前: i=0x05, ACC=0x06, C=0, 执行指令后: ACC=0xFE, C=0. (-2)</p> <p>(b) 执行指令前: i=0x05, ACC=0x06, C=1, 执行指令后: ACC=0xFF, C=0. (-1)</p> <p>(c) 执行指令前: i=0x06, ACC=0x05, C=0, 执行指令后: ACC=0x00, C=1. (-0), Z=1.</p> <p>(d) 执行指令前: i=0x06, ACC=0x05, C=1, 执行指令后: ACC=0x1, C=1. (+1)</p>

SFUN	Load S-page SFR from ACC
语法	SFUN S
操作数	$0 \leq S \leq 21$
操作	$ACC \rightarrow S\text{-page SFR}$
状态影响	--
说明	ACC写到S-page特殊寄存器
周期	1
举例	<p>SFUN S</p> <p>执行指令前: S=0x55, ACC=0xAA.</p> <p>执行指令后: S=0xAA, ACC=0xAA.</p>

SFUNR	Move S-page SFR to ACC	SUBAR	Subtract ACC from R
语法	SFUNR S	语法	SUBAR R, d
操作数	$0 \leq S \leq 21$	操作数	$0 \leq R \leq 127$
操作	S-page SFR $\rightarrow$ ACC	操作	$d = 0, 1.$
状态影响	--	状态影响	$R - ACC \rightarrow dest$
说明	读S-page特殊寄存器到ACC	说明	$Z, DC, C$
周期	1	说明	$R$ 减去ACC, 若 $d=0$ , 结果存入ACC。 若 $d=1$ , 结果存入R。
举例	SFUNR S 执行指令前: $S=0x55, ACC=0xAA.$ 执行指令后: $S=0x55, ACC=0x55.$	周期	1
		举例	SUBAR R, d (a) 执行指令前: $R=0x05, ACC=0x06, d=1,$ 执行指令后: $R=0xFF, C=0. (-1)$ (b) 执行指令前: $R=0x06, ACC=0x05, d=1,$ 执行指令后: $R=0x01, C=1. (+1)$

SLEEP	Enter Halt Mode	SUBIA	Subtract ACC from Immediate
语法	SLEEP	语法	SUBIA i
操作数	--	操作数	$0 \leq i < 255$
操作	$00h \rightarrow WDT,$ $00h \rightarrow WDT prescaler$ $1 \rightarrow /TO$ $0 \rightarrow /PD$	操作	$i - ACC \rightarrow ACC$
状态影响	/TO, /PD	状态影响	$Z, DC, C$
说明	WDT和分频器 0 清零。/TO标志为0, /PD清零, IC进入睡眠。	说明	8位立即数减ACC, 结果存入ACC。
周期	1	周期	1
举例	SLEEP 执行指令前: $/PD=1, /TO=0.$ 执行指令后: $/PD=0, /TO=1.$	举例	SUBIA i (a) 执行指令前: $i=0x05, ACC=0x06.$ 执行指令后: $ACC=0xFF, C=0. (-1)$ (b) 执行指令前: $i=0x06, ACC=0x05, d=1,$ 执行指令后: $ACC=0x01, C=1. (+1)$

SWAPR	Swap High/Low Nibble in R	T0MD	Load ACC to T0MD
语法	SWAPR R, d	语法	T0MD
操作数	0 ≤ R ≤ 127 d = 0, 1.	操作数	--
操作	R[3:0] → dest[7:4]. R[7:4] → dest[3:0]	操作	ACC → T0MD
状态影响	--	状态影响	--
说明	寄存器半字节交换, 若d="0", 结果存入ACC; 若d="1", 结果存入R。	说明	ACC写入T0MD寄存器。
周期	1	周期	1
举例	SWAPR R, d 执行指令前: R=0xA5, d=1. 执行指令后: R=0x5A.	举例	T0MD 执行指令前: T0MD=0x55, ACC=0xAA. 执行指令后: T0MD=0xAA.

TABLEA	Read ROM data	T0MDR	Move T0MD to ACC
语法	TABLEA	语法	T0MDR
操作数	--	操作数	--
操作	ROM data{ TBHP, ACC } [7:0] →ACC ROM data{TBHP, ACC} [13:8] →TBHD.	操作	T0MD → ACC
状态影响	--	状态影响	--
说明	ROM查表指令, 高字节存入TBHD, 低字节存入ACC。	说明	读T0MD寄存器到ACC.
周期	2	周期	1
举例	TABLEA 执行指令前: TBHP=0x02, CC=0x34. TBHD=0x01. ROM data[0x234]= 0x35AA 执行指令后: TBHD=0x35, ACC=0xAA.	举例	T0MDR 执行指令前: T0MD=0x55, ACC=0xAA. 执行指令后: ACC=0x55.

**XORAR**      **Exclusive-OR ACC with R**

语法	XORAR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$ACC \oplus R \rightarrow dest$
状态影响	Z
说明	ACC和R做“XOR”运算，若d=“0”，结果存入ACC；若d=“1”，结果存入R。
周期	1
举例	XORAR R, d 执行指令前： R=0xA5, ACC=0xF0, d=1. 执行指令后： R=0x55.

**XORIA**      **Exclusive-OR Immediate with ACC**

语法	XORIA i
操作数	$0 \leq i < 255$
操作	$ACC \oplus i \rightarrow ACC$
状态影响	Z
说明	ACC和8位立即数做“XOR”运算。
周期	1
举例	XORIA i 执行指令前： i=0xA5, ACC=0xF0. 执行指令后： ACC=0x55.

