



WS51F7030 系列物联网 MCU

用户手册

1T 8051 内核, 8KB Flash, 512B SRAM, 128B EEPROM, 12 通道过 CS-10V 低功耗 Touch Key, 1 个 5 位精度 DAC, 3 对 6 通道 16 位互补输出 PWM, 3 个通用定时器, 1 个 UART, 1 个 I2C, 1 个 WDT, 1 个 WKT, 软件 LED 驱动, 软件 LCD 驱动

版本: 1.1

深圳前海维晟智能技术有限公司

电话: +086-0755-23310575

<http://www.wisesun.com>

目录

| | |
|------------------------|----|
| 1 特性 | 5 |
| 2 概述 | 6 |
| 3 命名规则..... | 6 |
| 4 订购信息..... | 7 |
| 5 产品信息..... | 7 |
| 5.1 产品资源..... | 7 |
| 5.2 管脚定义..... | 8 |
| 5.2.1 封装类型：SOP8 | 8 |
| 5.2.2 封装类型：MSOP10..... | 8 |
| 5.2.3 封装类型：SOP14 | 8 |
| 5.2.4 封装类型：SOP16 | 9 |
| 5.3 管脚说明..... | 9 |
| 6 总体架构..... | 11 |
| 6.1 8051 内核..... | 11 |
| 6.1.1 简介..... | 11 |
| 6.1.2 寻址方式..... | 11 |
| 6.1.3 寄存器描述..... | 12 |
| 6.2 芯片结构框图..... | 15 |
| 6.3 存储器分布..... | 16 |
| 6.4 SFR 映像..... | 17 |
| 7 基本功能..... | 19 |
| 7.1 外部 RAM..... | 19 |
| 7.2 Flash 存储器 | 19 |
| 7.2.1 Flash 数据区 | 19 |
| 7.2.2 NVR 数据区..... | 21 |
| 7.2.3 EEPROM 数据区..... | 23 |
| 7.2.4 Checksum 校验..... | 24 |
| 7.2.5 寄存器描述..... | 24 |
| 7.3 IO 端口..... | 28 |
| 7.3.1 简介..... | 28 |
| 7.3.2 IO 端口功能复用 | 28 |
| 7.3.3 寄存器描述..... | 29 |
| 7.4 定时器 0/1/2 | 31 |
| 7.4.1 定时器 0..... | 31 |
| 7.4.2 定时器 1..... | 33 |
| 7.4.3 定时器 2..... | 35 |
| 7.4.4 寄存器描述..... | 37 |
| 7.5 中断系统..... | 41 |
| 7.5.1 简介..... | 41 |
| 7.5.2 外部中断..... | 41 |
| 7.5.3 寄存器描述..... | 41 |
| 7.6 时钟系统..... | 46 |
| 7.6.1 时钟源..... | 46 |

| | | |
|-------|-------------------------------|----|
| 7.6.2 | 时钟控制关系..... | 46 |
| 7.6.3 | 寄存器描述..... | 46 |
| 7.7 | 复位系统..... | 48 |
| 7.7.1 | 上电复位（POR）和掉电复位（BOR）..... | 48 |
| 7.7.2 | 看门狗（WDT）复位..... | 48 |
| 7.7.3 | 低电压检测（LVD）复位..... | 48 |
| 7.7.4 | 外部硬复位和内部软复位..... | 49 |
| 7.7.5 | 寄存器描述..... | 49 |
| 7.8 | 电源管理..... | 50 |
| 7.8.1 | LDO..... | 50 |
| 7.8.2 | IDLE 模式..... | 50 |
| 7.8.3 | STOP 模式..... | 50 |
| 7.8.4 | 寄存器描述..... | 50 |
| 8 | 增强功能..... | 52 |
| 8.1 | 脉宽调制（PWM）..... | 52 |
| 8.1.1 | 简介..... | 52 |
| 8.1.2 | PWM 工作原理..... | 52 |
| 8.1.3 | 寄存器描述..... | 54 |
| 8.2 | 通用串行接口（UART 0）..... | 57 |
| 8.2.1 | 简介..... | 57 |
| 8.2.2 | UART0..... | 57 |
| 8.2.3 | 寄存器描述..... | 60 |
| 8.3 | I ² C 接口（I2C）..... | 62 |
| 8.3.1 | 简介..... | 62 |
| 8.3.2 | I2C 总线互联..... | 62 |
| 8.3.3 | I2C 通信原理..... | 62 |
| 8.3.4 | I2C 工作模式..... | 63 |
| 8.3.5 | 寄存器描述..... | 63 |
| 8.4 | 看门狗定时器（WDT）..... | 66 |
| 8.4.1 | 简介..... | 66 |
| 8.4.2 | 功能描述..... | 66 |
| 8.4.3 | 寄存器描述..... | 66 |
| 8.5 | 自唤醒定时器（WKT）..... | 68 |
| 8.5.1 | 简介..... | 68 |
| 8.5.2 | 功能描述..... | 68 |
| 8.5.3 | 寄存器描述..... | 68 |
| 8.6 | 低压差线性稳压器（LDO）..... | 69 |
| 8.6.1 | 简介..... | 69 |
| 8.6.2 | 内部基准及输出电压校准..... | 69 |
| 8.6.3 | LDO 工作模式..... | 69 |
| 8.6.4 | 寄存器描述..... | 69 |
| 8.7 | 低电压检测（LVD）..... | 71 |
| 8.7.1 | 简介..... | 71 |
| 8.7.2 | 功能描述..... | 71 |
| 8.7.3 | 寄存器描述..... | 71 |

| | | |
|--------|------------------------|----|
| 8.8 | 数模转换器 (DAC) | 72 |
| 8.8.1 | 简介 | 72 |
| 8.8.2 | 功能描述 | 72 |
| 8.8.3 | 寄存器描述 | 73 |
| 8.9 | 触摸按键 (Touch Key) | 74 |
| 8.9.1 | 简介 | 74 |
| 8.9.2 | 工作模式 | 74 |
| 8.10 | 软件 LED 驱动 (SLED) | 75 |
| 8.10.1 | 简介 | 75 |
| 8.10.2 | LED 显示原理 | 75 |
| 8.10.3 | LED 与 Touch Key 共用实现方法 | 75 |
| 8.10.4 | 寄存器描述 | 76 |
| 8.11 | 软件 LCD 驱动 (SLCD) | 79 |
| 8.11.1 | 简介 | 79 |
| 8.11.2 | 1/2 偏压软件 LCD | 79 |
| 8.11.3 | 寄存器描述 | 79 |
| 9 | 程序下载和仿真 | 81 |
| 9.1 | 程序下载 | 81 |
| 9.2 | 在线仿真 | 81 |
| 10 | 电气特性 | 82 |
| 10.1 | 极限参数 | 82 |
| 10.2 | 直流电气特性 | 82 |
| 10.3 | 交流电气特性 | 84 |
| 10.4 | POR 电气特性 | 84 |
| 10.5 | 内部 HRC 电气特性 | 85 |
| 10.6 | 内部 LRC 电气特性 | 86 |
| 10.7 | LVD 电气特性 | 87 |
| 10.8 | DAC 电气特性 | 87 |
| 11 | 封装信息 | 90 |
| 11.1 | WS51F7030S08U (SOP8) | 90 |
| 11.2 | WS51F7030M10U (MSOP10) | 90 |
| 11.3 | WS51F7030S14U (SOP14) | 91 |
| 11.4 | WS51F7030S16U (SOP16) | 91 |
| 12 | 附录 | 92 |
| 13 | 版本历史 | 99 |

1 特性

- **工作电压:** 1.7V-5.5V
- **工作温度:** -40~+105°C
- **封装类型:** SOP16/SOP14/MSOP10/SOP8
- **内核:** 增强型 1T 8051
- **最高工作频率:** 16MHz
- **Flash ROM:** 8K Bytes, 10 万次擦写
- **EEPROM:** 128 Bytes, 无需擦, 10 万次写入
- **SRAM:** 内部 256 Bytes, 外部 512 Bytes
- **时钟 (3.3V@25°C)**
 - 内置 RC 振荡器 LRC: 32KHz, 可调精度±1%
 - 内置 RC 振荡器 HRC: 16MHz, 可调精度±1%
- **中断**
 - 7 个有效中断源, 两级中断优先级
 - 4 个外部中断, 可配置任意引脚输入, 支持上沿/下沿/双沿触发
- **定时器**
 - 3 个 16 位通用定时器 0/1/2, 兼容标准 8051
- **IO 端口**
 - 14 个通用 GPIO 口
 - 支持推挽/开漏/上拉/下拉/高阻等模式
 - 上拉可选 60KΩ 或 10KΩ, 下拉为 15 KΩ
 - 推电流支持 20mA, 灌电流 45mA
- **通用串行接口 (UART 0)**
 - 1 个 UART 接口: UART0, 兼容标准 8051
- **I2C 接口 (I2C)**
 - 内置 1 路 I2C 接口, 支持主从模式, 支持标准/快速模式
- **看门狗 (WDT)**
 - 15 位看门狗定时器, 计数时钟为 LRC 时钟
 - 8 位调节精度, 调整范围为 7.8125ms-1s
 - 可配置看门狗产生复位或中断
- **自唤醒定时器 (WKT)**
 - 可选 0.5s/1s 中断, 支持中断唤醒
- **脉宽调制 (PWM)**
 - 3 对 6 通道 16 位 PWM, 每通道可独立设置周期和占空比
- 可选时钟源, 可直接输出内部时钟
- 支持互补输出和死区控制
- **低电压检测 (LVD)**
 - 8 级电压检测 1.8/2.0/2.4/2.8/3.0/3.4/3.7/4.2V
 - 可设置低电压复位或中断
- **触摸按键 (Touch Key)**
 - 共 12 通道, 可一次全部扫描
 - 支持触摸待机唤醒
 - 支持触摸按键与软件 LED 驱动共用引脚
 - 超低功耗, 增加一个按键功耗增加 0.3uA 功耗
 - 通过 CS 动态 10V 测试
- **数模转换器 (DAC)**
 - 1 个 5 位精度 DAC
 - 可选外部 VDD、内部 1.5V 基准或外部 DAVF 作为基准电源
- **软件 LED 驱动**
 - 采用交替轮流 COM 端口方式进行 LED 扫描
 - 最多支持 11seg*12com 的 LED 驱动
- **软件 LCD 驱动**
 - 支持 1/2 偏压软件 LCD 驱动
 - 最多支持 5 个 COM
- **芯片复位**
 - 支持硬复位、软复位、看门狗复位、LVD 复位和上电/掉电复位
- **程序加密及保护**
 - 内置程序读保护
 - 内置程序 ID 加密保护功能
- **程序下载和仿真**
 - 支持 ISP 和 IAP, IAP 可配置大小
 - 支持在线仿真功能
- **低功耗模式**
 - STOP 模式最低电流 1.7uA
 - 16Mhz@5V 运行典型功耗 1.5mA
- **抗干扰能力**
 - ESD > 8KV
 - EFT > 4.5KV

2 概述

WS51F7030 系列芯片是基于增强型 1T 8051 内核的 8 位微控制器，指令完全兼容传统 8051，而运行速度比传统 8051 快 10 倍。WS51F7030 集成 8KB Flash、512B SRAM、128B 独立 EEPROM、12 通道过 CS-10V 低功耗 Touch Key、1 个 5 位精度 DAC、3 对 6 通道 12 位互补输出 PWM、3 个 16 位定时器、1 个 UART、1 个 I2C、16MHz 内部 RC 振荡器、32KHz 内部 RC 振荡器、14 个 GPIO、软件 LED 驱动、软件 LCD 驱动等资源。为了提高芯片可靠性，WS51F7030 还集成了上电掉电复位、8 级可选电压低电压检测（LVD）、低功耗独立看门狗计数器（WDT）、自唤醒定时器（WKT）等模块。WS51F7030 具有非常优异的抗干扰能力和低功耗特性，非常适合用在家用照明、控制开关、无线通讯、小家电等物联网工业控制及消费电子领域。

3 命名规则

| 名称 | WS | 51 | F | 7 | 0 | 3 | 0 | X | S | 16 | U |
|----|----|----|---|---|---|---|---|---|---|----|---|
| 序号 | ① | ② | ③ | ④ | ⑤ | ⑥ | ⑦ | ⑧ | ⑨ | ⑩ | ⑪ |

| 序号 | 含义 |
|----|--|
| ① | WISESUN 的简称 |
| ② | 8: RISC; 51: 8051; |
| ③ | 产品类型 (F: Flash; M: MTP; P: OTP; V: RISC-V) |
| ④ | 产品系列 (5: RSIC 通用型; 6: 8051 通用型; 7: 8051 触摸型;) |
| ⑤ | 芯片功能类别编号 (0、1、2、3...) 产品为 5 系列时 0: IO 型 1: IO+AD 型 2: IO+TK 型 3: IO+AD+TK 型 产品为 6 系列时 0: IO 型 1: IO+ADC 型 2: IO+ADC+OP+CMP 型 3: IO+LCD/LED 型 4: IO+ADC+LCD/LED 型 5: IO+ADC+OP+CMP+LCD/LED 型 产品为 7 系列时 0: IO+TK 型 1: IO+ADC+TK 2: IO+ADC+OP+CMP+TK 型 3: IO+LCD/LED+TK 型 4: IO+ADC+LCD/LED+TK 型 5: IO+ADC+OP+CMP+LCD/LED+TK 型 |
| ⑥ | 存储大小 (1: 2KB; 2: 4KB; 3: 8KB; 4: 16KB; 5: 32KB; 6: 64KB; 7: 128KB; 8: 256KB; 9: 512KB; A: 1MB) |
| ⑦ | 子编号 (0、1、2、3...) |
| ⑧ | 芯片版本号 (缺省、B、C、D...) |

| | |
|---|--|
| ⑨ | 封装形式 (R: SOT23-6; D: DIP; I: SDIP; S: SOP; P: SSOP; T: TSSOP; M: MSOP; Q: QFP; L: LQFP; F: TQFP; N: QFN; K: SKDIP) |
| ⑩ | 引脚数 |
| ⑪ | 包装方式 (R: 盘装; U: 管装; T: 编带) |

4 订购信息

| 产品编号 | 封装 | 包装 |
|---------------|--------|----|
| WS51F7030S08U | SOP8 | 管装 |
| WS51F7030M10U | MSOP10 | 管装 |
| WS51F7030S14U | SOP14 | 管装 |
| WS51F7030S16U | SOP16 | 管装 |

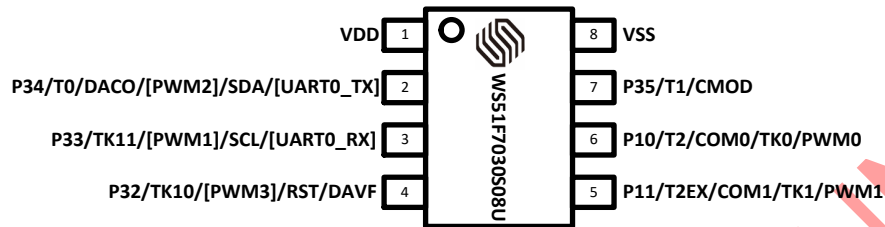
5 产品信息

5.1 产品资源

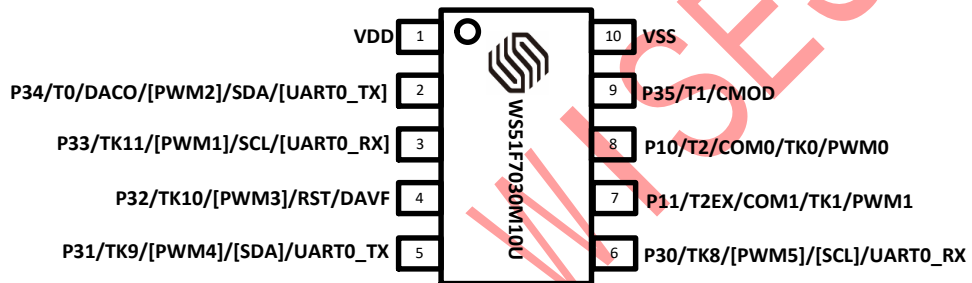
| 型号 资源 | WS51F7030S08U | WS51F7030M10U | WS51F7030S14U | WS51F7030S16U |
|-------------------------|---------------|---------------|---------------|---------------|
| Flash 存储器 (Bytes) | 8K | 8K | 8K | 8K |
| XDATA RAM (Bytes) | 512 | 512 | 512 | 512 |
| GPIO | 6 | 8 | 12 | 14 |
| UART | 1 | 1 | 1 | 1 |
| I2C | 1 | 1 | 1 | 1 |
| 12 位 PWM (channels) | 4 | 6 | 6 | 6 |
| Touch Key (channels) | 4 | 6 | 10 | 12 |
| 软件 LED 驱动 | √ | √ | √ | √ |
| 软件 LCD 驱动 | √ | √ | √ | √ |
| WDT | √ | √ | √ | √ |
| WKT | √ | √ | √ | √ |
| 8 级电压 LVD | √ | √ | √ | √ |
| 5 位 DAC | √ | √ | √ | √ |
| HRC (16MHz) | √ | √ | √ | √ |
| LRC (32KHz) | √ | √ | √ | √ |
| 程序加密保护 | √ | √ | √ | √ |
| ISP/IAP | √ | √ | √ | √ |
| 在线仿真 | √ | √ | √ | √ |
| 工作电压 | 1.7V-5.5V | 1.7V-5.5V | 1.7V-5.5V | 1.7V-5.5V |
| 封装形式 | SOP8 | MSOP10 | SOP14 | SOP16 |

5.2 管脚定义

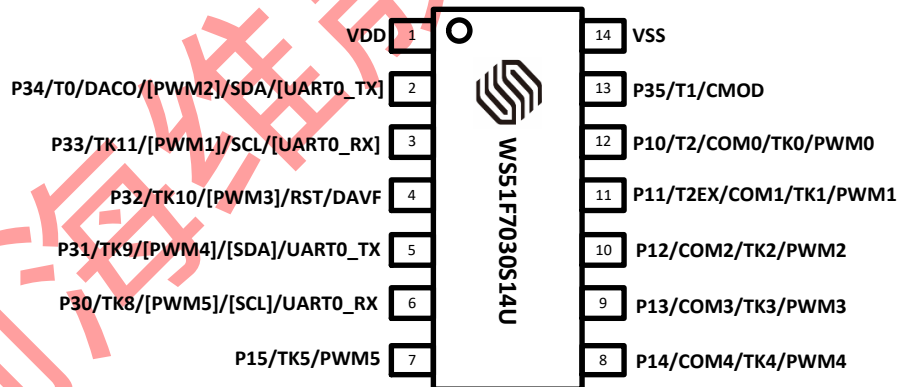
5.2.1 封装类型：SOP8

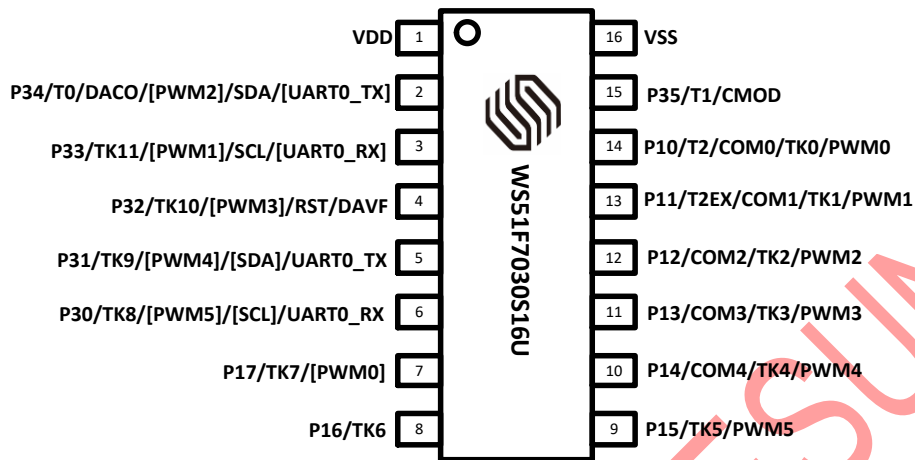


5.2.2 封装类型：MSOP10



5.2.3 封装类型：SOP14



5.2.4 封装类型：SOP16

5.3 管脚说明

| 管脚名称 | | | | 功能说明 | 默认状态 | |
|-------|-------|-------|------|-----------------------------------|---|------------------|
| 16PIN | 14PIN | 10PIN | 8PIN | | | |
| 1 | 1 | 1 | 1 | VDD | 电源 VDD | VDD |
| 2 | 2 | 2 | 2 | P34/T0/DACO/[PWM2]/SDA/[UART0_TX] | 通用双向 GPIO 口 P34 Timer 0 端口 T0 DAC 数模转换输出 DACO PWM2 备选输出端口 I2C 的 SDA 端口 UART0 的 UART0_TX 端口 | SDA |
| 3 | 3 | 3 | 3 | P33/TK11/[PWM1]/SCL/[UART0_RX] | 通用双向 GPIO 口 P33 触摸按键通道 TK11 PWM1 备选输出端口 I2C 的 SCL 端口 UART0 的 UART0_RX 端口 | SCL |
| 4 | 4 | 4 | 4 | P32/TK10/[PWM3]/RST/DAVF | 通用双向 GPIO 口 P32 触摸按键通道 TK10 PWM3 备选输出端口 外部硬复位 RST DAC 基准电源输入 DAVF | 可选 高阻 /RST |
| 5 | 5 | 5 | - | P31/TK9/[PWM4]/[SDA]/UART0_TX | 通用双向 GPIO 口 P31 触摸按键通道 TK9 PWM4 备选输出端口 I2C 的 SDA 端口 UART0 的 UART0_TX 端口 | 高阻 |
| 6 | 6 | 6 | - | P30/TK8/[PWM5]/[SCL]/UART0_RX | 通用双向 GPIO 口 P30 触摸按键通道 TK8 | 高阻 |

| | | | | | | |
|----|----|----|---|------------------------|--|-----|
| | | | | | PWM5 备选输出端口 I2C 的 SCL 端口 UART0 的 UART0_RX 端口 | |
| 7 | - | - | - | P17/TK7/[PWM0] | 通用双向 GPIO 口 P17 触摸按键通道 TK7 PWM0 备选输出端口 | 高阻 |
| 8 | - | - | - | P16/TK6 | 通用双向 GPIO 口 P16 触摸按键通道 TK6 | 高阻 |
| 9 | 7 | - | - | P15/TK5/PWM5 | 通用双向 GPIO 口 P15 触摸按键通道 TK5 PWM5 输出端口 | 高阻 |
| 10 | 8 | - | - | P14/COM4/TK4/PWM4 | 通用双向 GPIO 口 P14 软件 LCD 的 COM4 口 触摸按键通道 TK4 PWM4 输出端口 | 高阻 |
| 11 | 9 | - | - | P13/COM3/TK3/PWM3 | 通用双向 GPIO 口 P13 软件 LCD 的 COM3 口 触摸按键通道 TK3 PWM3 输出端口 | 高阻 |
| 12 | 10 | - | - | P12/COM2/TK2/PWM2 | 通用双向 GPIO 口 P12 软件 LCD 的 COM2 口 触摸按键通道 TK2 PWM2 输出端口 | 高阻 |
| 13 | 11 | 7 | 5 | P11/T2EX/COM1/TK1/PWM1 | 通用双向 GPIO 口 P11 Timer 2 端口 T2EX 软件 LCD 的 COM1 口 触摸按键通道 TK1 PWM1 输出端口 | 高阻 |
| 14 | 12 | 8 | 6 | P10/T2/COM0/TK0/PWM0 | 通用双向 GPIO 口 P10 Timer 2 端口 T2 软件 LCD 的 COM0 口 触摸按键通道 TK0 PWM0 输出端口 | 高阻 |
| 15 | 13 | 9 | 7 | P35/T1/CMOD | 通用双向 GPIO 口 P35 Timer 1 端口 T1 触摸按键电容 CMOD | 高阻 |
| 16 | 14 | 10 | 8 | VSS | 地线 VSS | VSS |

备注:

1. 信号端口复用功能设置方法详见[表 7.3.2-1](#)和[表 7.3.3-2](#)
2. LED 输出与 Touch Key 和 GPIO 输出引脚功能复用, 详见[表 7.3.2-1](#)及[8.10 软件 LED 驱动](#)章节描述

6 总体架构

6.1 8051 内核

6.1.1 简介

标准 8051 即 MCS-51 为 12 周期的 8051 CPU，而 WS51F7030 系列芯片采用的是单周期 8051 CPU，完全兼容 MCS-51 指令集。

6.1.2 寻址方式

8051 的内核寻址方式有：立即寻址、直接寻址、间接寻址、寄存器寻址、相对寻址、变址寻址和位寻址。

6.1.2.1 立即寻址

立即寻址是在指令操作数中直接给出参与运算的操作数，指令举例如下：

MOV A, #60H (这条指令把 60H 直接送到累加器 A)

6.1.2.2 直接寻址

直接寻址指令操作数给出的是参与运算操作数的地址。直接寻址方式只能用于特殊功能寄存器、内部数据区和位地址空间，而特殊功能寄存器和位地址空间只能用直接寻址方式访问。指令举例如下：

ANL 50H, #91H (把 50H 单元中的数与 91H 相“与”，结果存放在 50H 单元中，其中 50H 为内部 RAM 的地址)

6.1.2.3 间接寻址

间接寻址采用 R0 或 R1 前添加“@”符号来表示。假设 R1 中的数据是 40H，内部数据存储器 40H 单元的数据为 55H，则指令为

MOV A, @R1 (把数据 55H 传送至累加器 A)

6.1.2.4 寄存器寻址

寄存器寻址对选定的工作寄存器 R7-R0、累加器 A、通用寄存器 B、地址寄存器和进位 C 中的数进行操作，其中寄存器 R7-R0 由指令码的低 3 位表示，ACC、B、DPTR 及进位 C 隐含在指令码中。因此，寄存器寻址也包含一种隐含寻址方式。工作寄存器工作区由状态寄存器 PSW 的 RS1、RS0 决定。指令操作数指定的寄存器均指当前工作区的寄存器。

INC R0 (R0 中的数据+1 写回 R0)

6.1.2.5 相对寻址

相对寻址是将程序计数器 PC 中的当前值与指令第二字节给出的数相加，其结果作为转移指令的转移地址。转移地址也成为转移目的地址，PC 中的当前值成为基地址，指令第二字节给出的数成为偏移量。由于目的地址是相对于 PC 中的基地址而言，所以这种寻址方式成为相对寻址。偏移量为带符号的数，所能表示的范围为 +127--128。这种寻址方式主要用于转移指令。

JC \$+50H (表示若进位位 C 为 0，则程序计数器 PC 中的内容不改变，即不转移。若进位位 C 为 1，则以 PC 中的当前值及基地址，加上偏移量 50H 后得到的结果作为该转移指令的目的地址)

6.1.2.6 变址寻址

在变址寻址方式中,指令操作数制定一个存放变址基址的变址寄存器。变址寻址时,偏移量与变址基值相加,其结果作为操作数的地址。变址寄存器有程序计数器 PC 和地址寄存器 DPTR。

MOVC A, @A+DPTR (表示累加器 A 为偏移量寄存器,其内容与地址寄存器 DPTR 中的内容相加,其结果作为操作数的地址,取出该单元中的数送入累加器 A 中)

6.1.2.7 位寻址

位寻址是指对一些可进行位操作的内部 RAM 和特殊功能寄存器进行位操作时的寻址方式。在进行位操作时,借助于进位位 C 作为位操作累加器,指令操作数直接给出该位的地址,然后根据操作码对该位进行位操作。

MOV C, 20H (将地址为 20H 的位操作寄存器值送入进位位 C 中)

6.1.3 寄存器描述

6.1.3.1 程序计数器 PC

程序计数器 PC 寄存器为 16 位,是专门用来控制指令执行顺序的寄存器,它没有寄存器地址。单片机上电或复位后,PC 值为 0,单片机从零地址开始执行程序。

6.1.3.2 累加器 ACC

累加器 ACC 是一个常用的专用寄存器,指令系统中采用 A 作为累加器的助记符,常用于存放算术或逻辑运算的操作数及运算结果。

表 6.1.3-1 累加器 ACC

| E0H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|----------|---|---|---|---|---|---|---|
| ACC | ACC[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

6.1.3.3 通用寄存器 B

B 在乘除法运算中需要和 ACC 配合使用。MUL AB 指令把 ACC 和 B 中 8 位无符号数相乘,所得的 16 位乘积的低字节存放在 A 中,高字节存放在 B 中。DIV AB 指令用 B 除以 A,整数商存放在 A 中,余数存放在 B 中。寄存器 B 还可以用作通用暂存寄存器。

表 6.1.3-2 通用寄存器 B

| F0H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|---|---|---|---|---|---|---|
| B | B | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

6.1.3.4 堆栈指针 SP

堆栈指针 SP 是一个 8 位专用寄存器。它指示出堆栈顶部在内部 RAM 块中的位置。系统复位后,SP 初始化为 07H,使得堆栈事实上由 08H 单元开始,考虑 08H-1FH 单元分别属于工作寄存器组 1-3,若在程序设计中用到这些区,则最好 SP 改变为 80H 或更大的为宜。

在程序调试中,经常出现堆栈溢出的情况,为了方便找到程序故障,设置了一个记录最大堆栈的寄存器 SPMAX,可以通过 Debug 接口读出,地址 0x1A。

表 6.1.3-3 堆栈指针 SP

| 81H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|---|---|---|---|---|---|---|
| SP | SP | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

表 6.1.3-4 堆栈最大值 SPMAX

| 8407H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-------|---|---|---|---|---|---|---|
| SP | SPMAX | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

6.1.3.5 数据指针 DPTR

数据指针 DPTR0/DPTR1 是两个 16 位专用寄存器，它们的高位字节寄存器用 DPOH/DP1H 表示，低位字节寄存器用 DPOL/DP1L 表示，通过 DPS(PSW.1) 可选择使用 DPTR0/DPTR1。每个 DPTR 既可以作为一个 16 位寄存器来处理，也可以作为 2 个独立的 8 位寄存器 DPOH/DP1H 和 DPOL/DP1L 来处理。

表 6.1.3-5 数据指针 DP0L

| 82H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|---|---|---|---|---|---|---|
| DP0L | DP0L | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

表 6.1.3-6 数据指针 DP0H

| 83H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|---|---|---|---|---|---|---|
| DPOH | DPOH | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

表 6.1.3-7 数据指针 DP1L

| 84H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|---|---|---|---|---|---|---|
| DP1L | DP1L | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

表 6.1.3-8 数据指针 DP1H

| 85H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|---|---|---|---|---|---|---|
| DP1H | DP1H | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

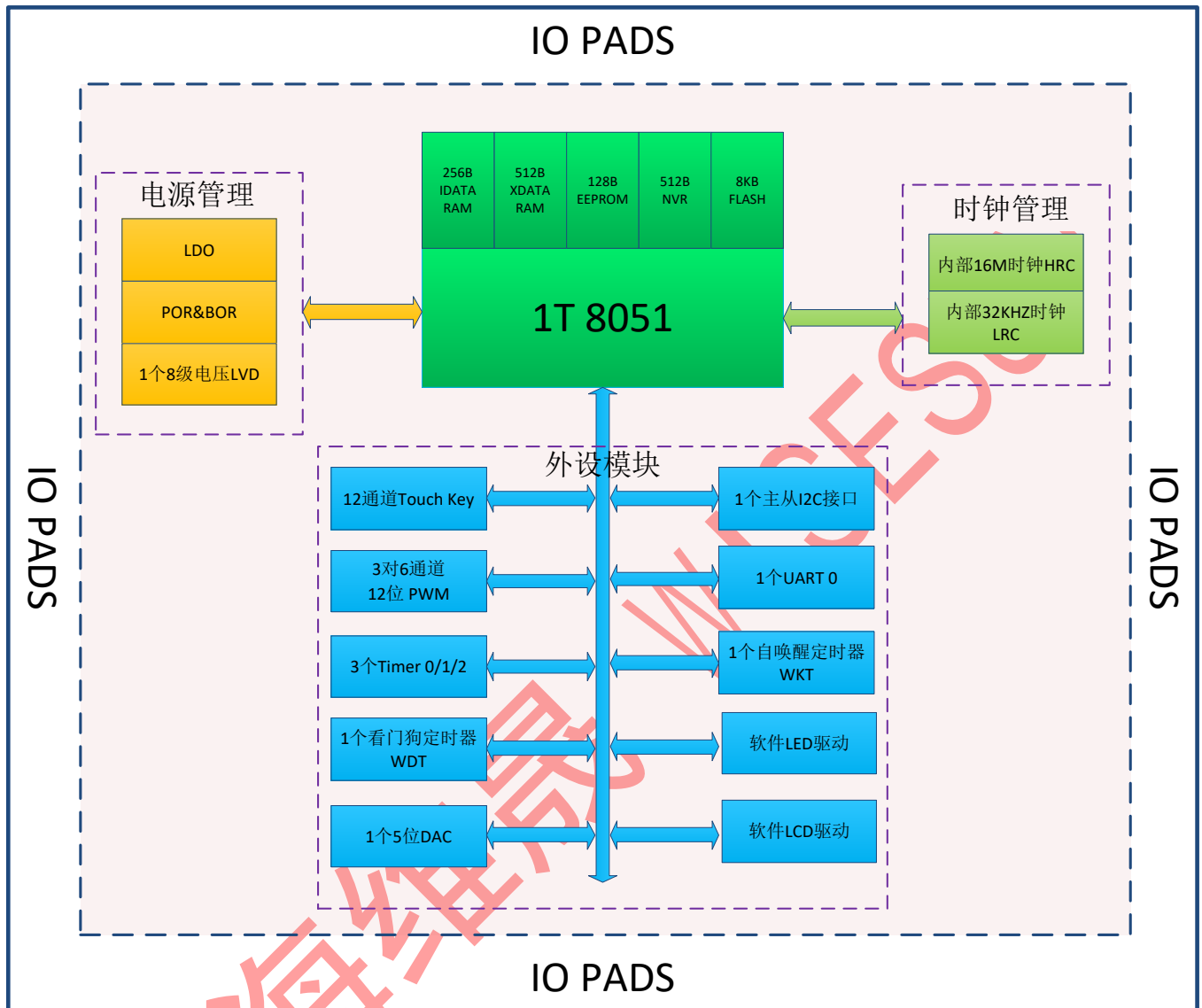
6.1.3.6 状态寄存器 PSW

状态寄存器 PSW 是 CPU 的状态寄存器。在 CPU 做算术运算或者逻辑运算时，对应的 PSW 状态位会发生改变。

表 6.1.3-9 状态寄存器 PSW

| D0H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|---|-----|-----|---|-----|-----|---|
| PSW | CY | AC | F0 | RS | | OV | DPS | P |
| R/W | R/W | R/W | R/W | R/W | | R/W | R | R |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | CY | 进位标志位 0: 算术或逻辑运算中, 没有进位或借位发生 1: 算术或逻辑运算中, 有进位或借位发生 | | | | | | |
| 6 | AC | 辅助进位标志位 0: 算术或逻辑运算中, 没有辅助进位或借位发生 1: 算术或逻辑运算中, 有辅助进位或借位发生 | | | | | | |
| 5 | F0 | F0 标志位 用户自定义标志位 | | | | | | |
| 4-3 | RS | R0-R7 寄存器页选择位 00: 页 0 (映射到 00H-07H) 01: 页 1 (映射到 08H-0FH) 10: 页 2 (映射到 10H-17H) 11: 页 3 (映射到 18H-1FH) | | | | | | |
| 2 | OV | 溢出标志位 0: 没有溢出发生 1: 有溢出发生 | | | | | | |
| 1 | DPS | DPTR 选择寄存器, 0 为选择 DPTR0, 1 为选择 DPTR1 | | | | | | |
| 0 | P | 奇偶校验位 0: 累加器 A 值为 1 的位数为偶数 1: 累加器 A 值为 1 的位数为奇数 | | | | | | |

