

AT8A513K

8 位 EPROM-Based 6 I/O 单片机

Version 1.4

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	2025/02/27	新发布。	-
1.1	2025/05/26	更新比较器段落。	48
1.2	2025/06/09	更新VIH值。	7, 70
1.3	2025/08/26	1. 更新电压范围。	6
		2. 更新Vref硬件连接和LVD框图。	50
		3. 更新VDD（工作电压）的F _{INST} 定义。	70
		4. 更新推荐工作电压。	79
1.4	2025/11/28	1. 更新比较器/LVD特性。	72
		2. 更新推荐工作电压。	79

目 录

1. 概述	6
1.1 功能	6
1.2 方框图	8
1.3 引脚图	9
1.4 引脚说明	9
2. 内存结构	10
2.1 程序存储器	10
2.2 数据存储器	11
3. 功能概述	14
3.1 R页面特殊功能寄存器	14
3.1.1 <i>INDF</i> (间接寻址寄存器)	14
3.1.2 <i>TMR0</i> (定时器T0 寄存器)	14
3.1.3 <i>PCL</i> (程序计数器低字节)	14
3.1.4 <i>STATUS</i> (状态寄存器)	15
3.1.5 <i>FSR</i> (数据指针寄存器)	15
3.1.6 <i>PortB</i> (<i>PortB</i> 数据寄存器)	15
3.1.7 <i>PCON</i> (电源控制寄存器)	16
3.1.8 <i>BWUCON</i> (<i>PortB</i> 唤醒控制寄存器)	16
3.1.9 <i>PCHBUF</i> (程序计数器的高字节)	17
3.1.10 <i>BPLCON</i> (<i>PortB</i> 下拉电阻控制寄存器)	17
3.1.11 <i>BPHCON</i> (<i>PortB</i> 上拉电阻控制寄存器)	17
3.1.12 <i>INTE</i> (中断使能寄存器)	17
3.1.13 <i>INTF</i> (中断标志寄存器)	18
3.2 <i>T0MD</i> 寄存器	19
3.3 F页面特殊功能寄存器	20
3.3.1 <i>IOSTB</i> (<i>PortB</i> I/O控制寄存器)	20
3.3.2 <i>PS0CV</i> (预分频器P0 计数器值寄存器)	20
3.3.3 <i>BODCON</i> (<i>PortB</i> 开漏极寄存器)	20
3.3.4 <i>CMPCR</i> (比较器电压选择控制寄存器)	21
3.3.5 <i>PCON1</i> (电源控制寄存器1)	21
3.4 S页面特殊功能寄存器	22

3.4.1	TMR1 (定时器T1 寄存器)	22
3.4.2	T1CR1 (定时器T1 控制寄存器 1)	22
3.4.3	T1CR2 (定时器T1 控制寄存器 2)	23
3.4.4	PWM1DUTY (PWM1 占空比寄存器)	24
3.4.5	PS1CV (预分频器PS1 计数值寄存器)	24
3.4.6	BZ1CR (蜂鸣器BZ1 计数寄存器)	25
3.4.7	IRCR (IR控制寄存器)	25
3.4.8	TBHP (表格存取高字节地址指针寄存器)	26
3.4.9	TBHD (表格存取高字节数据寄存器)	26
3.4.10	P2CR1 (PWM2 控制寄存器 1)	27
3.4.11	PWM2DUTY (PWM2 占空比寄存器)	27
3.4.12	P3CR1 (PWM3 控制寄存器 1)	27
3.4.13	PWM3DUTY (PWM3 占空比寄存器)	28
3.4.14	OSCCR (振荡控制寄存器)	28
3.5	I/O端口	29
3.5.1	I/O引脚结构框图	30
3.6	定时器T0	36
3.7	Timer1/PWM1/Buzzer1	37
3.8	PWM2	39
3.9	PWM3	39
3.10	红外线载波	40
3.11	看门狗定时器 (WDT)	40
3.12	中断	41
3.12.1	Timer0 上溢中断	41
3.12.2	Timer1 下溢中断	41
3.12.3	WDT超时中断	42
3.12.4	PB输入状态改变中断	42
3.12.5	外部中断	42
3.12.6	LVD中断	42
3.13	振动配置	42
3.14	工作模式	44
3.14.1	正常模式	45
3.14.2	慢速模式	45
3.14.3	待机模式	45

3.14.4 睡眠模式	46
3.14.5 唤醒稳定时间	46
3.14.6 工作模式概述	46
3.15 复位	47
3.16 比较器相关功能	48
3.16.1 比较器功能简介	48
3.16.2 电压比较器	48
3.16.3 比较器参考电压 (V_{ref})	49
3.16.4 低电压检测 (LVD)	51
3.16.5 LVD中断	51
4. 指令设置	52
5. 配置表	68
6. 电气特性	70
6.1 最大绝对值	70
6.2 直流电气特性	70
6.3 OSC特性	72
6.4 比较器 / LVD 特性	72
6.5 特性图	72
6.5.1 频率与高速振荡电压	72
6.5.2 频率与高速振荡温度	73
6.5.3 频率与低速振荡电压	73
6.5.4 频率与低速振荡温度	74
6.5.5 上拉电阻与VDD	74
6.5.6 V_{IH}/V_{IL} 与VDD	75
6.5.7 V_{IH}/V_{IL} 与温度	77
6.6 建议工作电压	79
6.7 LVR与温度	79
7. 封装尺寸	80
7.1 8-Pin Plastic SOP (150 mil)	80
8. 订购信息	80

1. 概述

AT8A513K是以EPROM作为记忆体的 8 位微控制器，专为多IO产品的应用而设计，例如遥控器、风扇/灯光控制或是游戏机周边等等。采用CMOS制程并同时提供客户低成本、高性能等显著优势。AT8A513K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。

在I/O的资源方面，AT8A513K有 6 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，AT8A513K内建了可选择频率的红外载波发射口。

AT8A513K有两组计时器，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AT8A513K提供 3 组 8 位解析度的PWM输出或者蜂鸣器输出，可用来驱动马达、LED、或蜂鸣器等等。

AT8A513K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，AT8A513K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 32768Hz来计时。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AT8A513K进入正常操作模式(Normal) 或 慢速模式(Slow mode) 来处理突发事件。

AT8A513K内建低速晶振(32768Hz)电容，C_L有Disable、7pF、9pF、12.5pF四种选项可选，方便客户减少外部零件，仍可得到较精准的PPM值。

三路PWM1/PWM2/PWM3 共用Timer 1，可不经除级，PB3 复位脚可高推输出。

1.1 功能

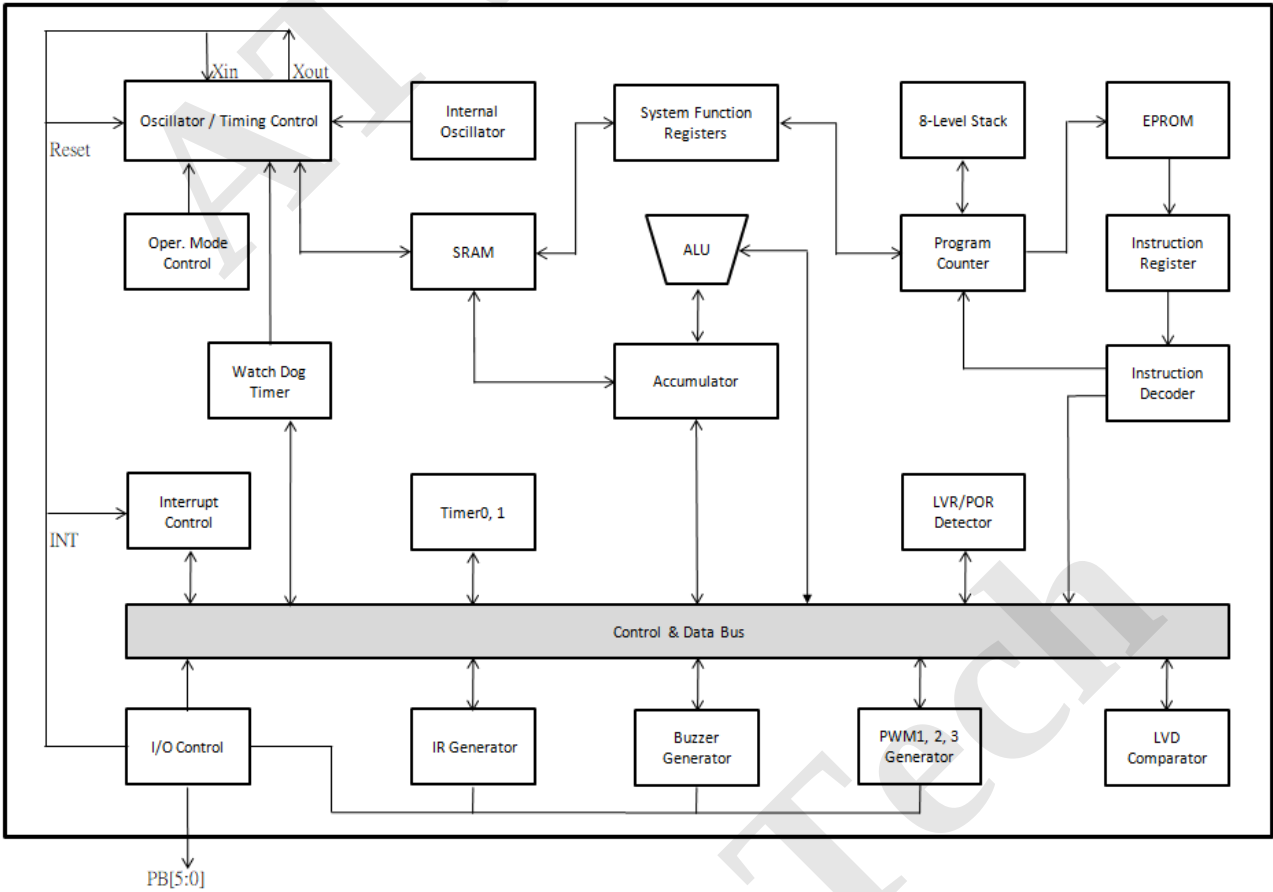
- 宽广的工作电压：（指令周期为 4 个CPU clock，即 4T模式）
 - 3.0V ~ 5.5V @20MHz/2T。
 - 1.6V ~ 5.5V @4MHz/4T。
- 宽广的工作温度：-40°C ~ 85°C。
- 1Kx14 bits EPROM。
- 48 bytes SRAM。
- 内建 14 级准确的低电压侦测电路功能。
- 6 根可分别单独控制输入输出方向的I/O脚(GPIO)、PB[5:0]。
- PB[3:0]可选择输入时使用内建上拉及下拉电阻。
- PB[5:0]可选择上拉电阻或开漏极输出(Open-Drain)。
- 输入有三种组态可选(TTL / CMOS / Without Schmitt)。
- 所有I/O脚输出可选择小灌电流(Small Sink Current) 或一般灌电流(Normal Sink Current)。所有I/O脚输出可选择小推电流(Small Drive Current)或一般推电流(Normal Drive Current)，PB3 只有一般推电流(Normal Drive Current)。

- V_{IH} 输入有三种组态可选, 分别为 $0.8V_{DD}$ 、 $0.6V_{DD}$ 及 $0.5V_{DD}$ (No Schmitt); V_{IL} 输入有三种组态可选, 分别为 $0.3V_{DD}$ 、 $0.2V_{DD}$ 及 $0.5V_{DD}$ (No Schmitt)。
- 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)包含可程序化的频率预除线路。
- 一组 8 位下数计时器(Timer1)可选重复载入或连续下数计时。可设定频率预除或频率不经除级。
- 三个 8 位脉冲宽度调变输出(PWM1/PWM2/PWM3)。
- 一个蜂鸣器输出(BZ1)。
- 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。
- 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: I_HRC (1~20MHz内部高速RC振荡)
E_XT (外部 455KHz~4MHz高速石英振荡)
 - 低速振荡: I_LRC (内部 32KHz低速RC振荡)
E_LXT (外部 32KHz低速石英振荡)
- 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)。
- 六种硬件中断:
 - Timer0 上溢中断。
 - Timer1 下溢中断。
 - WDT超时中断。
 - PB输入状态改变中断。
 - 外部中断。
 - 低电压侦测中断。
- AT8A513K在待机模式(Standby mode)下的六种唤醒中断:
 - Timer0 上溢中断。
 - Timer1 下溢中断。
 - WDT超时中断。
 - PB输入状态改变中断。
 - 外部中断。
 - 低电压侦测中断。
- AT8A513K在睡眠模式(Halt mode)下的三种唤醒中断:
 - WDT超时中断。
 - PB输入状态改变中断。
 - 外部中断。
- 复位脚可以输出高。
- PWM1/PWM2/PWM3 共用Timer 1, 可不经除级。

AT8A513H/AT8A513J/AT8A513K/AT8A513L/AT8A513G比较:

	AT8A513H	AT8A513J	AT8A513K	AT8A513L	AT8A513G
PWM	1	3	3	3	1
Crystal Cap	-	-	V	V	V
220mA drive current	-	-	-	V	V
Timer1 时钟源来自内部高频（16M, 20MHz）	-	V	V	V	-

1.2 方框图



1.3 引脚图

AT8A513K提供SOP8 封装类型:

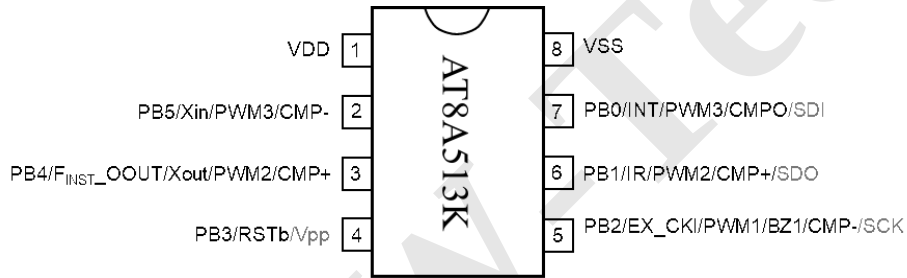


图 1 封装引脚图

1.4 引脚说明

引脚名	I/O	描述
PB0 / INT / SDI / PWM3 / CMPO	I/O	PB0 是一个双向I/O脚，同时它可作比较器输出脚。 当EIS=1 & INTIE=1 时，PB0 是外部中断的输入引脚。 PB0 可作编程脚SDI。 PB0 可作PWM3 输出脚。 PB0 可作比较器输出脚。
PB1 / IR / SDO / PWM2 / CMP+	I/O	PB1 是一个双向I/O脚，同时它可作比较器输入脚。 如果启用红外模式，该引脚为红外载波输出脚。 PB1 可作编程脚SDO。 PB1 可作PWM2 输出脚。 PB1 可作比较器正相输入脚。
PB2 / EX_CKl / BZ1 / SCK / PWM1 / CMP-	I/O	PB2 是一个双向I/O脚，同时它可作比较器输入脚。 PB2 可作定时器频率来源EX_CKl。 PB2 可作蜂鸣器输出脚。 PB2 可作编程脚SCK。 PB2 可作PWM1 输出脚。 PB2 可作比较器反相输入脚。
PB3 / RSTb / VPP	I/O	PB3 是一个双向 I/O 脚，同时它可作开漏极输出脚。 PB3 是复位脚 RSTb。 PB3 可作编程脚 VPP。
PB4 / FINST_OUT / Xout / PWM2 / CMP+	I/O	PB4 是一个双向I/O脚，同时它可作比较器输入脚。 PB4 可作指令时钟输出脚。 如果采用E_LXT，PB4 可作外部晶振输出脚。 PB4 可作PWM2 输出脚。 PB4 可作比较器正向输入脚。
PB5 / Xin / PWM3 / CMP-	I/O	PB5 是一个双向I/O脚，同时它可作比较器输入脚。 如果采用E_LXT，PB5 可作外部晶振输入脚。 PB5 可作PWM3 输出脚。 PB5 可作比较器反相输入脚。
VDD	-	电源正极。
VSS	-	电源负极。

2. 内存结构

AT8A513K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

AT8A513K程序存储器空间是 1K。因此，程序计数器（PC）是 10 位宽，用来处理程序存储器的任何位置。

程序存储器的某些位置保留为中断入口。上电复位向量位于 0x000，软件中断向量位于 0x001，内部和外部硬件中断向量位于 0x008。

AT8A513K提供CALL、GOTOA和CALLA等指令去处理程序空间的 256 个位置。还提供GOTO指令去处理程序空间 512 个位置、LCALL和LGOTO指令处理程序空间的任何位置。

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

AT8A513K程序ROM地址 0x3FE~0x3FF是预留空间。如果用户试图在这些地址写代码会得到意料之外的错误函数。

AT8A513K程序ROM地址 0x00E~0x00F是预设滚码地址，它可以被释放用于普通程序空间。

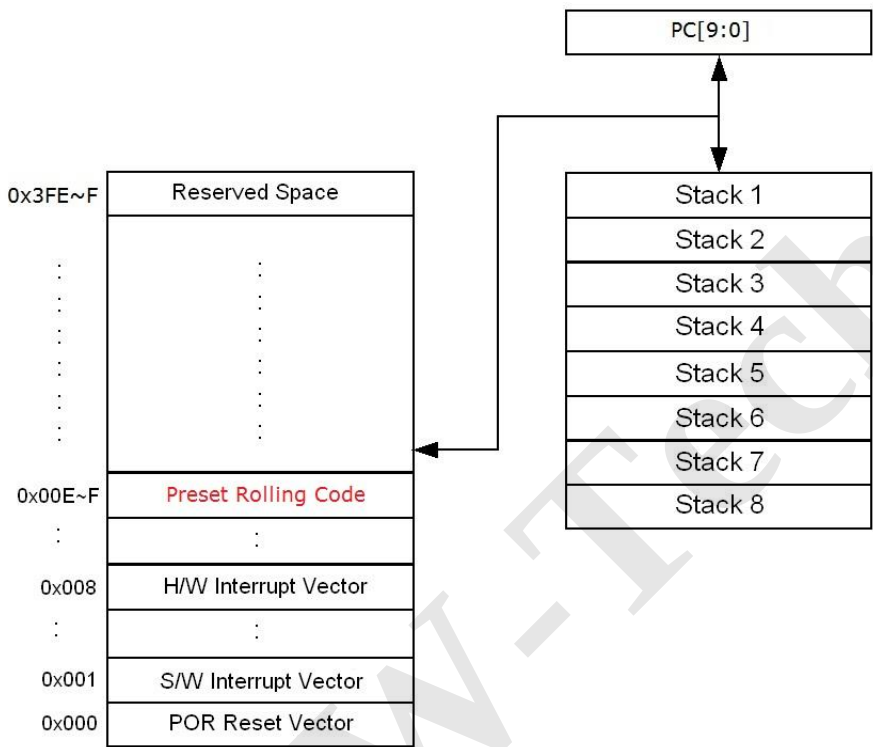


图 2 程序存储器对应地址

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**是由静态存储器组成，用户可以使用它们来存储变量或中间结果。

R-page数据存储器分为 4 组存储页面，可直接或间接通过**SFR**寄存器（亦为数据指针寄存器**FSR**）使用。**FSR[7:6]** 作为存储页面寄存器**BK[1:0]**从 4 个存储页面中选择一个。

R-page寄存器可分为直接寻址方式和间接寻址方式。

数据存储使用的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。存储页面选择是由**FSR[7:6]**决定，区位选择则是由**FSR[5:0]**而定。

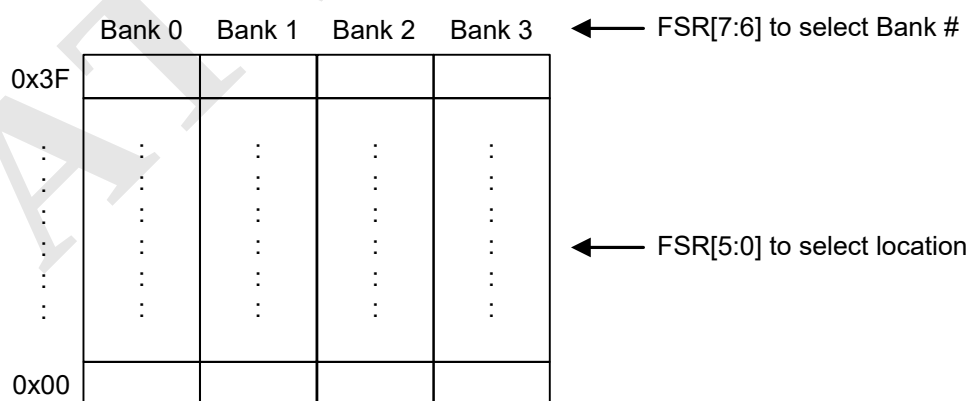


图 3 数据存储器存取的间接寻址方式

下面描述了数据存储使用的直接寻址方式。存储页面选择是由**FSR[7:6]**决定，而区位选择则是由**op-code[5:0]**指令直接决定。

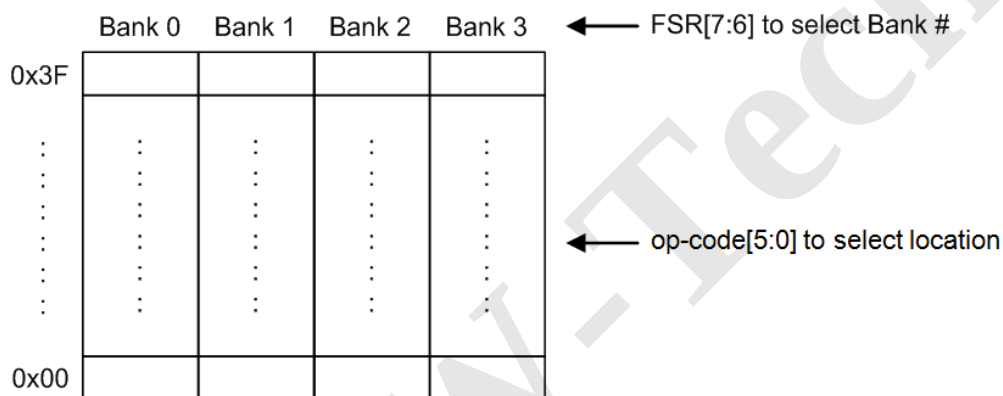


图 4 数据存储器存取的直接寻址方式

特殊功能寄存器**R-page**可以通过一般的指令使用，如算术指令和数据移动指令。**R-page SFR**占用了从 0x0 到 Bank 0 的 0xF。然而，Bank 1、Bank 2 和 Bank 3 的相同地址会映射回 Bank 0。换句话说，**R-page**完全存在于 Bank 0。**GPR**完全占用了存储页面的 0x10 到 0x3F，和其他存储页面 0x10 到 0x3F 映射后的表 1 所示。

