C8051F93x-C8051F92x
低功耗混合信号 ISP FLASH 微控制器
数　据　手　册

潘 琢 金　译

Rev 0.31　2008.03

版权所有

版 权 声 明

本手册中文版版权归译者和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。

原文中比较明显的错误已经在译文中更正。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 计算机学院　　潘琢金

电话：024-89723476，13066535936
Email：panzhuojin@sina.com 或 panzhj@syiae.edu.cn

C8051F93X- C8051F92X　混合信号 ISP FLASH 微控制器

供电电压 0.9V ~ 3.6V

- 单节电池模式支持 0.9~1.8V 操作
- 双节电池模式支持 1.8~3.6V 操作
- 内建 DC-DC 转换器，可输出 1.8~3.3V，用于单节电池模式
- 内建 LDO 稳压器，允许高模拟电压和低数字内核电压
- 两个内建欠压检测器

10 位 ADC

- ±1LSB INL；无失码
- 可编程转换速率，最高 300ksps
- 可多达 23 个外部输入
- 片内 1.65V±TBD 电压基准
- 片内 PGA 允许测量两倍于基准电压的电压
- 具有突发方式的 16 位自动平均累加器，提供增加的 ADC 分辨率
- 数据窗口中断发生器
- 内建温度传感器（±3°C）

两个比较器

- 可编程回差电压和响应时间
- 可配置为唤醒或复位源
- 多达 23 个容性触感输入

6 位可编程电流基准

- 达±500 μA。用作电流偏置或产生定制基准电压

在片调试

- 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/修改存储器和寄存器
- 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）
- 扩展的中断系统

存储器

- 4352 字节内部数据 RAM（256+4096）
- 64 KB（F93x）或 32 KB（F92x） FLASH；可在系统编程，扇区大小为 1024 字节

数字外设

- 24 或 16 个端口 I/O；全部耐 5 V 电压并具有大灌电流能力和可编程的驱动强度
- 可同时使用的硬件 SMBus（I^2C 兼容）、2 个 SPI 和 UART 串口
- 4 个通用 16 位计数器/定时器
- 16 位可编程计数器/定时器阵列（PCA），有 6 个捕捉/比较模块和看门狗定时器
- 硬件实时时钟（smaRTClock），工作电压可低至 0.9V，仅需要小于 0.5μ A 的供电电流

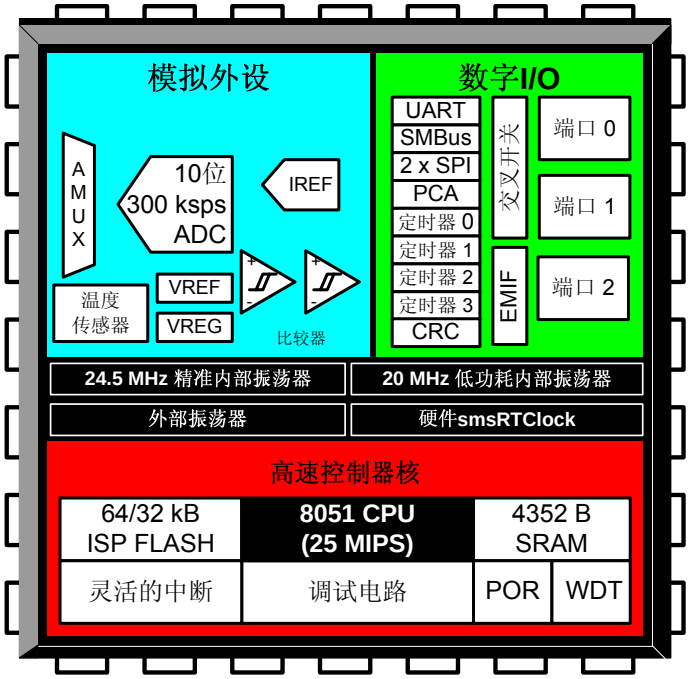
时钟源

- 内部振荡器：24.5MHz，2%精度，可支持 UART 操作；20MHz 低功耗振荡器，仅需很小的偏置电流
- 外部振荡器：晶体、RC、C、或 CMOS 时钟
- smaRTClock 振荡器：32KHz 晶体或内部
- 可在运行中切换时钟源，用于实现多种节电模式

封装

- 32 脚 QFN（5x5 mm）
- 24 脚 QFN（4x4 mm）
- 32 脚 LQFP（易于手工焊接）

温度范围：-40°C - +85°C



目 录

1 . 系统概述 9

1.1 CIP-51™ 微控制器核 12

1.1.1 与 8051 指令集完全兼容 12

1.1.2 速度提升 12

1.1.3 增加的功能 12

1.2 端口输入/输出 13

1.3 串行端口 14

1.4 可编程计数器阵列 14

1.5 10 位 SAR ADC 15

1.6 可编程电流基准（IREF0） 17

1.7 比较器 17

2. 定货信息 19

3. 引脚和封装定义 20

4. 电气特性 33

4.1 极限参数 33

4.2 总体电气特性 34

5. 10 位 SAR ADC（ADC0） 44

5.1 输出码格式 45

5.2 工作方式 46

5.2.1 启动转换 46

5.2.2 跟踪方式 47

5.2.3 突发方式 48

5.2.4 建立时间要求 49

5.2.5 增益设置 50

5.3 8 位方式 50

5.4 可编程窗口检测器 56

5.4.1 单端方式下的窗口检测器 58

5.4.2 ADC0 技术指标 59

5.5 ADC0 模拟多路选择器 59

5.6 温度传感器 61

5.6.1 校准 62

5.7 电压基准和地基准选项 64

5.8 外部电压基准 65

5.9 内部电压基准 65

5.10 模拟地基准 65

5.11 温度传感器使能 65
5.12 电压基准电气指标 66

6. 可编程电流基准（IREF0） 67

6.1 IREF0 技术指标 67

7. 比较器 68

7.1 比较器输入 68
7.2 比较器输出 69
7.3 比较器响应时间 70
7.4 比较器回差电压 70
7.5 比较器寄存器说明 71
7.6 比较器 0 和比较器 1 模拟多路器 75

8. CIP-51 微控制器 78

8.1 指令集 79
8.1.1 指令和 CPU 时序 79
8.2 CIP-51 寄存器说明 83

9. 存储器组织 86

9.1 程序存储器 87
9.1.1 MOVX 指令和程序存储器 87
9.2 数据存储器 88
9.2.1 内部 RAM 88
9.2.2 外部 RAM 89

10. 外部数据存储器接口和片内 XRAM 90

10.1 访问 XRAM 90
10.1.1 16 位 MOVX 示例 90
10.1.2 8 位 MOVX 示例 90
10.2 配置访问片外空间的外部存储器接口 91
10.3 外部存储器接口的端口输入/输出配置 91
10.4 复用的外部存储器接口 92
10.5 外部存储器接口工作模式 93
10.5.1 只用内部 XRAM 93
10.5.2 无块选择的分片模式 93
10.5.3 带块选择的分片模式 94
10.5.4 只用外部存储器 94
10.6 EMIF 时序 95
10.7 EMIF 特殊功能寄存器 96
10.8 EMIF 时序图 99
10.8.1 复用方式 16 位 MOVX：EMI0CF[3:2] = 01、10 或 11 99

10.8.2 无块选择的复用方式 8 位 MOVX：EMI0CF[3:2] = 01 或 11 100
10.8.3 带块选择的复用方式 8 位 MOVX：EMI0CF[3:2] = 10 101

11. 特殊功能寄存器 103

11.1 SFR 分页 104

12. 中断系统 109

12.1 使能中断源 109
12.2 MCU 中断源和中断向量 109
12.3 中断优先级 109
12.4 中断响应时间 110
12.5 中断寄存器说明 112
12.6 外部中断 118

13. FLASH 存储器 120

13.1 FLASH 存储器编程 120
13.1.1 FLASH 锁定和关键码功能 120
13.1.2 FLASH 擦除过程 120
13.1.3 FLASH 写过程 121
13.2 非易失性数据存储 121
13.3 安全选项 121
13.4 FLASH 写和擦除指南 124
13.4.1 VDD 维护和 VDD 监视器 124
13.4.2 PSWE 维护 125
13.4.3 系统时钟 125

14. 电源管理 128

14.1 正常方式 129
14.2 空闲方式 130
14.3 停机方式 130
14.4 挂起方式 130
14.5 休眠方式 131
14.6 配置唤醒源 131
14.7 确定引起唤醒的事件 132
14.8 电源管理单元技术指标 134

15. 循环冗余检查单元（CRC0） 135

15.1 CRC 算法 135
15.2 CRC 计算前的准备 137
15.3 执行 CRC 计算 137
15.4 访问 CRC 结果 137
15.5 CRC0 的位序反转功能 140

16. 片上 DC-DC 转换器（DC0）.....141

16.1 启动表现.....142
16.2 脉冲跳过方式.....142
16.3 使能 DC-DC 转换器.....143
16.4 DC-DC 转换器寄存器说明.....144
16.5 DC-DC 转换器技术指标.....145

17. 稳压器（VREG0）.....146

17.1 稳压器技术指标.....146

18. 复位源.....147

18.1 上电（VBAT 电源监视器）复位.....148
18.2 掉电（VDD/DC+电源监视器）复位.....149
18.3 外部复位.....151
18.4 时钟丢失检测器复位.....151
18.5 比较器 0 复位.....152
18.6 PCA 看门狗定时器复位.....152
18.7 FLASH 错误复位.....152
18.8 SMARTCLOCK（实时时钟）复位.....152
18.9 软件复位.....153

19. 时钟源.....154

19.1 可编程精准内部振荡器.....154
19.2 低功耗内部振荡器.....155
19.3 外部振荡器驱动电路.....155
19.3.1 外部晶体方式.....155
19.3.3 外部 RC 方式.....156
19.3.4 外部电容方式.....157
19.3.4 外部 CMOS 时钟方式.....158
19.4 用于选择和配置系统时钟的特殊功能寄存器.....159

20. SMARTCLOCK（实时时钟）.....162

20.1 SMARTCLOCK 接口.....163
20.1.1 smaRTClock 锁定和关键码功能.....163
20.1.2 使用 RTC0ADR 和 RTC0DAT 访问 smaRTClock 的内部寄存器.....164
20.1.3 smaRTClock 接口的自动读功能.....164
20.1.4 RTC0ADR 自动增 1 功能.....164
20.1.5 RTC0ADR 短选通功能.....164
20.2 SMARTCLOCK 时钟源.....167
20.2.1 使用 smaRTClock 振荡器的晶体方式.....167
20.2.2 使用 smaRTClock 振荡器的自振荡方式.....167

20.2.3 可编程负载电容.....168
20.2.4 自动增益控制（仅限于晶体方式）和 smaRTClock 偏置加倍.....169
20.2.5 smaRTClock 时钟丢失检测器.....170
20.2.6 smaRTClock 振荡器晶体有效检测器.....170
20.3 SMARTCLOCK 定时器和报警功能.....170
20.3.1 设置和读取 smaRTClock 定时器值.....170
20.3.2 设置 smaRTClock 告警值.....171
20.3.3 使用 smaRTClock 定时器和告警的软件考虑.....171

21. 端口输入/输出.....176

21.1 端口 I/O 的工作方式.....177
21.1.1 端口引脚配置为模拟 I/O.....177
21.1.2 端口引脚配置为数字 I/O.....177
21.1.3 端口 I/O 与 5V 逻辑的接口.....178
21.1.4 增加端口 I/O 驱动强度.....178
21.2 端口 I/O 引脚分配给模拟和数字功能.....178
21.2.1 端口引脚分配给模拟功能.....178
21.2.2 端口引脚分配给数字功能.....179
21.2.3 端口引脚分配给外部数字事件捕捉功能.....179
21.3 优先权交叉开关译码器.....180
21.4 端口匹配.....186
21.5 访问和配置端口 I/O 的特殊功能寄存器.....188

22. SMBUS.....197

22.1 支持文档.....198
22.2 SMBUS 配置.....198
22.3 SMBUS 操作.....199
22.3.1 发送器和接收器的概念.....199
22.3.2 总线仲裁.....200
22.3.3 时钟低电平扩展.....200
22.3.4 SCL 低电平超时.....200
22.3.5 SCL 高电平（SMBus 空闲）超时.....200
22.4 SMBUS 的使用.....201
22.4.1 SMBus 配置寄存器.....201
22.4.2 SMBus 控制寄存器.....205
22.4.3 硬件从地址识别.....208
22.4.4 数据寄存器.....210
22.5 SMBUS 传输方式.....211
22.5.1 写序列（主方式）.....211
22.5.2 读序列（主方式）.....212
22.5.3 写序列（从方式）.....213

22.5.4 读序列（从方式）.....214
22.6 SMBUS 状态译码.....215

23. UART0.....219

23.1 增强的波特率发生器.....220
23.2 工作方式.....221
23.2.1 8 位 UART.....221
23.2.2 9 位 UART.....222
23.3 多机通信.....223

24. 增强型串行外设接口（SPI0 和 SPI1）.....227

24.1 信号说明.....228
24.1.1 主输出、从输入（MOSI）.....228
24.1.2 主输入、从输出（MISO）.....228
24.1.3 串行时钟（SCK）.....228
24.1.4 从选择（NSS）.....228
24.2 SPI 主方式.....230
24.3 SPI 从方式.....231
24.4 SPI 中断源.....231
24.5 串行时钟相位和极性.....232
24.6 SPI 特殊功能寄存器.....234

25. 定时器.....240

25.1 定时器 0 和定时器 1.....242
25.1.1 方式 0 — 13 位计数器/定时器.....242
25.1.2 方式 1 — 16 位计数器/定时器.....243
25.1.3 方式 2 — 自动重装载的 8 位计数器/定时器.....244
25.1.4 方式 3 — 两个 8 位计数器/定时器（仅定时器 0）.....245
25.2 定时器 2.....249
25.2.1 16 位自动重装载定时器.....249
25.2.2 8 位自动重装载定时器.....250
25.2.3 比较器 0/smaRTClock 捕捉方式.....251
25.3 定时器 3.....254
25.3.1 16 位自动重装载定时器.....254
25.3.2 8 位自动重装载定时器.....255
25.3.3 比较器 1/外部振荡器捕捉方式.....256

26. 可编程计数器阵列.....259

26.1 PCA 计数器/定时器.....260
26.2 PCA0 中断源.....262
26.3 捕捉/比较模块.....263
26.3.1 边沿触发的捕捉方式.....264

26.3.2 软件定时器（比较）方式.....265
26.3.3 高速输出方式.....266
26.3.4 频率输出方式.....267
26.3.5 8 位、9 位、10 位和 11 位脉宽调制器方式.....268
26.3.6 16 位脉宽调制器方式.....270
26.4 看门狗定时器方式.....271
26.4.1 看门狗定时器操作.....271
26.4.2 看门狗定时器的使用.....272
26.5 PCA0 寄存器说明.....273

27. C2 接口.....279

27.1 C2 接口寄存器.....279
27.2 C2 引脚共享.....281

1．系统概述

C8051F93x-C8051F92x 器件是完全集成的低功耗混合信号片上系统型 MCU。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 2.1。

- 单节/双节电操作，内建 DC-DC 升压型转换器
- 高速、流水线结构的与 8051 兼容的微控制器核（可达 25MIPS）
- 全速、非侵入式的在系统调试接口（片内）
- 真 10 位、300 ksps 、23 通道单端 ADC，带模拟多路器
- 6 位可编程电流基准
- 高精度可编程的 24.5MHz 内部振荡器（使用扩频技术）
- 64KB 或 32KB 的片内 FLASH 存储器
- 4352 字节片内 RAM
- 硬件实现的 SMBus/ I^2C、增强型 UART 和两个 SPI 串行接口
- 4 个通用的 16 位定时器
- 具有 6 个捕捉/比较模块和看门狗定时器功能的可编程计数器/定时器阵列（PCA）
- 硬件实时时钟（smaRTClock），工作电压可低至 0.9V
- 硬件 CRC 引擎
- 片内上电复位、VDD 监视器和温度传感器
- 两个片内电压比较器，有 23 个触感输入
- 24 或 16 个端口 I/O（耐 5V）

具有片内上电复位、VDD 监视器、看门狗定时器和时钟振荡器的 C8051F93x-C8051F92x 器件是真正能独立工作的片上系统。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。用户软件对所有外设具有完全的控制，可以关断任何一个或所有外设以节省功耗。

片内 Silicon Labs 二线（C2）开发接口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。调试逻辑支持观察和修改存储器和寄存器，支持断点、单步、运行和停机命令。在使用 C2 进行调试时，所有的模拟和数字外设都全功能运行。两个 C2 接口引脚可以与用户功能共享，使在系统编程和调试功能不占用封装引脚。

每种器件都可在工业温度范围（-40℃到+85℃）内用 0.9V ~ 1.8V 或 1.8V ~ 3.6V 的电压工作。端口 I/O 和/RST 引脚可容许高达 5V 的输入信号。C8051F930/20 有 32 脚 QFN 和 LQFP 两种封装，C8051F931/21 为 24 脚 QFN 封装，均为无铅封装（符合 RoHS 标准）。定货信息见表 2.1。原理框图见图 1.1 ~ 图 1.4。

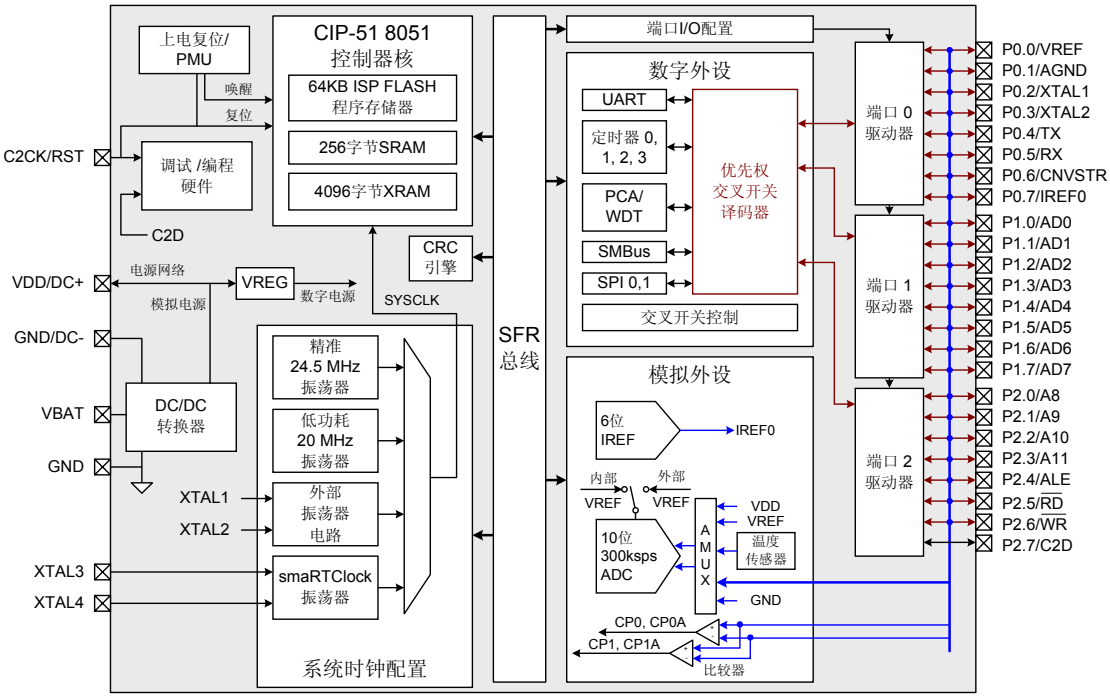


图 1.1 C8051F930 原理框图

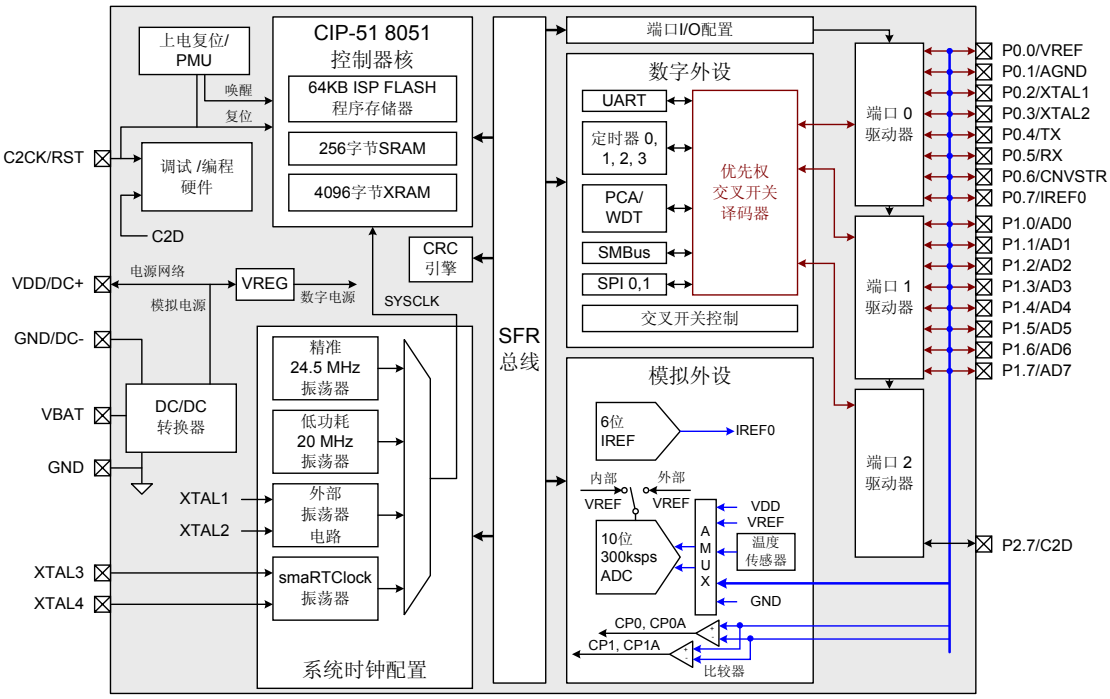


图 1.2 C8051F931 原理框图

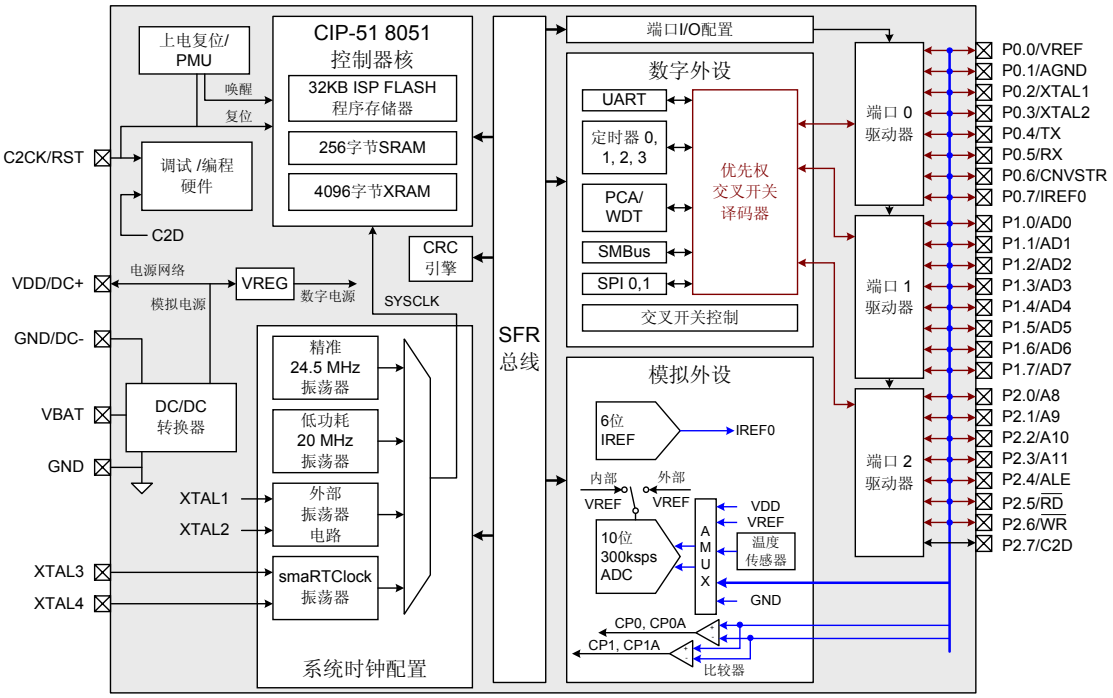


图 1.3 C8051F920 原理框图

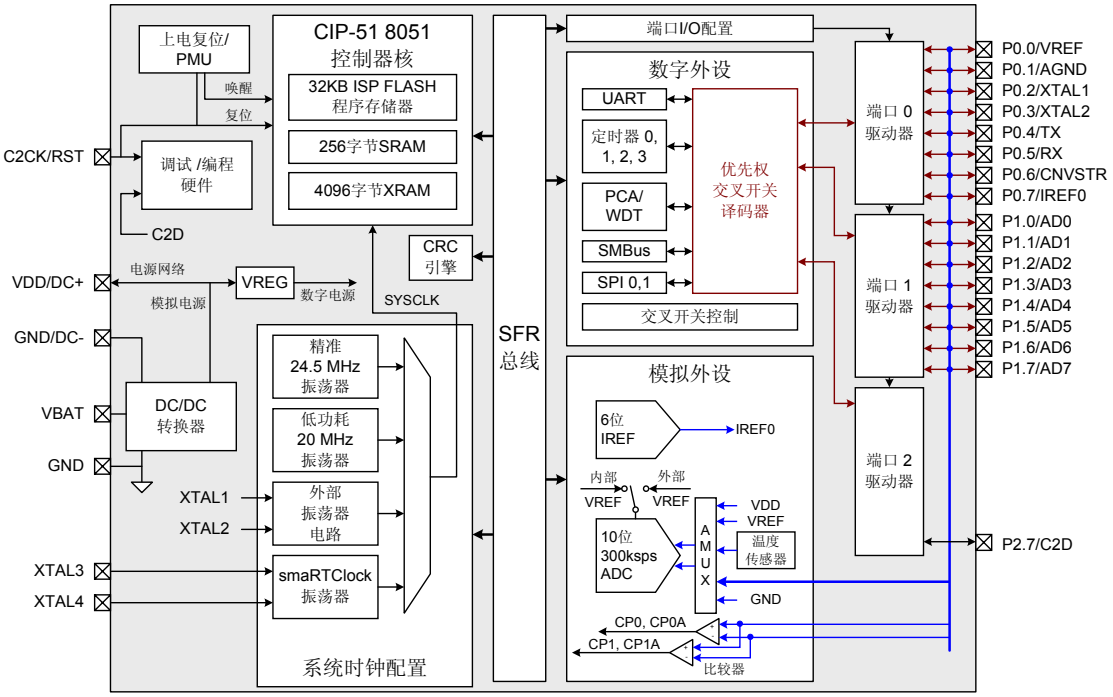


图 1.4 C8051F921 原理框图

1.1 CIP-51™ 微控制器核

1.1.1 与 8051 指令集完全兼容

C8051F93x-C8051F92x 系列器件使用 Silicon Labs 的专利 CIP-51 微控制器核。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。C8051F93x-C8051F92x 系列器件的外设是标准 8052 的所有外设的超集，包括四个 16 位计数器/定时器、一个具有扩展波特率配置的全双工 UART、两个增强型 SPI 端口、4352 字节内部 RAM、128 字节特殊功能寄存器（SFR）空间和 24/17 个 I/O 引脚。。

1.1.2 速度提升

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 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 工作在 25MHz 的时钟频率时，它的峰值速度达到 25MIPS。CIP-51 共有 111 条指令。下表列出了各种指令执行时间（指令执行时所需的系统时钟周期数）所对应的指令条数。

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

1.1.3 增加的功能

C8051F93x-C8051F92x SoC 系列 MCU 在 CIP-51 内核和外设方面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。

扩展的中断系统为 CIP-51 提供多个中断源，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，因而有更高的实际性能。这些增加的中断源在建立一个多任务实时系统时非常有用。

MCU 有 8 个复位源：上电复位电路（POR）、片内 VDD 监视器（当电源电压下降到安全电平以下时强制复位）、看门狗定时器、时钟丢失检测器、由比较器 0 提供的电压检测器、软件强制复位、外部复位引脚复位和 FLASH 非法访问保护电路复位。除了 POR、复位输入引脚及 FLASH 操作错误这三个复位源之外，其他复位源都可以被软件禁止。在一次上电复位之后的 MCU 初始化期间，WDT 可以通过软件被永久性使能。

C8051F93x-C8051F92x 器件的内部振荡器在出厂时已经被校准为 24.5MHz，其精度在整个温度和电压范围内为±2%。该内部振荡器的周期可以由用户固件调整。还有一个低精度的低功耗 20MHz 振荡器。器件内还集成了外部振荡器驱动电路，允许使用外部晶体、陶瓷谐振器、电容、RC 或 CMOS 时钟源产生系统时钟。如果需要，系统时钟源可以在内部和外部振荡器电路之间自由切换。外部振荡器在低功耗应用中也是非常有用的，它允许 MCU 从一个低速（节电）时钟源运行，在需要时周期性地切换到快速（最高达 25 MHz）的内部振荡器。

1.2 端口输入/输出

数字和模拟资源可以通过 24（C8051F930/920）或 17（C8051F931/921）个 I/O 引脚使用。端口引脚组织为三个 8 位口，P0.0 ~ P2.6 可以被定义为数字或模拟 I/O。数字 I/O 引脚可以被分配给一个内部数字资源，或用作通用 I/O（GPIO）。模拟 I/O 引脚由内部模拟资源使用。P2.7 可以被用作 GPIO 并且与 C2 接口数据信号（C2D）共享，详见“27. C2 接口”。

设计者完全控制哪些数字和模拟功能分配给哪些端口引脚，只受物理 I/O 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。有关交叉开关的详细信息见“21.3 优先权交叉开关译码器”。

当被用作数字输入或漏极开路输出时，所有端口 I/O 都耐 5V 电压。对于被配置为推挽输出的端口 I/O，电流由 VDD/DC+电源引脚供给。用于模拟功能的端口 I/O 工作电压可以达到 VDD/DC+电源电压。有关端口 I/O 的工作方式见“21.1 端口 I/O 工作方式”。 端口 I/O 的详细电气特性见电气特性一章。

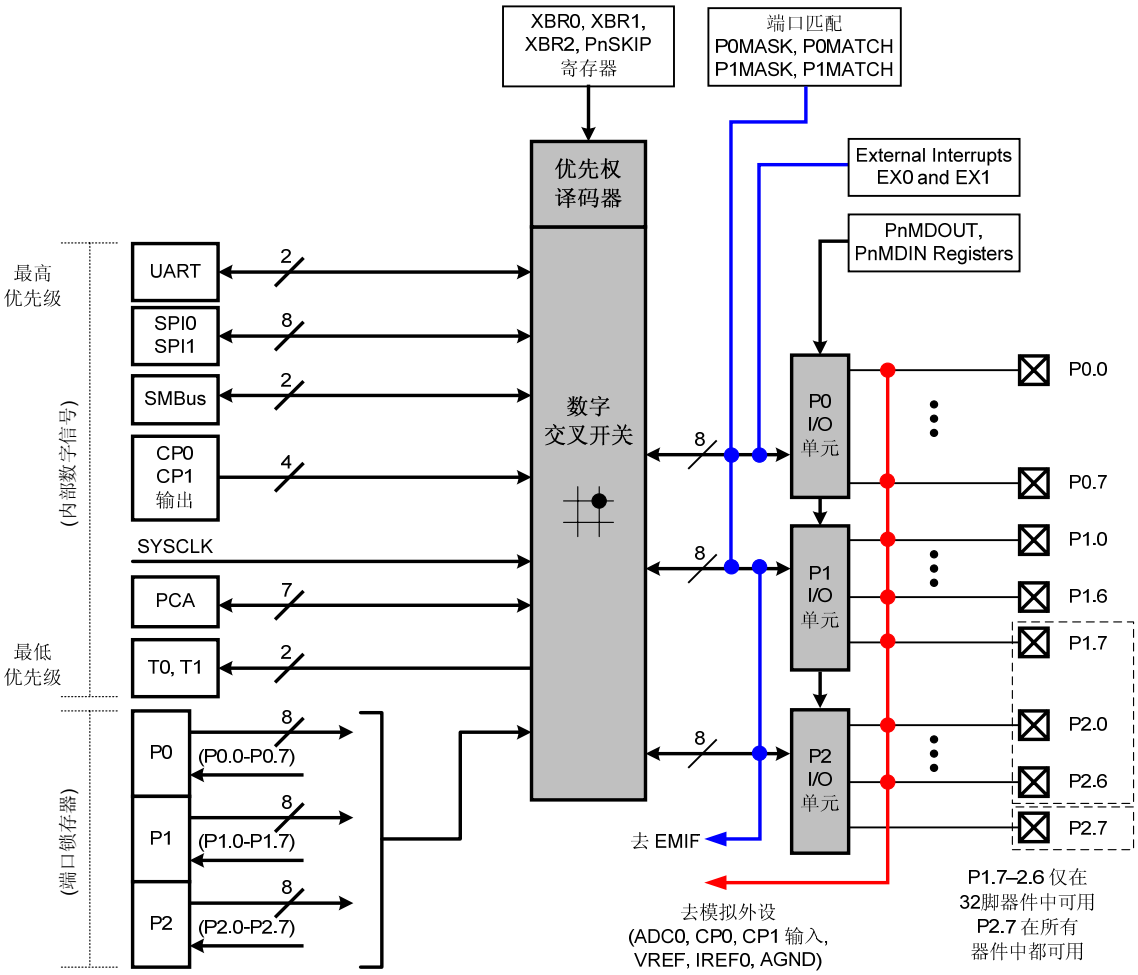


图1.5 端口I/O原理框图

1.3 串行端口

C8051F93x-C8051F92x系列MCU内部有一个SMBus/I^2C接口、一个具有增强型波特率配置的全双工UART和两个增强型SPI接口。每个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。

1.4 可编程计数器阵列

除了4个16位通用计数器/定时器之外，C8051F93x-C8051F92x系列MCU内部还有一个可编程计数器阵列（PCA）。PCA包括一个专用的16位计数器/定时器和6个16位捕捉/比较模块。PCA时钟可以在下面的7个时钟源中选择：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、实时时钟频率/8和外部振荡源频率/8。

每个捕捉/比较模块都有多种工作方式：边沿触发捕捉、软件定时器、高速输出、脉冲宽度调制器（8、9、10、11或16位）、频率输出。此外，捕捉/比较模块5还提供看门狗定时器（WDT）功能。在系统复位后，捕捉/比较模块5被配置并被使能为WDT方式。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端口I/O。

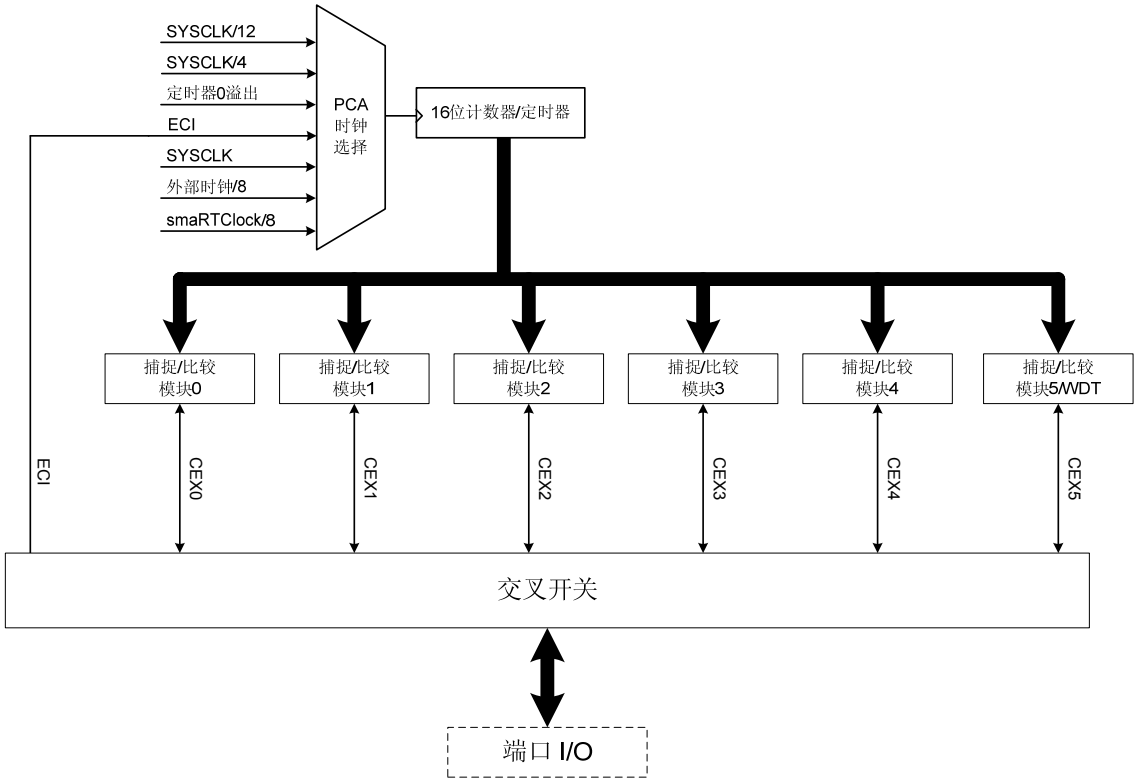


图1.6　PCA原理框图

1.5 10 位 SAR ADC

C8051F93x-C8051F92x器件内部有一个300 ksps的10位逐次逼近寄存器（SAR）ADC，该ADC集成了跟踪和保持电路及可编程窗口检测器。ADC0还具有自动低功耗突发模式，可以自动使能ADC0，采集和累加样本，然后再将ADC0置于低功耗断电状态，所有这些操作都不需要CPU干预。ADC0还有一个16位的累加器，可以自动平均ADC结果值，提供有效的11、12或13位ADC结果而不需任何额外的CPU干预。

ADC可以对任何一个GPIO引脚（P2.7除外）采样，其内部的衰减器允许它测量最高达2倍于电压基准的电压；另外，片内温度传感器、VDD/DC+、VBAT和内部数字电源电压也可以作为ADC的输入。

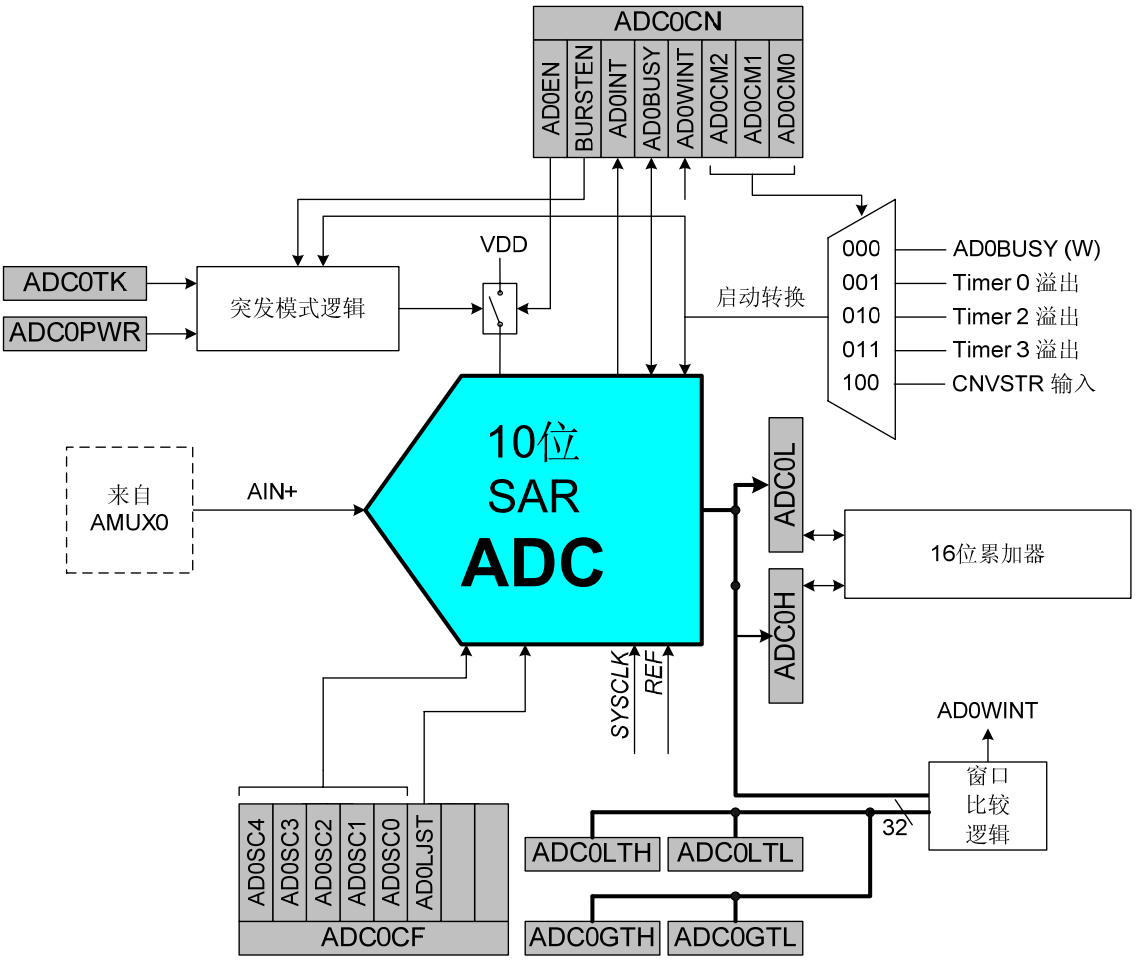


图1.7　ADC0 原理框图

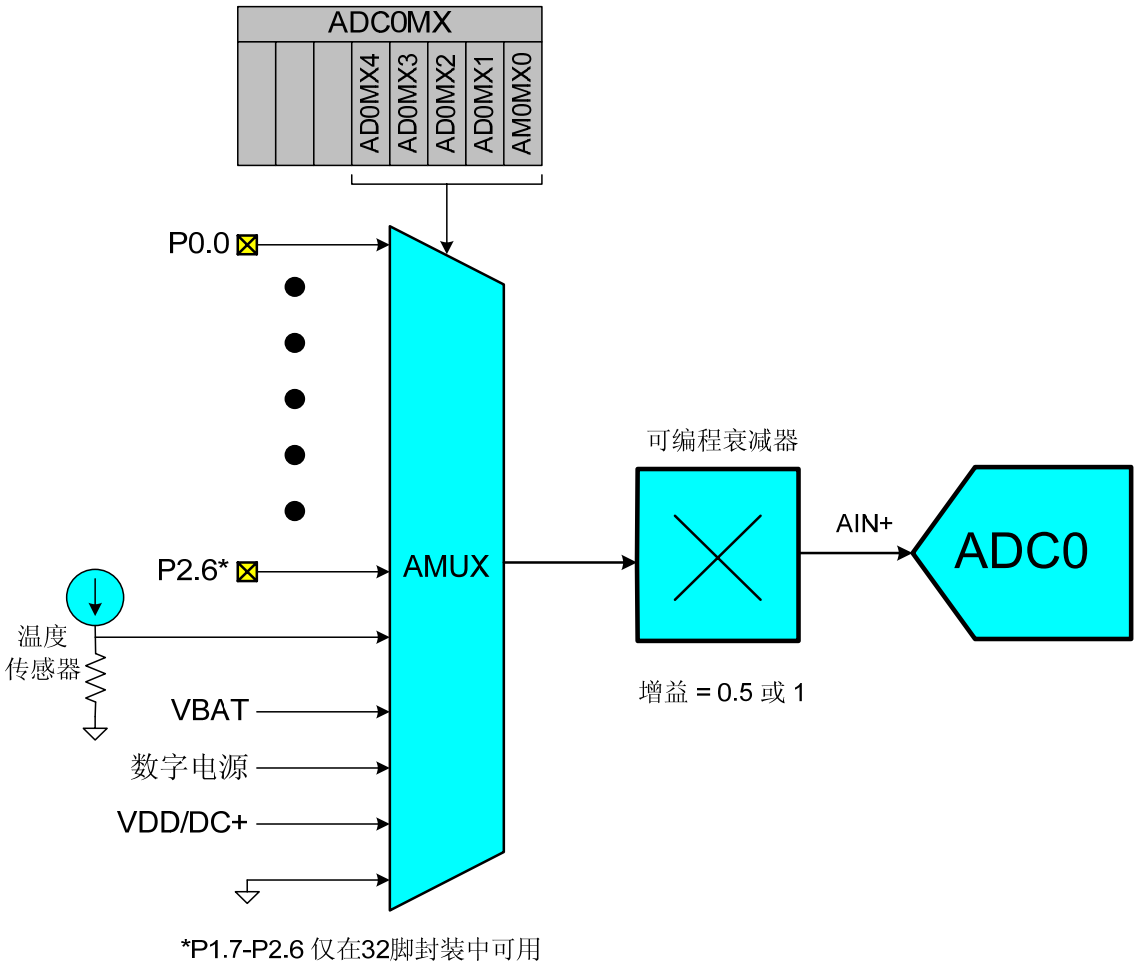


图1.8　ADC0 多路选择器原理框图

1.6 可编程电流基准（IREF0）

C8051F93x-C8051F92x内部有一个可编程电流基准（拉电流或灌电流），它具有两种输出电流模式：低功耗模式或大电流模式。低功耗模式的最大电流输出为63 μA（每级1 μA），大电流模式的最大电流输出为504 μA（每级8 μA）。

1.7 比较器

C8051F93x-C8051F92x器件内部有两个可编程的电压比较器：比较器0（CPT0，示于图1.9）和比较器1（CPT1，示于图1.10）。除了比较器0可以被用作复位源或唤醒源之外，两个比较器的操作完全相同。有关复位源和低功耗模式唤醒源的详细信息，请分别见“18. 复位源”和“14. 电源管理”。

比较器提供可编程的响应时间和和回差电压、一个模拟输入多路器和两个可选择连到端口引脚的输出：一个同步“锁存”输出（CP0和CP1），一个异步“直接”输出（CP0A和CP1A）。即使在系统时钟停止时，CP0A信号仍然可用，这就允许比较器在器件处于某些低功耗模式时工作并产生输出。

比较器输入可以连接到端口I/O引脚或其它内部信号。端口引脚还可用于直接检测容性触摸开关。

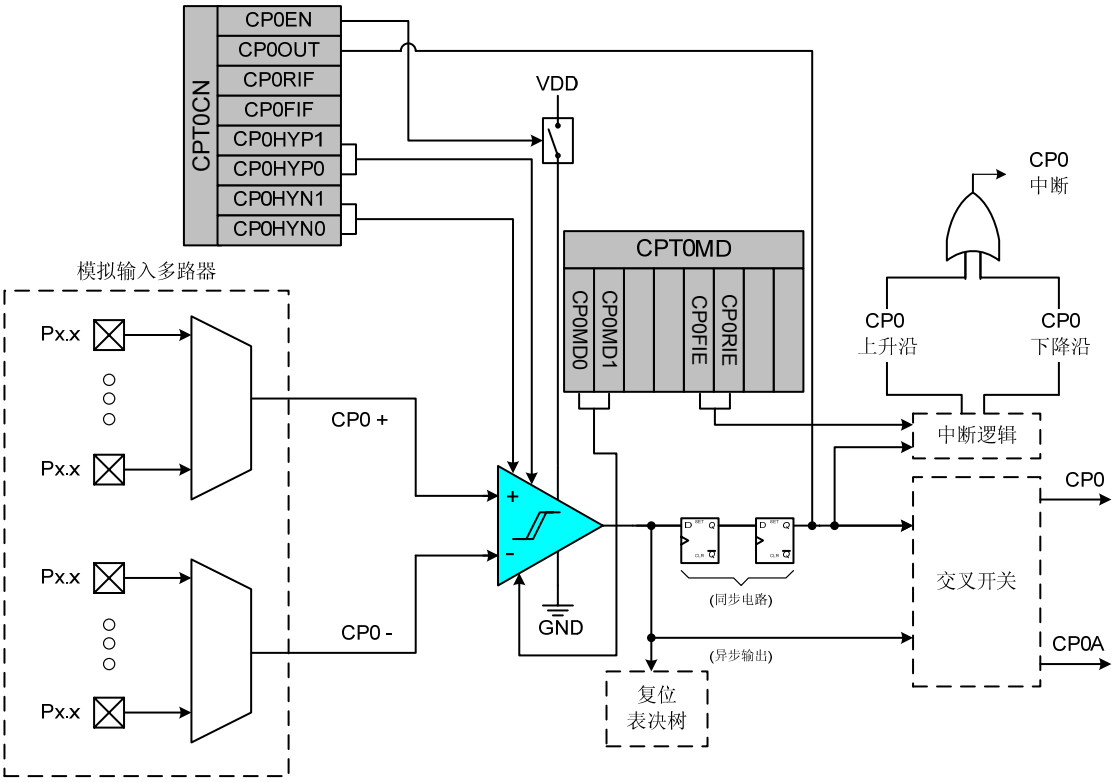


图1.9 比较器0原理框图

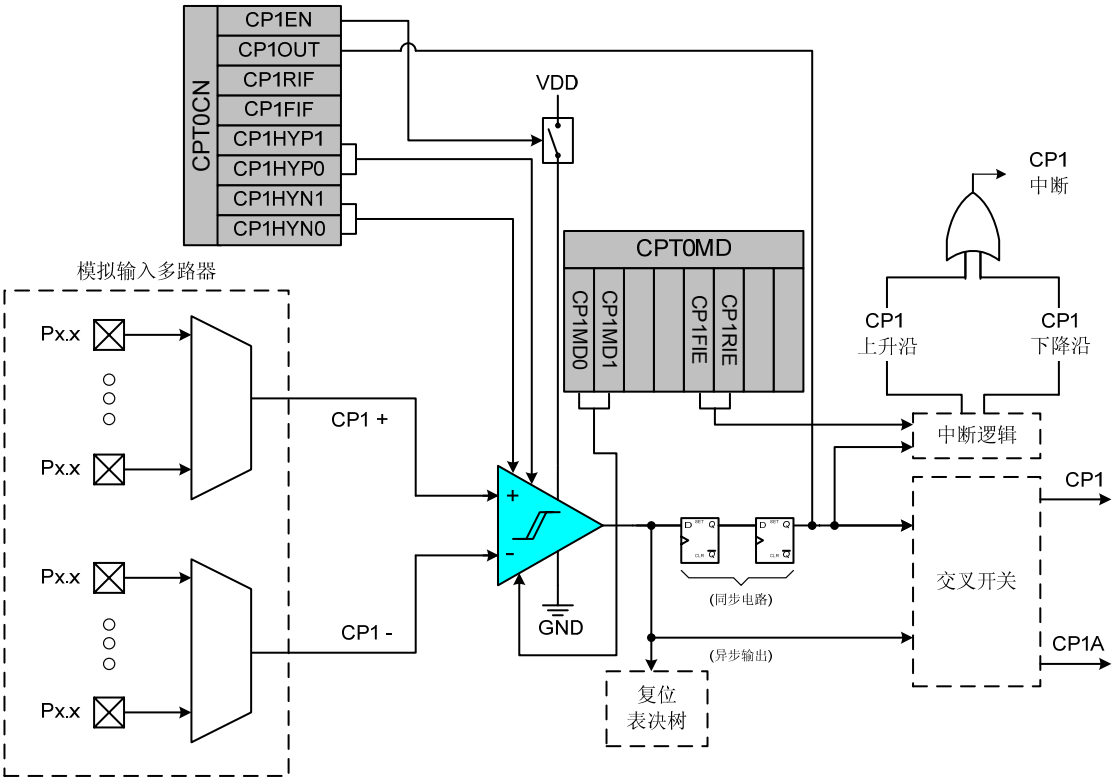


图1.10 比较器1原理框图

2. 定货信息

表 2.1　产品选择指南

型号	MIPS（峰值）	FLASH 存储器（KB）	RAM（字节）	SmaTRClock（实时时钟）	SMBus/I^2C	UART	增强型 SPI	定时器(16 位)	可编程计数器阵列	数字端口 I/O	10 位 300ksps ADC	可编程电流基准	内部电压基准	温度传感器	模拟比较器	封装
C8051F930-GM	25	64	4352	√	1	1	2	4	√	24	√	√	√	√	2	QFN-32
C8051F930-GQ	25	64	4352	√	1	1	2	4	√	24	√	√	√	√	2	LQFP-32
C8051F931-GM	25	64	4352	√	1	1	2	4	√	16	√	√	√	√	2	QFN-24
C8051F920-GM	25	32	4352	√	1	1	2	4	√	24	√	√	√	√	2	QFN-32
C8051F920-GQ	25	32	4352	√	1	1	2	4	√	24	√	√	√	√	2	LQFP-32
C8051F921-GM	25	32	4352	√	1	1	2	4	√	16	√	√	√	√	2	QFN-24

3. 引脚和封装定义

表 3.1 C8051F92x-C8051F93x 引脚定义

引脚名称	引脚号		引脚类型	说　明
	F920/30	F921/31		
VBAT	5	5	电源输入	电池电源电压。在单节电池模式必须是 0.9~1.8V，在双节电池模式必须是 1.8~3.6V。
V_{DD} DC+	3	3	电源输入 电源输出	电源电压。必须是 1.8~3.6V。该电源电压在低功耗休眠模式不需要。 DC-DC 转换器的正输出。在单节电池模式，DC+与 DC-之间需要一个 1μF 的陶瓷电容。当工作在单节电池模式时，该引脚可以为外部器件供电。
DC- GND	1	1	电源输入 地	DC-DC 转换器的返回电流路径。在单节电池模式，该引脚一定不能连接到地。 在双节电池模式，该引脚必须直接连接到地。
GND	2	2	地	地。
DCEN	4	4	电源输入 地	DC-DC 使能引脚。在单节电池模式，该引脚必须通过一个 0.68μH 的电感连接到 VBAT。 在双节电池模式，该引脚必须直接连接到地。
/RST C2CK	6	6	数字 I/O 数字 I/O	器件复位。内部上电复位或 VDD 监视器的漏极开路输出。一个外部源可以通过将该引脚驱动为低电平（至少 15μs）来启动一次系统复位。建议在该引脚与 V_{DD} 之间接 1KΩ~5KΩ 的上拉电阻。 C2 调试接口的时钟信号。
P2.7 C2D	7	7	数字 I/O 数字 I/O	端口 P2.7。该引脚只能用作 GPIO。交叉开关不能将信号连到该引脚，该引脚也不能被配置为模拟输入。 C2 调试接口的双向数据信号。
XTAL3	10	9	模拟输入	smaRTClock 振荡器晶体输入。
XTAL4	9	8	模拟输出	smaRTClock 振荡器晶体输出。
P0.0/ VREF	32	24	数字 I/O 或模拟输入 模拟输入 模拟输出	端口 P0.0 外部 VREF 输入 内部 VREF 输出。建议外接去耦电容。
P0.1 AGND	31	23	数字 I/O 或模拟输入 地	端口 P0.1 可选的模拟地。

表 3.1 C8051F92x-C8051F93x 引脚定义（续）

引脚名称	引脚号		引脚类型	说　明
	F920/30	F921/31		
P0.2 XTAL1	30	22	数字 I/O 或模拟输入 模拟输入	端口 P0.2 外部时钟输入。对于晶体或陶瓷谐振器，该引脚是外部振荡器电路的反馈输入。
P0.3 XTAL2	29	21	数字 I/O 或模拟输入 模拟输出 数字输入 模拟输入	端口 P0.3 外部时钟输出。该引脚是外部晶体或陶瓷谐振器的激励驱动器。 外部时钟输入。对于外部 CMOS 时钟方式、该引脚是外部时钟输入。 外部时钟输入。该引脚是电容或 RC 振荡器配置的外部时钟输入。
P0.4 TX	28	20	数字 I/O 或模拟输入 数字输出	端口 P0.4 UART TX 引脚
P0.5 RX	27	19	数字 I/O 或模拟输入 数字输入	端口 P0.5 UART RX 引脚
P0.6 CNVSTR	26	18	数字 I/O 或模拟输入 数字输入	端口 P0.6 ADC0 的外部转换启动输入。
P0.7 IREF0	25	17	数字 I/O 或模拟输入 模拟输出	端口 P0.7 IREF0 输出。
P1.0 AD0*	24	16	数字 I/O 或模拟输入 数字 I/O	端口 P1.0 可用作 SPI1 的 SCK。 地址/数据 0。
P1.1 AD1*	23	15	数字 I/O 或模拟输入 数字 I/O	端口 P1.1 可用作 SPI1 的 MISO。 地址/数据 1。
P1.2 AD2*	22	14	数字 I/O 或模拟输入 数字 I/O	端口 P1.2 可用作 SPI1 的 MOSI。 地址/数据 2。
P1.3 AD3*	21	13	数字 I/O 或模拟输入 数字 I/O	端口 P1.3 可用做 SPI1 的 NSS。 地址/数据 3。
P1.4 AD4*	20	12	数字 I/O 或模拟输入 数字 I/O	端口 P1.4 地址/数据 4。
*注：仅在 C8051F920/30 中可用。				

表 3.1 C8051F92x-C8051F93x 引脚定义（续）

引脚名称	引脚号		引脚类型	说　明
	F920/30	F921/31		
P1.5 AD5*	19	11	数字 I/O 或模拟输入 数字 I/O	端口 P1.5 地址/数据 5。
P1.6 AD6*	18	10	数字 I/O 或模拟输入 数字 I/O	端口 P1.6 地址/数据 6。
P1.7* AD7*	17		数字 I/O 或模拟输入 数字 I/O	端口 P1.7 地址/数据 7。
P2.0* A8*	16		数字 I/O 或模拟输入 数字输出	端口 P2.1 地址 8
P2.1* A9*	15		数字 I/O 或模拟输入 数字输出	端口 P2.1 地址 9
P2.2* A10*	14		数字 I/O 或模拟输入 数字输出	端口 P2.2 地址 10
P2.3* A11*	13		数字 I/O 或模拟输入 数字输出	端口 P2.3 地址 11
P2.4* ALE*	12		数字 I/O 或模拟输入 数字输出	端口 P2.4 地址锁存使能
P2.5* /RD*	11		数字 I/O 或模拟输入 数字输出	端口 P2.5 读选通
P2.6* /WR*	8		数字 I/O 或模拟输入 数字输出	端口 P2.6 写选通
*注：仅在 C8051F920/30 中可用。				

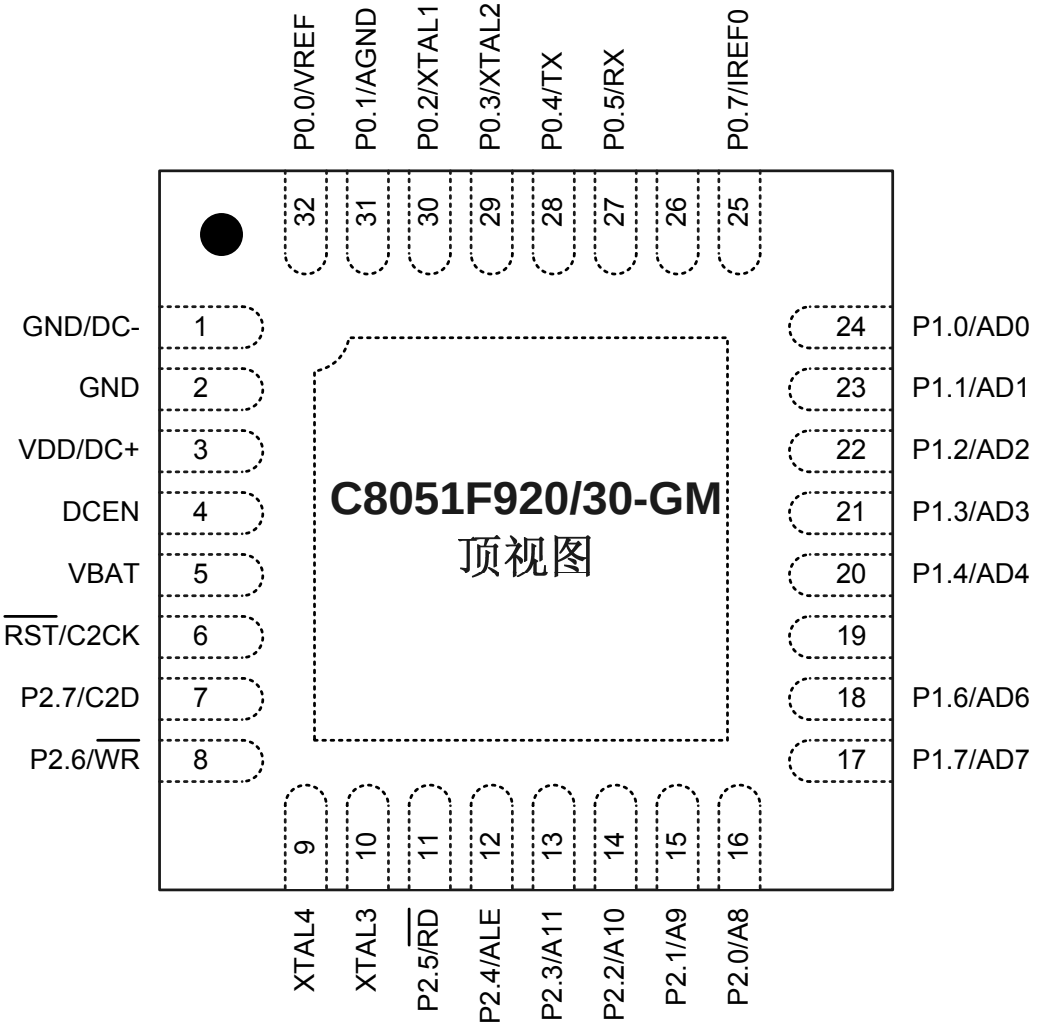


图 3.1　QFN-32 引脚图（顶视图）

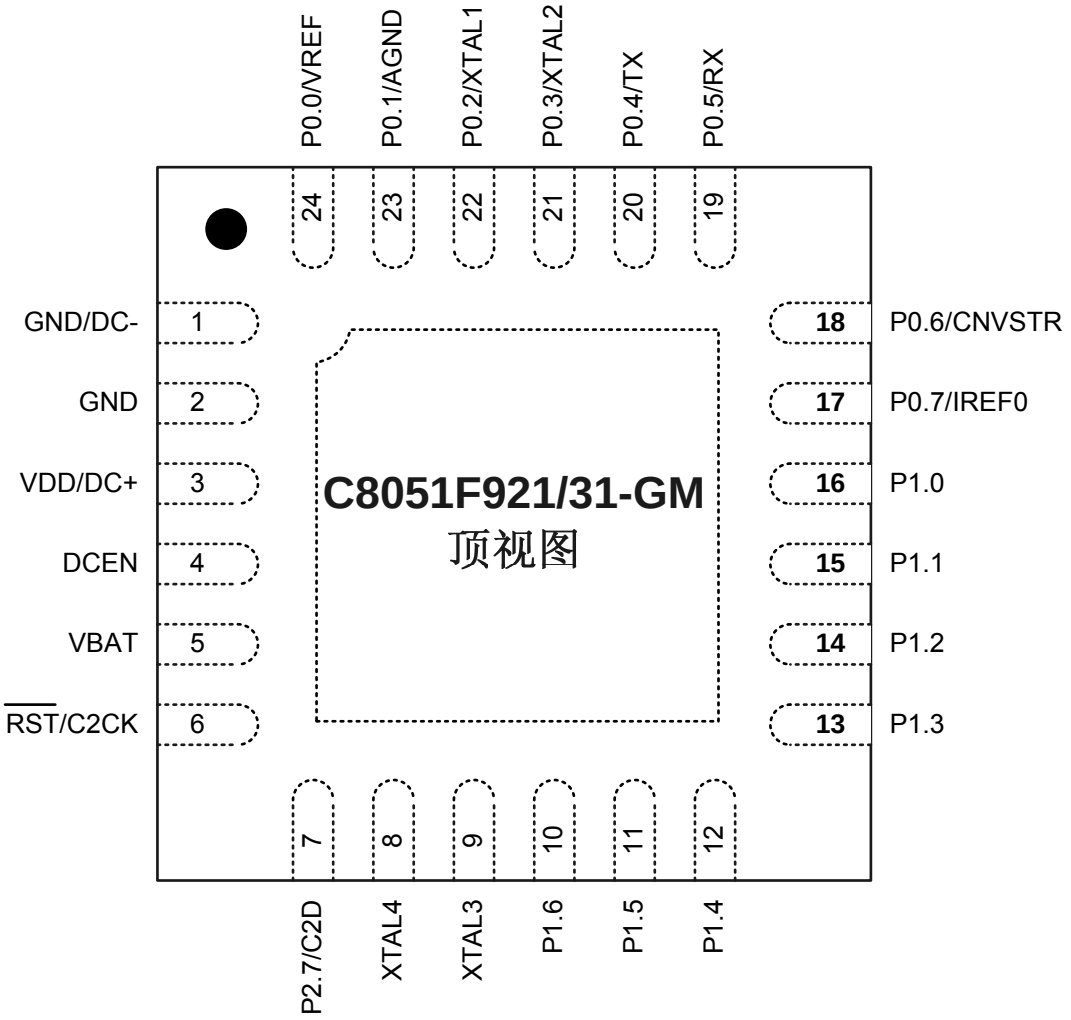


图 3.2　QFN-24 引脚图（顶视图）

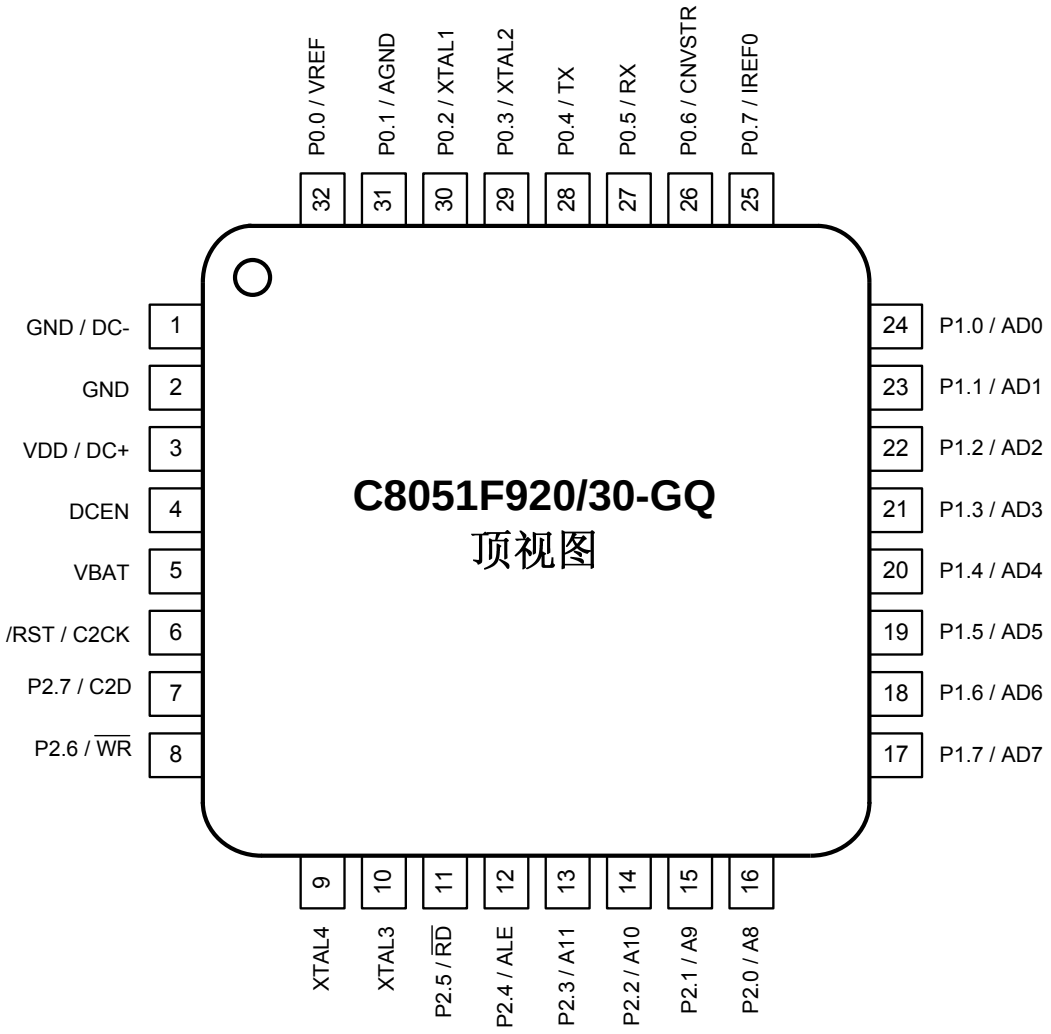


图 3.3　LQFP-32 引脚图

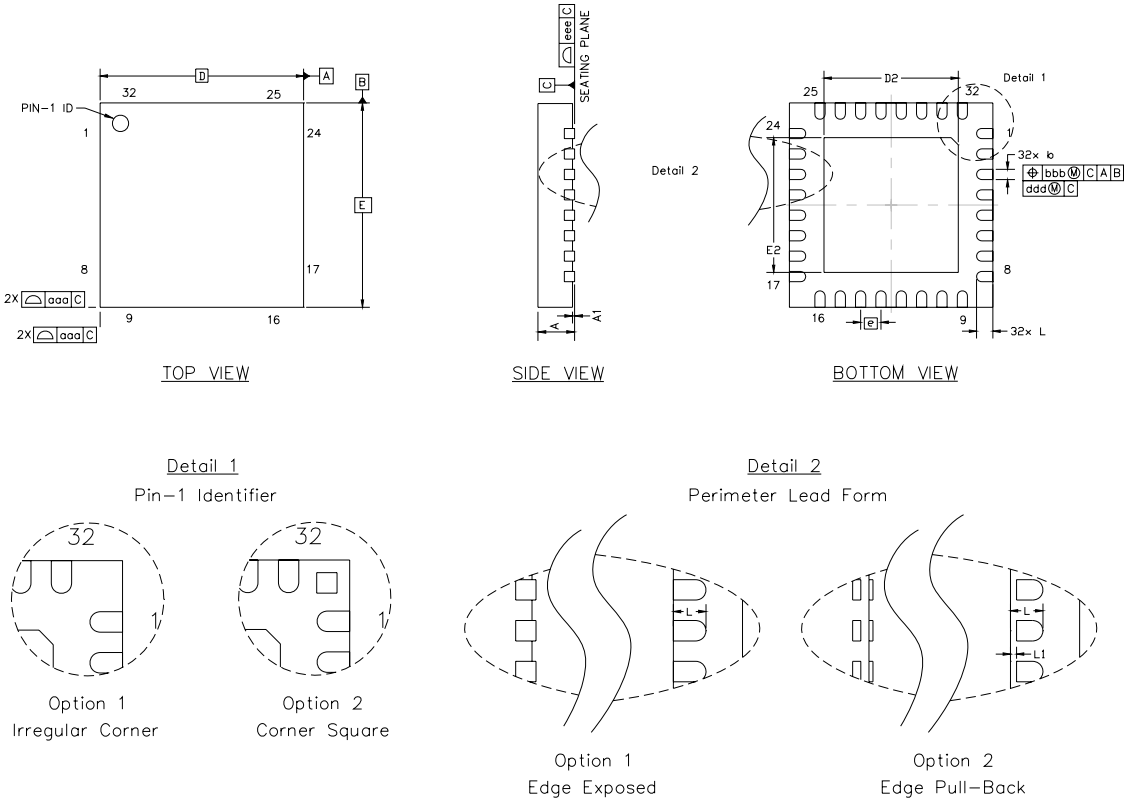


图 3.4　QFN-32 封装图

表 3.2 QFN-32 封装尺寸

尺寸	最小值	典型值	最大值
A	0.80	0.90	1.00
A1	0.00	0.02	0.05
B	0.18	0.25	0.30
D	5.0 BSC		
D2	3.20	3.30	3.40
e	0.50 BSC		
E	5.0 BSC		

尺寸	最小值	典型值	最大值
E2	3.20	3.30	3.40
L	0.30	0.40	0.50
L1	0.00	—	0.15
aaa	—	—	0.15
bbb	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08

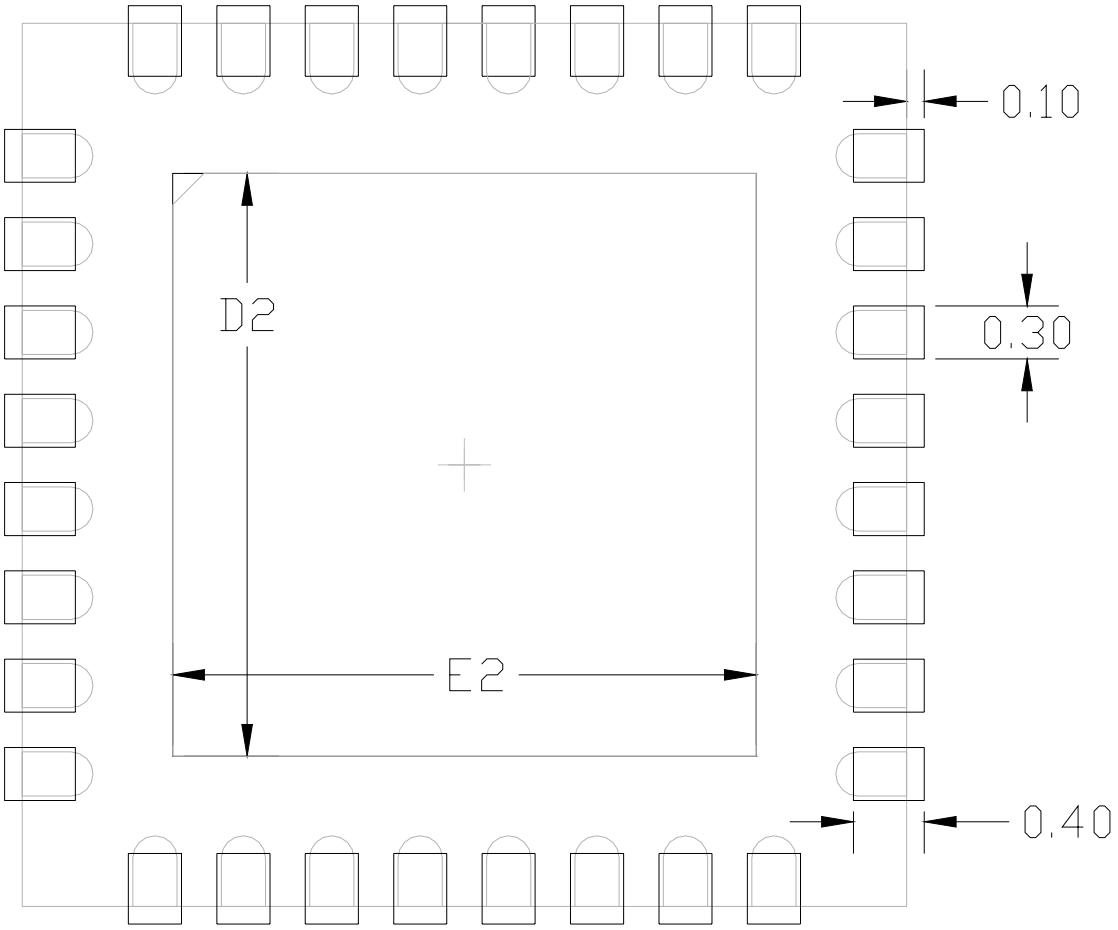


图 3.5　典型的 QFN-32 焊盘图

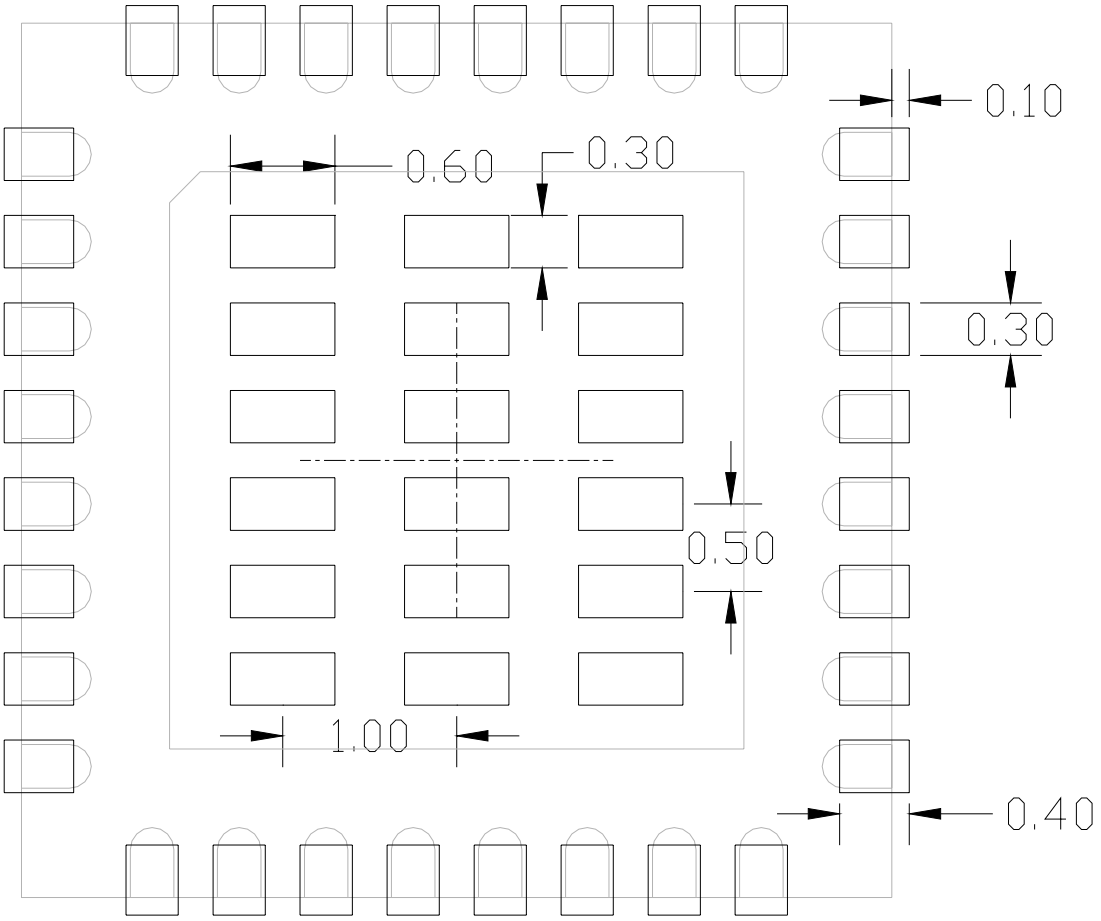


图 3.6 QFN-32 锡膏层建议图

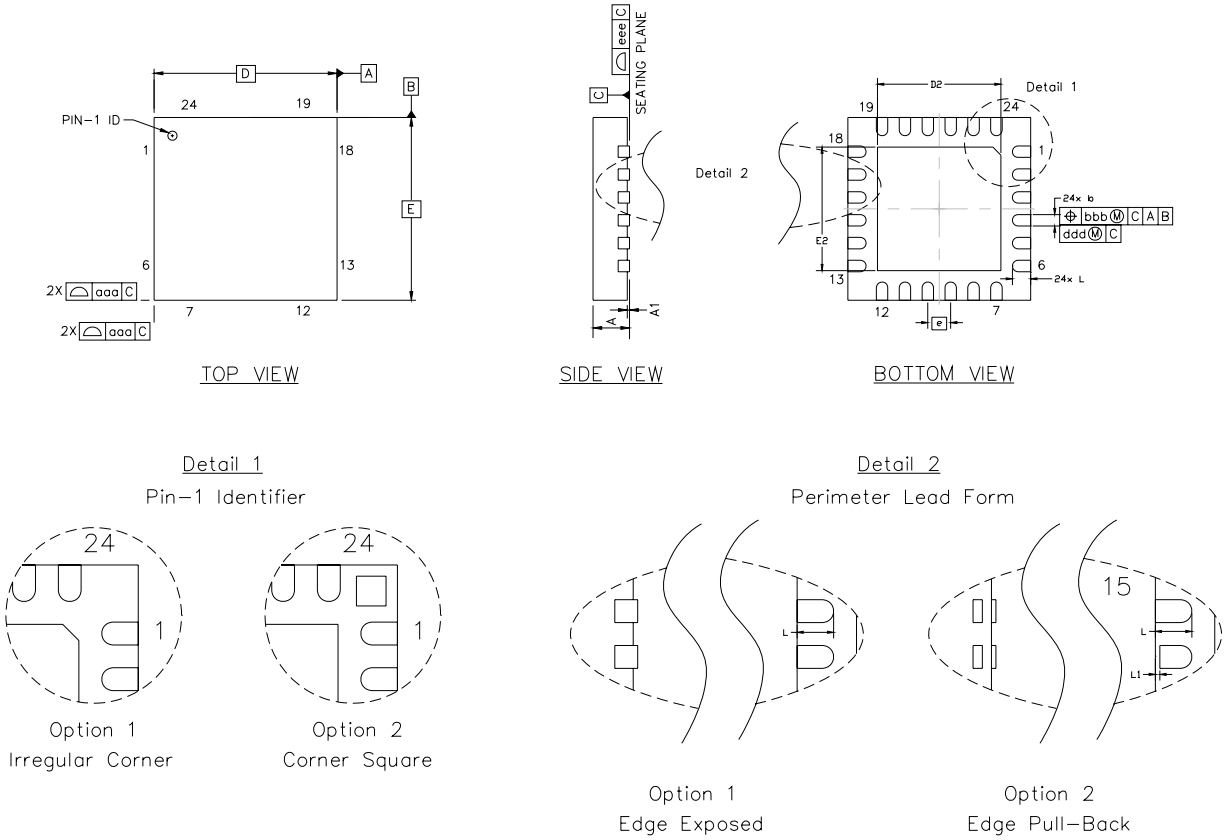


图 3.7　QFN-24 封装图

表 3.3 QFN-24 封装尺寸

尺寸	最小值	典型值	最大值
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
D	4.0 BSC		
D2	2.60	2.70	2.80
e	0.50 BSC		
E	4.0 BSC		

