

C8051F020/1/2/3
混合信号 ISP FLASH 微控制器
数 据 手 册

潘 琢 金 译

Rev 1.1 2002.10

版权所有

版 权 声 明

本手册中文版版权归译者和沈阳新华龙电子有限公司所有。研究和开发人员可以自由使用本手册。任何单位和个人未经版权所有者授权不得在任何形式的出版物中摘抄本手册内容。

译者将在本手册英文版更新后及时更新中文版内容。译文中一定存在不少错误和不准确之处，望各位同仁不吝赐教，以便在新版本中更正。

译者联系方式：

沈阳航空工业学院计算机系 潘琢金

电话：024-86802267 ； 13066535936

Email: panzhuojin@sina.com 或 panzhj@syiae.edu.cn

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

模拟外设

- SAR ADC
 - 12 位 (C8051F020/1)
 - 10 位 (C8051F022/3)
 - ± 1 LSB INL
 - 可编程转换速率, 最大 100ksps
 - 可多达 8 个外部输入: 可编程为单端输入或差分输入
 - 可编程放大器增益: 16、8、4、2、1、0.5
 - 数据相关窗口中断发生器
 - 内置温度传感器 ($\pm 3^{\circ}\text{C}$)
- 8 位 ADC
 - 可编程转换速率, 最大 500ksps
 - 8 个外部输入
 - 可编程放大器增益: 4、2、1、0.5
- 两个 12 位 DAC
 - 可以同步输出, 用于产生无抖动波形
- 两个模拟比较器
- 电压基准
- 精确的 VDD 监视器和欠压检测器

片内 JTAG 调试和边界扫描

- 片内调试电路提供全速、非侵入式的在系统调试 (不需仿真器)
- 支持断点、单步、观察点、堆栈监视器: 可以观察/修改存储器和寄存器
- 比使用仿真芯片、目标仿真头和仿真插座的仿真系统有更好的性能
- 符合 IEEE1149.1 边界扫描标准
- 廉价而完全的开发套件

高速 8051 微控制器内核

- 流水线指令结构: 70% 的指令的执行时间为一个或两个系统时钟周期
- 速度可达 25MIPS (时钟频率为 25MHz 时)
- 22 个矢量中断源

存储器

- 4352 字节内部数据 RAM (4K + 256)
- 64K 字节闪存存储器: 可以在系统编程, 扇区大小为 512 字节
- 外部 64K 字节数据存储器接口 (可编程为复用方式或非复用方式)

数字外设

- 8 个字节宽的端口 I/O (C8051F020/2) 或 4 个字节宽的端口 I/O (C8051F021/3); 所有口线均耐 5V 电压
- 可同时使用的硬件 SMBus(I²C™ 兼容)、SPI™ 及两个 UART 串口
- 可编程的 16 位计数器/定时器阵列, 有 5 个捕捉/比较模块
- 5 个通用 16 位计数器/定时器
- 专用的看门狗定时器; 双向复位引脚

时钟源

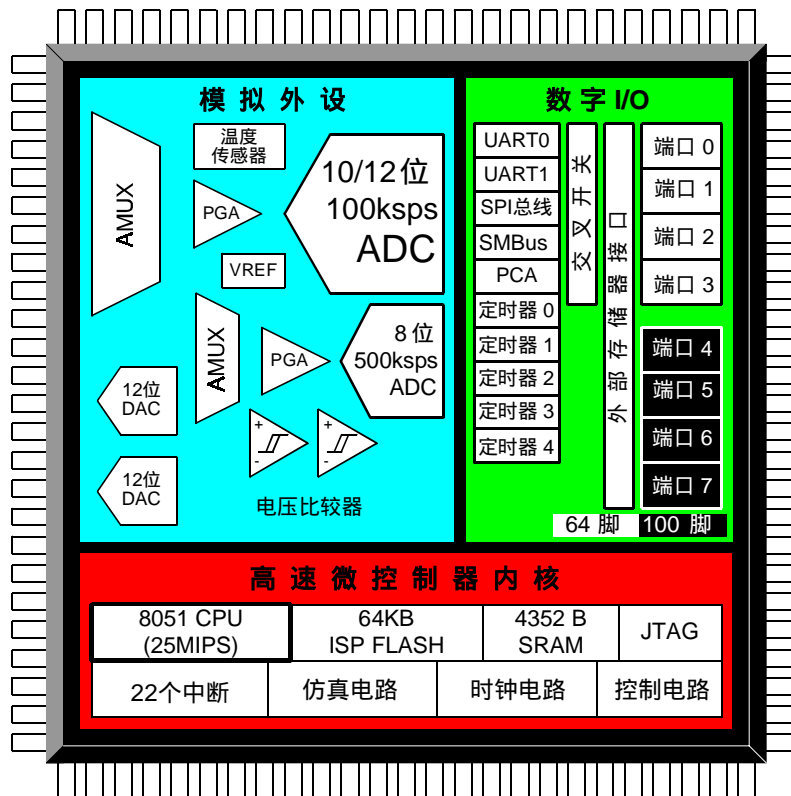
- 内部可编程振荡器: 2-16MHz
- 外部振荡器: 晶体、RC、C、或外部时钟
- 实时时钟方式 (用定时器 3 或 PCA);

供电电压.....2.7V - 3.6V

- 典型工作电流: 10mA @ 20MHz
- 多种节电休眠和停机方式

100 脚 TQFP 和 64 脚 TQFP 封装

温度范围: $-40^{\circ}\text{C} - +85^{\circ}\text{C}$



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

| | |
|--|-----------|
| 1. 系统概述 | 8 |
| 1.1 CIP-51™ CPU | 13 |
| 1.1.1 与 8051 完全兼容 | 13 |
| 1.1.2 速度提高 | 13 |
| 1.1.3 增加的功能 | 14 |
| 1.2 片内存储器 | 15 |
| 1.3 JTAG 调试和边界扫描 | 16 |
| 1.4 可编程数字 I/O 和交叉开关 | 17 |
| 1.5 可编程计数器阵列 | 18 |
| 1.6 串行端口 | 18 |
| 1.7 12 位模/数转换器 | 19 |
| 1.8 8 位模/数转换器 | 20 |
| 1.9 比较器和 DAC | 21 |
| 2. 极限参数 | 22 |
| 3. 总体直流电气特性 | 23 |
| 4. 引脚和封装定义 | 24 |
| 5. ADC0 (12 位, 只限于 C8051F020/1) | 33 |
| 5.1 模拟多路开关和 PGA | 33 |
| 5.2 ADC 的工作方式 | 34 |
| 5.2.1 启动转换 | 34 |
| 5.2.2 跟踪方式 | 35 |
| 5.2.3 建立时间要求 | 36 |
| 5.3 ADC0 可编程窗口检测器 | 43 |
| 6. ADC0 (10 位, 只限于 C8051F022/3) | 49 |
| 6.1 模拟多路开关和 PGA | 49 |
| 6.2 ADC 的工作方式 | 50 |
| 6.2.1 启动转换 | 50 |
| 6.2.2 跟踪方式 | 51 |
| 6.2.3 建立时间要求 | 52 |
| 6.3 ADC0 可编程窗口检测器 | 59 |
| 7. ADC1 (8 位 ADC) | 65 |
| 7.1 模拟多路开关和 PGA | 65 |
| 7.2 ADC1 的工作方式 | 66 |
| 7.2.1 启动转换 | 66 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

| | | |
|--------|--------------------|-----|
| 7.2.2 | 跟踪方式 | 66 |
| 7.2.3 | 建立时间要求 | 68 |
| 8. | 12 位电压输出 DAC | 73 |
| 8.1 | DAC 输出更新 | 73 |
| 8.1.1 | 根据软件命令更新输出 | 73 |
| 8.1.2 | 基于定时器溢出的输出更新 | 74 |
| 8.2 | DAC 输出定标/调整 | 74 |
| 9. | 电压基准 (C8051F020/2) | 79 |
| 10. | 电压基准 (C8051F021/3) | 81 |
| 11. | 比较器 | 83 |
| 12. | CIP-51 微控制器 | 88 |
| 12.1 | 指令集 | 89 |
| 12.1.1 | 指令和 CPU 时序 | 89 |
| 12.1.2 | MOVX 指令和程序存储器 | 89 |
| 12.2 | 存储器组织 | 93 |
| 12.2.1 | 程序存储器 | 93 |
| 12.2.2 | 数据存储器 | 94 |
| 12.2.3 | 通用寄存器 | 94 |
| 12.2.4 | 位寻址空间 | 94 |
| 12.2.5 | 堆栈 | 94 |
| 12.2.6 | 特殊功能寄存器 | 95 |
| 12.2.7 | 寄存器说明 | 99 |
| 12.3 | 中断系统 | 102 |
| 12.3.1 | MCU 中断源和中断向量 | 102 |
| 12.3.2 | 外部中断 | 102 |
| 12.3.3 | 中断优先级 | 104 |
| 12.3.4 | 中断响应时间 | 104 |
| 12.3.5 | 中断寄存器说明 | 104 |
| 12.4 | 电源管理方式 | 111 |
| 12.4.1 | 等待方式 | 111 |
| 12.4.2 | 停机方式 | 111 |
| 13. | 复位源 | 113 |
| 13.1 | 上电复位 | 114 |
| 13.2 | 掉电复位 | 114 |
| 13.3 | 外部复位 | 114 |
| 13.4 | 软件强制复位 | 115 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

| | |
|-----------------------------------|------------|
| 13.5 时钟丢失检测器复位..... | 115 |
| 13.6 比较器 0 复位..... | 115 |
| 13.7 外部 CNVSTR 引脚复位..... | 115 |
| 13.8 看门狗定时器复位..... | 115 |
| 13.8.1 允许/复位 WDT..... | 116 |
| 13.8.2 禁止 WDT..... | 116 |
| 13.8.3 禁止 WDT 锁定..... | 116 |
| 13.8.4 设置 WDT 定时间隔..... | 116 |
| 14. 振荡器..... | 119 |
| 14.1 外部晶体举例..... | 122 |
| 14.2 外部 RC 举例..... | 122 |
| 14.3 外部电容举例..... | 122 |
| 15. FLASH 存储器..... | 123 |
| 15.1 FLASH 存储器编程..... | 123 |
| 15.2 非易失性数据存储..... | 124 |
| 15.3 安全选项..... | 124 |
| 16. 外部数据存储器接口和片内 XRAM..... | 128 |
| 16.1 访问 XRAM..... | 128 |
| 16.1.1 16 位 MOVX 示例..... | 128 |
| 16.1.2 8 位 MOVX 示例..... | 128 |
| 16.2 配置外部存储器接口..... | 129 |
| 16.3 端口选择和配置..... | 129 |
| 16.4 复用和非复用选择..... | 131 |
| 16.4.1 复用方式配置..... | 131 |
| 16.4.2 非复用方式配置..... | 132 |
| 16.5 存储器模式选择..... | 133 |
| 16.5.1 只用内部 XRAM..... | 133 |
| 16.5.2 无块选择的分片模式..... | 133 |
| 16.5.3 带块选择的分片模式..... | 134 |
| 16.5.4 只用外部存储器..... | 134 |
| 16.6 时序..... | 135 |
| 16.6.1 非复用方式..... | 136 |
| 16.6.2 复用方式..... | 139 |
| 17. 端口输入/输出..... | 143 |
| 17.1 端口 0-3 和优先权交叉开关译码器..... | 144 |
| 17.1.1 交叉开关引脚分配..... | 145 |
| 17.1.2 配置端口引脚的输出方式..... | 146 |
| 17.1.3 配置端口引脚为数字输入..... | 146 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

| | |
|--|------------|
| 17.1.4 外部中断 (IE6 和 IE7) | 146 |
| 17.1.5 弱上拉 | 147 |
| 17.1.6 配置端口 1 的引脚为模拟输入 (AIN.[7:0]) | 147 |
| 17.1.7 外部存储器接口引脚分配 | 147 |
| 17.1.8 交叉开关引脚分配示例 | 150 |
| 17.2 端口 4-7 (只对 C8051F020/2) | 159 |
| 17.2.1 配置无引出脚的端口 | 159 |
| 17.2.2 配置端口引脚的输出方式 | 159 |
| 17.2.3 配置端口引脚为数字输入 | 160 |
| 17.2.4 弱上拉 | 160 |
| 17.2.5 外部存储器接口 | 160 |
| 18. SMBUS | 164 |
| 18.1 支持文档 | 165 |
| 18.2 SMBUS 协议 | 166 |
| 18.2.1 总线仲裁 | 166 |
| 18.2.2 时钟低电平扩展 | 167 |
| 18.2.3 SCL 低电平超时 | 167 |
| 18.2.4 SCL 高电平 (SMBus 空闲) 超时 | 167 |
| 18.3 SMBUS 数据传输方式 | 167 |
| 18.3.1 主发送器方式 | 167 |
| 18.3.2 主接收器方式 | 168 |
| 18.3.3 从发送器方式 | 168 |
| 18.3.4 从接收器方式 | 169 |
| 18.4 SMBUS 特殊功能寄存器 | 169 |
| 18.4.1 控制寄存器 | 169 |
| 18.4.2 时钟速率寄存器 | 172 |
| 18.4.3 数据寄存器 | 172 |
| 18.4.4 地址寄存器 | 173 |
| 18.4.5 状态寄存器 | 173 |
| 19. 串行外设接口总线 | 176 |
| 19.1 信号说明 | 177 |
| 19.1.1 主输出、从输入 (MOSI) | 177 |
| 19.1.2 主输入、从输出 (MISO) | 177 |
| 19.1.3 串行时钟 (SCK) | 177 |
| 19.1.4 从选择 (NSS) | 177 |
| 19.2 SPI 操作 | 178 |
| 19.3 串行时钟时序 | 179 |
| 19.4 SPI 特殊功能寄存器 | 180 |
| 20. UART0 | 183 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

| | | |
|------------|-----------------------------|------------|
| 20.1 | UART0 工作方式 | 184 |
| 20.1.1 | 方式0: 同步方式 | 184 |
| 20.1.2 | 方式1: 8 位 UART, 可变波特率 | 185 |
| 20.1.3 | 方式2: 9 位 UART, 固定波特率 | 186 |
| 20.1.4 | 方式3: 9 位 UART, 可变波特率 | 187 |
| 20.2 | 多机通信 | 187 |
| 20.3 | 帧错误和传输错误检测 | 188 |
| 21. | UART1 | 192 |
| 21.1 | UART1 工作方式 | 193 |
| 21.1.1 | 方式0: 同步方式 | 193 |
| 21.1.2 | 方式1: 8 位 UART, 可变波特率 | 194 |
| 21.1.3 | 方式2: 9 位 UART, 固定波特率 | 195 |
| 21.1.4 | 方式3: 9 位 UART, 可变波特率 | 196 |
| 21.2 | 多机通信 | 196 |
| 21.3 | 帧错误和传输错误检测 | 197 |
| 22. | 定时器 | 201 |
| 22.1 | 定时器 0 和定时器 1 | 202 |
| 22.1.1 | 方式0: 13 位计数器/定时器 | 202 |
| 22.1.2 | 方式1: 16 位计数器/定时器 | 204 |
| 22.1.3 | 方式2: 8 位自动重载的计数器/定时器 | 204 |
| 22.1.4 | 方式3: 两个 8 位计数器/定时器 (仅定时器 0) | 205 |
| 22.2 | 定时器 2 | 209 |
| 22.2.1 | 方式0: 带捕捉的 16 位计数器/定时器 | 210 |
| 22.2.2 | 方式1: 自动重载的 16 位计数器/定时器 | 211 |
| 22.2.3 | 方式2: 波特率发生器 | 212 |
| 22.3 | 定时器 3 | 215 |
| 22.4 | 定时器 4 | 218 |
| 22.4.1 | 方式0: 带捕捉的 16 位计数器/定时器 | 219 |
| 22.4.2 | 方式1: 自动重载的 16 位计数器/定时器 | 220 |
| 22.4.3 | 方式2: 波特率发生器 | 221 |
| 23. | 可编程计数器阵列 | 224 |
| 23.1 | PCA 计数器/定时器 | 225 |
| 23.2 | 捕捉/比较模块 | 226 |
| 23.2.1 | 边沿触发的捕捉方式 | 227 |
| 23.2.2 | 软件定时器 (比较) 方式 | 228 |
| 23.2.3 | 高速输出方式 | 229 |
| 23.2.4 | 频率输出方式 | 230 |
| 23.2.5 | 8 位脉宽调制器方式 | 231 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

| | |
|------------------------------------|------------|
| 23.2.6 16 位脉宽调制器方式..... | 232 |
| 23.3 PCA0 的寄存器说明..... | 233 |
| 24. JTAG (IEEE 1149.1)..... | 237 |
| 24.1 边界扫描..... | 238 |
| 24.1.1 EXTEST 指令..... | 239 |
| 24.1.2 SAMPLE 指令..... | 239 |
| 24.1.3 BYPASS 指令..... | 239 |
| 24.1.4 IDCODE 指令..... | 239 |
| 24.2 闪存编程命令..... | 240 |
| 24.3 调试支持..... | 243 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

1. 系统概述

C8051F020/1/2/3 器件是完全集成的混合信号系统级 MCU 芯片，具有 64 个数字 I/O 引脚（C8051F020/2）或 32 个数字 I/O 引脚（C8051F021/3）。下面列出了一些主要特性；有关某一产品的具体特性参见表 1.1。

- 高速、流水线结构的 8051 兼容的 CIP-51 内核（可达 25MIPS）
- 全速、非侵入式的在系统调试接口（片内）
- 真正 12 位（C8051F020/1）或 10 位（C8051F022/3）、100 ksp/s 的 8 通道 ADC，带 PGA 和模拟多路开关。
- 两个 12 位 DAC，可编程更新时序
- 64K 字节可在系统编程的 FLASH 存储器
- 4352（4096+256）字节的片内 RAM
- 可寻址 64K 字节地址空间的外部数据存储接口
- 硬件实现的 SPI、SMBus/I²C 和两个 UART 串行接口
- 5 个通用的 16 位定时器
- 具有 5 个捕捉/比较模块的可编程计数器/定时器阵列
- 片内看门狗定时器、VDD 监视器和温度传感器

具有片内 VDD 监视器、看门狗定时器和时钟振荡器的 C8051F020/1/2/3 是真正能独立工作的片上系统。所有模拟和数字外设均可由用户固件配置为使能或禁止。FLASH 存储器还具有在系统重新编程能力，可用于非易失性数据存储，并允许现场更新 8051 固件。

片内 JTAG 调试电路允许使用安装在最终应用系统上的产品 MCU 进行非侵入式（不占用片内资源）、全速、在系统调试。该调试系统支持观察和修改存储器和寄存器，支持断点、观察点、单步及运行和停机命令。在使用 JTAG 调试时，所有的模拟和数字外设都可全功能运行。

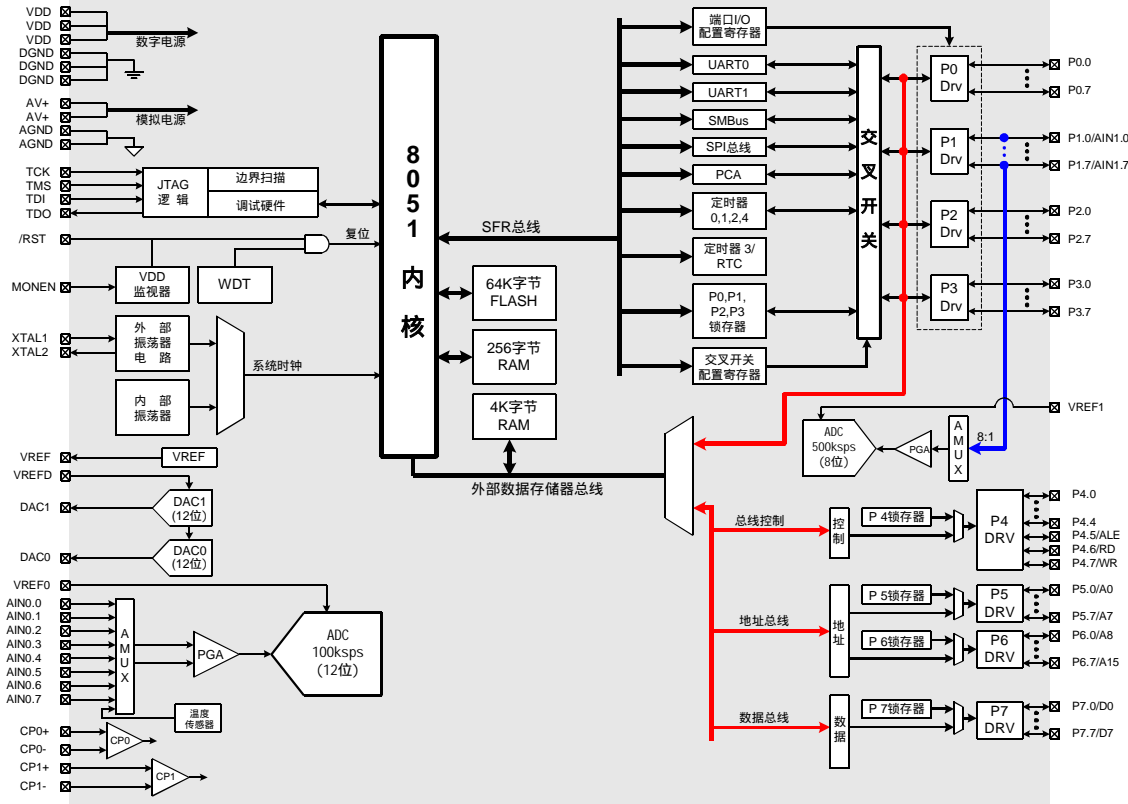
每个 MCU 都可在工业温度范围（-45℃到+85℃）内用 2.7V-3.6V 的电压工作。端口 I/O、/RST 和 JTAG 引脚都容许 5V 的输入信号电压。C8051F020/2 为 100 脚 TQFP 封装（见图 1.1 和图 1.3 的框图）。C8051F021/3 为 64 脚 TQFP 封装（见图 1.2 和图 1.4 的框图）。

表 1.1 产品选择指南

| | MIPS(峰值) | FLASH 存储器 | RAM | 外部存储器接口 | SMBus/I ² C | SPI | UART | 定时器(16位) | 可编程计数器阵列 | 数字端口 I/O | 12 位 100ksp/s ADC 输入 | 10 位 100ksp/s ADC 输入 | 8 位 500ksp/s ADC 输入 | 电压基准 | 温度传感器 | DAC 分辨率 | DAC 输出 | 电压比较器 | 封装 |
|-----------|----------|-----------|------|---------|------------------------|-----|------|----------|----------|----------|----------------------|----------------------|---------------------|------|-------|---------|--------|-------|---------|
| C8051F020 | 25 | 64k | 4352 | √ | √ | √ | 2 | 5 | √ | 64 | 8 | - | 8 | √ | √ | 12 | 2 | 2 | 100TQFP |
| C8051F021 | 25 | 64k | 4352 | √ | √ | √ | 2 | 5 | √ | 32 | 8 | - | 8 | √ | √ | 12 | 2 | 2 | 64TQFP |
| C8051F022 | 25 | 64k | 4352 | √ | √ | √ | 2 | 5 | √ | 64 | - | 8 | 8 | √ | √ | 12 | 2 | 2 | 100TQFP |
| C8051F023 | 25 | 64k | 4352 | √ | √ | √ | 2 | 5 | √ | 32 | - | 8 | 8 | √ | √ | 12 | 2 | 2 | 64TQFP |

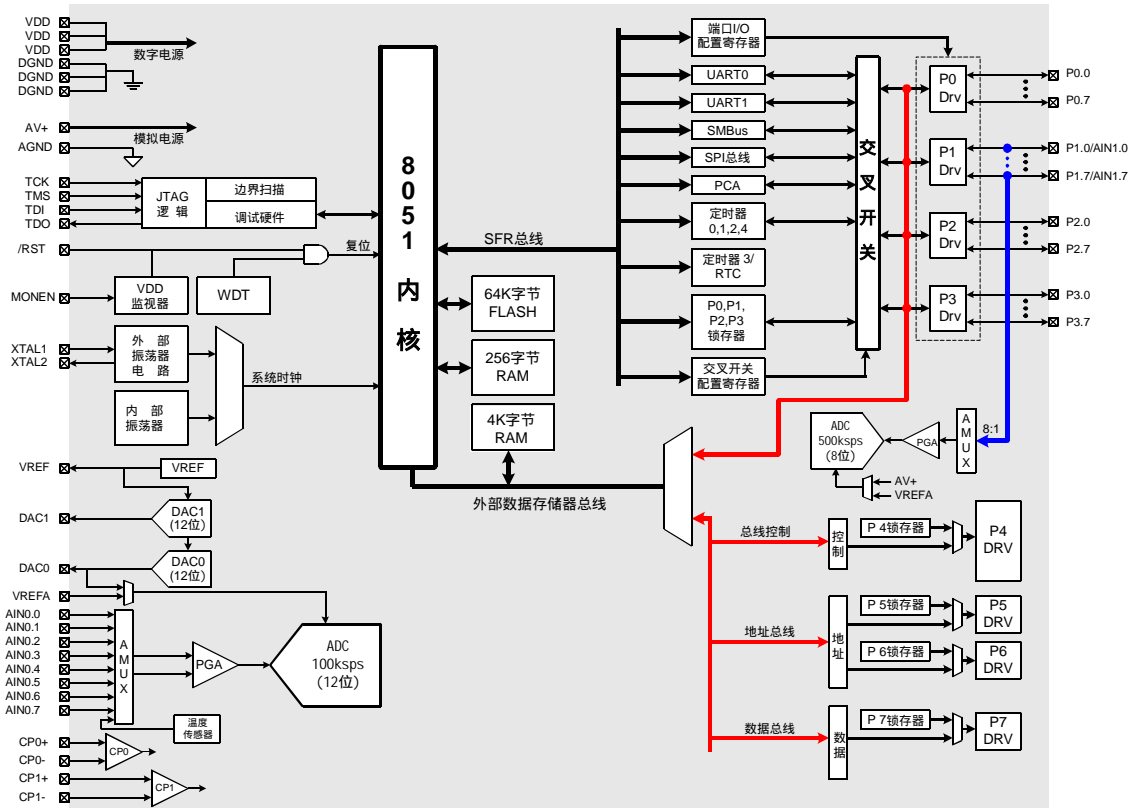
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 1.1 C8051F020 原理框图



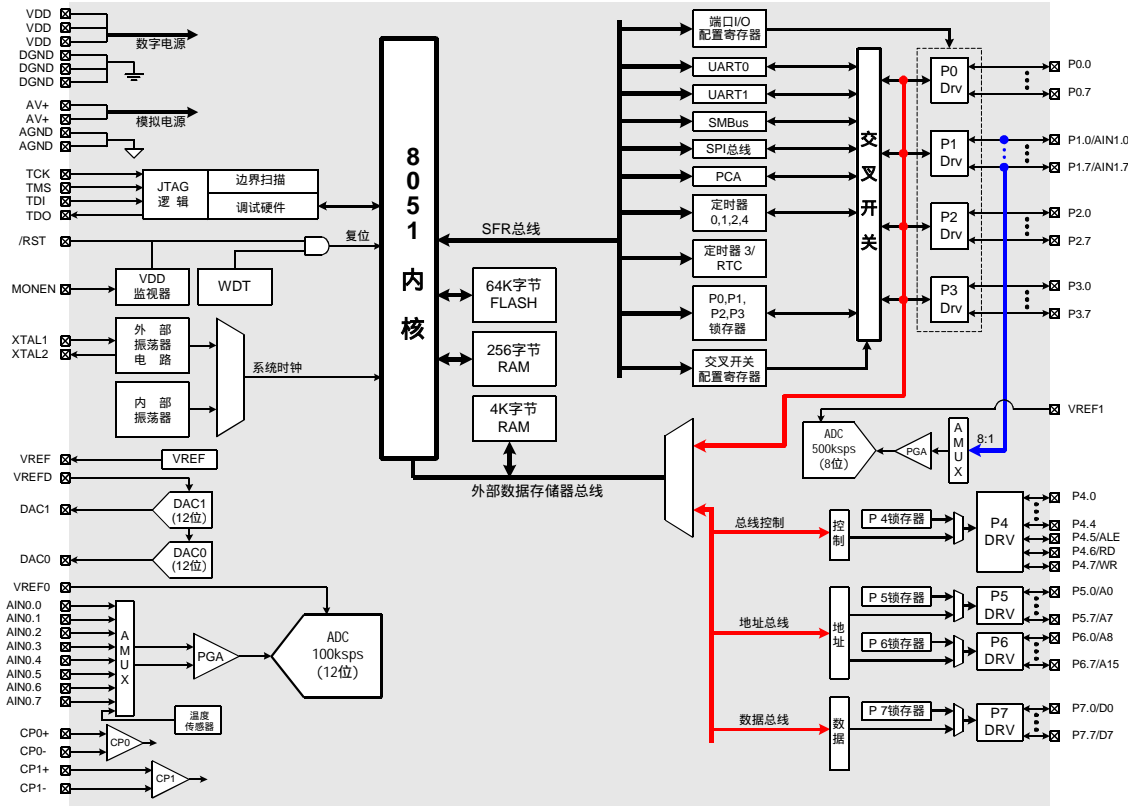
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 1.2 C8051F021 原理框图



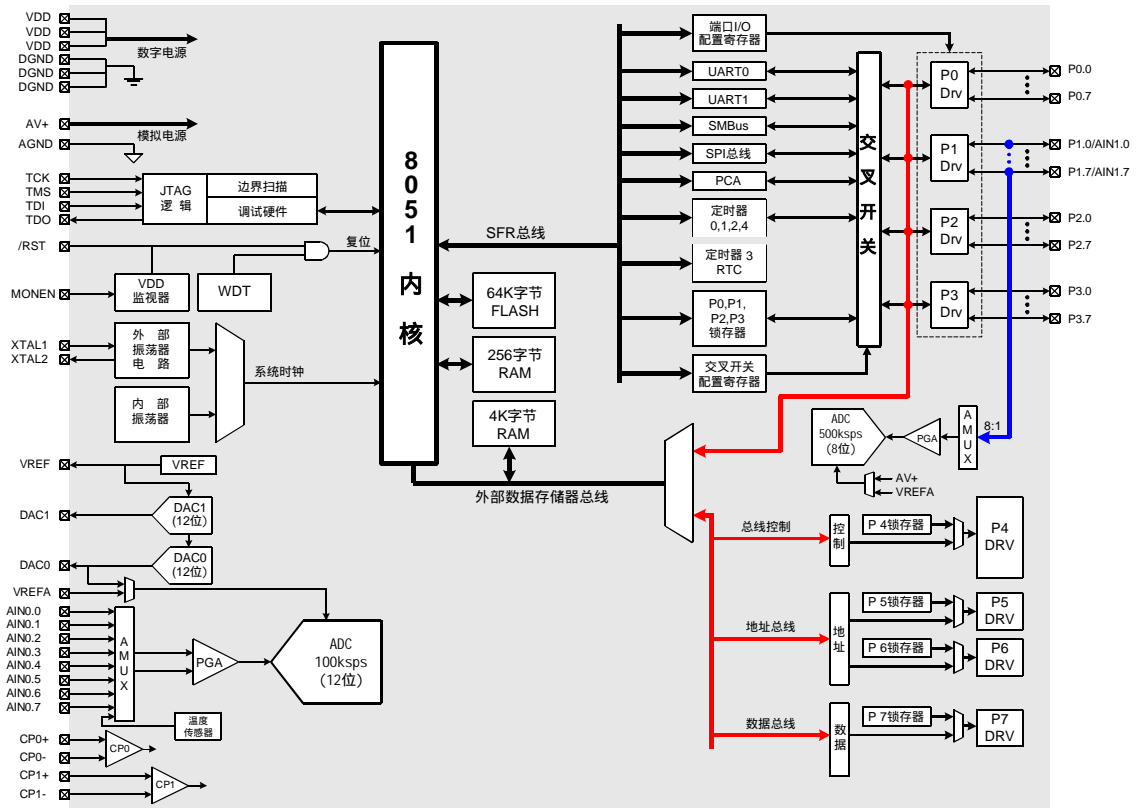
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 1.3 C8051F022 原理框图



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 1.4 C8051F023 原理框图



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

1.1 CIP-51™ CPU

1.1.1 与 8051 完全兼容

C8051F020 系列器件使用 Cygnal 的专利 CIP-51 微控制器内核。CIP-51 与 MCS-51™ 指令集完全兼容，可以使用标准 803x/805x 的汇编器和编译器进行软件开发。CIP-51 内核具有标准 8052 的所有外设部件，包括 5 个 16 位的计数器/定时器、两个全双工 UART、256 字节内部 RAM、128 字节特殊功能寄存器（SFR）地址空间及 8/4 个字节宽的 I/O 端口。

1.1.2 速度提高

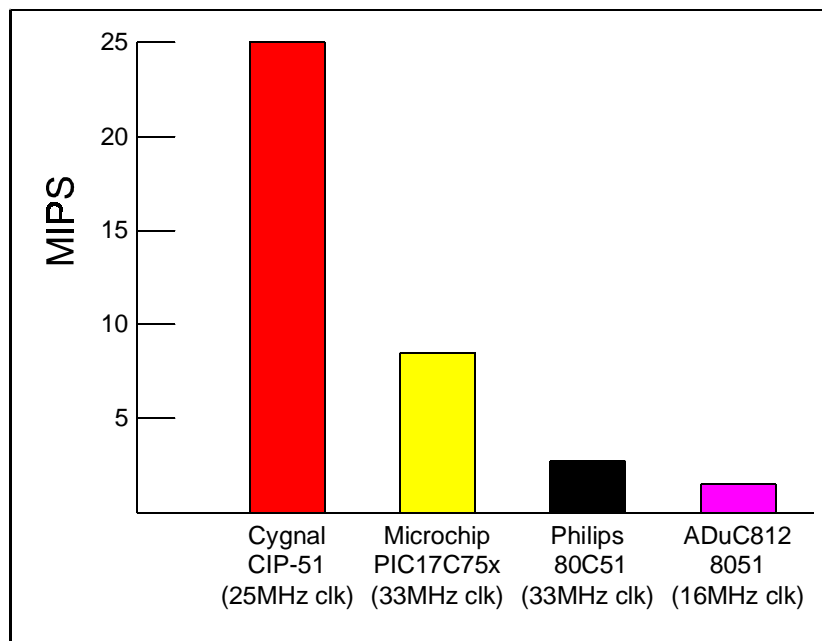
CIP-51 采用流水线结构，与标准的 8051 结构相比指令执行速度有很大的提高。在一个标准的 8051 中，除 MUL 和 DIV 以外所有指令都需要 12 或 24 个系统时钟周期，最大系统时钟频率为 12-24MHz。而对于 CIP-51 内核，70% 的指令的执行时间为 1 或 2 个系统时钟周期，只有 4 条指令的执行时间大于 4 个系统时钟周期。

CIP-51 共有 111 条指令。下表列出了指令条数与执行时所需的系统时钟周期数的关系。

| | | | | | | | | | |
|-------|----|----|-----|----|-----|---|-----|---|---|
| 执行周期数 | 1 | 2 | 2/3 | 3 | 3/4 | 4 | 4/5 | 5 | 8 |
| 指令数 | 26 | 50 | 5 | 16 | 7 | 3 | 1 | 2 | 1 |

CIP-51 工作在最大系统时钟频率 25MHz 时，它的峰值速度达到 25MIPS。图 1.5 给出了几种 8 位微控制器内核工作在最大系统时钟时的峰值速度的比较关系。

图 1.5 MCU 峰值执行速度比较



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

1.1.3 增加的功能

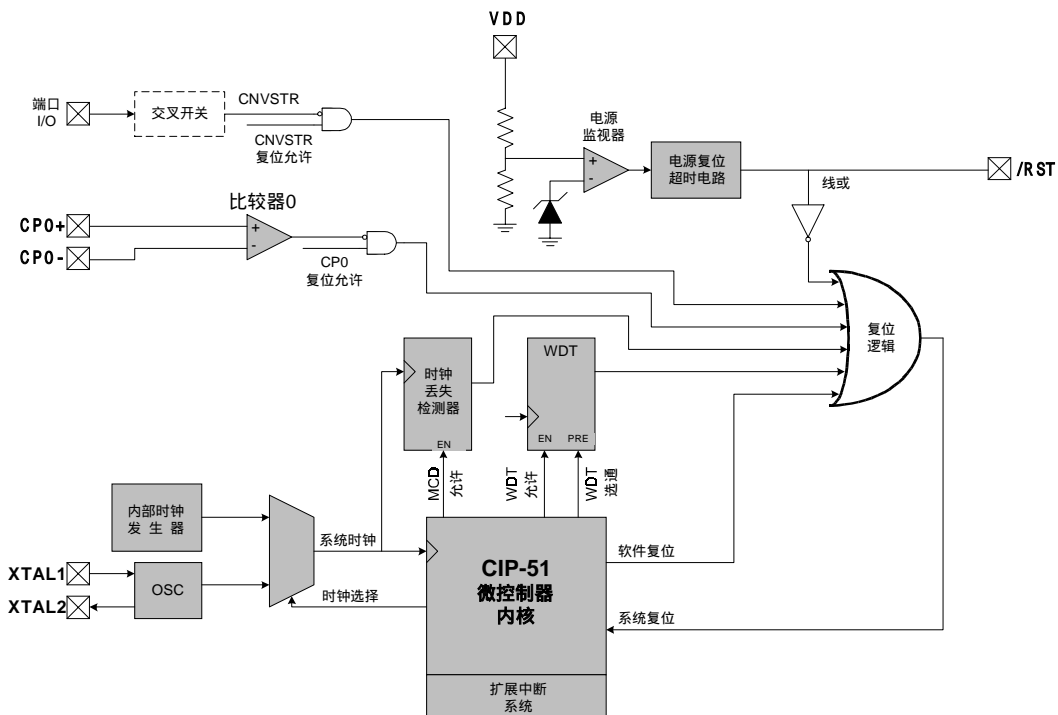
C8051F020 系列 MCU 在 CIP-51 内核和外设有几项关键性的改进，提高了整体性能，更易于在最终应用中使用。

扩展的中断系统向 CIP-51 提供 22 个中断源（标准 8051 只有 7 个中断源），允许大量的模拟和数字外设中断微控制器。一个中断驱动的系统需要较少的 MCU 干预，因而有更高的执行效率。在设计一个多任务实时系统时，这些增加的中断源是非常有用的。

MCU 可有多达 7 个复位源：一个片内 VDD 监视器、一个看门狗定时器、一个时钟丢失检测器、一个由比较器 0 提供的电压检测器、一个软件强制复位、CNVSTR 引脚及 /RST 引脚。/RST 引脚是双向的，可接受外部复位或将内部产生的上电复位信号输出到 /RST 引脚。除了 VDD 监视器和复位输入引脚以外，每个复位源都可以由用户用软件禁止；使用 MONEN 引脚允许/禁止 VDD 监视器。在一次上电复位之后的 MCU 初始化期间，WDT 可以被永久性使能。

MCU 内部有一个独立运行的时钟发生器，在复位后被默认为系统时钟。如果需要，时钟源可以在运行时切换到外部振荡器，外部振荡器可以使用晶体、陶瓷谐振器、电容、RC 或外部时钟源产生系统时钟。时钟切换功能在低功耗系统中是非常有用的，它允许 MCU 从一个低频率（节电）外部晶体源运行，当需要时再周期性地切换到高速（可达 16MHz）的内部振荡器。

图 1.6 片内时钟和复位



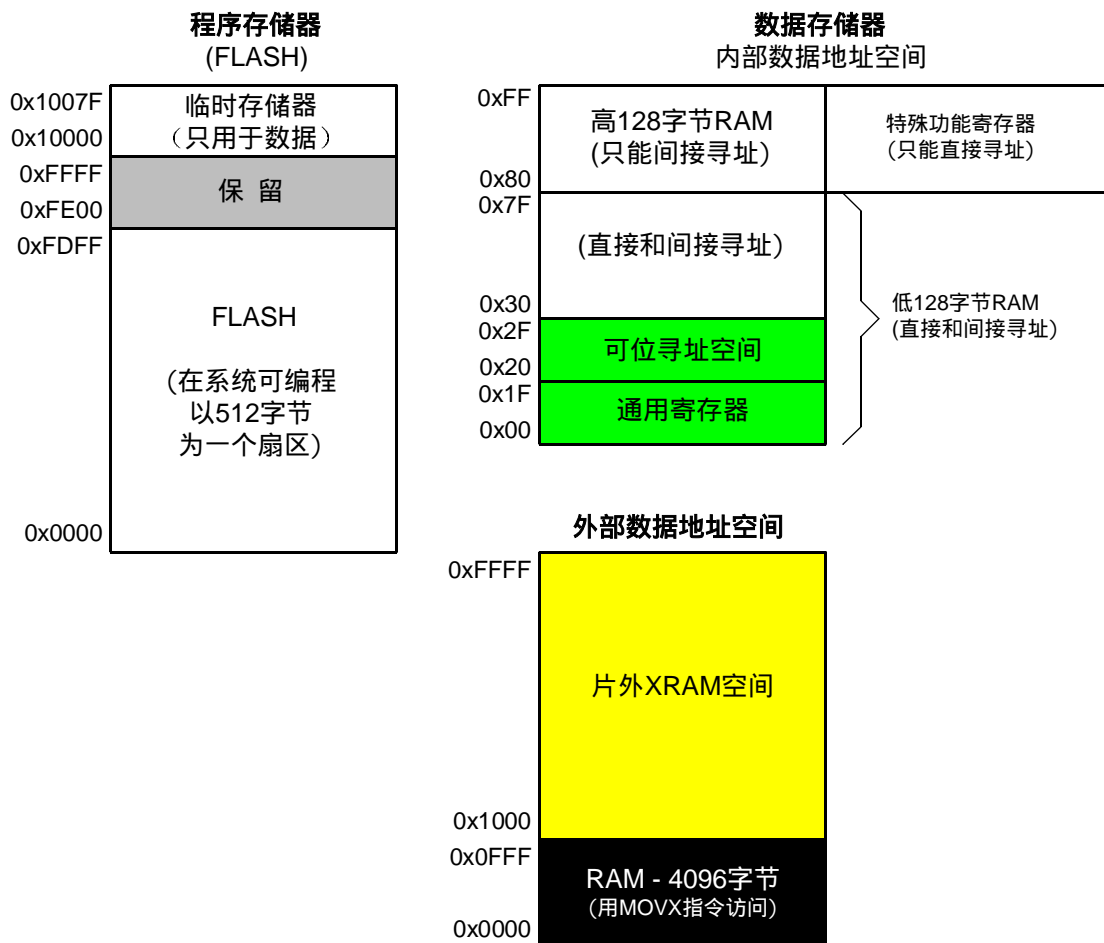
1.2 片内存储器

CIP-51 有标准的 8051 程序和数据地址配置。它包括 256 字节的数据 RAM，其中高 128 字节为双映射。用间接寻址访问通用 RAM 的高 128 字节，用直接寻址访问 128 字节的 SFR 地址空间。数据 RAM 的低 128 字节可用直接或间接寻址方式访问。前 32 个字节为 4 个通用寄存器区，接下来的 16 字节既可以按字节寻址也可以按位寻址。

C8051F020/1/2/3 中的 CIP-51 还另有位于外部数据存储器地址空间的 4K 字节的 RAM 块和一个可用于访问外部数据存储器的外部存储器接口（EMIF）。这个片内的 4K 字节 RAM 块可以在整个 64K 外部数据存储器地址空间中被寻址（以 4K 为边界重叠）。外部数据存储器地址空间可以只映射到片内存储器、只映射到片外存储器、或两者的组合（4K 以下的地址指向片内，4K 以上的地址指向 EMIF）。EMIF 可以被配置为地址/数据线复用方式或非复用方式。

MCU 的程序存储器包含 64K 字节的 FLASH。该存储器以 512 字节为一个扇区，可以在系统编程，且不需特别的编程电压。从 0xFE00 到 0xFFFF 的 512 字节被保留，由工厂使用。还有一个位于地址 0x10000 - 0x1007F 的 128 字节的扇区，该扇区可作为一个小的软件常数表使用。图 1.7 给出了 MCU 系统的存储器结构。

图 1.7 片内存储器组织



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

1.3 JTAG 调试和边界扫描

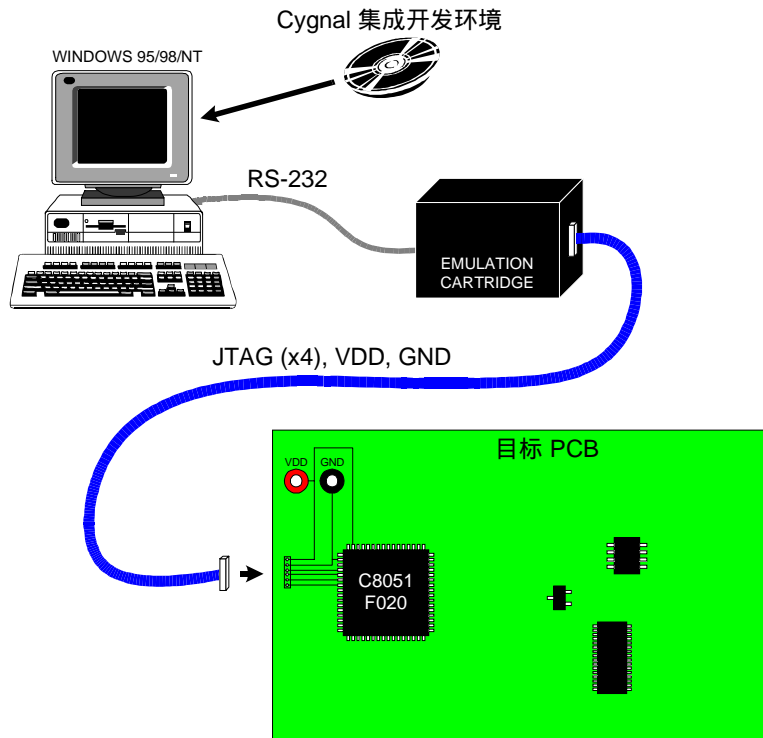
C8051F020系列具有片内JTAG边界扫描和调试电路,通过4脚JTAG接口并使用安装在最终应用系统中的产品器件就可以进行非侵入式、全速的在系统调试。该JTAG接口完全符合IEEE 1149.1规范,为生产和测试提供完全的边界扫描功能。

Cygnal的调试系统支持观察和修改存储器和寄存器,支持断点、观察点、堆栈指示器和单步执行。不需要额外的目标RAM、程序存储器、定时器或通信通道。在调试时所有的模拟和数字外设都正常工作。当MCU单步执行或遇到断点而停止运行时,所有的外设(ADC和SMBus除外)都停止运行,以保持同步。

开发套件C8051F020DK具有开发应用代码所需要的全部硬件和软件,并可对C8051F020/1/2/3 MCU进行在系统调试。开发套件中包括开发者工作室软件和调试器、一个集成的8051汇编器和一个RS-232转换到JTAG的串行适配器。套件中还有一个目标应用板,上面有对应的MCU和一大块样机区域。套件中还包括RS-232和JTAG电缆及一个墙装电源。开发套件需要一个运行Windows 95/98/Me/NT并有一个可用RS-232串口的计算机。如图1.8所示,PC机通过RS-232与串行适配器连接。一条六英寸的扁平电缆将串行适配器和用户的应用板连接起来,使用4个JTAG引脚和VDD及GND。串行适配器从应用板获取其电源,在2.7-3.6V时其电源电流大约为20 mA。对于不能从目标板上获取足够电流的应用,可以将套件中提供的电源直接连到串行适配器上。

对于开发和调试嵌入式应用来说,该系统的调试功能比采用标准MCU仿真器要优越得多。标准的MCU仿真器要使用在板仿真芯片和目标电缆,还需要在应用板上有MCU的插座。Cygnal的调试环境既便于使用又能保证精确模拟外设的性能。

图1.8 调试环境示意图



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

1.4 可编程数字 I/O 和交叉开关

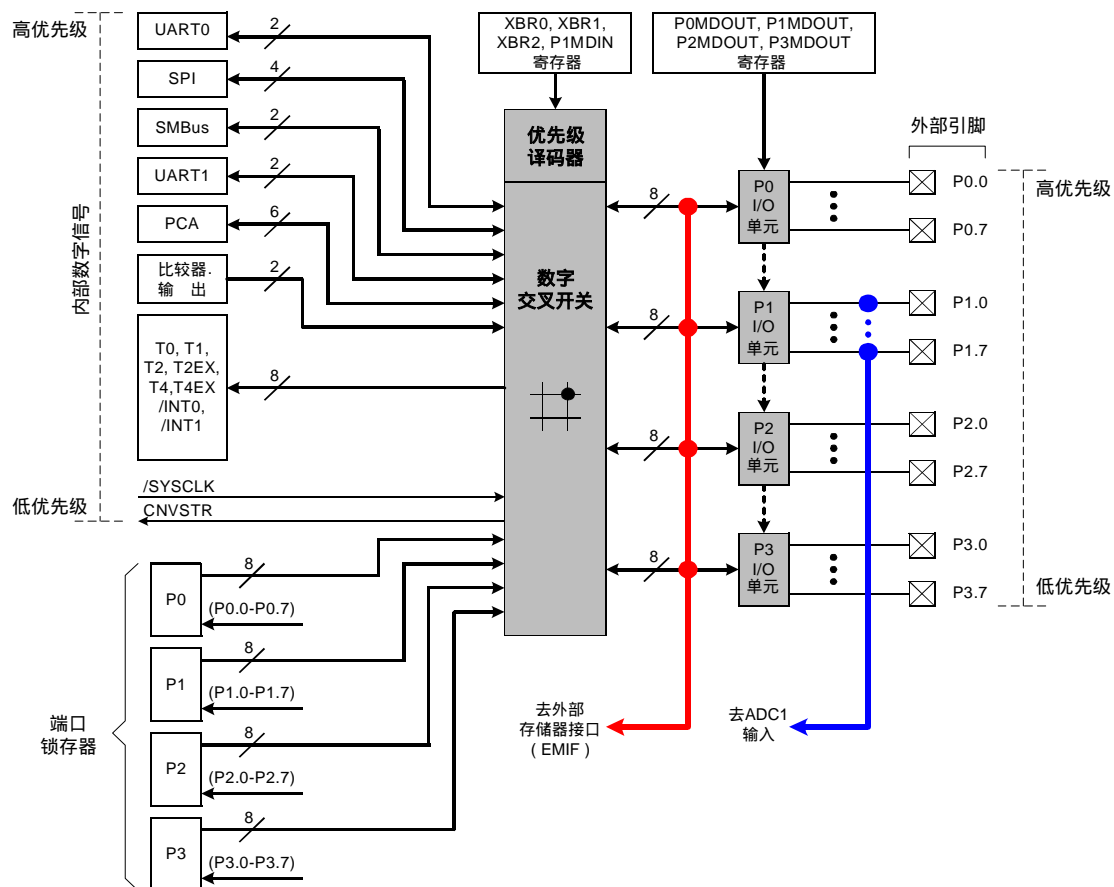
该系列MCU具有标准8051的端口（0、1、2和3）。在F020/2中有4个附加的端口（4、5、6和7），因此共有64个通用端口I/O。这些端口I/O的工作情况与标准8051相似，但有一些改进。

每个端口I/O引脚都可以被配置为推挽或漏极开路输出。在标准8051中固定的“弱上拉”可以被总体禁止，这为低功耗应用提供了进一步节电的能力。

可能最独特的改进是引入了数字交叉开关。这是一个大的数字开关网络，允许将内部数字系统资源映射到P0、P1、P2和P3的端口I/O引脚（见图1.9）。与具有标准复用数字I/O的微控制器不同，这种结构可支持所有的功能组合。

可通过设置交叉开关控制寄存器将片内的计数器/定时器、串行总线、硬件中断、ADC转换启动输入、比较器输出以及微控制器内部的其它数字信号配置为出现在端口I/O引脚。这一特性允许用户根据自己的特定应用选择通用端口I/O和所需数字资源的组合。

图1.9 数字交叉开关原理框图

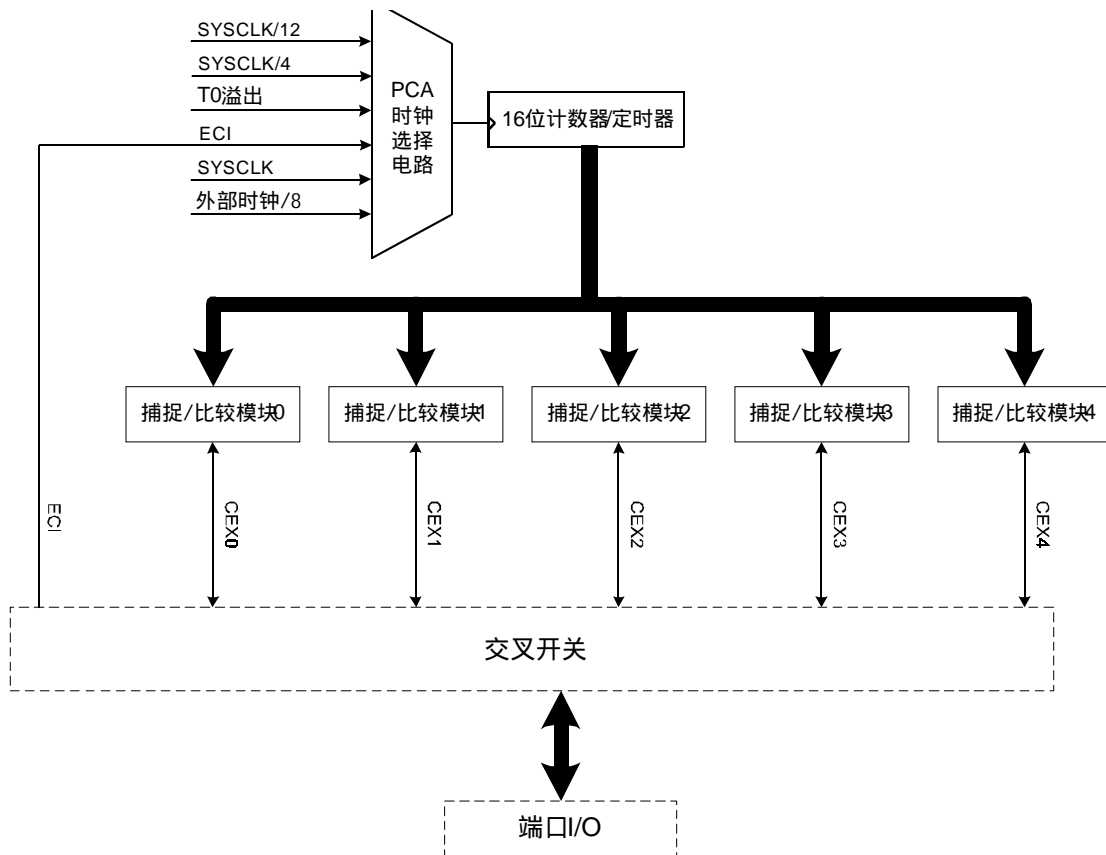


1.5 可编程计数器阵列

除了5个16位的通用计数器/定时器之外，C8051F020 MCU系列还有一个片内可编程计数器/定时器阵列（PCA）。PCA包括一个专用的16位计数器/定时器时间基准和5个可编程的捕捉/比较模块。时间基准的时钟可以是下面的六个时钟源之一：系统时钟/12、系统时钟/4、定时器0溢出、外部时钟输入（ECI）、系统时钟和外部振荡源频率/8。

每个捕捉/比较模块都有六种工作方式：边沿触发捕捉、软件定时器、高速输出、频率输出、8位脉冲宽度调制器和16位脉冲宽度调制器。PCA捕捉/比较模块的I/O和外部时钟输入可以通过数字交叉开关连到MCU的端口I/O引脚。

图1.10 PCA原理框图



1.6 串行端口

C8051F020系列MCU内部有两个增强型全双工UART、SPI总线和SMBus/I²C。每种串行总线都完全用硬件实现，都能向CIP-51产生中断，因此需要很少的CPU干预。这些串行总线不“共享”定时器、中断或端口I/O等资源，所以可以使用任何一个或全部同时使用。

1.7 12 位模/数转换器

C8051F020/1有一个片内12位SAR ADC (ADC0)，一个9通道输入多路选择开关和可编程增益放大器。该ADC工作在100kpsps的最大采样速率时可提供真正的12位精度，INL为±1LSB。C8051F022/3有一个片内10位SAR ADC，技术指标和配置选项与C8051F020/1的ADC类似。ADC0的电压基准可以在DAC0输出和外部VREF引脚之间选择。对于C8051F020/2器件，ADC0有其专用的VREF0输入引脚；对于C8051F021/3器件，ADC0与8位的ADC1共享VREFA输入引脚。片内15ppm/°C的电压基准可通过VREF输出引脚为其它系统部件或片内ADC产生基准电压。

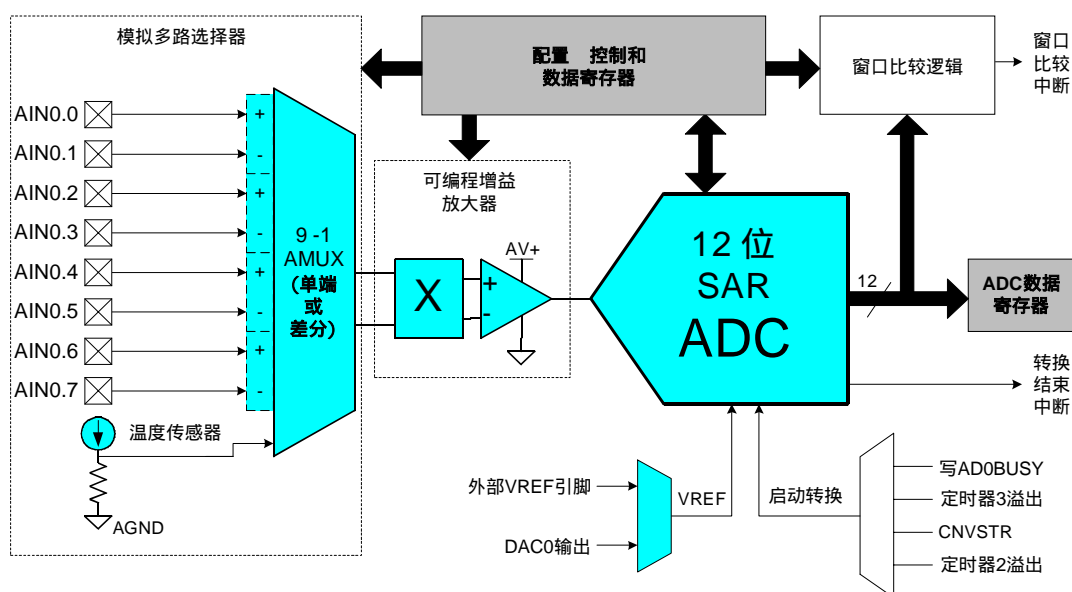
ADC完全由CIP-51通过特殊功能寄存器控制。有一个输入通道被连到内部温度传感器，其它8个通道接外部输入。8个外部输入通道的每一对都可被配置为两个单端输入或一个差分输入。系统控制器可以将ADC置于关断状态以节省功耗。

可编程增益放大器接在模拟多路选择器之后，增益可以用软件设置，从0.5到16以2的整数次幂递增。当不同ADC输入通道之间输入的电压信号范围差距较大或需要放大一个具有较大直流偏移的信号时（在差分方式，DAC可用于提供直流偏移），这个放大环节是非常有用的。

A/D转换有4种启动方式：软件命令、定时器2溢出、定时器3溢出和外部信号输入。这种灵活性允许用软件事件、外部硬件信号或周期性的定时器溢出信号触发转换。一次转换完成可以产生一个中断，或者用软件查询一个状态位来判断转换结束。在转换完成后，10或12位转换结果数据字被锁存到两个特殊功能寄存器中。这些数据字可以用软件控制为左对齐或右对齐。

窗口比较寄存器可被配置为当ADC数据位于一个规定的范围之内或之外时向控制器申请中断。ADC可以用后台方式监视一个关键电压，当转换数据位于规定的窗口之内时才向控制器申请中断。

图1.11 12位ADC原理框图



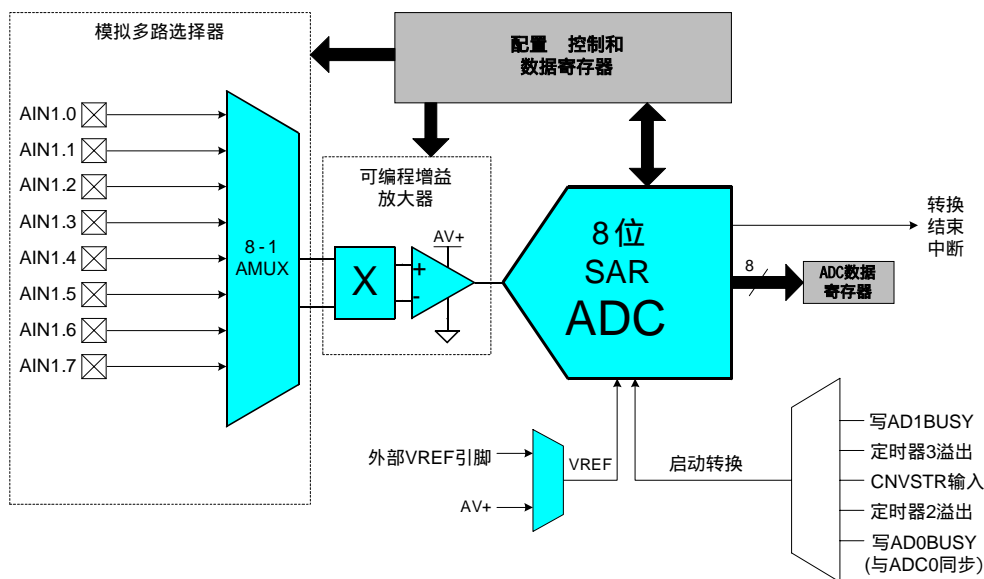
1.8 8 位模/数转换器

C8051F020/1/2/3有一个片内8位SAR ADC（ADC1），带有一个8通道输入多路选择器和可编程增益放大器。该ADC工作在500ksp/s的最大采样速率时可提供真正的8位精度，INL为±1LSB。有8个用于测量的输入端。ADC1完全由CIP-51通过特殊功能寄存器控制。ADC0的电压基准可以在模拟电源电压（AV+）和外部VREF引脚之间选择。对于C8051F020/2器件，ADC1有其专用的VREF1输入引脚；对于C8051F021/3器件，ADC1与12/10位的ADC0共享VREFA输入引脚。用户软件可以将ADC1置于关断状态以节省功耗。

可编程增益放大器接在模拟多路选择器之后。当不同ADC输入通道之间输入的电压信号范围差距较大或需要局部放大一个具有较大直流偏移的信号时（在差分方式，DAC可用于提供直流偏移），这个放大环节是非常有用的。PGA增益可以用软件设置为0.5、1、2或4。

灵活的转换控制系统允许用软件命令、定时器溢出或外部信号启动ADC1转换。用软件命令可以使ADC1与ADC0同步转换。一次转换完成可以产生中断（如果被允许），或者用软件查询一个状态位来判断转换结束。在转换完成后，8位数据字被锁存到一个特殊功能寄存器中。

图1.12 8位ADC原理框图



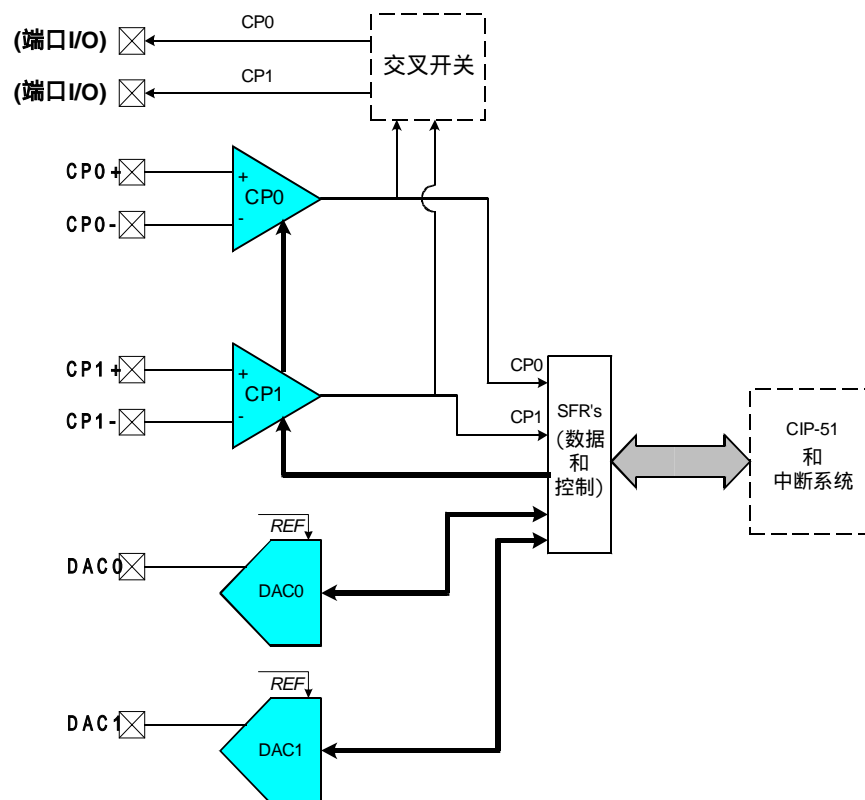
1.9 比较器和 DAC

C8051F020/1/2/3系列MCU内部有两个12位DAC和两个比较器。MCU与每个比较器和DAC之间的数据和控制接口通过特殊功能寄存器实现。MCU可以将任何一个DAC或比较器置于低功耗关断方式。

比较器的回差电压可以用软件编程。每个比较器都能在上升沿或下降沿产生中断，或在两个边沿都产生中断。这些中断能将MCU从等待方式唤醒。比较器的输出状态可以用软件查询。可通过设置交叉开关将比较器的输出接到端口I/O引脚。

DAC为电压输出方式，有灵活的输出更新机制。这一机制允许用软件写和定时器2、定时器3及定时器4的溢出信号更新DAC输出。C8051F020/2的DAC之电压基准由专用的VREFD输入引脚提供，而C8051F021/3的DAC之电压基准由器件内部的电压基准提供。DAC在作为比较器的参考电压或为ADC差分输入提供偏移电压时非常有用。

图1.13 比较器和DAC原理框图



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

2. 极限参数

表 2.1 极限参数*

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|----------------------------|-----|------|-----|-----------|-----|
| 环境温度（通电情况下） | | -55 | | 125 | °C |
| 储存温度 | | -65 | | 150 | °C |
| 任何引脚相对DGND的电压（VDD和端口I/O除外） | | -0.3 | | VDD + 0.3 | V |
| 任何端口I/O引脚或/RST相对DGND的电压 | | -0.3 | | 5.8 | V |
| VDD引脚相对DGND的电压 | | -0.3 | | 4.2 | V |
| 通过VDD、AV+、DGND和AGND的最大总电流 | | | | 800 | mA |
| 任何端口引脚的最大输出灌电流 | | | | 100 | mA |
| 任何其它I/O引脚的最大输出灌电流 | | | | 50 | mA |
| 任何端口引脚的最大输出拉电流 | | | | 100 | mA |
| 任何其它I/O引脚的最大输出拉电流 | | | | 50 | mA |

*注：超过这些列出的“极限参数”可能导致器件永久性损坏。

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

3. 总体直流电气特性

表 3.1 总体直流电气特性

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|-----------------------------|---|-----|-----------------|-----|----------------|
| 模拟电源电压 | (注1) | 2.7 | 3.0 | 3.6 | V |
| 模拟电源电流 | 内部REF、ADC、DAC、比较器都工作 | | 1.7 | | mA |
| 模拟电源电流 (模拟子系统不工作) | 内部REF、ADC、DAC、比较器都不工作；振荡器被禁止；VDD监视器被禁止 | | 0.2 | | μA |
| 模拟与数字电源电压之差 (VDD-AV+) | | | | 0.5 | V |
| 数字电源电压 | | 2.7 | 3.0 | 3.6 | V |
| 数字电源电流 CPU工作 | VDD=2.7V, CLK=25MHz VDD=2.7V, CLK= 1MHz VDD=2.7V, CLK=32kHz | | 10 0.5 20 | | mA mA μA |
| 数字电源电流 CPU不工作 (不访问FLASH) | VDD=2.7V, CLK=25MHz VDD=2.7V, CLK= 1MHz VDD=2.7V, CLK=32kHz | | 5 0.2 10 | | mA mA μA |
| 数字电源电流 (停机方式) | VDD=2.7V, 振荡器不运行, VDD监视器被禁止 | | 0.2 | | μA |
| 数字电源 (RAM数据保持电压) | | | 1.5 | | V |
| 额定工作温度范围 | | -40 | | +85 | ℃ |

注 1: 模拟电源 AV+必须大于 1V 才能使 VDD 监视器工作。

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

4. 引脚和封装定义

表 4.1 引脚定义

| 引脚名称 | 引脚号 | | 类型 | 说明 |
|-------|--------------|---------------|--------|---|
| | F020 F022 | F021 F023 | | |
| VDD | 37,64, 90 | 24,41, 57 | | 数字电源。必须接 +2.7V ~ +3.6V。 |
| DGND | 38,63, 89 | 25,40, 56, | | 数字地 |
| AV+ | 11,14 | 6 | | 模拟电源。必须接 +2.7V ~ +3.6V。 |
| AGND | 10,13 | 5 | | 模拟地 |
| TMS | 1 | 58 | 数字输入 | 带内部上拉的 JTAG 测试模式选择 |
| TCK | 2 | 59 | 数字输入 | 带内部上拉的 JTAG 测试时钟 |
| TDI | 3 | 60 | 数字输入 | 带内部上拉的 JTAG 测试数据输入。TDI 在 TCK 上升沿被锁存。 |
| TDO | 4 | 61 | 数字输出 | 带内部上拉的 JTAG 测试数据输出。数据在 TCK 的下降沿从 TDO 引脚输出。TDO 输出是一个三态驱动器。 |
| /RST | 5 | 62 | 数字 I/O | 器件复位。内部 VDD 监视器的漏极开路输出，当 $VDD < 2.7V$ 并且 MONEN 为高时被驱动为低电平。一个外部源可以通过将该引脚置为低电平启动一次系统复位。 |
| XTAL1 | 26 | 17 | 模拟输入 | 晶体输入。该引脚为晶体或陶瓷谐振器的内部振荡器电路的反馈输入。为了得到一个精确的内部时钟，可以在 XTAL1 和 XTAL2 之间接上一个晶体或陶瓷谐振器。如果被一个外部 CMOS 时钟驱动，则该引脚提供系统时钟。 |
| XTAL2 | 27 | 18 | 模拟输出 | 晶体输出。该引脚是晶体或陶瓷谐振器的激励驱动器。 |
| MONEN | 28 | 19 | 数字输入 | VDD 监视器允许。该引脚接高电平时允许内部 VDD 监视器工作，当 $VDD < 2.7V$ 时强制系统复位。该引脚接低电平时内部 VDD 监视器被禁止。 |
| VREF | 12 | 7 | 模拟 I/O | 带隙电压基准输出（所有器件）。 DAC 电压基准输入（只限于 F021/3）。 |
| VREFA | | 8 | 模拟输入 | ADC0 和 ADC1 的电压基准输入。 |
| VREF0 | 16 | | 模拟输入 | ADC0 的电压基准输入。 |
| VREF1 | 17 | | 模拟输入 | ADC1 的电压基准输入。 |
| VRED | 15 | | 模拟输入 | DAC 的电压基准输入。 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 4.1 引脚定义 (续)

| 引脚名称 | 引脚号 | | 类型 | 说明 |
|----------|--------------|--------------|--------|---|
| | F020 F022 | F021 F023 | | |
| AIN0.0 | 18 | 9 | 模拟输入 | ADC0 输入通道 0 (详见 ADC0 说明) |
| AIN0.1 | 19 | 10 | 模拟输入 | ADC0 输入通道 1 (详见 ADC0 说明) |
| AIN0.2 | 20 | 11 | 模拟输入 | ADC0 输入通道 2 (详见 ADC0 说明) |
| AIN0.3 | 21 | 12 | 模拟输入 | ADC0 输入通道 3 (详见 ADC0 说明) |
| AIN0.4 | 22 | 13 | 模拟输入 | ADC0 输入通道 4 (详见 ADC0 说明) |
| AIN0.5 | 23 | 14 | 模拟输入 | ADC0 输入通道 5 (详见 ADC0 说明) |
| AIN0.6 | 24 | 15 | 模拟输入 | ADC0 输入通道 6 (详见 ADC0 说明) |
| AIN0.7 | 25 | 16 | 模拟输入 | ADC0 输入通道 7 (详见 ADC0 说明) |
| CP0+ | 9 | 4 | 模拟输入 | 比较器 0 的同相输入端 |
| CP0- | 8 | 3 | 模拟输入 | 比较器 0 的反相输入端 |
| CP1+ | 7 | 2 | 模拟输入 | 比较器 1 的同相输入端 |
| CP1- | 6 | 1 | 模拟输入 | 比较器 1 的反相输入端 |
| DAC0 | 100 | 64 | 模拟输出 | 数模转换器 0 的电压输出。(见 DAC 说明) |
| DAC1 | 99 | 63 | 模拟输出 | 数模转换器 1 的电压输出。(见 DAC 说明) |
| P0.0 | 62 | 55 | 数字 I/O | P0.0。详见端口输入/输出部分。 |
| P0.1 | 61 | 54 | 数字 I/O | P0.1。详见端口输入/输出部分。 |
| P0.2 | 60 | 53 | 数字 I/O | P0.2。详见端口输入/输出部分。 |
| P0.3 | 59 | 52 | 数字 I/O | P0.3。详见端口输入/输出部分。 |
| P0.4 | 58 | 51 | 数字 I/O | P0.4。详见端口输入/输出部分。 |
| ALE/P0.5 | 57 | 50 | 数字 I/O | 外部存储器地址总线 ALE 选通 (复用方式)。 P0.5。详见端口输入/输出部分。 |
| /RD/P0.6 | 56 | 49 | 数字 I/O | 外部存储器接口的/RD 选通。 P0.6。详见端口输入/输出部分。 |
| /WR/P0.7 | 55 | 48 | 数字 I/O | 外部存储器接口的/WR 选通。 P0.7。详见端口输入/输出部分。 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 4.1 引脚定义 (续)

| 引脚名称 | 引脚号 | | 类型 | 说明 |
|-----------------|--------------|--------------|----------------|---|
| | F020 F022 | F021 F023 | | |
| AIN1.0/A8/P1.0 | 36 | 29 | 模拟输入 | ADC1 输入通道 0 (详见 ADC1 说明)。外部存储器地址总线位 8 (非复用方式)。P1.0。详见端口输入/输出部分。 |
| AIN1.1/A9/P1.1 | 35 | 28 | 模拟输入 数字 I/O | P1.1。详见端口输入/输出部分。 |
| AIN1.2/A10/P1.2 | 34 | 27 | 模拟输入 数字 I/O | P1.2。详见端口输入/输出部分。 |
| AIN1.3/A11/P1.3 | 33 | 26 | 模拟输入 数字 I/O | P1.3。详见端口输入/输出部分。 |
| AIN1.4/A12/P1.4 | 32 | 23 | 模拟输入 数字 I/O | P1.4。详见端口输入/输出部分。 |
| AIN1.5/A13/P1.5 | 31 | 22 | 模拟输入 数字 I/O | P1.5。详见端口输入/输出部分。 |
| AIN1.6/A14/P1.6 | 30 | 21 | 模拟输入 数字 I/O | P1.6。详见端口输入/输出部分。 |
| AIN1.7/A15/P1.7 | 29 | 20 | 模拟输入 数字 I/O | P1.7。详见端口输入/输出部分。 |
| A8m/A0/P2.0 | 46 | 37 | 数字 I/O | 外部存储器地址总线位 8 (复用方式)。外部存储器地址总线位 0 (非复用方式)。P2.0。详见端口输入/输出部分。 |
| A9m/A1/P2.1 | 45 | 36 | 数字 I/O | P2.1。详见端口输入/输出部分。 |
| A10m/A2/P2.2 | 44 | 35 | 数字 I/O | P2.2。详见端口输入/输出部分。 |
| A11m/A3/P2.3 | 43 | 34 | 数字 I/O | P2.3。详见端口输入/输出部分。 |
| A12m/A4/P2.4 | 42 | 33 | 数字 I/O | P2.4。详见端口输入/输出部分。 |
| A13m/A5/P2.5 | 41 | 32 | 数字 I/O | P2.5。详见端口输入/输出部分。 |
| A14m/A6/P2.6 | 40 | 31 | 数字 I/O | P2.6。详见端口输入/输出部分。 |
| A15m/A7/P2.7 | 39 | 30 | 数字 I/O | P2.7。详见端口输入/输出部分。 |
| AD0/D0/P3.0 | 54 | 47 | 数字 I/O | 外部存储器地址/数据总线位 0 (复用方式)。外部存储器数据总线位 0 (非复用方式)。P3.0。详见端口输入/输出部分。 |
| AD1/D1/P3.1 | 53 | 46 | 数字 I/O | P3.1。详见端口输入/输出部分。 |
| AD2/D2/P3.2 | 52 | 45 | 数字 I/O | P3.2。详见端口输入/输出部分。 |
| AD3/D3/P3.3 | 51 | 44 | 数字 I/O | P3.3。详见端口输入/输出部分。 |
| AD4/D4/P3.4 | 50 | 43 | 数字 I/O | P3.4。详见端口输入/输出部分。 |
| AD5/D5/P3.5 | 49 | 42 | 数字 I/O | P3.5。详见端口输入/输出部分。 |
| AD6/D6/P3.6/IE6 | 48 | 39 | 数字 I/O | P3.6。详见端口输入/输出部分。 |
| AD7/D7/P3.7/IE7 | 47 | 38 | 数字 I/O | P3.7。详见端口输入/输出部分。 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 4.1 引脚定义 (续)

| 引脚名称 | 引脚号 | | 类型 | 说明 |
|----------|--------------|--------------|--------|---|
| | F020 F022 | F021 F023 | | |
| P4.0 | 98 | | 数字 I/O | P4.0。详见端口输入/输出部分。 |
| P4.1 | 97 | | 数字 I/O | P4.1。详见端口输入/输出部分。 |
| P4.2 | 96 | | 数字 I/O | P4.2。详见端口输入/输出部分。 |
| P4.3 | 95 | | 数字 I/O | P4.3。详见端口输入/输出部分。 |
| P4.4 | 94 | | 数字 I/O | P4.4。详见端口输入/输出部分。 |
| ALE/P4.5 | 93 | | 数字 I/O | 外部存储器地址总线 ALE 选通 (复用方式)。 P4.5。详见端口输入/输出部分。 |
| /RD/P4.6 | 92 | | 数字 I/O | 外部存储器接口的/RD 选通。 P4.6。详见端口输入/输出部分。 |
| /WR/P4.7 | 91 | | 数字 I/O | 外部存储器接口的/WR 选通。 P4.7。详见端口输入/输出部分。 |
| A8/P5.0 | 88 | | 数字 I/O | 外部存储器地址总线位 8 (非复用方式)。 P5.0。详见端口输入/输出部分。 |
| A9/P5.1 | 87 | | 数字 I/O | P5.1。详见端口输入/输出部分。 |
| A10/P5.2 | 86 | | 数字 I/O | P5.2。详见端口输入/输出部分。 |
| A11/P5.3 | 85 | | 数字 I/O | P5.3。详见端口输入/输出部分。 |
| A12/P5.4 | 84 | | 数字 I/O | P5.4。详见端口输入/输出部分。 |
| A13/P5.5 | 83 | | 数字 I/O | P5.5。详见端口输入/输出部分。 |
| A14/P5.6 | 82 | | 数字 I/O | P5.6。详见端口输入/输出部分。 |
| A15/P5.7 | 81 | | 数字 I/O | P5.7。详见端口输入/输出部分。 |
| A8/P5.0 | 88 | | 数字 I/O | 外部存储器地址总线位 8 (非复用方式)。 P5.0。详见端口输入/输出部分。 |
| A9/P5.1 | 87 | | 数字 I/O | P5.1。详见端口输入/输出部分。 |
| A10/P5.2 | 86 | | 数字 I/O | P5.2。详见端口输入/输出部分。 |
| A11/P5.3 | 85 | | 数字 I/O | P5.3。详见端口输入/输出部分。 |
| A12/P5.4 | 84 | | 数字 I/O | P5.4。详见端口输入/输出部分。 |
| A13/P5.5 | 83 | | 数字 I/O | P5.5。详见端口输入/输出部分。 |
| A14/P5.6 | 82 | | 数字 I/O | P5.6。详见端口输入/输出部分。 |
| A15/P5.7 | 81 | | 数字 I/O | P5.7。详见端口输入/输出部分。 |

