C8051F000/1/2/5/6/7
C8051F010/1/2/5/6/7
混合信号 ISP FLASH 微控制器

数 据 手 册

潘 琢 金 译

Rev 1.4 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

混合信号 32KB ISP FLASH 微控制器

模拟外设

- **SAR ADC**
  - 12 位（C8051F000/1/2，C8051F005/6/7）
  - 10 位（C8051F010/1/2，C8051F015/6/7）
  - ±1LSB INL；无失码
  - 可编程转换速率，最大 100ksps
  - 可多达 8 个外部输入；可编程为单端输入或差分输入
  - 可编程放大器增益：16、8、4、2、1、0.5
  - 数据相关窗口中断发生器
  - 内置温度传感器（±3°C）
- **两个 12 位 DAC**
- **两个模拟比较器**
  - 16 个可编程回差电压值
  - 可用于产生中断或复位
- **电压基准**
  - 2.4V；15ppm/°C
  - 可输出到外部引脚
- **精确的 VDD 监视器和欠压检测器**

片内 JTAG 调试和边界扫描

- 片内调试电路提供全速、非侵入式的在系统调试（不需仿真器）
- 支持断点、单步、观察点、堆栈监视器
- 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
- 完全符合 IEEE1149.1 边界扫描标准
- 廉价的开发套件

高速 8051 微控制器内核

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

存储器

- 256 字节内部数据 RAM（F000/01/02/10/11/12）
- 2304 字节内部数据 RAM（F005/06/07/15/16/17）
- 32K 字节闪速存储器；可以在系统编程，扇区大小为 512 字节