尺寸	最小值	典型值	最大值
E2	2.60	2.70	2.80
L	0.30	0.40	0.50
L1	0.00	—	0.15
aaa	—	—	0.15
bbb	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08

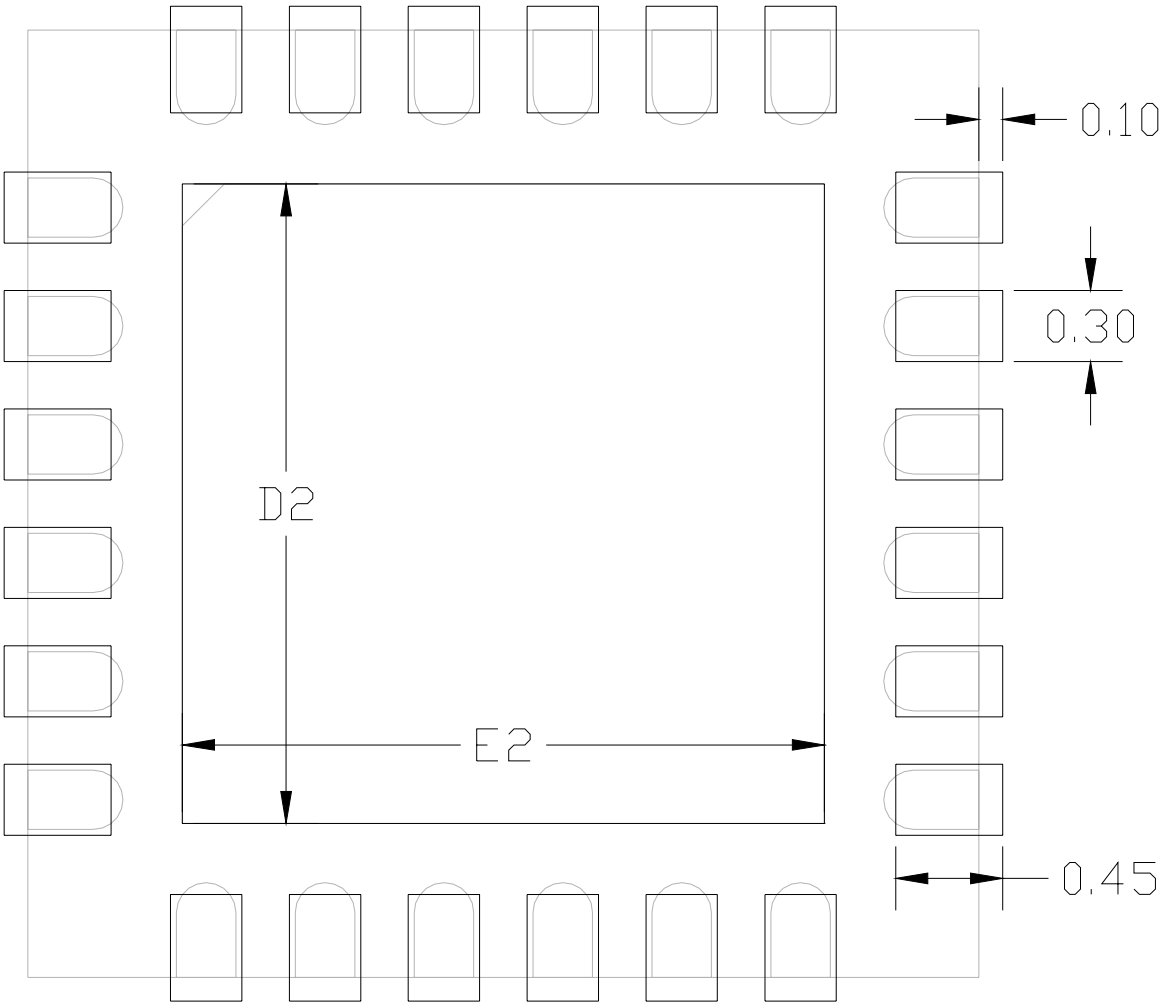


图 3.8　典型的 QFN-24 焊盘图

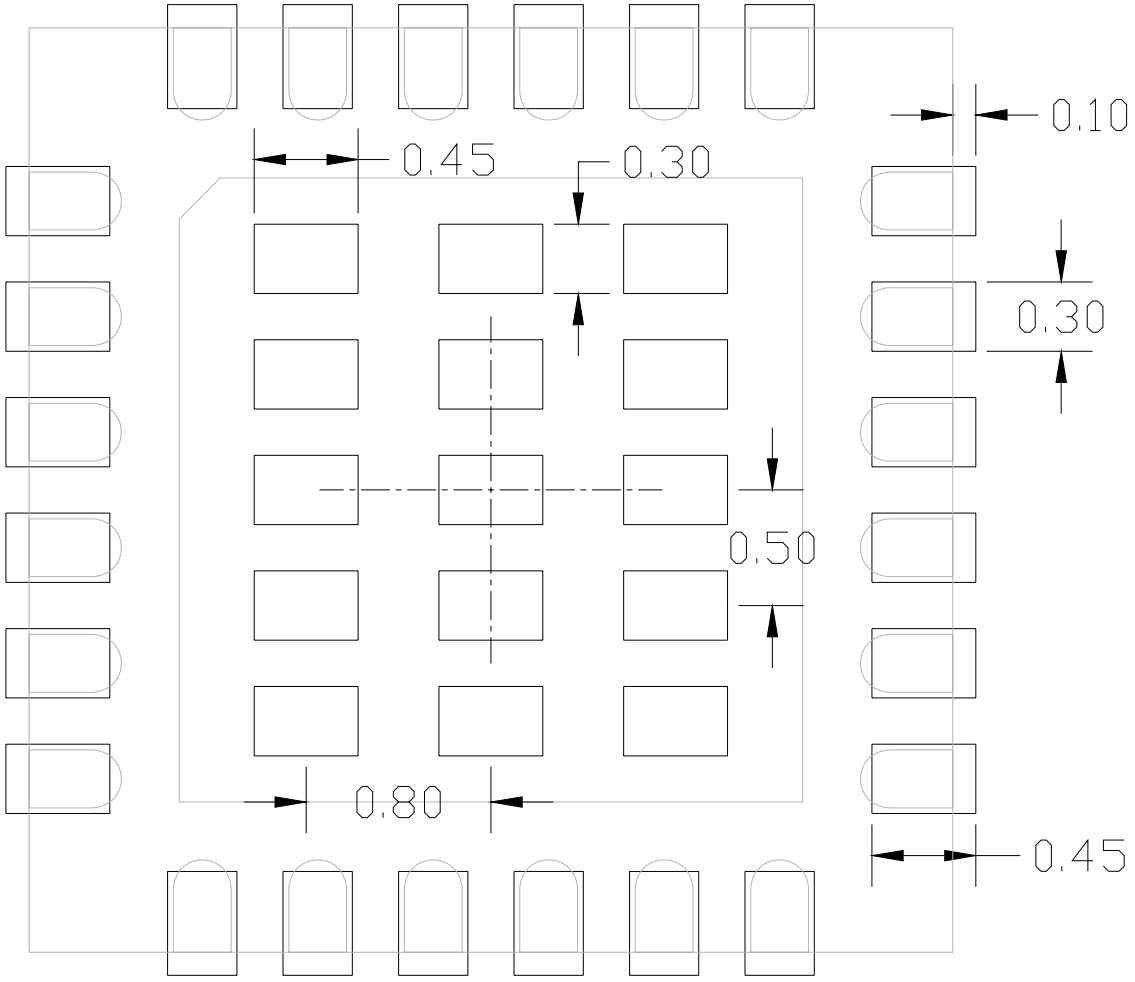
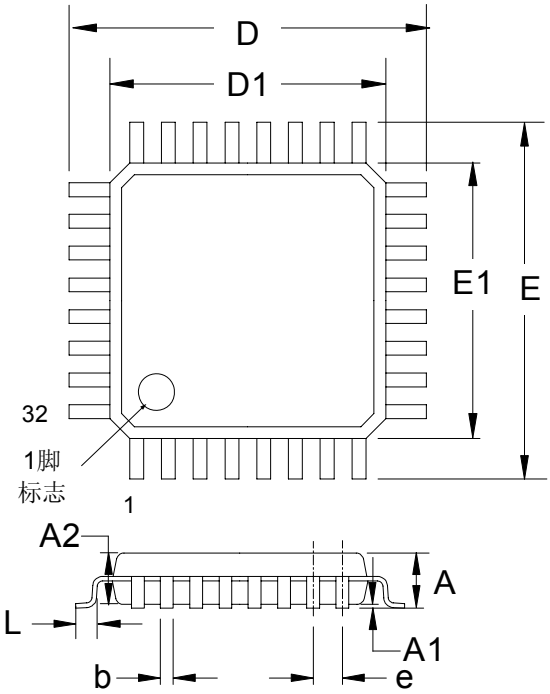


图 3.9　QFN-24 锡膏层建议图



LQFP-32
封装尺寸

	MM		
	MIN	TYP	MAX
A	-	-	1.60
A1	0.05	-	0.15
A2	1.35	1.40	1.45
b	0.30	0.37	0.45
D	-	9.00	-
D1	-	7.00	-
e	-	0.80	-
E	-	9.00	-
E1	-	7.00	-
L	0.45	0.60	0.75

图 3.10　LQFP-32 封装图

4. 电气特性

4.1 极限参数

表 4.1 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	℃
储存温度		-65	—	150	℃
任何端口I/O引脚或/RST相对GND的电压	VDD>2.2V VDD<2.2V	-0.3 -0.3	— —	5.8 VDD+3.6	V
VBAT引脚相对GND的电压	单节模式 双节模式	-0.3 -0.3	— —	2.0 4.0	V
VDD/DC+引脚相对GND的电压	稳压器正常模式 稳压器旁路模式	-0.3 -0.3	— —	4.0 2.0	V
通过VBAT、DCEN、VDD/DC+或GND的最大总电流		—	—	500	mA
/RST或任何端口引脚的最大输出灌电流		—	—	100	mA
*注：超过这些列出的“极限参数”可能导致器件永久性损坏。长时间在最大允许值或超过最大允许值的条件下工作可能影响器件的可靠性。					

4.2 总体电气特性

表 4.2 总体电气特性

-40℃到+85℃，25MHz 系统时钟（除非特别说明）。

参　数	条　件	最小值	典型值	最大值	单 位
电池电压（VBAT）	单节模式 双节模式	0.9 1.8	1.5 3.0	1.8 3.6	V V
电源电压（VDD/DC+）[1]	稳压器正常模式 稳压器旁路模式	1.8 1.7	3.0 1.8	3.6 1.9	V V
RAM数据保持电源电压	VDD（非休眠模式） VBAT（休眠模式）	— —	TBD TBD	— TBD	V
SYSCLK（系统时钟）[2]		0	—	25	MHz
T_{SYSH}（SYSCLK高电平时间）		18	—	—	ns
T_{SYSL}（SYSCLK低电平时间）		18	—	—	ns
额定工作温度范围		-40	—	+85	℃
数字电源电流 — CPU 活动（正常方式，从 FLASH 中取指令）					
I_{DD}[3,4,5]	V_{DD} = 1.8 ~ 3.6V, F = 24.5 MHz V_{DD} = 1.8 ~ 3.6V, F = 1.25 MHz V_{DD} = 1.8 ~ 3.6V, F = 32.768 kHz	— — —	4.1 470 90	TBD — —	mA µA µA
I_{DD} 频率敏感度[3,5]	V_{DD} = 1.8 ~ 3.6V, T = 25℃ F < 10 MHz(单稳使能) V_{DD} = 1.8 ~ 3.6V, T = 25℃ F > 10 MHz(单稳旁路)	— —	226 120	— —	µA/MHz µA/MHz
数字电源电流 — CPU 不活动（空闲方式，不从 FLASH 中取指令）					
I_{DD}[3,4,6]	V_{DD} = 1.8 ~ 3.6V, F = 24.5 MHz V_{DD} = 1.8 ~ 3.6V, F = 1.25 MHz V_{DD} = 1.8 ~ 3.6V, F = 32.768 kHz	— — —	2.5 400 84	TBD — —	mA µA µA
I_{DD} 频率敏感度[3,6]	V_{DD} = 1.8 ~ 3.6V, T = 25℃	—	95	—	µA/MHz
数字电源电流（挂起模式）	V_{DD} = 1.8 ~ 3.6V	—	77	—	µA
数字电源电流（休眠模式，smaRTClock运行）	1.8V, T = 25℃（包括 smaRTClock 振荡器、VBAT 电源监视器和稳压器偏置电流）	—	0.6	—	µA
数字电源电流（休眠模式）	1.8V, T = 25℃ 1.8V, T = 85℃ （包括 VBAT 电源监视器）	— —	50 1.2	— —	nA µA

注：

1. 当 VDD/DC+低于 1.8V 时，模拟性能下降。
2. 为能使用调试功能，SYSCLK 至少应为 32kHz。
3. 基于器件特性数据，未经产品测试。
4. 包括振荡器和稳压器供电电流。
5. 在频率≤10 MHz 时，可以通过简单地将感兴趣的频率乘以该范围的频率敏感度来估算 I_{DD}。当使用这些数字估算频率>10 MHz 时的 I_{DD} 时，估算值应为 25 MHz 时的电流减去由频率敏感度计算出的电流差。例如：V_{DD} = 3.0V，F = 20MHz，则 I_{DD} = 4.1mA – (25MHz – 20MHz)*0.12mA/MHz = 3.5mA。
6. 当估算频率空闲方式 I_{DD} 时，估算值应为 25 MHz 时的电流减去由频率敏感度计算出的电流差。例如：V_{DD} = 3.0V，F = 5MHz，则空闲方式 I_{DD} = 2.5mA – (25MHz – 5MHz)*0.095mA/MHz = 0.6mA。

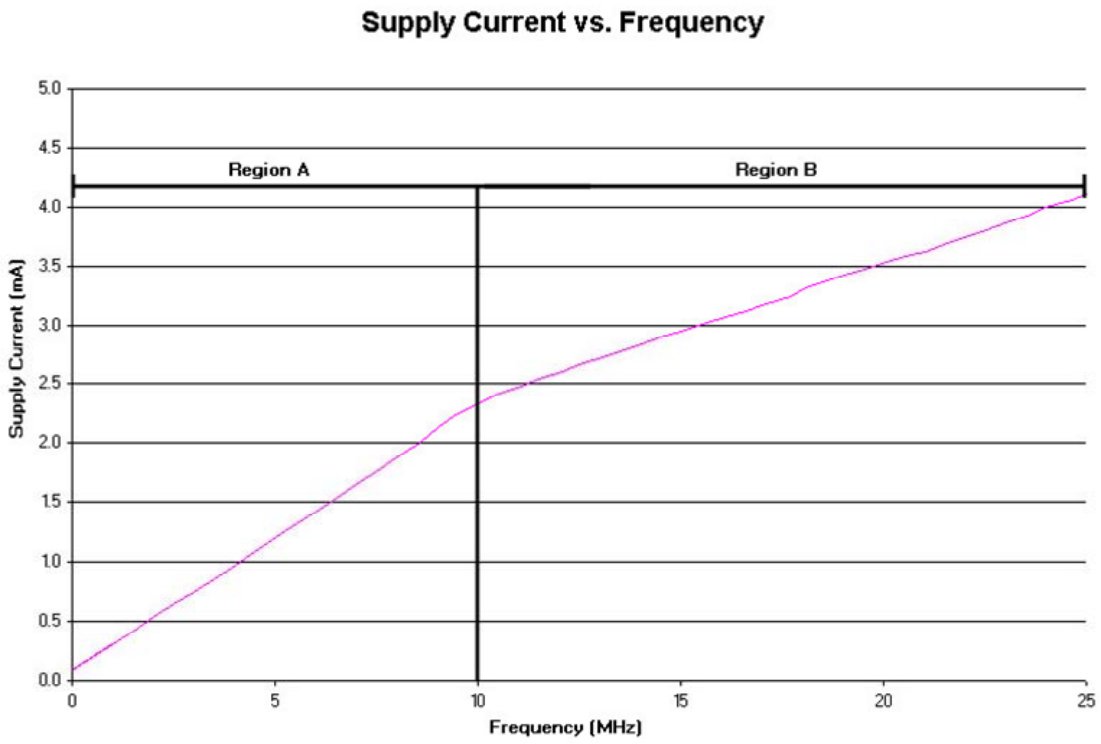


图 **4.1**　活动模式电流

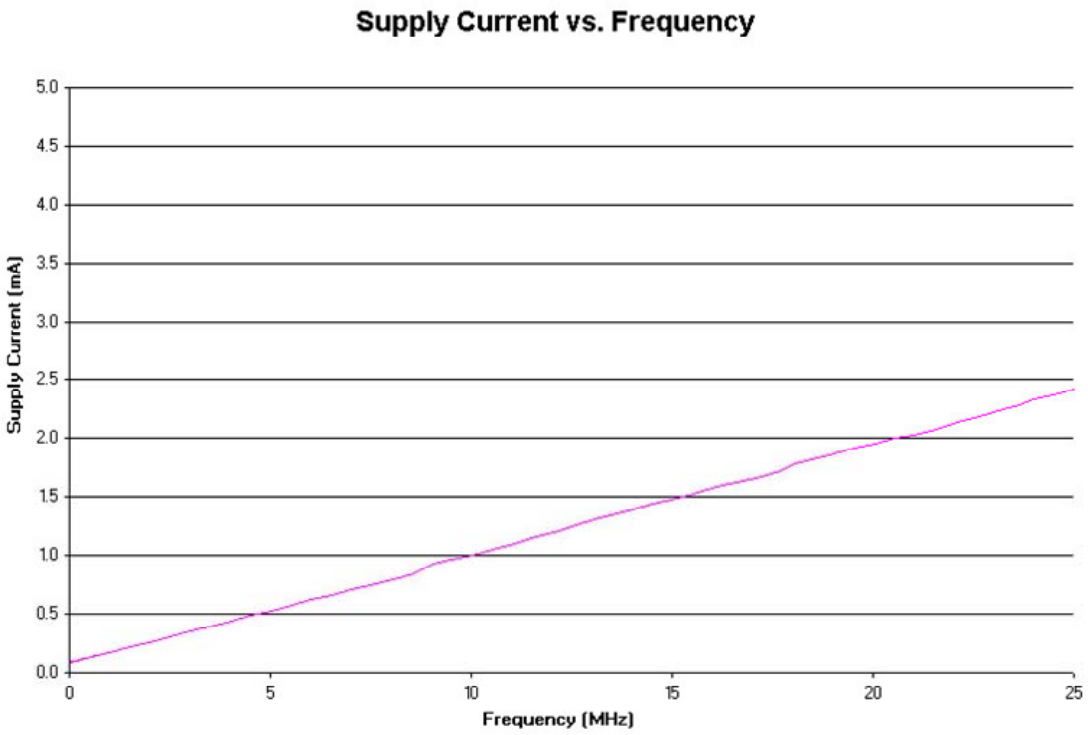


图 **4.2**　空闲模式电流

表 4.3　端口 I/O 直流电气特性

VDD = 1.8V – 3.6V, -40℃到+85℃（除非特别说明）。

参　数	条　件	最小值	典型值	最大值	单 位
输出高电压	高驱动强度，PnDRV.n = 1 I_{OH} = -3mA，端口 I/O 为推挽方式 I_{OH} = -10μA，端口 I/O 为推挽方式 I_{OH} = -10mA，端口 I/O 为推挽方式 低驱动强度，PnDRV.n = 0 I_{OH} = -1mA，端口 I/O 为推挽方式 I_{OH} = -10μA，端口 I/O 为推挽方式 I_{OH} = -3mA，端口 I/O 为推挽方式	 VDD-0.7 VDD-0.1 — VDD-0.7 VDD-0.1 —	 — — VDD-0.8 — — VDD-0.8	 — — — — — —	V
输出低电压	高驱动强度，PnDRV.n = 1 I_{OL} = 8.5mA I_{OL} = 10μA I_{OL} = 25mA 低驱动强度，PnDRV.n = 0 I_{OL} = 1.4mA I_{OL} = 10μA I_{OL} = 4mA	 — — — — — —	 — — 1.0 — — 1.0	 0.6 0.1 — 0.6 0.1 —	V
输入高电压		TBD	—	—	V
输入低电压		—	—	TBD	V
输入漏电流	弱上拉禁止 弱上拉使能，V_{IN} = 0 V	— —	— TBD	±1 TBD	μA

表 4.4 复位电气特性

VDD = 1.8V ~ 3.6V, -40℃到+85℃（除非特别说明）。

参　数	条　件	最小值	典型值	最大值	单位
/RST 输出低电平	I_{OL}= 8.5 mA	—	—	0.6	V
/RST 输入高电平		TBD	—	—	V
/RST 输入低电平		—	—	TBD	V
/RST 输入上拉电流	/RST=0.0V	—	TBD	TBD	µA
VDD/DC+监视器门限 (V_{RST})	早期告警 复位触发（除休眠外的所有电源模式）	TBD TBD	1.85 1.75	TBD TBD	V
VBAT 上电时的上升时间	VBAT 从 0V 上升到 0.9V	—	—	3	ms
VBAT 监视器门限 (V_{POR})	初次上电 欠压条件	TBD TBD	0.75 0.82	TBD TBD	V
时钟丢失检测器超时	从最后一个系统时钟上升沿到产生复位	TBD	500	TBD	µs
复位时间延迟	从退出复位到开始执行位于 0x0000 地址的代码之间的延时	—	—	TBD	µs
产生系统复位的最小 /RST 低电平时间		15	—	—	µs
V_{DD} 监视器启动时间		—	300	—	ns
V_{DD} 监视器供电电流		—	7	TBD	µA

表 4.5　电源管理电气特性

VDD = 1.8V ~ 3.6V, -40℃到+85℃（除非特别说明）。

参　数	条　　件	最小值	典型值	最大值	单 位
空闲模式唤醒时间		2	—	3	SYSCLK
挂起模式唤醒时间	低功耗振荡器 精准振荡器	— —	400 1.3	— —	ns µs
休眠模式唤醒时间	双节电池模式 单节电池模式	— —	2 10	— —	µs µs

表 4.6　FLASH 电气特性

VDD = 1.8V ~ 3.6V, -40℃到+85℃（除非特别说明）。

参　数	条　　件	最小值	典型值	最大值	单 位
FLASH 大小	C8051F930/1 C8051F920/1	65536* 32768	— —	— —	字节 字节
临时块大小		1024	—	1024	字节
擦写寿命		TBD	TBD	—	擦/写
擦除时间		1.8	2.0	2.2	ms
写入时间		3.6	4.0	4.4	µs
*注：位于地址 0xFC00~FFFF 的 1024 字节保留。					

表 4.7 内部精准振荡器电气特性

VDD = 1.8V ~3.6V；-40℃到+85℃（除非特别说明）。

参　数	条　件	最小值	典型值	最大值	单位
振荡器频率	-40℃到+85℃ VDD = 1.8V ~ 3.6V	24	24.5	25	MHz
振荡器供电电流（自 VDD）	25℃	—	300	TBD	µA
电源敏感度	恒温	—	—	TBD	MHz
温度敏感度	恒压	—	—	TBD	MHz

表 4.8 内部低功耗振荡器电气特性

VDD = 1.8V ~3.6V；-40℃到+85℃（除非特别说明）

参　数	条　件	最小值	典型值	最大值	单位
振荡器频率	-40℃到+85℃ VDD = 1.8V ~ 3.6V	18	20	22	MHz
振荡器电源电流（自 VDD）	25℃	—	100	TBD	µA
电源敏感度	恒温	—	—	TBD	MHz
温度敏感度	恒压	—	—	TBD	MHz

表 4.9　ADC0 电气特性

VDD=1.8 ~ 3.6V, VREF=1.65V(REFSL[1:0]=11), -40°C 到+85°C（除非特别说明）

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		—	±0.5	±1	LSB
微分非线性	保证单调	—	±0.5	±1	LSB
偏移误差		TBD	1.0	TBD	LSB
满度误差		TBD	1.0	TBD	LSB
偏移温度系数		—	TBD	—	ppm/℃
动态性能（10kHz 正弦波单端输入，满度值之下 1dB，300ksps）					
信号与噪声加失真比		TBD	TBD	—	dB
总谐波失真	到 5 次谐波	—	TBD	—	dB
无失真动态范围		—	TBD	—	dB
转换速率					
SAR 转换时钟		—	—	8.33	MHz
转换时间（SAR 时钟数）	10 位方式 8 位方式	13 11	— —	— —	时钟
跟踪/保持捕获时间		1.7	—	—	μs
转换速率		—	—	300	ksps
模拟输入					
ADC 输入电压范围	单端方式（AIN+-GND）	0	—	VREF	V
引脚输入电压（相对于 GND）	单端	0	—	VDD	V
采样电容	1 倍增益 0.5 倍增益	— —	5 3	— —	pF
输入多路器阻抗		—	5	—	KΩ
电源指标					
电源电流（VDD 给 ADC0 供电）	工作方式，300ksps	—	TBD	—	μA
电源抑制比		—	TBD	—	mV/V
***注：**代表偏离平均值一个标准差。					

表 4.10　温度传感器电气特性

VDD = 1.8V ~3.6V；-40℃到+85℃（除非特别说明）

参　数	条　件	最小值	典型值	最大值	单 位
线性度		—	±0.2	—	℃
相对精度		—	TBD	—	℃
增益		—	4.032	—	mV/℃
增益误差*		—	TBD	—	μV/℃
偏移	温度=0℃	—	929.2	—	mV
偏移误差*	温度=0℃	—	TBD	—	mV
*注：代表偏离平均值一个标准差。					

表 4.11　电压基准电气特性

VDD = 1.8V ~3.6V；-40℃到+85℃（除非特别说明）

参　数	条　　件	最小值	典型值	最大值	单 位
内部高速基准（REFSL[1:0]＝11）					
输出电压	环境温度 25℃	TBD	1.65	TBD	V
VREF 温度系数		—	TBD	—	ppm/℃
VREF 启动时间		—	—	1.7	μs
电源抑制比	包括 ADC0 的 PSRR	—	TBD	—	ppm/V
内部精密基准（REFOE＝1）					
输出电压	环境温度 25℃	TBD	1.68	TBD	V
VREF 短路电流		—	3.5	—	mA
VREF 温度系数		—	TBD	—	ppm/℃
负载调整	负载 ＝0-200μA 到 AGND	—	400	—	μV/μA
VREF 启动时间 1	4.7μF 钽电容，0.1μF 陶瓷旁路电容稳定到 0.5 LSB	—	15	—	ms
VREF 启动时间 2	0.1μF 陶瓷旁路电容，稳定到 0.5 LSB	—	300	—	μs
VREF 启动时间 3	无旁路电容，稳定到 0.5 LSB	—	25	—	μs
电源抑制比		—	TBD	—	ppm/V
外部基准（REFOE＝0）					
输入电压范围		0	—	V_{DD}	V
输入电流	采样频率 ＝300ksps，VREF=3.0V	—	5.25	—	μA

表 4.12　IREF0 电气特性

VDD = 1.8V ~3.6V；-40℃到+85℃（除非特别说明）

参　数	条　件	最小值	典型值	最大值	单 位
静态性能					
分辨率			6		位
输出范围	低功耗模式，拉电流 大电流模式，拉电流 低功耗模式，灌电流 大电流模式，灌电流	0 0 0.3 0.8	— — — —	V_{DD}-0.4 V_{DD}-0.8 V_{DD} V_{DD}	V
绝对电流误差	低功耗模式，10μA 大电流模式，80μA	— —	1.0 1.0	— —	%
微分电流误差	大电流模式，拉电流	—	—	0.5	LSB
动态性能					
输出建立时间（到 1/2LSB）		—	300	—	ns
启动时间		—	1	—	μs
功耗					
电源电流（VDD 给 REF0 供电的电流减去任何输出拉电流）	低功耗模式，拉电流 IREF0DAT = 000001 IREF0DAT = 111111 大电流模式，拉电流 IREF0DAT = 000001 IREF0DAT = 111111 低功耗模式，灌电流 IREF0DAT = 000001 IREF0DAT = 111111 大电流模式，灌电流 IREF0DAT = 000001 IREF0DAT = 111111	— — — — — — — — — —	10 10 10 10 1 11 12 81	— — — — — — — —	μA μA μA μA μA μA μA μA

表 4.13 比较器电气特性

VDD = 1.8V ~3.6V；-40℃到+85℃（除非特别说明）

参　数	条　件	最小值	典型值	最大值	单位
响应时间：方式 0，Vcm*=1.5V	(CP0+) - (CP0-) = 100mV	—	100	—	ns
	(CP0+) - (CP0-) = -100mV	—	250	—	ns
响应时间：方式 1，Vcm*=1.5V	(CP0+) - (CP0-) = 100mV	—	175	—	ns
	(CP0+) - (CP0-) = -100mV	—	500	—	ns
响应时间：方式 2，Vcm*=1.5V	(CP0+) - (CP0-) = 100mV	—	320	—	ns
	(CP0+) - (CP0-) = -100mV	—	1100	—	ns
响应时间：方式 3，Vcm*=1.5V	(CP0+) - (CP0-) = 100mV	—	1050	—	ns
	(CP0+) - (CP0-) = -100mV	—	5200	—	ns
共模抑制比		—	1. 5	4	mV/V
正向回差电压 1	CP0HYP1-0 = 00	—	0	1	mV
正向回差电压 2	CP0HYP1-0 = 01	2	5	10	mV
正向回差电压 3	CP0HYP1-0 = 10	7	10	20	mV
正向回差电压 4	CP0HYP1-0 = 11	15	20	30	mV
负向回差电压 1	CP0HYN1-0 = 00	—	0	1	mV
负向回差电压 2	CP0HYN1-0 = 01	2	5	10	mV
负向回差电压 3	CP0HYN1-0 = 10	7	10	20	mV
负向回差电压 4	CP0HYN1-0 = 11	15	20	30	mV
反相或同相输入电压范围		-0.25	—	VDD+0.25	V
输入电容		—	4	—	pF
输入偏置电流		—	0.001	—	nA
输入偏移电压		-5	—	+5	mV
电源					
电源抑制比		—	0.1	—	mV/V
上电时间		—	10	—	µS
供电电流（DC）	方式 0	—	7.6	—	µA
	方式 1	—	3.2	—	µA
	方式 2	—	1.3	—	µA
	方式 3	—	0.4	—	µA
*注：Vcm 是 CP0+和 CP0-上的共模电压。					

表 4.14 DC-DC 转换器（DC0）电气特性

VDD = 1.8V ~3.6V；-40℃到+85℃（除非特别说明）

参　数	条　件	最小值	典型值	最大值	单位
输入电压范围		0.9	—	1.8	V
输入电感值		—	680	—	nH
输入电感电流		250	—	—	mA
建议的输入电感直流阻抗		—	5	—	Ω
输入电容值		—	4.7	—	μF
输入与输出的电压差（Vout - Vin）		0.2	—	2.4	V
输出电压范围	目标输出 = 1.8 V	—	1.8	—	V
	目标输出 = 1.9 V	—	1.9	—	
	目标输出 = 2.0 V	—	2.0	—	
	目标输出 = 2.1 V	—	2.1	—	
	目标输出 = 2.4 V	—	2.4	—	
	目标输出 = 2.7 V	—	2.7	—	
	目标输出 = 3.0 V	—	3.0	—	
	目标输出 = 3.3 V	—	3.3	—	
输出电流（基于输出功率指标）	目标输出 = 1.8 V	—	—	36	mA
	目标输出 = 1.9 V	—	—	34	
	目标输出 = 2.0 V	—	—	32	
	目标输出 = 2.1 V	—	—	30	
	目标输出 = 2.4 V	—	—	27	
	目标输出 = 2.7 V	—	—	24	
	目标输出 = 3.0 V	—	—	21	
	目标输出 = 3.3 V	—	—	19	
输出功率		—	—	65	mW
偏置电流		—	TBD	—	μA
时钟频率		1.6	2.4	3.2	MHz
启动期间的最大直流负载电流	电池阻抗 = 1 Ω 电池阻抗 = 10 Ω	— —	TBD TBD	— —	μA
连接到输出的电容		—	1.0	—	μF

表 4.15　VREG0 电气特性

VDD = 1.8V ~3.6V；-40℃到+85℃（除非特别说明）

参　数	条　　件	最小值	典型值	最大值	单 位
输入电压范围		1.8	—	3.6	V
偏置电流	正常、空闲、挂起或停机模式	—	20	—	μA

5. 10 位 SAR ADC（ADC0）

C8051F93x-C8051F92x 的 ADC0 是一个 300ksps 的 10 位逐次逼近寄存器（SAR）型 ADC，ADC 中集成了跟踪保持电路和可编程窗口检测器。ADC0 还具有自动低功耗突发模式，可以自动使能 ADC0，采集和累加样本，然后再将 ADC0 置于低功耗断电状态，所有这些操作都不需要 CPU 干预。ADC0 还有一个 16 位的累加器，可以自动过采样和对 ADC 结果求均值。

ADC0 可完全用软件通过特殊功能寄存器来配置。ADC0 可以工作在单端方式，可以被配置为测量由模拟多路器选择的不同信号（模拟多路器的说明见“5.5 ADC0 模拟多路器”）。ADC 的电压基准选择见“5.7 电压基准和地基准选项”）。

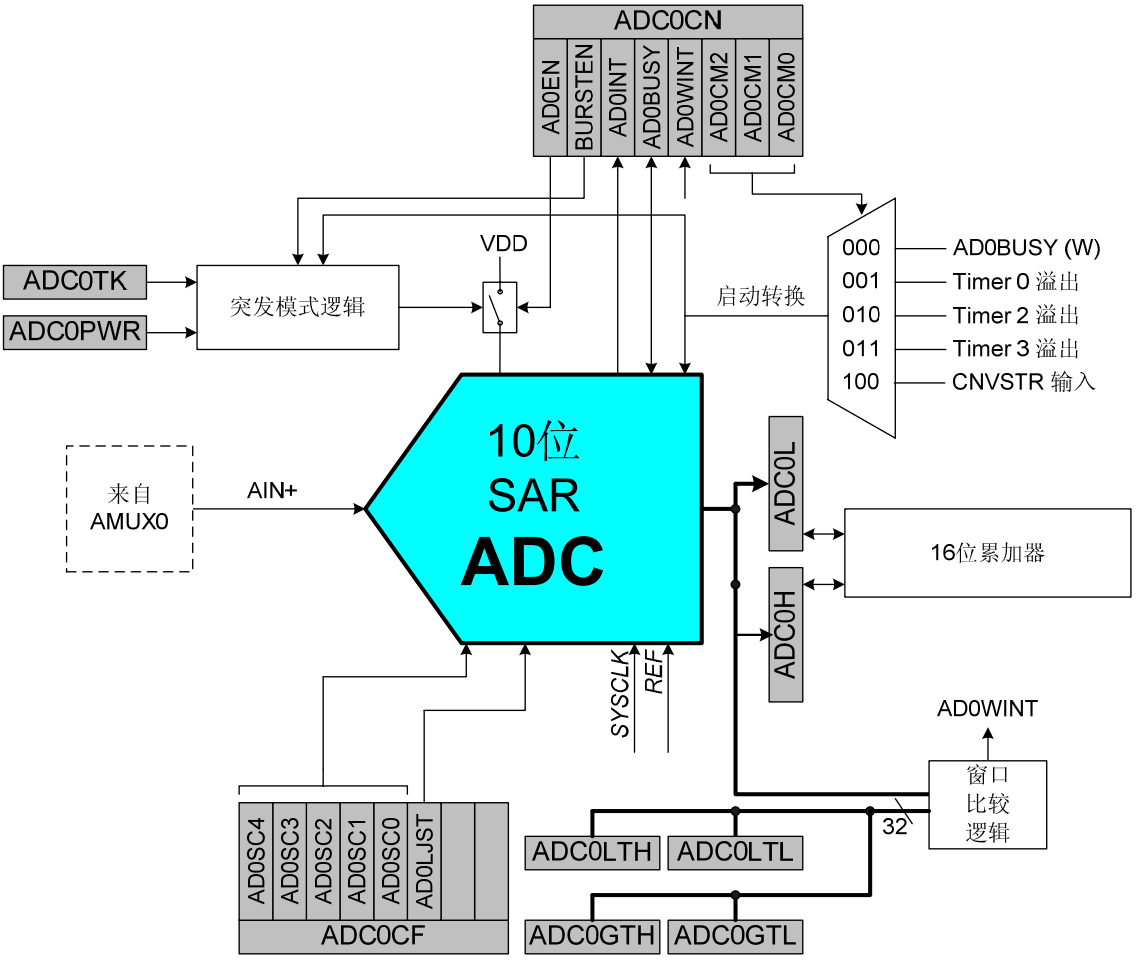


图 5.1　ADC0 功能框图

5.1 输出码格式

每次转换结束后，寄存器 ADC0H 和 ADC0L 中保存 ADC 转换结果的高字节和低字节。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 AD0LJST 位的设置决定。当重复次数被设置为 1 时，转换码以 10 位无符号整数形式表示。输入测量范围为 0 ~ VREF×1023/1024。下面的表给出了右对齐和左对齐数据的代码示例。ADC0H 和 ADC0L 寄存器中未使用的位被清 0。

输入电压	右对齐的 ADC0H:ADC0L (AD0LJST = 0)	左对齐的 ADC0H:ADC0L (AD0LJST = 1)
VREF×1023/1024	0x03FF	0xFFC0
VREF×512/1024	0x0200	0x8000
VREF×256/1024	0x0100	0x4000
0	0x0000	0x0000

当 ADC0 重复次数大于 1 时，输出转换码代表转换值累加的结果，并在一系列转换中的最后一次转换结束后被更新。可以将 4、8、16、32 或 64 个连续采样值累加并以无符号整数形式表示。重复次数用 ADC0AC 寄存器中的 AD0RPT 位选择。当重复次数大于 1 时，ADC 输出必须是右对齐的（AD0SJST = 0xx），ADC0H 和 ADC0L 寄存器中未使用的位被清 0。下面的表给出了对应不同输入电压和重复次数的右对齐结果示例。注意：当从 ADC 返回的所有采样结果都相同时，累加 2^n 个采样值等价于左移 n 位。

输入电压	重复次数 = 4	重复次数 = 16	重复次数 = 64
VREF×1023/1024	0x0FFC	0x3FF0	0xFFC0
VREF×512/1024	0x0800	0x2000	0x8000
VREF×511/1024	0x07FC	0x1FF0	0x7FC0
0	0x0000	0x0000	0x0000

AD0SJST 位可用于格式化 16 位累加器的内容。累加结果可以被右移 1、2 或 3 位。根据过采样和求均值原理，每当过采样率提高到 4 倍时，实际的 ADC 分辨率增加 1 位。下表中的样例数据演示了在没有 CPU 干预的情况下如何将实际 ADC 分辨率增加 1、2 或 3 位来分别获得 11 位、12 位或 13 位的实际 ADC 分辨率。

输入电压	重复次数 = 4 右移 1 位 11 位结果	重复次数 = 16 右移 2 位 12 位结果	重复次数 = 64 右移 3 位 13 位结果
VREF×1023/1024	0x07F7	0x0FFC	0x1FF8
VREF×512/1024	0x0400	0x0800	0x1000
VREF×511/1024	0x03FE	0x04FC	0x0FF8
0	0x0000	0x0000	0x0000

5.2 工作方式

ADC0 的最高转换速度为 300ksps。当突发模式被禁止时（BURSTEN = 0）ADC0 的转换时钟（SARCLK）由系统时钟分频得到，当突发模式被使能时（BURSTEN = 1）ADC0 的转换时钟由低功耗振荡器分频得到。时钟分频系数由 ADC0CF 寄存器的 AD0SC 位决定。

5.2.1 启动转换

有 5 种 ADC 转换启动方式，由寄存器 ADC0CN 中的 ADC0 转换启动方式位（AD0CM2-0）的状态决定采用哪一种方式。转换触发源有：

1．写 1 到寄存器 ADC0CN 的 AD0BUSY 位；
2．定时器 0 溢出（即定时的连续转换）；
3．定时器 2 溢出；
4．定时器 3 溢出；
5．CNVSTR 输入信号（P0.6）的上升沿。

向 AD0BUSY 写 1 方式提供了用软件按需要控制 ADC0 转换的能力。AD0BUSY 位在转换期间被置 1，转换结束后被清 0。AD0BUSY 位的下降沿触发中断（当被允许时）并置位 ADC0 中断标志（AD0INT）。注意：当查询 ADC 转换是否结束时，应使用 ADC0 中断标志（AD0INT）。当 AD0INT 位为逻辑 1 时，ADC0 数据寄存器（ADC0H:ADC0L）中的转换结果有效。注意：当转换源是定时器 2 溢出或定时器 3 溢出时，如果定时器 2 或定时器 3 工作在 8 位方式，使用定时器 2/3 的低字节溢出；如果定时器 2/3 工作在 16 位方式，则使用定时器 2/3 的高字节溢出。有关定时器配置方面的信息见“25. 定时器”。

需要注意的是，CNVSTR 输入引脚还是端口引脚 P0.6。当使用 CNVSTR 输入作为转换启动源时，端口引脚 P0.6 应被数字交叉开关跳过。为使交叉开关跳过 P0.6，应将寄存器 P0SKIP 中的位 6 置 1。有关端口 I/O 配置的详细信息，见“21. 端口输入/输出”。

注意：当器件工作在单节电池方式时，有一个选项可以使启动转换与 DC-DC 转换器开关周期中干扰最小的部分同步。使能该选项可以帮助减小由 DC-DC 转换器产生的内部或外部电源干扰。使能该同步控制位将使由上述任何一种转换源发出的转换启动信号被保持，直到 ADC 接收到来自 DC-DC 转换器的同步信号才开始转换。这个转换启动延迟最大为一个 DC-DC 转换器时钟周期。在 1.6MHz 的最小 DC-DC 转换器时钟频率，一个时钟周期为 625 ns。更详细的信息请见“SFR 定义 16.1 DC0CN：DC-DC 转换器控制寄存器”中对 SYNC 位的说明。ADC 在双节电池模式下工作时，该位必须被清 0。

5.2.2 跟踪方式

每次 ADC0 转换之前都必须有一个最小的跟踪时间，以保证转换结果准确。最小跟踪时间由表 4.8 给出。寄存器 ADC0CN 中的 AD0TM 位控制 ADC0 的跟踪保持方式。在缺省状态，ADC0 输入被连续跟踪（转换期间除外）。在缺省状态，当突发模式被禁止时，ADC0 输入被连续跟踪，只有转换正在进行期间除外。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，每次转换前有 3 个 SAR 时钟的跟踪时间（跟踪发生在转换启动信号有效之后）。在低功耗跟踪保持方式下使用 CNVSTR 信号启动转换时，ADC0 只在 CNVSTR 输入为低电平时跟踪；从 CNVSTR 的上升沿开始转换（见图 5.2）。当器件处于低功耗停机或休眠方式时，可以禁止跟踪（断电）。低功耗跟踪和保持方式在 AMUX 的设置经常改变时也是很有用的，因为 ADC 有建立时间要求（见“5.2.4 建立时间要求”）。

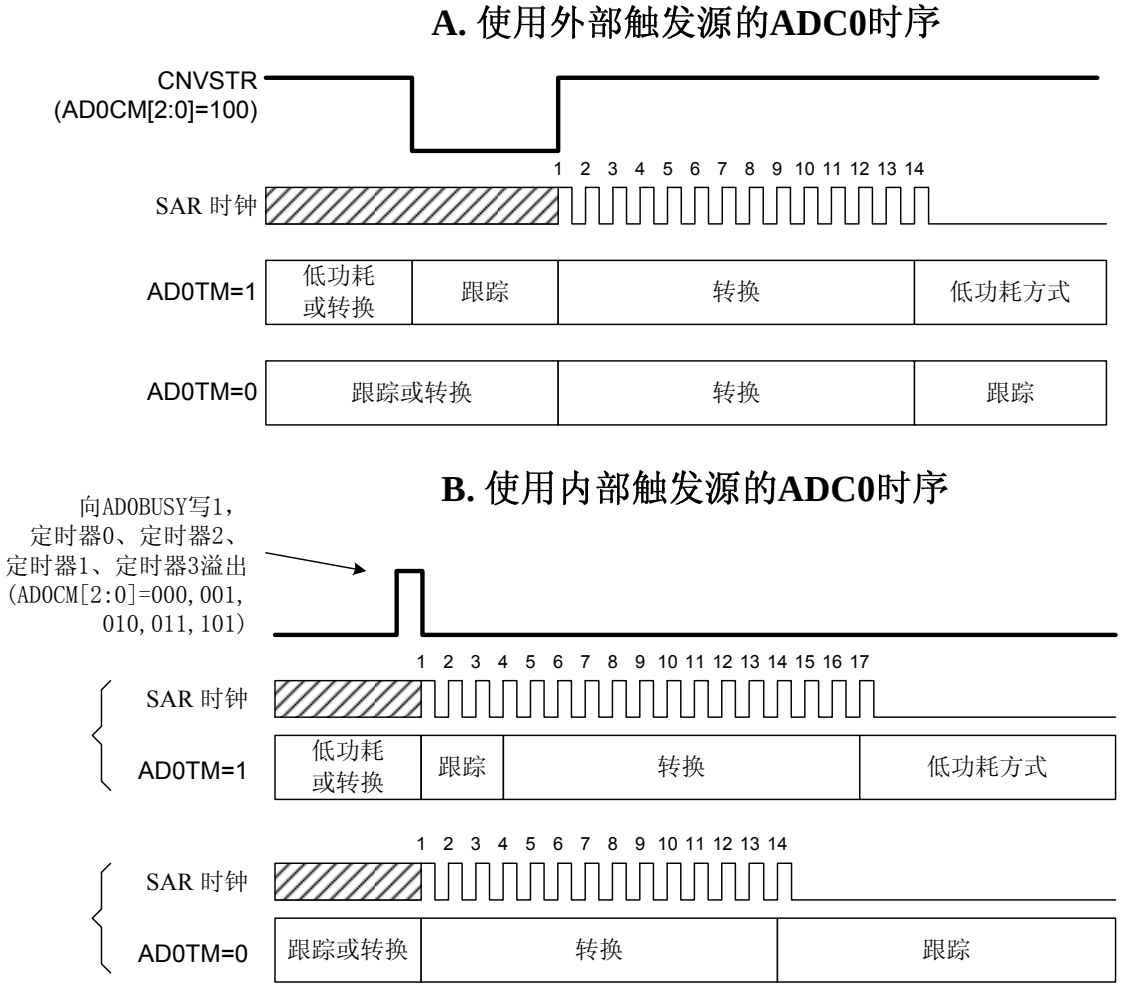


图 5.2　10 位 ADC0 跟踪和转换时序（BURSTEN = 0）

5.2.3 突发方式

突发模式是一种节省功耗的功能特性，允许 ADC0 在两次转换期间保持低功耗状态。当突发模式被使能时，ADC0 从低功耗状态被唤醒，用内部突发模式时钟（约 25 MHz）累加 1、4、8、16、32 或 64 个采样值，然后又重新进入低功耗状态。由于突发模式时钟独立于系统时钟，ADC0 可以在一个系统时钟周期内完成多次转换并重新进入低功耗状态，即使系统时钟频率很低（如 32.768 KHz）或被挂起。

将 BURSTEN 设置为逻辑 1 即可使能突发模式。当工作在突发模式时，AD0EN 控制 ADC0 的空闲电源状态（即 ADC0 不跟踪也不执行转换时进入的状态）。如果 AD0EN 被设置为逻辑 0，ADC0 在每次突发转换后进入断电状态；如果 AD0EN 被设置为逻辑 1，ADC0 在每次突发转换后保持使能状态。每来一次转换启动信号，ADC0 被从其空闲电源状态唤醒。如果 ADC0 被断电，它会自动上电并等待一个可编程的上电时间，该上电时间由 AD0PWR 位控制。否则，ADC0 会立即启动跟踪和转换。图 5.3 给出了使用慢速系统时钟且重复次数为 4 时的突发模式示例。

当突发模式被使能时，一次转换启动将进行多次转换，转换次数等于重复次数。当突发模式被禁止时，每次转换都需要有转换启动信号。在这两种情况下，在完成“重复次数”次转换和累加后，ADC0 转换结束中断会被置 1。类似地，在完成“重复次数”次转换和累加之前，窗口比较器不会将结果与“大于”或“小于”寄存器进行比较。

在突发模式，AD0PWR 和 AD0TK 的设置决定如何跟踪。在大多数应用中，可以使用这些寄存器的默认设置而无需修改。但在有些应用中，由于建立时间的要求，要对默认设置进行调整。详见“5.2.4 建立时间要求”。

注：置位 AD0TM 将在每次转换前插入 3 个额外的 SAR 跟踪时钟，这与 AD0PWR 和 AD0TK 的设置无关。

注：使用突发模式时必须注意，转换启动信号的频率不能高于 SYSCLK 频率的 1/4，包括外部转换启动信号。

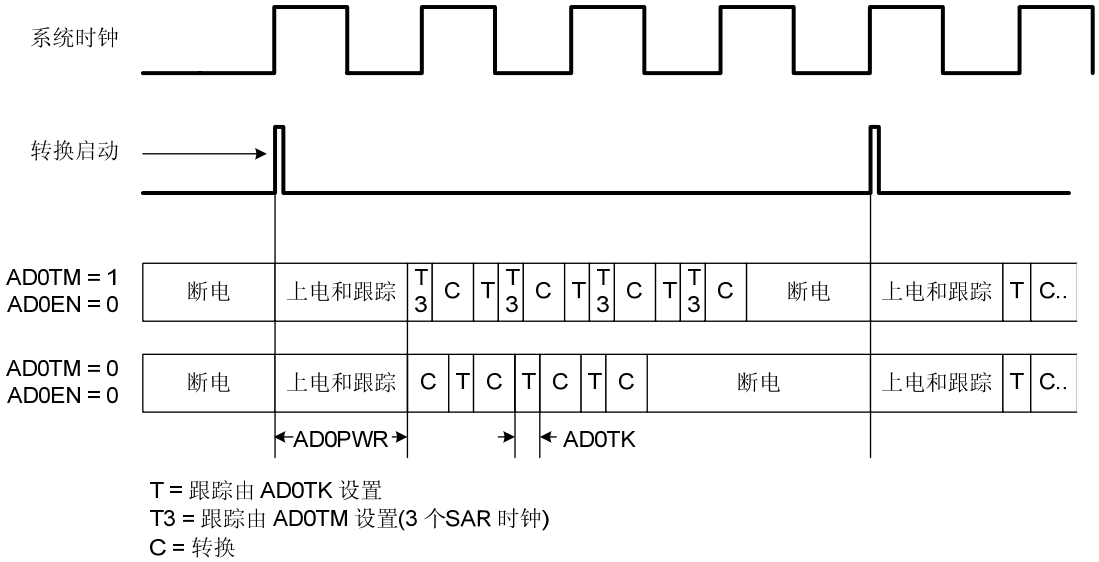


图 5.3　突发模式跟踪示例（重复次数设置为 4）

5.2.4 建立时间要求

在每次转换之前需要有一个最小的跟踪时间，以允许采样电容的电压建立起来。该跟踪时间由 AMUX0 的电阻、ADC0 采样电容、任何外部源阻抗以及所要求的转换精度决定。注意：在低功耗跟踪方式，每次转换开始时需要用三个 SAR 时钟跟踪。对于很多应用来说，三个 SAR 时钟可以满足最小跟踪时间的要求。外部源阻抗较大时，所需要的建立时间增加。

图 5.4 给出了等效的 ADC0 输入电路。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或 VDD（相对于 GND）时，R_{TOTAL} 减小为 R_{MUX}。表 4.9 给出了 ADC0 的最小建立时间要求以及多路器的阻抗和采样电容值。

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

方程 5.1　ADC0 建立时间要求

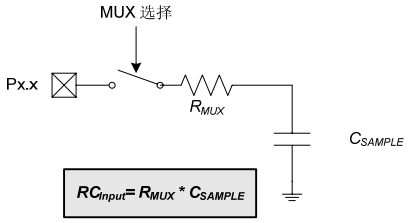
其中：

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 AMUX0 电阻与外部信号源电阻之和；

n 为 ADC 的分辨率，用比特表示（10）。



注：CSAMPLE 的值取决于 PGA 增益。详见表 4.9。

图 5.4　ADC0 等效输入电路

5.2.5 增益设置

ADC 的增益可以设置为 1 或 0.5。在增益为 1 时，ADC 的满度测量值直接由 VREF 决定。在增益为 0.5 时，ADC 的满度测量值发生在输入电压为 VREF × 2 时。0.5 倍的增益设置对于使用较低的 VREF 测量一个较高的输入电压范围或测量 VREF 和 VDD 之间的输入电压是非常有用的。ADC 的增益设置由寄存器 ADC0CF 中的 AMP0GN 位控制。

5.3 8 位方式

置位寄存器 ADC0CF 中的 ADC08BE 位将 ADC 设置为 8 位方式。8 位方式只转换数据的高 8 位，这就允许比 10 位转换少用两个 SAR 时钟周期。这也会使总体功耗降低，因为系统处于低功耗模式的时间会更长。在该方式，转换数据的最低 2 位总是为 00b，即 ADC0L 寄存器的读出值总是为 0x00。

SFR 定义 5.1 ADC0CN: ADC0 控制寄存器

SFR 页：0x00
SFR 地址：0xE8 可位寻址

R/W	R/W	R/W	W	R/W	R/W	R/W	R/W	复位值
AD0EN	BURSTEN	AD0INT	AD0BUSY	AD0WINT	AD0CM[2:0]			00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7： AD0EN：ADC0 使能位
0：ADC0 禁止（低耗断电状态）。
1：ADC0 使能。（活动状态，可以进行转换数据）。

位 6： BURSTEN：ADC0 突发模式使能位
0：突发模式禁止。
1：突发模式使能。

位 5： AD0INT：ADC0 转换结束中断标志
0：在一次数据转换（BURSTEN = 0）或一次突发转换（BURSTEN = 1）完成后由硬件置 1。可以触发中断，必须用软件清 0。

位 4： AD0BUSY：ADC0 忙标志位
当 AD0CM[2:0] = 000b 时，向该位写 1 将启动 ADC 转换。

位 3： AD0WINT：ADC0 窗口比较中断标志
在 ADC0H:ADC0L 中的内容落入由 ADC0GTH:ADC0GTL 和 ADC0LTH:ADC0LTL 所规定的窗口时由硬件置 1。
1：发生了 ADC0 窗口比较数据匹配。可以触发中断，必须用软件清 0。

位 2-0： AD0CM[2:0]：ADC0 转换启动方式选择
指定 ADC0 的转换启动源。
000：向 AD0BUSY 写 1 时启动 ADC0 转换。
001：定时器 0 溢出启动 ADC0 转换。
010：定时器 2 溢出启动 ADC0 转换。
011：定时器 3 溢出启动 ADC0 转换。
1xx：外部 CNVSTR 输入信号的上升沿启动 ADC0 转换。

SFR 定义 5.2　ADC0CF: ADC0 配置寄存器

SFR 页：0x00
SFR 地址：0xBC

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AD0SC[4:0]					AD08BE	AD0TM	AMP0GN	11111000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-3：　AD0SC4-0：ADC0 SAR 转换时钟分频系数
SAR 转换时钟来源于 FCLK，由下面的方程给出，其中 *AD0SC* 表示 AD0SC[4:0] 中保存的 5 位数值。对 SAR 转换时钟的要求见表 4.9。
BURSTEN = 0：FCLK 为当前系统时钟。
BURSTEN = 1：FCLK 20 MHz 低功耗振荡器，独立于系统时钟。

$$AD0SC = \frac{FCLK}{CLK_{SAR}} - 1*$$

*注：结果向上取整。

或

$$CLK_{SAR} = \frac{FCLK}{AD0SC + 1}$$

位 2：　AD08BE：ADC0 8 位方式使能
0：ADC0 工作在 10 位方式（正常方式）。
1：ADC0 工作在 8 位方式。

位 1：　AD0TM：ADC0 跟踪方式
选择正常方式或延迟跟踪方式。
0：正常跟踪方式。当 ADC0 被使能时，在转换启动信号有效后立即开始转换。
1：延迟跟踪方式。当 ADC0 被使能时，在转换启动信号有效后经过 3 个 SAR 时钟才开始转换。在此期间允许 ADC 跟踪。

位 0：　AMP0GN：ADC0 增益控制。
0：片内 PGA 增益为 0.5。
1：片内 PGA 增益为 1。

SFR 定义 5.3　ADC0AC: ADC0 累加器配置寄存器

SFR 页：0x00
SFR 地址：0xBA

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	AD0AE	AD0SJST[2:0]			AD0RPT[2:0]			00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　未用。读 = 0b，写 = 忽略。

位 6：　AD0AE：ADC0 累加使能
在突发模式被禁止时允许对多次转换值进行累加。
0：在突发模式被禁止时，ADC0H:ADC0L 包含最后一次转换结果。
1：在突发模式被禁止时，ADC0H:ADC0L 包含多次转换的累加结果。要清除累加结果，软件必须向 ADC0H:ADC0L 写 0x0000。

位 5-3：　AD0SJST[2:0]：ADC0 累加器移位和对齐控制
指定从 ADC0H:ADC0L 读出的数据的格式。
000：右对齐。无移位。
001：右对齐。右移 1 位。
010：右对齐。右移 2 位。
011：右对齐。右移 3 位。
100：左对齐。无移位。
所有其他位组合保留。

位 2-0：　AD0RPT[2:0]：ADC0 重复次数
选择在突发模式执行的转换和累加次数。
000：执行和累加 1 次转换。
001：执行和累加 4 次转换。
010：执行和累加 8 次转换。
011：执行和累加 16 次转换。
100：执行和累加 32 次转换。
101：执行和累加 64 次转换。
所有其他位组合保留。

SFR 定义 5.4　ADC0PWR: ADC0 突发模式上电时间控制寄存器

SFR 页：0x0F
SFR 地址：0xBA

R	R	R	R	R/W	R/W	R/W	R/W	复位值
—	—	—	—	AD0PWR[3:0]				00001111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-4：　未用。读 = 0000b，写 = 忽略。

位 3-0：　AD0PWR[3:0]：ADC0 突发模式上电时间控制

设置 ADC0 从低功耗状态上电所需要的延迟时间。

对于 BURSTEN = 0：

ADC0 的电源状态由 AD0EN 控制。

对于 BURSTEN = 1 且 AD0EN = 1：

ADC0 保持使能状态，在所有转换都结束后不进入低功耗状态。在转换启动信号有效后可立即开始转换。

对于 BURSTEN = 1 且 AD0EN = 0：

ADC0 在所有转换都结束后进入低功耗状态（如表 5.1 所示）。在转换启动信号有效后经过一个编程的延迟再开始转换。

ADC0 突发模式的上电时间根据下面的方程编程：

$$AD0PWR = \frac{\text{启动时间}}{200ns} - 1$$

或

$$\text{启动时间} = (AD0PWR + 1) \times 200ns$$

SFR 定义 5.5　ADC0TK: ADC0 突发模式跟踪时间控制寄存器

SFR 页：0x0F
SFR 地址：0xBD

R	R	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	AD0TK[5:0]						00111110
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-6：　未用。读 ＝00b，写 ＝ 忽略。
位 5-0：　AD0TK[5:0]：ADC0 突发模式跟踪时间控制
设置在突发模式下两次连续转换之间的延迟时间。

ADC0 突发模式的跟踪时间根据下面的方程编程：

$$AD0TK = \frac{\text{跟踪时间}}{50ns} - 1$$

或

$$\text{跟踪时间} = (AD0TK + 1) \times 50ns$$

注：如果 AD0TM 被置 1，在开始转换前要插入 3 个 SAR 时钟周期的跟踪时间。
注：在突发模式，在第一次转换之前不插入突发模式跟踪延迟。第一次转换所需要的跟踪时间应满足突发模式的上电时间。

SFR 定义 5.6　ADC0H: ADC0 数据字高字节寄存器

SFR 页：0x00
SFR 地址：0xBE

R/W								复位值
ADC0[15:8]								00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　ADC0 数据字高字节。
读：
根据 AD0SJST[2:0]的设置被格式化后的 16 位 ADC0 累加器的高字节。
写：
将 16 位 ADC0 累加器的高字节设置为写入值。

注：如果累加器移位被使能，则该寄存器的最高几位的读出值是 0。

SFR 定义 5.7　ADC0L: ADC0 数据字低字节寄存器

SFR 页：0x00
SFR 地址：0xBD

R/W								复位值
ADC0[7:0]								00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　ADC0 数据字低字节。
读：
根据 AD0SJST[2:0]的设置被格式化后的 16 位 ADC0 累加器的低字节。
写：
将 16 位 ADC0 累加器的低字节设置为写入值。

注：如果累加器移位被使能，则该寄存器的最高几位的值是累加器高字节后的低几位。

5.4 可编程窗口检测器

ADC 可编程窗口检测器不停地将 ADC0 输出寄存器与用户编程的极限值进行比较，并在检测到所期望的条件时通知系统。这在一个中断驱动的系统中尤其有效，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（寄存器 ADC0CN 中的 AD0WINT）也可被用于查询方式。ADC0 下限（大于）寄存器（ADC0GTH:ADC0GTL）和 ADC0 上限（小于）寄存器（ADC0LTH:ADC0LTL）中保持比较值。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 ADC0GT 和 ADC0LT 寄存器的内容。

SFR 定义 5.8 ADC0GTH: ADC0 下限（大于）数据字高字节寄存器

SFR 页：0x00
SFR 地址：0xC4

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0： AD0GT[15:8]：ADC0 下限数据字高字节。
16 位下限窗口比较寄存器的高字节。

SFR 定义 5.9 ADC0GTL: ADC0 下限（大于）数据字低字节寄存器

SFR 页：0x00
SFR 地址：0xC3

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0： AD0GT[7:0]：ADC0 下限数据字低字节。
16 位下限窗口比较寄存器的低字节。

注：在 8 位方式，该寄存器应被设置为 0x00。

SFR 定义 5.10 ADC0LTH: ADC0 上限（小于）数据字高字节寄存器

SFR 页：0x00
SFR 地址：0xC6

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	

位 7-0：　AD0LT[15:8]：ADC0 上限数据字高字节。
16 位上限窗口比较寄存器的高字节。

SFR 定义 5.11 ADC0LTL: ADC0 上限（小于）数据字低字节寄存器

SFR 页：0x00
SFR 地址：0xC5

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	

位 7-0：　AD0LT[7:0]：ADC0 上限数据字低字节。
16 位上限窗口比较寄存器的低字节。

注：在 8 位方式，该寄存器应被设置为 0x00。

5.4.1 单端方式下的窗口检测器

图 5.5 给出了单端方式下数据右对齐窗口比较的两个例子。左边的例子所使用的极限值为：ADC0LTH:ADC0LTL = 0x0080（128d）和 ADC0GTH:ADC0GTL = 0x0040（64d）；右边的例子所使用的极限值为：ADC0LTH:ADC0LTL = 0x0040 和 ADC0GTH:ADC0GTL = 0x0080。输入电压范围为 0 ~ VREF×1023/1024（相对于 GND），转换码是 10 位无符号整数。对于左边的例子，如果 ADC0 转换结果数据字位于由 ADC0GTH:ADC0GTL 和 ADC0LTH:ADC0LTL 定义的范围之内（即 0x0040 < ADC0H:ADC0L < 0x0080），则会产生一个 AD0WINT 中断。对于右边的例子，如果 ADC0 转换结果数据字位于由 ADC0GT 和 ADC0LT 定义的范围之外（即 ADC0H:ADC0L < 0x0040 或 ADC0H:ADC0L > 0x0080），则会产生一个 AD0WINT 中断。图 5.6 给出了使用相同比较值的单端方式下数据左对齐窗口比较的例子。

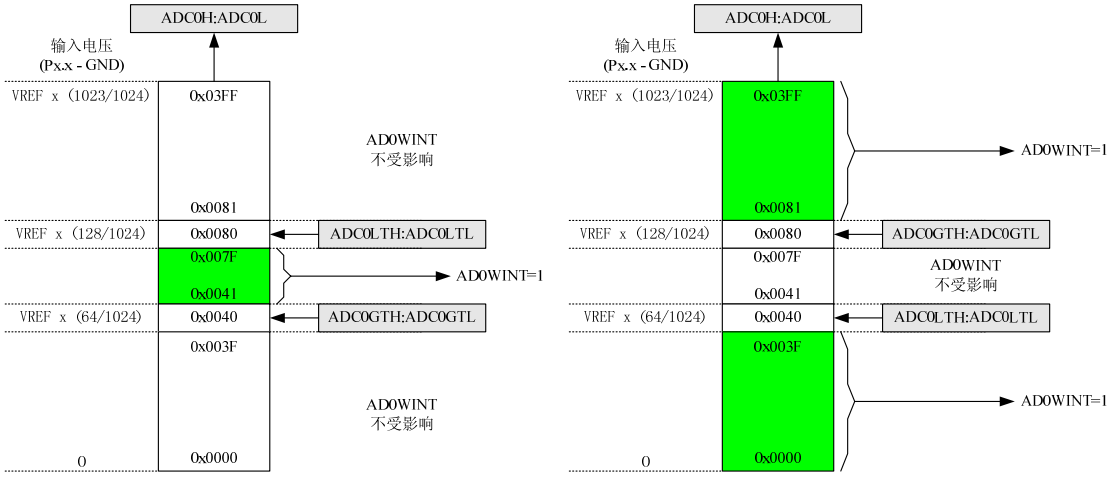


图 5.5　ADC 窗口比较示例（单端方式右对齐数据）

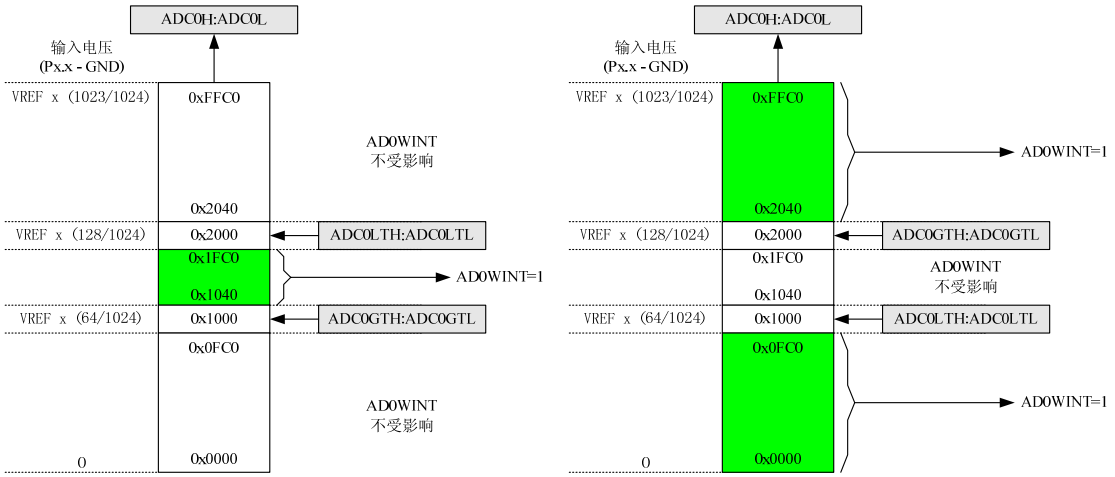


图 5.6　ADC 窗口比较示例（单端方式左对齐数据）

5.4.2 ADC0 技术指标

详细的 ADC0 技术指标见“4. 电气特性”。

5.5　ADC0 模拟多路选择器

C8051F93x-C8051F92x 有一个模拟多路器，成为 AMUX0。

AMUX0 选择单端 ADC0 的正输入。端口 I/O 引脚、片内温度传感器输出、VBAT 电源、调整后的数字电源电压（REG1 的输出）VDD/DC+中的任何一个都可以被选择为正输入，正输入也可以连接到地。ADC0 的输入通道由寄存器 ADC0MX 选择（见 SFR 定义 5.12）。

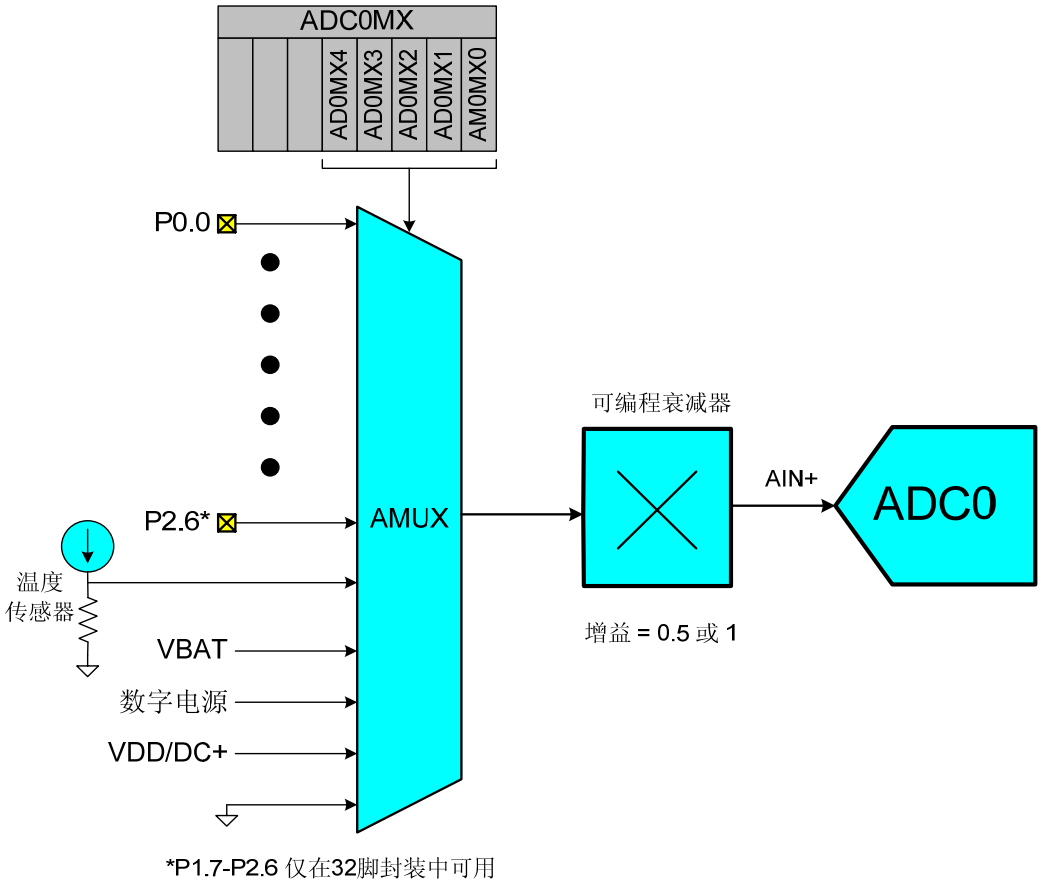


图 5.7　ADC0 多路选择器原理框图

需要特别注意的是，被选择为 ADC0 输入的引脚应被配置为模拟输入，并且应被数字交叉开关跳过。要将一个端口引脚配置为模拟输入，应将 PnMDIN 寄存器中的对应位置 0，并禁止数字驱动器（PnMDOUT = 0，端口锁存器 = 1）。为了使交叉开关跳过一个端口引脚，应将 PnSKIP 寄存器中的对应位置 1。有关端口 I/O 配置的详细信息见“21. 端口输入/输出”。

SFR 定义 5.12　ADC0MX: ADC0 输入通道选择寄存器

SFR 页：0x00
SFR 地址：0xBB

R	R	R	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	AD0MX					00011111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-5：　未使用。读=000b，写=忽略。
位 4-0：　AD0MX[4:0]：AMUX0 输入选择
选择 ADC0 的正输入通道。

AD0MX4-0	ADC0 输入通道
00000	P0.0
00001	P0.1
00010	P0.2
00011	P0.3
00100	P0.4
00101	P0.5
00110	P0.6
00111	P0.7
01000	P1.0
01001	P1.1
01010	P1.2
01011	P1.3
01100	P1.4
01101	P1.5
01110	P1.6
01111	P1.7（仅 C8051F920/30）
10000	P2.0（仅 C8051F920/30）
10001	P2.1（仅 C8051F920/30）
10010	P2.2（仅 C8051F920/30）
10011	P2.3（仅 C8051F920/30）
10100	P2.4（仅 C8051F920/30）
10101	P2.5（仅 C8051F920/30）
10110	P2.6（仅 C8051F920/30）
10111~11010	保留
11011	温度传感器
11100	VBAT 电源电压（0.9 ~ 1.8V 或 1.8 ~ 3.6V）
11101	数字电源电压（REG1 输出，典型值为 1.7V）
11110	VDD/DC+电源电压（1.8 ~ 3.6V）
11111	GND

5.6 温度传感器

C8051F93x-C8051F92x 内部包含一个温度传感器，可以通过单端方式的 ADC 多路器访问。为了使用 ADC 测量温度传感器，ADC MUX 的正通道应被配置为连接到温度传感器，ADC MUX 的负通道应被配置为连接到 GND。温度传感器的传输函数示于图 5.8。当 ADC 多路器被正确设置时，输出电压（V_{TEMP}）为 ADC 的正输入。寄存器 REF0CN 中的 TEMPE 位用于使能/禁止温度传感器（见 SFR 定义 5.15）。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 ADC0 测量结果都是无意义的。温度传感器的增益和偏移参数请参见表 4.9。

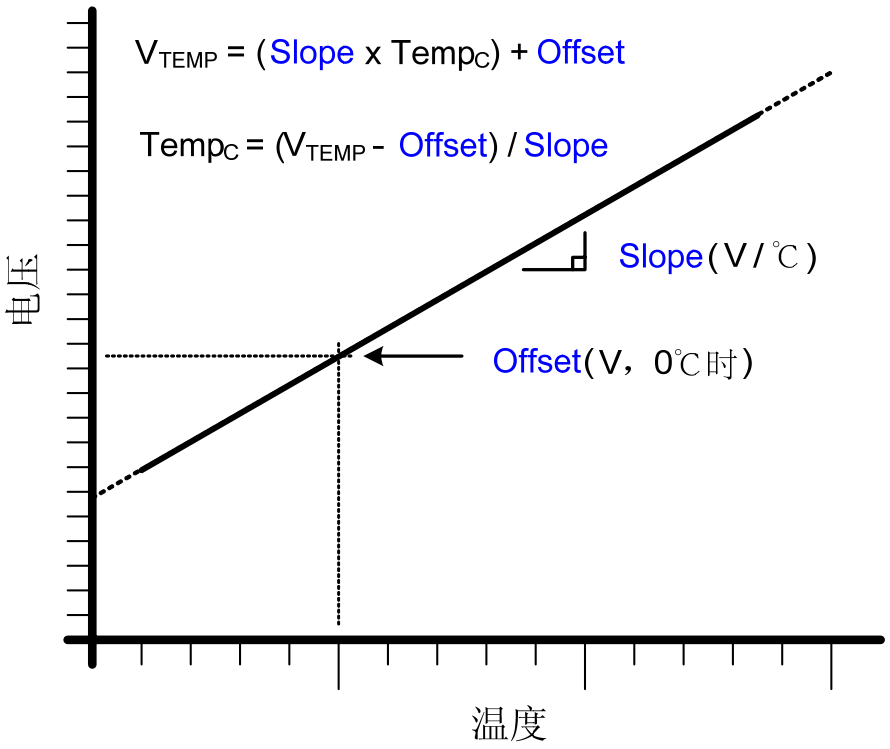


图 5.2 温度传感器典型传输函数

5.6.1 校准

未经校准的温度传感器输出的线性度非常好，适合于相对温度测量（见表 4.10 的线性度指标）。对于绝对温度测量，建议对偏移和/或增益进行校准。典型的单点（偏移）校准包括下面几个步骤：

1. 控制/测量环境温度（该温度必须是已知的）。
2. 给器件上电，等待几秒钟使器件自热。
3. 将温度传感器选择为正输入，将 GND 选择为负输入，进行一次 ADC 转换。
4. 计算偏移特性值，并将该值存入到非易失性存储器中，以备以后使用温度传感器进行温度测量时使用。

图 5.9 示出了典型的温度传感器误差曲线（假设在 25℃进行一点校准）。**注意：影响 ADC 测量精度的参数，尤其是基准电压值，也同样会影响温度测量的精度。**

在产品测试期间要对每个温度传感器进行单点偏移测量。该测量在 25℃±TBD℃下用 ADC 进行，选择内部高速基准缓冲器作为电压基准。ADC 的直接测量结果存储在 SFR 寄存器 TOFFH 和 TOFFL 中，见 SFR 定义 5.13 和 SFR 定义 5.14。

