



深圳市矽源特科技有限公司
ShenZhen ChipSourceTek Technology Co.,Ltd.

ZH262F

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



改版记录

版本	日期	内容描述	修正页
1.0	2022/11/30	初始版本	-



目 录

1. 概述	8
1.1 功能	8
1.2 ZH262F与ZH262E, ZH262D的主要差异	10
1.2系统框图	11
1.3 引脚图	11
1.4 引脚说明	12
2. 内存结构	14
2.1 程序存储器	14
2.2 数据存储器	15
3. 功能概述	18
3.1 R-page特殊功能寄存器	18
3.1.1 <i>INDF</i> (间接寻址寄存器)	18
3.1.2 <i>TMR0</i> (定时器 0 寄存器)	18
3.1.3 <i>PCL</i> (程序计数器低字节)	18
3.1.4 <i>STATUS</i> (状态寄存器)	18
3.1.5 <i>FSR</i> (数据指针寄存器)	19
3.1.6 <i>PortA</i> (<i>PortA</i> 数据寄存器)	19
3.1.7 <i>PortB</i> (<i>PortB</i> 数据寄存器)	20
3.1.8 <i>PCON</i> (<i>Power</i> 寄存器)	20
3.1.9 <i>BWUCON</i> (<i>PortB</i> 唤醒控制寄存器)	20
3.1.10 <i>PCHBUF</i> (程序计数器高字节)	21
3.1.11 <i>ABPLCON</i> (<i>PortA/PortB</i> 下拉电阻控制寄存器)	21
3.1.12 <i>BPHCON</i> (<i>PortB</i> 上拉电阻控制寄存器)	21
3.1.13 <i>INTE</i> (中断使能寄存器)	22
3.1.14 <i>INTF</i> (中断标志寄存器)	22
3.1.15 <i>ADMD</i> (<i>ADC</i> 模式寄存器)	23
3.1.16 <i>ADR</i> (<i>ADC</i> 时钟, <i>ADC</i> 中断标志位与 <i>ADC</i> 转换结果低四位数据寄存器)	24
3.1.17 <i>ADD</i> (<i>ADC</i> 输出数据寄存器)	25
3.1.18 <i>ADVREFH</i> (<i>ADC</i> 参考电压寄存器)	25
3.1.19 <i>ADCR</i> (<i>ADC</i> 采样时间与 <i>ADC</i> 位数寄存器)	25
3.1.20 <i>AWUCON</i> (<i>PortA</i> 唤醒控制寄存器)	25



3.1.21	PACON (ADC引脚数模控制寄存器)	26
3.1.22	ADJMD (ADC调校模式)	26
3.1.23	INTEDG (外部中断控制寄存器)	26
3.1.24	TMRH (定时器 1/2 高字节寄存器)	27
3.1.25	ANAEN (比较器使能寄存器)	27
3.1.26	RFC (电阻频率转换控制寄存器)	28
3.1.27	TM3RH (定时器 3 高字节寄存器)	28
3.1.28	INTE2 (第 2 中断屏蔽寄存器)	29
3.2	T0MD (定时器 0 控制寄存器)	29
3.3	F-page特殊功能寄存器	30
3.3.1	IOSTA (PortA I/O控制寄存器)	30
3.3.2	IOSTB (PortB I/O控制寄存器)	30
3.3.3	APHCON (PortA上拉电阻控制寄存器)	31
3.3.4	PS0CV (预分频器 0 寄存器)	31
3.3.5	BODCON (PortB开漏控制寄存器)	31
3.3.6	CMPCR (比较器控制寄存器)	31
3.3.7	PCON1 (Power控制寄存器 1)	32
3.4	S-page特殊功能寄存器	33
3.4.1	TMR1 (定时器 1 寄存器)	33
3.4.2	T1CR1 (定时器 1 控制寄存器 1)	33
3.4.3	T1CR2 (定时器 1 控制寄存器 2)	34
3.4.4	PWM1DUTY (PWM1 占空比寄存器)	35
3.4.5	PS1CV (预分频器 1 寄存器)	35
3.4.6	BZ1CR (蜂鸣器 1 控制寄存器)	35
3.4.7	IRCR (IR控制寄存器)	36
3.4.8	TBHP (表格指针高字节寄存器)	37
3.4.9	TBHD (表格数据高字节寄存器)	37
3.4.10	TMR2 (定时器 2 寄存器)	37
3.4.11	T2CR1 (定时器 2 控制寄存器 1)	38
3.4.12	T2CR2 (定时器 2 控制寄存器 2)	39
3.4.13	PWM2DUTY (PWM2 占空比寄存器)	39
3.4.14	PS2CV (预分频器 2 寄存器)	40
3.4.15	BZ2CR (蜂鸣器 2 控制寄存器)	40
3.4.16	OSCCR (振荡器控制寄存器)	41
3.4.17	TMR3 (定时器 3 寄存器)	42



3.4.18	T3CR1 (定时器3 控制寄存器1)	42
3.4.19	T3CR2 (定时器3 控制寄存器2)	43
3.4.20	PWM3DUTY (PWM3 占空比寄存器)	43
3.4.21	PS3CV (预分频器3 寄存器)	44
3.4.22	BZ3CR (蜂鸣器3 控制寄存器)	44
3.4.23	P4CR1 (PWM4 控制寄存器)	45
3.4.24	PWM4DUTY (PWM4 占空比寄存器)	45
3.5	I/O Port	45
3.5.1	IO 引脚结构框图	47
3.6	定时器 0	57
3.7	定时器 1/PWM1/Buzzer1	58
3.8	定时器 2/PWM2/Buzzer2	60
3.9	定时器 3/PWM3/Buzzer3/PWM4	63
3.10	RFC (电阻/频率转换器模式)	66
3.11	IR 载波	66
3.12	低电压侦测 (LVD)	67
3.13	电压比较器	67
3.14	ADC模数转换器	70
3.14.1	ADC参考电压	70
3.14.2	ADC模拟输入通道	71
3.14.3	ADC时钟 (ADCLK), 采样时钟 (SHCLK) 与位数选择	72
3.14.4	ADC操作顺序	73
3.15	看门狗定时器 (WDT)	73
3.16	中断	73
3.16.1	Timer0 上溢中断	74
3.16.2	Timer1 下溢中断	74
3.16.3	Timer2 下溢中断	74
3.16.4	Timer3 下溢中断	74
3.16.5	看门狗超时中断	74
3.16.6	PA/PB输入状态改变中断	74
3.16.7	外部中断0 输入	75
3.16.8	外部中断1 输入	75
3.16.9	低电压侦测中断	75
3.16.10	比较器输出翻转中断	75



3.16.11 ADC模数转换完成中断.....	75
3.17 振荡器配置.....	75
3.18 工作模式.....	78
3.18.1 正常模式.....	79
3.18.2 慢速模式.....	79
3.18.3 待机模式.....	79
3.18.4 睡眠模式.....	80
3.18.5 唤醒稳定时间.....	80
3.18.6 工作模式概述.....	80
3.19 复位.....	81
4. 指令设置.....	83
5. 配置字节表.....	99
6. 电气特性.....	101
6.1 最大绝对值.....	101
6.2 直流电气特性.....	101
6.3 OSC电气特性.....	103
6.4 比较器/LVD电气特性.....	103
6.5 ADC电气特性.....	103
6.6 特性曲线图.....	104
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.....	104
6.6.2 高速RC振荡频率(I_HRC)与温度曲线图.....	104
6.6.3 低速RC振荡频率(I_LRC)与温度曲线图.....	105
6.6.4 内部参考电压LDO与温度曲线图.....	105
6.6.5 内部上拉电阻与电源电压(VDD)曲线图.....	106
6.6.6 VIH/VIL与电源电压(VDD)曲线图.....	106
6.7 建议工作电压.....	107
6.8 LVR电压与温度曲线图.....	108
6.9 LVD电压与温度曲线图.....	108
6.10 LDO与温度曲线图.....	109
7. 晶圆脚位图.....	109
8. 封装尺寸.....	110
8.1 8引脚SOP（150 毫寸）.....	110



8.2 14 引脚SOP (150 毫寸)	110
8.3 16 引脚SOP (150 毫寸)	111
9. 订购信息	111

矽源特科技
ChipSourceTek



1. ZH262F概述

ZH262F是以EPROM作为存储器的 8 位单片机，专为家电或量测等等的I/O应用设计。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。ZH262F核心建立在RISC精简指令集架构可以很容易地做编程和控制，共有 55 条指令。除了少数指令需要 2 个指令时钟，大多数指令都是 1 个指令时钟能完成，可以让用户轻松地以过程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。

ZH262F内建高精度十一加一通道 12 位ADC模数转换器，与高精度电压比较器，足以应付各种模拟接口的侦测与 量测。

在 I/O 的资源方面，ZH262F 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 脚位都能通过控制相关的寄存器达成如上拉或下拉电阻或开漏（Open-Drain）输出。此外针对红外线遥控的产品方面，ZH262F 内置了可选择频率的红外载波发射口。

ZH262F 有四组定时器，可用系统时钟当作一般的计时应用或者从外部讯号触发来计数。另外 ZH262F 提供 4 组 10 位的 PWM 输出，3 组蜂鸣器输出，可用来驱动马达、LED、或蜂鸣器等等。

ZH262F 采用双时钟机制，高速振荡时钟或者低速振荡时钟都由内部 RC 振荡或外部晶振输入。在双时钟机制下，ZH262F 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。并且单片机在使用内部 RC 高速振荡时，低速振荡可以同时使用外部精准的晶振计时。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，有多个中断源可以触发来唤醒ZH262F 进入正常操作模式（Normal mode）或慢速模式（Slow mode）来处理突发事件。

1.1 功能

- 宽广的工作电压：
 - 2.0V ~ 5.5V @系统时钟 $\leq 8\text{MHz}$ 。
 - 2.2V ~ 5.5V @系统时钟 $> 8\text{MHz}$ 。
- 宽广的工作温度：-40°C ~ 85°C。
- 2Kx14 位的程序存储器空间。
- 128 字节的通用数据存储器空间。
- 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部下拉电阻。
- PA[7:0]及PB[5:0]可选择输入时使用内部上拉电阻。
- PB[5:0]可选择开漏输出（Open-Drain）。
- PA[5]可选择当作输入或开漏输出（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 级深度硬件堆栈（Stack）。



- 存取数据有直接或间接寻址模式。
- 一组 8 位上数定时器 (Timer0) 包含可编程的预分频器。
- 三组 10 位下数定时器 (Timer1、2、3) 可选自动重载或连续下数计时。
- 四个 10 位的脉冲宽度调变输出 (PWM1、2、3、4)。
- 三个蜂鸣器输出 (BZ1、2、3)。
- 38/57KHz 红外线载波 (IR) 频率可供选择, 同时载波之极性也可以通过寄存器选择。
- 内置准确的低电压侦测电路 (LVD)。
- 内置十一加一通道 12 位 ADC 模数转换器 (Analog to Digital Converter)。
- 内置精准的电压比较器 (Voltage Comparator)。
- 内置上电复位电路 (POR)。
- 内置低压复位功能 (LVR)。
- 内置看门狗计时 (WDT), 可由配置字节 (Configuration Word) 控制开/关。
- 内置电阻/频率转换器 (RFC) 功能。
- 双时钟机制, 系统时钟可以随时切换高速振荡或者低速振荡。
 - 高速振荡时钟: E_HXT (超过 6MHz 外部高速晶振)
E_XT (455K~6MHz 外部晶振)
I_HRC (1~20MHz 内部高速 RC 振荡)
 - 低速振荡时钟: E_LXT (32KHz 外部低速晶振)
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 mode)、慢速模式 (Slow mode)、待机模式 (Standby mode) 与睡眠模式 (Halt mode)。
- 十一种硬件中断:
 - Timer0 上溢中断。
 - Timer1 下溢中断。
 - Timer2 下溢中断。
 - Timer3 下溢中断。
 - WDT 中断。
 - PA/PB 输入状态改变中断。
 - 两组外部中断。
 - 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。
- ZH262F 在待机模式 (Standby mode) 下的十一种唤醒中断:
 - Timer0 上溢中断。



- Timer1 下溢中断。
 - Timer2 下溢中断
 - Timer3 下溢中断。
 - WDT中断。
 - PA/PB输入状态改变中断。
 - 两组外部中断。
 - 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。
- ZH262F在睡眠模式（Halt mode）下的四种唤醒中断： ➢
- WDT中断。
 - PA/PB输入状态改变中断。
 - 两组外部中断。

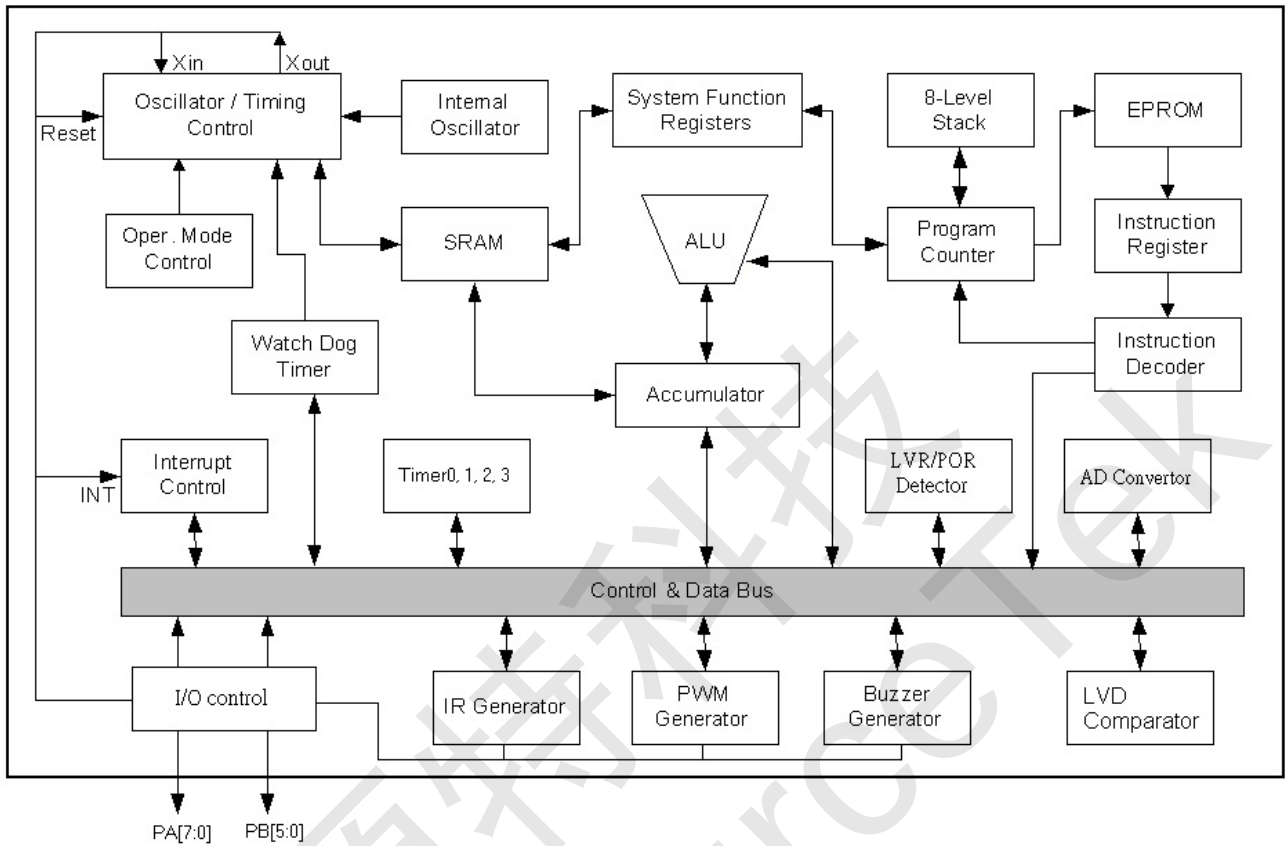
1.2 ZH262F 与 ZH262E, ZH262D的主要差异

项目	功能	ZH262F	ZH262E	ZH262D
1	ADC 偏移校准 ^{註*}	Yes	Yes	---
2	ADC 耗电	500uA @5V	500uA @5V	3mA @5V
3	内部 LDO	±0.5%	±0.5%	±1.0%
4	输入施密特触发	开启/关闭	开启/关闭	---
5	PWM 通道	PWM1, 2, 3, 4	PWM1, 2, 3, 4	PWM1, 2, 3

*注：需要增加 ADC 零点校准初始化程式 (可参考范例程式)



1.2 系统框图



1.3 引脚图

ZH262F提供三种封装类型：SOP16、SOP14 及SOP8。

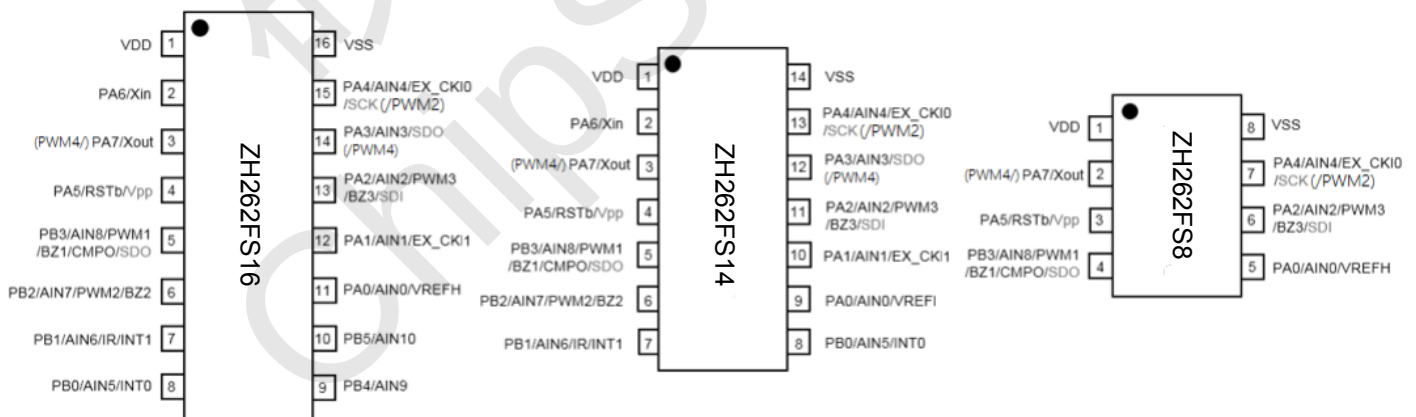


图 1 封装引脚图



1.4 引脚说明

引脚名	I/O	描述
PA0 AIN0 VREFH	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AIN0。 PA0 可作ADC外部参考电压输入引脚VREFH。
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AIN1。 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AIN2。 PA2 可输出PWM3 或BZ3。 PA2 也是编程数据输入SDI。
PA3 AIN3 PWM4 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AIN3。 PA3 可输出PWM4。 PA3 也是编程数据输出SDO。
PA4 AIN4 PWM2 EX_CK10 SCK	I/O	PA4 是一个双向I/O引脚。 PA4 可作ADC的模拟输入引脚AIN4。 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。
PA5 RSTb Vpp	I/O	PA5 是一个双向I/O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。
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	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也可以当成指令时钟输出。
PB0 AIN5 INT0	I/O	PB0 是一个双向I/O引脚。 PB0 可作ADC的模拟输入引脚AIN5。 PB0 可当作外部中断 0 的输入引脚INT0。
PB1 AIN6 IR INT1	I/O	PB0 是一个双向I/O引脚。 PB0 可作ADC的模拟输入引脚AIN5。 如果开启IR模式，此脚可作IR载波输出。 PB0 可当作外部中断 0 的输入引脚INT0。
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AIN7。 PB2 可输出PWM2 或BZ2。



引脚名	I/O	描述
PB3 AIN8 PWM1/BZ1 CMPO SDO	I/O	PB3 是一个双向I/O引脚。 PB3 可作ADC的模拟输入引脚AIN8。 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。
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AIN9。
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AIN10。
VDD	-	电源正端。
VSS	-	电源负端。



2. ZH262F内存结构

ZH262F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

ZH262F程序存储器空间是 2Kx14 位。因此，11 位宽的计数器（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。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。

ZH262F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。

ZH262F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。

ZH262F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。

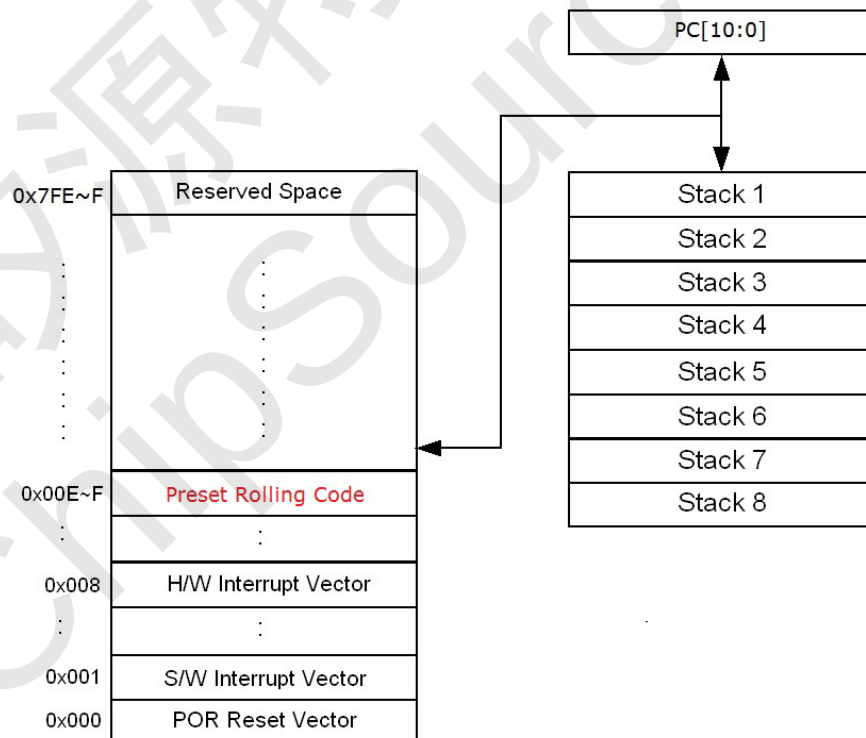


图2 程序存储器对应地址



2.2 数据存储器的

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，可透过数据指针寄存器（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来进行存取。