AT8A513K寄存器名称和特殊功能寄存器**R-page**的映射地址说明如下表：

FSR[7:6] Address	00 (Bank 0)	01 (Bank 1)	10 (Bank 2)	11 (Bank 3)
0x0	INDF	映射至bank0		
0x1	TMR0			
0x2	PCL			
0x3	STATUS			
0x4	FSR			
0x5	-			
0x6	PORTB			
0x7	-			
0x8	PCON			
0x9	BWUCON			
0xA	PCHBUF			
0xB	BPLCON			
0xC	BPHCON			
0xD	-			
0xE	INTE			
0xF	INTF			
0x10 ~ 0x1F	通用寄存器	未使用		
0x20 ~ 0x3F	通用寄存器	未使用		

表 1 特殊功能寄存器R-page地址映射表

特殊功能寄存器F-page只能被指令IOST和IOSTR使用，特殊功能寄存器S-page只能被指令SFUN和SFUNR使用。当F-page和S-page寄存器被使用时，FSR[7:6]存储页面选择位会被忽略。

寄存器名称和F-page、S-page的映射地址说明如下表。

SFR 种类 地址	F-page SFR	S-page SFR
0x0	-	TMR1
0x1	-	T1CR1
0x2	-	T1CR2
0x3	-	PWM1DUTY
0x4	-	PS1CV

SFR 种类 地址	F-page SFR	S-page SFR
0x5	-	BZ1CR
0x6	IOSTB	IRCR
0x7	-	TBHP
0x8	-	TBHD
0x9	-	P2CR1
0xA	PS0CV	PWM2DUTY
0xB	-	P3CR1
0xC	BODCON	PWM3DUTY
0xD	-	-
0xE	CMPCR	-
0xF	PCON1	OSCCR

表 2 特殊功能寄存器F-page和S-page地址映射表

3. 功能概述

本章节将详细描述AT8A513K的操作方式。

3.1 R页面特殊功能寄存器

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。任何指令存取间接寻址寄存器时，实际上是读取数据指针寄存器FSR所指定的寄存器。

3.1.2 TMR0（定时器 T0 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，实际上是读取Timer0 当前运行的数值。

写入TMR0 寄存器会改变Timer0 当前的数值。

通过T0MD与配置字设定，Timer0 时钟源可以从指令时钟FINST、外部脚位EX_CKl或低振荡频率中择一。

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							

程序计数寄存器是最显著的10位PC字节（LSB）。在指令一一被执行后，PCL会增加，除了某些指令会直接改变PC。PC的高字节如PC[9:8]并不能直接存取。更新PC[9:8]必须通过PCHBUF寄存器完成。

对于GOTO指令，PC[8:0]是从指令值取得，而PC[9]是从PCHBUF[1]加载。对于CALL指令，PC[7:0]是从指令值取得，而PC[9:8]是从PCHBUF[1:0]加载。下一个PC地址，如PC+1，将会被推到堆栈的顶部。对于LGOTO指令，PC[9:0]是从指令值取得。

对于LCALL指令，PC[9:0]是从指令值取得；当下一个PC地址，如PC+1，将会被推到堆栈的顶部。

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	GP7	GP6	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指令的结果和导致复位的原因。

C: 进位/借位标志位

C=1 时，加法指令进位，减法指令不借位。

C=0 时，加法指令不进位，减法指令借位。

DC: 半进位/半借位标志位

DC=1 时，加法指令的第四个LSB进位，减法指令的第四个LSB不借位。

DC=0 时，加法指令的第四个LSB不进位，减法指令的第四个LSB借位。

Z: 零位

Z=1 时，逻辑运算的结果为零。

Z=0 时，逻辑运算的结果不为零。

/PD: 掉电标志位

/PD=1 时，上电或执行CLRWDWT指令后。

/PD=0 时，执行SLEEP指令后

/TO: 时间上溢标志位

/TO=1 时，上电或执行CLRWDWT或SLEEP指令后。

/TO=0 时，发生WDT超时。

GP7、GP6、GP5: 通用读写寄存器。

(*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	BK[1:0]		FSR[5:0]					
读/写属性			读/写							
初始值			0	0	X	X	X	X	X	X

FSR[5:0]: 从特定存储区的 64 个寄存器中选择一个。

3.1.6 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			数据锁存值是xxxxxx，读取值是xxxxxx，端口值(PB5~PB0)							

当读取PortB时，若特定脚位被配置为输入脚，寄存器将得到该脚位的状态。然而，若该脚位被配置为输出脚，不论是得到该脚位的状态或相对应的输出数据锁存值，寄存器都是依据配置字RD_OPT决定。当写入PortB时，数据被写入PB口的输出数据锁存器中。

GP7, GP6: 通用读/写寄存器。

3.1.7 PCON（电源控制寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	R	0x8	WDTEN	EIS	LVDEN	GP4	LVREN	CMPEN	GP1	GP0
读/写属性			读/写							
初始值			1	0	0	0	1	0	0	0

GP1~0: 通用读/写寄存器。

LVREN: 打开/关闭LVR。

LVREN=1 时，打开LVR。

LVREN=0 时，关闭LVR。

CMPEN: 打开/关闭CMP。

CMPEN=1 时，打开CMP。

CMPEN=0 时，关闭CMP。

EIS: 外部中断选择位。

EIS=1 时，PB0 是外部中断。

EIS=0 时，PB0 是GPIO。

LVDEN: 打开/关闭LVD。

LVDEN=1 时，打开LVD。

LVDEN=0 时，关闭LVD。

WDTEN: 打开/关闭WDT。

WDTEN=1 时，打开WDT。

WDTEN=0 时，关闭WDT。

3.1.8 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: 打开/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。

3.1.9 PCHBUF（程序计数器的高字节）

名称	SFR 类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCHBUF	R	0xA	-	XSPD_STP	-	-	-	GP2	PA1(1k)	PA0
读/写属性			-	写	-	-	-	读/写	写	
初始值			X	0	X	X	X	0	00	

PCHBUF[1:0]: 程序计数器PC的 bit9 ~ bit8。

GP2: 通用读/写寄存器。

XSPD_STP: 写 1 来停止晶体管 32.768K加速功能，只写。

3.1.10 BPLCON（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
BPLCON	R	0xB	/PLPB3	/PLPB2	/PLPB1	/PLPB0	-	-	-	-
读/写属性			读/写	读/写	读/写	读/写	-	-	-	-
初始值			1	1	1	1	X	X	X	X

/PLPBx: 关闭/打开PBx下拉电阻， $0 \leq x \leq 3$ 。

/PLPBx=1时，关闭PBx下拉电阻。

/PLPBx=0 时，打开PBx下拉电阻。

3.1.11 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

/PHPBx: 关闭/打开PBx上拉电阻， $0 \leq x \leq 5$ 。

/PHPBx=1 时，关闭PBx上拉电阻。

/PHPBx=0 时，打开PBx上拉电阻。

注意: 当 **PB4** 和 **PB5** 用作晶振脚时，其上拉电阻不能打开，否则可能导致振荡失败。

3.1.12 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	-	WDIE	-	LVDIE	T1IE	INTIE	PBIE	T0IE
读/写属性			-	读/写	-	读/写	读/写	读/写	读/写	读/写
初始值			X	0	X	0	0	0	0	0

T0IE: 定时器T0上溢（overflow）中断使能位。

T0IE=1时，打开定时器T0上溢中断。

T0IE=0时，关闭定时器T0上溢中断。

PBIE: PortB输入变化中断使能位。

PBIE=1时，打开PortB输入变化中断。

PBIE=0时，关闭PortB输入变化中断。

INTIE: 外部中断使能位。

INTIE=1时，打开外部中断。

INTIE=0时，关闭外部中断。

T1IE: 定时器T1下溢中断使能位。

T1IE=1时，打开定时器T1下溢中断。

T1IE=0时，关闭定时器T1下溢中断。

LVDIE: LVD中断使能位。

LVDIE=1时，打开LVD中断。

LVDIE=0时，关闭LVD中断。

WDTIE: WDT超时中断使能。

WDTIE=1时，打开WDT超时中断。

WDTIE=0时，关闭WDT超时中断。

3.1.13 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	-	WDTIF	-	LVDIF	T1IF	INTIF	PBIF	T0IF
读/写属性			-	读/写	-	读/写	读/写	读/写	读/写	读/写
初始值(注意*)			X	0	X	0	0	0	0	0

T0IF: 定时器T0上溢中断标志位。

T0IF=1时，发生定时器T0上溢中断。

T0IF必须被程序清零。

PBIF: PortB输入变化中断标志位。

PBIF=1时，发生PortB输入变化中断。

PBIF必须被程序清零。

INTIF: 外部中断标志位。

INTIF=1时，发生外部中断。

INTIF必须被程序清零。

T1IF: 定时器T1下溢中断标志位。

T1IF=1时，发生定时器T1下溢中断。

T1IF必须被程序清零。

LVDIF: LVD中断标志位。

LVDIF=1时，发生LVD中断。

LVDIF必须被程序清零。

WDTIF: WDT超时中断标志位。

WDTIF=1时，发生WDT超时中断。

WDTIF必须被程序清零。

注意：当对应的 **INTE** 位未使能，读取中断标志是 **0**。

3.2 T0MD寄存器

T0MD是一个可读/可写寄存器但它只能由指令T0MD / T0MDR存取。

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

PS0SEL[2:0]：选择预分频器P0 分配比。分配比是根据预分频器P0 分配给定时器T0 或WDT决定。当预分频器P0 被分配给WDT，分配比取决于选择哪种超时机制。

PS0SEL[2:0]	分配比		
	PS0WDT=0 (定时器T0)	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 3 预分频器P0 分配比

PS0WDT：预分频器P0分配。

PS0WDT=1时，预分频器P0被分配到WDT。

PS0WDT=0时，预分频器P0被分配到定时器T0。

注意：在使能看门狗或时间中断前，要设定 **PS0WDT** 和 **PS0SEL[2:0]**，否则复位或中断可能导致错误触发。

T0CE：定时器T0 外部时钟边沿选择。

T0CE=1时，当高到低转换发生在EX_CKI脚时，定时器T0加一。

T0CE=0时，当低到高转换发生在EX_CKI脚时，定时器T0加一。

注意：T0CE也应用在低频振荡频率作为定时器T0时钟源条件。

T0CS：定时器T0时钟源选择。

T0CS=1时，选择EX_CKI脚的外部时钟或低振荡频率。

T0CS=0时，选择指令时钟F_{INST}。

INTEDG：外部中断的边沿选择。

INTEDG=1，当上升沿发生在PB0脚时，INTIF将被设定。

INTEDG=0，当下降沿发生在PB0脚时，INTIF将被设定。

LCKTM0: T0CS=1时，定时器T0时钟源可以选择低频振荡器。

T0CS=0时，指令时钟F_{INST}被选作定时器T0时钟源。

T0CS=1, LCKTM0=0时，EX_CK1脚的外部时钟被选择当作定时器T0时钟源。

T0CS=1, LCKTM0=1时，低振荡频率输出代替EX_CK1脚作为定时器T0时钟源。

注意：有关定时器 T0 时钟源选择的详细说明，请参考定时器 T0 章节。

3.3 F页面特殊功能寄存器

3.3.1 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	GP7	GP6	IOPB5	IOPB4	IOPB3	IOPB2	IOPB1	IOPB0
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	1	1	1	1	1	1

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

IOPBx=1时，PBx是输入模式。

IOPBx=0时，PBx是输出模式。

GP7, GP6: 通用寄存器。

3.3.2 PS0CV (预分频器 P0 计数器值寄存器)

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

当读取PS0CV时，寄存器会得到预分频器P0计数器的当前数值。

3.3.3 BODCON (PortB 开漏极寄存器)

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: 打开/关闭PBx的开漏极， $0 \leq x \leq 5$ 。

ODPBx=1时，打开PBx的开漏极。

ODPBx=0时，关闭PBx的开漏极。

3.3.4 CMPCR（比较器电压选择控制寄存器）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCR	F	0xE	GP7	RBIAS_H	RBIAS_L	CMF_INV	PS1	PS0	NS1	NS0
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	1	1	0	0

NS[1:0]: 比较器反相输入选择。

NS[1:0]	反相输入
00	PB2
01	PB5
10	Bandgap (0.65V)
11	Vref

PS[1:0]: 比较器正相输入选择。

PS[1:0]	正相输入
00	PB1
01	PB4
10	Vref
11	-

CMF_INV: 比较器输出反相控制位。

CMF_INV = 1, 反相比较器输出。

CMF_INV = 0, 正相比较器输出。

RBIAS_L, RBIAS_H: 设置相应的电压参考电平。

3.3.5 PCON1（电源控制寄存器 1）

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON1	F	0xF	GIE	LVDOOUT	LVDS3	LVDS2	LVDS1	LVDS0	GP1	T0EN
读/写属性			读/写(1*)	读	读/写	读/写	读/写	读/写	读/写	读/写
初始值			0	X	1	1	1	1	0	1

T0EN: 打开/关闭定时器T0。

T0EN=1时, 打开定时器T0。

T0EN=0时, 关闭定时器T0。

GIE: 全局中断使能位。

GIE=1时, 打开所有未标记中断。

GIE=0时, 关闭所有的中断。

LVDOOUT: 低电压检测输出, 只读。

LVDS3~0: 从14种LVD电压中选择其中一个。

RBIAS[H:L]	LVDS[3:0]	LVD Voltage (V)
00	0001	2.20
00	0010	2.40
00	1011	2.60
00	0011	2.80
00	0100	2.90
00	0101	3.00
00	1101	3.15
00	0110	3.30
00	1110	3.45
00	0111	3.60
00	1111	3.75
00	1000	3.90
00	1010	4.05
00	1100	4.15

表 4 LVD 电压选择

VDD电压由高变低与VDD电压由低变高之间的滞后电压约为±0.1V。

GP1：通用读/写寄存器。

(1*)：由指令 ENI 所设定、指令 DISI 清除、指令 IOSTR 读取。

3.4 S页面特殊功能寄存器

3.4.1 TMR1（定时器 T1 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，会得到目前8-bit递减计数定时器T1的目前数值。当写入TMR1寄存器时，数据将写入定时器T1寄存器并更新定时器T1目前内容。

3.4.2 T1CR1（定时器 T1 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T1 功能。

T1EN: 打开/关闭定时器T1。

T1EN=1时，打开定时器T1。

T1EN=0时，关闭定时器T1。

T1RL: 当连续模式被选择（T1OS=0），配置定时器T1递减计数机制。

T1RL=1时，初始值从TMR1寄存器被重新加载。

T1RL=0时，当发生下溢时会继续从0xFF递减计数。

T1OS: 当发生下溢时，配置定时器T1 操作模式。

T1OS=1时，单次模式。定时器T1会从初始值到0x00计数一次。

T1OS=0时，连续模式。下溢后，定时器T1会持续递减计数。

TM1_HRC: 打开/关闭内部高频时钟源定时器T1。

TM1_HRC=1 时，定时器T1 时钟源来自内部高频。

TM1_HRC=0 时，定时器T1 时钟源来自CPU运行时钟。

T1OS	T1RL	定时器T1 递减计数功能
0	0	定时器T1从重载的数值倒数到0x00。 当发生下溢时，0xFF被重载并继续递减计数。
0	1	定时器T1从重载的数值倒数到0x00。 当发生下溢时，再次重载数值并继续递减计数。
1	x	定时器T1从初始值倒数到0x00。 当发生下溢时，定时器T1 停止计数。

表 5 定时器T1 功能

PWM1OAL: 定义PWM1 输出有效状态。

PWM1OAL=1时，PWM1输出为低电平有效。

PWM1OAL=0时，PWM1输出为高电平有效。

PWM1OEN: 打开/关闭PWM1输出。

PWM1OEN=1时，PB2输出PWM1。

PWM1OEN=0时，PB2是通用I/O脚。

3.4.3 T1CR2（定时器 T1 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

该寄存器用于配置定时器T1功能。

PS1SEL[2:0]: 预分频器PS1 分配比选项。

PS1SEL[2:0]	分配比
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 6 预分频器PS1 分配比

注意：须设定 **PS1SEL[2:0]**在**PS1EN=1**，否则中断可能会被误触发。

/PS1EN：关闭/打开预分频器PS1。

/PS1EN=1时，关闭预分频器PS1。

/PS1EN=0时，开启预分频器PS1。

T1CE：定时器T1外部时钟边沿选项。

T1CE=1时，当高到低转换发生在EX_CKI脚时，定时器T1会减一。

T1CE=0时，当低到高转换发生在EX_CKI脚时，定时器T1会减一。

T1CS：定时器T1时钟源选项。

T1CS=1时，选择在EX_CKI脚的外部时钟。

T1CS=0时，选择指令时钟。

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							

此寄存器仅能写入。当定时器T1被开启并开始计时后，PWM1输出会保持在非有效状态。当定时器T1数值等同PWM1DUTY时，PWM1输出会变为有效状态直到发生下溢。

定时器T1重加载的数值储存在TMR1寄存器以用来定义PWM1帧率，PWM1DUTY寄存器用于定义PWM1的占空比。

3.4.5 PS1CV（预分频器 PS1 计数值寄存器）

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PS1 计数器的目前数值。

3.4.6 BZ1CR（蜂鸣器 BZ1 计数寄存器）

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

BZ1FSEL[3:0]: BZ1 输出的频率选项。

BZ1FSEL[3:0]	BZ1 频率选项	
	时钟源	分配比
0000	预分频器PS1 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T1 输出	定时器T1 bit 0
1001		定时器T1 bit 1
1010		定时器T1 bit 2
1011		定时器T1 bit 3
1100		定时器T1 bit 4
1101		定时器T1 bit 5
1110		定时器T1 bit 6
1111		定时器T1 bit 7

表 7 蜂鸣器BZ1 输出（PB2）频率选项

BZ1EN: 打开/关闭BZ1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: 打开/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: 选择IR载波频率。

IRF57K=1时, IR载波频率是57KHz。

IRF57K=0时, IR载波频率是38KHz。

IRCSEL: 选择IR载波极性。

IRCSEL=0, 且I/O脚数据是1时, 产生IR载波。

IRCSEL=1, 且I/O脚数据是0时, 产生IR载波。

IROSC358M: 当使用外部晶振时, 该位由使用的晶振各类决定。如果使用内部高频振荡, 该位会被忽略。

IROSC358M=1时, 晶振频率为3.58MHz。

IROSC358M=0时, 晶振频率为455KHz。

注意:

1. 仅有高频振荡 (F_{Hosc}) (详见章节3.11) 可以当作 IR 时钟源。
2. 不同振荡类型的分配比。

OSC. Type	57KHz	38KHz	条件
High IRC	64	96	HIRC模式 (不论系统频率是多少, IR模块的输入都设定为4MHz)。

表 8 不同振荡类型的分配比

3.4.8 TBHP (表格存取高字节地址指针寄存器)

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TBHP	S	0x7	-	-	-	-	-	GP2	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, 目标地址是由TBHP[2:0]与ACC组成。ACC是PC[9:0]的低字节, TBHP[1:0]是PC[9:0]的高字节。GP2 是AT8A513K的通用寄存器。

3.4.9 TBHD (表格存取高字节数据寄存器)

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 P2CR1 (PWM2 控制寄存器 1)

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2CR1	S	0x9	PWM2OEN	PWM2OAL	-	-	-	-	-	-
读/写属性			读/写	读/写	-	-	-	-	-	-
初始值			0	0	X	X	X	X	X	X

PWM2OAL: 定义PWM2输出有效状态。

PWM2OAL=1时, PWM2输出为低电平有效。

PWM2OAL=0时, PWM2输出为高电平有效。

PWM2OEN: 开启/关闭PWM2输出。

PWM2OEN=1时, PB1或PB4输出PWM2 (Option)。

PWM2OEN=0时, PB1或PB4是GPIO。

3.4.11 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	0xA	PWM2DUTY[7:0]							
读/写属性			写							
初始值			XXXXXXXX							

此寄存器仅能写入。当定时器T1 被开启并开始计时后, PWM2 输出会保持在非有效状态。当定时器T1 数值等同PWM2DUTY时, PWM2 输出会变为有效状态直到发生下溢。

3.4.12 P3CR1 (PWM3 控制寄存器 1)

名称	SFR类型	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3CR1	S	0xB	PWM3OEN	PWM3OAL	-	-	-	-	-	-
读/写属性			读/写	读/写	-	-	-	-	-	-
初始值			0	0	X	X	X	X	X	X

PWM3OAL: 定义PWM3输出有效状态。

PWM3OAL=1时, PWM3输出为低电平有效。

PWM3OAL=0时, PWM3输出为高电平有效。

PWM3OEN: 开启/关闭PWM3输出。

PWM3OEN=1时, PB0或PB5输出PWM3 (Option)。

PWM3OEN=0时, PB0或PB5是GPIO。

3.4.13 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	0xC	PWM3DUTY[7:0]							
读/写属性			写							
初始值			XXXXXXXX							

此寄存器仅能写入。当定时器T1 被开启并开始计时后，PWM3 输出会保持在非有效状态。当定时器T1 数值等同PWM3DUTY时，PWM3 输出会变为有效状态直到发生下溢。

3.4.14 OSCCR (振荡控制寄存器)

名称	SFR类型	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCCR	S	0xF	-	CMP OPEN	-	-	OPMD[1:0]	STPHOSC	SELHOSC	
读/写属性			-	读/写	-	-	读/写	读/写	读/写	
初始值			X	0	X	X	00	0	1	

SELHOSC: 系统振荡选项 (F_{osc})

SELHOSC=1时，F_{osc}是高频率振荡 (F_{HOSC})

SELHOSC=0时，F_{osc}是低频率振荡 (F_{LOSC})

STPHOSC: 关闭/打开高频率振荡 (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	一般模式
01	睡眠模式
10	待机模式
11	保留