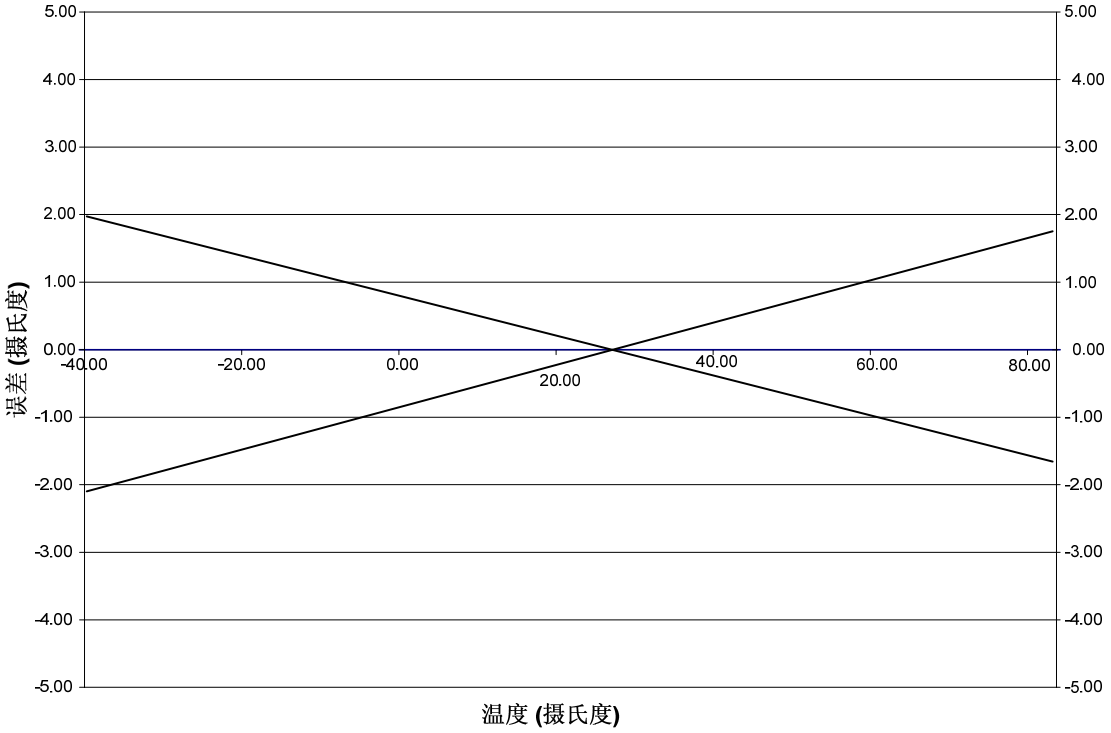


图 5.9 单点校准的温度传感器误差（VERF = 1.68 V）

SFR 定义 5.13　TOFFH: ADC0 校准数据字高字节

SFR 页：0x0F
SFR 地址：0x86

R	R	R	R	R	R	R	R	复位值
—	—	—	—	—	—	TOFF[9:8]		00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-2：　未用。读 = 000000b，写 = 忽略。
位 1-0：　TOFF[9:8]：温度传感器偏移值高位
10 位温度传感器偏移测量值的高 2 位。

SFR 定义 5.14　TOFFL: ADC0 校准数据字低字节

SFR 页：0x0F
SFR 地址：0x85

R								复位值
TOFF[7:0]								00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　TOFF[9:8]：温度传感器偏移值低字节
10 位温度传感器偏移测量值的低字节。

5.7 电压基准和地基准选项

C8051F93x-C8051F92x 的电压基准 MUX 可以被配置为连接到外部电压基准、两个内部电压基准之一或两个电源电压之一（见图 5.10）。地基准 MUX 允许为 ADC0 选择一个地基准，地基准可以在地引脚（GND）和一个专用的模拟地（P0.1/AGND）之间选择。

用特殊功能寄存器 REF0CN 选择电压基准和地基准。这两个基准的技术指标见电气特性一章。

有关 VREF 和 AGND 输入的重要注意事项：外部 VREF 和 AGND 要使用端口引脚。当使用外部电压基准或内部精密基准时，P0.0/VREF 应被配置为模拟输入并被数字交叉开关跳过。当使用 AGND 作为 ADC0 的地基准时，P0.1/AGND 应被配置为模拟输入并被数字交叉开关跳过。有关端口 I/O 配置的详细信息见“21. 端口输入/输出”。外部基准的电压必须在 0≤VREF≤VDD/DC+的范围内，外部地基准必须与 GND 的直流电压电平相同。

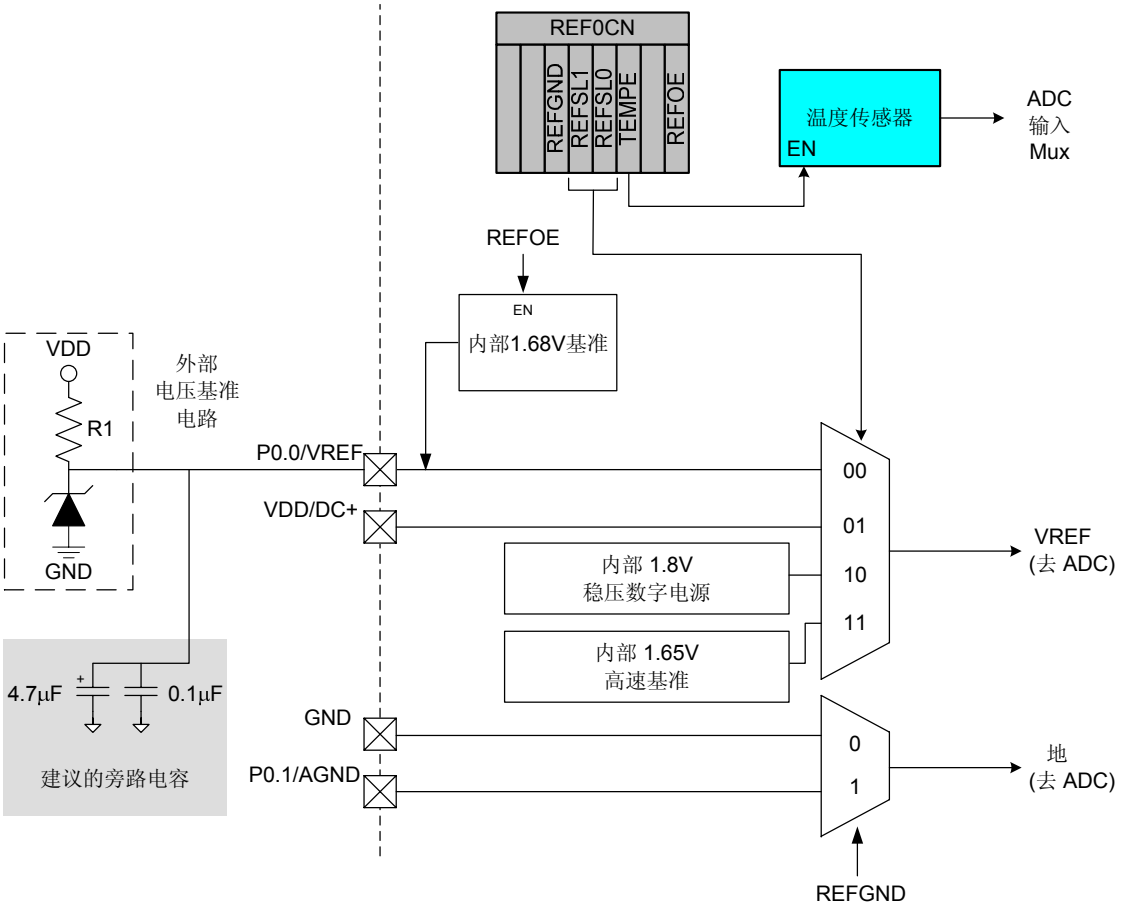


图 5.10 电压基准功能框图

5.8 外部电压基准

为了使用一个外部电压基准，REFSL[1:0]应被设置为 00，并且应禁止内部的 1.68V 精密基准（通过将 REFOE 设置为 0）。应按外部电压基准生产商的建议加旁路电容。

5.9 内部电压基准

对于需要最多端口 I/O 引脚或很短 VREF 启动时间的应用来说，1.65V 高速基准是最佳的内部基准选项。通过将 REFSL[1:0]设置为 11 来选择高速内部基准。当被选择时，高速内部基准会根据 ADC0 的需要被自动使能/禁止。

对于需要最高绝对精度的应用来说，1.68V 精密基准是最佳的内部基准选项。通过将 REFOE 置 1 和将 REFSL[1:0]设置为 00 来使能和选择 1.68V 精密基准。当被选择时，高速内部基准会根据 ADC0 的需要被自动使能/禁止。当使用精密电压基准时，建议接一个至少为 0.1μF 的外部电容。

在精密内部振荡器一直处于运行状态的应用中，使用精密电压基准不需要额外的功耗。在所有其他应用中，使用高速基准会导致较低的总体功耗，因为高速基准的启动时间很短，并且在不进行 ADC 转换时保持在低功耗状态。

注：当使用精密内部振荡器作为系统时钟源时，不应将精密电压基准从禁止状态使能。为了同时使用精密振荡器和精密电压基准，应首先使能精密电压基准，并允许其稳定到最终值（对外部电容充电）。然后再启动精密振荡器并将其选择为系统时钟。

对于电源电压保持不变的应用来说，使用电源作为电压基准可以为 ADC0 提供额外的动态范围，代价是电源噪声的抑制能力降低。为了使用 1.8 ~ 3.6 V 的电源电压（VDD/DC+）或 1.8V 的稳压后的数字电源电压作为基准源，REFSL[1:0]应分别设置为 01 或 10。

5.10 模拟地基准

为了防止开关数字逻辑产生的噪声影响敏感的模拟测量，器件提供了一个分开的模拟地基准选项。当该选项被使能时，ADC0 的地基准取自 P0.1/AGND 引脚。ADC0 采样的任何外部传感器都应以 P0.1/AGNG 引脚为基准。通过将 REFGND 置 1 来使能分开的模拟地基准选项。

5.11 温度传感器使能

寄存器 REF0CN 中的 TEMPE 位用于使能/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 ADC0 测量结果都是无意义的。有关温度传感器被使能时的特性的详细说明见“5.6 温度传感器”。

SFR 定义 5.15 REF0CN: 电压基准控制寄存器

SFR 页：0x00
SFR 地址：0xD1

R	R	R/W	R/W		R/W	R	R/W	复位值
—	—	REFGND	REFSL		TEMPE	—	REFOE	00011000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-6： 未用。读 = 00b，写 = 忽略。
位 5： REFGND：模拟地基准选择
选择 ADC0 的地基准。
0：ADC0 的地基准为 GND 引脚。
1：ADC0 的地基准为 P0.1/AGND 引脚。
位 4-3： REFSL：电压基准选择
该位选择 ADC0 的电压基准。
00：ADC0 的电压基准为 P0.0/VREF 引脚。
01：ADC0 的电压基准为 VDD/DC+引脚。
10：ADC0 的电压基准为内部 1.8V 数字电源电压。
11：ADC0 的电压基准为内部 1.65V 高速电压基准。
位 2： TEMPE：温度传感器使能
使能/禁止内部温度传感器。
0：温度传感器禁止。
1：温度传感器使能。
位 1： 未用。读 = 0b，写 = 忽略。
位 0： REFOE：内部电压基准输出使能
将内部电压基准与 P0.0/VREF 引脚连接/断开。
0：内部 1.68V 精密电压基准被禁止，不连接到 P0.0/VREF。
1：内部 1.68V 精密电压基准被使能，连接到 P0.0/VREF。

5.12 电压基准电气指标

有关电压基准电气指标的详细信息见“4. 电气特性”。

6. 可编程电流基准（IREF0）

C8051F93x-C8051F92x 内部包含一个可编程的电流基准（拉电流或灌电流），该电流基准有两种输出电流设置模式：低功耗模式和大电流模式。低功耗模式的最大电流输出为 63 μA（每级 1 μA），大电流模式的最大电流输出为 504 μA（每级 8 μA）。

拉电流/灌电流通过 IREF0CN 特殊功能寄存器控制。通过设置一个非零值的期望输出电流来使能 IREF0，通过向 ISRC0CN 写 0x00 来禁止 IREF0。IREF0 的对应端口 I/O 引脚应被配置为模拟输入并被交叉开关跳过。详见“21. 端口输入/输出”。

SFR 定义 6.1　IREF0CN: 电流基准控制寄存器

SFR 页：0x00
SFR 地址：0xB9

R/W	R/W	R/W						复位值
SINK	MDSEL	IREF0DAT						00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　SINK：IREF0 灌电流使能
选择 IREF0 是提供拉电流还是提供灌电流。
0：IREF0 提供拉电流。
1：IREF0 提供灌电流。

位 6：　MDSEL：IREF0 输出方式选择
选择低功耗或大电流模式。
0：选择低功耗模式（步长 = 1 μA）。
1：选择大电流模式（步长 = 8 μA）。

位 3：　保留。读 = 0b。写 = 0b。

位 5-0：　IREFDAT[5:0]：IREF0 数据字
指定获得期望输出电流所需要的步数。
输出电流 = 方向 × 步长 × IREF0DAT。
当 IREF0DAT 被设置为 0x00 时，IREF0 处于低功耗状态。

6.1　IREF0 技术指标

有关 IREF0 技术指标的详细信息见“4. 电气特性”。

7. 比较器

C8051F93x-C8051F92x 器件内部有 2 个可编程电压比较器：比较器 0（CPT0）示于图 7.1，比较器 1（CPT1）示于图 7.2。两个比较器的操作完全相同，但只有比较器 0 具有作复位源或唤醒源的能力。有关复位源和低功耗唤醒源的详细信息，请分别见复位源和电源管理的章节。

比较器提供可编程的响应时间和回差电压。比较器有一个模拟输入多路器和两个可以通过交叉开关接到外部引脚的输出：一个同步“锁存”输出（CP0、CP1）和一个异步“直接”输出（CP0A、CP1A）。即使在系统时钟停止时，异步的 CP0A 信号仍然可用，这就允许比较器在器件处于某些低功耗模式时工作并产生输出。

7.1 比较器输入

每个比较器对其正输入（CP0+或 CP1+）和负输入（CP0-或 CP1-）端的电平进行模拟比较。两个比较器都支持多个端口引脚输入，用模拟输入多路器将这些引脚复用到比较器的正输入和负输入。模拟输入多路器完全由软件通过特殊功能寄存器控制和配置。

有关比较器输入的重要注意事项：被选择为比较器输入的引脚应被配置为模拟输入并被交叉开关跳过，有关如何将端口 I/O 配置为模拟输入的详细信息见端口 I/O 一章。比较器也可以用于比较数字信号的逻辑电平，但被配置为数字输入的端口 I/O 引脚必须被驱动到一个有效的逻辑状态（高或低），以避免增加功耗。

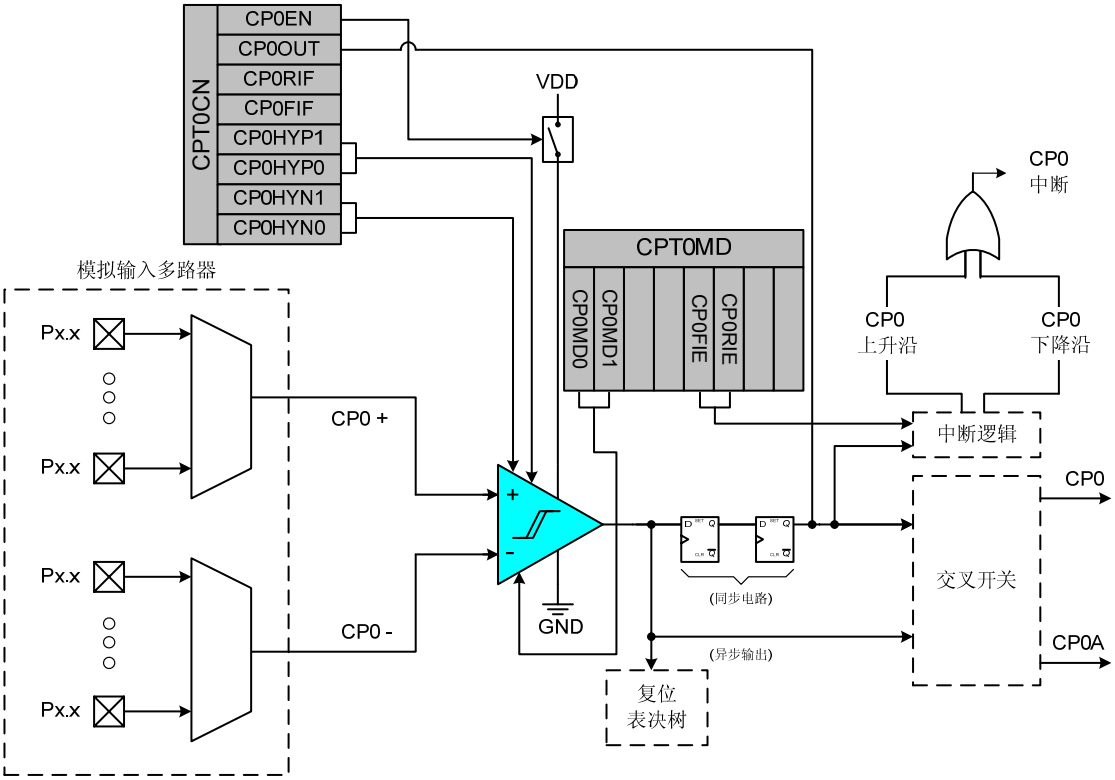


图 7.1 比较器 0 功能框图

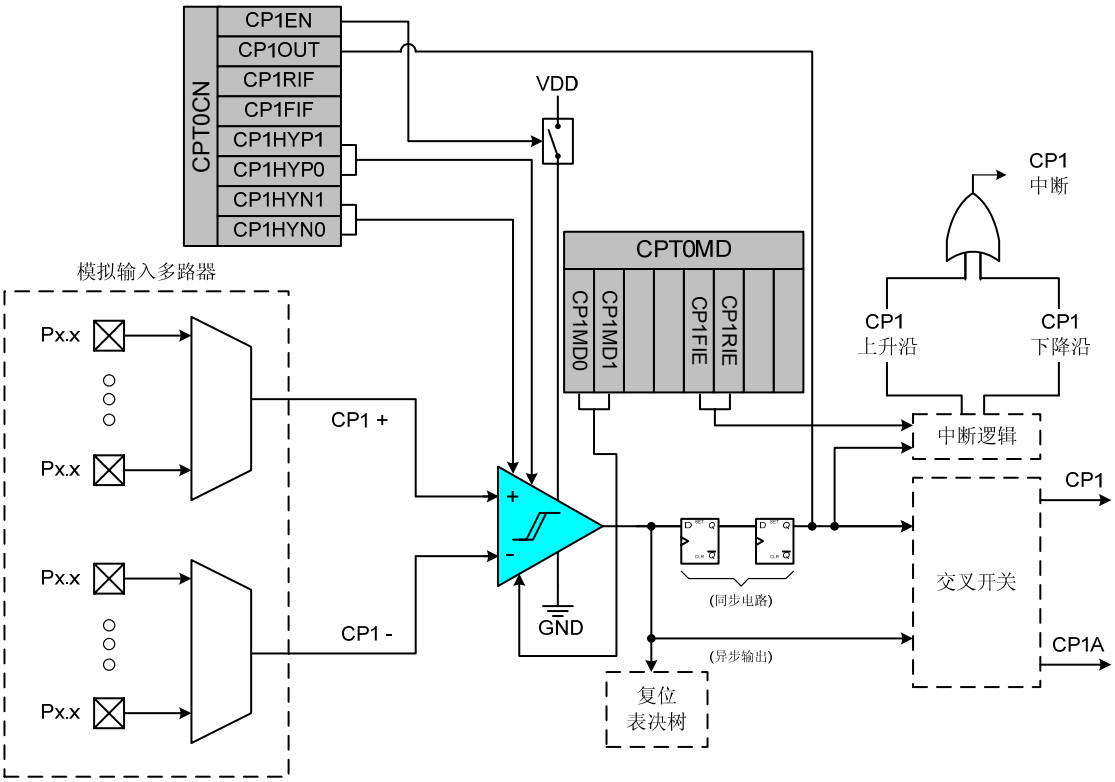


图 7.2 比较器 1 功能框图

7.2 比较器输出

当一个比较器被使能时，如果正输入端的电压大于负输入端的电压，则其输出为逻辑 1。当被禁止时，比较器输出为逻辑 0。异步比较器输出与系统时钟同步后得到同步输出（如图 7.2 所示）。该同步“锁存”输出（CP0, CP1）可以被软件查询（用 CPnOUT 位），用作中断源，或通过交叉开关连接到一个端口引脚。

比较器的异步“直接”输出（CP0A, CP1A）由低功耗模式唤醒逻辑和复位表决逻辑使用。有关异步比较器输出如何用作唤醒源和复位源的详细信息，请见电源选项和复位源的章节。异步比较器输出也可以被交叉开关直接连接到一个端口引脚，即使在系统时钟停止时也可被器件外部的电路使用。

当使用比较器作为中断源时，比较器输出的上升沿和/或下降沿都可以产生中断。两个独立的中断标志（CPnRIF 和 CPnFIF）允许软件确定是哪个边沿引起的比较中断。当比较器检测到上升沿或下降沿时，相应的中断标志被硬件置 1，与中断使能状态无关。这两个标志位一旦被置 1，将一直保持置 1 状态，直到被软件清除。

可以用 CPTnMD 寄存器中的中断使能位 CPnRIE 和 CPnFIE 分别使能上升沿和下降沿中断。为了使 CPnRIF 和/或 CPnFIF 中断标志能向 CPU 请求中断，比较器必须被使能为中断源，并且总体中断必须被使能。更详细的信息见中断系统一章。

7.3 比较器响应时间

可以通过对寄存器 CPTnMD（见“CPT0MD：比较器 0 方式选择”和“CPT1MD：比较器 1 方式选择”）编程来设置比较器的响应时间。有四种响应时间设置：方式 0（最快响应时间）、方式 1、方式 2 和方式 3（最低功耗）。选择较长的响应时间可以减小比较器活动电源电流。比较器还有低功耗断电状态，比较器在被禁止时进入该状态。比较器上升沿和下降沿响应时间通常是不相等的。

7.4 比较器回差电压

比较器有可编程的回差电压，回差电压用于在输入端发生干扰时稳定比较器的输出。用户可以使用 CPTnCN 寄存器对回差电压值（输入电压）编程，也可以对该回差电压围绕门限电压（即比较器的负输入）的正向和负向回差对称度编程。

图 7.3 示出了当正向回差电压被使能时，比较器输出在正输入电压不超过门限电压之上一个编程的回差电压之前不会从逻辑 0 变到逻辑 1。该图也示出了当负向回差电压被使能时，比较器输出在正输入电压没低于门限电压之下一个编程的回差电压之前不会从逻辑 1 变到逻辑 0。

正向回差电压由 CPTnCN 寄存器中 CPnHYP 位的设置决定，负向回差电压由 CPTnCN 寄存器中 CPnHYN 位的设置决定。可以设置 20 mV、10 mV、5 mV 或 0 mV 的正向和负向回差电压值。

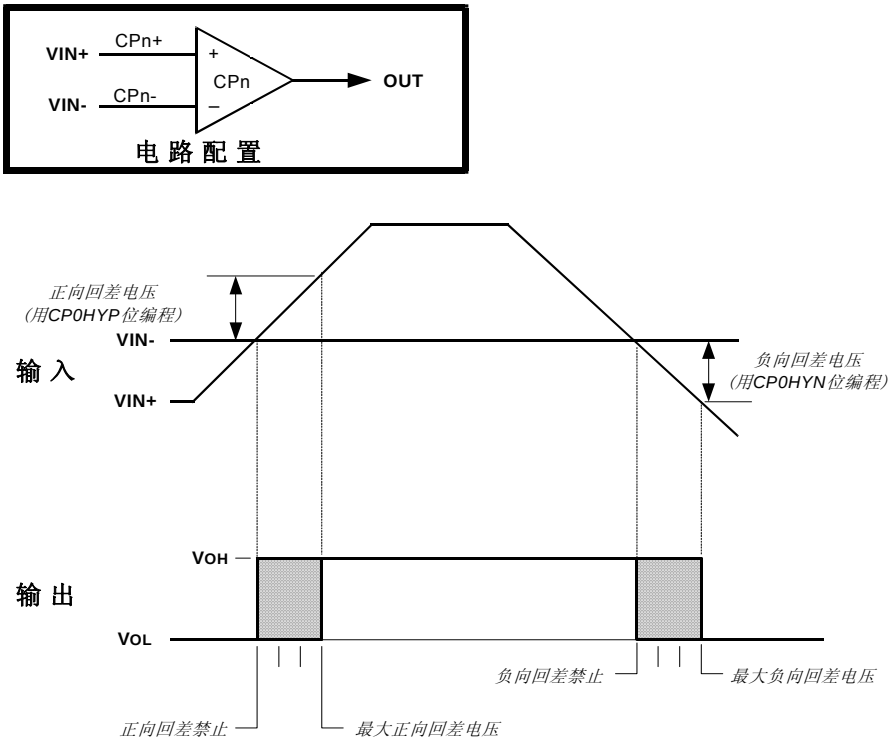


图 7.3 比较器回差电压曲线

7.5 比较器寄存器说明

下面的寄存器说明对用于使能和配置比较器的特殊功能寄存器进行描述。在使用比较器前必须通过将 CPnEN 位置 1 来使能比较器。通过将 CPnEN 位清 0 可以将比较器从使能状态禁止，并将其置于低功耗状态。

有关比较器设置的重要注意事项：在对比较器上电或改变比较器的回差电压或响应时间控制位时，可能会检测到假上升沿或下降沿。因此，建议在比较器被使能或方式位改变后经过一段延时后将上升沿和下降沿标志显式清 0。

SFR 定义 7.1　CPT0CN: 比较器 0 控制寄存器

SFR 页：0x00
SFR 地址：0x9B

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CP0EN	CP0OUT	CP0RIF	CP0FIF	CP0HYP1	CP0HYP0	CP0HYN1	CP0HYN0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　CP0EN：比较器 0 使能位
0：比较器 0 禁止。
1：比较器 0 使能。

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，必须用软件清 0。
0：自该标志位最后一次被清除后，未检测到比较器 0 上升沿。
1：检测到比较器 0 上升沿。

位 4：　CP0FIF：比较器 0 下降沿中断标志，必须用软件清 0
0：自该标志位最后一次被清除后，未检测到比较器 0 下降沿。
1：检测到比较器 0 下降沿。

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= 5mV。
10：正向回差电压= 10mV。
11：正向回差电压= 20mV。

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= 5mV。
10：负向回差电压= 10mV。
11：负向回差电压= 20mV。

SFR 定义 7.2　CPT0MD: 比较器 0 方式选择寄存器

SFR 页：0x00
SFR 地址：0x9D

R/W	R	R/W	R/W	R	R	R/W	R/W	复位值
保留	—	CP0RIE	CP0FIE	—	—	CP0MD1	CP0MD0	10000010
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　保留：读 ＝1b，必须写 1b。
位 6：　未用。读 ＝0b，写 ＝ 忽略。
位 5：　CP0RIE：比较器 0 上升沿中断允许
　　0：比较器 0 上升沿中断禁止
　　1：比较器 0 上升沿中断允许
位 4：　CP0FIE：比较器 0 下降沿中断允许
　　0：比较器 0 下降沿中断禁止
　　1：比较器 0 下降沿中断允许
位 3-2：　未用。读 ＝00b，写 ＝ 忽略。
位 1-0：　CP0MD1-0：比较器 0 方式选择
　　这两位影响比较器 0 的响应时间和功耗。

方式	CP0MD1	CP0MD0	CP0 下降沿响应时间（典型值）
0	0	0	最快响应时间，最高功耗
1	0	1	—
2	1	0	—
3	1	1	最慢响应时间，最低功耗

SFR 定义 7.3　CPT1CN: 比较器 1 控制寄存器

SFR 页：0x00
SFR 地址：0x9A

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP1	CP1HYP0	CP1HYN1	CP1HYN0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　CP1EN：比较器 1 使能位
0：比较器 1 禁止。
1：比较器 1 使能。

位 6：　CP1OUT：比较器 1 输出状态标志
0：电压值 CP1+ < CP1-。
1：电压值 CP1+ > CP1-。

位 5：　CP1RIF：比较器 1 上升沿中断标志，必须用软件清 0。
0：自该标志位最后一次被清除后，未检测到比较器 1 上升沿。
1：检测到比较器 1 上升沿。

位 4：　CP1FIF：比较器 1 下降沿中断标志，必须用软件清 0
0：自该标志位最后一次被清除后，未检测到比较器 1 下降沿。
1：检测到比较器 1 下降沿。

位 3-2：　CP1HYP1-0：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= 5mV。
10：正向回差电压= 10mV。
11：正向回差电压= 20mV。

位 1-0：　CP1HYN1-0：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= 5mV。
10：负向回差电压= 10mV。
11：负向回差电压= 20mV。

SFR 定义 7.4　CPT1MD: 比较器 1 方式选择寄存器

SFR 页：0x00
SFR 地址：0x9C

R/W	R	R/W	R/W	R	R	R/W	R/W	复位值
保留	—	CP1RIE	CP1FIE	—	—	CP1MD1	CP1MD0	10000010
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0x9C

位 7：　保留。读 ＝1b，必须写 1b。
位 6：　未用。读 ＝0b，写 ＝ 忽略。
位 5：　CP1RIE：比较器 1 上升沿中断允许
　　0：比较器 1 上升沿中断禁止
　　1：比较器 1 上升沿中断允许
位 4：　CP1FIE：比较器 1 下降沿中断允许
　　0：比较器 1 下降沿中断禁止
　　1：比较器 1 下降沿中断允许
位 3-2：　未用。读 ＝00b，写 ＝ 忽略。
位 1-0：　CP1MD1-0：比较器 1 方式选择
　　这两位影响比较器 1 的响应时间和功耗。

方式	CP1MD1	CP1MD0	CP1 下降沿响应时间（典型值）
0	0	0	最快响应时间，最高功耗
1	0	1	—
2	1	0	—
3	1	1	最慢响应时间，最低功耗

7.6 比较器 0 和比较器 1 模拟多路器

C8051F93x-92x 器件内部包含一个模拟输入多路器，将端口 I/O 引脚和内部信号连接到比较器输入。CP0+/CP0-输入 MUX 是比较器 0 的正输入和负输入多路选择器，CP1+/CP1-输入 MUX 是比较器 1 的正输入和负输入多路选择器。

比较器输入多路选择器直接支持容性触摸开关。当容性触感比较输入被正或负多路器选中时，连接到另一个多路器的任一端口 I/O 引脚可被直接连接到一个容性触摸开关，不需要任何额外的外部元件。**容性触感比较**信号为检测容性触摸开关通过片内 Rsense 电阻充电或放电提供合适的参考电平。比较器输出可以连接到定时器 2 或定时器 3，以捕捉感测电容的充电和放电时间。详见“25. 定时器”。

下面的任何一种电压信号都可以被选择为比较器的输入：端口 I/O 引脚、容性触感比较、VDD/DC+电源电压、调整后的数字电源电压（REG1 的输出）、VBAT 电源电压或 GND。比较器的电源电压/2 也可作为一个输入，只有在选择了该设置时，分压电阻才消耗电流。使用 CPT0MX 和 CPT1MX 寄存器（见 SFR 定义 7.5 和 SFR 定义 7.6 的说明）来配置比较器的输入多路器。

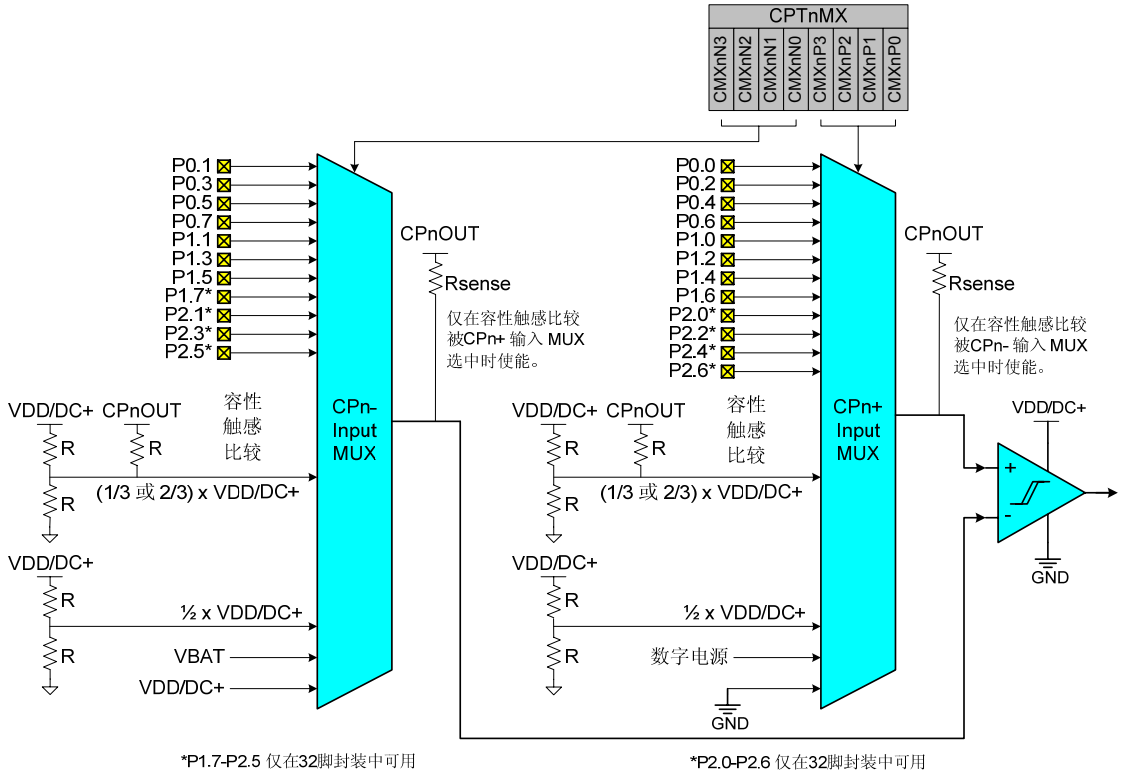


图 7.4 CPn 输入多路器原理框图

有关比较器输入配置的重要注意事项：被选择为比较器输入的端口引脚应被配置为模拟输入并被数字交叉开关跳过。为将一个端口引脚配置为模拟输入，要将寄存器 PnMDIN 中的对应位清 0 并禁止数字驱动器（PnMDOUT = 0 且端口锁存器 = 1）。为强制交叉开关跳过一个端口引脚，应将 PnSKIP 中的对应位置 1。有关端口配置的详细信息见“21. 端口输入/输出”。

SFR 定义 7.5　CPT0MX: 比较器 0 输入通道选择寄存器

SFR 页：0x00
SFR 地址：0x9F

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CMX0N3	CMX0N2	CMX0N1	CMX0N0	CMX0P3	CMX0P2	CMX0P1	CMX0P0	11111111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-4：　CMX0N3-0: 比较器 0 负输入选择
这些位选择比较器 0 的负输入通道。

CMX0N3	CMX0N2	CMX0N1	CMX0N0	负输入
0	0	0	0	P0.1
0	0	0	1	P0.3
0	0	1	0	P0.5
0	0	1	1	P0.7
0	1	0	0	P1.1
0	1	0	1	P1.3
0	1	1	0	P1.5
0	1	1	1	P1.7（仅 C8051F920/30）
1	0	0	0	P2.1（仅 C8051F920/30）
1	0	0	1	P2.3（仅 C8051F920/30）
1	0	1	0	P2.5（仅 C8051F920/30）
1	0	1	1	保留
1	1	0	0	容性触感比较
1	1	0	1	VDD/DC+/2
1	1	1	0	数字电源电压
1	1	1	1	地

位 3-0：　CMX0P3-0: 比较器 0 正输入选择
这些位选择比较器 0 的正输入通道。

CMX0P3	CMX0P2	CMX0P1	CMX0P0	正输入
0	0	0	0	P0.0
0	0	0	1	P0.2
0	0	1	0	P0.4
0	0	1	1	P0.6
0	1	0	0	P1.0
0	1	0	1	P1.2
0	1	1	0	P1.4
0	1	1	1	P1.6
1	0	0	0	P2.0（仅 C8051F920/30）
1	0	0	1	P2.2（仅 C8051F920/30）
1	0	1	0	P2.4（仅 C8051F920/30）
1	0	1	1	P2.6（仅 C8051F920/30）
1	1	0	0	容性触感比较
1	1	0	1	VDD/DC+/2
1	1	1	0	VBAT 电源电压
1	1	1	1	VDD/DC+电源电压

SFR 定义 7.6　CPT1MX: 比较器 1 输入通道选择寄存器

SFR 页：0x00
SFR 地址：0x9E

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CMX1N3	CMX1N2	CMX1N1	CMX1N0	CMX1P3	CMX1P2	CMX1P1	CMX1P0	11111111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-4：　CMX1N3-0: 比较器 1 负输入选择
这些位选择比较器的 1 负输入通道。

CMX1N3	CMX1N2	CMX1N1	CMX1N0	负输入
0	0	0	0	P0.1
0	0	0	1	P0.3
0	0	1	0	P0.5
0	0	1	1	P0.7
0	1	0	0	P1.1
0	1	0	1	P1.3
0	1	1	0	P1.5
0	1	1	1	P1.7（仅 C8051F920/30）
1	0	0	0	P2.1（仅 C8051F920/30）
1	0	0	1	P2.3（仅 C8051F920/30）
1	0	1	0	P2.5（仅 C8051F920/30）
1	0	1	1	保留
1	1	0	0	容性触感比较
1	1	0	1	VDD/DC+/2
1	1	1	0	数字电源电压
1	1	1	1	地

位 3-0：　CMX1P3-0: 比较器 1 正输入选择
这些位选择比较器 1 的正输入通道。

CMX1P3	CMX1P2	CMX1P1	CMX1P0	正输入
0	0	0	0	P0.0
0	0	0	1	P0.2
0	0	1	0	P0.4
0	0	1	1	P0.6
0	1	0	0	P1.0
0	1	0	1	P1.2
0	1	1	0	P1.4
0	1	1	1	P1.6
1	0	0	0	P2.0（仅 C8051F920/30）
1	0	0	1	P2.2（仅 C8051F920/30）
1	0	1	0	P2.4（仅 C8051F920/30）
1	0	1	1	P2.6（仅 C8051F920/30）
1	1	0	0	容性触感比较
1	1	0	1	VDD/DC+/2
1	1	1	0	VBAT 电源电压
1	1	1	1	VDD/DC+电源电压

8. 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。C8051F93x-C8051F92x 的外设是标准 8051 的所有外设的超集。CIP-51 片内还包含与模拟和数字子系统直接接口的调试硬件，在一片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 8.1 的原理框图）。CIP-51 具有下列特性：

- 与 MCS-51 指令集完全兼容
- 峰值速度为 25MIPS（25MHz 时钟）
- 0 ~ 25 MHz 时钟频率
- 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

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。

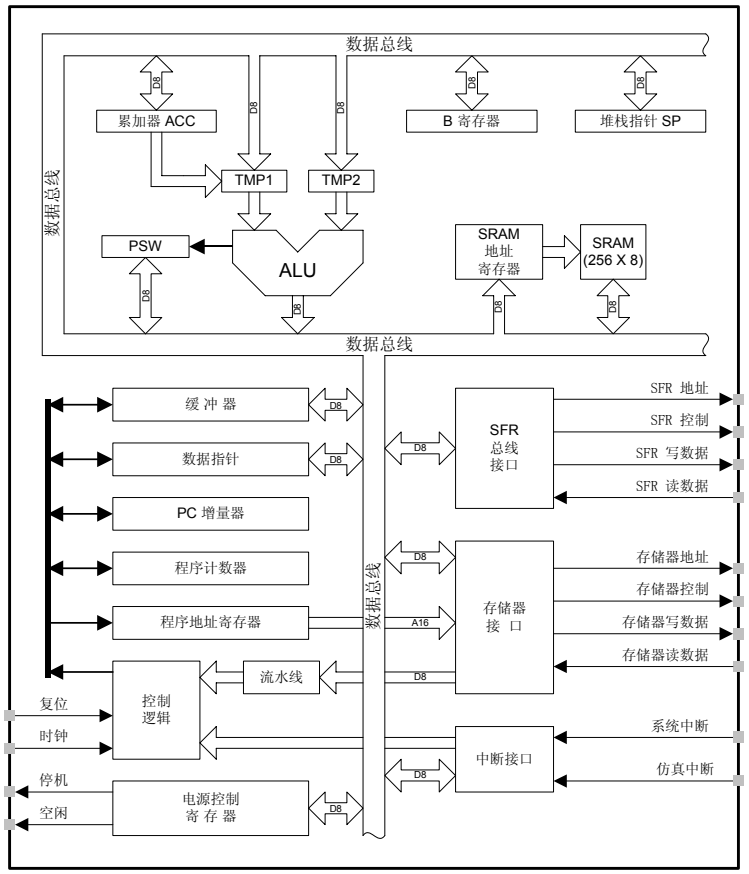


图 8.1 CIP-51 原理框图

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。下表列出了各种指令执行时间所对应的指令条数。

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

编程和调试支持

对 FLASH 程序存储器的在系统编程和与片内调试支持逻辑的通信是通过 Silicon Labs 2 线开发接口（C2）实现的。

片内调试逻辑支持全速的在系统调试，允许设置硬件断点，支持开始、停止和单步执行（包括中断服务程序）命令，支持检查程序调用堆栈及读/写寄存器和存储器。这种片内调试方法完全是非侵入式的，不需要占用 RAM、堆栈、定时器或其它片内资源。有关 C2 接口的详细信息见“27. C2 接口”。

CIP-51 有 Silicon Labs 和第三方供应商的开发工具支持。Silicon Labs 提供一个集成开发环境（IDE），包括编辑器、调试器和编程器。IDE 的调试器和编程器与 CIP-51 之间通过 C2 接口连接，提供快速和有效的在系统编程和调试。也有第三方的宏汇编器和 C 编译器可用。

8.1 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同类的 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。

8.1.1 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 1 个。表 8.1 给出了所有 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。

表 8.1 CIP-51 指令集一览表 [1]

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 内容加到累加器	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	累加器减去直接寻址字节(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	数据指针加 1	1	1
MUL AB	累加器与寄存器 B 相乘	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	直接寻址字节“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	累加器循环左移	1	1
RLC A	带进位的累加器循环左移	1	1

助记符	功能说明	字节数	时钟周期数
RR A	累加器循环右移	1	1
RRC A	带进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
数据传送类指令			
MOV A,Rn	寄存器传送到累加器	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	直接寻址字节传送到间址 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 DPTR	3	3
MOVC A,@A+DPTR	相对于 DPTR 的代码字节传送到累加器	1	3
MOVC A,@A+PC	相对于 PC 的代码字节传送到累加器	1	3
MOVX A,@Ri	外部 RAM(8 位地址)传送到累加器	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)传送到累加器	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

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（相对地址）	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累加器，不相等则转移	3	3/4
CJNE A,#data,rel	比较立即数与累加器，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 或 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 位常数

#data16 – 16 位常数

bit – 数据 RAM 或 SFR 中的直接寻址位

addr11 – ACALL 和 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 和 LJMP 使用的 16 位目的地址。目的地址可以是 64KB 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。

8.2 CIP-51 寄存器说明

下面对与 CIP-51 系统控制器操作有关的 SFR 进行说明。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。

SFR 定义 8.1　DPL：数据指针低字节

SFR 页：所有页
SFR 地址：0x82

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	

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 或 FLASH 存储器。

SFR 定义 8.2　DPH：数据指针高字节

SFR 页：所有页
SFR 地址：0x83

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	

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 或 FLASH 存储器。

SFR 定义 8.3　SP：堆栈指针

SFR 页：所有页
SFR 地址：0x81

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000111
位7	位6	位5	位4	位3	位2	位1	位0	

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。

SFR 定义 8.4　ACC：累加器

SFR 页：所有页
SFR 地址：0xE0　可位寻址

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	

位 7-0：　ACC：累加器
该寄存器为算术操作用的累加器。

SFR 定义 8.5　B：B 寄存器

SFR 页：所有页
SFR 地址：0xF0　可位寻址

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	

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。

SFR 定义 8.6　PSW：程序状态字

SFR 页：所有页
SFR 地址：0xD0　　可位寻址

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	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	

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。
该两位选择访问寄存器时所使用的寄存器区。

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。
该位在下列情况下被置 1：
- ADD、ADDC 或 SUBB 指令引起符号位变化溢出。
- MUL 指令引起溢出（结果大于 255）。
- DIV 指令的除数为 0。

ADD、ADDC、SUBB、MUL 和 DIV 指令的其它情况使该位清 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。
累加器中 8 个位的和为奇数时该位被置 1，为偶数时被清 0。

9. 存储器组织

CIP-51 系统控制器的存储器组织与标准 8051 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。C8051F93x-C8051F92x 的存储器组织如图 9.1 所示。

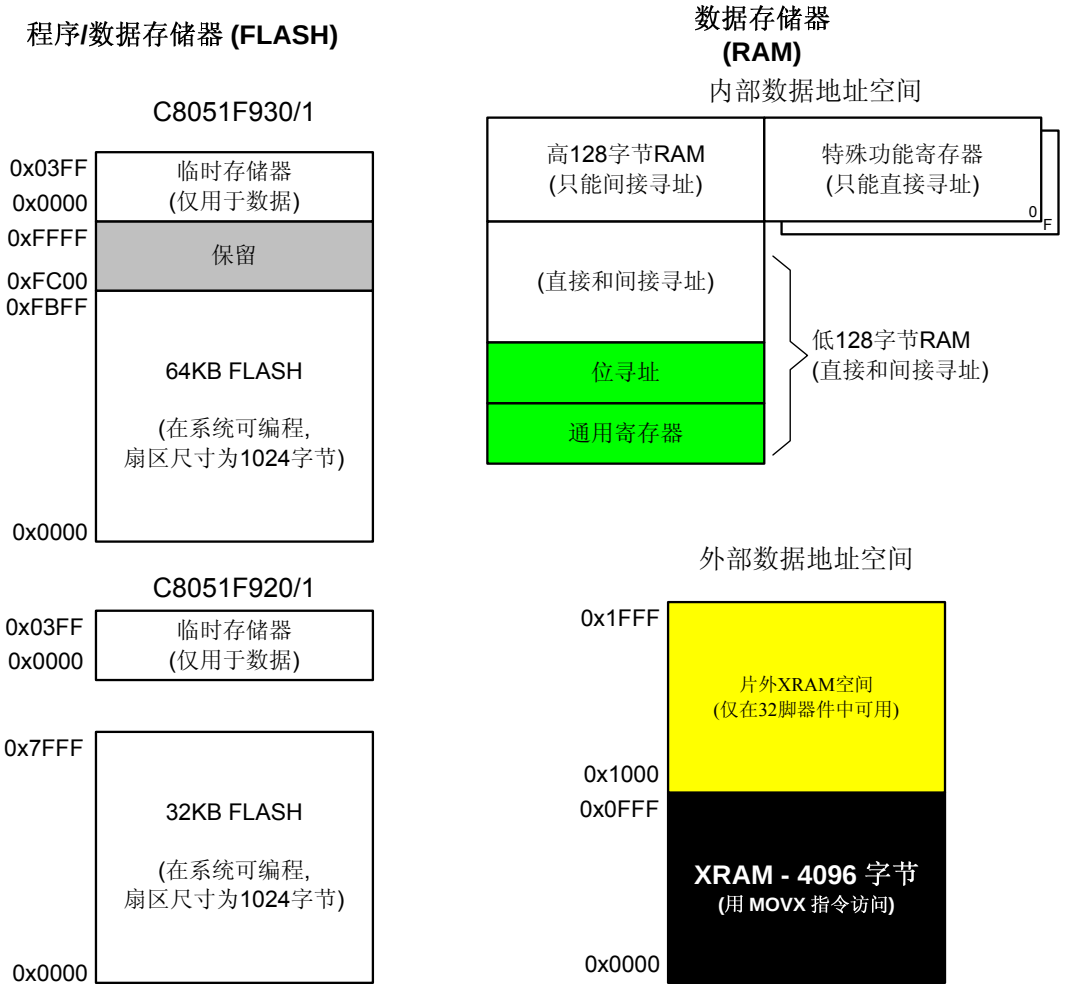


图 9.1 C8051F93x-C8051F92x 存储器组织图

9.1 程序存储器

CIP-51 有 64KB 的程序存储器空间。C8051F93x-C8051F92x 在这个程序存储器空间中实现了 64KB（C8051F930/1）或 32KB（C8051F920/1）的可在系统重编程的 FLASH 存储器，组织在一个地址范围为 0x0000 ~ 0xFBFF（C8051F930/1）或 0x0000 ~ 0x7FFF（C8051F920/1）的连续存储块内。地址 0xFBFF（C8051F930/1）或 0x7FFF（C8051F920/1）为器件的安全锁定字节。锁定字节以上的任何地址均被保留

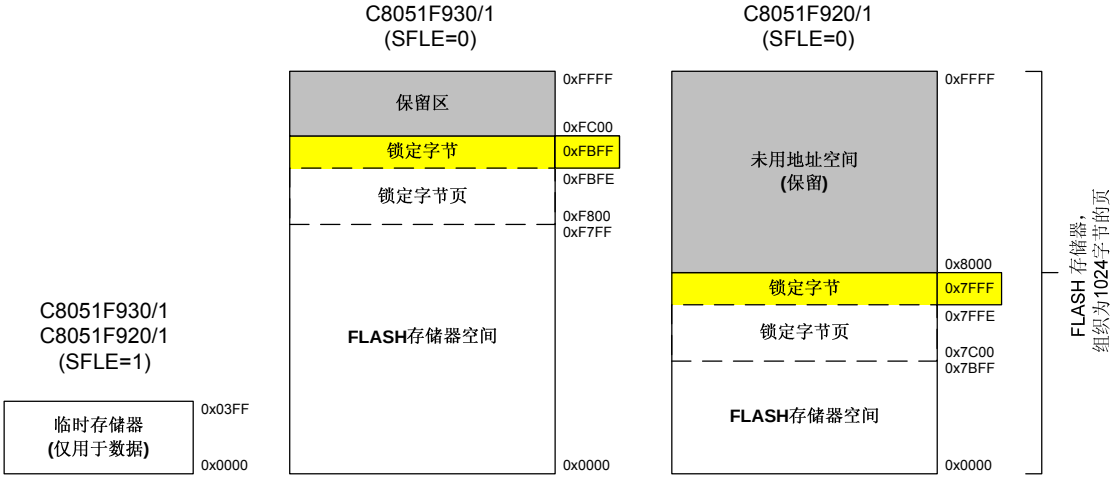


图 9.2 FLASH 程序存储器组织图

9.1.1　MOVX 指令和程序存储器

8051 器件的 MOVX 指令通常用于访问外部数据存储器。在 C8051F93x-C8051F92x 器件中，MOVX 指令一般用于读和写片内 XRAM，但也可以被配置为写和擦除片内 FLASH 存储器空间。MOVC 指令总是用于读 FLASH 存储器，MOVX 指令用于擦除和写 FLASH 存储器。这一 FLASH 访问特性为 C8051F93x-C8051F92x 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，详见“13. FLASH 存储器”。

9.2 数据存储器

C8051F93x-C8051F92x 器件内部有 4352 字节的 RAM 数据存储器。该存储器中有 256 字节被映射到 8051 的内部 RAM 空间，另外 4096 字节是片内“外部”存储器（XRAM）。图 9.1 给出了数据存储器组织图。

9.2.1 内部 RAM

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的低 128 字节。从 0x00 到 0x1F 为 4 个通用寄存器区，每个区有 8 个 8 位寄存器。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 9.1 给出了 C8051F93x-C8051F92x 数据存储器组织的示意图。

9.2.1.1 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 8 位寄存器，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 SFR 定义 8.6 中 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 作为间址寄存器。

9.2.1.2 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-51™ 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，  22 . 3 h
```

将 0x13 中的布尔值（字节地址 0x22 中的位 3）传送到进位标志。

9.2.1.3 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。

9.2.2 外部 RAM

C8051F93x-C8051F92x 器件内部有映射到外部数据存储器空间的 4096 字节 RAM。可以用外部传送指令（MOVX）和数据指针（DPTR）访问这些地址单元，或者用 MOVX 间接寻址方式（例如 @R1）与 EMI0CN 寄存器结合访问。片外存储器或存储器映射的器件可以被映射到外部存储器地址空间，用外部存储器接口访问。详见“10. 外部数据存储器接口和片内 XRAM”。

10. 外部数据存储器接口和片内 XRAM

C8051F93x-C8051F92x 器件内部有映射到外部数据存储器空间的片内 RAM（XRAM）和一个可用于访问连接到 GPIO 端口的外部存储器及存储器映射器件的外部数据存储器接口（EMIF）。外部存储器空间可以用外部传送指令（MOVX）和数据指针（DPTR）中指定的目标地址访问，或者用位于 R0 或 R1 中的目标地址低字节和位于外部存储器接口控制寄存器（EMI0CN，见 SFR 定义 10.1）中的目标地址高字节访问。

当使用 MOVX 指令访问片内 RAM 时，不需要进行额外的初始化，MOVX 指令的执行时间见 CIP-51 一章的指令一览表。当使用 MOVX 指令访问片外 RAM 或存储器映射的器件时，端口 I/O 和 EMIF 都应被配置为与外部器件通信（见 10.2 节），MOVX 指令时序由外部存储器接口时序控制寄存器（EMI0TC，见“10.6 外部存储器接口时序”）中的编程值决定。

注意：MOVX 写操作可被配置为写 FLASH 存储器，而非 XRAM。详见“13. FLASH 存储器”。缺省状态下 MOVX 指令访问 XRAM。

10.1 访问 XRAM

用 MOVX 指令访问 XRAM 存储器空间。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 16 位寄存器中含有待读或写的 XRAM 单元的实际地址。第二种方法使用 R0 或 R1，与 EMI0CN 寄存器一起形成实际 XRAM 地址。下面举例说明这两种方法。

10.1.1 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。下面的指令将位于地址 0x1234 的字节值读入累加器 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。

10.1.2 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。下面的指令将位于地址 0x1234 的字节值读入累加器 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, @R0              ; 将地址 0x1234 的内容装入累加器 A
```

10.2 配置访问片外空间的外部存储器接口

配置访问片外存储器空间的外部存储器接口的过程包括下面 4 个步骤：

1. 配置相应端口引脚的输出方式为推挽或漏极开路（最常用的是推挽方式），并在交叉开关中跳过这些引脚（如果需要的话）。要确定外部存储器接口使用哪些引脚，见“21. 端口输入/输出”。
2. 配置对应 EMIF 引脚的端口锁存器为休眠态（通常将它们设置为逻辑 1）。
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。

下面将对上述 4 个步骤作出详细说明。配置选择位在 EMI0CF 寄存器中，见 SFR 定义 10.2。

10.3 外部存储器接口的端口输入/输出配置

外部存储器接口用于访问片外存储器空间，EMIF 与 GPIO 端口锁存器共享相关的端口引脚。交叉开关应被配置为不对相关引脚分配任何数字信号。在大多数配置中，/RD、/WR 和 ALE 需要在交叉开关中被跳过，以保证它们由端口锁存器控制。要确定外部存储器接口使用哪些引脚，见“21. 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，端口锁存器重新恢复对端口引脚的控制。**端口锁存器应被显式地配置为使外部存储器接口引脚处于休眠状态（不使用时），通常将它们设置为逻辑 1。**

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]）。对于作为输出的端口引脚（例如，写操作期间的 Data[7:0]），外部存储器接口不自动使能输出驱动器。双向和只作为输出的引脚的输出方式（无论引脚被配置为漏极开路或是推挽方式）应被配置为所希望的方式。

当端口引脚由 GPIO 锁存器控制时，其输出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。在大多数情况下，所有 EMIF 引脚的输出方式都应被配置为推挽方式。

10.4 复用的外部存储器接口

对于复用的外部存储器接口，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]。对于大多数具有 8 位接口的器件，如果外部存储器接口用于 8 位不分块方式，则不用的高位地址可以被作为 GPIO 使用。如果外部存储器接口用于 8 位分块方式或 16 位方式，则地址引脚被高 4 位地址位驱动，这些引脚不能作为 GPIO 使用。

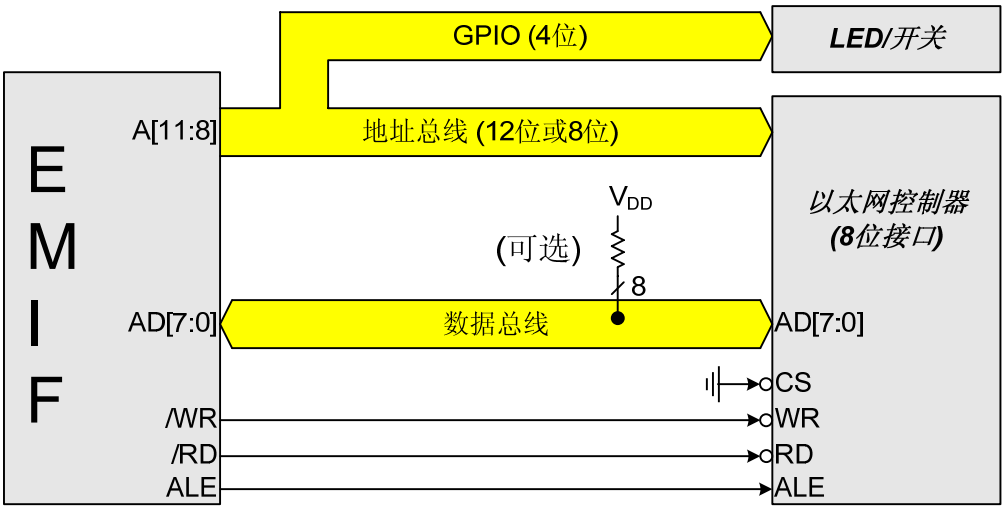


图 10.1　复用方式配置示例

很多器件具有从方式并行存储器接口，如 SRAM 芯片，它们只支持非复用存储器总线。当与这样的器件接口时，要用一个外部锁存器（74HC373 或相同功能的逻辑门）在存储器周期的后半周期（此时地址/数据总线上为数据）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自动驱动。图 10.2 给出了一个复用到非复用转换的 SRAM 接口的例子。

从该例可以看出，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，并且与锁存器的输入不再相关。在第二阶段稍后，当/RD 或/WR 有效时，数据总线控制 AD[7:0]端口的状态。

更详细的时序信息见“10.6 外部存储器接口时序”。

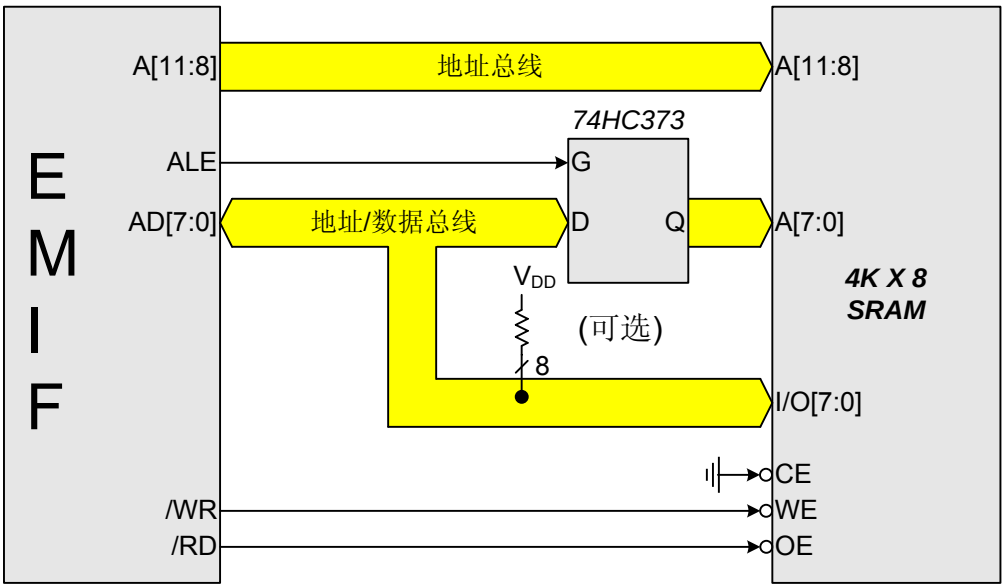


图 10.2　复用到非复用方式转换配置示例

10.5 外部存储器接口工作模式

可以用 EMI0CF 寄存器（SFR 定义 10.2）中的 EMIF 模式选择位将外部数据存储器空间配置为图 10.3 所示的四种工作模式之一。下面简要介绍这些模式。有关不同模式的时序信息见“10.6 外部存储器接口时序”。

10.5.1 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回绕，并总是指向片内 XRAM。例如，如果对整个地址空间进行连续写入，数据指针在每次写操作后加 1，则在对片内 XRAM 的最后一个字节执行了写操作后，写指针总是指向片内 XRAM 的第一个字节。

- 8 位 MOVX 操作使用 EMI0CN 的内容作为有效地址的高字节，由 R0 或 R1 给出有效地址的低字节。
- 16 位 MOVX 操作使用 16 位寄存器 DPTR 的内容确定有效地址。

10.5.2 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。

- 有效地址低于内部 XRAM 尺寸边界时将访问片内 XRAM 空间。
- 有效地址高于内部 XRAM 尺寸边界时将访问片外 XRAM 空间。
- 8 位 MOVX 操作使用 EMI0CN 的内容确定是访问片内还是片外存储器。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 4 位 A[11:8]。这就允许用户通过直接设置端口锁存器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。地址总线的低 8 位 A[7:0]被驱动，由 R0 或 R1 给出。

- 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[11:0]的全部 12 位都被驱动。

10.5.3 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。

- 有效地址低于内部 XRAM 尺寸边界时将访问片内 XRAM 空间。
- 有效地址高于内部 XRAM 尺寸边界时将访问片外空间。
- 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 4 位 A[11:8]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[11:0]的全部 12 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，在访问片外存储器时地址总线 A[11:0]的全部 12 位都被驱动。

10.5.4 只用外部存储器

当 EMI0CF.[3:2]被设置为‘11’时，所有 MOVX 指令都将访问片外空间。片内 XRAM 对 CPU 为不可见。该方式在访问从 0x0000 开始到内部 XRAM 尺寸边界之间的片外存储器时有用。

- 8 位 MOVX 操作忽略 EMI0CN 的内容。高地址位 A[11:8]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[11:0]。在访问片外存储器时地址总线 A[11:0]的全部 12 位都被驱动。

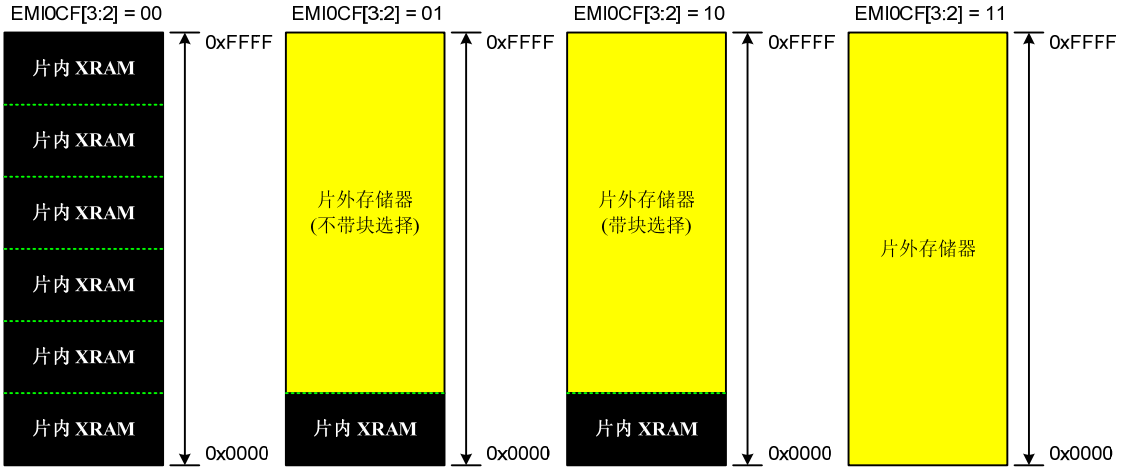


图 10.3　EMIF 工作模式

10.6　EMIF 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 选通脉冲的宽度以及复用方式下 ALE 脉冲的宽度都可以通过 EMI0TC（见 SFR 定义 10.3）和 EMI0CF[1:0]编程，编程单位为 SYSCLK 周期。

片外 MOVX 指令的时序可以通过将 EMI0TC 寄存器中定义的时序参数加上 4 个 SYSCLK 周期来计算。在非复用方式，一次片外 XRAM 操作的最小执行时间为 5 个 SYSCLK 周期（用于/RD 或/WR 脉冲的 1 个 SYSCLK + 4 个 SYSCLK）。对于复用方式，地址锁存使能信号至少需要 2 个附加的 SYSCLK 周期。因此，在复用方式，一次片外 XRAM 操作的最小执行时间为 7 个 SYSCLK 周期（用于 ALE 的 2 个 SYSCLK ＋ 用于/RD 或/WR 脉冲的 1 个 SYSCLK ＋ 4 个 SYSCLK）。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 10.1 列出了外部存储器接口的 AC 参数，图 10.4～ 图 10.6 给出了对应不同外部存储器接口模式和 MOVX 操作的时序图。要确定哪些端口引脚被映射到 ADDR[11:8]、AD[7:0]、ALE、/RD 和/WR 信号，见“21. 端口输入/输出”。

10.7 EMIF 特殊功能寄存器

EMIF 使用的特殊功能寄存器有 EMI0CN、EMI0CF 和 EMI0TC。下面对这些寄存器进行说明。

SFR 定义 10.1 EMI0CN: 外部存储器接口控制

SFR 页：0x00
SFR 地址：0xAA

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	—	PGSEL4	PGSEL3	PGSEL2	PGSEL1	PGSEL0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-5： 未用。读 = 000b，写 = 忽略。

位 7-0： PGSEL[4:0]：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。由于该寄存器的最高几位（未使用）总是为 0，所以 PGSEL 决定访问 XRAM 的哪个页。当 PGSEL 的最高位被置 1 且 EMIF 被配置为两种分片模式之一时，8 位 MOVX 指令访问片外存储器。

例如：
如果 EMI0CN = 0x01，则访问片内存储器地址 0x0100 ~ 0x01FF。
如果 EMI0CN = 0x0F，则访问片内存储器地址 0x0F00 ~ 0x0FFF。
如果 EMI0CN = 0x11，则访问片外存储器地址 0x0100 ~ 0x01FF。
如果 EMI0CN = 0x1F，则访问片外存储器地址 0x0F00 ~ 0x0FFF。

SFR 定义 10.2　EMI0CF: 外部存储器接口配置寄存器

SFR 页：0x00
SFR 地址：0xAB

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	—	—	EMD1	EMD0	EALE1	EALE0	00000011
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-4：　未用。读 = 000b，写 = 忽略。

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（见“10.5 外部存储器接口工作方式”）。

00：只用内部存储器。

01：不带块选择的分片方式。

10：带块选择的分片方式。

11：只用外部存储器。

位 1-0：　EALE1-0：ALE 脉冲宽度选择位

选择 ALE 脉冲宽度。

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。

SFR 定义 10.3　EMI0TC：外部存储器时序控制寄存器

SFR 页：0x00
SFR 地址：0xAF

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	

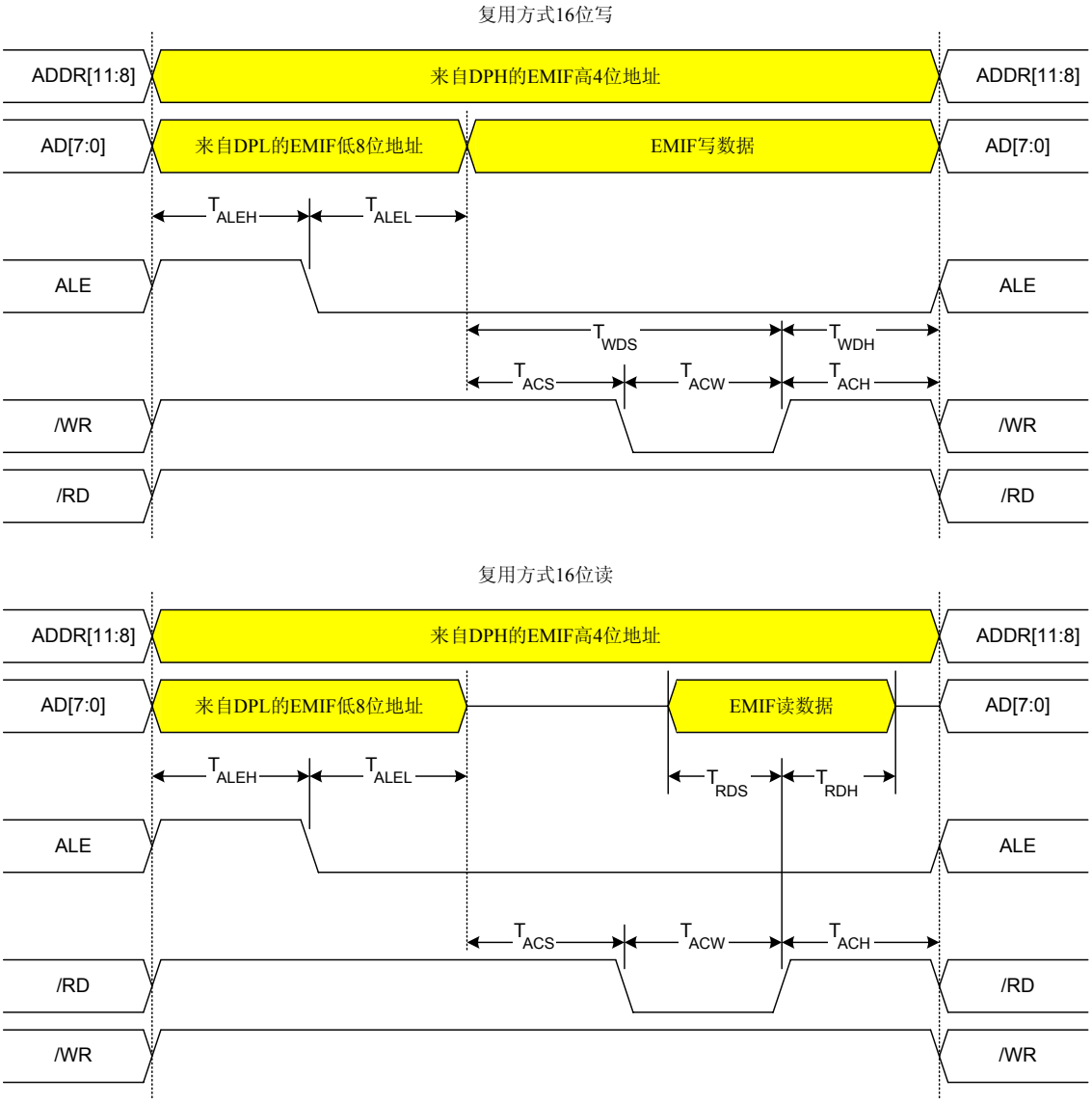
位 7-6：　EAS1-0：EMIF 地址建立时间选择位。
控制时序参数 $\mathbf{T_{ACS}}$。
00：地址建立时间 ＝0 个 SYSCLK 周期。
01：地址建立时间 ＝1 个 SYSCLK 周期。
10：地址建立时间 ＝2 个 SYSCLK 周期。
11：地址建立时间 ＝3 个 SYSCLK 周期。

位 5-2：　EWR3-0：/WR 和/RD 脉冲宽度选择位。
控制时序参数 $\mathbf{T_{ACW}}$。
0000：/WR 和/RD 脉冲宽度 ＝1 个 SYSCLK 周期。
0001：/WR 和/RD 脉冲宽度 ＝2 个 SYSCLK 周期。
0010：/WR 和/RD 脉冲宽度 ＝3 个 SYSCLK 周期。
0011：/WR 和/RD 脉冲宽度 ＝4 个 SYSCLK 周期。
0100：/WR 和/RD 脉冲宽度 ＝5 个 SYSCLK 周期。
0101：/WR 和/RD 脉冲宽度 ＝6 个 SYSCLK 周期。
0110：/WR 和/RD 脉冲宽度 ＝7 个 SYSCLK 周期。
0111：/WR 和/RD 脉冲宽度 ＝8 个 SYSCLK 周期。
1000：/WR 和/RD 脉冲宽度 ＝9 个 SYSCLK 周期。
1001：/WR 和/RD 脉冲宽度 ＝10 个 SYSCLK 周期。
1010：/WR 和/RD 脉冲宽度 ＝11 个 SYSCLK 周期。
1011：/WR 和/RD 脉冲宽度 ＝12 个 SYSCLK 周期。
1100：/WR 和/RD 脉冲宽度 ＝13 个 SYSCLK 周期。
1101：/WR 和/RD 脉冲宽度 ＝14 个 SYSCLK 周期。
1110：/WR 和/RD 脉冲宽度 ＝15 个 SYSCLK 周期。
1111：/WR 和/RD 脉冲宽度 ＝16 个 SYSCLK 周期。

位 1-0：　EAH1-0：EMIF 地址保持时间选择位。
控制时序参数 $\mathbf{T_{ACH}}$。
00：地址保持时间 ＝0 个 SYSCLK 周期。
01：地址保持时间 ＝1 个 SYSCLK 周期。
10：地址保持时间 ＝2 个 SYSCLK 周期。
11：地址保持时间 ＝3 个 SYSCLK 周期。

10.8　EMIF 时序图

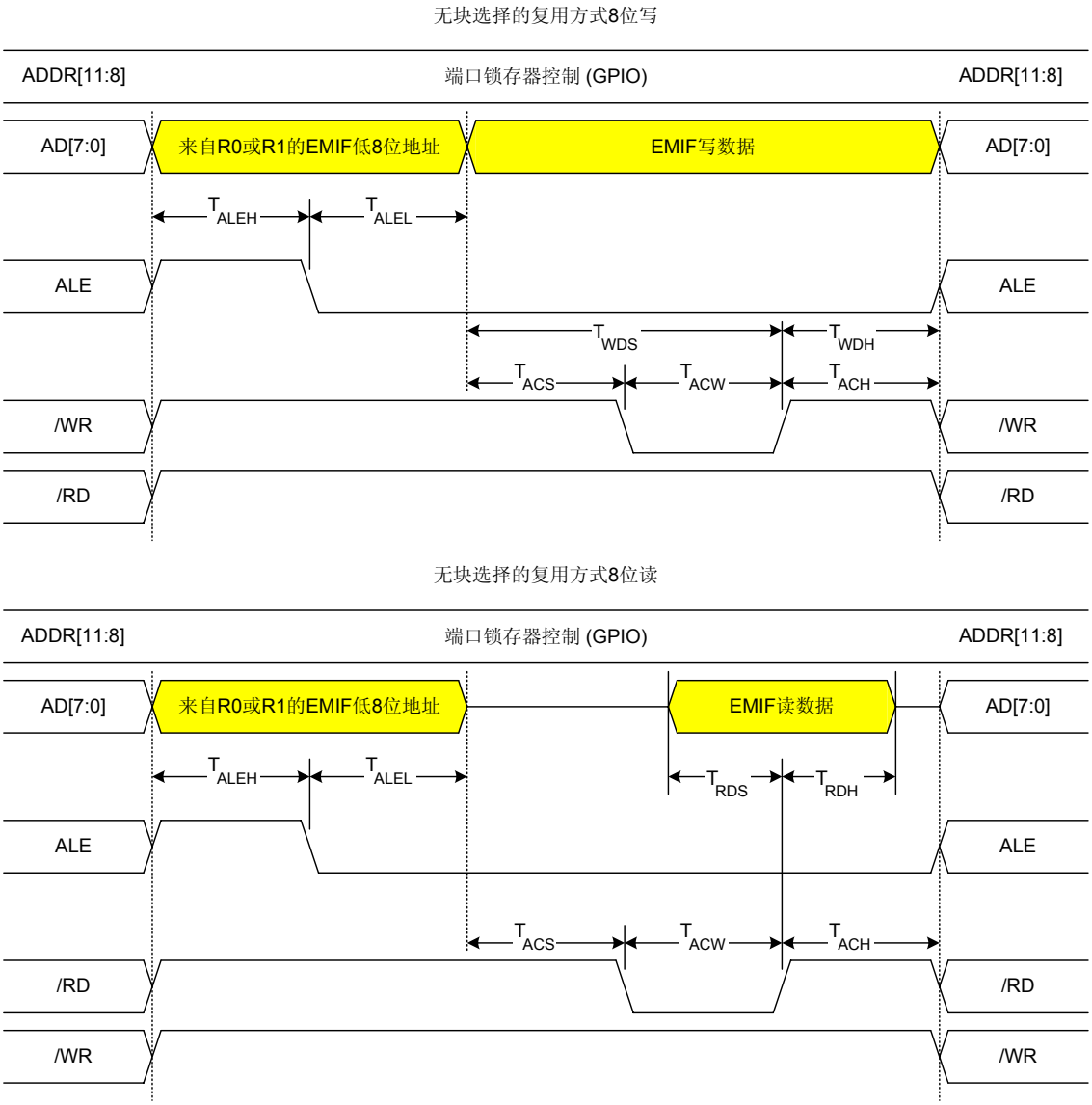
10.8.1 复用方式 16 位 MOVX：EMI0CF[3:2] = 01、10 或 11



注：要确定哪些端口引脚被映射到 ADDR[11:8]、AD[7:0]、ALE、/RD和/WR信号，见端口输入/输出一章。

图 10.4　复用方式 16 位 MOVX 时序

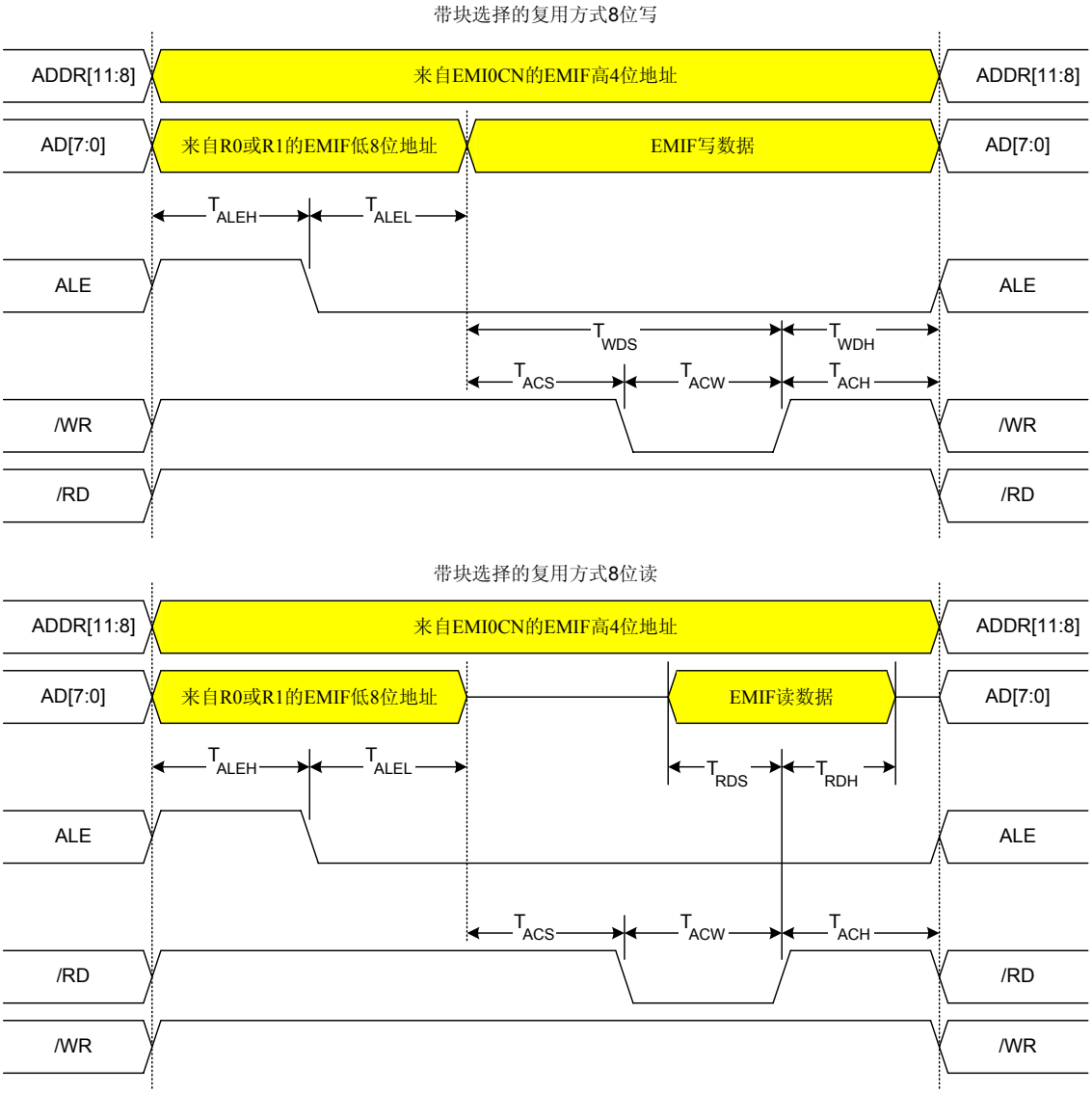
10.8.2 无块选择的复用方式 8 位 MOVX：EMI0CF[3:2] = 01 或 11