6.2 芯片结构框图

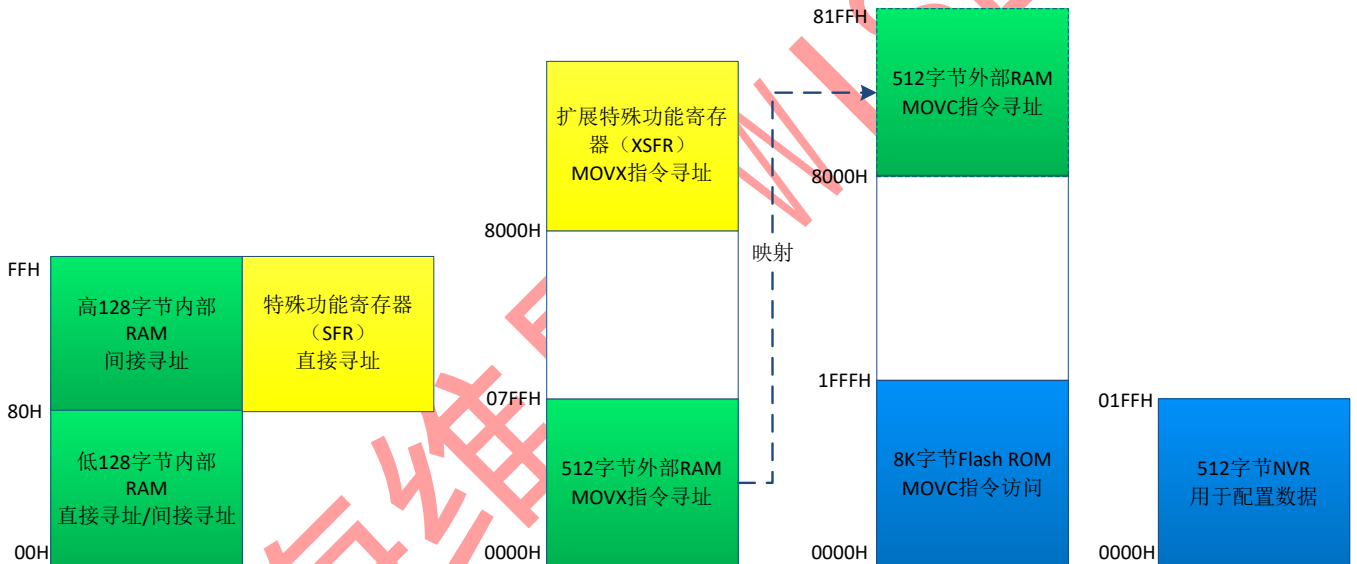


6.3 存储器分布

WS51F7030 系列芯片内含 256B 内部 RAM、512B 外部 RAM、特殊功能寄存器(SFR)、512B NVR 和 8KB Flash ROM，存储器地址分配如下：

- 低 128B 的内部 RAM（地址：00H - 7FH）可直接寻址或间接寻址
- 高 128B 的内部 RAM（地址：80H - FFH）只能间接寻址
- 512B 的外部 RAM（地址：0000H - 01FFFH）可通过 MOVX 指令间接寻址，可映射到程序地址空间 8000H-81FFFH，也可直接作为程序空间 0000H-01FFFH
- 特殊功能寄存器(SFR)（地址：80H - FFH）可直接寻址
- 扩展特殊功能寄存器(XSFR)（从 8000H 开始），通过 MOVX 指令寻址
- 512B NVR 包含 4 个扇区，主要用于系统配置数据
- 8KB Flash ROM（地址：0000H-1FFFH），通过 MOVC 指令可访问

图 6.1.3-1 存储器分布图



6.4 SFR 映像

WS51F7030 系列芯片提供了兼容传统 8051 的 SFR 分布，SFR 和高 128 字节内部 RAM 共用地址 80H - FFH，只能直接寻址，SFR 映射如下表所示。

表 6.1.3-1 特殊功能寄存器 (SFR) 映射表

| 地址 | 可位寻址 | 不可位寻址 | | | | | | |
|------------|-------|--------|--------|--------|--------|--------|--------|--------|
| | 0/8 | 1/9 | 2/A | 3/B | 4/C | 5/D | 6/E | 7/F |
| F8H | - | PWCON | IT0CON | IT1CON | IDLF | - | STPF | - |
| F0H | B | - | - | - | - | - | - | - |
| E8H | - | - | - | - | - | - | - | - |
| E0H | ACC | - | - | - | PCOMS | PDRVS | - | - |
| D8H | EPIE | EPIF | EPOCON | EP1CON | - | - | - | - |
| DOH | PSW | WKTCON | - | - | - | - | - | - |
| C8H | T2CON | T2MOD | T2CL | T2CH | TL2 | TH2 | - | - |
| COH | - | - | - | - | - | - | - | - |
| B8H | IP | LVDCON | - | - | - | - | - | - |
| B0H | P3 | DACON | - | - | - | - | - | - |
| A8H | IE | WDTCON | WDTFLG | - | - | - | - | - |
| A0H | - | I2CCON | I2CADR | I2CFG0 | I2CFG1 | I2CTXD | I2CRXD | I2CFLG |
| 98H | S0CON | S0BUF | S0CFG | - | - | - | - | RSTFLG |
| 90H | P1 | - | - | - | - | - | HRCON | LRCON |
| 88H | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | - | - |
| 80H | - | SP | DP0L | DP0H | DP1L | DP1H | SCCON | PCON |

由于 SFR 地址空间有限，WS51F7030 系列芯片在外部 RAM 地址空间增加了扩展特殊功能寄存器(XSFR)，XSFR 映射如下表所示。

表 6.1.3-2 扩展特殊功能寄存器 (XSFR) 映射表

| 地址 | 0/8 | 1/9 | 2/A | 3/B | 4/C | 5/D | 6/E | 7/F |
|--------------|---------|-------|-------|--------|----------|----------|----------|----------|
| 8000H | - | - | - | - | - | - | - | - |
| 8008H | P10F | P11F | P12F | P13F | P14F | P15F | P16F | P17F |
| 8010H | - | - | - | - | - | - | - | - |
| 8018H | P30F | P31F | P32F | P33F | P34F | P35F | - | - |
| --- | - | - | - | - | - | - | - | - |
| 8040H | PWMEN | - | PWMIE | - | PWMFLG | - | - | - |
| 8048H | - | - | - | - | - | - | - | - |
| 8050H | PWM0CFG | - | - | - | PWM0DUTL | PWM0DUTH | PWM0DIVL | PWM0DIVH |
| 8058H | - | - | - | - | PWM1DUTL | PWM1DUTH | PWM1DIVL | PWM1DIVH |
| 8060H | PWM2CFG | - | - | - | PWM2DUTL | PWM2DUTH | PWM2DIVL | PWM2DIVH |
| 8068H | - | - | - | - | PWM3DUTL | PWM3DUTH | PWM3DIVL | PWM3DIVH |
| 8070H | PWM4CFG | - | - | - | PWM4DUTL | PWM4DUTH | PWM4DIVL | PWM4DIVH |
| 8078H | - | - | - | - | PWM5DUTL | PWM5DUTH | PWM5DIVL | PWM5DIVH |
| --- | - | - | - | - | - | - | - | - |
| 8290H | LSCON | LSCFG | - | LSDTM0 | LSDTM1 | LSPTM0 | LSPTM1 | LSCOM |
| 8298H | LSPE0 | LSPE1 | - | - | - | - | - | - |

| | | | | | | | | |
|--------------|---------|---------|---------|---------|---------|------|--------|--------|
| 82A0H | LSDAT0 | LSDAT1 | - | - | - | - | - | - |
| --- | - | - | - | - | - | - | - | - |
| 8300H | MECON | FSCMD | BTMSK | FSDAT | FSFLG | PADR | FSADRL | FSADRH |
| 8308H | CKSUML | CKSUMH | IDCODE0 | IDCODE1 | IDCODE2 | - | - | - |
| 8400H | CHIPIDL | CHIPIDH | - | - | - | - | - | SPMAX |

前海维晟 WISESUN

7 基本功能

7.1 外部 RAM

512B 的外部 RAM 主要用于数据暂存，但也可映射为程序空间使用，其默认映射地址为 8000H-81FFH。用户可以下载程序到外部 RAM，然后直接执行跳转指令跳到该映射区执行程序。同样，用户也可把 BOOT（详见寄存器 MECON）的值设置为 1，然后执行软复位，此时外部 RAM 映射到地址 0000H-01FFH，复位后程序从外部 RAM 空间开始执行。外部 RAM 映射功能可用来实现 IAP/ISP 等功能。

7.2 Flash 存储器

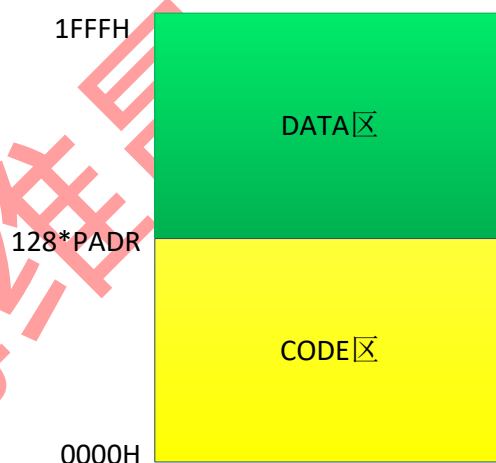
7.2.1 Flash 数据区

7.2.1.1 Flash 数据区逻辑划分

Flash 数据区的逻辑划分见图，主要特点如下：

- Flash 数据区由若干个扇区组成，扇区是进行擦除操作的最小单位，每个扇区大小为 128 字节。
- Flash 区可以通过寄存器 PADR 按功能划分为 CODE 区和 DATA 区，划分单位为 1 个逻辑扇区。
- CODE 区用于存储用户的程序，DATA 区是用于存储一些掉电需要保存的数据。

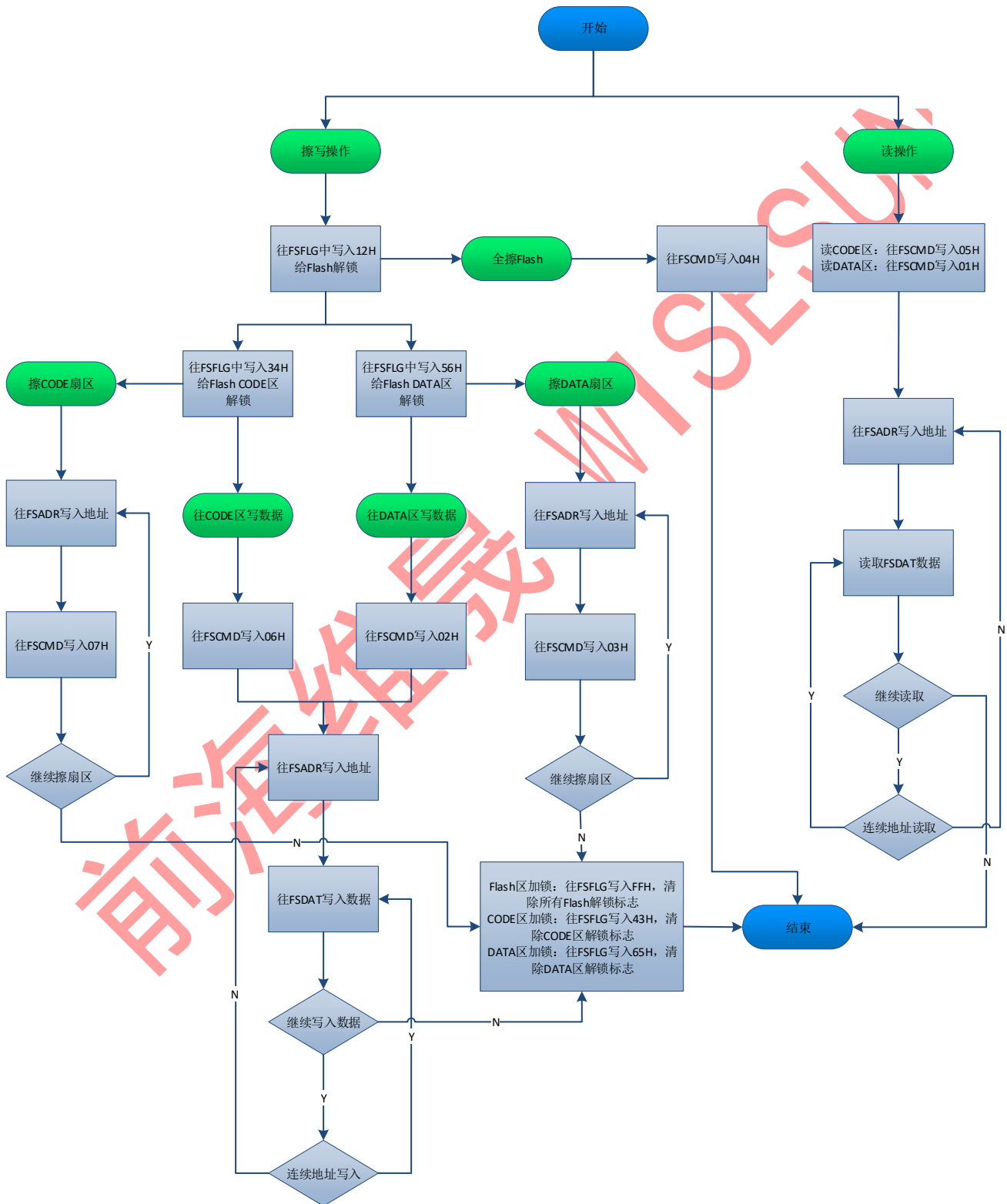
图 7.2.1-1 Flash 数据区结构



7.2.1.2 访问 Flash 数据区

下图所示为访问 Flash 数据区流程图，相关示例程序请参考 WS51F7030 应用手册。需要说明的是，在擦操作时往 FSCMD 写入擦命令或在写操作时往 FSDAT 写入数据时，CPU 会自动暂停，以避免因为中断等原因导致擦写操作中止。

图 7.2.1-2 访问 Flash 数据区流程图



7.2.2 NVR 数据区

7.2.2.1 NVR 数据区功能划分

芯片内置了 512B 的 NVR 数据区，该区共 4 个扇区，每个扇区 128 字节。

- NVR1 保留给用户使用
- NVR2 可用于实现用户 ID 功能
- NVR3 用于保存原厂数据
- NVR4 用于保存 Flash 配置数据

其中 NVR2 和 NVR3 在不使用上述功能时，可直接给用户使用。

表 7.2.2-1 NVR 数据功能表

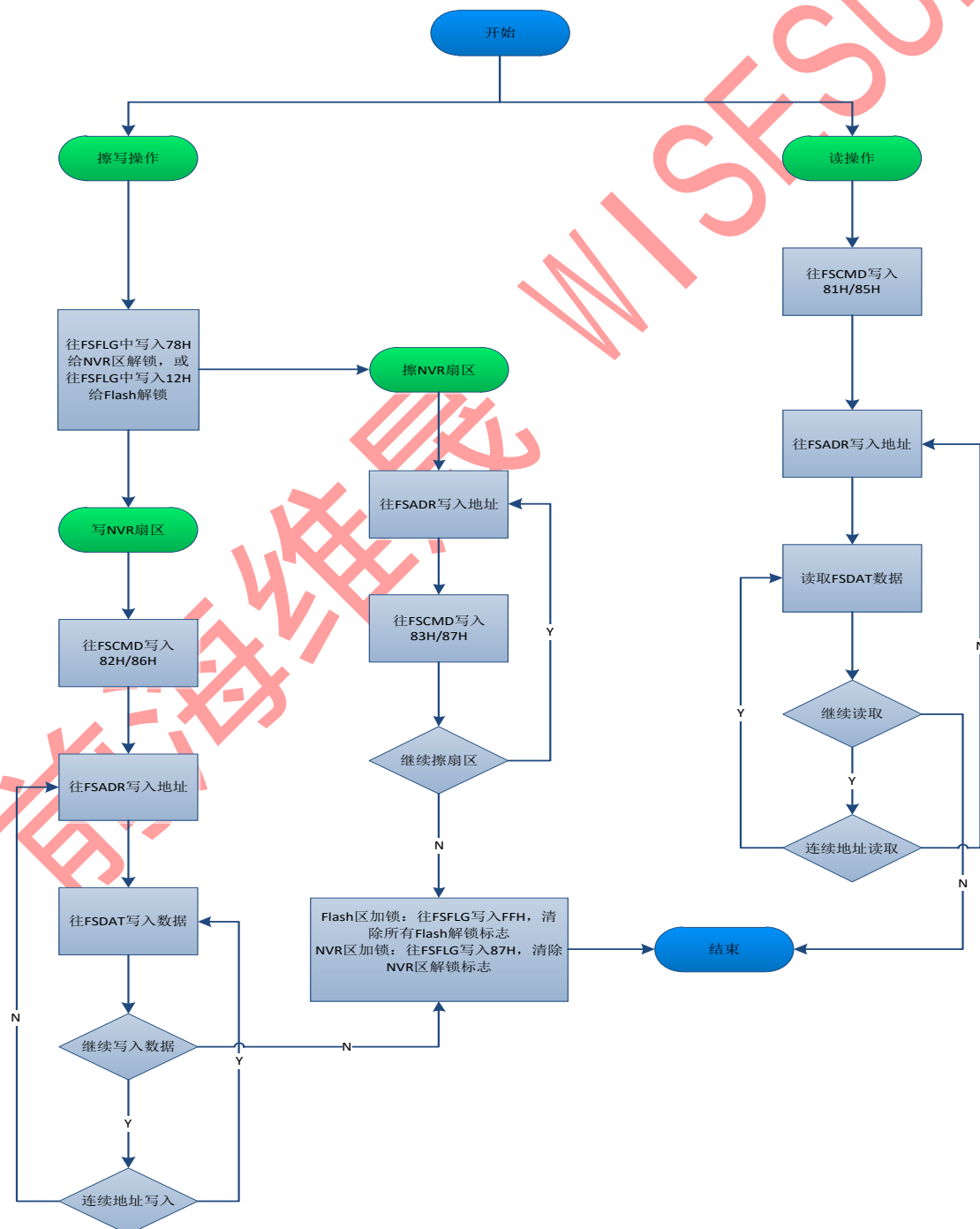
| 扇区 | 地址 | 功能描述 |
|------|-------------|---|
| NVR1 | 0x000-0x6F | 保留给用户使用 |
| | 0x070-0x07f | 芯片滚码区，滚码方式 0x7030+年/月/日+滚码数字 |
| NVR2 | 0x080-0x08F | 该 16 个字节写入 0xaa 后 ID 功能开启 |
| | 0x090 | ID 错误计数字节。当用户写入 IDCODE 寄存器的值与目标值不一致错误次数 4 次就会擦掉一个比特，总共错误次数超过 32 次，寄存器 IDERR 会保持为 1，则不能再使用 ID 功能。开启 ID 功能后出厂时需要烧写为 0xFF。 |
| | 0x091 | 保存用户 ID 目标值，上电时自动加载到内部寄存器。 |
| | 0x092 | |
| | 0x093 | |
| | --- | - |
| | 0x0a0-0x0af | 该 16 个字节写入 0xaa 后开启程序读保护功能 |
| | --- | - |
| | 0x0d0 | 配置 P32 引脚复位后功能。复位后默认情况下为复位功能，设置为 0x5A 后，默认功能为高阻状态。 |
| | --- | - |
| | 0x0e0-0x0ff | 该 32 个字节写入 0xaa，重新上电后 NVR2 不可访问 |
| NVR3 | 0x100 | 该字节为 0xaa 时 bandgap trimming 值有效 |
| | 0x101 | Bandgap trimming 值，上电后自动加载 |
| | 0x102 | 该字节为 0xaa 时 LDO trimming 值有效 |
| | 0x103 | LDO trimming 值，上电后自动加载 |
| | 0x104 | - |
| | 0x105 | - |
| | 0x106 | 该字节为 0xaa 时 LRC 时钟配置值有效 |
| | 0x107 | LRC 时钟配置值，上电后自动加载 |
| | 0x108 | - |
| | 0x109 | 该字节为 0xaa 时 HRC 时钟配置值有效 |
| | 0x10a | HRC 时钟配置值，上电后自动加载 |
| | 0x10b | - |
| | --- | - |
| | 0x150 | CHIPID 的低 8 位 |

| | | |
|------|-------------|---------------------------------|
| | 0x151 | CHIPID 的高 8 位 |
| | --- | - |
| | 0x160-0x17f | 该 32 个字节写入 0xaa，重现上电后 NVR3 不可访问 |
| NVR4 | 0x180-0x1ff | Flash 配置数据，不可访问 |

7.2.2.2 访问 NVR 数据区

下图所示为访问 NVR 数据区流程图，相关示例程序请参考 WS51F7030 应用手册。需要说明的是，在擦操作时往 FSCMD 写入擦命令或在写操作时往 FSDAT 写入数据时，CPU 会自动暂停，以避免因为中断等原因导致擦写操作中止。

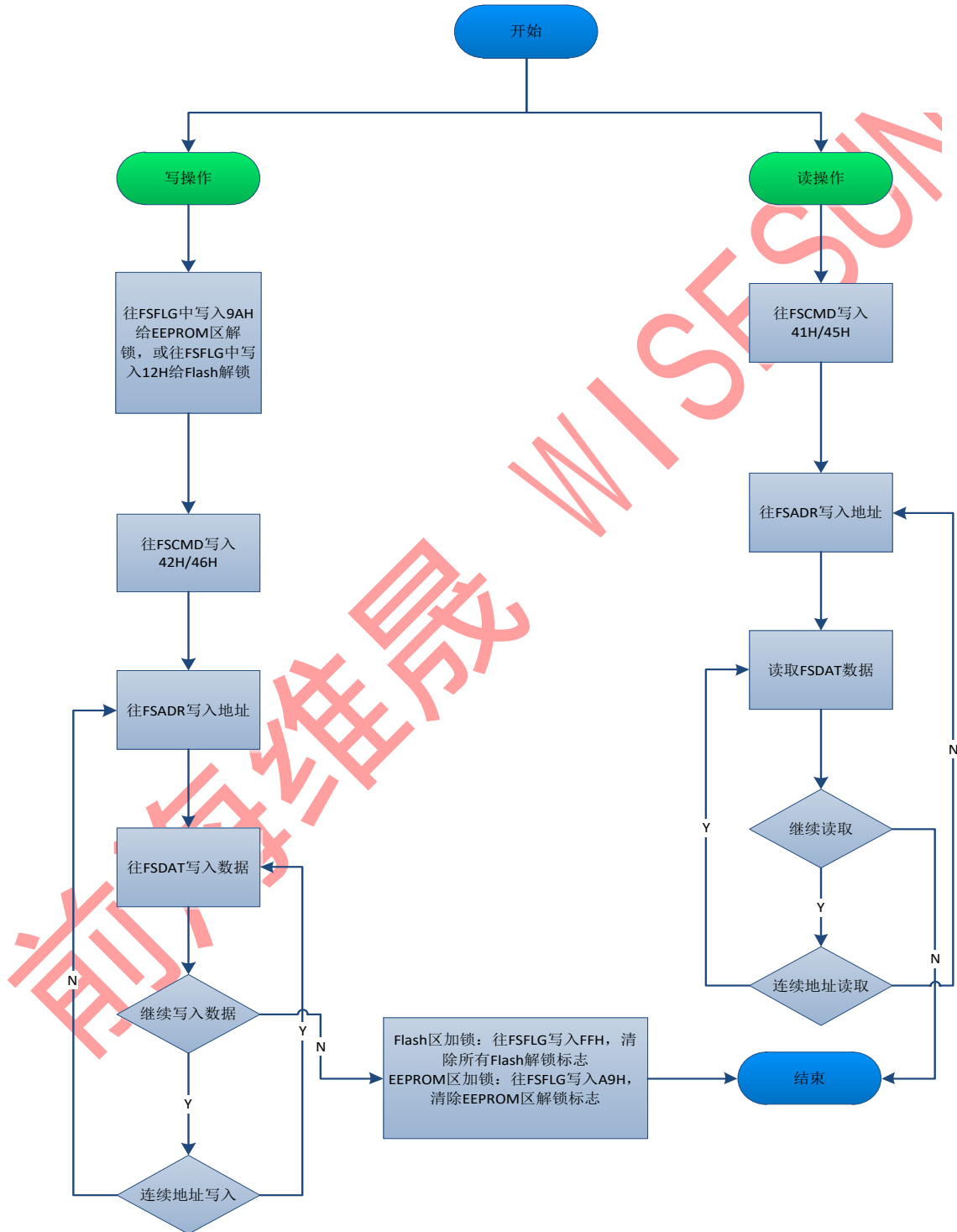
图 7.2.2-2 访问 NVR 数据区流程图



7.2.3 EEPROM 数据区

芯片内置了 128B 的 EEPROM 数据区，读写该区不需要擦可直接改写，主要用于存储需要掉电保留的数据。EEPROM 数据区操作流程如下图所示。

图 7.2.3-1 访问用户数据区流程图



7.2.4 Checksum 校验

为了保证 Flash 烧写正确，芯片内置了 Checksum 功能。设置命令寄存器 CMD(FSCMD 低 4 位)为 4，即可启动 checksum 计算，模块会自动读取地址 FSADR 指定的扇区，并进行累加计算。在计算过程中，CPU 会自动暂停，计算结束后，命令寄存器 CMD 会自动清零。用户通过判断 Checksum 模块计算的结果和实际烧写的程序的 checksum 值是否一致，即可确定烧写是否正确。

7.2.5 寄存器描述

表 7.2.5-1 寄存器 MECON

| 8300H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-------|--|---|---|---|-------|------|---|
| MECON | DPSTB | - | - | - | - | IDERR | BOOT | |
| R/W | R/W | - | - | - | - | R | R/W | |
| 初始值 | 0 | - | - | - | - | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | DPSTB | IDLE/STOP 模式下 Flash 进入睡眠模式控制位 0: IDLE/STOP 模式下，Flash 处于正常工作模式 1: IDLE/STOP 模式下，Flash 进入睡眠模式 备注：如果 DPSTB=1，当芯片进入 IDLE/STOP 模式，Flash 也同时进入睡眠模式，Flash 在睡眠模式的功耗为 50nA，当芯片退出 IDLE/STOP 模式，Flash 也同时退出睡眠模式。 | | | | | | |
| 6-3 | - | - | | | | | | |
| 2 | IDERR | 用户 IDCODE 寄存器配置错误 | | | | | | |
| 1 | - | - | | | | | | |
| 0 | BOOT | 设置软复位后程序启动空间选择位 0: 复位后程序从 Flash 启动运行 1: 软复位后程序从外部 RAM 启动运行 | | | | | | |

表 7.2.5-2 寄存器 FSCMD

| 8301H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|------|---|---|---|-----|---|---|---|
| FSCMD | NVRE | EEPE | - | - | CMD | | | |
| R/W | R/W | R/W | - | - | R/W | | | |
| 初始值 | 0 | 0 | - | - | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | NVRE | NVR 区选择信号 | | | | | | |
| 6 | EEPE | EEPROM 区选择信号 | | | | | | |
| 5-4 | - | - | | | | | | |
| 3-0 | CMD | 命令寄存器 读写主数据区时 0000: 无操作 0001: 读 Flash DATA 区 | | | | | | |

| | | |
|--|--|---|
| | | 0010: 写 Flash DATA 区 0011: 擦除 Flash DATA 区一个扇区 0100: Flash 整片擦除 0101: 读 Flash CODE 区 0110: 写 Flash CODE 区 0111: 擦除 Flash CODE 区一个扇区 1xxx: CKSUM 当前 Flash 扇区 读写 NVR/EEPROM 区时 0000: 无操作 0001/0101: 读数据区 0010/0110: 写数据区 0011/0111: 擦除数据区 1xxx: CKSUM 当前 Flash 扇区 备注: 1. 擦除命令执行后 CMD 自动清零。 2. 读和写命令写入后 CMD 保持不变然后通过读写 FSDAT 完成。 |
|--|--|---|

表 7.2.5-3 寄存器 BTMSK

| 8302H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-------|--------------------------|---|---|---|---|---|---|
| BTMSK | BTMSK | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | BTMSK | 位掩膜寄存器，分别对应操作寄存器的第 7-0 位 | | | | | | |

表 7.2.5-4 寄存器 FSDAT

| 8303H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|------------|-------------|---|---|---|---|---|---|
| FSDAT | FSDAT[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | FSDAT | Flash 数据寄存器 | | | | | | |

表 7.2.5-5 寄存器 FSFLG

| 8304H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|------|---|---|------|------|------|------|------|
| FSFLG | - | - | - | FFLG | CFLG | DFLG | NFLG | EFLG |
| R/W | - | - | - | R | R | R | R | R |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-5 | - | | | | | | | |
| 4 | FFLG | Flash 解锁标志, 往 FSFLG 寄存器写入 0x12 解锁 Flash 区, 写入 0x21 锁定 Flash | | | | | | |

| | | |
|---|------|---|
| | | 区 |
| 3 | CFLG | CODE 区解锁标志, 往 FSFLG 寄存器写入 0x34 解锁 CODE 区, 写入 0x43 锁定 CODE 区 |
| 2 | DFLG | DATA 区解锁标志, 往 FSFLG 寄存器写入 0x56 解锁 DATA 区, 写入 0x65 锁定 DATA 区 |
| 1 | NFLG | NVR 区解锁标志, 往 FSFLG 寄存器写入 0x78 解锁 DATA 区, 写入 0x87 锁定 DATA 区 |
| 0 | EFLG | EEPROM 区解锁标志, 往 FSFLG 寄存器写入 0x9A 解锁 DATA 区, 写入 0xA9 锁定 DATA 区 |

表 7.2.5-6 寄存器 PADR

| | | | | | | | | |
|-------|------|---|---|---|---|---|---|---|
| 8305H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PADR | PADR | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | PADR | CODE 区和 DATA 区划分配置寄存器 程序区和数据区以 128 字节为单位进行划分, 当 PADR >0 时, 程序区的地址空间为: 0 - (PADR ×128 - 1), 数据区的地址空间为: (PADR ×128) - 1FFFH. 备注: 当 PADR =0 时, 整个 Flash 空间都是数据空间。 PADR 的最大值分别为 40H, PADR 的设置值不能超过最大值。 | | | | | | |

表 7.2.5-7 寄存器 FSADR

| | | | | | | | | |
|--------|------------|-------------|---|---|---|---|---|---|
| 8306H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FSADRL | FSADR[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8307H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FSADRH | - | FSADR[14:8] | | | | | | |
| R/W | - | R/W | | | | | | |
| 初始值 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 15 | - | - | | | | | | |
| 14-0 | FSADR | 目标地址指针寄存器 | | | | | | |

表 7.2.5-8 寄存器 CKSUM

| | | | | | | | | |
|--------|------------|---|---|---|---|---|---|---|
| 8308H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CKSUML | CKSUM[7:0] | | | | | | | |
| R/W | R | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | | |
|--------|-------------|------------------|---|---|---|---|---|---|
| 8307H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CKSUMH | CKSUM[15:8] | | | | | | | |
| R/W | R | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 15-0 | CKSUM | Check Sum 累加计数结果 | | | | | | |