表 9 选择OPMD[1:0]的操作模式

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

CMPOEN: 打开/关闭比较器输出到PB0。

CMPOEN=1时，打开比较器输出到PB0。

CMPOEN=0 时，关闭比较器输出到PB0。

3.5 I/O端口

AT8A513K提供 6 个I/O脚位（PB[5:0]）。用户可以通过寄存器PortB读/写这些脚位。每个I/O脚位都有一个对应的寄存器位以定义该脚位是输入或输出脚。寄存器IOSTB[5:0]定义PB[5:0]输入/输出方向。

当一个I/O脚位被配置为输入脚，它可以通过寄存器开启或关闭上拉或下拉电阻。寄存器BPHCON[5:0]用于开启或关闭PB[5:0]的上拉电阻。寄存器BPLCON[7:4]则是用于开启或关闭PB[3:0]的下拉电阻。

当一个I/O脚位被配置为输出脚，相对应的个别寄存器会选择开漏极输出脚。寄存器BODCON[5:0]决定PB[5:0]是否为开漏极。（除了PB[3]，当它配置为输出脚时，会一直保持开漏极模式。）

I/O功能摘要如下表：

功能		PB[2:0]	PB[3]	PB[5:4]
输入	上拉电阻	V	V	V
	下拉电阻	V	V	X
输出	开漏极	V	V	V

表 10 I/O功能摘要

PB的每个I/O脚位的电平变化可能会产生中断。寄存器BWUCON[5:0]会在选择PB任一I/O脚位时产生中断。只要选择到对应到BWUCON的任一PB脚位，且电平变化发生在任一选择的脚位时，寄存器 PBIF (INTF[1])就会设定为 1。如果寄存器 PBIE (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。

AT8A513K仅提供一个外部中断，当寄存器 EIS (PCON[6])设定为 1，PB0 则被当作外部中断的输入脚。

注意：当 PB0 同时设定成电平变化触发与外部中断，外部中断有较高的优先，而 PB0 电平变化触发则会被关闭，但 PB5~PB1 电平变化功能不会被影响。

AT8A513K提供IR载波生成器。IR载波生成器是由寄存器 IREN (IRCR[0])使能，脚位PB1 会输出载波。

PB3 可当作外部复位输入，由配置字决定。当低电平信号应用于PB3 时将会导致AT8A513K进入复位程序。

当AT8A513K处于一般模式或待机模式并打开配置字，用户可以在PB4 观察指令时钟。

如果T0MD T0CS=1 和LCK_TM0=0，PB2 可以当作定时器T0 外部时钟源EX_CK1。如果T1CS=1，PB2 可以当作定时器T1 外部时钟源。

如果T1CR1[7] PWM1OEN=1 并且对应到PB2 配置字，PB2 可以当作PWM1 输出。若BZ1CR[7] BZ1EN=1 并对应到PB2 配置字，PB2 也可以当作蜂鸣器输出。

3.5.1 I/O 引脚结构框图

IO_SEL: 设定引脚的属性为输入或输出。

WRITE_EN: 将资料写入引脚。

READ_EN: 读取引脚。

OD_EN: 打开开漏极。

PULLUP_ENB: 打开Pull-High。

PULLDOWN_EN: 打开Pull-Low。

RD_TYPE: 选择读取脚位或门锁。

PBEN: 打开PWM3/CMPO功能。

PBDT: PWM3/CMPO数据。

EIS: 打开外部中断功能。

INTEDGE: 外部中断边沿选择。

EX_INT: 外部中断信号。

WUB: 打开port B唤醒功能。

SET_PBIF: port B唤醒标志。

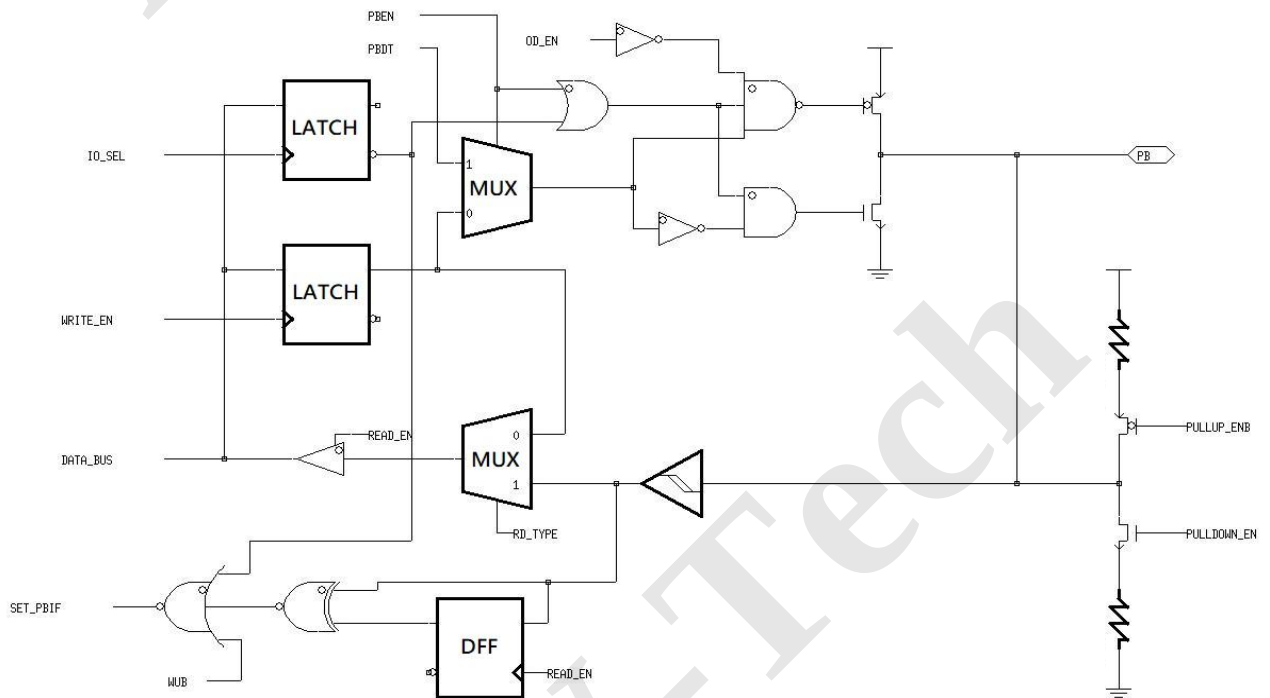


图 5 PB0 结构框图

IO_SEL: 设定引脚的属性为输入或输出。

WRITE_EN: 将资料写入引脚。

READ_EN: 读取引脚。

OD_EN: 打开开漏极。

PULLUP_ENB: 打开Pull-High。

PULLDOWN_EN: 打开Pull-Low。

RD_TYPE: 选择读取脚位或门锁。

PBEN: 打开PWM2/IR功能。

PBDT: PWM2/IR数据。

VPEN: 打开比较器正相输入功能。

CMPVP: 比较器正相输入。

WUB: 打开port B唤醒功能。

SET_PBIF: port B唤醒标志。

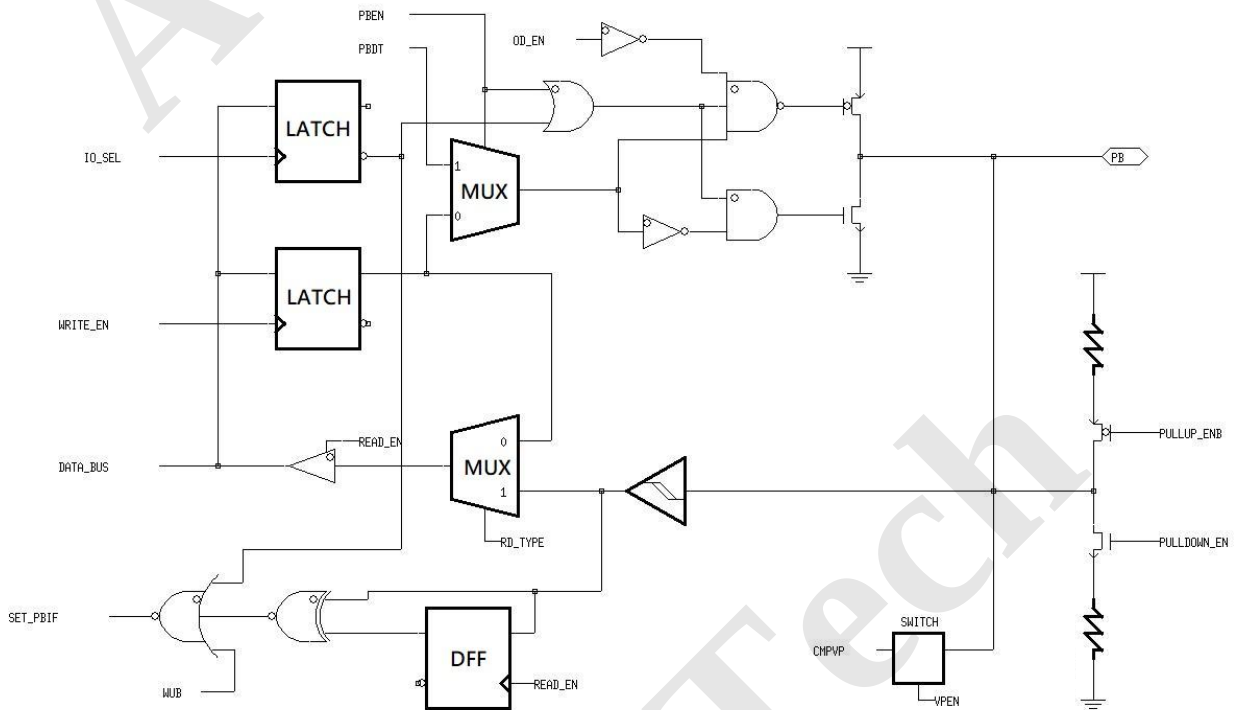


图 6 PB1 结构框图

EX CKI: 外部时钟输入。

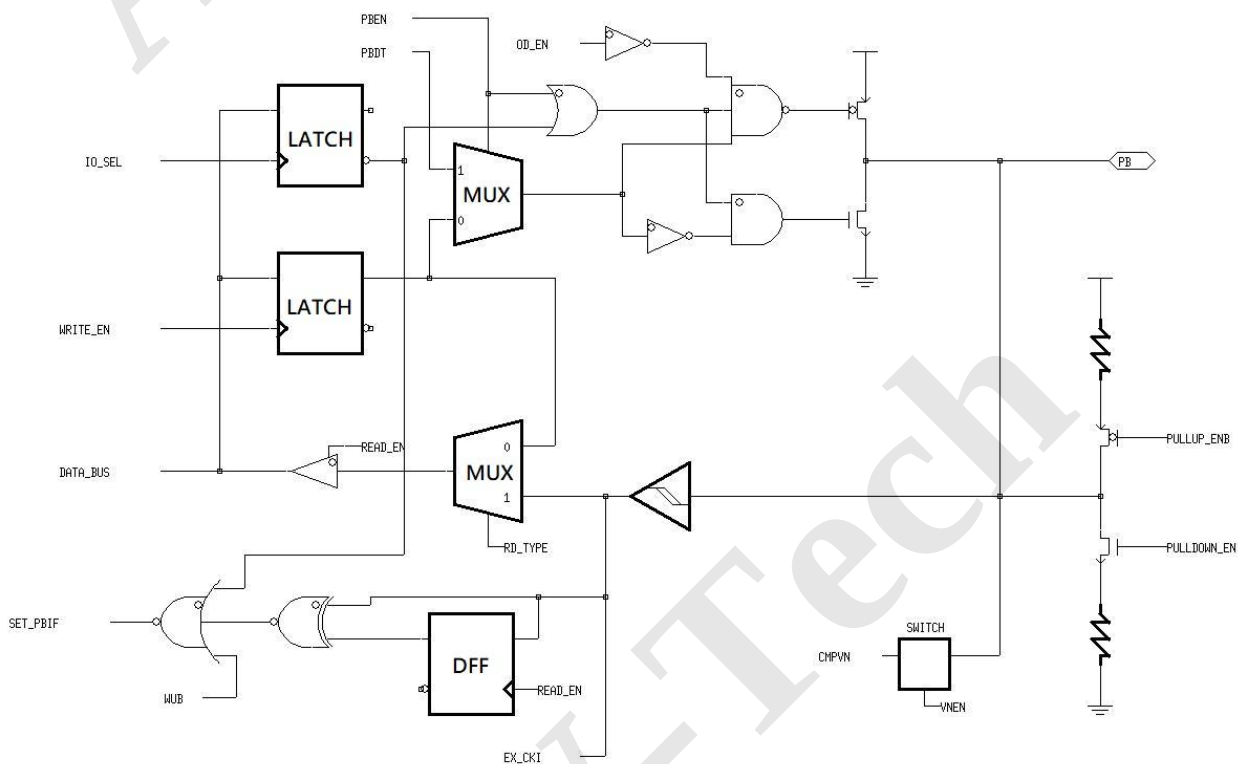


图 7 PB2 结构框图

IO_SEL: 设定引脚的属性为输入或输出。

WRITE_EN: 将资料写入引脚。

READ_EN: 读取引脚。

RSTPAD_EN: 打开复位脚。

RSTB_IN: 复位脚输入。

PULLDOWN_EN: 打开Pull-Low。

RD_TYPE: 选择读取脚位或门锁。

WUB: 打开port B唤醒功能。

SET_PBIF: port B唤醒标志。

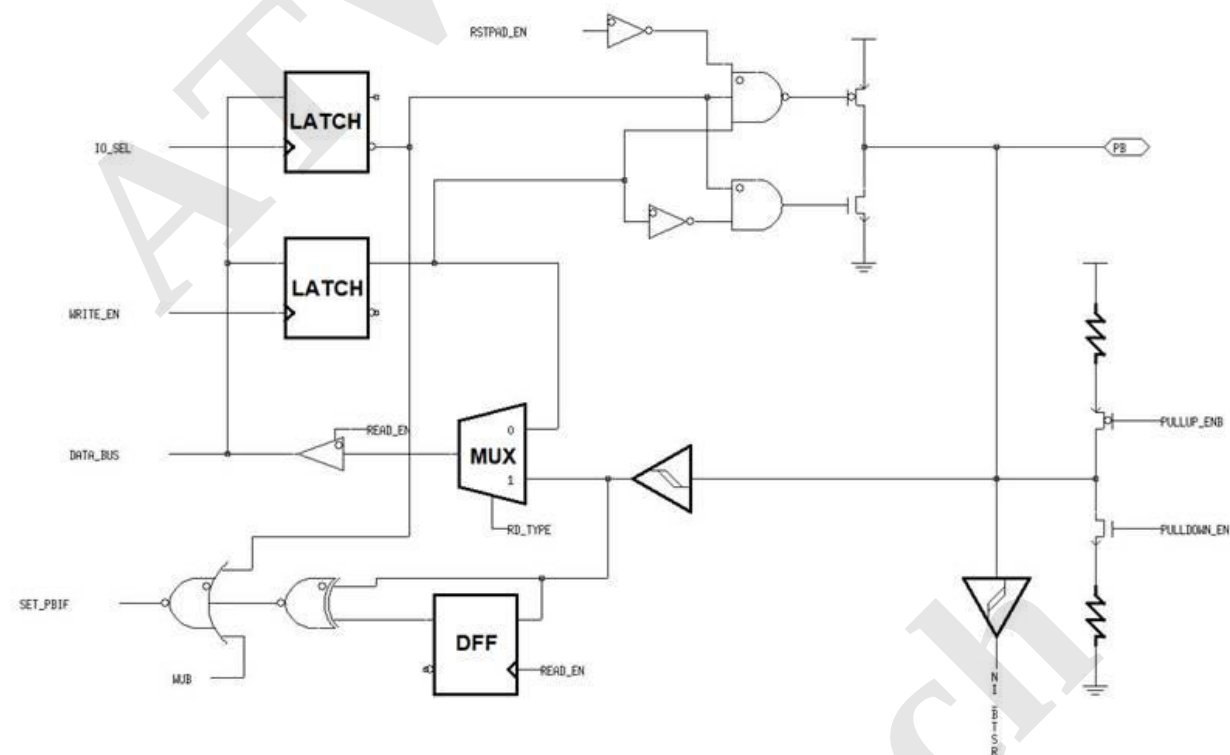


图 8 PB3 结构框图

注意：当 PB3 设置为输入脚时，输入电压不能大于 VDD。

SET_PBIF: port B唤醒标志。

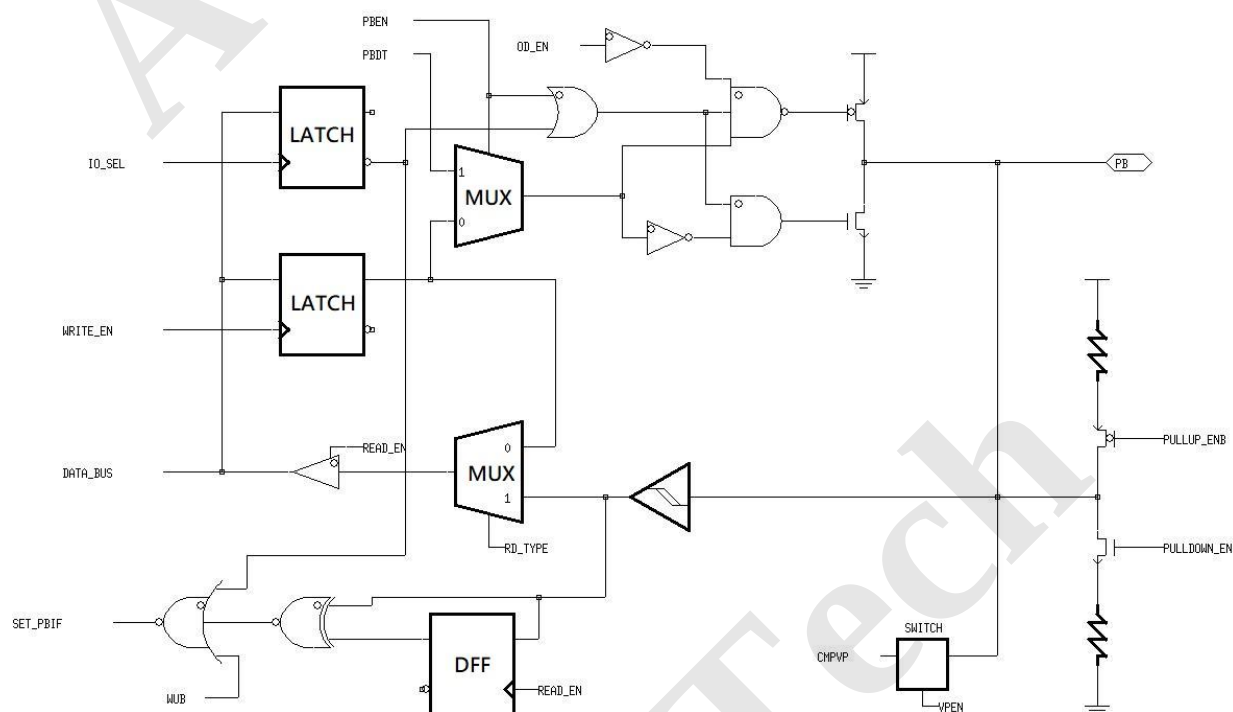


图 9 PB4 结构框图

SET_PBIF: port B唤醒标志。

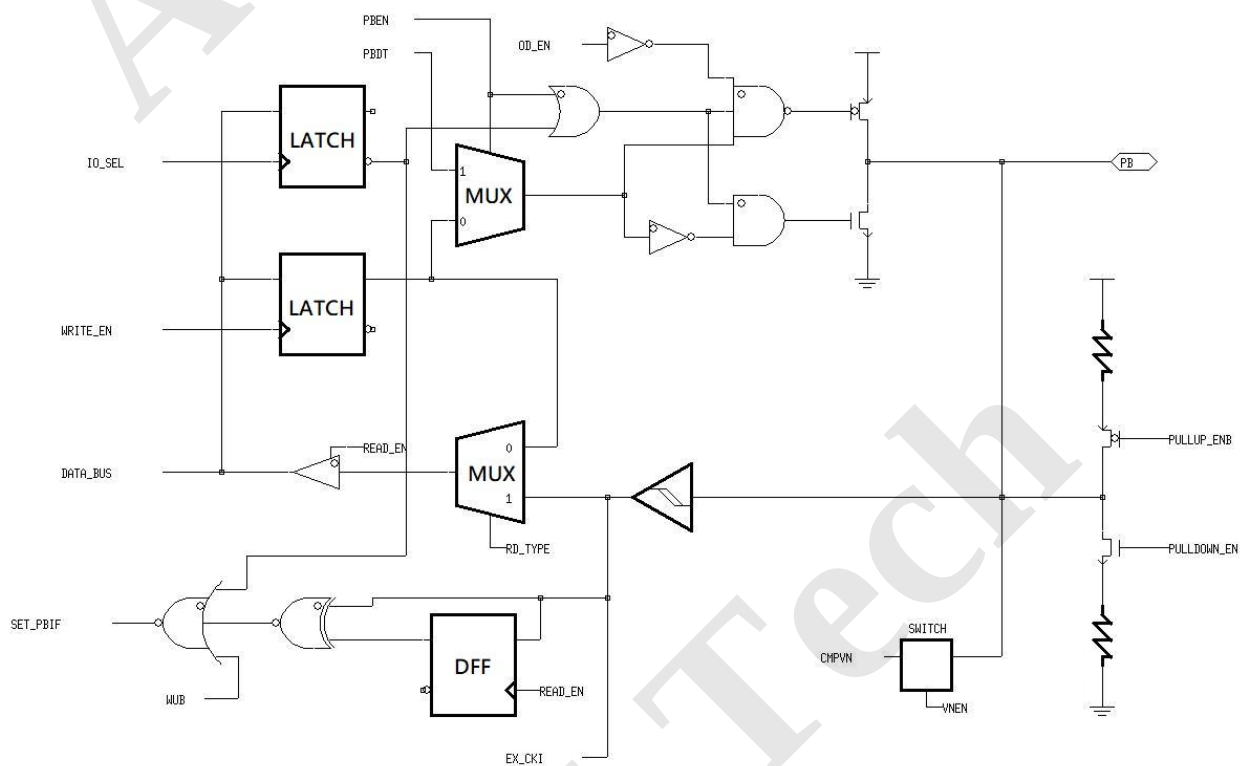


图 10 PB5 结构框图

3.6 定时器T0

定时器T0 是 8 位递增定时器，通过寄存器T0EN (PCON1[0]) 打开操作。写入定时器T0 将会设定其初始值；读取定时器T0 时则会显示目前的计数值。

定时器T0 的时钟源根据寄存器T0CS、LCK_TM0 (T0MD[5]与T0MD[7]) 可以从指令时钟、外部脚位EX_CKI或低速时钟低振荡器频率中选择。当T0CS为 0，指令时钟会被选择当作定时器T0 时钟源。当T0CS是 1 且LCK_TM0 为 0，EX_CKI 会被当作定时器T0 时钟源。当T0CS是 1 且LCK_TM0 为 1 (定时器T0 来源必须设定为 1)，会选择低振荡器频率 (I_LRC)输出。汇总成表格如下。(也请参考图 10)

