

SinoMCU 8 位单片机

MC2722

用户手册

V1.0



sinomcu
晟矽微电子

上海晟矽微电子股份有限公司

Shanghai SinoMCU Microelectronics Co., Ltd.

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

目录

| | | |
|------|--------------------|----|
| 1 | 产品简介 | 4 |
| 1.1 | 产品特性 | 4 |
| 1.2 | 订购信息 | 5 |
| 1.3 | 引脚排列 | 6 |
| 1.4 | 引脚说明 | 7 |
| 2 | CPU 及存储器 | 8 |
| 2.1 | 指令集 | 8 |
| 2.2 | 程序存储器 | 10 |
| 2.3 | 数据存储器 | 11 |
| 2.4 | 堆栈 | 12 |
| 2.5 | 用户配置字 | 13 |
| 2.6 | 控制寄存器 | 15 |
| 3 | 系统时钟 | 18 |
| 3.1 | 外接晶体振荡器 | 18 |
| 3.2 | 外部 RC 振荡器 | 18 |
| 3.3 | 外部时钟输入 | 18 |
| 3.4 | 内置高频 RC 振荡器 | 19 |
| 3.5 | 内置低频 RC 振荡器 | 19 |
| 3.6 | 系统工作模式 | 19 |
| 3.7 | 高速模式 | 21 |
| 3.8 | 低速模式 | 21 |
| 3.9 | 休眠模式 | 21 |
| 3.10 | 空闲模式 | 22 |
| 4 | 复位 | 23 |
| 4.1 | 复位条件 | 23 |
| 4.2 | 上电复位 | 23 |
| 4.3 | 外部复位 | 23 |
| 4.4 | 低电压复位 | 23 |
| 4.5 | 看门狗复位 | 24 |
| 5 | I/O 端口 | 25 |
| 5.1 | IO 工作模式 | 25 |
| 5.2 | 上拉电阻控制 | 26 |
| 5.3 | 端口模式控制 | 27 |
| 5.4 | 端口电平唤醒功能 | 27 |
| 6 | 定时器 | 28 |
| 6.1 | 看门狗定时器 WDT | 28 |
| 6.2 | 定时器 T0 | 28 |
| 6.3 | 定时器 T1 | 29 |
| 7 | 蜂鸣器输出 | 32 |
| 7.1 | BUZZER 概述 | 32 |
| 7.2 | BUZZER 相关寄存器 | 32 |
| 8 | 模数转换器 | 33 |

| | | |
|------|---------------------------|----|
| 8.1 | ADC 概述..... | 33 |
| 8.2 | ADC 转换时序图..... | 33 |
| 8.3 | ADC 操作步骤..... | 33 |
| 8.4 | ADC 相关寄存器..... | 34 |
| 9 | 中断..... | 36 |
| 9.1 | 外部中断..... | 36 |
| 9.2 | 定时器中断..... | 36 |
| 9.3 | ADC 转换中断..... | 37 |
| 9.4 | 中断相关寄存器..... | 37 |
| 10 | 电气参数..... | 39 |
| 10.1 | 极限参数..... | 39 |
| 10.2 | 直流特性参数..... | 39 |
| 10.3 | ADC 特性参数..... | 40 |
| 10.4 | 交流电气参数..... | 40 |
| 11 | 特性曲线图..... | 42 |
| 11.1 | 输入施密特电压 VS 电源电压..... | 42 |
| 11.2 | 上拉电阻 VS 电源电压..... | 43 |
| 11.3 | 驱动电流 VS 端口电压..... | 44 |
| 11.4 | 内部高频 RC 动态功耗 VS 电源电压..... | 46 |
| 11.5 | 外部 RC 时钟源 VS 电源电压..... | 49 |
| 11.6 | 外部晶振功耗 VS 电源电压..... | 52 |
| 11.7 | 内部低频 RC VS 电源电压..... | 53 |
| 11.8 | 静态功耗 VS 温度..... | 55 |
| 11.9 | 内部高频 RC VS 电源电压/温度..... | 56 |
| 12 | 封装外形尺寸..... | 58 |
| 13 | 版本修订记录..... | 59 |

1 产品简介

1.1 产品特性

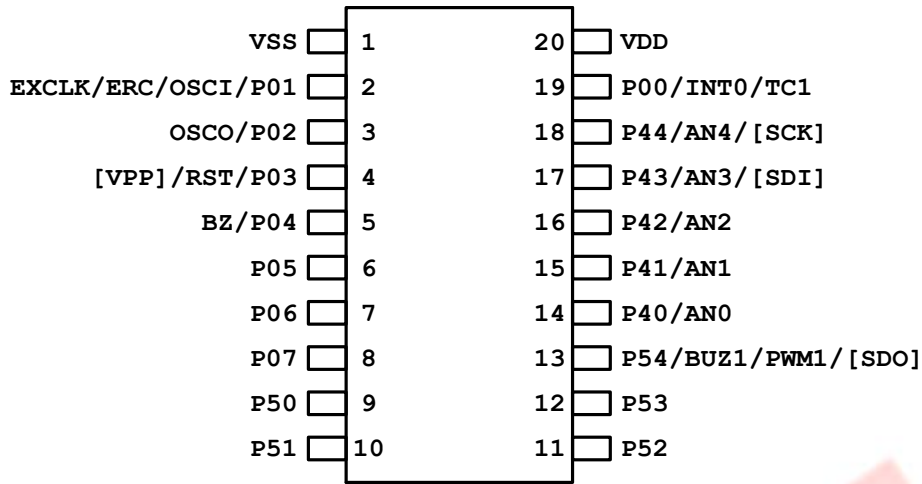
- ◇ 8 位 CPU 内核
 - ✓ 精简指令集
 - ✓ 8 级深度硬件堆栈
 - ✓ 高频模式下 2T/4T/8T/16T/32T/64T/128T/256T 可设；低频工作模式下为 4T
- ◇ 程序存储器
 - ✓ 2K*16 位 OTP 型程序存储器，烧写 1 次
 - ✓ 1K*16 位 OTP 型程序存储器，烧写 2 次
 - ✓ 可通过间接寻址读取存储器内容
- ◇ 数据存储器
 - ✓ 128 字节 SRAM 通用数据存储器
 - ✓ 支持直接寻址、间接寻址等多种寻址方式
- ◇ 3 组共 18 个 IO
 - ✓ P00~P07、P40~P44、P50~P54
 - ✓ P03 为输入口，可复用于编程高压 VPP 输入
 - ✓ 部分端口可复用为外部中断输入、ADC 模拟输入、PWM 输出、BUZ 输出等功能
 - ✓ 端口输出低电平时的驱动电流，可选择 16mA、25mA
 - ✓ 除 P03 外其他端口均内置上拉电阻，并可单独使能/禁止
 - ✓ P00~P07 支持端口电平变化唤醒功能
- ◇ 时钟振荡模式
 - ✓ 内嵌高频振荡器（16MHz）+ 内嵌低频振荡器（32KHz）
 - ✓ 外接高频晶体振荡器 + 内嵌低频振荡器（32KHz）
 - ✓ 外接 RC 振荡器 + 内嵌低频振荡器（32KHz）
 - ✓ 外部时钟输入 + 内嵌低频振荡器（32KHz）
- ◇ 4 种工作模式
 - ✓ 高速运行模式：CPU 在高频时钟下运行
 - ✓ 低速运行模式：CPU 在低频时钟下运行
 - ✓ 空闲模式：CPU 停止运行，高频振荡器可选停止或工作，低频振荡器工作
 - ✓ 休眠模式：所有振荡器停止运行
- ◇ 内部自振式看门狗计数器（WDT）
 - ✓ 溢出时间 = 8192/内部低频 RC 振荡器频率（FLIRC），约为 256ms@VDD=5V
 - ✓ 可配置成始终开启、始终关闭、低功耗模式下停止等多种工作模式
- ◇ 定时器
 - ✓ 1 个 8 位定时器 T0，可设置溢出中断，支持溢出唤醒空闲模式
 - ✓ 1 个 8 位定时/计数器 T1，可用于定时/外部计数/PWM 输出/BUZ 输出，可设置溢出中断
- ◇ 1 路 BUZZER 输出
 - ✓ 占空比固定为 50%
 - ✓ 输出频率可设
- ◇ 1 个 12 位高精度 ADC
 - ✓ 5 路外部通道：AN0~AN4

- ✓ 参考电压: VDD
- ✓ 可配置 ADC 时钟: F_{CPU} 经 4/8/32/64 分频
- ✧ 外部中断
 - ✓ 1 路外部中断: INT0, 可设为上升沿/下降沿/双沿中断
- ✧ 中断
 - ✓ 外部中断 (INT0)
 - ✓ 定时器中断 (T0、T1)
 - ✓ ADC 中断
- ✧ 4 级低电压复位 LVR
 - ✓ 2.2V/2.7V/3.2V/3.6V
- ✧ 2 级低电压检测 LVD
 - ✓ 2.4V/3.6V
- ✧ 工作电压
 - ✓ V_{LVR32} ~ 5.5V (F_{cpu} = 0~8MHz)
 - ✓ V_{LVR27} ~ 5.5V (F_{cpu} = 0~4MHz)
 - ✓ V_{LVR22} ~ 5.5V (F_{cpu} = 0~2MHz)
 - ✓ V_{LVR22} ~ 5.5V (F_{cpu} = 0~1MHz)
 - ✓ V_{LVR22} ~ 5.5V (F_{cpu} = 0~455KHz/2)
 - ✓ V_{LVR22} ~ 5.5V (F_{cpu} = 0~32KHz/4)
- ✧ 封装形式
 - ✓ SOP20、SOP16、SOP14

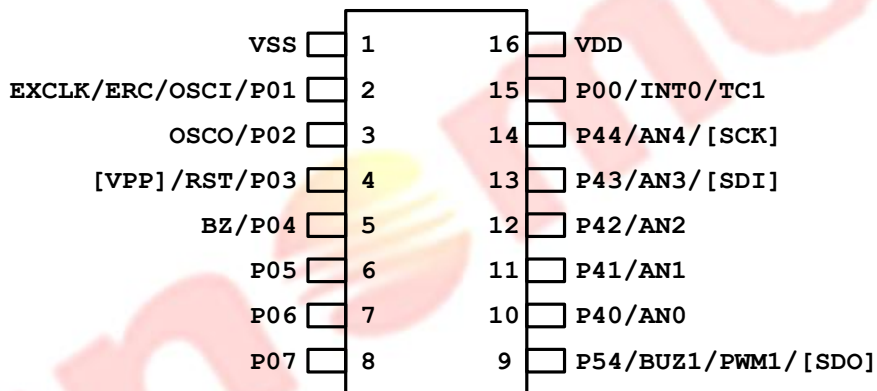
1.2 订购信息

| 产品名称 | 封装形式 | 特性 |
|-----------|-------|----|
| MC2722A0M | SOP20 | |
| MC2722A0K | SOP16 | |
| MC2722A0J | SOP14 | |

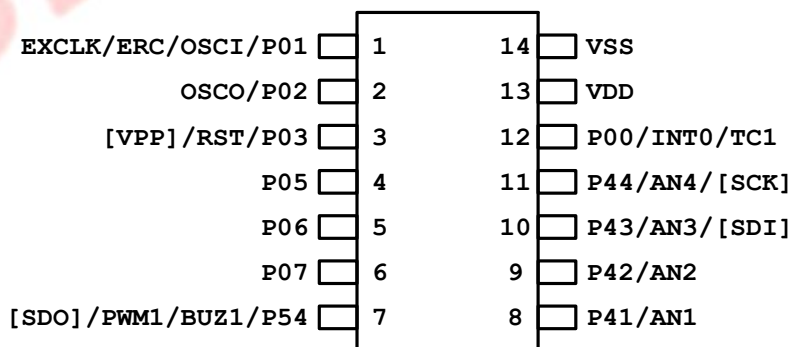
1.3 引脚排列



SOP20



SOP16



SOP14

1.4 引脚说明

| 引脚名称 | 类型 | 功能说明 |
|----------------------|-----|---------------------|
| VDD | P | 电源 |
| VSS | P | 地 |
| OSCI、OSCO | I/O | 外部晶体振荡器输入/输出 |
| ERC, EXCLK | I | 外部 RC 振荡器输入, 外部时钟输入 |
| P0, P4, P5, (P03 除外) | I/O | GPIO, 内部上拉 |
| P03 | I | 输入端口 |
| INT0 | I | 外部中断输入 |
| TC1 | I | T1 外部输入 |
| PWM1/BUZ1 | O | T1 产生的 PWM1/BUZ1 输出 |
| BZ | O | BUZZER 输出 |
| AN0~AN4 | A | ADC 模拟通道 0~4 |
| RST | I | 外部复位输入 |
| SCK、SDI、SDO | I/O | 编程时钟/数据输入/数据输出 |
| VPP | P | 编程高压输入 |

2 CPU及存储器

2.1 指令集

本芯片指令集为精简指令集。所有指令均为单字指令，即指令码只占用 1 个程序存储器地址空间。

指令汇总表

| 助记符 | 说明 | 操作 | 周期 | 标志 |
|----------|-----------------------------------|---------------------------------------------------------------------------------|-----|--------|
| ADDAR R | 寄存器 R 内容和 ACC 相加，结果存到 ACC | $R+ACC \rightarrow ACC$ | 1 | C,DC,Z |
| ADDRA R | 寄存器 R 内容和 ACC 相加，结果存到 R | $R+ACC \rightarrow R$ | 1+M | C,DC,Z |
| ADCAR R | 带 C 标志的加法，结果存到 ACC | $R+ACC+C \rightarrow ACC$ | 1 | C,DC,Z |
| ADCRA R | 带 C 标志的加法，结果存到 R | $R+ACC+C \rightarrow R$ | 1+M | C,DC,Z |
| ASUBAR R | ACC 和寄存器 R 内容相减，结果存到 ACC | $ACC-R \rightarrow ACC$ | 1 | C,DC,Z |
| ASUBRA R | ACC 和寄存器 R 内容相减，结果存到 R | $ACC-R \rightarrow R$ | 1+M | C,DC,Z |
| ASBCAR R | ACC 和寄存器 R 内容相减(带 C 标志)，结果存到 ACC | $ACC-R-/C \rightarrow ACC$ | 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 | $ACC-R-/C \rightarrow R$ | 1+M | C,DC,Z |
| MULAR R | 寄存器 R 内容和 ACC 相乘，结果存到[HIBYTE:ACC] | $R*ACC \rightarrow [HIBYTE:ACC]$ | 1 | - |
| ANDAR R | 寄存器 R 内容和 ACC 与操作，结果存到 ACC | $R \text{ and } ACC \rightarrow ACC$ | 1 | Z |
| ANDRA R | 寄存器 R 内容和 ACC 与操作，结果存到 R | $R \text{ and } ACC \rightarrow R$ | 1+M | Z |
| ORAR R | 寄存器 R 内容和 ACC 或操作，结果存到 ACC | $R \text{ or } ACC \rightarrow ACC$ | 1 | Z |
| ORRA R | 寄存器 R 内容和 ACC 或操作，结果存到 R | $R \text{ or } ACC \rightarrow R$ | 1+M | Z |
| XORAR R | 寄存器 R 内容和 ACC 异或操作，结果存到 ACC | $R \text{ xor } ACC \rightarrow ACC$ | 1 | Z |
| XORRA R | 寄存器 R 内容和 ACC 异或操作，结果存到 R | $R \text{ xor } ACC \rightarrow R$ | 1+M | Z |
| CLRR R | 对 R 清零 | $0 \rightarrow R$ | 1 | - |
| RLAR R | 寄存器 R 循环左移(带 C 标志)，结果存到 ACC | $R[7] \rightarrow C$ $R[6:0] \rightarrow ACC[7:1]$ $C \rightarrow ACC[0]$ | 1 | C |
| RLR R | 寄存器 R 循环左移(带 C 标志)，结果存到 R | $R[7] \rightarrow C$ $R[6:0] \rightarrow R[7:1]$ $C \rightarrow R[0]$ | 1+M | C |
| RRAR R | 寄存器 R 循环右移(带 C 标志)，结果存到 ACC | $C \rightarrow ACC[7]$ $R[7:1] \rightarrow ACC[6:0]$ $R[0] \rightarrow C$ | 1 | C |
| RRR R | 寄存器 R 循环右移(带 C 标志)，结果存到 R | $C \rightarrow R[7]$ $R[7:1] \rightarrow R[6:0]$ $R[0] \rightarrow C$ | 1+M | C |
| SWAPAR R | 交换 R 的高低字节，结果存到 ACC | $R[7:4] \rightarrow ACC[3:0]$ $R[3:0] \rightarrow ACC[7:4]$ | 1 | - |
| SWAPR R | 交换 R 的高低字节，结果存到 R | $R[7:4] \rightarrow R[3:0]$ $R[3:0] \rightarrow R[7:4]$ | 1+M | - |
| MOVAR R | 将 R 存到 ACC | $R \rightarrow ACC$ | 1 | Z |
| MOVRA R | 将 ACC 存到 R | $ACC \rightarrow R$ | 1 | - |

| | | | | | |
|--------|------|----------------------------------|--------------------------------------------------------|-------|--------|
| JZAR | R | R 加 1, 结果存到 ACC; 结果为 0, 则跳过下一条指令 | $R+1 \rightarrow ACC$, 结果为 0, 则 $PC+2 \rightarrow PC$ | 1+J | - |
| JZR | R | R 加 1, 结果存到 R; 结果为 0, 则跳过下一条指令 | $R+1 \rightarrow R$, 结果为 0, 则 $PC+2 \rightarrow PC$ | 1+M+J | - |
| DJZAR | R | R 减 1, 结果存到 ACC; 结果为 0, 则跳过下一条指令 | $R-1 \rightarrow ACC$, 结果为 0, 则 $PC+2 \rightarrow PC$ | 1+J | - |
| DJZR | R | R 减 1, 结果存到 R; 结果为 0, 则跳过下一条指令 | $R-1 \rightarrow R$, 结果为 0, 则 $PC+2 \rightarrow PC$ | 1+M+J | - |
| BCLR | R, b | 对 R 的第 b 位清零 | $0 \rightarrow R[b]$ | 1+M | - |
| BSET | R, b | 对 R 的第 b 位置 1 | $1 \rightarrow R[b]$ | 1+M | - |
| JBCLR | R, b | 如 R 的第 b 位为 0, 则跳过下一条指令 | 如 $R[b]=0$, 则 $PC+2 \rightarrow PC$ | 1+J | - |
| JBSET | R, b | 如 R 的第 b 位为 1, 则跳过下一条指令 | 如 $R[b]=1$, 则 $PC+2 \rightarrow PC$ | 1+J | - |
| ADDAI | K | 立即数 K 和 ACC 相加, 结果存到 ACC | $K+ACC \rightarrow ACC$ | 1 | C,DC,Z |
| ASUBAI | K | ACC 和立即数相减, 结果存到 ACC | $ACC-K \rightarrow ACC$ | 1 | C,DC,Z |
| ANDAI | K | 立即数 K 和 ACC 与操作, 结果存到 ACC | $K \text{ and } ACC \rightarrow ACC$ | 1 | Z |
| ORAI | K | 立即数 K 和 ACC 或操作, 结果存到 ACC | $K \text{ or } ACC \rightarrow ACC$ | 1 | Z |
| XORAI | K | 立即数和 ACC 异或, 结果存到 ACC | $K \text{ xor } ACC \rightarrow ACC$ | 1 | Z |
| MOVAI | K | 将立即数存到 ACC | $K \rightarrow ACC$ | 1 | - |
| RETURN | | 从子程序返回 | $TOS \rightarrow PC$ | 2 | - |
| RETIE | | 从中断返回 | $TOS \rightarrow PC$ $1 \rightarrow GIE$ | 2 | - |
| CALL | K | 子程序调用 | $PC+1 \rightarrow TOS$ $K \rightarrow PC$ | 2 | - |
| GOTO | K | 无条件跳转 | $K \rightarrow PC$ | 2 | - |
| NOP | | 空操作 | 空操作 | 1 | - |
| CMPI | K | ACC 与立即数比较; 如果相等, 则跳过下一条指令 | $A-K$, 结果若为 0 则 $PC+1 \rightarrow PC$ | 1+J | C, Z |
| CMPR | R | ACC 与 R 比较; 如果相等, 则跳过下一条指令 | $A-R$, 结果若为 0 则 $PC+1 \rightarrow PC$ | 1+J | C, Z |
| PUSH | | 暂存 ACC 和 C,DC,Z | $ACC \text{ 和 } C,DC,Z \rightarrow BUF$ | 1 | - |
| POP | | 恢复 ACC 和 C,DC,Z | $BUF \rightarrow ACC \text{ 和 } C,DC,Z$ | 1 | C,DC,Z |
| XCH | R | ACC 与 R 交换 | $ACC \leftrightarrow R$ | 1+M | - |
| MOVC | | 读取 ROM | $ROM[FSR1:FSR0] \rightarrow [HIBYTE:ACC]$ | 2 | - |