注：要确定哪些端口引脚被映射到 ADDR[11:8]、AD[7:0]、ALE、/RD和/WR信号，见端口输入/输出一章。

图 10.5　无块选择的复用方式 8 位 MOVX 时序

10.8.3 带块选择的复用方式 8 位 MOVX：EMI0CF[3:2] = 10



注：要确定哪些端口引脚被映射到 ADDR[11:8]、AD[7:0]、ALE、/RD和/WR信号，见端口输入/输出一章。

图 10.6　带块选择的复用方式 8 位 MOVX 时序

表 10.1 外部存储器接口的 AC 参数

参 数	说 明	最小值*	最大值*	单 位
T_{ACS}	地址/控制信号建立时间	0	3× T_{SYSCLK}	ns
T_{ACW}	地址/控制信号脉冲宽度	1× T_{SYSCLK}	16× T_{SYSCLK}	ns
T_{ACH}	地址/控制信号保持时间	0	3× T_{SYSCLK}	ns
T_{ALEH}	地址锁存使能信号高电平时间	1× T_{SYSCLK}	4× T_{SYSCLK}	ns
T_{ALEL}	地址锁存使能信号低电平时间	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
*注：T_{SYSCLK} 等于器件系统时钟（SYSCLK）的一个周期。				

11. 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 C8051F93x-C8051F92x 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 C8051F93x-C8051F92x 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 11.1 和表 11.2 列出了 C8051F93x-C8051F92x 器件中实现的全部 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、SCON0、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。有关表 11.3 中列出的每个寄存器的详细说明请参见本数据表的相关部分。

表 11.1 特殊功能寄存器（SFR）存储器映象（页 0）

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	PCA0CPL4	PCA0CPH4	VDM0CN
F0	B	P0MDIN	P1MDIN	P2MDIN	SMB0ADR	SMB0ADM	EIP1	EIP2
E8	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	RSTSRC
E0	ACC	XBR0	XBR1	XBR2	IT01CF		EIE1	EIE2
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	PCA0PWM
D0	PSW	REF0CN	PCA0CPL5	PCA0CPH5	P0SKIP	P1SKIP	P2SKIP	P0MAT
C8	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	PCA0CPM5	P1MAT
C0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	P0MASK
B8	IP	IREF0CN	ADC0AC	ADC0MX	ADC0CF	ADC0L	ADC0H	P1MASK
B0	SPI1CN	OSCXCN	OSCICN	OSCICL		PMU0CF	FLSCL	FLKEY
A8	IE	CLKSEL	EMI0CN	EMI0CF	RTC0ADR	RTC0DAT	RTC0KEY	EMI0TC
A0	P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	SFRPAGE
98	SCON0	SBUF0	CPT1CN	CPT0CN	CPT1MD	CPT0MD	CPT1MX	CPT0MX
90	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H	DC0CF	DC0CN
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH	SPI1CFG	SPI1CKR	SPI1DAT	PCON
	0(8) 可位寻址	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

11.1 SFR 分页

CIP-51 实现了 SFR 分页机制，允许器件能在 0x80 ~ 0xFF 这个地址空间访问的 SFR 数大于 128。默认状态下，所有的 SFR 访问都指向页 0，允许访问表 11.1 中列出的寄存器。在器件初始化期间，可能也需要访问位于 SFR 页 F 的某些寄存器。表 11.2 列出了可以从 SFR 页 F 访问的 SFR。有些 SFR 从页 0 和页 F 都可以访问，包括 SFRPAGE 寄存器。只能从页 F 访问的 SFR 用粗体表示。

从页 F 访问 SFR 的步骤如下：

1. 保存当前中断状态（EA_SAVE = EA）。
2. 禁止中断（EA = 0）。
3. 设置 SFRPAGE = 0x0F
4. 访问位于 SFR 页 F 的 SFR。
5. 设置 SFRPAGE = 0x00
6. 恢复中断状态（EA = EA_SAVE）。

表 11.2 特殊功能寄存器（SFR）存储器映象（页 F）

F8								
F0	B						EIP1	EIP2
E8								
E0	ACC						EIE1	EIE2
D8								
D0	PSW							
C8								
C0								
B8			**ADC0PWR**			**ADC0TK**		
B0								
A8	IE	CLKSEL						
A0	P2				**P0DRV**	**P1DRV**	**P2DRV**	SFRPAGE
98								
90	P1	**CRC0DAT**	**CRC0CN**	**CRC0IN**	**CRC0FLIP**		**CRC0AUTO**	**CRC0CNT**
88								
80	P0	SP	DPL	DPH		**TOFFL**	**TOFFH**	PCON
	0(8) 可位寻址	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

SFR 定义 1.1　SFRPAGE: SFR 页寄存器

SFR 页：所有页
SFR 地址：0xA7

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	

位 7-0：　SFR 页
指明读、写或修改特殊功能寄存器时使用的 SFR 页。

表 11.3 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	SFR 页	说　明	页码
ACC	0xE0	所有页	累加器	
ADC0AC	0xBA	0x00	ADC0 累加器配置寄存器	
ADC0CF	0xBC	0x00	ADC0 配置寄存器	
ADC0CN	0xE8	0x00	ADC0 控制寄存器	
ADC0GTH	0xC4	0x00	ADC0 下限（大于）数据字（高字节）	
ADC0GTL	0xC3	0x00	ADC0 下限（大于）数据字（低字节）	
ADC0H	0xBE	0x00	ADC0 数据字（高字节）	
ADC0L	0xBD	0x00	ADC0 数据字（低字节）	
ADC0LTH	0xC6	0x00	ADC0 上限（小于）数据字（高字节）	
ADC0LTL	0xC5	0x00	ADC0 上限（小于）数据字（低字节）	
ADC0MX	0xBB	0x00	AMUX0 通道选择寄存器	
ADC0PWR	0xBA	0x0F	ADC0 突发模式上电时间寄存器	
ADC0TK	0xBD	0x0F	ADC0 跟踪控制寄存器	
B	0xF0	所有页	B 寄存器	
CKCON	0x8E	0x00	时钟控制寄存器	
CLKSEL	0xA9	所有页	系统时钟选择寄存器	
CPT0CN	0x9B	0x00	比较器 0 控制寄存器	
CPT0MD	0x9D	0x00	比较器 0 方式选择寄存器	
CPT0MX	0x9F	0x00	比较器 0 MUX 选择寄存器	
CPT1CN	0x9A	0x00	比较器 1 控制寄存器	
CPT1MD	0x9C	0x00	比较器 1 方式选择寄存器	
CPT1MX	0x9E	0x00	比较器 1MUX 选择寄存器	
CRC0AUTO	0x96	0x0F	CRC0 自动控制寄存器	
CRC0CN	0x92	0x0F	CRC0 控制寄存器	
CRC0CNT	0x97	0x0F	CRC0 自动 FLASH 扇区计数	
CRC0DAT	0x91	0x0F	CRC0 数据寄存器	
CRC0FLIP	0x94	0x0F	CRC0 位反转寄存器	
CRC0IN	0x93	0x0F	CRC0 数据输入寄存器	

表 11.3 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	SFR 页	说　明	页码
DPH	0x83	所有页	数据指针（高字节）	
DPL	0x82	所有页	数据指针（低字节）	
EIE1	0xE6	所有页	扩展中断允许 1	
EIE2	0xE7	所有页	扩展中断允许 2	
EIP1	0xF6	0x00	扩展中断优先级 1	
EIP2	0xF7	0x00	扩展中断优先级 2	
EMI0CF	0xAB	0x00	外部存储器接口配置寄存器	
EMI0CN	0xAA	0x00	外部存储器接口控制寄存器	
EMI0TC	0xAF	0x00	外部存储器接口时序控制寄存器	
FLKEY	0xB7	0x00	FLASH 锁定和关键码寄存器	
FLSCL	0xB6	0x00	FLASH 存储器定时预分频器	
IE	0xA8	所有页	中断允许寄存器	
IP	0xB8	0x00	中断优先级控制寄存器	
IREF0CN	0xB9	0x00	电流基准 IREF 控制寄存器	
IT01CF	0xE4	0x00	INT0/INT1 配置寄存器	
OSCICL	0xB3	0x00	内部振荡器校准寄存器	
OSCICN	0xB2	0x00	内部振荡器控制寄存器	
OSCXCN	0xB1	0x00	外部振荡器控制寄存器	
P0	0x80	所有页	端口 0 锁存器	
P0DRV	0xA4	0x0F	端口 0 驱动强度寄存器	
P0MASK	0xC7	0x00	端口 0 屏蔽寄存器	
P0MAT	0xD7	0x00	端口 0 匹配寄存器	
P0MDIN	0xF1	0x00	端口 0 输入方式配置寄存器	
P0MDOUT	0xA4	0x00	端口 0 输出方式配置寄存器	
P0SKIP	0xD4	0x00	端口 0 跳过寄存器	
P1	0x90	所有页	端口 1 锁存器	
P1DRV	0xA5	0x0F	端口 1 驱动强度寄存器	
P1MASK	0xBF	0x00	端口 1 屏蔽寄存器	
P1MAT	0xCF	0x00	端口 1 匹配寄存器	
P1MDIN	0xF2	0x00	端口 1 输入方式配置寄存器	
P1MDOUT	0xA5	0x00	端口 1 输出方式配置寄存器	
P1SKIP	0xD5	0x00	端口 1 跳过寄存器	
P2	0xA0	所有页	端口 2 锁存器	
P2DRV	0xA6	0x0F	端口 2 驱动强度寄存器	
P2MDIN	0xF3	0x00	端口 2 输入方式配置寄存器	
P2MDOUT	0xA6	0x00	端口 2 输出方式配置寄存器	
P2SKIP	0xD6	0x00	端口 2 跳过寄存器	

表 11.3 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	SFR 页	说　明	页码
PCA0CN	0xD8	0x00	PCA0 控制寄存器	
PCA0CPH0	0xFC	0x00	PCA0 模块 0 捕捉/比较高字节	
PCA0CPH1	0xEA	0x00	PCA0 模块 1 捕捉/比较高字节	
PCA0CPH2	0xEC	0x00	PCA0 模块 2 捕捉/比较高字节	
PCA0CPH3	0xEE	0x00	PCA0 模块 3 捕捉/比较高字节	
PCA0CPH4	0xFE	0x00	PCA0 模块 4 捕捉/比较高字节	
PCA0CPH5	0xD3	0x00	PCA0 模块 5 捕捉/比较高字节	
PCA0CPL0	0xFB	0x00	PCA0 模块 0 捕捉/比较低字节	
PCA0CPL1	0xE9	0x00	PCA0 模块 1 捕捉/比较低字节	
PCA0CPL2	0xEB	0x00	PCA0 模块 2 捕捉/比较低字节	
PCA0CPL3	0xED	0x00	PCA0 模块 3 捕捉/比较低字节	
PCA0CPL4	0xFD	0x00	PCA0 模块 4 捕捉/比较低字节	
PCA0CPL5	0xD2	0x00	PCA0 模块 5 捕捉/比较低字节	
PCA0CPM0	0xDA	0x00	PCA0 模块 0 方式寄存器	
PCA0CPM1	0xDB	0x00	PCA0 模块 1 方式寄存器	
PCA0CPM2	0xDC	0x00	PCA0 模块 2 方式寄存器	
PCA0CPM3	0xDD	0x00	PCA0 模块 3 方式寄存器	
PCA0CPM4	0xDE	0x00	PCA0 模块 4 方式寄存器	
PCA0CPM5	0xCE	0x00	PCA0 模块 5 方式寄存器	
PCA0H	0xFA	0x00	PCA0 计数器高字节	
PCA0L	0xF9	0x00	PCA0 计数器低字节	
PCA0MD	0xD9	0x00	PCA0 方式寄存器	
PCA0PWM	0xDF	0x00	PCA0 PWM 配置寄存器	
PCON	0x87	0x00	电源控制寄存器	
PMU0CF	0xB5	0x00	PMU0 配置寄存器	
PSCTL	0x8F	0x00	程序存储器读/写控制寄存器	
PSW	0xD0	所有页	程序状态字	
REF0CN	0xD1	0x00	电压基准控制寄存器	
DC0CF	0x96	0x00	DC0（DC-DC 转换器）配置寄存器	
DC0CN	0x97	0x00	DC0（DC-DC 转换器）控制寄存器	
REG0CN	0xC9	0x00	稳压器（VREG0）控制寄存器	
RSTSRC	0xEF	0x00	复位源寄存器	
RTC0ADR	0xAC	0x00	RTC0 地址寄存器	
RTC0DAT	0xAD	0x00	RTC0 数据寄存器	
RTC0KEY	0xAE	0x00	RTC0 关键码寄存器	
SBUF0	0x99	0x00	UART0 数据缓冲器	
SCON0	0x98	0x00	UART0 控制寄存器	
SFRPAGE	0xA7	所有页	SFR 页寄存器	

表 11.3 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	SFR 页	说　明	页码
SMB0ADM	0xF5	0x00	SMBus 0 从地址掩码	
SMB0ADM	0xF4	0x00	SMBus 0 从地址	
SMB0CF	0xC1	0x00	SMBus 0 配置寄存器	
SMB0CN	0xC0	0x00	SMBus 0 控制寄存器	
SMB0DAT	0xC2	0x00	SMBus 0 数据寄存器	
SP	0x81	所有页	堆栈指针	
SPI0CFG	0xA1	0x00	SPI0 配置寄存器	
SPI0CKR	0xA2	0x00	SPI0 时钟速率控制寄存器	
SPI0CN	0xF8	0x00	SPI0 控制寄存器	
SPI0DAT	0xA3	0x00	SPI0 数据寄存器	
SPI1CFG	0x84	0x00	SPI1 配置寄存器	
SPI1CKR	0x85	0x00	SPI1 时钟速率控制寄存器	
SPI1CN	0xB0	0x00	SPI1 控制寄存器	
SPI1DAT	0x86	0x00	SPI1 数据寄存器	
TCON	0x88	0x00	定时器/计数器控制寄存器	
TH0	0x8C	0x00	定时器/计数器 0 高字节	
TH1	0x8D	0x00	定时器/计数器 1 高字节	
TL0	0x8A	0x00	定时器/计数器 0 低字节	
TL1	0x8B	0x00	定时器/计数器 1 低字节	
TMOD	0x89	0x00	定时器/计数器方式寄存器	
TMR2CN	0xC8	0x00	定时器 2 控制寄存器	
TMR2H	0xCD	0x00	定时器 2 高字节	
TMR2L	0xCC	0x00	定时器 2 低字节	
TMR2RLH	0xCB	0x00	计数器/定时器 2 重载值高字节	
TMR2RLL	0xCA	0x00	计数器/定时器 2 重载值低字节	
TMR3CN	0x91	0x00	定时器 3 控制寄存器	
TMR3H	0x95	0x00	定时器 3 高字节	
TMR3L	0x94	0x00	定时器 3 低字节	
TMR3RLH	0x93	0x00	计数器/定时器 3 重载值高字节	
TMR3RLL	0x92	0x00	计数器/定时器 3 重载值低字节	
TOFFH	0x86	0x0F	温度偏移高字节	
TOFFL	0x85	0x0F	温度偏移低字节	
VDM0CN	0xFF	0x00	VDD 监视器控制寄存器	
XBR0	0xE1	0x00	端口 I/O 交叉开关控制寄存器 0	
XBR1	0xE2	0x00	端口 I/O 交叉开关控制寄存器 1	
XBR2	0xE3	0x00	端口 I/O 交叉开关控制寄存器 2	

12. 中断系统

C8051F93x-C8051F92x 包含一个扩展的中断系统，支持多个中断源，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。器件支持的所有中断源列于表 12.1“中断一览表”中。关于某个外设的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。

每个中断源可以在一个 SFR 或一个间接寄存器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。如果全局中断和具体的中断源均被使能，则当中断标志被置 1 时会向 CPU 请求中断。

一旦当前指令执行结束，CPU 立即产生一个 LCALL 到预定地址，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。）

某些中断标志在 CPU 进入 ISR 时被硬件自动清除，但大多数中断标志不是由硬件清除的，必须在从 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再次进入该 ISR。

12.1 使能中断源

每个中断源都可以用中断允许或扩展中断允许寄存器中的相应中断使能位来独立地允许或禁止，但是必须通过将 EA 位（IE.7）置 1 来总体使能中断，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，清除 EA 位将禁止所有中断。在 EA 位被清 0 期间所发生的中断请求被保持，直到 EA 位被置 1 后才能得到服务。

12.2 MCU 中断源和中断向量

CPU 通过产生一个 LCALL 到预定地址（中断向量地址）开始执行中断服务程序（ISR），提供中断服务。表 12.1 列出了每个中断源的中断向量地址。软件应保证每个被使能的中断源的中断向量包含一个有效的中断服务程序。

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。

12.3 中断优先级

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。如果一个高优先级的中断抢占了一个低优先级的中断，则低优先级的中断要等到高优先级的中断执行完后再完成其中断服务。每个中断在中断优先级和扩展中断优先级寄存器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。

如果两个中断同时被识别，则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中断先得到服务。参见表 12.1 来确定同时被识别的中断之固定的优先级顺序。

12.4 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请求标志采样并对优先级译码。最快的响应时间为 7 个系统时钟周期：一个周期用于检测中断，一个周期用于执行一条指令，5 个周期用于完成对 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 19 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 个时钟周期完成 DIV 指令，5 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。

表 12.1 中断一览表

中断源	中断向量	优先级	中断标志	位寻址	硬件清除	中断允许	优先级控　制
复位	0x0000	最高	无	N/A	N/A	始终允许	总是最高
外部中断 0 (/INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
定时器 0 溢出	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
外部中断 1 (/INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
定时器 1 溢出	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
定时器 2 溢出	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN(SPI0CN.4)	Y	N	ESPI0 (IE.6)	PSPI0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
smaRTClock 告警	0x0043	8	ALRM (RTC0CN.2) [2]	N	N	ERTC0A (EIE1.1)	PRTC0A (EIP1.1)
ADC0 窗口比较	0x004B	9	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
ADC0 转换结束	0x0053	10	AD0INT (ADC0CN.5)	Y	N	EADC0 (EIE1.3)	PADC0 (EIP1.3)
可编程计数器阵列	0x005B	11	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.4)	PPCA0 (EIP1.4)
比较器 0	0x0063	12	CP0FIF(CPT0CN.4) CP0RIF(CPT0CN.5)	N	N	ECP0 (EIE1.5)	PCP0 (EIP1.5)
比较器 1	0x006B	13	CP1FIF(CPT1CN.4) CP1RIF(CPT1CN.5)	N	N	ECP1 (EIE1.6)	PCP1 (EIP1.6)
定时器 3 溢出	0x0073	14	TF3H(TMR3CN.7) TF3L(TMR3CN.6)	N	N	ET3 (EIE1.7)	PT3 (EIP1.7)
VDD/DC+电源监视器早期告警	0x007B	15	VDDOK(VDM0CN.5) [1]			EWARN (EIE2.0)	PWARN (EIP2.0)
端口匹配	0x0083	16	无			EMAT (EIE2.1)	PMAT (EIP2.1)
smaRTClock 振荡器故障	0x008B	17	OSCFAIL(RTC0CN.5)[2]	N	N	ERTC0F (EIE2.2)	PRTC0F (EIP2.2)
SPI1	0x0093	18	SPIF (SPI1CN.7) WCOL (SPI1CN.6) MODF (SPI1CN.5) RXOVRN(SPI1CN.4)	N	N	ESPI1 (EIE2.3)	PSPI1 (EIP2.3)

注：
1. 表示是一个只读中断标志。中断使能可用于防止软件转向相应的中断服务程序。
2. 表示是一个位于间接存储器空间的寄存器。

12.5 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。关于外设的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。

SFR 定义 12.1　IE：中断允许寄存器

SFR 页：所有页
SFR 地址：0xA8　可位寻址

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　EA：允许所有中断。
该位总体允许 / 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：　ESPI0：串行外设接口（SPI0）中断允许位。
该位用于设置 SPI0 的中断屏蔽。
0：禁止 SPI0 中断。
1：允许 SPI0 的中断请求。

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 TF2L 或 TF2H 标志的中断请求。

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 UART0 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。
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

SFR 定义 12.2　IP：中断优先级寄存器

SFR 页：0x00
SFR 地址：0xB8　　可位寻址

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	PSPI0	PT2	PS0	PT1	PX1	PT0	PX0	10000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　未用。读=1b，写=忽略。

位 6：　PSPI0：串行外设接口（SPI0）中断优先级控制
该位设置 SPI0 中断的优先级。
0：SPI0 为低优先级。
1：SPI0 为高优先级。

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 为低优先级。
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 为低优先级。
1：UART0 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 为低优先级。
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 为低优先级。
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 为低优先级。
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 为低优先级。
1：外部中断 0 为高优先级。

SFR 定义 12.3　EIE1：扩展中断允许 1

SFR 页：所有页
SFR 地址：0xE6

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
ET3	ECP1	ECP0	EPCA0	EADC0	EWADC0	ERTC0A	ESMB0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　ET3：定时器 3 中断允许位
该位设置定时器 3 的中断屏蔽。
0：禁止定时器 3 中断。
1：允许 TF3L 或 TF3H 标志的中断请求。

位 6：　ECP1：比较器 1（CP1）中断允许位
该位设置 CP1 的中断屏蔽。
0：禁止 CP1 中断。
1：允许 CP1RIF 或 CP1FIF 标志产生的中断请求。

位 5：　ECP0：比较器 0（CP0）中断允许位
该位设置 CP0 的中断屏蔽。
0：禁止 CP0 中断。
1：允许 CP0RIF 或 CP0FIF 标志产生的中断请求。

位 4：　EPCA0：可编程计数器阵列（PCA0）中断允许位
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 3：　EADC0：ADC0 转换结束中断允许位
该位设置 ADC0 转换结束中断屏蔽。
0：禁止 ADC0 转换结束中断。
1：允许 AD0INT 标志的中断请求。

位 2：　EWADC0：ADC0 窗口比较中断允许位
该位设置 ADC0 窗口比较中断屏蔽。
0：禁止 ADC0 窗口比较中断。
1：允许 ADC0 窗口比较标志（AD0WINT）的中断请求。

位 1：　ERTC0A：smaRTClock 告警中断允许位
该位设置 smaRTClock 告警中断屏蔽。
0：禁止 smaRTClock 告警中断。
1：允许一个 smaRTClock 告警产生的中断请求。

位 0：　ESMB0：SMBus 中断允许位
该位设置 SMBus（SMB0）的中断屏蔽。
0：禁止 SMB0 中断。
1：允许 SMB0 产生的中断请求。

SFR 定义 12.4　EIP1：扩展中断优先级 1

SFR 页：所有页
SFR 地址：0xF6

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
PT3	PCP1	PCP0	PPCA0	PADC0	PWADC0	PRTC0A	PSMB0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　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。

位 6：　PCP1：比较器 1（CP1）中断优先级控制
该位设置 CP1 中断的优先级。
0：CP1 中断为低优先级。
1：CP1 中断为高优先级。

位 5：　PCP0：比较器 0（CP0）中断优先级控制
该位设置 CP0 中断的优先级。
0：CP0 中断为低优先级。
1：CP0 中断为高优先级。

位 4：　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 中断为低优先级。
1：PCA0 中断为高优先级。

位 3：　PADC0：ADC0 转换结束中断优先级控制
该位设置 ADC0 转换结束中断的优先级。
0：ADC0 转换结束中断为低优先级。
1：ADC0 转换结束中断为高优先级。

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。

位 1：　PRTC0A：smaRTClock 告警中断优先级控制
该位设置 smaRTClock 告警中断的优先级。
0：smaRTClock 告警中断为低优先级。
1：smaRTClock 告警中断为高优先级。

位 0：　PSMB0：SMBus（SMB0）中断优先级控制
该位设置 SMB0 中断的优先级。
0：SMB0 中断为低优先级。
1：SMB0 中断为高优先级。

SFR 定义 12.5　EIE2：扩展中断允许 2

SFR 页：所有页
SFR 地址：0xE7

R	R	R	R	R/W	R/W	R/W	R/W	复位值
—	—	—	—	ESPI1	ERTC0F	EMAT	EWARN	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-4：　未用。读 = 0000b，写 = 忽略。
位 3：　ESPI1：串行外设接口（SPI1）中断允许位。
该位用于设置 SPI1 的中断屏蔽。
0：禁止 SPI1 中断。
1：允许 SPI1 的中断请求。
位 2：　ERTC0F：smaRTClock 振荡器故障中断允许位
该位设置 smaRTClock 振荡器故障中断屏蔽。
0：禁止 smaRTClock 振荡器故障中断。
1：允许一个 smaRTClock 振荡器故障产生的中断请求。
位 1：　EMAT：端口匹配中断允许位
该位设置端口匹配事件的中断屏蔽。
0：禁止端口匹配中断。
1：允许端口匹配中断。
位 0：　EWARN：VDD/DC+电源监视器早期告警中断允许位
该位设置 VDD/DC+电源监视器早期告警中断屏蔽。
0：禁止 VDD/DC+电源监视器早期告警中断。
1：允许 VDD/DC+电源监视器产生的早期告警中断请求。

SFR 定义 12.6　EIP2：扩展中断优先级 2

SFR 页：所有页
SFR 地址：0xF7

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	PSPI1	PRTC0F	PMAT	PWARN	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-4：　未用。读 = 000000b，写 = 忽略。

位 3：　PSPI1：串行外设接口（SPI1）中断优先级控制
该位设置 SPI1 中断的优先级。
0：SPI1 为低优先级。
1：SPI1 为高优先级。

位 2：　PRTC0F：smaRTClock 振荡器故障中断优先级控制
该位设置 smaRTClock 振荡器故障中断的优先级。
0：smaRTClock 振荡器故障中断为低优先级。
1：smaRTClock 振荡器故障中断为高优先级。

位 1：　PMAT：端口匹配中断优先级控制
该位设置端口匹配事件中断的优先级。
0：端口匹配中断为低优先级。
1：端口匹配中断为高优先级。

位 0：　PWARN：VDD/DC+电源监视器早期告警中断优先级控制
该位设置 VDD/DC+电源监视器早期告警中断的优先级。
0：VDD/DC+电源监视器早期告警中断为低优先级。
1：VDD/DC+电源监视器早期告警中断为高优先级。

12.6 外部中断

两个外部中断源/INT0 和/INT1 可被配置为低电平有效或高电平有效，边沿触发或电平触发。IT01CF 寄存器中的 IN0PL（/INT0 极性）和 IN1PL（/INT1 极性）位用于选择高电平有效还是低电平有效；TCON 中的 IT0 和 IT1 用于选择电平或边沿触发。下面的表列出了可能的配置。

IT0	IN0PL	/INT0 中断
1	0	低电平有效，边沿触发
1	1	高电平有效，边沿触发
0	0	低电平有效，电平触发
0	1	高电平有效，电平触发

IT1	IN1PL	/INT1 中断
1	0	低电平有效，边沿触发
1	1	高电平有效，边沿触发
0	0	低电平有效，电平触发
0	1	高电平有效，电平触发

/INT0 和/INT1 所使用的端口引脚在 IT01CF 寄存器中定义（见 SFR 定义 12.7）。注意，/INT0 和/INT0 端口引脚分配与交叉开关的设置无关。/INT0 和/INT1 监视分配给它们的端口引脚，不影响被交叉开关分配了相同引脚的外设。如果要将一个端口引脚只分配给/INT0 或/INT1，则应使交叉开关跳过这个引脚。这可以通过设置寄存器 XBR0 中的相应位来实现（有关配置交叉开关的详细信息见“21.3 优先权交叉开关译码器”）。

IE0（TCON.1）和 IE1（TCON.3）分别为外部中断/INT0 和/INT1 的中断标志。如果/INT0 或/INT1 外部中断被配置为边沿触发，CPU 在转向 ISR 时用硬件自动清除相应的中断标志。当被配置为电平触发时，在输入有效期间（根据极性控制位 IN0PL 或 IN1PL 的定义）中断标志将保持在逻辑 1 状态；在输入无效期间该标志保持逻辑 0 状态。电平触发的外部中断源必须一直保持输入有效直到中断请求被识别，在 ISR 执行结束之前必须使该中断请求无效，否则将产生另一个中断请求。

SFR 定义 12.7　IT01CF：INT0/INT1 配置寄存器

SFR 页：0x00
SFR 地址：0xE4

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
IN1PL	IN1SL2	IN1SL1	IN1SL0	IN0PL	IN0SL2	IN0SL1	IN0SL0	00000001
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　IN1PL：/INT1 极性
0：/INT1 为低电平有效。
1：/INT1 为高电平有效。

位 6-4：　IN1SL[2:0]：/INT1 端口引脚选择位
这些位用于选择分配给/INT1 的端口引脚。注意，该引脚分配与交叉开关无关；/INT1 将监视分配给它的端口引脚，但不影响被交叉开关分配了相同引脚的外设。如果将交叉开关配置为跳过这个引脚（通过将寄存器 P0SKIP 中的对应位置 1 来实现），则该引脚将不会被分配给外设。

IN1SL[2:0]	/INT1 端口引脚
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6
111	P0.7

位 3：　IN0PL：/INT0 极性
0：/INT0 为低电平有效。
1：/INT0 为高电平有效。

位 2-0：　IN0SL[2:0]：/INT0 端口引脚选择位
这些位用于选择分配给/INT0 的端口引脚。注意，该引脚分配与交叉开关无关；/INT0 将监视分配给它的端口引脚，但不影响被交叉开关分配了相同引脚的外设。如果将交叉开关配置为跳过这个引脚（通过将寄存器 P0SKIP 中的对应位置 1 来实现），则该引脚将不会被分配给外设。

IN0SL[2:0]	/INT0 端口引脚
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6
111	P0.7

13. FLASH 存储器

C8051F93x-C8051F92x 内部有可编程的 FLASH 存储器，用于程序代码和非易失性数据存储。可以通过 C2 接口或由软件使用 MOVX 写指令对 FLASH 存储器进行在系统编程。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 FLASH 字节擦除（置为 0xFF）。为了保证操作正确，写和擦除操作由硬件自动定时，不需要进行数据查询来判断写/擦除操作何时结束。在 FLASH 写/擦除操作期间，程序停止执行。表 4.6 给出了 FLASH 存储器的电气特性。

13.1 FLASH 存储器编程

对 FLASH 存储器编程的最简单的方法是使用由 Silicon Labs 公司或第三方供应商提供的编程工具，通过 C2 接口编程，这是对未被初始化过的器件的唯一编程方法。有关对 FLASH 程序存储器编程的 C2 命令的详细信息见“27. C2 接口”。

可以在软件中使用 MOVX 指令对 FLASH 存储器编程，像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 存储器编程之前，必须用下面的步骤使能 FLASH 编程操作：1）将程序存储写允许位 PSWE（PSCTL.0）设置为逻辑 1（这将使 MOVX 写操作指向目标 FLASH 存储器）；2）按顺序向 FLASH 锁定寄存器（FLKEY）写入 FLASH 关键码。在用软件清除之前 PSWE 位将一直保持置位状态。有关从固件对 FLASH 编程的详细指南见“13.4 FLASH 写和擦除指南”。

在包含写和/或擦除 FLASH 存储器的代码的任何系统中，为了保证 FLASH 内容的完整性，必须使能 VDD 监视器并将其设置为复位源。而且在使能 VDD 监视器和将其使能为复位源之间不应有延时。在 VDD 监视器被禁止或未被使能为复位源期间，对 FLASH 存储器执行任何擦除或写操作都将导致 FLASH 错误器件复位。

13.1.1 FLASH 锁定和关键码功能

从用户软件写和擦除 FLASH 受 FLASH 锁定和关键码功能的保护。在进行 FLASH 操作之前，必须按顺序向 FLASH 锁定和关键码寄存器（FLKEY）写入正确的关键码。关键码为：0xA5，0xF1。写关键码的时序并不重要，但必须按顺序写。如果写关键码的顺序不对或写入了错误的关键码，FLASH 写和擦除操作将被禁止，直到下一次系统复位。如果在正确写入关键码之前进行了 FLASH 写或擦除操作，FLASH 写和擦除也将被禁止。每次 FLASH 写和擦除操作之后，FLASH 锁定功能复位；在进行下一次 FLASH 写或擦除操作之前，必须重新写关键码。FLKEY 寄存器的详细说明见 SFR 定义 13.2。

13.1.2 FLASH 擦除过程

FLASH 存储器是以 1024 字节的页为单位组织的，擦除操作将擦除一个整页（将页内的所有字节置为 0xFF）。擦除一个 1024 字节页的步骤如下：

1. 保存当前中断状态并禁止中断。
2. 置 1 程序存储器擦除允许位（PSCTL 中的 PSEE）。
3. 置 1 程序存储器写允许位（PSCTL 中的 PSWE）。
4. 向 FLKEY 写第一个关键码：0xA5。
5. 向 FLKEY 写第二个关键码：0xF1。

6. 用 MOVX 指令向待擦除页内的任何一个地址写入一个数据字节。
7. 清除 PSWE 和 PSEE 位。
8. 恢复操作前的中断状态。

对每个要擦除的 1024 字节页都必须重复步骤 4-6。

注：FLASH 安全设置可以防止擦除某些 FLASH 页，如保留区和包含锁定字节的页。有关影响 FLASH 擦除操作的 FLASH 安全设置和限制的说明，见“13.3 安全选项”。

13.1.3 FLASH 写过程

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。**在写入新值之前，必须先擦除待编程的字节地址**。

用软件对 FLASH 字节编程的步骤如下：

1. 保存当前中断状态并禁止中断。
2. 保证 FLASH 字节已经被擦除（值为 0xFF）。
3. 置位寄存器 PSCTL 中的 PSWE 位。
4. 清除寄存器 PSCTL 中的 PSEE 位。
5. 向 FLKEY 写第一个关键码：0xA5。
6. 向 FLKEY 写第二个关键码：0xF1。
7. 用 MOVX 指令向 1024 字节扇区内的目标地址写入一个数据字节。
8. 清除 PSWE 位。
9. 恢复操作前的中断状态。

对每个要写入的字节重复步骤 5-7。

注：FLASH 安全设置可以防止擦除某些 FLASH 页，如保留区和包含锁定字节的页。有关影响 FLASH 擦除操作的 FLASH 安全设置和限制的说明，见“13.3 安全选项”。

13.2 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。数据写入时用 MOVX 写指令，读出时用 MOVC 指令。注意：MOVX 读指令总是指向 XRAM。

另有 1024 字节的临时存储区可用于非易失性数据存储。当 SFLE 被置 1 时，可以用 0x0000 ~ 0x03FF 的地址范围访问该存储区。临时存储区不能用于执行代码。

13.3 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。程序存储器写允许（PSCTL 寄存器中的 PSWE 位）和程序存储器擦除允许（PSCTL 寄存器中的 PSEE 位）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，PSWE 必须被置 1；在用软件擦除 FLASH 存储器之前，PSWE 位和 PSEE 位都必须被置 1。此外，CIP-51 还提供了可以防止通过 C2 接口读取或修改产权程序代码和数据常数这一安全功能。

位于 FLASH 用户空间的最后一个字节中的安全锁定字节保护 FLASH 存储器，使其不能

被非保护代码或通过 C2 接口读、写或擦除。FLASH 安全机制允许用户从 0 页（地址 0x0000～0x03FF）开始锁定 n 个 1024 字节的 FLASH 页，其中 n 是安全锁定字节的反码。**注意：在没有其它 FLASH 页被锁定时（锁定字节的所有位均为 1），包含 FLASH 安全锁定字节的页不被锁定。当任何一个其他 FLASH 页被锁定时（锁定字节有任何一位为 0），包含 FLASH 安全锁定字节的页也被锁定。**下面是针对 C8051F930 的锁定字节示例。

安全锁定字节：	11111101 b
反码：	00000010 b
被锁定的 FLASH 页：	3（前两个 FLASH 页 + 锁定字节页）
被锁定的地址：	0x0000～0x07FF（前两个 FLASH 页）和 0xF800～0xFBFF（锁定字节页）

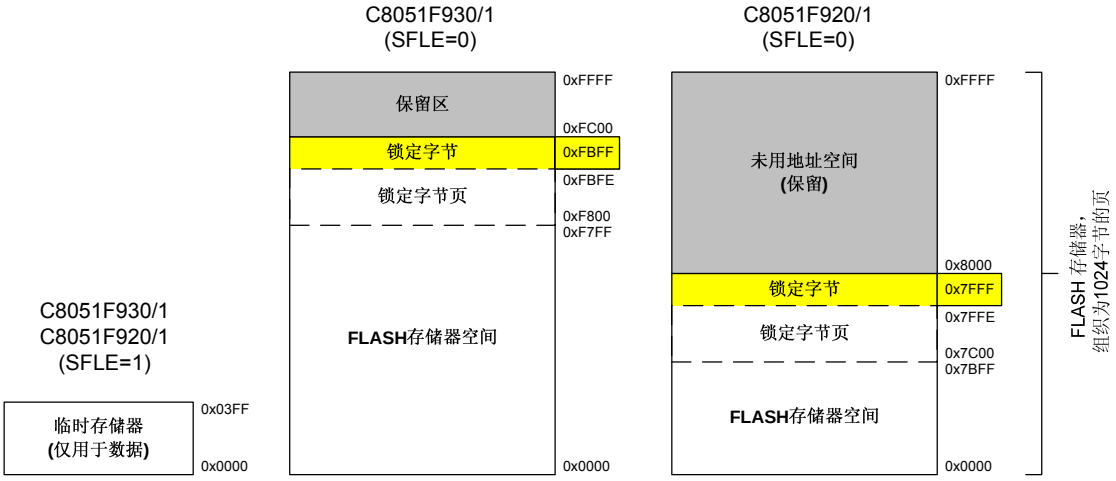


图 13.1　FLASH 程序存储器组织

FLASH 安全级别取决于对 FLASH 访问的方式。有 3 种可被限制的访问方式：经 C2 调试接口、在非锁定页执行的用户固件、在锁定页执行的用户固件对 FLASH 的读、写和擦除。表 13.1 概述了 C8051F93x-C8051F92x 器件的 FLASH 安全特性。

表 13.1　FLASH 安全一览表

操作	C2 调试接口	用户固件所在执行区域：	
		未锁定页	被锁定页
读、写或擦除未锁定页 （锁定字节所在页除外）	允许	允许	允许
读、写或擦除被锁定页 （锁定字节所在页除外）	不允许	FLASH 错误复位	允许
读或写锁定字节所在页 （如果没有被锁定的页）	允许	允许	允许
读或写锁定字节所在页 （如果有任何页被锁定）	不允许	FLASH 错误复位	允许
读锁定字节的内容 （如果没有被锁定的页）	允许	允许	允许
读锁定字节的内容 （如果有任何页被锁定）	不允许	FLASH 错误复位	允许
擦除锁定字节所在页 （如果没有被锁定的页）	允许	FLASH 错误复位	FLASH 错误复位
擦除锁定字节所在页—解锁所有页（如果有任何页被锁定）	只能进行 C2 器件擦除	FLASH 错误复位	FLASH 错误复位
锁定附加页 （将锁定字节中的 1 变成 0）	不允许	FLASH 错误复位	FLASH 错误复位
解锁单个页 （将锁定字节中的 0 变成 1）	不允许	FLASH 错误复位	FLASH 错误复位
读、写或擦除保留区	不允许	FLASH 错误复位	FLASH 错误复位

C2 器件擦除——擦除所有 FLASH 页，包括锁定字节所在页。

FLASH 错误复位——不允许相应的操作；导致 FLASH 错误器件复位（复位后寄存器 RSTSRC 中的 FERROR 位为 1）。

— 经 C2 接口的所有被禁止的操作都被忽略（不会导致器件复位）。
— 锁定任何一个 FLASH 页时，包含锁定字节的页也被锁定。
— 锁定字节一旦被写入便不能被修改，除非执行一次 C2 器件擦除。
— 如果用户代码写锁定字节，则在下一次复位之前锁定功能不会生效。
— 在所有其他 FLASH 页都被锁定时，临时存储区被锁定。
— 当执行一次 FLASH 器件擦除命令时，临时存储区被擦除。

13.4 FLASH 写和擦除指南

如果 CPU 工作在 VDD、系统时钟频率或温度的额定范围之外，任何包含写或擦除 FLASH 代码的系统都存在这样的危险，即意外执行写或擦除 FLASH 的代码。修改 FLASH 内容的代码之意外执行会导致 FLASH 存储器内容的改变，所引发的系统故障只能通过重新烧写 FLASH 来解决。

为了防止固件意外修改 FLASH，VDD 监视器必须被使能并被选择为复位源，只有这样 FLASH 才能被成功改写。**如果 VDD 监视器或 VDD 监视器复位源未被使能，则当固件试图改写 FLASH 时会产生 FLASH 错误器件复位。**

建议在任何包含写或擦除 FLASH 代码的系统中遵循下述指南。

13.4.1 VDD 维护和 VDD 监视器

1. 如果系统电源易受电压或电流尖峰的干扰，应在电源部分增加瞬变保护器件，确保电源电压不超过极限值。
2. 保证满足 1ms 的最小上升时间。如果系统不满足这个最小上升时间指标，则要在器件的复位引脚加一个外部 VDD 欠压检测电路，以使器件在 VDD 达到最小器件工作电压前保持复位状态和在 VDD 下降到低于最小器件工作电压时使复位引脚有效。
3. 在代码中尽可能早地使能片内 VDD 监视器并将其使能为复位源。这应该是复位向量之后最先被执行的指令。对于用 C 语言开发的系统，要做到这一点需修改随 C 编译器提供的启动代码。有关这方面的详细信息，请参见所用编译器的文档。要保证在使能 VDD 监视器和将其使能为复位源之间的软件没有延时。在 Silicon Laboratories 网站上提供的“AN201：从固件写 FLASH”中给出了示例代码。

   注：对于 C8051F93x-C8051F92x 器件，VDD 监视器和 VDD 监视器复位源都必须被使能，只有如此才不会在写或擦除 FLASH 时产生 FLASH 错误器件复位。

   注：对于 C8051F93x-C8051F92x 器件，VDD 监视器和 VDD 监视器复位源在上电复位后由硬件使能。

4. 可以增加一层预防措施，即在写和擦除 FLASH 存储器的函数中显式地使能 VDD 监视器和将其使能为复位源。使能 VDD 监视器的指令应紧接在将 PSWE 置 1 的指令之后，但位于 FLASH 写或擦除操作指令之前。
5. 保证所有写 RSTSRC（复位源）寄存器的指令都使用直接赋值操作符显式赋值，不要使用位操作（如 AND 或 OR）。例如，“RSTSRC = 0x02”是正确的，而“RSTSRC |= 0x02”是不正确的。
6. 保证所有写 RSTSRC 寄存器的指令都显式地将 PORSF 位置 1。检查使能其它复位源的初始化代码（例如时钟丢失检测器或比较器）和强制软件复位的指令。通过全局搜索“RSTSRC”可以快速完成检查。

13.4.2　PSWE 维护

1. 在代码中尽量减少将 PSWE 位（PSCTL 的位 0）置 1 的位置数。在代码中应只使用一个将 PSWE 置 1 的例程（写 FLASH 字节）和一个将 PSWE 及 PSEE 置 1 的例程（擦除 FLASH 页）。
2. 在 PSWE 被置 1 期间，尽量减少变量访问次数。在“PSWE = 1; … PSWE = 0;”的区域之外处理指针地址更新和改变循环变量。在 Silicon Laboratories 网站上提供的“AN201：从固件写 FLASH”中给出了示例代码。
3. 在将 PSWE 置 1 之前禁止中断，并保持中断的禁止状态直到 PSWE 被清 0。在 FLASH 写或擦除操作期间所产生的任何中断都会在 FLASH 操作完成和中断被软件重新使能之后按优先级顺序得到服务。
4. 保证 FLASH 写和擦除指针变量不位于 XRAM 空间。有关如何显式地将变量定位到不同存储器区域的说明请参见您所使用的编译器的文档。
5. 在写或擦除 FLASH 存储器的例程中增加地址边界检查，以保证在使用非法地址调用一个例程时不会修改 FLASH。

13.4.3　系统时钟

1. 如果 CPU 使用外部晶体工作，应注意晶体的性能容易受到电气干扰的影响和对布局布线及温度变化敏感。如果系统工作在有强电气噪声的环境，应使用内部振荡器或外部 CMOS 时钟。
2. 如果 CPU 使用外部振荡器工作，在 FLASH 写或擦除操作期间将系统时钟切换到内部振荡器。外部振荡器可以继续运行，CPU 可以在 FLASH 操作结束后切换回外部振荡器。

在 Silicon Laboratories 网站上提供的“AN201：从固件写 FLASH”中可以找到其他 FLASH 操作建议和示例代码。

SFR 定义 13.1　PSCTL：程序存储读写控制

SFR 页：0x00
SFR 地址：0x8F

R	R	R	R	R	R	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	

位 7-3：　未使用。读 = 00000b，写 = 忽略。

位 2：　SFLE：临时 FLASH 存储区访问使能
当该位被置 1 时，用户软件中的 MOVC 读和 MOVX 写操作指向临时 FLASH 扇区。当该位被置 1 时，不应访问地址范围 0x0000 ~ 0x03FF 以外的 FLASH 地址，否则会得到不确定的结果。
0：用户软件的 FLASH 访问指向程序/数据 FLASH 扇区。
1：用户软件的 FLASH 访问指向临时存储扇区。

位 1：　PSEE：程序存储擦除允许
将该位置 1 后允许擦除 FLASH 存储器中的一个页(前提是 PSWE 位也被置 1)。如果该位被置 1 并且 FLASH 写操作被使能（PSWE 为逻辑 1），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。写入的数据字节可以是任意值。
0：禁止擦除 FLASH 程序存储器。
1：允许擦除 FLASH 程序存储器。

位 0：　PSWE：程序存储写允许
将该位置 1 后允许用 MOVX 写指令向 FLASH 程序存储器写一个字节。在写数据之前必须先将待写地址擦除。
0：禁止写 FLASH 程序存储器。
1：允许写 FLASH 程序存储器；MOVX 写指令寻址 FLASH 存储器。

SFR 定义 13.2　FLKEY：FLASH 锁定和关键码寄存器

SFR 页：0x00
SFR 地址：0xB7

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	

位 7-0：　FLKEY：FLASH 锁定和关键码寄存器

写：

该寄存器为 FLASH 擦除和写操作提供锁定和关键码功能。通过向该寄存器按顺序写入关键码 0xA5 和 0xF1 来使能 FLASH 写和擦除。在执行完一次写或擦除操作后，后续的 FLASH 写或擦除操作被自动禁止。如果写 FLKEY 操作不正确或在写或擦除操作被禁止时试图写或擦除 FLASH，则 FLASH 将被永久性锁定（不能写或擦除），直到下一次器件复位。如果应用固件从不写 FLASH，可以用软件向 FLKEY 写入一个非 0xA5 的值，以锁定 FLASH。

读：

位 1-0 指示当前的 FLASH 锁定状态

00：FLASH 写/擦除被锁定。

01：第一个关键码已被写入（0xA5）。

10：FLASH 处于解锁状态（允许写/擦除）

11：FLASH 写/擦除操作被禁止，直到下一次复位。

SFR 定义 13.3　FLSCL：FLASH 定时寄存器

SFR 页：0x00
SFR 地址：0xB6

R	R/W	R	R	R	R	R	R	复位值
保留	BYPASS	保留	保留	保留	保留	保留	保留	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　保留。总是写 0。

位 6：　BYPASS：FLASH 读定时单稳旁路控制。

0：单稳决定 FLASH 读时间。在工作频率小于 10 MHz 时应使用该设置。

1：系统时钟决定 FLASH 读时间。在工作频率大于 10 MHz 时应使用该设置。

位 5-0：　保留。总是写 000000b。

14. 电源管理

C8051F93x-C8051F92x 器件支持 5 种电源方式：正常、空闲、停机、挂起和休眠。电源管理单元（PMU0）允许器件进入可用的电源方式和从某种方式唤醒。表 14.1 给出了这些工作方式的简要说明。对每种方式的详细说明见后续各节。

表14.1 工作方式一览表

	描述	唤醒源	省电情况
正常	器件全功能工作	N/A	极佳的MIPS/mW
空闲	所有外设全功能工作。 非常容易唤醒。	任何中断	好 不执行代码
停机	遗留的8051低功耗方式。 需要靠复位来唤醒。	任何复位	好 不执行代码 精准振荡器被禁止
挂起	类似于停机方式，但具有很快的唤醒时间。代码从下一条指令恢复执行。	smaRTClock、 端口匹配、 比较器0、 /RST引脚	很好 不执行代码 所有内部振荡器被禁止 系统时钟被解除
休眠	极低的功耗和灵活的唤醒源。 代码从下一条指令恢复执行。 比较器0只能在双节电池模式工作。	smaRTClock、 端口匹配、 比较器0、 /RST引脚	极好 电源被解除 除smaRTClock之外的 所有振荡器都被禁止

电池供电的系统应尽可能长时间地处于休眠方式，以保持电池寿命。当一个具有固定时钟周期数的任务需要执行时，器件应切换到正常方式，尽可能快地完成任务，再返回到休眠方式。空闲方式和挂起方式提供很快的唤醒时间，但这些方式的节电效果不如休眠方式。停机方式是由于遗留的原因而保留下来的。当系统使用空闲、挂起或休眠方式时，电源效率高并且易于唤醒。

虽然开关电源方式是电源管理的一个组成部分，但是在所有电源方式，根据需要使能/禁止每个外设有助于降低功耗。每个模拟外设在不使用时都可以被禁止或被置于低功耗方式。数字外设如定时器或串行总线在不使用时消耗很小的功率。数字外设在休眠方式不消耗功率。

14.1 正常方式

MCU 在正常方式全功能工作。图 14.1 示出了片内电源在各外设之间的分布。有 3 个电源电压为片上的不同部分供电：VBAT、VDD/DC+和 1.8V 的内核电源。REG0、PMU0 和 smaRTClock 总是由 VBAT 引脚直接供电。所有模拟外设都直接由 VDD/DC+引脚供电，VDD/DC+是单节电池模式下的输出和双节电池模式下的输入。所有数字外设和 CIP-51 核都由 1.8V 的内核电源供电。RAM 在正常方式下也由内核电源供电。

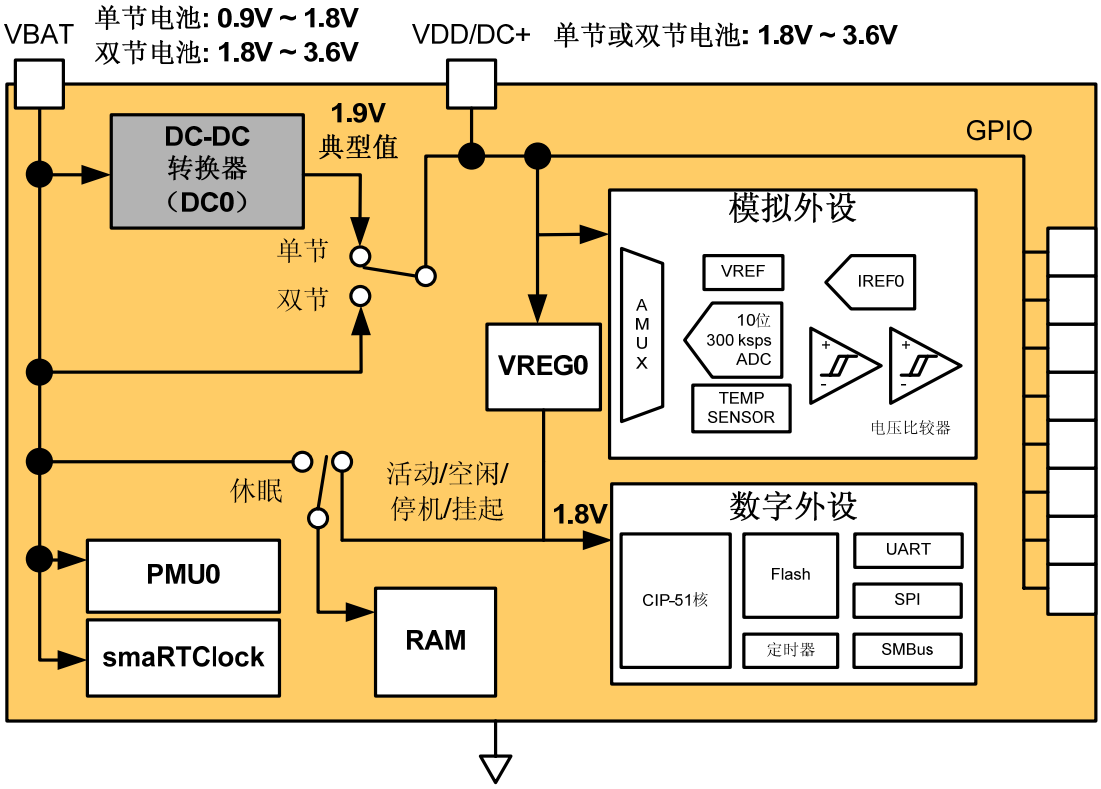


图 14.1　C8051F93x-C8051F92x 电源分布

14.2 空闲方式

将空闲方式选择位（PCON.0）置 1 使 CIP-51 停止 CPU 运行并进入空闲方式，在执行完对该位置 1 的指令后 MCU 立即进入空闲方式。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在空闲方式期间都可以保持活动状态。

有被允许的中断发生或复位有效将结束空闲方式。当有一个被允许的中断发生时，空闲方式选择位（PCON.0）被清 0，CPU 将继续工作。该中断将得到服务，中断返回（RETI）后将开始执行设置空闲方式选择位的那条指令的下一条指令。如果空闲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。

如果被使能，看门狗定时器（WDT）将产生一个内部看门狗复位，从而结束空闲方式。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，可以在进入空闲方式之前用软件禁止 WDT（如果 WDT 被初始化为允许该操作）。这将进一步节省功耗，允许系统一直保持在空闲状态，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，见“18.6 PCA 看门狗定时器复位”。

14.3 停机方式

将停机方式选择位（PCON.1）置 1 使 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，精准内部振荡器和 CPU 停止工作，但低功耗振荡器和外部振荡器电路的状态不受影响。在进入停机方式之前，每个模拟外设（包括外部振荡器电路）都可以被单独关断。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。

如果被使能，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。

停机方式是一种遗留的 8051 电源方式，它不会产生最佳的节电效果。如果 MCU 需要长时间处于非活动状态，休眠或挂起方式可以提供更好的节电能力。

14.4 挂起方式

通过置 1 挂起方式选择位（PMU0CF.6）可以解除系统时钟和禁止所有内部振荡器。所有数字逻辑（定时器、通信外设、中断、CPU 等）都停止工作，直到有一个被使能的唤醒源事件发生。

注意：当进入挂起或休眠方式时，全局时钟分频系数必须被设置为“1 分频”。

下面的唤醒源可以被配置为将器件从挂起方式唤醒：

- smaRTClock 振荡器故障事件
- smaRTClock 告警事件
- 端口匹配事件。
- 比较器 0 上升沿。

另外，/RST 引脚上出现的宽度不足以使器件复位的短时间噪声脉冲会导致器件退出挂起方式。建议在/RST 引脚加一个 4.75 KΩ的上拉电阻到 VDD/DC+，以防止短时间噪声脉冲将器件唤醒。

14.5 休眠方式

通过置 1 休眠方式选择位（PMU0CF.7）可以关闭内部 1.8 V 稳压器（REG1），并将所有片内 RAM 的供电电压切换到 VBAT 引脚（见图 14.1）。片上大部分数字逻辑被断电，只有 PMU0 和 smaRTClock 保持供电。模拟外设在双节电池模式下保持供电，但在单节电池模式下会失去电源，因为在这种情况下 DC-DC 变换器被禁止。在双节电池模式下，当器件进入休眠方式时，只有比较器仍然可以工作。所有其他模拟外设（ADC0、IREF0、外部振荡器等）都应在器件进入休眠方式之前被禁止。

注意：当进入挂起或休眠方式时，全局时钟分频系数必须被设置为“1 分频”。

被配置为数字输出的 GPIO 引脚在器件处于休眠方式期间仍然保持其输出状态。在双节电池模式，这些引脚在休眠方式下的电流驱动能力与在正常方式保持相同。在单节电池模式，VDD/DC+电源电压会降低到 VBAT 电平，这将降低输出高电压的电平以及拉电流和灌电流能力。

被配置为数字输入的 GPIO 引脚在器件处于休眠方式期间可被用作唤醒源（使用端口匹配功能）。在双节电池模式，这些引脚在休眠方式下会保持与正常方式相同的输入电平指标。在单节电池模式，VDD/DC+电源电压会降低到 VBAT 电平，这将降低开关电平和增加延迟。在 VBAT 电压范围的低端，该延迟可能达到几百个纳秒。

只要 VBAT 引脚上的电压不降到 V_{POR} 以下，休眠方式下 RAM 和 SFR 寄存器的内容就会保持不变。PC 计数器及所有其他易失性状态信息都被保持，这就允许器件在从休眠方式被唤醒时恢复代码执行。

下面的唤醒源可以被配置为将器件从休眠方式唤醒：

- smaRTClock 振荡器故障事件
- smaRTClock 告警事件
- 端口匹配事件。
- 比较器 0 上升沿。

比较器 0 上升沿唤醒只在双节电池模式有效。比较器需要一个至少为 1.8V 的供电电压才能正常工作。

另外，/RST 引脚上出现的宽度不足以使器件复位的短时间噪声脉冲会导致器件退出休眠方式。建议在/RST 引脚加一个 4.75 KΩ的上拉电阻到 VDD/DC+，以防止短时间噪声脉冲将器件唤醒。

14.6 配置唤醒源

在将器件置于一种低功耗方式之前，应使能一个或多个唤醒源，以使器件不会无限期地保持在这种低功耗方式。对于空闲方式，这还包括使能任何中断。对于停机方式，这包括使能任何复位源，或者依赖/RST 引脚复位器件。

挂起方式和休眠方式的唤醒源通过 PMU0CF 寄存器配置。通过向对应的唤醒源使能位写 1 来使能唤醒源。每次器件被置于挂起方式或休眠方式时都必须重新使能唤醒源，使能唤醒源与将器件置于低功耗方式使用同一个写操作。

14.7 确定引起唤醒的事件

当从空闲方式被唤醒时，CPU 将转向唤醒它的中断。当从停机方式被唤醒时，应读 RSTSRC 寄存器来确定最后一次复位的原因。

当器件从挂起方式或休眠方式退出时，可以读 PMU0CF 寄存器中的唤醒标志来确定使器件唤醒的事件。在器件被唤醒后，如果有任何唤醒事件发生，唤醒标志将继续被更新。唤醒标志总是被更新，即使它们没有被使能为唤醒源。

在器件进入挂起方式或休眠方式之前，PMU0CF 中被使能为唤醒源的所有唤醒标志都必须被清 0。在清除唤醒标志后，应检查外设中每个被使能的唤醒事件，以保证在唤醒标志被清除期间没有唤醒事件发生。

SFR 定义 14.1　PMU0CF：电源管理单元配置寄存器 [1,2]

SFR 页：0x00
SFR 地址：0xB5

W	W	W	R	R/W	R/W	R/W	R/W	复位值
SLEEP	SUSPEND	CLEAR	RSTWK	RTCFWK	RTCAWK	PMATWK	CPT0WK	000xxxxx
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　SLEEP：休眠方式选择
写 1 将器件置于休眠方式。

位 6：　SUSPEND：挂起方式选择
写 1 将器件置于挂起方式。

位 5：　CLEAR：唤醒标志清除
写 1 将清除所有唤醒标志。

位 4：　RSTWK：复位引脚唤醒标志
/RST 引脚检测到一个短脉冲时被置 1。。

位 3：　RTCFWK：smaRTClock 振荡器故障唤醒源使能和标志
写：
0：禁止 smaRTClock 振荡器故障为唤醒源。
1：使能 smaRTClock 振荡器故障为唤醒源。
读：
发生 smaRTClock 振荡器故障时被置 1。

位 2：　RTCAWK：smaRTClock 振荡器告警唤醒源使能和标志
写：
0：禁止 smaRTClock 振荡器告警为唤醒源。
1：使能 smaRTClock 振荡器告警为唤醒源。
读：
发生 smaRTClock 振荡器告警时被置 1。

位 1：　PMATWK：端口匹配唤醒源使能和标志
写：
0：禁止端口匹配事件为唤醒源。
1：使能端口匹配事件为唤醒源。
读：
发生端口匹配事件时被置 1。

位 0：　CPT0WK：比较器 0 唤醒源使能和标志
写：
0：禁止比较器 0 上升沿为唤醒源。
1：使能比较器 0 上升沿为唤醒源。
读：
发生比较器 0 上升沿时被置 1。

注：
1. 不应对该寄存器使用读-修改-写操作（ORL、ANL 等）。每当 SLEEP 或 SUSPEND 位被写 1 时必须重新使能唤醒源。
2. 如果有任何唤醒标志被置 1，则低功耗内部振荡器不能被禁止，MCU 也不能被置于挂起方式或休眠方式。在器件每次复位和从挂起或休眠方式被唤醒时，软件应清除所有的唤醒源。

SFR 定义 14.2 PCON：电源控制寄存器

SFR 页：所有页
SFR 地址：0x87

R/W	R/W	R/W	R/W	R/W	R/W	W	W	复位值
GF5	GF4	GF3	GF2	GF1	GF0	STOP	IDLE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-2： GF[5:0]：通用标志。
写：设置逻辑值。
读：返回逻辑值。
位 1： STOP：停机方式选择。
写 1 使器件进入停机方式。
位 0： IDLE：空闲方式选择。
写 1 使器件进入空闲方式。

14.8 电源管理单元技术指标

电源管理单元技术指标的详细说明见“4. 电气特性”。

15. 循环冗余检查单元（CRC0）

C8051F93x-C8051F92x 器件包含一个循环冗余检查单元（CRC0）。CRC0 能使用 16 位或 32 位多项式执行 CRC。CRC0 接收写到 CRC0IN 寄存器的 8 位数据流，向一个内部寄存器输出 16 位或 32 位的结果。内部结果寄存器可以用 CRC0PNT 位和 CRC0DAT 寄存器间接访问，如图 15.1 所示。CRC0 还有一个位反转寄存器，以加速数据操作。

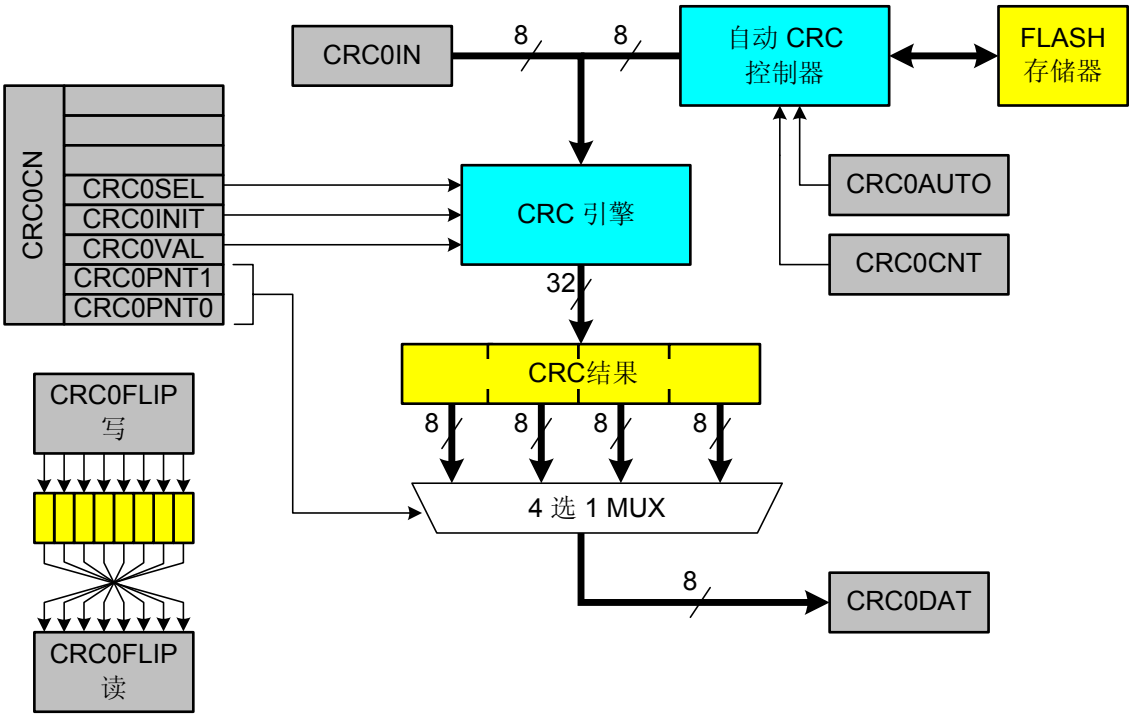


图 15.1 CRC0 框图

15.1 CRC 算法

C8051F93x-C8051F92x 的 CRC 单元产生与下面的算法等价的 CRC 结果：

第一步：将输入与当前 CRC 结果的高字节进行异或（XOR）运算。如果这是 CRC 单元的第一次迭代运算，则当前 CRC 值是被设置的初始值（0x00000000，或 0xFFFFFFFF）。

第二步 a：如果 CRC 值的最高位（MSB）被置 1，对 CRC 结果移位并将移位后的结果与所选多项式进行异或运算。

第二步 b：如果 CRC 值的最高位（MSB）不为 1，则只对 CRC 结果移位。

重复步骤 2a/2b，重复次数为输入比特数（8）。下面的例子也描述了该算法。

16 位 C8051F93x-C8051F92x CRC 算法可以用下面的代码描述：

```
unsigned short UpdateCRC (unsigned short CRC_acc, unsigned char CRC_input)
{
        unsigned char i;                    // loop counter
        #define POLY 0x1021
```

```
    // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
    // with no carries)
    CRC_acc = CRC_acc ^ (CRC_input << 8);

    // "Divide" the poly into the dividend using CRC XOR subtraction
    // CRC_acc holds the "remainder" of each divide
    //
    // Only complete this division for 8 bits since input is 1 byte
    for (i = 0; i < 8; i++)
    {
        // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
        // into the "dividend")
        if ((CRC_acc & 0x8000) == 0x8000)
        {
            // if so, shift the CRC value, and XOR "subtract" the poly
            CRC_acc = CRC_acc << 1;
            CRC_acc ^= POLY;
        }
        else
        {
            // if not, just shift the CRC value
            CRC_acc = CRC_acc << 1;
        }
    }
    // Return the final remainder (CRC value)
    return CRC_acc;
}
```

下表列出了几组输入值和使用 16 位 C8051F93x-C8051F92x CRC 算法所对应的输出值。

表 15.1 16 位 CRC 输出示例

输入	输出
0x63	0xBD35
0x8C	0xB1F4
0x7D	0x4ECA
0xAA, 0xBB, 0xCC	0x6CF6
0x00, 0x00, 0xAA, 0xBB, 0xCC	0xB166

15.2 CRC 计算前的准备

为使 CRC0 准备好进行 CRC 计算，软件应选择所希望的多项式和设置结果的初始值。有两个多项式可用：0x1021（16 位）和 0x04C11DB7（32 位）。CRC0 结果可以被初始化为两个值之一：0x00000000 或 0xFFFFFFFF。可用下面的步骤初始化 CRC0：

第一步：选择多项式（CRC0SEL 为 0 时选择 32 位，为 1 时选择 16 位）。

第二步：选择结果的初始值（设置 CRC0VAL 为 0 时选择 0x00000000，为 1 时选择 0xFFFFFFFF）。

第三步：设置结果为其初始值（向 CRC0INIT 写 1）。

15.3 执行 CRC 计算

一旦 CRC0 被初始化，即可按顺序向 CRC0IN 写入数据流，每次写一个字节。每写入一个字节后，CRC0 结果被自动更新。CRC 引擎也可以被配置为对一个或多个 FLASH 扇区自动执行 CRC。下面的步骤可用于自动对 FLASH 存储器执行 CRC。

第一步：按上述步骤使 CRC0 准备好计算 CRC。

第二步：将起始页索引写入到 CRC0AUTO。

第三步：将 CRC0AUTO 中的 AUTOEN 位置 1。

第四步：将要执行 CRC 计算的 FLASH 扇区数写入 CRC0CNT。

注：每个 FLASH 扇区为 1024 字节。

第五步：向 CRC0CN 写入任何值（或者将其内容与 0x00 相或）以启动 CRC 计算。

第六步：将 CRC0AUTO 中的 AUTOEN 位清 0。

第七步：用下述步骤读 CRC 结果。

15.4 访问 CRC 结果

内部 CRC0 结果为 32 位（CRC0SEL = 0b）或 16 位（CRC0SEL = 1b）。CRC0PNT 位选择对 CRC0DAT 进行读或写操作时的目标字节，并在每次读或写后加 1。计算结果将一直保持在内部 CRC0 结果寄存器中，直到结果寄存器被重新设置、覆盖或有新数据写入 CRC0IN。

SFR 定义 15.1　CRC0CN：CRC0 控制寄存器

SFR 页：0x0F
SFR 地址：0x92

R/W	R/W	R/W	R/W	W	R/W	R/W	R/W	复位值
—	—	—	CRC0SEL	CRC0INIT	CRC0VAL	CRC0PNT[1:0]		00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-5：　未用。读 = 000b，写 = 忽略。

位 4：　CRC0SEL：CRC0 多项式选择位
该位选择 CRC0 多项式和结果的长度（32 位或 16 位）。
0：使用 32 位多项式 0x04C11DB7 计算 CRC 结果。
1：使用 16 位多项式 0x1021 计算 CRC 结果。

位 3：　CRC0INIT：CRC0 结果初始化位
向该位写 1 将根据 CRC0VAL 初始化整个 CRC 结果。

位 2：　CRC0VAL：CRC0 设置值选择位
该位选择 CRC 结果的设置值。
0：向 CRC0INIT 写 0 时将 CRC 结果设置为 0x00000000。
1：向 CRC0INIT 写 1 时将 CRC 结果设置为 0xFFFFFFFF。

位 1-0：　CRC0PNT[1:0]：CRC0 结果指针
这两位选择下一次访问 CRC0DAT 时将读/写 CRC 结果的哪一个字节。每次读或写后这两位的值会自动增 1。
CRC0SEL = 0 时：
00：CRC0DAT 访问 32 位 CRC 结果的位 7~0。
01：CRC0DAT 访问 32 位 CRC 结果的位 15~8。
10：CRC0DAT 访问 32 位 CRC 结果的位 23~16。
11：CRC0DAT 访问 32 位 CRC 结果的位 31~24。
CRC0SEL = 1 时：
00：CRC0DAT 访问 16 位 CRC 结果的位 7~0。
01：CRC0DAT 访问 16 位 CRC 结果的位 15~8。
10：CRC0DAT 访问 16 位 CRC 结果的位 7~0。
11：CRC0DAT 访问 16 位 CRC 结果的位 15~8。

SFR 定义 15.2　CRC0IN：CRC0 数据输入寄存器

SFR 页：0x0F
SFR 地址：0x93

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	

位 7-0：　CRC0IN[7:0]：CRC0 数据输入
每次写 CRC0IN 时，使写入的数据被计算到现有的 CRC 结果中（根据 15.1 节中描述的 CRC 算法）。

SFR 定义 15.3　CRC0DAT：CRC0 数据输出寄存器

SFR 页：0x0F
SFR 地址：0x91

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	

位 7-0：　CRC0DAT[7:0]：　CRC0 数据输出。
每次读或写 CRC0DAT 时，目标为由 CRC0 结果指针（CRC0CN 中的 CRC0PNT 位）指向的 CRC 结果位。

SFR 定义 15.4　CRC0AUTO：CRC0 自动控制寄存器

SFR 页：0x0F
SFR 地址：0x96

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AUTOEN	保留	CRC0ST[5:0]						01000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　AUTOEN：自动 CRC 计算使能。
当 AUTOEN 被置 1 时，对 CRC0CN 的任何写操作都将启动一次对 FLASH 存储器的自动 CRC。FLASH 的起始扇区由 CRC0ST 指定，扇区个数由 CRC0CNT 指定。

位 6：　保留。自动 CRC 计算完成。
当 CRC 计算正在进行时该位为 0。注意：在 CRC 计算期间，代码停止执行，所以从固件中读该位时总是返回 1。

位 5-0：　CRC0ST[5:0]：自动 CRC 计算起始扇区。
这些位指定自动 CRC 计算的起始 FLASH 扇区。自动 CRC 计算所包含的第一个 FLASH 扇区的起始地址为 CRC0ST×1024。

SFR 定义 15.5　CRC0CNT：CRC0 自动 FLASH 扇区计数

SFR 页：0x0F
SFR 地址：0x97

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
		CRC0CNT[5:0]						00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-6：　未用。读 ＝00b，写 ＝ 忽略。

位 5-0：　CRC0CNT[5:0]：自动 CRC 计算 FLASH 扇区计数。
这些位指定自动 CRC 计算包含的 FLASH 扇区数。自动 CRC 计算所包含的最后一个 FLASH 扇区的起始地址为 (CRC0ST+CRC0CNT) × 1024。

15.5 CRC0 的位序反转功能

CRC0 包含能使一个字节中每一位的位序反转的硬件，如图 15.2 所示。写入到 CRC0FLIP 的每个字节数据在读出时都是位反转的。例如，如果将 0xC0 写入 CRC0FLIP，则读回的数据是 0x03。位反转在一些算法（如 FFT）中是一种很有用的数学功能。

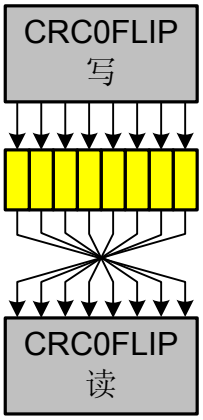


图 15.2 位序反转寄存器

SFR 定义 15.6 CRC0FLIP：CRC0 位反转寄存器

SFR 页：0x0F
SFR 地址：0x94

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	

位 7-0： CRC0FLIP[7:0]：CRC 位反转数据位
写入到 CRC0FLIP 的任何字节被读出时都是位序反转的。即写入时的 LSB 在读出时变成 MSB。例如：

如果将 0xC0 写入 CRC0FLIP，则读回的数据是 0x03。
如果将 0x05 写入 CRC0FLIP，则读回的数据是 0xA0。

16. 片上 DC-DC 转换器（DC0）

C8051F93x-C8051F92x器件内部包含一个DC-DC转换器，允许由一节电池供电，供电电压可低至0.9V。DC-DC转换器是一个开关升压型转换器，其输入电压范围为0.9V ~ 1.8V，输出电压可编程，范围为1.8V ~ 3.3V。缺省状态下的输出电压为1.9V。输入电压必须至少低于输出电压0.2V。DC-DC转换器可以为系统提供最高65mW的稳压电源，该稳压电源可以为系统中的其他器件供电。一般来说，传感器或其他模拟信号需要一个比单节电池所能提供的电压更高的电源电压，因此这一功能为与这类模拟信号接口提供了很大的方便。

图16.1示出了DC-DC转换器的原理框图。在正常工作方式的开关周期的前半个周期，占空比控制开关闭合，二极管旁路开关断开。由于输出电压高于DCEN引脚的电压，所以没有电流通过二极管，负载由输出电容供电。在该阶段，DCEN通过占空比控制开关连接到地，在电感上产生一个正电压并使电感的电流上升。

在开关周期的后半个周期，占空比控制开关断开，二极管旁路开关闭合。这样，DCEN引脚直接连接到VDD/DC+，使电感电流为输出电容充电。一旦电感将其储存的能量传送给输出电容，则占空比控制开关闭合，二极管旁路开关断开，开关周期重新开始。

DC-DC转换器有一个内建电压基准和振荡器，会在输出电压增加到超过编程的目标值时自动限制或关闭开关活动周期。这就允许DC-DC转换器的输出可以被一个辅助电源（当可用时）过驱动，以保持电池寿命。DC-DC转换器的设置可以用SFR寄存器修改，可以改变目标输出电压、振荡器频率或时钟源、二极管旁路开关阻抗、峰值电感电流以及最小占空比。

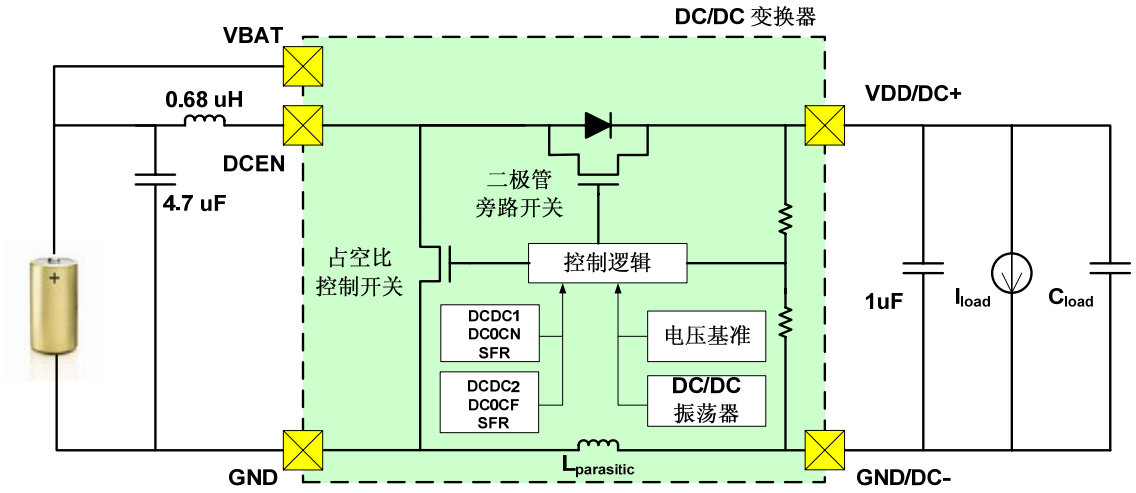


图 16.1　DC-DC 转换器原理框图

16.1 启动表现

DC-DC转换器要求目标输出电压比输入电压至少高0.2V，以维持输出电压的正常调整。在上电之初，DC-DC转换器输出一个恒定的50%占空比，直到输出电容上有足够的电压维持输出调整。输出电容的大小和启动期间负载电流的大小决定输出电容的充电时间。