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。

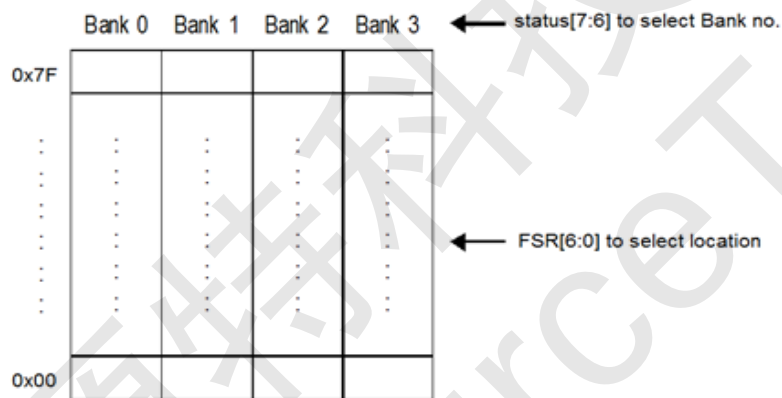


图 3 间接寻址方式存取数据存储器

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。

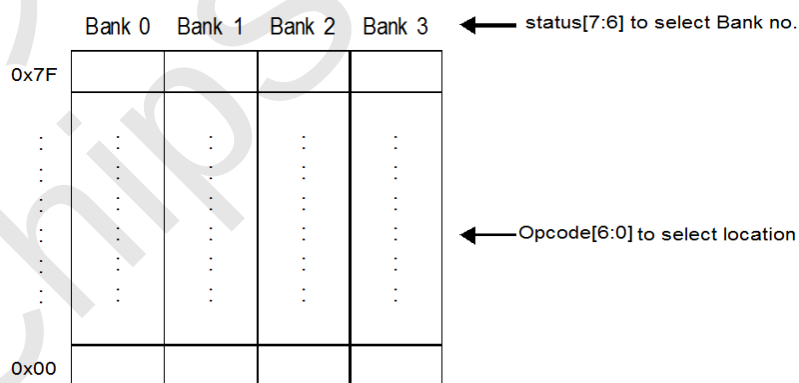


图 4 直接寻址方式存取数据存储器

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 的相同地址会映像到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映像如表 1 所示。



ZH262F寄存器名称和R-page特殊功能寄存器的映像地址说明如下表。

Status [7:6] 地址	00 (Bank 0)	01 (Bank 1)	10 (Bank 2)	11 (Bank 3)
0x0	INDF	映射至Bank 0		
0x1	TMR0			
0x2	PCL			
0x3	STATUS			
0x4	FSR			
0x5	PORTA			
0x6	PORTB			
0x7	-			
0x8	PCON			
0x9	BWUCON			
0xA	PCHBUF			
0xB	ABPLCON			
0xC	BPHCON			
0xD	-			
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像表



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。

地址	特殊功能寄存器种类	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

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. ZH262F功能概述

本章节将详细描述ZH262F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与配置字节（Configuration Word），定时器 0 时钟源可以从指令时钟FINST、外部时钟EX_CK10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(*2)	读/写(*1)	读/写	读/写	读/写
初始值			0	0	0	1	1	X	X	X



状态寄存器包含算术/逻辑指令的结果和是否发生看门狗超时复位。

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。

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。

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。

/PD: 睡眠模式标志位。

/PD=1 时，上电或执行CLRWDWT指令后。

/PD=0 时，执行SLEEP指令后。

/TO: 看门狗超时标志位。

/TO=1 时，上电或执行CLRWDWT或SLEEP指令后。

/TO=0 时，发生WDT上溢。

GP5: 通用寄存器数据位。

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CLRWDWT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

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

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，依据配置字节选项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，依据配置字节选项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。

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

GP6, GP4, GP2, GP1, GP0: 通用寄存器数据位。

LVREN: 开启/关闭 LVR。

LVREN=1 时，开启LVR。

LVREN=0 时，关闭LVR。

/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。

LVDEN: 开启/关闭 LVD。

LVDEN=1 时，开启LVD。

LVDEN=0 时，关闭LVD。

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
-----	---	---	---	---	---	---	---	---

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	-	-	-	-	-	-
读/写属性			-	W	-	-	-	-	-	-
初始值			X	0	-	X	-	-	-	-

PCHBUF[2:0]: 程序计数器PC的第十个位到第八个位。

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

/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。

/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
名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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

/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 INTE（中断使能寄存器）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTE	R	0xE	INT1IE	WDTIE	T2IE	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

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。

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。

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。

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。

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。

T2IE: 定时器 2 下溢（underflow）中断使能位。

T2IE=1 时，开启定时器 2 下溢中断。

T2IE=0 时，关闭定时器 2 下溢中断。

WDTIE: WDT上溢中断使能位。

WDTIE=1 时，开启WDT上溢中断。

WDTIE=0 时，关闭WDT上溢中断。

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	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



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必须由程序清零。

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必须由程序清零。

INT0IF: 外部中断 0 标志位。

INT0IF=1 时, 发生外部 0 中断。

INT0IF必须由程序清零。

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必须由程序清零。

LVDIF: 低电压侦测中断标志位。

LVDIF=1, 发生低电压侦测中断。

LVDIF必须由程序清零。

T2IF: 定时器 2 下溢中断标志位。

T2IF=1 时, 发生定时器 2 下溢中断。

T2IF必须由程序清零。

WDTIF: WDT超时上溢标志位。

WDTIF=1 时, 发生WDT上溢中断。

WDTIF必须由程序清零。

INT1IF: 外部中断 1 标志位。

INT1IF=1 时, 发生外部 1 中断。

INT1IF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	W	R	读/写	读/写	读/写	读/写	读/写
初始值			0	0	1	0	0	0	0	0

ADEN: 开启/关闭ADC功能。

ADEN=1 时, 开启ADC功能。

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。

EOC: ADC状态位, 只读。



EOC=1: ADC转换完成。可由ADR与ADD读取转换结果数据。

EOC=0: ADC转换中。

GCHS: 开启/关闭ADC总通道。

GCHS=0: 关闭所有ADC模拟输入通道。

GCHS=1: 开启所有ADC模拟输入通道。

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。

3.1.16 ADR (ADC 时钟, ADC 中断标志位与 ADC 转换结果低四位数据寄存器)

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	R	R	R	R
初始值			0	0	0	0	X	X	X	X

ADIF: ADC中断标志位。

ADIF=1 时, 发生ADC转换完成中断。

ADIF必须由程序清零。

ADIE: ADC中断使能位。

ADIE=1 时, 开启ADC中断。

ADIE=0 时, 关闭ADC中断。

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= $F_{INST}/4$, 11: ADC时钟= $F_{INST}/2$ 。

AD3~0: ADC转换结果低四位数据。



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			R	R	R	R	R	R	R	R
初始值			0	0	0	0	0	0	0	0

AD11~4: ADC转换结果高八位数据。

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

EVHENB: ADC参考电压 (VREFH) 选择控制位。

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

EVHENB=1: ADC参考电压由引脚PA0 提供。

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	Addr	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

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。

ADCR1~0: ADC位数选择位。

00: 8 位ADC, 01: 10 位ADC, 1x: 12 位ADC。

PBCONx: PB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。

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	读/写	读/写	读/写	读/写	读/写	读/写	读/写
初始值	1	1	1	1	1	1	1	1

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 引脚数模控制寄存器)

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	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

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。

PBCONx: PA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。

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

ADJ[x]: 调校位选择, $0 \leq x \leq 4$ 。

00000: 偏移 0mV

11111: 偏移 12.5mV

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

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。

EIS0: 外部中断 0 引脚选择位。

EIS0=1 时, PB0 选择为外部中断 0 引脚。

EIS0=0 时, PB0 选择为GPIO。

INT1G1~0: INT1 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。

INT0G1~0: INT0 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。

3.1.24 TMRH (定时器 1/2 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

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。

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。

PWM2DUTY9~8: PWM2 占空比高 2 位。

PWM1DUTY9~8: PWM1 占空比高 2 位。

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

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			

RFCEN: 关闭/开启RFC。

RFCEN=1, 开启RFC。

RFCEN=0, 关闭RFC。

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	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM3RH	R	0x1C	-	-	TMR39	TMR38	PWM4D9	PWM4D8	PWM3D9	PWM3D8
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

TMR39~8: 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。

PWM4DUTY9~8: PWM4 占空比高 2 位。

PWM3DUTY9~8: PWM3 占空比高 2 位。



3.1.28 INTE2（第2中断屏蔽寄存器）

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

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必须由程序清零。

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。

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			读/写							
初始值(note*)			0	0	1	1	1	111		

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]	预分频比选项		
	PS0WDT=0 (定时器0)	PS0WDT=1 (WDT复位)	PS0WDT=1 (WDT中断)
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

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：在使能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。



T0CE: 定时器 0 外部时钟源触发沿选择。

T0CE=1 时，上升沿时定时器 0 加一。

T0CE=0 时，下降沿时定时器 0 加一。

注意: T0CE应用在外部 EX_CKIO 脚作为定时器 0 时钟源。

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_{INST}。

GP6: 通用读写器寄存器位。

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_{INST}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/E_LXT为定时器 0 时钟源。

低频振荡输出代替引脚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	读/写	读/写	读/写	读/写	读/写	读/写	读/写
名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
初始值			1	1	1	1	1	1	1	1

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

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。

/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。

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			R							
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

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	0	0	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]确定 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 _{DD}	0001	PA1
0010	2 / 16 V _{DD}	0010	PA2
0011	3 / 16 V _{DD}	0011	PA3
0100	4 / 16 V _{DD}	0100	-
0101	5 / 16 V _{DD}	0101	-
0110	6 / 16 V _{DD}	0110	-
0111	7 / 16 V _{DD}	0111	-
1000	8 / 16 V _{DD}	1000	-
1001	9 / 16 V _{DD}	1001	-
1010	10 / 16 V _{DD}	1010	-
1011	11 / 16 V _{DD}	1011	-
1100	12 / 16 V _{DD}	1100	-
1101	13 / 16 V _{DD}	1101	-
1110	14 / 16 V _{DD}	1110	-
1111	15 / 16 V _{DD}	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	LVDOUT	GP5	LVDS2	LVDS1	LVDS0	GP1	T0EN
读/写属性			读/写(1*)	R	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	X	0	1	1	1	0	1

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。



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

LVDOOUT: 低电压检测输出，只读。

GIE: 开启/关闭总中断屏蔽位。

GIE=1 时，开启总中断。

GIE=0 时，关闭总中断。

GP5, GP1: 通用寄存器数据位。

(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	-	-	-	T1OS	T1RL	T1EN
读/写属性			读/写	读/写	-	-	-	读/写	读/写	读/写
初始值			0	0	X	X	X	0	0	0

此寄存器用于配置定时器 1 功能。

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。



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。

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 0x3FF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，定时器 1 从TMR1[9:0]重新载入数值并继续下数。
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

PWM1OAL: 定义PWM1 输出有效状态。

PWM1OAL=1 时，PWM1 为低电平有效位输出。

PWM1OAL=0 时，PWM1 为高电平有效位输出。

PWM1OEN: 开启/关闭PWM1 输出。

PWM1OEN=1，PB3 输出PWM1。

PWM1OEN=0，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。

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** 前须先设定 **PS1SEL[2:0]**，否则可能会误发生中断。

/PS1EN: 关闭/开启预除器 1。

/PS1EN=1 时，关闭预分频器 1。

/PS1EN=0 时，开启预分频器 1。

T1CE: 定时器 1 外部时钟触发沿选项。

T1CE=1 时，EX_CKIO 脚下降沿时定时器 1 减一。

T1CE=0 时，EX_CKIO 脚上升沿时定时器 1 减一。

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_{INST}。

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			W	-	-	-	W			
初始值			0	X	X	X	1	1	1	1

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



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	定时器 1 bit 0
1001		定时器 1 bit 1
1010		定时器 1 bit 2
1011		定时器 1 bit 3
1100		定时器 1 bit 4
1101		定时器 1 bit 5
1110		定时器 1 bit 6
1111		定时器 1 bit 7

表 10 蜂鸣器BZ1 输出频率选项

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

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。

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。

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。

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	E_XT 模式& IROSC358M=1
Xtal 455KHz	8	12	E_XT 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。

3.4.10 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	-	-	-	T2OS	T2RL	T2EN
读/写属性			读/写	读/写	-	-	-	读/写	读/写	读/写
初始值			0	0	X	X	X	0	0	0

此寄存器用于配置定时器 2 功能。

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。

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[9:0]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。

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。

T2OS	T2RL	定时器 2 计数选项
0	0	定时器 2 从 0x3FF 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，定时器 2 从TMR2[9:0]重新载入数值并继续下数。
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

PWM2OAL: 定义PWM2 输出有效状态。

PWM2OAL=1 时，PWM2 为低电平有效位输出。

PWM2OAL=0 时，PWM2 为高电平有效位输出。

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。

PS2SEL[2:0]: 预分频器 2 预分频比选项。

PS2SEL[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 13 预分频器 2 预分频比选项

注意: 在 **PS2EN=1** 前须先设定 **PS2SEL[2:0]**, 否则可能会误发生中断。

/PS2EN: 关闭/开启预除器 2。

/PS2EN=1 时, 关闭预分频器 2。

/PS2EN=0 时, 开启预分频器 2。

T2CE: 定时器 2 外部时钟触发沿选项。

T2CE=1 时, EX_CK11 脚下降沿时定时器 2 减一。

T2CE=0 时, EX_CK11 脚上升沿时定时器 2 减一。

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			写							
初始值			XXXXXXXX							

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 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 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			W	-	-	-	写			
初始值			0	X	X	X	1	1	1	1

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	定时器 2 输出	定时器 2 bit 0
1001		定时器 2 bit 1
1010		定时器 2 bit 2
1011		定时器 2 bit 3
1100		定时器 2 bit 4
1101		定时器 2 bit 5
1110		定时器 2 bit 6
1111		定时器 2 bit 7

表 14 蜂鸣器BZ2 输出频率选项

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			R	读/写	读/写	读/写	读/写		读/写	读/写
初始值			X	0	0	0	00		0	1

SELHOSC: 系统振荡器选择 (F_{osc})。

SELHOSC=1 时, F_{osc} 是高频率振荡器 (F_{HOSC})。

SELHOSC=0 时, F_{osc} 是低频率振荡器 (F_{LOSC})。

STPHOSC: 关闭/开启高频率振荡器 (F_{HOSC})。

STPHOSC=1 时, F_{HOSC} 会停止振荡并被关闭。

STPHOSC=0 时, F_{HOSC} 保持振荡。

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 选择OPMD[1:0]的操作模式

CMPIE: 比较器中断使能位。

CMPIE=1 时, 开启比较器中断。

CMPIE=0 时, 关闭比较器中断。

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。

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。

CMPOUT: 比较器输出状态位。

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TMR3RH[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	-	-	-	T3OS	T3RL	T3EN
读/写属性			读/写	读/写	-	-	-	读/写	读/写	读/写
初始值			0	0	X	X	X	0	0	0

此寄存器用于配置定时器 3 功能。

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。

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。

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 0x3FF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，定时器 3 从TMR3[9:0]重新载入数值并继续下数。
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

PWM3OAL: 定义PWM3 输出有效状态。

PWM3OAL=1 时，PWM3 为低电平有效位输出。

PWM3OAL=0 时，PWM3 为高电平有效位输出。

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。

PS3SEL[2:0]: 预分频器 3 预分频比选项。

PS3SEL[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 17 预分频器 3 预分频比选项

注意: 在**PS3EN=1** 前须先设定**PS3SEL[2:0]**, 否则可能会误发生中断。

/PS3EN: 关闭/开启预除器 3。

/PS3EN=1 时, 关闭预分频器 3。

/PS3EN=0 时, 开启预分频器 3。

T3CE: 定时器 3 外部时钟触发沿选项。

T3CE=1 时, EX_CK11 脚下降沿时定时器 3 减一。

T3CE=0 时, EX_CK11 脚上升沿时定时器 3 减一。

T3CS: 定时器 3 时钟源选项。

T3CS=1 时, 选择EX_CK11 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 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			W	-	-	-	写			
初始值			0	X	X	X	1	1	1	1

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
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 3 输出	定时器 3 bit 0
1001		定时器 3 bit 1
1010		定时器 3 bit 2
1011		定时器 3 bit 3
1100		定时器 3 bit 4
1101		定时器 3 bit 5
1110		定时器 3 bit 6
1111		定时器 3 bit 7

表 18 蜂鸣器BZ3 输出频率选项

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）

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

PWM4OAL: 定义PWM4 输出有效状态。

PWM4OAL=1 时, PWM4 为低电平有效位输出。

PWM4OAL=0 时, PWM4 为高电平有效位输出。

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。

3.5 I/O Port

ZH262F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, 寄存器IOSTA[7:0]定义PA[7:0]为输入或输出, 寄存器IOSTB[5:0]定义PB[5:0]为输入或输出。

当一个I/O脚位被配置为输入, 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BPHCON[3:0]用于开启或关闭PB[3:0]的内部上拉电阻。寄存器ABPLCON[7:4]则是用于开启或关闭PB[3:0]的内部下拉电阻。ABPLCON[3:0]则是用于开启或关闭PA[3:0]的内部下拉电阻。APHCON[5]用于开启或关闭PA[5]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, 可由寄存器开启或关闭开漏。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



功能		PA[3:0]	PA[7:6]&PA[4]	PA[5]	PB[3:0]	PB[5:4]
输出	开漏	X	X	总是	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]会使能或禁能任一PA和PB脚位的唤醒功能。只要AWUCON和BWUCON对应到的任一PA和PB脚位被置为 1 时，且在此输入脚位有状态改变时，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。

ZH262F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。

ZH262F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。

由配置字节决定PA5 可否当作外部复位输入RSTb。当PA5 为低电平时将导致ZH262F发生复位。

当配置字节设置外置晶振（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）。

当配置字节设置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_{INST}。

此外，当T0MD T0CS=1 和LCK_TM0=0 时，PA4 可以是定时器 0 外部时钟源EX_CKIO。当T1CS=1 时，PA4 为定时器 1 外部时钟源EX_CKIO。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节设置为一般灌电流（19mA@VDD=3V），大灌电流（28mA@VDD=3V）。如下表 21 所示。