注:

- 1、条件跳转指令为真时, $J=1$, 否则 $J=0$; 目的寄存器为 RAM 时, $M=1$, 否则 $M=0$;
- 2、PUSH/POP 指令涉及的缓存器 BUF 仅有 1 层, 所以 PUSH/POP 必须成对使用, 否则会导致数据错误;

2.2 程序存储器

本芯片的程序存储器为 OTP ROM，可通过用户配置字选择程序存储器的地址空间范围：

2K*16BIT 的程序存储器空间，只能烧写 1 次，程序存储器空间（0000H~07FFH）

| |
|----------------------|
| 复位向量（0000H） |
| 通用程序区（0001H - 0007H） |
| 中断向量（0008H） |
| 通用程序区（0009H - 07FFH） |

1K*16BIT 的程序存储器空间，可烧写 2 次，程序存储器空间（0000H~03FFH）

| |
|----------------------|
| 复位向量（0000H） |
| 通用程序区（0001H - 0007H） |
| 中断向量（0008H） |
| 通用程序区（0009H - 03FFH） |

程序存储器可通过 MOVC 指令访问，如下例所示：

例：通过 MOVC 指令访问 FSR1*256+FSR0 指向的程序存储器内容，高 8 位存入数据存储器 11H 地址，低 8 位存入数据存储器 10H 地址

```

MOVAI    55H
MOVRA    FSR0           ; 将 55H 写入 FSR0
MOVAI    01H
MOVRA    FSR1           ; 将 01H 写入 FSR1
MOVC                    ; 读取 FSR1*256+FSR0 指向（0155H）程序存储器的内容，
                        ; 高 8 位缓存在 HIBYTE，低 8 位缓存在 ACC
MOVRA    10H           ; 低 8 位存入数据存储器 10H 地址
MOVAR    HIBYTE        ; 从 HIBYTE 读取高 8 位
MOVRA    11H           ; 高 8 位存入数据存储器 11H 地址
    
```

2.3 数据存储器

数据存储器分为通用数据存储器 GPR 和特殊功能寄存器 SFR，具体地址分配参照下表。GPR 和 SFR 可直接寻址或通过 INDF 间接寻址。

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 |
|-----------|-----|----------|--------|--------|--------|--------|--------|--------|--------|
| 000H~07FH | GPR | 通用数据存储器区 | | | | | | | |
| 080H~087H | SFR | | | HIBYTE | FSR0 | FSR1 | | PFLAG | |
| 088H~08FH | | | | | | | | | |
| 090H~097H | | | | | | | | | |
| 098H~09FH | | | | | | | | | |
| 0A0H~0A7H | | | | | | | | | |
| 0A8H~0AFH | | | | | | | | | ADIOS2 |
| 0B0H~0B7H | | | ADM | ADB | ADR | | | | |
| 0B8H~0BFH | | OEP0 | | | | | | | EINTCR |
| 0C0H~0C7H | | | | | | OEP4 | OEP5 | | |
| 0C8H~0CFH | | INTF | INTE | OSCM | | WDTCR | T1LDR | PCL | PCH |
| 0D0H~0D7H | | IOP0 | | | | IOP4 | IOP5 | | |
| 0D8H~0DFH | | T0CR | T0CNT | T1CR | T1CNT | BUZCR | | | MCR |
| 0E0H~0E7H | | PUP0 | | | | PUP4 | PUP5 | | INDF |
| 0E8H~0EFH | | | | | | | | | |
| 0F0H~0F7H | | STKR0L | STKR0H | STKR1L | STKR1H | STKR2L | STKR2H | STKR3L | STKR3H |
| 0F8H~0FFH | | STKR4L | STKR4H | STKR5L | STKR5H | STKR6L | STKR6H | STKR7L | STKR7H |

注：上表中灰色部分数据存储器地址未定义，读出数据不确定，写入数据可能会影响其他地址的内容。

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数据 55H 写入数据存储器 10H 地址

MOVA I 55H

MOVRA 10H

：把数据 55H 写入数据存储器 10H 地址

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

例：通过间接寻址模式把数据 55H 写入数据存储器 0010H 地址

MOVA I 00H

MOVRA FSR1

MOVA I 10H

MOVRA FSR0

MOVA I 55H

MOVRA INDF

：把数据 55H 写入 FSR1*256+FSR0 指向的数据存储器中

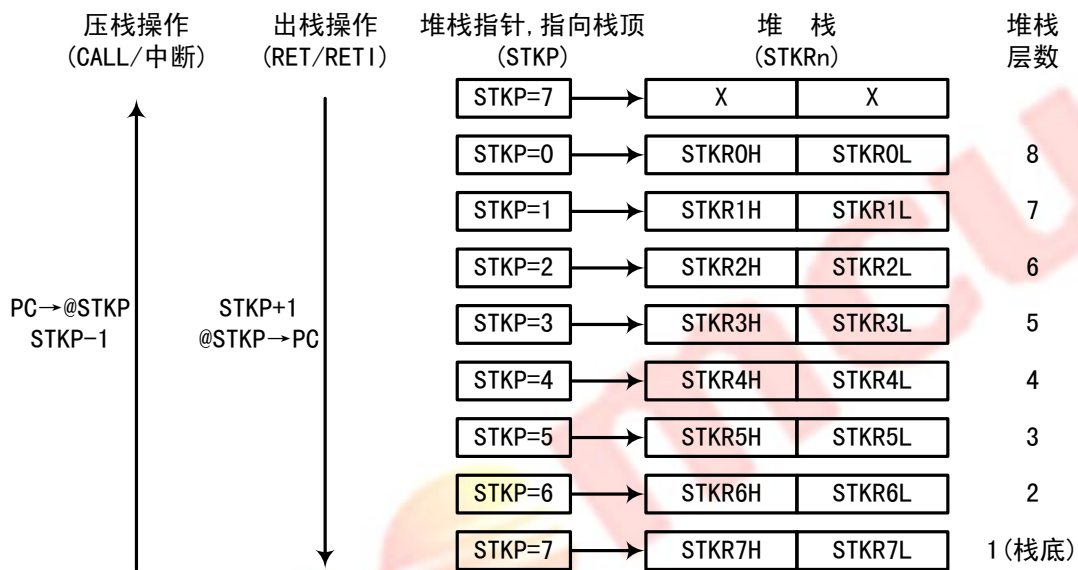
注：本间接寻址模式理论上可寻址 0000H~FFFFH，但用来访问本芯片数据存储器中未定义地址时，读出数据不确定，写入数据可能会更改其他地址中的数据。

2.4 堆栈

8 级 11 位堆栈缓存寄存器 STKRn (n=7-0) 位于特殊功能寄存器 SFR 区，用于暂存堆栈数据；3 位堆栈指针 STKP 位于杂凑寄存器 MCR 中，用于存放被访问堆栈单元的地址。堆栈操作遵循后进先出的原则，压栈时数据存入栈顶后 STKP-1，出栈时 STKP+1 后数据移出，从而 STKP 始终指向堆栈栈顶。

当程序响应中断或执行子程序调用指令时 CPU 会将 PC 自动压入栈顶；当执行中断返回指令或子程序返回指令时，堆栈数据移出至 PC。

堆栈操作示意



注：本芯片支持 8 层堆栈，若在堆栈满时压栈或堆栈空时出栈，会导致数据错误。

2.5 用户配置字

芯片为保证系统正常工作，会将关键模块的配置信息预先存储于单独的存储器区域中，在上电或其他复位发生后将配置信息载入寄存器中，通过寄存器配置关键模块的工作状态。该部分存储器中用户可选的内容即为用户配置字，可在烧录用户程序代码时进行配置与烧录。

| 符号 | 功能说明 |
|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PAGE | OTP 容量选择： OTP 模式； MTP 模式，第一次烧录； MTP 模式，第二次烧录； |
| WDTM | WDT 工作模式选择： 始终关闭看门狗； 低功耗模式下关闭看门狗； 始终开启看门狗； |
| VLVRS | LVR 电压选择： 2.2V；2.7V；3.2V；3.6V； |
| RSTEN | 外部复位使能： 屏蔽外部复位功能，端口用于其他功能 使能外部复位功能，端口用于外部复位 |
| ENCR | 代码加密选择： 使能代码加密； 不使能代码加密； |
| ENHANCE | 固定写“0”； |
| FOSCS HMLSEL FILTER XTDRVB | 高速时钟源选择： 内部高频 RC 振荡器 HIRC； 外部晶体振荡器 32768Hz； 外部晶体振荡器 455KHz； 外部晶体振荡器 1MHz； 外部晶体振荡器 2MHz； 外部晶体振荡器 4MHz； 外部晶体振荡器 8MHz； 外部晶体振荡器 16MHz； 外部时钟输入 ECK (<1MHz)； 外部时钟输入 ECK (<2MHz)； 外部时钟输入 ECK (<4MHz)； 外部 RC 振荡器 ERC (<1MHz)； 外部 RC 振荡器 ERC (<2MHz)； 外部 RC 振荡器 ERC (<4MHz)； 外部 RC 振荡器 ERC (<8MHz)； |
| FCPUS | 高速模式 CPU 频率选择： 指令周期为 2 个高速时钟周期， $F_{CPU}=F_{HOSC}/2$ ； 指令周期为 4 个高速时钟周期， $F_{CPU}=F_{HOSC}/4$ ； |

| | |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | 指令周期为 8 个高速时钟周期, $F_{CPU}=F_{HOSC}/8$; 指令周期为 16 个高速时钟周期, $F_{CPU}=F_{HOSC}/16$; 指令周期为 32 个高速时钟周期, $F_{CPU}=F_{HOSC}/32$; 指令周期为 64 个高速时钟周期, $F_{CPU}=F_{HOSC}/64$; 指令周期为 128 个高速时钟周期, $F_{CPU}=F_{HOSC}/128$; 指令周期为 256 个高速时钟周期, $F_{CPU}=F_{HOSC}/256$; |
| DRVS | IO 端口驱动能力选择 : 正常驱动 ; 增强驱动 ; |
| VLVDS | LVD 电压选择 : 2.4V ; 3.6V ; |

注 :

- 1、 配置不同的高速时钟源, 芯片最小工作电压会有不同, 小于最小工作电压的 LVR 电压将不可选 ;

2.6 控制寄存器

间接寻址寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| INDF | INDF7 | INDF6 | INDF5 | INDF4 | INDF3 | INDF2 | INDF1 | INDF0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | X | X | X | X | X | X | X | X |

BIT[7:0] **INDF[7:0]** – 间接寻址寄存器

INDF: INDF 不是物理寄存器，对 INDF 寻址实际上是对 $FSR1*256+FSR0$ 指向的数据存储器地址进行访问，从而实现间接寻址模式。

字操作高8位缓存器

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

BIT[7:0] **HIBYTE[7:0]** – 字操作高字节缓冲器

HIBYTE: 通过 MOV_C 指令对程序存储器进行读取操作，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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| FSR0 | FSR07 | FSR06 | FSR05 | FSR04 | FSR03 | FSR02 | FSR01 | FSR00 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | 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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| FSR1 | FSR17 | FSR16 | FSR15 | FSR14 | FSR13 | FSR12 | FSR11 | FSR10 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | X | X | X | X | X | X | X | X |

BIT[7:0] **FSR1[7:0]** – 数据指针寄存器 1

FSR1: 间接寻址模式指针高 8 位。

程序指针计数器高字节

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|
| PCH | - | - | - | - | - | PC10 | PC9 | PC8 |
| R/W | - | - | - | - | - | R/W | R/W | R/W |
| 初始值 | - | - | - | - | - | 0 | 0 | 0 |

BIT[2:0] **PC[10:8]** – 程序指针计数器高 3 位

程序指针计数器低字节

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

BIT[7:0] **PC[7:0]** – 程序指针计数器低 8 位

对PCL操作指令： $PC = (PC[10:0]+A[7:0])$ （对PCL操作的加法指令）；

$PC = \{PC[10:8],ALU[7:0](ALU运算结果)\}$ （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 |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| PFLAG | - | - | LVD36 | LVD24 | - | C | DC | Z |
| R/W | - | - | R | R | - | R/W | R/W | R/W |
| 初始值 | - | - | 0 | 0 | - | 0 | 0 | 0 |

BIT[5] **LVD36** – LVD3.6V 标志位（用户配置中 LVD 选为 3.6V 时有效）

0: 系统工作电压 VDD 高于 3.6V；

1: 系统工作电压 VDD 低于 3.6V；

BIT[4] **LVD24** – LVD2.4V 标志位（用户配置中 LVD 选为 2.4V 时有效）

0: 系统工作电压 VDD 高于 2.4V；

1: 系统工作电压 VDD 低于 2.4V；

BIT[2] **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；

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] **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；

杂用寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|
| MCR | GIE | - | - | - | - | STKP2 | STKP1 | STKP0 |
| R/W | R/W | - | - | - | - | R/W | R/W | R/W |
| 初始值 | 0 | - | - | - | - | 1 | 1 | 1 |

BIT[7] **GIE** – 总中断使能位
 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[2:0] **STKP[2:0]** – 堆栈指针

堆栈缓存寄存器高字节

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------------|-------|-------|-------|-------|-------|----------|---------|---------|
| STKRnH | - | - | - | - | - | STKRnB10 | STKRnB9 | STKRnB8 |
| R/W | - | - | - | - | - | R/W | R/W | R/W |
| 初始值 | - | - | - | - | - | 0 | 0 | 0 |

BIT[2:0] **STKRnB[10:8]** – 堆栈缓存寄存器 STKRn 的高 3 位 (n=7-0)

堆栈缓存寄存器低字节

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------------|---------|---------|---------|---------|---------|---------|---------|---------|
| STKRnL | STKRnB7 | STKRnB6 | STKRnB5 | STKRnB4 | STKRnB3 | STKRnB2 | STKRnB1 | STKRnB0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] **STKRnB[7:0]** – 堆栈缓存寄存器 STKRn 的低 8 位 (n=7-0)

注：虽然堆栈指针的复位初始值已设置为 111B，但仍推荐在程序初始部分重新设定 STKP=7 以及禁止程序再访问堆栈相关寄存器，以确保芯片能正常使用堆栈。

3 系统时钟

本芯片为双时钟系统，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。高速时钟源可通过用户配置字选择内置 16MHz 的高频 RC 振荡器 HIRC、外部晶体振荡器 OSC、外部 RC 振荡器 ERC、外部时钟输入 ECK；低速时钟源为内置 32KHz 的低频 RC 振荡器。

系统选用高速时钟时，CPU 的运行速度由用户配置字的 FCPUS 配置；系统选用低速时钟时，CPU 的指令周期为 4 个低频时钟周期。

低频 RC 振荡器也用作 WDT（看门狗）电路时钟源。

3.1 外接晶体振荡器

本芯片可外接 32768Hz/455KHz/4MHz~16MHz 晶体振荡器，该振荡器可用于系统高频时钟。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稳定性。下表列出几种频率的晶振选用电容 Cx 的推荐值。

| 晶体频率 (Hz) | 电容 Cx (pF) | 最低起振电压 (V) |
|-----------|------------|------------|
| 16M | 10 | 2.7 |
| 8M | 10/20 | 2.4 |
| 4M | 10/20 | 2.2 |
| 455K | 220/470 | 2.2 |
| 32768 | 10/20 | 2.2 |

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 外部 RC 振荡器

本芯片可外接 RC 振荡器，外部 RC 振荡电路只需要和 OSCI 引脚连接，电容容值不能低于 100pF，电阻值和电容值共同决定 RC 的振荡频率，最高支持 8MHz。

注：本文中所有提到的外部 RC 频率或 FERC，均指外部 RC 振荡器的输出频率，其值为外部 RC 输入端口 OSCI 波形频率的 2 分频。

3.3 外部时钟输入

本芯片可从 OSCI 引脚直接输入外部时钟来提供系统时钟，外部时钟最高支持 4MHz。

3.4 内置高频 RC 振荡器

本芯片内置一个高精度 16MHz 的 HIRC 振荡器，该振荡器可用于系统高速时钟。

3.5 内置低频 RC 振荡器

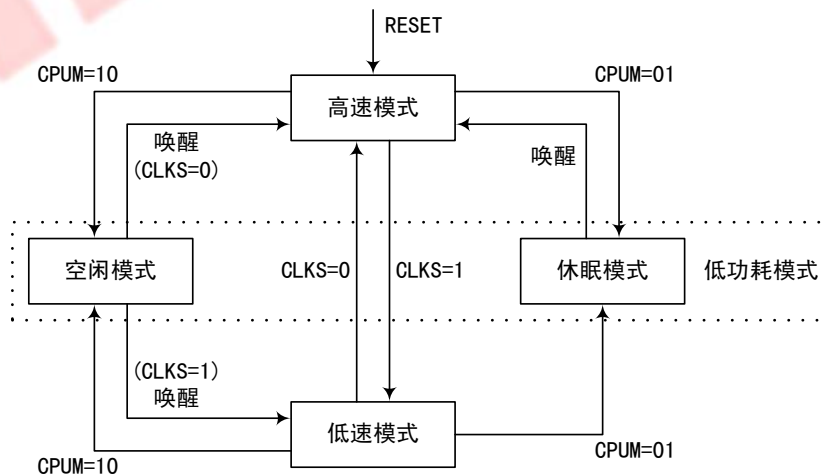
本芯片内置一个典型值为 32KHz 的 LIRC 振荡器，该振荡器可用于系统低频时钟，同时用于上电延时定时器、WDT 定时器等电路。