在初始上电复位期间，触发过流保护的最大峰值电感电流阈值被设置为约125 mA。这将产生一个“软启动”，限制输出电压的上升速度，防止向输出电容流入过大的电流。为了保证DC-DC转换器的可靠启动，需要加以下面的限制：

- 表4.14中给出了启动期间所允许的最大直流负载电流。如果DC-DC转换器通过VDD/DC+或GPIO引脚为外部传感器或器件供电，则供给这些传感器或器件的电流也要计算到这个极限值中。流入电容中的电流不计算到这个极限值中。
- 表4.14中给出了最大总输出电容。该值包含所要求的1 μF陶瓷输出电容和任何连接到VDD/DC+引脚的其他电容。

一旦初始上电结束，软件可以使用DC0CF寄存器将峰值电感电流阈值增加到500 mA。峰值电感电流取决于多个因素（包括直流负载电流），可以用下面的方程估算：

$$I_{PK} = \frac{2 \times I_{LOAD}(VDD/DC + -VBAT)}{\text{效率} \times \text{电感} \times \text{频率}}$$

效率 = 80%

电感 = 0.68 μH

频率 = 2.4 MHz

16.2 脉冲跳过方式

在为了维持输出的正常调整，需要将占空比降到某个宽度以下时，DC-DC转换器允许用户设置最小的脉冲宽度，甚至可以跳过一个完整的脉冲。

16.3 使能 DC-DC 转换器

在上电复位时，DCEN引脚的状态被采样，以确定器件是工作在单节电池模式还是双节电池模式。在双节电池模式，DC-DC转换器总是被禁止。在单节电池模式，DC-DC转换器在休眠方式被禁止，在所有其他电源方式被使能。有关可用的电源方式的详细信息见“14. 电源管理”。

DC-DC转换器由硬件连接使能（单节电池模式），通过在DCEN和VBAT之间放置一个0.68μH的电感实现。通过将DCEN与GND直接短接来禁止DC-DC转换器（双节电池模式）。DCEN引脚不允许浮空。注意，器件只能在上电复位期间在单节和双节电池模式之间切换。有关复位行为的详细信息见“18. 复位源”。

图16.2示出了这两种DC-DC转换器配置选项。

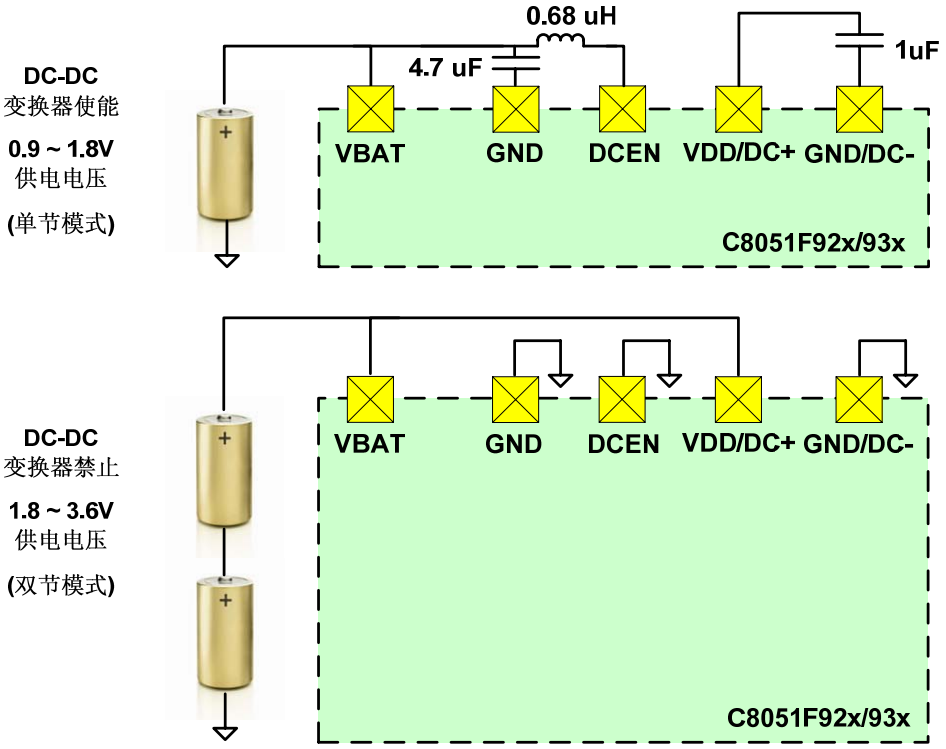


图 16.2　DC-DC 转换器配置选项

当选择DC-DC转换器的“使能”配置（单节电池模式）时，请遵循下面的指南：

- GND/DC-引脚不应在外部连接到GND。
- 0.68μH的电感应放在离DCEN引脚尽可能近的位置，以获得最大频率。
- 4.7μF的电容应放在离电感尽可能近的位置。
- 包括GND、4.7μF电容、0.68μH电感和DCEN引脚的电流回路应尽可能短。
- 连接VDD/DC+与输出电容的PCB线和连接输出电容与GND/DC-的PCB线应尽可能宽，以减小寄生电感。

16.4 DC-DC 转换器寄存器说明

下面介绍用于配置DC-DC转换器的特殊功能寄存器。在大多数系统中，可以使用这些寄存器的复位值，因此不需要软件干预或初始化。

SFR 定义 16.1　DC0CN: DC-DC 转换器控制寄存器

SFR 页：0x00
SFR 地址：0x97

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
MINPW1	MINPW0	SWSEL	保留	SYNC	VSEL2	VSEL1	VSEL0	00100001
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-6：　MINPW[1:0]：DC-DC 转换器最小脉冲宽度控制位
指定最小脉冲宽度。
00：无最小占空比。
01：最小脉冲宽度为 20ns。
10：最小脉冲宽度为 40ns。
11：最小脉冲宽度为 80ns。

位 5：　SWSEL：DC-DC 转换器开关选择位
在两个可能的转换器开关尺寸中选择其一，以使效率最大化。
0：选择大尺寸开关（在输出电流大于 3mA 时效率最高）。
1：选择小尺寸开关（在输出电流等于或小于 3mA 时效率最高）。

位 4：　保留。总是写 0。

位 3：　SYNC：ADC0 同步使能位
0：ADC 不与 DC-DC 转换器同步。
1：ADC 与 DC-DC 转换器同步。在 DC-DC 转换器的最长安静时间内进行 ADC0 跟踪。

位 2-0：　VSEL[2:0]：DC-DC 转换器输出电压选择位
指定目标输出电压。
000：目标输出电压为 1.8V。
001：目标输出电压为 1.9V。
010：目标输出电压为 2.0V。
011：目标输出电压为 2.1V。
100：目标输出电压为 2.4V。
101：目标输出电压为 2.7V。
110：目标输出电压为 3.0V。
111：目标输出电压为 3.3V。

SFR 定义 16.2　DC0CF: DC-DC 转换器配置寄存器

SFR 页：0x00
SFR 地址：0x96

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	CLKDIV1	CLKDIV0	CLKSKW	CLKINV	ILIMIT	VDDSLP	CLKSEL	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　未用。读 ＝0b，写 ＝ 忽略。

位 6-5：　CLKDIV[1:0]：DC-DC 时钟分频系数
当选择系统时钟为 DC-DC 转换器的时钟源时，这些位控制 DC-DC 转换器的分频系数。当 DC-DC 转换器由其本地振荡器提供时钟时，这些位被忽略。
00：DC-DC 转换器时钟分频系数为 1。
01：DC-DC 转换器时钟分频系数为 2。
10：DC-DC 转换器时钟分频系数为 4。
11：DC-DC 转换器时钟分频系数为 8。

位 4：　CLKSKW：DC-DC 转换器时钟偏斜控制
偏斜 DC-DC 转换器时钟。
0：DC-DC 转换器时钟不被偏斜。
1：DC-DC 转换器时钟被偏斜 TBD ns。

位 3：　CLKINV：DC-DC 转换器时钟反向控制
反向 DC-DC 转换器时钟
0：DC-DC 转换器时钟不反向。
1：DC-DC 转换器时钟被反向。

位 2：　ILIMIT：峰值电流极限阈值控制
设置所允许的最大峰值电感电流阈值。
0：峰值电感电流被限制到 125mA。
1：峰值电感电流被限制到 500mA。

位 1：　VDDSLP：VDD/DC+休眠方式连接
在 DC-DC 转换器被使能时，指定休眠方式下的 VDD/DC+电源连接。
0：在休眠方式下，VDD/DC+连接到 VBAT。
1：在休眠方式下，VDD/DC+浮空。

位 0：　CLKSEL：DC-DC 转换器时钟源选择
指定 DC-DC 转换器时钟源。
0：DC-DC 转换器由其本地振荡器提供时钟。
1：DC-DC 转换器由系统时钟提供时钟。

16.5 DC-DC 转换器技术指标

有关DC-DC转换器的详细技术指标见“4. 电气特性”。

17. 稳压器（VREG0）

C8051F93x-C8051F92x器件内部包含一个稳压器（REG0），用于从一个1.8 ~ 3.6V的VDD/DC+电源电压稳压得到1.8V的内核电源。也可以使用一个外部稳压器直接给CPU核及其他数字逻辑供电。该方式通过REG0CN（SFR定义17.1）寄存器设置。片内稳压器的电气特性见电气特性一章。

如果使用一个外部稳压器直接给CPU核及其他数字逻辑供电，可用BYPASS位将内部稳压器置于旁路方式。**除非使用一个外部1.8V稳压器给VDD/DC+供电，否则不应将内部稳压器置于旁路方式。如果这样做，会导致器件的永久性损坏。**

REG0CN寄存器允许精准振荡器偏置电路被禁止，可以在所有非休眠电源方式节省大约15μA。该偏置电路只能在不使用精准振荡器时被禁止。

当器件进入休眠方式时，内部稳压器（REG1）被禁止；当器件进入挂起方式时，内部稳压器保持使能状态。有关低功耗方式的详细信息见“14. 电源管理”。

SFR 定义 17.1　REG0CN: DC-DC 稳压器控制寄存器

SFR 页：0x00
SFR 地址：0xC9

R	R/W	R/W	R/W	R	R	R	R/W	复位值
—	BYPASS	保留	OSCBIAS	—	—	—	保留	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　未用。读 = 0b，写 = 忽略。
位 6：　BYPASS：内部稳压器旁路控制
当置 1 时，内核电源由 VDD/DC+直接供电。
当清 0 时（缺省状态），内核电源是经过稳压的 1.8V 电压。
位 5：　保留。读 = 0b，必须写 0b。
位 4：　OSCBIAS：精准振荡器偏压控制
当置 1 时，精准振荡器使用的偏压被强制使能。如果不使用精准振荡器，该位可以被清 0，这样可以在所有非休眠电源方式节省大约 15μA 的供电电流。如果在禁止后重新使能，精准振荡器需要 4μs 的稳定时间。
位 3-1：　未用。读 = 000b，写 = 忽略。
位 0：　保留。读 = 0b，必须写 0b。

17.1 稳压器技术指标

有关稳压器的详细电气指标见“4. 电气特性”。

18. 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受影响。只要不掉电，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。

端口 I/O 锁存器的复位值为 0xFF（全部为逻辑 1），处于漏极开路方式。在复位期间和复位之后弱上拉被使能。对于 VDD 监视器和上电复位，/RST 引脚被驱动为低电平，直到器件退出复位状态。

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“19. 时钟源”。看门狗定时器被使能，使用系统时钟的 12 分频作为其时钟源（有关使用看门狗定时器的详细信息见“26.4 看门狗定时器方式”）。程序从地址 0x0000 开始执行。

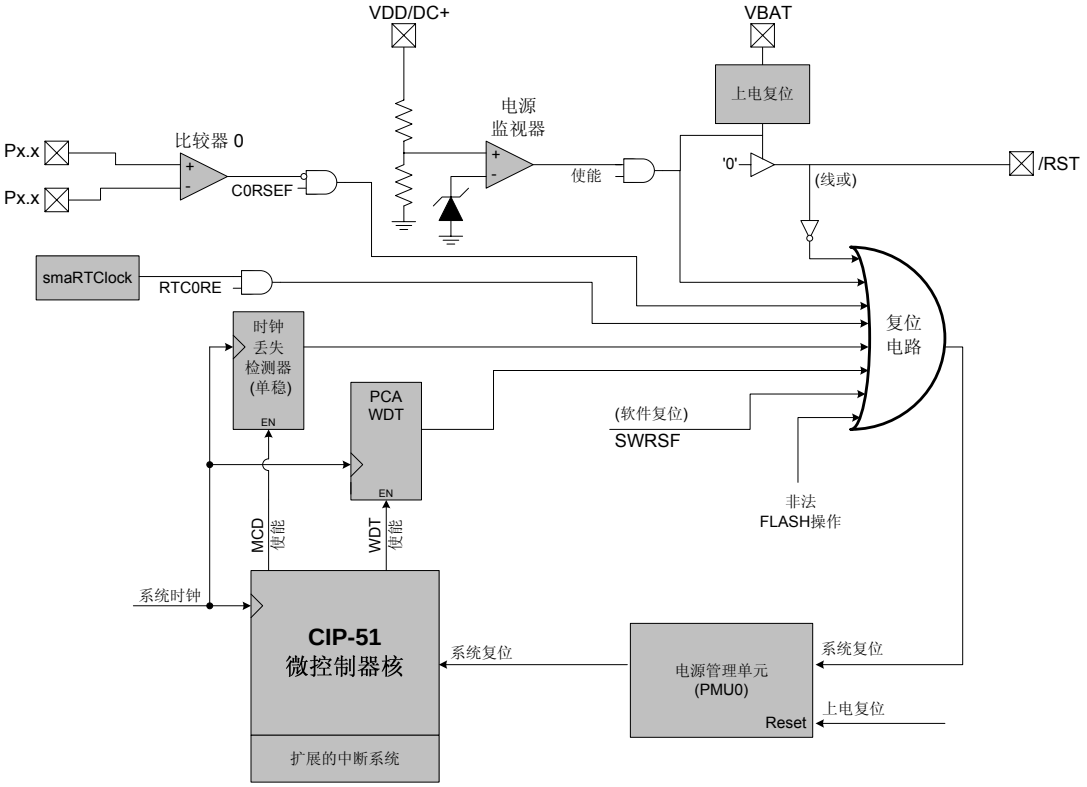


图 18.1 复位源框图

18.1 上电（VBAT 电源监视器）复位

在上电期间，器件保持在复位状态，/RST 引脚被驱动到低电平，直到 VBAT 稳定在超过 V_{POR} 的电平。从复位开始到退出复位状态要经过一个延时；该延时随着 VBAT 上升时间的增大而减小（VBAT 上升时间被定义为 VBAT 从 0V 上升到 V_{POR} 的时间）。图 18.3 给出了上电和 VBAT 监视器复位的时序。对于有效的上升时间（小于 3ms），上电复位延时（$T_{PORDelay}$）通常为 1~3ms。

注：最大的 VBAT 上升时间为 3ms；上升时间超过该最大值时可能导致器件在 VBAT 达到 V_{POR} 电平之前退出复位状态。

在退出复位状态时，PORSF 标志（RSTSRC.1）被硬件置为逻辑 1。当 PORSF 标志被置位时，RSTSRC 寄存器中的所有其它复位标志都是不确定的（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产生的复位。在一次上电复位后，内部数据存储器中的内容应被认为是不确定的。

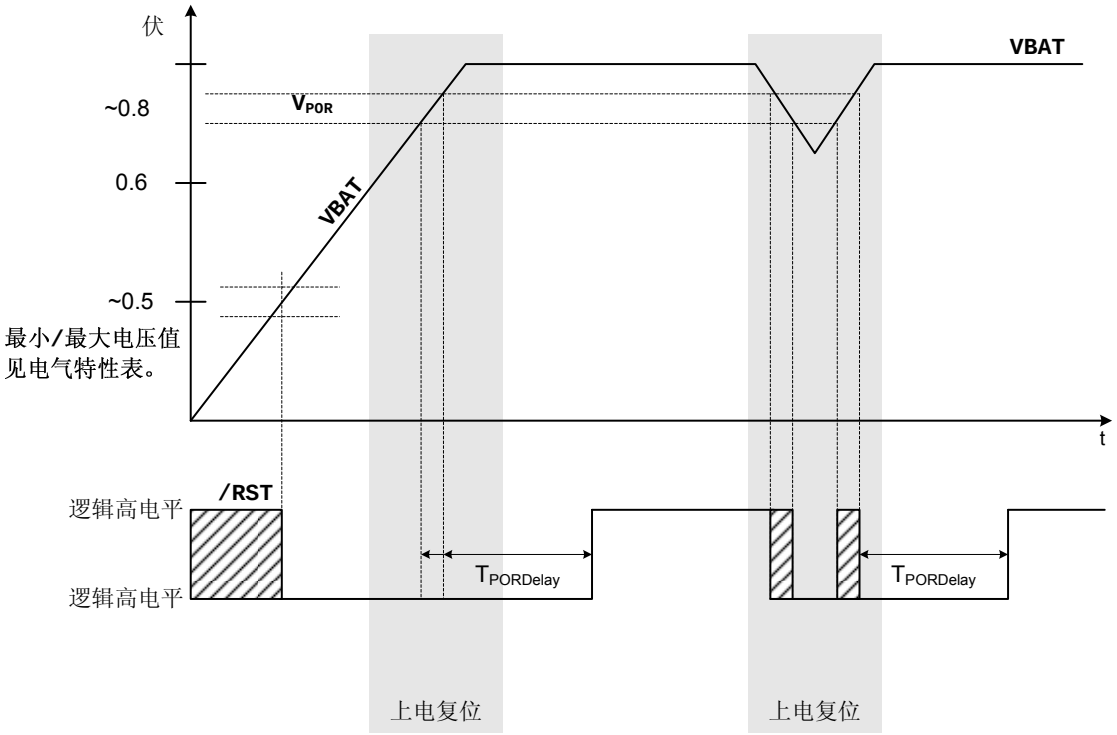


图 18.2 上电和 VBAT 监视器复位时序

18.2 掉电（VDD/DC+电源监视器）复位

C8051F93x-C8051F92x 器件内部有一个 VDD/DC+电源监视器，在每次上电或掉电复位后都被使能和选择为复位源。当被使能和选择为复位源时，发生掉电或因电源波动导致 VDD/DC+降到 V_{RST} 以下时，电源监视器将/RST 引脚驱动为低电平并使 CIP-51 保持复位状态（见图 18.3）。当 VDD/DC+又回到高于 V_{RST} 的电平时，CIP-51 将退出复位状态。

在发生掉电复位后，PORSF 标志的读出值为 1，RAM 的内容无效，VDD/DC+电源监视器被使能和选择为复位源。VDD/DC+电源监视器的使能状态和是否被选择为复位源只能由上电或掉电复位来改变。例如，如果 VDD/DC+电源监视器未被选择为复位源并被软件禁止，则在执行一次软件复位后，VDD/DC+电源监视器仍然保持禁止状态和非复位源设置。

在使用电池工作的系统中，如果器件在掉电复位发生前进入休眠状态，则 RAM 的内容可以一直保持到电池的使用寿命要结束时。当器件处于休眠方式时，掉电复位被自动禁止，只要 VBAT 电源电压不降到 V_{POR} 之下，RAM 的内容将一直保持。当用户更换电池时，可用一个大电容来保持电源电压高于 V_{POR}。当器件从休眠方式被唤醒时，VDD/DC+电源监视器的使能和复位源选择状态被恢复到用户的最后设置值。

为了使软件能早些通告将要发生的掉电，当 VDD/DC+电源电压降到低于 VWARN 阈值时，VDDOK 位被清 0。VDDOK 可以被配置为产生中断。详见“12. 中断系统”。

注意：如果软件包含擦除或写 FLASH 存储器的例程，为了保护 FLASH 内容的完整性，必须将 VDD/DC+电源监视器使能并将其选择为复位源。如果 VDD/DC+电源监视器未被使能，对 FLASH 存储器执行任何擦除或写操作都将导致 FLASH 错误器件复位。

注意：在 VDD/DC+电源监视器复位后，不会引发上电复位延时。有关 VDD/DC+电源监视器的详细电气特性见“4. 电气特性”。

注意：当写 RSTSRC 以使能其他复位源或触发一次软件复位时，软件应谨慎处理，避免意外禁止 VDD/DC+电源监视器为复位源。所有对 RSTSRC 的写操作都应明确地将 PORSF 置 1，以保持 VDD 监视器被使能为复位源。

注意：在选择 VDD/DC+电源监视器作为复位源之前，必须先使能 VDD/DC+电源监视器。在 VDD/DC+电源监视器稳定之前选其为复位源可能导致系统复位。在需要避免产生这种复位的系统中，应在使能 VDD/DC+电源监视器和将其选择为复位源之间引入一个延时。有关最小 VDD/DC+电源监视器启动时间的信息见“4. 电气特性”。**注意：在软件包含擦和写 FLASH 存储器例程的系统中不应引入延时**。使能 VDD/DC+电源监视器和将其选择为复位源的步骤如下：

1. 使能 VDD/DC+电源监视器（VDM0CN 中的 VDMEN 位 = 1）；
2. 等待 VDD/DC+电源监视器稳定（可选项）；
3. 选择 VDD/DC+电源监视器为复位源（RSTSRC 中的 PORSF 位 = 1）。

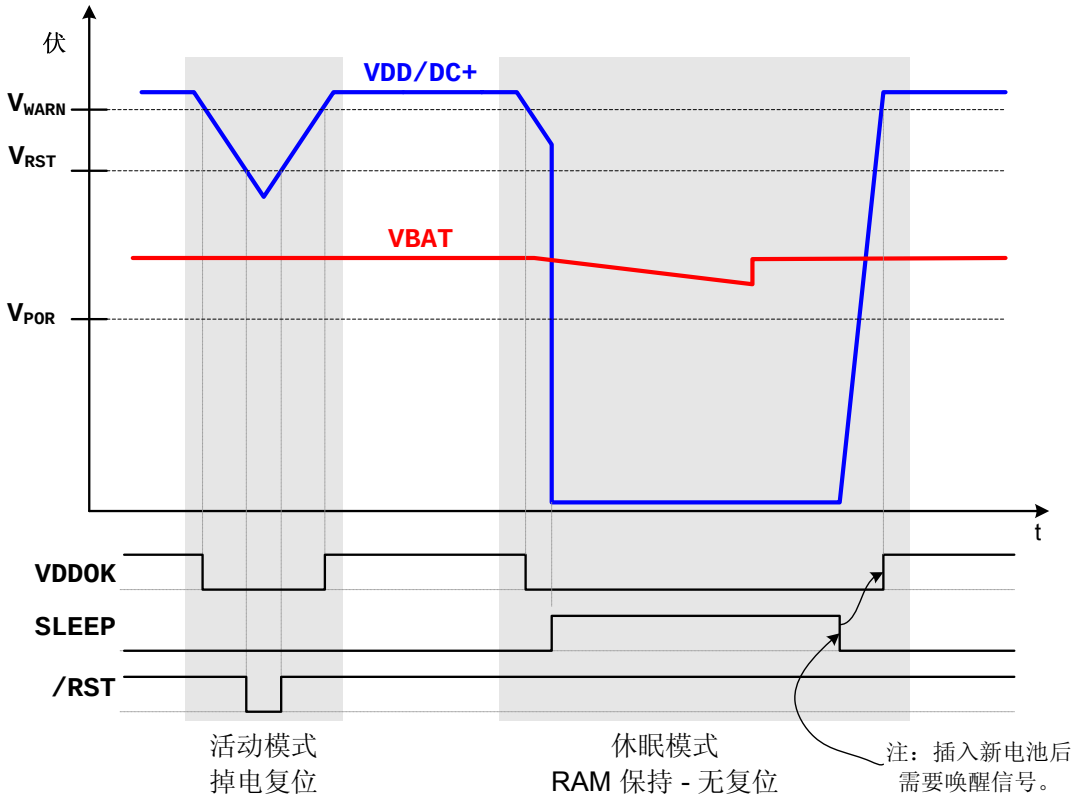

伏
VDD/DC+
VWARN
VRST
VBAT
VPOR
t
VDDOK
SLEEP
/RST
活动模式
掉电复位
休眠模式
RAM 保持 - 无复位
注：插入新电池后
需要唤醒信号。

图 18.3 掉电复位时序

SFR 定义 18.1　VDM0CN：VDD/DC+电源监视器控制寄存器

SFR 页：0x00
SFR 地址：0xFF

R/W	R	R	R/W	R/W	R/W	R/W	R/W	复位值
VDMEN	VDDSTAT	VDDOK	—	—	—	—	—	1vv00000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　VDMEN：VDD/DC+电源监视器使能位。
该位控制 VDDDC+电源监视器电源的通断。VDD/DC+电源监视器在被选择为复位源之前不可能产生系统复位。
0：禁止 VDD/DC+电源监视器。
1：使能 VDD/DC+电源监视器。

位 6：　VDDSTAT：VDD/DC+电源状态。
该位指示当前电源状态。
0：VDD/DC+等于或低于 V_{RST} 阈值。
1：VDD/DC+高于 V_{RST} 阈值

位 5：　VDDOK：VDD/DC+电源状态（早期告警）
该位指示当前电源状态
0：VDD/DC+等于或低于 V_{WARN} 阈值。
1：VDD/DC+高于 V_{WARN} 阈值

位 4-0：　未用。读 = 00000b，写 = 忽略。

18.3 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产生复位，可能有必要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。/RST 引脚的电气特性见表 4.4。即使在器件处于低功耗挂起或休眠方式时，外部复位仍然起作用。从外部复位状态退出后，PINRSF 标志（RSTSRC.0）被置 1。

18.4 时钟丢失检测器复位

时钟丢失检测器（MCD）是一个由系统时钟触发的单稳态电路。如果系统时钟保持在高电平或低电平的时间大于 100 微秒，单稳态电路将超时并产生复位。在发生 MCD 复位后，MCDRSF 标志（RSTSRC.2）的读出值为 1，表示本次复位源为 MCD；否则该位读出值为 0。向 MCDRSF 位写 1 使能时钟丢失检测器；写 0 禁止时钟丢失检测器。在器件处于低功耗挂起或休眠方式时，时钟丢失检测器复位被自动禁止。在从任何一个低功耗状态退出时，该复位源的使能/禁止状态被恢复为此前的值。/RST 引脚的状态不受该复位的影响。

18.5 比较器 0 复位

向 C0RSEF 标志（RSTSRC.5）写 1 可以将比较器 0 配置为复位源。应在写 C0RSEF 之前使能比较器 0 并等待输出稳定，以防止通电瞬间在输出端产生抖动，从而导致不希望的复位。比较器 0 复位为低电平有效：如果同相端输入电压（CP0+）小于反相端输入电压（CP0-），则器件被置于复位状态。在发生比较器 0 复位后，C0RSEF 标志（RSTSRC.5）的读出值为 1，表示本次复位源为比较器 0；否则该位读出值为 0。只要比较器 0 被使能为唤醒源，即使在器件处于低功耗挂起或休眠方式时，比较器 0 复位仍然起作用。/RST 引脚的状态不受该复位的影响。

18.6 PCA 看门狗定时器复位

可编程计数器阵列（PCA）的可编程看门狗定时器（WDT）功能可用于在系统出现错误的情况下防止软件运行失控。可以通过软件使能或禁止 PCA 的 WDT 功能（见“26.4 看门狗定时器方式”）。在每次复位后，WDT 被使能并使用 SYSCLK/12 作为时钟。如果因系统出错使用户软件不能更新 WDT，则 WDT 将产生复位，WDTRSF 位（RSTSRC.5）被置 1。在器件处于低功耗挂起或休眠方式时，PCA 看门狗定时器复位源被自动禁止。在从任何一个低功耗状态退出时，该复位源的使能/禁止状态被恢复为此前的值。/RST 引脚的状态不受该复位的影响。

18.7 FLASH 错误复位

如果 FLASH 读/写/擦除操作的地址或程序读地址为非法地址，将发生系统复位。下述的任何一种情况都会导致 FLASH 操作错误复位：

- FLASH 写或擦除地址超出了用户代码空间。这种情况发生在 PSWE 被置 1，并且 MOVX 写操作的地址大于锁定字节地址时。
- FLASH 读地址超出了用户代码空间，即 MOVC 操作的地址大于锁定字节地址。
- 程序读超出了用户代码空间。这种情况发生在用户代码试图转移到大于锁定字节地址地址时。
- 当 FLASH 读、写或擦除被安全设置禁止时（见“13.3 安全选项”）。
- 当 VDD 监视器被禁止时，试图进行 FLASH 写或擦除操作。

在发生 FLASH 错误复位后，FERROR 位（RSTSRC.6）被置 1。/RST 引脚的状态不受该复位的影响。

18.8 smaRTClock（实时时钟）复位

有两种事件可使 smaRTClock 产生系统复位：smaRTClock 振荡器故障和 smaRTClock 告警。当 smaRTClock 时钟丢失检测器被使能时，如果 smaRTClock 的时钟频率低于约 20 KHz，则会发生 smaRTClock 振荡器故障事件。当 smaRTClock 告警被使能且 smaRTClock 定时器值与 ALARMn 寄存器一致时，会发生 smaRTClock 告警事件。通过向 RTC0RE 位（RSTSRC.7）写 1 来将 smaRTClock 配置为复位源。即使在器件处于低功耗挂起或休眠方式时，smaRTClock 复位仍然起作用。/RST 引脚的状态不受该复位的影响。

18.9 软件复位

软件可以通过向 SWRSF 位（RSTSRC.4）写 1 强制产生一次系统复位。在发生软件强制复位后，SWRSF 位的读出值为 1。/RST 引脚的状态不受该复位的影响。

SFR 定义 18.2　RSTSRC：复位源寄存器

SFR 页：0x00
SFR 地址：0xEF

R/W	R	R/W	R/W	R	R/W	R/W	R	复位值
RTC0RE	FERROR	C0RSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF	可变
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　RTC0RE：smaRTClock（实时时钟）复位使能和标志位
读：如果最后一次复位是由 smaRTClock 告警或振荡器故障引起的，则置 1。
写：
0：禁止 smaRTClock 为复位源。
1：使能 smaRTClock 为复位源。

位 6：　FERROR：FLASH 错误标志。
读：如果最后一次复位是由 FLASH 读/写/擦除错误引起的，则置 1。

位 5：　C0RSEF：比较器 0 复位使能和标志
读：如果最后一次复位是由比较器 0 引起的，则置 1。
写：
0：禁止比较器 0 为复位源。
1：使能比较器 0 为复位源。

位 4：　SWRSF：软件强制复位和标志
读：如果最后一次复位是由写 SWRSF 位引起的，则置 1。
写：写 1 强制产生一次系统复位。

位 3：　WDTRSF：看门狗定时器复位标志
读：如果最后一次复位是由 WDT 超时引起的，则置 1。

位 2：　MCDRSF：时钟丢失检测器（MCD）使能和标志
读：如果最后一次复位是由时钟丢失检测器超时引起的，则置 1。
写：
0：禁止 MCD。
1：使能 MCD；MCD 在检测到时钟丢失条件时触发复位。

位 1：　PORSF：上电/掉电复位标志和掉电复位使能
读：任何时刻发生上电复位或 VDD/DC+监视器复位时，该位置 1[2]。
写：
0：禁止 VDD/DC+电源监视器为复位源。
1：使能 VDD/DC+电源监视器为复位源[3]。

位 0：　PINRSF：硬件引脚复位标志
读：如果最后一次复位是由/RST 引脚引起的，则置 1。

注：
1. 使用读-修改-写操作（ORL、ANL 等）使能或禁止特定中断源是安全的。
2. 如果 PORSF 的读出值为 1，则该寄存器中所有其他位都是不确定的。
3. 在 VDD/DC+电源监视器稳定之前向 PORSF 写 1 可能产生系统复位。

19. 时钟源

C8051F93x-C8051F92x 器件有一个可编程精准内部振荡器、一个外部振荡器驱动电路、一个低功耗内部振荡器和 smaRTClock 实时时钟振荡器。可以通过 OSCICN 和 OSCICL 寄存器来使能/禁止内部振荡器和调节其输出频率（如图 19.1 所示）。外部振荡器可以通过 OSCXCN 寄存器配置。低功耗内部振荡器在被选择为时钟源和解除选择时自动使能和禁止。smaRTClock 的操作在 smaRTClock 振荡器一章描述。

系统时钟（SYSCLK）可以由精准内部振荡器、外部振荡器、低功耗内部振荡器或 smaRTClock 振荡器提供。全局时钟分频器可以产生比所选择的时钟源低 1、2、4、8、16、32、64 或 128 倍的系统时钟。振荡器的电气指标见电气特性一章。

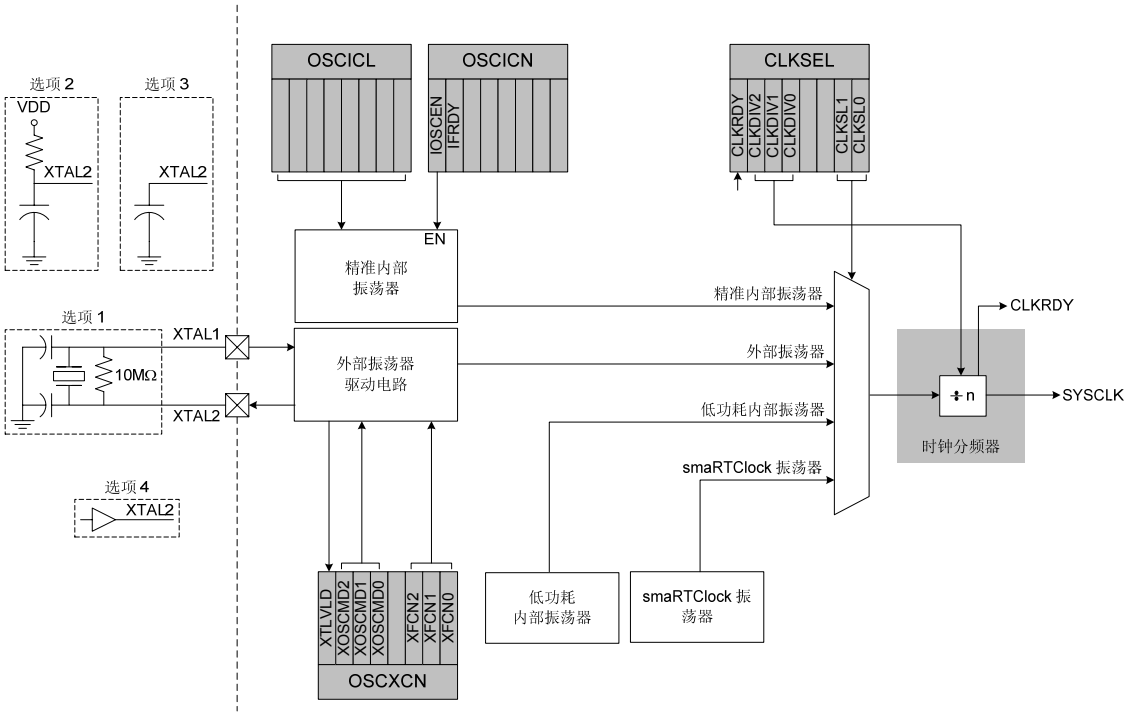


图 19.1 振荡器框图

19.1 可编程精准内部振荡器

所有C8051F93x-C8051F92x器件都包含一个可编程精准内部振荡器，该振荡器可以被选择作为系统时钟。OSCICL寄存器已经过工厂校准，对应的频率为24.5 MHz。有关振荡器的详细技术指标见“4. 电气特性”。

该精准振荡器支持扩频模式，对输出频率进行调制以降低系统产生的 EMI。当扩频模式被使能时（SSE = 1），振荡器输出频由一个步进三角波调制，三角波的频率等于振荡器频率除以 384（使用出厂校准值时为 63.8 KHz）。与中心频率的最大偏差为±0.75%。输出频率每 32 个周期更新一次，步长为中心频率的 0.25%。

19.2 低功耗内部振荡器

所有C8051F93x-C8051F92x器件都包含一个低功耗内部振荡器，系统复位后该振荡器被默认为系统时钟。低功耗内部振荡器的频率为20 MHz±10%，当被选择为系统时钟时被自动使能，在不使用时被自动禁止。有关振荡器的详细技术指标见“4. 电气特性”。

19.3 外部振荡器驱动电路

所有 C8051F93x-C8051F92x 器件都包含一个外部振荡器电路，可以驱动外部晶体、陶瓷谐振器、电容或 RC 网络。也可以使用一个外部 CMOS 时钟提供系统时钟。图 19.1 给出了四种外部振荡器选项的原理框图。使用 OSCXCN 寄存器来使能和配置外部振荡器。

外部振荡器输出可以被选择作为系统时钟或用于给一些数字外设（如定时器、PCA 等）提供时钟。有关每个数字外设的详细信息，见本手册的相关章节。有关振荡器的详细技术指标见“4. 电气特性”。

19.3.1 外部晶体方式

如果外部振荡器使用晶体或陶瓷谐振器，则必须在 XTAL1 和 XTAL2 引脚跨接一个外部晶体/谐振器和一个 10MΩ的电阻，如图 19.1 中的选项 1 所示。应在 XTAL1 和 XTAL2 引脚加合适的负载电容。XTAL1 和 XTAL2 引脚都应被配置为模拟 I/O，并禁止其数字输出驱动器。

图 19.2 给出了一个使用 20MHz 石英晶体的外部振荡器电路，厂商建议的负载电容为 12.5pF。从晶体来看，这两个电容是“串联”的，但与 XTAL1 和 XTAL2 引脚的寄生电容“并联”。每个负载电容与 XTAL 引脚的寄生电容的总值应为 12.5pF × 2 = 25 pF。如果每个 XTAL 引脚的寄生电容值为 3 pF，则采用 22 pF 的负载电容后跨接在晶体两端的等效电容为 12.5 pF。

注：建议的负载电容取决于晶体的振荡频率和生产厂家。计算负载电容值时请参考晶体的数据手册。

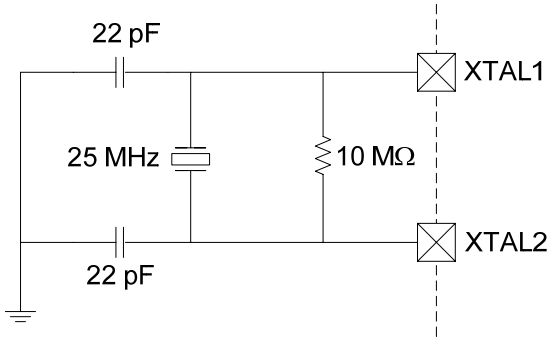


图 19.2　25 MHz 外部晶体示例

注意，晶体振荡器电路对 PCB 布局非常敏感。晶体应尽可能地靠近器件的 XTAL 引脚，布线应尽可能地短并用地平面屏蔽，以防止其它引线引入噪声或干扰。

当使用外部晶体时，必须用软件将外部振荡器驱动电路配置为*晶体振荡器方式*或*晶体振荡器 2 分频方式*。2 分频电路保证从外部振荡器获取的时钟具有 50%的占空比。还必须根据晶体频率确定外部振荡器频率控制值（XFCN），见表 19.1。例如，一个 25MHz 的晶体要求 XFCN 设置值为 111b。

表 19.1 晶体方式下建议的 XFCN 设置

XFCN	晶体频率
000	f ≤ 20kHz
001	20 kHz < f ≤ 58 kHz
010	58 kHz < f ≤ 155 kHz
011	155 kHz < f ≤ 415 kHz
100	415 kHz < f ≤ 1.1 MHz
101	1.1 MHz < f ≤ 3.1 MHz
110	3.1 MHz < f ≤ 8.2 MHz
111	8.2 MHz < f ≤ 25 MHz

在晶体振荡器被使能时，外部振荡器有效检测电路允许软件确定何时可以使用外部时钟。在晶体振荡器稳定之前就切换到外部晶体振荡器可能产生不可预见的后果。启动晶体的建议步骤如下：

第一步：配置 XTAL1 和 XTAL2 为模拟 I/O 并禁止数字输出驱动器。

第二步：配置并使能外部振荡器。

第三步：查询 XTLVLD => 1。

第四步：将系统时钟切换到外部振荡器。

19.3.3 外部 RC 方式

如果使用一个 RC 网络作为外部振荡器，则电路应被配置为如图 19.1 中的选项 2 所示。RC 网络应加到 XTAL2 引脚，应将 XTAL2 配置为模拟 I/O 并禁止其数字输出驱动器。XTAL1 在 RC 方式不使用。

电容不应大于 100pF；但当电容值很小时，PCB 的寄生电容将在总电容中占支配地位。电阻不应小于 10KΩ。振荡频率可由下面的方程确定：

$$f = \frac{1.23 \times 10^3}{R \times C}$$

其中：

f = 时钟频率（以 MHz 为单位）

R = 上拉电阻值（以 KΩ 为单位）

C = XTAL2 引脚上的电容值（以 pF 为单位）

为了确定 OSCXCN 寄存器中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。例如，如果所期望的频率是 100kHz，选 R = 246kΩ 和 C = 50pF：

$$f = \frac{1.23 \times 10^3}{R \times C} = \frac{1.23 \times 10^3}{246 \times 50} = 100kHz$$

查表 19.2，得到建议的 XFCN 设置值为 010b。

表 19.2 RC 和 C 方式下建议的 XFCN 设置

XFCN	大致频率范围 （RC 和 C 方式）	K 因子（C 方式）
000	f ≤ 25 kHz	K 因子 = 0.87
001	25 kHz < f ≤ 50 kHz	K 因子 = 2.6
010	50 kHz < f ≤ 100 kHz	K 因子 = 7.7
011	100 kHz < f ≤ 200 kHz	K 因子 = 22
100	200 kHz < f ≤ 400 kHz	K 因子 = 65
101	400 kHz < f ≤ 800 kHz	K 因子 = 180
110	800 kHz < f ≤ 1.6 MHz	K 因子 = 664
111	1.6 MHz < f ≤ 3.2 MHz	K 因子 = 1590

当 RC 振荡器被使能时，外部振荡器有效检测电路允许软件确定振荡何时稳定。启动 RC 振荡器的建议步骤如下：

第一步：配置 XTAL2 为模拟 I/O 并禁止数字输出驱动器。

第二步：配置并使能外部振荡器。

第三步：查询 XTLVLD => 1。

第四步：将系统时钟切换到外部振荡器。

19.3.4 外部电容方式

如使用一个电容作为外部振荡器，则电路应被配置为如图 19.1 中的选项 3 所示。电容应加到 XTAL2 引脚，应将 XTAL2 配置为模拟 I/O 并禁止其数字输出驱动器。XTAL1 在 C 方式不使用。

电容不应大于 100pF；但当电容值很小时，PCB 的寄生电容将在总电容中占支配地位。振荡频率和 OSCXCN 寄存器中所需用的外部振荡器频率控制值（XFCN）可由下面的方程确定：

$$f = \frac{KF}{C \times V_{DD}}$$

其中：

f = 时钟频率（以 MHz 为单位）

V_{DD} = 电源电压（以 V 为单位）

R = 上拉电阻值（以 KΩ 为单位）

C = XTAL2 引脚上的电容值（以 pF 为单位）

下面是根据振荡频率选择电容的一个例子。假设 V_{DD} = 3.0V，f = 150 KHz：

$$f = \frac{KF}{C \times V_{DD}}$$

$$0.150MHz = \frac{KF}{C \times 3.0}$$

由于所需要的频率大约为 150 kHz，从表 19.2 中选择 K 因子，得到 KF = 22：

$$0.150MHz = \frac{22}{C \times 3.0}$$

$$C = \frac{22}{0.150MHz \times 3.0}$$

C = 48.8 pF

因此，本例中要用的 XFCN 值为 011b，C 大约为 50 pF。

C 方式的建议启动过程与 RC 方式相同。

19.3.4 外部 CMOS 时钟方式

如果使用一个外部 CMOS 时钟作为外部振荡器，则时钟应直接连接到 XTAL2。XTAL2 引脚应被配置为数字输入。XTAL1 在外部 CMOS 时钟方式未使用。

当外部振荡器被配置为外部 CMOS 时钟方式时，外部振荡器有效检测器将总是返回 0 值。

19.4 用于选择和配置系统时钟的特殊功能寄存器

C8051F93x-C8051F92x 器件时钟源的使能和配置使用 OSCICN、OSCICL、OSCXCN 及 smaRTClock 的内部寄存器。对 smaRTClock 寄存器的说明见“20. smaRTClock（实时时钟）”。用 CLKSEL 寄存器选择 MCU 的系统时钟源。为了减小活动方式电流，当系统时钟大于 1MHz 时，设置 FLASH 读时间的单稳定时器应被旁路。详见 FLSCL 寄存器的说明。

系统时钟可以通过对其时钟源进行 1、2、4、8、16、32、64 或 128 分频得到。当在两个分频值之间切换时，需要 128 个未分频时钟源周期。可以查询 CLKRDY 标志来确定新的时钟分频值是否已经正常工作。当进入挂起或休眠方式时，时钟分频器必须被设置为“1 分频”。

可以在系统运行时切换时钟源。这种切换在经过频率较低的振荡器的一个时钟周期后生效。

SFR 定义 19.1　CLKSEL：时钟选择寄存器

SFR 页：所有页
SFR 地址：0xA9

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CLKRDY	CLKDIV[2:0]			—	CLKSEL[2:0]			00110100
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　CLKRDY：系统时钟分频器时钟准备好标志。
0：所选择的时钟分频设置尚未应用于系统时钟。
1：所选择的时钟分频设置尚已应用于系统时钟。

位 6-4：　CLKDIV[2:0]：系统时钟分频位
选择要应用于未分频系统时钟源的时钟分频值。
000：系统时钟 1 分频。
001：系统时钟 2 分频。
010：系统时钟 4 分频。
011：系统时钟 8 分频。
100：系统时钟 16 分频。
101：系统时钟 32 分频。
110：系统时钟 64 分频。
111：系统时钟 128 分频。

位 3：　未用。读 = 0b，必须写 0b。

位 2-0：　CLKSL[2:0]：系统时钟选择位
这些位选择用作未分频系统时钟源系统时钟源的振荡器。

CLKSEL[2:0]	选择的时钟
000	精准内部振荡器
001	外部振荡器
010	保留
011	smaRTClock 振荡器
1xx	低功耗振荡器

SFR 定义 19.2　OSCICN：内部振荡器控制寄存器

SFR 页：0x00
SFR 地址：0xB2

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	复位值
IOSCEN	IFRDY	保留						00001111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　IOSCEN：内部振荡器使能位
0：内部振荡器禁止。
1：内部振荡器使能。

位 6：　IFRDY：内部振荡器频率准备好标志
0：内部振荡器未运行在编程频率。
1：内部振荡器运行在其编程频率。

位 5-0：　保留。读 = 001111b，必须写 001111b。

注：建议使用读-修改-写操作（如 ORL 和 ANL）来置位或清除该寄存器中的使能位。

SFR 定义 19.3　OSCICL：内部振荡器校准寄存器

SFR 页：0x00
SFR 地址：0xB3

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SSE	OSCICL[6:0]							0vvvvvvv
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　SSE：扩频方式使能位
0：扩频时钟抖动禁止。
1：扩频时钟抖动使能。

位 6-0：　OSCICL：内部振荡器校准位
工厂校准值对应 24.5 MHz 的频率。该值增加则振荡器频率降低，该值减小则振荡器频率提高。步长约为校准频率的 1%。

SFR 定义 19.4　OSCXCN：外部振荡器控制寄存器

SFR 页：0x00
SFR 地址：0xB1

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
XCLKVLD	XOSCMD[2:0]			保留	XFCN[2:0]			00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　XCLKVLD：外部振荡器有效标志
提供外部振荡器状态，除了外部 CMOS 时钟方式和外部 CMOS 时钟 2 分频方式外，在所有的时间对所有的工作方式都有效。在外部 CMOS 时钟方式和外部 CMOS 时钟 2 分频方式，XCLKVLD 总是返回 0 值。
0：晶体振荡器未用或未稳定。
1：晶体振荡器稳定运行。

位 6-4：　XOSCMD[2:0]：外部振荡器方式位
将外部振荡器电路配置为所选择的方式。
00x：外部振荡器电路禁止。
010：外部 CMOS 时钟方式。
011：外部 CMOS 时钟方式二分频。
100：RC 振荡器方式。
101：电容振荡器方式。
110：晶体振荡器方式。
111：晶体振荡器方式二分频。

位 3：　保留。读 = 0b，必须写 0b。

位 2-0：　XFCN[2:0]：外部振荡器频率控制位。
控制外部振荡器偏置电流。
000-111：见表 19.1 或表 19.2 中的建议设置值。

20. smaRTClock（实时时钟）

C8051F93x-C8051F92x器件内部有一个功耗极低的32位smaRTClock外设（实时时钟）。smaRTClock有一个专用的32KHz振荡器，可以被配置为使用或不使用晶体，不需要外部电阻或负载电容。片内负载电容是可编程的，有16个级别，允许与很宽范围的晶体兼容。smaRTClock可以直接用0.9~3.6V的电池电压工作，即使在器件进入其最低功耗的掉电方式也仍然能正常工作。

当使用32.768KHz的钟表晶体时，smaRTClock的32位独立计时器允许36小时的最长计时周期。smaRTClock提供告警和smaRTClock时钟丢失事件，这两个事件可以被用作复位源或唤醒源。有关复位源和低功耗方式唤醒源的详细信息分别见“18. 复位源”和“14. 电源管理”。

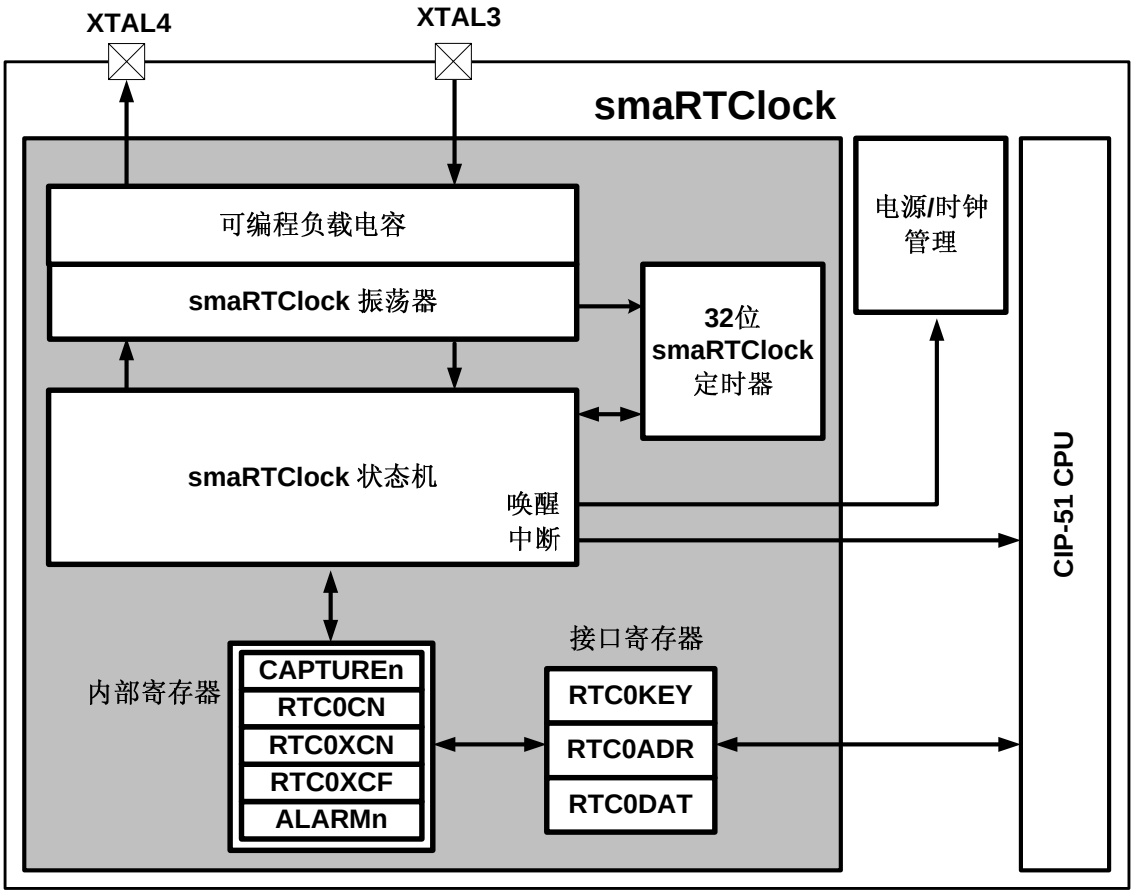


图20.1　smaRTClock原理框图

20.1 smaRTClock 接口

smaRTClock 接口包括三个寄存器：RTC0KEY、RTC0ADR 和 RTC0DAT。这些接口寄存器位于 CIP-51 的 SFR 空间，提供对表 20.1 中列出的 smaRTClock 内部寄存器的访问。smaRTClock 内部寄存器只能通过 smaRTClock 接口间接访问。

表 20.1 smaRTClock 内部寄存器

smaRTClock 地址	smaRTClock 寄存器	寄存器名称	说 明
0x00 ~ 0x03	CAPTUREn	smaRTClock 捕捉寄存器	4 个寄存器。用于设置 32 位的 smaRTClock 定时器或读该定时器的当前值。
0x04	RTC0CN	smaRTClock 控制寄存器	控制 smaRTClock 状态机的工作。
0x05	RTC0XCN	smaRTClock 振荡器配置寄存器	控制 smaRTClock 振荡器的工作。
0x06	RTC0XCF	smaRTClock 振荡器控制寄存器	控制可编程负载电容的值和使能/禁止自动步进功能。
0x07	RTC0PIN	smaRTClock 引脚配置寄存器	强制 XTAL3 和 XTAL4 在内部短接。 注：该寄存器还包含其它保留位，它们的值得是不应被修改的。
0x08 ~ 0x0B	ALARMn	smaRTClock 告警寄存器	4 个寄存器。用于设置或读 32 位的 smaRTClock 告警值。

20.1.1 smaRTClock 锁定和关键码功能

smaRTClock 接口受锁定和关键码功能的保护。在对 RTC0ADR 和 RTC0DAT 进行读或写之前，必须向 smaRTClock 锁定和关键码寄存器（RTC0KEY）按顺序正确地写入关键码。关键码为：0xA5，0xF1。写关键码的时间并不重要，但必须按顺序写。如果写关键码的顺序不对或写入了错误的关键码，或者在接口被锁定时试图读或写间接寄存器，则 smaRTClock 接口将被禁止，RTC0ADR 和 RTC0DAT 变成不可访问，直到下一次系统复位。一旦 smaRTClock 接口被解锁，软件即可访问 smaRTClock 寄存器而没有次数限制，直到接口被重新锁定或器件被复位。在 smaRTClock 接口处于非锁定状态时，对 RTC0KEY 的任何一次写操作都将重新锁定接口。

在任何时刻读 RTC0KEY 寄存器都可提供 smaRTClock 接口的状态，不会影响写入操作。SFR 定义 20.1 中的 RTC0KEY 寄存器说明列出了每个状态码的定义。

20.1.2 使用 RTC0ADR 和 RTC0DAT 访问 smaRTClock 的内部寄存器

smaRTClock 内部寄存器可以用 RTC0ADR 和 RTC0DAT 进行读和写。RTC0ADR 寄存器选择后面要读或写的 smaRTClock 内部寄存器。在每次读或写之前，应先检查 BUSY（RTC0ADR.7），以确保 smaRTClock 接口不在进行另一次读或写操作。通过写 RTC0DAT 寄存器来启动一次 smaRTClock 写操作。下面是写 smaRTClock 内部寄存器的一个例子。

第一步：查询 BUSY（RTC0ADR.7）直到其返回 0 值，或插入 6 个系统时钟周期的延时。
第二步：向 RTC0ADR 写 0x05。该操作选择位于 smaRTClock 地址 0x05 的内部寄存器 RTC0CN。
第三步：向 RTC0DAT 写 0x00。该操作向内部 RTC0CN 寄存器写 0x00。

通过将 smaRTClock 接口的 BUSY 位置 1 来启动一次 smaRTClock 读操作。该操作将由 RTC0ADR 选择的内部寄存器的内容传送到 RTC0DAT。传送的数据将一直保持在 RTC0DAT 中，直到下一次读或写操作发生。下面是读 smaRTClock 内部寄存器的一个例子。

第一步：查询 BUSY（RTC0ADR.7）直到其返回 0 值，或插入 4 个系统时钟周期的延时。
第二步：向 RTC0ADR 写 0x05。该操作选择位于 smaRTClock 地址 0x05 的内部寄存器 RTC0CN。
第三步：向 BUSY 写 1。该操作启动从 RTC0CN 到 RTC0DAT 的数据传送。
第四步：查询 BUSY（RTC0ADR.7）直到其返回 0 值，或插入 4 个系统时钟周期的延时。
第五步：从 RTC0DAT 读取数据。该数据是 RTC0CN 寄存器的拷贝。

注：假设短选通被使能。如果短选通被禁止，则应插入 5 个系统时钟周期的延时。

注：RTC0ADR 和 RTC0DAT 寄存器将保持其状态，直到器件复位。

20.1.3 smaRTClock 接口的自动读功能

当自动读被使能时，每次读 RTC0DAT 都会启动下一次对 smaRTClock 内部寄存器（由 RTC0ADR 选择）的间接读操作。软件只应在每次连续读操作的开始将 BUSY 位置 1。软件在读 RTC0DAT 之前必须检查 smaRTClock 接口是否忙。通过将 AUTORD（RTC0ADR.6）置 1 来使能自动读功能。

20.1.4 RTC0ADR 自动增 1 功能

为了便于读和写 32 位的 CAPTURE 和 ALARM 值，RTC0ADR 具有在每次读/写一个 CAPTUREn 或 ALARMn 寄存器之后自动增 1 的功能。该功能加速了设置一个告警值或读取 smaRTClock 定时器值的过程。自动增 1 功能总是被使能。

20.1.5 RTC0ADR 短选通功能

读和写间接 smaRTClock 寄存器一般需要 7 个系统时钟。为了缩短间接寄存器的访问时间，可以使用短选通功能，该功能将读和写访问时间缩短为 6 个系统时钟。短选通功能在器件复位后被自动使能，也可以使用 SHORT（RTC0ADR.4）控制位手动使能/禁止该功能。

SFR 定义 20.1　RTC0KEY：smaRTClock 锁定和关键码寄存器

SFR 页：0x00
SFR 地址：0xAE

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
RTC0ST[7:0]								00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　RTC0ST[7:0]：smaRTClock 接口锁定/关键码和状态
写入操作锁定/解锁 smaRTClock 接口。读操作提供锁定状态。

读：

0x00：smaRTClock 接口被锁定。

0x01：smaRTClock 接口被锁定，第一个关键码（0xA5）已被写入，等待第二个关键码。

0x02：smaRTClock 接口被解锁，第一和第二个关键码（0xA5，0xF1）已被写入。

0x03：smaRTClock 接口被禁止，直到下一次系统复位。

写：

当 RTC0ST = 0x00 时（锁定），写 0xA5 后再写 0xF1 将解锁 smaRTClock 接口。

当 RTC0ST = 0x01 时(等待第二个关键码),写任何不同于第二个关键码(0xF1)的值将使 RTC0ST 变为 0x03，smaRTClock 接口被禁止，直到下一次系统复位。

当 RTC0STATE = 0x02 时（解锁），任何对 RTC0KEY 的写操作将锁定 smaRTClock 接口。

当 RTC0STATE = 0x03 时（禁止），对 RTC0KEY 的写操作不起作用。

SFR 定义 20.2　RTC0ADR：smaRTClock 地址寄存器

SFR 页：0x00
SFR 地址：0xAC

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	复位值
BUSY	AUTORD	—	SHORT	ADDR[3:0]				00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　BUSY：smaRTClock 接口忙指示位
指示 smaRTClock 接口的状态。向该位写 1 启动一次 smaRTClock 间接读操作。
读：
0：smaRTClock 不忙。
1：smaRTClock 忙于执行一次读或写操作。

位 6：　AUTORD：smaRTClock 接口自动读使能
使能/禁止自动读。
0：禁止自动读。
1：使能自动读。

位 5：　未使用。读 =0x0b，写 = 忽略。

位 4：　SHORT：短选通使能
使能/禁止短选通功能。
0：短选通禁止。
1：短选通使能。

位 3-0：　ADDR[3:0]：smaRTClock 间接寄存器地址
设置当前选择的 smaRTClock 寄存器。

注：在每次对 CAPTUREn 或 ALARMn 内部 smaRTClock 寄存器进行读/写操作后，ADDR[3:0]的值加 1。

SFR 定义 20.3　RTC0DAT：smaRTClock 数据寄存器

SFR 页：0x00
SFR 地址：0xAD

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	

位 7-0：　RTC0DAT[7:0]：smaRTClock 接口数据位。
保持传送到内部 smaRTClock 寄存器或从内部 smaRTClock 寄存器读取的数据，内部 smaRTClock 寄存器由 RTC0ADR 选择。

注：不应对该寄存器使用读-修改-写指令（orl、anl 等）。

20.2 smaRTClock 时钟源

smaRTClock 外设有其自己的时间基准，该时基独立于系统时钟。smaRTClock 的时基来自 smaRTClock 振荡器电路。该振荡器有两种工作方式：晶体方式和自振荡方式。晶体方式的振荡频率为 32.768 KHz；自振荡方式的振荡频率可被编程为大约 TBD KHz 到 TBD KHz。可以使用另一个振荡器为参考，用片内定时器测量 smaRTClock 振荡器的频率。“25. 定时器”介绍了如何实现这种测量。

注：smaRTClock 时钟可以被选择为系统时钟和连接到一个端口引脚。有关选择系统时钟源的详细信息“19. 时钟源”。有关如何将系统时钟连接到一个端口引脚的详细信息见“18. 端口输入输出”。

20.2.1 使用 smaRTClock 振荡器的晶体方式

当使用晶体方式时，应在 XTAL3 和 XTAL4 之间连接一个 32.768 KHz 的晶体。不需要其他外部元件。下面的步骤说明了如何在软件中启动 smaRTClock 晶体振荡器。

第一步：设置 smaRTClock 为晶体方式（XMODE = 1）。

第二步：*选项*。使能/禁止自动增益控制（AGCEN）和偏置加倍（BIASX2）。使用这些振荡器功能的建议见 20.2.4 节。

第三步：设置所期望的负载电容（RTC0XCF）。

第四步：使能 smaRTClock 振荡器电路的电源（RTC0EN = 1）。

第五步：等待 2ms。

第六步：查询 smaRTClock 时钟有效位（CLKVLD），直到晶体振荡器稳定。

第七步：查询 smaRTClock 负载电容准备好位（LOADRDY），直到负载电容达到其编程值。

第八步：使能 smaRTClock 时钟丢失监测器。

第九步：等待 2ms。

第十步：清除 PMU0CF 唤醒源标志。

20.2.2 使用 smaRTClock 振荡器的自振荡方式

当使用自振荡方式时，XTAL3 和 XTAL4 引脚应被短接。下面的步骤说明了如何将 smaRTClock 配置为使用自振荡方式。

第一步：设置 smaRTClock 为自振荡方式（XMODE = 0）。

第二步：选择所期望的振荡频率：

要得到约 20 KHz 的频率，设置 BIASX2 = 0。

要得到约 40 KHz 的频率，设置 BIASX2 = 1。

第三步：启动振荡器，振荡器会立即起振。

第四步：通过调整负载电容（RTC0XCF）来微调振荡器频率。

20.2.3 可编程负载电容

可编程负载电容有 16 个设置值，能支持需要很宽范围的建议负载电容的晶体振荡器。如果自动负载电容步进功能被使能，晶体负载电容从最小的设置值开始（允许快速启动时间）缓慢增加，直到达到最终的编程值。最终的编程负载电容值用 RTC0XCF 寄存器中的 LOADCAP 位设置。LOADCAP 的设置只定义片内负载电容的大小，不包括任何寄生 PCB 电容。一旦达到了最终的编程负载电容值，LOADRDY 标志即被硬件置为逻辑 1。

当在自振荡方式使用 smaRTClock 振荡器时，可编程负载电容可用于微调振荡频率。在大多数情况下，增加负载电容值会使振荡频率下降。

表 20.2 列出了不同的 LOADCAP 设置所对应的晶体负载电容。

表 20.2 smaRTClock 负载电容设置

LOADCAP	晶体负载电容	从 XTAL3 和 XTAL4 看的等效负载电容
0000	4.0 pF	8.0 pF
0001	4.5 pF	9.0 pF
0010	5.0 pF	10.0 pF
0011	5.5 pF	11.0 pF
0100	6.0 pF	12.0 pF
0101	6.5 pF	13.0 pF
0110	7.0 pF	14.0 pF
0111	7.5 pF	15.0 pF
1000	8.0 pF	16.0 pF
1001	8.5 pF	17.0 pF
1010	9.0 pF	18.0 pF
1011	9.5 pF	19.0 pF
1100	10.5 pF	21.0 pF
1101	11.5 pF	23.0 pF
1110	12.5 pF	25.0 pF
1111	13.5 pF	27.0 pF

20.2.4 自动增益控制（仅限于晶体方式）和 smaRTClock 偏置加倍

自动增益控制允许 smaRTClock 振荡器调整晶体的振荡幅值以获得最低的可能功耗。自动增益控制电路自动检测振荡幅值是否已经达到了可以安全地降低驱动电流的门限值，因此，可以在晶体启动期间使能自动增益控制。建议在使用 smaRTClock 振荡器晶体方式的任何系统中使能自动增益控制。

关闭自动增益控制会使振荡器启动后的晶体驱动强度保持在启动晶体时的水平。这会导致功耗增加，但晶体会有更强的抗外部干扰的能力。

注意：如果在自振荡方式下使用 smaRTClock 振荡器，则自动增益功能必须被禁止。否则，振荡频率可能会发生不可预见的变化。

smaRTClock 偏置加倍功能允许自振荡频率增加（几乎加倍），在晶体方式下可以获得更高的晶体驱动强度。当晶体使用高 ESR 的大负载电容时，建议使用高晶体驱动强度。表 20.3 列出了振荡器工作方式和所允许的工作条件。通过将 BIASX2（RTC0XCN.5）置 1 来使能 smaRTClock 偏置加倍。

表 20.3 smaRTClock 偏置设置和所允许的工作条件

方式	设置	功耗	允许的工作条件
晶体	偏置加倍禁止，AGC 使能	最低	ESR< 40 KΩ，任何负载 ESR< 50 KΩ，负载< 10 pF ESR< 80 KΩ，负载< 8 pF
	偏置加倍禁止，AGC 禁止	低	ESR< 80 KΩ，负载< 10 pF
	偏置加倍使能，AGC 使能	高	ESR< 50 KΩ，任何负载 ESR< 80 KΩ，负载< 10 pF
	偏置加倍使能，AGC 禁止	最高	该方式只建议用于调试目的， 因为该方式使功耗增加。
自振荡	偏置加倍禁止	低	20KHz
	偏置加倍使能	高	40KHz

20.2.5 smaRTClock 时钟丢失检测器

smaRTClock 时钟丢失检测器是一个单稳态电路，在 MCLKEN（RTC0XCN.6）置 1 时被使能。当 smaRTClock 时钟丢失检测器被使能时，如果 smaRTClock 振荡器保持高或低电平的时间大于 100 微秒，则 OSCFAIL（RTC0XCN.5）被硬件置 1。

smaRTClock 时钟丢失检测器超时可触发中断，将器件从低功耗方式唤醒，或将器件复位。详见“12. 中断系统”、“14. 电源管理”和“18. 复位源”。

注意：当改变 RTC0XCN 中的振荡器设置时，smaRTClock 时钟丢失检测器应被禁止。

20.2.6 smaRTClock 振荡器晶体有效检测器

smaRTClock 振荡器晶体有效检测器是一个振荡幅值检测电路，在晶体启动期间使用，以确定振荡器何时启动和接近稳定。该检测器的输出可以用 CLKVLD 位（RTC0XCN.4）读出。

注意：CLKVLD 位需要一个 2ms 的等待时间。在晶体振荡器上电后的前 2ms，CLKVLD 输出无效。

注意：该 smaRTClock 振荡器晶体有效检测器（CLKVLD）不是用来检测振荡器故障的。检测振荡器故障应使用 smaRTClock 时钟丢失检测器（CLKFAIL）。

20.3 smaRTClock 定时器和报警功能

smaRTClock 定时器是一个 32 位的计数器，当运行时（RTC0TR = 1），在每个 smaRTClock 振荡器周期增 1。该定时器有告警功能，该功能可以被设置为在某一特定时间产生中断，将器件从一种低功耗方式唤醒，或复位 MCU。详见“12. 中断系统”、“14. 电源管理”和“18. 复位源”。

smaRTClock 定时器有自动复位功能，该功能在发生告警的一个 smaRTClock 时钟周期后会自动将定时器复位到 0 值。当使用自动复位时，告警匹配值应总是被设置为比期望的匹配值少 1。通过向 ALRM（RTC0CN.2）写 1 来使能自动复位。

20.3.1 设置和读取 smaRTClock 定时器值

用 4 个 CAPTUREn 内部寄存器来设置或读取 32 位的 smaRTClock 定时器值。注意：在读取或设置定时器值之前不需要停止定时器。可用下面的步骤设置定时器值：

第一步：将 32 位的设置值写入 CAPTUREn 寄存器。

第二步：向 RTC0SET 写 1。该操作将 CAPTUREn 寄存器的内容传送到 smaRTClock 定时器。

第三步：当 RTC0SET 被硬件清 0 时操作结束。

可用下面的步骤读当前的定时器值：

第一步：向 RTC0CAP 写 1。该操作将定时器的内容传送到 CAPTUREn 寄存器。

第二步：查询 RTC0CAP，直到其被硬件清 0。

第三步：可以从 CAPTUREn 寄存器读取定时器的快照值。

20.3.2 设置 smaRTClock 告警值

smaRTClock 告警功能将 32 位的 smaRTClock 定时器值与 ALARMn 寄存器的值进行比较。如果 smaRTClock 定时器值等于 ALARMn 寄存器的值，则会触发一个告警事件。如果自动复位功能被使能，则 32 位的定时器在发生告警事件的一个 smaRTClock 时钟周期后被清 0。

smaRTClock 告警事件可以被配置为复位 MCU，将 MCU 从一种低功耗方式唤醒，或产生一个中断。详见“12. 中断系统”、“14. 电源管理”和“18. 复位源”。

可用下面的步骤设置 smaRTClock 告警：

第一步：禁止 smaRTClock 告警事件（RTC0AEN = 0）。
第二步：将 ALARMn 寄存器设置为期望值。
第三步：使能 smaRTClock 告警事件（RTC0AEN = 1）。

注意：用作 smaRTClock 告警事件标志的 ALRM 位可以通过禁止 smaRTClock 告警事件（RTC0AEN = 0）来清 0。

注意：在发生一次 smaRTClock 告警后，先禁止（RTC0AEN = 0）然后再使能告警事件（RTC0AEN = 1）会自动将下一次告警设置为在 2^{32} 个 smaRTClock 时钟周期之后发生（使用 32.768KHz 的晶体时约为 36 小时），并且不改变 ALARMn 寄存器。

注意：smaRTClock 告警事件标志能保持有效状态的最长时间为一个 smaRTClock 时钟周期。有关如何使用一个不被硬件自动清 0 的标志来捕捉一个 smaRTClock 告警事件的详细信息，见“14. 电源管理”。

20.3.3 使用 smaRTClock 定时器和告警的软件考虑

smaRTClock 定时器值和告警有两种工作方式，以适应不同的应用。这两种工作方式描述如下：

方式 1：

第一种方式使用 smaRTClock 定时器作为一个不会被清 0 的永久性时基。该定时器每 36 个小时溢出一次，并且不会被停止或中断。告警间隔由软件管理，该间隔在每次告警发生后由软件加到 ALRMn 寄存器。这样就允许告警匹配值总是比软件管理的告警间隔多 1。如果软件使用 32 位的无符号加法使告警匹配值加 1，则它不需要处理溢出，因为定时器和告警匹配值都会以同样的方式溢出。

该方式适合于具有很长告警间隔（如 24 或 36 小时）和/或需要一个永久性时基的应用。需要一个永久性时基的应用例子之一就是唤醒间隔不断变化的应用。对于这样一些应用，软件可以用一个 16 位的变量跟踪定时器的溢出次数，将 32 位（36 小时）的定时器扩展为 48 位（272 年）的永久性时基。

方式 2：

第二种方式使用 smaRTClock 定时器作为一个通用的加 1 计数器，该计数器在每次告警后由硬件清 0。告警间隔由硬件管理并保存在 ALRMn 寄存器中。软件只需要在器件初始化期间设置一次告警间隔。在每次告警后，软件应保持发生过的告警计数值，以便跟踪时间。

该方式适合于需要最少软件干预和/或具有固定告警间隔的应用。该方式的电源效率最高，因为每次告警处理需要较短的 CPU 时间。

内部寄存器定义 20.4　RTC0CN：smaRTClock 控制寄存器

smaRTClock 地址：0x04

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
RTC0EN	MCLKEN	OSCFAIL	RTC0TR	RTC0AEN	ALRM	RTC0SET	RTC0CAP	00v00000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　RTC0EN：smaRTClock 使能位
使能/禁止 smaRTClock 振荡器及其偏置电流。
0：smaRTClock 振荡器禁止。
1：smaRTClock 振荡器使能。

位 6：　MCLKEN：smaRTClock 时钟丢失检测器使能位
使能/禁止 smaRTClock 时钟丢失检测器。
0：smaRTClock 时钟丢失检测器禁止。
1：smaRTClock 时钟丢失检测器使能。

位 5：　OSCFAIL：smaRTClock 振荡器故障标志
当 smaRTClock 时钟丢失检测器发生超时时，该位被硬件置 1。该位必须由软件清 0。当 smaRTClock 振荡器被禁止时，该位的值没有定义。

位 4：　RTC0TR：smaRTClock 定时器运行控制
控制 smaRTClock 定时器运行还是停止（保持当前值）。
0：smaRTClock 定时器停止。
1：smaRTClock 定时器运行。

位 3：　RTC0AEN：smaRTClock 告警使能位
使能/禁止 smaRTClock 告警功能。还清除 ALRM 标志。
0：smaRTClock 告警事件禁止。
1：smaRTClock 告警事件使能。

位 2：　ALRM：smaRTClock 告警事件标志和自动复位使能
读操作返回告警事件标志的状态。写操作使能/禁止自动复位功能。
读：
0：smaRTClock 告警事件标志无效。
1：smaRTClock 告警事件标志有效。
写：
0：禁止自动复位。
1：使能自动复位。

位 1：　RTC0SET：smaRTClock 定时器设置位
向该位写 1 启动一次 smaRTClock 定时器设置操作。该位由硬件自动清 0 时指示定时器设置操作完成。

位 0：　RTC0CAP：smaRTClock 定时器捕捉位
向该位写 1 启动一次 smaRTClock 定时器捕捉操作。该位由硬件自动清 0 时指示定时器捕捉操作完成。

内部寄存器定义 20.5　RTC0XCN：smaRTClock 振荡器控制寄存器

smaRTClock 地址：0x05

R/W	R/W	R/W	R	R	R	R	R	复位值
AGCEN	XMODE	BIASX2	CLKVLD	—	—	—	—	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　AGCEN：smaRTClock 振荡器自动增益控制（AGC）使能位
0：AGC 禁止。
1：AGC 使能。

位 6：　XMODE：smaRTClock 振荡器方式位
选择晶体或自振荡方式。
0：选择自振荡方式。
1：选择晶体方式。

位 5：　BIASX2：smaRTClock 振荡器偏置加倍使能位
使能/禁止偏置加倍功能。
0：偏置加倍禁止。
1：偏置加倍使能。

位 4：　CLKVLD：smaRTClock 振荡器晶体有效标志位
指示振荡幅值是否足够维持振荡。
0：振荡未启动或正当幅值过低，不能维持振荡。
1：检测到足够强的振荡幅值。

位 3-0：　未用。读 = 0000b，写 = 忽略。

内部寄存器定义 20.6　RTC0XCN：smaRTClock 振荡器配置寄存器

smaRTClock 地址：0x06

R/W	R	R	R	R/W	R/W	R/W	R/W	复位值
AUTOSTP	LOADRDY	—	—	LOADCAP				00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　AUTOSTP：自动负载电容步进使能位
使能/禁止自动负载电容步进功能。
0：禁止自动负载电容步进功能。
1：使能自动负载电容步进功能。

位 6：　LOADRDY：负载电容准备好标志
当负载电容与编程值匹配时由硬件置 1。
0：负载电容正在步进调整。
1：负载电容已达到其编程值。

位 5-4：　未用。读 = 00b，写 = 忽略。

位 3-0：　LOADCAP：负载电容编程值
保持用户期望的负载电容值。见表 20.2。

内部寄存器定义 20.7　RTC0PIN：smaRTClock 引脚配置寄存器

smaRTClock 地址：0x07

W	W	W	W	W	W	W	W	复位值
RTC0PIN7	RTC0PIN6	RTC0PIN5	RTC0PIN4	RTC0PIN3	RTC0PIN2	RTC0PIN1	RTC0PIN0	01100111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　RTC0PIN[7:0]：smaRTClock 引脚配置
向该寄存器写 0xE7 强制 XTAL3 和 XTAL4 在内部短接，以使用自振荡方式。
向该寄存器写 0x67 使 XTAL3 和 XTAL4 返回到正常方式。

内部寄存器定义 20.8　CAPTUREn：smaRTClock 定时器捕捉寄存器

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	

smaRTClock 地址：CAPTURE0 = 0x00; CAPTURE1= 0x01; CAPTURE2 = 0x02; CAPTURE3 = 0x03

位 7-0：　CAPTUREn：smaRTClock 定时器捕捉值
这 4 个寄存器（CAPTURE3 ～ CAPTURE0）用于读或设置 32 位的 smaRTClock 定时器。当 RTC0SET 或 RTC0CAP 位被置 1 时，数据被传送到 smaRTClock 定时器或从 smaRTClock 定时器读出。

注：定时器捕捉值的最低位是 CAPTURE0.0。

内部寄存器定义 20.9　ALARMn：smaRTClock 告警编程值寄存器

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	

smaRTClock 地址：ALARM0= 0x08; ALARM1= 0x09; ALARM2 = 0x0A; ALARM3= 0x0B 别

位 7-0：　ALARMn：smaRTClock 告警编程值
这 4 个寄存器（ALARM3 ~ ALARM0）用于设置 smaRTClock 定时器的告警事件。当更新这些寄存器时，smaRTClock 告警应被禁止（RTC0AEN = 0）。

注：告警编程值的最低位是 ALARM0.0。

21. 端口输入/输出

数字和模拟资源可以通过 24 个 I/O 引脚（C8051F930/920）或 16 个 I/O 引脚（C8051F931/921）使用。端口引脚被组织为三个 8 位端口。端口引脚 P0.0 ~ P2.6 可以被定义为数字或模拟 I/O。数字 I/O 引脚可以被分配给内部数字资源或用作通用 I/O（GPIO）。模拟 I/O 引脚可以被内部模拟资源使用。P2.7 可以用作 GPIO，与 C2 接口数据信号（C2D）共享。详见“27. C2 接口”。

设计者完全控制数字和模拟功能分配到哪些引脚，只受物理 I/O 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。有关交叉开关的详细信息见 21.3 节。