表 7.2.5-9 寄存器 IDCODE

| | | | | | | | | |
|---------|---------------|---|---|---|---|---|---|---|
| 8308H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IDCODE0 | IDCODE[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8309H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IDCODE1 | IDCODE[15:8] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 830AH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IDCODE2 | IDCODE[23:16] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 23-0 | IDCODE | 用户 ID 寄存器。开启用户 ID 功能后,如果 ID 寄存器内容和 NVR2 区的 IDCODE 数据不一致, IDERR 寄存器会拉高。在写 IDCODE2 寄存器时芯片会产生一次判断。 | | | | | | |

表 7.2.5-10 寄存器 CHIPID

| | | | | | | | | |
|---------|--------------|--------------------------------------|---|---|---|---|---|---|
| 8400H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CHIPIDL | CHIPID[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8401H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CHIPIDH | CHIPID[15:8] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 15-0 | CHIPID | 芯片 ID 寄存器, 上电时自动读取 NVR2 中的 CHIPID 数据 | | | | | | |

7.3 IO 端口

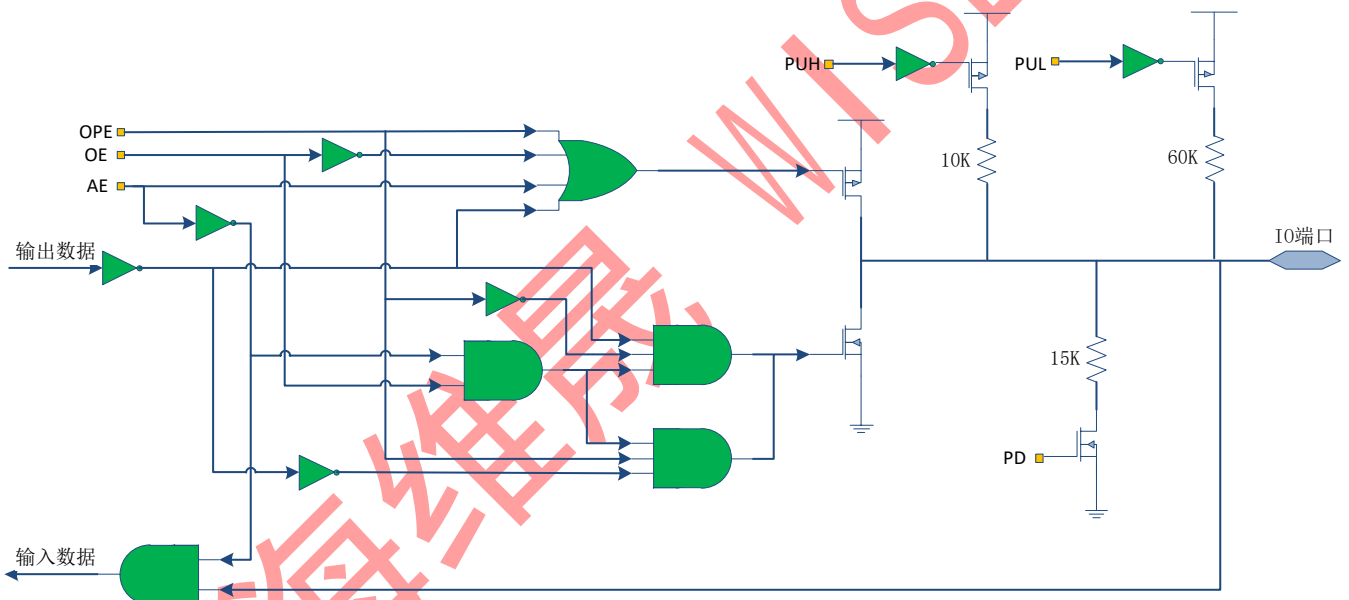
7.3.1 简介

WS51F7030 系列芯片除电源和地端口外，还有 14 个可编程多功能 I/O 端口，其主要特点如下。

- 可配置为高阻模式
- I/O 结构可独立设置强上拉（10KΩ）、弱上拉（60KΩ）和强下拉（15KΩ）
- 输出模式可选开漏输出或推挽输出
- 数据输出锁存支持读-修改-写
- 支持 1.7-5.5V 宽电压范围
- 设为推挽输出时，推电流大于 20mA，灌电流大于 45mA

IO 端口结构图如下图所示。

图 7.3.1-1 I/O 端口结构示意图



备注：

- OPE: IO 端口开漏使能信号
- OE: IO 端口输出使能信号
- AE: IO 端口模拟功能使能信号
- PUH: IO 端口强上拉使能信号
- PUL: IO 端口弱上拉使能信号
- PD: IO 端口下拉使能信号

7.3.2 IO 端口功能复用

芯片的 14 个 IO 端口在功能上被不同的外设功能端口复用。每个端口都分配了一个功能设置寄存器 PnxF（分别对应端口 Pnx，其中 n=1/3，代表 P1/P3，x=0-7，代表 Pn.0-Pn.7），用户可通过寄存器 PnxF 配置端口的主功能和其他选项功能。

表 7.3.2-1 IO 端口复用功能映射表

| 取值 寄存器 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------|----------|------------------|------------------|------------|--------|-------|------------|----|
| P10S | 高阻/LED0 | GPIO 输入 /T2 | GPIO 输出 /COM0 | TK0/LED0 | PWM0 | 高阻 | 高阻 | 高阻 |
| P11S | 高阻/LED1 | GPIO 输入 /T2EX | GPIO 输出 /COM1 | TK1/LED1 | PWM1 | 高阻 | 高阻 | 高阻 |
| P12S | 高阻/LED2 | GPIO 输入 | GPIO 输出 /COM2 | TK2/LED2 | PWM2 | 高阻 | 高阻 | 高阻 |
| P13S | 高阻/LED3 | GPIO 输入 | GPIO 输出 /COM3 | TK3/LED3 | PWM3 | 高阻 | 高阻 | 高阻 |
| P14S | 高阻/LED4 | GPIO 输入 | GPIO 输出 /COM4 | TK4/LED4 | PWM4 | 高阻 | 高阻 | 高阻 |
| P15S | 高阻/LED5 | GPIO 输入 | GPIO 输出 | TK5/LED5 | PWM5 | 高阻 | 高阻 | 高阻 |
| P16S | 高阻/LED6 | GPIO 输入 | GPIO 输出 | TK6/LED6 | 高阻 | 高阻 | 高阻 | 高阻 |
| P17S | 高阻/LED7 | GPIO 输入 | GPIO 输出 | TK7/LED7 | [PWM0] | 高阻 | 高阻 | 高阻 |
| P30S | 高阻/LED8 | GPIO 输入 | GPIO 输出 | TK8/LED8 | [PWM5] | [SCL] | UART0_RX | 高阻 |
| P31S | 高阻/LED9 | GPIO 输入 | GPIO 输出 | TK9/LED9 | [PWM4] | [SDA] | UART0_TX | 高阻 |
| P32S | 高阻/LED10 | GPIO 输入 | GPIO 输出 | TK10/LED10 | [PWM3] | RST | DAVF | 高阻 |
| P33S | 高阻/LED11 | GPIO 输入 | GPIO 输出 | TK11/LED11 | [PWM1] | SCL | [UART0_RX] | 高阻 |
| P34S | 高阻 | GPIO 输入 /T0 | GPIO 输出 | DAC | [PWM2] | SDA | [UART0_TX] | 高阻 |
| P35S | 高阻 | GPIO 输入 /T1 | GPIO 输出 | CMOD | 高阻 | 高阻 | 高阻 | 高阻 |

备注:

- 带[]的通道为备选通道, 具有较低优先级。I2C 端口如果同时打开首选和备选通道, 信号会从首选通道输出, 备选通道此时为高阻。而 PWM 如果同时打开首选和备选通道, 两个通道会同时输出该信号。
- GPIO 输入和 T0/T1 共用引脚, 其中 T0/T1 有更高优先级。
- 设置寄存器 LSPE0/LSPE1 开启 LED 端口功能, 当 IO 端口为高阻或 TK 功能时, LED 输出有效, 并与其他功能共用 IO 端口, 详细见 [8.10 软件 LED 驱动 \(SLED\)](#) 章节描述。

7.3.3 寄存器描述

表 7.3.3-1 IO 端口寄存器 Pn

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--|-----|---------------------------------------|-----|-----|-----|-----|-----|-----|
| Pn | Pn7 | Pn6 | Pn5 | Pn4 | Pn3 | Pn2 | Pn1 | Pn0 |
| 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 |
| n=1/3, 代表 P1/P3 寄存器地址 P1: 90H P3: B0H | | | | | | | | |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | Pn | 端口 Pn 的数据寄存器, IO 端口功能设置为 GPIO 输出时输出该值 | | | | | | |

表 7.3.3-2 IO 端口功能寄存器 PnxF

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--|--------|---|--------|---|--------|------|---|---|
| PnxF | PnxPUS | | PnxOPR | - | PnxDRS | PnxS | | |
| R/W | R/W | | R/W | - | R/W - | R/W | | |
| n=1/3, 代表 P1/P3 x=0-7, 代表 Pn0/Pn1/ Pn2/Pn3/ Pn4/Pn5/ Pn6/Pn7 寄存器地址 P10F-P17F: 8008H-800FH P30F-P35H: 8018H-801DH 其中 P32F 初始值为 0x85 P33F 初始值为 0xA5 P34F 初始值为 0xA5 其他寄存器初始值为 0x00 | | | | | | | | |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-6 | PnxPUS | 上下拉电阻使能控制位 0: 没有打开上下拉电阻 1: 60 KΩ 上拉电阻打开 2: 10 KΩ 上拉电阻打开 3: 15 KΩ 下拉电阻打开 | | | | | | |
| 5 | PnxOPR | 开漏使能控制位, 端口设为 GPIO 输出时才有效 0: 开漏关闭 1: 开漏打开 | | | | | | |
| 4-3 | - | - | | | | | | |
| 2-0 | PnxS | IO 端口复用功能选择寄存器, 见表 7.3.2-1 描述 | | | | | | |

7.4 定时器 0/1/2

7.4.1 定时器 0

定时器或计数器功能通过控制位 CT0 来选择，CT0=0 选择为定时器，CT0=1 选择为计数器。作为定时器时，时钟是系统时钟的 12 分频。作为计数器时，时钟是 T0 端口的输入。由于检测 T0 输入边沿变化需要 2 个时钟周期，所以作为计数器时最大的输入波特率是内部系统时钟频率的 1/2。T0 输入信号在占空比上没有限制，为了完全识别 0 或 1 的状态，信号至少需要保持 1 个内部系统时钟周期时间。定时器 0 有 4 个工作模式，通过模式选择位 TOM0 和 TOM1 来选择。

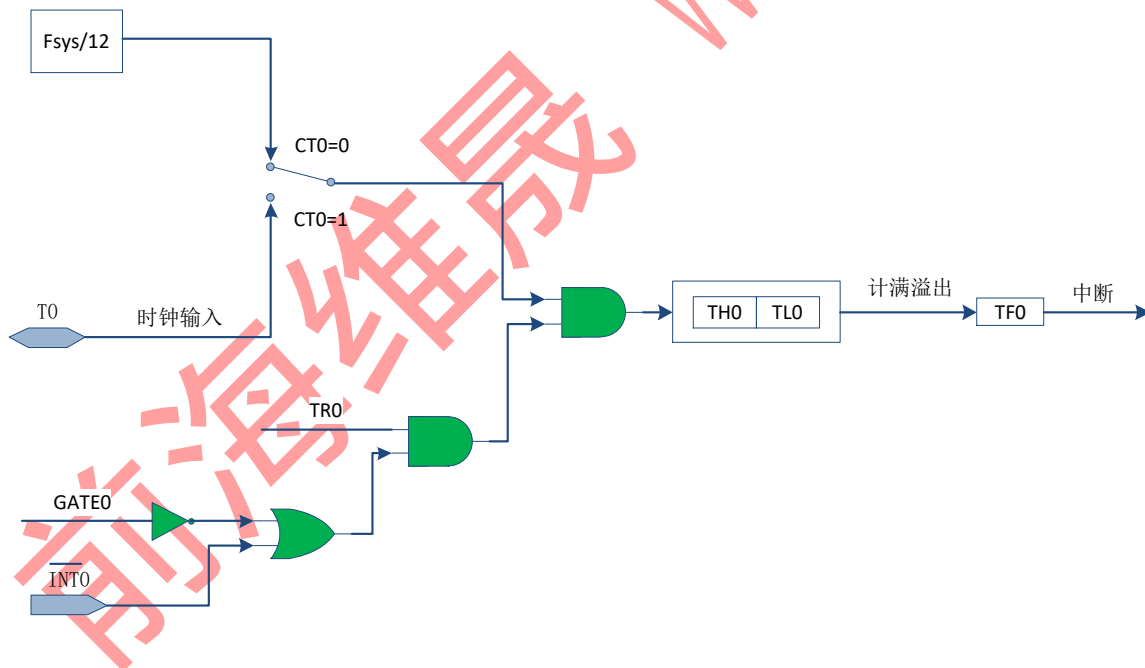
7.4.1.1 模式 0

在此模式下，定时器 0 作为 13 位定时器/计数器，寄存器 TH0 存放 13 位定时器/计数器的高 8 位，寄存器 TLO 的低 5 位存放定时器/计数器低 5 位，TLO 的高 3 位是无效的，在读取时应被忽略。当定时器 0 溢出，中断标志位 TFO 会被置 1。中断响应后，TFO 位会自动清 0。当 GATE0=0 时，定时器/计数器由 TR0 位使能计数，当 GATE0=1 时，定时器/计数器由端口 INTO 控制使能，INTO 为高电平时计数，INTO 为低电平时停止计数。

7.4.1.2 模式 1

此模式下，定时器 0 作为 16 位定时器/计数器，除此之外，功能与模式 0 完全相同。

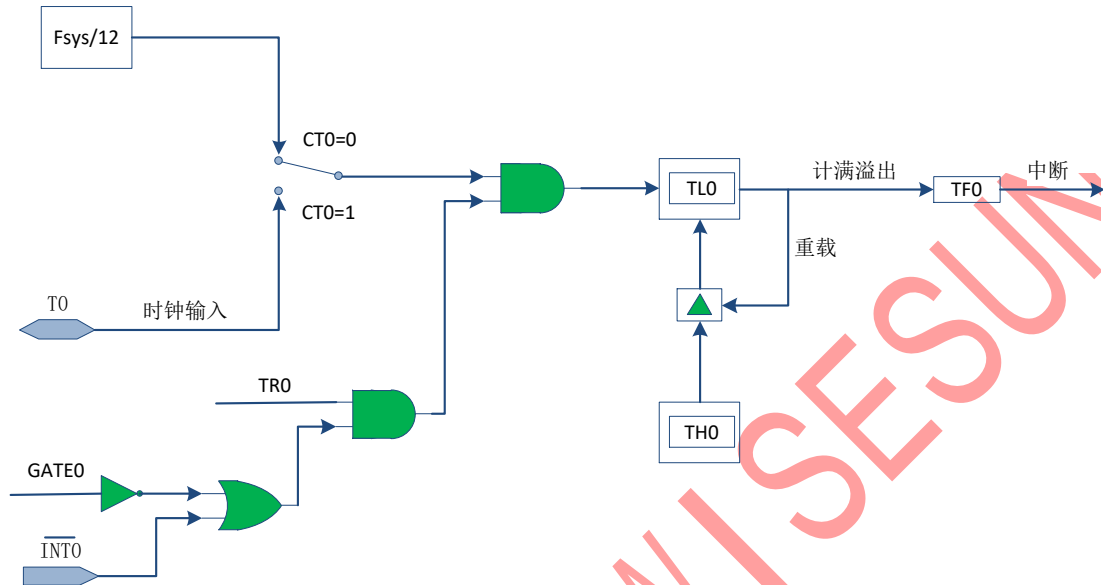
图 7.4.1-1 定时器 0 的模式 0 与 1



7.4.1.3 模式 2

在此模式中，定时器 0 作为 8 位自动重载定时器/计数器，只有 TLO 自动累加。当 TLO 计数溢出时，不但产生中断标志 TF0，而且从 TH0 中自动装载计数初始值到 TLO。其他设置方法和模式 0、1 相同。

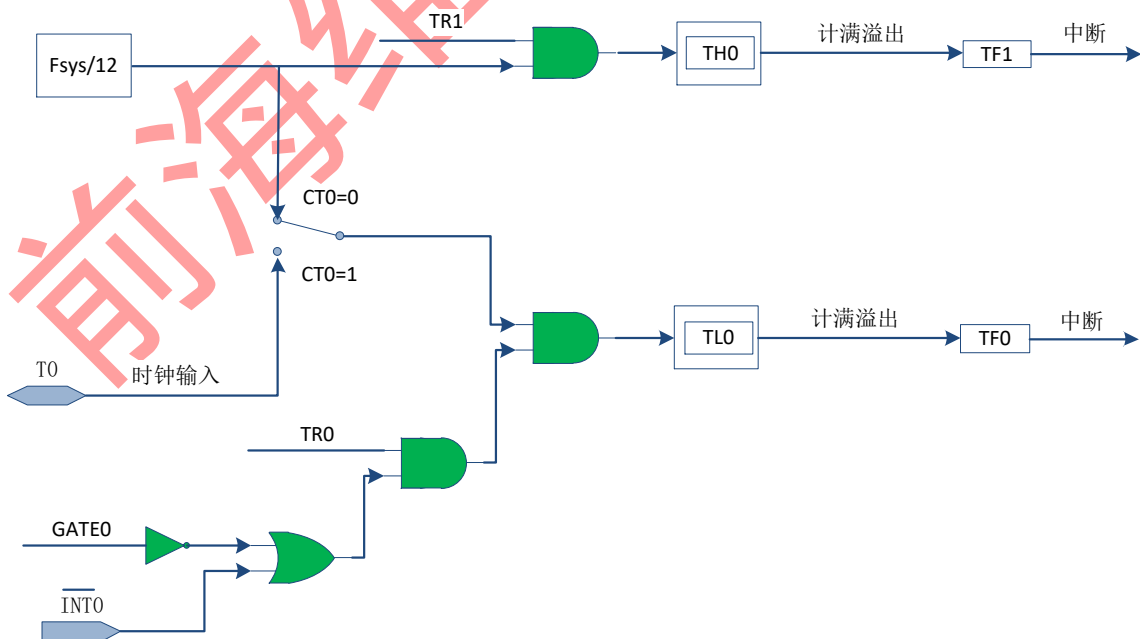
图 7.4.1-2 定时器 0 的模式 2



7.4.1.4 模式 3

在此模式中，TLO 和 TH0 作为两个独立的 8 位定时器/计数器。TLO 可以作为定时器或计数器，而 TH0 只能作为定时器。其中 TLO 占用定时器 0 的控制位 CT0、GATE0、TR0、TF0、INT0，而 TH0 占用定时器 1 的控制位 TR1、TF1。其他控制方法和模式 0、1 相同。当定时器 0 工作于模式 3 时，定时器 1 和 TH0 共用控制位 TR1，但定时器 1 由于 TF1 已被 TH0 占用，所以只能工作于不需要产生中断的场合，例如作为 UART0 的波特率产生器。

图 7.4.1-3 定时器 0 的模式 3



7.4.2 定时器 1

定时器或计数器功能通过 CT1 来选择，CT1=0 选择为定时器，CT1=1 选择为计数器。作为定时器时，时钟是系统时钟的 12 分频。作为计数器时，时钟是 T1 的输入时钟。由于检测 T1 输入边沿变化需要 2 个时钟周期，所以作为计数器时最大的输入波特率是内部系统时钟频率的 1/2。T1 输入信号在占空比上没有限制，然而为了完全识别 0 或 1 的状态，信号至少需要保持 1 个内部系统时钟周期时间。定时器 1 有 4 个工作模式，通过 T1M0 和 T1M1 来选择。

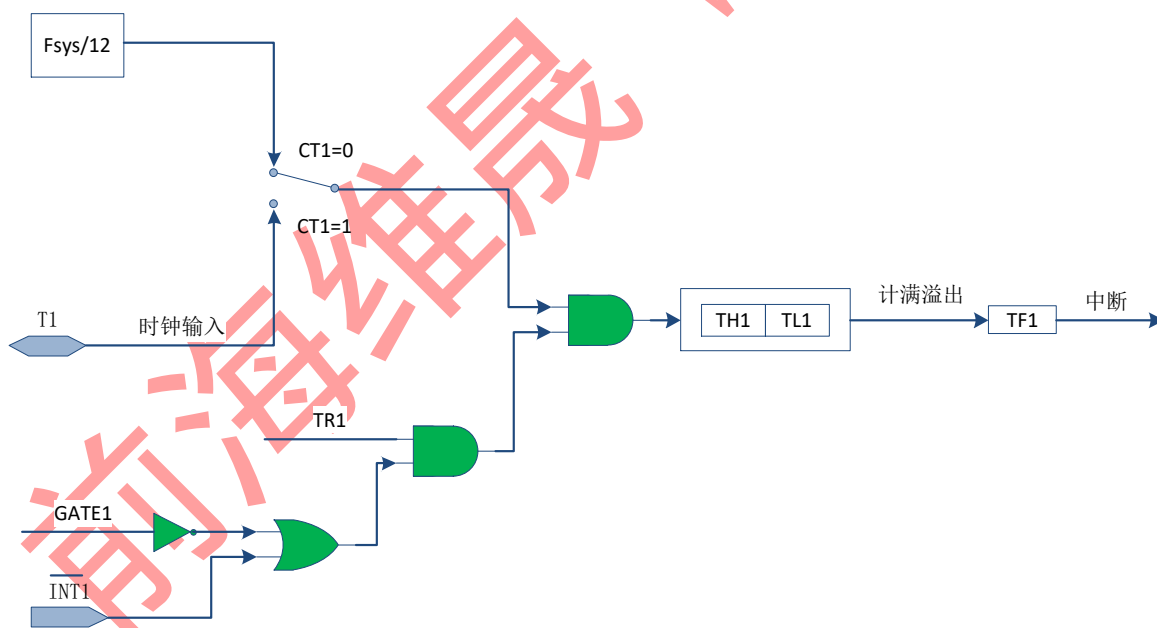
7.4.2.1 模式 0

在此模式下，定时器 1 作为 13 位定时器/计数器，TH1 存放 13 位定时器/计数器的高 8 位，TL1[4:0] 存放低 5 位，而 TL1[7:5] 是无效的，在读取时应被忽略。当定时器 1 溢出，中断标志位 TF1 会被置 1。中断响应后，TF1 位会自动清 0。当 GATE1=0 时，定时器/计数器由 TR1 位使能计数，当 GATE1=1 时，定时器/计数器由端口 INT1 控制使能，INT1 为高电平时计数，INT1 为低电平则停止计数。

7.4.2.2 模式 1

在此模式下，定时器 1 作为 16 位定时器/计数器，TH1 存放 16 位定时器/计数器的高 8 位，TL1 存放低 8 位。当定时器 1 溢出，中断标志位 TF1 会被置 1。中断被响应后，TF1 位会自动清 0。当 GATE1=0 时，定时器/计数器由 TR1 位使能计数，当 GATE1=1 时，定时器/计数器由端口 INT1 控制使能，INT1 为高电平时计数，INT1 为低电平则停止计数。

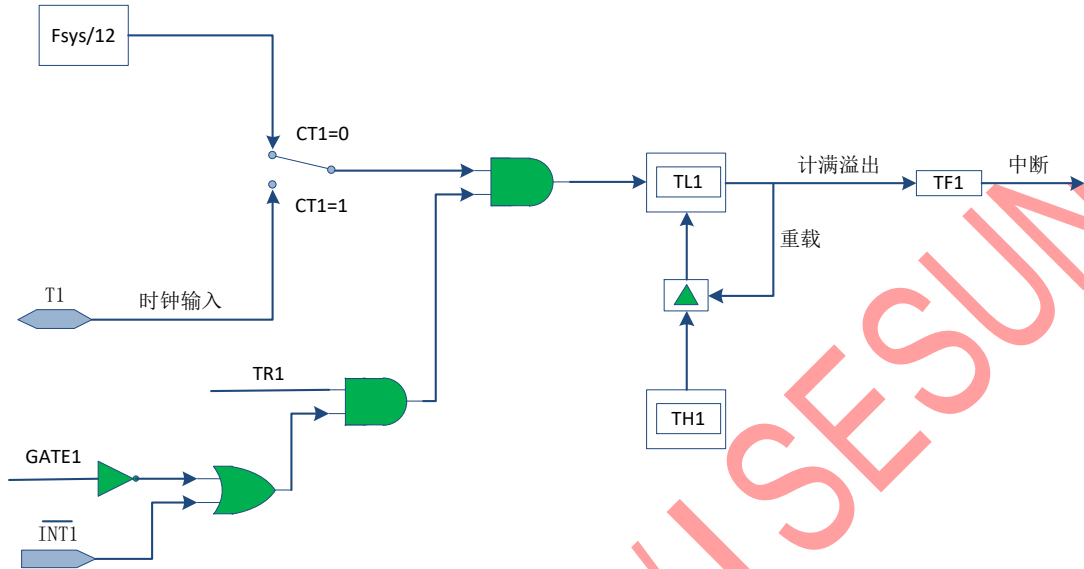
图 7.4.2-1 定时器 1 的模式 0 和 1



7.4.2.3 模式 2

在此模式中，定时器 1 作为 8 位自动重载定时器/计数器，只有 TL1 自动累加。当 TL1 计数溢出时，不但产生中断标志 TF1，而且从 TH1 中自动装载计数初始值到 TL1。其他设置方法和模式 0、1 相同。

图 7.4.2-2 定时器 1 的模式 2



7.4.2.4 模式 3

此模式下，TH1、TL1 会被锁住，等效于 TR1=0。

7.4.3 定时器 2

定时器 2 是一个 16 位 (TH2、TL2) 的定时器或计数器，其定时或计数功能通过 CT2 来选择，CT2=0 选择为定时功能，CT2=1 选择为计数功能。作为计数器时，其输入时钟为 T2 引脚。

定时器 2 有 4 种工作模式：重载模式、抓取模式、波特率产生器模式和时钟输出模式。

T2 引脚为定时器 2 的时钟输入输出引脚，在时钟输出模式，该引脚输出可编程时钟，而在其他模式，T2 为时钟输入引脚。T2EX 引脚为定时器 2 的门控信号，在使用 T2EX 边沿时，可通过 T2XPS 选择是上升沿还是下降沿。

定时器 2 的计数溢出标志为 TF2，T2EX 的边沿触发标志为 EXF2，而在波特率产生器模式和时钟输出模式，定时器不会产生任何标志。

7.4.3.1 重载模式

重载模式有两种工作方式：模式 0 和模式 1。

当 DCEN 为 0 时，设置 CPRL2 为 0 则进入重载模式 0，如图 7.4.3-1 所示。此时如果 EXEN2 也为 1，则 T2EX 边沿和计数溢出时将触发重载，否则只有计数溢出才能触发重载。

当 DCEN 为 1 时则进入重载模式 1，如图 7.4.3-2 所示。此时 T2EX 为低时定时器递减计数，T2EX 为高时定时器递增计数。EXF2 标志不再产生中断，而是作为计数器得第 17 位翻转。

图 7.4.3-1 定时器 2 的重载模式 0

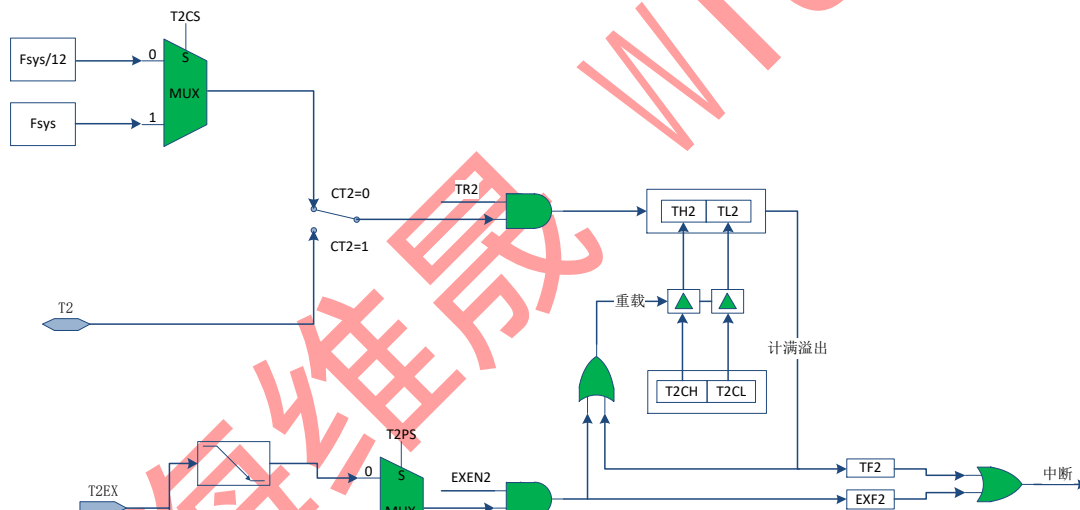
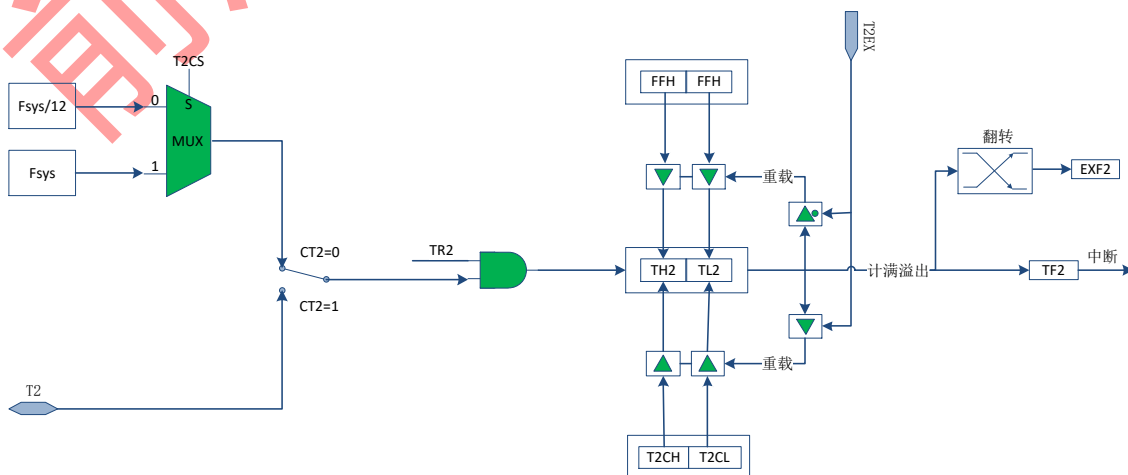


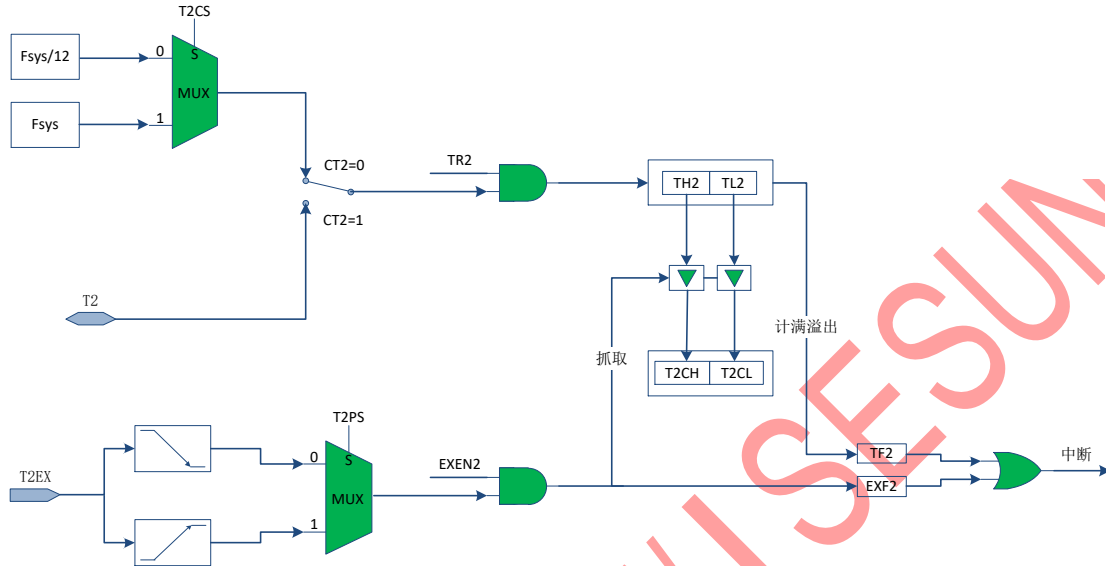
图 7.4.3-2 定时器 2 的重载模式 1



7.4.3.2 抓取模式

设置 CPRL2 为 1 则进入抓取模式。如果设置 T2CM 为 0，且 EXEN2 为 1，则由 T2PS 选择 T2EX 上沿或下沿触发抓取操作。如果设置 T2CM 为 1，则软件写入寄存器 T2CL 将触发一次抓取操作。

