

C8051F320/1
混合信号 ISP FLASH 微控制器
数 据 手 册

潘 琢 金 译

Rev 1.1 2004.12

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：024-86802267，13066535936

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

C8051F320/1 全速 USB ISP FLASH 微控制器

模拟外设

- 10 位 ADC
 - 转换速率可达 200ksps
 - 可多达 17 或 13 个外部单端或差分输入
 - VREF 可在内部 VREF、外部引脚或 VDD 中选择
 - 外部转换启动输入
 - 内置温度传感器
- 两个比较器
- 内部电压基准
- 上电复位/掉电检测器

USB 控制器

- 符合 USB 规范 2.0 版
- 全速(12Mbps)或低速(1.5Mbps)
- 集成时钟恢复电路；无需外部晶体
- 支持 8 个端点
- 1KB USB 缓存
- 集成收发器；无需外部电阻

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
- 比使用仿真芯片、目标仿真头和仿真插座的仿真系统有更优越的性能

电压调整器输入：4.0V - 5.25V

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

- 2304 B 内部数据 RAM(1K+256+1K USB FIFO)
- 16KB FLASH；可在系统编程，扇区大小为 512 字节

数字外设

- 25/21 个端口 I/O；均耐 5V 电压，大灌电流
- 硬件增强型 SPI、增强型 UART 和 SMBus 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 5 个捕捉/比较模块
- 使用外部时钟源和 PCA 或定时器的实时时钟方式

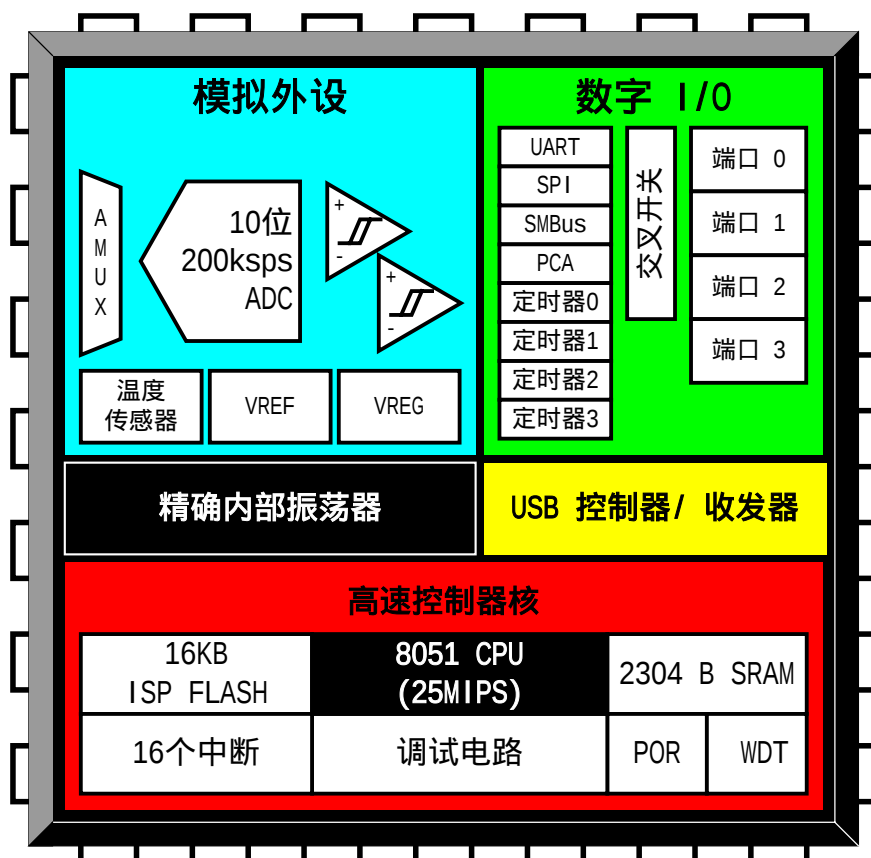
时钟源

- 内部振荡器：0.25%的精度(时钟恢复被使能时)。支持 USB 和 UART 操作
- 外部振荡器：晶体、RC、C、或外部时钟
- 可在运行中切换时钟源，适用于节电方式

封装

- 32 脚 LQFP（C8051F320）
- 28 脚 MLP（C8051F321）

温度范围：-40°C - +85°C



C8051F320/1 全速 USB ISP FLASH 微控制器

1. 系统概述	1
1.1 CIP-51™ 微控制器核	4
1.1.1 与 8051 完全兼容	4
1.1.2 速度提高	4
1.1.3 增加的功能	5
1.2 片内存储器	6
1.3 通用串行总线控制器	7
1.4 稳压器	7
1.5 片内调试电路	8
1.6 可编程数字 I/O 和交叉开关	9
1.7 串行端口	10
1.8 可编程计数器阵列	10
1.9 10 位模/数转换器	11
1.10 比较器	12
2. 极限参数	13
3. 总体直流电气特性	14
4. 引脚和封装定义	15
5. 10 位 ADC (ADC0)	20
5.1 模拟多路选择器	21
5.2 温度传感器	22
5.3 工作方式	23
5.3.1 转换启动方式	23
5.3.2 跟踪方式	24
5.3.3 建立时间要求	25
5.4 可编程窗口检测器	30
5.4.1 单端方式下的窗口检测器	31
5.4.2 差分方式下的窗口检测器	32
6. 电压基准	35
7. 比较器	37
8. 稳压器 (REG0)	47
8.1 稳压器方式选择	47
8.2 VBUS 检测	47

C8051F320/1 全速 USB ISP FLASH 微控制器

9. CIP-51 微控制器	51
9.1 指令集	52
9.1.1 指令和CPU时序	52
9.1.2 MOVX指令和程序存储器	52
9.2 存储器组织	56
9.2.1 程序存储器	56
9.2.2 数据存储器	57
9.2.3 通用寄存器	57
9.2.4 位寻址空间	57
9.2.5 堆栈	57
9.2.6 特殊功能寄存器	58
9.2.7 寄存器说明	62
9.3 中断系统	65
9.3.1 MCU中断源和中断向量	65
9.3.2 外部中断	65
9.3.3 中断优先级	66
9.3.4 中断响应时间	66
9.3.5 中断寄存器说明	68
9.4 电源管理方式	74
9.4.1 空闲方式	74
9.4.2 停机方式	74
10. 复位源	76
10.1 上电复位	77
10.2 掉电复位和VDD监视器	78
10.3 外部复位	79
10.4 时钟丢失检测器复位	79
10.5 比较器 0 复位	79
10.6 PCA看门狗定时器复位	79
10.7 FLASH错误复位	79
10.8 软件复位	80
10.9 USB复位	80
11. FLASH存储器	83
11.1 FLASH存储器编程	83
11.1.1 FLASH锁定和关键码功能	83
11.1.2 FLASH擦除	83
11.1.3 FLASH写	84
11.2 非易失性数据存储	84
11.3 安全选项	85

C8051F320/1 全速 USB ISP FLASH 微控制器

12 . 外部RAM.....	88
12.1 访问用户XRAM.....	88
12.2 访问USB FIFO	89
13. 振荡器	91
13.1 可编程内部振荡器.....	92
13.1.1 C8051F320/1 的内部振荡器编程.....	92
13.1.2 内部振荡器挂起方式.....	93
13.2 外部振荡器驱动电路.....	94
13.2.1 外部振荡器作为定时器的时钟.....	94
13.2.2 外部晶体示例.....	94
13.2.3 外部RC示例.....	95
13.2.4 外部电容示例.....	95
13.3 4 倍时钟乘法器.....	97
13.4 系统时钟和USB时钟选择.....	98
13.4.1 系统时钟选择.....	98
13.4.2 USB时钟选择.....	98
14. 端口输入/输出	101
14.1 优先级交叉开关译码器.....	102
14.2 端口I/O初始化	105
14.3 通用端口I/O	108
15. 通用串行总线控制器（USB0）	114
15.1 端点地址	115
15.2 USB收发器.....	115
15.3 USB寄存器访问.....	117
15.4 USB时钟配置.....	121
15.5 FIFO管理.....	122
15.5.1 FIFO分割方式.....	122
15.5.2 FIFO双缓冲.....	123
15.5.3 FIFO访问.....	123
15.6 功能地址	124
15.7 功能配置和控制.....	124
15.8 中断	128
15.9 串行接口引擎	132
15.10 端点 0	132
15.10.1 端点 0 SETUP 传输.....	133
15.10.2 端点 0 IN 传输.....	133
15.10.3 端点 0 OUT 传输.....	134
15.11 配置端点 1~3	136

C8051F320/1 全速 USB ISP FLASH 微控制器

15.12 控制端点 1~3 输入 (IN)	136
15.12.1 端点 1~3 输入 (IN) 中断或批量方式.....	136
15.12.2 端点 1~3 输入 (IN) 同步方式.....	137
15.13 控制端点 1~3 输出 (OUT)	140
15.13.1 端点 1~3 输出 (OUT) 中断或批量方式.....	140
15.13.2 端点 1~3 输出 (OUT) 同步方式.....	140
16. SMBUS.....	144
16.1 支持文档	145
16.2 SMBUS配置	145
16.3 SMBUS操作	146
16.3.1 总线仲裁	146
16.3.2 时钟低电平扩展.....	147
16.3.3 SCL 低电平超时.....	147
16.3.4 SCL 高电平 (SMBus 空闲) 超时.....	147
16.4 SMBUS的使用	147
16.4.1 SMBus 配置寄存器.....	148
16.4.2 SMBus 控制寄存器.....	151
16.4.3 数据寄存器.....	154
16.5 SMBUS传输方式	155
16.5.1 主发送器方式.....	155
16.5.2 主接收器方式.....	156
16.5.3 从接收器方式.....	157
16.5.4 从发送器方式.....	158
16.6 SMBUS状态译码	159
17. UART0	161
17.1 增强的波特率发生器.....	162
17.2 工作方式	163
17.2.1 8 位UART.....	163
17.2.2 9 位UART.....	164
17.3 多机通信	165
18. 增强型串行外设接口 (SPI0)	170
18.1 信号说明	171
18.1.1 主输出、从输入 (MOSI)	171
18.1.2 主输入、从输出 (MISO)	171
18.1.3 串行时钟 (SCK)	171
18.1.4 从选择 (NSS)	171
18.2 SPI0 主方式.....	173
18.3 SPI0 从方式.....	174

C8051F320/1 全速 USB ISP FLASH 微控制器

18.4 SPI0 中断源.....	174
18.5 串行时钟时序	175
18.6 SPI特殊功能寄存器.....	177
19. 定时器	183
19.1 定时器 0 和定时器 1.....	183
19.1.1 方式 0	183
19.1.2 方式 1	185
19.1.3 方式 2	185
19.1.4 方式 3	186
19.2 定时器 2	191
19.2.1 16 位自动重载方式.....	191
19.2.2 8 位自动重载定时器方式.....	192
19.2.3 USB 帧起始捕捉方式.....	193
19.3 定时器 3	197
19.3.1 16 位自动重载方式.....	197
19.3.2 8 位自动重载定时器方式.....	198
19.3.3 USB 帧起始捕捉方式.....	199
20. 可编程计数器阵列 (PCA0)	203
20.1 PCA计数器/定时器.....	204
20.2 捕捉/比较模块	206
20.2.1 边沿触发的捕捉方式.....	207
20.2.2 软件定时器方式.....	208
20.2.3 高速输出方式.....	209
20.2.4 频率输出方式.....	210
20.2.5 8 位脉宽调制器方式.....	211
20.2.6 16 位脉宽调制器方式.....	212
20.3 看门狗定时器方式.....	213
20.3.1 看门狗定时器操作.....	213
20.3.2 看门狗定时器的使用.....	214
20.4 PCA寄存器说明.....	215
21. C2 接口.....	219
21.1 C2 接口寄存器	219
21.2 C2 引脚共享	221

C8051F320/1 全速 USB ISP FLASH 微控制器

1. 系统概述

C8051F320/1 器件是完全集成的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 1.1。

- 高速、流水线结构的 8051 兼容的微控制器内核（可达 25MIPS）
- 全速、非侵入式的在系统调试接口（片内）
- 通用串行总线（USB）功能控制器，有 8 个灵活的端点管道，集成收发器和 1K FIFO RAM
- 电源稳压器（5V 至 3V）
- 真正 10 位 200 ksp/s 的 17 通道单端/差分 ADC，带模拟多路器
- 片内电压基准和温度传感器
- 片内电压比较器（两个）
- 高精度可编程的 12MHz 内部振荡器和 4 倍时钟乘法器
- 16KB 可在系统编程的 FLASH 存储器
- 2304 字节片内 RAM（256+1K+1K USB FIFO）
- 硬件实现的 SMBus/I²C、增强型 UART 和增强型 SPI 串行接口
- 4 个通用的 16 位定时器
- 具有 5 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）
- 片内上电复位、VDD 监视器和时钟丢失检测器
- 25/21 个端口 I/O（容许 5V 输入）

具有片内上电复位、VDD 监视器、电压调整器、看门狗定时器和时钟振荡器的 C8051F320/1 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可全功能运行。两个 C2 接口引脚可以与用户功能共享，使在系统调试功能不占用封装引脚。

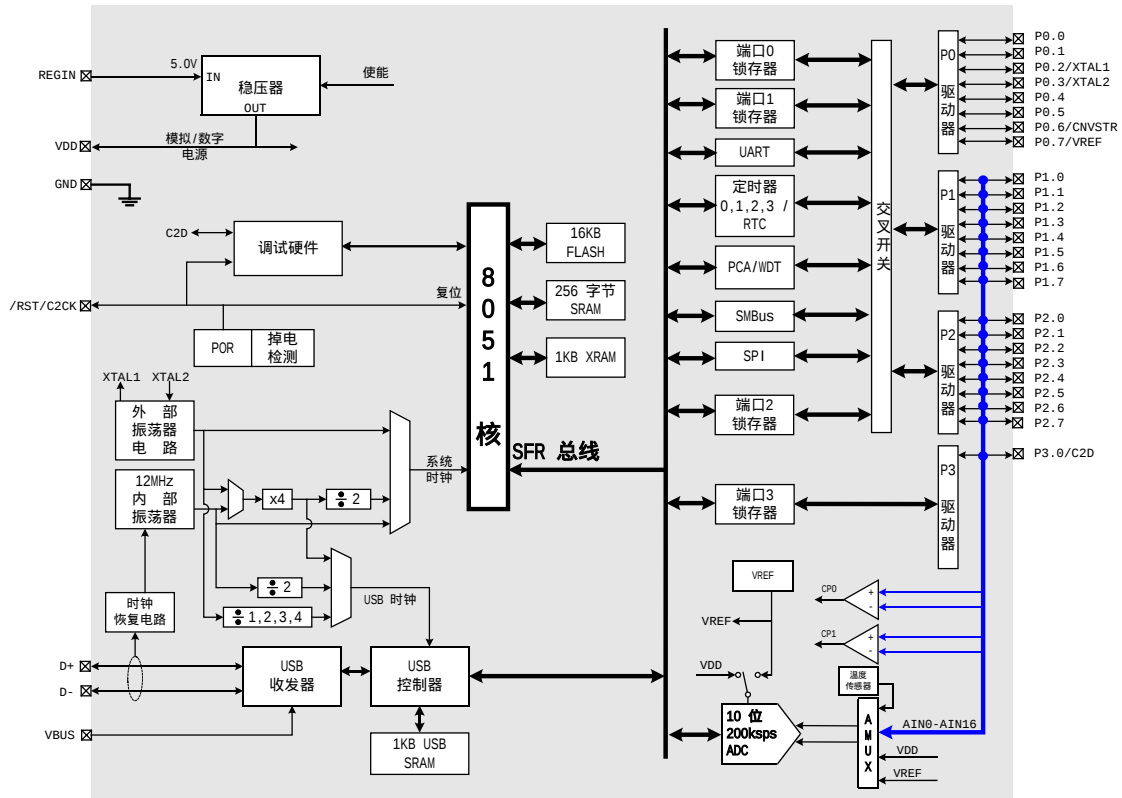
每种器件都可在工业温度范围（-45 到+85）内用 2.7V-3.6V 的电压工作。端口 I/O 和 /RST 引脚都容许 5V 的输入信号电压。C8051F320/1 采用 32 脚 LQFP 封装或 28 脚 MLP 封装。

表 1.1 产品选择指南

	MIPS（峰值）	FLASH 存储器	RAM	校准的内部 24.5MHz 振荡器	USB	电源电压调整器	SMBus/I ² C	增强型 SPI	UART	定时器(16 位)	可编程计数器阵列	数字端口 I/O	10 位 200ksp/s ADC	温度传感器	电压基准	模拟比较器	封装
C8051F320	25	16K	2304	√	√	√	√	√	√	4	√	25	√	√	√	2	LQFP-32
C8051F321	25	16K	2304	√	√	√	√	√	√	4	√	21	√	√	√	2	MLP-28

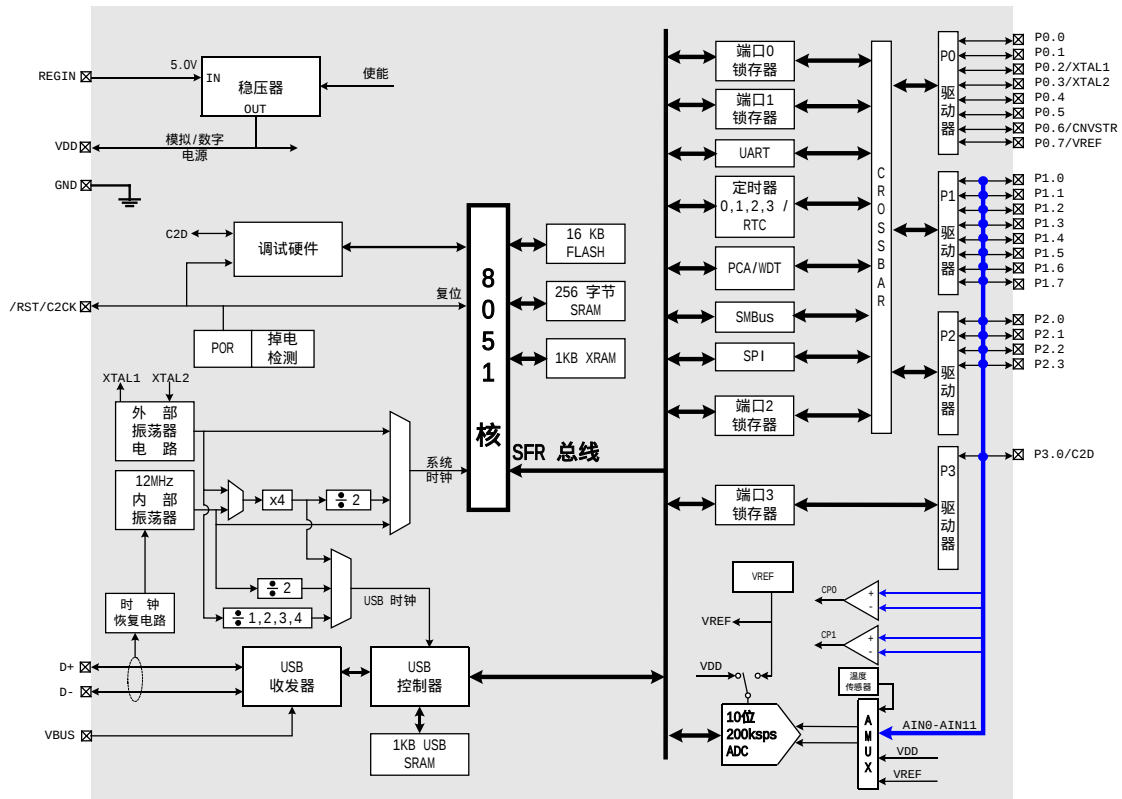
C8051F320/1 全速 USB ISP FLASH 微控制器

图 1.1 C8051F320 原理框图



C8051F320/1 全速 USB ISP FLASH 微控制器

图 1.2 C8051F321 原理框图



C8051F320/1 全速 USB ISP FLASH 微控制器

1.1 CIP-51™ 微控制器核

1.1.1 与 8051 完全兼容

C8051F320/1 系列器件使用Silicon Labs的专利CIP-51 微控制器内核。CIP-51 与MCS-51™ 指令集完全兼容，可以使用标准 803x/805x的汇编器和编译器进行软件开发。CIP-51 内核具有标准 8052 的所有外设部件，包括 4 个 16 位计数器/定时器、一个具有增强波特率配置的全双工UART、一个增强型SPI端口、2304 字节内部RAM、128 字节特殊功能寄存器（SFR）地址空间及 25/21 个I/O引脚。

1.1.2 速度提高

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

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

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

CIP-51 工作在最大系统时钟频率 25MHz 时，它的峰值速度达到 25MIPS。图 1.3 给出了几种 8 位微控制器内核工作在最大系统时钟时峰值速度的比较关系。

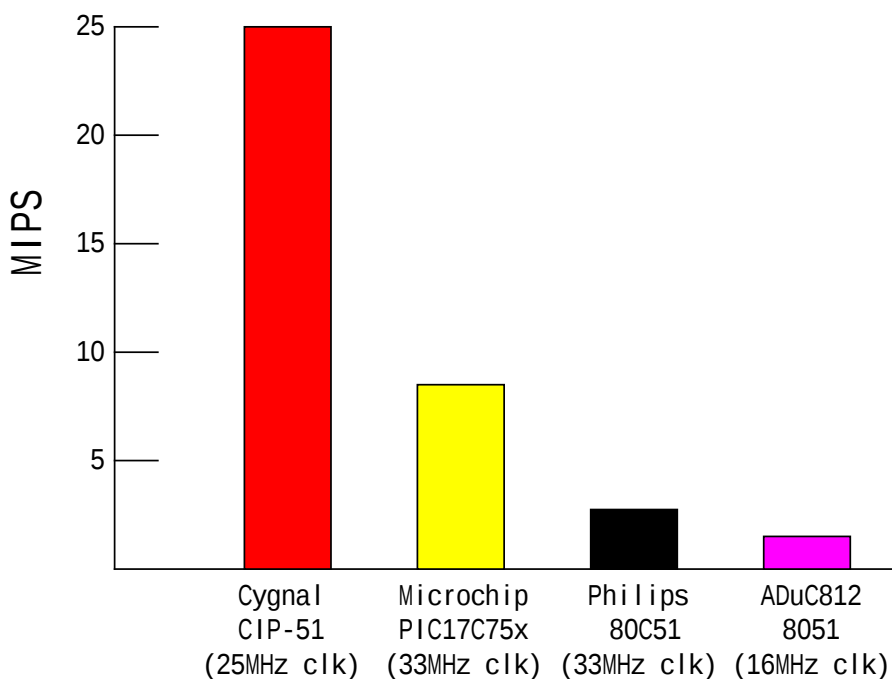


图 1.3 MCU 峰值执行速度比较

C8051F320/1 全速 USB ISP FLASH 微控制器

1.1.3 增加的功能

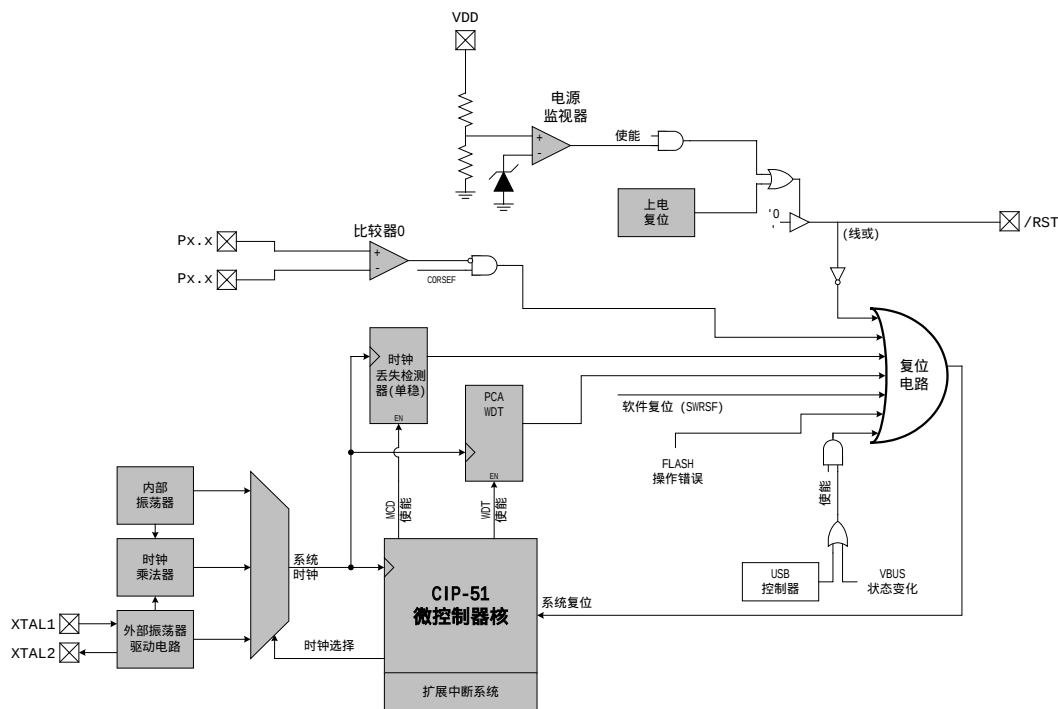
C8051F320/1 系列 MCU 在 CIP-51 内核和外设方面有几项关键性的改进，提高了整体性能，更易于在最终应用中使用。

扩展的中断系统向 CIP-51 提供 16 个中断源(标准 8051 只有 7 个中断源)，允许大量的模拟和数字外设中断微控制器。一个中断驱动的系统需要较少的 MCU 干预，因而有更高的执行效率。在设计一个多任务实时系统时，这些增加的中断源是非常有用的。

C8051F320/1 有多达 9 个复位源：上电复位电路 (POR)、片内 VDD 监视器 (当电源电压低于 V_{RST} 时强制复位)、USB 控制器 (USB 总线复位或 VBUS 状态变化)、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、外部复位输入引脚和 FLASH 读/写错误保护电路复位。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 可以被永久性使能。

C8051F320/1 器件的内部振荡器在出厂时已经被校准为 $12\text{MHz} \pm 1.5\%$ ，该振荡器的周期可以由用户以大约 0.25% 的增量编程。时钟恢复电路允许内部振荡器与 4 倍时钟乘法器配合，提供全速方式 USB 时钟源。内部振荡器还被用作低速方式下的 USB 时钟源。外部振荡器也可以与 4 倍时钟乘法器配合使用。器件内集成了外部振荡器驱动电路，允许使用晶体、陶瓷谐振器、电容、RC 或外部 CMOS 时钟源产生系统时钟。系统时钟可以被配置为使用内部振荡器、外部振荡器或时钟乘法器输出二分频。如果需要，可以在 CPU 运行时切换振荡源。外部振荡器在低功耗系统中是非常有用的，它允许 MCU 从一个低频率 (节电) 外部时钟源运行，当需要时再周期性地切换到高速的内部振荡器。

图 1.4 片内时钟和复位电路



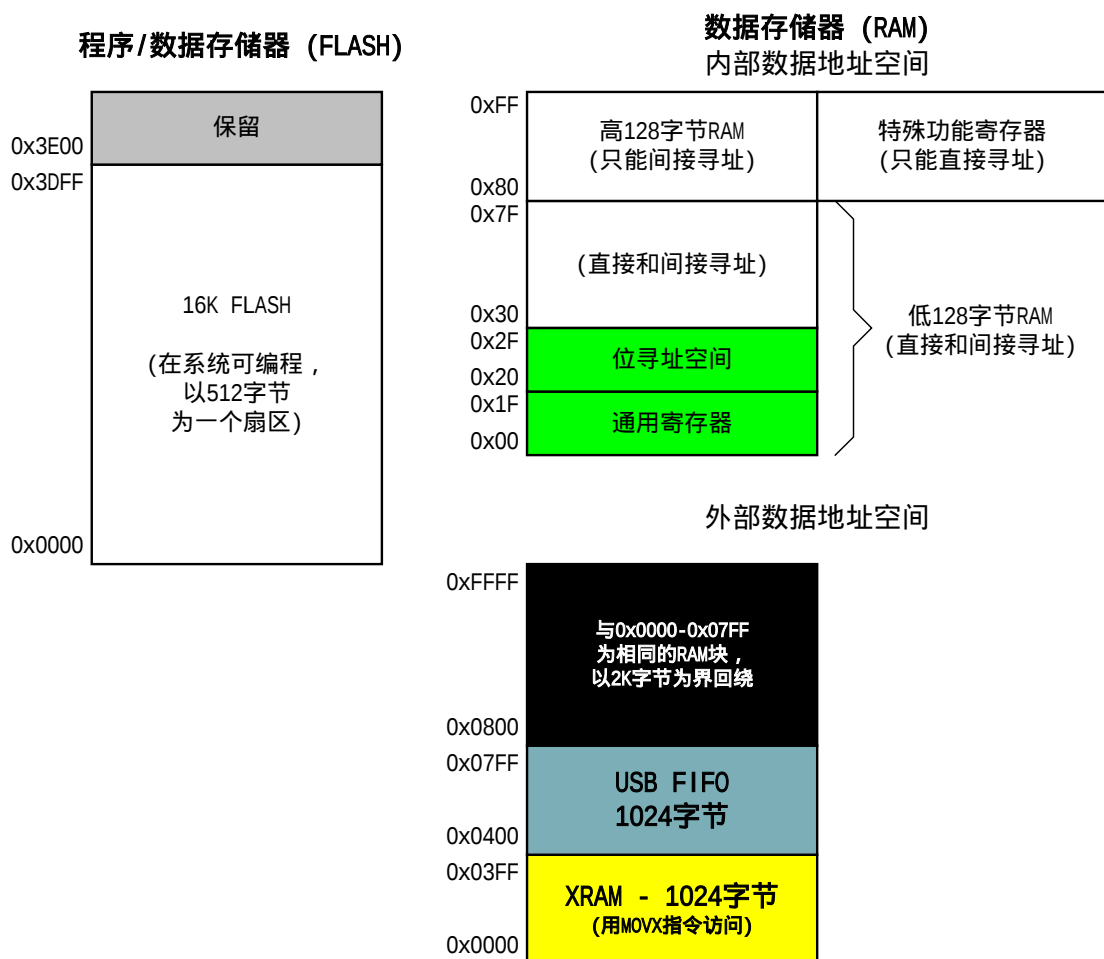
C8051F320/1 全速 USB ISP FLASH 微控制器

1.2 片内存储器

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

程序存储器包含 16KB 的 FLASH。该存储器以 512 字节为一个扇区，可以在系统编程，且不需特别的编程电压。图 1.5 给出了 MCU 系统的存储器结构。

图 1.5 片内存储器组织



C8051F320/1 全速 USB ISP FLASH 微控制器

1.3 通用串行总线控制器

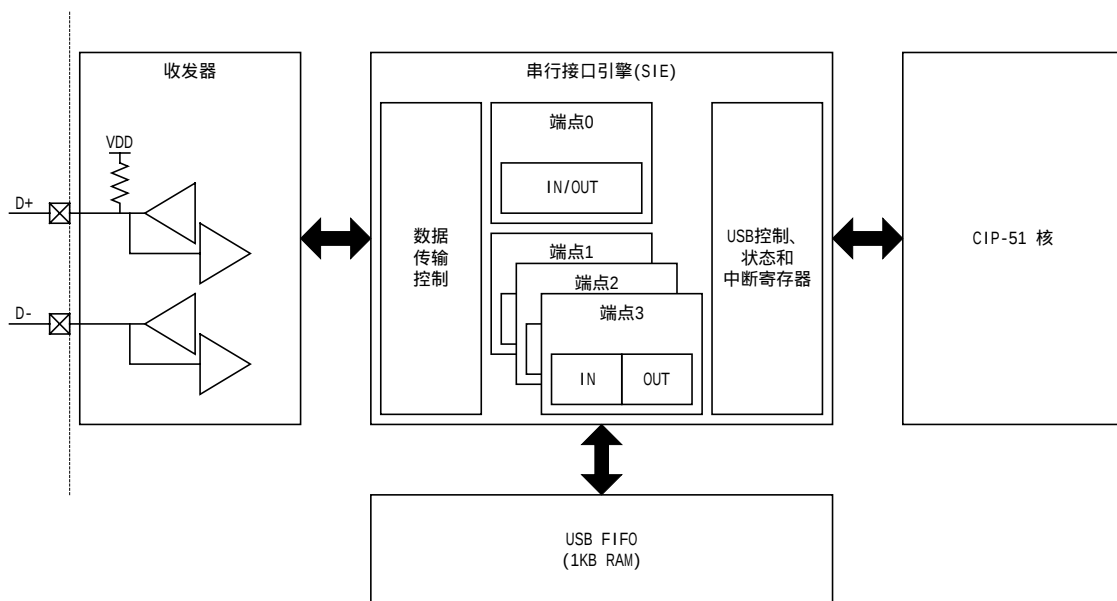
通用串行总线控制器（USB0）符合USB2.0规范，可以全速或低速工作，集成了收发器和端点FIFO RAM。共有8个端点：一个双向控制端点（端点0）和三对输入/输出端点（端点1-3 输入/输出）。

XRAM中有1KB的存储块被专门用作USB FIFO空间。该FIFO空间被分配给端点0-3；端点1-3 的FIFO可以被配置为输入（IN）、输出（OUT）或输入/输出（分割模式）。最大的FIFO大小为512字节（端点3）。

USB0可以工作在全速或低速方式。片内4倍时钟乘法器和时钟恢复电路允许使用内部高精度振荡器作为USB时钟源，实现全速和低速通信。外部振荡器也可以与4倍时钟乘法器配合使用来产生USB时钟。CPU时钟源与USB时钟相互独立。

USB收发器符合USB2.0规范，并包含内部匹配和上拉电阻。上拉电阻可以被用软件使能/禁止，可以根据软件选择的速度设置（全速或低速）出现在D+或D-引脚。

图1.6 USB控制器原理框图



1.4 稳压器

C8051F320/1器件内部包含一个5V至3V的稳压器（REG0）。当被使能时，REG0输出连到VDD引脚，并可为外部器件提供电源。REG0可以被软件使能/禁止。

C8051F320/1 全速 USB ISP FLASH 微控制器

1.5 片内调试电路

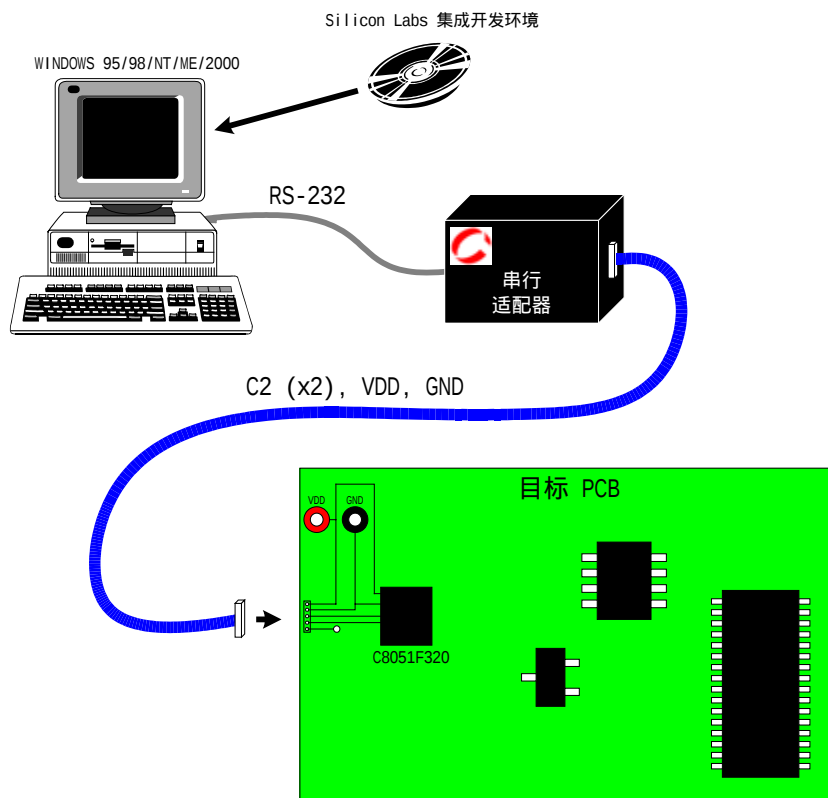
C8051F320/1器件具有片内Silicon Labs 2线（C2）接口调试电路，支持使用安装在最终应用系统中的产品器件进行非侵入式、全速的在系统调试。

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

开发套件C8051F320DK具有开发应用代码和对C8051F320/1 MCU进行在系统调试所需要的全部硬件和软件。开发套件中包括开发者工作室软件和调试器、一个集成的8051汇编器和一个RS-232转换到C2的串行适配器。套件中还有一个目标应用板，上面有对应的MCU和样机区。套件中还包括RS-232和C2电缆及一个墙装电源。开发套件需要一个运行Windows 95/98/Me/2000并有一个可用RS-232串口的计算机。如图1.7所示，PC机通过RS-232与串行适配器连接。一条六英寸的扁平电缆将串行适配器和用户的应用板连接起来，使用2个C2引脚和VDD及GND。串行适配器从应用板获取其电源。对于不能从目标板上提取足够电源的应用，可以将套件中提供的电源直接连到串行适配器上。

对于开发和调试来说，Silicon Labs IDE接口比采用标准MCU仿真器要优越得多。标准的MCU仿真器要使用在板仿真芯片和目标电缆，还需要在应用板上有MCU的插座。Silicon Labs的调试环境既便于使用又能保证精确模拟外设的性能。

图1.7 开发/在系统调试示意图



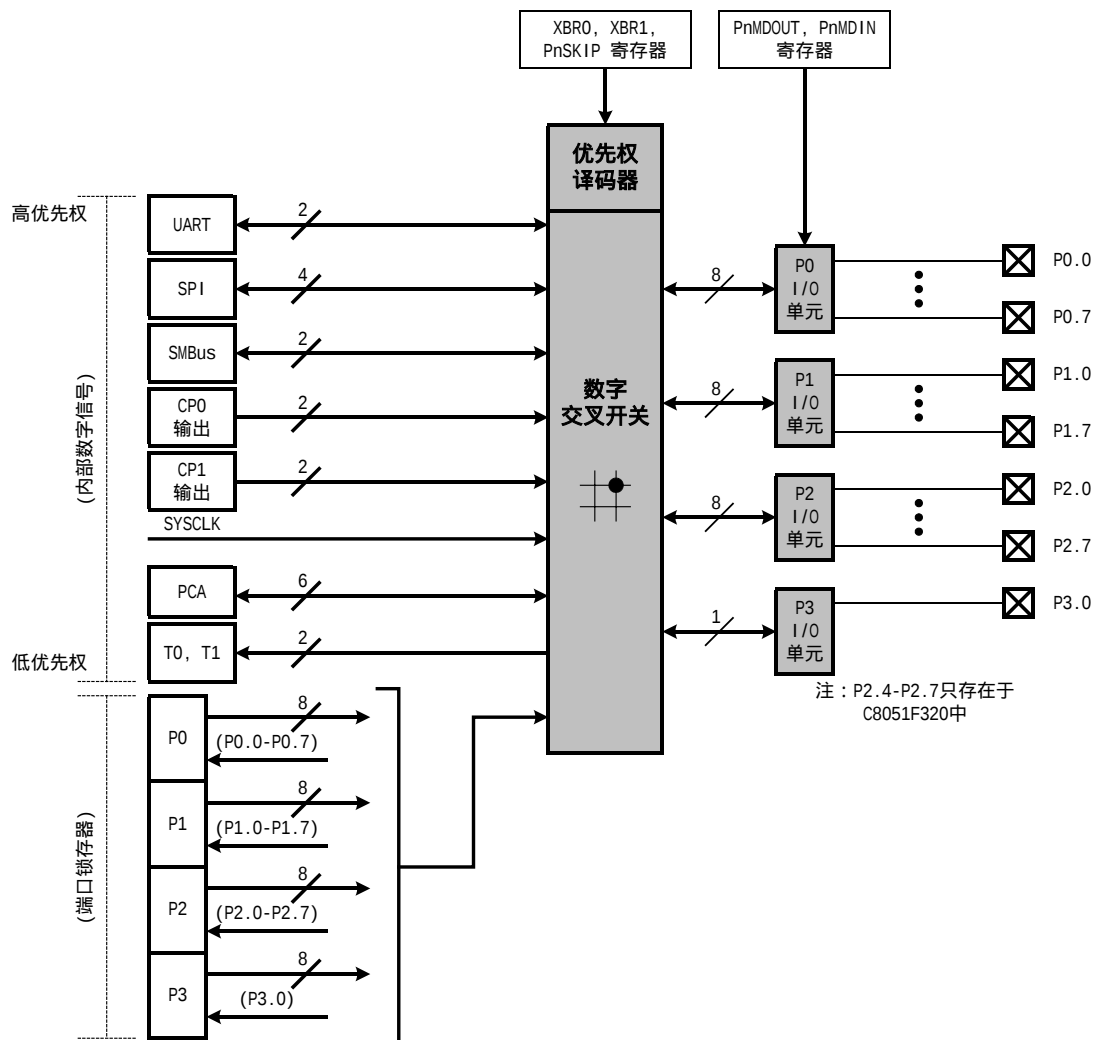
C8051F320/1 全速 USB ISP FLASH 微控制器

1.6 可编程数字 I/O 和交叉开关

C8051F320有25个I/O引脚（三个8位口和一个1位口）；C8051F321有21个I/O引脚（两个8位口、一个4位口和一个1位口）。C8051F320/1端口的工作情况与标准8051相似，但有一些改进。每个端口引脚都可以被配置为模拟输入或数字I/O引脚。被选择作为数字I/O的引脚还可以被配置为推挽或漏极开路输出。在标准8051中固定的“弱上拉”可以被总体禁止，这为低功耗应用提供了进一步节电的能力。

数字交叉开关允许将内部数字系统资源映射到端口I/O引脚（见图1.8）。可通过设置交叉开关控制寄存器将片内的计数器/定时器、串行总线、硬件中断、比较器输出以及微控制器内部的其它数字信号配置为出现在端口I/O引脚。这一特性允许用户根据自己的特定应用选择通用端口I/O和所需数字资源的组合。

图1.8 数字交叉开关原理框图



C8051F320/1 全速 USB ISP FLASH 微控制器

1.7 串行端口

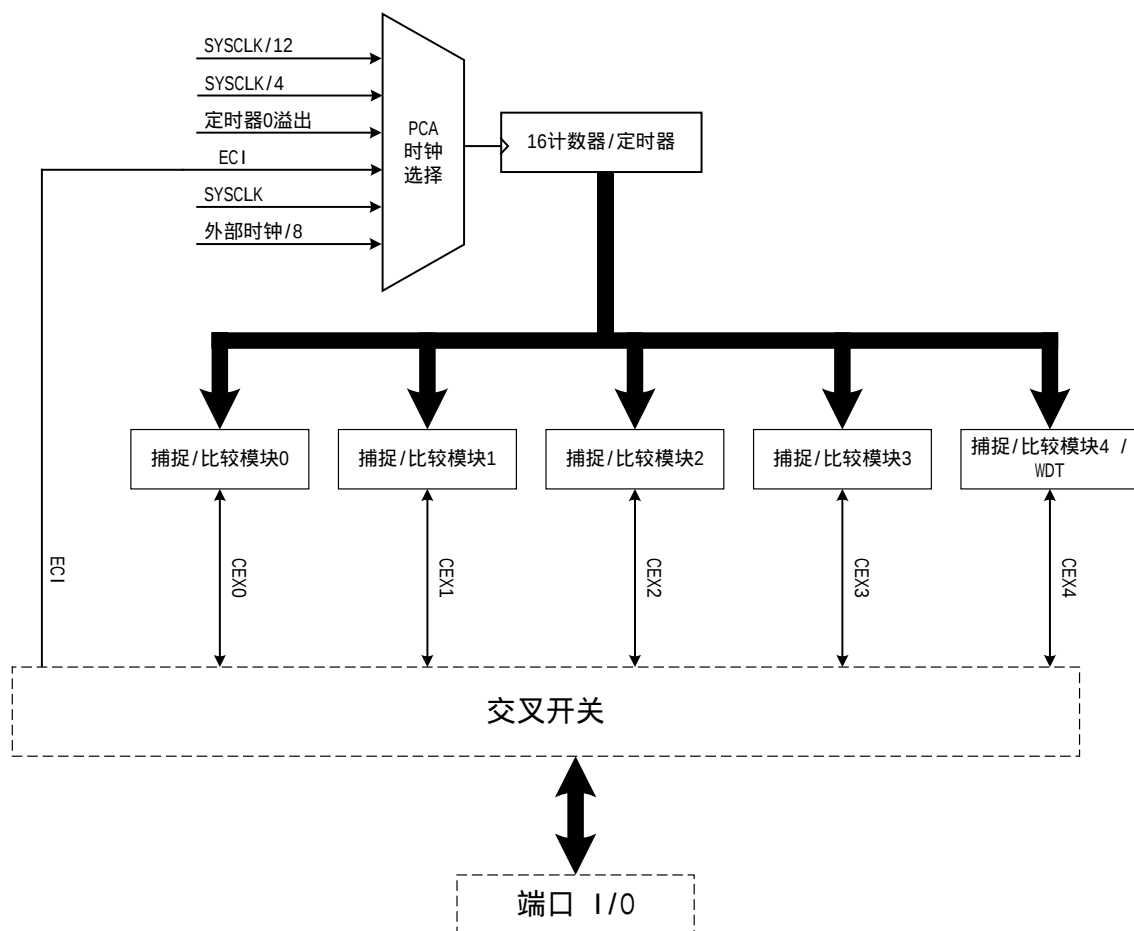
C8051F320/1系列MCU内部有一个SMBus/I²C接口、一个具有增强型波特率配置的全双工UART和一个增强型SPI接口。每种串行总线都完全用硬件实现，都能向CIP-51产生中断，因此需要很少的CPU干预。

1.8 可编程计数器阵列

除了4个16位的通用计数器/定时器之外，MCU中还有一个片内可编程计数器/定时器阵列（PCA）。PCA包括一个专用的16位计数器/定时器时间基准和5个可编程的捕捉/比较模块。PCA的时钟可以是下面的六个时钟源之一：系统时钟/12、系统时钟/4、定时器0溢出、外部时钟输入（ECI）、系统时钟和外部振荡源频率/8。外部时钟源对于实时时钟功能是非常有用的，可以在使用内部振荡器驱动系统时钟的同时由外部振荡器给PCA提供时钟。

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、8位或16位脉冲宽度调制器、频率输出。此外，捕捉/比较模块4还提供看门狗定时器（WDT）功能。在系统复位后，捕捉/比较模块4被配置并被使能作为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1.9 PCA原理框图



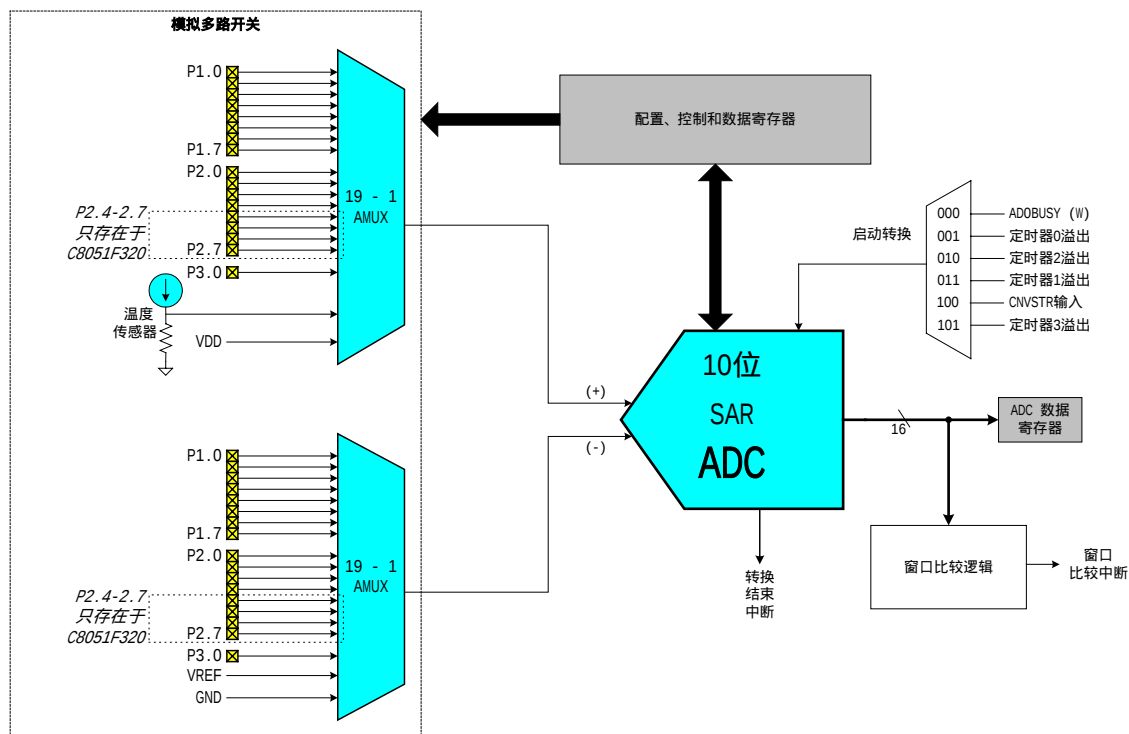
1.9 10 位模/数转换器

C8051F320/1内部有一个10位SAR ADC和一个17通道差分输入多路选择器。该ADC工作在200ksps的最大采样速率时可提供真正10位的线性度，INL为 $\pm 1\text{LSB}$ 。ADC系统包含一个可编程的模拟多路选择器，用于选择ADC的正输入和负输入。端口1~3可作为ADC的输入；另外，片内温度传感器的输出和电源电压（VDD）也可以作为ADC的输入。用户固件可以将ADC置于关断状态以节省功耗。

A/D转换可以有6种启动方式：软件命令、定时器0溢出、定时器1溢出、定时器2溢出、定时器3溢出或外部转换启动信号。这种灵活性允许用软件事件、周期性信号（定时器溢出）或外部硬件信号触发转换。一个状态位用于指示转换完成，或产生中断（如果被允许）。转换结束后10位结果数据字被锁存到ADC数据寄存器中。

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

图1.10 10位ADC原理框图



C8051F320/1 全速 USB ISP FLASH 微控制器

2. 极限参数

表 2.1 极限参数*

参 数	条 件	最小值	典型值	最大值	单位
环境温度（通电情况下）		-55		125	
储存温度		-65		150	
任何端口I/O引脚或/RST相对GND的电压		-0.3		5.8	V
VDD引脚相对DGND的电压		-0.3		4.2	V
通过VDD和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。

C8051F320/1 全速 USB ISP FLASH 微控制器

3. 总体直流电气特性

表 3.1 总体直流电气特性

-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
数字电源电压 (注1)		2.7	3.3	3.6	V
数字电源电流 (CPU处于活动状态)	VDD=3.3V, CLK=24MHz		10		mA
	VDD=3.3V, CLK= 1MHz		0.6		mA
	VDD=3.3V, CLK=32kHz		30		μA
数字电源电流 (CPU和USB处于活动状态 , 全速或低速)	VDD=3.3V, CLK=24MHz		TBD		mA
	VDD=3.3V, CLK= 6MHz		TBD		mA
数字电源电流 (CPU不活动 , 即不访问FLASH)	VDD=3.3V, CLK=24MHz		5		mA
	VDD=3.3V, CLK= 1MHz		0.3		mA
	VDD=3.3V, CLK=32kHz		14		μA
数字电源电流 (停机方式)	振荡器停止运行		<0.1		μA
RAM数据保持电源电压			1.5		V
SYSCLK (系统时钟) (注2)		0		25	MHz
TSYSH(SYSCLK高电平时间)		18			ns
TSYSL(SYSCLK低电平时间)		18			ns
额定工作温度范围		-40		+85	
注 1 : USB 需要最小 3.0V 的电源电压。					
注 2 : 为能使用调试功能 , SYSCLK 至少应为 32kHz。					

C8051F320/1 全速 USB ISP FLASH 微控制器

4. 引脚和封装定义

表 4.1 C8051F320/1 引脚定义

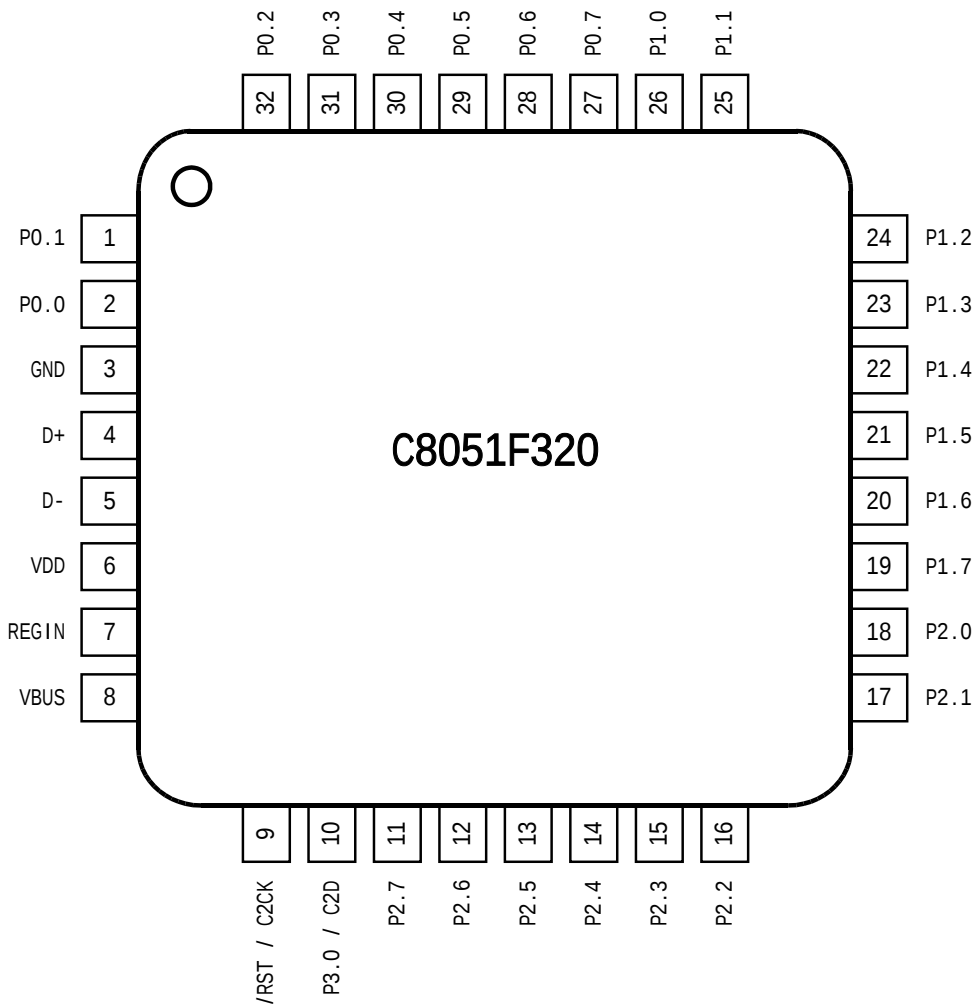
引脚名称	引脚号		引脚类型	说 明
	F320	F321		
VDD	6	6	电源输入 电源输出	2.7V – 3.6V 电源电压输入 3.3V 稳压器输出。见第 8 章。
GND	3	3		地。
/RST	9	9	数字 I/O	器件复位。内部上电复位或 VDD 监视器的漏极开路输出。一个外部源可以通过将该引脚驱动为低电平（至少 15 μ s）来启动一次系统复位。见第 10 章。
C2CK			数字 I/O	C2 调试接口的时钟信号。
P3.0	10	10	数字 I/O	端口 P3.0
C2D			数字 I/O	C2 调试接口的双向数据信号。
REGIN	7	7	电源输入	稳压器的 5V 输入。该引脚是片内稳压器的输入。
VBUS	8	8	数字输入	VBUS 检测输入。该引脚应被连接到 USB 网络的 VBUS 信号。该引脚上出现 5V 信号表示有 USB 连接。
D+	4	4	数字 I/O	USB 的 D+。
D-	5	5	数字 I/O	USB 的 D-。
P0.0	2	2	数字 I/O	端口 P0.0
P0.1	1	1	数字 I/O	端口 P0.1
P0.2	32	28	数字 I/O	端口 P0.2
XTAL1			模拟输入	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	31	27	数字 I/O	端口 P0.3
XTAL2			模拟 I/O 或数字输入	外部时钟输出。该引脚是晶体或陶瓷谐振器的激励驱动器。对于 CMOS 时钟、电容或 RC 振荡器配置，该引脚是外部时钟输入。
P0.4	30	26	数字 I/O	端口 P0.4
P0.5	29	25	数字 I/O	端口 P0.5
P0.6	28	24	数字 I/O	端口 P0.6
CNVSTR			数字输入	ADC0 外部转换启动输入。
P0.7	27	23	数字 I/O	端口 P0.7
VREF			模拟 I/O	外部 VREF 输入或输出。
P1.0	26	22	数字 I/O 或模拟输入	端口 P1.0
P1.1	25	21	数字 I/O 或模拟输入	端口 P1.1
P1.2	24	20	数字 I/O 或模拟输入	端口 P1.2
P1.3	23	19	数字 I/O 或模拟输入	端口 P1.3
P1.4	22	18	数字 I/O 或模拟输入	端口 P1.4
P1.5	21	17	数字 I/O 或模拟输入	端口 P1.5
P1.6	20	16	数字 I/O 或模拟输入	端口 P1.6
P1.7	19	15	数字 I/O 或模拟输入	端口 P1.7