3.6 系统工作模式

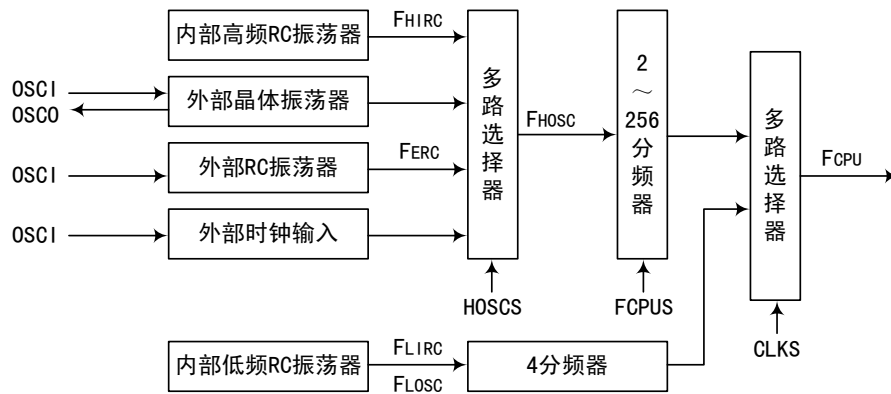
本芯片支持高速模式、低速模式、空闲模式和休眠模式等 4 种工作模式，其中空闲模式和休眠模式为低功耗模式。

| 工作模式 | 进入条件 |
|------|----------------------------|
| 高速模式 | 系统时钟切换到高速时钟 (CLKS=0) |
| 低速模式 | 系统时钟切换到低速时钟 (CLKS=1) |
| 空闲模式 | CPUM=10 |
| 休眠模式 | CPUM=01, 或 CLKS=0 且 HFDE=1 |

工作模式间的切换



系统时钟选择



FHIRC: 内部高频 RC 振荡器频率; FLIRC: 内部低频 RC 振荡器频率;
 FHOSC: 高速时钟频率; FLOSC: 低速时钟频率;
 FCPU: 系统时钟, CPU 时钟源的频率, CPU 停止不表示 Fcpu = 0;

工作模式下的系统状态

| | 高速模式 | 低速模式 | 空闲模式 | 休眠模式 |
|------------------|---------|---------|--------------------|-------------|
| 内/外部高速时钟源 HOSC | 工作 | HFDE 控制 | HFDE 决定 | 停止 |
| 内部低频 RC 振荡器 LIRC | 工作 | 工作 | 工作 | WDTM 决定 |
| CPU | 工作 | 工作 | 停止 | 停止 |
| 定时器 T0 | TOEN 控制 | TOEN 控制 | TOEN 决定 | 停止 |
| 定时器 T1 | T1EN 控制 | T1EN 控制 | 停止 | 停止 |
| 看门狗定时器 WDT | WDTM 配置 | WDTM 配置 | WDTM 配置 | WDTM 配置 |
| 内部中断 | 全部有效 | 全部有效 | T0 | 全部无效 |
| INT0 中断 | 有效 | 有效 | 有效 (标志位置 1) | 无效 |
| 唤醒方法 | - | - | P0 电平变化, T0 溢出, 复位 | P0 电平变化, 复位 |

工作模式寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| OSCM | - | - | - | CPUM1 | CPUM0 | CLKS | HFDE | - |
| R/W | - | - | - | R/W | R/W | R/W | R/W | - |
| 初始值 | - | - | - | 0 | 0 | 0 | 0 | - |

BIT[4:3] **CPUM[1:0]** – 工作模式控制位

| CPUM[1:0] | 工作模式 |
|-----------|----------|
| 00 | 高/低速工作模式 |
| 01 | 休眠模式 |
| 10 | 空闲模式 |
| 11 | 保留 |

BIT[2] **CLKS** – 系统工作时钟选择位

- 0: 高速时钟作为系统时钟;
- 1: 低速时钟作为系统时钟;

- BIT[1] **HFDE** – 高速时钟源禁止位
- 0: 高速时钟源工作;
 - 1: 高速时钟源停止;

3.7 高速模式

高速模式下，CPU 工作时钟由高速时钟源提供，其他模块可在高速或低速下工作。上电复位或其他复位触发后，系统以高速模式工作，当系统从休眠模式唤醒后也进入高速模式。

- ✓ CPU 以高速时钟执行指令代码;
- ✓ 系统在高速时钟下工作；所有模块都可控制，并在各自选定的时钟源下工作；
- ✓ 高速时钟源和内部低速 RC 振荡器都正常工作；
- ✓ 系统可从高速模式切换到其他任何一种模式；
- ✓ 系统从休眠模式唤醒后，进入高速模式；
- ✓ 系统可从低速模式下切换到高速模式；
- ✓ 从高速模式切换到空闲模式后，再唤醒将返回到高速模式；

3.8 低速模式

低速模式下，CPU 工作时钟由内部低速 RC 振荡器提供，其他模块可在高速或低速下工作。高/低速模式的切换由 CLKS 位控制，CLKS=0 时为高速模式，CLKS=1 时为低速模式。从高速模式切换到低速模式时，高速时钟源并不自动停止，可通过 HFDE 位关闭高速时钟源以降低功耗；而从低速模式切换到高速模式时，需先通过 HFDE 开启高速时钟源，然后再将 CLKS 位清 0 切换。低速模式下，CPU 工作速率固定为 FLOSC/4。

- ✓ CPU 以低速时钟执行指令代码；
- ✓ 系统在低速时钟下工作；所有模块都可控制，并在各自选择的时钟源下工作（若选择高速时钟源，而高速时钟源又被关闭，则模块暂停工作）；
- ✓ 内部低速 RC 振荡器正常工作，高速时钟源由 HFDE 位控制；
- ✓ 从低速模式切换到休眠模式，唤醒后进入高速模式；
- ✓ 系统可从高速模式下切换到低速模式；
- ✓ 从低速模式切换到空闲模式后，再唤醒将返回到低速模式；

3.9 休眠模式

休眠模式属于低功耗模式，CPU 和系统均停止工作，高速时钟源停止工作，内部低频 RC 振荡器 LIRC 根据 WDT 模式决定（若 WDT 始终工作，LIRC 工作；否则 LIRC 停止）。休眠模式可由指定端口的电平变化来触发唤醒，从任何工作模式进入到休眠模式，再唤醒后系统都会以高速模式恢复运行。通过 CPUM 位控制是否进入休眠模式，当 CPUM=01，系统进入休眠模式，唤醒后 CPUM 自动恢复成 00，CLKS 也自动清为 0，系统进入高速模式。

- ✓ CPU 停止工作，程序代码暂停执行；

- ✓ 包括内部高/低速 RC 振荡器、外部高速振荡器在内的所有时钟源都停止工作。
- ✓ 系统从休眠模式唤醒后进入高速模式；
- ✓ 指定端口的电平变化将唤醒休眠模式；

注：当 $CLKS=0$ 且 $HFDE=1$ 时，系统也将进入休眠模式。所以在高速模式下将 $HFDE$ 置 1，会进入休眠模式；在低速模式且 $HFDE=1$ 关闭高频时钟源时，若要切换到高速模式，需先将 $HFDE$ 清 0 以开启高频时钟源（一般延时 3~5 个指令周期以待时钟源稳定），再将 $CLKS$ 清 0 以进入高速模式，否则若直接将 $CLKS$ 清 0 会误进入休眠模式。

3.10 空闲模式

空闲模式也属于低功耗模式，CPU 停止工作，但选定的系统时钟 F_{CPU} 仍保持工作，内部模块若选定的时钟源与此时的系统时钟源相同时，也会继续工作。空闲模式下，端口电平变换、或具有唤醒功能的定时器溢出时，将唤醒系统。通过 $CPUM$ 位控制是否进入空闲模式，当 $CPUM=10$ 时，系统进入空闲模式，唤醒后 $CPUM$ 自动恢复成 00，系统按之前的模式恢复工作。

- ✓ CPU 停止工作，程序代码暂停执行；
- ✓ 作为系统时钟源的振荡器正常工作，其他振荡器的工作状态取决于系统工作模式的配置；
- ✓ 具有唤醒功能的定时器正常工作；
- ✓ 由高速模式切换到空闲模式，唤醒后系统返回到高速模式；
- ✓ 由低速模式切换到空闲模式，唤醒后系统返回到低速模式；
- ✓ 指定端口的电平变化、具有唤醒功能的定时器溢出，将唤醒系统；

4 复位

4.1 复位条件

本芯片有四种复位方式：

- ✓ 上电复位 POR；
- ✓ 外部复位（需外部复位引脚处于使能状态）；
- ✓ 低电压复位 LVR；
- ✓ WDT 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特殊功能寄存器 SFR 重置为默认初始值。

上电复位、LVR 复位、外部复位和 WDT 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，由于振荡器起振和稳定需要一定的时间，所以系统会在上电后延时等待振荡稳定后开始工作。

4.2 上电复位

本芯片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。

上电复位过程可以概括为以下几个步骤：

- (1) 检测系统工作电压，等待电压高于 VPOR 并保持稳定；
- (2) 如果外部复位功能开启，则需等待复位引脚电压高于 VIH；
- (3) 初始化包括 SFR 在内的所有寄存器；
- (4) 开启低频时钟振荡器，等待 2048 个周期，即 Tcfg 时间；
- (5) 开启主时钟振荡器，等待振荡稳定，时间为 Tost；
- (6) 上电结束，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用户配置字的 RSTEN 位配置，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

本芯片的 LVR 电压有 4 级（详见用户配置字），通过 VLVR5 位进行配置。电压检测电路有一定的回滞特性，通常回滞电压典型值为 0.1V，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+0.1V 左右时 LVR 复位才会解除。

4.5 看门狗复位

看门狗（WDT）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。

注：在低功耗模式下，CPU 停止工作，若此时有 WDT 溢出，也将产生复位。

5 I/O 端口

5.1 IO 工作模式

芯片共有一组 8 位端口 P0 和两组 5 位端口 P4、P5，除 P03 仅为输入端口外，其余端口均为双向端口。除用作通用输入/输出端口外，部分端口还可复用为 ADC 模拟输入端口、外部中断输入端口、PWM/BUZ 输出端口等工作模式。端口 P0 还具有电平变化唤醒功能。

端口数据寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| IOP0 | P07D | P06D | P05D | P04D | P03D | P02D | P01D | P00D |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| IOP4 | - | - | - | P44D | P43D | P42D | P41D | P40D |
| R/W | - | - | - | R/W | R/W | R/W | R/W | R/W |
| 初始值 | - | - | - | 0 | 0 | 0 | 0 | 0 |

BIT[4:0] **P4nD** – P4 口数据位 (n=4-0)

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| IOP5 | - | - | - | P54D | P53D | P52D | P51D | P50D |
| R/W | - | - | - | R/W | R/W | R/W | R/W | R/W |
| 初始值 | - | - | - | 0 | 0 | 0 | 0 | 0 |

BIT[4:0] **P5nD** – P5 口数据位 (n=4-0)

端口方向寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| OEP0 | P07OE | P06OE | P05OE | P04OE | - | P02OE | P01OE | P00OE |
| R/W | R/W | R/W | R/W | R/W | - | R/W | R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 |

BIT[7:4,2:0] **P0nOE** – P0 口输出使能寄存器 (n=7-4,2-0)

0: 作为输入口，读 P0n 口将读取端口状态；

1: 作为输出口，读 P0n 口将读取端口数据寄存器的值；

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| OEP4 | - | - | - | P44OE | P43OE | P42OE | P41OE | P40OE |
| R/W | - | - | - | R/W | R/W | R/W | R/W | R/W |
| 初始值 | - | - | - | 0 | 0 | 0 | 0 | 0 |

BIT[4:0] **P4nOE** – P4 口输出使能寄存器 (n=4-0)
 0: 作为输入口, 读 P4n 口将读取端口状态;
 1: 作为输出口, 读 P4n 口将读取端口数据寄存器的值;

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| OEP5 | - | - | - | P54OE | P53OE | P52OE | P51OE | P50OE |
| R/W | - | - | - | R/W | R/W | R/W | R/W | R/W |
| 初始值 | - | - | - | 0 | 0 | 0 | 0 | 0 |

BIT[4:0] **P5nOE** – P5 口输出使能寄存器 (n=4-0)
 0: 作为输入口, 读 P5n 口将读取端口状态;
 1: 作为输出口, 读 P5n 口将读取端口数据寄存器的值;

5.2 上拉电阻控制

P0、P4 和 P5 口除 P03 外其余都有独立的上拉电阻控制寄存器位, 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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| PUP0 | P07PU | P06PU | P05PU | P04PU | - | P02PU | P01PU | P00PU |
| R/W | W | W | W | W | - | W | W | W |
| 初始值 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 |

BIT[7:4,2:0] **P0nPU** – P0 口上拉电阻选择位 (n=7-4,2-0)
 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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| PUP4 | - | - | - | P44PU | P43PU | P42PU | P41PU | P40PU |
| R/W | - | - | - | W | W | W | W | W |
| 初始值 | - | - | - | 0 | 0 | 0 | 0 | 0 |

BIT[4:0] **P4nPU** – P4 口上拉电阻选择位 (n=4-0)
 0: P4n 上拉电阻无效;
 1: P4n 上拉电阻有效;

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| PUP5 | - | - | - | P54PU | P53PU | P52PU | P51PU | P50PU |
| R/W | - | - | - | W | W | W | W | W |
| 初始值 | - | - | - | 0 | 0 | 0 | 0 | 0 |

BIT[4:0] **P5nPU** – P5 口上拉电阻选择位 (n=4-0)
 0: P5n 上拉电阻无效;
 1: P5n 上拉电阻有效;

5.3 端口模式控制

P0、P4 和 P5 部分端口可以作为通用 IO 或数字端口，也可以复用为模拟信号输入端口。可通过 ADIOS 寄存器将端口设置成固定的模拟输入端口，以免数/模混合信号或多路模拟信号产生漏电流。

模拟端口设置寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| ADIOS2 | - | - | - | AN4EN | AN3EN | AN2EN | AN1EN | AN0EN |
| R/W | - | - | - | R/W | R/W | R/W | R/W | R/W |
| 初始值 | - | - | - | 0 | 0 | 0 | 0 | 0 |

BIT[4:0] **ANnEN** – 模拟输入通道 ANn 控制位 (n=4-0)

0: 端口可用作通用 IO、数字端口或模拟输入端口；

1: 端口固定用作模拟输入端口，屏蔽通用 IO 和数字端口功能；

5.4 端口电平唤醒功能

端口 P0 在输入状态时，输入信号的电平变化可将系统从低功耗模式中唤醒。

注：应用端口电平唤醒功能时，端口不能浮空且需外部输入电平稳定，否则易产生误触发。

6 定时器

6.1 看门狗定时器 WDT

看门狗定时器的时钟源为内置低频 RC 振荡器，可由用户配置字的 WDTT 位设置溢出时间、WDTM 位设置工作模式。

若选择始终开启看门狗功能，则不管是高速/低速模式还是低功耗模式，WDT 会一直运行，当 WDT 溢出时将产生复位；若选择低功耗模式下关闭看门狗功能，在空闲/休眠模式下 WDT 被硬件自动关闭，唤醒后自动继续运行。

WDT 溢出时间 = 8192 / 内部低频 RC 振荡器频率 (FLIRC)。

WDT 控制寄存器 WDTCR 写 5AH 将清零 WDT 计数器。

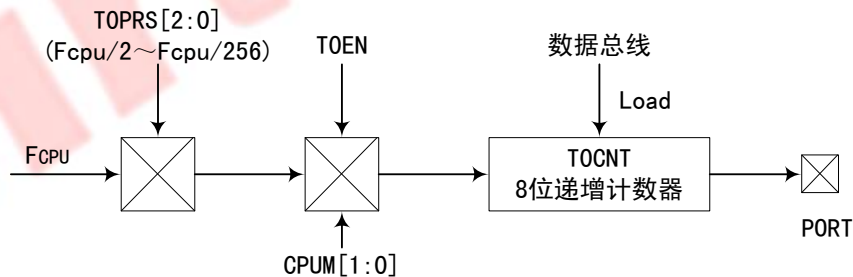
WDT控制寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| WDTCR | WDTCR7 | WDTCR6 | WDTCR5 | WDTCR4 | WDTCR3 | WDTCR2 | WDTCR1 | WDTCR0 |
| R/W | W | W | W | W | W | W | W | W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] WDTCR[7:0] – WDT 控制寄存器，写 5AH 将清零 WDT 计数器

6.2 定时器 T0

定时器 T0 为 8 位定时器，时钟源为系统时钟，当 8 位递增计数器从 0FFH 溢出至 00H 后，产生溢出信号触发 T0 中断。若 T0 在空闲模式下继续工作，则溢出时将唤醒系统。



定时器T0控制寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------|-------|--------|--------|--------|--------|--------|-------|-------|
| TOCR | TOEN | TOPRS2 | TOPRS1 | TOPRS0 | T1CKS2 | T1CKS1 | - | - |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | - | - |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | - | - |

BIT[7] TOEN – T0 使能控制位

0: 关闭 T0;

1: 开启 T0;

BIT[6:4] TOPRS[2:0] – T0 预分频比选择位

| TOPRS[2:0] | T0 时钟频率 |
|------------|----------|
| 000 | FCPU/256 |
| 001 | FCPU/128 |
| 010 | FCPU/64 |
| 011 | FCPU/32 |
| 100 | FCPU/16 |
| 101 | FCPU/8 |
| 110 | FCPU/4 |
| 111 | FCPU/2 |

BIT[3:2] T1CKS[2:1] – T1 时钟源选择位，参见 T1CR 说明

定时器T0计数器

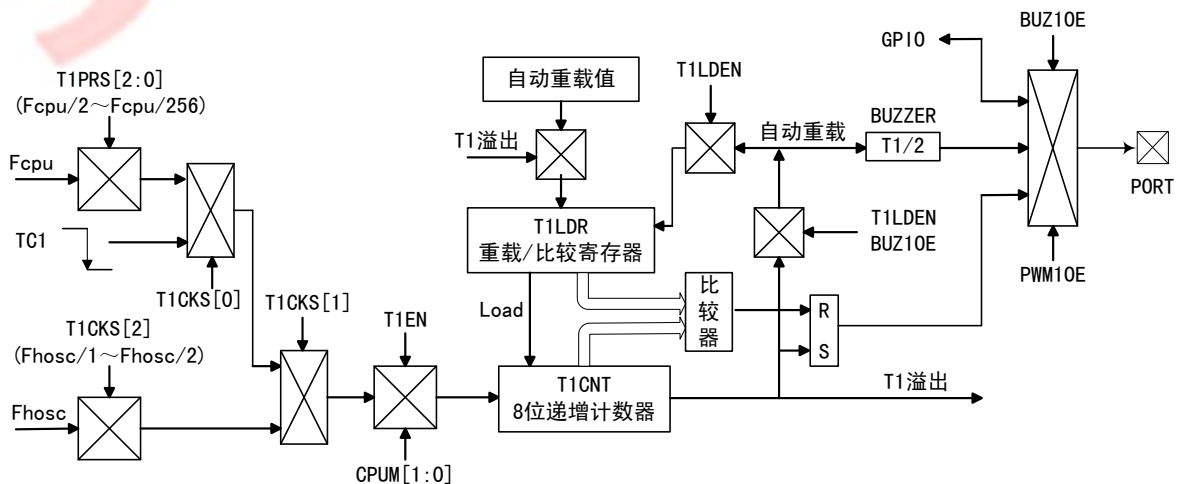
| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| TOCNT | TOCNT7 | TOCNT6 | TOCNT5 | TOCNT4 | TOCNT3 | TOCNT2 | TOCNT1 | TOCNT0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] TOCNT[7:0] – T0 计数器，为可读写的递增计数器。