数字外设

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

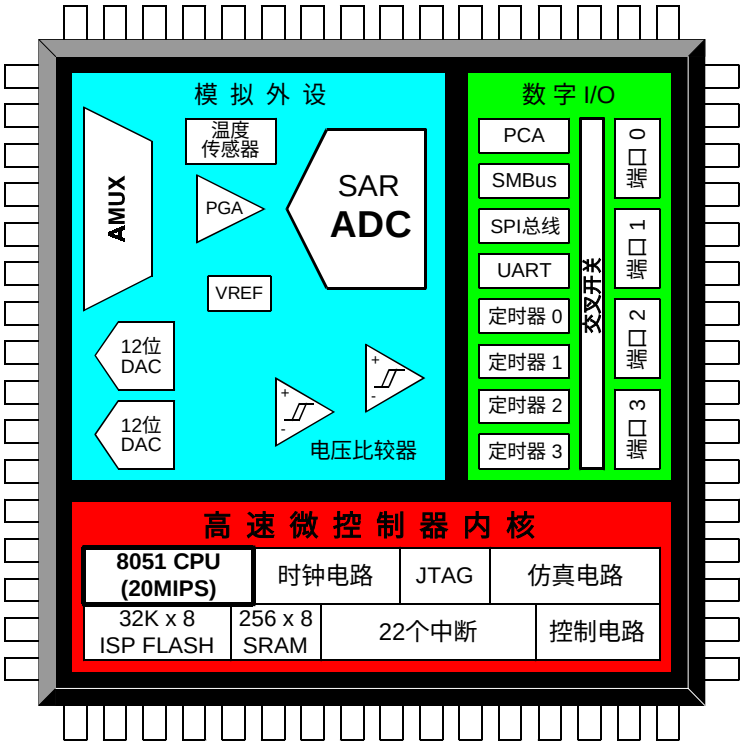
时钟源

- 内部可编程振荡器：2-16MHz
- 外部振荡器：晶体、RC、C、或外部时钟
- 可在运行中切换时钟源；适于节电模式时使用

供电电压……2.7V - 3.6V

- 典型工作电流：10mA @ 20MHz
- 多种节电休眠和停机方式

64 脚 TQFP、48 脚 TQFP、32 脚 LQFP 封装
温度范围：-40°C - +85°C



1 . 系统概述 6

1.1 CIP-51™ CPU 10

1.1.1 与 8051 完全兼容 10

1.1.2 速度提高 10

1.1.3 增加的功能 11

1.2 片内存储器 12

1.3 JTAG 调试和边界扫描 13

1.4 可编程数字 I/O 和交叉开关 14

1.5 可编程计数器阵列 15

1.6 串行端口 15

1.7 模数转换器 16

1.8 比较器和 DAC 17

2. 极限参数* 18

3. 总体直流电气特性 18

4. 引脚和封装定义 19

5. ADC（12 位，只适用于 C8051F000/1/2/5/6/7） 27

5.1 模拟多路开关和 PGA 27

5.2 ADC 的工作方式 28

5.3 ADC 可编程窗口检测器 35

6. ADC（10 位，只适用于 C8051F010/1/2/5/6/7） 39

6.1 模拟多路开关和 PGA 39

6.2 ADC 的工作方式 40

6.3 ADC 可编程窗口检测器 47

7. 12 位电压输出 DAC 52

8. 比较器 56

9. 电压基准 61

10. CIP-51 CPU 63

10.1 指令集 64

10.1.1 指令和 CPU 时序 64

10.1.2 MOVX 指令和程序存储器 64

10.2 存储器组织 68

10.2.1 程序存储器 68

10.2.2 数据存储器.....68
10.2.3 通用寄存器.....68
10.2.4 位寻址空间.....68
10.2.5 堆栈.....69
10.3 特殊功能寄存器.....70
10.3.1 寄存器说明.....74
10.4 中断系统.....77
10.4.1 MCU 中断源和中断向量.....77
10.4.2 外部中断.....77
10.4.3 中断优先级.....78
10.4.4 中断响应时间.....78
10.4.5 中断寄存器说明.....79
10.5 电源管理方式.....85
10.5.1 等待方式.....85
10.5.1 停机方式.....85

11. FLASH 存储器.....87

11.1 FLASH 存储器编程.....87
11.2 非易失性数据存储.....88
11.3 安全选项.....88

12．外部 RAM（C8051F005/06/07/15/16/17）.....92

13. 复位源.....93

13.1 上电复位.....94
13.2 软件强制复位.....94
13.3 掉电复位.....94
13.4 外部复位.....95
13.5 时钟丢失检测器复位.....95
13.6 比较器 0 复位.....95
13.7 外部 CNVSTR 引脚复位.....95
13.8 看门狗定时器复位.....95
13.8.1 看门狗的使用.....96

14. 振荡器.....100

14.1 外部晶体举例.....103
14.2 外部 RC 举例.....103
14.3 外部电容举例.....103

15. 端口输入/输出.....104

15.1 优先权交叉开关译码器.....104
15.2 端口 I/O 初始化.....104

15.3 通用端口 I/O 110
15.4 配置无对应引脚的端口 110

16. SMBUS 114

16.1 支持文档 115
16.2 工作过程 116
16.2.1 主发送器方式 116
16.2.2 主接收器方式 116
16.2.3 从发送器方式 117
16.2.4 从接收器方式 117
16.3 总线仲裁 117
16.4 时钟低电平扩展 117
16.5 超时 117
16.5.1 SCL 低电平超时 117
16.5.2 SCL 高电平（SMBus 空闲）超时 117
16.6 SMBUS 特殊功能寄存器 118
16.6.1 控制寄存器 118
16.6.2 时钟频率寄存器 120
16.6.3 数据寄存器 121
16.6.4 地址寄存器 121
16.6.5 状态寄存器 122

17. 串行外设接口总线 124

17.1 信号说明 125
17.1.1 主输出、从输入 125
17.1.2 主输入、从输出 125
17.1.3 串行时钟 125
17.1.4 从选择 125
17.2 操作 126
17.3 串行时钟时序 127
17.4 SPI 特殊功能寄存器 128

18. UART 131

18.1 UART 工作方式 132
18.1.1 方式 0：同步方式 132
18.1.2 方式 1：8 位 UART，可变波特率 133
18.1.3 方式 2：9 位 UART，固定波特率 134
18.1.4 方式 3：9 位 UART，可变波特率 135
18.2 多机通信 135

19. 定时器 138

19.1 定时器 0 和定时器 1 138

19.1.1 方式 0：13 位计数器/定时器.....138
19.1.2 方式 1：16 位计数器/定时器.....139
19.1.3 方式 2：8 位自动重装载的计数器/定时器.....140
19.1.4 方式 3：两个 8 位计数器/定时器（只限于定时器 0）.....141
19.2 定时器 2 146
19.2.1 方式 0：带捕捉的 16 位计数器/定时器.....147
19.2.2 方式 1：自动重装载的 16 位计数器/定时器.....148
19.2.3 方式 2：波特率发生器.....149
19.3 定时器 3 152

20. 可编程计数器阵列 154

20.1 捕捉/比较模块 155
20.1.1 边沿触发的捕捉方式.....156
20.1.2 软件定时器（比较）方式.....157
20.1.3 高速输出方式.....158
20.1.4 脉宽调制器方式.....159
20.2 PCA 计数器定时器.....160
20.3 PCA 寄存器说明.....161

21. JTAG (IEEE 1149.1) 165

21.1 边界扫描 166
21.1.1 EXTEST 指令.....167
21.1.2 SAMPLE 指令.....167
21.1.3 BYPASS 指令 167
21.1.4 IDCODE 指令 167
21.2 FLASH 编程命令.....168
21.3 调试支持 171