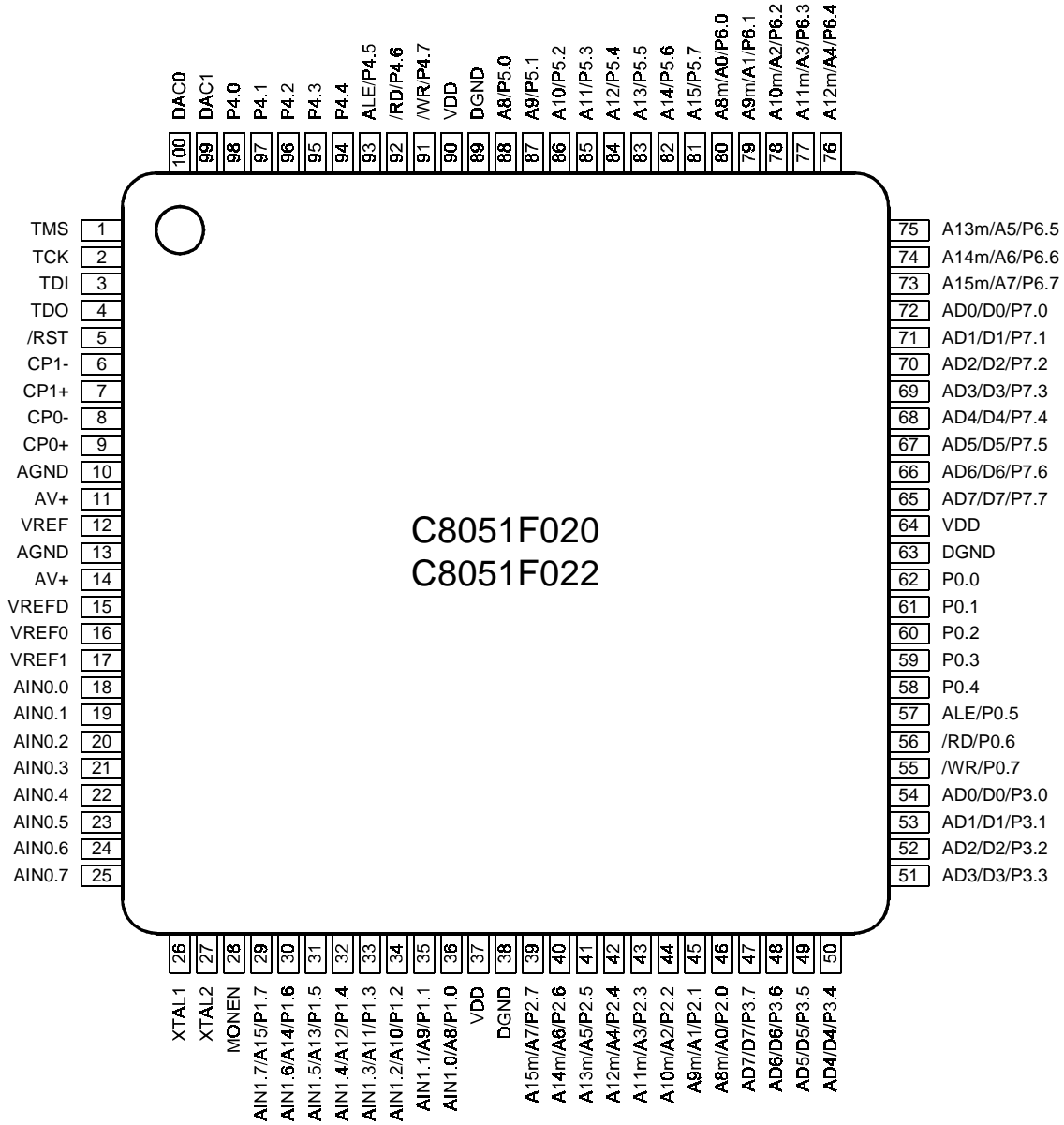
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 4.1 引脚定义 (续)

| 引脚名称 | 引脚号 | | 类型 | 说明 |
|--------------|--------------|--------------|--------|---|
| | F020 F022 | F021 F023 | | |
| A8m/A0/P6.0 | 80 | | 数字 I/O | 外部存储器地址总线位 8 (复用方式)。 外部存储器地址总线位 0 (非复用方式)。 P6.0。详见端口输入/输出部分。 |
| A9m/A1/P6.1 | 79 | | 数字 I/O | P6.1。详见端口输入/输出部分。 |
| A10m/A2/P6.2 | 78 | | 数字 I/O | P6.2。详见端口输入/输出部分。 |
| A11m/A3/P6.3 | 77 | | 数字 I/O | P6.3。详见端口输入/输出部分。 |
| A12m/A4/P6.4 | 76 | | 数字 I/O | P6.4。详见端口输入/输出部分。 |
| A13m/A5/P6.5 | 75 | | 数字 I/O | P6.5。详见端口输入/输出部分。 |
| A14m/A6/P6.6 | 74 | | 数字 I/O | P6.6。详见端口输入/输出部分。 |
| A15m/A7/P6.7 | 73 | | 数字 I/O | P6.7。详见端口输入/输出部分。 |
| AD0/D0/P7.0 | 72 | | 数字 I/O | 外部存储器地址/数据总线位 0 (复用方式)。 外部存储器数据总线位 0 (非复用方式)。 P7.0。详见端口输入/输出部分。 |
| AD1/D1/P7.1 | 71 | | 数字 I/O | P7.1。详见端口输入/输出部分。 |
| AD2/D2/P7.2 | 70 | | 数字 I/O | P7.2。详见端口输入/输出部分。 |
| AD3/D3/P7.3 | 69 | | 数字 I/O | P7.3。详见端口输入/输出部分。 |
| AD4/D4/P7.4 | 68 | | 数字 I/O | P7.4。详见端口输入/输出部分。 |
| AD5/D5/P7.5 | 67 | | 数字 I/O | P7.5。详见端口输入/输出部分。 |
| AD6/D6/P7.6 | 66 | | 数字 I/O | P7.6。详见端口输入/输出部分。 |
| AD7/D7/P7.7 | 65 | | 数字 I/O | P7.7。详见端口输入/输出部分。 |

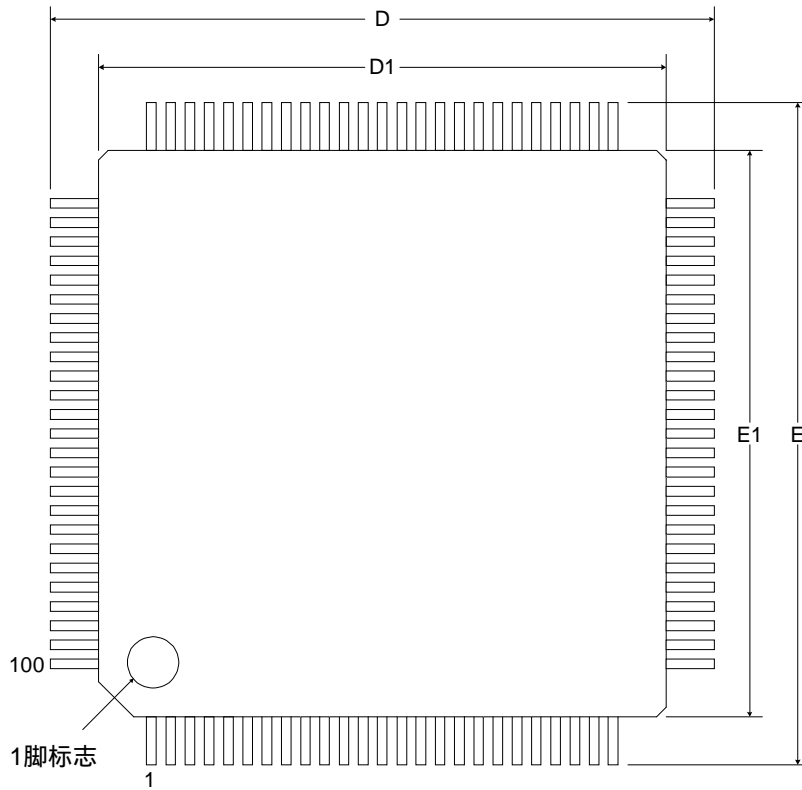
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 4.1 TQFP-100 引脚图

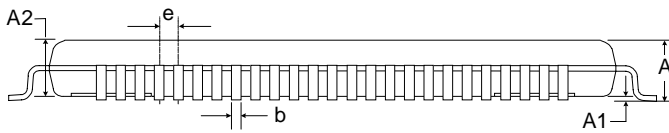


C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 4.2 TQFP-100 封装图

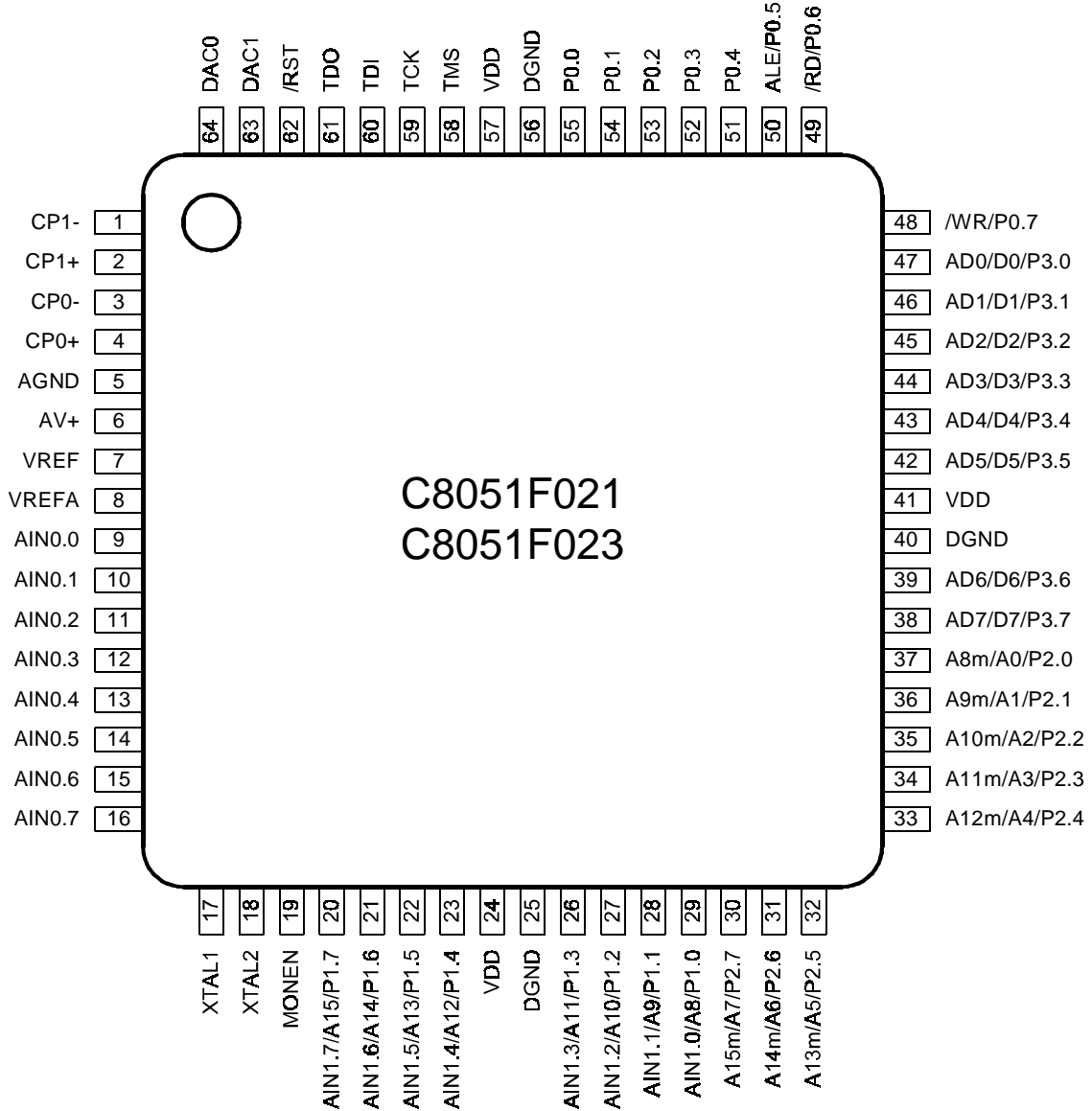


| | MIN (mm) | NOM (mm) | MAX (mm) |
|----|----------|----------|----------|
| A | - | - | 1.20 |
| A1 | 0.05 | - | 0.15 |
| A2 | 0.95 | 1.00 | 1.05 |
| b | 0.17 | 0.22 | 0.27 |
| D | - | 16.00 | - |
| D1 | - | 14.00 | - |
| e | - | 0.50 | - |
| E | - | 16.00 | - |
| E1 | - | 14.00 | - |



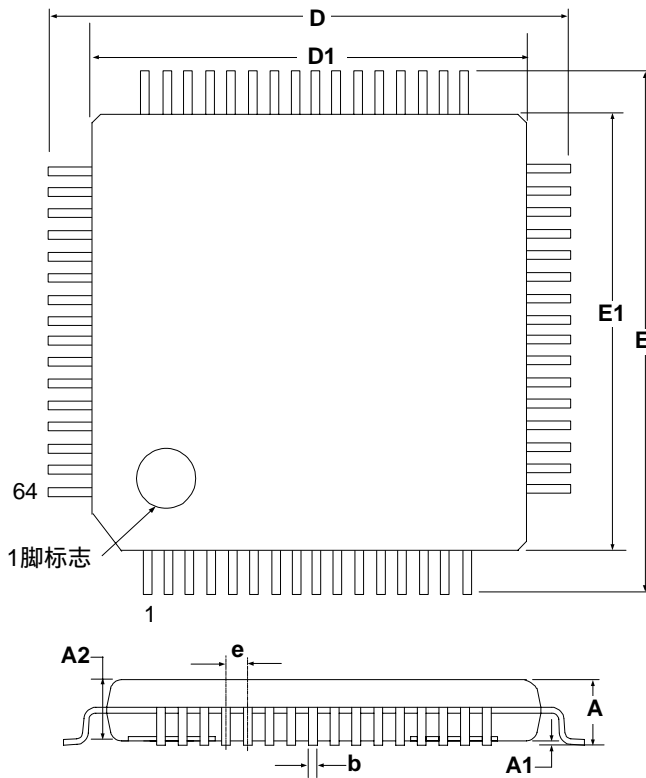
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 4.3 TQFP-64 引脚图



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 4.4 TQFP-64 封装图

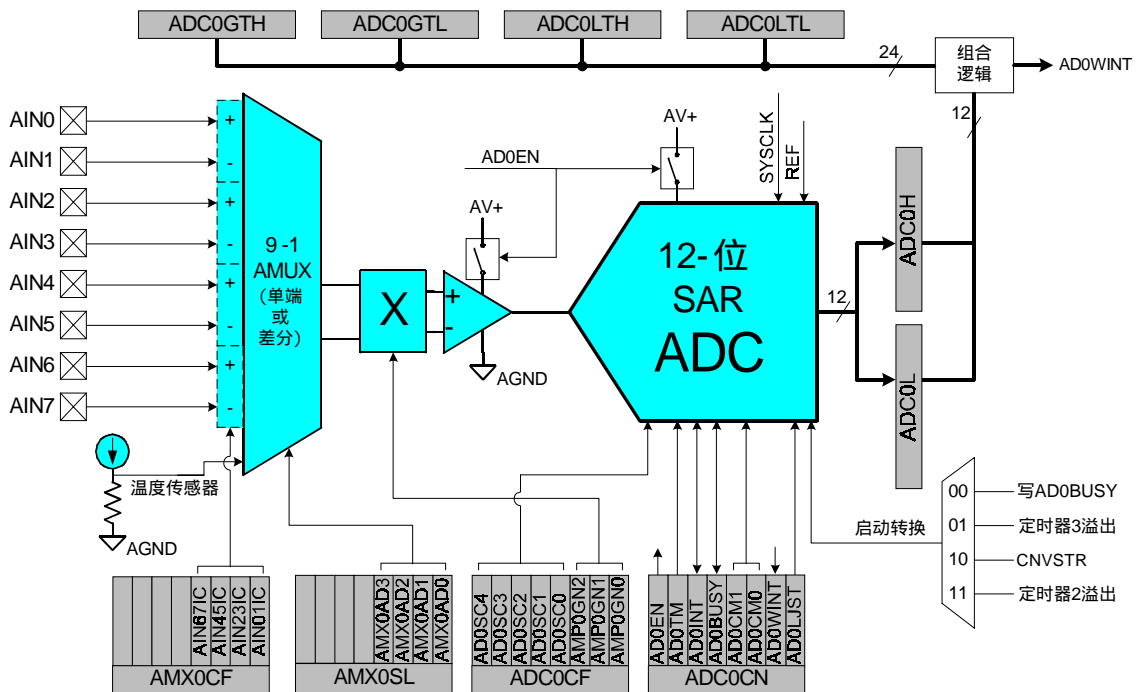


| | MIN (mm) | NOM (mm) | MAX (mm) |
|----|-------------|-------------|-------------|
| A | - | - | 1.20 |
| A1 | 0.05 | - | 0.15 |
| A2 | 0.95 | - | 1.05 |
| b | 0.17 | 0.22 | 0.27 |
| D | - | 12.00 | - |
| D1 | - | 10.00 | - |
| e | - | 0.50 | - |
| E | - | 12.00 | - |
| E1 | - | 10.00 | - |

5. ADC0（12 位，只限于 C8051F020/1）

C8051F020/1 的 ADC0 子系统包括一个 9 通道的可编程模拟多路选择器 (AMUX0)，一个可编程增益放大器 (PGA0) 和一个 100kpsps、12 位分辨率的逐次逼近寄存器型 ADC，ADC 中集成了跟踪保持电路和可编程窗口检测器 (见图 5.1 的原理框图)。AMUX0、PGA0、数据转换方式及窗口检测器都可用软件通过图 5.1 所示的特殊功能寄存器来控制。ADC0 所使用的电压基准按“9. 电压基准 (C8051F020/2)”或“10. 电压基准 (C8051F021/3)”所述选择。只有当 ADC0 控制寄存器中的 AD0EN 位被置‘1’时 ADC0 子系统 (ADC0、跟踪保持器和 PGA0) 才被允许工作。当 AD0EN 位为‘0’时，AD0C 子系统处于低功耗关断方式。

图 5.1 12 位 ADC0 功能框图



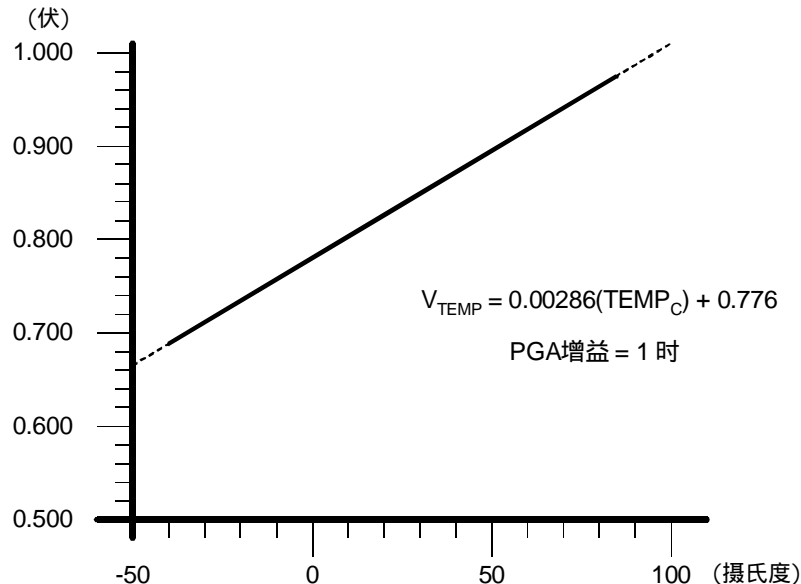
5.1 模拟多路开关和 PGA

AMUX 中的 8 个通道用于外部测量，而第九通道在内部被接到片内温度传感器 (温度传输函数示于图 5.2)。注意，PGA0 的增益对温度传感器也起作用。可以将 AMUX 输入对编程为工作在差分或单端方式。这就允许用户对每个通道选择最佳的测量技术，甚至可以在测量过程中改变方式。在系统复位后 AMUX 的默认方式为单端输入。有两个与 AMUX 相关的寄存器：通道选择寄存器 AMX0SL (图 5.6) 和配置寄存器 AMX0CF (图 5.7)。图 5.6 中的表给出了每种配置下各通道的功能。PGA 对 AMUX 输出信号的放大倍数由 ADC0 配置寄存器 ADC0CF (图 5.7) 中的 AMP0GN2-0 确定。PGA 增益可以用软件编程为 0.5、1、2、4、8 或 16，复位后的默认增益为 1。

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

温度传感器的传输函数示于图 5.2。当温度传感器被选中(用 AMX0SL 中的 AMX0AD3-0)时,其输出电压 (V_{TEMP}) 是 PGA 的输入;PGA 对该电压的放大倍数由用户编程的 PGA 设置值决定。

图 5.2 温度传感器传输函数



5.2 ADC 的工作方式

ADC0 的最高转换速度为 100kps,其转换时钟来源于系统时钟分频,分频值保存在寄存器 ADC0CF 的 ADCSC 位。

5.2.1 启动转换

有 4 种转换启动方式,由 ADC0CN 中的 ADC0 启动转换方式位 (AD0CM1, AD0CM0) 的状态决定。转换触发源有:

1. 向 ADC0CN 的 AD0BUSY 位写 1;
2. 定时器 3 溢出 (即定时的连续转换);
3. 外部 ADC 转换启动信号的上升沿, CNVSTR;
4. 定时器 2 溢出 (即定时的连续转换)。

AD0BUSY 位在转换期间被置 '1',转换结束后复 '0'。AD0BUSY 位的下降沿触发一个中断 (当被允许时)并将中断标志 AD0INT (ADC0CN.5)置 '1'。转换数据被保存在 ADC 数据字的 MSB 和 LSB 寄存器: ADC0H 和 ADC0L。转换数据在寄存器对 ADC0H:ADC0L 中的存储方式可以是左对齐或右对齐,由 ADC0CN 寄存器中 AD0LJST 位的编程状态决定。

当通过向 AD0BUSY 写 '1' 启动数据转换时,应查询 AD0INT 位以确定转换何时结束 (也可以使用 ADC0 中断)。建议的查询步骤如下:

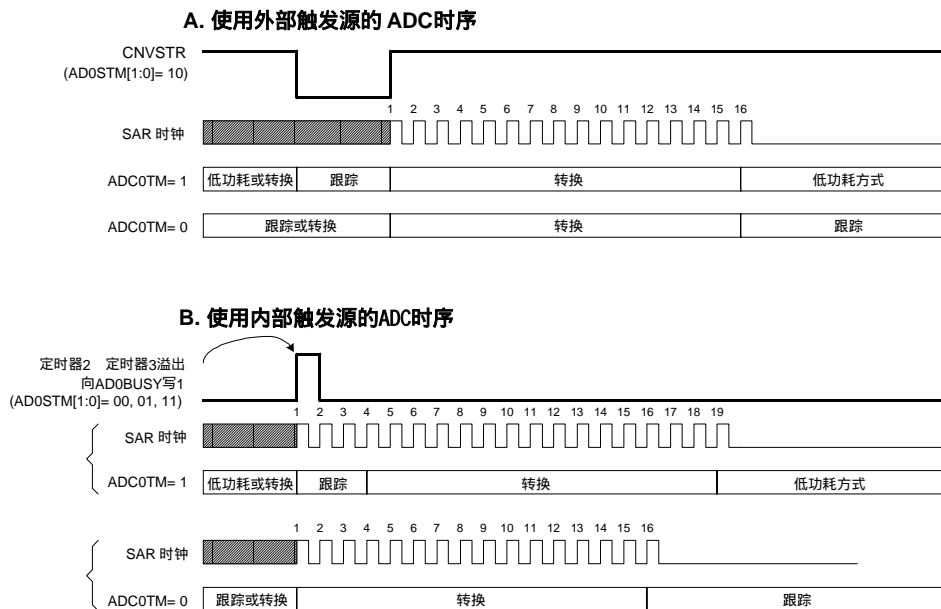
1. 写 '0' 到 AD0INT;
2. 向 ADBUSY 写 '1';

3. 查询并等待 AD0INT 变 ‘1’;
4. 处理 ADC0 数据

5.2.2 跟踪方式

ADC0CN 中的 AD0TM 位控制 ADC0 的跟踪保持方式。在缺省状态，除了转换期间之外 ADC0 输入被连续跟踪。当 AD0TM 位为逻辑 ‘1’ 时，ADC0 工作在低功耗跟踪保持方式。在该方式下，每次转换之前都有 3 个 SAR 时钟的跟踪周期（在启动转换信号有效之后）。当 CNVSTR 信号用于在低功耗跟踪保持方式启动转换时，ADC0 只在 CNVSTR 为低电平时跟踪；在 CNVSTR 的上升沿开始转换（见图 5.3）。当整个芯片处于低功耗待机或休眠方式时，跟踪可以被禁止（关断）。当 AMUX 或 PGA 的设置频繁改变时，低功耗跟踪保持方式也非常有用，可以保证建立时间需求得到满足（见“5.2.3 建立时间要求”）。

图 5.3 12 位 ADC 跟踪和转换时序举例



5.2.3 建立时间要求

当 ADC0 输入配置发生改变时 (AMUX 或 PGA 的选择发生变化), 在进行一次精确的转换之前需要有一个最小的跟踪时间。该跟踪时间由 ADC0 模拟多路器的电阻、ADC0 采样电容、外部信号源阻抗及所要求的转换精度决定。图 5.4 给出了单端和差分方式下等效的 ADC0 输入电路。注意: 这两种等效电路的时间常数完全相同。对于一个给定的建立精度 (SA), 所需要的 ADC0 建立时间可以用方程 5.1 估算。当测量温度传感器的输出时, R_{TOTAL} 等于 R_{MUX} 。注意: 在低功耗跟踪方式, 每次转换需要用三个 SAR 时钟跟踪。对于大多数应用, 三个 SAR 时钟可以满足跟踪需要。

方程 5.1 ADC0 建立时间要求

$$t = -\ln\left(\frac{SA}{2^n}\right) \times R_{TOTAL} C_{SAMPLE}$$

其中:

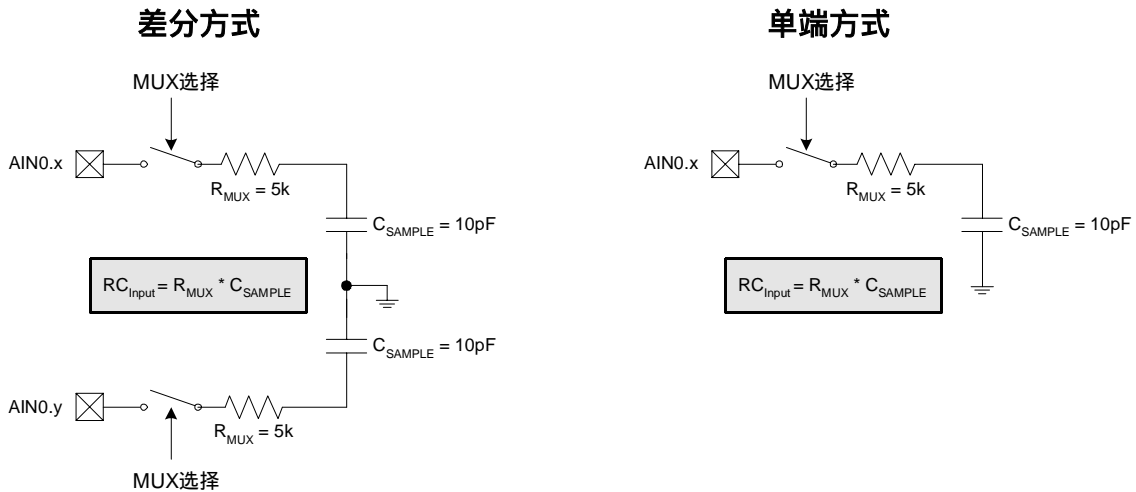
SA 是建立精度, 用一个 LSB 的分数表示 (例如, 建立精度 0.25 对应 1/4 LSB)

t 为所需要的建立时间, 以秒为单位

R_{TOTAL} 为 ADC0 模拟多路器电阻与外部信号源电阻之和

n 为 ADC0 的分辨率, 用比特表示 (12)。

图 5.4 ADC0 等效输入电路



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 5.5 AMX0CF: AMUX0 配置寄存器(C8051F020/1)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|-----|-----|-----|---------|---------|---------|---------|----------------|
| - | - | - | - | AIN67IC | AIN45IC | AIN23IC | AIN01IC | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xBA |
| <p>位 7-4: 未使用。读 = 0000b; 写 = 忽略</p> <p>位 3 AIN67IC: AIN6、AIN7 输入对配置位 0: AIN6 和 AIN7 为独立的单端输入 1: AIN6, AIN7 为 (分别为) +, -差分输入对</p> <p>位 2 AIN45IC: AIN4、AIN5 输入对配置位 0: AIN4 和 AIN5 为独立的单端输入 1: AIN4, AIN5 为 (分别为) +, -差分输入对</p> <p>位 1 AIN23IC: AIN2、AIN3 输入对配置位 0: AIN2 和 AIN2 为独立的单端输入 1: AIN2, AIN2 为 (分别为) +, -差分输入对</p> <p>位 0 AIN01IC: AIN0、AIN1 输入对配置位 0: AIN0 和 AIN1 为独立的单端输入 1: AIN0, AIN1 为 (分别为) +, -差分输入对</p> <p>注: 对于被配置成差分输入的通道, ADC0 数据字格式为 2 的补码。</p> | | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 5.6 AMUX0SL: AMUX0 通道选择寄存器(C8051F020/1)

| | | | | | | | | | | |
|--|-----|-----|-----|-----|---------|---------|---------|---------|-----|----------------|
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| | - | - | - | - | AMX0AD3 | AMX0AD2 | AMX0AD1 | AMX0AD0 | | 00000000 |
| | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | | SFR地址: 0xBB |

位 7-4: 未使用。读 = 0000b; 写 = 忽略
 位 3-0: AMX0AD3-0: AMUX0 地址位
 0000-1111b: 根据下表选择 ADC 输入

| | | AMX0AD3-0 | | | | | | | | |
|--------------|------|--------------------|------|--------------------|------|--------------------|------|--------------------|------|-------|
| | | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1xxx |
| AMX0CF 位 3-0 | 0000 | AIN0 | AIN1 | AIN2 | AIN3 | AIN4 | AIN5 | AIN6 | AIN7 | 温度传感器 |
| | 0001 | +(AIN0) -(AIN1) | | AIN2 | AIN3 | AIN4 | AIN5 | AIN6 | AIN7 | 温度传感器 |
| | 0010 | AIN0 | AIN1 | +(AIN2) -(AIN3) | | AIN4 | AIN5 | AIN6 | AIN7 | 温度传感器 |
| | 0011 | +(AIN0) -(AIN1) | | +(AIN2) -(AIN3) | | AIN4 | AIN5 | AIN6 | AIN7 | 温度传感器 |
| | 0100 | AIN0 | AIN1 | AIN2 | AIN3 | +(AIN4) -(AIN5) | | AIN6 | AIN7 | 温度传感器 |
| | 0101 | +(AIN0) -(AIN1) | | AIN2 | AIN3 | +(AIN4) -(AIN5) | | AIN6 | AIN7 | 温度传感器 |
| | 0110 | AIN0 | AIN1 | +(AIN2) -(AIN3) | | +(AIN4) -(AIN5) | | AIN6 | AIN7 | 温度传感器 |
| | 0111 | +(AIN0) -(AIN1) | | +(AIN2) -(AIN3) | | +(AIN4) -(AIN5) | | AIN6 | AIN7 | 温度传感器 |
| | 1000 | AIN0 | AIN1 | AIN2 | AIN3 | AIN4 | AIN5 | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1001 | +(AIN0) -(AIN1) | | AIN2 | AIN3 | AIN4 | AIN5 | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1010 | AIN0 | AIN1 | +(AIN2) -(AIN3) | | AIN4 | AIN5 | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1011 | +(AIN0) -(AIN1) | | +(AIN2) -(AIN3) | | AIN4 | AIN5 | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1100 | AIN0 | AIN1 | AIN2 | AIN3 | +(AIN4) -(AIN5) | | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1101 | +(AIN0) -(AIN1) | | AIN2 | AIN3 | +(AIN4) -(AIN5) | | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1110 | AIN0 | AIN1 | +(AIN2) -(AIN3) | | +(AIN4) -(AIN5) | | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1111 | +(AIN0) -(AIN1) | | +(AIN2) -(AIN3) | | +(AIN4) -(AIN5) | | +(AIN6) -(AIN7) | | 温度传感器 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 5.7 ADC0CF: ADC0 配置寄存器(C8051F020/1)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--------|--------|--------|--------|--------|---------|---------|---------|----------------|
| AD0SC4 | AD0SC3 | AD0SC2 | AD0SC1 | AD0SC0 | AMP0GN2 | AMP0GN1 | AMP0GN0 | 11111000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xBC |

位 7-3: AD0SC4-0: ADC0 SAR 转换时钟周期控制位
SAR 转换时钟来源于系统时钟, 由下面的方程给出, 其中 $AD0SC$ 表示 AD0SC4-0 中保持的数值, CLK_{SAR0} 表示所需要的 ADC0 SAR 时钟 (注: ADC0 SAR 时钟应小于或等于 2.5MHz)。

$$AD0SC = \frac{SYSCLK}{CLK_{SAR0}} - 1$$

位 2-0: AMP0GN2-0: ADC0 内部放大器增益

000: 增益 = 1
001: 增益 = 2
010: 增益 = 4
011: 增益 = 8
10x: 增益 = 16
11x: 增益 = 0.5

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 5.8 ADC0CN: ADC0 控制寄存器(C8051F020/1)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--------|--|--------|---------|--------|--------|---------|--------------|----------------|
| AD0EN | AD0TM | AD0INT | AD0BUSY | AD0CM1 | AD0CM0 | AD0WINT | AD0LJST | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0xE8 |
| 位 7: | AD0EN: ADC0 允许位 0: ADC0 禁止。ADC0 处于低功耗停机状态。 1: ADC0 允许。ADC0 处于活动状态, 并准备转换数据。 | | | | | | | |
| 位 6: | AD0TM: ADC 跟踪方式位 0: 当 ADC 被允许时, 除了转换期间之外一直处于跟踪方式。 1: 由 ADSTM1-0 定义跟踪方式。 | | | | | | | |
| 位 5: | AD0INT: ADC0 转换结束中断标志 该标志必须用软件清 '0'。 0: 从最后一次将该位清 0 后, ADC0 还没有完成一次数据转换。 1: ADC 完成了一次数据转换。 | | | | | | | |
| 位 4: | AD0BUSY: ADC0 忙标志位 读: 0: ADC0 转换结束或当前没有正在进行的数据转换。AD0INT 在 AD0BUSY 的下降沿被置 '1'。 1: ADC0 正在进行转换。 写: 0: 无作用 1: 若 ADSTM1-0=00b 则启动 ADC0 转换。 | | | | | | | |
| 位 3-2: | AD0CM1-0: ADC0 转换启动方式选择位。 如果 AD0TM = 0: 00: 向 AD0BUSY 写 1 启动 ADC0 转换。 01: 定时器 3 溢出启动 ADC0 转换。 10: CNVSTR 上升沿启动 ADC0 转换。 11: 定时器 2 溢出启动 ADC0 转换。 如果 AD0TM = 1: 00: 向 AD0BUSY 写 1 时启动跟踪, 持续 3 个 SAR 时钟, 然后进行转换。 01: 定时器 3 溢出启动跟踪, 持续 3 个 SAR 时钟, 然后进行转换。 10: 只有当 CNVSTR 输入为逻辑低电平时 ADC0 跟踪, 在 CNVSTR 的上升沿开始转换。 11: 定时器 2 溢出启动跟踪, 持续 3 个 SAR 时钟, 然后进行转换。 | | | | | | | |
| 位 1: | AD0WINT: ADC0 窗口比较中断标志。 0: 自该标志被清除后未发生过 ADC0 窗口比较匹配。 1: 发生了 ADC0 窗口比较匹配。 | | | | | | | |
| 位 0: | AD0LJST: ADC0 数据左对齐选择位。 0: ADC0H:ADC0L 寄存器数据右对齐。 1: ADC0H:ADC0L 寄存器数据左对齐。 | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 5.9 ADC0H: ADC 数据字 MSB 寄存器(C8051F020/1)

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xBF |

位 7-0: ADC0 数据字高字节。
 当 AD0LJST=0: 位 7-4 为位 3 的符号扩展位。位 3-0 是 12 位 ADC0 数据字的高 4 位。
 当 AD0LJST=1: 位 7-0 为 12 位 ADC0 数据字的高 8 位。

图 5.10 ADC0L: ADC0 数据字 LSB 寄存器(C8051F020/1)

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xBE |

位 7-0: ADC0 数据字低字节。
 当 AD0LJST=0: 位 7-0 是 12 位 ADC 数据字的低 8 位。
 当 AD0LJST=1: 位 7-4 是 12 位 ADC 数据字的低 4 位。位 3-0 读出值总是为 0。

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 5.11 ADC0 数据字示例 (C8051F020/1)

12 位 ADC 结果数据字在 ADC0 数据字寄存器中存放如下:

ADC0H[3:0]:ADC0L[7:0], 如果 AD0LJST=0。

(如果是差分输入, ADC0H[7:4]为 ADC0H.3 的符号扩展位, 否则=0000b)

ADC0H[7:0]:ADC0L[7:4], 如果 AD0LJST=1。

(ADC0L[3:0]=0000b)

例: ADC 数据字转换表, AIN0 为单端输入方式

(AMX0CF=0x00, AMX0SL=0x00)

| AIN0-AGND(伏) | ADC0H:ADC0L (AD0LJST=0) | ADC0H:ADC0L (AD0LJST=1) |
|--------------------|----------------------------|----------------------------|
| VREF * (4095/4096) | 0x0FFF | 0xFFF0 |
| VREF / 2 | 0x0800 | 0x8000 |
| VREF * (2047/4096) | 0x07FF | 0x7FF0 |
| 0 | 0x0000 | 0x0000 |

例: ADC 数据字转换表, AIN0-AIN1 为差分输入对

(AMX0CF=0x01, AMX0SL=0x00)

| AIN0-AIN1(伏) | ADC0H:ADC0L (AD0LJST=0) | ADC0H:ADC0L (AD0LJST=1) |
|--------------------|----------------------------|----------------------------|
| VREF * (2047/2048) | 0x07FF | 0x7FF0 |
| VREF / 2 | 0x0400 | 0x4000 |
| VREF x (1/2048) | 0x0001 | 0x0010 |
| 0 | 0x0000 | 0x0000 |
| -VREF x (1/2048) | 0xFFFF (-1d) | 0xFFF0 |
| -VREF / 2 | 0xFC00 (-1024d) | 0xC000 |
| -VREF | 0xF800 (-2048d) | 0x8000 |

对于 AD0LJST = 0:

转换代码 = $V_{in} \times \frac{Gain}{VREF} \times 2^n$; 单端方式时 n = 12; 差分方式时 n = 11。

5.3 ADC0 可编程窗口检测器

ADC0 可编程窗口检测器不停地将 ADC0 输出与用户编程的极限值进行比较,并在检测到越限条件时通知系统控制器。这在一个中断驱动的系统尤其有效,既可以节省代码空间和 CPU 带宽又能提供快速响应时间。窗口检测器中断标志 (ADC0CN 中的 AD0WINT 位) 也可被用于查询方式。参考字的高和低字节被装入到 ADC0 下限 (大于) 和 ADC0 上限 (小于) 寄存器 (ADC0GTH、ADC0GTL、ADC0LTH 和 ADC0LTL)。图 5.16 - 5.19 给出了比较示例供参考。注意,窗口检测器标志既可以在测量数据位于用户编程的极限值以内时有效,也可以在测量数据位于用户编程的极限值以外时有效,这取决于 ADC0GTx 和 ADC0LTx 的编程值。

图 5.12 ADC0GTH: ADC0 下限数据高字节寄存器 (C8051F020/1)

| | | | | | | | | |
|--------------------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 11111111 |
| | | | | | | | | SFR地址: 0xC5 |
| 位 7-0: ADC0 下限数据字的高字节 | | | | | | | | |

图 5.13 ADC0GTL: ADC0 下限数据低字节寄存器 (C8051F020/1)

| | | | | | | | | |
|--------------------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 11111111 |
| | | | | | | | | SFR地址: 0xC4 |
| 位 7-0: ADC0 下限数据字的低字节 | | | | | | | | |

图 5.14 ADC0LTH: ADC0 上限数据高字节寄存器(C8051F020/1)

| | | | | | | | | |
|----------------------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xC7 |
| 位 7-0: ADC0 数据上限数据字的高字节 | | | | | | | | |

图 5.15 ADC0LTL: ADC0 上限数据低字节寄存器(C8051F020/1)

| | | | | | | | | |
|--------------------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xC6 |
| 位 7-0: ADC0 上限数据字的低字节 | | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 5.16 12 位 ADC0 窗口中断示例：右对齐的单端数据

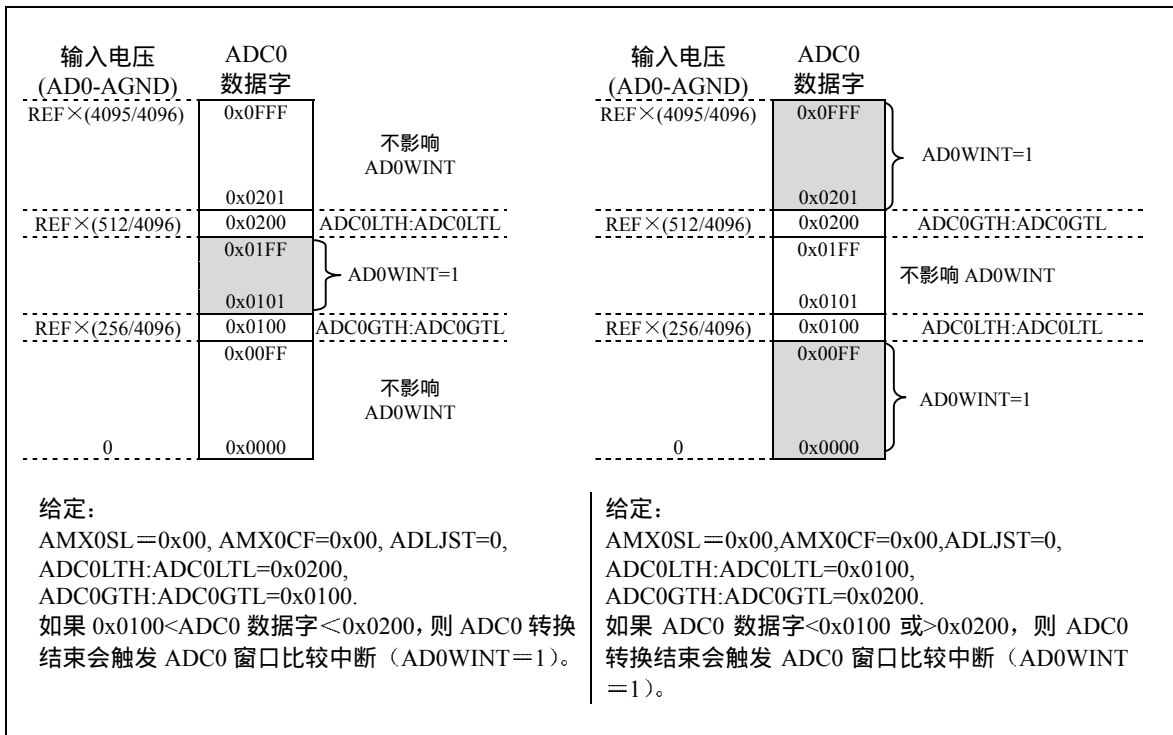
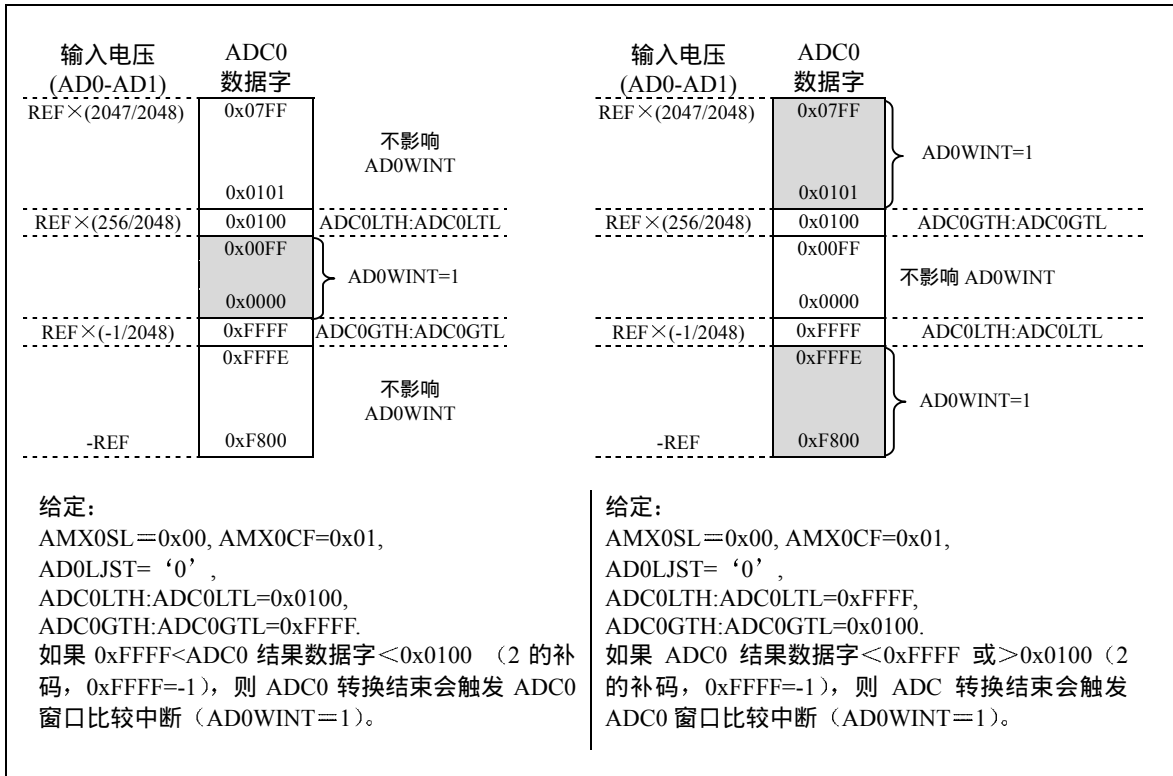
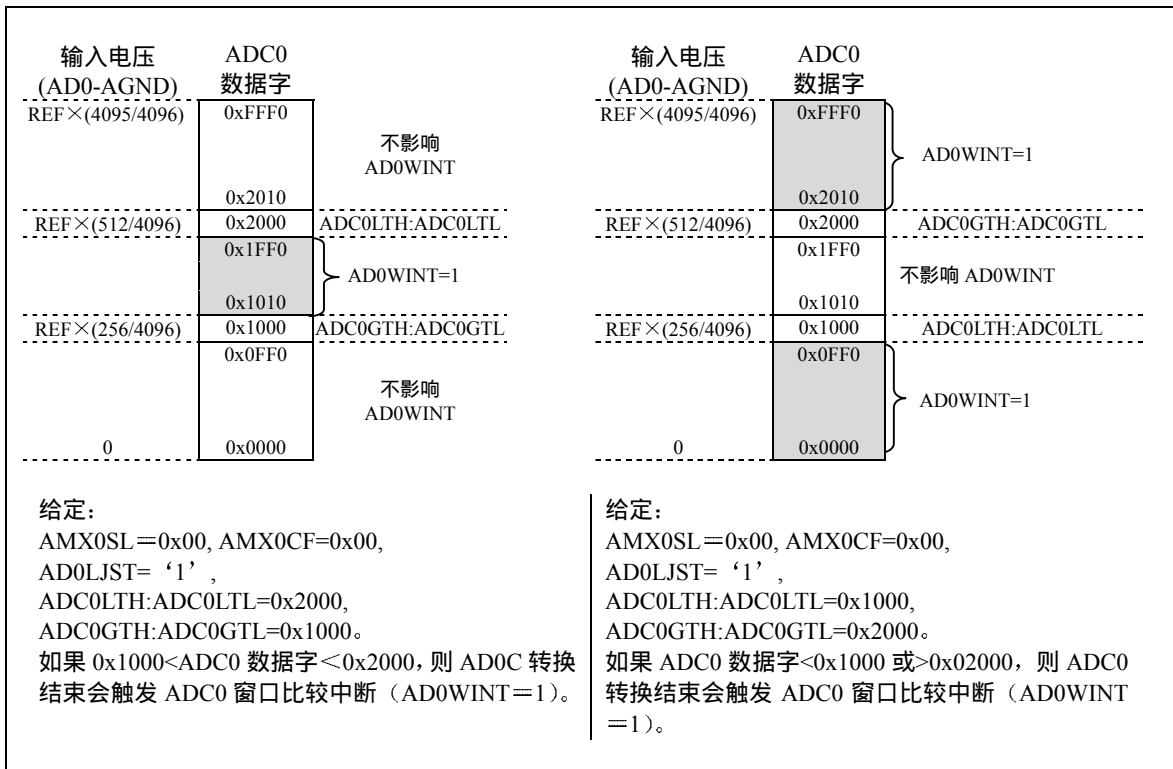


图 5.17 12 位 ADC0 窗口中断示例：右对齐的差分数据



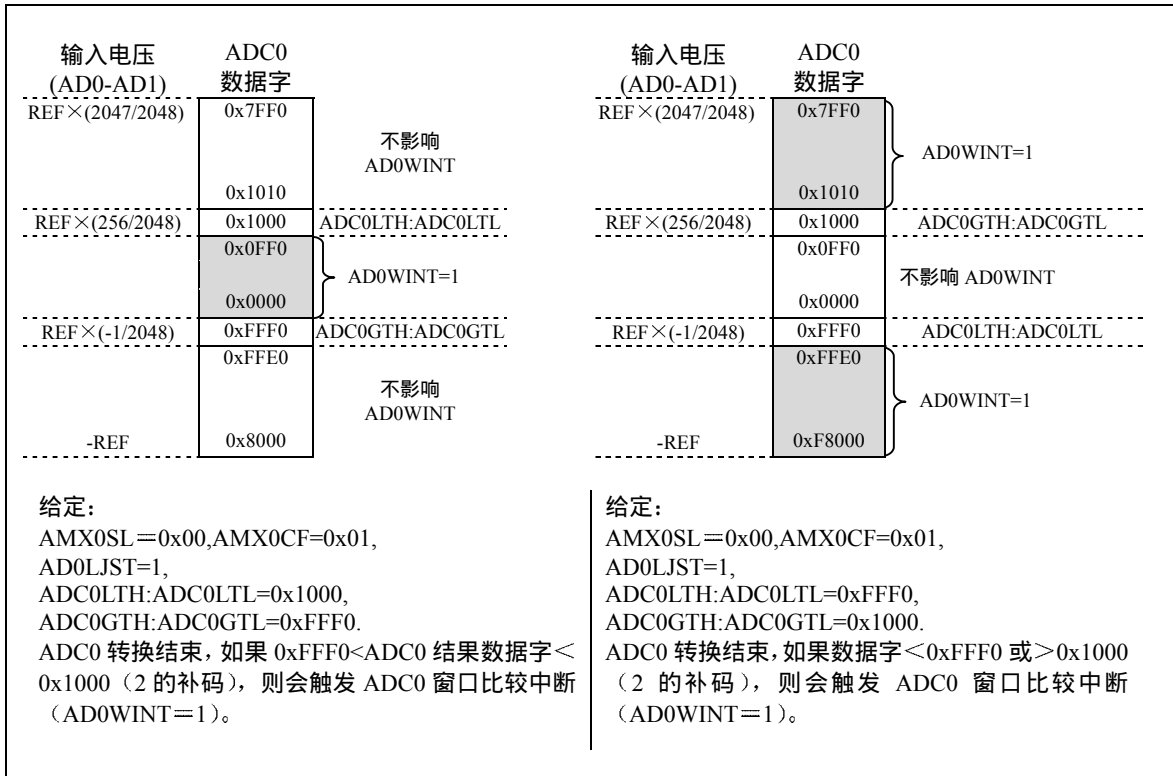
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 5.18 12 位 ADC0 窗口中断示例：左对齐的单端数据



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 5.19 12 位 ADC0 窗口中断示例：左对齐的差分数据



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 5.1 12 位 ADC0 电气特性 (C8051F020/1)

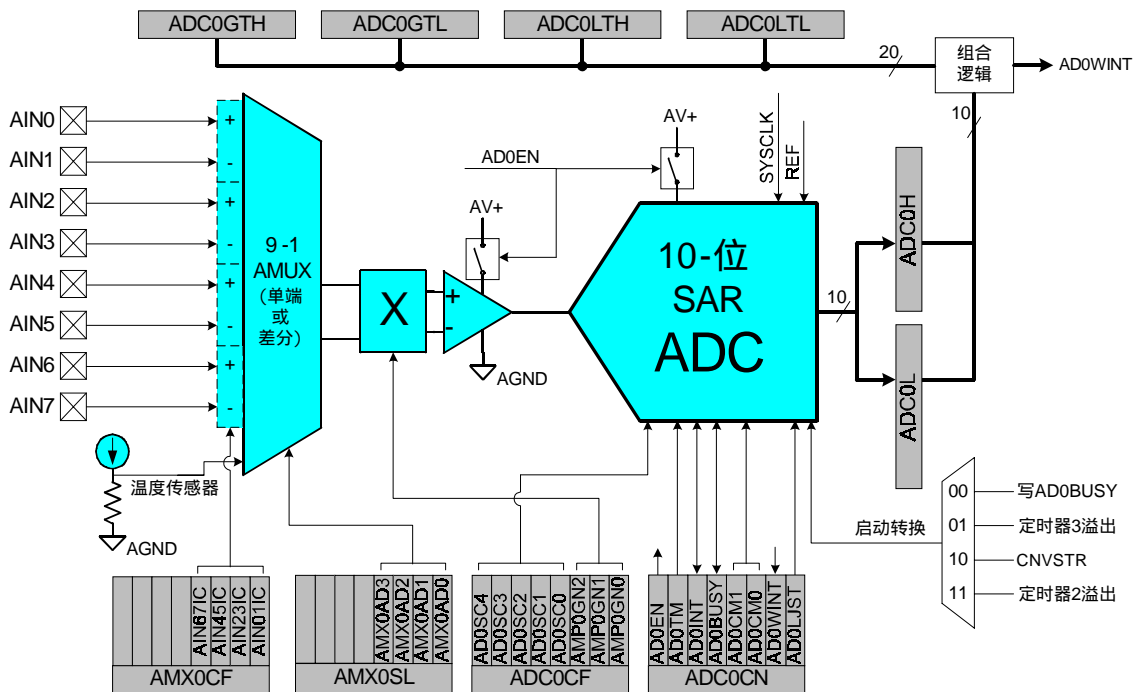
VDD=3.0V, AV+=3.0V, VREF=2.40V(REFBE=0), PGA 增益=1, -40°C 到+85°C (除非另有说明)

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|--|-------------------|------|-------|------|--------|
| 直流精度 | | | | | |
| 分辨率 | | 12 | | | 位 |
| 积分非线性 | | | | ±1 | LSB |
| 微分非线性 | 保证单调 | | | ±1 | LSB |
| 偏移误差 | | | -3±1 | | LSB |
| 满度误差 | 差分方式 | | -7±3 | | LSB |
| 偏移温度系数 | | | ±0.25 | | ppm/°C |
| 动态性能 (10kHz 正弦波输入, 满度值的 0 到-1dB, 100ksps) | | | | | |
| 信号与噪声失真比 | | 66 | | | dB |
| 总谐波失真 | 到 5 次谐波 | | -75 | | dB |
| 有效动态范围 | | | 80 | | dB |
| 转换速率 | | | | | |
| SAR 时钟频率 | | | | 2.5 | MHz |
| 转换占用 SAR 时钟数 | | 16 | | | 周期 |
| 跟踪/保持捕获时间 | | 1.5 | | | μS |
| 转换速率 | | | | 100 | ksps |
| 模拟输入 | | | | | |
| 电压转换范围 | 单端方式 | 0 | | VREF | V |
| *共模电压范围 | 差分方式 | AGND | | AV+ | V |
| 输入电容 | | | 10 | | pF |
| 温度传感器 | | | | | |
| 非线性度 | | -1.0 | | 1.0 | °C |
| 绝对精度 | | | ±3 | | °C |
| 增益 | PGA 增益=1 | | 2.86 | | mV/°C |
| 偏移 | PGA 增益=1, 温度 =0°C | | 0.776 | | V |
| 电源指标 | | | | | |
| 电源电流 (AV+ 给 ADC 供电) | 工作方式, 100ksps | | 450 | 900 | μA |
| 电源抑制比 | | | ±0.3 | | mV/V |

6. ADC0（10 位，只限于 C8051F022/3）

C8051F022/3 的 ADC0 子系统包括一个 9 通道的可编程模拟多路选择器 (AMUX0)，一个可编程增益放大器 (PGA0) 和一个 100kps、10 位分辨率的逐次逼近寄存器型 ADC，ADC 中集成了跟踪保持电路和可编程窗口检测器（见图 6.1 的原理框图）。AMUX0、PGA0、数据转换方式及窗口检测器都可用软件通过图 6.1 所示的特殊功能寄存器来控制。ADC0 所使用的电压基准按“9. 电压基准 (C8051F020/2)”或“10. 电压基准 (C8051F021/3)”所述选择。只有当 ADC0 控制寄存器中的 AD0EN 位被置‘1’时 ADC0 子系统 (ADC0、跟踪保持器和 PGA0) 才被允许工作。当 AD0EN 位为‘0’时，ADC0 子系统处于低功耗关断方式。

图 6.1 10 位 ADC0 功能框图



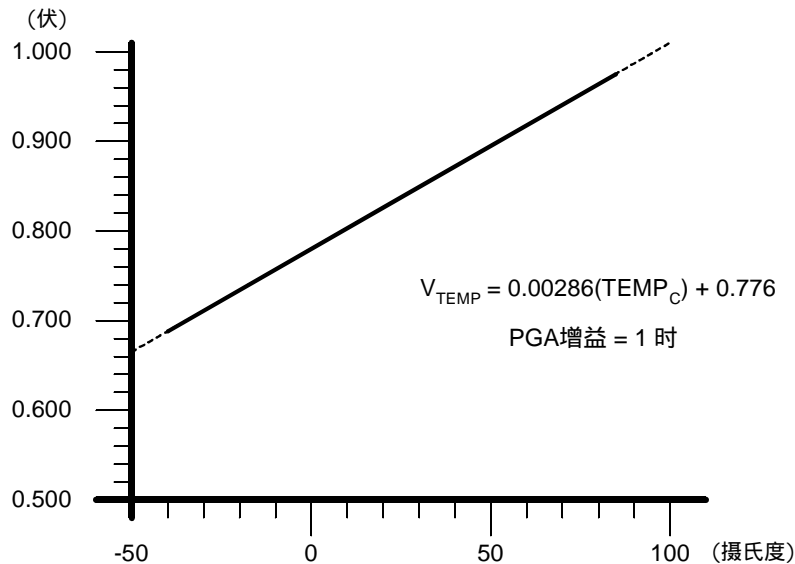
6.1 模拟多路开关和 PGA

AMUX 中的 8 个通道用于外部测量，而第九通道在内部被接到片内温度传感器（温度传输函数示于图 6.2）。注意，PGA0 的增益对温度传感器也起作用。可以将 AMUX 输入对编程为工作在差分或单端方式。这就允许用户对每个通道选择最佳的测量技术，甚至可以在测量过程中改变方式。在系统复位后 AMUX 的默认方式为单端输入。有两个与 AMUX 相关的寄存器：通道选择寄存器 AMX0SL（图 6.6）和配置寄存器 AMX0CF（图 6.7）。图 6.6 中的表给出了每种配置下各通道的功能。PGA 对 AMUX 输出信号的放大倍数由 ADC0 配置寄存器 ADC0CF（图 6.7）中的 AMP0GN2-0 确定。PGA 增益可以用软件编程为 0.5、1、2、4、8 或 16。复位后的默认增益为 1。

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

温度传感器的传输函数示于图 6.2。当温度传感器被选中(用 AMX0SL 中的 AMX0AD3-0)时,其输出电压 (V_{TEMP}) 是 PGA 的输入;PGA 对该电压的放大倍数由用户编程的 PGA 设置值决定。

图 6.2 温度传感器传输函数



6.2 ADC 的工作方式

ADC0 的最高转换速度为 100kps,其转换时钟来源于系统时钟分频,分频值保存在寄存器 ADC0CF 的 ADCSC 位。

6.2.1 启动转换

有 4 种转换启动方式,由 ADC0CN 中的 ADC0 启动转换方式位 (AD0CM1, AD0CM0) 的状态决定。转换触发源有:

1. ADC0CN 的 AD0BUSY 位写 1;
2. 时器 3 溢出 (即定时的连续转换);
3. 外部 ADC 转换启动信号的上升沿, CNVSTR;
4. 定时器 2 溢出 (即定时的连续转换)。

AD0BUSY 位在转换期间被置 '1',转换结束后复 '0'。AD0BUSY 位的下降沿触发一个中断 (当被允许时) 并将中断标志 AD0INT (ADC0CN.5) 置 '1'。转换数据被保存在 ADC 数据字的 MSB 和 LSB 寄存器: ADC0H 和 ADC0L。转换数据在寄存器对 ADC0H:ADC0L 中的存储方式可以是左对齐或右对齐,由 ADC0CN 寄存器中 AD0LJST 位的编程状态决定。

当通过向 AD0BUSY 写 '1' 启动数据转换时,应查询 AD0INT 位以确定转换何时结束 (也可以使用 ADC0 中断)。建议的查询步骤如下:

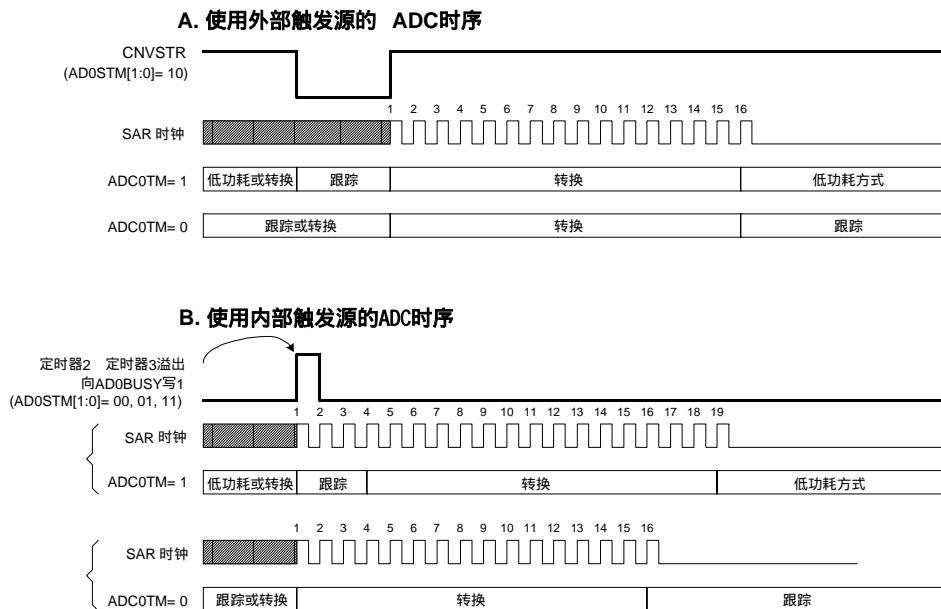
1. 写 '0' 到 AD0INT;
2. 向 ADBUSY 写 '1';

3. 查询并等待 AD0INT 变 ‘1’;
4. 处理 ADC0 数据

6.2.2 跟踪方式

ADC0CN 中的 AD0TM 位控制 ADC0 的跟踪保持方式。在缺省状态，除了转换期间之外 ADC0 输入被连续跟踪。当 AD0TM 位为逻辑 ‘1’ 时，ADC0 工作在低功耗跟踪保持方式。在该方式下，每次转换之前都有 3 个 SAR 时钟的跟踪周期（在启动转换信号有效之后）。当 CNVSTR 信号用于在低功耗跟踪保持方式启动转换时，ADC0 只在 CNVSTR 为低电平时跟踪；在 CNVSTR 的上升沿开始转换（见图 6.3）。当整个芯片处于低功耗待机或休眠方式时，跟踪可以被禁止（关断）。当 AMUX 或 PGA 的设置频繁改变时，低功耗跟踪保持方式也非常有用，可以保证建立时间需求得到满足（见“6.2.3 建立时间要求”）。

图 6.3 10 位 ADC 跟踪和转换时序举例



6.2.3 建立时间要求

当 ADC0 输入配置发生改变时 (AMUX 或 PGA 的选择发生变化), 在进行一次精确的转换之前需要有一个最小的跟踪时间。该跟踪时间由 ADC0 模拟多路器的电阻、ADC0 采样电容、外部信号源阻抗及所要求的转换精度决定。图 6.4 给出了单端和差分方式下等效的 ADC0 输入电路。注意: 这两种等效电路的时间常数完全相同。对于一个给定的建立精度 (SA), 所需要的 ADC0 建立时间可以用方程 6.1 估算。当测量温度传感器的输出时, R_{TOTAL} 等于 R_{MUX} 。注意: 在低功耗跟踪方式, 每次转换需要用三个 SAR 时钟跟踪。对于大多数应用, 三个 SAR 时钟可以满足跟踪需要。

方程 6.1 ADC0 建立时间要求

$$t = -\ln\left(\frac{SA}{2^n}\right) \times R_{TOTAL} C_{SAMPLE}$$

其中:

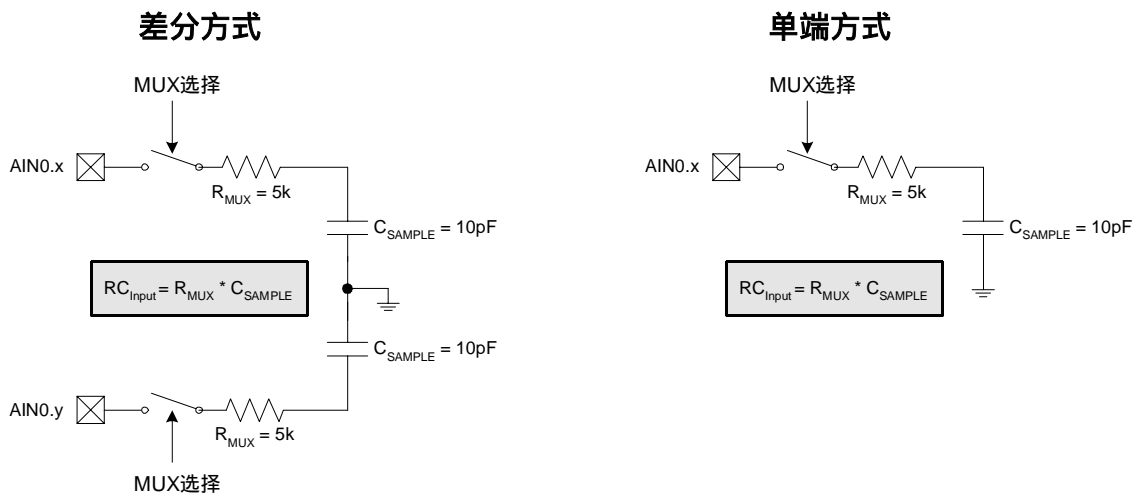
SA 是建立精度, 用一个 LSB 的分数表示 (例如, 建立精度 0.25 对应 1/4 LSB)

t 为所需要的建立时间, 以秒为单位

R_{TOTAL} 为 ADC0 模拟多路器电阻与外部信号源电阻之和

n 为 ADC0 的分辨率, 用比特表示 (10)。

图 6.4 ADC0 等效输入电路



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 6.5 AMX0CF: AMUX0 配置寄存器(C8051F022/3)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|-----|-----|-----|---------|---------|---------|---------|----------------|
| - | - | - | - | AIN67IC | AIN45IC | AIN23IC | AIN01IC | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xBA |
| <p>位 7-4: 未使用。读 = 0000b; 写 = 忽略</p> <p>位 3 AIN67IC: AIN6、AIN7 输入对配置位 0: AIN6 和 AIN7 为独立的单端输入 1: AIN6, AIN7 为 (分别为) +, -差分输入对</p> <p>位 2 AIN45IC: AIN4、AIN5 输入对配置位 0: AIN4 和 AIN5 为独立的单端输入 1: AIN4, AIN5 为 (分别为) +, -差分输入对</p> <p>位 1 AIN23IC: AIN2、AIN3 输入对配置位 0: AIN2 和 AIN2 为独立的单端输入 1: AIN2, AIN2 为 (分别为) +, -差分输入对</p> <p>位 0 AIN01IC: AIN0、AIN1 输入对配置位 0: AIN0 和 AIN1 为独立的单端输入 1: AIN0, AIN1 为 (分别为) +, -差分输入对</p> <p>注: 对于被配置成差分输入的通道, ADC0 数据字格式为 2 的补码。</p> | | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 6.6 AMUX0SL: AMUX0 通道选择寄存器(C8051F022/3)

| | | | | | | | | | | |
|--|-----|-----|-----|-----|---------|---------|---------|---------|-----|----------------|
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| | - | - | - | - | AMX0AD3 | AMX0AD2 | AMX0AD1 | AMX0AD0 | | 00000000 |
| | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | | SFR地址: 0xBB |

位 7-4: 未使用。读 = 0000b; 写 = 忽略
 位 3-0: AMX0AD3-0: AMUX 地址位
 0000-1111b: 根据下表选择 ADC 输入

| | | AMX0AD3-0 | | | | | | | | |
|--------------|------|--------------------|------|--------------------|------|--------------------|------|--------------------|------|-------|
| | | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1xxx |
| AMX0CF 位 3-0 | 0000 | AIN0 | AIN1 | AIN2 | AIN3 | AIN4 | AIN5 | AIN6 | AIN7 | 温度传感器 |
| | 0001 | +(AIN0) -(AIN1) | | AIN2 | AIN3 | AIN4 | AIN5 | AIN6 | AIN7 | 温度传感器 |
| | 0010 | AIN0 | AIN1 | +(AIN2) -(AIN3) | | AIN4 | AIN5 | AIN6 | AIN7 | 温度传感器 |
| | 0011 | +(AIN0) -(AIN1) | | +(AIN2) -(AIN3) | | AIN4 | AIN5 | AIN6 | AIN7 | 温度传感器 |
| | 0100 | AIN0 | AIN1 | AIN2 | AIN3 | +(AIN4) -(AIN5) | | AIN6 | AIN7 | 温度传感器 |
| | 0101 | +(AIN0) -(AIN1) | | AIN2 | AIN3 | +(AIN4) -(AIN5) | | AIN6 | AIN7 | 温度传感器 |
| | 0110 | AIN0 | AIN1 | +(AIN2) -(AIN3) | | +(AIN4) -(AIN5) | | AIN6 | AIN7 | 温度传感器 |
| | 0111 | +(AIN0) -(AIN1) | | +(AIN2) -(AIN3) | | +(AIN4) -(AIN5) | | AIN6 | AIN7 | 温度传感器 |
| | 1000 | AIN0 | AIN1 | AIN2 | AIN3 | AIN4 | AIN5 | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1001 | +(AIN0) -(AIN1) | | AIN2 | AIN3 | AIN4 | AIN5 | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1010 | AIN0 | AIN1 | +(AIN2) -(AIN3) | | AIN4 | AIN5 | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1011 | +(AIN0) -(AIN1) | | +(AIN2) -(AIN3) | | AIN4 | AIN5 | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1100 | AIN0 | AIN1 | AIN2 | AIN3 | +(AIN4) -(AIN5) | | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1101 | +(AIN0) -(AIN1) | | AIN2 | AIN3 | +(AIN4) -(AIN5) | | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1110 | AIN0 | AIN1 | +(AIN2) -(AIN3) | | +(AIN4) -(AIN5) | | +(AIN6) -(AIN7) | | 温度传感器 |
| | 1111 | +(AIN0) -(AIN1) | | +(AIN2) -(AIN3) | | +(AIN4) -(AIN5) | | +(AIN6) -(AIN7) | | 温度传感器 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 6.7 ADC0CF: ADC0 配置寄存器(C8051F022/3)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|--------|--------|--------|--------|---------|---------|---------|----------------|
| AD0SC4 | AD0SC3 | AD0SC2 | AD0SC1 | AD0SC0 | AMP0GN2 | AMP0GN1 | AMP0GN0 | 11111000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xBC |
| <p>位 7-3: AD0SC4-0: ADC0 SAR 转换时钟周期控制位 SAR 转换时钟来源于系统时钟, 由下面的方程给出, 其中 $AD0SC$ 表示 AD0SC4-0 中保持的数值, CLK_{SAR0} 表示所需要的 ADC0 SAR 时钟 (注: ADC0 SAR 时钟应小于或等于 2.5MHz)。</p> $AD0SC = \frac{SYSCLK}{CLK_{SAR0}} - 1$ <p>位 2-0: AMP0GN2-0: ADC0 内部放大器增益</p> <p>000: 增益 = 1 001: 增益 = 2 010: 增益 = 4 011: 增益 = 8 10x: 增益 = 16 11x: 增益 = 0.5</p> | | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 6.8 ADC0CN: ADC0 控制寄存器(C8051F022/3)

| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--------|--|-------|--------|---------|--------|--------|---------|----------------------------|
| | AD0EN | AD0TM | AD0INT | AD0BUSY | AD0CM1 | AD0CM0 | AD0WINT | AD0LJST |
| | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) |
| | | | | | | | | 00000000 SFR地址: 0xE8 |
| 位 7: | AD0EN: ADC0 允许位 0: ADC0 禁止。ADC0 处于低耗停机状态。 1: ADC0 允许。ADC0 处于活动状态, 并准备转换数据。 | | | | | | | |
| 位 6: | AD0TM: ADC0 跟踪方式位 0: 当 ADC0 被允许时, 除了转换期间之外一直处于跟踪方式。 1: 由 ADSTM1-0 定义跟踪方式。 | | | | | | | |
| 位 5: | AD0INT: ADC0 转换结束中断标志 该标志必须用软件清 '0'。 0: 从最后一次将该位清 0 后, ADC0 还没有完成一次数据转换。 1: ADC0 完成了一次数据转换。 | | | | | | | |
| 位 4: | AD0BUSY: ADC0 忙标志位 读: 0: ADC0 转换结束或当前没有正在进行的数据转换。AD0INT 在 AD0BUSY 的下降沿被置 '1'。 1: ADC0 正在进行转换。 写: 0: 无作用 1: 若 ADSTM1-0=00b 则启动 ADC0 转换。 | | | | | | | |
| 位 3-2: | AD0CM1-0: ADC0 转换启动方式选择位。 如果 AD0TM = 0: 00: 向 AD0BUSY 写 1 启动 ADC0 转换。 01: 定时器 3 溢出启动 ADC0 转换。 10: CNVSTR 上升沿启动 ADC0 转换。 11: 定时器 2 溢出启动 ADC0 转换。 如果 AD0TM = 1: 00: 向 AD0BUSY 写 1 时启动跟踪, 持续 3 个 SAR 时钟, 然后进行转换。 01: 定时器 3 溢出启动跟踪, 持续 3 个 SAR 时钟, 然后进行转换。 10: 只有当 CNVSTR 输入为逻辑低电平时 ADC0 跟踪, 在 CNVSTR 的上升沿开始转换。 11: 定时器 2 溢出启动跟踪, 持续 3 个 SAR 时钟, 然后进行转换。 | | | | | | | |
| 位 1: | AD0WINT: ADC0 窗口比较中断标志。 0: 自该标志被清除后未发生过 ADC0 窗口比较匹配。 1: 发生了 ADC0 窗口比较匹配。 | | | | | | | |
| 位 0: | AD0LJST: ADC0 数据左对齐选择位。 0: ADC0H:ADC0L 寄存器数据右对齐。 1: ADC0H:ADC0L 寄存器数据左对齐。 | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 6.9 ADC0H: ADC 数据字 MSB 寄存器(C8051F022/3)

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xBF |

位 7-0: ADC0 数据字高字节。
 当 AD0LJST=0: 位 7-2 为位 1 的符号扩展位。位 1-0 是 10 位 ADC0 数据字的高 2 位。
 当 AD0LJST=1: 位 7-0 为 10 位 ADC0 数据字的高 8 位。

图 6.10 ADC0L: ADC0 数据字 LSB 寄存器(C8051F022/3)

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xBE |

位 7-0: ADC0 数据字低字节。
 当 AD0LJST=0: 位 7-0 是 10 位 ADC 数据字的低 8 位。
 当 AD0LJST=1: 位 7-6 是 10 位 ADC 数据字的低 2 位。位 5-0 读出值总是为 00000b。

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 6.11 ADC0 数据字示例 (C8051F022/3)

10 位 ADC 结果数据字在 ADC0 数据字寄存器中存放如下:

ADC0H[1:0]:ADC0L[7:0], 如果 AD0LJST=0。

(如果是差分输入, ADC0H[7:2]为 ADC0H.1 的符号扩展位, 否则=000000b)

ADC0H[7:0]:ADC0L[7:6], 如果 AD0LJST=1。

(ADC0L[5:0]=000000b)

例: ADC 数据字转换表, AIN0 为单端输入方式

(AMX0CF=0x00, AMX0SL=0x00)

| AIN0-AGND(伏) | ADC0H:ADC0L (AD0LJST=0) | ADC0H:ADC0L (AD0LJST=1) |
|--------------------|----------------------------|----------------------------|
| VREF * (1023/1024) | 0x03FF | 0xFFC0 |
| VREF / 2 | 0x0200 | 0x8000 |
| VREF * (511/1024) | 0x01FF | 0x7FC0 |
| 0 | 0x0000 | 0x0000 |

例: ADC 数据字转换表, AIN0-AIN1 为差分输入对

(AMX0CF=0x01, AMX0SL=0x00)

| AIN0-AIN1(伏) | ADC0H:ADC0L (AD0LJST=0) | ADC0H:ADC0L (AD0LJST=1) |
|------------------|----------------------------|----------------------------|
| VREF * (511/512) | 0x01FF | 0x7FC0 |
| VREF / 2 | 0x0100 | 0x4000 |
| VREF x (1/512) | 0x0001 | 0x0040 |
| 0 | 0x0000 | 0x0000 |
| -VREF x (1/512) | 0xFFFF (-1d) | 0xFFC0 |
| -VREF / 2 | 0xFF00 (-256d) | 0xC000 |
| -VREF | 0xFE00 (-512d) | 0x8000 |

对于 AD0LJST = 0:

转换代码 = $V_{in} \times \frac{Gain}{VREF} \times 2^n$; 单端方式时 n = 10; 差分方式时 n = 9。

6.3 ADC0 可编程窗口检测器

ADC0 可编程窗口检测器不停地将 ADC0 输出与用户编程的极限值进行比较,并在检测到越限条件时通知系统控制器。这在一个中断驱动的系统尤其有效,既可以节省代码空间和 CPU 带宽又能提供快速响应时间。窗口检测器中断标志 (ADC0CN 中的 AD0WINT 位) 也可被用于查询方式。参考字的高和低字节被装入到 ADC0 下限 (大于) 和 ADC0 上限 (小于) 寄存器 (ADC0GTH、ADC0GTL、ADC0LTH 和 ADC0LTL)。图 6.16 - 6.19 给出了比较示例供参考。注意,窗口检测器标志既可以在测量数据位于用户编程的极限值以内时有效,也可以在测量数据位于用户编程的极限值以外时有效,这取决于 ADC0GTx 和 ADC0LTx 的编程值。

图 6.12 ADC0GTH: ADC0 下限数据高字节寄存器 (C8051F022/3)

| | | | | | | | | |
|--------------------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 11111111 |
| | | | | | | | | SFR地址: 0xC5 |
| 位 7-0: ADC0 下限数据字的高字节 | | | | | | | | |

图 6.13 ADC0GTL: ADC0 下限数据低字节寄存器 (C8051F022/3)

| | | | | | | | | |
|--------------------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 11111111 |
| | | | | | | | | SFR地址: 0xC4 |
| 位 7-0: ADC0 下限数据字的低字节 | | | | | | | | |

图 6.14 ADC0LTH: ADC0 上限数据高字节寄存器(C8051F022/3)

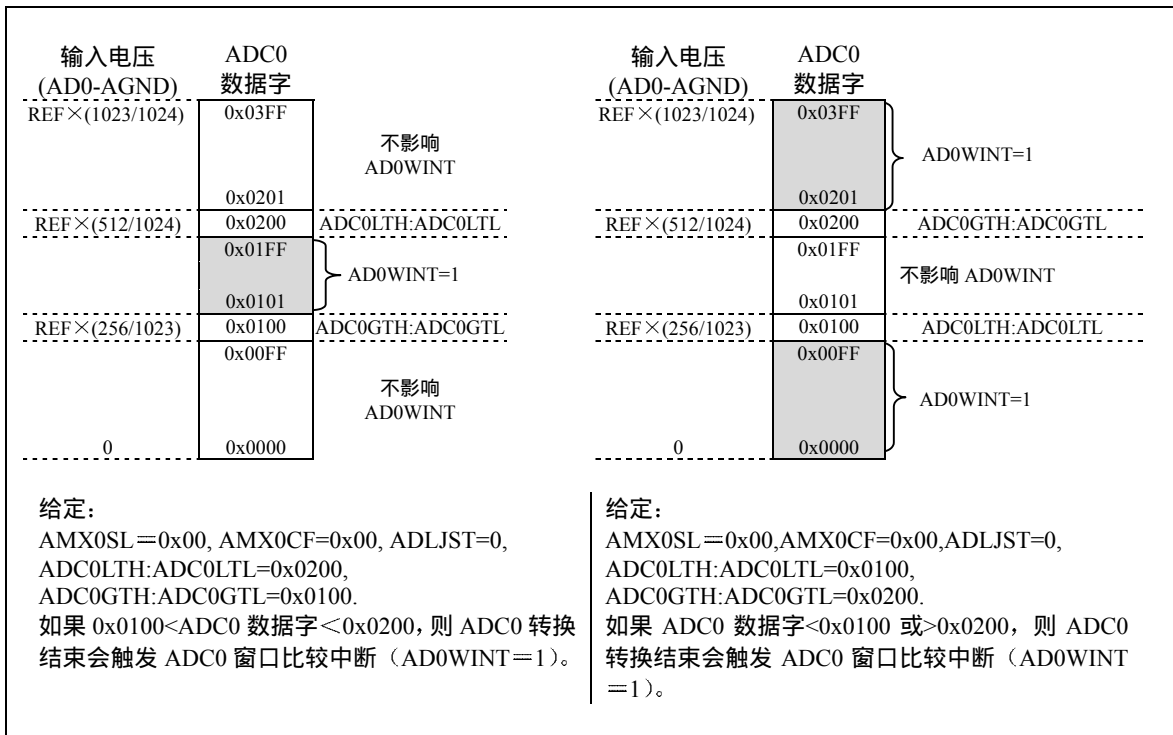
| | | | | | | | | |
|----------------------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xC7 |
| 位 7-0: ADC0 数据上限数据字的高字节 | | | | | | | | |

图 6.15 ADC0LTL: ADC0 上限数据低字节寄存器(C8051F022/3)

| | | | | | | | | |
|--------------------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xC6 |
| 位 7-0: ADC0 上限数据字的低字节 | | | | | | | | |