C8051F320/1 全速 USB ISP FLASH 微控制器

表 4.1 C8051F320/1 引脚定义（续）

引脚名称	引脚号		引脚类型	说 明
	F320	F321		
P2.0	18	14	数字 I/O 或模拟输入	端口 P2.0
P2.1	17	13	数字 I/O 或模拟输入	端口 P2.1
P2.2	16	12	数字 I/O 或模拟输入	端口 P2.2
P2.3	15	11	数字 I/O 或模拟输入	端口 P2.3
P2.4	14		数字 I/O 或模拟输入	端口 P2.4
P2.5	13		数字 I/O 或模拟输入	端口 P2.5
P2.6	12		数字 I/O 或模拟输入	端口 P2.6
P2.7	11		数字 I/O 或模拟输入	端口 P2.7

图 4.1 LQFP-32 引脚图（顶视图）



C8051F320/1 全速 USB ISP FLASH 微控制器

图 4.2 LQFP-32 封装图

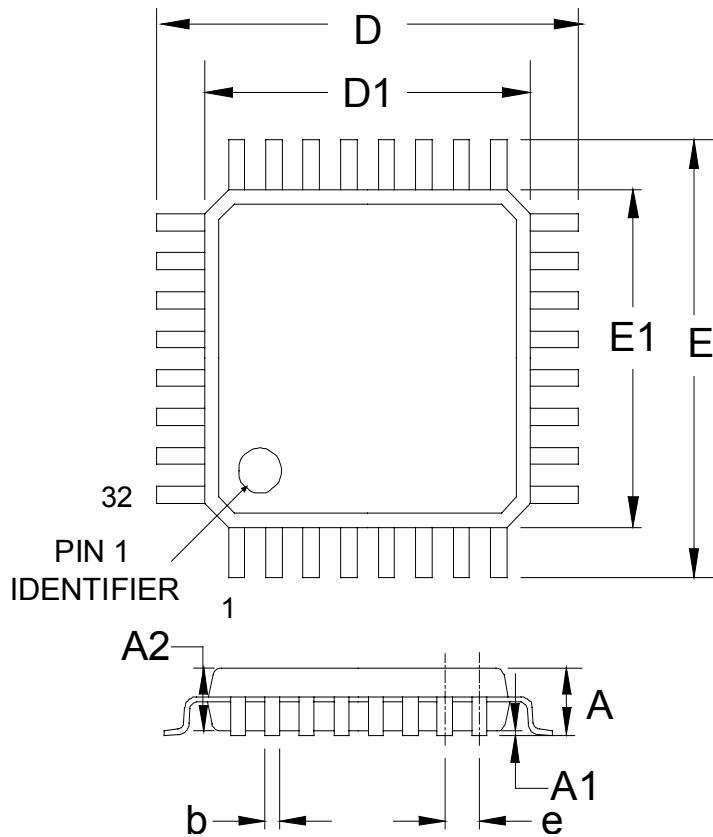
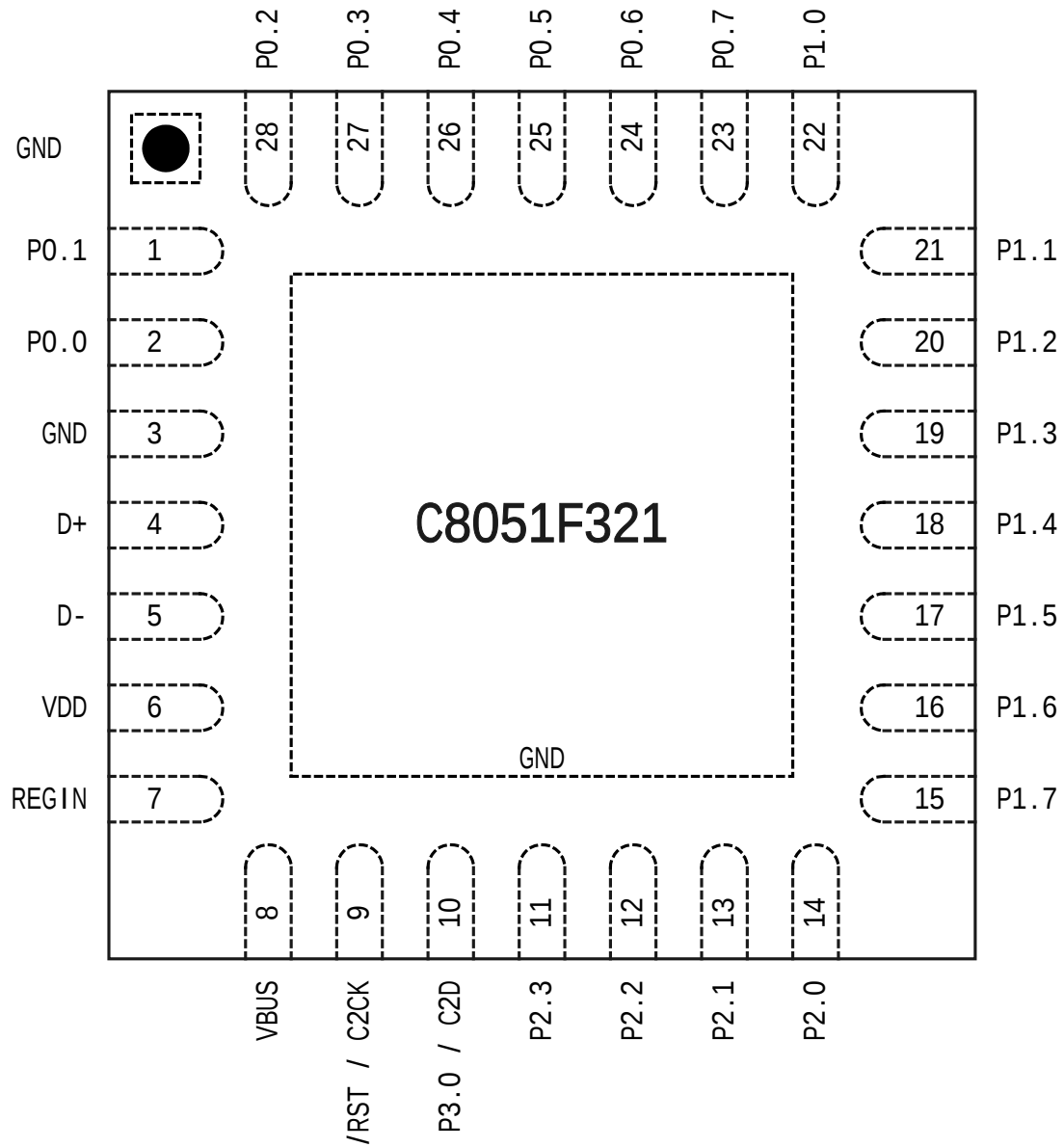


Table 4.2. LQFP-32
Package Dimensions

	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	-

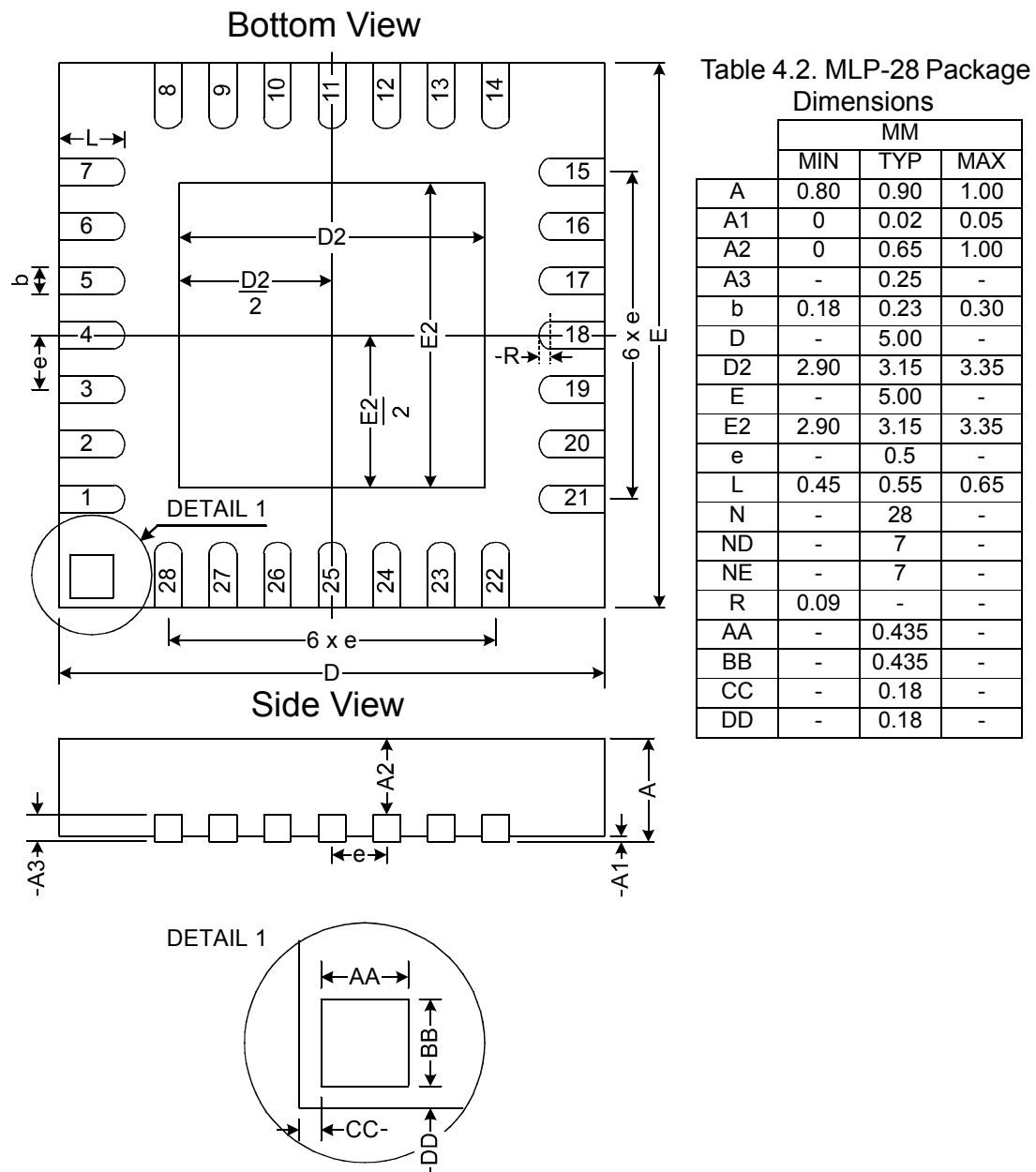
C8051F320/1 全速 USB ISP FLASH 微控制器

图 4.3 MLP-28 引脚图 (顶视图)



C8051F320/1 全速 USB ISP FLASH 微控制器

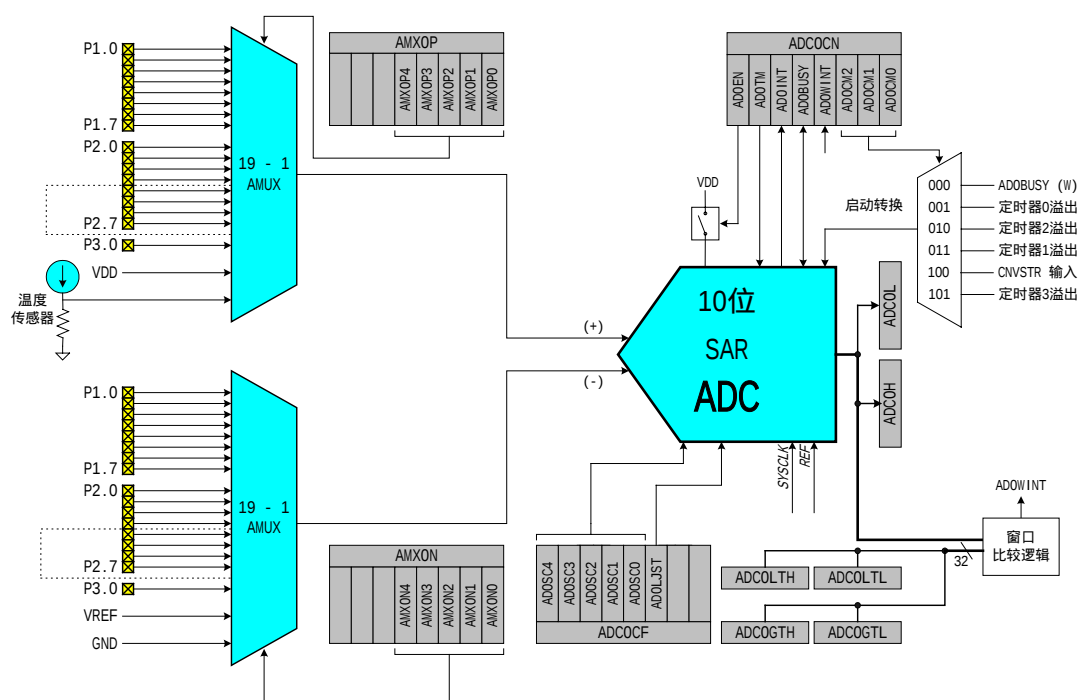
图 4.4 MLP-28 封装图



5. 10 位 ADC (ADC0)

C8051F320/1 的 ADC0 子系统集成了两个 17 通道模拟多路选择器 (合称 AMUX0) 和一个 200kps 的 10 位逐次逼近寄存器型 ADC, ADC 中集成了跟踪保持电路和可编程窗口检测器。AMUX0、数据转换方式及窗口检测器都可用软件通过特殊功能寄存器来配置 (见图 5.1)。ADC0 可以工作在单端方式或差分方式, 可以被配置为用于测量 P1.0~P3.0、温度传感器输出或 VDD (相对于 P1.0~P3.0、VREF 或 GND)。只有当 ADC 控制寄存器(ADCON)中的 AD0EN 位被置 '1' 时 ADC0 子系统才被使能。当 AD0EN 位为 '0' 时, ADC0 子系统处于低功耗关断方式。

图 5.1 ADC0 功能框图



5.1 模拟多路选择器

模拟多路选择器 (AMUX0) 选择去 ADC 的正输入和负输入, P1.0~P3.0、片内温度传感器输出和正电源 (VDD) 中的任何一个都可以被选择为正输入; P1.0~P3.0、VREF 和 GND 中的任何一个都可以被选择为负输入。当 GND 被选择为负输入时, ADC0 工作在单端方式; 在所有其它时间, ADC0 工作在差分方式。ADC0 的输入通道由寄存器 AMX0P 和 AMX0N 选择 (见图 5.5 和图 5.6)。

转换码的格式在单端方式和差分方式下是不同的。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 (ADC0CN.0) 的设置决定。当工作在单端方式时, 转化码为 10 位无符号整数, 所测量的输入范围为 $0 \sim VREF \cdot 1023/1024$ 。下面是单端方式下数据左对齐和右对齐的例子。ADC0H 和 ADC0L 寄存器中未使用的位被设置为 '0'。

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

当工作在差分方式时, 转化码为 10 位有符号整数(2 的补码), 所测量的输入范围为 $-VREF \sim VREF \cdot 511/512$ 。下面是差分方式下数据左对齐和右对齐的例子。对于右对齐数据, ADC0H 寄存器中未用的高位填充数据字的符号扩展位。对于左对齐数据, ADC0L 寄存器中未用的低位被清 0。

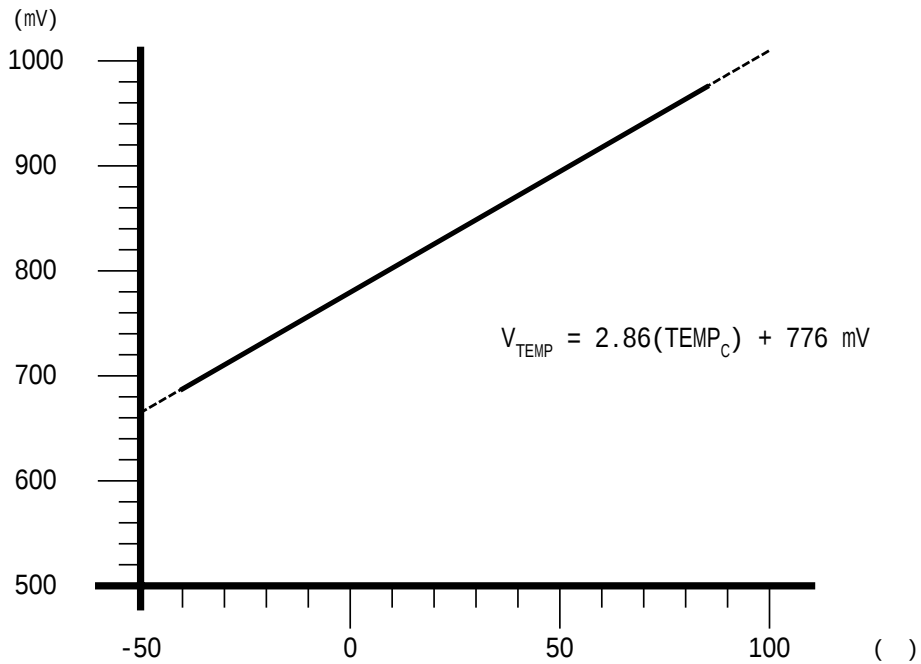
输入电压 (差分)	右对齐的 ADC0H:ADC0L (AD0LJST = 0)	左对齐的 ADC0H:ADC0L (AD0LJST = 1)
$VREF \cdot 511/512$	0x01FF	0x7FC0
$VREF \cdot 256/512$	0x0100	0x4000
0	0x0000	0x0000
$-VREF \cdot 256/512$	0xFF00	0xC000
$-VREF$	0xFE00	0x8000

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 (n=0,1,2,3) 寄存器中的对应位置 0。为了使交叉开关跳过一个端口引脚, 应将 PnSKIP (n=0,1,2) 寄存器中的对应位置 1。有关端口 I/O 配置的详细信息见“14. 端口输入/输出”。

5.2 温度传感器

温度传感器的典型传输函数示于图 5.2。当温度传感器被寄存器 AMX0P 中的 AMX0P4-0 位选中时，输出电压（ V_{TEMP} ）为 ADC 的正输入。

图 5.2 温度传感器典型传输函数



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。

5.3 工作方式

ADC0 的最高转换速度为 200ksp/s。ADC0 的转换时钟由系统时钟分频得到，分频系数由 ADC0CF 寄存器的 AD0SC 位决定（转换时钟为系统时钟/（AD0SC+1）， $0 \leq \text{AD0SC} \leq 31$ ）。

5.3.1 转换启动方式

有 6 种 A/D 转换启动方式，由 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 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 1 溢出；
5. CNVSTR 输入信号（P0.6）的上升沿；
6. 定时器 3 溢出。

向 AD0BUSY 写‘1’方式提供了用软件控制 ADC0 转换的能力。AD0BUSY 位在转换期间被置‘1’，转换结束后复‘0’。AD0BUSY 位的下降沿触发中断（当被允许时）并置位 ADC0CN 中的中断标志（AD0INT）。注意：当工作在查询方式时，应使用 ADC0 中断标志（AD0INT）来查询 ADC 转换是否完成。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“19. 定时器”。

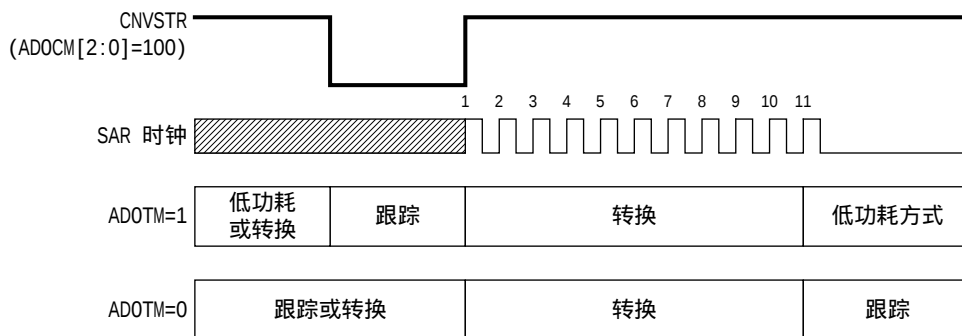
需要注意的是，CNVSTR 输入引脚还是端口引脚 P0.6。当使用 CNVSTR 输入作为转换启动源时，P0.6 应被数字交叉开关跳过。为使交叉开关跳过 P0.6，应将寄存器 P0SKIP 中的位 6 置‘1’。有关端口 I/O 配置的详细信息，见“14. 端口输入/输出”。

5.3.2 跟踪方式

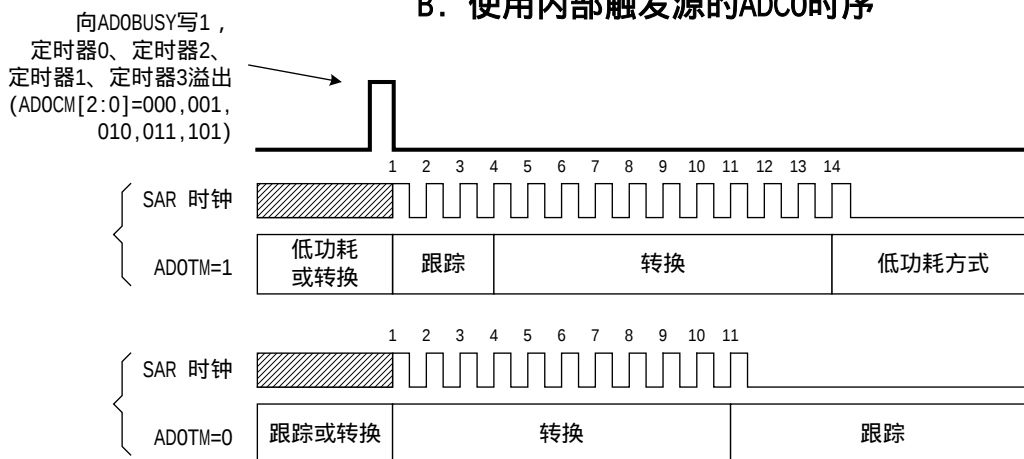
寄存器 ADC0CN 中的 AD0TM 位控制 ADC0 的跟踪保持方式。在缺省状态，ADC0 输入被连续跟踪（转换期间除外）。当 AD0TM 位被置‘1’时，ADC0 工作在低功耗跟踪保持方式。在该方式，每次转换前有 3 个 SAR 时钟的跟踪时间（跟踪发生在转换启动信号有效之后）。在低功耗跟踪保持方式下使用 CNVSTR 信号作为转换启动源时，只在 CNVSTR 输入为低电平时跟踪；从 CNVSTR 的上升沿开始转换（见图 5.3）。当器件处于低功耗停机或休眠方式时，可以禁止跟踪。低功耗跟踪和保持方式在 AMUX 的设置经常改变时也是很有用的，因为 ADC 有建立时间要求（见“5.3.3 建立时间要求”）。

图 5.3 10 位 ADC0 跟踪和转换时序示例

A. 使用外部触发源的ADC0时序



B. 使用内部触发源的ADC0时序



5.3.3 建立时间要求

当 ADC 输入配置发生改变时（即 AMUX0 的选择发生变化），在进行一次精确的转换之前需要有一个最小的跟踪时间。该跟踪时间由 AMUX0 的电阻、ADC0 采样电容、外部信号源阻抗及所要求的转换精度决定。注意：在低功耗跟踪方式，每次转换需要用三个 SAR 时钟跟踪。对于大多数应用，三个 SAR 时钟可以满足最小跟踪时间的要求。

图 5.4 给出了单端和差分方式下等效的 ADC0 输入电路，这两种电路的时间常数相等。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} 等于 R_{MUX} 。表 5.1 给出了 ADC0 的最小建立时间要求。

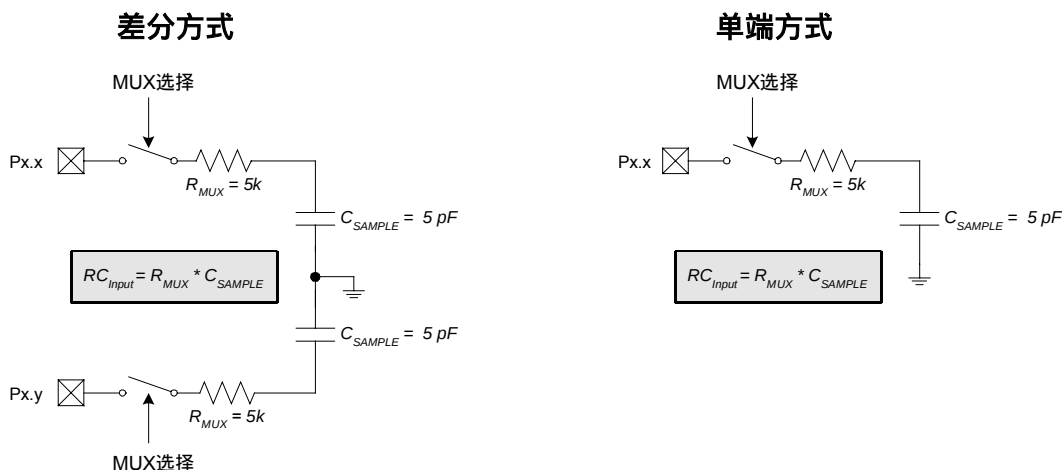
方程 5.1 ADC0 建立时间要求

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

其中：

- SA 是建立精度，用一个 LSB 的分数表示（例如，建立精度 0.25 对应 1/4 LSB）；
- t 为所需要的建立时间，以秒为单位；
- R_{TOTAL} 为 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）。

图 5.4 ADC0 等效输入电路



C8051F320/1 全速 USB ISP FLASH 微控制器

图 5.5 AMX0P: AMUX0 正输入通道选择寄存器

R	R	R	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	AMX0P4	AMX0P3	AMX0P2	AMX0P1	AMX0P0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xBB

位 7-5：未使用。读=000b，写=忽略。
 位 4-0：AMX0P4-0：AMUX0 正输入选择

AMX0P4-0	ADC0 正输入
00000	P1.0
00001	P1.1
00010	P1.2
00011	P1.3
00100	P1.4
00101	P1.5
00110	P1.6
00111	P1.7
01000	P2.0
01001	P2.1
01010	P2.2
01011	P2.3
01100	P2.4
01101	P2.5
01110	P2.6
01111	P2.7
10000	P3.0
10001~11101	保留
11110	温度传感器
11111	VDD

注：01100 ~ 01111 只适用于 C8051F320；对于 C8051F321，这些选择值保留。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 5.6 AMX0N: AMUX0 负输入通道选择寄存器

R	R	R	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	AMX0N4	AMX0N3	AMX0N2	AMX0N1	AMX0N0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xBA

位 7-5：未使用。读=000b，写=忽略。

位 4-0：AMX0N4-0：AMUX0 负输入选择

当 GND 被选择为负输入时，ADC0 工作在单端方式。对于所有其它负输入选择，ADC0 工作在差分方式。

AMX0N4-0	ADC0 负输入
00000	P1.0
00001	P1.1
00010	P1.2
00011	P1.3
00100	P1.4
00101	P1.5
00110	P1.6
00111	P1.7
01000	P2.0
01001	P2.1
01010	P2.2
01011	P2.3
01100	P2.4
01101	P2.5
01110	P2.6
01111	P2.7
10000	P3.0
10000	VREF
10001~11101	保留
11110	VREF
11111	GND (ADC 工作在单端方式)

注：01100 ~ 01111 只适用于 C8051F320；对于 C8051F321，这些选择值保留。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 5.7 ADC0CF: ADC0 配置寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AD0LJST	-	-	11111000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xBC

位 7-3： AD0SC4-0：ADC0 SAR 转换时钟周期控制位
SAR 转换时钟来源于系统时钟，由下面的方程给出，其中 $AD0SC$ 表示 AD0SC4-0 中保存的 5 位数值。对 SAR 转换时钟频率的要求见表 5.1。

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

位 2 AD0LJST：ADC0 左对齐选择位。
0：ADC0H:ADC0L 中的数据为右对齐。
1：ADC0H:ADC0L 中的数据为左对齐。

位 1-0：未使用。读=00b，写=忽略。

图 5.8 ADC0H: ADC0 数据字高字节寄存器

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

位 7-0： ADC0 数据字高字节。

AD0LJST = 0 时：位 7~2 为位 1 的符号扩展位，位 1~0 为 10 位 ADC0 数据的高 2 位。

AD0LJST = 1 时：位 7~0 是 10 位 ADC0 数据的高 8 位。

图 5.9 ADC0L: ADC0 数据字低字节寄存器

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

位 7-0： ADC0 数据字低字节。

AD0LJST = 0 时：位 7~0 是 10 位 ADC0 数据的低 8 位。

AD0LJST = 1 时：位 7~6 是 10 位 ADC0 数据的低 2 位，位 5~0 的读出值总是为 0。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 5.10 ADC0CN: ADC0 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AD0EN	AD0TM	AD0INT	AD0BUSY	AD0WINT	AD0CM2	AD0CM1	AD0CM0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： (可位寻址) 0xE8
位 7： AD0EN：ADC0 使能位 0：ADC0 禁止。ADC0 处于低耗停机状态。 1：ADC0 使能。ADC0 处于活动状态，可以进行转换数据。 位 6： AD0TM：ADC0 跟踪方式位 0：正常方式：当 ADC0 被使能时，除了转换期间之外一直处于跟踪方式。 1：低功耗跟踪方式：由 AD0CM2-0 定义跟踪方式（见下面的说明）。 位 5： AD0INT：ADC0 转换结束中断标志 0：从最后一次将该位清‘0’后，ADC0 还没有完成一次数据转换。 1：ADC0 完成了一次数据转换。 位 4： AD0BUSY：ADC0 忙标志位 读 0：ADC0 转换结束或当前不在进行数据转换。AD0INT 在 AD0BUSY 的下降沿被置‘1’。 1：ADC0 正在进行转换。 写 0：无作用。 1：若 AD0CM2-0 = 000b 则启动 ADC0 转换。 位 3： AD0WINT：ADC0 窗口比较中断标志 0：未发生 ADC0 窗口比较匹配。 1：发生 ADC0 窗口比较匹配。 位 2-0： AD0CM2-0：ADC0 转换启动方式选择 当 AD0TM = 0 时： 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 1 溢出启动 ADC0 转换。 100：外部 CNVSTR 输入信号的上升沿启动 ADC0 转换。 101：定时器 3 溢出启动 ADC0 转换。 11x：保留。 当 AD0TM = 1 时： 000：向 AD0BUSY 写 1 时启动跟踪，持续 3 个 SAR 时钟后开始转换。 001：定时器 0 溢出启动跟踪，持续 3 个 SAR 时钟后开始转换。 010：定时器 2 溢出启动跟踪，持续 3 个 SAR 时钟后开始转换。 011：定时器 1 溢出启动跟踪，持续 3 个 SAR 时钟后开始转换。 100：ADC0 只在 CNVSTR 输入为逻辑低电平时跟踪，在 CNVSTR 的上升沿开始转换。 101：定时器 3 溢出启动跟踪，持续 3 个 SAR 时钟后开始转换。 11x：保留。								

C8051F320/1 全速 USB ISP FLASH 微控制器

5.4 可编程窗口检测器

ADC 可编程窗口检测器不停地将 ADC0 输出与用户编程的极限值进行比较，并在检测到越限条件时通知系统控制器。这在一个中断驱动的系统尤其有效，既可以节省代码空间和 CPU 带宽又能提供快速响应时间。窗口检测器中断标志（ADC0CN 中的 AD0WINT）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 寄存器的编程值。

窗口检测器寄存器的数据格式（左/右对齐，有符号/无符号）必须与当前的 ADC 配置格式（左/右对齐，单端/差分）一致。

图 5.11 ADC0GTH: ADC0 下限（大于）数据字高字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xC4
位 7-0： ADC0 下限数据字高字节。								

图 5.12 ADC0GTL: ADC0 下限（大于）数据字低字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xC3
位 7-0： ADC0 下限数据字低字节。								

图 5.13 ADC0LTH: ADC0 上限（小于）数据字高字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xC6
位 7-0： ADC0 上限数据字高字节。								

图 5.14 ADC0LTL: ADC0 上限（小于）数据字低字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xC5
位 7-0： ADC0 上限数据字低字节。								

5.4.1 单端方式下的窗口检测器

图 5.15 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 $ADC0GTH:ADC0GTL = 0x0080$ 。在单端方式，转换码是 10 位无符号整数，对应的电压范围为 $0 \sim VREF \times 1023/1024$ 。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 5.16 给出了单端方式下数据左对齐窗口比较的例子。

图 5.15 ADC 窗口中断示例（单端方式右对齐数据）

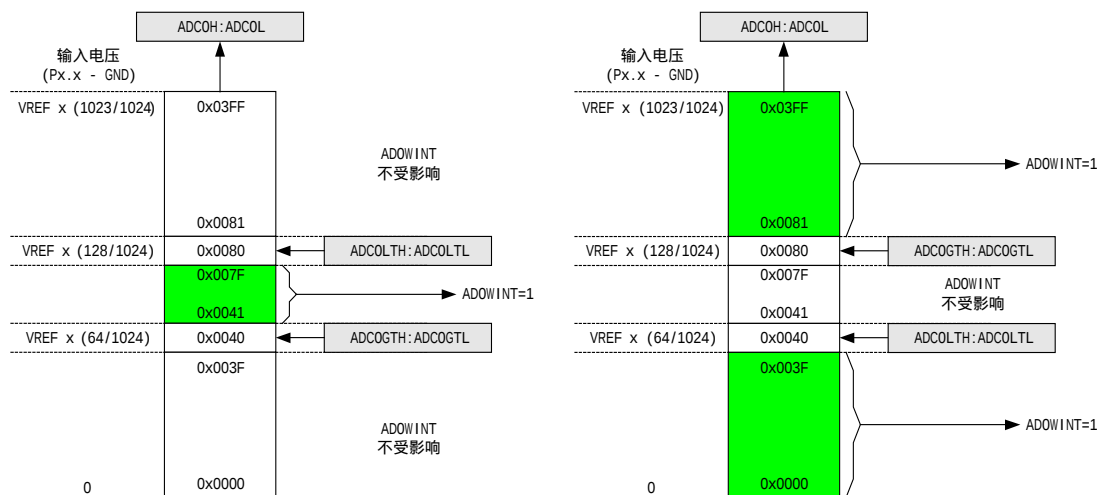
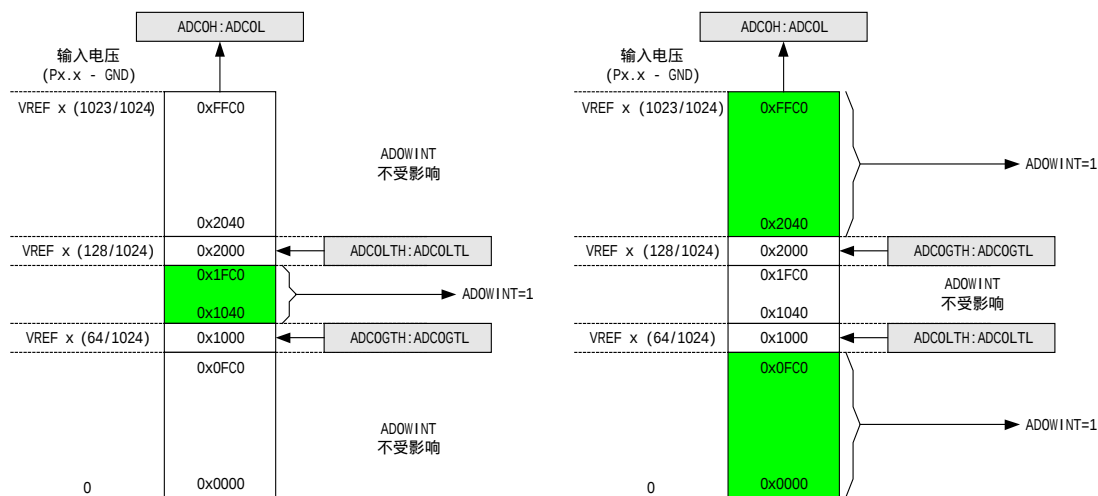


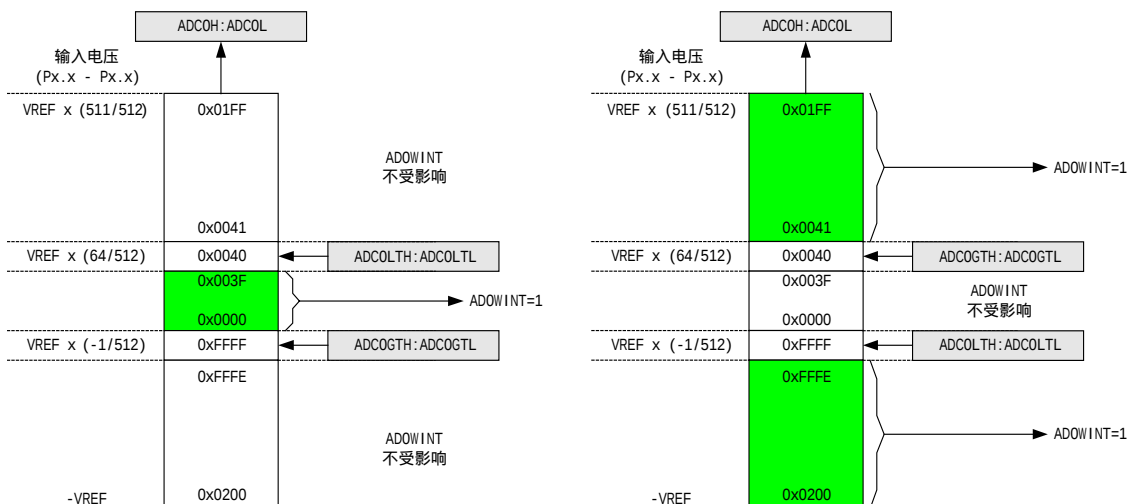
图 5.16 ADC 窗口中断示例（单端方式左对齐数据）



5.4.2 差分方式下的窗口检测器

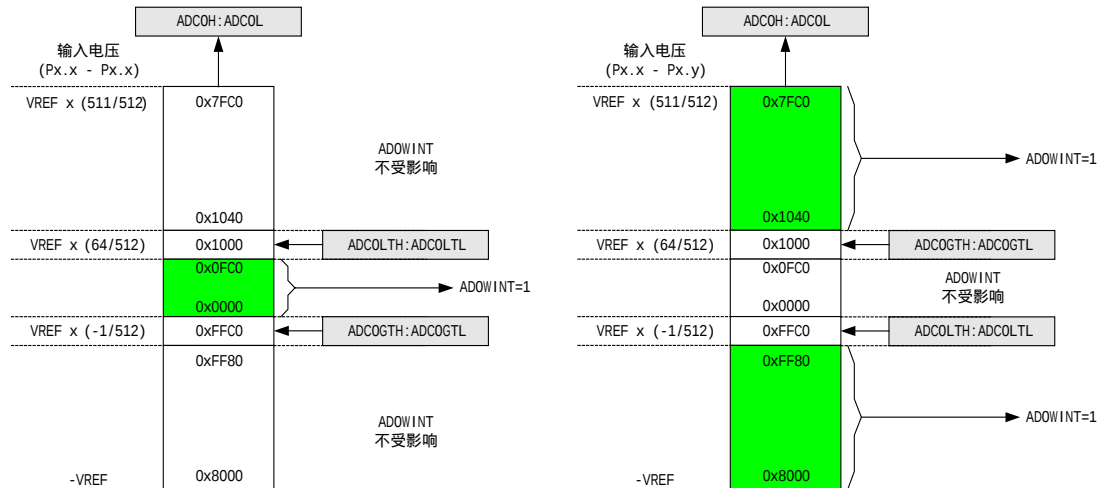
图 5.17 给出了差分方式下数据右对齐窗口比较的两个例子。左边的例子所使用的极限值为：ADC0LTH:ADC0LTL = 0x0040 (+64d) 和 ADC0GTH:ADC0GTL = 0xFFFF (-1d)；右边的例子所使用的极限值为：ADC0LTH:ADC0LTL = 0xFFFF (-1d) 和 ADC0GTH:ADC0GTL = 0x0040(+64d)。在差分方式，转换码是 10 位有符号整数(2 的补码)，对应的电压范围为-VREF ~ VREF*511/512。对于左边的例子，如果 ADC0 转换结果数据字 (ADC0H:ADC0L) 位于由 ADC0GTH:ADC0GTL 和 ADC0LTH:ADC0LTL 定义的范围之内 (即 $0xFFFF (-1d) < ADC0H:ADC0L < 0x0040 (+64d)$)，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 (ADC0H:ADC0L) 位于由 ADC0GT 和 ADC0LT 定义的范围之外 (即 $ADC0H:ADC0L < 0xFFFF (-1d)$ 或 $ADC0H:ADC0L > 0x0040(+64d)$)，则会产生一个 AD0WINT 中断。图 5.18 给出了差分方式下数据左对齐窗口比较的例子。

图 5.17 ADC 窗口比较示例 (差分方式右对齐数据)



C8051F320/1 全速 USB ISP FLASH 微控制器

图 5.18 ADC 窗口比较示例（差分方式左对齐数据）



C8051F320/1 全速 USB ISP FLASH 微控制器

表 5.1 ADC0 电气特性

VDD=3.0V, VREF=2.40V(REFSL=0), -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			0		LSB
满度误差	差分方式		-1		LSB
偏移温度系数			10		ppm/
动态性能 (10kHz 正弦波差分输入, 满度值之下 1dB, 200ksps)					
信号与噪声加失真比		53	55.5		dB
总谐波失真	到 5 次谐波		-67		dB
无失真动态范围			78		dB
转换速率					
SAR 转换时钟				3	MHz
转换时间 (SAR 时钟数)		10			时钟
跟踪/保持捕获时间		300			ns
转换速率				200	ksps
模拟输入					
ADC 输入电压范围	单端方式(AIN+-GND)	0		VREF	V
	差分方式(AIN+-AIN-)	-VREF		VREF	V
引脚输入电压 (相对于 GND)	单端或差分方式	0		VDD	V
输入电容			5		pF
温度传感器					
线性度	注 1		± 0.1		
增益	注 2		2.86		mV/
偏移	注 1, 2 (温度=0)		776 ± 8.5		mV
电源指标					
电 源 电 流 (VDD 给 ADC0 供电)	工作方式, 200ksps		400	900	μA
电源抑制比			± 0.3		mV/V

注 1 : 包括 ADC 偏移、增益和线性度偏差。

注 2 : 代表偏离平均值一个标准偏差。

6. 电压基准

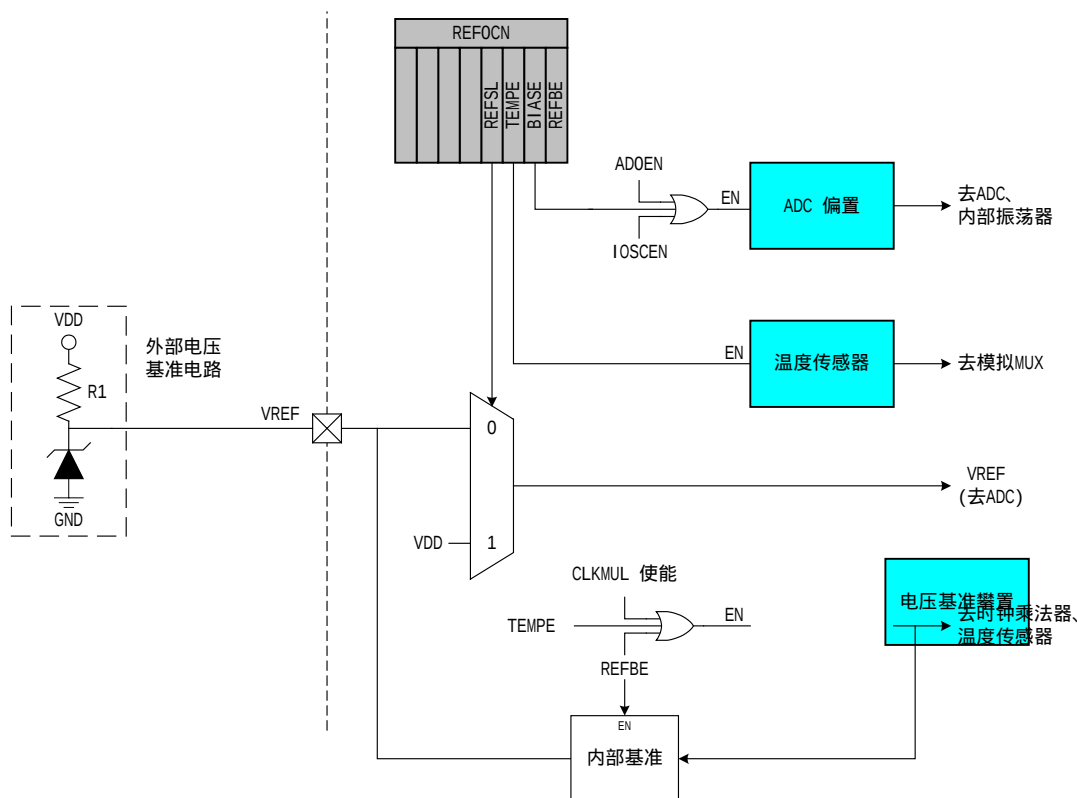
C8051F320/1 的电压基准 MUX 可以被配置为连接到外部电压基准、内部电压基准或电源电压 VDD (见图 6.1)。基准控制寄存器 REF0CN 中的 REFSL 位用于选择基准源。选择使用外部或内部基准时, REFSL 位应被设置 ‘0’; 选择 VDD 作为基准源时, REFSL 应被置 ‘1’。

REF0CN 中的 BIASE 位使能内部 ADC 偏压发生器。ADC 和内部振荡器要使用偏压发生器提供的偏置电压。当这些外设中的任何一个被使能时, BIASE 位被自动置 ‘1’。也可以通过向 REF0CN 中的 BIASE 位写 ‘1’ 来使能偏压发生器, 见图 6.2 对 REF0CN 寄存器的详细说明。基准偏压发生器 (见图 6.1) 用于内部电压基准、温度传感器和时钟乘法器。当这些部件中的任何一个被使能时, 基准偏压发生器被自动使能。表 6.1 给出了电压基准和偏压电路的电气特性。

注意: 端口引脚 P0.7 用作外部 VREF 输入和内部 VREF 的输出。当使用外部电压基准时, P0.7 应被配置为模拟输入并被数字交叉开关跳过。为了将 P0.7 配置为模拟输入, 应将 P0MDIN 寄存器的位 7 设置为 ‘0’。为使交叉开关跳过 P0.7, 应将 P0SKIP 寄存器的位 7 置 ‘1’。有关端口 I/O 配置的详细信息, 见 “14 端口输入/输出”。

温度传感器连接到 ADC0 正输入多路选择器 (见 “5.1 模拟多路选择器”)。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。

图 6.1 电压基准功能框图



C8051F320/1 全速 USB ISP FLASH 微控制器

图 6.2 REF0CN: 电压基准控制寄存器

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

位 7-4：未用。读 = 0000b，写 = 忽略。
 位 3：REFSL：电压基准选择
 0：VREF 引脚作为电压基准。
 1：VDD 作为电压基准。
 位 2：TEMPE：温度传感器使能位
 0：内部温度传感器关闭。
 1：内部温度传感器工作。
 位 1：BIASE：内部模拟偏压发生器使能位
 0：内部偏压发生器关闭。
 1：内部偏压发生器工作。
 位 0：REFBE：内部基准缓冲器使能位
 0：内部基准缓冲器被禁止。
 1：内部基准缓冲器被使能。内部电压基准被驱动到 VREF 引脚。

表 6.1 电压基准的电气特性

VDD=3.0V，-40 到+85 （除非特别说明）

参 数	条 件	最小值	典型值	最大值	单 位
内部基准 (REFBE = 1)					
输出电压	环境温度 25	2.38	2.44	2.50	V
VREF 短路电流				10	mA
VREF 温度系数			15		ppm/
负载调整	负载 = 0-200 μ A 到 AGND		1.5		ppm/ μ A
VREF 开启时间 1	4.7 μ F 钽电容，0.1 μ F 陶瓷旁路电容		2		ms
VREF 开启时间 2	0.1 μ F 陶瓷旁路电容		20		μ s
VREF 开启时间 3	无旁路电容		10		μ s
电源抑制			140		ppm/V
外部基准 (REFBE = 0)					
输入电压范围		0		VDD	V
输入电流	采样频率 = 200ksps，VREF=3.0V		12		μ A
偏压发生器					
ADC 偏压发生器	BIASE = 1		100		μ A
基准偏压发生器			40		μ A

7. 比较器

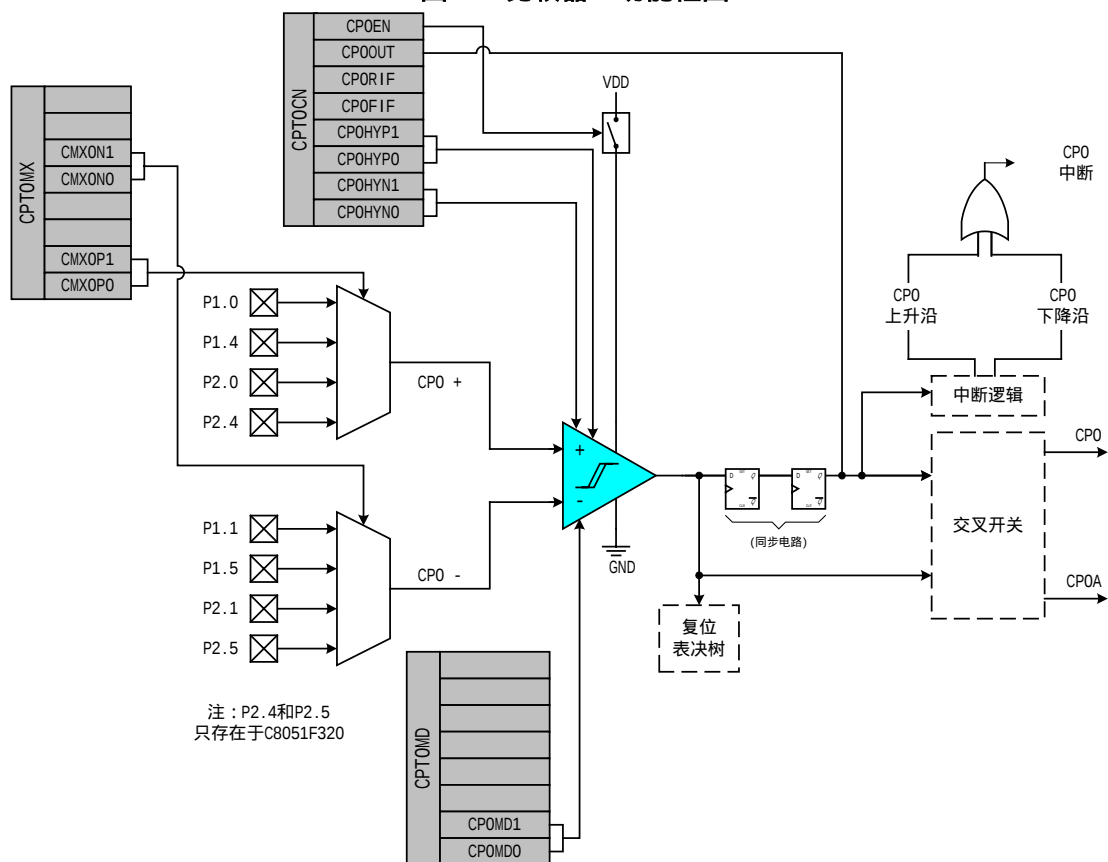
C8051F320/1 器件内部有两个可编程电压比较器，即比较器 0（如图 7.1 所示）和比较器 1（如图 7.2 所示）。除了输入引脚不同和比较器 0 可以作为复位源之外，这两个比较器的操作完全相同。

每个比较器的响应时间和回差电压都是可编程的。每个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，异步输出信号仍然可用，这就允许比较器在器件处于停机（STOP）方式时工作并产生输出。当被分配了端口引脚时，比较器的输出可以被配置为漏极开路或推挽方式。比较器 0 可以被用作复位源（见“10.5 比较器复位”）。

比较器 0 的输入用 CPT0MX 寄存器（见图 7.5）来选择。CMX0P1 和 CMX0P0 位选择比较器 0 的正输入，CMX0N1 和 CMX0N0 位选择比较器 0 的负输入。比较器 1 的输入用 CPT1MX 寄存器（见图 7.8）来选择。CMX1P1 和 CMX1P0 位选择比较器 1 的正输入，CMX1N1 和 CMX1N0 位选择比较器 1 的负输入。

注意：被选择为比较器输入的引脚应被配置为模拟输入（通过相应的端口配置寄存器），交叉开关应被配置为跳过这些引脚，有关端口配置的详细信息见“14.3 通用端口 I/O”。