图 7.4.3-3 定时器 2 的抓取模式



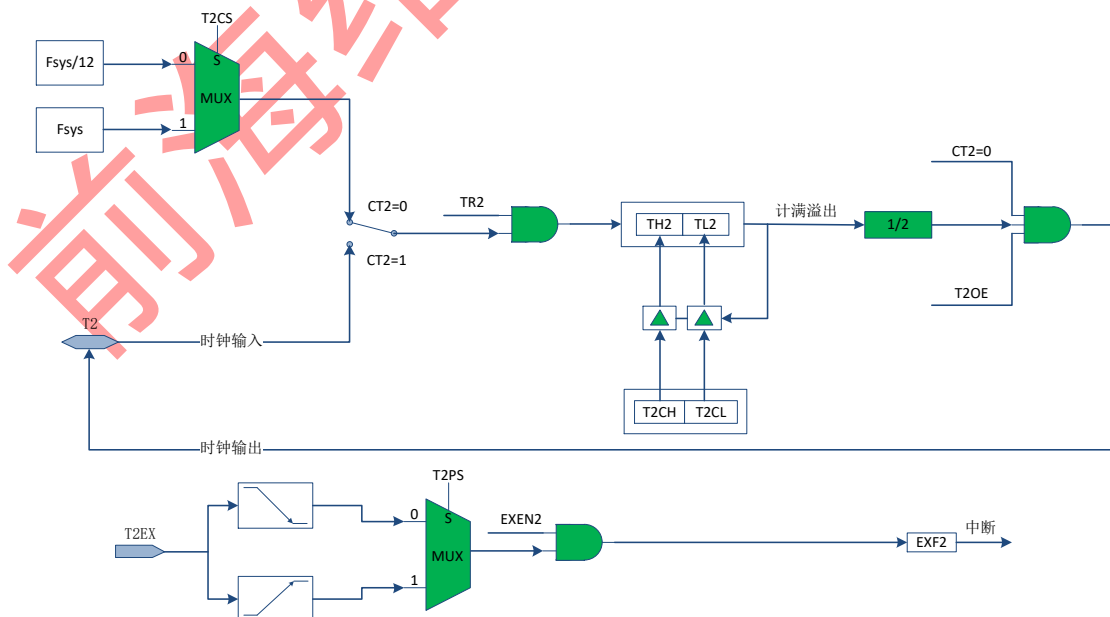
7.4.3.3 波特率产生器模式

设置 RCLK 或 TCLK 为 1 则进入波特率产生器模式，此时定时器自动进入计数溢出重载模式，相关波特率产生参考 8.2 UART0 章节。

7.4.3.4 时钟输出模式

设置 T2OE 为 1 则进入时钟输出模式，此时定时器自动进入计数溢出重载模式，T2 引脚输出频率为 2 倍溢出周期的时钟。此模式不能使用 T2 作为时钟输入。

图 7.4.3-4 定时器 2 的时钟输出模式



7.4.4 寄存器描述
表 7.4.4-1 寄存器 TCON

| 88H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-----|--|-----|-----|-----|-----|-----|-----|
| TCON | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
| 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 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | TF1 | 定时器 0 模式 3 的 TH0 溢出/定时器 1 溢出标志位，中断响应后自动清 0. | | | | | | |
| 6 | TR1 | 定时器 1 运行控制位，1 有效 | | | | | | |
| 5 | TF0 | 定时器 0 溢出标志位，中断响应后自动清 0. | | | | | | |
| 4 | TR0 | 定时器 0 运行控制位，1 有效 | | | | | | |
| 3 | IE1 | 外部中断 1 使能位，1 有效 | | | | | | |
| 2 | IT1 | 外部中断 1 触发类型控制位 0: 外部中断 1 在输入管脚低电平时触发 1: 外部中断 1 在输入管脚下降沿时触发 | | | | | | |
| 1 | IE0 | 外部中断 0 使能位，1 有效 | | | | | | |
| 0 | IT0 | 外部中断 0 触发类型控制位 0: 外部中断 0 在输入管脚低电平时触发 1: 外部中断 0 在输入管脚下降沿时触发 | | | | | | |

表 7.4.4-2 寄存器 TMOD

| 89H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-------|--|------|------|-------|-----|------|------|
| TMOD | GATE1 | CT1 | T1M1 | T1M0 | GATE0 | CT0 | T0M1 | T0M0 |
| 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 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | GATE1 | 定时器 1 门控控制位，1 有效。有效时定时器 1 由 INT1 控制开关 | | | | | | |
| 6 | CT1 | 定时器 1 计数器/定时器选择位 0: 定时器，时钟为系统时钟 12 分频 1: 计数器，时钟为 T1 输入时钟 | | | | | | |
| 5 | T1M1 | [T1M1,T1M0]为定时器 1 模式选择位 00: 模式 0, TL1 和 TH1 组成 13 位定时器/计数器 01: 模式 1, TL1 和 TH1 组成 16 位定时器/计数器 10: 模式 2, TL1 作为 8 位定时器/计数器, TH1 作为自动重载寄存器 11: 模式 3, 此模式会锁住 TH1/TL1, 等效于 TR1=0 | | | | | | |
| 4 | T1M0 | | | | | | | |
| 3 | GATE0 | 定时器 0 门控控制位，1 有效。有效时定时器 0 由 INTO 控制开关 | | | | | | |
| 2 | CT0 | 定时器 0 计数器/定时器选择位 0: 定时器，时钟为系统时钟 12 分频 1: 计数器，时钟为 T0 输入时钟 | | | | | | |

| | | |
|---|------|---|
| 1 | TOM1 | [TOM1,TOM0]为定时器 0 模式选择位 00: 模式 0, TLO 和 TH0 组成 13 位定时器/计数器 01: 模式 1, TLO 和 TH0 组成 16 位定时器/计数器 |
| 0 | TOM0 | 10: 模式 2, TLO 作为 8 位定时器/计数器, TH0 作为自动重载寄存器 11: 模式 3, TLO 和 TH0 作为两个完全独立的 8 位定时器/计数器 |

表 7.4.4-3 寄存器 TLO

| | | | | | | | | |
|-----|-----|----------------------------------|---|---|---|---|---|---|
| 8AH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TLO | TLO | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | TLO | 定时器 0 模式 0/1 计数值的低字节, 模式 2/3 计数值 | | | | | | |

表 7.4.4-4 寄存器 TH0

| | | | | | | | | |
|-----|-----|--|---|---|---|---|---|---|
| 8CH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TH0 | TH0 | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | TH0 | 定时器 0 模式 0/1 计数值的高字节, 模式 2 重载值, 模式 3 计数值 | | | | | | |

表 7.4.4-5 寄存器 TL1

| | | | | | | | | |
|-----|-----|----------------------------------|---|---|---|---|---|---|
| 8BH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TL1 | TL1 | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | TL1 | 定时器 1 模式 0/1 计数值的低字节, 模式 2/3 计数值 | | | | | | |

表 7.4.4-6 寄存器 TH1

| | | | | | | | | |
|-----|-----|--|---|---|---|---|---|---|
| 8DH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TH1 | TH1 | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | TH1 | 定时器 1 模式 0/1 计数值的高字节, 模式 2 重载值, 模式 3 计数值 | | | | | | |

表 7.4.4-7 寄存器 T2CON

| | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|
| C8H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|

| T2CON | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | CT2 | CPRL2 |
|-------|-------|---|------|------|-------|-----|-----|-------|
| 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 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | TF2 | 定时器 2 溢出标志位。当 RCLK 或 TCLK 为 1，TF2 不会被置 1 | | | | | | |
| 6 | EXF2 | T2EX 和软件抓取触发标志。 在 EXEN2=1 时，T2EX 边沿将触发 EXF2 并产生中断。当 DCEN=1 时，EXF2 标志不产生中断，此时 EXF2 作为计数器得第 17 位。 在 T2CM=1 时，软件写入 T2CL 寄存器将触发一次抓取，此时将置位 EXF2 标志 | | | | | | |
| 5 | RCLK | UART0 在模式 1 和 3 时使用定时器 1/2 作为接收波特率时钟 0: 定时器 1 1: 定时器 2 | | | | | | |
| 4 | TCLK | UART0 在模式 1 和 3 时使用定时器 1/2 作为发送波特率时钟 0: 定时器 1 1: 定时器 2 | | | | | | |
| 3 | EXEN2 | T2EX 使能信号，在定时器没有作为 UART0 波特率时钟时 T2EX 有效 | | | | | | |
| 2 | TR2 | 定时器 2 启动信号，1 有效 | | | | | | |
| 1 | CT2 | 定时功能或计数功能选择信号 0: 定时功能，使用系统内部时钟计数 1: 计数功能，使用 T2 输入时钟计数 | | | | | | |
| 0 | CPRL2 | 抓取或重载功能选择信号。当 RCLK 或 TCLK 为 1 时，定时器强制进入 Timer 2 重载模式 0: 重载功能 1: 抓取功能 | | | | | | |

表 7.4.4-8 寄存器 T2MOD

| C9H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-------|--|---|------|-------|------|------|------|
| T2MOD | T2IE | - | - | T2CM | T2XPS | T2PS | T2OE | DCEN |
| R/W | R/W | - | - | R/W | R/W | R/W | R/W | R/W |
| 初始值 | 0 | - | - | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | T2IE | 定时器 2 中断使能 | | | | | | |
| 6-5 | - | - | | | | | | |
| 4 | T2CM | 定时器 2 抓取模式选择 0: T2EX 触发抓取 1: 软件写 T2CL 抓取 | | | | | | |
| 3 | T2XPS | T2EX 有效边沿选择 0: 下降沿 1: 上升沿 | | | | | | |
| 2 | T2PS | T2 边沿选择 0: 下降沿 | | | | | | |

| | | |
|---|------|-------------------|
| | | 1: 上升沿 |
| 1 | T2OE | 定时器 2 时钟输出模式使能 |
| 0 | DCEN | 重载模式 1 的使能位, 1 有效 |

表 7.4.4-9 寄存器 T2CL

| CAH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|---|---|---|---|---|---|---|
| T2CL | T2CL | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | T2CL | 在重载模式, T2CL 是重载值的低字节 在比较模式, T2CL 是比较值的低字节 在抓取模式, T2CL 保存捕获值的低字节 | | | | | | |

表 7.4.4-10 寄存器 T2CH

| CBH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|---|---|---|---|---|---|---|
| T2CH | T2CH | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | T2CH | 在重载模式, T2CH 是重载值的高字节 在比较模式, T2CH 是比较值的高字节 在捕获模式, T2CH 保存捕获值的高字节 | | | | | | |

表 7.4.4-11 寄存器 TL2

| CCH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|---------------|---|---|---|---|---|---|
| TL2 | TL2 | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | TL2 | 定时器 2 计数值的低字节 | | | | | | |

表 7.4.4-12 寄存器 TH2

| CDH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|---------------|---|---|---|---|---|---|
| TH2 | TH2 | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | TH2 | 定时器 2 计数值的高字节 | | | | | | |

7.5 中断系统

7.5.1 简介

WS51F7030 系列芯片有一个增强的中断控制系统，共有 7 个中断入口，如下表所示。

每个中断源有独立的中断向量地址、中断使能位和中断标志。CPU 在响应中断后，进入该中断对应的中断服务程序，接到 RETI 指令后将返回中断前状态。

每个中断源有两级中断优先级，如果同时有多个有效中断产生中断请求，CPU 将根据设置的中断优先级依次响应；如果优先级相同，则根据它们的自然优先级（中断向量地址从低到高）依次响应。

寄存器 IE 和 EXIE 为中断使能寄存器，寄存器 IP 和 EXIP 为中断优先级寄存器，详见 7.5.3 寄存器描述。

表 7.5.1-1 中断向量表

| 中断 | 中断源 | 向量地址 | 中断号 |
|-------|--|------|-----|
| INT0 | External Interrupt 0 | 03H | 0 |
| TF0 | Timer 0 | 0BH | 1 |
| INT1 | External Interrupt 1 | 13H | 2 |
| TF1 | Timer 1 | 1BH | 3 |
| RI+TI | UART 0 | 23H | 4 |
| INT2 | I2C/Timer2/PWM/LED/External Interrupt 2 | 2BH | 5 |
| INT3 | Touch Key/WDT/LVD/WKT/External Interrupt 3 | 33H | 6 |

7.5.2 外部中断

除了标准 8051 的 INT0 和 INT1 以外，系统还扩展了 2 个中断入口 INT2/INT3 作为外部中断 EP0/ EP1。扩展的每个外部中断都可选择任意输入口作为中断触发源，也可各自单独设置上升沿/下降沿/双沿触发中断。为了提高外部中断的抗干扰性能，每个外部中断都集成了模拟滤波器。

所有外部中断都可以用于 STOP 模式唤醒。EPIF 为 EP0/EP1 外部中断状态寄存器，EP0/EP1 对应的各个配置寄存器为 EPOCON/EP1CON。

7.5.3 寄存器描述

表 7.5.3-1 中断使能寄存器 IE

| A8H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-------|--|-------|-----|-----|-------|-----|-------|
| IE | EA | EINT3 | EINT2 | ES0 | ET1 | EINT1 | ETO | EINT0 |
| 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 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | EA | 全局中断使能控制位 0: 关闭 1: 打开 | | | | | | |
| 6 | EINT3 | INT3 使能控制位，用于 Touch Key/WDT/LVD/WKT/EP1 中断 0: 关闭 1: 打开 | | | | | | |

| | | |
|---|-------|---|
| 5 | EINT2 | INT2 使能控制位，用于 I2C/Timer2/PWM/LED/EPO 中断 0: 关闭 1: 打开 |
| 4 | ES0 | UART0 中断使能控制位 0: 关闭 1: 打开 |
| 3 | ET1 | 定时器 1 中断使能控制位 0: 关闭 1: 打开 |
| 2 | EINT1 | 中断 1 使能控制位（中断 1 用于外部中断 1） 0: 关闭 1: 打开 |
| 1 | ET0 | 定时器 0 中断使能控制位 0: 关闭 1: 打开 |
| 0 | EINT0 | 0 使能控制位（中断 0 用于外部中断 0） 0: 关闭 1: 打开 |

表 7.5.3-2 中断优先级寄存器 IP

| B8H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|---|-----|-----|-----|-----|-----|-----|
| IP | - | PX3 | PX2 | PS0 | PT1 | PX1 | PT0 | PX0 |
| 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 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | - | - | | | | | | |
| 6 | PX3 | INT3 优先级控制位，用于 Touch Key/WDT/LVD/WKT/EP1 中断 0: 低优先级 1: 高优先级 | | | | | | |
| 5 | PX2 | INT2 优先级控制位，用于 I2C/Timer2/PWM/LED/EPO 中断 0: 低优先级 1: 高优先级 | | | | | | |
| 4 | PS0 | UART0 优先级控制位 0: 低优先级 1: 高优先级 | | | | | | |
| 3 | PT1 | Timer 1 优先级控制位 0: 低优先级 1: 高优先级 | | | | | | |
| 2 | PX1 | INT1 优先级控制位 0: 低优先级 1: 高优先级 | | | | | | |
| 1 | PT0 | Timer 0 优先级控制位 0: 低优先级 | | | | | | |

| | | |
|---|-----|-----------------------------------|
| | | 1: 高优先级 |
| 0 | PX0 | INT0 优先级控制位 0: 低优先级 1: 高优先级 |

表 7.5.3-3 INT0 控制寄存器 ITOCON

| FAH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|---|-------|---|---|---|---|---|
| ITOCON | ITOP | | ITOPS | | | | | |
| R/W | R/W | | R/W | | | | | |
| 初始值 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-6 | ITOP | 当 ITO (TCON[0]) 为 0 时, ITOP 为触发电平选择 00: 低电平 01: 高电平 10/11: 保留 当 ITO (TCON[0]) 为 1 时, ITOP 为触发边沿选择 00: 下降沿 01: 上升沿 10/11: 双沿 | | | | | | |
| 5-0 | ITOPS | INTO 中断引脚选择 001000: P10 001001: P11 001010: P12 001011: P13 001100: P14 001101: P15 001110: P16 001111: P17 011000: P30 011001: P31 011010: P32 011011: P33 011100: P34 011101: P35 | | | | | | |

表 7.5.3-4 INT1 控制寄存器 IT1CON

| FBH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------|--|-------|---|---|---|---|---|
| ITOCON | IT1P | | IT1PS | | | | | |
| R/W | R/W | | R/W | | | | | |
| 初始值 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-6 | IT1P | 当 IT1 (TCON[0]) 为 0 时, IT1P 为触发电平选择 00: 低电平 | | | | | | |

| | | |
|-----|-------|---|
| | | 01: 高电平 10/11: 保留 当 IT1 (TCON[2]) 为 1 时, IT1P 为触发边沿选择 00: 下降沿 01: 上升沿 10/11: 双沿 |
| 5-0 | IT1PS | INT1 中断引脚选择 001000: P10 001001: P11 001010: P12 001011: P13 001100: P14 001101: P15 001110: P16 001111: P17 011000: P30 011001: P31 011010: P32 011011: P33 011100: P34 011101: P35 |

表 7.5.3-5 EPn 控制寄存器 EPnCON

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--|-------|---|-------|---|---|---|---|---|
| EPnCON | EPnP | | EPnPS | | | | | |
| R/W | R/W | | R/W | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| n=0-1, EPn 代表 EP0/EP1 寄存器地址 EP0CON: DAH EP1CON: DBH | | | | | | | | |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-6 | EPnP | EPnP 为触边沿选择 00: 下降沿 01: 上升沿 10/11: 双沿 | | | | | | |
| 5-0 | EPnPS | EPn 中断引脚选择 001000: P10 001001: P11 001010: P12 001011: P13 001100: P14 001101: P15 001110: P16 | | | | | | |

| | | |
|--|--|---|
| | | 001111: P17 011000: P30 011001: P31 011010: P32 011011: P33 011100: P34 011101: P35 |
|--|--|---|

表 7.5.3-6 外部中断使能寄存器 EPIE

| D8H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-------|------------|---|---|---|---|-------|-------|
| EPIE | - | - | - | - | - | - | EP1IE | EPOIE |
| R/W | - | - | - | - | - | - | R/W | R/W |
| 初始值 | - | - | - | - | - | - | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-2 | - | - | | | | | | |
| 1 | EP1IE | EP1 中断使能信号 | | | | | | |
| 0 | EPOIE | EPO 中断使能信号 | | | | | | |

表 7.5.3-7 外部中断标志寄存器 EPIF

| D9H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-------|----------|---|---|---|---|-------|-------|
| EPIE | - | - | - | - | - | - | EP1IF | EPOIF |
| R/W | - | - | - | - | - | - | R/W | R/W |
| 初始值 | - | - | - | - | - | - | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-2 | - | - | | | | | | |
| 1 | EP1IF | EP1 中断标志 | | | | | | |
| 0 | EPOIF | EPO 中断标志 | | | | | | |

7.6 时钟系统

7.6.1 时钟源

WS51F7030 系列芯片共支持以下时钟源：

- 内部 16MHz RC 振荡器 HRC
- 内部 32KHz RC 振荡器 LRC

7.6.1.1 内部 16MHz RC 振荡器 (HRC)

HRC 是芯片上电后默认的系统时钟，可通过 HRE 位打开或关闭。芯片出厂后，HRC 的频率校正为 16MHz@3.3V/25℃，时钟精度为±1%。

7.6.1.2 内部 32KHz RC 振荡器 (LRC)

LRC 可通过 LRE 位打开或关闭。LRC 设为系统时钟可实现系统低功耗。芯片出厂后，LRC 的频率校正为 32KHz@3.3V/25℃，时钟精度为±1%。

7.6.2 时钟控制关系

HRC 和 LRC 均可作为系统时钟的时钟源，同时它们的 1-16 整数分频时钟也可作为系统时钟的时钟源。寄存器 SCCON 是系统时钟的控制寄存器，通过它们可以设置系统时钟的切换和分频等操作。

每个源时钟都可以单独打开或关闭，当源时钟作为 CPU 时钟或外设时钟时，该时钟自动开启，如下表所示。

表 7.6.2-1 外设时钟映射表

| 外设模块 | 使用的时钟 | 时钟是否自动开启 |
|------|--------|----------|
| 系统时钟 | LRC 时钟 | 是 |
| | HRC 时钟 | 是 |
| PWM | 系统时钟 | 是 |
| | LRC 时钟 | 是 |
| | HRC 时钟 | 是 |
| I2C | HRC 时钟 | 是 |
| WDT | LRC 时钟 | 是 |
| WKT | LRC 时钟 | 是 |
| 其他模块 | 系统时钟 | 否 |

7.6.3 寄存器描述

表 7.6.3-1 寄存器 SCCON

| 86H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------|------------|-----------|---|---|--------|---|---|---|
| SCCON | SCKS | - | - | - | SCKDIV | | | |
| R/W | R/W | - | - | - | R/W | | | |
| 初始值 | 0 | - | - | - | 0 | 0 | 0 | 1 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | SCKS | 系统时钟选择 | | | | | | |

| | | |
|-----|--------|--|
| | | 0: 选择 HRC 时钟 1: 选择 LRC 时钟 |
| 6-4 | - | - |
| 3-0 | SCKDIV | 系统时钟分频 0: 不分频 1: 2 分频 2: 3 分频 ... 15: 16 分频 |

表 7.6.3-2 寄存器 HRCON

| 96H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-------|---|------------|---|---|---|---|---|
| HRCFG | HRE | - | HRCFG[5:0] | | | | | |
| R/W | R/W | - | R/W | | | | | |
| 初始值 | 0 | - | 1 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | HRE | HRC 使能信号, 该位为高则 HRC 打开, 该位为低并且外设没有使用该时钟则 HRC 关闭 | | | | | | |
| 6 | - | - | | | | | | |
| 5-0 | HRCFG | HRC 频率调整寄存器 | | | | | | |

表 7.6.3-3 寄存器 LRCON

| 97H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-------|---|------------|---|---|---|---|---|
| LRCFG | LRE | - | LRCFG[5:0] | | | | | |
| R/W | R/W | - | R/W | | | | | |
| 初始值 | 0 | - | 1 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | LRE | LRC 使能信号, 该位为高则 LRC 打开, 该位为低并且外设没有使用该时钟则 LRC 关闭 | | | | | | |
| 6 | - | - | | | | | | |
| 5-0 | LRCFG | LRC 频率调整寄存器 | | | | | | |

7.7 复位系统

芯片内部有 5 种复位源：上电复位（POR）、掉电复位（BOR）、看门狗复位（WDT）、低电压检测复位（LVD）、外部硬复位和内部软复位。

软复位后 CPU 将从 BOOT（MECON[0]）位指定的位置启动，而其他复位后 CPU 将从 Flash 的 CODE 区启动运行。

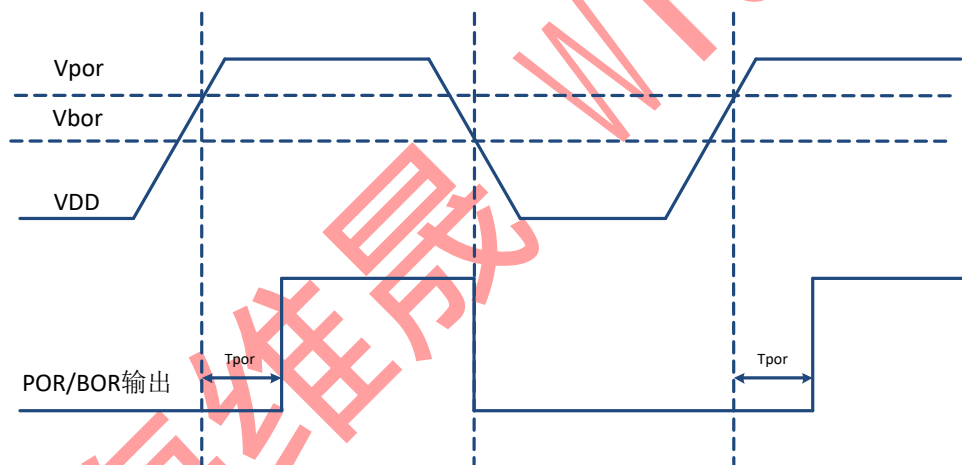
7.7.1 上电复位（POR）和掉电复位（BOR）

芯片上电需要一定时间才能达到正常的工作电压，上电复位电路能够保证芯片上电到正常工作电压前一直处于复位状态，从而保证芯片从初始默认状态开始运行。

芯片掉电是从正常工作电压到不正常工作电压的渐变过程，在这个过程中由于电压的降低芯片内部工作出现异常状态，在这个异常工作状态下芯片可能会执行错误的操作，从而可能导致整个电路系统出错。掉电复位电路强制芯片在电压低于某个电压(Vbor)时持续处于复位状态，直到电压低到所有电路都不能正常工作为止，从而避免芯片在掉电时执行错误操作。

如下图所示为 POR 和 BOR 工作过程，POR 和 BOR 的具体电气参数参照第 10 章电气特性的 [POR 电气特性](#)。

图 7.7.1-1 POR 和 BOR 工作过程



备注:

- Vpor: 上电复位截止电压值
- Vbor: 掉电复位起始电压值
- Trpor: 上电复位截止后持续复位时间

7.7.2 看门狗（WDT）复位

WDT 主要用于监控 CPU 执行指令的情况。如果在指定时间内没有刷新 WDT，则系统可能出现死锁或跑飞，此时 WDT 会产生系统复位。详细描述请参照 [8.4 看门狗定时器（WDT）](#) 章节。

7.7.3 低电压检测（LVD）复位

LVD 的作用类似于掉电复位，不同的是 LVD 工作在芯片正常工作电压区间。开启 LVD 后它将持续监控电源电压 VDD，当 VDD 低于 LVD 设定的域值电压超过 20us 就可以产生触发信号。如果设定 LVD 触发信号产生复位，将复位芯片系统。详细描述请参照 [8.7 低电压检测（LVD）](#) 章节。

7.7.4 外部硬复位和内部软复位

P32 默认情况可配置为外部硬复位 RST 端口或者高阻（[表 7.3.2-1](#) 描述）。如果配置为 RST 端口，拉低 RST 将产生复位信号。RST 复位可以复位整个芯片，在 STOP 状态，RST 复位会唤醒芯片后再复位。一般工作情况下，RST 端口被内部拉高，不会影响内部的复位电路。

通过对 SWRST (PCON[5]) 位写 1，CPU 可以发出复位指令。

7.7.5 寄存器描述

表 7.7.5-1 寄存器 RSTFLG

| 9FH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|------|------------------------------|---|------|------|-------|-------|-------|
| RSTFLG | - | - | - | LVDF | WDRF | PDRF | HWRF | SWRF |
| R/W | - | - | - | R | R | R/WOC | R/WOC | R/WOC |
| 初始值 | - | - | - | 0 | 0 | 0 | 0 | 0 |
| 备注： WOC: 写 0 清除 | | | | | | | | |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-5 | - | - | | | | | | |
| 4 | LVDF | LVD 复位标志，清除 LVDCON 中相关标志自动清除 | | | | | | |
| 3 | WDRF | WDT 复位标志，喂狗刷新 WDT 自动清除 | | | | | | |
| 2 | PDRF | 上电或掉电复位标志，写 0 清除 | | | | | | |
| 1 | HWRF | 硬复位标志，写 0 清除 | | | | | | |
| 0 | SWRF | 软复位标志，写 0 清除 | | | | | | |

7.8 电源管理

7.8.1 LDO

WS51F7030 系列芯片有两种低功耗模式: IDLE 和 STOP 模式, 其中 IDLE 模式时系统最小功耗小于 10uA, STOP 模式时系统最小功耗典型值为 2uA。

7.8.2 IDLE 模式

在 IDLE 模式下, CPU 将停止工作。进入 IDLE 模式前, 除了主时钟, 其他的时钟源可根据需要选择关闭, 以便节省功耗。同样进入 IDLE 模式前, 可根据需要设定芯片某些外设的开关, 打开的外设在 IDLE 下仍然可以正常工作。

进入 IDLE 模式前, 需要先查看一下寄存器 IDLF, 如果所有位都为 0, 则 CPU 将正常进入 IDLE 模式。如果 IDLF 的位不全为 0, 即使有设置进入 IDLE 模式的操作, CPU 也不会进入 IDLE 模式, 而是继续停留在正常工作模式。此时用户需先把对应的中断处理完成, 再重新设置进入 IDLE 模式。

所有复位事件和任何中断事件都将唤醒芯片。中断唤醒 CPU 后, 芯片首先恢复时钟, 然后响应中断, 进入该中断的服务程序。退出中断服务程序后, 芯片将执行置位 IDLE 指令后的指令。退出 IDLE 模式时, IDLE 位将自动清零。**需要注意的是, 在置位 IDLE 的指令后面需要紧接两条 nop 指令, 防止程序出错。**

7.8.3 STOP 模式

STOP 模式是比 IDLE 更深层次的低功耗模式。STOP 模式可以停止所有时钟 (包括主时钟) 和时钟产生电路。如果 WDT 和 WKT 处于打开状态, 则它们使用的时钟模块将处于工作状态, 可以有选择地关闭 WDT 和 WKT 以节省功耗。

类似于 IDLE 模式, 进入 STOP 模式前, 需要先查看 STPF 寄存器, 若有置 1 的位存在, 需要先行处理, 以确保能顺利进入 STOP 模式。

可唤醒的 STOP 模式事件: **外部中断、LVD 中断或复位、外部硬复位、WKT 中断、WDT 中断或复位、触摸中断。**

如果是中断唤醒, 那么唤醒 MCU 后, 芯片首先恢复时钟, 然后响应该中断, 进入该中断的服务程序。退出中断服务程序后, 芯片将执行置位 STOP 指令后的指令。退出 STOP 模式时, STOP 位将自动清零。

在进入 STOP 模式时, 最后一个时钟沿将关闭系统时钟, 然后芯片完全进入 STOP 模式。**需要注意的是, 在置位 STOP 的指令后面需要紧接两条 nop 指令, 防止程序出错。**

7.8.4 寄存器描述

表 7.8.4-1 寄存器 PCON

| 87H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|----------------|-------|--------|---|---|------|------|
| PCON | SMOD | - | SWRST | TSMODE | - | - | STOP | IDLE |
| R/W | R/W | - | W | R | - | - | W | W |
| 初始值 | 0 | - | 0 | 0 | - | - | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | SMOD | UART0 波特率倍频控制位 | | | | | | |

| | | |
|-----|--------|---|
| | | 在 UART0 工作于模式 1,2,3 时, 设置 SMOD=1 会使波特率倍频, 与标准 8051 相同。 |
| 6 | - | - |
| 5 | SWRST | 软复位控制位, 1 有效 设置 SWRST=1 产生软复位, 复位产生后自动清 0。 |
| 4 | TSMODE | 芯片进入测试模式标志 |
| 3-2 | - | - |
| 1 | STOP | STOP 模式控制位, 1 有效 当设置 STOP=1 且 STPF 为 0 时, 芯片进入 STOP 模式, 退出 STOP 模式后自动清 0 |
| 0 | IDLE | IDLE 模式控制位, 1 有效 当设置 IDLE=1 且 IDLF 为 0 时, 芯片进入 IDLE 模式, 退出 IDLE 模式后自动清 0 |

表 7.8.4-2 寄存器 IDLF

| FCH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|---------|--|---|---|---|---|---|---|
| IDLF | - | IDLF | | | | | | |
| R/W | - | R | | | | | | |
| 初始值 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | - | - | | | | | | |
| 6 | IDLF[6] | Touch Key/WDT/LVD/WKT/ External Interrupt 3 事件 | | | | | | |
| 5 | IDLF[5] | I2C/ Timer2/PWM/LED/External Interrupt 2 事件 | | | | | | |
| 4 | IDLF[4] | UART 0 事件 | | | | | | |
| 3 | IDLF[3] | Timer 1 事件 | | | | | | |
| 2 | IDLF[2] | External Interrupt 1 事件 | | | | | | |
| 1 | IDLF[1] | Timer 0 事件 | | | | | | |
| 0 | IDLF[0] | External Interrupt 0 事件 | | | | | | |

表 7.8.4-3 寄存器 STPF

| FEH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|----------|-------------------|---|---|---|---|---|---|
| STPF | - | STPF [6:0] | | | | | | |
| R/W | - | R | | | | | | |
| 初始值 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-6 | - | - | | | | | | |
| 5 | STPF [5] | Touch Key 事件 | | | | | | |
| 4 | STPF [4] | WDT 事件 | | | | | | |
| 3 | STPF [3] | LVD 事件 | | | | | | |
| 2 | STPF [2] | WKT 事件 | | | | | | |
| 1 | STPF [1] | 外部中断 INT2/INT3 事件 | | | | | | |
| 0 | STPF [0] | 外部中断 INTO/INT1 事件 | | | | | | |

8 增强功能

8.1 脉宽调制 (PWM)