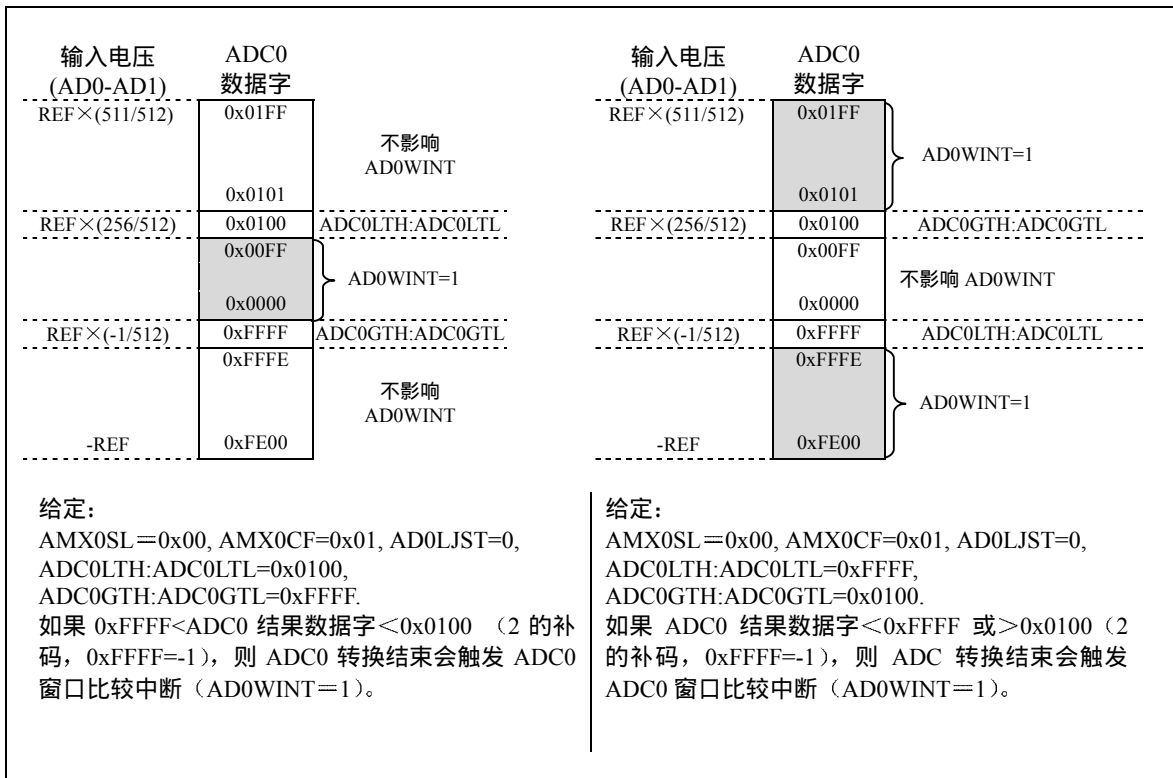
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 6.16 10 位 ADC0 窗口中断示例：右对齐的单端数据



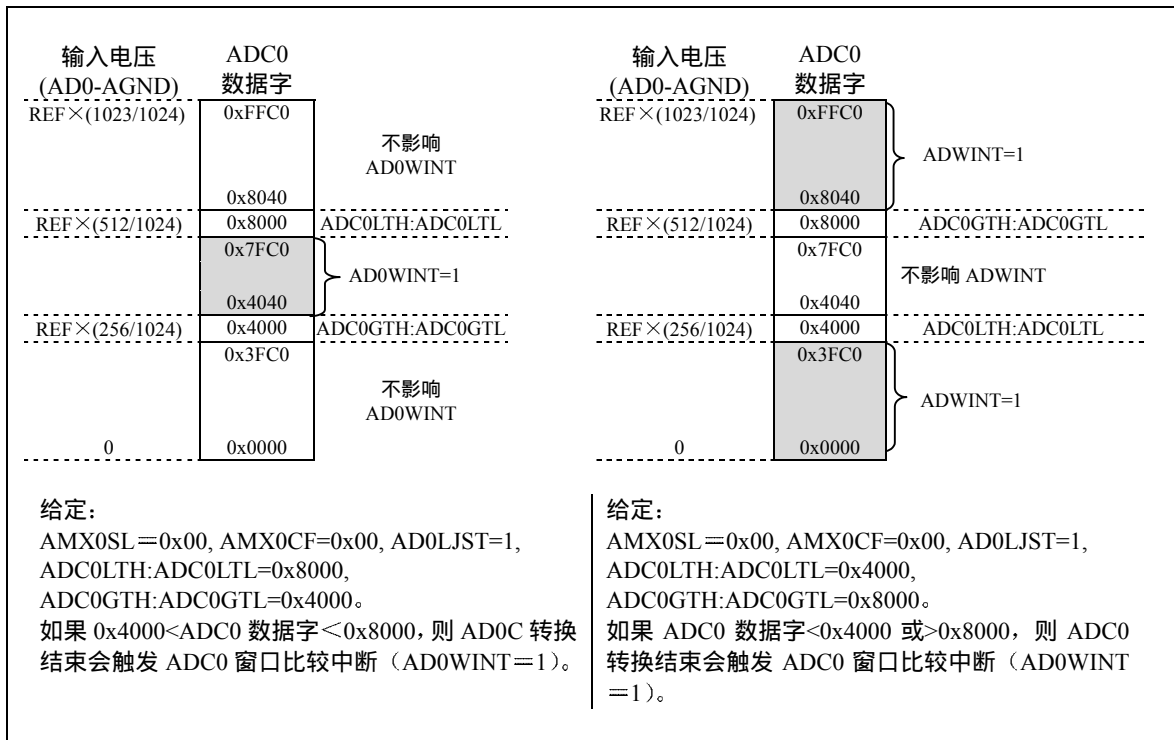
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 6.17 10 位 ADC0 窗口中断示例：右对齐的差分数据



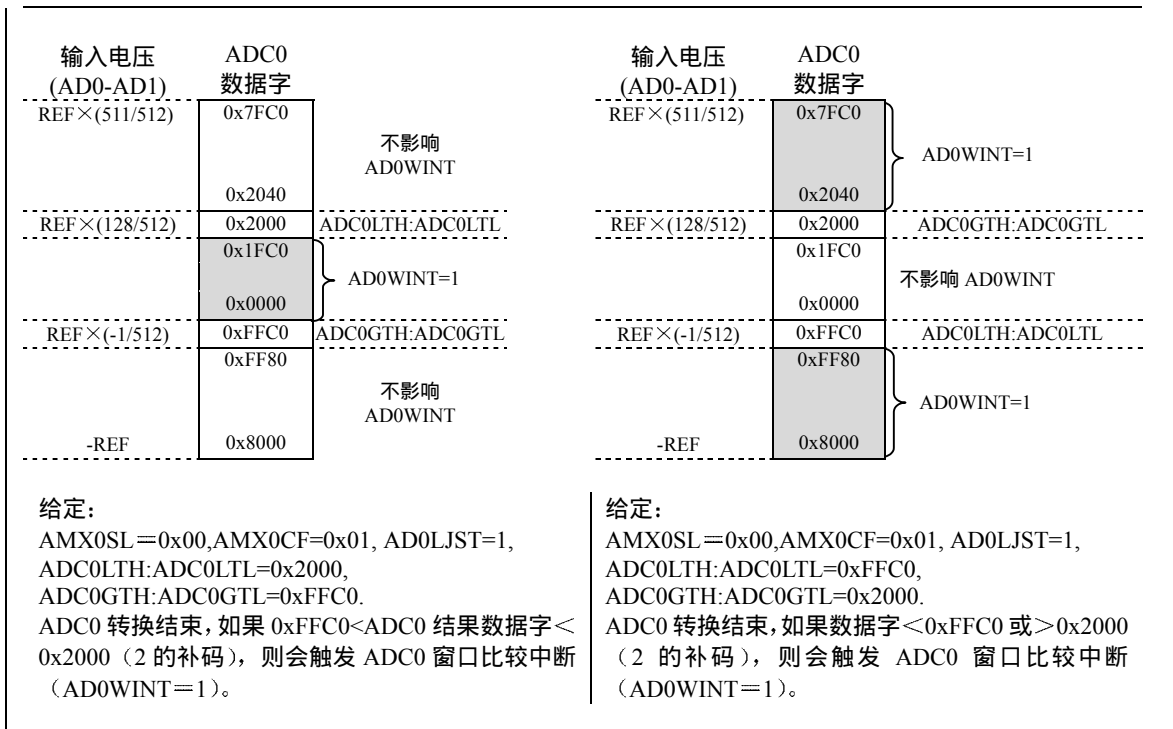
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 6.18 10 位 ADC0 窗口中断示例：左对齐的单端数据



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 6.19 10 位 ADC0 窗口中断示例：左对齐的差分数据



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 6.1 10 位 ADC0 电气特性 (C8051F022/3)

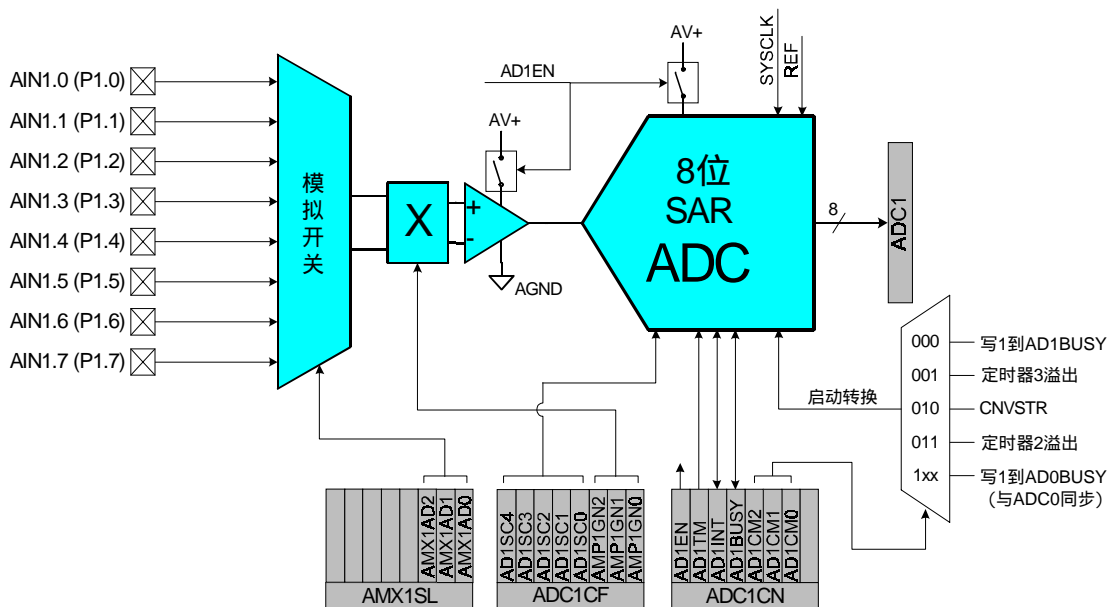
VDD=3.0V, AV+=3.0V, VREF=2.40V(REFBE=0), PGA 增益=1, -40°C 到+85°C (除非另有说明)

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|--|-------------------|------|----------|------|--------|
| 直流精度 | | | | | |
| 分辨率 | | 10 | | | 位 |
| 积分非线性 | | | | ±1 | LSB |
| 微分非线性 | 保证单调 | | | ±1 | LSB |
| 偏移误差 | | | ±0.5 | | LSB |
| 满度误差 | 差分方式 | | -1.5±0.5 | | LSB |
| 偏移温度系数 | | | ±0.25 | | ppm/°C |
| 动态性能 (10kHz 正弦波输入, 满度值的 0 到-1dB, 100ksps) | | | | | |
| 信号与噪声失真比 | | 59 | | | dB |
| 总谐波失真 | 到 5 次谐波 | | -70 | | dB |
| 有效动态范围 | | | 80 | | dB |
| 转换速率 | | | | | |
| SAR 时钟频率 | | | | 2.5 | MHz |
| 转换占用 SAR 时钟数 | | 16 | | | 周期 |
| 跟踪/保持捕获时间 | | 1.5 | | | μS |
| 转换速率 | | | | 100 | ksps |
| 模拟输入 | | | | | |
| 电压转换范围 | 单端方式 | 0 | | VREF | V |
| *共模电压范围 | 差分方式 | AGND | | AV+ | V |
| 输入电容 | | | 10 | | pF |
| 温度传感器 | | | | | |
| 非线性度 | | -1.0 | | 1.0 | °C |
| 绝对精度 | | | ±3 | | °C |
| 增益 | PGA 增益=1 | | 2.86 | | mV/°C |
| 偏移 | PGA 增益=1, 温度 =0°C | | 0.776 | | V |
| 电源指标 | | | | | |
| 电源电流 (AV+ 给 ADC 供电) | 工作方式, 100ksps | | 450 | 900 | μA |
| 电源抑制比 | | | ±0.3 | | mV/V |

7. ADC1 (8 位 ADC)

C8051F020/1/2/3 的 ADC1 子系统包括一个 8 通道的可配置模拟多路开关 (AMUX1), 一个可编程增益放大器 (PGA1) 和一个 500kpsps、8 位分辨率的逐次逼近寄存器型 ADC, 该 ADC 中集成了跟踪保持电路 (见图 7.1 的原理框图)。AMUX1、PGA1 及数据转换方式都可用软件通过图 7.1 所示的特殊功能寄存器来配置。只有当 ADC1 控制寄存器 (ADC1CN) 中的 AD1EN 位被置 ‘1’ 时 ADC1 子系统 (8 位 ADC、跟踪保持器和 PGA) 才被使能。当 AD1EN 位为 ‘0’ 时, ADC1 子系统处于低功耗关断方式。关于 ADC1 电压基准的选择见 “9. 电压基准 (C8051F020/2)” 或 “10. 电压基准 (C8051F021/3)”。

图 7.1 ADC1 功能框图



7.1 模拟多路开关和 PGA

ADC1 有 8 个通道用于测量, 用寄存器 AMX1SL (图 7.5) 选择通道。PGA 对 AMUX 输出信号的放大倍数由 ADC1 配置寄存器 ADC0CF (图 7.4) 中的 AMPGN2-0 确定。PGA 增益可以用软件编程为 0.5、1、2、4。复位时的默认增益为 0.5。

注意: AIN1 引脚也作为端口 1 的 I/O 引脚, 当用作 ADC1 输入时必须被配置为模拟输入。为了将 AIN1 的某个引脚配置为模拟输入, 要将寄存器 P1MDIN 中的对应位置 ‘1’。被选作模拟输入的端口 1 引脚不进入数字 I/O 交叉开关。有关配置 AIN1 引脚的详细信息见 “17.1.6. 配置端口 1 引脚为模拟输入 (AIN1.[7:0])”。

7.2 ADC1 的工作方式

ADC1 的最高转换速度为 500ksp/s。ADC1 的转换时钟 (SAR1 时钟) 来源于系统时钟分频。由 ADC1CF 寄存器的 AD1SC 位决定 (系统时钟 / (AD1SC+1), $0 \leq \text{AD1SC} \leq 31$)。ADC1 转换时钟频率最大为 6 MHz。

7.2.1 启动转换

有 5 种 A/D 转换启动方式, 由 ADC1CN 中的 ADC1 启动转换方式位 (AD1CM2-0) 的编程状态决定。转换启动源有:

1. 向 ADC1CN 的 AD1BUSY 位写 '1';
2. 定时器 3 溢出 (即定时的连续转换);
3. 外部 ADC 转换启动信号 CNVSTR 的上升沿;
4. 定时器 2 溢出 (即定时的连续转换)。
5. 向 ADC0CN 的 AD0BUSY 位写 1 (用一个软件命令启动 ADC1 和 ADC0)。

AD1BUSY 位在转换期间被置 '1', 转换结束后复 '0'。AD1BUSY 位的下降沿触发一个中断 (当被允许时) 并将 ADC1CN 中的中断标志置 '1'。转换结果保存在 ADC1 的数据字 ADC1 中。

当采用向 AD1BUSY 位写 '1' 这一启动方式时, 建议通过查询 AD1INT 来确定转换何时完成。建议的查询步骤如下:

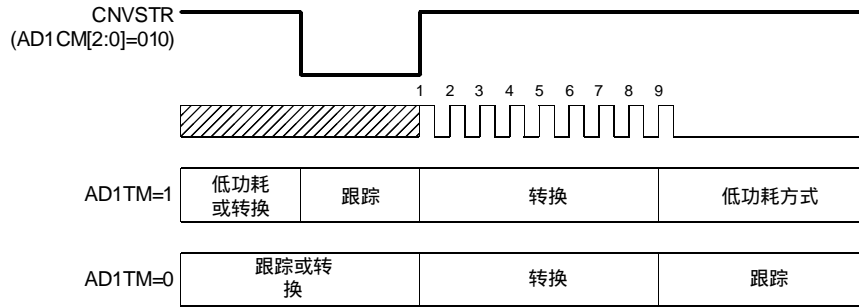
1. 向 AD1INT 写 '0';
2. 向 AD1BUSY 写 '1';
3. 查询并等待 AD1INT 变为 '1';
4. 处理 ADC1 数据。

7.2.2 跟踪方式

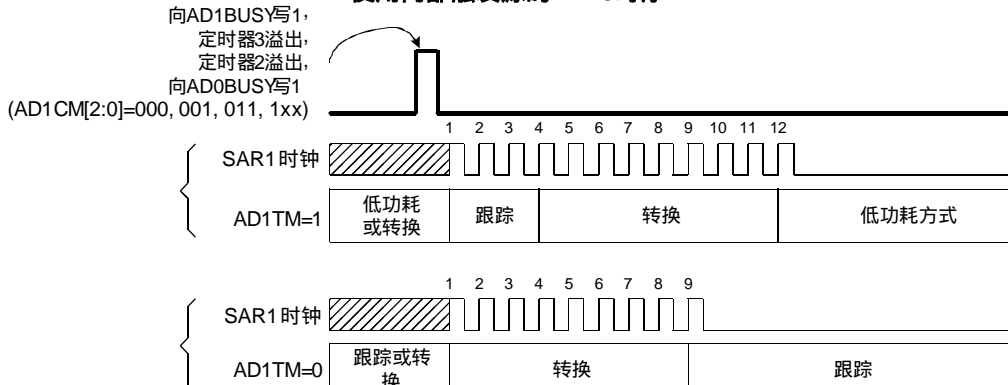
ADC1CN 中的 AD1TM 位控制 ADC1 的跟踪保持方式。在缺省状态, ADC1 输入被连续跟踪 (转换期间除外)。当 AD1TM 位被设置为逻辑 '1' 时, ADC1 工作在低功耗跟踪方式。在该方式下, 每次转换之前都要有三个 SAR 时钟的跟踪周期 (在启动转换信号之后)。当在低功耗跟踪方式下用 CNVSTR 信号作为转换启动源时, 只在 CNVSTR 为低电平时跟踪, 从 CNVSTR 的上升沿开始转换 (见图 7.2)。当整个芯片处于低功耗停机或休眠方式时, 跟踪被禁止。由于需要有建立时间, 所以低功耗跟踪保持方式在需要频繁改变 AMUX 和 PGA 的场合也是非常有用的。关于建立时间要求, 详见“7.2.3. 建立时间要求”。

图 7.2 ADC1 跟踪和转换时序举例

A. 使用外部触发源的 ADC时序



B. 使用内部触发源的 ADC时序



7.2.3 建立时间要求

当 ADC1 输入配置发生改变时 (AMUX 或 PGA 的选择发生变化), 在进行一次精确的转换之前需要有一个最小的跟踪时间。该跟踪时间由 ADC1 模拟多路开关的电阻、ADC1 采样电容、外部信号源阻抗及所要求的转换精度决定。图 7.3 给出了等效的 ADC1 输入电路。对于一个给定的建立精度 (SA), 所需要的 ADC1 建立时间可以用方程 7.1 估算。注意: 在 MUX 选择发生改变后, 最少需要 0.8 μ s 的建立时间; 在低功耗跟踪方式, 每次转换需要用三个 SAR 时钟跟踪。对于大多数应用, 三个 SAR 时钟可以满足跟踪需要。

方程 7.1 ADC1 建立时间要求

$$t = -\ln\left(\frac{SA}{2^n}\right) \times R_{TOTAL} C_{SAMPLE}$$

其中:

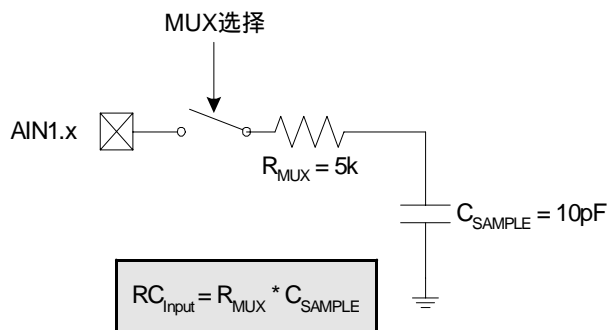
SA 是建立精度, 用一个 LSB 的分数表示 (例如, 建立精度 0.25 对应 1/4 LSB)

t 为所需要的建立时间, 以秒为单位

R_{TOTAL} 为 ADC1 模拟开关电阻与外部信号源电阻之和

n 为 ADC 的分辨率, 用比特表示 (8)。

图 7.3 ADC1 等效输入电路



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 7.4 ADC1CF: ADC1 配置寄存器(C8051F020/1/2/3)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|--------|--------|--------|--------|-----|---------|---------|----------------|
| AD1SC4 | AD1SC3 | AD1SC2 | AD1SC1 | AD1SC0 | - | AMP1GN1 | AMP1GN0 | 11111000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xAB |
| <p>位 7-3: AD1SC4-0: ADC1 SAR 转换时钟周期控制位 SAR 转换时钟频率由下面的方程计算:</p> $AD1SC = \frac{SYSCLK}{CLK_{SAR1}} - 1$ <p>其中: SYSCLK 为系统时钟, AD1SC 为 AD1SC4-0 中的 5 位数值 (注意: ADC1 SAR 转换时钟应小于或等于 6 MHz) .</p> <p>位 2: 未使用。读 = 0b; 写 = 忽略</p> <p>位 1-0: AMP1GN1-0: ADC1 内部放大器增益 (PGA)</p> <p>00: 增益 = 0.5 01: 增益 = 1 10: 增益 = 2 11: 增益 = 4</p> | | | | | | | | |

图 7.5 AMX1SL: AMUX 配置寄存器(C8051F020/1/2/3)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--|-----|-----|-----|-----|---------|---------|---------|----------------|
| - | - | - | - | - | AMX1AD2 | AMX1AD1 | AMX1AD0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xAC |
| <p>位 7-3: 未使用。读 = 00000b; 写 = 忽略</p> <p>位 2-0: AMX1 地址位</p> <p>000-111b: ADC1 输入选择如下:</p> <p>000: 选择 AIN1.0 001: 选择 AIN1.1 010: 选择 AIN1.2 011: 选择 AIN1.3 100: 选择 AIN1.4 101: 选择 AIN1.5 110: 选择 AIN1.6 111: 选择 AIN1.7</p> | | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 7.6 ADC1CN: ADC1 控制寄存器 (C8051F020/1/2/3)

| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
|--------|--|-------|--------|---------|--------|--------|--------|-----------------------------------|
| | AD1EN | AD1TM | AD1INT | AD1BUSY | AD1CM2 | AD1CM1 | AD1CM0 | - |
| | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 |
| | | | | | | | | 复位值 00000000 SFR地址: 0xAA |
| 位 7: | AD1EN: ADC1 允许位 0: ADC1 禁止。ADC1 处于低功耗停机状态。 1: ADC1 允许。ADC1 处于活动状态, 并准备转换数据。 | | | | | | | |
| 位 6: | AD1TM: ADC1 跟踪方式位 0: 一般跟踪方式。当 ADC1 被允许时, 除了转换期间之外一直处于跟踪方式。 1: 低功耗跟踪方式。由 AD1CM2-0 定义跟踪方式 (见下面的说明)。 | | | | | | | |
| 位 5: | AD1INT: ADC1 转换结束中断标志 该标志必须用软件清 '0'。 0: 从最后一次将该位清 0 后, ADC1 还没有完成一次数据转换。 1: ADC1 完成一次数据转换。 | | | | | | | |
| 位 4: | ADCBUSY: ADC1 忙标志位 读 0: ADC1 转换结束或不在进行数据转换。AD1INT 在 AD1BUSY 的下降沿被置 '1'。 1: ADC1 正在进行转换。 写 0: 无效 1: 若 AD1CM2-0=000b 则启动 ADC1 转换。 | | | | | | | |
| 位 3-1: | AD1CM2-0: ADC1 转换启动方式选择 AD1TM = 0: 000: 向 AD1BUSY 写 1 启动 ADC1 转换。 001: 定时器 3 溢出启动 ADC1 转换。 010: CNVSTR 上升沿启动 ADC1 转换。 011: 定时器 2 溢出启动 ADC1 转换。 1xx: 向 AD0BUSY 写 1 启动 ADC1 转换 (与 ADC0 软件命令转换同步)。 AD1TM = 1: 000: 向 ADBUSY 写 1 时启动跟踪并持续 3 个 SAR1 时钟, 然后进行转换。 001: 定时器 3 溢出启动跟踪并持续 3 个 SAR1 时钟, 然后进行转换。 010: 只有当 CNVSTR 输入为逻辑低电平时才启动 ADC1 跟踪, 在 CNVSTR 上升沿开始转换。 011: 定时器 2 溢出启动跟踪并持续 3 个 SAR1 时钟, 然后进行转换。 1xx: 向 AD0BUSY 写 1 启动跟踪并持续 3 个 SAR1 时钟, 然后进行转换。 | | | | | | | |
| 位 0: | 未使用。读 = 0b; 写 = 忽略 | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 7.7 ADC1: ADC1 数据字寄存器 (C8051F020/1/2/3)

| | | | | | | | | |
|-----------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0x9C |
| 位 7-0: ADC1 数据字 | | | | | | | | |

图 7.8 ADC1 数据字示例

8位ADC数据字在ADC1数据字寄存器中的意义如下:

例: ADC1数据字转换, AIN1.0 输入 (AMX1SL = 0x00)

| AIN1.0-AGND (伏) | ADC1 |
|--------------------|------|
| VREF*(255/256) | 0xFF |
| VREF/2 | 0x80 |
| VREF*(127/256) | 0x7F |
| 0 | 0x00 |

$$\text{转换结果} = V_{in} \times \frac{\text{Gain}}{V_{REF}} \times 256$$

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 7.1. ADC1 电气特性

VDD=3.0V, AV+=3.0V, VREF1=2.40V(REFBE=0), PGA1=1, -40°C 到+85°C (除非另有说明)

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|--|---------------|-----|---------|------|--------|
| 直流精度 | | | | | |
| 分辨率 | | 8 | | | 位 |
| 积分非线性 | | | | ±1 | LSB |
| 微分非线性 | 保证单调 | | | ±1 | LSB |
| 偏移误差 | | | 0.5±0.3 | | LSB |
| 满度误差 | 差分方式 | | -1±0.2 | | LSB |
| 偏移温度系数 | | | TBD | | ppm/°C |
| 动态性能 (10kHz 正弦波输入, 满度值的 0 到-1dB, 500ksps) | | | | | |
| 信号与噪声加失真比 | | 45 | 47 | | dB |
| 总谐波失真 | 到 5 次谐波 | | -51 | | dB |
| 有效动态范围 | | | 52 | | dB |
| 转换速率 | | | | | |
| SAR 转换时钟 | | | | 6 | MHz |
| 转换占用 SAR 时钟数 | | 8 | | | 周期 |
| 跟踪/保持捕获时间 | | 300 | | | ns |
| 最大转换速率 | | | | 500 | Ksps |
| 模拟输入 | | | | | |
| 电压转换范围 | | 0 | | VREF | V |
| 输入电容 | | | 10 | | pF |
| 电源指标 | | | | | |
| 电源电流 (AV+ 给 ADC1 供电) | 工作方式, 500ksps | | 420 | 900 | μA |
| 电源抑制比 | | | ±0.3 | | mV/V |

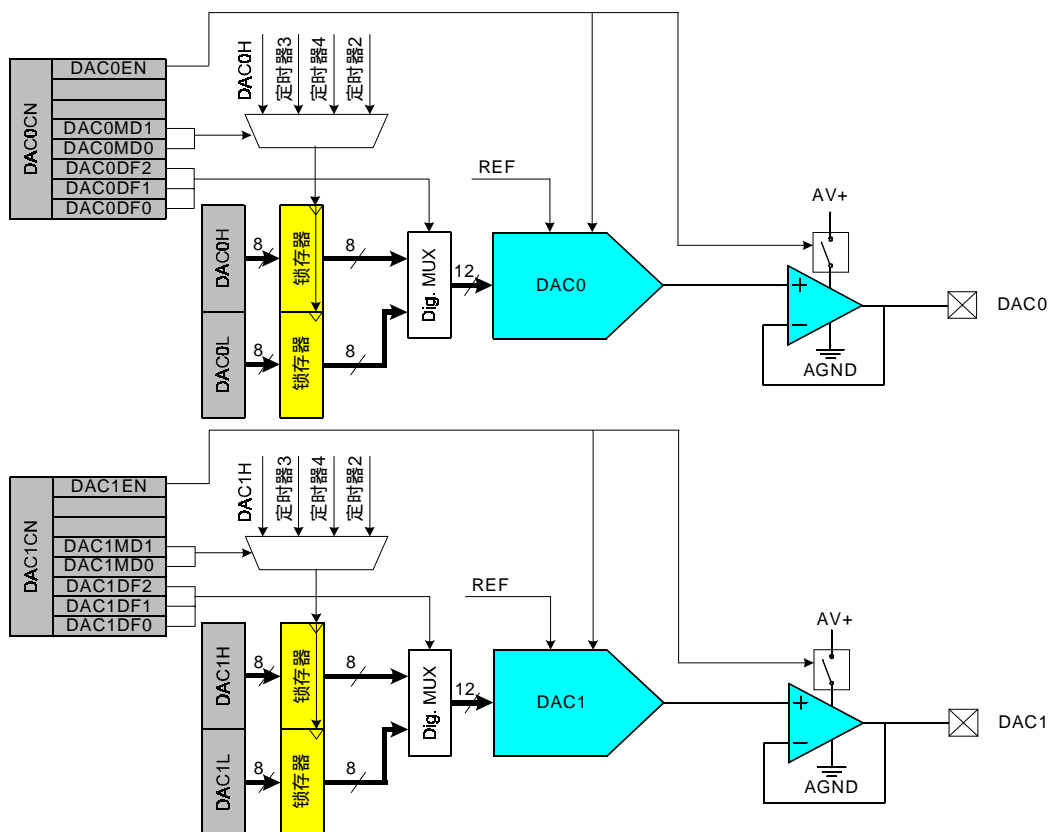
8. 12 位电压输出 DAC

每个 C8051F020/1/2/3 器件都有两个片内 12 位电压方式数/模转换器 (DAC)。每个 DAC 的输出摆幅均为 0V 到 (VREF-1LSB)，对应的输入码范围是 0x000 到 0xFFF。可以用对应的控制寄存器 DAC0CN 和 DAC1CN 允许/禁止 DAC0 和 DAC1。在被禁止时，DAC 的输出保持在高阻状态，DAC 的供电电流降到 1 μ A 或更小。每个 DAC 的电压基准在 VREFD (C8051F020/2) 或 VREF (C8051F021/3) 引脚提供。注意：C8051F021/3 的 VREF 引脚可以由内部电压基准或一个外部源驱动。如果使用内部电压基准，为了使 DAC 输出有效，该基准必须被使能。有关配置 DAC 电压基准的详细信息，见“9. 电压基准 (C8051F020/2)”或“10. 电压基准 (C8051F021/3)”。

8.1 DAC 输出更新

每个 DAC 都具有灵活的输出更新机制，允许无缝的满度变化并支持无抖动输出更新，适合于波形发生器应用。下面的描述都是以 DAC0 为例，DAC1 的操作与 DAC0 完全相同。

图 8.1 DAC 功能框图



8.1.1 根据软件命令更新输出

在缺省方式下 (DAC0CN.[4:3] = '00')，DAC0 的输出在写 DAC0 数据寄存器高字节 (DAC0H) 时更新。注意：写 DAC0L 时数据被保持，对 DAC0 输出没有影响，直到对 DAC0H 的写操作发生。如果向 DAC 数据寄存器写入一个 12 位字，则 12 位的数据字被写到低字节

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

(DAC0L) 和高字节 (DAC0H) 数据寄存器。在写 DAC0H 寄存器后数据被锁存到 DAC0。因此, 如果需要 12 位分辨率, 应在写入 DAC0L 之后写 DAC0H。DAC 可被用于 8 位方式, 这种情况是将 DAC0L 初始化一个所希望的数值 (通常为 0x00), 将数据只写入 DAC0H (有关在 16 位 SFR 空间内对 12 位 DAC 数据字格式化的说明见 8.2 节)。

8.1.2 基于定时器溢出的输出更新

在 ADC 转换操作中, ADC 转换可以由定时器溢出启动, 不用处理器干预。与之类似, DAC 的输出更新也可以用定时器溢出事件触发。这一特点在用 DAC 产生一个固定采样频率的波形时尤其有用, 可以消除中断响应时间不同和指令执行时间不同对 DAC 输出时序的影响。当 DAC0MD 位 (DAC0CN.[4:3]) 被设置为 '01'、'10' 或 '11' 时, 对 DAC 数据寄存器的写操作被保持, 直到相应的定时器溢出事件 (分别为定时器 3、定时器 4 或定时器 2) 发生时 DAC0H:DAC0L 的内容才被复制到 DAC 输入锁存器, 允许 DAC 数据改变为新值。

8.2 DAC 输出定标/调整

在某些情况下, 对 DAC0 进行写入操作之前应对输入数据移位, 以正确调整 DAC 输入寄存器中的数据。这种操作一般需要一个或多个装入和移位指令, 因而增加软件开销和降低 DAC 的数据通过率。为了减少这方面的负担, 数据格式化功能为用户提供了一种能对数据寄存器 DAC0H 和 DAC0L 中的数据格式编程的手段。三个 DAC0DF 位 (DAC0CN.[2:0]) 允许用户在 5 种数据字格式指定一种, 见 DAC0CN 寄存器定义。

DAC1 的功能与上述 DAC0 的功能完全相同。表 8.1 给出了 DAC0 和 DAC1 的电气特性。

图 8.2 DAC0H: DAC0 高字节寄存器

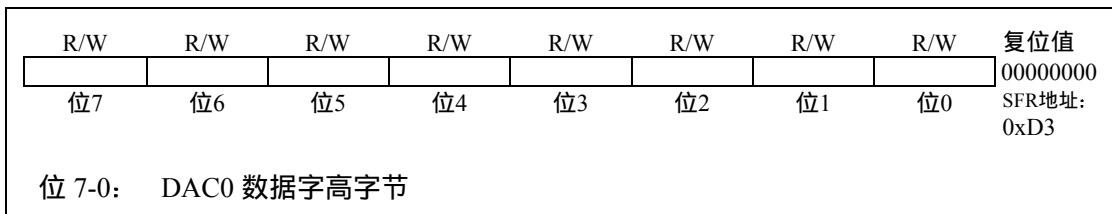
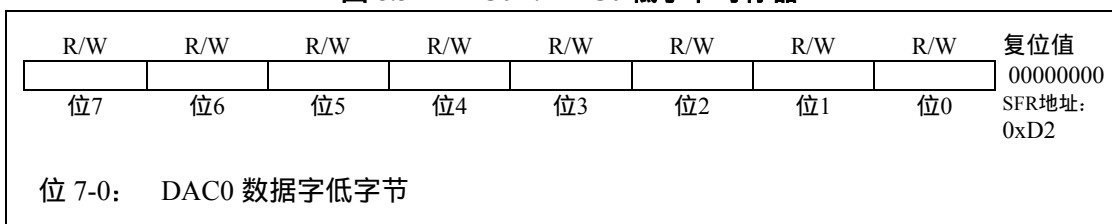


图 8.3 DAC0L: DAC0 低字节寄存器



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 8.4 DAC0CN: DAC0 控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--------|-----|-----|---------|---------|---------|---------|---------|----------------|
| DAC0EN | - | - | DAC0MD1 | DAC0MD0 | DAC0DF2 | DAC0DF1 | DAC0DF0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xD4 |

位 7: DAC0EN: DAC0 允许位
 0: DAC0 禁止。DAC0 输出引脚为高阻态，DAC0 处于节电停机方式
 1: DAC0 允许。DAC0 正常输出；DAC0 处于工作状态。

位 6-5: 未用。读 = 00b；写 = 忽略。

位 4-3: DAC0MD1-0: DAC0 方式位。
 00: DAC 输出更新发生在写 DAC0H 时。
 01: DAC 输出更新发生在定时器 3 溢出时。
 10: DAC 输出更新发生在定时器 4 溢出时。
 11: DAC 输出更新发生在定时器 2 溢出时。

位 2-0: DAC0DF2-0: DAC0 数据格式位：
 000: DAC0 数据字的高 4 位在 DAC0H[3:0]，低字节在 DAC0L 中。

| | | | | | | | |
|-------|--|--|-----|-------|--|--|-----|
| DAC0H | | | | DAC0L | | | |
| | | | MSB | | | | LSB |

 001: DAC0 数据字的高 5 位在 DAC0H[4:0]，低 7 位在 DAC0L[7:1]。

| | | | | | | | |
|-------|--|--|-----|-------|--|--|-----|
| DAC0H | | | | DAC0L | | | |
| | | | MSB | | | | LSB |

 010: DAC0 数据字的高 6 位在 DAC0H[5:0]，低 6 位在 DAC0L[7:2]。

| | | | | | | | |
|-------|--|--|-----|-------|--|--|-----|
| DAC0H | | | | DAC0L | | | |
| | | | MSB | | | | LSB |

 011: DAC0 数据字的高 7 位在 DAC0H[6:0]，低 5 位在 DAC0L[7:3]。

| | | | | | | | |
|-------|--|--|-----|-------|--|--|-----|
| DAC0H | | | | DAC0L | | | |
| | | | MSB | | | | LSB |

 1xx: 高有效字节在 DAC0H 中，低 4 位在 DAC0L[7:4]。

| | | | | | | | |
|-------|--|--|-----|-------|--|--|-----|
| DAC0H | | | | DAC0L | | | |
| | | | MSB | | | | LSB |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 8.5 DAC1H: DAC1 高字节寄存器

| | | | | | | | | |
|--------------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| 位 7-0: DAC1 数据字高字节 | | | | | | | | SFR地址: 0xD6 |

图 8.6 DAC1L: DAC1 低字节寄存器

| | | | | | | | | |
|--------------------|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| 位 7-0: DAC1 数据字低字节 | | | | | | | | SFR地址: 0xD5 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 8.7 DAC1CN: DAC1 控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|-----|-----|---------|---------|---------|---------|---------|----------------|-------|--|--|--|-------|--|--|--|--|--|--|-----|--|--|--|-----|-------|--|--|--|-------|--|--|--|--|--|--|-----|--|--|--|-----|-------|--|--|--|-------|--|--|--|--|--|--|-----|--|--|--|-----|-------|--|--|--|-------|--|--|--|--|--|--|-----|--|--|--|-----|-------|--|--|--|-------|--|--|--|--|--|--|-----|--|--|--|-----|
| DAC1EN | - | - | DAC1MD1 | DAC1MD0 | DAC1DF2 | DAC1DF1 | DAC1DF0 | 00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xD7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>位 7: DAC1EN: DAC1 允许位 0: DAC1 禁止。DAC1 输出引脚为高阻态，DAC1 处于节电停机方式 1: DAC1 允许。DAC1 正常输出，DAC1 处于工作状态。</p> <p>位 6-5: 未用。读 = 00b; 写 = 忽略</p> <p>位 4-3: DAC1MD1-0: DAC1 方式位。 00: DAC 输出更新发生在写 DAC1H 时。 01: DAC 输出更新发生在定时器 3 溢出时。 10: DAC 输出更新发生在定时器 4 溢出时。 11: DAC 输出更新发生在定时器 2 溢出时。</p> <p>位 2-0: DAC1DF2-0: DAC1 数据格式位: 000: DAC1 数据字的高 4 位在 DAC1H[3:0], 低字节在 DAC1L 中。</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <tr> <td colspan="4" style="text-align: center;">DAC1H</td> <td colspan="4" style="text-align: center;">DAC1L</td> </tr> <tr> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: center;">MSB</td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: center;">LSB</td> </tr> </table> <p>001: DAC1 数据字的高 5 位在 DAC1H[4:0], 低 7 位在 DAC1L[7:1]。</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <tr> <td colspan="4" style="text-align: center;">DAC1H</td> <td colspan="4" style="text-align: center;">DAC1L</td> </tr> <tr> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: center;">MSB</td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: center;">LSB</td> </tr> </table> <p>010: DAC1 数据字的高 6 位在 DAC1H[5:0], 低 6 位在 DAC1L[7:2]。</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <tr> <td colspan="4" style="text-align: center;">DAC1H</td> <td colspan="4" style="text-align: center;">DAC1L</td> </tr> <tr> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: center;">MSB</td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: center;">LSB</td> </tr> </table> <p>011: DAC1 数据字的高 7 位在 DAC1H[6:0], 低 5 位在 DAC1L[7:3]。</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <tr> <td colspan="4" style="text-align: center;">DAC1H</td> <td colspan="4" style="text-align: center;">DAC1L</td> </tr> <tr> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: center;">MSB</td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: center;">LSB</td> </tr> </table> <p>1xx: DAC1 数据字的高字节在 DAC1H 中, 低 4 位在 DAC1L[7:4]。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="4" style="text-align: center;">DAC1H</td> <td colspan="4" style="text-align: center;">DAC1L</td> </tr> <tr> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: center;">MSB</td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: center;">LSB</td> </tr> </table> | | | | | | | | | DAC1H | | | | DAC1L | | | | | | | MSB | | | | LSB | DAC1H | | | | DAC1L | | | | | | | MSB | | | | LSB | DAC1H | | | | DAC1L | | | | | | | MSB | | | | LSB | DAC1H | | | | DAC1L | | | | | | | MSB | | | | LSB | DAC1H | | | | DAC1L | | | | | | | MSB | | | | LSB |
| DAC1H | | | | DAC1L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | MSB | | | | LSB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DAC1H | | | | DAC1L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | MSB | | | | LSB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DAC1H | | | | DAC1L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | MSB | | | | LSB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DAC1H | | | | DAC1L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | MSB | | | | LSB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DAC1H | | | | DAC1L | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | MSB | | | | LSB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 8.1. DAC 电气特性。

VDD=3.0V, AV+=3.0V, VREF=2.40V(REFBE=0), 无输出负载 (除非另有说明)

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|----------------------|---|-----|------------------|--------------|--------|
| 静态性能 | | | | | |
| 分辨率 | | | 12 | | 位 |
| 积分非线性度 | 数据字从 0x014 到 0xFEB | | ±2 | | LSB |
| 微分非线性度 | 保证单调(数据从 0x014 到 0xFEB) | | | ±1 | LSB |
| 输出噪声 | 无输出滤波器 100kHz 输出滤波器 10kHz 输出滤波器 | | 250 128 41 | | μVrms |
| 偏移误差 | 数据字= 0x014 | | ±3 | ±30 | mV |
| 偏移误差温度系数 | | | 6 | | ppm/°C |
| 增益误差 | | | ±20 | ±60 | mV |
| 增益误差温度系数 | | | 10 | | ppm/°C |
| VDD 电源抑制率 | | | -60 | | dB |
| 关断方式下的 输出阻抗 | DACnEN=0 | | 100 | | kΩ |
| 输出灌电流 | | | 300 | | μA |
| 输出短路电流 | 数据字=0xFF | | 15 | | mA |
| 动态性能 | | | | | |
| 输出压摆率 | 负载= 40pF | | 0.44 | | V/μS |
| 输出建立时间 (到 ½ LSB) | 负载= 40pF, 输出摆幅为从数 据字 0xFF 到 0x014 | | 10 | | μS |
| 输出电压摆幅 | | 0 | | REF- 1LSB | V |
| 启动时间 | DAC 允许有效 | | 10 | | μS |
| 模拟输出 | | | | | |
| 负载调整率 | I _L = 0.01mA 到 0.3mA 当数据字为 0xFF 时 | | 60 | | ppm |
| 消耗电流 (每个 DAC) | | | | | |
| 电源电流(AV+供电 给 DAC) | 数据字= 0x7FF | | 110 | 400 | μA |

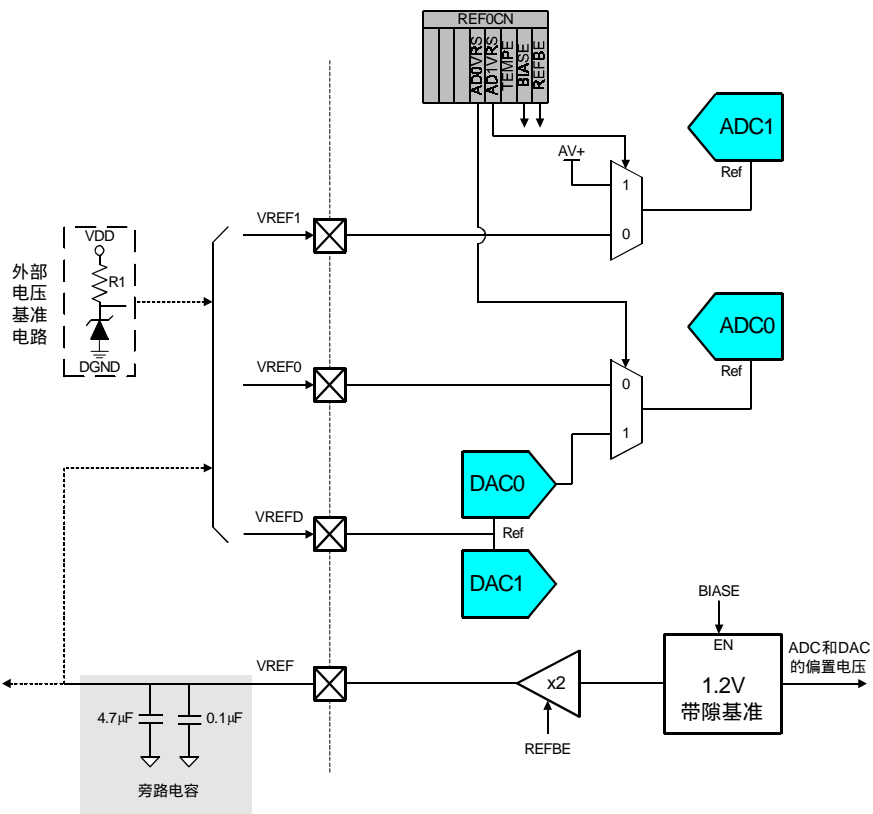
9. 电压基准 (C8051F020/2)

电压基准电路为控制 ADC 和 DAC 模块工作提供了灵活性。有三个电压基准输入引脚，允许每个 ADC 和两个 DAC 使用一个外部电压基准或片内电压基准输出。通过配置 VREF 模拟开关，ADC0 还可以使用 DAC0 的输出作为内部基准，ADC1 可以使用模拟电源电压作为基准，见图 9.1。

内部电压基准电路由一个 1.2V、15ppm/°C（典型值）的带隙电压基准发生器和一个两倍增益的输出缓冲放大器组成。内部基准电压可以通过 VREF 引脚连到应用系统中的外部器件或图 9.1 所示的电压基准输入引脚。建议在 VREF 引脚与 AGND 之间接入 0.1 μF 和 4.7 μF 的旁路电容，如图 9.1 所示。

基准电压控制寄存器 REF0CN（见图 9.2）提供了允许/禁止内部基准发生器和选择 ADC0、ADC1 基准输入的手段。REF0CN 中的 BIASE 位控制片内电压基准发生器工作，而 REFBE 位控制驱动 VREF 引脚的缓冲放大器。当被禁止时，带隙基准和缓冲放大器消耗的电流小于 1μA（典型值），缓冲放大器的输出进入高阻状态。如果要使用内部带隙基准作为基准电压发生器，则 BIASE 和 REFBE 位必须被置‘1’。如果不使用内部基准，REFBE 位必须被清‘0’。注意：如果使用 ADC 或 DAC，则不管电压基准取自片内还是片外，BIASE 位必须被置为逻辑‘1’。如果既不使用 ADC 也不使用 DAC，则这两位都应被清‘0’以节省功耗。AD0VRS 和 AD1VRS 位分别用于选择 ADC0 和 ADC1 的电压基准源。表 9.1 给出了电压基准的电气特性。

图 9.1 电压基准功能框图



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

温度传感器接在 ADC0 输入多路开关的最后一个输入端（详见“5.1 模拟多路开关和 PGA（C8051F020/1）”或“6.1 模拟多路开关和 PGA（C8051F022/3）”）。REF0CN 中的 TEMPE 位用于允许和禁止温度传感器。当被禁止时，温度传感器为缺省的高阻状态，此时对温度传感器的任何 A/D 测量结果都是无意义的。

图 9.2 REF0CN: 电压基准控制寄存器

| | | | | | | | | |
|-----|-----|-----|--------|--------|-------|-------|-------|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| - | - | - | AD0VRS | AD1VRS | TEMPE | BIASE | REFBE | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xD1 |

位 7-5: 未用。读 = 000b, 写 = 忽略。
 位 4: AD0VRS: ADC0 电压基准选择位
 0: ADC0 电压基准取自 VREF0 引脚。
 1: ADC0 电压基准取自 DAC0 输出。
 位 3: AD1VRS: ADC1 电压基准选择位
 0: ADC1 电压基准取自 VREF1 引脚。
 1: ADC1 电压基准取自 AV+。
 位 2: TEMPE: 温度传感器允许位
 0: 内部温度传感器关闭。
 1: 内部温度传感器工作。
 位 1: BIASE: ADC/DAC 偏压发生器允许位（使用 ADC 和 DAC 时该位必须为 1）
 0: 内部偏压发生器关闭。
 1: 内部偏压发生器工作。
 位 0: REFBE: 内部电压基准缓冲器允许位
 0: 内部电压基准缓冲器关闭。
 1: 内部电压基准缓冲器工作。内部电压基准提供从 VREF 引脚输出。

表 9.1 电压基准的电气特性

VDD=3.0V, AV+=3.0V, -40°C到+85°C（除非另有说明）

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|----------------------|-------------------------|------|------|---------------|--------|
| 内部基准（REFBE=1） | | | | | |
| 输出电压 | 环境温度 25°C | 2.36 | 2.43 | 2.48 | V |
| VREF 短路电流 | | | | 30 | mA |
| VREF 温度系数 | | | 15 | | ppm/°C |
| 负载调整 | 负载 = 0~200μA 到 AGND | | 0.5 | | ppm/μA |
| VREF 开启时间 1 | 4.7μF 钽电容, 0.1μF 陶瓷旁路电容 | | 2 | | ms |
| VREF 开启时间 2 | 0.1μF 陶瓷旁路电容 | | 20 | | μs |
| VREF 开启时间 3 | 无旁路电容 | | 10 | | μs |
| 外部基准（REFBE=0） | | | | | |
| 输入电压范围 | | 1.00 | | (AV+) -0.3 | V |
| 输入电流 | | | 0 | 1 | μA |

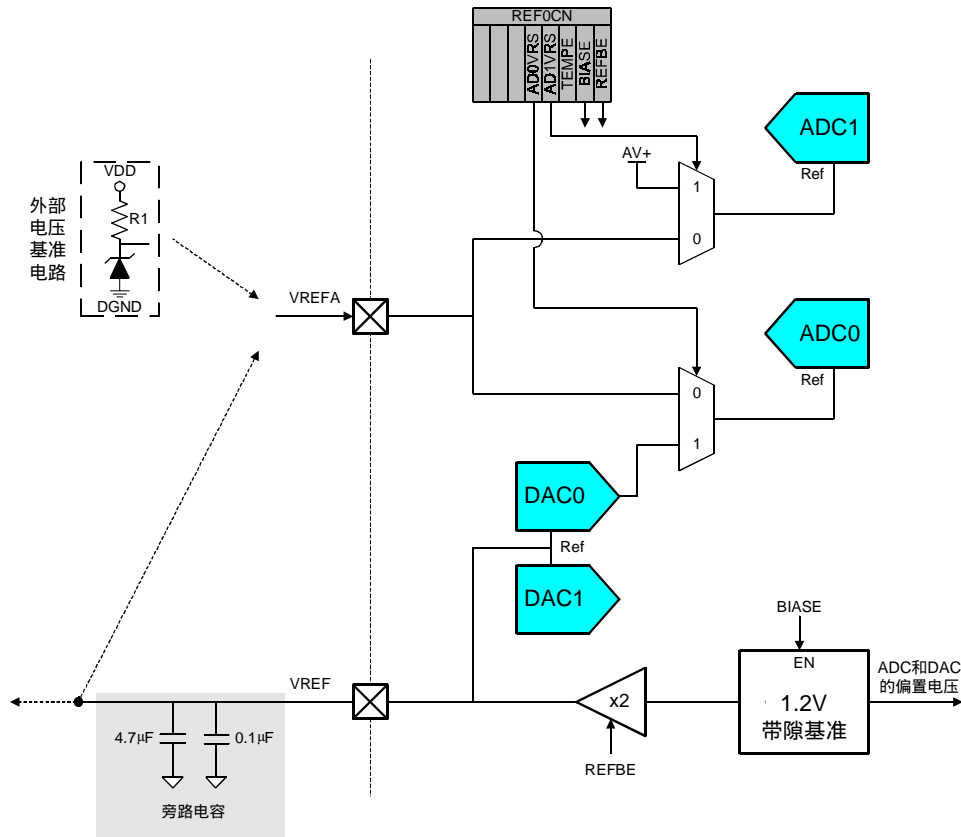
10. 电压基准 (C8051F021/3)

内部电压基准电路由一个 1.2V、15ppm/°C (典型值) 的带隙电压基准发生器和一个两倍增益的输出缓冲放大器组成。内部基准电压可以通过 VREF 引脚连到应用系统中的外部器件或图 10.1 所示的 VREFA 输入引脚。建议在 VREF 引脚与 AGND 之间接入 0.1 μF 和 4.7 μF 的旁路电容, 如图 10.1 所示。

VREFA 引脚为 ADC0 和 ADC1 提供电压基准输入。通过配置 VREF 模拟开关, ADC0 还可以使用 DAC0 的输出作为内部基准, ADC1 可以使用模拟电源电压作为基准, 见图 10.1。

基准电压控制寄存器 REF0CN(见图 10.2)提供了允许/禁止内部基准发生器和选择 ADC0、ADC1 基准输入的手段。REF0CN 中的 BIAS 位控制片内电压基准发生器工作, 而 REFBE 位控制驱动 VREF 引脚的缓冲放大器。当被禁止时, 带隙基准和缓冲放大器消耗的电流小于 1μA (典型值), 缓冲放大器的输出进入高阻状态。如果要使用内部带隙基准作为基准电压发生器, 则 BIAS 和 REFBE 位必须被置 '1'。如果不使用内部基准, REFBE 位必须被清 '0'。注意: 如果使用 ADC 或 DAC, 则不管电压基准取自片内还是片外, BIAS 位必须被置为逻辑 '1'。如果既不使用 ADC 也不使用 DAC, 则这两位都应被清 '0' 以节省功耗。AD0VRS 和 AD1VRS 位分别用于选择 ADC0 和 ADC1 的电压基准源。表 10.1 给出了电压基准的电气特性。

图 10.1 电压基准功能框图



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

温度传感器接在 ADC0 输入多路开关的最后一个输入端（详见“5.1 模拟多路开关和 PGA（C8051F020/1）”或“6.1 模拟多路开关和 PGA（C8051F022/3）”）。REF0CN 中的 TEMPE 位用于允许和禁止温度传感器。当被禁止时，温度传感器为缺省的高阻状态，此时对温度传感器的任何 A/D 测量结果都是无意义的。

图 10.2 REF0CN: 电压基准控制寄存器

| | | | | | | | | |
|-----|-----|-----|--------|--------|-------|-------|-------|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| - | - | - | AD0VRS | AD1VRS | TEMPE | BIASE | REFBE | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xD1 |

位 7-5: 未用。读 = 000b, 写 = 忽略。
 位 4: AD0VRS: ADC0 电压基准选择位
 0: ADC0 电压基准取自 VREFA 引脚。
 1: ADC0 电压基准取自 DAC0 输出。
 位 3: AD1VRS: ADC1 电压基准选择位
 0: ADC1 电压基准取自 VREFA 引脚。
 1: ADC1 电压基准取自 AV+。
 位 2: TEMPE: 温度传感器允许位
 0: 内部温度传感器关闭。
 1: 内部温度传感器工作。
 位 1: BIASE: ADC/DAC 偏压发生器允许位（使用 ADC 和 DAC 时该位必须为 1）
 0: 内部偏压发生器关闭。
 1: 内部偏压发生器工作。
 位 0: REFBE: 内部电压基准缓冲器允许位
 0: 内部电压基准缓冲器关闭。
 1: 内部电压基准缓冲器工作。内部电压基准提供从 VREF 引脚输出。

表 10.1 电压基准的电气特性

VDD=3.0V, AV+=3.0V, -40°C到+85°C（除非另有说明）

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|----------------------|-------------------------|------|------|---------------|--------|
| 内部基准（REFBE=1） | | | | | |
| 输出电压 | 环境温度 25°C | 2.36 | 2.43 | 2.48 | V |
| VREF 短路电流 | | | | 30 | mA |
| VREF 温度系数 | | | 15 | | ppm/°C |
| 负载调整率 | 负载 = (0 至 200μA)到 AGND | | 0.5 | | ppm/μA |
| VREF 开启时间 1 | 4.7μF 钽电容, 0.1μF 陶瓷旁路电容 | | 2 | | ms |
| VREF 开启时间 2 | 0.1μF 陶瓷旁路电容 | | 20 | | μs |
| VREF 开启时间 3 | 无旁路电容 | | 10 | | μs |
| 外部基准（REFBE=0） | | | | | |
| 输入电压范围 | | 1.00 | | (AV+) -0.3 | V |
| 输入电流 | | | 0 | 1 | μA |

11. 比较器

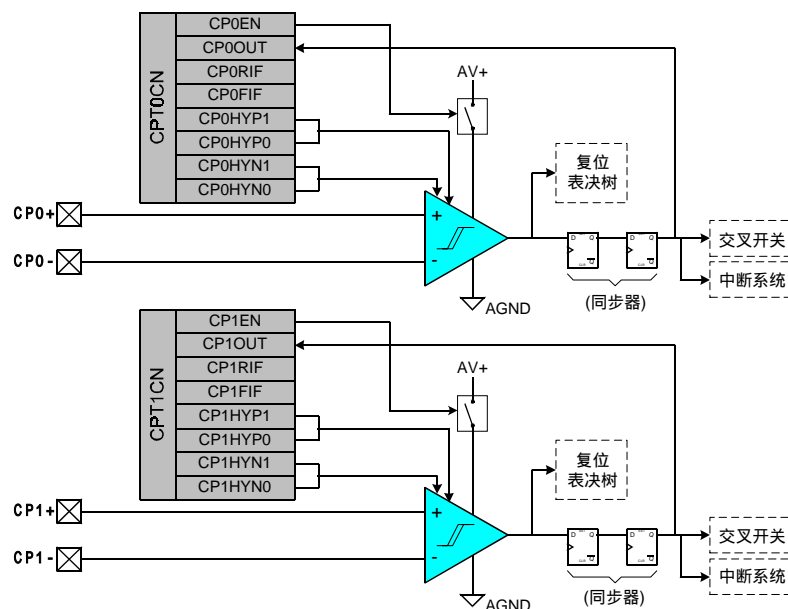
该 MCU 系列有两个片内电压比较器，如图 11.1 所示。每个比较器都有输入引脚。每个比较器的输出都可以经 I/O 交叉开关连到外部引脚。当被分配了封装引脚时，每个比较器输出都可以被编程为工作在漏极开路或推挽方式。关于交叉开关和端口初始化的详细信息见“17. 端口输入/输出”。

每个比较器的回差电压都可以通过对应的比较器控制寄存器（CPT0CN 和 CPT1CN 分别对应比较器 0 和比较器 1）用软件编程。用户既可以对回差电压值（这里指输入电压）编程，也可以对门限电压两侧的正向和负向回差对称度编程。比较器的输出可以被软件查询，也可以作为中断源。每个比较器可以被单独允许或禁止（关断）。当被禁止时，比较器的输出（如果已通过交叉开关分配到端口 I/O 引脚）缺省值为逻辑低电平，它的中断能力被停止，电源电流降到小于 1 μ A。比较器的输入可以承受 -0.25V 到 $(AV+) + 0.25V$ 的外部驱动电压而不至损坏或发生工作错误。

使用比较器 0 控制寄存器 CPT0CN（见图 11.1）中的位 3-0 对比较器 0 的回差值进行编程。负向回差电压值由 CP0HYN 位的设置决定。类似地，通过设置 CP0HYP 位决定正向回差电压值。有关回差电压指标见表 11.1。

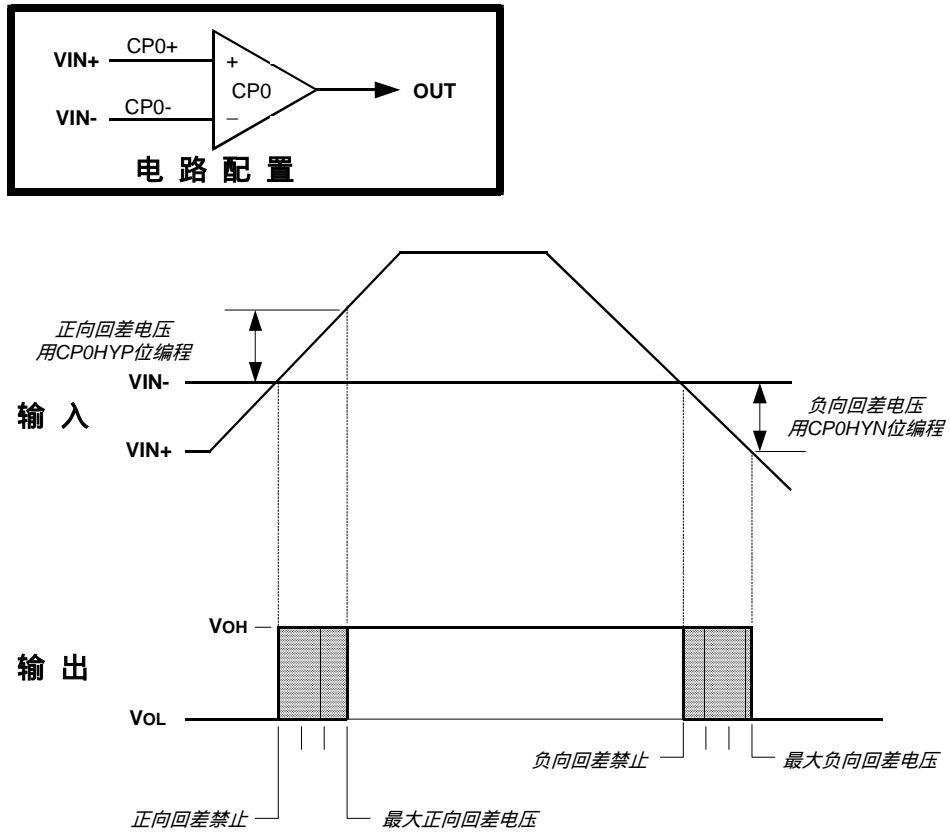
在比较器的上升沿和/或下降沿都可以产生中断。（有关中断允许和优先级控制的内容见“12.3 中断系统”）。比较器的下降沿中断置‘1’ CP0FIF 标志，比较器的上升沿中断置‘1’ CP0RIF 标志。这些位一旦被置‘1’，将一直保持‘1’状态直到被软件清除。可以在任意时刻通过读取 CP0OUT 位得到比较器 0 的输出状态。注意，在上电后直到比较器能稳定工作之前应忽略比较器输出和中断。通过置‘1’ CP0EN 位允许比较器 0 工作，通过清除该位禁止比较器 0。比较器 0 还可被编程为复位源，详见“13.6 比较器 0 复位”。

图 11.1 比较器功能框图



比较器 1 的操作与比较器 0 完全相同，只是比较器 1 不能被配置为复位源。比较器 1 受 CPT1CN 寄存器（图 11.4）控制。表 11.1 给出了比较器的详细电特性。

图 11.2 比较器回差电压曲线



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 11.3 CPT0CN: 比较器 0 控制寄存器

| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | | |
|--------|---|--------|--------|--------|---------|---------|---------|---------|-----------------|
| | CP0EN | CP0OUT | CP0RIF | CP0FIF | CP0HYP1 | CP0HYP0 | CP0HYN1 | CP0HYN0 | 复位值 00000000 |
| | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x9E |
| 位 7: | CP0EN: 比较器 0 允许位 0: 比较器 0 禁止。 1: 比较器 0 允许。 | | | | | | | | |
| 位 6: | CP0OUT: 比较器 0 输出状态标志 0: 电压值 CP0+ < CP0- 1: 电压值 CP0+ > CP0- | | | | | | | | |
| 位 5: | CP0RIF: 比较器 0 上升沿中断标志 0: 自该标志位被清除后, 没有发生过比较器 0 上升沿中断 1: 自该标志位被清除后, 发生了比较器 0 上升沿中断 | | | | | | | | |
| 位 4: | CP0FIF: 比较器 0 下降沿中断标志 0: 自该标志位被清除后, 没有发生比较器 0 下降沿中断 1: 自该标志位被清除后, 发生了比较器 0 下降沿中断 | | | | | | | | |
| 位 3-2: | CP0HYP1-0: 比较器 0 正向回差电压控制位 00: 禁止正向回差电压 01: 正向回差电压=2mV 10: 正向回差电压=4mV 11: 正向回差电压=10mV | | | | | | | | |
| 位 1-0: | CP0HYN1-0: 比较器 0 负向回差电压控制位 00: 禁止负向回差电压 01: 负向回差电压=2mV 10: 负向回差电压=4mV 11: 负向回差电压=10mV | | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 11.4 CPT1CN: 比较器 1 控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--|--------|--------|--------|---------|---------|---------|---------|----------------|
| CPIEN | CPIOUT | CPIRIF | CPIFIF | CPIHYP1 | CPIHYP0 | CPIHYN1 | CPIHYN0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x9F |
| <p>位 7: CPIEN: 比较器 1 允许位 0: 比较器 1 禁止。 1: 比较器 1 允许。</p> <p>位 6: CPIOUT: 比较器 1 输出状态位 0: 电压值 $CP1+ < CP1-$ 1: 电压值 $CP1+ > CP1-$</p> <p>位 5: CPIRIF: 比较器 1 上升沿中断标志 0: 自该标志位被清除后, 没有发生比较器 1 上升沿中断 1: 自该标志位被清除后, 发生了比较器 1 上升沿中断</p> <p>位 4: CPIFIF: 比较器 1 下降沿中断标志 0: 自该标志位被清除后, 没有发生比较器 1 下降沿中断 1: 自该标志位被清除后, 发生了比较器 1 下降沿中断</p> <p>位 3-2: CPIHYP1-0: 比较器 1 正向回差电压控制位 00: 禁止正向回差电压 01: 正向回差电压=2mV 10: 正向回差电压=4mV 11: 正向回差电压=10mV</p> <p>位 1-0: CPIHYN1-0: 比较器 1 负向回差电压控制位 00: 禁止负向回差电压 01: 负向回差电压=2mV 10: 负向回差电压=4mV 11: 负向回差电压=10mV</p> | | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 11.1. 比较器电气特性

VDD=3.0V, AV+=3.0V, -40°C到+85°C (除非另有说明)

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|-----------------|-----------------------|-------|-------|----------------|------|
| 响应时间 1 | (CP+) - (CP-) = 100mV | | 4 | | μS |
| 响应时间 2 | (CP+) - (CP-) = 10mV | | 12 | | μS |
| 共模抑制比 | | | 1.5 | 4 | mV/V |
| 正向回差电压 1 | CPnHYP1-0 = 00 | | 0 | 1 | mV |
| 正向回差电压 2 | CPnHYP1-0 = 01 | 2 | 4.5 | 7 | mV |
| 正向回差电压 3 | CPnHYP1-0 = 10 | 4 | 9 | 13 | mV |
| 正向回差电压 4 | CPnHYP1-0 = 11 | 10 | 17 | 25 | mV |
| 负向回差电压 1 | CPnHYN1-0 = 00 | | 0 | 1 | mV |
| 负向回差电压 2 | CPnHYN1-0 = 01 | 2 | 4.5 | 7 | mV |
| 负向回差电压 3 | CPnHYN1-0 = 10 | 4 | 9 | 13 | mV |
| 负向回差电压 4 | CPnHYN1-0 = 11 | 10 | 17 | 25 | mV |
| 反相或同相 输入电压范围 | | -0.25 | | (AV+) +0.25 | V |
| 输入电容 | | | 7 | | pF |
| 输入偏置电流 | | -5 | 0.001 | +5 | nA |
| 输入偏移电压 | | -10 | | +10 | mV |
| 电源 | | | | | |
| 上电时间 | CPnEN 从 0 到 1 | | 20 | | μS |
| 电源抑制比 | | | 0.1 | 1 | mV/V |
| 电源电流 | 在直流工作方式 (每个比较器) | | 1.5 | 10 | μA |

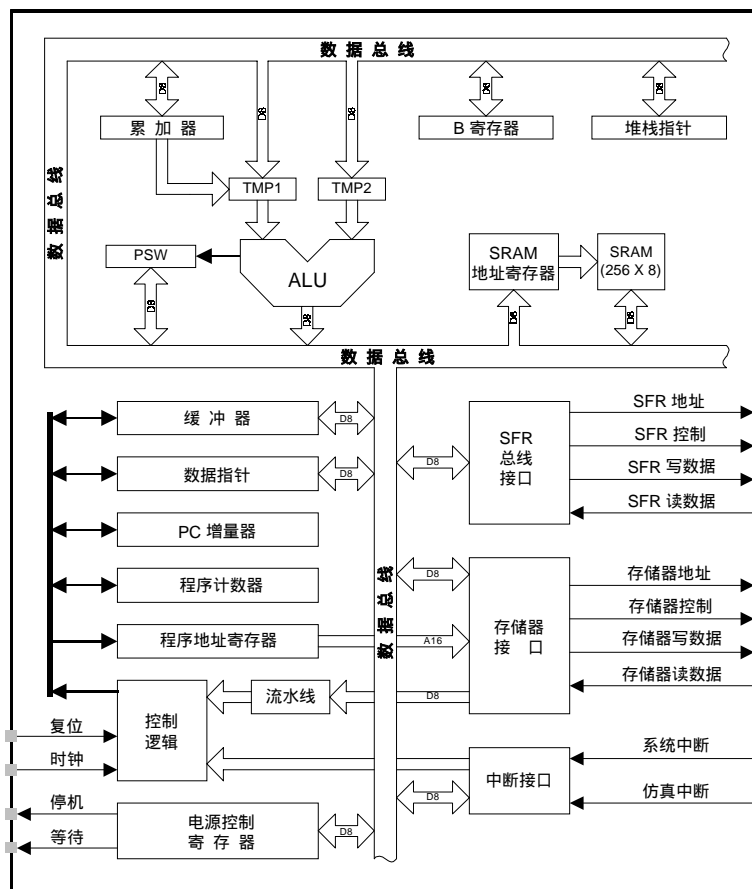
12. CIP-51 微控制器

MCU 系统控制器的内核是 CIP-51 微控制器。CIP-51 与 MCS-51™ 指令集完全兼容，可以使用标准 803x/805x 的汇编器和编译器进行软件开发。该系列 MCU 具有标准 8051 的所有外设部件，包括 5 个 16 位的计数器/定时器（详见第 22 章）、两个全双工 UART（详见第 20 章和第 21 章）、256 字节内部 RAM、128 字节特殊功能寄存器（SFR）地址空间及 8/4 个 8 位宽的 I/O 端口（详见第 17 章）。CIP-51 还包含片内调试硬件（详见第 24 章）和与 MCU 直接接口的模拟和数字子系统，在一个集成电路内提供了完整的数据采集或控制系统解决方案。

CIP-51 微控制器内核除了具有标准 8051 的组织结构和外设以外，另有增加的定制外设和功能，大大增强了它的处理能力（见图 12.1 的原理框图）。CIP-51 具有下列特点：

- 与 MCS-51 指令集完全兼容
- 在 25MHz 时钟时最大速度为 25MIPS
- 0 到 25MHz 的时钟频率
- 256 字节内部 RAM
- 8/4 个 8 位 I/O 端口
- 扩展的中断处理系统
- 复位输入
- 电源管理方式
- 片内调试逻辑
- 程序和数据存储器安全

图 12.1 CIP-51 原理框图



性能

CIP-51采用流水线结构，与标准的8051结构相比指令执行速度有很大的提高。在一个标准的8051中，除MUL和DIV以外所有指令都需要12或24个系统时钟周期，并且通常最大系统时钟频率为12 MHz。而对于CIP-51内核，70%的指令的执行时间为1或2个系统时钟周期，没有执行时间超过8个系统时钟周期的指令。

CIP-51工作在最大系统时钟频率 25MHz 时，它的峰值速度达到 25MIPS。CIP-51 共有 111 条指令。下表列出了指令条数与执行时所需的系统时钟周期数的关系。

| 执行周期数 | 1 | 2 | 2/3 | 3 | 3/4 | 4 | 4/5 | 5 | 8 |
|-------|----|----|-----|----|-----|---|-----|---|---|
| 指令 | 26 | 50 | 5 | 16 | 7 | 3 | 1 | 2 | 1 |

编程和调试支持

C8051F000系列MCU提供了JTAG串行接口，通过该接口能对FLASH程序存储器进行在系统编程并可与片内调试支持电路通信。应用程序可以使用MOVC和MOVX指令对可再编程FLASH进行读或改写，每次只能读或写一个字节。这一特性允许将程序存储器用于非易失性数据存储以及在软件控制下更新程序代码。

片内 JTAG 调试支持电路允许全速的在系统调试，支持设置硬件断点和观察点，支持开始、停止和单步执行（包括中断服务程序）命令，支持程序调用堆栈检查、读/写存储器和寄存器内容。在片调试方法完全是非侵入式的，不需要 RAM、堆栈、定时器或其它片内资源。

CIP-51 有 Cygnal 集成产品公司和第三方供应商的开发工具支持。Cygnal 提供一个集成开发环境（IDE），包括编辑器、宏汇编器、调试器和编程器。IDE 的调试器和编程器与 CIP-51 之间通过 JTAG 实现接口，提供快速和有效的在系统编程和调试。也有第三方提供的宏汇编器和 C 编译器。

12.1 指令集

CIP-51 系统控制器的指令集与标准 MCS-51™ 指令集完全兼容，可以使用标准 8051 的开发工具开发 CIP-51 的软件。所有的 CIP-51 指令在二进制码和功能上与 MCS-51™ 产品完全等价，包括操作码、寻址方式和对 PSW 标志的影响，但是指令时序与标准 8051 不同。

12.1.1 指令和 CPU 时序

在很多的 8051 产品中，机器周期和时钟周期是不同的，机器周期的长度在 2 到 12 个时钟周期之间。但是 CIP-51 只基于时钟周期，所有指令时序都以时钟周期计算。

由于 CIP-51 采用了流水线结构，大多数指令执行所需的时钟周期数与指令的字节数一致。条件转移指令在不发生转移时的执行周期数比发生转移时少一个。表 12.1 给出了 CIP-51 指令一览表，包括每条指令的助记符、字节数和时钟周期数。

12.1.2 MOVX 指令和程序存储器

在 CIP-51 中，MOVX 指令有三种作用：访问片内 XRAM、访问片外 XRAM 和访问片内 FLASH 程序存储器。CIP-51 的 FLASH 访问特性提供了由用户程序更新程序代码和将程序存储器空间用于非易失性数据存储的机制（见“15. FLASH 存储器”）。通过外部存储器接口，可用 MOVX 指令快速访问片外 XRAM（或存储器编址的外设）。详见“16. 外部数据存储器接口和片内 XRAM”。

表 12.1 CIP-51 指令集

| 助记符 | 功能说明 | 字节数 | 时钟周期数 |
|------------------|--------------------|-----|-------|
| 算术操作类指令 | | | |
| ADD A,Rn | 寄存器加到累加器 | 1 | 1 |
| ADD A,direct | 直接寻址字节加到累加器 | 2 | 2 |
| ADD A,@Ri | 间址 RAM 加到累加器 | 1 | 2 |
| ADD A,#data | 立即数加到累加器 | 2 | 2 |
| ADDC A,Rn | 寄存器加到累加器(带进位) | 1 | 1 |
| ADDC A,direct | 直接寻址字节加到累加器(带进位) | 2 | 2 |
| ADDC A,@Ri | 间址 RAM 加到累加器(带进位) | 1 | 2 |
| ADDC A,#data | 立即数加到累加器(带进位) | 2 | 2 |
| SUBB A,Rn | 累加器减去寄存器(带借位) | 1 | 1 |
| SUBB A,direct | 累加器减去间接寻址 RAM(带借位) | 2 | 2 |
| SUBB A,@Ri | 累加器减去间址 RAM(带借位) | 1 | 2 |
| SUBB A,#data | 累加器减去立即数(带借位) | 2 | 2 |
| INC A | 累加器加 1 | 1 | 1 |
| INC Rn | 寄存器加 1 | 1 | 1 |
| INC direct | 直接寻址字节加 1 | 2 | 2 |
| INC @Ri | 间址 RAM 加 1 | 1 | 2 |
| DEC A | 累加器减 1 | 1 | 1 |
| DEC Rn | 寄存器减 1 | 1 | 1 |
| DEC direct | 直接寻址字节减 1 | 2 | 2 |
| DEC @Ri | 间址 RAM 减 1 | 1 | 2 |
| INC DPTR | 数据地址加 1 | 1 | 1 |
| MUL AB | 累加器和寄存器 B 加乘 | 1 | 4 |
| DIV AB | 累加器除以寄存器 B | 1 | 8 |
| DA A | 累加器十进制调整 | 1 | 1 |
| 逻辑操作类指令 | | | |
| ANL A,Rn | 寄存器“与”到累加器 | 1 | 1 |
| ANL A,direct | 直接寻址字节“与”到累加器 | 2 | 2 |
| ANL A,@Ri | 间址 RAM “与”到累加器 | 1 | 2 |
| ANL A,#data | 立即数“与”到累加器 | 2 | 2 |
| ANL direct,A | 累加器“与”到直接寻址字节 | 2 | 2 |
| ANL direct,#data | 立即数“与”到直接寻址字节 | 3 | 3 |
| ORL A,Rn | 寄存器“或”到累加器 | 1 | 1 |
| ORL A,direct | 直接寻址字节“或”到累加器 | 2 | 2 |
| ORL A,@Ri | 间址 RAM “或”到累加器 | 1 | 2 |
| ORL A,#data | 立即数“或”到累加器 | 2 | 2 |
| ORL direct,A | 累加器“或”到直接寻址字节 | 2 | 2 |
| ORL direct,#data | 立即数“或”到直接寻址字节 | 3 | 3 |
| XRL A,Rn | 寄存器“异或”到累加器 | 1 | 1 |
| XRL A,direct | 直接寻址数“异或”到累加器 | 2 | 2 |
| XRL A,@Ri | 间址 RAM “异或”到累加器 | 1 | 2 |
| XRL A,#data | 立即数“异或”到累加器 | 2 | 2 |
| XRL direct,A | 累加器“异或”到直接寻址字节 | 2 | 2 |
| XRL direct,#data | 立即数“异或”到直接寻址字节 | 3 | 3 |
| CLR A | 累加器清零 | 1 | 1 |
| CPL A | 累加器求反 | 1 | 1 |
| RL A | 循环循环左移 | 1 | 1 |
| RLC A | 经过进位位的累加器循环左移 | 1 | 1 |
| RR A | 累加器循环右移 | 1 | 1 |

表 12.1 CIP-51 指令集 (续)

| 助记符 | 功能说明 | 字节数 | 时钟周期数 |
|-------------------|----------------------|-----|-------|
| RRC A | 经过进位位的累加器循环右移 | 1 | 1 |
| SWAP A | 累加器内高低半字节交换 | 1 | 1 |
| 数据传输类指令 | | | |
| MOV A,Rn | 寄存器传送到累加器 A | 1 | 1 |
| MOV A,direct | 直接寻址字节传送到累加器 | 2 | 2 |
| MOV A,@Ri | 间址 RAM 传送到累加器 | 1 | 2 |
| MOV A,#data | 立即数传送到累加器 | 2 | 2 |
| MOV Rn,A | 累加器传送到寄存器 | 1 | 1 |
| MOV Rn,direct | 直接寻址字节传送到寄存器 | 2 | 2 |
| MOV Rn,#data | 立即数传送到寄存器 | 2 | 2 |
| MOV direct,A | 累加器传送到直接寻址字节 | 2 | 2 |
| MOV direct,Rn | 寄存器传送到直接寻址字节 | 2 | 2 |
| MOV direct,direct | 直接寻址字节传送到直接寻址字节 | 3 | 3 |
| MOV direct,@Ri | 间址 RAM 传送到直接寻址字节 | 2 | 2 |
| MOV direct,#data | 立即数传送到直接寻址字节 | 3 | 3 |
| MOV @Ri,A | 累加器传送到间址 RAM | 1 | 2 |
| MOV @Ri,direct | 直接寻址数传送到间址 RAM | 2 | 2 |
| MOV @Ri,#data | 立即数传送到间址 RAM | 2 | 2 |
| MOV DPTR,#data16 | 16 位常数装入数据指针 | 3 | 3 |
| MOVC A,@A+DPTR | 代码字节传送到累加器 | 1 | 3 |
| MOVC A,@A+PC | 代码字节传送到累加器 | 1 | 3 |
| MOVX A,@Ri | 外部 RAM(8 位地址)数传送到 A | 1 | 3 |
| MOVX @Ri,A | 累加器传到外部 RAM (8 位地址) | 1 | 3 |
| MOVX A,@DPTR | 外部 RAM(16 位地址)传送到 A | 1 | 3 |
| MOVX @DPTR,A | 累加器传到外部 RAM (16 位地址) | 1 | 3 |
| PUSH direct | 直接寻址字节压入栈顶 | 2 | 2 |
| POP direct | 栈顶数据弹出到直接寻址字节 | 2 | 2 |
| XCH A,Rn | 寄存器和累加器交换 | 1 | 1 |
| XCH A,direct | 直接寻址字节与累加器交换 | 2 | 2 |
| XCH A,@Ri | 间址 RAM 与累加器交换 | 1 | 2 |
| XCHD A,@Ri | 间址 RAM 和累加器交换低半字节 | 1 | 2 |
| 位操作类指令 | | | |
| CLR C | 清进位位 | 1 | 1 |
| CLR bit | 清直接寻址位 | 2 | 2 |
| SETB C | 进位位置 1 | 1 | 1 |
| SETB bit | 直接寻址位置位 | 2 | 2 |
| CPL C | 进位位取反 | 1 | 1 |
| CPL bit | 直接寻址位取反 | 2 | 2 |
| ANL C,bit | 直接寻址位“与”到进位位 | 2 | 2 |
| ANL C,/bit | 直接寻址位的反码“与”到进位位 | 2 | 2 |
| ORL C,bit | 直接寻址位“或”到进位位 | 2 | 2 |
| ORL C,/bit | 直接寻址位的反码“或”到进位位 | 2 | 2 |
| MOV C,bit | 直接寻址位传送到进位位 | 2 | 2 |
| MOV bit,C | 进位位传送到直接寻址位 | 2 | 2 |
| JC rel | 若进位位为 1 则跳转 | 2 | 2/3 |
| JNC rel | 若进位位为零则跳转 | 2 | 2/3 |
| JB bit,rel | 若直接寻址位为 1 则跳转 | 3 | 3/4 |
| JNB bit,rel | 若直接寻址位为零则跳转 | 3 | 3/4 |
| JBC bit,rel | 若直接寻址位为 1 则跳转, 并清除该位 | 3 | 3/4 |