定时器 T0 时钟源	T0CS	LCKTM0	Timer0 source	低振荡频率
Instruction clock	0	X	X	X
EX_CKI	1	0	X	X
		X	0	
E_LXT	1	1	1	1
I_LRC	1	1	1	0

表 11 定时器T0 时钟源控制概述

激活EX_CKI的边沿或低振荡器频率，寄存器T0CE (T0MD[4]) 可以使定时器T0 增加。当T0CE是 1，EX_CKI或低振荡器频率上的高到低转换将使定时器T0 增加。当T0CE是 0，EX_CKI或低振荡器频率上的低到高转换将使定时器T0 增加。

在定时器T0 时钟源被使用前，如果寄存器PS0WDT (T0MD[3]) 被清零，定时器T0 时钟源可以由预分频器P0 分频。当通过指令在PS0WDT写入 0，预分频器 P0 会被指定到定时器T0，且会在执行指令后被清除。预分频器P0 的分频比是由寄存器PS0SEL[2:0] 从 1:2 到 1:256 中确定。

进入定时器T0 之前，定时器T0 时钟源默认与指令时钟同步。如果EX_CKI或低振荡器频率被用来当作定时器T0 时钟源，用户必须注意他们的频率不能超过指令时钟，否则会导致错误计数。当低振荡器频率同时被当作定时器T0 时钟源与指令时钟，用户必须指定AT8A513K预分频器P0 到定时器T0，预分频器P0 的分频比不得少于 4。

当定时器T0 上溢时，寄存器T0IF (INTF[0])将会设定为 1 以表明定时器T0 发生上溢。如果寄存器T0IE (INTE[0]) 与GIE都设定为 1，会发生中断请求，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T0 的结构框图与WDT如下图：

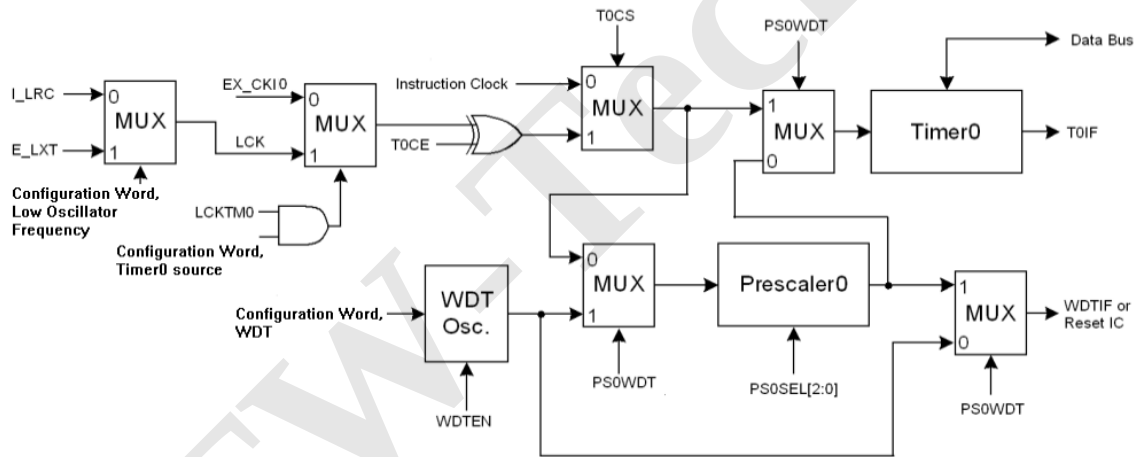


图 11 定时器T0 与WDT的结构框图

3.7 Timer1/PWM1/Buzzer1

定时器T1 是具有预分频器P1 的 8 位递减定时器，其分频率是可编程的。定时器T1 的输出可以被用于产生 PWM1/PWM2/PWM3 输出（可直接处理，不需要预分频器）与Buzzer1 输出。写入定时器T1 时也会写入定时器T1 重载寄存器(T1rd)与定时器T1 计数器。读取定时器T1 会显示定时器T1 目前技术值的内容。TM1_HRC 可选择 16MHz或 20MHz时钟源。

定时器T1 的结构框图如下图所示：

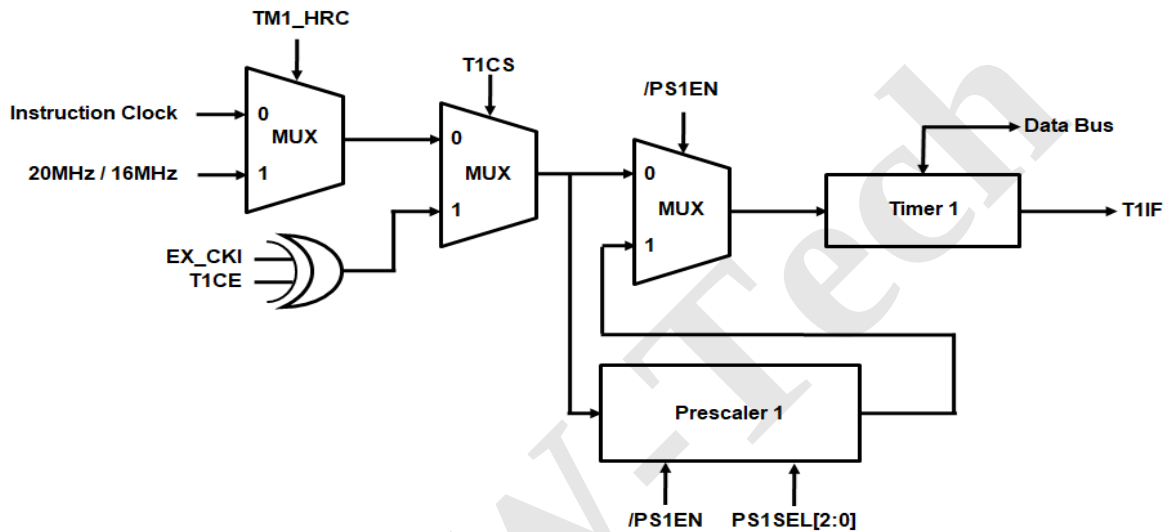


图 12 定时器T1 结构框图

定时器T1 的操作方式可以通过寄存器T1EN (T1CR1[0])开启或关闭。定时器T1 被打开后，它的时钟源是由寄存器T1CS (T1CR2[5])来决定是指令时钟或EX_CK1脚位。当T1CS为 1，EX_CK1会被选作时钟源。当T1CS为 0，指令时钟被选作时钟源。当EX_CK1被选取，寄存器T1CE (T1CR2[4])决定触发边沿去减少定时器T1。当 T1CE 为 1，EX_CK1的高到低转换将会使定时器T1 减少。当T1CE为 0，EX_CK1的低到高转换将会使定时器T1 减少。

被选择的时钟源应用到定时器T1 之前，它可以被预分频器P1 进一步分频。预分频器P1 可通过写 0 到寄存器/PS1EN (T1CR2[3]来打开，并且其分频比从 1:2 到 1:256 是由寄存器PS1SEL[2:0] (T1CR2[2:0])来决定。预分频器P1 目前数值可以通过读取寄存器PS1CV取得。

定时器T1 提供两种操作模式：单触发模式与连续模式。当寄存器T1OS (T1CR1[2]) 为 1，即选择单触发。定时器T1 会从储存在寄存器TMR1 的初始值计数递减到 0x00，例如下溢发生时。当寄存器T1OS (T1CR1[2])为 0，表示选择连续模式。当下溢发生，寄存器T1RL (T1CR1[1])会决定下一个递减计数的开始。当T1RL为 1，储存在寄存器TMR1 的初始值会被重新储存并从其初始值开始下一个递减计数。当T1RL为 0，定时器T1 从 0xFF开始下一个递减计数。

当定时器T1 下溢，寄存器T1IF (INTF[3])会被设定为 1 并表明定时器T1 发生下溢事件。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 0 至T1IF，T1IF才会被清除。

定时器T1 时序图如下图所示：

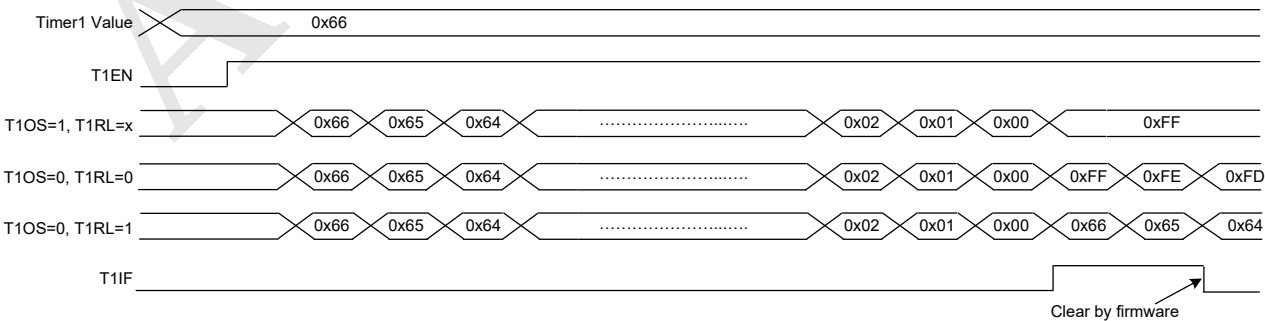


图 13 定时器T1 时序图

当寄存器PWM1OEN (T1CR1[7])被设定为 1 且配置字PB2 是PWM时，PWM1 由PB2 输出。当 PWM1OEN=1，PB2 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 输出低有效；PWM1OAL为0，PWM1 输出高有效。PWM1 的占空比与帧率都是可编程的。占空比是由寄存器PWM1DUTY决定。当PWM1DUTY为 0，PWM1 输出无效。当PWM1DUTY为 0xFF，PWM1 将会输出 255 个定时器T1 输入时钟。帧率是由TMR1 初始值所决定。因此，PWM1DUTY数值必须小于或等于TMR1。PWM1 的结构框图如下：

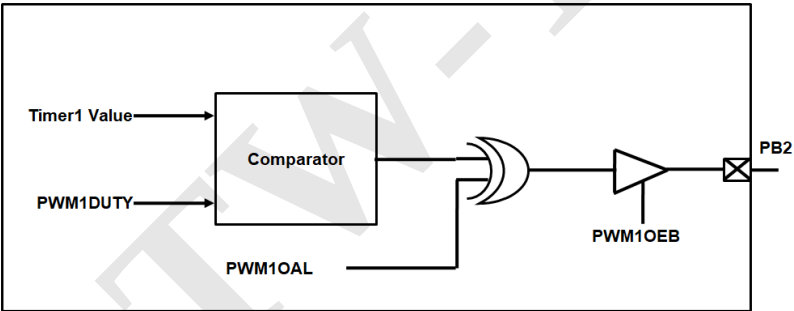


图 14 PWM1 结构框图

当寄存器BZ1EN (BZ1CR1[7]) 设定成 1 且对应的配置字PB2 是蜂鸣器时, Buzzer1 由PB2 输出。当BZ1EN设定为 1, PB2 将会自动成为输出脚。BZ1 的频率来源于定时器T1 输出或预分频器P1 输出, 且分频比是由寄存器BZ1FSEL[3:0] (BZ1CR[3:0])决定。当BZ1FSEL[3]为 0, 预分频器P1 输出被选为产生BZ1 输出。当BZ1FSEL[3]为 1, 定时器T1 输出被选为产生BZ1 输出。为了产生各种各样的频率, 分频比的范围可以从 1:2 到 1:256。Buzzer1 结构框图如下所示:

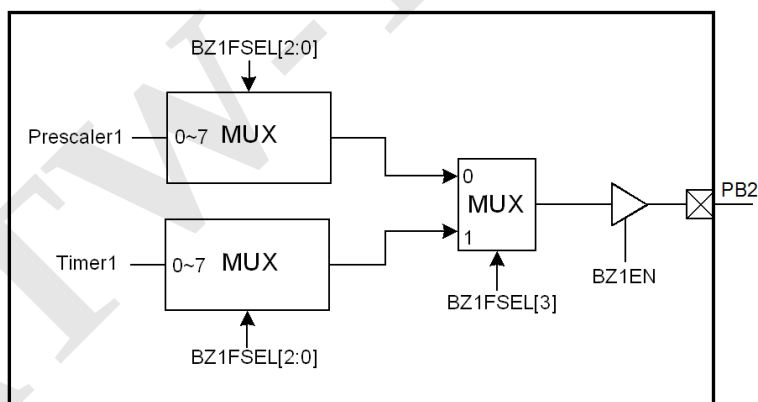


图 15 Buzzer1 结构框图

3.8 PWM2

当寄存器PWM2OEN (P2CR1[7])被设定为 1 且配置字PB1 或PB4 是PWM时, PWM2 由 PB1 或 PB4 输出。当 PWM2OEN=1, PB1 或 PB4 会自动成为输出脚。PWM2 输出的有效状态是由寄存器PWM2OAL (P2CR1[6])决定。当PWM2OAL为 1, PWM2 输出低有效; PWM2OAL为 0, PWM2 输出高有效。PWM2 的占空比与帧率都是可编程的。占空比是由寄存器PWM2DUTY决定。当PWM2DUTY为 0, PWM2 输出无效。当PWM2DUTY为 0xFF, PWM2 将会输出 255 个定时器T1 输入时钟。帧率是由TMR1 初始值所决定。因此, PWM2DUTY数值必须小于或等于TMR1。PWM2 的结构框图如下:

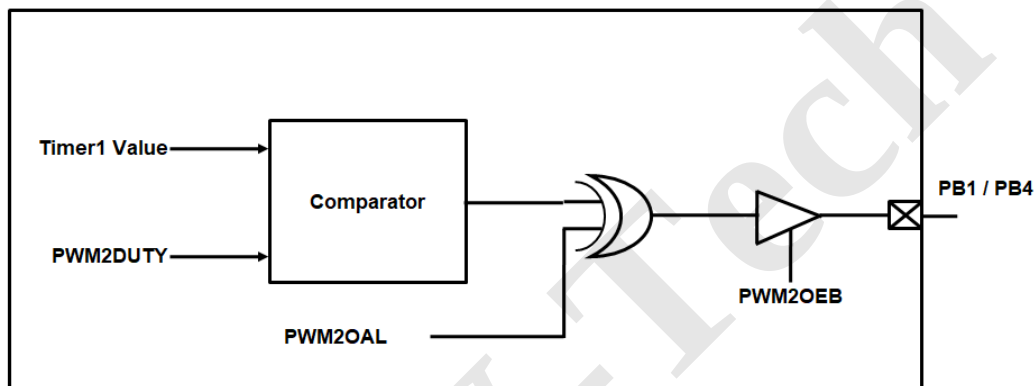


图 16 PWM2 结构框图

3.9 PWM3

当寄存器PWM3OEN (P3CR1[7])被设定为 1 且配置字PB0 或PB5 是PWM时, PWM3 由PB0 或PB5 输出。当 PWM3OEN=1, PB0 或 PB5 会自动成为输出脚。PWM3 输出的有效状态是由寄存器PWM3OAL (P3CR1[6])决定。当PWM3OAL为 1, PWM3 输出低有效; PWM3OAL为 0, PWM3 输出高有效。PWM3 的占空比与帧率都是可编程的。占空比是由寄存器PWM3DUTY决定。当PWM3DUTY为 0, PWM3 输出无效。当PWM3DUTY为

0xFF, PWM3 将会输出 255 个定时器T1 输入时钟。帧率是由TMR1 初始值所决定。因此, PWM3DUTY数值必须小于或等于TMR1。PWM3 的结构框图如下:

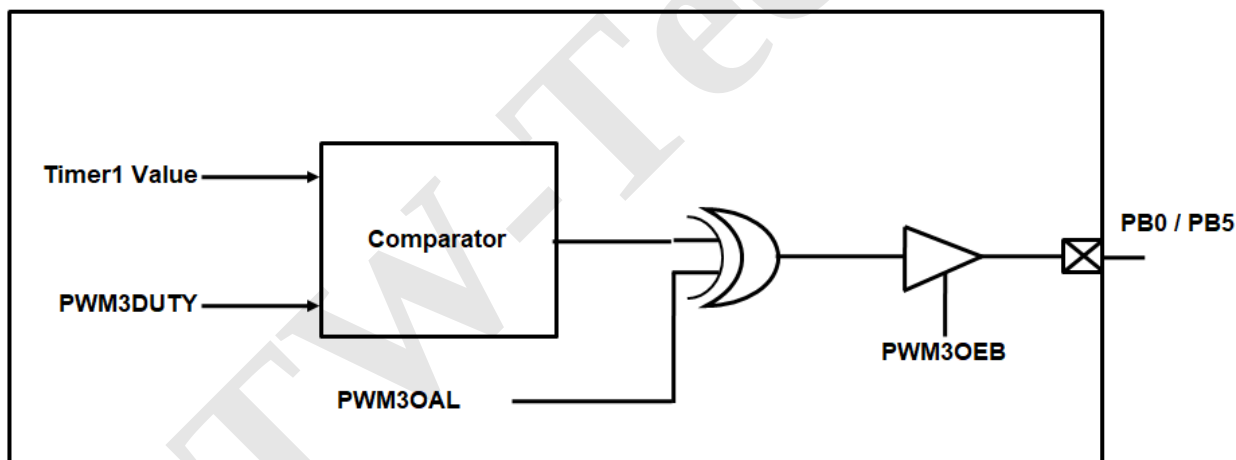


图 17 PWM3 结构框图

3.10 红外线载波

寄存器IREN (IRCR[0])被设定为 1 后, 会产生红外线载波, 而PB1 会自动成为输出脚。当IREN清零, PB1 将会如它被配置的一样, 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。

红外线载波的有效状态(极性)是根据PB1 输出数据所选择。当寄存器IRCSEL (IRCR[2])为 1 且PB1 输出数据为 0 时, PB1 将输出红外线载波。当寄存器IRCSEL (IRCR[2])为 0 且PB1 输出数据为 1, PB1 将输出红外线载波。红外线载波的极性如下图所示:

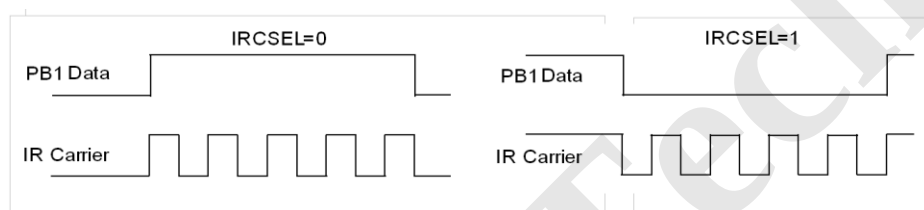


图 18 红外线载波的极性vs.输出数据

3.11 看门狗定时器 (WDT)

AT8A513K中有一个芯片内的, 自由运行的振荡器被WDT所使用。由于该振荡器独立于其它振荡电路, 故在待机模式和睡眠模式中WDT仍能继续工作。

WDT能被配置字打开或关闭。当WDT被配置字打开时, 在程序运行过程中, 其操作仍然可以通过寄存器WDTEN (PCON[7])来控制。此外, WDT超时后的机制可以复位AT8A513K或发出由另一个配置字决定的中断请求。同时, 在WDT超时后, 寄存器/TO (STATUS[4])将被清除为 0。

WDT超时周期的基线由两个配置字决定，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，则可以延长超时周期。通过将 1 写入寄存器PS0WDT，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]决定，并取决于WDT超时机制。分频比从 1:1 到 1:128，如果WDT超时，AT8A513K将会被复位。分频比从 1:2 到 1:256，如果WDT超时，AT8A513K将会被中断。

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 并且设置/TO标志位为 1。

如果用户选择中断WDT超时机制，在WDT失效后，寄存器WDTIF (INTF [6])将设置为 1。如果寄存器WDTIE (INTE [6])和GIE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。

3.12 中断

AT8A513K提供两种中断：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引起。硬件中断则有以下 6 种：

- Timer0 上溢中断。
- Timer1 下溢中断。
- WDT中断。
- PB输入状态改变中断。
- 外部中断。
- LVD中断。

GIE是全局中断打开标志，它必须为 1 才能打开硬件中断功能。GIE可以通过ENI指令设置，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 取得。同时，GIE将由AT8A513K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 并返回到原始执行顺序。

当发生任何硬件中断时，中断标志寄存器INTF的相应位将被设置为 1。该位只有在程序将 0 写入该位时才会被清除为零。因此，用户可以通过轮询寄存器INTF来获得哪个事件导致硬件中断的信息。需注意只有当中断使能寄存器INTE的相应位设置为 1 时，才会读取相应的中断标志。如果中断使能寄存器INTE的相应位设置为 1，GIE也为 1，将发生硬件中断，下一条指令将从 0x008 中取得。同时，AT8A513K将自动清除寄存器GIE位为零。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 并返回原始执行顺序。

3.12.1 Timer0 上溢中断

Timer0 上溢（从 0x00 到 0xFF）将设置寄存器T0IF位。如果T0IE和GIE设置为 1，则将处理此中断请求。

3.12.2 Timer1 下溢中断

Timer1 下溢（从 0xFF到 0x00）将设置寄存器T1IF位。如果T1IE和GIE设置为 1，则将处理此中断请求。

3.12.3 WDT 超时中断

当WDT超时且配置字选择WDT超时，产生中断请求时，它将设置寄存器WDTIF。如果WDTIE和GIE设置为 1，则将处理此中断请求。

3.12.4 PB 输入状态改变中断

当PBx ($0 \leq x \leq 5$) 配置为输入引脚且相应的寄存器WUPBx位设置为 1 时，这些选定I/O引脚上的电平变化将设置寄存器PBIF。如果PBIE和GIE设置为 1，则将处理此中断请求。需注意当PB0 同时设置为电平变化中断和外部中断时，外部中断标志EIS将关闭PB0 电平变化操作。

