

**C8051F300/1/2/3/4/5**  
**混合信号 ISP FLASH 微控制器**  
**数 据 手 册**

**潘 琢 金 译**

**Rev 2.1 2002.10**

**版权所有**

## 版 权 声 明

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

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

### 译者联系方式：

沈阳航空工业学院计算机系 潘琢金

电话：024-86802267 ； 13066535936

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

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 模拟外设

- 8 位 ADC
  - 可编程转换速率, 最大 100ksps
  - 可多达 8 个外部输入
  - 可编程放大器增益: 4、2、1、0.5
  - VREF 可在外部引脚或 VDD 中选择
  - 内置温度传感器 ( $\pm 3^{\circ}\text{C}$ )
  - 外部转换启动输入
- 两个模拟比较器
  - 可编程回差电压和响应时间
  - 可配置为中断或复位源
  - 小电流 ( $<0.5\mu\text{A}$ )

## 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好的性能
- 完整的开发套件

## 供电电压.....2.7V - 3.6V

- 典型工作电流: 5mA @ 25MHz  
11 $\mu\text{A}$  @ 32KHz
- 典型停机电流: 0.1 $\mu\text{A}$
- 温度范围:  $-40^{\circ}\text{C}$  -  $+85^{\circ}\text{C}$

## 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

- 256 字节内部数据 RAM
- 8K 字节闪存存储器; 可以在系统编程, 扇区大小为 512 字节

## 数字外设

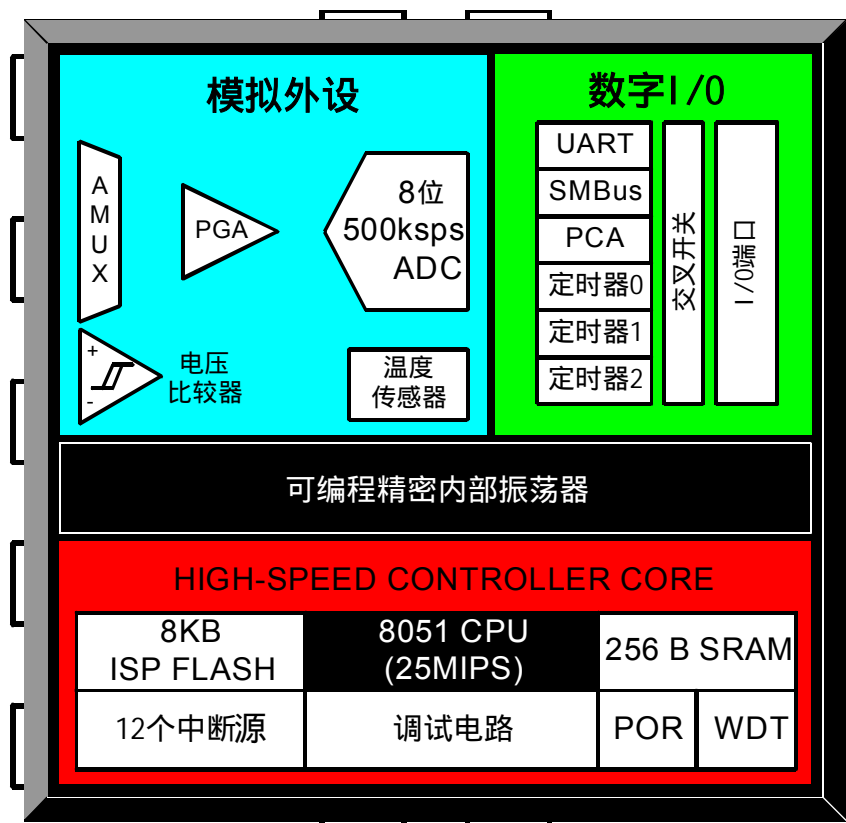
- 8 个端口 I/O; 所有口线均耐 5V 电压
- 硬件增强型 UART 和 SMBus 串口
- 16 位可编程计数器/定时器阵列, 有 3 个捕捉/比较模块
- 使用 PCA 或定时器和外部时钟源的实时时钟方式
- 专用的看门狗定时器; 双向复位引脚

## 时钟源

- 内部可编程振荡器: 24.5MHz,  $\pm 2\%$ 的精度, 可支持 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, 用于节电方式

## 11 脚微型封装

- 3x3mm; 实际 MLP 尺寸:



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

---

<b>1. 系统概述</b> .....	<b>6</b>
1.1 CIP-51™ CPU .....	8
1.1.1 与 8051 完全兼容 .....	8
1.1.2 速度提高 .....	8
1.1.3 增加的功能 .....	9
1.2 片内存储器 .....	10
1.3 片内调试电路 .....	11
1.4 可编程数字 I/O 和交叉开关 .....	12
1.5 串行端口 .....	12
1.6 可编程计数器阵列 .....	13
1.7 8 位模/数转换器 (只限于 C8051F300/302) .....	14
1.8 比较器 .....	15
<b>2. 极限参数</b> .....	<b>16</b>
<b>3. 总体直流电气特性</b> .....	<b>16</b>
<b>4. 引脚和封装定义</b> .....	<b>17</b>
<b>5. ADC0 (8 位, 只限于 C8051F300/2)</b> .....	<b>20</b>
5.1 模拟多路选择器和 PGA .....	21
5.1 温度传感器 .....	22
5.3 工作方式 .....	24
5.3.1 转换启动方式 .....	24
5.3.2 跟踪方式 .....	25
5.3.3 建立时间要求 .....	26
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 .....	30
5.4.2 差分方式下的窗口检测器 .....	31
<b>6. 电压基准</b> .....	<b>33</b>
<b>7. 比较器</b> .....	<b>35</b>
<b>8. CIP-51 CPU</b> .....	<b>40</b>
8.1 指令集 .....	41
8.1.1 指令和 CPU 时序 .....	41
8.1.2 MOVX 指令和程序存储器 .....	41
8.2 存储器组织 .....	45

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

8.2.1 程序存储器.....	45
8.2.2 数据存储器.....	45
8.2.3 通用寄存器.....	46
8.2.4 位寻址空间.....	46
8.2.5 堆栈.....	46
8.2.6 特殊功能寄存器.....	47
8.2.7 寄存器说明.....	50
8.3 中断系统.....	53
8.3.1 MCU 中断源和中断向量.....	53
8.3.2 外部中断.....	53
8.3.3 中断优先级.....	54
8.3.4 中断响应时间.....	54
8.3.5 中断寄存器说明.....	56
8.4 电源管理方式.....	61
8.4.1 等待方式.....	61
8.4.2 停机方式.....	61
<b>9. 复位源.....</b>	<b>63</b>
9.1 上电复位.....	64
9.2 掉电复位和 VDD 监视器.....	64
9.3 外部复位.....	65
9.4 时钟丢失检测器复位.....	65
9.5 比较器 0 复位.....	65
9.6 PCA 看门狗定时器复位.....	65
9.7 FLASH 错误复位.....	65
9.8 软件复位.....	66
<b>10. FLASH 存储器.....</b>	<b>68</b>
10.1 FLASH 存储器编程.....	68
10.1.1 FLASH 锁定和关键码功能.....	68
10.1.2 FLASH 擦除.....	68
10.1.3 FLASH 写.....	69
10.2 非易失性数据存储.....	69
10.3 安全选项.....	69
<b>11. 振荡器.....</b>	<b>72</b>
11.1 可编程内部振荡器.....	72
11.1.1 C8051F300/1 的内部振荡器编程.....	73
11.1.2 C8051F302/3/4/5 的内部振荡器编程.....	73
11.2 外部振荡器驱动电路.....	75
11.3 系统时钟选择.....	75

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

11.4 外部晶体示例.....	77
11.5 外部 RC 示例 .....	77
11.6 外部电容示例.....	77
<b>12. 端口输入/输出 .....</b>	<b>78</b>
12.1 优先级交叉开关译码器.....	79
12.2 端口 I/O 初始化.....	80
12.3 通用端口 I/O .....	84
<b>13. SMBUS.....</b>	<b>86</b>
13.1 支持文档 .....	87
13.2 SMBUS 配置 .....	87
13.2 SMBUS 操作 .....	88
13.3.1 总线仲裁.....	88
13.3.2 时钟低电平扩展.....	89
13.3.3 SCL 低电平超时.....	89
13.3.4 SCL 高电平 (SMBus 空闲) 超时.....	89
13.4 SMBUS 的使用 .....	89
13.4.1 SMBus 配置寄存器.....	90
13.4.2 SMBus 控制寄存器.....	93
13.4.3 数据寄存器.....	95
13.5 SMBUS 传输方式 .....	96
13.5.1 主发送器方式.....	96
13.5.2 主接收器方式.....	97
13.5.3 从接收器方式.....	98
13.5.4 从发送器方式.....	99
13.6 SMBUS 状态译码 .....	100
<b>14. UART0 .....</b>	<b>102</b>
14.1 增强的波特率发生器.....	103
14.2 工作方式 .....	104
14.2.1 8 位 UART.....	104
14.2.2 9 位 UART.....	105
14.3 多机通信 .....	106
<b>15. 定时器 .....</b>	<b>111</b>
15.1 定时器 0 和定时器 1.....	111
15.1.1 方式 0 .....	111
15.1.2 方式 1 .....	112
15.1.3 方式 2 .....	113
15.1.4 方式 3 .....	114
15.2 定时器 2 .....	119

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

---

15.2.1	16 位自动重装载方式.....	119
15.2.2	8 位自动重装载定时器方式.....	120
<b>16.</b>	<b>可编程计数器阵列 .....</b>	<b>123</b>
16.1	PCA 计数器/定时器.....	124
16.2	捕捉/比较模块 .....	125
16.2.1	边沿触发的捕捉方式.....	126
16.2.2	软件定时器方式.....	127
16.2.3	高速输出方式.....	128
16.2.4	频率输出方式.....	129
16.2.5	8 位脉宽调制器方式.....	130
16.2.6	16 位脉宽调制器方式.....	131
16.3	看门狗定时器方式.....	132
16.3.1	看门狗定时器操作.....	132
16.3.2	看门狗定时器的使用.....	133
16.4	PCA 寄存器说明.....	134
<b>17.</b>	<b>C2 接口.....</b>	<b>138</b>
17.1	C2 接口寄存器 .....	138
17.2	C2 引脚共享.....	140

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 1. 系统概述

C8051F300/1/2/3/4/5 器件是完全集成的混合信号系统级 MCU 芯片。下面列出了一些主要特性，有关某一产品的具体特性参见表 1.1。

- 高速、流水线结构的 8051 兼容的 CIP-51 内核（可达 25MIPS）
- 全速、非侵入式的在系统调试接口（片内）
- 真正 8 位 500 ksp/s 的 11 通道 ADC，带 PGA 和模拟多路器。
- 高精度可编程的 25MHz 内部振荡器
- 2/4/8K 字节可在系统编程的 FLASH 存储器
- 256 字节片内 RAM
- 硬件实现的 SMBus/I<sup>2</sup>C 和增强型 UART 串行接口
- 3 个通用的 16 位定时器
- 具有 3 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 监视器和温度传感器
- 片内电压比较器
- 8 个端口 I/O（容许 5V 输入）

具有片内 VDD 监视器、VDD 监视器和时钟振荡器的 C8051F300/1/2/3/4/5 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 Cygnal 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°C 到 +85°C）内用 2.7V-3.6V 的电压工作。端口 I/O、/RST 和 JTAG 引脚都容许 5V 的输入信号电压。C8051F300/1/2/3 为 11 脚 MLP 封装（见图 4.2）。

表 1.1 产品选择指南

	MIPS (峰值)	FLASH 存储器 (字节)	RAM (字节)	校准的内部振荡器	SMBus/I <sup>2</sup> C	UART	定时器(16位)	可编程计数器阵列	数字端口 I/O	8 位 500ksp/s ADC	温度传感器	模拟比较器	封装
C8051F300	25	8K	256	√	√	√	3	√	8	√	√	1	MLP-11
C8051F301	25	8K	256	√	√	√	3	√	8	-	-	1	MLP-11
C8051F302	25	8K	256	-	√	√	3	√	8	√	√	1	MLP-11
C8051F303	25	8K	256	-	√	√	3	√	8	-	-	1	MLP-11
C8051F304	25	4K	256	-	√	√	3	√	8	-	-	1	MLP-11
C8051F305	25	2K	256	-	√	√	3	√	8	-	-	1	MLP-11



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 1.1 C8051F300/2 原理框图

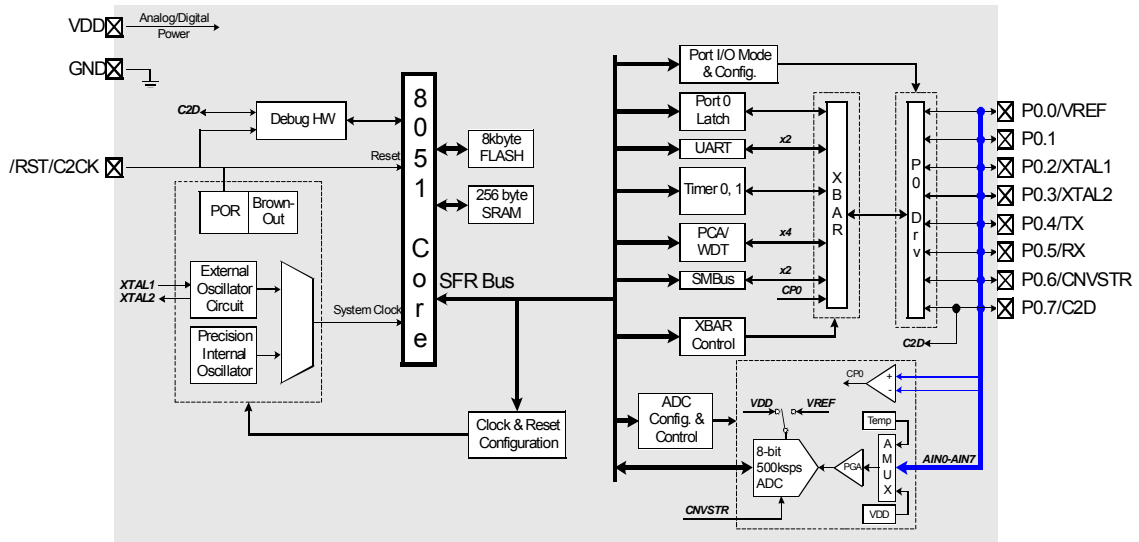
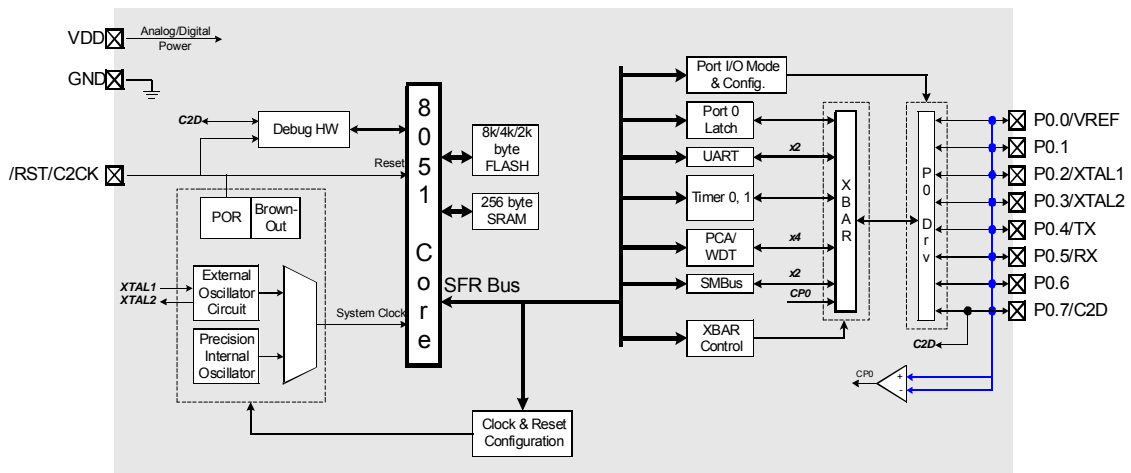


图 1.2 C8051F301/3/4/5 原理框图



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 1.1 CIP-51™ CPU

### 1.1.1 与 8051 完全兼容

C8051F300/1/2/3/4/5 系列器件使用 Cygnal 的专利 CIP-51 微控制器内核。CIP-51 与 MCS-51™ 指令集完全兼容, 可以使用标准 803x/805x 的汇编器和编译器进行软件开发。CIP-51 内核具有标准 8052 的所有外设部件, 包括两个标准的 16 位计数器/定时器、一个具有外部振荡器输入的增强型 16 位计数器/定时器、一个具有增强波特率配置的全双工 UART、256 字节内部 RAM、128 字节特殊功能寄存器 (SFR) 地址空间及 8 个 I/O 端口。

### 1.1.2 速度提高

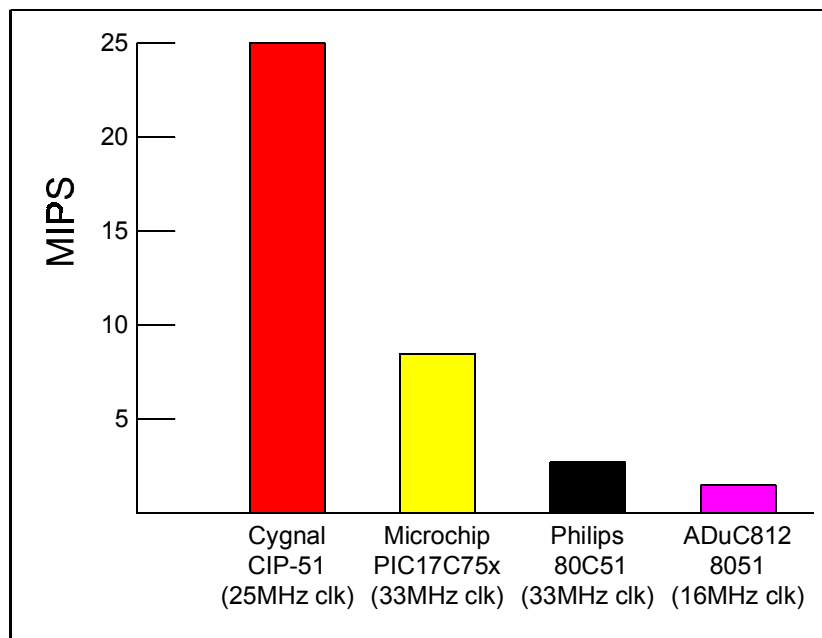
CIP-51 采用流水线结构, 与标准的 8051 结构相比指令执行速度有很大的提高。在一个标准的 8051 中, 除 MUL 和 DIV 以外所有指令都需要 12 或 24 个系统时钟周期, 最大系统时钟频率为 12-24MHz。而对于 CIP-51 内核, 70% 的指令的执行时间为 1 或 2 个系统时钟周期, 只有 4 条指令的执行时间大于 4 个系统时钟周期。

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

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

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

图 1.3 MCU 峰值执行速度比较



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 1.1.3 增加的功能

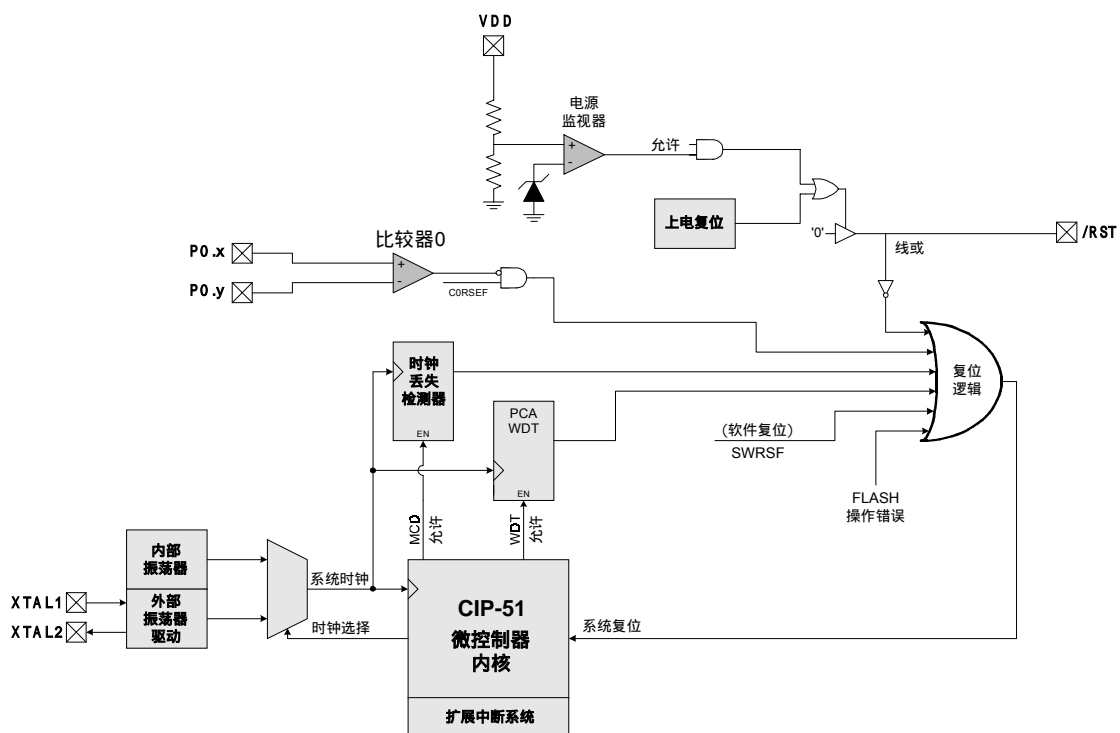
C8051F300/1/2/3/4/5 系列 MCU 在 CIP-51 内核和外设方面有几项关键性的改进，提高了整体性能，更易于在最终应用中使用。

扩展的中断系统向 CIP-51 提供 12 个中断源（标准 8051 只有 7 个中断源），允许大量的模拟和数字外设中断微控制器。一个中断驱动的系统需要较少的 MCU 干预，因而有更高的执行效率。在设计一个多任务实时系统时，这些增加的中断源是非常有用的。

MCU 有多达 8 个复位源：上电复位电路、一个片内 VDD 监视器（当电源电压低于 2.7V 时强制复位）、一个看门狗定时器、一个时钟丢失检测器、一个由比较器 0 提供的电压检测器、一个软件强制复位、复位输入引脚和 FLASH 保护复位。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。

C8051F300/1 器件内部有经过工厂校准的振荡器（ $24.5\text{MHz} \pm 2\%$ ），C8051F302/3/4/5 器件内部的振荡器未经过工厂校准。所有 C8051F300/1/2/3/4/5 器件的内部振荡器的周期都可以由用户以 0.5% 的增量编程。器件内还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 或外部时钟源产生系统时钟。如果需要，时钟源可以在运行时切换到外部振荡器。外部振荡器在低功耗系统中是非常有用的，它允许 MCU 从一个低频率（节电）外部晶体源运行，当需要时再周期性地切换到高速（可达 25MHz）的内部振荡器。

图 1.4 片内时钟和复位



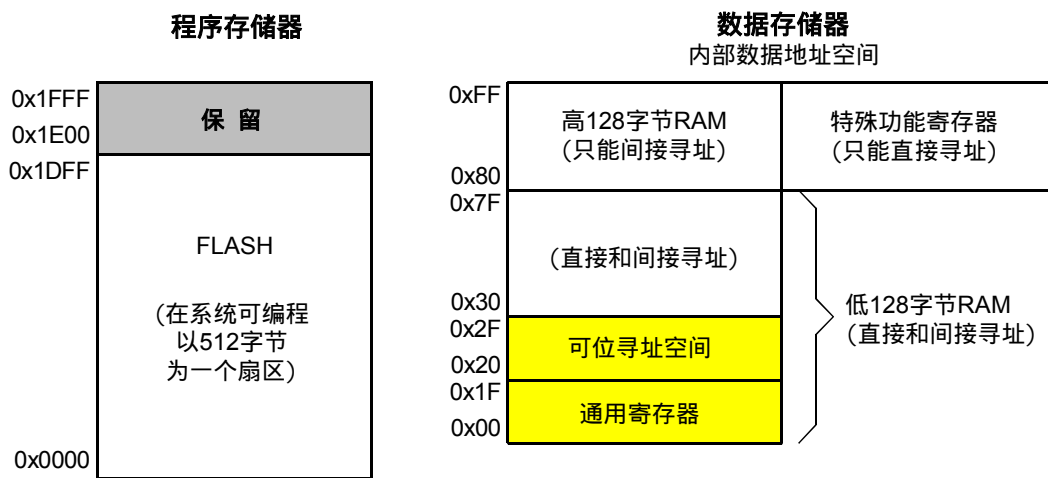
# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 1.2 片内存储器

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

C8051F300/1/2/3 包含 8K 字节的 FLASH 程序存储器(C8051F304 包含 4K 字节,C8051F305 包含 2K 字节)。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 C8051F300/1/2/3 系统的存储器结构。

图 1.5 片内存储器组织 (C8051F300/1/2/3)



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 1.3 片内调试电路

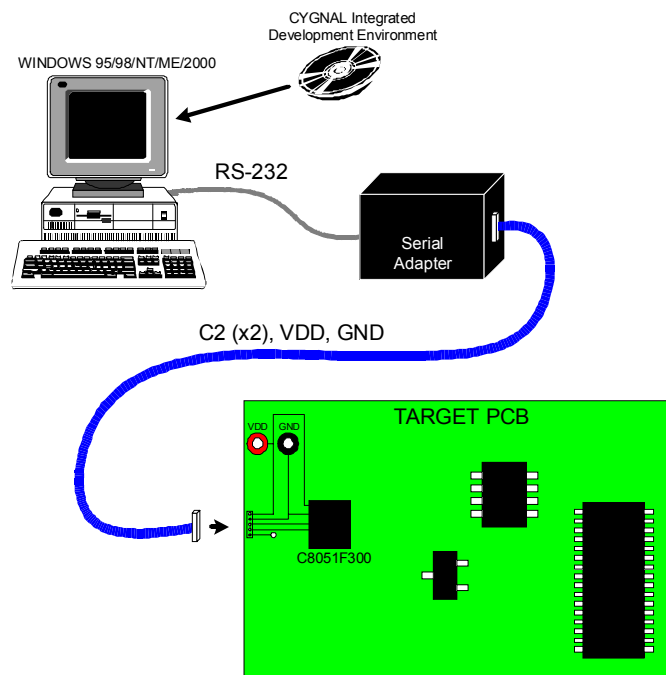
C8051F300/1/2/3/4/5系列具有片内Cygnal 2线（C2）接口调试电路，支持使用安装在最终应用系统中的产品器件进行非侵入式、全速的在系统调试。

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

开发套件C8051F300DK具有开发应用代码和对C8051F300/1/2/3/4/5 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和一大块样机区域。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1.6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。在2.7-3.6V时其电源电流大约为20 mA。对于不能从目标板上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，Cygnal 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。Cygnal的调试环境既便于使用又能保证精确模拟外设的性能。

图1.6 开发/在系统调试示意图



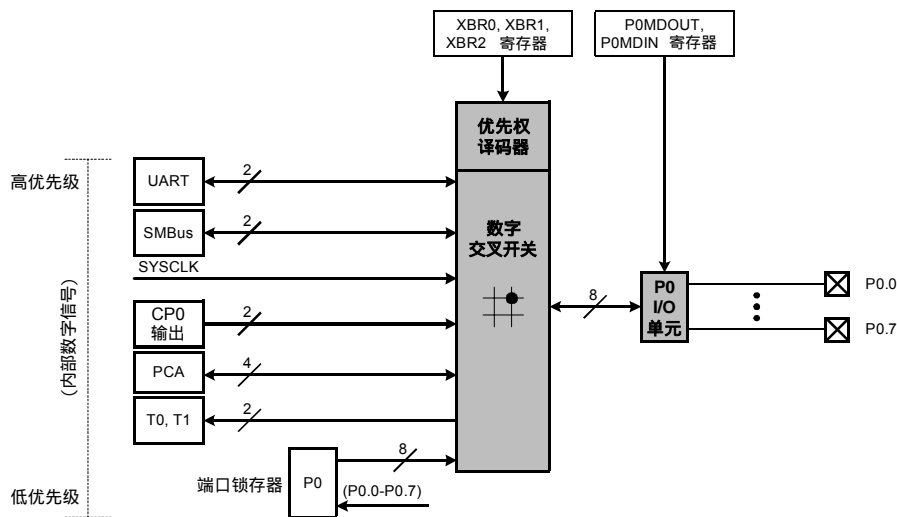
# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 1.4 可编程数字 I/O 和交叉开关

C8051F300/1/2/3/4/5有8个通用端口I/O，这些端口I/O的工作情况与标准8051相似，但有一些改进。每个端口引脚都可以被配置为模拟输入或数字I/O。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。

可能最独特的改进是引入了数字交叉开关。这是一个数字开关网络，允许将内部数字系统资源映射到端口I/O引脚（见图1.7）。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1.7 数字交叉开关原理框图



## 1.5 串行端口

C8051F300/1/2/3/4/5系列MCU内部有一个SMBus/I<sup>2</sup>C接口和一个增强波特率配置的全双工UART。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。

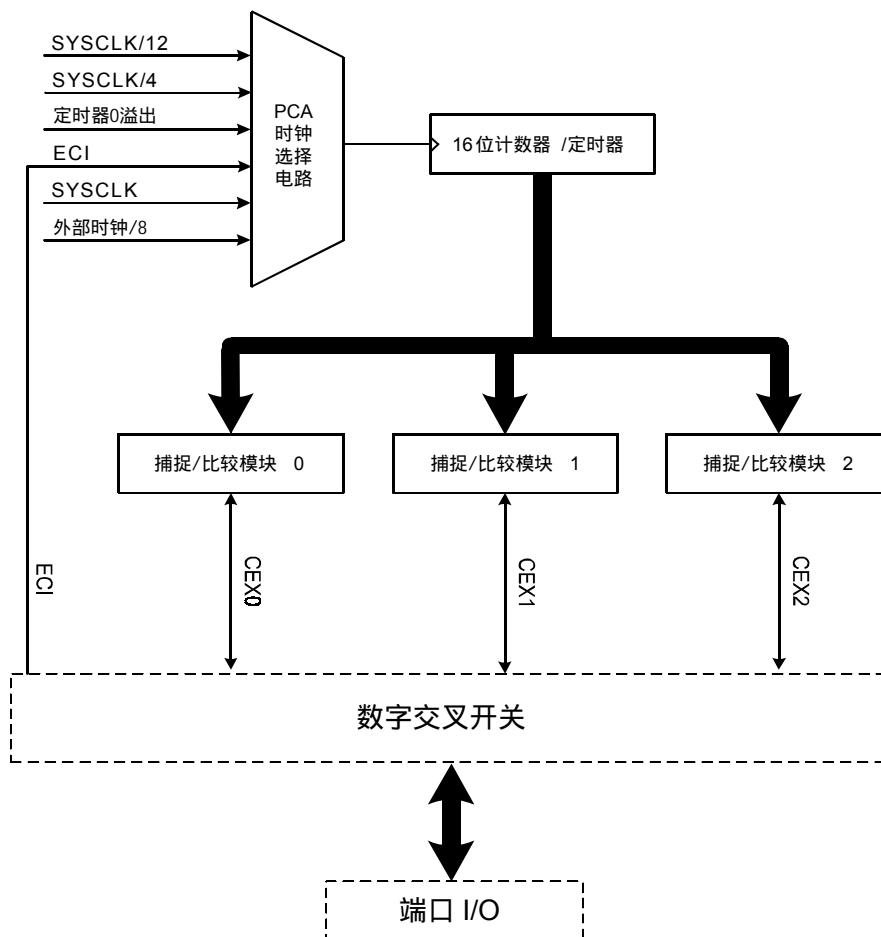
# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 1.6 可编程计数器阵列