配置字节	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。

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。

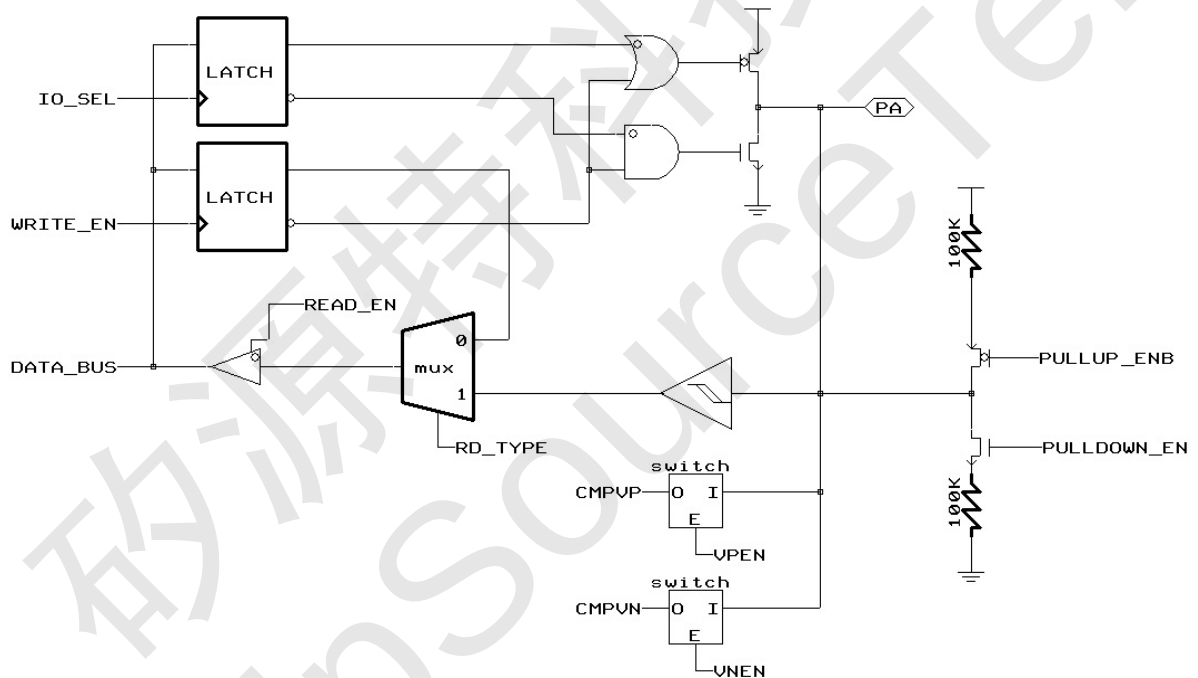


图 5 PA[3:2], PA0 引脚结构框图



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。

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。

EX_CK11: Timer 2/3 外部时钟输入。

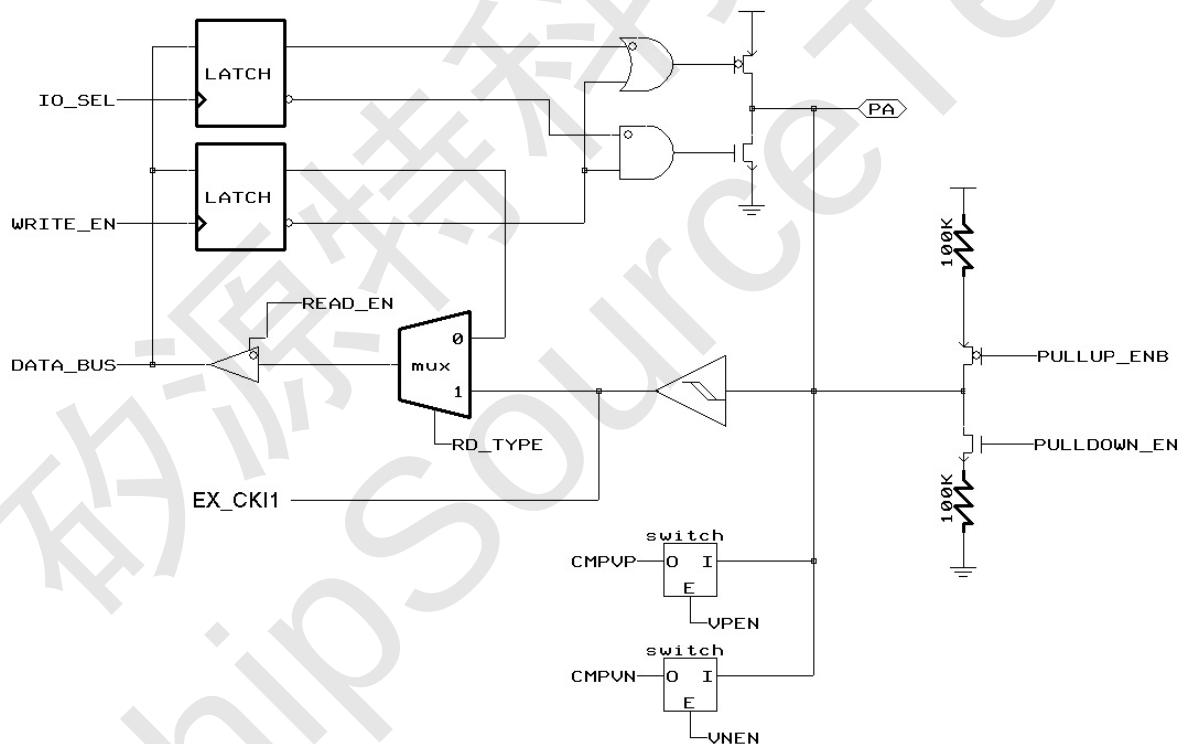


图 6 PA1 引脚结构框图

EX_CK10: Timer0,1 外部时钟输入。





RSTPAD_EN: 开启外部复位引脚。

RSTB_IN: 复位引脚输入。

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。

PULLDOWN_EN: 开启内部下拉电阻。

RD_TYPE: 选择读取脚位或数据锁存器。

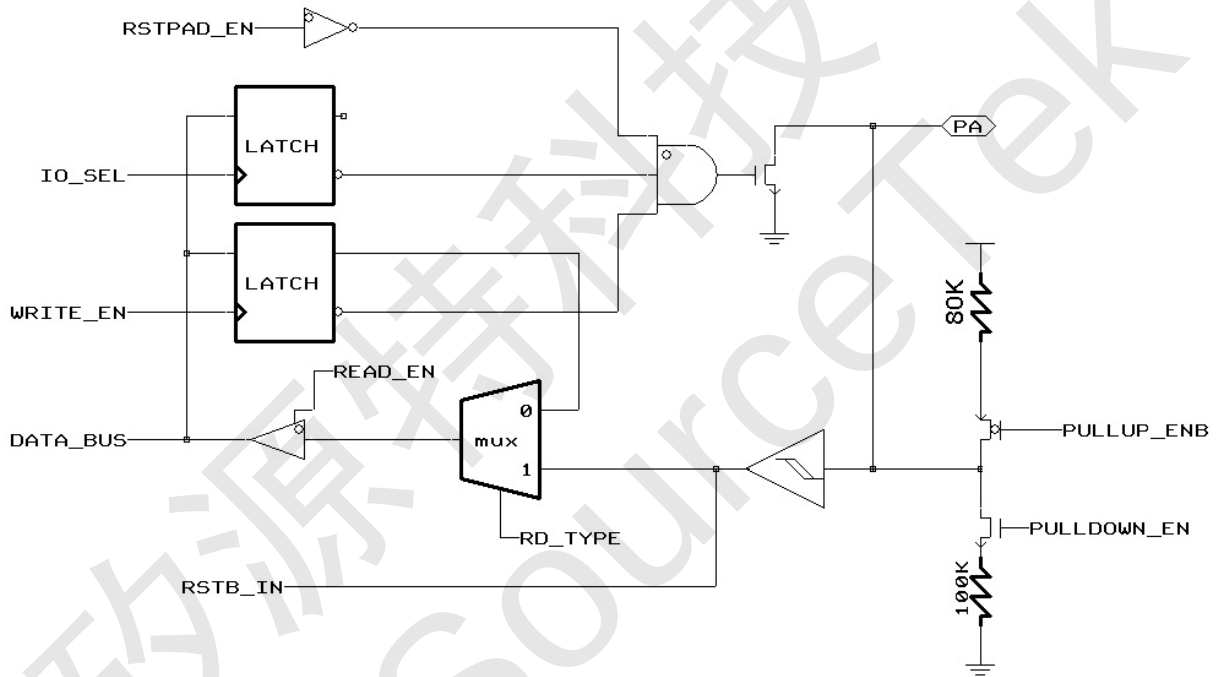


图 8 PA5 引脚结构框图



XTL_EN: 开启外置晶振。

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。

PULLUP_ENB: 开启内部上拉电阻。

RD_TYPE: 选择读取脚位或数据锁存器。

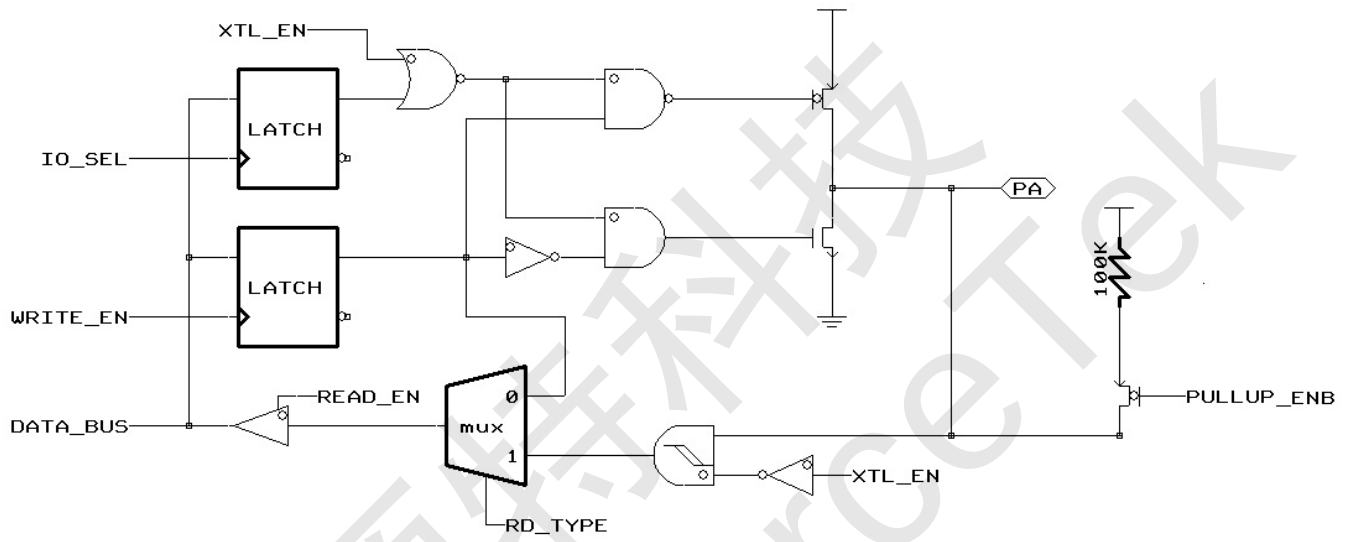


图 9 PA6, PA7 引脚结构框图



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。

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。

EIS0: 使能外部中断 0 功能。

INTEDG[1:0]: 选择外部中断 0 触发沿。

EX_INT0: 外部中断 0 信号。

WUB: 使能PB唤醒功能。

SET_PBIF: PB唤醒标志。

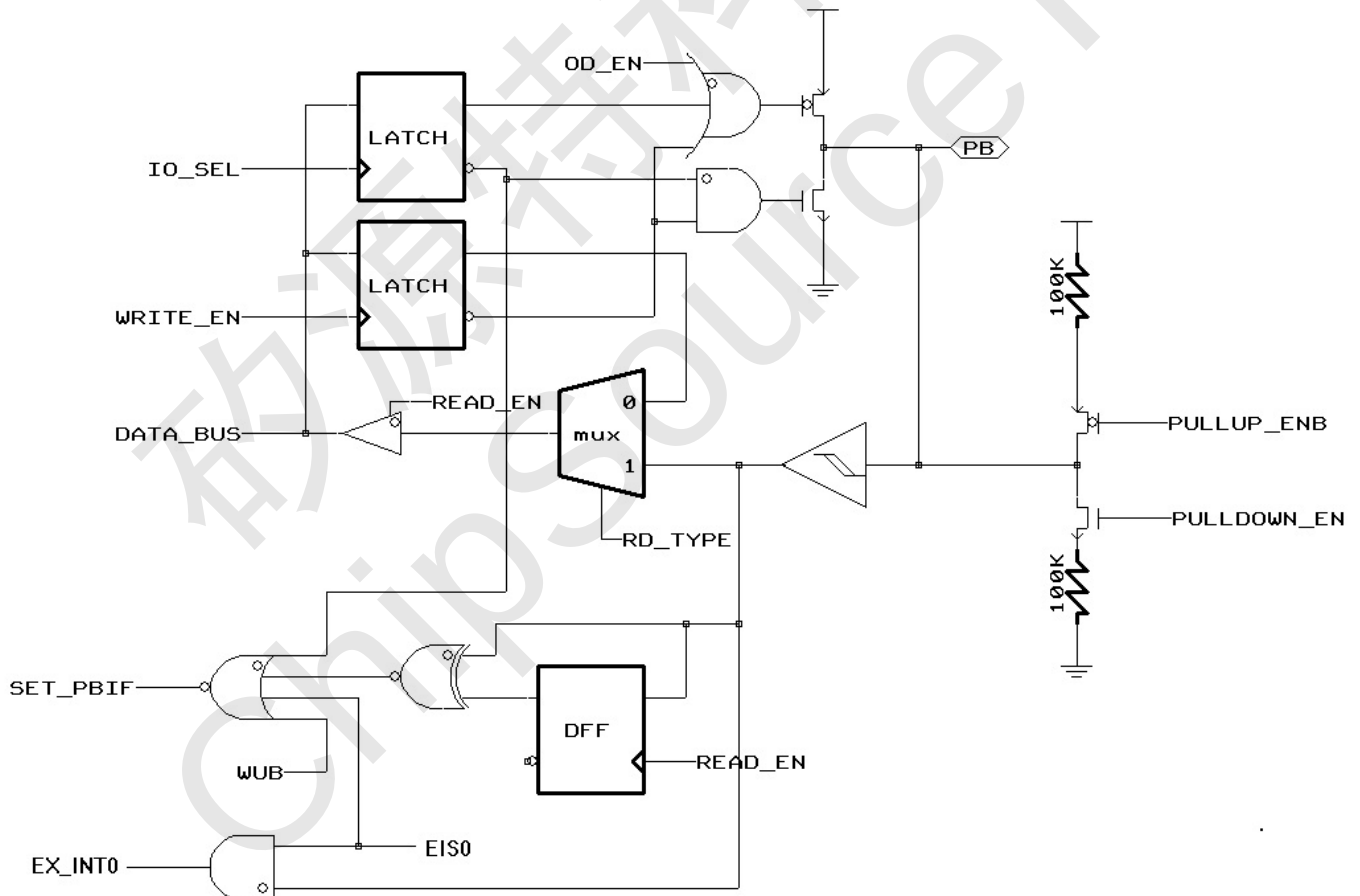
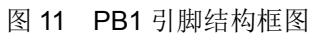


图 10 PB0 引脚结构框图

SET PBIF: PB唤醒标志。





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: 使能PB唤醒功能。

SET_PBIF: PB唤醒标志。

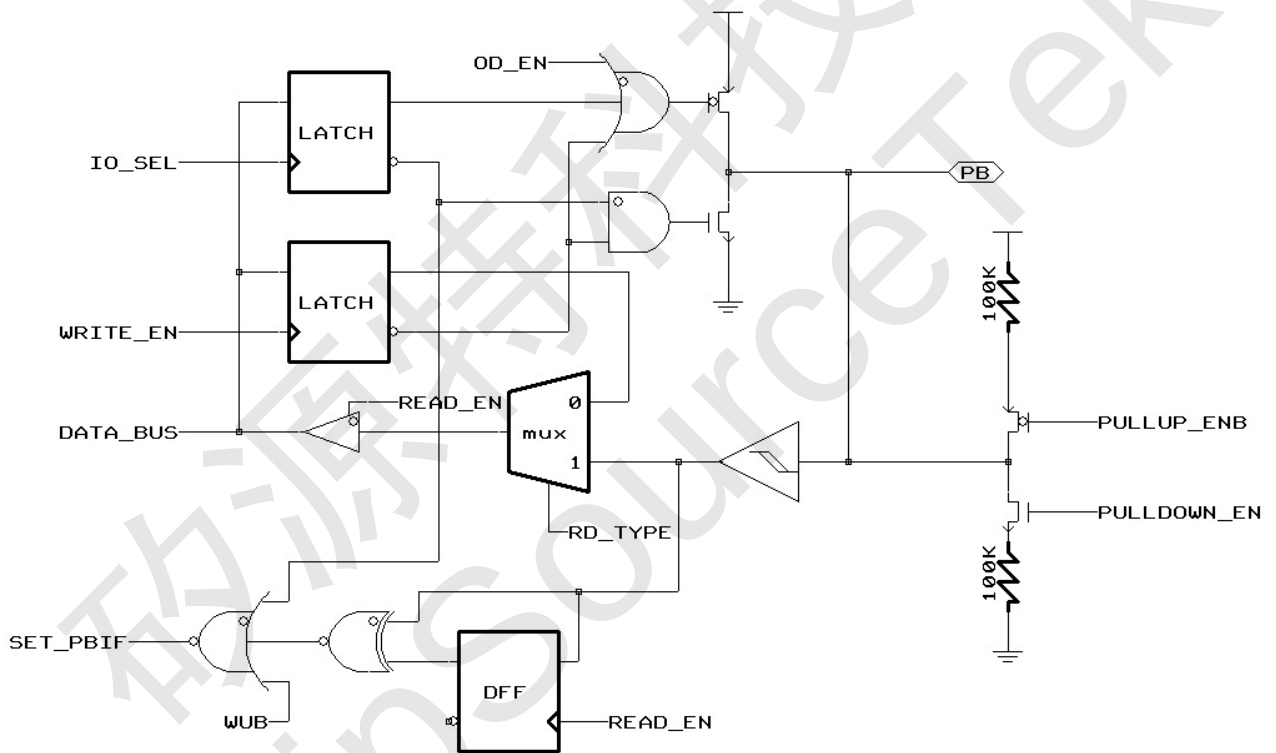


图 12 PB2 引脚结构框图



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: 使能PB唤醒功能。

SET_PBIF: PB唤醒标志。

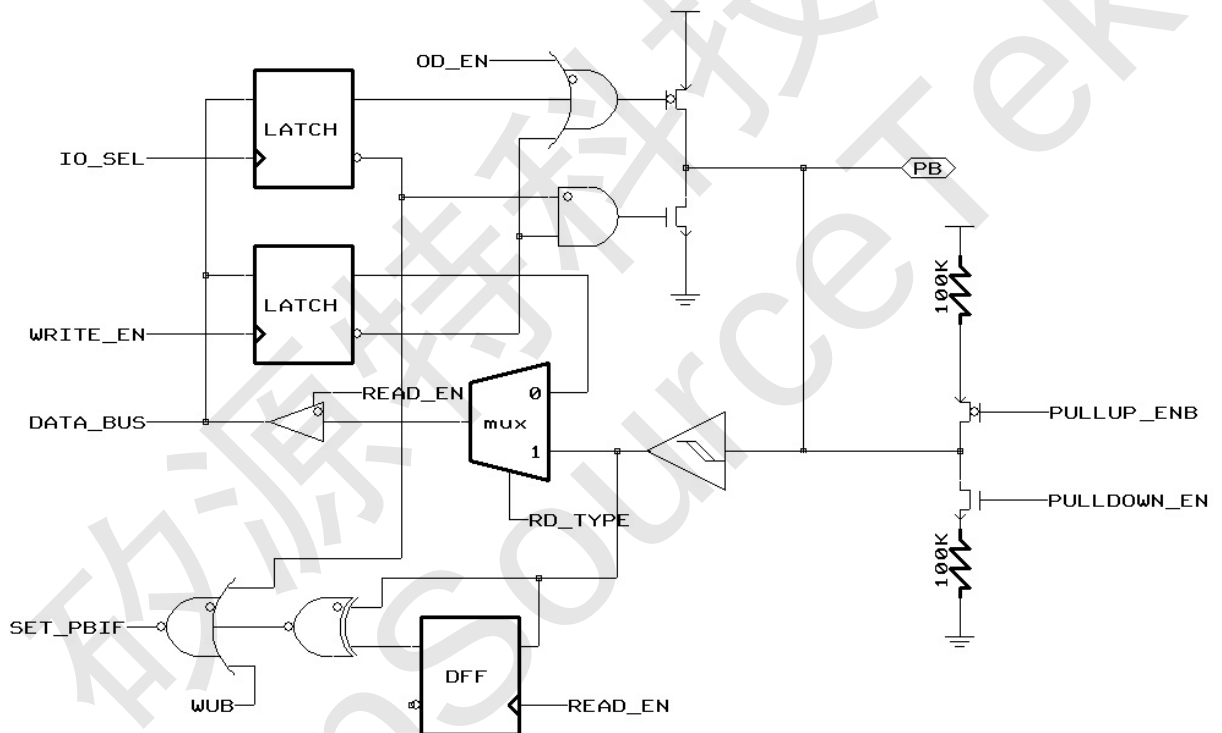


图 13 PB3 引脚结构框图



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: 使能PB唤醒功能。

SET_PBIF: PB唤醒标志。

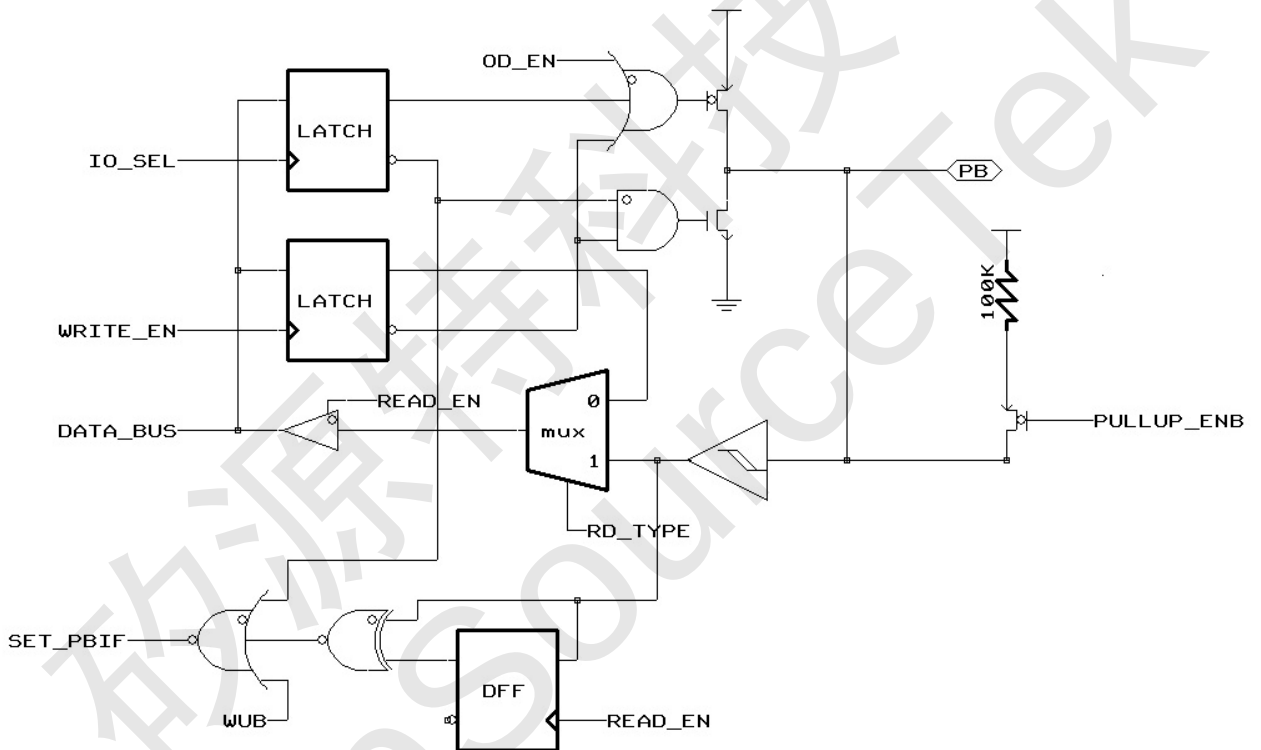


图 14 PB4, 5 引脚结构框图



3.6 定时器 0

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 的上升沿将让定时器 0 计数加一。当 T0CE 是 0，EX_CKIO 或 I_LRC/E_LXT 的下降沿将让定时器 0 计数加一。

如果寄存器 PS0WDT (T0MD[3]) 为 0，定时器 0 时钟源可以由预分频器 0 所分频，预分频器 0 会被指定到定时器 0，且会在 PS0WDT 设为 0 时清除 Timer0 与寄存器 PS0CV。寄存器 PS0SEL[2:0] (T0MD[2:0]) 决定预分频器 0 的预分频比，其数值从 1:2 到 1:256。

定时器 0 时钟源默认为指令时钟。如果外部时钟脚 EX_CKIO 或低频振荡 I_LRC/E_LXT 被用来当作定时器 0 时钟源，用户必须注意分频后的频率不能超过指令时钟，否则会导致错误计数。当 I_LRC/E_LXT 同时被当作定时器 0 时钟源与指令时钟，用户必须指定预分频器 0 到定时器 0，且须注意预分频器 0 的预分频比不得小于 4。当配置字节设定为异步 (Async.)，定时器 0 时外部钟源 EX_CKIO 频率就可高于指令时钟。

当定时器 0 上溢，寄存器 T0IF (INTF[0]) 将会设定为 1，以标明定时器 0 发生上溢中断。如果寄存器 T0IE (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 T0IF，T0IF 才会被清除。

定时器 0 与 WDT 的结构框图如下图：

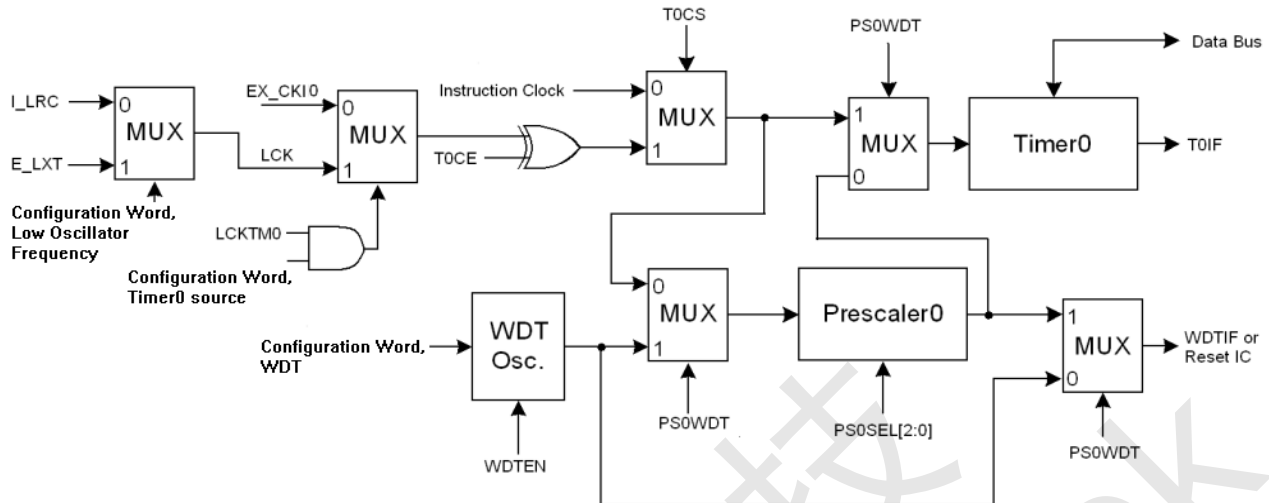


图 15 定时器 0 与WDT结构框图

3.7 定时器 1/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。写入定时器 1 高 2 位 (TMRH[5:4]) 再写入 TMR1[7:0] 时，就可更新定时器 1 重载缓存寄存器。当 T1EN=0 时，定时器 1 重载缓存寄存器会立即写入定时器 1。当 T1EN=1 时，会等到定时器 1 下溢后，定时器 1 重载缓存寄存器才会写入定时器 1。读取寄存器 TMR1[9:0] 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：