3.12.5 外部中断

根据EIS=1 和INTEDG的配置，I/O引脚PB0 上选定的有效边沿将设置寄存器INTIF，如果INTIE和GIE设置为 1，则将处理此中断请求。

3.12.6 LVD 中断

当VDD电平低于LVD电压时，LVD标志位由高变低，寄存器LVDIF=1。如果LVDIE和GIE设置为 1，这个中断请求将被处理。

3.13 振荡配置

因为AT8A513K是双时钟IC，有高频振荡 (F_{HOSC}) 和低频振荡 (F_{LOSC}) 可选择作为系统振荡 (F_{OSC})。可用作 F_{HOSC} 的振荡器是内部高频RC振荡器 (I_HRC)。可用作 F_{LOSC} 的振荡器是内部低频RC振荡器 (I_LRC) 或外部低频晶振振荡 (E_LXT)。

- (1) $STPHOSC(OSCCR[1]) = 1$ will stop F_{HOSC} .
(2) F_{HOSC} will be disabled automatically at Halt mode

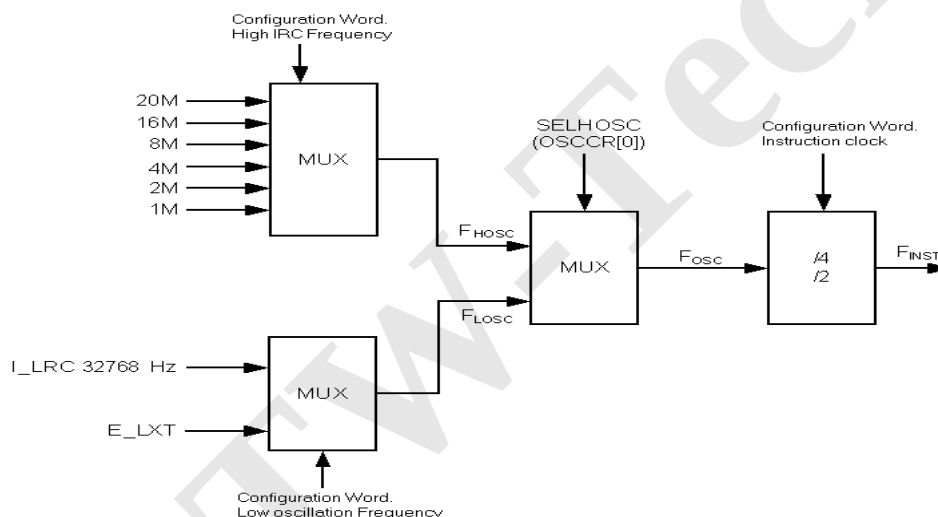


图 19 AT8A513K振荡配置结构图

决定用作 F_{Hosc} 的振荡器有两个配置字。当 I_HRC 选择作为 F_{Hosc} 时， I_HRC 输出频率由三个配置字决定，可以是1M、2M、4M、8M、16M或20MHz。此外，外部晶振脚PB4和PB5可用作I/O脚。另一方面，PB4可以根据一个配置字的设置来用作指令时钟输出脚。PB4/PB5不能用作I/O脚，他们必须用作晶振的输出和输入脚。PB4是晶振输出脚（Xout），PB5是晶振输入脚（Xin）。

决定用作 F_{Losc} 的振荡器有一个配置字。当选择 I_LRC 时，其频率以32768Hz为主。如果 F_{Losc} 需要外部晶振，选择 E_LXT ，并且只允许32768Hz的晶体管。当选择了 E_LXT 时，PB4/PB5不能用作I/O脚，它们必须用作晶振的输出和输入脚。PB4是晶振输出脚（Xout），PB5是晶振输入脚（Xin）。 F_{Hosc} and F_{Losc} 的双时钟组合列表如下：

No.	F_{Hosc}	F_{Losc}
1	I_HRC	I_LRC
2	I_HRC	E_LXT

表 12 双时钟组合

当 E_LXT 作为其中一个振荡时，晶体或谐振器连接到Xin和Xout提供振荡。此外，芯片内还内置了两个电容。用户可以选择禁用/7pf/9pf/12.5pf。

C1和C2的推荐值如下表所示。

振荡模式	晶体管频率(Hz)	C1, C2 (pF)
E_LXT	32768	10 ~ 30

表 13 晶体振荡 E_LXT 的推荐值C1和C2

当选择禁用内置Xtal电容时，为了获得精确而稳定的32.768k频率，选择正确的C1和C2值很重要。您需要将C1/C2电容与您选择的特定晶体相匹配。每个晶体数据表都列出了负载电容（CL），C1和C2值是用以下公式选择的：

$$C1=C2=2*CL$$

用户可以在配置选项中选择C1/C2。

例如，当晶体 $CL=12.5pF$ 时，建议设置 $C1=C2=25pF$ 。

根据寄存器SELHOSC (OSCCR [0])的值，可以选择 F_{Hosc} 或 F_{Losc} 作为系统振荡频率 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} ，根据配置字的值，指令时钟 F_{INST} 可以是 $F_{Osc}/2$ 或 $F_{Osc}/4$ 。

3.14 工作模式

AT8A513K提供了四种工作模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，AT8A513K将停止几乎所有的运作，除了定时器T0/T1/看门狗定时器以定期唤醒。在睡眠模式下，AT8A513K将睡眠直到外部事件或WDT触发IC来唤醒。

四种工作模式如下图所示。

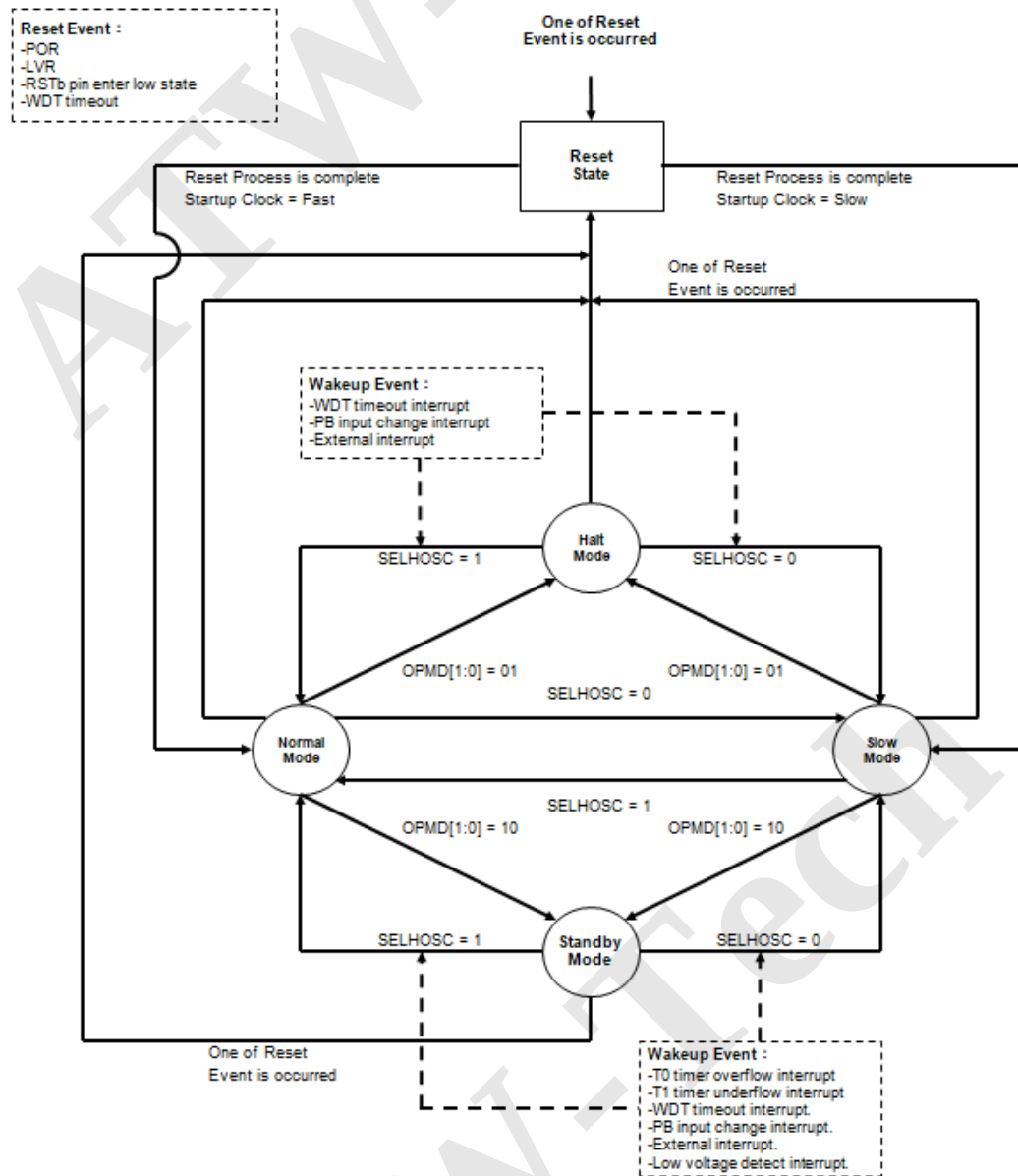


图 20 四种工作模式

3.14.1 正常模式

发生任何复位事件并且复位过程完成后，AT8A513K将在正常模式或慢速模式下开始执行程序。复位过程后选择的模式由启动时钟配置字决定。如果启动时钟为I_HRC，AT8A513K将进入正常模式，如果启动时钟为I_LRC，AT8A513K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，其功耗在四种工作模式中将是最大的。在上电或任何复位触发器被释放后，待复位程序完成，AT8A513K将进入正常模式。

- 基于F_{LOSC}和所有外围模块的执行指令，根据相应的模块使能位可能被激活。
- F_{LOSC}仍处于有效状态并运行。
- IC可通过写 0 至寄存器SELHOSC (OSCCR[0])切换为慢速模式。
- IC可通过编程寄存器OPMD[1:0] (OSCCR[3:2])切换为待机或睡眠模式。
- 关于实时时钟的应用，AT8A513K在运行正常模式可同时为低频时钟。低振荡频率连接到Timer0 时钟，这是通过设置LCKTM0 为 1 和相应的配置字Timer0 源设置为 1 来实现。

3.14.2 慢速模式

通过写 0 至寄存器SELHOSC，AT8A513K将进入慢速模式。在慢速模式下，为节省功耗，F_{LOSC}被选为系统振荡器但仍能保持IC运行。然而，F_{HOSC}将不会自动被AT8A513K关闭。因此在慢速模式下，用户可写 0 至寄存器STPHOSC (OSCCR[1])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}，否则该程序可能会当机。

- 基于F_{LOSC}和所有外围模块的执行指令，根据相应的模块使能位可能被激活。
- 通过写 1 至寄存器STPHOSC，F_{HOSC}可以被关闭。
- IC可通过编程寄存器OPMD[1: 0]切换为待机模式或睡眠模式。
- IC可通过写 1 至寄存器SELHOSC切换为正常模式。

3.14.3 待机模式

通过写入 10b至寄存器OPMD[1:0]，AT8A513K将进入待机模式。然而，在待机模式下，F_{HOSC}不会自动被AT8A513K关闭，用户必须进入慢速模式并写入 1 至寄存器STPHOSC，以停止F_{HOSC}振荡寄存器。大部分AT8A513K的外围模块会被关闭，但是如果T0EN/T1EN被设置为 1 则定时器仍然有效。因此Timer0/Timer1 失效后AT8A513K能唤醒，失效时间是由寄存器TMR0/TMR1、F_{INST}和用于Timer0/Timer1 的其他配置字决定。

- 执行指令停止，一些外围模块根据相应的模块使能位可能被激活。
- 通过写 1 至寄存器STPHOSC，F_{HOSC}可以被关闭。
- F_{LOSC}仍保持有效并持续运行。
- 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 (b)看门狗超时中断 (c)PB输入状态改变中断 (d)发生INT外部中断 (e)LVD中断。
- 在从待机模式唤醒后，如果SELHOSC=1，IC将返回到正常模式，如果SELHOSC=0，IC将返回到慢速模式。
- 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。

3.14.4 睡眠模式

通过执行SLEEP指令或写入 01b至寄存器OPMD[1:0]，AT8A513K将进入睡眠模式。在进入睡眠模式后，寄存器/PD (STATUS[3])将被清除为 0，寄存器/TO (STATUS[4])将被设置为 1 且WDT将清除但保持运行状态。

在睡眠模式下，所有外围模块是被关闭的，执行指令是停止的且AT8A513K只能通过一些特殊事件唤醒。因此，睡眠模式是AT8A513K所提供最省电的模式。

- 执行指令停止，所有外围模块关闭。
- 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)PB输入状态改变中断 (c)发生INT外部中断。
- 从睡眠模式唤醒后，如果SELHOSC=1，IC将返回到正常模式，如果SELHOSC=0，IC将返回到慢速模式。
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。

3.14.5 唤醒稳定时间

睡眠模式的唤醒稳定时间为 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AT8A513K进入待机模式或睡眠模式之前，用户可以执行指令ENI。在这种情况下，AT8A513K将跳转到地址 0x008，以便在唤醒后执行中断服务程序。如果在进入待机模式或睡眠模式之前执行DISI指令，在唤醒后，下一条指令将会被执行。

3.14.6 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
指令执行	执行	执行	停止	停止
Timer0/1	T0 使能 / T1 使能	T0 使能 / T1 使能	T0 使能 / T1 使能	关闭
WDT	选项和WDT使能	选项和WDT使能	选项和WDT使能	选项和WDT使能
其它模块	模块使能位	模块使能位	模块使能位	关闭所有
唤醒源	-	-	- Timer0 上溢 - Timer1 下溢 - WDT超时 - PB输入口改变 -中断 - LVD	- WDT超时 - PB输入口改变 -中断

表 14 工作模式概述

3.15 复位

当以下任一复位事件发生时，AT8A513K将会进入复位状态并开始复位动作：

- 当VDD检测到上升沿时，出现上电复位。
- 当工作电压低于预设的电压时，出现低压复位（LVR）。
- RSTb引脚输入低电平。
- 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它们的初始值或保持不变。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
RSTb 从非睡眠模式复位	不变	不变
RSTb 从睡眠模式复位	1	1
WDT 从非睡眠模式复位	0	1
WDT 从睡眠模式复位	0	0
执行SLEEP指令	1	0
执行CLRWDT指令	1	1

表 15 /TO和/PD值和相关事件概述

复位事件发生后，AT8A513K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。振荡器稳定后，AT8A513K将进一步等待 Fosc 的 16 个时钟周期（OST，振荡器启动时间），复位过程完成。

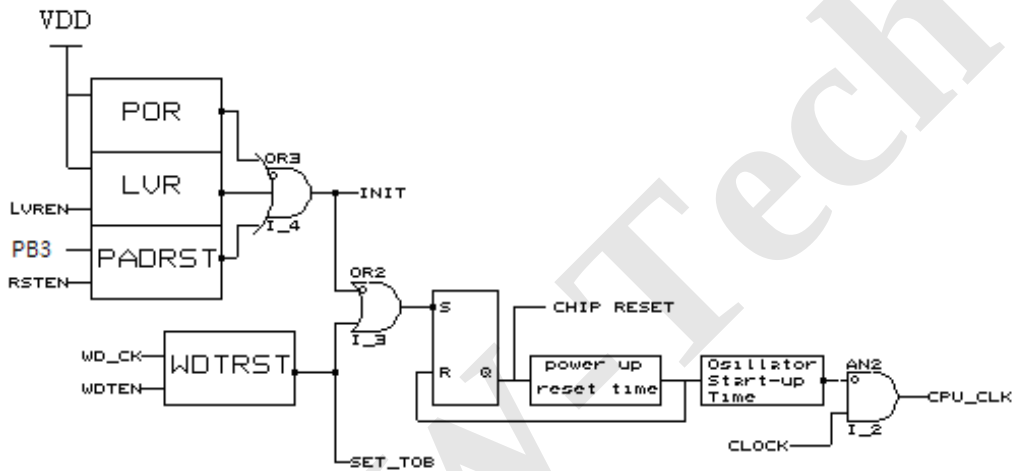


图 21 芯片复位电路框图

为降低上电电压VDD，建议使用RSTb复位，如下图。

- 建议R阻值不大于 40kΩ。
- R1 值= 100Ω ~ 1kΩ时，将阻止大电流，ESD或电气信号进入复位引脚。
- 二极管D 使电容C 能在VDD 掉电时快速放电。

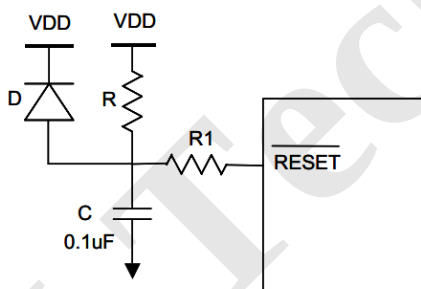


图 22 复位应用结构框图

3.16 比较器相关功能

3.16.1 比较器功能简介

- 电压比较器
- 低电压检测 (LVD)
- 内置电阻分压器模块，监控电源电压 VDD
- LVD 中断

3.16.2 电压比较器

AT8A513K 提供电压比较器和内部参考电压，具有各种模拟比较模式。比较器非反相输入和反相输入可与 GPIO 共享。

CMPEN (寄存器 PCON[2]) 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。

比较器结构如下图所示：

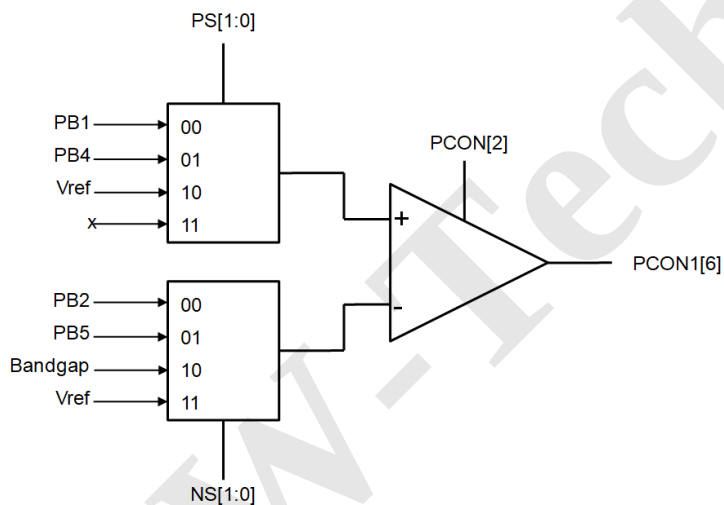


图 23 比较器结构图

比较器的非反相输入由PS[1:0]（寄存器CMPCR[3:2]）确定。

表格如下所示：

PS[1:0]	正相输入
00	PB1
01	PB4
10	Vref
11	---

表 16 正相输入选择

比较器的反相输入由 NS[1:0] 确定(寄存器CMPCR[1:0])。

表格如下所示：

NS[1:0]	反相输入
00	PB2
01	PB5
10	Bandgap (0.65V)
11	Vref

表 17 反相输入选择

有两种方法可以获得比较器的输出结果：一种是通过寄存器轮询，另一种是通过探测输出脚。

比较器输出可以通过LVDOOUT(寄存器PCON1[6])进行轮询。

为了探测输出脚上的比较器输出，请设置 CMPOE(寄存器 OSCCR[6])为 1，然后 PB1 就是比较器输出的实时状态。需要注意，当 CMPOE=1 时，PWM3 的功能如果是启用状态，此时它将会被禁用。

3.16.3 比较器参考电压（Vref）

内部参考电压Vref由串联电阻构成，提供不同电平的参考电压。RBIAS_H和RBIAS_L用于选择Vref的最大值和最小值，LVDS[3:0]用于选择 64 个电压电平中的一个。当VDD电平低于Vref电压时，LVD标志由高变低，并设置寄存器位LVDIF=1。当比较器的结果与之前的值不同时，会产生比较器中断。如果LVDIE和GIE设置为 1，这个中断请求将被处理。

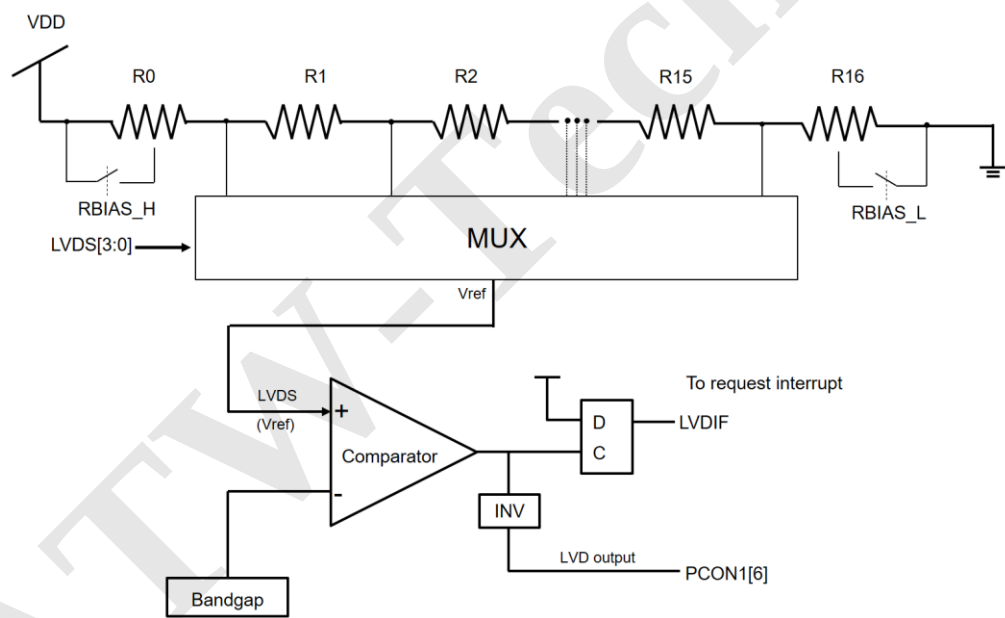


图 24 Vref硬件连接与 LVD结构框图

Vref由RBIAS_H， RBIAS_L和LVDS[3:0]决定。LVDS[3:0]用于从 64 个参考电压中选择一个，如下表所示。

Condition				
	VIN- = Vref(V)			
LVRs[3:0]	RBIAS[H:L]=[0:0]	RBIAS[H:L]=[1:0]	RBIAS[H:L]=[0:1]	RBIAS[H:L]=[1:1]
1100	0.161*VDD	0.257*VDD	0.075*VDD	0.127*VDD
1010	0.164*VDD	0.262*VDD	0.078*VDD	0.133*VDD
1000	0.170*VDD	0.271*VDD	0.084*VDD	0.143*VDD
1111	0.177*VDD	0.283*VDD	0.093*VDD	0.157*VDD
0111	0.183*VDD	0.292*VDD	0.099*VDD	0.168*VDD
1110	0.190*VDD	0.304*VDD	0.107*VDD	0.182*VDD
0110	0.199*VDD	0.318*VDD	0.117*VDD	0.199*VDD
1101	0.207*VDD	0.330*VDD	0.125*VDD	0.213*VDD
0101	0.216*VDD	0.345*VDD	0.135*VDD	0.230*VDD
0100	0.224*VDD	0.356*VDD	0.144*VDD	0.244*VDD
0011	0.231*VDD	0.369*VDD	0.152*VDD	0.258*VDD
1011	0.248*VDD	0.395*VDD	0.170*VDD	0.289*VDD
0010	0.268*VDD	0.427*VDD	0.192*VDD	0.327*VDD
0001	0.291*VDD	0.463*VDD	0.218*VDD	0.370*VDD
0000	0.317*VDD	0.505*VDD	0.246*VDD	0.419*VDD
1001	0.327*VDD	0.521*VDD	0.258*VDD	0.438*VDD