## 5. 配置表

项目	名称	选项					
1	High Oscillator Frequency 高频振荡模式	1. I_HRC	2. E_HXT	3. E_XT			
2	Low Oscillator Frequency 低频振荡模式	1. I_LRC	2. E_LXT				
3	High IRC Frequency 内部高速 RC 频率	1. 1MHz 4. 8MHz	2. 2MHz 5. 16MHz	3. 4MHz 6. 20MHz			
4	High Crystal Oscillator 外部高频振荡器	1. 8MHz > F <sub>HOSC</sub> > 6MHz 3. 12MHz > F <sub>HOSC</sub> ≥ 10MHz 5. 20MHz > F <sub>HOSC</sub> ≥ 16MHz	2. 10MHz > F <sub>HOSC</sub> ≥ 8MHz 4. 16MHz > F <sub>HOSC</sub> ≥ 12MHz 6. 20MHz				
5	Instruction Clock 指令时钟	1. 4 oscillator period 4 个振荡周期	2. 2 oscillator period 2 个振荡周期				
6	WDT 看门狗定时器	1. Watchdog Enable (Software control) 看门狗开启 (软件控制) 2. Watchdog Disable (Always disable) 看门狗关闭 (永远关闭)					
7	WDT Event 看门狗定时器事件	1. Watchdog Reset 看门狗复位	2. Watchdog Interrupt 看门狗中断				
8	Timer0 Source 定时器 0 时钟源	1. EX_CKIO	2. Low Oscillator (I_LRC/E_LXT) 低频振荡器 (I_LRC/E_LXT)				
9	PA.5	1. PA5 is I/O PA5 是 I/O。	2. PA5 is reset PA5 是复位。				
10	PA.7	1. PA7 is I/O PA7 是 I/O。	2. PA7 is instruction clock output PA.7 是指令时钟输出。				
11	Startup Time 上电复位时间	1. 140us 2. 4.5ms 3. 18ms 4. 72ms 5. 288ms					
12	WDT Time Base 看门狗定时器时基	1. 3.5ms 2. 15ms 3. 60ms 4. 250ms					
13	LVR Setting LVR 开关设定	1. Register Control 寄存器控制	2. LVR Always On LVR 永远开启				
14	LVR Voltage LVR 电压	1. 1.8V 6. 3.0V 7. 3.3V 8. 3.6V	2. 2.0V 3. 2.2V 4. 2.4V 5. 2.7V				
15	VDD Voltage VDD 电压	1. 3.0V 2. 4.5V 3. 5.0V					
16	Sink current (exclude PA5) 灌电流 (包括 PA5)	1. Large 大灌电流	2. Normal 一般灌电流				
17	Comparator Input pin select 比较器输入引脚选择	1. Enable 开启	2. Disable 关闭				
18	Read Output Data 读取输出口数据	1. I/O Port I/O 口	2. Register 寄存器				
19	E_LXT Backup Control E_LXT 强化起振开关设定	1. Auto Off 自动关闭	2. Register Off 寄存器关闭				
20	EX_CKIO to Inst. Clock EX_CKIO 到指令时钟	1. Sync 同步	2. Async 异步				
21	Startup Clock 上电时钟源	1. Fast (I_HRC/E_HXT/E_XT) 高速	2. Slow (I_LRC/E_LXT) 低速				
22	Input Schmitt Trigger 输入施密特触发	1. Enable 开启	2. Disable (0.5VDD) 关闭				

项目	名称	选项		
23	Input High Voltage (VIH) 输入高电压 (VIH)	1. 0.7VDD	2. 0.5VDD	
24	Input Low Voltage (VIL) 输入低电压 (VIL)	1. 0.3VDD	2. 0.2VDD	
25	PWM2 output PWM2 输出	1. PA4	2. PB2	
26	PWM4 output PWM4 输出	1. PA3	2. PA7	3. Disable
27	PWM5 output PWM5 输出	1. PB3	2. PB0	3. Disable

表 37 配置表

## 6. 电气特性

### 6.1 最大绝对值

符号	参数	额定值	单位
$V_{DD} - V_{SS}$	工作电压	-0.5 ~ +6.0	V
$V_{IN}$	输入电压	$V_{SS}-0.3V \sim V_{DD}+0.3$	V
$T_{OP}$	工作温度	-40 ~ +85	°C
$T_{ST}$	储存温度	-40 ~ +125	°C

### 6.2 直流电气特性

(所有参考  $F_{INST}=F_{HOSC}/4$ ,  $F_{HOSC}=16MHz$  或  $20MHz$  @  $I_{HRC}$ , WDT 开启, 环境温度  $T_A=25^\circ C$ , 除其他指定说明外。)

符号	参数	$V_{DD}$	最小值	典型值	最大值	单位	条件
$V_{DD}$	工作电压	--	3.0	5.5	V		$F_{INST}=10MHz @ I_{HRC} 20MHz/2T$
			2.2				$F_{INST}=5MHz @ I_{HRC} 20MHz/4T$
			2.7				$F_{INST}=8MHz @ I_{HRC} 16MHz/2T$
			2.0				$F_{INST}=4MHz @ I_{HRC} 16MHz/4T$
			1.8				$F_{INST}=4MHz @ E_{HXT} 8MHz/2T$
			1.6				$F_{INST}=2MHz @ I_{HRC} 8MHz/4T$
			1.6				$F_{INST}=2MHz @ I_{HRC} 4MHz/2T$
							$F_{INST}=1MHz @ I_{HRC} 4MHz/4T$
							$F_{INST}=8KHz @ I_{LRC} 32KHz/4T$
							$F_{INST}=8KHz @ E_{LXT} 32KHz/4T$
$V_{IH}$	输入高电压	5V	4.1	--	--	V	$RSTb (0.8 V_{DD})$
		3V	2.4	--	--		
		5V	3.5	--	--	V	All other I/O pins, EX_CK10/1, INT0/1 (0.7 $V_{DD}$ )
		3V	2.1	--	--		
		5V	2.5	--	--	V	All other I/O pins, EX_CK10/1 (0.5 $V_{DD}$ )
		3V	1.5	--	--		
$V_{IL}$	输入低电压	5V	--	--	1.0	V	$RSTb (0.2 V_{DD})$
		3V	--	--	0.6		
		5V	--	--	1.5	V	All other I/O pins, EX_CK10/1, INT0/1 (0.3 $V_{DD}$ )
		3V	--	--	0.9		
		5V	--	--	1.1	V	All other I/O pins, EX_CK10/1 (0.2 $V_{DD}$ )
		3V	--	--	0.6		
$I_{OH}$	输出高推电流	5V	--	18	--	mA	$V_{OH}=4.0V$
		3V	--	10	--		$V_{OH}=2.0V$
$I_{OL}$	输出低灌电流 (大电流)	5V	--	40	--	mA	$V_{OL}=1.0V$
		3V	--	25	--		
$I_{OL}$	输出低灌电流 (一般电流)	5V	--	28	--	mA	$V_{OL}=1.0V$
		3V	--	17	--		
$I_{IR}$	IR灌电流	5V	--	43	--	mA	$V_{OL}=1.0V$
		3V	--	28	--		