图 7.1 比较器 0 功能框图

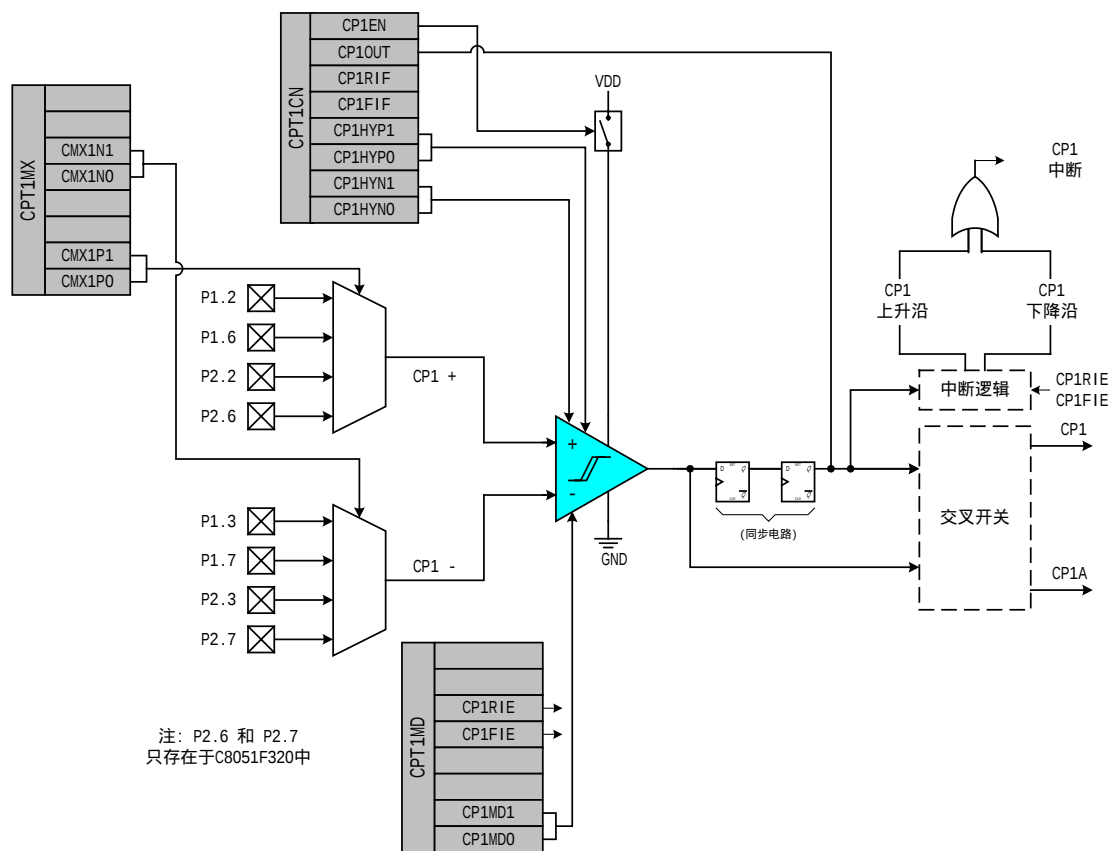


C8051F320/1 全速 USB ISP FLASH 微控制器

比较器的输出可以被软件查询，可以作为中断源，也可以被连到端口引脚。当被连到端口引脚时，比较器的输出可以是与系统时钟同步的或者是不同步的。即使在停机方式（系统时钟停止），异步输出信号仍然可用。当被禁止时，比较器输出（如果已通过交叉开关分配了端口 I/O 引脚）的缺省值为逻辑低电平，电源电流降到小于 100 nA。有关通过交叉开关配置比较器输出的详细信息见“14.1 优先权交叉开关译码器”。比较器的输入可以承受-0.25V 到 (VDD) + 0.25V 的外部驱动电压而不至损坏或发生工作错误。表 7.1 给出了比较器的电气特性。

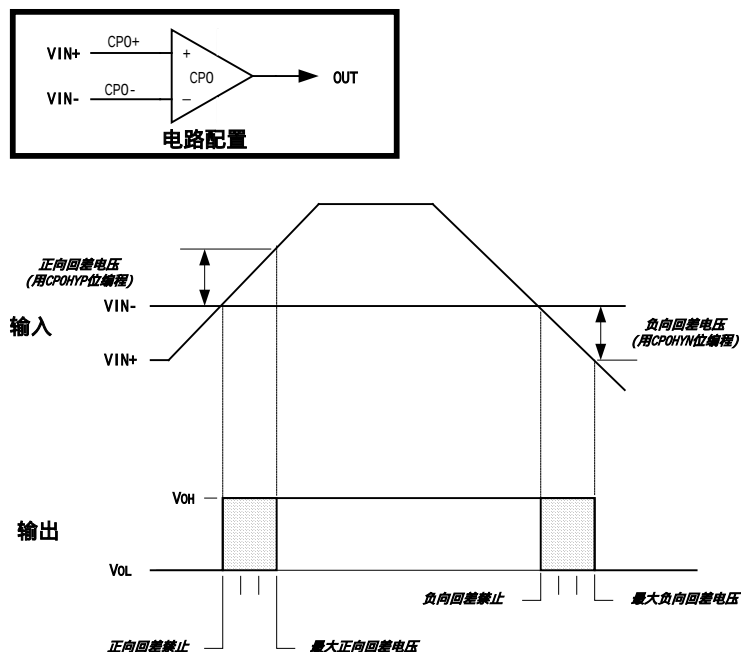
可以通过对寄存器 CPTnMD（见图 7.6 和图 7.9）编程来设置比较器的响应时间。选择较长的响应时间可以减小比较器消耗的电流。有关时序和电源电流的详细信息见表 7.1。

图 7.2 比较器 1 功能框图



注：P2.6 和 P2.7
只存在于 C8051F320 中

图 7.3 比较器回差电压曲线



比较器的回差电压可以通过比较器控制寄存器 CPTnCN (见图 7.4 和图 7.7) 用软件编程。负向回差电压值由 CPnHYN 位的设置决定。如图 7.3 所示，可以设置 20mV、10mV 或 5mV 的负向回差电压值，或者禁止负向回差电压。类似地，通过编程 CPnHYP 位设置正向回差电压值。

在比较器输出的上升沿和下降沿都可以产生中断。比较器的下降沿置 '1' CPnFIF 中断标志，比较器的上升沿置 '1' CPnRIF 中断标志。这些位一旦被置 '1'，将一直保持 '1' 状态直到被软件清除。可以在任意时刻通过读取 CPnOUT 位得到比较器的输出状态。通过置 '1' CPnEN 位来使能比较器，通过将该位清 '0' 来禁止比较器。

注意：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，可能产生假上升沿或下降沿。建议在比较器被使能或方式位发生变化后经过一段延时将上升沿和下降沿标志清 0。表 7.1 中给出了上电时间。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 7.3 CPT0CN: 比较器 0 控制寄存器

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	SFR地址： 0x9B
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：自该标志位被清除后，没有发生过比较器 0 上升沿中断。 1：自该标志位被清除后，发生了比较器 0 上升沿中断。 位 4： CP0FIF：比较器 0 下降沿中断标志，必须用软件清 0 0：自该标志位被清除后，没有发生过比较器 0 下降沿中断。 1：自该标志位被清除后，发生了比较器 0 下降沿中断。 位 3-2： CP0HYP1-0：比较器 0 正向回差电压控制位 00：禁止正向回差电压。 01：正向回差电压= 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。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 7.4 CPT0MX: 比较器 0 MUX 选择寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	CMX0N1	CMX0N0	-	-	CMX0P1	CMX0P0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0x9F

位 7-6：未用。读 = 00b，写 = 忽略。

位 5-4：CMX0N1-0: 比较器 0 负输入 MUX 选择
这些位选择作为比较器 0 负输入的端口引脚。

CMX0N1	CMX0N0	负输入
0	0	P1.1
0	1	P1.5
1	0	P2.1
1	1	P2.5

位 3-2：未用。读 = 00b，写 = 忽略。

位 1-0：CMX0P1-0: 比较器 0 正输入 MUX 选择
这些位选择作为比较器 0 正输入的端口引脚。

CMX0P1	CMX0P0	负输入
0	0	P1.0
0	1	P1.4
1	0	P2.0
1	1	P2.4

注：P2.4 和 P2.5 只存在于 C8051F320 中，对应这两个引脚的选项在 C8051F321 中被保留。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 7.5 CPT0MD: 比较器方式选择寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	CP0RIE	CP0FIE	-	-	CP0MD1	CP0MD0	00000010
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0x9D

位 7-6：未用。读 = 00b，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
这两位选择比较器 0 的响应时间。

方式	CP0MD1	CP0MD0	CP0 响应时间（典型值）
0	0	0	100ns
1	0	1	175ns
2	1	0	320ns
3	1	1	1050ns

C8051F320/1 全速 USB ISP FLASH 微控制器

图 7.7 CPT1CN: 比较器 1 控制寄存器

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	SFR地址： 0x9A
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 0：自该标志位被清除后，没有发生过比较器 1 上升沿中断。 1：自该标志位被清除后，发生了比较器 1 上升沿中断。 位 4： CP1FIF：比较器 1 下降沿中断标志 0：自该标志位被清除后，没有发生过比较器 1 下降沿中断。 1：自该标志位被清除后，发生了比较器 1 下降沿中断。 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。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 7.8 CPT1MX: 比较器 1 MUX 选择寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	CMX1N1	CMX1N0	-	-	CMX1P1	CMX1P0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0x9E

位 7-6：未用。读 = 00b，写 = 忽略。

位 5-4：CMX1N1-0: 比较器 1 负输入 MUX 选择
这两位选择作为比较器 1 负输入的端口引脚。

CMX1N1	CMX1N0	负输入
0	0	P1.3
0	1	P1.7
1	0	P2.3
1	1	P2.7

位 3-2：未用。读 = 00b，写 = 忽略。

位 1-0：CMX1P1-0: 比较器 1 正输入 MUX 选择
这两位选择作为比较器 1 正输入的端口引脚。

CMX1P1	CMX1P0	正输入
0	0	P1.2
0	1	P1.6
1	0	P2.2
1	1	P2.6

注：P2.6 和 P2.7 只存在于 C8051F320 中，对应这两个引脚的选项在 C8051F321 中被保留。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 7.9 CPT1MD: 比较器方式选择寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	CP1RIE	CP1FIE	-	-	CP1MD1	CP1MD0	00000010
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0x9C
位 7-6：未用。读 = 00b，写 = 忽略。 位 5：CP1RIE：比较器 1 上升沿中断允许 0：禁止比较器 1 上升沿中断 1：允许比较器 1 上升沿中断 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 这两位选择比较器 1 的响应时间。								
方式	CP1MD1	CP1MD0	CP1 响应时间(典型值)					
0	0	0	100ns					
1	0	1	175ns					
2	1	0	320ns					
3	1	1	1050ns					

C8051F320/1 全速 USB ISP FLASH 微控制器

表 7.1 比较器电气特性

VDD=3.0V, -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：	(CP0+) - (CP0-) = 100mV		100		ns
方式 0, $V_{cm}^{\dagger}=1.5V$	(CP0+) - (CP0-) = -100mV		250		ns
响应时间：	(CP0+) - (CP0-) = 100mV		175		ns
方式 1, $V_{cm}^{\dagger}=1.5V$	(CP0+) - (CP0-) = -100mV		500		ns
响应时间：	(CP0+) - (CP0-) = 100mV		320		ns
方式 2, $V_{cm}^{\dagger}=1.5V$	(CP0+) - (CP0-) = -100mV		1100		ns
响应时间：	(CP0+) - (CP0-) = 100mV		1050		ns
方式 3, $V_{cm}^{\dagger}=1.5V$	(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			3		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		μ S
电源电流 (DC)	方式 0		7.6		μ A
	方式 1		3.2		μ A
	方式 2		1.3		μ A
	方式 3		0.4		μ A

$^{\dagger}V_{cm}$ 是CP0+和CP0-上的共模电压。

C8051F320/1 全速 USB ISP FLASH 微控制器

8. 稳压器 (REG0)

C8051F320/1 器件内部包含一个5V至3V的稳压器 (REG0)。当被使能时，REG0输出连到VDD引脚，并可为外部器件提供电源。可以用软件通过寄存器REG0CN中的REGEN位使能/禁止REG0。REG0的电气特性见表8.1。

当在 UAB 网络中使用 C8051F320/1 器件时，VBUS 信号必须被连到 VBUS 引脚。当器件工作在总线供电方式时，VBUS 信号只应连接到 REGIN 引脚。REG0 的配置选项示于图 8.1 ~ 图 8.4 中。

8.1 稳压器方式选择

REG0 提供低功耗工作方式，在器件处于挂起方式时使用。在低功耗方式，REG0 输出电压保持正常，但 REG0 的动态性能（响应时间）降低。表 8.1 给出了正常方式和低功耗方式下的电源电流。REG0 方式选择由寄存器 REG0CN 中的 REGMOD 位控制。

8.2 VBUS 检测

当使用 USB 功能控制器时，VBUS 信号应被连接到 VBUS 引脚。VBSTAT 位（寄存器 REG0CN）指示 VBUS 信号的当前逻辑电平。当 VBUS 信号与寄存器 REG0CN 中的 VBPOL 位所选择的极性一致时，会产生 VBUS 中断（如果被允许）。VBUS 中断是电平敏感的，没有相应的中断请求标志。只要 VBUS 信号与 VBPOL 选择的极性一致，VBUS 中断就有效。表 8.1 给出了 VBUS 输入信号的参数。

注意：在 USB0 被选择为复位源的情况下，当 VBUS 信号与 VBPOL 位选择的极性匹配时会导致产生系统复位。

表 8.1 稳压器电气特性

VDD=3.0V，-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		4.0		5.25	V
输出电压	输出电流 = 1 ~ 100 mA	3.0	3.3	3.6	V
VBUS 检测输入门限		1.0	1.8	4.0	V
输入电流			12		μA
偏置电流	正常方式 (REGMOD = '0')		90	TBD	μA
	低功耗方式 (REGMOD = '1')		60	TBD	

C8051F320/1 全速 USB ISP FLASH 微控制器

图 8.1 REG0 配置: USB 总线供电

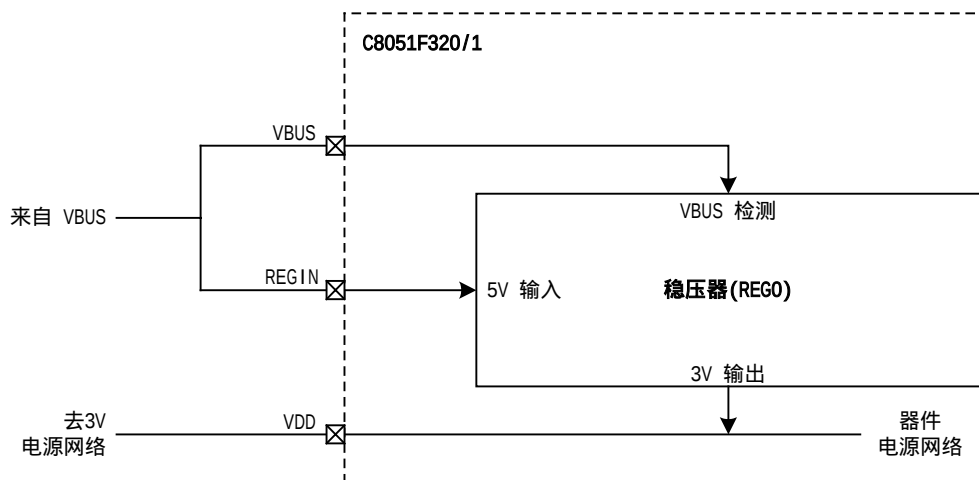
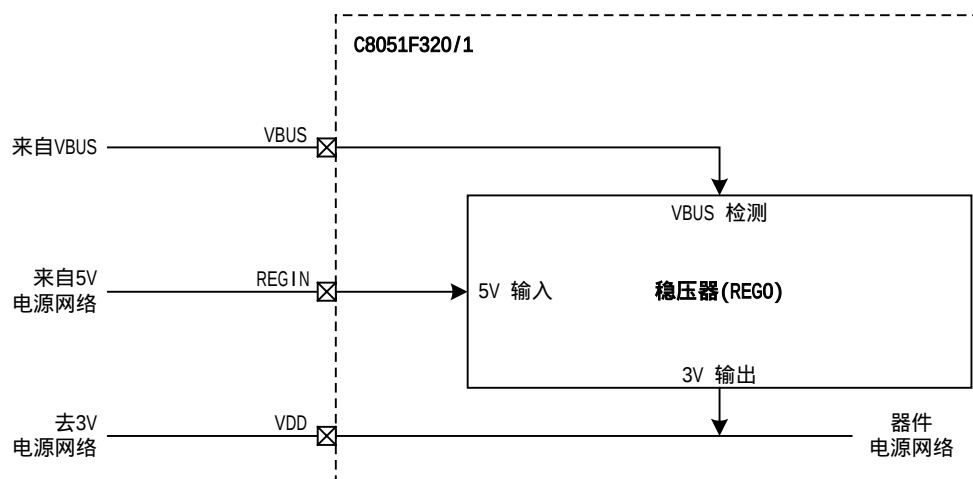


图 8.2 REG0 配置: USB 自供电



C8051F320/1 全速 USB ISP FLASH 微控制器

图 8.3 REG0 配置: USB 自供电, 稳压器禁止

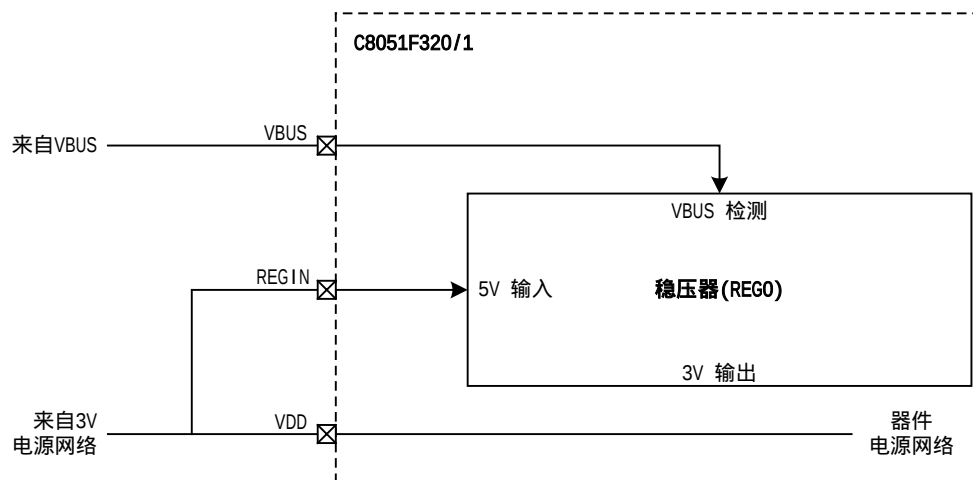
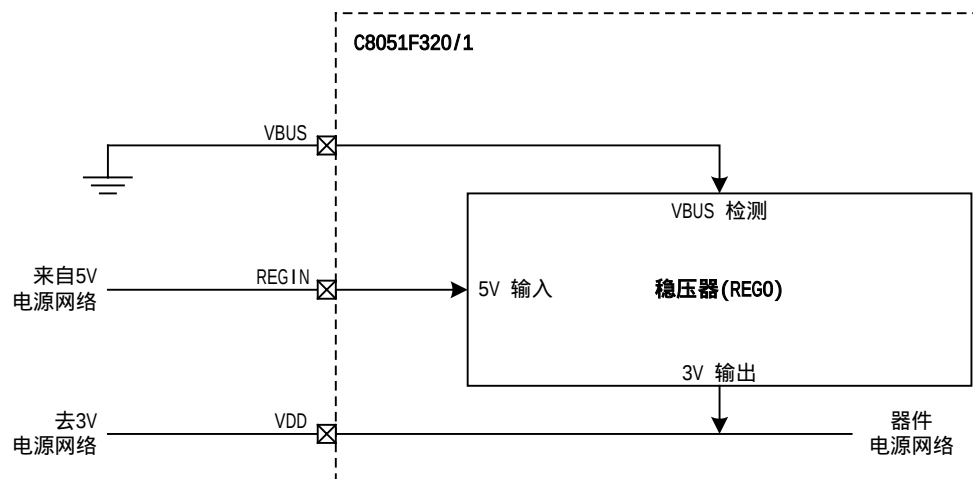


图 8.4 REG0 配置: 无 USB 连接



C8051F320/1 全速 USB ISP FLASH 微控制器

图 8.5 REG0CN: 稳压器控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
REGDIS	VBSTAT	VBPOL	REGMOD	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	SFR地址： 0xC9
位 7： REGDIS：稳压器禁止位 0：稳压器使能。 1：稳压器禁止。。 位 6： VBSTAT：VBUS 信号状态位 0：未检测到 VBUS 信号（器件未连接到 USB 网络）。 1：检测到 VBUS 信号（器件连接到 USB 网络）。 位 5： VBPOL：VBUS 中断极性选择位 该位选择 VBUS 中断极性。 0：当 VBUS 为低电平时 VBUS 中断有效。 1：当 VBUS 高低电平时 VBUS 中断有效。 位 4： REGMOD：稳压器方式选择位 该位选择稳压器工作方式。当 REGMOD 被置‘1’时，稳压器工作在低功耗（挂起）方式。 0：USB0 稳压器工作在正常方式。 1：USB0 稳压器工作在低功耗方式。 位 3-0： 保留。读 = 0000b。必须写 0000b。								

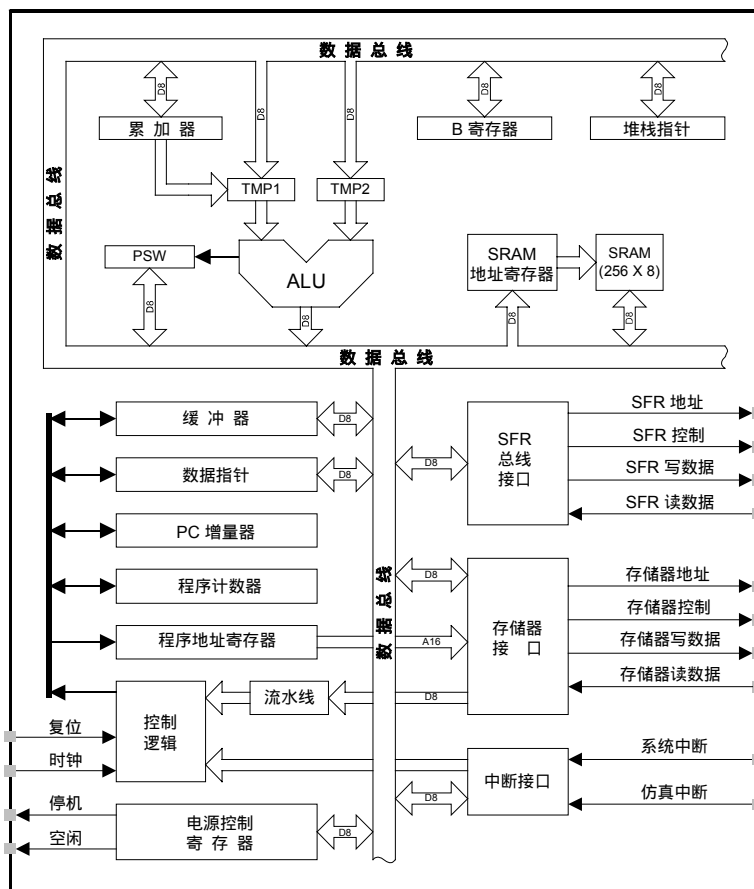
9. CIP-51 微控制器

MCU系统控制器的内核是CIP-51 微控制器。CIP-51 与MCS-51TM指令集完全兼容，可以使用标准 803x/805x的汇编器和编译器进行软件开发。该系列MCU具有标准 8051 的所有外设部件，包括 4 个 16 位的计数器/定时器（详见第 19 章）一个增强型全双工UART（详见第 17 章）一个增强型SPI（详见第 18 章）256 字节内部RAM、128 字节特殊功能寄存器（SFR）地址空间及 25 个端口I/O（详见第 14 章）。CIP-51 还包含片内调试硬件（详见第 21 章），与模拟和数字子系统直接接口，在一个集成电路内提供了完全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 9.1 的原理框图）。CIP-51 具有下列特点：

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

图 9.1 CIP-51 原理框图



C8051F320/1 全速 USB ISP FLASH 微控制器

性能

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

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

执行周期数	1	2	2/3	3	3/4	4	4/5	5	8
指令数	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。注意，可以在用户软件中使用 MOVC 和 MOVX 指令对可再编程的 FLASH 读和写，每次读或写一个字节。这一特性允许将程序存储器用于非易失性数据存储以及在软件控制下更新代码。

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，不需要额外的目标 RAM、堆栈、定时器或其它片内资源。有关 C2 接口的详细信息见“21. 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、宏汇编器、调试器和编程器。IDE 的调试器和编程器与 CIP-51 之间通过 C2 接口，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。

9.1 指令集

CIP-51 系统控制器的指令集与标准MCS-51TM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-51TM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。

9.1.1 指令和 CPU 时序

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

由于 CIP-51 采用了流水线结构，大多数指令执行所需的时钟周期数与指令的字节数一致。条件转移指令在不发生转移时的执行周期数比发生转移时少一个。表 9.1 给出了 CIP-51 指令一览表，包括每条指令的助记符、字节数和时钟周期数。

9.1.2 MOVX 指令和程序存储器

MOVC 指令通常用于访问外部数据存储器（注：C8051F320/1 不支持外部数据或程序存储器）。在 CIP-51 中，MOVC 指令可用于访问片内 XRAM 或片内 FLASH 程序存储器。这一特性为 CIP-51 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“11. FLASH 存储器”。

C8051F320/1 全速 USB ISP FLASH 微控制器

表 9.1 CIP-51 指令集

助记符	功能说明	字节数	时钟周期数
算术操作类指令			
ADD A,Rn	寄存器加到累加器	1	1
ADD A,direct	直接寻址字节加到累加器	2	2
ADD A,@Ri	间址 RAM 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	数据地址加 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

C8051F320/1 全速 USB ISP FLASH 微控制器

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

C8051F320/1 全速 USB ISP FLASH 微控制器

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-R1 间接寻址的数据 RAM 地址。

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

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

#data – 8 位立即数

#data16 – 16 位立即数

bit – 数据 RAM 或 SFR 中的直接寻址位

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

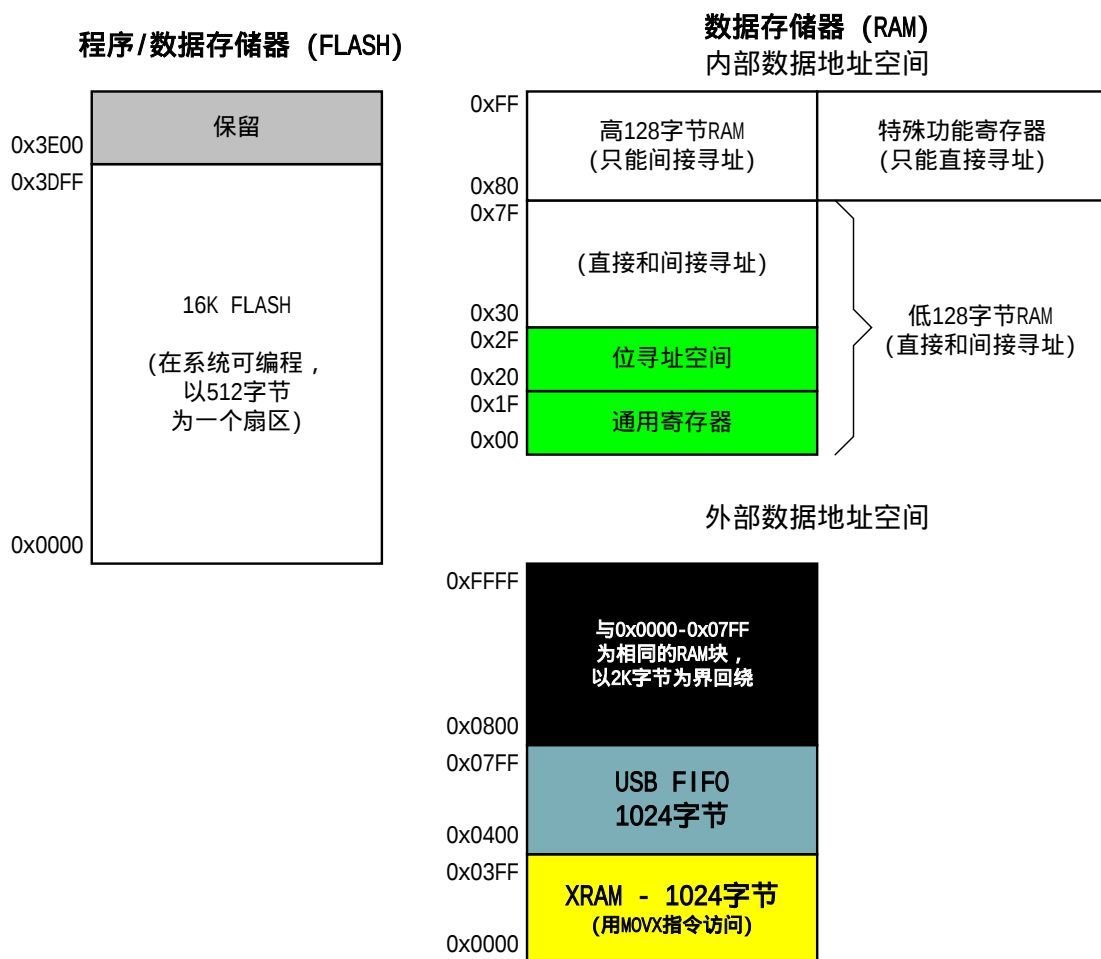
addr16 – LCALL 或 LJMP 使用的 16 位目的地址。目的地址可以是 8K 程序存储器空间内的任何位置。

有一个未使用的操作码（0xA5），它执行与 NOP 指令相同的功能。

9.2 存储器组织

CIP-51 系统控制器的存储器组织与标准 8051 的存储器组织类似。有两个独立的存储器空间：程序存储器和数据存储器。程序和数据存储器共享同一个地址空间，但用不同的指令类型访问。CIP-51 的存储器组织如图 9.2 所示。

图 9.2 程序存储器结构图



9.2.1 程序存储器

CIP-51 有 64KB 的程序存储器空间。C8051F320/1 在这个程序存储器空间中实现了 16KB 的可在系统编程的 FLASH 存储器，组织在一个连续的存储块内 (0x0000 – 0x3FFF)。注意：0x3DFF 以上的地址被保留。

程序存储器通常被认为是只读的。但是 CIP-51 可以通过设置程序存储写允许位 (PSCTL.0) 用 MOVX 指令对程序存储器写入。这一特性为 CIP-51 提供了更新程序代码和将程序存储器空间用于非易失性数据存储的机制。更详细的信息见“11. FLASH 存储器”。

9.2.2 数据存储器

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

数据存储器中的高 128 字节只能用间接寻址访问。该存储区与特殊功能寄存器 (SFR) 占据相同的地址空间,但物理上与 SFR 空间是分开的。当寻址高于 0x7F 的地址时,指令所用的寻址方式决定了 CPU 是访问数据存储器的高 128 字节还是访问 SFR。使用直接寻址方式的指令将访问 SFR 空间,间接寻址高于 0x7F 地址的指令将访问数据存储器的高 128 字节。图 9.2 给出了 CIP-51 数据存储器组织的示意图。

9.2.3 通用寄存器

数据存储器的低 32 字节,从地址 0x00 到 0x1F,可以作为 4 个通用寄存器区访问。每个区有 8 个 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 (见图 9.6 中关于 PSW 的说明)。这允许在进入子程序或中断服务程序时进行快速现场切换。间接寻址方式使用 R0 和 R1 作为间接寻址寄存器。

9.2.4 位寻址空间

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

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

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

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

9.2.5 堆栈

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

C8051F320/1 全速 USB ISP FLASH 微控制器

9.2.6 特殊功能寄存器

从 0x80 到 0xFF 的直接寻址存储器空间为特殊功能寄存器 (SFR)。SFR 提供对 CIP-51 的资源和外设的控制及 CIP-51 与这些资源和外设之间的数据交换。CIP-51 具有标准 8051 中的全部 SFR，还增加了一些用于配置和访问专有子系统的 SFR。这就允许在保证与 MCS-51™ 指令集兼容的前提下增加新的功能。表 9.2 列出了 CIP-51 系统控制器中的全部 SFR。

任何时刻用直接寻址方式访问 0x80~0xFF 的存储器空间将访问特殊功能寄存器 (SFR)。地址以 0x0 或 0x8 结尾的 SFR (例如 P0、TCON、P1、SCON、IE 等) 既可以按字节寻址也可以按位寻址，所有其它 SFR 只能按字节寻址。SFR 空间中未使用的地址保留为将来使用，访问这些地址会产生不确定的结果，应予避免。有关每个寄存器的详细说明请参见本数据表的相关部分 (表 9.3 中已标明)。

表 9.2 特殊功能寄存器 (SFR) 存储器映象

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	PCA0CPL4	PCA0CPH4	VDM0CN
F0	B	P0MDIN	P1MDIN	P2MDIN	P3MDIN		EIP1	EIP2
E8	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	RSTSRC
E0	ACC	XBR0	XBR1		IT01CF		EIE1	EIE2
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	
D0	PSW	REF0CN			P0SKIP	P1SKIP	P2SKIP	USB0XCEN
C8	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H		
C0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	
B8	IP	CLKMUL	AMX0N	AMX0P	ADC0CF	ADC0L	ADC0H	
B0		OSCXCEN	OSCICN	OSCICL			FLSCL	FLKEY
A8	IE	CLKSL	EMI0CN					
A0	P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	P3MDOUT
98	SCON0	SBUF0	CPT1CN	CPT0CN	CPT1MD	CPT0MD	CPT1MX	CPT0MX
90	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H	USB0ADR	USB0DAT
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH				PCON
	0(8) 可位寻址	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

C8051F320/1 全速 USB ISP FLASH 微控制器

表 9.3 特殊功能寄存器

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

寄存器	地址	说 明	页码
ACC	0xE0	累加器	
ADC0CF	0xBC	ADC0 配置寄存器	
ADC0CN	0xE8	ADC0 控制寄存器	
ADC0GTH	0xC4	ADC0 下限（大于）比较字高字节	
ADC0GTL	0xC3	ADC0 下限（大于）比较字低字节	
ADC0H	0xBE	ADC0 数据字高字节	
ADC0L	0xBD	ADC0 数据字低字节	
ADC0LTH	0xC6	ADC0 上限（小于）比较字	
ADC0LTL	0xC5	ADC0 上限（小于）比较字	
AMX0N	0xBA	AMUX0 负通道选择寄存器	
AMX0P	0xBB	AMUX0 正通道选择寄存器	
B	0xF0	B 寄存器	
CKCON	0x8E	时钟控制寄存器	
CLKSEL	0xA9	时钟选择寄存器	
CPT0CN	0x9B	比较器 0 控制寄存器	
CPT0MD	0x9D	比较器 0 方式选择寄存器	
CPT0MX	0x9F	比较器 0 MUX 选择寄存器	
CPT1CN	0x9A	比较器 1 控制寄存器	
CPT1MD	0x9C	比较器 1 方式选择寄存器	
CPT1MX	0x9E	比较器 1 MUX 选择寄存器	
DPH	0x83	数据指针高字节	
DPL	0x82	数据指针低字节	
EIE1	0xE6	扩展中断允许寄存器 1	
EIE2	0xE7	扩展中断允许寄存器 2	
EIP1	0xF6	扩展中断优先级寄存器 1	
EIP2	0xF7	扩展中断优先级寄存器 2	
EMI0CN	0xAA	外部存储器接口控制寄存器	
FLKEY	0xB7	FLASH 锁定和关键码寄存器	
FLSCL	0xB6	FLASH 存储器时序预分频器	
IE	0xA8	中断允许寄存器	
IP	0xB8	中断优先级寄存器	
IT01CF	0xE4	INT0/INT1 配置寄存器	
OSCICL	0xB3	内部振荡器校准寄存器	
OSICN	0xB2	内部振荡器控制寄存器	
OSCXCN	0xB1	外部振荡器控制寄存器	
P0	0x80	端口 0 锁存器	
P0MDIN	0xF1	端口 0 输入方式配置寄存器	
P0MDOUT	0xA4	端口 0 输出方式配置寄存器	
P0SKIP	0xD4	端口 0 跳过寄存器	

C8051F320/1 全速 USB ISP FLASH 微控制器

表 9.3 特殊功能寄存器（续）

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

寄存器	地址	说 明	页码
P1	0x90	端口 1 锁存器	
P1MDIN	0xF2	端口 1 输入方式配置寄存器	
P1MDOUT	0xA5	端口 1 输出方式配置寄存器	
P1SKIP	0xD5	端口 1 跳过寄存器	
P2	0xA0	端口 2 锁存器	
P2MDIN	0xF3	端口 2 输入方式配置寄存器	
P2MDOUT	0xA6	端口 2 输出方式配置寄存器	
P2SKIP	0xD6	端口 2 跳过寄存器	
P3	0xB0	端口 3 锁存器	
P3MDIN	0xF4	端口 3 输入方式配置寄存器	
P3MDOUT	0xA7	端口 3 输出方式配置寄存器	
PCA0CN	0xD8	PCA 控制寄存器	
PCA0CPH0	0xFC	PCA 捕捉模块 0 高字节	
PCA0CPH1	0xEA	PCA 捕捉模块 1 高字节	
PCA0CPH2	0xEC	PCA 捕捉模块 2 高字节	
PCA0CPH3	0xEE	PCA 捕捉模块 3 高字节	
PCA0CPH4	0xFE	PCA 捕捉模块 4 高字节	
PCA0CPL0	0xFB	PCA 捕捉模块 0 低字节	
PCA0CPL1	0xE9	PCA 捕捉模块 1 低字节	
PCA0CPL2	0xEB	PCA 捕捉模块 2 低字节	
PCA0CPL3	0xED	PCA 捕捉模块 3 低字节	
PCA0CPL4	0xFD	PCA 捕捉模块 4 低字节	
PCA0CPM0	0xDA	PCA 模块 0 方式寄存器	
PCA0CPM1	0xDB	PCA 模块 1 方式寄存器	
PCA0CPM2	0xDC	PCA 模块 2 方式寄存器	
PCA0CPM3	0xDD	PCA 模块 3 方式寄存器	
PCA0CPM4	0xDE	PCA 模块 4 方式寄存器	
PCA0H	0xFA	PCA 计数器高字节	
PCA0L	0xF9	PCA 计数器低字节	
PCA0MD	0xD9	PCA 方式寄存器	
PCON	0x87	电源控制寄存器	
PSCTL	0x8F	程序存储读/写控制寄存器	
PSW	0xD0	程序状态字	
REF0CN	0xD1	电压基准控制寄存器	
RSTSRC	0xEF	复位源寄存器	
SBUF0	0x99	UART0 数据缓冲器	
SCON0	0x98	UART0 控制寄存器	
SMB0CF	0xC1	SMBus 配置寄存器	
SMB0CN	0xC0	SMBus 控制寄存器	
SMB0DAT	0xC2	SMBus 数据寄存器	

C8051F320/1 全速 USB ISP FLASH 微控制器

表 9.3 特殊功能寄存器（续）

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

寄存器	地址	说 明	页码
SP	0x81	堆栈指针	
SPI0CFG	0xA1	SPI 配置寄存器	
SPI0CKR	0xA2	SPI 时钟频率控制寄存器	
SPI0CN	0xF8	SPI 控制寄存器	
SPI0DAT	0xA3	SPI 数据寄存器	
TCON	0x88	计数器/定时器控制寄存器	
TH0	0x8C	计数器/定时器 0 高字节	
TH1	0x8D	计数器/定时器 1 高字节	
TL0	0x8A	计数器/定时器 0 低字节	
TL1	0x8B	计数器/定时器 1 低字节	
TMOD	0x89	计数器/定时器方式寄存器	
TMR2CN	0xC8	计数器/定时器 2 控制寄存器	
TMR2H	0xCD	计数器/定时器 2 高字节	
TMR2L	0xCC	计数器/定时器 2 低字节	
TMR2RLH	0xCB	计数器/定时器 2 重载值高字节	
TMR2RLL	0xCA	计数器/定时器 2 重载值低字节	
TMR3CN	0x91	计数器/定时器 3 控制寄存器	
TMR3H	0x95	计数器/定时器 3 高字节	
TMR3L	0x94	计数器/定时器 3 低字节	
TMR3RLH	0x93	计数器/定时器 3 重载值高字节	
TMR3RLL	0x92	计数器/定时器 3 重载值低字节	
VDM0CN	0xFF	VDD 监视器控制寄存器	
XBR0	0xE1	端口 I/O 交叉开关控制 0	
XBR1	0xE2	端口 I/O 交叉开关控制 1	

C8051F320/1 全速 USB ISP FLASH 微控制器

9.2.7 寄存器说明

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

图 9.3 DPL：数据指针低字节

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

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

图 9.4 DPH：数据指针高字节

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

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

图 9.5 SP：堆栈指针

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	SFR地址： 0x81

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

C8051F320/1 全速 USB ISP FLASH 微控制器

图 9.6 PSW : 程序状态字

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

位 7：

CY：进位标志。

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

位 6：

AC：辅助进位标志。

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

位 5：

F0：用户标志 0。

这是一个可位寻址、受软件控制的通用标志位。

位 4-3：

RS1-RS0：寄存器区选择。

该两位在寄存器访问时用于选择寄存器区。

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

位 2：

OV：溢出标志。

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。

若累加器中 8 个位的和为奇数时该位置 1，为偶数时清 0。

图 9.7 ACC：累加器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：
							(可位寻址)	0xE0
位 7-0： ACC：累加器								
该寄存器为算术操作的累加器。								

图 9.8 B：B 寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：
							(可位寻址)	0xF0
位 7-0： B：B 寄存器								
该寄存器为某些算术操作的第二累加器。								

9.3 中断系统

CIP-51 包含一个扩展的中断系统，支持 16 个中断源，每个中断源有两个优先级。中断源在片内外设与外部输入引脚之间的分配随器件的不同而变化。每个中断源可以在一个 SFR 中有一个或多个中断标志。当一个外设或外部源满足有效的中断条件时，相应的中断标志被置为逻辑‘1’。

如果一个中断源被允许，则在中断标志被置位时将产生一个中断。一旦当前指令执行完，CPU 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。

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

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

9.3.1 MCU 中断源和中断向量

MCU 支持 16 个中断源。软件可以通过将任何一个中断标志设置为逻辑‘1’来模拟一个中断。如果中断标志被允许，系统将产生一个中断请求，CPU 将转向与该中断标志对应的 ISR 地址。表 9.4 给出了 MCU 中断源、对应的向量地址、优先级和控制位一览表。关于外设有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。

9.3.2 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组合。

IT0	IN0PL	/INT0 中断	IT1	IN1PL	/INT1 中断
1	0	低电平有效，边沿触发	1	0	低电平有效，边沿触发
1	1	高电平有效，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	0	低电平有效，电平触发
0	1	高电平有效，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（见图 9.15）。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“14.1 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 时将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响应,在 ISR 返回前必须使该中断请求无效,否则将产生另一个中断请求。

9.3.3 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。每个中断在 SFR (IP 或 EIP1、EIP2) 中都有一个配置其优先级的中断优先级设置位,缺省值为低优先级。如果两个中断同时发生,具有高优先级的中断先得到服务。如果这两个中断的优先级相同,则由固定的优先级顺序(见表 9.4)决定哪一个中断先得到服务。

9.3.4 中断响应时间

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

注意:在 FLASH 写/擦除操作期间和 USB FIFO MOVX 访问(见“12.2 访问 USB FIFO 空间”)期间,CPU 暂停执行指令。对于在 CPU 暂停执行指令期间发生的中断,中断服务响应时间将延长。这些情况下的中断延迟时间由标准中断服务响应过程(如前所述)和 CPU 暂停执行指令的时间决定。

C8051F320/1 全速 USB ISP FLASH 微控制器

表 9.4 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) RXOV RN(SPI0CN.4)	Y	N	ESPI0 (IE.6)	PSPI0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
USB0	0x0043	8	特殊	N	N	EUSB0 (EIE1.1)	PUSB0 (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	EADC0C (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)
VBUS 电平	0x007B	15	N/A	N/A	N/A	EVBUS (EIE2.0)	PVBUS (EIP2.0)

9.3.5 中断寄存器说明

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

图 9.9 IE：中断允许寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： (可位寻址) 0xA8
位 7： EA：允许所有中断。 该位允许 / 禁止所有中断。它超越所有的单个中断屏蔽设置。 0：禁止所有中断源。 1：开放中断。每个中断由它对应的中断屏蔽设置决定。 位 6： 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								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 9.10 IP：中断优先级寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	PSPI0	PT2	PS0	PT1	PX1	PT0	PX0	10000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xB8
(可位寻址)								
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：UART1 为高优先级。 位 3： PT1：定时器 1 中断优先级控制 该位设置定时器 1 中断的优先级。 0：定时器 1 为低优先级。 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。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 9.11 EIE1：扩展中断允许 1

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	EUSB0	ESMB0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xE6
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： EADC0C：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： EUSB0：USB0 中断允许位 该位设置 USB0 的中断屏蔽。 0：禁止 USB0 中断。 1：允许 USB0 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 的中断请求。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 9.12 EIP1 : 扩展中断优先级 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
PT3	PCP1	PCP0	PPCA0	PADC0	PWADC0	PUSB0	PSMB0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xF6
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 : PUSB0 : USB0 中断优先级控制 该位设置 USB0 中断的优先级。 0 : USB0 中断为低优先级。 1 : USB0 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。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 9.13 EIE2：扩展中断允许 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	-	-	-	EVBUS	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xE7
位 7-1：未使用。读 = 0000000b，写 = 忽略。								
位 0：EVBUS：VBUS 电平中断允许位								
该位设置 VBUS 的中断屏蔽。								
0：禁止所有 VBUS 中断。								
1：允许 VBUS 电平检测电路的中断请求。								

图 9.14 EIP2：扩展中断优先级 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	-	-	-	PVBUS	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xF7
位 7-1：未使用。读 = 0000000b，写 = 忽略。								
位 0：PVBUS：VBUS 电平中断优先级控制								
该位设置 VBUS 中断的优先级。								
0：VBUS 中断为低优先级。								
1：VBUS 中断为高优先级。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 9.15 IT01CF : INT0/INT1 配置寄存器

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	SFR地址： 0xE4

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 : IN1SL2-0 : /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 POSKIP 中的对应位置‘1’来实现），则该引脚将不会被分配给外设。

IN1SL2-0	/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 : IN0SL2-0 : /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 POSKIP 中的对应位置‘1’来实现），则该引脚将不会被分配给外设。

IN0SL2-0	/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

9.4 电源管理方式

CIP-51 有两种可软件编程的电源管理方式：空闲和停机。在空闲方式，CPU 停止运行，而外设和时钟处于活动状态。在停机方式，CPU 停止运行，所有的中断都处于非活动状态，内部振荡器被停止（模拟外设保持在所选择的状态；外部振荡器不受影响）。由于在空闲方式下时钟仍然运行，所以功耗与进入空闲方式之前的系统时钟频率和处于活动状态的外设数目有关。停机方式消耗最少的功率。图 9.16 对用于控制 CIP-51 电源管理方式的电源控制寄存器作出了说明。

虽然 CIP-51 具有空闲和停机方式（与任何标准 8051 结构一样），但通过管理系统时钟和每个外设，可以使整个 MCU 的功耗最小。每个模拟外设在不使用时都可以被禁止，使其进入低功耗方式。象定时器、串行总线这样的数字外设在不使用时消耗很少的功率。关闭振荡器可以大大降低消耗功率，但需要复位来重新启动 MCU。

内部振荡器可以被置于挂起方式（见“13. 振荡器”）。在挂起方式，内部振荡器停止运行，直到检测到一个非空闲 USB 事件或 VBUS 输入信号与 VBPOL 位（位于寄存器 REG0CN）所选择的极性匹配。

9.4.1 空闲方式

将空闲方式选择位（PCON.0）置 1 导致 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（RET1）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。

如果被允许，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禁止 WDT。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，请参见“10.6 PCA 看门狗定时器复位”。

9.4.2 停机方式

将停机方式选择位（PCON.1）置 1 导致 CIP-51 进入停机方式，在执行完对该位置 1 的指令后 MCU 立即进入停机方式。在停机方式，内部振荡器、CPU 和所有的数字外设都停止工作，但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。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 9.14 PCON : 电源控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
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	SFR地址： 0x87
位 7-2 : GF5-GF0 : 通用标志位 5-0。 这些位是用作软件控制的通用标志位。 位 1 : STOP : 停机方式选择。 将该位置 ‘ 1 ’ 使 CIP-51 进入停机方式。该位读出值总是为 0。 1 : 进入掉电方式 (内部振荡器停止运行) 位 0 : IDLE : 空闲方式选择。 将该位置 ‘ 1 ’ 使 CIP-51 进入空闲方式。该位读出值总是为 0。 1 : CPU 进入空闲方式。(关闭供给 CPU 的时钟信号 , 但定时器、中断、串口和模拟外设保持活动状态。)								

10. 复位源

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

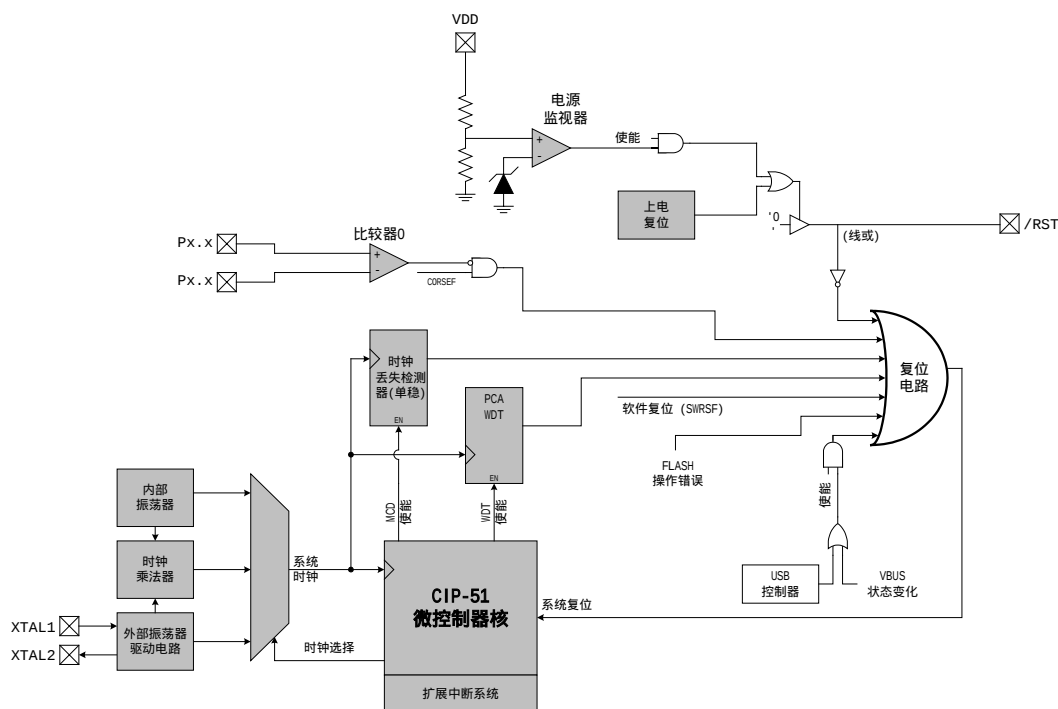
- CIP-51 停止程序执行
- 特殊功能寄存器（SFR）被初始化为所定义的复位值
- 外部端口引脚被置于一个已知状态
- 中断和定时器被禁止。

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

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“13. 振荡器”。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“20.3 看门狗定时器方式”）。程序从地址 0x0000 开始执行。

图 10.1 复位源框图



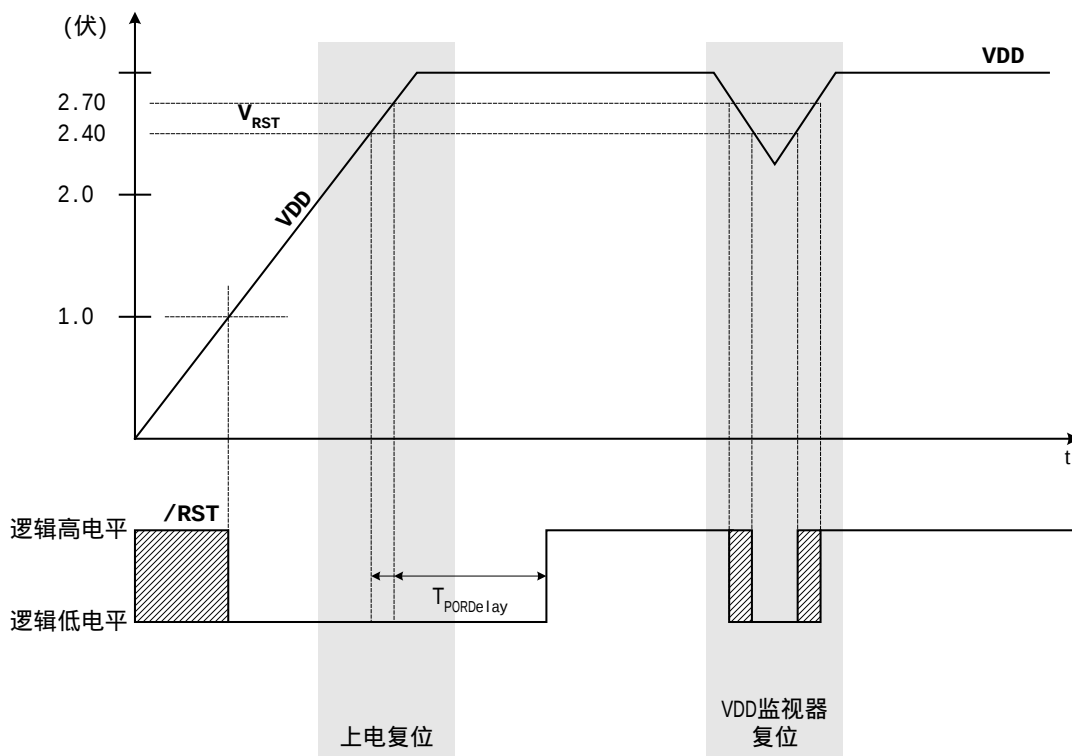
10.1 上电复位

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VDD上升到超过 V_{RST} 电平。从复位开始到退出复位状态要经过一个上电复位延时（ T_{PORDelay} ）；该延时通常小于0.3ms。图 10.2 给出了上电复位和VDD监视器复位时序。

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，内部数据存储单元中的内容应被认为是不确定的。在上电复位后，VDD 监视器被使能。

软件可以通过向寄存器 RSTSRC 中的 PINRSF 位写‘1’来强制产生一次上电复位。

图 10.2 上电复位和 VDD 监视器复位时序



10.2 掉电复位和 VDD 监视器

当发生掉电或因电源波动导致VDD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 10.2）。当VDD又回到高于 V_{RST} 的电平时，CIP-51 将退出复位状态。注意，尽管内部数据存储器的内容可能没有因掉电复位而发生改变，但无法确定VDD是否降到了数据保持所要求的最低电平以下。如果PORSF标志的读出值为 1，则内部RAM的数据可能不再有效。在上电复位后VDD监视器被禁止，但它的状态（使能/禁止）不受任何其它复位源的影响。例如，在VDD监视器被使能后执行一次软件复位，复位后VDD监视器仍然为使能状态。