所有用作数字输入或漏极开路输出的端口 I/O 都耐 5V 电压。所有被配置为推挽输出的端口 I/O 的拉电流由 VDD/DC+电源供给。用于模拟功能的端口 I/O 的工作电压可以达到 VDD/DC+电源电压。有关端口 I/O 工作方式的详细信息见 21.1 节。端口 I/O 的详细电气特性见电气特性一章。

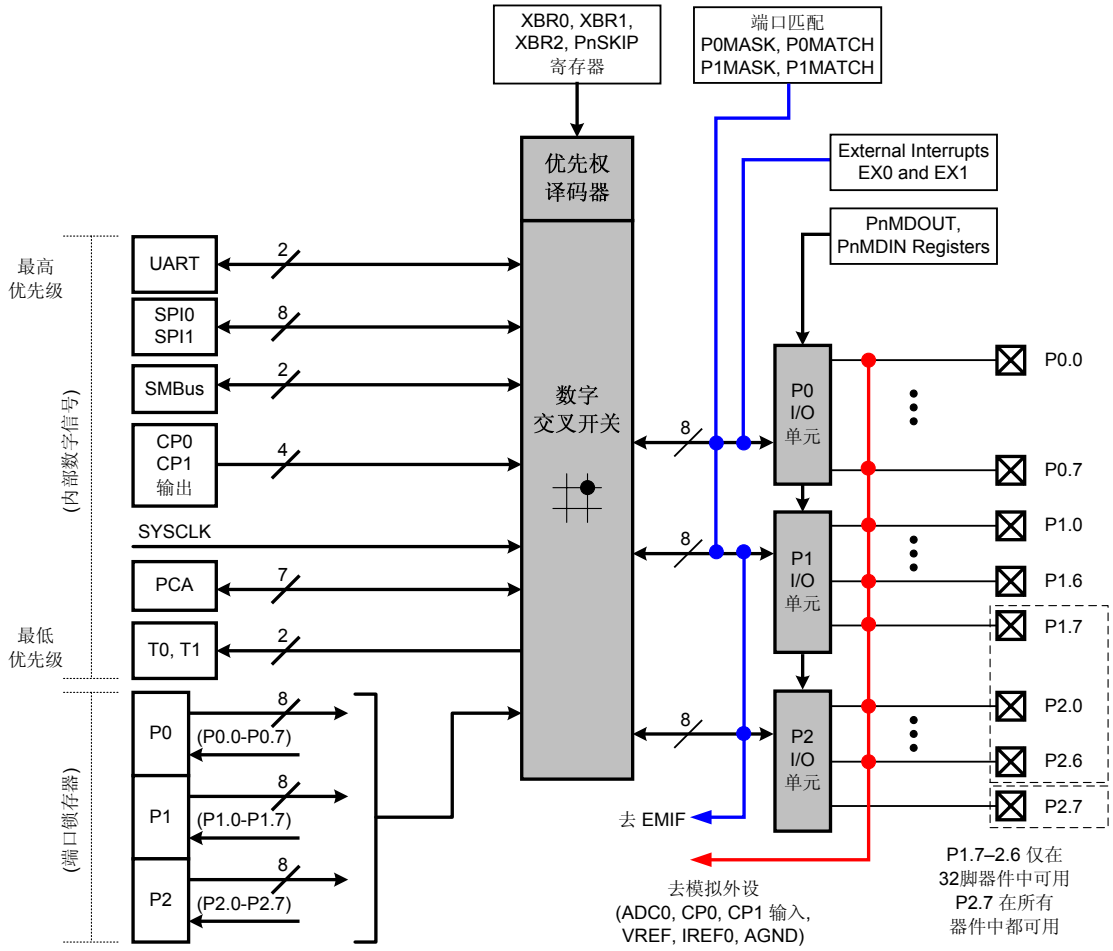


图 21.1 端口 I/O 功能框图

21.1 端口 I/O 的工作方式

P0.0 ~ P2.6 使用如图 21.2 所示的端口 I/O 单元。每个端口 I/O 单元都可以由软件用 PnMDIN 寄存器配置为模拟 I/O 或数字 I/O。在复位后，所有端口 I/O 单元均默认为数字高阻状态且弱上拉使能，直到交叉开关被使能（XBARE = 1）。

21.1.1 端口引脚配置为模拟 I/O

任何用作比较器或 ADC 输入、外部振荡器输入/输出、AGND、VREF 或电流基准输出的引脚都应被配置为模拟 I/O（PnMDIN.n = 0）。当一个引脚被配置为模拟 I/O 时，其弱上拉和数字接收器均被禁止。在大多数情况下，软件应禁止数字输出驱动器。读取被配置为模拟 I/O 的引脚时将总是返回 0 值，与引脚上的实际电压无关。

将引脚配置为模拟 I/O 可以节省功耗并将其与数字干扰隔离。被配置为数字输入的引脚仍可被模拟外设使用，但不建议这样做，因为这样可能导致测量误差。

21.1.2 端口引脚配置为数字 I/O

任何被数字外设（UART、SPI、SMBus 等）、外部数字事件捕捉功能使用的引脚和作为 GPIO 使用的引脚都应被配置为数字 I/O（PnMDIN.n = 1）。对于数字 I/O 引脚，必须用 PnMDOUT 寄存器在两种输出方式（漏极开路或推挽方式）中选择其一。

推挽输出（PnMDOUT.n = 1）根据端口引脚的输出逻辑值将端口引脚驱动到 VDD/DC+ 或 GND。漏极开路输出时高端驱动器被禁止，因此当输出逻辑值为 0 时，端口引脚只被驱动到 GND；而当输出逻辑值为 1 时，端口引脚变成高阻输入（高端和低端驱动器均被断开）。

当一个数字 I/O 单元被置于高阻状态时，一个弱上拉晶体管将端口引脚拉向 VDD/DC+电源电压，以保证数字输入处于一个有定义的逻辑状态。当数字 I/O 单元被驱动到 GND 时，弱上拉被禁止以节省功耗。通过将 WEAKPUD 置 1 可以总体禁止弱上拉。用户应保证数字 I/O 总是被内部或外部拉向或驱动到一个有效的逻辑状态。被配置为数字 I/O 的端口引脚总是读回端口引脚的逻辑状态，而与该端口引脚的输出逻辑值无关。

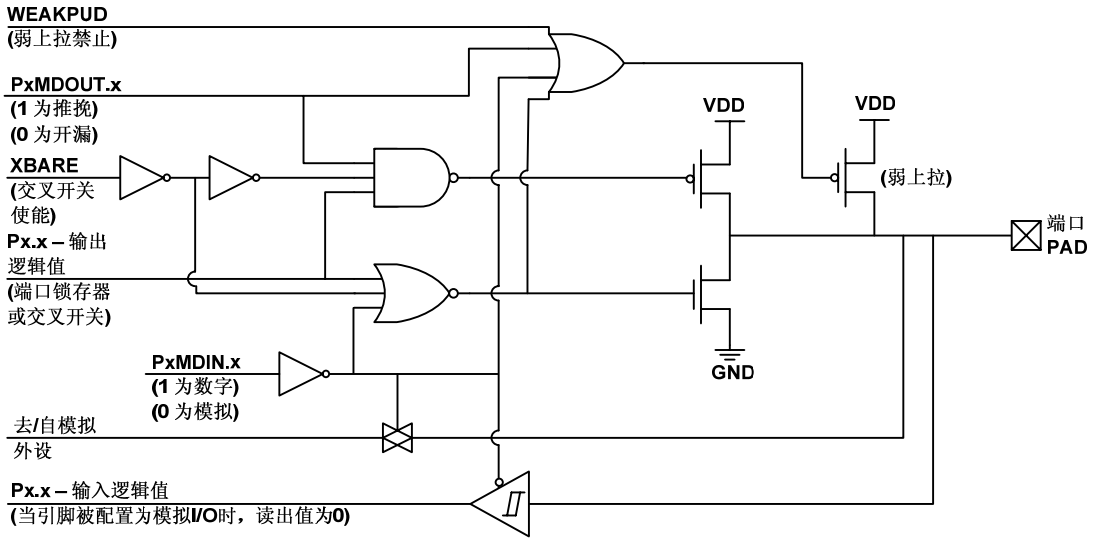


图 20.2 端口 I/O 单元框图

21.1.3 端口 I/O 与 5V 逻辑的接口

被配置为数字、漏极开路操作的所有端口 I/O 都能与工作在高于 VDD/DC+但小于 5.25V 的电源电压的数字逻辑接口。对于大多数系统来说，通常需要一个连接到较高电源电压的外部上拉电阻。

注意：在一个多电压接口中，外部上拉电阻的选择应满足：当电源电压位于（VDD/DC+加 0.6V）与（VDD/DC+加 1.0V）之间时，应允许至少 150 μA 流入端口引脚。一旦端口引脚的电压增加到高于这一范围，流入端口引脚的电流为最小。

21.1.4 增加端口 I/O 驱动强度

端口 I/O 输出驱动器支持高驱动强度和低驱动强度；默认设置为低驱动强度。可以用 PnDRV 寄存器配置端口 I/O 的驱动强度。有关两种方式之间输出驱动强度的差别见“4. 电气特性”。

21.2 端口 I/O 引脚分配给模拟和数字功能

端口 I/O 引脚 P0.0 ~ P2.6 可以被可以被分配给不同的模拟、数字和外部中断功能。分配给模拟功能的端口引脚应被配置为模拟 I/O，分配给数字或外部中断功能的端口引脚应被配置为数字 I/O。

21.2.1 端口引脚分配给模拟功能

表 21.1 列出了所有需要分配端口 I/O 的模拟功能。**对于为这些模拟功能选择的引脚，其数字驱动器应被禁止（PnMDOUT.n = 0 且端口锁存器 = 1），在 PnSKIP 寄存器中的对应位应被置 1**。这样将保留这些引脚为模拟功能所用，而不允许这些引脚被交叉开关分配。表 21.1 列出了端口 I/O 到每个模拟功能的潜在映射。

表 21.1 模拟功能的端口 I/O 分配

模拟功能	潜在的可分配端口引脚	用于引脚分配的SFR
ADC输入	P0.0 ~ P2.6	ADC0MX, PnSKIP
比较器0输入	P0.0 ~ P2.6	CPT0MX, PnSKIP
比较器1输入	P0.0 ~ P2.6	CPT1MX, PnSKIP
电压基准（VREF0）	P0.0	REF0CN, PnSKIP
模拟地基准（AGND）	P0.1	REF0CN, PnSKIP
电流基准（IREF0）	P0.7	IREF0CN, PnSKIP
外部振荡器输入（XTAL1）	P0.2	OSCXCN, PnSKIP
外部振荡器输出（XTAL2）	P0.3	OSCXCN, PnSKIP

21.2.2 端口引脚分配给数字功能

任何未分配给模拟功能的端口 I/O 都可以被分配给数字功能或或用作 GPIO。大多数数字功能依赖交叉开关分配引脚；但是，有些数字功能以与前面列出的模拟功能类似的方式旁路了交叉开关。**这些数字功能使用的端口引脚和任何被选择作为 GPIO 使用的端口引脚在 PnSKIP 寄存器中的对应位应被置 1**。表 21.2 列出了所有可用的数字功能和端口 I/O 到每个数字功能的潜在映射。

表 21.2 数字功能的端口 I/O 分配

数字功能	潜在的可分配端口引脚	用于引脚分配的SFR
UART0、SPI1、SPI0、SMBus、CP0和CP1输出、系统时钟输出、PCA0、定时器0和定时器1外部输入。	任何可供交叉开关分配的端口引脚。这包括PnSKIP位为0的P0.0 ~ P2.6引脚。 注：交叉开关总是将UART0和SPI1引脚分配到固定的位置。	XBR0、XBR1、XBR2
用作GPIO的任何引脚	P0.0 ~ P2.6	P0SKIP、P1SKIP、P2SKIP
外部存储器接口	P1.0 ~ P2.6	P1SKIP、P2SKIP、EMI0CF

21.2.3 端口引脚分配给外部数字事件捕捉功能

外部数字事件捕捉功能可用于触发中断或将器件从一个低功耗方式唤醒（当一个数字 I/O 引脚上发生状态改变时）。数字事件捕捉功能不需要专用的引脚，它既可以使用 GPIO 引脚工作（PnSKIP = 1），也可以使用交叉开关所使用的引脚工作（PnSKIP = 0）。外部数字事件捕捉功能不能在那些被配置为模拟 I/O 的引脚上使用。表 21.3 列出了所有可用的外部数字事件捕捉功能。

表 21.3 外部数字事件捕捉功能的端口 I/O 分配

数字功能	潜在的可分配端口引脚	用于引脚分配的SFR
外部中断0	P0.0 ~ P0.7	IT01CF
外部中断1	P0.0 ~ P0.7	IT01CF
端口匹配	P0.0 ~ P1.7	P0MASK、P0MAT P1MASK、P1MAT

21.3 优先权交叉开关译码器

优先权交叉开关译码器使用如图 21.3 所示的固定优先权外设顺序为软件选择的每个数字功能分配端口 I/O 引脚。在 SFR 定义 21.1、SFR 定义 21.2 和 SFR 定义 21.3 中定义的 XBR0、XBR1 和 XBR2 用于选择交叉开关中的数字功能。可被交叉开关分配的端口引脚包括 PnSKIP 中对应位被设置为 0 的所有引脚（P2.0 ~ P2.6）。

从图 21.3 中可以看出，优先权最高的外设为 UART0。如果 UART0 在交叉开关中被选择（使用 XBRn 寄存器），则 P0.4 和 P0.5 被分配给 UART0。下一个优先权最高的外设为 SPI1。如果 SPI1 在交叉开关中被选择，则 P1.0 ~ P1.3 被分配给 SPI1。用户应保证被交叉开关分配的引脚在 PnSKIP 中的对应位被设置为 0。

对于交叉开关中被选择的所有其它数字功能，引脚分配从图 21.3 的顶部向下进行，当一个数字资源被选择时，将位序最低的未被跳过的未分配端口引脚分配给该功能。如果一个端口引脚已经被分配（例如 UART0 或 SPI1 引脚），或其 PnSKIP 位被置 1，则交叉开关会跳过该引脚并查找下一个可用的未被跳过也未被分配的端口引脚。用于模拟功能、GPIO 或专用数字功能如 EMIF 的所有端口引脚所对应的 PnSKIP 位都应被置 1。

图 21.3 给出了没有端口引脚被跳过（P0SKIP, P1SKIP, P2SKIP = 0x00）的优先权交叉开关译码表；图 21.4 给出了外部振荡器引脚（XTAL1 和 XTAL2）被跳过情况下（P0SKIP = 0x0C）的交叉开关优先权译码表。

注意：在端口引脚被用作数字输出之前，交叉开关必须被使能（XBARE = 1）。当交叉开关被禁止时，端口输出驱动器被禁止。

注意：当 SMBus 被交叉开关选择时，SDA 和 SCL 的对应引脚被自动强制为漏极开路输出方式，与 PnMDOUT 的设置无关。

注意：SPI0 可以工作在三线或四线方式，由 SPI0CN 寄存器中的 NSSMD1-NSSMD0 位的状态决定。NSS 信号只在选择了四线方式时才被连到端口引脚。当 SPI0 被交叉开关选择时，SPI0 方式（三线或四线）影响优先权低于 SPI0 的所有数字功能的引脚分配。

注意：对于给定的 XBRn、PnSKIP 和 SPInCN 寄存器设置，我们可以使用图 21.3 和图 21.4 来确定器件的 I/O 引脚分配。

	P0								P1								P2							
特殊功能信号	VREF	AGND	XTAL1	XTAL2			CNVSTR	IREF0	AD0	AD1	AD2	AD3	AD4	AD5	AD6	AD7	A8	A9	A10	A11	ALE	/RD	/WR	C2D
PIN I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
TX0					■																			
RX0						■																		
SCK (SPI1)									■															
MISO (SPI1)										■														
MOSI (SPI1)											■													
NSS* (SPI1)												■					(*仅用于4线SPI)							
SCK (SPI0)	■																							
MISO (SPI0)		■																						
MOSI (SPI0)			■																					
NSS* (SPI0)				■					(*仅用于4线SPI)															
SDA	■			■	■		■																	
SCL		■			■	■		■																
CP0	■		■	■		■	■	■	■			■	■											
CP0A	■	■	■	■	■	■	■	■	■	■		■	■	■										
CP1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■									
CP1A	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■								
/SYSCLK	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■							
CEX0	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■						
CEX1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					
CEX2	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				
CEX3	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
CEX4	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
CEX5	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
ECI	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
T0	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
T1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X
	P0SKIP[0:7]								P1SKIP[0:7]								P2SKIP[0:7]							

图 21.3 没有引脚被跳过的交叉开关优先权译码表

	P0								P1								P2							
特殊功能信号	VREF	AGND	XTAL1	XTAL2			CNVSTR	IREF0	AD0	AD1	AD2	AD3	AD4	AD5	AD6	AD7	A8	A9	A10	A11	ALE	/RD	/WR	C2D
PIN I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
TX0					■																			
RX0						■																		
SCK (SPI1)									■															
MISO (SPI1)										■														
MOSI (SPI1)											■													
NSS* (SPI1)												■					(*仅用于4线SPI)							
SCK (SPI0)	■																							
MISO (SPI0)		■																						
MOSI (SPI0)					■		■																	
NSS* (SPI0)						■		■	(*仅用于4线SPI)															
SDA	■				■	■	■	■	■			■	■											
SCL		■				■	■	■	■	■			■	■										
CP0	■	■			■	■	■	■	■	■	■		■	■	■									
CP0A	■	■			■	■	■	■	■	■	■	■	■	■	■	■								
CP1	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■							
CP1A	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■						
/SYSCLK	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					
CEX0	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				
CEX1	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
CEX2	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
CEX3	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
CEX4	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
CEX5	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
ECI	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
T0	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
T1	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X
	P0SKIP[0:7]								P1SKIP[0:7]								P2SKIP[0:7]							

图 21.4 晶体引脚被跳过的交叉开关优先权译码表

SFR 定义 21.1　XBR0：端口 I/O 交叉开关寄存器 0

SFR 页：0x00
SFR 地址：0xE1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CP1AE	CP1E	CP0AE	CP0E	SYSCKE	SMB0E	SPI0E	URT0E	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　CP1AE：比较器 1 异步输出使能位
0：异步 CP1 输出不连到端口引脚。
1：异步 CP1 输出连到端口引脚。

位 6：　CP1E：比较器 1 输出使能位
0：CP1 输出不连到端口引脚。
1：CP1 输出连到端口引脚。

位 5：　CP0AE：比较器 0 异步输出使能位
0：异步 CP0 输出不连到端口引脚。
1：异步 CP0 输出连到端口引脚。

位 4：　CP0E：比较器 0 输出使能位
0：CP0 输出不连到端口引脚。
1：CP0 输出连到端口引脚。

位 3：　SYSCKE：/SYSCLK 输出使能位
0：/SYSCLK 输出不连到端口引脚；
1：/SYSCLK 输出连到端口引脚。

位 2：　SMB0E：SMBus I/O 使能位
0：SMBus I/O 不连到端口引脚。
1：SDA 和 SCL 连到端口引脚。

位 1：　SPI0E：SPI0 I/O 使能位
0：SPI0 I/O 不连到端口引脚。
1：SCK、MISO 和 MOSI（对 SPI0）连到端口引脚。
NSS（对 SPI0）只在 SPI0 被配置为 4 线方式时才被连到端口引脚。

位 0：　URT0E：UART I/O 使能位
0：UART I/O 不连到端口引脚。
1：TX0 和 RX0 连到端口引脚 P0.4 和 P0.5。

注：SPI0 可被分配 3 或 4 个端口 I/O 引脚。

SFR 定义 21.2　XBR1：端口 I/O 交叉开关寄存器 1

SFR 页：0x00
SFR 地址：0xE2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	SPI1E	T1E	T0E	ECIE	PCA0ME[2:0]			00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　未用。读 = 0b，写 = 忽略。

位 6：　SPI1E：SPI1 I/O 使能位
0：SPI1 I/O 不连到端口引脚。
1：SCK（对 SPI1）连到 P1.0。
MISO（对 SPI1）连到 P1.1。
MOSI（对 SPI1）连到 P1.2。
NSS（对 SPI1）只在 SPI1 被配置为 4 线方式时才被连到 P1.3。

位 5：　T1E：定时器 1 输入使能位
0：定时器 1 输入不连到端口引脚。
1：定时器 1 输入连到端口引脚。

位 4：　T0E：定时器 0 输入使能位
0：定时器 0 输入不连到端口引脚。
1：定时器 0 输入连到端口引脚。

位 3：　ECIE：PCA0 外部计数输入（ECI）使能位
0：PCA0 外部计数输入不连到端口引脚。
1：PCA0 外部计数输入连到端口引脚。

位 2-0：　PCA0ME：PCA0 模块 I/O 使能位
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 连到端口引脚。
011：CEX0、CEX1、CEX2 连到端口引脚。
100：CEX0、CEX1、CEX2、CXE3 连到端口引脚。
101：CEX0、CEX1、CEX2、CXE3、CXE4 连到端口引脚。
110：CEX0、CEX1、CEX2、CXE3、CXE4、CXE5 连到端口引脚。
111：保留。

注：SPI1 可被分配 3 或 4 个端口 I/O 引脚。

SFR 定义 21.3　XBR2：端口 I/O 交叉开关寄存器 2

SFR 页：0x00
SFR 地址：0xE2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
WEAKPUD	XBARE	—	—	—	—	—	—	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　WEAKPUD：端口 I/O 弱上拉禁止位。
0：弱上拉使能（被配置为模拟方式的端口 I/O 引脚除外）。
1：弱上拉禁止。

位 6：　XBARE：交叉开关使能位
0：交叉开关禁止。
1：交叉开关使能。

位 5-0：　未用。读 = 000000b，写 = 忽略。

注：如果要使用任何端口引脚作为数字输出，必须使能交叉开关（XBARE = 1）。

21.4 端口匹配

端口匹配功能允许 P0 或 P1 口的逻辑值变化触发系统事件。存储在 PnMATCH 寄存器中的软件控制值定义了 P0 和 P1 的期望或正常逻辑值。如果端口输入引脚上的逻辑电平与该软件控制值不匹配，则会产生一个端口不匹配事件。该功能允许在 P0 或 P1 输入引脚发生某种变化时软件会得到通知，与 XBRn 的设置无关。

PnMASK 寄存器用于选择 P0 和 P1 的哪些引脚应与 PnMATCH 寄存器比较。如果（P0 & P0MASK）不等于（P0MATCH & P0MASK）或如果（P1 & P1MASK）不等于（P1MATCH & P1MASK），则会产生一个端口不匹配事件。

端口不匹配事件可以用于产生中断或将器件从低功耗方式唤醒。有关中断和唤醒源的详细信息见“12. 中断系统”和“14. 电源管理”。

SFR 定义 21.4　P0MASK：端口 0 屏蔽寄存器

SFR 页：0x00
SFR 地址：0xC7

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	

位 7-0：　P0MASK[7:0]：端口 0 屏蔽值
选择要与 P0MAT 中的对应位比较的端口引脚。
0：对应的 P0.n 引脚逻辑值被忽略，不能产生端口不匹配事件。
1：对应的 P0.n 引脚逻辑值与 P0MAT.n 比较。

SFR 定义 21.5　P0MAT：端口 0 匹配寄存器

SFR 页：0x00
SFR 地址：0xD7

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　P0MAT[7:0]：端口 0 匹配值
用于端口 0 的匹配比较值（对于 P0MAT 中被置 1 的那些位）。
0：P0.n 引脚逻辑值与逻辑低电平比较。
1：P0.n 引脚逻辑值与逻辑高电平比较。

SFR 定义 21.6　P1MASK：端口 1 屏蔽寄存器

SFR 页：0x00
SFR 地址：0xBF

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	

位 7-0：　P1MASK[7:0]：端口 1 屏蔽值
选择要与 P1MAT 中的对应位比较的 P1 引脚。
0：对应的 P1.n 引脚逻辑值被忽略，不能产生端口不匹配事件。
1：对应的 P1.n 引脚逻辑值与 P1MAT.n 比较。

SFR 定义 21.7　P1MAT：端口 1 匹配寄存器

SFR 页：0x00
SFR 地址：0xCF

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　P1MAT[7:0]：端口 1 匹配值
用于端口 1 的匹配比较值（对于 P1MAT 中被置 1 的那些位）。
0：P1.n 引脚逻辑值与逻辑低电平比较。
1：P1.n 引脚逻辑值与逻辑高电平比较。

21.5 访问和配置端口 I/O 的特殊功能寄存器

所有端口 I/O 都通过对应的特殊功能寄存器访问，这些寄存器既可以按位寻址也可以按字节寻址。向端口写入时，数值被锁存到 SFR 中，以保持每个引脚上的输出数据值不变。读端口时总是返回端口输入引脚本身的逻辑状态，而与 XBRn 的设置值无关（即使在引脚被交叉开关分配给另一信号时，端口寄存器总是可以读其对应的端口 I/O 引脚）。例外情况是执行目的操作数为端口锁存器的读-修改-写指令。对端口锁存器执行的读-修改-写指令如下：ANL、ORL、XRL、JBC、CPL、INC、DEC、DJNZ 和对端口 SFR 中的单个位执行的 MOV、CLR、SETB。这些指令读端口寄存器（而不是引脚）的值，修改后再写回端口 SFR。

每个端口都有一个对应的 PnSKIP 寄存器，该寄存器允许单独的端口引脚被分配给数字功能或被交叉开关跳过。应将用于模拟功能、GPIO 或专用数字功能（如 EMIF）的所有端口引脚的对应 PnSKIP 位置 1。

用端口输入方式寄存器（PnMDIN）定义端口引脚的输入方式。每个端口单元都可以被配置为模拟或数字 I/O。即使对于在 XBRn 寄存器中被选择的数字资源，也需要进行这种选择（不是自动的）。唯一的例外是 P2.7，它只能被用作数字 I/O。

I/O 引脚的输出驱动器特性由端口输出方式寄存器（PnMDOUT）定义，每个端口输出驱动器都可被配置为漏极开路或推挽方式。即使对于在 XBRn 中被选择的数字资源，也需要进行这种选择（不是自动的）。唯一的例外是 SMBus 引脚（SDA，SCL），不管 PnMDOUT 的设置如何，这两个引脚总是被配置为漏极开路。

端口驱动强度（PnDRV）寄存器控制输出驱动器的驱动强度。默认设置为低驱动强度。有关两种方式之间输出驱动强度的差别见“4. 电气特性”。

SFR 定义 21.8　P0：端口 0 寄存器

SFR 页：所有页
SFR 地址：0x80　（可位寻址）

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	

位 7-0：　P0[7:0]：端口 0 数据。
在端口单元被配置为数字 I/O 时，设置端口锁存器逻辑值或读端口引脚的逻辑状态。
写 － 设置端口锁存器逻辑值。
0：设置输出锁存器为逻辑低电平。
1：设置输出锁存器为逻辑高电平。

读 － 读那些被配置为数字 I/O 的端口引脚的逻辑状态。
0：P0.n 端口引脚为逻辑低电平。
1：P0.n 端口引脚为逻辑高电平。

SFR 定义 21.9　P0SKIP：端口 0 跳过寄存器

SFR 页：0x00
SFR 地址：0xD4

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	

位 7-0：　P0SKIP[7:0]：端口 0 交叉开关跳过使能位。
这些位选择被交叉开关译码器跳过的端口 0 引脚。用作模拟功能、特殊功能或 GPIO 的端口引脚应被交叉开关跳过。
0：对应的 P0.n 不被交叉开关跳过。
1：对应的 P0.n 被交叉开关跳过。

SFR 定义 21.10　P0MDIN：端口 0 输入方式寄存器

SFR 页：0x00
SFR 地址：0xF1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　P0MDIN[7:0]：P0.7 – P0.0 的模拟配置位（分别对应）
当端口引脚被配置为模拟方式时，其弱上拉和数字接收器都被禁止。数字驱动器未被显式禁止。
0：对应的 P0.n 引脚被配置为模拟方式。
1：对应的 P0.n 引脚不配置为模拟方式。

SFR 定义 21.11　P0MDOUT：端口 0 输出方式寄存器

SFR 页：0x00
SFR 地址：0xA4

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	

位 7-0：　P0MDOUT[7:0]：P0.7 – P0.0 的输出方式配置位（分别对应）。
这些位控制数字驱动器，即使 P0MDIN 寄存器中的对应位为逻辑 0 也不例外。
0：对应的 P0.n 输出为漏极开路。
1：对应的 P0.n 输出为推挽方式。

SFR 定义 21.12　P0DRV：端口 0 驱动强度寄存器

SFR 页：0x0F
SFR 地址：0xA4

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	

位 7-0：　P0DRV[7:0]：P0.7 – P0.0 的驱动强度配置位（分别对应）。
配置数字 I/O 端口单元为高或低输出驱动强度。
0：对应的 P0.n 为低输出驱动强度。
1：对应的 P0.n 为高输出驱动强度。

SFR 定义 21.13　P1：端口 1 寄存器

SFR 页：所有页
SFR 地址：0x90　（可位寻址）

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	

位 7-0：　P1[7:0]：端口 1 数据。
在端口单元被配置为数字 I/O 时，设置端口锁存器逻辑值或读端口引脚的逻辑状态。
写 － 设置端口锁存器逻辑值。
0：设置输出锁存器为逻辑低电平。
1：设置输出锁存器为逻辑高电平。

读 － 读那些被配置为数字 I/O 的端口引脚的逻辑状态。
0：P1.n 端口引脚为逻辑低电平。
1：P1.n 端口引脚为逻辑高电平。

注：引脚 P1.7 仅在 32 脚器件中可用。

SFR 定义 21.14　P1SKIP：端口 1 跳过寄存器

SFR 页：0x00
SFR 地址：0xD5

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	

位 7-0：　P1SKIP[7:0]：端口 1 交叉开关跳过使能位。
这些位选择被交叉开关译码器跳过的端口 1 引脚。用作模拟功能、特殊功能或 GPIO 的端口引脚应被交叉开关跳过。
0：对应的 P1.n 不被交叉开关跳过。
1：对应的 P1.n 被交叉开关跳过。

注：引脚 P1.7 仅在 32 脚器件中可用。

SFR 定义 21.15　P1MDIN：端口 1 输入方式寄存器

SFR 页：0x00
SFR 地址：0xF2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　P1MDIN[7:0]：P1.7 – P1.0 的模拟配置位（分别对应）
当端口引脚被配置为模拟方式时，其弱上拉和数字接收器都被禁止。数字驱动器未被显式禁止。
0：对应的 P1.n 引脚被配置为模拟方式。
1：对应的 P1.n 引脚不配置为模拟方式。

注：引脚 P1.7 仅在 32 脚器件中可用。

SFR 定义 21.16　P1MDOUT：端口 1 输出方式寄存器

SFR 页：0x00
SFR 地址：0xA5

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	

位 7-0：　P1MDOUT7:0]：P1.7 – P1.0 的输出方式配置位（分别对应）。
这些位控制数字驱动器，即使 P1MDIN 寄存器中的对应位为逻辑 0 也不例外。
0：对应的 P1.n 输出为漏极开路。
1：对应的 P1.n 输出为推挽方式。

注：引脚 P1.7 仅在 32 脚器件中可用。

SFR 定义 21.17　P1DRV：端口 1 驱动强度寄存器

SFR 页：0x0F
SFR 地址：0xA5

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	

位 7-0：　P0DRV[7:0]：P1.7 – P1.0 的驱动强度配置位（分别对应）。
配置数字 I/O 端口单元为高或低输出驱动强度。
0：对应的 P1.n 为低输出驱动强度。
1：对应的 P1.n 为高输出驱动强度。

注：引脚 P1.7 仅在 32 脚器件中可用。

SFR 定义 21.18　P2：端口 2 寄存器

SFR 页：所有页
SFR 地址：0xA0　（可位寻址）

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	

位 7-0：　P2[7:0]：端口 2 数据。
在端口单元被配置为数字 I/O 时，设置端口锁存器逻辑值或读端口引脚的逻辑状态。
写 － 设置端口锁存器逻辑值。
0：设置输出锁存器为逻辑低电平。
1：设置输出锁存器为逻辑高电平。

读 － 读那些被配置为数字 I/O 的端口引脚的逻辑状态。
0：P2.n 端口引脚为逻辑低电平。
1：P2.n 端口引脚为逻辑高电平。

注：引脚 P2.0 ~ P2.6 仅在 32 脚器件中可用。

SFR 定义 21.19　P2SKIP：端口 2 跳过寄存器

SFR 页：0x00
SFR 地址：0xD6

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	

位 7-0：　P2SKIP[7:0]：端口 2 交叉开关跳过使能位。
这些位选择被交叉开关译码器跳过的端口 2 引脚。用作模拟功能、特殊功能或 GPIO 的端口引脚应被交叉开关跳过。
0：对应的 P2.n 不被交叉开关跳过。
1：对应的 P2.n 被交叉开关跳过。

注：引脚 P2.0 ~ P2.6 仅在 32 脚器件中可用。

SFR 定义 21.20　P2MDIN：端口 2 输入方式寄存器

SFR 页：0x00
SFR 地址：0xF3

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
保留								11111111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　保留。读 = 1b，必须写 1b。
位 6-0：　P2MDIN[6:0]：P2.6 – P2.0 的模拟配置位（分别对应）
当端口引脚被配置为模拟方式时，其弱上拉和数字接收器都被禁止。数字驱动器未被显式禁止。
0：对应的 P2.n 引脚被配置为模拟方式。
1：对应的 P2.n 引脚不配置为模拟方式。

注：引脚 P2.0 ~ P2.6 仅在 32 脚器件中可用。

SFR 定义 21.21　P2MDOUT：端口 2 输出方式寄存器

SFR 页：0x00
SFR 地址：0xA6

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	

位 7-0：　P2MDIN[7:0]：P2.7 – P2.0 的输出方式配置位（分别对应）。
这些位控制数字驱动器，即使 P2MDIN 寄存器中的对应位为逻辑 0 也不例外。
0：对应的 P2.n 输出为漏极开路。
1：对应的 P2.n 输出为推挽方式。

注：引脚 P2.0 ~ P2.6 仅在 32 脚器件中可用。

SFR 定义 21.22　P2DRV：端口 2 驱动强度寄存器

SFR 页：0x0F
SFR 地址：0xA6

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	

位 7-0：　P2DRV[7:0]：P2.7 – P2.0 的驱动强度配置位（分别对应）。
配置数字 I/O 端口单元为高或低输出驱动强度。
0：对应的 P2.n 为低输出驱动强度。
1：对应的 P2.n 为高输出驱动强度。

注：引脚 P2.0 ~ P2.6 仅在 32 脚器件中可用。

22. SMBus

SMBus I/O 接口是一个二线的双向串行总线。SMBus 完全符合系统管理总线规范 1.1 版，与 I^2C 串行总线兼容。系统控制器对接口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。在作为主或从器件时，数据传输的最大速率可达系统时钟频率的二十分之一（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。SMBus 外设可以完全由软件驱动（即软件接受/拒绝从地址和产生 ACK），或者被使能为由硬件识别从地址和自动产生 ACK 以减小软件开销。SMBus 外设的原理框图和相关的 SFR 示于图 22.1。

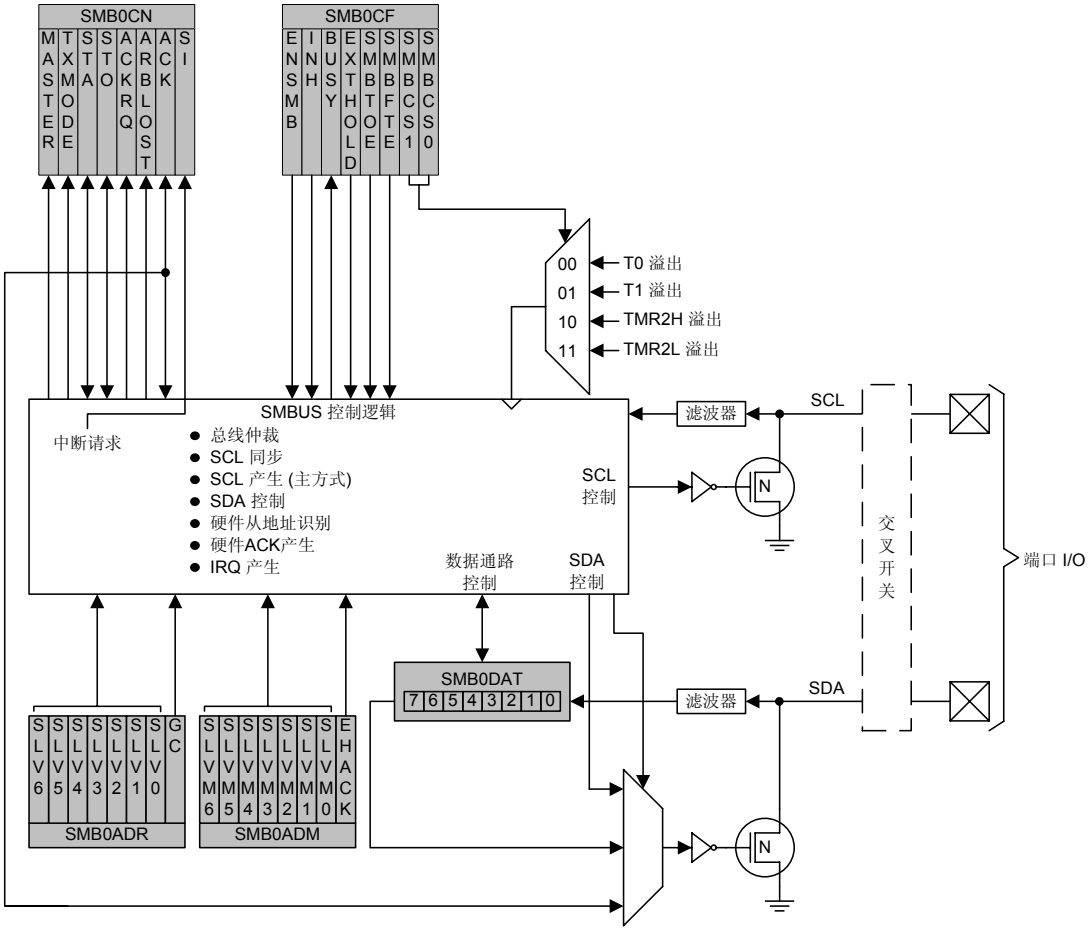


图 22.1　SMBus 原理框图

22.1 支持文档

假设读者熟悉或有条件阅读下列支持文档：

1. I^2C 总线及使用（包含规范），飞利浦半导体。
2. I^2C 总线规范—2.0 版，飞利浦半导体。
3. 系统管理总线规范—1.1 版。

22.2 SMBus 配置

图 22.2 给出了一个典型的 SMBus 配置。SMBus 规范允许的隐性状态电压可以在 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等效电路将它们连到正电源电压。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（隐性状态）。总线上的最大器件数只受规定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。

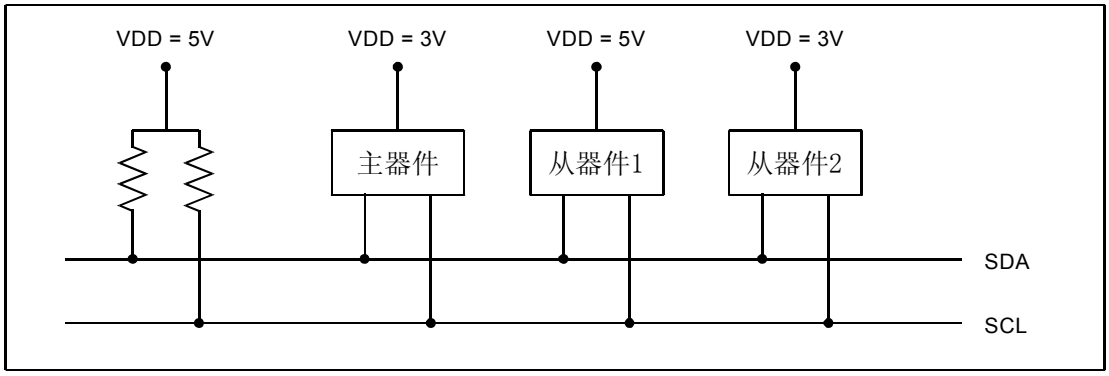


图 22.2　典型的 SMBus 配置

22.3　SMBus 操作

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，主器件还在 SCL 上提供串行时钟。SMBus 接口可以工作在主方式或从方式，同一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 22.3）来确认（ACK）。如果接收器件不确认（ACK），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(R/W)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，然后发送从地址和方向位。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在每个字节结束后等待主器件的确认。在数据传输结束时，主器件产生一个停止条件，结束数据传输并释放总线。图 22.3 示出了一次典型的 SMBus 数据传输过程。

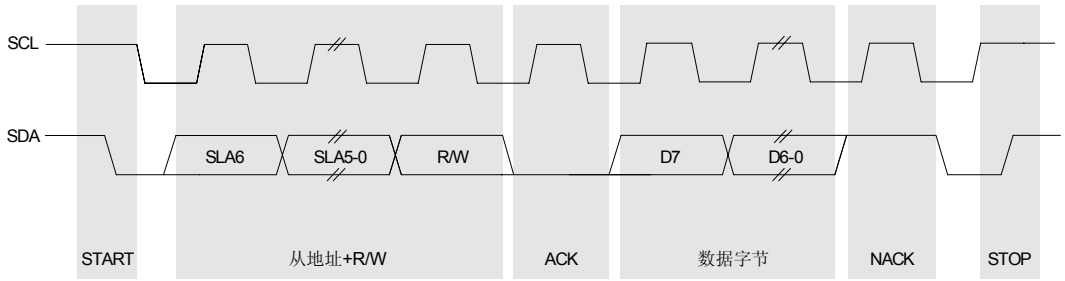


图 22.3 SMBus 数据传输

22.3.1 发送器和接收器的概念

在 SMBus 通信接口中，发送器是指在正在总线上发送一个地址或数据字节到其他器件的器件，而接收器是指总线上的另一器件正在向它发送一个地址或数据字节。发送器在传输地址或数据字节期间控制 SDA 线。在发送器每发送完一个地址或数据信息字节之后，接收器在本次传送的 ACK 阶段发送一个 ACK 或 NACK 位，在此期间接收器控制 SDA 线。

22.3.2 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规定时间（见“22.3.5 SCL 高电平超时”），则总线是空闲的。在两个或多个器件试图同时开始传送的情况下，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。由于总线是漏极开路的，因此被拉为低电平。试图发送高电平的主器件将检测到 SDA 上的低电平而退出竞争。赢得总线的主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，如果被寻址，则在剩下的传输过程中接收数据。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。

22.3.3 时钟低电平扩展

SMBus 提供一种与 I^2C 类似的时钟同步机制，允许不同速度的器件共存于同一个总线上。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。从器件可以临时保持 SCL 线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。

22.3.4 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 以内对通信复位。

当 SMB0CF 中的 SMBTOE 位被置 1 时，定时器 3 被用于检测 SCL 低电平超时。定时器 3 在 SCL 为高电平时被强制重装载，在 SCL 为低电平时开始计数。如果定时器 3 被使能并且溢出周期被配置为 25ms（且 SMBTOE 被置 1），则可在发生 SCL 低电平超时事件时用定时器 3 中断服务程序对 SMBus 复位（禁止后重新使能）。

22.3.5 SCL 高电平（SMBus 空闲）超时

SMBus 规范规定：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。当 SMB0CF 中的 SMBFTE 位被置 1 时，如果 SCL 和 SDA 保持高电平的时间超过 10 个 SMBus 时钟周期，总线将被视为空闲。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。注意：总线空闲超时检测需要一个时钟源，即使对从器件方式也不例外。

22.4　SMBus 的使用

SMBus 可以工作在主方式或从方式。接口电路提供串行传输的时序和移位控制；更高层的协议由用户软件实现。SMBus 接口提供下述与应用无关的特性：

- 以字节为单位的串行数据传输；
- SCL 时钟信号产生（只限于主方式）及 SDA 数据同步；
- 超时/总线错误识别（在配置寄存器 SMB0CF 中定义）；
- START/STOP 定时、检测和产生；
- 总线仲裁；
- 中断产生；
- 状态信息；
- 可选的硬件从地址识别和地址/数据的自动确认。

每次数据字节或从地址传输都产生 SMBus 中断。当硬件确认被禁止时，中断产生的时刻取决于硬件正在作为数据发送者还是作为接收者。当作为发送者时（即发送地址/数据，接收 ACK），该中断产生在 ACK 周期之后，以使软件能读取收到的 ACK 值；当接收数据时（即接收地址/数据，发送 ACK），中断产生在 ACK 周期之前，使软件能确定要发出的 ACK 值。如果硬件确认被使能，这些中断总是产生在 ACK 周期之后。有关传输时序的详细信息见 22.5 节。

主器件产生起始条件时也会产生一个中断，指示数据传输开始；从器件在检测到停止条件时产生一个中断，指示数据传输结束。软件应通过读 SMB0CN（SMBus 控制寄存器）来确定 SMBus 中断的原因。SMB0CN 寄存器的说明见 22.4.2 节。表 22.5 给出了快速 SMB0CN 译码参考。

22.4.1　SMBus 配置寄存器

SMBus 配置寄存器（SMB0CF）用于使能 SMBus 主和/或从方式，选择 SMBus 时钟源和设置 SMBus 定时和超时选项。当 ENSMB 位被置 1 时，SMBus 的所有主和从事件都被允许。可以通过将 INH 位置 1 来禁止从事件。在从事件被禁止的情况下，SMBus 接口仍然监视 SCL 和 SDA 引脚；但在接收到地址时会发出 NACK（非确认）信号，并且不会产生任何从中断。当 INH 被置位时，在下一个起始条件（START）后所有的从事件都将被禁止（当前传输过程的中断将继续）。

SMBCS1-0 位选择 SMBus 时钟源（见表 22.1），时钟源只在主方式或空闲超时检测被使能时使用。当 SMBus 接口工作在主方式时，所选择的时钟源的溢出周期决定 SCL 低电平和高电平的最小时间，该最小时间由方程 22.1 给出。注意，SMBus 可以与其它外设共享该时钟源，前提是时钟源定时器一直保持运行状态。例如，定时器 1 溢出可以同时用于产生 SMBus 和 UART 波特率。有关定时器配置见“25. 定时器”。

表 22.1　SMBus 时钟源选择

SMBCS1	SMBCS0	SMBus 时钟源
0	0	定时器 0 溢出
0	1	定时器 1 溢出
1	0	定时器 2 高字节溢出
1	1	定时器 2 低字节溢出

$$T_{HighMin} = T_{LowMin} = \frac{1}{f_{ClockSourceOverflow}}$$

方程 22.1 最小 SCL 高和低电平时间

其中 $T_{HighMin}$ 为最小 SCL 高电平时间，T_{LowMin} 为最小 SCL 低电平时间，$f_{ClockSourceOverflow}$ 为时钟源的溢出频率。

所选择的时钟源应被配置为能产生由方程 22.1 所定义的最小 SCL 高电平和低电平时间。当接口工作在主方式时（并且 SCL 不被总线上的任何其它器件驱动或扩展），典型的 SMBus 位速率可由方程 22.2 估算：

$$\text{位速率} = \frac{f_{ClockSourceOverflow}}{3}$$

方程 22.2 典型 SMBus 位速率

图 22.4 给出了由方程 22.2 定义的典型 SCL 波形。注意，T_{HIGH} 通常为 T_{LOW} 的两倍。实际的 SCL 输出波形可能会因总线上有其它器件而发生改变（SCL 可能被低速从器件扩展低电平，或被其它参与竞争的主器件驱动为低电平）。当工作在主方式时，位速率不能超过由方程 22.1 定义的极限值。

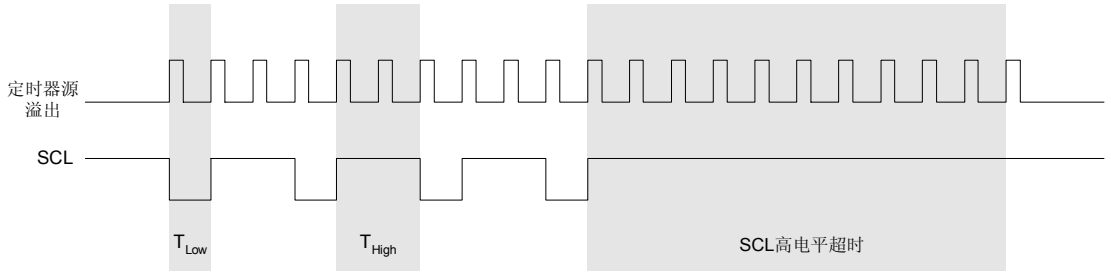


图 22.4 典型的 SMBus SCL 波形

设置 EXTHOLD 位为逻辑 1 将扩展 SDA 线的最小建立时间和保持时间。最小 SDA 建立时间定义了在 SCL 上升沿到来之前 SDA 的最小稳定时间。最小 SDA 保持时间定义了在 SCL 下降沿过去之后当前 SDA 值继续保持稳定的最小时间。SMBus 规定的最小建立和保持时间分别为 250ns 和 300ns。必要时应将 EXTHOLD 位置 1，以保证最小建立和保持时间符合 SMBus 规范。表 22.2 列出了对应两种 EXTHOLD 设置情况的最小建立和保持时间。当 SYSCLK 大于 10MHz 时，通常需要扩展建立和保持时间。

表 22.2　最小 SDA 建立和保持时间

EXTHOLD	最小 SDA 建立时间	最小 SDA 保持时间
0	T_{low} – 4 个系统时钟 或 1 个系统时钟 + 软件延时*	3 个系统时钟
1	11 个系统时钟	12 个系统时钟
*注：发送 ACK 位和所有数据传输中 MSB 的建立时间。当使用软件确认时，软件延时发生在写 SMB0DAT 或 ACK 到 SI 被清除之间。注意，如果写 ACK 和清除 SI 发生在同一个写操作，则软件延时为 0。		

在 SMBTOE 位被置 1 的情况下，定时器 3 应被配置为以 25ms 为周期溢出，以检测 SCL 低电平超时（见“22.3.4 SCL 低电平超时”）。SMBus 接口在 SCL 为高电平时强制重装载定时器 3，并允许定时器 3 在 SCL 为低电平时开始计数。应使用定时器 3 中断服务程序对 SMBus 通信复位，这可通过先禁止然后再重新使能 SMBus 接口来实现。

通过将 SMBFTE 位置 1 来使能 SMBus 空闲超时检测。当该位被置 1 时，如果 SCL 和 SDA 保持高电平的时间超过 10 个 SMBus 时钟周期（见图 22.4），总线将被视为空闲。

SFR 定义 22.1　SMB0CF：SMBus 配置寄存器

SFR 页：0x00
SFR 地址：0xC1

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	复位值
ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS1	SMBCS0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　ENSMB：SMBus 使能
该位使能/禁止 SMBus 接口。当被使能时，接口一直监视 SDA 和 SCL 引脚。
0：禁止 SMBus 接口。
1：使能 SMBus 接口。

位 6：　INH：SMBus 从禁止
当该位被设置为逻辑 1 时，SMBus 接口在有从事件发生时不产生从事件中断。这实际上相当于将 SMBus 从器件移出总线。主方式中断不受影响。
0：SMBus 从方式使能。
1：SMBus 从方式禁止。

位 5：　BUSY：SMBus 忙状态标志
当正在进行一次传输时，该位由硬件置 1。当检测到停止条件或空闲超时时，该位被清 0。

位 4：　EXTHOLD：SMBus 建立和保持时间扩展允许
该位控制 SDA 的建立和保持时间（根据表 22.2）。
0：禁止 SDA 建立和保持时间扩展。
1：允许 SDA 建立和保持时间扩展。

位 3：　SMBTOE：SMBus SCL 超时检测允许位
该位使能 SCL 低电平超时检测。当被置 1 时，SMBus 接口在 SCL 为高电平时强制重装载定时器 3，并允许定时器 3 在 SCL 为低电平时开始计数。如果定时器 3 被配置为分割方式（两个 8 位计数器，T3SPLIT 被置 1），则当 SCL 为高电平时，只有定时器 3 的高字节被重装载。应将定时器 3 编程为每 25ms 产生一次中断，并使用定时器 3 中断服务程序对 SMBus 通信复位。

位 2：　SMBFTE：SMBus 空闲超时检测允许位
当该位被置 1 时，如果 SCL 和 SDA 保持高电平的时间超过 10 个 SMBus 时钟周期，总线将被视为空闲。

位 1-0：　SMBCS1-SMBCS0：SMBus 时钟源选择位
这两位选择用于产生 SMBus 位速率的时钟源。应根据方程 22.1 配置所选时钟源。

SMBCS1	SMBCS0	SMBus 时钟源
0	0	定时器 0 溢出
0	1	定时器 1 溢出
1	0	定时器 2 高字节溢出
1	1	定时器 2 低字节溢出

22.4.2　SMBus 控制寄存器

SMBus 控制寄存器（SMB0CN）用于控制 SMBus 接口和提供状态信息（见 SFR 定义 22.2）。SMB0CN 中的高 4 位（MASTER、TXMODE、STA 和 STO）组成一个状态向量，可利用该状态向量转移到中断服务程序。MASTER 指示在当前的传送期间器件工作在主方式还是从方式。TXMODE 指示对于当前字节器件是在发送数据还是接收数据。

STA 和 STO 指示自上次 SMBus 中断以来检测到或产生了一个起始（START）和/或停止条件（STOP）。当 SMBus 工作在主方式时，STA 和 STO 还用于产生起始和停止条件。当总线变成空闲时（在起始条件产生后，STA 未被硬件清 0），向 STA 写 1 将使 SMBus 接口进入主方式并产生一个起始条件。在主方式，向 STO 写 1 将使 SMBus 接口产生一个停止条件，并在下一个 ACK 周期后结束当前的数据传输。如果 STA 和 STO 都被置位（在主方式），则产生一个停止条件后再产生一个起始条件。

ARBLOST 位指示 SMBus 接口是否在一次总线竞争中失败。当接口正在发送时（主或从方式），可能会出现这种情况。当工作在从方式时，出现这种情况表示发生了总线错误条件。在每次 SI 被清除后，ARBLOST 被硬件清除。

在每次传输的开始和结束、每个字节帧之后或竞争失败时，SI 位（SMBus 中断标志）被硬件置 1，详见表 22.3。

有关 SI 位的重要注意事项：当 SI 标志被置 1 时，SMBus 接口被冻结；SCL 线被保持为低电平，总线状态被冻结，直到 SI 被软件清 0 为止。

22.4.2.1　软件 ACK 产生

当寄存器 SMB0ADM 中的 EHACK 位被清 0 时，器件的固件必须检查接收的从地址，并对该从地址和接收的数据字节发 ACK 或 NACK。当 SMBus 接口作为接收器时，写 ACK 位定义要发出的 ACK 值；当作为发送器时，读 ACK 位将返回最后一个 ACK 周期的接收值。ACKRQ 在每接收到一个字节后置位，表示需要写待发出的 ACK 值。当 ACKRQ 置位时，软件应在清除 SI 之前向 ACK 位写入要发出的 ACK 值。如果在清除 SI 之前软件未写 ACK 位，接口电路将产生一个 NACK。在向 ACK 位写入后，SDA 线将立即出现所定义的 ACK 值；但 SCL 将保持低电平，直到 SI 被清除。如果接收的从地址未被确认，则以后的从事件将被忽略，直到检测到下一个起始条件。

22.4.2.2　硬件 ACK 产生

当寄存器 SMB0ADM 中的 EHACK 位被置 1 时，自动从地址识别和 ACK 产生被使能。有关自动从地址识别的更详细的信息可以在 22.4.3 节找到。作为接收器时，ACK 位的当前值会在输入数据字节的 ACK 周期被自动发送；当作为发送器时，读 ACK 位将返回最后一个 ACK 周期的接收值。当硬件 ACK 产生被使能时，不使用 ACKRQ 位。如果接收的从地址被硬件否认（NACK），则以后的从事件将被忽略，直到检测到下一个起始条件，并且不产生中断。

表 22.3 列出了影响 SMB0CN 寄存器中各个位的硬件变化源。有关使用 SMB0CN 寄存器进行 SMBus 的状态译码的详细信息见表 22.5。

SFR 定义 22.2 SMB0CN：SMBus 控制寄存器

SFR 页：0x00
SFR 地址：0xC0　　（可位寻址）

R	R	R/W	R/W	R	R	R/W	R/W	复位值
MASTER	TXMODE	STA	STO	ACKRQ	ARBLOST	ACK	SI	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　MASTER：SMBus 主/从标志
该只读位指示 SMBus 是否工作在主方式。
0：SMBus 工作在从方式。
1：SMBus 工作在主方式。

位 6：　TXMODE：SMBus 发送方式标志
该只读位指示 SMBus 是否工作在发送器方式。
0：SMBus 工作在接收器方式。
1：SMBus 工作在发送器方式。

位 5：　STA：SMBus 起始标志
写：
0：不产生起始条件。
1：当配置为主方式时，发送出一个起始条件或重复起始条件。
读：
0：未检测到起始条件或重复起始条件。1：检测到起始条件或重复起始条件。

位 4：　STO：SMBus 停止标志
写：
0：不发送停止条件。
1：当配置为主方式时，导致发送一个停止条件（在下一个 ACK 周期之后）。由硬件清除。
读：
0：未检测到停止条件。1：检测到停止条件（在从方式）或挂起（在主方式）。

位 3：　ACKRQ：SMBus 确认请求
0：无 ACK 请求。
1：有 ACK 请求。

位 2：　ARBLOST：SMBus 竞争失败标志
0：无仲裁错误。
1：竞争失败。

位 1：　ACK：SMBus 确认位
写：
0：发送 NACK。1：发送 ACK。
读：
0：收到 NACK。1：收到 ACK。

位 0：　SI：SMBus 中断标志
当出现表 22.3 列出的条件时该位被硬件置 1。SI 必须用软件清除。当 SI 被置 1 时，SCL 被保持低电平，SMBus 被冻结。
写：
0：清除中断并启动下一个状态机事件。1：强制中断。
读：
0：无中断请求。1：有中断请求。

表 22.3 影响 SMB0CN 的硬件变化源

位	在下述情况被硬件置 1：	在下述情况被硬件清 0：
MASTER	• 产生了起始条件。	• 产生了停止条件。 • 竞争失败。
TXMODE	• 产生了起始条件。 • 在一个 SMBus 帧开始之前写了 SMB0DAT。	• 检测到起始条件。 • 竞争失败。 • 在一个 SMBus 帧开始之前未写 SMB0DAT。
STA	• 在起始条件后接收到一个地址字节。	• 必须用软件清除。
STO	• 在作为从器件被寻址时检测到一个停止条件。 • 因检测到停止条件而导致竞争失败。	• 产生了一个挂起的停止条件。
ACKRQ	• 接收到一个字节并需要一个 ACK 响应值。（仅在硬件 ACK 未被使能时）。	• 每个 ACK 周期之后。
ARBLOST	• 当 STA 为低时，主器件检测到一个重复起始条件（不希望的重复起始条件）。 • 在试图产生一个停止条件或重复起始条件时检测到 SCL 为低电平。 • 在试图发送 1 时检测到 SDA 为低电平（ACK 位除外）。	• 每次 SI 被清除时。
ACK	• 输入的 ACK 值为低（确认）。	• 输入的 ACK 值为高（非确认）。
SI	• 产生了一个起始条件。 • 竞争失败。 • 发送了一个字节并收到一个 ACK/NACK • 收到一个字节。 • 在起始条件或重复起始条件之后接收到一个从地址+R/W。 • 收到一个停止条件。	•必须用软件清除。

22.4.3 硬件从地址识别

SMBus 具有硬件自动识别进入的从地址和发送 ACK 的能力，无需软件干预。通过将寄存器 SMB0ADM 中的 EHACK 位置 1 来使能自动从地址识别。这将同时使能自动从地址识别和自动硬件产生接收字节的 ACK（主或从方式）。有关自动硬件 ACK 产生的详细信息见 22.4.2.2 节。

用于定义要被硬件识别的地址的寄存器是 SMBus 从地址寄存器（SFR 定义 22.3）和 SMBus 从地址掩码寄存器（SFR 定义 22.4）。用这两个寄存器可以指定单个地址或一个地址范围（包括全局呼叫地址 0x00）。这两个寄存器的高 7 位用于定义要被确认的地址。从地址掩码 SLVM[6:0]中为 1 的位使能接收的从地址与硬件从地址 SLV[6:0]中对应位的比较。从地址掩码中某一位为 0 表示该位在比较时被忽略。在这种情况下，不管进入的从地址的对应位是 1 还是 0，该地址都是可接受的。另外，当寄存器 SMB0ADR 中的 GC 位被置 1 时，硬件会识别全局呼叫地址（0x00）。表 22.4 列出了一些示例参数设置和在这些条件下能被硬件识别从地址。

表 22.4 硬件地址识别示例（EHACK = 1）

硬件从地址 SLV[6:0]	从地址掩码 SLVM[6:0]	GC 位	被硬件识别的从地址
0x34	0x7F	0	0x34
0x34	0x7F	1	0x34, 0x00（全局呼叫）
0x34	0x7E	0	0x34, 0x35
0x34	0x7E	1	0x34, 0x35, 0x00（全局呼叫）
0x70	0x73	0	0x70, 0x74, 0x78, 0x7C

SFR 定义 22.3　SMB0ADR：SMBus 从地址寄存器

SFR 页：0x00
SFR 地址：0xF4

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	

位 7-1：　SLV[6:0]：SMBus 硬件从地址
定义用于自动硬件确认的 SMBus 从地址。只有 SLVM[6:0]中的对应位为 1 的那些地址位与接收的地址比较。这样就允许多个地址被识别。

位 0：　GC：全局呼叫地址使能
当硬件地址识别被使能时（EHACK = 1），该位决定硬件是否也识别全局呼叫地址（0x00）。
0：全局呼叫地址被忽略。
1：全局呼叫地址被识别。

SFR 定义 22.4　SMB0ADM：SMBus 从地址掩码寄存器

SFR 页：0x00
SFR 地址：0xF5

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SLVM6	SLVM5	SLVM4	SLVM3	SLVM2	SLVM1	SLVM0	EHACK	11111110
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-1：　SLVM[6:0]：SMBus 从地址掩码
定义寄存器 SMB0ADR 中哪些位与接收的地址字节比较，哪些位被忽略。SLVM[6:0]中任何被置 1 的位使能与 SLV[6:0]中对应位的比较。值为 0 的那些位被忽略（这些位在接收的地址中可以是 0，也可以是 1）。

位 0：　EHACK：硬件确认使能
使能硬件对从地址和所接收的数据字节的确认。
0：固件必须手动确认所有进入的地址和数据字节。
1：自动从地址识别和硬件确认被使能。

22.4.4 数据寄存器

SMBus 数据寄存器 SMB0DAT 保存要发送或刚接收的串行数据字节。在 SI 标志被置 1 时，软件可以安全地读/写数据寄存器。当 SMBus 被使能但 SI 标志被清为逻辑 0 时软件不应访问 SMB0DAT 寄存器，因为接口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。在竞争失败后，从主发送器变为从接收器时 SMB0DAT 中的数据或地址保持不变。

SFR 定义 22.5　SMB0DAT：SMBus 数据寄存器

SFR 页：0x00
SFR 地址：0xC2

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	

位 7-0：　SMB0DAT[7:0]：SMBus 数据
SMB0DAT 寄存器保存要发送到 SMBus 串行接口上的一个数据字节，或刚从 SMBus 串行接口接收到的一个字节。一旦 SI 串行中断标志被置 1，CPU 即可读或写该寄存器。只要 SI 串行中断标志位（SMB0CN.0）为逻辑 1，该寄存器内的串行数据就是稳定的。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。

22.5　SMBus 传输方式

SMBus 接口可以被配置为工作在主方式和/或从方式。在任一时刻，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。SMBus 在产生起始条件时进入主方式，并保持在该方式直到产生一个停止条件或在总线竞争中失败。SMBus 在每个字节帧结束后都产生一个中断。注意：在作为接收器工作时，ACK 中断的位置取决于硬件 ACK 产生是否被使能。作为接收器时，在硬件 ACK 产生被禁止的情况下，ACK 中断发生在 ACK 之前；在硬件 ACK 产生被使能的情况下，ACK 中断发生在 ACK 之后。作为发送器时中断发生在 ACK 之后，与硬件 ACK 产生是否被使能无关。

22.5.1 写序列（主方式）

在一次写序列期间，SMBus 主器件向从器件写数据。这次传送中的主器件在地址字节和所有数据字节期间都将作为发送器。SMBus 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（WRITE），表示这是一个写操作。主器件接着发送一个或多个字节的串行数据。在每发送一个字节后，从器件产生一个确认位。当 STO 位被置 1 并产生一个停止条件后，本次传送结束。注意，如果在发生主发送器中断后没有向 SMB0DAT 写入数据，则接口将切换到主接收器方式。图 22.5 给出了典型的主器件写序列，只给出了发送两个字节的传输时序，尽管可以发送任意多个字节。注意，在该方式下，所有的“数据字节传输结束”中断都发生在 ACK 周期之后，与硬件 ACK 产生是否被使能无关。

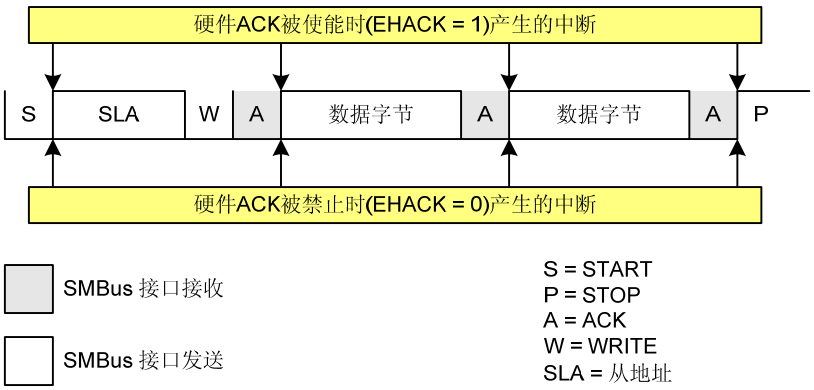


图 22.5　典型的主器件写序列

22.5.2 读序列（主方式）

在一次读序列期间，SMBus 主器件从一个从器件读数据。这次传送中的主器件在地址字节期间为发送器，在所有的数据字节期间都将作为接收器。SMBus 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（READ），表示这是一个读操作。然后，SMBus 接口接收 SDA 线上来自从器件的串行数据，并输出 SMBus 串行时钟。从器件发送一个或多个字节的串行数据。

如果硬件 ACK 产生被禁止，则每接收到一个字节后 ACKRQ 位被置 1 并产生中断。此时软件必须写 ACK 位以确认（ACK）或否认（NACK）所接收的字节。

在硬件 ACK 产生被使能的情况下，SMBus 硬件会自动产生 ACK/NACK，并产生中断。**注意：当硬件 ACK 产生被使能时，软件应在接收字节前写 ACK 或 NACK 值。**

向 ACK 位写 1 产生一个 ACK，写 0 产生一个 NACK。软件应在最后一次数据传送时向 ACK 位写 0，以发送 NACK。接口电路将在对 STO 位置 1 并产生一个停止条件后退出主接收器方式。在主接收器方式，如果执行了 SMB0DAT 写操作，接口将切换到主发送器方式。图 22.6 给出了典型的主方式读序列，只给出了接收两个字节的传输时序，尽管可以接收任意多个字节。注意，在该方式下，“数据字节传输结束”中断可发生在序列中的不同位置，取决与硬件 ACK 产生是否被使能。当硬件 ACK 产生被禁止时，中断发生在 ACK 之前；当硬件 ACK 产生被使能时，中断发生在 ACK 之后。

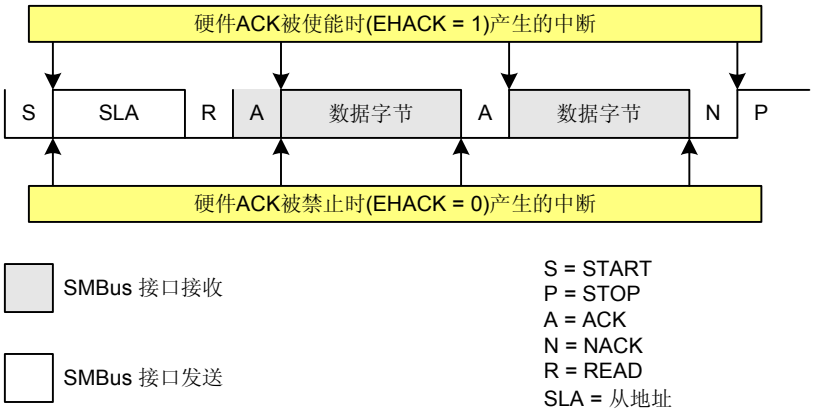


图 22.6　典型的主器件读序列

22.5.3 写序列（从方式）

在一次写序列期间，SMBus 主器件向从器件写数据。这次传送中的从器件在地址字节期间为接收器，在所有的数据字节期间作为接收器。在从事件被允许的情况下（INH = 0），当接收到一个起始条件（START）和一个含有从地址和数据方向位（此处应为写）的字节时，SMBus 接口进入从接收器方式。如果硬件 ACK 产生被禁止，在进入从接收器方式时将产生一个中断，并且 ACKRQ 被置 1。软件必须用一个 ACK 对接收到的从地址确认，或用一个 NACK 忽略接收到的从地址。如果硬件 ACK 产生被使能，硬件会对一个匹配的从地址产生 ACK，该地址要与由 SMB0ADR 和 SMB0ADM 设置的判别条件匹配。中断发生在 ACK 周期之后。

如果接收到的从地址被忽略（由硬件或软件），从事件中断将被禁止，直到检测到下一个起始条件。如果收到的从地址被确认，将接收 0 个或多个字节的数据。

如果硬件 ACK 产生被禁止，则每接收到一个字节后 ACKRQ 位被置 1 并产生中断。此时软件必须写 ACK 位以确认（ACK）或否认（NACK）所接收的字节。

在硬件 ACK 产生被使能的情况下，SMBus 硬件会自动产生 ACK/NACK，并产生中断。**注意：当硬件 ACK 产生被使能时，软件应在接收字节前写 ACK 或 NACK 值。**

在收到一个停止条件后，SMBus 接口退出从接收器方式。注意，如果在从接收器方式对 SMB0DAT 进行写操作，接口将切换到从发送器方式。图 22.7 给出了典型的从方式写序列，只给出了接收两个字节的传输时序，尽管可以接收任意多个字节。注意，在该方式下，“数据字节传输结束”中断可发生在序列中的不同位置，取决与硬件 ACK 产生是否被使能。当硬件 ACK 产生被禁止时，中断发生在 ACK 之前；当硬件 ACK 产生被使能时，中断发生在 ACK 之后。

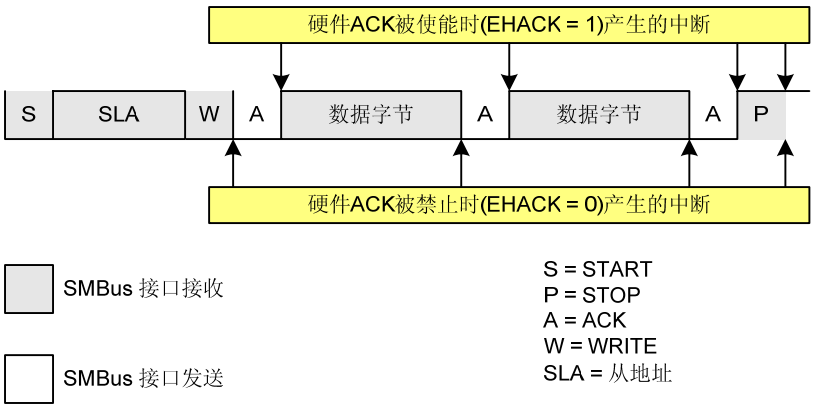


图 22.7　典型的从器件写序列

22.5.4 读序列（从方式）

在一次读序列期间，SMBus 主器件从一个从器件读数据。这次传送中的从器件在地址字节期间为接收器，在所有的数据字节期间都将作为发送器。在从事件被允许的情况下（INH = 0），当接收到一个起始条件（START）和一个含有从地址和数据方向位（此处应为读）的字节时，SMBus 接口进入从接收器方式（接收从地址）。如果硬件 ACK 产生被禁止，在进入从接收器方式时将产生一个中断，并且 ACKRQ 被置 1。软件必须用一个 ACK 对接收到的从地址确认，或用一个 NACK 忽略接收到的从地址。如果硬件 ACK 产生被使能，硬件会对一个匹配的从地址产生 ACK，该地址要与由 SMB0ADR 和 SMB0ADM 设置的判别条件匹配。中断发生在 ACK 周期之后。

如果接收到的从地址被忽略（由硬件或软件），从事件中断将被禁止，直到检测到下一个起始条件。如果收到的从地址被确认，将发送 0 个或多个字节的数据。如果收到的从地址被确认，软件应向 SMB0DAT 写入待发送的数据，SMBus 接口进入从发送器方式，并发送一个或多个字节的数据。在每发送一个字节后，主器件发送一个确认位。如果确认位为 ACK，应向 SMB0DAT 写入下一个数据字节；如果确认位为 NACK，在 SI 被清除前不应再写 SMB0DAT（注：在从发送器方式，如果在收到 NACK 后写 SMB0DAT，将会导致一个错误条件）。在收到停止条件后，SMBus 接口退出从发送器方式。注意，如果在一个从发送器中断发生之后没有对 SMB0DAT 进行写操作，接口将切换到从接收器方式。图 22.8 给出了典型的从方式读序列，只给出了发送两个字节的传输时序，尽管可以发送任意多个字节。注意，在该方式下“数据字节传输结束”中断发生在 ACK 周期之后，与硬件 ACK 产生是否被使能无关。

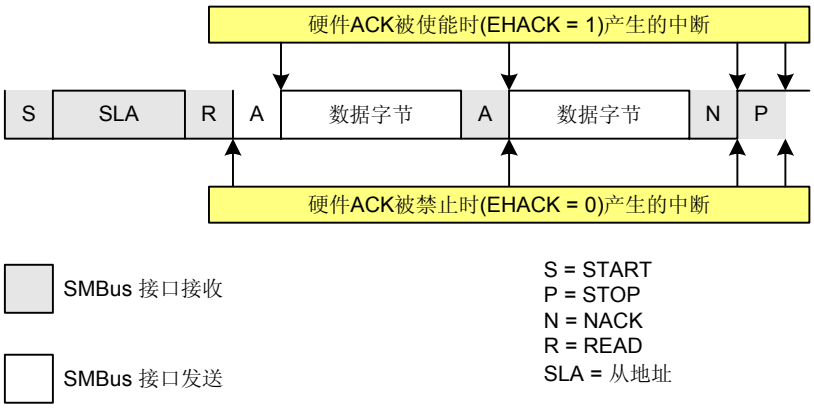


图 22.8　典型的从器件读序列

22.6　SMBus 状态译码

用 SMB0CN 寄存器可以很容易地对 SMBus 的当前状态译码。响应一个 SMBus 事件要采取的合适动作取决于硬件从地址识别和 ACK 产生是被使能还是被禁止。表 22.5 中描述了硬件从地址识别和 ACK 产生被禁止时的典型动作。表 22.6 中描述了硬件从地址识别和 ACK 产生被使能时的典型动作。在这两个表中，状态向量指的是 SMB0CN 中的高 4 位：MASTER、TXMODE、STA 和 STO。注意，表中只列出了典型的响应选项。只要符合 SMBus 规范，特定应用过程是允许的。表中被突出显示的响应选项是硬件允许的，但不符合 SMBus 规范。

表 22.5　硬件 ACK 产生被禁止时的 SMBus 状态译码（EHACK = 0）

方式	读取值				SMBus 的当前状态	典型响应选项	写入值			下一个期望的状态向量
	状态向量	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
主发送器	1110	0	0	X	起始条件已发出。	将从地址+R/W 装入到 SMB0DAT。	0	0	X	1100
	1100	0	0	0	数据或地址字节已发出；收到 NACK。	置位 STA 以重新启动数据传输。	1	0	X	1110
						放弃发送。	0	1	X	-
		0	0	1	数据或地址字节已发出；收到 ACK。	将下一数据字节装入到 SMB0DAT。	0	0	X	1100
						用停止条件结束数据传输。	0	1	X	-
						用停止条件结束数据传输并开始另一次传输。	1	1	X	-
						发送重复起始条件。	1	0	X	1110
						切换到主接收器方式（清除 SI，不向 SMB0DAT 写新数据）。	0	0	X	1000
主接收器	1000	1	0	X	收到数据字节；请求 ACK。	确认接收字节；读 SMB0DAT。	0	0	1	1000
						发 NACK，表示这是最后一个字节，发停止条件。	0	1	0	-
						发 NACK，表示这是最后一个字节，接着发停止条件，再发起始条件。	1	1	0	1110
						发 ACK 后再发重复起始条件。	1	0	1	1110
						发 NACK，表示这是最后一个字节，接着发重复起始条件。	1	0	0	1110
						发 ACK 并切换到主发送器方式（在清除 SI 之前写 SMB0DAT）。	**0**	**0**	**1**	**1100**
						发 NACK 并切换到主发送器方式（在清除 SI 之前写 SMB0DAT）。	**0**	**0**	**0**	**1100**

表 22.5　硬件 ACK 产生被禁止时的 SMBus 状态译码（EHACK = 0）（续）

方式	读取值				SMBus 的当前状态	典型响应选项	写入值			下一个期望的状态向量
	状态向量	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
从发送器	0100	0	0	0	字节已发送；收到 NACK。	不需任何操作（等待停止条件）。	0	0	X	0001
		0	0	1	字节已发送；收到 ACK。	将下一个要发送的数据字节装入到 SMB0DAT。	0	0	X	0100
		0	1	X	字节已发送；检测到错误。	不需任何操作（等待主器件结束传输）。	0	0	X	0001
	0101	0	X	X	从发送期间检测到非法停止条件或总线错误。	清 STO。	0	0	X	-
从接收器	0010	1	0	X	接收到从地址+R/W；请求 ACK。	如果写，对接收到的地址进行确认。	0	0	1	0000
						如果读，向 SMB0DAT 装入数据字节；对接收到的地址进行确认。	0	0	1	0100
						否认接收到的地址。	0	0	0	-
		1	1	X	作为主器件竞争失败；收到从地址+R/W；请求 ACK。	如果写，对接收到的地址进行确认。	0	0	1	0000
						如果读，向 SMB0DAT 装入数据字节；对接收到的地址进行确认。	0	0	1	0100
						否认接收到的地址。	0	0	0	-
						重新安排失败的传输；否认接收到的地址。	1	0	0	1110
	0001	0	0	X	作为从发送器或从接收器被寻址时，检测到停止条件。	清 STO。	0	0	X	-
		1	1	X	试图发送停止条件时竞争失败。	不需任何操作（传输完成/放弃）。	0	0	0	-
	0000	1	0	X	接收到从字节；请求 ACK。	确认接收的字节；读 SMB0DAT。	0	0	1	0000
						否认接收到的字节。	0	0	0	-
总线错误条件	0010	0	1	X	试图发送重复起始条件时竞争失败。	放弃失败的传输。	0	0	X	-
						重新安排失败的传输。	1	0	X	1110
	0001	0	1	X	由于检测到停止条件而竞争失败。	放弃失败的传输。	0	0	X	-
						重新安排失败的传输。	1	0	X	1110
	0000	1	1	X	试图作为主器件发送数据字节时竞争失败。	放弃失败的传输。	0	0	0	-
						重新安排失败的传输。	1	0	0	1110