符号	参数	V <sub>DD</sub>	最小值	典型值	最大值	单位	条件
I <sub>OP</sub>	工作电流	正常模式					
		5V	--	1.7	--	mA	$F_{INST} = 10MHz$ @I_HRC&E_HXT 20MHz/2T
		3V	--	0.7	--	mA	$F_{INST} = 5MHz/4T$ @I_HRC&E_HXT 20MHz/4T
		5V	--	1.4	--	mA	$F_{INST} = 8MHz$ @I_HRC & E_HXT 16MHz/2T
		3V	--	0.5	--	mA	$F_{INST} = 4MHz$ @I_HRC & E_HXT 16MHz/4T
		5V	--	1.6	--	mA	$F_{INST} = 2MHz$ @I_HRC& E_HXT 8MHz/4T
		3V	--	0.6	--	mA	$F_{INST} = 1MHz$ @I_HRC & E_XT 4MHz/4T
		5V	--	1.3	--	mA	$F_{INST} = 32KHz/2T$ (I_LRC)
		3V	--	0.5	--	mA	$F_{INST} = 32KHz/4T$ (E_LXT)
		5V	--	1.1	--	mA	$F_{INST} = 32KHz/4T$ (I_LRC)
		3V	--	0.4	--	mA	$F_{INST} = 32KHz/4T$ (E_LXT)
慢速模式							
I <sub>STB</sub>	待机电流	5V	--	11	--	uA	$F_{HOSC}$ 关闭 $F_{LOSC}=32KHz/2T$ (I_LRC)
		3V	--	6.1	--	uA	$F_{HOSC}$ 关闭 $F_{LOSC}=32KHz/4T$ (E_LXT)
I <sub>HALT</sub>	睡眠电流	5V	--	11	--	uA	$F_{HOSC}$ 关闭 $F_{LOSC}=32KHz/4T$ (I_LRC)
		3V	--	4.9	--	uA	$F_{HOSC}$ 关闭 $F_{LOSC}=32KHz/4T$ (E_LXT)
		5V	--	7.3	--	uA	$F_{HOSC}$ 关闭 $F_{LOSC}=32KHz/4T$ (I_LRC)
		3V	--	4.3	--	uA	$F_{HOSC}$ 关闭 $F_{LOSC}=32KHz/4T$ (E_LXT)
R <sub>PH</sub>	上拉电阻	5V	--	3.8	--	uA	待机模式, $F_{Hosc}$ 关闭 $F_{LOSC}=32KHz/4T$ (I_LRC)
		3V	--	2.6	--	uA	
		5V	--	--	0.5	uA	睡眠模式 WDT关闭
		3V	--	--	0.2	uA	
		5V	--	--	5.0	uA	睡眠模式 WDT关闭
		3V	--	--	3.0	uA	
R <sub>PL</sub>	下拉电阻	5V	--	50	--	KΩ	上拉电阻(包括PA4)
		3V	--	100	--	KΩ	
		5V	--	85	--	KΩ	上拉电阻(PA5)
		3V	--	85	--	MΩ	
		5V		0.6		MΩ	上拉电阻 (PA4 1M+HiLH)
		3V		1		MΩ	

### 6.3 OSC电气特性

(测量条件VDD电压, TA 温度等于编程条件)

参数	最小值	典型值	最大值	单位	条件
烧录座的 I_HRC 偏差			±1	%	直接将烧录座装在Writer上。
烧录机台的 I_HRC 偏差			±3	%	正确设置烧录机台。
烧录机台的 I_LRC 偏差			±5	%	

### 6.4 比较器/LVD电气特性

(VDD=5V, Vss=0V, TA=25°C 除其他指定说明外)

符号	参数	最小值	典型值	最大值	单位	条件
V <sub>IVR</sub>	比较器输入电压范围	0	--	5	V	F <sub>HOSC</sub> =1MHz
T <sub>ENO</sub>	比较器启动等待时间	--	20	--	us	F <sub>HOSC</sub> =1MHz
I <sub>CO</sub>	比较器电流消耗	--	135	--	uA	F <sub>HOSC</sub> =1MHz, P2V mode
I <sub>LVD</sub>	LVD电流消耗	--	150	--	uA	F <sub>HOSC</sub> =1MHz, LVD=4.3V
E <sub>LVD</sub>	LVD电压误差	--	3	--	%	F <sub>HOSC</sub> =1MHz, LVD=4.3V

\*注：这些参数仅供设计参考，并非针对每个芯片进行测试。

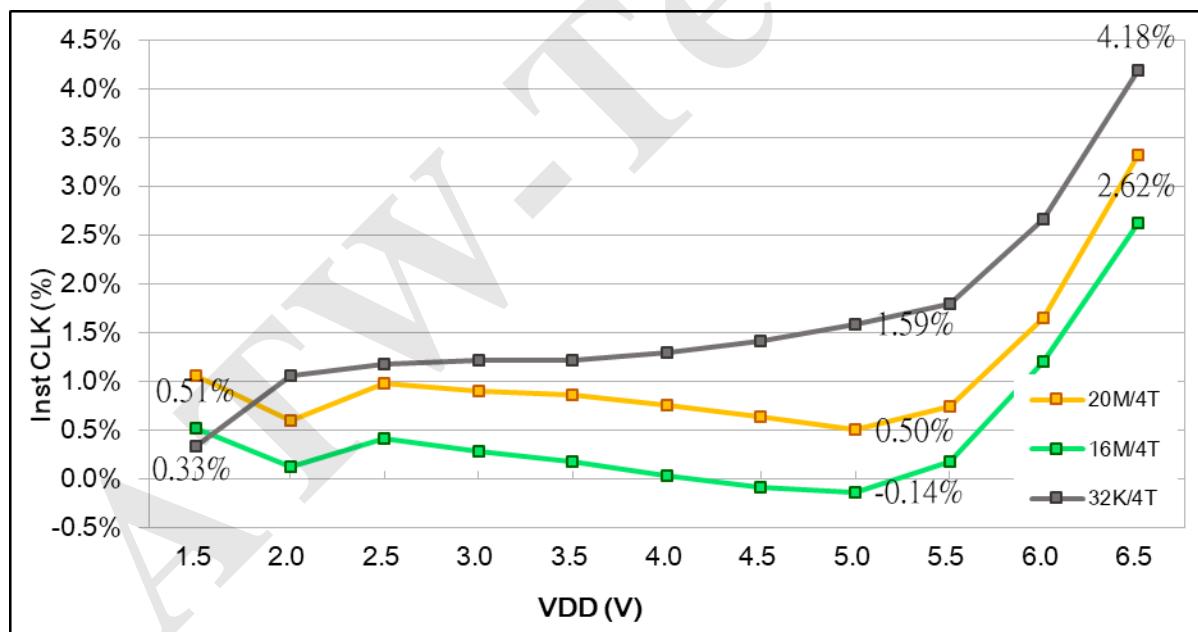
### 6.5 ADC电气特性

(VDD=5V, Vss=0V, TA=25°C 除其他指定说明外)