表 12.1 CIP-51 指令集 (续)

| 助记符 | 功能说明 | 字节数 | 时钟周期数 |
|--------------------|------------------------|-----|-------|
| 控制转移类指令 | | | |
| ACALL addr11 | 绝对调用子程序 | 2 | 3 |
| LCALL addr16 | 长调用子程序 | 3 | 4 |
| RET | 从子程序返回 | 1 | 5 |
| RETI | 从中断返回 | 1 | 5 |
| AJMP addr11 | 绝对转移 | 2 | 3 |
| LJMP addr16 | 长转移 | 3 | 4 |
| SJMP rel | 短转移 (相对偏移) | 2 | 3 |
| JMP @A+DPTR | 相对 DPTR 的间接转移 | 1 | 3 |
| JZ rel | 累加器为 0 则转移 | 2 | 2/3 |
| JNZ rel | 累加器为非 0 则转移 | 2 | 2/3 |
| CJNE A,direct,rel | 比较直接寻址字节与 A, 不相等则转移 | 3 | 3/4 |
| CJNE A,#data,rel | 比较立即数与 A, 不相等则转移 | 3 | 3/4 |
| CJNE Rn,#data,rel | 比较立即数与寄存器, 不相等则转移 | 3 | 3/4 |
| CJNE @Ri,#data,rel | 比较立即数与间接寻址 RAM, 不相等则转移 | 3 | 4/5 |
| DJNZ Rn,rel | 寄存器减 1, 不为零则转移 | 2 | 2/3 |
| DJNZ direct,rel | 直接寻址字节减 1, 不为零则转移 | 3 | 3/4 |
| NOP | 空操作 | 1 | 1 |

寄存器、操作数和寻址方式说明:

Rn – 当前选择的寄存器区的寄存器 R0-R7。

@Ri – 通过寄存器 R0-R1 间接寻址的数据 RAM 地址。

rel – 相对于下一条指令第一个字节的 8 位有符号 (2 的补码) 偏移量。SJMP 和所有条件转移指令使用。

direct – 8 位内部数据存储器地址。可以是直接访问数据 RAM 地址 (0x00-0x7F) 或一个 SFR 地址 (0x80-0xFF)。

#data – 8 位立即数

#data16 – 16 位立即数

bit – 数据 RAM 或 SFR 中的直接寻址位

addr11 – ACALL 或 AJMP 使用的 11 位目的地址。目的地址必须与下一条指令第一个字节处于同一个 2K 字节的程序存储器页。

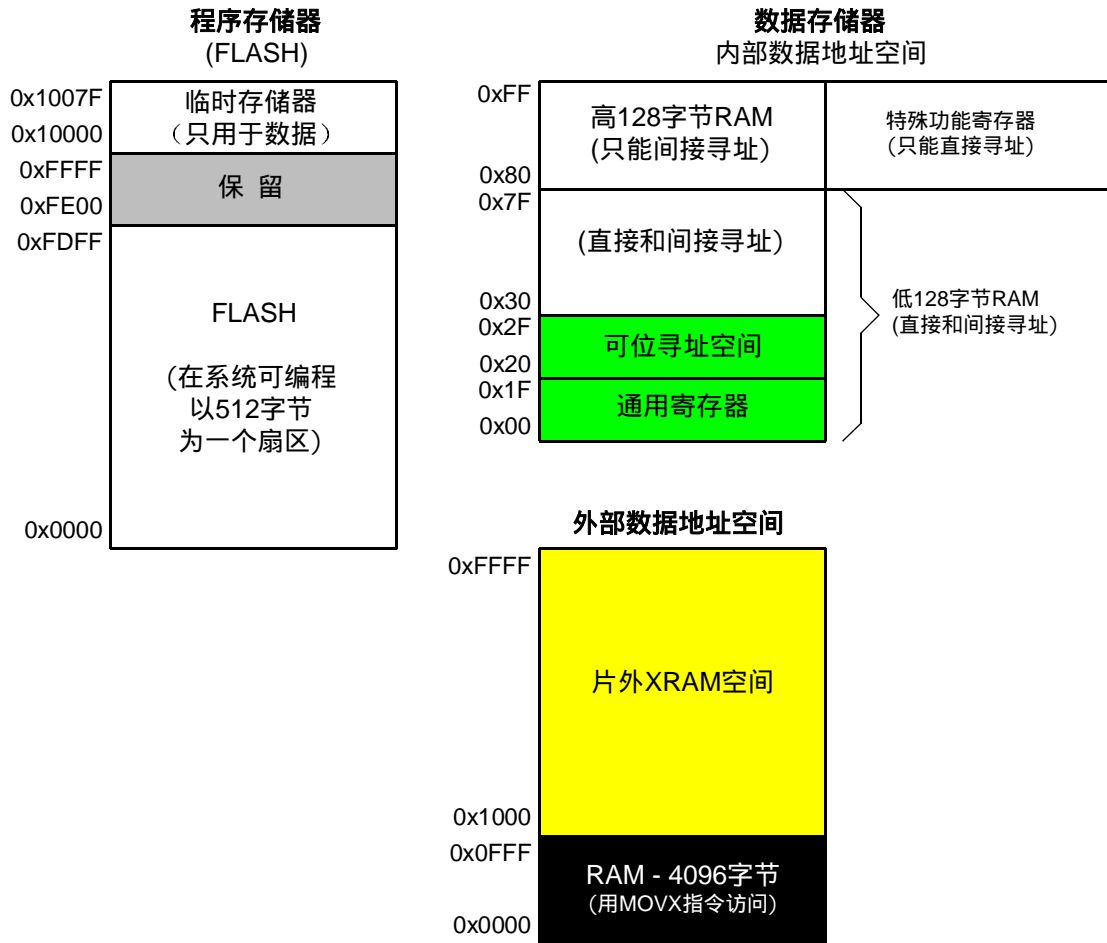
addr16 – LCALL 或 LJMP 使用的 16 位目的地址。目的地址可以是 64K 程序存储器空间内的任何位置。

有一个未使用的操作码 (0xA5), 它执行与 NOP 指令相同的功能。

12.2 存储器组织

CIP-51 系统控制器的存储器组织与标准 8051 的存储器组织类似。有两个独立的存储器空间：程序存储器和数据存储器。程序和数据存储器共享同一个地址空间，但用不同的指令类型访问。CIP-51 内部有 256 字节的内部数据存储器 and 64K 字节的内部程序存储器地址空间。CIP-51 的存储器组织如图 12.2 所示。

图 12.2 存储器结构图



12.2.1 程序存储器

CIP-51 有 64K 字节的程序存储器空间。MCU 在这个程序存储器空间中实现了 65536 字节可在系统编程的 FLASH 存储器，组织在一个连续的存储块内，从地址 0x0000 到 0xFFFF。注意：该存储器中有 512 字节 (0xEE00 – 0xFFFF) 保留给工厂使用，不能用于存储用户程序。

程序存储器通常被认为是只读的，但是 CIP-51 可以通过设置程序存储写允许位 (PSCTL.0) 用 MOVX 指令对程序存储器写入。这一特性为 CIP-51 提供了更新程序代码和将程序存储器空间用于非易失性数据存储的机制。更进一步的详细信息见“15. FLASH 存储器”。

12.2.2 数据存储器

CIP-51 的数据存储器空间中有 256 字节的内部 RAM, 位于地址 0x00 到 0xFF 的地址空间。数据存储器中的低 128 字节用于通用寄存器和临时存储器。可以用直接或间接寻址方式访问数据存储器中的低 128 字节。从 0x00 到 0x1F 为 4 个通用寄存器区, 每个区有 8 个寄存器。接下来的 16 字节, 从地址 0x20 到 0x2F, 既可以按字节寻址又可以作为 128 个位地址用直接位寻址方式访问。

数据存储器中的高 128 字节只能用间接寻址访问。该存储区与特殊功能寄存器 (SFR) 占据相同的地址空间, 但物理上与 SFR 空间是分开的。当寻址高于 0x7F 的地址时, 指令所用的寻址方式决定了 CPU 是访问数据存储器的高 128 字节还是访问 SFR。使用直接寻址方式的指令将访问 SFR 空间, 间接寻址高于 0x7F 地址的指令将访问数据存储器的高 128 字节。图 12.2 给出了 CIP-51 数据存储器组织的示意图。

12.2.3 通用寄存器

数据存储器的低 32 字节, 从地址 0x00 到 0x1F, 可以作为 4 个通用寄存器区访问。每个区有 8 个寄存器, 称为 R0 - R7。在某一时刻只能选择一个寄存器区。程序状态字中的 RS0 (PSW.3) 和 RS1 (PSW.4) 位用于选择当前的寄存器区 (见图 12.6 中关于 PSW 的说明)。这允许在进入子程序或中断服务程序时进行快速现场切换。间接寻址方式使用 R0 和 R1 作为间接寻址寄存器。

12.2.4 位寻址空间

除了直接访问按字节组织的数据存储器外, 从 0x20 到 0x2F 的 16 个数据存储器单元还可以作为 128 个独立寻址位访问。每个位有一个位地址, 从 0x00 到 0x7F。位于地址 0x20 的数据字节的位 0 具有位地址 0x00, 位于 0x20 的数据字节的位 7 具有位地址 0x07。位于 0x2F 的数据字节的位 7 具有位地址 0x7F。由所用指令的类型来区分是位寻址还是字节寻址。

MCS-51TM 汇编语言允许用 XX.B 的形式替代位地址, XX 为字节地址, B 为寻址位在字节中的位置。例如, 指令:

```
MOV    C,  22h.3
```

将 0x13 中的布尔值 (字节地址 0x22 中的位 3) 传送到用户进位标志。

12.2.5 堆栈

程序的堆栈可以位于 256 字节数据存储器中的任何位置。堆栈区域用堆栈指针 (SP, 0x81) SFR 指定。SP 指向最后使用的位置。下一个压入堆栈的数据将被存放在 SP+1, 然后 SP 加 1。复位后堆栈指针被初始化为地址 0x07, 因此第一个被压入堆栈的数据将被存放在地址 0x08, 这也是寄存器区 1 的第一个寄存器 (R0)。如果使用不止一个寄存器区, SP 应被初始化为数据存储器中不用于数据存储的位置。堆栈深度最大可达 256 字节。

MCU 内部有用于堆栈记录的硬件。堆栈记录是一个 32 位的移位寄存器, 每次压栈 (PUSH) 或 SP 增 1 都向该寄存器压入一个记录位, 每次调用或中断向该寄存器压入两个记录位。(一次出栈 (POP) 或 SP 减 1 弹出一个记录位, 一次返回弹出两个记录位。) 堆栈记录电路可以检测该 32 位移位寄存器的上溢和下溢, 即使在 MCU 运行全速运行时也可以通知调试软件。

12.2.6 特殊功能寄存器

从 0x80 到 0xFF 的直接寻址存储器空间为特殊功能寄存器 (SFR)。SFR 提供对 CIP-51 的资源和外设的控制及 CIP-51 与这些资源和外设之间的数据交换。CIP-51 具有标准 8051 中的全部 SFR，还增加了一些用于配置和访问专有子系统的 SFR。这就允许在保证与 MCS-51™ 指令集兼容的前提下增加新的功能。表 12.2 列出了 CIP-51 系统控制器中的全部 SFR。

任何时刻用直接寻址访问从 0x80 到 0xFF 的存储器空间将访问特殊功能寄存器 (SFR)。地址以 0x0 或 0x8 结尾的 SFR (例如 P0、TCON、P1、SCON、IE 等) 既可以按字节寻址也可以按位寻址，所有其它 SFR 只能按字节寻址。SFR 空间中未使用的地址保留为将来使用，访问这些地址会产生不确定的结果，应予避免。有关每个寄存器的详细说明请参见本数据表的相关部分 (表 12.3 中已标明)。

表 12.2 特殊功能寄存器存储器映象

| | | | | | | | | |
|----|--------------|---------|----------|----------|----------|----------|----------|---------|
| F8 | SPI0CN | PCA0H | PCA0CPH0 | PCA0CPH1 | PCA0CPH2 | PCA0CPH3 | PCA0CPH4 | WDTCN |
| F0 | B | SCON1 | SBUF1 | SADDR1 | TL4 | TH4 | EIP1 | EIP2 |
| E8 | ADC0CN | PCA0L | PCA0CPL0 | PCA0CPL1 | PCA0CPL2 | PCA0CPL3 | PCA0CPL4 | RSTSRC |
| E0 | ACC | XBR0 | XBR1 | XBR2 | RCAP4L | RCAP4H | EIE1 | EIE2 |
| D8 | PCA0CN | PCA0MD | PCA0CPM0 | PCA0CPM1 | PCA0CPM2 | PCA0CPM3 | PCA0CPM4 | |
| D0 | PSW | REF0CN | DAC0L | DAC0H | DAC0CN | DAC1L | DAC1H | DAC1CN |
| C8 | T2CON | T4CON | RCAP2L | RCAP2H | TL2 | TH2 | | SMB0CR |
| C0 | SMB0CN | SMB0STA | SMB0DAT | SMB0ADR | ADC0GTL | ADC0GTH | ADC0LTL | ADC0LTH |
| B8 | IP | SADEN0 | AMX0CF | AMX0SL | ADC0CF | P1MDIN | ADC0L | ADC0H |
| B0 | P3 | OSCXCN | OSCICN | | | P74OUT† | FLSCL | FLACL |
| A8 | IE | SADDR0 | ADC1CN | ADC1CF | AMX1SL | P3IF | | EMI0CN |
| A0 | P2 | EMI0TC | | EMI0CF | P0MDOUT | P1MDOUT | P2MDOUT | P3MDOUT |
| 98 | SCON | SBUF0 | SPI0CFG | SPI0DAT | ADC1 | SPI0CKR | CPT0CN | CPT1CN |
| 90 | P1 | TMR3CN | TMR3RLL | TMR3RLH | TMR3L | TMR3H | P7† | |
| 88 | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | CKCON | PSCTL |
| 80 | P0 | SP | DPL | DPH | P4† | P5† | P6† | PCON |
| | 0(8) 可位寻址 | 1(9) | 2(A) | 3(B) | 4(C) | 5(D) | 6(E) | 7(F) |

表 12.3 特殊功能寄存器

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

| 寄存器 | 地址 | 说明 | 页码 |
|---------|------|------------------------|----------|
| ACC | 0xE0 | 累加器 | 115 |
| ADC0CF | 0xBC | ADC0 配置寄存器 | 49*,65** |
| ADC0CN | 0xE8 | ADC0 控制寄存器 | 50*,66** |
| ADC0GTH | 0xC5 | ADC0 下限 (大于) 数据字 (高字节) | 53*,69** |
| ADC0GTL | 0xC4 | ADC0 下限 (大于) 数据字 (低字节) | 53*,69** |
| ADC0H | 0xBF | ADC0 数据字 (高字节) | 51*,67** |
| ADC0L | 0xBE | ADC0 数据字 (低字节) | 51*,67** |
| ADC0LTH | 0xC7 | ADC0 上限 (小于) 数据字 (高字节) | 53*,69** |
| ADC0LTL | 0xC6 | ADC0 上限 (小于) 数据字 (低字节) | 53*,69** |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 12.3 特殊功能寄存器（续）

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

| 寄存器 | 地址 | 说明 | 页码 |
|---------|------|------------------|----------|
| ADC1CF | 0xAB | ADC1 配置寄存器 | 79 |
| ADC1CN | 0xAA | ADC1 控制寄存器 | 80 |
| ADC1 | 0x9C | ADC1 数据字 | 81 |
| AMX0CF | 0xBA | ADC0 MUX 配置寄存器 | 47*,63** |
| AMX0SL | 0xBB | ADC0 MUX 通道选择寄存器 | 48*,64** |
| AMX1SL | 0xAC | ADC1 MUX 通道选择寄存器 | 79 |
| B | 0xF0 | B 寄存器 | 115 |
| CKCON | 0x8E | 时钟控制寄存器 | 224 |
| CPT0CN | 0x9E | 比较器 0 控制寄存器 | 97 |
| CPT1CN | 0x9F | 比较器 1 控制寄存器 | 98 |
| DAC0CN | 0xD4 | DAC0 控制寄存器 | 86 |
| DAC0H | 0xD3 | DAC0 数据字（高字节） | 85 |
| DAC0L | 0xD2 | DAC0 数据字（低字节） | 85 |
| DAC1CN | 0xD7 | DAC1 控制寄存器 | 88 |
| DAC1H | 0xD6 | DAC1 数据字（高字节） | 87 |
| DAC1L | 0xD5 | DAC1 数据字（低字节） | 87 |
| DPH | 0x83 | 数据指针（高字节） | 113 |
| DPL | 0x82 | 数据指针（低字节） | 113 |
| EIE1 | 0xE6 | 扩展中断允许 1 | 121 |
| EIE2 | 0xE7 | 扩展中断允许 2 | 122 |
| EIP1 | 0xF6 | 扩展中断优先级 1 | 123 |
| EIP2 | 0xF7 | 扩展中断优先级 2 | 124 |
| EMI0CN | 0xAF | 外部存储器接口控制寄存器 | 145 |
| EMI0CF | 0xA3 | 外部存储器接口配置寄存器 | 145 |
| EMI0TC | 0xA1 | 外部存储器接口时序控制寄存器 | 150 |
| FLACL | 0xB7 | FLASH 访问限制 | 140 |
| FLSCL | 0xB6 | FLASH 存储器定时预分频器 | 141 |
| IE | 0xA8 | 中断允许寄存器 | 119 |
| IP | 0xB8 | 中断优先级控制寄存器 | 120 |
| OSICN | 0xB2 | 内部振荡器控制寄存器 | 134 |
| OSCXCN | 0xB1 | 外部振荡器控制寄存器 | 135 |
| P0 | 0x80 | 端口 0 锁存器 | 171 |
| P0MDOUT | 0xA4 | 端口 0 输出方式配置寄存器 | 171 |
| P1 | 0x90 | 端口 1 锁存器 | 172 |
| P1MDIN | 0xBD | 端口 1 输入方式寄存器 | 172 |
| P1MDOUT | 0xA5 | 端口 1 输出方式配置寄存器 | 173 |
| P2 | 0xA0 | 端口 2 锁存器 | 173 |
| P2MDOUT | 0xA6 | 端口 2 输出方式配置寄存器 | 173 |
| P3 | 0xB0 | 端口 3 锁存器 | 174 |
| P3IF | 0xAD | 端口 3 中断标志寄存器 | 175 |
| P3MDOUT | 0xA7 | 端口 3 输出方式配置寄存器 | 174 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 12.3 特殊功能寄存器（续）

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

| 寄存器 | 地址 | 说明 | 页码 |
|----------|------|-------------------|----------|
| †P4 | 0x84 | 端口 4 锁存器 | 178† |
| †P5 | 0x85 | 端口 5 锁存器 | 178† |
| †P6 | 0x86 | 端口 6 锁存器 | 179† |
| †P7 | 0x96 | 端口 7 锁存器 | 179† |
| P74OUT | 0xB5 | 端口 4 - 7 输出方式寄存器 | 177† |
| PCA0CN | 0xD8 | PCA 控制寄存器 | 256 |
| PCA0CPH0 | 0xFA | PCA 捕捉模块 0 高字节 | 260 |
| PCA0CPH1 | 0xFB | PCA 捕捉模块 1 高字节 | 260 |
| PCA0CPH2 | 0xFC | PCA 捕捉模块 2 高字节 | 260 |
| PCA0CPH3 | 0xFD | PCA 捕捉模块 3 高字节 | 260 |
| PCA0CPH4 | 0xFE | PCA 捕捉模块 4 高字节 | 260 |
| PCA0CPL0 | 0xEA | PCA 捕捉模块 0 低字节 | 260 |
| PCA0CPL1 | 0xEB | PCA 捕捉模块 1 低字节 | 260 |
| PCA0CPL2 | 0xEC | PCA 捕捉模块 2 低字节 | 260 |
| PCA0CPL3 | 0xED | PCA 捕捉模块 3 低字节 | 260 |
| PCA0CPL4 | 0xEE | PCA 捕捉模块 4 低字节 | 260 |
| PCA0CPM0 | 0xDA | PCA 模块 0 方式寄存器 | 258 |
| PCA0CPM1 | 0xDB | PCA 模块 1 方式寄存器 | 258 |
| PCA0CPM2 | 0xDC | PCA 模块 2 方式寄存器 | 258 |
| PCA0CPM3 | 0xDD | PCA 模块 3 方式寄存器 | 258 |
| PCA0CPM4 | 0xDE | PCA 模块 4 方式寄存器 | 258 |
| PCA0H | 0xF9 | PCA 计数器高字节 | 259 |
| PCA0L | 0xE9 | PCA 计数器低字节 | 259 |
| PCA0MD | 0xD9 | PCA 方式寄存器 | 257 |
| PCON | 0x87 | 电源控制寄存器 | 126 |
| PSCTL | 0x8F | 程序存储读写控制寄存器 | 142 |
| PSW | 0xD0 | 程序状态字 | 114 |
| RCAP2H | 0xCB | 定时器/计数器 2 捕捉（高字节） | 237 |
| RCAP2L | 0xCA | 定时器/计数器 2 捕捉（低字节） | 237 |
| RCAP4H | 0xE5 | 定时器/计数器 4 捕捉（高字节） | 246 |
| RCAP4L | 0xE4 | 定时器/计数器 4 捕捉（低字节） | 246 |
| REF0CN | 0xD1 | 电压基准控制寄存器 | 92†, 94† |
| RSTSRC | 0xEF | 复位源寄存器 | 131 |
| SADDR0 | 0xA9 | UART0 从地址寄存器 | 212 |
| SADDR1 | 0xF3 | UART1 从地址寄存器 | 222 |
| SADEN0 | 0xB9 | UART0 从地址允许寄存器 | 212 |
| SADEN1 | 0xAE | UART1 从地址允许寄存器 | 222 |
| SBUF0 | 0x99 | UART0 数据缓冲器 | 212 |
| SBUF1 | 0xF2 | UART1 数据缓冲器 | 222 |
| SCON0 | 0x98 | UART0 控制寄存器 | 211 |
| SCON1 | 0xF1 | UART1 控制寄存器 | 221 |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 12.3 特殊功能寄存器（续）

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

| 寄存器 | 地址 | 说明 | 页码 |
|---------------------------------------|------|-----------------|-----|
| SMB0ADR | 0xC3 | SMBus 0 地址寄存器 | 190 |
| SMB0CN | 0xC0 | SMBus 0 控制寄存器 | 188 |
| SMB0CR | 0xCF | SMBus 0 时钟频率寄存器 | 189 |
| SMB0DAT | 0xC2 | SMBus 0 数据寄存器 | 190 |
| SMB0STA | 0xC1 | SMBus 0 状态寄存器 | 191 |
| SP | 0x81 | 堆栈指针 | 113 |
| SPI0CFG | 0x9A | SPI 配置寄存器 | 199 |
| SPI0CKR | 0x9D | SPI 时钟频率寄存器 | 201 |
| SPI0CN | 0xF8 | SPI 总线控制寄存器 | 200 |
| SPI0DAT | 0x9B | SPI 端口 1 数据寄存器 | 201 |
| T2CON | 0xC8 | 定时器/计数器 2 控制寄存器 | 236 |
| T4CON | 0xC8 | 定时器/计数器 4 控制寄存器 | 245 |
| TCON | 0x88 | 定时器/计数器控制寄存器 | 229 |
| TH0 | 0x8C | 定时器/计数器 0 高字节 | 231 |
| TH1 | 0x8D | 定时器/计数器 1 高字节 | 231 |
| TH2 | 0xCD | 定时器/计数器 2 高字节 | 237 |
| TH4 | 0xF5 | 定时器/计数器 4 高字节 | 246 |
| TL0 | 0x8A | 定时器/计数器 0 低字节 | 241 |
| TL1 | 0x8B | 定时器/计数器 1 低字节 | 231 |
| TL2 | 0xCC | 定时器/计数器 2 低字节 | 237 |
| TL4 | 0xF4 | 定时器/计数器 4 低字节 | 246 |
| TMOD | 0x89 | 定时器/计数器方式寄存器 | 230 |
| TMR3CN | 0x91 | 定时器 3 控制寄存器 | 239 |
| TMR3H | 0x95 | 定时器 3 高字节 | 240 |
| TMR3L | 0x94 | 定时器 3 低字节 | 240 |
| TMR3RLH | 0x93 | 定时器 3 重载值高字节 | 240 |
| TMR3RLL | 0x92 | 定时器 3 重载值低字节 | 239 |
| WDTCN | 0xFF | 看门狗定时器控制 | 130 |
| XBR0 | 0xE1 | 端口 I/O 交叉开关控制 0 | 168 |
| XBR1 | 0xE2 | 端口 I/O 交叉开关控制 1 | 169 |
| XBR2 | 0xE3 | 端口 I/O 交叉开关控制 2 | 170 |
| 0x97, 0xA2, 0xB3, 0xB4, 0xCE, 0xDF | | 保留 | |

*寄存器只在 C8051F020/1 中存在。

**寄存器只在 C8051F022/3 中存在。

†寄存器只在 C8051F020/2 中存在。

††寄存器只在 C8051F021/3 中存在。

12.2.7 寄存器说明

下面对与 CIP-51 系统控制器操作有关的 SFR 加以说明。保留位不应被置为逻辑‘1’。将来的产品版本可能会使用这些位实现新功能，在这种情况下各位的复位值将是逻辑‘0’以选择缺省状态。有关其它 SFR 的详细说明见本数据表中与它们对应的系统功能相关的章节。

图 12.3 SP: 堆栈指针

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000111 |
| | | | | | | | | SFR地址: 0x81 |

位 7-0: SP: 堆栈指针
堆栈指针保持栈顶位置。在每次执行 PUSH 操作前，堆栈指针加 1。SP 寄存器复位后的默认值为 0x07。

图 12.4 DPL: 数据指针低字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0x82 |

位 7-0: DPL: 数据指针低字节
DPL 为 16 位数据指针 (DPTR) 的低字节。DPTR 用于访问间接寻址的 XRAM 和 FLASH 存储器。

图 12.5 DPH: 数据指针高字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0x83 |

位 7-0: DPH: 数据指针高字节
DPH 为 16 位数据指针 (DPTR) 的高字节。DPTR 用于访问间接寻址的 XRAM 和 FLASH 存储器。

图 12.6 PSW: 程序状态字

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|-----|-----|-----|-----|-----|-----|-----|--------------|----------------|
| CY | AC | F0 | RS1 | RS0 | OV | F1 | PARITY | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0xD0 |

位 7: CY: 进位标志。
当最后一次算术操作产生进位（加法）或借位（减法）时，该位置 1。其它算术操作将其清 0。

位 6: AC: 辅助进位标志。
当最后一次算术操作向高半字节有进位（加法）或借位（减法）时，该位置 1。其它算术操作将其清 0。

位 5: F0: 用户标志 0。
这是一个可位寻址、受软件控制的通用标志位。

位 4-3: RS1-RS0: 寄存器区选择。
该两位在寄存器访问时用于选择寄存器区。

| RS1 | RS0 | 寄存器区 | 地址 |
|-----|-----|------|-----------|
| 0 | 0 | 0 | 0x00-0x07 |
| 0 | 1 | 1 | 0x08-0x0F |
| 1 | 0 | 2 | 0x10-0x17 |
| 1 | 1 | 3 | 0x18-0x1F |

位 2: OV: 溢出标志。
当最后一次算术操作有进位(加)、借位(减)或溢出（乘或除）时，该位置 1。被其它算术操作清 0。

位 1: F1: 用户标志 1。
这是一个可位寻址、受软件控制的通用标志位。

位 0: PARITY: 奇偶标志。
若累加器中 8 个位的和为奇数时该位置 1，为偶数时清 0。

图 12.7 ACC: 累加器

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|--------------|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0xE0 |

位 7-0: ACC: 累加器
该寄存器为算术操作的累加器。

图 12.8 B: B 寄存器

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|--------------|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0xF0 |

位 7-0: B: B 寄存器
该寄存器为某些算术操作的第二累加器。

12.3 中断系统

CIP-51 包含一个扩展的中断系统，支持 22 个中断源，每个中断源有两个优先级。中断源在片内外设与外部输入引脚之间的分配随器件的不同而变化。每个中断源可以在一个 SFR 中有一个或多个中断标志。当一个外设或外部源满足有效的中断条件时，相应的中断标志被置为逻辑‘1’。

如果中断被允许，在中断标志被置位时将产生中断。一旦当前指令执行完，CPU 产生一个 LCALL 到一个预定地址，开始执行中断服务程序 (ISR)。每个 ISR 必须以 RETI 指令结束，使程序回到中断前执行完的那条指令的下一条指令。如果中断未被允许，中断标志将被硬件忽略，程序继续正常执行。(中断标志置 1 与否不受中断允许/禁止状态的影响。)

每个中断源都可以用一个 SFR (IE-EIE2) 中的相关中断允许位允许或禁止，但是必须首先置‘1’ EA 位 (IE.7) 以保证每个单独的中断允许位有效。不管每个中断允许位的设置如何，清‘0’ EA 位将禁止所有中断。

某些中断标志在 CPU 进入 ISR 时被自动清除。但大多数中断标志不是由硬件清除的，必须在 ISR 返回前用软件清除。如果一个中断标志在 CPU 执行完中断返回 (RETI) 指令后仍然保持置位状态，则会立即产生一个新的中断请求，CPU 将在执行完下一条指令后重新进入 ISR。

12.3.1 MCU 中断源和中断向量

MCU 支持 22 个中断源。软件可以通过将任何一个中断标志设置为逻辑‘1’来模拟一个中断。如果中断标志被允许，系统将产生一个中断请求，CPU 将转向与该中断标志对应的 ISR 地址。表 12.4 给出了 MCU 中断源、对应的向量地址、优先级和控制位一览表。关于外设有效中断条件和中断标志位工作状态方面的详细信息，请见与特定外设相关的章节。

12.3.2 外部中断

两个外部中断源 (/INT0 和 /INT1) 可被配置为低电平触发或下降沿触发输入，由 IT0 (TCON.0) 和 IT1 (TCON.2) 的设置决定。IE0 (TCON.1) 和 IE1 (TCON.3) 分别为外部中断 /INT0 和 /INT1 的中断标志。如果一个 /INT0 或 /INT1 外部中断被配置为边沿触发，CPU 在转向 ISR 时将自动清除相应的中断标志。当被配置为电平触发时，中断标志将跟随外部中断输入引脚的状态，外部中断源必须一直保持输入有效直到中断请求被响应。在 ISR 返回前必须使该中断请求无效，否则将产生另一个中断请求。

其余的 2 个外部中断 (外部中断 6-7) 为边沿触发输入，可以被配置为下降沿触发或上升沿触发。这些中断的中断标志和配置位在端口 3 中断标志寄存器中，见“图 17.19 P3IF: 端口 3 中断标志寄存器”。

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 12.4 中断一览表

| 中断源 | 中断向量 | 优先级 | 中断标志 | 使能 | 优先级控制 |
|-------------------|--------|-----|--|--------------------|--------------------|
| 复位 | 0x0000 | 最高 | 无 | 始终使能 | 总是最高 |
| 外部中断 0 (/INT0) | 0x0003 | 0 | IE0 (TCON.1) | EX0 (IE.0) | PX0 (IP.0) |
| 定时器 0 溢出 | 0x000B | 1 | IE0 (TCON.5) | ET0 (IE.1) | PT0 (IP.1) |
| 外部中断 1 (INT1) | 0x0013 | 2 | IE1 (TCON.3) | EX1 (IE.2) | PX1 (IP.2) |
| 定时器 1 溢出 | 0x001B | 3 | TF1 (TCON.7) | ET1 (IE.3) | PT1 (IP.3) |
| UART0 | 0x0023 | 4 | RI (SCON0.0) TI (SCON0.1) | ES0 (IE.4) | PS0 (IP.4) |
| 定时器 2 溢出 (或 EXF2) | 0x002B | 5 | TF2 (T2CON.7) | ET2 (IE.5) | PT2 (IP.5) |
| 串行外设接口 | 0x0033 | 6 | SPIF (SPI0CN.7) | ESPI0 (EIE1.0) | PSPI0 (EIP1.0) |
| SMBus 接口 | 0x003B | 7 | SI (SMB0CN.3) | ESMB0 (EIE1.1) | PSMB0 (EIP1.1) |
| ADC0 窗口比较 | 0x0043 | 8 | AD0WINT (ADC0CN.2) | EWADC0 (EIE1.2) | PWADC0 (EIP1.2) |
| 可编程计数器阵列 | 0x004B | 9 | CF (PCA0CN.7) CCFn (PCA0CN.n) | EPCA0 (EIE1.3) | PPCA0 (EIP1.3) |
| 比较器 0 下降沿 | 0x0053 | 10 | CP0FIF (CPT0CN.4) | ECP0F (EIE1.4) | PCP0F (EIP1.4) |
| 比较器 0 上升沿 | 0x005B | 11 | CP0RIF (CPT0CN.3) | ECP0R (EIE1.5) | PCP0R (EIP1.5) |
| 比较器 1 下降沿 | 0x0063 | 12 | CP1FIF (CPT1CN.4) | ECP1F (EIE1.6) | PCP1F (EIP1.6) |
| 比较器 1 上升沿 | 0x006B | 13 | CP1RIF (CPT1CN.3) | ECP1R (EIE1.7) | PCP1R (EIP1.7) |
| 定时器 3 溢出 | 0x0073 | 14 | TF3 (TMR3CN.7) | ET3 (EIE2.0) | PT3 (EIP2.0) |
| ADC0 转换结束 | 0x007B | 15 | AD0INT (ADC0CN.5) | EADC0 (EIE2.1) | PADC0 (EIP2.1) |
| 定时器 4 溢出 | 0x0083 | 16 | TF4 (T4CON.7) | ET4 (EIE2.2) | PT4 (EIP2.2) |
| ADC1 转换结束 | 0x008B | 17 | AD1INT (ADC1CN.5) | EADC1 (EIE2.3) | EADC1 (EIE2.3) |
| 外部中断 6 | 0x0093 | 18 | IE6 (PRT3IF.5) | EX6 (EIE2.4) | PX6 (EIP2.4) |
| 外部中断 7 | 0x009B | 19 | IE7 (PRT3IF.6) | EX7 (EIE2.5) | PX7 (EIP2.5) |
| UART1 | 0x00A3 | 20 | RI (SCON1.0) TI (SCON1.1) | ES1(EIE2.6) | PS1(EIP2.6) |
| 外部晶体振荡器准备好 | 0x00AB | 21 | XTLVLD(OSC XCN.7) | EXVLD (EIE2.7) | PXVLD (EIP2.7) |

12.3.3 中断优先权

每个中断源都可以被独立地编程为两个优先级中的一个：低优先级或高优先级。一个低优先级的中断服务程序可以被高优先级的中断所中断，但高优先级的中断不能被中断。每个中断在 SFR (IP-EIP2) 中都有一个配置其优先级的中断优先级设置位，缺省值为低优先级。如果两个中断同时发生，具有高优先级的中断先得到服务。如果这两个中断的优先级相同，则由固定的优先级顺序决定哪一个先得到服务（如表 12.4 所示）。

12.3.4 中断响应时间

中断响应时间取决于中断发生时 CPU 的状态。中断系统在每个系统时钟周期对中断标志采样并对优先级译码。因此最快的响应时间为 5 个系统时钟周期：一个周期用于检测中断，4 个周期完成对 ISR 的长调用 (LCALL)。如果中断标志有效时 CPU 正在执行 RETI 指令，则需要再执行一条指令才能进入中断服务程序。因此，最长的中断响应时间（没有其它中断正被服务或新中断具有较高优先级）发生在 CPU 正在执行 RETI 指令，而下一条指令是 DIV 的情况。在这种情况下，响应时间为 18 个系统时钟周期：1 个时钟周期检测中断，5 个时钟周期执行 RETI，8 时钟周期完成 DIV 指令，4 个时钟周期执行对 ISR 的长调用 (LCALL)。如果 CPU 正在执行一个具有相同或更高优先级的中断的 ISR，则新中断要等到当前 ISR 执行完（包括 RETI 和下一条指令）才能得到服务。

12.3.5 中断寄存器说明

下面介绍用于允许中断源和设置中断优先级的特殊功能寄存器。关于外设有有效中断条件和中断标志位工作状态方面的详细信息，请见与特定片内外设相关的章节。

图 12.9 IE: 中断允许寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|-----|-------|-----|-----|-----|-----|-----|--------------|----------------|
| EA | IEGF0 | ET2 | ES0 | ET1 | EX1 | ET0 | EX0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0xA8 |

位 7: EA: 允许所有中断。
该位允许 / 禁止所有中断。它超越所有的单个中断屏蔽设置。
0: 禁止所有中断源。
1: 开放中断。每个中断由它对应的中断屏蔽设置决定。

位 6: IEGF0: 通用标志位 0。
该位用作软件控制的通用标志位。

位 5: ET2: 定时器 2 中断允许位。
该位用于设置定时器 2 的中断屏蔽。
0: 禁止定时器 2 中断。
1: 允许 TF2 标志位 (T2CON.7) 的中断请求。

位 4: ES0: UART0 中断允许位。
该位设置 UART0 的中断屏蔽。
0: 禁止 UART0 中断。
1: 允许 RI0 标志位 (SCON0.0) 或 TI0 标志位 (SCON0.1) 产生的中断。

位 3: ET1: 定时器 1 中断允许位。
该位用于设置定时器 1 的中断屏蔽。
0: 禁止定时器 1 中断。
1: 允许 TF1 标志位 (TCON.7) 的中断请求。

位 2: EX1: 外部中断 1 允许位。
该位用于设置外部中断 1 的中断屏蔽。
0: 禁止外部中断 1 中断。
1: 允许/INT1 引脚的中断请求

位 1: ET0: 定时器 0 中断允许位。
该位用于设置定时器 0 的中断屏蔽。
0: 禁止定时器 0 中断。
1: 允许 TF0 标志位 (TCON.5) 的中断请求。

位 0: EX0: 外部中断 0 允许位。
该位用于设置外部中断 0 的中断屏蔽。
0: 禁止外部中断 0。
1: 允许/INT0 引脚的中断请求

图 12.10 IP: 中断优先级寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|-----|-----|-----|-----|-----|-----|-----|--------------|----------------|
| - | - | PT2 | PS0 | PT1 | PX1 | PT0 | PX0 | 11000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0xB8 |

位 7-6: 未用。读=11b, 写=忽略。

位 5: PT2: 定时器 2 中断优先级控制
该位设置定时器 2 中断的优先级。
0: 定时器 2 为默认优先级。
1: 定时器 2 为高优先级。

位 4: PS0: UART0 中断优先级控制。
该位设置 UART0 中断的优先级。
0: UART0 为默认优先级。
1: UART1 为高优先级。

位 3: PT1: 定时器 1 中断优先级控制
该位设置定时器 1 中断的优先级。
0: 定时器 1 为默认优先级。
1: 定时器 1 为高优先级。

位 2: PX1: 外部中断 1 优先级控制
该位设置外部中断 1 的优先级。
0: 外部中断 1 为默认优先级。
1: 外部中断 1 为高优先级。

位 1: PT0: 定时器 0 中断优先级控制
该位设置定时器 0 中断的优先级。
0: 定时器 0 为默认优先级。
1: 定时器 0 为高优先级。

位 0: PX0: 外部中断 0 优先级控制
该位设置外部中断 0 的优先级。
0: 外部中断 0 为默认优先级。
1: 外部中断 0 为高优先级。

图 12.11 EIE1: 扩展中断允许 1

| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 | |
|------|--|-------|-------|-------|-------|--------|-------|-------|----------------|
| | ECP1R | ECP1F | ECP0R | ECP0F | EPCA0 | EWADC0 | ESMB0 | ESPIO | 00000000 |
| | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xE6 |
| 位 7: | ECP1R: 允许比较器 1 (CP1) 上升沿中断。 该位设置 CP1 的中断屏蔽。 0: 禁止 CP1 上升沿中断。 1: 允许 CP1RIF 标志位 (CPT1CN.5) 的中断请求。 | | | | | | | | |
| 位 6: | ECP1F: 允许比较器 1 (CP1) 下降沿中断。 该位设置 CP1 的中断屏蔽。 0: 禁止 CP1 下降沿中断。 1: 允许 CP1FIF 标志位 (CPT1CN.4) 的中断请求。 | | | | | | | | |
| 位 5: | ECP0R: 允许比较器 0 (CP0) 上升沿中断。 该位设置 CP0 的中断屏蔽。 0: 禁止 CP0 上升沿中断。 1: 允许 CP0RIF 标志位 (CPT0CN.5) 的中断请求。 | | | | | | | | |
| 位 4: | ECP0F: 允许比较器 0 (CP0) 下降沿中断。 该位设置 CP0 的中断屏蔽。 0: 禁止 CP0 下降沿中断。 1: 允许 CP0FIF 标志位 (CPT0CN.4) 的中断请求。 | | | | | | | | |
| 位 3: | EPCA0: 允许可编程计数器阵列 (PCA0) 中断。 该位设置 PCA0 的中断屏蔽。 0: 禁止所有 PCA0 中断。 1: 允许 PCA0 的中断请求。 | | | | | | | | |
| 位 2: | EWADC0: 允许 ADC0 窗口比较中断 该位设置 ADC0 窗口比较的中断屏蔽。 0: 禁止 ADC0 窗口比较中断。 1: 允许 ADC0 窗口比较中断请求。 | | | | | | | | |
| 位 1: | ESMB0: 允许 SMBus0 中断 该位设置 SMBus0 的中断屏蔽。 0: 禁止 SMBus0 中断。 1: 允许 SI 标志位 (SMB0CN.3) 的中断请求。 | | | | | | | | |
| 位 0: | ESPIO: 允许串行外设接口 0 中断 该位设置 SPI0 的中断屏蔽。 0: 禁止 SPI0 中断 1: 允许 SPIF 标志位 (SPI0CN.7) 的中断请求。 | | | | | | | | |

图 12.12 EIE2: 扩展中断允许 2

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|-----|-----|-----|-------|-----|-------|-----|----------------|
| EXVLD | ES1 | EX7 | EX6 | EADC1 | ET4 | EADC0 | ET3 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xE7 |
| <p>位 7: EXVLD: 允许外部时钟源有效 (XTLVLD) 中断。 该位设置 XTLVLD 的中断屏蔽。 0: 禁止 XTLVLD 中断。 1: 允许 XTLVLD 标志位 (OSCXCN.7) 的中断请求。</p> <p>位 6: ES1: 允许 UART1 中断。 该位设置 UART1 的中断屏蔽。 0: 禁止 UART1 中断。 1: 允许 UART1 中断。</p> <p>位 5: EX7: 允许外部中断 7。 该位设置外部中断 7 的中断屏蔽。 0: 禁止外部中断 7。 1: 允许外部中断 7 输入引脚的中断请求。</p> <p>位 4: EX6: 允许外部中断 6。 该位设置外部中断 6 的中断屏蔽。 0: 禁止外部中断 6。 1: 允许外部中断 6 输入引脚的中断请求。</p> <p>位 3: EADC1: 允许 ADC1 转换结束中断。 该位设置 ADC1 转换结束的中断屏蔽。 0: 禁止 ADC1 转换结束中断。 1: 允许 ADC1 转换结束产生的中断请求。</p> <p>位 2: ET4: 允许定时器 4 中断。 该位设置定时器 4 的中断屏蔽。 0: 禁止定时器 4 中断。 1: 允许 TF4 标志 (T4CON.7) 产生的中断请求。</p> <p>位 1: EADC0: 允许 ADC0 转换结束中断。 该位设置 ADC0 转换结束的中断屏蔽。 0: 禁止 ADC0 转换结束中断。 1: 允许 ADC0 转换结束产生的中断请求。</p> <p>位 0: EX3: 允许定时器 3 中断。 该位设置定时器 3 中断屏蔽。 0: 禁止定时器 3 中断。 1: 允许 TF3 标志位 (TMR3CN.7) 的中断请求。</p> | | | | | | | | |

图 12.13 EIP1: 扩展中断优先级 1

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--|-------|-------|-------|-------|--------|-------|-------|----------------|
| PCP1R | PCP1F | PCP0R | PCP0F | PPCA0 | PWADC0 | PSMB0 | PSPI0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xF6 |
| <p>位 7: PCP1R: 比较器 1 (CP1) 上升沿中断优先级控制 该位设置 CP1 中断的优先级。 0: CP1 上升沿中断为低优先级。 1: CP1 上升沿中断为高优先级。</p> | | | | | | | | |
| <p>位 6: PCP1F: 比较器 1 (CP1) 下降沿中断优先级控制 该位设置 CP1 中断的优先级。 0: CP1 下降沿中断为低优先级。 1: CP1 下降沿中断为高优先级。</p> | | | | | | | | |
| <p>位 5: PCP0R: 比较器 0 (CP0) 上升沿中断优先级控制 该位设置 CP0 中断的优先级。 0: CP0 上升沿中断为低优先级。 1: CP0 上升沿中断为高优先级。</p> | | | | | | | | |
| <p>位 4: PCP0F: 比较器 0 (CP0) 下降沿中断优先级控制 该位设置 CP0 中断的优先级。 0: CP0 下降沿中断设置为低优先级。 1: CP0 下降沿中断设置为高优先级。</p> | | | | | | | | |
| <p>位 3: PPCA0: 可编程计数器阵列 (PCA0) 中断优先级控制 该位设置 PCA0 中断的优先级。 0: PCA0 中断设置为低优先级。 1: PCA0 中断设置为高优先级。</p> | | | | | | | | |
| <p>位 2: PWADC0: ADC0 窗口比较器中断优先级控制 该位设置 ADC0 窗口中断的优先级。 0: ADC0 窗口中断为低优先级。 1: ADC0 窗口中断为高优先级。</p> | | | | | | | | |
| <p>位 1: PSMB0: SMBus0 中断优先级控制 该位设置 SMBus0 中断的优先级。 0: SMBus 中断为低优先级。 1: SMBus 中断为高优先级。</p> | | | | | | | | |
| <p>位 0: PSPI0: 串行外设接口 0 中断优先级控制 该位设置 SPI0 中断的优先级。 0: SPI0 中断为低优先级。 1: SPI0 中断为高优先级。</p> | | | | | | | | |

图 12.14 EIP2: 扩展中断优先级 2

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--|-----|-----|-----|-------|-----|-------|-----|----------------|
| PXVLD | PS1 | PX7 | PX6 | PADC1 | PT4 | PADC0 | PT3 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xF7 |
| <p>位 7: PXVLD: 外部时钟源有效 (XTLVLD) 中断优先级控制 该位设置 XTLVLD 中断的优先级。 0: XTLVLD 中断为低优先级 1: XTLVLD 中断为高优先级</p> <p>位 6: PS1: UART1 中断优先级控制 该位设置 UART1 中断的优先级。 0: UART1 中断为低优先级 1: UART1 中断为高优先级</p> <p>位 5: PX7: 外部中断 7 优先级控制 该位设置外部中断 7 的优先级。 0: 外部中断 7 为低优先级 1: 外部中断 7 为高优先级</p> <p>位 4: PX6: 外部中断 6 优先级控制 该位设置外部中断 6 的优先级。 0: 外部中断 6 设置为低优先级 1: 外部中断 6 设置为高优先级</p> <p>位 3: PADC1: ADC1 转换结束中断优先级控制 该位设置 ADC1 转换结束中断的优先级。 0: ADC1 转换结束中断为低优先级 1: ADC1 转换结束中断为高优先级</p> <p>位 2: PT4: 定时器 4 中断优先级控制 该位设置定时器 4 中断的优先级。 0: 定时器 4 中断设置为低优先级 1: 定时器 4 中断设置为高优先级</p> <p>位 1: PADC0: ADC0 转换结束中断优先级控制 该位设置 ADC 转换结束中断的优先级。 0: ADC 转换结束中断为低优先级 1: ADC 转换结束中断为高优先级</p> <p>位 0: PT3: 定时器 3 中断优先级控制 该位设置定时器 3 中断的优先级。 0: 定时器 3 中断为低优先级 1: 定时器 3 中断为高优先级</p> | | | | | | | | |

12.4 电源管理方式

CIP-51 有两种可软件编程的电源管理方式：等待和停机。在等待方式，CPU 停止运行，而外设和时钟处于活动状态。在停机方式，CPU 停止运行，所有的中断和定时器（时钟丢失检测器除外）都处于非活动状态，系统时钟停止。由于在等待方式下时钟仍然运行，所以功耗与进入等待方式之前的系统时钟频率和处于活动状态的外设数目有关。停机方式消耗最少的功率。图 12.15 对用于控制 CIP-51 电源管理方式的电源控制寄存器作出了说明。

虽然 CIP-51 具有等待和停机方式（与任何标准 8051 结构一样），但最好禁止不需要的外设，以使整个 MCU 的功耗最小。每个模拟外设在不使用时都可以被禁止，使其进入低功耗方式。象定时器、串行总线这样的数字外设在不使用时消耗很少的功率。关闭闪存存储器可以减小功耗，与进入等待方式类似。关闭振荡器可以消耗更少的功率，但需要复位来重新启动 MCU。

12.4.1 等待方式

将等待方式选择位（PCON.0）置 1 导致 CIP-51 停止 CPU 运行并进入等待方式，在执行完对该位置 1 的指令后 MCU 立即进入等待方式。所有内部寄存器和存储器都保持原来的数据不变。所有模拟和数字外设等待方式期间都可以保持活动状态。

有被允许的中断发生或 RST 有效将结束等待方式。当有一个被允许的中断发生时，等待方式选择位（PCON.0）被清 0，CPU 将继续工作。该中断将得到服务，中断返回（RETI）后将开始执行设置等待方式选择位的那条指令的下一条指令。如果等待方式因一个内部或外部复位而结束，则 CIP-51 进行正常的复位过程并从地址 0x0000 开始执行程序。

如果被允许，WDT 将产生一个内部看门狗复位，从而结束等待方式。这一功能可以保护系统不会因为对 PCON 寄存器的意外写入而导致永久性停机。如果不需要这种功能，可以在进入等待方式之前禁止看门狗。这将进一步节省功耗，允许系统一直保持在等待状态，等待一个外部激励唤醒系统。有关使用和配置 WDT 的详细信息，请参见第 16.3 节。

12.4.2 停机方式

将停机方式选择位（PCON.1）置 1 导致 CIP-51 进入停机方式，在执行完对该位置 1 的指令后 MCU 立即进入停机方式。在停机方式，CPU 和振荡器都被停止，实际上所有的数字外设都停止工作。在进入停机方式之前，必须关闭每个模拟外设。只有内部或外部复位能结束停机方式。复位时，CIP-51 进行正常的复位过程并从地址 0x0000 开始执行程序。

如果被允许，时钟丢失检测器将产生一个内部复位，从而结束停机方式。如果想要使 CPU 的休眠时间长于 100 微秒的 MCD 超时时间，则应禁止时钟丢失检测器。

图 12.15 PCON: 电源控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|--------|-----|-------|--------|-----|------|------|----------------|
| SMOD0 | SSTAT0 | 保留 | SMOD1 | SSTAT1 | 保留 | STOP | IDLE | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x87 |
| <p>位 7: SMOD0: UART0 波特率加倍允许。 该位允许/禁止 UART0 波特率逻辑的波特率/2 功能。 0: 允许 UART0 的波特率/2 功能。 1: 禁止 UART0 的波特率/2 功能。</p> <p>位 6: SSTAT0: UART0 增强状态方式选择。 该位控制对 SCON0 寄存器中 SM20-SM00 位的访问方式。 0: 读/写 SM20-SM00 时访问 UART0 方式设置位 SM20-SM00。 1: 读/写 SM20-SM00 时访问帧错误 (FE0)、RX 溢出 (RXOV0) 和 TX 冲突 (TXCOL0) 状态位。</p> <p>位 5: 保留。读出值无定义, 必须写入 0。</p> <p>位 4: SMOD1: UART1 波特率加倍允许。 该位允许/禁止 UART1 波特率逻辑的波特率/2 功能。 0: 允许 UART1 的波特率/2 功能。 1: 禁止 UART1 的波特率/2 功能。</p> <p>位 3: SSTAT1: UART1 增强状态方式选择。 该位控制对 SCON1 寄存器中 SM21-SM01 位的访问方式。 0: 读/写 SM21-SM01 时访问 UART1 方式设置位 SM21-SM01。 1: 读/写 SM21-SM01 时访问帧错误 (FE1)、RX 溢出 (RXOV1) 和 TX 冲突 (TXCOL1) 状态位。</p> <p>位 2: 保留。读出值无定义, 必须写入 0。</p> <p>位 1: STOP: 停机方式选择。 向该位写 '1' 将使 CIP-51 进入停机方式。该位读出值总是为 0。 1: CIP-51 被强制进入掉电方式 (关闭振荡器)</p> <p>位 0: IDLE: 等待方式选择。 向该位写 '1' 将使 CIP-51 进入等待方式。该位读出值总是为 0。 1: CIP-51 被强制进入等待方式。(关闭供给 CPU 的时钟信号, 但定时器、中断和所有外设保持活动状态。)</p> | | | | | | | | |

13. 复位源

复位电路允许很容易地将控制器置于一个预定的缺省状态。在进入复位状态时，将发生以下过程：

- CIP-51 停止程序执行
- 特殊功能寄存器（SFR）被初始化为所定义的复位值
- 外部端口引脚被置于一个已知状态
- 中断和定时器被禁止。

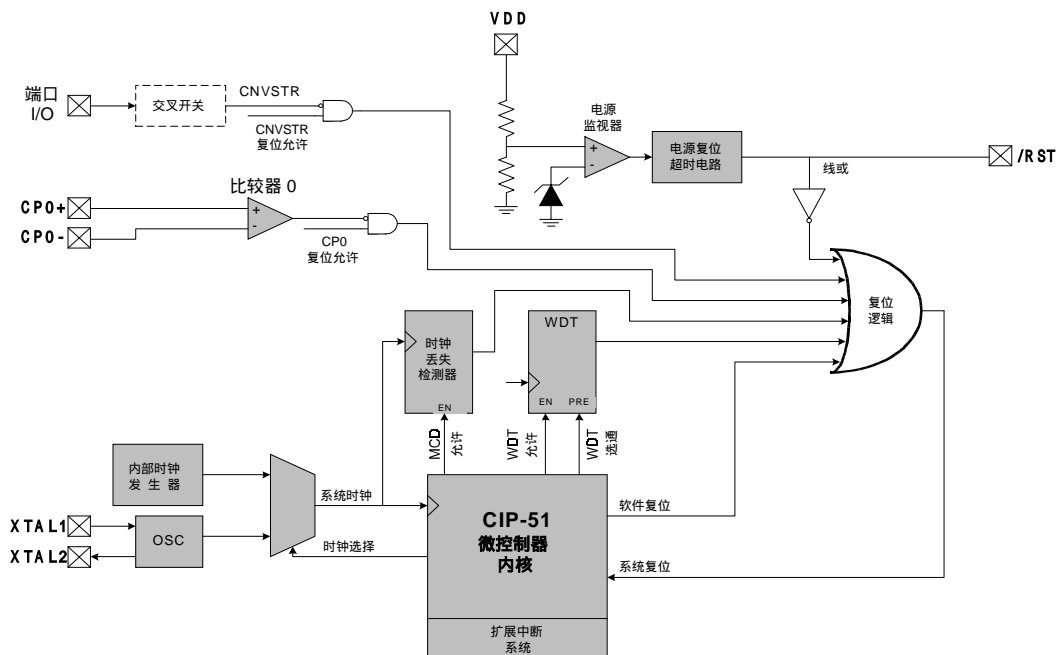
所有的 SFR 都被初始化为预定值，SFR 中各位的复位值在 SFR 的详细说明中定义。在复位期间内部数据存储器的内容不发生改变，复位前存储的数据保持不变。但由于堆栈指针 SFR 被复位，堆栈实际上已丢失，尽管堆栈中的数据未发生变化。

I/O 端口锁存器的复位值为 0xFF（全部为逻辑‘1’），内部弱上拉有效，使外部 I/O 引脚处于高电平状态。外部 I/O 引脚并不立即进入高电平状态，而是在进入复位状态后的四个系统时钟之内。注意：在复位期间弱上拉是被禁止的，在器件退出复位状态时弱上拉被允许。这就使得在器件保持在复位状态期间可以节省功耗。对于 VDD 监视器复位，/RST 引脚被驱动为低电平，直到 VDD 复位超时结束。

在退出复位状态时，程序计数器（PC）被复位，MCU 使用内部振荡器运行在 2MHz 作为默认的系统时钟。有关选择和配置系统时钟源的详细说明见“14 振荡器”。看门狗定时器被允许，使用其最长的超时时间（见“13.8 看门狗定时器复位”）。一旦系统时钟源稳定，程序从地址 0x0000 开始执行。

有 7 个能使 MCU 进入复位状态的复位源：上电/掉电、外部/RST 引脚、外部 CNVSTR 信号、软件命令、比较器 0、时钟丢失检测器及看门狗定时器。下面将对每个复位源进行说明。

图 13.1 复位源框图

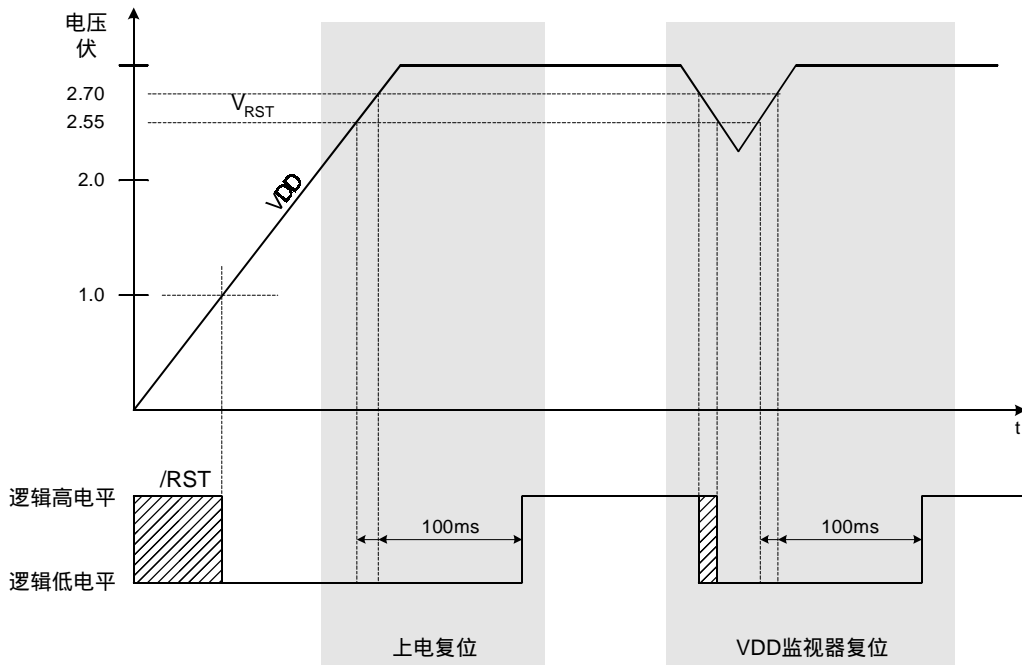


13.1 上电复位

C8051F020/1/2/3 有一个电源监视器，在上电期间该监视器使 MCU 保持在复位状态，直到 VDD 上升到超过 V_{RST} 电平。见图 13.2 的时序图，有关电源监视器电路的电气特性见表 13.1。/RST 引脚一直被置为低电平，直到 100 毫秒的 VDD 监视器超时时间结束，这 100 毫秒的等待时间是为了使 VDD 电源稳定。

在退出上电复位状态时，PORSF 标志 (RSTSRC.1) 被硬件置为逻辑 ‘1’，RSTSRC 寄存器中的其它复位标志是不确定的。PORSF 被任何其它复位源清 0。由于所有的复位都导致程序从同一个地址 (0x0000) 开始执行，软件可以通过读 PORSF 标志来确定是否为上电导致的复位。在一次上电复位后，内部数据存储器的内容应被认为是不确定的。

图 13.2 VDD 监视器时序图



13.2 掉电复位

当发生掉电或因电源不稳定而导致 VDD 下降到低于 V_{RST} 电平时，电源监视器将 /RST 引脚置于低电平并是 CIP-51 回到复位状态。当 VDD 回升到超过 V_{RST} 电平时，CIP-51 将离开复位状态，过程与上电复位相同（见图 13.2）。注意：即使内部数据存储器的内容未因掉电复位而发生变化，也无法确定 VDD 是否下降到维持数据有效所需要的电压以下。如果 PORSF 标志被置 ‘1’，则数据不再有效。

13.3 外部复位

外部 /RST 引脚提供了使用外部电路强制 MCU 进入复位状态的手段。在 /RST 引脚上加一个低电平有效信号将导致 MCU 进入复位状态。最好能提供一个外部上拉和/或对 /RST 引脚去耦以防止强噪声引起复位。在低有效的 /RST 信号撤出后，MCU 将保持在复位状态至少 12 个

时钟周期。从外部复位状态退出后，PINRSF 标志（RSTSRC.0）被置位。

13.4 软件强制复位

如 13.1 节所述，向 PORSF 位写 1 将强制产生一个上电复位。

13.5 时钟丢失检测器复位

时钟丢失检测器实际上是由 MCU 系统时钟触发的单稳态电路。如果未收到系统时钟的时间大于 100 微秒，单稳态电路将超时并产生一个复位。在发生时钟丢失检测器复位后，MCDRSF 标志（RSTSRC.2）将被置‘1’，表示本次复位源为 MSD；否则该位被清‘0’。/RST 引脚的状态不受该复位的影响。把 OSCIN 寄存器（见图 14.2）中的 MSCLKE 位置‘1’将允许时钟丢失检测器。

13.6 比较器 0 复位

向 CORSEF 标志（RSTSRC.5）写‘1’可以将比较器 0 配置为复位源。应在写 CORSEF 之前用 CPT0CN.7（见图 8.3）允许比较器 0，以防止通电瞬间在输出端产生抖动，从而产生不希望的复位。比较器 0 复位是低电平有效，如果同相端输入电压（在 CP0+引脚）小于反相端输入电压（在 CP0-引脚），则 MCU 被置于复位状态。在发生比较器 0 复位之后，CORSEF 标志（RSTSRC.5）被置位，表示本次复位源为比较器 0；否则该位被清‘0’。/RST 引脚的状态不受该复位的影响。

13.7 外部 CNVSTR 引脚复位

向 CNVRSEF 标志（RSTSRC.6）写‘1’可以将外部 CNVSTR 信号配置为复位源。CNVSTR 信号可以出现在 P0、P1、P2 或 P3 的任何 I/O 引脚，见“17.1 端口 0 – 端口 3 和优先权交叉开关译码器”。注意：交叉开关必许被配置为使 CNVSTR 信号接到正确的端口 I/O。应该在将 CNVRSEF 置‘1’之前配置并允许交叉开关。当被配置为复位源时，CNVSTR 为低电平有效。在发生 CNVSTR 复位之后，CNVRSEF 标志（RSTSRC.6）的读出值为‘1’，表示本次复位源为 CNVSTR；否则该位读出值‘0’。/RST 引脚的状态不受该复位的影响。

13.8 看门狗定时器复位

MCU 内部有一个使用系统时钟的可编程看门狗定时器（WDT）。当看门狗定时器溢出时，WDT 将强制 CPU 进入复位状态。为了防止复位，必须在溢出发生前由应用软件重新触发 WDT。如果系统出现了软件/硬件错误，使应用软件不能重新触发 WDT，则 WDT 将溢出并产生一个复位，这可以防止系统失控。

在从任何一种复位退出时，WDT 被自动允许并使用缺省的最大时间间隔运行。系统软件可以根据需要禁止 WDT 或将其锁定为运行状态以防止意外产生的禁止操作。WDT 一旦被锁定，在下次系统复位之前将不能被禁止。/RST 引脚的状态不受该复位的影响。

WDT 是一个 21 位的使用系统时钟的定时器。该定时器测量对其控制寄存器的两次特定写操作的时间间隔。如果这个时间间隔超过了编程的极限值，将产生一个 WDT 复位。可以根据需要用软件允许和禁止 WDT，或根据要求将其设置为永久性允许状态。看门狗的功能可以通过看门狗定时器控制寄存器（WDTCN）控制，见图 13.3。

13.8.1 允许/复位 WDT

向 WDTCN 寄存器写入 0xA5 将允许并复位看门狗定时器。用户的应用软件应周期性地向 WDTCN 写入 0xA5，以防止看门狗定时器溢出。每次系统复位都将允许并复位 WDT。

13.8.2 禁止 WDT

向 WDTCN 寄存器写入 0xDE 后再写入 0xAD 将禁止 WDT。下面的代码段说明禁止 WDT 的过程。

```

CLR    EA                ; 禁止所有中断
MOV    WDTCN, #0DEh    ; 禁止软件看门狗定时器
MOV    WDTCN, #0Adh
SETB   EA                ; 重新允许中断
    
```

写 0xDE 和写 0xAD 必须发生在 4 个时钟周期之内，否则禁止操作将被忽略。在这个过程中应禁止中断，以避免两次写操作之间有延时。

13.8.3 禁止 WDT 锁定

向 WDTCN 写入 0xFF 将使禁止功能无效。一旦锁定，在下次复位之前禁止操作将被忽略。写 0xFF 并不允许或复位看门狗定时器。如果应用程序想一直使用看门狗，则应在初始化代码中向 WDTCN 写入 0xFF。

13.8.4 设置 WDT 定时间隔

WDTCN.[2:0]控制看门狗超时间隔。超时间隔由下式给出：

$$4^{3+WDTCN[2:0]} \times T_{SYSCLK} ; \text{ (其中 } T_{SYSCLK} \text{ 为系统时钟周期)}$$

对于 2MHz 的系统时钟，超时间隔的范围是 0.032ms 到 524ms。在设置这个超时间隔时，WDTCN.7 必须为 0。读 WDTCN 将返回编程的超时间隔。在系统复位后，WDTCN.[2:0]为 111b。

图 13.3 WDTCN: 看门狗定时器控制寄存器

| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | | |
|--------|--|-----|-----|-----|-----|-----|-----|----|----------------------------------|
| | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 复位值 xxxx111 SFR地址: 0xFF |
| 位 7-0: | WDT 控制 写入 0xA5 将允许并重载 WDT。 写入 0xDE 后四个系统周期内写入 0xAD，将禁止 WDT。 写入 0xFF 锁定禁止功能。 | | | | | | | | |
| 位 4: | 看门狗状态位（读） 读 WDTCN.[4]得到看门狗定时器的状态。 0: WDT 处于不活动状态。 1: WDT 处于活动状态。 | | | | | | | | |
| 位 2-0: | 看门狗超时间隔位 位 WDTCN.[2:0]设置看门狗的超时间隔。在写这些位时，WDTCN.7 必须被置为 '0'。 | | | | | | | | |

图 13.4 RSTSRC: 复位源寄存器

| R | R/W | R/W | R/W | R | R | R/W | R | 复位值 |
|---------|---------|--------|--------|--------|--------|-------|--------|----------------|
| JTAGRST | CNVRSEF | CORSEF | SWRSEF | WDTRSF | MCDRSF | PORSF | PINRSF | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xEF |

(注: 不要对该寄存器进行读/修改/写操作)

位 7: JTAGRST: JTAG 复位标志
 0: JTAG 逻辑不处于复位状态。
 1: JTAG 逻辑处于复位状态。

位 6: CNVRSEF: 转换启动复位源允许和标志。
 写
 0: CNVSTR 不是复位源。
 1: CNVSTR 是复位源 (低电平有效)。
 读:
 0: 前面的复位不是来自 CNVSTR。
 1: 前面的复位来自 CNVSTR。

位 5: CORSEF: 比较器 0 复位允许和标志
 写
 0: 比较器 0 不是复位源。
 1: 比较器 0 是复位源 (低电平有效)。
 读:
 0: 前面的复位不是来自比较器 0。
 1: 前面的复位来自比较器 0。

位 4: SWRSF: 软件强制复位和标志
 写
 0: 无作用
 1: 强制产生一个内部复位。/RST 引脚不受影响。
 读:
 0: 前面的复位不是来自写 SWRSF 位。
 1: 前面的复位来自写 SWRSF 位。

位 3: WDTRSF: 看门狗定时器复位标志
 0: 前面的复位不是来自 WDT 超时。
 1: 前面的复位来自 WDT 超时。

位 2: MCDRSF: 时钟丢失检测器标志
 0: 前面的复位不是来自时钟丢失检测器超时。
 1: 前面的复位来自时钟丢失检测器超时。

位 1: PORSF: 强制上电复位和标志
 写
 0: 无作用
 1: 强制产生一个上电复位。/RST 引脚被驱动为低电平。
 读:
 0: 前面的复位不是来自 POR。
 1: 前面的复位来自 POR。

位 0: PINRSF: 硬件引脚复位标志
 0: 前面的复位不是来自 /RST 引脚。
 1: 前面的复位来自 /RST 引脚。

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 13.1. 复位源电气特性

-40°C到+85°C（除非另有说明）

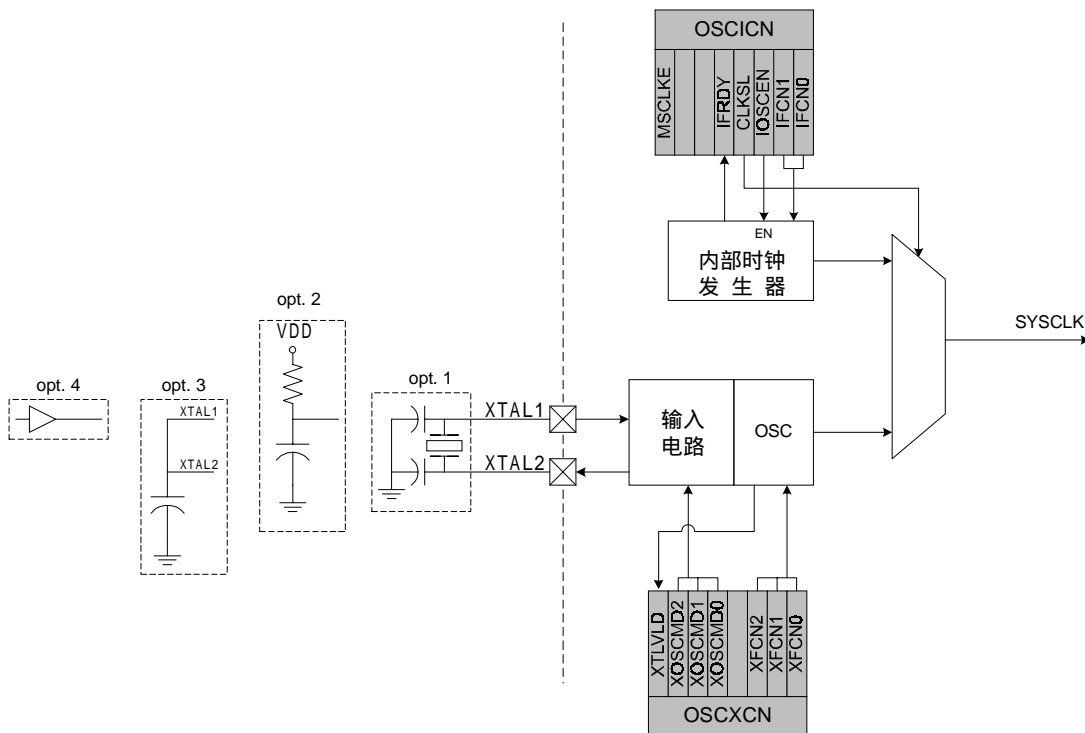
| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|-------------------------|--|----------|------|----------|---------------|
| /RST 输出高电平 | $I_{OH} = -3 \text{ mA}$ | VDD-0.7 | | | V |
| /RST 输出低电平 | $I_{OL} = 8.5 \text{ mA}$, VDD=2.7 到 3.6V | | | 0.6 | V |
| /RST 输入高电平 | | 0.7x VDD | | | V |
| /RST 输入低电平 | | | | 0.3x VDD | V |
| /RST 输入漏电流 | /RST=0.0V | | 50 | | μA |
| /RST 输出有效 VDD | | 1.0 | | | V |
| /RST 输出有效 AV+ | | 1.0 | | | V |
| VDD POR 门限(V_{RST}) | | 2.40 | 2.55 | 2.70 | V |
| 产生复位系统复位的最小/RST 低电平时间 | | 10 | | | ns |
| 复位时间延迟 | 从 VDD 超过复位门限到/RST 的上升沿 | 80 | 100 | 120 | ms |
| 时钟丢失检测器超时 | 从最后一个系统时钟到产生复位 | 100 | 220 | 500 | μs |

14. 振荡器

每个 MCU 都有一个内部振荡器和一个外部振荡器驱动电路，每个驱动电路都能产生系统时钟。MCU 在复位后从内部振荡器启动。内部振荡器可以被允许/禁止，其振荡频率可以用内部振荡器控制寄存器（OSCICN）设置，如图 14.1 所示。表 14.1 给出了内部振荡器的电气特性。

当/RST 引脚为低电平时，两个振荡器都被禁止。MCU 可以从内部振荡器或外部振荡器运行，可使用 OSCICN 寄存器中的 CLKSL 位在两个振荡器之间随意切换。外部振荡器需要一个外部谐振器、并行方式的晶体、电容或 RC 网络连接到 XTAL1/XTAL2 引脚（见图 14.1）。必须在 OSCXCN 寄存器中为这些振荡源中的某一个配置振荡器电路。一个外部 CMOS 时钟也可以通过驱动 XTAL1 引脚提供系统时钟；在这种配置下，XTAL1 引脚用作 CMOS 时钟输入。XTAL1 和 XTAL2 不耐 5V 电压。

图 14.1 振荡器框图



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 14.2 OSCICN: 内部振荡器控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|-----|-----|-------|-------|--------|-------|-------|----------------|
| MSCLKE | - | - | IFRDY | CLKSL | IOSCEN | IFCN1 | IFCN0 | 00010100 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xB2 |
| <p>位 7: MSCLKE: 时钟丢失检测器允许位 0: 禁止时钟丢失检测器。 1: 允许时钟丢失检测器; 检测到时钟丢失时间大于 100 微秒时将触发复位。</p> <p>位 6-5: 未用。读=00b, 写=忽略。</p> <p>位 4: IFRDY: 内部振荡器频率准备好标志 0: 内部振荡器频率不是按 IFCN 位指定的速度运行。 1: 内部振荡器频率按照 IFCN 位指定的速度运行。</p> <p>位 3: CLKSL: 系统时钟源选择位 0: 选择内部振荡器作为系统时钟。 1: 选择外部振荡器作为系统时钟。</p> <p>位 2: IOSCEN: 内部振荡器允许位 0: 内部振荡器禁止。 1: 内部振荡器允许。</p> <p>位 1-0: IFCN1-0: 内部振荡器频率控制位 00: 内部振荡器典型频率为 2MHz。 01: 内部振荡器典型频率为 4MHz。 10: 内部振荡器典型频率为 8MHz。 11: 内部振荡器典型频率为 16MHz。</p> | | | | | | | | |

表 14.1 内部振荡器电气特性

-40°C ~ +85°C (除非另有说明)。

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|------------------|-----------------|------|-----|------|-----|
| 内部振荡器频率 | OSCICN.[1:0]=00 | 1.5 | 2 | 2.4 | MHz |
| | OSCICN.[1:0]=01 | 3.1 | 4 | 4.8 | |
| | OSCICN.[1:0]=10 | 6.3 | 8 | 9.6 | |
| | OSCICN.[1:0]=11 | 12.3 | 16 | 19.2 | |
| 内部振荡器电流消耗(从 VDD) | OSCICN.2=1 | | 200 | | μA |

图 14.3 OSCXCN: 外部振荡器控制寄存器

| R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|--|---------------|-----|-------|-------|-------|----------------|------|-----------------|----------------|---------------|-----|-----------------------|-----------------------|------------|-----|--------------------------------------|--------------------------------------|-----------|-----|--------------------------------------|---------------------------------------|-----------|-----|---------------------------------------|--|----------|-----|--|--|----------|-----|--|--|-----------|-----|--|--|-----------|-----|---------------------|--|------------|
| XTLVLD | XOSCMD2 | XOSCMD1 | XOSCMD0 | - | XFCN2 | XFCN1 | XFCN0 | 00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xB1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>位 7: XTLVLD: 晶体振荡器有效标志 (只在 XOSCMD=1xx 时有效) 0: 晶体振荡器未用或未稳定。 1: 晶体振荡器正在运行并且工作稳定。</p> <p>位 6-4: XOSCMD2-0: 外部振荡器方式位 00x: 关闭。XTAL1 引脚内部接地。 010: 系统时钟为来自 XTAL1 引脚的外部 CMOS 时钟。 011: 系统时钟为来自 XTAL1 引脚的外部 CMOS 时钟的二分频。 10x: RC/C 振荡器方式二分频。 110: 晶体振荡器方式。 111: 晶体振荡器方式二分频。</p> <p>位 3: 保留。读 = 无定义, 写 = 忽略。</p> <p>位 2-0: XFCN2-0: 外部振荡器频率控制位。 000-111: 见下表</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">XFCN</th> <th style="width: 30%;">晶体 (XOSCMD=11x)</th> <th style="width: 30%;">RC(XOSCMD=10x)</th> <th style="width: 30%;">C(XOSCMD=10x)</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">000</td> <td>$f \leq 12\text{kHz}$</td> <td>$f \leq 25\text{kHz}$</td> <td>K 因子= 0.44</td> </tr> <tr> <td style="text-align: center;">001</td> <td>$12\text{kHz} < f \leq 30\text{kHz}$</td> <td>$25\text{kHz} < f \leq 50\text{kHz}$</td> <td>K 因子= 1.4</td> </tr> <tr> <td style="text-align: center;">010</td> <td>$30\text{kHz} < f \leq 95\text{kHz}$</td> <td>$50\text{kHz} < f \leq 100\text{kHz}$</td> <td>K 因子= 4.4</td> </tr> <tr> <td style="text-align: center;">011</td> <td>$95\text{kHz} < f \leq 270\text{kHz}$</td> <td>$100\text{kHz} < f \leq 200\text{kHz}$</td> <td>K 因子= 13</td> </tr> <tr> <td style="text-align: center;">100</td> <td>$270\text{kHz} < f \leq 720\text{kHz}$</td> <td>$200\text{kHz} < f \leq 400\text{kHz}$</td> <td>K 因子= 38</td> </tr> <tr> <td style="text-align: center;">101</td> <td>$720\text{kHz} < f \leq 2.2\text{MHz}$</td> <td>$400\text{kHz} < f \leq 800\text{kHz}$</td> <td>K 因子= 100</td> </tr> <tr> <td style="text-align: center;">110</td> <td>$2.2\text{MHz} < f \leq 6.7\text{MHz}$</td> <td>$800\text{kHz} < f \leq 1.6\text{MHz}$</td> <td>K 因子= 420</td> </tr> <tr> <td style="text-align: center;">111</td> <td>$f > 6.7\text{MHz}$</td> <td>$1.6\text{MHz} < f \leq 3.2\text{MHz}$</td> <td>K 因子= 1400</td> </tr> </tbody> </table> | | | | | | | | | XFCN | 晶体 (XOSCMD=11x) | RC(XOSCMD=10x) | C(XOSCMD=10x) | 000 | $f \leq 12\text{kHz}$ | $f \leq 25\text{kHz}$ | K 因子= 0.44 | 001 | $12\text{kHz} < f \leq 30\text{kHz}$ | $25\text{kHz} < f \leq 50\text{kHz}$ | K 因子= 1.4 | 010 | $30\text{kHz} < f \leq 95\text{kHz}$ | $50\text{kHz} < f \leq 100\text{kHz}$ | K 因子= 4.4 | 011 | $95\text{kHz} < f \leq 270\text{kHz}$ | $100\text{kHz} < f \leq 200\text{kHz}$ | K 因子= 13 | 100 | $270\text{kHz} < f \leq 720\text{kHz}$ | $200\text{kHz} < f \leq 400\text{kHz}$ | K 因子= 38 | 101 | $720\text{kHz} < f \leq 2.2\text{MHz}$ | $400\text{kHz} < f \leq 800\text{kHz}$ | K 因子= 100 | 110 | $2.2\text{MHz} < f \leq 6.7\text{MHz}$ | $800\text{kHz} < f \leq 1.6\text{MHz}$ | K 因子= 420 | 111 | $f > 6.7\text{MHz}$ | $1.6\text{MHz} < f \leq 3.2\text{MHz}$ | K 因子= 1400 |
| XFCN | 晶体 (XOSCMD=11x) | RC(XOSCMD=10x) | C(XOSCMD=10x) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 000 | $f \leq 12\text{kHz}$ | $f \leq 25\text{kHz}$ | K 因子= 0.44 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 001 | $12\text{kHz} < f \leq 30\text{kHz}$ | $25\text{kHz} < f \leq 50\text{kHz}$ | K 因子= 1.4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 010 | $30\text{kHz} < f \leq 95\text{kHz}$ | $50\text{kHz} < f \leq 100\text{kHz}$ | K 因子= 4.4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 011 | $95\text{kHz} < f \leq 270\text{kHz}$ | $100\text{kHz} < f \leq 200\text{kHz}$ | K 因子= 13 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 100 | $270\text{kHz} < f \leq 720\text{kHz}$ | $200\text{kHz} < f \leq 400\text{kHz}$ | K 因子= 38 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 101 | $720\text{kHz} < f \leq 2.2\text{MHz}$ | $400\text{kHz} < f \leq 800\text{kHz}$ | K 因子= 100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 110 | $2.2\text{MHz} < f \leq 6.7\text{MHz}$ | $800\text{kHz} < f \leq 1.6\text{MHz}$ | K 因子= 420 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 111 | $f > 6.7\text{MHz}$ | $1.6\text{MHz} < f \leq 3.2\text{MHz}$ | K 因子= 1400 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>晶体方式 (电路见图 14.1, 选项 1; XOSCMD=11x) 选择 XFCN 值匹配晶体振荡器频率。</p> <p>RC 方式 (电路见图 14.1, 选项 2; XOSCMD=10x) 选择振荡器频率范围: $f=1.23(10^3)/(R*C)$, 其中: f=以 MHz 为单位的振荡频率 C=以 pF 为单位的电容值 R=以 kΩ 为单位的上拉电阻值</p> <p>C 方式 (电路见图 14.1, 选项 3; XOSCMD=10x) 对于所需的振荡频率选择 K 因子 (KF): $f=KF/(C*AV+)$, 其中: f=以 MHz 为单位的振荡频率 C=XTAL1、XTAL2 引脚上的电容值, 以 pF 为单位 AV+=供给 MCU 的模拟电源电压值, 以伏特为单位</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

