

# MC32P8112 用户手册

(原产品名 MC32P64)

*SinoMCU 8 位单片机*

2018/05/10



**sino**mcu  
晟矽微电子

上海晟矽微电子股份有限公司

Shanghai SinoMCU Microelectronics Co., Ltd.

---

本公司保留对产品在可靠性、功能和设计方面的改进作进一步说明的权利  
用户手册的更改，恕不另行通知

目录

1.	产品简介 .....	4
1.1	产品特性 .....	4
1.2	系统框图 .....	6
1.3	引脚排列 .....	7
1.4	引脚说明 .....	10
1.5	管脚结构 .....	10
2.	中央处理器 .....	13
2.1	指令集 .....	13
2.2	程序存储器 .....	16
2.3	数据存储器 .....	17
2.4	堆栈 .....	18
2.5	烧录配置选项 .....	19
2.6	控制寄存器 .....	21
3.	系统时钟 .....	26
3.1	外接晶体振荡器 .....	26
3.2	内置高频 RC 振荡器 .....	27
3.3	内置低频 RC 振荡器 .....	27
3.4	工作模式 .....	27
3.5	低功耗模式 .....	29
4.	复位 .....	30
4.1	复位条件 .....	30
4.2	上电复位 .....	31
4.3	外部复位 .....	31
4.4	低电压复位 .....	31
4.5	看门狗复位 .....	31
5.	IO 口 .....	32
5.1	IO 工作模式 .....	32
5.2	上拉电阻控制 .....	34
5.3	端口模式控制 .....	35
6.	定时器 .....	36
6.1	看门狗定时器 WDT .....	36
6.2	定时器 T0 .....	36
6.3	定时器 T1 .....	39
6.4	定时器 T2 .....	41
7.	触摸按键 .....	44
7.1	触摸功能介绍 .....	44
7.2	触摸模块操作步骤 .....	45
7.3	触摸模块相关寄存器 .....	46
8.	I2C 总线接口 .....	48
8.1	I2C 接口介绍 .....	48
8.2	I2C 操作步骤 .....	49
8.3	I2C 相关寄存器 .....	49

9	模数转换器.....	51
9.1	ADC 功能介绍.....	51
9.2	ADC 转换时序图.....	51
9.3	ADC 操作步骤.....	51
9.4	ADC 相关寄存器.....	52
10	中断.....	54
10.1	外中断.....	54
10.2	定时器中断.....	54
10.3	触摸计数器中断.....	54
10.4	I2C 中断.....	54
10.5	ADC 转换中断.....	54
10.6	中断相关寄存器.....	55
11	电气参数.....	57
11.1	极限参数.....	57
11.2	直流特性参数.....	57
11.3	ADC 特性参数.....	61
11.4	交流电气参数.....	62
12	特性曲线图.....	63
12.1	IO 口低电平驱动电流 VS 输出电平.....	63
12.2	IO 口高电平驱动电流 VS 输出电平.....	64
12.3	IO 口上拉电阻 VS 电源电压.....	65
12.4	内部低频 RC 频率 VS 电源电压.....	65
12.5	内部低频 RC 动态功耗 VS 电源电压.....	66
12.6	内部低频 RC 静态功耗 VS 电源电压.....	66
12.7	外部 32768Hz 动态功耗 VS 电源电压.....	67
12.8	外部 32768Hz 静态功耗 VS 电源电压.....	67
12.9	常温内部高频 HIRC 电压频率 VS 电源电压.....	68
12.10	常压内部高频 HIRC 电压频率 VS 温度.....	68
12.11	内部高频 HIRC 频率 VS 温度 VS 电源电压.....	69
12.12	最低工作电压 VS 系统时钟 Fcpu 关系图.....	69
13	封装外形尺寸.....	70
14	版本修订记录.....	74

## MC32P8112 用户手册

### 1. 产品简介

MC32P8112 是一款高性能 8 位触摸型 MCU，产品的高抗干扰性能为带触摸按键的小家电产品提供良好的解决方案。

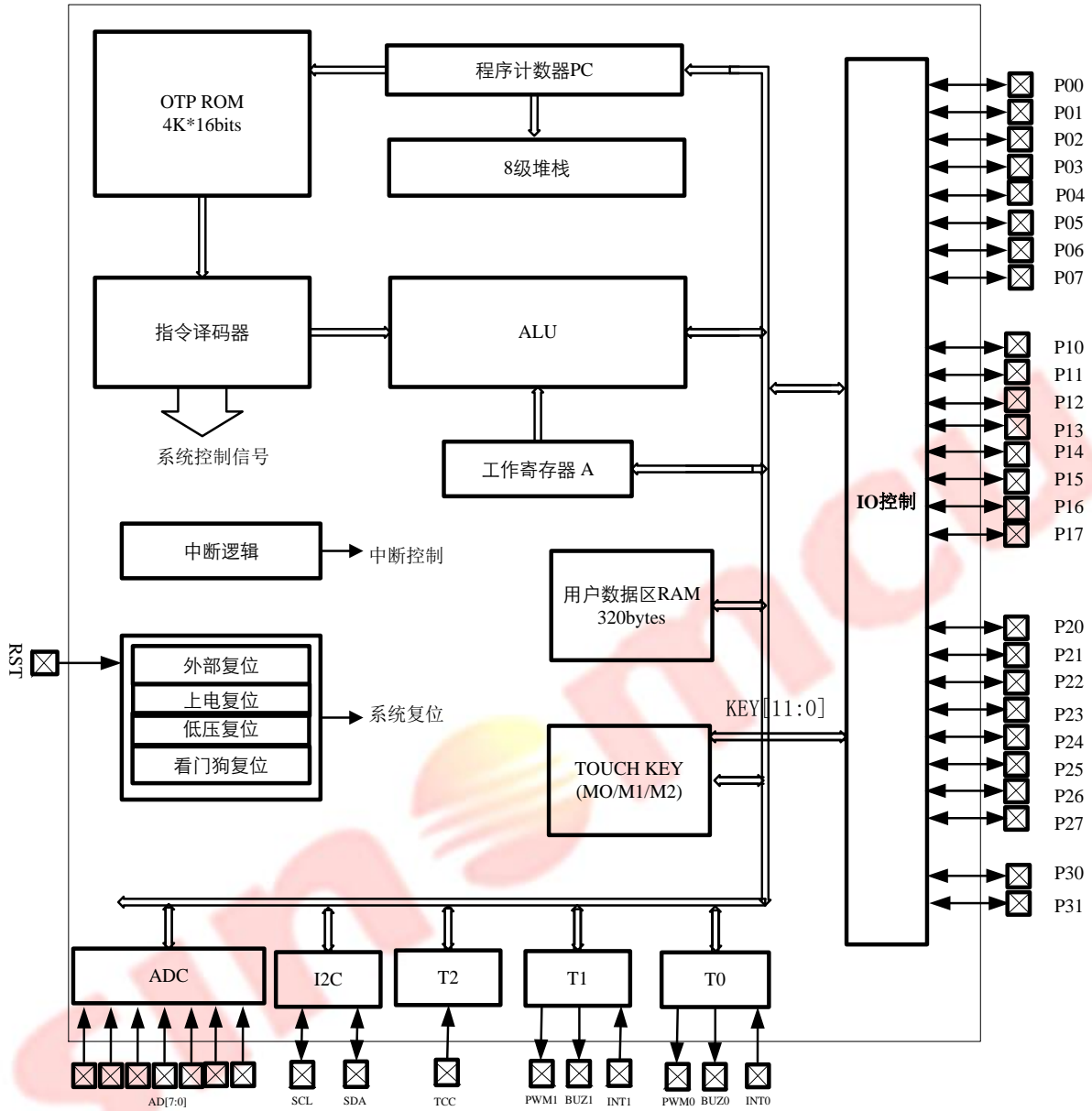
#### 1.1 产品特性

- ◇ 8 位 CPU 内核
  - ✓ 精简指令集
  - ✓ 高频模式下 2T/4T/8T/16T/32T/64T/128T/256T 可设；低频工作模式下为 2T
- ◇ 存储器
  - ✓ 4K\*16 程序存储器 (OTP), 8 级深度硬件堆栈, 通过 INDF3 可读取 ROM 区内容
  - ✓ 320 字节 RAM
- ◇ 25 IO + 1 开漏 IO
  - ✓ 8 位 P0 端口, P00-P07 与 KEY0-KEY7 复用
  - ✓ 8 位 P1 端口, P10-P13 与 KEY8-KEY11 复用, P14-P17 与 AD 复用
  - ✓ 8 位 P2 端口, P20-P23 与 AD 复用, P25 为开漏 IO (编程高压 VPP 复用), P26-P27 与 I2C 复用, P23-P24 与 OSC 复用
  - ✓ 2 位 P3 端口
- ◇ 5 种工作模式
  - ✓ 高速运行模式: 系统在高频时钟下运行
  - ✓ 低速运行模式: 系统在低频时钟下运行
  - ✓ 休眠模式: 所有振荡器停止运行
  - ✓ HOLD 模式 1: CPU 停止运行, 高频振荡器工作
  - ✓ HOLD 模式 2: CPU 停止运行, 高频振荡器停止工作, 低频振荡器工作
- ◇ 内部自振式看门狗计数器 (WDT)
  - ✓ 可配置溢出时间: 16ms/64ms/256ms/1024ms/2048ms/4096ms
  - ✓ 可配置工作模式
- ◇ 2 个带有 PWM、BUZ 和外部计数功能 8 位定时器, 可设置溢出中断
  - ✓ 带有 7 位预分频器的 8 位递减计数器
  - ✓ 自动加载寄存器
  - ✓ 可配置 4 个时钟源: 系统时钟、高频时钟、低频时钟和外部管脚
  - ✓ 8 位 PWM 脉宽设置寄存器 (最小可调制脉宽宽度 62.5ns, 8 位分辨率最小 PWM 周期 16us)
  - ✓ 溢出中断
  - ✓ BUZ 输出
- ◇ 1 个带有外部计数功能 16 位定时器, 可设置为触摸/定时功能, 可设置溢出中断
  - ✓ 带有 7 位预分频器的 16 位递减计数器
  - ✓ 自动加载寄存器
  - ✓ 可配置 4 个时钟源: 系统时钟、高频时钟、低频时钟和外部管脚
  - ✓ 溢出中断
  - ✓ 触摸/定时功能

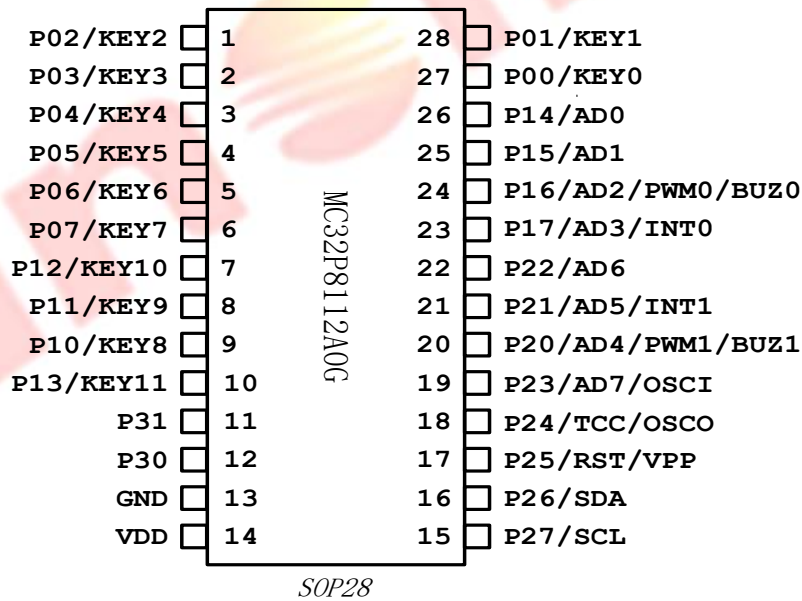
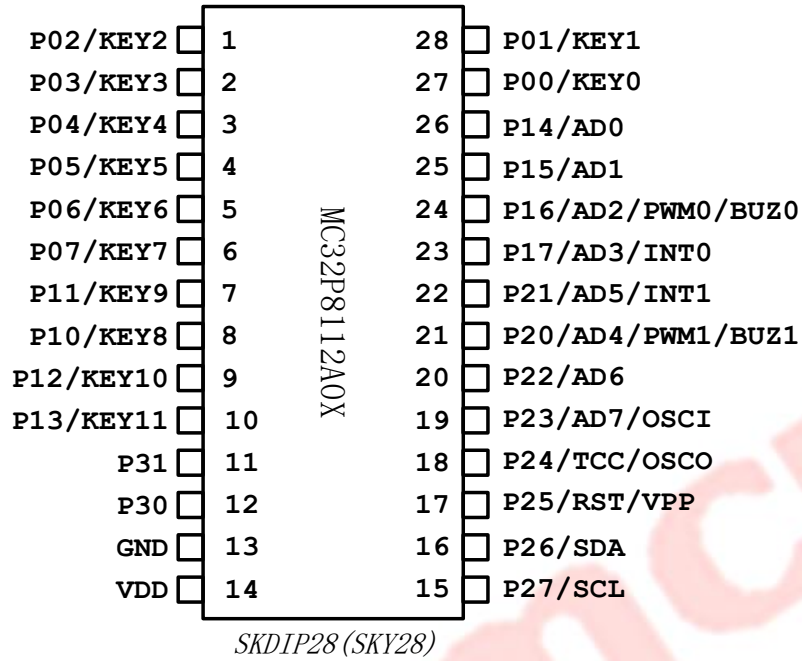
- ◇ 外部中断
  - ✓ 两路外部中断源，可唤醒
- ◇ I2C slave 接口
- ◇ 集成 3 组共 12 个触摸按键，每组 4 个按键共用 1 个 16 位 C/F 计数器
- ◇ 8 路 12 位模数转换器
- ◇ 中断
  - ✓ 两路外部中断源（INT0、INT1）
  - ✓ 定时器 0 中断
  - ✓ 定时器 1 中断
  - ✓ 定时器 2 中断
  - ✓ 触摸计数器 0 中断
  - ✓ 触摸计数器 1 中断
  - ✓ 触摸计数器 2 中断
  - ✓ I2C 中断
  - ✓ ADC 中断
- ◇ 时钟振荡模式
  - ✓ 外接高频晶体振荡器（432KHz-16MHz）+ 内嵌低频振荡器模式（28KHz）
  - ✓ 内嵌高频振荡器（16MHz）+ 外接低频振荡器模式（32768Hz）
  - ✓ 内嵌高频振荡器（16MHz）+ 内嵌低频振荡器（28KHz）
- ◇ 7 级低电压复位 LVR
- ◇ 工作电压
  - ✓ 3.0V-5.5V @Fcpu=8MHz（内嵌高频振荡器）
  - ✓ 2.4V-5.5V @Fcpu=4MHz（内嵌高频振荡器）
  - ✓ 2.0V-5.5V @Fcpu=2MHz（内嵌高频振荡器）
  - ✓ 2.0V-5.5V @Fcpu=1MHz（内嵌高频振荡器）
  - ✓ 1.8V-5.5V @Fcpu=32768Hz/2（外接低频振荡器）
- ◇ 封装形式：
  - ✓ SKDIP28 (SKY28)/SOP28/SOP24/DIP24/DIP20/SOP20/DIP16/SOP16/DIP24
- ◇ 订购信息：

产品名称	封装形式	其他
MC32P8112A0X	SKDIP28(SKY28)	注：SKDIP28 和 SOP28 脚位 排列不同。
MC32P8112A0G	SOP28	
MC32P8112A0N	SOP24	
MC32P8112A0E	DIP20	
MC32P8112A1C	DIP16	
MC32P8112A1K	SOP16	
MC32P8112A0C	DIP16	注：DIP16 和 SOP16 脚位排 列不同。
MC32P8112A0K	SOP16	
MC32P8112A0M	SOP20	
MC32P8112A0ZE	DIP24	

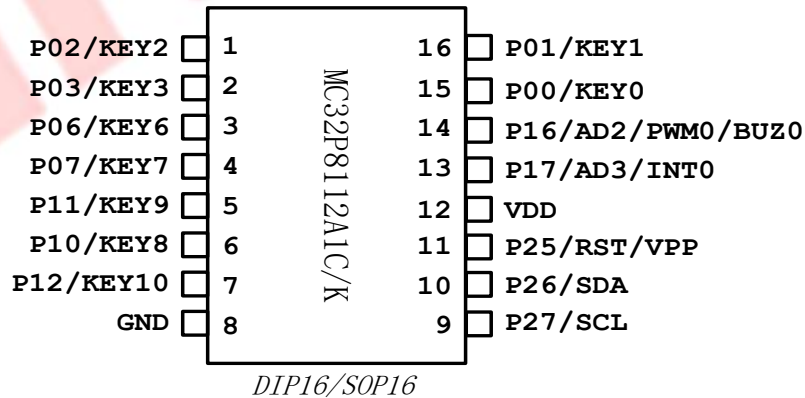
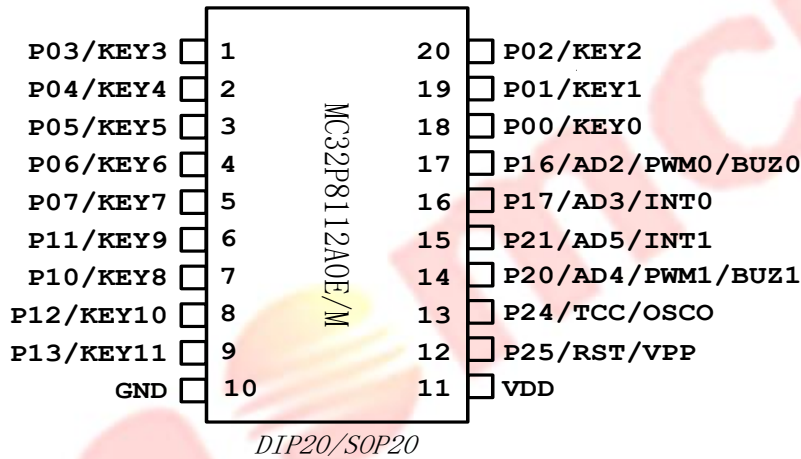
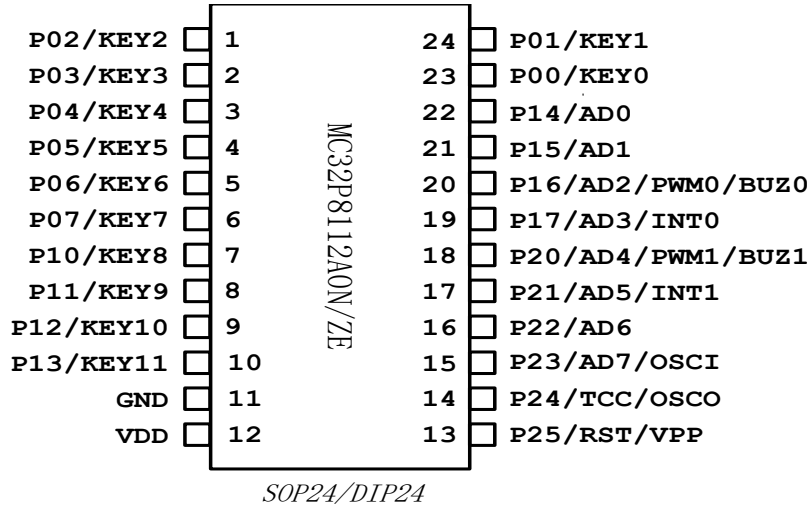
1.2 系统框图



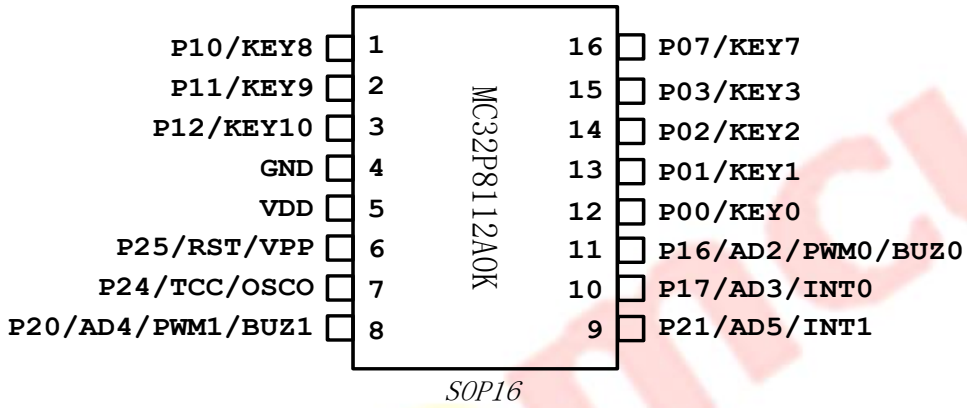
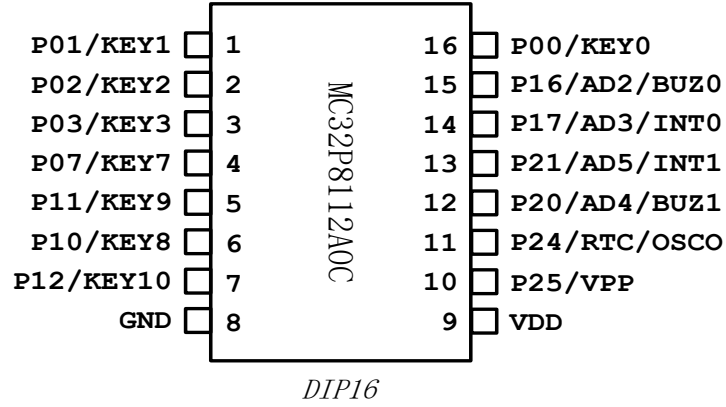
1.3 引脚排列