1．系统概述

C8051F000 系列器件是完全集成的混合信号系统级 MCU 芯片，有一个真正的 12 位多通道 ADC（F000/01/02/05/06/07）或一个 10 位的多通道 ADC（F010/11/12/15/16/17）。参见表 1.1 的产品选择指南可快速查看每个 MCU 的特性。每种器件都有一个可编程增益放大器、两个 12 位 DAC、两个电压比较器（F002/07/12/17 例外，只有一个）、一个电压基准、一个具有 32K 字节 FLASH 存储器并与 8051 兼容的微控制器内核。还有硬件实现的（不是在用户软件中用位操作模拟）I2C/SMBus、UART、SPI 串行接口及一个具有 5 个捕捉/比较模块的可编程计数器/定时器阵列（PCA）。还有 4 个通用的 16 位定时器和 4 字节宽的通用数字 I/O 端口。C8051F000/01/02/10/11/12 有 256 字节的 RAM，执行速度可达 20MIPS，而 C8051F005/06/07/15/16/17 有 2304 字节的 RAM，执行速度可达 25MIPS。

这些具有片内 VDD 监视器、WDT、和时钟振荡器的 MCU 是真正能独立工作的片上系统。每个 MCU 都能有效地管理模拟和数字外设。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。每个 MCU 都可以关闭任何一个或全部外设以节省功耗。

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

每个 MCU 都可在工业温度范围（-45℃到+85℃）内用 2.7V-3.6V 的电压工作。端口 I/O、/RST 和 JTAG 引脚都容许 5V 的输入信号电压。C8051F000/05/10/15 为 64 脚 TQFP 封装（见图 1.1 的框图）。C8051F001/06/11/16 为 48 脚 TQFP 封装（见图 1.2 的框图）。C8051F002/07/12/17 为 32 脚 LQFP 封装（见图 1.3 的框图）。

表 1.1 产品选择指南

	MIPS(峰值)	FLASH 存储器	RAM	SMBus/12C	SPI	UART	定时器(16 位)	可编程计数器阵列	数字端口 I/O	ADC 分辨率(位)	ADC 最大速度(ksps)	ADC 输入	电压基准	温度传感器	DAC 分辨率	DAC 输出	电压比较器	封装
C8051F000	20	32k	256	√	√	√	4	√	32	12	100	8	√	√	12	2	2	64TQFP
C8051F001	20	32k	256	√	√	√	4	√	16	12	100	8	√	√	12	2	2	48TQFP
C8051F002	20	32k	256	√	√	√	4	√	8	12	100	4	√	√	12	2	1	32LQFP
C8051F005	25	32k	2304	√	√	√	4	√	32	12	100	8	√	√	12	2	2	64TQFP
C8051F006	25	32k	2304	√	√	√	4	√	16	12	100	8	√	√	12	2	2	48TQFP
C8051F007	25	32k	2304	√	√	√	4	√	8	12	100	4	√	√	12	2	1	32LQFP
C8051F010	20	32k	256	√	√	√	4	√	32	10	100	8	√	√	12	2	2	64TQFP
C8051F011	20	32k	256	√	√	√	4	√	16	10	100	8	√	√	12	2	2	48TQFP
C8051F012	20	32k	256	√	√	√	4	√	8	10	100	4	√	√	12	2	1	32LQFP
C8051F015	25	32k	2304	√	√	√	4	√	32	10	100	8	√	√	12	2	2	64TQFP
C8051F016	25	32k	2304	√	√	√	4	√	16	10	100	8	√	√	12	2	2	48TQFP
C8051F017	25	32k	2304	√	√	√	4	√	8	10	100	4	√	√	12	2	1	32LQFP