6.3 定时器 T1

定时器 T1 为 8 位定时/计数器，包含 8 位递增计数器、可编程预分频器、控制寄存器和 8 位重载/比较寄存器。

- ✓ 时钟源可选择系统时钟 Fcpu 分频、高速时钟 Fhosc、或外部 TC1 输入（下降沿触发）；
- ✓ 可通过预分频比设置计数频率；
- ✓ 当用作定时/计数器时，通过重载/比较寄存器控制溢出周期；当允许 PWM 输出时，可通过重载/比较寄存器设置 PWM 占空比；
- ✓ 溢出产生中断；低功耗模式下暂停工作；
- ✓ 可输出 2 分频占空比 50% 的方波，用于蜂鸣器输出；



定时器T1控制寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|--------|--------|--------|--------|--------|--------|--------|
| T1CR | T1EN | T1PRS2 | T1PRS1 | T1PRS0 | T1CKS0 | T1LDEN | BUZ1OE | PWM1OE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7] **T1EN** – T1 使能控制位
0: 关闭 T1;
1: 开启 T1;

BIT[6:4] **T1PRS[2:0]** – T1 预分频比选择位

| T1PRS[2:0] | T1 时钟频率 |
|-------------------|----------------|
| 000 | FCPU/256 |
| 001 | FCPU/128 |
| 010 | FCPU/64 |
| 011 | FCPU/32 |
| 100 | FCPU/16 |
| 101 | FCPU/8 |
| 110 | FCPU/4 |
| 111 | FCPU/2 |

BIT[3] **T1CKS0** – T1 时钟源选择位

| T1CKS1 (T0CR[2]) | T1CKS2 (T0CR[3]) | T1CKS0 | T1 时钟频率 |
|---------------------------|---------------------------|---------------|-------------------|
| 0 : T1 时钟源由 T1CKS0 选择 | X | 0 : FCPU | FCPU/2 ~ FCPU/256 |
| | X | 1 : TC1 | TC1 下降沿 |
| 1 : T1 时钟源由 T1CKS2 控制 | 0 : FHOSC/2 | X | FHOSC/2 |
| | 1 : FHOSC | X | FHOSC |

BIT[2,1] **T1LDEN, BUZ1OE** – 控制作用取决于 PWM1OE
PWM1OE=0

| T1LDEN | 功能定义 |
|---------------|------------------------------------------------------------|
| 0 | 禁止 T1 自动重载；T1CNT 计数范围 0~255，T1 溢出周期=256*时钟周期 |
| 1 | 使能 T1 自动重载；T1CNT 计数范围 T1LDR ~ 255，T1 溢出周期=(256-T1LDR)*时钟周期 |

| BUZ1OE | 功能定义 |
|---------------|------------------------------------|
| 0 | 禁止 T1 输出方波，端口用作其他功能 |
| 1 | 使能 T1 从端口输出方波：占空比 50%，频率=T1 溢出频率/2 |

PWM1OE=1

| T1LDEN | BUZ1OE | T1CNT 计数范围 | PWM1 周期 |
|---------------|---------------|-------------------|----------------|
| 0 | 0 | 0~255 | 256*时钟周期 |
| 0 | 1 | 0~63 | 64*时钟周期 |
| 1 | 0 | 0~31 | 32*时钟周期 |
| 1 | 1 | 0~15 | 16*时钟周期 |

BIT[0] **PWM1OE** – PWM1 输出使能位
 0: 禁止 PWM1 输出;
 1: 允许 PWM1 输出;

定时器T1计数器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
| T1CNT | T1CNT7 | T1CNT6 | T1CNT5 | T1CNT4 | T1CNT3 | T1CNT2 | T1CNT1 | T1CNT0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] **T1CNT[7:0]** – T1 计数器，为可读写的递增计数器。

定时器T1重载/比较寄存器

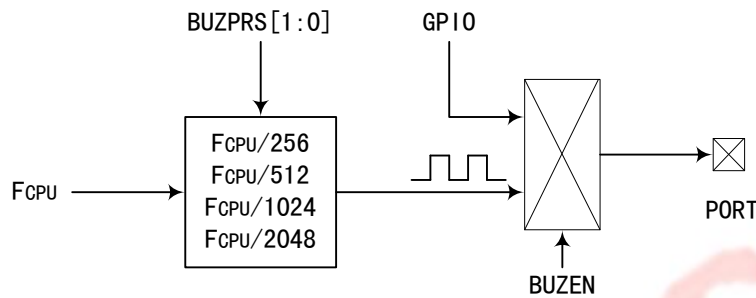
| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
| T1LDR | T1LDR7 | T1LDR6 | T1LDR5 | T1LDR4 | T1LDR3 | T1LDR2 | T1LDR1 | T1LDR0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] **T1LDR[7:0]** – T1 重载/比较寄存器，用于设置 T1 计数周期或设置 PWM1 占空比。

7 蜂鸣器输出

7.1 BUZZER 概述

芯片内建一个具有独立计数器且频率可设置的蜂鸣器输出电路 BUZZER，与 IO 端口复用引脚。BUZZER 的时钟源为系统时钟 F_{CPU} 分频，输出波形占空比固定为 50%。BUZEN=1 时，端口输出蜂鸣器驱动波形；BUZEN=0 时，端口恢复成上一个 I/O 状态。



蜂鸣器输出频率可通过分频选择位设定，如下表所示：

| BUZPRS[1:0] | 时钟分频比 | 蜂鸣器输出频率 | | |
|-------------|------------------------|------------------------|------------------------|------------------------|
| | | F _{CPU} =1MHz | F _{CPU} =2MHz | F _{CPU} =4MHz |
| 00 | F _{CPU} /256 | 4KHz | 8KHz | 16KHz |
| 01 | F _{CPU} /512 | 2KHz | 4KHz | 8KHz |
| 10 | F _{CPU} /1024 | 1KHz | 2KHz | 4KHz |
| 11 | F _{CPU} /2048 | 0.5KHz | 1KHz | 2KHz |

7.2 BUZZER 相关寄存器

BUZ控制寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|---------|---------|-------|-------|-------|-------|-------|
| BUZCR | BUZEN | BUZPRS1 | BUZPRS0 | - | - | - | - | - |
| R/W | R/W | R/W | R/W | - | - | - | - | - |
| 初始值 | 0 | 0 | 0 | - | - | - | - | - |

BIT[7] **BUZEN** – BUZZER 功能使能位
 0: 关闭 BUZZER，端口用作其他功能；
 1: 开启 BUZZER，端口用作蜂鸣器输出 BZ；

BIT[6:5] **BUZPRS[1:0]** – BUZZER 预分频比选择位

| BUZPRS[1:0] | BUZZER 时钟频率 |
|-------------|------------------------|
| 00 | F _{CPU} /256 |
| 01 | F _{CPU} /512 |
| 10 | F _{CPU} /1024 |
| 11 | F _{CPU} /2048 |

8 模数转换器

8.1 ADC 概述

芯片内置的 ADC 模块为 12 位高精度逐次逼近型 ADC。

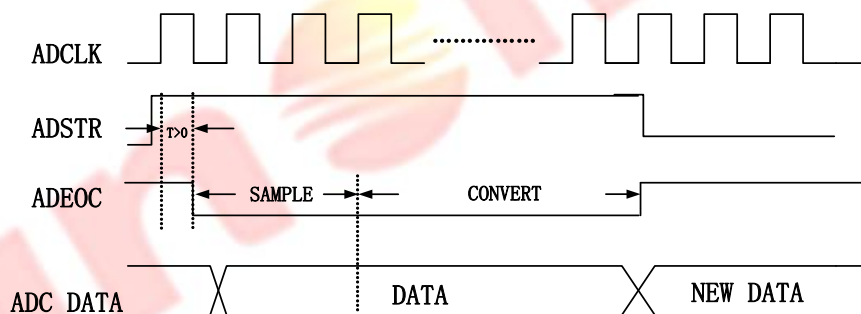
- ✓ 外部 5 通道：AN0~AN4；
- ✓ 参考电压：VDD；
- ✓ 可配置 ADC 时钟：F_{CPU} 经 4/8/32/64 分频；

ADC 模块通过 ADEN 位使能，通过 ADCHS 位选择转换的模拟通道，GCHS 位控制模拟通道开/关，ADCKS 位选择转换速度，ADSTR 位为启动控制位，ADEOC 为结束标志位。对 ADSTR 位写 1 将启动 AD 转换，12 位转换值的转换时间需 20 个 AD 转换时钟周期，转换结束后，ADSTR 位自动清 0，ADEOC 位自动置 1，转换结果放在 ADB/ADR 中，同时中断标志位 ADIF 置 1，提出中断请求。

为保证转换效果及防止漏电流产生，应通过 ADIOS 寄存器将端口设置为纯模拟输入端口，并关闭内部上拉电阻。

8.2 ADC 转换时序图

ADC 转换时序如下图所示



注： SAMPLE (采样) 时间为 8 个 ADCLK，CONVERT (转换) 时间为 12 个 ADCLK，一次 ADC 转换为 20 个 ADCLK。

8.3 ADC 操作步骤

模数转换设置步骤：

- (1) 设置相应端口为输入端口，关闭上拉电阻；
- (2) 通过 ADIOS 寄存器，设置相应端口为纯模拟输入端口；
- (3) 设置 ADCKS 位，选取适当的 ADC 转换时钟；
- (4) ADEN 位置 1，使能 ADC 模块；GCHS 位置 1，打开模拟通道；
- (5) 设置 ADCHS 位，选择 ADC 转换通道；
- (6) ADSTR 位写 1，启动 AD 转换；

- (7) 等待 ADEOC 硬件置 1（或利用 ADC 中断）；
- (8) 读取 ADC 转换结果（ADB、ADR）；
- (9) 重复（7）~（10），对不同的通道进行转换或对同一通道进行多次转换；

注：

- 1、AD 转换过程中或者 ADEN 未使能时，ADB/ADR 中的数据未知，应在 AD 转换结束且 ADEN 使能的情况下读取 AD 转换数据；
- 2、使能 ADC 模块后，需待电路稳定（时间 > 200us）后才能启动 AD 转换；切换输入通道后，前两次转换的结果会有误差，建议舍弃；
- 3、转换时钟越慢，则越能过滤外部输入的波动，越能保证 AD 转换的精度；

8.4 ADC 相关寄存器

ADC 模式寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------------|-------|-------|-------|-------|-------|--------|--------|--------|
| ADM | ADEN | ADSTR | ADEOC | GCHS | - | ADCHS2 | ADCHS1 | ADCHS0 |
| R/W | R/W | R/W | R/W | R/W | - | R/W | R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 |

BIT[7] **ADEN** – ADC 功能使能位

- 0: 关闭 ADC 模块；
- 1: 使能 ADC 模块；

BIT[6] **ADSTR** – ADC 启动位

- 0: AD 转换结束后硬件自动清 0；
- 1: 写 1 启动 AD 转换；

BIT[5] **ADEOC** – ADC 转换结束标志位

- 0: AD 转换过程中；
- 1: AD 转换结束；

BIT[4] **GCHS** – 模拟通道控制位

- 0: 关闭模拟通道；
- 1: 开启模拟通道；

BIT[2:0] **ADCHS[2:0]** – ADC 模拟通道选择

| ADCHS[2:0] | ADC 模拟通道选择 |
|------------|------------|
| 000 | AN0 |
| 001 | AN1 |
| 010 | AN2 |
| 011 | AN3 |
| 100 | AN4 |
| 101 ~ 111 | - |

ADC转换结果寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|
| ADB | ADB11 | ADB10 | ADB9 | ADB8 | ADB7 | ADB6 | ADB3 | ADB4 |
| R/W | R | R | R | R | R | R | R | R |
| 初始值 | X | X | X | X | X | X | X | X |

BIT[7:0] **ADB[11:4]** – ADC 转换结果高 8 位

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------------|-------|--------|-------|--------|-------|-------|-------|-------|
| ADR | - | ADCKS1 | - | ADCKS0 | ADB3 | ADB2 | ADB1 | ADB0 |
| R/W | - | R/W | - | R/W | R | R | R | R |
| 初始值 | - | 0 | - | 0 | X | X | X | X |

BIT[6,4] **ADCKS[1:0]** – AD 转换时钟选择位

| ADCKS[1:0] | AD 转换时钟 F_{ADC} |
|-------------------|-------------------------------------|
| 00 | FCPU/64 |
| 01 | FCPU/32 |
| 10 | FCPU/4 |
| 11 | FCPU/8 |

BIT[3:0] **ADB[3:0]** – ADC 转换结果低 4 位

9 中断

中断有外部中断 (INT0)、定时器中断 (T0, T1)、ADC 转换中断。可通过 CPU 杂用寄存器 MCR 的 GIE 位屏蔽所有中断。

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 ACC 和状态寄存器 PFLAG 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，程序跳到中断向量地址开始执行中断服务程序，中断服务程序应先执行 PUSH 指令以保存累加器 ACC 和状态寄存器 PFLAG，然后判断是哪一个中断响应。
- ✓ 中断服务程序执行完中断内容后应执行 POP 指令以恢复累加器 ACC 和状态寄存器 PFLAG，然后执行 RETIE 指令返回主程序。此时芯片将从堆栈取出 PC 值，然后从中断发生时当前指令的下一条指令继续执行。

本芯片的中断向量地址是 0008H。

9.1 外部中断

本芯片有 1 路外部中断源 INT0，可设置为上升沿触发、下降沿触发和双沿触发三种模式，当外部中断触发时，外部中断标志 (INT0IF) 将被置 1，若中断总使能位 GIE 为 1 且外部中断使能位 (INT0IE) 为 1，则产生外部中断。

外部中断控制寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|--------|--------|-------|-------|-------|
| EINTCR | - | - | - | MINT01 | MINT00 | - | - | - |
| R/W | - | - | - | R/W | R/W | - | - | - |
| 初始值 | - | - | - | 0 | 0 | - | - | - |

BIT[4:3] MINT0[1:0] – 外部中断 INT0 触发条件选择位

| MINT0[1:0] | INT0 触发条件 |
|------------|-----------|
| 00 | 保留 |
| 01 | 上升沿触发 |
| 10 | 下降沿触发 |
| 11 | 上升/下降沿触发 |

9.2 定时器中断

定时器 T0、T1 在计数溢出时中断标志 (T0IF、T1IF) 将被置 1，若中断总使能位 GIE 为 1 且定时器中断使能位 (T0IE、T1IE) 为 1，则产生定时器中断。

9.3 ADC 转换中断

ADC 转换完成后中断标志 (ADIF) 将被置 1, 若中断总使能位 GIE 为 1 且 ADC 中断使能位 (ADIE) 为 1, 则产生 ADC 中断。

9.4 中断相关寄存器

中断使能寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|--------|
| INTE | ADIE | - | T1IE | TOIE | - | - | - | INTOIE |
| R/W | R/W | - | R/W | R/W | - | - | - | R/W |
| 初始值 | 0 | - | 0 | 0 | - | - | - | 0 |

BIT[7] **ADIE** – ADC 中断使能位

- 0: 屏蔽 ADC 中断;
- 1: 使能 ADC 中断;

BIT[5] **T1IE** – 定时器 T1 中断使能位

- 0: 屏蔽定时器 T1 中断;
- 1: 使能定时器 T1 中断;

BIT[4] **TOIE** – 定时器 T0 中断使能位

- 0: 屏蔽定时器 T0 中断;
- 1: 使能定时器 T0 中断;

BIT[0] **INTOIE** – INTO 中断使能位

- 0: 屏蔽 INTO 中断;
- 1: 使能 INTO 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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|--------|
| INTF | ADIF | - | T1IF | TOIF | - | - | - | INTOIF |
| R/W | R/W | - | R/W | R/W | - | - | - | R/W |
| 初始值 | 0 | - | 0 | 0 | - | - | - | 0 |

BIT[7] **ADIF** – ADC 中断标志位

- 0: 未发生 ADC 中断;
- 1: 发生 ADC 中断, 需软件清零;

BIT[5] **T1IF** – 定时器 T1 中断标志位

- 0: 未发生定时器 T1 中断;
- 1: 发生定时器 T1 中断, 需软件清零;

- BIT[4] **TOIF** – 定时器 T0 中断标志位
0: 未发生定时器 T0 中断;
1: 发生定时器 T0 中断, 需软件清零;
- BIT[0] **INTOIF** – INTO 中断标志位
0: 未发生 INTO 中断;
1: 发生 INTO 中断, 需软件清零;

10 电气参数

10.1 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 | -0.3 ~ VDD+0.3 | V |
| 工作温度 | TA | -40 ~ 85 | °C |
| 储存温度 | Tstg | -65 ~ 150 | °C |
| 流入 VDD 最大电流 | IVDDmax | 50 | mA |
| 流出 VSS 最大电流 | IVSSmax | 50 | mA |