表 22.6　硬件 ACK 产生被使能时的 SMBus 状态译码（EHACK = 1）

方式	读取值				SMBus 的当前状态	典型响应选项	写入值			下一个期望的状态向量
	状态向量	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
主发送器	1110	0	0	X	起始条件已发出。	将从地址+R/W 装入到 SMB0DAT。	0	0	X	1100
	1100	0	0	0	数据或地址字节已发出；收到 NACK。	置位 STA 以重新启动数据传输。	1	0	X	1110
						放弃发送。	0	1	X	-
		0	0	1	数据或地址字节已发出；收到 ACK。	将下一数据字节装入到 SMB0DAT。	0	0	X	1100
						用停止条件结束数据传输。	0	1	X	-
						用停止条件结束数据传输并开始另一次传输。	1	1	X	-
						发送重复起始条件。	1	0	X	1110
						切换到主接收器方式（清除 SI，不向 SMB0DAT 写新数据）。为初始数据字节置位 ACK。	0	0	1	1000
主接收器	1000	0	0	1	收到数据字节；已发送 ACK。	发 ACK 准备接收下一数据字节；读 SMB0DAT。	0	0	1	1000
						发 NACK，表示下一数据字节是最后一个数据字节；读 SMB0DAT。	0	0	0	1000
						发送一个重复起始条件	1	0	0	1110
						切换到主发送器方式（在清除 SI 之前写 SMB0DAT）。	**0**	**0**	**X**	**1100**
		0	0	0	收到数据字节；已发送 NACK（最后一个字节）。	读 SMB0DAT，发送停止条件。	0	1	0	-
						读 SMB0DAT，发送停止条件后再发起始条件。	1	1	0	1110
						发送重复起始条件。	1	0	0	1110
						切换到主发送器方式（在清除 SI 之前写 SMB0DAT）。	**0**	**0**	**X**	**1100**

表 22.6　硬件 ACK 产生被使能时的 SMBus 状态译码（EHACK = 1）（续）

方式	读取值				SMBus 的当前状态	典型响应选项	写入值			下一个期望的状态向量
	状态向量	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
从发送器	0100	0	0	0	字节已发送；收到 NACK。	不需任何操作（等待停止条件）。	0	0	X	0001
		0	0	1	字节已发送；收到 ACK。	将下一个要发送的数据字节装入到 SMB0DAT。	0	0	X	0100
		0	1	X	字节已发送；检测到错误。	不需任何操作（等待主器件结束传输）。	0	0	X	0001
	0101	0	X	X	从发送期间检测到非法停止条件或总线错误。	清 STO。	0	0	X	-
从接收器	0010	0	0	X	接收到从地址+R/W；ACK 已发送。	如果写，设置第一个数据字节的 ACK。	0	0	1	0000
						如果读，向 SMB0DAT 装入数据字节。	0	0	X	0100
		0	1	X	作为主器件竞争失败；收到从地址+R/W；ACK 已发送。	如果写，设置第一个数据字节的 ACK。	0	0	1	0000
						如果读，向 SMB0DAT 装入数据字节。	0	0	X	0100
						重新安排失败的传输。	1	0	X	1110
	0001	0	0	X	作为从发送器或从接收器被寻址时，检测到停止条件。	清 STO。	0	0	X	-
		0	1	X	试图发送停止条件时竞争失败。	不需任何操作（传输完成/放弃）。	0	0	0	-
	0000	0	0	X	接收到从字节。	为下一数据字节设置 ACK；读 SMB0DAT。	0	0	1	0000
						为下一数据字节设置 NACK；读 SMB0DAT。	0	0	0	0000
总线错误条件	0010	0	1	X	试图发送重复起始条件时竞争失败。	放弃失败的传输。	0	0	X	-
						重新安排失败的传输。	1	0	X	1110
	0001	0	1	X	由于检测到停止条件而竞争失败。	放弃失败的传输。	0	0	X	-
						重新安排失败的传输。	1	0	X	1110
	0000	0	1	X	试图作为主器件发送数据字节时竞争失败。	放弃失败的传输。	0	0	X	-
						重新安排失败的传输。	1	0	X	1110

23. UART0

UART0 是一个异步、全双工串口，它提供标准 8051 UART 的方式 1 和方式 3。增强的波特率支持允许使用宽范围的时钟源来产生标准波特率（详见“23.1 增强波特率产生”）。接收数据缓冲机制允许 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。

UART0 有两个相关的特殊功能寄存器：串行控制寄存器（SCON0）和串行数据缓冲器（SBUF0）。用同一个 SBUF0 地址可以访问发送寄存器和接收寄存器。**写 SBUF0 时总是访问发送寄存器；读 SBUF0 时总是访问接收寄存器，不可能从发送寄存器中读数据。**

如果 UART0 中断被允许，则每次发送完成（SCON0 中的 TI0 位被置 1）或接收到一个数据字节（SCON0 中的 RI0 位被置 1）时将产生中断。当 CPU 转向中断服务程序时硬件不清除 UART0 中断标志。中断标志必须用软件清除，这就允许软件判断 UART0 中断的原因（发送完成或接收完成）。

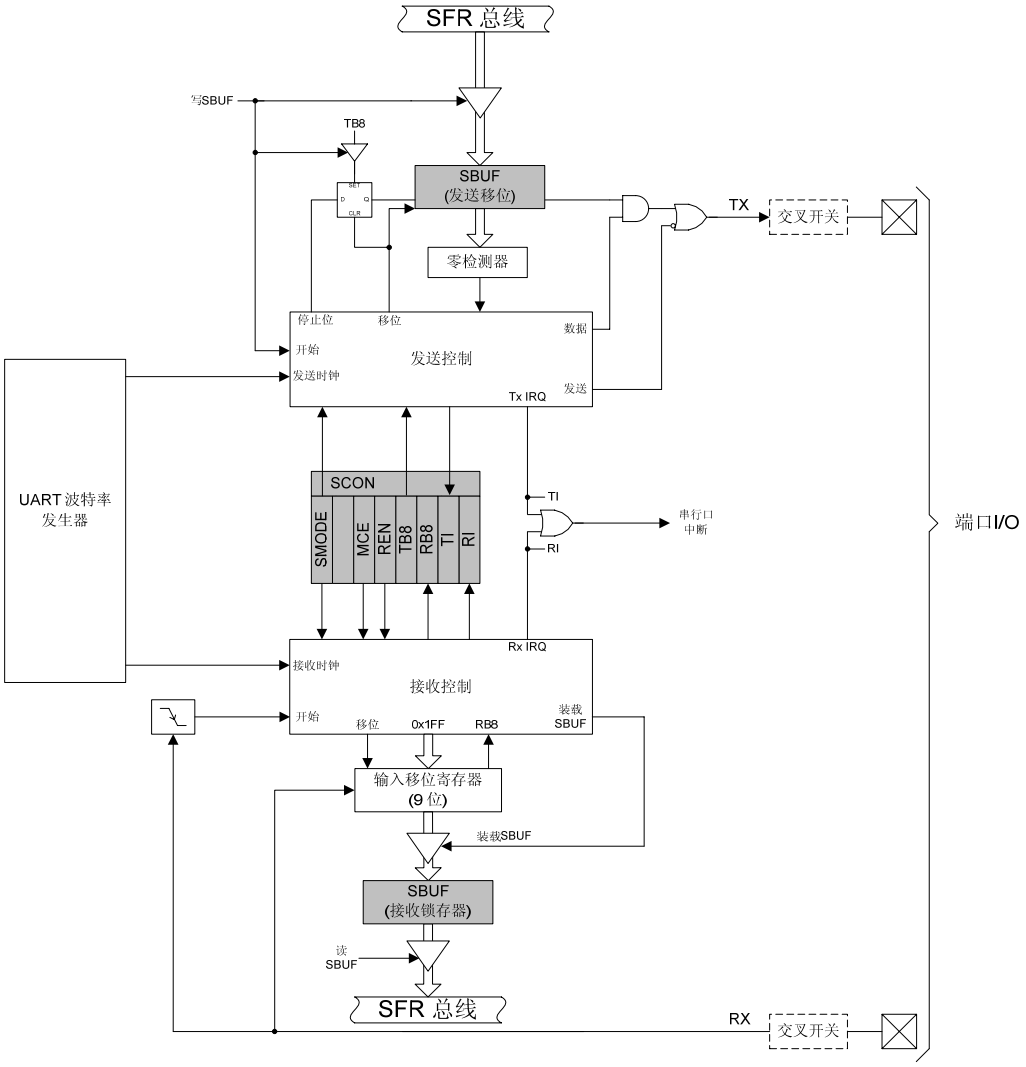


图 23.1　UART0 原理框图

23.1 增强的波特率发生器

UART0 波特率由定时器 1 工作在 8 位自动重装载方式产生。发送（TX）时钟由 TL1 产生；接收（RX）时钟由 TL1 的副本寄存器（图 23.2 中的 RX 定时器）产生，该寄存器不能被用户访问。TX 和 RX 定时器的溢出信号经过二分频后用于产生 TX 和 RX 波特率。当定时器 1 被允许时，RX 定时器运行并使用与定时器 1 相同的重载值（TH1）。在检测到 RX 引脚上的起始条件时 RX 定时器被强制重载，这允许在检测到起始位时立即开始接收过程，而与 TX 定时器的状态无关。

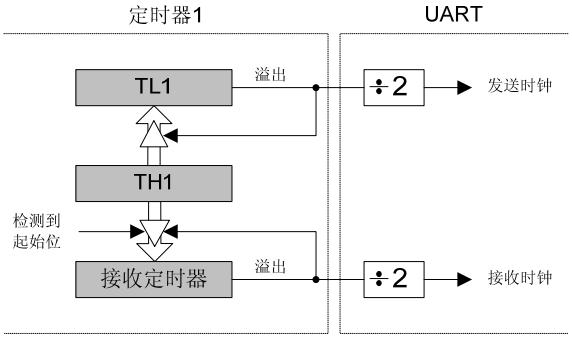


图 23.2　UART0 波特率逻辑

定时器 1 应被配置为方式 2，即 8 位自动重装载方式（见“25.1.3 方式 2：8 位重装载计数器/定时器”）。定时器 1 的重载值应设置为使其溢出频率为所期望的 UART 波特率频率的两倍。注意，定时器 1 的时钟可以在 6 个时钟源中选择：SYSCLK、SYSCLK/4、SYSCLK/12、SYSCLK/48、外部振荡器时钟/8 和外部输入 T1。对于任何给定的定时器 1 时钟源，UART0 的波特率由方程 23.1-A 和方程 23.1-B 决定：

A）$$UART\text{波特率} = \frac{1}{2} \times T1\text{溢出率}$$

B）$$T1\text{溢出率} = \frac{T1_{CLK}}{(256 - TH1)}$$

方程 23.1 UART0 波特率

其中 $T1_{CLK}$ 是定时器 1 的时钟频率，*TH1* 是定时器 1 的高字节（重载值）。

定时器 1 时钟频率的选择见“25.1 定时器 0 和定时器 1”。表 23.1 – 23.2 给出了典型波特率和系统时钟频率的对照表。注意，当外部振荡器驱动定时器 1 时，内部振荡器仍可产生系统时钟。

23.2 工作方式

UART0 提供标准的异步、全双工通信，其工作方式（8 位或 9 位）通过 S0MODE 位（SCON0.7）来选择。典型的 UART 连接方式如图 23.3 所示。

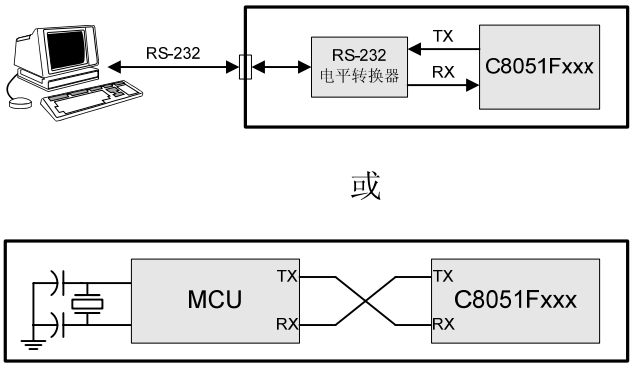


图 23.3　UART 连接图

23.2.1　8 位 UART

在 8 位 UART 方式，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（LSB 在先），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）。

当软件向 SBUF0 寄存器写入一个数据字节时开始数据发送。在发送结束时（停止位开始）发送中断标志 TI0（SCON0.1）被置 1。在接收允许位 REN0（SCON0.4）被置 1 后，数据接收可以在任何时刻开始。收到停止位后，如果满足下述条件则数据字节将被装入到接收寄存器 SBUF0：RI0 必须为逻辑 0；如果 MCE0 为逻辑 1，则停止位必须为 1。在发生接收数据溢出的情况下，先接收到的 8 位数据被锁存到 SBUF0，而后面的溢出数据被丢弃。

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 标志也不会被置 1。如果中断被允许，在 TI0 或 RI0 置位时将产生一个中断。

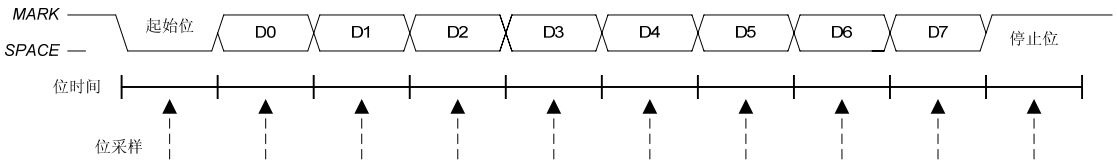


图 23.4　8 位 UART 时序图

23.2.2 9 位 UART

在 9 位 UART 方式，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。第九发送数据位由 TB80（SCON0.3）中的值决定，由用户软件赋值。它可以被赋值为 PSW 中的奇偶位 P（用于错误检测），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 寄存器写一个数据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 被置 1。在接收允许位 REN0（SCON0.4）被置 1 后，数据接收可以在任何时刻开始。收到停止位后如果满足下述条件则数据字节将被装入到接收寄存器 SBUF0：RI0 为逻辑 0；如果 MCE0 为逻辑 1，则第九位必须为逻辑 1（当 MCE0 为逻辑 0 时，第九位数据的状态并不重要）。如果这些条件满足，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 标志也不会被置 1。如果中断被允许，在 TI0 或 RI0 置位时将产生 UART0 中断。

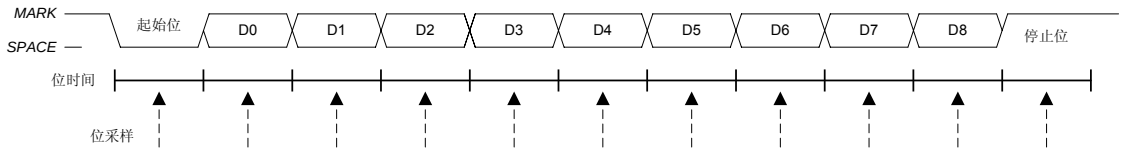


图 23.5 9 位 UART 时序图

23.3 多机通信

9 位 UART 方式通过使用第 9 数据位可以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。当主机要发送数据给一个或多个从机时，它先发送一个用于选择目标的地址字节。地址字节与数据字节的区别是：地址字节的第 9 位为逻辑 1；数据字节的第 9 位总是设置为逻辑 0。

如果从机的 MCE0 位（SCON.5）被置 1，则只有当 UART 接收到的第九位为逻辑 1（RB80 = 1）并收到有效的停止位后 UART 才会产生中断，意味着接收到一个地址字节。在 UART 的中断处理程序中，软件将接收到的地址与从机自身的 8 位地址进行比较。如果地址匹配，从机将清除它的 MCE0 位以允许后面接收数据字节时产生中断。未被寻址的从机仍保持其 MCE0 位为 1，在收到后续的数据字节时不产生中断，从而忽略收到的数据。一旦接收完整个消息，被寻址的从机将它的 MCE0 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。

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“广播”发送。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。

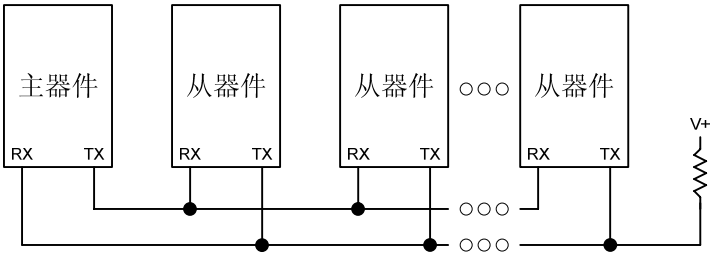


图 23.6　UART 多机方式连接图

SFR 定义 23.1　SCON0：　UART0 控制寄存器

SFR 页：0x00
SFR 地址：0x98　（可位寻址）

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	复位值
S0MODE	-	MCE0	REN0	TB80	RB80	TI0	RI0	01000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　S0MODE：UART0 工作方式位
该位选择 UART0 的工作方式。
0：波特率可编程的 8 位 UART。
1：波特率可编程的 9 位 UART。

位 6：　未使用。读 = 1b。写 = 忽略。

位 5：　MCE0：多处理器通信允许
该位的功能取决于 UART0 工作方式。
对方式 0（8 位 UART）：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 1 时 RI0 激活。
对方式 1（9 位 UART）：多处理器通信允许。
0：第 9 位的逻辑电平被忽略。
1：只有当第 9 位为逻辑 1 时 RI0 才被置位并产生中断。

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

位 3：　TB80：第 9 发送位
该位的逻辑电平被赋值给 9 位 UART 方式的第 9 发送位。在 8 位 UART 方式（方式 0）中未用。

位 2：　RB80：第 9 接收位
在方式 0，则 RB80 被赋值为停止位的值。在方式 1，该位被赋值为 9 位 UART 方式中第九数据位的值。

位 1：　TI0：发送中断标志
当 UART0 发送完一个字节数据后该位被硬件置 1（在 8 位 UART 方式时，是在发送第 8 位后；在 9 位 UART 方式时，是在停止位开始）。当 UART0 中断被允许时，置 1 该位将导致 CPU 转到 UART0 中断服务程序。该位必须用软件清 0。

位 0：　RI0：接收中断标志
当 UART0 接收到一个字节数据时该位被硬件置 1（在采样停止位时）。当 UART0 中断被允许时，置 1 该位将会使 CPU 转到 UART0 中断服务程序。该位必须用软件清 0。

SFR 定义 23.2　SBUF0：UART0 数据缓冲器

SFR 页：0x00
SFR 地址：0x99

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	

位 7-0：　SBUF0[7:0]：UART0 数据缓冲器位 7-0（MSB-LSB）
该 SFR 访问两个寄存器：发送移位寄存器和接收锁存寄存器。当数据被写到 SBUF0 时，它进入发送移位寄存器等待串行发送。向 SBUF0 写入一个字节即启动发送过程。读 SBUF0 时返回接收锁存器的内容。

表 23.1 对应标准波特率的定时器设置（使用内部 24.5MHz 振荡器）

	频率：24.5MHz						
	目标波特率（bps）	波特率误差（%）	振荡器分频系数	定时器时钟源	SCA1-SCA0（分频选择）[1]	T1M[1]	定时器 1 重载值（hex）
SYSCLK 源自内部振荡器	230400	-0.32%	106	SYSCLK	XX[2]	1	0xCB
	115200	-0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	-0.32%	848	SYSCLK/4	01	0	0x96
	14400	0.15%	1704	SYSCLK/12	00	0	0xB9
	9600	-0.32%	2544	SYSCLK/12	00	0	0x96
	2400	-0.32%	10176	SYSCLK/48	10	0	0x96
	1200	0.15%	20448	SYSCLK/48	10	0	0x2B

注：
1．SCA1-SCA0 和 T1M 位的定义见 25.1 节。
2．X=忽略。

表 23.2 对应标准波特率的定时器设置（使用 22.1184MHz 外部振荡器）

	频率：22.1184MHz						
	目标波特率（bps）	波特率误差（%）	振荡器分频系数	定时器时钟源	SCA1-SCA0（分频选择）[1]	T1M[1]	定时器 1 重载值（hex）
SYSCLK 源自外部振荡器	230400	0.00%	96	SYSCLK	XX[2]	1	0xD0
	115200	0.00%	192	SYSCLK	XX	1	0xA0
	57600	0.00%	384	SYSCLK	XX	1	0x40
	28800	0.00%	768	SYSCLK/12	00	0	0xE0
	14400	0.00%	1536	SYSCLK/12	00	0	0xC0
	9600	0.00%	2304	SYSCLK/12	00	0	0xA0
	2400	0.00%	9216	SYSCLK/48	10	0	0xA0
	1200	0.00%	18432	SYSCLK/48	10	0	0x40
SYSCLK 源自内部振荡器	230400	0.00%	96	EXTCLK/8	11	0	0xFA
	115200	0.00%	192	EXTCLK/8	11	0	0xF4
	57600	0.00%	384	EXTCLK/8	11	0	0xE8
	28800	0.00%	768	EXTCLK/8	11	0	0xD0
	14400	0.00%	1536	EXTCLK/8	11	0	0xA0
	9600	0.00%	2304	EXTCLK/8	11	0	0x70

注：

1. SCA1-SCA0 和 T1M 位的定义见 25.1 节。
2. X=忽略。

24. 增强型串行外设接口（SPI0 和 SPI1）

增强型串行外设接口（SPI0 和 SPI1）提供访问两个完全相同的、灵活的全双工同步串行总线的能力。SPI0 和 SPI1 统称为 SPIn。SPIn 可以作为主器件或从器件工作，可以使用 3 线或 4 线方式，并可在同一 SPI 总线上支持多个主器件和从器件。从选择信号（NSS）可被配置为输入以选择工作在从方式的 SPIn，或在多主环境中禁止主方式操作，以避免两个以上主器件试图同时进行数据传输时发生 SPI 总线冲突。NSS 可以被配置为片选输出（在主方式），或在 3 线操作时被禁止。在主方式，可以用其他通用端口 I/O 引脚选择多个从器件。

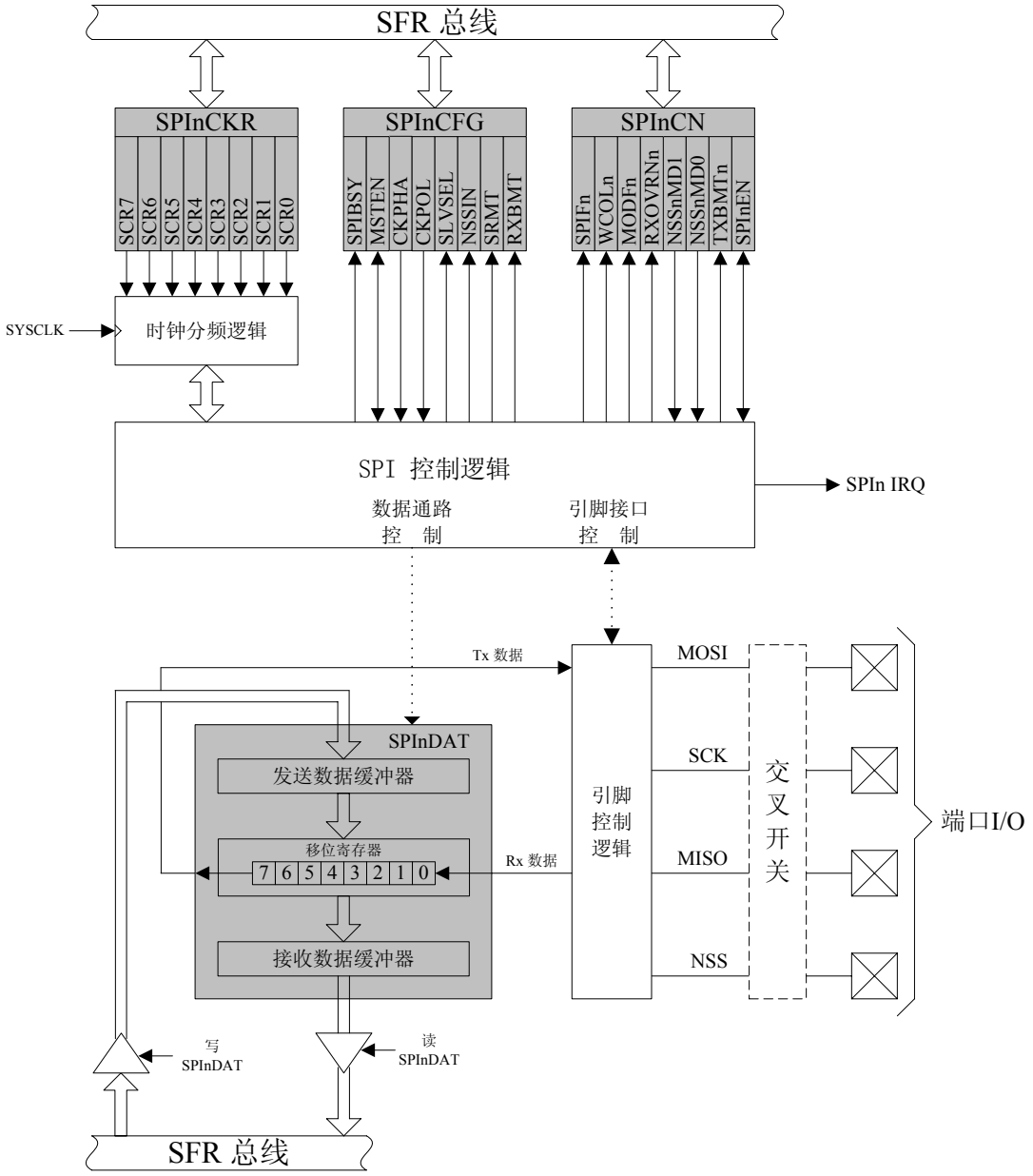


图 24.1　SPI 原理框图

24.1 信号说明

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

24.1.1 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 SPIn 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。当被配置为主器件时，MOSI 由移位寄存器的 MSB 驱动。

24.1.2 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 SPIn 作为主器件时，该信号是输入；当 SPIn 作为从器件时，该信号是输出。数据传输时最高位在先。当 SPI 被禁止或工作在 4 线从方式而未被选中时， MISO 引脚被置于高阻态。当作为从器件工作在 3 线方式时，MISO 总是由移位寄存器的 MSB 驱动。

24.1.3 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 SPIn 作为主器件时产生该信号。在 4 线从方式，当从器件未被选中时（NSS=1），SCK 信号被忽略。

24.1.4 从选择（NSS）

从选择（NSS）信号的功能取决于 SPInCN 寄存器中 NSSnMD1 和 NSSnMD0 位的设置。有 3 种可能的方式：

1. NSSnMD[1:0] = 00：3 线主方式或从方式：SPIn 工作在 3 线方式，NSS 被禁止。当作为从器件工作在 3 线方式时，SPIn 总是被选择。由于没有选择信号，SPIn 工作在 3 线方式时必须是总线唯一的从器件。这种情况用于一个主器件和一个从器件之间点对点通信。
2. NSSnMD[1:0] = 01：4 线从方式或多主方式：SPIn 工作在 4 线方式，NSS 被使能为输入。当作为从器件时，NSS 选择 SPIn 器件。当作为主器件时，NSS 信号的负跳变禁止 SPIn 的主器件功能，以便可以在同一个 SPI 总线上使用多个主器件。
3. NSSnMD[1:0] = 1x：4 线主方式：SPIn 工作在 4 线方式，NSS 被使能为输出。NSSnMD0 的设置值决定 NSS 引脚的输出逻辑电平。这种配置只能在 SPIn 作为主器件时使用。

图 24.2 ~ 24.4 给出了不同工作方式下的典型连接图。注意：NSSnMD 位的设置影响器件的引脚分配。当工作在 3 线主或从方式时，NSS 不被交叉开关分配引脚。在所有其他方式，NSS 必须被映射到器件引脚。有关通用端口 I/O 和交叉开关的信息见“21. 端口输入/输出”。

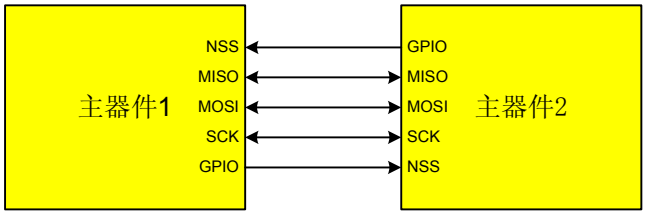


图 24.2　多主方式连接图

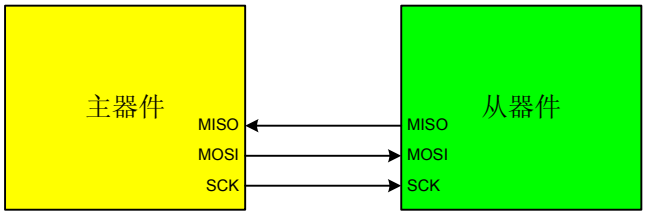


图 24.3　3 线单主方式和 3 线单从方式连接图

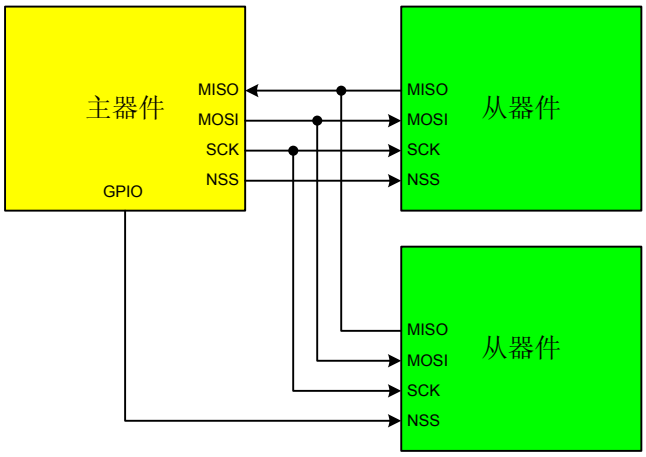


图 24.4　4 线单主方式和 4 线从方式连接图

24.2 SPI 主方式

SPI 总线上的所有数据传输都由 SPI 主器件启动。通过将主允许标志（MSTEN，SPInCFG.6）置 1 将 SPIn 置于主方式。当处于主方式时，向 SPIn 数据寄存器（SPInDAT）写入一个数据字节时是写发送缓冲器。如果 SPI 移位寄存器为空，发送缓冲器中的数据字节被传送到移位寄存器，数据传输开始。SPIn 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 SPIFn（SPInCN.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。在全双工操作中，当 SPI 主器件在 MOSI 线向从器件发送数据时，被寻址的 SPI 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。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。从从器件接收的数据字节以 MSB 在先的形式传送到主器件的移位寄存器。当一个数据字节被完全移入移位寄存器时，便被传送到接收缓冲器，处理器通过读 SPInDAT 来读该缓冲器。

当被配置为主器件时，SPIn 可以工作在下面的三种方式之一：多主方式、3 线单主方式或 4 线单主方式。当 NSSnMD1（SPInCN.3）=0 且 NSSnMD0（SPInCN.2）=1 时，是默认的多主方式。在该方式，NSS 是器件的输入，用于禁止主 SPIn，以允许另一主器件访问总线。在该方式，当 NSS 被拉为低电平时，MSTENn（SPInCN.6）和 SPIENn（SPInCN.0）位被清 0，以禁止 SPI 主器件，且方式错误标志（MODFn，SPInCN.5）被置 1。如果中断被允许，将产生方式错误中断。在这种情况下，必须用软件重新使能 SPIn。在多主系统中，当器件不作为系统主器件使用时，一般被默认为从器件。在多主方式，可以用通用 I/O 引脚对从器件单独寻址（如果需要）。图 24.2 给出了两个主器件在多主方式下的连接图。

当 NSSnMD1（SPI0CN.3）=0 且 NSSnMD0（SPInCN.2）=0 时，SPIn 工作在 3 线单主方式。在该方式，NSS 未被使用，也不被交叉开关映射到外部端口引脚。在该方式，应使用通用 I/O 引脚选择要寻址的任何从器件。图 24.3 给出了一个 3 线主方式主器件和一个从器件的连接图。

当 NSSnMD1（SPInCN.3）=1 时，SPIn 工作在 4 线单主方式。在该方式，NSS 被配置为输出引脚，可被用作从选择信号去选中一个 SPI 器件。在该方式，NSS 的输出值由 NSSnMD0（SPInCN.2）控制（用软件）。可以用通用 I/O 引脚寻址另外的从器件。图 24.4 给出了一个 4 线主方式主器件和两个从器件的连接图。

24.3　SPI 从方式

当 SPIn 被使能而未被配置为主器件时，它将作为 SPI 从器件工作。作为从器件，由主器件控制串行时钟（SCK），从 MOSI 移入数据，从 MISO 引脚移出数据。SPIn 逻辑中的位计数器对 SCK 边沿计数。当 8 位数据经过移位寄存器后，SPIF 标志被置为逻辑 1，接收到的字节被复制到接收缓冲器。通过读 SPInDAT 来读取接收缓冲器中的数据。从器件不能启动数据传送。通过写 SPInDAT 来预装要发送给主器件的数据到移位寄存器。写往 SPInDAT 的数据是双缓冲的，首先被放在发送缓冲器。如果移位寄存器为空，发送缓冲器中的数据会立即被传送到移位寄存器。当移位寄存器中已经有数据时，SPI 将在下一次（或当前）SPI 传输的最后一个 SCK 边沿过去后再将发送缓冲器的内容装入移位寄存器。

当被配置为从器件时，SPIn 可以工作 4 线或 3 线方式。当 NSSnMD1（SPInCN.3）=0 且 NSSnMD0（SPInCN.2）=1 时，是默认的 4 线从方式。在 4 线方式，NSS 被分配到一个端口引脚并被配置为数字输入。当 NSS 为逻辑 0 时，SPIn 被使能；当 NSS 为逻辑 1 时，SPIn 被禁止。在 NSS 的下降沿，位计数器被复位。注意，对应每次字节传输，在第一个有效 SCK 边沿到来之前，NSS 信号必须被驱动到低电平至少两个系统时钟周期。图 24.4 给出了两个 4 线方式从器件和一个主器件的连接图。

当 NSSnMD1（SPInCN.3）=0 且 NSSnMD0（SPInCN.2）=0 时，SPIn 工作在 3 线从方式。在该方式，NSS 未被使用，也不被交叉开关映射到外部端口引脚。由于在 3 线从方式无法唯一地寻址从器件，所以 SPIn 必须是总线上唯一的从器件。需要注意的是，在 3 线从方式，没有外部手段对位计数器复位以判断是否收到一个完整的字节。只能通过用 SPIEN 位禁止并重新使能 SPIn 来复位位计数器。图 24.3 给出了一个 3 线从器件和一个主器件的连接图。

24.4　SPI 中断源

如果 SPIn 中断被允许，在下述 4 个标志位被置 1 时将产生中断。

注意：这 4 个标志位都必须用软件清 0。

1. 在每次字节传输结束时，SPI 中断标志 SPIFn（SPInCN.7）被置 1。该标志适用于所有 SPI 方式。
2. 如果在发送缓冲器中的数据尚未被传送到 SPI 移位寄存器时写 SPInDAT，写冲突标志 WCOL（SPInCN.6）被置 1。发生这种情况时，写 SPInDAT 的操作被忽略，不会对发送缓冲器写入。该标志适用于所有 SPI 方式。
3. 当 SPIn 被配置为工作于多主方式的主器件而 NSS 被拉为低电平时，方式错误标志 MODFn（SPInCN.5）被置 1。当发生方式错误时，SPInCN 中的 MSTEN 和 SPInEN 位被清 0，以禁止 SPIn 并允许另一个主器件访问总线。
4. 当 SPIn 被配置为从器件并且一次传输结束，而接收缓冲器中还保持着上一次传输的数据未被读取时，接收溢出标志 RXOVRNn（SPInCN.4）被置 1。新接收的字节将不被传送到接收缓冲器，允许前面接收的字节被读取。引起溢出的数据字节丢失。

24.5 串行时钟相位和极性

使用 SPIn 配置寄存器（SPInCFG）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（SPInCFG.5）选择两种时钟相位（锁存数据所用的边沿）中的一种。CKPOL 位（SPInCFG.4）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。在改变时钟相位和极性期间应禁止 SPIn（通过清除 SPInEN 位，SPInCN.0）。主方式下时钟和数据线的时序关系示于图 24.5。从方式下时钟和数据的时序关系示于图 24.6 和图 24.7。注意：当 C8051F04x, C8051F06x, C8051F12x, C8051F31x, C8051F32x, C8051F33x 中的两个器件通信时，不论工作在主方式还是从方式，CKPHA 必须被置 0。

SFR 定义 24.3 所示的 SPIn 时钟速率寄存器（SPInCKR）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。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或 12.5MHz（取较低者）。当 SPI 被配置为从器件时，全双工操作的最大数据传输率（位/秒）是系统时钟频率的十分之一，前提是主器件与从器件的系统时钟同步发出 SCK、NSS（在 4 线从方式）和串行输入数据。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发出 SCK、NSS 和串行输入数据与从器件的系统时钟同步的情况下。

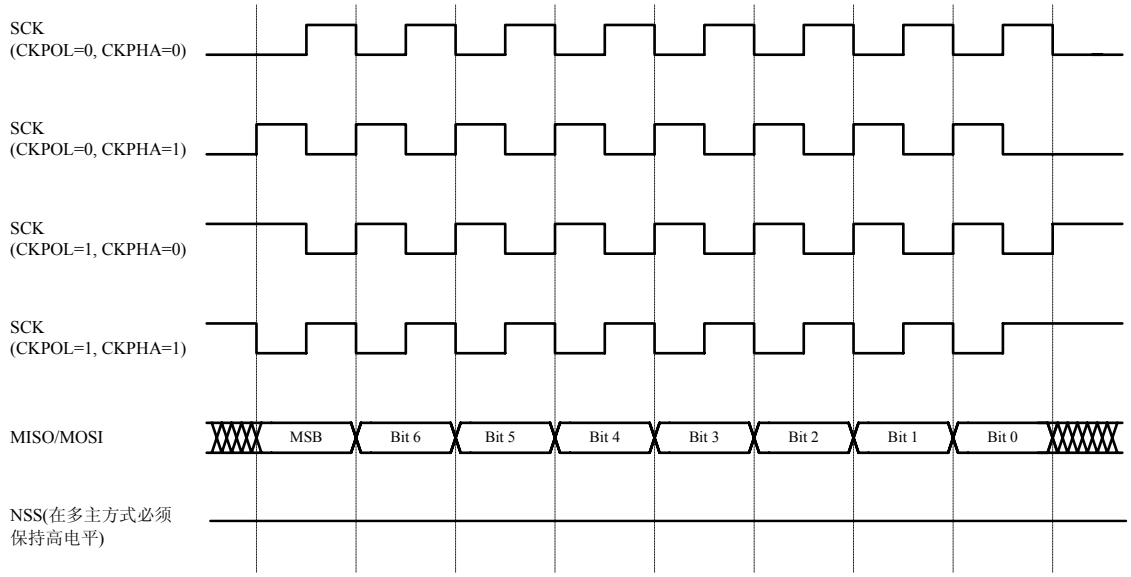


图 24.5　主方式数据/时钟时序图

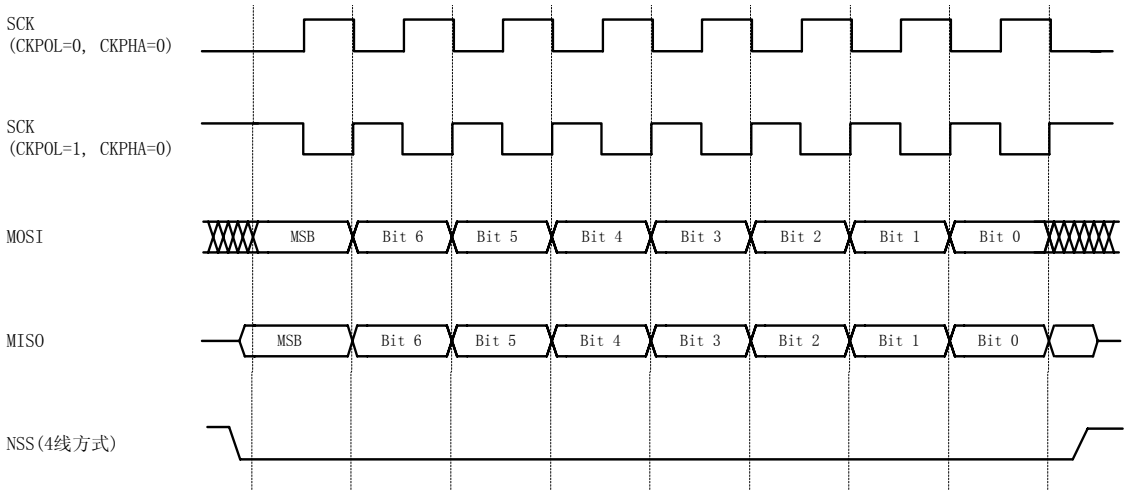


图 24.6　从方式数据/时钟时序图（CKPHA = 0）

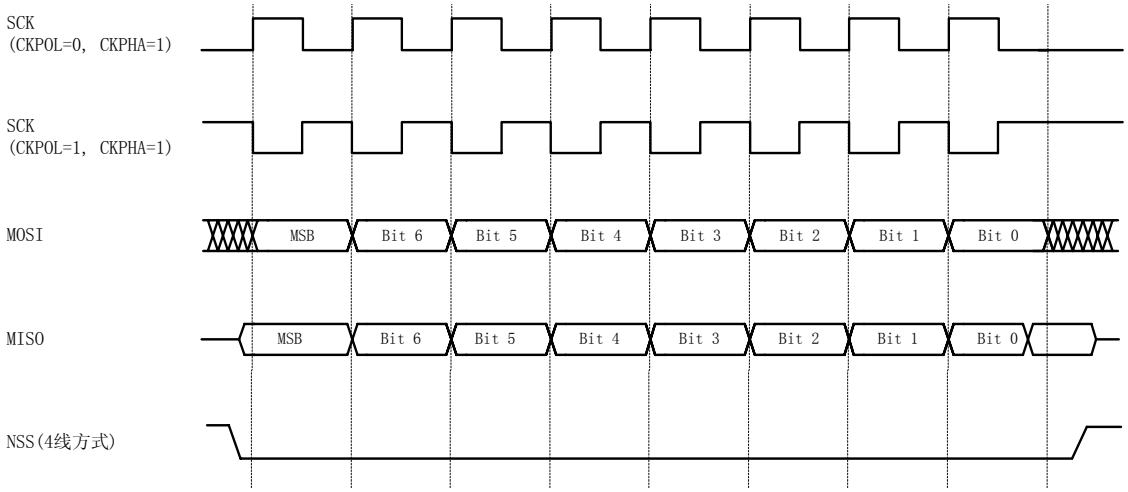


图 24.7　从方式数据/时钟时序图（CKPHA = 1）

24.6 SPI 特殊功能寄存器

对 SPI0 和 SPI1 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(共 8 个寄存器)实现的：控制寄存器 SPInCN、数据寄存器 SPInDAT、配置寄存器 SPInCFG 和时钟频率寄存器 SPInCKR。下面的 SFR 定义将介绍这些与 SPI0 和 SPI1 总线操作有关的特殊功能寄存器。

SFR 定义 24.1 SPInCFG：SPIn 配置寄存器

SFR 页：0x00
SFR 地址：SPI0CFG = 0xA1，SPI1CFG = 0x84

R	R/W	R/W	R/W	R	R	R	R	复位值
SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT	00000111
位7	位6	位5	位4	位3	位2	位1	位0	

位 7： SPIBSY：SPI 忙标志（只读）
当一次 SPI 传输正在进行时（主或从方式），该位被置为逻辑 1。

位 6： MSTEN：主方式使能位
0：禁止主方式，工作在从方式。
1：使能主方式，工作在主器件方式。

位 5： CKPHA：SPI 时钟相位。
该位控制 SPI 时钟的相位。
0：数据以 SCK 周期的第一个边沿为中心*。
1：数据以 SCK 周期的第二个边沿为中心*。

位 4： CKPOL：SPI 时钟极性
该位控制 SPI 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 3： SLVSEL：从选择标志（只读）。
当 NSS 引脚为低电平时该位被置 1，指示 SPIn 是被选中的从器件。当 NSS 引脚为高电平时（未被选中为从器件）该位被清 0。该位不指示 NSS 引脚的即时值，而是该引脚输入的去噪版本。

位 2： NSSIN：NSS 引脚的即时输入值（只读）
该位指示读该寄存器时 NSS 引脚的即时值。该信号未被去噪。

位 1： SRMT：移位寄存器空标志（只在从方式有效，(只读)）。
当所有数据都被移入/移出移位寄存器并且没有新数据可以从发送缓冲器读出或向接收缓冲器写入时，该位被置 1。当数据字节被从发送缓冲器传送到移位寄存器或 SCK 发生变化时，该位被清 0。
注：在主方式时 SRMT = 1。

位 0： RXBMT：接收缓冲器空（在从方式有效，只读）
当接收缓冲器被读取且没有新数据时，该位被置 1。如果在接收缓冲器中有新数据未被读取，则该位被清 0。注意：在主方式时，RXBMT = 1。

*在从方式，MOSI 上的数据是在每个数据位的中间被采样。在主方式，MISO 上的数据是在每个数据结束前一个 SYSCLK 位被采样，为从器件提供最大的建立时间。时序参数见表 24.1。

SFR 定义 24.2　SPInCN：SPIn 控制寄存器

SFR 页：0x00
SFR 地址：SPI0CN = 0xF8，SPI1CN = 0xB0　(可位寻址)

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	复位值
SPIFn	WCOLn	MODFn	RXOVRNn	NSSnMD1	NSSnMD0	TXBMTn	SPInEN	00000110
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　SPIFn：SPIn 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，该位置 1 将会使 CPU 转到 SPIn 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。

位 6：　WCOLn：写冲突标志
该位由硬件置为逻辑 1（并产生一个 SPIn 中断），指示数据传送期间对 SPIn 数据寄存器进行了写操作。该位不能被硬件自动清 0，必须用软件清 0。

位 5：　MODFn：方式错误标志
当检测到主方式冲突（NSS 为低电平，MSTEN＝1，NSSnMD[1:0] = 01）时，该位由硬件置为逻辑 1（并产生一个 SPIn 中断）。该位不能被硬件自动清 0，必须用软件清 0。

位 4：　RXOVRNn：接收溢出标志（只适用于从方式）
当前传输的最后一位已经移入 SPI 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 SPIn 中断）。该位不会被硬件自动清 0，必须用软件清 0。

位 3-2：　NSSnMD1-NSSnMD0：从选择方式位
选择 NSS 工作方式：
00：3 线从方式或 3 线主方式。NSS 信号不连到端口引脚。
01：4 线从方式或多主方式（默认值）。NSS 是器件的输入。
1x：4 线单主方式。NSS 信号被分配一个输出引脚并输出 NSSMD0 的值。

位 1：　TXBMTn：发送缓冲器空标志
当新数据被写入发送缓冲器时，该位被清 0。当发送缓冲器中的数据被传送到 SPI 移位寄存器时，该位被置 1，表示可以安全地向发送缓冲器写新字节。

位 0：　SPInEN：SPIn 使能位
该位使能 / 禁止 SPIn。
0：禁止 SPIn
1：使能 SPIn

SFR 定义 24.3　SPInCKR：SPIn 时钟速率寄存器

SFR 页：0x00
SFR 地址：SPI0CKR = 0xA2，SPI1CKR = 0x85　(可位寻址)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SCRn7	SCRn6	SCRn5	SCRn4	SCRn3	SCRn2	SCRn1	SCRn0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　SCRn[7:0]：SPIn 时钟频率
当 SPIn 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，*SPInCKR* 是 SPInCKR 寄存器中的 8 位值。

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

（0 ≤SPInCKR≤255）

例如：如果 SYSCLK = 2MHz，SPInCKR = 0x04，则

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

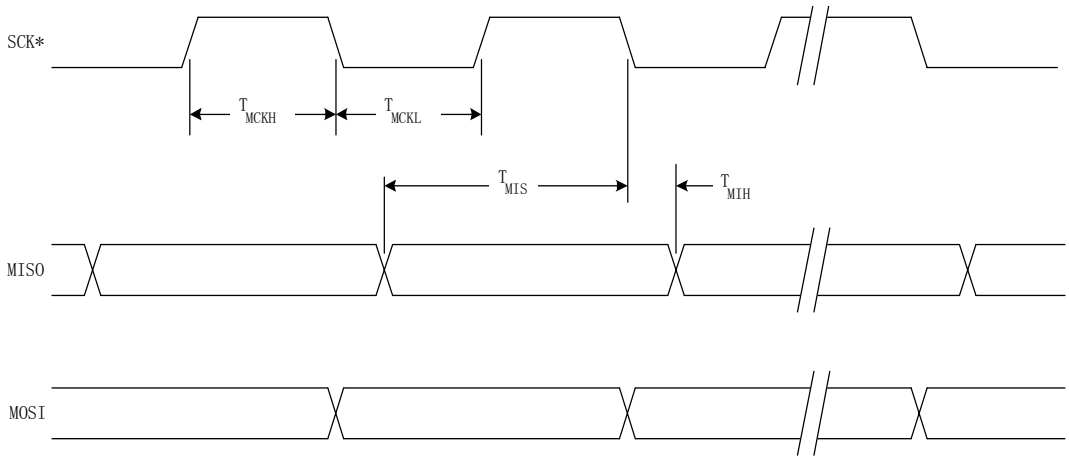
$$f_{SCK} = 200kHz$$

SFR 定义 24.4　SPInDAT：SPIn 数据寄存器

SFR 页：0x00
SFR 地址：SPI0DAT = 0xA3，SPI1DAT = 0x86

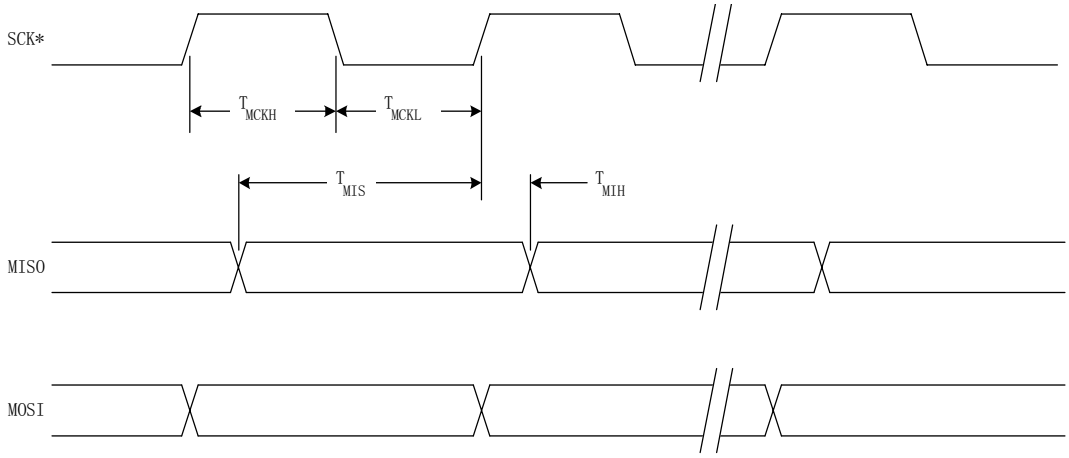
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-0：　SPInDAT：SPIn 发送和接收数据寄存器。
SPInDAT 寄存器用于发送和接收 SPIn 数据。在主方式下，向 SPInDAT 写入数据时，数据被放到发送缓冲器并启动发送。读 SPInDAT 返回接收缓冲器的内容。



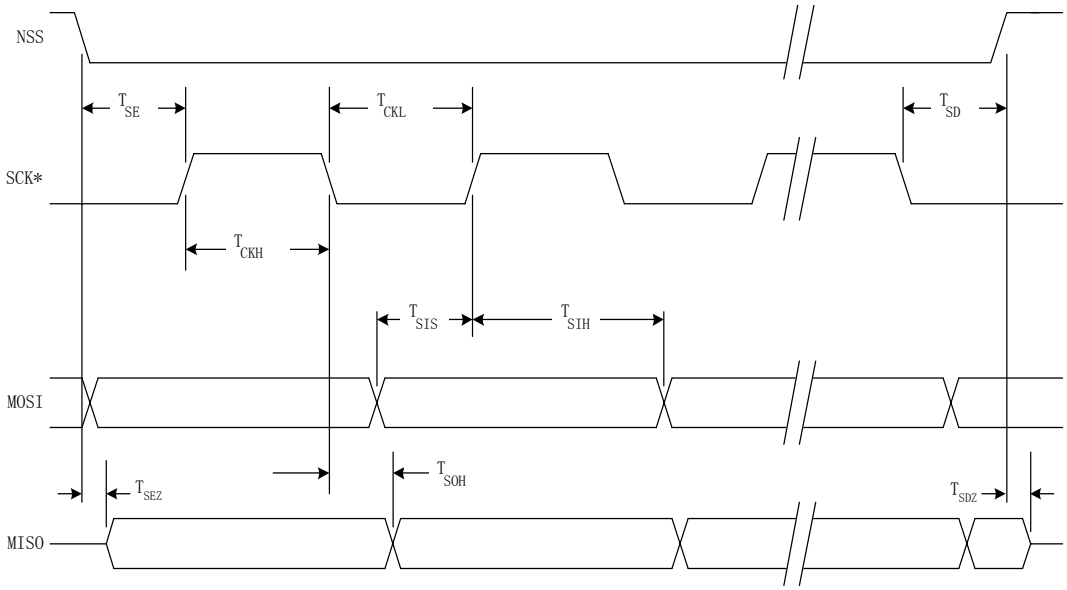
* 这是对应 CKPOL = 0时的 SCK 波形 对于 CKPOL = 1，SCK波形的极性反向。

图 24.8　SPI 主方式时序（CKPHA = 0）



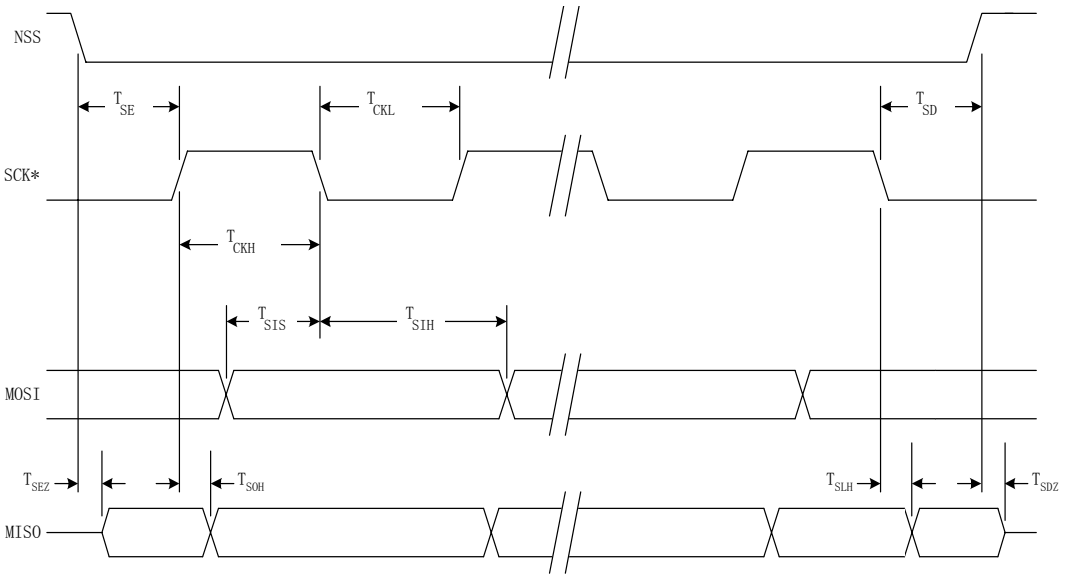
* 这是对应 CKPOL = 0时的 SCK 波形。对于 CKPOL = 1，SCK波形的极性反向。

图 24.9　SPI 主方式时序（CKPHA = 1）



* 这是对应 CKPOL = 0时的 SCK 波形。对于 CKPOL = 1，SCK波形的极性反向。

图 24.10　SPI 从方式时序（CKPHA = 0）



* 这是对应 CKPOL = 0时的 SCK 波形。对于 CKPOL = 1，SCK波形的极性反向。

图 24.11　SPI 从方式时序（CKPHA = 1）

表 24.1　SPI 从方式时序参数

参　数	说　明	最小值	最大值	单位
主方式时序*（见图 24.8 和图 24.9）				
T_{MCKH}	SCK 高电平时间	$1\times T_{SYSCLK}$	—	ns
T_{MCKL}	SCK 低电平时间	$1\times T_{SYSCLK}$	—	ns
T_{MIS}	MISO 有效到 SCK 移位边沿	$1\times T_{SYSCLK}+20$	—	ns
T_{MIH}	SCK 移位边沿到 MISO 发生改变	0	—	ns
从方式时序*（见图 24.10 和图 24.11）				
T_{SE}	NSS 下降沿到第一个 SCK 边沿	$2\times T_{SYSCLK}$	—	ns
T_{SD}	最后一个 SCK 边沿到 NSS 上升沿	$2\times T_{SYSCLK}$	—	ns
T_{SEZ}	NSS 下降沿到 MISO 有效	—	$4\times T_{SYSCLK}$	ns
T_{SDZ}	NSS 上升沿到 MISO 变为高阻态	—	$4\times T_{SYSCLK}$	ns
T_{CKH}	SCK 高电平时间	$5\times T_{SYSCLK}$	—	ns
T_{CKL}	SCK 低电平时间	$5\times T_{SYSCLK}$	—	ns
T_{SIS}	MOSI 有效到 SCK 采样边沿	$2\times T_{SYSCLK}$	—	ns
T_{SIH}	SCK 采样边沿到 MOSI 发生改变	$2\times T_{SYSCLK}$	—	ns
T_{SOH}	SCK 移位边沿到 MISO 发生改变	—	$4\times T_{SYSCLK}$	ns
T_{SLH}	最后一个 SCK 边沿到 MISO 发生改变（只限于 CKPHA = 1）	$6\times T_{SYSCLK}$	$8\times T_{SYSCLK}$	ns
*注：T_{SYSCLK} 为器件系统时钟（SYSCLK）的一个周期。				

25. 定时器

C8051F93x-C8051F92x 内部有 4 个计数器/定时器：其中两个 16 位计数器/定时器与标准 8051 中的计数器/定时器兼容，另外两个是 16 位自动重装载定时器，可用于 ADC、SMBus 或作为通用定时器使用。这些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 和定时器 3 均可作为一个 16 位或两个 8 位自动重装载定时器。另外，定时器 2 和定时器 3 还具有捕捉方式，可用于测量 smaRTClock 或比较器周期（相对于另一振荡器）。该功能在使用触摸开关时非常有用。

<table>
<tr><th>定时器 0 和定时器 1 工作方式</th><th>定时器 2 工作方式</th><th>定时器 3 工作方式</th></tr>
<tr><td>13 位计数器/定时器</td><td rowspan="2">16 位自动重装载定时器</td><td rowspan="2">16 位自动重装载定时器</td></tr>
<tr><td>16 位计数器/定时器</td></tr>
<tr><td>8 位自动重装载的计数器/定时器</td><td rowspan="2">两个 8 位自动重装载定时器</td><td rowspan="2">两个 8 位自动重装载定时器</td></tr>
<tr><td>两个 8 位计数器/定时器
（仅限于定时器 0）</td></tr>
</table>

定时器 0 和定时器 1 有 5 个可选择的时钟源，由定时器方式选择位（T1M-T0M）和时钟分频位（SCA1-SCA0）决定。时钟分频位定义一个预分频时钟，作为定时器 0 和/或定时器 1 的时钟源（见 SFR 定义 25.1）。

定时器 0 和定时器 1 可以被配置为使用预分频时钟或系统时钟。定时器 2 和定时器 3 可以使用系统时钟或系统时钟/12。定时器 2 还可以用 smaRTClock/8 或比较器 0 输出作为时钟源。定时器 3 还可以用外部振荡器时钟源/8 或比较器 1 输出作为时钟源。

定时器 0 和定时器 1 也可以工作在计数器方式。当作为计数器使用时，在为定时器所选择的输入引脚（T0 或 T1）上出现负跳变时计数器/定时器寄存器的值加 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正确采样。

SFR 定义 25.1　CKCON：时钟控制寄存器

SFR 页：0x00
SFR 地址：0x8E

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
T3MH	T3ML	T2MH	T2ML	T1M	T0M	SCA1	SCA0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　T3MH：定时器 3 高字节时钟选择。
该位选择供给定时器 3 高字节的时钟(如果定时器 3 被配置为两个 8 位定时器，定时器 3 工作在其他方式时该位被忽略)。
0：定时器 3 高字节使用 TMR3CN 中的 T3XCLK 位定义的时钟。
1：定时器 3 高字节使用系统时钟。

位 6：　T3ML：定时器 3 低字节时钟选择。
该位选择供给定时器 3 的时钟。如果定时器 3 被配置为两个 8 位定时器，该位选择供给低 8 位定时器的时钟。
0：定时器 3 低字节使用 TMR3CN 中的 T3XCLK 位定义的时钟。
1：定时器 3 低字节使用系统时钟。

位 5：　T2MH：定时器 2 高字节时钟选择。
该位选择供给定时器 2 高字节的时钟(如果定时器 2 被配置为两个 8 位定时器，定时器 2 工作在其他方式时该位被忽略)。
0：定时器 2 高字节使用 TMR2CN 中的 T2XCLK 位定义的时钟。
1：定时器 2 高字节使用系统时钟。

位 4：　T2ML：定时器 2 低字节时钟选择。
该位选择供给定时器 2 的时钟。如果定时器 2 被配置为两个 8 位定时器，该位选择供给低 8 位定时器的时钟。
0：定时器 2 低字节使用 TMR2CN 中的 T2XCLK 位定义的时钟。
1：定时器 2 低字节使用系统时钟。

位 3：　T1M：定时器 1 时钟选择。
该位选择定时器 1 的时钟源。当 C/T1 被设置为逻辑 1 时，T1M 被忽略。
0：定时器 1 使用由分频位（SCA1[1:0]）定义的时钟。
1：定时器 1 使用系统时钟。

位 2：　T0M：定时器 0 时钟选择。
该位选择定时器 0 的时钟源。当 C/T0 被设置为逻辑 1 时，T0M 被忽略。
0：定时器 0 使用由分频位（SCA1-SAC0）定义的时钟。
1：定时器 0 使用系统时钟。

位 1-0：　SCA1[1:0]：定时器 0/1 预分频位
如果定时器 0/1 被配置为使用预分频时钟，则这两位控制时钟分频数。

SCA1	SCA0	预分频时钟
0	0	系统时钟/12
0	1	系统时钟/4
1	0	系统时钟/48
1	1	外部时钟/8（与系统时钟同步）

25.1 定时器 0 和定时器 1

每个计数器/定时器都是一个 16 位的寄存器，在被访问时分为两个字节：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）用于使能定时器 0 和定时器 1 以及指示它们的状态。通过将 IE 寄存器中的 ET0 位置 1 来允许定时器 0 中断，通过将 ET1 位置 1 来允许定时器 1 中断。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 T1M1-T0M0 来选择工作方式，每个定时器都可以被独立配置。下面对每种工作方式进行详细说明。

25.1.1 方式 0 — 13 位计数器/定时器

在方式 0，定时器 0 和定时器 1 被作为 13 位的计数器/定时器使用。图 25.1 给出了定时器 0 工作在方式 0 时的原理框图。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，所以定时器 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（如果定时器 0 中断被允许）。

C/T0 位（TMOD.2）选择计数器/定时器的时钟源。当 C/T0 被设置为逻辑 1 时，出现在所选定时器 0 输入引脚（T0）上的负跳变使定时器寄存器加 1（有关选择和配置外部 I/O 引脚的详细信息见“21.3 优先权交叉开关译码器”）。清除 C/T0 位将选择由 T0M 位（CKCON.3）定义的时钟作为定时器的输入。当 T0M 被置 1 时，定时器 0 的时钟为系统时钟；当 T0M 位被清 0 时，定时器 0 的时钟源由 CKCON（见 SFR 定义 25.1）中的时钟分频位定义。

当 GATE0（TMOD.3）为逻辑 0 或输入信号/INT0 有效时（有效电平由 IT01CF 寄存器中的 IN0PL 位定义，见 SFR 定义 12.7），将 TR0 位（TCON.4）置 1 允许定时器 0 工作。设置 GATE0 为逻辑 1 允许定时器受外部输入信号/INT0 的控制，便于脉冲宽度测量。

表 25.1 定时器 0 工作方式

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
注：X=忽略。			

注意，置位 TR0 并不强制定时器 0 复位。应在定时器被使能前将定时器寄存器装入所期望的初值。

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 中的对应位。输入信号/INT1 为定时器 1 所用，其极性由 IT01CF 寄存器中的 IN1PL 位定义（见 SFR 定义 12.7）。

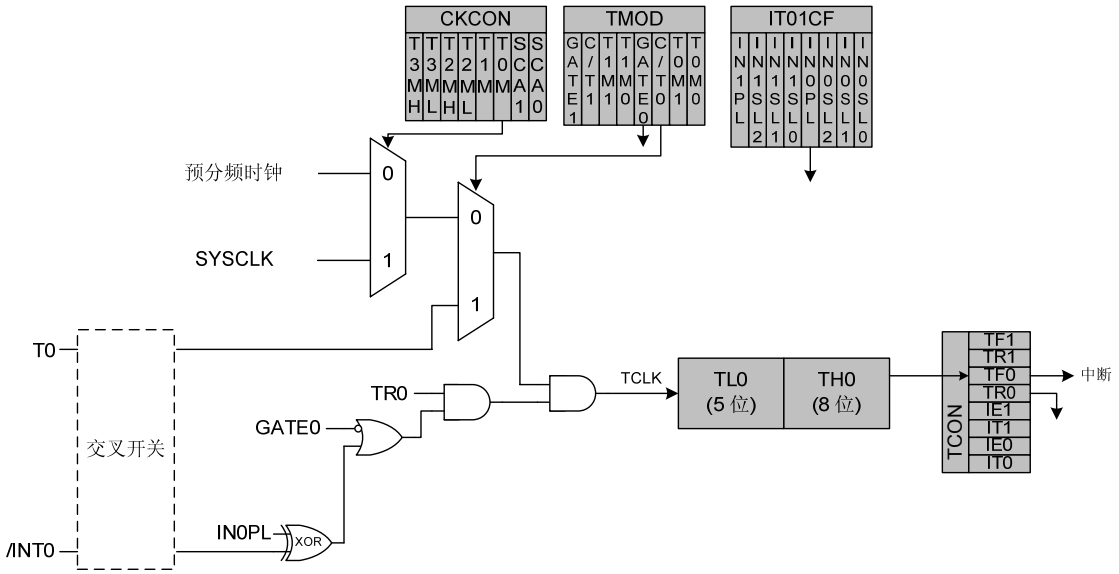


图 25.1　T0 方式 0 原理框图

25.1.2 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 相同的方法使能和配置工作在方式 1 的计数器/定时器。

25.1.3 方式 2 — 自动重装载的 8 位计数器/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。如果定时器 0 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 1 和定时器 0 的使能和配置方法与方式 0 相同。当 GATE0（TMOD.3）为逻辑 0 或输入信号/INT0 有效时（有效电平由 IT01CF 寄存器中的 IN0PL 为定义，有关外部输入信号/INT0 和/INT1 的详细说明见“12.6 外部中断 INT0 和 INT1”），置位 TR0（TCON.4）将允许定时器 0 工作。

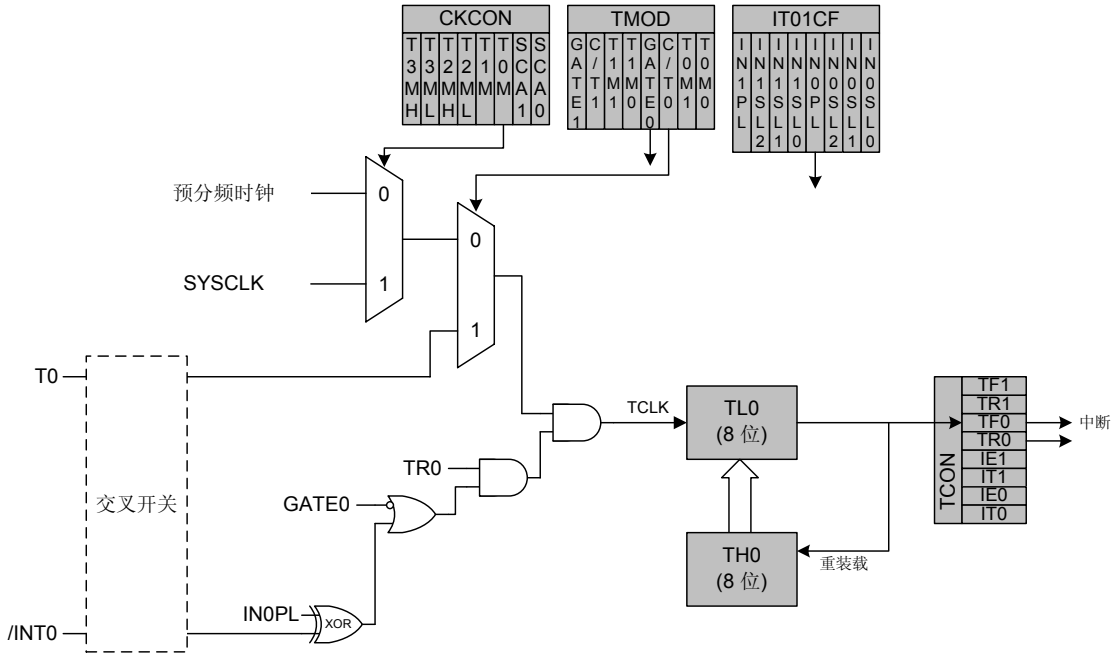


图 25.2　T0 方式 2 原理框图

25.1.4 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 被配置为两个独立的 8 位定时器/计数器，计数值分别在 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。TL0 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基。TH0 寄存器只能作为定时器使用，由系统时钟或预分频时钟提供时基。TH0 使用定时器 1 的运行控制位 TR1，并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。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 溢出可以用于为 SMBus 和/或 UART 产生波特率，也可以用于启动 ADC 转换。当定时器 0 工作在方式 3 时，定时器 1 的运行控制由其方式设置决定。为了在定时器 0 工作于方式 3 时使用定时器 1，应将定时器 1 的工作方式设置为 0、1 或 2。如果要禁止定时器，可以将其配置为方式 3。

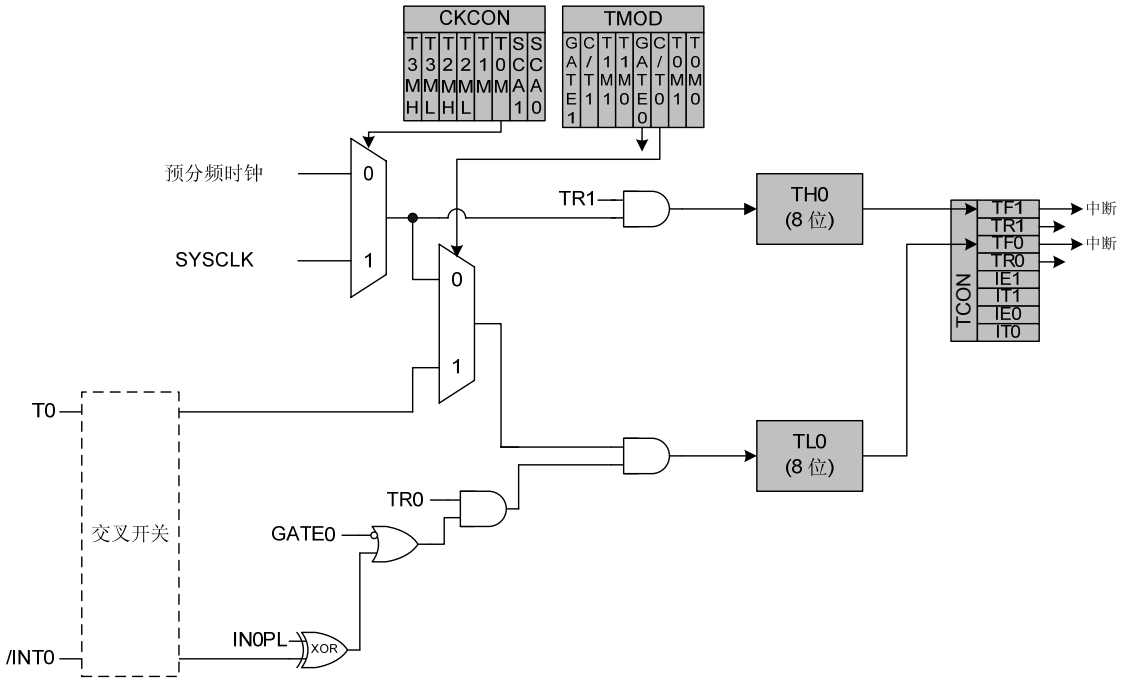


图 25.3　T0 方式 3 原理框图

SFR 定义 25.2　TCON：定时器控制寄存器

SFR 页：0x00
SFR 地址：0x88　　（可位寻址）

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	

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。

位 6：　TR1：定时器 1 运行控制
通过将该位置 1 来使能定时器 1。

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。

位 4：　TR0：定时器 0 运行控制
通过将该位置 1 来使能定时器 0。

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 的中断服务程序时该位被自动清 0（如果 IT1=1）。当 IT1=0 时，该标志在/INT1 有效时被置 1（有效电平由 IT01CF 寄存器中的 IN1PL 位定义）。

位 2：　IT1：中断 1 类型选择
该位选择/INT1 中断是边沿触发还是电平触发。可以用 IT01CF 寄存器中的 IN1PL 位将/INT1 配置为低电平有效或高电平有效（见 SFR 定义 12.7）。
0：/INT1 为电平触发。
1：/INT1 为边沿触发。

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 的中断服务程序时该位被自动清 0（如果 IT0=1）。当 IT0=0 时，该标志在/INT0 有效时被置 1（有效电平由 IT01CF 寄存器中的 IN0PL 位定义）。

位 0：　IT0：中断 0 类型选择
该位选择/INT0 中断是边沿触发还是电平触发。可以用 IT01CF 寄存器中的 IN0PL 位将/INT0 配置为低电平有效或高电平有效（见 SFR 定义 12.7）。
0：/INT0 为电平触发。
1：/INT0 为边沿触发。

SFR 定义 25.3　TMOD：定时器方式寄存器

SFR 页：0x00
SFR 地址：0x89

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	

位 7：　GATE1：定时器 1 门控位
0：当 TR1=1 时定时器 1 工作，与/INT1 的逻辑电平无关。
1：只有当 TR1=1 并且/INT1 有效时定时器 1 才工作（有效电平由 IT01CF 寄存器中的 IN1PL 位定义，见 SFR 定义 12.7）。

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。

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。

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

位 3：　GATE0：定时器 0 门控位
0：当 TR0=1 时定时器 0 工作，与/INT0 的逻辑电平无关。
1：只有当 TR0=1 并且/INT0 有效时定时器 0 才工作（有效电平由 IT01CF 寄存器中的 IN0PL 位定义，见 SFR 定义 12.7）。

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。

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。

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 3：两个 8 位计数器/定时器

SFR 定义 25.4　TL0：定时器 0 低字节

SFR 页：0x00
SFR 地址：0x8A

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	

位 7-0：　TL0[7:0]：定时器 0 低字节
TL0 寄存器是 16 位定时器 0 的低字节。

SFR 定义 25.5　TL1：定时器 1 低字节

SFR 页：0x00
SFR 地址：0x8B

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	B

位 7-0：　TL1[7:0]：定时器 1 低字节
TL1 寄存器是 16 位定时器 1 的低字节。

SFR 定义 25.6　TH0：定时器 0 高字节

SFR 页：0x00
SFR 地址：0x8C

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	

位 7-0：　TH0[7:0]：定时器 0 高字节
TH0 寄存器是 16 位定时器 0 的高字节。

SFR 定义 25.7　TH1：定时器 1 高字节

SFR 页：0x00
SFR 地址：0x8D

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	

位 7-0：　TH1[7:0]：定时器 1 高字节
TH1 寄存器是 16 位定时器 1 的高字节。

25.2 定时器 2

定时器 2 是一个 16 位的定时器，由两个 8 位的 SFR 组成：TMR2L（低字节）和 TMR2H（高字节）。定时器 2 可以工作在 16 位自动重装载方式或 8 位自动重装载方式（两个 8 位定时器）。T2SPLIT 位（TMR2CN.3）定义定时器 2 的工作方式。定时器 2 还可被用于捕捉方式，以测量 smaRTClock 或比较器 0 的周期（相对于另一振荡器）。测量比较器 0 周期（相对于系统时钟）的能力使触摸开关的实现非常容易。

定时器 2 的时钟源可以是系统时钟、系统时钟/12、smaRTClock/8 或比较器 0 输出。注意，smaRTClock/8 和比较器 0 输出与系统时钟同步。

25.2.1　16 位自动重装载定时器

当 T2SPLIT 位（TMR2CN.3）为 0 时，定时器 2 工作在自动重装载的 16 位定时器方式（见图 25.4）。定时器 2 可以使用 SYSCLK、SYSCLK/12、smaRTClock /8 或比较器 0 输出作为其时钟源。当 16 位定时器寄存器加 1 后发生溢出（从 0xFFFF 到 0x0000）时，定时器 2 重载寄存器（TMR2RLH 和 TMR2RLL）中的 16 位计数初值被自动装入到定时器 2 寄存器，定时器 2 高字节溢出标志 TF2H（TMR2CN.7）被置 1。如果定时器 2 中断被允许（如果 IE.5 被置 1），则每次溢出都将产生中断。如果定时器 2 中断被允许并且 TF2LEN 位（TMR2CN.5）被置 1，则每次低 8 位（TMR2L）溢出时（从 0xFF 到 0x00）将产生一个中断。

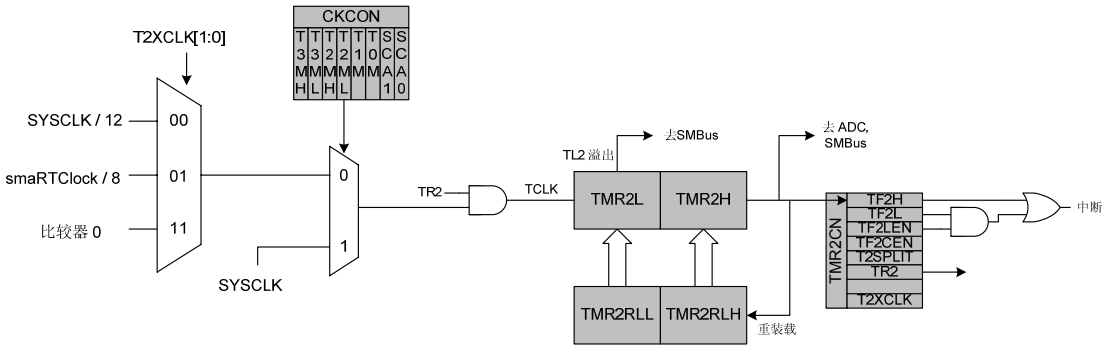


图 25.4　定时器 2 的 16 位方式原理框图

25.2.2 8 位自动重装载定时器

当 T2SPLIT 位被置 1 时，定时器 2 工作在双 8 位定时器方式（TMR2H 和 TMR2L）。这两个 8 位定时器都工作在自动重装载方式，如图 25.5 所示。TMR2RLL 保持 TMR2L 的重载值，而 TMR2RLH 保持 TMR2H 的重载值。TMR2CN 中的 TR2 是 TMR2H 的运行控制位。当定时器 2 被配置为 8 位方式时，TMR2L 总是处于运行状态。