表 18 参考电压Vref选择表

注：Vref偏差为±0.1V。

基于串联电阻分压器产生的内部参考电压（Vref），比较器可用于监控电源电压。当反相输入设置为内部带隙基准（0.65V），非反相输入连接到内部电阻分压器输出（Vref）时，比较器将电源电压与配置的阈值进行比较。图 24 是Vref硬件连接和LVD框图。

- 当电源电压低于阈值时，比较器输出为 0。
- 当电源电压高于阈值时，比较器输出为 1。

3.16.4 低电压检测（LVD）

AT8A513K低电压检测器（LVD）内置精密带隙基准，精确检测VDD电平。内置 14 级精密低压检测电路，用户可直接选择所需检测电压点。如果LV DEN（寄存器PCON[5]）=1，且VDD电压值低于LVDS[3:0]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中断标志将是高的，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。

下表为LVD电压选择表。

RBIAS[H:L]	LVDS[3:0]	LVD Voltage (V)
00	0001	2.20
00	0010	2.40
00	1011	2.60
00	0011	2.80
00	0100	2.90
00	0101	3.00
00	1101	3.15
00	0110	3.30
00	1110	3.45
00	0111	3.60
00	1111	3.75
00	1000	3.90
00	1010	4.05
00	1100	4.15

表 19 LVD 电压选择

LVD 操作流程：

- 第一步 由 LVDS[3:0] 选择LVD 电压
- 第二步 设置 CMPCR = 0x0A
- 第三步 设置 PCON[5]=1 （开启LVD）
- 第四步 通过 PCON1[6] 检查LVD 状态

注意：若LVDS[3:0] 选择的电压是变化的，用户需延迟至少 50us (@F_{HOSC}=1MHz)方能通过PCON1[6]得到正确的LVD状态。

3.16.5 LVD 中断

当VDD电平低于LVD电压时，LVD标志由高变低，并设置寄存器位LVDIF=1。当比较器的结果与之前的值不同时，会产生比较器中断。如果LVDIE和GIE设置为 1，这个中断请求将被执行。

4. 指令设置

AT8A513K 为各种应用程序提供了 55 个强大的指令。

指令	操作符		操作	周期	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	操作符		操作	周期	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	-
CALL	adr		Call subroutine	2	-
GOTO	adr		unconditional branch	2	-
LCALL	adr		Call subroutine	2	-
LGOTO	adr		unconditional branch	2	-

表 20 指令设置

ACC: 累加器。

adr: 即时寻址。

bit: 一个 8 位寄存器R的位寻址。

C: 进位/借位。

C=1, 加法指令发生进位, 减法指令不发生借位。

C=0, 加法指令不发生进位, 减法指令发生借位。

d: 目标

若d= “0”, 结果存入ACC。

若d= “1”, 结果存入R寄存器。

DC: 数字进位标记。

dest: 目标。

F: F 页面特殊功能寄存器, F 值为 0x5~0xF。

i: 8 位即时数据。

PC: 程序计数器。

PCHBUF: 程序计数器的高位缓冲器。

/PD: 电源中断标志位。

/PD=1, 上电或CLRWDWT指令执行后。

/PD=0, SLEEP指令执行后。

Prescaler: 预分频器 0 分频率。

R: R页面特殊功能寄存器, R值为 0x00~0x3F。

S: S页面特殊功能寄存器, S值为 0x0 ~ 0xF。

T0MD: T0MD寄存器。

TBHP: ROM高位目标地址。

TBHD: ROM高位目标地址的存储数据。

/TO: 计时器溢出标志位。

/TO=1, 上电或执行 CLRWDWT 或 SLEEP 指令后。

/TO=0, WDT 溢出。

WDT: 看门狗计时器。

Z: 清零标志。

ADCAR		ADDAR	
Add ACC and R with Carry		Add ACC and R	
语法	ADCAR R, d	语法	ADDAR R, d
操作数	$0 \leq R \leq 63$ $d = 0, 1.$	操作数	$0 \leq R \leq 63$ $d = 0, 1.$
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		ADDIA	
Add ACC and Immediate with Carry		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	$0 \leq R \leq 63$. $d = 0, 1$.	操作数	$0 \leq R \leq 63$ $0 \leq \text{bit} \leq 7$
操作	$\text{ACC} \& R \rightarrow \text{dest}$	操作	$0 \rightarrow R[\text{bit}]$
状态影响	Z	状态影响	--
说明	ACC和R做“与”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	$0 \leq R \leq 63$ $0 \leq \text{bit} \leq 7$
操作	$\text{ACC} \& i \rightarrow \text{ACC}$	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“与”运算，结果存入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	CALL	Call Subroutine
语法	BTRSC R, bit	语法	CALL adr
操作数	$0 \leq R \leq 63$ $0 \leq \text{bit} \leq 7$	操作数	$0 \leq \text{adr} < 255$
操作	Skip next instruction, if R[bit] = 0.	操作	PC + 1 → Top of Stack {PCHBUF, adr} → PC
状态影响	--	状态影响	--
说明	位测试指令，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 8 位立即地址载入 PC[7:0]，将 PCHBUF[1:0] 载入 PC[9:8]
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	CALL SUB 执行指令前： PC=A0. Stack pointer=1 执行指令后： PC=address of SUB, Stack[1] = A0+1, Stack pointer=2.

BTRSS	Test Bit in R and Skip if Set	CALLA	Call Subroutine
语法	BTRSS R, bit	语法	CALLA
操作数	$0 \leq R \leq 63$ $0 \leq \text{bit} \leq 7$	操作数	--
操作	Skip next instruction, if R[bit] = 1.	操作	PC + 1 → Top of Stack {TBHP, ACC} → PC
状态影响	--	状态影响	--
说明	位测试指令，为“1”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[1:0] 赋值给 PC[9:8]，将ACC 赋值给PC[7:0]
周期	1 or 2（跳过）	周期	2
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	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.

CLRA	Clear ACC	CLRWDT	Clear Watch-Dog Timer
语法	CLRA	语法	CLRWDT
操作数	--	操作数	--
操作	00h → ACC 1 → Z	操作	00h → WDT, 00h → WDT prescaler 1 → /TO 1 → /PD
状态影响	Z	状态影响	/TO, /PD
说明	ACC清零, Z标志位置“1”	说明	清WDT计数器和预分频器; /TO和 /PD标志位置“1”
周期	1	周期	1
举例	CLRA 执行指令前: ACC=0x55, Z=0. 执行指令后: ACC=0x00, Z=1.	举例	CLRWDT 执行指令前: /TO=0 执行指令后: /TO=1

CLRR	Clear R	COMR	Complement R
语法	CLRR R	语法	COMR R, d
操作数	$0 \leq R \leq 63$	操作数	$0 \leq R \leq 63$ $d = 0, 1.$
操作	00h → R 1 → Z	操作	$\sim R \rightarrow \text{dest}$
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寄存器取补, 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.

CMPAR	Compare ACC and R	DECR	Decrease R
语法	CMPAR R	语法	DECR R, d
操作数	$0 \leq R \leq 63$	操作数	$0 \leq R \leq 63$ $d = 0, 1.$
操作	$R - ACC \rightarrow$ (No restore)	操作	$R - 1 \rightarrow \text{dest}$
状态影响	Z, C	状态影响	Z
说明	ACC和R比较: 执行ACC-R, 不改变ACC和R的值, 只改变Z和C标志位	说明	R递减, 若d="0", 结果存入ACC; 若d="1", 结果存入R。
周期	1	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.	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.

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	$0 \leq R \leq 63$ $d = 0, 1.$
操作	ACC(hex) $\rightarrow$ ACC(dec)	操作	$R - 1 \rightarrow \text{dest},$ Skip if result = 0
状态影响	C	状态影响	--
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。需注意, 中断要在加法指令前关闭DAA指令后打开, 避免错误结果。	说明	R 先递减, 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, 因此这条指令要执行两个周期。
周期	1	周期	1 or 2 (跳过)
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	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, 因为操作结果为 0, 指令将会跳过指令 2 执行。

DISI		GOTO	
Disable Interrupt Globally		Unconditional Branch	
语法	DISI	语法	GOTO adr
操作数	--	操作数	0 ≤ adr < 511
操作	Disable Interrupt, 0 → GIE	操作	{PCHBUF, adr} → PC
状态影响	--	状态影响	--
说明	GIE设置为 0，关闭全局中断	说明	无条件短跳转指令，9 位立即地址装入PC<8:0>，PCHBUF[1]装入PC[9]
周期	1	周期	2
举例	DISI 执行指令前： GIE=1. 执行指令后： GIE=0.	举例	GOTO Level 执行指令前： PC=A0. 执行指令后： PC=address of Level.

ENI		GOTOA	
Enable Interrupt Globally		Unconditional Branch	
语法	ENI	语法	GOTOA
操作数	--	操作数	--
操作	Enable Interrupt, 1 → GIE	操作	{TBHP, ACC} → PC
状态影响	--	状态影响	--
说明	GIE设置为 1，开启全局中断	说明	无条件跳转指令，ACC值装入PC<7:0>；TBHP[1:0] 值装入PC<9:8>
周期	1	周期	2
举例	ENI 执行指令前： GIE=0. 执行指令后： GIE=1.	举例	GOTOA 执行指令前： PC=A0. TBHP=0x02, ACC=0x34. 执行指令后： PC=0x234.

INCR	Increase R	INT	Software Interrupt
语法	INCR R, d	语法	INT
操作数	$0 \leq R \leq 63$ $d = 0, 1.$	操作数	--
操作	$R + 1 \rightarrow \text{dest.}$	操作	$PC + 1 \rightarrow \text{Top of Stack,}$ $001h \rightarrow PC$
状态影响	Z	状态影响	--
说明	R递增, 若d="0", 结果存入ACC; 若d="1", 结果存入R	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[9:0]。
周期	1	周期	3
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.	举例	INT 执行指令前: PC=address of INT code. 执行指令后: PC=0x01.

INCRSZ	Increase R, Skip if 0	IORAR	OR ACC with R
语法	INCRSZ R, d	语法	IORAR R, d
操作数	$0 \leq R \leq 63$ $d = 0, 1.$	操作数	$0 \leq R \leq 63$ $d = 0, 1.$
操作	$R + 1 \rightarrow \text{dest,}$ Skip if result = 0	操作	$ACC R \rightarrow \text{dest}$
状态影响	--	状态影响	Z
说明	R先递增, 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, 因此这条指令要执行两个周期。	说明	ACC和R做或运算, 若d="0", 结果 存入ACC; 若d="1", 结果存入R
周期	1 or 2 (跳过)	周期	1
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	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.

IORIA	OR Immediate with ACC
语法	IORIA i
操作数	$0 \leq i < 255$
操作	$ACC \mid i \rightarrow ACC$
状态影响	Z
说明	ACC和 8 位立即数做或运算，结果存入ACC
周期	1
举例	IORIA i 执行指令前： i=0x50, ACC=0xAA, Z=0. 执行指令后： ACC=0xFA, Z=0.

IOSTR	Move F-page SFR to ACC
语法	IOSTR F
操作数	$0 \leq F \leq 15$
操作	F-page SFR $\rightarrow$ ACC
状态影响	--
说明	将F页面特殊寄存器的F赋值给ACC
周期	1
举例	IOSTR F 执行指令前： F=0x55, ACC=0xAA. 执行指令后： F=0x55, ACC=0x55.

IOST	Load F-page SFR from ACC
语法	IOST F
操作数	$0 \leq F \leq 15$
操作	$ACC \rightarrow F\text{-page SFR}$
状态影响	--
说明	将ACC的值赋给F页面特殊寄存器的F
周期	1
举例	IOST F 执行指令前： F=0x55, ACC=0xAA. 执行指令后： F=0xAA, ACC=0xAA.

LCALL	Call Subroutine
语法	LCALL adr
操作数	$0 \leq \text{adr} \leq 1023$
操作	$PC + 1 \rightarrow \text{Top of Stack}$, $\text{adr} \rightarrow PC[9:0]$
状态影响	--
说明	长调用子程序。首先将PC+1压入栈顶，然后将 10 位立即数载入PC[9:0]。
周期	2
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.

LGOTO Unconditional Branch		MOVIA Move Immediate to ACC	
语法	LGOTO adr	语法	MOVIA i
操作数	$0 \leq \text{adr} \leq 1023$	操作数	$0 \leq i < 255$
操作	$\text{adr} \rightarrow \text{PC}[9:0]$.	操作	$i \rightarrow \text{ACC}$
状态影响	--	状态影响	--
说明	无条件长跳转，10位立即数装入PC[9:0]	说明	8位立即数赋值给ACC
周期	2	周期	1
举例	LGOTO Level 执行指令前： PC=A0. 执行指令后： PC=address of Level.	举例	MOVIA i 执行指令前： i=0x55, ACC=0xAA. 执行指令后： ACC=0x55.

MOVAR Move ACC to R		MOVR Move to ACC or R	
语法	MOVAR R	语法	MOVR R, d
操作数	$0 \leq R \leq 63$	操作数	$0 \leq R \leq 63$ $d = 0, 1$.
操作	$\text{ACC} \rightarrow R$	操作	$R \rightarrow \text{dest}$
状态影响	--	状态影响	Z
说明	ACC赋值给R	说明	R赋值给d, 若d="0", 结果存入ACC; 若d="1", 结果存入R。指令执行后, 通过状态标志位Z检查R是否为0。
周期	1	周期	1
举例	MOVAR R 执行指令前： R=0x55, ACC=0xAA. 执行指令后： R=0xAA, ACC=0xAA.	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.

NOP	No Operation	RETIA	Return with Data in ACC
语法	NOP	语法	RETIA i
操作数	--	操作数	$0 \leq i < 255$
操作	No operation.	操作	$i \rightarrow \text{ACC}$, Top of Stack $\rightarrow$ PC
状态影响	--	状态影响	--
说明	空操作	说明	带参数返回：8 位立即数赋值给 ACC，栈顶地址载入PC，GIE标志 为 1
周期	1	周期	2
举例	NOP 执行指令前： PC=A0 执行指令后： PC=A0+1	举例	RETIA i 执行指令前： GIE=0, Stack pointer =2, i=0x55, ACC=0xAA. 执行指令后： GIE=1, PC=Stack[2], Stack pointer =1, ACC=0x55.

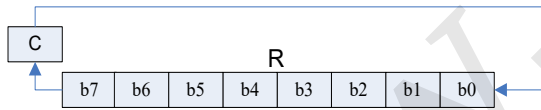
RETIE	Return from Interrupt and Enable Interrupt Globally	RET	Return from Subroutine
语法	RETIE	语法	RET
操作数	--	操作数	--
操作	Top of Stack $\rightarrow$ PC 1 $\rightarrow$ GIE	操作	Top of Stack $\rightarrow$ PC
状态影响	--	状态影响	--
说明	中断返回，栈顶地址载入PC同时使 能中断	说明	子程序返回，栈顶载入PC
周期	2	周期	2
举例	RETIE 执行指令前： GIE=0, Stack level=2. 执行指令后： GIE=1, PC=Stack[2], Stack level =1.	举例	RET 执行指令前： Stack level=2. 执行指令后： PC=Stack[2], Stack level=1.

RLR Rotate Left R Through Carry

语法 RLR R, d

操作数 $0 \leq R \leq 63$
d = 0, 1.

操作 $C \rightarrow \text{dest}[0], R[7] \rightarrow C,$
 $R[6:0] \rightarrow \text{dest}[7:1]$



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

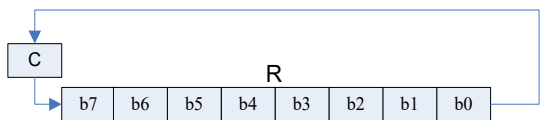
举例 RLR R, d
执行指令前：
R=0xA5, d=1, C=0.
执行指令后：
R=0x4A, C=1.

RRR Rotate Right R Through Carry

语法 RRR R, d

操作数 $0 \leq R \leq 63$
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 1

举例 RRR R, d
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 $0 \leq R \leq 63$
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	$0 \leq S \leq 15$
操作	ACC $\rightarrow$ S-page SFR
状态影响	--
说明	写S页面特殊函数寄存器
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	$0 \leq S \leq 15$
操作	S-page SFR $\rightarrow$ ACC
状态影响	--
说明	读S页面特殊函数寄存器
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.

SLEEP**Enter Halt Mode**

语法	SLEEP
操作数	--
操作	00h $\rightarrow$ WDT, 00h $\rightarrow$ WDT prescaler 1 $\rightarrow$ /TO 0 $\rightarrow$ /PD
状态影响	/TO, /PD
说明	WDT和0分频器清零。/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.

SUBAR Subtract ACC from R		SWAPR Swap High/Low Nibble in R	
语法	SUBAR R, d	语法	SWAPR R, d
操作数	$0 \leq R \leq 63$ $d = 0, 1.$	操作数	$0 \leq R \leq 63$ $d = 0, 1.$
操作	$R - ACC \rightarrow dest$	操作	$R[3:0] \rightarrow dest[7:4].$ $R[7:4] \rightarrow dest[3:0]$
状态影响	Z, DC, C	状态影响	--
说明	R 减去Acc, 若d="0", 结果存入Acc; 若d="1", 结果存入R	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
周期	1	周期	1
举例	SBCAR 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)	举例	SWAPR R, d 执行指令前: R=0xA5, d=1. 执行指令后: R=0x5A.

SUBIA Subtract ACC from Immediate		TABLEA Read ROM data	
语法	SUBIA i	语法	TABLEA
操作数	$0 \leq i < 255$	操作数	--
操作	$i - ACC \rightarrow ACC$	操作	ROM data{ TBHP, ACC } [7:0] $\rightarrow ACC$ ROM data{TBHP, ACC} [13:8] $\rightarrow TBHD.$
状态影响	Z, DC, C	状态影响	--
说明	常数减ACC, 结果存入ACC	说明	ROM查表指令, 高字节存入TBH, 低字节存入ACC
周期	1	周期	2
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)	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.

T0MD	Load ACC to T0MD	XORAR	Exclusive-OR ACC with R
语法	T0MD	语法	XORAR R, d
操作数	--	操作数	$0 \leq R \leq 63$ $d = 0, 1.$
操作	ACC $\rightarrow$ T0MD	操作	ACC $\oplus$ R $\rightarrow$ dest
状态影响	--	状态影响	Z
说明	写T0 模式寄存器	说明	ACC和R做“异或”运算，若d=“0”，结果存入ACC；若d=“1”，结果存入R
周期	1	周期	1
举例	T0MD 执行指令前： T0MD=0x55, ACC=0xAA. 执行指令后： T0MD=0xAA.	举例	XORAR R, d 执行指令前： R=0xA5, ACC=0xF0, d=1. 执行指令后： R=0x55.