注：MC32P8112A0X(SKDIP28)脚位与MC32P8112A0G(SOP28)脚位排列不同。







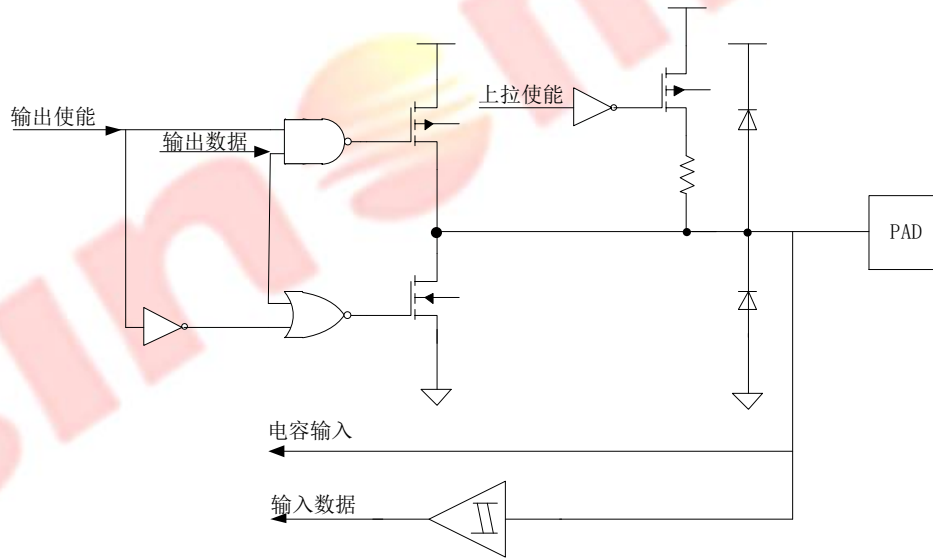
**注：MC32P8112A0C(DIP16)脚位与MC32P8112A0K(SOP16)脚位排列不同。**

### 1.4 引脚说明

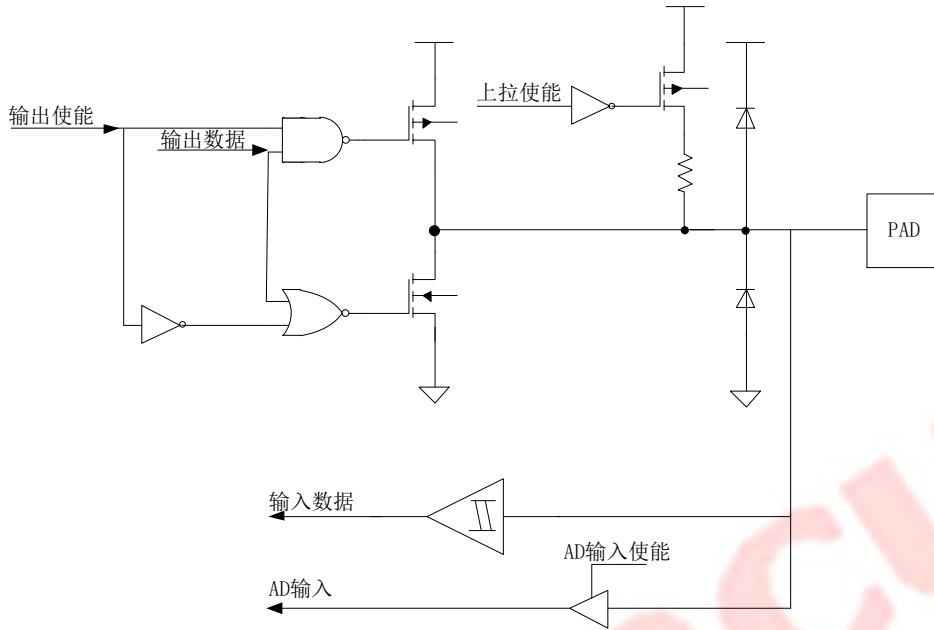
编号	引脚名	方向	功能描述
1	VDD	SOURCE	电源
2	P00-P07	I/O	双向 IO 口,可上拉,触摸按键 KEY0-KEY7
3	P10-P13	I/O	双向 IO 口,可上拉,触摸按键 KEY8-KEY11
4	P14-P17	I/O	双向 IO 口,可上拉, AD 口
5	P20-P22	I/O	双向 IO 口,可上拉, AD 口
6	P23	I/O	双向 IO 口,可上拉, AD 口, 晶振管脚 OSC1
7	P24	I/O	双向 IO 口,可上拉, 晶振管脚 OSC0
8	P25	I/O	双向 IO 口,可上拉, 外部复位脚 RST, 编程高压输入 VPP
9	P26-P27	I/O	双向 IO 口,可上拉, I2C 通讯口
10	P30-P31	I/O	双向 IO 口,可上拉
11	GND	SOURCE	地

### 1.5 管脚结构

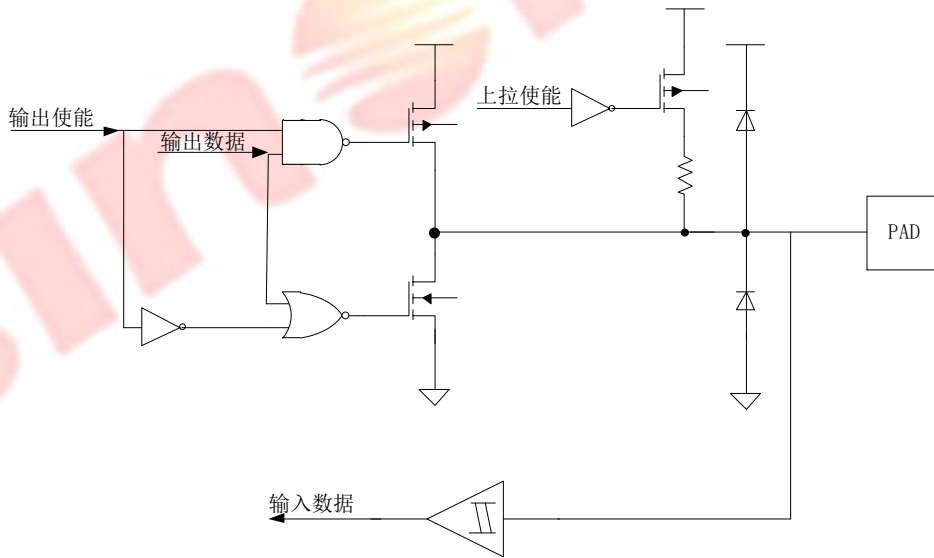
#### P00-P07 和 P10-P13



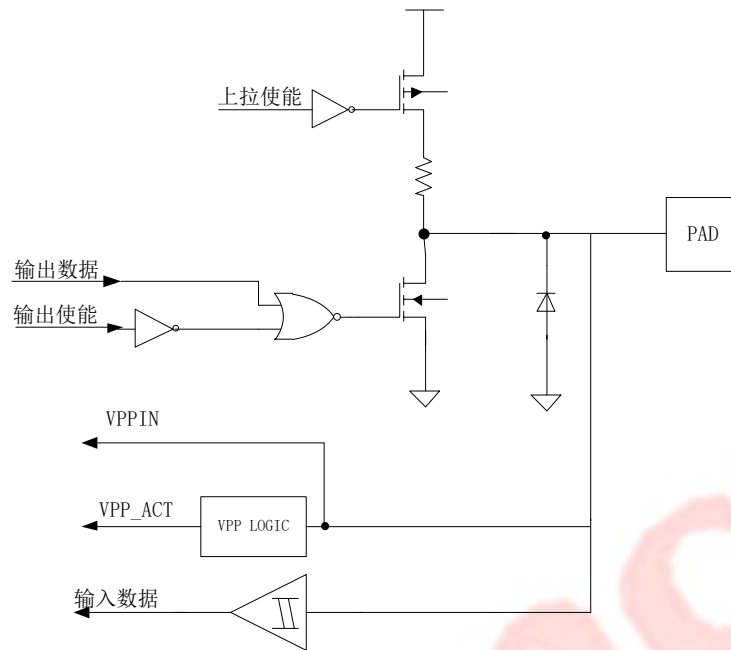
P14-P17 和 P20-P23



P24, P26-P27 和 P30-P31



P25



## 2 中央处理器

### 2.1 指令集

MC32P8112 的指令是精简指令集。下表是指令汇总表。

助记符	说明	操作	周期数	影响
ADDAR R	寄存器 R 内容和 ACC 相加, 结果存到 ACC	R+ACC→ACC	1	C, DC, Z
ADDRA R	寄存器 R 内容和 ACC 相加, 结果存到 R	R+ACC→R	1	C, DC, Z
ADCAR R	带 C 标志的加法, 结果存到 ACC	R+ACC+C→ACC	1	C, DC, Z
ADCRA R	带 C 标志的加法, 结果存到 R	R+ACC+C→R	1	C, DC, Z
RSUBAR R	寄存器 R 内容和 ACC 相减, 结果存到 ACC	R-ACC→ACC	1	C, DC, Z
RSUBRA R	寄存器 R 内容和 ACC 相减, 结果存到 R	R-ACC→R	1	C, DC, Z
RSBCAR R	寄存器 R 内容和 ACC 相减(带 C 标志), 结果存到 ACC	R-ACC-/C→ACC	1	C, DC, Z
RSBCRA R	寄存器 R 内容和 ACC 相减(带 C 标志), 结果存到 R	R-ACC-/C→R	1	C, DC, Z
ASUBAR R	ACC 和寄存器 R 内容相减, 结果存到 ACC	ACC-R→ACC	1	C, DC, Z
ASUBRA R	ACC 和寄存器 R 内容相减, 结果存到 R	ACC-R→R	1	C, DC, Z
ASBCAR R	ACC 和寄存器 R 内容相减(带 C 标志), 结果存到 ACC	ACC-R-/C→ACC	1	C, DC, Z
ASBCRA R	ACC 和寄存器 R 内容相减(带 C 标志), 结果存到 R	ACC-R-/C→R	1	C, DC, Z
ANDAR R	寄存器 R 内容和 ACC 与操作, 结果存到 ACC	R and ACC→ACC	1	Z
ANDRA R	寄存器 R 内容和 ACC 与操作, 结果存到 R	R and ACC→R	1	Z
ORAR R	寄存器 R 内容和 ACC 或操作, 结果存到 ACC	R or ACC→ACC	1	Z
ORRA R	寄存器 R 内容和 ACC 或操作, 结果存到 R	R or ACC→R	1	Z
XORAR R	寄存器 R 内容和 ACC 异或操作, 结果存到 ACC	R xor ACC→ACC	1	Z
XORRA R	寄存器 R 内容和 ACC 异或操作, 结果存到 R	R xor ACC→R	1	Z
COMAR R	对 R 取反, 结果存到 ACC	R 取反→ACC	1	Z
COMR R	对 R 取反, 结果存到 R	R 取反→R	1	Z
CLRA	对 ACC 清零	0→ACC	1	Z
CLRR R	对 R 清零	0→R	1	Z
RLA	ACC 循环左移(带 C 标志)	ACC[7]→C ACC[6:0]→ACC[7:1] C→ACC[0]	1	C
RLAR R	寄存器 R 循环左移(带 C 标志), 结果存到 ACC	R[7]→C R[6:0]→ACC[7:1] C→ACC[0]	1	C
RLR R	寄存器 R 循环左移(带 C 标志), 结果存到 R	R[7]→C R[6:0]→R[7:1] C→R[0]	1	C

RRA	ACC 循环右移(带 C 标志)	C→ACC[7] ACC[7:1]→ACC[6:0] ACC[0]→C	1	C
RRAR R	寄存器 R 循环右移(带 C 标志), 结果存到 ACC	C→ACC[7] R[7:1]→ACC[6:0] R[0]→C	1	C
RRR R	寄存器 R 循环右移(带 C 标志), 结果存到 R	C→R[7] R[7:1]→R[6:0] R[0]→C	1	C
SWAPAR R	交换 R 的高低字节, 结果存到 ACC	R[7:4]→ACC[3:0] R[3:0]→ACC[7:4]	1	-
SWAPR R	交换 R 的高低字节, 结果存到 R	R[7:4]→R[3:0] R[3:0]→R[7:4]	1	-
MOVAR R	将 R 存到 ACC	R→ACC	1	Z
MOVR R	将 R 存到 R	R→R	1	Z
MOVRA R	将 ACC 存到 R	ACC→R	1	-
INCA	ACC 自加 1	ACC+1→ACC	1	-
INCAR R	R 加 1, 结果存到 ACC	R+1→ACC	1	Z
INCR R	R 加 1, 结果存到 R	R+1→R	1	Z
DECA	ACC 自减 1	ACC-1→ACC	1	-
DECAR R	R 减 1, 结果存到 ACC	R-1→ACC	1	Z
DECR R	R 减 1, 结果存到 R	R-1→R	1	Z
JZA	ACC 自加 1; 结果为 0, 则跳过下一条指令	ACC+1→ACC, 结果为 0, 则 PC+2→PC	1 或 2	-
JZAR R	R 加 1, 结果存到 ACC; 结果为 0, 则跳过下一条指令	R+1→ACC, 结果为 0, 则 PC+2→PC	1 或 2	-
JZR R	R 加 1, 结果存到 R; 结果为 0, 则跳过下一条指令	R+1→R, 结果为 0, 则 PC+2→PC	1 或 2	-
DJZA	ACC 自减 1; 结果为 0, 则跳过下一条指令	ACC-1→ACC, 结果为 0, 则 PC+2→PC	1 或 2	-
DJZAR R	R 减 1, 结果存到 ACC; 结果为 0, 则跳过下一条指令	R-1→ACC, 结果为 0, 则 PC+2→PC	1 或 2	-
DJZR R	R 减 1, 结果存到 R; 结果为 0, 则跳过下一条指令	R-1→R, 结果为 0, 则 PC+2→PC	1 或 2	-
BCLR R,b	对 R 的第 b 位清零	0→R[b]	1	-
BSET R,b	对 R 的第 b 位置 1	1→R[b]	1	-
JBCLR R,b	如 R 的第 b 位为 0, 则跳过下一条指令	如 R[b]=0, 则 PC+2→PC	1 或 2	-
JBSET R,b	如 R 的第 b 位为 1, 则跳过下一条指令	如 R[b]=1, 则 PC+2→PC	1 或 2	-
ADDAI K	立即数 K 和 ACC 相加, 结果存到 ACC	K+ACC→ACC	1	C, DC, Z
ISUBAI K	立即数和 ACC 相减, 结果存到 ACC	K-ACC→ACC	1	C, DC, Z
ISBCAI K	立即数和 ACC 相减(带 C 标志), 结果存到 ACC	K-ACC-/C→ACC	1	C, DC, Z

ASUBAI K	ACC 和立即数相减, 结果存到 ACC	ACC-K→ACC	1	C, DC, Z
ASBCAI K	ACC 和立即数相减(带 C 标志), 结果存到 ACC	ACC-K-/C→ACC	1	C, DC, Z
ANDAI K	立即数 K 和 ACC 与操作, 结果存到 ACC	K and ACC→ACC	1	Z
ORAI K	立即数 K 和 ACC 或操作, 结果存到 ACC	K or ACC→ACC	1	Z
XORAI K	立即数和 ACC 异或, 结果存到 ACC	K xor ACC→ACC	1	Z
MOVAI K	将立即数存到 ACC	K→ACC	1	-
RETAI K	从子程序返回, 并将立即数存到 ACC	TOS→PC K→ACC	2	-
RETURN	从子程序返回	TOS→PC	2	-
RETIE	从中断返回	TOS→PC 1→GIE	2	-
CALL K	子程序调用	PC+1→TOS K→PC	2	-
GOTO K	无条件跳转	K→PC	2	-
NOP	空操作	空操作	1	-
DAA	加法后, 将 ACC 的值调整到十进制	ACC(十六进制)→ ACC(十进制)	1	C
DSA	减法后, 将 ACC 的值调整到十进制	ACC(十六进制)→ ACC(十进制)	1	-
CLRWDT	清看门狗定时器	0→WDT	1	TO, PD
STOP	进入休眠模式	0→WDT 进入休眠模式	1	TO, PD

## 2.2 程序存储器

4K\*16 的程序存储器空间，程序存储器空间（0000H - 0FFFH）可通过 INDF3 间接访问

复位向量 (0000H)
通用程序区 (0001H - 0007H)
中断向量 (0008H)
通用程序区 (0009H - 0FFFH)
厂商保留区 (1000H - 7FFFH)
OPBIT0 (8000H)
OPBIT1 (8001H)
OPBIT2 (8002H)

例：通过 INDF3 访问  $FSR1*256+FSR0$  指向的程序存储器中内容，高 8 位存放在数据寄存器区 11H，低 8 位存放在数据寄存器区 10H

```

MOVAI      55H
MOVRA     FSR0      ; 将 55H 写入 FSR0
MOVAI     01H
MOVRA     FSR1      ; 将 01H 写入 FSR1
MOVAR     INDF3     ; 读取 FSR1*256+FSR0 指向 (0155H) 程序存储器
                ; 的内容，其中高 8 位放在 HIBYTE 寄存器，低 8
                ; 位放在 A 寄存器
MOVRA     10H       ; 低 8 位放到数据寄存器 10H 地址
MOVAR     HIBYTE    ; 从 HIBYTE 读取高 8 位
MOVRA     11H       ; 高 8 位放到数据寄存器 11H 地址
    
```



## 2.3 数据存储器

数据寄存器分为两个区，快速通用寄存器区GPR（320Byte空间）和特殊功能寄存器区SFR，具体地址分配参照下表。

**数据存储器区地址映射表：**

地址	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
数据存储器区，直接寻址， INDF0, INDF1, INDF2间接寻址								
000H – 0FFH	通用数据区(INDF0/INDF2)							
100H – 13FH	通用数据区(INDF1/INDF2)							
140H – 1AFH								
1B0H – 1B7H	INDF0	INDF1	INDF2	HIBYTE	FSR0	FSR1	PCL	PFLAG
1B8H – 1BFH	MCR	INDF3	INTE	INTF	OSCM			
1C0H – 1C7H	IOP0	OEP0	PUP0		IOP1	OEP1	PUP1	
1C8H – 1CFH	IOP2	OEP2	PUP2		IOP3	OEP3	PUP3	
1D0H – 1D7H	T0CR	T0CNT	T0LOAD	T0DATA	T1CR	T1CNT	T1LOAD	T1DATA
1D8H – 1DFH	T2CR	T2CNTH	T2CNTL	T2LOADH	T2LOADL			
1E0H – 1E7H	TK0CRH	TK0CRL	TK0CNTH	TK0CNTL	TK1CRH	TK1CRL	TK1CNTH	TK1CNTL
1E8H – 1EFH	TK2CRH	TK2CRL	TK2CNTH	TK2CNTL	ADCR0	ADCR1	ADRH	ADRL
1F0H – 1F8H	I2CCR	I2CADDR	I2CDATA					
1F8H – 1FFH				OSCCAL	保留			

注 1：上表中灰色部分数据存储器区地址未用，读出数据为 0

注 2：上表中“保留”指该寄存器有某些特殊功能，用户不能随意更改

**数据寄存器地址组成**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	寻址方式	
0	0	0	0	0	0	0	来自指令的 9 位地址									直接寻址模式	
0	0	0	0	0	0	0	0	FSR0									间接寻址模式 0
0	0	0	0	0	0	0	1	FSR1									间接寻址模式 1
FSR1								FSR0									间接寻址模式 2

直接寻址模式：以指令的低9位作为数据存储器地址

例：通过直接寻址模式把 55H 数据写入 10H 地址

```
MOVAI      55H
MOVRA     10H ; 把数据 55H 写入 10H 地址数据存储器中
```

间接寻址模式0：当访问INDF0时，FSR0作为数据存储器地址

例：通过间接寻址模式 0 把 55H 数据写入 10H 地址

MOVAI	10H
MOVRA	FSR0
MOVAI	55H
MOVRA	INDF0 ; 把数据 55H 写入 FSR0 指向数据存储器中

间接寻址模式1：当访问INDF1时，FSR1作为数据存储器地址

例：通过间接寻址模式 1 把 55H 数据写入 110H 地址

MOVAI	10H
MOVRA	FSR1
MOVAI	55H
MOVRA	INDF1 ; 把数据 55H 写入 FSR1 指向数据存储器中

间接寻址模式2：当访问INDF2时，FSR1\*256+FSR0作为数据存储器地址

例：通过间接寻址模式 2 把 55H 数据写入 0110H 地址数据存储器

MOVAI	10H
MOVRA	FSR0
MOVAI	01H
MOVRA	FSR1
MOVAI	55H
MOVRA	INDF2 ; 把数据 55H 写入 FSR1*256+FSR0 指向数据存储器中

## 2.4 堆栈

8级堆栈深度，当程序响应中断或执行子程序调用指令时CPU会将PC自动压栈；当运行子程序返回指令时，栈顶数据赋予PC。

## 2.5 烧录配置选项

用户配置字简称OPBIT是OTP中的3个特殊字，用于对系统功能进行配置。OPBIT在烧写用户程序时通过专用烧写器来设置。MC32P8112的OPBIT定义如下。

### OPBIT0:

位	符号	功能说明
<b>BIT[1:0]</b>	WDTC	WDT 工作模式控制位 00: 始终关闭看门狗 01: 休眠模式下关闭看门狗 1X: 始终开启看门狗
<b>BIT13, BIT[3:2]</b>	WDTT	WDT 溢出时间选择位 001: 上电延时=WDT 溢出时间=16Ms 010: 上电延时=WDT 溢出时间=64Ms 011: 上电延时=WDT 溢出时间=256Ms 101: 上电延时=16Ms, WDT 溢出时间=1024Ms 110: 上电延时=64Ms, WDT 溢出时间=2048Ms 111: 上电延时=256Ms, WDT 溢出时间=4096Ms
<b>BIT[6:4]</b>	FCPU	高速模式 Fcpu 速度选择 000: 机器周期 Fcpu 为 2 个高速时钟周期 Fhosc/Fhirc 001: 机器周期 Fcpu 为 4 个高速时钟周期 Fhosc/Fhirc 010: 机器周期 Fcpu 为 8 个高速时钟周期 Fhosc/Fhirc 011: 机器周期 Fcpu 为 16 个高速时钟周期 Fhosc/Fhirc 100: 机器周期 Fcpu 为 32 个高速时钟周期 Fhosc/Fhirc 101: 机器周期 Fcpu 为 64 个高速时钟周期 Fhosc/Fhirc 110: 机器周期 Fcpu 为 128 个高速时钟周期 Fhosc/Fhirc 111: 机器周期 Fcpu 为 256 个高速时钟周期 Fhosc/Fhirc
<b>BIT[7]</b>	MCLRE	外部复位使能位 0: 不使能外部复位, P25 作为 IO 1: 使能外部复位, P25 作为复位引脚
<b>BIT[9:8]</b>	FOSC	时钟选择位 00: 内部高频+内部低频 01: 内部高频+外部低频 1x: 外部高频+内部低频
<b>BIT[12:10]</b>	VLVRS	系统复位电压选择位 001: LVR 电压=1.8V 010: LVR 电压=2.0V 011: LVR 电压=2.2V 100: LVR 电压=2.4V 101: LVR 电压=2.6V 110: LVR 电压=3.0V 111: LVR 电压=3.6V
<b>BIT[14]</b>	未用	-

<b>BIT[15]</b>	ENCR	代码加密选项 0: 使能代码加密 1: 不使能代码加密
----------------	------	-----------------------------------

