

MC32P5213 用户手册

(原产品名 MC33P78)

SinoMCU 8 位单片机

2014/03/10



SinoMCU
晟矽微电子

上海晟矽微电子股份有限公司

Shanghai SinoMCU Microelectronics Co., Ltd.

目录

| | | |
|-----|-------------------|----|
| 1 | 产品简介 | 4 |
| 1.1 | 产品特性 | 4 |
| 1.2 | 系统框图 | 5 |
| 1.3 | 引脚排列 | 6 |
| 1.4 | 引脚说明 | 7 |
| 1.5 | 管脚结构 | 7 |
| 2 | 中央处理器 | 11 |
| 2.1 | 指令集 | 11 |
| 2.2 | 程序存储器 | 14 |
| 2.3 | 数据存储器 | 15 |
| 2.4 | 堆栈 | 16 |
| 2.5 | 烧录配置选项 | 17 |
| 2.6 | 控制寄存器 | 19 |
| 3 | 系统时钟 | 23 |
| 3.1 | 外接晶体振荡器 | 23 |
| 3.2 | 内置高频 RC 振荡器 | 23 |
| 3.3 | 内置低频 RC 振荡器 | 23 |
| 3.4 | 工作模式 | 24 |
| 3.5 | 低功耗模式 | 27 |
| 4 | 复位 | 28 |
| 4.1 | 复位条件 | 28 |
| 4.2 | 上电复位 | 29 |
| 4.3 | 外部复位 | 29 |
| 4.4 | 低电压复位 | 29 |
| 4.5 | 看门狗复位 | 29 |
| 5 | I/O 口 | 30 |
| 5.1 | IO 工作模式 | 30 |
| 5.2 | 上拉电阻控制 | 31 |
| 5.3 | 端口模式控制 | 31 |
| 6 | 定时器 | 32 |
| 6.1 | 看门狗定时器 WDT | 32 |
| 6.2 | 定时器 T0 | 32 |
| 6.3 | 定时器 T1 | 35 |
| 7 | 键盘扫描模块 | 38 |
| 7.1 | 键盘扫描介绍 | 38 |
| 7.2 | 键盘扫描操作步骤 | 38 |
| 7.3 | 键盘扫描相关寄存器 | 39 |
| 8 | OPA | 41 |
| 8.1 | OPA 模块介绍 | 41 |
| 8.2 | OPA 操作步骤 | 41 |
| 8.3 | OPA 相关寄存器 | 42 |
| 9 | RAM 电源 | 43 |

| | | |
|-------|-------------------------------|----|
| 9.1 | RAM 电源外置 | 43 |
| 9.2 | RAM 电源内置 | 44 |
| 10 | 中断 | 45 |
| 10.1 | 外中断 | 45 |
| 10.2 | 定时器中断 | 45 |
| 10.3 | 中断相关寄存器 | 46 |
| 11 | 电气参数 | 48 |
| 11.1 | 极限参数 | 48 |
| 11.2 | 直流特性参数 | 48 |
| 11.3 | 交流电气参数 | 52 |
| 12 | 特性曲线图 | 53 |
| 12.1 | IROUT 驱动电流 VS 输出电压 | 53 |
| 12.2 | LED 驱动电流 VS 输出电压 | 53 |
| 12.3 | IO 输出低电平驱动电流 VS 输出电压 | 54 |
| 12.4 | IO 输出高电平驱动电流 VS 输出电压 | 54 |
| 12.5 | IO 口上拉电阻 VS 电源电压 | 55 |
| 12.6 | IO 口输入高电平 VS 电源电压 | 56 |
| 12.7 | IO 口输入低电平 VS 电源电压 | 57 |
| 12.8 | WDT 溢出时间 VS 电源电压 | 57 |
| 12.9 | 键盘扫描时间 VS 电源电压 | 58 |
| 12.10 | 键盘扫描静态功耗 VS 电源电压 | 58 |
| 12.11 | 动态功耗 VS 电源电压 | 59 |
| 12.12 | HOLD 模式 1 功耗 VS 电源电压 | 63 |
| 12.13 | HOLD 模式 2 功耗 VS 电源电压 | 64 |
| 12.14 | 休眠模式功耗 VS 电源电压 | 64 |
| 12.15 | 常温高频振荡频率 VS 电源电压 | 65 |
| 12.16 | 常压高频振荡频率 VS 温度 | 65 |
| 12.17 | 高频振荡频率 VS 电源电压 VS 温度 | 66 |
| 12.18 | 低电压复位电压 VS 温度 | 66 |
| 12.19 | 最低工作电压 VS 系统时钟 FCPU 关系图 | 67 |
| 13 | 封装外形尺寸 | 68 |
| 14 | 版本修订记录 | 69 |

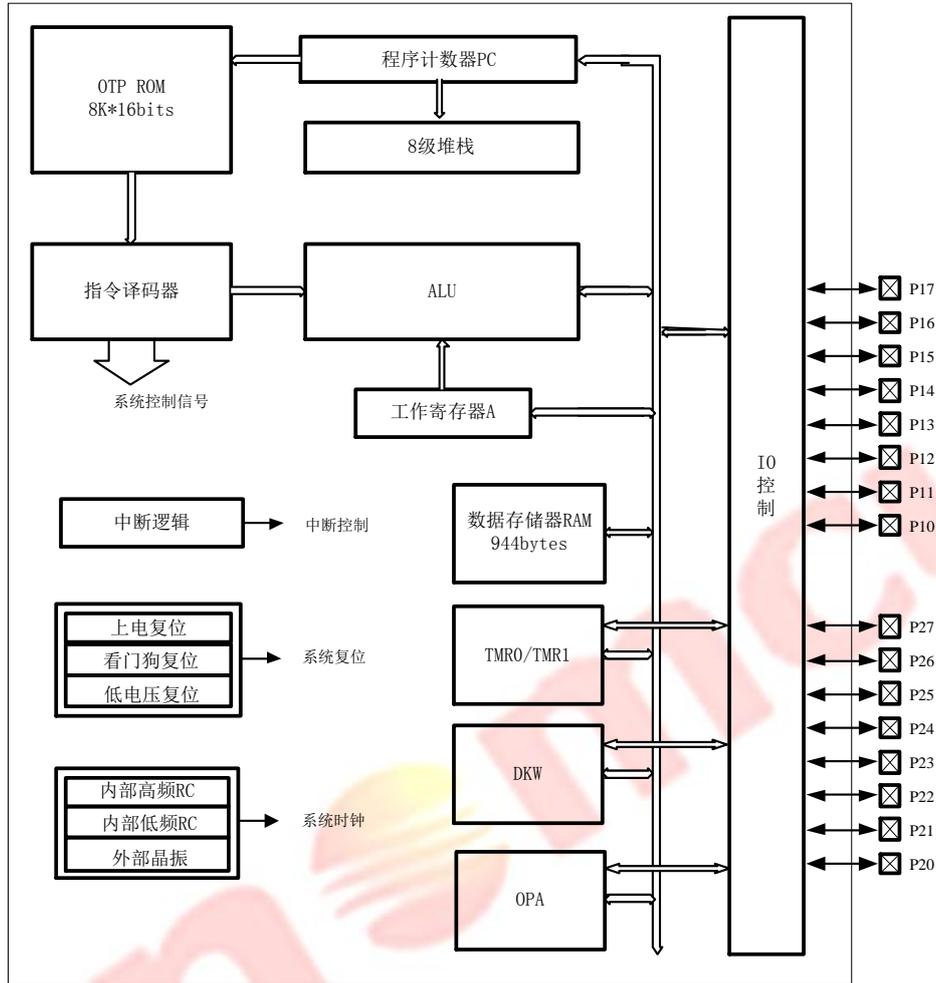
1 产品简介

MC32P5213 是一款 OTP 型 MCU 产品，其红外信号学习功能可以为学习型遥控器产品提供良好的解决方案。

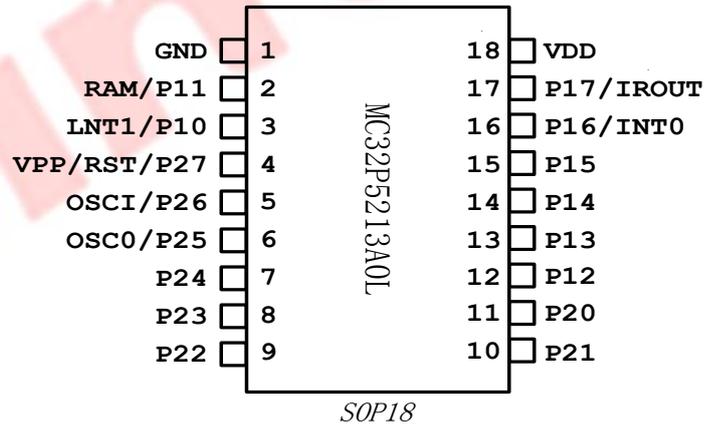
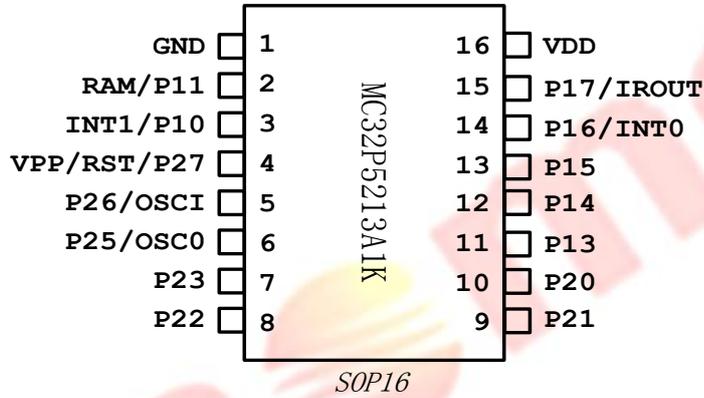
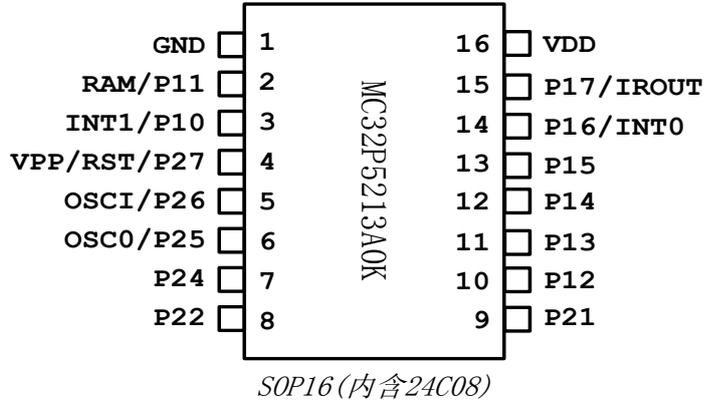
1.1 产品特性

- ◇ 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
 - ✓ 8K*16 程序存储器空间，8 级深度硬件堆栈，通过 INDF3 可读取 ROM 区内容
 - ✓ 944 字节 RAM
- ◇ 16 个 IO 口 (RAM 和 VPP 管脚开漏)
 - ✓ 8 位 P1 端口，其中 P11 复用 RAM 供电，P17 复用 IROUT
 - ✓ 8 位 P2 端口，其中 P27 复用 VPP，P25/P26 复用晶振口，P20/P23 可设置不同上拉电阻，P21/P22 可设置低电平输出限流功能
- ◇ 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
- ◇ 定时器
 - ✓ 内部自振式看门狗计数器 (WDT)
 - ✓ 1 个带捕捉功能的 16 位定时器 T0，可设置溢出中断
 - ✓ 1 个带 PWM 功能的 8 位定时器 T1，可设置溢出中断 (PWM 最小可调制脉宽宽度 62.5ns)
- ◇ 内置灵敏度可调的 OPA，实现红外信号放大作用
- ◇ 内置驱动电流可变的大电流输出管脚 IROUT (125mA、250mA、375mA、500mA)
- ◇ 中断
 - ✓ 两路外部中断源 (INT0、INT1)
 - ✓ 定时器 T0 中断
 - ✓ 定时器 T1 中断
- ◇ 时钟振荡模式
 - ✓ 外接高频晶体振荡器 HOSC (432KHz-16MHz) + 内置低频振荡器 LIRC (8KHz)
 - ✓ 内置高频振荡器 HIRC (16.12MHz) + 内置低频振荡器 LIRC (8KHz)
- ◇ 8 级低电压复位 LVR
- ◇ 工作电压
 - ✓ 2.7V-3.6V @FCPU=8.06MHz (内置高频振荡器)
 - ✓ 2.0V-3.6V @FCPU=4.03MHz (内置高频振荡器)
 - ✓ 1.8V-3.6V @FCPU=2.015MHz (内置高频振荡器)
- ◇ 封装形式：SOP18、SOP16

1.2 系统框图



1.3 引脚排列



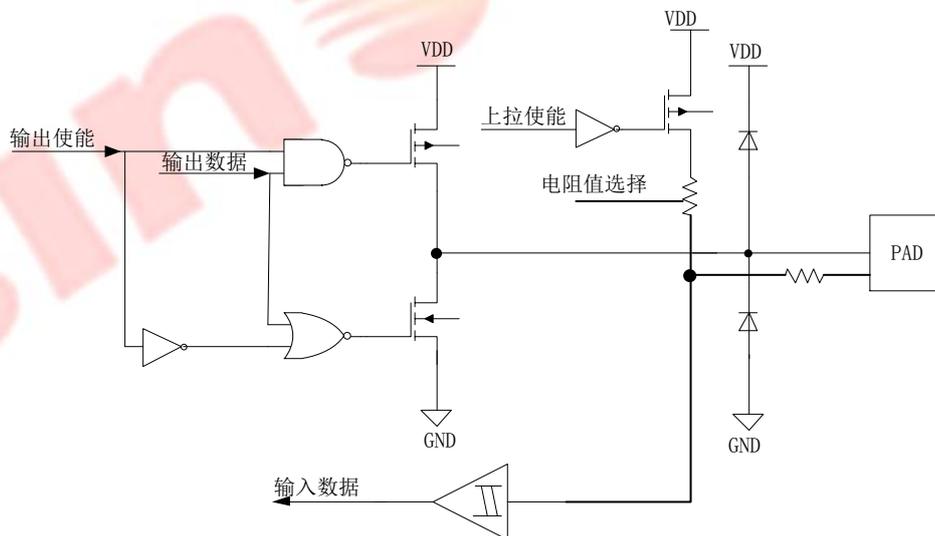
注：中间 SOP16 内部集成 24C08，使用方法详见《MC32P5213 内置 EEPROM 使用说明》文件

1.4 引脚说明

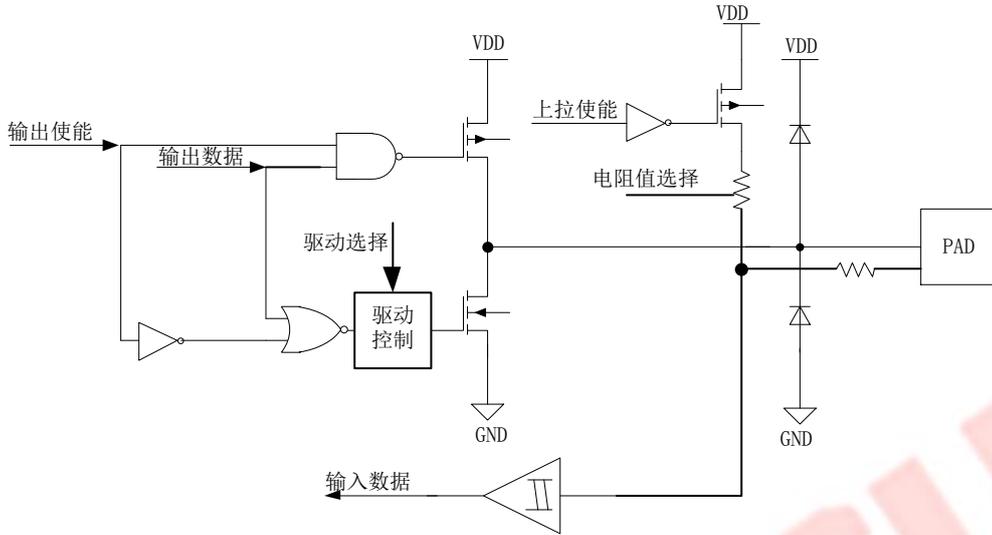
| 编号 | 引脚名 | 方向 | 功能描述 |
|----|----------------|--------|---|
| 1 | VDD | SOURCE | 电源 |
| 2 | P11 | I/O | 双向 IO 口, 可上拉, 键盘扫描管脚, RAM 供电管脚, 开漏输出 |
| 3 | P10/P16 | I/O | 双向 IO 口, 可上拉, 键盘扫描管脚, 外部中断管脚 |
| 4 | P27 | I/O | 双向 IO 口, 可上拉, 外部复位脚 RST, 编程高压输入 VPP, 开漏输出 |
| 5 | P26 | I/O | 双向 IO 口, 可上拉, 键盘扫描管脚, 晶振管脚 OSCO |
| 6 | P25 | I/O | 双向 IO 口, 可上拉, 键盘扫描管脚, 晶振管脚 OSCI |
| 7 | P24 P15-P12 | I/O | 双向 IO 口, 可上拉, 键盘扫描管脚 |
| 8 | P23/P20 | I/O | 双向 IO 口, 可上拉, 键盘扫描管脚, 10K 上拉可选择 |
| 9 | P22/P21 | I/O | 双向 IO 口, 可上拉, 键盘扫描管脚, 低电平驱动可选择 |
| 10 | P17 | I/O | 双向 IO 口, 可上拉, IROUT 管脚, OPA 输入管脚 |
| 11 | GND | SOURCE | 地 |