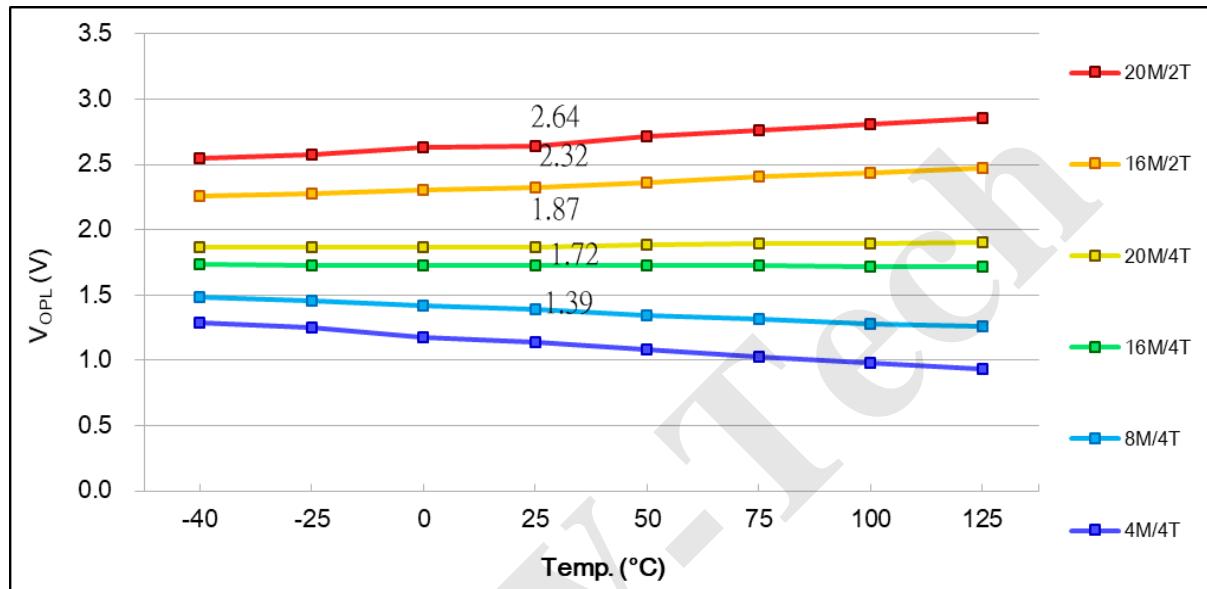
符号	参数	最小值	典型值	最大值	单位	条件
V <sub>REFH</sub>	模拟参考电压范围	2V	--	V <sub>DD</sub>	V	外部参考电压
V <sub>REF4</sub>	内部参考电压 4V, V <sub>DD</sub> =5V	3.96	4	4.04	V	
V <sub>REF3</sub>	内部参考电压 3V, V <sub>DD</sub> =5V	2.97	3	3.03	V	
V <sub>REF2</sub>	内部参考电压 2V, V <sub>DD</sub> =5V	1.98	2	2.02	V	
V <sub>REF</sub>	内部参考电压V <sub>DD</sub> , V <sub>DD</sub> =5V	--	V <sub>DD</sub>	--	V	
	内部参考电压	V <sub>REF</sub> +0.5	--	--	V	最小供电电压
	ADC 模拟输入电压	0	--	V <sub>REFH</sub>	V	
I <sub>OP(ADC)</sub>	ADC 电流消耗	--	0.5	--	mA	
ADCLK	ADC 时钟频率	32K	--	1M	Hz	
ADCYCLE	ADC 转换时间周期	16	--		1/ADCLK	SHCLK=2 ADC clock
ADC <sub>sample</sub>	ADC 转换率	--	--	125	K/sec	V <sub>DD</sub> =5V
DNL	非线性微分误差	±1	--	--	LSB	V <sub>DD</sub> =5.0V, AVREFH=5V, FADSM <sub>P</sub> =62.5K
INL	非线性积分误差	±2	--	--	LSB	
NMC	无缺码分辨率	10	11	12	Bits	

## 6.6 特性曲线图

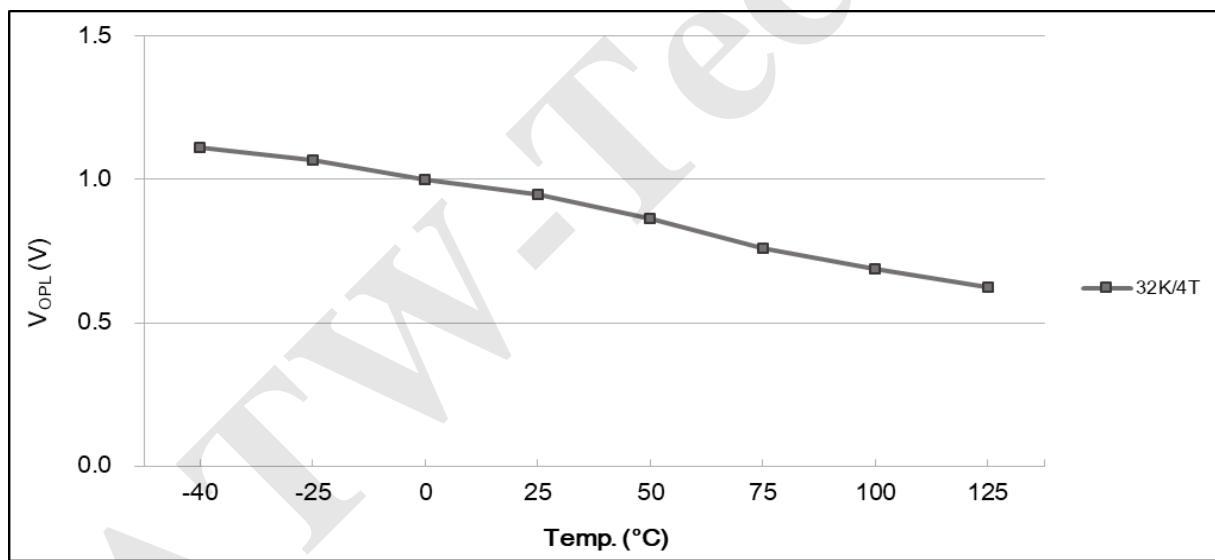
### 6.6.1 高速 RC 振荡频率(I\_HRC)及低速 RC 振荡频率(I\_LRC)与电源电压(VDD)曲线图