8.1.1 简介

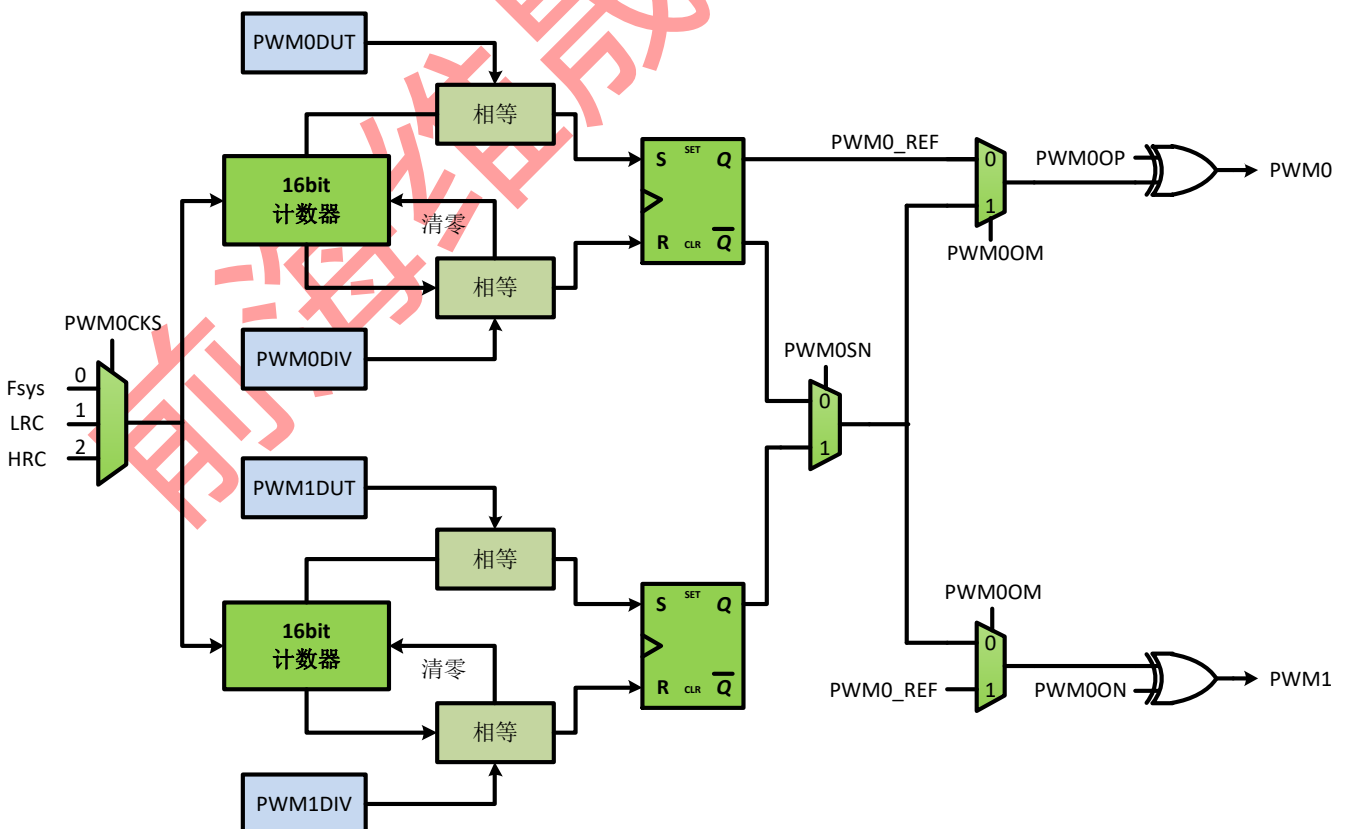
WS51F7030 系列芯片最多可输出 6 路独立的 PWM 波形，每路 PWM 的周期和占空比都可在 16 位范围内任意配置。这 6 路 PWM 又可以两两结合，组成 3 组带 8 位死区控制的互补 PWM 输出。

8.1.2 PWM 工作原理

以为 PWM0/1 为例子 (PWM2/3, PWM4/5 与 PWM0/1 原理一致):

1. 它们的时钟相同，均由 PWM0CKS 从选择系统时钟、LRC、HRC 中选择其一。
2. 当 PWMnSN 为 1 时，它们各自独立。这时 PWM0 的周期为 $PWM0DIV+1$ ，占空比为 $PWM0DUT/(PWM0DIV+1)$ ，输出极性由 PWM0OP 控制；PWM1 的周期为 $PWM1DIV+1$ ，占空比为 $PWM1DUT/(PWM1DIV+1)$ ，输出极性由 PWM0ON 控制。无死区时间。
3. 当 PWMnSN 为 0 时，PWM0 与 PWM1 组成互补 PWM 输出。PWM0 和 PWM1 是在原始参考波形 PWM0_REF 上，通过选择互补模式，配置极性，插入死区时间等操作生成的。其中，原始参考波形的周期为 $PWM0DIV+1$ ，占空比为 $PWM0DUT/(PWM0DIV+1)$ ；互补模式由 PWM0OM 控制；极性由 PWM0OP、PWM0ON 控制；死区时间为 $PWM0DTS+1$ (PWM1DUTL 在互补输出时复用成 PWM0DTS)。波形参考下图。

图 8.1.2-1 PWM0/1 原理示意图



8.1.2.1 PWM 输出波形

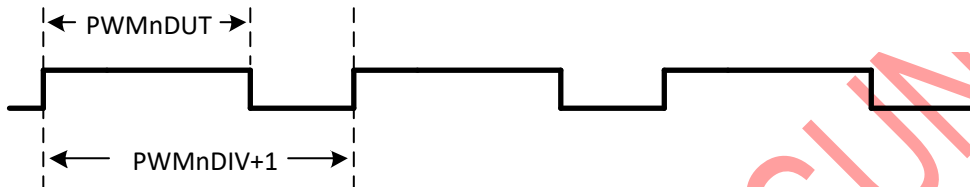
各路 PWM 独立时:

若 PWMnDIV=0 时, PWMn 输出所选的 PWM 时钟源波形。

若 PWMnDIV 不为 0, 而 PWMnDUT=0 时, 如果对应的 PWMnOP 或 PWMnON 为 0 则输出低电平, 否则输出高电平。

当满足条件 $PWMnDIV > PWMnDUT > 0$, PWMnOP 或 PWMnON 为 0 时, PWMn 波形如下图所示。

图 8.1.2-2 各路 PWM 独立时的波形图

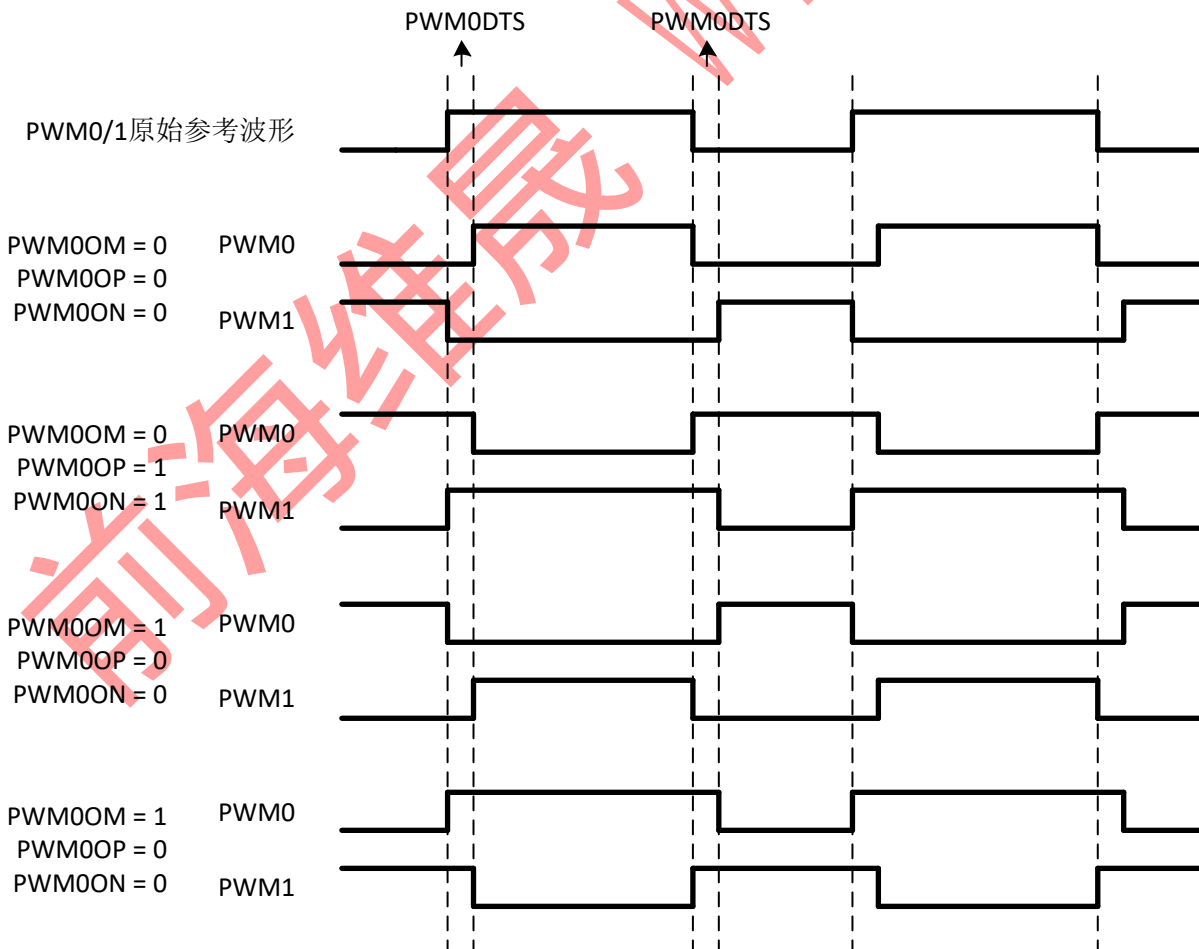


输出互补 PWM 时 (以为 PWM0 和 PWM1 为例):

PWM0/1 原始参考波形同上图;

在 PWM0OP 和 PWM0ON 不变的情况下, PWM0OM 为 1 时, PWM0 输出 PWM0OM 为 0 时 PWM1 的波形, PWM1 输出 PWM0OM 为 0 时 PWM0 的波形。即 PWM0OM 的作用相当于将 PWM0 和 PWM1 的波形互换。

图 8.1.2-3 插入死区时间的 PWM0/1 互补波形配置示意图



8.1.2.2 PWM 中断

PWMn 中断通过 PWMIE 的对应位使能,当 PWM 计数器计数到等于 PWMnDIV 时产生的中断。寄存器 PWMFLG 包含 6 路 PWM 的中断标志位。

8.1.3 寄存器描述
表 8.1.3-1 寄存器 PWMEN

| 8040H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|--------|------------------|--------|--------|--------|--------|--------|--------|
| PWMEN | - | - | PWM5EN | PWM4EN | PWM3EN | PWM2EN | PWM1EN | PWM0EN |
| R/W | - | - | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | - | - | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-6 | - | - | | | | | | |
| 5 | PWM5EN | PWM5 使能控制位, 1 有效 | | | | | | |
| 4 | PWM4EN | PWM4 使能控制位, 1 有效 | | | | | | |
| 3 | PWM3EN | PWM3 使能控制位, 1 有效 | | | | | | |
| 2 | PWM2EN | PWM2 使能控制位, 1 有效 | | | | | | |
| 1 | PWM1EN | PWM1 使能控制位, 1 有效 | | | | | | |
| 0 | PWM0EN | PWM0 使能控制位, 1 有效 | | | | | | |

表 8.1.3-2 寄存器 PWMIE

| 8042H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|--------|--------------------|--------|--------|--------|--------|--------|--------|
| PWMEN | - | - | PWM5IE | PWM4IE | PWM3IE | PWM2IE | PWM1IE | PWM0IE |
| R/W | - | - | R/W | R/W | R/W | R/W | R/W | R/W |
| 初始值 | - | - | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-6 | - | - | | | | | | |
| 5 | PWM5IE | PWM5 中断使能控制位, 1 有效 | | | | | | |
| 4 | PWM4IE | PWM4 中断使能控制位, 1 有效 | | | | | | |
| 3 | PWM3IE | PWM3 中断使能控制位, 1 有效 | | | | | | |
| 2 | PWM2IE | PWM2 中断使能控制位, 1 有效 | | | | | | |
| 1 | PWM1IE | PWM1 中断使能控制位, 1 有效 | | | | | | |
| 0 | PWM0IE | PWM0 中断使能控制位, 1 有效 | | | | | | |

表 8.1.3-3 寄存器 PWMFLG

| 8044H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-----|----|---------|---------|---------|---------|---------|---------|
| PWMFLG | - | - | PWM5FLG | PWM4FLG | PWM3FLG | PWM2FLG | PWM1FLG | PWM0FLG |
| R/W | - | - | WOC/R | WOC/R | WOC/R | WOC/R | WOC/R | WOC/R |
| 初始值 | - | - | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |

| | | |
|-----|---------|-------------------------|
| 7-6 | - | - |
| 5 | PWM5FLG | PWM5 中断标志位, 1 有效, 写 0 清 |
| 4 | PWM4FLG | PWM4 中断标志位, 1 有效, 写 0 清 |
| 3 | PWM3FLG | PWM3 中断标志位, 1 有效, 写 0 清 |
| 2 | PWM2FLG | PWM2 中断标志位, 1 有效, 写 0 清 |
| 1 | PWM1FLG | PWM1 中断标志位, 1 有效, 写 0 清 |
| 0 | PWM0FLG | PWM0 中断标志位, 1 有效, 写 0 清 |

表 8.1.3-4 寄存器 PWMnCFG

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|--------|--------|--------|--------|---|---|---------|---|
| PWMnCFG | PWMnOP | PWMnON | PWMnOM | PWMnSN | - | - | PWMnCKS | |
| R/W | R/W | R/W | 0 | R/W | - | - | R/W | |
| 初始值 | 0 | 0 | 0 | - | - | - | 0 | 0 |

其中 n=0/2/4

寄存器 PWM0CFG 地址: 8050H, 用于配置 PWM0 和 PWM1

寄存器 PWM2CFG 地址: 8060H, 用于配置 PWM2 和 PWM3

寄存器 PWM4CFG 地址: 8070H, 用于配置 PWM4 和 PWM5

| 位编号 | 位符号 | 说明 |
|-----|---------|---|
| 7 | PWMnOP | PWM0/2/4 输出极性选择, 为 1 时输出取反 |
| 6 | PWMnON | PWM1/3/5 输出极性选择, 为 1 时输出取反 |
| 5 | PWMnOM | 互补 PWM 模式选择, 为 1 时 PWMn 与 PWM(n+1)的波形互换后输出。 |
| 4 | PWMnSN | PWM 独立输出使能 0: PWMn 和 PWM(n+1)组成互补 PWM 输出 1: PWMn 和 PWM(n+1)独立输出 |
| 3-2 | - | - |
| 1-0 | PWMnCKS | PWM 时钟选择 00: 系统时钟 01: LRC 内部低速时钟 10: HRC 内部高速时钟 11: 保留 |

表 8.1.3-5 寄存器 PWMnDIV

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|---------------|---|---|---|---|---|---|---|
| PWMnDIVL | PWMnDIV[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PWMnDIVH | PWMnDIV[15:8] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

其中 n=0/1/2/3/4/5/6/7

寄存器 PWM0DIVL 地址: 8056H

寄存器 PWM0DIVH 地址: 8057H

寄存器 PWM1DIVL 地址: 805EH

| 寄存器 PWM1DIVH 地址: 805FH | | |
|------------------------|----------|-------------------------------------|
| 寄存器 PWM2DIVL 地址: 8066H | | |
| 寄存器 PWM2DIVH 地址: 8067H | | |
| 寄存器 PWM3DIVL 地址: 806EH | | |
| 寄存器 PWM3DIVH 地址: 806FH | | |
| 寄存器 PWM4DIVL 地址: 8076H | | |
| 寄存器 PWM4DIVH 地址: 8077H | | |
| 寄存器 PWM5DIVL 地址: 807EH | | |
| 寄存器 PWM5DIVH 地址: 807FH | | |
| 位编号 | 位符号 | 说明 |
| 7-0 | PWMnDIVL | PWM 周期计数寄存器的低 8 位, PWM 周期为 PWMDIV+1 |
| 7-0 | PWMnDIVH | PWM 周期计数寄存器的高 8 位, PWM 周期为 PWMDIV+1 |

表 8.1.3-6 寄存器 PWMnDUT/PWM(n-1)DTS

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------------------|-------------------------------|--|---|---|---|---|---|---|
| PWMnDUTL | PWMnDUT[7:0]/PWM(n-1)DTS[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PWMnDUTH | PWMnDUT[15:8] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 其中 n=0/1/2/3/4/5 | | | | | | | | |
| 寄存器 PWM0DUTL 地址: 8054H | | | | | | | | |
| 寄存器 PWM0DUTH 地址: 8055H | | | | | | | | |
| 寄存器 PWM1DUTL/PWM0DTS 地址: 805CH | | | | | | | | |
| 寄存器 PWM1DUTH 地址: 805DH | | | | | | | | |
| 寄存器 PWM2DUTL 地址: 8064H | | | | | | | | |
| 寄存器 PWM2DUTH 地址: 8065H | | | | | | | | |
| 寄存器 PWM3DUTL/PWM2DTS 地址: 806CH | | | | | | | | |
| 寄存器 PWM3DUTH 地址: 806DH | | | | | | | | |
| 寄存器 PWM4DUTL 地址: 8074H | | | | | | | | |
| 寄存器 PWM4DUTH 地址: 8075H | | | | | | | | |
| 寄存器 PWM5DUTL/PWM4DTS 地址: 807CH | | | | | | | | |
| 寄存器 PWM5DUTH 地址: 807DH | | | | | | | | |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | PWMnDUTL/ PWM(n-1)DTS | PWM 占空比计数寄存器的低 8 位, PWM 占空比为 PWMnDUT/(PWMnDIV+1) 当 n=1/3/5 时, 若 PWMnSN=0, 则该八位寄存器为 PWM(n-1)DTS[7:0], 用于死区时间控制 | | | | | | |
| 7-0 | PWMnDUTH | PWM 占空比计数寄存器的高 8 位, PWM 占空比为 PWMnDUT/(PWMnDIV+1) | | | | | | |

8.2 通用串行接口 (UART 0)

8.2.1 简介

UART0 是一个全双工同步/异步串行数据收发器，完全兼容标准 8051，但不同于标准 8051 的是本 UART0 可以设置任意波特率。

8.2.2 UART0

UART0 是一个全双工同步/异步串行数据收发器，可同时发送和接收数据，其接收器有一字节的缓存，接收完的一个字节数据会被立即送到缓存寄存器，接着可以接收新的数据。在新的一字节数据接收完之前，前一个字节数据必须被读取，否则会被新数据覆盖。寄存器 SOBUF 是 UART0 的发送/接收数据寄存器，在物理上 SOBUF 实际是两个寄存器，一个是数据发送寄存器，另一个是数据接收寄存器，写 SOBUF 会将数据写入发送寄存器并启动数据发送，而读 SOBUF 会读取接收寄存器中接收到的一字节数据。

UART0 有 4 种工作模式，如下表所示。

表 8.2.2-1 UART0 通信工作模式

| SM00 | SM10 | 模式 | 描述 | 波特率 |
|------|------|----|---------|---|
| 0 | 0 | 0 | 同步移位模式 | $F_{sys}/12$ |
| 0 | 1 | 1 | 8 位异步模式 | 波特率为 $2^{SMOD} \times F_{sys}/(32 * \text{定时器 } 1/2 \text{ 溢出率})$ ， 详见 T2CON 中 RCLK 和 TCLK |
| 1 | 0 | 2 | 9 位异步模式 | 当 SMOD=0 时，波特率为 $F_{sys}/64$ 当 SMOD=1 时，波特率为 $F_{sys}/32$ |
| 1 | 1 | 3 | 9 位异步模式 | 波特率为 $2^{SMOD} \times F_{sys}/(32 * \text{定时器 } 1/2 \text{ 溢出率})$ ， 详见 T2CON 中 RCLK 和 TCLK |

备注：

1. F_{sys} 即为系统时钟。
2. 以上波特率的前提条件是 SODE=0，当 SODE=1 时，波特率由 SODIV 值决定。
3. 由于定时器 2 的时钟是包含不分频 F_{sys} ，所以选择定时器 2 作为 UART0 时钟发生器会有更高的波特率。

8.2.2.1 模式 0

在模式 0，UART0 同步收发数据。端口 UART0_TX 输出移位时钟，端口 UART0_RX 输出或接收数据。传输数据长度为 8 位，从最低位开始传输，波特率固定为系统时钟频率的 1/12。写数据到寄存器 S0BUF 会启动 UART0 发送数据。如果要接收数据，需设置 REN=1 并清除 RI0 标志，当接收到一字节数据时，RI0 会置 1。

图 8.2.2-1 UART0 模式 0 发送数据波形

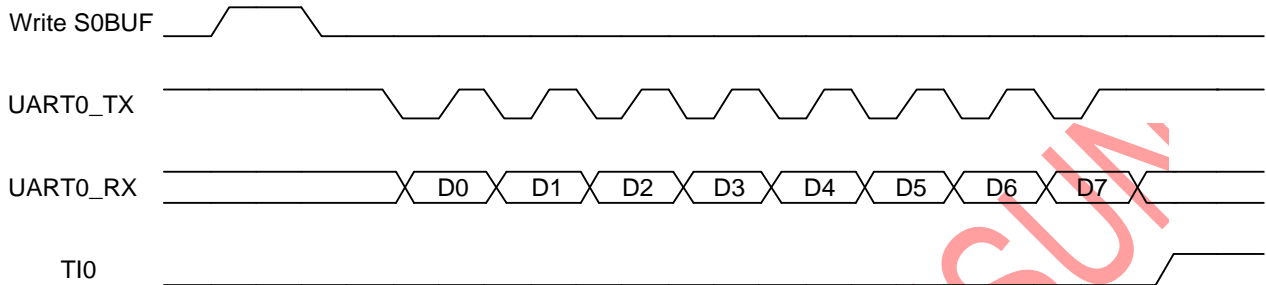
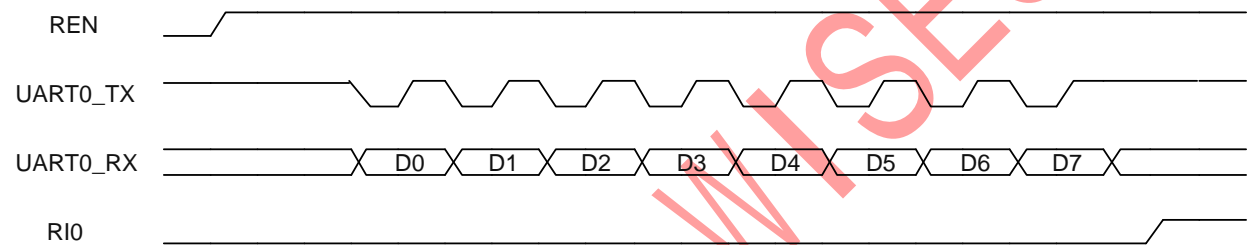


图 8.2.2-2 UART0 模式 0 接收数据波形



8.2.2.2 模式 1

在模式 1，UART0 为 8 位异步收发模式。可通过设置 RCLK(T2CON[5])和 TCLK(T2CON[4])来选择定时器 1 或定时器 2 的溢出信号作为 UART0 的时钟，此时设置定时器的溢出频率就可以调整 UART0 的波特率。另外，可通过 SMOD(PCON[7])来选择波特率倍频。

写入数据到寄存器 S0BUF 会启动 UART0 发送。第一个传送的位是 START 位（0），然后是 8 位数据（低位先发），最后传送的是 STOP 位（1）。

在接收状态，UART0 通过检测端口 UART0_RX 的下降沿来同步。传送过程完成后，8 位数据存放在寄存器 S0BUF，有效停止位值存放在 RB80 位。

图 8.2.2-3 UART0 模式 1 发送数据波形

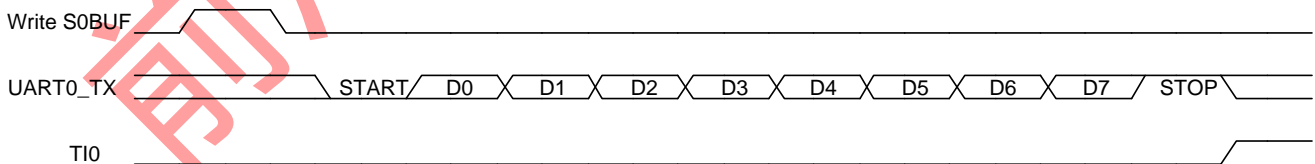
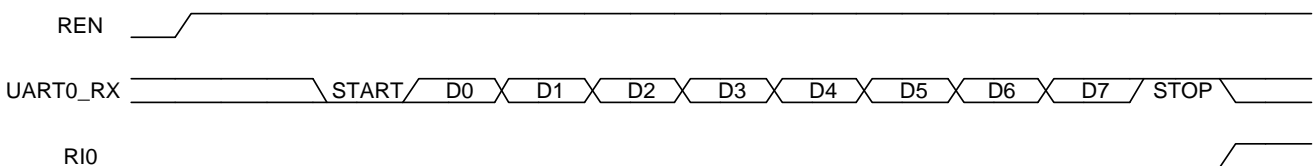


图 8.2.2-4 UART0 模式 1 接收数据波形



8.2.2.3 模式 2

在模式 2, UART0 为 9 位异步收发模式。通过设置 SMOD(PCON[7])位可选择波特率固定为 $F_{sys}/32$ 或 $F_{sys}/64$ 。

写入数据到寄存器 S0BUF 会启动 UART0 发送。第一个传送的位是 START 位 (0), 然后是 9 位数据 (低位先发), 第 9 位数据是 TB80 位, 最后传送的是 STOP 位 (1)。

在接收状态, UART0 通过检测端口 UART0_RX 的下降沿来同步。传送过程完成后, 8 位数据存放在寄存器 S0BUF, 第 9 位数据存放在 RB80 位。

图 8.2.2-6 UART0 模式 2 发送数据波形

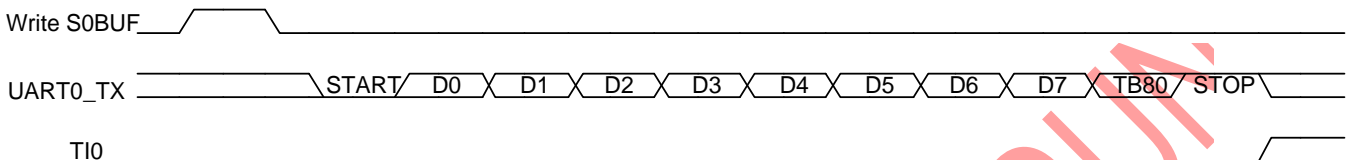
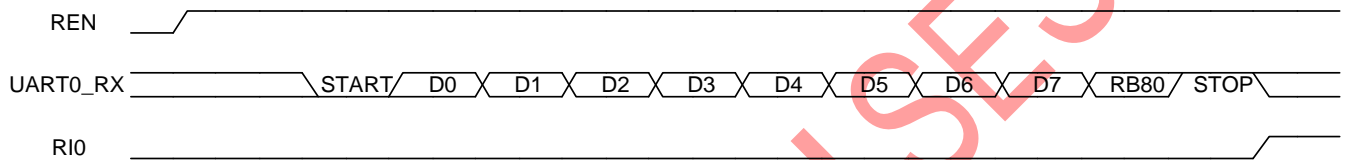


图 8.2.2-6 UART0 模式 2 接收数据波形



8.2.2.4 模式 3

模式 2 和模式 3 的唯一不同的是模式 3 的波特率可通过定时器 1 或定时器 2 来产生, 可参考模式 1 的示意图。波特率设置可参考模式 1 介绍, 而其他功能描述参考模式 2。

8.2.2.5 UART0 多机通信

UART0 模式 2 和 3 中有一个专门适用于多机通信的机制, 当 SM20 位置 1, 只有接收到第 9 位数据为 1(RB80=1) 的从机才会产生接收中断, 利用这个功能可进行多机通信, 操作方法如下:

- ① 所有主机和从机都设置为工作在模式 2 或模式 3, 并且将 SM20 位都置为 1。
- ② 主机首先发送从机地址, 并将第 9 位数据设为 1, 这样所有的从机都会接收该地址数据并置位 RIO。
- ③ 从机的软件用预先定义的地址和接收的地址数据进行比较, 地址相同的从机设置 SM20=0。
- ④ 主机继续传送后面的数据并设置第 9 位为 0, 此时其他从机 SM20 仍然为 1 不会置位 RIO, 就只有被寻址的从机才会接收数据并置位 RIO, 从而实现多机通讯。

8.2.2.6 定时器波特率分频设置

在标准 8051 中, UART0 的波特率固定为定时器溢出率的 32 分频。由于 WS51F7030 系列芯片的主时钟最高为 16MHz (或 16MHz 的分频), 配置比较高的波特率时有较大误差, 因此 UART0 的波特率不再固定为定时器溢出率的 32 分频, 而是可以由 SODIV 设置。

例如: 当 UART0 的波特率固定为定时器溢出率的 32 分频时, 选择定时器 2 作为 UART0 的波特率发生器, 如果要配置波特率为 115200, 计算公式为: $16000000 \div 32 \div 115200 = 4.34$, 由于定时器只能取整数, 所以取 4 (即每 4 个系统时钟周期定时器溢出一次), 误差率约为 8.5%, 这么大的误差率会导致通信不正常。由于系统时钟是固定的, 要达到更准确的波特率, 只能通过修改分频系数来实现。如果设置定时器 5 个时钟周期溢出, 那么: $16000000 \div 115200 \div 5 = 27.78$ 。取分频数为 28, 那么波特率为 114285, 和 115200 相比, 误差率约为 0.8%, 一般情况下不会影响 UART0 通信。另外, 更小的分频数也可以实现更高的波特率配置。

芯片默认的分频系数为 32, 与标准 8051 相同。如果要更改分频系数, 通过设置 SODE=1 来使能, SODIV 的数

值表示不同的分频系数，详见寄存器 SOCFG 描述。

另外，当 SODE=1 时，SMOD 位将不起作用。

8.2.3 寄存器描述

表 8.2.3-1 寄存器 SOCON

| 98H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|------|--|------|------|------|------|-----|-----|
| SOCON | SM00 | SM10 | SM20 | RENO | TB80 | RB80 | TIO | RIO |
| 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 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | SM00 | 串口 0 模式选择位，详见表 8.2.2-1 | | | | | | |
| 6 | SM10 | | | | | | | |
| 5 | SM20 | 多机通信使能位，1 有效 | | | | | | |
| 4 | RENO | 串行接收使能位，1 有效 | | | | | | |
| 3 | TB80 | 发送的第 9 位数据 在模式 2 和 3，这个位用于 UART0 发送数据，对应发送数据的第 9 位 (例如奇偶校验或多机通信)，由软件控制 | | | | | | |
| 2 | RB80 | 接收的第 9 位数据 在模式 2 和 3，这个位用于 UART0 接收数据，对应接收数据第 9 位；模式 1 时该位 为停止位；如果 SM20=1，该位为多主机判断位；在模式 0 这个位没有使用。 | | | | | | |
| 1 | TIO | 发送中断标志，1 有效，写 0 清除 | | | | | | |
| 0 | RIO | 接收中断标志，1 有效，写 0 清除 | | | | | | |

表 8.2.3-2 寄存器 SOBUF

| 99H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|------------|---|---|---|---|---|---|---|
| SOBUF | SOBUF[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | SOBUF | 发送接收缓冲器 写 SOBUF 将启动发送所写的 数据 读 SOBUF 将读取已经接收的 数据 | | | | | | |

表 8.2.3-3 寄存器 SOCFG

| 9AH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|------|--------------|---|------------|---|---|---|---|
| SOCFG | SODE | - | - | SODIV[4:0] | | | | |
| R/W | R/W | - | - | R/W | | | | |
| 初始值 | 0 | - | - | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | SODE | 定时器分频配置使能控制位 | | | | | | |

| | | |
|-----|-------|--|
| | | 0: UART0 波特率按照原来的配置 1: UART0 波特率由 SODIV 来配置 |
| 6-5 | - | - |
| 4-0 | SODIV | 定时器分频配置寄存器，仅在 SODE=1 时有效 模式 0: 发送时，须满足 $SODIV \geq 2$ ；接收时， $SODIV \geq 3$ 模式 1: 发送时，须满足 $SODIV \geq 0$ ；接收时， $SODIV \geq 6$ 模式 2: 发送时，须满足 $SODIV \geq 2$ ；接收时， $SODIV \geq 7$ 模式 3: 发送时，须满足 $SODIV \geq 0$ ；接收时， $SODIV \geq 0$ |

前海维晟 WISESUN

8.3 I²C 接口 (I2C)

8.3.1 简介

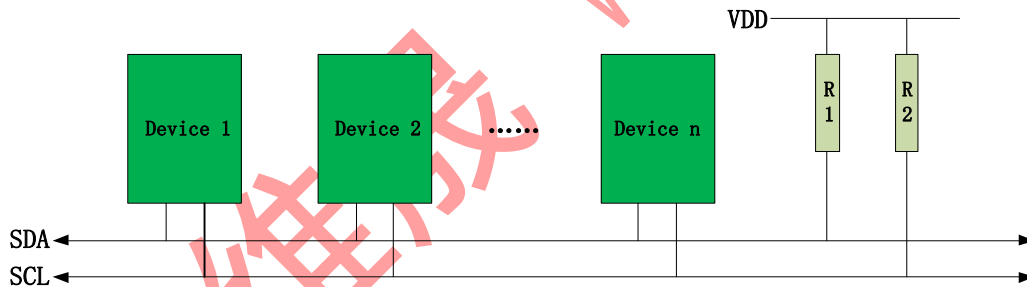
I2C 接口支持芯片与外围 I2C 器件以标准 I2C 协议进行串行数据传输，其主要特点如下

- 支持主机或从机模式，可自动切换角色
- 7 位从机地址
- 支持多主机仲裁
- 支持广播功能
- 支持标准/快速/高速模式，速度分别达到 100K / 400K / 2M bits/s
- 对系统时钟没有要求，在任何系统时钟 I2C 接口都可正常通讯

8.3.2 I2C 总线互联

如下图所示，I2C 总线用 2 根线在设备间传输数据，分别为 SCL（串行时钟线）和 SDA（串行数据线）。连接 SDA 和 SCL 的 IO 端口都是开漏结构，所以总线上必须有上拉电阻。如果 I2C 接口速度只需要工作在标准/快速模式下，则可使能内部强上拉电阻（10 K Ω ）或外挂 10 K Ω 的电阻；如果 I2C 接口需要工作在高速模式，则必须外挂 100 Ω 左右的电阻。