**OPBIT1:**

位	符号	功能说明
<b>BIT[7:0]</b>	OSCCAL	内部 16M 高频振荡器频率校准位, 产品出厂前厂家已写入
<b>BIT[9:8]</b>	VLDO	触摸按键模块工作电压选择位 00: 2.1V 01: 2.4V 10: 2.7V 11: 3.0V
<b>BIT[12:10]</b>	保留	-
<b>BIT[15:13]</b>	未用	-

**OPBIT2:**

位	符号	功能说明
<b>BIT[15:0]</b>	LOCK	代码块读加密使能位: 每位可控制一个块(256 个字), BIT0 控制地址 0000H-00FFH, 以此类推; 当通过 INDF3 读程序区时, 如果 INDF3 指向的代码块读加密位被置为“0”, 则当前 PC 地址和 INDF3 指向地址处于同一块时可以读取; 如果当前 PC 地址和 INDF3 指向地址不处于同一块时不能读取数据, 读回的数据为零

注: “保留”指该位有某些特殊功能, 用户不能随意更改; “未用”指该位无功能

## 2.6 控制寄存器

MC32P8112全部控制寄存器列在下表中，具体功能详见各功能模块的说明。

地址	助记符	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	初始值
1B0H	INDF0	INDF07	INDF06	INDF05	INDF04	INDF03	INDF02	INDF01	INDF00	XXXX XXXX
1B1H	INDF1	INDF17	INDF16	INDF15	INDF14	INDF13	INDF12	INDF11	INDF10	XXXX XXXX
1B2H	INDF2	INDF27	INDF26	INDF25	INDF24	INDF23	INDF22	INDF21	INDF20	XXXX XXXX
1B3H	HIBYTE	HIBYTE7	HIBYTE6	HIBYTE5	HIBYTE4	HIBYTE3	HIBYTE2	HIBYTE1	HIBYTE0	XXXX XXXX
1B4H	FSR0	FSR07	FSR06	FSR05	FSR04	FSR03	FSR02	FSR01	FSR00	XXXX XXXX
1B5H	FSR1	FSR17	FSR16	FSR15	FSR14	FSR13	FSR12	FSR11	FSR10	XXXX XXXX
1B6H	PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0000 0000
1B7H	PFLAG	-	-	-	-	-	Z	DC	C	---- -XXX
1B8H	MCR	GIE	-	TO	PD	MINT11	MINT10	MINT01	MINT00	0-00 0000
1B9H	INDF3	INDF37	INDF36	INDF35	INDF34	INDF33	INDF32	INDF31	INDF30	XXXX XXXX
1BAH	INTE	-	ADIE	I2CIE	T2IE	INT1IE	INT0IE	T1IE	TOIE	-000 0000
1BBH	INTF	-	ADIF	I2CIF	T2IF	INT1IF	INT0IF	T1IF	TOIF	-000 XX00
1BCH	OSCM	-	-	STBL	STBH	-	CLKS	LFEN	HFEN	--X1 -000
1C0H	IOP0	P07D	P06D	P05D	P04D	P03D	P02D	P01D	P00D	XXXX XXXX
1C1H	OEP0	P07OE	P06OE	P05OE	P04OE	P03OE	P02OE	P01OE	P00OE	0000 0000
1C2H	PUP0	P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU	0000 0000
1C4H	IOP1	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D	XXXX XXXX
1C5H	OEP1	P17OE	P16OE	P15OE	P14OE	P13OE	P12OE	P11OE	P10OE	0000 0000
1C6H	PUP1	P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU	0000 0000
1C8H	IOP2	P27D	P26D	P25D	P24D	P23D	P22D	P21D	P20D	XXXX XXXX
1C9H	OEP2	P27OE	P26OE	P25OE	P24OE	P23OE	P22OE	P21OE	P20OE	0000 0000
1CAH	PUP2	P27PU	P26PU	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU	0000 0000
1CCH	IOP3	-	-	-	-	-	-	P31D	P30D	---- --XX
1CDH	OEP3	-	-	-	-	-	-	P31OE	P30OE	---- --00
1CEH	PUP3	-	-	-	-	-	-	P31PU	P30PU	---- --00
1D0H	T0CR	TC0EN	PWM0OE	BUZ0OE	T0PS1	T0PS0	T0PR2	T0PR1	T0PR0	0000 0000
1D1H	T0CNT	T0C7	T0C6	T0C5	T0C4	T0C3	T0C2	T0C1	T0C0	1111 1111
1D2H	T0LOAD	T0LOAD7	T0LOAD6	T0LOAD5	T0LOAD4	T0LOAD3	T0LOAD2	T0LOAD1	T0LOAD0	1111 1111
1D3H	T0DATA	T0DATA7	T0DATA6	T0DATA5	T0DATA4	T0DATA3	T0DATA2	T0DATA1	T0DATA0	0000 0000
1D4H	T1CR	TC1EN	PWM1OE	BUZ1OE	T1PS1	T1PS0	T1PR2	T1PR1	T1PR0	0000 0000
1D5H	T1CNT	T1C7	T1C6	T1C5	T1C4	T1C3	T1C2	T1C1	T1C0	1111 1111
1D6H	T1LOAD	T1LOAD7	T1LOAD6	T1LOAD5	T1LOAD4	T1LOAD3	T1LOAD2	T1LOAD1	T1LOAD0	1111 1111
1D7H	T1DATA	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0	0000 0000
1D8H	T2CR	TC2EN	-	T2MODE	T2PS1	T2PS0	T2PR2	T2PR1	T2PR0	00-0 0000
1D9H	T2CNTH	T2C15	T2C14	T2C13	T2C12	T2C11	T2C10	T2C9	T2C8	1111 1111
1DAH	T2CNTL	T2C7	T2C6	T2C5	T2C4	T2C3	T2C2	T2C1	T2C0	1111 1111
1DBH	T2LOADH	T2LOAD15	T2LOAD14	T2LOAD13	T2LOAD12	T2LOAD11	T2LOAD10	T2LOAD9	T2LOAD8	1111 1111
1CCH	T2LOADL	T2LOAD7	T2LOAD6	T2LOAD5	T2LOAD4	T2LOAD3	T2LOAD2	T2LOAD1	T2LOAD0	1111 1111
1E0H	TK0CRH	TK0K1	TK0K0	TK0JE	TKORCE	TK0CLR	TK0FQ2	TK0FQ1	TK0FQ0	0000 0000
1E1H	TK0CRL	TK0IF	TK0IE	TK0FL1	TK0FL0	TK0IO3	TK0IO2	TK0IO1	TK0IO0	0000 0000
1E2H	TK0CNTH	TK0C15	TK0C14	TK0C13	TK0C12	TK0C11	TK0C10	TK0C9	TK0C8	0000 0000
1E3H	TK0CNTL	TK0C7	TK0C6	TK0C5	TK0C4	TK0C3	TK0C2	TK0C1	TK0C0	0000 0000
1E4H	TK1CRH	TK1K1	TK1K0	TK1JE	TK1RCE	TK1CLR	TK1FQ2	TK1FQ1	TK1FQ0	0000 0000
1E5H	TK1CRL	TK1IF	TK1IE	TK1FL1	TK1FL0	TK1IO3	TK1IO2	TK1IO1	TK1IO0	0000 0000
1E6H	TK1CNTH	TK1C15	TK1C14	TK1C13	TK1C12	TK1C11	TK1C10	TK1C9	TK1C8	0000 0000
1E7H	TK1CNTL	TK1C7	TK1C6	TK1C5	TK1C4	TK1C3	TK1C2	TK1C1	TK1C0	0000 0000

1E8H	TK2CRH	TK2K1	TK2K0	TK2JE	TK2RCE	TK2CLR	TK2FQ2	TK2FQ1	TK2FQ0	0000 0000
1E9H	TK2CRL	TK2IF	TK2IE	TK2FL1	TK2FL0	TK2IO3	TK2IO2	TK2IO1	TK2IO0	0000 0000
1EAH	TK2CNTH	TK2C15	TK2C14	TK2C13	TK2C12	TK2C11	TK2C10	TK2C9	TK2C8	0000 0000
1EBH	TK2CNTL	TK2C7	TK2C6	TK2C5	TK2C4	TK2C3	TK2C2	TK2C1	TK2C0	0000 0000
1ECH	ADCR0	ADCHS3	ADCHS2	ADCHS1	ADCHS0	ADCKS1	ADCKS0	ADEOC	ADON	0000 0000
1EDH	ADCR1	ADIOS7	ADIOS6	ADIOS5	ADIOS4	ADIOS3	ADIOS2	ADIOS1	ADIOS0	0000 0000
1EEH	ADRH	ADR11	ADR10	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	XXXX XXXX
1EFH	ADRL	-	-	-	-	ADR3	ADR2	ADR1	ADR0	---- XXXX
1F0H	I2CCR	HCF	HAAS	HBB	HTX	TXAK	SRW	HEN	RXAK	0000 1000
1F1H	I2CADDR	I2CAD7	I2CAD6	I2CAD5	I2CAD4	I2CAD3	I2CAD2	I2CAD1	I2CAD0	0000 0000
1F2H	I2CDATA	I2CDA7	I2CDA6	I2CDA5	I2CDA4	I2CDA3	I2CDA2	I2CDA1	I2CDA0	0000 0000
1FBH	OSCCAL	OSCCAL7	OSCCAL6	OSCCAL5	OSCCAL4	OSCCAL3	OSCCAL2	OSCCAL1	OSCCAL0	**** ****

注：-表示未定义；X表示不确定；\*表示自动加载默认值，详见内置高频 RC 振荡器章节

### 间接寻址寄存器0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INDF0</b>	INDF07	INDF06	INDF05	INDF04	INDF03	INDF02	INDF01	INDF00
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **INDF0[7:0]** – 间接寻址寄存器 0

INDF0: INDF0 不是物理寄存器，对 INDF0 寻址时间上是对 FSR0 指向的数据存储器地址进行访问，从而实现间接寻址模式。

### 间接寻址寄存器1

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INDF1</b>	INDF17	INDF16	INDF15	INDF14	INDF13	INDF12	INDF11	INDF10
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **INDF1[7:0]** – 间接寻址寄存器 1

INDF1: INDF1 不是物理寄存器，对 INDF1 的寻址时间上是对 FSR1+256 指向的数据存储器地址进行访问，从而实现间接寻址模式。

### 间接寻址寄存器2

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INDF2</b>	INDF27	INDF26	INDF25	INDF24	INDF23	INDF22	INDF21	INDF20
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **INDF2[7:0]** – 间接寻址寄存器 2

INDF2: INDF2 不是物理寄存器，对 INDF2 的寻址时间上是对 FSR1\*256+FSR0 指向的数据存储器地址进行访问，从而实现间接寻址模式。

### 间接寻址寄存器3

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INDF3</b>	INDF37	INDF36	INDF35	INDF34	INDF33	INDF32	INDF31	INDF30
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

**BIT[7:0] INDF3[7:0]** – 间接寻址寄存器 3

INDF3: INDF3 不是物理寄存器, 对 INDF3 的寻址时间上是对 FSR1\*256+FSR0 指向的程序存储器地址进行访问, 从而实现间接寻址模式。

注: 对 INDF3 仅可进行使用读取指令(MOVAR INDF3)进行读取访问, 读取内容高 8 位存放在 HIBYTE, 低 8 位存放在 A 寄存器

### 字操作高8位缓冲器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>HIBYTE</b>	HIBYTE7	HIBYTE6	HIBYTE5	HIBYTE4	HIBYTE3	HIBYTE2	HIBYTE1	HIBYTE0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

**BIT[7:0] HIBYTE[7:0]** – 字操作高字节缓冲器

HIBYTE: 对 INDF3 读取操作, 用于存放 FSR1\*256+FSR0 指向的程序存储器内容高 8 位数据。

### 数据指针寄存器0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>FSR0</b>	FSR07	FSR06	FSR05	FSR04	FSR03	FSR02	FSR01	FSR00
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

**BIT[7:0] FSR0[7:0]** – 数据指针寄存器 0

FSR0: 间接寻址模式 0 指针或间接寻址模式 2、3 指针低 8 位。

### 数据指针寄存器1

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>FSR1</b>	FSR17	FSR16	FSR15	FSR14	FSR13	FSR12	FSR11	FSR10
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

**BIT[7:0] FSR1[7:0]** – 数据指针寄存器 1

FSR1: 间接寻址模式 1 指针或间接寻址模式 2、3 指针高位。

**程序指针计数器低位**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PCL</b>	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7:0]** **PC[7:0]** – 程序指针计数器低 8 位

程序指针计数器 (PC) 有以下几种操作模式

顺序运行指令:  $C=PC+1$

分支指令GOTO/CALL:  $PC=$ 指令码低10位

子程序返回指令RETIE/RETURN/RETAI:  $PC=$ 堆栈栈顶

对PCL操作指令:  $PC = (PC[15:0]+A[7:0])$  (对PCL操作的加法指令)

$PC = \{PC[15:8], ALU[7:0]\}$  (ALU运算结果) (对PCL操作的其它指令)



**CPU状态寄存器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PFLAG</b>	-	-	-	-	-	Z	DC	C
<b>R/W</b>	-	-	-	-	-	R/W	R/W	R/W
<b>初始值</b>	-	-	-	-	-	X	X	X

**BIT[7:3]** 未用

**BIT[2]** **Z** – 零标志

0: 算术或逻辑运算的结果不为零

1: 算术或逻辑运算的结果为零

**BIT[1]** **DC** – 半进位标志

0: 加法运算时低四位没有进位/减法运算时有向高四位借位

1: 加法运算时低四位有进位/减法运算时没有向高四位借位

**BIT[0]** **C** – 进位标志

0: 加法运算时没有进位/减法运算时有借位发生/移位后移出逻辑0

1: 加法运算时有进位/减法运算时没有借位发生/移位后移出逻辑1

**杂用寄存器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>MCR</b>	GIE	-	TO	PD	MINT11	MINT10	MINT01	MINT00
<b>R/W</b>	R/W	-	R	R	R/W	R/W	R/W	R/W
<b>初始值</b>	0	-	0	0	0	0	0	0

**BIT[7]** **GIE** – 总中断使能

0: 屏蔽所有中断

1: 中断源是否产生中断由相应的控制位决定

**BIT[6]** 未用

**BIT[5]** **TO** – 看门狗溢出标志

0: 上电复位, 执行CLRWDT或STOP指令

1: 发生WDT溢出

**BIT[4]** **PD** – 进入低功耗休眠模式标志

0: 上电复位, 执行CLRWDT

1: 执行STOP指令

**BIT[3:2]** **MINT1** – 外部中断 1 模式寄存器

00: INT1 上升沿触发

01: INT1 下降沿触发

1x: INT1 电平变化触发

**BIT[1:0]** **MINT0** – 外部中断 0 模式寄存器

00: INT0 上升沿触发

01: INT0 下降沿触发

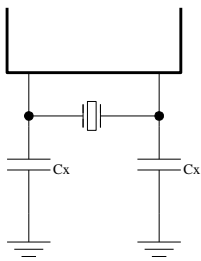
1x: INT0 电平变化触发

### 3 系统时钟

MC32P8112 为双时钟系统，可根据需要通过软件在高速时钟和低速时钟之间任意切换。高速时钟可选择外接高频振荡器或内置 16MHz 的高精度 RC 振荡器；低速时钟可选用外接 32768Hz 低频振荡器或内置低频 RC 振荡器。

系统选用高速时钟时 CPU 的机器周期  $F_{cpu}$  由 OPBIT 的 FCPU 配置，选用低速时钟时 CPU 的机器周期  $F_{cpu}$  为 2 个低频振荡周期  $F_{osc}/F_{lirc}$ 。

#### 3.1 外接晶体振荡器



外部晶体有外接高频和外接低频两种振荡工作模式，连接方式见左图。高频晶体可选用 432KHz~16MHz，低频一般是接 32768Hz 晶体，通常  $C_x$  是必须的。在实际使用中，用户应使晶体离 OSCI、OSCO 引脚的距离尽可能短，这样有助于振荡器的起振和振荡的稳定性。

下表列出几种典型频率晶振选用电容  $C_x$  的推荐值和相应最低起振电压参考值。

晶体频率	电容 $C_x$ (F)	最低起振电压 (V)
16MHz	10p	2.7
8MHz	15p	2.4
4MHz	15p/30p	2.2
455KHz	220p/470p	2.0
32768Hz	10p~30p	1.8

注：因为晶体的品牌很多，电容值仅为推荐值，起振电压仅供参考，具体参数请根据实际使用的晶振性能而定。

### 3.2 内置高频 RC 振荡器

MC32P8112 的内置高精度 16MHz RC 振荡器，该振荡器可用于系统高速时钟。

#### 内部高频振荡器频率校准寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>OSCCAL</b>	OSCCAL7	OSCCAL6	OSCCAL5	OSCCAL4	OSCCAL3	OSCCAL2	OSCCAL1	OSCCAL0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	*	*	*	*	*	*	*	*

**BIT[7:0] OSCCAL[7:0]** – 内部高频 RC 振荡器频率校准寄存器

芯片复位后，OSCCAL 自动加载出厂默认值，该默认值将内部高频 RC 振荡器频率调整到 16MHz，该寄存器允许用户通过程序进行修改，以满足客户的其它频率要求。

例：

INCR                                      OSCCAL                                      ； 频率在 16M 基础上增大 1 个步长  
 DECR                                        OSCCAL                                      ； 频率在 16M 基础上减小 1 个步长

注：步长是非线性的，设计值为 70KHz，最大调节范围在 10MHz~20MHz（以实际芯片为准）。

### 3.3 内置低频 RC 振荡器

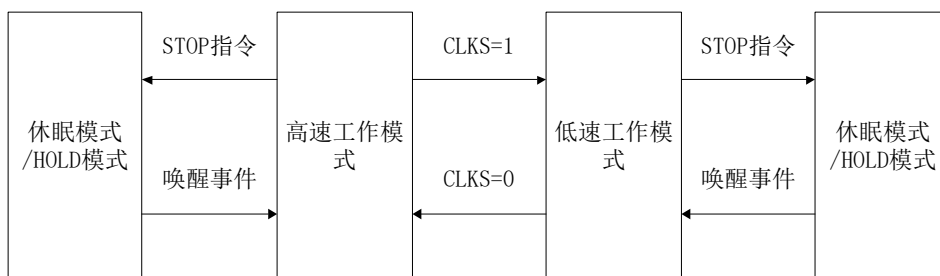
MC32P8112 的内置一个低频 RC 振荡器，该振荡器可用于系统低频时钟，同时用于上电延时、WDT。该振荡器频率典型值 28KHz。

### 3.4 工作模式

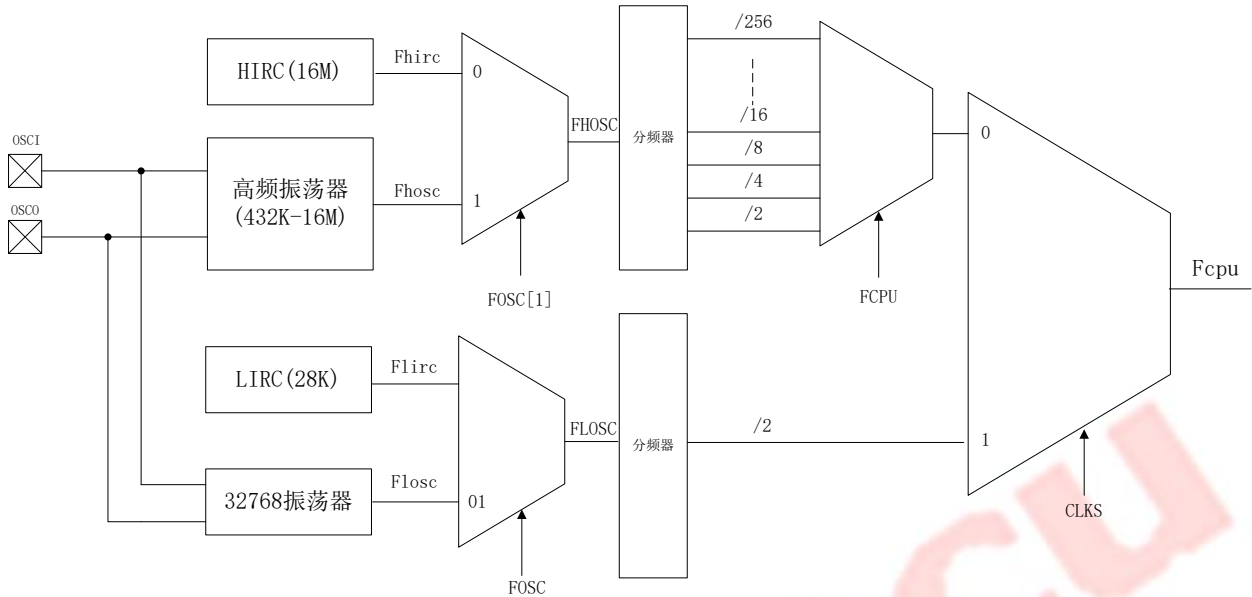
MC32P8112 支持高速工作模式、低速工作模式、休眠模式、HOLD 模式 1 和 HOLD 模式 2 共 5 种工作模式。

工作模式	进入条件
高速工作模式	系统时钟切换到高频振荡器 (CLKS=0)
低速工作模式	系统时钟切换到低频振荡器 (CLKS=1)
休眠模式	执行 STOP 指令, HFEN=0, LFEN=0
HOLD 模式 1	执行 STOP 指令, HFEN=1, LFEN=X (定时器可在高频时钟模式下继续工作, 溢出可唤醒)
HOLD 模式 2	执行 STOP 指令, HFEN=0, LFEN=1 (定时器可在低频时钟模式下继续工作, 溢出可唤醒)

#### 工作模式间的切换



### 系统时钟选择



	高速工作模式	低速工作模式	休眠模式/HOLD 模式
高频振荡器	工作	HFEN 决定	HFEN 决定
低频振荡器	工作	工作	LFEN 决定
WDT 振荡器	工作	工作	WDTC 决定

注：当低频时钟选择为内部低频 RC 振荡器，则低频振荡器和 WDT 振荡器共用同一振荡器，低频振荡器工作或 WDT 振荡器工作都会使内部低频 RC 振荡器工作。

### 工作模式寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>OSCM</b>	-	-	STBL	STBH	-	CLKS	LFEN	HFEN
<b>R/W</b>	-	-	R	R	-	R/W	R/W	R/W
<b>初始值</b>	-	-	X	1	-	0	0	0