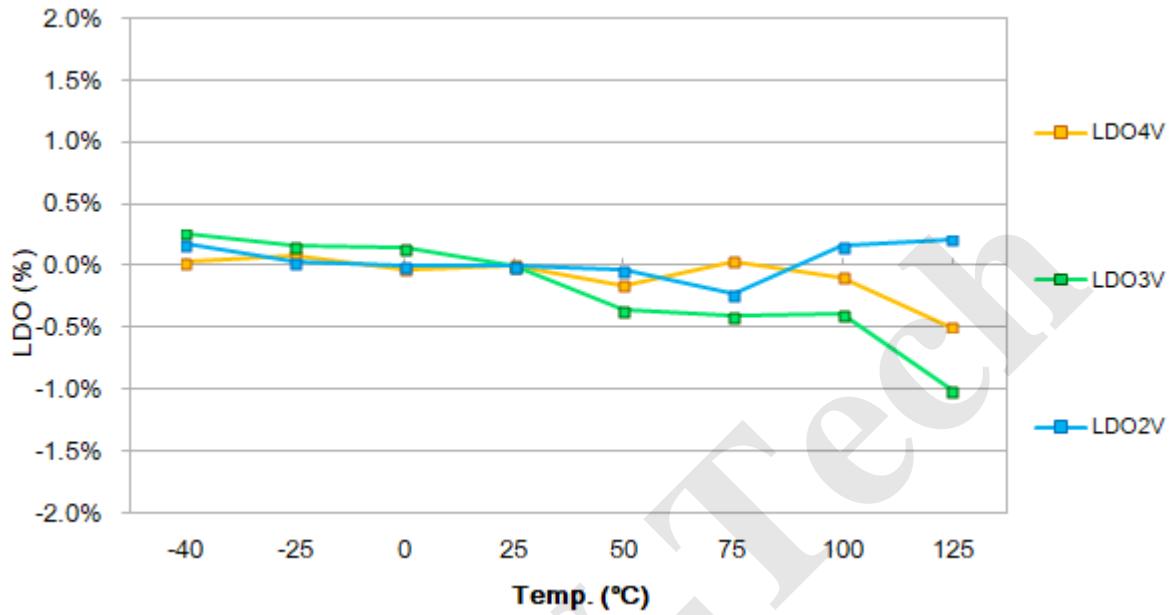
### 6.6.2 高速 RC 振荡频率(I\_HRC)与温度曲线图



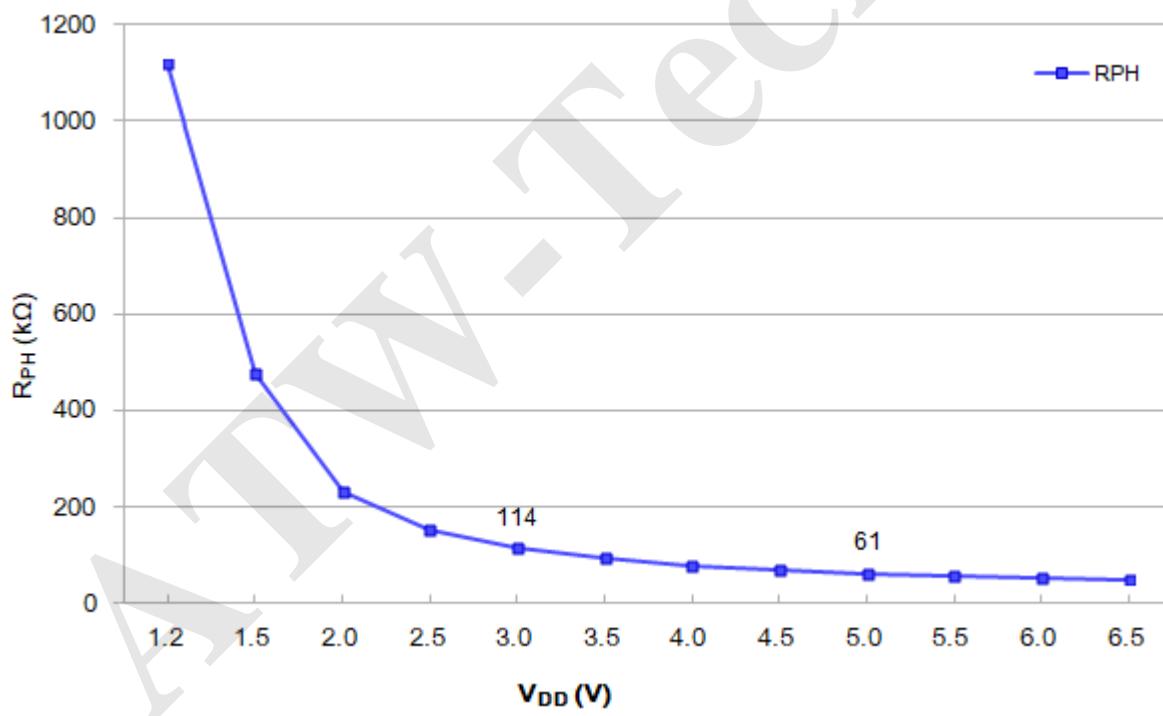
### 6.6.3 低速 RC 振荡频率(I\_LRC)与温度曲线图



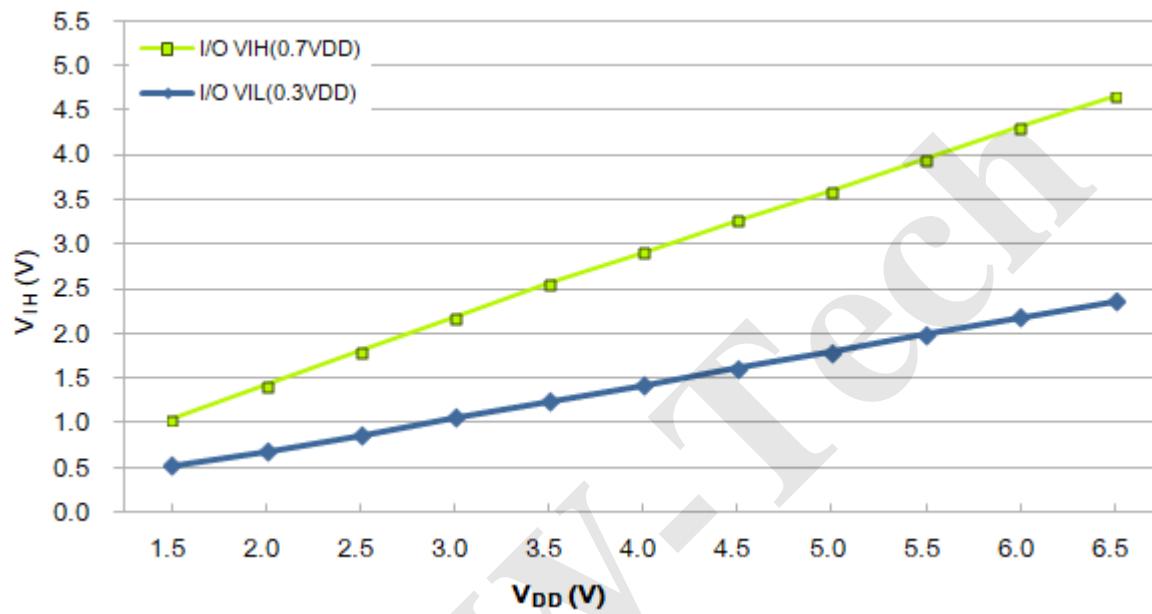
### 6.6.4 内部参考电压 LDO 与温度曲线图

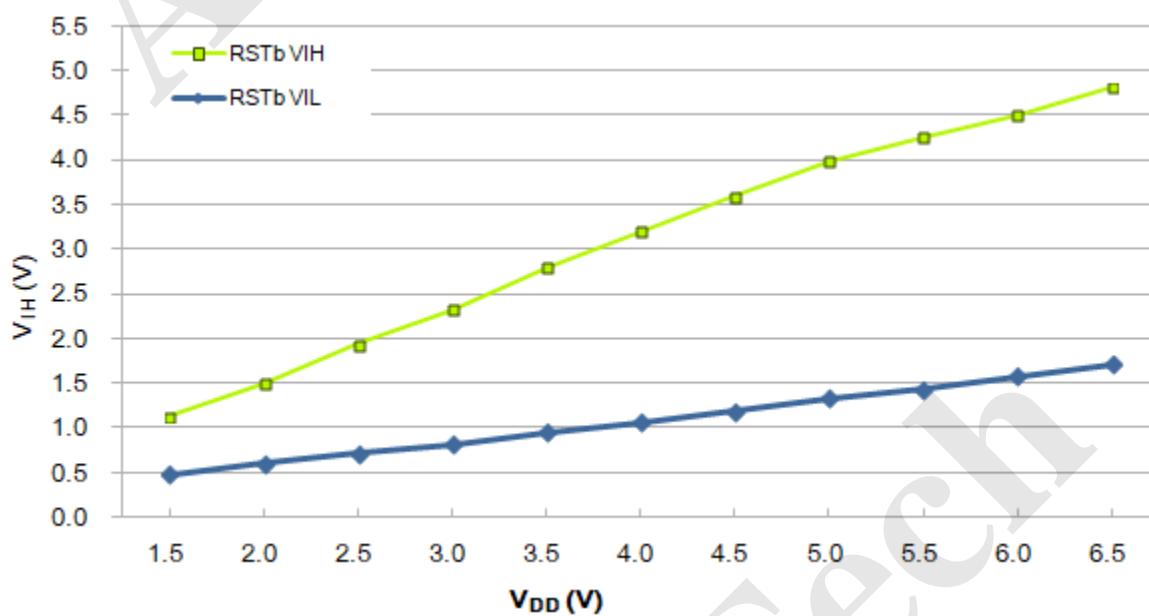