图 1.1 C8051F000/05/10/15 原理框图

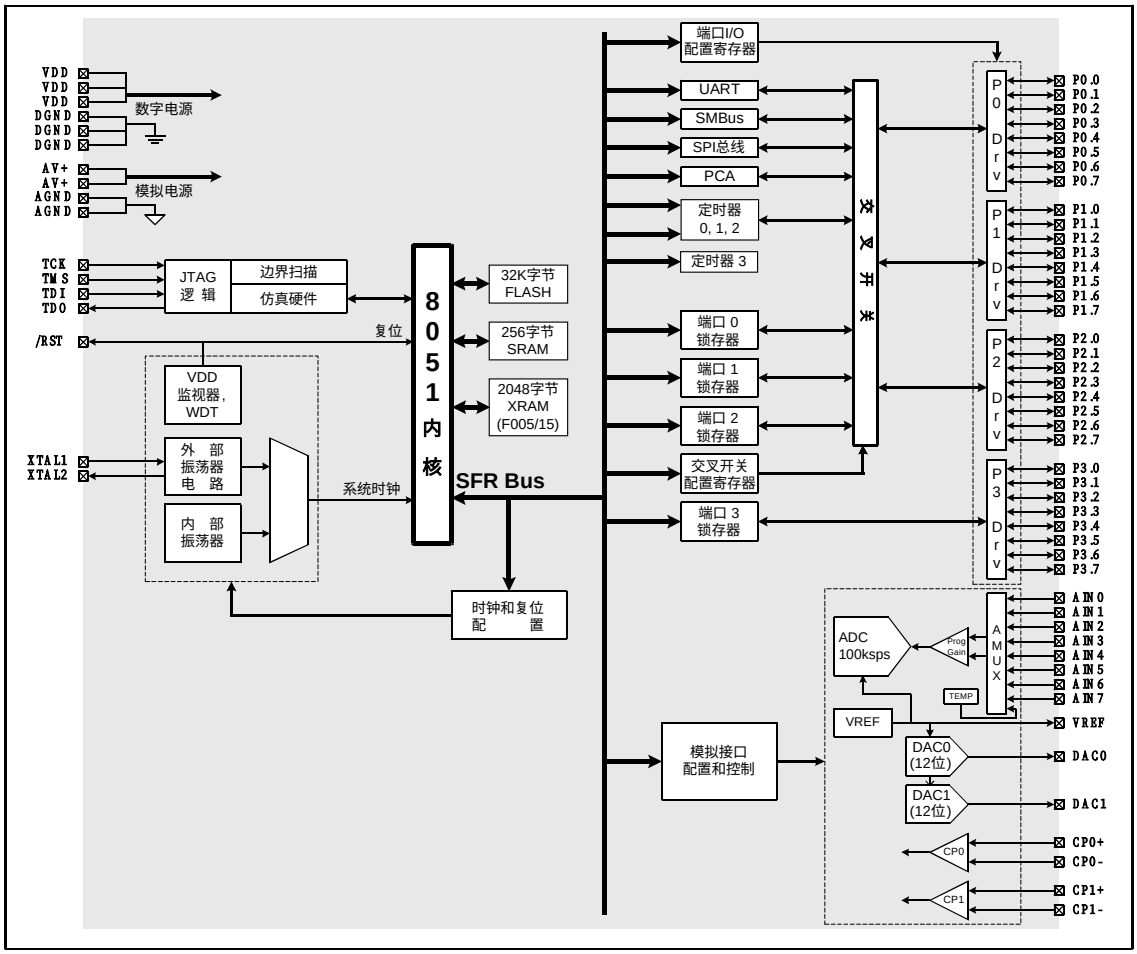


图 1.2 C8051F001/06/11/16 原理框图