除了3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3个可编程的捕捉/比较模块。时间基准的时钟可以是下面的六个时钟源之一：系统时钟/12、系统时钟/4、定时器0溢出、外部时钟输入（ECI）、系统时钟和外部振荡源频率/8。外部时钟源对与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2还提供看门狗定时器（WDT）功能。在系统复位后，捕捉/比较模块2被配置并被使能工作为工作在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引脚。

图1.8 PCA原理框图



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 1.7 8 位模/数转换器（只限于 C8051F300/302）

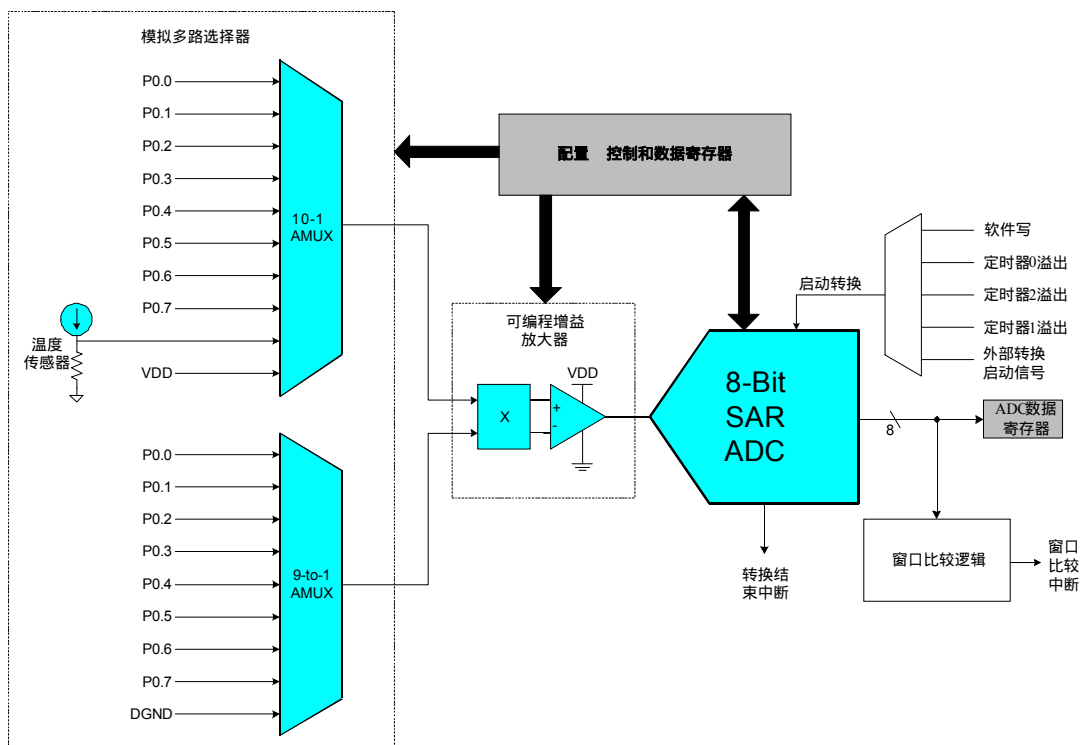
C8051F300/302有一个片内8位SAR ADC，一个10通道差分输入多路选择器和可编程增益放大器。该ADC工作在500ksps的最大采样速率时可提供真正的8位精度，INL为±1LSB。ADC系统包含一个可编程的模拟多路选择器，用于选择ADC的正输入和负输入。每个输入引脚都可以作为ADC的输入；另外，片内温度传感器的输出和电源电压也可以作为ADC的输入。用户固件可以将ADC置于关断状态以节省功耗。

可编程增益放大器（PGA）对ADC输入信号进行放大，增益可以用软件设置为0.5、1、2、4。当不同ADC输入通道之间输入的电压信号范围差距较大或需要放大一个具有较大直流偏移的信号时，这个放大环节是非常有用的。

A/D转换可以有5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溢出或外部转换启动信号。这种灵活性允许用软件事件、外部硬件信号或周期性的定时器溢出信号触发转换。一次转换完成可以产生中断（如果被允许），或者用软件查询一个状态位来判断转换结束。

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

图1.9 8位ADC原理框图





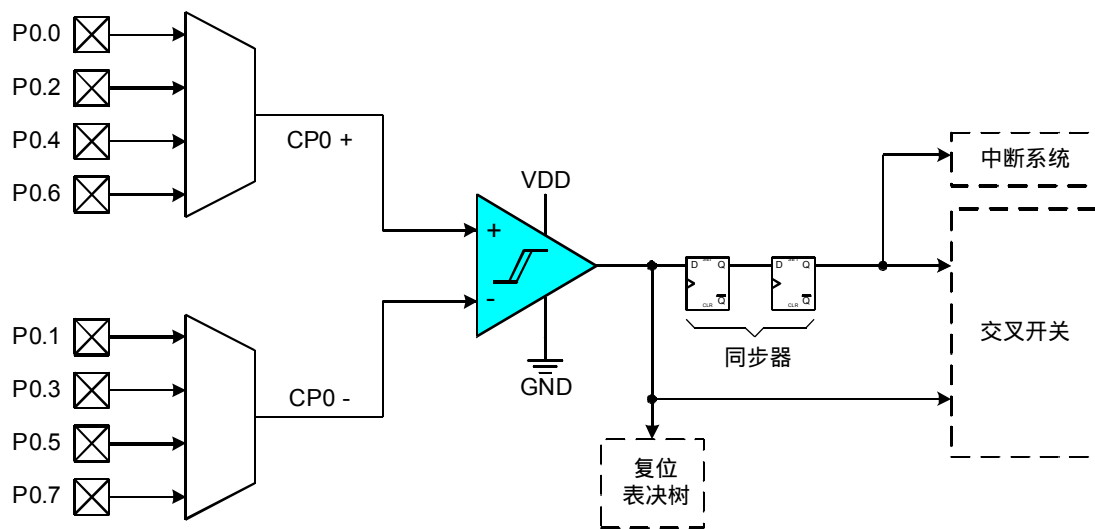
# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 1.8 比较器

C8051F300/1/2/3/4/5器件内部有两个电压比较器，可以由用户软件允许/禁止和配置。所有端口I/O引脚都可被配置为比较器输入。如果需要，可以将两个比较器输出连到端口引脚：一个锁存输出和/或一个未锁存的输出（异步）。比较器的响应时间是可编程的，允许用户在高速和低功耗方式之间选择。比较器的正和负回差电压也是可配置的。

比较器能在上升沿、下降沿产生中断，或在两个边沿都产生中断。当MCU工作在等待方式时，这些中断可用于唤醒MCU。比较器还可以被配置为复位源。

图1.10 比较器原理框图



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 2. 极限参数

表 2.1 极限参数\*

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

## 3. 总体直流电气特性

表 3.1 总体直流电气特性

-40°C到+85°C，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		2.7	3.0	3.6	V
数字电源电流（CPU处于活动状态）	VDD=2.7V, CLK=25MHz VDD=2.7V, CLK= 1MHz VDD=2.7V, CLK=32kHz		5.8 0.34 12		mA mA μA
数字电源电流（CPU不活动，不访问FLASH）	VDD=2.7V, CLK=25MHz VDD=2.7V, CLK= 1MHz VDD=2.7V, CLK=32kHz		2.1 83 2.8		mA μA 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
额定工作温度范围		-40		+85	°C

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

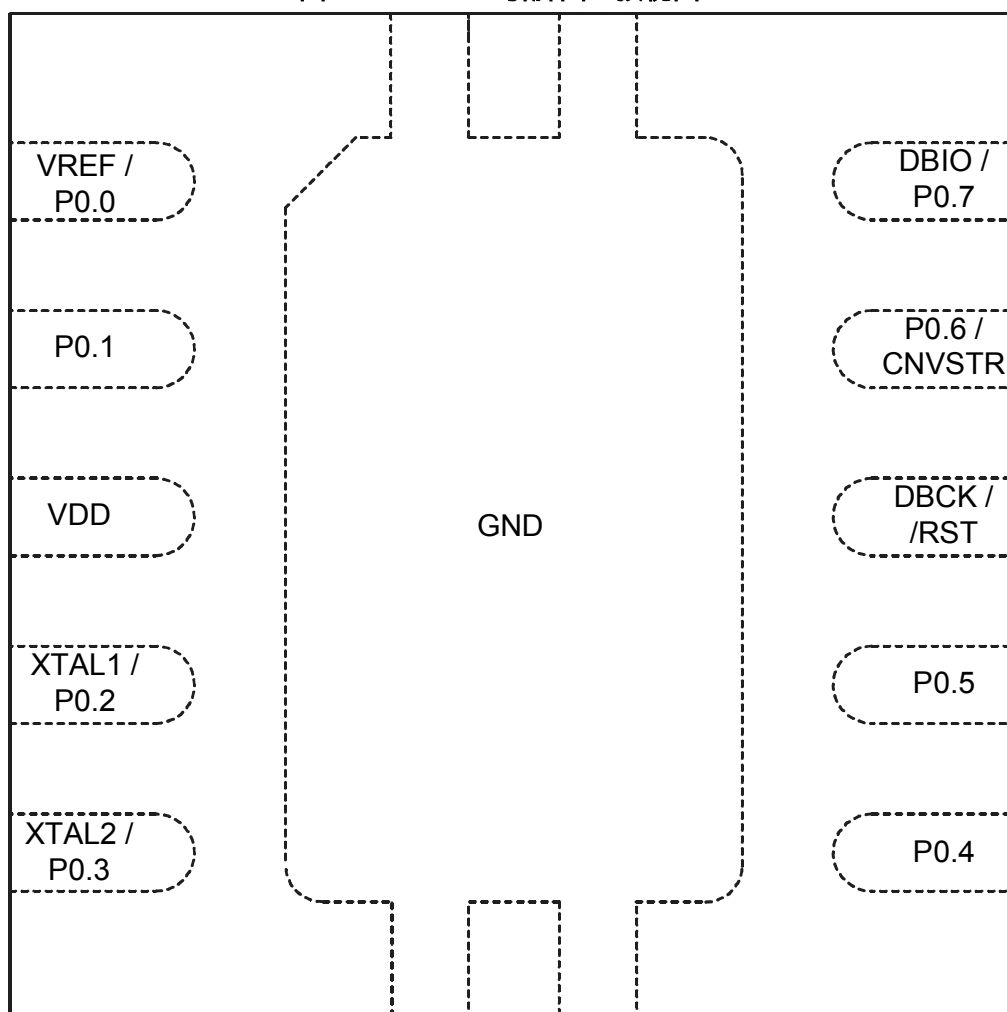
## 4. 引脚和封装定义

表 4.1 引脚定义

引脚号	引脚名称	类型	说明
1	VREF/ P0.0	模拟输入 数字 I/O 或 模拟输入	外部电压基准输入 端口 0 位 0
2	P0.1	数字 I/O 或 模拟输入	端口 0 位 1
3	VDD		电源
4	XTAL1/ P0.2	模拟输入 数字 I/O 或 模拟输入	晶体输入。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。 端口 0 位 2
5	XTAL2/ P0.3	模拟输出/ 输入 数字 I/O	晶体输出。该引脚是晶体或陶瓷谐振器的激励驱动器。对于 CMOS、电容或 RC 网络配置，该引脚是外部时钟输入。 端口 0 位 3
6	P0.4	数字 I/O 或 模拟输入	端口 0 位 4
7	P0.5	数字 I/O 或 模拟输入	端口 0 位 5
8	C2CK/ /RST	数字 I/O 数字 I/O	C2 开发接口的时钟输入。 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 10 $\mu$ s）来启动一次系统复位。
9	P0.6/ CNVSTR	数字 I/O 或 模拟输入 数字 I/O	端口 0 位 6 ADC 外部转换启动输入。
10	C2D/ P0.7	数字 I/O 数字 I/O 或 模拟输入	C2 开发接口的数据信号。 端口 0 位 7
11	GND		地。

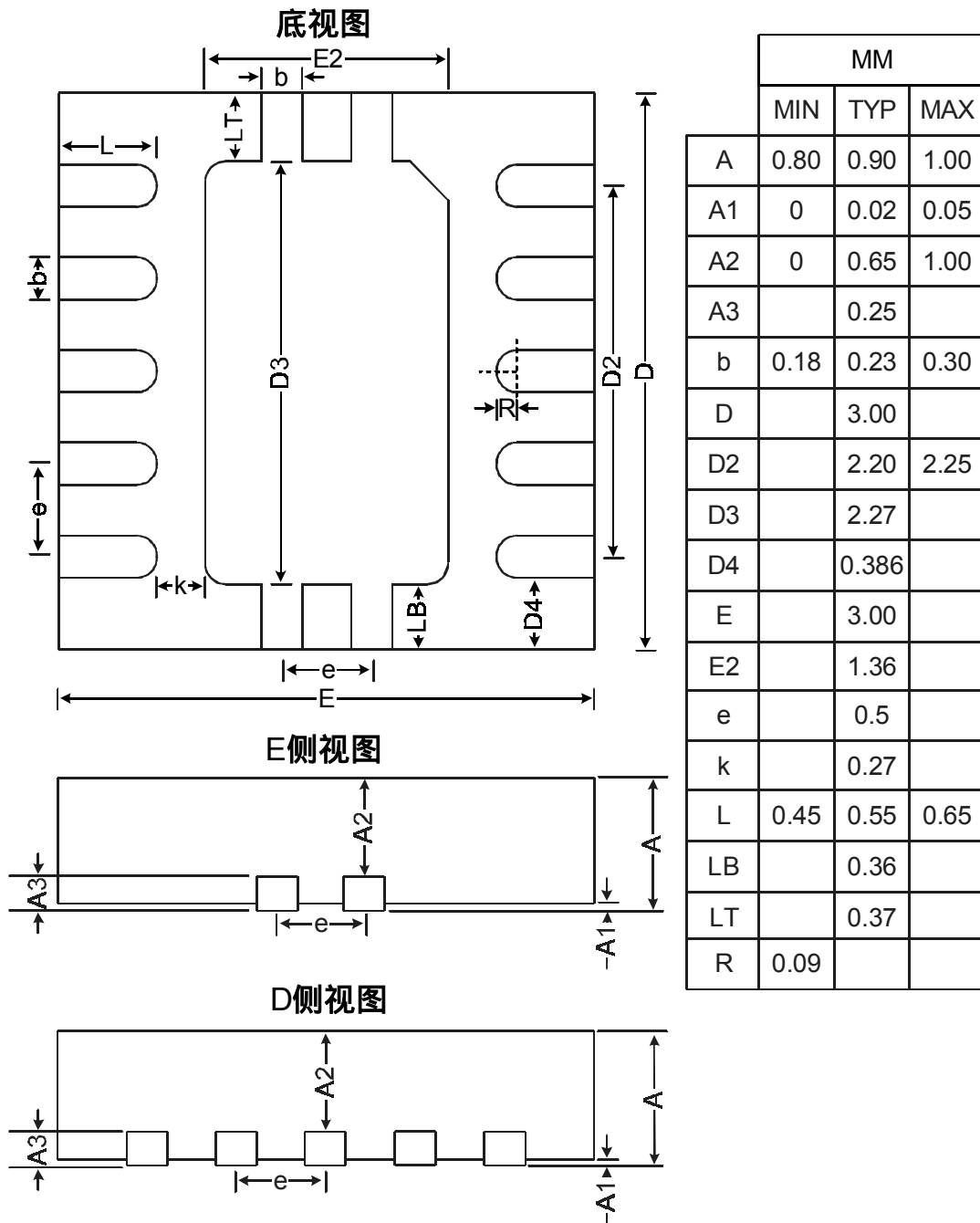
# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 4.1 MLP-11 引脚图 (顶视图)



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 4.2 MLP-11 封装图

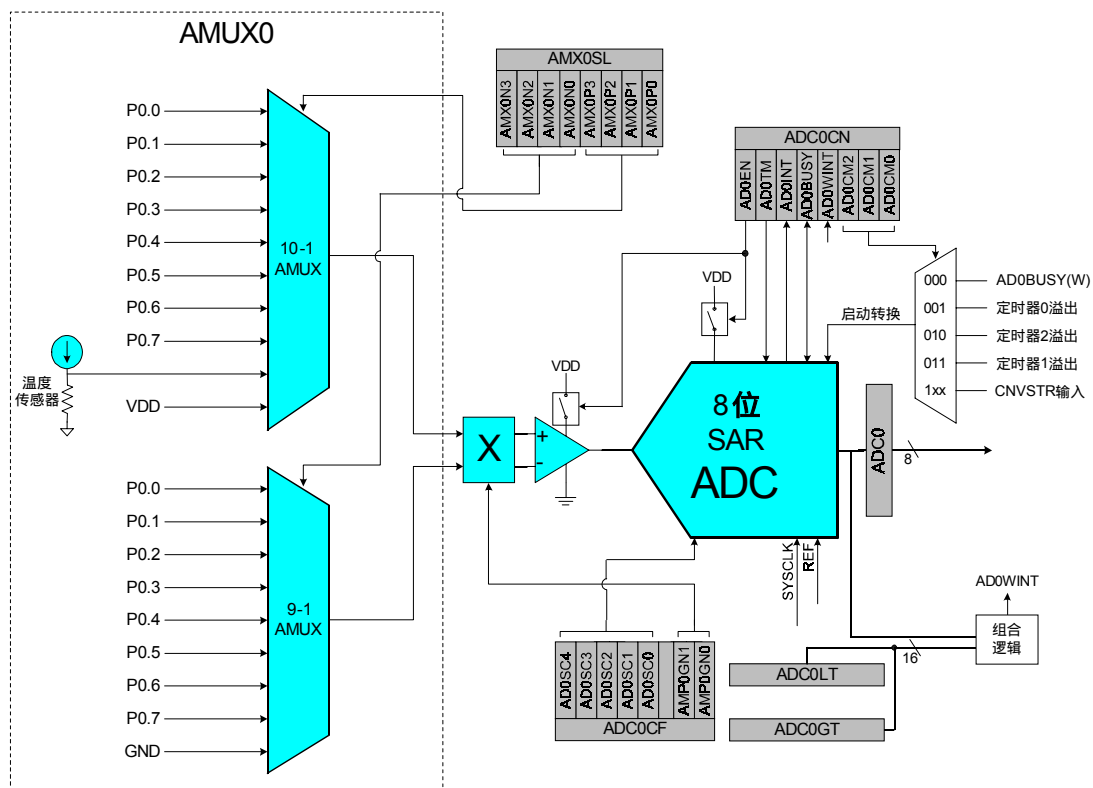


# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 5. ADC0（8 位，只限于 C8051F300/2）

C8051F300/2 的 ADC 子系统集成了两个可编程模拟多路选择器（合称 AMUX0）、一个差分可编程增益放大器（PGA）和一个 500ksps 的 8 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（见框图 5.1）。AMUX0、PGA、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。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任何一个端口引脚、温度传感器输出或相对于任何一个引脚或 GND 的 VDD。只有当 ADC 控制寄存器（ADC0CN）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



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 5.1 模拟多路选择器和 PGA

模拟多路选择器 (AMUX0) 选择去 PGA 的正输入和负输入, 允许测量任意两个端口引脚之间的电压或任何一个引脚与 GND 之间的电压。此外, 片内温度传感器输出或正电源 (VDD) 可以被选择为 PGA 的正输入。**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 AMX0SL (图 5.6) 选择。

转换码的格式在单端方式和差分方式下是不同的。当工作在单端方式时(负输入为 GND), 转化码为 8 位无符号整数, 所测量的输入范围为  $0 - VREF * 255 / 256$ 。下面是单端方式转换码的例子。

输入电压	ADC0 输出 (转换码)
$VREF * 255 / 256$	0xFF
$VREF * 128 / 256$	0x80
$VREF * 64 / 256$	0x40
0	0x00

当工作在差分方式时 (负输入不是 GND), 转化码为 8 位有符号整数 (2 的补码), 所测量的输入范围为  $-VREF - VREF * 127 / 128$ 。下面是差分方式转换码的例子。

输入电压	ADC0 输出 (转换码)
$VREF * 127 / 128$	0x7F
$VREF * 64 / 128$	0x40
0	0x00
$-VREF * 64 / 128$	0xC0
$-VREF$	0x80

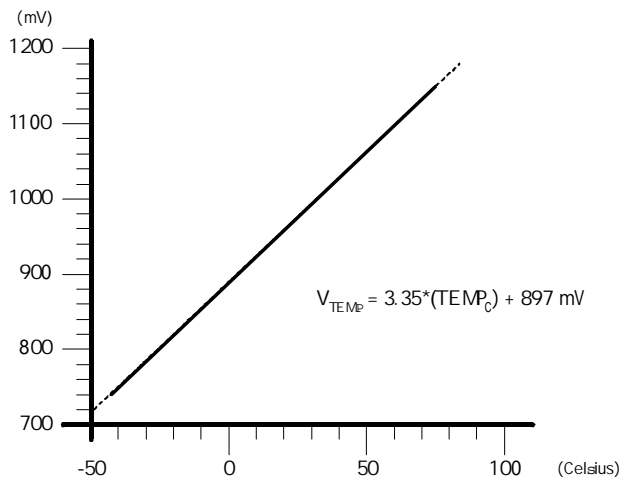
需要特别注意的是, 被选择为 ADC0 输入的引脚应被配置为模拟输入, 并且应被数字交叉开关跳过。要将一个端口引脚配置为模拟输入, 应对 P0MDIN 寄存器中的相应位置 0。为了使交叉开关跳过一个端口引脚, 应将寄存器 XBR0 中的相应位置 1。

PGA 对 AMUX0 的输出信号放大, 放大倍数由 ADC0 配置寄存器中的 AMP0GN1-0 位定义 (见图 5.7)。PGA 的增益可以用软件编程为 0.5、1、2、4, 复位后的增益缺省值为 0.5。

## 5.1 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 AMX0SL 中的 AMX0P2-0 位选中时，输出电压 ( $V_{TEMP}$ ) 为 PGA 的正输入；该电压由 PGA 根据用户编程的放大倍数进行放大。

图 5.2 温度传感器典型传输函数



未经校准的温度传感器输出的线性度非常好，适合于相对温度测量（见表 5.1 的线性度指标）。对于绝对温度测量，建议对增益和/或偏移进行校准。典型的一点校准包括下面几个步骤：

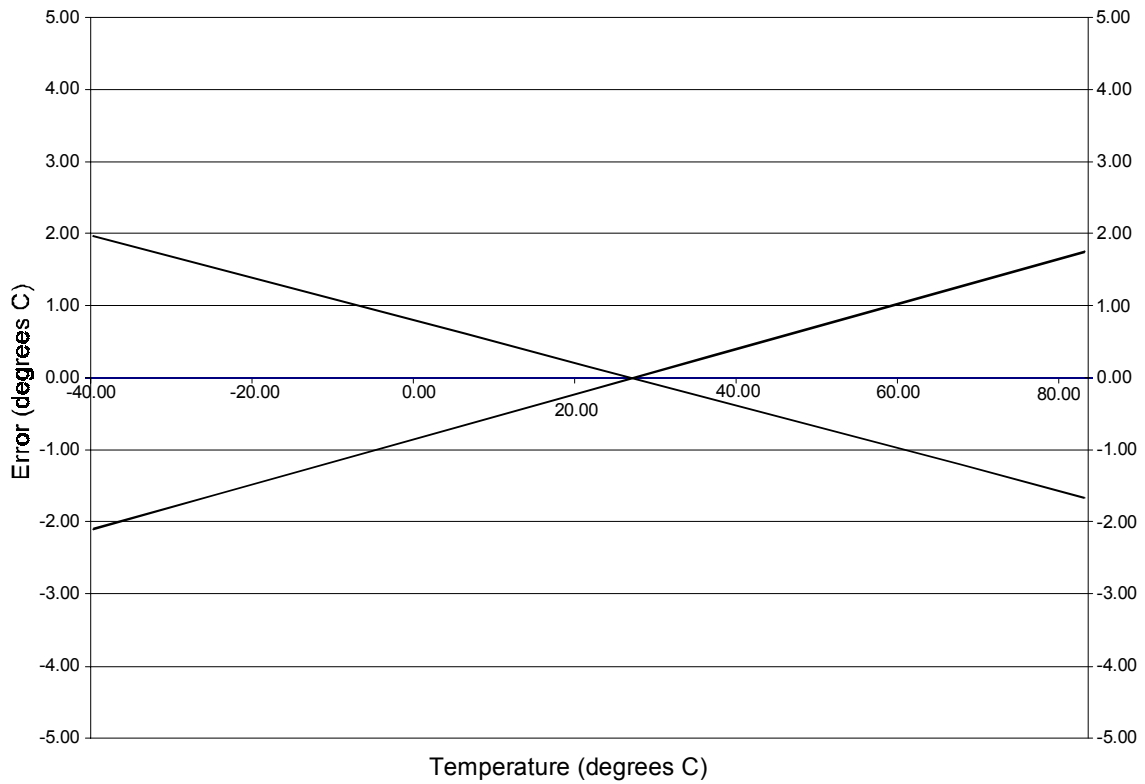
1. 控制/测量环境温度（该温度必须是已知的）。
2. 给器件上电，等待几秒钟使器件自热。
3. 将温度传感器选择为正输入，将 GND 选择为负输入，进行一次 AD 转换。
4. 计算偏移和/或增益，并将这些计算值存入到非易失性存储器中，以备以后使用温度传感器进行温度测量时使用。

图 5.3 示出了典型的温度传感器误差曲线（假设在 25°C 进行一点校准）。**注意：影响 ADC 测量精度的参数，尤其是基准电压值，也同样会影响温度测量精度。**



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 5.3 一点校准的温度传感器误差 (VREF = 2.40 V)



## 5.3 工作方式

ADC0 的最高转换速度为 500ksp/s。ADC0 的转换时钟由系统时钟分频得到，分频数由 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 5 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。

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：应使用 ADC0 中断标志来查询 ADC 转换是否完成。当 AD0INT 位为逻辑 ‘1’ 时，ADC0 数据寄存器中的转换结果有效。注意：当转换源是定时器 2 溢出时，如果定时器 2 工作在 8 位方式，使用定时器 2 的低字节溢出；如果定时器 2 工作在 16 位方式，则使用定时器 2 的高字节溢出。有关定时器配置方面的信息见 “15 定时器”。

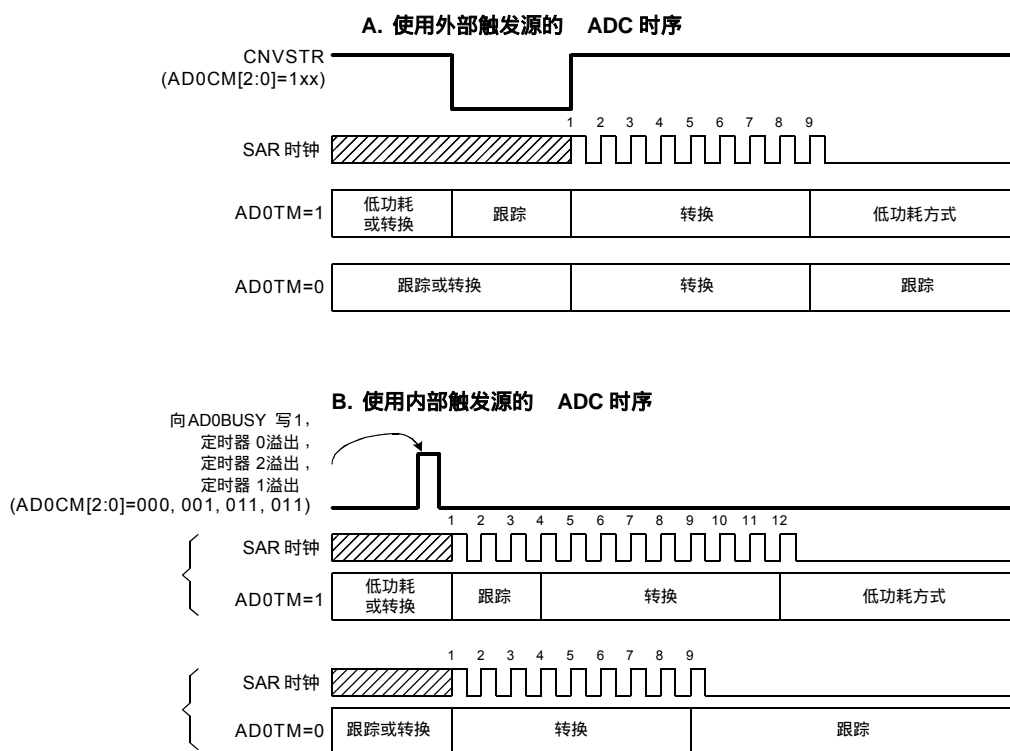
需要注意的是，CNVSTR 输入引脚还是端口引脚 P0.6。当使用 CNVSTR 输入作为转换启动源时，应将寄存器 XBR0 中的位 6 置 ‘1’，以使交叉开关跳过 P0.6。有关端口 I/O 配置的详细信息，见 “12 端口输入/输出”。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 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 5.4）。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 或 PGA 的设置经常变化时也是很有用的，因为 ADC 有建立时间要求（见“5.3.3 建立时间要求”）。

图 5.4 8 位 ADC 跟踪和转换时序示例



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 5.3.3 建立时间要求

当 ADC 输入配置发生改变时（即 AMUX0 或 PGA 的选择发生变化），在进行一次精确的转换之前需要有一个最小的跟踪时间。该跟踪时间由 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要求。

图 5.5 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

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

其中：

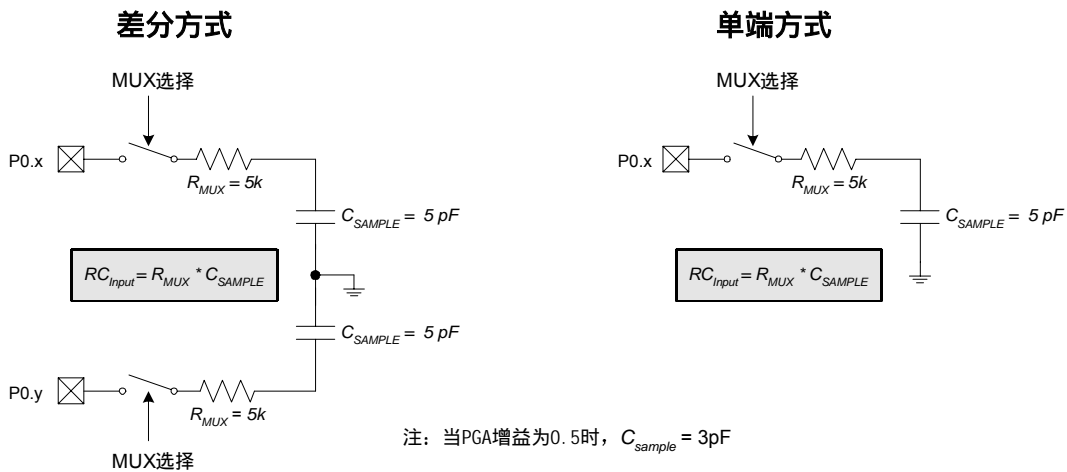
SA 是建立精度，用一个 LSB 的分数表示（例如，建立精度 0.25 对应 1/4 LSB）；

t 为所需要的建立时间，以秒为单位；

$R_{TOTAL}$  为 AMUX0 电阻与外部信号源电阻之和；

n 为 ADC0 的分辨率，用比特表示（8）。

图 5.5 ADC0 等效输入电路



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 5.6 AMUX0SL: AMUX 通道选择寄存器 (C8051F300/2)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AMX0N3	AMX0N2	AMX0N1	AMX0N0	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 7-4: AMX0N3-0: AMUX0 负输入选择  
 注意: 当 GND 被选择为负输入时, ADC0 工作在单端方式; 其它情况下 ADC0 工作在差分方式。  
 0000-1000b: 根据下表选择 ADC0 的负输入。