注意：在选择 VDD 监视器作为复位源之前，必须使能 VDD 监视器。在 VDD 监视器被使能或稳定之前选其为复位源可能导致系统复位。将 VDD 监视器配置为复位源的步骤如下：

1. 使能 VDD 监视器（VDM0CN.7 = 1）；
2. 等待 VDD 监视器稳定（见表 10.1 的 VDD 监视器启动时间）；
3. 选择 VDD 监视器作为复位源（RSTSR.1 = 1）。

图 10.2 给出了 VDD 监视器的时序。表 10.1 给出了 VDD 监视器的电气特性。

图 10.3 VDM0CN：VDD 监视器控制寄存器

R/W	R	R	R	R	R	R	R	复位值
VDMEN	VDDSTAT	保留	保留	保留	保留	保留	保留	可变
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xFF
位 7： VDMEN：VDD 监视器使能位。 该位控制 VDD 监视器电源的通断。VDD 监视器在被选择为复位源之前不能产生系统复位。在被选择为复位源之前，VDD 监视器必须稳定。在 VDD 监视器稳定之前选其为复位源可能导致系统复位。VDD 监视器的最小启动时间见表 10.1。在上电复位后 VDD 监视器被使能。 0：禁止 VDD 监视器。 1：使能 VDD 监视器。 位 6： VDDSTAT：VDD 状态。 该位指示当前电源状态（VDD 监视器输出）。 0：VDD 等于或低于 VDD 监视器阈值。 1：VDD 高于 VDD 监视器阈值 位 5-0： 保留。读 = 00000b，写 = 忽略。								

10.3 外部复位

外部/RST 引脚提供了使用外部电路强制 MCU 进入复位状态的手段。在/RST 引脚上加一个低电平有效信号将导致 MCU 进入复位状态。尽管在内部有弱上拉,但最好能提供一个外部上拉和/或对/RST 引脚去耦以防止强噪声引起复位。从外部复位状态退出后,PINRSF 标志(RSTSRC.0)被置'1'。

10.4 时钟丢失检测器复位

时钟丢失检测器(MCD)实际上是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。在发生时钟丢失检测器复位后,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'将使能时钟丢失检测器;写'0'将禁止时钟丢失检测器。/RST 引脚的状态不受该复位的影响。

10.5 比较器 0 复位

向 CORSEF 标志(RSTSRC.5)写'1'可以将比较器 0 配置为复位源。应在写 CORSEF 之前使能比较器 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-),则产生系统复位。在发生比较器 0 复位后,CORSEF 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'。/RST 引脚的状态不受该复位的影响。

10.6 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“20.3 看门狗定时器方式”)。在每次复位后,WDT 被使能并使用 SYSCLK/12 作为计数时钟。如果因系统出错使用户软件不能更新 WDT,则 WDT 将产生一次复位,WDTRSF 位(RSTSRC.5)被置'1'。/RST 引脚的状态不受该复位的影响。

10.7 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:

- 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 操作的地址大于 0x3DFF。
- FLASH 读地址超出了用户代码空间,即 MOVC 操作的地址大于 0x3DFF。
- 程序读超出了用户代码地址空间。这种情况发生在用户代码试图转移到大于 0x3DFF 的地址时。
- 当 FLASH 读、写或擦除被安全设置禁止时(见 11.3 安全选项)。

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

10.8 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。

10.9 USB 复位

向寄存器 RSTSRC 中的 USBRSF 位写 ‘1’ 将选择 USB0 为复位源。在 USB0 被选择为复位源的情况下，下面的任何事件都会导致产生系统复位。

1. 在 USB 网络上检测到 RESET 命令。USB 功能控制器 (USB0) 必须被使能为检测 RESET 命令。有关 USB 功能控制器的详细信息见 “15. 通用串行总线控制器 (USB0)”。

2. VBUS 引脚的电压与 VBPOL 位(位于寄存器 REG0CN)选择的极性匹配。有关 VBUS 检测电路的详细信息见 “8. 稳压器 (REG0)”。

发生 USB 复位后，USBRSF 位的读出值为 ‘1’。/RST 引脚的状态不受该复位的影响。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 10.4 RSTSRC : 复位源寄存器

R/W	R	R/W	R/W	R	R/W	R/W	R	复位值
USBRSF	FERROR	CORSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF	可变
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址 : 0xEF
位 7 :	USBRSF : USB 复位标志							
	0 : 读 : 最后一次复位不是来自 USB ; 写 : 禁止 USB 复位。							
	1 : 读 : 最后一次复位是来自 USB ; 写 : 允许 USB 复位。							
位 6 :	FERROR : FLASH 错误标志。							
	0 : 最后一次复位不是来自 FLASH 读/写/擦除错误。							
	1 : 最后一次复位是由于 FLASH 读/写/擦除错误。							
位 5 :	CORSEF : 比较器 0 复位使能和标志							
	写							
	0 : 比较器 0 不是复位源。							
	1 : 比较器 0 是复位源 (低电平有效)。							
	读							
	0 : 最后一次复位不是来自比较器 0。							
	1 : 最后一次复位来自比较器 0。							
位 4 :	SWRSF : 软件强制复位和标志							
	写							
	0 : 无作用							
	1 : 强制产生一次系统复位。							
	读							
	0 : 最后一次复位不是来自写 SWRSF 位。							
	1 : 最后一次复位来自写 SWRSF 位。							
位 3 :	WDTRSF : 看门狗定时器复位标志							
	0 : 最后一次复位不是来自 WDT 超时。							
	1 : 最后一次复位来自 WDT 超时。							
位 2 :	MCDRSF : 时钟丢失检测器标志							
	写							
	0 : 禁止时钟丢失检测器。							
	1 : 使能时钟丢失检测器 ; 检测到时钟丢失条件时触发复位。							
	读							
	0 : 最后一次复位不是来自时钟丢失检测器超时。							
	1 : 最后一次复位来自时钟丢失检测器超时。							
位 1 :	PORSF : 上电复位/VDD 监视器复位标志							
	该位在上电复位后被置位。对该位写入可以使能/禁止 VDD 监视器作为复位源。 注意 : 在 VDD 监视器被使能和稳定之前向该位写 1 可能导致系统复位。 见寄存器 VDM0CN(图 10.3)。							
	写							
	0 : 禁止 VDD 监视器为复位源。							
	1 : 使能 VDD 监视器为复位源。							
	读							
	0 : 最后一次复位不是上电复位或 VDD 监视器复位。							
	1 : 最后一次复位是上电或 VDD 监视器复位 , 所有其它复位标志是不确定的。							
位 0 :	PINRSF : 硬件引脚复位标志							
	0 : 最后一次复位不是来自/RST 引脚。							
	1 : 最后一次复位来自/RST 引脚。							

注 : 对于既作为复位源使能 (写) 又作为复位指示标志 (读) 的位 , 读-修改-写指令只读和修改复位源使能位。这中情况适用于 USBRSF、CORSEF、SWRSF、MCDRSF 和 PORSF。

C8051F320/1 全速 USB ISP FLASH 微控制器

表 10.1 复位源电气特性

VDD = 2.7V ~ 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	$I_{OL} = 8.5 \text{ mA}, V_{DD} = 2.7 - 3.6 \text{ V}$			0.6	V
/RST 输入高电平		$0.7 \times V_{DD}$			V
/RST 输入低电平				$0.3 \times V_{DD}$	V
/RST 输入上拉电流	/RST=0.0V		25	40	μA
VDD 上电复位门限 (V_{RST})		2.40	2.55	2.70	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	100	220	500	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	5.0			μs
产生系统复位的最小 /RST 低电平时间		15			μs
VDD 监视器启动时间		100			μs
VDD 监视器电源电流			20	50	μA

11. FLASH 存储器

C8051F32x 内部有 16KB 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 指令对 FLASH 存储器进行在系统编程，每次一个字节。一个 FLASH 位一旦被清‘0’，必须经过擦除才能再回到‘1’状态。在进行重新编程之前，一般要将数据字节擦除（置为 0xFF）。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，程序暂停执行。表 11.1 给出了 FLASH 存储器的电气特性。

11.1 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“21. C2 接口”。

为了保证 FLASH 内容的正确性，强烈建议在使用用户软件对 FLASH 存储器进行写和/或擦除操作的系统中使能片内 VDD 监视器。

11.1.1 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。图 11.3 给出了 FLKEY 寄存器的详细说明。

11.1.2 FLASH 擦除

可以用软件使用 MOVX 指令对 FLASH 存储器编程，象一般的操作数一样为 MOVX 指令提供待编程的地址和数据字节。在使用 MOVX 指令对 FLASH 存储器写入之前，必须先允许 FLASH 写操作。允许 FLASH 写操作的过程是：1）按顺序向 FLASH 锁定寄存器（FLKEY）写入 FLASH 关键码；2）将程序存储写允许位 PSWE（PSCCTL.0）设置为逻辑‘1’（这将使 MOVX 操作指向目标 FLASH 存储器）。PSWE 位将保持置位状态，直到被软件清除。

写 FLASH 存储器可以清除数据位，但不能使数据位置‘1’，只有擦除操作能将 FLASH 中的数据位置‘1’。所以在写入新值之前，必须先擦除待编程的地址。FLASH 存储器是以 512 字节的扇区为单位组织的，一次擦除操作将擦除整个扇区（将扇区内的所有字节置为 0xFF）。擦除一个扇区（页）的步骤如下：

1. 禁止中断（建议这样做）。
2. 向 FLKEY 写第一个关键码：0xA5。
3. 向 FLKEY 写第二个关键码：0xF1。
4. 置‘1’PSEE 位（寄存器 PSCCTL），以允许 FLASH 扇区擦除。
5. 置‘1’PSWE 位（寄存器 PSCCTL），以允许 FLASH 写入。
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 位 (寄存器 PSCTL)。
8. 清除 PSEE 位 (寄存器 PSCTL)。

11.1.3 FLASH 写

用软件对 FLASH 字节编程的步骤如下：

1. 禁止中断 (建议这样做)。
2. 擦除包含目标地址的 FLASH 页。
3. 向 FLKEY 写第一个关键码：0xA5。
4. 向 FLKEY 写第二个关键码：0xF1。
5. 置 ‘ 1 ’ PSCTL 中的 PSWE 位。
6. 清除 PSCTL 中的 PSEE 位。
7. 用 MOVX 指令向扇区内的目标地址写入一个数据字节。
8. 清除 PSWE 位。

重复步骤 3-8，直到写完每个字节。在完成了对 FLASH 的数据写入后，PSWE 位应被清 ‘ 0 ’，以使 MOVX 指令不再指向程序存储器。

表 11.1 FLASH 存储器电气特性

参 数	条 件	最小值	典型值	最大值	单 位
FLASH 规模	C8051F320/1	16384*			字节
擦写寿命		20k	100k		擦/写
擦除时间	25 MHz 系统时钟	10	15	20	ms
写入时间	25 MHz 系统时钟	40	55	70	μs

*注：位于 0x3E00 ~ 0x3FFF 的 512 字节保留。

11.2 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 指令，读出时用 MOVC 指令。注意：MOVX 读指令总是指向 XRAM。

尽管 FLASH 存储器可以每次写一个字节，但必须首先擦除整个扇区。为了修改一个多字节数据集中的某一个字节，整个数据集必须被保存到一个临时存储区。接下来将扇区擦除，更新数据集，最后将数据集写回到原扇区。

11.3 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) 和程序存储器擦除允许 (PSCTL 寄存器中的 PSEE) 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 必须被置为逻辑 '1'; 在用软件擦除 FLASH 存储器之前, PSWE 位和 PSEE 位都必须被置为逻辑 '1'。此外, CIP-51 还提供了可以防止通过 C2 接口读取产权程序代码和常数这一安全功能。

保存在 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 ~ 0x01FF) 开始锁定 n 个 512 字节的 FLASH 页, 其中 n 是安全锁定字节的反码。注意: 当任何一个其他 FLASH 页被锁定时, 包含 FLASH 安全锁定字节的页也被锁定。见下面的例子。

安全锁定字节: 11111101 b
反码: 00000010 b

被锁定的 FLASH 页: 2

被锁定的地址: 0x0000 - 0x03FF

有关 FLASH 安全的重要注意事项:

1. 对安全锁定字节的任何位清 0 将锁定所选择的页及包含锁定字节的 FLASH 页;
2. 不能通过 C2 接口对被锁定的页进行读、写或擦除操作;
3. 位于未被锁定的存储器空间的用户固件不能对被锁定的页进行读、写或擦除操作;
4. 位于被锁定的页内的用户固件可以对被锁定的和未被锁定的任何页进行读和写操作, 保留区除外;
5. 位于被锁定的页内的用户固件可以擦除任何被锁定的和未被锁定的页, 保留区和包含锁定字节的页除外;
6. 被锁定的页只能通过 C2 接口用器件擦除命令解锁;
7. 如果用户固件的 FLASH 访问操作被禁止 (根据上面的 3、4、5 条), 将产生 FLASH 错误系统复位。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 11.1 FLASH 程序存储器结构和安全字节

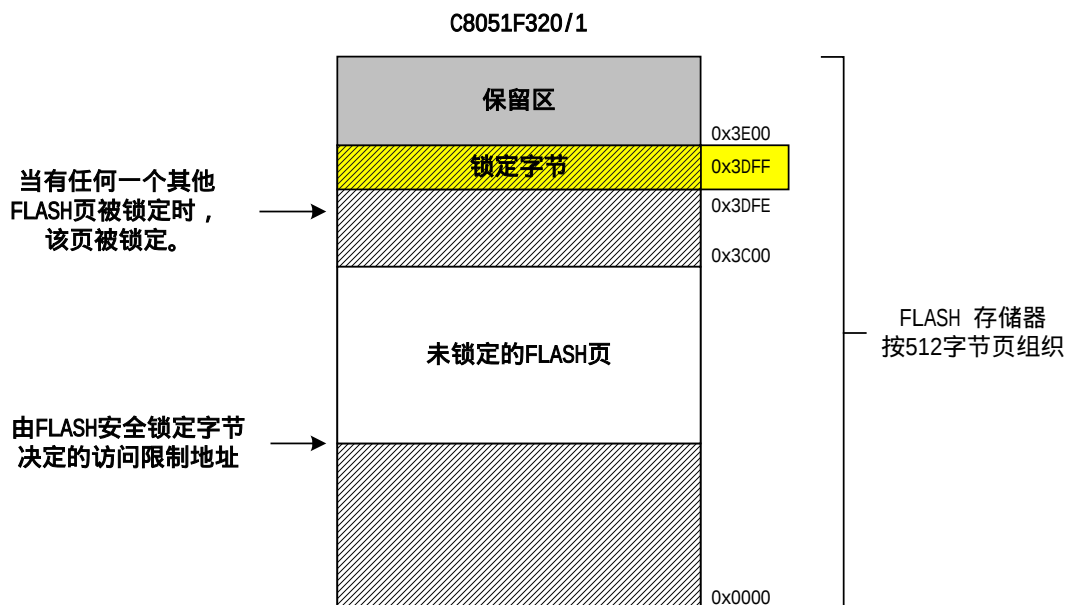


图 11.2 PSCTL：程序存储读写控制

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

位 7-3：未使用。读 = 00000b，写 = 忽略。

位 2：保留。读 = 0b，必须写 0b。

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'）。在将该位置 '1' 后，用 MOVX 指令进行一次写操作将擦除包含 MOVX 指令寻址地址的那个 FLASH 页。用于写操作的数据可以是任意值。
0：禁止擦除 FLASH 存储器。
1：允许擦除 FLASH 存储器。

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

C8051F320/1 全速 USB ISP FLASH 微控制器

图 11.3 FLKEY : FLASH 锁定和关键码寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址 : 0xB7
位 7-0 : FLKEY : FLASH 锁定和关键码寄存器								
写 :								
在进行 FLASH 擦除和写操作之前必须写该寄存器。写操作的时间并不重要，但必须按顺序写。每次写或擦除操作都必须写关键码。如果写 FLKEY 操作不正确或在正确写入关键码之前进行了 FLASH 操作，则 FLASH 将被锁定，直到下一次系统复位。								
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。								

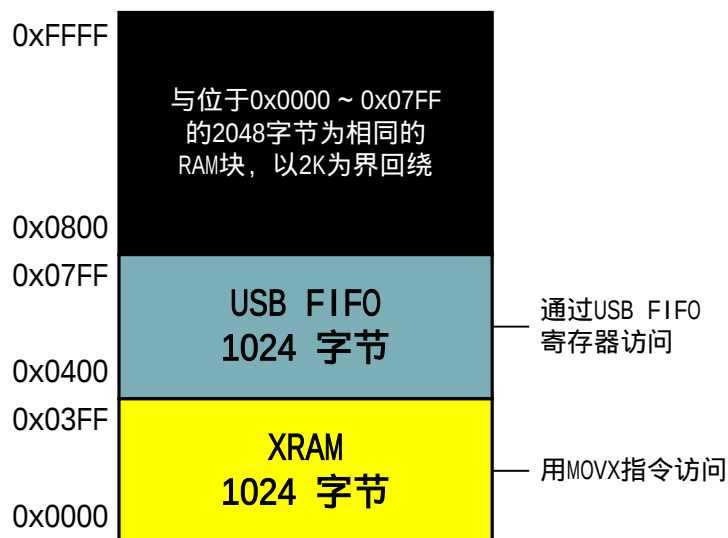
图 11.4 FLSCL : FLASH 定时预分频

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
FOSE	保留	保留	保留	保留	保留	保留	保留	10000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址 : 0xB6
位 7 : FOSE : FLASH 单稳态定时器使能								
该位用于使能 FLASH 读操作单稳态定时器。当 FLASH 单稳态定时器被禁止时，FLASH 读出放大器在整个 FLASH 读周期内被使能。当系统时钟频率低于 10MHz 时，禁止 FLASH 单稳态定时器会增加系统功耗。								
0 : 禁止 FLASH 单稳态定时器。								
1 : 使能 FLASH 单稳态定时器。								
位 6-0 : 保留。读 = 0，必须写 0。								

12 . 外部 RAM

C8051F320/1 器件内部有 2048 字节的 XRAM。该 XRAM 空间被分成用户 RAM (地址为 0x0000 ~ 0x03FF) 和 USB FIFO 空间 (地址为 0x0400 ~ 0x07FF)。

图 12.1 外部数据存储器组织



12.1 访问用户 XRAM

可以用外部传送指令 (MOVX) 和数据指针 (DPTR) 访问 XRAM , 或者用 MOVX 间接寻址方式。如果 MOVX 指令使用一个 8 位地址操作数 (例如 @R1), 则 16 位地址的高字节由外部存储器接口控制寄存器 (EMI0CN , 如图 12.3 所示) 提供。注 : MOVX 指令还用于写 FLASH 存储器 , 详见 “ 11. FLASH 存储器 ”。缺省情况下 MOVX 指令访问 XRAM。

对于任何一种寻址方式 , 16 位外部数据存储器地址的高 5 位是被 “ 忽略 ” 的。因此 , 这个 2048 字节的 RAM 以取模的方式映射到整个 64KB 的外部数据存储器地址范围。例如 , 位于地址 0x0000 的 XRAM 字节也位于 0x0800、0x1000、0x1800、0x2000 等地址。在进行线性存储器填充时这是一个很有用的特性 , 因为在达到 RAM 块的边界时不必对地址指针复位。

注意 : 在 2KB XRAM 中 , 位于高地址的 1KB 为 USB FIFO 空间。有关访问该存储器空间的详细信息见 12.2。

12.2 访问 USB FIFO

位于 XRAM 高地址的 1KB 为 USB FIFO 空间。图 12.2 给出了 FIFO 空间和用户 XRAM 的扩展示意图。通过 USB FIFO 寄存器访问 FIFO 空间 ,有关访问 FIFO 空间的详细信息见“ 15.5 FIFO 管理 ”。不应用 MOVX 指令装载或修改位于 FIFO 空间的 USB 数据。

FIFO 空间中未使用的区域可被用作通用 XRAM，可以按 12.1 节所述对其访问。FIFO 存储块工作在 USB 时钟域，因此在访问 FIFO 空间时，USB 时钟必须处于活动状态。注意：在访问 USB FIFO 空间时，MOVX 指令所需要的 SYSCLK 周期数增加。

注意：在访问 FIFO 空间时，USB 时钟必须处于活动状态。

图 12.2 XRAM 存储器扩展示意图

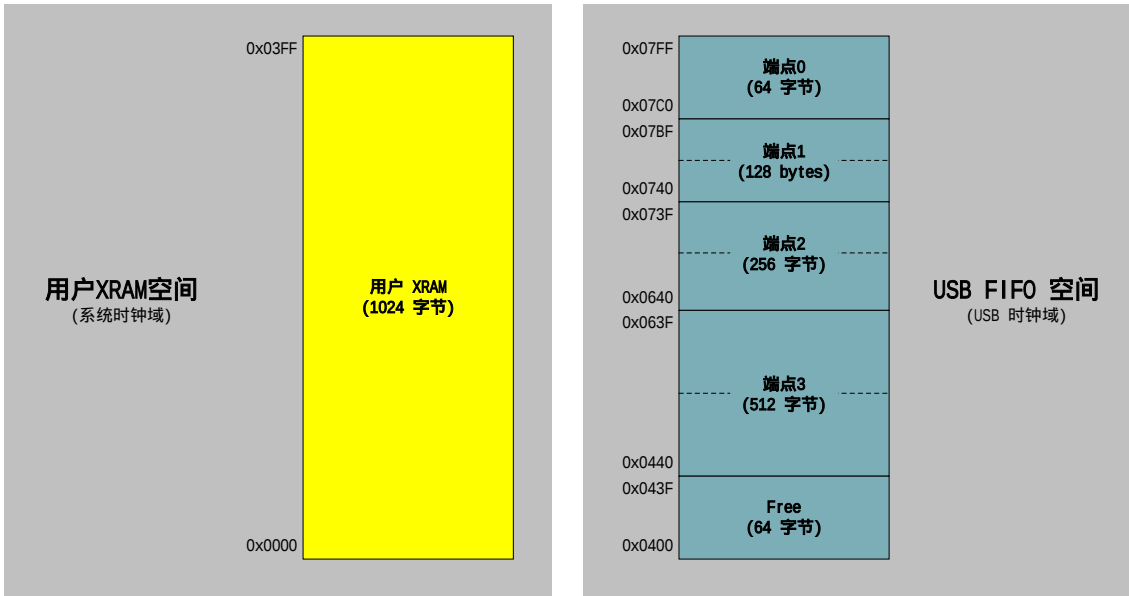


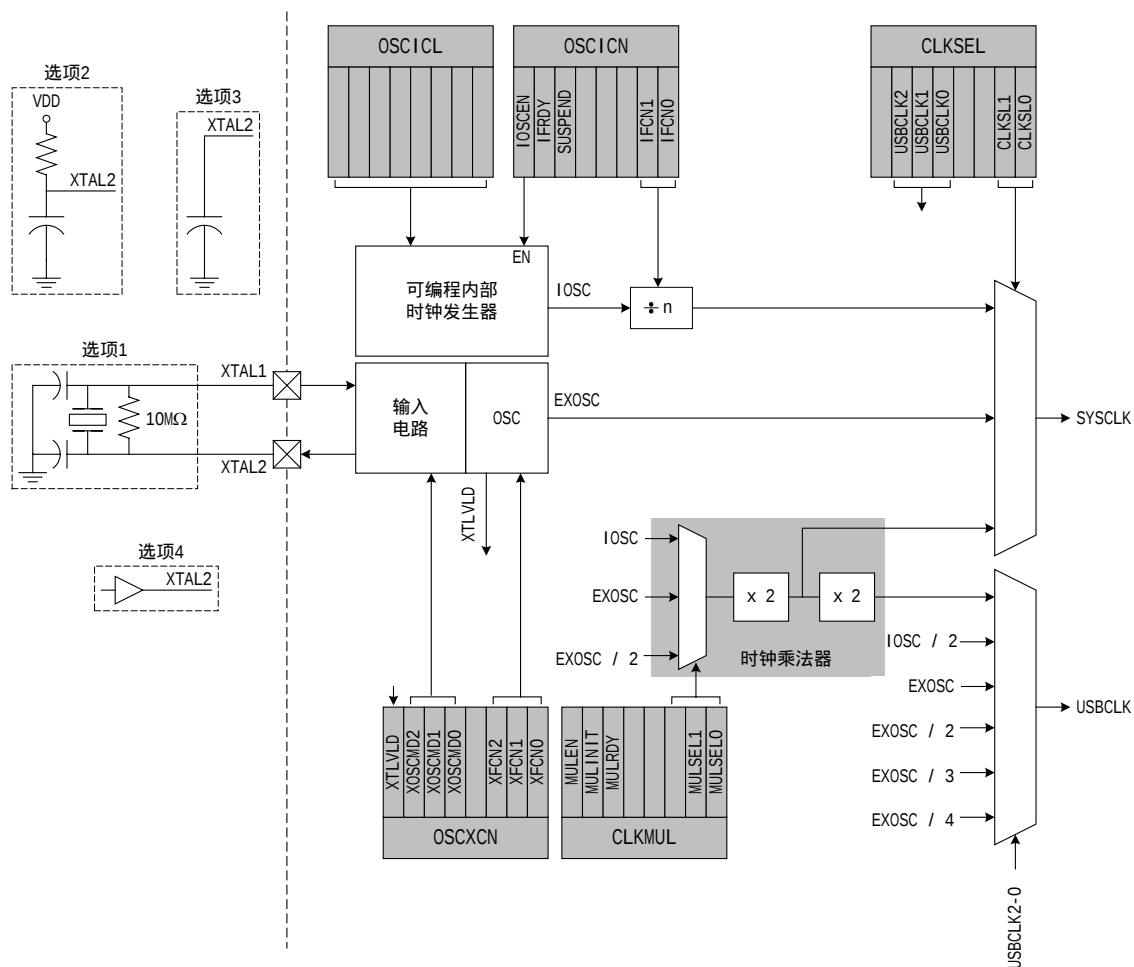
图 12.3 EMI0CN: 外部存储器接口控制

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	-	PGSEL2	PGSEL1	PGSEL0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xAA
位 7-3： 未用。读为 00000b。 位 2-0： PGSEL[2:0]：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。由于该寄存器的高 5 位被忽略，所以 2K 地址块以取模的方式映射到整个 64KB 外部数据存储器地址空间。								

13. 振荡器

C8051F320/1 有一个可编程内部振荡器、一个外部振荡器驱动电路和一个 4 倍时钟乘法器。可以通过对 OSCICN 和 OSCICL 寄存器编程来使能/禁止内部振荡器和调节其输出频率（如图 13.1 所示）。系统时钟（SYSCLK）可以来自内部振荡器、外部振荡器电路或 4 倍时钟乘法器二分频。USB 时钟（USBCLK）可以来自内部振荡器、外部振荡器电路或 4 倍时钟乘法器。表 13.3 给出了内部振荡器的电气特性。

图 13.1 振荡器框图



13.1 可编程内部振荡器

C8051F320/1 器件包含一个可编程内部振荡器，该振荡器在系统复位后被默认为系统时钟。内部振荡器的周期可以通过OSCICL寄存器编程，根据方程 13.1 计算振荡周期，其中 f_{BASE} 为复位后内部振荡器的频率， ΔT 为内部振荡器周期的变化值， $\Delta OSCICL$ 为寄存器OSCICL中的数值的变化量。

方程 13.1 内部振荡器周期随 OSCICL 的典型变化量

$$\Delta T \cong 0.0025 \times \frac{1}{f_{BASE}} \times \Delta OSCICL$$

对于C8051F320/1 器件，OSCICL已经过工厂校准，基频 (f_{BASE}) 为 12MHz。表 13.3 给出了精密内部振荡器的电气特性。注意：系统时钟可以从内部振荡器分频得到，分频系数由寄存器OSCICN中的IFCN位设定，可为 1、2、4 或 8。复位后的缺省分频系数为 8。

13.1.1 C8051F320/1 的内部振荡器编程

OSCICL 复位值已经过工厂校准，对应的基频为 12MHz，精度为 $\pm 1.5\%$ ，该频率适合作为 USB 时钟。软件可以下面所述修改内部振荡器的频率。

注意：一旦内部振荡器的频率被修改，则它不能再被用作 USB 时钟。在发生任何器件复位后，内部振荡器的频率恢复为初始的出厂校准频率，只有在这个频率内部振荡器才适合作为 USB 时钟。

软件应读 OSCICL 的值并根据方程 13.1 对其进行调整以获得所期望的频率。下面的例子说明如何获得 11.6MHz 的内部振荡器频率。

f_{BASE} 为复位后内部振荡器的频率； T_{BASE} 为复位后内部振荡器的周期。

f_{DES} 为所期望的内部振荡器频率； T_{DES} 为所期望的内部振荡器周期。

$$f_{BASE} = 12000000 \text{ Hz}$$

$$f_{DES} = 11600000 \text{ Hz}$$

$$T_{BASE} = \frac{1}{12000000} \text{ s}$$

$$T_{DES} = \frac{1}{11600000} \text{ s}$$

所要求的周期变化量 (ΔT_{DES}) 为基频周期与期望周期之差。

$$\Delta T_{DES} = \frac{1}{11600000} - \frac{1}{12000000} = 2.87 \times 10^{-9} \text{ s}$$

将上面的计算值代入方程 13.1，求 $\Delta OSCICL$ ：

$$2.87 \times 10^{-9} = 0.0025 \times \frac{1}{f_{BASE}} \times \Delta OSCICL$$

$$\Delta OSCICL = 13.79$$

将 $\Delta OSCICL$ 截断为最近似的整数 (14) 后加到寄存器 OSCICL 的复位值中。

C8051F320/1 全速 USB ISP FLASH 微控制器

注意：如果 OSCICL 的复位值与 Δ OSCICL 之和大于 31 或小于 0，则器件不能产生所希望的频率。

13.1.2 内部振荡器挂起方式

通过向寄存器 OSCICN 中的 SUSPEND 位写 ‘1’ 可以将内部振荡器置于挂起方式。在挂起方式，内部振荡器停止运行，直到检测到一个非空闲 USB 事件或 VBUS 输入信号与 VBPOL 位（位于寄存器 REG0CN）所选择的极性匹配。注意：要检测 USB 事件，USB 收发器必须被使能。

图 13.2 OSCICN：内部振荡器控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
IOSCEN	IFRDY	SUSPEND	-	-	-	IFCN1	IFCN0	00010100
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xB2
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：内部振荡器按编程频率运行。								
位 5： SUSPEND：强制挂起控制位 向该位写 ‘1’ 将强制内部振荡器停止运行。在发生 USB 非空闲事件（即 RESUME 命令）或 VBUS 中断事件（见图 8.5）后内部振荡器重新启动。								
位 4-2： 未用。读 = 000b，写 = 忽略。								
位 1-0： IFCN1-0：内部振荡器频率控制位 00：SYSCLK 为内部振荡器 8 分频。 01：SYSCLK 为内部振荡器 4 分频。 10：SYSCLK 为内部振荡器 2 分频。 11：SYSCLK 为内部振荡器输出（不分频）。								

图 13.3 OSCICL：内部振荡器校准寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	OSCCAL				-	可变
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xB3
位 7-5： 未用。读 = 000b，写 = 忽略。								
位 4-0： OSCCAL：内部振荡器校准值。 这些位决定内部振荡器的周期（根据方程 13.1）。								
注：当时钟恢复电路被使能时，该寄存器的内容不确定。有关时钟恢复电路的详细信息见“15.4 USB 时钟配置”。								

13.2 外部振荡器驱动电路

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。对于晶体和陶瓷谐振器配置，晶体/陶瓷谐振器必须并接到 XTAL1 和 XTAL2 引脚（见图 13.1，选项 1），还必须在 XTAL1 和 XTAL2 引脚之间并接一个 10M Ω 的电阻。对于 RC、电容或 CMOS 时钟配置，时钟源应接到 XTAL2 引脚（见图 13.1，选项 2、3、4）。必须在 OSCXCN 寄存器中选择外部振荡器类型，还必须正确选择频率控制位 XFCN（见图 13.4）。

注意：当使用外部振荡器电路时，必须对所用端口引脚进行配置。当外部振荡器电路被配置为晶体/谐振器方式时，端口引脚 P0.2 和 P0.3 分别被用作 XTAL1 和 XTAL2。当外部振荡器电路被配置为 RC、电容或 CMOS 时钟方式时，端口引脚 P0.3 被用作 XTAL2。端口 I/O 交叉开关应被配置为跳过被振荡器占用的引脚。当在晶体/陶瓷谐振器、电容或 RC 方式使用外部振荡器电路时，应将所用的端口引脚配置为**模拟输入**。在 CMOS 时钟方式，应将所用的端口引脚配置为**数字输入**。有关端口输入方式选择的详细信息见“14.2 端口 I/O 初始化”。

13.2.1 外部振荡器作为定时器的时钟

外部振荡器 8 分频是定时器和可编程计数器阵列（PCA）的一个时钟选项（见“19. 定时器”和“20. 可编程计数器阵列（PCA0）”）。当外部振荡器被用作这些外设的时钟而不用作系统时钟时，外部振荡器频率必须小于或等于系统时钟频率。在这种配置下，提供给外设的时钟（外部振荡器/8）与系统时钟同步，这种同步的抖动被限制为 ± 0.5 个系统时钟周期。

13.2.2 外部晶体示例

如果使用晶体或陶瓷谐振器作为 MCU 的外部振荡源，则电路为图 13.1 中的选项 1。应从图 13.4（OSCXCN 寄存器）中的晶体列选择外部振荡器频率控制值（XFCN）。例如，一个 12MHz 的晶体要求 XFCN 设置值为 111b。

在晶体振荡器被使能时，振荡器幅度检测电路需要一个建立时间来达到合适的偏置。在使能晶体振荡器和检查 XTLVLD 位之间引入 1ms 的延时可以防止提前将系统时钟切换到外部振荡器。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。建议的步骤如下：

1. 使能外部振荡器。
2. 等待至少 1ms。
3. 查询 XTLVLD $\Rightarrow$ ‘1’。
4. 将系统时钟切换到外部振荡器。

注意，晶体振荡器电路对 PCB 布局非常敏感。应将晶体尽可能地靠近器件的 XTAL 引脚，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。

13.2.3 外部 RC 示例

如果使用外部 RC 网络作为 MCU 的外部振荡源，则配置电路为图 13.1 中的选项 2。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，使频率偏差很大。为了确定 OSCXCN 寄存器所需要的外部振荡器频率控制值 (XFCN)，首先选择能产生所要求的振荡频率的 RC 网络值。如果所希望的频率是 100kHz，选 $R = 246\text{k}\Omega$ 和 $C = 50\text{pF}$ ：

$$f = 1.23 \times 10^3 / RC = 1.23 \times 10^3 / [246 \times 50] = 0.1\text{MHz} = 100\text{kHz}$$

查图 13.4 中的表，得到所需要的 XFCN 值为 010b。在 RC 方式，将 XFCN 编程为较高的设置值会改善频率精度，但外部振荡器的电源电流增加。

13.2.4 外部电容示例

如使用外部电容作为 MCU 的外部振荡源，则电路为图 13.1 中的选项 3。电容应不大于 100pF；但当电容值很小时，PCB 的寄生将在总电容中占支配地位，使频率偏差很大。为了确定 OSCXCN 寄存器所需要的外部振荡器频率控制值 (XFCN)，选择要用的电容并利用下面的公式计算振荡频率。假设 $VDD = 3.0\text{V}$ 和 $C = 50\text{pF}$ ：

$$f = KF / (C * VDD) = KF / (50 \times 3) \text{ MHz}$$

$$f = KF / 150 \text{ MHz}$$

如果所需要的频率大约为 150kHz，从图 13.4 的表中选择 K 因子，得到 $KF = 22$ ：

$$f = 22 / 150 = 0.146 \text{ MHz}，\text{或 } 146 \text{ kHz}。$$

因此，本例中要用的 XFCN 值为 011b。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 13.4 OSCXCN：外部振荡器控制寄存器

R	R/W	R/W	R/W	R	R/W	R/W	R/W	复位值
XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	-	XFCN2	XFCN1	XFCN0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xB1

位 7：XTLVLD：晶体振荡器有效标志
 （在 XOSCMD = 11x 时有效，只读）
 0：晶体振荡器未用或未稳定。
 1：晶体振荡器正在运行并且工作稳定。

位 6-4：XOSCMD2-0：外部振荡器方式位
 00x：外部振荡器电路关闭。
 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 = 0，写 = 忽略。

位 2-0：XFCN2-0：外部振荡器频率控制位。
 000-111：见下表

XFCN	晶体 (XOSCMD=11x)	RC(XOSCMD=10x)	C(XOSCMD=10x)
000	$f \leq 32\text{kHz}$	$f \leq 25\text{kHz}$	K 因子= 0.87
001	$32\text{kHz} < f \leq 84\text{kHz}$	$25\text{kHz} < f \leq 50\text{kHz}$	K 因子= 2.6
010	$84\text{kHz} < f \leq 225\text{kHz}$	$50\text{kHz} < f \leq 100\text{kHz}$	K 因子= 7.7
011	$225\text{kHz} < f \leq 590\text{kHz}$	$100\text{kHz} < f \leq 200\text{kHz}$	K 因子= 22
100	$590\text{kHz} < f \leq 1.5\text{MHz}$	$200\text{kHz} < f \leq 400\text{kHz}$	K 因子= 65
101	$1.5\text{MHz} < f \leq 4\text{MHz}$	$400\text{kHz} < f \leq 800\text{kHz}$	K 因子= 180
110	$4\text{MHz} < f \leq 10\text{MHz}$	$800\text{kHz} < f \leq 1.6\text{MHz}$	K 因子= 664
111	$10\text{MHz} < f \leq 30\text{MHz}$	$1.6\text{MHz} < f \leq 3.2\text{MHz}$	K 因子= 1590

晶体方式（电路见图 13.1，选项 1；XOSCMD=11x）
 选择与晶体振荡器频率匹配的 XFCN 值。

RC 方式（电路见图 13.1，选项 2；XOSCMD=10x）
 选择与频率范围匹配的 XFCN 值：
 $f = 1.23(10^3)/(R \cdot C)$ ，其中：
 f = 振荡器频率（MHz）
 C = 电容值（pF）
 R = 上拉电阻值（kΩ）

C 方式（电路见图 13.1，选项 3；XOSCMD=10x）
 根据所期望的振荡器频率选择 K 因子（KF）：
 $f = KF/(C \cdot VDD)$ ，其中：
 f = 振荡器频率（MHz）
 C = XTAL2 引脚的电容值（pF）
 VDD = MCU 的电源电压值（伏）

13.3 4 倍时钟乘法器

4 倍时钟乘法器允许使用 12MHz 振荡器产生全速 USB 通信所需要的 48MHz 时钟(见“ 15.4 USB 时钟配置”)。时钟乘法器输出的二分频也可以被用作系统时钟。关于系统时钟和 USB 时钟选择见 13.4 节。

用 CLKMUL 寄存器配置 4 倍时钟乘法器。配置和使能 4 倍时钟乘法器的步骤如下：

1. 通过向寄存器 CLKMUL 写 0x00 来复位时钟乘法器。
2. 用 MULSEL 位选择时钟乘法器的输入。
3. 用 MULEN 位使能时钟乘法器 (CLKMUL |= 0x80)。
4. 延时大于 5 μ s。
5. 用 MULINIT 位初始化时钟乘法器 (CLKMUL |= 0xC0)。
6. 查询等待 MULRDY => ‘1’。

注意：当使用外部振荡器作为 4 倍时钟乘法器的输入时，外部振荡源必须在乘法器被初始化之前被使能并稳定运行。

图 13.5 CLKMUL：时钟乘法器控制寄存器

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	复位值
MULEN	MULINIT	MULRDY	-	-	-	MULSEL		00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xB9

位 7： MULEN：时钟乘法器使能位
0：时钟乘法器禁止。
1：时钟乘法器使能。

位 6： MULINIT：时钟乘法器初始化控制位
0：当时钟乘法器被使能时，该位应为‘ 0 ’。时钟乘法器被使能后向该位写‘ 1 ’
将初始化时钟乘法器。当时钟乘法器稳定后，MULRDY 的读出值为‘ 1 ’。

位 5： MULRDY：时钟乘法器准备好标志
该只读位指示时钟乘法器的状态。
0：时钟乘法器未准备好。
1：时钟乘法器已准备好（锁定）。

位 4-2： 未用。读 = 000b，写 = 忽略。

位 1-0： MULSEL：时钟乘法器输入选择位
这两位选择去时钟乘法器的时钟。

MULSEL	选择的时钟
00	内部振荡器
01	外部振荡器
10	外部振荡器/2
11	保留

13.4 系统时钟和 USB 时钟选择

内部振荡器的起动时间很短，因此可以在使能内部振荡器的 OSCICN 写操作后立即选择内部振荡器为系统时钟或 USB 时钟。外部晶体和陶瓷谐振器通常需要较长的启动时间才能达到稳定。当外部振荡器稳定后，晶体有效标志(寄存器 OSCXCN 中的 XTLVLD)被硬件置‘ 1 ’。在晶体方式，为了防止读到假 XTLVLD 标志，软件在使能外部振荡器和检查 XTLVLD 之间至少应延时 1ms。RC 和 C 方式通常不需要启动时间。

13.4.1 系统时钟选择

寄存器 CLKSEL 中的 CLKSL[1:0]位选择用作系统时钟的振荡源。当选择外部振荡器作为系统时钟时，CLKSL[1:0]必须被设置为 01b。当选择内部振荡器作为系统时钟时，外部振荡器仍然可以给外设（定时器、PCA、USB）提供时钟。系统时钟可以在内部振荡器、外部振荡器和 4 倍时钟乘法器之间自由切换，只要所选择的振荡器被使能并稳定运行。

13.4.2 USB 时钟选择

寄存器 CLKSEL 中的 USBCLK[2:0]位选择用作 USB 时钟的振荡源。USB 时钟可以来自 4 倍时钟乘法器输出、内部振荡器分频或外部振荡器分频。注意：当 USB0 工作在全速方式时，USB 时钟必须是 48 MHz；当 USB0 工作在低速方式时，USB 时钟必须是 6 MHz。USB 的时钟选择见图 13.6。

下面给出了全速和低速方式下 USB 时钟配置的几个例子：

表 13.1 典型 USB 全速方式时钟设置

内部振荡器		
时钟信号	输入源选择	寄存器位设置
USB 时钟	时钟乘法器	USBCLK = 000b
时钟乘法器输入	内部振荡器*	MULSEL = 00b
内部振荡器	不分频	IFCN = 11b
外部振荡器		
时钟信号	输入源选择	寄存器位设置
USB 时钟	时钟乘法器	USBCLK = 000b
时钟乘法器输入	外部振荡器*	MULSEL = 01b
外部振荡器	晶体振荡器方式 12 MHz 晶体	XOSCMD = 110b IFCN = 111b
*对于该配置，时钟恢复电路必须被使能。		

表 13.2 典型 USB 低速方式时钟设置

内部振荡器		
时钟信号	输入源选择	寄存器位设置
USB 时钟	内部振荡器/2	USBCLK = 001b
时钟乘法器输入	不分频	IFCN = 11b
外部振荡器		
时钟信号	输入源选择	寄存器位设置
USB 时钟	外部振荡器/4	USBCLK = 101b
外部振荡器	晶体振荡器方式 24 MHz 晶体	XOSCMD = 110b IFCN = 111b

图 13.6 CLKSEL：时钟选择寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	USBCLK			-	-	CLKSL		00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xA9

位 7： 未用。读 = 0b，写 = 忽略。

位 6-4： USBCLK2-0：USB 时钟选择位

这些位选择去 USB0 的时钟。当 USB0 工作于全速方式时，所选时钟应为 48 MHz。当 USB0 工作于低速方式时，所选时钟应为 6 MHz。

USBCLK	选择的时钟
000	4 倍时钟乘法器
001	内部振荡器/2
010	外部振荡器
011	外部振荡器/2
100	外部振荡器/3
101	外部振荡器/4
110	保留
111	保留

位 3-2： 未用。读 = 00b，写 = 忽略。

位 1-0： CLKSL1-0：系统时钟选择位

这些位选择系统时钟源。

CLKSL	选择的时钟
00	内部振荡器（由寄存器 OSCICN 中的 IFCN 位选择分频系数）
01	外部振荡器
10	4 倍时钟乘法器/2
11	保留

C8051F320/1 全速 USB ISP FLASH 微控制器

表 13.3 内部振荡器电气特性

-40 到+85 （除非特别说明）

参 数	条 件	最小值	典型值	最大值	单位
内部振荡器频率	复位频率	11.82	12	12.18	MHz
内部振荡器电源电流(自 VDD)	OSCICN.7 = 1		450		μA
USB 时钟频率*	全速	47.88	48	48.12	MHz
	低速	5.91	6	6.09	
*只适用于外部振荡源。					

14. 端口输入/输出

数字和模拟资源可以通过 25 个 I/O 引脚 (C8051F320) 或 21 个 I/O 引脚 (C8051F321) 使用。端口引脚被组织如图 14.1 所示。每个端口引脚都可以被定义为通用 I/O (GPIO) 或模拟输入。P0.0 ~ P2.3 可以被分配给内部数字资源, 如图 14.3 所示。设计者完全控制数字功能的引脚分配, 只受 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。注意, 不论交叉开关的设置如何, 端口 I/O 引脚的状态总是可以被读到相应的端口锁存器。

交叉开关根据优先权译码表 (图 14.3 和图 14.4) 为所选择的内部数字资源分配 I/O 引脚。寄存器 XBR0 和 XBR1 (见图 14.5 和图 14.6) 用于选择内部数字功能。

所有端口 I/O 都耐 5V 电压 (参见图 14.2 的端口 I/O 单元电路)。端口 I/O 单元可以被配置为漏极开路或推挽方式 (在端口输出方式寄存器 PnMDOUT 中设置, $n = 0, 1, 2, 3$)。表 14.1 给出了端口 I/O 的电气特性。

图 14.1 端口 I/O 功能框图

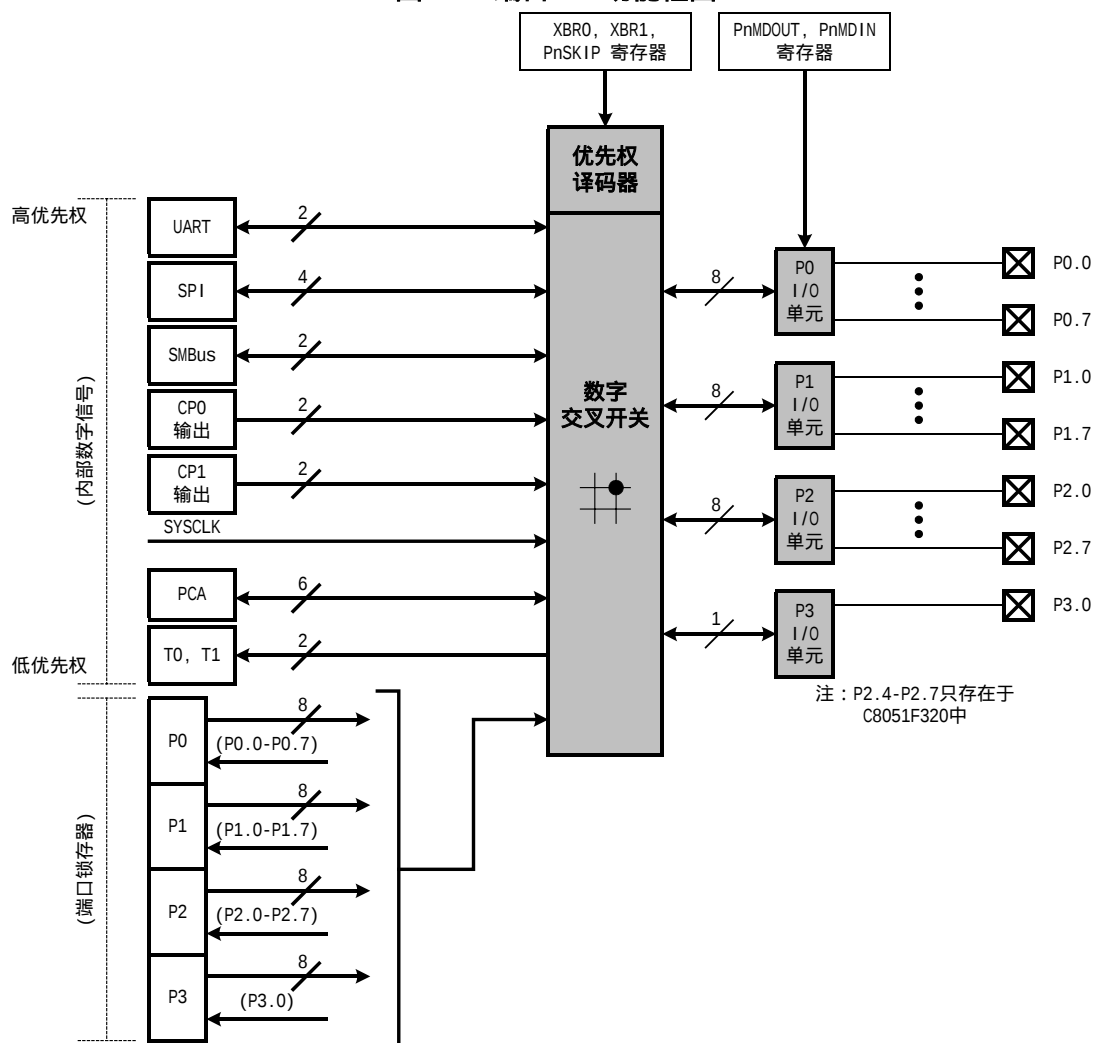
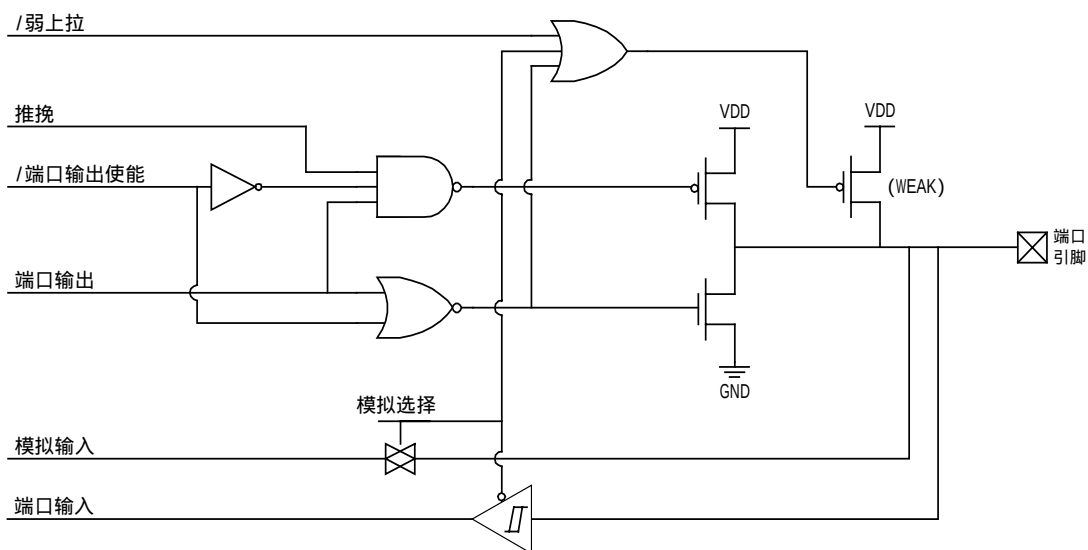


图 14.2 端口 I/O 单元框图



14.1 优先权交叉开关译码器

优先权交叉开关译码器（图 14.3）为每个 I/O 功能分配优先权，从优先权最高的 UART0 开始。当一个数字资源被选择时，尚未分配的端口引脚中的最低位被分配给该资源（UART0 总是使用引脚 4 和 5）。如果一个端口引脚已经被分配，则交叉开关在为下一个被选择的资源分配引脚时将跳过该引脚。此外，交叉开关还将跳过在 PnSKIP 寄存器中被置‘1’的那些位所对应的引脚。PnSKIP 寄存器允许软件跳过那些被用作模拟输入、特定功能或 GPIO 的引脚。

注意：如果一个端口引脚被一个外设使用而不经交叉开关，则该引脚在 PnSKIP 寄存器中的对应位应被置‘1’。这种情况适用于 P0.7（如果使用 VREF）、P0.3 和/或 P0.2（如果外部振荡器电路被使能）、P0.6（如果 ADC 被配置为使用外部转换启动信号 CNVSTR）和任何被选择为 ADC 或比较器输入的引脚。交叉开关跳过那些已经被分配的引脚，移向下一个未被分配的引脚。图 14.3 示出没有引脚被跳过（P0SKIP, P1SKIP, P2SKIP = 0x00）的优先权交叉开关译码表；图 14.4 给出了 XTAL1(P0.2)脚和 XTAL2(P0.3)脚被跳过情况下（P0SKIP = 0x0C）的交叉开关优先权译码表。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 14.3 没有引脚被跳过的交叉开关优先级译码表

	P0								P1								P2							
特殊信号	XTAL1 XTAL2				CNVSTR VREF																			
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																								
MISO																								
MOSI																								
NSS*									*NSS只在4线SPI方式有引出脚															
SDA																								
SCL																								
CP0																								
CP0A																								
CP1																								
CP1A																								
SYSCCLK																								
CEX0																								
CEX1																								
CEX2																								
CEX3																								
CEX4																								
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	0
	P0SKIP[0:7]								P1SKIP[0:7]								P2SKIP[0:3]							

 可被外设使用的端口引脚
 特殊信号 未被交叉开关分配引脚的特殊功能信号。
 当这些信号被使能时，交叉开关必须被配置为跳过它们对应的端口引脚。

信号不可用

C8051F320/1 全速 USB ISP FLASH 微控制器

图 14.4 晶体引脚被跳过的交叉开关优先权译码表

	P0								P1								P2							
特殊信号	XTAL1 XTAL2				CNVSTR VREF																			
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																								
MISO																								
MOSI																								
NSS*									* NSS只在4线SPI方式有引出脚															
SDA																								
SCL																								
CP0																								
CP0A																								
CP1																								
CP1A																								
SYSCLK																								
CEX0																								
CEX1																								
CEX2																								
CEX3																								
CEX4																								
ECI																								
T0																								
T1																								
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	P0SKIP[0:7]								P1SKIP[0:7]								P2SKIP[0:3]							

可被外设使用的端口引脚

特殊信号

未被交叉开关分配引脚的特殊功能信号。
当这些信号被使能时，交叉开关必须被配置为跳过它们对应的端口引脚。

信号不可用

寄存器 XBR0 和 XBR1 用于为数字 I/O 资源分配物理 I/O 引脚。注意，当 SMBus 被选择时，交叉开关将为其分配两个引脚（SDA 和 SCL）。当 UART 被选择时，交叉开关也为其分配两个引脚（TX 和 RX）。UART0 的引脚分配是固定的（这是出于引导装入的目的）：UART TX0 总是被分配到 P0.4；UART RX0 总是被分配到 P0.5。在优先功能被分配之后，标准端口 I/O 是连续的。

注意：SPI 可以工作在三线或四线方式，取决于 SPI0CN 寄存器中 NSSMD1-NSSMD0 位的状态。根据 SPI 方式不同，NSS 信号可以连到端口引脚，也可以不连到端口引脚。