14.1 外部晶体举例

如果使用晶体或陶瓷谐振器作为 MCU 的外部振荡器源，则电路应为图 14.1 中的选项 1。应从图 14.3 的表中的晶体列选择外部振荡器频率控制值 (XFCN)。例如，一个 11.0592MHz 的晶体要求的 XFCN 值为 111b。

当外部晶体振荡器稳定运行时，晶体振荡器有效标志 (OSCXCN 寄存器中的 XTLVLD) 被硬件置 '1'。XTLVLD 检测电路要求在允许振荡器工作和检测 XTLVLD 之间至少有 1 ms 的启动时间。在外部振荡器稳定之前就切换到外部振荡器可能导致不可预见的后果。建议的过程为：

1. 允许外部振荡器
2. 等待至少 1ms
3. 查询 XTLVLD '0' => '1'
4. 将系统时钟切换到外部振荡器

注：晶体振荡器电路对 PCB 布局非常敏感。应将晶体尽可能地靠近器件的 XTAL 引脚，并在晶体引脚接负载电容。布线应尽可能地短并用地平面屏蔽，防止其它引线引入噪声或干扰。

14.2 外部 RC 举例

如果使用 RC 网络作为 MCU 的外部振荡器源，则电路为图 14.1 中的选项 2。电容不能大于 100pF，但如果使用很小的电容 (小于 ~20pF)，则总电容可能主要由 PWB 的寄生电容决定。为了确定 OSCXCN 寄存器所需要的外部振荡器频率控制值 (XFCN)，首先选择能产生所要求的振荡频率的 RC 网络值。如果所希望的频率是 100kHz，选 $R=246k\Omega$ ， $C=50pF$ ：

$$f = 1.23(10^3)/RC = 1.23(10^3)/[246*50] = 0.1MHz = 100kHz$$

$$XFCN \geq \log_2(f/25kHz)$$

$$XFCN \geq \log_2(100kHz/25kHz) = \log_2(4)$$

$$XFCN \geq 2, \text{ 或代码 } 010b$$

14.3 外部电容举例

如果使用外部电容作为 MCU 的外部振荡器源，则电路为图 14.1 中的选项 3。电容不能大于 100pF，但如果使用很小的电容 (小于 ~20pF)，则总电容将主要由 PWB 的寄生电容决定。为了确定 OSCXCN 寄存器所需要的外部振荡器频率控制值 (XFCN)，选择要用的电容并利用下面的方程计算振荡频率。假设 $VDD = 3.0V$ ， $C=50pF$ ：

$$f = KF / (C * VDD) = KF / (50*3)$$

$$f = KF / 150$$

如果所需要的频率大约为 90kHz，从图 14.3 的表中选择 K 因子，得到 $KF=13$ ：

$$f = 13 / 150 = 0.087MHz, \text{ 或 } 87kHz。$$

因此，本例中要用的 XFCN 值为 011b。

15. FLASH 存储器

C8051F020/1/2/3 系列 MCU 内部有 64k + 128 字节的可编程 FLASH 存储器，用于程序代码和非易失性数据存储。可以通过 JTAG 接口或由软件使用 MOVX 指令对 FLASH 存储器进行在系统编程，每次一个字节。一个 FLASH 位一旦被清 ‘0’，必须经过擦除才能再回到 ‘1’ 状态。在进行重新编程之前，一般要将数据字节擦除（置为 0xFF）。写和擦除操作由硬件自动定时，以保证操作正确，不需进行数据查询来判断写/擦除操作何时结束。FLASH 存储器被设计为能承受至少 20000 个擦/写周期。表 15.1 给出了 FLASH 存储器的电气特性。

15.1 FLASH 存储器编程

对 FLASH 存储器编程的最简单的方法是使用由 Cygnal 或第三方供应商提供的编程工具，通过 JTAG 接口编程。这是对未初始化器件的唯一的编程方法。有关 FLASH 存储器编程的 JTAG 命令方面的详细信息，见“24.2 FLASH 编程命令”。

可以用软件使用 MOVX 指令对 FLASH 存储器编程，象一般的操作数一样为 MOVX 指令提供待编程的地址和数据字节。在使用 MOVX 指令对 FLASH 存储器写入之前，必须将程序存储写允许位 PSWE (PSCTL.0) 设置为逻辑 ‘1’，以允许 FLASH 写操作。这将使 MOVX 指令执行对 FLASH 的写操作而不是对 XRAM 写入。在用软件清除 PSWE 位之前将一直允许写操作。为了避免对 FLASH 的错写，建议在 PSWE 为逻辑 ‘1’ 期间禁止中断。

用 MOVX 指令读 FLASH 存储器；MOVX 读操作将总是指向 XRAM，与 PSWE 的状态无关。

写 FLASH 存储器可以清除数据位，但不能使数据位置 ‘1’；只有擦除操作能将 FLASH 中的数据位置 ‘1’。所以在写入新值之前，必须先擦除待编程的地址。64k 字节的 FLASH 存储器是以 512 字节的扇区为单位组织的。一次擦除操作将擦除整个扇区（将扇区内的所有字节设置为 0xFF）。将程序存储擦除允许位 PSEE (PSCTL.1) 和 PSWE (PSCTL.0) 设置为逻辑 ‘1’ 后，用 MOVX 命令写一个数据字节到扇区内的任何地址将擦除整个 512 字节的扇区。写入的数据字节可以是任何值，因为不是真正写入到 FLASH。在用软件清除 PSEE 位之前将一直允许擦除操作。下面的步骤说明了用软件对 FLASH 编程的过程：

1. 禁止中断。
2. 置位 FLWE (FLSCL.0)，以允许由用户软件写/擦除 FLASH。
3. 置位 PSEE (PSCTL.1)，以允许 FLASH 扇区擦除。
4. 置位 PSWE (PSCTL.0)，以允许 FLASH 写。
5. 用 MOVX 指令向待擦除扇区内的任何一个地址写入一个数据字节。
6. 清除 PSEE 以禁止 FLASH 扇区擦除。
7. 用 MOVX 指令向刚擦除的扇区中所希望的地址写入数据字节。重复该步直到所有字节都已写入（目标扇区内）。
8. 清除 PSWE 以禁止 FLASH 写，使 MOVX 操作指向 XRAM 数据空间。
9. 重新允许中断。

写/擦除时序由硬件自动控制。注意，在 FLASH 正被编程或擦除期间，8051 中的程序停止执行。

表 15.1 FLASH 存储器电特性

VDD=2.7 - 3.6V, -40°C到+85°C (除非另有说明)

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|------|-----|-----|------|-----|-----|
| 擦写寿命 | | 20k | 100k | | 擦/写 |
| 擦除时间 | | 10 | 12 | 14 | ms |
| 写入时间 | | 40 | 50 | 60 | μs |

15.2 非易失性数据存储

FLASH 存储器除了用于存储程序代码之外还可以用于非易失性数据存储。这就允许在程序运行时计算和存储类似标定系数这样的数据。数据写入用 MOVX 指令，读出用 MOVC 指令。

MCU 的 FLASH 存储器中有一个附加的 128 字节的扇区，可用于非易失性数据存储。然而它较小的扇区容量使其尤其适于作为通用的非易失性临时存储器。尽管 FLASH 存储器可以每次写一个字节，但必须首先擦除整个扇区。128 字节的扇区规模使数据更新更加容易，可以不浪费程序存储器或 RAM 空间。该 128 字节的扇区在 64K 字节 FLASH 存储器中是双映射的；它的地址范围是从 0x00 到 0x7F (见图 15.1)。为了访问该扇区，PSCTL 寄存器中的 SFLE 位必须被设置为逻辑 ‘1’。该扇区不能用于存储程序代码。

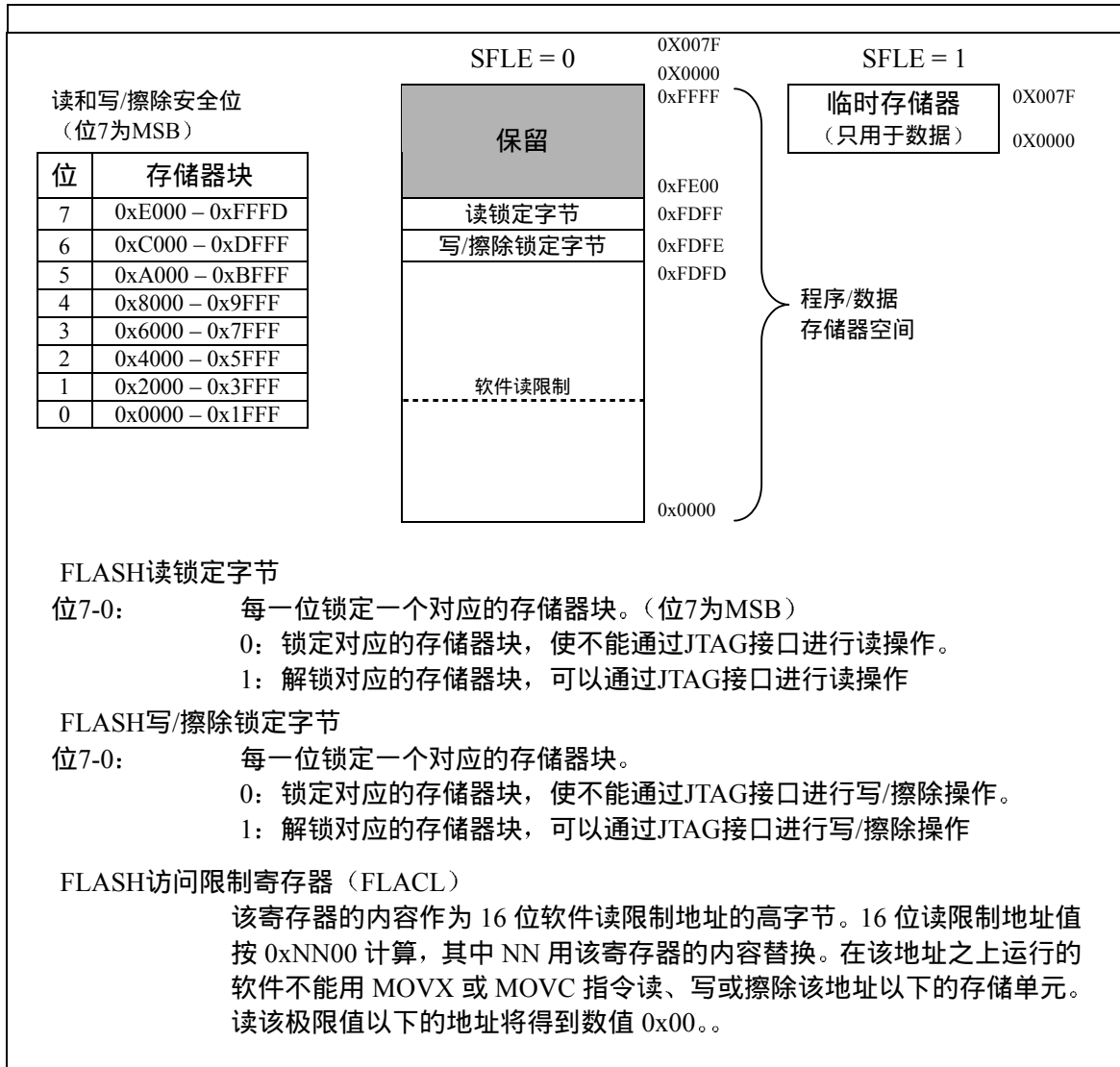
15.3 安全选项

CIP-51 提供了安全选项以保护 FLASH 存储器不会被软件意外修改，以及防止产权程序代码和常数被读取。程序存储写允许 (PSCTL.0) 和程序存储擦除允许 (PSCTL.1) 位保护 FLASH 存储器不会被软件意外修改。在用软件修改 FLASH 存储器的内容之前，这些位必须被置 ‘1’。另外的安全功能是防止通过 JTAG 接口或通过运行在系统控制器上的软件读取产权程序代码和常数。

保存在地址 0xFDFE 和 0xFDFE 中的安全锁定字节集可以保护 FLASH 存储器，使得不能通过 JTAG 接口读取或修改其内容。安全锁定字节中的每一位保护一个 8k 字节的存储器块。将读锁定字节中的一位清 ‘0’ 可防止通过 JTAG 接口读对应的 FLASH 存储器块。将写/擦除锁定字节中的一位清 ‘0’ 可防止通过 JTAG 接口写/擦除对应的存储器块。当所有其它扇区都被锁定后，128 字节的扇区才被锁定。

读锁定字节位于 0xFDFE。写/擦除锁定字节位于 0xFDFE。图 15.2 给出了安全字节的地址和位定义。包含锁定字节的 512 字节扇区可以用软件写入，但不能用软件擦除。对读锁定字节读取将返回无定义的数据。不能通过 JTAG 口调试位于读锁定扇区内的代码。

图 15.1 FLASH 程序存储器组织和安全字节



不管安全字节所在存储块的安全设置如何, 锁定位总是可读的并可以被清 ‘0’。这就允许在锁定了安全字节所在的存储块以后还可以追加要保护的存储块。但是, 一旦设置了锁定位, 解锁的唯一办法是用 JTAG 擦除操作擦除整个程序存储器空间 (即不能由用户固件解锁)。**注意: 解除锁定的唯一方法是通过执行 JTAG 擦除操作将整个程序存储器空间 (即不能由用户固件擦除)。使用任何一个安全字节地址执行 JTAG 擦除操作将自动启动对整个程序存储器空间的擦除 (保留区除外)。该擦除操作只能通过 JTAG 进行。如果擦除操作寻址的是 0xFC00 - 0xFDFD 页内的一个非安全字节, 则只有该页 (包括安全字节) 被擦除。**

FLASH 访问限制这一安全功能 (见图 15.1) 保护产权程序代码和数据不被运行在 C8051F020/1/2/3 MCU 上的软件读取。该功能为那些想在产品发行前在 MCU 中加入增值产权固件的 OEM 生产商提供了支持。这一功能在使增值固件得到保护的同时允许以后在其余的程序存储器中写入代码。

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

软件读限制（SRL）是一个 16 位地址，它将程序存储器空间分成两个逻辑分区。第一个是上分区，包括 SRL 地址之上（含该地址）的所有程序存储器地址；第二个是下分区，包括从 0x0000 到（但不包括）SRL 的所有程序存储器地址。位于上分区的软件可以执行下分区的代码，但不能用 MOVC 指令读下分区中的内容。（使用位于下分区的源地址从上分区执行 MOVC 指令将总是读到 0x00。）运行在下分区中的软件可以不受限制地访问上分区和下分区。

增值固件应存放在下分区。复位时通过复位向量将控制转到增值固件。一旦增值固件完成初始化操作，程序将转到上分区中的预定位置。如果程序入口是公开的，运行在上分区中的软件就可以执行下分区中的代码，但不能读下分区中的内容。有两种方法向下分区中的程序代码传递参数：一种是通常使用的方法，即调用前将参数放在堆栈或寄存器中；另一种是将参数放在上分区中的指定位置。

SRL 地址由 FLASH 访问寄存器中的内容指定。16 位的 SRL 地址值按 0xNN00 计算，其中 NN 用 SRL 安全寄存器的内容。因此，SRL 可位于程序存储器空间中以 256 字节为界的任何位置。然而 512 字节的擦除扇区规模要求以 512 字节为界。未初始化过的 SRL 安全字节的内容是 0x00，因此所设置的 SRL 地址为 0x0000，在这种缺省情况下允许读取全部程序存储器空间。

图 15.2 FLACL: FLASH 访问限制

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| | | | | | | | | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xB7 |

位 7-0: FLACL: FLASH 访问限制。
该寄存器保持 16 位程序存储器读 / 写 / 擦除限制地址的高字节。完整的 16 位访问限制地址按 0xNN00 计算，其中 NN 用 FLACL 的内容替换。向该地址写将设置 FLASH 访问限制地址。在任何一次复位后只能向该寄存器写入一次。在下一次复位之前任何后续的写操作都将被忽略。

图 15.3 FLSC: FLASH 存储器控制

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|------|------|-----|-----|-----|-----|-----|------|----------------|
| FOSE | FRAE | 保留 | 保留 | 保留 | 保留 | 保留 | FLWE | 10000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xB6 |

位 7: FOSE: FLASH 单稳态定时器允许。
 0: 禁止单稳态定时器
 1: 允许单稳态定时器

位 6: FRAE: FLASH 一直读允许。
 0: 在单稳态定时器间隔读一次 FLASH。
 1: FLASH 总是处于读方式。

位 5-1: 保留。读 = 0000b, 写入值必须是 0000b。

位 0: FLWE: FLASH 读/写允许。
 该位必须置 ‘1’ 才能从用户软件写 FLASH。
 0: 禁止 FLASH 写。
 1: 允许 FLASH 写。

图 15.4 PSCTL: 程序存储读写控制

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|-----|-----|-----|-----|-----|------|------|------|----------------|
| - | - | - | - | - | SFLE | PSEE | PSWE | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x8F |

位 7-3: 未使用。读 = 00000b, 写 = 忽略。

位 2: SFLE: 临时 FLASH 存储器访问允许。
 当该位被置 1 时, 用户软件对 FLASH 的读/写操作将指向 128 字节的 FLASH 临时存储扇区。当该位被设置为逻辑 ‘1’ 时, 不应访问 0x00 – 0x7F 以外的地址范围。对该地址范围以外的地址进行读/写可能产生不可预料的结果。
 0: 从用户软件访问 FLASH 时将访问 64K 字节的程序/数据 FLASH 扇区。
 1: 从用户软件访问 FLASH 时将访问 128 字节的临时存储器扇区。

位 1: PSEE: 程序存储擦除允许。
 将该位置 1 将允许擦除 FLASH 存储器中的一个页(前提是 PSWE 位也被置 1)。在将该位置 1 后, 用 MOVX 指令进行一次写操作将擦除包含 MOVX 指令寻址地址的那个 FLASH 页。用于写操作的数据可以是任意值。
 0: 禁止擦除 FLASH 存储器
 1: 允许擦除 FLASH 存储器

位 0: PSWE: 程序存储写允许。
 将该位置 1 后允许用 MOVX 指令向 FLASH 存储器写一个字节。在写数据之前必须先进行擦除。
 0: 禁止写 FLASH 存储器
 1: 允许写 FLASH 存储器

16. 外部数据存储器接口和片内 XRAM

C8051F020/1/1/3 MCU 内部有位于外部数据存储器空间的 4096 字节 RAM (XRAM)，还有外部数据存储器接口 (EMIF) 可用于访问片外存储器和存储器映射的 I/O 器件。外部存储器空间可以用外部传送指令 (MOVX) 和数据指针 (DPTR) 访问，或者通过使用 R0 或 R1 用间接寻址方式访问。如果 MOVX 指令使用一个 8 位地址操作数 (例如 @R1)，则 16 位地址的高字节由外部存储器接口控制寄存器 (EMI0CN，如图 16.1 所示) 提供。**注意：MOVX 指令还用于写 FLASH 存储器，详见“15. FLASH 存储器”。缺省情况下 MOVX 指令访问 XRAM。**EMIF 可被配置为使用低 I/O 端口 (P0-P3) 或高 I/O 端口 (P4-P7)。

16.1 访问 XRAM

XRAM 存储器空间用 MOVX 指令访问。MOVX 指令有两种形式，这两种形式都使用间接寻址方式。第一种方法使用数据指针 DPTR，该寄存器中含有待读或写的 XRAM 单元的有效地址。第二种方法使用 R0 或 R1，与 EMI0CN 寄存器一起形成有效 XRAM 地址。下面举例说明这两种方法。

16.1.1 16 位 MOVX 示例

16 位形式的 MOVX 指令访问由 DPTR 寄存器的内容所指向的存储器单元。下面的指令将地址 0x1234 的内容读入累加器 A：

```
MOV   DPTR, #1234h      ; 将待读单元的 16 位地址 (0x1234) 装入 DPTR
MOVX  A, @DPTR         ; 将地址 0x1234 的内容装入累加器 A
```

上面的例子使用 16 位立即数 MOV 指令设置 DPTR 的内容。还可以通过访问特殊功能寄存器 DPH (DPTR 的高 8 位) 和 DPL (DPTR 的低 8 位) 来改变 DPTR 的内容。

16.1.2 8 位 MOVX 示例

8 位形式的 MOVX 指令使用特殊功能寄存器 EMI0CN 的内容给出待访问地址的高 8 位，由 R0 或 R1 的内容给出待访问地址的低 8 位。下面的指令将地址 0x1234 的内容读入累加器 A：

```
MOV   EMI0CN, #12h     ; 将地址的高字节装入 EMI0CN
MOV   R0, #34h        ; 将地址的低字节装入 R0 (或 R1)
MOVX  A, @DPTR       ; 将地址 0x1234 的内容装入累加器 A
```

16.2 配置外部存储器接口

配置外部存储器接口的过程包括下面 4 个步骤：

1. 将 EMIF 选到低端口（P3、P2、P1 和 P0）或选到高端口（P7、P6、P5 和 P4）。
2. 选择复用方式或非复用方式。
3. 选择存储器模式（只用片内存储器、不带块选择的分片方式、带块选择的分片方式或只用片外存储器）。
4. 设置与片外存储器或外设接口的时序。
5. 选择所需要的相关端口的输出方式（寄存器 PnMDOUT 和 P74OUT）。

下面将对上述每个步骤作出详细说明。端口选择、复用方式选择和存储器模式位都位于 EMIOCN 寄存器中，如图 16.2 所示。

16.3 端口选择和配置

外部存储器接口可以位于端口 3、2、1 和 0（C8051F020/1/2/3）或端口 7、6、5 和 4（只限于 C8051F020/2），由 PRTSEL 位（EMIOCF.5）的状态决定。如果选择低端口，则 EMIFLE 位（XBR2.1）必须被置‘1’，以使交叉开关跳过 P0.7(W/R)、P0.6(R/D)和 P0.5(ALE，如果选择复用方式)。有关配置交叉开关的详细信息见“17. 端口输入/输出”。

外部存储器接口只在执行片外 MOVX 指令期间使用相关的端口引脚。一旦 MOVX 指令执行完毕，端口锁存器或交叉开关又重新恢复对端口引脚的控制（端口 3、2、1 和 0），有关交叉开关和端口操作及控制的详细信息见“17. 端口输入/输出”。**端口锁存器应被明确地配置为使外部存储器接口引脚处于休眠状态，通常是通过将它们设置为逻辑‘1’来实现这一要求。**

在执行 MOVX 指令期间，外部存储器接口将禁止所有作为输入的那些引脚的驱动器（例如，读操作期间的 Data[7:0]）。端口引脚的输出方式（无论引脚被配置为漏极开路或是推挽方式）不受外部存储器接口操作的影响，始终受 PnMDOUT 寄存器的控制。有关端口输出方式配置的详细信息见“17. 端口输入/输出”。

图 16.1 EMI0CN: 外部存储器接口控制

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--------|--------|--------|--------|--------|--------|--------|--------|----------------|
| PGSEL7 | PGSEL6 | PGSEL5 | PGSEL4 | PGSEL3 | PGSEL2 | PGSEL1 | PGSEL0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xAF |

位 7-0: PGSEL[7:0]: XRAM 页选择位
 当使用 8 位的 MOVX 命令时, XRAM 页选择位提供 16 位外部数据存储器地址的高字节, 实际上是选择一个 256 字节的 RAM 页。
 0x00: 0x0000 – 0x00FF
 0x01: 0x0100 – 0x01FF
 ...
 0xFE: 0xFE00 – 0xFEFF
 0xFF: 0xFF00 – 0xFFFF

图 16.2 EMI0CF: 外部存储器接口配置

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|-----|-----|--------|------|------|------|-------|-------|----------------|
| - | - | PRTSEL | EMD2 | EMD1 | EMD0 | EALE1 | EALE0 | 00000011 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xA3 |

位 7-6: 未用。读 = 00b, 写 = 忽略。
 位 5: PRTSEL: EMIF 端口选择位
 0: EMIF 在 P0-P3。
 1: EMIF 在 P4-P7。
 位 4: EMD2: EMIF 复用方式选择位
 0: EMIF 工作在地址/数据复用方式。
 1: EMIF 工作在非复用方式 (分离的地址和数据引脚)。
 位 3-2: EMD1-0: EMIF 工作模式选择位
 这两位控制外部存储器接口的工作模式
 00: 只用内部存储器: MOVX 只寻址片内 XRAM。所有有效地址都指向片内存储器空间
 01: 不带块选择的分片方式。寻址低于 4K 边界的地址时访问片内存储器, 寻址高于 4K 边界的地址时访问片外存储器。8 位片外 MOVX 操作使用地址高端口锁存器的当前内容作为地址的高字节。注意: 为了能访问片外存储器空间, EMI0CN 必须被设置成一个不属于片内地址空间的页地址。
 10: 带块选择的分片方式。寻址低于 4K 边界的地址时访问片内存储器, 寻址高于 4K 边界的地址时访问片外存储器。8 位片外 MOVX 操作使用 EMI0CN 的内容作为地址的高字节。
 11: 只用外部存储器: MOVX 只寻址片外 XRAM。片内 XRAM 对 CPU 为不可见。
 位 1-0: EALE1-0: ALE 脉冲宽度选择位 (只在 EMD2=0 时有效)
 00: ALE 高和 ALE 低脉冲宽度 = 1 个 SYSCLK 周期。
 01: ALE 高和 ALE 低脉冲宽度 = 2 个 SYSCLK 周期。
 10: ALE 高和 ALE 低脉冲宽度 = 3 个 SYSCLK 周期。
 11: ALE 高和 ALE 低脉冲宽度 = 4 个 SYSCLK 周期。

16.4 复用和非复用选择

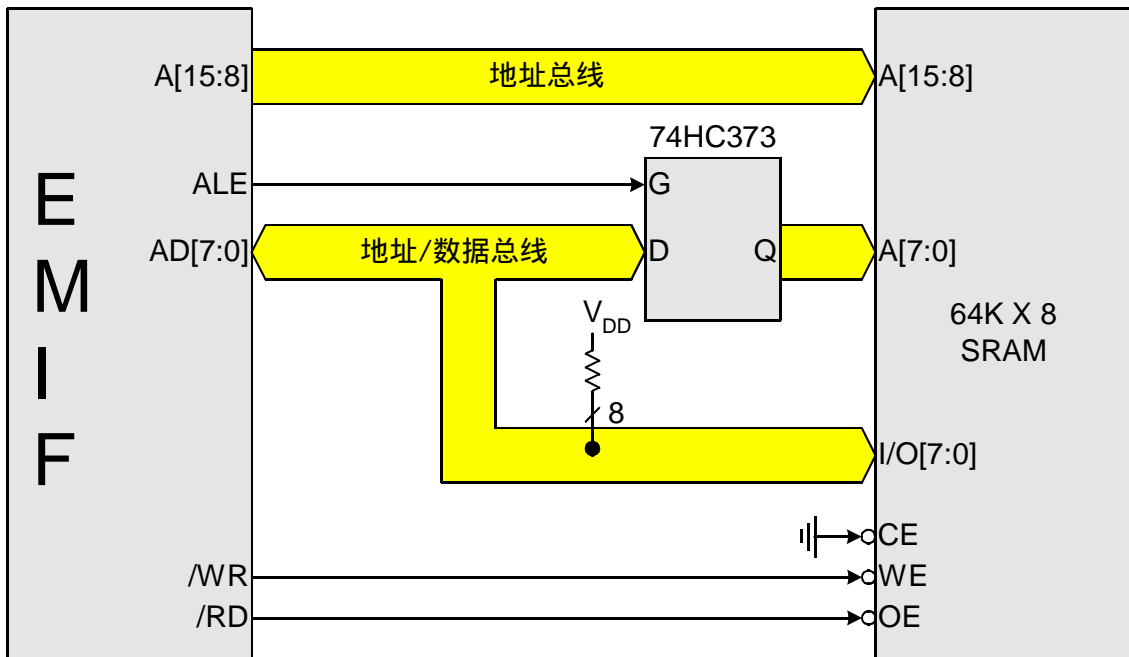
外部存储器接口可以工作在复用方式或非复用方式，由 EMD2 位（EMI0CF.4）的状态决定。

16.4.1 复用方式配置

在复用方式，数据总线和地址总线的低 8 位共享相同的端口引脚：AD[7:0]。在该方式下，要用一个外部锁存器（74HC373 或相同功能的逻辑门）保持 RAM 地址的低 8 位。外部锁存器由 ALE（地址锁存使能）信号控制，ALE 信号由外部存储器接口逻辑驱动。图 16.3 给出了复用方式配置的一个例子。

在复用方式，可以根据 ALE 信号的状态将外部 MOVX 操作分成两个阶段。在第一个阶段，ALE 为高电平，地址总线的低 8 位出现在 AD[7:0]。在该阶段，地址锁存器的‘Q’输出与‘D’输入的状态相同。ALE 由高变低时标志第二阶段开始，地址锁存器的输出保持不变，即与锁存器的输入无关。在第二阶段后期，当 /RD 或 /WR 有效时，数据总线控制 AD[7:0]端口的状态。更详细的信息见“16.6.2 复用方式”。

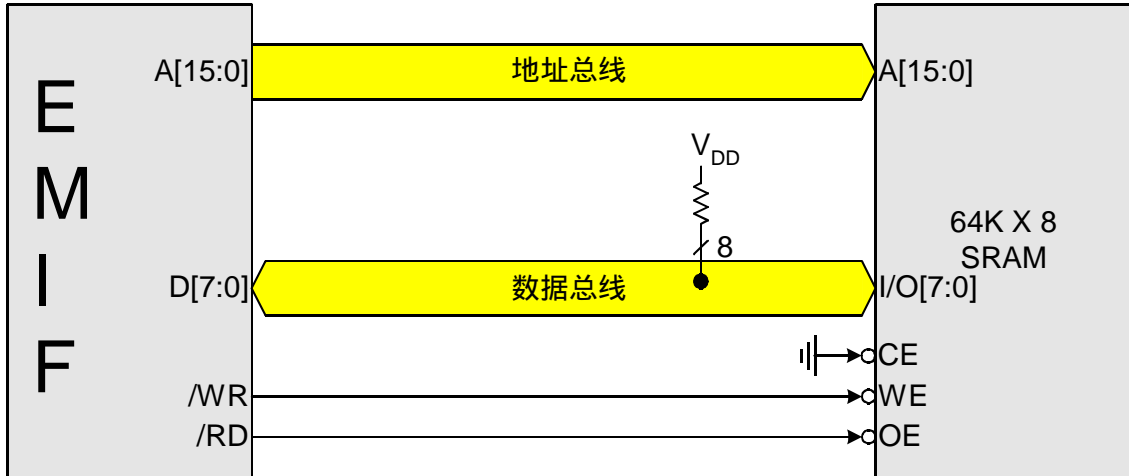
图 16.3 复用方式配置示例



16.4.2 非复用方式配置

在非复用方式，数据总线和地址总线是分开的。图 16.4 给出了非复用方式配置的一个例子。关于非复用方式操作的更详细的信息见“16.6.1 非复用方式”。

图 16.4 非复用方式配置示例



16.5 存储器模式选择

可以用 EMI0CF 寄存器（图 16.2）中 EMIF 模式选择位将外部数据存储器空间配置为四种工作模式之一。下面简要介绍这些模式。有关不同模式的更详细的信息见“16.6 时序”。

16.5.1 只用内部 XRAM

当 EMI0CF.[3:2]被设置为‘00’时，所有 MOVX 指令都将访问器件内部的 XRAM 空间。存储器寻址的地址大于实际地址空间时将以 4K 为边界回绕。例如：地址 0x1000 和 0x2000 都指向片内 XRAM 空间的 0x0000 地址。

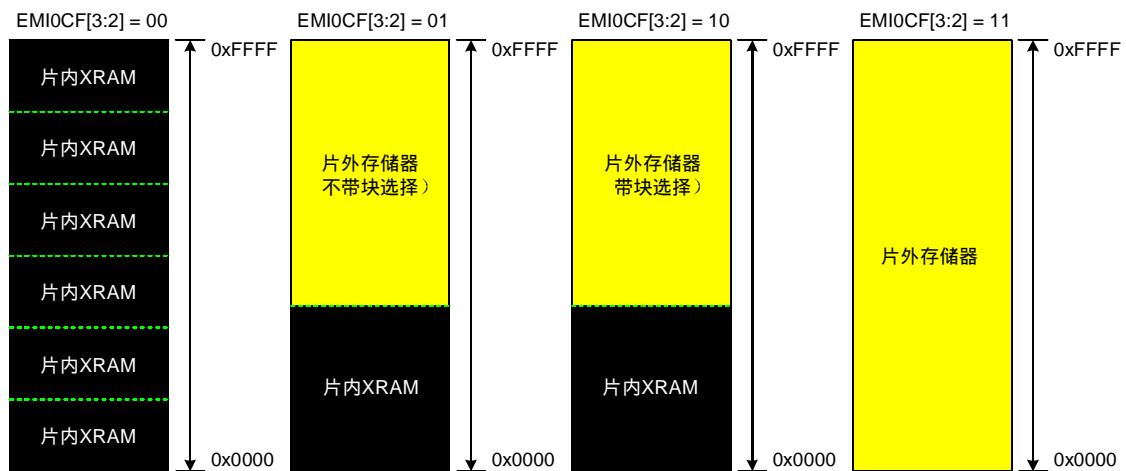
- 8 位 MOVX 操作使用特殊功能寄存器 EMI0CN 的内容作为有效地址的高字节，由 R0 或 R1 给出待有效地址的低字节。
- 16 位 MOVX 操作使用 16 位寄存器 DPTR 的内容作为有效地址。

16.5.2 无块选择的分片模式

当 EMI0CF.[3:2]被设置为‘01’时，XRAM 存储器空间被分成两个区域（片），即片内空间和片外空间。

- 有效地址低于 4K 将访问片内 XRAM 空间。
- 有效地址高于 4K 将访问片外 XRAM 空间。
- 8 位 MOVX 操作使用特殊功能寄存器 EMI0CN 的内容确定是访问片内还是片外存储器，地址总线的低 8 位 A[7:0]由 R0 或 R1 给出。然而对于“无块选择”模式，在访问片外存储器期间一个 8 位 MOVX 操作不驱动地址总线的高 8 位 A[15:8]。这就允许用户通过直接设置端口的状态来按自己的意愿操作高位地址。下面将要描述的“带块选择的分片模式”则与此相反。
- 16 位 MOVX 操作使用 DPTR 的内容确定是访问片内还是片外存储器，与 8 位 MOVX 操作不同的是，在访问片外存储器时地址总线 A[15:0]的全部 16 位都被驱动。

图 16.5 EMIF 工作模式



16.5.3 带块选择的分片模式

当 EMI0CF.[3:2]被设置为‘10’时，XRAM 存储器空间被分成两个区域（片），即片内空间和片外空间。

- 有效地址低于 4K 将访问片内 XRAM 空间。
- 有效地址高于 4K 将访问片外 XRAM 空间。
- 8 位 MOVX 操作使用特殊功能寄存器 EMI0CN 的内容确定是访问片内还是片外存储器，地址总线的高 8 位 A[15:8]由 EMI0CN 给出，而地址总线的低 8 位 A[7:0]由 R0 或 R1 给出。在“块选择”模式，地址总线 A[15:0]的全部 16 位都被驱动。
- 16 位 MOVX 操作使用 DPTR 的内容确定是访问片内还是片外存储器，与 8 位 MOVX 操作不同的是，在访问片外存储器时地址总线 A[15:0]的全部 16 位都被驱动。

16.5.4 只用外部存储器

当 EMI0CF.[3:2]被设置为‘11’时，所有 MOVX 指令都将访问器件外部 XRAM 空间。片内 XRAM 对 CPU 为不可见。该方式在访问从 0x0000 开始的 4K 片外存储器时有用。

- 8 位 MOVX 操作忽略 EMI0CN 的内容。高地址位 A[15:8]不被驱动（与“不带块选择的分片模式”中描述的访问片外存储器的行为相同）。这就允许用户通过直接设置端口的状态来按自己的意愿操作高位地址。有效地址的低 8 位 A[7:0]由 R0 或 R1 给出。
- 16 位 MOVX 操作使用 DPTR 的内容确定有效地址 A[15:0]。在访问片外存储器时地址总线 A[15:0]的全部 16 位都被驱动。

16.6 时序

外部存储器接口的时序参数是可编程的，这就允许连接具有不同建立时间和保持时间要求的器件。地址建立时间、地址保持时间、/RD 和 /WR 选通脉冲宽度以及复用方式下的 ALE 脉冲宽度都可以通过 EM0TC（见图 16.6）和 EMI0CF[1:0]编程，编程单位为系统时钟周期。

在非复用方式，一次片外 XRAM 操作的最小执行时间为 4 个 SYSCLK 周期：3 个周期用于 MOVX 操作，1 个周期用于 /RD 或 /WR 持续时间。在器件复位后，可编程建立和保持时间的缺省值为最大延迟设置。

表 16.1 列出了外部存储器接口的 AC 参数，图 16.7 到图 16.11 给出了对应不同外部存储器接口模式和 MOVX 操作的时序图。

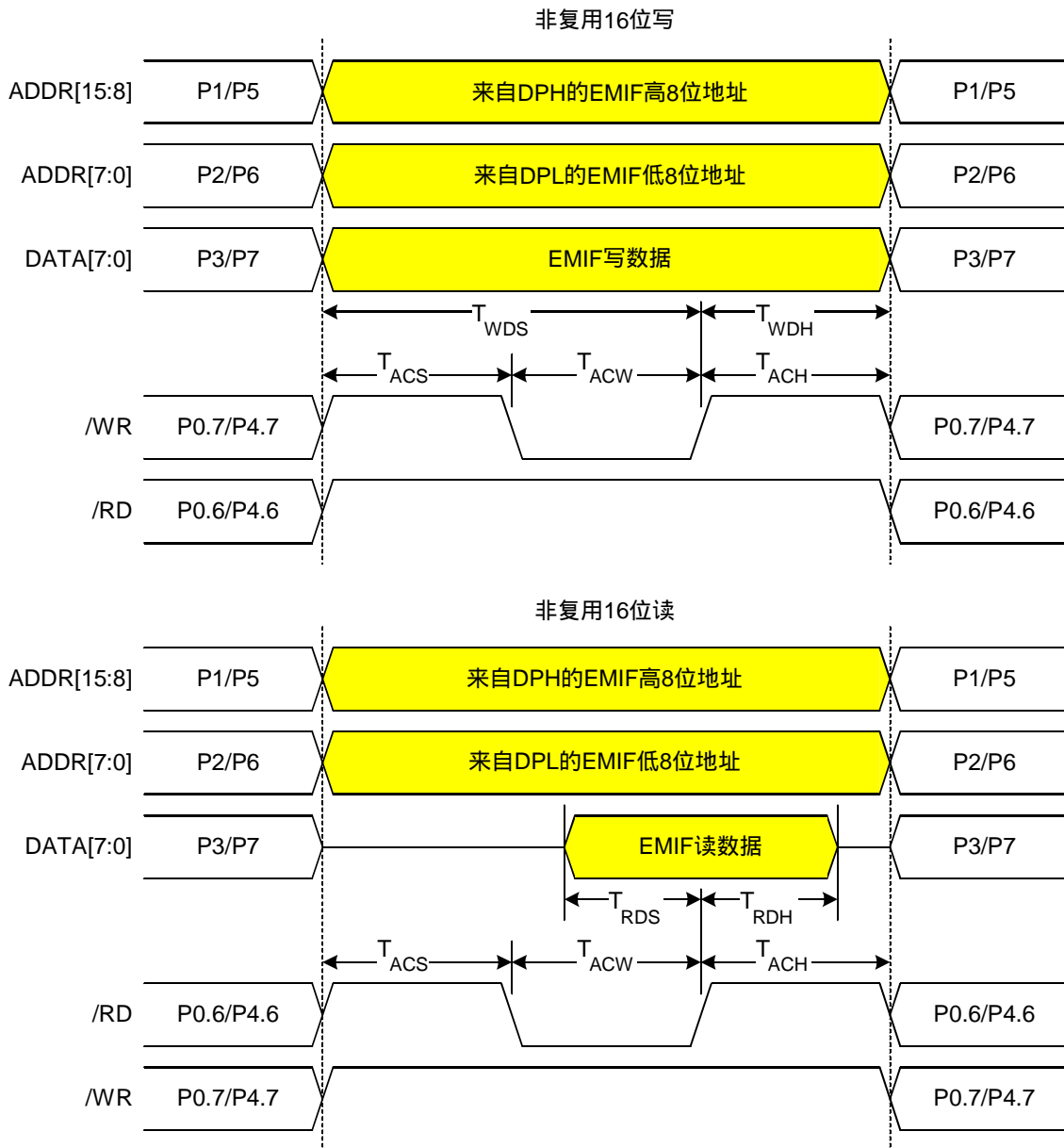
图 16.6 EMI0TC: 外部存储器时序控制

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|------|------|------|------|------|------|------|----------------|
| EAS1 | EAS0 | EWR3 | EWR2 | EWR1 | EWR0 | EAH1 | EAH0 | 11111111 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xA1 |
| <p>位 7-6: EAS1-0: EMIF 地址建立时间位。</p> <p>00: 地址建立时间 = 0 个 SYSCLK 周期。</p> <p>01: 地址建立时间 = 1 个 SYSCLK 周期。</p> <p>10: 地址建立时间 = 2 个 SYSCLK 周期。</p> <p>11: 地址建立时间 = 3 个 SYSCLK 周期。</p> <p>位 5-2: EWR3-0: EMIF /WR 和 /RD 脉冲宽度控制位。</p> <p>0000: /WR 和 /RD 脉冲宽度 = 1 个 SYSCLK 周期。</p> <p>0001: /WR 和 /RD 脉冲宽度 = 2 个 SYSCLK 周期。</p> <p>0010: /WR 和 /RD 脉冲宽度 = 3 个 SYSCLK 周期。</p> <p>0011: /WR 和 /RD 脉冲宽度 = 4 个 SYSCLK 周期。</p> <p>0100: /WR 和 /RD 脉冲宽度 = 5 个 SYSCLK 周期。</p> <p>0101: /WR 和 /RD 脉冲宽度 = 6 个 SYSCLK 周期。</p> <p>0110: /WR 和 /RD 脉冲宽度 = 7 个 SYSCLK 周期。</p> <p>0111: /WR 和 /RD 脉冲宽度 = 8 个 SYSCLK 周期。</p> <p>1000: /WR 和 /RD 脉冲宽度 = 9 个 SYSCLK 周期。</p> <p>1001: /WR 和 /RD 脉冲宽度 = 10 个 SYSCLK 周期。</p> <p>1010: /WR 和 /RD 脉冲宽度 = 11 个 SYSCLK 周期。</p> <p>1011: /WR 和 /RD 脉冲宽度 = 12 个 SYSCLK 周期。</p> <p>1100: /WR 和 /RD 脉冲宽度 = 13 个 SYSCLK 周期。</p> <p>1101: /WR 和 /RD 脉冲宽度 = 14 个 SYSCLK 周期。</p> <p>1110: /WR 和 /RD 脉冲宽度 = 15 个 SYSCLK 周期。</p> <p>1111: /WR 和 /RD 脉冲宽度 = 16 个 SYSCLK 周期。</p> <p>位 1-0: EAH1-0: EMIF 地址保持时间位。</p> <p>00: 地址保持时间 = 0 个 SYSCLK 周期。</p> <p>01: 地址保持时间 = 1 个 SYSCLK 周期。</p> <p>10: 地址保持时间 = 2 个 SYSCLK 周期。</p> <p>11: 地址保持时间 = 3 个 SYSCLK 周期。</p> | | | | | | | | |

16.6.1 非复用方式

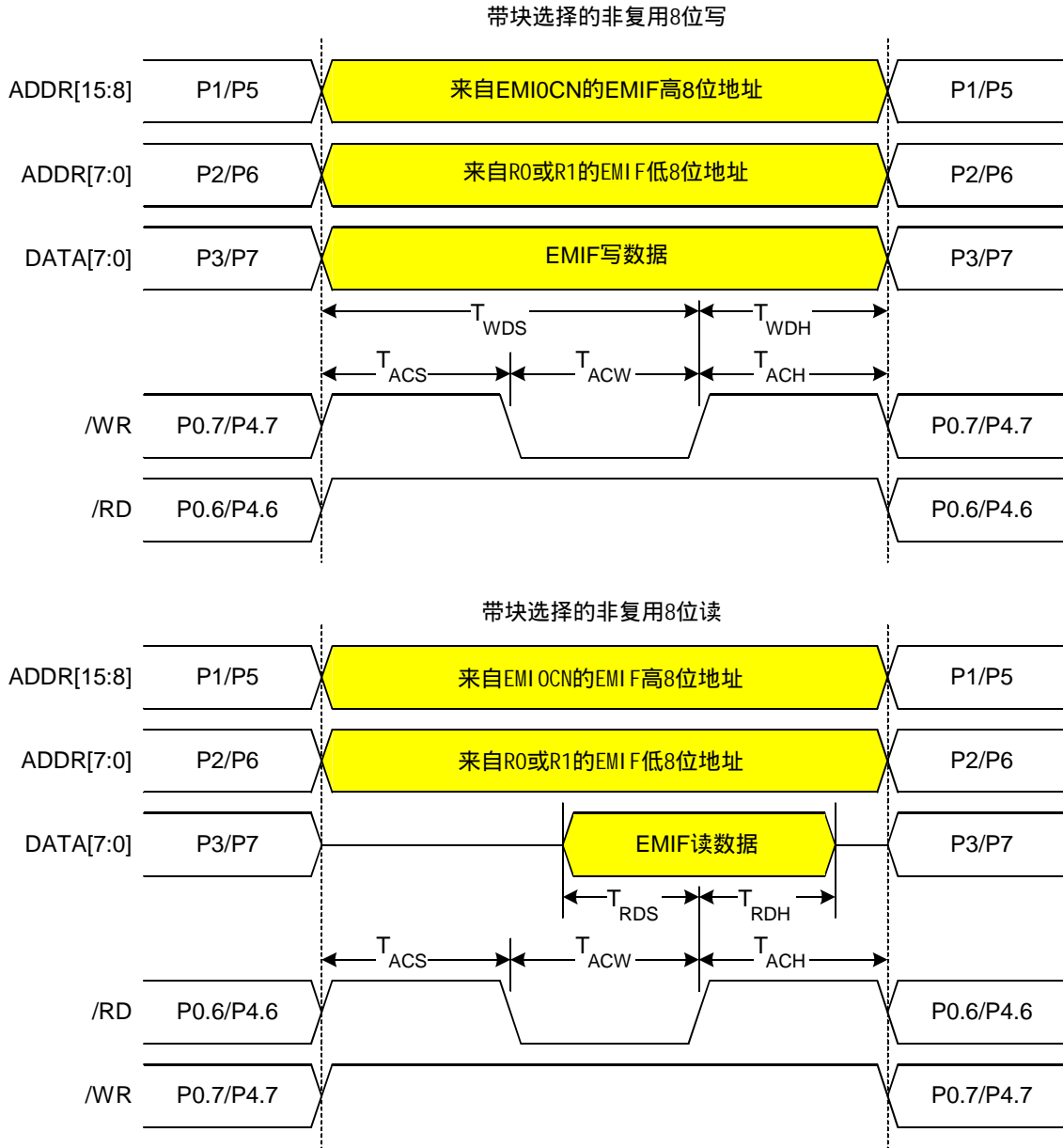
16.6.1.1 16 位 MOVX: EMI0CF[4:2] = '101', '110', '111'

图 16.7 非复用 16 位 MOVX 时序



16.6.1.3 带块选择的8位 MOVX: EMI0CF[4:2] = '110'

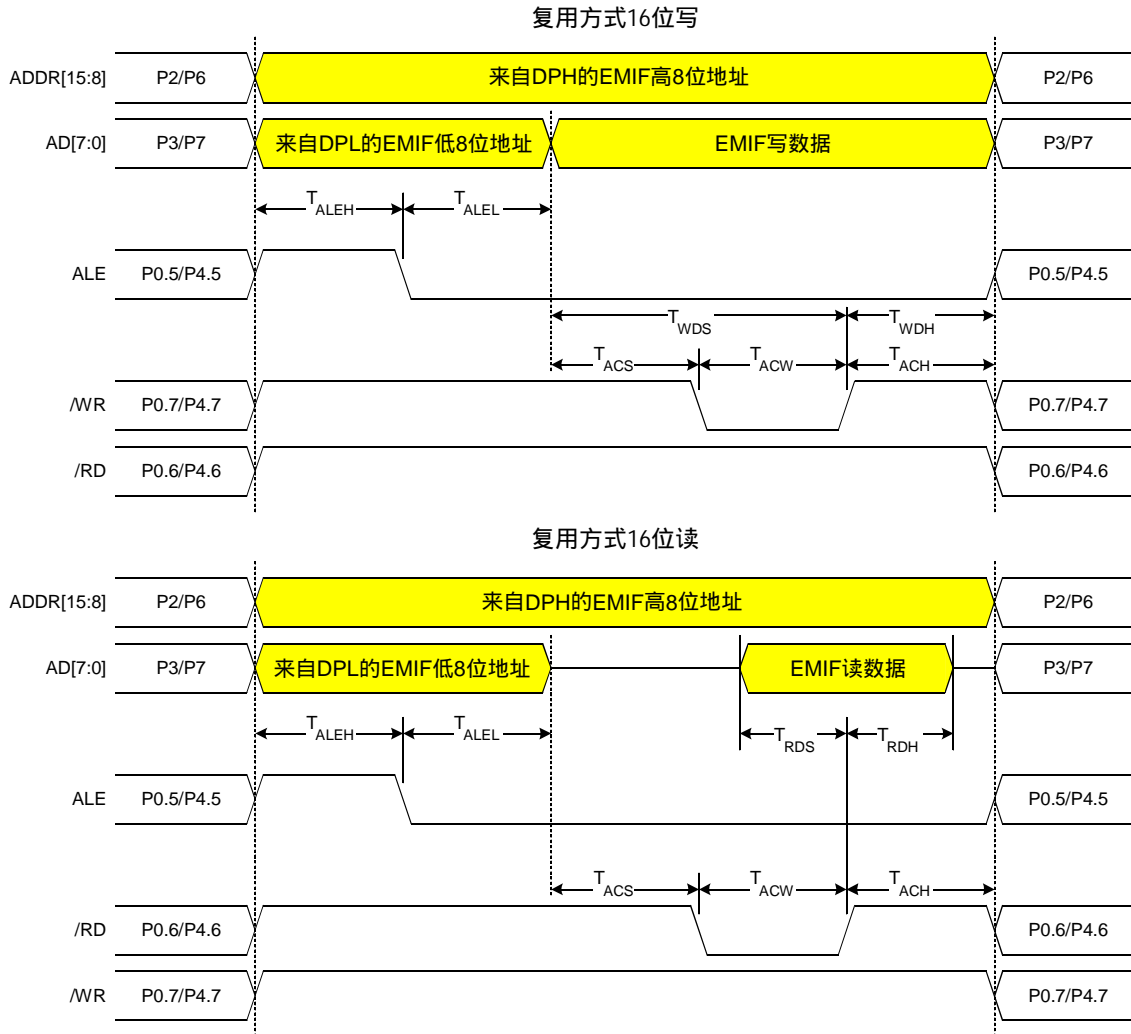
图 16.9 带块选择的非复用8位 MOVX 时序



16.6.2 复用方式

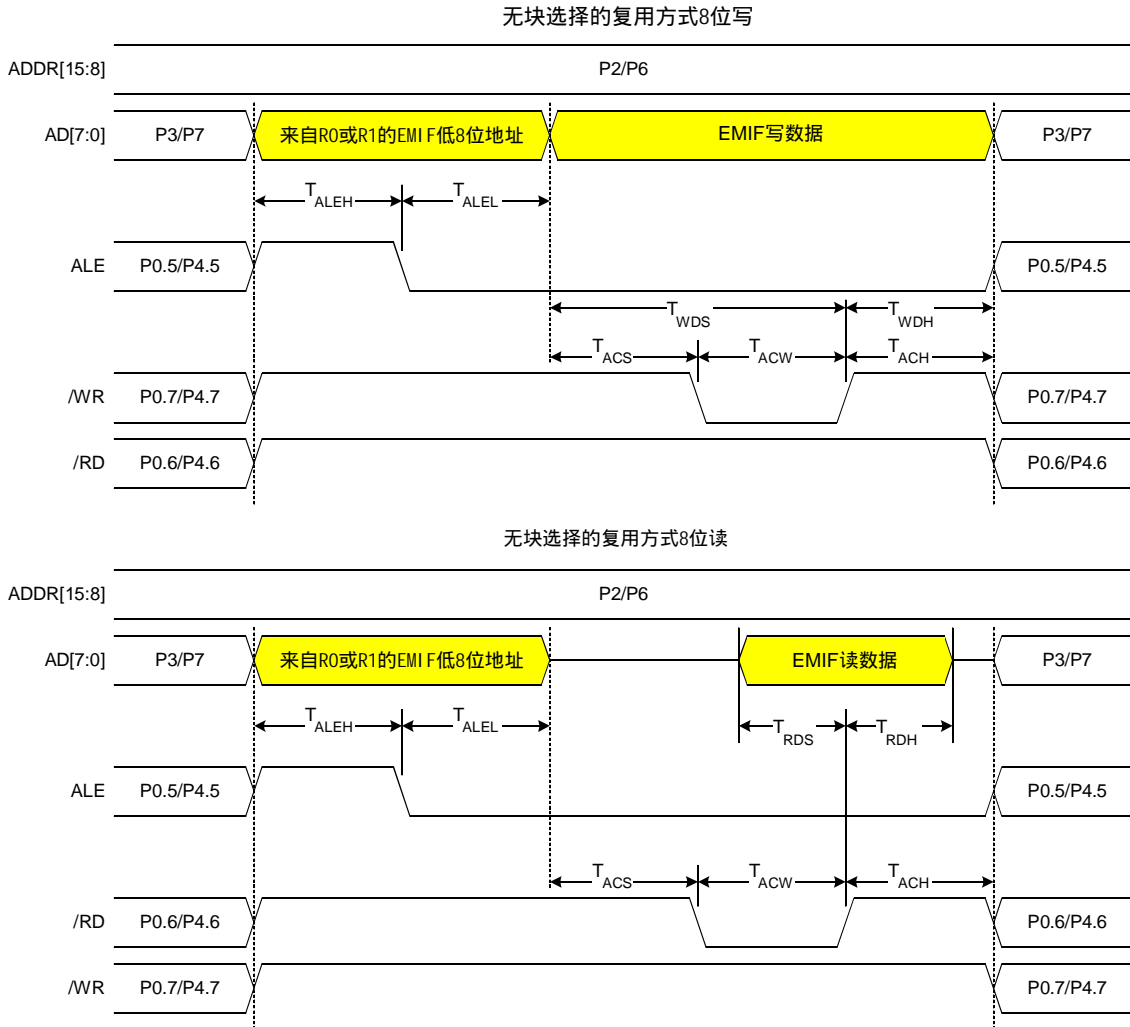
16.6.2.1 16 位 MOVX: EMI0CF[4:2] = '001', '010', '011'

图 16.10 复用方式 16 位 MOVX 时序



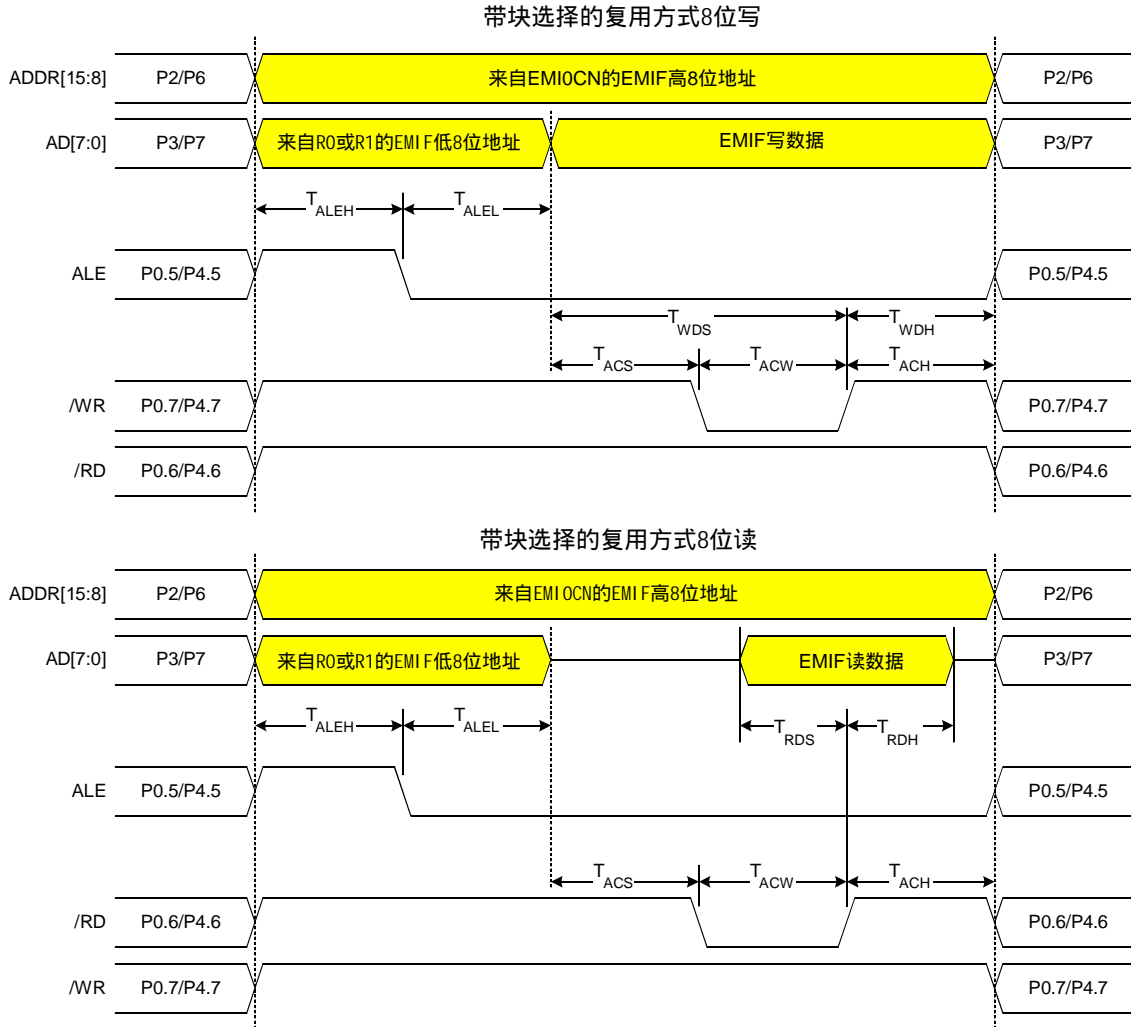
16.6.2.2 无块选择的 8 位 MOVX: EMI0CF[4:2] = '001' 或 '011'

图 16.11 无块选择的复用方式 8 位 MOVX 时序



16.6.2.3 带块选择的 8 位 MOVX: EMI0CF[4:2] = '010'

图 16.12 带块选择的复用方式 8 位 MOVX 时序



C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 16.1 外部存储器接口的 AC 参数

| 参数 | 说明 | 最小值 | 最大值 | 单位 |
|---------------------|-------------|------------------------|-------------------------|----|
| T _{SYSClk} | 系统时钟周期 | 40 | | ns |
| T _{ACS} | 地址/控制建立时间 | 0 | 3* T _{SYSClk} | ns |
| T _{ACW} | 地址/控制脉冲宽度 | 1* T _{SYSClk} | 16* T _{SYSClk} | ns |
| T _{ACH} | 地址/控制保持时间 | 0 | 3* T _{SYSClk} | ns |
| T _{ALEH} | 地址锁存使能高电平时间 | 1* T _{SYSClk} | 4* T _{SYSClk} | ns |
| T _{ALEL} | 地址锁存使能低电平时间 | 1* T _{SYSClk} | 4* T _{SYSClk} | ns |
| T _{WDS} | 写数据建立时间 | 1* T _{SYSClk} | 19* T _{SYSClk} | ns |
| T _{WDH} | 写数据保持时间 | 0 | 3* T _{SYSClk} | ns |
| T _{RDS} | 读数据建立时间 | 20 | | ns |
| T _{RDH} | 读数据保持时间 | 0 | | ns |

17. 端口输入/输出

C8051F020/1/2/3 MCU 是高集成度的混合信号片上系统，有按 8 位端口组织的 64 个数字 I/O 引脚（C8051F020/2）或 32 个数字 I/O 引脚（C8051F021/3）。低端口（P0、P1、P2 和 P3）既可以按位寻址也可以按字节寻址。高端口（P4、P5、P6 和 P7）只能按字节寻址。所有引脚都耐 5V 电压，都可以被配置为漏极开路或推挽输出方式和弱上拉。端口 I/O 单元的原理框图示于图 17.1。表 17.1 给出了端口 I/O 引脚的电气特性。

图 17.1 端口 I/O 单元功能框图

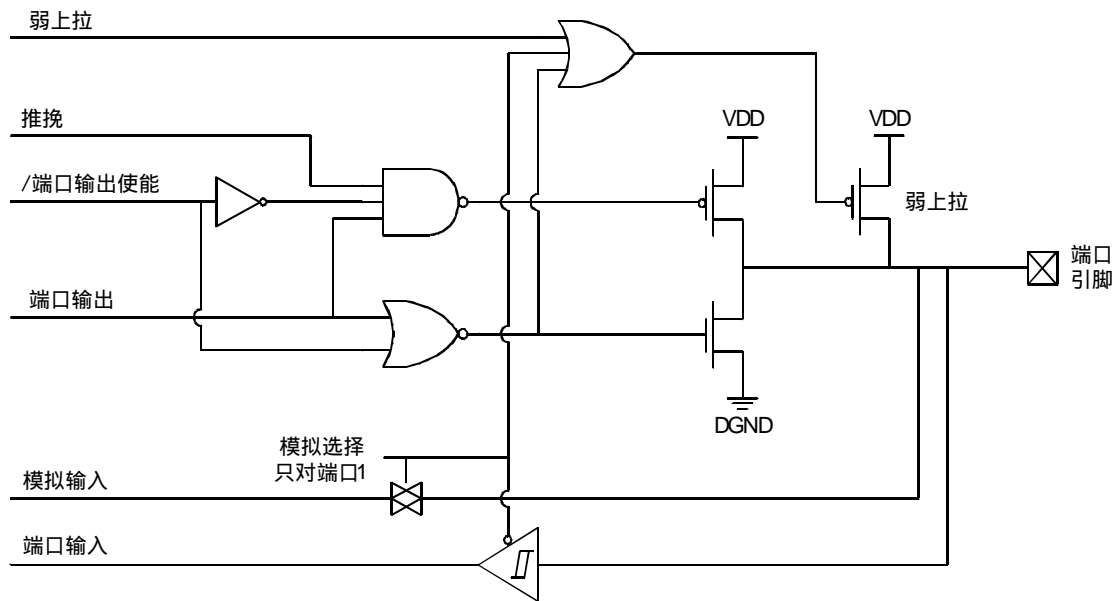


表 17.1 端口 I/O 直流电气特性

VDD = 2.7V – 3.6V, -40°C 到 +85°C (除非另有说明)。

| 参 数 | 条 件 | 最小值 | 典型值 | 最大值 | 单 位 |
|--------------------------|--|--------------------|---------|------------|-----|
| 输出高电压 (V _{OH}) | I _{OH} = -10μA, 端口 I/O 为推挽方式 I _{OH} = -3mA, 端口 I/O 为推挽方式 I _{OH} = -10mA, 端口 I/O 为推挽方式 | VDD-0.1 VDD-0.7 | VDD-0.8 | | V |
| 输出低电压 (V _{OL}) | I _{OL} = 10μA I _{OL} = 8.5mA I _{OL} = 25mA | | 1.0 | 0.1 0.6 | V |
| 输入高电压 (V _{IH}) | | 0.7×VDD | | | V |
| 输入低电压 (V _{IL}) | | | | 0.3×VDD | V |
| 输入漏电流 | DGND < 端口引脚 < VDD, 高阻态 弱上拉关闭 弱上拉打开 | | 10 | ±1 | μA |
| 输入电容 | | | 5 | | pF |

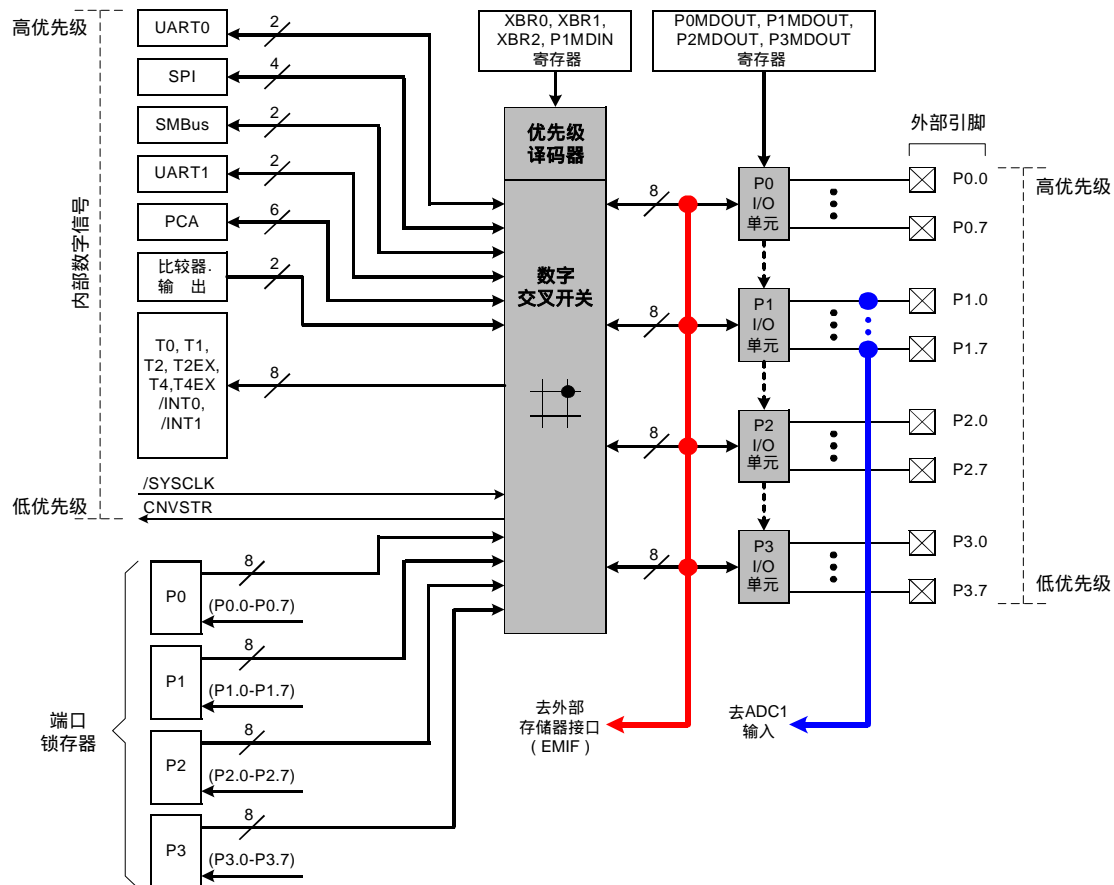
C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

C8051F020/1/2/3 器件有大量的数字资源需要通过 4 个低端 I/O 端口 P0、P1、P2 和 P3 才可以使用。P0、P1、P2 和 P3 中的每个引脚即可定义为通用的端口 I/O (GPIO) 引脚，又可以分配给一个数字外设或功能（例如：UART0 或 INT1），如图 17.2 所示。系统设计者控制数字功能的引脚分配，只受可用引脚数的限制。这种资源分配的灵活性是通过使用优先权交叉开关译码器实现的。注意，不管引脚被分配给一个数字外设或是作为通用 I/O，总是可以通过读相应的数据寄存器得到端口 I/O 引脚的状态。端口 1 的引脚可以用做 ADC1 的模拟输入。

在执行目标地址为片外 XRAM 的 MOVX 指令时，外部存储器接口可以在低端口或高端口有效。有关外部存储器接口的详细信息见“16. 外部数据存储器接口和片内 XRAM”。

高端口（存在于 C8051F020/2 中）可以作为 GPIO 引脚按字节访问。

图 17.2 低端口 I/O 功能框图



17.1 端口 0 – 3 和优先权交叉开关译码器

优先权交叉开关译码器，或称为“交叉开关”，按优先权顺序将端口 0 – 3 的引脚分配给器件上的数字外设（UART、SMBus、PCA、定时器等）。端口引脚的分配顺序是从 P0.0 开始，可以一直分配到 P3.7。为数字外设分配端口引脚的优先权顺序列于图 17.3，UART0 具有最高优先权，而 CNVSTR 具有最低优先权。

引脚本身的逻辑状态。唯一的例外发生在执行读-修改-写指令（ANL、ORL、XRL、CPL、INC、DEC、DJNZ、JBC、CLR、SET 和位传送操作）期间。在读-修改-写指令的读周期，所读的值是端口数据寄存器的内容，而不是端口引脚本身的状态。

因为交叉开关寄存器影响器件外设的引出脚，所以它们通常在外设被配置前由系统的初始化代码配置。一旦配置完毕，将不再对其重新编程。

交叉开关寄存器被正确配置后，通过将 XBARE（XBR2.4）设置为逻辑‘1’来使能交叉开关。在 XBARE 被设置为逻辑‘1’之前，端口 0-3 的输出驱动器被禁止，以防止对交叉开关寄存器和其它寄存器写入时在端口引脚上产生争用。

被交叉开关分配给输入信号（例如 RX0）的引脚所对应的输出驱动器被禁止；因此端口数据寄存器和 PnMDOUT 寄存器的值对这些引脚的状态没有影响。

17.1.2 配置端口引脚的输出方式

在 XBARE（XBR2.4）被设置为逻辑‘1’之前，端口 0-3 的输出驱动器保持禁止状态。

每个端口引脚的输出方式都可被配置为漏极开路或推挽方式，缺省状态为漏极开路。在推挽方式，向端口数据寄存器中的相应位写逻辑‘0’将使端口引脚被驱动到 GND，写逻辑‘1’将使端口引脚被驱动到 VDD。在漏极开路方式，向端口数据寄存器中的相应位写逻辑‘0’将使端口引脚被驱动到 GND，写逻辑‘1’将使端口引脚处于高阻状态。当系统中不同器件的端口引脚有共享连接，即多个输出连接到同一个物理线时（例如 SMBus 连接中的 SDA 信号），使用漏极开路方式可以防止不同器件之间的争用。

端口 0-3 引脚的输出方式由 PnMDOUT 寄存器中的对应位决定（见图 17.11、图 17.14、图 17.16 和图 17.18）。例如，P3MDOUT.7 为逻辑‘1’时将 P3.7 配置为推挽方式；P3MDOUT.7 为逻辑‘0’时将 P3.7 配置为漏极开路方式。所有端口引脚的缺省方式均为漏极开路。

不管交叉开关是否将端口引脚分配给某个数字外设，端口引脚的输出方式都受 PnMDOUT 寄存器控制。例外情况是：连接到 SDA、SCL、RX0（如果 UART0 工作于方式 0）、RX1（如果 UART1 工作于方式 0）的端口引脚总是被配置为漏极开路输出，而与 PnMDOUT 寄存器中的对应位的设置值无关。

17.1.3 配置端口引脚为数字输入

通过设置输出方式为“漏极开路”并向端口数据寄存器中的相应位写‘1’将端口引脚配置为数字输入。例如，设置 P3MDOUT.7 为逻辑‘0’并设置 P3.7 为逻辑‘1’即可将 P3.7 配置为数字输入。

如果一个端口引脚被交叉开关分配给某个数字外设，并且该引脚的功能为输入（例如 UART0 的接收引脚 RX0），则该引脚的输出驱动器被自动禁止。

17.1.4 外部中断（IE6 和 IE7）

除了外部中断/INT0和/INT1（其引脚由交叉开关分配）之外，P3.6和P3.7可被配置为边沿触发的中断源，用IE6CF（P3IF.2）和IE7CF（P3IF.3）位可以将这两个中断源配置为下降沿或上升沿触发。当检测到P3.6或P3.7有下降沿或上升沿发生时，P3IF寄存器（见图4.36）中对应

的外部中断标志 (IE6或IE7) 将被置 ‘1’。如果对应的中断被允许, 将会产生一个中断, CPU 将转向对应的中断向量地址。

17.1.5 弱上拉

每个端口引脚都有一个内部弱上拉部件, 在引脚与 VDD 之间提供阻性连接 (约 100 kΩ), 在缺省情况下该上拉器件被使能。弱上拉部件可以被总体禁止, 通过向弱上拉禁止位 (WEAKPUD, XBR2.7) 写 ‘1’ 实现。当任何引脚被驱动为逻辑 ‘0’ 时, 弱上拉自动取消; 即输出引脚不能与其自身的上拉部件冲突。对于端口 1 的引脚, 将引脚配置为模拟输入时上拉部件也被禁止, 见下面的说明。

17.1.6 配置端口 1 的引脚为模拟输入 (AIN.[7:0])

端口 1 的引脚可以用作 ADC1 模拟多路开关的模拟输入。通过向 P1MDIN 寄存器 (见图 17.13) 中的对应位写 ‘0’ 即可将端口引脚配置为模拟输入。缺省情况下端口引脚为数字输入方式。将一个端口引脚配置为模拟输入的过程如下:

1. 禁止引脚的数字输入路径。这可以防止在引脚上的电压接近 VDD / 2 时消耗额外的电源电流。读端口数据为将返回逻辑 ‘0’, 与加在引脚上的电压无关。
2. 禁止引脚的弱上拉部件。
3. 使交叉开关在为数字外设分配引脚时跳过该引脚。

注意: 被配置为模拟输入的引脚的输出驱动器并没有被明确地禁止。因此被配置为模拟输入的引脚所对应的 P1MDOUT 位应被设置为逻辑 ‘0’ (漏极开路方式), 对应的端口数据位应被设置为逻辑 ‘1’ (高阻态)。需要注意的是, 将一个端口引脚用作 ADC1 模拟多路开关的输入时并不要求将其配置为模拟输入, 但强烈建议这样做。有关 ADC1 的更详细信息见 “7. ADC1 (8 位 ADC)”。

17.1.7 外部存储器接口引脚分配

如果外部存储器接口 (EMIF) 被设置在低端口 (端口 0-3), EMIFLE (XBR2.5) 位应被设置为逻辑 ‘1’, 以使交叉开关不将 P0.7 (/WR)、P0.6 (/RD)和 P0.5 (/ALE) (如果外部存储器接口使用复用方式) 分配给外设。图 17.4 给出了 EMIFLE=1 并且 EMIF 工作在复用方式时的交叉开关译码表的示例。图 17.5 给出了 EMIFLE=1 并且 EMIF 工作在非复用方式时的交叉开关译码表的示例。

如果外部存储器接口被设置在低端口并且发生一次片外 MOVX 操作, 则在该 MOVX 指令执行期间外部存储器接口将控制有关端口引脚的输出状态, 而不管交叉开关寄存器和端口数据寄存器的设置如何。端口引脚的输出配置不受 EMIF 操作的影响, 但读操作将禁止数据总线上的输出驱动器。有关外部存储器接口的详细信息见 “16. 外部数据存储器接口和片内 XRAM”。

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

图 17.4 优先权交叉开关译码表

(EMIFLE = 1; EMIF 工作在复用方式; P1MDIN = 0xFF)

| 引脚 I/O | P0 | | | | | | | P1 | | | | | | | P2 | | | | | | | P3 | | | | | | | 交叉开关寄存器位 | | |
|--|----|---|---|---|---|---|---|----|---|---|---|---|---|---|----|---|---|---|---|---|---|----|---|---|---|---|---|---|----------|-------------------|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | | 4 | 5 |
| TX0 RX0 | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | | UART0EN:XBR0.2 | |
| SCK MISO MOSI NSS | ● | ● | ● | ● | ● | | | ● | | | | | | | | | | | | | | | | | | | | | | SPI0EN:XBR0.1 | |
| SDA SCL | ● | ● | ● | ● | ● | | | ● | ● | ● | | | | | | | | | | | | | | | | | | | | SMB0EN:XBR0.0 | |
| TX1 RX1 | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | UART1EN:XBR2.2 | |
| CEX0 CEX1 CEX2 CEX3 CEX4 | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | PCA0ME:XBR0.[5:3] | |
| ECl | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | ECI0E:XBR0.6 | |
| CP0 | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | CP0E:XBR0.7 | |
| CP1 | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | CP1E:XBR1.0 | |
| T0 | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | T0E:XBR1.1 | |
| /INT0 | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | INT0E:XBR1.2 | |
| T1 | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | T1E:XBR1.3 | |
| /INT1 | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | INT1E:XBR1.4 | |
| T2 | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | T2E:XBR1.5 | |
| T2EX | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | T2EXE:XBR1.6 | |
| T4 | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | T4E:XBR2.3 | |
| T4EX | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | T4EXE:XBR2.4 | |
| /SYSCLK | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | SYSCKE:XBR1.7 | |
| CNVSTR | ● | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | CNVSTE:XBR2.0 | |
| ALE /RD /WR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AIN1.0/A8 AIN1.1/A9 AIN1.2/A10 AIN1.3/A11 AIN1.4/A12 AIN1.5/A13 AIN1.6/A14 AIN1.7/A15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | AIN1 输入/非复用地址高 | |
| A8m/A0 A9m/A1 A10m/A2 A11m/A3 A12m/A4 A13m/A5 A14m/A6 A15m/A7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 复用地址高/非复用地址低 | |
| AD0/D0 AD1/D1 AD2/D2 AD3/D3 AD4/D4 AD5/D5 AD6/D6 AD7/D7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 复用数据/非复用数据 | |

17.1.8 交叉开关引脚分配示例

在本例中（图 17.6），我们将配置交叉开关，为 UART0、SMBus、UART1、/INT0 和/INT1 分配端口引脚（共 8 个引脚）。另外，我们将外部存储器接口配置为复用方式并使用低端口。我们还将 P1.2、P1.3 和 P1.4 配置为模拟输入，以使用 ADC1 测量加在这些引脚上的电压。配置步骤如下：

1. 按 $UART0EN = 1$ 、 $SMB0EN = 1$ 、 $INT0E = 1$ 、 $INT1E = 1$ 和 $EMIFLE = 1$ 设置 XBR0、XBR1 和 XBR2，则有： $XBR0 = 0x05$ ， $XBR1 = 0x14$ ， $XBR2 = 0x02$ 。
2. 将外部存储器接口配置为复用方式并使用低端口，有： $PRTSEL = 0$ ， $EMD2 = 0$ 。
3. 将作为模拟输入的端口 1 引脚配置为模拟输入方式：设置 P1MDIN 为 $0xE3$ （P1.4、P1.3 和 P1.2 为模拟输入，所以它们的对应 P1MDIN 被设置为逻辑 ‘0’）。
4. 设置 $XBARE = 1$ 以允许交叉开关： $XBR2 = 0x42$ 。
 - UART0 有最高优先权，所以 P0.0 被分配给 TX0，P0.1 被分配给 RX0。
 - SMBus 的优先权次之，所以 P0.2 被分配给 SDA，P0.3 被分配给 SCL。
 - 接下来是 UART1，所以 P0.4 被分配给 TX1。由于外部存储器接口选在低端口（ $EMIFLE = 1$ ），所以交叉开关跳过 P0.6(/RD)和 P0.7(/WR)。又因为外部存储器接口被配置为复用方式，所以交叉开关也跳过 P0.5(ALE)。下一个未被跳过的引脚 P1.0 被分配给 RX1。
 - 接下来是/INT0，被分配到引脚 P1.1。
 - 将 P1MDIN 设置为 $0xE3$ ，使 P1.2、P1.3 和 P1.4 被配置为模拟输入，导致交叉开关跳过这些引脚。
 - 下面优先权高的是/INT1，所以下一个未跳过的引脚 P1.5 被分配给/INT1。
 - 在执行对片外操作的 MOVX 指令期间，外部存储器接口将驱动端口 2 和端口 3（由图 17.6 中的红点表示）。
5. 我们将 UART0 的 TX 引脚（TX0，P0.0）、UART1 的 TX 引脚（TX1，P0.4）、ALE、/RD、/WR（P0.[7:3]）的输出设置为推挽方式，通过设置 $P0MDOUT = 0xF1$ 来实现。
6. 我们通过设置 $P2MDOUT = 0xFF$ 和 $P3MDOUT = 0xFF$ 将 EMIF 端口（P2、P3）的输出方式配置为推挽方式。
7. 我们通过设置 $P1MDOUT = 0x00$ （配置输出为漏极开路）和 $P1 = 0xFF$ （逻辑 ‘1’ 选择高阻态）禁止 3 个模拟输入引脚的输出驱动器。