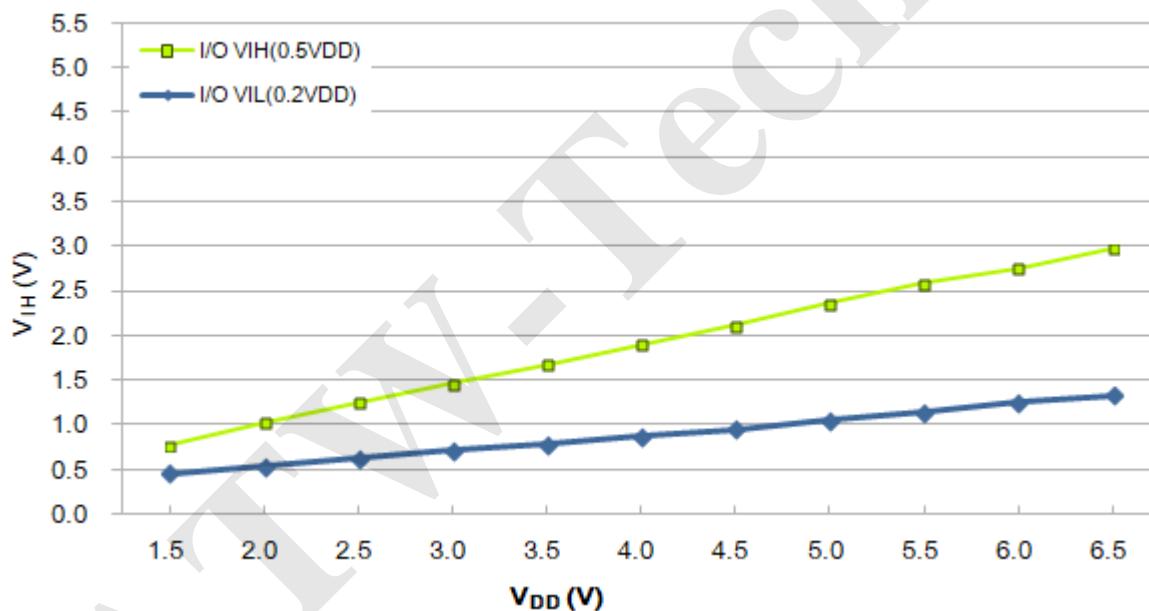


## 6.6.5 内部上拉电阻与电源电压(VDD)曲线图



## 6.6.6 VIH/VIL 与电源电压(VDD)曲线图





## 6.7 建议工作电压

频率	LVR 默认设置	LVR 设置 (最小值 @ 25°C)
20M/2T	3.6V	3.3V
16M/2T	3.3V	3.0V
20M/4T	2.4V	2.2V
16M/4T	2.2V	2.0V
8M/4T	1.8V	1.8V
4M/4T	1.8V	1.8V