BIT[7:6] 未用

BIT[5] **STBL** – 低频振荡器稳定标志  
0: 低频振荡器停振或未稳定  
1: 低频振荡器已稳定运行

BIT[4] **STBH** – 高频振荡器稳定标志  
0: 高频振荡器停振或未稳定  
1: 高频振荡器已稳定运行

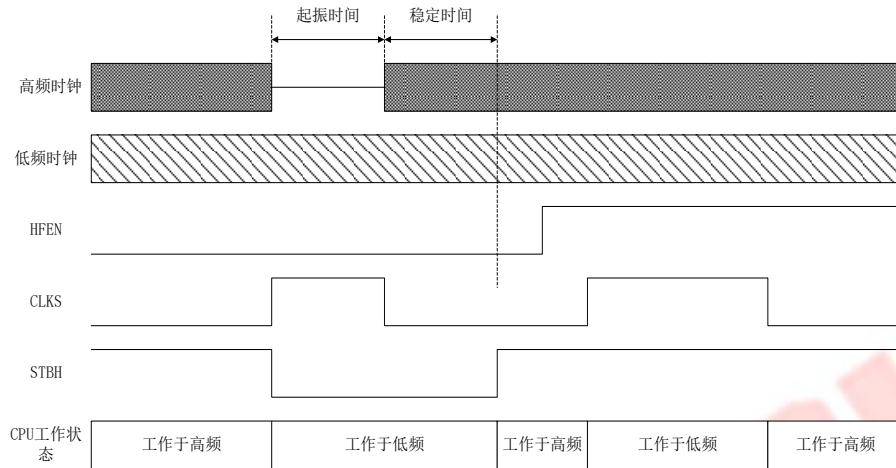
BIT[3] 未用

BIT[2] **CLKS** – 系统工作时钟选择位  
0: 高频时钟作为系统时钟  
1: 低频时钟作为系统时钟

BIT[1] **LFEN** – 低频振荡器使能  
0: 在休眠/HOLD模式下，低频振荡器停止工作  
1: 低频振荡器始终工作

- BIT[0] HFEN** – 高频振荡器使能
- 0: 在低速/休眠/HOLD模式下，高频振荡器停止工作
  - 1: 高频振荡器始终工作

### 高低频时钟切换时序图



### 3.5 低功耗模式

低功耗模式包括休眠模式、HOLD 模式 1、HOLD 模式 2。

STOP 指令可使 MCU 进入低功耗模式，同时对 MCU 会产生以下影响

- ✧ 根据不同模式停止相应振荡器振荡
- ✧ RAM 内容保持不变
- ✧ 所有的输入输出端口保持不变
- ✧ 定时器 0、定时器 1 和定时器 2 根据其工作模式，可以保持继续工作

以下情况可使 MCU 退出低功耗模式

- ✧ 有外部中断请求发生
- ✧ 定时器 0、定时器 1 和定时器 2 计数溢出中断请求发生
- ✧ 有 WDT 溢出
- ✧ I2C 地址匹配中断
- ✧ 复位

MCU 退出低功耗模式后，经过振荡等待（外部高/低频晶振等待 1024 个周期，内部高频等待 32 个周期，内部低频等待 16 个周期）然后开始工作。

## 4 复位

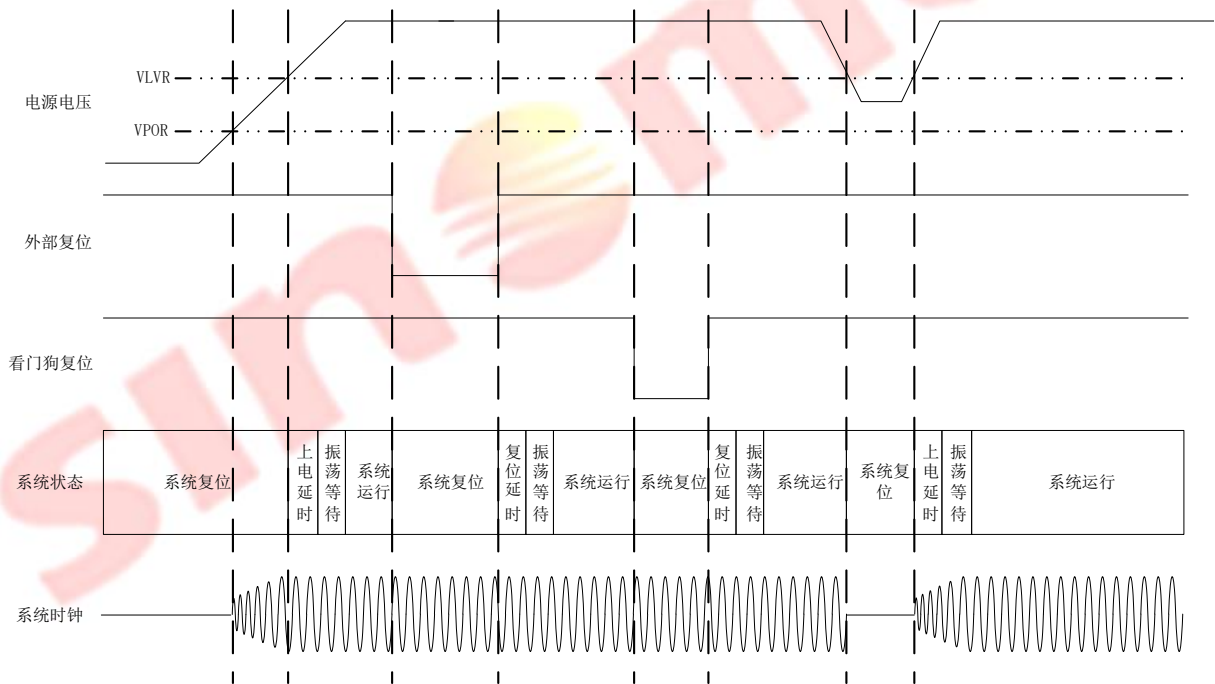
### 4.1 复位条件

MC32P8112 有四种复位方式

- ◇ 上电复位 POR
- ◇ 外部复位
- ◇ 低电压复位 LVR
- ◇ 看门狗复位

任何一种复位发生时，系统将会重新从 0000H 地址处开始执行指令；另外系统还会将所有的特殊功能寄存器重置为默认初始值。

上电复位和低电压复位会关闭系统主时钟的振荡器，复位解除后才重新打开振荡器，系统会在上电延时后经过振荡等待（外部高/低频晶振等待 1024 个周期，内部高频等待 32 个周期，内部低频等待 16 个周期）然后开始工作。外部复位和看门狗复位不会关闭系统主时钟振荡器，系统会在复位延时（约 3ms）后经过振荡等待（外部高/低频晶振等待 1024 个周期，内部高频等待 32 个周期，内部低频等待 16 个周期）然后开始工作。下图是复位产生和系统工作状态之间的关系示意图。



## 4.2 上电复位

MC32P8112 的上电复位电路可以适应快速、慢速上电的情况，并且当芯片上电过程中出现电源电压抖动时都能保证系统可靠的复位。

上电复位过程系统工作步骤：

- ◇ 等待电压高于 VPOR 并保持稳定
- ◇ 开启内部低频振荡器加载 OPBIT，并根据 OPBIT 设置开启相应振荡器和进行相应时间的上电延时
- ◇ 如果外部复位功能开启，则需等待复位引脚电压高于 VIH 才开始上电延时计数
- ◇ 等待电压高于 VLVR 才开始上电延时计数
- ◇ 上电延时结束，振荡等待结束后系统开始执行指令

## 4.3 外部复位

外部复位功能是否开启可以通过 OPBIT 的 MCLRE 配置，选择外部复位功能后复位引脚的内部上拉电阻自动有效。外部复位引脚 RST 是施密特结构的，低电平有效。当外复位引脚为高电平时，系统正常运行；为低电平时，系统产生复位。

## 4.4 低电压复位

MC32P8112 的 LVR 电压有 7 级（详见烧录配置选项），通过 OPBIT 的 VLVRS 进行配置。电压检测电路有一定的回滞特性，通常回滞电压为 0.05V 左右，则当电源电压下降到 LVR 电压时 LVR 复位有效，而电压需要上升到 LVR 电压+0.05V 时 LVR 复位才会解除。

## 4.5 看门狗复位

看门狗复位是一种对程序正常运行的保护机制。正常情况下，用户软件需要按时对 WDT 定时器进行清零操作，保证 WDT 不溢出。若出现异常状况，程序未按时对 WDT 定时器清零，WDT 会溢出从而产生看门狗复位，系统重新初始化，返回受控状态。

## 5 IO口

### 5.1 IO 工作模式

三组 8 位端口 P0、P1、P2 和一组 2 位端口 P3。P17-P14 分别和模拟通道 AD3-AD0 复用，P23-P20 模拟通道 AD7-AD4 复用，当模拟功能使能时端口上拉电阻仍可以设置。P17、P21 设置为输入端口时，可用作外部中断输入 INT0、INT1 及定时器 0 和定时器 1 的外部时钟输入；P24 设置为输入端口时，可用作定时器 2 外部时钟输入 TCC；当 PWM0OE=1，P16 自动转换为输出口，输出 PWM0，当 PWM1OE=1 时，P20 自动转换为输出口，输出 PWM1；当 BUZZ0OE=1 且 PWM0OE=0 时，P16 自动转换为输出口，输出 BUZZ0，当 BUZZ1OE=1 且 PWM1OE=0 时，P20 自动转换为输出口，输出 BUZZ1。当 HEN=1 时，P26 和 P27 用作 I2C 通讯端口 SCL 和 SDA，当外部晶振使能时，P23 和 P24 用作晶振端口。

#### 端口数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>IOP0</b>	P07D	P06D	P05D	P04D	P03D	P02D	P01D	P00D
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

**BIT[7:0] P0Nd** – P0 口数据位 (n=7-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>IOP1</b>	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

**BIT[7:0] P1Nd** – P1 口数据位 (n=7-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>IOP2</b>	P27D	P26D	P25D	P24D	P23D	P22D	P21D	P20D
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

**P2Nd** – P2 口数据位 (n=7-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>IOP3</b>	-	-	-	-	-	-	P31D	P30D
<b>R/W</b>	-	-	-	-	-	-	R/W	R/W
<b>初始值</b>	-	-	-	-	-	-	X	X

**BIT[1:0] P3Nd** – P3 口数据位 (n=1-0)



**端口方向寄存器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>OEPO</b>	P07OE	P06OE	P05OE	P04OE	P03OE	P02OE	P01OE	P00OE
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7:0] P0Noe** – P0 口输出使能寄存器 (n=7-0)

- 0: 作为输入口, 读 P0 口读取端口状态
- 1: 作为输出口, 读 P0 口读取 P0 口数据寄存器值

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>OEPI</b>	P17OE	P16OE	P15OE	P14OE	P13OE	P12OE	P11OE	P10OE
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7:0] P1Noe** – P1 口输出使能寄存器 (n=7-0)

- 0: 作为输入口, 读 P1 口读取端口状态
- 1: 作为输出口, 读 P1 口读取 P1 口数据寄存器值

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>OEPI</b>	P27OE	P26OE	P25OE	P24OE	P23OE	P22OE	P21OE	P20OE
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7:0] P2Noe** – P2 口输出使能寄存器 (n=7-0)

- 0: 作为输入口, 读 P2 口读取端口状态
- 1: 作为输出口, 读 P2 口读取 P2 口数据寄存器值

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>OEPI</b>	-	-	-	-	-	-	P31OE	P30OE
<b>R/W</b>	-	-	-	-	-	-	R/W	R/W
<b>初始值</b>	-	-	-	-	-	-	0	0

**BIT[1:0] P3Noe** – P3 口输出使能寄存器 (n=1-0)

- 0: 作为输入口, 读 P3 口读取端口状态
- 1: 作为输出口, 读 P3 口读取 P3 口数据寄存器值

## 5.2 上拉电阻控制

P0、P1、P2 和 P3 口每位都有独立的上拉控制寄存器位，控制其上拉电阻在端口作为输入状态时是否有效，端口处于输出状态时，上拉电阻控制位无效。

### 上拉电阻控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PUP0</b>	P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7:0] P0Npu** – P0 口上拉电阻选择 (n=7-0)

0: P0n 上拉电阻无效

1: P0n 上拉电阻有效

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PUP1</b>	P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7:0] P1Npu** – P1 口上拉电阻选择 (n=7-0)

0: P1n 上拉电阻无效

1: P1n 上拉电阻有效

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PUP2</b>	P27PU	P26PU	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7:0] P2Npu** – P2 口上拉电阻选择 (n=7-0)

0: P2n 上拉电阻无效

1: P2n 上拉电阻有效

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PUP3</b>	-	-	-	-	-	-	P31PU	P30PU
<b>R/W</b>	-	-	-	-	-	-	R/W	R/W
<b>初始值</b>	-	-	-	-	-	-	0	0

**BIT[1:0] P3Npu** – P3 口上拉电阻选择 (n=1-0)

0: P3n 上拉电阻无效

1: P3n 上拉电阻有效

### 5.3 端口模式控制

P17-P14 和 P23-P20 口可以作为通用 IO 口，也可以复用为模拟信号输入端口，ADCR1 寄存器可以设置这些端口的工作模式。当设置为通用 IO 时，相应端口的模拟输入被屏蔽；设置为模拟输入模式时，相应端口的输入功能被屏蔽。

#### ADC输入屏蔽寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADCR1</b>	ADIOS7	ADIOS6	ADIOS5	ADIOS4	ADIOS3	ADIOS2	ADIOS1	ADIOS0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

- BIT[7] ADIOS7** – AD7 端口选择位  
 0: P23 用作数字 IO，模拟输入被屏蔽  
 1: P23 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[6] ADIOS6** – AD6 端口选择位  
 0: P22 用作数字 IO，模拟输入被屏蔽  
 1: P22 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[5] ADIOS5** – AD5 端口选择位  
 0: P21 用作数字 IO，模拟输入被屏蔽  
 1: P21 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[4] ADIOS4** – AD4 端口选择位  
 0: P20 用作数字 IO，模拟输入被屏蔽  
 1: P20 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[3] ADIOS3** – AD3 端口选择位  
 0: P17 用作数字 IO，模拟输入被屏蔽  
 1: P17 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[2] ADIOS2** – AD2 端口选择位  
 0: P16 用作数字 IO，模拟输入被屏蔽  
 1: P16 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[1] ADIOS1** – AD1 端口选择位  
 0: P15 用作数字 IO，模拟输入被屏蔽  
 1: P15 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[0] ADIOS0** – AD0 端口选择位  
 0: P14 用作数字 IO，模拟输入被屏蔽  
 1: P14 用作 AD 模拟通道输入，数字输入被屏蔽

注：当引脚 AD 功能使能时，端口上拉电阻需要软件关闭

## 6 定时器

### 6.1 看门狗定时器 WDT

看门狗定时器的时钟为内置低频 RC 振荡输出，由 OPBIT 的 WDTC 设置看门狗定时器的工作状态。

若选择始终开启看门狗功能，在休眠模式下 WDT 依然运行，WDT 溢出时将唤醒休眠，CPU 继续运行；若 CPU 在运行时产生 WDT 溢出，WDT 溢出时复位芯片。

若选择休眠模式下关闭看门狗功能，在休眠模式下 WDT 被硬件自动关闭。

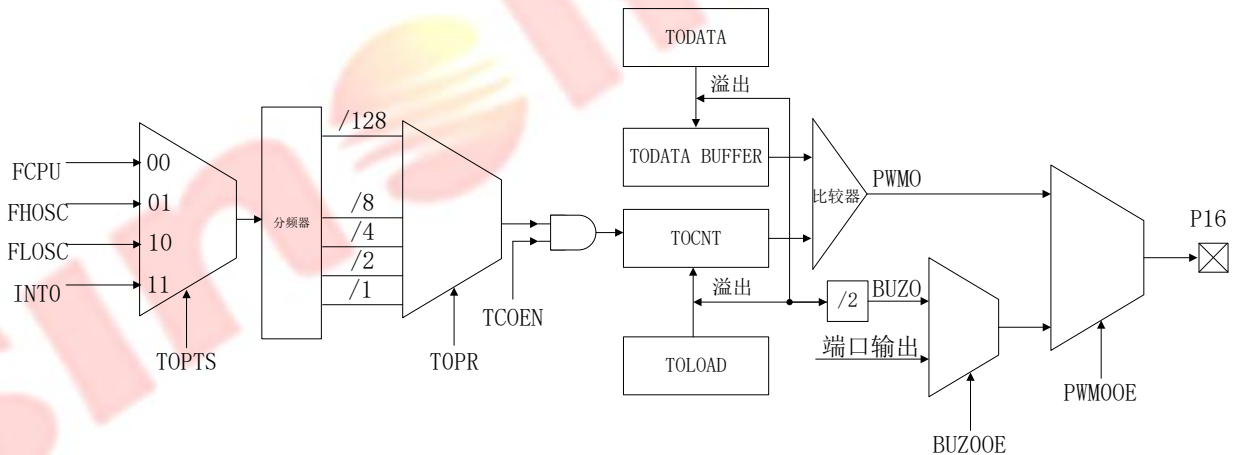
执行 CLRWDT 指令清 WDT。

WDT 溢出时间可通过 OPTION 配置设置为 16ms、64ms、256ms、1024ms、2048ms 和 4096ms。

### 6.2 定时器 T0

定时/计数器 T0 包含 1 个可编程预分频器，控制寄存器、重载寄存器及比较寄存器

- 可通过预分频比设置频率
- 通过重载寄存器设置周期
- 通过比较寄存器设置 PWM 占空比（仅 PWM 模式）
- BUZ 功能
- 溢出中断功能
- 溢出唤醒功能



注 1: FHOSC 指系统高频时钟源; FLOSC 指系统低频时钟源;

注 2: 当定时器选择高频时钟源 FHOSC 且时钟控制模块 OSCM 寄存器的 HFEN=1 时, 定时器在低频工作模式或休眠模式时可继续工作, 溢出可唤醒休眠模式; 若 HFEN=0 时, 定时器在低频工作模式或休眠模式下将停止工作

注 3: 当定时器选择低频时钟源 FLOSC 且时钟控制模块 OSCM 寄存器的 LFEN=1 时, 定时器在休眠模式时可继续工作, 溢出可唤醒休眠模式; 若 LFEN=0 时, 定时器在休眠模式下将停止工作

TOPTS 可选择 T0 的时钟源, TOPR 可选择 T0 的预分频比, 所选中的时钟源通过预分频器后产生 TOCNT 的时钟。

当 TOCNT 递减到 0 时, 此时产生 T0 溢出中断请求标志 TOIF 置 1, 重载寄存器值自动置入 TOCNT, TODATA 的值写入缓冲器 TODATABUFER 用于新的占空比波形生成, BUZO 信号反相。

通过 TOPR 可选择时钟源的分频比, 可选择范围为 1-128 分频, 对 TOCNT 的写操作将使预分频器清零, 分频比保持不变。

当 PWM0OE=1 时, 将输出 PWM 波形, 当 TOCNT 计数到与 TODATA 相等时, PWM0 输出置 1; 当 TOCNT 计数溢出时, PWM0 输出清 0, PWM0 占空比的计算如下:

$$\text{PWM0 高电平时间} = (\text{TODATA}) * \text{TOCNT 计数时钟周期}$$

$$\text{PWM0 周期 (T0 的溢出周期)} = (\text{TOLOAD}+1) * \text{TOCNT 的计数周期}$$

$$\text{PWM0 占空比} = (\text{TODATA} / (\text{TOLOAD}+1))$$

当 BUZOOE=1 且 PWM0OE=0 时, 输出 BUZO 信号, BUZO 信号的输出频率为 T0 溢出频率的 2 分频。

### 与定时器T0相关的寄存器说明如下

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TOCR</b>	TCOEN	PWM0OE	BUZOOE	TOPTS1	TOPTS0	TOPR2	TOPR1	TOPR0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7] TCOEN – T0 使能控制**

0: 关闭 T0

1: 启动 T0

**BIT[6] PWM0OE – PWM0 选择**

0: 禁止 PWM0 输出, 端口作为 I/O 口

1: 允许 PWM0 输出, 端口输出 PWM 信号

**BIT[5] BUZOOE – BUZO 选择**

0: 禁止 BUZO 输出, 端口作为 I/O 口

1: 允许 BUZO 输出 (PWM0OE 禁止), 端口输出 BUZ 信号

**BIT[4:3] TOPTS[1:0] – T0 时钟源选择**

TOPTS[1:0]	T0 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	INT0

**BIT[2:0] TOPR[2:0] – T0 预分频倍数选择**

TOPR2	TOPR1	TOPR0	TOCNT
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TOCNT</b>	T0C7	T0C6	T0C5	T0C4	T0C3	T0C2	T0C1	T0C0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] **T0C[7:0]** – T0CNT 的值，这是一个读写寄存器。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T0LOAD</b>	T0LOAD7	T0LOAD6	T0LOAD5	T0LOAD4	T0LOAD3	T0LOAD2	T0LOAD1	T0LOAD0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] **T0LOAD[7:0]** – T0LOAD 的值，这是一个读写寄存器，用于设置 T0 重载值。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T0DATA</b>	T0DATA7	T0DATA6	T0DATA5	T0DATA4	T0DATA3	T0DATA2	T0DATA1	T0DATA0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **T0DATA[7:0]** – 这是一个读写寄存器，用于设置 PWM0 高电平时间。

注：当 **TC0EN=0** 时，写 **T0LOAD** 将自动加载到 **T0CNT**；当 **TC0EN=1** 时，写 **T0LOAD** 时不自动加载到 **T0CNT**，在 T0 溢出时自动加载到 **T0CNT**

### 6.3 定时器 T1

定时器 T1 与定时器 T0 的功能及操作模式完全相同，在此仅做寄存器介绍。

#### 与定时器T1相关的寄存器说明如下

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T1CR</b>	TC1EN	PWM1OE	BUZ1OE	T1PTS1	T1PTS0	T1PR2	T1PR1	T1PR0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7] TC1EN** – T1 使能控制

0: 关闭 T1

1: 启动 T1

**BIT[6] PWM1OE** – PWM1 选择

0: 禁止 PWM1 输出，端口作为 I/O 口

1: 允许 PWM1 输出，端口输出 PWM 信号

**BIT[5] BUZ1OE** – BUZ1 选择

0: 禁止 BUZ1 输出，端口作为 I/O 口

1: 允许 BUZ1 输出（PWM1OE 禁止），端口输出 BUZ 信号

**BIT[4:3] T1PTS[1:0]** – T1 时钟源选择

T1PTS[1:0]	T1 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	INT1

**BIT[2:0] T1PR[2:0]** – T1 预分频倍数选择

T1PR2	T1PR1	T1PR0	T1CNT
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T1CNT</b>	T1C7	T1C6	T1C5	T1C4	T1C3	T1C2	T1C1	T1C0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