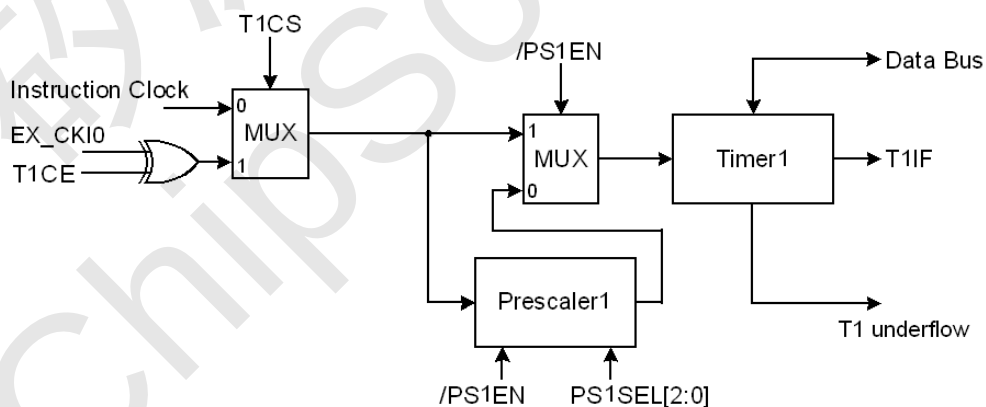


图 16 定时器 1 结构框图

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]) 可决定时钟源是指令时钟 F_{INST} 或外部时钟 EX_CKIO。当 T1CS 为 0，指令时钟会被选择当做时钟源。当 T1CS 为 1，则是 EX_CKIO 当做时钟源。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。当 T1CE 是 1，EX_CKIO 的下降沿将让定时器 1 计数减一。当 T1CE 是 0，EX_CKIO 的上升沿将让定时器 1 计数减一。定时器 1 时钟源可以由预分频器 1 所分频。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：单次计数与连续计数。当寄存器T1OS (T1CR1[2]) 为 1，即为单次计数模式。定时器 1 从储存在寄存器TMR1[9:0]的初始值下数到 0x00，当下溢发生时，定时器 1 停止计数。当寄存器T1OS (T1CR1[2]) 为 0，即为连续计数模式。当下溢发生，寄存器T1RL (T1CR1[1]) 会决定计数的初始值。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，标明定时器 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：

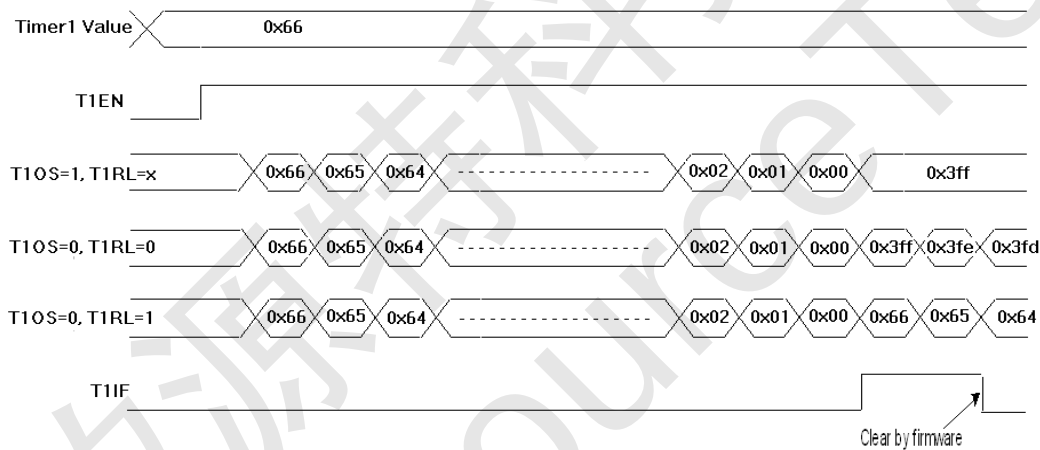


图 17 定时器 1 时序图

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[9:0] 为 0，PWM1 无法输出占空比。当PWM1DUTY[9:0]为 0x3FF，PWM1 将输出 1023/1024 的占空比（当PWM1OAL 为 0）。帧率是由TMRH[5:4] +TMR1[7:0]初始值所决定。因此，PWM1DUTY[9:0]数值必须小于或等于TMR1[9:0]。先写入PWM1 高 2 位 (TMRH[1:0]) 再写入PWM1DUTY[7:0]时，就可更新PWM1DUTY[9:0]重载缓存寄存器。等到定时器 1 下溢后，PWM1DUTY[9:0]重载缓存寄存器才会写入PWM1DUTY[9:0]。

PWM1 的结构框图如下：

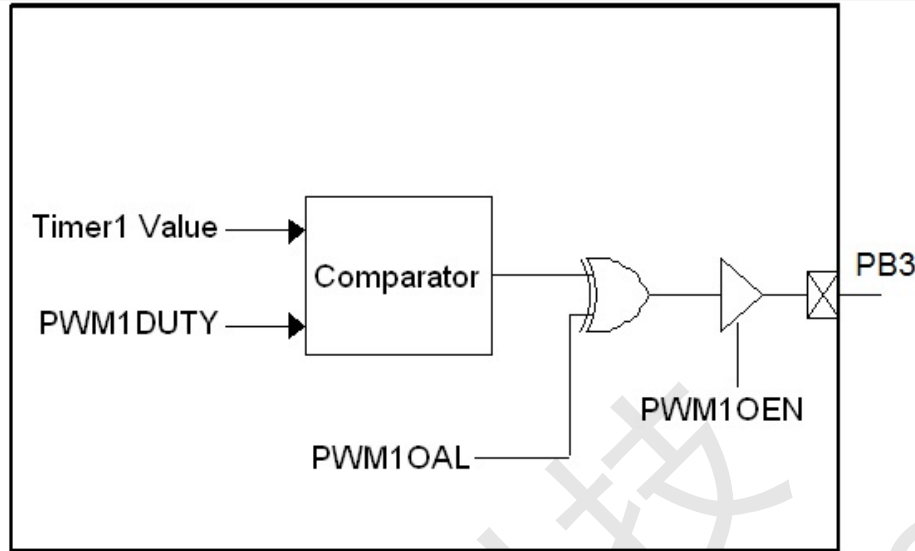


图 18 PWM1 结构框图

当寄存器BZ1EN (BZ1CR1[7]) 设定成 1 且使能配置字节, 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。预分频比的范围是 1:2 到 1:256。蜂鸣器 1 结构框图如下所示:

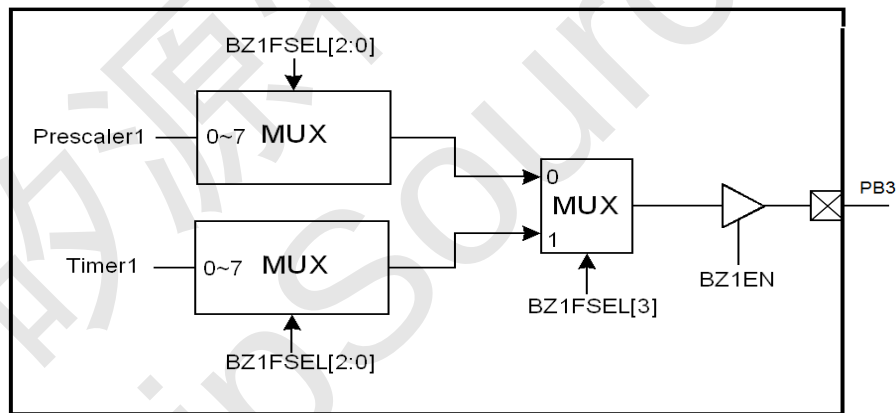


图 19 蜂鸣器 1 结构框图

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

3.8 定时器 2/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。写入定时器 2 高 2 位 (TMRH[7:6]) 再写入TMR2[7:0]时, 就可更新定时器 2 重载缓存寄存器。当T2EN=0 时, 定时器 2 重载缓存寄存器会立即写入定时器 2。当T2EN=1 时, 会等到定时器 2 下溢后, 定时器 2 重载缓存寄存器才会写入定时器 2。读取寄存器TMR2[9:0]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:

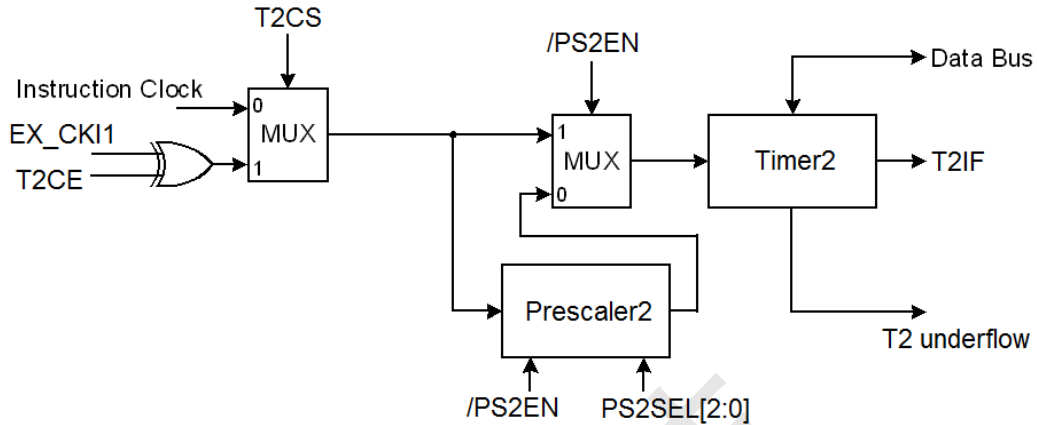


图 20 定时器 2 结构框图

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]) 可决定时钟源是指令时钟 F_{INST} 或外部时钟 EX_CK11。当 T2CS 为 0, 指令时钟会被选择当做时钟源。当 T2CS 为 1, 则是 EX_CK11 当做时钟源。当 EX_CK11 被选取, 寄存器控制位 T2CE (T2CR2[4]) 可决定 EX_CK11 的时钟触发沿。当 T2CE 是 1, EX_CK11 的上升沿将让定时器 2 计数减一。当 T2CE 是 0, EX_CK11 的下降沿将让定时器 2 计数减一。

定时器 2 时钟源可以由预分频器 2 所分频。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: 单次计数与连续计数。当寄存器 T2OS (T2CR1[2]) 为 1, 即为单次计数模式。定时器 2 从储存在寄存器 TMR2[9:0] 的初始值下数到 0x00, 当下溢发生时, 定时器 2 停止计数。当寄存器 T2OS (T2CR1[2]) 为 0, 即为连续计数模式。当下溢发生, 寄存器 T2RL (T2CR1[1]) 会决定计数的初始值。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, 标明定时器 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:

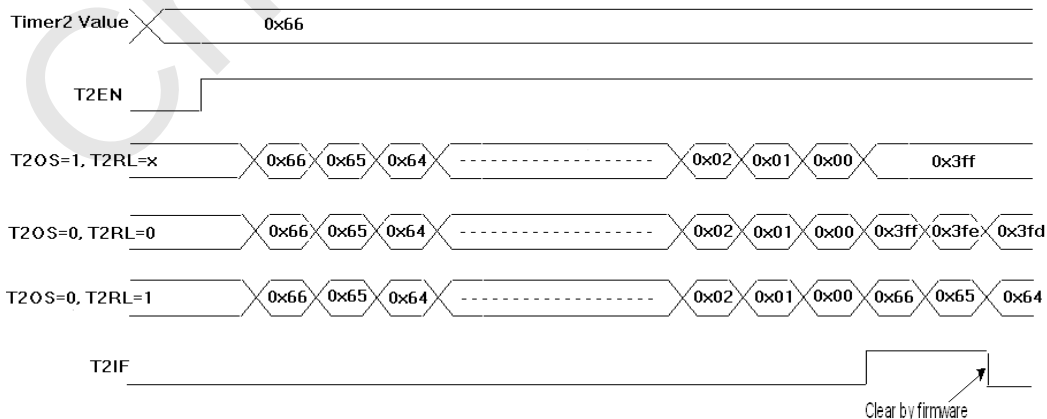


图 21 定时器 2 时序图



当寄存器PWM2OEN (T2CR1[7]) 设定为 1, PB2 为PWM2 输出。当PWM2OEN为 1, 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[9:0]为 0, PWM2 无法输出占空比。当PWM2DUTY[9:0]为 0x3FF, PWM2 将输出 1023/1024 的占空比(当PWM2OAL为 0)。帧率是由TMRH[7:6] +TMR2[7:0]初始值所决定。因此, PWM2DUTY[9:0]数值必须小于或等于TMR2[9:0]。先写入PWM2 高 2 位 (TMRH[3:2]) 再写入PWM2DUTY[7:0]时, 就可更新PWM2DUTY[9:0]重载缓存寄存器。等到定时器 2 下溢后, PWM2DUTY[9:0]重载缓存寄存器才会写入PWM2DUTY[9:0]。

PWM2 的结构框图如下:

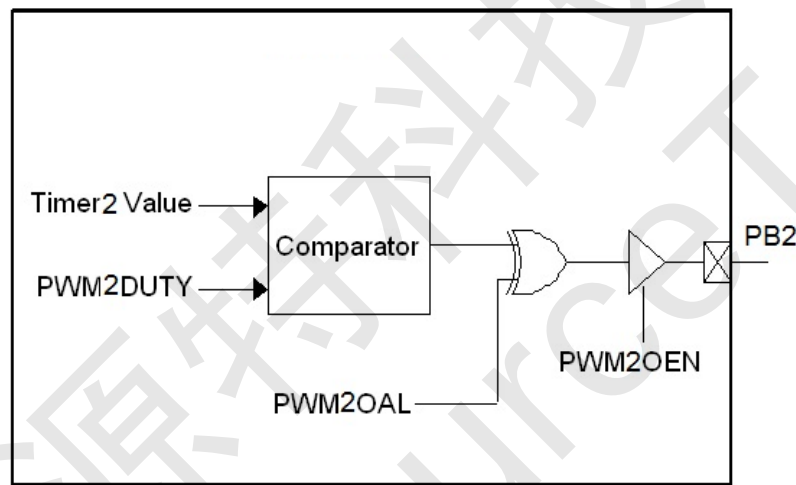


图 22 PWM2 结构框图

当寄存器BZ2EN (BZ2CR1[7]) 设定成 1 且使能配置字节, PB2 为蜂鸣器 2 输出。当BZ2EN设定为 1, 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。预分频比的范围是 1:2 到 1:256。蜂鸣器 2 结构框图如下所示:

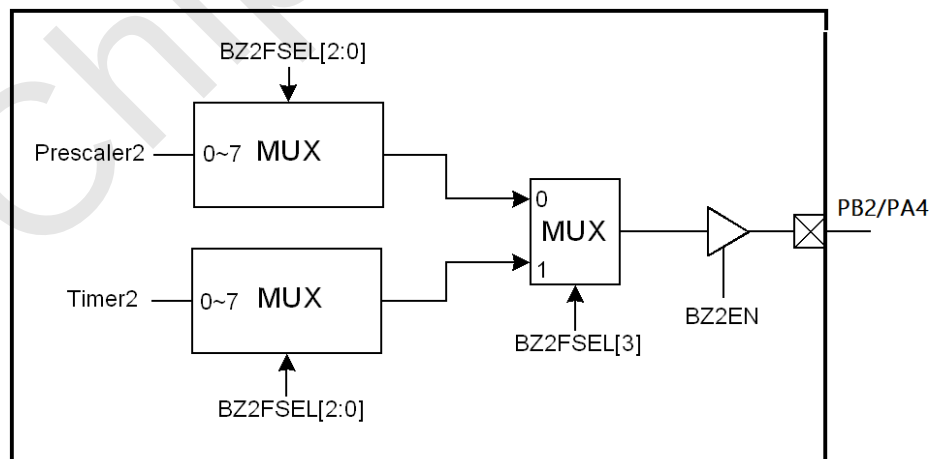


图 23 蜂鸣器 2 结构框图



3.9 定时器 3/PWM3/Buzzer3/PWM4

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。写入定时器 3 高 2 位 (TM3RH[5:4]) 再写入 TMR3[7:0] 时，就可更新定时器 3 重载缓存寄存器。当 T3EN=0 时，定时器 3 重载缓存寄存器会立即写入定时器 3。当 T3EN=1 时，会等到定时器 3 下溢后，定时器 3 重载缓存寄存器才会写入定时器 3。读取寄存器 TMR3[9:0] 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：

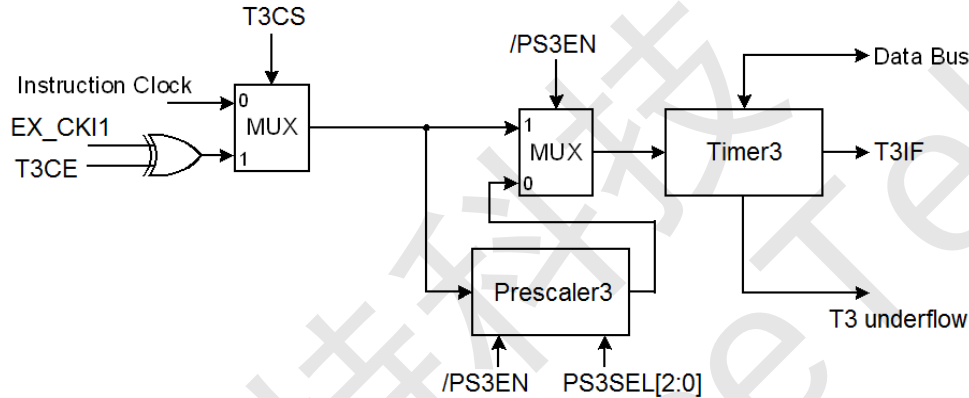


图 24 定时器 3 结构框图

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]) 可决定时钟源是指令时钟 F_{INST} 或外部时钟 EX_CK11。当 T3CS 为 0，指令时钟会被选择当做时钟源。当 T3CS 为 1，则是 EX_CK11 当做时钟源。当 EX_CK11 被选取，寄存器控制位 T3CE (T3CR2[4]) 可决定 EX_CK11 的时钟触发沿。当 T3CE 是 1，EX_CK11 的上升沿将让定时器 3 计数减一。当 T3CE 是 0，EX_CK11 的下降沿将让定时器 3 计数减一。定时器 3 时钟源可以由预分频器 3 所分频。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：单次计数与连续计数。当寄存器 T3OS (T3CR1[2]) 为 1，即为单次计数模式。定时器 3 从储存在寄存器 TMR3[9:0] 的初始值下数到 0x00，当下溢发生时，定时器 3 停止计数。当寄存器 T3OS (T3CR1[2]) 为 0，即为连续计数模式。当下溢发生，寄存器 T3RL (T3CR1[1]) 会决定计数的初始值。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，标明定时器 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：

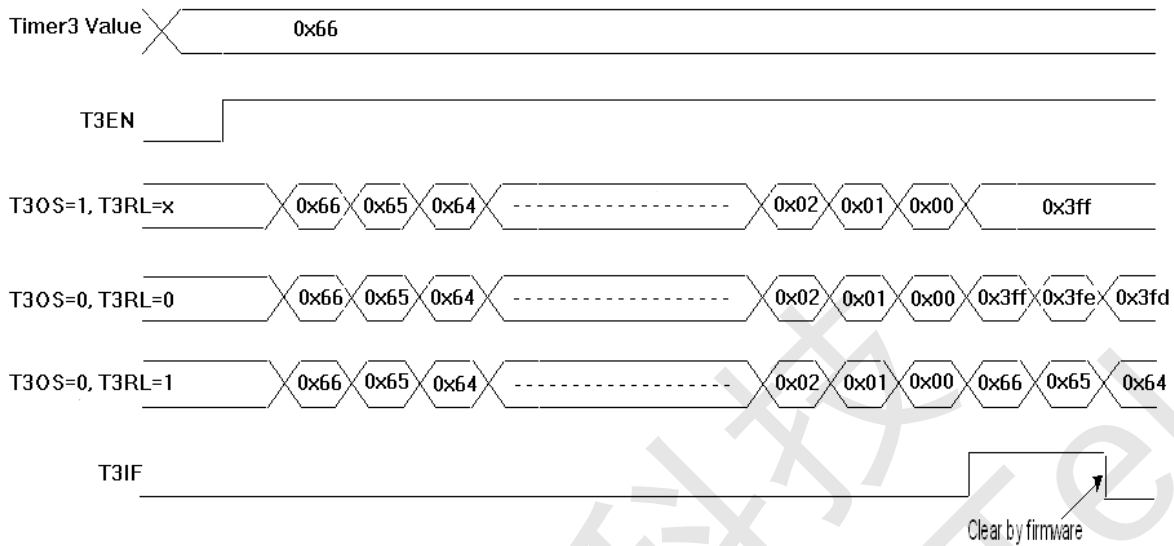


图 25 定时器 3 结构框图

当寄存器PWM3OEN（T3CR1[7]）设定为 1，PA2 为PWM3 输出。当PWM3OEN为 1，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。先写当PWM3DUTY[9:0]为 0，PWM3 无法输出占空比。当PWM3DUTY[9:0]为 0x3FF，PWM3 将输出 1023/1024 的占空比（当PWM3OAL为 0）。帧率是由TM3RH[5:4] +TMR3[7:0]初始值所决定。因此，PWM3DUTY[9:0]数值必须小于或等于TMR3[9:0]。先写入PWM3 高 2 位（TM3RH[1:0]）再写入PWM3DUTY[7:0]时，就可更新PWM3DUTY[9:0]重载缓存寄存器。等到定时器 3 下溢后，PWM3DUTY[9:0]重载缓存寄存器才会写入PWM3DUTY[9:0]。

PWM3 的结构框图如下：

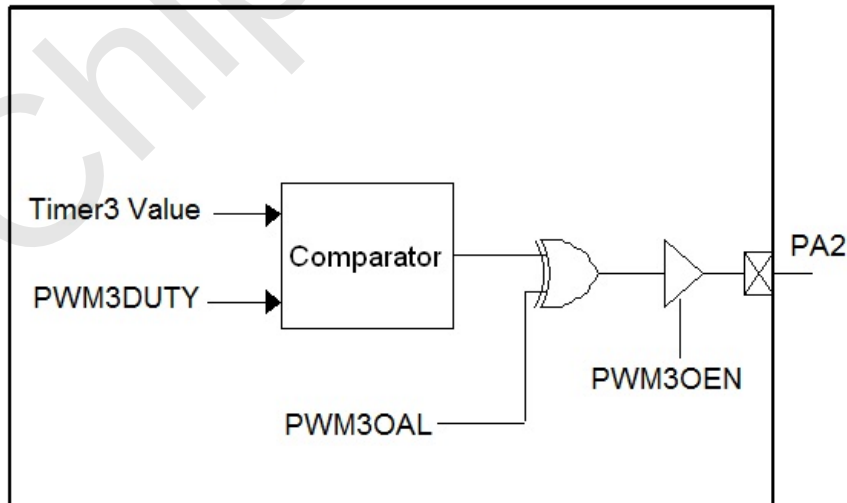


图 26 PWM3 结构框图



当寄存器BZ3EN (BZ3CR1[7]) 设定成 1 且使能配置字节, PA2 为蜂鸣器 3 输出。当BZ3EN设定为 1, 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。预分频比的范围是 1:2 到 1:256。蜂鸣器 3 结构框图如下所示:

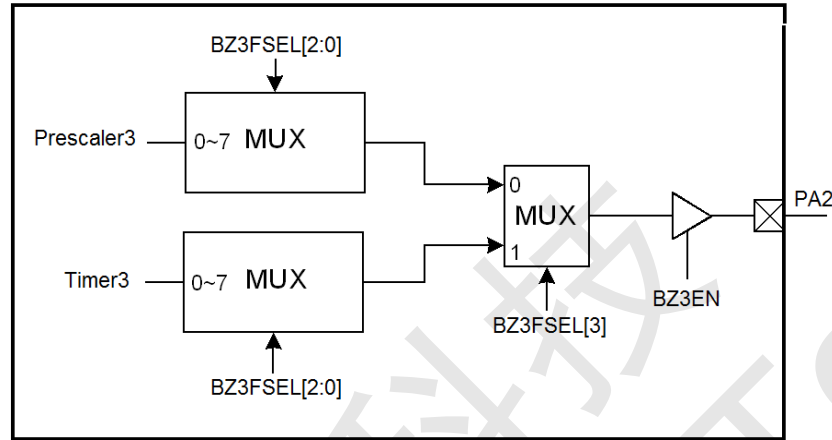


图 27 蜂鸣器 3 结构框图

当寄存器PWM4OEN (P4CR1[7]) 设定为 1, PA3 或PA7 为PWM4 输出。当PWM4OEN为 1, 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。先写当PWM4DUTY 为 0, PWM4 无法输出占空比。当PWM4DUTY为 0x3FF, PWM4 将输出 1023/1024 的占空比 (当PWM4OAL为 0)。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], 当Timer3 发生溢出时, PWM4 duty寄存器会更新。

PWM4 的结构框图如下:

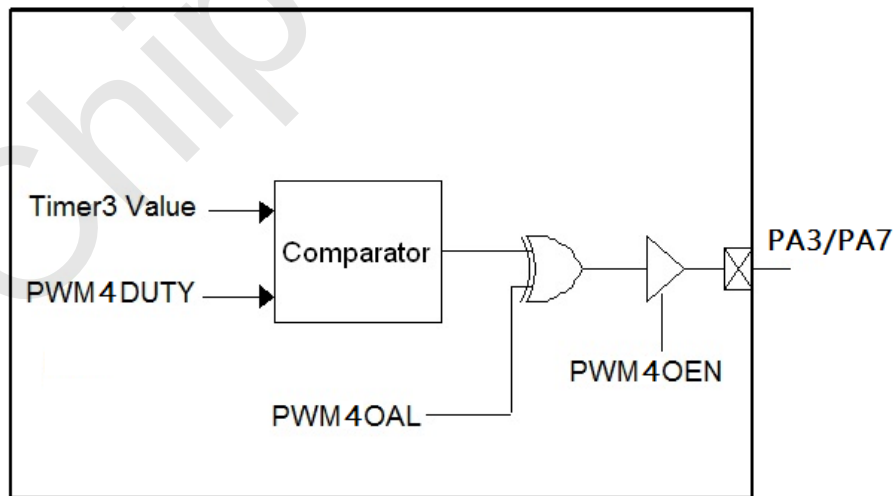


图 28 PWM4 结构框图



3.10 RFC（电阻/频率转换器模式）

ZH262F内置RFC功能，当开启RFC功能（RFCEN=1），选择的RFC输入引脚的电压电平将会控制定时器 1 的计数行为。只要RFC输入引脚电压低于 V_{IL} ，定时器 1 将会持续计数，等到RFC输入引脚电压高于 V_{IH} ，定时器 1 才会停止计数。下图显示RFC的功能模式：PSEL3~0 用来选择RFC输入引脚。RFCEN用于在正常的使能信号T1EN和RFC选择的输入状态之间切换Timer1 使能信号。

RFC模式的一个应用是测量电容-电阻充电时间，如下图所示，当PSEL3~0=0x01，PA1 为RFC输入引脚。首先设置PA1 输出 0（低于 V_{IL} ），接着设置TMR1[9:0]初始值后将PA1 设置为输入引脚，定时器 1 会开始下数。这时RC电路开始对PA1 引脚充电。当PA1 引脚电压高于 V_{IH} 时，定时器 1 会停止计数。定时器 1 相当于记录RC电路充电时间。

首先将PA1 设置为输出低(PA1 的电压放电为 0)，下一步清除Timer1 内容，将PA1 设置为输入，开启RFC模式。然后Timer1 开始计数，RC电路开始给PA1 充电。当PA1 充电到 V_{IH} 电压时，由于PA1 输入高，Timer1 计数停止。Timer1 内容将显示RC电路充电时间。(注:Timer1 是倒计时。)

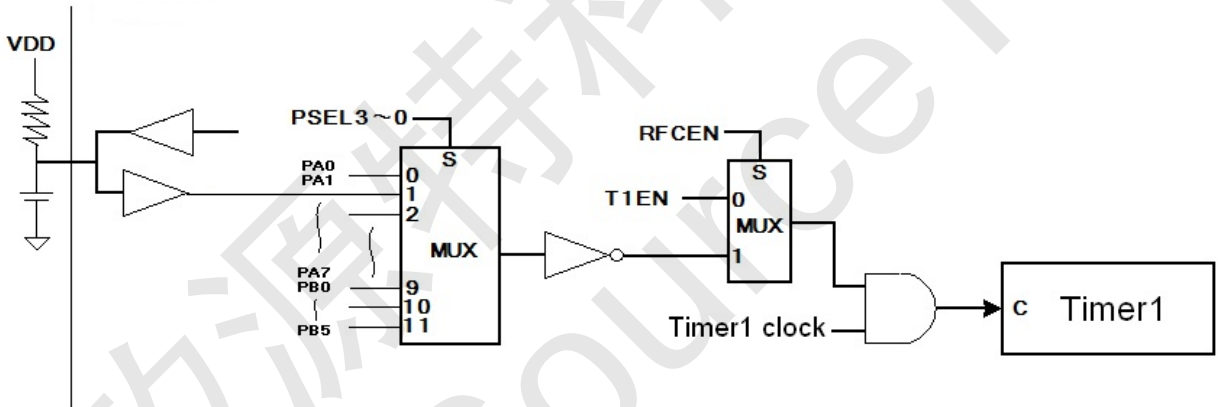


图 29 RFC结构框图

3.11 IR 载波

寄存器IREN（IRCR[0]）被设定为 1 后，PB1 为红外线载波输出，而PB1 会自动成为输出脚。当IREN清零，PB1 将会成为一般I/O脚。

红外线载波频率是由寄存器IRF57K（IRCR[1]）所选择。当IRF57K为 1，红外线载波频率是 57KHz。当IRF57K为 0，频率是 38KHz。由于红外载波频率是由 F_{HOSC} 的高频系统振荡推导出来的，所以当使用外晶体时，需要明确系统振荡的频率。寄存器IROSC358M（IRCR[7]）用于向ZH262F提供该信息。当IROSC358M为 1 时，外晶频率为 3.58MHz，当IROSC358M为 0 时，外晶频率为 455KHz。当采用内部高频振荡时，忽略该寄存器，并且为IR模块提供 4MHz的时钟。

红外线载波的极性会根据PB1 输出数据所决定。当寄存器IRCSEL（IRCR[2]）为 1 且PB1 输出数据为 0，红外线载波将由PB1 输出。当寄存器IRCSEL（IRCR[2]）为 0 且PB1 输出数据为 1，红外线载波将由PB1 输出。红外线载波的极性如下图所示：

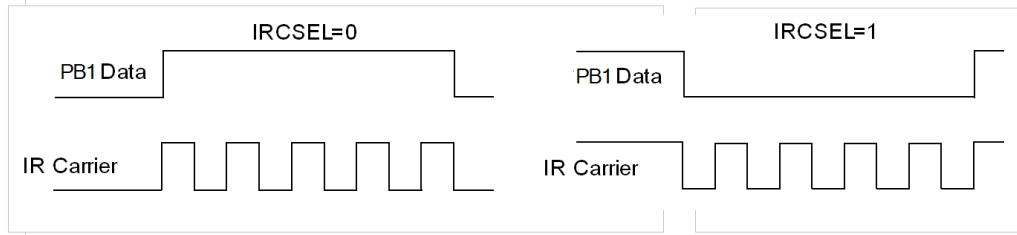


图 30 红外线载波的极性 vs. PB1 数据

3.12 低电压侦测 (LVD)

ZH262F 内置准确的低电压侦测电路来侦测VDD电压水平。将LV DEN 设为 1（寄存器PCON[5]）后，当VDD 电压低于下表LVDS[2:0]选择的电压值时，读取LV DOUT（寄存器PCON1[6]）会得到 0。如果开启LVD中断使能位且GIE=1 时，LVD中断标志位将会被设置为 1，程序将跳入中断子程序。LVD结构框图如下：

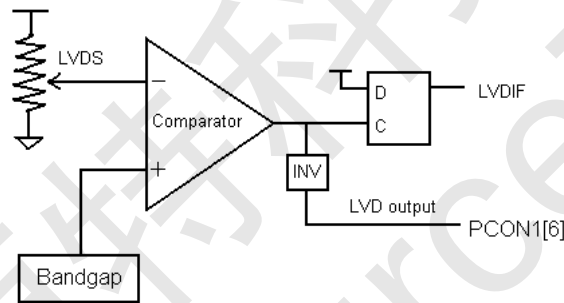


图 31 LVD结构框图

下表为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 2 LVD 电压选择

3.13 电压比较器

ZH262F 内置一组电压比较器。比较器的正输入源与负输入源和GPIO口复用，内部参考电压只能在P2V模式下 提供给比较器的负输入源使用。

CMPEN（寄存器ANAEN[7]）用来开启或关闭比较器，在睡眠模式（Halt mode）中比较器将自动关闭。



ZH262F的电压比较器有P2V或P2P两种模式，由VS[3:0]（寄存器CMPCR[3:0]）来选择。当VS[3:0]=0，比较器为P2P模式。当VS[3:0]=1~15，比较器为P2V模式。比较器正/负输入源如果为GPIO复用引脚，必须在配置字节（Configuration Word）中勾选为“Comparator input”。

P2V模式具有比较指定模拟板和指定参考板之间的电压的功能。P2V模式结构如下图所示：

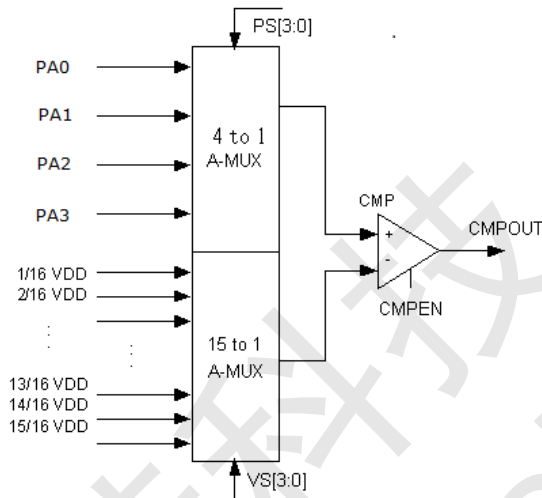


图 32 比较器P2V模式结构框图

在P2V模式中，比较器负输入源由VS[3:0]用来选择内部参考电压 1/16 VDD ~ 15/16 VDD。

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

表 23 选择P2V模式中比较器负输入源的内部参考电压

在P2V模式中，比较器正输入源由PS[3:0]（寄存器CMPCR[7:4]）用来选择PA3~0 中一根引脚。



表格如下：

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

表 24 P2V 模式引脚选择

P2P模式中，比较器正/负输入源都为外部引脚。须设置VS[3:0]=0，由PS[3:0]来选择 4 种外部输入引脚组合。选择表格如下：

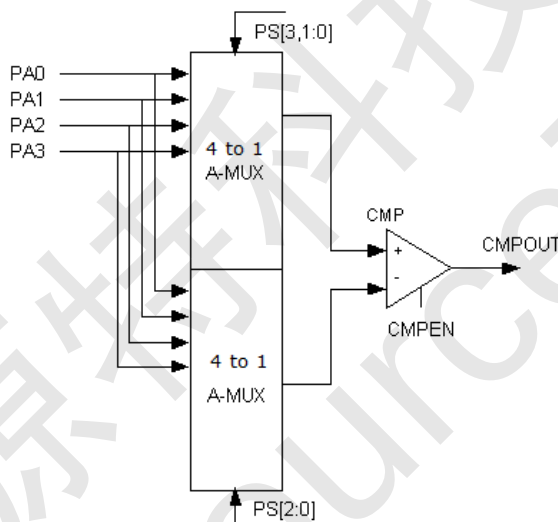


图 33 比较器P2P模式结构框图

PS[3:0]	负输入源	正输入源
0000	PA0	PA1
0001	PA1	PA0
0010	PA2	PA3
0011	PA3	PA2
0100~1111	-	-

表 25 P2P模式引脚选择

有三种方式可以取得比较器的比较结果。一是通过硬件中断设置，二是查询 CMPOUT（寄存器 OSCCR[7]），三是由 CMPO 引脚（PB3）输出。

使用比较器硬件中断功能时，须设置 CMPEN=1 与 CMPIE=1。先读取 CMPOUT（寄存器 OSCCR[7]）来清除上一次比较器比较结果，再清除中断标志位 CMPIF=0。等到下一次比较器输出翻转时，CMPIF 中断标志位将会被设置为 1。

设置 CMPOE（寄存器 OSCCR[6]）为 1，PB3 引脚将实时输出比较器结果，此时 PWM1 会自动关闭。



3.14 ADC模数转换器

ZH262F提供 11+1 通道 12 位ADC模数转换器。可将模拟信号转换为 12 位数字值。ADC参考电压可选外部引脚PA0 输入或由内部VDD, 4V, 3V, 2V提供。共有十一个外部模拟输入通道PA0~PA4 引脚与PB0~PB5, 也可选择内部 1/4VDD为模拟输入通道。ADC时钟 (ADCLK) 能够选择 $F_{INST}/1$, $F_{INST}/2$, $F_{INST}/8$ 或 $F_{INST}/16$ 四种。ADC采样时间可选择为 1 个ADCLK, 2 个ADCLK, 4 个ADCLK或 8 个ADCLK四种。先设置ADEN=1 后须等待 256us, 再将START写 1 来启动ADC模数转换。寄存器EOC=0 表示ADC还在转换中, EOC=1 表示ADC已完成一次模数转换。如果寄存器ADIE=1 与GIE设置为 1, 在EOC自动从 0→1 后, 中断标志ADIF位将被硬件设为 1 并处理此中断请求。结构框图如下:

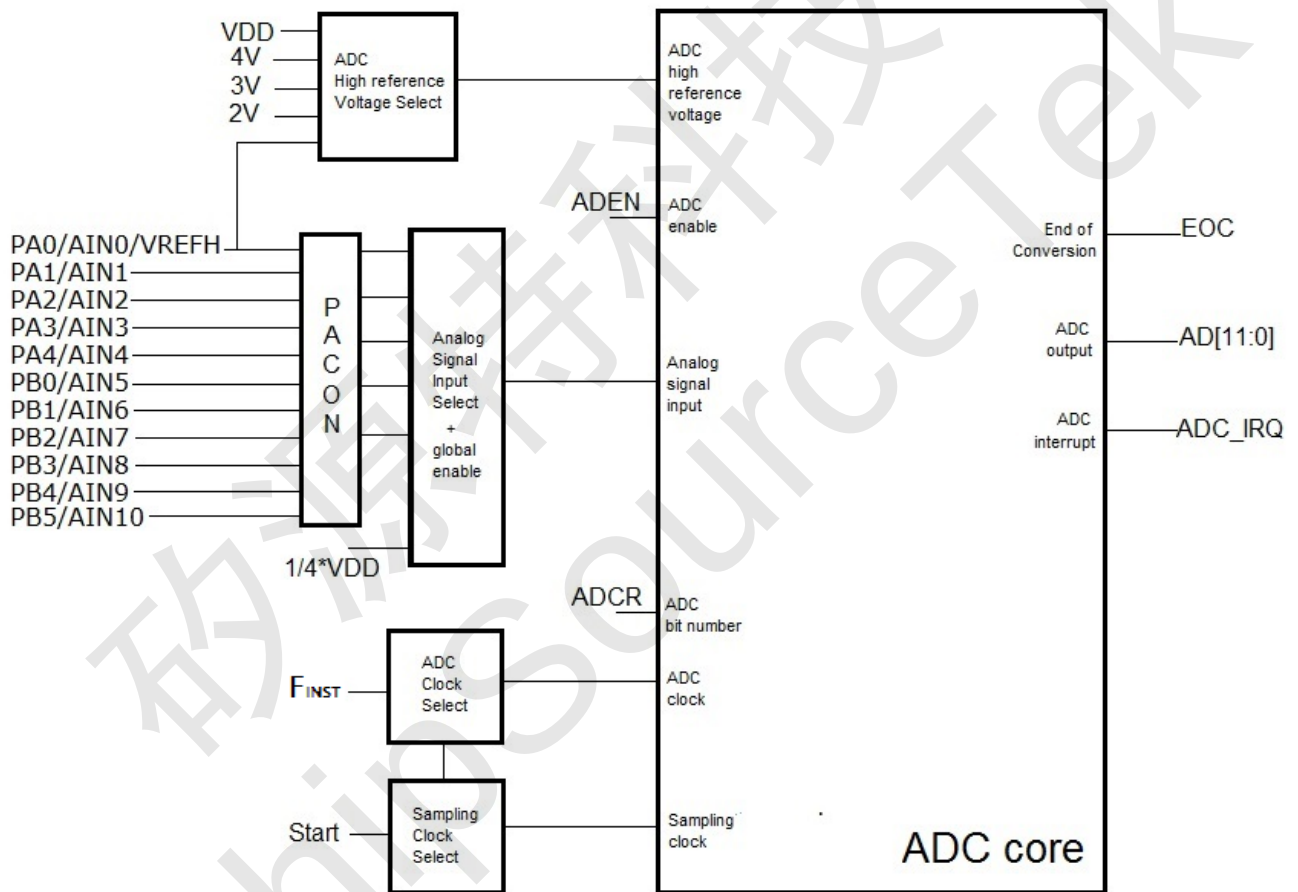


图 34 ADC 结构框图

3.14.1 ADC 参考电压

ADC可选 5 种高参考电压, 可由寄存器ADVREFH来设置。这些高参考电压源是一个外部电压源(PA0)和四个内部电压源(VDD, 4V, 3V, 2V)。当EVHENB=1, ADC参考电压由引脚 PA0 提供。PA0 输入的参考电压水平必须在VDD~2V之间。当EVHENB=0, ADC参考电压由VHS[1:0]决定。如果VHS[1:0] =11, ADC参考电压为VDD。如果VHS[1:0] =10, ADC参考电压为内部 4V。如果VHS[1:0] =01, ADC参考电压为内部 3V。如果VHS[1:0] =00, ADC参考电压为内部 2V。引脚VDD电压水平不得低于选择的ADC内部参考电压 (4V / 3V / 2V)。ADC模拟输入电压水平不得低于引脚VSS电压, 且不得高于ADC的参考电压。



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这个范围，ADC转换结果不是预期的值(满量程或零)。

EVHENB	VHS[1:0]	参考电压
1	x x	PA0
0	1 1	VDD
0	1 0	4V
0	0 1	3V
0	0 0	2V

表 26 选择ADC参考电压

3.14.2 ADC 模拟输入通道

ADC依据CHS[3:0]与GCHS来选择模拟输入通道。GCHS为所有模拟输入通道的总开关，任何模拟输入通道在转换前必须将GCHS设置为 1。

GCHS	CHS[3:0]	ADC模拟输入通道
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模拟输入通道选择

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]为PA[0:4]配置寄存器，PACON[5:7]为PB[0:2]配置寄存器，ADCR[4:6]为PB[3:5]配置寄存器，用来解决上述问题。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，以避免电流泄漏，不能作为正常的I/O使用。

除了设置PACON和ADCR[4:6]寄存器外，所选模拟输入引脚必须设置为输入模式，并且必须禁用内部的上拉/下拉，否则可能会影响模拟输入电平。



3.14.3 ADC 时钟（ADCLK），采样时钟（SHCLK）与位数选择

转换速度和转换精度受到ADC时钟(ADCLK)、采样时钟(SHCLK)和转换位数选择的影响。ADCLK是ADC的基本时钟。SAR ADC运行时，位操作与ADCLK同步。SHCLK是模拟信号采样时间的持续时间，SHCLK越大，恢复原始模拟信号电平越接近，但会降低ADC转换速度。反之亦然。ADC可以根据ADCR[1:0]寄存器位来选择不同的转换位数。有 2 位数字可供选择，分别是 12 位、10 位和 8 位。转换位数越少，ADC转换速度越快，但有效ADC位数越少。转换位数越多，转换速度越慢，但精度越高。

ADC时钟来源于F_{INST}，可从ADCK[1:0]进行选择。

ADCK[1:0]	ADC时钟频率
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

采样时钟由ADCLK得到，可由SHCK[1:0]选择：

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

表 29 ADC采样时间选择

ADC位数选择来自ADCR[1:0]。

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

表 30 ADC位数选择

ADC转换时间从START（寄存器ADMD[6]）写 1 开始一直到EOC从 0→1 为止。持续时间取决于ADC分辨率、ADC时钟速率和采样时间宽度。

ADC转换时间 ≈ ADC采样时间 + (ADC位数 + 2) * ADC时钟周期。

下表为ADC在不同条件下的转换时间与转换周期。

ADC位数	ADC时钟频率 (ADCLK)	ADC采样时间 (SHCLK)	ADC转换时间 (ADC时钟数)	F _{INST} =2MHz		F _{INST} =250K	
				转换时间	转换率	转换时间	转换率
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转换时间与转换率



3.14.4 ADC 操作顺序

依序设定ADC时钟(ADCLK), ADC采样时间(SHCLK), ADC位数(ADCR), ADC参考电压(寄存器ADVREFH), 选择模拟输入通道并将寄存器PACON相应位设置为 1, 再将ADEN位设置为 1。

在ADEN设置为 1 后必须等待 256us (ADC电路启动时间), 再将START位写 1 来启动ADC模数转换。ADC转换尚未完成时, 读取EOC位会得到 0。当ADC模数转换完成后会自动将EOC位设置为 1。

3.15 看门狗定时器 (WDT)

ZH262F中有独立振荡器被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能被配置字节开启或关闭。当WDT被配置字节开启时, 仍然可以通过WDTEN位(寄存器PCON[7])来开启/关闭。此外, WDT上溢后可由配置字节决定的复位ZH262F或发出的中断请求。同时, 在WDT上溢后, 寄存器/TO (STATUS[4]) 位将被清除为 0。

WDT上溢的时基由配置字节决定, 可以是 3.5 毫秒、15 毫秒、60 毫秒或 250 毫秒。如果将预分频器 0 分配给WDT, 则可以延长上溢周期。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。如果WDT上溢将复位ZH262F, 分频速率从 1:1 到 1:128, 如果选为WDT中断时, 则分频速率从 1:2 到 1:256。

当预分频器 0 分配给WDT时, 执行CLRWDWT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上溢后, 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。

3.16 中断

ZH262F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。
- 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 才能使能硬件中断功能。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ZH262F 自动清除为零，这将防止嵌套中断的发生。软件中断的中断服务程序最后一条指令必须是RETIE。执行此指令将设置GIE为 1 并返回中断前程序执行序列。

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，用户可以通过轮呼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，ZH262F将自动清除寄存器GIE位为零。如果用户想要实现嵌套中断，可以使用ENI指令作为中断服务程序的第一条指令，将GIE设置为 1，并允许其他中断事件再次中断ZH262F。指令RETIE必须是中断服务程序的最后一条指令，它将GIE设置为 1 并返回中断前程序执行序列。