AMX0N3-0	ADC0 负输入
0000	P0.0
0001	P0.1
0010	P0.2
0011	P0.3
0100	P0.4
0101	P0.5
0110	P0.6
0111	P0.7
1xxx	GND (ADC 工作在单端方式)

位 3-0: AMX0P3-0: AMUX0 正输入选择  
 0000-1001b: 根据下表选择 ADC0 的正输入。  
 1010-1111b: 保留

AMX0P3-0	ADC0 正输入
0000	P0.0
0001	P0.1
0010	P0.2
0011	P0.3
0100	P0.4
0101	P0.5
0110	P0.6
0111	P0.7
1000	温度传感器
1001	VDD

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 5.7 ADC0CF: ADC0 配置寄存器 (C8051F300/2)

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

位 7-3: AD0SC4-0: ADC0 SAR 转换时钟周期控制位  
SAR 转换时钟来源于系统时钟, 由下面的方程给出, 其中  $AD0SC$  表示 AD0SC4-0 中保存的 5 位数值,  $CLK_{SAR}$  表示所需要的 ADC0 SAR 转换时钟。

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

位 2 未用。读 = 0b; 写 = 忽略

位 1-0: AMP0GN1-0: ADC0 内部放大器增益 (PGA)

00: 增益 = 0.5  
01: 增益 = 1  
10: 增益 = 2  
11: 增益 = 4

图 5.8 ADC0: ADC0 数据字寄存器 (C8051F300/2)

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: 0xBF

位 7-0: ADC0 数据字。

ADC0 保持最后一次 ADC0 转换的数据字节。在单端方式, ADC0 中的数据字为 8 位无符号整数; 在差分方式, ADC0 中的数据字为 8 位有符号整数 (2 的补码)。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 5.9 ADC0CN: ADC0 控制寄存器 (C8051F300/2)

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 允许位  
 0: ADC0 禁止。ADC0 处于低耗停机状态。  
 1: ADC0 允许。ADC0 处于活动状态, 并准备转换数据。

位 6: AD0TM: ADC0 跟踪方式位  
 0: 正常方式: 当 ADC0 被允许时, 除了转换期间之外一直处于跟踪方式。  
 1: 低功耗跟踪方式: 由 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。  
 1xx: 外部 CNVSTR 输入信号的上升沿启动 ADC0 转换。  
 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。  
 011: 定时器 2 溢出启动跟踪, 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。  
 1xx: 只在 CNVSTR 输入为逻辑低电平时 ADC0 跟踪, 在 CNVSTR 的上升沿开始转换。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

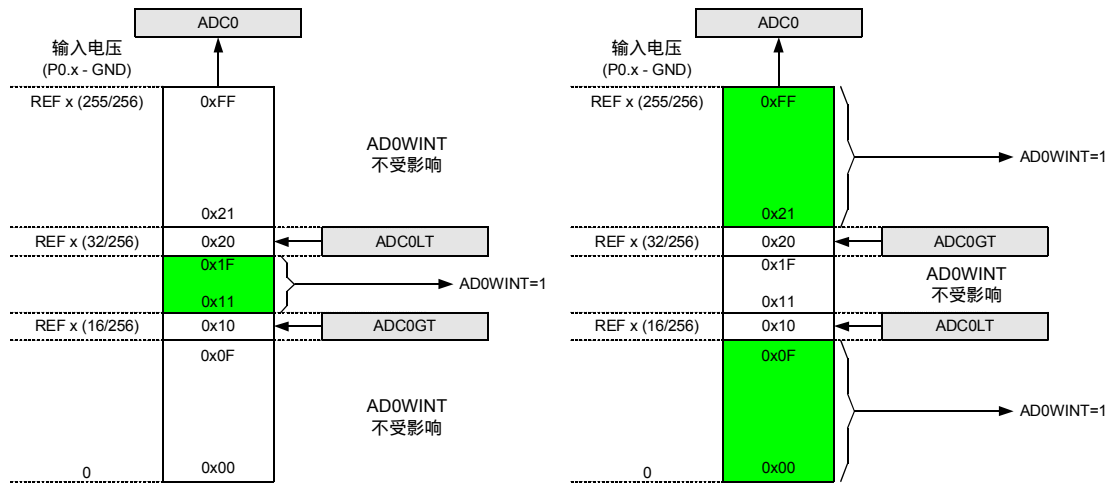
## 5.4 可编程窗口检测器

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（ADC0GT）和 ADC0 上限（小于）寄存器（ADC0LT）中保持比较值。图 5.10 和图 5.11 分别给出了单端和差分方式下窗口比较的例子。注意，窗口检测器标志既可以在测量数据位于用户编程的极限值以内时有效，也可以在测量数据位于用户编程的极限值以外时有效，这取决于 ADC0GT 和 ADC0LT 寄存器的编程值。

### 5.4.1 单端方式下的窗口检测器

图 5.10 给出了单端方式下窗口比较的两个例子。左边的例子所使用的极限值为：ADC0LT = 0x20 和 ADC0GT = 0x10；右边的例子所使用的极限值为：ADC0LT = 0x10 和 ADC0GT = 0x20。注意：在单端方式，转换码是 8 位无符号整数，对应的电压范围为 0 ~ VREF\*255/256。对于左边的例子，如果 ADC0 转换结果数据字（ADC0）位于由 ADC0GT 和 ADC0LT 定义的范围之内（即  $0x10 < ADC0 < 0x20$ ），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（ADC0）位于由 ADC0GT 和 ADC0LT 定义的范围之外（即  $ADC0 < 0x10$  或  $ADC0 > 0x20$ ），则会产生一个 AD0WINT 中断。

图 5.10 ADC 窗口中断示例（单端方式）





# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 5.4.2 差分方式下的窗口检测器

图 5.11 给出了差分方式下窗口比较的两个例子。左边的例子所使用的极限值为：ADC0LT = 0x10 (+16d) 和 ADC0GT = 0xFF (-1d)；右边的例子所使用的极限值为：ADC0LT = 0xFF (-1d) 和 ADC0GT = 0x10 (+16d)。注意：在差分方式，转换码是 8 位有符号整数（2 的补码），对应的电压范围为 -VREF ~ VREF\*127/128。对于左边的例子，如果 ADC0 转换结果数据字（ADC0）位于由 ADC0GT 和 ADC0LT 定义的范围之内（即  $0xFF(-1d) < ADC0 < 0x10(+16d)$ ），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（ADC0）位于由 ADC0GT 和 ADC0LT 定义的范围之外（即  $ADC0 < 0xFF(-1d)$  或  $ADC0 > 0x10(+16d)$ ），则会产生一个 AD0WINT 中断。

图 7.10 ADC 窗口中断示例（差分方式）

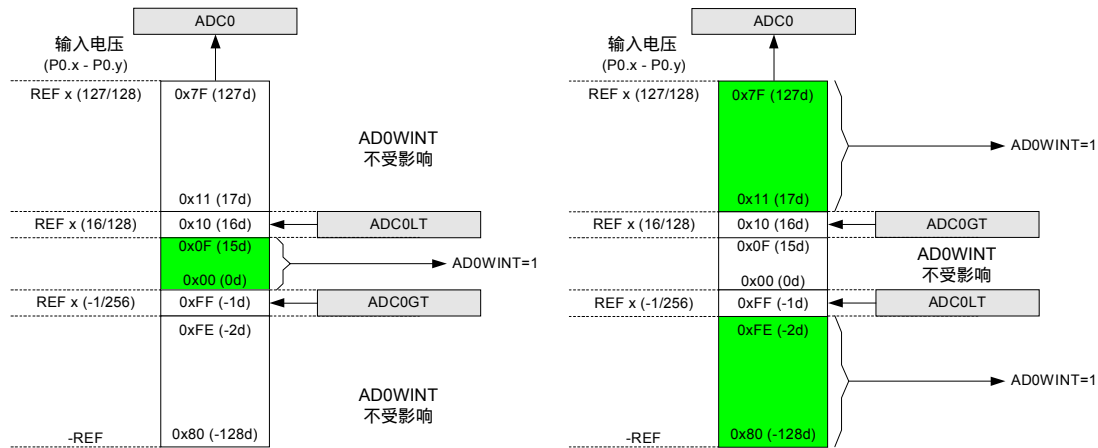


图 5.12 ADC0GT: ADC 下限数据字节寄存器（C8051F300/2）

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

图 5.13 ADC0LT: ADC 上限数据字节寄存器（C8051F300/2）

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

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**表 5.1 ADC0 电气特性**

VDD=3.0V, VREF=2.40V(REFSL=0), PGA 增益=1, -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
<b>直流精度</b>					
分辨率		8			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	注 1		0.5± 0.6		LSB
满度误差	差分方式; 见注 1		-1±0.5		LSB
偏移温度系数			TBD		ppm/°C
<b>动态性能 (10kHz 正弦波差分输入, 满度值的 0 到-1dB, 500ksps)</b>					
信噪比加失真		45	48		dB
总谐波失真	到 5 次谐波		-56		dB
无失真动态范围			58		dB
<b>转换速率</b>					
SAR 转换时钟				6	MHz
转换时间 (SAR 时钟数)		8			时钟
跟踪/保持捕获时间		300			ns
转换速率				500	ksps
<b>模拟输入</b>					
输入电压范围		0		VREF	V
输入电容			10		pF
<b>温度传感器</b>					
线性度	注 1, 2, 3		±0.5		°C
增益	注 1, 2, 3		3350 ±110		μV/°C
偏移	注 1, 2, 3 (温度=0°C)		897±31		mV
<b>电源指标</b>					
电源电流 (VDD 给 ADC0 供电)	工作方式, 500ksps		400	900	μA
电源抑制比			±0.3		mV/V

注 1: 代表偏离平均值的一个标准偏差。

注 2: 在 PGA 增益=2 时的测量值。

注 3: 包括 ADC 偏移、增益和线性度变化引起的误差。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

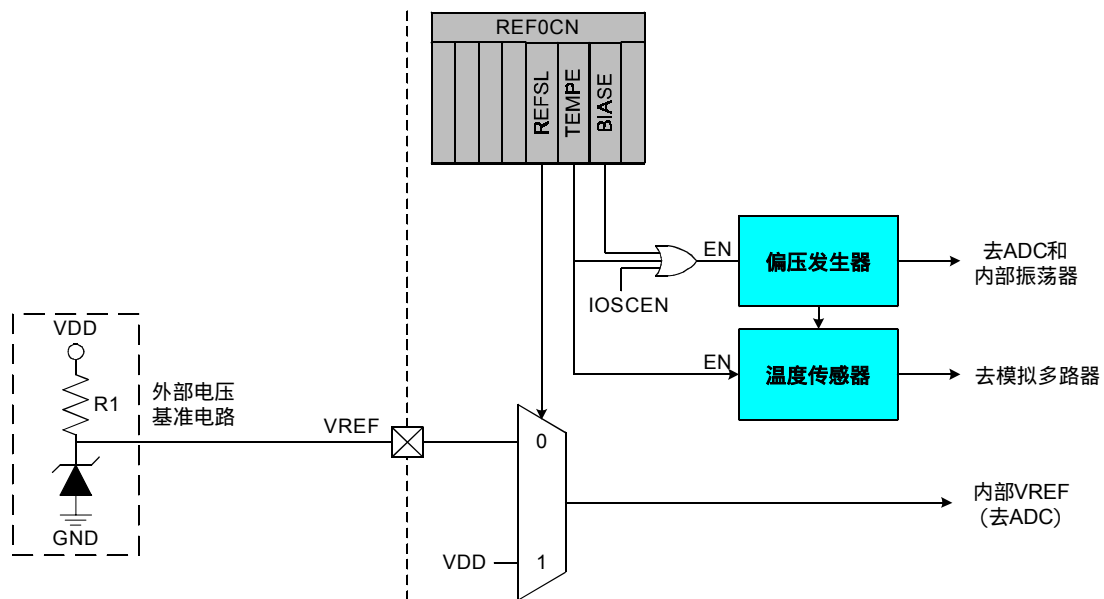
## 6. 电压基准

C8051F300/2 的电压基准 MUX 可以被配置为连接到外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 位控制内部偏置电压发生器。ADC、温度传感器和内部振荡器都要使用偏置电压发生器提供的偏置电压。当这些外设中的任何一个被允许时，BIASE 位被自动置‘1’，也可以通过向 REF0CN 中的 BIASE 位写‘1’来允许偏置电压发生器，见图 6.2 对 REF0CN 寄存器的详细说明。表 6.1 给出了电压基准电路的电气特性。

C8051F300/2 的内部温度传感器接到 ADC0 的正输入多路选择器（见“5.1 模拟多路选择器和 PGA”）。REF0CN 中的 TEMPE 位用于允许/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测量结果都是无意义的。

图 6.1 电压基准功能框图



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 6.2 REF0CN: 电压基准控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	REFSL	TEMPE	BIASE	-	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xD1
<p>位 7-4: 未用。读 = 0000b, 写 = 忽略。</p> <p>位 3: REFSL: 电压基准选择 0: VREF 引脚输入作为电压基准。 1: VDD 作为电压基准。</p> <p>位 2: TEMPE: 温度传感器允许位 0: 内部温度传感器关闭。 1: 内部温度传感器工作。</p> <p>位 1: BIASE: 内部模拟偏压发生器允许位 0: 内部偏压发生器关闭。 1: 内部偏压发生器工作 (使用 ADC 时该位必须为 '1')。</p> <p>位 0: 未用。读 = 0b, 写 = 忽略。</p>								

表 6.1 外部电压基准电路的电气特性

VDD = 3.0V, -40°C到+85°C (除非另有说明)

参 数	条 件	最小值	典型值	最大值	单 位
输入电压范围		0		VDD	V
输入电流	采样速率 = 500ksps; VREF = 3.0V		12		μA

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

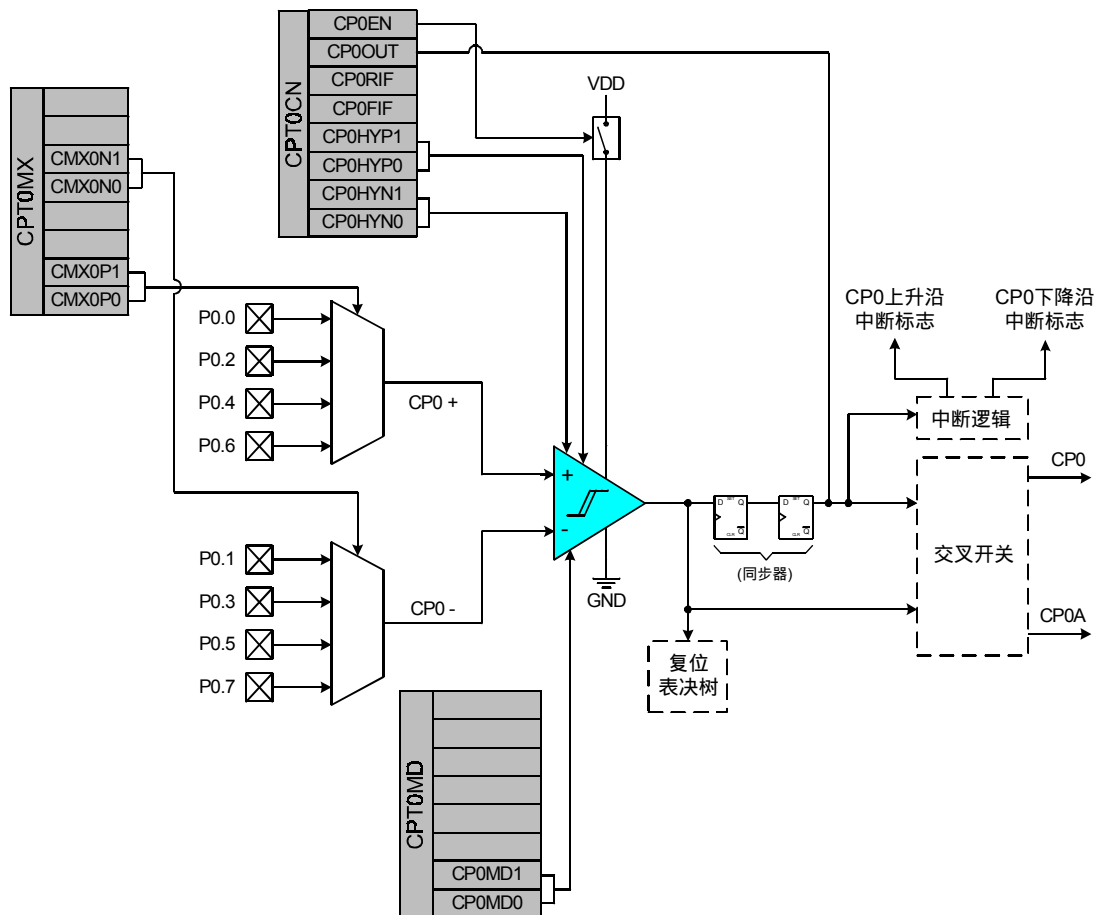
## 7. 比较器

C8051F300/1/2/3/4/5 器件内部有一个可编程电压比较器（即比较器 0），如图 7.1 所示。比较器 0 的响应时间和回差电压都是可编程的。比较器 0 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)和一个异步“直接”输出(CP0A)。即使在系统时钟停止时，CP0A 信号仍然可用，这就允许比较器 0 在器件处于停机方式时工作并产生输出。当被分配了端口引脚时，比较器 0 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 9.5 比较器复位）。

比较器 0 的输入用 CPT0MX 寄存器（图 7.4）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。

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“12.3 通用端口 I/O”）。

图 17.1 比较器功能框图

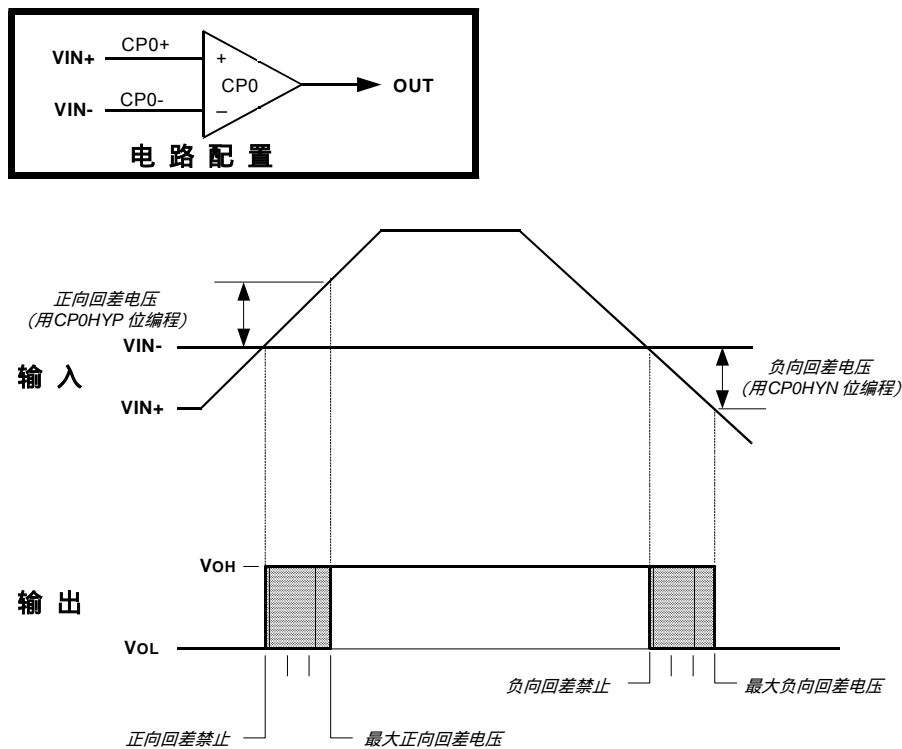


# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

比较器 0 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 0 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 0 输出（如果已通过交叉开关分配到端口 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 0 输出的详细信息见“12.1 优先级交叉开关译码器”。比较器 0 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 0 的电气特性。

可以通过对寄存器 CPT0MD（见图 7.5）中的 CP0MD1-0 位编程来设置比较器 0 的响应时间。选择较长的响应时间可以减小比较器 0 消耗的电流。

图 7.2 比较器回差电压曲线



比较器 0 的回差电压可以通过比较器 0 控制寄存器（CPT0CN）用软件编程。用户既可以对回差电压值（这里指输入电压）编程，也可以对门限电压两侧的正向和负向回差对称度编程。

使用比较器 0 控制寄存器 CPT0CN（见图 7.3）中的位 3-0 对比较器 0 的回差值进行编程。负向回差电压值由 CP0HYN 位的设置决定。如图 7.2 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 CP0HYP 位设置正向回差电压值。

## C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

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’ CP0FIF 中断标志，比较器的上升沿置‘1’ CP0RIF 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 CP0OUT 位得到比较器 0 的输出状态。通过置‘1’ CP0EN 位允许比较器 0 工作，通过将该位清‘0’来禁止比较器 0。

**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: 0xF8
位 7:	CP0EN: 比较器 0 允许位 0: 比较器 0 禁止。 1: 比较器 0 允许。								
位 6:	CP0OUT: 比较器 0 输出状态标志 0: 电压值 CP0+ < CP0-。 1: 电压值 CP0+ > CP0-。								
位 5:	CP0RIF: 比较器 0 上升沿中断标志 0: 自该标志位被清除后，没有发生过比较器 0 上升沿中断。 1: 自该标志位被清除后，发生了比较器 0 上升沿中断。								
位 4:	CP0FIF: 比较器 0 下降沿中断标志 0: 自该标志位被清除后，没有发生过比较器 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。								

# C8051F300/1/2/3/4/5 混合信号 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  
这两位选择作为比较器 0 负输入的端口引脚。

CMX0N1	CMX0N0	比较器负输入
0	0	P0.1
0	1	P0.3
1	0	P0.5
1	1	P0.7

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  
这两位选择作为比较器 0 正输入的端口引脚。

CMX0P1	CMX0P0	比较器正输入
0	0	P0.0
0	1	P0.2
1	0	P0.4
1	1	P0.6

图 10.5 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
-	-	-	-	-	-	CP0MD1	CP0MD0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x9D

位 7-2: 未用。读 = 000000b, 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



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**表 7.1 比较器电气特性**

VDD=3.0V, -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
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	CPnHYP1-0 = 00		0	1	mV
正向回差电压 2	CPnHYP1-0 = 01	3	5	7	mV
正向回差电压 3	CPnHYP1-0 = 10	7	10	15	mV
正向回差电压 4	CPnHYP1-0 = 11	15	20	25	mV
负向回差电压 1	CPnHYN1-0 = 00		0	1	mV
负向回差电压 2	CPnHYN1-0 = 01	3	5	7	mV
负向回差电压 3	CPnHYN1-0 = 10	7	10	15	mV
负向回差电压 4	CPnHYN1-0 = 11	15	20	25	mV
反相或同相输入电压范围		-0.25		VDD+0.25	V
输入电容			7		pF
输入偏置电流		-5	0.001	+5	nA
输入偏移电压		-5		+5	mV
<b>电源</b>					
电源抑制比			0.1	1	mV/V
上电时间			10		$\mu$ S
电源电流 (DC)	方式 0		7.6		$\mu$ A
	方式 1		3.2		$\mu$ A
	方式 2		1.3		$\mu$ A
	方式 3		0.4		$\mu$ A

$\dagger V_{cm}$  是 CP0+和 CP0-上的共模电压。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

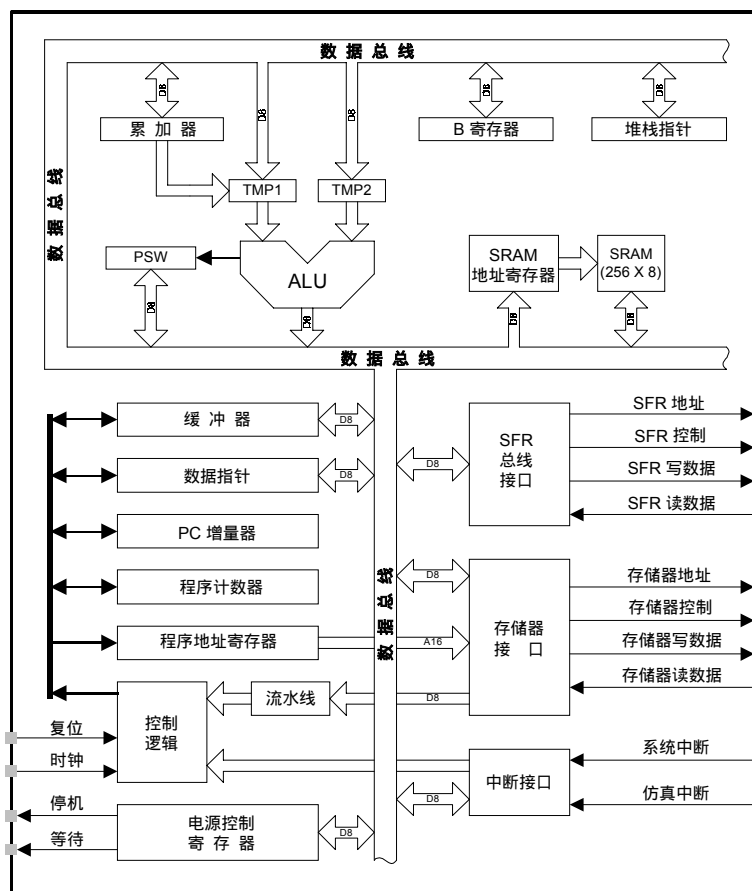
## 8. CIP-51 CPU

MCU 系统控制器的内核是 CIP-51 微控制器。CIP-51 与 MCS-51<sup>TM</sup> 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 3 个 16 位的计数器/定时器（详见第 15 章）、一个增强型全双工 UART（详见第 14 章）、256 字节内部 RAM、128 字节特殊功能寄存器（SFR）地址空间及 8 个端口 I/O（详见第 12 章）。CIP-51 还包含片内调试硬件（详见第 17 章）和与 MCU 直接接口的模拟和数字子系统，在一个集成电路内提供了完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 8.1 的原理框图）。CIP-51 具有下列特点：

- 与 MCS-51 指令集完全兼容
- 在 25MHz 时钟时最大速度为 25MIPS
- 0 到 25MHz 的时钟频率
- 256 字节内部 RAM
- 8 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 8.1 CIP-51 原理框图



# C8051F300/1/2/3/4/5 混合信号 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 2 线开发接口 (C2) 实现的。注意，可以在用户软件中使用 MOV<sub>C</sub> 和 MOV<sub>X</sub> 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（包括单步通过中断服务程序）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“17. C2 接口”。

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

## 8.1 指令集

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

### 8.1.1 指令和 CPU 时序

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

由于 CIP-51 采用了流水线结构，大多数指令执行所需的时钟周期数与指令的字节数一致。条件转移指令在不发生转移时的执行周期数比发生转移时少一个。表 8.1 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。

### 8.1.2 MOV<sub>X</sub> 指令和程序存储器

MOV<sub>X</sub> 指令通常用于访问外部数据存储器（注：C8051F300/1/2/3/4/5 不支持外部数据或程序存储器）。在 CIP-51 中，MOV<sub>X</sub> 指令访问片内 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“10. FLASH 存储器”。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**表 8.1 CIP-51 指令集**

助记符	功能说明	字节数	时钟周期数
<b>算术操作类指令</b>			
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	累加器减去间接寻址 RAM(带借位)	2	2
SUBB A,@Ri	累加器减去间址 RAM(带借位)	1	2
SUBB A,#data	累加器减去立即数(带借位)	2	2
INC A	累加器加 1	1	1
INC Rn	寄存器加 1	1	1
INC direct	直接寻址字节加 1	2	2
INC @Ri	间址 RAM 加 1	1	2
DEC A	累加器减 1	1	1
DEC Rn	寄存器减 1	1	1
DEC direct	直接寻址字节减 1	2	2
DEC @Ri	间址 RAM 减 1	1	2
INC DPTR	数据地址加 1	1	1
MUL AB	累加器与寄存器 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
<b>逻辑操作类指令</b>			
ANL A,Rn	寄存器“与”到累加器	1	1
ANL A,direct	直接寻址字节“与”到累加器	2	2
ANL A,@Ri	间址 RAM “与”到累加器	1	2
ANL A,#data	立即数“与”到累加器	2	2
ANL direct,A	累加器“与”到直接寻址字节	2	2
ANL direct,#data	立即数“与”到直接寻址字节	3	3
ORL A,Rn	寄存器“或”到累加器	1	1
ORL A,direct	直接寻址字节“或”到累加器	2	2
ORL A,@Ri	间址 RAM “或”到累加器	1	2
ORL A,#data	立即数“或”到累加器	2	2
ORL direct,A	累加器“或”到直接寻址字节	2	2
ORL direct,#data	立即数“或”到直接寻址字节	3	3
XRL A,Rn	寄存器“异或”到累加器	1	1
XRL A,direct	直接寻址数“异或”到累加器	2	2
XRL A,@Ri	间址 RAM “异或”到累加器	1	2
XRL A,#data	立即数“异或”到累加器	2	2
XRL direct,A	累加器“异或”到直接寻址字节	2	2
XRL direct,#data	立即数“异或”到直接寻址字节	3	3
CLR A	累加器清零	1	1
CPL A	累加器求反	1	1
RL A	循环循环左移	1	1
RLC A	经过进位位的累加器循环左移	1	1

# C8051F300/1/2/3/4/5 混合信号 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	经过进位位的累加器循环右移	1	1
<b>数据传输类指令</b>			
MOV A,Rn	寄存器传送到累加器 A	1	1
MOV A,direct	直接寻址字节传送到累加器	2	2
MOV A,@Ri	间址 RAM 传送到累加器	1	2
MOV A,#data	立即数传送到累加器	2	2
MOV Rn,A	累加器传送到寄存器	1	1
MOV Rn,direct	直接寻址字节传送到寄存器	2	2
MOV Rn,#data	立即数传送到寄存器	2	2
MOV direct,A	累加器传送到直接寻址字节	2	2
MOV direct,Rn	寄存器传送到直接寻址字节	2	2
MOV direct,direct	直接寻址字节传送到直接寻址字节	3	3
MOV direct,@Ri	间址 RAM 传送到直接寻址字节	2	2
MOV direct,#data	立即数传送到直接寻址字节	3	3
MOV @Ri,A	累加器传送到间址 RAM	1	2
MOV @Ri,direct	直接寻址数传送到间址 RAM	2	2
MOV @Ri,#data	立即数传送到间址 RAM	2	2
MOV DPTR,#data16	16 位常数装入数据指针	3	3
MOVC A,@A+DPTR	代码字节传送到累加器	1	3
MOVC A,@A+PC	代码字节传送到累加器	1	3
MOVX A,@Ri	外部 RAM(8 位地址)数传送到 A	1	3
MOVX @Ri,A	累加器传到外部 RAM (8 位地址)	1	3
MOVX A,@DPTR	外部 RAM(16 位地址)传送到 A	1	3
MOVX @DPTR,A	累加器传到外部 RAM (16 位地址)	1	3
PUSH direct	直接寻址字节压入栈顶	2	2
POP direct	栈顶数据弹出到直接寻址字节	2	2
XCH A,Rn	寄存器和累加器交换	1	1
XCH A,direct	直接寻址字节与累加器交换	2	2
XCH A,@Ri	间址 RAM 与累加器交换	1	2
XCHD A,@Ri	间址 RAM 和累加器交换低半字节	1	2
SWAP A	累加器内高低半字节交换	1	1
<b>位操作类指令</b>			
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

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

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

### 寄存器、操作数和寻址方式说明:

Rn – 当前选择的寄存器区的寄存器 R0-R7。

@Ri – 通过寄存器 R0-R1 间接寻址的数据 RAM 地址。

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

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

#data – 8 位立即数

#data16 – 16 位立即数

bit – 数据 RAM 或 SFR 中的直接寻址位

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

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

有一个未使用的操作码（0xA5），它执行与 NOP 指令相同的功能。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