14.2 端口 I/O 初始化

端口 I/O 初始化包括以下步骤：

1. 用端口输入方式寄存器 (PnMDIN) 选择所有端口引脚的输入方式 (模拟或数字)。
2. 用端口输出方式寄存器 (PnMDOUT) 选择所有端口引脚的输出方式 (漏极开路或推挽)。
3. 用端口跳过寄存器 (PnSKIP) 选择应被交叉开关跳过的那些引脚。
4. 将引脚分配给要使用的外设 (XBR0、XBR1)。
5. 使能交叉开关 (XBARE = '1')。

所有端口引脚都必须被配置为模拟或数字输入。被用作比较器或 ADC 输入的任何引脚都应被配置为模拟输入。当一个引脚被配置为模拟输入时，它的弱上拉、数字驱动器和数字接收器都被禁止，这可以节省功耗并减小模拟输入的噪声。被配置为数字输入的引脚可以被模拟外设使用，但不建议这样做。要将一个端口引脚配置为数字输入，向 PnMDOUT 寄存器中的对应位写 '0'，并向对应的端口锁存器 (Pn) 写 '1'。

此外，应将交叉开关配置为跳过所有被用作模拟输入的引脚 (通过将 PnSKIP 寄存器中的对应位置 '1' 来实现)。端口输入方式在 PnMDIN 寄存器中设置，其中 '1' 表示数字输入，'0' 表示模拟输入。复位后所有引脚的缺省设置都是数字输入。对 PnMDIN 寄存器的详细说明见图 14.8。

I/O 引脚的输出驱动器特性由端口输出方式寄存器 PnMDOUT 中的对应位决定，每个端口输出驱动器都可被配置为漏极开路或推挽方式。不管交叉开关是否将端口引脚分配给某个数字外设，都需要对端口驱动器的输出方式进行设置。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。

当 XBR1 寄存器中的 WEAKPUD 位为 '0' 时，输出方式为漏极开路的那些引脚的弱上拉被使能。WEAKPUD 不影响被配置为推挽方式的端口 I/O。当漏极开路输出被驱动为逻辑 '0' 时，弱上拉被自动关断 (禁止) 以避免不必要的功率消耗。

寄存器 XBR0 和 XBR1 必须被装入正确的值以选择所需要的数字 I/O 功能。置 '1' XBR1 中的 XBARE 位将使能交叉开关。不管 XBRn 寄存器的设置如何，在交叉开关被使能之前，外部引脚保持标准端口 I/O 方式 (输入)。对于给定的 XBRn 设置，可以使用优先权译码表确定 I/O 引脚分配；另一种方法是使用 Silicon Labs IDE 软件的配置向导功能来确定基于 XBRn 寄存器设置的端口 I/O 引脚分配。

注意：为使端口 P0、P1 和 P2.0 ~ P2.3 工作在标准端口 I/O 输入/输出方式，交叉开关必须被使能。当交叉开关被禁止时，端口输出驱动器被禁止。P2.4 ~ P2.7 和 P3.0 总是作为标准 GPIO 使用。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 14.5 XBR0：端口 I/O 交叉开关寄存器 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	SFR地址： 0xE1
位 5： CP1AE：比较器 1 异步输出使能位 0：CP1A 不连到端口引脚。 1：CP1A 连到端口引脚。 位 4： CP1E：比较器 1 输出使能位 0：CP1 不连到端口引脚。 1：CP1 连到端口引脚。 位 5： CP0AE：比较器 0 异步输出使能位 0：CP0A 不连到端口引脚。 1：CP0A 连到端口引脚。 位 4： CP0E：比较器 0 输出使能位 0：CP0 不连到端口引脚。 1：CP0 连到端口引脚。 位 3： SYSCKE：/SYSCLK 输出使能位 0：/SYSCLK 不连到端口引脚； 1：/SYSCLK 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：SMBus I/O 连到端口引脚。 位 1： SPI0E：SPI I/O 使能位 0：SPI I/O 不连到端口引脚。 1：SPI I/O 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：UART TX0, RX0 连到端口引脚 P0.4 和 P0.5。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 14.6 XBR1：端口 I/O 交叉开关寄存器 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
WEAKPUD	XBARE	T1E	T0E	ECIE	PCA0ME			00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xE2
位 7：WEAKPUD：端口 I/O 弱上拉禁止位。 0：弱上拉使能（被配置为模拟输入或推挽输出的端口除外）。 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：T1E：T1 使能位 0：T1 不连到端口引脚。 1：T1 连到端口引脚。 位 4：T0E：T0 使能位 0：T0 不连到端口引脚。 1：T0 连到端口引脚。 位 3：ECIE：PCA0 外部计数输入使能位 0：ECI 不连到端口引脚。 1：ECI 连到端口引脚。 位 2-0：PCA0ME：PCA 模块 I/O 使能位 000：所有的 PCA 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、CEX3 连到端口引脚。 101：CEX0、CEX1、CEX2、CEX3、CEX4 连到端口引脚。 110：保留。 111：保留。								

14.3 通用端口 I/O

未被交叉开关分配的端口引脚和未被模拟外设使用的端口引脚都可以作为通用 I/O。通过对应的端口数据寄存器访问端口 P3-0，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，数据被锁存到端口数据寄存器中，以保持引脚上的输出数据值不变。读端口数据寄存器（或端口位）将总是返回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其它信号时，端口寄存器总是读其对应的端口 I/O 引脚。但在对端口 SFR 执行下面的读-修改-写指令（ANL、ORL、XRL、JBC、CPL、INC、DEC、DJNZ）和对端口 SFR 中的某一位执行 MOV、CLR、SETB 期间例外。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。

图 14.7 P0：端口 0 寄存器

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

位 7-0： P0.[7:0]
 写 - 输出出现在 I/O 引脚（根据交叉开关寄存器的设置，XBARE = '1'），
 0：逻辑低电平输出。
 1：逻辑高电平输出。（若相应的 P0MDOUT.n 位 = 0，则为高阻态）
 读 - 读那些在 P0MDIN 中被选择为模拟输入的引脚时总是返回 '0'。被配置为数字输入时直接读端口引脚。
 0：P0.n 为逻辑低电平。
 1：P0.n 为逻辑高电平。

图 14.8 P0MDIN：端口 0 输入方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xF1

位 7-0： P0.7 – P0.0 模拟输入配置位（分别对应）
 当端口引脚被配置为模拟输入时，其弱上拉、数字驱动器和数字接收器都被禁止。
 0：对应的 P0.n 引脚被配置为模拟输入。
 1：对应的 P0.n 引脚不配置为模拟输入。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 14.9 P0MDOUT：端口 0 输出方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xA4
位 7-0： P0.7 – P0.0 输出方式配置位（分别对应）。如果 P0MDIN 寄存器中的对应位为逻辑‘0’，则输出方式配置位被忽略。 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。 注：当 SDA 和 SCL 出现在端口引脚时，总是被配置为漏极开路，与 P0MDOUT 的设置值无关。								

图 14.10 P0SKIP：端口 0 跳过寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xD4
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引脚。作为模拟输入（ADC 或比较器）或特殊功能（VREF 输入、外部振荡器电路、CNVSTR 输入）的引脚应被交叉开关跳过。 0：对应的 P0.n 引脚不被交叉开关跳过。 1：对应的 P0.n 引脚被交叉开关跳过。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 14.11 P1：端口 1 寄存器

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

位 7-0： P1.[7:0]
写 - 输出出现在 I/O 引脚（根据交叉开关寄存器的设置，XBARE = '1'），
0：逻辑低电平输出。
1：逻辑高电平输出。（若相应的 P1MDOUT.n 位 = 0，则为高阻态）
读 - 读那些在 P1MDIN 中被选择为模拟输入的引脚时总是返回 '0'。被配置为数字输入时直接读端口引脚。
0：P1.n 为逻辑低电平。
1：P1.n 为逻辑高电平。

图 14.12 P1MDIN：端口 1 输入方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xF2

位 7-0： P1MDIN.[7:0]；P1.7 – P1.0 模拟输入配置位（分别对应）
当端口引脚被配置为模拟输入时，其弱上拉、数字驱动器和数字接收器都被禁止。
0：对应的 P1.n 引脚被配置为模拟输入。
1：对应的 P1.n 引脚不配置为模拟输入。

图 14.13 P1MDOUT：端口 1 输出方式寄存器

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

位 7-0： P1MDOUT.[7:0]；P1.7 – P1.0 输出方式配置位（分别对应）。如果 P1MDIN 寄存器中的对应位为逻辑 '0'，则输出方式配置位被忽略。
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。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 14.14 P1SKIP：端口 1 跳过寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xD5

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引脚。作为模拟输入（ADC 或比较器）或特殊功能（VREF 输入、外部振荡器电路、CNVSTR 输入）的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。

图 14.15 P2：端口 2 寄存器

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

位 7-0： P2.[7:0]
写 - 输出出现在 I/O 引脚（根据交叉开关寄存器的设置）。
0：逻辑低电平输出。
1：逻辑高电平输出。（若相应的 P2MDOUT.n 位 = 0，则为高阻态）。
读 - 读那些在 P2MDIN 中被选择为模拟输入的引脚时总是返回‘0’。被配置为数字输入时直接读端口引脚。
0：P2.n 为逻辑低电平。
1：P2.n 为逻辑高电平。
注：P2.7-P2.4 只存在于 C8051F320 中，写这些端口时不需要 XBARE = ‘1’。

图 14.16 P2MDIN：端口 2 输入方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xF3

位 7-0： P2.7 – P2.0 模拟输入配置位（分别对应）
当端口引脚被配置为模拟输入时，其弱上拉、数字驱动器和数字接收器都被禁止。
0：对应的 P2.n 引脚被配置为模拟输入。
1：对应的 P2.n 引脚不配置为模拟输入。
注：P2.7-P2.4 只存在于 C8051F320 中。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 14.17 P2MDOUT：端口 2 输出方式寄存器

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

位 7-0：P2.7 – P2.0 输出方式配置位（分别对应）。如果 P2MDIN 寄存器中的对应位为逻辑‘0’，则输出方式配置位被忽略。
 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：P2.7-P2.4 只存在于 C8051F320 中。

图 14.18 P2SKIP：端口 2 跳过寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xD6

位 7-4：未使用。读 = 0000b。写 = 忽略。
 位 3-0：P2SKIP.[3:0]：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引脚。作为模拟输入（ADC 或比较器）或特殊功能（VREF 输入、外部振荡器电路、CNVSTR 输入）的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。

图 14.19 P3：端口 3 寄存器

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

位 7-0：P3.[7:0]
 写 - 输出出现在 I/O 引脚（根据交叉开关寄存器的设置）。
 0：逻辑低电平输出。
 1：逻辑高电平输出。（若相应的 P2MDOUT.n 位 = 0，则为高阻态）
 读 - 读那些在 P3MDIN 中被选择为模拟输入的引脚时总是返回‘0’。被配置为数字输入时直接读端口引脚。
 0：P3.n 为逻辑低电平。
 1：P3.n 为逻辑高电平。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 14.20 P3MDIN：端口 3 输入方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000001
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xF4

位 7-1：未使用。读 = 0000000b。写 = 忽略。
 位 0：P3.0 模拟输入配置位
 当端口引脚被配置为模拟输入时，其弱上拉、数字驱动器和数字接收器都被禁止。
 0：P3.0 引脚被配置为模拟输入。
 1：P3.0 引脚不配置为模拟输入。

图 14.21 P3MDOUT：端口 3 输出方式寄存器

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

位 7-1：未使用。读 = 0000000b。写 = 忽略。
 位 0：P3.0 输出方式配置位。如果 P3MDIN 寄存器中的对应位为逻辑‘0’，则输出方式配置位被忽略。
 0：P3.0 输出为漏极开路。
 1：P3.0 输出为推挽方式。

表 14.1 端口 I/O 直流电气特性

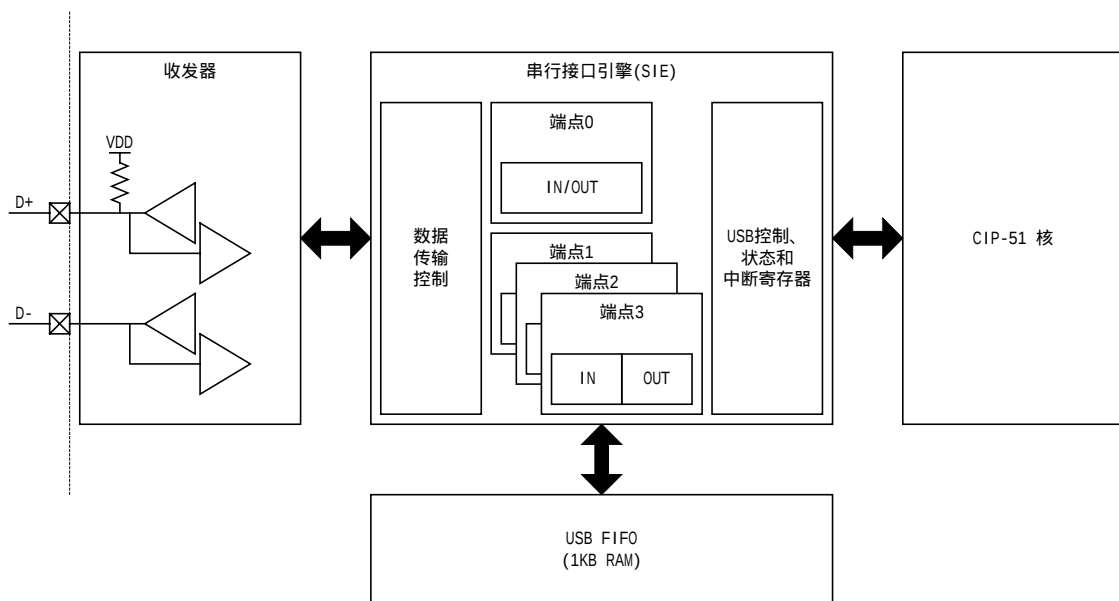
VDD = 2.7V – 3.6V, -40 到+85 （除非特别说明）

参 数	条 件	最小值	典型值	最大值	单 位
输出高电压 (V _{OH})	I _{OH} = -10μA，端口I/O为推挽方式 I _{OH} = -3mA，端口I/O为推挽方式 I _{OH} = -10mA，端口I/O为推挽方式	VDD-0.1 VDD-0.7	VDD-0.8		V
输出低电压 (V _{OL})	I _{OL} = 10μA I _{OL} = 8.5mA I _{OL} = 25mA		1.0	0.1 0.6	V
输入高电压 (V _{IH})		2.0			V
输入低电压 (V _{IL})				0.8	V
输入漏电流	弱上拉禁止 弱上拉使能，V _{IN} = 0 V		25	± 1 50	μA

15. 通用串行总线控制器 (USB0)

C8051F320/1 器件集成了全速/低速 USB 功能控制器，用于实现 USB 接口的外部设备（C8051F320/1 不能被用作 USB 主设备）。USB 功能控制器 (USB0) 由串行接口引擎 (SIE)、USB 收发器（包括匹配电阻和可配置上拉电阻）、1KB FIFO 存储器和时钟恢复电路（可以不用晶体）组成，不需要外部元件。USB 功能控制器和收发器符合通用串行总线规范 2.0 版。

图 1.5 USB0 原理框图



注意：本文档假设读者了解 USB 协议。本文档所使用的术语和缩略语在 USB 规范中有定义。我们希望您能在进一步阅读之前重温 USB 规范的最新版本。

15.1 端点地址

共有 8 个端点管道。控制端点（端点 0）总是作为双向 IN/OUT 端点。其它端点被作为 3 对 IN/OUT 端点管道。

表 15.1 端点地址

端点	相应的管道	USB 协议地址
端点 0	端点 0 输入（IN）	0x00
	端点 0 输出（OUT）	0x00
端点 1	端点 1 输入（IN）	0x81
	端点 1 输出（OUT）	0x01
端点 2	端点 2 输入（IN）	0x82
	端点 2 输出（OUT）	0x02
端点 3	端点 3 输入（IN）	0x83
	端点 3 输出（OUT）	0x03

15.2 USB 收发器

用 USB0XC� 寄存器（见图 15.2）配置 USB 收发器。配置内容包括收发器使能/禁止、上拉电阻使能/禁止及器件速度选择（全速或低速）。当位 SPEED = ‘1’ 时，USB0 工作在全速 USB 方式，片内上拉电阻（如果被使能）出现在 D+ 引脚。当位 SPEED = ‘0’ 时，USB0 工作在低速 USB 方式，片内上拉电阻（如果被使能）出现在 D- 引脚。寄存器 USB0XC� 的位 4~0 用于收发器测试（见图 15.2 的说明）。上拉电阻只有在有 VBUS 信号被使能（有关 VBUS 检测的详细信息见“8.2 VBUS 检测”）。

注意：USB 时钟应在收发器被使能之前就处于活动状态。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.2 USB0XCN : USB0 收发器控制寄存器

R/W	R/W	R/W	R/W	R/W	R	R	R	复位值
PREN	PHYEN	SPEED	PHYTST1	PHYTST0	DFREC	Dp	Dn	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xD7

位 7：PREN：内部上拉电阻使能
上拉电阻的位置（D+或 D-）由 SPEED 位决定。
0：禁止内部上拉电阻（器件实际上脱离 USB 网络）。
1：使能内部上拉电阻（器件与 USB 网络连接）。

位 6：PHYEN：物理层使能
该位使能/禁止 USB0 物理层收发器。
0：禁止收发器（挂起）。
1：使能收发器（正常）。

位 5：SPEED：USB0 速度选择
该位选择 USB0 的速度。
0：USB0 工作在低速方式。内部上拉电阻（如果被使能）出现在 D-引脚。
1：USB0 工作在全速方式。内部上拉电阻（如果被使能）出现在 D+引脚。

位 4-3：PHYTST1-0：物理层测试
这两位用于测试 USB0 收发器。

PHYTST[1:0]	方 式	D+	D-
00b	方式 0：正常（非测试方式）	X	X
01b	方式 1：强加差分‘1’	1	0
10b	方式 2：强加差分‘0’	0	1
11b	方式 3：强加单端‘0’	0	0

位 2：DFREC：差分接收器
该位的状态指示出现在 D+和 D-引脚的当前差分值（当 PHYEN = ‘1’ 时）。
0：差分‘0’出现在总线上。
1：差分‘1’出现在总线上。

位 1：Dp：D+信号状态
该位指示 D+引脚的当前逻辑电平。
0：D+信号当前为逻辑 0。
1：D+信号当前为逻辑 1。

位 0：Dn：D-信号状态
该位指示 D-引脚的当前逻辑电平。
0：D-信号当前为逻辑 0。
1：D-信号当前为逻辑 1。

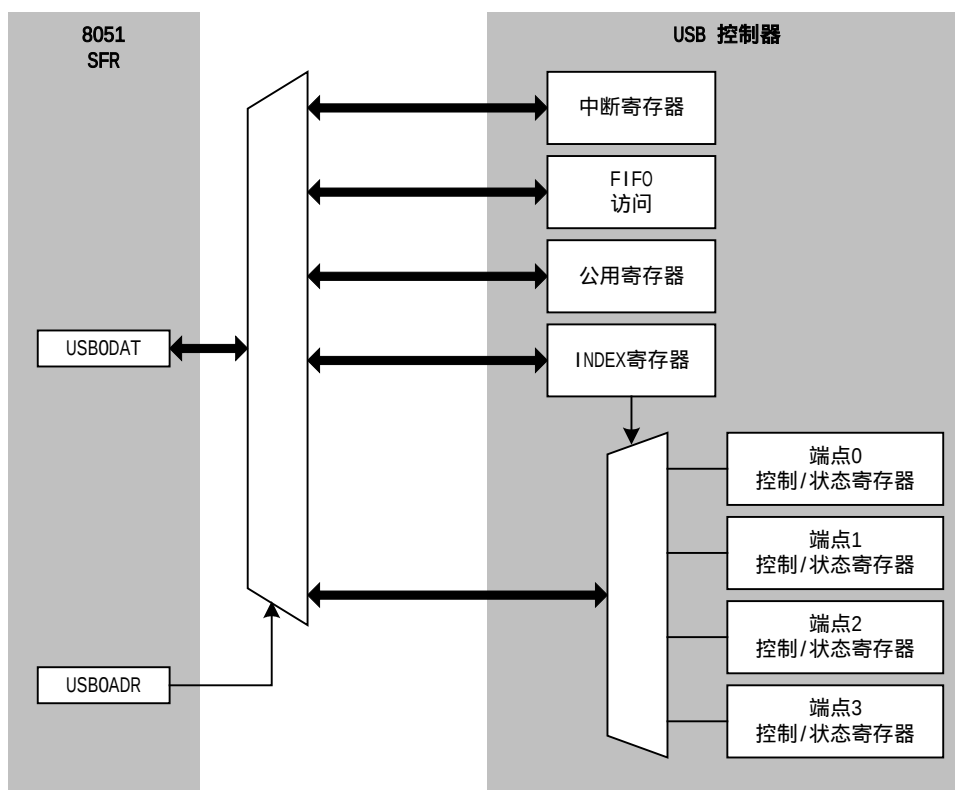
15.3 USB 寄存器访问

对表 15.2 中列出的 USB0 控制器寄存器的访问是通过两个特殊功能寄存器来实现的，这两个特殊功能寄存器是：USB0 地址寄存器（USB0ADR）和 USB0 数据寄存器（USB0DAT）。USB0ADR 寄存器选择读/写 USB0DAT 寄存器时的目标 USB 寄存器。

对端点控制/状态寄存器的访问是通过先写 USB 寄存器 INDEX（用目标端点号）实现的。目标端点号被写入 INDEX 寄存器后，就可以访问与目标端点对应的控制/状态寄存器。表 15.2 列出了端点控制/状态寄存器。

注意：访问 USB 寄存器时，USB 时钟必须是活动的。

图 15.3 USB0 寄存器访问示意图



C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.4 USB0ADR : USB0 间接地址寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
BUSY	AUTORD	USBADDR						00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0x96
位 7 : BUSY : USB0 寄存器读忙标志 该位在间接 USB0 寄存器访问期间使用。软件应通过向该位写 ‘ 1 ’ 来启动对 USB0 寄存器的读操作，目标 USB0 寄存器的地址由 USBADDR 位 (USB0ADR.[5-0]) 给出。可以在同一个对 USB0ADR 的写操作中写目标地址和 BUSY 位。在 BUSY 位被置 ‘ 1 ’ 后，当目标寄存器的数据已经在 USB0DAT 寄存器中准备好时，硬件将 BUSY 位清 ‘ 0 ’。软件在写 USB0DAT 之前应先检查 BUSY 位是否为 ‘ 0 ’。 写： 0：无作用。 1：启动 USB0 间接寄存器读操作，目标地址由 USBADDR 位给出。 读： 0：USB0DAT 寄存器数据有效。 1：USB0 正在访问间接寄存器；USB0DAT 寄存器数据无效。 位 6 : AUTORD : USB0 寄存器自动读标志位 该位用于 FIFO 块读。 0：每次 USB0 间接寄存器读操作都必须写 BUSY 位。 1：当软件读 USB0DAT 时，下一个间接寄存器读操作将被自动启动(USBADDR 位不变)。 位 5-0 : USBADDR : USB0 间接寄存器地址 这些位保持用于间接访问 USB0 寄存器的 6 位地址。表 15.2 列出了 USB0 寄存器和它们的间接地址。读和写 USB0DAT 将访问由 USBADDR 位指向的目标寄存器。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.5 USB0DAT : USB0 数据寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
USB0DAT								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0x97

该 SFR 用于间接读和写 USB0 寄存器。

间接写步骤：

1. 查询 BUSY 位 (USB0ADR.7) => '0'。
2. 将目标 USB0 寄存器地址装入到寄存器 USB0ADR 中的 USBADDR 位。
3. 向 USB0DAT 写数据。
4. 重复上述步骤 (当写同一个 USB0 寄存器时，步骤 2 可以被跳过)。

间接读步骤：

1. 查询 BUSY 位 (USB0ADR.7) => '0'。
2. 将目标 USB0 寄存器地址装入到寄存器 USB0ADR 中的 USBADDR 位。
3. 向寄存器 USB0ADR 中的 BUSY 位写 '1' (步骤 2 和步骤 3 可以在同一个写操作完成)。
4. 查询 BUSY 位 (USB0ADR.7) => '0'。
5. 从 USB0DAT 读数据。
6. 重复从步骤 2 开始的步骤 (当读同一个 USB0 寄存器时，步骤 2 可以被跳过；当 AUTORD 位 (USB0ADR.6) 为逻辑 1 时，步骤 3 可以被跳过)。

图 15.6 INDEX : USB0 端点索引寄存器 (USB 寄存器)

R	R	R	R	R/W	R/W	R/W	R/W	复位值
-	-	-	-	EPSEL				00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x0E

位 7-4：未使用。读 = 0000b，写 = 忽略。

位 3-0：EPSEL：端点选择

当访问 USB0 索引寄存器时，这些位选择目标端点。

索引号	目标端点
0x0	0
0x1	1
0x2	2
0x3	3
0x4 - 0xF	保留

C8051F320/1 全速 USB ISP FLASH 微控制器

表 15.2 USB0 控制器寄存器

USB 寄存器名称	USB 寄存器地址	说 明
中断寄存器		
IN1INT	0x02	端点 0 和端点 1-3 输入中断标志
OUT1INT	0x04	端点 1-3 输出中断标志
CMINT	0x06	公共 USB 中断标志
IN1IE	0x07	端点 0 和端点 1-3 输入中断使能
OUT1IE	0x09	端点 1-3 输出中断使能
CMIE	0x0B	公共 USB 中断使能
公共寄存器		
FADDR	0x00	功能地址
POWER	0x01	电源管理
FRAMEL	0x0C	帧号低字节
FRAMEH	0x0D	帧号高字节
INDEX	0x0E	端点索引选择
CLKREC	0x0F	时钟恢复控制
FIFO _n	0x20 - 0x23	端点 0 – 3 FIFO
索引寄存器		
E0CSR	0x11	端点 0 控制/状态
EINCSRL		端点输入控制/状态低字节
EINCSRH	0x12	端点输入控制/状态高字节
EOUTCSRL	0x14	端点输出控制/状态低字节
EOUTCSRH	0x15	端点输出控制/状态高字节
E0CNT	0x16	端点 0 FIFO 中的接收字节数
EOUTCNTRL		端点输出包计数低字节
EOUTCNTH	0x17	端点输出包计数高字节

15.4 USB 时钟配置

USB0 具有全速和低速 USB 通信功能。通信速度通过特殊功能寄存器 USB0XCN 中的 SPEED 位选择。当 USB0 工作在低速方式时，USB0 时钟必须为 6 MHz；当 USB0 工作在高速方式时，USB0 时钟必须为 48 MHz。时钟选择见“13. 振荡器”。USB0 时钟通过特殊功能寄存器 CLKSEL 选择（见图 13.6）。

时钟恢复电路用输入 USB 数据流调节内部振荡器，这使内部振荡器（和 4 倍时钟乘法器）能满足 USB 时钟的精度要求。时钟恢复功能应在下面的配置中使用。

通信速度	USB 时钟	4 倍时钟乘法器
全速	4 倍时钟乘法器	内部振荡器
低速	内部振荡器/2	N/A

当 USB0 工作在低速方式并使用时钟恢复功能时，软件必须向 CRLOW 写‘1’以使能低速时钟恢复功能。在低速方式一般不需要使用时钟恢复功能。

当 USB 网络上有强噪声时，可以使用单步方式帮助时钟恢复电路锁定。在一般的 USB 环境中不需要（或不建议）使用该方式。

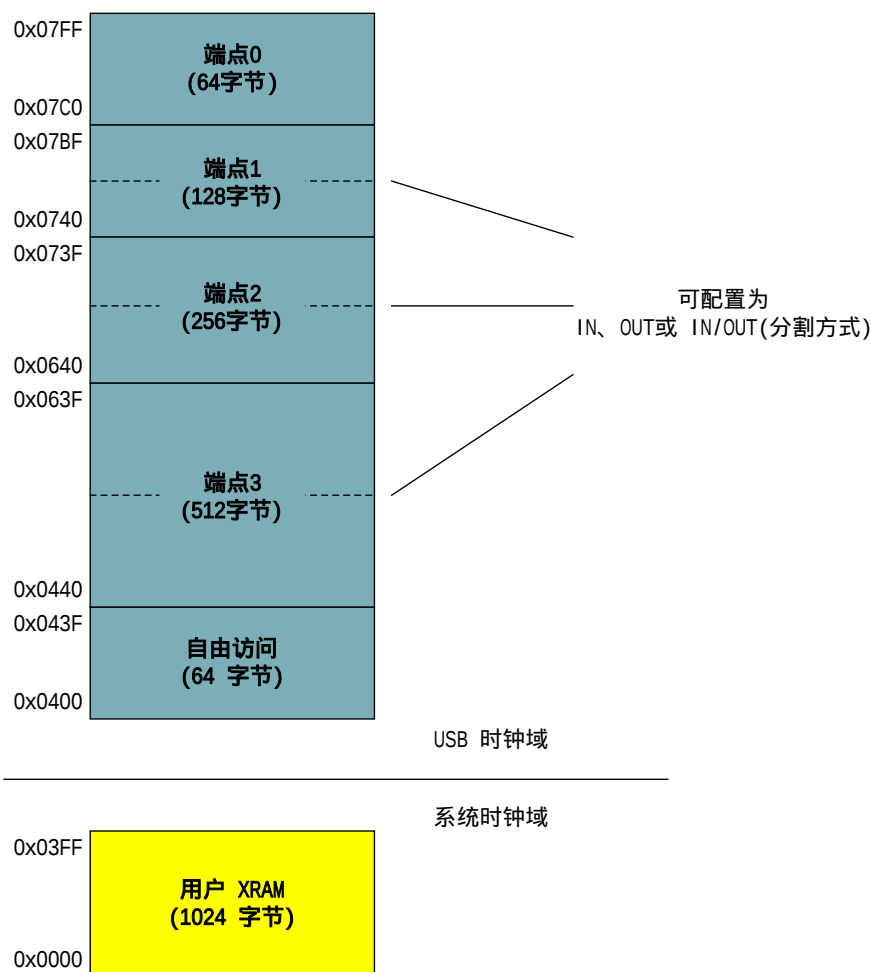
图 15.7 CLKREC：时钟恢复控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CRE	CRSSEN	CRLOW			保留			00001001
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x0F
位 7： CRE：时钟恢复使能 该位使能/禁止 USB 时钟恢复功能。 0：禁止时钟恢复。 1：使能时钟恢复。								
位 6： CRSSEN：时钟恢复单步 该位在时钟恢复期间强制采用“单步”方式进行振荡器校准。 0：正常校准方式。 1：单步方式。								
位 5： CRLOW：低速时钟恢复方式 如果在 USB0 工作于低速方式时使用时钟恢复功能，则该位必须被置‘1’。 0：全速方式。 1：低速方式。								
位 4-0： 保留。读 = 可变，必须写 1001b。								

15.5 FIFO 管理

1024 字节的片内 XRAM 被用作 USB0 的 FIFO 空间。FIFO 空间在端点 0 – 3 之间的分配如图 15.8 所示。分配给端点 1 – 3 的 FIFO 空间可以被配置为输入（IN）、输出（OUT）或两者兼有（分割方式：一半空间给输入，一半空间给输出）。

图 15.8 USB FIFO 分配



15.5.1 FIFO 分割方式

端点 1 – 3 的 FIFO 空间可以被分割成两半，高半部分 FIFO 空间由输入端点使用，低半部分 FIFO 空间由输出端点使用。例如，端点 3 被配置为分割方式，则高 256 字节（0x0540 – 0x063F）为端点 3 输入（IN）使用，低 256 字节（0x0440 – 0x053F）为端点 3 输出（OUT）使用。

如果一个端点的 FIFO 不被配置为分割方式，则端点输入/输出对（IN/OUT）的 FIFO 被组合成一个单独的输入（IN）或输出（OUT）FIFO。在这种情况下，端点输入/输出对在某一时刻只能有一个方向被使用。端点方向（输入/输出）由相应端点的 EINC SRH 寄存器中的 DIRSEL 位决定（见图 15.23）。

15.5.2 FIFO 双缓冲

端点 1 – 3 的 FIFO 空间可以被配置为双缓冲模式。在该模式 ,最大数据包的字节数为 FIFO 空间的一半 ,FIFO 中可以同时有两个数据包。该模式只适用于端点 1 – 3。当一个端点被配置为分割方式时 ,输入 (IN) 端点和/或输出 (OUT) 端点都可以被配置为双缓冲模式。当分割方式未被使能时 ,整个端点 FIFO 可以被设置为双缓冲模式。表 15.3 列出了每种 FIFO 配置配置下的最大数据包的字节数。

表 15.3 FIFO 配置

端点号	分割方式使能 ?	最大输入数据包字节数 (双缓冲禁止/使能)	最大输出数据包字节数 (双缓冲禁止/使能)
0	N/A	64	
1	N	128/64	
	Y	64/32	64/32
2	N	256/128	
	Y	128/64	128/64
3	N	512/256	
	Y	256/128	256/128

15.5.3 FIFO 访问

对每个端点 FIFO 的访问是通过对应的 FIFO_n 寄存器来实现的。对一个端点 FIFO_n 寄存器的读操作将数据从该端点的 FIFO 中读出 ;对一个端点 FIFO_n 寄存器的写操作将一个字节装入该端点的 FIFO。当一个端点 FIFO 被配置为分割方式时 ,对端点 FIFO_n 寄存器的读操作从输出端点 FIFO 中读一个字节 ,对端点 FIFO_n 寄存器的写操作将一个字节装入输入端点 FIFO。

图 15.9 FIFO_n : USB0 端点 FIFO 访问寄存器 (USB 寄存器)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
FIFODATA								00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x20-0x23

使用 USB 地址 0x20 – 0x23 访问 4 对端点 FIFO。

输入/输出端点 FIFO	USB 地址
0	0x20
1	0x21
2	0x22
3	0x23

对 FIFO 地址的写操作将数据装入相应端点的 IN FIFO。
对 FIFO 地址的读操作将数据从相应端点的 OUT FIFO 中读出。

15.6 功能地址

FADDR 寄存器中保存有当前的 USB0 功能地址。USB0 接收到的 SET_ADDRESS 命令中有 7 位功能地址，软件应在接收到该命令时将主机分配的 7 位功能地址写入 FADDR 寄存器。只有在当前传输结束后（一般是在 SET_ADDRESS 命令传输的状态段），写入 FADDR 的新地址才生效（生效之前 USB0 不响应新地址）。当软件向 FADDR 寄存器写入新地址时，UPDATE 位（FADDR.7）被硬件置‘1’。当新地址生效后，硬件将 UPDATE 位清‘0’。

图 15.10 FADDR：USB0 功能地址寄存器（USB 寄存器）

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
UPDATE	功能地址							00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x00
位 7： UPDATE：功能地址更新								
当软件写 FADDR 寄存器时被硬件置‘1’。当新地址生效后，硬件将该位清‘0’。								
0：最后写入 FADDR 的地址已经生效。								
1：最后写入 FADDR 的地址尚未生效。								
位 6-0： 功能地址								
保持 USB0 的 7 位功能地址。当端点 0 收到 SET_ADDRESS 标准设备请求时，软件应写该地址。设备请求完成后新地址生效。								

15.7 功能配置和控制

USB 寄存器 POWER（见图 15.11）用于在器件级控制 USB0（使能/禁止，复位/挂起/恢复控制等）。

USB 复位：当检测到总线上的复位信号时，USBRST 位（POWER.3）被硬件置‘1’。检测到复位信号时会发生以下过程：

- 1．USB0 地址被复位（FADDR = 0x00）。
- 2．端点 FIFO 被清空。
- 3．控制/状态寄存器被复位为 0x00（E0CSR、EINCSRL、EINCSRH、EOUTCSRL、EOUTCSRH）。
- 4．USB 寄存器 INDEX 被复位为 0x00。
- 5．所有 USB 中断（挂起中断除外）都被使能，中断标志被清除。
- 6．产生 USB 复位中断（如果被使能）。

向 USBRST 位写‘1’将产生异步 USB0 复位。在发生异步复位后，所有 USB 寄存器复位到它们的缺省值。

C8051F320/1 全速 USB ISP FLASH 微控制器

挂起方式：在挂起检测被使能的情况下（SUSEN = '1'），当总线上检测到挂起信号时，USB0 进入挂起方式，并产生中断（如果中断被使能，即 SUSINTE = '1'）。挂起中断服务程序（ISR）应执行与应用相关的配置任务，如禁止相应的外设和/或配置低速方式的时钟源。有关内部振荡器配置（包括内部振荡器的挂起方式功能）的详细信息见“13. 振荡器”。

当有下面任何一种情况发生时，USB0 退出挂起方式：1）检测到或产生恢复信号，2）检测到复位信号，3）发生设备或 USB 复位。如果 USB0 被挂起，则在发生上述事件时内部振荡器退出挂起方式。

恢复信号：当总线上检测到恢复信号时，USB0 退出挂起方式，并产生恢复中断（如果中断被使能，即 RESINTE = '1'）。软件可以通过向 RESUME 位（POWER.2）写 '1' 来强制产生远程唤醒。当强制产生远程唤醒时，软件应在远程唤醒被启动（RESUME = '1'）后 10 – 15ms 向 RESUME 位写 '0'，以结束恢复信号。

ISO 更新：当软件向 ISOUP 位（POWER.7）写 '1' 后，ISO 更新功能被使能。在 ISO 更新功能被使能时，写入到一个 ISO IN 端点的新数据包要等到收到一个新的帧起始（SOF）信号后才被发送。如果 ISO 输入端点在收到 SOF 之前收到一个 IN 令牌，USB0 将发送一个长度为零的数据包。当 ISOUP = '1' 时，所有 ISO 端点的 ISO 更新功能都被使能。

USB 使能：在上电复位（POR）后 USB0 被禁止。通过清除 USBINH 位（POWER.4）来使能 USB0。USBINH 一旦被写入 '0'，则只能在发生下列事件之一时被置 '1'：1）上电复位（POR），2）向 USBRST 位（POWER.3）位写 '1' 导致的异步 USB0 复位。

软件应在使能 USB0 之前执行所有的 USB0 配置。建议的配置步骤如下：

1. 选择并使能 USB 时钟源。
2. 通过向 USBRST 位写 '1' 来复位 USB0。
3. 配置并使能 USB 收发器。
4. 执行任何一种 USB0 功能配置（中断、挂起检测）。
5. 通过向 USBINH 位写 '0' 来使能 USB0。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.11 POWER : USB0 电源控制寄存器 (USB 寄存器)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
ISOUD	-	-	USBINH	USBRST	RESUME	SUSMD	SUSEN	00010000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址 : 0x01
位 7 : ISOUD : ISO 更新 该位影响所有 IN 同步端点。 0 : 当软件向 INPRDY 写 ' 1 ' 时, USB0 在收到下一个 IN 令牌后发送数据包。 1 : 当软件向 INPRDY 写 ' 1 ' 时, USB0 等待 SOF 令牌, 然后发送数据包。如果在 SOF 令牌之前收到 IN 令牌, 则 USB0 将发送长度为 0 的数据包。								
位 6-5 : 未使用。读 = 00b。写 = 忽略。								
位 4 : USBINH : USB0 禁止 该位在发生上电复位(POR)或异步 USB0 复位(见位 3 :USBRST)后被置' 1 '。 软件应在完成所有 USB0 和收发器的初始化后将该位清 ' 0 '。软件不能将该位置 ' 1 '。 0 : 使能 USB。 1 : 禁止 USB。所有 USB 通信都被忽略。								
位 3 : USBRST : 复位检测 向该位写 ' 1 ' 强制产生异步 USB0 复位。读该位返回总线复位状态信息。 读 : 0 : 总线上未检测到复位信号。 1 : 总线上未检测到复位信号。								
位 2 : RESUME : 强制恢复 软件可以强制在总线上产生恢复信号, 将 USB0 从挂起方式唤醒。在挂起方式 (SUSMD = ' 1 ') 向该位写 ' 1 ' 将强制 USB0 在总线上产生恢复信号 (一个远程唤醒事件)。软件应在 10 – 15ms 后向 RESUME 位写 ' 0 ' , 以结束恢复信号。软件向 RESUME 位写 ' 0 ' 后将产生中断, 并且硬件将 SUSMD 清 ' 0 '。								
位 1 : SUSMD : 挂起方式 当 USB0 进入挂起方式时, 该位被硬件置 ' 1 '。当软件向 RESUME 位写 ' 0 ' (后面跟随一次远程唤醒) 或在检测到总线上的恢复信号后读 CMINT 寄存器时硬件将该位清 ' 0 '。								
位 0 : SUSEN : 挂起检测使能 0 : 禁止挂起检测。USB0 将忽略总线上的挂起信号。 1 : 使能挂起检测。如果检测到总线上的挂起信号, 则 USB0 进入挂起方式。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.12 FRAMEL : USB0 帧号低字节 (USB 寄存器)

R	R	R	R	R	R	R	R	复位值
帧号低字节								00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址 : 0x0C
位 7-0 : 帧号低字节 该寄存器包含最后接收帧号的位 7-0。								

图 15.13 FRAMEH : USB0 帧号高字节 (USB 寄存器)

R	R	R	R	R	R	R	R	复位值
-	-	-	-	-	帧号高位			00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址 : 0x0D
位 7-3 : 未使用。读 = 00000b。写 = 忽略。 位 2-0 : 帧号高字节 该寄存器包含最后接收帧号的位 10-8。								

15.8 中断

USB0 的中断标志（只读）位于图 15.14 ~ 15.16 所示的 USB 寄存器中。相应的中断使能位位于图 15.17 ~ 15.19 所示的 USB 寄存器中。当这些 USB 中断标志中的任何一个被置‘1’时，就会产生 USB0 中断。通过特殊功能寄存器 EIE1 来使能 USB0 中断（见“9.3 中断系统”）。

注意：读一个中断标志寄存器将使该寄存器中的所有标志位复位为‘0’。

图 15.14 IN1INT：USB0 输入端点中断寄存器（USB 寄存器）

R	R	R	R	R	R	R	R	复位值
	-	-	-	IN3	IN2	IN1	EP0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x02
位 7-4：未使用。读 = 0000b。写 = 忽略。								
位 3：IN3：IN 端点 3 中断标志								
该位在软件读 IN1INT 寄存器后被清‘0’。								
0：IN 端点 3 中断无效。								
1：IN 端点 3 中断有效。								
位 2：IN2：IN 端点 2 中断标志								
该位在软件读 IN1INT 寄存器后被清‘0’。								
0：IN 端点 2 中断无效。								
1：IN 端点 2 中断有效。								
位 1：IN1：IN 端点端点 1 中断标志								
该位在软件读 IN1INT 寄存器后被清‘0’。								
0：IN 端点 1 中断无效。								
1：IN 端点 1 中断有效。								
位 0：EP0：端点 0 中断标志								
该位在软件读 IN1INT 寄存器后被清‘0’。								
0：端点 0 中断无效。								
1：端点 0 中断有效。								

图 15.15 OUT1INT：USB0 输出端点中断寄存器（USB 寄存器）

R	R	R	R	R	R	R	R	复位值
	-	-	-	OUT3	OUT2	OUT1	-	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x04
位 7-4：未使用。读 = 0000b。写 = 忽略。								
位 3：OUT3：OUT 端点 3 中断标志								
该位在软件读 OUT1INT 寄存器后被清‘0’。								
0：OUT 端点 3 中断无效。								
1：OUT 端点 3 中断有效。								
位 2：OUT2：OUT 端点 2 中断标志								
该位在软件读 OUT1INT 寄存器后被清‘0’。								
0：OUT 端点 2 中断无效。								
1：OUT 端点 2 中断有效。								
位 1：OUT1：OUT 端点 1 中断标志								
该位在软件读 OUT1INT 寄存器后被清‘0’。								
0：OUT 端点 1 中断无效。								
1：OUT 端点 1 中断有效。								
位 0：未使用。读 = 0b。写 = 忽略。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.16 CMINT：USB0 公共中断寄存器（USB 寄存器）

R	R	R	R	R	R	R	R	复位值
-	-	-	-	SOF	RSTINT	RSUINT	SUSINT	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x06
位 7-4：未使用。读 = 0000b。写 = 忽略。 位 3：SOF：帧起始中断标志 该位在收到 SOF 令牌时被硬件置‘1’。当硬件期望收到一个 SOF 事件时，就会产生中断，即使实际的 SOF 信号丢失或损坏。 该位在软件读 CMINT 寄存器后被清‘0’。 0：帧起始中断无效。 1：帧起始中断有效。 位 2：RSTINT：复位中断标志 当总线上检测到复位信号时该位被硬件置‘1’。 该位在软件读 CMINT 寄存器后被清‘0’。 0：复位中断无效。 1：复位中断有效。 位 1：RSUINT：恢复中断标志 在 USB0 处于挂起方式期间，当总线上检测到恢复信号时该位被硬件置‘1’。 该位在软件读 CMINT 寄存器后被清‘0’。 0：恢复中断无效。 1：恢复中断有效。 位 0：SUSINT：挂起中断标志 在挂起检测被使能的情况下（寄存器 POWER 中的 SUSEN 位），当总线上检测到挂起信号时该位被硬件置‘1’。 该位在软件读 CMINT 寄存器后被清‘0’。 0：挂起中断无效。 1：挂起中断有效。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.17 IN1IE : USB0 输入端点中断允许寄存器 (USB 寄存器)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	IN3E	IN2E	IN1E	EP0E	00001111
位7	位6	位5	位4	位3	位2	位1	位0	USB地址 : 0x07
位 7-4 : 未使用。读 = 0000b。写 = 忽略。								
位 3 : IN3E : IN 端点 3 中断允许								
0 : 禁止 IN 端点 3 中断。								
1 : 允许 IN 端点 3 中断。								
位 2 : IN2E : IN 端点 2 中断允许								
0 : 禁止 IN 端点 2 中断。								
1 : 允许 IN 端点 2 中断。								
位 1 : IN1 : IN 端点 1 中断允许								
0 : 禁止 IN 端点 1 中断。								
1 : 允许 IN 端点 1 中断。								
位 0 : EP0 : 端点 0 中断允许								
0 : 禁止端点 0 中断。								
1 : 允许端点 0 中断。								

图 15.18 OUTIE : USB0 输出端点中断允许寄存器 (USB 寄存器)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	OUT3E	OUT2E	OUT1E	-	00001110
位7	位6	位5	位4	位3	位2	位1	位0	USB地址 : 0x09
位 7-4 : 未使用。读 = 0000b。写 = 忽略。								
位 3 : OUT3 : OUT 端点 3 中断允许								
0 : 禁止 OUT 端点 3 中断。								
1 : 允许 OUT 端点 3 中断。								
位 2 : OUT2 : OUT 端点 2 中断允许								
0 : 禁止 OUT 端点 2 中断。								
1 : 允许 OUT 端点 2 中断。								
位 1 : OUT1 : OUT 端点 1 中断允许								
0 : 禁止 OUT 端点 1 中断。								
1 : 允许 OUT 端点 1 中断。								
位 0 : 未使用。读 = 0b。写 = 忽略。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.19 CMIE : USB0 公共中断允许寄存器 (USB 寄存器)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	SOFE	RSTINTE	RSUINTE	SUSINTE	00000110
位7	位6	位5	位4	位3	位2	位1	位0	USB地址 : 0x0B
位 7-4 : 未使用。读 = 0000b。写 = 忽略。								
位 3 : SOF : 帧起始中断允许								
0 : 禁止帧起始中断。								
1 : 允许帧起始中断。								
位 2 : RSTINT : 复位中断允许								
0 : 禁止复位中断。								
1 : 允许复位中断。								
位 1 : RSUINT : 恢复中断允许								
0 : 禁止恢复中断。								
1 : 允许恢复中断。								
位 0 : SUSINT : 挂起中断允许								
0 : 禁止挂起中断。								
1 : 允许挂起中断。								

15.9 串行接口引擎

串行接口引擎 (SIE) 执行所有低层 USB 协议，在数据被成功发送或接收后中断处理器。当接收数据时，SIE 在接收完一个完整的数据包后中断处理器；相应的握手信号由 SIE 自动产生。当发送数据时，SIE 在发送完一个完整的数据包并且收到相应的握手信号后中断处理器。

当收到损坏/错误的数据包时，SIE 不中断处理器。

15.10 端点 0

通过 USB 寄存器 E0CSR (见图 15.20) 管理端点 0。访问 E0CSR 寄存器时，INDEX 寄存器必须被装入 0x00 值。

当发生下列情况时产生端点 0 中断：

1. 当收到一个数据包 (OUT 或 SETUP) 并将其装入端点 0 FIFO 时。OPRDY 位 (E0CSR.0) 被硬件置 '1'。
2. 当一个 IN 数据包从端点 0 FIFO 成功卸载并发送到主机时。INPRDY 位被硬件清 '0'。
3. 一次 IN 传输结束 (该中断在数据交换的状态阶段产生)。
4. 在控制传输因违反协议而结束后，硬件将 STSTL 位 (E0CSR.2) 置 '1'。
5. 因控制传输在固件置 '1' DATAEND 位 (E0CSR.3) 之前结束，导致硬件将 SUSPEND 位 (E0CSR.4) 置 '1'。

E0CNT 寄存器 (见图 15.21) 保持端点 0 FIFO 中的接收数据字节数。

硬件将自动检测协议错误并发送 STALL 条件作为应答。固件可以强制产生一个 STALL 条件以放弃当前的传输。当产生 STALL 条件时，STALL 位被置‘1’并产生中断。下述情况可以导致硬件产生 STALL 条件。

1. 当 DATAEND 位被置‘1’后，主机在 OUT 数据阶段发送一个 OUT 令牌。
2. 当 DATAEND 位被置‘1’后，主机在 IN 数据阶段发送一个 IN 令牌。
3. 主机发送一个超过端点 0 最大数据包大小的数据包。
4. 主机在一次 IN 传输的状态段发送一个非零长度的 DATA1 数据包。

固件将 SDSTL 位 (E0CSR.5) 置‘1’。

15.10.1 端点 0 SETUP 传输

所有控制传输都必须从一个 SETUP 包开始。SETUP 包与 OUT 包类似，包含一个由主机发送的 8 字节数据域。包含一个非 8 字节命令域的任何 SETUP 包都会被 USB0 自动丢弃。当 SETUP 包的数据被装入到端点 0 FIFO 时会产生端点 0 中断。软件应从端点 0 FIFO 卸载命令，对命令译码，执行所要求的任务，并将 SOPRDY 位置‘1’以指示执行完 OUT 包服务。

15.10.2 端点 0 IN 传输

当 USB0 收到一个要求发送数据到主机的 SETUP 请求时，主机会发送一个或多个 IN 请求。对于第一个 IN 事务，固件应将一个 IN 数据包装入到端点 0 FIFO 并将 INPRDY 位 (E0CSR.1) 置‘1’。IN 数据包被成功发送后会产生中断。注意：如果在固件将一个数据包装入到端点 0 FIFO 之前收到 IN 请求，则不会产生中断。如果所请求的数据超过了端点 0 最大数据包的大小 (已报告给主机)，则数据应被分成多个包；除最后一个数据包外，每个包应为最大数据包长度。如果所请求的数据是端点 0 最大数据包长度的整数倍，则最后一个数据包应为零长度包，指示传输结束。固件在将一次传输的最后一个数据包装入端点 0 FIFO 之后，应将 DATAEND 位置‘1’。

对于特定的控制传输，在收到第一个 IN 令牌后，端点 0 处于发送方式。在该方式，主机应向端点 0 只发送 IN 令牌。在端点 0 处于发送方式时，如果收到一个 SETUP 或 OUT 令牌，则 SUEND 位 (E0CSR.4) 被置‘1’。

在发生下列事件之前，端点 0 将一直保持在发送方式：

1. USB0 接收一个端点 0 SETUP 或 OUT 令牌。
2. 固件发送一个小于端点 0 最大数据包长度的数据包。
3. 固件发送一个零长度的数据包。

当执行上述的(2)和(3)时，固件应将 DATAEND 位 (E0CSR.3) 置‘1’。

如果在 IN FIFO 中没有准备好的数据包 (INPRDY = ‘0’)，则 SIE 会发送一个 NAK 作为对 IN 令牌的应答。

15.10.3 端点 0 OUT 传输

当 USB0 收到一个要求主机发送数据到 USB0 的 SETUP 请求时，主机会发送一个或多个 OUT 请求。当 USB0 成功接收到一个 OUT 数据包时，硬件将 OPRDY 位 (E0CSR.0) 置 '1' 并产生端点 0 中断。在发生该中断后，固件应将 OUT 数据包从端点 0 FIFO 卸载并将 SOPRDY 位 (E0CSR.6) 置 '1'。

如果所请求的数据超过了端点 0 最大数据包长度，则数据应被分成多个包。如果所请求的数据是端点 0 最大数据包长度的整数倍 (已报告给主机)，则主机发送的最后一个数据包应为零长度包，以指示传输结束。

对于特定的控制传输，在收到第一个 OUT 令牌后，端点 0 处于接收方式。在该方式，主机应向端点 0 只发送 OUT 令牌。在端点 0 处于接收方式时，如果收到一个 SETUP 或 IN 令牌，则 SUEND 位 (E0CSR.4) 被置 '1'。

在发生下列事件之前，端点 0 将一直保持在接收方式：

1. SIE 接收一个 SETUP 或 IN 令牌。
2. 主机发送一个小于端点 0 最大数据包长度的数据包。
3. 主机发送一个零长度的数据包。

当接收到所期望数量的数据时，固件应将 DATAEND 位 (E0CSR.3) 置 '1'。如果主机在固件将 DATAEND 位置 '1' 之后发送一个 OUT 包，则 SIE 将发送一个 STALL 条件。在 STALL 条件被发送后，STSTL 位 (E0CSR.2) 被置 '1' 并产生端点 0 中断。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.20 E0CSR : USB0 端点 0 控制寄存器 (USB 寄存器)

R/W	R/W	R/W	R	R/W	R/W	R/W	R	复位值
SSUEND	SOPRDY	SDSTL	SUEND	DATAEND	STSTL	INPRDY	OPRDY	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x11
位 7 :	SSUEND : SETUP 结束事件服务位							
	写 : 软件应在处理完 SETUP 结束事件 (SUEND) 后将该位置 '1'。当软件向 SSUEND 写 '1' 后, 硬件将 SUEND 位清 '0'。							
	读 : 该位的读出值总是为 '0'。							
位 6 :	SOPRDY : OPRDY 服务位							
	写 : 软件应在处理完接收到的端点 0 数据包后将该位置 '1'。当软件向 SOPRDY 写 '1' 后, 硬件将 OPRDY 位清 '0'。							
	读 : 该位的读出值总是为 '0'。							
位 5 :	SDSTL : 发送 STALL							
	软件可以通过向该位写 '1' 来结束当前的数据传输 (因为错误条件、不希望的传输请求等)。当 STALL 信号被发送后, 硬件将该位清 '0'。							
位 4 :	SUEND : SETUP 结束标志							
	当一次控制传输在软件向 DATAEND 位写 '1' 之前结束时, 硬件将该只读位置 '1'。当软件向 SSUEND 写 '1' 后, 硬件将该位清 '0'。							
位 3 :	DATAEND : 数据结束							
	软件应在下列情况向该位写 '1'。							
	1. 当发送最后一个数据包后固件向 INPRDY 写 '1' 时。							
	2. 当发送一个零长度数据包后固件向 INPRDY 写 '1' 时。							
	3. 当接收完最后一个数据包后固件向 SOPRDY 写 '1' 时。							
	该位被硬件自动清 '0'。							
位 2 :	STSTL : 已发送 STALL							
	发送 STALL 信号后, 硬件将该位置 '1'。该位必须用软件清 '0'。							
位 1 :	INPRDY : 输入 (IN) 数据包准备好							
	软件应在将一个要发送的数据包装入端点 0 FIFO 后向该位写 '1'。在发生下列条件之一时硬件将该位清 '0'。							
	1. 数据包已发送。							
	2. 数据包被一个输入 SETUP 包覆盖。							
	3. 数据包被一个输入 OUT 包覆盖。							
位 0 :	OPRDY : 输出 (OUT) 数据包准备好							
	当收到一个数据包时, 硬件将该只读位置 '1' 并产生中断。该位只在软件向 SOPRDY 位写 '1' 时被清 '0'。							