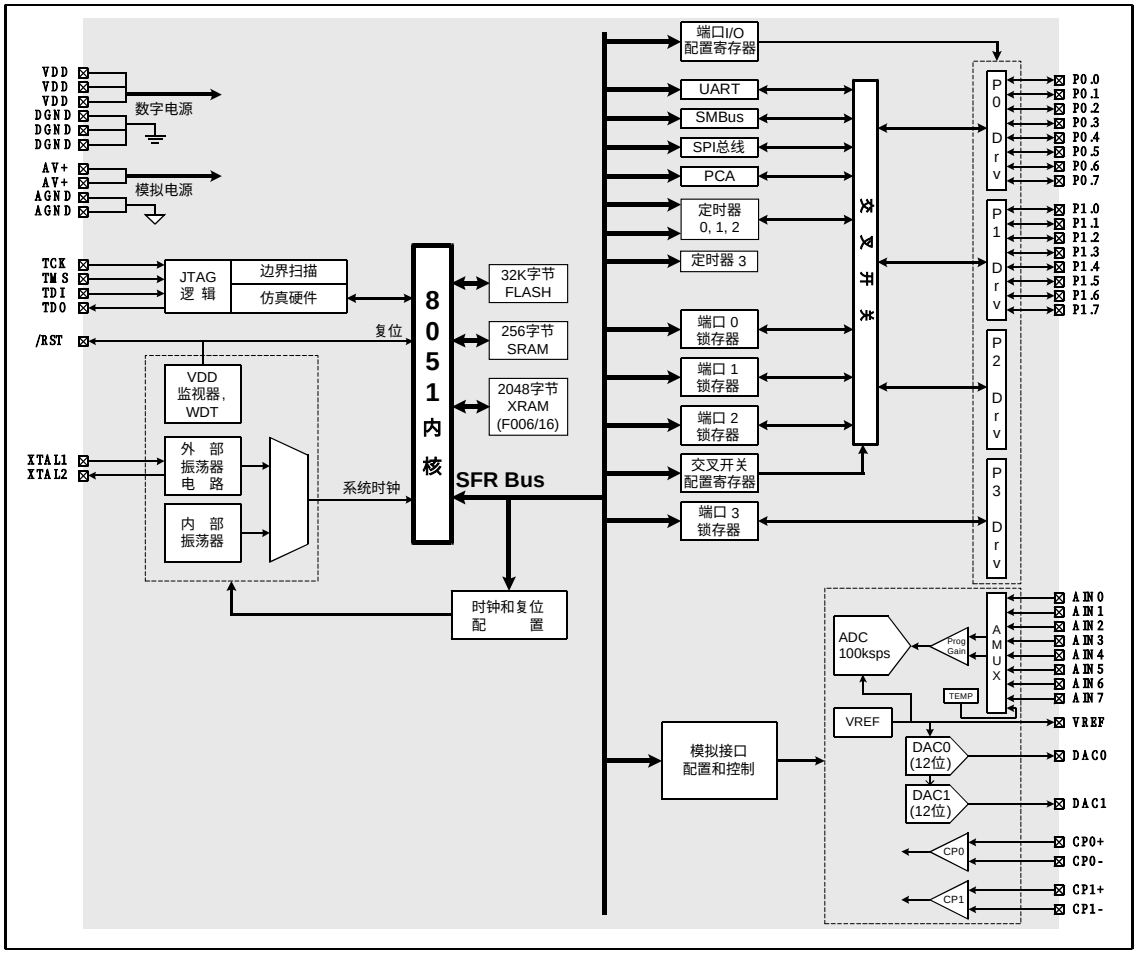
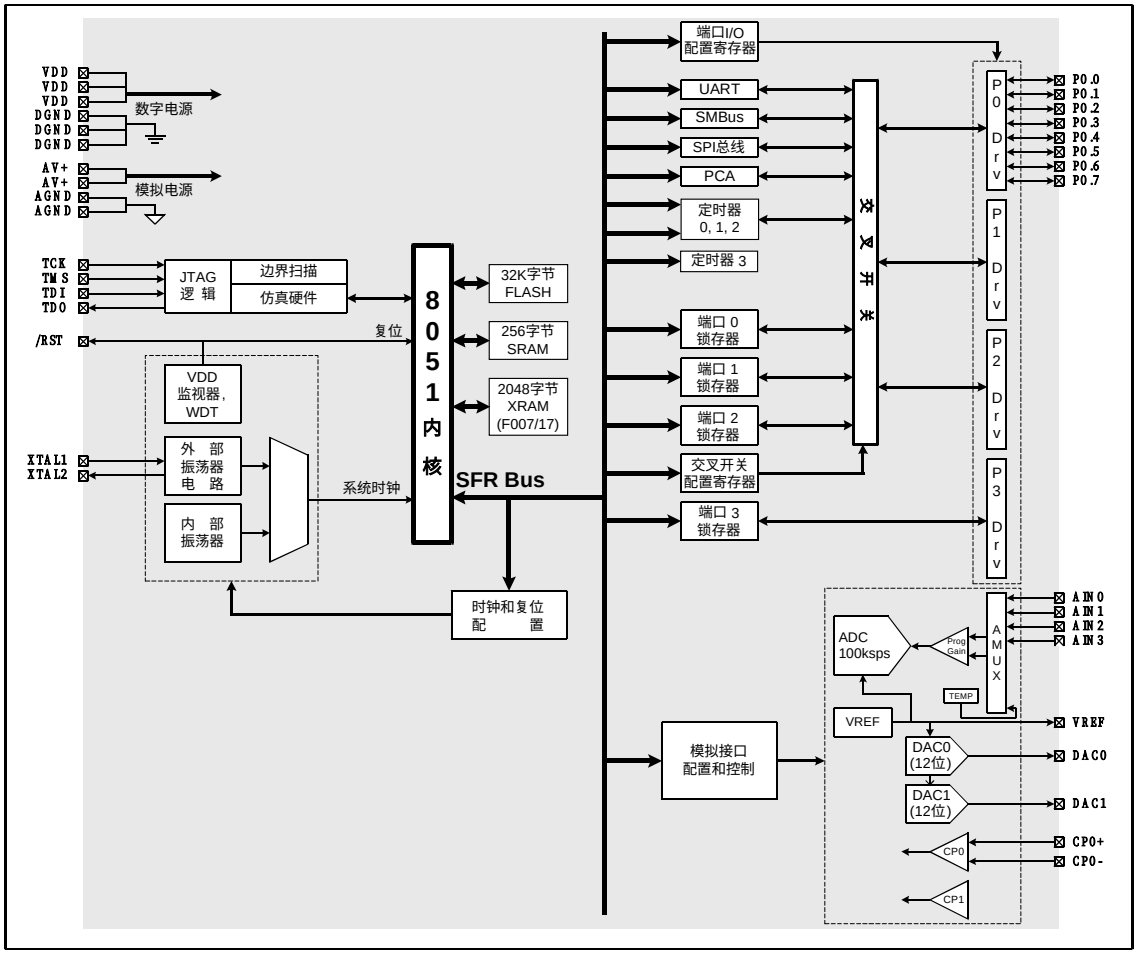