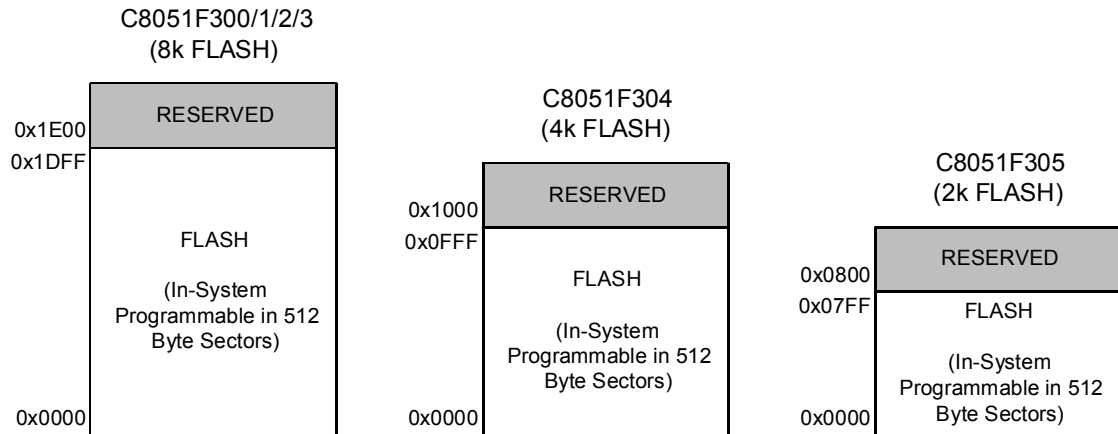
## 8.2 存储器组织

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 8.2 和图 8.3 所示。

### 8.2.1 程序存储器

CIP-51 有 64K 字节的程序存储器空间。C8051F300/1/2/3 在这个程序存储器空间中实现了 8192 字节的可在系统编程的 FLASH 存储器，组织在一个连续的存储块内 (0x0000 – 0x1FFF)。注意：该 FLASH 存储器中有 512 字节 (0x1E00 – 0x1FFF) 保留给工厂使用，不能用于存储用户程序。C8051F304 实现了 4096 字节的可在系统编程的 FLASH 存储器，C8051F305 实现了 2048 字节的可在系统编程的 FLASH 存储器。图 8.2 给出了 C8051F300/1/2/3/4/5 器件的程序存储器结构。

图 8.2 程序存储器结构图



程序存储器通常被认为是只读的。但是 CIP-51 可以通过设置程序存储写允许位 (PSCTL.0) 用 MOVX 指令对程序存储器写入。这一特性为 CIP-51 提供了更新程序代码和将程序存储器空间用于非易失性数据存储的机制。更进一步的详细信息见“10. FLASH 存储器”。

### 8.2.2 数据存储器

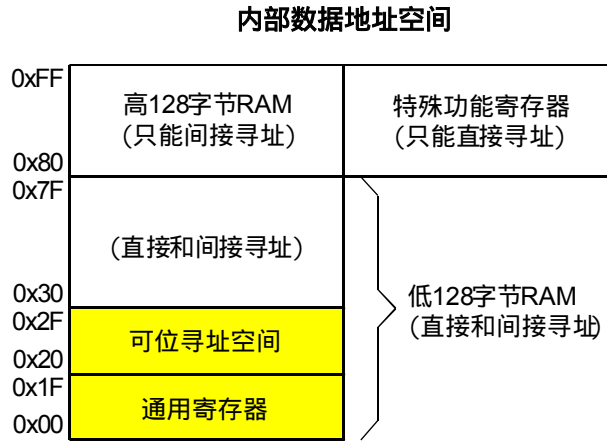
CIP-51 的数据存储器空间中有 256 字节的内部 RAM，位于地址 0x00 到 0xFF 的地址空间。数据存储器中的低 128 字节用于通用寄存器和临时存储器。可以用直接或间接寻址方式访问数据存储器中的低 128 字节。从 0x00 到 0x1F 为 4 个通用寄存器区，每个区有 8 个寄存器。接下来的 16 字节，从地址 0x20 到 0x2F，既可以按字节寻址又可以作为 128 个位地址用直接位寻址方式访问。

数据存储器中的高 128 字节只能用间接寻址访问。该存储区与特殊功能寄存器 (SFR) 占据相同的地址空间，但物理上与 SFR 空间是分开的。当寻址高于 0x7F 的地址时，指令所用的寻址方式决定了 CPU 是访问数据存储器的高 128 字节还是访问 SFR。使用直接寻址方式的指令将访问 SFR 空间，间接寻址高于 0x7F 地址的指令将访问数据存储器的高 128 字节。图 8.3

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

给出了 CIP-51 数据存储器的组织示意图。

图 8.3 数据存储器结构图



## 8.2.3 通用寄存器

数据存储器的低 32 字节，从地址 0x00 到 0x1F，可以作为 4 个通用寄存器区访问。每个区有 8 个寄存器，称为 R0 - R7。在某一时刻只能选择一个寄存器区。程序状态字中的 RS0 (PSW.3) 和 RS1 (PSW.4) 位用于选择当前的寄存器区 (见图 8.7 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 作为间址寄存器。

## 8.2.4 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-51™ 汇编语言允许用 XX.B 的形式替代位地址，XX 为字节地址，B 为寻址位在字节中的位置。例如，指令：

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

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

## 8.2.5 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。



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 8.2.6 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<sup>TM</sup> 指令集兼容的前提下增加新的功能。表 18.3 列出了 CIP-51 系统控制器中的全部 SFR。

任何时刻用直接寻址访问从 0x80 到 0xFF 的存储器空间将访问特殊功能寄存器 (SFR)。地址以 0x0 或 0x8 结尾的 SFR (例如 P0、TCON、P1、SCON、IE 等) 既可以字节寻址也可以位寻址，所有其它 SFR 只能字节寻址。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 8.3 中已标明)。

**表 8.2 特殊功能寄存器存储器映象**

F8	CPT0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0			
F0	B	P0MDIN					EIP1	
E8	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2			RSTSRC
E0	ACC	XBR0	XBR1	XBR2	IT01CF		EIE1	
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2			
D0	PSW	REF0CN						
C8	TMR2CN		TMR2RLL	TMR2RLH	TMR2L	TMR2H		
C0	SMB0CN	SMB0CF	SMB0DAT		ADC0GT		ADC0LT	
B8	IP			AMX0SL	ADC0CF		ADC0	
B0		OSCXCN	OSCICN	OSCICL			FLSCL	FLKEY
A8	IE							
A0					P0MDOUT			
98	SCON0	SBUF0				CPT0MD		CPT0MX
90								
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)

**表 8.3 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	64
ADC0CF	0xBC	ADC0 配置寄存器	37
ADC0CN	0xE8	ADC0 控制寄存器	38
ADC0GT	0xC4	ADC0 下限 (大于) 比较字	41
ADC0LT	0xC6	ADC0 上限 (小于) 比较字	41
ADC0	0xBE	ADC0 数据字	37
AMX0SL	0xBB	ADC0 模拟通道选择寄存器	36
B	0xF0	B 寄存器	64
CKCON	0x8E	时钟控制寄存器	137

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**表 8.3 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。

CPT0CN	0xF8	比较器 0 控制寄存器	47
CPT0MD	0x9D	比较器 0 方式选择寄存器	49
CPT0MX	0x9F	比较器 0 MUX 选择寄存器	48
DPH	0x83	数据指针高字节	62
DPL	0x82	数据指针低字节	62
EIE1	0xE6	扩展中断允许寄存器 1	70
EIP1	0xF6	扩展中断优先级寄存器 1	71
FLKEY	0xB7	FLASH 锁定和关键码寄存器	85
FLSCL	0xB6	FLASH 存储器时序预分频器	85
IE	0xA8	中断允许寄存器	68
IP	0xB8	中断优先级寄存器	69
IT01CF	0xE4	INT0/INT1 配置寄存器	72
OSICL	0xB3	内部振荡器校准寄存器	89
OSICN	0xB2	内部振荡器控制寄存器	89
OSCXCN	0xB1	外部振荡器控制寄存器	91
P0	0x80	端口 0 锁存器	99
P0MDIN	0xF1	端口 0 输入方式配置寄存器	99
P0MDOUT	0xA4	端口 0 输出方式配置寄存器	100
PCA0CN	0xD8	PCA 控制寄存器	154
PCA0MD	0xD9	PCA 方式寄存器	155
PCA0CPH0	0xFC	PCA 捕捉模块 0 高字节	158
PCA0CPH1	0xEA	PCA 捕捉模块 1 高字节	158
PCA0CPH2	0xEC	PCA 捕捉模块 2 高字节	158
PCA0CPL0	0xFB	PCA 捕捉模块 0 低字节	158
PCA0CPL1	0xE9	PCA 捕捉模块 1 低字节	158
PCA0CPL2	0xEB	PCA 捕捉模块 2 低字节	158
PCA0CPM0	0xDA	PCA 模块 0 方式寄存器	156
PCA0CPM1	0xDB	PCA 模块 1 方式寄存器	156
PCA0CPM2	0xDC	PCA 模块 2 方式寄存器	156
PCA0H	0xFA	PCA 计数器高字节	157
PCA0L	0xF9	PCA 计数器低字节	157
PCON	0x87	电源控制寄存器	74
PSCTL	0x8F	程序存储读写控制寄存器	84
PSW	0xD0	程序状态字	63
REF0CN	0xD1	电压基准控制寄存器	43
RSTSRC	0xEF	复位源寄存器	79
SBUF0	0x99	UART0 数据缓冲器	127
SCON0	0x98	UART0 控制寄存器	126
SMB0CF	0xC1	SMBus 配置寄存器	108
SMB0CN	0xC0	SMBus 控制寄存器	110
SMB0DAT	0xC2	SMBus 数据寄存器	112
SP	0x81	堆栈指针	63

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**表 8.3 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
TMR2CN	0xC8	计数器/定时器 2 控制寄存器	141
TCON	0x88	计数器/定时器控制寄存器	135
TH0	0x8C	计数器/定时器 0 高字节	138
TH1	0x8D	计数器/定时器 1 高字节	138
TL0	0x8A	计数器/定时器 0 低字节	138
TL1	0x8B	计数器/定时器 1 低字节	138
TMOD	0x89	计数器/定时器方式寄存器	136
TMR2RLH	0xCB	计数器/定时器 2 重载值高字节	142
TMR2RLL	0xCA	计数器/定时器 2 重载值低字节	142
TMR2H	0xCD	计数器/定时器 2 高字节	142
TMR2L	0xCC	计数器/定时器 2 低字节	142
XBR0	0xE1	端口 I/O 交叉开关控制 0	97
XBR1	0xE2	端口 I/O 交叉开关控制 1	97
XBR2	0xE3	端口 I/O 交叉开关控制 2	98
0x97, 0xAE, 0xAF, 0xB4, 0xB6, 0xBF, 0xCE, 0xD2, 0xD3, 0xD4, 0xD5, 0xD6, 0xD7, 0xDD, 0xDE, 0xDF, 0xF5,		保留	

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 8.2.7 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 8.4 DPL: 数据指针低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								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。

图 8.5 DPH: 数据指针高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								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。

图 8.6 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。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 8.7 PSW: 程序状态字

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

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

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

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

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

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

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

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

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

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 8.8 ACC: 累加器**

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

位 7-0: ACC: 累加器  
该寄存器为算术操作的累加器。

**图 8.9 B: B 寄存器**

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

位 7-0: B: B 寄存器  
该寄存器为某些算术操作的第二累加器。

## 8.3 中断系统

CIP-51 包含一个扩展的中断系统，支持 12 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 - EIE1) 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。

### 8.3.1 MCU 中断源和中断向量

MCU 支持 12 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 8.4 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。

### 8.3.2 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。IT0ICF 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
0	0	低电平有效，边沿触发	0	0	低电平有效，边沿触发
0	1	高电平有效，边沿触发	0	1	高电平有效，边沿触发
1	0	低电平有效，电平触发	1	0	低电平有效，电平触发
1	1	高电平有效，电平触发	1	1	高电平有效，电平触发

/INT0 和/INT1 所使用的端口引脚在 IT0ICF 寄存器中定义 (见图 8.14)。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“12.1 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

## C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

---

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。

### 8.3.3 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）中都有一个配置其优先级的中断优先级设置位，缺省值为低优先级。如果两个中断同时发生，具有高优先级的中断先得到服务。如果这两个中断的优先级相同，则由固定的优先级顺序决定哪一个中断先得到服务。

### 8.3.4 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。



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

表 8.4 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)
SMBus 接口	0x0033	6	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
ADC0 窗口比较	0x003B	6	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.1)	PWADC0 (EIP1.1)
ADC0 转换结束	0x0043	8	AD0INT (ADC0CN.5)	Y	N	EADC0C (EIE1.2)	PADC0C (EIP1.2)
可编程计数器阵列	0x004B	9	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.3)	PPCA0 (EIP1.3)
比较器 0 下降沿	0x0053	10	CP0FIF (CPT0CN.4)	N	N	ECP0F (EIE1.4)	PCP0F (EIP1.4)
比较器 0 上升沿	0x005B	11	CP0RIF (CPT0CN.5)	N	N	ECP0R (EIE1.5)	PCP0R (EIP1.5)

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 8.3.5 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 8.10 IE: 中断允许寄存器**

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

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

位 6: IEGF0: 通用标志位 0。  
该位用作软件控制的通用标志位。

位 5: ET2: 定时器 2 中断允许位。  
该位用于设置定时器 2 的中断屏蔽。  
0: 禁止定时器 2 中断。  
1: 允许 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

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 8.11 IP: 中断优先级寄存器

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

位 7-6: 未用。读=11b, 写=忽略。

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

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

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

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

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

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

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 8.12 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
-	-	ECP0R	ECP0F	EPCA0	EADC0C	EWADC0	ESMB0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xE6

位 7-6: 未用。读 = 00b, 写 = 忽略。

位 5: ECP0R: 比较器 0 (CP0) 上升沿中断允许位  
该位设置 CP0 上升沿的中断屏蔽。  
0: 禁止 CP0 上升沿中断。  
1: 允许 CP0RIF 标志位的中断请求。

位 4: ECP0F: 比较器 0 (CP0) 下降沿中断允许位  
该位设置 CP0 下降沿的中断屏蔽。  
0: 禁止 CP0 下降沿中断。  
1: 允许 CP0FIF 标志位的中断请求。

位 3: 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 2: 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 1: 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 窗口比较标志的中断请求。

位 0: ESMB0: SMBus 中断允许位  
该位设置 SMBus 的中断屏蔽。  
0: 禁止 SMBus 中断。  
1: 允许 SI 标志位的中断请求。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 8.13 EIP1: 扩展中断优先级 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	PCP0R	PCP0F	PPCA0	PADC0C	PWADC0	PSMB0	11000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xF6

位 7-6: 未用。读 = 11b, 写 = 忽略。

位 5: PCP0R: 比较器 0 (CP0) 上升沿中断优先级控制  
该位设置 CP0 上升沿中断的优先级。  
0: CP0 上升沿中断为低优先级。  
1: CP0 上升沿中断为高优先级。

位 4: PCP0F: 比较器 0 (CP0) 下降沿中断优先级控制  
该位设置 CP0 下降沿中断的优先级。  
0: CP0 下降沿中断为低优先级。  
1: CP0 下降沿中断为高优先级。

位 3: PPCA0: 可编程计数器阵列中断优先级控制  
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 2: PADC0C: 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 1: 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 0: PSMB0: SMBus 中断优先级控制  
该位设置 SMBus 中断的优先级。  
0: SMBus 中断为低优先级。  
1: SMBus 中断为高优先级。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 8.14 INT01CF: 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 输入为低电平有效。  
1: /INT1 输入为高电平有效。

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 XBAR0 中的相应位置 '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 输入为低电平有效。  
1: /INT0 输入为高电平有效。

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 XBAR0 中的相应位置 '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

## 8.4 电源管理方式

CIP-51 有两种可软件编程的电源管理方式：等待和停机。在等待方式，CPU 停止运行，而外设和时钟处于活动状态。在停机方式，CPU 停止运行，所有的中断和定时器（时钟丢失检测器除外）都处于非活动状态，系统时钟停止（模拟外设保持在所选择的状态）。由于在等待方式下时钟仍然运行，所以功耗与进入等待方式之前的系统时钟频率和处于活动状态的外设数目有关。停机方式消耗最少的功率。图 8.15 对用于控制 CIP-51 电源管理方式的电源控制寄存器作出了说明。

虽然 CIP-51 具有等待和停机方式（与任何标准 8051 结构一样），但最好禁止不必要的外设，以使整个 MCU 的功耗最小。每个模拟外设在不用时都可以被禁止，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。

### 8.4.1 等待方式

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

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

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

### 8.4.2 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。在进入停机方式之前，每个模拟外设都可以被单独关断。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。

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

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 8.15 PCON: 电源控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
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 (关闭内部振荡器)

位 0: IDLE: 等待方式选择。  
将该位置 ‘1’ 使 CIP-51 进入等待方式。该位读出值总是为 0。  
1: 进入等待方式。(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。)



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 9. 复位源

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

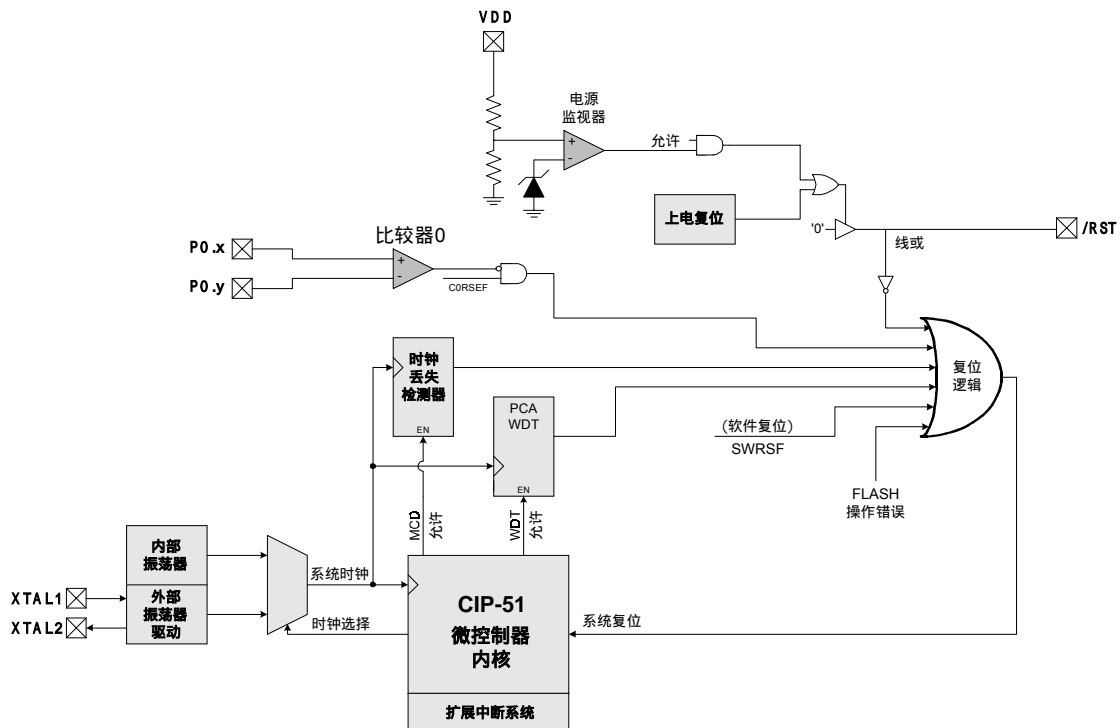
- CIP-51 停止程序执行
- 特殊功能寄存器（SFR）被初始化为所定义的复位值
- 外部端口引脚被置于一个已知状态
- 中断和定时器被禁止。

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

I/O 端口锁存器的复位值为 0xFF（全部为逻辑‘1’），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，直到器件推出复位状态。

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“11. 振荡器”。看门狗定时器被允许，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“16.3 看门狗定时器方式”）。一旦系统时钟源稳定，程序从地址 0x0000 开始执行。

图 9.1 复位源框图

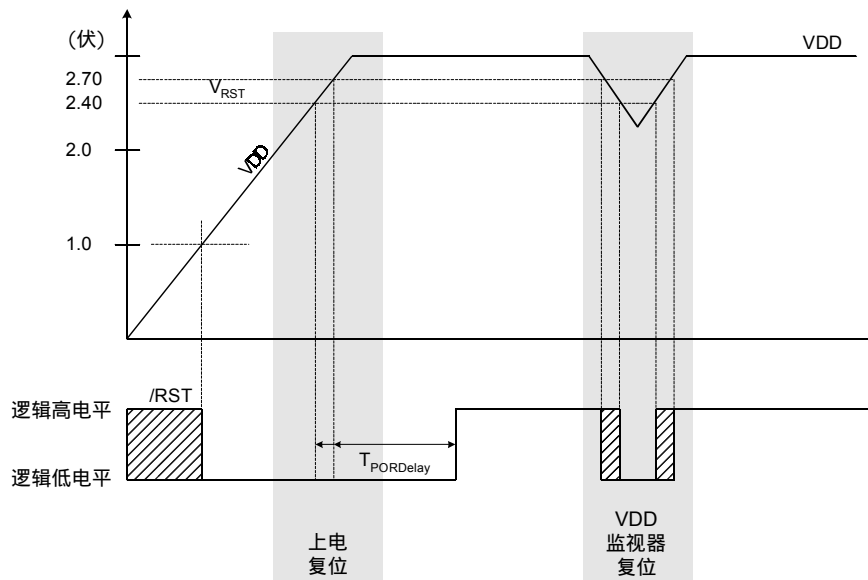


## 9.1 上电复位

在上电期间，器件保持在复位状态， $\overline{\text{RST}}$  引脚被驱动到低电平，直到 VDD 上升到超过  $V_{\text{RST}}$  电平。从复位开始到退出复位状态要经过一个延时；该延时随着 VDD 上升时间的增大而减小（VDD 上升时间被定义为 VDD 从 0V 上升到 2.7V 的时间）。图 9.2 给出了上电和 VDD 监视器复位的时序。最大的 VDD 上升时间为 1ms；上升时间超过该最大值时可能导致器件在 VDD 达到  $V_{\text{RST}}$  电平之前退出复位状态。当上升时间小于 1ms 时，上电复位延时（ $T_{\text{PORDelay}}$ ）通常小于 0.3ms。

在退出复位状态时，PORSF 标志（RSTSRC.1）被硬件置为逻辑‘1’。当 PORSF 标志被置位时，RSTSRC 寄存器中的所有其它复位标志都是不确定的。PORSF 被任何其它复位源清 0。由于所有的复位都导致程序从同一个地址（0x0000）开始执行，软件可以通过读 PORSF 标志来确定是否为上电产生的复位。在一次上电复位后，内部数据存储器的内容应被认为是不确定的。在一次上电复位后，VDD 监视器被禁止。

图 9.2 上电和 VDD 监视器复位时序



## 9.2 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  $V_{\text{RST}}$  以下时，电源监视器将  $\overline{\text{RST}}$  引脚驱动为低电平并使 CIP-51 保持复位状态（见图 9.2）。当 VDD 又回到高于  $V_{\text{RST}}$  的电平时，CIP-51 将退出复位状态，其过程与上电复位时一样。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 标志被置位，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（允许/禁止）不受任何其它复位源的影响。例如，在 VDD 监视器被允许后执行一次软件复位，复位后 VDD 监视器仍然为允许状态。通过向寄存器 RSTSRC 中的 PORSF 位写‘1’来允许 VDD 监视器。图 9.2 给出了 VDD 监视器的时序。注意，在 VDD 监视器复位后没有复位延时。表 9.2 给出了 VDD 监视器的电气特性。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

---

注意：允许 VDD 监视器的同时将产生一次系统复位。器件从复位状态返回后，VDD 监视器处于允许状态。

## 9.3 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'。

## 9.4 时钟丢失检测器复位

时钟丢失检测器 (MSD) 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) 的读值为 '1'，表示本次复位源为 MSD；否则该位读值为 '0'。向 MCDRSF 位写 '1' 将允许时钟丢失检测器；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。

## 9.5 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 之前允许比较器 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 MCU 被置于复位状态。在发生比较器 0 复位后，CORSEF 标志 (RSTSRC.5) 的读值为 '1'，表示本次复位源为比较器 0；否则该位读值为 '0'。/RST 引脚的状态不受该复位的影响。

## 9.6 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。可以通过软件允许或禁止 PCA 的 WDT 功能。在每次复位后，WDT 被允许并使用 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。

## 9.7 FLASH 错误复位

如果一次 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操作失败：

- 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 操作的地址超出了用户代码空间的最大地址。
- FLASH 读地址超出了用户代码空间，即 MOVC 操作的地址超出了用户代码空间的最大地址。
- 程序读超出了保留地址空间。这种情况发生在用户代码试图转移到超过用户代码空间的最大地址时。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

表 9.1 用户代码空间最大地址

器件	用户代码空间最大地址
C8051F300/1/2/3	0x1DFF
C8051F304	0x0FFF
C8051F305	0x07FF

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。

## 9.8 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。

表 9.2 复位源电气特性

-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
/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 输入漏电流	/RST=0.0V		25	40	$\mu\text{A}$
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	$\mu\text{s}$
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			$\mu\text{s}$
产生系统复位的最小/RST 低电平时间		15			$\mu\text{s}$

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 9.3 RSTSRC: 复位源寄存器

R	R	R/W	R/W	R	R/W	R/W	R	复位值
-	FERROR	CORSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF	xxxxxxxx
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xEF

(注: 不要对该寄存器进行读/修改/写操作 (ORL, ANL))

位 7: 未用。读 = 0, 写 = 忽略。

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 复位允许和标志  
写  
0: 比较器 0 不是复位源。  
1: 比较器 0 是复位源 (低电平有效)。  
读  
0: 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: 允许时钟丢失检测器; 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: 上电复位和 VDD 监视器复位标志  
该位在上电复位后被置位。这可以是真正的上电复位或 VDD 监视器复位。不论属于哪种情况, 复位后数据存储寄存器中的内容应被视为是不确定的。  
写  
0: 禁止 VDD 监视器。  
1: 允许 VDD 监视器。  
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。

## 10. FLASH 存储器

C8051F30x 内部有 8KB 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，程序停止执行。表 10.1 给出了 FLASH 存储器的电气特性。

### 10.1 FLASH 存储器编程

对 FLASH 存储器编程的最简单的方法是使用由 Cygnal 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“17. 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。**

#### 10.1.1 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；FLASH 读不受限制。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 10.3 给出了 FLKEY 寄存器的详细说明。

#### 10.1.2 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）将程序存储写允许位 PSWE（PSCTL.0）设置为逻辑 ‘1’（这将使 MOVX 操作指向目标 FLASH 存储器）；2）按顺序向 FLASH 锁定寄存器（FLKEY）写入 FLASH 关键码。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. 置 ‘1’ 程序存储器擦除允许位（PSCTL 中的 PSEE），以允许 FLASH 扇区擦除。
3. 置 ‘1’ 程序存储器写允许位（PSCTL 中的 PSWE），以允许 FLASH 写入。
4. 向 FLKEY 写第一个关键码：0xA5。
5. 向 FLKEY 写第二个关键码：0xF1。
6. 用 MOVX 指令向待擦除页内的任何一个地址写入一个数据字节。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 10.1.3 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（见上节的说明）。
3. 置 ‘1’ PSCTL 中的 PSWE 位。
4. 清除 PSCTL 中的 PSEE 位。
5. 向 FLKEY 写第一个密钥：0xA5。
6. 向 FLKEY 写第二个密钥：0xF1。
7. 用 MOVX 指令向扇区内的目标地址写入一个数据字节。

重复步骤 5-7，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 10.1 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	C8051F300/1/2/3	8192*			字节
FLASH 规模	C8051F304	4096			字节
FLASH 规模	C8051F305	2048			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 0x1E00 ~ 0x1FFF 的 512 字节保留为工厂使用。

## 10.2 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。

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。

## 10.3 安全选项

CIP-51 提供了安全选项以保护 FLASH 存储器不会被软件意外修改，以及防止产权程序代码和常数被读取。程序存储写允许 PSWE (PSCTL.0) 和程序存储擦除允许 PSEE (PSCTL.1) 位保护 FLASH 存储器不会被软件意外修改。在用软件修改 FLASH 存储器的内容之前，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，使其不能通过 C2 接口被读取或改写。见表 10.2 的安全字节说明。图 10.1 给出了每个器件的程序存储器结构和安全字节所在的位置。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

表 10.2 安全字节的意义

位	说明
7-4	<b>写锁定:</b> 将这些位中的任何一位清 0 即可保护全部 FLASH 存储器不能被写入或页擦除 (通过 C2 接口)。
3-0	<b>读/写锁定:</b> 将这些位中的任何一位清 0 即可保护全部 FLASH 存储器不能被读取、写入或页擦除 (通过 C2 接口)。

不管安全字节的设置如何, 锁定位总是可读的并可被清 0。

**注意:** 锁定位一旦被设置, 解除锁定 (写或读/写) 的唯一方法是用 C2 器件擦除命令将整个程序存储器空间擦除。

图 10.1 FLASH 程序存储器结构

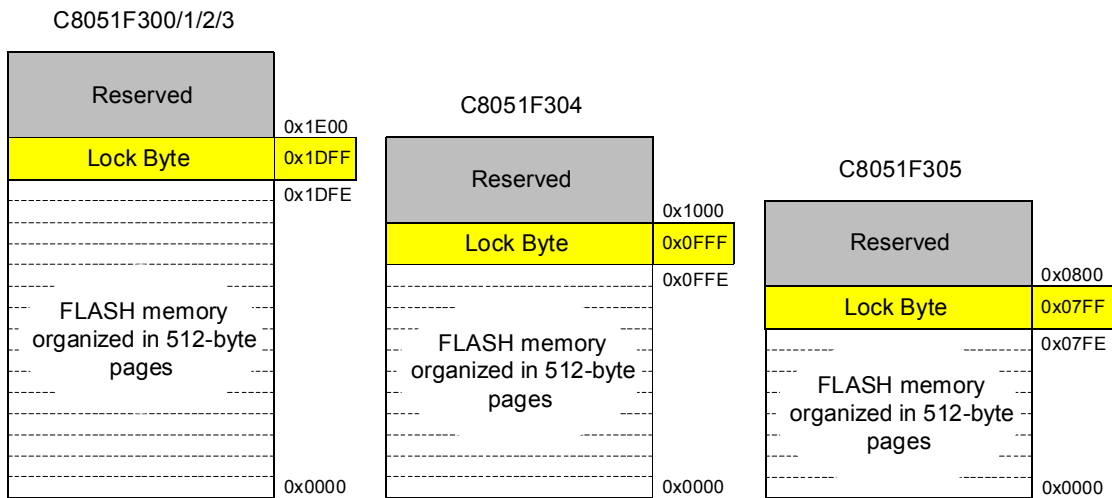


图 10.2 PSCTL: 程序存储读写控制

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

位 7-2: 未使用。读 = 000000b, 写 = 忽略。

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 指令寻址 FLASH 存储器。



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 10.3 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
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xB7

位 7-0: FLKEY: FLASH 锁定和关键码寄存器

写:  
 必须在写或擦除 FLASH 之前写该寄存器。FLASH 保持在锁定状态, 直到该寄存器被写入下面的关键码: 0xA5, 0xF1。写入的时序并不重要, 只要按顺序写即可。每次写或擦除 FLASH 之前都必须写关键码。如果写入了错误的键码或在写关键码之前进行了 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。