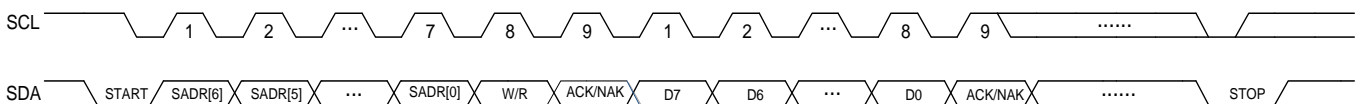
图 8.3.2-1 I2C 总线互连图



8.3.3 I2C 通信原理

标准的 I2C 通信由四部分组成：START 信号、从机地址传输、数据传输和 STOP 信号，数据传输是以 START 信号开始，以 STOP 信号结束。如下图所示，I2C 主机发送 START 信号开启 I2C 通信，然后主机发送从机地址寻址从机，接收到从机 ACK 信号后，开始发送或接收数据，在全部数据传送结束后，由主机发送 STOP 信号，结束通信。

图 8.3.3-1 I2C 通信时序图



备注：

START：如波形所示，SCL 为高时 SDA 由高变低

STOP：如波形所示，SCL 为高时 SDA 由低变高

SADR：I2C 的从机地址

W/R：该位为 0 时为主机写数据传输，该位为 1 时为主机读数据传输

ACK：SCL 为高时 SDA 为低

NAK: SCL 为高时 SDA 为高

8.3.4 I2C 工作模式

I2C 接口有 3 种工作模式：等待模式、从机模式、主机模式。

默认情况下，I2C 处于等待模式。

设置 STA 为 1，I2C 主动发送 START 信号，并从等待模式切换到主机模式。作为主机，I2C 发送的第一个字节数据为从机地址，该字节需要在设置 STA (I2CCON[3]) 之前写入到寄存器 I2CTXD。如果从机回应 NAK，I2C 将回到等待模式并产生中断；如果从机回应 ACK，则总线根据第一个字节第 8 位的读写方向进行数据传输。在从机地址和数据传输过程中，如果收到 NAK 回应、出现仲裁丢失、出现总线错误等，I2C 将回到等待模式并产生中断。如果数据传输完成，则设置 STP 为 1，I2C 将发送 STOP 信号结束总线传输。

在等待模式时，如果接受到 START 信号，则 I2C 从等待模式切换到从机模式。作为从机，I2C 能识别从机地址、掩码地址和广播地址 (0x00)，设置 I2CADM 则可打开地址掩码功能，设置 GCE 为 1 则可开启广播地址的识别。如果地址正确则 I2C 回应 ACK，并通过地址字节的第 8 位判断数据传输的方向，否则回应 NAK，回到等待模式。在从机地址和数据传输过程中，如果接收到 STOP 信号、收到 NAK 回应、出现仲裁丢失或出现总线错误，I2C 将回到等待模式并产生中断。

设置 BSHE 为 1，开启总线暂停功能，此时 I2C 在地址和数据传输时，如果来不及回应会拉低 SCL，从而暂停总线传输直到软件清除相应标志位(IF_RXADR/IF_RXDAT/IF_TXDAT)。设置 BSHE 为 0，关闭总线暂停功能，此时 I2C 必须及时清除对应标志位，否则 I2C 回应 NAK，导致传输出错。

如果没有开启总线暂停功能，则在使用时软件要尽量快速处理，避免 I2C 来不及清除标志位(IF_RXADR/IF_RXDAT/IF_TXDAT)时，产生错误的 ACK/NAK 回应。

备注：

如果 I2C 工作在复合传输方式（请参考 I2C 协议），本芯片 I2C 需要在 RESTART 前发送一个 STOP 信号，相关例程请参考 WS51F7030 的应用软件库。

8.3.5 寄存器描述

表 8.3.5-1 寄存器 I2CCON

| A1H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|---|-------|-------|-----|-----|-------|---|
| I2CCON | I2CE | I2CIE | STAIE | STPIE | STA | STP | I2CKD | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| 初始值 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | I2CE | I2C 模块使能位，1 有效 | | | | | | |
| 6 | I2CIE | I2C 中断使能位，1 有效 | | | | | | |
| 5 | STAIE | 总线出现 START 信号，中断使能信号 | | | | | | |
| 4 | STPIE | 总线出现 STOP 信号，中断使能信号 | | | | | | |
| 3 | STA | 发送 START 信号 | | | | | | |
| 2 | STP | 发送 STOP 信号 | | | | | | |
| 1-0 | I2CKD | I2C 时钟分频选择寄存器 00: I2C 模块时钟 F_{i2c} 选择内部 16MHz 高速时钟 01: I2C 模块时钟 F_{i2c} 选择内部 16MHz 高速时钟/2 10: I2C 模块时钟 F_{i2c} 选择内部 16MHz 高速时钟/4 | | | | | | |

11: I2C 模块时钟 F_{i2c} 选择内部 16MHz 高速时钟/8

表 8.3.5-2 寄存器 I2CADR

| A2H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|-----------------------|---|---|---|---|---|---|
| I2CADR | GCE | DEVADR | | | | | | |
| R/W | R/W | R/W | | | | | | |
| 初始值 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | GCE | 识别广播地址 0x00 使能位, 1 有效 | | | | | | |
| 6-0 | DEVADR | I2C 从机地址, 在作从机时有效 | | | | | | |

表 8.3.5-3 寄存器 I2CFG0

| A3H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|--|---|---|---|---|---|---|
| I2CFG0 | BSHE | I2CADM | | | | | | |
| R/W | R/W | R/W | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | BSHE | I2C 暂停总线功能使能位, I2C 做主机时建议开启该功能 设置 BSHE 为 1 开启拉低 SCL 暂停总线功能, I2C 在如下情况下暂停总线 1. 作为从机接收到地址且为主机读从机操作时, 如果没有清除 IF_RXADR 标志, 表明从机没有准备好要输出的数据, 从而暂停总线 2. 接收到数据后, 如果没有清除 IF_RXDAT 标志, 表明 I2C 接口忙暂停总线 3. 作为主机进行主机写从机操作时, 发送完从机地址后, 如果没有清除 IF_TXDAT 标志, 表明主机没有准备好要输出的数据, 从而暂停总线 备注: 这样设计是为了兼容主机是软件模拟 I2C 的应用, 此时, 主机的软件必须等待足够长的时间让从机响应每字节数据传输的处理。 | | | | | | |
| 6-0 | I2CADM | 掩膜 I2C 接收到的从机地址, 跟 7 位地址一一对应 | | | | | | |

表 8.3.5-4 寄存器 I2CFG1

| A4H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|---|---|---|---|---|---|---|
| I2CFG1 | - | I2CCKD | | | | | | |
| R/W | - | R/W | | | | | | |
| 初始值 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | - | - | | | | | | |
| 6-0 | I2CCKD | I2C 作为 master 输出数据频率选择 $F_{i2c}/(I2CCKD + 8)$ | | | | | | |

表 8.3.5-5 寄存器 I2CTXD

| A5H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|--------|---|---|---|---|---|---|---|
| I2CTXD | I2CTXD | | | | | | | |
| R/W | R/W | | | | | | | |

| | | | | | | | | |
|-----|--------|-------------|---|---|---|---|---|---|
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | I2CTXD | I2C 发送数据寄存器 | | | | | | |

表 8.3.5-6 寄存器 I2CRXD

| | | | | | | | | |
|--------|--------|-------------|---|---|---|---|---|---|
| A6H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2CRXD | I2CRXD | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-0 | I2CRXD | I2C 接收数据寄存器 | | | | | | |

表 8.3.5-7 寄存器 I2CFLG

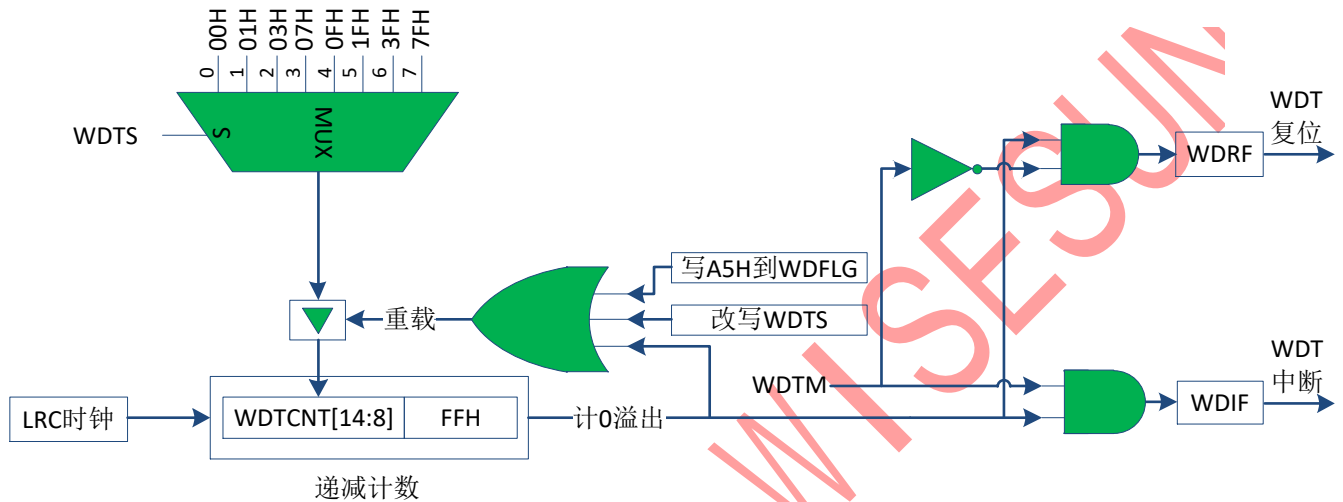
| | | | | | | | | |
|--------|-----------|--|-----------|----------|----------|----------|----------|----------|
| A7H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2CFLG | BUSIDLE | RXNAK | IF_LSTARB | IF_RXSTA | IF_RXSTP | IF_TXDAT | IF_RXDAT | IF_RXADR |
| R/W | R | R | R/WOC | R/WOC | R/WOC | R/WOC | R/WOC | R/WOC |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | BUSIDLE | 总线状态标志 0: 总线繁忙 1: 总线空闲 | | | | | | |
| 6 | RXNAK | 总线接收响应标志 0: 总线接收到 ACK 1: 总线接收到 NAK | | | | | | |
| 5 | IF_LSTARB | I2C 发送的数据和接收到的数据不一致时出现总线丢失, 高有效, 写 0 清 | | | | | | |
| 4 | IF_RXSTA | I2C 总线接收到 START 信号 当 I2C 为主机时发送的 START 标志也可以在此查看 START 信号是否成功发送 | | | | | | |
| 3 | IF_RXSTP | 总线接收到 STOP 信号, 作为主机时发送的 STOP 标志也可以在此查看是否已经成功发送 | | | | | | |
| 2 | IF_TXDAT | 当 I2C 为主机时表明已发送一个字节数据并接收到响应标志, 查看 RXNAK 可知从机响应类型 当 I2C 为从机时表明已发送一个字节数据, 此时不需要查看主机响应类型 | | | | | | |
| 1 | IF_RXDAT | I2C 接收一个字节数据完成 | | | | | | |
| 0 | IF_RXADR | I2C 作为从机正确接收到从机地址 软件通过寄存器 I2CRXD 的第 0 位为 1/0 来判断 I2C 主机是读/写操作 | | | | | | |

8.4 看门狗定时器 (WDT)

8.4.1 简介

看门狗定时器 (WDT) 为一个 15 位递减计数器，如下图所示，其计数时钟为 LRC 时钟，有 8 位调节精度，计数范围为 7.8125ms-1s。

图 8.4.1-1 看门狗模块结构图



8.4.2 功能描述

WDT 主要用于监控系统，避免芯片因程序跑飞或外界干扰出现死机。

写 0xA5 到寄存器 WDFLG 将开启并刷新 WDT 计数器，而后 WDT 内部计数器进行递减计数，如果软件不能在计数到 0 之前再次刷新 WDT，WDT 将产生内部复位或中断。另外，修改 WDT 时间配置寄存器 WDTs 也会刷新 WDT。写 0xA5 到寄存器 WDFLG 将关闭 WDT 计数器，读寄存器 WDFLG 可得到 WDT 状态标志。

在 STOP 模式下，如果 WDT 处于开启状态，则 WDT 所选的时钟源正常工作，如果看门狗事件设为中断，则看门狗事件可唤醒 CPU。

8.4.3 寄存器描述

表 8.4.3-1 寄存器 WDTCON

| A9H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------|--|---|---|---|------|---|---|
| WDTCON | WDTM | - | - | - | - | WDTs | | |
| R/W | R/W | - | - | - | - | R/W | | |
| 初始值 | 0 | - | - | - | - | 1 | 1 | 1 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | WDTM | WDT 触发模式选择 0: WDT 溢出后产生复位 1: WDT 溢出后产生中断 | | | | | | |

| | | |
|-----|------|--|
| 6-3 | - | - |
| 2-0 | WDTS | WDT 时间选择 0: 256 个时钟周期, 约为 7.8125ms 1: 512 个时钟周期, 约为 15.625ms 2: 1024 个时钟周期, 约为 31.25ms 3: 2048 个时钟周期, 约为 62.5ms 4: 4096 个时钟周期, 约为 125ms 5: 8192 个时钟周期, 约为 250ms 6: 16384 个时钟周期, 约为 500ms 7: 32768 个时钟周期, 约为 1s |

表 8.4.3-2 寄存器 WDTFLG

| AAH | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--|-------|-----------------------------------|---|---|---|---|-------|-------|
| WDTFLG | WDTON | - | - | - | - | - | WDTIF | WDTRF |
| R/W | R | - | - | - | - | - | R | R |
| 初始值 | 0 | - | - | - | - | - | 0 | 0 |
| 备注: 写 0xA5 到寄存器 WDTFLG 将开启并刷新 WDT 计数器, 写 0x5A 到寄存器 WDTFLG 将关闭 WDT 计数器 | | | | | | | | |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | WDTON | WDT 工作状态, 为 1 则 WDT 已开启 | | | | | | |
| 6-2 | - | - | | | | | | |
| 1 | WDTIF | WDT 中断标志, 对 WDTFLG 写 0xA5 时将清除该标志 | | | | | | |
| 0 | WDTRF | WDT 复位标志, WDTFLG 写 0xA5 时将清除该标志 | | | | | | |

8.5 自唤醒定时器 (WKT)

8.5.1 简介

芯片内置了一个自唤醒定时器 (WKT)，用于低功耗模式下的周期唤醒芯片，也可用作通用定时器。

8.5.2 功能描述

WKT 固定使用内部 32K 时钟源 LRC，如果开启 WKT 计数，则 LRC 会自动开启。

WKT 为一个简单的 15 位递增计数定时器，其时钟源固定为内部 32K 时钟 LRC，因此其计数范围约为 1s，通过 WKTSS 可选每 0.5/1 秒产生一次中断，如下图所示。



8.5.3 寄存器描述

表 8.5.3-1 寄存器 WKTCON

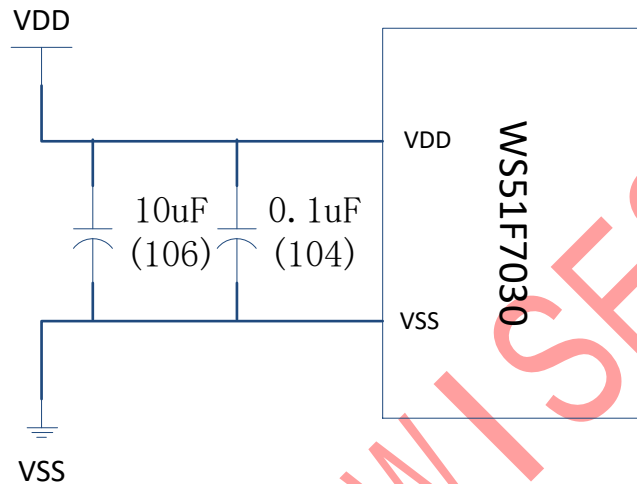
| D1H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|---------------------------------|---|---|---|---|---|-------|
| WKTCON | WKTE | WKTSS | - | - | - | - | - | WKTIF |
| R/W | R/W | R/W | - | - | - | - | - | R/W |
| 初始值 | 0 | 0 | - | - | - | - | - | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | WKTE | WKT 使能信号 | | | | | | |
| 6 | WKTSS | 0.5/1 秒中断选择 0: 0.5s 1: 1s | | | | | | |
| 5-1 | - | - | | | | | | |
| 0 | WKTIF | WKT 中断标志 | | | | | | |

8.6 低压差线性稳压器 (LDO)

8.6.1 简介

WS51F7030 系列芯片内置了一个 LDO，用于给内核供电，其典型外围电路如下图。

图 8.6.1-1 典型供电电路图



8.6.2 内部基准及输出电压校准

LDO 内部的基准和 LDO 输出电压都需要校准，分别通过 BGP_TRIM(PWCON[2:0])和 LDO_TRIM(PWCON[5:3])校正，在 NVR3 区（见表 7.2.2-1）填写校准值后上电时芯片会自动从 NVR3 区导入校准值，校正后基准为 1.0V，电压为 1.5V。当 VDD 小于 LDO_TRIM 位设定的输出电压时，LDO 直接输出 VDD；当 VDD 大于设定电压时，LDO 输出设定的电压。

8.6.3 LDO 工作模式

LDO 有三种不同的工作模式：高功耗模式、低功耗模式和小负载模式，其中小负载模式芯片具有最低系统功耗，高功耗模式芯片具有最强抗干扰能力。

表 8.6.3-1 LDO 工作模式

| 工作模式 | 何时开启 |
|-------|---|
| 高功耗模式 | 正常工作 |
| | 仿真调试模式 |
| | IDLE/STOP 模式时开启了 HRC |
| | IDLE/STOP 模式时寄存器 LPMD 为 0 |
| 低功耗模式 | 芯片进入 IDLE/STOP 模式时，如果高功耗模式所有条件不成立则进入低功耗模式 |
| 小负载模式 | 在低功耗模式如果寄存器 SPMD 为 1 则进入小负载模式 |

8.6.4 寄存器描述

表 8.6.4-1 寄存器 PWCON

| F9H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|----------|-------------------------------|----------|---|---|----------|---|---|
| PWCON | LPMD | SPMD | LDO_TRIM | | | BGP_TRIM | | |
| R/W | R/W | R/W | R/W | | | R/W | | |
| 初始值 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | LPMD | 在 IDLE/STOP 模式时 LDO 低功耗模式开启信号 | | | | | | |
| 6 | SPMD | 在 IDLE/STOP 模式时 LDO 小负载模式开启信号 | | | | | | |
| 5-3 | LDO_TRIM | LDO 校正参数 | | | | | | |
| 2-0 | BGP_TRIM | Bandgap 校正参数 | | | | | | |

前海维晟 WISESUN

8.7 低电压检测 (LVD)

8.7.1 简介

低电压检测 (LVD) 用于监控芯片自身的供电电压 VDD，可设置检测电压范围为 1.8V-4.2V，精度±50mV。内含去抖功能。当 VDD 小于所设定的电压值时，可设置触发中断或复位。

8.7.2 功能描述

配置寄存器 LVDS 不为 0 时开启 LVD 功能，其触发电压阈值由寄存器 LVDTH 设置。

当芯片 VDD 出现由高变低到 LVDTH 设置的阈值时，LVDF 位将置 1。此时，如果 LVDS 为 1 则产生 LVD 中断，如果 LVDS 为 2 则产生 LVD 复位，如果 LVDS 为 3 则芯片即不复位也不产生中断，此时通过 LVDF 标志可初步判断芯片电源电压。如果产生了 LVD 复位，不会复位寄存器 LVDCON。

8.7.3 寄存器描述

表 8.7.3-1 寄存器 LVDCON

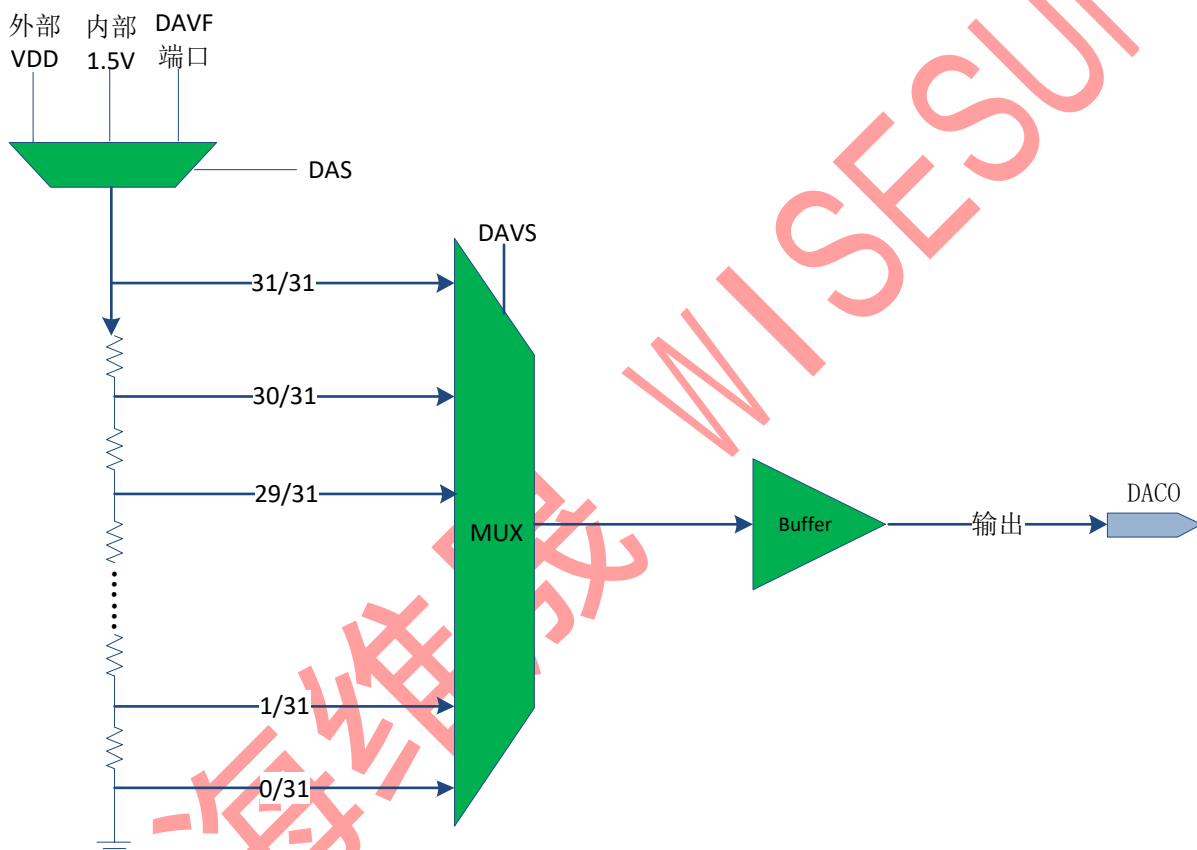
| B9H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-------|--|---|---|-------|---|---|------|
| LVDCON | LVDS | | - | - | LVDTH | | | LVDF |
| R/W | R/W | R/W | - | - | R/W | | | R/W |
| 初始值 | 0 | 0 | - | - | 1 | 1 | 1 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-6 | LVDS | LVD 功能选择位 00: LVD 关闭 01: LVDF 为高时，产生中断 10: LVDF 为高时，产生复位 11: LVDF 为高时，不产生中断和复位 | | | | | | |
| 5-4 | - | - | | | | | | |
| 3-1 | LVDTH | LVD 检测电压设置位域，当 VDD 电压小于设置值时，置位低电压标志 LVDF 000: 1.8V 001: 2.0V 010: 2.4V 011: 2.8V 100: 3.0V 101: 3.4V 110: 3.7V 111: 4.2V | | | | | | |
| 4 | LVDF | LVD 产生标志位 0: LVD 未检测到低电压 1: LVD 检测到低电压 | | | | | | |

8.8 数模转换器 (DAC)

8.8.1 简介

WS51F7030 集成了一个 5 位可编程数模转换器 (DAC)。DAC 的输入参考可以选择为来自芯片外部电源、内部电压基准或者外部端口 DAVF 输入。DAC 的输出可作为内部比较器 CMP 的输入参考,也可以直接输出至芯片的外部引脚 DACO 作为外部参考使用。DAC 内部结构如下所示。

图 8.8.1-1 DAC 原理结构图



8.8.2 功能描述

DAC 模块通过 31 个大小相等的电阻串联形成均匀的 32 级分压,其电压源可选外部电源 VDD、内部运放输出的 1.5V 电压或外部 DAVF 端口。使用时通过 DAVS 选择电压大小,该电压经过一个 buffer 输出到外部端口 DACO,可驱动 3mA 的电流。

备注:

选择输出内部 1.5V 基准电压时芯片无法输出,使用时请注意。

8.8.3 寄存器描述
表 8.8.3-1 寄存器 DACON

| C3H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|------|--|---|------|---|---|---|---|
| DACON | DAS | | - | DAVS | | | | |
| R/W | R/W | R/W | - | R/W | | | | |
| 初始值 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-6 | DAS | DA 开启并选择基本电压 00: 关闭 DA 01: 内部 1.5 V 基准 10: 外部电源 VDD 11: 外部 DAVF 输入 | | | | | | |
| 5 | - | - | | | | | | |
| 4-0 | DAVS | DA 输出电压为 (基准电压*DAVS/31) | | | | | | |

8.9 触摸按键 (Touch Key)

8.9.1 简介

WS51F7030 系列芯片内置了最大 12 通道的电荷迁移触摸按键模块,它具有优越的抗干扰性能和低功耗特性,其特点如下:

- 高可靠性,可通过 10V 动态 CS 测试及 EFT 4.5KV 测试
- 超低功耗,单键功耗低至 4uA,典型情况下每增加一个按键平均功耗增加 0.3uA
- 支持待机状态触摸唤醒功能
- 支持触摸引脚与 LED 引脚共用
- 高灵活度开发软件库支持,低开发难度
- 自动化调试软件支持,极大的方便用户开发

用户在应用时需 CMOD 端口外接一个 10nF-47nF 电容,电容精度 10%以内,建议使用涤纶电容、X7R 材质电容或 NPO 材质贴片电容。该电容可直接影响触摸灵敏度,电容容值越小,灵敏度越低,容值越大,灵敏度越高。

关于触摸按键的使用,请参考 WS51F7030 应用软件库及相关参考程序

8.9.2 工作模式

触摸按键模块有三种工作模式:普通模式、待机工作模式、触摸唤醒模式。

- 普通模式芯片处于正常工作状态
- 待机工作模式芯片处于待机状态,可根据需要设置触摸扫描时间,典型时间设置为 250ms
- 触摸唤醒模式芯片处于待机状态,此时有手指触摸时芯片才会恢复工作,用户可根据需要设置触摸扫描时间

8.10 软件 LED 驱动 (SLED)

8.10.1 简介

WS51F7030 系列芯片内置了一个软件 LED 驱动，该驱动可独立使用，也可与 Touch Key 模块同时使用。当 IO 端口配置为高阻时，LSPE0/LSPE1 为 1 时对应的 LED IO 打开。当 IO 端口配置为 Touch Key 功能时，LED 与 Touch Key 共用引脚。

LED 共用 12 个引脚，在使用时采用轮流选择 COM 口的方式进行 LED 扫描，从而最大可驱动 11seg*12com 点阵的 LED 屏。

8.10.2 LED 显示原理

设置 LSE 为 1 开启 LED 扫描功能。LED 扫描时先通过寄存器 LSCOM 选定一个引脚作为 COM 口，设置寄存器 LSDAT，而后扫描接在该 COM 口上的一串 LED 灯珠，扫描完后再选定另一个引脚作为 COM 口并设置 LSDAT，再扫描此 COM 口上的另一串 LED 灯珠...，以此类推，扫描完所有的 COM 口后即为一帧 LED 图像，其中寄存器 LSDAT 对应每串 LED 灯珠的亮灭配置选择。在使用时一般要求 LED 显示不出现闪屏，所以 LED 帧频要达到 64Hz，因此在配置 LED 时需准确计算每串 LED 的显示时间和熄灭时间。LED 固定使用内部 LRC 时钟，寄存器 LSDTM 和 LSPTM 为 LED 显示时间和熄灭时间配置寄存器，其计算方法如下。

LED 显示时间 = LSDTM * LRC 时钟周期(≈32KHz, 即≈ 31.25us)

LED 熄灭时间 = LSPTM * LRC 时钟周期(≈32KHz, 即≈31.25us)

备注:

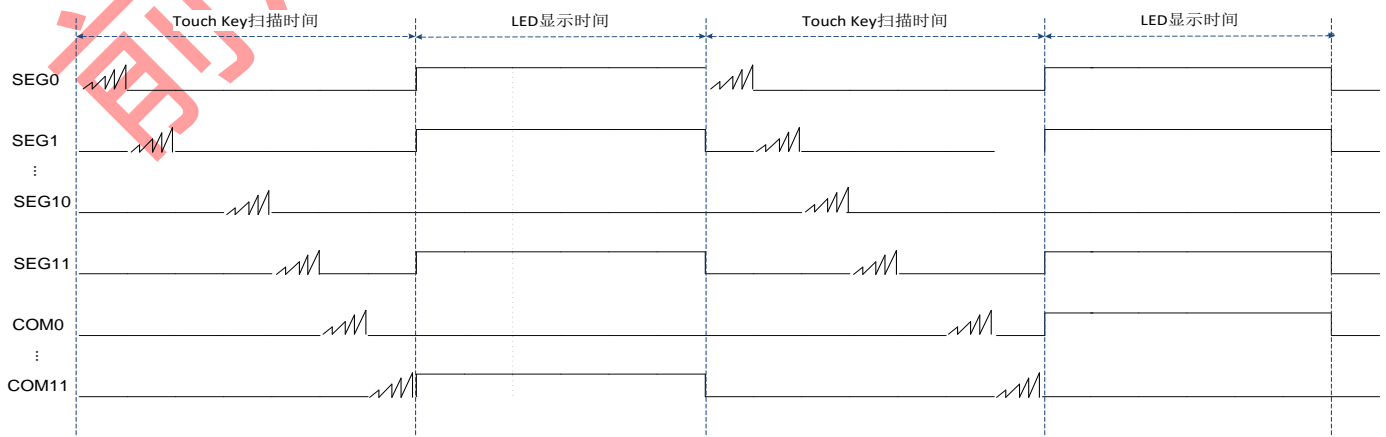
LSDTM 与 LSPTM 的和不要小于 2，否则无法产生中断标志

8.10.3 LED 与 Touch Key 共用实现方法

LED 与 Touch Key 可共用引脚并同时使用，其实现方法为在 LED 熄灭时间软件开启触摸按键扫描功能。

LED 点亮时间结束时硬件拉高 LDF 标志，通知软件可以开启触摸按键扫描功能。触摸按键模块必须在设定的 LED 熄灭时间内完成所有触摸通道的扫描，否则 LED 熄灭时间结束后马上进入 LED 点亮时间，会产生 IO 引脚功能冲突，因此在 LED 和 Touch Key 共用时需要调整 Touch Key 扫描时间和 LED 熄灭时间，直到完全满足 LED 显示和 Touch Key 按键扫描需要。具体使用方法请参考 WS51F7030 系列 MCU 应用手册及相关应用软件库。

图 8.10.3-1 LED 与 Touch Key 共用波形



8.10.4 寄存器描述
表 8.10.4-1 寄存器 LSCON

| 8290H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|------|--|------|------|---|-----|-----|-----|
| LSCON | LSE | LDIE | LPIE | LTIE | - | LDF | LPF | LTF |
| 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 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7 | LSE | LED 显示使能信号 | | | | | | |
| 6 | LDIE | LDF 中断使能信号 | | | | | | |
| 5 | LPIE | LPF 中断使能信号 | | | | | | |
| 4 | LTIE | LTF 中断使能信号 | | | | | | |
| 3 | - | - | | | | | | |
| 2 | LDF | LED 点亮时间结束标志 | | | | | | |
| 1 | LPF | LED 熄灭时间结束标志 | | | | | | |
| 0 | LTF | Touch Key 扫描阶段时间设置过短标志 0: 表示用户设置的时间足够触摸按键扫描的时间 1: 表示用户设置的时间不足, 定时器计数完触摸通道没扫描完 备注: 此位为硬件自动置位和自动清零位, 考虑到用户使用的便捷性, 此位也可以用软件对其写 0 清 0。 | | | | | | |

表 8.10.4-2 寄存器 LSCFG

| 8291H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-------|---|-------|---|---|---|-------|-------|
| LSCFG | LSGVS | | LSMVS | | - | - | LSEGV | LCOMV |
| R/W | R/W | | R/W | | - | - | R/W | R/W |
| 初始值 | 0 | 0 | 0 | 0 | - | - | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-6 | LSGVS | LED 熄灭阶段, SEG 引脚电平选择 00/11: 高阻 01: 输出高 10: 输出低 | | | | | | |
| 5-4 | LSMVS | LED 熄灭阶段, COM 引脚电平选择 00/11: 高阻 01: 输出高 10: 输出低 | | | | | | |
| 3-2 | - | - | | | | | | |
| 1 | LSEGV | LED 点亮阶段, SEG 有效电平选择 0: 低电平 1: 高电平 | | | | | | |
| 0 | LCOMV | LED 点亮阶段, COM 有效电平选择 | | | | | | |

| | | |
|--|--|------------------|
| | | 0: 低电平 1: 高电平 |
|--|--|------------------|

表 8.10.4-3 寄存器 LSDTMn

| 8293H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------|------------|------------|---|---|---|---|------------|---|
| LSDTM0 | LSDTM[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8294H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LSDTM1 | - | - | - | - | - | - | LSDTM[9:8] | |
| R/W | - | - | - | - | - | - | R/W | |
| 初始值 | - | - | - | - | - | - | 1 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 9-0 | LSDTM | LED 显示时间设置 | | | | | | |