**BIT[7:0] T1C[7:0]** – T1CNT 的值，这是一个读写寄存器。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T1LOAD</b>	T1LOAD7	T1LOAD6	T1LOAD5	T1LOAD4	T1LOAD3	T1LOAD2	T1LOAD1	T1LOAD0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] **T1LOAD[7:0]** – T1LOAD 的值，这是一个读写寄存器，用于设置 T1 重载值。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T1DATA</b>	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **T1DATA[7:0]** – 这是一个读写寄存器，用于设置 PWM1 高电平时间。

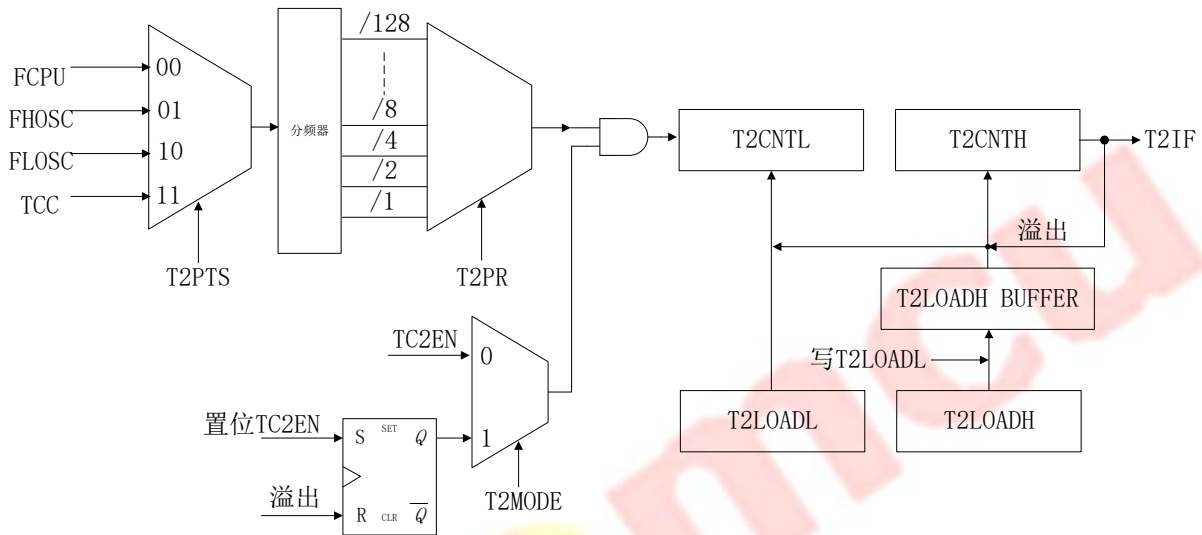
注：当 **TC1EN=0** 时，写 **T1LOAD** 将自动加载到 **T1CNT**；当 **TC1EN=1** 时，写 **T1LOAD** 时不自动加载到 **T1CNT**，在 T1 溢出时自动加载到 **T1CNT**



## 6.4 定时器 T2

定时/计数器 T2 包含 1 个可编程预分频器，控制寄存器、重载寄存器

- 可通过预分频比设置频率
- 通过重载寄存器设置周期
- 溢出中断功能
- 溢出唤醒功能
- 触摸按键时基功能



T2PTS 可选择 T2 的时钟源，T2PR 可选择 T2 的预分频比，所选中的时钟源通过预分频器后产生 T2CNT 的时钟。

当 T2CNT 递减到 0 时，此时产生 T2 溢出中断请求标志 T2IF 置 1，重载寄存器值自动置入 T2CNT。

通过 T2PR 可选择时钟源的分频比，可选择范围为 1-128 分频，对 T2CNT 的写操作将使预分频器清零，分频比保持不变。

上电后第一次开启 T2 前必须要先写一次 T2LOADL，否则 T2CNT 不加载 T2LOAD 的值。

T2 工作在触摸模式时，如果 TC2EN 位设置为 1，将同时使 3 组感应振荡器分别为对应的 16-bit C/F 计数器 (TKnCNT) 提供计数时钟，当 T2 溢出时，定时器溢出中断请求标志位(T2IF)将被置起，同时清除 TC2EN 为 0 并且 16-bit C/F 计数器 (TKnCNT) 停止计数，T2 定时器停止直到软件置位 TC2EN。

### 与定时器T2相关的寄存器说明如下

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T2CR</b>	TC2EN	-	T2MODE	T2PTS1	T2PTS0	TPR2	T2PR1	T2PR0
<b>R/W</b>	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	-	0	0	0	0	0	0

**BIT[7] TC2EN** – T2 使能控制

0: 关闭 T2

1: 启动 T2

**BIT[5] T2MODE** – T2 模式控制

0: 定时模式

1: 触摸模式

BIT[4:3] T2PTS[1:0] – T2 时钟源选择

T2PTS[1:0]	T2 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	TCC

BIT[2:0] T2PR[2:0] – T2 预分频倍数选择

T2PR2	T2PR1	T2PR0	T2CNT
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

注：时钟源选择 FHOSC 时，不能使用 1 分频。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T2CNTH</b>	T2C15	T2C14	T2C13	T2C12	T2C11	T2C10	T2C9	T2C8
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] T2CH[7:0] – T2CNTH 的值，这是一个读写寄存器。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T2CNTL</b>	T2C7	T2C6	T2C5	T2C4	T2C3	T2C2	T2C1	T2C0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] T2CL[7:0] – T2CNTL 的值，这是一个读写寄存器。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T2LOADH</b>	T2LOAD15	T2LOAD14	T2LOAD13	T2LOAD12	T2LOAD11	T2LOAD10	T2LOAD9	T2LOAD8
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] T2LOADH[7:0] – T2LOADH 的值，这是一个读写寄存器，用于设置 T2 重载值高 8 位。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T2LOADL</b>	T2LOAD7	T2LOAD6	T2LOAD5	T2LOAD4	T2LOAD3	T2LOAD2	T2LOAD1	T2LOAD0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] T2LOADL[7:0] – T2LOADL 的值，这是一个读写寄存器，用于设置 T2 重载值低 8 位。

---

注 1: 当 TC2EN=0 时, 写 T2LOAD 将自动加载到 T2CNT; 当 TC2EN=1 时, 写 T2LOAD 时不自动加载到 T2CNT, 在 T2 溢出时自动加载到 T2CNT

注 2: 写 T2LOAD 时要先写 T2LOADH, 再写 T2LOADL。读 T2CNT 时要先读取 T2CNTH, 后读 T2CNTL。

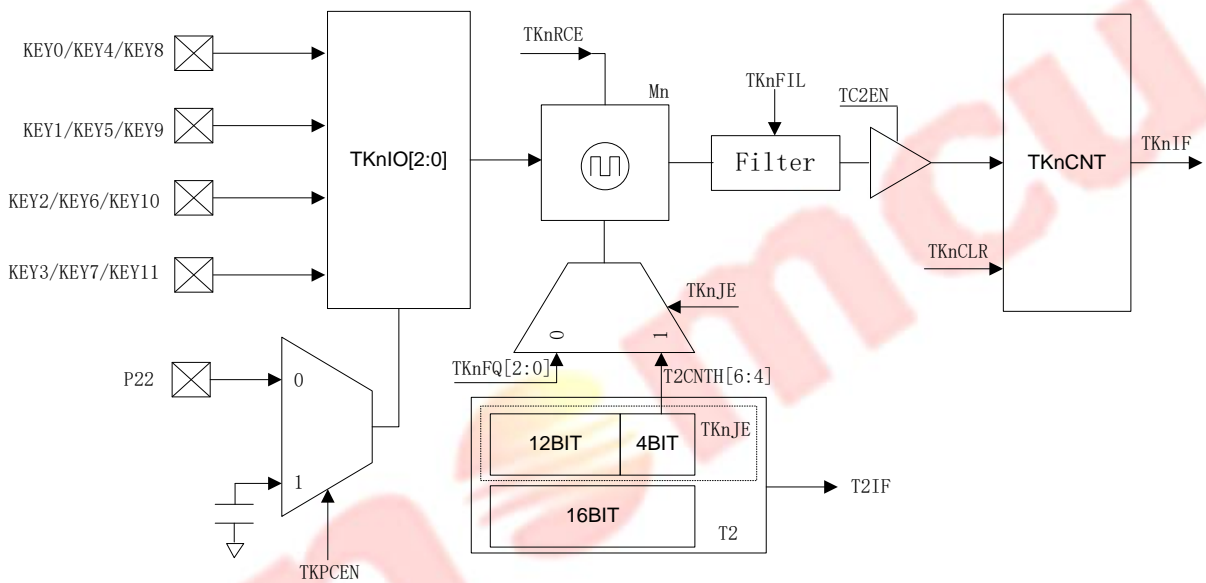
## 7 触摸按键

### 7.1 触摸功能介绍

芯片提供了 12 个触摸按键，不需外接元件，通过对内部寄存器进行简单的操作实现按键功能。

触摸按键与 P07-P00 和 P13-P10 管脚复用。通过寄存器的位来选择此功能。按键分成 M0、M1、M2 三个模块。每个模块具有自己的模拟电路和寄存器。寄存器的名称和它对应的模块编号相关联。

按键数	按键模块	按键	复用 IO
12	M0	KEY3-KEY0	P03-P00
	M1	KEY7-KEY4	P07-P04
	M2	KEY11-KEY8	P13-P10



## 7.2 触摸模块操作步骤

- ◇ S1: 设置 TknI03-TknI00 将相应的端口设置为触摸按键端口
- ◇ S2: 设置 TKnRCE 开启 Mn 按键感应振荡器
- ◇ S3: 设置 T2MODE 选择 T2 触摸模式
- ◇ S4: 设置 TKnFL 选择相应的滤波能力
- ◇ S5: 设置 TKnK 选择相应按键
- ◇ S6: 设置 TKnFQ 选择相应振荡频率
- ◇ S7: 设置 TKnCLR 为 1 开启 TKnCNT
- ◇ S8: 设置 TC2EN 为 1 启动 T2 定时
- ◇ S9: 等待 T2IF 置 1 或利用 T2 中断或等待 TC2EN 清零
- ◇ S10: 读取计数结果 (TKnCNTH、TKnCNTL)
- ◇ S11: 设置 TKnCLR 为 0 清零计数结果
- ◇ 重复 S5~S11 对不同的通道进行转换或对同一通道多次转换

**注 1:** T2 设置为触摸模式时, TC2EN 写入 1 启动 T2 定时, T2 溢出后自动清零 TC2EN 关闭 T2。

**注 2:** T2 设置为触摸模式时, TKnCNT 开始计数的条件为 TKnRCE=1, TKnCLR=1 后 TC2EN 写入 1, 当 T2 溢出后自动停止计数; T2 设置为定时模式时, TKnCNT 开始计数的条件为 TKnRCE=1 后 TKnCLR 写入 1, TKnCLR 写入 0 停止计数但同时清零 TKnCNT, 所以需要先读取 TKnCNT 的值再清零 TKnCLR。

**注 3:** T2 设置为触摸模式时且有任意一个 TknJE 使能的情况下 T2 将配置为 4bit×12bit 工作模式, 每次低 12bit 减至零溢出后高 4bit 值减 1 直至产生溢出中断, T2 的即 bit14-bit12 位为跳频频率选择位, 此时 TKnFQ 设置无效, TKnFQ 的设置等于 T2 的 bit14-bit12 位。

**注 4:** TKnFL 选择计数器 TKnCNT 的输入信号滤波能力, 可以滤除触摸按键引脚的杂波信号, 提高按键抗干扰性。如 TKnFL=11, 表示 2MHz 以上的信号被滤除。

### 7.3 触摸模块相关寄存器

每个触摸模块包含 4 个触摸按键，且都有自己相匹配的 4 个寄存器。以下表格显示了每个触摸模块的寄存器设置。寄存器名称里的 n(n=0~2)表明了触摸模块的序号 M0~M2。

名称	描述
TKnCNTH	16-bit C/F 高字节计数器
TKnCNTL	16-bit C/F 低字节计数器
TKnCRH	控制寄存器 0 按键选择、16-bit C/F 计数器控制、按键感应振荡器控制等
TKnCRL	控制寄存器 1 I/O 引脚或触摸引脚选择及中断控制等

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<i>TKnCNTH</i>	TKnC15	TKnC14	TKnC13	TKnC12	TKnC11	TKnC10	TKnC9	TKnC8
<i>R/W</i>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<i>初始值</i>	0	0	0	0	0	0	0	0

**BIT[7:0] TKnCNTH[7:0]** – 计数器 n 高 8 位值。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<i>TKnCNTL</i>	TKnC7	TKnC6	TKnC5	TKnC4	TKnC3	TKnC2	TKnC1	TKnC0
<i>R/W</i>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<i>初始值</i>	0	0	0	0	0	0	0	0

**BIT[7:0] TKnCNTL[7:0]** – 计数器 n 低 8 位值。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<i>TKnCRH</i>	TKnK1	TKnK0	TknJE	TKnRCE	TKnCLR	TKnFQ2	TKnFQ1	TKnFQ0
<i>R/W</i>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<i>初始值</i>	0	0	0	0	0	0	0	0

**BIT[7:6] TKnK[1:0]** – 按键选择位

00: KEY0/KEY4/KEY8

01: KEY1/KEY5/KEY9

10: KEY2/KEY6/KEY10

11: KEY3/KEY7/KEY11

**BIT[5] TknJE** – 跳频使能位

0: 跳频关闭

1: 跳频启动

**BIT[4] TKnRCE** – 振荡器使能位

0: Mn 按键感应振荡器关闭

1: Mn 按键感应振荡器开启

**BIT[3] TKnCLR** – 计数器开关位

0: TKnCNT 清零并关闭

1: TKnCNT 开启

**BIT[2:0]**     **TKnFQ [2:0]** – 振荡频率选择位  
 000: 800KHz (参考值)  
 001: 1050KHz (参考值)  
 010: 1300KHz (参考值)  
 011: 1550KHz (参考值)  
 100: 1800KHz (参考值)  
 101: 2050KHz (参考值)  
 110: 2300KHz (参考值)  
 111: 2600KHz (参考值)

注：上述频率参考值为引脚外接 15Pf 电容时设计值，实际值与引脚寄生电容有关。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TKnCRL</b>	TknIF	TknIE	TKnFL1	TKnFL0	TknIO3	TknIO2	TknIO1	TknIO0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7]**     **TknIF** – 计数器 n 溢出标志  
 0: 未发生计数器 n 溢出中断  
 1: 发生计数器 n 溢出中断，需软件清零

**BIT[6]**     **TknIE** – 计数器 n 使能  
 0: 屏蔽计数器 n 中断  
 1: 使能计数器 n 中断

**BIT[5:4]**   **TKnFL[1:0]** – 低通滤波选择位  
 00: 16MHz 滤波  
 01: 8MHz 滤波  
 10: 4MHz 滤波  
 11: 2MHz 滤波

**BIT[3]**     **TknIO3** – 按键端口选择位  
 0: 按键端口为 IO 口  
 1: 按键端口为 KEY3/KEY7/KEY11

**BIT[2]**     **TknIO2** – 按键端口选择位  
 0: 按键端口为 IO 口  
 1: 按键端口为 KEY2/KEY6/KEY10

**BIT[1]**     **TknIO1** – 按键端口选择位  
 0: 按键端口为 IO 口  
 1: 键端口为 KEY1/KEY5/KEY9

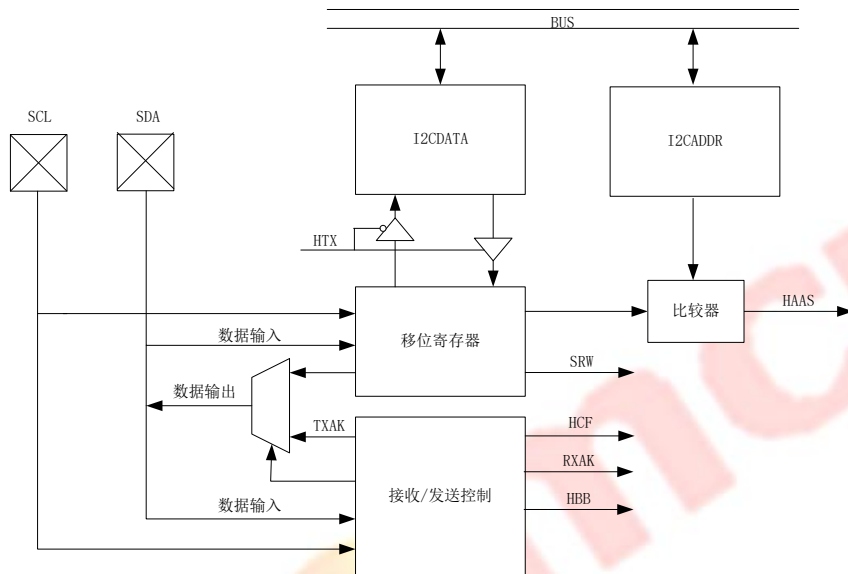
**BIT[0]**     **TknIO0** – 按键端口选择位  
 0: 按键端口为 IO 口  
 1: 按键端口为 KEY0/KEY4/KEY8

注：当引脚触摸按键功能使能时，端口上拉电阻需要软件关闭

## 8 I2C总线接口

### 8.1 I2C 接口介绍

芯片提供 I2C 总线接口，此芯片只能做为从机，I2C 总线是双向两线结构，数据线和时钟线分别为 SDA 和 SCL。SDA 和 SCL 是 NMOS 开漏输出，使用时必须接上拉电阻。



芯片有两种数据传输方式，一种为从器件发送模式，另一种为从器件接收模式。

I2CADDR 寄存器用来存放从器件地址，第 7-1 位定义从器件地址（第 0 位不需要定义），开始数据传送时，主机通过发送从器件地址来选择此芯片，如果主机发送的地址与 I2CADDR 中的地址匹配，则表明此芯片被选中。

I2CDATA 是数据输入/输出寄存器，发送和接收的数据都必须经过 I2CDATA 寄存器。

I2CCR 是 I2C 总线控制寄存器，用来控制 I2C 总线功能的开/关状态、I2C 总线是工作于发送模式还是接收模式和 I2C 总线的工作状态。

I2C 总线控制寄存器包括 8 位：HEN、HCF、HAAS、HBB、SRW、RXAK、HTX 和 TXAK。

- ✧ HEN 位控制 I2C 总线打开/关闭，如果数据想通过 I2C 总线传送，该位就必须置“1”。
- ✧ HTX 位配置芯片是处于发送模式还是接收模式，如果是发送模式，该位就必须置“1”。
- ✧ TXAK 位定义接收模式下是否发送应答信号，当芯片接收到 8 位数据后，在第 9 个时钟时，芯片将该位送到 I2C 总线上，如果要继续接收下一个数据，在接收数据前该位必须清“0”。
- ✧ HCF 位在开始传送数据时被清“0”；在数据传送结束后被置“1”。
- ✧ HAAS 位在地址匹配时被置“1”，同时 I2C 总线中断请求标志被置“1”；如果地址不匹配，HAAS 被清“0”。
- ✧ HBB 位置“1”表示 I2C 总线忙，即系统检测到“START”信号；HBB 清“0”表示 I2C 总线空闲，即系统检测到“STOP”信号，此时 I2C 总线空闲。
- ✧ SRW 位表示地址匹配时主机的读/写请求。当 HAAS 被置“1”，可以通过检测 SRW 来确定主机配置芯片是工作在发送模式还是接收模式。当 SRW 被置“1”，表示主机要从芯片中读数据，芯片必须将数据写到 I2C 总线，即芯片为发送模式；当 SRW 被清“0”，表示主机要写数据到芯片，芯片要从总线读取数据，即芯片为接收模式。
- ✧ RXAK 位清“0”，表示芯片在发送模式下接收到主机的应答信号。在发送模式，芯片通过检



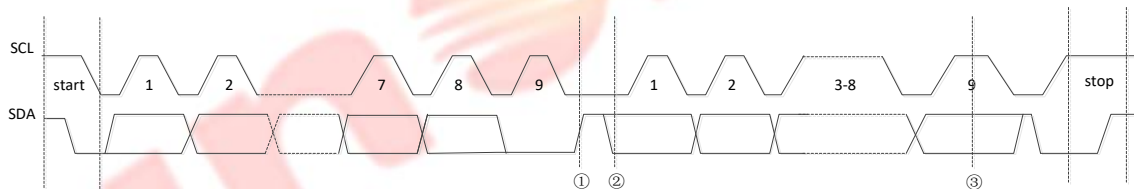
测 RXAK 以确定是否要接收下一个数据，芯片会一直送数据到 I2C 总线直到 RXAK 置“1”，此时芯片释放 SDA，这样主机可以发送 STOP 信号来释放 I2C 总线。

一旦 I2C 总线接上出现起始信号，所有的从器件都会接收连续的 8 位数，该数据的前 7 位是从器件地址，高位在前，低位在后。如果从器件地址匹配，芯片会置位 HAAS，同时产生 I2C 总线中断。进入中断服务程序后，芯片要检测 HAAS 位，以确定 I2C 总线中断是来自地址匹配，还是来自 8 位数据传送完毕。8 位数据的最后一位是读/写控制位，该位会反映到 SRW。芯片通检测 SRW 以确定主机是要发送数据还是接收数据，并确定自己是作发送器还是接收器。

## 8.2 I2C 操作步骤

- ◇ 向 I2C 总线地址寄存器 I2CADDR 写入从器件地址。
- ◇ 置位中断控制寄存器的 I2CIE 位，以允许 I2C 总线中断。
- ◇ 置位 I2C 总线控制寄存器的 HEN 位，以打开 I2C 总线。
- ◇ 当从器件地址匹配且 TXAK=0 时，图中①处 HAAS 置 1 产生中断，并将时钟线 SCL 拉低。
- ◇ 通过检测 SRW 来确定芯片是工作在发送模式还是接收模式，如果为发送模式，设置 HTX 为 1，将数据写入 I2CDATA，如果为接收模式，设置 HTX 为 0，设置 TXAK 已决定下一帧是否应答，图中③处所示。
- ◇ 清零 I2CIF 释放 SCL 总线，图中②处所示。
- ◇ 当下一帧第 9 个时钟结束时，HCF 置 1 产生中断，如果芯片处于接收模式并且 TXAK 为 0 或者芯片处于发送模式并且 RXAK 为 0，将拉低时钟线 SCL；否则芯片会释放时钟线 SCL 和数据线 SDA 等待主机发 STOP 命令。