图 15.21 E0CNT：USB0 端点 0 数据计数寄存器（USB 寄存器）

R	R	R	R	R	R	R	R	复位值
-	E0CNT							00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x16
位 7：未使用。读 = 0b。写 = 忽略。 位 6-0：E0CNT：端点 0 数据计数 该 7 位值指示端点 0 FIFO 中接收的数据字节数。该值只在 OPRDY 位为 ‘1’ 期间有效。								

15.11 配置端点 1~3

对端点 1~3 的配置和控制是通过它们各自的一组控制/状态寄存器来实现的，这些寄存器是：IN 寄存器 EINCSSL、EINCSRH 和 OUT 寄存器 EOUTCSRL、EOUTCSRH。在某一时刻只能有一组端点控制/状态寄存器被映射到 USB 寄存器地址空间，由 INDEX 寄存器定义（见图 15.6）。

如 15.1.1 节所述，端点 1~3 可以被配置为 IN、OUT 或 IN/OUT（分割方式）。由寄存器 EINCSRH 中的 SPLIT 位选择端点方式（分割方式/正常方式）。

当 SPLIT = ‘1’ 时，相应的端点 FIFO 被分割成两部分，IN 和 OUT 管道都可用。

当 SPLIT = ‘0’ 时，相应的端点功能为 IN 或 OUT，端点方向由寄存器 EINCSRH 中的 DIRSEL 位选择。

15.12 控制端点 1~3 输入（IN）

对端点 1~3 输入（IN）的管理是通过 USB 寄存器 EINCSSL 和 EINCSRH 来实现的。所有 IN 端点都可以被用于中断、批量或同步传输。通过向寄存器 EINCSRH 中的 ISO 位写 ‘1’ 来使能同步（ISO）方式。硬件对批量和中断传输的控制与此完全相同。

在发生下述条件之一时，会产生端点 1~3 输入（IN）中断：

- 1．当一个 IN 数据包被成功发送到主机时。
- 2．当目标 FIFO 未空，而软件向 FLUSH 位（EINCSSL.3）写 ‘1’ 时。
- 3．硬件产生一个 STALL 条件。

15.12.1 端点 1~3 输入（IN）中断或批量方式

当 ISO 位（EINCSRH.6）= ‘0’ 时，目标端点工作在批量或中断方式。一旦一个端点被配置为批量/中断 IN 方式（通常在端点 0 的 SET_INTERFACE 命令之后），固件应将一个 IN 数据包装入端点的 IN FIFO 并将 INPRDY 位（EINCSSL.0）位置 ‘1’。在收到一个 IN 令牌后，硬件将发送数据，清除 INPRDY 位，并产生中断。

在没有向端点 FIFO 写任何数据的情况下向 INPRDY 写 ‘1’ 会导致在收到下一个 IN 令牌时发送一个零长度的数据包。

可以通过向 SDSTL 位（EINCSSL.4）写 ‘1’ 来停止批量或中断管道。当 SDSTL = ‘1’

时，硬件对所有的 IN 请求都以 STALL 条件回应。每当硬件产生一个 STALL 条件时，会产生一个中断，且 STSTL 位 (EINCSRL.5) 被置 '1'。STSTL 位必须用固件清 '0'。

当端点 FIFO 中有一个打开的数据包 SLOT 时，硬件自动将 INPRDY 位清 '0'。注意：如果目标端点被使能为双缓冲模式，则固件可以在同一时间向 IN FIFO 装入两个数据包。在这种情况下，在固件将第一个数据包装入 FIFO 并将 INPRDY 位置 '1' 后，硬件会立即清 0 INPRDY 位。在这种情况下没有中断产生，只在数据包已被发送时产生中断。

当固件向 FCDT 位 (EINCSRH.3) 写 '1' 时，每个 IN 数据包的会不停地 Toggle，与从主机接收到的握手信号无关。该特性一般在中断端点中使用，为同步端点提供速率反馈通信。当 FCDT='0' 时，数据 toggle 位只在主机发送了 ACK (对一个 IN 数据包应答) 后被 toggle。

15.12.2 端点 1~3 输入 (IN) 同步方式

当 ISO 位 (EINCSRH.6) 被置 '1' 时，目标端点工作在同步 (ISO) 方式。一旦一个端点被配置为 ISO 输入 (IN) 方式，主机将每帧发送一个 IN 令牌 (数据请求)；数据在每帧中的位置可能不同。因此建议将 ISO IN 端点使能为双缓冲方式。

当端点 FIFO 中有一个打开的数据包 SLOT 时，硬件自动将 INPRDY 位 (EINCSRL.0) 清 '0'。注意：如果目标端点被使能为双缓冲模式，则固件可以在同一时间向 IN FIFO 装入两个数据包。在这种情况下，在固件将第一个数据包装入 FIFO 并将 INPRDY 位置 '1' 后，硬件会立即清 '0' INPRDY 位。在这种情况下没有中断产生，只在数据包已被发送时产生中断。

如果在 USB0 从主机接收一个 IN 令牌时，在端点 FIFO 中没有已准备好的数据包，则 USB0 会发送一个零长度的数据包并将 UNDRUN 位 (EINCSRL.2) 置 '1'。

ISO 更新功能 (见 15.7 节) 在开始一个双缓冲的 ISO IN 端点时可能有用。如果在固件向端点 FIFO 写第一个数据包时主机已经建立了 ISO IN 管道 (已经开始发送 IN Token)，则在固件向 FIFO 写完第二个 (双缓冲) 数据包之前，下一个 IN 令牌可以到达，第一个数据包已经发送。ISO 更新功能保证写到端点 FIFO 中的任何数据包不在当前帧发送，只在接收到 SOF 信号后发送该数据包。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.22 EINCSSL : USB0 IN 端点控制寄存器低字节 (USB 寄存器)

R	W	R/W	R/W	W	R/W	R/W	R/W	复位值
-	CLRDT	STSTL	SDSTL	FLUSH	UNDRUN	FIFONE	INPRDY	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址 : 0x11
位 7 : 未使用。读 = 0b , 写 = 忽略。 位 6 : CLRDT : 数据 Toggle 清除位 写 : 软件应向该位写 ' 1 ' , 以将 IN 端点数据 toggle 复位为 ' 0 '。 读 : 该位的读出值总是为 ' 0 '。 位 5 : STSTL : 已发送 STALL 当 STALL 信号被发送后 , 硬件将该位置 ' 1 '。FIFO 被清空 , INPRDY 位被清 ' 0 '。 该标志必须用软件清 ' 0 '。 位 4 : SDSTL : 发送 STALL 软件应向该位写 ' 1 ' 以产生 STALL 信号作为对一个 IN 令牌的应答。软件应向该位写 ' 0 ' 以结束 STALL 信号。该位对 ISO 方式没有影响。 位 3 : FLUSH : FIFO 清空 向该位写 ' 1 ' 将从 IN 端点 FIFO 中清除待发送的下一个数据包。FIFO 指针被复位 , INPRDY 位被清除。如果 FIFO 中包含多个数据包 , 软件必须对每个数据包向 FLUSH 写 ' 1 '。当 FIFO 清空完成后 , 硬件将 FLUSH 位清 ' 0 '。 位 2 : UNDRUN : 数据不足 该位的功能取决于 IN 端点的方式 : ISO : 在 INPRDY = ' 0 ' 并且收到一个 IN 令牌后发送了一个零长度数据包时 , 该位被置 ' 1 '。 中断/批量 : 当作为对一个 IN 令牌的应答而返回一个 NAK 时该位被置 ' 1 '。 该位必须用软件清 ' 0 '。 位 1 : FIFONE : FIFO 非空 0 : IN 端点 FIFO 为空。 1 : IN 端点 FIFO 包含一个或多个数据包。 位 0 : INPRDY : IN 数据包准备好 软件应在将一个数据包装入 IN 端点 FIFO 后向该位写 ' 1 '。在发生下列条件之一时硬件将该位清 ' 0 '。 1 . 数据包已发送。 2 . 双缓冲被使能 (DBIEN = ' 1 ') 且有一个打开的 FIFO 数据包 SLOT。 3 . 如果端点处于同步方式并且 ISOUD = ' 1 ' , 在收到下一个 SOF 之前 INPRDY 的读出值总是为 0。 当硬件因发送数据包而将 INPRDY 清 ' 0 ' 时会产生中断 (如果被允许)。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.23 EINC5RH : USB0 IN 端点控制寄存器高字节 (USB 寄存器)

R/W	W	R/W	R	R/W	R/W	R	R	复位值
DBIEN	ISO	DIRSEL	-	FCDT	SPLIT	-	-	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x12
位 7 : DBIEN : IN 端点双缓冲使能 0 : 禁止所选 IN 端点的双缓冲功能。 1 : 使能所选 IN 端点的双缓冲功能。 位 6 : ISO : 同步传输使能 该位使能/禁止在当前端点进行同步传输 0 : 端点被配置为批量/中断传输。 1 : 端点被配置为同步传输。 位 5 : DIRSEL : 端点方向选择 该位只在所选 FIFO 为不分割方式 (SPLIT = ' 0 ') 时有效。 0 : 选择端点方向为 OUT。 1 : 选择端点方向为 IN。 位 4 : 未使用。读 = 0b , 写 = 忽略。 位 3 : FCDT : 强制数据 Toggle。 0 : 端点数据 toggle 只在发送完一个数据包后收到 ACK 时切换。 1 : 端点数据 toggle 在每发送完一个数据包后被强制切换, 不管是否收到 ACK。 位 2 : SPLIT : FIFO 分割使能 当 SPLIT = ' 1 ' 时, 所选端点的 FIFO 被分割。所选 FIFO 的高半部分为 IN 端点使用, 所选 FIFO 的低半部分为 OUT 端点使用。 位 1-0 : 未使用。读 = 00b , 写 = 忽略。								

15.13 控制端点 1~3 输出 (OUT)

对端点 1~3 输出 (OUT) 的管理是通过 USB 寄存器 EOUTCSRL 和 EOUTCSRH 来实现的。所有 OUT 端点都可以被用于中断、批量或同步传输。通过向寄存器 EOUTCSRH 中的 ISO 位写 '1' 来使能同步 (ISO) 方式。硬件对批量和中断传输的控制与此完全相同。

在发生下述条件之一时, 会产生端点 1~3 输出 (OUT) 中断:

1. 硬件将 OPRDY 位 (EOUTCSRL.0) 置 '1'。
2. 硬件产生一个 STALL 条件。

15.13.1 端点 1~3 输出 (OUT) 中断或批量方式

当 ISO 位 (EOUTCSRH.6) = '0' 时, 目标端点工作在批量或中断方式。一旦一个端点被配置为批量/中断输出 (OUT) 方式 (通常在端点 0 的 SET_INTERFACE 命令之后), 硬件将 OPRDY 位 (EOUTCSRL.0) 位置 '1' 并在收到一个 OUT 令牌和数据包后产生中断。当前 OUT 数据包 (已准备好从 FIFO 卸载的数据包) 中的字节数由寄存器 EOUTCNTH 和 EOUTCNTL 给出。响应该中断时, 固件应将数据包从 OUT FIFO 中卸载并将 OPRDY 位清 '0'。

可以通过向 SDSTL 位 (EOUTCSRL.5) 写 '1' 来停止批量或中断管道。当 SDSTL = '1' 时, 硬件对所有的 OUT 请求都以 STALL 条件回应。每当硬件产生一个 STALL 条件时, 会产生一个中断, 且 STSTL 位 (EOUTCSRL.6) 被置 '1'。STSTL 位必须用固件清 '0'。

当 OUT FIFO 中有一个准备好的数据包时, 硬件自动将 OPRDY 位置 '1'。注意: 如果目标端点被使能为双缓冲模式, 则 OUT FIFO 中可以同时有两个数据包准备好。在这种情况下, 在固件将第一个数据包卸载并将 OPRDY 位清 '0' 后, 硬件会立即将 OPRDY 位置 '1'。在这种情况下会产生第二个中断。

15.13.2 端点 1~3 输出 (OUT) 同步方式

当 ISO 位 (EOUTCSRH.6) 被置 '1' 时, 目标端点工作在同步 (ISO) 方式。一旦一个端点被配置为 ISO OUT 方式, 主机将每帧发送一个数据包; 但数据包在每帧中的位置可能不同。因此建议将 ISO OUT 端点使能为双缓冲方式。

每次收到一个数据包时, 硬件将接收到的数据包装入端点 FIFO, 将 OPRDY 位 (EOUTCSRL.0) 置 '1', 并产生中断 (如果被允许)。固件一般用该中断将数据包从端点 FIFO 卸载, 并将 OPRDY 位清 '0'。

如果在端点 FIFO 没有空间的情况下收到一个数据包, 则会产生中断, 并且 OVRUN 位 (EOUTCSRL.2) 被置 '1'。如果 USB0 接收到一个有 CRC 错误的 ISO 数据包, 则该数据包被装入端点 FIFO, OPRDY 被置 '1', 产生中断 (如果被允许), DATAERR 位 (EOUTCSRL.3) 被置 '1'。软件应在每次将数据包从 ISO OUT 端点 FIFO 卸载后检查 DATAERR 位。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.24 EOUTCSRL : USB0 OUT 端点控制寄存器低字节 (USB 寄存器)

W	R/W	R/W	W	R	R/W	R	R/W	复位值
CLRDT	STSTL	SDSTL	FLUSH	DATERR	OVRUN	FIFOFUL	OPRDY	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x14
位 7 :	CLRDT : 数据 Toggle 清除位 写：软件应向该位写 ‘ 1 ’，以将 OUT 端点数据 toggle 复位为 ‘ 0 ’。 读：该位的读出值总是为 ‘ 0 ’。							
位 6 :	STSTL : 已发送 STALL 当 STALL 信号被发送后，硬件将该位置 ‘ 1 ’。该标志必须用软件清 ‘ 0 ’。							
位 5 :	SDSTL : 发送 STALL 软件应向该位写 ‘ 1 ’ 以产生 STALL 信号。软件应向该位写 ‘ 0 ’ 以结束 STALL 信号。该位对 ISO 方式没有影响。							
位 4 :	FLUSH : FIFO 清空 向该位写 ‘ 1 ’ 将从 OUT 端点 FIFO 中清除下一个数据包。FIFO 指针被复位，OPRDY 位被清除。如果 FIFO 中包含多个数据包，软件必须对每个数据包向 FLUSH 写 ‘ 1 ’。当 FIFO 清空完成后，硬件将 FLUSH 位清 ‘ 0 ’。							
位 3 :	DATERR : 数据错误 在 ISO 方式，如果接收到的数据包有 CRC 或位填充错误，该位被硬件置 ‘ 1 ’。当软件清除 OPRDY 时，该位被清 ‘ 0 ’。该位只在 ISO 方式有效。							
位 2 :	OVRUN : 数据溢出 当一个输入数据包不能被装入到 OUT 端点 FIFO 时，该位被硬件置 ‘ 1 ’。该位只在 ISO 方式有效。该位必须用软件清 ‘ 0 ’。 0：无数据溢出。 1：自该标志最后一次被清除以来，因 FIFO 已满导致数据包丢失。							
位 1 :	FIFOFUL : OUT FIFO 满标志 该位指示 OUT FIFO 的内容。如果端点被使能为双缓冲模式 (DBIEN = ‘ 1 ’)，当 FIFO 中有两个数据包时 FIFO 已满。如果 DBIEN = ‘ 0 ’)，当 FIFO 中有一个数据包时 FIFO 已满。 0：OUT 端点 FIFO 未滿。 1：OUT 端点 FIFO 已滿。							
位 0 :	OPRDY : 输出 (OUT) 数据包准备好 当有数据包可用时硬件将该位置 ‘ 1 ’。软件应在将每个数据包从 OUT 端点 FIFO 卸载后将该位清 ‘ 0 ’。							

C8051F320/1 全速 USB ISP FLASH 微控制器

图 15.25 EOUTCSRH : USB0 IN 端点控制寄存器高字节 (USB 寄存器)

R/W	R/W	R	R	R	R	R	R	复位值
DBOEN	ISO	-	-	-	-	-	-	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x15
位 7 : DBOEN : OUT 端点双缓冲使能 0 : 禁止所选 OUT 端点的双缓冲功能。 1 : 使能所选 OUT 端点的双缓冲功能。 位 6 : ISO : 同步传输使能 该位使能/禁止在当前端点进行同步传输 0 : 端点被配置为批量/中断传输。 1 : 端点被配置为同步传输。 位 5-0 : 未使用。读 = 000000b , 写 = 忽略。								

图 15.26 EOUTCNTH : USB0 OUT 端点计数器寄存器低字节 (USB 寄存器)

R	R	R	R	R	R	R	R	复位值
EOCL								00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x16
位 7-0 : EOCL : OUT 端点计数器低字节 EOCL 保持在当前 OUT 端点 FIFO 中的最后一个接收数据包之 10 位数据字节数的低 8 位。该数值只在 OPRDY = ' 1 ' 时有效。								

图 15.27 EOUTCNTH : USB0 OUT 端点计数器寄存器高字节 (USB 寄存器)

R	R	R	R	R	R	R	R	复位值
-	-	-	-	-	-	EOCH		00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x17
位 7-2 : 未使用。读 = 000000b , 写 = 忽略。 位 1-0 : EPOCH : OUT 端点计数器高字节 EOCH 保持在当前 OUT 端点 FIFO 中的最后一个接收数据包之 10 位数据字节数的高 2 位。该数值只在 OPRDY = ' 1 ' 时有效。								

C8051F320/1 全速 USB ISP FLASH 微控制器

表 15.4 USB 收发器电气特性

VDD=3.0 ~ 3.6V, -40°C到+85°C (除非特别说明)

参 数	符号	条 件	最小值	典型值	最大值	单 位
发送器						
输出高电平	V _{OH}		2.8			V
输出低电平	V _{OL}				0.8	V
输出交叉点	V _{CRS}		1.3		2.0	V
输出电阻	Z _{DRV}	驱动高电平 驱动低电平		38 38		Ω
上拉电阻	R _{PU}	全速 (D+上拉) 低速 (D-上拉)	1.425	1.5	1.575	kΩ
输出上升时间	T _R	全速 低速	75 4		300 20	ns
输出下降时间	T _F	全速 低速	75 4		300 20	ns
接收器						
差分输入敏感度	V _{DI}	(D+) - (D-)	0.2			V
差分输入共模范围	V _{CM}		0.8		2.5	V
输入漏电流	I _L	上拉电阻禁止		<1.0		μA

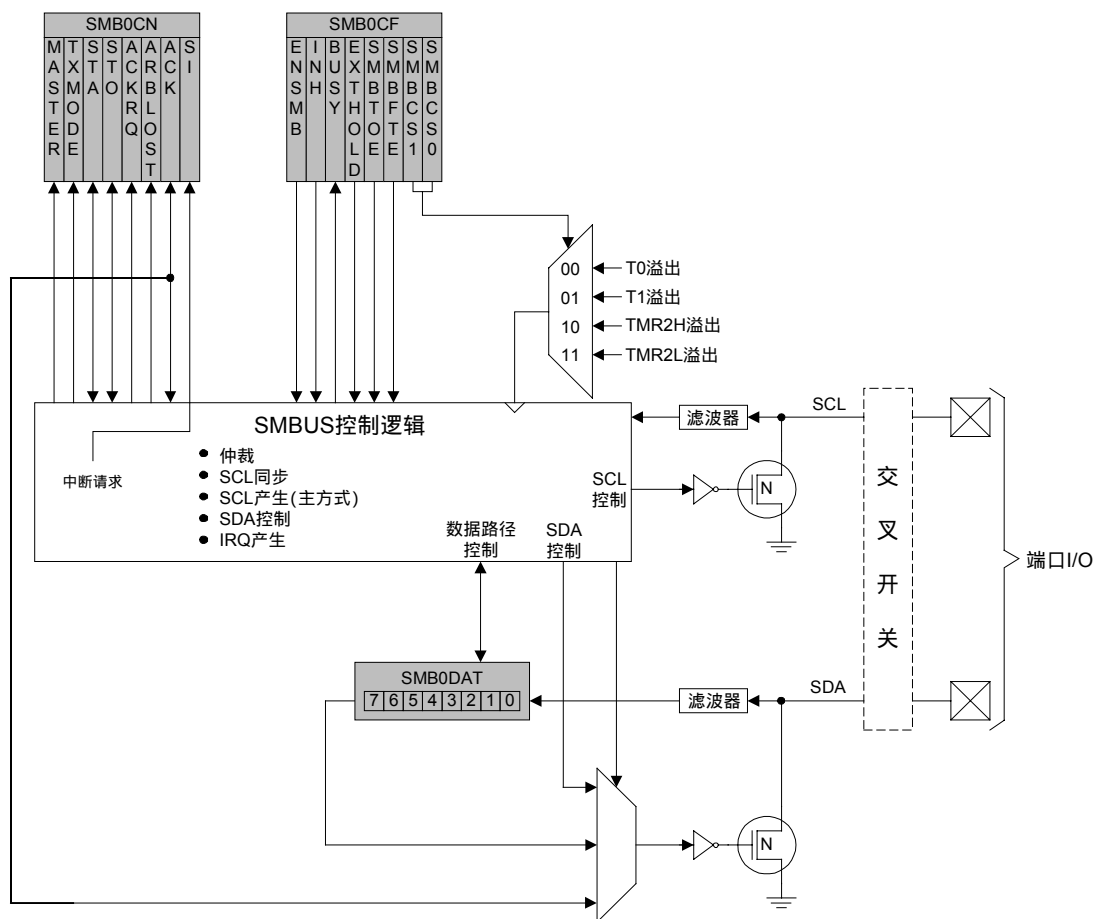
注：时序图和符号定义请参见 USB 规范。

16. SMBus

SMBus I/O接口是一个双线的双向串行总线。SMBus完全符合系统管理总线规范 1.1 版，与I²C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。有三个与 SMBus 相关的特殊功能寄存器：SMB0CF 配置 SMBus，SMB0CN 控制 SMBus 的状态，SMB0DAT 为数据寄存器，用于发送和接收 SMBus 数据和从地址。

图 16.1 SMBus 原理框图



16.1 支持文档

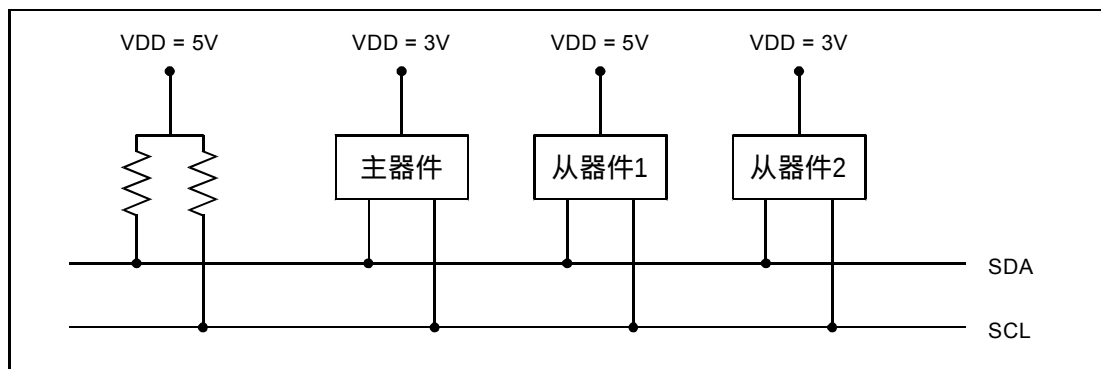
假设读者熟悉或有条件阅读下列支持文档：

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

16.2 SMBus 配置

图 16.2 给出了一个典型的 SMBus 配置。SMBus 接口的工作电压可以在 3.0V 和 5.0V 之间，总线上不同器件的工作电压可以不同。SCL（串行时钟）和 SDA（串行数据）线是双向的，必须通过一个上拉电阻或等效电路将它们连到电源电压。连接在总线上的每个器件的 SCL 和 SDA 都必须是漏极开路或集电极开路的，因此当总线空闲时，这两条线都被拉到高电平。总线上的最大器件数只受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 16.2 典型 SMBus 配置



16.3 SMBus 操作

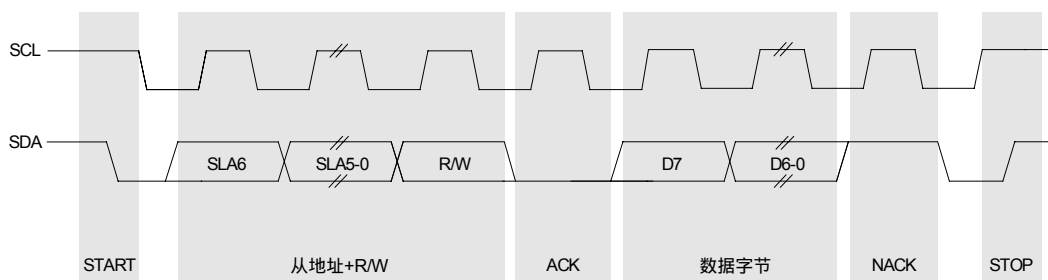
有两种可能的数据传输类型：从主发送器到所寻址的从接收器（写）和从被寻址的从发送器到主接收器（读）。这两种数据传输都由主器件启动，主器件还在 SCL 上提供串行时钟。SMBus 接口可以工作在主方式或从方式，总线上可以有多个主器件。如果两个或多个主器件同时启动数据传输，仲裁机制将保证有一个主器件会赢得总线。注意：没有必要在一个系统中指定某个器件作为主器件；任何一个发送起始条件（START）和从器件地址的器件就成为该次数据传输的主器件。

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 16.3）来确认（ACK）。如果接收器件不确认，则发送器件将读到一个“非确认”（NACK），这用 SCL 高电平期间的 SDA 高电平表示。

方向位（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，然后发送地址和方向位。如果本次数据传输是一个从主器件到从器件的写操作，则主器件每发送一个数据字节后等待来自从器件的确认。如果是一个读操作，则由从器件发送数据并等待主器件的确认。在数据传输结束时，主器件产生一个停止条件，结束数据交换并释放总线。图 16.3 示出了一次典型的 SMBus 数据传输过程。

图 16.3 SMBus 数据传输



16.3.1 总线仲裁

一个主器件只能在总线空闲时启动一次传输。在一个停止条件之后或 SCL 和 SDA 保持高电平已经超过了指定时间（见“16.3.4 SCL 高电平超时”），则总线是空闲的。两个或多个主器件可能在同一时刻启动数据传输，所以使用仲裁机制迫使一个主器件放弃总线。这些主器件继续发送起始条件，直到其中一个主器件发送高电平而另一个主器件在 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，而未赢得总线的主器件成为从器件。该仲裁机制是非破坏性的：总会有一个器件赢得总线，不会发生数据丢失。

16.3.2 时钟低电平扩展

SMBus 提供一种与 I²C 类似的同步机制，允许不同速度的器件共存于同一个总线上。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。从器件可以保持 SCL 为低电平以扩展时钟低电平时间，这实际上相当于降低了串行时钟频率。

16.3.3 SCL 低电平超时

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

当 SMB0CF 中的 SMBTOE 位被置位时，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）。

16.3.4 SCL 高电平（SMBus 空闲）超时

SMBus 标准规定：如果一个器件保持 SCL 和 SDA 线为高电平的时间超过 50 微秒，则认为总线处于空闲状态。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。

16.4 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。

每次数据字节或从地址传输都产生 SMBus 中断。发送数据时在 ACK 周期后产生中断，使软件能读取接收到的 ACK 值；接收数据时在 ACK 周期之前产生中断，使软件能确定要发出的 ACK 值。有关传输时序的详细信息见“16.5 SMBus 传输方式”。

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。

SMBus 配置选项包括：

1. 超时检测（SCL 低电平超时和/或总线空闲超时）；
2. SDA 建立和保持时间扩展；
3. 从事件允许/禁止
4. 时钟源选择。

这些选项在 SMB0CF 寄存器中设定，见“16.4.1 SMBus 配置寄存器”。

16.4.1 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 时序和超时选项。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 16.1），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，该最小时间由下面的方程 16.1 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。

表 16.1 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

方程 16.1 最小 SCL 高和低电平时间

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

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。

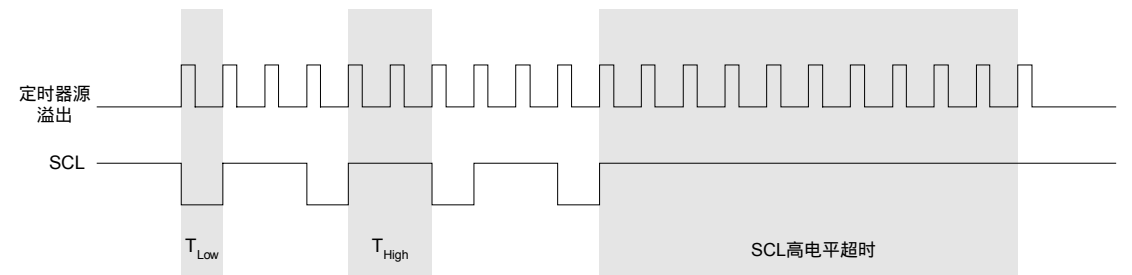
所选择的时钟源应被配置为能产生由方程 16.1 所定义的最小 SCL 高电平和低电平时间。当接口工作在主方式时（并且 SCL 不被总线上的任何其它器件驱动），典型的 SMBus 位速率可由下面的方程 16.2 估算：

方程 16.2 典型 SMBus 位速率

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

图 16.4 给出了由方程 16.2 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 16.1 定义的极限值。

图 16.4 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 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 SCL 下降沿过去之后 SDA 继续保持稳定的最小时间。EXTHOLD 位应被置 ‘1’，以保证最小建立和保持时间符合 SMBus 规范，SMBus 规定的最小建立和保持时间分别为 250ns 和 300ns。表 16.2 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 16.2 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

[†]发送ACK位和所有数据传输中MSB的建立时间。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。

当 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。SMBus 接口在 SCL 为高电平时强制重装定时器 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 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 总线超时检测。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 16.4），总线将被视为空闲。当检测到空闲超时，SMBus 接口的响应就如同检测到一个停止条件（产生一个中断，STO 被置位）。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 16.5 SMB0CF：SMBus 配置寄存器

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	SFR地址： 0xC1

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，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。

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

该位允许/禁止 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定时器 3，并允许定时器 3 在 SCL 为低电平时开始计数。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 复位。

位 2：

SMBFTE：SMBus 空闭超时检测允许位

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 16.1 配置所选器件的时钟源。

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

16.4.2 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 (见图 16.6)。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 和 TXMODE 分别指示主/从状态和发送/接收方式。

STA 和 STO 指示自上次 SMBus 中断以来检测到一个起始 (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。当总线空闲时, 向 STA 写 '1' 将使 SMBus 接口进入主方式并产生一个起始条件。在产生起始条件后 STA 不能由硬件清除, 必须用软件清除。在主方式, 向 STO 写 '1' 将使硬件产生一个停止条件, 并在下一个 ACK 周期之后结束当前的数据传输。如果 STA 和 STO 都被置位 (在主方式), 则发送一个停止条件后再发送一个起始条件。

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。

ARBLOST 位指示 SMBus 接口是否在一次总线竞争中失败。当接口工作在发送方式时(主或从), 可能出现这种情况。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 16.3。

注意: 当 SI 标志被置 '1' 时, SMBus 接口暂停工作; SCL 线被保持为低电平, 总线状态被冻结, 直到 SI 被清 '0' 为止。

表 16.3 列出了影响 SMB0CN 寄存器中各个位的硬件源。有关 SMBus 的状态请参见表 16.4 的 SMBus 状态译码表。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 16.6 SMB0CN : SMBus 控制寄存器

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	SFR地址： (可位寻址) 0xC0
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 : 当工作在主方式时, 若总线空闲, 则发送出一个起始条件 (如果总线不空闲, 在收到停止条件或检测到超时后再发送起始条件)。当工作在主方式时, 如果 STA 被软件置 '1', 在下一个 ACK 周期之后将产生一个重复起始条件。 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 : 将 STO 置为逻辑 '1' 将导致发送一个停止条件 (在下一个 ACK 周期之后)。在产生停止条件之后, 硬件将 STO 清为逻辑 '0'。如果 STA 和 STO 都被置 '1', 则发送一个停止条件后再发送一个起始条件。 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 当 SMBus 接收到一个字节并需要向 ACK 位写 ACK 响应值时, 该只读位被硬件置 '1'。							
位 2 :	ARBLOST : SMBus 竞争失败标志 当 SMBus 作为发送器在总线竞争中失败时该只读位被置 '1'。在从方式时, 竞争失败表示发生了总线错误条件。							
位 1 :	ACK : SMBus 确认标志 该位定义要发出的 ACK 电平和记录接收的 ACK 电平。应在每接收到一个字节后写 ACK 位 (当 ACKRQ = 1 时), 或在发送一个字节后读 ACK 位。 0 : 接收到 "非确认" (在发送器方式) 或将发出 "非确认" (在接收器方式)。 1 : 接收到 "确认" (在发送器方式) 或将发出 "确认" (在接收器方式)。							
位 0 :	SI : SMBus 中断标志 当出现表 16.3 列出的条件时该位被硬件置 '1'。SI 只能用软件清除。当 SI 被置 '1' 时, SCL 被保持为低电平, SMBus 被冻结。							

C8051F320/1 全速 USB ISP FLASH 微控制器

表 16.3 影响 SMB0CN 的硬件源

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。 - 在总线竞争中失败。
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 帧开始之前没写 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。	每个 ACK 周期之后。
ARBLOST	- 当 STA 为 '0' 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 - 接收到一个字节。 - 在起始条件或重复起始条件之后接收到一个从地址字节+R/W。 - 收到一个停止条件。	必须用软件清除。

16.4.3 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 ’ 时数据是稳定的，此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，因为硬件可能正在对该寄存器中的数据字节进行移入或移出操作。

SMB0DAT 中的数据总是先移出 MSB。在收到一个字节后，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。

图 16.7 SMB0DAT：SMBus 数据寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xC2
位 7-0： SMB0DAT：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。								

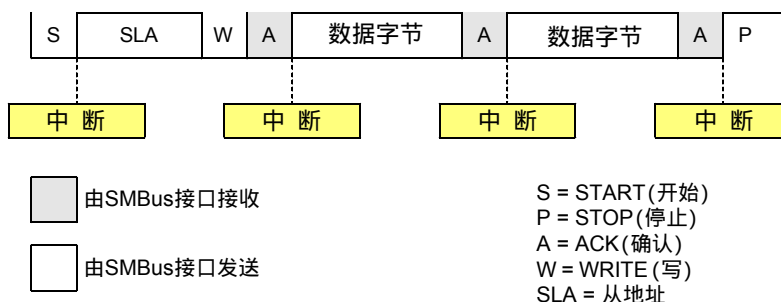
16.5 SMBus 传输方式

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；但作为接收器时中断在 ACK 周期之前产生，作为发送器时中断在 ACK 周期之后产生。

16.5.1 主发送器方式

在 SDA 上发送串行数据，在 SCL 上输出串行时钟。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。在主发送器方式数据方向位(R/W)应为逻辑 ‘ 0 ’，表示这是一个“写”操作。主发送器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，从器件发出确认位。当 STO 位被置 ‘ 1 ’ 并产生一个停止条件后，串行传输结束。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 16.8 给出了典型的主发送器时序，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。

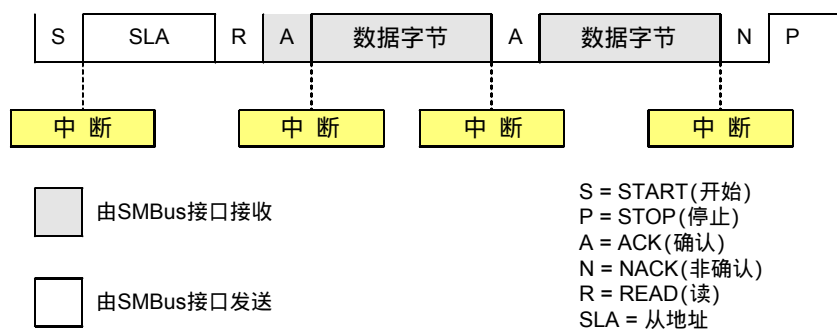
图 16.8 典型的主发送器时序



16.5.2 主接收器方式

在 SDA 上接收串行数据，在 SCL 上输出串行时钟。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’，表示这是一个“读”操作。接着从 SDA 接收来自从器件的串行数据并在 SCL 上输出串行时钟。从器件发送一个或多个字节的串行数据。每收到一个字节后，ACKRQ 被置‘1’并产生一个中断。软件必须写 ACK 位（SMB0CN.1），以定义要发出的确认值（注：向 ACK 位写‘1’产生一个 ACK，写‘0’产生一个 NACK）。软件应在接收到最后一个字节后向 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。注意：在主接收器方式，如果执行 SMB0DAT 写操作，接口将切换到主发送器方式。图 16.9 给出了典型的主接收器时序，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”中断发生在 ACK 周期之前。

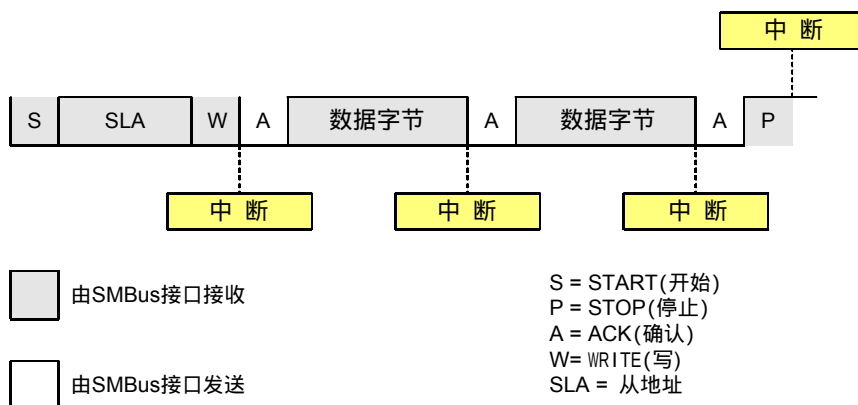
图 16.9 典型的主接收器时序



16.5.3 从接收器方式

在 SDA 上接收串行数据，在 SCL 上接收串行时钟。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。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。如果接收到的从地址被忽略，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。在每接收到一个字节后，软件必须向 ACK 位写 ACK 或 NACK。在收到主器件发出的停止条件后，SMBus 接口退出从发送器方式。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 16.10 给出了典型的从接收器时序，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“数据字节传输中断”发生在 ACK 周期之前。

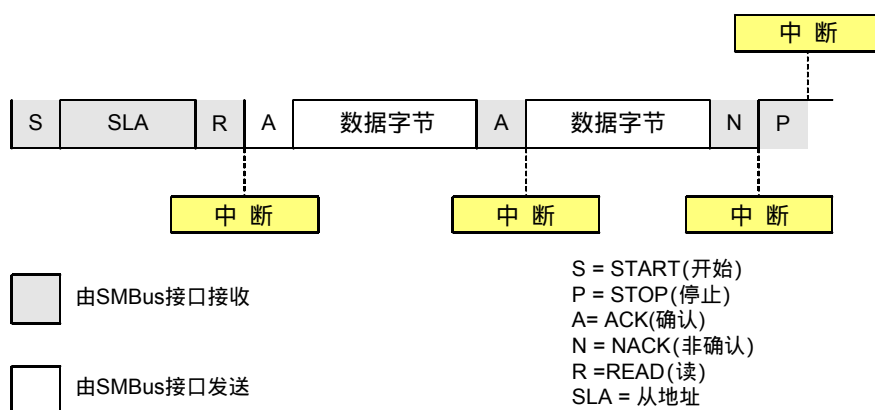
图 16.10 典型的从接收器时序



16.5.4 从发送器方式

在 SDA 上发送串行数据，在 SCL 上接收串行时钟。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)。在进入从发送器方式时，会产生一个中断，并且 ACKRQ 位被置 '1'。软件用一个 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。如果接收到的从地址被忽略，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，软件应向 SMB0DAT 写入待发送的数据，SMBus 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，主器件发出确认位。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)。在收到主器件发出的停止条件后，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 16.11 给出了典型的从发送器时序。注意，在该方式下“数据字节传输”中断发生在 ACK 周期之后。

图 16.11 典型的从发送器时序



16.6 SMBus 状态译码

读 SMB0CN 寄存器可以得到 SMBus 的当前状态。在下面的表 16.4 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允许的，但不符合 SMBus 规范。

表 16.4 SMBus 状态译码

方式	读取值				SMBus 的当前状态	典型响应选项	写入值		
	状态向量	ACKRQ	ARBLOST	ACK			STA		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	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
		0	0	1	数据或地址字节已发出；收到 ACK。	放弃发送。	0	1	X
						将下一字节装入到 SMB0DAT。	0	0	X
						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
						切换到主接收器方式（清除 SI，不向 SMB0DAT 写新数据）。	0	0	X
主接收器	1000	1	0	X	收到数据字节；请求确认。	确认接收字节；读 SMB0DAT。	0	0	1
						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
						发 ACK 后再发重复起始条件。	1	0	1
						发 NACK，表示这是最后一个字节，接着发重复起始条件。	1	0	0
						发 ACK 并切换到主发送器方式（在清除 SI 之前写 SMB0DAT）。	0	0	
						发 NACK 并切换到主发送器方式（在清除 SI 之前写 SMB0DAT）。	0	0	0

C8051F320/1 全速 USB ISP FLASH 微控制器

表 16.4 SMBus 状态译码 (续)

方式	读取值				SMBus 的当前状态	典型响应选项	写入值		
	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
		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
		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
	0101	0	X	X	检测到停止条件。	不需任何操作（传输结束）。	0	0	X
从接收器	0010	1	0	X	接收到从地址；请求确认。	对接收到的地址进行确认。	0	0	1
		1	1	X	竞争主器件失败；收到从地址；请求确认。	不对接收到的地址进行确认。	0	0	0
						对接收到的地址进行确认。	0	0	1
						不对接收到的地址进行确认。	0	0	0
	0010	0	1	X	试图发送重复起始条件时竞争失败。	重新启动失败的传输；不对接收到的地址进行确认。	1	0	0
						放弃失败的传输。	0	0	X
	0001	1	1	X	试图发送停止条件时竞争失败。	重新启动失败的传输	1	0	X
		0	0	X	检测到停止条件。	不需任何操作（传输完成/放弃）。	0	0	0
						不需任何操作（传输完成）。	0	0	X
						放弃传输。	0	0	X
	0000	1	0	X	接收到字节；请求确认。	重新启动失败的传输。	1	0	X
						确认接收字节；读 SMB0DAT。	0	0	1
						不对接收到的字节进行确认。	0	0	0
		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
						重新启动失败的传输。	0	0	0

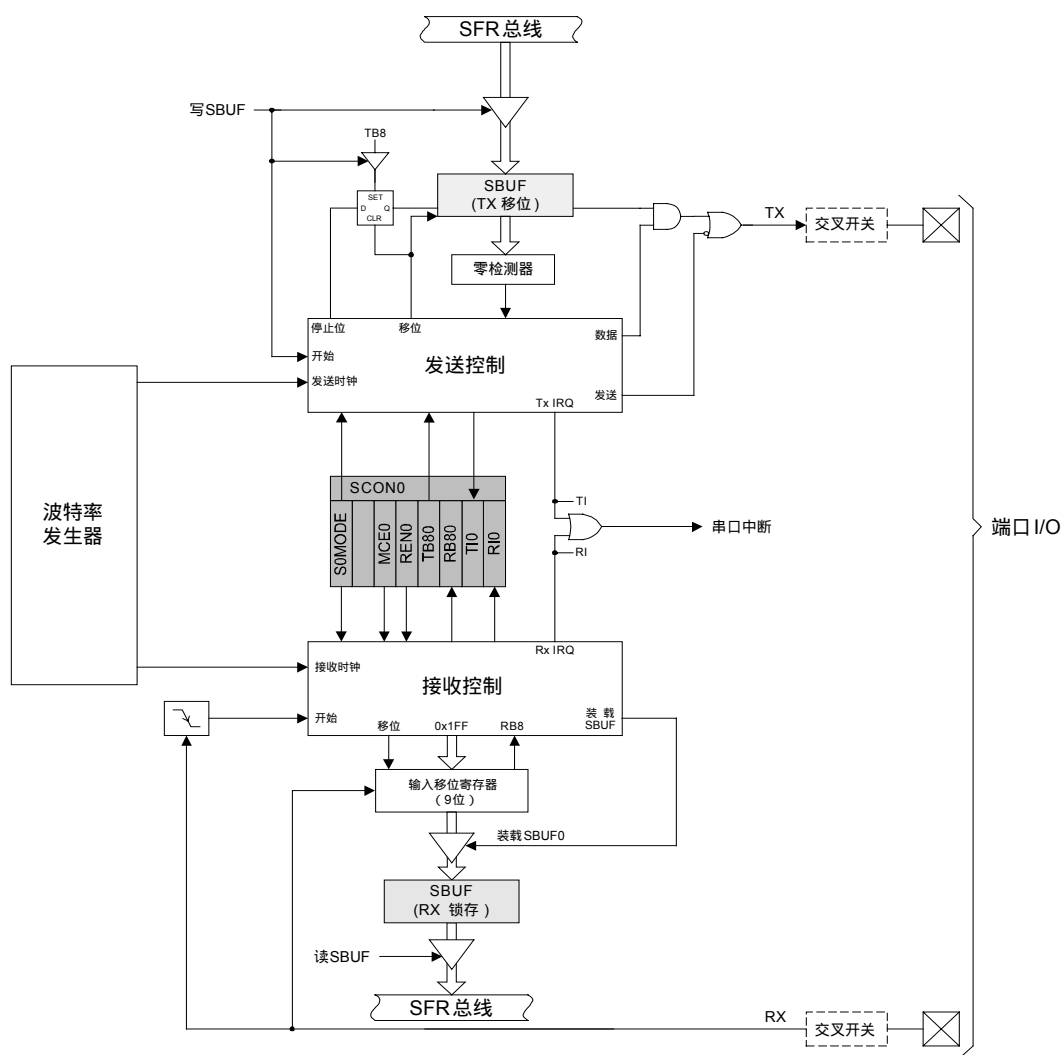
17. UART0

UART0 是一个异步、全双工串口，它提供标准 8051 串行口的方式 1 和方式 3。UART0 具有增强的波特率发生器电路，有多个时钟源可用于产生标准波特率。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 时自动访问发送寄存器；读 SBUF0 时自动访问接收寄存器，不可能从发送数据寄存器中读数据。**

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，这就允许软件查询 UART0 中断的原因（发送完成或接收完成）。

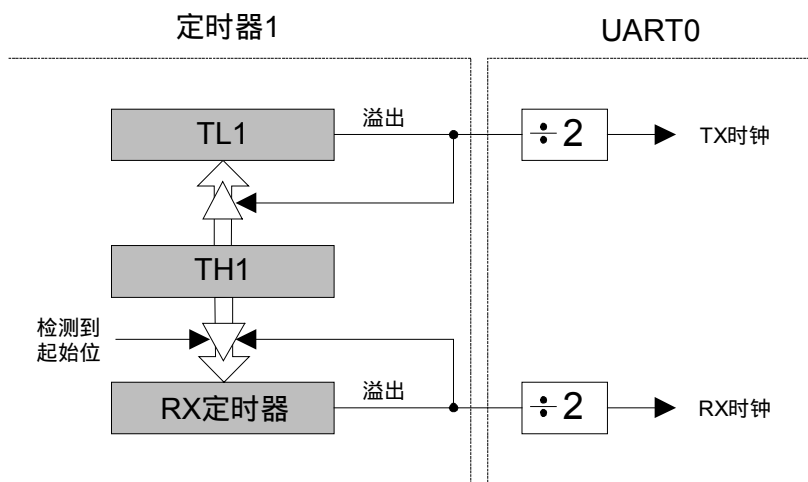
图 17.1 UART0 原理框图



17.1 增强的波特率发生器

UART0 波特率由定时器 1 工作在 8 位自动重载方式产生。发送 (TX) 时钟由 TL1 产生；接收 (RX) 时钟由 TL1 的拷贝寄存器 (图 17.2 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 17.2 UART0 波特率逻辑



定时器 1 应被配置为方式 2，即 8 位自动重载方式。定时器 1 的重载值应设置为使其溢出频率为所期望的波特率频率的两倍。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 17.1 决定：

方程 17.1 UART0 波特率

$$UART\text{波特率} = \frac{T1_{CLK}}{(256 - TH1)} \times \frac{1}{2}$$

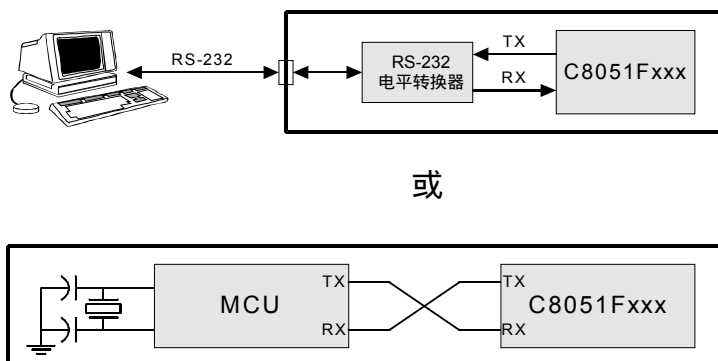
其中 $T1_{CLK}$ 是定时器 1 的时钟频率， $TH1$ 是定时器 1 的高字节（重载值）。

定时器 1 时钟频率的选择方法见“19. 定时器”。表 17.1 – 17.6 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。

17.2 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 17.3 所示。

图 17.3 UART 连接图



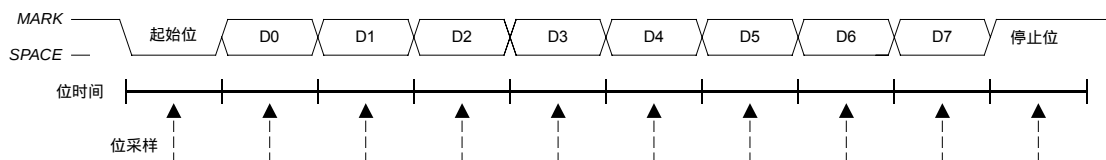
17.2.1 8 位 UART

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，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 17.4 8 位 UART 时序图

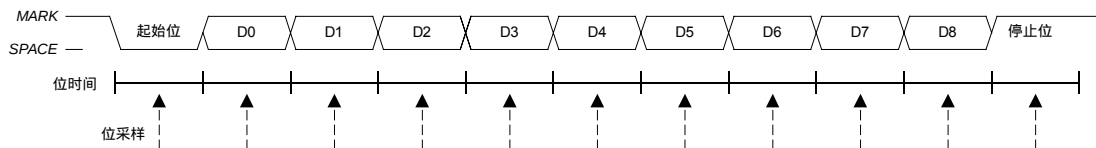


17.2.2 9 位 UART

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 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 置位时将产生一个中断。

图 17.5 9 位 UART 时序图



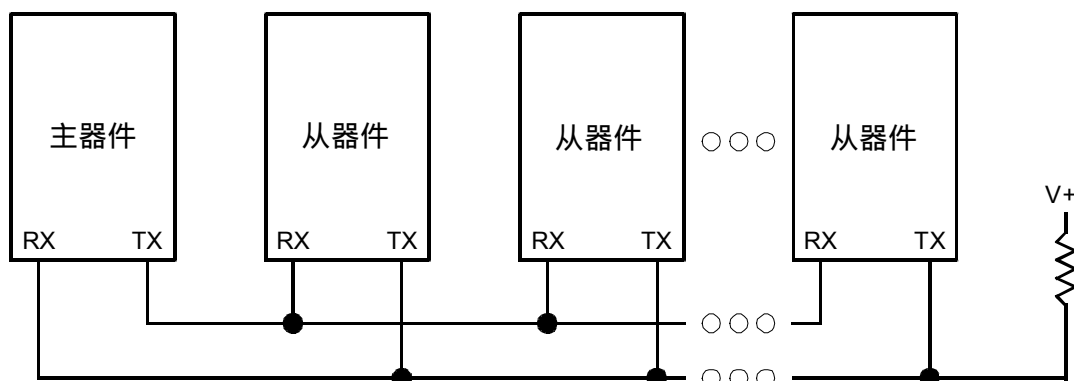
17.3 多机通信

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。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 17.6 UART 多机方式连接图



C8051F320/1 全速 USB ISP FLASH 微控制器

图 17.7 SCON0：UART0 控制寄存器

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	SFR地址： (可位寻址) 0x98
位 7： S0MODE：串行口工作方式选择位 该位选择 UART0 的工作方式。 0：方式 0：波特率可编程的 8 位 UART。 1：方式 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 该位的功能取决于串行口工作方式。 S0MODE = 0：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。 S0MODE = 1：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 方式中未用。跟据需要用软件置‘1’或清‘0’。 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’（在停止位后）。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’。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 17.8 SBUF0：UART0 串行数据缓冲寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0x99

位 7-0：SBUF0.[7:0]：UART0 数据缓冲器位 7-0（MSB-LSB）
实际上是两个寄存器：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 17.1 对应标准波特率的定时器设置（使用内部振荡器）

频率：24.5MHz							
	(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)	T1M	定时器 1 重载值 (hex)
SYSCLK 源自 内部振荡器	230400	-0.32%	106	SYSCLK	XX	1	0xC8
	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

X = 忽略

表 17.2 对应标准波特率的定时器设置（使用外部振荡器）

频率：25.0MHz							
	目标波特率 (bps)	波特率 误差 (%)	分频系数	定时器 时钟源	SCA1-SCA0	T1M	定时器 1 重载值 (hex)
SYSCLK 源自 外部振荡器	230400	-0.47%	108	SYSCLK	XX	1	0xCA
	115200	0.45%	218	SYSCLK	XX	1	0x93
	57600	-0.01%	434	SYSCLK	XX	1	0x27
	28800	0.45%	872	SYSCLK/4	01	0	0x93
	14400	-0.01%	1736	SYSCLK/4	01	0	0x27
	9600	0.15%	2608	EXTCLK/8	11	0	0x5D
	2400	0.45%	10464	SYSCLK/48	10	0	0x93
	1200	-0.01%	20832	SYSCLK/48	10	0	0x27
SYSCLK 源自	57600	-0.47%	432	EXTCLK/8	11	0	0xF5
	28800	-0.47%	864	EXTCLK/8	11	0	0xCA
	14400	0.45%	1744	EXTCLK/8	11	0	0x93
	9600	0.15%	2608	EXTCLK/8	11	0	0x5D