表 8.10.4-4 寄存器 LSPTMn

| 8295H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------|------------|------------|---|---|---|---|------------|---|
| LSPTM0 | LSPTM[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8296H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LSPTM1 | - | - | - | - | - | - | LSPTM[9:8] | |
| R/W | - | - | - | - | - | - | R/W | |
| 初始值 | - | - | - | - | - | - | 1 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 9-0 | LSPTM | LED 熄灭时间设置 | | | | | | |

表 8.10.4-5 寄存器 LSCOM

| 8297H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------|------------|---|---|---|-------|---|---|---|
| LSCOM | - | - | - | - | LSCOM | | | |
| R/W | - | - | - | - | R/W | | | |
| 初始值 | - | - | - | - | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-4 | - | - | | | | | | |
| 3-0 | LSCOM | LED 的 COM 选择寄存器 0000: 选择 TK0 引脚 0001: 选择 TK1 引脚 0010: 选择 TK2 引脚 | | | | | | |

| | | |
|--|--|-------------------------------|
| | | 1011: 选择 TK11 引脚 other: 保留 |
|--|--|-------------------------------|

表 8.10.4-6 寄存器 LSPE_n

| 8298H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-----------|-------------------------------------|---|---|------------|---|---|---|
| LSPE0 | LSPE[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8299H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LSPE1 | - | - | - | - | LSPE[11:8] | | | |
| R/W | - | - | - | - | R/W | | | |
| 初始值 | - | - | - | - | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 11-0 | LSPE | LED 驱动和 Touch Key 共用 Touch Key 引脚选择 | | | | | | |

表 8.10.4-7 寄存器 LSDAT_n

| 82A0H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|------------|-------------------------------|---|---|-------------|---|---|---|
| LSDAT0 | LSDAT[7:0] | | | | | | | |
| R/W | R/W | | | | | | | |
| 初始值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 82A1H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LSDAT1 | - | - | - | - | LSDAT[11:8] | | | |
| R/W | - | - | - | - | R/W | | | |
| 初始值 | - | - | - | - | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 11-0 | LSDAT | LED 的 SEG 输出值,分别对应 SEG11-SEG0 | | | | | | |

8.11 软件 LCD 驱动 (SLCD)

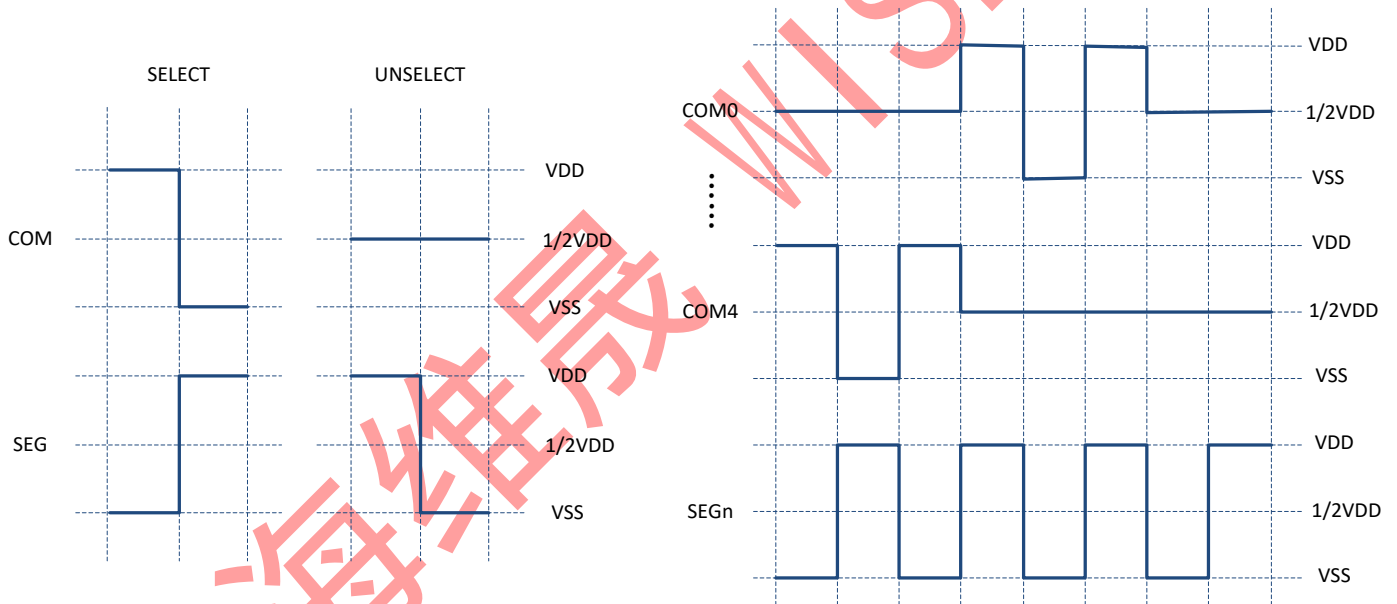
8.11.1 简介

WS51F7030 系列芯片的 P10-P14 可作为软件 LCD 的 COM 口, 这些 IO 可作为普通 IO 功能之外, 还可输出 1/2 VDD 电压。用户可根据使用情况, 选择相应 IO 作为 LCD 驱动的 COM 口, 此时对应的 IO 功能选择为 GPIO 输出, 见表 7.4.2-1 描述。

8.11.2 1/2 偏压软件 LCD

如下图所示 LCD 的 1/2 偏压输出波形, 其中 SEG 输出为 VDD 和 VSS 交替翻转。如果对应的 LCD 段码点为 SELECT, 则对应的 COM 输出为 SEG 的反相, 即 VSS 和 VDD 交替翻转。如果对应的 LCD 段码点为 UNSELECT, 则对应的 COM 输出为 1/2 VDD。因此只要对应的 COM 端口可以输出 1/2 VDD, 即可通过软件实现点亮 1/2 偏压 LCD 段码屏。

图 8.11.2-1 LCD1/2 偏压输出波形



8.11.3 寄存器描述

表 8.11.3-1 寄存器 PCOMS

| E4H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|--------|-------------------|---|--------|--------|--------|--------|--------|
| PCOMS | - | - | - | P4COMS | P3COMS | P2COMS | P1COMS | POCOMS |
| R/W | - | - | - | R/W | R/W | R/W | R/W | R/W |
| 初始值 | - | - | - | 0 | 0 | 0 | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-5 | - | | | | | | | |
| 4 | P4COMS | P14 口驱动 1/2VDD 选择 | | | | | | |

| | | |
|---|--------|---|
| | | 0: 普通 IO 1: 输出 1/2VDD |
| 3 | P3COMS | P13 口驱动 1/2VDD 选择 0: 普通 IO 1: 输出 1/2VDD |
| 2 | P2COMS | P12 口驱动 1/2VDD 选择 0: 普通 IO 1: 输出 1/2VDD |
| 1 | P1COMS | P11 口驱动 1/2VDD 选择 0: 普通 IO 1: 输出 1/2VDD |
| 0 | POCOMS | P10 口驱动 1/2VDD 选择 0: 普通 IO 1: 输出 1/2VDD |

表 8.11.3-2 寄存器 PDRVS

| E5H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-------|---|---|---|---|---|-------|---|
| PDRVS | - | - | - | - | - | - | PDRVS | |
| R/W | - | - | - | - | - | - | R/W | |
| 初始值 | - | - | - | - | - | - | 0 | 0 |
| 位编号 | 位符号 | 说明 | | | | | | |
| 7-2 | - | - | | | | | | |
| 1-0 | PDRVS | COM 口驱动 1/2VDD，分压电阻大小选择 00: 关闭分压电阻 01: 设定分压电阻为 12.5KΩ 10: 设定分压电阻为 37.5KΩ 11: 设定分压电阻为 87.5KΩ | | | | | | |

9 程序下载和仿真

WS51F7030 系列芯片使用我司自主研发的工具 WS_LINK 进行下载及仿真。该工具集成了在线下载、离线下载、在线仿真、USB 转 I2C 等功能，配合我司的 IDE 软件 WS_IDE、触摸调试工具 WS_TK_Tuning 及芯片软件库，可以极大的方便用户的使用。

备注：关于程序下载和仿真的流程及细节请参考“WISESUN 开发工具使用说明”

9.1 程序下载

通过 I2C 调试接口下载，接口为 P34（SDA）和 P33（SCL）。

用户在调试程序时，如果用到了 I2C 接口使用的 2 个端口，需要在程序开头加一定延迟，避免下次无法下载，请参考 WS51F7030 相关例程软件库。

9.2 在线仿真

WS51F7030 系列芯片支持在线仿真，与 Keil 软件完全兼容，支持无限断点。芯片与仿真器之间通过 I2C 接口进行通信，出厂默认的 I2C 接口是 P34(SDA) 和 P33(SCL)。

仿真调试器任何时候都可以进入和退出仿真模式，并且不受系统时钟的影响，另外在芯片进入 IDLE 或 STOP 模式时也能正常进行在线仿真。

芯片与仿真器间通过 I2C 通信，所以在做在线仿真时 I2C 端口不能设置为其他功能，并且应用程序里不能使用 I2C 功能，否则将无法进入仿真模式。

10 电气特性

10.1 极限参数

| 参数 | 最小值 | 最大值 | 单位 |
|------------|------|---------|-----|
| 直流供电电压 | -0.3 | 6 | V |
| I/O 端口输入电压 | -0.3 | VDD+0.3 | V |
| 工作环境温度 | -40 | 105 | °C |
| 储存温度 | -45 | 125 | °C |
| CPU 工作频率 | - | 16 | MHz |
| 最大通过电流 | - | 600 | mA |

备注：超过“极限参数”范围有可能对芯片造成损坏，无法预期芯片在上述范围外的工作状态，若长期在标示范围外工作，可能会影响芯片的可靠性。

10.2 直流电气特性

最小值的数据测量条件：VDD=1.8V, TA=25°C，除非另有说明。

典型值的数据测量条件：VDD=3.3V, TA=25°C，除非另有说明。

最大值的数据测量条件：VDD=5.5V, TA=25°C，除非另有说明。

| 芯片参数 | 符号 | 工作电压 | 最小值 | 典型值 | 最大值 | 单位 | 条件 | |
|---------------|-------|------|-----|------|-----|----|--|---|
| 工作电流 | Iop1 | 1.8V | - | 0.98 | - | mA | 系统时钟为 HRC (16MHz)，其他时钟关闭，所有输出端口无负载，LDO 设置为高功耗模式，所有 IO 端口处于高阻态，所有外设关闭，CPU 执行 NOP 指令 | |
| | | 3.3V | - | 1.00 | - | mA | | |
| | | 5.5V | - | 1.00 | - | mA | | |
| | Iop2 | 1.8V | - | 65 | - | uA | | 系统时钟为 LRC(32KHz)，其他时钟关闭，所有输出端口无负载，LDO 设置为低功耗模式，所有 IO 端口处于高阻态，所有外设关闭，CPU 执行 NOP 指令 |
| | | 3.3V | - | 75 | - | uA | | |
| | | 5.5V | - | 75 | - | uA | | |
| STOP 模式 电流 | Istp1 | 1.8V | - | 370 | - | uA | 所有时钟关闭，所有输出端口无负载，所有数字输入端口不浮动，所有外设关闭，LDO 设置为低功耗模式，Flash 进入睡眠模式，CPU 进入 STOP 模式。 | |
| | | 3.3V | - | 370 | - | uA | | |
| | | 5V | - | 370 | - | uA | | |
| | Istp2 | 1.8V | 1.5 | 2.0 | 2.5 | uA | | |
| | | 3.3V | 1.5 | 2.0 | 2.5 | uA | | |
| | | 5V | 1.5 | 2.2 | 3.0 | uA | | |
| IDLE 模式 电流 | Iidl1 | 1.8V | - | 370 | - | uA | 系统时钟设为 HRC (16MHz)，其他时钟关闭，所有输出端口无负载，所有数字输入端口不浮动，所有外设关闭，LDO 处于高功耗模式，Flash 进入睡眠模式， | |
| | | 3.3V | - | 370 | - | uA | | |
| | | 5V | - | 370 | - | uA | | |

| | | | | | | | |
|------------|-------|------|---|-----|---|----|---|
| | | | | | | | CPU 进入 IDLE 模式。 |
| | lidl2 | 1.8V | - | 410 | - | uA | 系统时钟设为 LRC (32KHz), 其他时钟关闭, 所有输出端口无负载, 所有数字输入端口不浮动, 所有外设关闭, LDO 设置为低功耗模式, CPU 进入 IDLE 模式。 |
| | | 3.3V | - | 405 | - | uA | |
| | | 5V | - | 405 | - | uA | |
| | lidl3 | 1.8V | - | 400 | - | uA | 系统时钟设为 HRC (16MHz), 其他时钟关闭, 所有输出端口无负载, 所有数字输入端口不浮动, 所有外设关闭, LDO 设置为小负载模式, Flash 进入睡眠模式, CPU 进入 IDLE 模式。 |
| | | 3.3V | - | 400 | - | uA | |
| | | 5V | - | 400 | - | uA | |
| | lidl4 | 1.8V | - | 400 | - | uA | 系统时钟设为 LRC (32KHz), 其他时钟关闭, 所有输出端口无负载, 所有数字输入端口不浮动, 所有外设关闭, LDO 设置为小负载模式, CPU 进入 IDLE 模式。 |
| | | 3.3V | - | 400 | - | uA | |
| | | 5V | - | 400 | - | uA | |
| IO 端口输入高电压 | Vhi | 1.8V | | | | V | VDD=1.8V-5.5V |
| | | 3.3V | | | | V | |
| | | 5V | | 1.6 | | V | |
| IO 端口输入低电压 | Vlo | 1.8V | | | | V | VDD=1.8V-5.5V |
| | | 3.3V | | | | V | |
| | | 5V | | 1.2 | | V | |
| IO 端口推电流 | lpu | | - | 15 | - | mA | VDD=5V Vpin=4.3V |
| IO 端口灌电流 | lol | | - | 31 | - | mA | VDD=5V Vpin=0.4V |
| IO 端口灌电流 | lol | | - | 56 | - | mA | VDD=5V Vpin=0.8V |
| IO 端口下拉电阻 | Rd | | - | 15 | - | KΩ | - |
| IO 端口强上拉电阻 | Rus | | - | 10 | - | KΩ | - |
| IO 端口弱上拉电阻 | Ruw | | - | 65 | - | KΩ | - |

10.3 交流电气特性

交流电气特性 (VDD=1.7-5.5V, TA=25°C, 除非其它说明)

| 芯片参数 | 最小值 | 典型值 | 最大值 | 单位 | 条件 |
|-------------------|------|-----|------|----|---------------|
| 上电复位时间 | 4.5 | 5 | 8 | ms | |
| 待机唤醒时间 | 12.6 | 13 | 13.8 | us | |
| 内部低速时钟 (LRC) 起振时间 | 250 | 340 | 450 | us | LRC 频率为 32KHz |
| 内部高速时钟 (HRC) 起振时间 | 57 | 62 | 67 | us | HRC 频率为 16MHz |
| 复位脉冲时间 | - | 0.5 | - | us | |

备注: VDD=3.3V,TA=25°C, LRC 出厂频率为 32KHz, HRC 出厂频率为 16MHz, 精度为±1%.

10.4 POR 电气特性

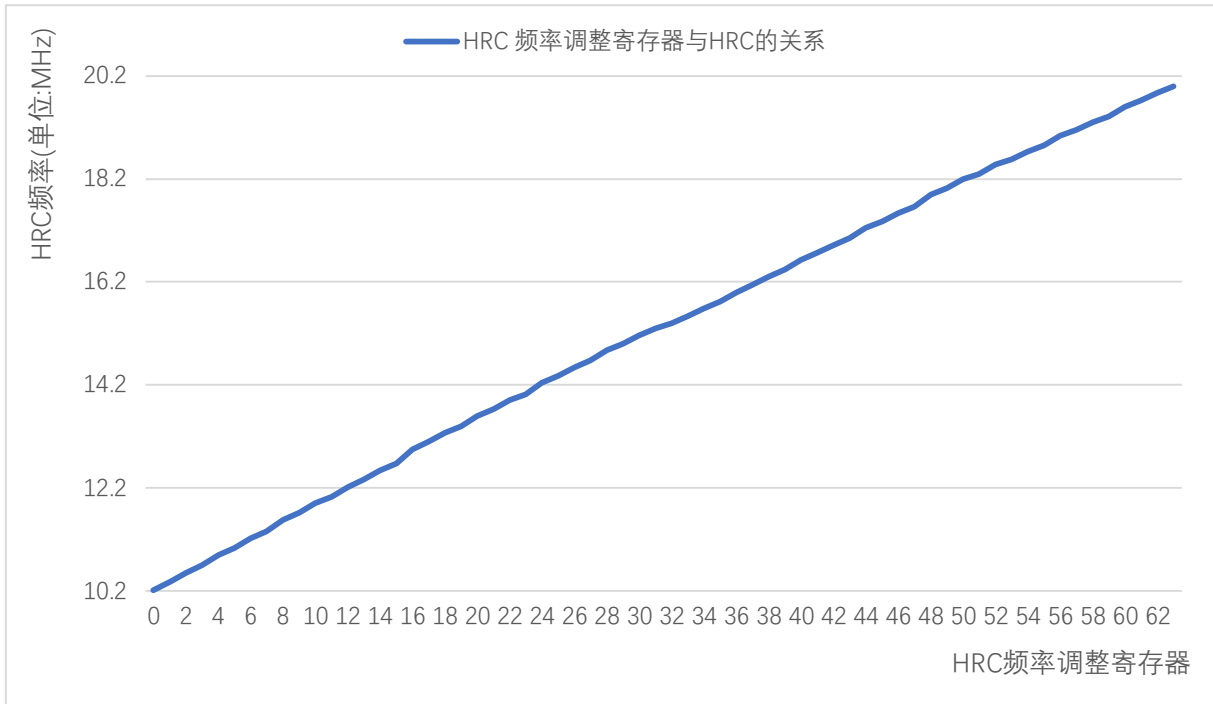
| 芯片参数 | 符号 | 最小值 | 典型值 | 最大值 | 单位 | 条件 |
|---------------|------|-----|-----|-----|----|----|
| 上电复位截止电压值 | Vpor | - | 1.6 | - | V | - |
| 掉电复位起始电压值 | Vbor | - | 1.4 | - | V | - |
| 上电复位截止后持续复位时间 | Tpor | - | 70 | - | us | - |

10.5 内部 HRC 电气特性

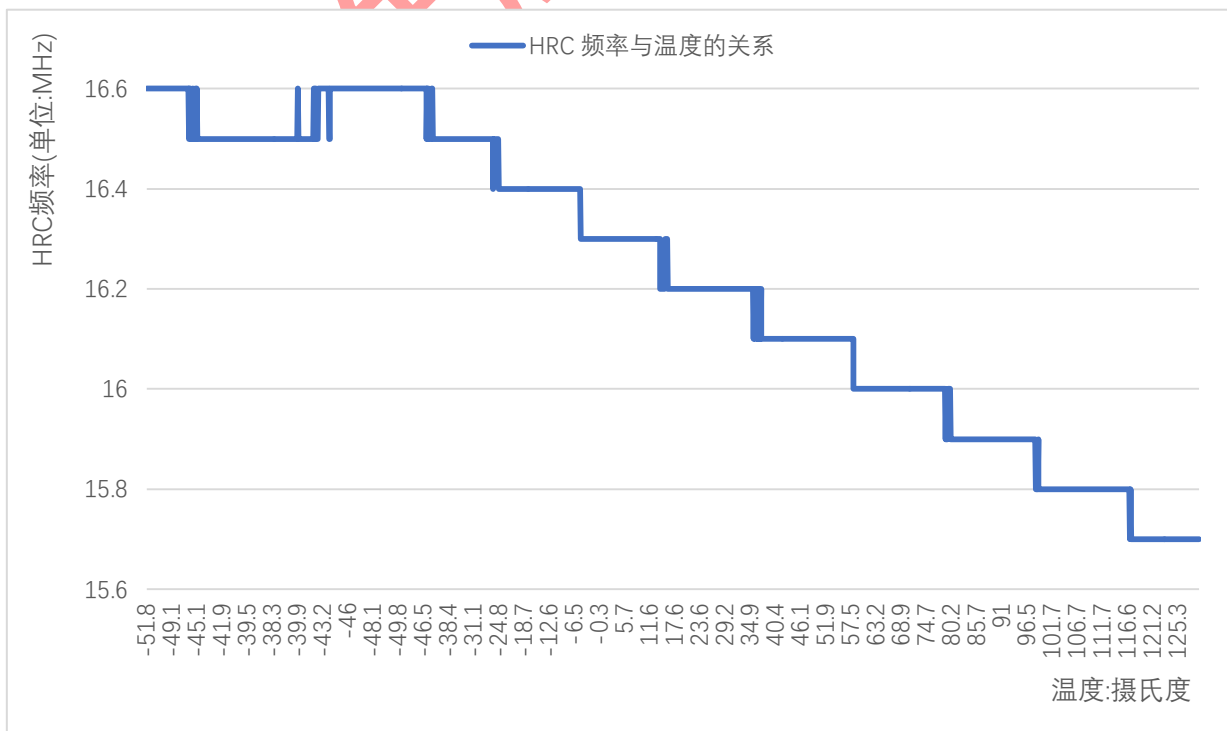
测试条件：VDD=5V，Ta=25℃。

测试数据为少数芯片测试结果，基本为测试到的最差情况，仅供参考。

内部 HRC 线性曲线



内部 HRC 温度特性

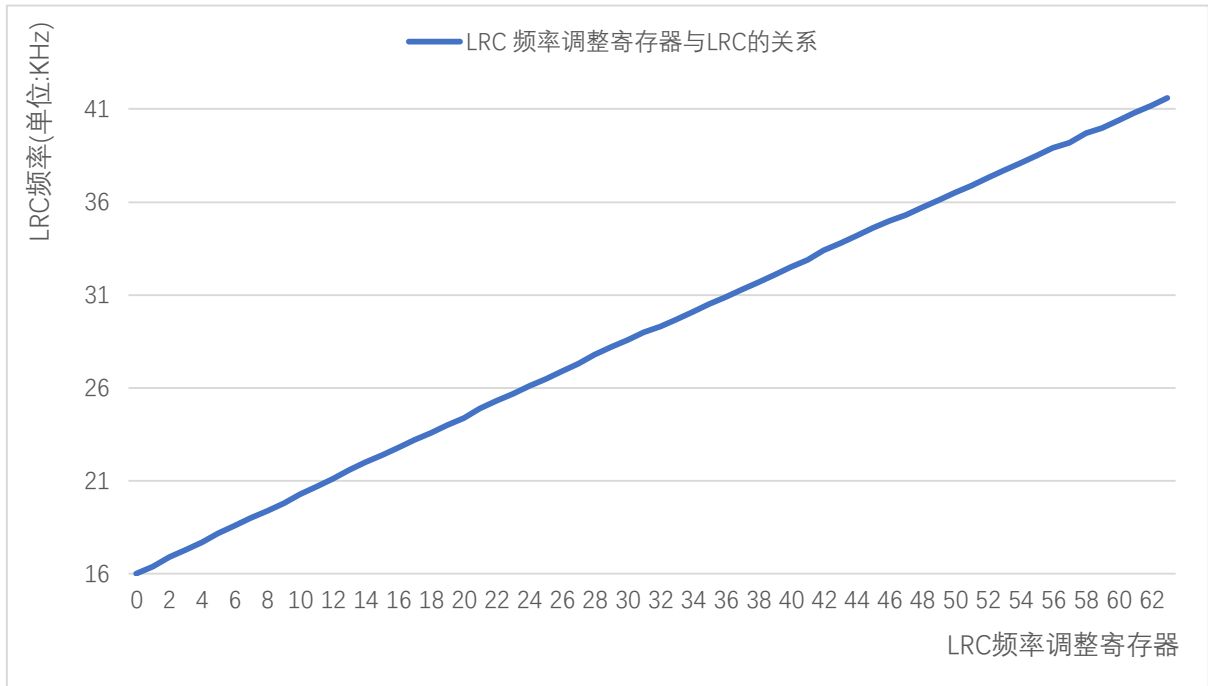


10.6 内部 LRC 电气特性

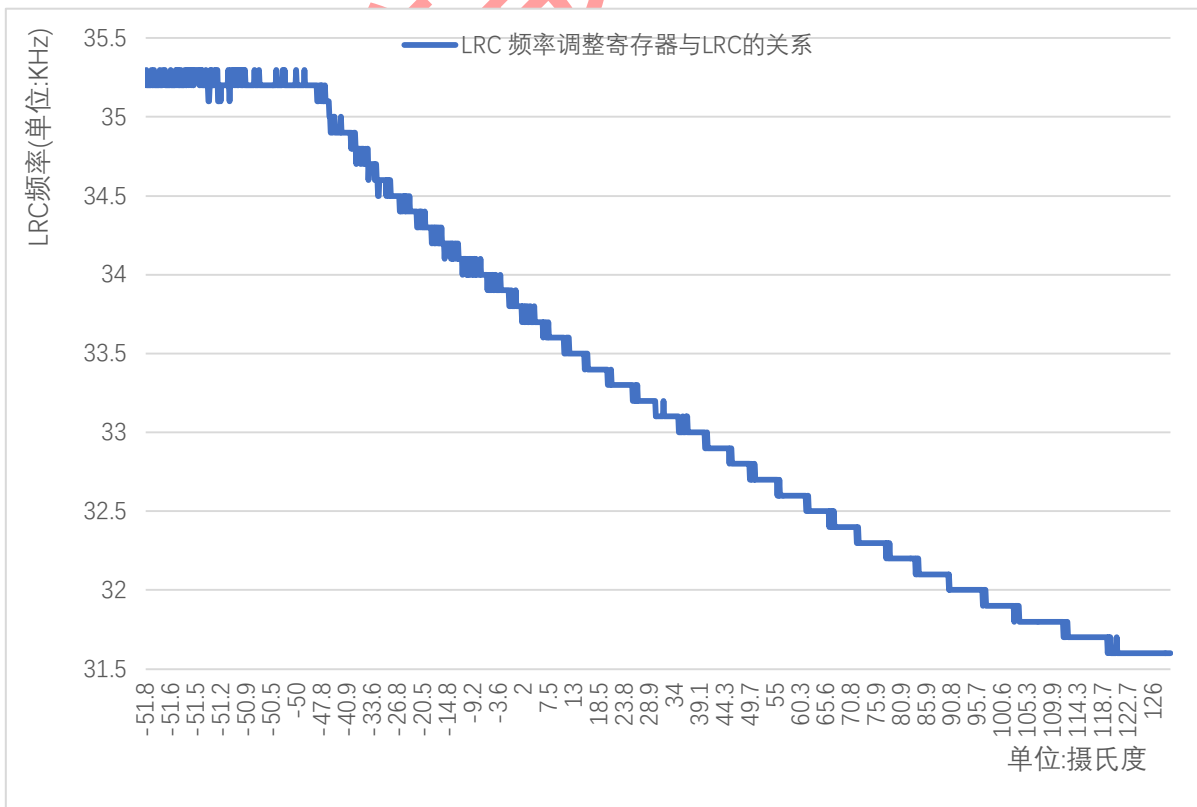
测试条件：VDD=5V，Ta=25°C。

测试数据为少数芯片测试结果，基本为最差测试情况，仅供参考。

内部 LRC 线性曲线



内部 LRC 温度特性



10.7 LVD 电气特性

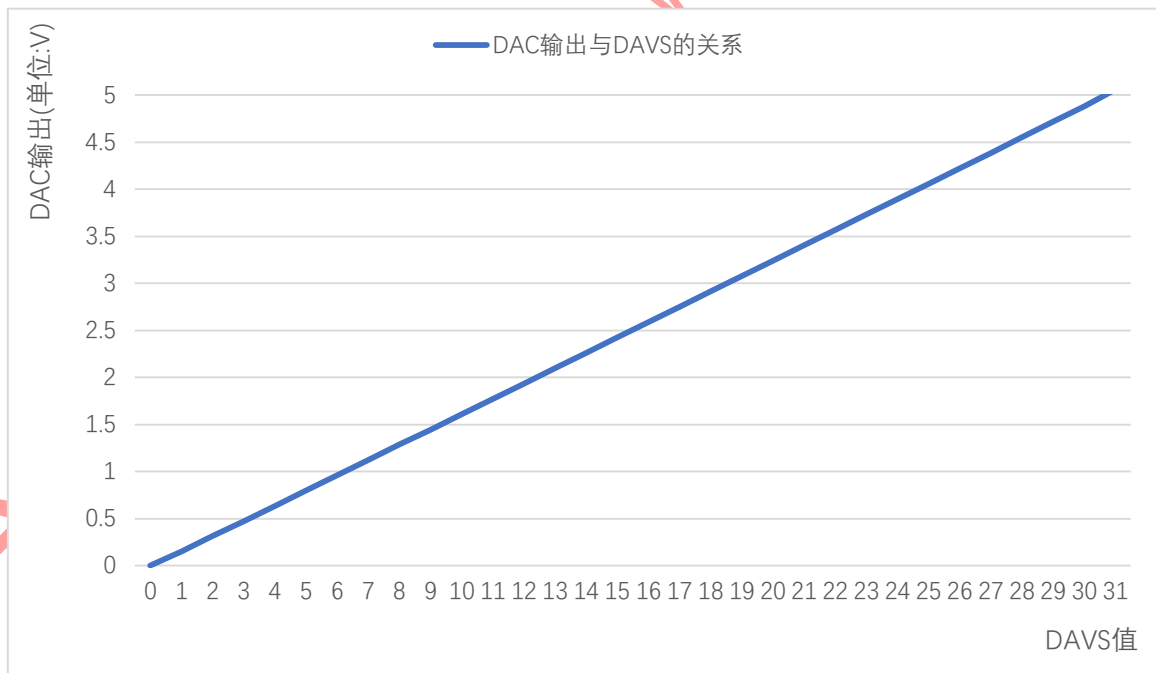
测试条件: $T_a=25^{\circ}\text{C}$, 内部基准准确校准到 0.8V

| 芯片参数 | 最小值 | 典型值 | 最大值 | 单位 | 条件 |
|---------------|------|------|------|----|-----------|
| LVD 门限电压 1.8V | 1.80 | 1.80 | 1.80 | V | LVDTH=000 |
| LVD 门限电压 2.0V | 1.99 | 1.99 | 2.00 | V | LVDTH=001 |
| LVD 门限电压 2.4V | 2.39 | 2.39 | 2.40 | V | LVDTH=010 |
| LVD 门限电压 2.8V | 2.77 | 2.78 | 2.79 | V | LVDTH=011 |
| LVD 门限电压 3.0V | 2.98 | 2.98 | 2.99 | V | LVDTH=100 |
| LVD 门限电压 3.4V | 3.35 | 3.36 | 3.38 | V | LVDTH=101 |
| LVD 门限电压 3.7V | 3.65 | 3.67 | 3.68 | V | LVDTH=110 |
| LVD 门限电压 4.2V | 4.17 | 4.18 | 4.20 | V | LVDTH=111 |

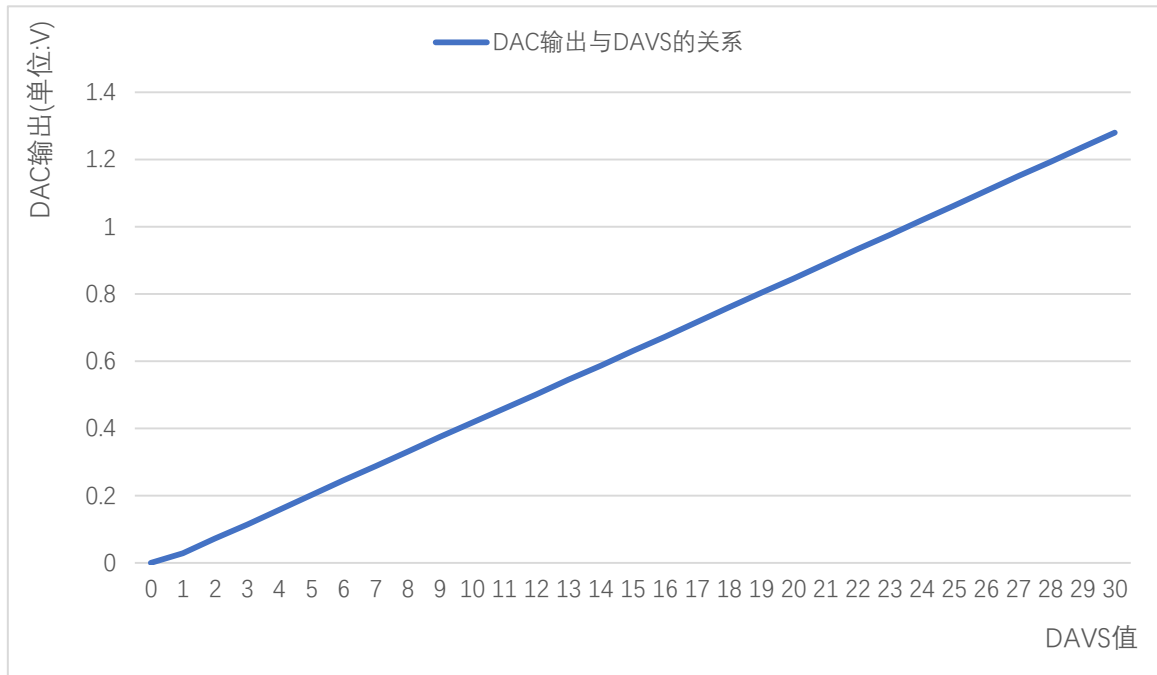
10.8 DAC 电气特性

测试条件: $V_{DD}=5\text{V}$, $T_a=25^{\circ}\text{C}$, $V_{d\text{avf}}=5\text{V}$ 。