**图 10.4 FLSC1: 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

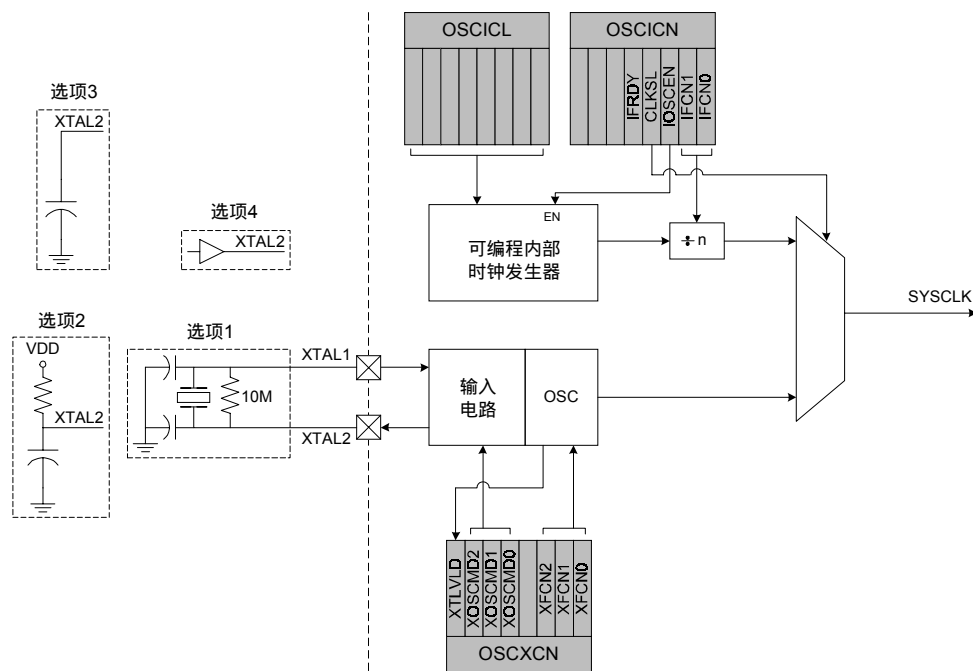
位 7: FOSE: 闪存单稳态定时器允许  
 该位用于允许 50 ns FLASH 读操作单稳态定时器。当 FLASH 单稳态定时器被禁止时, FLASH 读出放大器在整个 FLASH 读周期内被允许。  
 0: 禁止 FLASH 单稳态定时器。  
 1: 允许 FLASH 单稳态定时器。

位 6-0: 保留。读 = 0, 必须写 0。

## 11. 振荡器

C8051F300/1/2/3/4/5 MCU 有一个可编程内部振荡器和一个外部振荡器驱动电路。内部振荡器可以被允许/禁止,其输出频率可以通过 OSCICN 和 OSCICL 寄存器编程(如图 11.1 所示)。系统时钟可以由外部振荡器电路、内部振荡器或内部振荡器分频后提供。表 11.1 给出了内部振荡器的电气特性。

图 11.1 振荡器框图



### 11.1 可编程内部振荡器

C8051F300/1/2/3/4/5 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 11.1 计算振荡周期,其中  $f_{BASE}$  为复位后内部振荡器的频率,  $\Delta T$  为内部振荡器周期的变化值,  $\Delta OSCICL$  为寄存器 OSCICL 中的数值的变量。

方程 11.1 内部振荡器周期随 OSCICL 的典型变化量

$$\Delta T = 0.005 \times \frac{1}{f_{BASE}} \times \Delta OSCICL$$

对于 C8051F300/1 器件, OSCICL 已经过工厂校准,基频 ( $f_{BASE}$ ) 为 24.5MHz。对于 C8051F302/3/4/5,基频的标称值为 20MHz,而随着器件不同,该频率值可以在±20%的范围内变化。

表 11.1 给出了精密内部振荡器的电气特性。注意:可编程内部振荡器的频率不能超过 25MHz。系统时钟可以从内部振荡器分频得到,分频数由寄存器 OSCICN 中的 IFCN 位设定,

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

可为 1、2、4 或 8。复位后的缺省分频数为 8。

## 11.1.1 C8051F300/1 的内部振荡器编程

对于 C8051F300/1, OSCICL 复位值已经过工厂校准, 对应的基频为 24.5MHz, 精度为±2%。注意, OSCICL 的校准复位值可能随器件不同而不同。软件应读 OSCICL 的值并根据方程 11.1 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 20MHz 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。

$$\begin{aligned} f_{BASE} &= 24500000\text{Hz} & f_{DES} &= 20000000\text{Hz} \\ T_{BASE} &= \frac{1}{24500000} \text{ s} & T_{DES} &= \frac{1}{20000000} \text{ s} \end{aligned}$$

所要求的周期变化量 ( $\Delta T_{DES}$ ) 为基频周期和期望周期之差。

$$\Delta T_{DES} = \frac{1}{20000000} - \frac{1}{24500000} = 9.18 \times 10^{-9} \text{ s}$$

将上面的计算值代入方程 11.1, 求  $\Delta OSCICL$ :

$$9.18 \times 10^{-9} = 0.005 \times \frac{1}{f_{BASE}} \times \Delta OSCICL$$

$$\Delta OSCICL = 45$$

将  $\Delta OSCICL$  截断为最近似的整数 (45) 后加到寄存器 OSCICL 的复位值中。此时内部振荡器的频率变为:

$$f_{OSC} = 20000000\text{Hz}$$

注意: 如果 OSCICL 的复位值与  $\Delta OSCICL$  之和大于 128 或小于 0, 则器件不能产生所希望频率。

## 11.1.2 C8051F302/3/4/5 的内部振荡器编程

对于 C8051F302/3/4/5, OSCICL 的复位值不同, 对应的内部振荡器标称频率为 20MHz, 不同器件之间的频率变化范围为±20%。为了将这些器件编程在所期望的频率, 必须在已知 OSCICL 值的情况下测量内部振荡器频率。用所测得的频率和已知的 OSCICL 值, 根据方程 11.1 来逼近所期望的频率, 计算步骤与上节所述相同。对每个器件都必须进行同样的测量和计算过程。可以通过将 SYSCLK 信号连到一个端口引脚来测量内部振荡器频率。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 11.2 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
-								可变
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xB3

位 7: 未用。读=0b, 写 = 忽略。

位 6-0: OSCICL: 内部振荡器校准寄存器。  
该寄存器决定内部振荡器的周期 (根据方程 11.1)。OSCICL 的复位值定义内部振荡器的基频。对于 C8051F300/1, 复位值已经过工厂校准, 对应的基频为 24.5MHz。

图 11.3 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
-	-	-	IFRDY	CLKSL	IOSCEN	IFCN1	IFCN0	00010100
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xB2

位 7-5: 未用。读 = 000b, 写 = 忽略。

位 4: 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 3: CLKSL: 系统时钟源选择位  
0: SYSCLK 来自内部振荡器, 分频数由 IFCN 位决定。  
1: SYSCLK 来自外部振荡器电路。

位 2: IOSCEN: 内部振荡器允许位  
0: 内部振荡器禁止。  
1: 内部振荡器允许。

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)。

表 11.1 内部振荡器电气特性

-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
已校准的内部振荡器频率	C8051F300/1	24	24.5	25	MHz
未校准的内部振荡器频率	C8051F302/3/4/5	16	20	24	MHz
内部振荡器电源电流 (由 VDD 供电)	OSCICN.2 = 1		450		μA

## 11.2 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 11.1，选项 1），还必须在 XTAL1 和 XTAL2 引脚之间并接一个 10MΩ 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 11.1，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 XFNCN（见图 11.4）。

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 分别被用作 XTAL2。端口 I/O 交叉开关应被配置为跳过被占用的引脚。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“12.2 端口 I/O 初始化”。

## 11.3 系统时钟选择

寄存器 OSCICN 中的 CLKSL 位选择用于产生系统时钟的振荡器。如果要选择外部振荡器作为系统时钟，必须将 CLKSL 位置 ‘1’。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）提供时钟。系统时钟可以在内部振荡器和外部振荡器之间自由切换，只要所选择的振荡器被允许并稳定运行。内部振荡器的起动时间很短，因此可以在同一个 OSCICN 写操作中允许和选择内部振荡器。外部晶体和陶瓷谐振器通常需要较长的起动时间，应待其稳定后方可用作系统时钟。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，软件在允许外部振荡器和检查 XTLVLD 之间至少应延时 1ms。RC 和 C 方式通常不需要起动时间。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 11.4 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																																				
<p>位 7:     XTLVLD: 晶体振荡器有效标志           (只在 XOSCMD=11x 时有效)</p> <p>0: 晶体振荡器未用或未稳定。 1: 晶体振荡器正在运行并且工作稳定。</p> <p>位 6-4:   XOSCMD2-0: 外部振荡器方式位</p> <p>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 振荡器方式二分频。 101: 电容振荡器方式二分频。 110: 晶体振荡器方式。 111: 晶体振荡器方式二分频。</p> <p>位 3:     保留。读 = 0, 写 = 忽略。</p> <p>位 2-0:   XFCN2-0: 外部振荡器频率控制位。 000-111: 见下表</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">XFCN</th> <th style="width: 30%;">晶体 (XOSCMD=11x)</th> <th style="width: 30%;">RC(XOSCMD=10x)</th> <th style="width: 30%;">C(XOSCMD=10x)</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">000</td> <td style="text-align: center;">F ≤ 32kHz</td> <td style="text-align: center;">F ≤ 25kHz</td> <td style="text-align: center;">K 因子= 0.87</td> </tr> <tr> <td style="text-align: center;">001</td> <td style="text-align: center;">32kHz &lt; f ≤ 84kHz</td> <td style="text-align: center;">25kHz &lt; f ≤ 50kHz</td> <td style="text-align: center;">K 因子= 2.6</td> </tr> <tr> <td style="text-align: center;">010</td> <td style="text-align: center;">84kHz &lt; f ≤ 225kHz</td> <td style="text-align: center;">50kHz &lt; f ≤ 100kHz</td> <td style="text-align: center;">K 因子= 7.7</td> </tr> <tr> <td style="text-align: center;">011</td> <td style="text-align: center;">225kHz &lt; f ≤ 590kHz</td> <td style="text-align: center;">100kHz &lt; f ≤ 200kHz</td> <td style="text-align: center;">K 因子= 22</td> </tr> <tr> <td style="text-align: center;">100</td> <td style="text-align: center;">590kHz &lt; f ≤ 1.5MHz</td> <td style="text-align: center;">200kHz &lt; f ≤ 400kHz</td> <td style="text-align: center;">K 因子= 65</td> </tr> <tr> <td style="text-align: center;">101</td> <td style="text-align: center;">1.5MHz &lt; f ≤ 4MHz</td> <td style="text-align: center;">400kHz &lt; f ≤ 800kHz</td> <td style="text-align: center;">K 因子= 180</td> </tr> <tr> <td style="text-align: center;">110</td> <td style="text-align: center;">4MHz &lt; f ≤ 10MHz</td> <td style="text-align: center;">800kHz &lt; f ≤ 1.6MHz</td> <td style="text-align: center;">K 因子= 664</td> </tr> <tr> <td style="text-align: center;">111</td> <td style="text-align: center;">10MHz &lt; f ≤ 30MHz</td> <td style="text-align: center;">1.6MHz &lt; f ≤ 3.2MHz</td> <td style="text-align: center;">K 因子= 1590</td> </tr> </tbody> </table>									XFCN	晶体 (XOSCMD=11x)	RC(XOSCMD=10x)	C(XOSCMD=10x)	000	F ≤ 32kHz	F ≤ 25kHz	K 因子= 0.87	001	32kHz < f ≤ 84kHz	25kHz < f ≤ 50kHz	K 因子= 2.6	010	84kHz < f ≤ 225kHz	50kHz < f ≤ 100kHz	K 因子= 7.7	011	225kHz < f ≤ 590kHz	100kHz < f ≤ 200kHz	K 因子= 22	100	590kHz < f ≤ 1.5MHz	200kHz < f ≤ 400kHz	K 因子= 65	101	1.5MHz < f ≤ 4MHz	400kHz < f ≤ 800kHz	K 因子= 180	110	4MHz < f ≤ 10MHz	800kHz < f ≤ 1.6MHz	K 因子= 664	111	10MHz < f ≤ 30MHz	1.6MHz < f ≤ 3.2MHz	K 因子= 1590
XFCN	晶体 (XOSCMD=11x)	RC(XOSCMD=10x)	C(XOSCMD=10x)																																									
000	F ≤ 32kHz	F ≤ 25kHz	K 因子= 0.87																																									
001	32kHz < f ≤ 84kHz	25kHz < f ≤ 50kHz	K 因子= 2.6																																									
010	84kHz < f ≤ 225kHz	50kHz < f ≤ 100kHz	K 因子= 7.7																																									
011	225kHz < f ≤ 590kHz	100kHz < f ≤ 200kHz	K 因子= 22																																									
100	590kHz < f ≤ 1.5MHz	200kHz < f ≤ 400kHz	K 因子= 65																																									
101	1.5MHz < f ≤ 4MHz	400kHz < f ≤ 800kHz	K 因子= 180																																									
110	4MHz < f ≤ 10MHz	800kHz < f ≤ 1.6MHz	K 因子= 664																																									
111	10MHz < f ≤ 30MHz	1.6MHz < f ≤ 3.2MHz	K 因子= 1590																																									
<p><b>晶体方式</b> (电路见图 11.1, 选项 1; XOSCMD=11x) 选择与晶体振荡器频率匹配的 XFCN 值。</p> <p><b>RC 方式</b> (电路见图 11.1, 选项 2; XOSCMD=10x) 选择与频率范围匹配的 XFCN 值: <math>f = 1.23(10^3)/(R * C)</math>, 其中: f = 振荡器频率 (MHz) C = 电容值 (pF) R = 上拉电阻值 (kΩ)</p> <p><b>C 方式</b> (电路见图 11.1, 选项 3; XOSCMD=10x) 根据所期望的振荡器频率选择 K 因子 (KF): <math>f = KF/(C * VDD)</math>, 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)</p>																																												

## 11.4 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 11.1 中的选项 1。应从图 11.4 (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 11.0592MHz 的晶体要求 XFCN 设置值为 111b。

在晶体振荡器被允许时，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。在允许晶体振荡器和检查 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. 允许外部振荡器。
2. 等待至少 1ms。
3. 查询 XTLVLD => '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。

## 11.5 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 11.1 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 = 246kΩ 和 C = 50pF：

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

查图 11.4 中的表，得到所需要的 XFCN 值为 010b。

## 11.6 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 11.1 中的选项 3。电容不应大于 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 = 3.0V 和 C = 50pF：

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

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

如果所需要的频率大约为 150kHz，从图 11.4 的表中选择 K 因子，得到 KF = 22：

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

因此，本例中要用的 XFCN 值为 011b。

## 12. 端口输入/输出

数字和模拟资源要通过 8 位的 I/O 端口 P0 才能使用。每个引脚都可以被定义为通用 I/O (GPIO)、模拟输入或被分配给一个内部数字外设，如图 12.3 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 12.3 和图 12.4）为所选择的内部数字资源分配 I/O 引脚。寄存器 XBR0、XBR1 和 XBR2（见图 12.5 – 12.7）用于选择内部数字功能。

所有端口 I/O 都耐 5V 电压（端口 I/O 单元示于图 12.2）。端口 I/O 单元可以被配置为漏极开路或推挽方式（在端口 0 输出方式寄存器 P0MDOUT 中设置）。表 12.1 给出了端口 I/O 的电气特性。

图 12.1 端口 I/O 功能框图

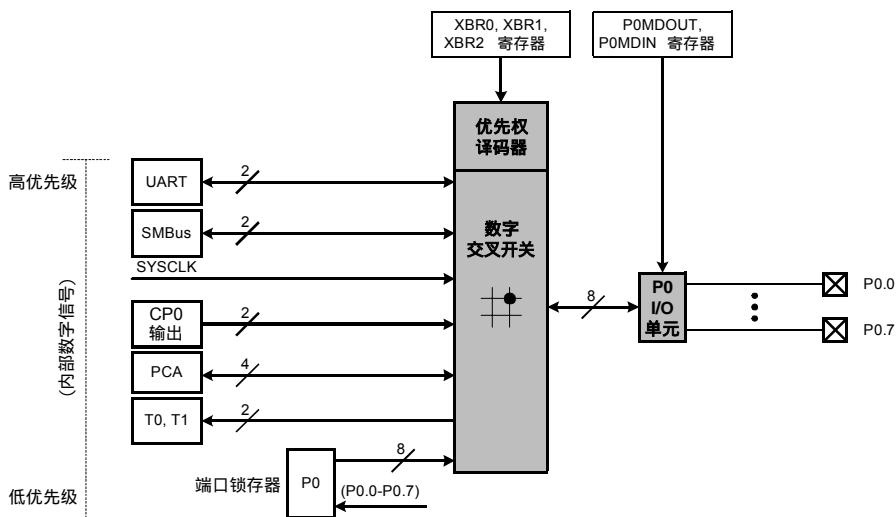
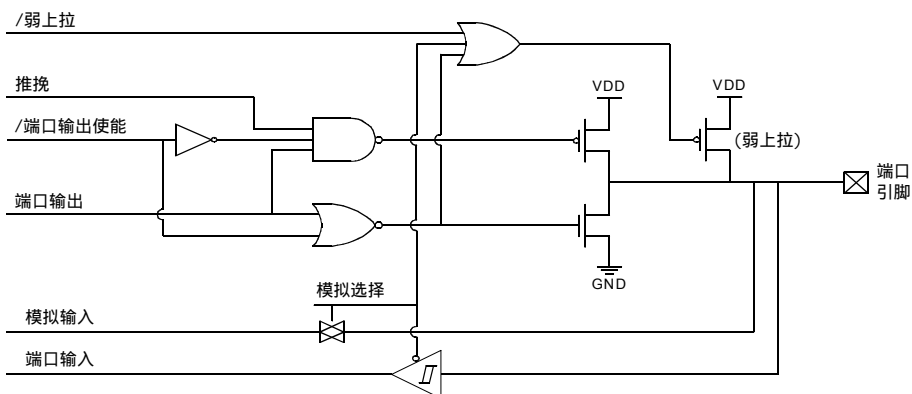


图 12.2 端口 I/O 单元框图





# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 12.1 优先权交叉开关译码器

优先权交叉开关译码器（图 12.3）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 XBR0 寄存器中被置‘1’的那些位所对应的引脚。XBR0 寄存器允许用软件控制交叉开关跳过那些被用作模拟输入或 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 XBR0 寄存器中的对应位应被置‘1’。这种情况适用于 P0.0（如果 VREF 被允许）、P0.3 和/或 P0.2（如果外部振荡器电路被允许）、P0.6（如果 ADC 被配置为使用外部转换启动信号 CNVSTR）和任何被选择的 ADC 或比较器输入。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 12.3 示出没有引脚被跳过（XBR0 = 0x00）的优先权交叉开关译码表；图 12.4 给出了 6 脚和 2 脚被跳过情况下（XBR0 = 0x44）的交叉开关优先权译码表。

图 12.3 交叉开关优先权译码表（XBAR0 = 0x00）

SF信号	P0								
	VREF	x1	x2	CNVSTR					
PIN I/O	0	1	2	3	4	5	6	7	
TX0					■	■			
RX0							■	■	
SDA	■								
SCL		■							
SYSCLK	■		■						
CP0	■	■	■	■					
CP0A	■	■	■	■	■				
CEX0	■	■	■	■	■	■			
CEX1		■	■	■	■	■	■		
CEX2			■	■	■	■	■	■	
ECl	■	■	■	■	■	■	■	■	■
T0	■	■	■	■	■	■	■	■	■
T1	■	■	■	■	■	■	■	■	■
	0	0	0	0	0	0	0	0	
	XBR0[0:7]								

信号不可用

- 可供外设使用的端口引脚
- SF信号 特殊功能信号不用交叉开关分配引脚  
当这些信号被允许时 交叉开关必须  
被配置为跳过相应的端口引脚。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 12.4 交叉开关优先级译码表 (XBAR0 = 0x44)

SF信号	P0							
	VREF		x1	x2	CNVSTR			7
	0	1	2	3	4	5	6	
PIN I/O	0	1	2	3	4	5	6	7
TX0								
RX0								
SDA								
SCL								
SYSCLK								
CP0								
CP0A								
CEX0								
CEX1								
CEX2								
ECI								
T0								
T1								
	0	0	1	0	0	0	1	0
	XBR0[0:7]							

- 可供外设使用的端口引脚
- 被交叉开关跳过的端口引脚
- SF信号 特殊功能信号不用交叉开关分配引脚  
当这些信号被允许时 交叉开关必须  
被配置为跳过相应的端口引脚。

寄存器 XBR1 和 XBR2 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，交叉开关可以为其分配一个或两个引脚。UART0 的引脚分配是固定的(这是出于引导装入的目的): 当 UART TX0 被选择时，它总是被分配到 P0.4; 当 UART RX0 被选择时，它总是被分配到 P0.5。在优先功能被分配之后，标准端口 I/O 是连续的。例如，如果被分配的功能使用 3 个端口 I/O (P0.[2:0]，则剩下的 5 个端口 I/O 用于模拟输入或 GPIO。

## 12.2 端口 I/O 初始化

端口 I/O 初始化包括以下步骤:

1. 用端口 0 输入方式寄存器 (P0MDIN) 选择所有端口引脚的输入方式 (模拟或数字)。
2. 用端口 0 输出方式寄存器 (P0MDOUT) 选择所有端口引脚的输出方式 (漏极开路或推挽)。
3. 设置 XBR0，使交叉开关跳过那些被选择为模拟输入或特殊功能的引脚。
4. 将引脚分配给要使用的外设。
5. 允许交叉开关。

## C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

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 输入的任何引脚都必须被配置为模拟输入。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。

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 XBR0 寄存器中的对应位置‘1’来实现）。端口输入方式在 P0MDIN 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 P0MDIN 寄存器的详细说明见图 12.9。

I/O 引脚的输出驱动器特性由端口 0 输出方式寄存器 P0MDOUT 中的对应位决定（见图 12.10），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 P0MDOUT 的设置如何，这两个引脚总是被配置为漏极开路。

当 XBR2 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允许。弱上拉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、XBR1 和 XBR2 必须被装入正确的值以选择设计所需要的数字 I/O 功能。置‘1’ XBR2 中的 XBARE 位将允许交叉开关。不管 XBRn 寄存器的设置如何，在交叉开关被允许之前，外部引脚保持标准数字输入方式。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 Cygnal IDE 软件的配置向导功能来确定基于 XBRn 寄存器设置的 I/O 引脚分配。

**图 12.5 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
-	XSKP6	XSKP5	XSKP4	XSKP3	XSKP2	XSKP1	XSKP0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xE1
<p>位 7: 未用。读 = 0b, 写 = 忽略。</p> <p>位 6-0: XSK[6:0]: 交叉开关跳过允许位                      这些位用于选择需要被交叉开关译码器跳过的端口引脚。用于模拟输入（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: 交叉开关不跳过对应的 P0.n 引脚。                      1: 交叉开关跳过对应的 P0.n 引脚。</p>								

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 12.6 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	
	PCA0ME	CP0AOEN	CP0OEN	SYSCKE	SMB0OEN	URX0EN	UTX0EN	复位值 00000000
	位7	位6	位5	位4	位3	位2	位1	SFR地址: 0xE2
位 7-6:	PCA0ME: PCA 模块 I/O 允许位							
	00: 所有的 PCA I/O 都不连到端口引脚。							
	01: CEX0 连到端口引脚。							
	10: CEX0、CEX1 连到两个端口引脚。							
	11: CEX0、CEX1、CEX2 连到三个端口引脚。							
位 5:	CP0AOEN: 比较器 0 异步输出允许位							
	0: CP0A 不连到端口引脚。							
	1: CP0A 连到端口引脚。							
位 4:	CP0OEN: 比较器 0 输出允许位							
	0: CP0 不连到端口引脚。							
	1: CP0 连到端口引脚。							
位 3:	SYSCKE: /SYSCLK 输出允许位							
	0: /SYSCLK 不连到端口引脚;							
	1: /SYSCLK 连到端口引脚。							
位 2:	SMB0OEN: SMBus I/O 允许位							
	0: SMBus I/O 不连到端口引脚。							
	1: SDA 和 SCL 连到端口引脚。							
位 1:	URX0EN: UART RX 允许位							
	0: UART RX0 不连到端口引脚。							
	1: UART RX0 连到端口引脚 P0.5。							
位 0:	UTX0EN: UART TX 输出允许位							
	0: UART TX0 不连到端口引脚。							
	1: UART TX0 连到端口引脚 P0.4。							

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 12.7 XBR2: 端口 I/O 交叉开关寄存器 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
WEAKPUD	XBARE	-	-	-	T1E	T0E	ECIE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xE3
位 7:	WEAKPUD: 端口 I/O 弱上拉禁止位。 0: 弱上拉允许 (被配置为推挽方式的 I/O 除外)。 1: 弱上拉禁止。							
位 6:	XBARE: 交叉开关允许位 0: 交叉开关禁止。 1: 交叉开关允许。							
位 5-3:	未用。读 = 000b, 写 = 忽略。							
位 2:	T1E: T1 允许位 0: T1 不连到端口引脚。 1: T1 连到端口引脚。							
位 1:	T0E: T0 允许位 0: T0 不连到端口引脚。 1: T0 连到端口引脚。							
位 0:	ECIE: PCA0 计数器输入允许位 0: ECI 不连到端口引脚。 1: ECI 连到端口引脚。							

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 12.3 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。通过端口数据寄存器 P0（见图 14.8）访问端口 0，该寄存器是一个既可以按位寻址也可以按字节寻址的特殊功能寄存器。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，但在执行读-修改-写指令（ANL、ORL、XRL、CPL、INC、DEC、DJNZ、JBC、CLR、SETB 和位传送操作）期间例外。在读-修改-写指令的读周期，所读的值是端口数据寄存器（或端口位）的内容，而不是端口引脚本身的状态。

图 12.8 P0: 端口 0 寄存器

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

位 7-0: P0.[7:0]  
 写 — 输出出现在 I/O 引脚（根据 XBR0、XBR1 和 XBR2 寄存器的设置）。  
 0: 逻辑低电平输出。  
 1: 逻辑高电平输出。（若相应的 P0MDOUT.n 位 = 0，则为漏极开路）。  
 读 — 读那些在 P0MDIN 中被选择为模拟输入的引脚时总是返回 ‘1’。读那些被选择为数字输入的引脚时返回端口锁存器的值。  
 0: P0.n 为逻辑低电平。  
 1: P0.n 为逻辑高电平。

图 12.9 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: P0MDIN.[7:0]: P0.7 – P0.0 输入方式配置位（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 引脚被配置为数字输入。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 12.10 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
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址: 0xA4
<p>位 7-0: P0MDOUT.[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。</p> <p>0: 对应的 P0.n 输出为漏极开路。</p> <p>1: 对应的 P0.n 输出为推挽方式。</p> <p>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。</p>								

**表 12.1 端口 I/O 直流电气特性**

VDD = 2.7V – 3.6V, -40°C到+85°C（除非特别说明）。

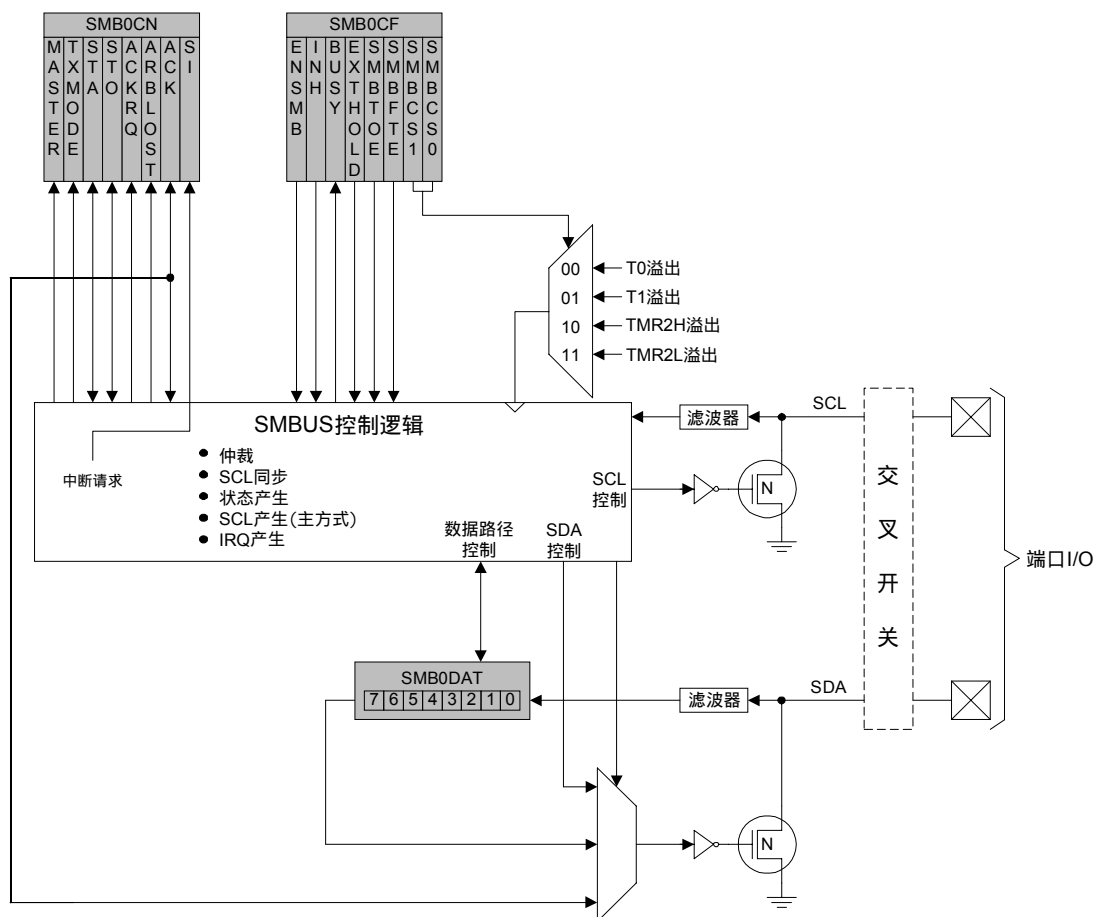
参 数	条 件	最小值	典型值	最大值	单 位
输出高电压 (V <sub>OH</sub> )	I <sub>OH</sub> = -10μA, 端口 I/O 为推挽方式 I <sub>OH</sub> = -3mA, 端口 I/O 为推挽方式 I <sub>OH</sub> = -10mA, 端口 I/O 为推挽方式	VDD-0.1 VDD-0.7	VDD-0.8		V
输出低电压 (V <sub>OL</sub> )	I <sub>OL</sub> = 10μA I <sub>OL</sub> = 8.5mA I <sub>OL</sub> = 25mA		1.0	0.1 0.6	V
输入高电压 (V <sub>IH</sub> )		2			V
输入低电压 (V <sub>IL</sub> )				0.8	V
输入漏电流	弱上拉禁止 弱上拉允许, V <sub>IN</sub> = 0 V		25	±1 40	μA

## 13. 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 I<sup>2</sup>C 串行总线兼容。系统控制器对总线的读写操作都是以字节为单位的，由 SMBus 接口自动控制数据的串行传输。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。