图 1.3 C8051F001/06/11/16 原理框图



1.1 CIP-51™ CPU

1.1.1 与 8051 完全兼容

C8051F000 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 4 个 16 位的计数器/定时器、一个全双工 UART、256 字节内部 RAM 空间、128 字节特殊功能寄存器（SFR）地址空间及 4 字节宽的 I/O 端口。

1.1.2 速度提高

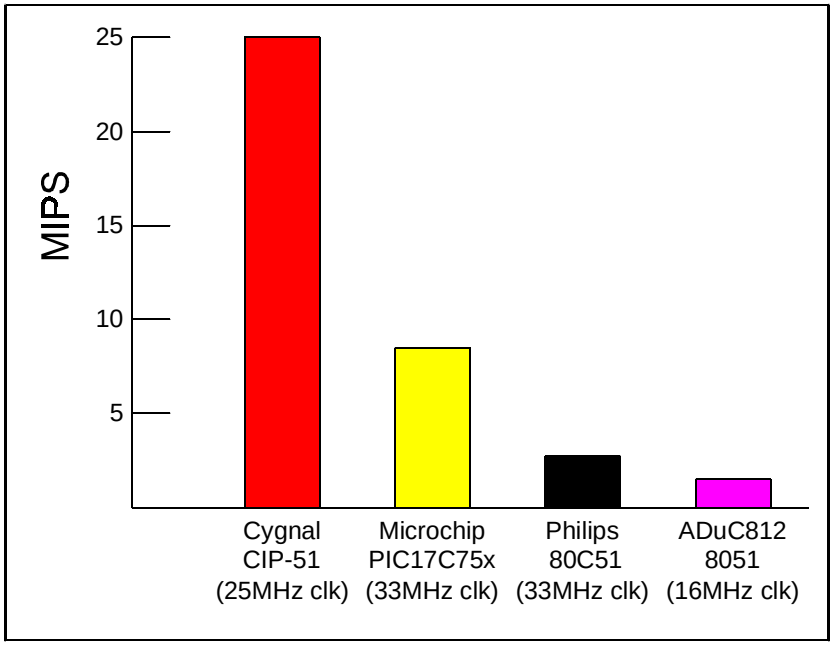
CIP-51 采用流水线结构，与标准的 8051 结构相比指令执行速度有很大的提高。在一个标准的 8051 中，除 MUL 和 DIV 以外所有指令都需要 12 或 24 个系统时钟周期，最大系统时钟频率为 12-24MHz。而对于 CIP-51 内核，70%的指令的执行时间为 1 或 2 个系统时钟周期，只有 4 条指令的执行时间大于 4 个系统时钟周期。

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

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

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 1.4 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 1.4 MCU 峰值执行速度比较



1.1.3 增加的功能

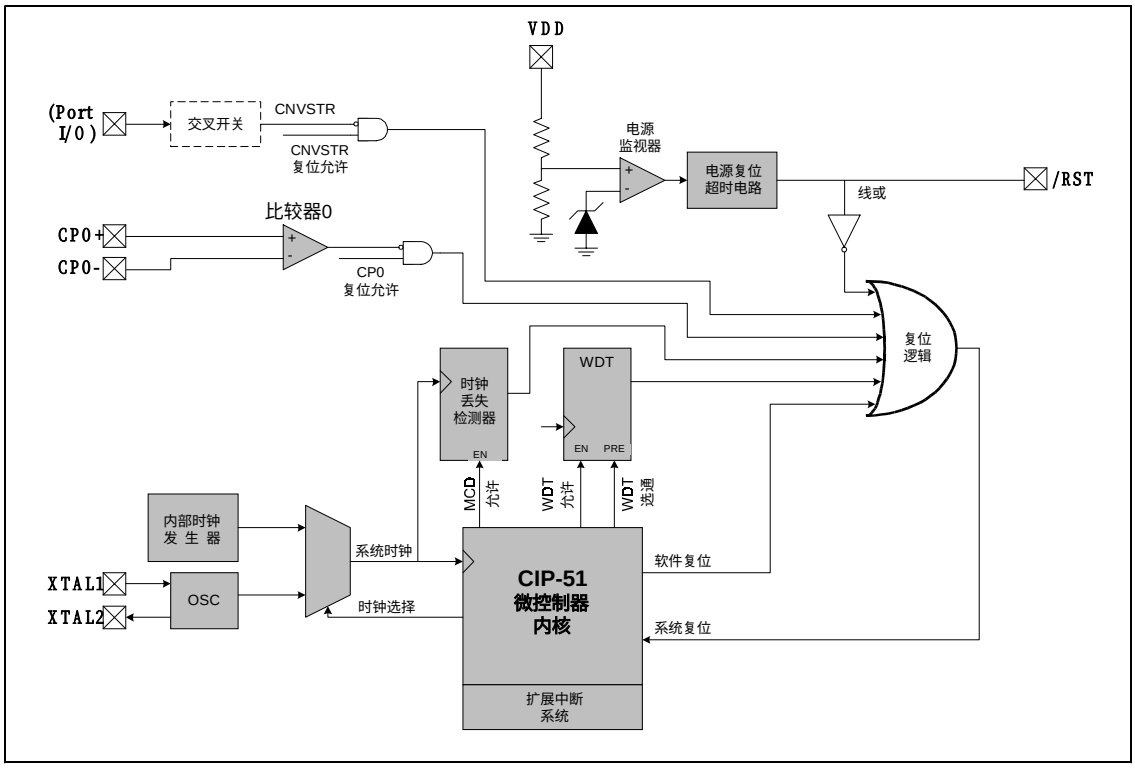
C8051F000 系列 MCU 在 CIP-51 内核的内部和外部有几项关键性的改进，提高了整体性能，更易于在最终应用中使用。