图 17.7 XBR0: 端口 I/O 交叉开关寄存器 0

| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|-------------|---|-------|--------|-----|---------|--------|--------|----------|
| | CPOE | ECIOE | PCA0ME | | UART0EN | SPIOEN | SMB0EN | 00000000 |
| | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 |
| SFR地址: 0xE1 | | | | | | | | |
| 位 7: | CPOE: 比较器 0 输出允许位 0: CP0 不连到端口引脚。 1: CP0 连到端口引脚。 | | | | | | | |
| 位 6: | ECIOE: PCA0 外部计数器输入允许位 0: PCA0 外部计数器输入不连到端口引脚。 1: PCA0 外部计数器输入连到端口引脚。 | | | | | | | |
| 位 5-3: | PCA0ME: PCA0 模块 I/O 允许位 000: 所有的 PCA0 I/O 都不连到端口引脚。 001: CEX0 连到端口引脚。 010: CEX0、CEX1 连到 2 个端口引脚。 011: CEX0、CEX1、CEX2 连到 3 个端口引脚。 100: CEX0、CEX1、CEX2、CEX3 连到 4 个端口引脚。 1xx: CEX0、CEX1、CEX2、CEX3、CEX4 连到 5 个端口引脚。 | | | | | | | |
| 位 2: | UART0EN: UART0 I/O 允许位。 0: UART0 I/O 不连到端口引脚。 1: RX0、TX0 连到 2 个端口引脚。 | | | | | | | |
| 位 1: | SPIOEN: SPI 总线 I/O 允许位。 0: SPI0 I/O 不连到端口引脚。 1: SPI0 的 SCK、MISO、MOSI 和 NSS 连到 4 个端口引脚。 | | | | | | | |
| 位 0: | SMB0EN: SMBus 总线 I/O 允许位 0: SMBus0 I/O 不连到端口引脚。 1: SMBus0 的 SDA 连到 2 个端口引脚。 | | | | | | | |

图 17.8 XBR1: 端口 I/O 交叉开关寄存器 1

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|-------|-----|-------|-----|-------|-----|------|----------------|
| SYSCKE | T2EXE | T2E | INT1E | T1E | INT0E | T0E | CP1E | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xE2 |
| <p>位 7: SYSCKE: SYSCLK 输出允许位 0: SYSCLK 不连到端口引脚。 1: SYSCLK 连到端口引脚。</p> <p>位 6: T2EXE: T2EX 允许位 0: T2EX 不连到端口引脚。 1: T2EX 连到端口引脚。</p> <p>位 5: T2E: T2 允许位 0: T2 不连到端口引脚。 1: T2 连到端口引脚。</p> <p>位 4: INT1E: /INT1 允许位。 0: /INT1 不连到端口引脚。 1: /INT1 连到端口引脚。</p> <p>位 3: T1E: T1 允许位 0: T1 不连到端口引脚。 1: T1 连到端口引脚。</p> <p>位 2: INT0E: /INT0 允许位 0: /INT0 不连到端口引脚。 1: /INT0 连到端口引脚。</p> <p>位 1: T0E: T0 允许位 0: T0 不连到端口引脚。 1: T0 连到端口引脚。</p> <p>位 0: CP1E: 比较器 1 输出允许位 0: CP1 不连到端口引脚。 1: CP1 连到端口引脚。</p> | | | | | | | | |

图 17.9 XBR2: 端口 I/O 交叉开关寄存器 2

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--|-------|-----|-------|-----|--------|--------|--------|----------------|
| WEAKPUD | XBARE | - | T4EXE | T4E | UART1E | EMIFLE | CNVSTE | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xE3 |
| <p>位 7: WEAKPUD: 弱上拉禁止位。 0: 弱上拉全局允许。 1: 弱上拉全局禁止。</p> <p>位 6: XBARE: 交叉开关允许位 0: 交叉开关禁止。端口 0、1、2 和 3 的所有引脚被强制为输入方式。 1: 交叉开关允许。</p> <p>位 5: 未用。读=0, 写=忽略。</p> <p>位 4: T4EXE: T4EX 输入允许位 0: T4EX 不连到端口引脚。 1: T4EX 连到端口引脚。</p> <p>位 3: T4E: T4 输入允许位 0: T4 不连到端口引脚。 1: T4 连到端口引脚。</p> <p>位 2: UART1E: UART1 I/O 允许位 0: UART1 I/O 不连到端口引脚。 1: UART1 TX 和 RX 连到两个端口引脚。</p> <p>位 1: EMIFLE: 外部存储器接口低端允许位 0: P0.7、P0.6 和 P0.5 的功能由交叉开关或端口锁存器决定。 1: 如果 EMI0CF.4 = '0' (外部存储器接口为复用方式) 则 P0.7 (/WR)、P0.6 (/RD)和 P0.5 (/ALE)被交叉开关跳过, 它们的输出状态由端口锁存器和外部存储器接口决定。 1: 如果 EMI0CF.4 = '1' (外部存储器接口为非复用方式) 则 P0.7 (/WR)和 P0.6 (/RD)被交叉开关跳过, 它们的输出状态由端口锁存器和外部存储器接口决定。</p> <p>位 0: CNVSTE: 外部转换启动输入允许位 0: CNVSTR 不连到端口引脚。 1: CNVSTR 连到端口引脚。</p> | | | | | | | | |

图 17.10 P0: 端口 0 寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|------|------|------|------|------|------|------|--------------|----------------|
| P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 | 11111111 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0x80 |

位 7-0: P0.[7:0]: 端口 0 输出锁存器位。
 (写 — 输出出现在 I/O 引脚, 根据 XBR0、XBR1 和 XBR2 寄存器的设置)
 0: 逻辑低电平输出。
 1: 逻辑高电平输出。(若相应的 P0MDOUT.n 位 = 0, 则为漏极开路)。
 (读 — 与 XBR0、XBR1 和 XBR2 寄存器的设置无关)
 0: P0.n 为逻辑低电平。
 1: P0.n 为逻辑高电平。

注: P0.7 (/WR)、P0.6 (/RD)和 P0.5 (/ALE)可由外部数据存储器接口驱动。更详细的信息见“16. 外部数据存储器接口和片内 XRAM”。关于为外部存储器访问配置交叉开关的信息见图 17.9。

图 17.11 P0MDOUT: 端口 0 输出方式寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| | | | | | | | | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xA4 |

位 7-0: P0MDOUT.[7:0]: 端口 0 输出方式位。
 0: 端口引脚的输出方式为漏极开路。
 1: 端口引脚的输出方式为推挽。

注: 当 SDA、SCL、RX0 (当 UART0 工作于方式 0 时) 和 RX1 (当 UART1 工作于方式 0 时) 出现在端口引脚时, 总是被配置为漏极开路输出。

图 17.12 P1: 端口 1 寄存器

| | | | | | | | | |
|------|------|------|------|------|------|------|--------------|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 | 11111111 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0x90 |

位 7-0: P1.[7:0]: 端口 1 输出锁存器位。
 (写 1 输出出现在 I/O 引脚, 根据 XBR0、XBR1 和 XBR2 寄存器的设置)

0: 逻辑低电平输出。
 1: 逻辑高电平输出。(若相应的 P1MDOUT.n 位 = 0, 则为漏极开路)。
 (读 1 与 XBR0、XBR1 和 XBR2 寄存器的设置无关)

0: P1.n 为逻辑低电平。
 1: P1.n 为逻辑高电平。

注:

- P1.[7:0]可以被配置为 ADC1 的输入 AIN1.[7:0]。在这种情况下, 交叉开关的引脚分配将跳过这些引脚, 它们的数字输入通路被禁止, 由 P1MDIN(见图 17.13) 决定。注意: 在模拟方式, 引脚的输出方式由端口 1 锁存器和 P1MDOUT(图 17.14) 决定。有关 ADC1 的更详细信息见“7. ADC1 (8 位 ADC)”。
- P1.[7:0]可以由外部数据存储器接口驱动(在非复用方式作为地址 A[15:8])。有关外部存储器接口的更详细信息见“16. 外部数据存储器接口和片内 XRAM”。

图 17.13 P1MDIN: 端口 1 输入方式寄存器

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 11111111 |
| | | | | | | | | SFR地址: 0xBD |

位 7-0: P1MDIN.[7:0]: 端口 1 输入方式位。

0: 端口引脚被配置为模拟输入方式。数字输入通路被禁止(读端口位将总是返回 '0')。引脚的弱上拉被禁止。
 1: 端口引脚被配置为数字输入方式。读端口位将返回引脚的逻辑电平。弱上拉状态由 WEAKPUD 位(XBR2.7, 见图 17.9) 决定。

图 17.14 P1MDOUT: 端口 1 输出方式寄存器

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| | | | | | | | | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xA5 |

位 7-0: P1MDOUT.[7:0]: 端口 1 输出方式位。
 0: 端口引脚的输出方式为漏极开路。
 1: 端口引脚的输出方式为推挽。

注: 当 SDA、SCL、RX0 (当 UART0 工作于方式 0 时) 和 RX1 (当 UART1 工作于方式 0 时) 出现在端口引脚时, 总是被配置为漏极开路输出。

图 17.15 P2: 端口 2 数据寄存器

| | | | | | | | | |
|------|------|------|------|------|------|------|--------------|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 | 11111111 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0xA0 |

位 7-0: P2.[7:0]: 端口 2 输出锁存器位。
 (写 — 输出出现在 I/O 引脚, 根据 XBR0、XBR1 和 XBR2 寄存器的设置)
 0: 逻辑低电平输出。
 1: 逻辑高电平输出。(若相应的 P1MDOUT.n 位 = 0, 则为漏极开路)。
 (读 — 与 XBR0、XBR1 和 XBR2 寄存器的设置无关)
 0: P2.n 为逻辑低电平。
 1: P2.n 为逻辑高电平。

注: P2.[7:0] 可以由外部数据存储器接口驱动 (在复用方式作为地址 A[15:8] 或在非复用方式作为地址 A[7:0])。有关外部存储器接口的更详细信息见 “16. 外部数据存储器接口和片内 XRAM”。

图 17.16 P2MDOUT: 端口 2 输出方式寄存器

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| | | | | | | | | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xA6 |

位 7-0: P2MDOUT.[7:0]: 端口 2 输出方式位。
 0: 端口引脚的输出方式为漏极开路。
 1: 端口引脚的输出方式为推挽。

注: 当 SDA、SCL、RX0 (当 UART0 工作于方式 0 时) 和 RX1 (当 UART1 工作于方式 0 时) 出现在端口引脚时, 总是被配置为漏极开路输出。

图 17.17 P3: 端口 3 数据寄存器

| | | | | | | | | |
|------|------|------|------|------|------|------|------|--------------------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 | 11111111 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xB0 (可位寻址) |

位 7-0: P3.[7:0]: 端口 3 输出锁存器位。
 (写 — 输出出现在 I/O 引脚, 根据 XBR0、XBR1 和 XBR2 寄存器的设置)
 0: 逻辑低电平输出。
 1: 逻辑高电平输出。(若相应的 P1MDOUT.n 位 = 0, 则为漏极开路)。
 (读 — 与 XBR0、XBR1 和 XBR2 寄存器的设置无关)
 0: P3.n 为逻辑低电平。
 1: P3.n 为逻辑高电平。

注: P3.[7:0] 可以由外部数据存储器接口驱动 (在复用方式作为 AD[7:0]或在非复用方式作为 D[7:0])。有关外部存储器接口的更详细信息见“16. 外部数据存储器接口和片内 XRAM”。

图 17.18 P3MDOUT: 端口 3 输出方式寄存器

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| | | | | | | | | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xA7 |

位 7-0: P3MDOUT.[7:0]: 端口 3 输出方式位。
 0: 端口引脚的输出方式为漏极开路。
 1: 端口引脚的输出方式为推挽。

注: 当 SDA、SCL、RX0 (当 UART0 工作于方式 0 时) 和 RX1 (当 UART1 工作于方式 0 时) 出现在端口引脚时, 总是被配置为漏极开路输出。

图 17.19 P3IF: 端口 3 中断标志寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--|-----|-----|-----|-------|-------|-----|-----|----------------|
| IE7 | IE6 | - | - | IE7CF | IE6CF | - | - | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xAD |
| <p>位 7: IE7: 外部中断 7 标志位 0: P3.7 引脚没有检测到下降沿。 1: 当检测到 P3.7 引脚的下降沿时该标志由硬件置位。</p> <p>位 6: IE6: 外部中断 6 标志 0: P3.6 引脚没有检测到下降沿。 1: 当检测到 P3.6 引脚的下降沿时该标志由硬件置位。</p> <p>位 5-4: 未使用。读= 00b, 写 = 忽略。</p> <p>位 3: IE7CF: 外部中断 7 边沿配置位 0: 外部中断 7 由 IE7 输入的下降沿触发。 1: 外部中断 7 由 IE7 输入的上升沿触发。</p> <p>位 2: IE6CF: 外部中断 6 边沿配置位 0: 外部中断 6 由 IE6 输入的下降沿触发。 1: 外部中断 6 由 IE6 输入的上升沿触发。</p> <p>位 1-0: 未用。读 = 00b, 写=忽略。</p> | | | | | | | | |

17.2 端口 4-7 (只对 C8051F020/2)

端口 4-7 的所有端口引脚都可用作通用 I/O (GPIO), 通过读和写相应的端口数据寄存器 (见图 17.21、图 17.22、图 17.23 和图 17.24) 访问每个端口, 这些端口数据寄存器是一组按字节寻址的特殊功能寄存器。

读端口数据寄存器时, 返回的是端口引脚本身的逻辑状态。例外的情况发生在执行读-修改-写指令 (ANL、ORL、XRL、JBC、CPL、INC、DEC、DJNZ) 期间。在读-修改-写指令的读周期, 读入的是端口数据寄存器的内容, 而不是端口引脚本身的状态。

17.2.1 配置无引出脚的端口

尽管 P4、P5、P6 和 P7 在 C8051F02/3 中没有对应的引脚, 但端口数据寄存器仍然存在并可被软件所用。由于数字输入通路保持活动状态, 随意建议不要将这些引脚处于“浮空”状态, 以避免因输入浮空为一个无效逻辑电平而导致不必要的功率消耗。下面的任何一种措施都可以防止这种情况出现:

1. 通过将 WEAKPUD (XBR2.7) 设置为逻辑 ‘0’ 来使能弱上拉部件。
2. 通过写 P74OUT = 0xFF 将 P4、P5、P6 和 P7 的输出方式配置为推挽方式。
3. 通过向端口数据寄存器写 ‘0’ 将 P4、P5、P6 和 P7 的输出状态强制为逻辑 ‘0’:
P4 = 0x00, P5 = 0x00, P6 = 0x00, P7 = 0x00。

17.2.2 配置端口引脚的输出方式

每个端口引脚的输出方式都可被配置为漏极开路或推挽方式。在推挽方式, 向端口数据寄存器中的相应位写逻辑 ‘0’ 将使端口引脚被驱动到 GND, 写逻辑 ‘1’ 将使端口引脚被驱

动到 VDD。在漏极开路方式，向端口数据寄存器中的相应位写逻辑 ‘0’ 将使端口引脚被驱动到 GND，写逻辑 ‘1’ 将使端口引脚处于高阻状态。当系统中不同器件的端口引脚有共享连接，即多个输出连接到同一个物理线时（例如 SMBus 连接中的 SDA 信号），使用漏极开路方式可以防止不同器件之间的冲突。

端口 4-7 引脚的输出方式由 P74OUT 寄存器（见图 17.20）中的位决定。P74OUT 中的每一位控制端口 4-7 中一组引脚（每组 4 位）的输出方式。P74OUT.7 为逻辑 ‘1’ 时将端口 7 中高 4 位（P7.[7:4]）的输出方式配置为推挽方式；P74OUT.7 为逻辑 ‘0’ 时将端口 7 中高 4 位（P7.[7:4]）的输出方式配置为漏极开路。

17.2.3 配置端口引脚为数字输入

通过设置输出方式为“漏极开路”并向端口数据寄存器中的相应位写 ‘1’ 将端口引脚配置为数字输入。例如，设置 P4OUT.7 为逻辑 ‘0’ 并设置 P7.7 为逻辑 ‘1’ 即可将 P7.7 配置为数字输入。

17.2.4 弱上拉

每个端口引脚都有一个内部弱上拉部件，在缺省情况下该上拉器件被使能，在引脚与 VDD 之间提供阻性连接（约 100 k Ω ）。弱上拉部件可以被总体禁止，通过向弱上拉禁止位（WEAKPUD，XBR2.7）写 ‘1’ 实现。当任何引脚被驱动为逻辑 ‘0’ 时，弱上拉自动取消；即输出引脚不能与其自身的上拉部件冲突。

17.2.5 外部存储器接口

如果外部存储器接口（EMIF）被设置在高端口（端口 4-7），EMIFLE（XBR2.5）位应被设置为逻辑 ‘0’。

如果外部存储器接口被设置在高端口并且发生一次片外 MOVX 操作，则在该 MOVX 指令执行期间外部存储器接口将控制有关端口引脚的输出状态，而不管端口数据寄存器的设置如何。端口引脚的输出配置不受 EMIF 操作的影响，但读操作将禁止数据总线上的输出驱动器。有关外部存储器接口的详细信息见“16. 外部数据存储器接口和片内 XRAM”。

图 17.20 P74OUT: 端口 7 - 4 输出方式寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|-----|-----|-----|-----|-----|-----|-----|----------------|
| P7H | P7L | P6H | P6L | P5H | P5L | P4H | P4L | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xB5 |
| <p>位 7: P7H: 端口 7 高 4 位输出方式位。 0: P7.[7:4]配置为漏极开路。 1: P7.[7:4]配置为推挽方式。</p> <p>位 6: P7L: 端口 7 低 4 位输出方式位。 0: P7.[3:0]配置为漏极开路。 1: P7.[3:0]配置为推挽方式。</p> <p>位 5: P6H: 端口 6 高 4 位输出方式位。 0: P6.[7:4]配置为漏极开路。 1: P6.[7:4]配置为推挽方式。</p> <p>位 4: P6L: 端口 6 低 4 位输出方式位。 0: P6.[3:0]配置为漏极开路。 1: P6.[3:0]配置为推挽方式。</p> <p>位 3: P5H: 端口 5 高 4 位输出方式位。 0: P5.[7:4]配置为漏极开路。 1: P5.[7:4]配置为推挽方式。</p> <p>位 2: P5L: 端口 5 低 4 位输出方式位。 0: P5.[3:0]配置为漏极开路。 1: P5.[3:0]配置为推挽方式。</p> <p>位 1: P4H: 端口 4 高 4 位输出方式位。 0: P4.[7:4]配置为漏极开路。 1: P4.[7:4]配置为推挽方式。</p> <p>位 0: P4L: 端口 4 低 4 位输出方式位。 0: P4.[3:0]配置为漏极开路。 1: P4.[3:0]配置为推挽方式。</p> | | | | | | | | |

图 17.21 P4: 端口 4 数据寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|------|------|------|------|------|------|------|------|----------------|
| P4.7 | P4.6 | P4.5 | P4.4 | P4.3 | P4.2 | P4.1 | P4.0 | 11111111 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x84 |

位 7-0: P4.[7:0]: 端口 4 输出锁存器位。
 写 — 输出出现在 I/O 引脚。
 0: 逻辑低电平输出。
 1: 逻辑高电平输出。(若相应的 P74OUT 位 = 0, 则为漏极开路)。见图 17.20。
 读 — 返回 I/O 引脚的状态。
 0: P4.n 为逻辑低电平。
 1: P4.n 为逻辑高电平。

注: P4.7 (/WR)、P4.6 (/RD) 和 P4.5 (ALE) 可以由外部数据存储器接口驱动。有关外部存储器接口的更详细信息见“16. 外部数据存储器接口和片内 XRAM”。

图 17.22 P5: 端口 5 数据寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|------|------|------|------|------|------|------|------|----------------|
| P5.7 | P5.6 | P5.5 | P5.4 | P5.3 | P5.2 | P5.1 | P5.0 | 11111111 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x85 |

位 7-0: P5.[7:0]: 端口 5 输出锁存器位。
 写 — 输出出现在 I/O 引脚。
 0: 逻辑低电平输出。
 1: 逻辑高电平输出。(若相应的 P74OUT 位 = 0, 则为漏极开路)。见图 17.20。
 读 — 返回 I/O 引脚的状态。
 0: P5.n 为逻辑低电平。
 1: P5.n 为逻辑高电平。

注: P5.[7:0] 可以由外部数据存储器接口驱动(在非复用方式作为地址 A[15:8])。有关外部存储器接口的更详细信息见“16. 外部数据存储器接口和片内 XRAM”。

图 17.23 P6: 端口 6 数据寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|------|------|------|------|------|------|------|------|----------------|
| P6.7 | P6.6 | P6.5 | P6.4 | P6.3 | P6.2 | P6.1 | P6.0 | 11111111 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x86 |

位 7-0: P6.[7:0]: 端口 6 输出锁存器位。
 写 — 输出出现在 I/O 引脚。
 0: 逻辑低电平输出。
 1: 逻辑高电平输出。(若相应的 P74OUT 位 = 0, 则为漏极开路)。见图 17.20。
 读 — 返回 I/O 引脚的状态。
 0: P6.n 为逻辑低电平。
 1: P6.n 为逻辑高电平。

注: P6.[7:0] 可以由外部数据存储器接口驱动 (在复用方式作为地址 A[15:8] 或在非复用方式作为地址 A[7:0])。有关外部存储器接口的更详细信息见“16. 外部数据存储器接口和片内 XRAM”。

图 17.24 P7: 端口 7 数据寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|------|------|------|------|------|------|------|------|----------------|
| P7.7 | P7.6 | P7.5 | P7.4 | P7.3 | P7.2 | P7.1 | P7.0 | 11111111 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x96 |

位 7-0: P7.[7:0]: 端口 7 输出锁存器位。
 写 — 输出出现在 I/O 引脚。
 0: 逻辑低电平输出。
 1: 逻辑高电平输出。(若相应的 P74OUT 位 = 0, 则为漏极开路)。见图 17.20。
 读 — 返回 I/O 引脚的状态。
 0: P7.n 为逻辑低电平。
 1: P7.n 为逻辑高电平。

注: P7.[7:0] 可以由外部数据存储器接口驱动 (在非复用方式作为 AD[7:0], 在非复用方式作为 D[7:0])。有关外部存储器接口的更详细信息见“16. 外部数据存储器接口和片内 XRAM”。

18. SMBus

SMBus0 I/O 接口是一个双线的双向串行总线。SMBus0 完全符合系统管理总线规范 1.1 版，与 I²C 串行总线兼容。系统控制器对总线的读写操作都是以字节为单位的，由 SMBus 接口自动控制数据的串行传输。数据传输的最大速率可达系统时钟频率的八分之一（这可能比 SMBus 的规定速度要快，取决于所使用的系统时钟）。可以采用延长低电平时间的方法协调同一总线上不同速度的器件。

SMBus 可以工作在主和/或从方式，一个总线上可以有多个主器件。SMBus 提供了 SDA（串行数据）控制、SCL（串行时钟）产生和同步、仲裁逻辑以及起始/停止的控制和产生电路。有三个与之相关的特殊功能寄存器：配置寄存器 SMB0CN、控制寄存器 SMB0CN 及用于发送和接收数据的数据寄存器 SMB0DAT。

图 18.1 SMBus 原理框图

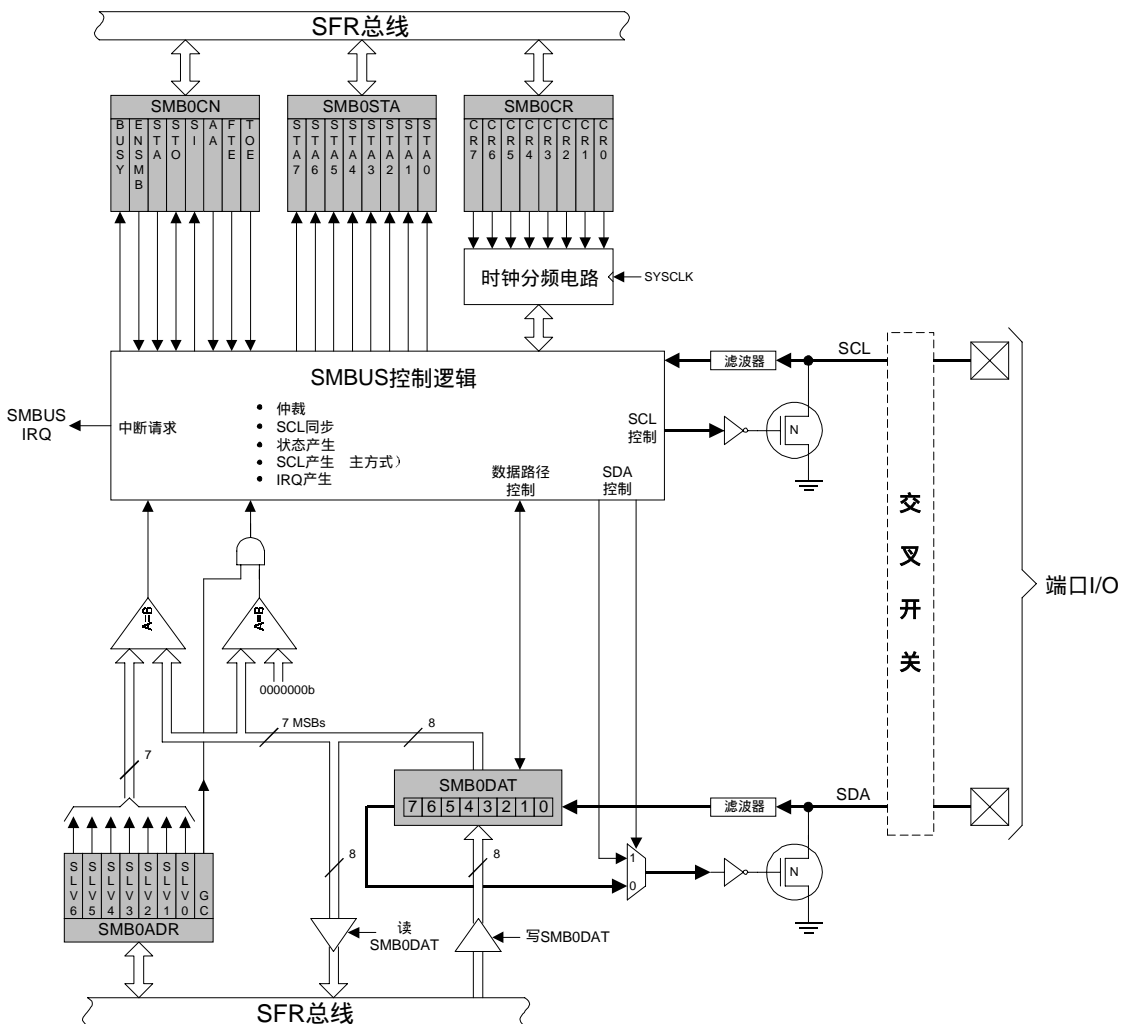
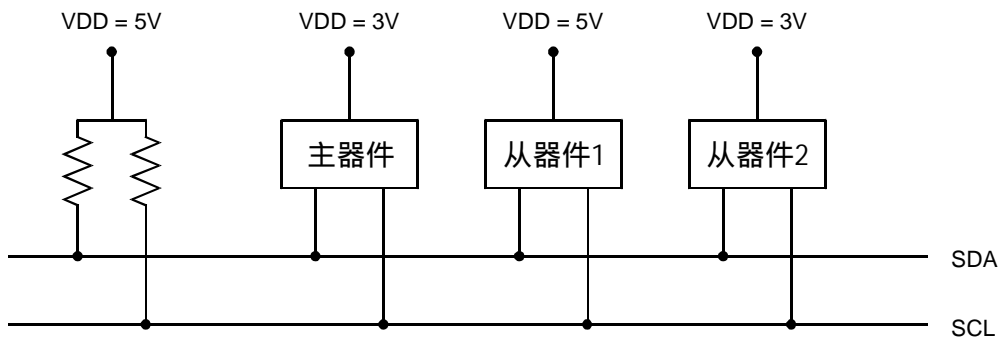


图 18.2 给出了一个典型的 SMBus 配置。SMBus 接口的工作电压可以在 3.0V 和 5.0V 之间，总线上不同器件的工作电压可以不同。SCL（串行时钟）和 SDA（串行数据）线是双向的，必须通过一个上拉电阻或类似电路将它们连到电源电压。连接在总线上的每个器件的 SCL 和 SDA 都必须是漏极开路或集电极开路的，因此当总线空闲时，这两条线都被拉到高电平。总线上的最大器件数只受所要求的上升和下降时间的限制，上升和下降时间分别不能超过 300ns 和 1000ns。

图 18.2 典型 SMBus 配置



18.1 支持文档

假设读者熟悉或有条件阅读下列支持文档：

1. I²C 总线及使用（包含规范），菲力浦半导体。
2. I²C 总线规范—2.0 版，菲力浦半导体。
3. 系统管理总线规范—1.1 版，SBS

18.2 SMBus 协议

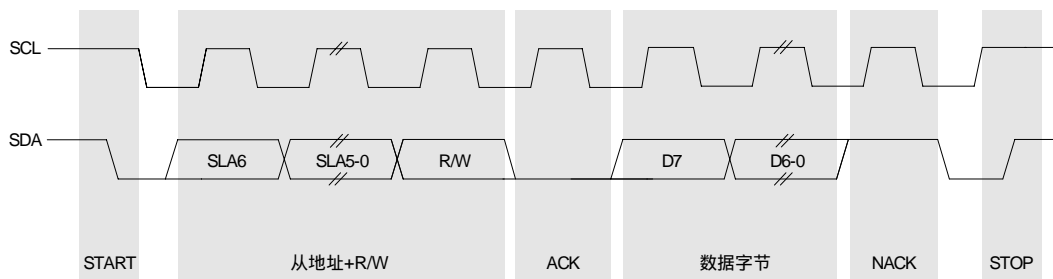
有两种可能的数据传输类型：从主发送器到所寻址的从接收器（写）和从被寻址的从发送器到主接收器（读）。这两种数据传输都由主器件启动，主器件还提供串行时钟。SMBus 接口可以工作在主方式或从方式。总线上可以有多个主器件。如果两个或多个主器件同时启动数据传输，仲裁机制将保证有一个主器件会赢得总线。注意：没有必要在一个系统中指定某个器件作为主器件；任何一个发送起始条件（START）和从器件地址的器件就成为该次数据传输的主器件。

一次典型的 SMBus 数据传输包括一个起始条件（START）、一个地址字节（位 7-1：7 位从地址；位 0：R/W 方向位）、一个或多个字节的数据和一个停止条件（STOP）。每个接收的字节（由一个主器件或从器件）都必须用 SCL 高电平期间的 SDA 低电平（见图 18.3）来确认（ACK）。如果接收器件不确认，则发送器件将读到一个“非确认”（NACK），这用 SCL 高电平期间的 SDA 高电平表示。

方向位占据地址字节的最低位。方向位被设置为逻辑 1 表示这是一个“读”（READ）操作，方向位为逻辑 0 表示这是一个“写”（WRITE）操作。

所有的数据传输都由主器件启动，可以寻址一个或多个目标从器件。主器件产生一个起始条件，然后发送地址和方向位。如果本次数据传输是一个从主器件到从器件的写操作，则主器件每发送一个数据字节后等待来自从器件的确认。如果是一个读操作，则由从器件发送数据并等待主器件的确认。在数据传输结束时，主器件产生一个停止条件，结束数据交换并释放总线。图 18.3 示出了一次典型的 SMBus 数据传输过程。

图 18.3 SMBus 数据传输



18.2.1 总线仲裁

一个主器件只能在总线空闲时启动一次传输。在一个停止条件之后或 SCL 和 SDA 保持高电平已经超过了指定时间，则总线是空闲的。两个或多个主器件可能在同一时刻产生起始条件。由于产生起始条件的器件并不知道其它器件也正想占用总线，所以使用仲裁机制迫使一个主器件放弃总线。这些主器件继续发送起始条件，直到其中一个主器件发送高电平而其它主器件在 SDA 上发送低电平。赢得总线的器件继续其数据传输过程，而未赢得总线的器件成为从器件。该仲裁机制是非破坏性的：总会有一个器件赢得总线，不会发生数据丢失。

18.2.2 时钟低电平扩展

SMBus 提供一种与 I²C 类似的同步机制，允许不同速度的器件共存于一个总线上。为了使低速从器件能与高速主器件通信，在传输期间采取低电平扩展。从器件可以保持 SCL 为低电平以扩展时钟低电平时间，这实际上相当于降低了串行时钟频率。

18.2.3 SCL 低电平超时

如果 SCL 线被总线上的从器件保持为低电平，则不能再进行通信，并且主器件也不能强制 SCL 为高电平来纠正这种错误情况。为了解决这一问题，SMBus 协议规定：参加一次数据传输的器件必须检查时钟低电平时间，若超过 25ms 则认为是“超时”。检测到超时条件的器件必须在 10ms 以内复位通信电路。

18.2.4 SCL 高电平（SMBus 空闲）超时

SMBus 标准规定：如果一个器件保持 SCL 和 SDA 线为高电平的时间超过 50 微秒，则认为总线处于空闲状态。如果一个 SMBus 器件正等待产生一个主起始条件，则该起始条件将在总线空闲超时之后立即产生。

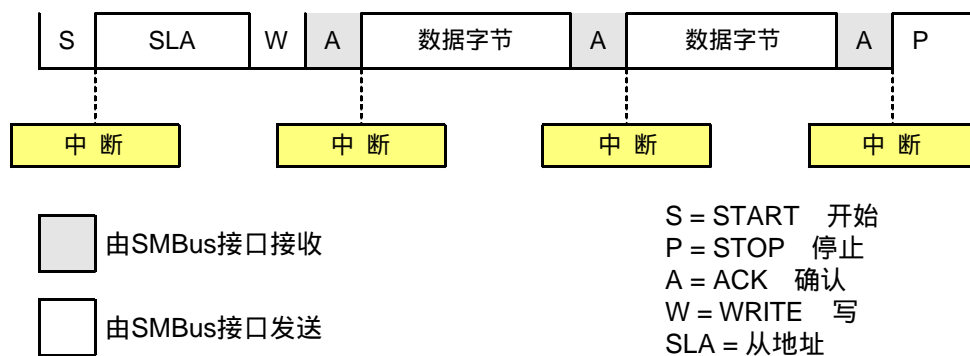
18.3 SMBus 数据传输方式

SMBus 接口可以被配置为工作在主方式和/或从方式。在某一时刻，它将工作在下述 4 种方式之一：主发送器、主接收器、从发送器或从接收器。下面以中断驱动的 SMBus0 应用为例来说明这四种工作方式；SMBus0 也可以工作在查询方式。

18.3.1 主发送器方式

在 SDA 上发送串行数据，在 SCL 上输出串行时钟。SMBus0 接口首先产生一个起始条件，然后发送含有目标从器件地址和数据方向位的第一个字节。在这种情况下数据方向位（R/W）应为逻辑 0，表示这是一个“写”操作。SMBus0 接口发送一个或多个字节的串行数据，并在每发送完一个字节后等待由从器件产生的确认信号（ACK）。最后，为了指示串行传输的结束，SMBus0 产生一个停止条件。

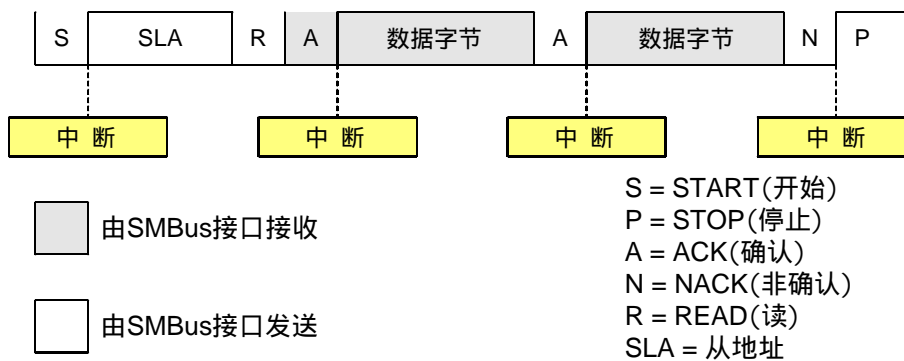
图 18.4 典型的主发送器时序



18.3.2 主接收器方式

在 SDA 上接收串行数据，在 SCL 上输出串行时钟。SMBus0 接口首先产生一个起始条件，然后发送含有目标从器件地址和数据方向位的第一个字节。在这种情况下数据方向位（R/W）应为逻辑 1，表示这是一个“读”操作。SMBus0 接口接收来自从器件的串行数据并在 SCL 上输出串行时钟。每收到一个字节后，SMBus0 接口根据寄存器 SMB0CN 中 AA 位的状态产生一个 ACK 或 NACK。最后，为了指示串行传输的结束，SMBus0 产生一个停止条件。

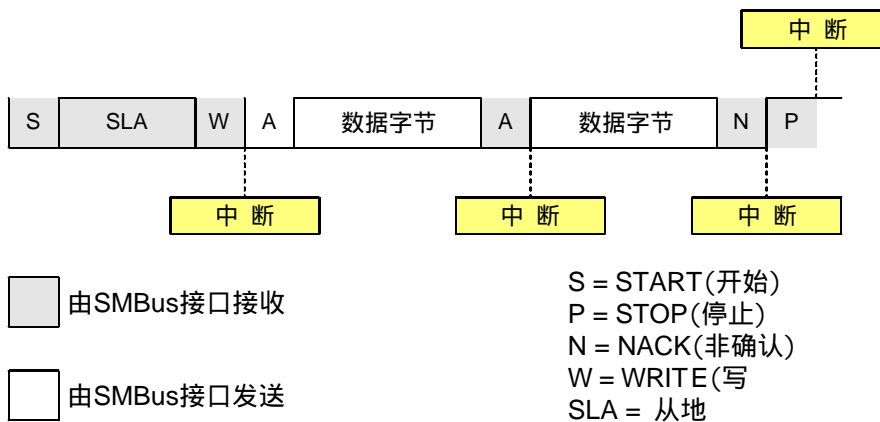
图 18.5 典型的主接收器时序



18.3.3 从发送器方式

在 SDA 上发送串行数据，在 SCL 上接收串行时钟。SMBus0 接口首先收到一个起始条件（START）和一个含有从地址和数据方向位的字节。如果收到的从地址与寄存器 SMB0ADR 中保存的地址一致，则 SMBus0 接口产生一个 ACK。如果收到全局呼叫地址（0x00）并且全局呼叫地址允许位（SMB0ADR.0）被设置为逻辑 1，则 SMBus0 接口也会发出 ACK。在这种情况下数据方向位（R/W）应为逻辑 1，表示这是一个“读”操作。SMBus0 接口在 SCL 上接收串行时钟并发送一个或多个字节的串行数据，每发送一个字节后等待由主器件发送的 ACK。在收到主器件发出的停止条件后，SMBus0 接口退出从方式。

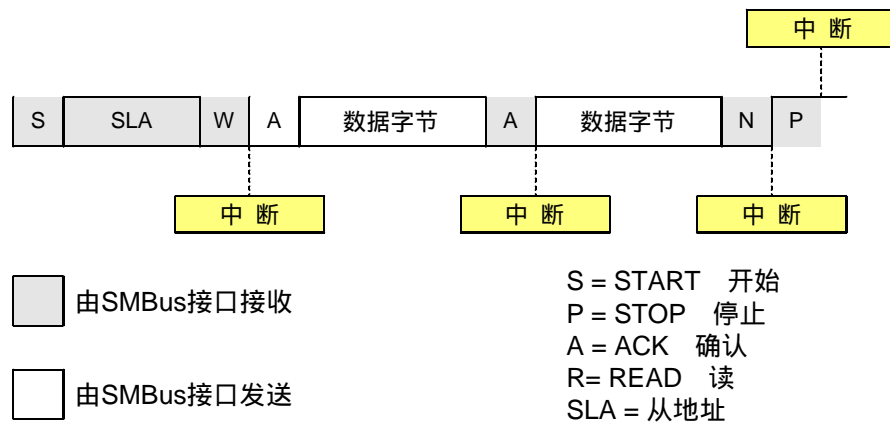
图 18.6 典型的从发送器时序



18.3.4 从接收器方式

在 SDA 上接收串行数据，在 SCL 上接收串行时钟。SMBus0 接口首先收到一个起始条件（START）和一个含有从地址和数据方向位的字节。如果收到的从地址与寄存器 SMB0ADR 中保存的地址一致，则 SMBus0 接口产生一个 ACK。如果收到全局呼叫地址（0x00）并且全局呼叫地址允许位（SMB0ADR.0）被设置为逻辑 1，则 SMBus0 接口也会发出 ACK。在这种情况下数据方向位（R/W）应为逻辑 0，表示这是一个“写”操作。SMBus0 接收一个或多个字节的串行数据；每收到一个字节后，SMBus0 接口根据寄存器 SMB0CN 中 AA 位的状态产生一个 ACK 或 NACK。在收到主器件发出的停止条件后，SMBus0 接口退出从接收器方式。

图 18.7 典型的从接收器时序



18.4 SMBus 特殊功能寄存器

对 SMBus 串行接口的访问和控制是通过 5 个特殊功能寄存器来实现的：控制寄存器 SMB0CN、时钟速率寄存器 SMB0CR、地址寄存器 SMB0ADR、数据寄存器 SMB0DAT 和状态寄存器 SMB0STA。下面对这 5 个与 SMBus 接口操作有关的特殊功能寄存器进行详细说明。

18.4.1 控制寄存器

SMBus 控制寄存器 SMB0CN 用于配置和控制 SMBus0 接口。该寄存器中的所有位都可以用软件读或写，有两个控制位受 SMBus0 硬件的影响。当发生一个有效的串行中断条件时，串行中断标志（SI，SMB0CN.3）被硬件设置为逻辑 1，该标志只能用软件清 0。当总线上出现一个停止条件时，停止标志（STO，SMB0CN.4）被硬件清 0。

设置 ENSMB 标志为逻辑 1 将使能 SMBus0 接口，把 ENSMB 标志清为逻辑 0 将禁止 SMBus0 接口并将其移出总线。对 ENSMB 标志瞬间清 0 后又重置为逻辑 1 将复位 SMBus0 通信逻辑。然而不应使用 ENSMB 从总线临时移出一个器件，因为这样做将使总线状态信息丢失。应使用确认标志（AA）从总线临时移出器件（见下面对 AA 标志的说明）。

设置起始标志（STA，SMB0CN.5）为逻辑 1 将使 SMBus0 工作于主方式。如果总线空闲，SMBus0 硬件将产生一个起始条件。如果总线不空闲，SMBus0 硬件将等待停止条件释放总线，然后根据 SMB0CR 的值在经过 5 微秒的延时后产生一个起始条件。（根据 SMBus 协议，如果总线处于等待状态的时间超过 50 微秒而没有检测到停止条件，SMBus0 接口可以认为总线是

空闲的。)如果 STA 被设置为逻辑 1, 而此时 SMBus 处于主方式并且已经发送了一个或多个字节, 则将产生一个重复起始条件。为保证操作正确, 应在对 STA 位置 1 之前, 将 STO 标志清 0。

当停止标志 (STO, SMB0CN.4) 被设置为逻辑 1, 而此时 SMBus0 接口处于主方式, 则接口将在 SMBus0 上产生一个停止条件。在从方式, STO 标志可以用于从一个错误条件恢复。在这种情况下, SMBus0 上不产生停止条件, 但 SMBus0 硬件的表现就象是收到了一个停止条件并进入“未寻址”的从接收器状态。注意, 这种模拟的停止条件并不能导致释放总线。总线将保持忙状态直到出现停止条件或发生总线空闲超时。当检测到总线上的停止条件时, SMBus0 硬件自动将 STO 标志清为逻辑 0。

当 SMBus0 接口进入到 27 个可能状态之一时, 串行中断标志 (SI, SMB0CN.3) 被硬件置为逻辑 1。如果 SMBus0 接口的中断被允许, 在 SI 标志置 1 时将产生一个中断请求。SI 标志必须用软件清除。

注意: 如果 SI 标志被置为逻辑 1 时 SCL 线为低电平, 则串行时钟的低电平时间将被延长, 串行传输暂时停止, 直到 SI 被清为逻辑 0 为止。SCL 的高电平不受 SI 标志设置值的影响。

确认标志 (AA, SMB0CN.2) 用于在 SCL 线的应答周期中设置 SDA 线的电平。如果器件被寻址, 设置 AA 标志为逻辑 1 将在应答周期发送一个确认位 (SDA 上的低电平)。设置 AA 标志为逻辑 0 将在应答周期发送一个非确认位 (SDA 上的高电平)。在从方式下, 发送完一个字节后可以通过清除 AA 标志使从器件暂时脱离总线。这样, 从器件自身地址或全局呼叫地址都将被忽略。为了恢复总线操作, 必须将 AA 标志重新设置为 1 以允许从地址被识别。

设置 SMBus0 空闲定时器允许位 (FTE, SMB0CN.1) 为逻辑 1 将使能 SMB0CR 中的定时器。当 SCL 变高时, SMB0CR 的定时器向上计数。定时器溢出指示总线空闲超时: 如果 SMBus0 等待产生一个起始条件, 则将在超时发生后进行。总线空闲周期应小于 50 微秒 (见图 18.19, SMBus0 时钟速率寄存器)。

当 SMB0CN 中的 TOE 位被设置为逻辑 1 时, 定时器 3 用于检测 SCL 低电平超时。如果定时器 3 被使能, 则在 SCL 为高电平时定时器 3 被强制重载, SCL 为低电平时使定时器 3 开始计数。当定时器 3 被使能并且溢出周期被编程为 25ms (且 TOE 置 1) 时, 定时器 3 溢出表示发生了 SCL 低电平超时; 定时器 3 中断服务程序可以用于在发生 SCL 低电平超时的情况下复位 SMBus0 通信逻辑。

图 18.8 SMB0CN: SMBus0 控制寄存器

| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
|------|--|-------|-----|-----|-----|-----|-----|--------------|
| | BUSY | ENSMB | STA | STO | SI | AA | FTE | TOE |
| | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) |
| | 复位值 00000000 | | | | | | | |
| | SFR地址: 0xC0 | | | | | | | |
| 位 7: | BUSY: 忙状态标志 0: SMBus0 空闲 1: SMBus0 忙 | | | | | | | |
| 位 6: | ENSMB: SMBus0 允许 该位允许/禁止 SMBus0 串行接口 0: 禁止 SMBus 1: 允许 SMBus | | | | | | | |
| 位 5: | STA: SMBus0 起始标志 0: 不发送起始条件。 1: 当作为主器件时, 若总线空闲, 则发送出一个起始条件。(如果总线不空闲, 在收到停止条件后再发送起始条件。) 如果 STA 被置 1, 而此时已经发送或接收了一个或多个字节并且没有收到停止条件, 则发送一个重复起始条件。为保证操作正确, 应在对 STA 位置 1 之前, 将 STO 标志清 0。 | | | | | | | |
| 位 4: | STO: SMBus0 停止标志 0: 不发送停止条件。 1: 将 STO 置为逻辑 1 将发送一个停止条件。当收到停止条件时, 硬件将 STO 清为逻辑 0。如果 STA 和 STO 都被置位, 则发送一个停止条件后再发送一个起始条件。在从方式, 置位 STO 标志将导致 SMBus 的行为象收到了停止条件一样。 | | | | | | | |
| 位 3: | SI: SMBus0 串行中断标志 当 SMBus0 进入 27 种状态之一时该位被硬件置位。(状态码 0xF8 不使 SI 置位。) 当 SI 中断被允许时, 该位置 1 将导致 CPU 转向 SMBus0 中断服务程序。该位不能被硬件自动清 0, 必须用软件清除。 | | | | | | | |
| 位 2: | AA: SMBus0 确认标志 该位定义在 SCL 线应答周期内返回的应答类型。 0: 在应答周期内返回“非确认”(SDA 线高电平)。 1: 在应答周期内返回“确认”(SDA 线低电平)。 | | | | | | | |
| 位 1: | FTE: SMBus0 空闲定时器允许位 0: 无 SCL 高电平超时。 1: 当 SCL 高电平时间超过由 SMB0CR 规定的极限值时发生超时。 | | | | | | | |
| 位 0: | TOE: SMBus0 超时允许位 0: 无 SCL 低电平超时。 1: 当 SCL 处于低电平的时间超过由定时器 3 (如果被允许) 定义的极限值时发生超时。 | | | | | | | |

18.4.2 时钟速率寄存器

图 18.9 SMB0CR: SMBus0 时钟速率寄存器

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| | | | | | | | | | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | | SFR地址: 0xCF |

位 7-0: SMB0CR.[7:0]: SMBus0 时钟速率预设值
SMB0CR 时钟速率寄存器用于控制主方式下串行时钟 SCL 的频率。存储在 SMB0CR 寄存器中的 8 位字预装一个专用的 8 位定时器。该定时器向上计数, 当计满回到 0x00 时 SCL 改变逻辑状态。SCL 时钟的周期由下面的方程给出, 其中 *SMB0CR* 是 SMB0CR 寄存器中的 8 位无符号数值。

$$f_{SCL} \cong \frac{1}{2} \times \frac{SYSCLK}{(256 - SMB0CR) + 2.5}$$

使用与上式相同的 SMB0CR 值, 总线空闲超时周期由下式给出:

$$T_{BFT} = 10 \times \frac{(256 - SMB0CR) + 1}{SYSCLK}$$

SMB0CR 的取值范围为 $0x00 \leq SMB0CR \leq 0xFE$ 。

18.4.3 数据寄存器

SMBus0 数据寄存器 SMB0DAT 保存要发送或刚接收的串行数据字节。在 SI 被置为逻辑 1 时软件可以读或写数据寄存器; 当 SMBus0 被允许并且 SI 标志被清为逻辑 0 时软件不应访问 SMB0DAT 寄存器, 因为硬件可能正在对该寄存器中的数据字节进行移入或移出操作。

SMB0DAT 中的数据总是先移出 MSB。在每收到一个字节后, 接收数据的第一位位于 SMB0DAT 的 MSB。在数据被移出的同时, 总线上的数据被移入。所以 SMB0DAT 中总是保存最后出现在总线上的数据字节。因此在竞争失败后, 从主发送器转为从接收器时 SMB0DAT 中的数据保持正确。

图 18.10 SMB0DAT: SMBus0 数据寄存器

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| | | | | | | | | | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | | SFR地址: 0xC2 |

位 7-0: SMB0DAT: SMBus0 数据
SMB0DAT 寄存器保存要发送到 SMBus0 串行接口上的一个数据字节, 或刚从 SMBus0 串行接口接收到的一个字节。一旦 SI 串行中断标志 (SMB0CN.3) 被置为逻辑 1, CPU 即可读或写该寄存器。当 SI 标志位不为 1 时, 系统可能正在移入/移出数据, 此时 CPU 不应访问该寄存器。

18.4.4 地址寄存器

地址寄存器 SMB0ADR 保存 SMBus0 接口的从地址。在从方式，该寄存器的高 7 位是从地址，最低位（位 0）用于允许全局呼叫地址（0x00）识别。如果该位被设置为逻辑 1，则允许识别全局呼叫地址。否则，全局呼叫地址被忽略。当 SMBus 硬件工作在主方式时，该寄存器的内容被忽略。

图 18.11 SMB0ADR: SMBus0 地址寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|------|------|------|------|------|------|-----|----------------|
| SLV6 | SLV5 | SLV4 | SLV3 | SLV2 | SLV1 | SLV0 | GC | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xC3 |
| <p>位 7-1: SLV6-SLV0: SMBus0 从地址 这些位用于存放 7 位从地址，当器件工作在从发送器或从接收器方式时，SMBus0 将应答该地址。SLV6 是地址的最高位，对应从 SMBus0 收到的地址字节的第一位。</p> <p>位 0: GC: 全局呼叫地址允许 该位用于允许全局呼叫地址（0x00）识别。 0: 忽略全局呼叫地址。 1: 识别全局呼叫地址。</p> | | | | | | | | |

18.4.5 状态寄存器

状态寄存器 SMB0STA 保存一个 8 位的状态码，用于指示 SMBus0 接口的当前状态。共有 28 个可能的 SMBus0 状态，每个状态有一个唯一的状态码与之对应。状态码的高 5 位是可变的，而一个有效状态码的低 3 位固定为 0（当 SI=1 时），因此所有有效的状态码都是 8 的整数倍。这使我们可以很容易地在软件中用状态码作为转移到正确的中断服务程序的索引（允许 8 字节的代码对状态提供服务或转到更长的中断服务程序）。

对于用户软件而言，SMB0STA 的内容只在 SI 标志为逻辑 1 时有定义。软件不应向 SMB0STA 寄存器写入；如果写入，将会产生不确定的结果。表 18.1 列出了 28 个 SMBus0 状态和对应的状态码。

图 18.12 SMB0STA: SMBus 状态寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|------|------|------|------|------|------|------|----------------|
| STA7 | STA6 | STA5 | STA4 | STA3 | STA2 | STA1 | STA0 | 11111000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xC1 |
| <p>位 7-3: STA7-STA3: SMBus0 状态代码 这些位含有 SMBus0 状态代码。共有 28 个可能的状态码，每个状态码对应一个 SMBus 状态。在 SI 标志（SMB0CN.3）置位时，SMB0STA 中的状态码有效。当 SI 标志为逻辑 0 时，SMB0STA 中的内容无定义。任何时候向 SMB0STA 寄存器写入将导致不确定的结果。</p> <p>位 2-0: STA2-STA0: 当 SI 标志位为逻辑 1 时，这三个 SMB0STA 最低位的读出值总是为逻辑 0。</p> | | | | | | | | |

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

表 18.1. SMB0STA 状态码和状态

| 方式 | 状态码 | SMBus 状态 | 典型操作 |
|---------------|------|--|--|
| 主发送器/ 主接收器 | 0x08 | 起始条件已发出 | 将从地址+R/W 装入到 SMB0DAT。清 '0' STA。 |
| | 0x10 | 重复起始条件已发出 | 将从地址+R/W 装入到 SMB0DAT。清 '0' STA。 |
| 主发送器 | 0x18 | 从地址+W 已发出。收到 ACK。 | 将要发送的数据装入到 SMB0DAT。 |
| | 0x20 | 从地址+W 已发出。收到 NACK。 | 确认查询重试。置位 STO+STA。 |
| | 0x28 | 数据字节已发出。收到 ACK。 | 1) 将下一字节装入到 SMB0DAT, 或 2) 置位 STO, 或 3) 置位 STO, 然后置位 STA 以发送重复起始条件。 |
| | 0x30 | 数据字节已发出。收到 NACK。 | 1) 重试传输或 2) 置位 STO。 |
| | 0x38 | 竞争失败 | 保存当前数据。 |
| 主接收器 | 0x40 | 从地址+R 已发出。收到 ACK。 | 如果只收到一个字节, 清 AA 位 (收到字节后发送 NACK)。等待接收数据。 |
| | 0x48 | 从地址+R 已发出。收到 NACK。 | 确认查询重试。置位 STO+STA |
| | 0x50 | 数据字节收到。ACK 已发出 | 读 SMB0DAT。等待下一字节。 如果下一字节是最后字节, 清除 AA。 |
| | 0x58 | 数据字节收到。NACK 已发出 | 置位 STO。 |
| 从接收器 | 0x60 | 收到自身的从地址+W。ACK 已发出。 | 等待数据 |
| | 0x68 | 在作为主器件发送 SLA+R/W 时竞争失败。收到自身地址+W。ACK 已发出。 | 保存当前数据以备总线空闲时重试。 |
| | 0x70 | 收到全局呼叫地址。ACK 已发出。 | 等待数据 |
| | 0x78 | 在作为主器件发送 SLA+R/W 时竞争失败。收到全局呼叫地址+W。ACK 已发出。 | 保存当前数据以备总线空闲时重试。 |
| | 0x80 | 收到数据字节。ACK 已发出。 | 读 SMB0DAT。等待下一字节或停止条件。 |
| | 0x88 | 收到数据字节。NACK 已发出。 | 置位 STO 以复位 SMBus |
| | 0x90 | 在全局呼叫地址之后收到数据字节。ACK 已发出。 | 读 SMB0DAT。等待下一字节或停止条件。 |
| | 0x98 | 在全局呼叫地址之后收到数据字节。NACK 已发出。 | 置位 STO 以复位 SMBus |
| | 0xA0 | 收到停止条件或重复起始条件。 | 不需操作 |

表 18.1 SMB0STA 状态码和状态 (续)

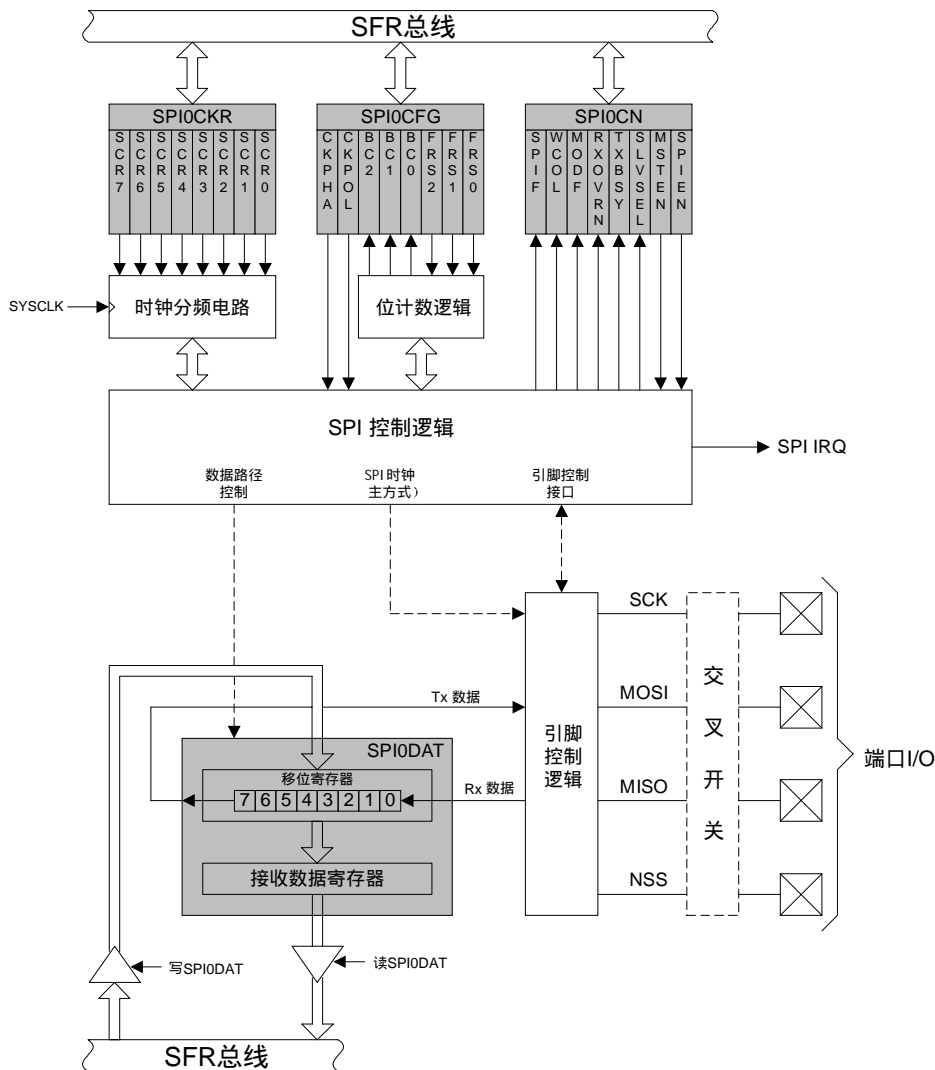
| | | | |
|------|------|--|-------------------------------------|
| 从发送器 | 0xA8 | 收到自己的从地址+R。ACK 已发出。 | 将要发送的数据装入到 SMB0DAT。 |
| | 0xB0 | 在作为主器件发送 SLA+R/W 时竞争失败。收到自身地址+R。ACK 已发出。 | 保存当前数据以备总线空闲时重试。将要发送的数据装入到 SMB0DAT。 |
| | 0xB8 | 数据字节已发送。收到 ACK。 | 将要发送的数据装入到 SMB0DAT。 |
| | 0xC0 | 数据字节已发送。收到 NACK。 | 等待停止条件 |
| | 0xC8 | 最后一个字节已发送 (AA=0)。收到 ACK。 | 置位 STO 以复位 SMBus |
| 从器件 | 0xD0 | SCL 时钟高电平定时器超时(根据 SMB0CR)。 | 置位 STO 以复位 SMBus |
| 所有方式 | 0x00 | 总线错误(非法起始条件或停止条件)。 | 置位 STO 以复位 SMBus |
| | 0xF8 | 空闲状态。 | 该状态不置位 SI |

19. 串行外设接口总线

串行外设接口（SPI0）提供访问一个 4 线、全双工串行总线的能力。SPI0 支持在同一总线上将多个从器件连接到一个主器件。一个独立的从选择信号（NSS）用于选择一个从器件并允许主器件和所选从器件之间进行数据传输。同一总线上可以有多个主器件。当两个或多个主器件试图同时进行数据传输时，系统提供了冲突检测功能。SPI0 可以工作在主方式或从方式。当 SPI 被配置为主器件时，最大数据传输率（位/秒）是系统时钟频率的二分之一。

当 SPI 被配置为从器件时，如果主器件与系统时钟同步发出 SCK、NSS 和串行输入数据，则全双工操作时的最大数据传输率（位/秒）是系统时钟频率的十分之一。如果主器件发出的 SCK、NSS 及串行输入数据不同步，则最大数据传输率（位/秒）必须小于系统时钟频率的十分之一。在主器件只想发送数据到从器件而不需要接收从器件发出的数据（即半双工操作）这一特殊情况下，SPI 从器件接收数据时的最大数据传输率（位/秒）是系统时钟频率的四分之一。这是在假设由主器件与系统时钟同步发出 SCK、NSS 和串行输入数据的情况下。

图 19.1 SPI 原理框图



19.1 信号说明

下面介绍 SPI 所使用的 4 个信号（MOSI、MISO、SCK、NSS）。

19.1.1 主输出、从输入（MOSI）

主出从入（MOSI）信号是主器件的输出和从器件的输入，用于从主器件到从器件的串行数据传输。当 SPI0 作为主器件时，该信号是输出；当 SPI0 作为从器件时，该信号是输入。数据传输时最高位在先。

19.1.2 主输入、从输出（MISO）

主入从出（MISO）信号是从器件的输出和主器件的输入，用于从从器件到主器件的串行数据传输。当 SPI0 作为主器件时，该信号是输入；当 SPI0 作为从器件时，该信号是输出。数据传输时最高位在先。当 SPI 从器件未被选中时，它将 MISO 引脚置于高阻状态。

19.1.3 串行时钟（SCK）

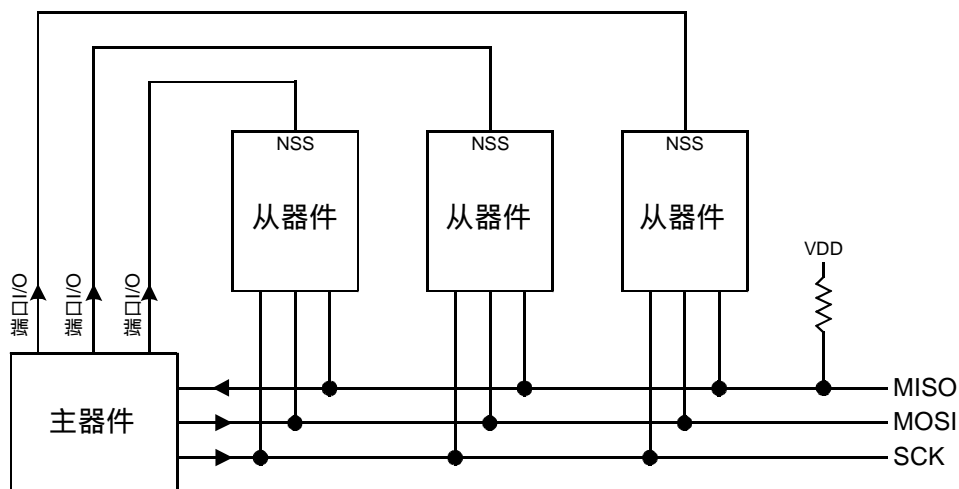
串行时钟（SCK）信号是主器件的输出和从器件的输入，用于同步主器件和从器件之间在 MOSI 和 MISO 线上的串行数据传输。当 SPI0 作为主器件时产生该信号。

19.1.4 从选择（NSS）

从选择（NSS）信号是一个输入信号，主器件用它来选择处于从方式的 SPI0 器件，在器件为主方式时用于禁止 SPI0。注意：NSS 信号总是作为 SPI0 的输入；当 SPI0 工作在主方式时，从选择信号必须是通用端口 I/O 引脚的输出。图 19.2 给出了一种典型配置。有关通用端口配置的信息见“17.1 端口 0-3 和优先权交叉开关译码器”。

当 SPI0 工作于从方式时，NSS 信号必须被拉为低电平以启动一次数据传输；当 NSS 被释放为高电平时，SPI0 将退出从方式。注意：在 NSS 变为高电平之前，接收的数据不会被锁存到接收缓冲器。对于多字节传输，在 SPI0 器件每接收一个字节后 NSS 必须被释放为高电平至少 4 个系统时钟。

图 19.2 典型 SPI 连接

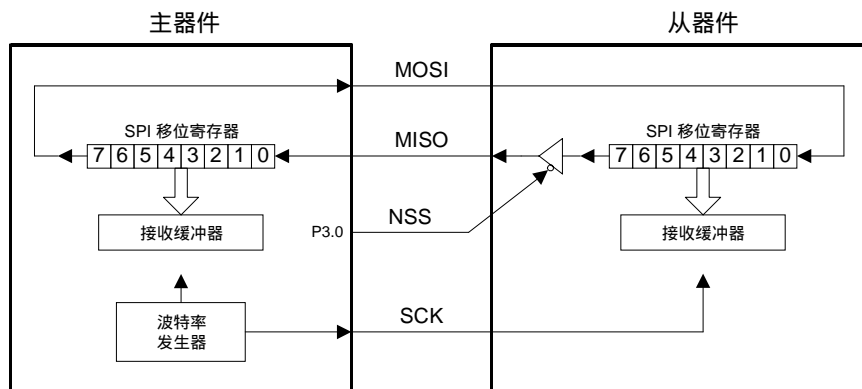


19.2 SPI0 操作

只有 SPI 主器件能启动数据传输。通过将主允许标志 (MSTEN, SPI0CN.1) 置 1 使 SPI0 处于主方式。当处于主方式时, 向 SPI0 数据寄存器 (SPI0DAT) 写入一个字节将启动一次数据传输。SPI0 主器件立即在 MOSI 线上串行移出数据, 同时在 SCK 上提供串行时钟。在传输结束后 SPIF (SPI0CN.7) 标志被置为逻辑 1。如果中断被允许, 在 SPIF 标志置位时将产生一个中断请求。SPI 主器件可以被配置为在一次传输操作中移入/移出 1 到 8 位数据, 以适应具有不同字长度的从器件。SPI0 配置寄存器中的 SPIFRS 位 (SPI0CFG[2:0]) 用于选择一次传输操作中移入/移出的位数。

在全双工操作中, SPI 主器件在 MOSI 线上向从器件发送数据, 被寻址的从器件可以同时 在 MISO 线上向主器件发送其移位寄存器中的内容。所接收到的来自从器件的数据替换主器件数据寄存器中的数据。因此, SPIF 标志既作为发送完成标志又作为接收数据准备好标志。两个方向上的数据传输由主器件产生的串行时钟同步。图 19.3 描述了一个 SPI 主器件和一个 SPI 从器件的全双工操作。

图 19.3 全双工操作



当 SPI0 被允许而未被配置为主器件时, 它将作为从器件工作。另一个 SPI 主器件通过将其 NSS 信号驱动为低电平启动一次数据传输。主器件用其串行时钟将移位寄存器中的数据移出到 MOSI 引脚。在一次数据传输结束后 (当 NSS 信号变为高电平时), SPIF 标志被设置为逻辑 1。注意, 在 NSS 的上升沿过后, 接收缓冲器将总是含有从器件移位寄存器中的最后 8 位。从器件可以通过写 SPI0 数据寄存器来为下一次数据传输装载它的移位寄存器。从器件必须在主器件开始下一次数据传输之前至少一个 SPI 串行时钟周期写数据寄存器。否则, 已经位于从器件移位寄存器中的数据字节将被发送。注意, NSS 信号必须在每次字节传输的第一个 SCK 有效沿之前至少两个系统时钟被驱动到低电平。

SPI0 数据寄存器对读操作而言是双缓冲的, 但写操作时不是。如果在一次数据传输期间试图写 SPI0DAT, 则 WCOL 标志 (SPI0CN.6) 将被设置为逻辑 1, 写操作被忽略, 而当前的数据传输不受影响。系统控制器读 SPI0 数据寄存器时, 实际上是读接收缓冲器。在任何时刻如果 SPI0 从器件检测到一个 NSS 上升沿, 而接收缓冲器中仍保存着前一次传输未被读取的数据, 则发生接收溢出, RXOVRN 标志 (SPI0CN.4) 被设置为逻辑 1。新数据不被传送到接收缓冲器, 允许前面接收的数据等待读取, 引起溢出的数据字节丢失。

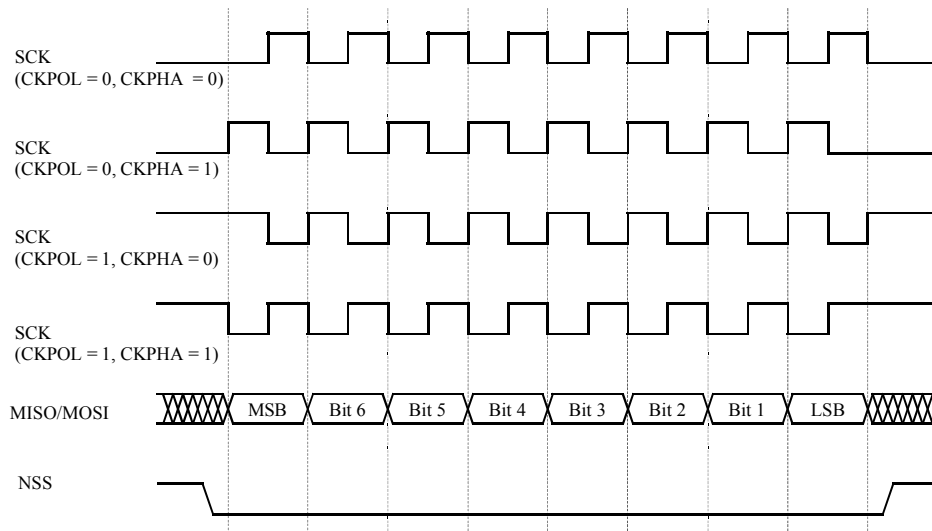
多个主器件可以共存于同一总线。当 SPI0 被配置为主器件 (MSTEN=1) 而其从选择信号 NSS 被拉为低电平时, 方式错误标志 (MODF, SPI0CN.5) 被设置为逻辑 1。当方式错误标志被置 1 时, SPI 控制寄存器中的 MSTEN 和 SPIEN 位被硬件清除, 从而将 SPI 模块置于“离线”状态。在一个多主环境, 系统控制器应检查 SLVSEL 标志 (SPI0CN.2) 的状态, 以保证在置 ‘1’ MSTEN 位和启动一次数据传输之前总线是空闲的。

19.3 串行时钟时序

如图 19.4 所示, 使用 SPI0 配置寄存器 (SPI0CFG) 中的时钟控制选择位可以在串行时钟相位和极性的 4 种组合中选择其一。CKPHA 位 (SPI0CFG.7) 选择两种时钟相位 (锁存数据所用的边沿) 中的一种。CKPOL 位 (SPI0CFG.6) 在高电平有效和低电平有效的时钟之间选择。主器件和从器件必须被配置为使用相同的时钟相位和极性。注意: 在改变时钟相位和极性期间应禁止 SPI0 (通过清除 SPIEN 位, SPI0CN.0)。

图 19.7 所示的 SP0I 时钟速率寄存器 (SPI0CKR) 控制主方式的串行时钟频率。当工作于从方式时该寄存器被忽略。

图 19.4 数据/时钟时序图



19.4 SPI 特殊功能寄存器

对 SPI0 的访问和控制是通过系统控制器中的 4 个特殊功能寄存器实现的：控制寄存器 SPI0CN、数据寄存器 SPI0DAT、配置寄存器 SPI0CFG 和时钟频率寄存器 SPI0CKR。下面将介绍这 4 个与 SPI0 总线操作有关的特殊功能寄存器。

图 19.5 SPI0CFG：SPI0 配置寄存器

| | | | | | | | | |
|-------|-------|-----|-----|-----|---------|---------|---------|----------------|
| R/W | R/W | R | R | R | R/W | R/W | R/W | 复位值 |
| CKPHA | CKPOL | BC2 | BC1 | BC0 | SPIFRS2 | SPIFRS1 | SPIFRS0 | 00000111 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址： 0x9A |

位 7: CKPHA: SPI0 时钟相位。
该位控制 SPI0 时钟的相位。
0: 在 SCK 周期的第一个边沿采样数据。
1: 在 SCK 周期的第二个边沿采样数据。

位 6: CKPOL: SPI0 时钟极性
该位控制 SPI0 时钟的极性。
0: SCK 在空闲状态时处于低电平。
1: SCK 在空闲状态时处于高电平。

位 5-3: BC2-BC0: SPI0 位计数
指示发送到了 SPI 字的哪一位。

| BC2—BC0 | | | 已发送的位 |
|---------|---|---|-----------|
| 0 | 0 | 0 | 位 0 (LSB) |
| 0 | 0 | 1 | 位 1 |
| 0 | 1 | 0 | 位 2 |
| 0 | 1 | 1 | 位 3 |
| 1 | 0 | 0 | 位 4 |
| 1 | 0 | 1 | 位 5 |
| 1 | 1 | 0 | 位 6 |
| 1 | 1 | 1 | 位 7 (MSB) |

位 2-0: SPIFRS2-SPIFRS0: SPI0 帧长度
这三位决定在主方式数据传输期间 SPI0 移位寄存器移入/出的位数。它们在从方式时被忽略。

| SPIFRS | | | 移位数 |
|--------|---|---|-----|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 3 |
| 0 | 1 | 1 | 4 |
| 1 | 0 | 0 | 5 |
| 1 | 0 | 1 | 6 |
| 1 | 1 | 0 | 7 |
| 1 | 1 | 1 | 8 |

图 19.6 SPI0CN: SPI0 控制寄存器

| R/W | R/W | R/W | R/W | R | R | R/W | R/W | 复位值 |
|---|------|------|--------|-------|--------|-------|--------------|----------------|
| SPIF | WCOL | MODF | RXOVRN | TXBSY | SLVSEL | MSTEN | SPIEN | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0xF8 |
| <p>位 7: SPIF: SPI0 中断标志 该位在数据传输结束后被硬件置为逻辑 1。如果中断被允许, 置 1 该位将会使 CPU 转到 SPI0 中断处理服务程序。该位不能被硬件自动清 0, 必须用软件清 0。</p> <p>位 6: WCOL: 写冲突标志 该位由硬件置为逻辑 1 (并产生一个 SPI0 中断), 表示数据传送期间对 SPI0 数据寄存器进行了写操作。该位必须用软件清 0。</p> <p>位 5: MODF: 方式错误标志 当检测到主方式冲突 (NSS 为低电平, MSTEN=1) 时, 该位由硬件置为逻辑 1 (并产生一个 SPI0 中断)。该位不能被硬件自动清 0, 必须用软件清 0。</p> <p>位 4: RXOVRN: 接收溢出标志 当前传输的最后一位已经移入 SPI0 移位寄存器, 而接收缓冲器中仍保存着前一次传输未被读取的数据时该位由硬件置为逻辑 1 (并产生一个 SPI0 中断)。该位不会被硬件自动清 0, 必须用软件清 0。</p> <p>位 3: TXBSY: 发送忙标志 当一个主方式传输正在进行时, 该位被硬件置为逻辑 1。在传输结束后由硬件清 0。</p> <p>位 2: SLVSEL: 从选择标志 该位在 NSS 引脚为低电平时置 1, 说明它被允许为从方式。它在 NSS 变为高电平时清 0 (从方式被禁止)</p> <p>位 1: MSTEN: 主方式允许位 0: 禁止主方式。以从方式操作。 1: 允许主方式。以主方式操作。</p> <p>位 0: SPIEN: SPI0 允许位 该位允许 / 禁止 SPI0。 0: 禁止 SPI0 1: 允许 SPI0</p> | | | | | | | | |

图 19.7 SPI0CKR: SPI0 时钟速率寄存器

| | | | | | | | | |
|------|------|------|------|------|------|------|------|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| SCR7 | SCR6 | SCR5 | SCR4 | SCR3 | SCR2 | SCR1 | SCR0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x9D |

位 7-0: SCR7-SCR0: SPI0 时钟频率

当 SPI0 模块被配置为工作于主方式时, 这些位决定 SCK 输出的频率。SCK 时钟频率是从系统时钟分频得到的, 由下面的方程给出, 其中: *SYSCLK* 是系统时钟频率, *SPI0CKR* 是 SPI0CKR 寄存器中的 8 位值。

$$f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR + 1)}$$

(0 ≤ SPI0CKR ≤ 255)

例如: 如果 *SYSCLK* = 2MHz, *SPI0CKR* = 0x04, 则

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

$$f_{SCK} = 200kHz$$

图 19.8 SPI0DAT: SPI0 数据寄存器

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0x9B |

位 7-0: SPI0DAT: SPI0 发送和接收数据寄存器。

SPI0DAT 寄存器用于发送和接收 SPI0 数据。在主方式下, 向 SPI0DAT 写入数据时, 数据立即进入移位寄存器并启动发送。读 SPI0DAT 返回接收缓冲器的内容。

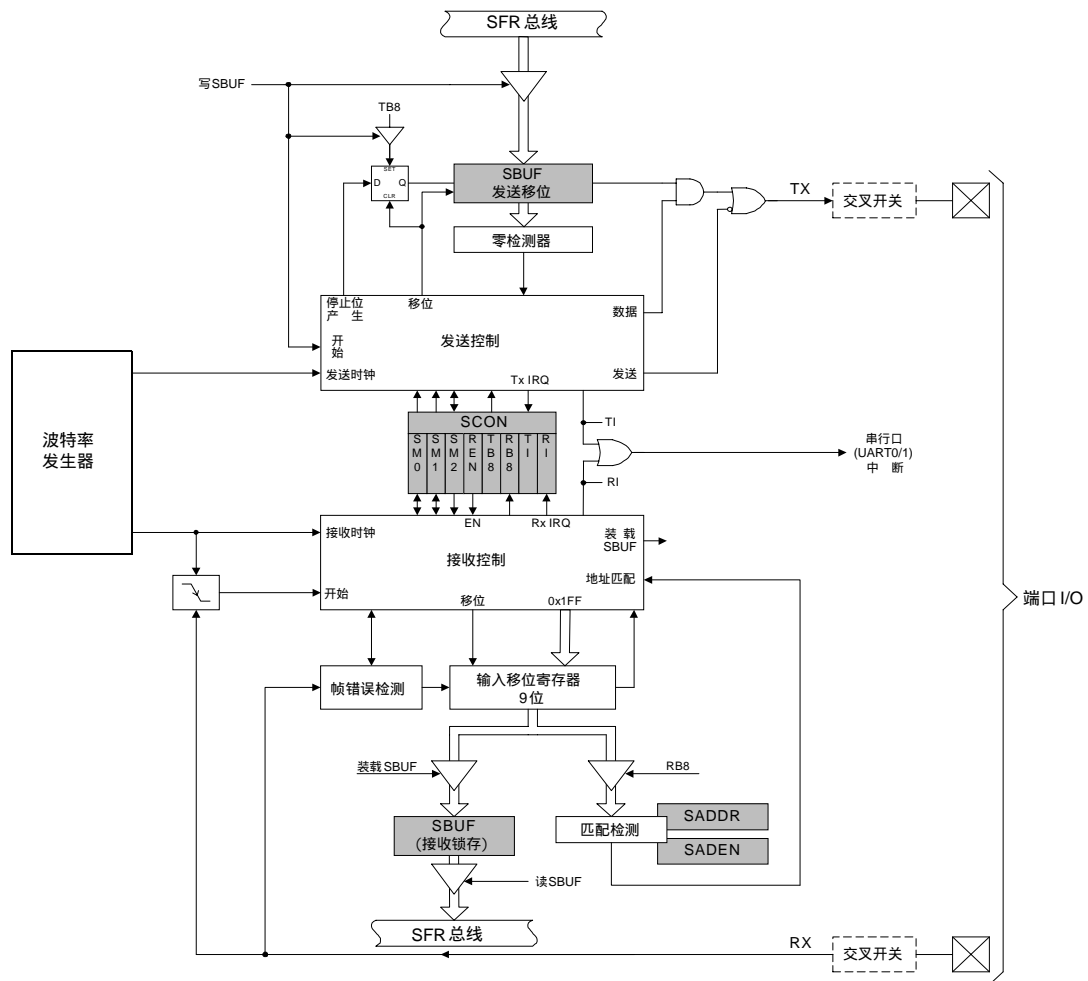
20. UART0

UART0 是一个具有帧错误检测和地址识别硬件的增强型串行口。UART0 可以工作在全双工异步方式或半双工同步方式，并且支持多处理器通信。接收数据被暂存于一个保持寄存器中，这就允许 UART0 在软件尚未读取前一个数据字节的情况下开始接收第二个输入数据字节。一个接收覆盖位用于指示新的接收数据已被锁存到接收缓冲器而前一个接收数据尚未被读取。