10.2 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@HIRC | V _{LVR32} | | 5.5 | V |
| | | | Fcpu=4MHz@HIRC | V _{LVR27} | | 5.5 | |
| | | | Fcpu=2MHz@HIRC | V _{LVR22} | | 5.5 | |
| | | | Fcpu=1MHz@HIRC | V _{LVR22} | | 5.5 | |
| | | | Fcpu=500KHz@HIRC | V _{LVR22} | | 5.5 | |
| | | | Fcpu=FLIRC/4 | V _{LVR22} | | 5.5 | |
| 输入漏电 | Vleak | 所有输入脚 | | -1 | | 1 | uA |
| 输入高电平 1 | Vih1 | 所有输入脚 (除 P03) | | 0.7VDD | | | V |
| 输入高电平 2 | Vih2 | P03 | | 0.8VDD | | | V |
| 输入低电平 1 | Vil1 | 所有输入脚 (除 P03) | | | | 0.3VDD | V |
| 输入低电平 2 | Vil2 | P03 | | | | 0.2VDD | V |
| 输出高电平 驱动电流 | Ioh | 所有输出脚 | VDD=5V, Voh=0.9VDD | | 16 | | mA |
| 输出低电平 驱动电流 | Iol1 | 所有输出脚 (除 P03) | VDD=5V, Vol=0.1VDD, DRVS=0 | | 16 | | mA |
| | | | VDD=5V, Vol=0.1VDD, DRVS=1 | 13 | 25 | 50 | mA |
| | Iol2 | P03 | VDD=5V, Vol=0.1VDD | | | | mA |
| 上拉电阻 | Rpu | 所有输入脚 | VDD=5V, Vin=0 | 40 | 85 | 160 | KΩ |
| 高速模式功耗 | Ihigh | VDD | VDD=5V, Fcpu=1MHz@HIRC | | 1.3 | | mA |
| | | | VDD=3V, Fcpu=1MHz@HIRC | | 0.8 | | mA |
| 低速模式功耗 | Islow | VDD | VDD=5V, Fcpu=FLIRC/4, HIRC 开 | | 500 | | uA |
| | | | VDD=5V, Fcpu=FLIRC/4, HIRC 关 | | 5 | 10 | uA |
| | | | VDD=3V, Fcpu=FLIRC/4, HIRC 关 | | 2 | 4 | uA |

| | | | | | | | |
|--------|--------|-----|---------------------------------------|------|-----|------|----|
| 空闲模式功耗 | Iidle | VDD | VDD=5V, CPU 停止, ADC 关, HIRC 关, LIRC 开 | | 3 | 6 | uA |
| | | | VDD=3V, CPU 停止, ADC 关, HIRC 关, LIRC 开 | | 1 | 3 | uA |
| 休眠模式功耗 | Istop | VDD | VDD=5V, CPU 停止, ADC 关, HIRC/LIRC 关 | | 1 | 3 | uA |
| 低压复位电压 | VLVR22 | VDD | | -15% | 2.2 | +15% | V |
| | VLVR27 | VDD | | -15% | 2.7 | +15% | V |
| | VLVR32 | VDD | | -15% | 3.2 | +15% | V |
| | VLVR36 | VDD | | -15% | 3.6 | +15% | V |
| 低压检测电压 | VLVD | VDD | | -15% | 2.4 | +15% | V |
| | | | | -15% | 3.6 | +15% | V |

10.3 ADC 特性参数

VDD=5V, 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 _{ADC} | | 2.7 | | 5.5 | V |
| 分辨率 | NR | | | 12 | | bit |
| 积分线性误差 | INL | V _{REF} =VDD=5V, F _{ADC} =1MHz, T _{con} =8us | | | ±4 | LSB |
| 微分线性误差 | DNL | V _{REF} =VDD=5V, F _{ADC} =1MHz, T _{con} =8us | | | ±2 | LSB |
| 零点偏移误差 | Ez | V _{REF} =VDD=5V, F _{ADC} =1MHz, T _{con} =8us | | | ±4 | LSB |
| 增益误差 | ET | V _{REF} =VDD=5V, F _{ADC} =1MHz, T _{con} =8us | | | ±4 | LSB |
| 转换时钟 | F _{ADC} | VDD=5V | | | 2 | MHz |
| 转换时间 | T _{con} | | | 20 | | 1/F _{ADC} |
| ADC 输入电压 | V _{AIN} | | VSS | | VDD | |
| ADC 输入阻抗 | R _{AIN} | | 2 | | | MΩ |
| ADC 输入电流 | I _{AIN} | VDD=5V | | | 2 | uA |
| ADC 动态电流 | I _{ADD} | VDD=5V, AD 转换中 | | 1 | 3 | mA |
| ADC 静态电流 | I _{ADS} | VDD=5V, ADEN=0 | | 0.1 | 1 | uA |
| 模拟信号源推荐阻抗 | Z _{AIN} | | | | 10 | KΩ |

10.4 交流电气参数

T=25°C

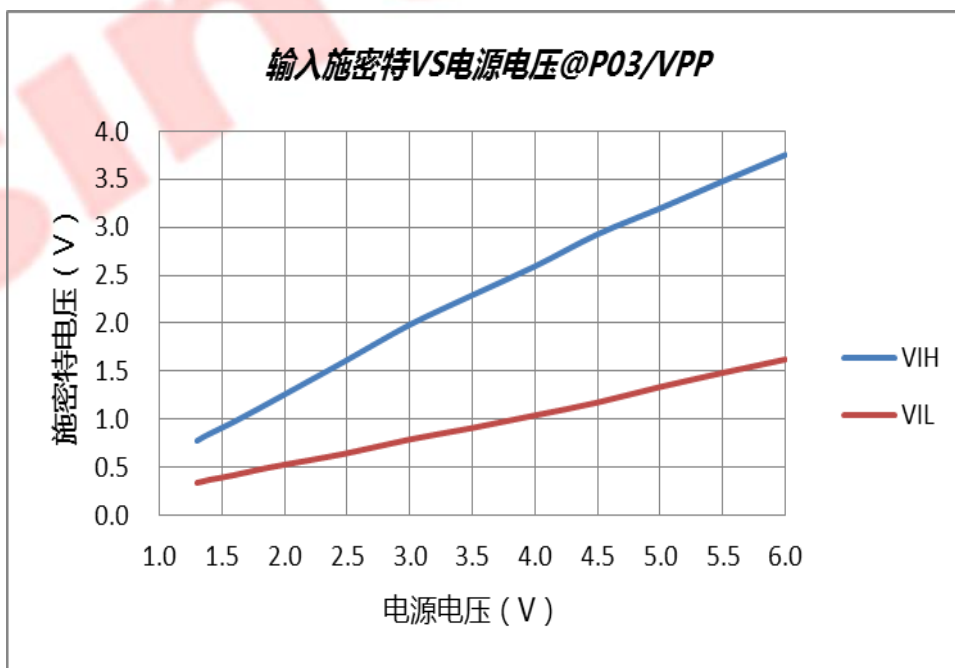
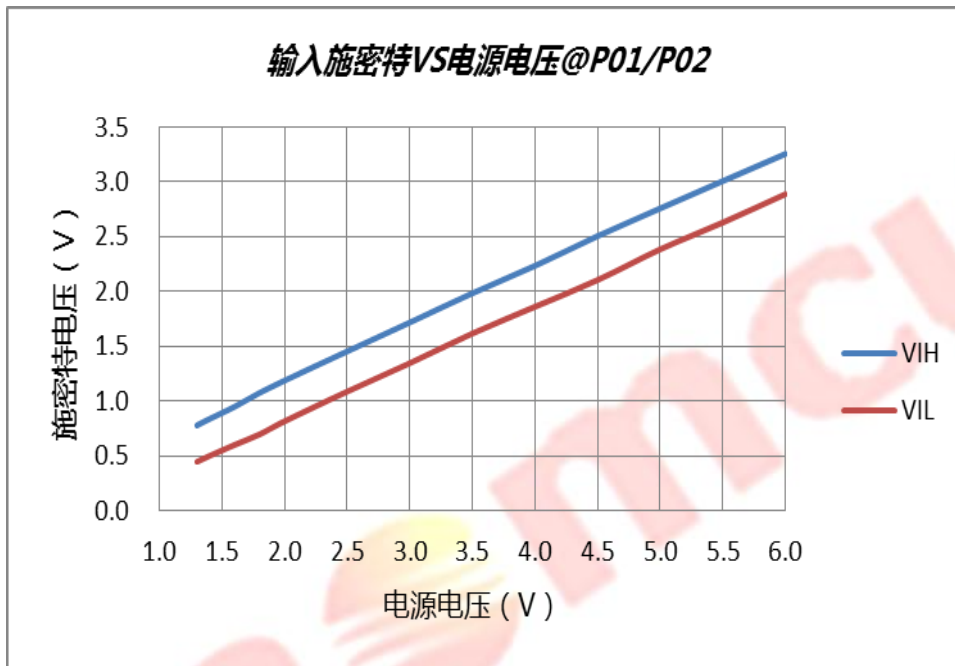
| 特性 | 符号 | 条件 | 最小 | 典型 | 最大 | 单位 |
|--------------|-------|----------------------------|------|----|------|-----|
| 内部高频 RC 振荡频率 | FHIRC | T=25°C, VDD=5V | -2% | 16 | +2% | MHz |
| | | T=-40°C~85°C, VDD=2.0~5.5V | -6% | 16 | +6% | MHz |
| 内部低频 RC 振荡频率 | FLIRC | VDD=5V | -50% | 32 | +50% | KHz |
| | | VDD=3V | -50% | 24 | +50% | KHz |

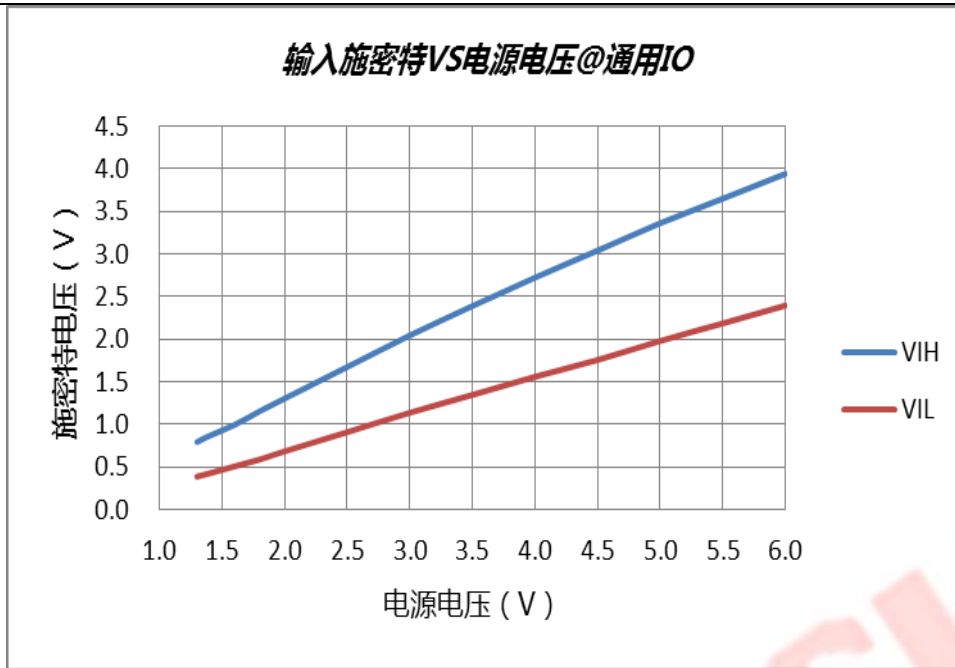
| | | | | | |
|-----------------|---------|--------------------------------|-----|------|-----|
| 外部 16M 晶体起振电压 | VXT16M | | 2.7 | | V |
| 外部 8M 晶体起振电压 | VXT8M | | 2.4 | | V |
| 外部 4M 晶体起振电压 | VXT4M | | 2.2 | | V |
| 外部 455K 晶体起振电压 | VXT455K | | 2.2 | | V |
| 外部 32768 晶体起振电压 | VLEXT | | 2.2 | | V |
| 外部 32768 晶体起振时间 | FLEXT | VDD=5V | | 1 | S |
| 外部 RC 频率 1 | FERC1 | VDD=5V , REXT=1K , CEXT=100pF | | 5.0 | MHz |
| 外部 RC 频率 2 | FERC2 | VDD=5V , REXT=3K , CEXT=100pF | | 2.0 | MHz |
| 外部 RC 频率 3 | FERC3 | VDD=5V , REXT=1K , CEXT=1000pF | | 0.65 | MHz |

11 特性曲线图

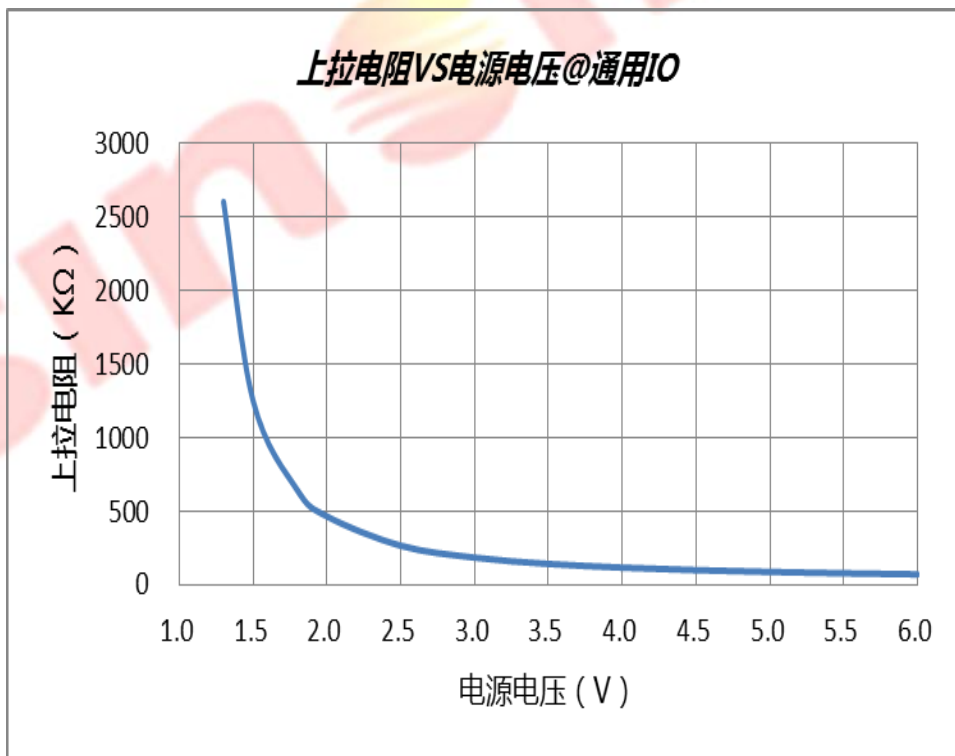
注：本章节列出的特性曲线图仅作为设计参考，部分数据可能超出芯片额定的工作条件范围，为保证芯片能正常工作，请严格按照电气参数说明。

11.1 输入施密特电压 VS 电源电压



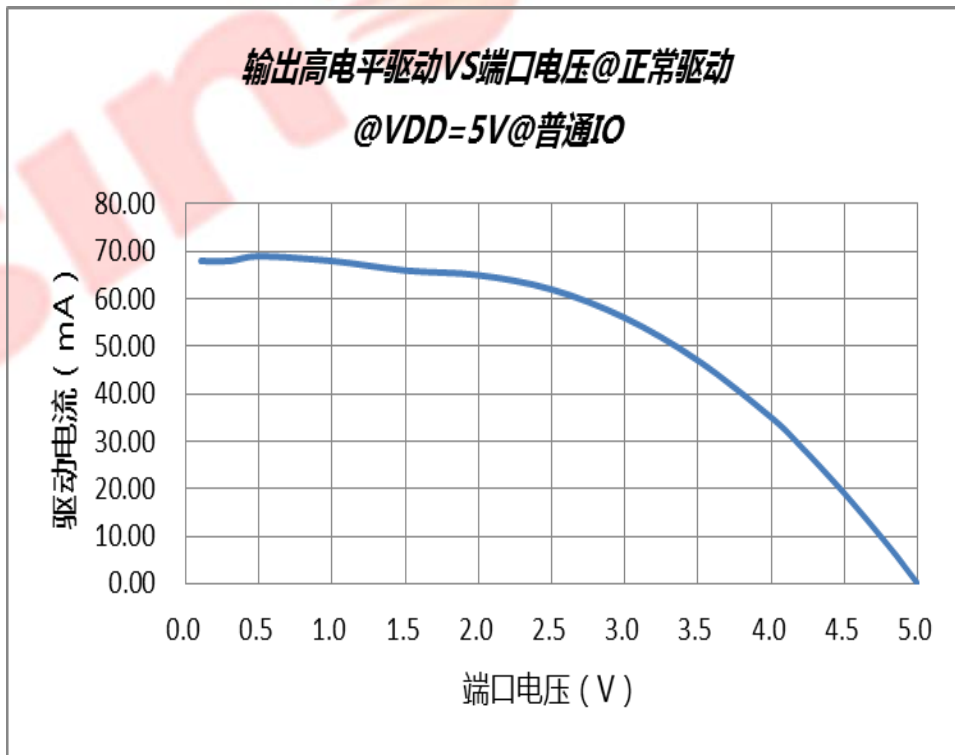
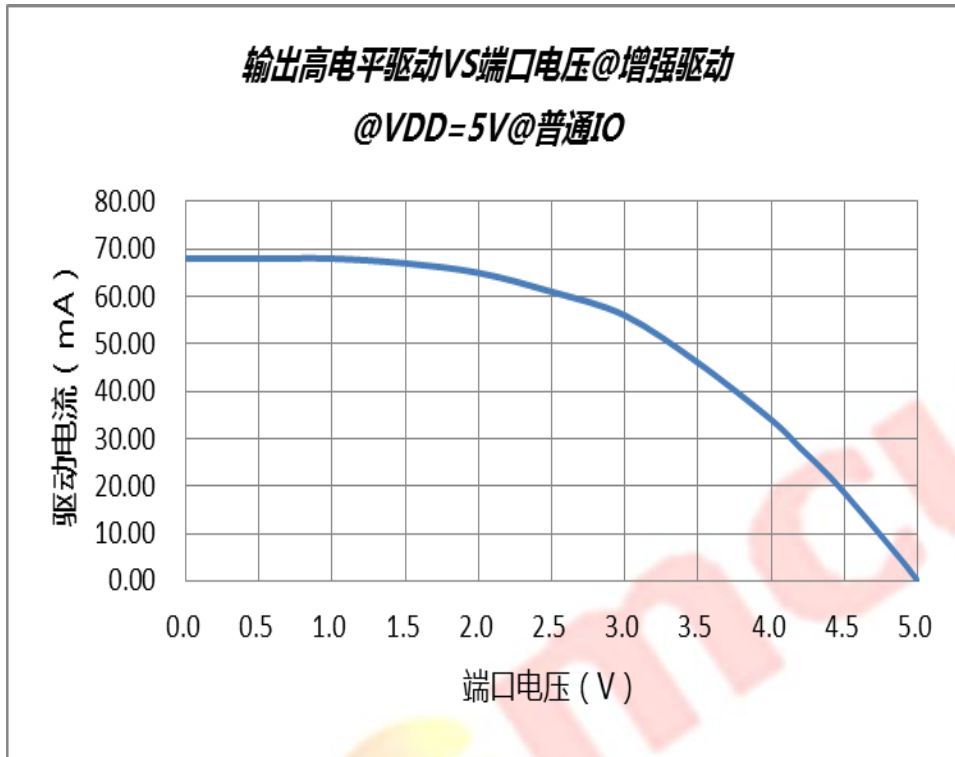