基准电压为 VDD5 时 DAC 输出曲线



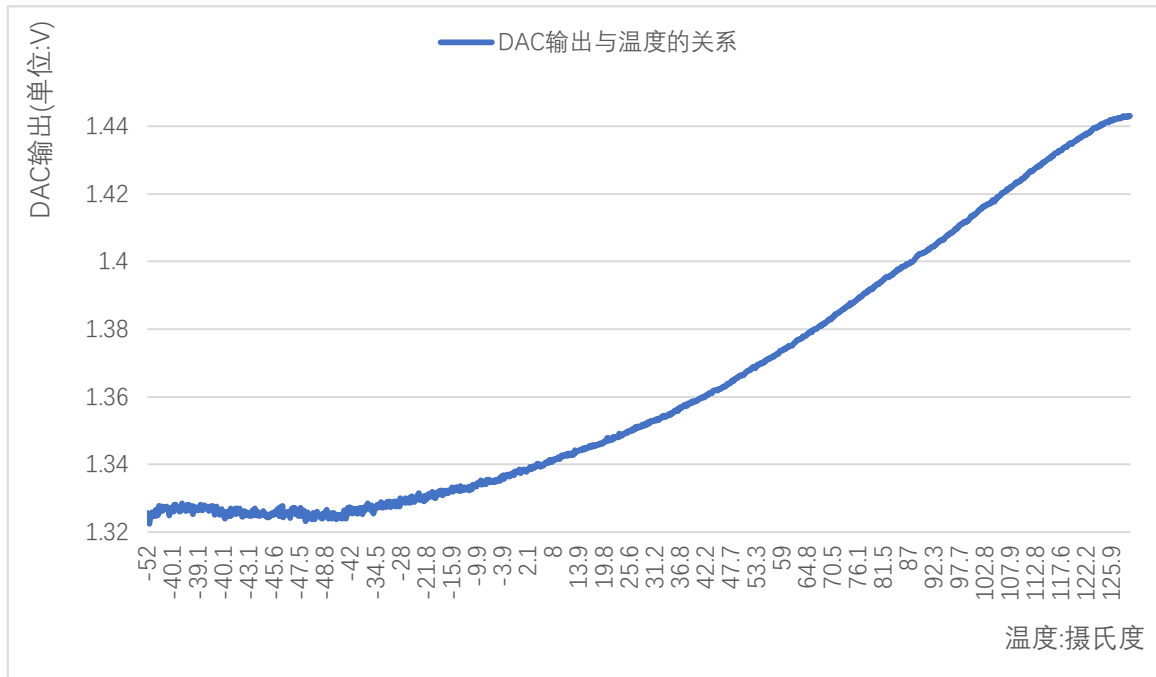
基准电压为内部 1.5V 时 DAC 输出曲线



基准电压为外部 DAVF(2.5V)时 DAC 输出曲线



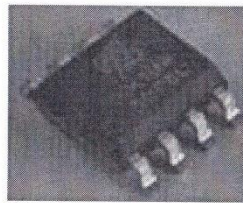
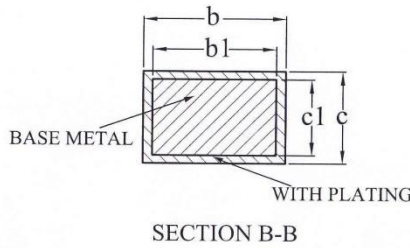
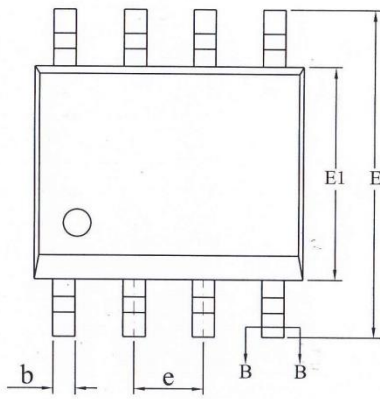
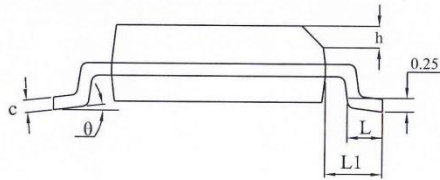
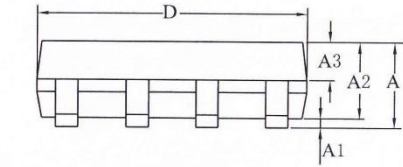
基准电压为内部 1.5V 时 DAC 输出随温度变化曲线



前海维晟

11 封装信息

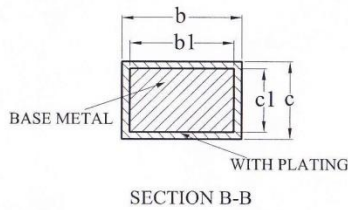
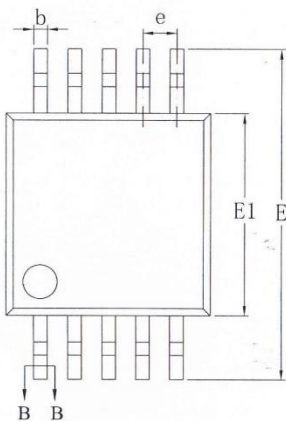
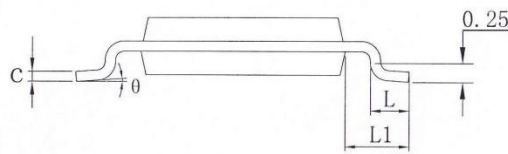
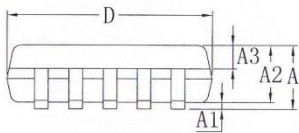
11.1 WS51F7030S08U (SOP8)



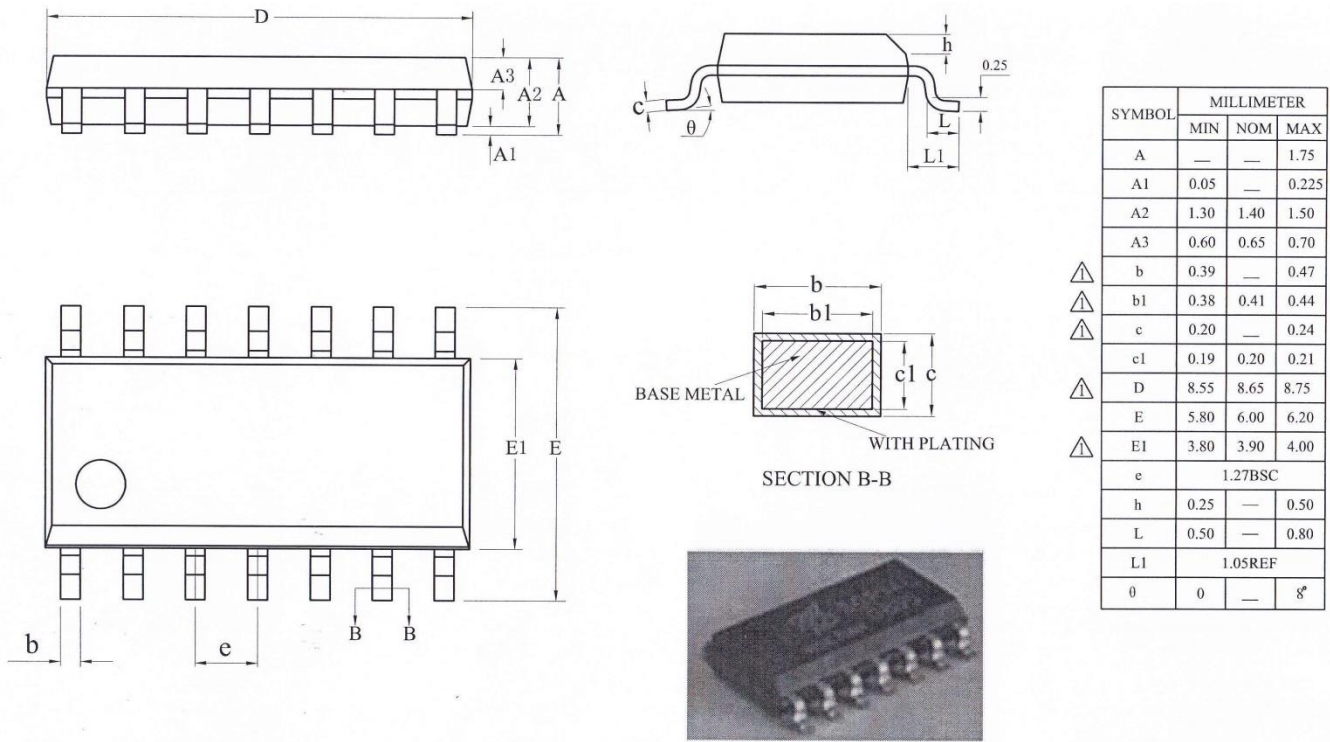
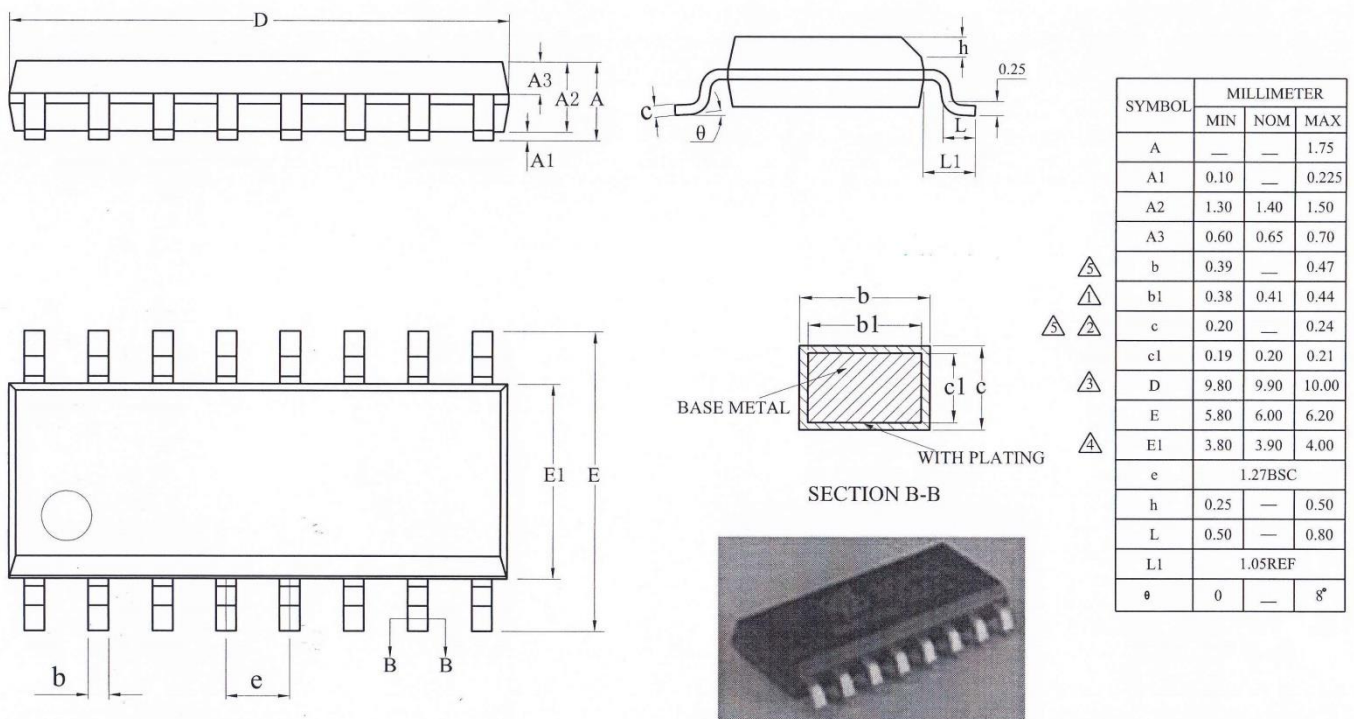
- ⚠
- ⚠
- ⚠
- ⚠
- ⚠
- ⚠

| SYMBOL | MILLIMETER | | |
|--------|------------|------|-------|
| | MIN | NOM | MAX |
| A | — | — | 1.75 |
| A1 | 0.10 | — | 0.225 |
| A2 | 1.30 | 1.40 | 1.50 |
| A3 | 0.60 | 0.65 | 0.70 |
| b | 0.39 | — | 0.47 |
| b1 | 0.38 | 0.41 | 0.44 |
| c | 0.20 | — | 0.24 |
| c1 | 0.19 | 0.20 | 0.21 |
| D | 4.80 | 4.90 | 5.00 |
| E | 5.80 | 6.00 | 6.20 |
| E1 | 3.80 | 3.90 | 4.00 |
| e | 1.27BSC | | |
| h | 0.25 | — | 0.50 |
| L | 0.50 | — | 0.80 |
| L1 | 1.05REF | | |
| θ | 0 | — | 8° |

11.2 WS51F7030M10U (MSOP10)



| SYMBOL | MILLIMETER | | |
|--------|------------|------|------|
| | MIN | NOM | MAX |
| A | — | — | 1.10 |
| A1 | 0.05 | — | 0.15 |
| A2 | 0.75 | 0.85 | 0.95 |
| A3 | 0.30 | 0.35 | 0.40 |
| b | 0.18 | — | 0.26 |
| b1 | 0.17 | 0.20 | 0.23 |
| c | 0.15 | — | 0.19 |
| c1 | 0.14 | 0.15 | 0.16 |
| D | 2.90 | 3.00 | 3.10 |
| E | 4.70 | 4.90 | 5.10 |
| E1 | 2.90 | 3.00 | 3.10 |
| e | 0.50BSC | | |
| L | 0.40 | — | 0.70 |
| L1 | 0.95REF | | |
| θ | 0 | — | 8° |

11.3 WS51F7030S14U (SOP14)

11.4 WS51F7030S16U (SOP16)


12 附录

附录 1 指令集速查表

| 指令 | 描述 | 说明 | 周期 |
|-------------------|----------------------------|---|----|
| 数据传送指令 | | | |
| MOV A,Rn | 寄存器内容送入累加器 | $(A) \leftarrow (Rn)$ | 1 |
| MOV A,direct | 直接地址单元中的数据送入累加器 | $(A) \leftarrow (\text{direct})$ | 1 |
| MOV A,@Ri | 间接 RAM 中的数据送入累加器 | $(A) \leftarrow ((Ri))$ | 1 |
| MOV A,#data8 | 8 位立即数送入累加器 | $(A) \leftarrow \#data$ | 1 |
| MOV Rn,A | 累加器内容送入寄存器 | $(Rn) \leftarrow (A)$ | 1 |
| MOV Rn,direct | 直接地址单元中的数据送入寄存器 | $(Rn) \leftarrow (\text{direct})$ | 2 |
| MOV Rn,#data8 | 8 位立即数送入寄存器 | $(Rn) \leftarrow \#data$ | 1 |
| MOV direct,A | 累加器内容送入直接地址单元 | $(\text{direct}) \leftarrow (A)$ | 1 |
| MOV direct,Rn | 寄存器内容送入直接地址单元 | $(\text{direct}) \leftarrow (Rn)$ | 2 |
| MOV direct,direct | 直接地址单元中的数据送入直接地址单元 | $(\text{direct}) \leftarrow (\text{direct})$ | 2 |
| MOV direct,@Ri | 间接 RAM 中的数据送入直接地址单元 | $(\text{direct}) \leftarrow ((Ri))$ | 2 |
| MOV direct,#data8 | 8 位立即数送入直接地址单元 | $(\text{direct}) \leftarrow \#data$ | 2 |
| MOV @Ri,A | 累加器内容送入间接 RAM 单元 | $((Ri)) \leftarrow (A)$ | 1 |
| MOV @Ri,direct | 直接地址单元中的数据送入间接 RAM 单元 | $((Ri)) \leftarrow (\text{direct})$ | 2 |
| MOV @Ri,#data8 | 8 位立即数送入间接 RAM 单元 | $((Ri)) \leftarrow \#data$ | 1 |
| MOV DPTR,#data16 | 16 位立即数地址送入地址寄存器 | $(DPTR) \leftarrow \#data16$ | 2 |
| MOV A,@A+DPTR | 以 DPTR 为基地址变址寻址单元中的数据送入累加器 | $(A) \leftarrow ((A)) + (DPTR)$ | 2 |
| MOV A,@A+PC | 以 PC 为基地址变址寻址单元中的数据送入累加器 | $(PC) \leftarrow (PC) + 1$ $(A) \leftarrow ((A) + (PC))$ | 2 |
| MOVX A,@Ri | 外部 RAM(8 位地址)送入累加器 | $(A) \leftarrow ((Ri))$ | 2 |
| MOVX A,@DPTR | 外部 RAM(16 位地址)送入累加器 | $(A) \leftarrow ((DPTR))$ | 2 |
| MOVX @Ri,A | 累加器送入外部 RAM(8 位地址) | $((Ri)) \leftarrow (A)$ | 2 |
| MOVX @DPTR,A | 累加器送入外部 RAM(16 位地址) | $(DPTR) \leftarrow (A)$ | 2 |
| PUSH direct | 直接地址单元中的数据压入堆栈 | $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (\text{direct})$ | 2 |
| POP DIRECT | 堆栈中的数据弹出到直接地址单元 | $(\text{direct}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ | 2 |
| XCH A,Rn | 寄存器与累加器交换 | $(A) \leftrightarrow (Rn)$ | 1 |
| XCH A,direct | 直接地址单元与累加器交换 | $(A) \leftrightarrow (\text{direct})$ | 1 |
| XCH A,@Ri | 间接 RAM 与累加器交换 | $(A) \leftrightarrow ((Ri))$ | 1 |
| XCHD A,@Ri | 间接 RAM 与累加器进行低半字节交换 | $(A.3, \dots, A.0) \leftrightarrow ((Ri).3, \dots, (Ri).0)$ | 1 |
| SWAP A | 累加器半字节交换 | $(A.3, \dots, A.0) \leftrightarrow (A.7, \dots, A.4)$ | 1 |
| 算术操作类指令 | | | |

| | | | |
|----------------|-------------------|---|---|
| ADD A, Rn | 寄存器内容加到累加器 | $(A) \leftarrow (A) + (Rn)$ | 1 |
| ADD A, direct | 直接地址单元加到累加器 | $(A) \leftarrow (A) + (\text{direct})$ | 1 |
| ADD A, @Ri | 间接 RAM 内容加到累加器 | $(A) \leftarrow (A) + ((Ri))$ | 1 |
| ADD A, #data8 | 8 位立即数加到累加器 | $(A) \leftarrow (A) + \#data$ | 1 |
| ADDC A, Rn | 寄存器内容带进位加到累加器 | $(A) \leftarrow (A) + (C) + (Rn)$ | 1 |
| ADDC A, direct | 直接地址单元带进位加到累加器 | $(A) \leftarrow (A) + (C) + (\text{direct})$ | 1 |
| ADDC A, @Ri | 间接 RAM 内容带进位加到累加器 | $(A) \leftarrow (A) + (C) + ((Ri))$ | 1 |
| ADDC A, #data8 | 8 位立即数带进位加到累加器 | $(A) \leftarrow (A) + (C) + \#data$ | 1 |
| SUBB A, Rn | 累加器带借位减寄存器内容 | $(A) \leftarrow (A) - (C) - (Rn)$ | 1 |
| SUBB A, direct | 累加器带借位减直接地址单元 | $(A) \leftarrow (A) - (C) - (\text{direct})$ | 1 |
| SUBB A, @Ri | 累加器带借位减间接 RAM 内容 | $(A) \leftarrow (A) - (C) - ((Ri))$ | 1 |
| SUBB A, #data8 | 累加器带借位减 8 位立即数 | $(A) \leftarrow (A) - (C) - \#data$ | 1 |
| INC A | 累加器加 1 | $(A) \leftarrow (A) + 1$ | 1 |
| INC Rn | 寄存器加 1 | $(Rn) \leftarrow (Rn) + 1$ | 1 |
| INC direct | 直接地址单元内容加 1 | $(\text{direct}) \leftarrow (\text{direct}) + 1$ | 1 |
| INC @Ri | 间接 RAM 内容加 1 | $((Ri)) \leftarrow ((Ri)) + 1$ | 1 |
| INC DPTR | DPTR 加 1 | $(DPTR) \leftarrow (DPTR) + 1$ | 2 |
| DEC A | 累加器减 1 | $(A) \leftarrow (A) - 1$ | 1 |
| DEC Rn | 寄存器减 1 | $(Rn) \leftarrow (Rn) - 1$ | 1 |
| DEC direct | 直接地址单元内容减 1 | $(\text{direct}) \leftarrow (\text{direct}) - 1$ | 1 |
| DEC @Ri | 间接 RAM 内容减 1 | $((Ri)) \leftarrow ((Ri)) - 1$ | 1 |
| MUL AB | A 乘以 B | temp16 \leftarrow (A) X (B) (A) \leftarrow (temp.7,temp.6,...,temp.0) (B) \leftarrow (temp.15,temp.14,...,temp.8) | 4 |
| DIV AB | A 除以 B | QUO \leftarrow (A) / (B)REM (A) \leftarrow QUO (B) \leftarrow REM | 4 |
| DA A | 累加器进行十进制转换 | IF (A.3,...,A.0) > 9 AC = 1 THEN temp16 \leftarrow (A) + 0x06 | 1 |

| | | | |
|--------------------|---------------------|--|---|
| | | <pre> (A) ← (temp.7,...,temp.0) IF (temp16) > 0xFF THEN CY ← 1 IF (A.7,...,A.4) > 9 CY = 1 THEN temp16 ← (A) + 0x60 (A) ← (temp.7,...,temp.0) IF (temp16) > 0xFF THEN CY ← 1 </pre> | |
| 逻辑操作类指令 | | | |
| ANL A, Rn | 累加器与寄存器相“与” | $(A) \leftarrow (A) \& (Rn)$ | 1 |
| ANL A, direct | 累加器与直接地址单元相“与” | $(A) \leftarrow (A) \& (\text{direct})$ | 1 |
| ANL A, @Ri | 累加器与间接 RAM 内容相“与” | $(A) \leftarrow (A) \& ((Ri))$ | 1 |
| ANL A, #data8 | 累加器与 8 位立即数相“与” | $(A) \leftarrow (A) \& \#data$ | 1 |
| ANL direct, A | 直接地址单元与累加器相“与” | $(\text{direct}) \leftarrow (\text{direct}) \& (A)$ | 1 |
| ANL direct, #data8 | 直接地址单元与 8 位立即数相“与” | $(\text{direct}) \leftarrow (\text{direct}) \& \#data$ | 2 |
| ORL A, Rn | 累加器与寄存器相“或” | $(A) \leftarrow (A) (Rn)$ | 1 |
| ORL A, direct | 累加器与直接地址单元相“或” | $(A) \leftarrow (A) (\text{direct})$ | 1 |
| ORL A, @Ri | 累加器与间接 RAM 内容相“或” | $(A) \leftarrow (A) ((Ri))$ | 1 |
| ORL A, #data8 | 累加器与 8 位立即数相“或” | $(A) \leftarrow (A) \#data$ | 1 |
| ORL direct, A | 直接地址单元与累加器相“或” | $(\text{direct}) \leftarrow (\text{direct}) (A)$ | 1 |
| ORL direct, #data8 | 直接地址单元与 8 位立即数相“或” | $(\text{direct}) \leftarrow (\text{direct}) \#data$ | 2 |
| XRL A, Rn | 累加器与寄存器相“异或” | $(A) \leftarrow (A) \wedge (Rn)$ | 1 |
| XRL A, direct | 累加器与直接地址单元相“异或” | $(A) \leftarrow (A) \wedge (\text{direct})$ | 1 |
| XRL A, @Ri | 累加器与间接 RAM 内容相“异或” | $(A) \leftarrow (A) \wedge ((Ri))$ | 1 |
| XRL A, #data8 | 累加器与 8 位立即数相“异或” | $(A) \leftarrow (A) \wedge \#data$ | 1 |
| XRL direct, A | 直接地址单元与累加器相“异或” | $(\text{direct}) \leftarrow (\text{direct}) \wedge (A)$ | 1 |
| XRL direct, #data8 | 直接地址单元与 8 位立即数相“异或” | $(\text{direct}) \leftarrow (\text{direct}) \wedge \#data$ | 2 |
| CLR A | 累加器清 0 | $(A) \leftarrow 0$ | 1 |

| | | | |
|--------------|----------------|--|---|
| CPL A | 累加器求反 | $(A) \leftarrow \neg(A)$ | 1 |
| RL A | 累加器循环左移 | $(A) \leftarrow (A.6, A.5, \dots, A.0, A.7)$ | 1 |
| RLC A | 累加器带进位循环左移 | $C \leftarrow A.7$ $(A) \leftarrow (A.6, A.5, \dots, A.0, C)$ | 1 |
| RR A | 累加器循环右移 | $(A) \leftarrow (A.0, A.7, \dots, A.2, A.1)$ | 1 |
| RRC A | 累加器带进位循环右移 | $C \leftarrow A.0$ $(A) \leftarrow (C, A.7, \dots, A.2, A.1)$ | 1 |
| 控制转移类指令 | | | |
| ACALL addr11 | 绝对短调用子程序 | $(PC) \leftarrow (PC) + 2$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC7-0)$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC15-8)$ $(PC10-0) \leftarrow \text{page address}$ | 2 |
| LACLL addr16 | 长调用子程序 | $(PC) \leftarrow (PC) + 3$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC7-0)$ $((SP)) \leftarrow (PC15-8)$ $(PC) \leftarrow \text{addr15-0}$ | 2 |
| RET | 子程序返回 | $(PC15-8) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ $(PC7-0) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ | 2 |
| RETI | 中断返回 | $(PC15-8) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ $(PC7-0) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ | 2 |
| AJMP addr11 | 绝对短转移 | $(PC) \leftarrow (PC) + 2$ $(PC10-0) \leftarrow \text{page address}$ | 2 |
| LJMP addr16 | 长转移 | $(PC) \leftarrow (PC) + 3$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC7-0)$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC15-8)$ $(PC10-0) \leftarrow \text{addr15-0}$ | 2 |
| SJMP rel | 相对转移 | $(PC) \leftarrow (PC) + 2$ $(PC) \leftarrow (PC) + \text{rel}$ | 2 |
| JMP @A+DPTR | 相对于 DPTR 的间接转移 | $(PC) \leftarrow (A) + (DPTR)$ | 2 |

| | | | |
|-----------------------|----------------------|--|---|
| JZ rel | 累加器为零转移 | $(PC) \leftarrow (PC) + 2$ IF (A) = 0 THEN $(PC) \leftarrow (PC) + rel$ | 2 |
| JNZ rel | 累加器非零转移 | $(PC) \leftarrow (PC) + 2$ IF (A) \neq 0 THEN $(PC) \leftarrow (PC) + rel$ | 2 |
| CJNE A, direct, rel | 累加器与直接地址单元比较, 不等则转移 | $(PC) \leftarrow (PC) + 3$ IF (A) \neq (direct) THEN $(PC) \leftarrow (PC) +$ relative offset IF (A) < (direct) THEN $(C) \leftarrow 1$ ELSE $(C) \leftarrow 0$ | 2 |
| CJNE A, #data8, rel | 累加器与 8 位立即数比较, 不等则转移 | $(PC) \leftarrow (PC) + 3$ IF (A) \neq data THEN $(PC) \leftarrow (PC) +$ relative offset IF (A) < data THEN $(C) \leftarrow 1$ ELSE $(C) \leftarrow 0$ | 2 |
| CJNE Rn, #data8, rel | 寄存器与 8 位立即数比较, 不等则转移 | $(PC) \leftarrow (PC) + 3$ IF (Rn) \neq data THEN $(PC) \leftarrow (PC) +$ relative offset IF (Rn) < data THEN $(C) \leftarrow 1$ ELSE $(C) \leftarrow 0$ | 2 |
| CJNE @Ri, #data8, rel | 间接 RAM 单元, 不等则转移 | $(PC) \leftarrow (PC) + 3$ IF ((Ri)) \neq data THEN $(PC) \leftarrow (PC) +$ relative offset IF ((Ri)) < data | 2 |

| | | | |
|------------------|-----------------------------|---|---|
| | | THEN (C) ← 1 ELSE (C) ← 0 | |
| DJNZ Rn, rel | 寄存器减 1, 非零转移 | (PC) ← (PC) + 2 (Rn) ← (Rn) - 1 IF (Rn) <> 0 THEN (PC) ← (PC) + rel | 2 |
| DJNZ direct, rel | 直接地址单元减 1, 非零转移 | (PC) ← (PC) + 2 (direct) ← (direct) - 1 IF (direct) <> 0 THEN (PC) ← (PC) + rel | 2 |
| NOP | 空操作 | (PC) ← (PC) + 1 | 1 |
| 布尔变量操作类指令 | | | |
| CLR C | 清进位位 | (C) ← 0 | 1 |
| CLR bit | 清直接地址位 | (bit) ← 0 | 1 |
| SETB C | 置进位位 | (C) ← 1 | 1 |
| SETB bit | 置直接地址位 | (bit) ← 1 | 1 |
| CPL C | 进位位求反 | (C) ← /(C) | 1 |
| CPL bit | 直接地址位求反 | (bit) ← /(bit) | 1 |
| ANL C, bit | 进位位和直接地址位相“与” | (C) ← (C) & (bit) | 2 |
| ANL C, /bit | 进位位和直接地址位的反码相“与” | (C) ← (C) & /(bit) | 2 |
| ORL C, bit | 进位位和直接地址位相“或” | (C) ← (C) (bit) | 2 |
| ORL C, /bit | 进位位和直接地址位的反码相“或” | (C) ← (C) /(bit) | 2 |
| MOV C, bit | 直接地址位送入进位位 | (C) ← (bit) | 1 |
| MOV bit, C | 进位位送入直接地址位 | (bit) ← (C) | 2 |
| JC rel | 进位位为 1 则转移(CY=0 不转移, =1 转移) | (PC) ← (PC) + 2 IF (C) = 1 THEN (PC) ← (PC) + rel | 2 |
| JNC rel | 进位位为 0 则转移 | (PC) ← (PC) + 2 IF (C) = 0 THEN (PC) ← (PC) + rel | 2 |
| JB bit, rel | 直接地址位为 1 则转移 | (PC) ← (PC) + 3 IF (bit) = 1 THEN (PC) ← (PC) + rel | 2 |
| JNB bit, rel | 直接地址位为 0 则转移 | (PC) ← (PC) + 3 IF (bit) = 0 THEN (PC) ← (PC) + rel | 2 |
| JBC bit, rel | 直接地址位为 1 则转移, 该位清零 | (PC) ← (PC) + 3 IF (bit) = 1 THEN (bit) ← 0 | 2 |

| | | | |
|---------|-------------------|-------------------|--|
| | | (PC) ← (PC) + rel | |
| 伪指令 | | | |
| ORG | 设置程序起始地址 | | |
| END | 标志源代码结束 | | |
| EQU | 定义常数 | | |
| SET | 定义整型数 | | |
| DATA | 给数据地址定值 | | |
| BYTE | 给字节类型符号定值 | | |
| WORD | 给字类型符号定值 | | |
| BIT | 给位地址取名 | | |
| ALTNAME | 用自定义名取代保留字 | | |
| DB | 给一块连续的存储区装载字节型数据 | | |
| DW | 给一块连续的存储区装载字型数据 | | |
| DS | 预留一个连续的存储区或装入指定字节 | | |
| INCLUDE | 将一个源文件插入程序中 | | |
| TITLE | 列表文件中加入标题行 | | |
| NOLIST | 汇编时不产生列表文件 | | |
| NOCODE | 条件汇编时，条件为假的不产生清单 | | |

前海维晟 WISESUN

13 版本历史

| 版本号 | 内容 | 时间 |
|------|-----------------|------------|
| V1.0 | 初版发布 | 2020.04.17 |
| V1.1 | 添加 check sum 说明 | 2020.07.29 |
| | | |
| | | |
| | | |
| | | |
| | | |

前海维晟 WISESUN

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [32-bit Microcontrollers - MCU category](#):

Click to view products by [Wisesun manufacturer](#):

Other Similar products are found below :

[MCF51AC256AVFUE](#) [MCF51AC256BCFUE](#) [MCF51AC256BVFUE](#) [MB91F464AAPMC-GSE2](#) [R5S726B0D216FP#V0](#) [MB91F248PFV-GE1](#) [MB91243PFV-GS-136E1](#) [SAK-TC1782F-320F180HR BA](#) [TC364DP64F300WAAKXUMA1](#) [R5F566NNDDFP#30](#)
[R5F566NNDDFC#30](#) [R5F566NNDDBD#20](#) [MC96F8216ADBN](#) [A96G181HDN](#) [A96G140KNN](#) [A96G174FDN](#) [A31G213CL2N](#)
[A96G148KNN](#) [A96G174AEN](#) [AC33M3064TLBN-01](#) [V3s](#) [T3](#) [A40i-H](#) [V526](#) [A83T](#) [R11](#) [V851s](#) [A133](#) [V833](#) [F1C100S](#) [T3L](#) [T507](#) [A33](#)
[A63](#) [T113-i](#) [H616](#) [V853](#) [V533](#) [R16-J](#) [V536-H](#) [A64-H](#) [V831](#) [V3LP](#) [T113-S3](#) [F1C200S](#) [F133-A](#) [R128-S2](#) [D1-H](#) [ADUCM360BCPZ128-TR](#)
[APT32S003F8PT](#)