T0MDR	Move T0MD to ACC	XORIA	Exclusive-OR Immediate with ACC
语法	T0MDR	语法	XORIA i
操作数	--	操作数	$0 \leq i < 255$
操作	T0MD $\rightarrow$ ACC	操作	ACC $\oplus$ i $\rightarrow$ ACC
状态影响	--	状态影响	Z
说明	读T0 模式寄存器	说明	ACC和常数做“异或”运算，若d=“0”，结果存入ACC；若d=“1”，结果存入R
周期	1	周期	1
举例	T0MDR 执行指令前： T0MD=0x55, ACC=0xAA. 执行指令后： ACC=0x55.	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 IRC Frequency (IRC 高频率)	1. 1MHz 4. 8MHz	2. 2MHz 5. 16MHz	3. 4MHz 6. 20MHz			
2	Low Oscillator Frequency (低频振荡)	1. I_LRC	2. E_LXT				
3	Inst Clock Output (指令时钟输出)	1. PB4 is IO (PB4 是一般 I/O 脚)	2. PB4 output instruction clock (PB4 是指令输出脚)				
4	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)			
5	WDT (看门狗定时器)	1. Watchdog Enable (Software control) (看门狗使能 (软件控制)) 2. Watchdog Disable (Always disable) (看门狗关闭 (永远关闭))					
6	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)			
7	Timer0 Source (定时源)	1. EX_CK1	2. Low Oscillator 3. E_XT(455KHz~4MHz)				
8	Reset (复位)	1. PB3 is reset (PB3 是复位脚)	2. PB3 is IO (PB3 是一般 I/O 脚)				
9	PWM1EN	1. PB2 is PWM output (PB2 是 PWM 输出脚)	2. PB2 is IO (PB2 是一般 I/O 脚)				
10	Startup Time (启动时间)	1. 140us	2. 4.5ms	3. 18ms	4. 72ms	5. 288ms	
11	WDT Time Base (看门狗定时器时基)	1. 3.5ms	2. 15ms	3. 60ms	4. 250ms		
12	LVR Setting (LVR 设定)	1. Register Control (寄存器控制) 3. Always On (一直)	2. Register Control + Halt mode Off (寄存器控制+睡眠模式关闭) 4. Operation mode On + Halt mode Off (工作模式开启+睡眠模式关闭)				
13	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	
14	VDD Voltage (VDD 电压)	1. 3.0V	2. 4.5V		3. 5.0V		
15	Read Output Data (读取输出数据)	1. I/O port		2. Register			
16	E_LXT Backup Control (E_LXT 控制)	1. Auto Off		2. Register Off			
17	Startup Clock (启动时钟)	1. Fast (I_HRC)		2. Slow (I_LRC/E_LXT)			
18	Input High Voltage (V _{IH}) (输入高电压 (V _{IH}))	1. CMOS (0.7VDD)		2. TTL (0.5VDD)			
19	Input Low Voltage (V _{IL}) (输入低电压 (V _{IL}))	1. CMOS (0.3VDD)		2. TTL (0.2VDD)			
20	Input Voltage (V _{IH} /V _{IL}) (输入电压 (V _{IH} / V _{IL}))	1. Schmitt		2. No Schmitt			
21	Crystal Load Capacitance (only for 32768Hz) (晶体负载电容) (只支持 32768Hz)	1. Disable	2. C _L =7.0pF	3. C _L =9.0pF	4. C _L =12.5pF		
22	Drive /Sink Current Type @5V	1. Small		2. Normal			

项目	名称	选项	
23	SPWM[0]	1. PB1 = PWM2, PB4 = IO	2. PB4 = PWM2, PB1 = IR(by Register)
24	SPWM[1]	1. PB0 = PWM3, PB5 = RESZ(by Register)	2. PB5 = PWM3, PB0 = IO
25	PB1	1. CMP analog input (CMP 模拟输入)	2. Digital input (数字输入)
26	PB2	1. CMP analog input (CMP 模拟输入)	2. Digital input (数字输入)
27	PB4	1. CMP analog input (CMP 模拟输入)	2. Digital input (数字输入)
28	PB5	1. CMP analog input (CMP 模拟输入)	2. Digital input (数字输入)

表 21 配置表

6. 电气特性

6.1 最大绝对值

符号	参数	额定值	单位
$V_{DD} - V_{SS}$	电源电压	-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

($F_{INST}=F_{HOSC}/4$, $F_{HOSC}=16MHz@I_HRC$, WDT 启用, 无特殊说明时, 环境温度 $T_A=25^{\circ}C$.)

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	$F_{INST}=10MHz @ I_HRC 20MHz/2T$
			2.7				$F_{INST}=8MHz @ I_HRC 16MHz/2T$
			2.2				$F_{INST}=5MHz @ I_HRC 20MHz/4T$
			2.0				$F_{INST}=4MHz @ I_HRC 16MHz/4T \& 8MHz/2T$
			1.6				$F_{INST}=2MHz @ I_HRC 8MHz/4T \& 4MHz/2T$
			2.0				$F_{INST}=1MHz @ E_XT 4MHz/4T$ (Crystal)
			1.6				$F_{INST}=1MHz @ I_HRC 4MHz/4T$
			1.6				$F_{INST}=8KHz @ I_LRC 32KHz/4T$
			1.6				$F_{INST}=8KHz @ E_LXT 32KHz/4T$ (Crystal)
V_{IH}	输入高电压	5V	3.9	--	--	V	RSTb (0.8V _{DD})
		3V	2.3	--	--		
		5V	3.7	--	--	V	All other I/O pins, EX_CK1, INT 0.8V _{DD}
		3V	2.3	--	--		
		5V	2.8	--	--	V	All other I/O pins, EX_CK1, INT 0.6V _{DD}
		3V	1.8	--	--		
		5V	--	2.9	--	V	All other I/O pins, EX_CK1, INT No Schmitt Trigger (0.5V _{DD})
		3V	--	1.8	--		
V_{IL}	输入低电压	5V	--	--	1.5	V	RSTb (0.2V _{DD})
		3V	--	--	1.0		
		5V	--	--	1.8	V	All other I/O pins, EX_CK1, INT 0.3V _{DD}
		3V	--	--	1.0		
		5V	--	--	1.0	V	All other I/O pins, EX_CK1, INT 0.2V _{DD}
		3V	--	--	0.7		
		5V	--	2.8	--	V	All other I/O pins, EX_CK1, INT No Schmitt Trigger (0.5V _{DD})
		3V	--	1.7	--		
I_{OH}	高输出电流 (小电流)	5V	--	2.2	--	mA	$V_{OH}=4.0V$
		3V	--	1.2	--		$V_{OH}=2.0V$
	高输出电流 (正常电流)	5V	--	16	--	mA	$V_{OH}=4.0V$
		3V	--	9	--		$V_{OH}=2.0V$
I_{OL}	低输出电流 (小电流)	5V	--	6.5	--	mA	$V_{OL}=1.0V$
		3V	--	3.6	--		
	低输出电流	5V	--	36	--	mA	$V_{OL}=1.0V$

符号	参数	V _{DD}	最小值	典型值	最大值	单位	条件
	(正常电流)	3V	--	22	--		
I _{OP}	工作电流	正常模式					
		5V	--	1.33	--	mA	F _{HOSC} =20MHz @ I _{_HRC} /2
		3V	--	0.51	--		
		5V	--	1.22	--	mA	F _{HOSC} =16MHz @ I _{_HRC} /2
		3V	--	0.46	--		
		5V	--	1.15	--	mA	F _{HOSC} =20MHz @ I _{_HRC} /4
		3V	--	0.41	--		
		5V	--	1.09	--	mA	F _{HOSC} =16MHz @ I _{_HRC} /4
		3V	--	0.37	--		
		5V	--	0.94	--	mA	F _{HOSC} =8MHz @ I _{_HRC} /4
		3V	--	0.32	--		
		5V	--	0.87	--	mA	F _{HOSC} =4MHz @ I _{_HRC} /4
		3V	--	0.28	--		
		5V	--	0.75	--	mA	F _{HOSC} =2MHz @ I _{_HRC} /4
		3V	--	0.26	--		
		5V	--	0.71	--	mA	F _{HOSC} =1MHz @ I _{_HRC} /4
		3V	--	0.24	--		
		慢速模式					
		5V	--	2.34	--	uA	F _{HOSC} 禁用 F _{LOSC} =32KHz @ I _{_LRC} /2
		3V	--	4.35	--		
		5V	--	10.1	--	uA	F _{HOSC} 禁用 F _{LOSC} =32KHz @ E _{_LXT} /2.
		3V	--	3.81	--		
		5V	--	2.92	--	uA	F _{HOSC} 禁用 F _{LOSC} =32KHz @ I _{_LRC} /4
		3V	--	1.59	--		
		5V	--	8.55	--	uA	F _{HOSC} 禁用 F _{LOSC} =32KHz @ E _{_LXT} /4.
		3V	--	3.05	--		
I _{STB}	待机电流	5V	--	1.41	--	uA	待机模式, F _{HOSC} 禁用 F _{LOSC} =32KHz @ I _{_LRC} /4
		3V	--	1.03	--		
I _{HALT}	睡眠电流	5V	--	--	0.1	uA	睡眠模式, WDT 禁用, LVR 禁用
		3V	--	--	0.1		
		5V	--	--	1.0	uA	睡眠模式, WDT 禁用, LVR 启用
		3V	--	--	0.6		
		5V	--	--	3.0	uA	睡眠模式, WDT 启用, LVR 禁用
		3V	--	--	2.0		
R _{PH}	上拉电阻	5V	--	130	--	KΩ	上拉电阻 (不含PB3)
		3V	--	66	--		
		5V	--	85	--	KΩ	上拉电阻 (含PB3)
		3V	--	85	--		
R _{PL}	下拉电阻	5V	--	125	--	KΩ	下拉电阻
		3V	--	60	--		

6.3 OSC特性

(测量条件VDD, TA温度等于程序条件)

参数	最小值	典型值	最大值	单位	条件
烧录座的I_HRC偏差			±1	%	烧录座直接安装在烧录器上。
烧录机台的I_HRC偏差			±3	%	正确设置烧录机台的条件。
烧录机台的I_LRC偏差			±5	%	

6.4 比较器 / LVD 特性

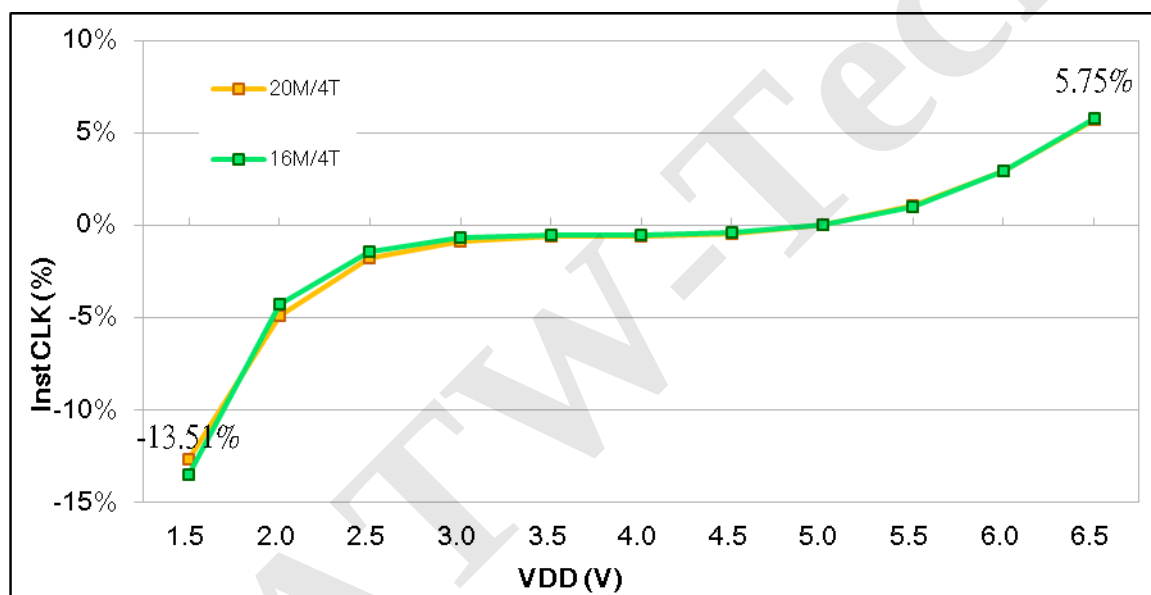
(无特别说明时, VDD=5V, VSS=0V, TA=25°C)

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	--	VDD-1.5	V	F _{HOSC} =1MHz
T _{ENO}	比较器启用有效输出	--	20	--	us	F _{HOSC} =1MHz
I _{CO}	比较器工作电流	--	70	--	uA	F _{HOSC} =1MHz, P2V mode
I _{LVD}	LVD工作电流	--	85	--	uA	F _{HOSC} =1MHz, LVD=4.15V
E _{LVD}	LVD电压误差	--	3	--	%	F _{HOSC} =1MHz, LVD=4.15V
CP _{OS}	比较器偏移量	--	10	20	mV	

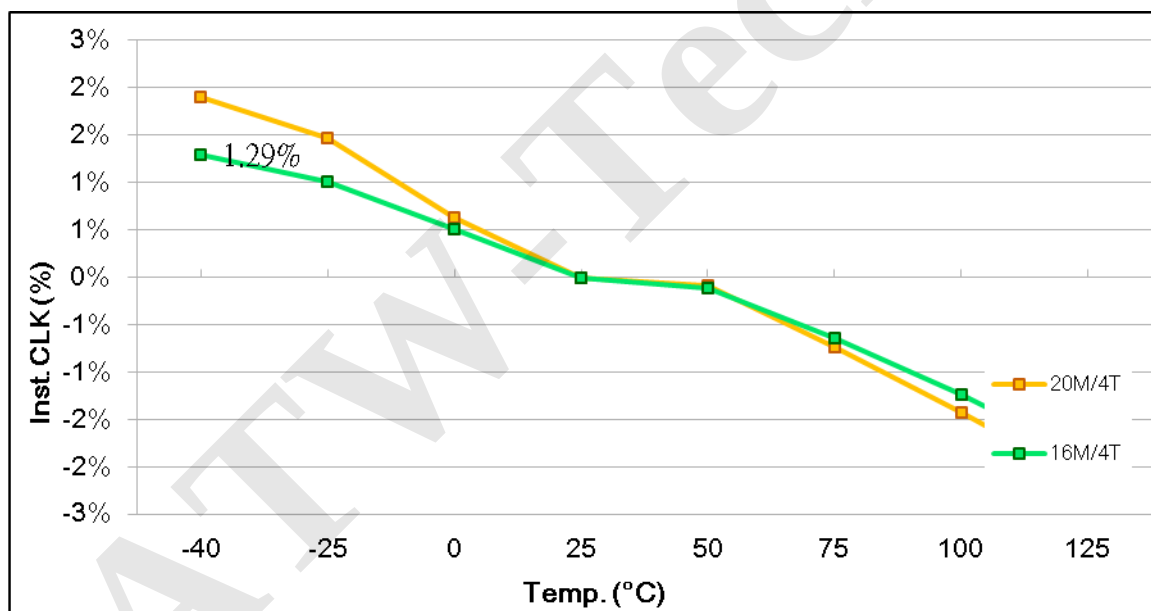
*注：这些参数仅供设计参考，并非针对每个芯片进行测试。

6.5 特性图

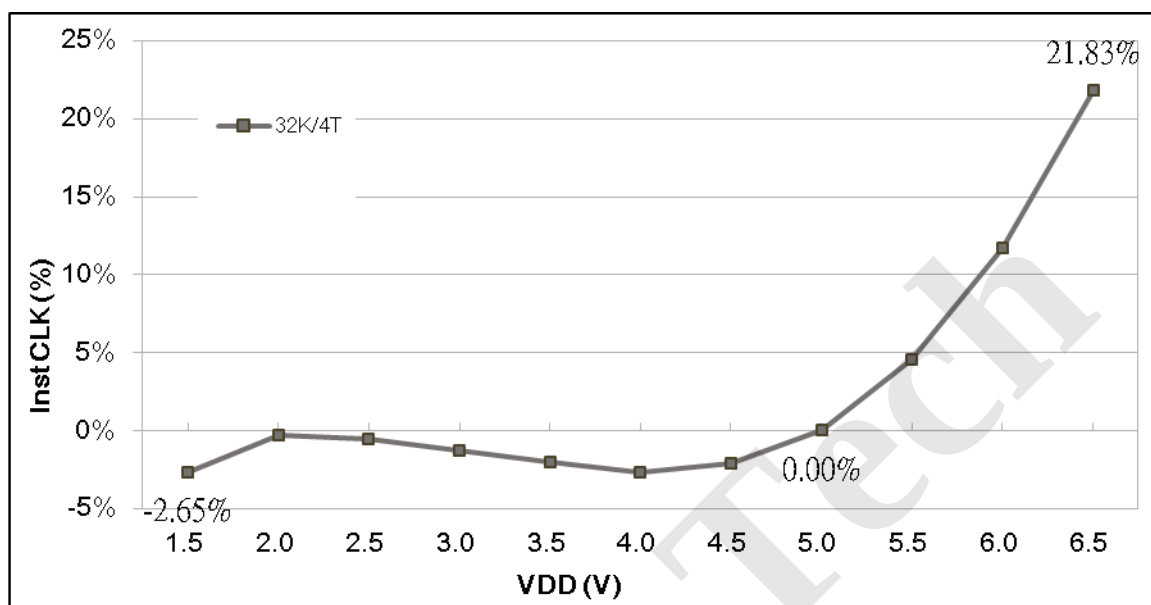
6.5.1 频率与高速振荡电压



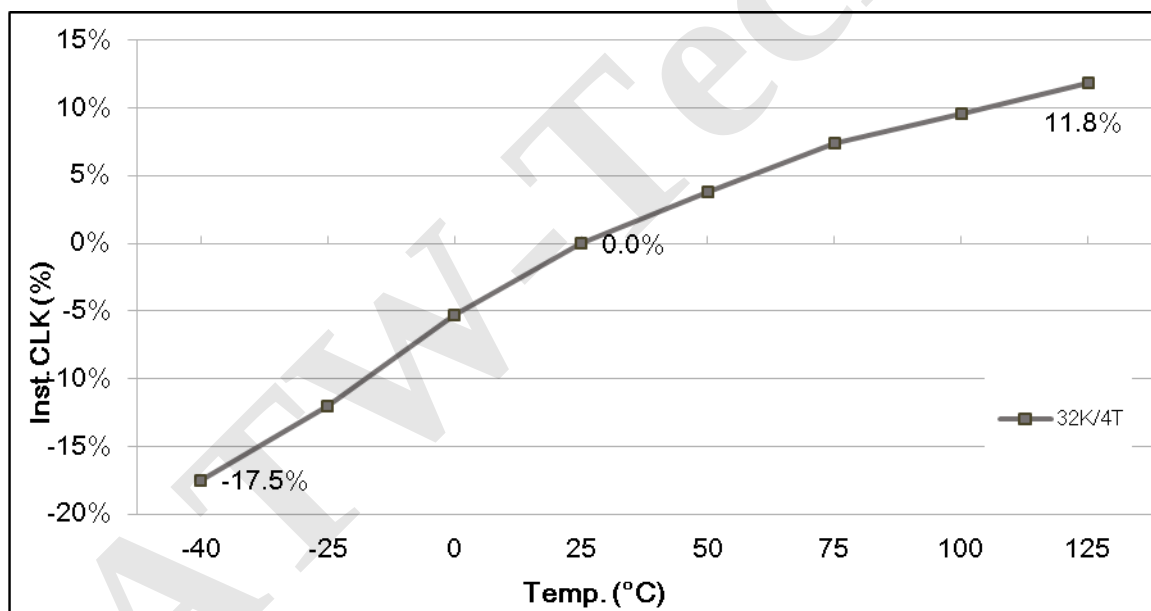
6.5.2 频率与高速振荡温度



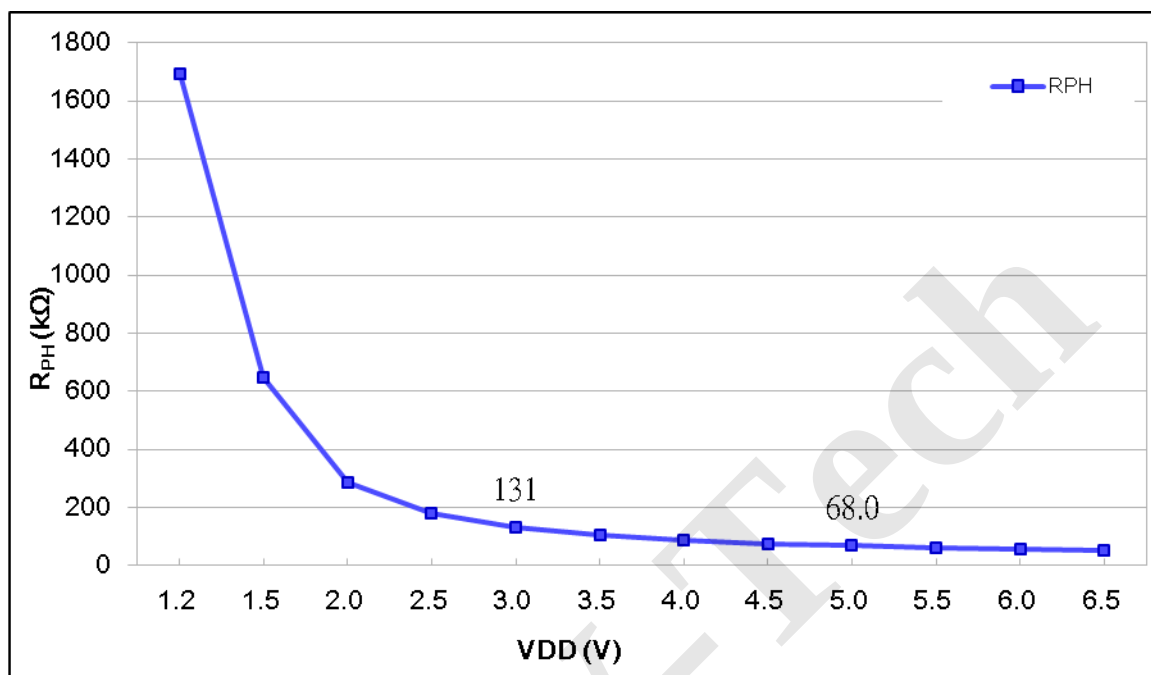
6.5.3 频率与低速振荡电压



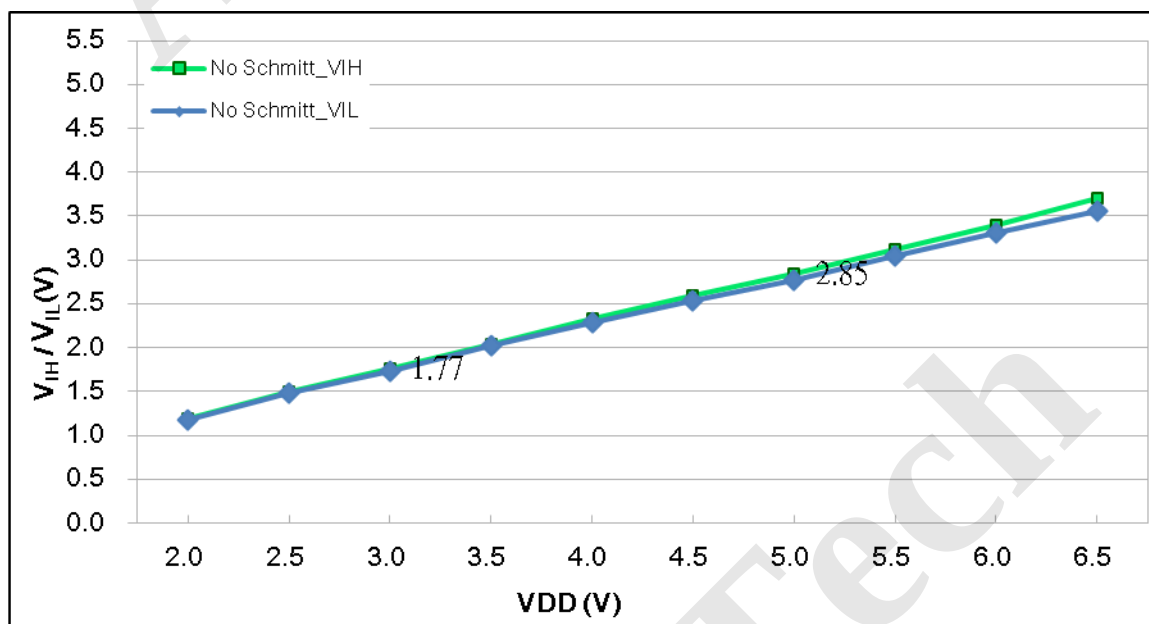
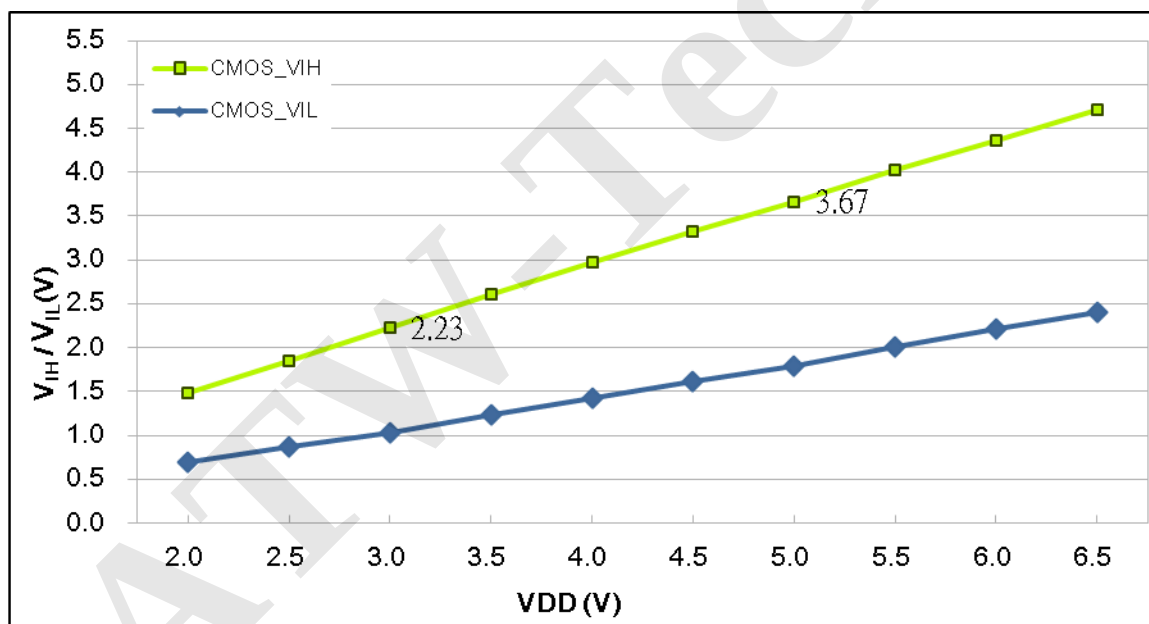
6.5.4 频率与低速振荡温度