X = 忽略

C8051F320/1 全速 USB ISP FLASH 微控制器

表 17.3 对应标准波特率的定时器设置（使用外部振荡器）

频率：22.1184MHz							
	目标波特率	波特率 误差（%）	分频系数	定时器 时钟源	SCA1-SCA0 （分频选择）	T1M	定时器 1 重载值（hex）
SYSCLK 源自 外部振荡器	230400	0.00%	96	SYSCLK	XX	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
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	0xF8
	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

X = 忽略

表 17.4 对应标准波特率的定时器设置（使用外部振荡器）

频率：18.432MHz							
	目标波特率 （bps）	波特率 误差（%）	分频系数	定时器 时钟源	SCA1-SCA0 （分频选择）	T1M	定时器 1 重载值（hex）
SYSCLK 源自 外部振荡器	230400	0.00%	80	SYSCLK	XX	1	0xD8
	115200	0.00%	160	SYSCLK	XX	1	0xB0
	57600	0.00%	320	SYSCLK	XX	1	0x60
	28800	0.00%	640	SYSCLK/4	01	0	0xB0
	14400	0.00%	1280	SYSCLK/4	01	0	0x60
	9600	0.00%	1920	SYSCLK/12	00	0	0xB0
	2400	0.00%	7680	SYSCLK/48	10	0	0xB0
	1200	0.00%	15360	SYSCLK/48	10	0	0x60
SYSCLK 源自 内部振荡器	230400	0.00%	80	EXTCLK/8	11	0	0xER
	115200	0.00%	160	EXTCLK/8	11	0	0xF6
	57600	0.00%	320	EXTCLK/8	11	0	0xFC
	28800	0.00%	640	EXTCLK/8	11	0	0xD8
	14400	0.00%	1280	EXTCLK/8	11	0	0xB0
	9600	0.00%	1920	EXTCLK/8	11	0	0x88

X = 忽略

C8051F320/1 全速 USB ISP FLASH 微控制器

表 17.5 对应标准波特率的定时器设置（使用外部振荡器）

频率：11.0592MHz							
	目标波特率 (bps)	波特率 误差 (%)	分频系数	定时器 时钟源	SCA1-SCA0 (分频选择)	T1M	定时器 1 重载值 (hex)
SYSCLK 源自 外部振荡器	230400	0.00%	48	SYSCLK	XX	1	0xE8
	115200	0.00%	96	SYSCLK	XX	1	0xD0
	57600	0.00%	192	SYSCLK	XX	1	0xA0
	28800	0.00%	384	SYSCLK	XX	1	0x40
	14400	0.00%	768	SYSCLK/12	00	0	0xE0
	9600	0.00%	1152	SYSCLK/12	00	0	0xD0
	2400	0.00%	4608	SYSCLK/12	00	0	0x40
	1200	0.00%	9216	SYSCLK/48	10	0	0xA0
SYSCLK 源自 内部振荡器	230400	0.00%	48	EXTCLK/8	11	0	0xF0
	115200	0.00%	96	EXTCLK/8	11	0	0xFA
	57600	0.00%	192	EXTCLK/8	11	0	0xF4
	28800	0.00%	384	EXTCLK/8	11	0	0xF8
	14400	0.00%	768	EXTCLK/8	11	0	0xD0
	9600	0.00%	1152	EXTCLK/8	11	0	0xB8

X = 忽略

表 17.6 对应标准波特率的定时器设置（使用外部振荡器）

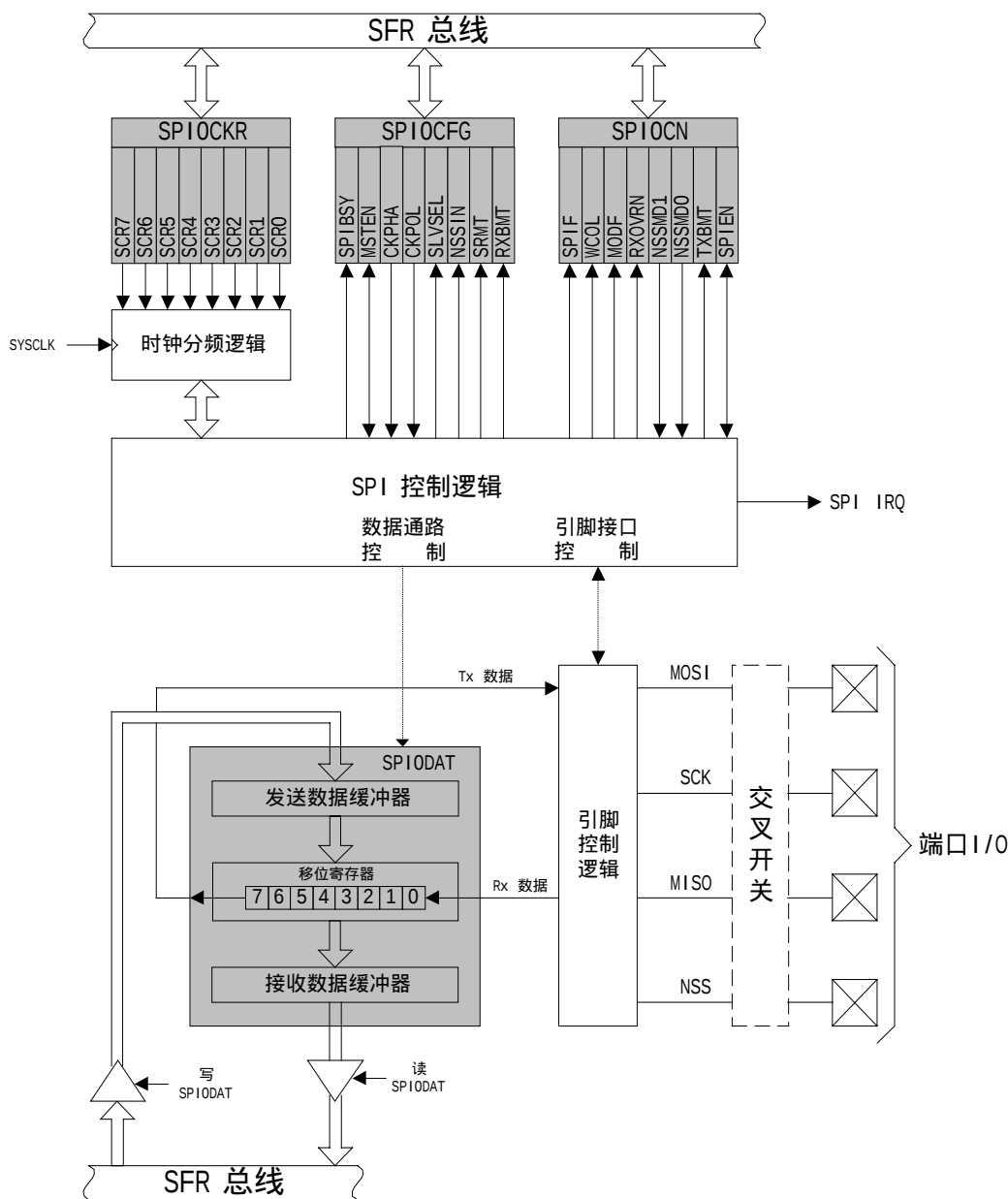
频率：3.6864MHz							
	目标波特率 (bps)	波特率 误差 (%)	振荡器 分频系数	定时器 时钟源	SCA1-SCA0 (分频选择)	T1M	定时器 1 重载值 (hex)
SYSCLK 源自 外部振荡器	230400	0.00%	16	SYSCLK	XX	1	0xF8
	115200	0.00%	32	SYSCLK	XX	1	0xF0
	57600	0.00%	64	SYSCLK	XX	1	0xE0
	28800	0.00%	128	SYSCLK	XX	1	0xC0
	14400	0.00%	256	SYSCLK	XX	1	0x80
	9600	0.00%	384	SYSCLK	XX	1	0x40
	2400	0.00%	1536	SYSCLK/12	00	0	0xC0
	1200	0.00%	3072	SYSCLK/12	00	0	0x80
SYSCLK 源自 内部振荡器	230400	0.00%	16	EXTCLK/8	11	0	0xFF
	115200	0.00%	32	EXTCLK/8	11	0	0xFE
	57600	0.00%	64	EXTCLK/8	11	0	0xFC
	28800	0.00%	128	EXTCLK/8	11	0	0xF8
	14400	0.00%	256	EXTCLK/8	11	0	0xF0
	9600	0.00%	384	EXTCLK/8	11	0	0xE8

X = 忽略

18. 增强型串行外设接口 (SPI0)

增强型串行外设接口 (SPI0) 提供访问一个全双工同步串行总线的能力。SPI0 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 SPI0，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 18.1 SPI 原理框图



18.1 信号说明

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

18.1.1 主输出、从输入 (MOSI)

主出从入 (MOSI) 信号是主器件的输出和从器件的输入，用于从主器件到从器件的串行数据传输。当 SPI0 作为主器件时，该信号是输出；当 SPI0 作为从器件时，该信号是输入。数据传输时最高位在先。当被配置为主器件时，MOSI 由移位寄存器的 MSB 驱动。

18.1.2 主输入、从输出 (MISO)

主入从出 (MISO) 信号是从器件的输出和主器件的输入，用于从从器件到主器件的串行数据传输。当 SPI0 作为主器件时，该信号是输入；当 SPI0 作为从器件时，该信号是输出。数据传输时最高位在先。当 SPI 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 由移位寄存器的 MSB 驱动。

18.1.3 串行时钟 (SCK)

串行时钟 (SCK) 信号是主器件的输出和从器件的输入，用于同步主器件和从器件之间在 MOSI 和 MISO 线上的串行数据传输。当 SPI0 作为主器件时产生该信号。在 4 线从方式，当从器件未被选中时 (NSS=1)，SCK 信号被忽略。

18.1.4 从选择 (NSS)

从选择 (NSS) 信号的功能取决于 SPI0CN 寄存器中 NSSMD1 和 NSSMD0 位的设置。有 3 种可能的方式：

1. NSSMD[1:0] = 00：3 线主方式或从方式：SPI0 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，SPI0 总是被选择。由于没有选择信号，SPI0 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. NSSMD[1:0] = 01：4 线从方式或多主方式：SPI0 工作在 4 线方式，NSS 作为输入。当作为从器件时，NSS 选择从 SPI0 器件。当作为主器件时，NSS 信号的负跳变禁止 SPI0 的主器件功能，因此可以在同一个 SPI 总线上使用多个主器件。
3. NSSMD[1:0] = 1x：4 线主方式：SPI0 工作在 4 线方式，NSS 作为输出。NSSMD0 的设置值决定 NSS 引脚的输出电平。这种配置只能在 SPI0 作为主器件时使用。

图 18.2 ~ 18.4 给出了不同方式下的典型连接图。注意：NSSMD 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。

图 18.2 多主方式连接图

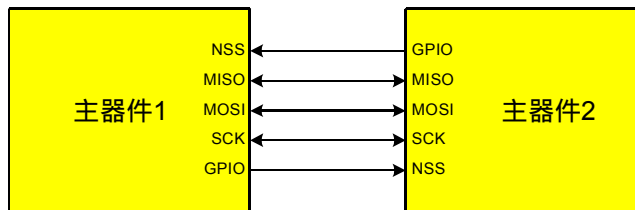


图 18.3 3 线单主方式和 3 线单从方式连接图

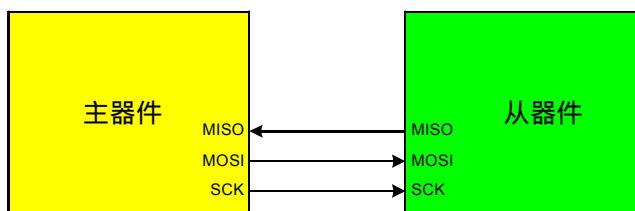
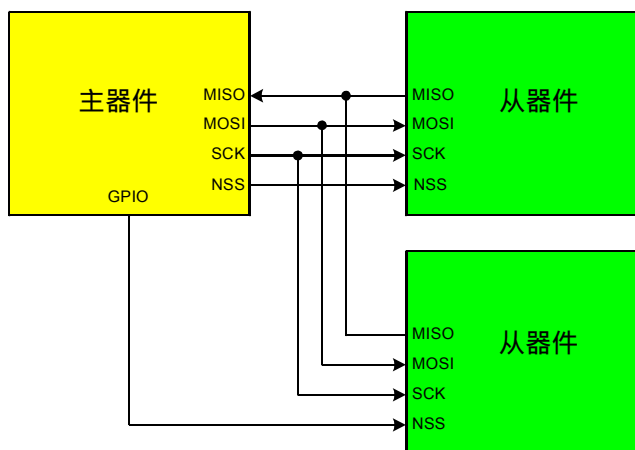


图 18.4 4 线单主方式和 4 线从方式连接图



18.2 SPI0 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, SPI0CFG.6)置 1 将 SPI0 置于主方式。当处于主方式时,向 SPI0 数据寄存器(SPI0DAT)写入一个字节时是写发送缓冲器。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。SPI0 主器件立即在 MOSI 线上串行移出数据,同时在 SCK 上提供串行时钟。在传输结束后 SPIF(SPI0CN.7)标志被置为逻辑 1。如果中断被允许,在 SPIF 标志置位时将产生一个中断请求。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 SPI0DAT 来读该缓冲器。

当被配置为主器件时, SPI0 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 NSSMD1(SPI0CN.3)=0 且 NSSMD0(SPI0CN.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 SPI0,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, MSTEN(SPI0CN.6)和 SPIEN(SPI0CN.0)位被硬件清 0,以禁止 SPI 主器件,且方式错误标志(MODF, SPI0CN.5)被置 1。如果中断被允许,将产生中断。在这种情况下,必须用软件重新使能 SPI0。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 18.2 给出了两个主器件在多主方式下的连接图。

当 NSSMD1(SPI0CN.3)=0 且 NSSMD0(SPI0CN.2)=0 时, SPI0 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 引脚选择要寻址的从器件。图 18.3 给出了一个 3 线主方式主器件和一个从器件的连接图。

当 NSSMD1(SPI0CN.3)=1 时, SPI0 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 NSSMD0(SPI0CN.2)控制(用软件)。可以用通用 I/O 引脚选择另外的从器件。图 18.4 给出了一个 4 线主方式主器件和两个从器件的连接图。

18.3 SPI0 从方式

当 SPI0 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。SPI0 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，接收到的字节被传送到接收缓冲器。通过读 SPI0DAT 来读取接收缓冲器中的数据。从器件不能启动数据传输。通过写 SPI0DAT 来预装要发送给主器件的数据。写往 SPI0DAT 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 将等到数据发送完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，SPI0 可以工作 4 线或 3 线方式。当 NSSMD1 (SPI0CN.3) = 0 且 NSSMD0 (SPI0CN.2) = 1 时，是默认的 4 线方式。在 4 线方式，NSS 被分配端口引脚并被配置为数字输入。当 NSS 为逻辑 0 时，SPI0 被使能；当 NSS 为逻辑 1 时，SPI0 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 18.4 给出了两个 4 线方式从器件和一个主器件的连接图。

当 NSSMD1 (SPI0CN.3) = 0 且 NSSMD0 (SPI0CN.2) = 0 时，SPI0 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 SPI0 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 SPI0 来复位位计数器。图 18.3 给出了一个 3 线从器件和一个主器件的连接图。

18.4 SPI0 中断源

如果 SPI0 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，SPI 中断标志 SPIF (SPI0CN.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移位寄存器时写 SPI0DAT，写冲突标志 WCOL (SPI0CN.6) 被置 1。发生这种情况时，写 SPI0DAT 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 SPI0 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 MODF (SPI0CN.5) 被置 1。当发生方式错误时，SPI0CN 中的 MSTEN 和 SPIEN 位被清 0，以禁止 SPI0 并允许另一个主器件访问总线。
4. 当 SPI0 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 RXOVRN (SPI0CN.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。

18.5 串行时钟时序

使用 SPI0 配置寄存器 (SPI0CFG) 中的时钟控制选择位可以在串行时钟相位和极性的 4 种组合中选择其一。CKPHA 位 (SPI0CFG.5) 选择两种时钟相位 (锁存数据所用的边沿) 中的一种。CKPOL 位 (SPI0CFG.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。注意 :在改变时钟相位和极性期间应禁止 SPI0 (通过清除 SPIEN 位, SPI0CN.0)。主方式下时钟和数据线的时序关系示于图 18.5 ;从方式下时钟和数据线的时序关系示于图 18.6 和图 18.7。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。

图 18.10 所示的 SPI0 时钟速率寄存器 (SPI0CKR) 控制主方式的串行时钟频率。当工作于从方式时该寄存器被忽略。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 (取较低的频率)。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系统时钟同步的情况下。

图 18.5 主方式数据/时钟时序图

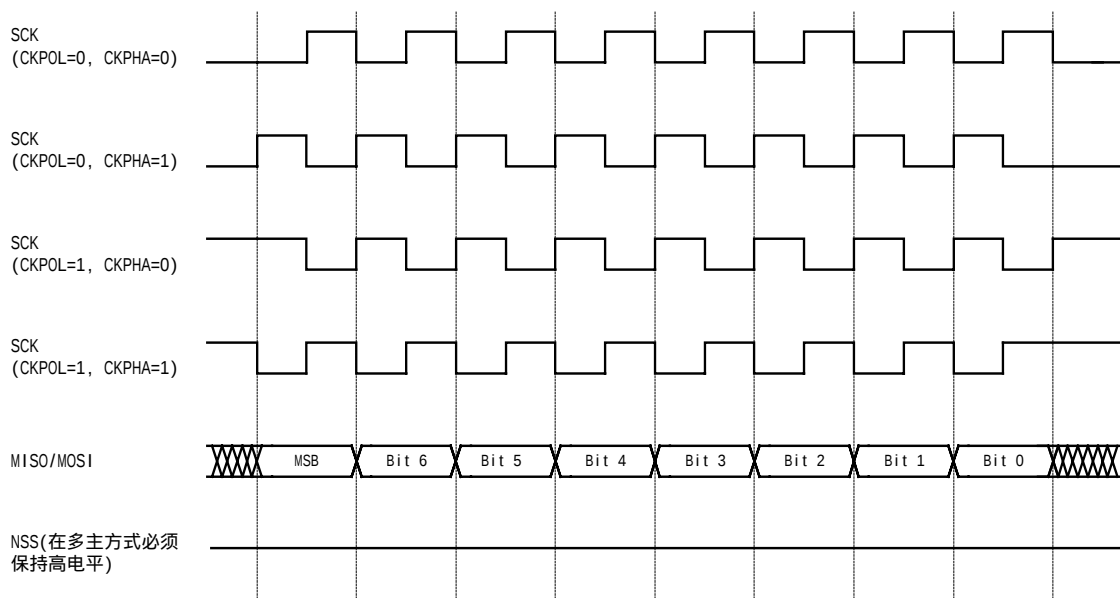


图 18.6 从方式数据/时钟时序图 (CKPHA = 0)

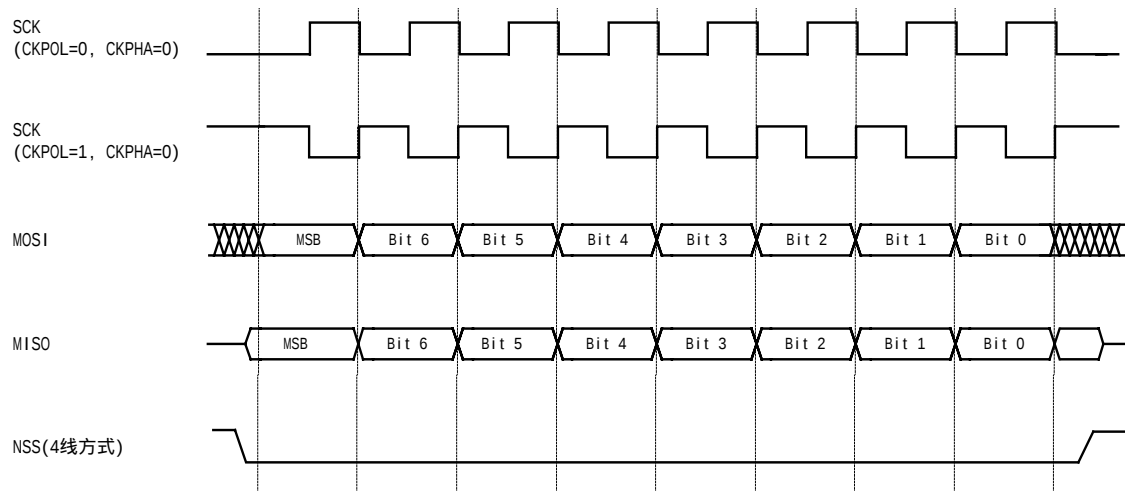
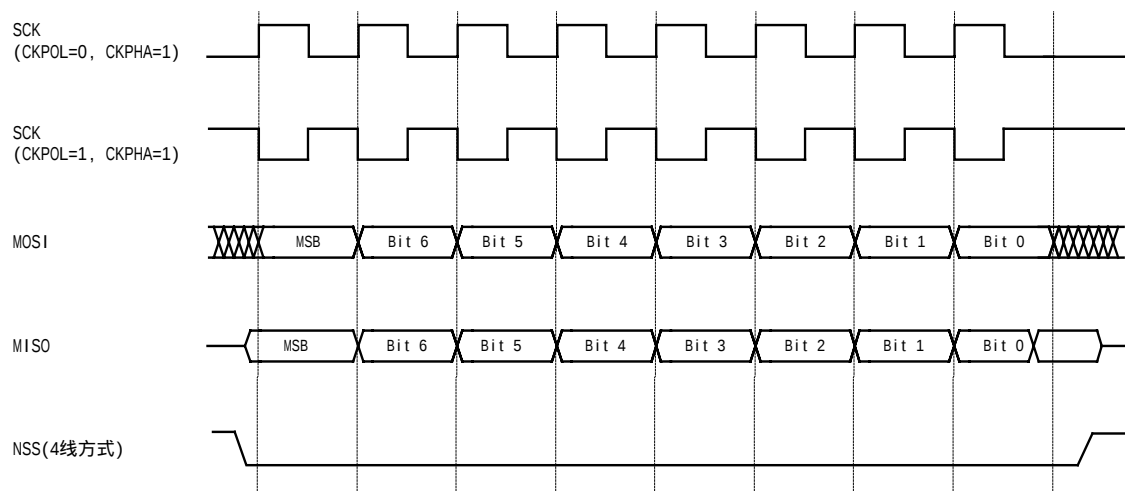


图 18.7 从方式数据/时钟时序图 (CKPHA = 1)



18.6 SPI 特殊功能寄存器

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

图 18.8 SPI0CFG：SPI0 配置寄存器

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	SFR地址： 0xA1
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：主方式允许位 0：禁止主方式，工作在从方式。 1：允许主方式，工作在主器件方式。 位 5： CKPHA：SPI0 时钟相位。 该位控制 SPI0 时钟的相位。 0：在SCK周期的第一个边沿采样数据[†]。 1：在SCK周期的第二个边沿采样数据[†]。 位 4： CKPOL：SPI0 时钟极性 该位控制 SPI0 时钟的极性。 0：SCK 在空闲状态时处于低电平。 1：SCK 在空闲状态时处于高电平。 位 3： SLVSEL：从选择标志（只读）。 当 NSS 引脚为低电平时该位被置 1，表示 SPI0 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信号。 位 2： NSSIN：NSS 引脚的瞬时值（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（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：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 18.1。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 18.7 SPI0CN : SPI0 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	复位值
SPIF	WCOL	MODF	RXOVRN	NSSMD1	NSSMD0	TXBMT	SPIEN	00000110
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址 : 0xF8 (可位寻址)
位 7 : SPIF : SPI0 中断标志 该位在数据传输结束后被硬件置为逻辑 1。如果中断被允许, 置 1 该位将会使 CPU 转到 SPI0 中断处理服务程序。该位不能被硬件自动清 0, 必须用软件清 0。 位 6 : WCOL : 写冲突标志 该位由硬件置为逻辑 1 (并产生一个 SPI0 中断), 表示数据传送期间对 SPI0 数据寄存器进行了写操作。该位必须用软件清 0。 位 5 : MODF : 方式错误标志 当检测到主方式冲突 (NSS 为低电平, MSTEN = 1, NSSMD[1:0] = 01) 时, 该位由硬件置为逻辑 1 (并产生一个 SPI0 中断)。该位不能被硬件自动清 0, 必须用软件清 0。 位 4 : RXOVRN : 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 SPI0 移位寄存器, 而接收缓冲器中仍保存着前一次传输未被读取的数据时该位由硬件置为逻辑 1 (并产生一个 SPI0 中断)。该位不会被硬件自动清 0, 必须用软件清 0。 位 3-2 : NSSMD1-NSSMD0 : 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 总是器件的输入。 1x : 4 线单主方式。NSS 被分配一个输出引脚并输出 NSSMD0 的值。 位 1 : TXBMT : 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, 表示可以向发送缓冲器写新数据。 位 0 : SPIEN : SPI0 使能位 该位使能 / 禁止 SPI0。 0 : 禁止 SPI0 1 : 使能 SPI0								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 18.10 SPI0CKR : SPI0 时钟速率寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xA2

位 7-0 : SCR7-SCR0 : SPI0 时钟频率

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

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

(0 ≤ SPI0CKR ≤ 255)

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

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$
$$f_{SCK} = 200kHz$$

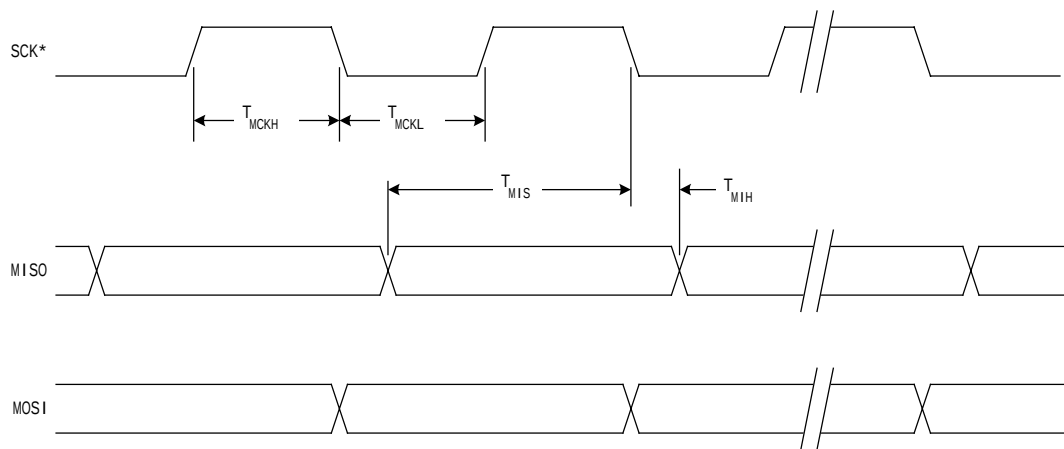
图 18.11 SPI0DAT : SPI0 数据寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xA3

位 7-0 : SPI0DAT : SPI0 发送和接收数据寄存器。

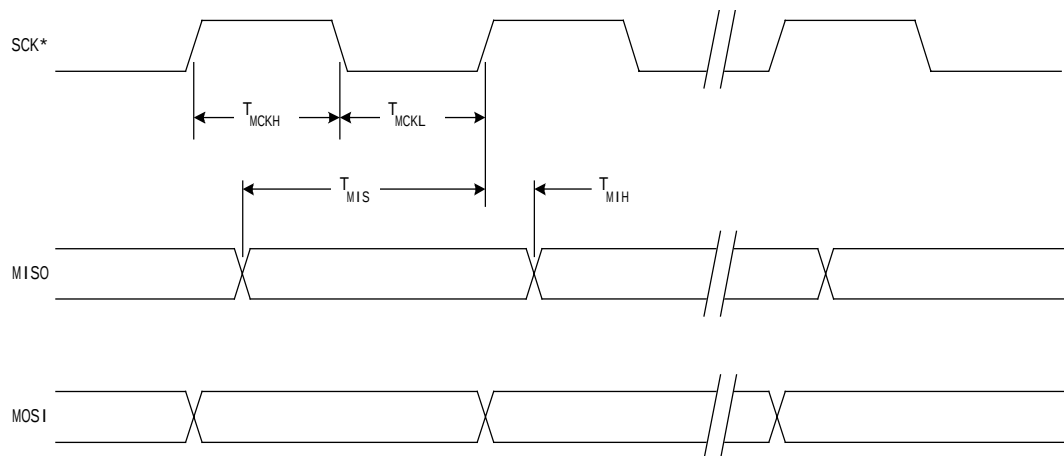
SPI0DAT 寄存器用于发送和接收 SPI0 数据。在主方式下, 向 SPI0DAT 写入数据时, 数据被放到发送缓冲器并启动发送。读 SPI0DAT 返回接收缓冲器的内容。

图 18.12 SPI 主方式时序 (CKPHA = 0)



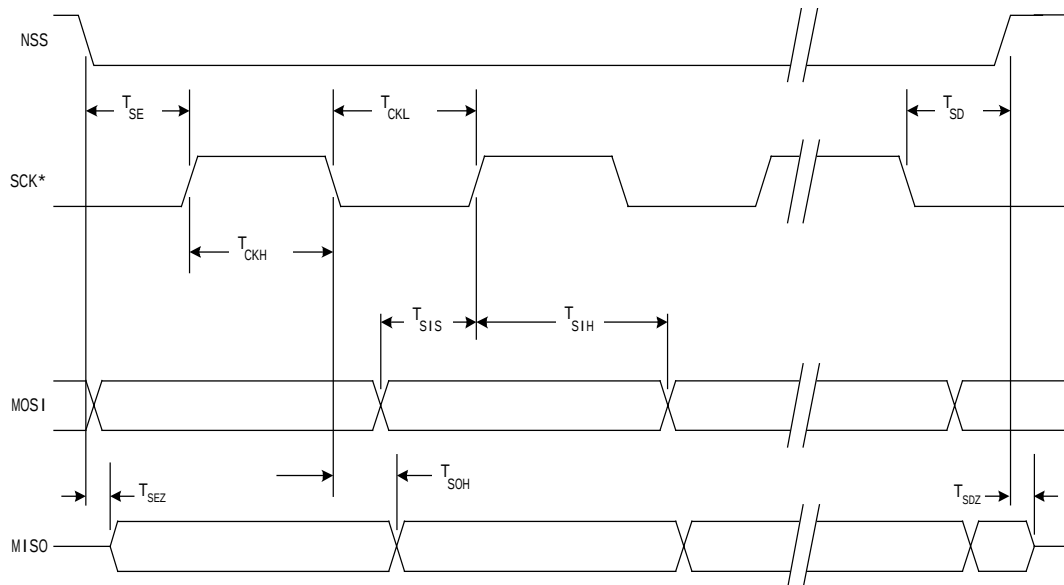
* 这是对应 CKPOL = 0时的 SCK 波形。对于 CKPOL = 1, SCK波形的极性反向。

图 18.13 SPI 主方式时序 (CKPHA = 1)



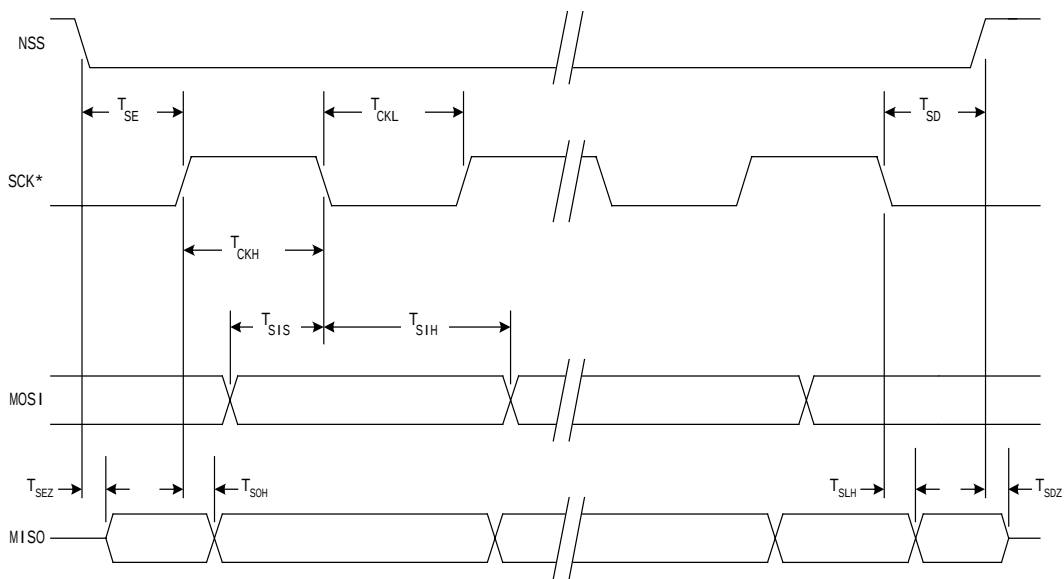
* 这是对应 CKPOL = 0时的 SCK 波形。对于 CKPOL = 1, SCK波形的极性反向。

图 18.14 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 18.15 SPI 从方式时序 (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。

C8051F320/1 全速 USB ISP FLASH 微控制器

表 18.1 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 18.12 和图 18.13)				
T_{MCKH}	SCK 高电平时间	$1 \cdot T_{SYSCLK}$		ns
T_{MCKL}	SCK 低电平时间	$1 \cdot T_{SYSCLK}$		ns
T_{MIS}	MISO 有效到 SCK 移位边沿	$1 \cdot T_{SYSCLK} + 20$		ns
T_{MIH}	SCK 移位边沿到 MISO 发生改变	0		ns
从方式时序 (见图 18.14 和图 18.15)				
T_{SE}	NSS 下降沿到第一个 SCK 边沿	$2 \cdot T_{SYSCLK}$		ns
T_{SD}	最后一个 SCK 边沿到 NSS 上升沿	$2 \cdot T_{SYSCLK}$		ns
T_{SEZ}	NSS 下降沿到 MISO 有效		$4 \cdot T_{SYSCLK}$	ns
T_{SDZ}	NSS 上升沿到 MISO 变为高阻态		$4 \cdot T_{SYSCLK}$	ns
T_{CKH}	SCK 高电平时间	$5 \cdot T_{SYSCLK}$		ns
T_{CKL}	SCK 低电平时间	$5 \cdot T_{SYSCLK}$		ns
T_{SIS}	MOSI 有效到 SCK 采样边沿	$2 \cdot T_{SYSCLK}$		ns
T_{SIH}	SCK 采样边沿到 MOSI 发生改变	$2 \cdot T_{SYSCLK}$		ns
T_{SOH}	SCK 移位边沿到 MISO 发生改变		$4 \cdot T_{SYSCLK}$	ns
T_{SLH}	最后一个 SCK 边沿到 MISO 发生改变 (只限于 CKPHA = 1)	$6 \cdot T_{SYSCLK}$	$8 \cdot T_{SYSCLK}$	ns
注： T_{SYSCLK} 为系统时钟 (SYSCLK) 周期。				

19. 定时器

C8051F32x 内部有 4 个 16 位计数器/定时器：其中两个与标准 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、USB（帧测量）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。

定时器 0 和定时器 1 工作方式	定时器 2 工作方式	定时器 3 工作方式
13 位计数/定时器	16 位自动重装载定时器	16 位自动重装载定时器
16 位计数器/定时器		
8 位自动重装载的计数器/定时器	两个 8 位自动重装载定时器	两个 8 位自动重装载定时器
两个 8 位计数器/定时器 (仅限于定时器 0)		

定时器 0 和定时器 1 有 5 个可选择的时钟源，由定时器时钟选择位（T1M-T0M）和时钟分频位（SCA1-SCA0）决定。时钟分频位定义一个分频时钟，作为定时器 0 和/或定时器 1 的时钟源（见图 19.6）。

定时器 0 和定时器 1 可以被配置为使用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、系统时钟/12 或外部振荡器时钟/8 作为时钟源。

定时器 0 和定时器 1 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。

19.1 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，在被访问时以两个字节的形式出现：一个低字节（TL0 或 TL1）和一个高字节（TH0 或 TH1）。计数器/定时器控制寄存器（TCON）用于允许定时器 0 和定时器 1 以及指示它们的状态。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（TMOD）寄存器中的方式选择位 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。

19.1.1 方式 0

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 19.1 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，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

C8051F320/1 全速 USB ISP FLASH 微控制器

时器溢出标志 TF0 (TCON.5) 被置位并产生一个中断 (如果该中断被允许)。

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。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 (见图 19.6) 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 INT01CF 寄存器中的 IN0PL 位定义，见图 19.13)，置 ‘ 1 ’ TR0 位 (TCON.4) 将允许定时器 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。

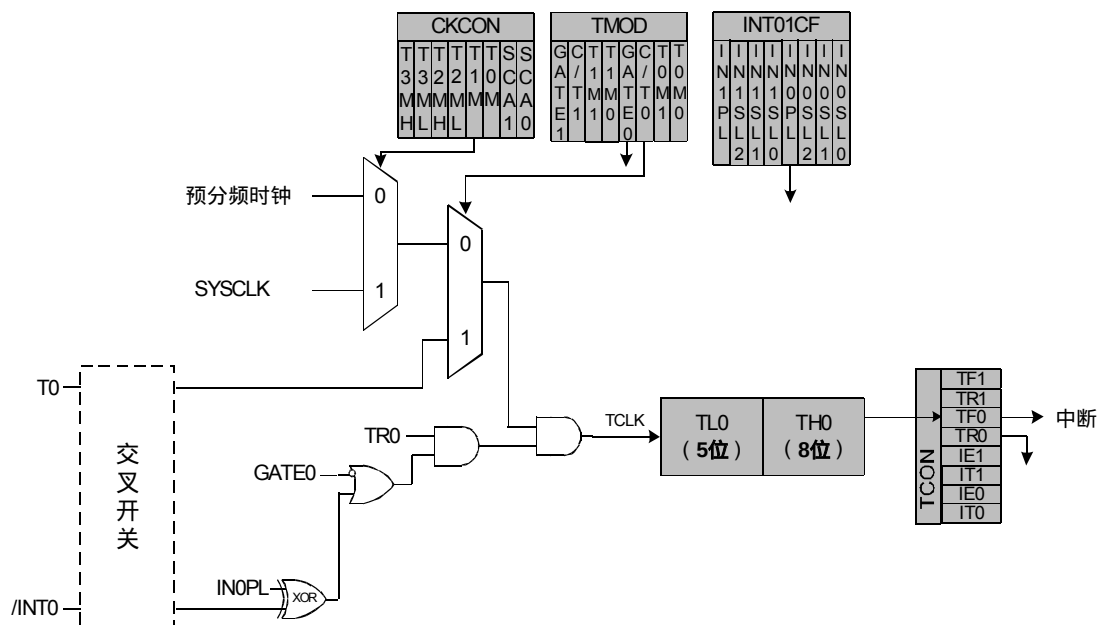
TR0	GATE0	/INT0	计数器/定时器
0	X	X	禁止
1	0	X	允许
1	1	0	禁止
1	1	1	允许

X=任意

注意，置位 TR0 并不强制定时器 0 复位。应在定时器被允许前将定时器寄存器装入所需要的初值。

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。

图 19.1 T0 方式 0 原理框图



19.1.2 方式 1

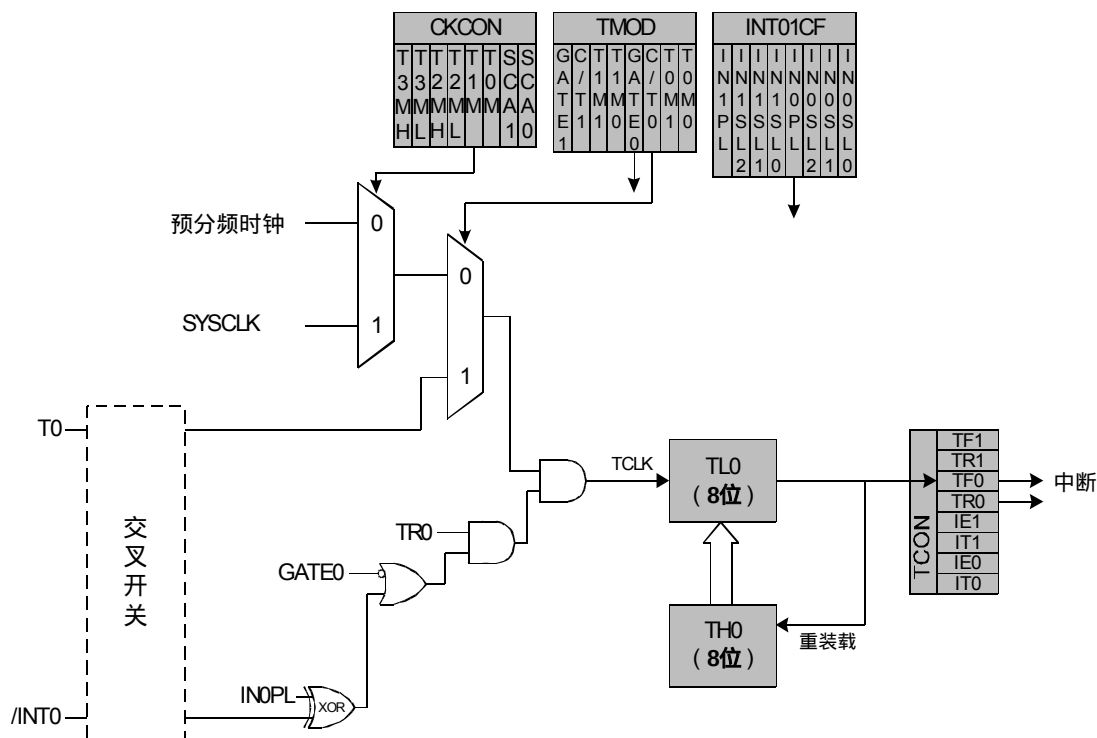
方式 1 的操作与方式 0 完全一样，所不同的是计数器/定时器使用全部 16 位。用与方式 0 相同的方法允许和控制工作在方式 1 的计数器/定时器。

19.1.3 方式 2

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

在方式 2，定时器 1 和定时器 0 的配置和控制方法与方式 0 一样。当 GATE0（TMOD.3）为逻辑 ‘0’ 或输入信号/INT0 有效时（有效电平由 INT01CF 寄存器中的 IN0PL 为定义），置 ‘1’ TR0 位（TCON.4）将允许定时器 0 工作。

图 19.2 T0 方式 2 原理框图

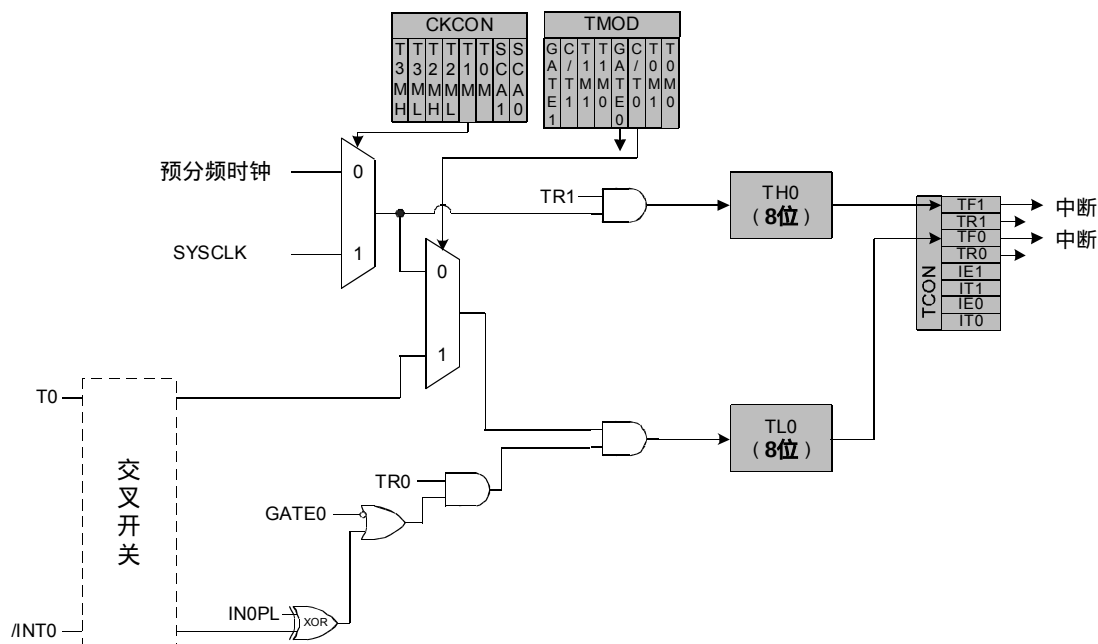


19.1.4 方式 3

在方式 3 时，定时器 0 被配置两个独立的 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,应使定时器 1 工作在方式 0、1 或 2。可以通过将定时器 1 切换到方式 3 使其停止运行。

图 19.3 T0 方式 3 原理框图



C8051F320/1 全速 USB ISP FLASH 微控制器

图 19.4 TCON : 定时器控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址 : (可位寻址) 0x88
位 7 :	TF1 : 定时器 1 溢出标志 当定时器 1 溢出时由硬件置位。该位可以用软件清 0 , 但当 CPU 转向定时器 1 中断服务程序时该位被自动清 0。 0 : 未检测到定时器 1 溢出。 1 : 定时器 1 发生溢出。							
位 6 :	TR1 : 定时器 1 运行控制 0 : 定时器 1 禁止。 1 : 定时器 1 允许。							
位 5 :	TF0 : 定时器 0 溢出标志 当定时器 0 溢出时由硬件置位。该位可以用软件清 0 , 但当 CPU 转向定时器 0 中断服务程序时该位被自动清 0。 0 : 未检测到定时器 1 溢出。 1 : 定时器 1 发生溢出。							
位 4 :	TR0 : 定时器 0 运行控制 0 : 定时器 0 禁止。 1 : 定时器 0 允许。							
位 3 :	IE1 : 外部中断 1 当检测到一个由 IT1 定义的边沿/电平时 , 该标志由硬件置位。该位可以用软件清 0 , 但当 CPU 转向外部中断 1 中断服务程序时该位被自动清 0 (如果 IT1=1) 。 当 IT1=0 时 , 该标志在 /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。 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 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。 0 : /INT0 为电平触发。 1 : /INT0 为边沿触发。							

C8051F320/1 全速 USB ISP FLASH 微控制器

图 19.5 TMOD : 定时器方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址 : 0x89

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。

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。

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

C8051F320/1 全速 USB ISP FLASH 微控制器

图 19.6 CKCON：时钟控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
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	SFR地址： 0x8E

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 5：

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-SAC0）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-SCA0：定时器 0/1 分频位

如果定时器 0/1 被配置为使用分频时钟，则这些位控制时钟分频数。

SCA1	SCA0	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

注：外部时钟 8 分频与系统时钟同步，在该方式，外部时钟频率必须小于或等于系统时钟频率。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 19.7 TL0：定时器 0 低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址： 0x8A
位 7-0： TL0：定时器 0 低字节								
TL0 寄存器是 16 位定时器 0 的低字节。								

图 19.8 TL1：定时器 1 低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址： 0x8B
位 7-0： TL1：定时器 1 低字节								
TL1 寄存器是 16 位定时器 1 的低字节。								

图 19.9 TH0：定时器 0 高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址： 0x8C
位 7-0： TH0：定时器 0 高字节								
TH0 寄存器是 16 位定时器 0 的高字节。								

图 19.10 TH1：定时器 1 高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址： 0x8D
位 7-0： TH1：定时器 1 高字节								
TH1 寄存器是 16 位定时器 1 的高字节。								

19.2 定时器 2

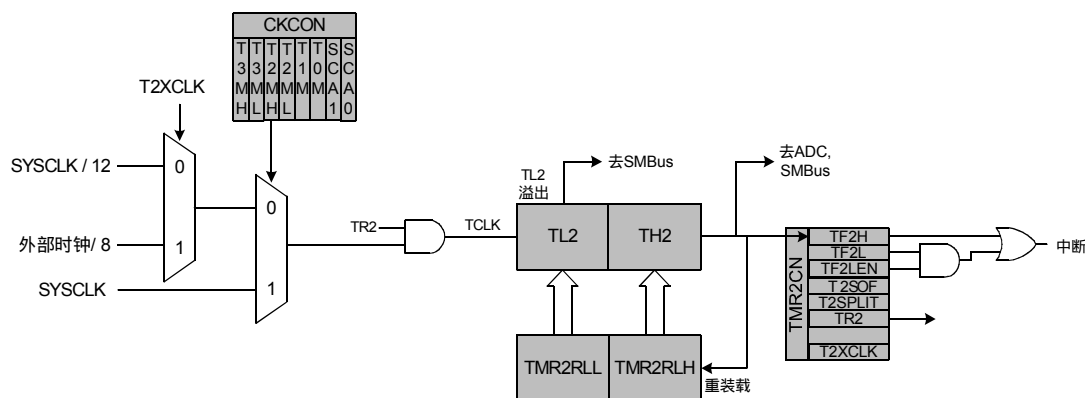
定时器 2 是一个 16 位的计数器/定时器，由两个 8 位的 SFR 组成：TMR2L（低字节）和 TMR2H（高字节）。定时器 2 可以工作在 16 位自动重载方式、8 位自动重载方式（两个 8 位定时器）或 USB 帧起始（SOF）捕捉方式。T2SPLIT 位（TMR2CN.3）和 T2SOF 位（TMR2CN.4）定义定时器 2 的工作方式。

定时器 2 的时钟源可以是系统时钟、系统时钟/12 或外部振荡源时钟/8。在使用实时时钟（RTC）功能时，外部时钟方式是理想的选择，此时用内部振荡器驱动系统时钟，而定时器 2（和/或 PCA）的时钟由一个精确的外部振荡器提供。注意，外部振荡源时钟/8 与系统时钟同步。

19.2.1 16 位自动重载方式

当 T2SPLIT = ‘0’ 且 T2SOF = ‘0’ 时，定时器 2 工作在自动重载的 16 位定时器方式（见图 19.11）。定时器 2 可以使用 SYSCLK、SYSCLK/12 或外部振荡源时钟/8 作为其时钟源。当 16 位定时器寄存器发生溢出（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，并将定时器 2 高字节溢出标志 TF2H（TMR2CN.7）置‘1’。如果定时器 2 中断被允许（如果 IE.5 被置 1），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 19.11 定时器 2 的 16 位方式原理框图



19.2.2 8 位自动重装载定时器方式

当 T2SPLIT = ' 1 ' 且 T2SOF = ' 0 ' 时, 定时器 2 工作双 8 位定时器方式。两个 8 位定时器 TMR2H 和 TMR2L 都工作在自动重装载方式 (见图 19.12)。TMR2RLH 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。

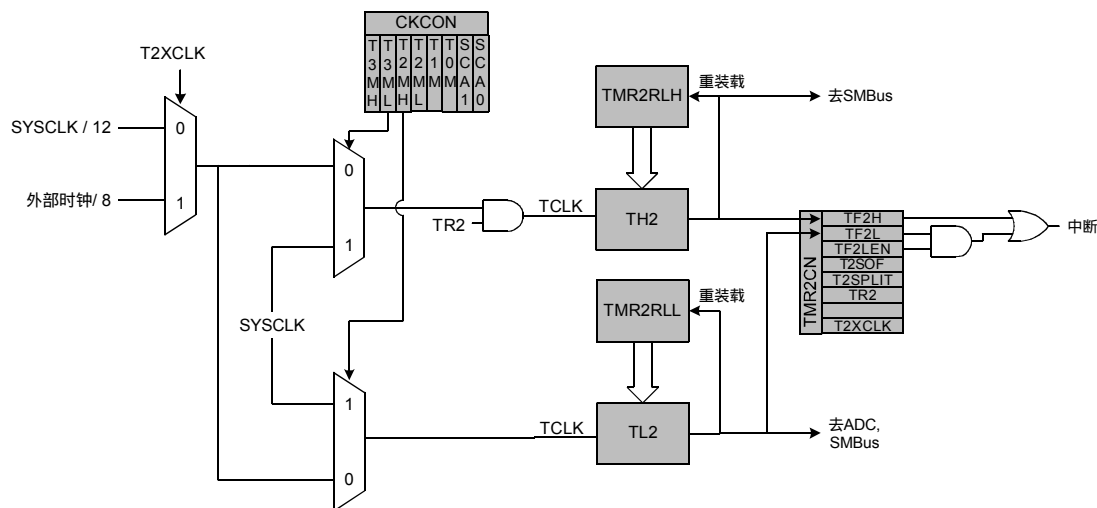
每个 8 位定时器都可以被配置为使用 SYSCLK、SYSCLK/12 或外部振荡器时钟/8 作为其时钟源。定时器 2 时钟选择位 T2MH 和 T2ML（位于 CKCON 中）选择 SYSCLK 或由定时器 2 外部时钟选择位（TMR2CN 中的 T2XCLK）定义的时钟源。时钟源的选择情况如下所示。

T2MH	T2XCLK	TMR2H 时钟源
0	0	SYSCLK/12
0	1	外部时钟/8
1	X	SYSCLK

T2ML	T2XCLK	TMR2L 时钟源
0	0	SYSCLK/12
0	1	外部时钟/8
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, 必须通过软件清除。

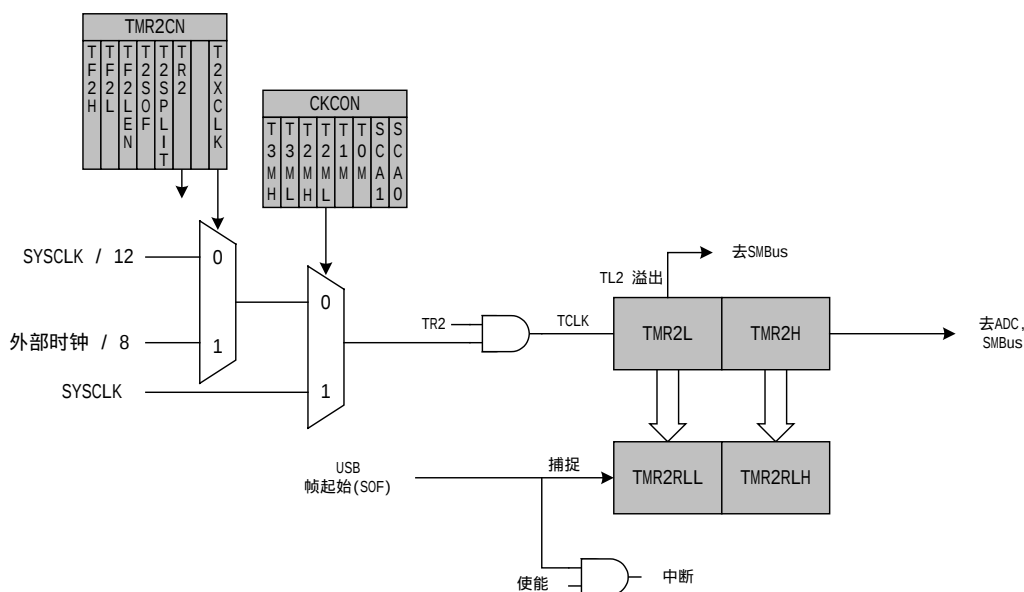
图 19.12 定时器 2 的 8 位方式原理框图



19.2.3 USB 帧起始捕捉方式

当 T2SOF = '1' 时，定时器 2 工作在 USB 帧起始（SOF）捕捉方式。当 T2SPLIT = '0' 时，定时器 2 向上计数，并在计数值从 0xFFFF 变为 0x0000 时溢出。每次收到 SOF 时，定时器 2 寄存器（TMR2H:TMR2L）的内容被锁存到定时器 2 重载寄存器（TMR2RLH:TMR2RLL）中，并产生定时器 2 中断（如果被使能）。该方式可被用于校准系统时钟或外部振荡器（用已知 USB 主机 SOF 时钟）。