用户应注意ENI指令不能放在RETIE指令之前，因为中断服务程序中的ENI指令将开启嵌套中断，但RETIE指令可能会误清除中断标志。

3.16.1 Timer0 上溢中断

Timer0 上溢（从 0x00 到 0xFF），如果T0IE和GIE设置为 1，寄存器T0IF位将被硬件设为 1 并处理此中断请求。

3.16.2 Timer1 下溢中断

Timer1 下溢（从 0x3FF到 0x00），如果T1IE和GIE设置为 1，寄存器T1IF位将被硬件设为 1 并处理此中断请求。

3.16.3 Timer2 下溢中断

Timer2 下溢（从 0x3FF到 0x00），如果T2IE和GIE设置为 1，寄存器T2IF位将被硬件设为 1 并处理此中断请求。

3.16.4 Timer3 下溢中断

Timer3 下溢（从 0x3FF到 0x00），如果T3IE和GIE设置为 1，寄存器T3IF位将被硬件设为 1 并处理此中断请求。

3.16.5 看门狗超时中断

当WDT上溢且配置字节选择WDT超时中断时，如果WDTIE和GIE设置为 1，寄存器WDTIF位将被硬件设为 1 并将处理此中断请求。

3.16.6 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。



3.16.7 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。

3.16.8 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。

3.16.9 低电压侦测中断

当VDD电压水平低于LVD电压，读取LVDOUT（寄存器PCON1[6]）会得到 0。如果LVDIE和GIE设置为 1，寄存器LVDIF位将被硬件设为 1 并处理此中断请求。

3.16.10 比较器输出翻转中断

当比较器输出状态翻转时，如果CMP1E和GIE设置为 1，寄存器CMP1IF位将被硬件设为 1 并处理此中断请求。请先读取CMP1OUT（寄存器OSCCR[7]）来清除上一次比较器比较结果。

3.16.11 ADC 模数转换完成中断

当ADC模数转换完成时，如果ADIE和GIE设置为 1，寄存器ADIF位将被硬件设为 1 并处理此中断请求。

3.17 振荡器配置

因为ZH262F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）。

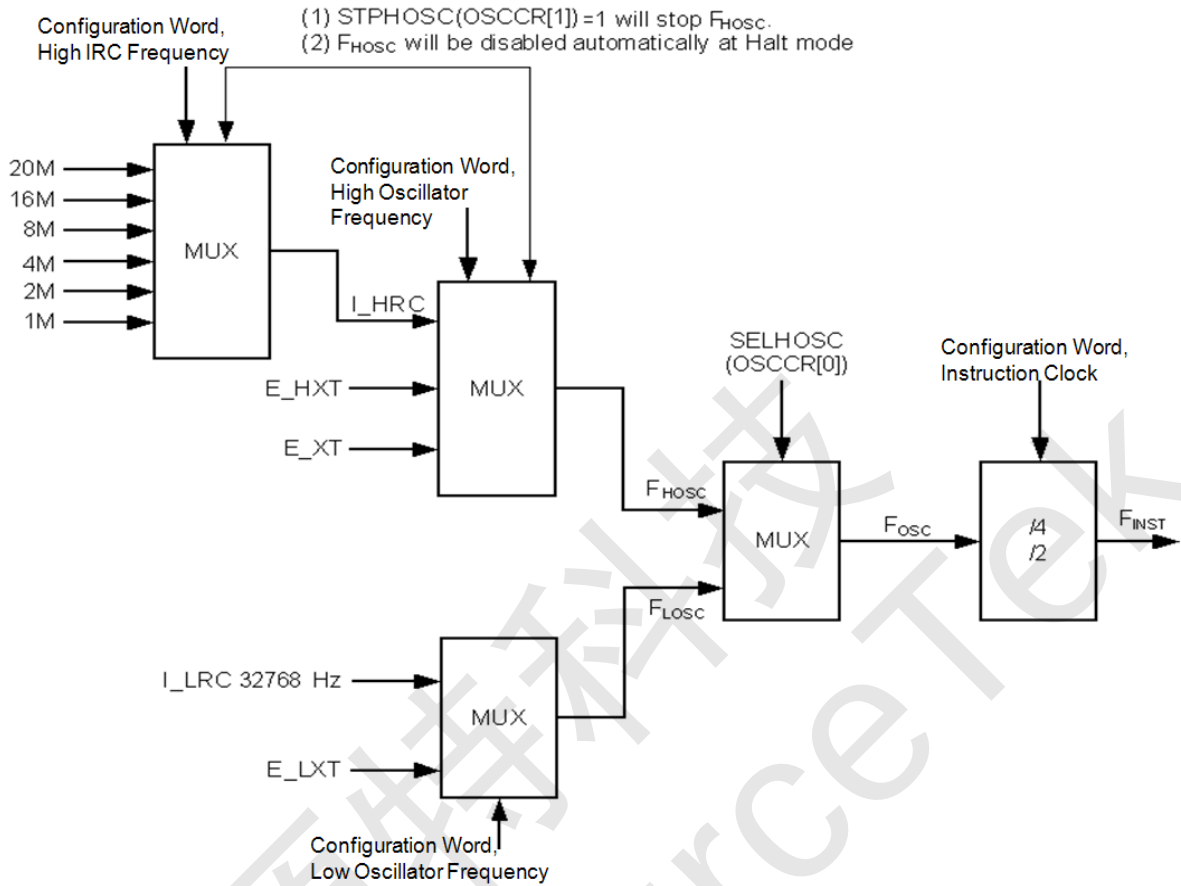


图 35 ZH262F振荡配置结构图

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节决定,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出脚和输入脚。PA7是晶体输出脚(Xout),PA6是晶体输入脚(Xin)。

有一个配置字节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。另外, 为了提供可靠的振荡, 建议一个电阻和两个电容连接如下图所示, 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 32768Hz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ZH262F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。

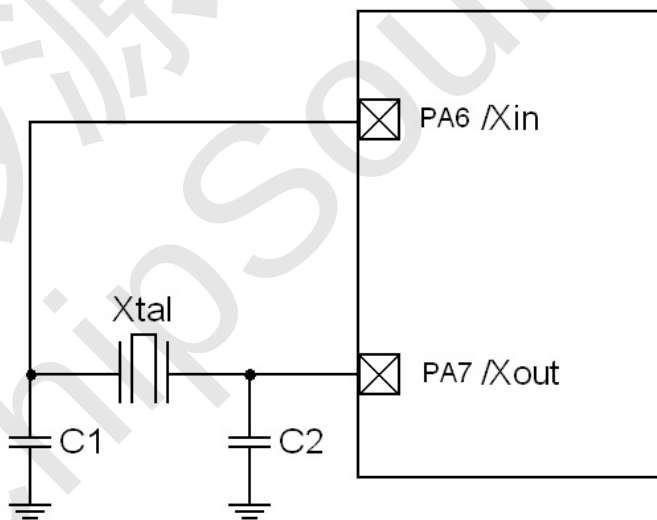


图 36 外部晶体振荡器的硬件连接图

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}, 根据配置字节设置, 指令时钟可以选择为F_{osc}/2 或F_{osc}/4。



3.18 工作模式

ZH262F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，ZH262F 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，ZH262F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。

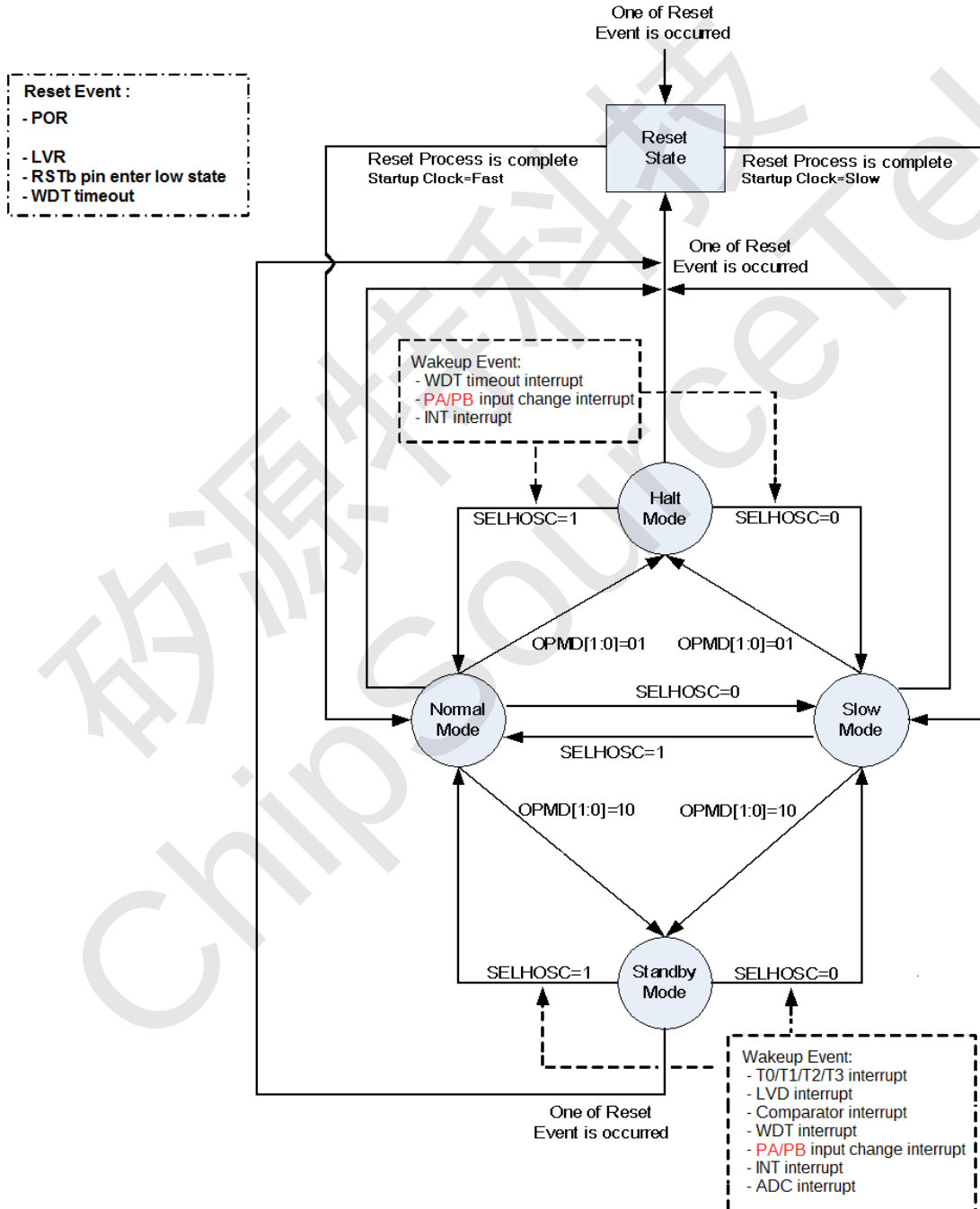


图 37 四种工作模式



3.18.1 正常模式

发生任何复位事件并且复位过程完成后，ZH262F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节决定。如果启动时钟为 F_{HOSC} ，ZH262F将进入正常模式，如果启动时钟为 F_{LOSC} ，ZH262F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ZH262F将进入正常模式。

- 指令的执行是基于 F_{HOSC} 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。
- 关于实时时钟的应用，ZH262F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LCKTM0 为 1 和配置字节中Timer0 时钟源来实现。

3.18.2 慢速模式

通过写 0 至寄存器SELHOSC位，ZH262F将进入慢速模式。在低速模式下，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ZH262F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。但需要注意的是，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。

3.18.3 待机模式

通过写入 10b至寄存器OPMD[1:0]，ZH262F将进入待机模式。然而，在待机模式下， F_{HOSC} 不会自动被ZH262F关闭，用户必须进入先低速模式后写入 1 至寄存器STPHOSC位，以停止 F_{HOSC} 。部分ZH262F 的硬件功能会被关闭，如T0EN / T1EN / T2EN / T3EN位被设置为 1 则定时器仍可运作。因此Timer0 / Timer1 / Timer2 / Timer3 溢出后ZH262F会被唤醒。

- 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 上溢中断 / Timer1 下溢中断 / Timer2 下溢中断 / Timer3 下溢中断 (b)看门狗超时中断
(c)PA/PB输入状态改变中断 (d)外部中断 0/1 (e)LVD中断 (f)比较器输出翻转中断 (g)ADC模数转换中断。
- 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。
- 不建议在同一时间进入待机模式并改变振荡模式（正常到慢速/慢速到正常）。



3.18.4 睡眠模式

通过执行SLEEP指令或写入 01b至寄存器OPMD[1:0]位，ZH262F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ZH262F只能通过一些特殊事件唤醒。因此，睡眠模式是ZH262F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)INT0/1 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：您可以在同一指令中更改STPHOSC并进入睡眠模式。**
- 不建议改变振荡模式（正常到慢速/慢速到正常），并在同一时间进入待机模式。

3.18.5 唤醒稳定时间

若外部晶体振荡器为系统振荡时钟来源，其睡眠模式的唤醒等待时间为 512*F_{osc}，若内部RC振荡器为系统振荡时钟来源，其睡眠模式的唤醒等待时间为 16*F_{osc}，由于待机模式下F_{HOSC}或F_{LOSC}仍在运行，因此无需为待机模式唤醒稳定时间。

在ZH262F进入待机模式或睡眠模式之前，用户可以执行指令ENI。在唤醒后，ZH262F将跳转到地址 0x008，以便执行中断服务程序。如果在进入待机模式或睡眠模式之前执行DISI指令，则在唤醒后执行下一条指令。

3.18.6 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	-	-	- Timer0/1/2/3 上溢 - WDT超时 - PA/PB 输入状态改变 -外部中断 0/1 - LVD 中断 -比较器中断 - ADC 中断	- WDT超时 - PA/PB输入状态改变 -外部中断 0/1

表 34 工作模式概述



- 二极管D使电容C能在VDD下电时快速放电。

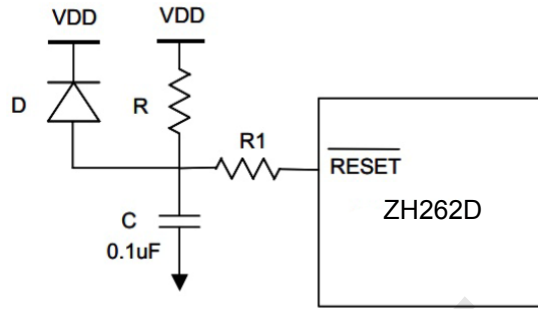


图 39 外部上电复位硬件连接图



4. ZH262F指令设置

ZH262F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			
算术指令					
ANDAR	R	d	dest = ACC & R	1	Z
IORAR	R	d	dest = ACC R	1	Z
XORAR	R	d	dest = ACC ⊕ R	1	Z
ANDIA	i		ACC = ACC & i	1	Z
IORIA	i		ACC = ACC i	1	Z
XORIA	i		ACC = ACC ⊕ 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
条件指令					
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	-
数据传送指令					
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			
算术指令					
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
其它指令					
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CLRWDWT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 0x00~0x7F。

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 CLRWDWT 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	ADDAR	Add ACC and R
语法	ADCAR R, d	语法	ADDAR 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	ADDAR 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.

ADCIA	Add ACC and Immediate with Carry	ADDIA	Add ACC and Immediate
语法	ADCIA 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	ADCIA i 执行指令前： ACC=0x12, i=0x34, C=1, 执行指令后： ACC=0x47, C=0.	举例	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	$0 \leq R \leq 127$ $0 \leq \text{bit} \leq 7$
操作	ACC & R $\rightarrow$ dest	操作	$0 \rightarrow R[\text{bit}]$
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	$0 \leq R \leq 127$ $0 \leq \text{bit} \leq 7$
操作	ACC & i $\rightarrow$ ACC	操作	$1 \rightarrow R[\text{bit}]$
状态影响	Z	状态影响	--
说明	ACC和8位立即数做“AND”运算。	说明	设置R寄存器的bit位为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		CALLA	
Test Bit in R and Skip if Clear		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[bit] = 0.	操作	PC + 1 → Top of Stack {TBHP, ACC} → 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		CLRA	
Test Bit in R and Skip if Set		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[bit] = 1.	操作	00h → ACC 1 → 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
语法	CLRR R
操作数	$0 \leq R \leq 127$
操作	00h $\rightarrow$ R 1 $\rightarrow$ Z
状态影响	Z
说明	寄存器R清零，Z标志位置“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.

COMR	Complement R
语法	COMR R, d
操作数	$0 \leq R \leq 127$ d = 0, 1.
操作	$\sim R \rightarrow \text{dest}$
状态影响	Z
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
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
语法	CLRWDT
操作数	--
操作	00h $\rightarrow$ WDT, 00h $\rightarrow$ WDT prescaler 1 $\rightarrow$ /TO 1 $\rightarrow$ /PD
状态影响	/TO, /PD
说明	清WDT计数器和预分频器；/TO和 /PD标志位置“1”。
周期	1
举例	CLRWDT 执行指令前： /TO=0 执行指令后： /TO=1

CMPAR	Compare ACC and R
语法	CMPAR R
操作数	$0 \leq R \leq 127$
操作	R - ACC $\rightarrow$ (No restore)
状态影响	Z, C
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
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

语法	DAA
操作数	--
操作	ACC(hex) → ACC(dec)
状态影响	C
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。
周期	1
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.

DECRSZ Decrease R, Skip if 0

语法	DECRSZ R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R - 1 \rightarrow \text{dest}$, Skip if result = 0
状态影响	--
说明	R 先- 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。
周期	1 or 2 (跳过)
举例	DECRSZ R, d 指令 2 指令 3 执行指令前: R=0x1, d=1, Z=0. 执行指令后: R=0x0, Z=1, 操作结果为 0, 指 令 2 被跳过。

DECR Decrease R

语法	DECR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R - 1 \rightarrow \text{dest}$
状态影响	Z
说明	R - 1, 若d="0", 结果存入ACC; 若d="1", 结果存入R。
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.

DISI Disable Interrupt Globally

语法	DISI
操作数	--
操作	Disable Interrupt, $0 \rightarrow \text{GIE}$
状态影响	--
说明	GIE 设置为 0, 关闭总中断。
周期	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	$R + 1 \rightarrow \text{dest.}$
状态影响	--	状态影响	Z
说明	GIE设置为 1，开启总中断。	说明	$R + 1$ ，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	$R + 1 \rightarrow \text{dest,}$ 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+ 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	$PC + 1 \rightarrow \text{Top of Stack}$, $001h \rightarrow PC$	操作	$ACC i \rightarrow ACC$
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	INT 执行指令前: PC=address of INT code 执行指令后: PC=0x01	举例	IORIA i 执行指令前: i=0x50, ACC=0xAA, Z=0. 执行指令后: ACC=0xFA, Z=0.
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	$ACC R \rightarrow \text{dest}$	操作	$ACC \rightarrow \text{F-page SFR}$
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	IORAR R, d 执行指令前: R=0x50, ACC=0xAA, d=1, Z=0. 执行指令后: R=0xFA, ACC=0xAA, Z=0.	举例	IOST F 执行指令前: F=0x55, ACC=0xAA. 执行指令后: F=0xAA, ACC=0xAA.



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		$0 \leq \text{adr} \leq 2047$	
操作		F-page SFR $\rightarrow$ ACC		操作		$\text{adr} \rightarrow \text{PC}[10:0]$.	
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		$0 \leq \text{adr} \leq 2047$		操作数		$0 \leq R \leq 127$	
操作		$\text{PC} + 1 \rightarrow \text{Top of Stack}$, $\text{adr} \rightarrow \text{PC}[10:0]$		操作		$\text{ACC} \rightarrow R$	
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
语法	MOVIA i
操作数	$0 \leq i < 255$
操作	$i \rightarrow \text{ACC}$
状态影响	--
说明	8 位立即数赋值给ACC。
周期	1
举例	MOVIA i 执行指令前: i=0x55, ACC=0xAA. 执行指令后: ACC=0x55.

NOP	No Operation
语法	NOP
操作数	--
操作	No operation.
状态影响	--
说明	空操作。
周期	1
举例	NOP 执行指令前: PC=A0 执行指令后: PC=A0+1

MOVR	Move R to ACC or R
语法	MOVR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R \rightarrow \text{dest}$
状态影响	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, 通过状态标位Z检查R是否为0。
周期	1
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.

RETIE	Return from Interrupt and Enable Interrupt Globally
语法	RETIE
操作数	--
操作	Top of Stack $\rightarrow$ PC $1 \rightarrow \text{GIE}$
状态影响	--
说明	中断返回, 栈顶地址载入PC同时使能中断。
周期	2
举例	RETIE 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

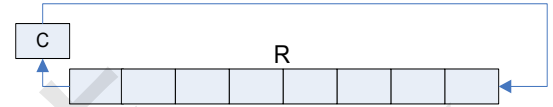
语法	RETIA i
操作数	$0 \leq i < 255$
操作	$i \rightarrow \text{ACC},$ $\text{Top of Stack} \rightarrow \text{PC}$
状态影响	--
说明	带参数返回：8 位立即数赋值给 ACC，栈顶地址载入 PC。

周期	2
举例	RETIA i 执行指令前： Stack pointer =2. i=0x55, ACC=0xAA. 执行指令后： PC=Stack[2], Stack pointer =1. ACC=0x55.

语法	
操作数	0 R 127 d = 0, 1.
操作	C dest[0]. R[7] C. R[6:0] dest[7:1]

状态影响	
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	
举例	RLR R, d 执行指令前： R=0xA5, d=1, C=0. 执行指令后：



RET Return from Subroutine

语法	RET
操作数	--
操作	$\text{Top of Stack} \rightarrow \text{PC}$
状态影响	--
说明	子程序返回，栈顶载入 PC。

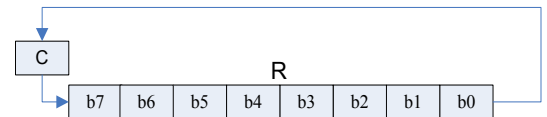
周期	2
举例	RET 执行指令前： Stack level=2. 执行指令后： PC=Stack[2], Stack level=1.

RRR Rotate Right R Through Carry

语法	RRR R, d
操作数	$0 \leq R \leq 127$ d = 0, 1.
操作	$C \rightarrow \text{dest}[7], R[7:1] \rightarrow \text{dest}[6:0],$ $R[0] \rightarrow C$

状态影响	C
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	
举例	执行指令前： R=0xA5, d=1, C=0. 执行指令后： R=0x52, C=1.





SBCAR	Subtract ACC and Carry from R
语法	SBCAR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R + (\sim \text{ACC}) + C \rightarrow \text{dest}$
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。

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

SBCIA	Subtract ACC and Carry from Immediate
语法	SBCIA i
操作数	$0 \leq i < 255$
操作	$i + (\sim \text{ACC}) + C \rightarrow \text{dest}$
状态影响	Z, DC, C
说明	常数和ACC带借位减法, 结果存入ACC。

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

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



SFUNR Move S-page SFR to ACC

语法	SFUNR S
操作数	$0 \leq S \leq 21$
操作	S-page SFR → ACC
状态影响	--
说明	读S-page特殊寄存器到ACC
周期	1
举例	SFUNR S 执行指令前: S=0x55, ACC=0xAA. 执行指令后: S=0x55, ACC=0x55.