1.5 管脚结构

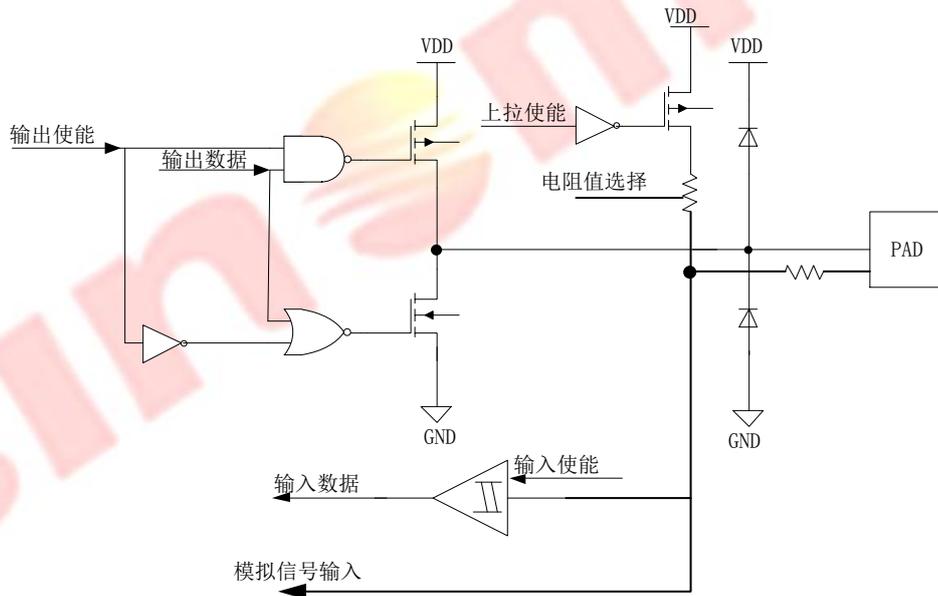
P10,P12-P16,P20,P23-P24



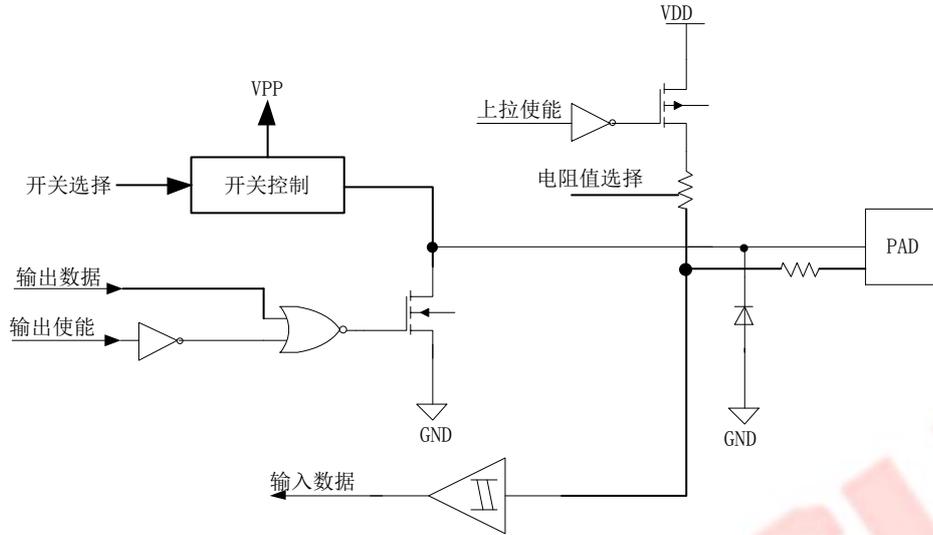
P21,P22



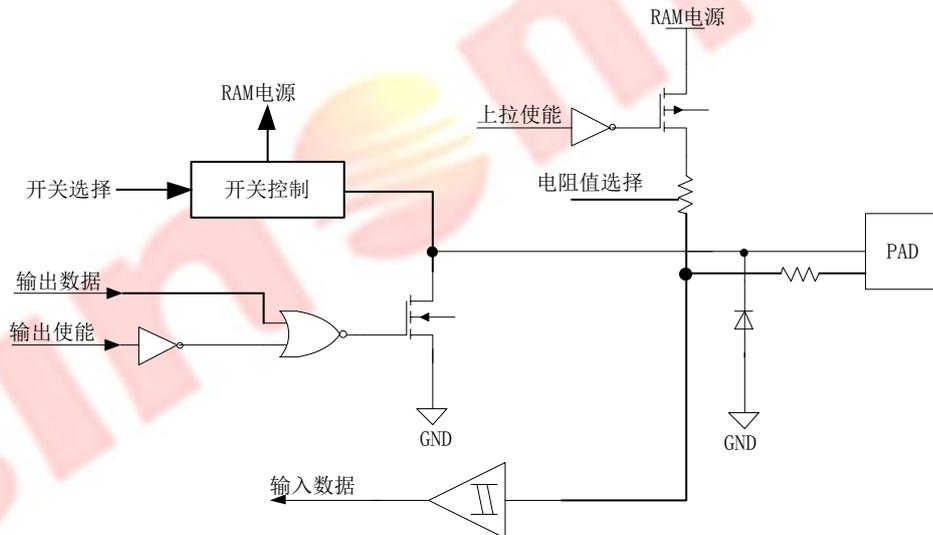
P25,P26

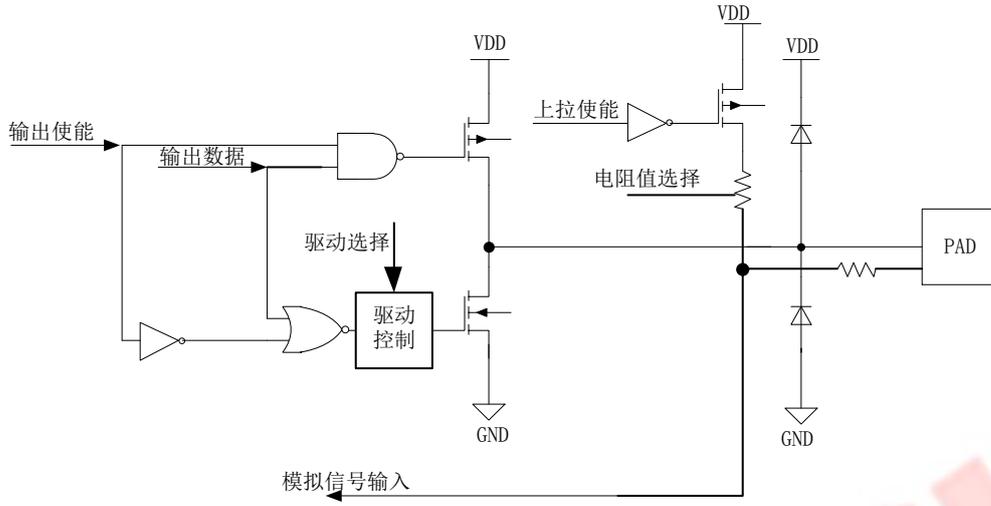


P27



P11





2 中央处理器

2.1 指令集

MC32P5213 的指令是精简指令集。指令集详细资料见本公司手册《MC33 指令集说明书》。

| 助记符 | 说明 | 操作 | 周期数 | 指令码 | 影响 |
|----------|------------------------------------|---|-----|---------------------|----------|
| ADDAR R | 寄存器 R 内容和 ACC 相加, 结果存到 ACC | R+ACC→ACC | 1 | 0111 110r rrrr rrrr | C, DC, Z |
| ADDRA R | 寄存器 R 内容和 ACC 相加, 结果存到 R | R+ACC→R | 1 | 0111 111r rrrr rrrr | C, DC, Z |
| ADCAR R | 带 C 标志的加法, 结果存到 ACC | R+ACC+C→ACC | 1 | 0001 000r rrrr rrrr | C, DC, Z |
| ADCRA R | 带 C 标志的加法, 结果存到 R | R+ACC+C→R | 1 | 0001 001r rrrr rrrr | C, DC, Z |
| RSUBAR R | 寄存器 R 内容和 ACC 相减, 结果存到 ACC | R-ACC→ACC | 1 | 0100 100r rrrr rrrr | C, DC, Z |
| RSUBRA R | 寄存器 R 内容和 ACC 相减, 结果存到 R | R-ACC→R | 1 | 0100 101r rrrr rrrr | 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 | 0001 100r rrrr rrrr | 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 | 0001 101r rrrr rrrr | C, DC, Z |
| ASUBAR R | ACC 和寄存器 R 内容相减, 结果存到 ACC | ACC-R→ACC | 1 | 0000 100r rrrr rrrr | C, DC, Z |
| ASUBRA R | ACC 和寄存器 R 内容相减, 结果存到 R | ACC-R→R | 1 | 0000 101r rrrr rrrr | 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 | 0000 010r rrrr rrrr | 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 | 0000 011r rrrr rrrr | 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 | 0111 100r rrrr rrrr | Z |
| ANDRA R | 寄存器 R 内容和 ACC 与操作, 结果存到 R | R and ACC→R | 1 | 0111 101r rrrr rrrr | Z |
| ORAR R | 寄存器 R 内容和 ACC 或操作, 结果存到 ACC | R or ACC→ACC | 1 | 0101 110r rrrr rrrr | Z |
| ORRA R | 寄存器 R 内容和 ACC 或操作, 结果存到 R | R or ACC→R | 1 | 0101 111r rrrr rrrr | 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 | 0100 000r rrrr rrrr | 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 | 0100 001r rrrr rrrr | Z |
| COMAR R | 对 R 取反, 结果存到 ACC | R 取反→ACC | 1 | 0111 000r rrrr rrrr | Z |
| COMR R | 对 R 取反, 结果存到 R | R 取反→R | 1 | 0111 001r rrrr rrrr | Z |
| CLRA | 对 ACC 清零 | 0→ACC | 1 | 0111 0100 0000 0000 | Z |
| CLRR R | 对 R 清零 | 0→R | 1 | 0111 011r rrrr rrrr | Z |
| RLA | ACC 循环左移(带 C 标志) | ACC[7]→C ACC[6:0]→ACC[7:1] C→ACC[0] | 1 | 0000 0000 0001 0100 | 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 | 0101 000r rrrr rrrr | 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 | 0101 001r rrrr rrrr | C |

| | | | | | |
|-----------|-----------------------------------|---|----------------------|---------------------|---|
| RRA | ACC 循环右移(带 C 标志) | C→ACC[7] ACC[7:1]→ACC[6:0] ACC[0]→C | 1 | 0000 0000 0001 0101 | 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 | 0100 110r rrrr rrrr | 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 | 0100 111r rrrr rrrr | 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 | 0100 010r rrrr rrrr | - |
| 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 | 0100 011r rrrr rrrr | - |
| MULAR R | 寄存器 R 内容和 ACC 相乘, 结果存到 HIBYTE:ACC | R*ACC→ HIBYTE:ACC ⁽²⁾ | 1 | 0001 010r rrrr rrrr | - |
| MULRA R | 寄存器 R 内容和 ACC 相乘 结果存到 HIBYTE:R | R*ACC→HIBYTE:R ⁽²⁾ | 1 | 0001 011r rrrr rrrr | - |
| MOVAR R | 将 R 存到 ACC | R→ACC | 1 | 0101 100r rrrr rrrr | Z |
| MOVR R | 将 R 存到 R | R→R | 1 | 0101 101r rrrr rrrr | Z |
| MOVRA R | 将 ACC 存到 R | ACC→R | 1 | 0101 011r rrrr rrrr | - |
| INCA | ACC 自加 1 | ACC+1→ACC | 1 | 0000 0000 0001 0011 | - |
| INCAR R | R 加 1, 结果存到 ACC | R+1→ACC | 1 | 0110 010r rrr rrrr | Z |
| INCR R | R 加 1, 结果存到 R | R+1→R | 1 | 0110 011r rrr rrrr | Z |
| DECA | ACC 自减 1 | ACC-1→ACC | 1 | 0000 0000 0001 0001 | - |
| DECAR R | R 减 1, 结果存到 ACC | R-1→ACC | 1 | 0110 110r rrrr rrrr | Z |
| DECR R | R 减 1, 结果存到 R | R-1→R | 1 | 0110 111r rrrr rrrr | 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 ⁽¹⁾ | 0000 0000 0001 0010 | - |
| 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 ⁽¹⁾ | 0110 000r rrrr rrrr | - |
| 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 ⁽¹⁾ | 0110 001r rrrr rrrr | - |
| 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 ⁽¹⁾ | 0000 0000 0001 0000 | - |
| 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 ⁽¹⁾ | 0110 100r rrrr rrrr | - |
| 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 ⁽¹⁾ | 0110 101r rrrr rrrr | - |
| BCLR R,b | 对 R 的第 b 位清零 | 0→R[b] | 1 | 1101 bbbr rrrr rrrr | - |
| BSET R,b | 对 R 的第 b 位置 1 | 1→R[b] | 1 | 1100 bbbr rrrr rrrr | - |
| 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 ⁽¹⁾ | 1111 bbbr rrrr rrrr | - |
| 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 ⁽¹⁾ | 1110 bbbr rrrr rrrr | - |

| | | | | | |
|----------|---------------------------------|------------------------|---|---------------------|----------|
| ADDAI K | 立即数 K 和 ACC 相加, 结果存到 ACC | K+ACC→ACC | 1 | 0011 1111 kkkk kkkk | C, DC, Z |
| ADCAI K | 立即数 K 和 ACC 相加(带 C 标志) 结果存到 ACC | K and ACC→ACC | 1 | 0011 1000 kkkk kkkk | C, DC, Z |
| ISUBAI K | 立即数和 ACC 相减, 结果存到 ACC | K-ACC→ACC | 1 | 0011 1010 kkkk kkkk | 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 | 0011 0111 kkkk kkkk | C, DC, Z |
| ASUBAI K | ACC 和立即数相减, 结果存到 ACC | ACC-K→ACC | 1 | 0011 0101 kkkk kkkk | C, DC, Z |
| ANDAI K | 立即数 K 和 ACC 与操作, 结果存到 ACC | K and ACC→ACC | 1 | 0011 1110 kkkk kkkk | Z |
| ORAI K | 立即数 K 和 ACC 或操作, 结果存到 ACC | K or ACC→ACC | 1 | 0011 1101 kkkk kkkk | Z |
| XORAI K | 立即数和 ACC 异或, 结果存到 ACC | K xor ACC→ACC | 1 | 0011 1001 kkkk kkkk | Z |
| MOVAI K | 将立即数存到 ACC | K→ACC | 1 | 0011 1100 kkkk kkkk | - |
| RETAI K | 从子程序返回, 并将立即数存到 ACC | TOS→PC K→ACC | 2 | 0011 1011 kkkk kkkk | - |
| RETURN | 从子程序返回 | TOS→PC | 2 | 0000 0000 0000 1100 | - |
| RETIE | 从中断返回 | TOS→PC 1→GIE | 2 | 0000 0000 0000 1101 | - |
| CALL K | 子程序调用 | PC+1→TOS K→PC[12:0] | 2 | 100k kkkk kkkk kkkk | - |
| GOTO K | 无条件跳转 | K→PC[12:0] | 2 | 101k kkkk kkkk kkkk | - |
| NOP | 空操作 | 空操作 | 1 | 0000 0000 0000 0000 | - |
| DAA | 加法后, 将 ACC 的值调整到十进制 | ACC(十六进制)→ ACC(十进制) | 1 | 0000 0000 0000 1010 | 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 | 0000 0000 0000 1001 | - |
| CLRWDI | 清看门狗定时器 | 0→WDT | 1 | 0000 0000 0000 1110 | TO, PD |
| STOP | 进入休眠模式 | 0→WDT 进入休眠模式 | 1 | 0000 0000 0000 1011 | TO, PD |

2.2 程序存储器

8K*16BIT 的程序存储器空间，程序存储器空间（0000H - 1FFFH）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 - 1FFFH) |
| 厂商保留区 (2000H - 7FFFH) |
| OPBIT0 (8000H) |
| OPBIT1 (8001H) |

注：1FF0H-1FFFH 只能存放表格，通过 INDF3 读出，无法填写程序

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

数据寄存器分为三个区，快速通用寄存器区GPR（256字节空间）和特殊功能寄存器区SFR，扩展寄存器区必须使用间接寻址模式2进行寻址，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 - 1AFH | 通用数据区 (INDF1/INDF2) | | | | | | | |
| 1B0H - 1B7H | INDF0 | INDF1 | INDF2 | HIBYTE | FSR0 | FSR1 | PCL | PFALG |
| 1B8H - 1BFH | MCR | INDF3 | INTE0 | INTF0 | OSCM | | | |
| 1C0H - 1C7H | | | | | | | | |
| 1C8H - 1CFH | IOP1 | OEP1 | PUP1 | | | | DKWP1 | |
| 1D0H - 1D7H | IOP2 | OEP2 | PUP2 | | | | DKWP2 | |
| 1D8H - 1DFH | T0CR | T0LOADH | T0LOADL | T0LATRH | T0LATRL | T0LATFH | T0LATFL | |
| 1E0H - 1E7H | T1CR | T1DATA | T1LOAD | | OPCR0 | OPCR1 | DWK0 | DWK1 |
| 1E8H - 1EFH | | | | | | | | |
| 1F0H - 1F7H | | | | | | | | |
| 1F8H - 1FFH | | | | | | | | |
| 200H-3FFH | 数据区 (INDF2) | | | | | | | |

注：上表中灰色部分数据存储器区地址未用，读出数据为0

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。MC32P5213的OPBIT定义如下。

OPBIT0:

| 位 | 符号 | 功能说明 |
|-----------------------|-------|---|
| BIT[1:0] | 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 |
| BIT13、BIT[3:2] | WDTT | WDT 溢出时间选择位 000：上电延时=WDT 溢出时间=16mS 001：上电延时=WDT 溢出时间=64mS 010：上电延时=WDT 溢出时间=256mS 011：上电延时=WDT 溢出时间=1024mS 100：上电延时=16mS，WDT 溢出时间=2048mS 101：上电延时=64mS，WDT 溢出时间=4096mS 110：上电延时=256mS，WDT 溢出时间=8192mS 111：上电延时=1024mS，WDT 溢出时间=16384mS |
| BIT[6 : 4] | 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 001：机器周期 FCPU 为 4 个高速时钟周期 FHOSC 010：机器周期 FCPU 为 8 个高速时钟周期 FHOSC 011：机器周期 FCPU 为 16 个高速时钟周期 FHOSC 100：机器周期 FCPU 为 32 个高速时钟周期 FHOSC 101：机器周期 FCPU 为 64 个高速时钟周期 FHOSC 110：机器周期 FCPU 为 128 个高速时钟周期 FHOSC 111：机器周期 FCPU 为 256 个高速时钟周期 FHOSC |
| BIT[7] | MCLRE | 外部复位使能位 0：不使能外部复位，P27 作为 IO 1：使能外部复位，P27 作为复位引脚 |
| BIT[8] | 未用 | - |
| BIT[9] | FOSCS | 时钟选择位 0：内部 16.12MHz 振荡器作为高频时钟 FHIRC 1：外部高频晶体振荡器作为高频时钟 FHXT |

| | | |
|-------------------|-------|--|
| BIT[12:10] | VLVRS | 低电压复位电压选择位 000 : LVR 电压=2.9V (参考值) 001 : LVR 电压=2.7V (参考值) 010 : LVR 电压=2.5V (参考值) 011 : LVR 电压=2.3V (参考值) 100 : LVR 电压=2.1V (参考值) 101 : LVR 电压=1.9V (参考值) 110 : LVR 电压=1.7V (参考值) 111 : LVR 电压=1.5V (参考值) |
| BIT[14] | RAMS | RAM 电源选择 0 : RAM 电源外接 1 : RAM 电源内置 |
| BIT[15] | 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 |
|------------------|----|------|
| BIT[15:0] | 保留 | - |

OPBIT2:

| 位 | 符号 | 功能说明 |
|------------------|-------|---|
| BIT[0] | 保留 | - |
| BIT[1] | HMSEL | 晶振模式选择位 0 : 455KHz 1 : 4MHz-16MHz |
| BIT[15:2] | 保留 | - |

2.6 控制寄存器

MC32P5213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 | - | - | - | - | INT1IE | INT0IE | T1IE | TOIE | 0--- 0000 |
| 1BBH | INTF | - | T0RF | - | - | INT1IF | INT0IF | T1IF | TOIF | X0-- XX00 |
| 1BCH | OSCM | - | - | STBL | STBH | - | CLKS | LFEN | HFEN | --X1 -000 |
| 1C8H | IOP1 | P17D | P16D | P15D | P14D | P13D | P12D | P11D | P10D | XXXX XXXX |
| 1C9H | OEP1 | P17OE | P16OE | P15OE | P14OE | P13OE | P12OE | P11OE | P10OE | 0000 0000 |
| 1CAH | PUP1 | P17PU | P16PU | P15PU | P14PU | P13PU | P12PU | P11PU | P10PU | 0000 0000 |
| 1CEH | DKWP1 | - | P16DKW | P15DKW | P14DKW | P13DKW | P12DKW | P11DKW | P10DKW | -000 0000 |
| 1D0H | IOP2 | P27D | P26D | P25D | P24D | P23D | P22D | P21D | P20D | XXXX XXXX |
| 1D1H | OEP2 | P27OE | P26OE | P25OE | P24OE | P23OE | P22OE | P21OE | P20OE | 0000 0000 |
| 1D2H | PUP2 | P27PU | P26PU | P25PU | P24PU | P23PU | P22PU | P21PU | P20PU | 0000 0000 |
| 1D6H | DKWP2 | P27DKW | P26DKW | P25DKW | P24DKW | P23DKW | P22DKW | P21DKW | P20DKW | 0000 0000 |
| 1D8H | T0CR | TC0EN | T0MOD1 | T0MOD0 | T0PTS1 | T0PTS0 | T0PR2 | T0PR1 | T0PR0 | 0000 0000 |
| 1D9H | T0LOADH | T0LOADH7 | T0LOADH6 | T0LOADH5 | T0LOADH4 | T0LOADH3 | T0LOADH2 | T0LOADH1 | T0LOADH0 | 1111 1111 |
| 1DAH | T0LOADL | T0LOADL7 | T0LOADL6 | T0LOADL5 | T0LOADL4 | T0LOADL3 | T0LOADL2 | T0LOADL1 | T0LOADL0 | 1111 1111 |
| 1DBH | T0LATFL | T0LATFL7 | T0LATFL6 | T0LATFL5 | T0LATFL4 | T0LATFL3 | T0LATFL2 | T0LATFL1 | T0LATFL0 | 0000 0000 |
| 1DCH | T0LATFH | T0LATFH7 | T0LATFH6 | T0LATFH5 | T0LATFH4 | T0LATFH3 | T0LATFH2 | T0LATFH1 | T0LATFH0 | 0000 0000 |
| 1DDH | T0LATRL | T0LATRL7 | T0LATRL6 | T0LATRL5 | T0LATRL4 | T0LATRL3 | T0LATRL2 | T0LATRL1 | T0LATRL0 | 0000 0000 |
| 1DEH | T0LATRH | T0LATRH7 | T0LATRH6 | T0LATRH5 | T0LATRH4 | T0LATRH3 | T0LATRH2 | T0LATRH1 | T0LATRH0 | 0000 0000 |
| 1E0H | T1CR | TC1EN | T1MOD1 | T1MOD0 | T1PTS1 | T1PTS0 | T1PR2 | T1PR1 | T1PR0 | 0000 0000 |
| 1E1H | T1DATA | T1DATA7 | T1DATA6 | T1DATA5 | T1DATA4 | T1DATA3 | T1DATA2 | T1DATA1 | T1DATA0 | 0000 0000 |
| 1E2H | T1LOAD | T1LOAD7 | T1LOAD6 | T1LOAD5 | T1LOAD4 | T1LOAD3 | T1LOAD2 | T1LOAD1 | T1LOAD0 | 0000 0000 |
| 1E4H | OPCR0 | OPAE | OPAM | OPFC5 | OPFC4 | OPFC3 | OPFC2 | OPFC1 | OPFC0 | 0000 0000 |
| 1E5H | OPCR1 | IRIN | VSEL | - | - | FSEL1 | FSEL0 | CSEL1 | CSEL0 | 00-- 0000 |
| 1E6H | DKW0 | DKWE | IROS | IROT | WSEL1 | WSEL0 | RSEL | ISEL1 | ISEL0 | 0000 0000 |
| 1E7H | DKW1 | - | - | IRES1 | IRES0 | LSEL11 | LSEL10 | LSEL01 | LSEL00 | --00 0000 |