注：MC32P8112 I2C 接口不支持多重机操作。



## 8.3 I2C 相关寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>I2CCR</b>	HCF	HAAS	HBB	HTX	TXAK	SRW	HEN	RXAK
<b>R/W</b>	R	R	R	R/W	R/W	R	R/W	R
<b>初始值</b>	0	0	0	0	1	0	0	0

**BIT[7] HCF** – 数据传输结束标志位

0: 数据传输中

1: 数据传输结束

**BIT[6] HAAS** – 地址匹配标志位

0: 地址不匹配

1: 地址匹配

- BIT[5] HBB** – 总线忙标志位  
 0: 总线闲  
 1: 总线忙
- BIT[4] HTX** – 模式标志位  
 0: 芯片设置为接收模式  
 1: 芯片设置为发送模式
- BIT[3] TXAK** – 接收应答位  
 0: 接收模式下芯片应答  
 1: 接收模式下芯片不应答
- BIT[2] SRW** – 读写标志位  
 0: 主机发送写命令  
 1: 主机发送读命令
- BIT[1] HEN** – 总线使能位  
 0: 总线不使能  
 1: 总线使能
- BIT[0] RXAK** – 发送应答标志位  
 0: 发送模式下主机应答  
 1: 发送模式下主机不应答

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>I2CADDR</b>	I2CAD7	I2CAD6	I2CAD5	I2CAD4	I2CAD3	I2CAD2	I2CAD1	-
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
<b>初始值</b>	0	0	0	0	0	0	0	-

**BIT[7:1] I2CAD[7:1]** – I2C 从器件地址位(n=7-1)。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>I2CDATA</b>	I2CDA7	I2CDA6	I2CDA5	I2CDA4	I2CDA3	I2CDA2	I2CDA1	I2CDA0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7:0] I2CDA[7:0]** – I2C 数据位(n=7-0)。

## 9 模数转换器

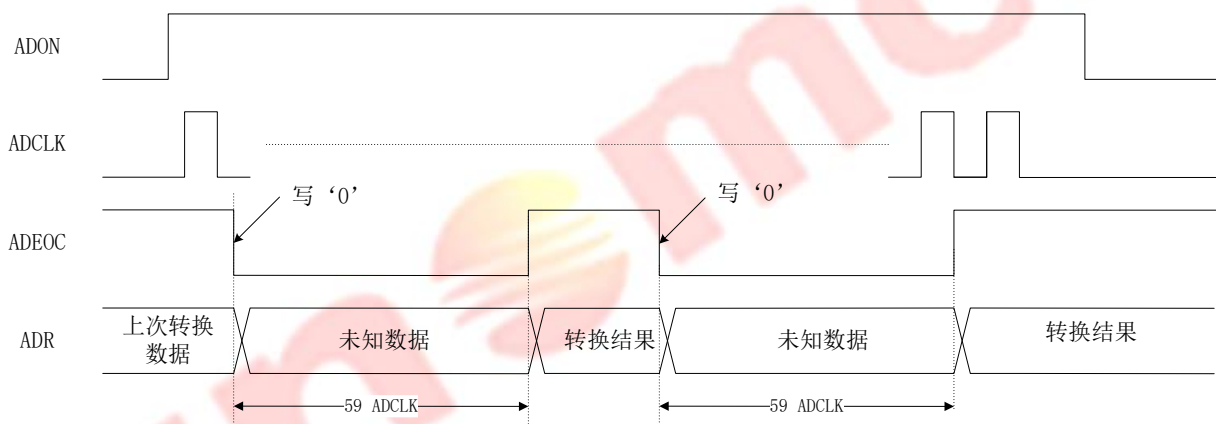
### 9.1 ADC 功能介绍

8 通道 12 位模数转换器，可通过 ADON 使能模数转换模块，ADCHS 选择转换的模拟通道，ADCKS 选择 AD 转换速度，ADEOC 为 AD 启动位及转换结束标志位。

当 ADEOC 标志为 1 时，对该寄存器写入 0 将启动模数转换，转换时间需要 59 个 AD 转换时钟周期，转换结果被放在 ADRH 和 ADRL 中，ADEOC 将自动置 1，同时中断标志 ADIF 置 1，若 GIE 和 ADIE 使能，将产生 AD 中断。

**注：要保证 A/D 转换的精度和线性度，芯片工作电压需在 3.0V 以上，低于此工作电压，A/D 转换的精度和线性度会变差，详细说明见《10 位/12 位 ADC 应用笔记》。**

### 9.2 ADC 转换时序图



### 9.3 ADC 操作步骤

模数转换设置步骤

- ◇ S1: 设置 OEPX 将相应的端口设置为输入端口，设置 PUPX 关闭上拉电阻
- ◇ S2: 设置 ADIOS 将相应的端口设置为模拟端口
- ◇ S3: 设置 ADCKS 选取适当的 AD 转换时钟
- ◇ S4: 使能 ADON
- ◇ S5: 设置 ADCHS 选取 AD 转换通道
- ◇ S6: ADEOC 写入 0 启动 AD 转换
- ◇ S7: 等待 ADEOC 置 1 (或利用 AD 中断)
- ◇ S8: 读取 AD 转换结果 (ADRH、ADRL)
- ◇ 重复 S5~S8 对不同的通道进行转换或对同一通道多次转换

**注：AD 转换过程中或者 ADON 未使能时，ADRH/ADRL 中的数据未知，应在 AD 转换结束且 ADON 使能的情况下读取 AD 转换数据**

## 9.4 ADC 相关寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADCR0</b>	ADCHS3	ADCHS2	ADCHS1	ADCHS0	ADCKS1	ADCKS0	ADEOC	ADON
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	1	0

**BIT[7:4] ADCHS[3:0] – ADC 模拟通道选择**

ADCHS[3:0]	AD 模拟通道
0000	AD0
0001	AD1
0010	AD2
0011	AD3
0100	AD4
0101	AD5
0110	AD6
0111	AD7
其它	-

**BIT[3:2] ADCKS[1:0] – AD 转换时钟选择**

ADCKS[1:0]	AD 转换时钟选择 FADC
00	Fcpu/16
01	Fcpu/8
10	Fcpu/4
11	Fcpu/2

**BIT[1] ADEOC – ADC 启动位及转换结束标志**

0: AD 转换过程中, 转换结束后自动置 1

1: AD 转结束, 对 ADEOC 写入 0 启动 AD 转换

**BIT[0] ADON – ADC 功能使能位**

0: 不使能 ADC 功能

1: 使能 ADC 功能

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADCR1</b>	ADIOS7	ADIOS6	ADIOS5	ADIOS4	ADIOS3	ADIOS2	ADIOS1	ADIOS0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

**BIT[7] ADIOS7 – AD7 端口选择位**

0: P23 用作数字 IO, 模拟输入被屏蔽

1: P23 用作 AD 模拟通道输入, 数字输入被屏蔽

**BIT[6] ADIOS6 – AD6 端口选择位**

0: P22 用作数字 IO, 模拟输入被屏蔽

1: P22 用作 AD 模拟通道输入, 数字输入被屏蔽

- BIT[5] ADIOS5** – AD5 端口选择位  
 0: P21 用作数字 IO，模拟输入被屏蔽  
 1: P21 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[4] ADIOS4** – AD4 端口选择位  
 0: P20 用作数字 IO，模拟输入被屏蔽  
 1: P20 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[3] ADIOS3** – AD3 端口选择位  
 0: P17 用作数字 IO，模拟输入被屏蔽  
 1: P17 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[2] ADIOS2** – AD2 端口选择位  
 0: P16 用作数字 IO，模拟输入被屏蔽  
 1: P16 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[1] ADIOS1** – AD1 端口选择位  
 0: P15 用作数字 IO，模拟输入被屏蔽  
 1: P15 用作 AD 模拟通道输入，数字输入被屏蔽
- BIT[0] ADIOS0** – AD0 端口选择位  
 0: P14 用作数字 IO，模拟输入被屏蔽  
 1: P14 用作 AD 模拟通道输入，数字输入被屏蔽

注：当引脚 AD 功能使能时，端口上拉电阻需要软件关闭

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADRH</b>	ADR11	ADR10	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

**BIT[7:0] ADR[11:4]** – AD 转换结果寄存器高 8 位。

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADRL</b>	-	-	-	-	ADR3	ADR2	ADR1	ADR0
<b>R</b>	-	-	-	-	R	R	R	R
<b>初始值</b>	-	-	-	-	X	X	X	X

**BIT[3:0] ADR[3:0]** – AD 转换结果寄存器低 4 位。

## 10 中断

MC32P8112 的中断有外中断(INT0,INT1)、定时器中断(T0,T1,T2)、触摸计数器中断(TK0, TK1, TK2)、I2C 中断和 ADC 转换中断。外部中断、定时器中断、触摸计数器中断、I2C 中断和 ADC 转换中断可被 CPU 状态寄存器 MCR 的 GIE 位屏蔽。

中断响应过程如下

- ✧ 当发生中断请求时，CPU 将相关下一条要执行的指令的地址压栈保存（累加器 A 和状态寄存器需要软件保护），对中断屏蔽位 GIE 清 0，禁止中断响应。与复位不同，硬件中断不停止当前指令的执行，而是暂时挂起中断直到当前指令执行完成。
- ✧ CPU 执行中断时，程序跳到中断向量 0008H 地址开始执行中断代码，中断代码应该先保存累加器 A 和状态寄存器，然后判断是哪一个中断响应。
- ✧ 执行中断内容后应该恢复累加器 A 和状态寄存器，然后执行 RETIE 返回主程序。这时，从堆栈取出 PC 的值，然后从中断发生时的那条指令的后一条指令继续执行。

MC32P8112 的中断向量地址是 0008H。

### 10.1 外中断

MC32P8112 有 2 路外部中断源，两路中断源可以设置为上升沿触发、下降沿触发和电平变化触发三种模式，当外部中断触发时，外部中断标志（INT0IF、INT1IF）将被置 1，若中断总使能位 GIE 为 1 且外部中断使能位（INT0IE、INT1IE）为 1，则产生外部中断。

### 10.2 定时器中断

定时器 T0、T1 和 T2 在计数溢出时会置位中断标志 T0IF、T1IF 和 T2IF，若中断总使能位 GIE 为 1 且定时器中断使能位（T0IE、T1IE 和 T2IE）为 1，则产生定时器中断。

### 10.3 触摸计数器中断

触摸计数器 TK0、TK1 和 TK2 在计数溢出时会置位中断标志 TK0IF、TK1IF 和 TK2IF，若中断总使能位 GIE 为 1 且触摸计数器中断使能位（TK0IE、TK1IE 和 TK2IE）为 1，则产生触摸计数器中断。

### 10.4 I2C 中断

I2C 器件地址匹配、数据发送完或者数据接收完后会置位中断标志 I2CIF，若中断总使能位 GIE 为 1 且 I2C 中断使能位（I2CIE）为 1，则产生 I2C 中断。

### 10.5 ADC 转换中断

ADC 转换完成后会置位中断标志 ADIF，若中断总使能位 GIE 为 1 且 AD 中断使能位(ADIE)为 1，则产生 ADC 中断。

## 10.6 中断相关寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>MCR</b>	GIE	-	TO	PD	MINT11	MINT10	MINT01	MINT00
<b>R/W</b>	R/W	-	R	R	R/W	R/W	R/W	R/W
<b>初始值</b>	0	-	0	0	0	0	0	0

- BIT[7] GIE** – 总中断使能  
0: 屏蔽所有中断  
1: 中断源是否产生中断由相应的控制位决定
- BIT[6]** 未用
- BIT[5] TO** – 看门狗溢出标志  
0: 上电复位, 执行CLRWDWT或STOP指令  
1: 发生WDT溢出
- BIT[4] PD** – 进入低功耗休眠模式标志  
0: 上电复位, 执行CLRWDWT  
1: 执行STOP指令
- BIT[3:2] MINT1[1:0]** – 外部中断 1 模式寄存器  
00: INT1 上升沿触发  
01: INT1 下降沿触发  
1x: INT1 电平变化触发
- BIT[1:0] MINT0[1:0]** – 外部中断 0 模式寄存器  
00: INT0 上升沿触发  
01: INT0 下降沿触发  
1x: INT0 电平变化触发

### 中断使能寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INTE</b>	-	ADIE	I2CIE	T2IE	INT1IE	INT0IE	T1IE	TOIE
<b>R/W</b>	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	-	0	0	0	0	0	0	0

- BIT[6] ADIE** – ADC 中断使能  
0: 屏蔽 ADC 中断  
1: 使能 ADC 中断
- BIT[5] I2CIE** – I2C 中断使能  
0: 屏蔽 I2C 中断  
1: 使能 I2C 中断
- BIT[4] T2IE** – 定时器 2 使能  
0: 屏蔽定时器 2 中断  
1: 使能定时器 2 中断
- BIT[3] INT1IE** – 外部中断 1 使能  
0: 屏蔽外部 1 中断  
1: 使能外部 1 中断

- BIT[2]**     **INT0IE** – 外部中断 0 使能  
           0: 屏蔽外部 0 中断  
           1: 使能外部 0 中断
- BIT[1]**     **T1IE** – 定时器 1 使能  
           0: 屏蔽定时器 1 中断  
           1: 使能定时器 1 中断
- BIT[0]**     **T0IE** – 定时器 0 使能  
           0: 屏蔽定时器 0 中断  
           1: 使能定时器 0 中断

**中断标志寄存器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INTF</b>	-	ADIF	I2CIF	T2IF	INT1IF	INT0IF	T1IF	T0IF
<b>R/W</b>	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	-	0	0	0	0	0	0	0

- BIT[6]**     **ADIF** – ADC 中断标志  
           0: 未发生 ADC 中断  
           1: 发生 ADC 中断，需软件清零
- BIT[5]**     **I2CIF** – I2C 中断标志  
           0: 未发生 I2C 中断  
           1: 发生 I2C 中断，需软件清零
- BIT[4]**     **T2IF** – 定时器 2 标志  
           0: 未发生定时器 2 中断  
           1: 发生定时器 2 中断，需软件清零
- BIT[3]**     **INT1IF** – 外部中断 1 标志  
           0: 未发生外部 1 中断  
           1: 发生外部 1 中断，需软件清零
- BIT[2]**     **INT0IF** – 外部中断 0 标志  
           0: 未发生外部 0 中断  
           1: 发生外部 0 中断，需软件清零
- BIT[1]**     **T1IF** – 定时器 1 标志  
           0: 未发生屏蔽定时器 1 中断  
           1: 发生定时器 1 中断，需软件清零
- BIT[0]**     **T0IF** – 定时器 0 标志  
           0: 未发生定时器 0 中断  
           1: 发生定时器 0 中断，需软件清零



## 11 电气参数

### 11.1 极限参数

参数	符号	值	单位
工作电压	VDD	-0.3~6.0	V
输入电压	VIN	GND-0.3 ~ VDD+0.3	V
工作温度	TA	-40 ~ 85	°C
储存温度	Tstg	-65 ~ 150	°C

### 11.2 直流特性参数

T=25°C

特性	符号	引脚	条件	最小	典型	最大	单位
工作电压	VDD	VDD	Fcpu=8MHz@内部高频 16MHz	3.0		5.5	V
			Fcpu=4MHz@内部高频 16MHz	2.4		5.5	
			Fcpu=2MHz@内部高频 16MHz	2.0		5.5	
			Fcpu=1MHz@内部高频 16MHz	2.0		5.5	
			Fcpu=FILRC/2@内部低频	1.8		5.5	
			Fcpu=32768Hz/2@外部低频	1.8		5.5	
			Fcpu=8MHz@外部晶体 16MHz	3.0		5.5	
			Fcpu=4MHz@外部晶体 16MHz	2.7		5.5	
			Fcpu=2MHz@外部晶体 16MHz	2.7		5.5	
			Fcpu=1MHz@外部晶体 16MHz	2.7		5.5	
			Fcpu=4MHz@外部晶体 8MHz	2.4		5.5	
			Fcpu=2MHz@外部晶体 8MHz	2.4		5.5	
			Fcpu=1MHz@外部晶体 8MHz	2.4		5.5	
			Fcpu=2MHz@外部晶体 4MHz	2.2		5.5	
			Fcpu=1MHz@外部晶体 4MHz	2.2		5.5	
Fcpu=227.5KHz@外部晶体 455KHz	2.0		5.5				
输入漏电	V <sub>leak</sub>	所有输入引脚		-1		1	uA
输入高电平	V <sub>ih</sub>	所有输入引脚		0.7VDD			V
输入低电平	V <sub>il</sub>	所有输入引脚				0.3VDD	V
上拉电阻 1	R <sub>pu1</sub>	P0 P1 P2 P3 (除 P25)	VDD=5V,Vin=GND	25	50	100	Kohm
			VDD=3V,Vin=GND	50	100	200	Kohm
上拉电阻 2	R <sub>pu2</sub>	P25	VDD=5V,Vin=GND	15	30	60	Kohm
			VDD=3V,Vin=GND	15	30	60	Kohm
输出高电平 驱动电流	I <sub>oh</sub>	P0 P1 P2 P3 (除 P25)	VDD=5V,Voh=4.4V	5	10		Ma
			VDD=3V,Voh=2.4V	4	8		
输出低电平	I <sub>ol</sub>	P0 P1 P2 P3	VDD=5V,Vol=0.6V	10	20		Ma

驱动电流			VDD=3V,Vol=0.6V	8	16		
动态功耗	I <sub>ddc</sub>	VDD=5V	内部高频振荡器为主时钟, Fcpu=8MHz		3.5		Ma
			内部高频振荡器为主时钟, Fcpu=4MHz		2.5		Ma
			内部高频振荡器为主时钟, Fcpu=2MHz		1.8		Ma
			内部高频振荡器为主时钟, Fcpu=1MHz		1.5		Ma
		VDD=3V	内部高频振荡器为主时钟, Fcpu=8MHz		1.8		Ma
			内部高频振荡器为主时钟, Fcpu=4MHz		1.3		Ma
			内部高频振荡器为主时钟, Fcpu=2MHz		1.1		Ma
			内部高频振荡器为主时钟, Fcpu=1MHz		0.9		Ma
		VDD=5V	外部晶体 16MHz 为主时钟, Fcpu=8MHz		4.2		Ma
			外部晶体 16MHz 为主时钟, Fcpu=4MHz		2.8		Ma
			外部晶体 16MHz 为主时钟, Fcpu=2MHz		1.8		Ma
			外部晶体 16MHz 为主时钟, Fcpu=1MHz		1.6		Ma
		VDD=3V	外部晶体 16MHz 为主时钟, Fcpu=8MHz		1.8		Ma
			外部晶体 16MHz 为主时钟, Fcpu=4MHz		1.2		Ma
			外部晶体 16MHz 为主时钟, Fcpu=2MHz		0.9		Ma
			外部晶体 16MHz 为主时钟, Fcpu=1MHz		0.8		Ma
		VDD=5V	外部晶体 8MHz 为主时钟, Fcpu=4MHz		2.7		Ma
			外部晶体 8MHz 为主时钟, Fcpu=2MHz		1.7		Ma
			外部晶体 8MHz 为主时钟, Fcpu=1MHz		1.3		Ma
			外部晶体 8MHz 为主时钟, Fcpu=500KHz		0.9		Ma
VDD=3V	外部晶体 8MHz 为主时钟,		1.2		Ma		

			Fcpu=4MHz				
			外部晶体 8MHz 为主时钟, Fcpu=2MHz		0.8		Ma
			外部晶体 8MHz 为主时钟, Fcpu=1MHz		0.6		Ma
			外部晶体 8MHz 为主时钟, Fcpu=500KHz		0.4		Ma
		VDD=5V	外部晶体 4MHz 为主时钟, Fcpu=2MHz		1.6		Ma
			外部晶体 4MHz 为主时钟, Fcpu=1MHz		1.2		Ma
			外部晶体 4MHz 为主时钟, Fcpu=500KHz		0.8		Ma
			外部晶体 4MHz 为主时钟, Fcpu=250KHz		0.7		Ma
		VDD=3V	外部晶体 4MHz 为主时钟, Fcpu=2MHz		0.7		Ma
			外部晶体 4MHz 为主时钟, Fcpu=1MHz		0.5		Ma
			外部晶体 4MHz 为主时钟, Fcpu=500KHz		0.3		Ma
			外部晶体 4MHz 为主时钟, Fcpu=250KHz		0.25		Ma
		VDD=5V	外部晶体 455KHz 为主时钟, Fcpu=227.5KHz		0.7		Ma
			外部晶体 455KHz 为主时钟, Fcpu=113.75Hz		0.65		Ma
			外部晶体 455KHz 为主时钟, Fcpu=56.875KHz		0.58		Ma
			外部晶体 455KHz 为主时钟, Fcpu=28.4375KHz		0.55		Ma
		VDD=3V	外部晶体 455KHz 为主时钟, Fcpu=4MHz		0.2		Ma
			外部晶体 455KHz 为主时钟, Fcpu=2MHz		0.18		Ma
			外部晶体 455KHz 为主时钟, Fcpu=1MHz		0.16		Ma
			外部晶体 455KHz 为主时钟, Fcpu=500KHz		0.14		Ma
VDD=5V	内部 28KHz 低频振荡器为主时钟		15	30	Ua		
	外部 32768Hz 低频振荡器为主时钟		20	40			
VDD=3V	内部 28KHz 低频振荡器为主时钟		5	10	Ua		

			外部 32768Hz 低频振荡器为主时钟	10	20	
HOLD 模式 1 功耗	I <sub>hold1</sub>	VDD=5V	ADC 关闭, 内部高频振荡开启, 执行 STOP 指令	700		Ua
			ADC 关闭, 外部 16MHz 高频振荡开启, 执行 STOP 指令	1200		Ua
			ADC 关闭, 外部 8MHz 高频振荡开启, 执行 STOP 指令	800		Ua
			ADC 关闭, 外部 4MHz 高频振荡开启, 执行 STOP 指令	600		Ua
			ADC 关闭, 外部 455KHz 高频振荡开启, 执行 STOP 指令	520		Ua
		VDD=3V	ADC 关闭, 内部高频振荡开启, 执行 STOP 指令	550		Ua
			ADC 关闭, 外部 16MHz 高频振荡开启, 执行 STOP 指令	450		Ua
			ADC 关闭, 外部 8MHz 高频振荡开启, 执行 STOP 指令	320		Ua
			ADC 关闭, 外部 4MHz 高频振荡开启, 执行 STOP 指令	220		Ua
			ADC 关闭, 外部 455KHz 高频振荡开启, 执行 STOP 指令	140		Ua
HOLD 模式 2 功耗	I <sub>hold2</sub>	VDD=5V	ADC 关闭, 高频振荡关闭, 内部低频振荡开启, 执行 STOP 指令	3	6	Ua
			ADC 关闭, 高频振荡关闭, 外部 32768 低频振荡开启, 执行 STOP 指令	10	20	Ua
		VDD=3V	ADC 关闭, 高频振荡关闭, 内部低频振荡开启, 执行 STOP 指令	1	2	Ua
			ADC 关闭, 高频振荡关闭, 外部 32768 低频振荡开启, 执行 STOP 指令	2	4	Ua
休眠模式功耗	I <sub>STOP</sub>	VDD	ADC 关闭, WDT 关闭, 高频振荡器关闭, 低频振荡器关闭, 执行 STOP 指令	0.1	1	Ua