扩展的中断系统向 CIP-51 提供 21 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，却有更高的执行效率。在设计一个多任务实时系统时，这些增加的中断源是非常有用的。

MCU 可有多达 7 个复位源：一个片内 VDD 监视器、一个看门狗定时器、一个时钟丢失检测器、一个由比较器 0 提供的电压检测器、一个软件强制复位、CNVSTR 引脚及/RST 引脚。/RST 引脚是双向的，可接受外部复位或将内部产生的上电复位信号输出到/RST 引脚。除了 VDD 监视器和复位输入引脚以外，每个复位源都可以由用户用软件禁止。在一次上电复位之后的 MCU 初始化期间，WDT 可以一直被允许。

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

图 1.5 片内时钟和复位



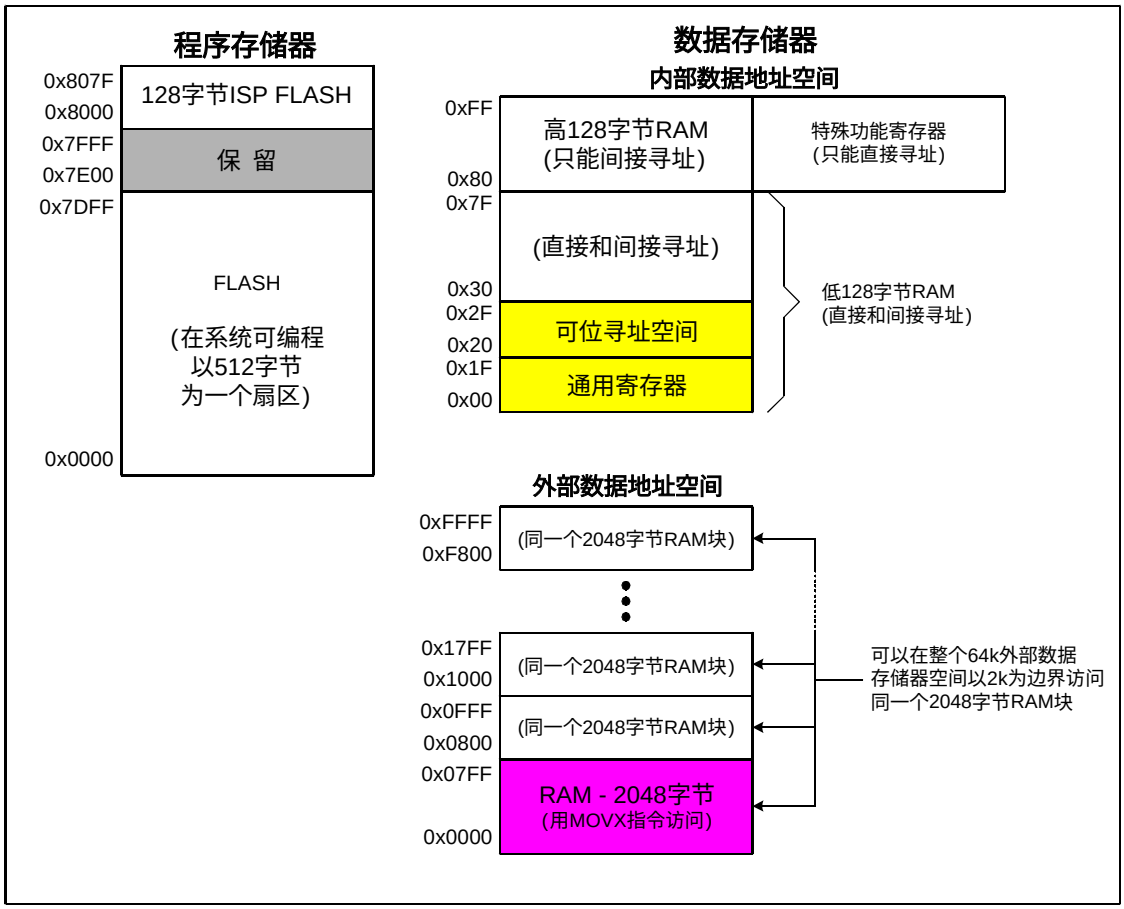
1.2 片内存储器