11.2 上拉电阻 VS 电源电压

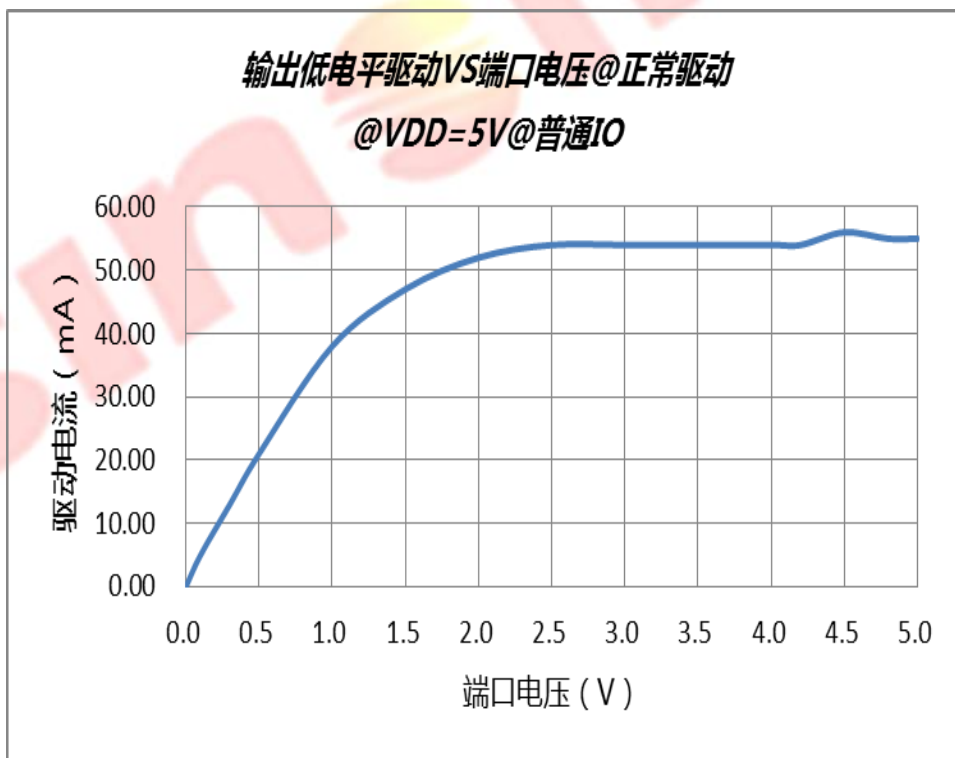
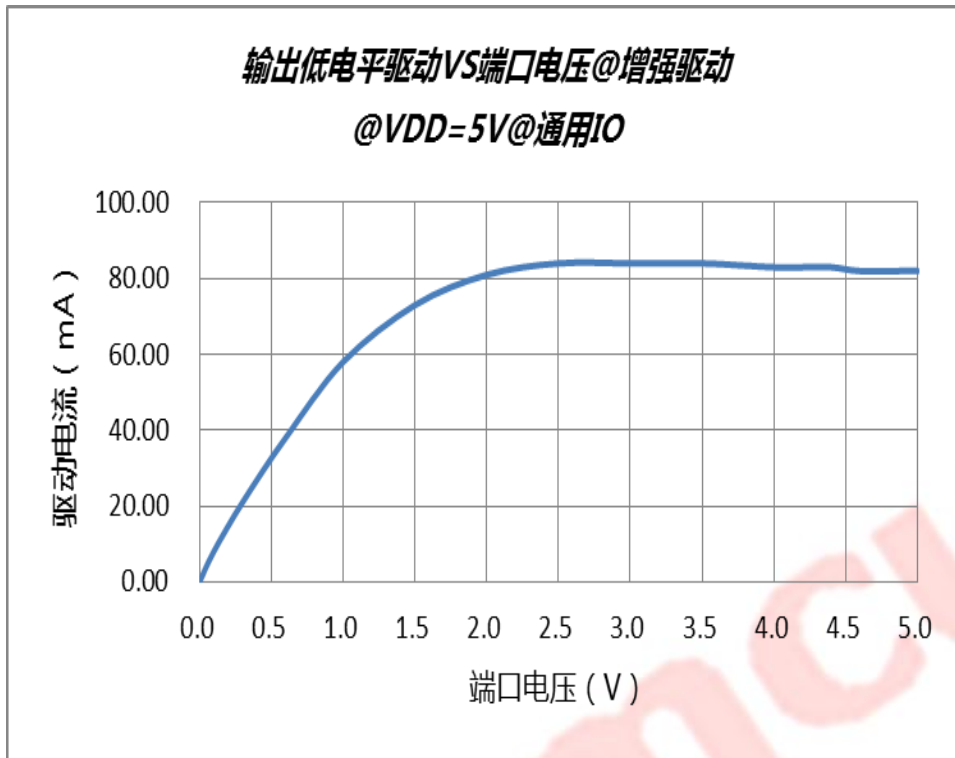