间接寻址寄存器0

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

BIT[7:0] **INDF0n** – 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 |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
| INDF1 | INDF17 | INDF16 | INDF15 | INDF14 | INDF13 | INDF12 | INDF11 | INDF10 |
| R/W | R/W |
| 初始值 | X | X | X | X | X | X | X | X |

BIT[7:0] **INDF1n** – 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 |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
| INDF2 | INDF27 | INDF26 | INDF25 | INDF24 | INDF23 | INDF22 | INDF21 | INDF20 |
| R/W | R/W |
| 初始值 | X | X | X | X | X | X | X | X |

BIT[7:0] **INDF2n** – 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 |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
| INDF3 | INDF37 | INDF36 | INDF35 | INDF34 | INDF33 | INDF32 | INDF31 | INDF30 |
| R/W | R/W |
| 初始值 | X | X | X | X | X | X | X | X |

BIT[7:0] **INDF3n** – 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 |
| 初始值 | X | X | X | X | X | X | X | X |

BIT[7:0] HIBYTE_n – 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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| FSR0 | FSR07 | FSR06 | FSR05 | FSR04 | FSR03 | FSR02 | FSR01 | FSR00 |
| R/W | R/W |
| 初始值 | X | X | X | X | X | X | X | X |

BIT[7:0] FSR0_n – 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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| FSR1 | FSR17 | FSR16 | FSR15 | FSR14 | FSR13 | FSR12 | FSR11 | FSR10 |
| R/W | R/W |
| 初始值 | X | X | X | X | X | X | X | X |

BIT[7:0] FSR1_n – 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 |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|
| PCL | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] PC_n – 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: $PC=PC+1$

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: $PC = (PC[9:0]+A[7:0])$ (对PCL操作的加法指令)

$PC = \{PC[9:8],ALU[7:0](ALU运算结果)\}$ (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 | - | - | - | - | - | Z | DC | C |
| R/W | - | - | - | - | - | R/W | R/W | R/W |
| 初始值 | - | - | - | - | - | 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 |
|------------|-------|-------|-------|-------|--------|--------|--------|--------|
| MCR | GIE | - | TO | PD | MINT11 | MINT10 | MINT01 | MINT00 |
| R/W | R/W | - | R | R | R/W | R/W | R/W | R/W |
| 初始值 | 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有相应的控制位决定

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** – 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

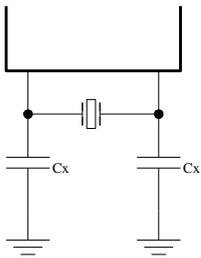
1x: INT0 电平变化触发

3 系统时钟

MC32P5213 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晶体振荡器或内置 16.12MHz 的高精度 RC 振荡器；低速时钟为内置低频 RC 振荡器。

系统选用高速时钟时 CPU 的机器周期 FCPU 由 OPBIT 的 FCPU 配置，选用低速时钟时 CPU 的机器周期 FCPU 为 2 个低频 RC 振荡周期。

3.1 外接晶体振荡器



外部晶体振荡器连接方式见下图。高频晶体可选用 432KHz~16MHz，通常 Cx 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 Cx 的推荐值。

| 晶体频率 | 电容 Cx (F) | 最低起振电压 (V) |
|--------|-------------|--------------|
| 16MHz | 10p | 2.5 |
| 8MHz | 15p | 1.8 |
| 4MHz | 15p/30p | 1.8 |
| 455KHz | 100p/220p | 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

MC32P5213 内置一个高精度高频 RC 振荡器 HIRC，振荡频率典型值 16.12MHz，该振荡器可用于系统高速时钟。

3.3 内置低频 RC 振荡器

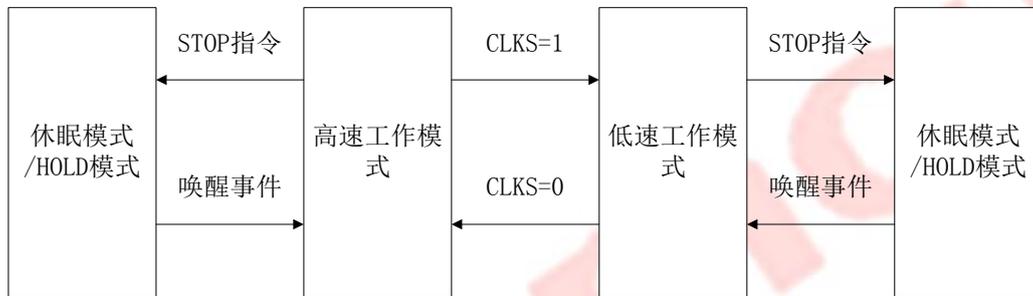
MC32P5213 内置一个低频 RC 振荡器 LIRC，振荡频率典型值 8KHz。该振荡器可用于系统低频时钟，同时用于上电延时、WDT，键盘扫描。

3.4 工作模式

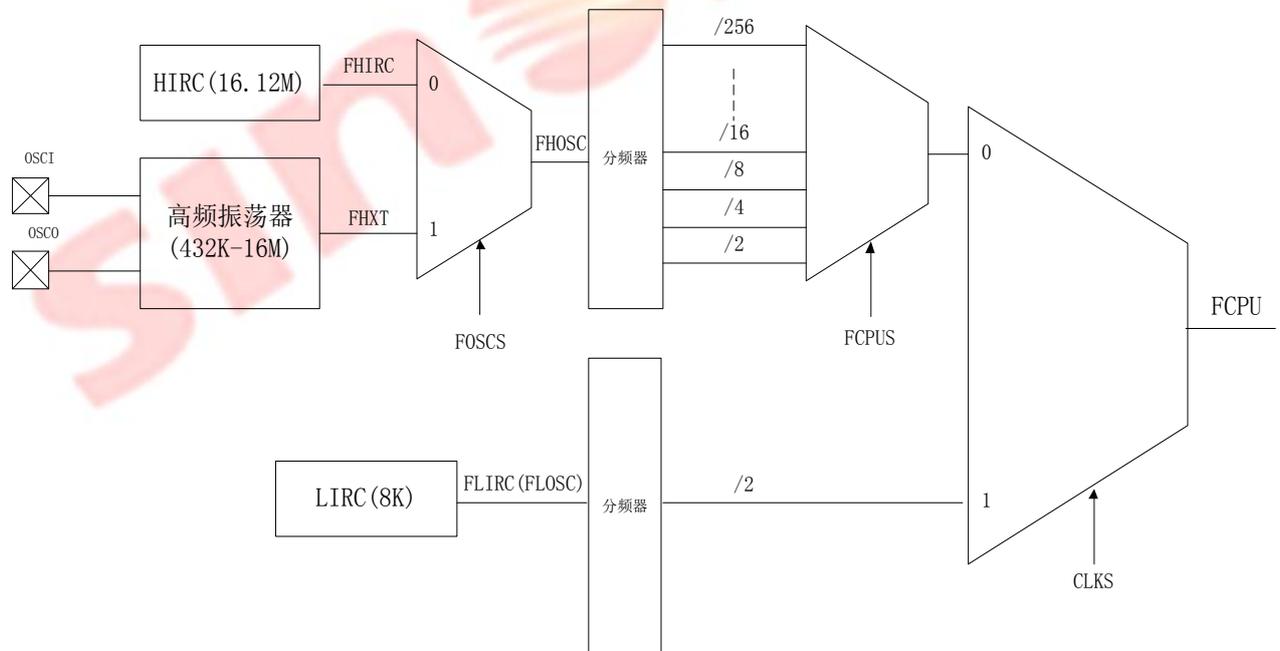
MC32P5213 支持高速工作模式、低速工作模式、休眠模式、HOLD 模式 1 和 HOLD 模式 2 共有 5 种工作模式。

| 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 (定时器可在高速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 (定时器可在低速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：当低速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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| OSCM | - | - | STBL | STBH | - | CLKS | LFEN | HFEN |
| R/W | - | - | R | R | - | R/W | R/W | R/W |
| 初始值 | - | - | 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

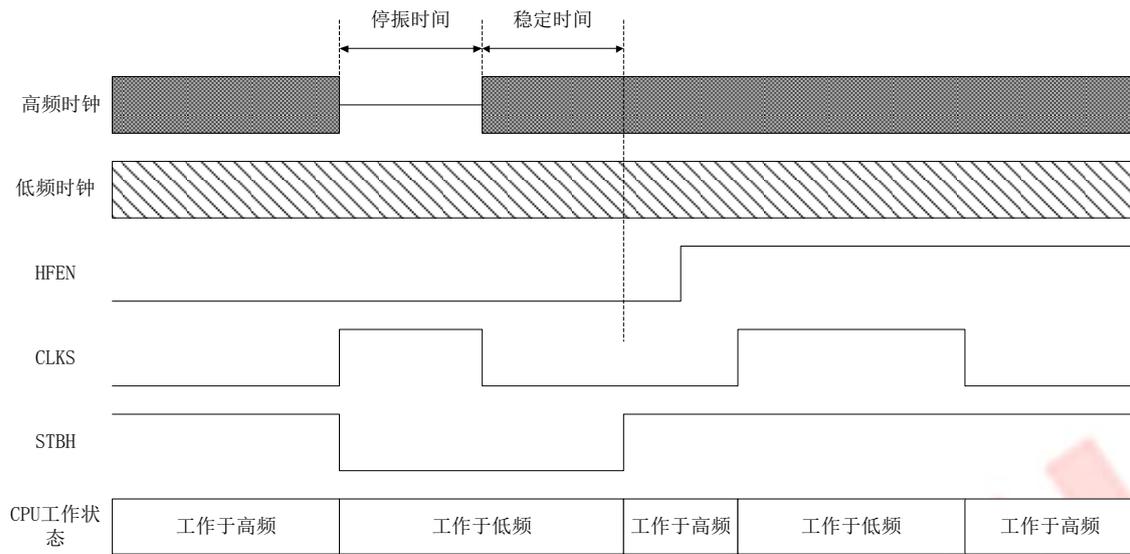
1: 低频振荡器始终工作

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

高低速时钟切换时序图



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 T0、定时器 T1 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 T0、定时器 T1 计数溢出中断请求发生
- ◇ 有 WDT 溢出
- ◇ 复位

MCU 退出低功耗模式后先经过 4 个低频周期和振荡等待（外部高频晶振等待 1024 个周期，内部高频等待 16 个周期，内部低频等待 16 个周期）然后开始工作。

4 复位

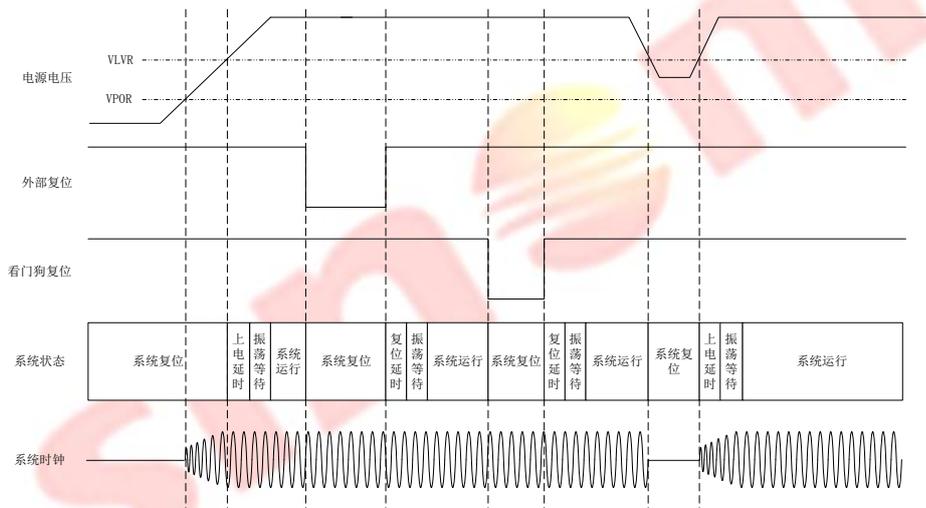
4.1 复位条件

MC32P5213 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 16 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 12ms）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 16 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

MC32P5213 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

MC32P5213 的 LVR 电压有八级（详见烧录配置选项），通过 OPBIT 的 VLVR5 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 0.02V 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+0.02V 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 I/O口

5.1 IO 工作模式

一组 8 位端口 P1 和一组 8 位端口 P2。P16 和外部中断 INTO 复用，P17 和 IR0UT 复用；P10 和外部中断 INT1 复用，P11 和 RAM 复用；INT0 和 INT1 分别作为定时器 T0 和定时器 T1 的外部时钟输入使用。

端口数据寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| IOP1 | P17D | P16D | P15D | P14D | P13D | P12D | P11D | P10D |
| R/W | R/W |
| 初始值 | 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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| IOP2 | P27D | P26D | P25D | P24D | P23D | P22D | P21D | P20D |
| R/W | R/W |
| 初始值 | X | X | X | X | X | X | X | X |

BIT[7:0] **P2nD** – P2 口数据位 (n=7-0)

端口方向寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| OEP1 | P17OE | P16OE | P15OE | P14OE | P13OE | P12OE | P11OE | P10OE |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] **P1nOE** – P1 口输出使能寄存器 (n=7-0)

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

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| OEP2 | P26OE | P26OE | P25OE | P24OE | P23OE | P22OE | P21OE | P20OE |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] **P2nOE** – P2 口输出使能寄存器 (n=7-0)

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

注 1: P11 配置为 RAM 电源管脚时相应控制位无效。

注 2: P11 作为 IO 输出时为开漏输出。

注 3: P17 低电平驱动电流可通过 DKW 寄存器的 ISEL 设置。

注 4: P17 在休眠和 HOLD 模式时必须设置为输入上拉状态, 否则会产生静态功耗。

注 5: P17 在 OPA 打开后自动转换为 OPA 模拟输入端, 相应控制位无效。

注 6: P27 作为 IO 输出时为开漏输出。配置为复位口时相应控制位无效。

注 7: OPA 开启时, P16 和 P10 的外部中断功能失效, INT0 和 INT1 的信号源将被 IRINS(IRIN 系统同步信号) 和 IRINT (IRIN 包络信号) 分别占用。

5.2 上拉电阻控制

P1 和 P2 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。

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| PUP1 | P17PU | P16PU | P15PU | P14PU | P13PU | P12PU | P11PU | P10PU |
| R/W | R/W |
| 初始值 | 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 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| PUP2 | P27PU | P26PU | P25PU | P24PU | P23PU | P22PU | P21PU | P20PU |
| R/W | R/W |
| 初始值 | 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

5.3 端口模式控制

P1 和 P2 口可以作为通用 IO 口, 也可以复用为 T 型键盘口, 通过设置使在休眠模式下可以键盘扫描。

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------|-------|--------|--------|--------|--------|--------|--------|--------|
| DKWP1 | - | P16DKW | P15DKW | P14DKW | P13DKW | P12DKW | P11DKW | P10DKW |
| R/W | - | R/W |
| 初始值 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[6:0] **P1nDKW** – P1 口模式选择 (n=6-0)

0: P1n 端口作为通用 IO 口

1: P1n 端口作为键盘口

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
| DKWP2 | P27DKW | P26DKW | P25DKW | P24DKW | P23DKW | P22DKW | P21DKW | P20DKW |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] **P2nDKW** – P2 口模式选择 (n=7-0)

0: P2n 端口作为通用 IO 口

1: P2n 端口作为键盘口

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。

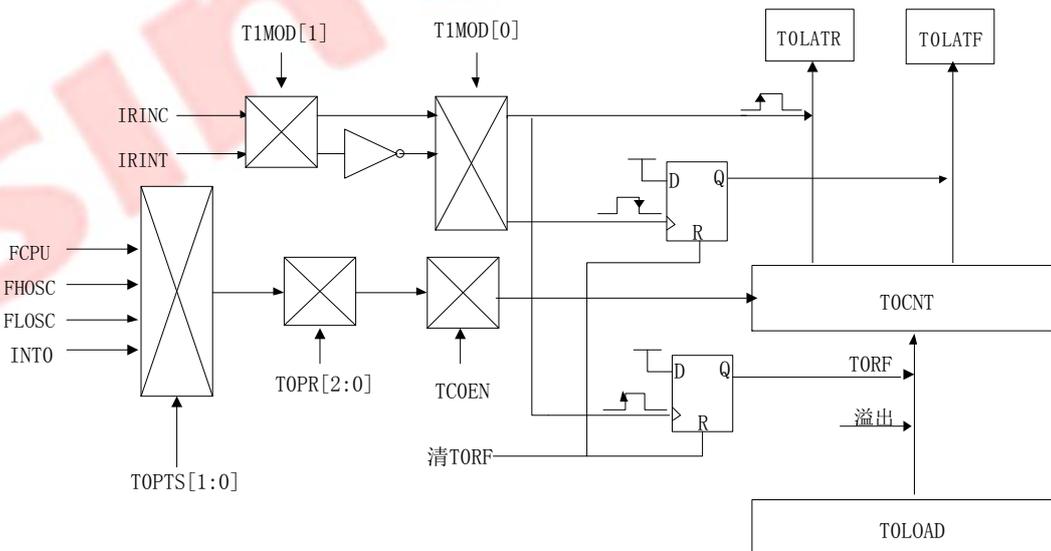
执行 CLRWDT 指令清 WDT。

WDT 溢出时间可通过 OPTION 配置设置为 16ms、64ms、256ms、1024ms、2048ms、4096ms、8192ms 和 16384ms。

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 2 个捕捉寄存器。

- 可通过预分频比设置计数频率
- 通过重载寄存器控制计数周期
- 通过捕捉寄存器获得上升/下降沿时间
- 可上升/下降沿触发数据重载
- 溢出中断功能
- 溢出唤醒功能



注 1: FHOSC 指系统高频时钟系统时钟; 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, 溢出中断可唤醒休眠模式; 若 HFEN=0 时, 定时器在低频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, 溢出中断可唤醒休眠模式; 若 LFEN=0 时, 定时器在休眠模式下将停止工作

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 |
|------------|-------|--------|--------|--------|--------|-------|-------|-------|
| TCR | TCOEN | T0MOD1 | T0MOD0 | TOPTS1 | TOPTS0 | TOPR2 | TOPR1 | TOPR0 |
| 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] TCOEN – T0 使能控制