对 UART0 的控制和访问是通过相关的特殊功能寄存器即串行控制寄存器 (SCON) 和串行数据缓冲器 (SBUF0) 来实现的。一个 SBUF0 地址可以访问发送寄存器和接收寄存器。读操作将自动访问接收寄存器，而写操作自动访问发送寄存器。

UART0 可以工作在查询或中断方式。UART0 有两个中断源：一个发送中断标志 TI0 (SCON0.1) (数据字节发送结束时置位) 和一个接收中断标志 RI0 (SCON0.0) (接收完一个数据字节后置位)。当 CPU 转向中断服务程序时硬件不清除 UART0 中断标志，中断标志必须由软件清除。这就允许软件查询 UART0 中断的原因 (发送完成或接收完成)。

图 20.1 UART0 原理框图



20.1 UART0 工作方式

UART0 提供四种工作方式（一种同步方式和三种异步方式），通过设置 SCON0 寄存器中的配置位选择。这四种方式提供不同的波特率和通信协议。下面的表 20.1 概述了这四种方式。

表 20.1 UART0 工作方式

| 方式 | 同步性 | 波特率时钟 | 数据位 | 起始/停止位 |
|----|-----|-----------------------|-----|-------------|
| 0 | 同步 | SYSCLK/12 | 8 | 无 |
| 1 | 异步 | 定时器 1 或定时器 2 溢出 | 8 | 一个起始位，一个停止位 |
| 2 | 异步 | SYSCLK/32 或 SYSCLK/64 | 9 | 一个起始位，一个停止位 |
| 3 | 异步 | 定时器 1 或定时器 2 溢出 | 9 | 一个起始位，一个停止位 |

20.1.1 方式 0：同步方式

方式 0 提供同步、半双工通信。在 RX0 引脚上发送和接收数据，TX0 引脚提供发送和接收的移位时钟。MCU 必须是主器件，因为它要为两个方向的数据传输产生移位时钟（见图 20.2 中的连接图）。

执行一条写 SBUF0 寄存器的指令时开始数据发送。发送/接收的数据为 8 位，LSB 在先（见图 20.3 中的时序图），在第 8 个位时间结束后发送中断标志 TI0（SCON0.1）置位。当接收允许位 REN0（SCON0.4）被设置为逻辑 1 并且接收中断标志 RI0（SCON0.0）被清 0 时开始数据接收。在第 8 位被移入后一个周期 RI0 标志置位，接收过程停止，直到软件清除 RI0 位。如果中断被允许，在 TI0 或 RI0 置位后将发生一次中断。

方式 0 的波特率是系统时钟频率/12。在方式 0，RX0 被强制为漏极开路方式，通常需要外接一个上拉电阻。

图 20.2 UART0 方式 0 连接

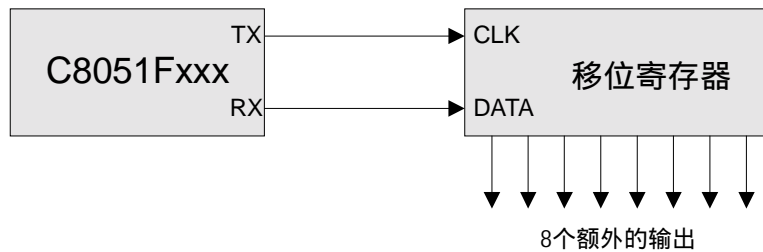
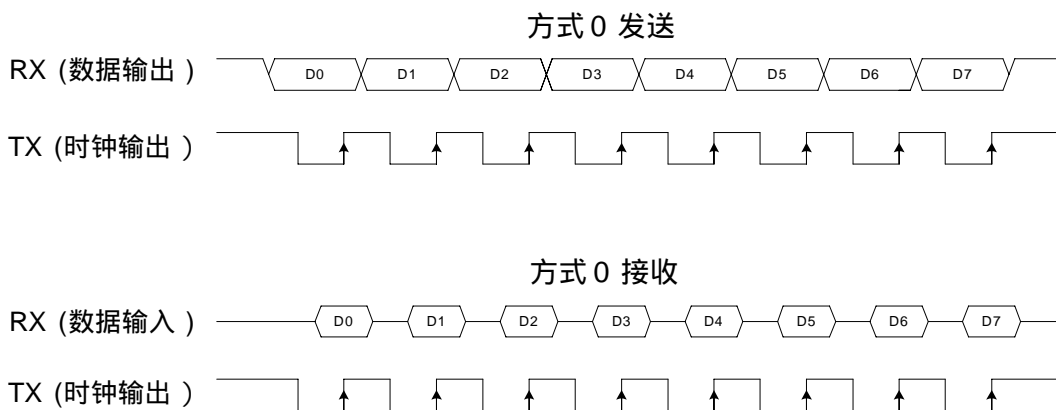


图 20.3 UART 方式 0 时序图



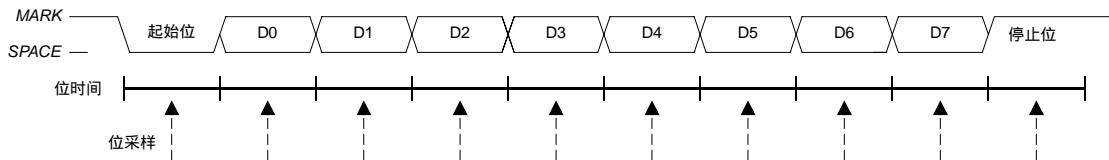
20.1.2 方式 1: 8 位 UART, 可变波特率

方式 1 提供标准的异步、全双工通信, 每个数据字节共使用 10 位: 一个起始位、8 个数据位 (LSB 在先) 和一个停止位。数据从 TX0 引脚发送, 在 RX0 引脚接收。在接收时, 8 个数据位存入 SBUF0, 停止位进入 RB80 (SCON0.2)。

当执行一条向 SBUF0 寄存器写入一个字节的指令时开始数据发送。在发送结束时 (停止位开始) 发送中断标志 TI0 (SCON0.1) 置位。在接收允许位 REN0 (SCON0.4) 被设置为逻辑 1 时开始数据接收。收到停止位后如果满足下述条件则数据字节将被装入接收寄存器 SBUF0: RI0 为逻辑 0, 并且如果 SM20 为逻辑 1 则停止位必须为 1。

如果这些条件满足, 则 8 位数据被存入 SBUF0, 停止位被存入 RB80, RI0 标志被置位。如果这些条件不满足, 则不装入 SBUF0 和 RB80, RI0 标志也不被置 1。如果中断被允许, 在 TI0 或 RI0 置位时将产生一个中断。

图 20.4 UART0 方式 1 时序图



方式 1 的波特率是定时器溢出时间的函数, 如方程 20.1 和方程 20.2 所示。UART0 可以使用定时器 1 工作在 8 位自动重装载方式或定时器 2 工作在波特率发生器方式产生波特率 (注意, TX 和 RX 时钟可以分别选择)。每次定时器发生溢出 (从全 1 (对定时器 1 为 0xFF, 对定时器 2 为 0xFFFF) 返回到 0) 时向波特率电路发送一个时钟脉冲。

通过将 TCLK0 (T2CON.4) 和/或 RCLK0 (T2CON.5) 为设置为逻辑 1 来选择定时器 2 为波特率时钟源 (有关定时器配置的详细信息见“22. 定时器”)。当 TCLK0 或 RCLK0 中的任何一个被置 1 时, 定时器 2 就被强制进入波特率发生器方式并使用系统时钟的二分频作为时钟源。如果 TCLK0 和/或 RCLK0 为逻辑 0, 定时器 1 就成为 TX 和/或 RX 电路的波特率时钟源。

下面给出了方式 1 的波特率方程, 其中: TIM 为定时器 1 时钟选择位 (CKCON.4), TH1 是定时器 1 的 8 位重装载寄存器, [RCAP2H:RCAP2L]是定时器 2 的重装载寄存器。

方程 20.1 使用定时器 1 的方式 1 波特率

$$\text{波特率} = \left(\frac{2^{SMOD0}}{32} \right) \times \left(\frac{SYSCLK \times 12^{(TIM-1)}}{(256 - TH1)} \right)$$

方程 20.2 使用定时器 2 的方式 1 波特率

$$\text{波特率} = \frac{SYSCLK}{32 \times (65536 - [RCAP2H : RCAP2L])}$$

20.1.3 方式 2: 9 位 UART, 固定波特率

方式 2 提供异步、全双工通信, 每个数据字节共使用 11 位: 一个起始位、8 个数据位 (LSB 在先)、一个可编程的第九位和一个停止位。方式 2 支持多处理器通信和硬件地址识别 (见 20.2 多处理器通信)。在发送时, 第九数据位由 TB80 (SCON0.3) 中的值决定。它可以被赋值为 PSW 中的奇偶标志 P, 或用于多处理器通信。在接收时, 第九数据位进入 RB80 (SCON0.2), 停止位被忽略。

当执行一条向 SBUF0 寄存器写入一个字节的指令时开始数据发送。在发送结束时 (停止位开始) 发送中断标志 TI0 (SCON0.1) 置位。在接收允许位 REN0 (SCON0.4) 被设置为逻辑 1 时开始数据接收。收到停止位后如果 RI0 为逻辑 0 并且满足下述条件则数据字节将被装入到接收寄存器 SBUF0:

1. SM20 为逻辑 0;
2. SM20 为逻辑 1, 接收的第九位为逻辑 1, 并且接收到的地址与 UART0 的地址匹配。

如果上述条件满足, 则 8 位数据被存入 SBUF0, 第九位被存入 RB80, RI0 标志被置位。如果这些条件不满足, 则不装入 SBUF0 和 RB80, RI0 标志也不被置 1。如果中断被允许, 在 TI0 或 RI0 置位时将产生一个中断。

方式 2 波特率为系统时钟频率的 1/32 或 1/64, 由寄存器 PCON 中的 SMOD0 位决定。

方程 20.3 方式 2 波特率

$$\text{波特率} = 2^{SMOD0} \times \left(\frac{SYSCLK}{64} \right)$$

图 20.5 UART0 方式 2 和 3 时序图

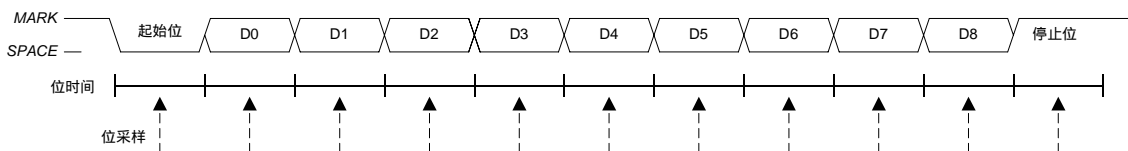
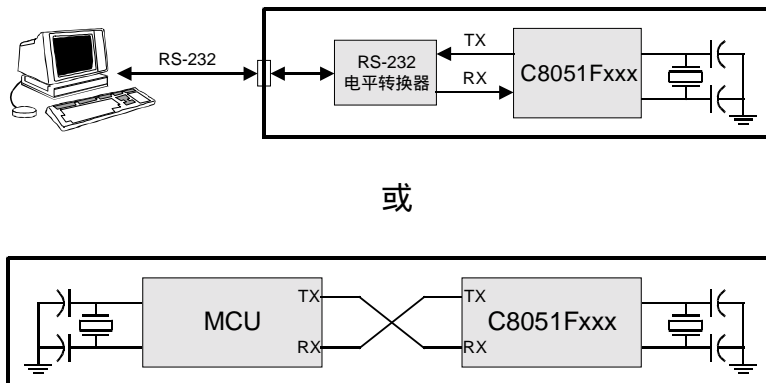


图 20.6 UART 方式 1、2 和 3 连接图



20.1.4 方式 3: 9 位 UART, 可变波特率

方式 3 使用方式 2 的传输协议, 波特率的确定方式与方式 1 相同。方式 3 操作使用 11 位: 一个起始位、8 个数据位 (LSB 在先)、一个可编程的第九位和一个停止位。用定时器 1 或定时器 2 溢出产生波特率, 见方程 20.1 和方程 20.2。支持多机通信和硬件地址识别。

20.2 多机通信

方式 2 和方式 3 通过使用第九数据位和内置 UART0 地址识别硬件支持一个主处理器与一个或多个从处理器之间的多机通信。当主机开始一次数据传输时先发送一个用于选择目标从机的地址字节。地址字节与数据字节的区别是: 地址字节的第九位为逻辑 1; 数据字节的第九位总是设置为逻辑 0。

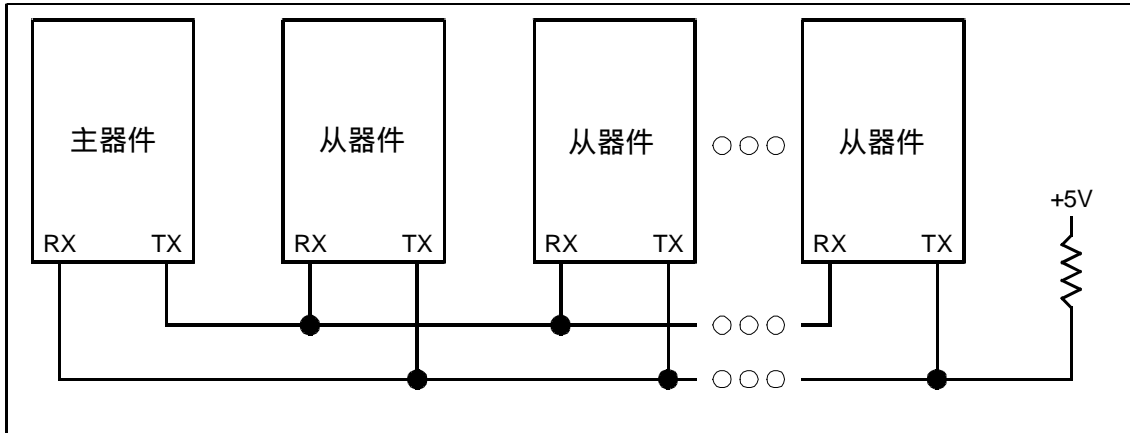
UART0 地址是通过两个 SFR 编程的: SADDR0 (串口地址) 和 SADEN0 (串口地址允许)。SADEN0 设置 SADDR0 中的地址的屏蔽位: SADEN0 中设置为逻辑 '1' 的位对应于 SADDR0 中那些用来检查接收到的地址字节的位; SADEN0 中被设置为逻辑 '0' 的位对应于 SADDR0 中那些“无关”位。

| 例 1 | 例 2 | 例 3 |
|---------------------|---------------------|---------------------|
| SADDR0 = 00110101 | SADDR0 = 00110101 | SADDR0 = 00110101 |
| SADEN0 = 00001111 | SADEN0 = 11110011 | SADEN0 = 11000000 |
| UART0 地址 = xxxx0101 | UART0 地址 = 0011xx01 | UART0 地址 = 00xxxxxx |

如果从机的 SM20 位 (SCON0.5) 被置 '1', 则只有当接收到的第九位为逻辑 1 (RB80=1), 收到有效的停止位并且接收的数据字节与 UART0 从地址匹配时 UART0 才会产生中断。在接收地址的中断处理程序中, 从机应清除它的 SM20 位以允许后面接收数据字节时产生中断。一旦接收完整个消息, 被寻址的从机应将它的 SM20 位重新置 '1' 以忽略所有的数据传输, 直到它收到下一个地址字节。在 SM20 为逻辑 '1' 时, UART0 忽略所有那些与 UART0 地址不匹配以及第九位不是逻辑 '1' 的字节。

可以将多个地址分配给一个从机, 或将一个地址分配给多个从机从而允许同时向多个从机进行“广播”式发送。广播地址是寄存器 SADDR0 和 SADEN0 的逻辑或, 结果为 '0' 的那些位被视为“无关”。一般来说广播地址 0xFF 会得到所有从机的响应, 这里假设将“无关”位视为 '1'。主机可以被配置为接收所有的传输数据, 或通过实现某种协议使主/从角色能临时变换以允许原来的主机和从机之间进行半双工通信。

图 20.7 UART 多机方式连接图



20.3 帧错误和传输错误检测

当寄存器 PCON 中的 SSTAT0 位被设置为逻辑 ‘1’ 时，下列方式具有帧错误检测功能。

注意：要访问状态位（FE0、RXOVR0 和 TXCOL0）时，SSTAT0 位必须被设置为逻辑 ‘1’；访问 UART0 方式选择位（SM00、SM10 和 SM20）时，SSTAT0 位必须被设置为逻辑 ‘0’。

所有方式：

当一次发送过程正在进行时，如果用户软件向 SBUF0 寄存器写数据，则发送冲突位（寄存器 SCON0 中的 TXCOL0）被置 ‘1’。注意：当寄存器 PCON 中的 SSTAT0 位为逻辑 ‘0’ 时，TXCOL0 位的功能是 SM20。

方式 1、2 和 3：

如果一个新的数据字节被锁存到接收缓冲器而前面接收的字节尚未被读取，则接收覆盖位（寄存器 SCON0 中的 RXOVR0）被置 ‘1’。注意：当寄存器 PCON 中的 SSTAT0 位为逻辑 ‘0’ 时，RXOVR0 位的功能是 SM10。

如果检测到一个无效（低电平）停止位，则帧错误位（寄存器 SCON0 中的 FE0）被置 ‘1’。注意：当寄存器 PCON 中的 SSTAT0 位为逻辑 ‘0’ 时，FE0 位的功能是 SM00。

表 20.2. 产生标准波特率的振荡器频率

| 振荡器频率(MHZ) | 分频系数 | 定时器 1 装载值* | 波特率** |
|------------|------|------------|----------------|
| 25.0 | 434 | 0xE5 | 57600(57870) |
| 25.0 | 868 | 0xCA | 28800 |
| 24.576 | 320 | 0xEC | 76800 |
| 24.576 | 848 | 0xCB | 28800(28921) |
| 24.0 | 208 | 0xF3 | 115200(115384) |
| 24.0 | 833 | 0xCC | 28800(28846) |
| 23.592 | 205 | 0xF3 | 115200(113423) |
| 23.592 | 819 | 0xCD | 28800(28911) |
| 22.1184 | 192 | 0xF4 | 115200 |
| 22.1184 | 768 | 0xD0 | 28800 |
| 18.432 | 160 | 0xF6 | 115200 |
| 18.432 | 640 | 0xD8 | 28800 |
| 16.5888 | 144 | 0xF7 | 115200 |
| 16.5888 | 576 | 0xDC | 28800 |
| 14.7456 | 128 | 0xF8 | 115200 |
| 14.7456 | 512 | 0xE0 | 28800 |
| 12.9024 | 112 | 0xF9 | 115200 |
| 12.9024 | 448 | 0xE4 | 28800 |
| 11.0592 | 96 | 0xFA | 115200 |
| 11.0592 | 384 | 0xE8 | 28800 |
| 9.216 | 80 | 0xFB | 115200 |
| 9.216 | 320 | 0xEC | 28800 |
| 7.3728 | 64 | 0xFC | 115200 |
| 7.3728 | 256 | 0xF0 | 28800 |
| 5.5296 | 48 | 0xFD | 115200 |
| 5.5296 | 192 | 0xF4 | 28800 |
| 3.6864 | 32 | 0xFE | 115200 |
| 3.6864 | 128 | 0xF8 | 28800 |
| 1.8432 | 16 | 0xFF | 115200 |
| 1.8432 | 64 | 0xFC | 28800 |

*假定 SMOD=1 且 TIM=1。

**括号里的数是实际波特率。

图 20.8 SCON0: UART0 控制寄存器

| | | | | | | | | |
|----------|------------|-------------|------|------|------|-----|--------------|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| SM00/FE0 | SM10/RXOV0 | SM20/TXCOL0 | REN0 | TB80 | RB80 | TI0 | RI0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0x98 |

位 7-6: 这些位的功能由 PCON 中的 SSTAT0 位决定。
 如果 SSTAT0 为逻辑 ‘1’，这些位是 UART0 的状态指示位，见 20.3 节。
 如果 SSTAT0 为逻辑 ‘0’，这些位选择串行口的工作方式。
 SM00-SM10: 串行口工作方式

| SM00 | SM10 | 方式 |
|------|------|-----------------------|
| 0 | 0 | 方式 0: 同步方式 |
| 0 | 1 | 方式 1: 8 位 UART, 可变波特率 |
| 1 | 0 | 方式 2: 9 位 UART, 固定波特率 |
| 1 | 1 | 方式 3: 9 位 UART, 可变波特率 |

位 5: SM20: 多处理器通信允许
 该位的功能取决于串行口工作方式。
 方式 0: 无作用。
 方式 1: 检查有效停止位
 0: 停止位的逻辑电平被忽略。
 1: 只有当停止位为逻辑电平 1 时 RI0 激活。
 方式 2 和方式 3: 多机通信允许
 0: 第九位的逻辑电平被忽略。
 1: 只有当第九位为逻辑 1 并且接收到的地址与 UART0 地址匹配时 RI0 才被置位并产生中断。

位 4: REN0: 接收允许
 该位允许/禁止 UART0 接收。
 0: UART0 接收禁止
 1: UART0 接收允许

位 3: TB80: 第九发送位
 该位的逻辑电平被赋值给方式 2 和 3 的第九发送位。在方式 0 和 1 中未用。跟据需要用软件置位或清 0。

位 2: RB80: 第九接收位
 该位被赋值为方式 2 和 3 中第九接收位的逻辑电平。在方式 1，如果 SM20 为逻辑 0，则 RB80 被赋值为所接收到的停止位的逻辑电平。RB80 在方式 0 中未用。

位 1: TI0: 发送中断标志
 当 UART0 发送完一个字节数据时（方式 0 时是在发送完第 8 位后，其它方式在停止位的开始）该位被硬件置 1。在 UART0 中断被允许时，置 1 该位将导致 CPU 转到 UART0 中断服务程序。该位必须用软件手动清 0。

位 0: RI0: 接收中断标志
 当 UART 接收到一个字节数据时（根据 SM20 位的选择）该位被硬件置 1。在 UART 中断被允许时，置 1 该位将会使 CPU 转到 UART 中断服务程序。该位必须用软件手动清 0。

图 20.9 SBUF0: UART0 数据缓冲寄存器

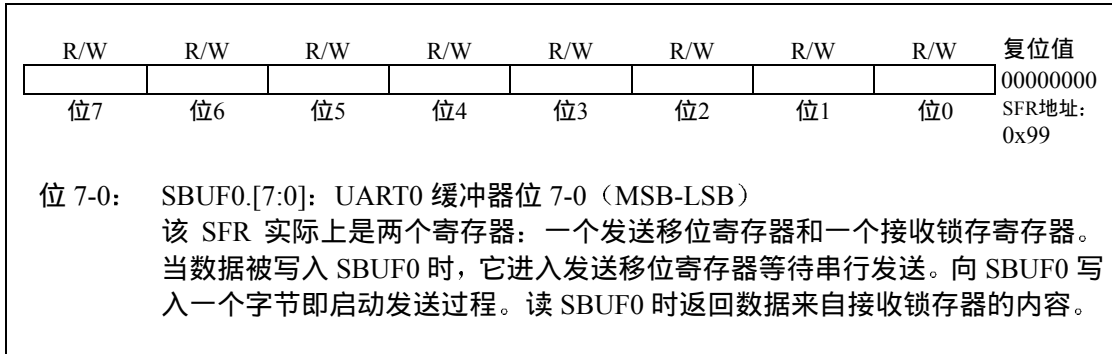


图 20.10 SADDR0: UART0 从地址寄存器

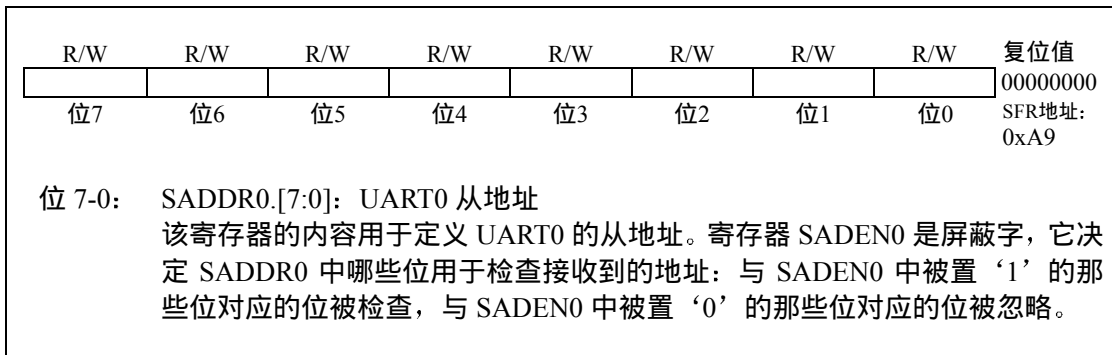
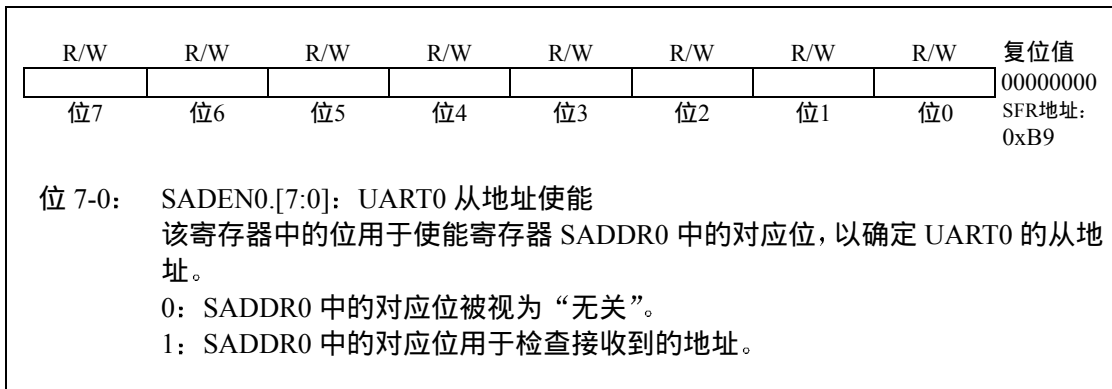


图 20.11 SADEN0: UART0 从地址使能寄存器



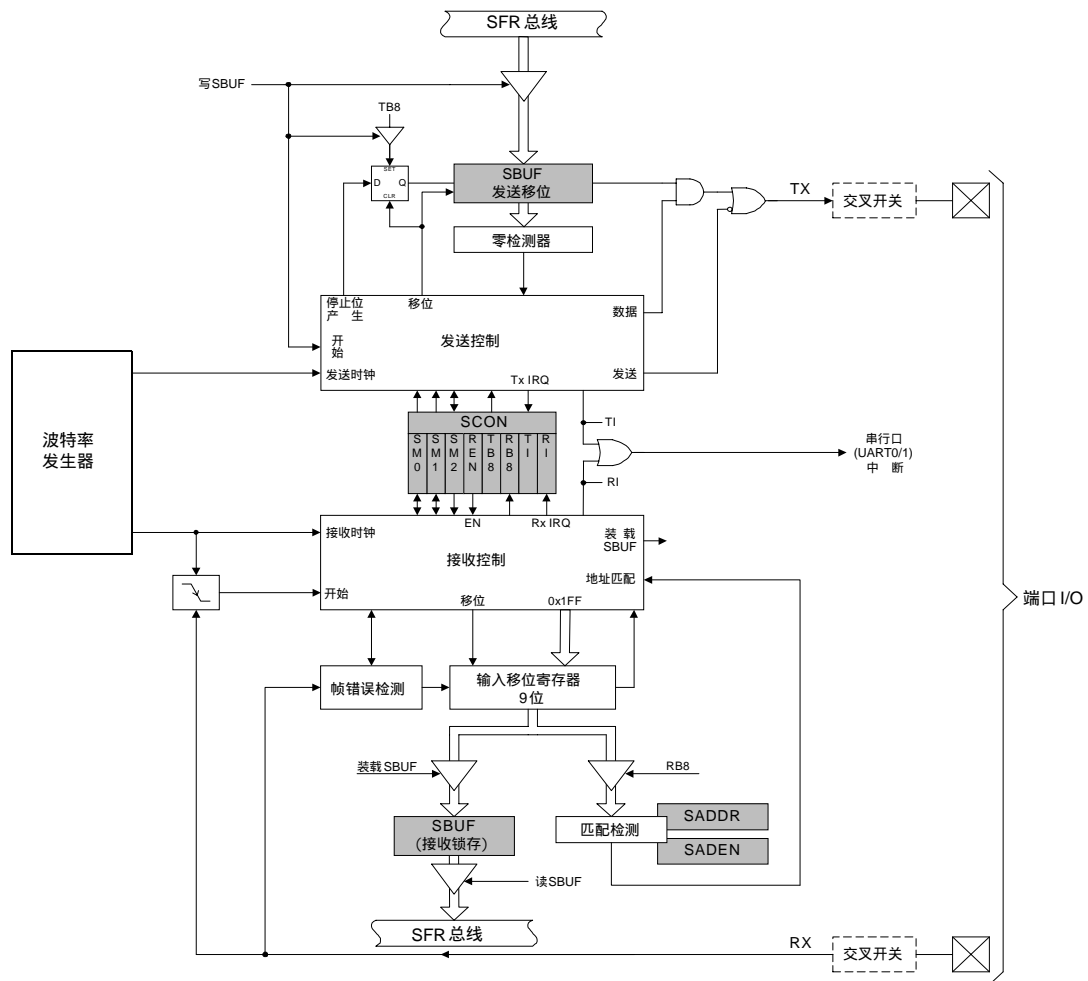
21. UART1

UART1 是一个具有帧错误检测和地址识别硬件的增强型串行口。UART1 可以工作在全双工异步方式或半双工同步方式，并且支持多处理器通信。接收数据被暂存于一个保持寄存器中，这就允许 UART1 在软件尚未读取前一个数据字节的情况下开始接收第二个输入数据字节。一个接收覆盖位用于指示新的接收数据已被锁存到接收缓冲器而前一个接收数据尚未被读取。

对 UART1 的控制和访问是通过相关的特殊功能寄存器即串行控制寄存器 (SCON1) 和串行数据缓冲器 (SBUF1) 来实现的。一个 SBUF1 地址可以访问发送寄存器和接收寄存器。读操作将自动访问接收寄存器，而写操作自动访问发送寄存器。

UART1 可以工作在查询或中断方式。UART1 有两个中断源：一个发送中断标志 TI1 (SCON1.1) (数据字节发送结束时置位) 和一个接收中断标志 RI1 (SCON1.0) (接收完一个数据字节后置位)。当 CPU 转向中断服务程序时硬件不清除 UART1 中断标志，中断标志必须由软件清除。这就允许软件查询 UART1 中断的原因 (发送完成或接收完成)。

图 21.1 UART1 原理框图



21.1 UART1 工作方式

UART1 提供四种工作方式（一种同步方式和三种异步方式），通过设置 SCON1 寄存器中的配置位选择。这四种方式提供不同的波特率和通信协议。下面的表 21.1 概述了这四种方式。

表 21.1 UART1 工作方式

| 方式 | 同步性 | 波特率时钟 | 数据位 | 起始/停止位 |
|----|-----|-----------------------|-----|-------------|
| 0 | 同步 | SYSCLK/12 | 8 | 无 |
| 1 | 异步 | 定时器 1 或定时器 2 溢出 | 8 | 一个起始位，一个停止位 |
| 2 | 异步 | SYSCLK/32 或 SYSCLK/64 | 9 | 一个起始位，一个停止位 |
| 3 | 异步 | 定时器 1 或定时器 2 溢出 | 9 | 一个起始位，一个停止位 |

21.1.1 方式 0：同步方式

方式 0 提供同步、半双工通信。在 RX1 引脚上发送和接收数据，TX1 引脚提供发送和接收的移位时钟。MCU 必须是主器件，因为它要为两个方向的数据传输产生移位时钟（见图 20.2 中的连接图）。

执行一条写 SBUF1 寄存器的指令时开始数据发送。发送/接收的数据为 8 位，LSB 在先（见图 21.3 中的时序图），在第 8 个位时间结束后发送中断标志 TI1（SCON1.1）置位。当接收允许位 REN1（SCON1.4）被设置为逻辑 1 并且接收中断标志 RI1（SCON1.0）被清 0 时开始数据接收。在第 8 位被移入后一个周期 RI1 标志置位，接收过程停止，直到软件清除 RI1 位。如果中断被允许，在 TI1 或 RI1 置位后将发生一次中断。

方式 0 的波特率是系统时钟频率除以 12。在方式 0，RX1 被强制为漏极开路方式，通常需要外接一个上拉电阻。

图 21.2 UART1 方式 0 连接

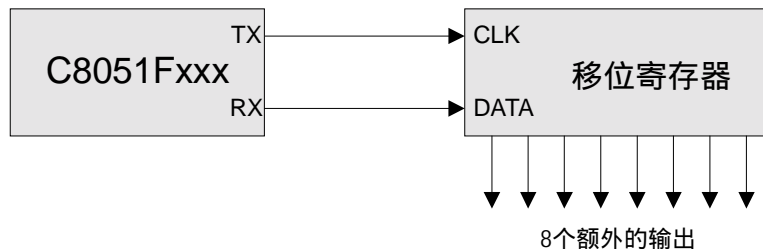
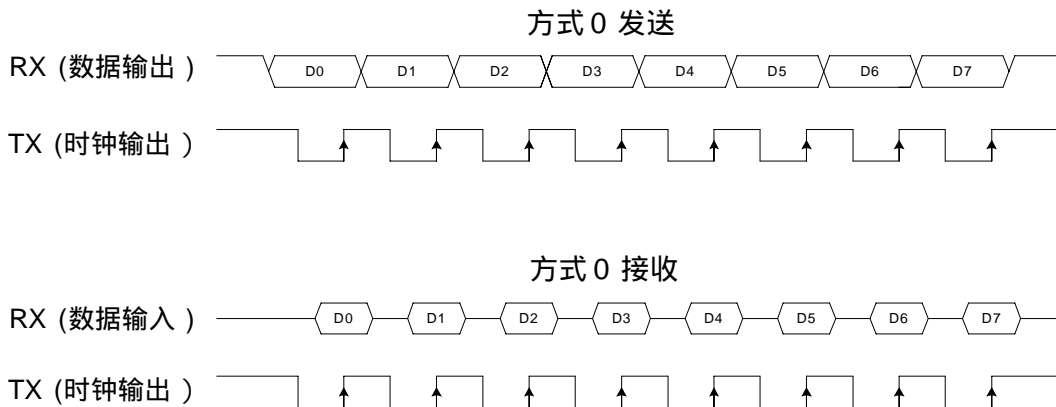


图 21.3 UART1 方式 0 时序图



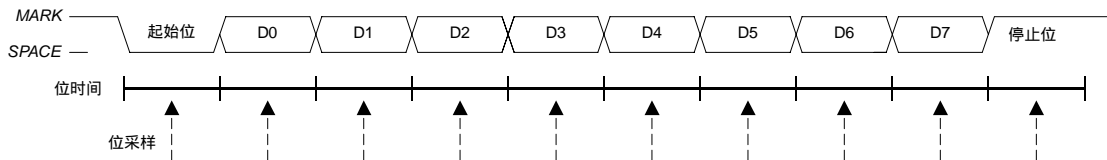
21.1.2 方式 1: 8 位 UART, 可变波特率

方式 1 提供标准的异步、全双工通信, 每个数据字节共使用 10 位: 一个起始位、8 个数据位 (LSB 在先) 和一个停止位。数据从 TX1 引脚发送, 在 RX1 引脚接收。在接收时, 8 个数据位存入 SBUF1, 停止位进入 RB81 (SCON1.2)。

当执行一条向 SBUF1 寄存器写入一个字节的指令时开始数据发送。在发送结束时 (停止位开始) 发送中断标志 TI1 (SCON1.1) 置位。在接收允许位 REN1 (SCON1.4) 被设置为逻辑 1 时开始数据接收。收到停止位后如果满足下述条件则数据字节将被装入接收寄存器 SBUF1: RI1 为逻辑 0, 并且如果 SM21 为逻辑 1 则停止位必须为 1。

如果这些条件满足, 则 8 位数据被存入 SBUF1, 停止位被存入 RB81, RI1 标志被置位。如果这些条件不满足, 则不装入 SBUF1 和 RB81, RI1 标志也不被置 1。如果中断被允许, 在 TI1 或 RI1 置位时将产生一个中断。

图 21.4 UART1 方式 1 时序图



方式 1 的波特率是定时器溢出时间的函数, 如方程 21.1 和方程 21.2 所示。UART1 可以使用定时器 1 工作在 8 位自动重装载方式或定时器 4 工作在波特率发生器方式产生波特率 (注意, TX 和 RX 时钟可以分别选择)。每次定时器发生溢出 (从全 1 (对定时器 1 为 0xFF, 对定时器 4 为 0xFFFF) 返回到 0) 时向波特率电路发送一个时钟脉冲。

通过将 TCLK1 (T4CON.4) 和/或 RCLK1 (T4CON.5) 为设置为逻辑 1 来选择定时器 4 为波特率时钟源 (有关定时器配置的详细信息见“22. 定时器”)。当 TCLK0 或 RCLK0 中的任何一个被置 1 时, 定时器 4 就被强制进入波特率发生器方式并使用系统时钟的二分频作为时钟源。如果 TCLK0 和/或 RCLK0 为逻辑 0, 定时器 1 就成为 TX 和/或 RX 电路的波特率时钟源。

下面给出了方式 1 的波特率方程, 其中: TIM 为定时器 1 时钟选择位 (CKCON.4), TH1 是定时器 1 的 8 位重装载寄存器, [RCAP4H:RCAP4L]是定时器 4 的重装载寄存器。

方程 21.1 使用定时器 1 的方式 1 波特率

$$\text{波特率} = \left(\frac{2^{SMOD0}}{32} \right) \times \left(\frac{SYSCLK \times 12^{(TIM-1)}}{(256 - TH1)} \right)$$

方程 21.2 使用定时器 4 的方式 1 波特率

$$\text{波特率} = \frac{SYSCLK}{32 \times (65536 - [RCAP4H : RCAP4L])}$$

21.1.3 方式 2: 9 位 UART, 固定波特率

方式 2 提供异步、全双工通信, 每个数据字节共使用 11 位: 一个起始位、8 个数据位 (LSB 在先)、一个可编程的第九位和一个停止位。方式 2 支持多处理器通信和硬件地址识别 (见 21.2 多处理器通信)。在发送时, 第九数据位由 TB81 (SCON1.3) 中的值决定。它可以被赋值为 PSW 中的奇偶标志 P, 或用于多处理器通信。在接收时, 第九数据位进入 RB81 (SCON1.2), 停止位被忽略。

当执行一条向 SBUF1 寄存器写入一个字节的指令时开始数据发送。在发送结束时 (停止位开始) 发送中断标志 TI1 (SCON1.1) 置位。在接收允许位 REN1 (SCON1.4) 被设置为逻辑 1 时开始数据接收。收到停止位后如果 RI1 为逻辑 0 并且满足下述条件则数据字节将被装入到接收寄存器 SBUF1:

1. SM21 为逻辑 0;
2. SM21 为逻辑 1, 接收的第九位为逻辑 1, 并且接收到的地址与 UART1 的地址匹配。

如果上述条件满足, 则 8 位数据被存入 SBUF1, 第九位被存入 RB81, RI1 标志被置位。如果这些条件不满足, 则不装入 SBUF1 和 RB81, RI1 标志也不被置 1。如果中断被允许, 在 TI1 或 RI1 置位时将产生一个中断。

方式 2 波特率为系统时钟频率的 1/32 或 1/64, 由寄存器 PCON 中的 SMOD1 位决定。

方程 21.3 方式 2 波特率

$$\text{波特率} = 2^{\text{SMOD1}} \times \left(\frac{\text{SYSCLK}}{64} \right)$$

图 21.5 UART1 方式 2 和 3 时序图

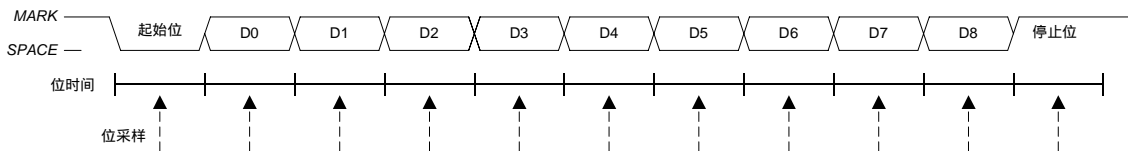
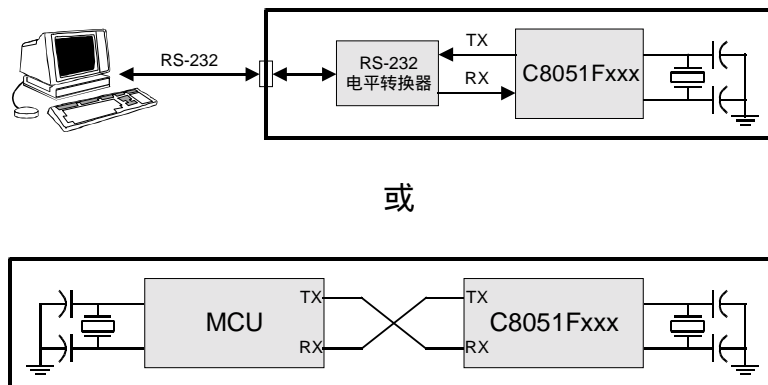


图 21.6 UART 方式 1、2 和 3 连接图



21.1.4 方式 3: 9 位 UART, 可变波特率

方式 3 使用方式 2 的传输协议, 波特率的确定方式与方式 1 相同。方式 3 操作使用 11 位: 一个起始位、8 个数据位 (LSB 在先)、一个可编程的第九位和一个停止位。用定时器 1 或定时器 4 溢出产生波特率, 见方程 21.1 和方程 21.2。

21.2 多机通信

方式 2 和方式 3 通过使用第九数据位和内置 UART1 地址识别硬件支持一个主处理器与一个或多个从处理器之间的多机通信。当主机开始一次数据传输时先发送一个用于选择目标从机的地址字节。地址字节与数据字节的区别是: 地址字节的第九位为逻辑 1; 数据字节的第九位总是设置为逻辑 0。

UART1 地址是通过两个 SFR 编程的: SADDR1 (串口地址) 和 SADEN1 (串口地址允许)。SADEN1 设置 SADDR1 中的地址的屏蔽位: SADEN1 中设置为逻辑 ‘1’ 的位对应于 SADDR1 中那些用来检查接收到的地址字节的位; SADEN1 中被设置为逻辑 ‘0’ 的位对应于 SADDR1 中那些 “无关” 位。

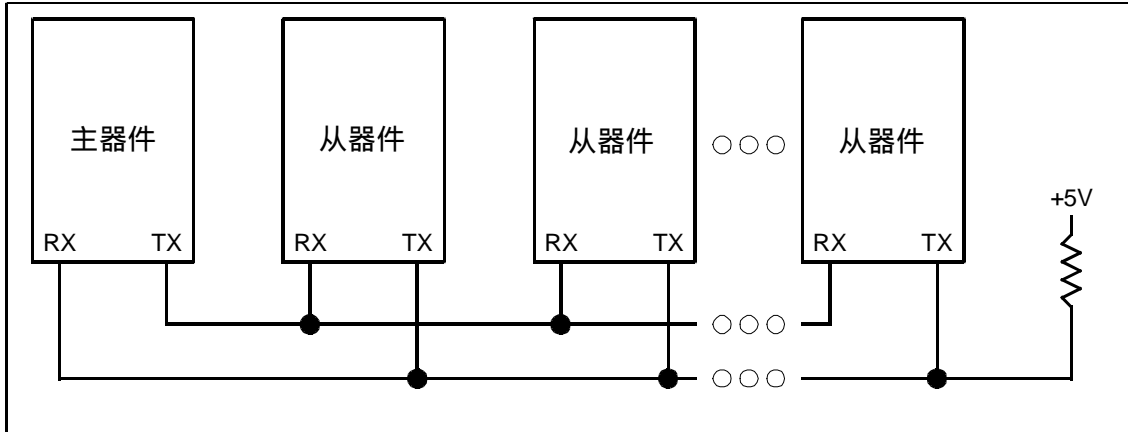
| 例 1 | 例 2 | 例 3 |
|---------------------|---------------------|---------------------|
| SADDR1 = 00110101 | SADDR1 = 00110101 | SADDR1 = 00110101 |
| SADEN1 = 00001111 | SADEN1 = 11110011 | SADEN1 = 11000000 |
| UART1 地址 = xxxx0101 | UART1 地址 = 0011xx01 | UART1 地址 = 00xxxxxx |

如果从机的 SM21 位 (SCON1.5) 被置 ‘1’, 则只有当接收到的第九位为逻辑 1 (RB81=1)、收到有效的停止位并且接收的数据字节与 UART1 从地址匹配时 UART1 才会产生中断。在接收地址的中断处理程序中, 从机应清除它的 SM21 位以允许后面接收数据字节时产生中断。一旦接收完整个消息, 被寻址的从机应将它的 SM21 位重新置 ‘1’ 以忽略所有的数据传输, 直到它收到下一个地址字节。在 SM21 为逻辑 ‘1’ 时, UART1 忽略所有那些与 UART1 地址不匹配以及第九位不是逻辑 ‘1’ 的字节。

可以将多个地址分配给一个从机, 或将一个地址分配给多个从机从而允许同时向多个从机进行 “广播” 式发送。广播地址是寄存器 SADDR1 和 SADEN1 的逻辑或, 结果为 ‘0’ 的

那些位被视为“无关”。一般来说广播地址 0xFF 会得到所有从机的响应，这里假设将“无关”位视为‘1’。主机可以被配置为接收所有的传输数据，或通过实现某种协议使主/从角色能临时变换以允许原来的主机和从机之间进行半双工通信。

图 21.7 UART 多机方式连接图



21.3 帧错误和传输错误检测

当寄存器 PCON 中的 SSTAT1 位被设置为逻辑‘1’时，下列方式具有帧错误检测功能。注意：访问状态位（FE1、RXOVR1 和 TXCOL1）时，SSTAT1 位必须被设置为逻辑‘1’；访问 UART1 方式选择位（SM01、SM11 和 SM21）时，SSTAT1 位必须被设置为逻辑‘0’。

所有方式：

当一次发送过程正在进行时，如果用户软件向 SBUF1 寄存器写数据，则发送冲突位（寄存器 SCON1 中的 TXCOL1）被置‘1’。注意：当寄存器 PCON 中的 SSTAT1 位为逻辑‘0’时，TXCOL1 位的功能是 SM21。

方式 1、2 和 3：

如果一个新的数据字节被锁存到接收缓冲器而前面接收的字节尚未被读取，则接收覆盖位（寄存器 SCON1 中的 RXOVR1）被置‘1’。注意：当寄存器 PCON 中的 SSTAT1 位为逻辑‘0’时，RXOVR1 位的功能是 SM11。

如果检测到一个无效（低电平）停止位，则帧错误位（寄存器 SCON1 中的 FE1）被置‘1’。注意：当寄存器 PCON 中的 SSTAT1 位为逻辑‘0’时，FE1 位的功能是 SM01。

表 21.2. 产生标准波特率的振荡器频率

| 振荡器频(MHZ) | 分频系数 | 定时器 1 装载值* | 波特率** |
|-----------|------|------------|----------------|
| 25.0 | 434 | 0xE5 | 57600(57870) |
| 25.0 | 868 | 0xCA | 28800 |
| 24.576 | 320 | 0xEC | 76800 |
| 24.576 | 848 | 0xCB | 28800(28921) |
| 24.0 | 208 | 0xF3 | 115200(115384) |
| 24.0 | 833 | 0xCC | 28800(28846) |
| 23.592 | 205 | 0xF3 | 115200(113423) |
| 23.592 | 819 | 0xCD | 28800(28911) |
| 22.1184 | 192 | 0xF4 | 115200 |
| 22.1184 | 768 | 0xD0 | 28800 |
| 18.432 | 160 | 0xF6 | 115200 |
| 18.432 | 640 | 0xD8 | 28800 |
| 16.5888 | 144 | 0xF7 | 115200 |
| 16.5888 | 576 | 0xDC | 28800 |
| 14.7456 | 128 | 0xF8 | 115200 |
| 14.7456 | 512 | 0xE0 | 28800 |
| 12.9024 | 112 | 0xF9 | 115200 |
| 12.9024 | 448 | 0xE4 | 28800 |
| 11.0592 | 96 | 0xFA | 115200 |
| 11.0592 | 384 | 0xE8 | 28800 |
| 9.216 | 80 | 0xFB | 115200 |
| 9.216 | 320 | 0xEC | 28800 |
| 7.3728 | 64 | 0xFC | 115200 |
| 7.3728 | 256 | 0xF0 | 28800 |
| 5.5296 | 48 | 0xFD | 115200 |
| 5.5296 | 192 | 0xF4 | 28800 |
| 3.6864 | 32 | 0xFE | 115200 |
| 3.6864 | 128 | 0xF8 | 28800 |
| 1.8432 | 16 | 0xFF | 115200 |
| 1.8432 | 64 | 0xFC | 28800 |

*假定 SMOD=1 且 TIM=1。

**括号里的数是实际波特率。

图 21.8 SCON1: UART1 控制寄存器

| | | | | | | | | |
|----------|------------|-------------|------|------|------|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| SM01/FE1 | SM11/RXOV1 | SM21/TXCOL1 | REN1 | TB81 | RB81 | TI1 | RI1 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xF1 |

位 7-6: 这些位的功能由 PCON 中的 SSTAT1 位决定。
 如果 SSTAT1 为逻辑 ‘1’，这些位是 UART1 的状态指示位，见 21.3 节。
 如果 SSTAT1 为逻辑 ‘0’，这些位选择串行口的工作方式。
 SM01-SM11: 串行口工作方式

| SM01 | SM11 | 方式 |
|------|------|-----------------------|
| 0 | 0 | 方式 0: 同步方式 |
| 0 | 1 | 方式 1: 8 位 UART, 可变波特率 |
| 1 | 0 | 方式 2: 9 位 UART, 固定波特率 |
| 1 | 1 | 方式 3: 9 位 UART, 可变波特率 |

位 5: SM21: 多处理器通信允许
 该位的功能取决于串行口工作方式。
 方式 0: 无作用。
 方式 1: 检查有效停止位
 0: 停止位的逻辑电平被忽略。
 1: 只有当停止位为逻辑电平 1 时 RI0 激活。
 方式 2 和方式 3: 多机通信允许
 0: 第九位的逻辑电平被忽略。
 1: 只有当第九位为逻辑 1 并且接收到的地址与 UART1 地址匹配时 RI1 才被置位并产生中断。

位 4: REN1: 接收允许
 该位允许/禁止 UART1 接收。
 0: UART1 接收禁止
 1: UART1 接收允许

位 3: TB81: 第九发送位
 该位的逻辑电平被赋值给方式 2 和 3 的第九发送位。在方式 0 和 1 中未用。跟据需要用软件置位和清 0。

位 2: RB81: 第九接收位
 该位被赋值为方式 2 和 3 中第九接收位的逻辑电平。在方式 1, 如果 SM21 为逻辑 0, 则 RB81 被赋值为所接收到的停止位的逻辑电平。RB81 在方式 0 中未用。

位 1: TI1: 发送中断标志
 当 UART1 发送完一个字节数据时 (方式 0 时是在发送完第 8 位后, 其它方式在停止位的开始) 该位被硬件置 1。在 UART1 中断被允许时, 置 1 该位将导致 CPU 转到 UART1 中断服务程序。该位必须用软件手动清 0。

位 0: RI1: 接收中断标志
 当 UART 接收到一个字节数据时 (根据 SM21 位的选择) 该位被硬件置 1。在 UART1 中断被允许时, 置 1 该位将会使 CPU 转到 UART1 中断服务程序。该位必须用软件手动清 0。

图 21.9 SBUF1: UART1 数据缓冲寄存器

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xF2 |

位 7-0: SBUF1.[7:0]: UART1 缓冲器位 7-0 (MSB-LSB)
 该 SFR 实际上是两个寄存器: 一个发送移位寄存器和一个接收锁存寄存器。
 当数据被写入 SBUF1 时, 它进入发送移位寄存器等待串行发送。向 SBUF1 写入一个字节即启动发送过程。读 SBUF1 时返回数据来自接收锁存器的内容。

图 21.10 SADDR1: UART1 从地址寄存器

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xF3 |

位 7-0: SADDR1.[7:0]: UART1 从地址
 该寄存器的内容用于定义 UART1 的从地址。寄存器 SADEN1 是屏蔽字, 它决定 SADDR1 中的哪些位用于检查接收到的地址: 与 SADEN1 中被置 ‘1’ 的那些位对应的位被检查, 与 SADEN1 中被置 ‘0’ 的那些位对应的位被忽略。

图 21.11 SADEN1: UART1 从地址使能寄存器

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xAE |

位 7-0: SADEN1.[7:0]: UART1 从地址使能
 该寄存器中的位用于使能寄存器 SADDR1 中的对应位, 以确定 UART1 的从地址。
 0: SADDR1 中的对应位被视为 “无关”。
 1: SADDR1 中的对应位用于检查接收到的地址。

22. 定时器

C8051F020/1/2/3 内部有 5 个计数器/定时器: 其中三个 16 位计数器/定时器与标准 8051 中的计数器/定时器兼容, 还有两个 16 位自动重载定时器可用于 ADC、SMBus 或作为通用定时器使用。这些计数器/定时器可以用于测量时间间隔, 对外部事件计数或产生周期性的中断请求。定时器 0 和定时器 1 几乎完全相同, 有四种工作方式。定时器 2 增加了一些定时器 0 和定时器 1 中所没有的功能。定时器 3 与定时器 2 类似, 但没有捕捉或波特率发生器方式。定时器 4 与定时器 2 完全相同, 可用作 UART1 的波特率发生源。

| 定时器 0 和定时器 1 | 定时器 2 | 定时器 3 | 定时器 4 |
|--------------------------|-------------------|-------------------|-------------------|
| 13 位计数/定时器 | 自动重载的 16 位计数器/定时器 | 自动重载的 16 位计数器/定时器 | 自动重载的 16 位计数器/定时器 |
| 16 位计数器/定时器 | 带捕捉的 16 位计数器/定时器 | | 带捕捉的 16 位计数器/定时器 |
| 8 位自动重载的计数器/定时器 | UART0 的波特率发生器 | | UART1 的波特率发生器 |
| 两个 8 位计数器/定时器 (只限于定时器 0) | | | |

当工作在定时器方式时, 计数器/定时器寄存器在每个时钟滴答加 1。时钟滴答为系统时钟除以 1 或系统时钟除以 12, 由 CKCON 中的定时器时钟选择位 (T4M-T0M) 指定, 如图 22.1 所示。每滴答为 12 个时钟的选项提供了与标准 8051 系列的兼容性。需要更快速定时器的应用可以使用每滴答 1 个时钟的选项。

当作为计数器使用时, 所选择的引脚上出现负跳变时计数器/定时器寄存器加 1。对事件计数的最大频率可达到系统时钟频率的四分之一。输入信号不需要是周期性的, 但在一个给定电平上的保持时间至少应为两个完整的系统时钟周期, 以保证该电平能够被采样。

图 22.1 CKCON: 时钟控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|-----|-----|-----|-----|-----|-----|-----|----------------|
| - | T4M | T2M | T1M | T0M | 保留 | 保留 | 保留 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x8E |
| <p>位 7: 未用。读=0b, 写=忽略。</p> <p>位 6: T4M: 定时器 4 时钟选择。 该位控制提供给定时器 4 的系统时钟的分频数。当定时器工作于波特率发生器方式或计数器方式 (即 C/T4=1) 时该位被忽略。 0: 定时器 4 使用系统时钟的 12 分频 1: 定时器 4 使用系统时钟</p> <p>位 5: T2M: 定时器 2 时钟选择。 该位控制提供给定时器 2 的系统时钟的分频数。当定时器工作于波特率发生器方式或计数器方式 (即 C/T2=1) 时该位被忽略。 0: 定时器 2 使用系统时钟的 12 分频 1: 定时器 2 使用系统时钟</p> <p>位 4: T1M: 定时器 1 时钟选择。 该位控制提供给定时器 1 的系统时钟的分频数。 0: 定时器 1 使用系统时钟的 12 分频 1: 定时器 1 使用系统时钟</p> <p>位 3: T0M: 定时器 0 时钟选择。 该位控制提供给定时器 0 的系统时钟的分频数。 0: 计数器/定时器使用系统时钟的 12 分频 1: 计数器/定时器使用系统时钟</p> <p>位 2-0: 保留。读=000b, 写入值必须是 000。</p> | | | | | | | | |

22.1 定时器 0 和定时器 1

对定时器 0 和定时器 1 的访问和控制是通过 SFR 实现的。每个计数器/定时器都是一个 16 位的寄存器, 在被访问时以两个字节的出现: 一个低字节 (TL0 或 TL1) 和一个高字节 (TH0 或 TH1)。计数器/定时器控制寄存器 (TCON) 用于允许定时器 0 和定时器 1 以及指示它们的状态。这两个计数器/定时器都有四种工作方式, 通过设置计数器/定时器方式寄存器 (TMOD) 中的方式选择位 M1-M0 来选择工作方式。每个定时器都可以被独立编程。下面对每种工作方式进行详细说明。

22.1.1 方式 0: 13 位计数器/定时器

在方式 0 时, 定时器 0 和定时器 1 被作为 13 位的计数器/定时器使用。下面介绍对定时器 0 的配置和操作。由于这两个定时器在工作上完全相同, 定时器 1 的配置过程与定时器 0 一样。

TH0 寄存器保持 13 位计数器/定时器的 8 个 MSB。TL0 在 TL0.4-TL0.0 位置保持 5 个 LSB。TL0 的高 3 位 (TL0.7-TL0.5) 是不确定的, 在读计数值时应屏蔽掉或忽略这 3 位。作为 13 位

C8051F020/1/2/3 混合信号 ISP FLASH 微控制器

定时器寄存器，计到 0x1FFF（全 1）后再计一次将发生溢出，使计数值回到 0x0000，此时定时器溢出标志 TF0（TCON.5）被置位并产生一个中断（如果被允许）。

C/T0 位（TMOD.2）选择计数器/定时器的时钟源。清除 C/T 将选择系统时钟作为定时器的输入。当 C/T0 被设置为逻辑 1 时，出现在所选输入引脚（T0）上的负跳变使定时器寄存器加 1。（有关选择和配置用于数字外设的外部 I/O 引脚的详细信息见“17.1 端口 0 - 端口 3 和优先级交叉开关译码器”。）

当 GATE0（TMOD.3）为 0 或输入信号/INT0 为逻辑 1 时，置 ‘1’ TR0（TCON.4）位将允许定时器 0 工作。设置 GATE0 为逻辑 1 允许定时器 0 受外部输入信号/INT0 的控制，便于脉冲宽度测量。

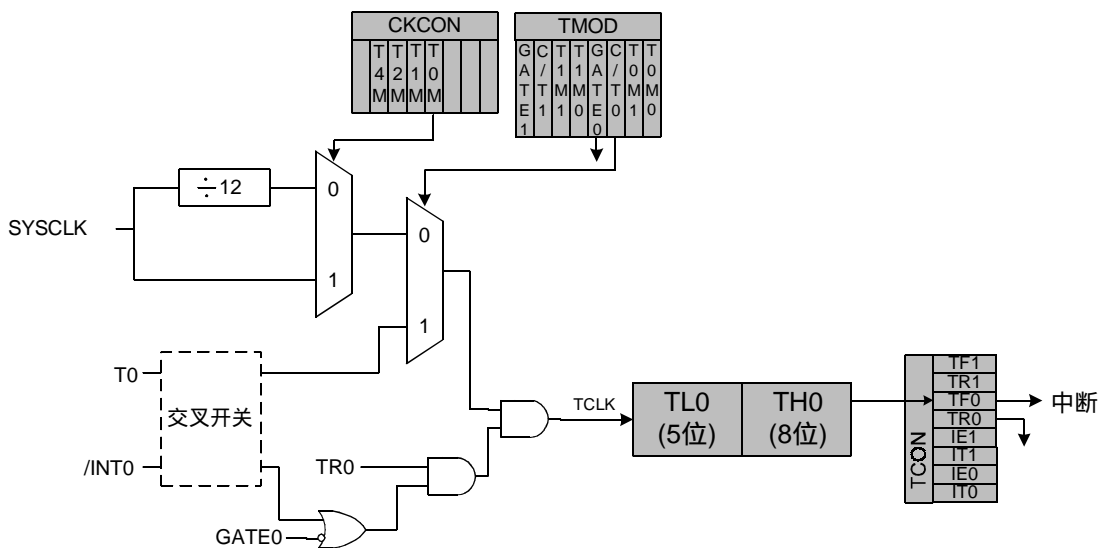
| TR0 | GATE0 | /INT0 | 计数器/定时器 |
|-----|-------|-------|---------|
| 0 | X | X | 禁止 |
| 1 | 0 | X | 允许 |
| 1 | 1 | 0 | 禁止 |
| 1 | 1 | 1 | 允许 |

X=任意

置 ‘1’ TR0 位（TCON.4）并不复位定时器寄存器。在允许定时器之前应对定时器寄存器赋初值。

与上述的 TL0 和 TH0 一样，TL1 和 TH1 构成定时器 1 的 13 位寄存器。定时器 1 的配置和控制方法与定时器 0 一样，使用 TCON 和 TMOD 中的相应位。

图 22.2 T0 方式 0 原理框图



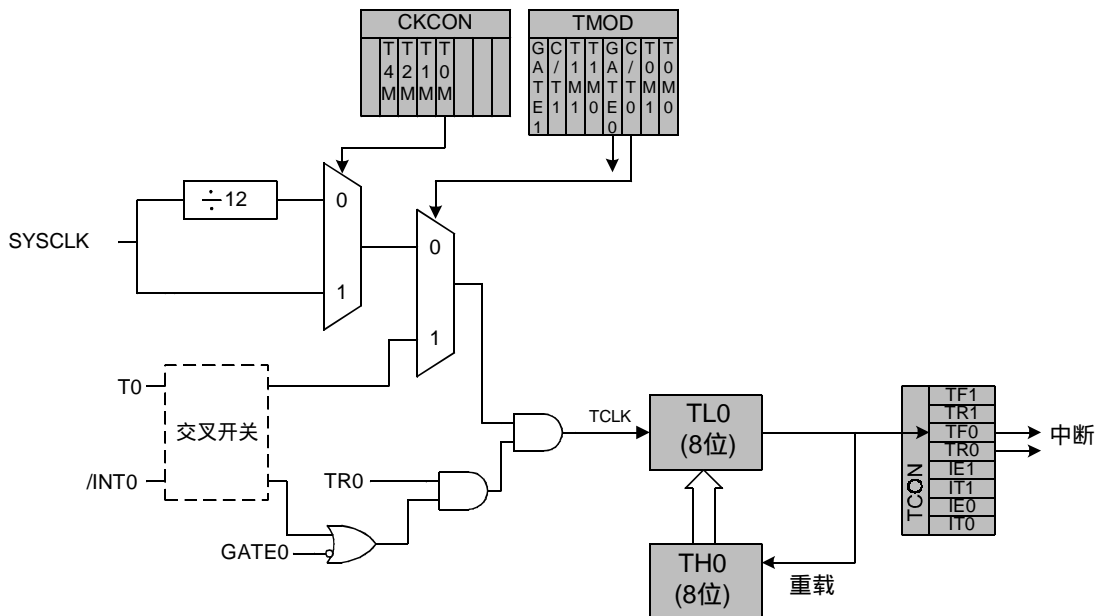
22.1.2 方式 1: 16 位计数器/定时器

方式 1 的操作与方式 0 完全一样，所不同的是计数器/定时器使用全部 16 位。用与方式 0 相同的方法允许和控制工作在方式 1 的计数器/定时器。

22.1.3 方式 2: 8 位自动重载的计数器/定时器

方式 2 将定时器 0 和定时器 1 配置为具有自动重新装入计数初值能力的 8 位计数器/定时器。TL0 保持计数值，而 TH0 保持重载值。当 TL0 中的计数值发生溢出（从全‘1’到 0x00）时，定时器溢出标志 TF0（TCON.5）被置位，TH0 中的重载值被重新装入到 TL0。如果中断被允许，在 TF0 被置位时将产生一个中断。TH0 中的重载值保持不变。为了保证第一次计数正确，必须在允许定时器之前将 TL0 初始化为所希望的计数初值。当工作于方式 2 时，定时器 1 的操作与定时器 0 完全相同。在方式 2，定时器 0 和 1 的允许和配置方法与方式 0 一样。

图 22.3 T0 方式 2（8 位重载）原理框图



22.1.4 方式 3：两个 8 位计数器/定时器（仅定时器 0）

在方式 3 时，定时器 0 和定时器 1 的功能不同。定时器 0 被配置为两个独立的 8 位定时器/计数器，计数值在 TL0 和 TH0 中。在 TL0 中的计数器/定时器使用 TCON 和 TMOD 中定时器 0 的控制/状态位：TR0、C/T0、GATE0 和 TF0。它既可以使用系统时钟也可以使用一个外部输入信号作为时间基准。TH0 寄存器只能作为定时器使用，由系统时钟提供时间基准。TH0 使用定时器 1 的运行控制位 TR1。TH0 在发生溢出时将定时器 1 的溢出标志位 TF1 置‘1’，所以它控制定时器 1 的中断。

定时器 1 在方式 3 时停止运行。当定时器 0 工作于方式 3 时，可以通过将定时器 1 切换到方式 3 使其停止运行。在定时器 0 工作于方式 3 时，定时器 1 可以工作在方式 0、1 或 2，但不能用外部信号作为时钟，也不能设置 TF1 标志和产生中断。但是定时器 1 溢出可以用于产生 UART0 和/或 UART1 的波特率时钟。有关将定时器 1 配置为波特率发生器的详细信息见“20. UART0”和“21. UART1”。

图 22.4 T0 方式 3（两个 8 位定时器）原理框图

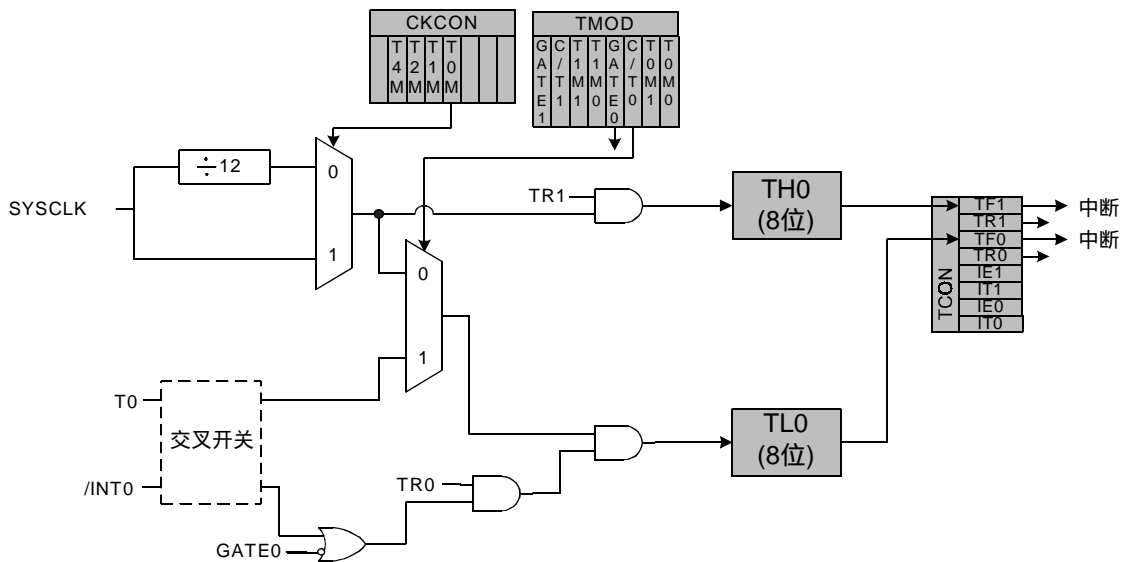


图 22.5 TCON: 定时器控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|-----|-----|-----|-----|-----|-----|--------------|----------------|
| TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0x88 |
| <p>位 7: TF1: 定时器 1 溢出标志 当定时器 1 溢出时由硬件置位。该位可以用软件清 0, 但当 CPU 转向定时器 1 中断服务程序时该位被自动清 0。 0: 未检测到定时器 1 溢出。 1: 定时器 1 发生溢出。</p> <p>位 6: TR1: 定时器 1 运行控制 0: 定时器 1 禁止。 1: 定时器 1 允许。</p> <p>位 5: TF0: 定时器 0 溢出标志 当定时器 0 溢出时由硬件置位。该位可以用软件清 0, 但当 CPU 转向定时器 0 中断服务程序时该位被自动清 0。 0: 未检测到定时器 0 溢出。 1: 定时器 0 发生溢出。</p> <p>位 4: TR0: 定时器 0 运行控制 0: 定时器 0 禁止 1: 定时器 0 允许</p> <p>位 3: IE1: 外部中断 1 当检测到一个由 IT1 定义的边沿/电平时, 该标志由硬件置位。该位可以用软件清 0, 但当 CPU 转向外部中断 1 中断服务程序时该位被自动清 0(如果 IT1=1)。当 IT1=0 时, 该标志是/INT1 输入信号的逻辑电平取反。</p> <p>位 2: IT1: 中断 1 类型选择 该位选择/INT1 信号检测下降沿中断还是检测低电平有效中断。 0: /INT1 为电平触发 1: /INT1 为边沿触发</p> <p>位 1: IE0: 外部中断 0 当检测到一个由 IT0 定义的边沿/电平时, 该标志由硬件置位。该位可以用软件清 0, 但当 CPU 转向外部中断 0 中断服务程序时该位被自动清 0(如果 IT0=1)。当 IT0=0 时, 该标志是/INT0 输入信号的逻辑电平取反。</p> <p>位 0: IT0: 中断 0 类型选择 该位选择/INT0 信号检测下降沿中断还是检测低电平有效中断。 0: /INT0 为电平触发 1: /INT0 为边沿触发</p> | | | | | | | | |

图 22.6 TMOD: 定时器方式寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|------|-------------------------|------|-------|------|------|------|----------------|------|------|----|---|---|-------------------|---|---|-------------------|---|---|-------------------------|---|---|------------------|------|------|----|---|---|-------------------|---|---|-------------------|---|---|-------------------------|---|---|--------------------|
| GATE1 | C/T1 | T1M1 | T1M0 | GATE0 | C/T0 | T0M1 | T0M0 | 00000000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x89 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>位 7: GATE1: 定时器 1 门控位 0: 当 TR1=1 时定时器 1 被允许, 与/INT1 的逻辑电平无关。 1: 只有当 TR1=1 并且/INT1=逻辑 1 时定时器 1 被允许。</p> <p>位 6: C/T1: 计数器/定时器 1 功能选择。 0: 定时器功能: 定时器 1 由 T1M 位 (CKCON.4) 定义的时钟加 1。 1: 计数器功能: 定时器 1 由外部输入引脚 (T1) 的负跳变加 1。</p> <p>位 5-4: T1M1-T1M0: 定时器 1 方式选择 这些位选择定时器 1 的工作方式。</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 10px 0;"> <thead> <tr> <th style="width: 10%;">T1M1</th> <th style="width: 10%;">T1M0</th> <th style="width: 80%;">方式</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>方式 0: 13 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>方式 1: 16 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>方式 2: 自动重装载的 8 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>方式 3: 定时器 1 停止运行</td> </tr> </tbody> </table> <p>位 3: GATE0: 定时器 0 门控位 0: 当 TR0=1 时定时器 0 被允许, 与/INT0 的逻辑电平无关。 1: 只有当 TR0=1 并且/INT0=逻辑 1 时定时器 1 被允许。</p> <p>位 2: C/T0: 计数器/定时器 0 功能选择。 0: 定时器功能: 定时器 0 由 T0M 位 (CKCON.3) 定义的时钟加 1。 1: 计数器功能: 定时器 0 由外部输入引脚 (T0) 的负跳变加 1。</p> <p>位 1-0: T0M1-T0M0: 定时器 0 方式选择 这些位选择定时器 0 的工作方式。</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 10px 0;"> <thead> <tr> <th style="width: 10%;">T0M1</th> <th style="width: 10%;">T0M0</th> <th style="width: 80%;">方式</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>方式 0: 13 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>方式 1: 16 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>方式 2: 自动重装载的 8 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>方式 3: 双 8 位计数器/定时器</td> </tr> </tbody> </table> | | | | | | | | | T1M1 | T1M0 | 方式 | 0 | 0 | 方式 0: 13 位计数器/定时器 | 0 | 1 | 方式 1: 16 位计数器/定时器 | 1 | 0 | 方式 2: 自动重装载的 8 位计数器/定时器 | 1 | 1 | 方式 3: 定时器 1 停止运行 | T0M1 | T0M0 | 方式 | 0 | 0 | 方式 0: 13 位计数器/定时器 | 0 | 1 | 方式 1: 16 位计数器/定时器 | 1 | 0 | 方式 2: 自动重装载的 8 位计数器/定时器 | 1 | 1 | 方式 3: 双 8 位计数器/定时器 |
| T1M1 | T1M0 | 方式 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 方式 0: 13 位计数器/定时器 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 方式 1: 16 位计数器/定时器 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 方式 2: 自动重装载的 8 位计数器/定时器 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 方式 3: 定时器 1 停止运行 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| T0M1 | T0M0 | 方式 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 方式 0: 13 位计数器/定时器 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 方式 1: 16 位计数器/定时器 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 方式 2: 自动重装载的 8 位计数器/定时器 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 方式 3: 双 8 位计数器/定时器 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

图 22.7 TL0: 定时器 0 低字节

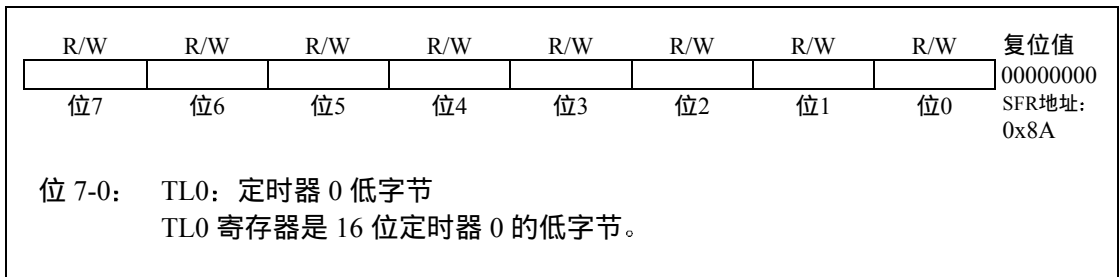


图 22.8 TL1: 定时器 1 低字节

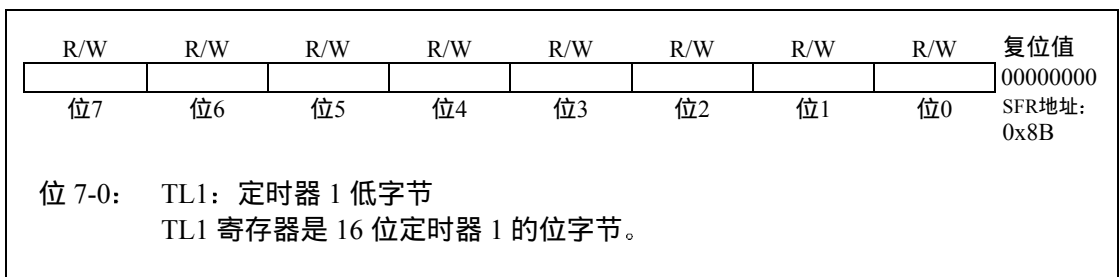


图 22.9 TH0: 定时器 0 高字节

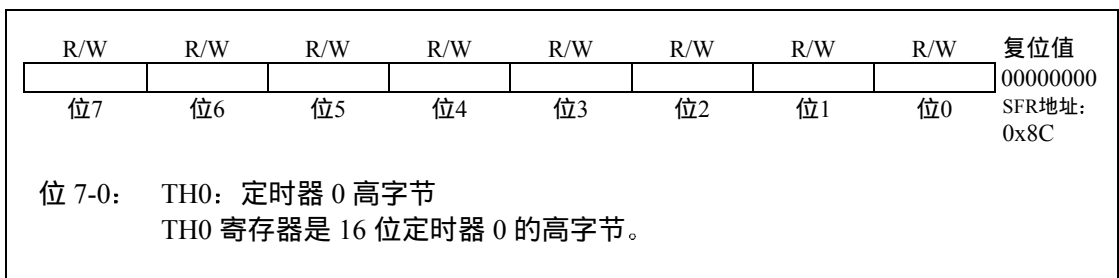
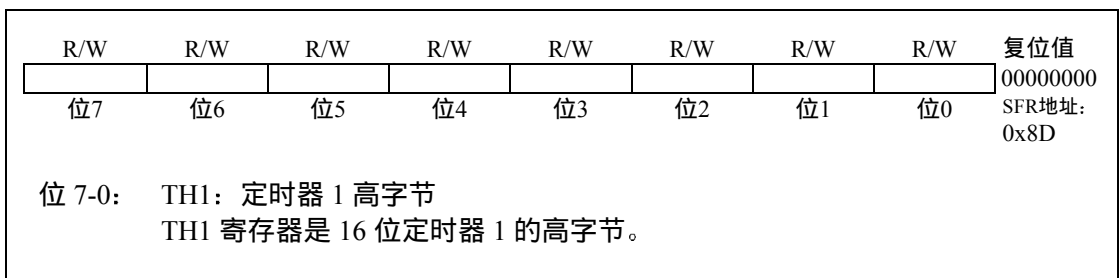


图 22.10 TH1: 定时器 1 高字节



22.2 定时器 2

定时器 2 是一个 16 位的计数器/定时器，由两个 8 位的 SFR 组成：TL2（低字节）和 TH2（高字节）。与定时器 0 和定时器 1 一样，它既可以使用系统时钟也可以使用一个外部输入引脚（T2）上的状态变化作为时钟源。计数器/定时器选择位 C/T2（T2CON.1）选择定时器 2 的时钟源。清除 C/T2 将选择系统时钟作为定时器的输入（由 CKCON 中的定时器时钟选择位 T2M 指定不分频或 12 分频）。当 C/T2 被置 ‘1’ 时，T2 输入引脚上的负跳变使计数器/定时器寄存器加 1。（有关选择和配置外部输入引脚的详细信息见“17.1 端口 0 – 端口 3 和有优先级交叉开关译码器”。）定时器 2 还可用于启动 ADC 数据转换。

定时器 2 提供了定时器 0 和定时器 1 所不具备的功能。它有三种工作方式：带捕捉的 16 位计数器/定时器、自动重载的 16 位计数器/定时器和波特率发生器方式。通过设置定时器 2 控制（T2CON）寄存器中的配置位来选择定时器 2 的工作方式。下面是定时器 2 工作方式和用于配置计数器/定时器的配置位的一览表，后面将对每种工作方式进行详细说明。

| RCLK | TCLK | CP/RL2 | TR2 | 方式 |
|------|------|--------|-----|-------------------|
| 0 | 0 | 1 | 1 | 带捕捉的 16 位计数器/定时器 |
| 0 | 0 | 0 | 1 | 自动重载的 16 位计数器/定时器 |
| 0 | 1 | X | 1 | UART0 波特率发生器 |
| 1 | 0 | X | 1 | UART0 波特率发生器 |
| 1 | 1 | X | 1 | UART0 波特率发生器 |
| X | X | X | 0 | 关闭 |

22.2.1 方式 0：带捕捉的 16 位计数器/定时器

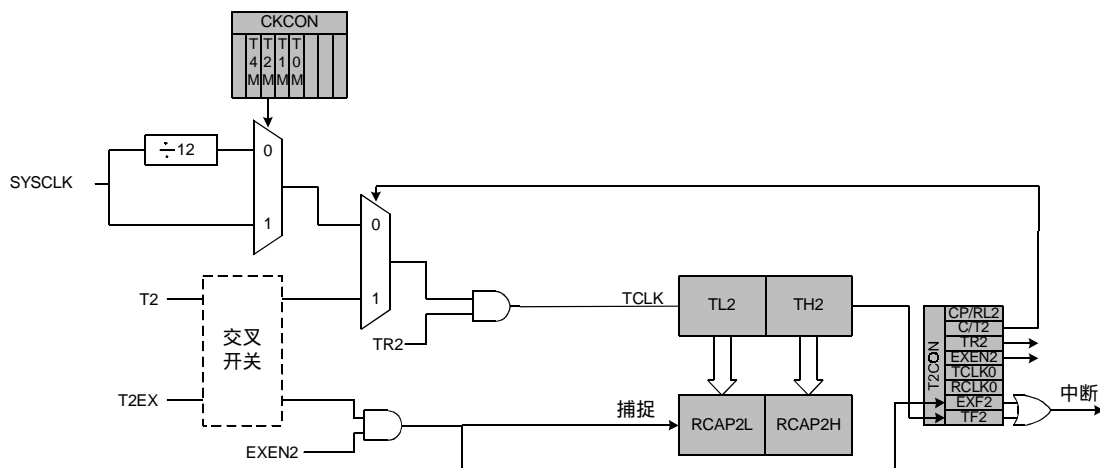
在该方式，定时器 2 被作为具有捕捉能力的 16 位计数器/定时器使用。T2EX 输入引脚上的负跳变导致下列事件发生：

1. 定时器 2 (TH2, TL2) 中的 16 位计数值被装入到捕捉寄存器 (RCAP2H, RCAP2L)。
2. 定时器 2 外部标志 (EXF2) 被置 ‘1’。
3. 产生定时器 2 中断 (如被允许)。

当工作在带捕捉的计数器/定时器方式时，定时器 2 可以使用 SYSCLK、SYSCLK/12 或 T2 输入引脚上的负跳变作为其时钟源。清除 C/T2 位 (T2CON.1) 将选择系统时钟作为定时器的输入 (由 CKCON 中的定时器时钟选择位 T2M 指定不分频或 12 分频)。当 C/T2 被置 ‘1’ 时，T2 输入引脚上的负跳变使计数器/定时器寄存器加 1。当 16 位计数器/定时器进行加 1 计数并发生溢出 (从 0xFFFF 到 0x0000) 时，定时器溢出标志 TF2 (T2CON.7) 被置位并产生一个中断 (如果中断被允许)。