11.3 驱动电流 VS 端口电压

高电平驱动电流 VS 端口电压

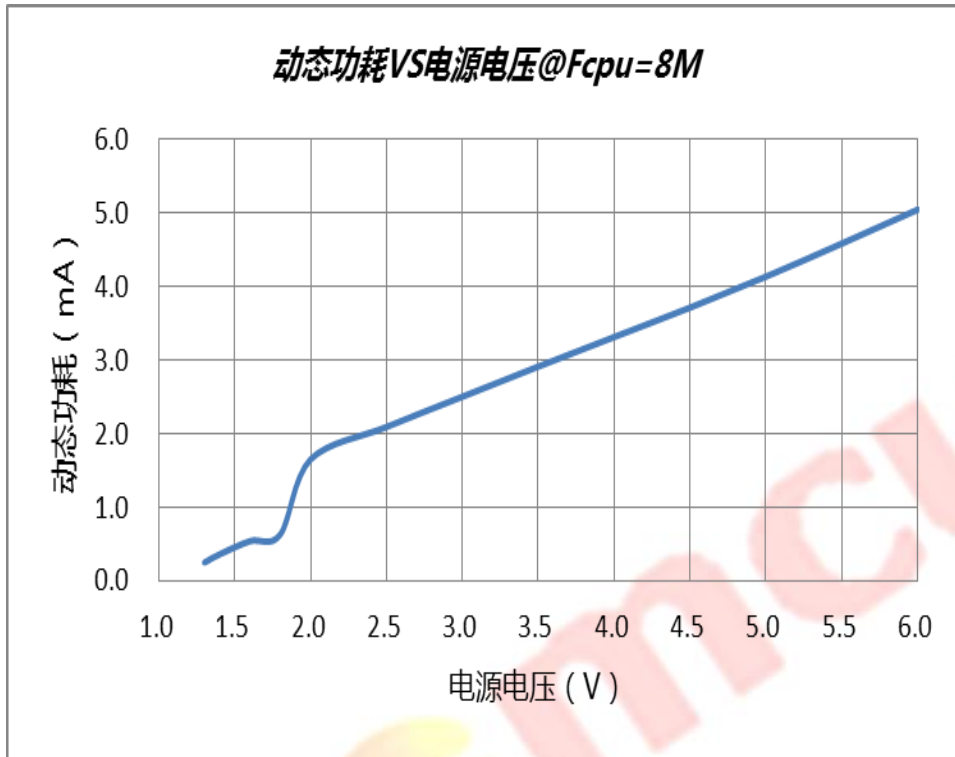


低电平驱动电流 VS 端口电压

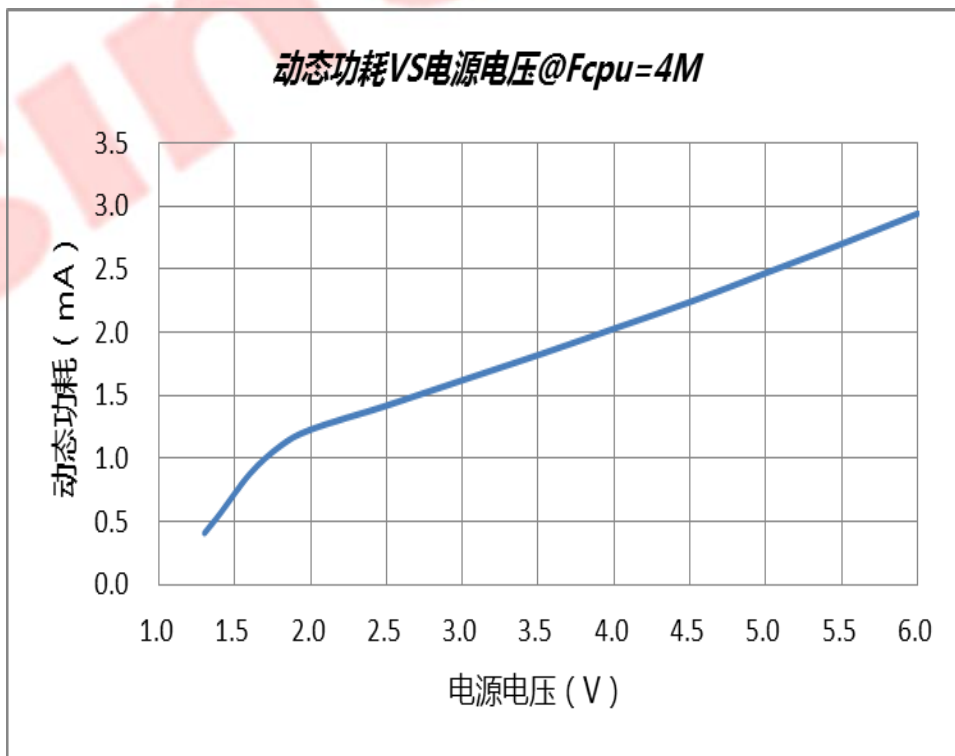


11.4 内部高频 RC 动态功耗 VS 电源电压

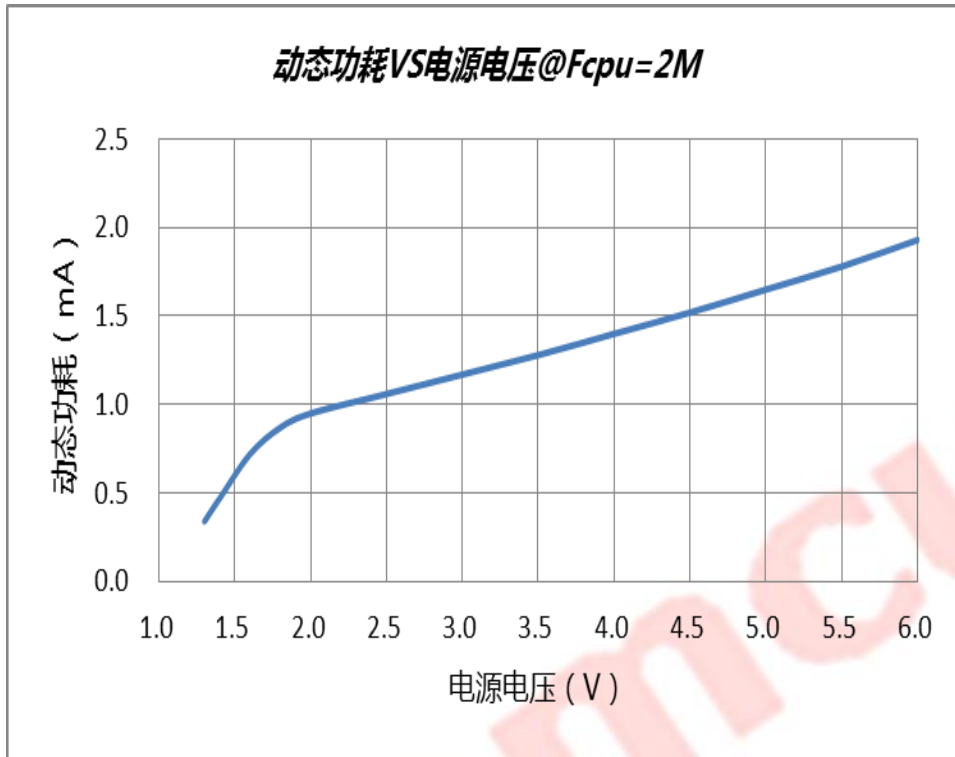
内部FCPU=8MHz动态功耗 VS 电源电压



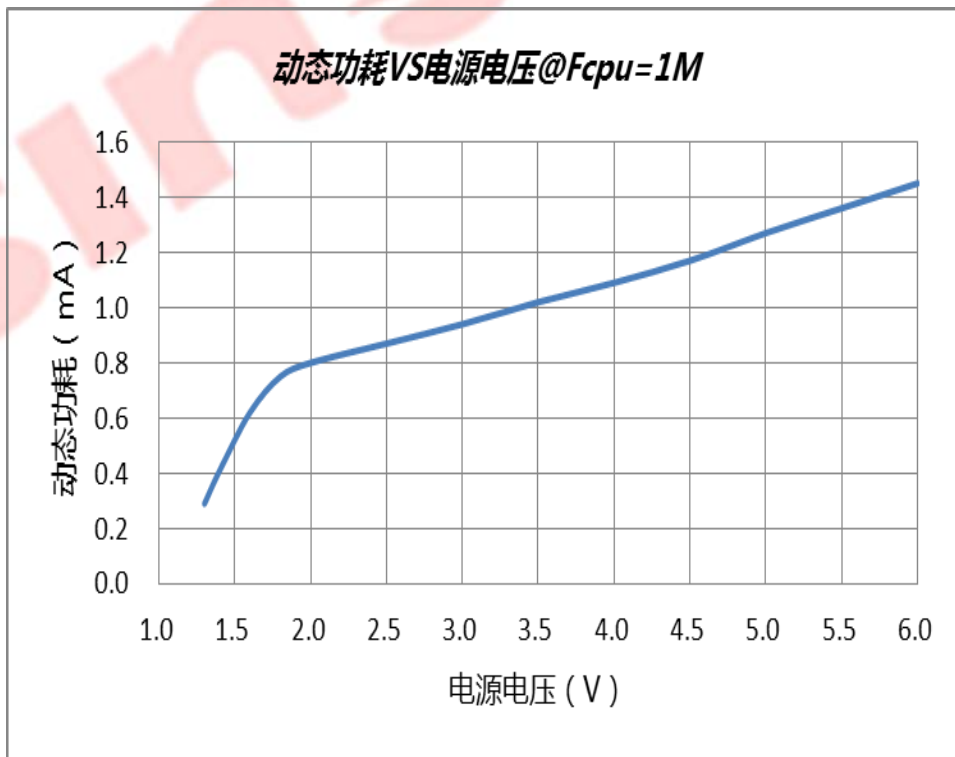
内部FCPU=4MHz动态功耗 VS 电源电压



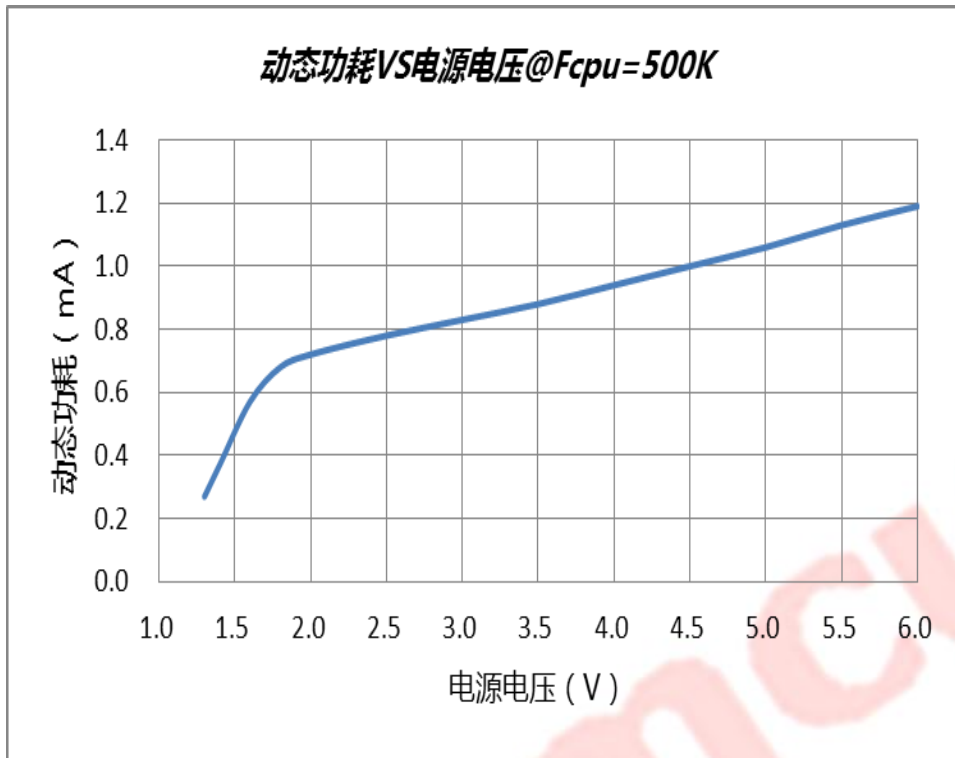
内部FCPU=2MHz动态功耗 VS 电源电压



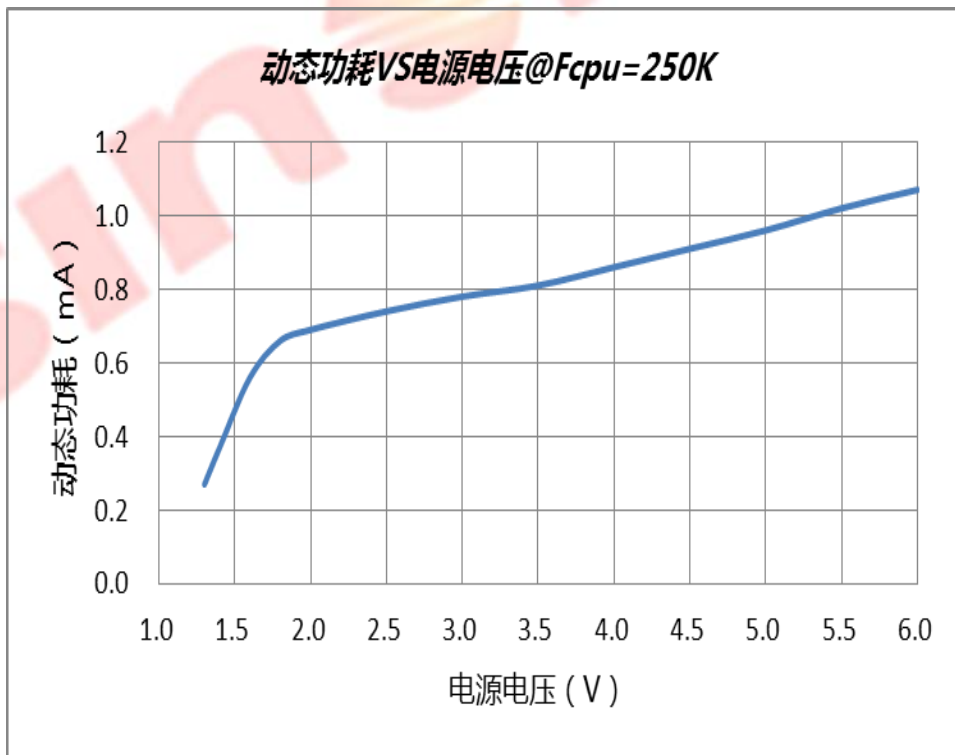
内部FCPU=1MHz动态功耗 VS 电源电压



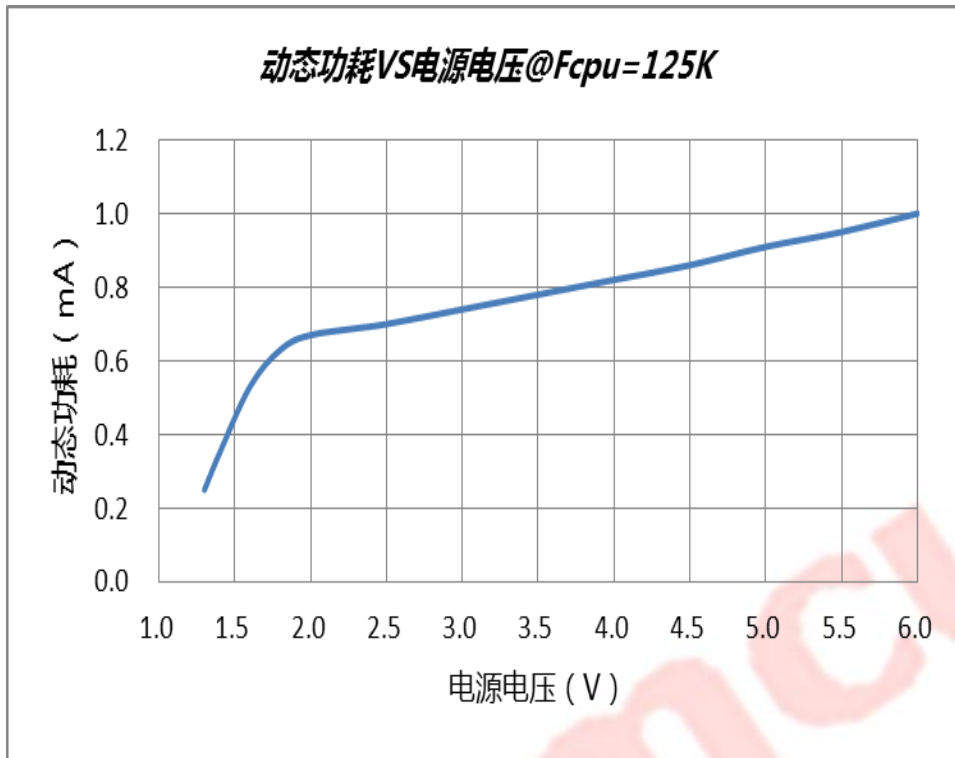
内部FCPU=500KHz动态功耗 VS 电源电压



内部FCPU=250KHz动态功耗 VS 电源电压

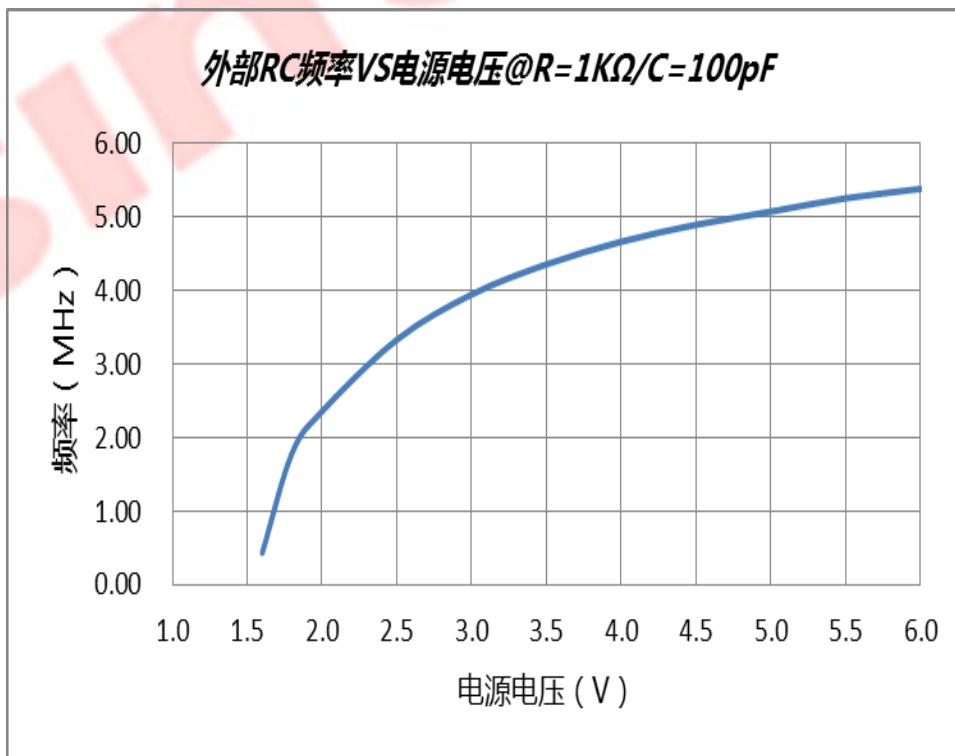


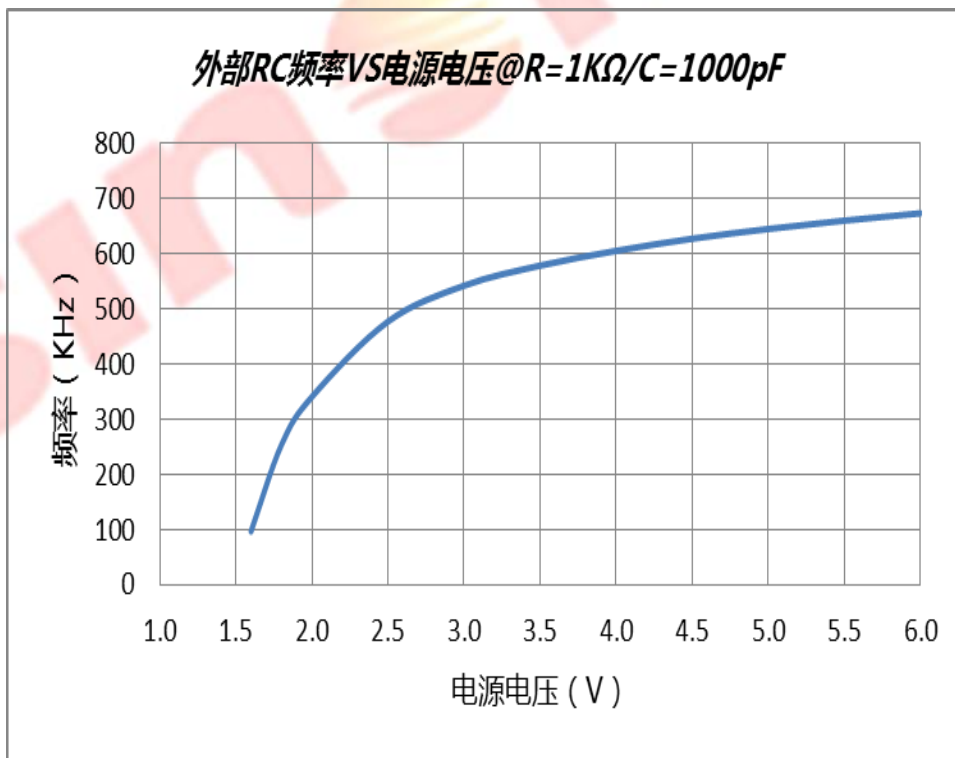
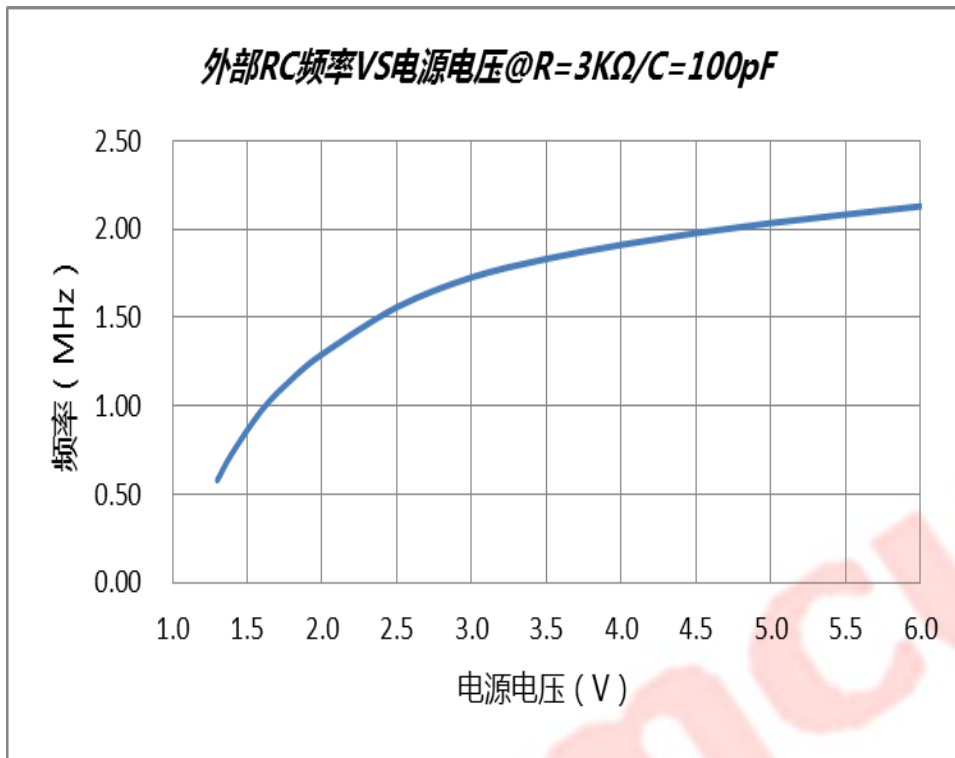
内部FCPU=125KHz动态功耗 VS 电源电压



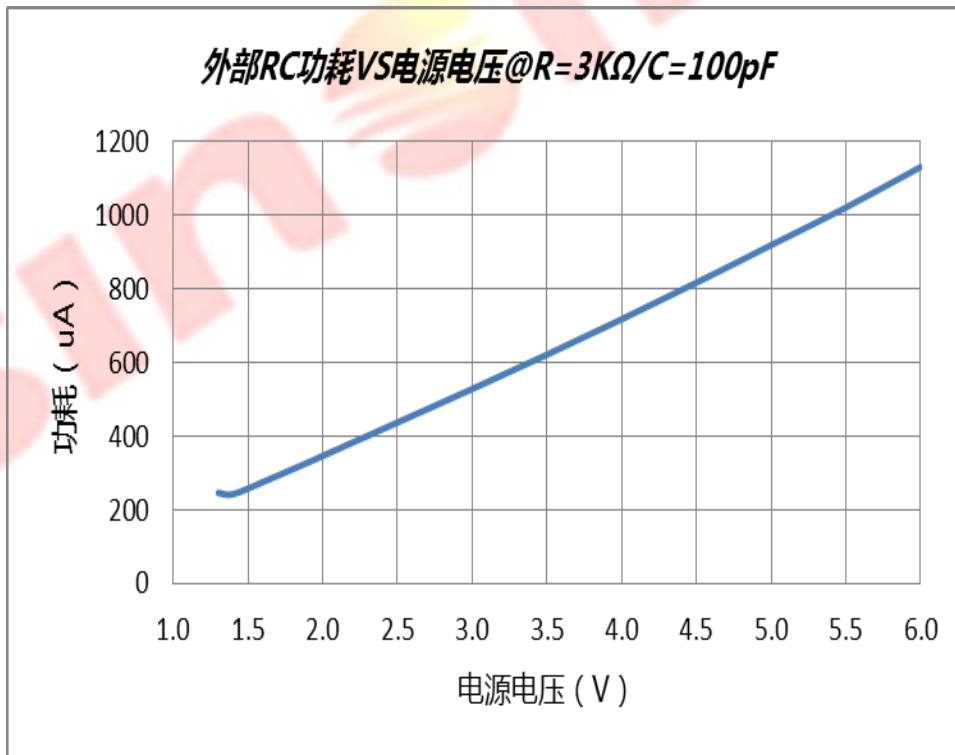
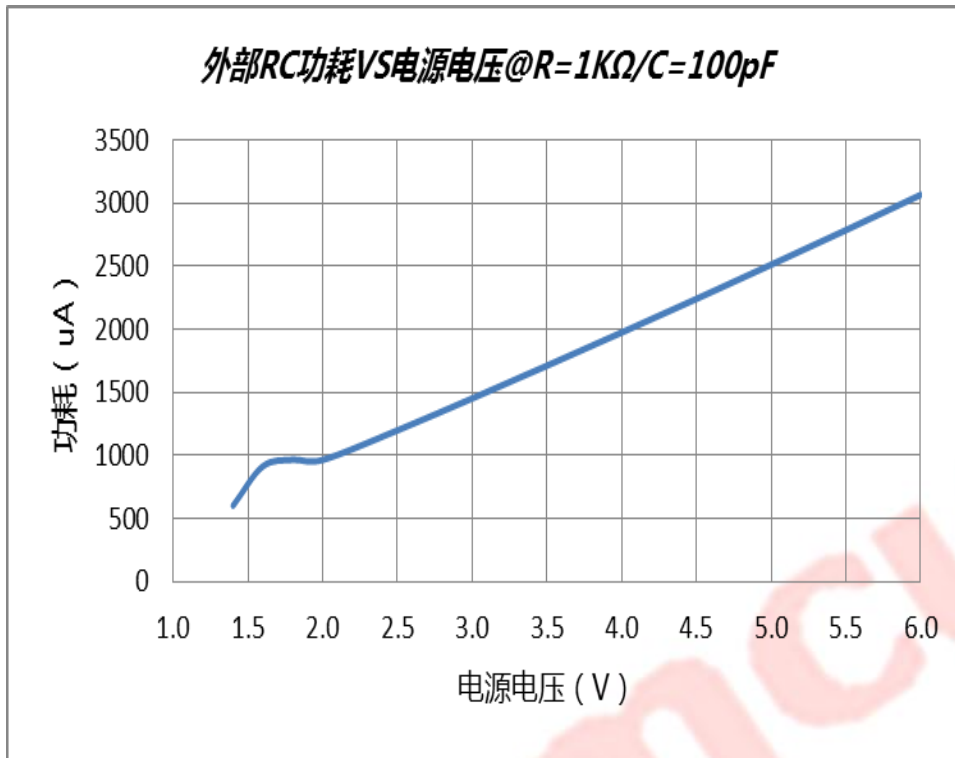
11.5 外部 RC 时钟源 VS 电源电压