图 13.1 SMBus 原理框图





## 13.1 支持文档

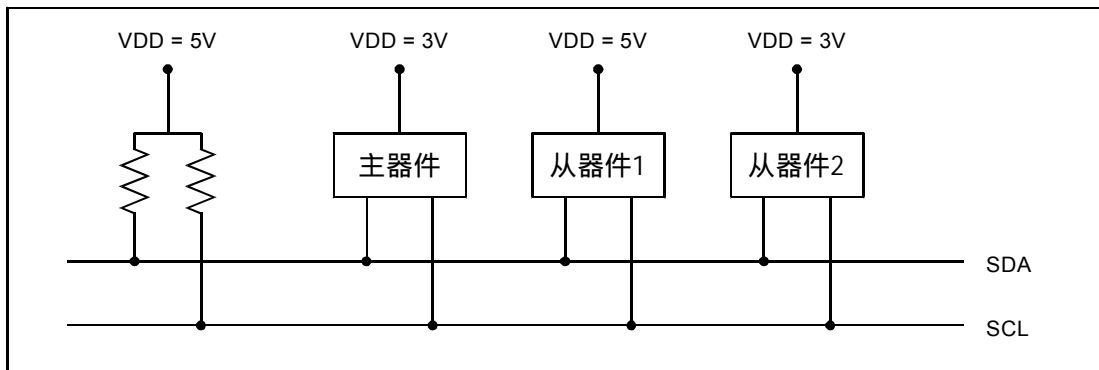
假设读者熟悉或有条件阅读下列支持文档：

1. I<sup>2</sup>C 总线及使用（包含规范），菲利浦半导体。
2. I<sup>2</sup>C 总线标准—2.0 版，菲利浦半导体。
3. 系统管理总线规范—1.1 版，SBS

## 13.2 SMBus 配置

图 13.2 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 都必须为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 13.2 典型 SMBus 配置



## 13.2 SMBus 操作

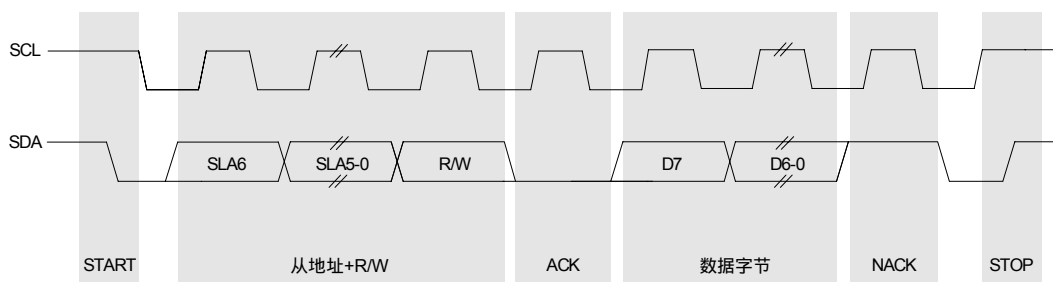
有两种可能的数据传输类型：从主发送器到所寻址的从接收器（写）和从被寻址的从发送器到主接收器（读）。这两种数据传输都由主器件启动，主器件还提供串行时钟。SMBus 接口可以工作在主方式或从方式。总线上可以有多个主器件。如果两个或多个主器件同时启动数据传输，仲裁机制将保证有一个主器件会赢得总线。注意：没有必要在一个系统中指定某个器件作为主器件；任何一个发送起始条件（START）和从器件地址的器件就成为该次数据传输的主器件。

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

方向位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 13.3 示出了一次典型的 SMBus 数据传输过程。

图 13.3 SMBus 数据传输



### 13.3.1 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，则总线是空闲的。两个或多个主器件可能在同一时刻产生起始条件。由于产生起始条件的器件并不知道其它器件也正想占用总线，所以使用仲裁机制迫使一个主器件放弃总线。这些主器件继续发送起始条件，直到其中一个主器件发送高电平而其它主器件在 SDA 上发送低电平。赢得总线的器件继续其数据传输过程，而未赢得总线的器件成为从器件。该仲裁机制是非破坏性的：总会有一个器件赢得总线，不会发生数据丢失。

## 13.3.2 时钟低电平扩展

SMBus 提供一种与 I<sup>2</sup>C 类似的同步机制，允许不同速度的器件共存于一个总线上。为了使低速从器件能与高速主器件通信，在传输期间采取低电平扩展。从器件可以保持 SCL 为低电平以扩展时钟低电平时间，这实际上相当于降低了串行时钟频率。

## 13.3.3 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 2 被用于检测 SCL 低电平超时。定时器 2 在 SCL 为高电平时被强制重装载，在 SCL 为低电平时开始计数。在定时器 2 被允许并且溢出周期被配置为 25ms 时（且 SMBTOE 被置 1），定时器 2 中断服务程序可以用于对 SMBus 复位（在发生 SCL 低电平超时）。

## 13.3.4 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。

## 13.4 SMBus 的使用

SMBus 刻有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）和 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。

主器件产生起始条件时会产生一个中断，指示数据传输开始；从器件在检测到停止条件时产生一个中断，指示数据传输结束。软件应通过读 SMB0CN（SMBus 控制寄存器）来确定 SMBus 中断的原因。

SMBus 配置选项包括：

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

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“13.4.1 SMBus 配置寄存器”。

## 13.4.1 SMBus 配置寄存器

SMBus 配置寄存器（SMB0CF，见图 13.5）用于允许 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 13.1），时钟源只在主方式或空闲超时检测被允许时使用。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 13.1 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 13.1 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 13.1 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 13.1 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 13.2 估算：

方程 13.2 典型 SMBus 位速率

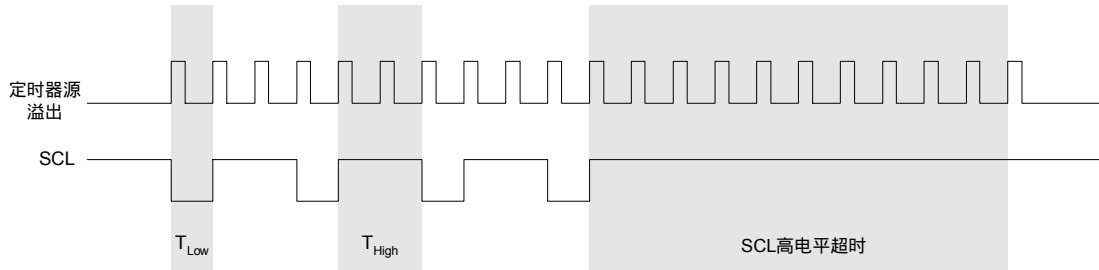
$$\text{位速率} = \frac{1}{f_{ClockSourceOverflow}}$$

图 13.4 给出了由方程 13.2 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，

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

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 13.1 定义的极限值。

图 13.4 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。SMBus 规定的最小建立和保持时间分别为 250ns 和 300ns。必要时应将 EXTHOLD 位置 ‘1’，以保证最小建立和保持时间符合 SMBus 规范。表 13.2 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 13.2 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 <sup>†</sup>	3 个系统时钟
1	11 个系统时钟	12 个系统时钟

<sup>†</sup>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 2 应被配置为以 25ms 的溢出周期溢出，以检测 SCL 低电平超时。SMBus 接口在 SCL 为高电平时强制重装载定时器 2，并允许定时器 2 在 SCL 为低电平时开始计数。应使用定时器 2 中断服务程序对 SMBus 复位，这可通过先禁止然后再重新允许 SMBus 接口来实现。定时器 2 的配置说明见 “15.2 定时器 2”。

通过将 SMBFTE 位置 ‘1’ 来允许 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 13.4），总线将被视为空闲。当检测到空闲超时，SMBus 接口的表现就如同检测到一个停止条件（产生一个中断，STO 被置位）。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 13.5 SMB0CF: SMBus 配置寄存器**

	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
	ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	FTE	TOE	00000000
	位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xC1
								(可位寻址)	
位 7:	ENSMB: SMBus 允许 该位允许/禁止 SMBus 串行接口。当被允许时, 接口一直监视 SDA 和 SCL 引脚。 0: 禁止 SMBus。 1: 允许 SMBus。								
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 从器件从总线上移出。主方式中断不受影响。 0: SMBus 从方式允许。 1: SMBus 从方式禁止。								
位 5:	BUSY: 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’。 0: SMBus 空闲。 1: SMBus 忙。								
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 (见表 13.2)。 0: 禁止 SDA 建立和保持时间扩展。 1: 允许 SDA 建立和保持时间扩展								
位 3:	SMBTOE: SMBus SCL 超时检测允许位 该位允许/禁止 SCL 低电平超时检测。当被置 ‘1’ 时, SMBus 接口在 SCL 为高电平时强制重载定时器 2, 并允许定时器 2 在 SCL 为低电平时开始计数。应将定时器 2 编程为每 25ms 产生一次中断, 并使用定时器 2 中断服务程序对 SMBus 复位。								
位 2:	SMBFTE: SMBus 空闲超时检测允许位 当该位被置 ‘1’ 时, 如果 SCL 和 SDA 保持高电平的时间超过 10 个 SMBus 时钟周期, 总线将被视为空闲。								
位 1-0:	SMBCS1-SMBCS0: SMBus 时钟源选择位 这两位选择用于产生 SMBus 位速率的时钟源。应根据方程 13.1 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				

### 13.4.2 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 13.6)。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。

当 SMBus 工作在主方式时, 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。在停止条件产生后 STO 由硬件清除。在从方式, STO 指示自上次 SMBus 中断以来检测到了停止条件。

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。

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

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

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

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。如果接收的从地址不被确认, 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 13.3。

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。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

表 13.3 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 13.4 的 SMBus 状态译码表。

**图 13.6 SMB0CN: SMBus 控制寄存器**

	R	R	R/W	R/W	R	R	R/W	R/W	
	MASTER	TXMODE	STA	STO	ACKRQ	ARBLOST	ACK	SI	复位值 00000000
	位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址: 0xC0
位 7:	MASTER: SMBus 主/从标志 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 串行中断标志 当出现表 13.3 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, 总线状态被冻结。								



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

表 13.3 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	<ul style="list-style-type: none"> <li>产生了起始条件。</li> </ul>	<ul style="list-style-type: none"> <li>产生了停止条件。</li> <li>在总线竞争中失败。</li> </ul>
TXMODE	<ul style="list-style-type: none"> <li>产生了起始条件。</li> <li>在一个 SMBus 帧开始之前写了 SMB0DAT。</li> </ul>	<ul style="list-style-type: none"> <li>检测到起始条件。</li> <li>竞争失败。</li> <li>在一个 SMBus 帧开始之前没写 SMB0DAT。</li> </ul>
STA	<ul style="list-style-type: none"> <li>在起始条件后接收到一个地址字节。</li> </ul>	<ul style="list-style-type: none"> <li>必须用软件清除。</li> </ul>
STO	<ul style="list-style-type: none"> <li>在作为从器件被寻址时检测到一个停止条件。</li> <li>因检测到停止条件而导致竞争失败。</li> </ul>	<ul style="list-style-type: none"> <li>产生了一个挂起的停止条件。</li> </ul>
ACKRQ	<ul style="list-style-type: none"> <li>接收到一个字节并需要一个 ACK 响应值。</li> </ul>	<ul style="list-style-type: none"> <li>每个 ACK 周期之后。</li> </ul>
ARBLOST	<ul style="list-style-type: none"> <li>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）。</li> <li>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。</li> <li>在试图发送 ‘1’ 时检测到 SDA 为低电平（ACK 位除外）。</li> </ul>	<ul style="list-style-type: none"> <li>每次 SI 被清除时。</li> </ul>
ACK	<ul style="list-style-type: none"> <li>输入的 ACK 值为低（确认）。</li> </ul>	<ul style="list-style-type: none"> <li>输入的 ACK 值为高（非确认）。</li> </ul>
SI	<ul style="list-style-type: none"> <li>产生了一个起始条件。</li> <li>竞争失败。</li> <li>发送了一个字节并收到一个 ACK/NACK</li> <li>接收到一个字节。</li> <li>在起始条件或重复起始条件之后接收到一个地址字节+R/W。</li> <li>收到一个停止条件。</li> </ul>	<ul style="list-style-type: none"> <li>必须用软件清除。</li> </ul>

## 13.4.3 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 被允许但 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 13.7 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
<p>位 7-0: SMB0DAT: SMBus 数据</p> <p>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。</p>								

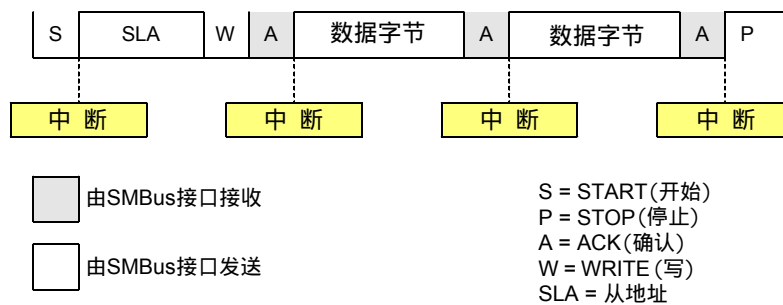
## 13.5 SMBus 传输方式

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。

### 13.5.1 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，并在每发送完一个字节后等待由从器件产生的确认位 (ACK)。最后，为了指示串行传输的结束，主器件产生一个停止条件。图 13.8 给出了典型的主发送器时序。

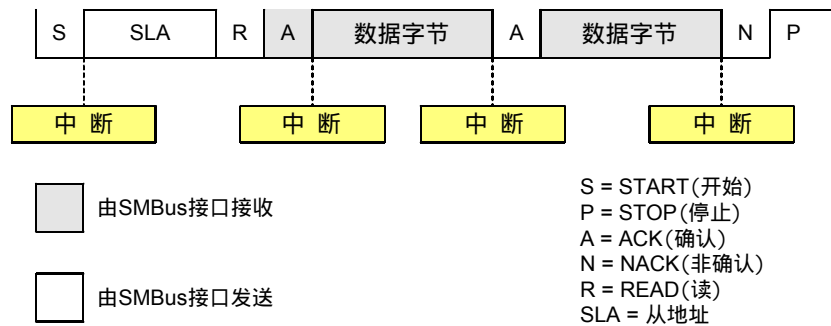
图 13.8 典型的主发送器时序



## 13.5.2 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。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。图 13.9 给出了典型的主接收器时序。

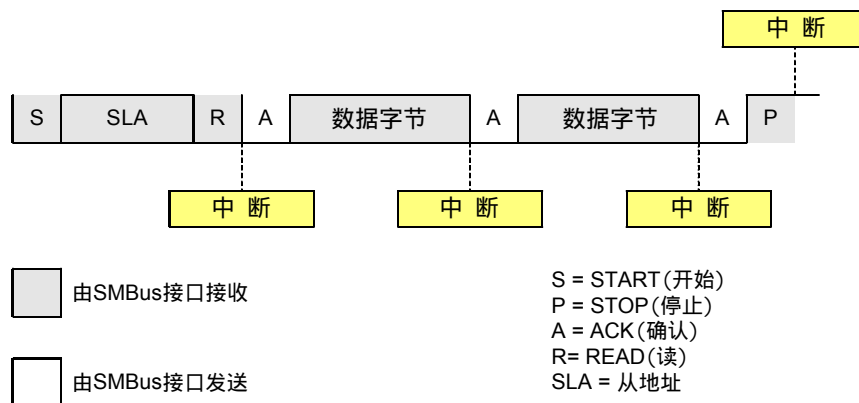
图 13.9 典型的主接收器时序



## 13.5.3 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，软件应向 ACK 位写 ACK 或 NACK，对接收字节作出应答。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 13.10 给出了典型的从接收器时序。注意，在该方式下“数据字节传输中断”发生在 ACK 周期之前。

图 13.10 典型的从接收器时序



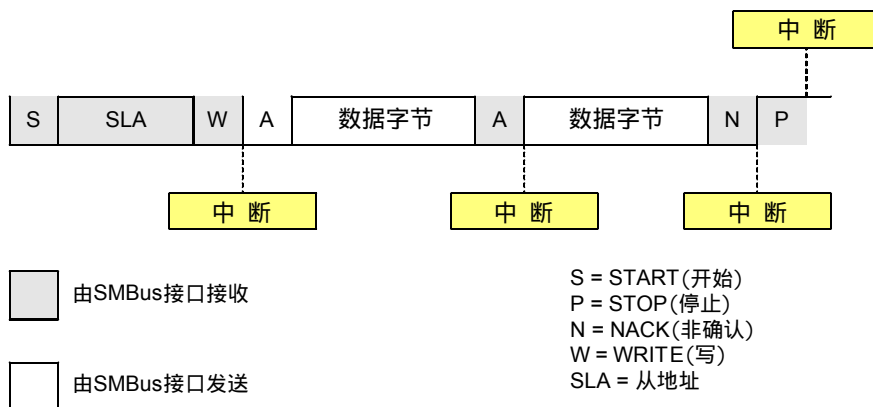
## 13.5.4 从发送器方式

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)。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 写入数据，使 SMBus 进入从发送器方式。由从接收器到从发送器的切换由软件管理。当收到一个有效的从地址时，软件应执行以下步骤：

1. 将 ACK 置 '1'。
2. 写输出数据到 SMB0DAT。
3. 检查 SMB0DAT.7，如果为 1，不执行步骤 4、6 或 7。
4. 将 STO 置 '1'。
5. 将 SI 清 '0'。
6. 查询 TXMODE --> '1'。
7. 将 STO 清 '0' (必须在下一个 ACK 周期之前完成)。

接口进入从发送器方式，并发送一个或多个字节的数据 (上述步骤只在传输第一个字节时需要)。在每发送一个字节后，等待由主器件发出的确认位。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 13.11 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 13.11 典型的从发送器时序



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 13.6 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 13.4 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，特定应用的过程是允许的。表中被突出显示的响应选项是允许的，但不符合 SMBus 规范。

表 13.4 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	STO	ACK
主发送器	1110	0	0	X	起始条件已发出。	将从地址+R/W 装入到 SMB0DAT。	0	0	X
		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。 放弃发送。	1	0	X
	1100	0	0	1	数据或地址字节已发出； 收到 ACK。	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
						<b>切换到主接收器方式（清除 SI，不向 SMB0DAT 写新数据）。</b>	<b>0</b>	<b>0</b>	<b>X</b>
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。	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
						<b>发 ACK 并切换到主发送器方式（在清除 SI 之前写 SMB0DAT）。</b>	<b>0</b>		<b>1</b>
						<b>发 NACK 并切换到主发送器方式（在清除 SI 之前写 SMB0DAT）。</b>	<b>0</b>	<b>0</b>	<b>0</b>

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

表 13.4 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	ACK			STA	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
					不对接收到的地址进行确认。	0	0	0	
		1	1	X	竞争主器件失败; 收到从地址; 请求确认。	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
						重新启动失败的传输; 不对接收到的地址进行确认。	1	0	0
	0010	0	1	X	试图发送重复起始条件时竞争失败。	放弃传输。	0	0	X
					重新启动失败的传输	1	0	X	
	0001	1	1	X	试图发送停止条件时竞争失败。	不需任何操作 (等待停止条件)。	0	0	0
					0	0	X	检测到停止条件。	将下一个要发送的数据字节装入到 SMB0DAT。
		0	1	X	因检测到停止条件而导致竞争失败。	放弃传输。	0	0	X
						重新启动失败的传输。	1	0	X
	0000	1	0	X	接收到字节; 请求确认。	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。	放弃传输。	0	0	0	
					重新启动失败的传输。	1	0	0	

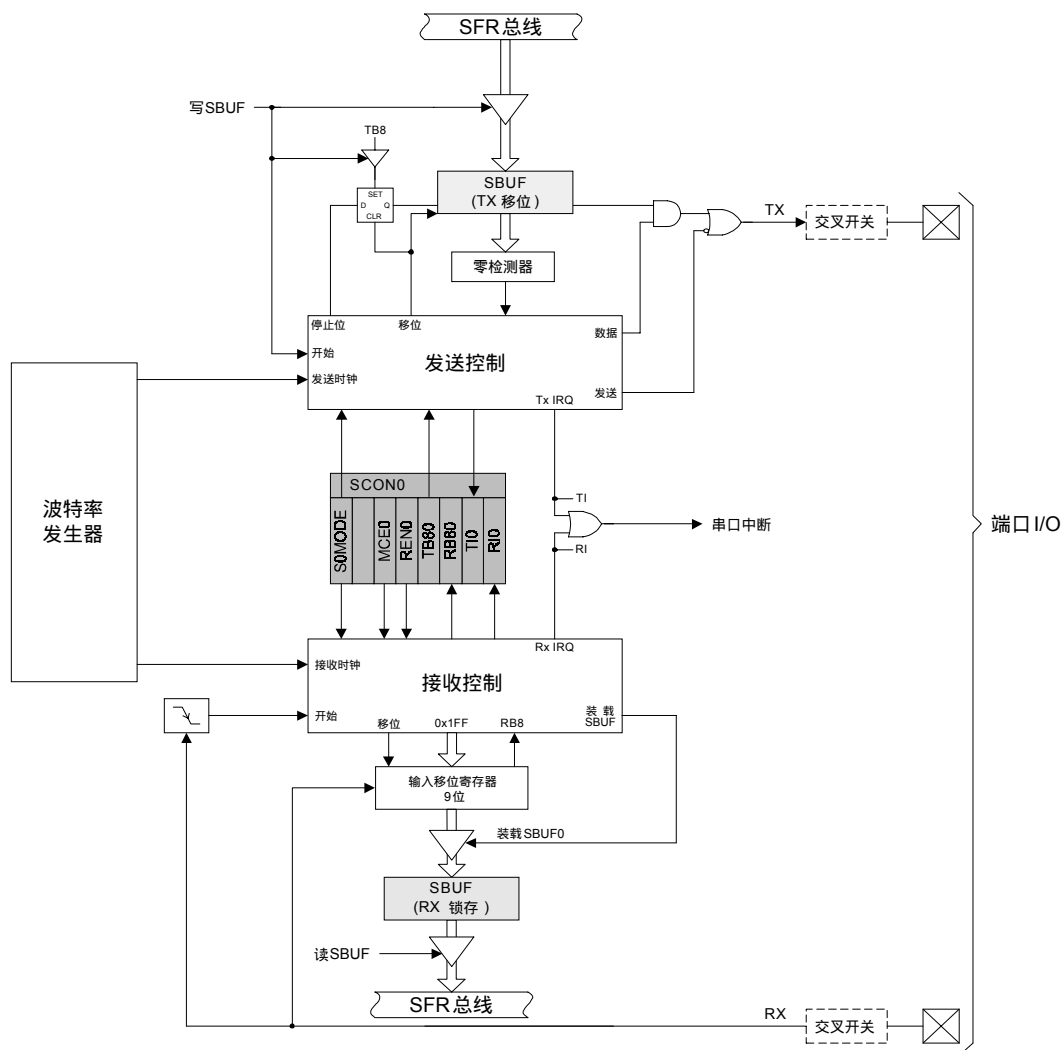
## 14. UART0

UART0 是一个异步、全双工串口，它提供标准 8051 串口的方式 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。读 SBUF0 时自动访问接收寄存器；写 SBUF0 时自动访问发送寄存器。

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’）时将产生一个中断。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 14.1 UART0 原理框图**

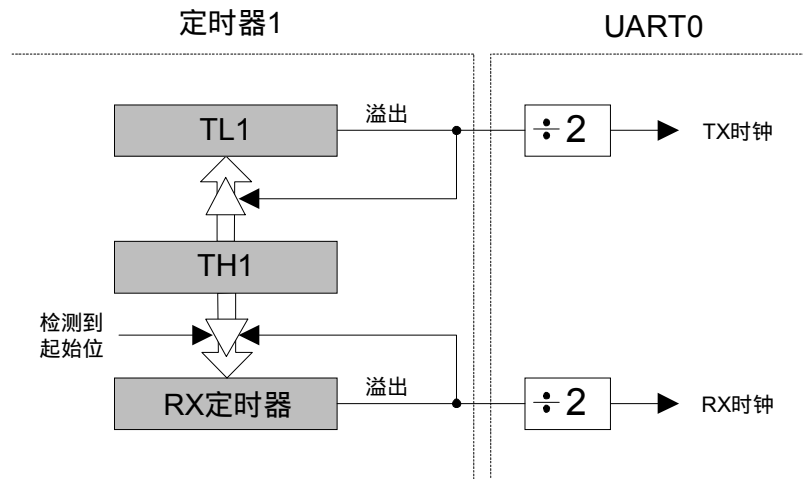




## 14.1 增强的波特率发生器

UART0 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 14.2 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 15.2 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。应将定时器 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 5 个时钟源中选择：SYSCLK、SYSCLK/4、SYSCLK/12、SYSCLK/48 和外部振荡器时钟/8。对于任何给定的定时器 1 时钟源，UART0 的波特率由方程 14.1 决定：

方程 14.1 UART0 波特率

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

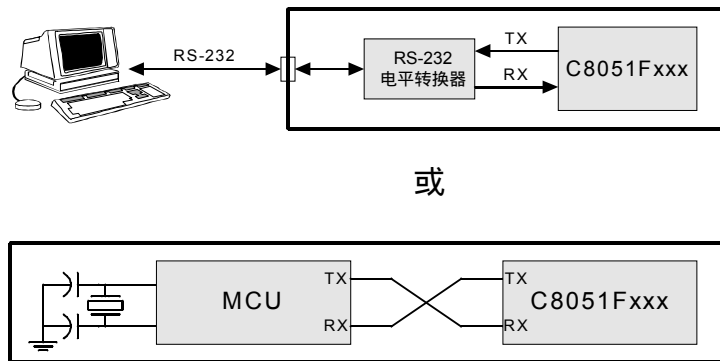
其中  $T1_{CLK}$  是定时器 1 的时钟频率， $TH$  是定时器 1 的高字节 (重载值)。

定时器 1 时钟频率的选择方法见“15.2 定时器 2”。表 14.1 – 14.6 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。

## 14.2 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 14.3 所示。

图 14.3 UART 连接图



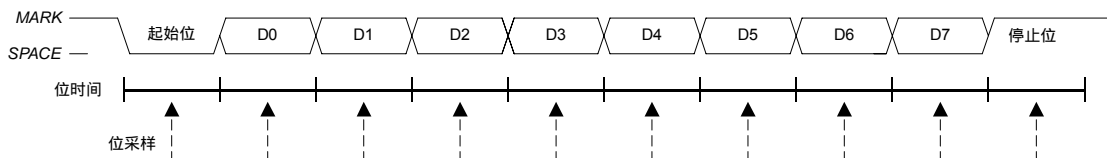
### 14.2.1 8 位 UART

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 TX 引脚发送，在 RX 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 14.4 8 位 UART 时序图

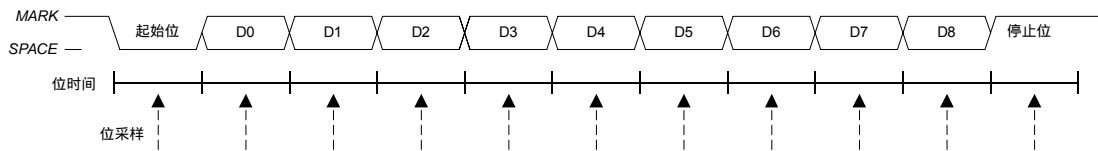


## 14.2.2 9 位 UART

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。在发送时，第九数据位由 TB80（SCON0.3）中的值决定，它可以被赋值为 PSW 中的奇偶位 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 14.5 9 位 UART 时序图



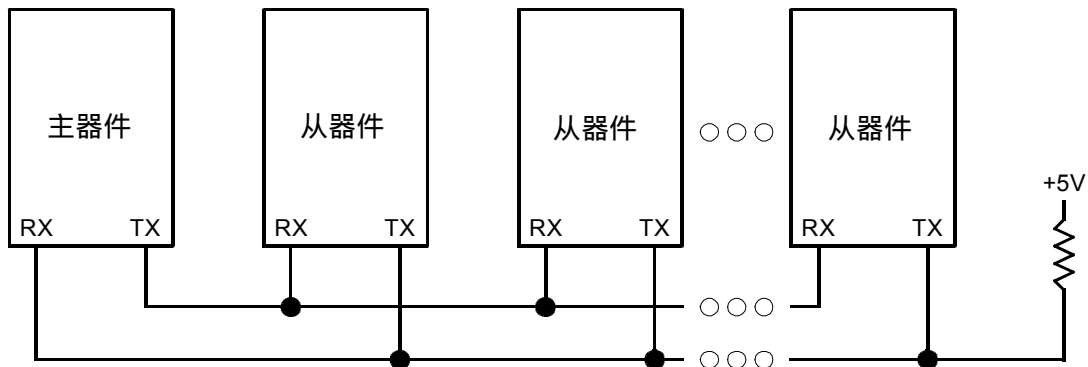
## 14.3 多机通信

9 位 UART 方式通过使用第九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。当主机想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：地址字节的第九位为逻辑‘1’；数据字节的第九位总是设置为逻辑‘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。第九位为逻辑‘1’表示接收到的是地址字节。在 UART 的中断处理程序中，软件将接收到的地址与从机自身的 8 位地址进行比较。如果地址匹配，从机将清除它的 MCE0 位以允许后面接收数据字节时产生中断。未被寻址的从机仍保持其 MCE0 位为‘1’，在收到后续的数据字节时不产生中断，从而忽略收到的数据。一旦接收完整个消息，被寻址的从机将它的 MCE0 位重新置‘1’以忽略所有的数据传输，直到它收到下一个地址字节。

可以将多个地址分配给一个从机，或将一个地址分配给多个从机从而允许同时向多个从机“广播”发送。主机可以被配置为接收所有的传输数据，或通过实现某种协议使主/从角色能临时变换以允许原来的主机和从机之间进行半双工通信。

图 14.6 UART 多机方式连接图



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 14.7 SCON0: UART0 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
S0MODE	-	MCE0	REN0	TB80	RB80	TI0	RI0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: (可位寻址) 0x98
<p>位 7: S0MODE: UART0 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 8 位 UART。</p> <p>位 6: 未使用。读 = 1b。写 = 忽略。</p> <p>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。 方式 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。 方式 1: 多处理器通信允许。 0: 第九位的逻辑电平被忽略。 1: 只有当第九位为逻辑 ‘1’ 时 RI0 才被置位并产生中断。</p> <p>位 4: REN0: 接收允许 该位允许/禁止 UART 接收器。 0: UART 接收禁止。 1: UART 接收允许。</p> <p>位 3: TB80: 第九发送位 该位的逻辑电平被赋值给 9 位 UART 方式的第九发送位。在 8 位 UART 方式中未用。跟需要用软件置 ‘1’ 或清 ‘0’。</p> <p>位 2: RB80: 第九接收位 该位被赋值为 9 位 UART 方式中第九数据位的值。在方式 0, 则 RB8 被赋值为停止位的值。</p> <p>位 1: TI0: 发送中断标志 当 UART 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’。</p> <p>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’。</p>								

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 14.9 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 14.1 对应标准波特率的定时器设置 (使用内部振荡器)

频率: 24.5MHz							
	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 (预分频选择)	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 14.2 对应标准波特率的定时器设置 (使用外部振荡器)

频率: 25MHz							
	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 (预分频选择)	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	SYSCLK/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 内部振荡器	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

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

表 14.3 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 (bps)	波特率误差 (%)	振荡器分频系数	定时器时钟源	SCA1-SCA0 (预分频选择)	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
SYSCLK 源自内部振荡器	230400	0.00%	96	EXTCLK/8	11	0	0xFA
	115200	0.00%	192	EXTCLK/8	11	0	0xF4
	57600	0.00%	384	EXTCLK/8	11	0	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 14.4 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 (%)	振荡器分频系数	定时器时钟源	SCA1-SCA0 (预分频选择)	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	0xFR
	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

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

表 14.5 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 (%)	振荡器 分频系数	定时器 时钟源	SCA1-SCA0 (预分频选择)	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	0xFD
	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 14.6 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 (预分频选择)	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



## 15. 定时器

C8051F30x 内部有 3 个 16 位计数器/定时器：其中 T0 和 T1 与标准 8051 中的计数器/定时器兼容，T2 是 16 位自动重载定时器，可用于 ADC、SMBus 或作为通用定时器使用。这些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 可以作为一个 16 位或两个 8 位自动重载定时器。