通过置 ‘1’ 捕捉/重装选择位 CP/RL2 (T2CON.0) 和定时器 2 运行控制位 TR2 (T2CON.2) 来选择带捕捉的计数器/定时器方式。定时器 2 外部允许 EXEN2 (T2CON.3) 也必须被设置为逻辑 1 以允许捕捉。如果 EXEN2 被清除，T2EX 上的电平变化将被忽略。

图 22.11 T2 方式 0 原理框图

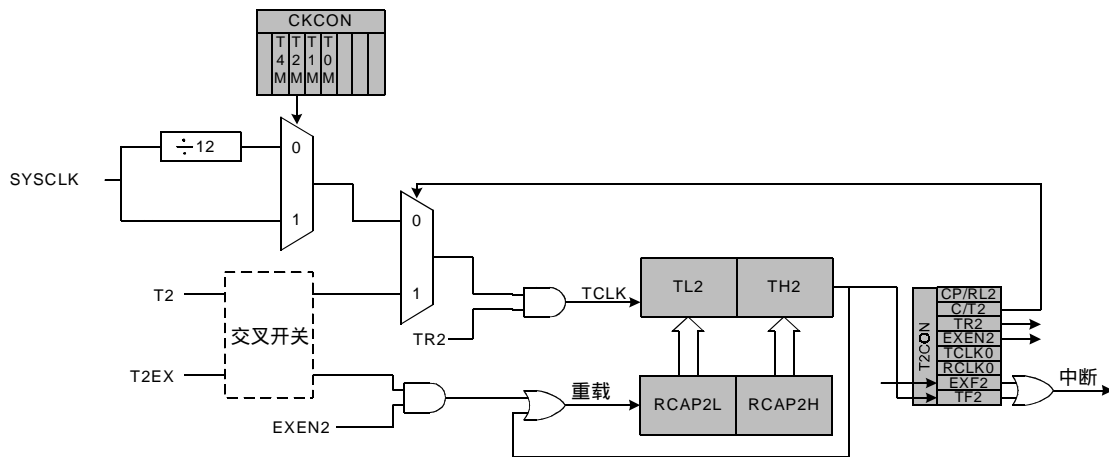


22.2.2 方式 1：自动重载的 16 位计数器/定时器

当计数器/定时器寄存器发生溢出（从 0xFFFF 到 0x0000）时，自动重载方式的计数器/定时器将定时器溢出标志 TF2 置‘1’。如果中断被允许，将产生一个中断。溢出时，两个捕捉寄存器（RCAP2H，RCAP2L）中的 16 位计数初值被自动装入到计数器/定时器寄存器，定时器重新开始计数。

清除 CP/RL2 位将选择自动重载的计数器/定时器方式。设置 TR2 为逻辑 1 允许并启动定时器 2。定时器 2 既可以选择系统时钟也可以选择外部输入引脚（T2）上的电平变化作为其时钟源，由 C/T2 位指定。如果 EXEN2 位被设置为逻辑 1，T2EX 上的负跳变将导致定时器 2 被重新装载，如果中断被允许，将产生一个中断。如果 EXEN2 被清 0，T2EX 上的电平变化将被忽略。

图 22.12 T2 方式 1 原理框图



22.2.3 方式 2: 波特率发生器

当 UART0 工作于方式 1 或方式 3 时, 定时器 2 可以用作 UART0 的波特率发生器 (有关 UART0 工作方式的详细信息见“20.1 UART0 工作方式”)。定时器 2 的波特率发生器方式与自动重载方式相似。在溢出时, 两个捕捉寄存器 (RCAP2H, RCAP2L) 中的 16 位计数初值被自动装入到计数器/定时器寄存器。但是 TF2 溢出标志不置位, 也不产生中断。溢出事件用作 UART 的移位时钟输入。定时器 2 溢出可以用于产生独立的发送和/或接收波特率。

设置 RCLK0 (T2CON.5) 和/或 TCLK0 (T2CON.4) 为逻辑 1 将选择波特率发生器方式。当 RCLK0 或 TCLK0 被设置为逻辑 1 时, 定时器 2 工作在自动重载方式, 与 CP/RL2 位的状态无关。注意: 在波特率发生器方式, 定时器 2 的时基信号为系统时钟/2。当被选择为 UART0 的波特率时钟源时, 定时器 2 定义 UART0 的波特率如下:

$$\text{波特率} = \text{SYSCLK} / ((65536 - [\text{RCAP2H:RCAP2L}]) * 32)$$

如果需要不同的时基信号, 可以通过将 C/T2 位设置为逻辑 ‘1’ 选择外部引脚 T2 上的输入作为时基。在这种情况下, UART 的波特率计算公式为:

$$\text{波特率} = F_{\text{CLK}} / ((65536 - [\text{RCAP2H:RCAP2L}]) * 16)$$

其中, F_{CLK} 为加在定时器 2 的信号的频率, 而 [RCAP2H:RCAP2L] 为捕捉寄存器中的 16 位数值。

如前所述, 定时器 2 工作在波特率发生器方式时不能置位 TF2 溢出标志, 因而不能产生中断。但是, 如果 EXEN2 位被设置为逻辑 ‘1’, 则 T2EX 输入引脚上的负跳变将置位 EXF2 标志, 并产生一个定时器 2 中断 (如果中断被允许)。因此, T2EX 输入可以被用作额外的外部中断源。

图 22.13 T2 方式 2 原理框图

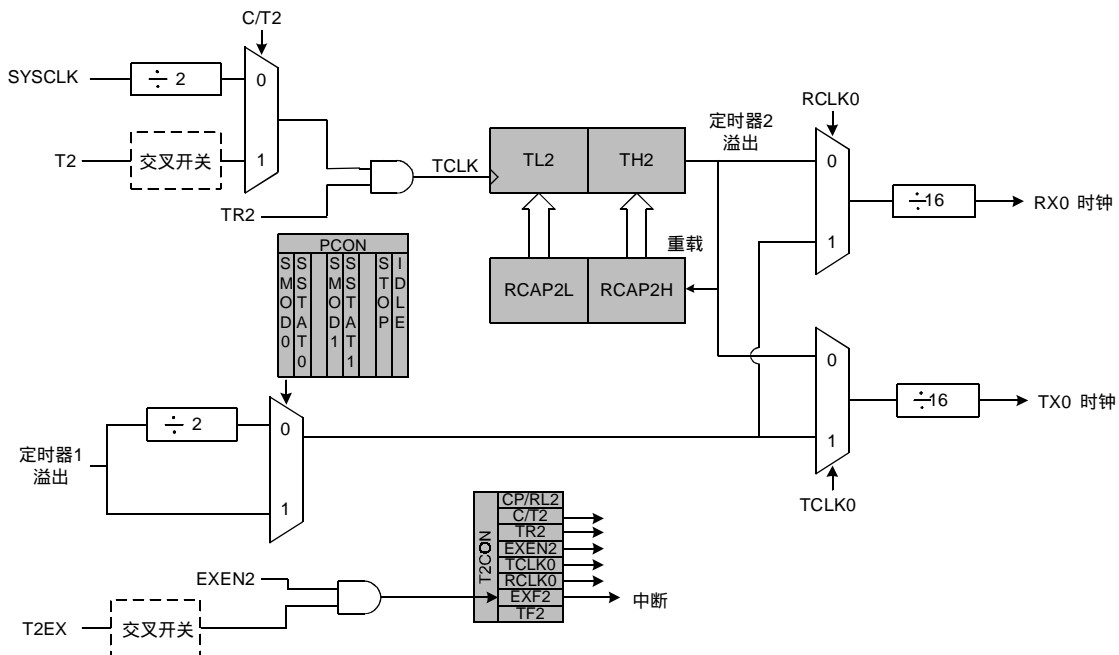


图 22.14 T2CON: 定时器 2 控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|------|-------|-------|-------|-----|------|--------------|----------------|
| TF2 | EXF2 | RCLK0 | TCLK0 | EXEN2 | TR2 | C/T2 | CP/RL2 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0xC8 |
| <p>位 7: TF2: 定时器 2 溢出标志 当定时器 2 溢出时由硬件置位。当定时器 2 中断被允许时, 该位置 1 导致 CPU 转向定时器 2 的中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。 当 RCLK0 和 / 或 TCLK0 为逻辑 1 时, TF2 不会被置 1。</p> <p>位 6: EXF2: 定时器 2 外部标志 当 T2EX 输入引脚的负跳变导致发生捕捉或重载并且 EXEN2 为逻辑 1 时, 该位由硬件置位。在定时器 2 中断被允许时, 该位置 ‘1’ 使 CPU 转向定时器 2 的中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。</p> <p>位 5: RCLK0: UART0 接收时钟标志 选择 UART0 工作在方式 1 或 3 时接收时钟使用的定时器。 0: 定时器 1 溢出作为接收时钟。 1: 定时器 2 溢出作为接收时钟。</p> <p>位 4: TCLK0: UART0 发送时钟标志 选择 UART0 工作在方式 1 或 3 时发送时钟使用的定时器。 0: 定时器 1 溢出作为发送时钟。 1: 定时器 2 溢出作为发送时钟。</p> <p>位 3: EXEN2: 定时器 2 外部允许 当定时器 2 不是工作在波特率发生器方式时, 允许 T2EX 上的负跳变触发捕捉或重载。 0: T2EX 上的负跳变被忽略。 1: T2EX 上的负跳变导致一次捕捉或重载。</p> <p>位 2: TR2: 定时器 2 运行控制 该位允许/禁止定时器 2。 0: 定时器 2 禁止。 1: 定时器 2 允许。</p> <p>位 1: C/T2: 计数器/定时器功能选择 0: 定时器功能: 定时器 2 由 T2M (CKCON.5) 定义的时钟加 1。 1: 计数器功能: 定时器 2 由外部输入引脚 (T2) 的负跳变加 1。</p> <p>位 0: CP/RL2: 捕捉/重载选择 该位选择定时器 2 为捕捉还是自动重载方式。EXEN2 必须为逻辑 1 才能使 T2EX 上的负跳变能够被识别并用于触发捕捉和重载。若 RCLK0 或 TCLK0 被置位, 该位将被忽略, 定时器 2 将工作在自动重载方式。 0: 当定时器 2 溢出或 T2EX 上发生负跳变时将自动重载 (EXEN2=1)。 1: 在 T2EX 发生负跳变时捕捉 (EXEN2=1)。</p> | | | | | | | | |

图 22.15 RCAP2L: 定时器 2 捕捉寄存器低字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xCA |

位 7-0: RCAP2L: 定时器 2 捕捉寄存器的低字节
 当定时器 2 被配置为捕捉方式时, RCAP2L 寄存器捕捉定时器 2 的低字节。当定时器 2 被配置为自动重载方式时, 它保存重载值的低字节。

图 22.16 RCAP2H: 定时器 2 捕捉寄存器高字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xCB |

位 7-0: RCAP2H: 定时器 2 捕捉寄存器的高字节
 当定时器 2 被配置为捕捉方式时, RCAP2H 寄存器捕捉定时器 2 的高字节。当定时器 2 被配置为自动重载方式时, 它保存重载值的高字节。

图 22.17 TL2: 定时器 2 低字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xCC |

位 7-0: TL2: 定时器 2 的低字节
 TL2 寄存器保存 16 位定时器 2 的低字节。

图 22.18 TH2: 定时器 2 高字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xCD |

位 7-0: TH2: 定时器 2 的高字节
 TH2 寄存器保存 16 位定时器 2 的高字节。

22.3 定时器 3

定时器 3 是一个 16 位的计数器/定时器，由两个 8 位的 SFR 组成：TMR3L（低字节）和 TMR3H（高字节）。定时器 3 的时钟输入可以是外部振荡器（8 分频）或系统时钟（不分频或 12 分频，由定时器 3 控制寄存器 TMR3CN 中的定时器 3 时钟选择位 T3M 指定）。定时器 3 总是被配置为自动重载方式定时器，重载值保存在 TMR3RLL（低字节）和 TMR3RLH（高字节）中。

定时器 3 的外部时钟源特性提供了实时时钟（RTC）方式。当 T3XCLK（TMR3CN.0）位被设置为逻辑‘1’时，定时器 3 用外部振荡器输入（8 分频）作为时钟，而与系统时钟选择无关。这种分离的时钟源允许定时器 3 使用精确的外部源，而系统时钟取自高速内部振荡器。定时器 3 可用于启动 ADC 数据转换、SMBus 定时（见“18. 系统管理总线/I2C 总线(SMBus0)”），或作为通用定时器使用。定时器 3 没有计数器方式。

图 22.19 定时器 3 原理框图

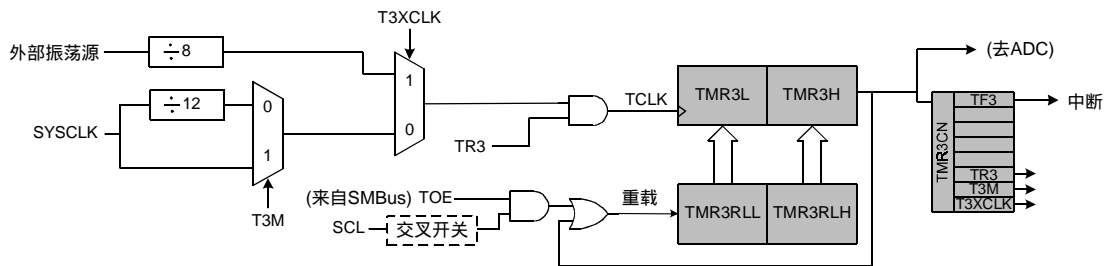


图 22.20 TMR3CN: 定时器 3 控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--|-----|-----|-----|-----|-----|-----|--------|----------------|
| TF3 | - | - | - | - | TR3 | T3M | T3XCLK | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0x91 |
| <p>位 7: TF3: 定时器 3 溢出标志 当定时器 3 从 0xFFFF 到 0x0000 溢出时由硬件置位。当定时器 3 中断被允许时, 该位置 ‘1’ 使 CPU 转向定时器 3 的中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。</p> <p>位 6-3: 未用。读=0000b, 写=忽略。</p> <p>位 2: TR3: 定时器 3 运行控制 该位允许/禁止定时器 3 0: 定时器 3 禁止。 1: 定时器 3 允许。</p> <p>位 1: T3M: 定时器 3 时钟选择 该位控制提供给计数器/定时器 3 的系统时钟的分频数。 0: 计数器/定时器 3 使用系统时钟的 12 分频。 1: 计数器/定时器 3 使用系统时钟。</p> <p>位 0: T3XCLK: 定时器 3 外部时钟选择 该位选择外部振荡器输入的 8 分频作为定时器 3 的时钟源。当 T3XCLK 为逻辑 ‘1’ 时, T3M (TMR3CN.1) 位被忽略。 0: 定时器 3 的时钟源由 T3M (TMR3CN.1) 位定义。 1: 定时器 3 的时钟源外部振荡器输入的 8 分频。</p> | | | | | | | | |

图 22.21 TMR3RLL: 定时器 3 重载寄存器低字节

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|-----|-----|-----|-----|-----|-----|-----|----------|
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| <p>位 7-0: TMR3RLL: 定时器 3 重载寄存器的低字节 定时器 3 被配置为自动重装载定时器。该寄存器保存重载值的低字节。</p> | | | | | | | | |

图 22.22 TMR3RLH: 定时器 3 重载寄存器高字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0x93 |

位 7-0: TMR3RLH: 定时器 3 重载寄存器的高字节
定时器 3 配置为自动重装载定时器。该寄存器保存重载值的高字节。

图 22.23 TMR3L: 定时器 3 低字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0x94 |

位 7-0: TMR3L: 定时器 3 的低字节
TMR3L 寄存器为定时器 3 的低字节。

图 22.24 TMR3H: 定时器 3 高字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0x95 |

位 7-0: TMR3H: 定时器 3 的高字节
TMR3H 寄存器为定时器 3 的高字节。

22.4 定时器 4

定时器 4 是一个 16 位的计数器/定时器，由两个 8 位的 SFR 组成：TL4（低字节）和 TH4（高字节）。与定时器 0 和定时器 1 一样，它既可以使用系统时钟也可以使用一个外部输入引脚（T4）上的状态变化作为时钟源。计数器/定时器选择位 C/T4（T4CON.1）选择定时器 4 的时钟源。清除 C/T4 将选择系统时钟作为定时器的输入（由 CKCON 中的定时器时钟选择位 T4M 指定不分频或 12 分频）。当 C/T4 被置 ‘1’ 时，T2 输入引脚上的负跳变使计数器/定时器寄存器加 1。（有关选择和配置外部输入引脚的详细信息见“17.1 端口 0– 端口 3 和有优先级交叉开关译码器”。）定时器 4 还可用于启动 ADC 数据转换。

定时器 4 提供了定时器 0 和定时器 1 所不具备的功能。它有三种工作方式：带捕捉的 16 位计数器/定时器、自动重载的 16 位计数器/定时器和波特率发生器方式。通过设置定时器 4 控制（T4CON）寄存器中的配置位来选择定时器 4 的工作方式。下面是定时器 4 工作方式和用于配置计数器/定时器的配置位的一览表，后面将对每种工作方式进行详细说明。

| RCLK1 | TCLK1 | CP / RL4 | TR4 | 方式 |
|-------|-------|----------|-----|-------------------|
| 0 | 0 | 1 | 1 | 带捕捉的 16 位计数器/定时器 |
| 0 | 0 | 0 | 1 | 自动重载的 16 位计数器/定时器 |
| 0 | 1 | X | 1 | UART1 波特率发生器 |
| 1 | 0 | X | 1 | UART1 波特率发生器 |
| 1 | 1 | X | 1 | UART1 波特率发生器 |
| X | X | X | 0 | 关闭 |

22.4.1 方式 0：带捕捉的 16 位计数器/定时器

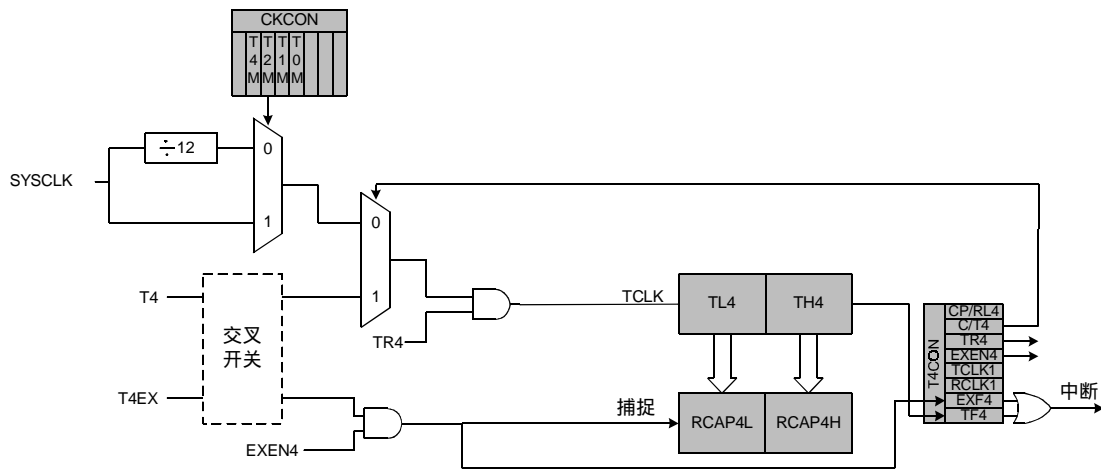
在该方式，定时器 4 被作为具有捕捉能力的 16 位计数器/定时器使用。T4EX 输入引脚上的负跳变导致下列事件发生：

1. 定时器 4 (TH4, TL4) 中的 16 位计数值被装入到捕捉寄存器 (RCAP4H, RCAP4L)。
2. 定时器 4 外部标志 (EXF2) 被置 ‘1’。
3. 产生定时器 4 中断 (如被允许)。

当工作在捕捉方式时，定时器 4 可以使用 SYSCLK、SYSCLK/12 或外部 T4 输入引脚上的负跳变作为其时钟源。清除 C/T4 位 (T4CON.1) 将选择系统时钟作为定时器的输入 (由 CKCON 中的定时器时钟选择位 T4M 指定不分频或 12 分频)。当 C/T4 被置 ‘1’ 时，T4 输入引脚上的负跳变使计数器/定时器寄存器加 1。当 16 位计数器/定时器进行加 1 计数并发生溢出 (从 0xFFFF 到 0x0000) 时，定时器溢出标志 TF4 (T4CON.7) 被置位并产生一个中断 (如果中断被允许)。

通过置 ‘1’ 捕捉/重装选择位 CP/RL4 (T4CON.0) 和定时器 4 运行控制位 TR4 (T4CON.4) 来选择带捕捉的计数器/定时器方式。定时器 4 外部允许 EXEN4 (T4CON.3) 必须被设置为逻辑 1 以允许捕捉。如果 EXEN4 被清除，T4EX 上的电平变化将被忽略。

图 22.25 T4 方式 0 原理框图

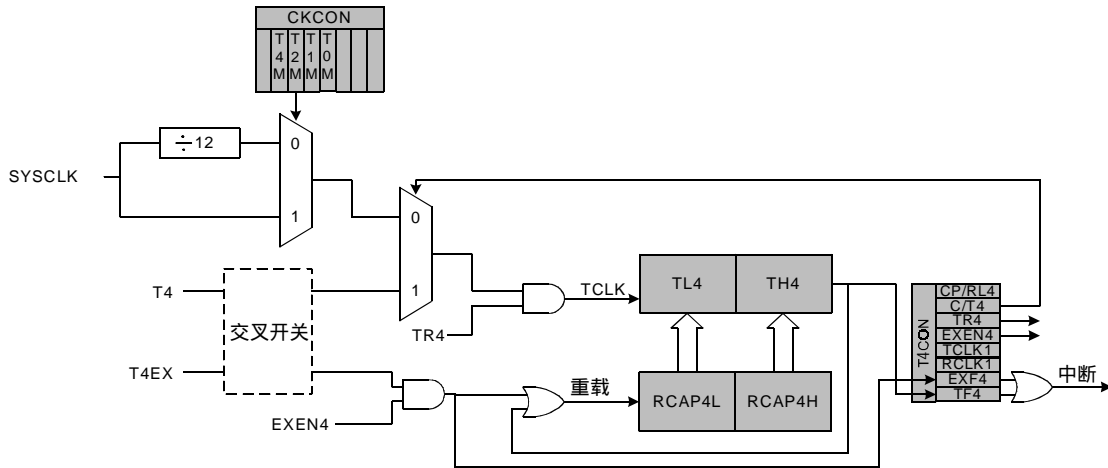


22.4.2 方式 1: 自动重载的 16 位计数器/定时器

当计数器/定时器寄存器发生溢出（从 0xFFFF 到 0x0000）时，自动重载方式的计数器/定时器将定时器溢出标志 TF4 置‘1’。如果中断被允许，将产生一个中断。溢出时，两个捕捉寄存器（RCAP4H, RCAP4L）中的 16 位计数初值被自动装入到计数器/定时器寄存器，定时器重新开始计数。

清除 CP/RL4 位将选择自动重载的计数器/定时器方式。设置 TR4 为逻辑 1 允许并启动定时器。定时器 4 既可以选择系统时钟也可以选择外部输入引脚（T4）上的电平变化作为其时钟源，由 C/T4 位指定。如果 EXEN4 位被设置为逻辑 1，T4EX 上的负跳变将导致定时器 4 被重新装载，如果中断被允许，将产生一个中断。如果 EXEN4 被清 0，T4EX 上的电平变化将被忽略。

图 22.26 T4 方式 1 原理框图



22.4.3 方式 2: 波特率发生器

当 UART1 工作于方式 1 或方式 3 时, 定时器 4 可以用作 UART1 的波特率发生器 (有关 UART1 工作方式的详细信息见“21.1 UART1 工作方式”)。定时器 4 的波特率发生器方式与自动重装载方式相似。在溢出时, 两个捕捉寄存器 (RCAP4H, RCAP4L) 中的 16 位计数初值被自动装入到计数器/定时器寄存器。但是 TF4 溢出标志不置位, 也不产生中断。溢出事件用作 UART1 的移位时钟输入。定时器 4 溢出可以用于产生独立的发送和/或接收波特率。

设置 RCLK1 (T4CON.5) 和/或 TCLK1 (T4CON.4) 为逻辑 1 将选择波特率发生器方式。当 RCLK1 或 TCLK1 被设置为逻辑 1 时, 定时器 4 工作在自动重装载方式, 与 CP/RL4 位的状态无关。注意: 在波特率发生器方式, 定时器 4 的时基信号为系统时钟/2。当被选择为 UART1 的波特率时钟源时, 定时器 4 定义 UART1 的波特率如下:

$$\text{波特率} = \text{SYSCLK} / ((65536 - [\text{RCAP4H}:\text{RCAP4L}]) * 32)$$

如果需要不同的时基信号, 可以通过将 C/T4 位设置为逻辑 ‘1’ 选择外部引脚 T4 上的输入作为时基。在这种情况下, UART1 的波特率计算公式为:

$$\text{波特率} = F_{\text{CLK}} / ((65536 - [\text{RCAP4H}:\text{RCAP4L}]) * 16)$$

其中, F_{CLK} 为加在定时器 4 的信号的频率, 而[RCAP4H:RCAP4L]为捕捉寄存器中的 16 位数值。

如前所述, 定时器 4 工作在波特率发生器方式时不能置位 TF4 溢出标志, 因而不能产生中断。但是, 如果 EXEN4 位被设置为逻辑 ‘1’, 则 T4EX 输入引脚上的负跳变将置位 EXF4 标志, 并产生一个定时器 4 中断 (如果中断被允许)。因此, T4EX 输入可以被用作额外的外部中断源。

图 22.27 T4 方式 2 原理框图

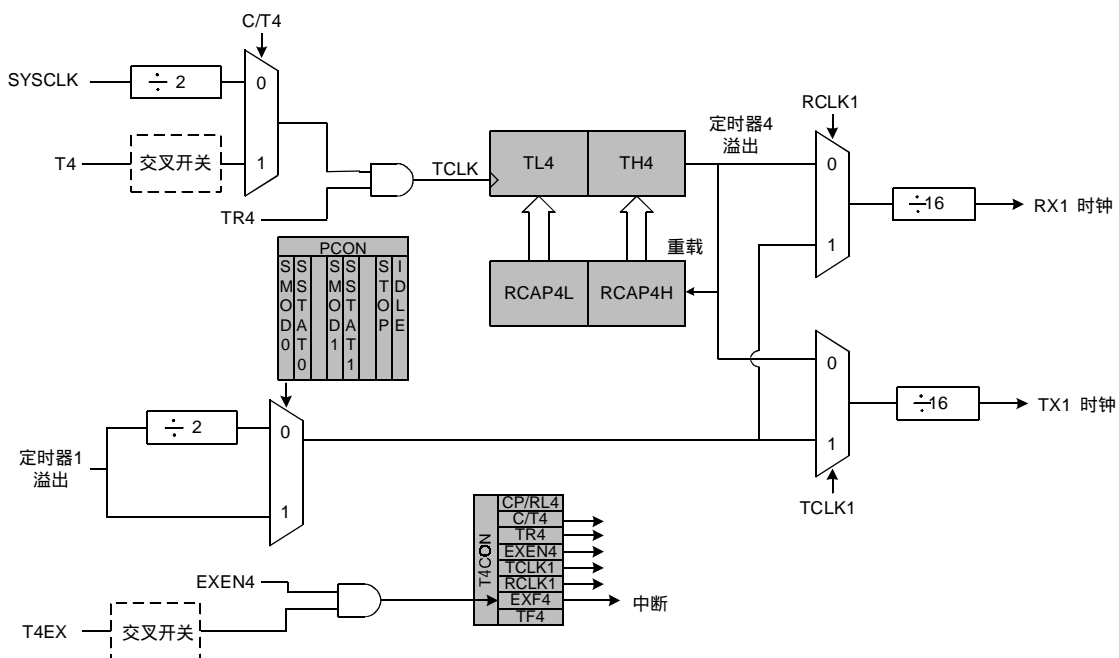


图 22.28 T4CON: 定时器 4 控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|---|------|-------|-------|-------|-----|------|--------|----------------|
| TF4 | EXF4 | RCLK1 | TCLK1 | EXEN4 | TR4 | C/T4 | CP/RL4 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xC9 |
| <p>位 7: TF4: 定时器 4 溢出标志 当定时器 4 从 0xFFFF 到 0x0000 溢出时由硬件置位。当定时器 4 中断被允许时, 该位置 1 导致 CPU 转向定时器 4 的中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。当 RCLK1 和 / 或 TCLK1 为逻辑 1 时, TF4 不会被置位。</p> <p>位 6: EXF4: 定时器 4 外部标志 当 T4EX 输入引脚的负跳变导致发生捕捉或重载并且 EXEN4 为逻辑 1 时, 该位由硬件置位。在定时器 4 中断被允许时, 该位置 ‘1’ 使 CPU 转向定时器 4 的中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。</p> <p>位 5: RCLK1: UART1 接收时钟标志 选择 UART1 工作在方式 1 或 3 时接收时钟使用的定时器。 0: 定时器 1 溢出作为接收时钟。 1: 定时器 4 溢出作为接收时钟。</p> <p>位 4: TCLK1: UART1 发送时钟标志 选择 UART1 工作在方式 1 或 3 时发送时钟使用的定时器。 0: 定时器 1 溢出作为发送时钟。 1: 定时器 4 溢出作为发送时钟。</p> <p>位 3: EXEN4: 定时器 4 外部允许 当定时器 4 不是工作在波特率发生器方式时, 允许 T4EX 上的负跳变触发捕捉或重载。 0: T4EX 上的负跳变被忽略。 1: T4EX 上的负跳变导致一次捕捉或重载。</p> <p>位 2: TR4: 定时器 4 运行控制 该位允许/禁止定时器 4。 0: 定时器 4 禁止。 1: 定时器 4 允许。</p> <p>位 1: C/T4: 计数器/定时器功能选择 0: 定时器功能: 定时器 4 由 T4M (CKCON.6) 定义的时钟触发加 1。 1: 计数器功能: 定时器 4 由外部输入引脚 (T4) 的负跳边触发加 1。</p> <p>位 0: CP/RL4: 捕捉/重载选择 该位选择定时器 4 为捕捉还是自动重装载方式。EXEN4 必须为逻辑 1 才能使 T4EX 上的负跳变能够被识别并用于触发捕捉和重载。若 RCLK1 或 TCLK1 被置位, 该位将被忽略, 定时器 4 将工作在自动重装载方式。 0: 当定时器 4 溢出或 T4EX 上发生负跳变时将自动重装载 (EXEN4=1)。 1: 在 T4EX 发生负跳变时捕捉 (EXEN4=1)。</p> | | | | | | | | |

图 22.29 RCAP4L: 定时器 4 捕捉寄存器低字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xE4 |

位 7-0: RCAP4L: 定时器 4 捕捉寄存器的低字节
 当定时器 4 被配置为捕捉方式时, RCAP4L 寄存器捕捉定时器 4 的低字节。当定时器 4 被配置为自动重载方式时, 它保存重载值的低字节。

图 22.30 RCAP4H: 定时器 4 捕捉寄存器高字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xE5 |

位 7-0: RCAP4H: 定时器 4 捕捉寄存器的高字节
 当定时器 4 被配置为捕捉方式时, RCAP4H 寄存器捕捉定时器 4 的高字节。当定时器 4 被配置为自动重载方式时, 它保存重载值的高字节。

图 22.31 TL4: 定时器 4 低字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xF4 |

位 7-0: TL4: 定时器 4 的低字节
 TL4 寄存器保存 16 位定时器 4 的低字节。

图 22.32 TH4: 定时器 4 高字节

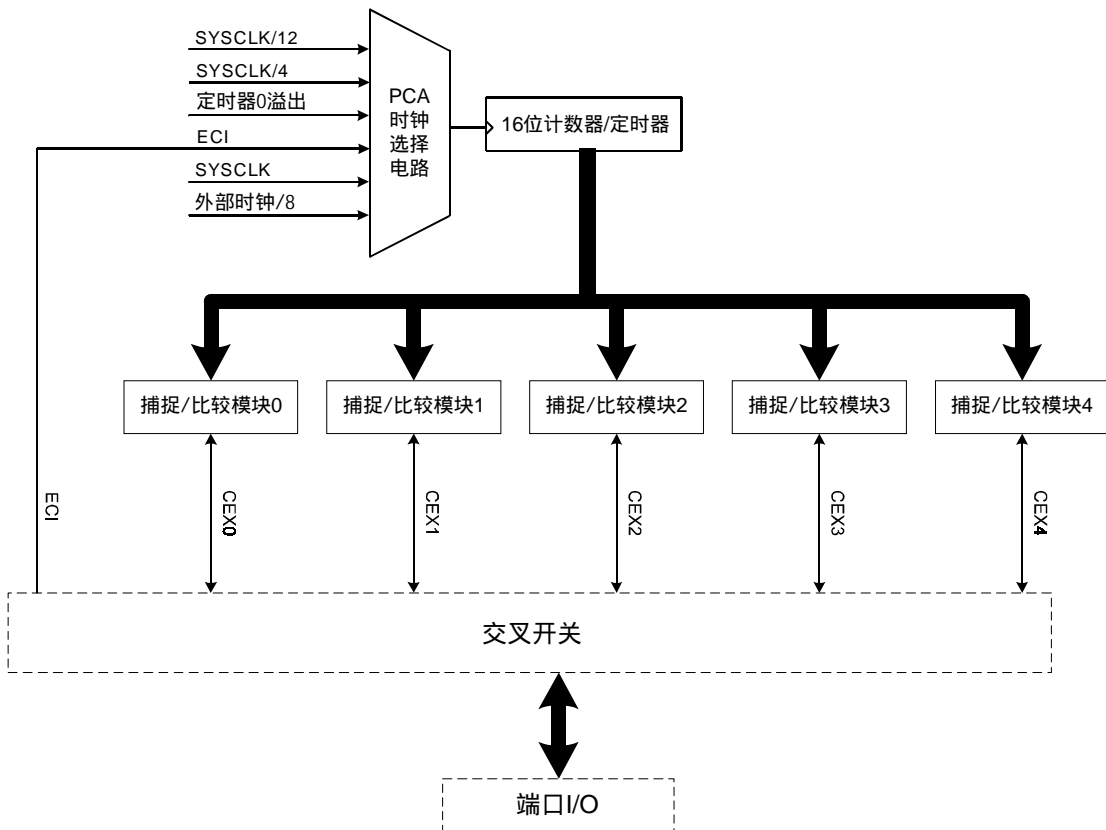
| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xF5 |

位 7-0: TH4: 定时器 4 的高字节
 TH4 寄存器保存 16 位定时器 4 的高字节。

23. 可编程计数器阵列

可编程计数器阵列 (PCA0) 提供增强的定时器功能, 与标准8051计数器/定时器相比, 它需要较少的CPU干预。PCA0包含一个专用的16位计数器/定时器和5个16位捕捉/比较模块。每个捕捉/比较模块有其自己的I/O线 (CEX_n)。当被允许时, I/O线通过交叉开关连到端口I/O (见“17.1 端口0 – 端口3和优先级交叉开关译码器”)。计数器/定时器由一个可编程的时基信号驱动, 时基信号有六个输入源: 系统时钟、系统时钟/4、系统时钟/12、外部振荡器时钟源8分频、定时器0溢出、ECI线上的外部时钟信号。每个捕捉/比较模块可以被编程为独立工作在下面的6种工作方式之一: 边沿触发捕捉、软件定时器、高速输出、频率输出、8位PWM或16位PWM (23.2节对每种方式进行说明)。对PCA的编程和控制是通过系统控制器的特殊功能寄存器来实现的。PCA的基本原理框图示于图23.1。

图 23.1 PCA 原理框图



23.1 PCA 计数器/定时器

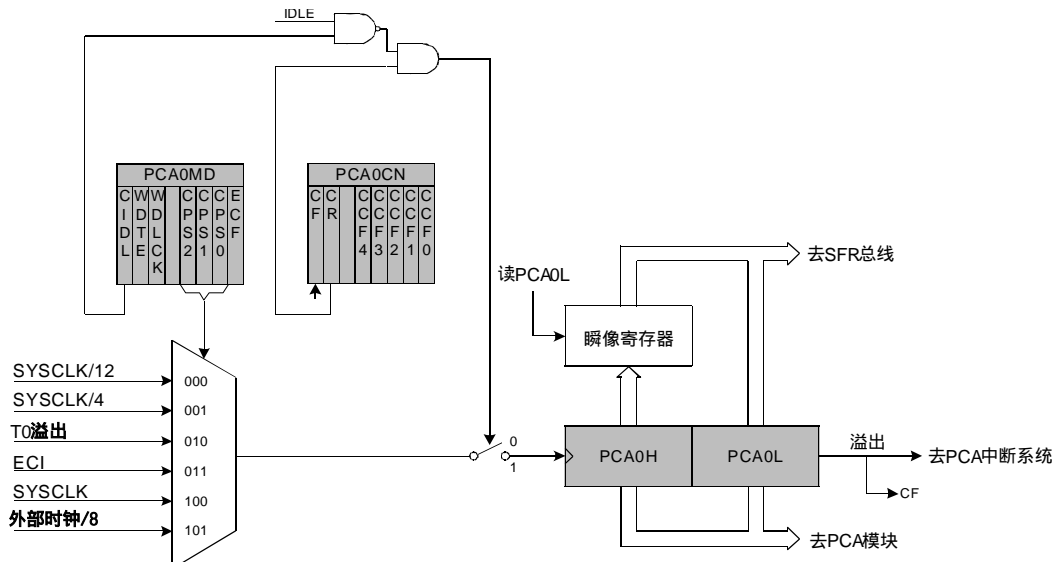
16 位的 PCA 计数器/定时器由两个 8 位的 SFR 组成：PCA0L 和 PCA0H。PCA0H 是 16 位计数器/定时器的高字节 (MSB)，而 PCA0L 是低字节 (LSB)。在读 PCA0L 的同时自动锁存 PCA0H 的值。先读 PCA0L 寄存器将使 PCA0H 的值得到保持 (在读 PCA0L 的同时)，直到用户读 PCA0H 寄存器为止。读 PCA0H 或 PCA0L 不影响计数器工作。PCA0MD 寄存器中的 CPS2-CPS0 位用于选择 PCA 计数器/定时器的时基信号，如表 23.1 所示。

当计数器/定时器溢出时 (从 0xFFFF 到 0x0000)，PCA0MD 中的计数器溢出标志 (CF) 被置为逻辑 1 并产生一个中断请求 (如果 CF 中断被允许)。将 PCA0MD 中 ECF 位设置为逻辑 1 即可允许 CF 标志产生中断请求。当 CPU 转向中断服务程序时，CF 位不能被硬件自动清除，必须用软件清 0。(注意：要使 CF 中断得到响应，必须先总体允许 PCA0 中断。通过将 EA 位 (IE.7) 和 EPCA0 位 (EIE1.3) 设置为逻辑 1 来总体允许 PCA0 中断。)清除 PCA0MD 寄存器中的 CIDL 位将允许 PCA 在微控制器内核处于等待方式时继续正常工作。

表 23.1. PCA 时基输入选择

| CPS2 | CPS1 | CPS0 | 时间基准 |
|------|------|------|---------------------------|
| 0 | 0 | 0 | 系统时钟的 12 分频 |
| 0 | 0 | 1 | 系统时钟的 4 分频 |
| 0 | 1 | 0 | 定时器 0 溢出 |
| 0 | 1 | 1 | ECI 负跳变 (最大速率 = 系统时钟频率/4) |
| 1 | 0 | 0 | 系统时钟 |
| 1 | 0 | 1 | 外部振荡源 8 分频 |

图 23.2 PCA 计数器/定时器原理框图



23.2 捕捉/比较模块

每个模块都可被配置为独立工作，有六种工作方式：边沿触发捕捉、软件定时器、高速输出、频率输出、8位脉宽调制器和16位脉宽调制器。每个模块在CIP-51系统控制器中都有属于自己的特殊功能寄存器（SFR）。这些寄存器用于配置模块的工作方式和与模块交换数据。

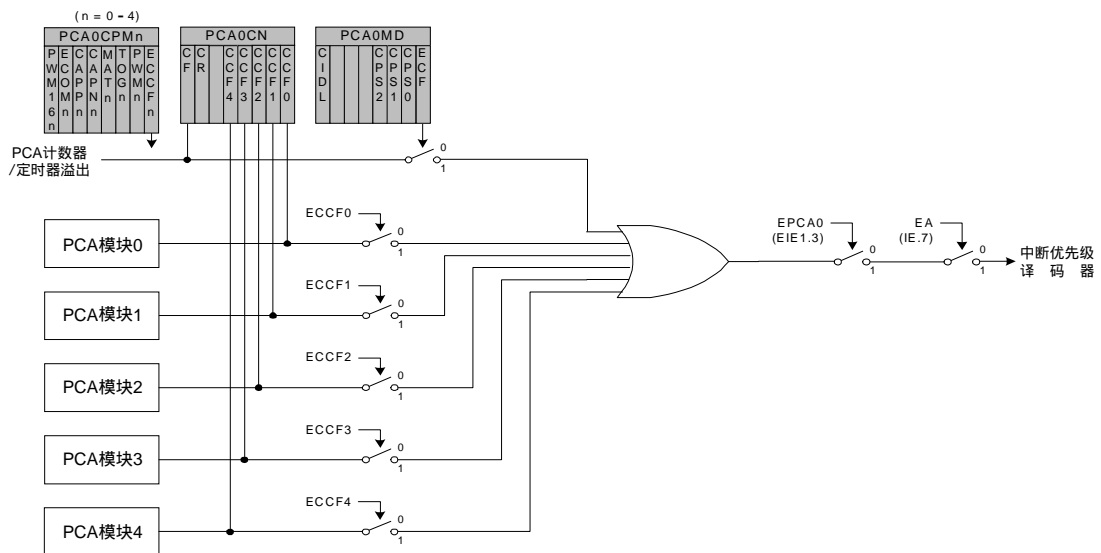
PCA0CPMn寄存器用于配置PCA捕捉/比较模块的工作方式，表23.2概述了模块工作在不同方式时该寄存器各位的设置情况。置‘1’ PCA0CPMn寄存器中的ECCFn位将允许模块的CCFn中断。注意：要使单独的CCFn中断得到响应，必须先整体允许PCA0中断。通过将EA位（IE.7）和EPCA0位（EIE1.3）设置为逻辑1来整体允许PCA0中断。PCA0中断配置的详细信息见图23.3。

表 23.3. PCA 捕捉/比较模块的 PCA0CPM 寄存器设置

| PWM16 | ECOM | CAPP | CAPN | MAT | TOG | PWM | ECCF | 工作方式 |
|-------|------|------|------|-----|-----|-----|------|------------------|
| X | X | 1 | 0 | 0 | 0 | 0 | X | 用 CEXn 的正沿触发捕捉 |
| X | X | 0 | 1 | 0 | 0 | 0 | X | 用 CEXn 的负沿触发捕捉 |
| X | X | 1 | 1 | 0 | 0 | 0 | X | 用 CEXn 的电平改变触发捕捉 |
| X | 1 | 0 | 0 | 1 | 0 | 0 | X | 软件定时器 |
| X | 1 | 0 | 0 | 1 | 1 | 0 | X | 高速输出 |
| X | 1 | 0 | 0 | X | 1 | 1 | X | 频率输出 |
| 0 | 1 | 0 | 0 | X | 0 | 1 | X | 8 位脉冲宽度调制器 |
| 1 | 1 | 0 | 0 | X | 0 | 1 | X | 16 位脉冲宽度调制器 |

X = 忽略

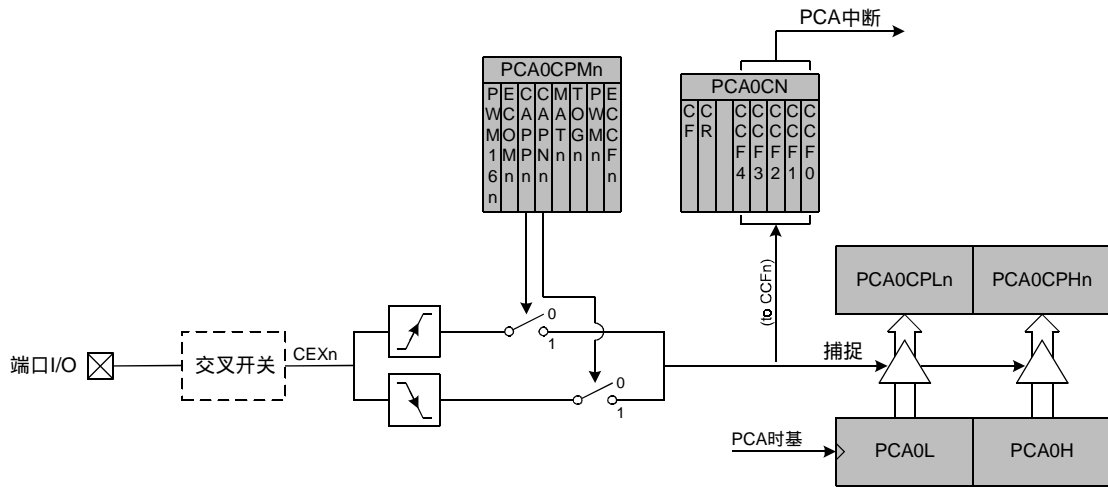
图 23.3 PCA 中断原理框图



23.2.1 边沿触发的捕捉方式

在该方式，CEX_n引脚上出现的有效电平变化导致PCA0捕捉PCA0计数器/定时器的值并将其装入到对应模块的16位捕捉/比较寄存器（PCA0CPL_n和PCA0CPH_n）。PCA0CPM_n寄存器中的CAP_{Pn}和CAP_{Nn}位用于选择触发捕捉的电平变化类型：低电平到高电平（正沿）、高电平到低电平（负沿）或任何一种变化（正沿或负沿）。当捕捉发生时，PCA0CN中的捕捉/比较标志（CCF_n）被置为逻辑1并产生一个中断请求（如果CCF中断被允许）。当CPU转向中断服务程序时，CCF_n位不能被硬件自动清除，必须用软件清0。

图 23.4 PCA 捕捉方式原理框图

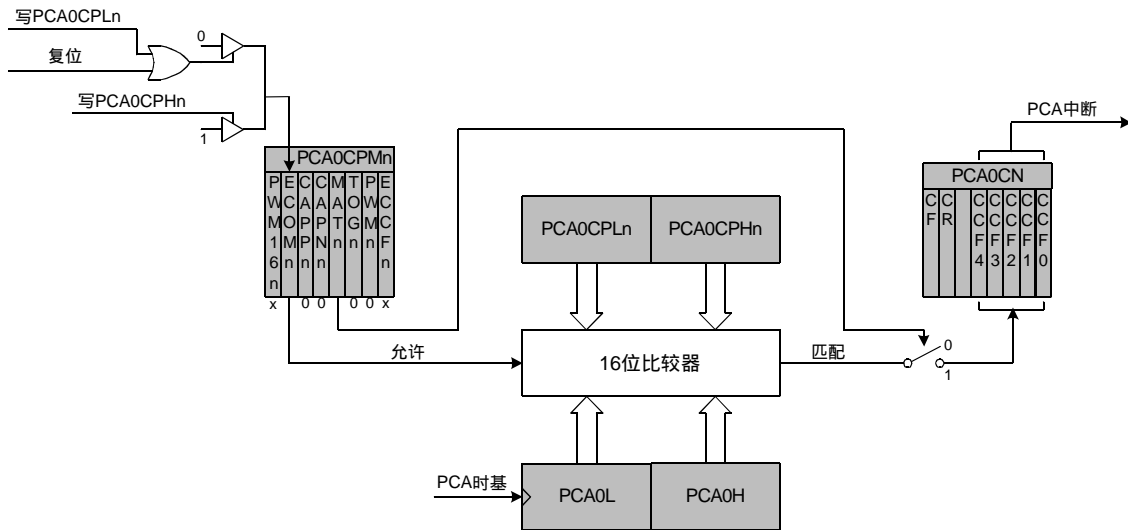


23.2.2 软件定时器（比较）方式

在软件定时器方式，系统将PCA0计数器/定时器与模块的16位捕捉/比较寄存器（PCA0CPHn和PCA0CPLn）进行比较。当发生匹配时，PCA0CN中的捕捉/比较标志（CCFn）被置为逻辑1并产生一个中断请求（如果CCF中断被允许）。当CPU转向中断服务程序时，CCFn位不能被硬件自动清除，必须用软件清0。置‘1’ PCA0CPMn寄存器中的ECOMn和MATn位将允许软件定时器方式。

关于捕捉/比较寄存器的重要注意事项：当向PCA0的捕捉/比较寄存器写入一个16位值时，应先写低字节。向PCA0CPLn的写入操作将清‘0’ ECOMn位；向PCA0CPHn写入时将置‘1’ ECOMn位。

图 23.5 PCA 软件定时器方式原理框图

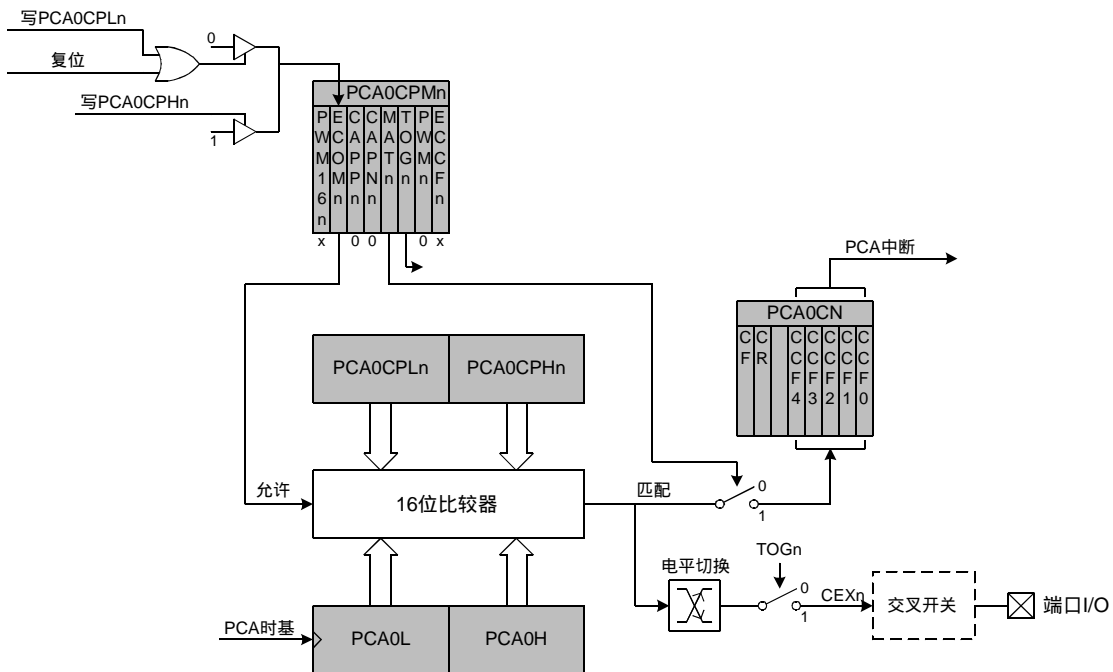


23.2.3 高速输出方式

在高速输出方式，每当PCA的计数器与模块的16位捕捉/比较寄存器（PCA0CPLn和PCA0CPHn）发生匹配时，模块的CEXn引脚上的逻辑电平将发生改变。置‘1’ PCA0CPMn寄存器中的TOGn、MATn和ECOMn位将允许高速输出方式。

关于捕捉/比较寄存器的重要注意事项：当向PCA0的捕捉/比较寄存器写入一个16位数值时，应先写低字节。向PCA0CPLn的写入操作将清‘0’ ECOMn位；向PCA0CPHn写入时将置‘1’ ECOMn位。

图 23.6 PCA 高速输出方式原理框图



23.2.4 频率输出方式

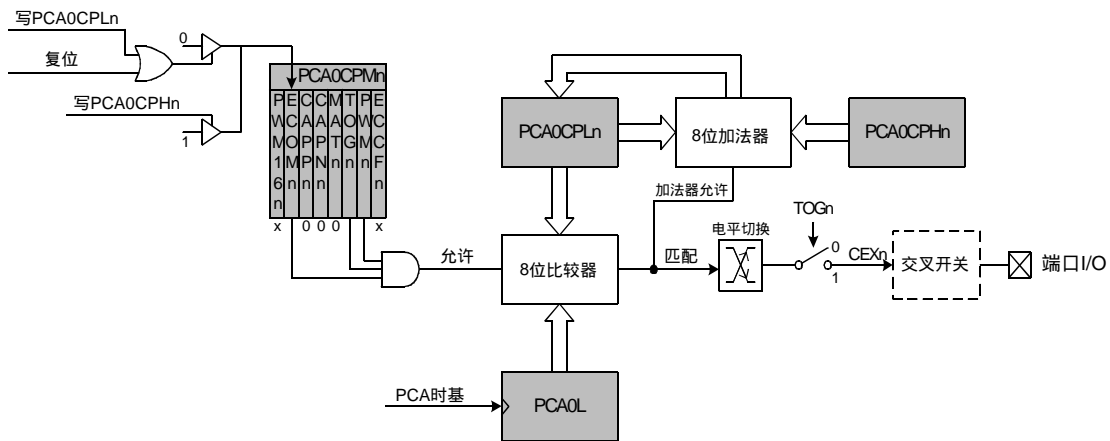
频率输出方式在对应的CEX_n引脚产生可编程频率的方波。捕捉/比较寄存器的高字节保持着输出电平改变前要计的PCA时钟数。所产生的方波的频率由方程23.1定义。

方程23.1 方波频率输出

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

其中： F_{PCA} 是由PCA方式寄存器PCA0MD中的CPS2-0位选择的PCA时钟的频率。捕捉/比较模块的低字节与PCA0计数器的低字节比较；两者匹配时，CEX_n的电平发生改变，高字节中的偏移值被加到PCA0CPL_n。注意：在该方式下如果允许模块匹配（CCF_n）中断，则发生中断的速率为 $2 * F_{CEXn}$ 。通过置位PCA0CPM_n寄存器中ECOM_n、TOG_n和PWM_n位来允许频率输出方式。

图 23.7 PCA 频率输出方式原理框图



23.2.5 8 位脉宽调制器方式

每个模块都可以独立地用于在对应的CEX_n引脚产生脉宽调制（PWM）输出。PWM输出信号的频率取决于PCA0计数器/定时器的时基。使用模块的捕捉/比较寄存器PCA0CPL_n改变PWM输出信号的占空比。当PCA0计数器/定时器的低字节（PCA0L）与PCA0CPL_n中的值相等时，CEX_n的输出被置‘1’。当PCA0L中的计数值溢出时，CEX_n输出被置为低电平（见图23.8）。当计数器/定时器的低字节PCA0L溢出时（从0xFF到0x00），保存在PCA0CPH_n中的值被自动装入PCA0CPL_n，不需软件干预。置‘1’ PCA0CPM_n寄存器中的ECOM_n和PWM_n位将允许8位脉冲宽度调制器方式。8位PWM方式的占空比由方程23.2给出。

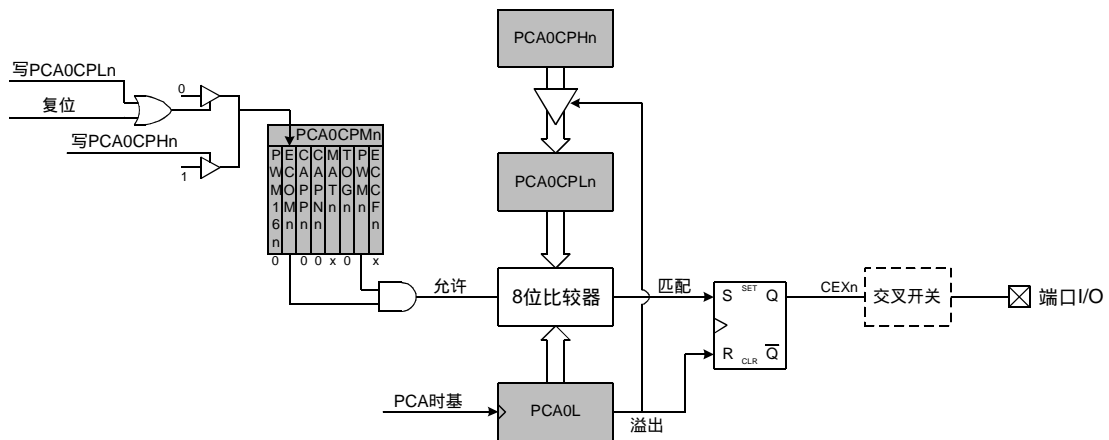
关于捕捉/比较寄存器的重要注意事项：当向PCA0的捕捉/比较寄存器写入一个16位数值时，应先写低字节。向PCA0CPL_n的写入操作将清‘0’ ECOM_n位；向PCA0CPH_n写入时将置‘1’ ECOM_n位。

方程23.2 8位PWM的占空比

$$\text{占空比} = \frac{(256 - \text{PCA0CPH}_n)}{256}$$

由方程23.2可知，最大占空比为100%（PCA0CPH_n = 0），最小占空比为0.39%（PCA0CPH_n = 0xFF）。可以通过清‘0’ ECOM_n位产生0%的占空比。

图 23.8 PCA 的 8 位 PWM 方式原理框图



23.2.6 16 位脉宽调制器方式

每个PCA0模块都可以工作在16位PWM方式。在该方式下，16位捕捉/比较模块定义PWM信号低电平时间的PCA0时钟数。当PCA0计数器与模块的值匹配时，CEX_n的输出被置‘1’；当计数器溢出时，CEX_n输出被置为低电平。为了输出一个占空比可变的波形，新值的写入应与PCA0 CCF_n匹配中断同步。置‘1’ PCA0CPM_n寄存器中的ECOM_n、PWM_n和PWM16_n位将允许16位脉冲宽度调制器方式。为了输出一个占空比可变的波形，应将CCF_n设置为逻辑‘1’以允许匹配中断。16位PWM方式的占空比由方程23.3给出。

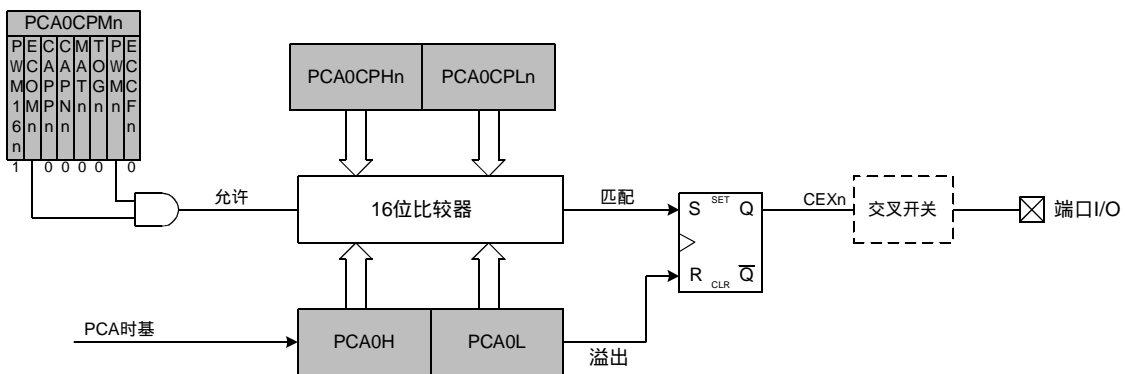
关于捕捉/比较寄存器的重要注意事项：当向PCA0的捕捉/比较寄存器写入一个16位数值时，应先写低字节。向PCA0CPL_n的写入操作将清‘0’ ECOM_n位；向PCA0CPH_n写入时将置‘1’ ECOM_n位。

方程23.3 16位PWM的占空比

$$\text{占空比} = \frac{(65536 - \text{PCA0CPn})}{65536}$$

由方程23.3可知，最大占空比为100% (PCA0CP_n = 0)，最小占空比为0.0015% (PCA0CP_n = 0xFFFF)。可以通过清‘0’ ECOM_n位产生0%的占空比。

图 23.9 PCA 的 16 位 PWM 方式原理框图



23.3 PCA0 的寄存器说明

下面对与 PCA0 工作有关的特殊功能寄存器进行详细说明。

图 23.10 PCA0CN: PCA 控制寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--|-----|-----|------|------|------|------|--------------|----------------|
| CF | CR | - | CCF4 | CCF3 | CCF2 | CCF1 | CCF0 | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 (可位寻址) | SFR地址: 0xD8 |
| <p>位 7: CF: PCA 计数器 / 定时器溢出标志 当 PCA0 计数器 / 定时器从 0xFFFF 到 0x0000 溢出时由硬件置位。在计数器 / 定时器溢出 (CF) 中断被允许时, 该位置 '1' 将导致 CPU 转向 CF 中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。</p> <p>位 6: CR: PCA0 计数器 / 定时器运行控制 该位允许 / 禁止 PCA0 计数器 / 定时器。 0: 禁止 PCA0 计数器 / 定时器 1: 允许 PCA0 计数器 / 定时器</p> <p>位 5: 未用。读=0b, 写=忽略。</p> <p>位 4: CCF4: PCA0 模块 4 捕捉 / 比较标志 在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时, 该位置 '1' 将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。</p> <p>位 3: CCF3: PCA0 模块 3 捕捉 / 比较标志 在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时, 该位置 '1' 将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。</p> <p>位 2: CCF2: PCA0 模块 2 捕捉 / 比较标志 在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时, 该位置 '1' 将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。</p> <p>位 1: CCF1: PCA0 模块 1 捕捉 / 比较标志 在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时, 该位置 '1' 将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。</p> <p>位 0: CCF0: PCA0 模块 0 捕捉 / 比较标志 在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时, 该位置 '1' 将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。</p> | | | | | | | | |

图 23.11 PCA0MD: PCA0 方式选择寄存器

| | | | | | | | | |
|------|-----|-----|-----|------|------|------|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| CIDL | - | - | - | CPS2 | CPS1 | CPS0 | ECF | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xD9 |

位 7: CIDL: PCA0 计数器 / 定时器等待控制
 规定 CPU 等待方式下的 PCA0 工作方式。
 0: 当系统控制器处于等待方式时, PCA0 继续正常工作。
 1: 当系统控制器处于等待方式时, PCA0 停止工作。

位 6-4: 未用。读=000b, 写=忽略。

位 3-1: CPS2-CPS0: PCA0 计数器 / 定时器脉冲选择
 这些位选择 PCA0 计数器的时基。

| CPS2 | CPS1 | CPS0 | 时间基准 |
|------|------|------|-------------------------|
| 0 | 0 | 0 | 系统时钟的 12 分频 |
| 0 | 0 | 1 | 系统时钟的 4 分频 |
| 0 | 1 | 0 | 定时器 0 溢出 |
| 0 | 1 | 1 | ECI 负跳变 (最大速率 = 系统时钟/4) |
| 1 | 0 | 0 | 系统时钟 |
| 1 | 0 | 1 | 外部时钟 8 分频 |
| 1 | 1 | 0 | 保留 |
| 1 | 1 | 1 | 保留 |

位 0: ECF: PCA 计数器 / 定时器溢出中断允许
 该位是 PCA0 计数器 / 定时器溢出 (CF) 中断的屏蔽位。
 0: 禁止 CF 中断。
 1: 当 CF (PCA0CN.7) 置位时, 允许 PCA0 计数器 / 定时器溢出中断请求。

图 23.12 PCA0CPMn: PCA0 捕捉 / 比较寄存器

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
|--|-------|-------|-------|------|------|------|-------|---------------------|
| PWM16n | ECOMn | CAPPn | CAPNn | MATn | TOGn | PWMn | ECCFn | 00000000 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | SFR地址: 0xDA-0xDE |
| <p>PCA0CPMn 地址: PCA0CPM0=0xDA(n=0) PCA0CPM1=0xDB(n=1) PCA0CPM2=0xDC(n=2) PCA0CPM3=0xDD(n=3) PCA0CPM4=0xDE(n=4)</p> | | | | | | | | |
| <p>位 7: PWM16n: 16 位脉冲宽度调制允许 当脉冲宽度调制方式被允许时 (PWMn = 1), 该位选择 16 位方式。 0: 选择 8 位 PWM。 1: 选择 16 位 PWM。</p> | | | | | | | | |
| <p>位 6: ECOMn: 比较器功能允许 该位允许 / 禁止 PCA0 模块 n 的比较器功能。 0: 禁止。1: 允许。</p> | | | | | | | | |
| <p>位 5: CAPPn: 正沿捕捉功能允许 该位允许 / 禁止 PCA0 模块 n 的正边沿捕捉。 0: 禁止。1: 允许。</p> | | | | | | | | |
| <p>位 4: CAPNn: 负沿捕捉功能允许 该位允许 / 禁止 PCA0 模块 n 的负边沿捕捉。 0: 禁止。1: 允许。</p> | | | | | | | | |
| <p>位 3: MATn: 匹配功能允许 该位允许 / 禁止 PCA0 模块 n 的匹配功能。如果被允许, 当 PCA0 计数器与一个模块的捕捉 / 比较寄存器匹配时, PCA0MD 寄存器中的 CCFn 位置位。 0: 禁止。 1: 允许。</p> | | | | | | | | |
| <p>位 2: TOGn: 电平切换功能允许 该位允许 / 禁止 PCA0 模块 n 的电平切换功能。如果被允许, 当 PCA0 计数器与一个模块的捕捉 / 比较寄存器匹配时, CEXn 引脚的逻辑电平切换。如果 PWMn 位也被置为逻辑 '1', 则模块工作在频率输出方式。 0: 禁止。 1: 允许。</p> | | | | | | | | |
| <p>位 1: PWMn: 脉宽调制方式允许 该位允许 / 禁止 PCA0 模块的 PWM 功能。如果被允许, CEXn 引脚输出脉冲宽度调制信号。如果 PWM16n 为逻辑 '0', 使用 8 位 PWM 方式; 如果 PWM16n 为逻辑 '1', 使用 16 位方式。如果 TOGn 位也被置为逻辑 '1', 则模块工作在频率输出方式。 0: 禁止。 1: 允许。</p> | | | | | | | | |
| <p>位 0: ECCFn: 捕捉 / 比较标志中断允许 该位设置捕捉 / 比较标志 (CCFn) 的中断屏蔽。 0: 禁止 CCFn 中断 1: 当 CCFn 位被置 1 时, 允许捕捉 / 比较标志的中断请求。</p> | | | | | | | | |

图 23.13 PCA0L: PCA0 计数器 / 定时器低字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xE9 |

位 7-0: PCA0L: PCA0 计数器 / 定时器的低字节
PCA0L 寄存器保存 16 位 PCA0 计数器 / 定时器的低字节 (LSB)。

图 23.14 PCA0H: PCA0 计数器 / 定时器高字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|----------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xF9 |

位 7-0: PCA0H: PCA0 计数器 / 定时器高字节
PCA0H 寄存器保存 16 位 PCA0 计数器 / 定时器的高字节 (MSB)。

图 23.15 PCA0CPLn: PCA 捕捉模块低字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|---------------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xEA-0xEE |

PCA0CPLn: PCA0CPL0 = 0xEA (n=0)
PCA0CPL1 = 0xEB (n=1)
PCA0CPL2 = 0xEC (n=2)
PCA0CPL3 = 0xED (n=3)
PCA0CPL4 = 0xEE (n=4)

位 7-0: PCA0CPLn: PCA0 捕捉模块低字节
PCA0CPLn 寄存器保存 16 位捕捉模块 n 的低字节 (LSB)。

图 23.16 PCA0CPHn: PCA0 捕捉模块高字节

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|---------------------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 00000000 |
| | | | | | | | | SFR地址: 0xFA-0xFE |

PCA0CPHn: PCA0CPH0=0xFA (n=0)
PCA0CPH1=0xFB (n=1)
PCA0CPH2=0xFC (n=2)
PCA0CPH3=0xFD (n=3)
PCA0CPH4=0xFE (n=4)

位 7-0: PCA0CPHn: PCA0 捕捉模块高字节
PCA0CPHn 寄存器保存 16 位捕捉模块 n 的高字节 (MSB)。

24. JTAG (IEEE 1149.1)

每个 MCU 都有一个片内 JTAG 接口和逻辑，提供生产和在系统测试所需要的边界扫描功能，支持闪存的读和写操作以及非侵入式在系统调试。MCU 中的 JTAG 接口完全符合 IEEE 1149.1 规范。关于测试接口和边界扫描结构方面的详细信息请参见该规范。在 IEEE 1149.1 规范的测试接口和操作部分介绍了如何访问 JTAG 指令寄存器 (IR) 和数据寄存器 (DR)。

JTAG 接口使用 MCU 上的四个专用引脚，它们是：TCK、TMS、TDI 和 TDO。

通过 16 位 JTAG 指令寄存器 (IR) 可以发出图 24.1 所示的 8 种指令。MCU 中有三个与 JTAG 边界扫描相关的数据寄存器和四个与 FLASH 读/写操作相关的寄存器。

图 24.1 IR: JTAG 指令寄存器

| | | | | | | | | | | | | | | | |
|---|--------------------|---|--|--|--|--|--|--|--|--|--|--|--|--|--|
| <div style="display: flex; justify-content: space-between; align-items: center;"> 位15 复位值 0x0000 </div> | | | | | | | | | | | | | | | |
| IR 值 | 指令 | 说明 | | | | | | | | | | | | | |
| 0x0000 | EXTEST | 选择用于控制和观察所有器件引脚的边界数据寄存器。 | | | | | | | | | | | | | |
| 0x0002 | SAMPLE/ PRELOAD | 选择用于观察和预设扫描路径锁存器的边界数据寄存器。 | | | | | | | | | | | | | |
| 0x0004 | IDCODE | 选择器件的 ID 寄存器。 | | | | | | | | | | | | | |
| 0xFFFF | BYPASS | 选择旁路数据寄存器。 | | | | | | | | | | | | | |
| 0x0082 | 闪存控制 | 选择 FLASHCON 寄存器，以控制接口逻辑对读和写 FLASHDAT 寄存器如何响应。 | | | | | | | | | | | | | |
| 0x0083 | 闪存数据 | 选择用于读和写闪存存储器的 FLASHDAT 寄存器。 | | | | | | | | | | | | | |
| 0x0084 | 闪存地址 | 选择 FLASHADR 寄存器，该寄存器保持 FLASH 读、写和擦除操作的地址。 | | | | | | | | | | | | | |
| 0x0085 | 闪存定标 | 选择 FLASHSCL 寄存器，该寄存器控制 FLASH 单稳态定时和常读允许。 | | | | | | | | | | | | | |

24.1 边界扫描

边界扫描路径中的数据寄存器是一个 134 位的移位寄存器。通过执行 EXTEST 和 SAMPLE 命令，边界数据寄存器能提供对所有器件引脚以及 SFR 总线和弱上拉功能的控制和观察。

表 24.1 边界数据寄存器位定义

EXTEST 支持捕捉和更新两种操作，而 SAMPLE 只执行捕捉操作。

| 位 | 操作 | 目标 |
|------------------------------|----|---|
| 0 | 捕捉 | 来自 MCU 的复位允许 (C8051F021/3) |
| | 更新 | 去/RST 引脚的复位允许 (C8051F021/3) |
| 1 | 捕捉 | 来自/RST 引脚的复位输入 (C8051F021/3) |
| | 更新 | 去/RST 引脚的复位输出 (C8051F021/3) |
| 2 | 捕捉 | 来自 MCU 的复位允许 (C8051F020/2) |
| | 更新 | 去/RST 引脚的复位允许 (C8051F020/2) |
| 3 | 捕捉 | 来自/RST 引脚的复位输入 (C8051F020/2) |
| | 更新 | 去/RST 引脚的复位输出 (C8051F020/2) |
| 4 | 捕捉 | 来自 XTAL1 引脚的外部时钟 |
| | 更新 | 未用 |
| 5 | 捕捉 | 来自 MCU 的弱上拉允许 |
| | 更新 | 去端口引脚的弱上拉允许 |
| 6,8,10,12, 14,16,18,20 | 捕捉 | 来自 MCU 的 P0.n 输出允许 (如 Bit6=P0.0, Bit8=P0.1 等) |
| | 更新 | 去引脚的 P0.n 输出允许 (如 Bit6=P0.0oe, Bit8=P0.1oe 等) |
| 7,9,11,13, 15,17,19,21 | 捕捉 | 来自引脚的 P0.n 输入 (如 Bit7=P0.0, Bit9=P0.1 等) |
| | 更新 | P0.n 输出到引脚 (如 Bit7=P0.0, Bit9=P0.1 等) |
| 22,24,26,28, 30,32,34,36 | 捕捉 | 来自 MCU 的 P1.n 输出允许 |
| | 更新 | 去引脚的 P1.n 输出允许 |
| 23,25,27,29, 31,33,35,37 | 捕捉 | 来自引脚的 P1.n 输入 |
| | 更新 | P1.n 输出到引脚 |
| 38,40,42,44, 46,48,50,52 | 捕捉 | 来自 MCU 的 P2.n 输出允许 |
| | 更新 | 去引脚的 P2.n 输出允许 |
| 39,41,43,45, 47,49,51,53 | 捕捉 | 来自引脚的 P2.n 输入 |
| | 更新 | P2.n 输出到引脚 |
| 54,56,58,60, 62,64,66,68 | 捕捉 | 来自 MCU 的 P3.n 输出允许 |
| | 更新 | 去引脚的 P3.n 输出允许 |
| 55,57,59,61, 63,65,67,69 | 捕捉 | 来自引脚的 P3.n 输入 |
| | 更新 | P3.n 输出到引脚 |
| 70,72,74,76, 78,80,82,84 | 捕捉 | 来自 MCU 的 P4.n 输出允许 |
| | 更新 | 去引脚的 P4.n 输出允许 |
| 71,73,75,77, 79,81,83,85 | 捕捉 | 来自引脚的 P4.n 输入 |
| | 更新 | P4.n 输出到引脚 |
| 86,88,90,92, 94,96,98,100 | 捕捉 | 来自 MCU 的 P5.n 输出允许 |
| | 更新 | 去引脚的 P5.n 输出允许 |
| 87,89,91,93, 95,97,99,101 | 捕捉 | 来自引脚的 P5.n 输入 |
| | 更新 | P5.n 输出到引脚 |

表 24.1 边界数据寄存器位定义（续）

| | | |
|-------------------------------------|----|--------------------|
| 102,104,106,108, 110,112,114,116 | 捕捉 | 来自 MCU 的 P6.n 输出允许 |
| | 更新 | 去引脚的 P6.n 输出允许 |
| 103,105,107,109, 111,113,115,117 | 捕捉 | 来自引脚的 P6.n 输入 |
| | 更新 | P6.n 输出到引脚 |
| 118,120,122,124, 126,128,130,132 | 捕捉 | 来自 MCU 的 P7.n 输出允许 |
| | 更新 | 去引脚的 P7.n 输出允许 |
| 119,121,123,125, 127,129,131,133 | 捕捉 | 来自引脚的 P7.n 输入 |
| | 更新 | P7.n 输出到引脚 |

24.1.1 EXTEST 指令

通过 IR 进入 EXTEST 指令。边界数据寄存器提供对所有器件引脚以及弱上拉功能的控制和观察。片内逻辑的所有输入都被设置为逻辑 ‘1’。

24.1.2 SAMPLE 指令

通过 IR 进入 SAMPLE 指令。边界数据寄存器提供对扫描路径锁存器的观察和预置。

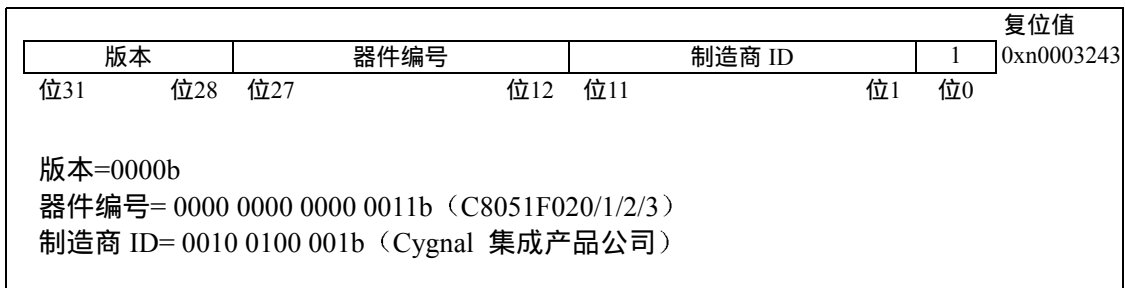
24.1.3 BYPASS 指令

通过 IR 进入 BYPASS 指令。它提供对标准 JTAG 旁路数据寄存器的访问。

24.1.4 IDCODE 指令

通过 IR 进入 IDCODE 指令。它提供对 32 位器件 ID 寄存器的访问。

图 24.2 DEVICEID: JTAG 器件 ID 寄存器



24.2 闪存编程命令

通过 JTAG 接口可以直接对闪存存储器编程，编程时要使用闪存控制、闪存数据、闪存地址和闪存定时寄存器。这些间接数据寄存器是通过 JTAG 指令寄存器访问的。对间接数据寄存器进行读和写操作时，首先要在 IR 寄存器中设置正确的数据寄存器地址。每次读或写都是通过向所选择的数据寄存器写入适当的间接操作码 (IndOpCode) 来启动的。输入到该寄存器的命令具有如下格式：

| | |
|--------------------|--------------------------|
| 19:18 IndOpCode | 17:0 WriteData (待写数据) |
|--------------------|--------------------------|

IndOpCode: 这些位根据下表来设置要执行的操作：

| IndOpCode | 操作 |
|-----------|----|
| 0x | 查询 |
| 10 | 读 |
| 11 | 写 |

查询操作用于检查下面将要介绍的 Busy (忙) 位。查询操作执行一次 Capture_DR (数据寄存器捕捉)，但不允许 Update_DR (数据寄存器更新) 操作。由于更新操作被禁止，查询操作只进行一个二进制位的移入/移出。

读操作启动一次对由 DRAdress 寻址的寄存器的读取。只需向间接寄存器移入两位即可启动读操作。在读操作被启动后，必须通过查询 Busy 位来确定该操作何时完成。

写操作启动一次对由 DRAdress 寻址的寄存器的写入，写入数据为 WriteData。可以写任何数据长度不超过 18 位的寄存器。如果待写寄存器的数据不足 18 位，应对位于 WriteData 字段的数据进行左对齐，即 MSB 应占据位 17。这就允许以较少的 JTAG 时钟周期对较短的寄存器进行写入。例如，只需进行 10 次移位即可完成对一个 8 位寄存器的写入。在写操作被启动后，必须通过查询 Busy 位来确定何时能启动下一次操作。在读或写操作正在进行时，不应改变指令寄存器中的内容。

从间接数据寄存器输出的数据具有如下格式：

| | | |
|---------|--------------------------|---------------|
| 19 0 | 18:1 ReadData (读出的数据) | 0 Busy (忙) |
|---------|--------------------------|---------------|

Busy 位指示当前操作是否完成。它在操作被启动后变高，在操作完成后变低。在 Busy 为高时，读和写命令均被忽略。实际上，如果需要在查到 Busy 位为低电平之后进行另一次读或写操作，则下一操作的 JTAG 写可以在查询 Busy 位是否为低电平时进行。该操作将被忽略，直到 Busy 位变低为止，此时将启动新操作。该位处于最低位 (位 0)，只需移位一次即可对其查询。当等待一个读操作完成而 Busy 位为 0 时，可以移出后面的 18 位以得到结果数据。ReadData (读出的数据) 总是右对齐的，这就允许以较少的移位次数读取长度小于 18 位的寄存器。例如，只需进行 9 次移位 (Busy + 8 位) 即可得到一次字节读的结果。

图 24.5 FLASHDAT: JTAG 闪存数据寄存器

| | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|------|-------|------------|
| DATA7 | DATA6 | DATA5 | DATA4 | DATA3 | DATA2 | DATA1 | DATA0 | FAIL | FBUSY | 复位值 |
| 位9 | 位8 | 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 0000000000 |

该寄存器用于通过 JTAG 接口从闪存中读数据或写数据到闪存。

位 9-2: DATA7-0: 闪存数据字节

位 1: FAIL: 闪存操作失败标志
 0: 闪存存储器操作成功。
 1: 闪存存储器操作失败。通常指示相关存储器单元被锁定。

位 0: FBUSY: 闪存忙标志
 0: 闪存接口逻辑不忙。
 1: 闪存接口逻辑正在处理一个请求。当 FBUSY=1 时, 读或写都不会启动另一个操作。

图 24.6 FLASHSCL: JTAG 闪存定时寄存器

| | | | | | | | | |
|------|------|----|----|----|----|----|------|----------|
| FOSE | FRAE | 保留 | 保留 | 保留 | 保留 | 保留 | FLWE | 复位值 |
| 位7 | 位6 | 位5 | 位4 | 位3 | 位2 | 位1 | 位0 | 10000000 |

位 7: FOSE: 闪存单稳态定时器允许位
 单稳态定时器用于在一次 FLASH 读操作结束后关闭读出放大器。
 0: 禁止闪存单稳态定时器。
 1: 允许闪存单稳态定时器。

位 6: FRAE: 闪存常读允许
 0: 读操作发生在每个单稳态定时间隔。
 1: 闪存总是处于读方式。

位 5-1: 未用。读=00000b, 写入值必须是 00000b。

位 0: FLWE: 闪存写/擦除允许
 要从用户软件写或擦除 FLASH 时, 必须将该位置 '1'。
 0: 禁止 FLASH 写操作。
 1: 允许 FLASH 写操作。

24.3 调试支持

每个 MCU 内部都有 JTAG 和调试电路，可以通过 JTAG 接口使用安装在最终应用系统上的产品 MCU 进行非侵入式、全速、在系统调试。Cygna1 的调试系统支持观察和修改存储器和寄存器、断点和单步执行；不需要额外的目标 RAM、程序存储器或通信通道。在调试时，所有的模拟和数字外设都全功能正确运行（保持同步）。当 MCU 因单步执行或执行到断点而停机时，WDT 被禁止。

开发套件 C8051F020DK 适用于 C8051F020 系列的所有 MCU，具有开发应用代码和进行在系统调试所需要的全部硬件和软件。每个套件包括一个具有调试器和 8051 汇编器的集成开发环境（IDE）。套件中还包括一个被称为串口适配器的 RS232 到 JTAG 的接口模块，还有一个安装有 C8051F020 的目标应用板（板上有大块试验区）。每个套件还包括 RS-232 和 JTAG 电缆及墙装电源。