### 11.3 ADC 特性参数

T=25°C

特性	符号	条件	最小	典型	最大	单位
ADC 有效工作电压	V <sub>ADC</sub>		3.0		5.5	V
积分线性误差	ILE	VDD=5V FADC=4MHz			±3	LSB
微分线性误差	DLE	VDD=5V FADC=4MHz			±1	LSB
上限偏置误差	EOT	VDD=5V FADC=4MHz		±1	±3	LSB
下限偏置误差	EOB	VDD=5V FADC=4MHz		±1	±3	LSB
转换精度	ACC	VDD=5V FADC=4MHz			±3	LSB
转换时钟	FADC	VDD=5V			4	MHz
		VDD=3V			2	MHz
转换时间	T <sub>con</sub>			59	1/FADC	
ADC 输入电压	V <sub>IAN</sub>		0		VDD	V
ADC 输入阻抗	R <sub>IAN</sub>		2			Mohm
ADC 输入电流	I <sub>IAN</sub>	VDD=5V			10	Ua
ADC 动态电流	I <sub>add</sub>	VDD=5V AD 转换中		1	3	Ma
ADC 静态电流	I <sub>ads</sub>	VDD=5V ADON=0		0.1	0.5	Ua

## 11.4 交流电气参数

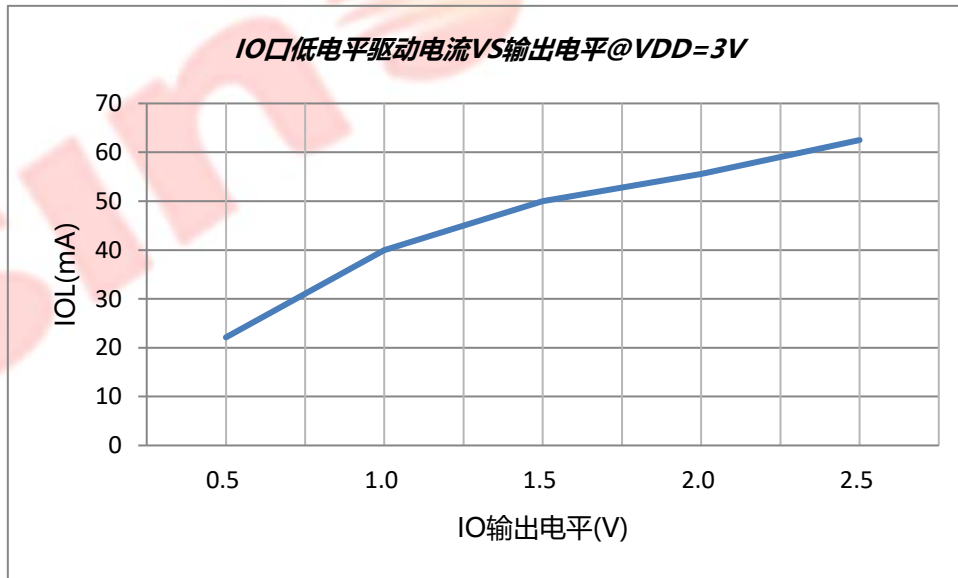
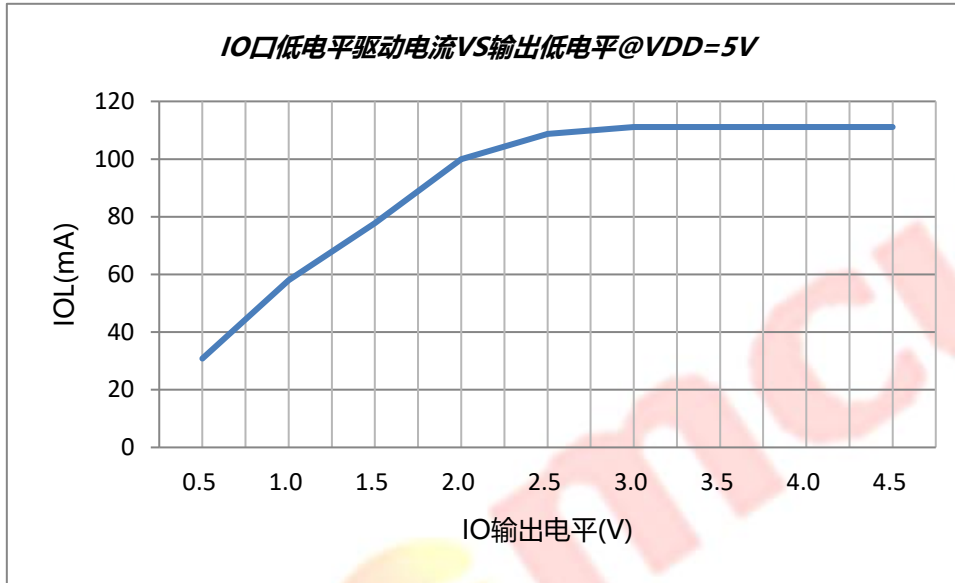
T=25°C

特性	符号	条件	最小	典型	最大	单位
外部高频晶振频率	F <sub>hosc</sub>		432K		16M	Hz
外部低频晶振频率	F <sub>losc</sub>			32768		Hz
内部高频 RC 振荡频率	F <sub>hirc1</sub>	T=25°C VDD=5V	-2%	16	+2%	MHz
	F <sub>hirc2</sub>	T=25°C VDD=2~5.5V	-2%	16	+2%	MHz
	F <sub>hirc3</sub>	T=-40°C~85°C VDD=5V	-5%	16	+5%	MHz
	F <sub>hirc4</sub>	T=-40°C~85°C VDD=2~5.5V	-5%	16	+5%	MHz
内部低频振荡器频率	F <sub>lirc</sub>	T=25°C VDD=5V	-20%	28	+20%	KHz
外部振荡器起振时间	T <sub>oxov</sub>				20	ms

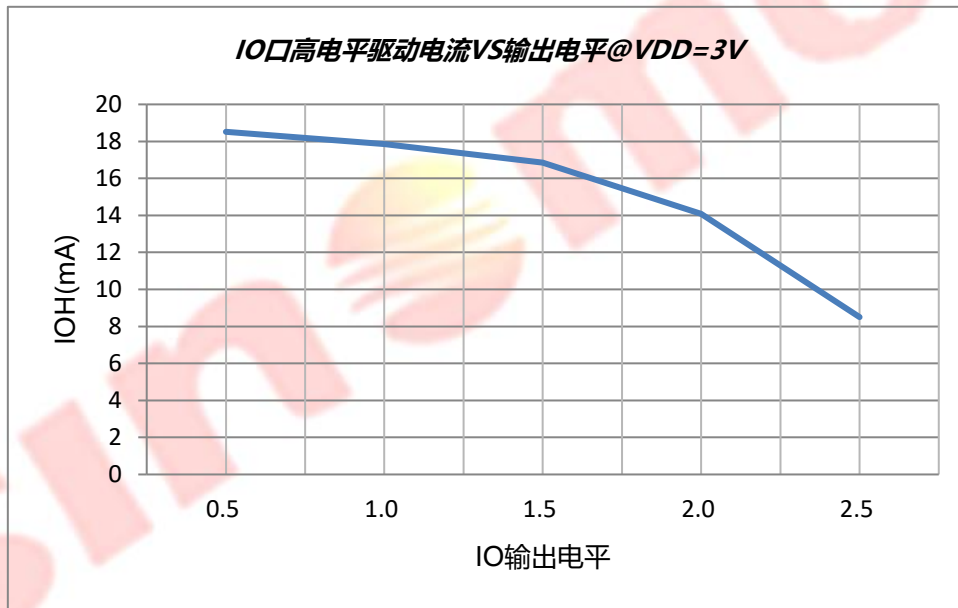
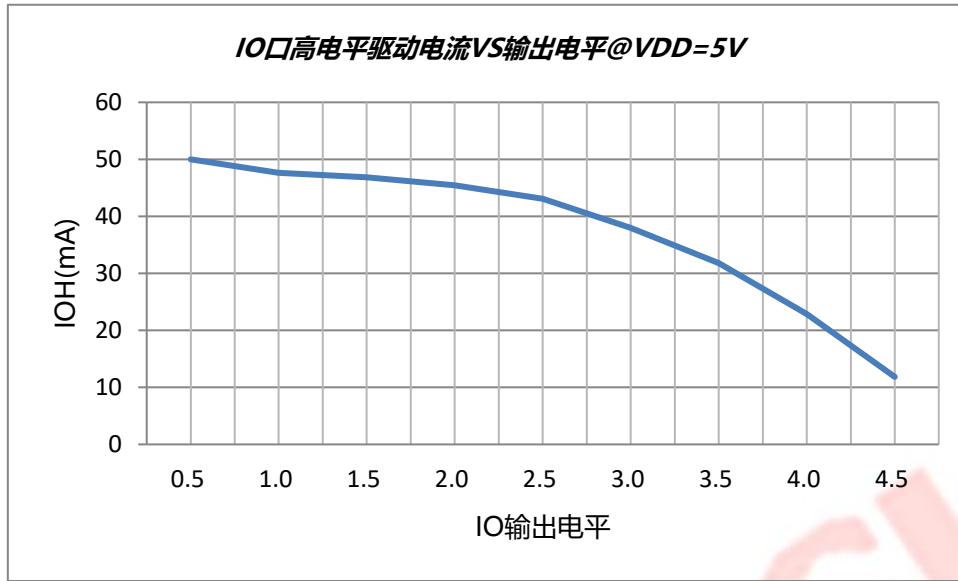
## 12 特性曲线图