- 0: 关闭 T0
- 1: 启动 T0

BIT[6:5] T0MOD[1:0] – TLATF/TLADR 触发模式选择

| T0MOD1 | T0MOD0 | 触发信号源 | 触发沿口 |
|--------|--------|-------|----------------------------|
| 0 | 0 | IRINC | TLATR 采样为上升沿, TLATF 时钟为下降沿 |
| 0 | 1 | IRINC | TLATR 采样为下降沿, TLATF 时钟为上升沿 |
| 1 | 0 | IRINT | TLATR 采样为上升沿, TLATF 时钟为下降沿 |
| 1 | 1 | IRINT | TLATR 采样为下降沿, TLATF 时钟为上升沿 |

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 | TCNT |
|-------|-------|-------|------|
| 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 |
|----------------|----------|----------|----------|----------|----------|----------|----------|----------|
| TOLATRH | TOLATRH7 | TOLATRH6 | TOLATRH5 | TOLATRH4 | TOLATRH3 | TOLATRH2 | TOLATRH1 | TOLATRH0 |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] **TOLATRH[7:0]** – TOCNT 的高 8 位值，这是一个只读寄存器，在有效沿口（TOMOD 选择）

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|----------------|----------|----------|----------|----------|----------|----------|----------|----------|
| TOLATRL | TOLATRL7 | TOLATRL6 | TOLATRL5 | TOLATRL4 | TOLATRL3 | TOLATRL2 | TOLATRL1 | TOLATRL0 |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] **TOLATRL[7:0]** – TOCNT 的低 8 位值，这是一个只读寄存器

注：在有效沿口（TOMOD 选择）将当前 TOCNT 的值存储到 TOLATR 中。

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|----------------|----------|----------|----------|----------|----------|----------|----------|----------|
| TOLATFH | TOLATFH7 | TOLATFH6 | TOLATFH5 | TOLATFH4 | TOLATFH3 | TOLATFH2 | TOLATFH1 | TOLATFH0 |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] **TOLATFH[7:0]** – TOCNT 的高 8 位值，这是一个只读寄存器。

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|----------------|----------|----------|----------|----------|----------|----------|----------|----------|
| TOLATFL | TOLATFL7 | TOLATFL6 | TOLATFL5 | TOLATFL4 | TOLATFL3 | TOLATFL2 | TOLATFL1 | TOLATFL0 |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7:0] **TOLATFL[7:0]** – TOCNT 的低 8 位值，这是一个只读寄存器。

注：当 TORF=0 时在有效沿口（TOMOD 选择）将当前 TOCNT 的值存储到 TOLATF 中，存储后 TORF 自动置 1 并停止存储，必须软件清除 TORF 才能再使能存储功能，清除 TORF 的同时将清零 TOLATF。

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------------|---------|---------|---------|---------|---------|---------|---------|---------|
| TLOADH | TLOADH7 | TLOADH6 | TLOADH5 | TLOADH4 | TLOADH3 | TLOADH2 | TLOADH1 | TLOADH0 |
| R/W | R/W |
| 初始值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

BIT[7:0] **TLOADH[7:0]** – TLOAD 的高 8 位值，这是一个读写寄存器，用于设置 T0 高 8 位重载值。

注：TLOADH 有缓冲寄存器，当写 TLOADL 寄存器后，TLOADH 寄存器将写入到 TLOADH 缓冲寄存器，当 T0 溢出时，TLOADH 缓冲寄存器的值自动装入 TOCNTH 中。

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------------|---------|---------|---------|---------|---------|---------|---------|---------|
| TLOADL | TLOADL7 | TLOADL6 | TLOADL5 | TLOADL4 | TLOADL3 | TLOADL2 | TLOADL1 | TLOADL0 |
| R/W | R/W |
| 初始值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

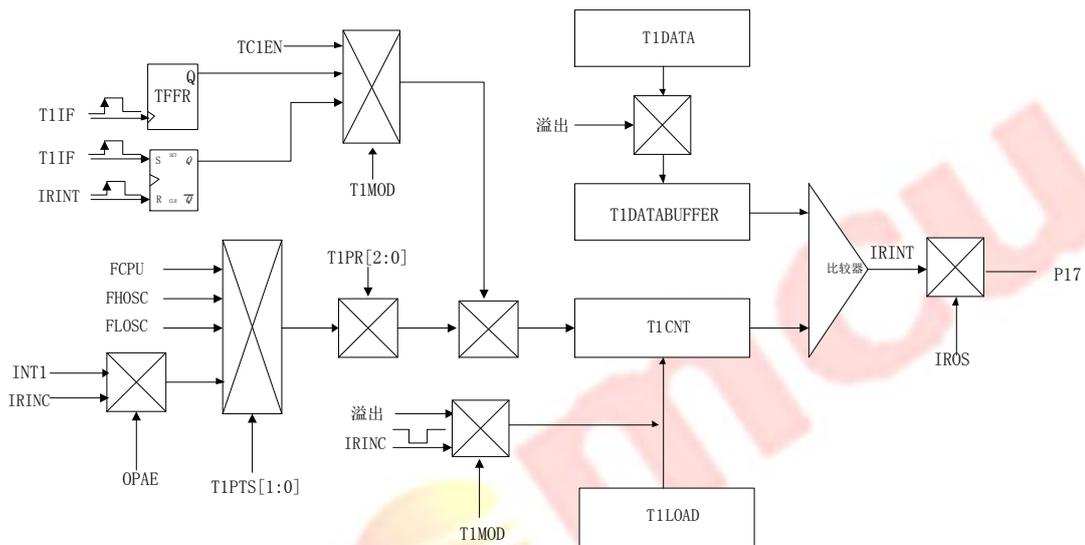
BIT[7:0] **TLOADL[7:0]** – TLO 低 8 位值，这是一个读写寄存器，用于设置 T0 低 8 位重载值。

注：当 TCOEN=0 时，写 TLOADL 将自动加载到 TOCNT；当 TCOEN=1 时，写 TLOADL 时不自动加载到 TOCNT，在 T0 溢出时自动加载到 TOCNT

6.3 定时器 T1

定时/计数器 T1 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计数频率
- 通过重载寄存器控制计数周期
- 通过比较寄存器设置 PWM 占空比
- 溢出中断功能
- 溢出唤醒功能



注 1: FHOSC 指系统高频时钟系统时钟; 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, 溢出中断可唤醒休眠模式; 若 HFEN=0 时, 定时器在低频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, 溢出中断可唤醒休眠模式; 若 LFEN=0 时, 定时器在休眠模式下将停止工作

T1PTS 可选择 T1 的时钟源, T1PR 可选择 T1 的预分频比, 所选中的时钟源通过预分频器后产生 T1CNT 的计数时钟。

当 T1CNT 递减到 0 时, 此时产生 T1 溢出中断请求标志 T1IF 置 1, 重载寄存器值自动置入 T1CNT, T1DATA 的值写入缓冲器 T1DATABUFFER 用于新的占空比波形生成。

通过 T1PR 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 T1CNT 计数到与 T1DATA 相等时, PWM 输出置 1; 当 T1CNT 计数溢出时, PWM 输出清 0, PWM 占空比的计算如下:

$$\begin{aligned} \text{PWM 高电平时间} &= (\text{T1DATA}) * \text{T1CNT 计数时钟周期} \\ \text{PWM 周期 (T1 的溢出周期)} &= (\text{T1LOAD}+1) * \text{T1CNT 的计数周期} \\ \text{PWM 占空比} &= (\text{T1DATA} / (\text{T1LOAD}+1)) \end{aligned}$$

T1 有 4 种工作模式，分别为定时/计数/PWM 模式，触发模式，包络模式和单次触发模式。

当设置为触发模式时，当 TC1EN=1 时 T1 由 T0 的溢出触发在开与关之间来回切换，当 TC1EN=0 时不触发，保存当前状态。

设置为包络模式时，当 IRINC 为低电平时 T1CNT 清零，并清零 IRINT，当 IRINC 为高电平时 T1CNT 重载并运行，当 T1CNT 溢出时置位 IRINT。

当设置为单次触发模式时，当 TC1EN=1 时 T1 由 T0 的溢出触发开启，当 T1 溢出时自动关闭，当 TC1EN=0 时不触发，保存当前状态。

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 |
|-------------|-------|--------|--------|--------|--------|-------|-------|-------|
| T1CR | TC1EN | T1MOD1 | T1MOD0 | T1PTS1 | T1PTS0 | T1PR2 | T1PR1 | T1PR0 |
| 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] TC1EN – T1 使能控制

0: 关闭 T1

1: 启动 T1

BIT[6:5] T1MOD[1:0] – 模式选择

| T1MOD1 | T1MOD0 | 模式 | 描述 |
|--------|--------|--------------|-------------------|
| 0 | 0 | 定时/计数/PWM 模式 | 开关由 TC1EN 决定 |
| 0 | 1 | 触发模式 | 由 T0IF 触发交替开关 |
| 1 | 0 | 包络模式 | IRINC 低电平清零 T1CNT |
| 1 | 1 | 单次触发模式 | 由 T0IF 触发开，溢出触发关 |

BIT[4:3] T1PTS[1:0] – T1 时钟源选择

| T1PTS[1:0] | T1 时钟源 (OPAE=1) | T1 时钟源 (OPAE=0) |
|------------|-------------------|-------------------|
| 00 | FCPU | FCPU |
| 01 | FHOSC | FHOSC |
| 10 | FLOSC | FLOSC |
| 11 | IRINS | 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 |
|---------------|---------|---------|---------|---------|---------|---------|---------|---------|
| T1LOAD | T1LOAD7 | T1LOAD6 | T1LOAD5 | T1LOAD4 | T1LOAD3 | T1LOAD2 | T1LOAD1 | T1LOAD0 |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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 |
|---------------|---------|---------|---------|---------|---------|---------|---------|---------|
| T1DATA | T1DATA7 | T1DATA6 | T1DATA5 | T1DATA4 | T1DATA3 | T1DATA2 | T1DATA1 | T1DATA0 |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

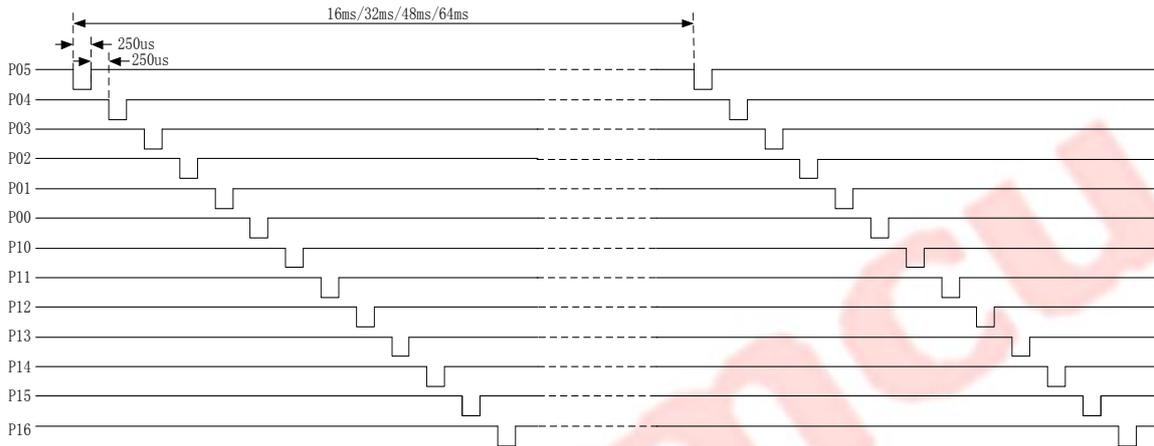
BIT[7:0] **T1DATA[7:0]** – 这是一个读写寄存器，用于设置 PWM1 高电平时间。

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

7 键盘扫描模块

7.1 键盘扫描介绍

MC32P5213 的 P16-P10 和 P27-P20 可设置为键盘扫描模式，当系统进入休眠/HOLD 模式时，硬件自动按照下图设置相应端口为输出低电平或者输入上拉状态。键盘扫描时钟使用低频 RC 振荡器，扫描周期由 WSEL 设置，可选择 16ms、32ms、48ms 和 64ms 四种，未设置为键盘扫描模式的端口不会输出波形但不会影响下图的时序。



7.2 键盘扫描操作步骤

- ◇ S1: 设置 DKWP1 和 DKWP2 将相应的端口设置为键盘口
- ◇ S2: 设置 PUP1 和 PUP2 开启相应的端口的上拉电阻
- ◇ S3: 设置 DKWE 位使能键盘扫描功能
- ◇ S4: 关闭总中断 GIE 位
- ◇ S5: 执行 STOP 指令
- ◇ S6: 等待端口唤醒后，系统将执行 STOP 后指令
- ◇ S7: 设置 DKWP1 和 DKWP2 将相应的端口设置为通用 IO 口

注：上述操作步骤中 S3 如果不使能 DKWE 位，芯片将只能由端口(设置为键盘口)的下降沿触发唤醒，此方法可使芯片支持矩形键盘。

7.3 键盘扫描相关寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| DKW0 | DKWE | IROS | IROT | WSEL1 | WSEL0 | RSEL | ISEL1 | ISEL0 |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7] DKWE– 键盘唤醒选择
 0: 矩形键盘唤醒
 1: T 型键盘唤醒

注: T 型键盘同时设置 OSCM 的 LFEN 位为 1; 矩形键盘同时设置 OSCM 的 LFEN 位为 0。

BIT[6] IROS– IROUT 输出选择
 0: P17
 1: PWM

BIT[5] IROT– IRIN 输出选择
 0: P14
 1: IRIN

BIT[4:3] WSEL[1:0]– 扫描周期选择
 00: 16ms
 01: 32ms
 10: 48ms
 11: 64ms

BIT[2] RSEL– 上拉电阻选择
 0: 上拉电阻 40K
 1: 上拉电阻 100K

BIT[1:0] ISEL[1:0]– IROUT 驱动电流选择位
 00: 125mA
 01: 250mA
 10: 375mA
 11: 500mA

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|--------|--------|--------|--------|
| DKW1 | - | - | IRES1 | IRES0 | LSEL11 | LSEL10 | LSEL01 | LSEL00 |
| R/W | - | - | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | - | - | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[5] **IRES1** – P23 口 I2C 上拉电阻选择

0: 上拉电阻为 10K

1: 上拉电阻由 RSEL 选择

BIT[4] **IRES0** – P20 口 I2C 上拉电阻选择

0: 上拉电阻为 10K

1: 上拉电阻由 RSEL 选择

BIT[3:2] **LSEL1[1:0]** – P22 口低电平驱动电流选择

00: 2mA

01: 4mA

10: 8mA

11: 12mA

BIT[1:0] **LSEL0[1:0]** – P21 口低电平驱动电流选择

00: 2mA

01: 4mA

10: 8mA

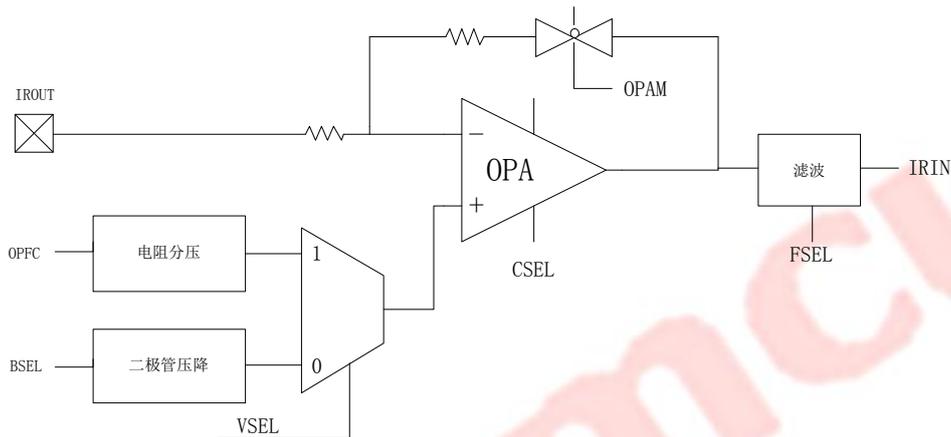
11: 12mA

8 OPA

8.1 OPA 模块介绍

MC32P5213 的 OPA 模块可实现 IROUT 管脚输入小信号的放大，从而实现红外信号的学习功能。

- 可设置灵敏度
- 可设置不同滤波能力
- 可设置电压基准的模式（二极管模式和电阻模式）



8.2 OPA 操作步骤

- ◇ S1: 设置 FSEL 和 CSEL
- ◇ S2: 设置 VSEL 位选择不同电压基准模式
- ◇ S3: 置位 OPAE 使能 OPA
- ◇ S4: 置位 OPAM 进入 OPA 校准模式
- ◇ S5: 根据不同的 VSEL，设置调整 OPFC 或 BSEL 从最大值以步长为 1 递减直到 IRIN 为 1，此时 OPFC 或 BSEL 的值即为校准值；每步读取 IRIN 需等待 100us 值，判断次数不少于 3 次
- ◇ S6: 清零 OPAM 进入正常工作模式

注：OPA 打开后，IROUT 管脚自动转换为 OPA 输入端，要保证其他端口始终处于输入上拉状态，避免干扰 OPA 工作。

8.3 OPA 相关寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| OPCR0 | OPAE | OPAM | OPFC5 | OPFC4 | OPFC3 | OPFC2 | OPFC1 | OPFC0 |
| R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7] OPAE – OPA 使能控制

0: 关闭 OPA

1: 启动 OPA

BIT[6] OPAM– OPA 模式控制

0: OPA 工作模式

1: OPA 校准模式

BIT[5:0] OPFC[5:0]电压基准校准位

| VSEL | 位置 | 描述 |
|------|-----------------|--------------------|
| 0 | BIT[5:3] | 无效 |
| | BIT[2:0] | 二极管压降调节位 BSEL[2:0] |
| 1 | BIT[5:0] | 电阻分压调节位 OPFC[5:0] |

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| OPCR1 | IRIN | VSEL | – | – | FSEL1 | FSEL0 | CSEL1 | CSEL0 |
| R/W | R | R/W |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BIT[7] IRIN– OPA 输出位

0: OPA 输出低

1: OPA 输出高

BIT[6] VSEL– 电压基准模式选择

0: 二极管模式

1: 电阻分压模式

BIT[4:3] FSEL[1:0] – 滤波能力选择

00: 1.5us (最弱) 推荐

01: 3us

10: 4.5us

11: 6us (最强)

BIT[4:3] CSEL[1:0] – 灵敏度选择

00: 5mV (最高) 推荐

01: 36mV

10: 65mV

11: 135mV (最低)

9 RAM 电源

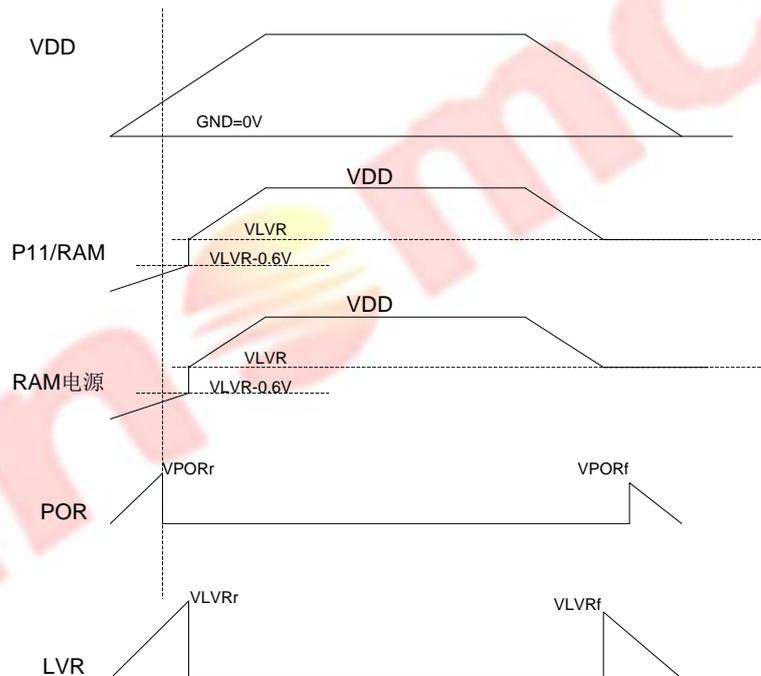
9.1 RAM 电源外置

设置烧录配置选项的 RAMS 位为 0，RAM 电源为 P11/RAM 引脚。

当芯片 VDD 高于 LVR 电压，P11/RAM 引脚与芯片 VDD 相连接，当芯片 VDD 低于 LVR 电压，P11/RAM 引脚与芯片 VDD 完全断开。