定时器 0 和定时器 1	定时器 2
13 位计数/定时器	16 位自动重载定时器
16 位计数器/定时器	
8 位自动重载的计数器/定时器	两个 8 位自动重载定时器
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 CKCON 寄存器中的定时器时钟选择位 (TIM-T0M) 和时钟预分频位 (SCA1-SCA0) 决定。时钟预分频位定义一个预分频时钟，作为定时器 0 和/或定时器 1 的时钟源 (见图 15.6)。

定时器 0 和定时器 1 可以被配置为使用预分频时钟或系统时钟。定时器 2 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引脚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。

### 15.1 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 (TCON) 寄存器用于允许定时器 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 TIM1-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。

#### 15.1.1 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 16.2 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)。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

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, 出现在所选输入引脚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 15.6) 中的时钟预分频位定义。

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 为定义, 见图 16.13), 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。

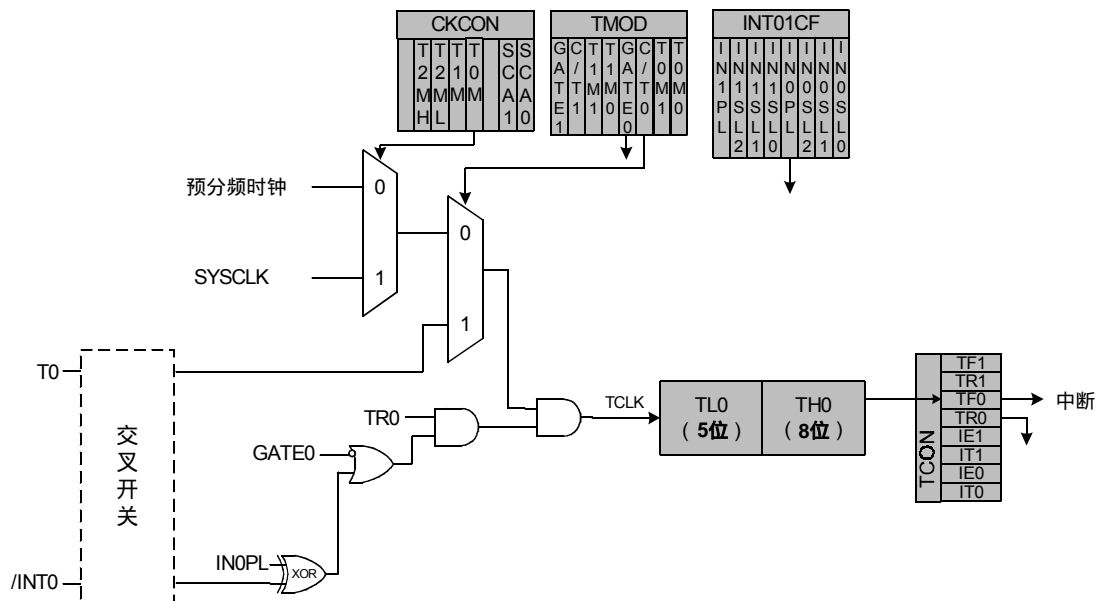
TR0	GATE0	IN0PL	/INT0	计数器/定时器
0	X	X	X	禁止
1	0	X	X	允许
1	1	0	0	禁止
1	1	0	1	允许
1	1	1	0	允许
1	1	1	1	禁止

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; 其有效电平由 INT01CF 寄存器中的 IN1PL 为定义。

图 15.1 T0 方式 0 原理框图



## 15.1.2 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。

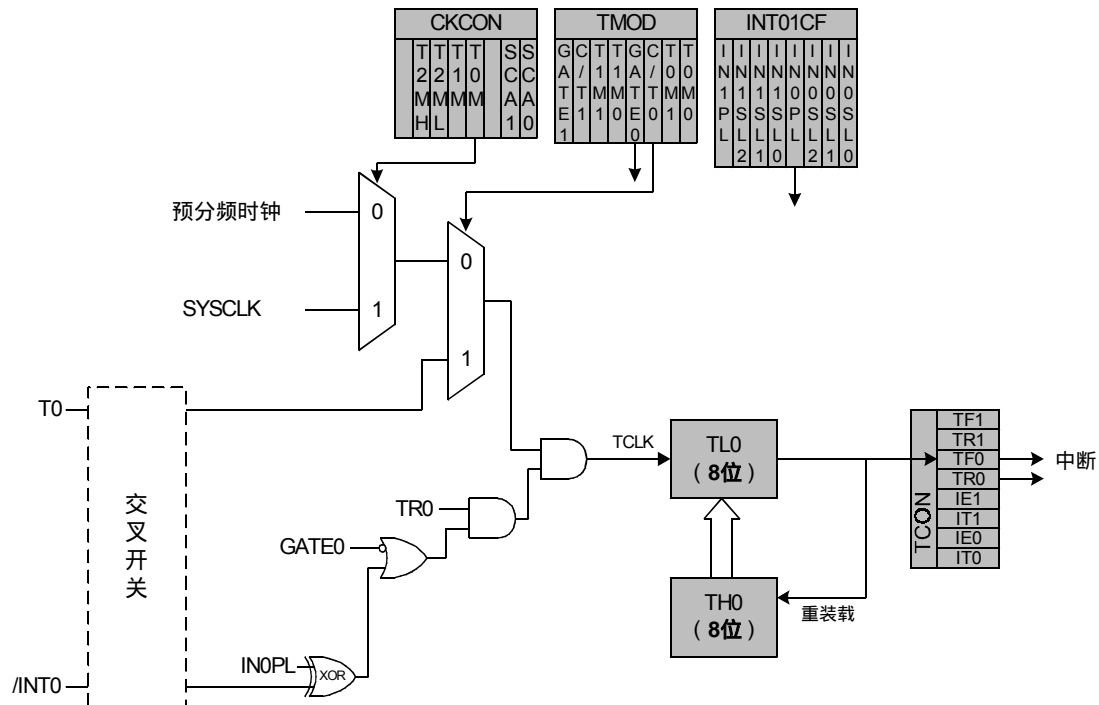
# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 15.1.3 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 15.2 T0 方式 2 原理框图



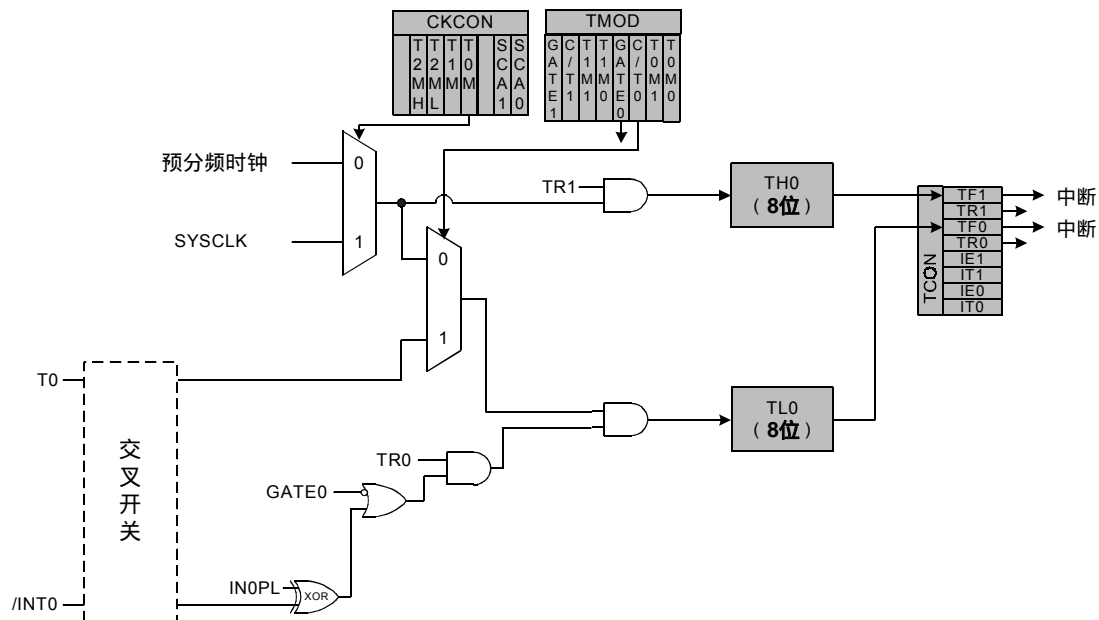
# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 15.1.4 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，由系统时钟或预分频时钟提供时基。TH0 使用定时器 1 的运行控制位 TR1，并在发生溢出时将定时器 1 的溢出标志位 TF1 置‘1’，所以它控制定时器 1 的中断。

定时器 1 在方式 3 时停止运行。在定时器 0 工作于方式 3 时，定时器 1 可以工作在方式 0、1 或 2，但不能用外部信号作为时钟，也不能设置 TF1 标志和产生中断。但是定时器 1 溢出可以用于为 SMBus 和/或 UART 产生波特率，也可以用于启动 ADC 转换。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 15.3 T0 方式 3 原理框图



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 15.4 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。							
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。							
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 IN01CF 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 IN01CF 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 IN01CF 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 IN01CF 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。							

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 15.5 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	TIM1	TIM0	GATE0	C/T0	T0M1	T0M0	00000000																														
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x89																														
<p>位 7: GATE1: 定时器 1 门控位            0: 当 TR1=1 时定时器 1 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。</p> <p>位 6: C/T1: 计数器/定时器 1 功能选择。            0: 定时器功能: 定时器 1 由 TIM 位 (CKCON.4) 定义的时钟加 1。            1: 计数器功能: 定时器 1 由外部输入引脚 (T1) 的负跳变加 1。</p> <p>位 5-4: TIM1-TIM0: 定时器 1 方式选择            这些位选择定时器 1 的工作方式。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>TIM1</th> <th>TIM0</th> <th>方式</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>方式 0: 13 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>方式 1: 16 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>方式 2: 自动重载的 8 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>方式 3: 定时器 1 停止运行</td> </tr> </tbody> </table> <p>位 3: GATE0: 定时器 0 门控位            0: 当 TR0=1 时定时器 0 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。</p> <p>位 2: C/T0: 计数器/定时器 0 功能选择。            0: 定时器功能: 定时器 0 由 T0M 位 (CKCON.3) 定义的时钟加 1。            1: 计数器功能: 定时器 0 由外部输入引脚 (T0) 的负跳变加 1。</p> <p>位 1-0: T0M1-T0M0: 定时器 0 方式选择            这些位选择定时器 0 的工作方式。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>T0M1</th> <th>T0M0</th> <th>方式</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>方式 0: 13 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>方式 1: 16 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>方式 2: 自动重载的 8 位计数器/定时器</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>方式 3: 两个 8 位计数器/定时器</td> </tr> </tbody> </table>									TIM1	TIM0	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	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
TIM1	TIM0	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																																				
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																																				

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 15.6 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															
-	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															
<p>位 7: 未用。读 = 0b, 写 = 忽略。</p> <p>位 6: 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)。 T2 工作在 16 位方式时该位被忽略。 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。</p> <p>位 5: 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。</p> <p>位 4: 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 使用由预分频位 (SCA1-SAC0) 定义的时钟。 1: 定时器 1 使用系统时钟</p> <p>位 3: T0M: 定时器 0 时钟选择。 该位选择定时器 0 的时钟源。当 C/T0 被设置为逻辑 '1' 时, T0M 被忽略。 0: 定时器 1 使用由预分频位 (SCA1-SAC0) 定义的时钟。 1: 计数器/定时器使用系统时钟</p> <p>位 2: 未用。读 = 0b, 写 = 忽略。</p> <p>位 1-0: SCA1-SCA0: 定时器 0/1 预分频位 如果定时器 0/1 被配置为使用预分频时钟, 则这些位控制时钟分频数。</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">SCA1</th> <th style="width: 10%;">SCA0</th> <th style="width: 80%;">预分频时钟</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">系统时钟/12</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">系统时钟/4</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">系统时钟/48</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">外部时钟/8</td> </tr> </tbody> </table>									SCA1	SCA0	预分频时钟	0	0	系统时钟/12	0	1	系统时钟/4	1	0	系统时钟/48	1	1	外部时钟/8
SCA1	SCA0	预分频时钟																					
0	0	系统时钟/12																					
0	1	系统时钟/4																					
1	0	系统时钟/48																					
1	1	外部时钟/8																					

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 15.7 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 15.8 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 15.9 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 15.10 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。



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 15.2 定时器 2

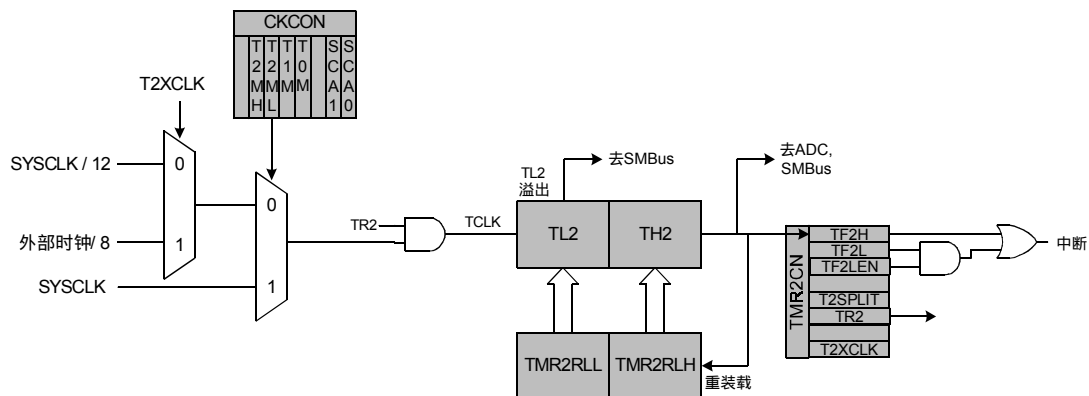
定时器 2 是一个 16 位的计数器/定时器，由两个 8 位的 SFR 组成：TL2（低字节）和 TH2（高字节）。定时器可以工作在 16 位自动重载方式或 8 位自动重载方式（两个 8 位定时器）。T2SPLIT 位（TMR2CN.3）定义定时器 2 的工作方式。

定时器 2 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。

### 15.2.1 16 位自动重载方式

当 T2SPLIT 位（TMR2CN.3）被设置为逻辑‘0’时，定时器 2 工作在自动重载的 16 位定时器方式（见图 15.11）。定时器 2 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 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，每次溢出都将产生一个中断。如果定时器 2 中断被允许并且 TF2LEN 位（TMR2CN.5）被置‘1’，则每次低 8 位（TL2）溢出时（从 0xFF 到 0x00）将产生一个中断。

图 15.11 定时器 2 的 16 位方式原理框图



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 15.2.2 8 位自动重载定时器方式

当 T2SPLIT 位 (TMR2CN.3) 被置 ‘1’ 时, 定时器 2 工作双 8 位定时器方式。两个 8 位定时器 TH2 和 TL2 都工作在自动重载方式 (见图 15.12)。TMR2RLL 保持 TL2 的重载值, 而 TMR2RLH 保持 TH2 的重载值。TMR2CN 中的 TR2 是 TH2 的运行控制位。当定时器 2 被配置为 8 位方式时, TL2 总是处于运行状态。

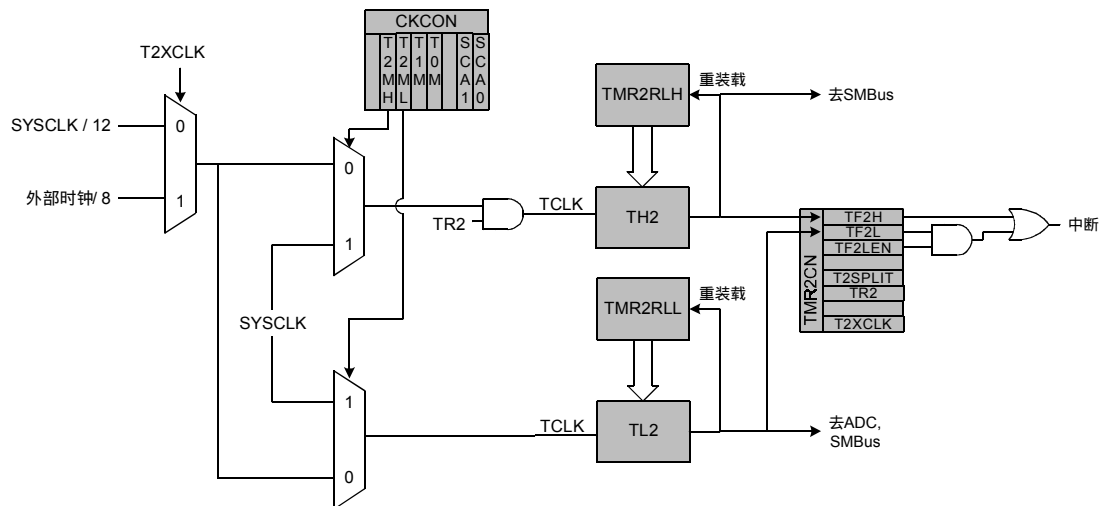
每个 8 位定时器都可以被配置为使用 SYSCLK、SYSCLK/12 或外部振荡器时钟/8 作为其时钟源。定时器 2 时钟选择位 T2MH 和 T2ML (位于 CKCON 中) 分别用于选择 T2H 和 T2L 的时钟源。当时钟选择位为 ‘1’ 时, 时钟源为 SYSCLK; 当时钟选择位为 ‘0’ 时, 时钟源由定时器 2 外部时钟选择位 (TMR2CN 中的 T2XCLK) 定义。时钟源的选择情况如下所示。

T2MH	T2XCLK	TH2 时钟源
0	0	SYSCLK/12
0	1	外部时钟/8
1	X	SYSCLK

T2ML	T2XCLK	TL2 时钟源
0	0	SYSCLK/12
0	1	外部时钟/8
1	X	SYSCLK

当 TH2 发生溢出时 (从 0xFF 到 0x00), TF2H 被置 ‘1’; 当 TL2 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 TH2 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 TL2 或 TH2 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 15.12 T2 方式 2 原理框图



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

图 15.13 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	-	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
<p>位 7: TF2H: 定时器 2 高字节溢出标志 在 16 位方式, 当定时器 2 发生溢出时 (从 0xFFFF 到 0x0000) 由硬件置 '1'; 在 8 位方式, 当定时器 2 高字节发生溢出时 (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。</p> <p>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。</p> <p>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 (IE.5), 则当定时器 2 低字节发生溢出时将产生一个中断。当定时器 2 工作在 16 位方式时, 该位应被清 '0'。 0: 禁止定时器 2 低字节中断。 1: 允许定时器 2 低字节中断。</p> <p>位 4: 未用。读 = 0b, 写 = 忽略。</p> <p>位 3: T2SPLIT: 定时器 2 双 8 位方式允许位 当该位被置 '1' 时, 定时器 2 工作在双 8 位自动重装载定时器方式。 0: 定时器 2 工作在 16 位自动重装载方式。 1: 定时器 2 工作在双 8 位自动重装载定时器方式。</p> <p>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 TH2, TL2 总是处于运行状态。 0: 定时器 2 禁止。 1: 定时器 2 允许。</p> <p>位 1: 未用。读 = 0b, 写 = 忽略。</p> <p>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。</p>								

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 15.14 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
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址: 0xCA

位 7-0: TMR2RLL: 定时器 2 重载寄存器的低字节  
TMR2RLL 保存定时器 2 重载值的低字节。

**图 15.15 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
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址: 0xCB

位 7-0: TMR2RLH: 定时器 2 重载寄存器的高字节  
TMR2RLH 保存定时器 2 重载值的高字节。

**图 15.16 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
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								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 TL2 的计数值。

**图 15.17 TH2: 定时器 2 高字节**

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

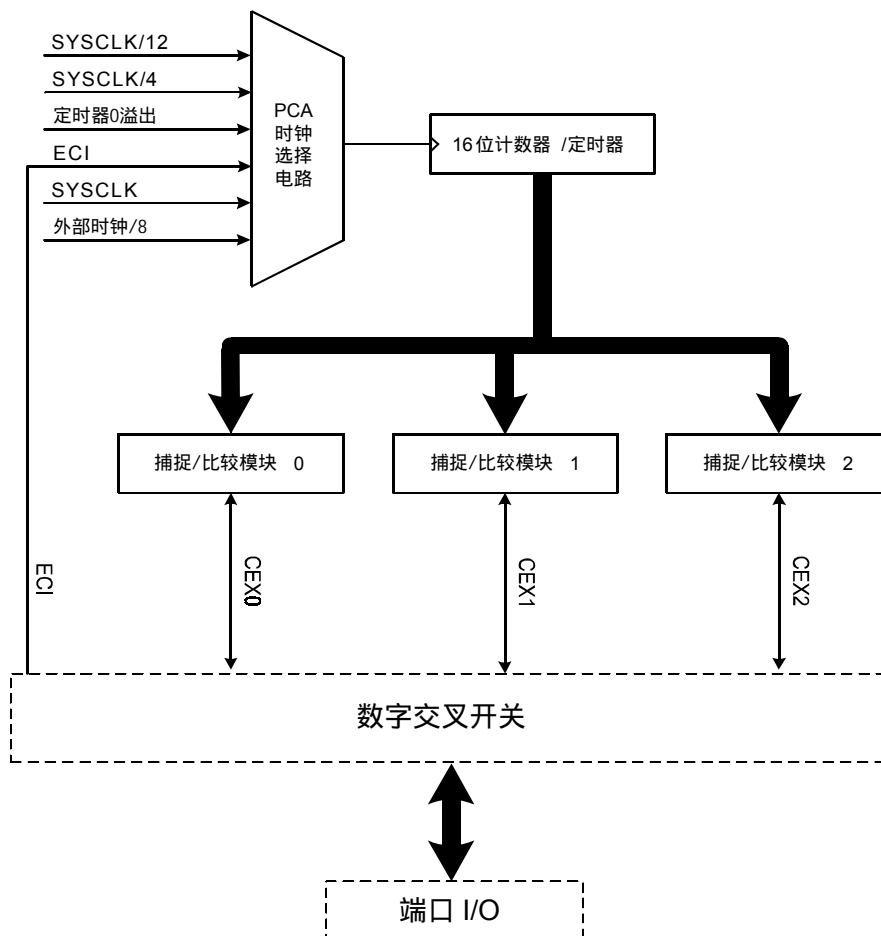
位 7-0: TMR2H: 定时器 2 的高字节  
在 16 位方式, TMR2H 寄存器保持 16 位定时器 2 的高字节。在 8 位方式, TMR2H 中保持 8 位高字节定时器 TH2 的计数值。

## 16. 可编程计数器阵列

可编程计数器阵列（PCA0）提供增强的定时器功能，与标准8051的计数器/定时器相比，它需要较少的CPU干预。PCA由一个专用的16位计数器/定时器和3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<sub>n</sub>），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16.1。

注意：PCA的模块2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16.3节。

图 16.1 PCA 原理框图



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 16.1 PCA 计数器/定时器

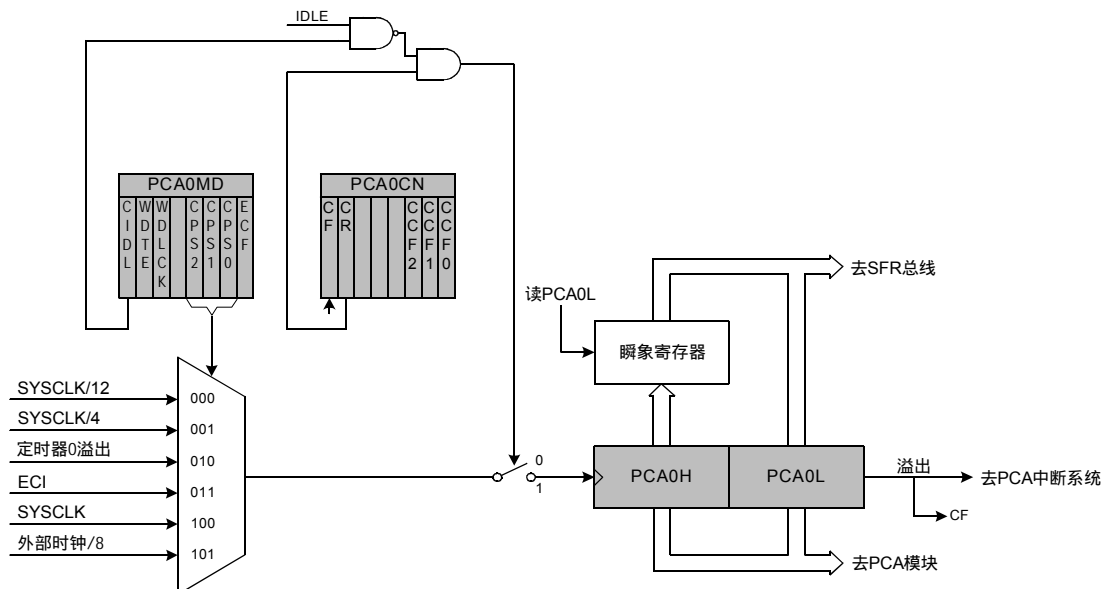
16 位的 PCA 计数器/定时器由两个 8 位的 SFR 组成：PCA0L 和 PCA0H。PCA0H 是 16 位计数器/定时器的高字节 (MSB)，而 PCA0L 是低字节 (LSB)。在读 PCA0L 时，“瞬象寄存器”自动锁存 PCA0H 的值，随后读 PCA0H 时将访问这个“瞬象寄存器”而不是 PCA0H 本身。先读 PCA0L 寄存器可以保证正确读取整个 16 位 PCA 计数器的值。读 PCA0H 或 PCA0L 不影响计数器工作。PCA0MD 寄存器中的 CPS2-CPS0 位用于选择 PCA 计数器/定时器的时基，如表 16.1 所示。

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

表 16.1 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

图 16.2 PCA 计数器/定时器原理框图



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 16.2 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。

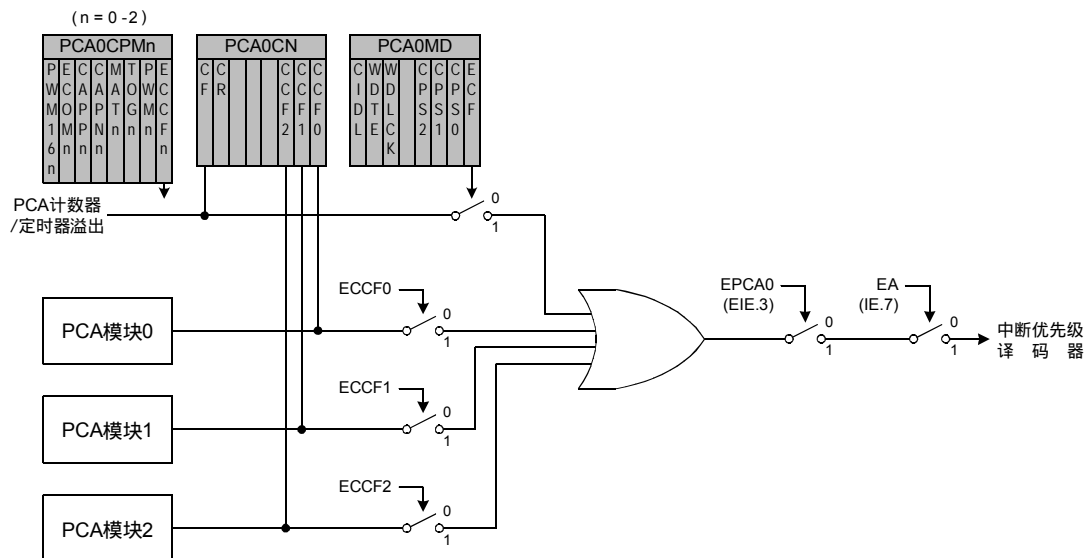
PCA0CPMn寄存器用于配置PCA捕捉/比较模块的工作方式，表16.2概述了模块工作在不同方式时该寄存器各位的设置情况。置‘1’PCA0CPMn寄存器中的ECCFn位将允许模块的CCFn中断。注意：要使单个的CCFn中断得到响应，必须先整体允许PCA0中断。通过将EA位（IE.7）和EPCA0位（EIE1.3）设置为逻辑‘1’来整体允许PCA0中断。PCA0中断配置的详细信息见图16.3。

表 16.2 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

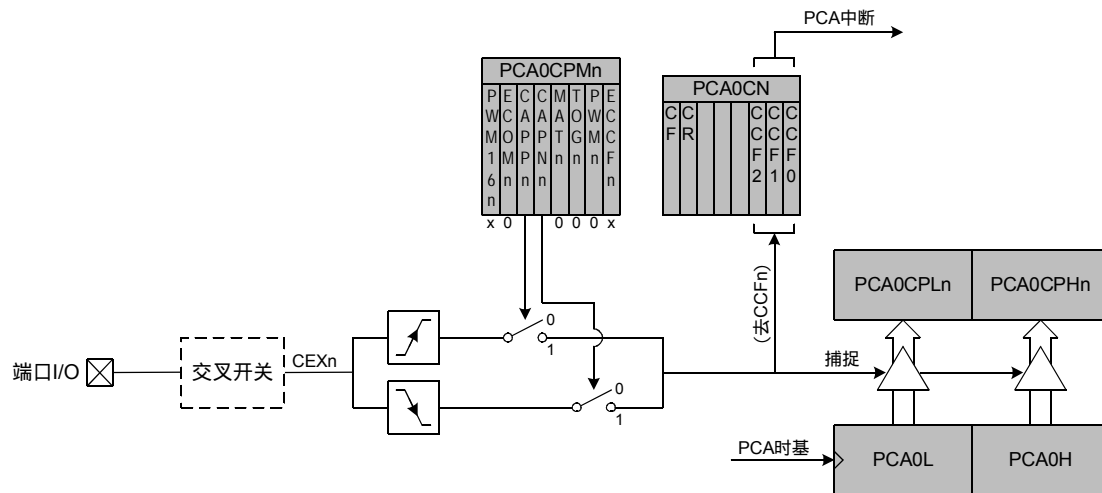
图 16.3 PCA 中断原理框图



## 16.2.1 边沿触发的捕捉方式

在该方式，CEX<sub>n</sub>引脚上出现的电平跳变导致PCA捕捉PCA计数器/定时器的值并将其装入到对应模块的16位捕捉/比较寄存器（PCA0CPL<sub>n</sub>和PCA0CPH<sub>n</sub>）。PCA0CPM<sub>n</sub>寄存器中的CAPP<sub>n</sub>和CAPN<sub>n</sub>位用于选择触发捕捉的电平变化类型：低电平到高电平（正沿）、高电平到低电平（负沿）或任何变化（正沿或负沿）。当捕捉发生时，PCA0CN中的捕捉/比较标志（CCF<sub>n</sub>）被置为逻辑‘1’并产生一个中断请求（如果CCF中断被允许）。当CPU转向中断服务程序时，CCF<sub>n</sub>位不能被硬件自动清除，必须用软件清0。如果CAPP<sub>n</sub>和CAPN<sub>n</sub>位都被设置为逻辑‘1’，可以通过直接读CEX<sub>n</sub>对应引脚的状态来确定本次捕捉是由上升沿触发还是由下降沿触发。