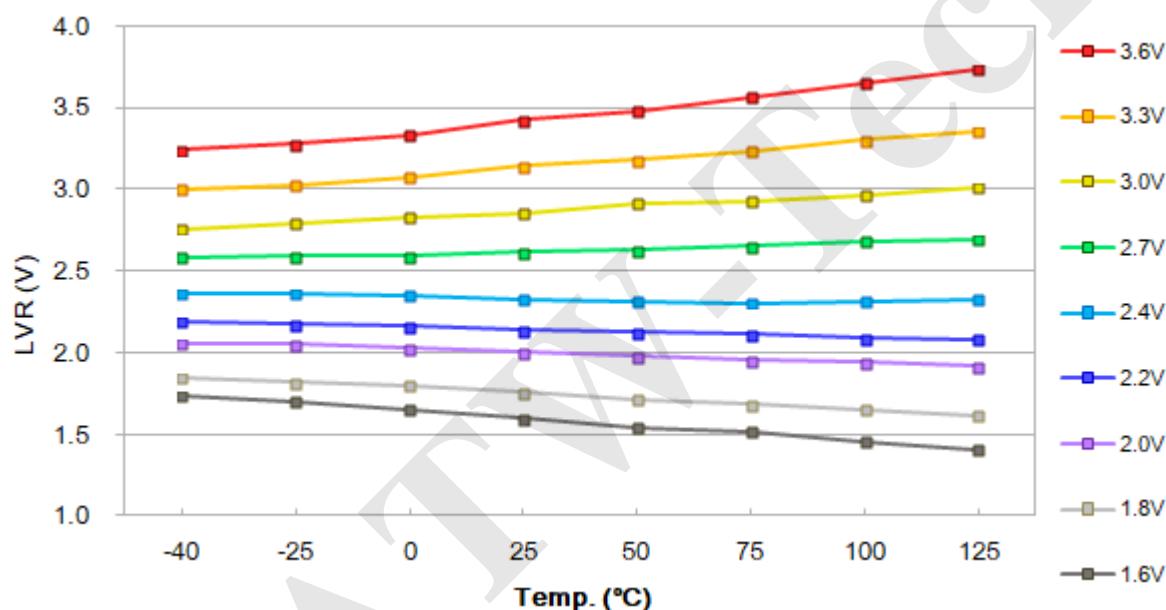
\*注：这些参数仅供设计参考，并非针对每个芯片进行测试。

### 旁路预分频器

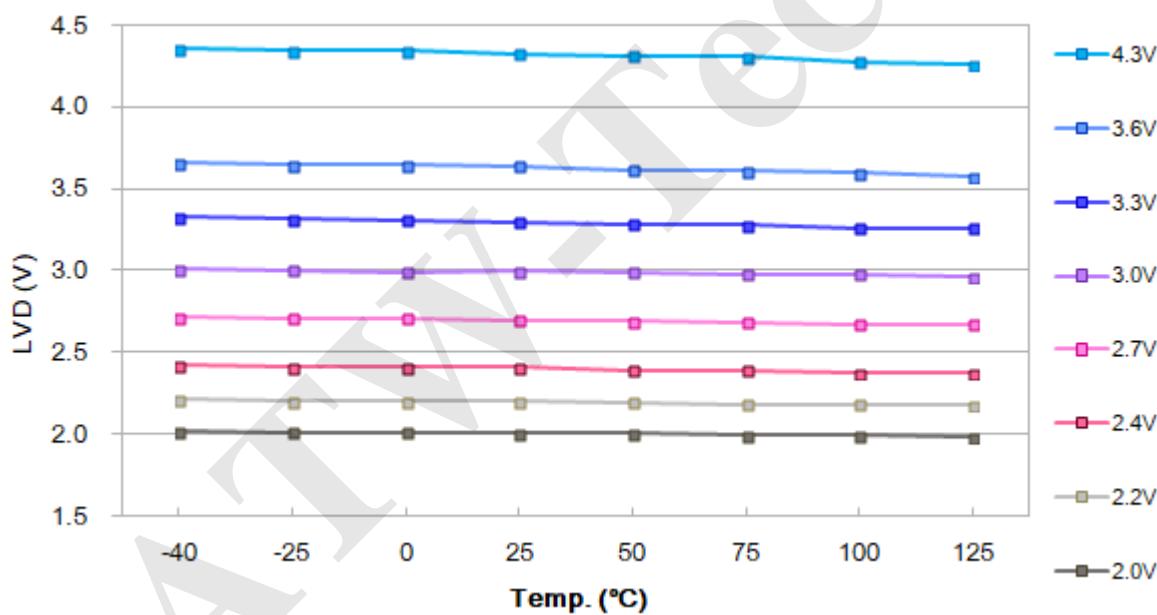
频率	LVR : 建议旁路预分频器
20M/2T	2.7V @ I_HRC=20MHz
16M/2T	2.7V @ I_HRC=16MHz
20M/4T	2.7V @ I_HRC=20MHz
16M/4T	2.4V @ I_HRC=16MHz
8M/4T	2.4V @ I_HRC=8MHz
4M/4T	2.4V @ I_HRC=4MHz