当 P11/RAM 引脚外接电容，VDD 电压降至 LVR 电压之后，RAM 模块电源与 VDD 完全断开，RAM 模块供电仅靠 P11/RAM 引脚外接电容供电，P11/RAM 引脚电压维持 VLVR 电压。

芯片进入 STOP 之后，RAM 进入静态保持，RAM 模块仅为半导体漏电，此时功耗极低(小于 50nA)。如果 P11/RAM 引脚外接 47uF 电容，LVR 电压选择 1.5V，芯片 VDD 为 1.8V，此时将 VDD 电压瞬间降为 0V，而 P11/RAM 引脚因为外接 47uF 电容，仍会保持 1.8V。RAM 电源供电也为 1.8V，由于 RAM 模块功耗极低，至少经过 24 小时，P11/RAM 引脚上电容可降至 RAM 保持最低电压 0.9V。在这种情况下 RAM 数据保持时间大于 24 小时。



RAMS=0时，RAM电源为P11/RAM引脚

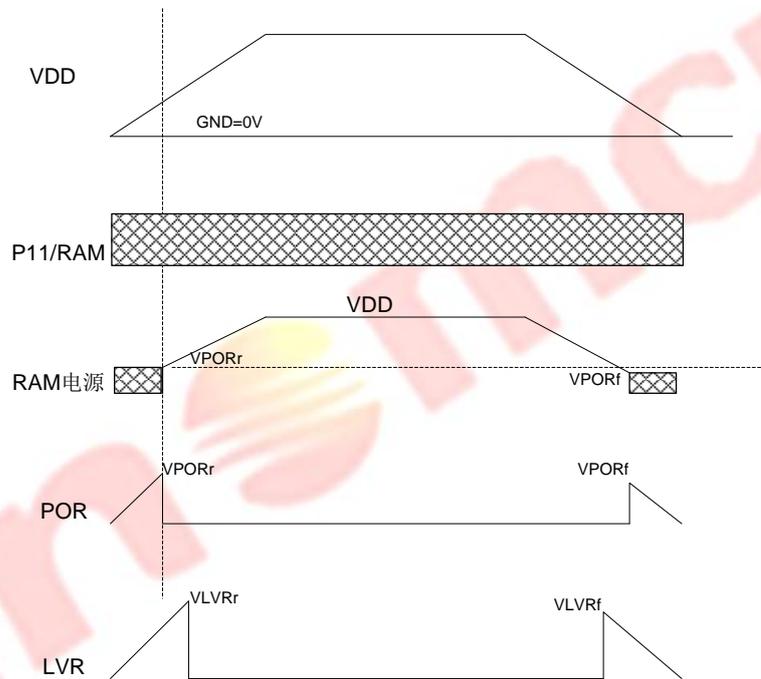
9.2 RAM 电源内置

设置烧录配置选项的 RAMS 位为 1，RAM 电源在内部与芯片 VDD 相连接。

当芯片 VDD 高于 POR 电压，RAM 电源与芯片 VDD 相连接，当芯片 VDD 低于 POR 电压，RAM 电源与芯片 P11/RAM 引脚相连接。

当 RAM 电源选择芯片内部 VDD 时，芯片电压降至 POR 电压之后，内部 RAM 模块电源与 VDD 引脚完全断开，RAM 电源同 P11/RAM 引脚相连接，RAM 无保持功能。要实现 RAM 保持，则必须保证 VDD 电压高于 POR 电压。

芯片进入 STOP 之后，整个芯片功耗有所降低，如果 VDD 外接 47uF 电容，芯片 VDD 为 1.8V，此时将 VDD 瞬间降为 0V，而 VDD 引脚因为外接 47uF 电容，仍会保持 1.8V，此时整个芯片功耗最大值为 1uA，经过 2 分钟后，VDD 引脚上电容可降至 RAM 保持最低电压 1.5V。在这种情况下 RAM 数据保持时间大于 2 分钟。



RAMS=1时，RAM电源为芯片VDD，P11/RAM作为IO

10 中断

MC32P5213 的中断有外中断 (INT0,INT1)、定时器中断 (T0,T1)。外部中断、定时器中断可被 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。

MC32P5213 的中断向量地址是 0008H。

10.1 外中断

MC32P5213 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。

注：OPA 开启时，INT0 和 INT1 的信号源将被 IRINS (IRIN 同步信号) 和 IRINT (IRIN 包络信号) 分别占用，外部端口信号将不起作用。

10.2 定时器中断

定时器 T0、T1 在计数溢出时会置位中断标志 T0IF、T1IF，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。

10.3 中断相关寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------------|-------|-------|-------|-------|--------|--------|--------|--------|
| MCR | GIE | - | TO | PD | MINT11 | MINT10 | MINT01 | MINT00 |
| R/W | R/W | - | R | R | R/W | R/W | R/W | R/W |
| 初始值 | 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有相应的控制位决定
- 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** – 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

中断使能寄存器

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|--------|--------|-------|-------|
| INTE | - | - | - | - | INT1IE | INT0IE | T1IE | T0IE |
| R/W | - | - | - | - | R/W | R/W | R/W | R/W |
| 初始值 | - | - | - | - | 0 | 0 | 0 | 0 |

- 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 T1 使能
0: 屏蔽定时器 T1 中断
1: 使能定时器 T1 中断
- BIT[0] T0IE** – 定时器 T0 使能
0: 屏蔽定时器 T0 中断
1: 使能定时器 T0 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 | - | TORF | - | - | INT1IF | INT0IF | T1IF | T0IF |
| R/W | - | R/W | - | - | R/W | R/W | R/W | R/W |
| 初始值 | - | 0 | - | - | 0 | 0 | 0 | 0 |

- BIT[6] TORF**– 定时器 T0 重载标志
 0: 未发生定时器 T0 重载
 1: 发生定时器 T0 重载, 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 T1 标志
 0: 未发生屏蔽定时器 T1 中断
 1: 发生定时器 T1 中断, 需软件清零
- BIT[0] T0IF**– 定时器 T0 标志
 0: 未发生屏蔽定时器 T0 中断
 1: 发生定时器 T0 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 | -0.3~4.0 | V |
| 输入电压 | VIN | VSS-0.3~VDD+0.3 | V |
| 工作温度 | TA | -20~70 | °C |
| 储存温度 | Tstg | -65~150 | °C |
| 焊接温度及时间 | Tsld | 260°C持续 10 秒 | |
| 流过 VDD 最大电流 | IVDDmax | 50 | mA |
| 流过 GND 最大电流 1 (除 P17 外) | IGNDmax1 | 50 | mA |
| 流过 GND 最大电流 2 (P17 引脚) | IGNDmax2 | 持续驱动 200 | mA |
| 流过 GND 最大电流 3 (P17 引脚) | IGNDmax3 | 间歇性驱动 600 (如: 1 秒钟有 50ms 驱动) | mA |

11.2 直流特性参数

所有参数均为 T=25°C, VDD=3V, 除非特殊说明

| 特性 | 符号 | 引脚 | 条件 | 最小 | 典型 | 最大 | 单位 |
|----------------------|-----|-----|----------------------------------|-----|----|-----|----|
| 工作电压 | VDD | VDD | FCPU=8.06MHz@内部高频 16.12MHz | 2.7 | 3 | 3.6 | V |
| | | | FCPU=4.03MHz@内部高频 16.12MHz | 2 | 3 | 3.6 | V |
| | | | FCPU=2.015MHz@内部高频 16.12MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=1.0075MHz@内部高频 16.12MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=503.75KHz@内部高频 16.12MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=251.875KHz@内部高频 16.12MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=125.94KHz@内部高频 16.12MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=62.97KHz@内部高频 16.12MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=FILRC/2@内部低频 32KHz | 1.6 | 3 | 3.6 | V |
| | | | FCPU=8MHz@外部晶体 16MHz | 2.7 | 3 | 3.6 | V |
| FCPU=4MHz@外部晶体 16MHz | 2 | 3 | 3.6 | V | | | |

| | | | | | | | |
|------------|-------|---------|--|-----|---|-----|-----|
| | | | FCPU=2MHz@外部晶体 16MHz | 2 | 3 | 3.6 | V |
| | | | FCPU=1MHz@外部晶体 16MHz | 2 | 3 | 3.6 | V |
| | | | FCPU=500KHz@外部晶体 16MHz | 2 | 3 | 3.6 | V |
| | | | FCPU=250KHz@外部晶体 16MHz | 2 | 3 | 3.6 | V |
| | | | FCPU=125KHz@外部晶体 16MHz | 2 | 3 | 3.6 | V |
| | | | FCPU=62.5KHz@外部晶体 16MHz | 2 | 3 | 3.6 | V |
| | | | FCPU=4MHz@外部晶体 8MHz | 2 | 3 | 3.6 | V |
| | | | FCPU=2MHz@外部晶体 8MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=1MHz@外部晶体 8MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=500KHz@外部晶体 8MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=250KHz@外部晶体 8MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=125KHz@外部晶体 8MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=62.5KHz@外部晶体 8MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=2MHz@外部晶体 4MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=1MHz@外部晶体 4MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=500KHz@外部晶体 4MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=250KHz@外部晶体 4MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=125KHz@外部晶体 4MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=62.5KHz@外部晶体 4MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=500KHz@外部晶体 1MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=250KHz@外部晶体 1MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=125KHz@外部晶体 1MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=62.5KHz@外部晶体 1MHz | 1.8 | 3 | 3.6 | V |
| | | | FCPU=227.5KHz@外部晶体 455KHz | 1.8 | 3 | 3.6 | V |
| RAM 保持电压 1 | VRAM1 | RAM/P11 | RAMS=0 (RAM 电源外接), VDD=0V | 0.9 | | | V |
| RAM 保持时间 1 | TRAM1 | | RAMS=0 (RAM 外接 47uF 电容), VDD=1.8V 瞬间降至 0V 后 | 24 | | | HRs |
| RAM 保持电压 2 | VRAM2 | VDD | RAMS=1 (RAM 电源内置) | 1.5 | | | V |

| | | | | | | | |
|-----------------|-------|--------------------------------|--|--------|-----|--------|------|
| RAM 保持时间 2 | TRAM2 | | RAMS=1 (RAM 电源为 VDD) VDD 外接 47uF 电容, VDD=1.8V , 拿掉电池 | 2 | | | MINs |
| 输入漏电 | VLEAK | 所有输入引脚 | | -1 | | 1 | uA |
| 输入高电平 | VIH | 所有输入引脚 | | 0.8VDD | | | V |
| 输入低电平 | VIL | 所有输入引脚 | | | | 0.2VDD | V |
| 上拉电阻 1 | RPU1 | P1 P2 (除 P23/P20 /P17) | Vin=0.9V, RSEL=0 | 50 | 100 | 200 | Kohm |
| | | | Vin=0.9V, RSEL=1 | 20 | 40 | 80 | |
| 上拉电阻 2 | RPU2 | P23 | Vin=0.9V, IRES1=1, RSEL=0 | 50 | 100 | 200 | Kohm |
| | | | Vin=0.9V, IRES1=1, RSEL=1 | 20 | 40 | 80 | |
| | | | Vin=0.9V, IRES1=0 | 6 | 10 | 14 | |
| 上拉电阻 3 | RPU3 | P20 | Vin=0.9V, IRES0=1, RSEL=0 | 50 | 100 | 200 | Kohm |
| | | | Vin=0.9V, IRES0=1, RSEL=1 | 20 | 40 | 80 | |
| | | | Vin=0.9V, IRES0=0 | 6 | 10 | 14 | |
| 上拉电阻 4 | RPU4 | P17 | Vin=0.9V | 6 | 10 | 14 | Kohm |
| 输出高电平 驱动电流 1 | IOH1 | P1 P2 (除 P11/P27) | Voh=2.4V | -4 | -8 | -16 | mA |
| 输出低电平 驱动电流 1 | IOL1 | P0 P1 (除 P22/P21/P17) | Vol=0.6V | 4 | 8 | 16 | mA |
| 输出低电平 驱动电流 2 | IOL2 | P17 | Vol=0.6V, ISEL=00 | 75 | 125 | 175 | mA |
| | | | Vol=0.6V, ISEL=01 | 150 | 250 | 350 | |
| | | | Vol=0.6V, ISEL=10 | 225 | 375 | 525 | |
| | | | Vol=0.6V, ISEL=11 | 300 | 500 | 700 | |
| 输出低电平 驱动能力 2 | IOL3 | P22 | Vol=0.6V,LSEL1=00 | 1 | 2 | 3 | mA |
| | | | Vol=0.6V,LSEL1=01 | 2 | 4 | 6 | |
| | | | Vol=0.6V,LSEL1=10 | 4 | 8 | 12 | |
| | | | Vol=0.6V,LSEL1=11 | 6 | 12 | 18 | |
| 输出低电平 驱动能力 3 | IOL4 | P21 | Vol=0.6V,LSEL0=00 | 1 | 2 | 3 | mA |
| | | | Vol=0.6V,LSEL0=01 | 2 | 4 | 6 | |
| | | | Vol=0.6V,LSEL0=10 | 4 | 8 | 12 | |
| | | | Vol=0.6V,LSEL0=11 | 6 | 12 | 18 | |
| 动态功耗 | IDDC | VDD | FCPU=8.06MHz@内部高频 16.12MHz | | 5 | | mA |
| | | | FCPU=4.03MHz@内部高频 16.12MHz | | 2.5 | | mA |
| | | | FCPU=2.015MHz@内部高频 16.12MHz | | 1.6 | | mA |
| | | | FCPU=1.0075MHz@内部高频 16.12MHz | | 1.2 | | mA |

| | | | | | | |
|--------------|--------|-----|----------------------------------|------|--|----|
| | | | FCPU=503.75KHz@内部高频 16.12MHz | 0.9 | | mA |
| | | | FCPU=251.875KHz@内部高频 16.12MHz | 0.8 | | mA |
| | | | FCPU=125.94KHz@内部高频 16.12MHz | 0.75 | | mA |
| | | | FCPU=62.97KHz@内部高频 16.12MHz | 0.7 | | mA |
| | | | FCPU=FILRC/2@内部低频 | 0.2 | | mA |
| | | | FCPU=8MHz@外部晶体 16MHz | 5 | | mA |
| | | | FCPU=4MHz@外部晶体 16MHz | 2.5 | | mA |
| | | | FCPU=2MHz@外部晶体 16MHz | 1.6 | | mA |
| | | | FCPU=1MHz@外部晶体 16MHz | 1.1 | | mA |
| | | | FCPU=500KHz@外部晶体 16MHz | 0.75 | | mA |
| | | | FCPU=250KHz@外部晶体 16MHz | 0.65 | | mA |
| | | | FCPU=125KHz@外部晶体 16MHz | 0.55 | | mA |
| | | | FCPU=62.5KHz@外部晶体 16MHz | 0.5 | | mA |
| | | | FCPU=4MHz@外部晶体 8MHz | 2.6 | | mA |
| | | | FCPU=2MHz@外部晶体 8MHz | 1.4 | | mA |
| | | | FCPU=1MHz@外部晶体 8MHz | 0.95 | | mA |
| | | | FCPU=500KHz@外部晶体 8MHz | 0.7 | | mA |
| | | | FCPU=250KHz@外部晶体 8MHz | 0.55 | | mA |
| | | | FCPU=125KHz@外部晶体 8MHz | 0.5 | | mA |
| | | | FCPU=62.5KHz@外部晶体 8MHz | 0.45 | | mA |
| | | | FCPU=2MHz@外部晶体 4MHz | 1.5 | | mA |
| | | | FCPU=1MHz@外部晶体 4MHz | 0.9 | | mA |
| | | | FCPU=500KHz@外部晶体 4MHz | 0.65 | | mA |
| | | | FCPU=250KHz@外部晶体 4MHz | 0.5 | | mA |
| | | | FCPU=125KHz@外部晶体 4MHz | 0.45 | | mA |
| | | | FCPU=62.5KHz@外部晶体 4MHz | 0.4 | | mA |
| | | | FCPU=500KHz@外部晶体 1MHz | 0.55 | | mA |
| | | | FCPU=250KHz@外部晶体 1MHz | 0.45 | | mA |
| | | | FCPU=125KHz@外部晶体 1MHz | 0.4 | | mA |
| | | | FCPU=62.5KHz@外部晶体 1MHz | 0.35 | | mA |
| | | | FCPU=227.5KHz@外部晶体 455KHz | 0.45 | | mA |
| HOLD 模式 1 功耗 | IHOLD1 | VDD | 高频振荡开启, 执行 STOP 指令 | | | |
| | | | 内部高频 16.12MHz | 500 | | uA |
| | | | 外部晶体 16MHz | 350 | | uA |
| | | | 外部晶体 8MHz | 250 | | uA |
| | | | 外部晶体 4MHz | 200 | | uA |
| | | | 外部晶体 1MHz | 150 | | uA |

| | | | | | | | |
|----------------------|--------|------------------------|--------------------------------------|-----|-----|---|----|
| | | | 外部晶体 455KHz | | 120 | | uA |
| HOLD 模式 2 功耗 | IHOLD2 | VDD | 高频振荡关闭, 低频振荡开启, 执行 STOP 指令 | | 0.5 | 3 | uA |
| 休眠模式功耗 | ISTOP | VDD | 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 |
| 低电压复位电压 | VLVR | VDD | VLVRS=111 | | 1.5 | | V |
| | | | VLVRS=110 | | 1.7 | | V |
| | | | VLVRS=101 | | 1.9 | | V |
| | | | VLVRS=100 | | 2.1 | | V |
| | | | VLVRS=011 | | 2.3 | | V |
| | | | VLVRS=010 | | 2.5 | | V |
| | | | VLVRS=001 | | 2.7 | | V |
| | | | VLVRS=000 | | 2.9 | | V |
| 系统时钟 FCPU 与低电压复位电压匹配 | | OPTION 配置时选择相应的 LVR 电压 | FCPU=8MHz | 2.5 | | | V |
| | | | FCPU=4MHz | 1.7 | | | V |
| | | | FCPU 低于 2MHz | 1.5 | | | V |

11.3 交流电气参数

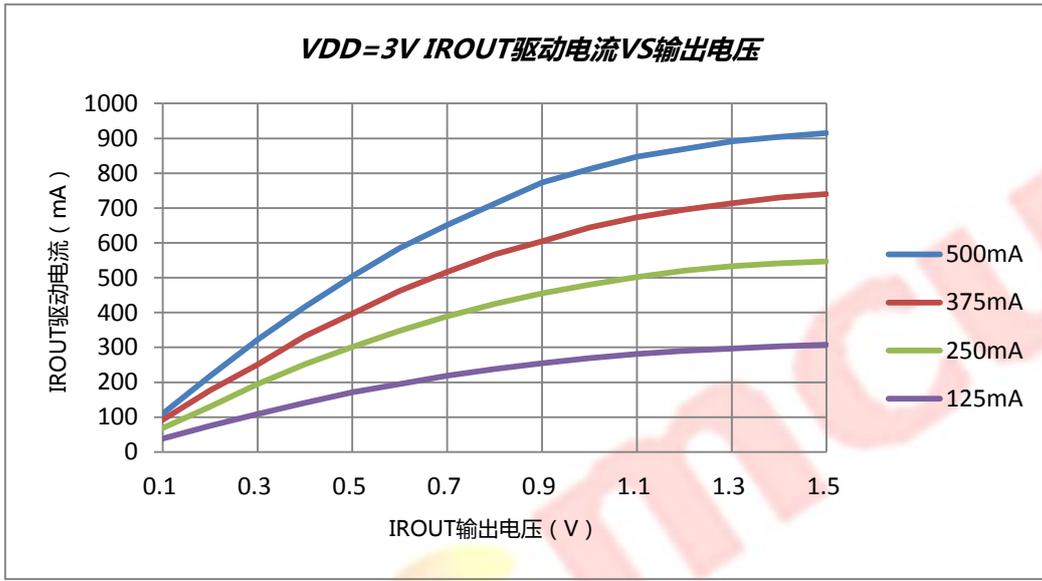
VDD=3V, T=25°C

| 特性 | 符号 | 条件 | 最小 | 典型 | 最大 | 单位 |
|--------------|--------|--------------------------------|------|-------|------|-----|
| 内部高频 RC 振荡频率 | FHIRC1 | T=25°C VDD=1.8~3.6V | -1% | 16.12 | +1% | MHz |
| | FHIRC2 | T=-20°C ~ 70°C VDD=1.8~3.6V | -2% | 16.12 | +2% | MHz |
| 内部低频 RC 振荡频率 | FLIRC | T=25°C VDD=3V | -20% | 8 | +20% | KHz |