SUBAR Subtract ACC from R

语法	SUBAR R, d
操作数	$0 \leq R \leq 127$ $d = 0, 1.$
操作	$R - ACC \rightarrow dest$
状态影响	Z, DC, C
说明	R 减去ACC, 若d="0", 结果存入ACC。 若d="1", 结果存入R。
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

语法	SLEEP
操作数	--
操作	00h → WDT, 00h → WDT prescaler 1 → /TO 0 → /PD
状态影响	/TO, /PD
说明	WDT和分频器 0 清零。/TO标志为0, /PD清零, IC进入睡眠。
周期	1
举例	SLEEP 执行指令前: /PD=1, /TO=0. 执行指令后: /PD=0, /TO=1.

SUBIA Subtract ACC from Immediate

语法	SUBIA i
操作数	$0 \leq i < 255$
操作	$i - ACC \rightarrow ACC$
状态影响	Z, DC, C
说明	8 位立即数减ACC, 结果存入ACC。
周期	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	$0 \leq R \leq 127$ $d = 0, 1.$	操作数	--
操作	$R[3:0] \rightarrow dest[7:4].$ $R[7:4] \rightarrow dest[3:0]$	操作	$ACC \rightarrow T0MD$
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	$ROM_data\{ TBHP, ACC \} [7:0] \rightarrow ACC$ $ROM_data\{TBHP, ACC\} [13:8] \rightarrow TBHD.$	操作	$T0MD \rightarrow ACC$
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. ZH262F配置字节表

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 _{HOSC} > 6MHz 3. 12MHz > F _{HOSC} ≥ 10MHz 5. 20MHz > F _{HOSC} ≥ 16MHz	2. 10MHz > F _{HOSC} ≥ 8MHz 4. 16MHz > F _{HOSC} ≥ 12MHz 6. 20MHz			
5	Instruction Clock 指令时钟	1. 2 oscillator period 2 个振荡周期	2. 4 oscillator period 4 个振荡周期			
6	WDT 看门狗定时器	1. Watchdog Enable (Software control) 2. Watchdog Disable (Always disable)	看门狗开启 (软件控制) 看门狗关闭 (永远关闭)			
7	WDT Event 看门狗定时器事件	1. Watchdog Reset	2. Watchdog Interrupt			
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. PA.5 is I/O PA5 是 I/O	2. PA.5 is reset PA5 是复位			
10	PA.7	1. PA.7 is I/O PA7 是 I/O	2. PA.7 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. 1.6V 6. 2.7V	2. 1.8V 7. 3.0V	3. 2.0V 8. 3.3V	4. 2.2V 9. 3.6V	5. 2.4V 10. 4.2V
15	VDD Voltage VDD 电压	1. 3.0V	2. 4.5V	3. 5.0V		
16	Sink current (exclude PA5) 灌电流 (不包括 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 不同步			
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			
23	Input High Voltage (V _{IH})	1. 0.7VDD	2. 0.5VDD			



项目	名称	选项		
	输入高电压 (VIH)			
24	Input Low Voltage (V _{IL}) 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

表 37 配置字节



6. ZH262F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 $F_{INST}=F_{HOSC}/4$, $F_{HOSC}=16MHz@I_HRC$, 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	--	3.3	--	5.5	V	$F_{INST}=20MHz @ I_HRC/2$
			2.2				$F_{INST}=20MHz @ I_HRC/4$
			2.7				$F_{INST}=16MHz @ E_HXT/2$
			2.0				$F_{INST}=16MHz @ E_HXT/4$
			2.0				$F_{INST}=8MHz @ I_HRC/4 \& I_HRC/2$
			1.8				$F_{INST}=8MHz @ E_HXT/4 \& E_HXT/2$
			1.8				$F_{INST}=4MHz @ I_HRC/4 \& I_HRC/2$
			1.6				$F_{INST}=4MHz @ E_XT/4 \& E_XT/2$
			1.6				$F_{INST}=32KHz @ I_LRC/4 \& I_LRC/2$
V_{IH}	输入高电平	5V	4.0	--	--	V	$RSTb (0.8 V_{DD})$
		3V	2.4	--	--		
		5V	3.5	--	--	V	所有I/O引脚, EX_CKIO/1, INT0/1 CMOS 选项 ($0.7 V_{DD}$)
		3V	2.1	--	--		
		5V	2.5	--	--	V	所有I/O引脚, EX_CKIO/1 TTL 选项 ($0.5 V_{DD}$)
		3V	1.5	--	--		
V_{IL}	输入低电平	5V	--	--	1.0	V	$RSTb (0.2 V_{DD})$
		3V	--	--	0.6		
		5V	--	--	1.5	V	所有I/O引脚, EX_CKIO/1, INT0/1 CMOS 选项 ($0.3 V_{DD}$)
		3V	--	--	0.9		
		5V	--	--	1.0	V	所有I/O引脚, EX_CKIO/1 TTL 选项 ($0.2 V_{DD}$)
		3V	--	--	0.6		
I_{OH}	输出高推电流 (小电流)	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	--	26	--	mA	$V_{OL}=1.0V$
		3V	--	16	--		
I_{IR}	IR 灌电流	5V	--	43	--	mA	$V_{OL}=1.0V$
		3V	--	28	--		



符号	参数	V _{DD}	最小值	典型值	最大值	单位	条件
I _{OP}	工作电流	正常模式					
		5V	--	1.7	--	mA	F _{HOSC} =20MHz @ I _{HRC} /2 & E _{HXT} /2
		3V	--	0.7	--		
		5V	--	1.4	--	mA	F _{HOSC} =20MHz @ I _{HRC} /4 & E _{HXT} /4
		3V	--	0.5	--		
		5V	--	1.6	--	mA	F _{HOSC} =16MHz @ I _{HRC} /2 & E _{HXT} /2
		3V	--	0.6	--		
		5V	--	1.3	--	mA	F _{HOSC} =16MHz @ I _{HRC} /4 & E _{HXT} /4
		3V	--	0.5	--		
		5V	--	1.3	--	mA	F _{HOSC} =8MHz @ I _{HRC} /2 & E _{HXT} /2
		3V	--	0.5	--		
		5V	--	1.1	--	mA	F _{HOSC} =8MHz @ I _{HRC} /4 & E _{HXT} /4
		3V	--	0.4	--		
		5V	--	1.1	--	mA	F _{HOSC} =4MHz @ I _{HRC} /2 & E _{XT} /2
		3V	--	0.4	--		
		5V	--	1.0	--	mA	F _{HOSC} =4MHz @ I _{HRC} /4 & E _{XT} /4
		3V	--	0.4	--		
		5V	--	1.0	--	mA	F _{HOSC} =1MHz @ I _{HRC} /2 & E _{XT} /2
		3V	--	0.3	--		
		5V	--	1.0	--	mA	F _{HOSC} =1MHz @ I _{HRC} /4 & E _{XT} /4
		3V	--	0.3	--		
		慢速模式					
		5V	--	11	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /2
		3V	--	6.1	--		
		5V	--	11	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ E _{LXT} /2
		3V	--	4.9	--		
		5V	--	7.3	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /4
		3V	--	4.3	--		
		5V	--	8.5	--	uA	F _{HOSC} 关闭, F _{LOSC} =32KHz @ E _{LXT} /4
		3V	--	3.6	--		
I _{STB}	待机电流	5V	--	3.8	--	uA	待机模式, F _{HOSC} 关闭, F _{LOSC} =32KHz @ I _{LRC} /4
		3V	--	2.6	--		
I _{HALT}	睡眠电流	5V	--	--	0.5	uA	睡眠模式 WDT关闭。
		3V	--	--	0.2		
		5V	--	--	5.0	uA	睡眠模式 WDT关闭。
		3V	--	--	3.0		
R _{PH}	上拉电阻	5V	--	50	--	KΩ	上拉电阻(不包括PA5)
		3V	--	100	--		
		5V	--	85	--	KΩ	上拉电阻(PA5)
		3V	--	85	--		
R _{PL}	下拉电阻	5V	--	50	--	KΩ	下拉电阻
		3V	--	100	--		



6.3 OSC电气特性

(测量条件VDD电压, T_A 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

(V_{DD}=5V, V_{SS}=0V, T_A=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 _{IVR}	比较器输入电压范围	0	--	5	V	F _{HOSC} =1MHz
T _{ENO}	比较器启动等待时间	--	20	--	us	F _{HOSC} =1MHz
I _{CO}	比较器电流消耗	--	135	--	uA	F _{HOSC} =1MHz, P2V mode
I _{LVD}	LVD电流消耗	--	150	--	uA	F _{HOSC} =1MHz, LVD=4.3V
E _{LVD}	LVD电压误差	--	--	3	%	F _{HOSC} =1MHz, LVD=4.3V

6.5 ADC电气特性

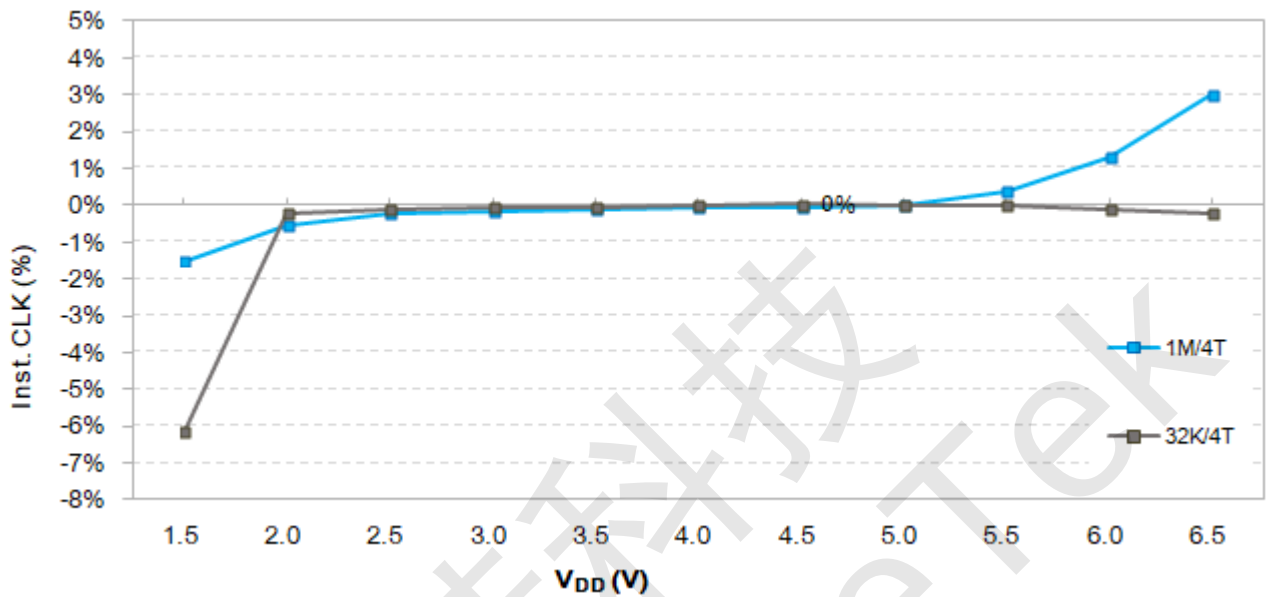
(V_{DD}=5V, V_{SS}=0V, T_A=25°C 除其他指定说明外)

符号	参数	最小值	典型值	最大值	单位	条件
V _{REFH}	模拟参考电压范围	2V	--	V _{DD}	V	外部参考电压
V _{REF4}	内部参考电压 4V, V _{DD} =5V	3.96	4	4.04	V	
V _{REF3}	内部参考电压 3V, V _{DD} =5V	2.97	3	3.03	V	
V _{REF2}	内部参考电压 2V, V _{DD} =5V	1.98	2	2.02	V	
V _{REF}	内部参考电压V _{DD} , V _{DD} =5V	--	V _{DD}	--	V	
	内部参考电压	V _{REF} +0.5	--	--	V	最小供电电压
	ADC 模拟输入电压	0	--	V _{REFH}	V	
I _{OP(ADC)}	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 _{sample}	ADC 转换率	--	--	125	K/sec	V _{DD} =5V
DNL	非线性微分误差	±1	--	--	LSB	V _{DD} =5.0V, AVREFH=5V, FADSMP=62.5K
INL	非线性积分误差	±2	--	--	LSB	
NMC	无缺码分辨率	10	11	12	Bits	