每个 8 位定时器都可以被配置为使用 SYSCLK、SYSCLK/12、smaRTClock /8 或比较器 0 输出作为时钟源。定时器 2 时钟选择位 T2MH 和 T2ML（位于 CKCON 中）选择 SYSCLK 或由定时器 2 外部时钟选择位（TMR2CN 中的 T2XCLK[1:0]）定义的时钟源。时钟源的选择情况如下所示。

T2MH	T2XCLK[1:0]	TMR2H 时钟源
0	00	SYSCLK/12
0	01	smaRTClock /8
0	10	保留
0	11	比较器 0
1	X	SYSCLK

T2ML	T2XCLK[1:0]	TMR2L 时钟源
0	00	SYSCLK/12
0	01	smaRTClock /8
0	10	保留
0	11	比较器 0
1	X	SYSCLK

当 TMR2H 发生溢出时（从 0xFF 到 0x00），TF2H 被置 1；当 TMR2L 发生溢出时（从 0xFF 到 0x00），TF2L 被置 1。如果定时器 2 中断被允许，则每次 TMR2H 溢出时都将产生一个中断。如果定时器 2 中断被允许并且 TF2LEN 位（TMR2CN.5）被置 1，则每当 TMR2L 或 TMR2H 发生溢出时将产生一个中断。在 TF2LEN 位被置 1 的情况下，软件应检查 TF2H 和 TF2L 标志，以确定中断的来源。TF2H 和 TF2L 标志不能被硬件自动清除，必须用软件手动清除。

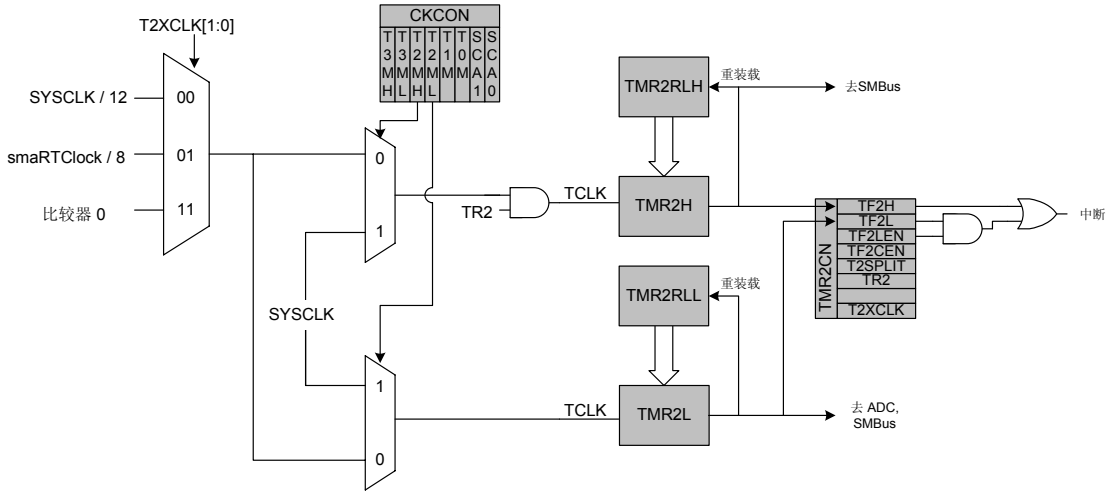


图 25.5 定时器 2 的 8 位方式原理框图

25.2.3　比较器 0/smaRTClock 捕捉方式

定时器 2 的捕捉方式允许使用系统时钟或系统时钟/12 测量比较器 0 和 smaRTClock 时钟。比较器 0 和 smaRTClock 周期也可以互相比较。通过将 TF2CEN 置 1 来使能定时器 2 的捕捉方式。当使用捕捉方式时，定时器 2 应被配置为 16 位自动重装载方式。

当捕捉方式被使能时，在每个比较器 0 的上升沿或每 8 个 smaRTClock 时钟周期产生一次捕捉事件，捕捉比较器 0 还是 smaRTClock 时钟取决于 T2RCLK 的设置。当捕捉事件发生时，定时器 2 的内容（TMR2H:TMR2L）被装入定时器 2 重装载寄存器（TMR2RLH:TMR2RLL），TF2H 标志被置位（如果定时器 2 中断被允许，会触发中断）。通过计算两个连续的定时器捕捉值的差值，可以确定比较器 0 或 smaRTClock 时钟的周期（相对于定时器 2 时钟）。为获得精确的测量值，定时器 2 的时钟频率应远高于捕捉时钟的频率。

例如：如果 T2ML = 1b，T2XCLK1 = 0b，TF2CEN = 1b，则定时器 2 使用 SYSCLK 作为时钟，每 8 个 smaRTClock 时钟进行一次捕捉。如果 SYSCLK 为 24.5 MHz，两次连续捕捉值的差值为 5984，则 smaRTClock 时钟频率为：

24.5 MHz / (5984/8) = 0.032754 MHz 或 32.754 KHz

该方式允许软件确定自振荡模式下准确的 smaRTClock 频率和两次连续的比较器 0 上升沿之间的时间，后者对检测触摸开关电容值的变化非常有用。

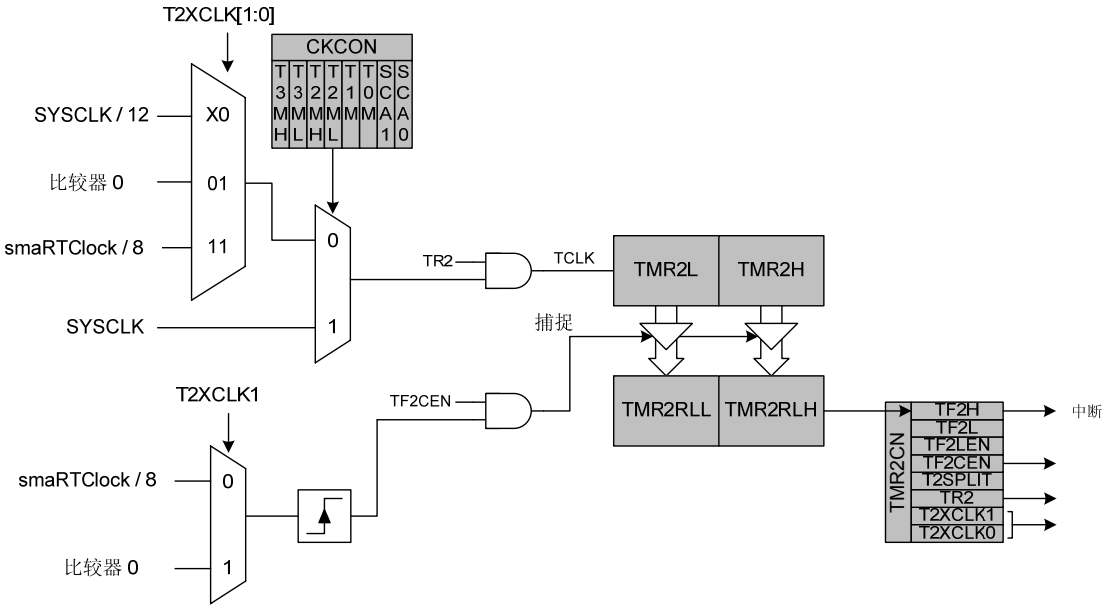


图 25.6　定时器 2 捕捉方式原理框图

SFR 定义 25.8　TMR2CN：定时器 2 控制寄存器

SFR 页：0x00
SFR 地址：0xC8　　（可位寻址）

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	复位值
TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2	T2XCLK1	T2XCLK0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　TF2H：定时器 2 高字节溢出标志
当定时器 2 高字节发生溢出时（从 0xFF 到 0x00）由硬件置 1。在 16 位方式，当定时器 2 发生溢出时（从 0xFFFF 到 0x0000）由硬件置 1。当定时器 2 中断被允许时，该位置 1 将导致 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。

位 6：　TF2L：定时器 2 低字节溢出标志
当定时器 2 低字节发生溢出时（从 0xFF 到 0x00）由硬件置 1。当定时器 2 中断被允许并且 TF2LEN 位被设置为逻辑 1 时，该位置 1 将产生中断。TF2L 在低字节溢出时置位，与定时器 2 的工作方式无关。该位不能由硬件自动清 0，必须用软件清 0。

位 5：　TF2LEN：定时器 2 低字节中断允许位
该位置 1 将允许定时器 2 低字节中断。如果定时器 2 中断也被允许（IE.5），则当定时器 2 低字节发生溢出时将产生一个中断。当定时器 2 工作在 16 位方式时，该位应被清 0。

位 4：　TF2CEN：定时器 2 捕捉使能位
当被置 1 时，该位使能定时器 2 的捕捉方式。

位 3：　T2SPLIT：定时器 2 双 8 位方式允许位
当该位被置 1 时，定时器 2 工作在双 8 位自动重装载定时器方式。否则，定时器 2 工作在 16 位自动重装载方式。

位 2：　TR2：定时器 2 运行控制
通过将该位置 1 来使能定时器 2。在 8 位方式，该位只使能/禁止 TMR2H，TMR2L 总是处于运行状态。

位 1-0：　T2XCLK[1:0]：定时器 2 外部时钟选择
该位选择定时器 2 的“外部”和“捕捉触发”时钟源。如果定时器 2 工作在 8 位方式，该位为两个 8 位定时器选择“外部”时钟源。但仍可用定时器 2 时钟选择位（CKCON 中的 T2MH 和 T2ML）在“外部”时钟和系统时钟之间作出选择。
注：外部时钟源与系统时钟同步。
00：外部时钟为 SYSCLK/12。捕捉触发时钟为 smaRTClock/8。
01：外部时钟为比较器 0。捕捉触发时钟为 smaRTClock/8。
10：外部时钟为 SYSCLK/12。捕捉触发时钟为比较器 0。
01：外部时钟为比较器 0。捕捉触发时钟为 smaRTClock/8。

SFR 定义 25.9　TMR2RLL：定时器 2 重载寄存器低字节

SFR 页：0x00
SFR 地址：0xCA

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	

位 7-0：　TMR2RLL[7:0]：定时器 2 重载寄存器的低字节
TMR2RLL 保持定时器 2 重载值的低字节。

SFR 定义 25.10　TMR2RLH：定时器 2 重载寄存器高字节

SFR 页：0x00
SFR 地址：0xCB

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	

位 7-0：　TMR2RLH[7:0]：定时器 2 重载寄存器的高字节
TMR2RLH 保持定时器 2 重载值的高字节。

SFR 定义 25.11　TMR2L：定时器 2 低字节

SFR 页：0x00
SFR 地址：0xCC

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	

位 7-0：　TMR2L[7:0]：定时器 2 的低字节
在 16 位方式，TMR2L 寄存器保持 16 位定时器 2 的低字节。在 8 位方式，TMR2L 中保持 8 位低字节定时器的计数值。

SFR 定义 25.12　TMR2H：定时器 2 高字节

SFR 页：0x00
SFR 地址：0xCD

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	

位 7-0：　TMR2H[7:0]：定时器 2 的高字节
在 16 位方式，TMR2H 寄存器保持 16 位定时器 2 的高字节。在 8 位方式，TMR2H 中保持 8 位高字节定时器的计数值。

25.3 定时器 3

定时器 3 是一个 16 位的定时器，由两个 8 位的 SFR 组成：TMR3L（低字节）和 TMR3H（高字节）。定时器 3 可以工作在 16 位自动重装载方式或 8 位自动重装载方式（两个 8 位定时器）。T3SPLIT 位（TMR3CN.3）定义定时器 3 的工作方式。定时器 3 还可被用于捕捉方式，以测量外部振荡器或比较器 1 的周期（相对于另一振荡器）。测量比较器 1 周期（相对于系统时钟）的能力使触摸开关的实现非常容易。

定时器 3 的时钟源可以是系统时钟、系统时钟/12、外部振荡源时钟/8 或比较器 1 输出。注意，外部振荡源时钟/8 和比较器 1 与系统时钟同步。

25.3.1 16 位自动重装载定时器

当 T3SPLIT 位（TMR3CN.3）为 0 时，定时器 3 工作在自动重装载的 16 位定时器方式。定时器 3 可以使用 SYSCLK、SYSCLK/12、外部振荡器时钟/8 或比较器 1 输出作为时钟源。当 16 位定时器寄存器发生溢出（从 0xFFFF 到 0x0000）时，定时器 3 重载寄存器（TMR3RLH 和 TMR3RLL）中的 16 位计数初值被自动装入到定时器 3 寄存器（如图 25.7 所示），并且定时器 3 高字节溢出标志 TF3H（TMR3CN.7）被置 1。如果定时器 3 中断被允许（EIE1.7 被置 1），则每次溢出都将产生中断。另外，如果定时器 3 中断被允许并且 TF3LEN 位（TMR3CN.5）被置 1，则每次低 8 位（TMR3L）溢出时（从 0xFF 到 0x00）将产生中断。

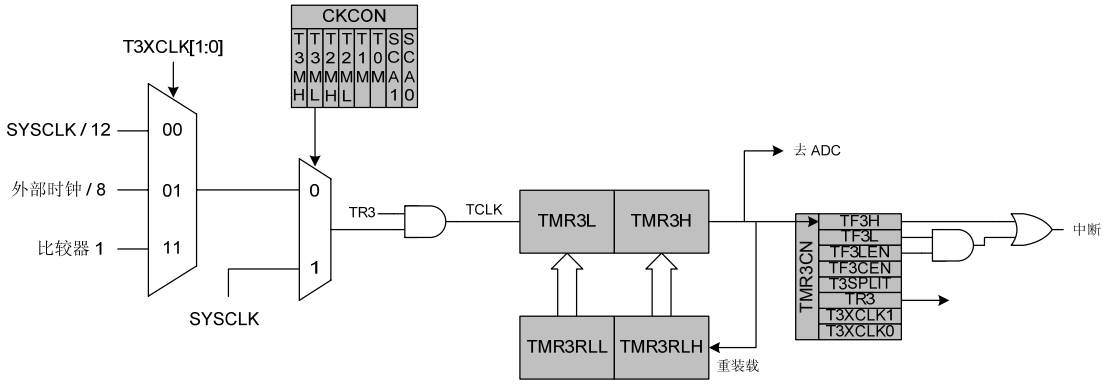


图 25.7 定时器 3 的 16 位方式原理框图

25.3.2　8 位自动重装载定时器

当 T3SPLIT 被置 1 时，定时器 3 工作双 8 位定时器方式（TMR3H 和 TMR3L）。这两个 8 位定时器都工作在自动重装载方式（如图 25.8 所示）。TMR3RLL 保持 TMR3L 的重载值，而 TMR3RLH 保持 TMR3H 的重载值。TMR3CN 中的 TR3 是 TMR3H 的运行控制位。当定时器 3 被配置为 8 位方式时，TMR3L 总是处于运行状态。

每个 8 位定时器都可以被配置为使用 SYSCLK、SYSCLK/12、外部振荡器时钟/8 或比较器 1 作为时钟源。定时器 3 时钟选择位（CKCON 的 T3MH 和 T3ML）选择 SYSCLK 或由定时器 3 外部时钟选择位（TMR3CN 中的 T3XCLK[1:0]）定义的时钟源。时钟源的选择情况如下所示。

T3MH	T3XCLK[1:0]	TMR3H 时钟源
0	00	SYSCLK/12
0	01	比较器 1
0	10	保留
0	11	外部时钟/8
1	X	SYSCLK

T3ML	T3XCLK[1:0]	TMR3L 时钟源
0	00	SYSCLK/12
0	01	比较器 1
0	10	保留
0	11	外部时钟/8
1	X	SYSCLK

当 TMR3H 发生溢出时（从 0xFF 到 0x00），TF3H 被置 1；当 TMR3L 发生溢出时（从 0xFF 到 0x00），TF3L 被置 1。如果定时器 3 中断被允许，则每次 TMR3H 溢出时都将产生一个中断。如果定时器 3 中断被允许并且 TF3LEN 位（TMR3CN.5）被置 1，则每当 TMR3L 或 TMR3H 发生溢出时将产生一个中断。在 TF3LEN 位被置 1 的情况下，软件必须检查 TF3H 和 TF3L 标志，以确定定时器 3 中断的来源。TF3H 和 TF3L 中断标志不能被硬件自动清除，必须通过软件手动清除。

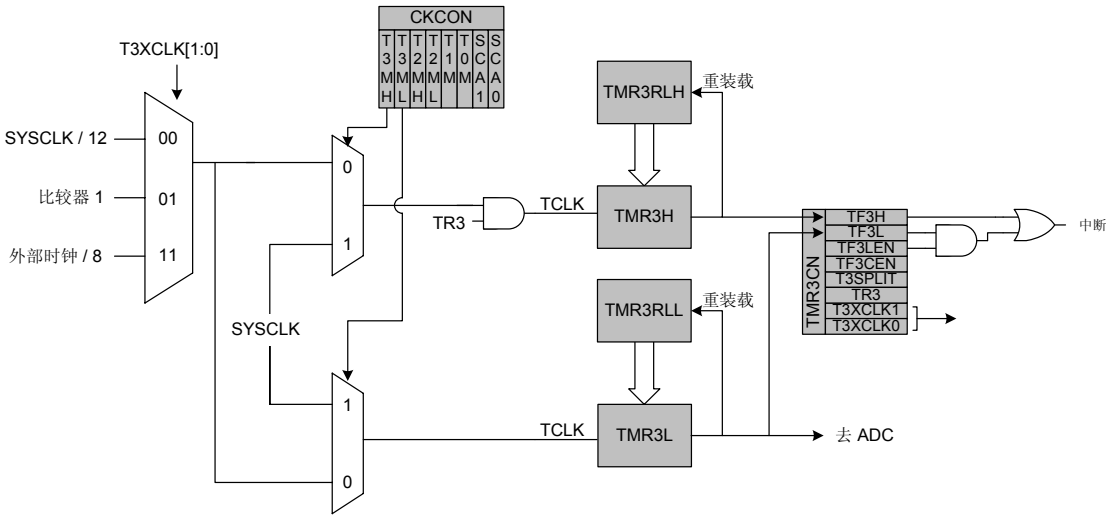


图 25.8　定时器 3 的 8 位方式原理框图

25.3.3 比较器 1/外部振荡器捕捉方式

定时器3的捕捉方式允许使用系统时钟或系统时钟/12测量比较器1或外部振荡器的周期。比较器 1 和外部振荡器周期也可以互相比较。

通过将 TF3CEN 置 1 来使能定时器 3 的比较器 1/外部振荡器捕捉方式。在该方式时，T3SPLIT 应被设置为 0，因为使用全 16 位定时器。

当捕捉方式被使能时，在每个比较器 1 的上升沿或每 8 个外部时钟周期产生一次捕捉事件，捕捉比较器 1 还是外部时钟周期取决于 T3XCLK1 的设置。当捕捉事件发生时，定时器 3 的内容（TMR3H:TMR3L）被装入定时器 3 重装载寄存器（TMR3RLH:TMR3RLL），TF3H 标志被置位（如果定时器 3 中断被允许，会触发中断）。通过计算两个连续的定时器捕捉值的差值，可以确定比较器 1 或外部时钟的周期（相对于定时器 3 时钟）。为获得精确的测量值，定时器 3 的时钟频率应远高于捕捉时钟的频率。

例如：如果 T3ML = 1b，T3XCLK1 = 0b，TF3CEN = 1b，则定时器 3 使用 SYSCLK 作为时钟，每个比较器上升沿进行一次捕捉。如果 SYSCLK 为 24.5 MHz，两次连续捕捉值的差值为 350，则比较器 1 的周期为：

350×(1/24.5 MHz) = 14.2 μs。

该方式允许软件确定 C 和 RC 方式外部振荡器的准确频率，或两次连续的比较器 1 上升沿之间的时间，后者对检测触摸开关电容值的变化非常有用。

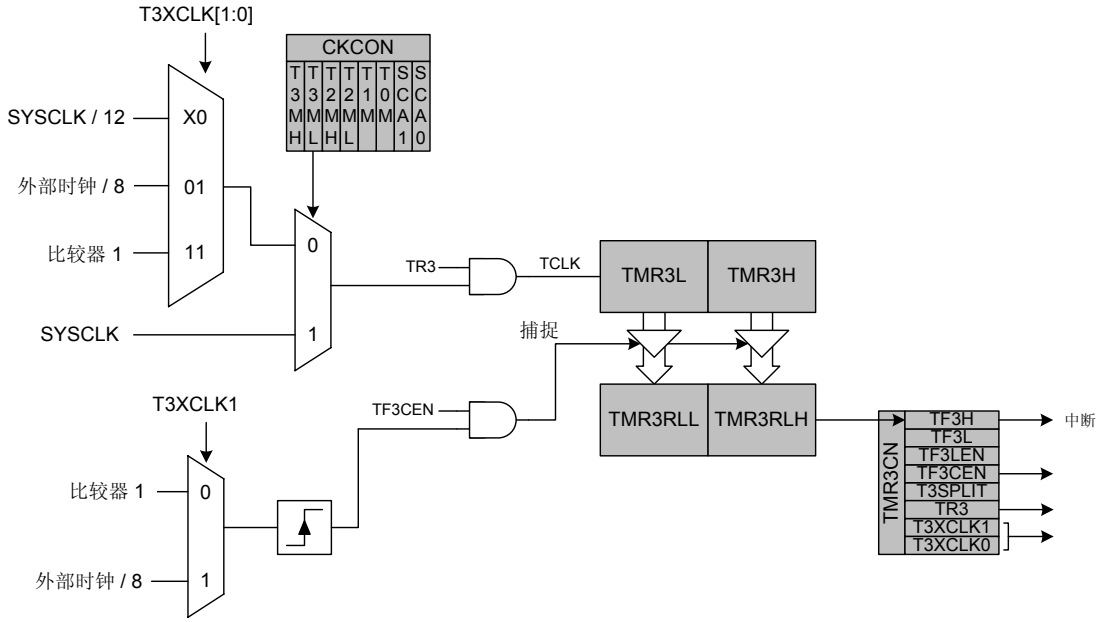


图 25.9 定时器 3 捕捉方式原理框图

SFR 定义 25.13　TMR3CN：定时器 3 控制寄存器

SFR 页：0x00
SFR 地址：0x91

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
TF3H	TF3L	TF3LEN	TF3CEN	T3SPLIT	TR3	T3XCLK1	T3XCLK0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　TF3H：定时器 3 高字节溢出标志
当定时器 3 高字节发生溢出时（从 0xFF 到 0x00）由硬件置 1。在 16 位方式，当定时器 3 发生溢出时（从 0xFFFF 到 0x0000）由硬件置 1。当定时器 3 中断被允许时，该位置 1 将导致 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。

位 6：　TF3L：定时器 3 低字节溢出标志
当定时器 3 低字节发生溢出时（从 0xFF 到 0x00）由硬件置 1。当定时器 3 中断被允许并且 TF3LEN 位被设置为逻辑 1 时，该位置 1 将产生中断。TF3L 在低字节溢出时置位，与定时器 3 的工作方式无关。该位不能由硬件自动清 0。

位 5：　TF3LEN：定时器 3 低字节中断允许位
该位置 1 将允许定时器 3 低字节中断。如果定时器 3 中断也被允许，则当定时器 3 低字节发生溢出时将产生一个中断。

位 4：　TF3CEN：定时器 3 的比较器 1/外部振荡器捕捉使能位
当被置 1 时，该位使能定时器 3 的捕捉方式。

位 3：　T3SPLIT：定时器 3 双 8 位方式允许位
当该位被置 1 时，定时器 3 工作在双 8 位自动重装载定时器方式。否则，定时器 3 工作在 16 位自动重装载方式。

位 2：　TR3：定时器 3 运行控制
通过将该位置 1 来使能定时器 3。在 8 位方式，该位只使能/禁止 TMR3H，TMR3L 总是处于运行状态。

位 1-0：　T3XCLK[1:0]：定时器 3 外部时钟选择
该位选择定时器 3 的“外部”和“捕捉触发”时钟源。如果定时器 3 工作在 8 位方式，该位为两个 8 位定时器选择“外部”时钟源。但仍可用定时器 3 时钟选择位（CKCON 中的 T3MH 和 T3ML）在“外部”时钟和系统时钟之间作出选择。
注：外部时钟源与系统时钟同步。
00：外部时钟为 SYSCLK/12。捕捉触发时钟为比较器 1。
01：外部时钟为外部振荡器/8。捕捉触发时钟为比较器 1。
10：外部时钟为 SYSCLK/12。捕捉触发时钟外部振荡器/8。
11：外部时钟为比较器 1。捕捉触发时钟为外部振荡器/8。

SFR 定义 25.14　TMR3RLL：定时器 3 重载寄存器低字节

SFR 页：0x00
SFR 地址：0x92

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	

位 7-0：　TMR3RLL[7:0]：定时器 3 重载寄存器的低字节
TMR3RLL 保存定时器 3 重载值的低字节。

SFR 定义 25.15　TMR3RLH：定时器 3 重载寄存器高字节

SFR 页：0x00
SFR 地址：0x93

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	

位 7-0：　TMR3RLH[7:0]：定时器 3 重载寄存器的高字节
TMR3RLH 保存定时器 3 重载值的高字节。

SFR 定义 25.16　TMR3L：定时器 3 低字节

SFR 页：0x00
SFR 地址：0x94

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	

位 7-0：　TMR3L[7:0]：定时器 3 的低字节
在 16 位方式，TMR3L 寄存器保持 16 位定时器 3 的低字节。在 8 位方式，TMR3L 中保持 8 位低字节定时器的计数值。

SFR 定义 25.17　TMR3H：定时器 3 高字节

SFR 页：0x00
SFR 地址：0x95

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	

位 7-0：　TMR3H[7:0]：定时器 3 的高字节
在 16 位方式，TMR3H 寄存器保持 16 位定时器 3 的高字节。在 8 位方式，TMR3H 中保持 8 位高字节定时器的计数值。

26. 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的计数器/定时器相比，它需要较少的CPU干预。PCA由一个专用的16位计数器/定时器和6个16位捕捉/比较模块组成。每个捕捉/比较模块有其自己的I/O线（CEXn），这些I/O线在被允许时通过交叉开关连到端口I/O。计数器/定时器由一个可编程的时基信号驱动，时基信号可以在七个时钟源中选择：系统时钟、系统时钟/4、系统时钟/12、外部振荡器时钟/8、smaRTClock/8、定时器0溢出或ECI输入引脚上的外部时钟信号。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、8~11位PWM和16位PWM，每个捕捉/比较模块的工作方式都可以被独立配置。在使用实时时钟（RTC）功能时，外部振荡器时钟选项是理想的选择，此时用内部振荡器驱动系统时钟，而PCA的时钟由一个精确的外部振荡器提供。对PCA的配置和控制是通过系统控制器的特殊功能寄存器来实现的。PCA的原理框图示于图26.1。

注意：PCA的模块5可被用作看门狗定时器（WDT），在系统复位后即被使能为该方式。**在看门狗方式被使能时，对某些PCA寄存器的访问受到限制**，详见26.4节。

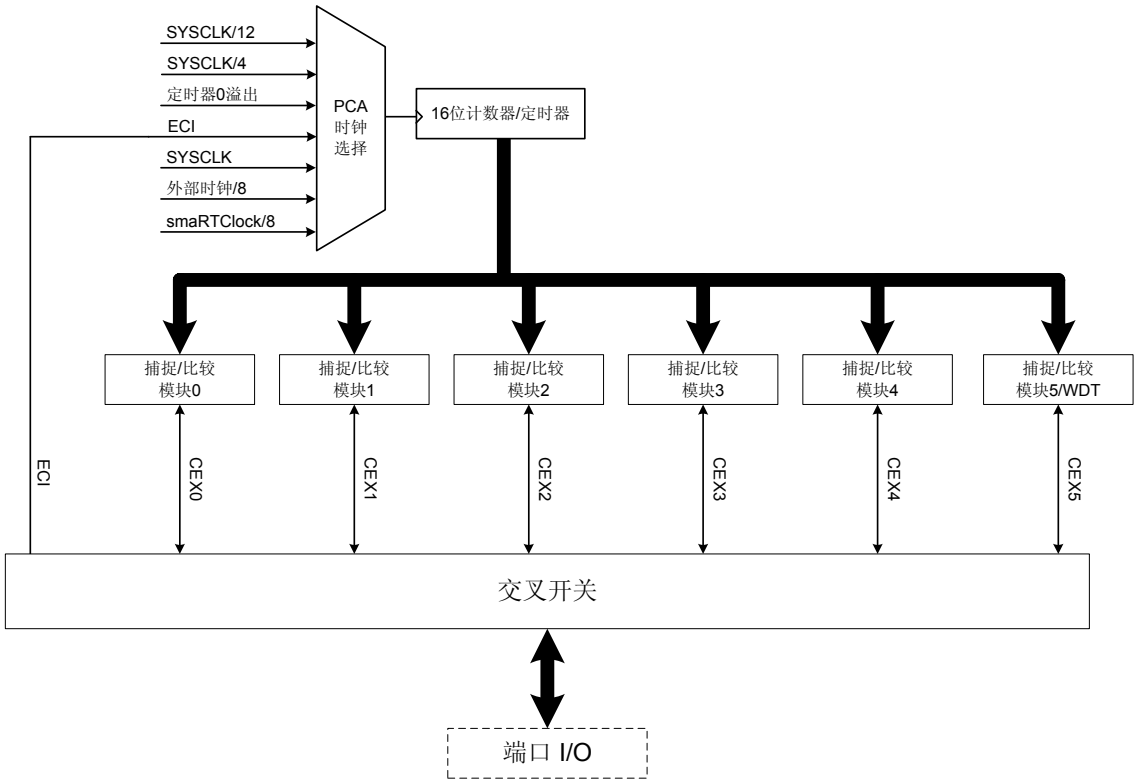


图 26.1　PCA 原理框图

26.1 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 时，“瞬象寄存器”自动锁存 PCA0H 的值，随后读 PCA0H 时将访问这个“瞬象寄存器”而不是 PCA0H 本身。**先读 PCA0L 寄存器可以保证正确读取整个 16 位 PCA0 计数器的值**。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，如表 26.1 所示。

当计数器/定时器溢出时（从 0xFFFF 到 0x0000），PCA0CN 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，必须用软件清除。清除 PCA0MD 寄存器中的 CIDL 位将允许 PCA 在微控制器内核处于空闲方式时继续正常工作。

表 26.1 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 下降沿（最大速率 = 系统时钟频率/4）
1	0	0	系统时钟
1	0	1	外部振荡器/8*
1	1	0	smaRTClock/8*
1	1	1	保留
*注：外部振荡器 8 分频和 smaRTClock 时钟 8 分频与系统时钟同步。			

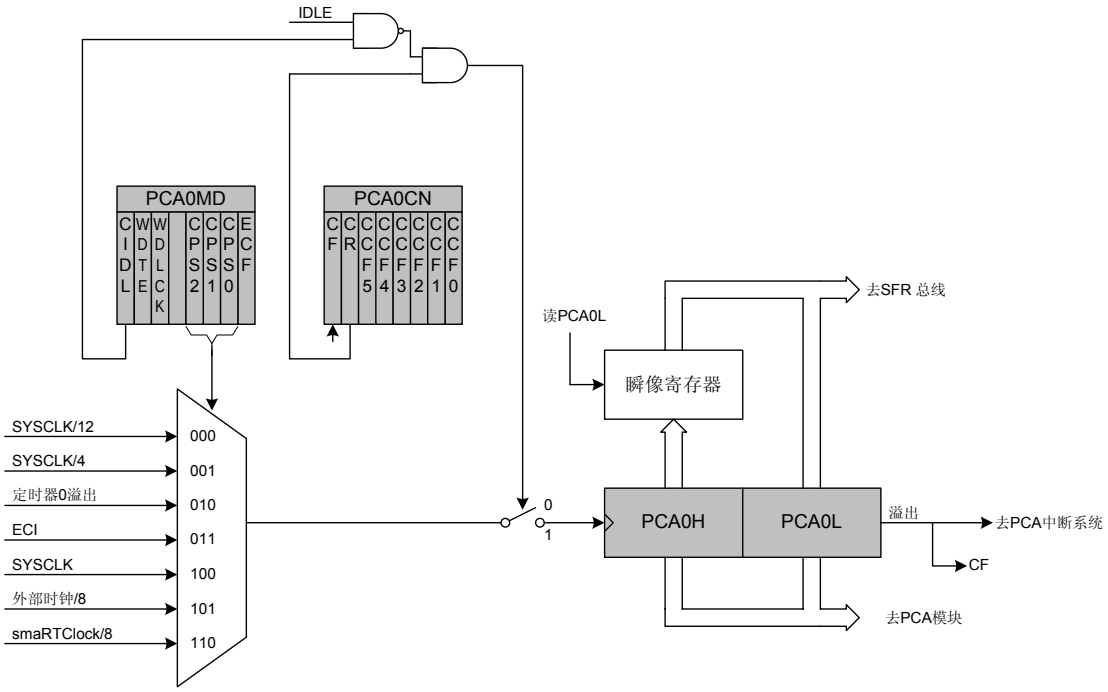


图 26.2　PCA 计数器/定时器原理框图

26.2 PCA0 中断源

图 26.3 给出了 PCA 中断树的原理框图。有 8 个独立的事件标志可用于产生 PCA0 中断。它们是：主 PCA 计数器溢出标志（CF），在 16 位 PCA0 计数器溢出时置 1；中间溢出标志（COVF），可以在 PCA0 计数器的第 8、9、10 或 11 位溢出时置 1；以及每个 PCA 通道的独立标志位（CCF0、CCF1、CCF2、CCF3、CCF4 和 CCF5），这些标志根据对应模块的工作方式被置 1。当触发条件发生时，这些事件标志将被置 1。通过使用相应的中断允许位（ECF 对应 CF，ECOV 对应 COVF，ECCFn 对应 CCFn）来选择要产生中断的每个标志。要想使任何一个独立的中断源能被处理器识别，必须先总体允许 PCA0 中断。通过将 EA 位（IE.7）和 EPCA0 位（EIE1.4）设置为逻辑 1 来总体允许 PCA0 中断。

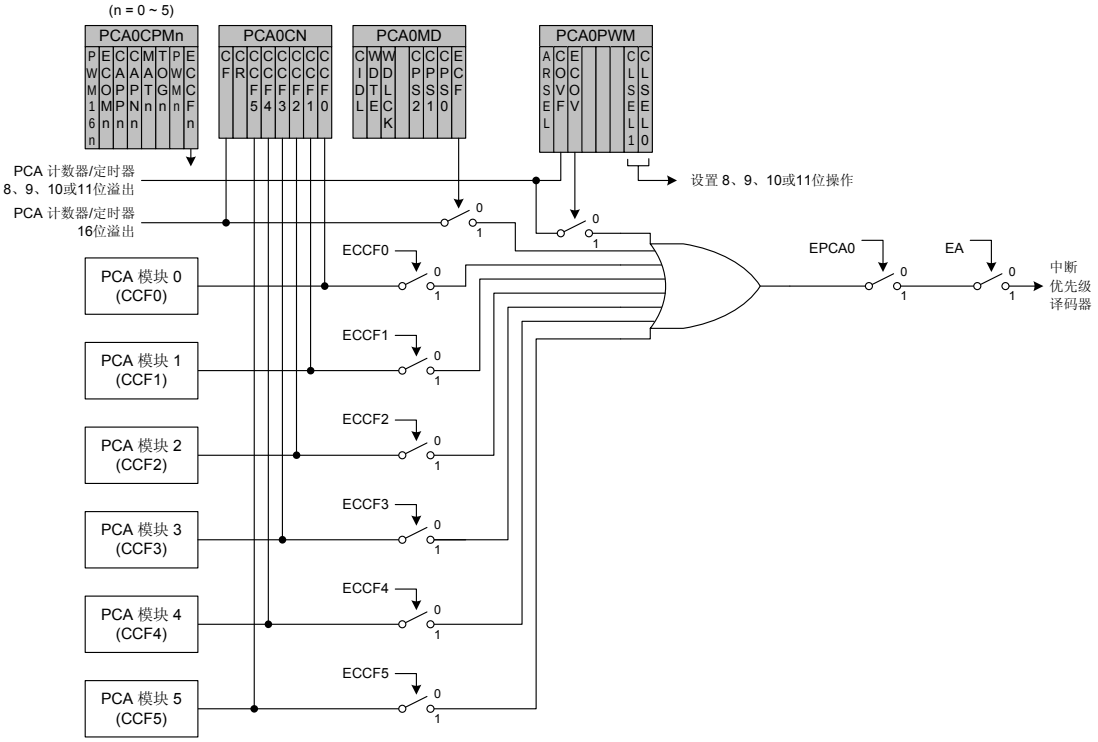


图 26.3 PCA 中断原理框图

26.3 捕捉/比较模块

每个模块都可被配置为独立工作在下述六种工作方式致意：边沿触发捕捉、软件定时器、高速输出、频率输出、8~11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。表26.2概述了用于选择PCA捕捉/比较模块工作方式的PCA0CPMn和PCA0PWM寄存器中各位的设置情况。注意：被设置为8、9、10或11位PWM方式的所有模块使用相同的周期长度（8~11位）。置位PCA0CPMn寄存器中的ECCFn位将允许模块的CCFn中断。

表 26.2　PCA 捕捉/比较模块的 PCA0CPM 和 PCA0PWM 寄存器的位设置

工作方式	PCA0CPMn								PCA0PWM				
位号	7	6	5	4	3	2	1	0	7	6	5	4-2	1-0
用 CEXn 的正沿触发捕捉	X	X	1	0	0	0	0	A	0	X	B	XXX	XX
用 CEXn 的负沿触发捕捉	X	X	0	1	0	0	0	A	0	X	B	XXX	XX
用 CEXn 的任一边沿触发捕捉	X	X	1	1	0	0	0	A	0	X	B	XXX	XX
软件定时器	X	C	0	0	1	0	0	A	0	X	B	XXX	XX
高速输出	X	C	0	0	1	1	0	A	0	X	B	XXX	XX
频率输出	X	C	0	0	0	1	1	A	0	X	B	XXX	XX
8 位脉冲宽度调制器（注 7）	0	C	0	0	E	0	1	A	0	X	B	XXX	00
9 位脉冲宽度调制器（注 7）	0	C	0	0	E	0	1	A	D	X	B	XXX	01
10 位脉冲宽度调制器（注 7）	0	C	0	0	E	0	1	A	D	X	B	XXX	10
11 位脉冲宽度调制器（注 7）	0	C	0	0	E	0	1	A	D	X	B	XXX	11
16 位脉冲宽度调制器	1	C	0	0	E	0	1	A	0	X	B	XXX	XX

注：

1. X = 任意值（对每个单独的模块而言，该位为 1 或为 0 没有功能差别）。
2. A = 使能该模块的中断（CCFn 被置 1 时触发 PCA 中断）。
3. B = 使能第 8、9、10、11 位溢出中断（取决于 CLSEL[1:0]的设置）。
4. C = 当被设置为 0 时，数字比较器不工作。对于高速和频率输出方式，相关引脚的电平不切换。在任何一种 PWM 方式，将产生 0%的占空比（输出 = 0）。
5. D = 选择通过地址 PCA0CPHn 和 PCA0CPLn 访问的是相关通道的捕捉/比较寄存器（0）还是自动重装载寄存器（1）。
6. E = 当被置 1 时，一个匹配事件会导致相关通道的 CCFn 标志被置 1。
7. 被设置为 8、9、10 或 11 位 PWM 方式的所有模块都使用相同的周期长度设置。

26.3.1 边沿触发的捕捉方式

在该方式，CEXn引脚上出现的有效电平跳变导致PCA捕捉PCA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（PCA0CPLn和PCA0CPHn）。PCA0CPMn寄存器中的CAPPn和CAPNn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）或任何变化（正沿或负沿）。当捕捉发生时，PCA0CN中的捕捉/比较标志（CCFn）被置为逻辑1。如果该模块的CCFn中断被允许，将产生一个中断请求。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。如果CAPPn和CAPNn位都被设置为逻辑1，则可以通过直接读CEXn对应端口引脚的状态来确定本次捕捉是由上升沿触发还是由下降沿触发。

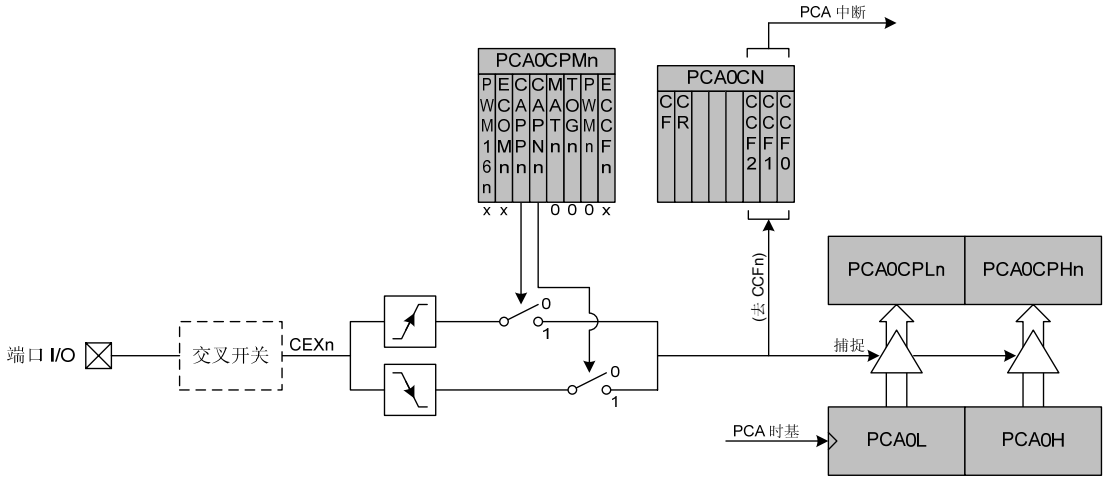


图26.4　PCA捕捉方式原理框图

注意：CEXn 输入信号必须在高电平或低电平期间至少保持两个系统时钟周期，以保证能够被硬件识别。

26.3.2 软件定时器（比较）方式

在软件定时器方式，PCA将计数器/定时器的值与模块的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。如果该模块的CCFn中断被允许，将产生一个中断请求。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。置位PCA0CPMn寄存器中的ECOMn和MATn位将使能软件定时器方式。

注意，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ECOMn位清0；向PCA0CPHn写入时将ECOMn位置1。

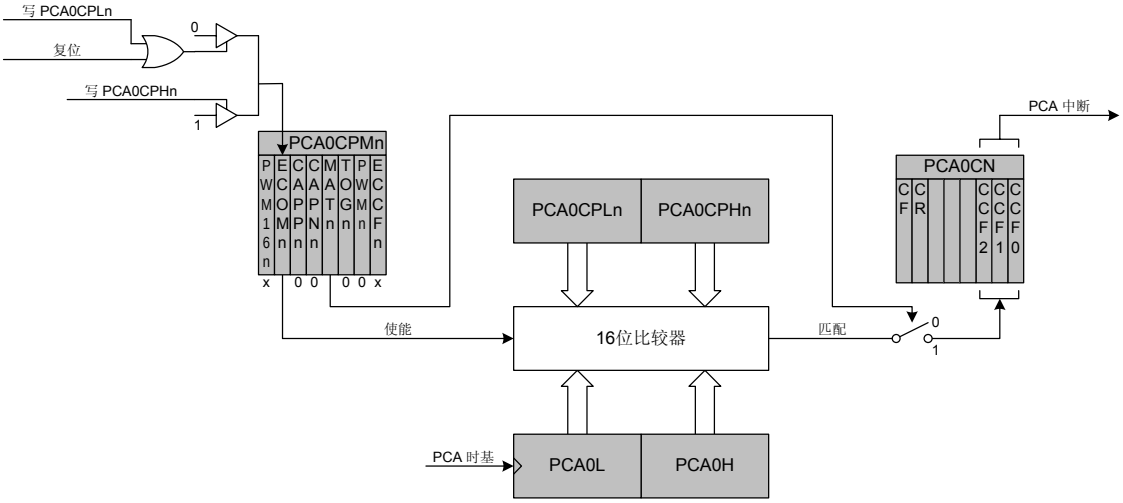


图 26.5　PCA 软件定时器方式原理框图

26.3.3 高速输出方式

在高速输出方式，每当PCA计数器与模块的16位捕捉/比较寄存器（PCA0CPHn和PCA0CPLn）发生匹配时，模块的CEXn引脚上的逻辑电平将发生切换。当发生匹配时，PCA0CN中的捕捉/比较标志（CCFn）被置为逻辑1。如果该模块的CCFn中断被允许，将产生一个中断请求。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。置位PCA0CPMn寄存器中的TOGn、MATn和ECOMn位将使能高速输出方式。如果ECOMn被清0，则模块的CEXn引脚将保持其状态，在下一个匹配事件到来时不会发生电平切换。

注意：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ECOMn位清0；向PCA0CPHn写入时将ECOMn位置1。

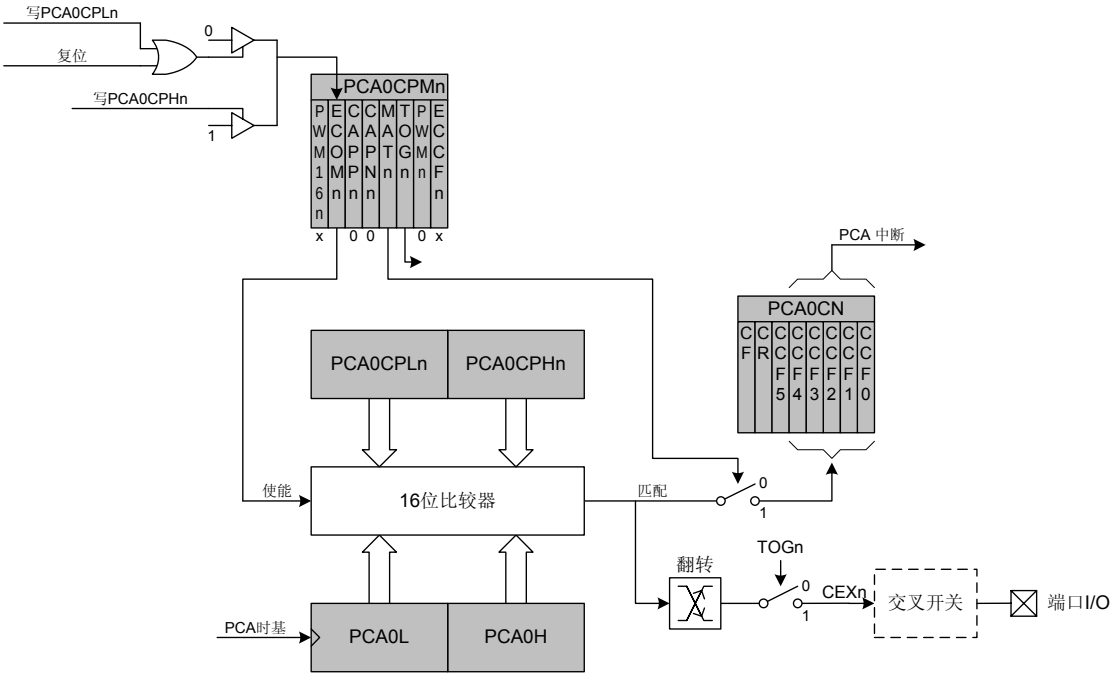


图 26.6　PCA 高速输出方式原理框图

26.3.4 频率输出方式

频率输出方式在模块的CEXn引脚产生可编程频率的方波。捕捉/比较模块的高字节保持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26.1定义：

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

注：对于该方程，PCA0CPHn寄存器中的值为0x00时，相当于256。

方程26.1 方波输出频率

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，CEXn的电平发生翻转，高字节中的偏移值被加到PCA0CPLn中的匹配值。通过将PCA0CPMn寄存器中ECOMn、TOGn和PWMn位置1来使能频率输出方式。注意：在该方式，MATn应被设置为0。如果MATn被设置为1，则当16位PCA0计数器与该模块的16位捕捉比较寄存器相等时，模块的CCFn标志被置1。

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ECOMn位清0；向PCA0CPHn写入时将ECOMn位置1。

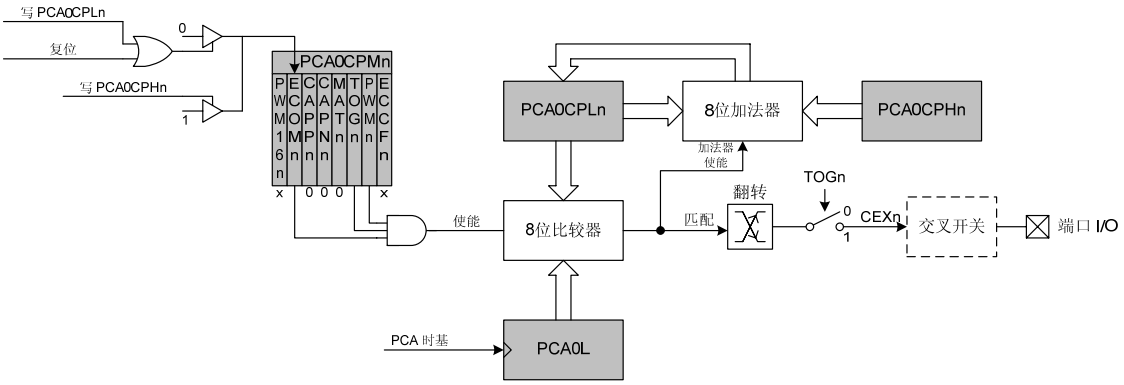


图 26.7　PCA 频率输出方式原理框图

26.3.5 8 位、9 位、10 位和 11 位脉宽调制器方式

每个模块都可以被独立地用于在对应的CEXn引脚产生脉宽调制（PWM）输出。PWM输出的频率取决于PCA计数器/定时器的时基和PWM周期长度的设置（8、9、10或11位）。为了向后兼容其它器件的8位PWM方式，8位PWM方式与9、10和11位PWM方式稍有不同。注意：被设置为8、9、10或11位PWM方式的所有模块使用相同的周期长度。例如，不可能将一个通道配置为8位PWM方式，而将另一通道配置为11位方式。但其它PCA通道可以被独立地配置为引脚捕捉、高速输出、软件定时器、频率输出或16位PWM方式。

26.3.5.1 8 位脉宽调制器方式

使用模块的捕捉/比较寄存器PCA0CPLn改变8位PWM方式PWM输出信号的占空比。当PCA计数器/定时器的低字节（PCA0L）与PCA0CPLn中的值相等时，CEXn引脚上的输出被置1；当PCA0L中的计数值溢出时，CEXn输出被复位（见图26.8）。当计数器/定时器的低字节PCA0L溢出时（从0xFF到0x00），保存在模块的捕捉/比较高字节（PCA0CPHn）中的值被自动装入到PCA0CPLn，不需软件干预。通过将PCA0CPMn寄存器中的ECOMn和PWMn位置1并将PCA0PWM寄存器中的CLSEL位设置为00b来使能8位脉冲宽度调制器方式。如果MATn被设置为1，则每当发生一次8位比较器匹配时（上升沿），模块的CCFn标志被置1。PCA0PWM中的COVF标志可用于检测溢出（下降沿），每256个PCA时钟周期发生一次溢出。8位PWM方式的占空比由方程26.2给出。

注意：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写入时将ECOMn位清0；向PCA0CPHn写入时将ECOMn位置1。

$$占空比 = \frac{(256 - PCA0CPHn)}{256}$$

方程26.2 8位PWM占空比

由方程26.2可知，最大占空比为100%（PCA0CPHn = 0），最小占空比为0.39%（PCA0CPHn = 0xFF）。可以通过将ECOMn位清0来产生0%的占空比。

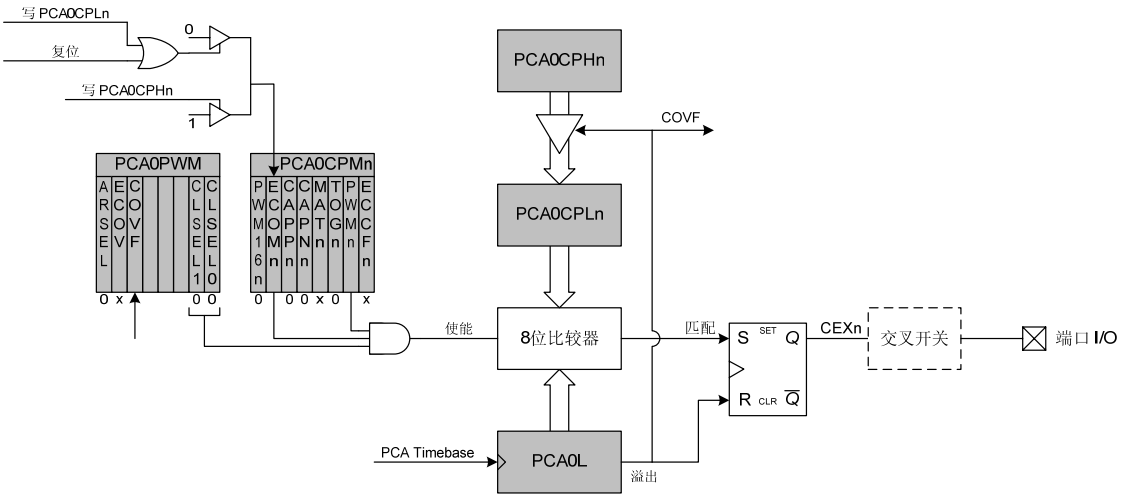


图 26.8 PCA 的 8 位 PWM 方式原理框图

26.3.5.2　9/10/11 位脉宽调制器方式

通过写一个“自动重装载”寄存器（该寄存器被双映射到PCA0CPHn和PCA0CPLn地址）改变9/10/11位PWM方式PWM输出信号的占空比。定义占空比的写入数据在寄存器中应被右对齐。当PCA0PWM中的ARSEL位被设置为1时，访问自动重装载寄存器。当PCA0PWM中的ARSEL位被设置为0时，访问捕捉/比较寄存器。

当PCA0计数器的低N位与模块的捕捉/比较寄存器（PCA0CPn）中的值一致时，CEXn引脚上的输出被置为高电平。当计数从N位溢出时，CEXn输出被置为低电平（见图26.9）。当计数从N位溢出时，COVF标志被置位，保存在模块的自动重装载寄存器中的值被装入到捕捉/比较寄存器。值N由PCA0PWM寄存器中的CLSEL位确定。

通过将PCA0CPMn寄存器中的ECOMn和PWMn位置1并将PCA0PWM寄存器中的CLSEL位设置为所希望的周期长度（非8位）来使能9、10或11位脉冲宽度调制器方式。如果MATn被设置为1，则每当发生一次比较器匹配时（上升沿），模块的CCFn标志被置1。PCA0PWM中的COVF标志可用于检测溢出（下降沿），每512（9位）、1024（10位）或2048（11位）个PCA时钟周期发生一次溢出。9/10/11位PWM方式的占空比由方程26.3给出，其中N为PWM周期的位数。

注意：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写入时将ECOMn位清0；向PCA0CPHn写入时将ECOMn位置1。

$$占空比 = \frac{\left(2^N - PCA0CPn\right)}{2^N}$$

方程26.3　9、10或11位PWM占空比

可以通过将ECOMn位清0来产生0%的占空比。

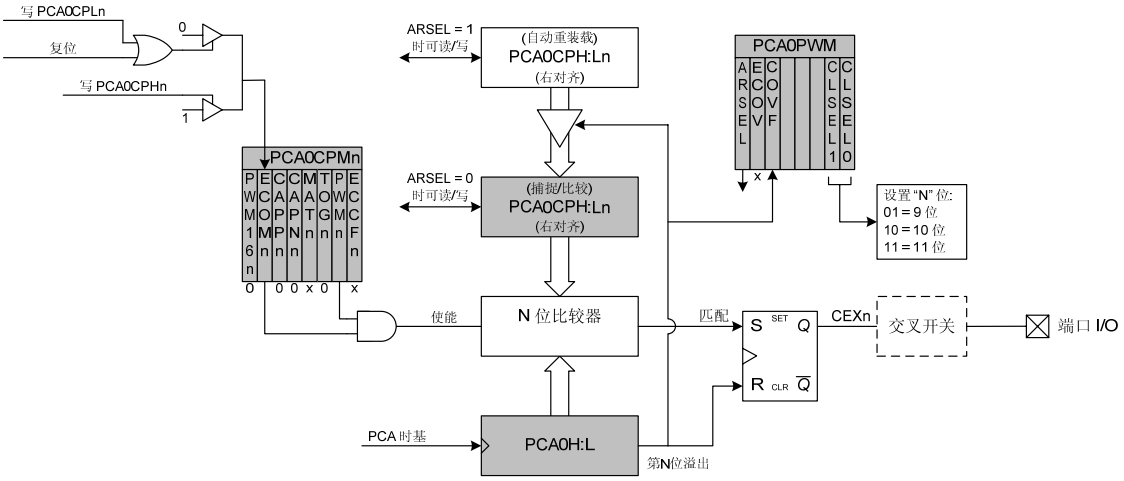


图 26.9 PCA 的 9、10 或 11 位 PWM 方式原理框图

26.3.6 16 位脉宽调制器方式

PCA模块还可以工作在16位PWM方式。16位PWM方式与其它（8/9/10/11位）PWM方式无关。在该方式，16位捕捉/比较模块定义PWM信号低电平时间的PCA时钟数。当PCA计数器与模块的内容一致时，CEXn的输出被置为高电平；当16位计数器溢出时，CEXn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PCA的CCFn匹配中断同步。通过将PCA0CPMn寄存器中的ECOMn、PWMn和PWM16n位置1来使能16位PWM方式。为了得到可变的占空比，应允许匹配中断（ECCFn = 1并且MATn = 1），以帮助对捕捉/比较寄存器的写操作同步。如果MATn被设置为1，则每当发生一次16位比较器匹配时（上升沿），模块的CCFn标志被置1。PCA0CN的CF标志可用于检测溢出（下降沿）。16位PWM方式的占空比由方程26.4给出。

注意：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写入时将ECOMn位清0；向PCA0CPHn写入时将ECOMn位置1。

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

方程26.4　16位PWM占空比

由方程26.4可知，最大占空比为100%（PCA0CPn = 0），最小占空比为0.0015%（PCA0CPn = 0xFFFF）。可以通过将ECOMn位清0来产生0%的占空比。

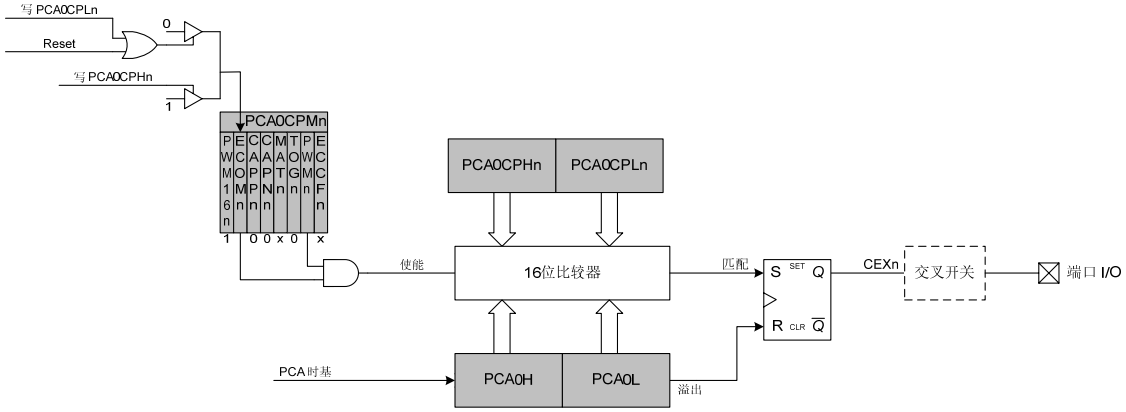


图 26.10　PCA 的 16 位 PWM 方式原理框图

26.4 看门狗定时器方式

通过 PCA 的模块 5 可以实现可编程看门狗定时器（WDT）功能。如果两次对 WDT 更新寄存器（PCA0CPH2）的写操作相隔的时间超过规定的极限，WDT 将产生一次复位。可以根据需要用软件配置和使能/禁止 WDT。

当 PCA0MD 寄存器中的 WDTE 位被置 1 时，模块 5 被作为看门狗定时器（WDT）使用。模块 5 高字节与 PCA 计数器的高字节比较；模块 5 低字节保持执行 WDT 更新时要使用的偏移值。**在系统复位后看门狗定时器被使能。在看门狗被使能时，对某些 PCA 寄存器的写操作受到限制。**WDT 会在代码开始执行后不长时间就产生复位。为了避免这种复位，WDT 应被显式禁止（如果需要，可以重新配置和使能 WDT）。

26.4.1 看门狗定时器操作

当 WDT 被使能时：

- PCA 计数器被强制运行。
- 不允许写 PCA0L 和 PCA0H。
- PCA 时钟源选择位（CPS2-CPS0）被冻结。
- PCA 空闲控制位（CIDL）被冻结。
- 模块 5 被强制进入软件定时器方式。
- 对模块 5 方式寄存器（PCA0CPM5）的写操作被禁止。

当 WDT 被使能时，写 CR 位并不改变 PCA 计数器的状态；计数器将一直保持运行状态，直到 WDT 被禁止。如果 WDT 被使能，但用户软件没有使能 PCA 计数器，则读 PCA 运行控制（CR）位时将返回 0。如果在 WDT 被使能时 PCA0CPH5 和 PCA0H 发生匹配，则会产生系统复位。为了防止 WDT 复位，需要通过向 PCA0CPH5 写入任意值来更新 WDT。在写 PCA0CPH5 时，PCA0H 的值加上 PCA0CPL5 中保存的偏移值后被装入到 PCA0CPH5（见图 26.11）。

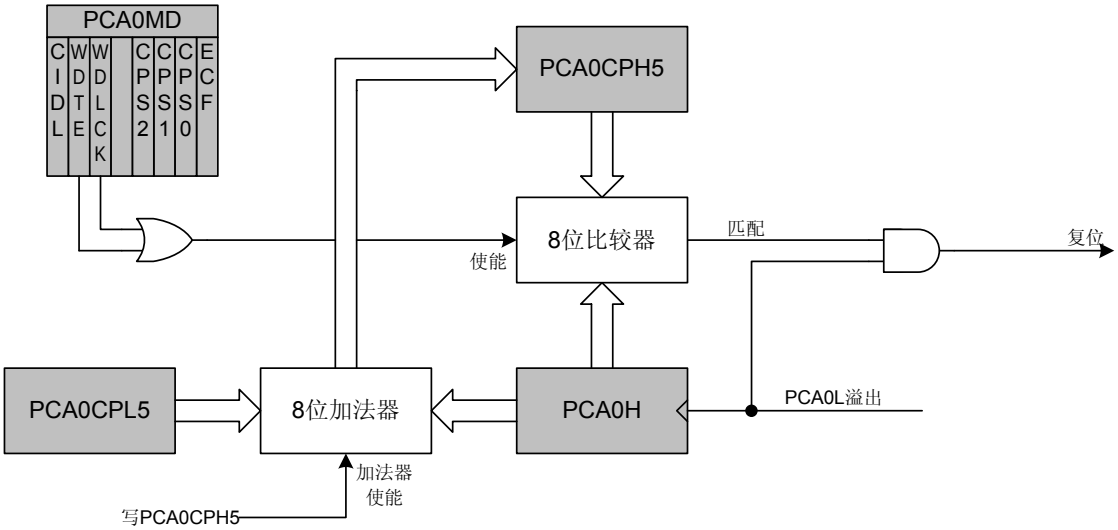


图 26.11　PCA 模块 5 的看门狗定时器方式

注意：保存在 PCA0CPH5 中的 8 位偏移值与 16 位 PCA 计数器的高字节进行比较，该偏移值是复位前 PCA0L 的溢出次数。PCA0L 的第一次溢出周期取决于进行更新操作时 PCA0L 的值，最长可达 256 个 PCA 时钟。总偏移值（PCA 时钟数）由方程 26.5 给出，其中 *PCA0L* 是执行更新操作时 PCA0L 寄存器的值。

$$偏移值 = (256 \times PCA0CPL5) + (256 - PCA0L)$$

方程26.5　看门狗定时器偏移值（PCA时钟数）

当 PCA0L 发生溢出并且 PCA0CPH5 和 PCA0H 匹配时，WDT 将产生一次复位。在 WDT 被使能的情况下，软件可以通过向 CCF5 标志（PCA0CN.5）写 1 来强制产生 WDT 复位。

26.4.2 看门狗定时器的使用

配置 WDT 的步骤如下：

- 通过向 WDTE 位写 0 来禁止 WDT。
- 选择 PCA 时钟源（用 CPS2-0 位）。
- 向 PCA0CPL5 装入所希望的 WDT 更新偏移值。
- 配置 PCA 的空闲方式位（如果希望在 CPU 处于空闲方式时 WDT 停止工作，则应将 CIDL 位置 1）。
- 通过向 WDTE 位写 1 来使能 WDT。
- 通过写 PCA0CPH5 来复位 WDT 定时器。

在 WDT 被使能时，不能改变 PCA 时钟源和空闲方式的设置值。通过向 PCA0MD 寄存器的 WDTE 或 WDLCK 位写 1 来使能 WDT。当 WDLCK 被置 1 时，在发生下一次系统复位之前将不能禁止 WDT。如果 WDCLK 未被置 1，清除 WDTE 位将禁止 WDT。

WDT 在任何一次系统复位之后都被使能。PCA0 计数器的默认时钟为系统时钟的十二分频（SYSCLK/12）。PCA0L 和 PCA0CPL5 的默认值均为 0x00。根据方程 26.5，这些值将产生 256 个 PCA 时钟周期或 3072 个系统时钟周期的 WDT 超时间隔。表 26.3 列出了对应某些典型系统时钟频率的超时间隔示例。

表 26.3 看门狗定时器超时间隔 [1]

系统时钟（Hz）	PCA0CPL2	超时间隔（ms）
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
3,062,500[2]	255	257
3,062,500[2]	128	129.5
3,062,500[2]	32	33.1
32,000	255	24576
32,000	128	12384
32,000	32	3168

注：
1．假设 PCA 使用 SYSCLK/12 作为时钟源，更新时 PCA0L 的值为 0x00。
2．内部 SYSCLK 复位频率 ＝ 内部振荡器频率/8。

26.5　PCA0 寄存器说明

下面对与 PCA 工作有关的特殊功能寄存器进行详细说明。

SFR 定义 26.1　PCA0CN：PCA 控制寄存器

SFR 页：0x00
SFR 地址：0xD8　　（可位寻址）

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　CF：PCA 计数器/定时器溢出标志
当 PCA 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 PCA 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 6：　CR：PCA 计数器/定时器运行控制
该位使能/禁止 PCA 计数器/定时器。
0：禁止 PCA 计数器/定时器。
1：使能 PCA 计数器/定时器。

位 5-0：　CCF[5:0]：PCA 模块 n 捕捉/比较标志
在 PCA 模块 n 发生一次匹配或捕捉时，相应的 CCFn 位由硬件置位。当 CCFn 中断被允许时，该位置 1 将导致 CPU 转向 PCA 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

SFR 定义 26.2　PCA0MD：PCA 方式寄存器

SFR 页：0x00
SFR 地址：0xD9

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	复位值
CIDL	WDTE	WDLCK	-	CPS2	CPS1	CPS0	ECF	01000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　CIDL：PCA 计数器/定时器空闲控制
设置 CPU 空闲方式下的 PCA 工作方式。
0：当系统控制器处于空闲方式时，PCA 继续正常工作。
1：当系统控制器处于空闲方式时，PCA 停止工作。

位 6：　WDTE：看门狗定时器使能位
如果该位被置 1，PCA 模块 5 被用作看门狗定时器。
0：看门狗定时器被禁止。
1：PCA 模块 5 被用作看门狗定时器。

位 5：　WDLCK：看门狗定时器锁定
该位对看门狗定时器使能位锁定/解锁。当 WDLCK 被置 1 时，在发生下一次系统复位之前将不能禁止 WDT。
0：看门狗定时器使能位未锁定。
1：看门狗定时器使能位被锁定。

位 4：　未用。读 ＝0b，写 ＝ 忽略。

位 3-1：　CPS2-CPS0：PCA 计数器/定时器时钟选择
这些位选择 PCA 计数器的时钟源。

CPS2	CPS1	CPS0	时钟源
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	外部时钟/8（与系统时钟同步）
1	1	0	smaRTClock 振荡器/8（与系统时钟同步）
1	1	1	保留

位 0：　ECF：PCA 计数器/定时器溢出中断允许
该位是 PCA 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）被置位时，允许 PCA 计数器/定时器溢出的中断请求。

注：当 WDTE 位被置 1 时，不能改变 PCA0MD 寄存器的其它位。若要改变 PCA0MD 的内容，必须先禁止看门狗定时器。

SFR 定义 26.3　PCA0PWM：PCA PWM 配置寄存器

SFR 页：0x00
SFR 地址：0xDF

R/W	R/W	R/W	R	R	R	R/W	R/W	复位值
ARSEL	ECOV	COVF	-	-	-	CLSEL1	CLSEL0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7：　ARSEL：自动重装载寄存器选择
该位选择是读和写正常的 PCA 捕捉/比较寄存器(PCA0CPn)还是位于相同 SFR 地址的自动重装载寄存器。该功能用于定义 9、10 或 11 位 PWM 方式的重载值。在所有其它方式，自动重装载寄存器没有功能。
0：读/写位于 PCA0CPHn 和 PCA0CPLn 的捕捉/比较寄存器。
1：读/写位于 PCA0CPHn 和 PCA0CPLn 的自动重装载寄存器。

位 6：　ECOV：周期溢出中断允许
该位设置周期溢出标志（COVF）中断的掩码。
0：COVF 不产生 PCA 中断。
1：当 COVF 置位时将产生一个 PCA 中断。

位 5：　COVF：周期溢出标志
该位指示主 PCA 计数器的第 8、9、10 或 11 位溢出。该标志具体使用哪一位取决于周期长度选择位的设置。该位可由硬件或软件置位，但必须用软件清 0。
0：自最后一次该位被清 0 以来没有发生溢出。
1：自最后一次该位被清 0 以来发生了溢出。

位 4-2：　未用。读 =000b，写 = 忽略。

位 1-0：　CLSEL[1:0]：周期长度选择位
当未选择 16 位 PWM 方式时，这些位选择 8、9、10 或 11 位的 PWM 周期长度。这种选择影响不使用 16 位 PWM 方式的所有被配置为 PWM 方式的通道。这些位对配置为 16 位 PWM 方式的通道而言被忽略。
00：8 位。
01：9 位。
10：10 位。
11：11 位。

SFR 定义 26.4　PCA0CPMn：PCA 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：　PCA0CPM0＝0xDA, PCA0CPM1＝0xDB, PCA0CPM2＝0xDC, PCA0CPM3＝0xDD, PCA0CPM4＝0xDE, PCA0CPM5＝0xCE

SFR 页：　0x00

位 7：　PWM16n：16 位脉冲宽度调制使能
当脉冲宽度调制方式被使能时（PWMn = 1），该位选择 16 位方式。
0：选择 8~11 位 PWM。
1：选择 16 位 PWM。

位 6：　ECOMn：比较器功能使能
该位置 1 时使能 PCA 模块 n 的比较器功能。

位 5：　CAPPn：正沿捕捉功能使能
该位置 1 时使能 PCA 模块 n 的正边沿捕捉。

位 4：　CAPNn：负沿捕捉功能使能
该位置 1 时使能 PCA 模块 n 的负边沿捕捉。

位 3：　MATn：匹配功能使能
该位置 1 时使能 PCA 模块 n 的匹配功能。如果被使能，当 PCA 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 位被置 1。

位 2：　TOGn：电平切换功能使能
该位置 1 时使能 PCA 模块 n 的电平切换功能。如果被使能，当 PCA 计数器与一个模块的捕捉/比较寄存器匹配时，CEXn 引脚的逻辑电平发生切换。如果 PWMn 位也被置 1，模块将工作在频率输出方式。

位 1：　PWMn：脉宽调制方式使能
该位置 1 时使能 PCA 模块 n 的 PWM 功能。当被使能时，CEXn 引脚输出脉冲宽度调制信号。PWM16n 为 0 时使用 8~11 位 PWM 方式；PWM16n 为 1 时使用 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。

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：允许 CCFn 位被置 1 时捕捉/比较标志的中断请求。

注：当 WDTE 位被置 1 时，不能修改 PCA0CPM5 寄存器，模块 5 作为看门狗定时器。若要改变 PCA0CPM2 寄存器的内容或模块 5 的功能，必须先禁止看门狗定时器。

SFR 定义 26.5　PCA0L：PCA 计数器 / 定时器低字节

SFR 页：0x00
SFR 地址：0xF9

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	

位 7-0：　PCA0[7:0]：PCA 计数器/定时器的低字节
PCA0L 寄存器保存 16 位 PCA 计数器/定时器的低字节（LSB）。

注：当 WDTE 位被置 1 时，不能用软件修改 PCA0L 寄存器。若要改变 PCA0L 寄存器的内容，必须先禁止看门狗定时器。

SFR 定义 26.6　PCA0H：PCA 计数器 / 定时器高字节

SFR 页：0x00
SFR 地址：0xFA

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	

位 7-0：　PCA0[15:8]：PCA 计数器/定时器高字节
PCA0H 寄存器保存 16 位 PCA 计数器/定时器的高字节（MSB）。读该寄存器时读的是“瞬像”寄存器的内容，是在读 PCA0L 的内容时被锁存的（见 26.1 节）。

注：当 WDTE 位被置 1 时，不能用软件修改 PCA0H 寄存器。若要改变 PCA0H 寄存器的内容，必须先禁止看门狗定时器。

SFR 定义 26.7　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
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：　PCA0CPL0 = 0xFB, PCA0CPL1 = 0xE9, PCA0CPL2 = 0xEB, PCA0CPL3 = 0xED, PCA0CPL4 = 0xFD, PCA0CPL5 = 0xD2

SFR 页：　0x00

位 7-0：　PCA0CPn[7:0]：PCA 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）。该寄存器还允许访问对应的 PCA 通道之 9、10 或 11 位 PWM 方式自动重载值的低字节。寄存器 PCA0PWM 中的 ARSEL 位控制访问哪个寄存器。

注：写该寄存器时将模块的 ECOMn 位清 0。

SFR 定义 26.8 PCA0CPHn：PCA 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
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：　PCA0CPH0=0xFC, PCA0CPH1=0xEA, PCA0CPH2=0xEC, PCA0CPH3=0xEE, PCA0CPH4=0xFE, PCA0CPH5=0xD3

SFR 页：　0x00

位 7-0：　PCA0CPn[15:8]：PCA 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）。该寄存器还允许访问对应的 PCA 通道之 9、10 或 11 位 PWM 方式自动重载值的低字节。寄存器 PCA0PWM 中的 ARSEL 位控制访问哪个寄存器。

注：写该寄存器时将模块的 ECOMn 位置 1。

27. C2 接口

C8051F93x-C8051F92x 器件有一个 Silicon Labs 2 线（C2）调试接口，支持 FLASH 编程和使用安装在最终应用系统中的器件进行在系统调试。C2 接口使用一个时钟信号（C2CK）和一个双向的 C2 数据信号（C2D）在器件和宿主机之间传送信息。有关 C2 协议的详细信息见 C2 接口规范。

27.1　C2 接口寄存器

下面对通过 C2 接口进行 FLASH 编程所必须的 C2 寄存器进行说明。对所有 C2 寄存器的访问都要通过 C2 接口实现（详见 C2 接口规范）。

C2 寄存器定义 27.1　C2ADD：C2 地址寄存器

位7	位6	位5	位4	位3	位2	位1	位0	复位值
								00000000

位 7-0：　C2ADD[7:0]：C2 地址寄存器
C2ADD 寄存器选择 C2 数据读和数据写命令的目标数据寄存器。

地址	说明
0x00	选择器件 ID 寄存器（数据读指令）。
0x01	选择版本 ID 寄存器（数据读指令）。
0x02	选择 C2 FLASH 编程控制寄存器（数据读/写指令）
0xB4	选择 C2 FLASH 编程数据寄存器（数据读/写指令）

C2 SFR 定义 27.2　DEVICEID：C2 器件 ID 寄存器

位7	位6	位5	位4	位3	位2	位1	位0	复位值	C2地址：
								00010110	0x00

位 7-0：　DEVICEID[7:0]：器件 ID 寄存器
该只读寄存器返回 8 位的器件 ID 号：0x16（C8051F93x-92x）。

C2 SFR 定义 27.3　REVID：C2 版本 ID 寄存器

位7	位6	位5	位4	位3	位2	位1	位0	复位值	C2地址：
								可变	0x01

位 7-0：　REVID[7:0]：版本 ID 寄存器
该只读寄存器返回 8 位的版本 ID 号：如 0x00（版本 A）。

C2 SFR 定义 27.4　FPCTL：C2 FLASH 编程控制寄存器

								复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	C2地址：0x02

位 7-0：　FPCTL[7:0]：C2 FLASH 编程控制寄存器。

该寄存器用于使能通过 C2 接口对 FLASH 编程。为了使能 C2 FLASH 编程，必须按顺序写代码：0x02、0x01。注意：一旦 C2 FLASH 编程被使能，必须进行一次复位才能使系统恢复正常工作。

C2 SFR 定义 27.5　FPDAT：C2 FLASH 编程数据寄存器

								复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	C2地址：0xB4

位 7-0：　FPDAT[7:0]：FLASH 编程数据寄存器。

该寄存器用于在 C2 FLASH 访问期间传递 FLASH 编程命令、地址和数据。下面列出了有效的编程命令。

代码	命令
0x06	读 FLASH 块
0x07	写 FLASH 块
0x08	擦除 FLASH 页
0x03	擦除器件

27.2　C2 引脚共享

C2 协议允许 C2 引脚与用户功能共享，可以进行在系统调试和 FLASH 编程。这种共享之所以可能，是因为 C2 通信通常发生在器件的停止运行状态。在这种状态下所有片内外设和用户软件都被冻结，C2 接口可以安全地“借用”C2CK（/RST）和 C2D 引脚。在大多数应用中，需要使用外部电阻对 C2 接口和用户应用进行隔离。典型的隔离电路如图 27.1 所示。

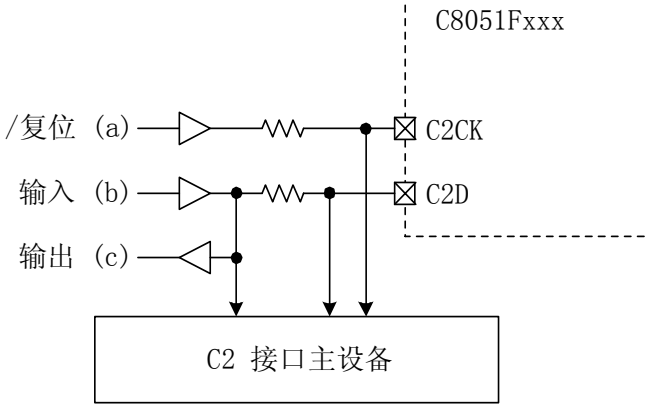


图 27.1 典型 C2 引脚共享电路

在图 27.1 的配置中，我们假设：

1．在目标器件的停止运行状态，用户输入（b）不能改变。

2．目标器件的/RST 引脚只能被作为输入使用。

对于具体的应用，可能还需要加入一些电阻。

联系信息

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78735
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032
Email: mcuinfo@silabs.com
Internet: www.silabs.com

新华龙电子有限公司

电话：0755-83645240　83645242　83645244　83645251
技术支持：0755-83645259
传真：0755-83645243
地址：深圳市福田区华强北路现代之窗大厦 A 座 13F C 室(518013)
Email：sales@xhl.com.cn
Email：shenzhen@xhl.com.cn
网站：　www.xhl.com.cn
技术支持：support-sz@xhl.com.cn