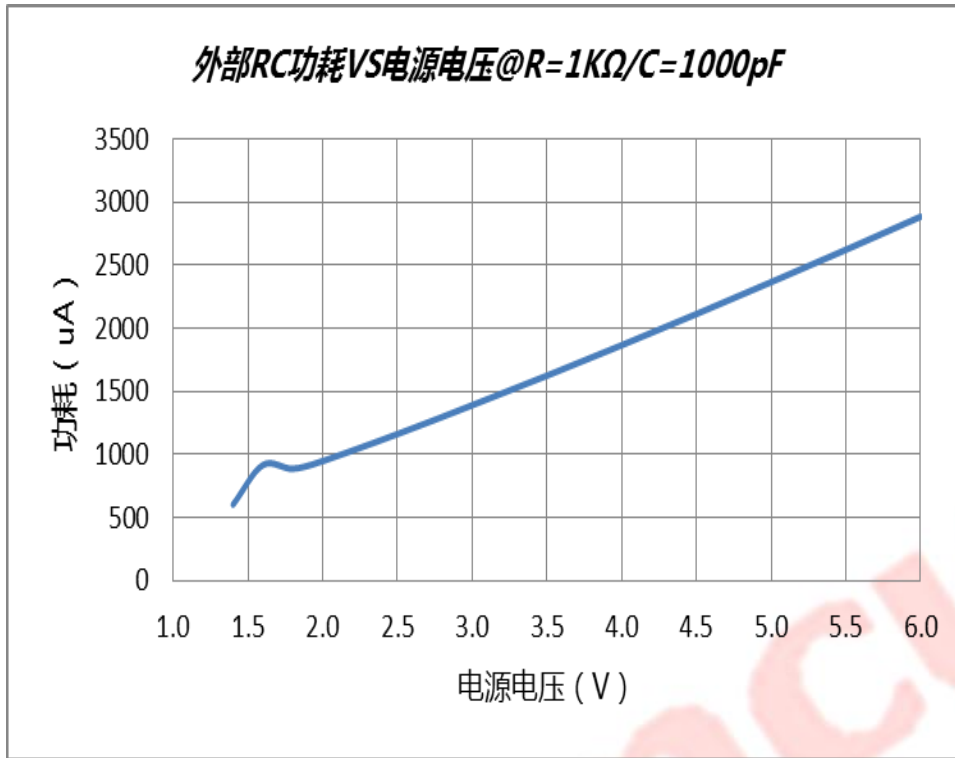
外部RC时钟源频率 VS 电源电压



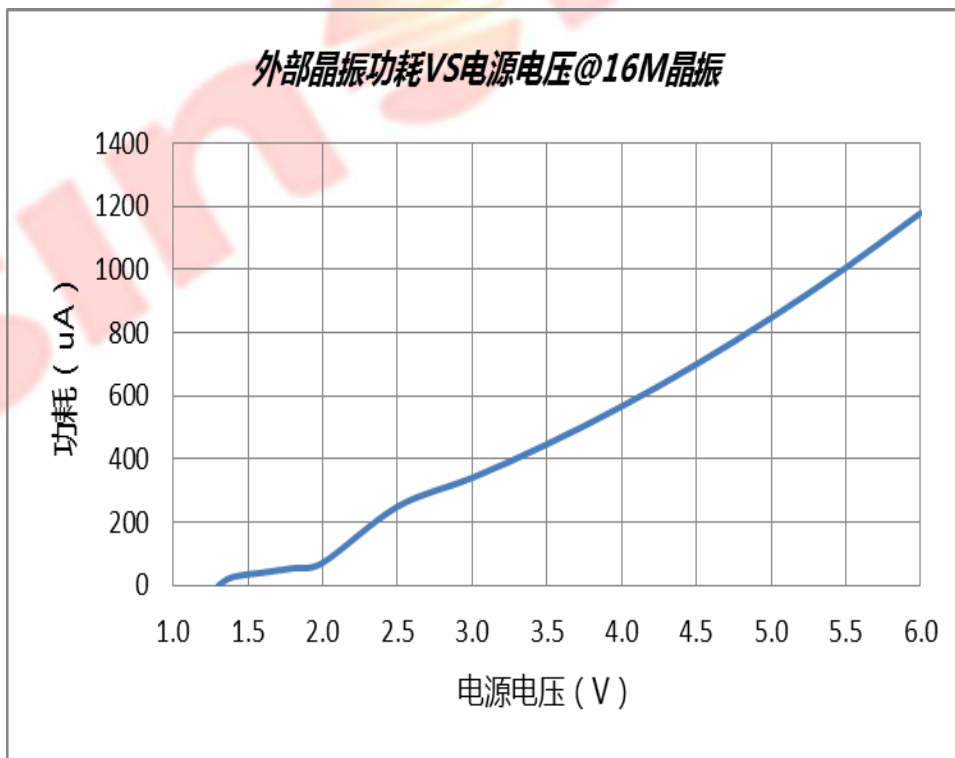


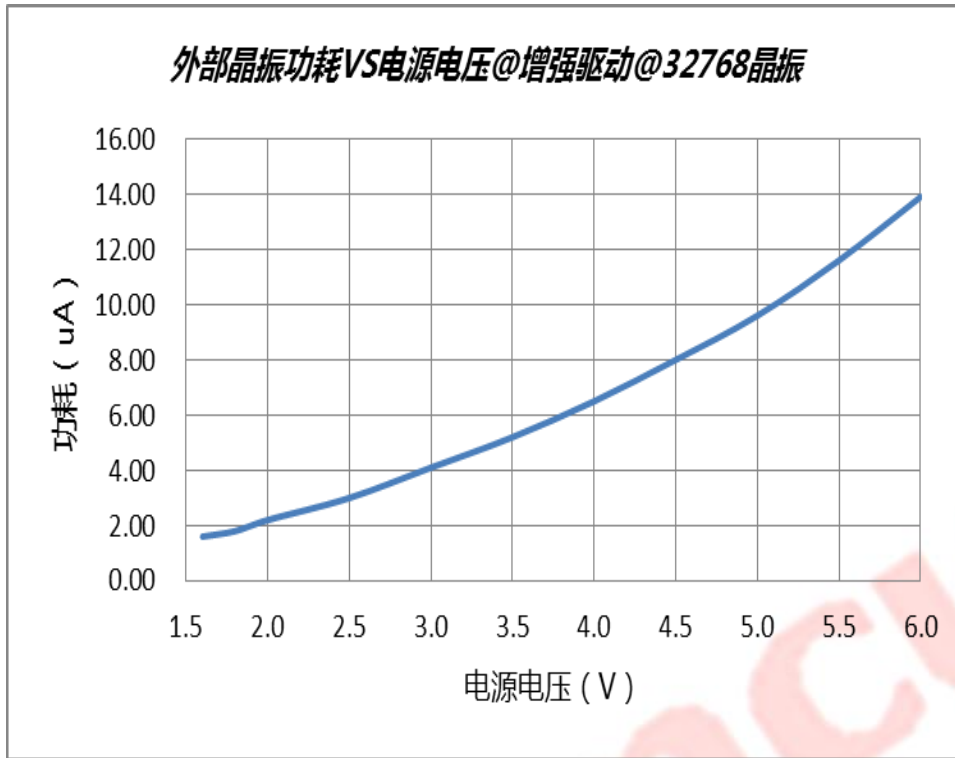
外部RC时钟源功耗 VS 电源电压





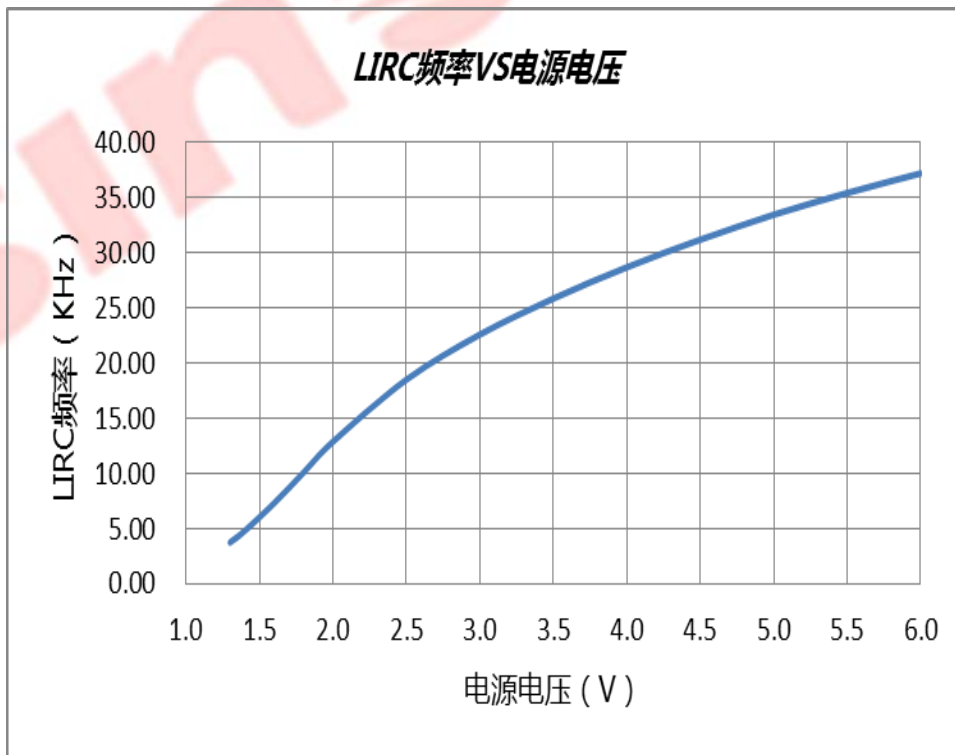
11.6 外部晶振功耗 VS 电源电压



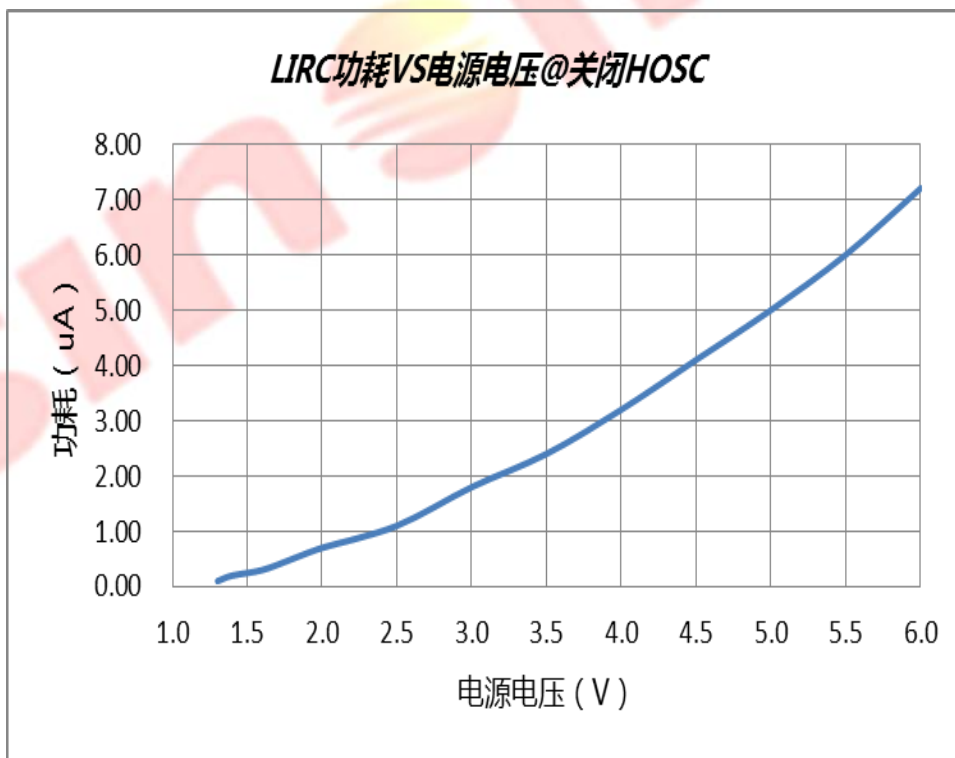
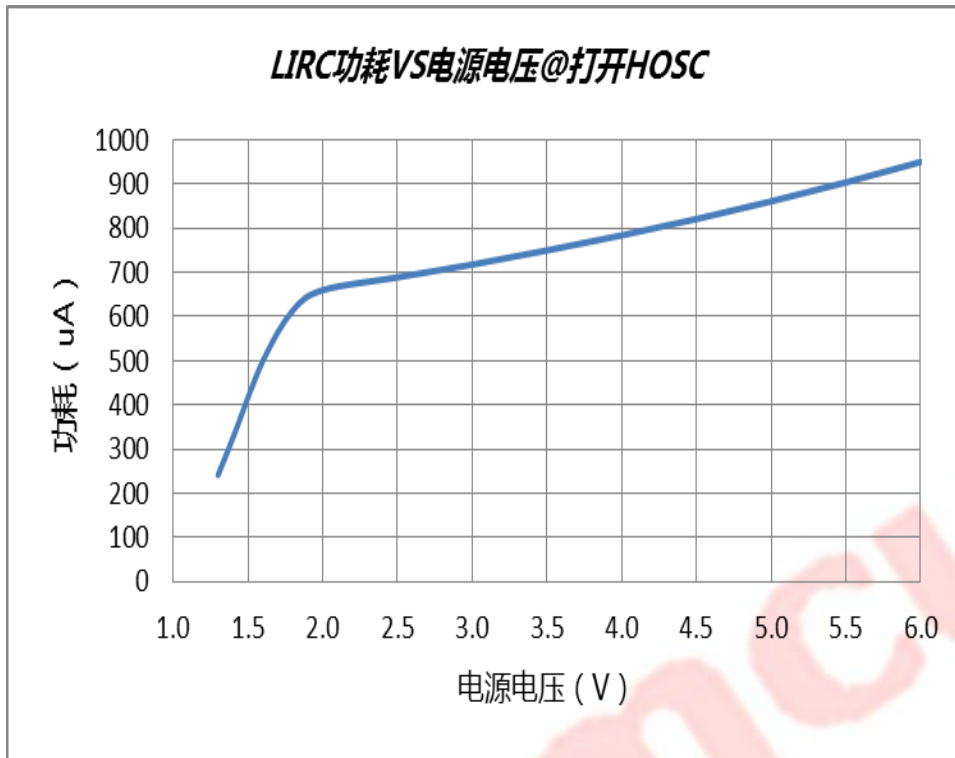


11.7 内部低频 RC VS 电源电压

LIRC频率 VS 电源电压



LIRC功耗 VS 电源电压

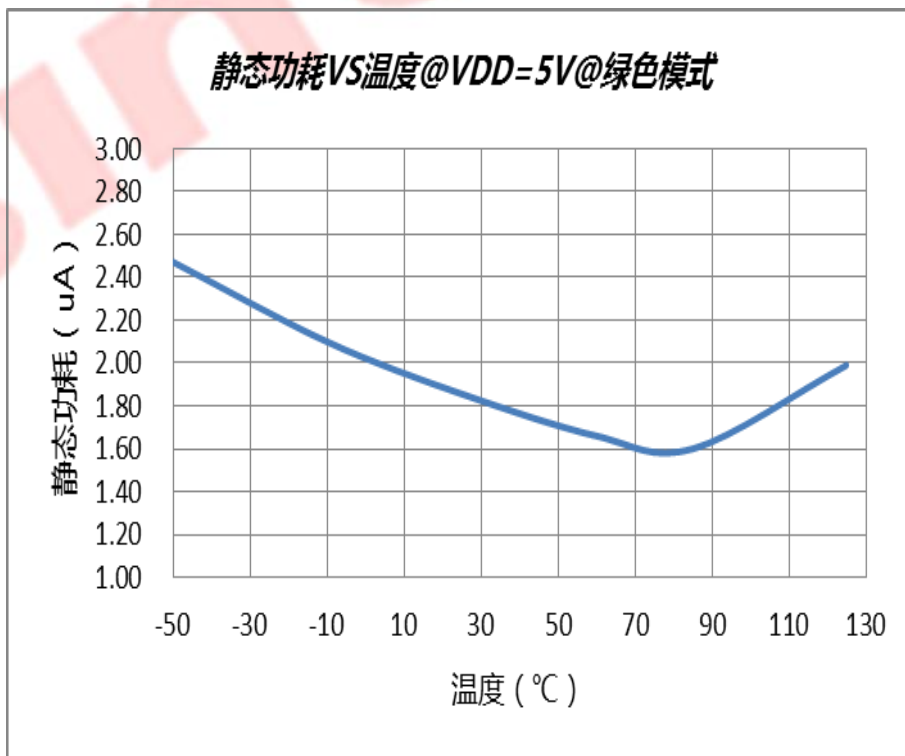


11.8 静态功耗 VS 温度

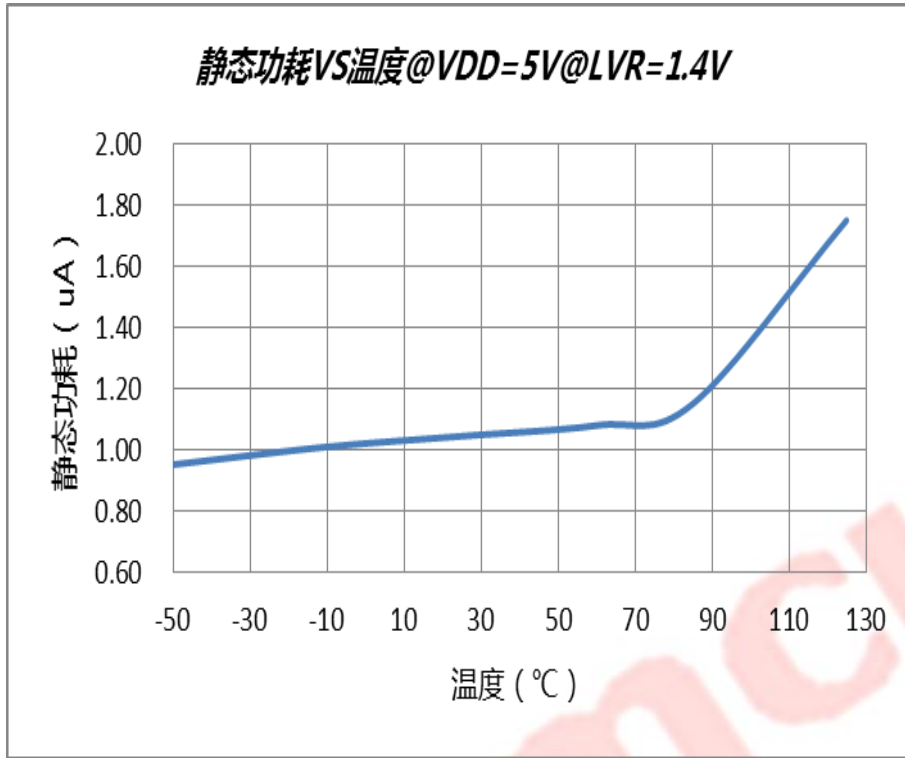
休眠模式静态功耗 VS 温度



空闲模式静态功耗 VS 温度

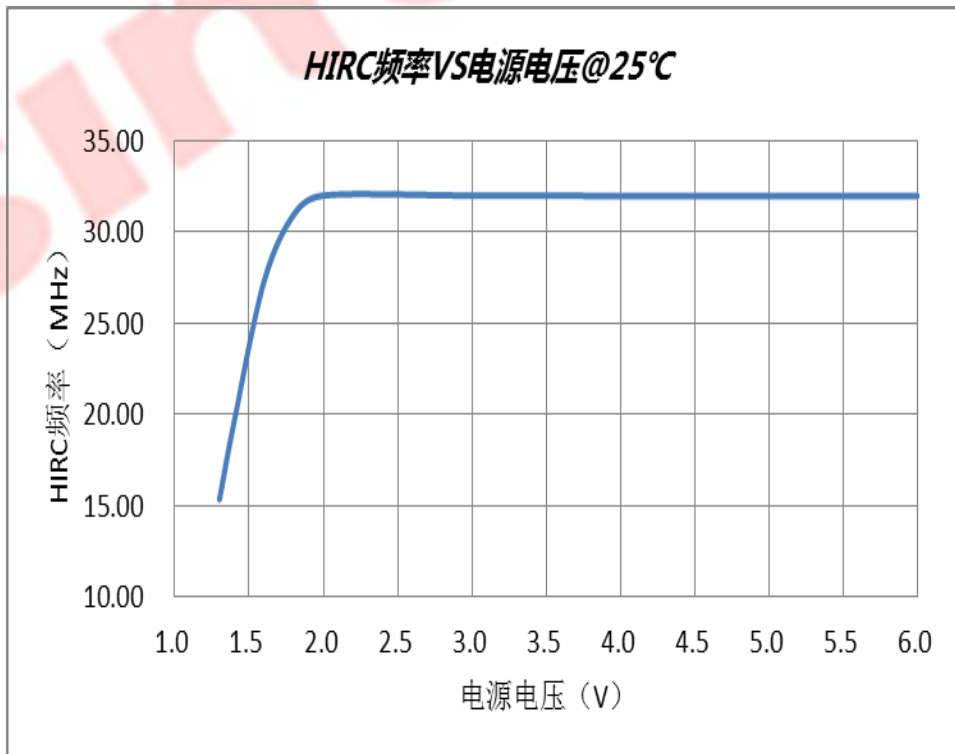


LVR模块静态功耗 VS 温度

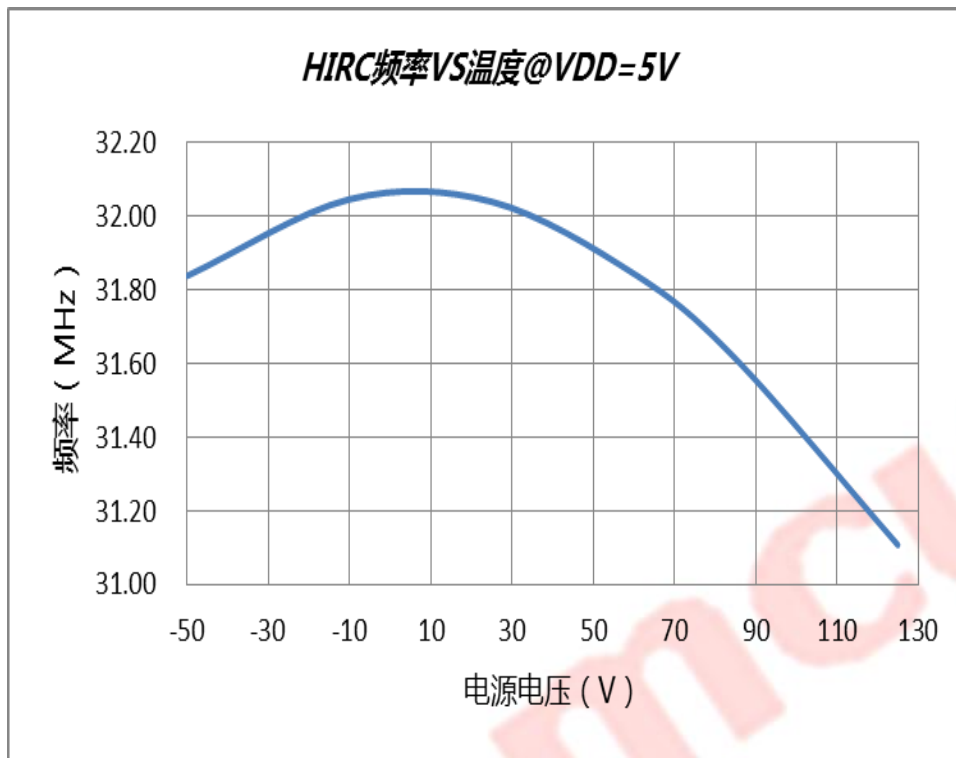


11.9 内部高频 RC VS 电源电压/温度

HIRC频率 VS 电源电压



HIRC频率 VS 温度



12 封装外形尺寸



13 版本修订记录

| 版本号 | 修订日期 | 修订内容 |
|------|------------|------|
| V1.0 | 2017-07-28 | 初版发布 |
| | | |