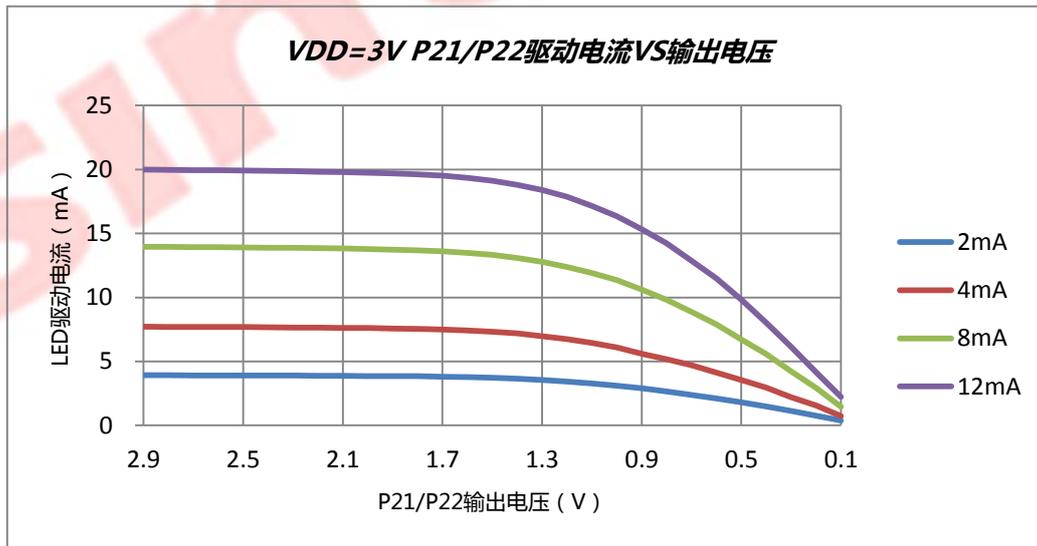
12 特性曲线图

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。

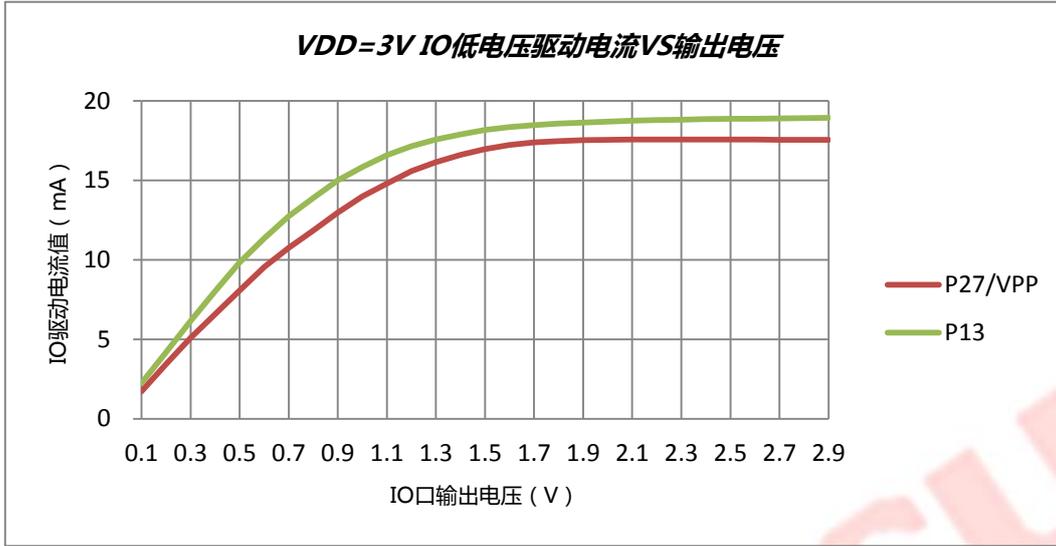
12.1 IROUT 驱动电流 VS 输出电压



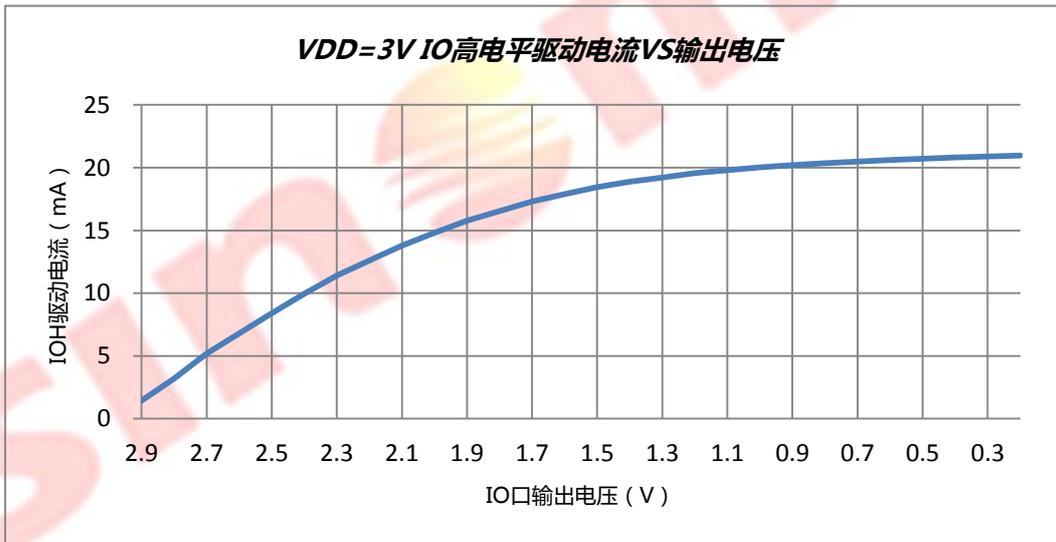
12.2 LED 驱动电流 VS 输出电压



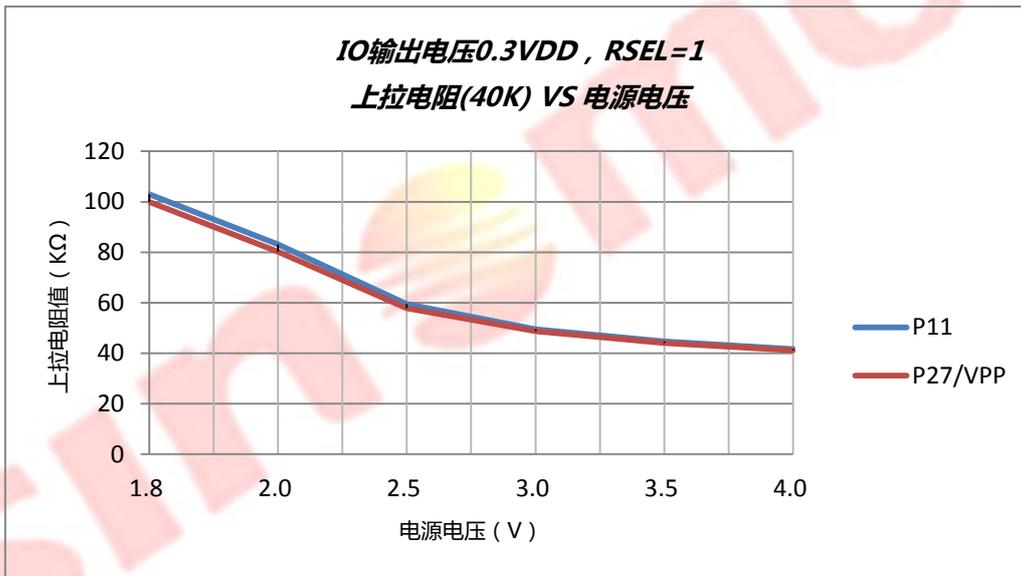
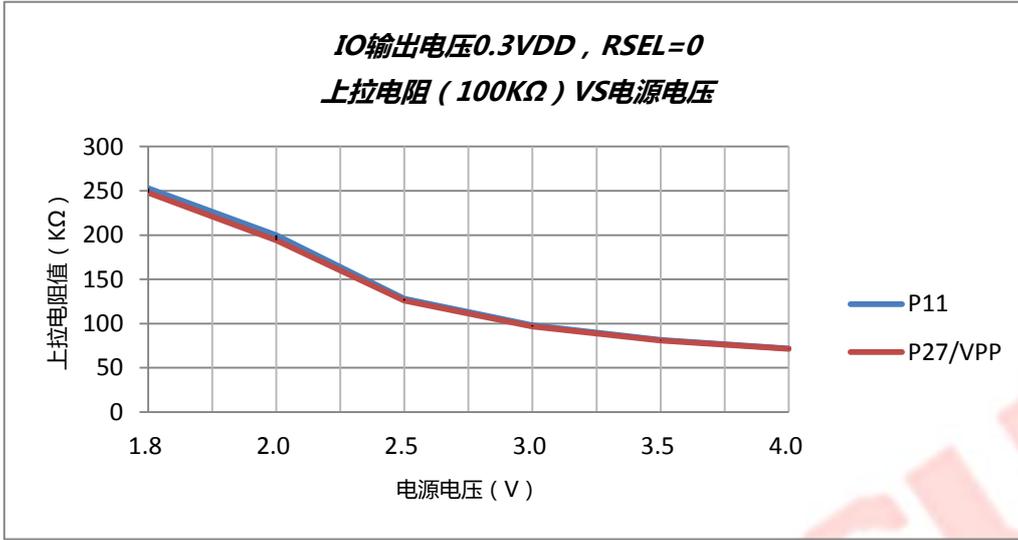
12.3 IO 输出低电平驱动电流 VS 输出电压