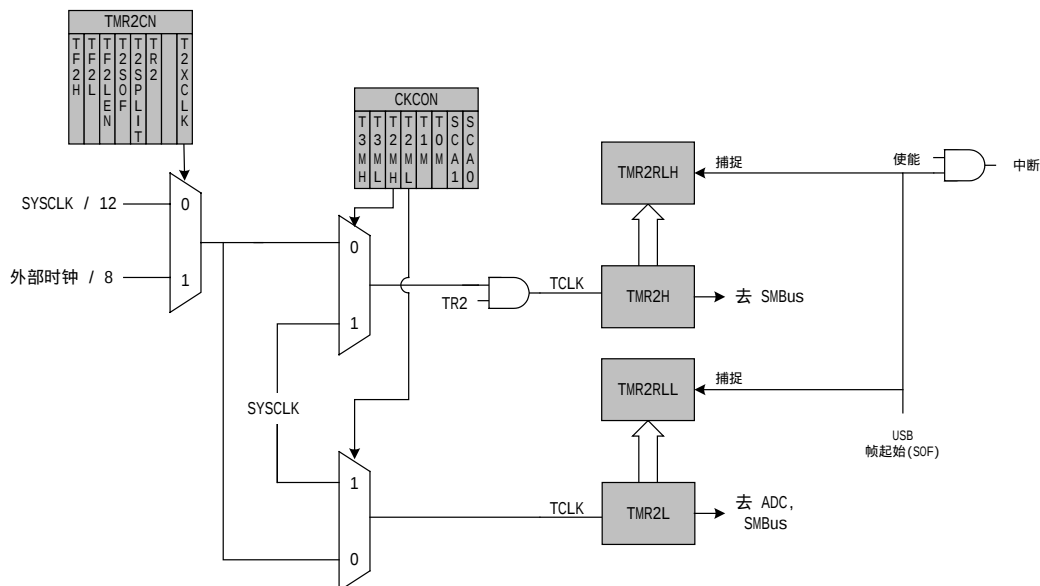
图 19.13 定时器 2 的 SOF 捕捉方式（T2SPLIT = '0'）



C8051F320/1 全速 USB ISP FLASH 微控制器

当 $T2SPLIT = '1'$ 时，定时器 2 寄存器（TMR2H:TMR2L）分成两个 8 位计数器。每个计数器都独立地向上计数并在计数值从 0xFF 变为 0x00 时溢出。每次收到 SOF 时，定时器 2 寄存器被锁存到定时器 2 重载寄存器（TMR2RLH:TMR2RLL）中，并产生定时器 2 中断（如果被使能）。

图 19.14 定时器 2 的 SOF 捕捉方式（ $T2SPLIT = '1'$ ）



C8051F320/1 全速 USB ISP FLASH 微控制器

图 19.15 TMR2CN：定时器 2 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
TF2H	TF2L	TF2LEN	T2SOF	T2SPLIT	TR2	-	T2XCLK	00000000
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址： 0xC8
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 该位允许/禁止定时器 2 低字节中断。如果 TF2LEN 被置‘1’并且定时器 2 中断被允许，则当定时器 2 低字节发生溢出时将产生中断。当定时器 2 工作在 16 位方式时，该位应被清‘0’。 0：禁止定时器 2 低字节中断。 1：允许定时器 2 低字节中断。 位 4： T2SOF：定时器 2 帧起始捕捉允许位 0：禁止 SOF 捕捉。 1：允许 SOF 捕捉。每次收到 SOF 时，定时器 2 寄存器（TMR2H:TMR2L）的内容被锁存到定时器 2 重载寄存器（TMR2RLH:TMR2RL）中，并产生定时器 2 中断（如果被使能）。 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 位自动重载定时器方式。 0：定时器 2 工作在 16 位自动重载方式。 1：定时器 2 工作在双 8 位自动重载定时器方式。 位 2： TR2：定时器 2 运行控制 该位允许/禁止定时器 2。在 8 位方式，该位只控制 TMR2H，TMR2L 总是处于使能状态。 0：定时器 2 禁止。 1：定时器 2 允许。 位 1： 未用。读 = 0b，写 = 忽略。 位 0： T2XCLK：定时器 2 外部时钟选择 该位选择定时器 2 的外部时钟源。如果定时器 2 工作在 8 位方式，该位为两个 8 位定时器选择外部振荡器时钟源。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。 0：选择定时器 2 外部时钟为系统时钟/12。 1：选择定时器 2 外部时钟为外部时钟/8。注意，外部振荡源时钟/8 与系统时钟同步。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 19.16 TMR2RLL：定时器 2 重载寄存器低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xCA
位 7-0： TMR2RLL：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（当定时器 2 工作在自动重装载方式时）或 TMR2L 寄存器的捕捉值（当定时器 2 工作在捕捉方式时）。								

图 19.17 TMR2RLH：定时器 2 重载寄存器高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xCB
位 7-0： TMR2RLH：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（当定时器 2 工作在自动重装载方式时）或 TMR2H 寄存器的捕捉值（当定时器 2 工作在捕捉方式时）。								

图 19.18 TMR2L：定时器 2 低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xCC
位 7-0： TMR2L：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。								

图 19.19 TMR2H：定时器 2 高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xCD
位 7-0： TMR2H：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。								

19.3 定时器 3

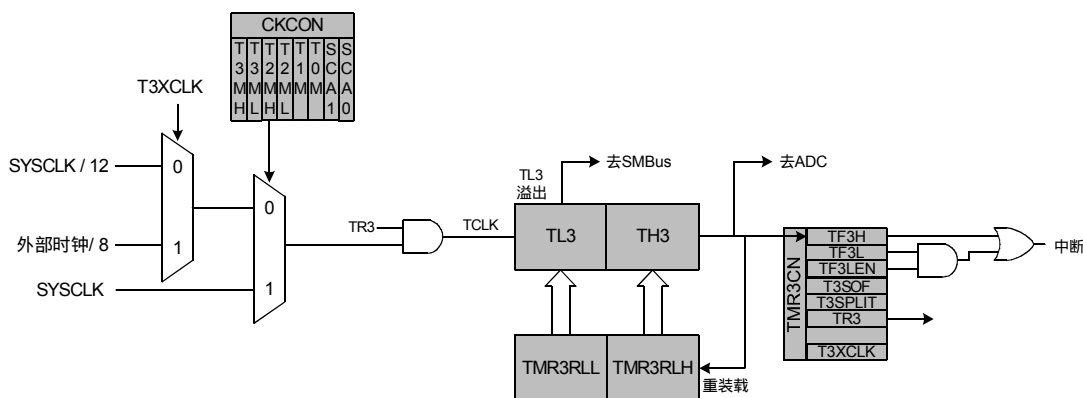
定时器 3 是一个 16 位的计数器/定时器，由两个 8 位的 SFR 组成：TMR3L（低字节）和 TMR3H（高字节）。定时器 3 可以工作在 16 位自动重载方式、8 位自动重载方式（两个 8 位定时器）或帧起始捕捉方式。T3SPLIT 位（TMR3CN.3）和 T3SOF 位（TMR3CN.4）定义定时器 3 的工作方式。

定时器 3 的时钟源可以是系统时钟、系统时钟/12 或外部振荡源时钟/8。在使用实时时钟（RTC）功能时，外部时钟方式是理想的选择，此时用内部振荡器驱动系统时钟，而定时器 3（和/或 PCA）的时钟由一个精确的外部振荡器提供。注意，外部振荡源时钟/8 与系统时钟同步。

19.3.1 16 位自动重载方式

当 T3SPLIT 位（TMR3CN.3）被设置为逻辑‘0’时，定时器 3 工作在自动重载的 16 位定时器方式（见图 19.20）。定时器 3 可以使用 SYSCLK、SYSCLK/12 或外部振荡器时钟/8 作为其时钟源。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，并将定时器 3 高字节溢出标志 TF3H（TMR3CN.7）置‘1’。如果定时器 3 中断被允许，每次溢出都将产生一个中断。如果定时器 3 中断被允许并且 TF3LEN 位（TMR3CN.5）被置‘1’，则每次低 8 位（TMR3L）溢出时（从 0xFF 到 0x00）将产生一个中断。

图 19.20 定时器 3 的 16 位方式原理框图



19.3.2 8 位自动重载定时器方式

当 T3SPLIT 位 (TMR3CN.3) 被置 ‘1’ 时, 定时器 3 工作双 8 位定时器方式。两个 8 位定时器 TMR3H 和 TMR3L 都工作在自动重载方式 (见图 19.21)。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。

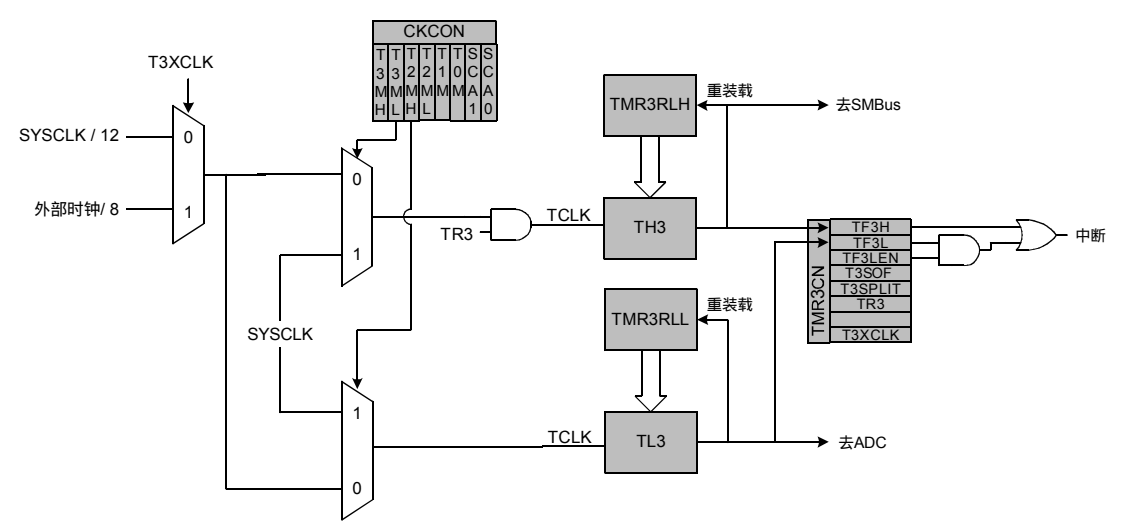
每个 8 位定时器都可以被配置为使用 SYSCLK、SYSCLK/12 或外部振荡器时钟/8 作为其时钟源。定时器 3 时钟选择位 T3MH 和 T3ML (位于 CKCON 中) 选择 SYSCLK 或由定时器 3 外部时钟选择位 (TMR3CN 中的 T3XCLK) 定义的时钟源。时钟源的选择情况如下所示。

T3MH	T3XCLK	TMR3H 时钟源	T3ML	T3XCLK	TMR3L 时钟源
0	0	SYSCLK/12	0	0	SYSCLK/12
0	1	外部时钟/8	0	1	外部时钟/8
1	X	SYSCLK	1	X	SYSCLK

注：外部时钟 8 分频与系统时钟同步，在该方式，外部时钟频率必须小于或等于系统时钟频率。

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, 软件应检查 TF3H 和 TF3L 标志, 以确定中断的来源。TF3H 和 TF3L 标志不能被硬件自动清除, 必须通过软件清除。

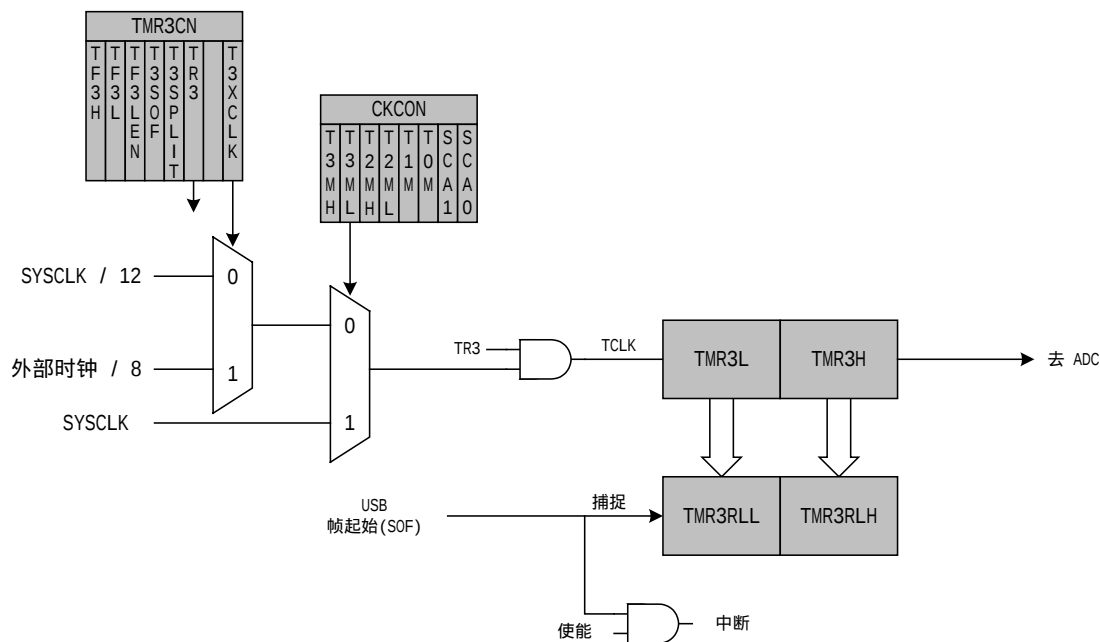
图 19.21 定时器 3 的 8 位方式原理框图



19.3.3 USB 帧起始捕捉方式

当 T3SOF = '1' 时，定时器 3 工作在 USB 帧起始（SOF）捕捉方式。当 T3SPLIT = '0' 时，定时器 3 向上计数，并在计数值从 0xFFFF 变为 0x0000 时溢出。每次收到 SOF 时，定时器 3 寄存器(TMR3H:TMR3L)的内容被锁存到定时器 3 重载寄存器(TMR3RLH:TMR3RLL)中，并产生定时器 3 中断（如果被使能）。该方式可被用于校准系统时钟或外部振荡器（用已知 USB 主机 SOF 时钟）。

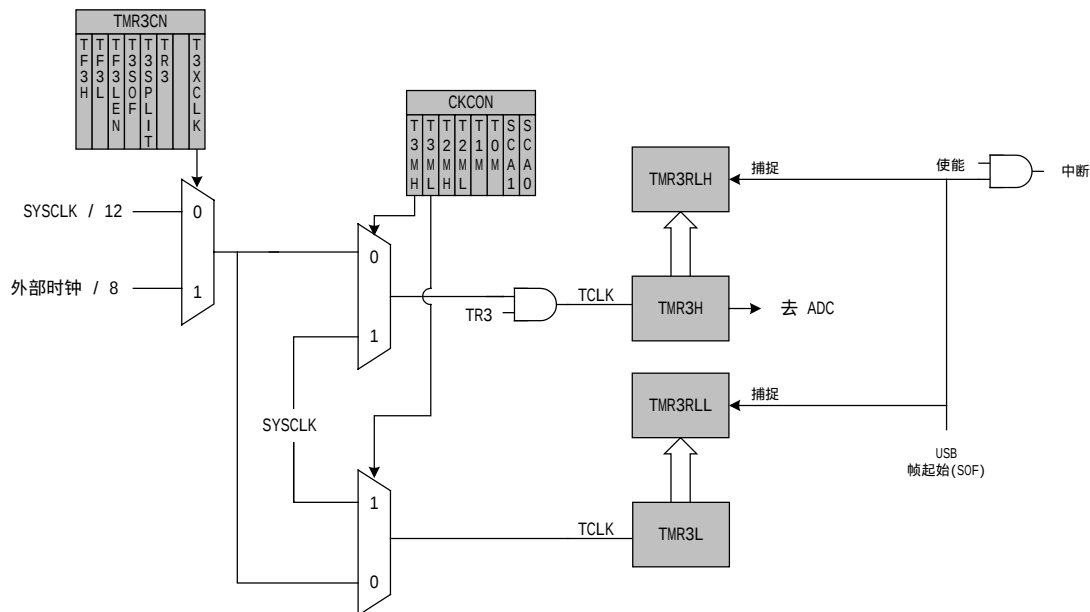
图 19.22 定时器 3 的 SOF 捕捉方式 (T3SPLIT = '0')



C8051F320/1 全速 USB ISP FLASH 微控制器

当 T3SPLIT = '1' 时，定时器 3 寄存器 (TMR3H:TMR3L) 分成两个 8 位计数器。每个计数器都独立地向上计数并在计数值从 0xFF 变为 0x00 时溢出。每次收到 SOF 时，定时器 3 寄存器被锁存到定时器 3 重载寄存器 (TMR3RLH:TMR3RLL) 中，并产生定时器 3 中断 (如果被使能)。

图 19.23 定时器 3 的 SOF 捕捉方式 (T2SPLIT = '1')



C8051F320/1 全速 USB ISP FLASH 微控制器

图 19.24 TMR3CN：定时器 3 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
TF3H	TF3L	TF3LEN	T3SOF	T3SPLIT	TR3	-	T3XCLK	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0x91
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 该位允许/禁止定时器 3 低字节中断。如果 TF3LEN 被置‘1’并且定时器 3 中断被允许，则当定时器 3 低字节发生溢出时将产生一个中断。 0：禁止定时器 3 低字节中断。 1：允许定时器 3 低字节中断。 位 4： T3SOF：定时器 3 帧起始捕捉允许位 0：禁止 SOF 捕捉。 1：允许 SOF 捕捉。每次收到 SOF 时，定时器 3 寄存器（TMR3H:TMR3L）的内容被锁存到定时器 3 重载寄存器（TMR3RLH:TMR3RLL）中，并产生定时器 3 中断（如果被使能）。 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 位自动重载定时器方式。 0：定时器 3 工作在 16 位自动重载方式。 1：定时器 3 工作在双 8 位自动重载定时器方式。 位 2： TR3：定时器 3 运行控制 该位允许/禁止定时器 3。在 8 位方式，该位只控制 TMR3H，TMR3L 总是处于运行状态。 0：定时器 3 禁止。 1：定时器 3 允许。 位 1： 未用。读 = 0b，写 = 忽略。 位 0： T3XCLK：定时器 3 外部时钟选择 该位选择定时器 3 的外部时钟源。如果定时器 3 工作在 8 位方式，该位为两个 8 位定时器选择外部振荡器时钟源。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。 0：选择定时器 3 外部时钟为系统时钟/12。 1：选择定时器 3 外部时钟为外部时钟/8。注意，外部振荡源时钟/8 与系统时钟同步。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 19.25 TMR3RLL：定时器 3 重载寄存器低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0x92

位 7-0： TMR3RLL：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（当定时器 3 工作在自动重装载方式时）或 TMR3L 寄存器的捕捉值（当定时器 3 工作在捕捉方式时）。

图 19.26 TMR3RLH：定时器 3 重载寄存器高字节

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

位 7-0： TMR3RLH：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（当定时器 3 工作在自动重装载方式时）或 TMR3H 寄存器的捕捉值（当定时器 3 工作在捕捉方式时）。

图 19.27 TMR3L：定时器 3 低字节

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

位 7-0： TMR3L：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。

图 19.28 TMR3H：定时器 3 高字节

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

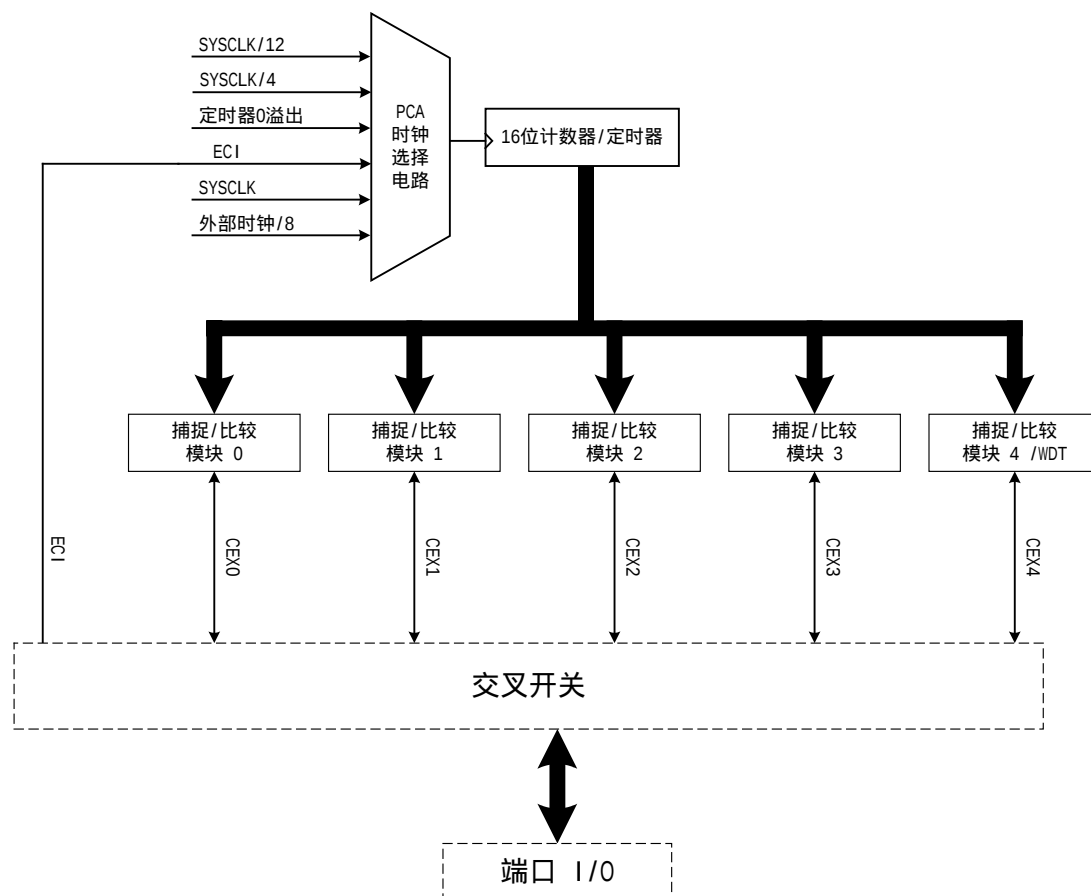
位 7-0： TMR3H：定时器 3 的高字节
在 16 位方式 ,TMR3H 寄存器保持 16 位定时器 3 的高字节。在 8 位方式 ,TMR3H 中保持 8 位高字节定时器的计数值。

20. 可编程计数器阵列 (PCA0)

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5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 (CEX_n)，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，时基信号可以在六个时钟源中选择：系统时钟、系统时钟/4、系统时钟/12、外部振荡器时钟/8、定时器0溢出或ECI输入引脚上的外部时钟信号。在使用实时时钟 (RTC) 功能时，外部振荡器时钟方式是理想的选择，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。捕捉/比较模块有六种工作方式：边沿触发捕捉、软件定时器、高速输出、频率输出、8位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。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20.1。

注意：PCA的模块4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，对某些寄存器的访问受到限制，详见20.3节。

图 20.1 PCA 原理框图



20.1 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 PCA 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 20.1 所示。**注意：在“外部时钟 8 分频”方式，外部振荡源与系统时钟同步，外部时钟频率必须小于或等于系统时钟频率。**

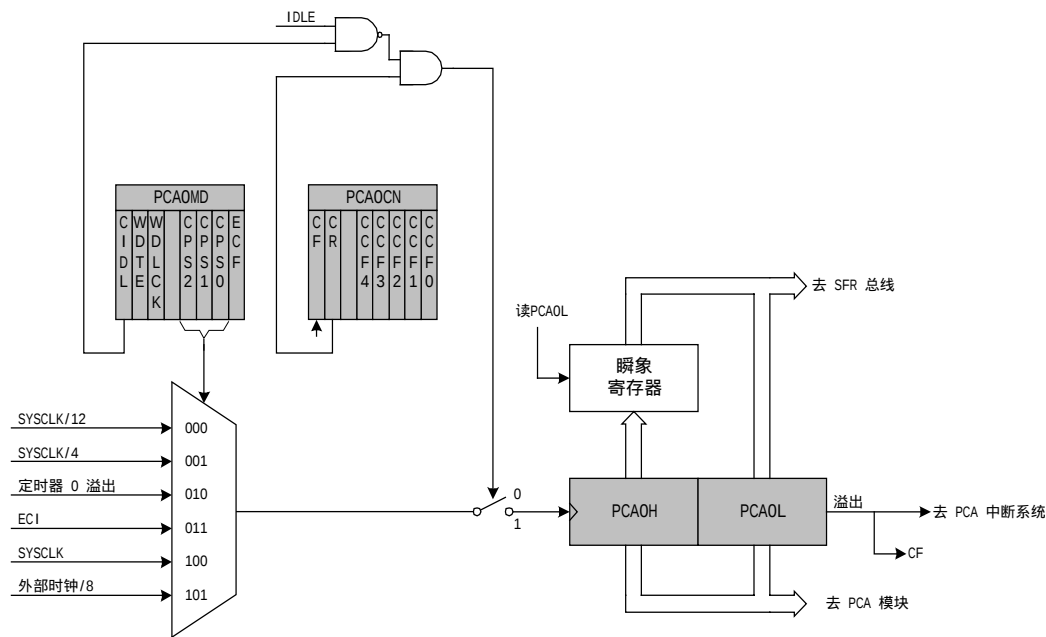
当计数器/定时器溢出时（从 0xFFFF 到 0x0000），PCA0MD 中的计数器溢出标志 (CF) 被置为逻辑‘1’并产生一个中断请求（如果 CF 中断被允许）。将 PCA0MD 中 ECF 位设置为逻辑‘1’即可允许 CF 标志产生中断请求。当 CPU 转向中断服务程序时，CF 位不能被硬件自动清除，必须用软件清除。注意，要使 CF 中断得到响应，必须先总体允许 PCA0 中断。通过将 EA 位(IE.7)和 EPCA0 位(IE1.4)设置为逻辑‘1’来总体允许 PCA0 中断。清除 PCA0MD 寄存器中的 CIDL 位将允许 PCA 在微控制器内核处于空闲方式时继续正常工作。

表 20.1 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	外部振荡器的 8 分频*

*外部振荡器 8 分频与系统时钟同步。

图 20.2 PCA 计数器/定时器原理框图



20.2 捕捉/比较模块

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

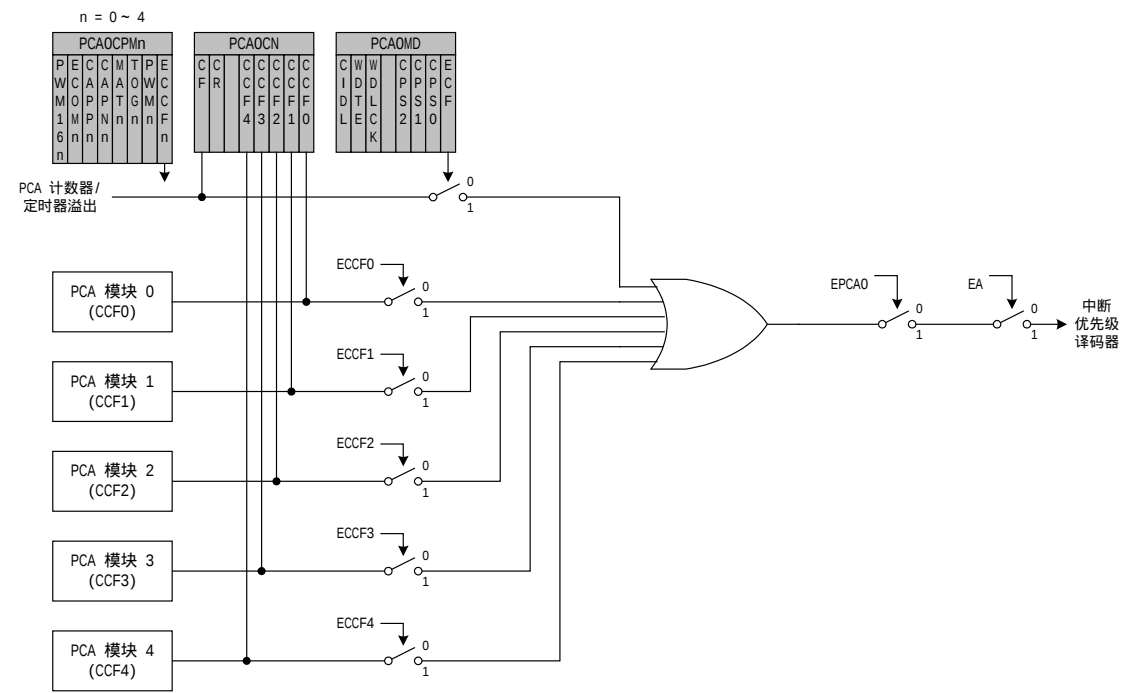
PCA0CPMn寄存器用于配置PCA捕捉/比较模块的工作方式，表20.2概述了模块工作在不同方式时该寄存器各位的设置情况。置‘1’PCA0CPMn寄存器中的ECCFn位将允许模块的CCFn中断。注意：要使单个的CCFn中断得到响应，必须先整体允许PCA0中断。通过将EA位（IE.7）和EPCA0位（EIE1.3）设置为逻辑‘1’来整体允许PCA0中断。PCA0中断配置的详细信息见图20.3。

表 20.2 PCA 捕捉/比较模块的 PCA0CPM 寄存器设置

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

X = 任意

图 20.3 PCA 中断原理框图

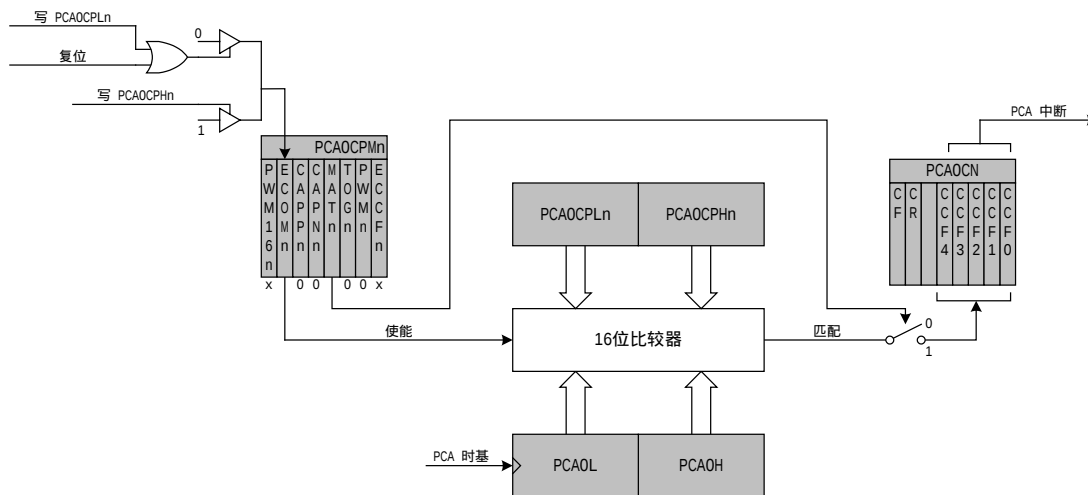


20.2.2 软件定时器方式

软件定时器方式也称为比较器方式。在该方式，PCA将计数器/定时器的计数值与模块的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’并产生一个中断请求（如果CCF中断被允许）。当CPU转向中断服务程序时，CCFn位不能被硬件自动清除，必须用软件清0。置‘1’PCA0CPMn寄存器中的ECOMn和MATn位将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的写入操作将清‘0’ ECOMn位；向PCA0CPHn写入时将置‘1’ ECOMn位。

图 20.5 PCA 软件定时器方式原理框图

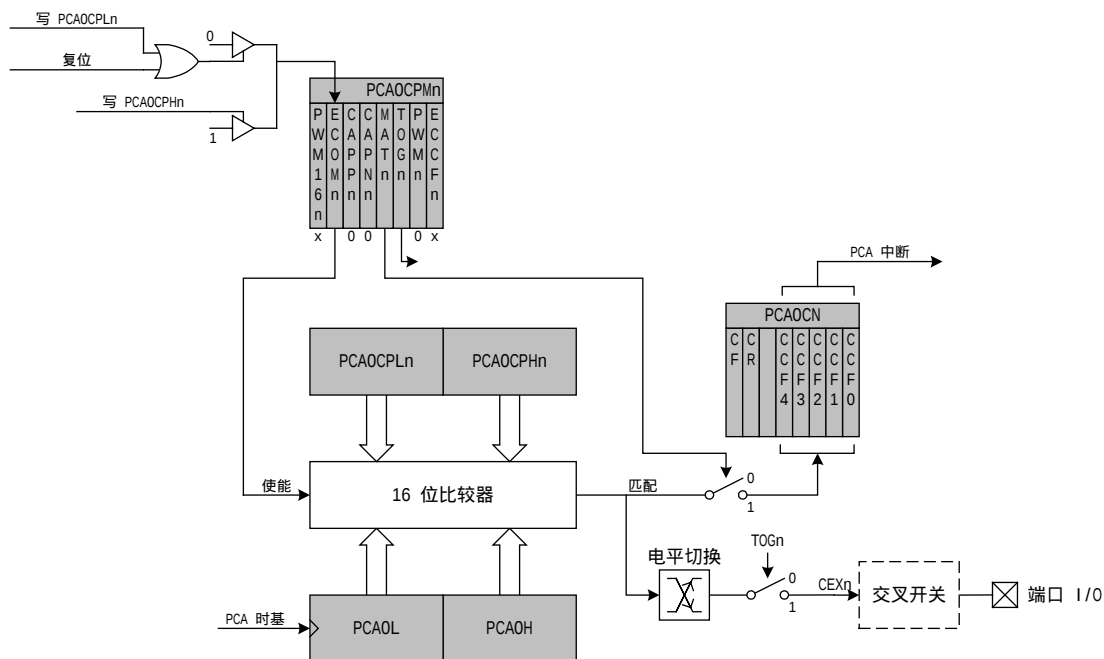


20.2.3 高速输出方式

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变化。置‘1’ PCA0CPMn寄存器中的TOGn、MATn和ECOMn位将使能高速输出方式。

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的写入操作将清‘0’ ECOMn位；向PCA0CPHn写入时将置‘1’ ECOMn位。

图 20.6 PCA 高速输出方式原理框图



20.2.4 频率输出方式

频率输出方式可在模块的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20.1定义：

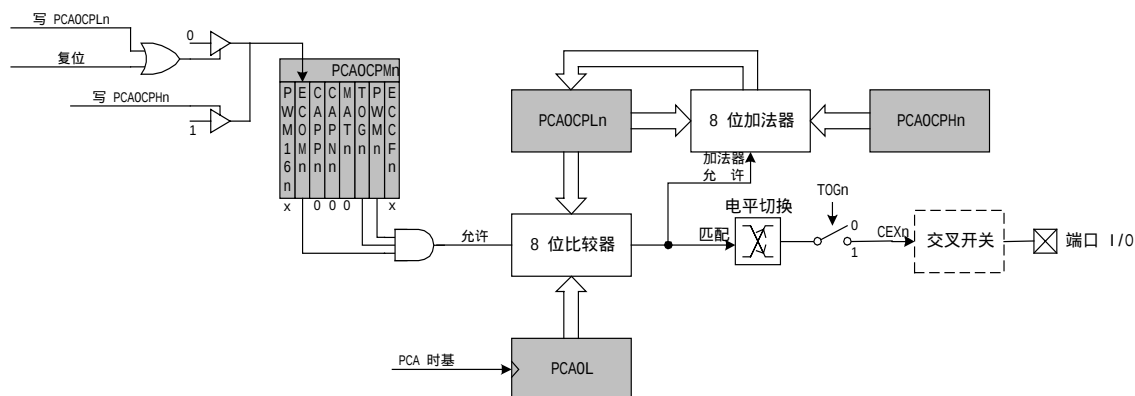
方程20.1 方波输出频率

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

注：对于该方程，PCA0CPHn中的值为0x00时，相当于256。

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的电平发生改变，高字节中的偏移值被加到PCA0CPLn。通过将PCA0CPMn寄存器中ECOMn、TOGn和PWMn位置‘1’来使能频率输出方式。

图 20.7 PCA 频率输出方式原理框图



每个模块都可以被独立地用于在对应的CEX_n引脚产生脉宽调制（PWM）输出。PWM输出的频率取决于PCA计数器/定时器的时基。使用模块的捕捉/比较寄存器PCA0CPL_n改变PWM输出信号的占空比。当PCA计数器/定时器的低字节（PCA0L）与PCA0CPL_n中的值相等时，CEX_n引脚上的输出被置‘1’；当PCA0L中的计数值溢出时，CEX_n输出被复位（见图20.8）。当计数器/定时器的低字节PCA0L溢出时（从0xFF到0x00），保存在PCA0CPH_n中的值被自动装入到PCA0CPL_n，不需软件干预。通过将PCA0CPM_n寄存器中的ECOM_n和PWM_n位置‘1’来使能8位脉冲宽度调制器方式。8位PWM方式的占空比由方程20.2给出。

由方程20.2可知,最大占空比为100%(PCA0CPHn = 0),最小占空比为0.39%(PCA0CPHn = 0xFF)。可以通过清‘0’ECOMn位产生0%的占空比。

20.2.6 16 位脉宽调制器方式

PCA模块还可以工作在16位PWM方式。在该方式下，16位捕捉/比较模块定义PWM信号低电平时间的PCA时钟数。当PCA计数器与模块的值匹配时，CEXn的输出被置为高电平；当计数器溢出时，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CCF_n匹配中断同步。通过将PCA0CPM_n寄存器中的ECOM_n、PWM_n和PWM16_n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（ECCF_n = 1并且MAT_n = 1），以同步对捕捉/比较寄存器的写操作。16位PWM方式的占空比由方程20.3给出。

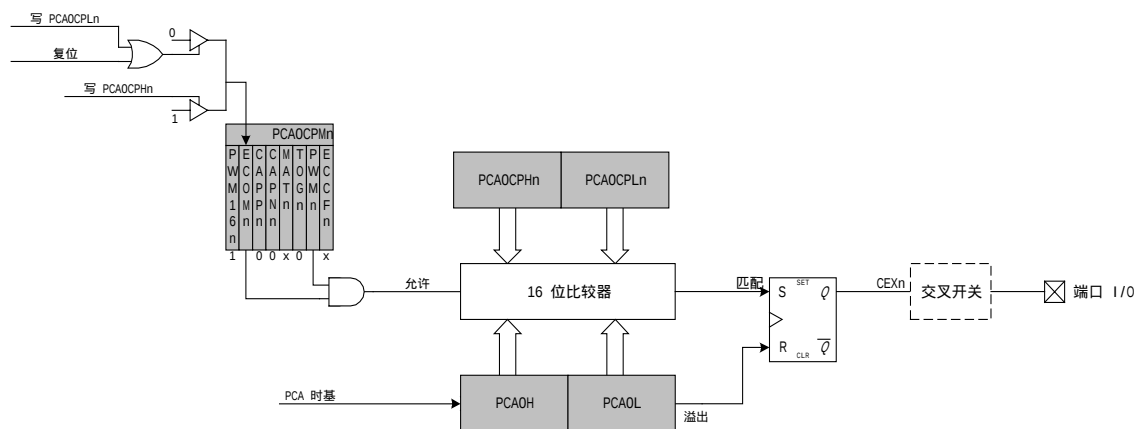
注意：当向PCA0的捕捉/比较寄存器写入一个16位数值时，应先写低字节。向PCA0CPL_n的写入操作将清‘0’ ECOM_n位；向PCA0CPH_n写入时将置‘1’ ECOM_n位。

方程20.3 16位PWM占空比

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

由方程20.3可知，最大占空比为100%（PCA0CP_n = 0），最小占空比为0.0015%（PCA0CP_n = 0xFFFF）。可以通过将ECOM_n位清‘0’产生0%的占空比。

图 20.9 PCA 的 16 位 PWM 方式原理框图



20.3 看门狗定时器方式

通过 PCA 的模块 4 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 (PCA0CPH4) 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 1 被作为看门狗定时器 (WDT) 使用。模块 4 高字节与 PCA 计数器的高字节比较; 模块 4 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。

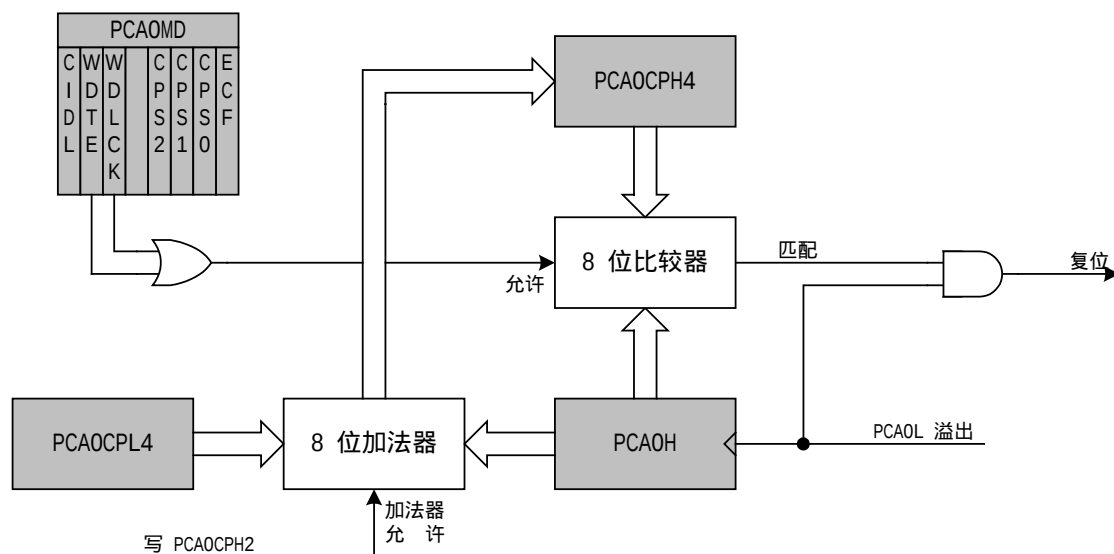
20.3.1 看门狗定时器操作

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 等待控制位 (CIDL) 被冻结。
- 模块 4 被强制进入软件定时器方式。
- 对模块 4 方式寄存器 (PCA0CPM4) 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 PCA0CPH4 和 PCA0H 发生匹配, 则系统将被复位。为了防止 WDT 复位, 需要通过写 PCA0CPH4 来更新 WDT(写入值可以是任意值)。在写 PCA0CPH4 时, PCA0H 的值加上 PCA0CPL4 中保存的偏移值后被装入到 PCA0CPH4(见图 20.10)。

图 20.10 PCA 模块 4 的看门狗定时器方式



保存在 PCA0CPH4 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 20.4 给出，其中 PCA0L 是执行更新操作时 PCA0L 寄存器的值。

方程20.4 看门狗定时器偏移值（PCA时钟数）

$$\text{偏移值} = (256 \times \text{PCA0CPL4}) + (256 - \text{PCA0L})$$

当 PCA0L 发生溢出并且 PCA0CPH4 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 CCF4 标志（PCA0CN.4）写 ‘1’ 来强制产生 WDT 复位。

20.3.2 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 PCA0CPL4 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。
- （选项）通过将 WDLCK 位置 ‘1’ 来锁定 WDT（防止在下一次系统复位前禁止 WDT）。

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 WDLCK 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设置为使能状态。PCA0 计数器的缺省时钟为系统时钟的十二分频（SYSCLK/12）。PCA0L 和 PCA0CPL4 的缺省值均为 0x00，因此 WDT 的超时间隔为 256 个系统时钟。表 20.3 列出了对应某些典型系统时钟频率的超时间隔。

表 20.3 看门狗定时器超时间隔*

系统时钟 (Hz)	PCA0CPL2	超时间隔 (ms)
12,000,000	255	65.5
12,000,000	128	33.0
12,000,000	32	8.4
18,432,000	255	42.7
18,432,000	128	21.5
18,432,000	32	5.5
11,059,200	255	71.1
11,059,200	128	35.8
11,059,200	32	9.2
4,000,000	255	196.6
4,000,000	128	99.1
4,000,000	32	25.3
32,000	255	24,576.0
32,000	128	12,384.0
32,000	32	3,168.0

C8051F320/1 全速 USB ISP FLASH 微控制器

*假设 PCA 使用 SYSCLK/12 作为时钟源，更新时 PCA0L 的值为 0x00。
**内部振荡器复位频率。

20.4 PCA 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。

图 20.11 PCA0CN : PCA 控制寄存器

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	复位值
CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： (可位寻址) 0xD8
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 该位允许/禁止 PCA 计数器/定时器。 0：禁止 PCA 计数器/定时器。 1：允许 PCA 计数器/定时器。								
位 5：未用。读 = 0b，写 = 忽略。								
位 4：CCF4：PCA 模块 4 捕捉/比较标志 在发生一次匹配或捕捉时该位由硬件置位。当 CCF4 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 3：CCF3：PCA 模块 3 捕捉/比较标志 在发生一次匹配或捕捉时该位由硬件置位。当 CCF3 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 2：CCF2：PCA 模块 2 捕捉/比较标志 在发生一次匹配或捕捉时该位由硬件置位。当 CCF2 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 1：CCF1：PCA 模块 1 捕捉/比较标志 在发生一次匹配或捕捉时该位由硬件置位。当 CCF1 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 0：CCF0：PCA 模块 0 捕捉/比较标志 在发生一次匹配或捕捉时该位由硬件置位。当 CCF0 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。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 20.12 PCA0MD : PCA 方式寄存器

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	SFR地址： 0xD9

位 7：

CIDL：PCA 计数器/定时器等待控制

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 4 被用作看门狗定时器。

0：看门狗定时器被禁止。

1：PCA 模块 4 被用作看门狗定时器。

位 5：

WDLCK：看门狗定时器锁定位

该位使能并锁定看门狗定时器。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：看门狗定时器未被锁定。

1：看门狗定时器被使能并被锁定。

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	外部时钟的 8 分频*
1	1	0	保留
1	1	1	保留

*外部振荡器 8 分频与系统时钟同步。

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 寄存器的值。若要改变 PCA0MD 的内容，必须先禁止看门狗定时器。

C8051F320/1 全速 USB ISP FLASH 微控制器

图 20.13 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	
PCA0CPMn 地址: PCA0CPM0 = 0xDA(n=0) PCA0CPM1 = 0xDB(n=1) PCA0CPM2 = 0xDC(n=2) PCA0CPM3 = 0xDD(n=3) PCA0CPM4 = 0xDE(n=4)								
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 位 PWM。 1: 选择 16 位 PWM。								
位 6: ECOMn: 比较器功能使能 该位使能/禁止 PCA 模块 n 的比较器功能。 0: 禁止。 1: 使能。								
位 5: CAPPn: 正沿捕捉功能使能 该位使能/禁止 PCA 模块 n 的正边沿捕捉。 0: 禁止。 1: 使能。								
位 4: CAPNn: 负沿捕捉功能使能 该位使能/禁止 PCA 模块 n 的负边沿捕捉。 0: 禁止。 1: 使能。								
位 3: MATn: 匹配功能使能 该位使能/禁止 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'。 0: 禁止。 1: 使能。								
位 2: TOGn: 电平切换功能使能 该位使能/禁止 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。 0: 禁止。1: 使能。								
位 1: PWMn: 脉宽调制方式使能 该位使能/禁止 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 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。 0: 禁止。 1: 使能。								
位 0: ECCFn: 捕捉/比较标志中断允许 该位设置捕捉/比较标志 (CCFn) 的中断屏蔽。 0: 禁止 CCFn 中断。 1: 当 CCFn 位被置 '1' 时, 允许捕捉/比较标志的中断请求。								

C8051F320/1 全速 USB ISP FLASH 微控制器

图 20.14 PCA0L : PCA 计数器 / 定时器低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址 : 0xF9
位 7-0 : PCA0L : 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)。								

图 20.15 PCA0H : PCA 计数器 / 定时器高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址 : 0xFA
位 7-0 : PCA0H : 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)。								

图 20.16 PCA0CPLn : PCA 捕捉模块低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
PCA0CPLn 地址 : PCA0CPL0 = 0xFB (n=0) PCA0CPL1 = 0xE9 (n=1)								
PCA0CPL2 = 0xEB (n=2) PCA0CPL3 = 0xED (n=3)								
PCA0CPL4 = 0xFD (n=4)								
位 7-0 : PCA0CPLn : 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)。								

图 20.17 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
位7	位6	位5	位4	位3	位2	位1	位0	00000000
PCA0CPHn 地址 : PCA0CPH0=0xFC (n=0) PCA0CPH1=0xEA (n=1)								
PCA0CPH2=0xEC (n=2) PCA0CPH3=0xEE (n=3)								
PCA0CPH4=0xFE (n=4)								
位 7-0 : PCA0CPHn : 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)。								

21. C2 接口

C8051F320/1 有一个 Silicon Labs 2 线 (C2) 调试接口, 支持 FLASH 编程、边界扫描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。

21.1 C2 接口寄存器

下面对与 FLASH 编程和边界扫描功能有关的 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)。

图 21.1 C2ADD : C2 地址寄存器

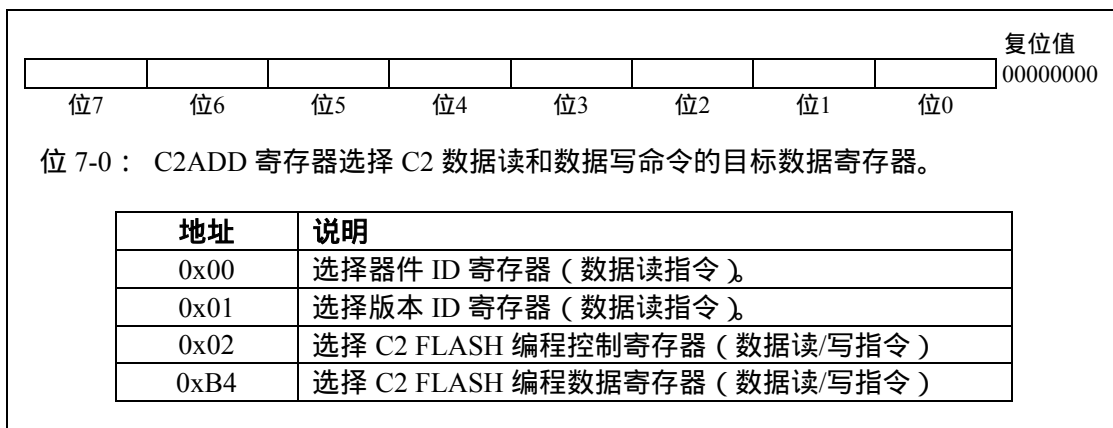


图 21.2 DEVICEID : C2 器件 ID 寄存器

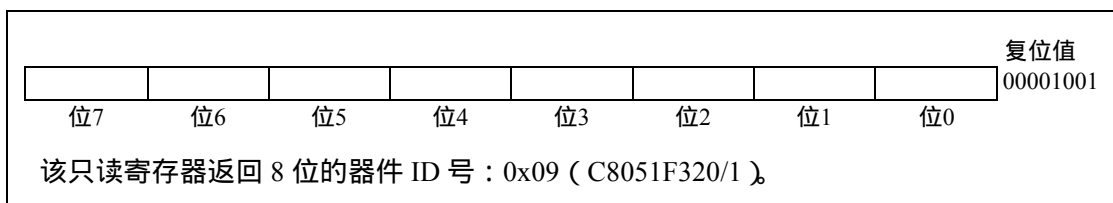


图 21.3 REVID : C2 版本 ID 寄存器

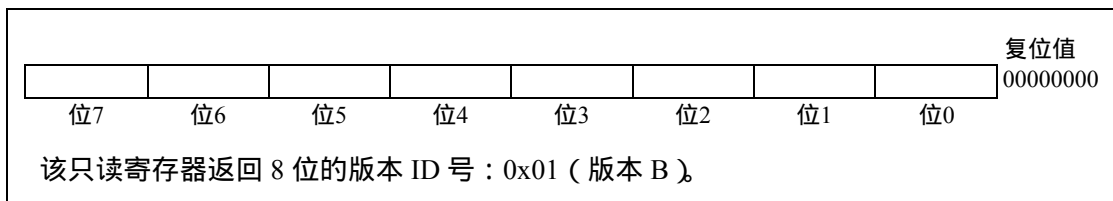


图 21.4 FPCTL：C2 FLASH 编程控制寄存器

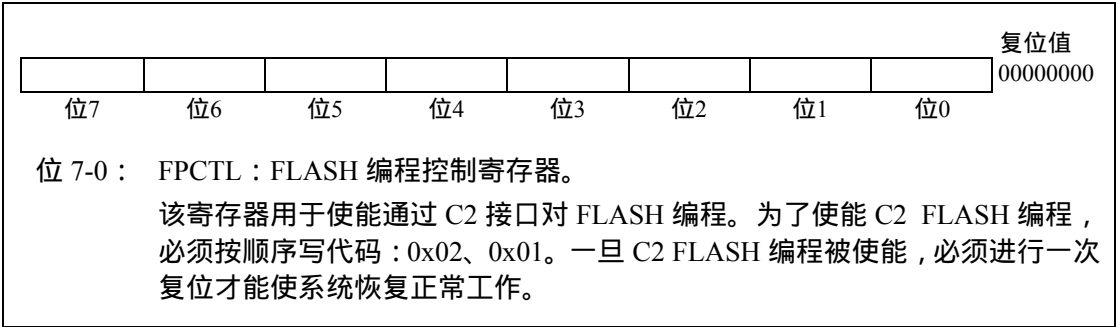


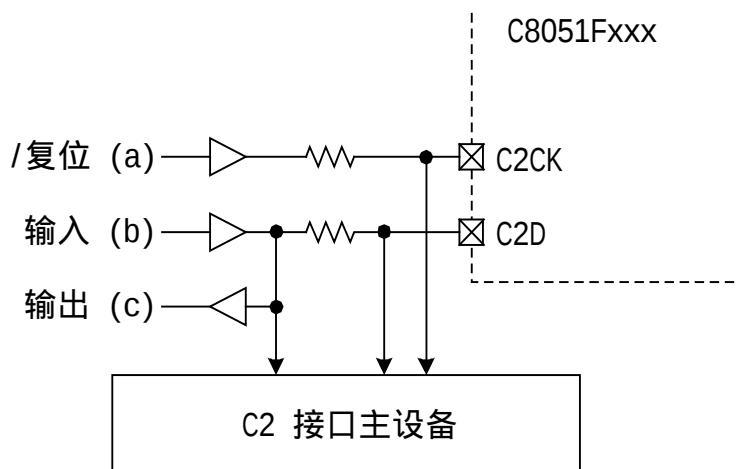
图 21.5 FPDAT：C2 FLASH 编程数据寄存器



21.2 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、FLASH 编程和边界扫描。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。在这种状态下片内外设和用户软件停止工作，C2 接口可以安全地“借用”C2CK（正常方式为/RST）和 C2D（正常方式为 P3.0）引脚。在大多数情况下，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 21.6 所示。

图 21.6 典型 C2 引脚共享电路



在图 21.6 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) 不能改变状态。
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.

4635 Boston Lane
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