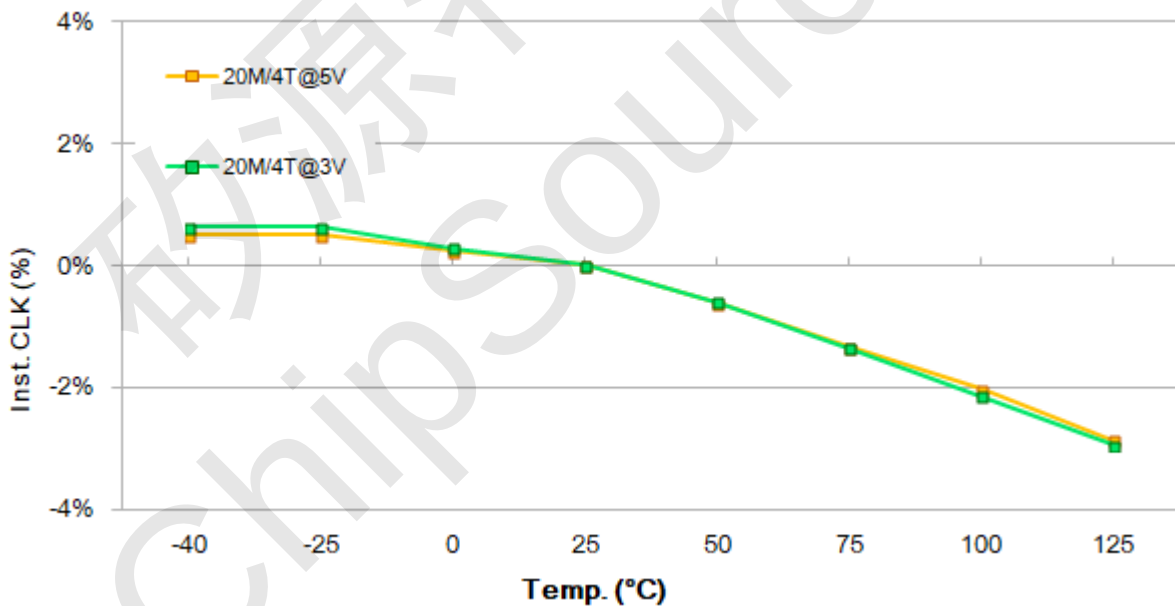


6.6 特性曲线图

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

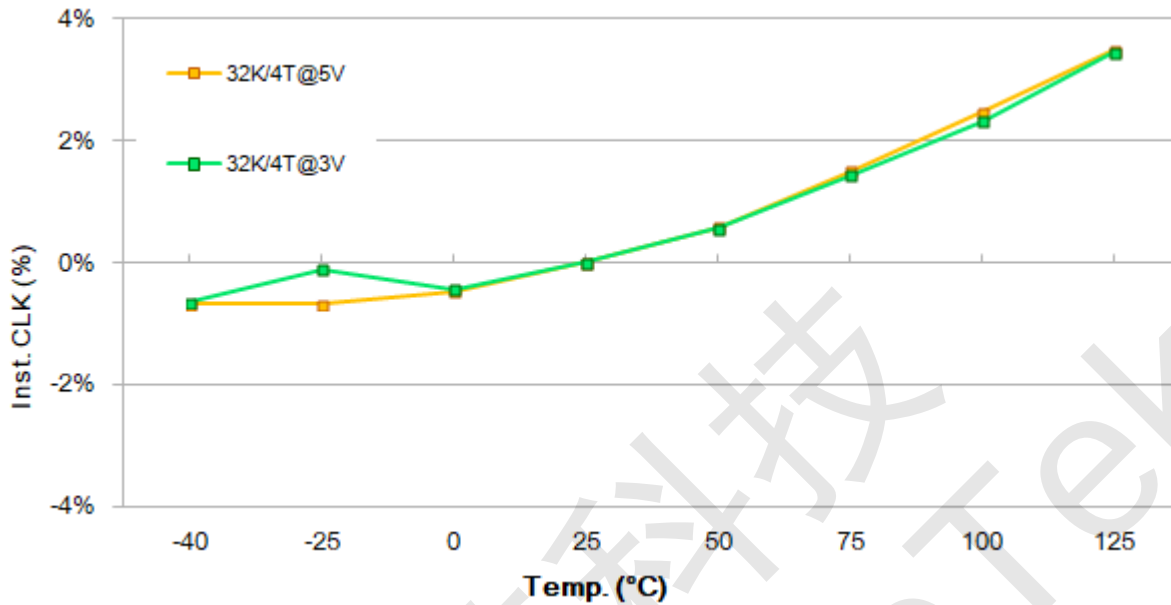


6.6.2 高速 RC 振荡频率(I_HRC)与温度曲线图

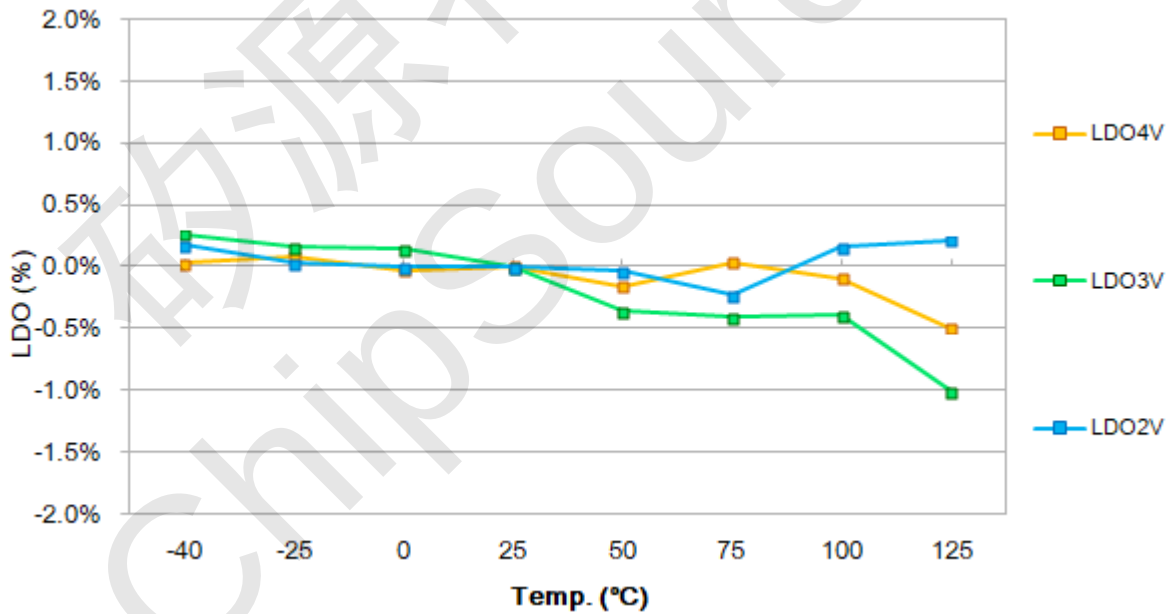




6.6.3 低速 RC 振荡频率(I_LRC)与温度曲线图

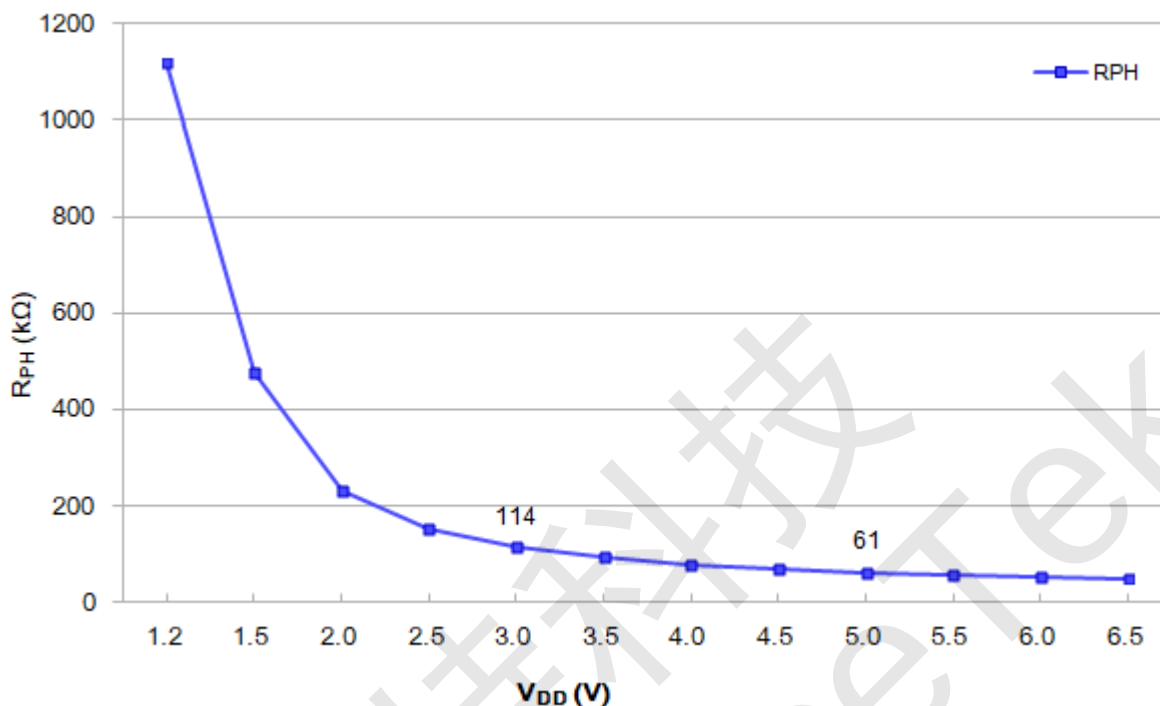


6.6.4 内部参考电压 LDO 与温度曲线图

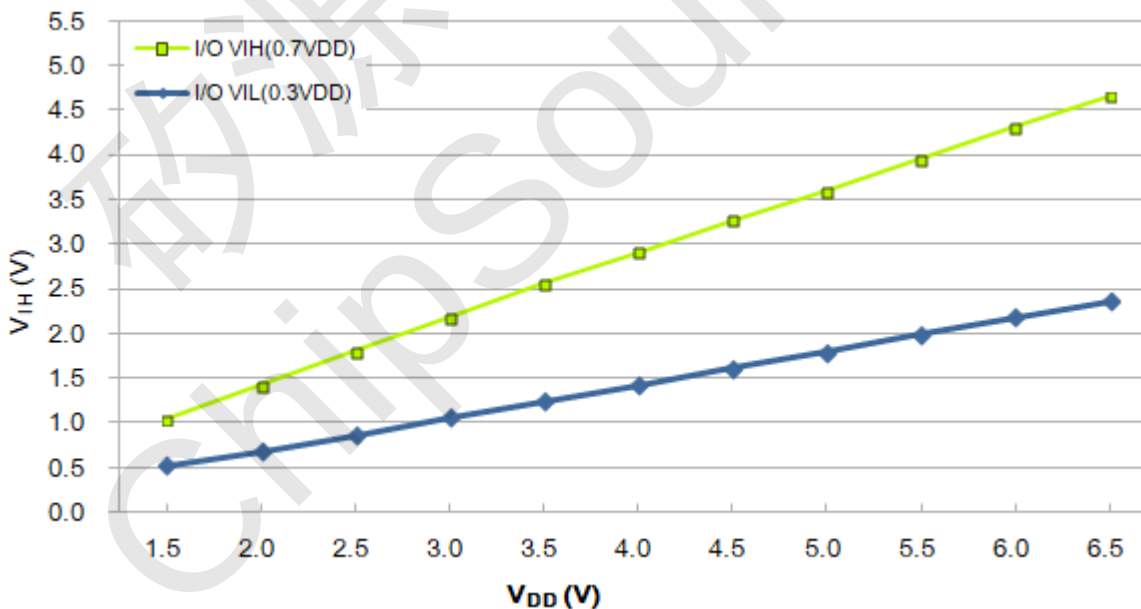


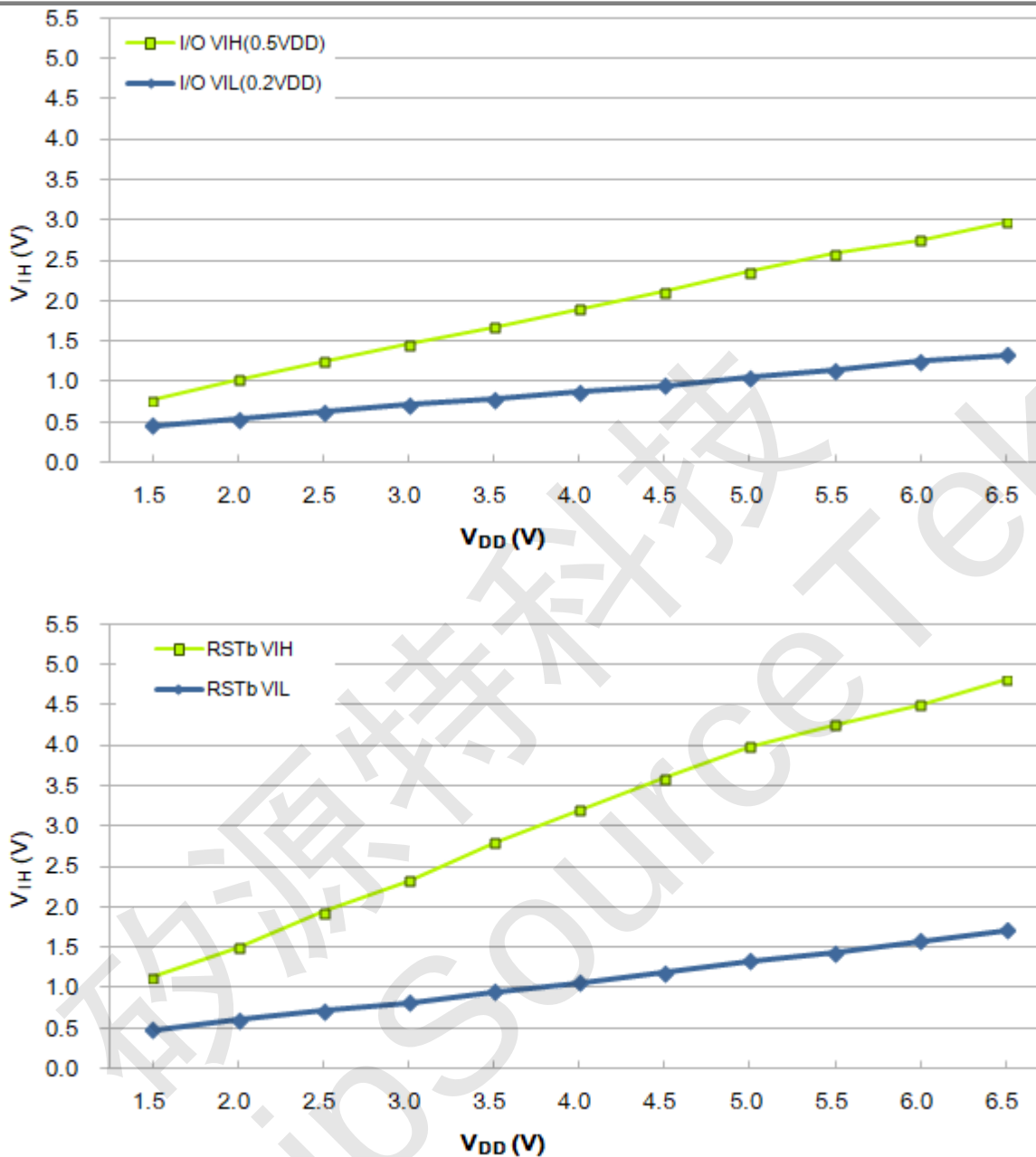


6.6.5 内部上拉电阻与电源电压(VDD)曲线图



6.6.6 V_{IH}/V_{IL} 与电源电压(VDD)曲线图





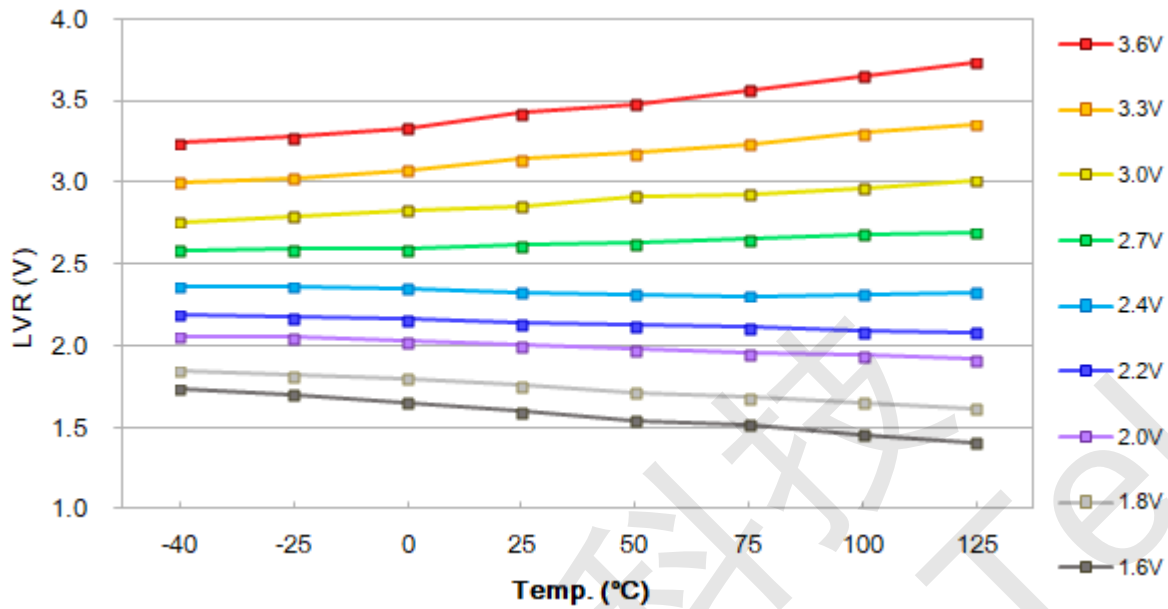
6.7 建议工作电压

建议工作电压（温度范围：-40 °C ~ +85 °C）

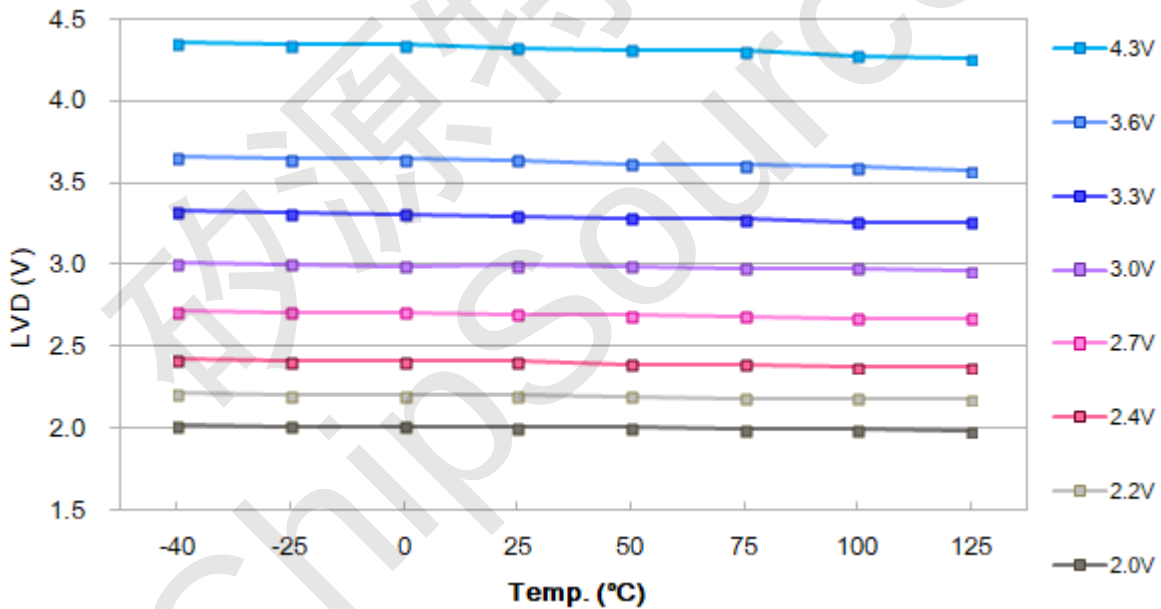
频率	最小电压	最大电压	LVR: 默认值 (25°C)	LVR: 建议值 (-40°C ~ +85°C)
20M/2T	3.3V	5.5V	3.6V	3.6V
16M/2T	3.0V	5.5V	3.3V	3.6V
20M/4T	2.2V	5.5V	2.4V	2.7V
16M/4T	2.0V	5.5V	2.2V	2.4V
8M(2T or 4T)	2.0V	5.5V	2.2V	2.4V
≤6M(2T or 4T)	1.8V	5.5V	2.0V	2.2V



6.8 LVR电压与温度曲线图

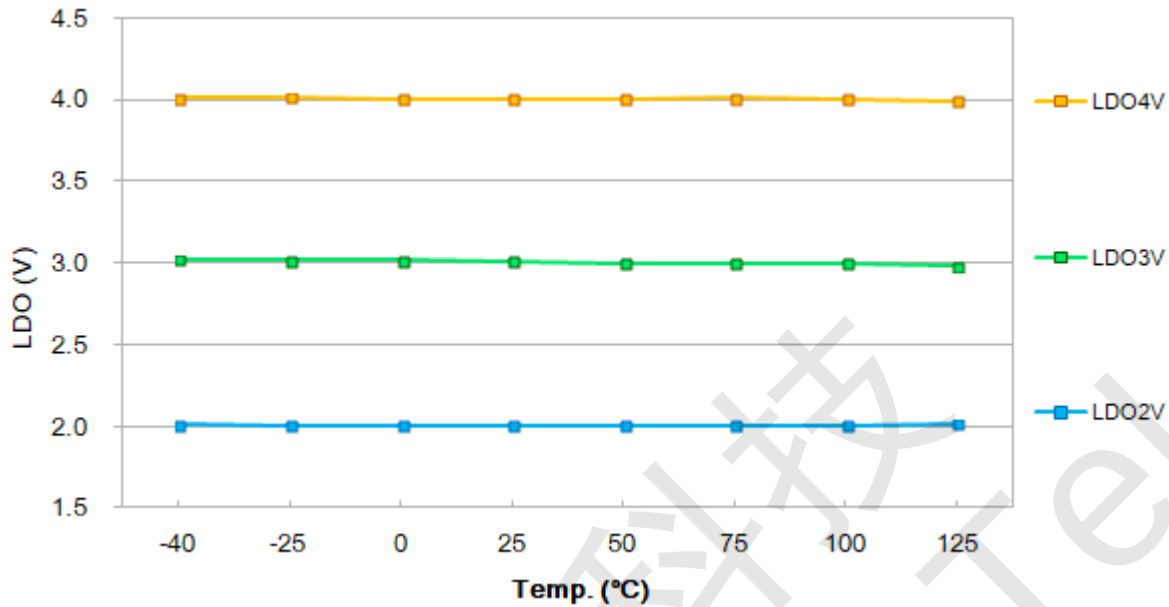


6.9 LVD电压与温度曲线图

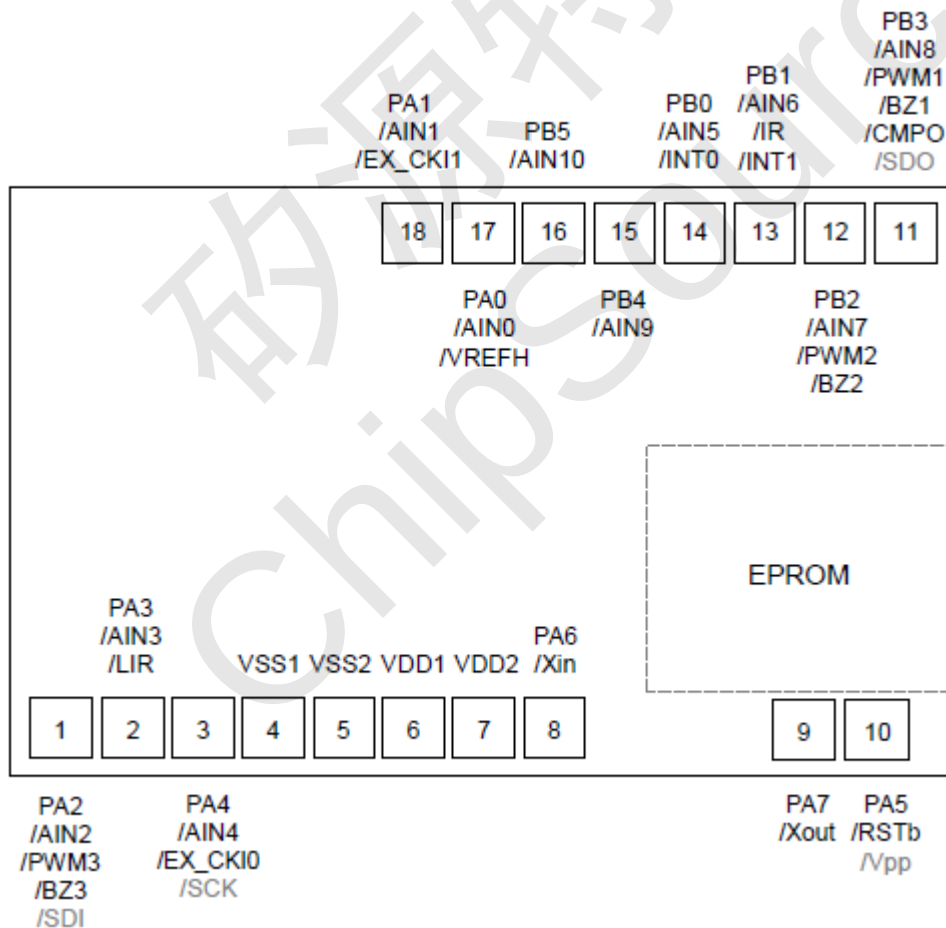




6.10 LDO与温度曲线图



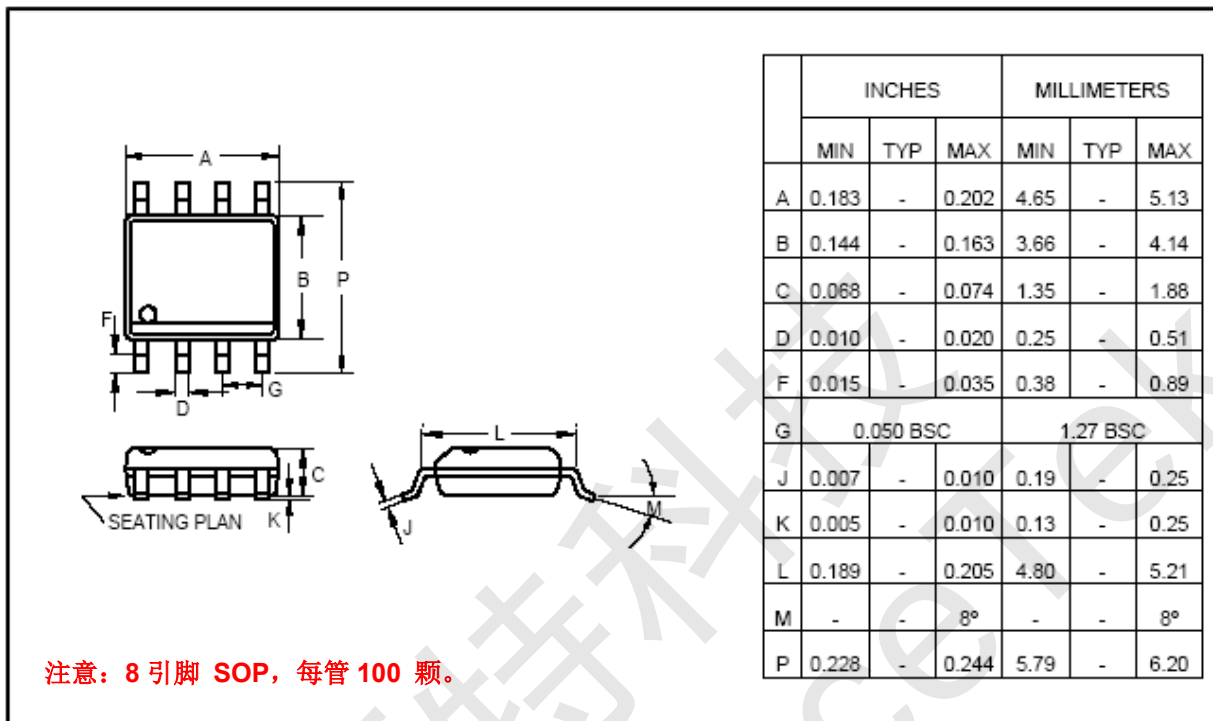
7. ZH262F晶圆脚位图



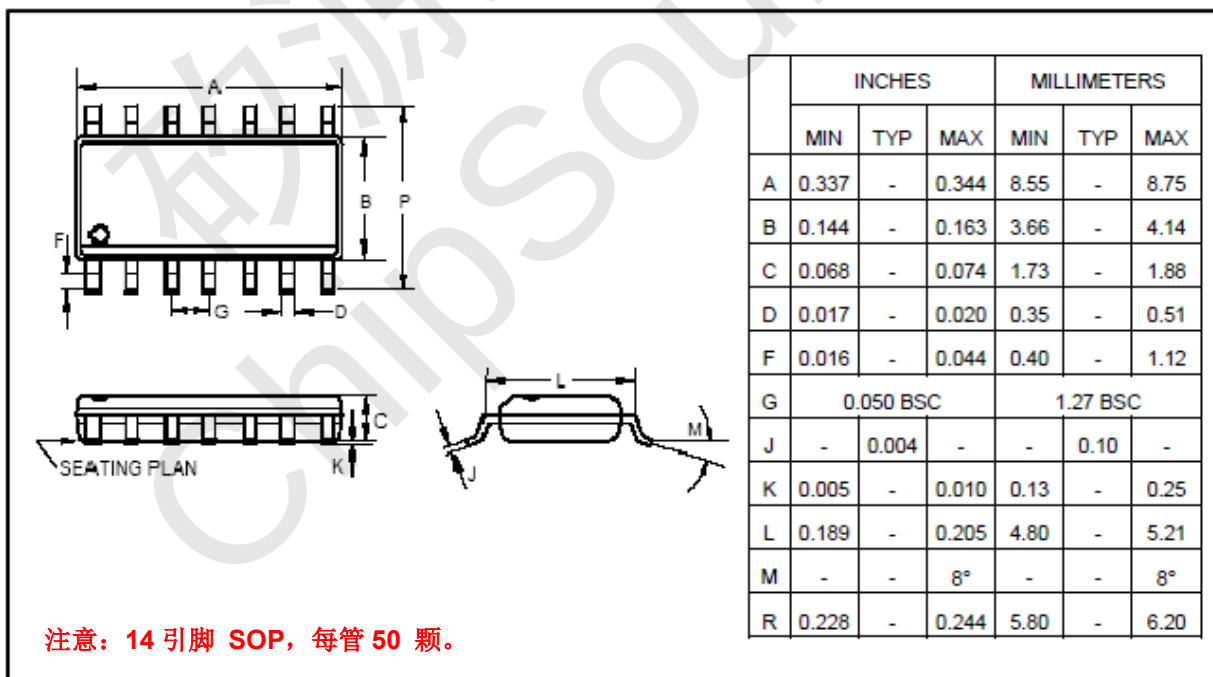


8. ZH262F封装尺寸

8.1 8 引脚SOP (150 毫寸)

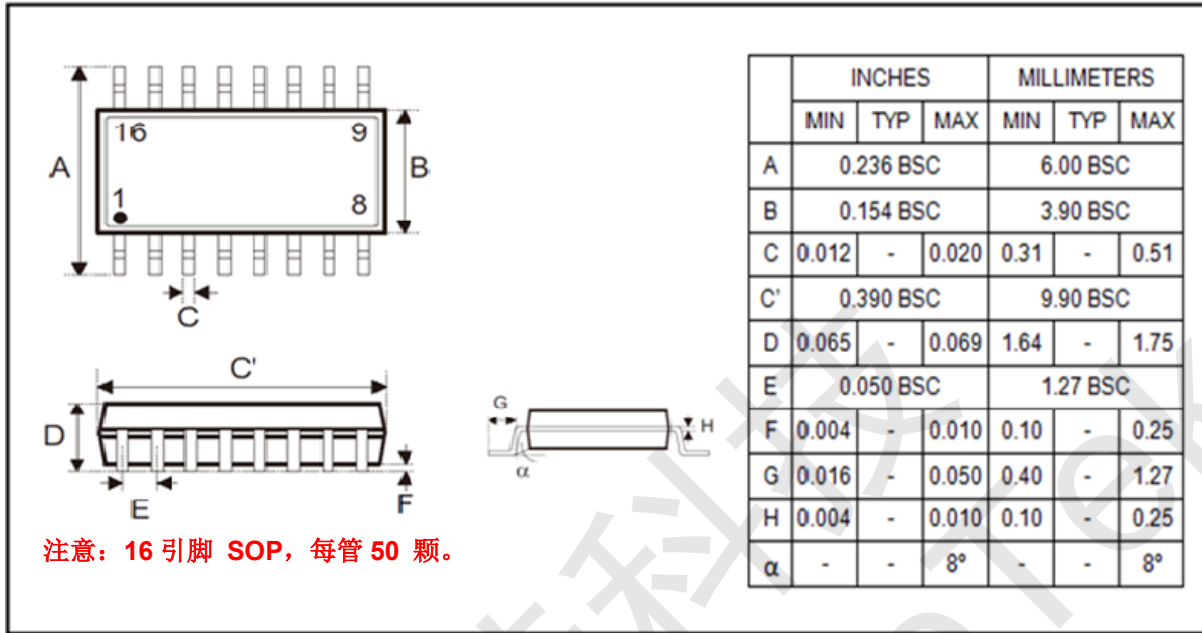


8.2 14 引脚SOP (150 毫寸)





8.3 16 引脚SOP (150 毫寸)



9. ZH262F订购信息

产品名称	封装类型	引脚数	封装尺寸	配送方式
ZH262F	Die	--	--	--
ZH262FS8	SOP	8	150 mil	卷带：每卷 2.5K。 管装：每管 100 颗。
ZH262FS14	SOP	14	150 mil	卷带：每卷 2.5K。 管装：每管 50 颗。
ZH262FS16	SOP	16	150 mil	卷带：每卷 2.5K。 管装：每管 50 颗。