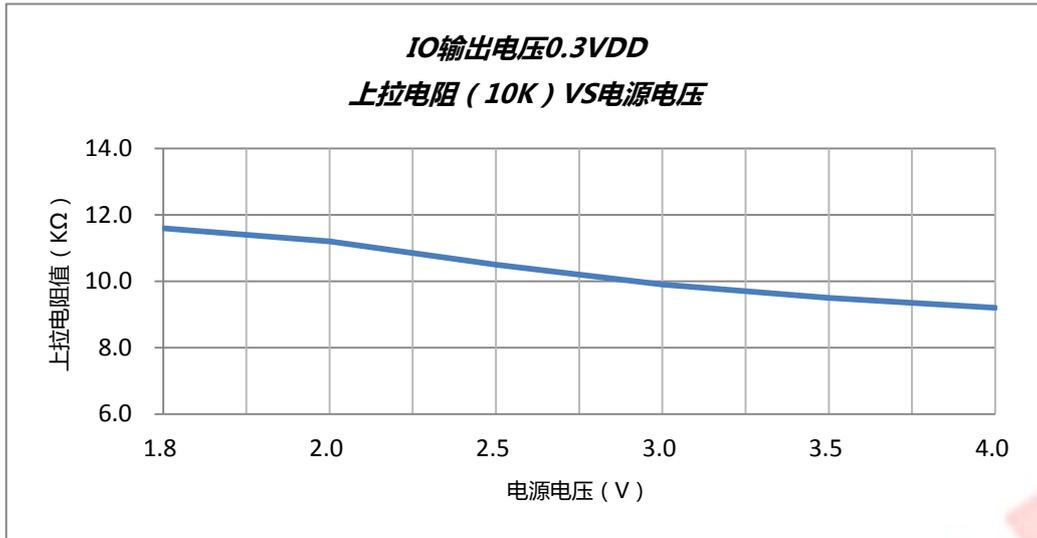


12.4 IO 输出高电平驱动电流 VS 输出电压



12.5 IO 口上拉电阻 VS 电源电压

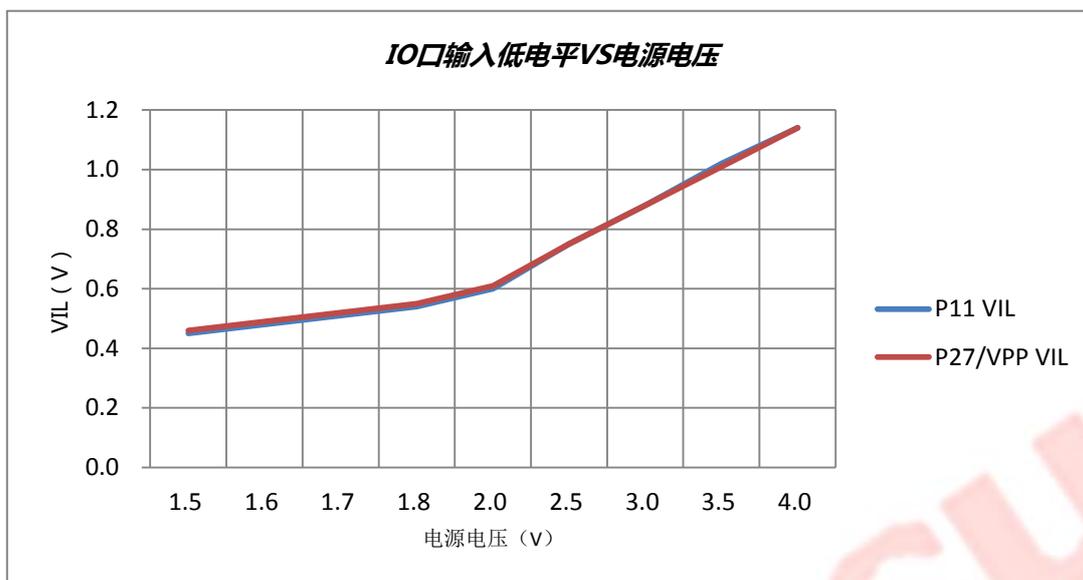




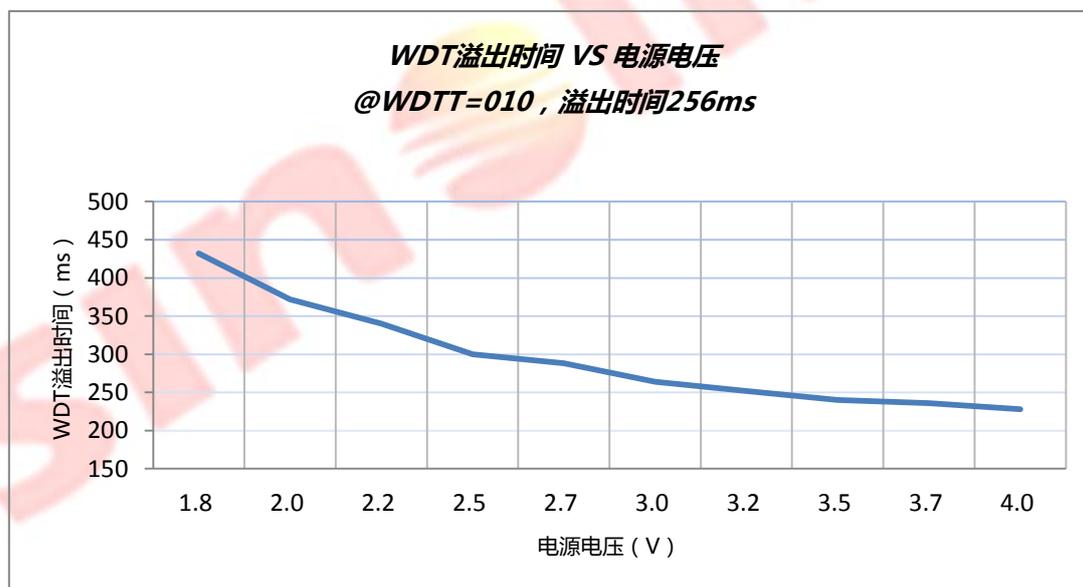
12.6 IO 口输入高电平 VS 电源电压



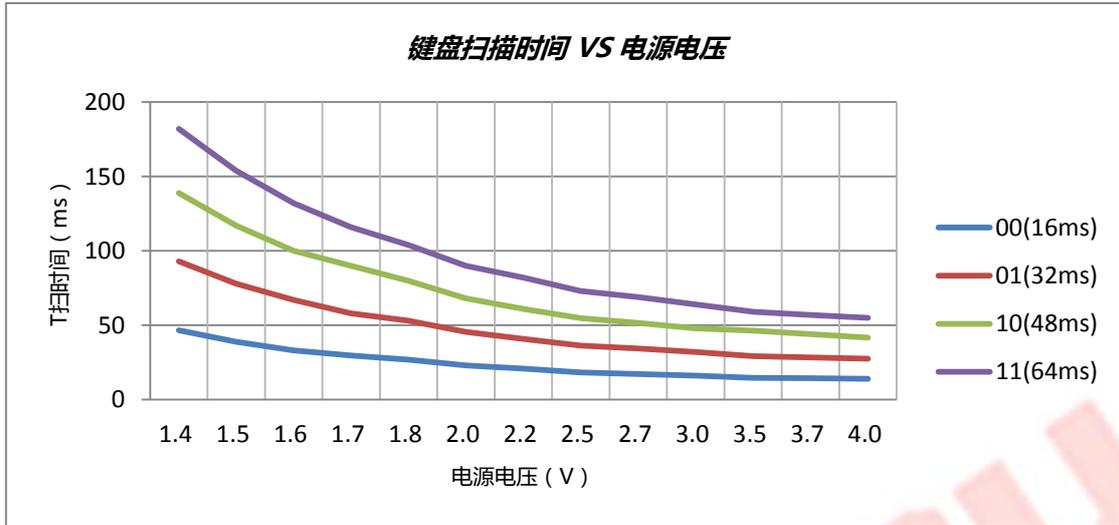
12.7 IO 口输入低电平 VS 电源电压



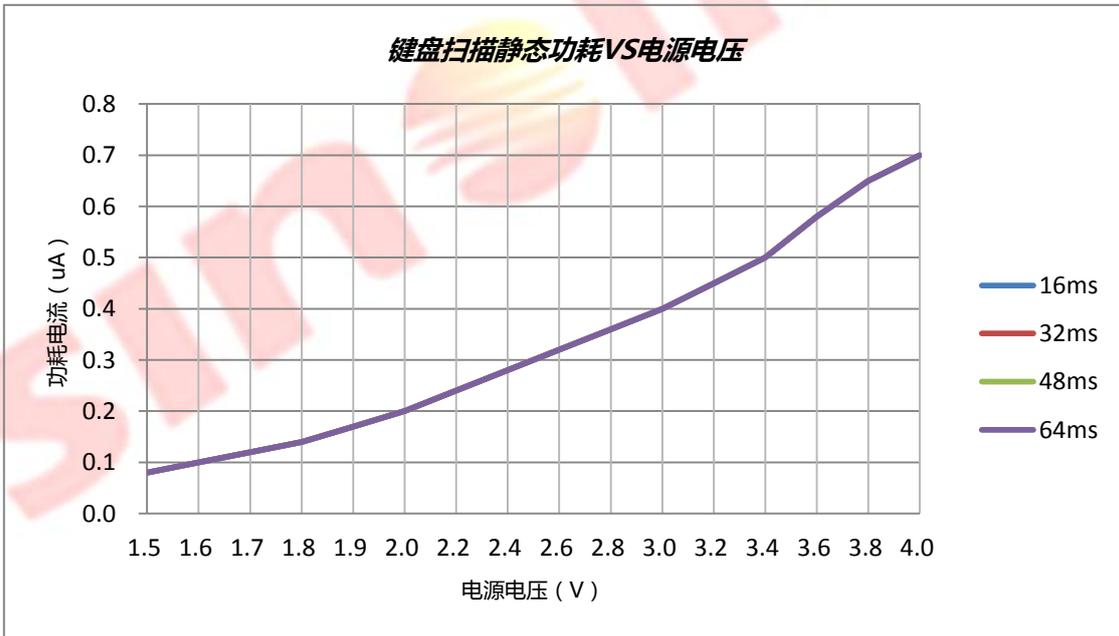
12.8 WDT 溢出时间 VS 电源电压



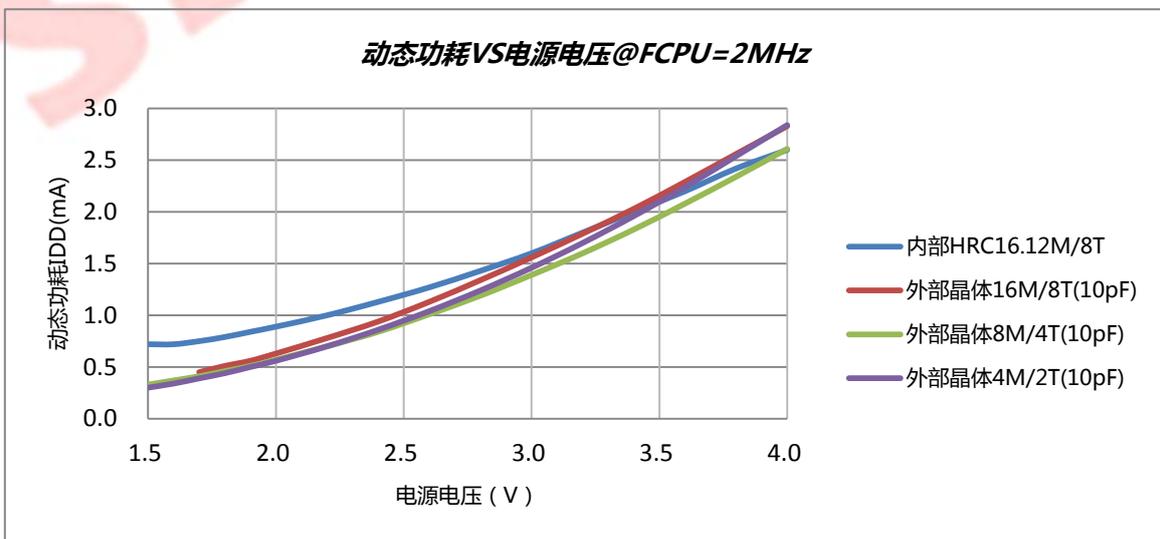
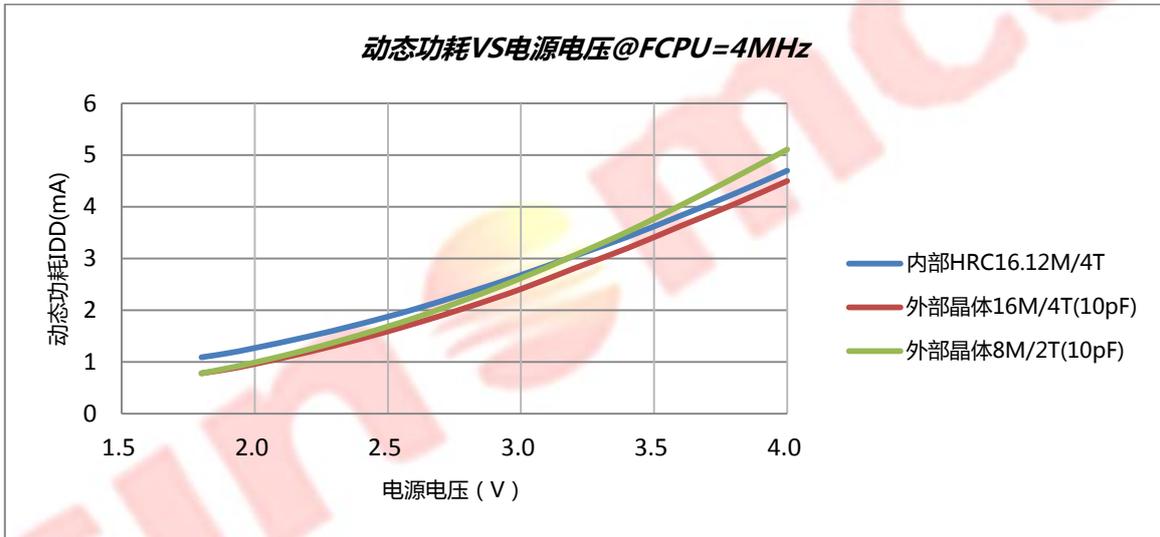
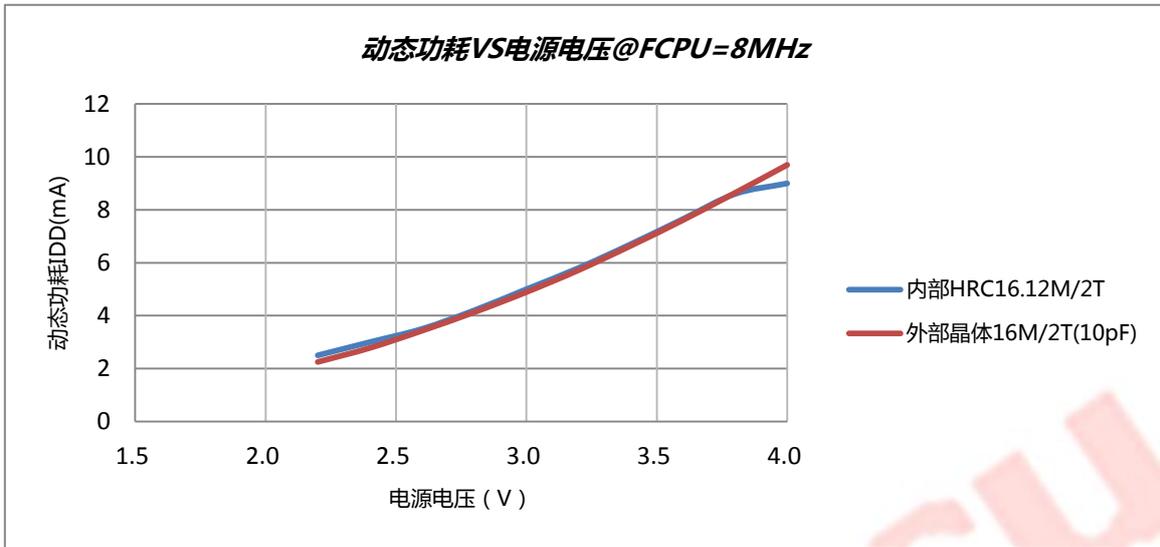
12.9 键盘扫描时间 VS 电源电压

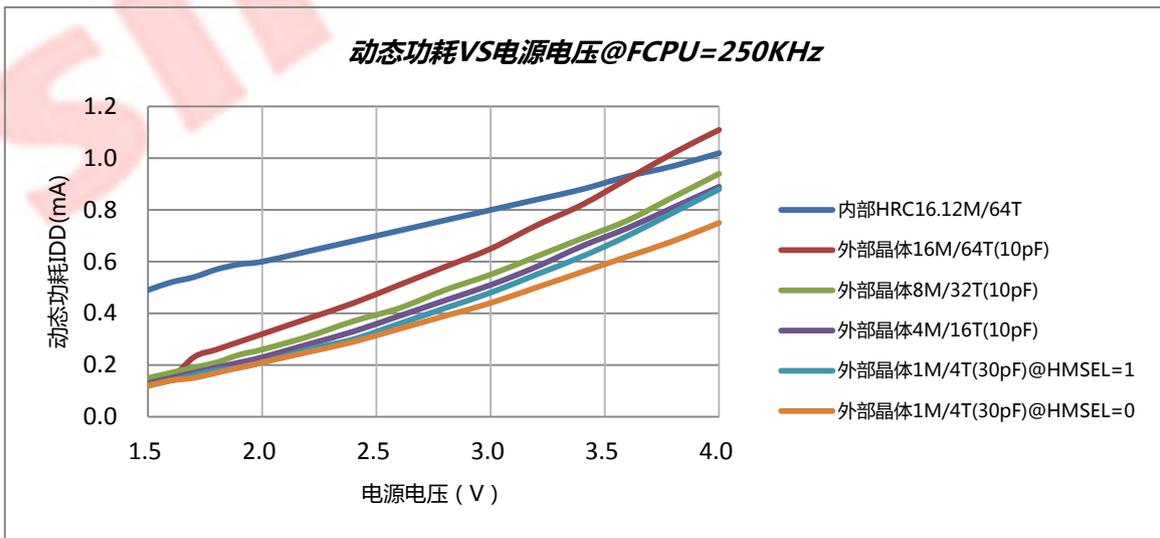
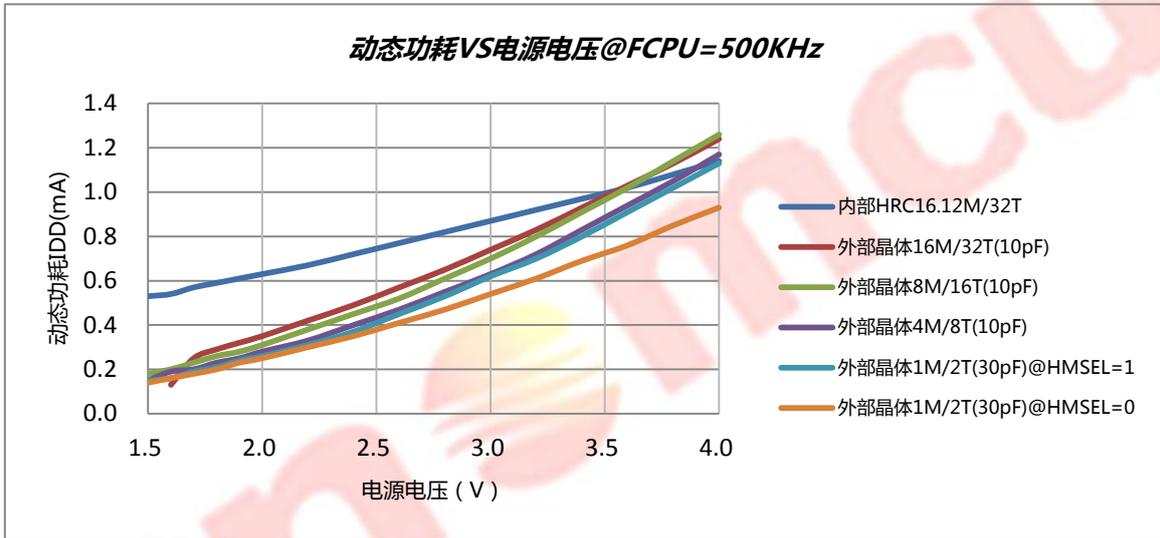
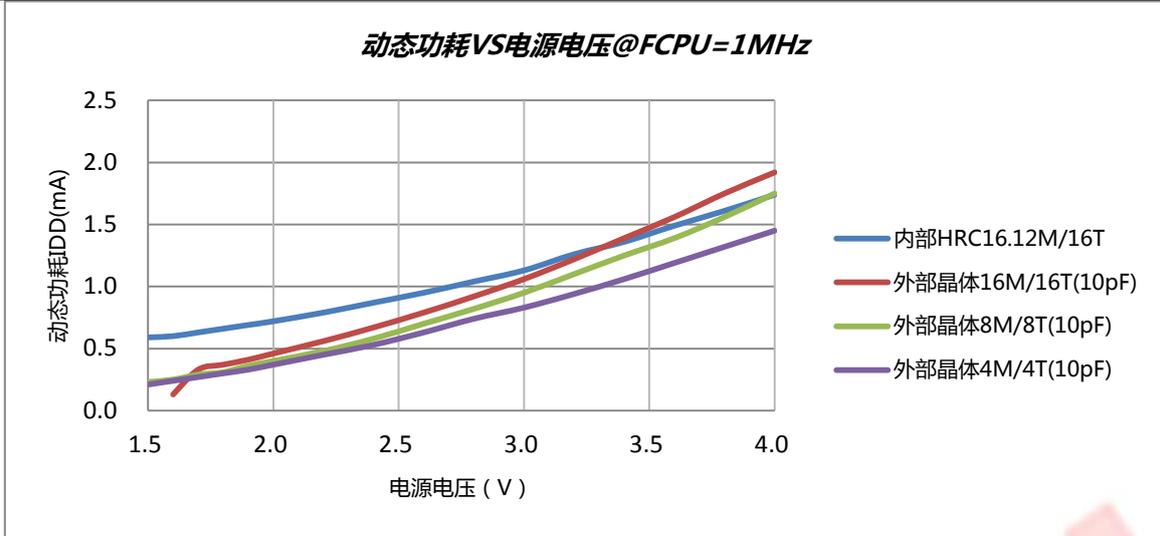


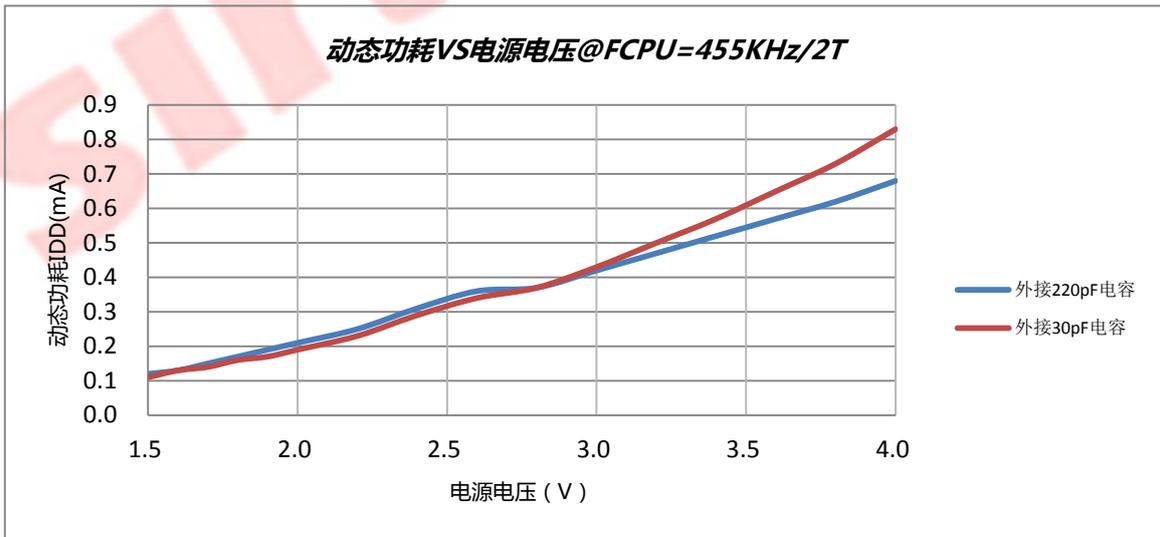
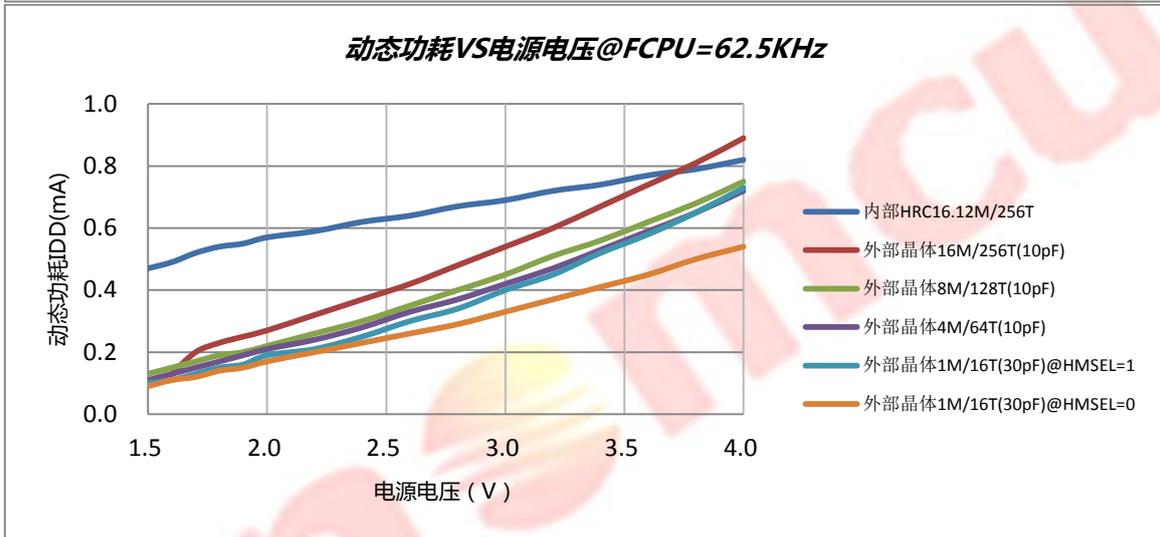
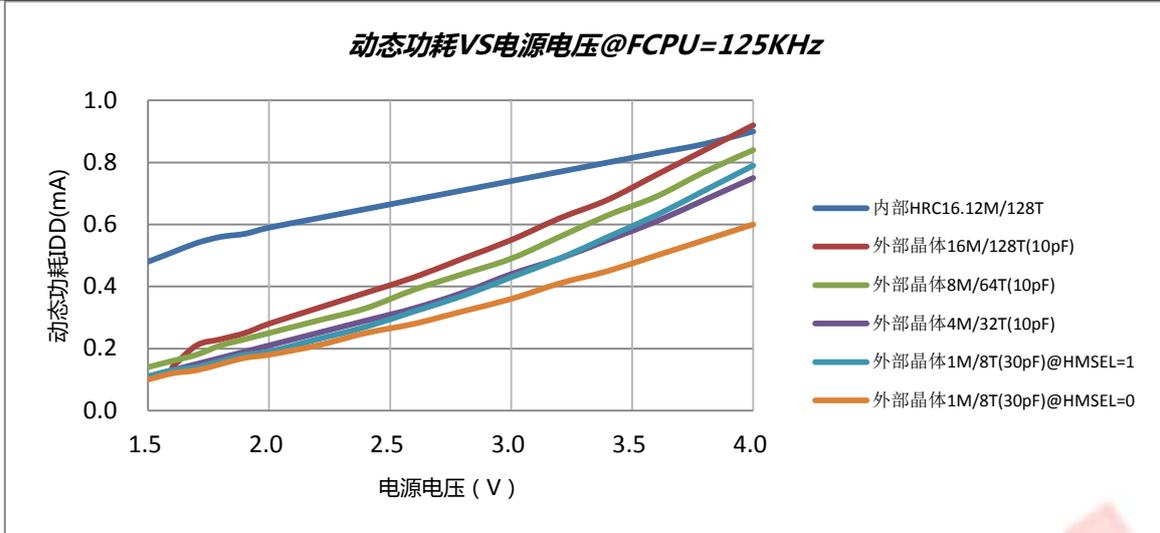
12.10 键盘扫描静态功耗 VS 电源电压