注：本节列出的特性曲线图仅作为设计参考，部分数据可能超出芯片额定的工作条件范围，为保证芯片能正常工作，请严格按照电气特性说明。

### 12.1 IO 口低电平驱动电流 VS 输出电平

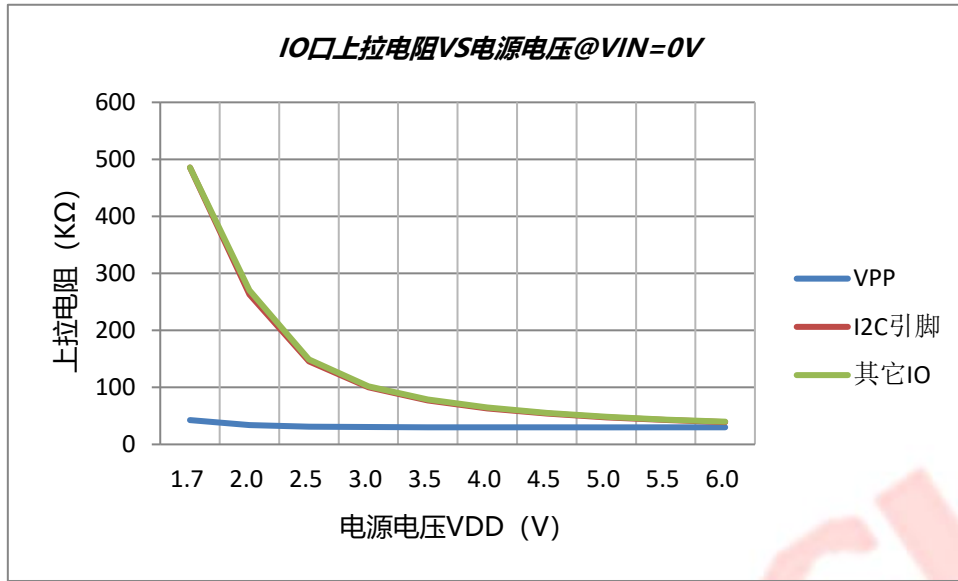


## 12.2 IO 口高电平驱动电流 VS 输出电平

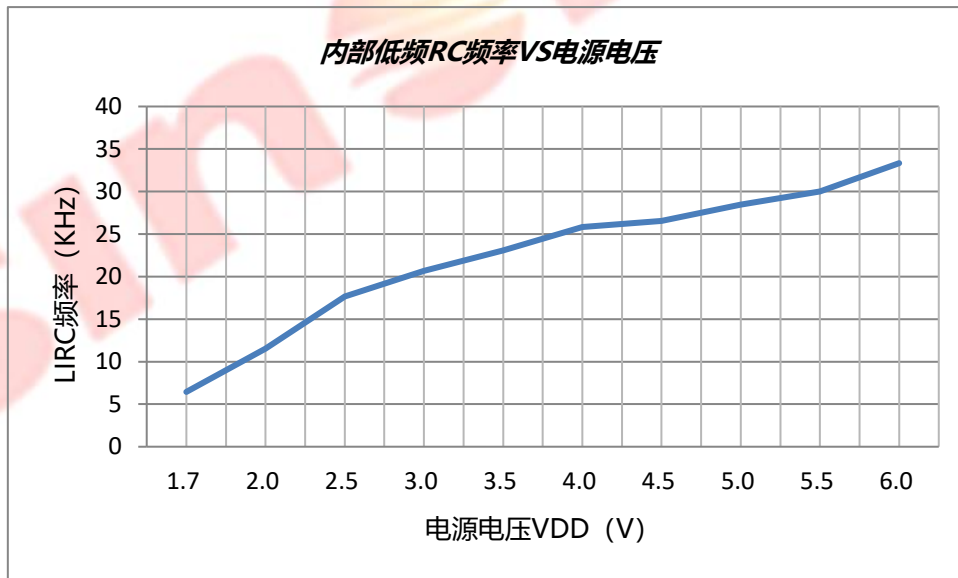




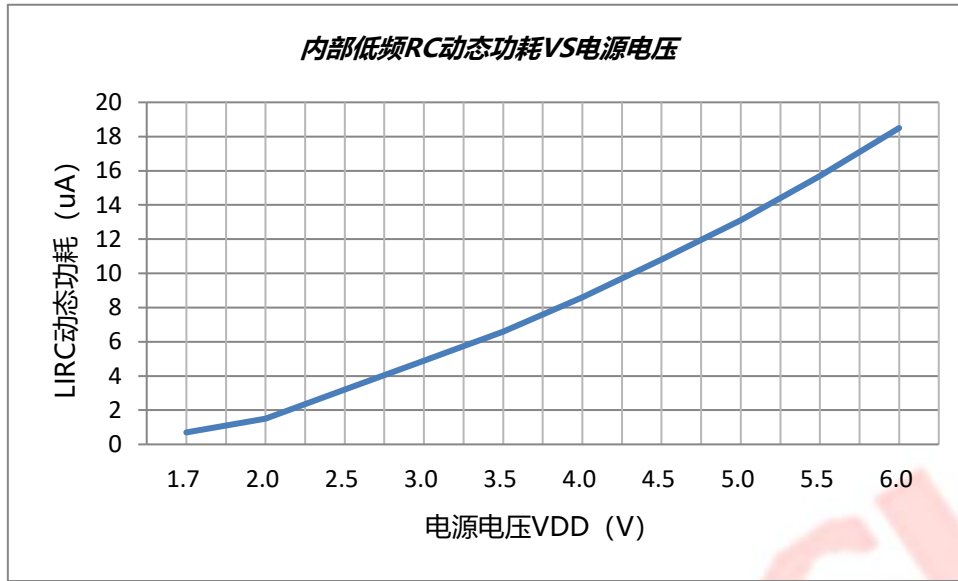
### 12.3 IO 口上拉电阻 VS 电源电压



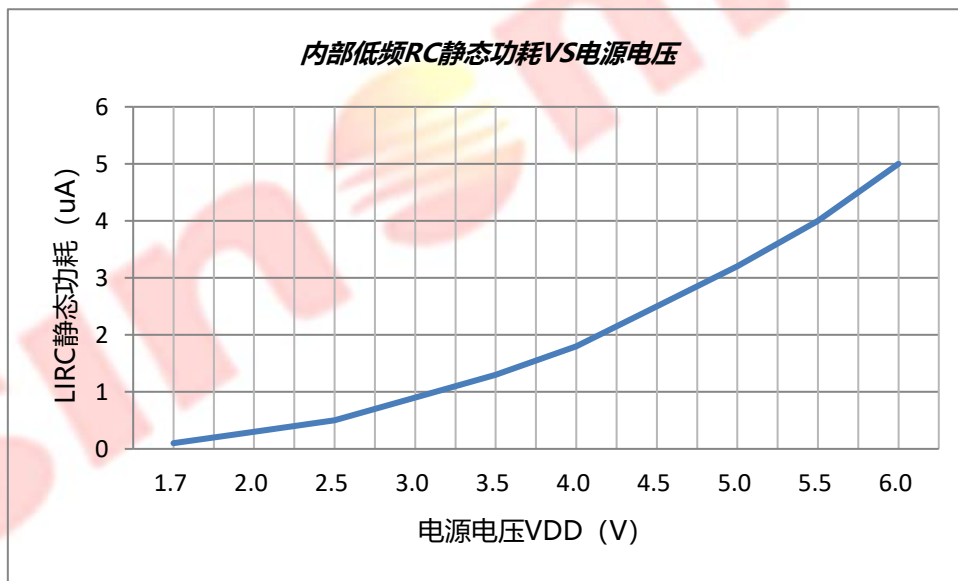
### 12.4 内部低频 RC 频率 VS 电源电压



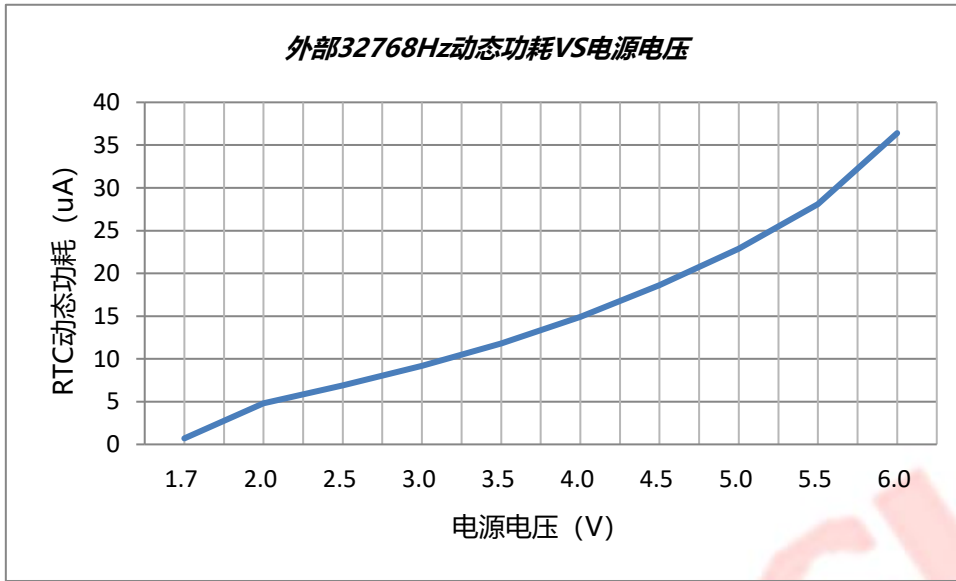
### 12.5 内部低频 RC 动态功耗 VS 电源电压



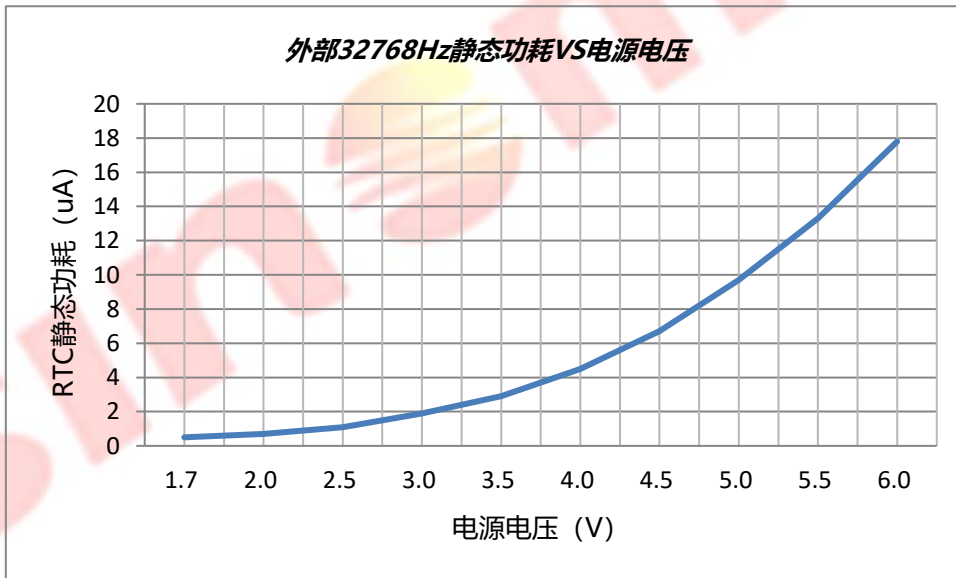
### 12.6 内部低频 RC 静态功耗 VS 电源电压



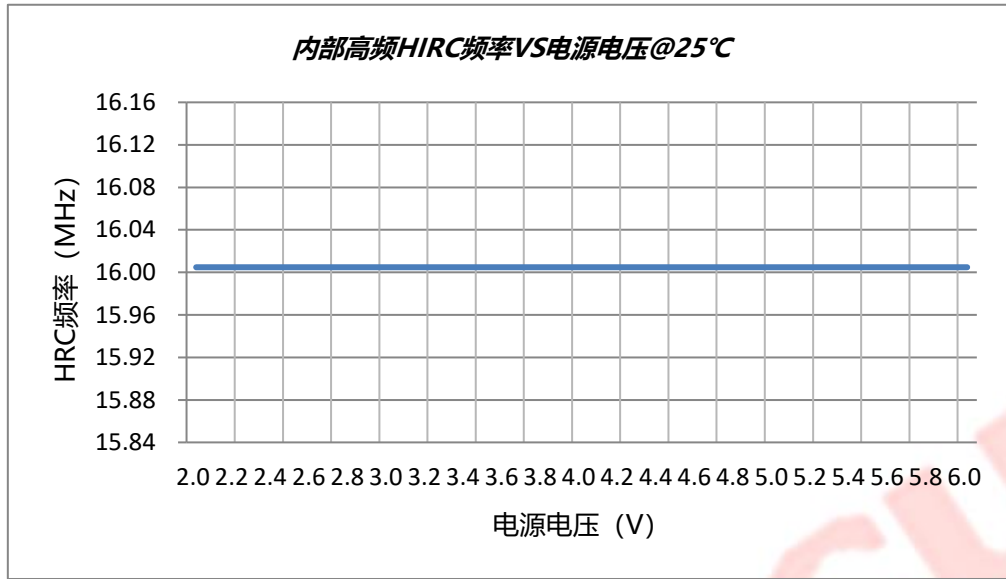
### 12.7 外部 32768Hz 动态功耗 VS 电源电压



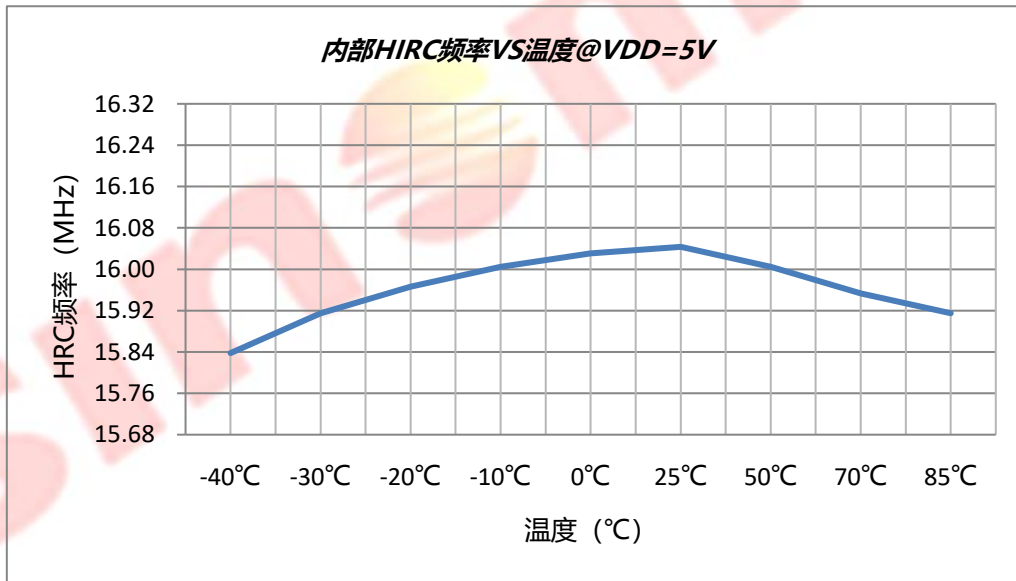
### 12.8 外部 32768Hz 静态功耗 VS 电源电压



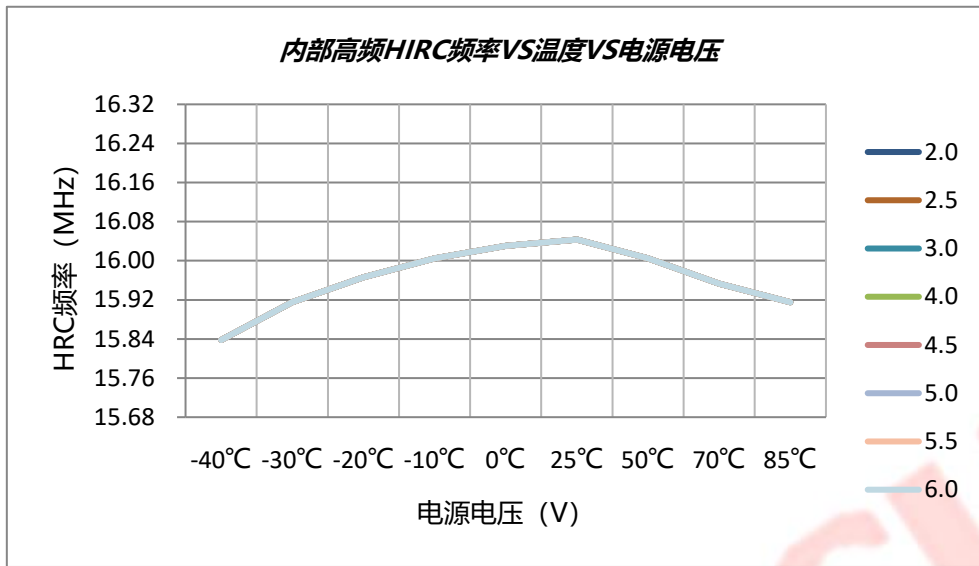
### 12.9 常温内部高频 HIRC 电压频率 VS 电源电压



### 12.10 常压内部高频 HIRC 电压频率 VS 温度



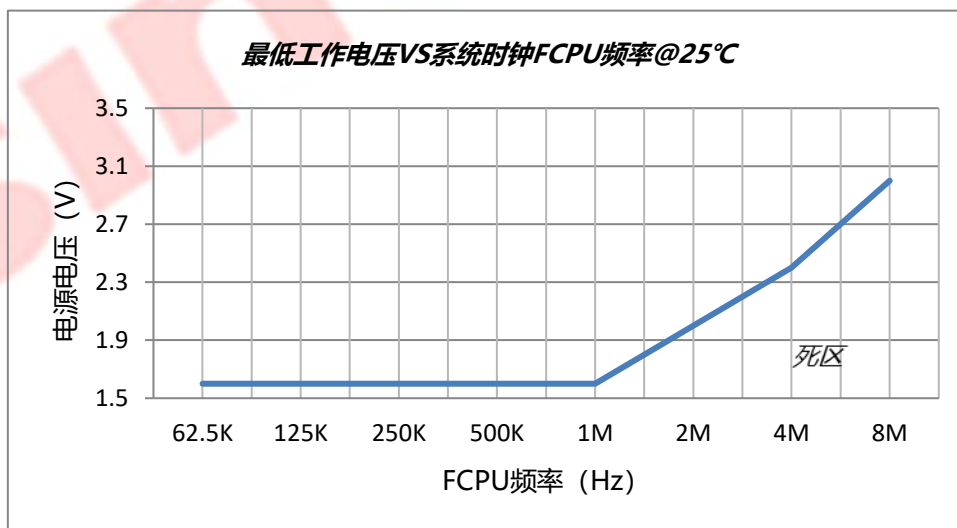
### 12.11 内部高频 HIRC 频率 VS 温度 VS 电源电压



### 12.12 最低工作电压 VS 系统时钟 Fcpu 关系图

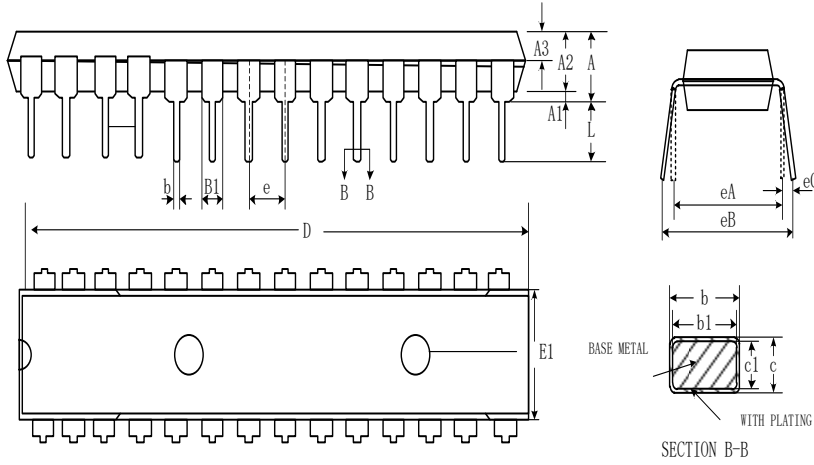
注：系统最低工作电压和系统工作频率  $F_{cpu}$  有关，不同的工作频率  $F_{cpu}$  最低工作电压不同。

如下图所示，当工作频率提高时系统正常工作电压也随之提高，但由于 **POR** 电压固定 ( $1.6V@25^{\circ}C$ )，在系统最低工作电压和 **POR** 电压之间就会出现一个电压区域，系统不能正常工作也不会产生 **POR** 复位，称之为死区，必须根据不同的工作频率设置合适的 **LVR** 电压避免出现死区。



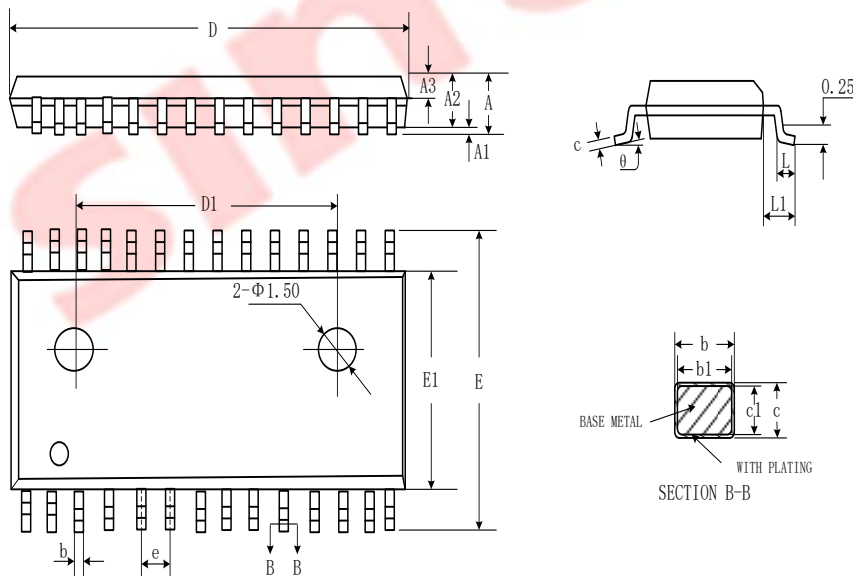
13 封装外形尺寸

**SKDIP28(SKY28)**



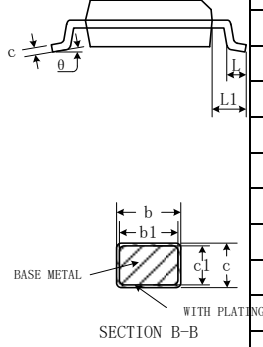
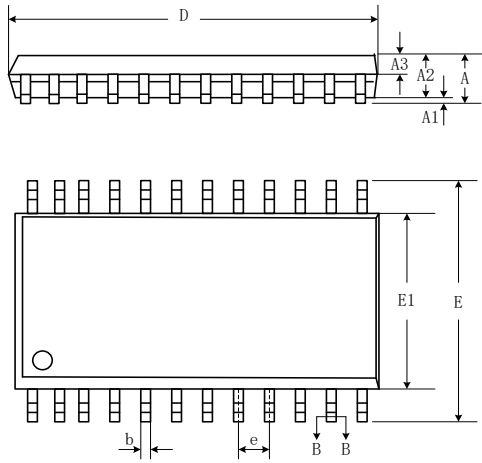
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	3.50	3.70	3.90
A1	0.42	--	--
A2	3.20	3.30	3.40
A3	1.50	1.55	1.60
b	0.44	--	0.53
b1	0.43	0.46	0.48
B1	1.52BSC		
c	0.25	--	0.31
c1	0.24	0.25	0.26
D	35.10	35.30	35.50
E1	7.10	7.30	7.50
e	2.54BSC		
Ea	7.87BSC		
Eb	7.87	--	9.50
Ec	0	--	0.82
L	3.20	--	--

**SOP28**



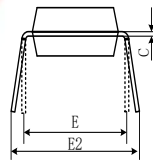
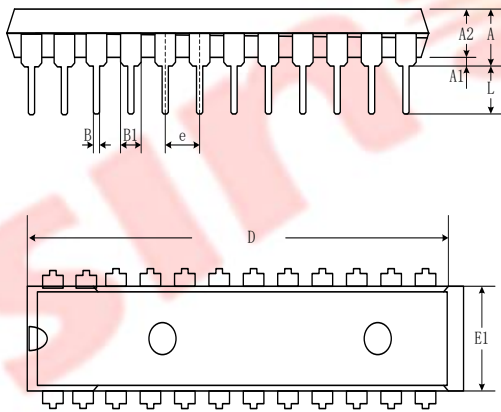
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	-	-	2.65
A1	0.10	-	0.30
A2	2.25	2.30	2.35
A3	0.97	1.02	1.07
b	0.39	-	0.48
b1	0.38	0.41	0.43
c	0.25	-	0.31
c1	0.24	0.25	0.26
D	17.80	18.00	18.20
D1	11.90	12.00	12.10
E	10.10	10.30	10.50
E1	7.30	7.50	7.70
e	1.27BSC		
L	0.70	-	1.00
L1	1.40BSC		
$\theta$	0	-	8°

**SOP24**



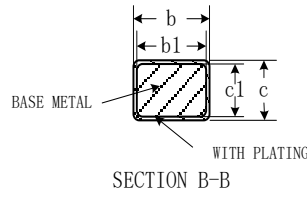
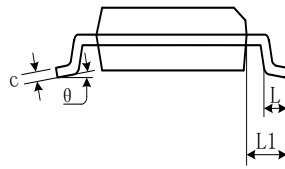
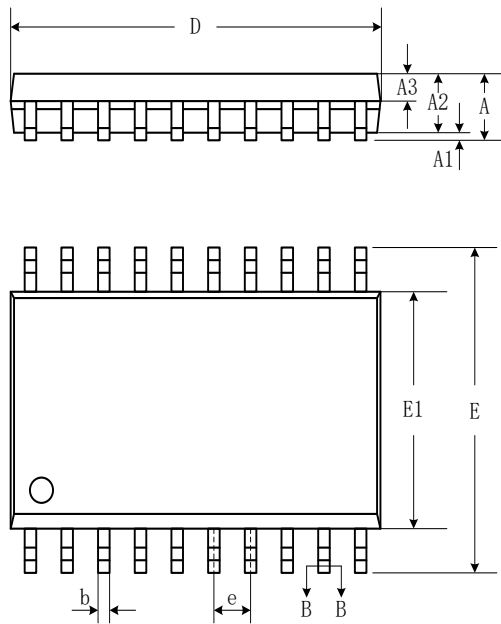
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	2.36	2.54	2.64
A1	0.10	0.20	0.30
A2	2.26	2.30	2.35
A3	0.97	1.02	1.07
b	0.39	-	0.48
b1	0.38	0.41	0.43
c	0.25	-	0.31
c1	0.24	0.25	0.26
D	15.20	15.40	15.60
E	10.10	10.30	10.50
E1	7.40	7.50	7.60
e	1.27BSC		
L	0.70	-	1.00
L1	1.40BSC		
h	0.25	-	0.75
$\theta$	0	-	8°

**DIP24**



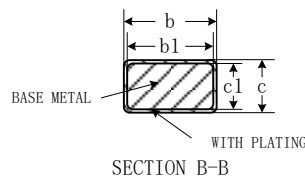
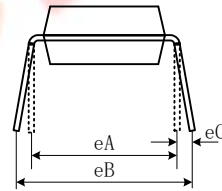
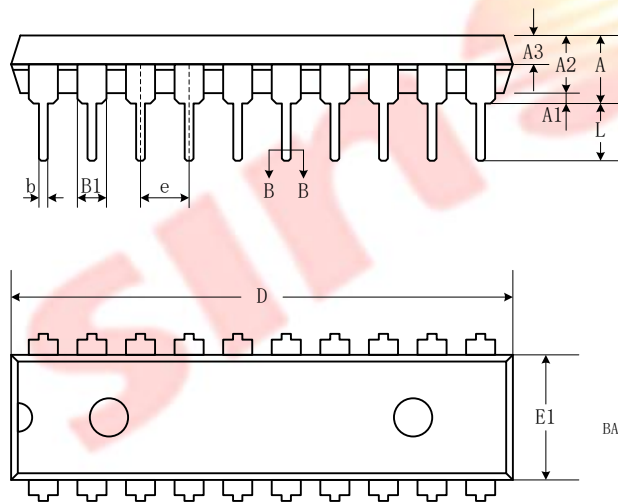
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	--	--	5.33
A1	0.38	--	--
A2	3.17	3.30	3.43
D	31.24	31.75	32.51
E	7.62 BSC		
E1	6.42	6.55	6.68
E2	8.51	9.02	9.53
L	2.92	3.30	3.81
e	2.54		
B	0.25		
B1	1.52		

**SOP20**



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	-	-	2.70
A1	0.10	0.20	0.30
A2	2.10	2.30	2.50
A3	0.92	1.02	1.12
b	0.35	-	0.44
b1	0.34	0.37	0.39
c	0.26	-	0.31
c1	0.24	0.25	0.26
D	12.60	12.80	13.00
E	10.10	10.30	10.50
E1	7.30	7.50	7.70
e	1.27BSC		
L	0.70	0.85	1.00
L1	1.40BSC		
$\theta$	0	-	8°

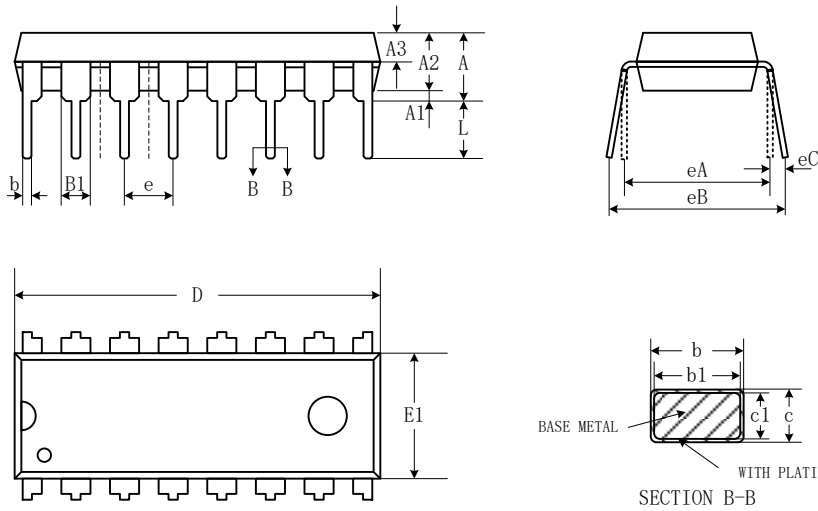
**DIP20**



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	3.60	3.80	4.00
A1	0.51	-	-
A2	3.10	3.30	3.50
A3	1.42	1.52	1.62
b	0.44	-	0.53
b1	0.43	0.46	0.48
B1	1.52BSC		
c	0.25	-	0.31
c1	0.24	0.25	0.26
D	26.03	26.23	26.43
E1	6.35	6.55	6.75
e	2.54BSC		
Ea	7.62BSC		
Eb	7.62	-	9.50
Ec	0	-	0.94
L	3.00	-	-

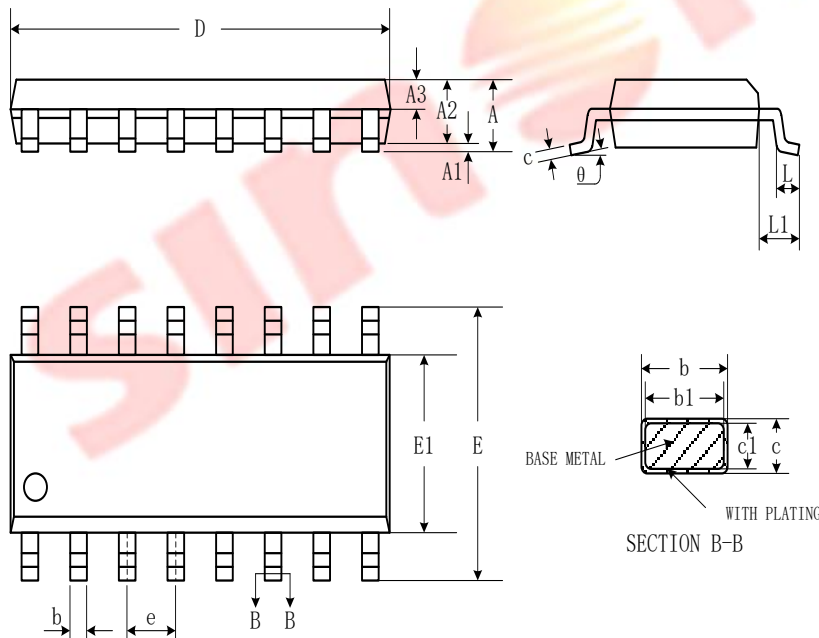


**DIP16**



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	3.60	3.80	4.00
A1	0.51	-	-
A2	3.10	3.30	3.50
A3	1.42	1.52	1.62
b	0.44	-	0.53
b1	0.43	0.46	0.48
B1	1.52BSC		
c	0.25	-	0.31
c1	0.24	0.25	0.26
D	18.90	19.10	19.30
E1	6.15	6.35	6.55
e	2.54BSC		
Ea	7.62BSC		
Eb	7.62	-	9.50
Ec	0	-	0.94
L	3.00	-	-

**SOP16**



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	-	-	1.77
A1	0.08	0.18	0.28
A2	1.20	1.40	1.60
A3	0.55	0.65	0.75
b	0.39	-	0.48
b1	0.38	0.41	0.43
c	0.21	-	0.26
c1	0.19	0.20	0.21
D	9.70	9.90	10.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
L	0.50	0.65	0.80
L1	1.05BSC		
theta	0	-	8°

## 14 版本修订记录

版本号	修订日期	修订内容
V1.0	2013-10-12	新建
V1.1	2013-11-13	修改§7.3 中 BIT[5:4]低通滤波选择位的描述。
V1.2	2014-02-18	(1) 用户手册名称由 MC32P64 变更为 MC32P8112。 (2) §2.1 增加指令汇总表。 (3) §9.1 增加 ADC 工作电压的说明。 (4) §11.3 增加 “ADC 有效工作电压”。
V1.3	2014-04-02	(1) §1.3 增加 SOP16 引脚排列。
V1.4	2014-04-23	(1) §1.3 修改 DIP16 引脚排列。 (2) 说明 SKDIP28 与 SKY28 封装形式相同。
V1.5	2014-06-18	(1) §1.1 增加订购信息。 (2) §1.3 修改 DIP16 引脚排列。 (3) §1.3 增加 MC32P64A1K SOP16 引脚排列。
V1.6	2016-01-08	(1) 增加 DIP24 和 SOP20 引脚排列图及封装尺寸图。
V1.7	2016-10-12	(1) §2.5 修改 WDTT、VLVRS 配置项； (2) §11.4 修改 HIRC 电气参数；
V1.8	2018-05-10	(1) 修改§2.3 的数据存储器映射表中 SFR 名称的笔误