图16.4 PCA捕捉方式原理框图



注意：CEX<sub>n</sub> 输入信号必须在高电平或低电平器间至少保持两个系统时钟周期，以保证能够被硬件识别。



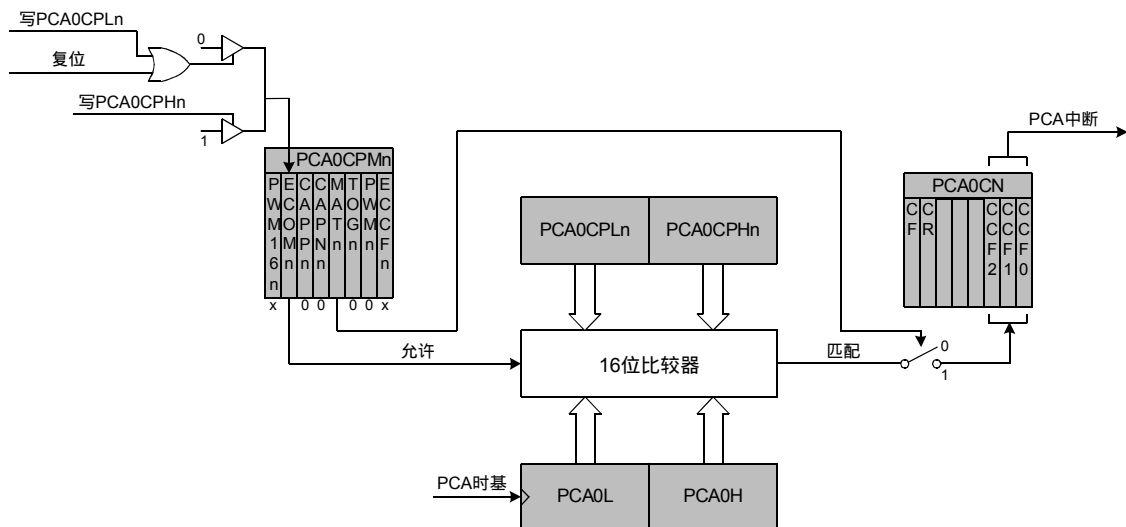
# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 16.2.2 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位将允许软件定时器方式。

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 16.5 PCA 软件定时器方式原理框图

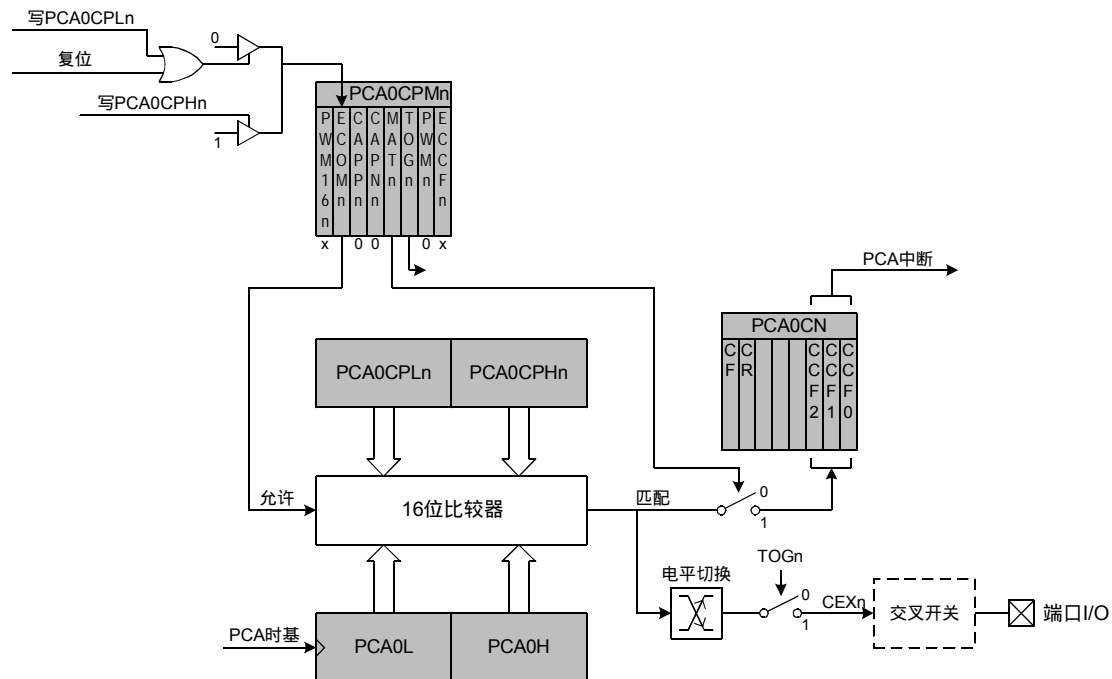


## 16.2.3 高速输出方式

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

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 16.6 PCA 高速输出方式原理框图**



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 16.2.4 频率输出方式

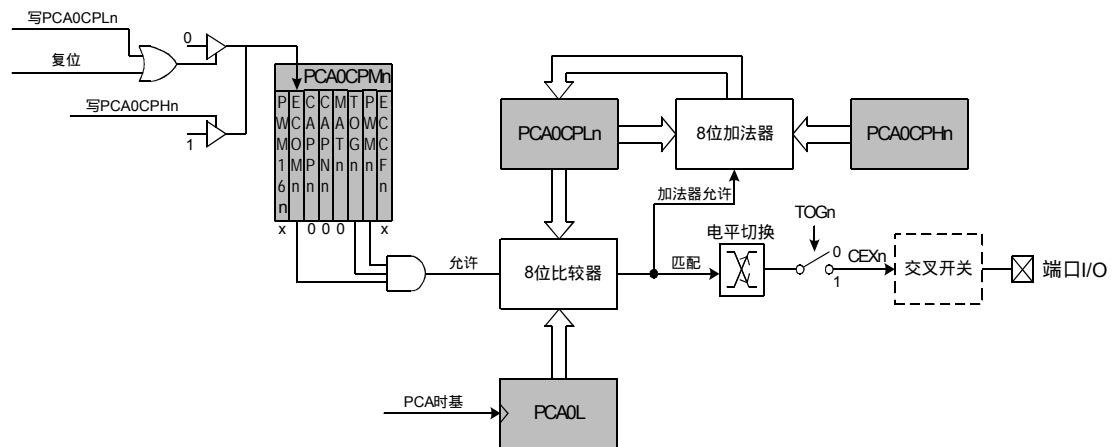
频率输出方式可在CEXn引脚产生可编程频率的方波。捕捉/比较模块的高字节保持输出电平改变前要计的PCA时钟数。所产生的方波的频率由方程16.1定义：

方程16.1 方波输出频率

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

其中： $F_{PCA}$ 是由PCA方式寄存器（PCA0MD）中的CPS2-0位选择的PCA时钟的频率。捕捉/比较模块的低字节与PCA0计数器的低字节比较；两者匹配时，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’来允许频率输出方式。

图 16.7 PCA 频率输出方式原理框图



## 16.2.5 8 位脉宽调制器方式

每个模块都可以被独立地用于在CEX<sub>n</sub>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<sub>n</sub>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<sub>n</sub>中的值相等时，CEX<sub>n</sub>引脚上的输出被置‘1’；当PCA0L中的计数值溢出时，CEX<sub>n</sub>输出被复位（见图16.8）。当计数器/定时器的低字节PCA0L溢出时（从0xFF到0x00），保存在PCA0CPH<sub>n</sub>中的值被自动装入到PCA0CPL<sub>n</sub>，不需软件干预。通过将PCA0CPM<sub>n</sub>寄存器中的ECOM<sub>n</sub>和PWM<sub>n</sub>位置‘1’来允许8位脉冲宽度调制器方式。8位PWM方式的占空比由方程16.2给出。

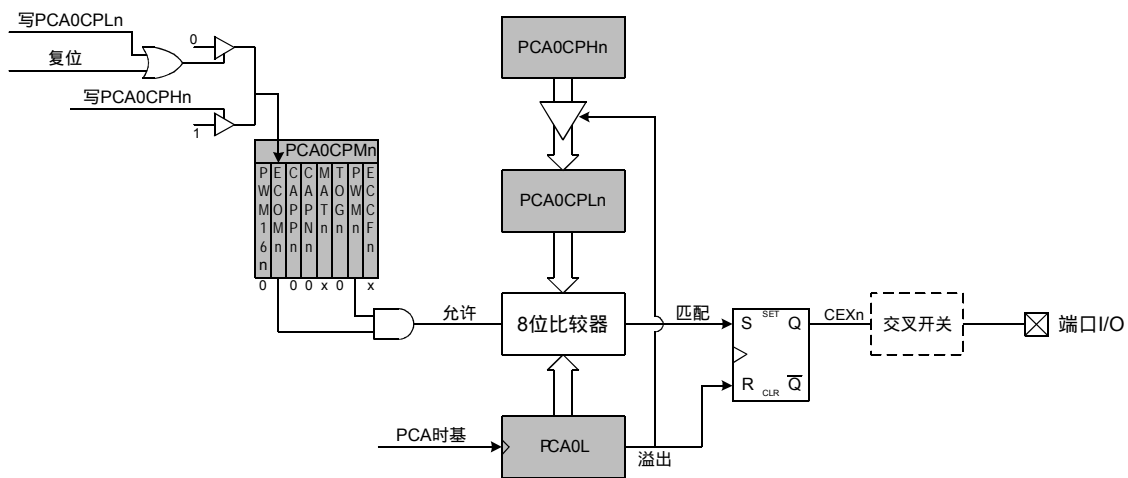
**注意：**当向PCA0的捕捉/比较寄存器写入一个16位数值时，应先写低字节。向PCA0CPL<sub>n</sub>的写入操作将清‘0’ ECOM<sub>n</sub>位；向PCA0CPH<sub>n</sub>写入时将置‘1’ ECOM<sub>n</sub>位。

**方程16.2 8位PWM占空比**

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

由方程16.2可知，最大占空比为100%（PCA0CPH<sub>n</sub> = 0），最小占空比为0.39%（PCA0CPH<sub>n</sub> = 0xFF）。可以通过清‘0’ ECOM<sub>n</sub>位产生0%的占空比。

**图 16.8 PCA 的 8 位 PWM 方式原理框图**



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 16.2.6 16 位脉宽调制器方式

PCA 模块还可被配置为工作在 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，CEX<sub>n</sub> 的输出被置为高电平；当计数器溢出时，CEX<sub>n</sub> 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<sub>n</sub> 匹配中断同步。通过将 CA0CPM<sub>n</sub> 寄存器中的 ECOM<sub>n</sub>、PWM<sub>n</sub> 和 PWM16<sub>n</sub> 位置 ‘1’ 来允许 16 位 PWM 方式。为了得到可变的占空比，应允许匹配中断（CCF<sub>n</sub> = 1 并且 MAT<sub>n</sub> = 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 16.3 给出。

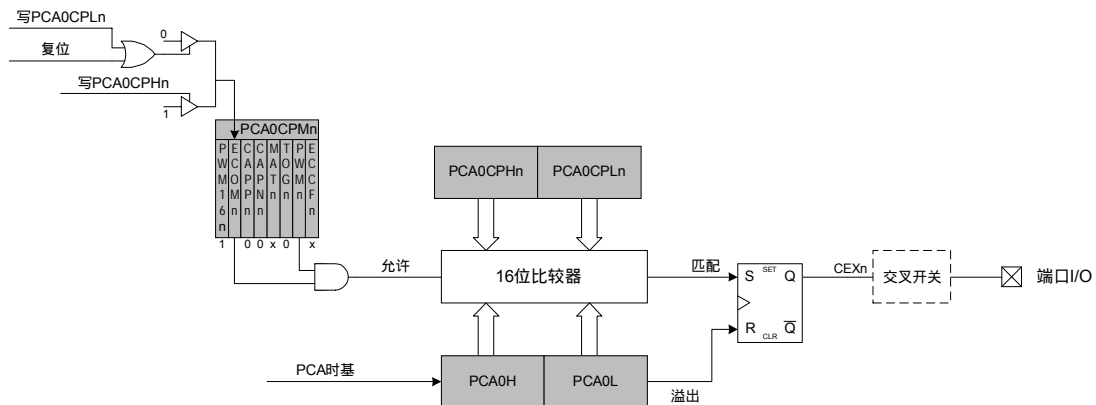
**注意：**当向 PCA 的捕捉/比较寄存器写入一个 16 位数值时，应先写低字节。向 PCA0CPL<sub>n</sub> 的写入操作将清 ‘0’ ECOM<sub>n</sub> 位；向 PCA0CPH<sub>n</sub> 写入时将置 ‘1’ ECOM<sub>n</sub> 位。

### 方程 16.3 16 位 PWM 占空比

$$\text{占空比} = \frac{(65536 - \text{PCA0CP}_n)}{65536}$$

由方程 16.3 可知，最大占空比为 100%（PCA0CP<sub>n</sub> = 0），最小占空比为 0.0015%（PCA0CP<sub>n</sub> = 0xFFFF）。可以通过将 ECOM<sub>n</sub> 位清 ‘0’ 产生 0% 的占空比。

图 17.9 PCA 的 16 位 PWM 方式原理框图



## 16.3 看门狗定时器方式

通过 PCA 的模块 2 可以实现可编程看门狗定时器 (WDT) 功能。如果两次对 WDT 更新寄存器 (PCA0CPH2) 的写操作相隔的时间超过规定的极限, WDT 将产生一次复位。可以根据需要用软件配置和允许/禁止 WDT。

当 PCA0MD 寄存器中的 WDTE 位被置 ‘1’ 时, 模块 2 被作为看门狗定时器 (WDT) 使用。模块 2 高字节与 PCA 计数器的高字节比较; 模块 2 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。**

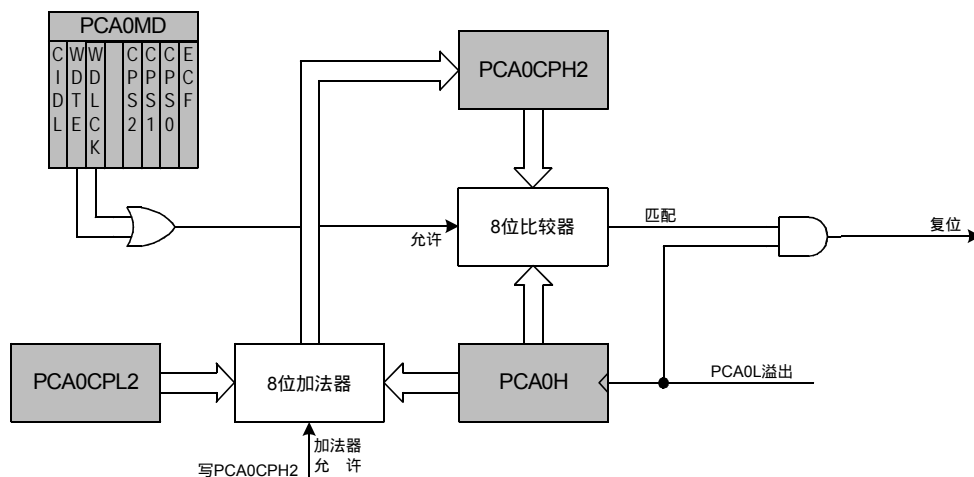
### 16.3.1 看门狗定时器操作

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 2 被强制进入软件定时器方式。
- 对模块 2 方式寄存器 (PCA0CPM2 ) 的写操作被禁止。

当 WDT 被允许时, 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 被允许, 但用户软件没有允许 PCA 计数器, 则读 PCA 运行控制 (CR) 位时将返回 ‘0’。如果在 WDT 被允许时 PCA0CPH2 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 PCA0CPH2 来更新 WDT (写入值可以是任意值)。在写 PCA0CPH2 时, PCA0H 的值加上 PCA0CPL2 中保存的偏移值后被装入到 PCA0CPH2 (见图 16.10)。

图 16.10 PCA 模块 2 的看门狗定时器方式



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

保存在 PCA0CPH2 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 16.4 给出, 其中 *PCA0L* 是执行更新操作时 PCA0L 寄存器的值。

## 方程16.4 看门狗定时器偏移值 (PCA时钟数)

$$\text{偏移值} = (256 \times \text{PCA0CPL2}) + (256 - \text{PCA0L})$$

当 PCA0L 发生溢出并且 PCA0CPH2 和 PCA0H 匹配时, WDT 将产生一次复位。在 WDT 被允许的情况下, 软件可以通过向 CCF2 标志 (PCA0CN.2) 写 ‘1’ 来强制产生 WDT 复位。

### 16.3.2 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 PCA0CPL2 装入所希望的 WDT 更新偏移值。
- 配置 PCA 的等待方式位 (如果希望在 CPU 处于等待方式时 WDT 停止工作, 则应将 CIDL 位置 ‘1’)。
- 通过向 WDTE 位写 ‘1’ 来使能 WDT。

在 WDT 被使能时, 不能改变 PCA 时钟源和等待方式的设置值。通过向 PCA0MD 寄存器的 WDTE 或 WDCLK 位写 ‘1’ 来允许 WDT。当 WDCLK 被置 ‘1’ 时, 在发生下一次系统复位之前将不能禁止 WDT。如果 WDCLK 未被置 ‘1’, 清除 WDTE 位将禁止 WDT。

WDT 在任何一次系统复位之后都被设置为允许状态。PCA0 计数器的缺省时钟为系统时钟的十二分频 (SYSCLK/12)。PCA0L 和 PCA0CPL2 的缺省值均为 0x00, 因此 WDT 的超时间隔为 256 个系统时钟。表 16.3 列出了对应某些典型系统时钟频率的超时间隔。

表 16.3 看门狗定时器超时间隔\*

系统时钟 (Hz)	PCA0CPL2	超时间隔 (ms)
24,500,000	255	31.97
24,500,000	128	16.05
24,500,000	32	4.01
18,432,000	255	42.50
18,432,000	128	21.33
18,432,000	32	5.33
11,059,200	255	70.83
11,059,200	128	35.56
11,059,200	32	8.89
3,060,000**	255	256
3,060,000**	128	128.5
3,060,000**	32	32.1
32,000	255	24480.0
32,000	128	12288.0
32,000	32	3072.0

\*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。

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 16.4 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 16.11 PCA0CN: PCA 控制寄存器**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CF	CR	-	-	-	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 5-3:	未用。读 = 000b, 写 = 忽略。							
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 CCF 中断服务程序。该位不能由硬件自动清 0, 必须用软件清 0。							
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。							



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 16.12 PCA0MD: PCA 方式寄存器**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值																																				
CIDL	WDTE	WDCLK	-	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																																				
<p>位 7:      CIDL: PCA 计数器/定时器等待控制                      设置 CPU 等待方式下的 PCA 工作方式。                      0: 当系统控制器处于等待方式时, PCA 继续正常工作。                      1: 当系统控制器处于等待方式时, PCA 停止工作。</p> <p>位 6:      WDTE: 看门狗定时器允许位                      如果该位被置 '1', PCA 模块 2 被用作看门狗定时器。                      0: 看门狗定时器被禁止。                      1: PCA 模块 2 被用作看门狗定时器。</p> <p>位 5:      WDCLK: 看门狗定时器锁定                      该位对看门狗定时器允许位锁定/解锁。当 WDCLK 被置 '1' 时, 在发生下一次系统复位之前将不能禁止 WDT。                      0: 对看门狗定时器允许位解锁。                      1: 锁定看门狗定时器允许位。</p> <p>位 4:      未用。读 = 0b, 写 = 忽略。</p> <p>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。</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 10px 0;"> <thead> <tr> <th style="width: 12.5%;">CPS2</th> <th style="width: 12.5%;">CPS1</th> <th style="width: 12.5%;">CPS0</th> <th style="width: 62.5%;">时钟源</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>系统时钟的 12 分频</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>系统时钟的 4 分频</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>定时器 0 溢出</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>ECl 负跳变 (最大速率 = 系统时钟频率/4)</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>系统时钟</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>外部时钟的 8 分频</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>保留</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>保留</td> </tr> </tbody> </table> <p>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。</p>									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	ECl 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
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	ECl 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																																									

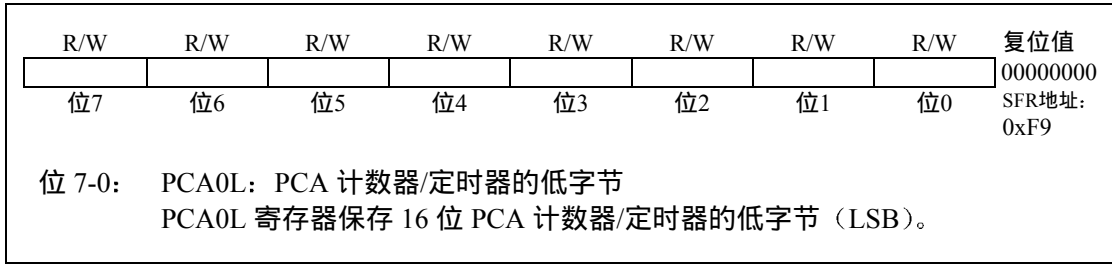
# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 16.13 PCA0CPMn: PCA 捕捉/比较寄存器**

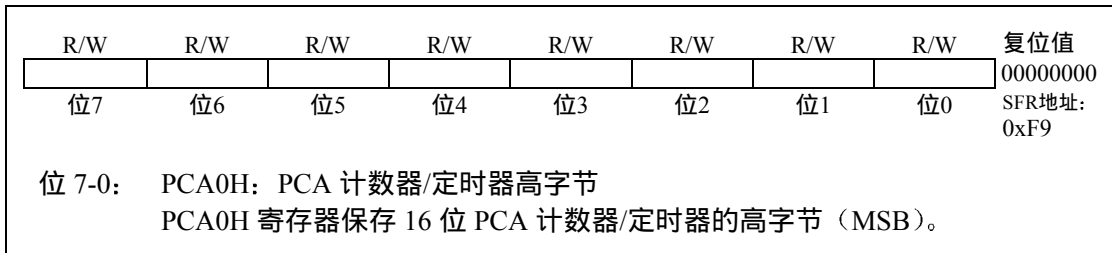
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xDA, 0xDB, 0xDC
PCA0CPMn 地址:   PCA0CPM0=0xDA(n=0) PCA0CPM1=0xDB(n=1) PCA0CPM2=0xDC(n=2)								
位 7:   PWM16n: 16 位脉冲宽度调制允许 当脉冲宽度调制方式被允许时 (PWMn = 1), 该位选择 16 位方式。 0: 选择 8 位 PWM。 1: 选择 16 位 PWM。								
位 6:   ECOMn: 比较器功能允许 该位允许/禁止 PCA 模块 n 的比较器功能。 0: 禁止。 1: 允许。								
位 5:   CAPPn: 正沿捕捉功能允许 该位允许/禁止 PCA 模块 n 的正边沿捕捉。 0: 禁止。 1: 允许。								
位 4:   CAPNn: 负沿捕捉功能允许 该位允许/禁止 PCA 模块 n 的负边沿捕捉。 0: 禁止。 1: 允许。								
位 3:   MATn: 匹配功能允许 该位允许/禁止 PCA 模块 n 的匹配功能。如果被允许, 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: 允许。								
位 2:   TOGn: 电平切换功能允许 该位允许/禁止 PCA 模块 n 的电平切换功能。如果被允许, 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: 允许。								
位 1:   PWMn: 脉宽调制方式允许 该位允许/禁止 PCA 模块 n 的 PWM 功能。当被允许时, 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。如果 TOGn 位也被置 '1', 模块将工作在频率输出方式。 0: 禁止。1: 允许。								
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。								

# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

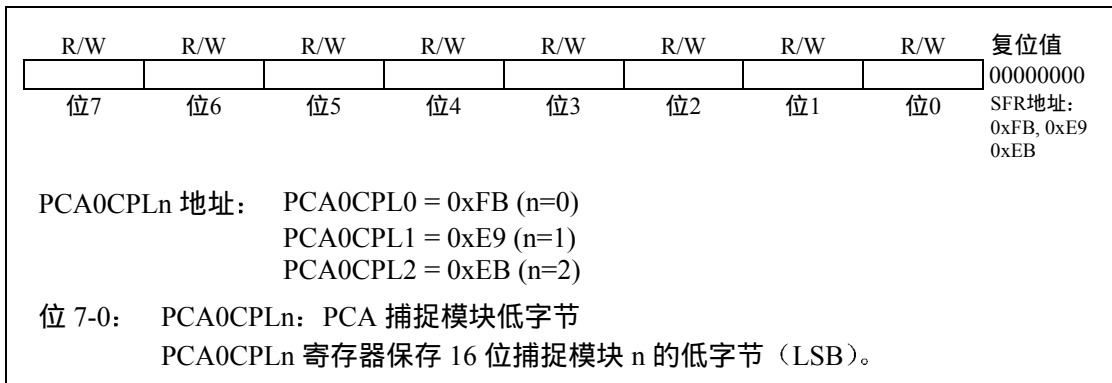
**图 16.14 PCA0L: PCA 计数器 / 定时器低字节**



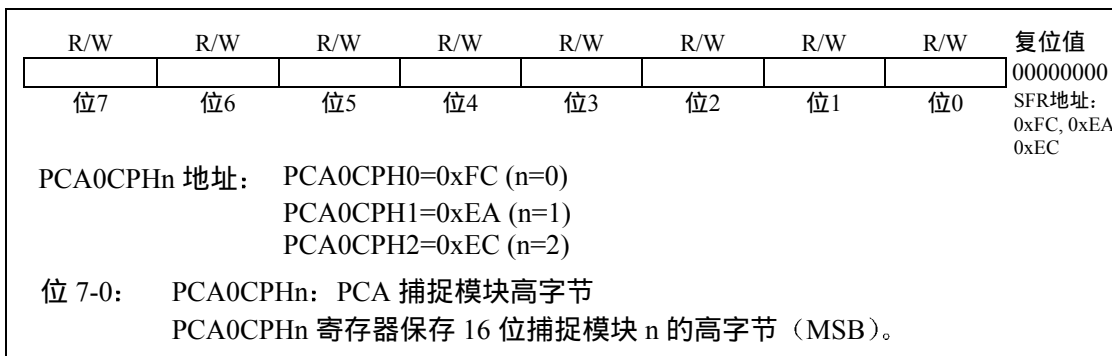
**图 16.15 PCA0H: PCA 计数器 / 定时器高字节**



**图 16.16 PCA0CPLn: PCA 捕捉模块低字节**



**图 16.17 PCA0CPHn: PCA 捕捉模块高字节**



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

## 17. C2 接口

C8051F300/1/2/3/4/5 有一个 Cygnal 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 接口与 JTAG 类似, 只是它将三个 JTAG 数据信号 (TDI、TDO、TMS) 映射到一个双向的 C2 数据信号 (C2D)。有关 C2 协议的详细信息见 C2 接口规范。

### 17.1 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 17.1 C2ADD: C2 地址寄存器

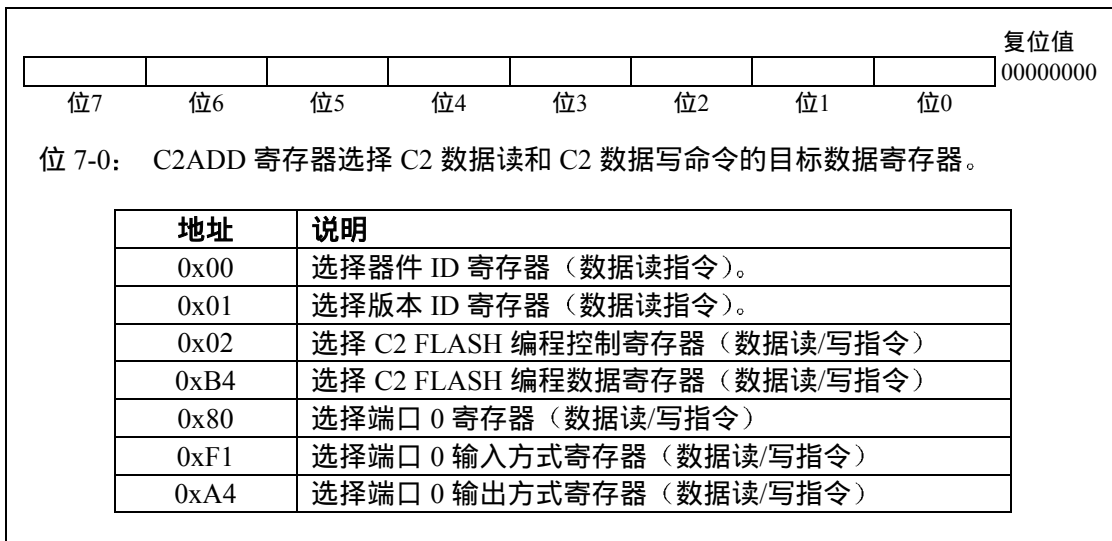


图 17.2 DEVICEID: C2 器件 ID 寄存器

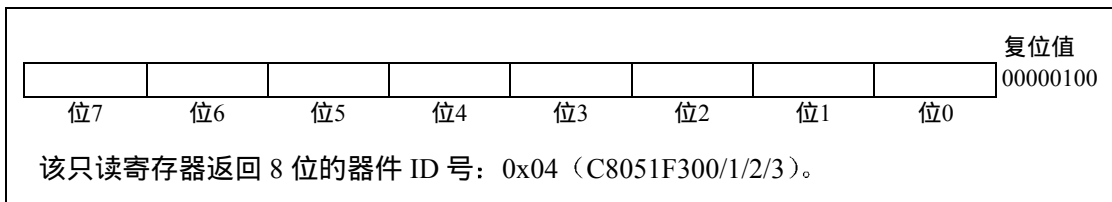
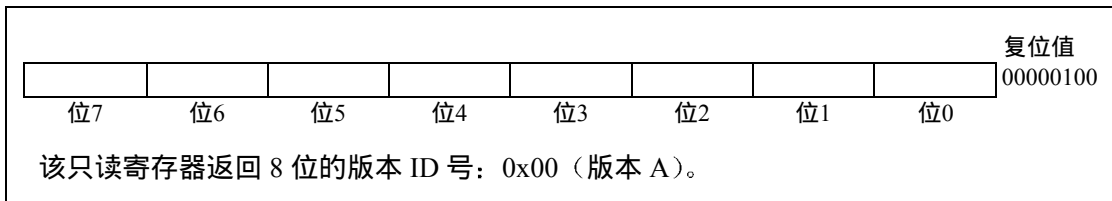


图 17.3 REVID: C2 版本 ID 寄存器



# C8051F300/1/2/3/4/5 混合信号 ISP FLASH 微控制器

**图 17.4 FPCTL: C2 FLASH 编程控制寄存器**

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	

位 7-0: FPCTL: FLASH 编程控制寄存器。  
 该寄存器用于允许通过 C2 接口对 FLASH 编程。为了允许 C2 FLASH 编程，必须按顺序写代码：0x02、0x01。一旦 C2 FLASH 编程被允许，必须进行一次复位才能使系统恢复正常工作。

**图 17.5 FPDAT: C2 FLASH 编程数据寄存器**

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	

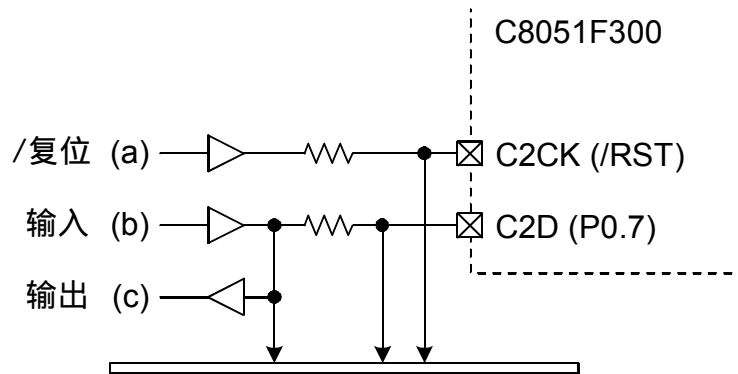
位 7-0: FPDAT: FLASH 编程数据寄存器。  
 该寄存器用于在 C2 FLASH 访问期间传递 FLASH 编程命令、地址和数据。下面列出了有效的编程命令。

代码	命令
0x06	读 FLASH 块
0x07	写 FLASH 块
0x08	擦除 FLASH 页
0x03	擦除器件

## 17.2 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 P0.7）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 17.6 所示。

图 17.6 典型 C2 引脚共享电路



在图 17.6 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。