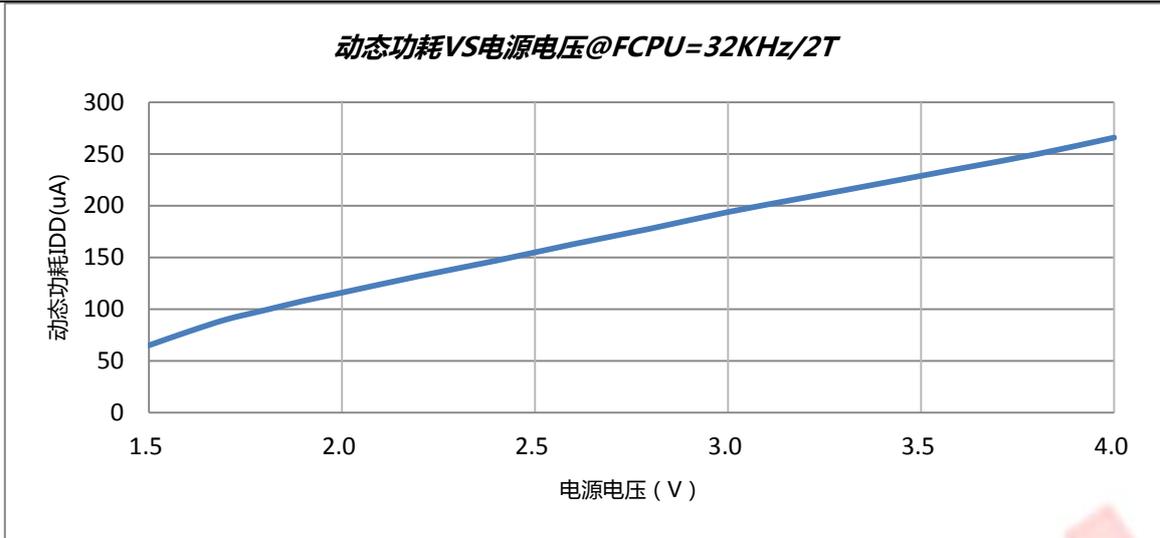


12.11 动态功耗 VS 电源电压

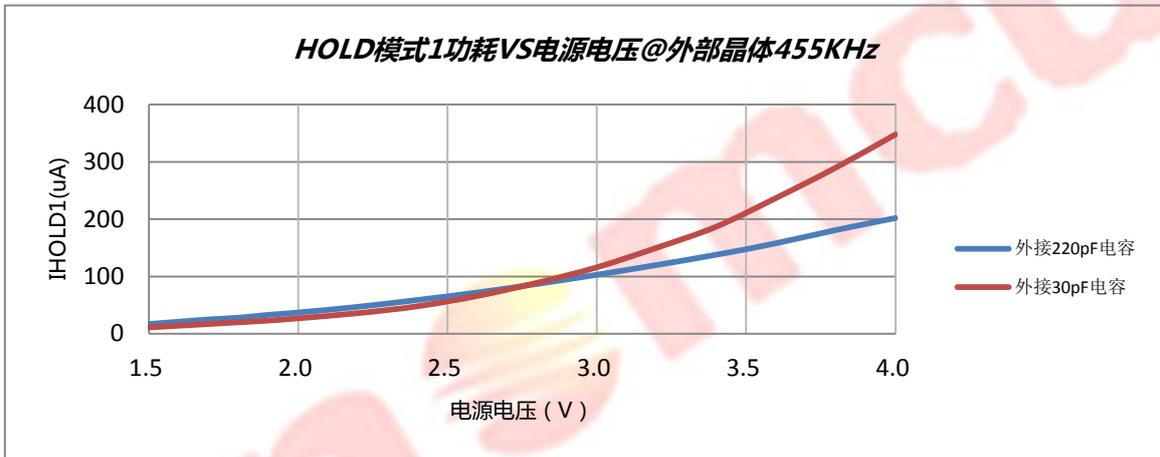
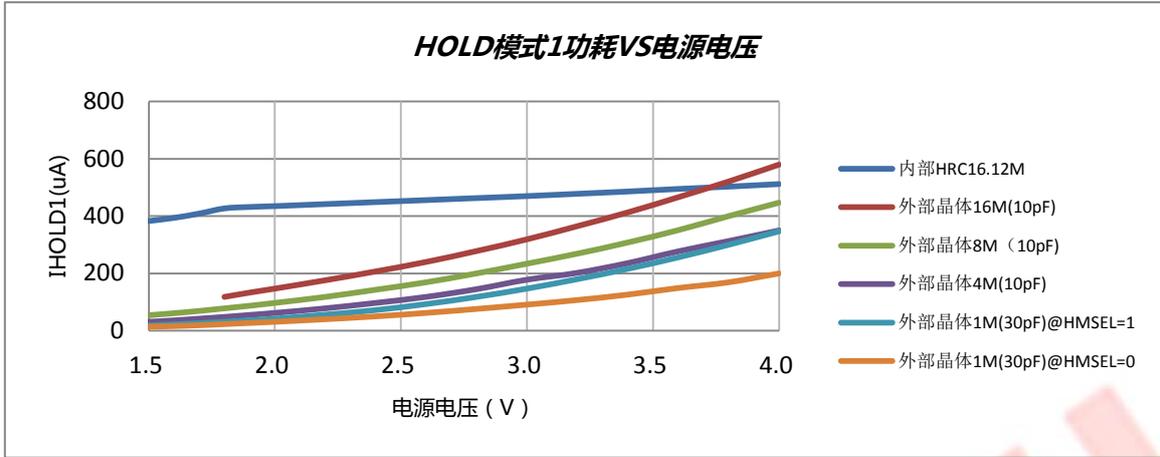




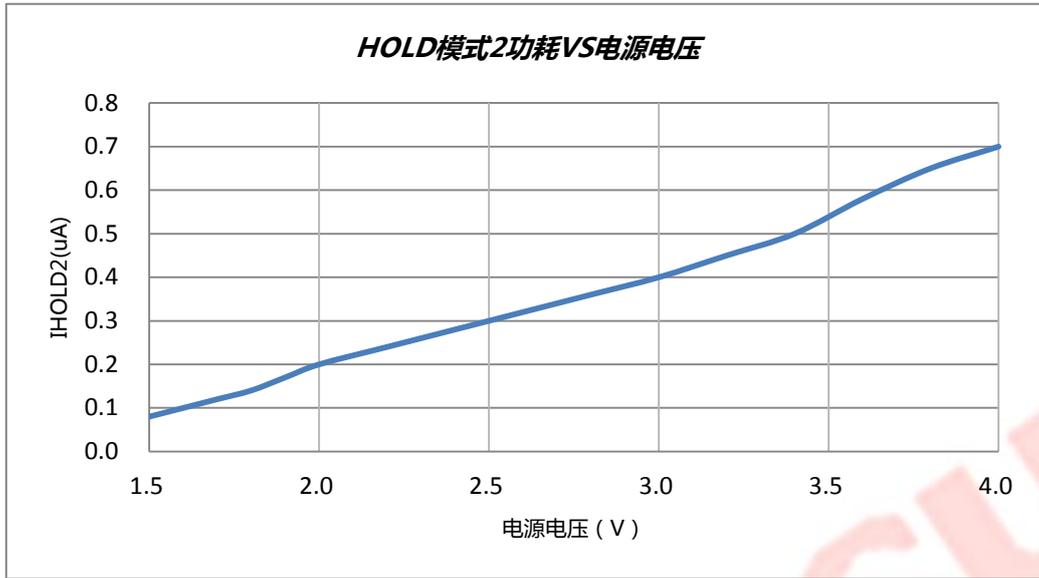




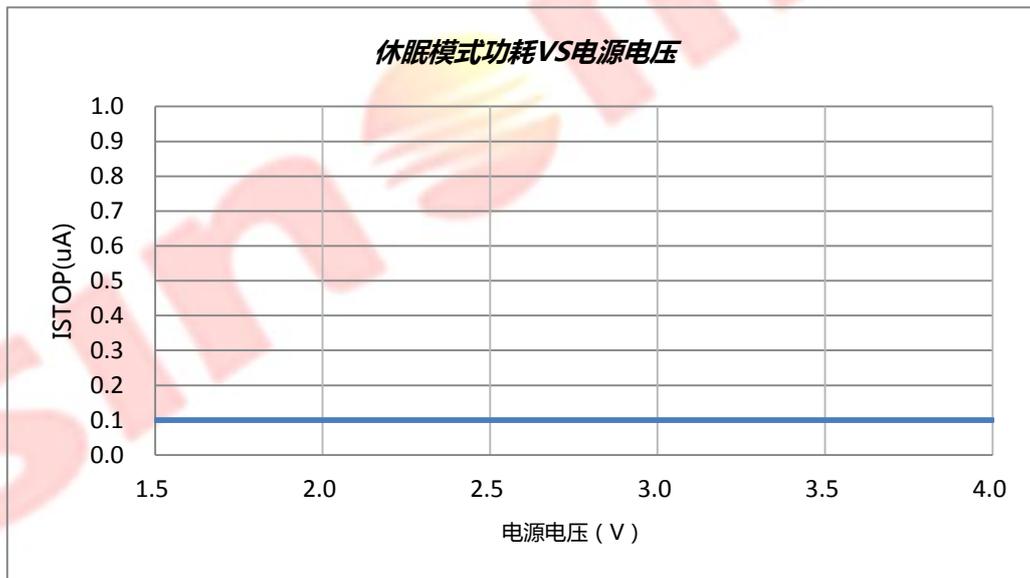
12.12 HOLD 模式 1 功耗 VS 电源电压



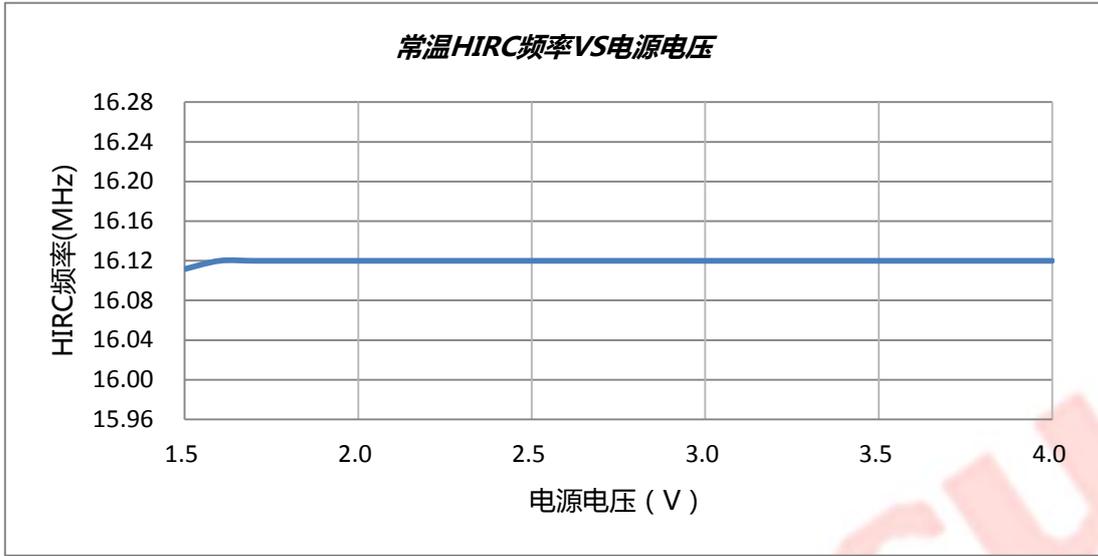
12.13 HOLD 模式2 功耗 VS 电源电压



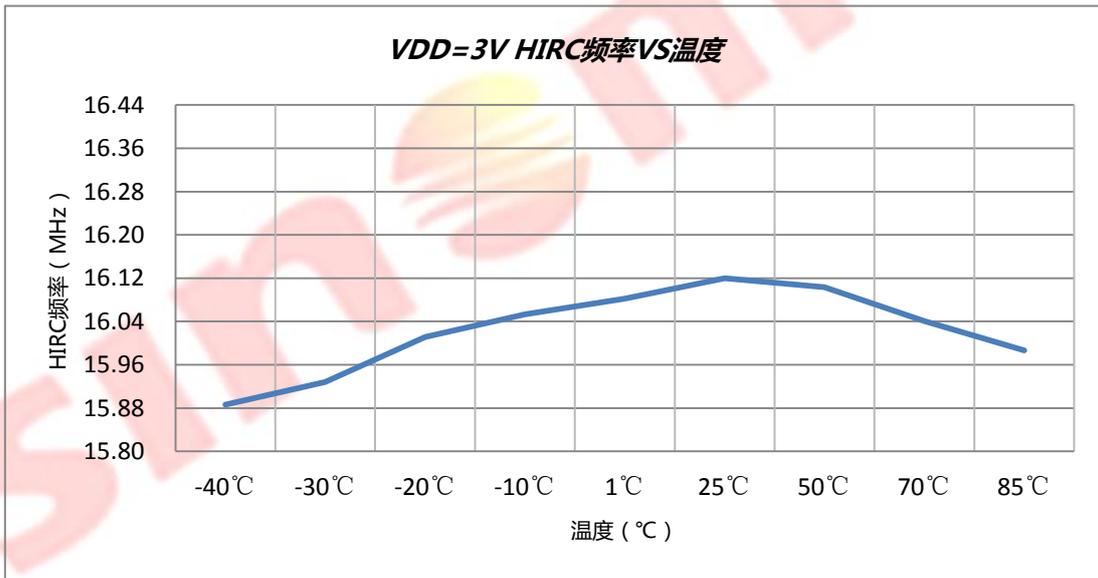
12.14 休眠模式功耗 VS 电源电压



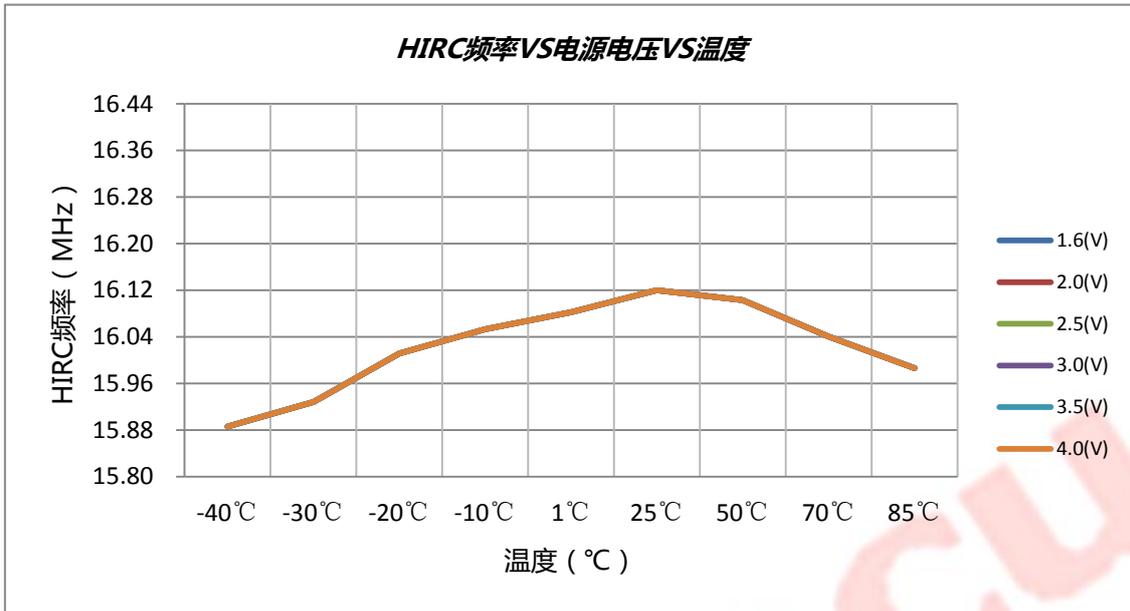
12.15 常温高频振荡频率 VS 电源电压



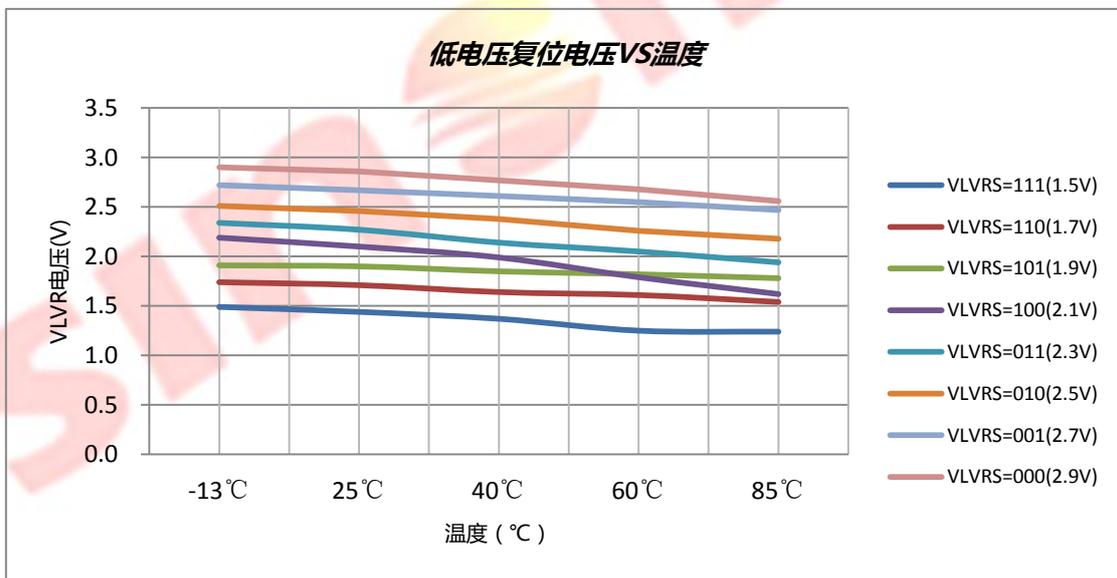
12.16 常压高频振荡频率 VS 温度



12.17 高频振荡频率 VS 电源电压 VS 温度



12.18 低电压复位电压 VS 温度



12.19 最低工作电压 VS 系统时钟 FCPU 关系图

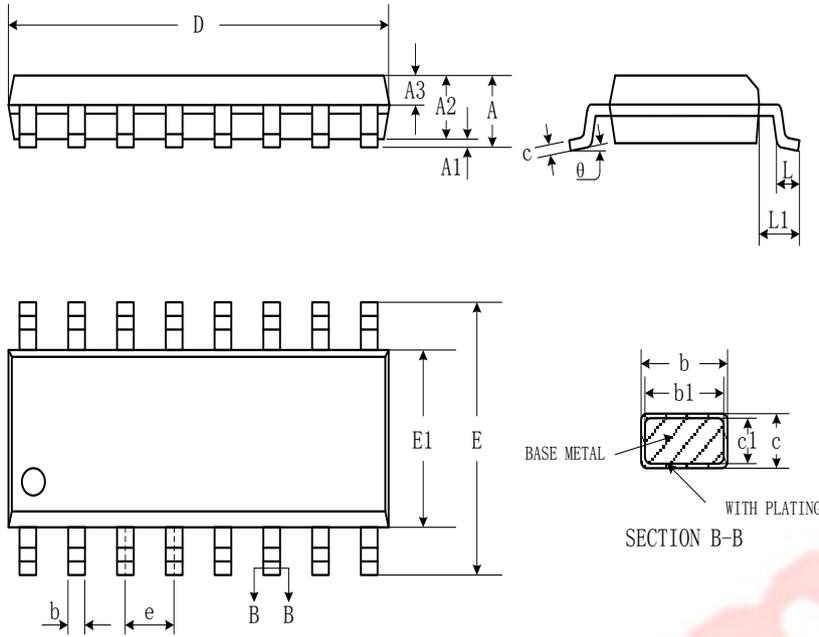
注：系统最低工作电压和系统工作频率 FCPU 有关，不同的工作频率 FCPU 最低工作电压不同。

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 (1.5V@25°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。



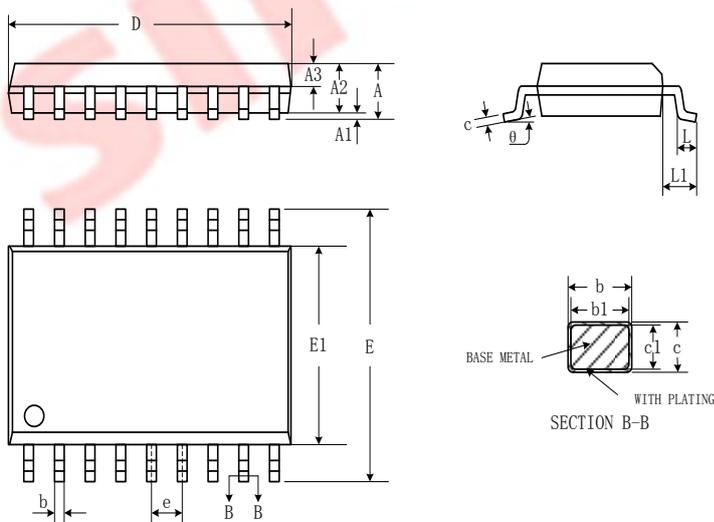
13 封装外形尺寸

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 | | |
| θ | 0 | - | 8° |

SOP18



| SYMBOL | MILLIMETER | | |
|--------|------------|-------|-------|
| | MIN | NOM | MAX |
| A | - | - | 2.70 |
| A1 | 0.08 | 0.18 | 0.28 |
| 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 | 11.25 | 11.45 | 11.65 |
| 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 | | |
| θ | 0 | - | 8° |

14 版本修订记录

| 版本号 | 修订日期 | 修订内容 |
|------|------------|------|
| V1.0 | 2014-03-10 | 新建 |