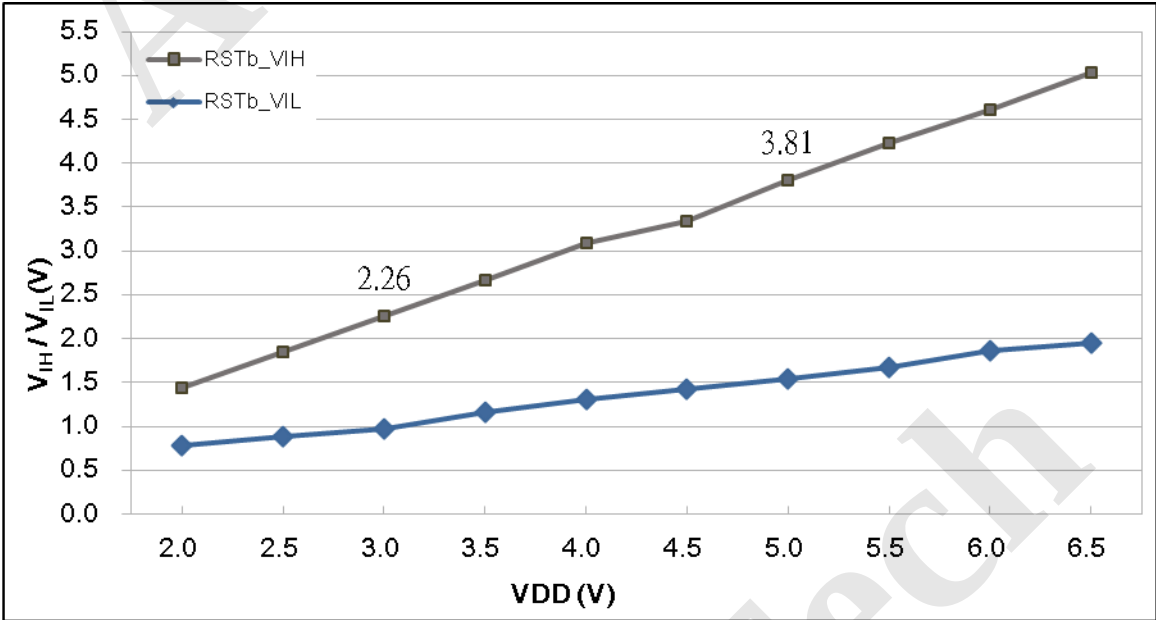
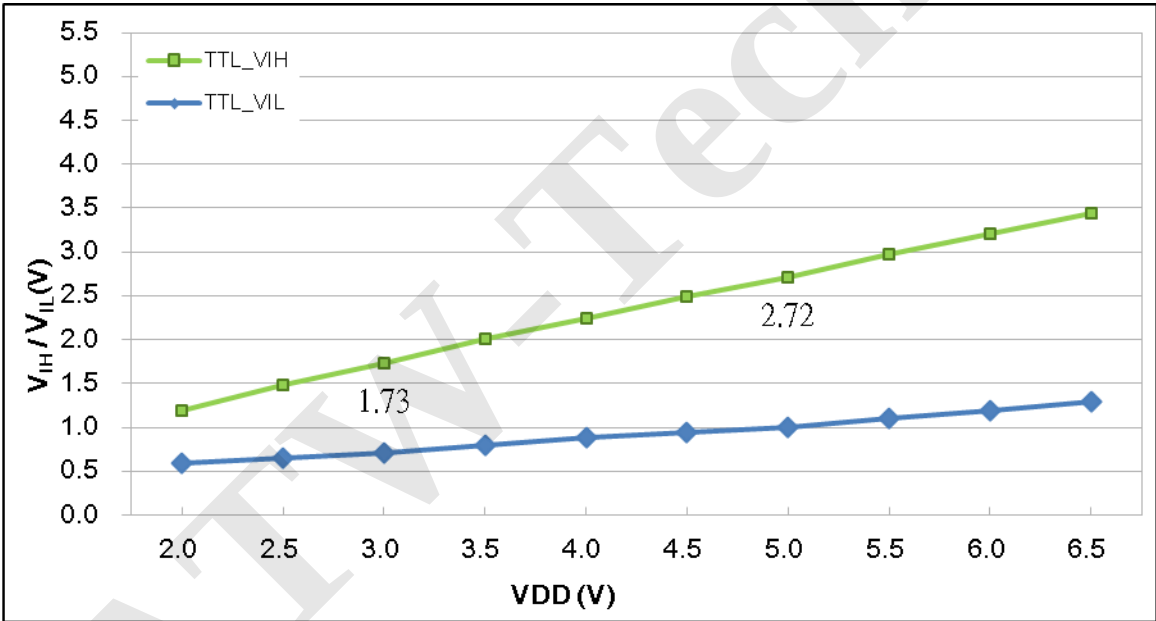


6.5.5 上拉电阻与 VDD

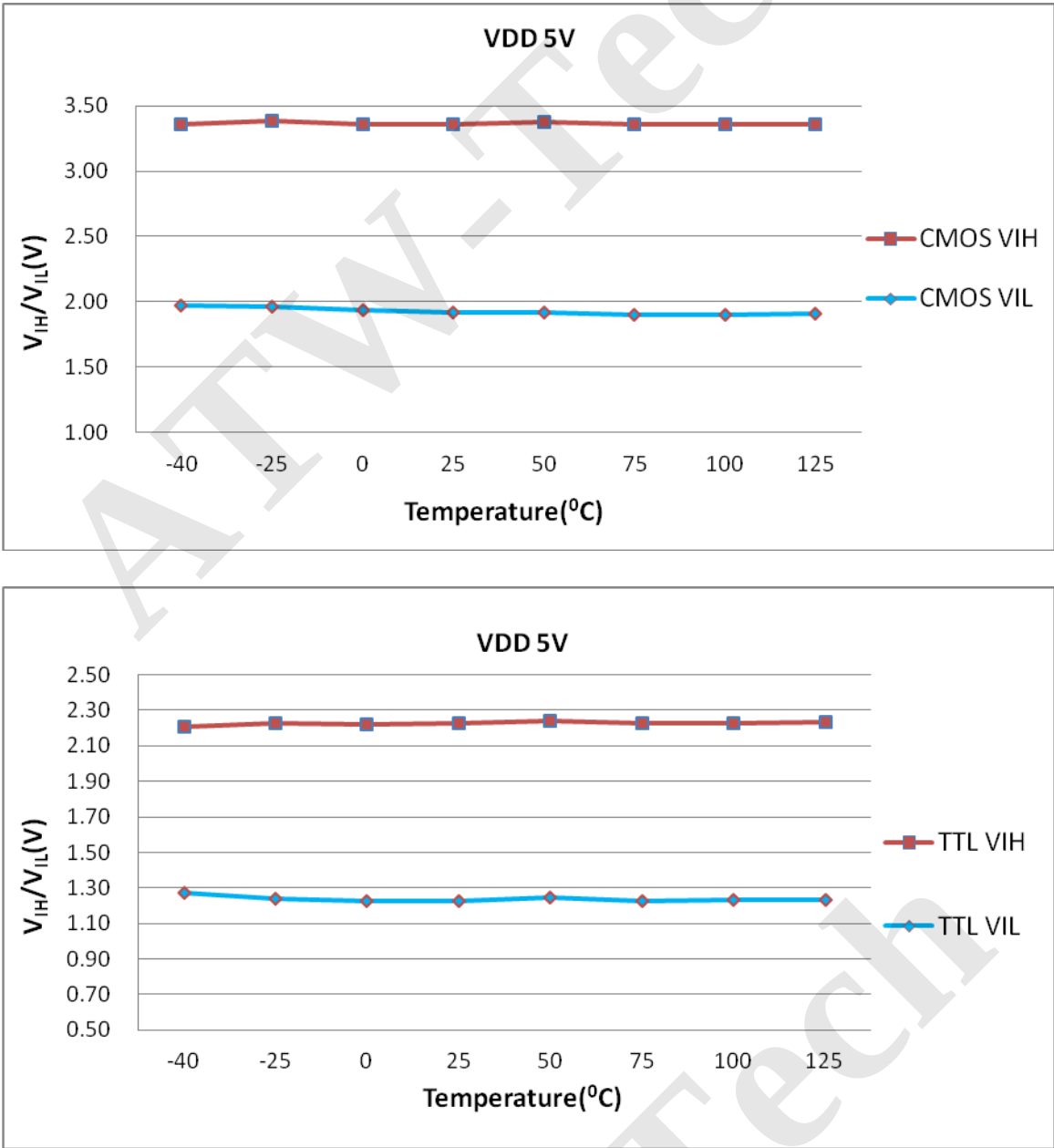


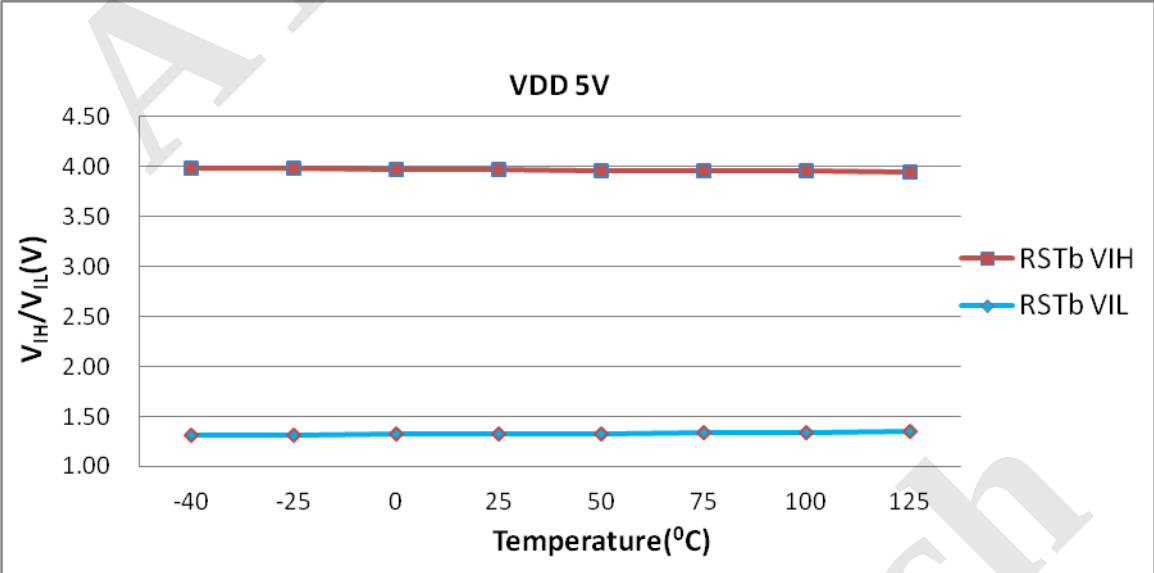
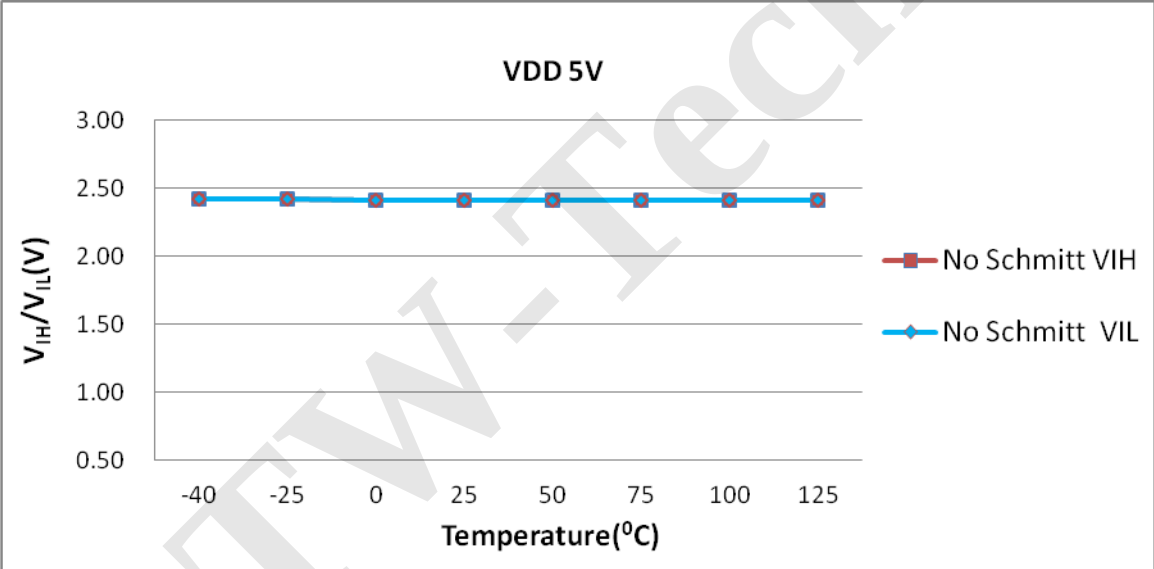
6.5.6 VIH/VIL 与 VDD





6.5.7 VIH/VIL 与温度





6.6 建议工作电压

1. 建议工作电压 (温度范围: -40 °C ~ +85 °C)

频率	LVR 默认设置	LVR 设置 (最小值@25°C)
20M/2T	3.6V	3.3V
16M/2T	3.0V	2.7V
20M/4T	2.4V	2.2V
16M/4T	2.2V	2.0V
8M/4T	2.0V	1.8V
4M/4T	1.8V	1.8V

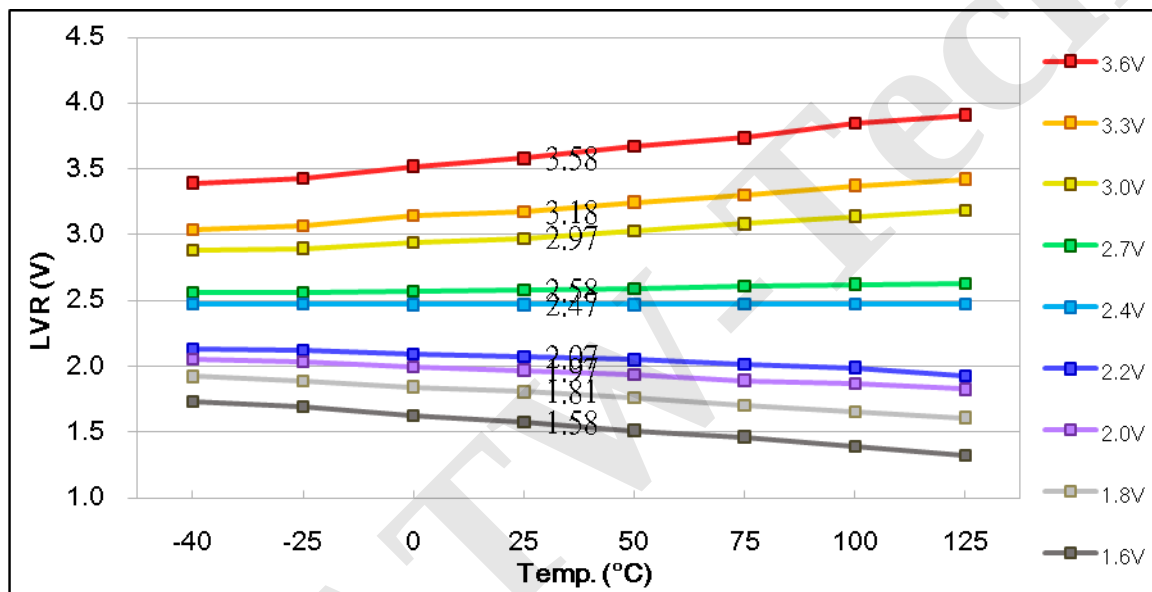
*注: 这些参数仅供设计参考, 并非针对每个芯片进行测试。

2. 旁路预分频器 (TM1_HRC=1)

频率	LVR : 推荐旁路预分频器
20M/2T	3.3V @ 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

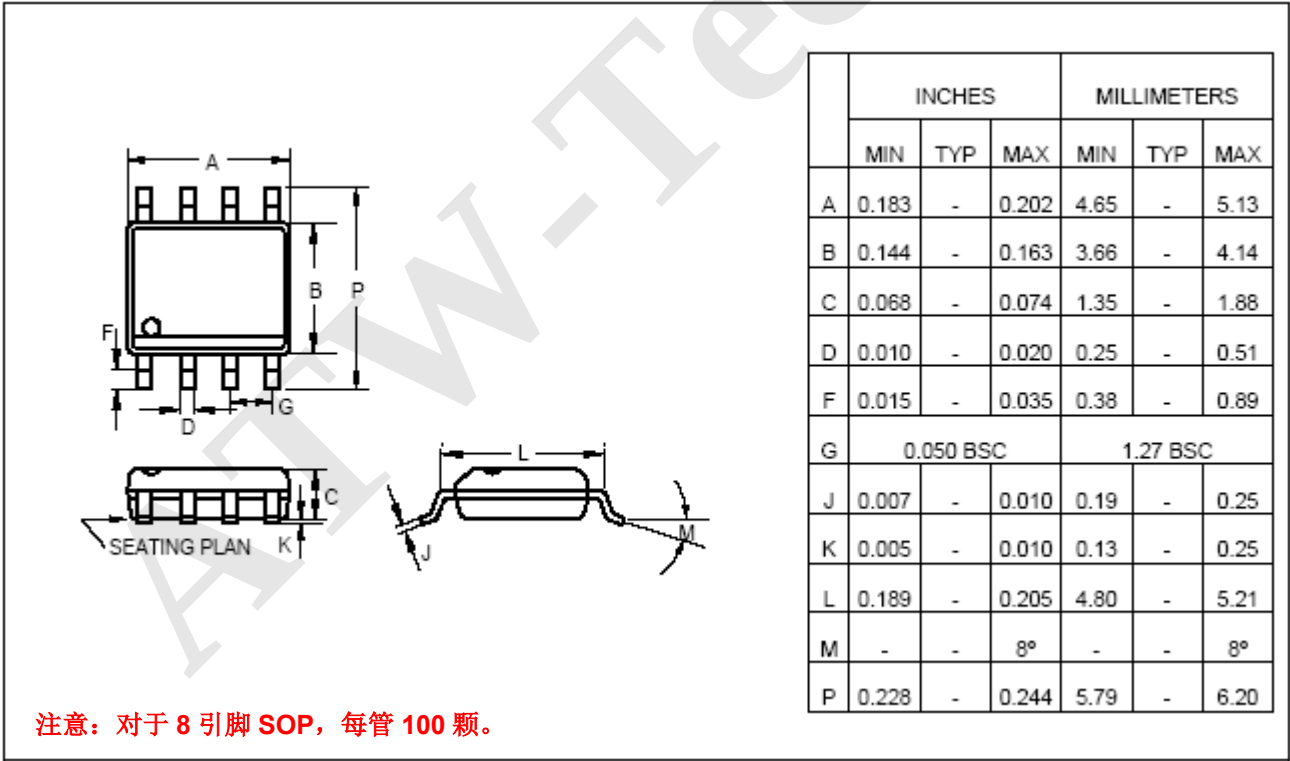
*注: 这些参数仅供设计参考, 并非针对每个芯片进行测试。

6.7 LVR与温度



7. 封装尺寸

7.1 8-Pin Plastic 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
AT8A513K	Die	--	--	--
AT8A513KS8	SOP	8	150 mil	卷装：每卷 2.5K颗。 管装：每管 100 颗。