\*注：这些参数仅供设计参考，并非针对每个芯片进行测试。

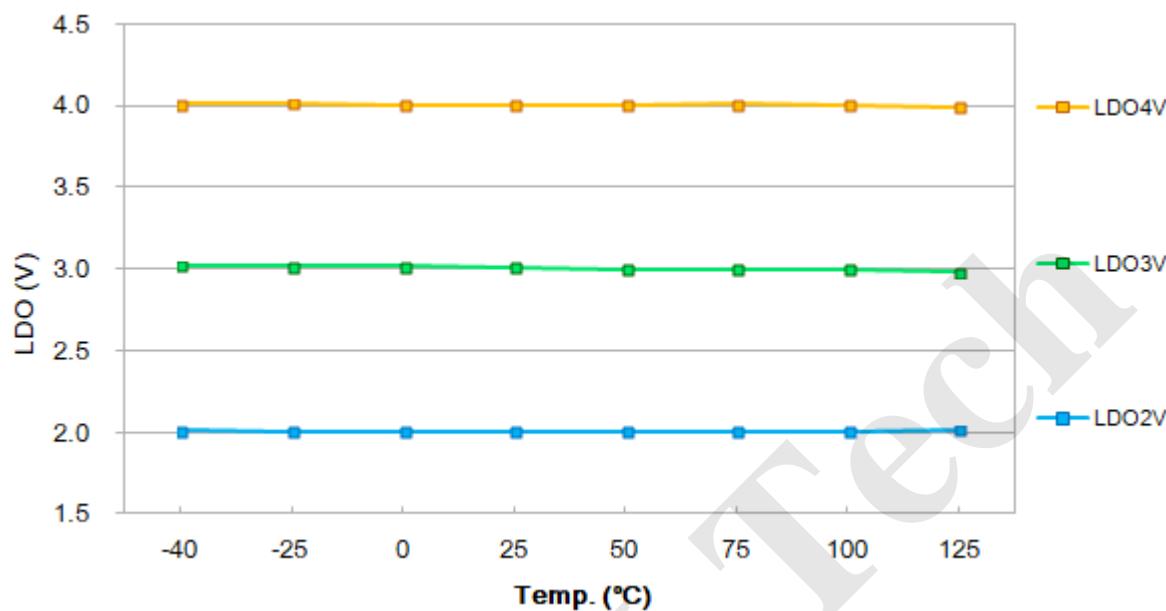
## 6.8 LVR电压与温度曲线图



### 6.9 LVD电压与温度曲线图

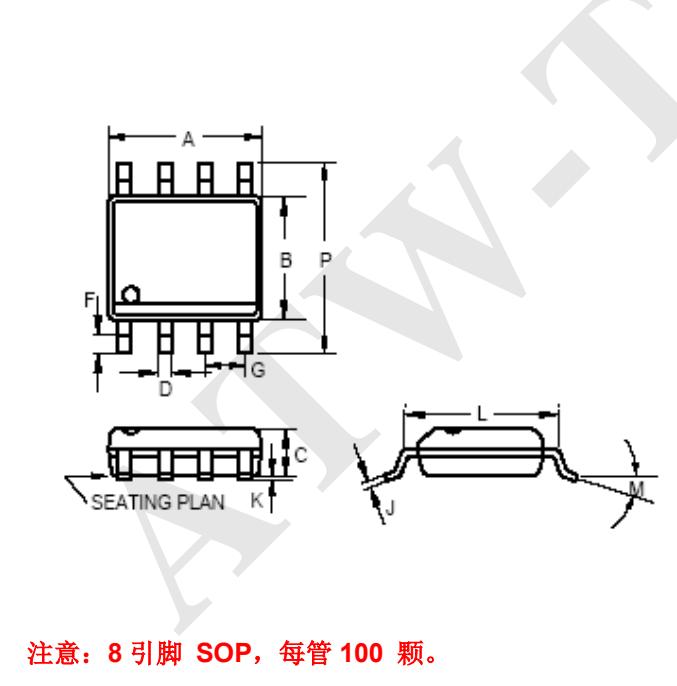


### 6.10 LDO与温度曲线图



## 7. 封装尺寸

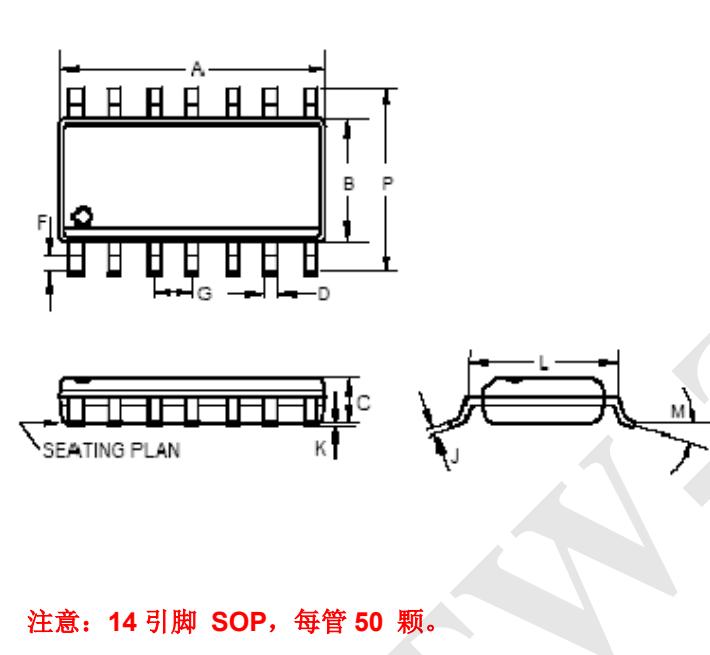
### 7.1 8 引脚SOP (150 mil)



注意: 8 引脚 SOP, 每管 100 颗。

	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.183	-	0.202	4.65	-	5.13
B	0.144	-	0.163	3.66	-	4.14
C	0.068	-	0.074	1.35	-	1.88
D	0.010	-	0.020	0.25	-	0.51
F	0.015	-	0.035	0.38	-	0.89
G	0.050 BSC			1.27 BSC		
J	0.007	-	0.010	0.19	-	0.25
K	0.005	-	0.010	0.13	-	0.25
L	0.189	-	0.205	4.80	-	5.21
M	-	-	8°	-	-	8°
P	0.228	-	0.244	5.79	-	6.20

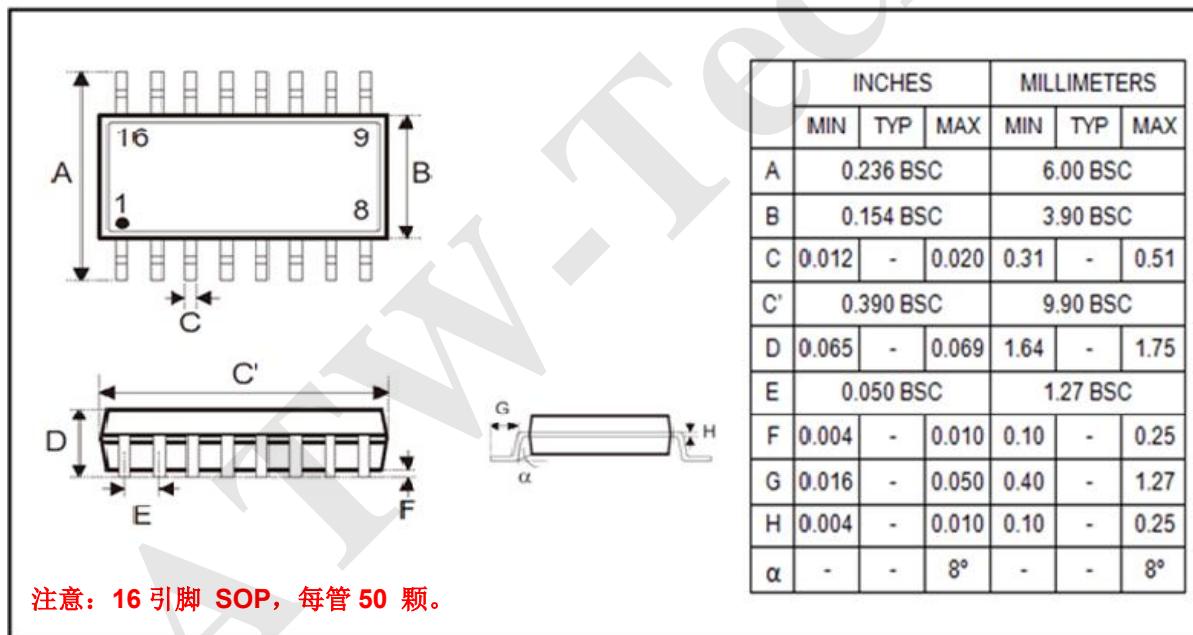
### 7.2 14 引脚SOP (150 mil)



注意: 14 引脚 SOP, 每管 50 颗。

	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.337	-	0.344	8.55	-	8.75
B	0.144	-	0.163	3.66	-	4.14
C	0.068	-	0.074	1.73	-	1.88
D	0.017	-	0.020	0.35	-	0.51
F	0.016	-	0.044	0.40	-	1.12
G	0.050 BSC			1.27 BSC		
J	-	0.004	-	-	0.10	-
K	0.005	-	0.010	0.13	-	0.25
L	0.189	-	0.205	4.80	-	5.21
M	-	-	8°	-	-	8°
R	0.228	-	0.244	5.80	-	6.20

## 7.3 16 引脚SOP (150 mil)



## 8. 订购信息

产品名称	封装类型	引脚数	封装尺寸	配送方式
AT8B62F1S8	SOP	8	150 mil	<u>卷带:</u> 每卷 2.5K。 <u>管装:</u> 每管 100 颗。
AT8B62F1S14	SOP	14	150 mil	<u>卷带:</u> 每卷 2.5K。 <u>管装:</u> 每管 50 颗。
AT8B62F1S16	SOP	16	150 mil	<u>卷带:</u> 每卷 2.5K。 <u>管装:</u> 每管 50 颗。