CIP-51 有标准的 8051 程序和数据地址配置。它包括 256 字节的数据 RAM，其中高 128 字节为两个地址空间。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 个通用工作寄存器区，接下来的 16 字节既可以按字节寻址又可以按位寻址。

C8051F005/06/07/15/16/17 中的 CIP-51 还另有位于外部数据存储器地址空间的 2048 字节的 RAM 块。这个 2048 字节的 RAM 块可以在整个 64k 外部数据存储器地址空间中被寻址（见图 1.6）。

MCU 的程序存储器包含 32k + 128 字节的 FLASH。该存储器以 512 字节为一个扇区，可以在系统编程，且不需在片外提供编程电压。从 0x7E00 到 0x7FFF 的 512 字节被保留，由工厂使用。还有一个位于地址 0x8000 - 0x807F 的 128 字节的扇区，该扇区可作为一个小的软件常数表或额外的程序空间。图 1.6 给出了 MCU 系统的存储器结构。

图 1.6 片内存储器结构



1.3 JTAG 调试和边界扫描

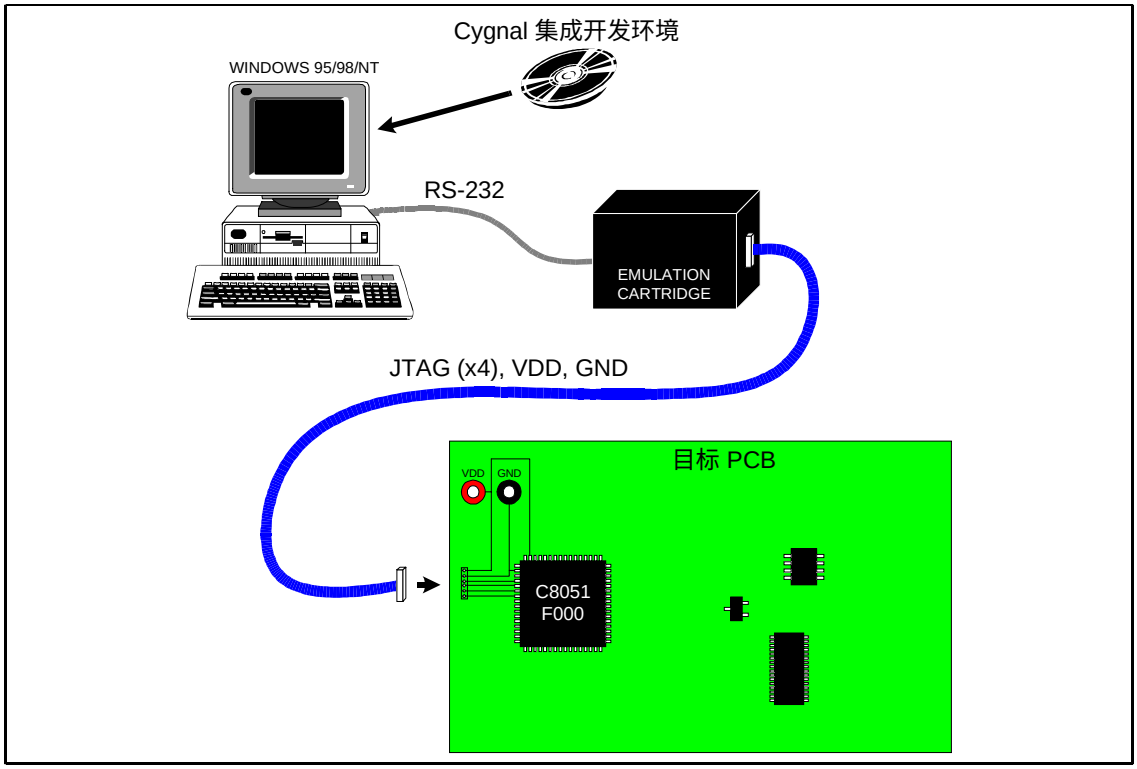
C8051F000系列具有片内JTAG和调试电路，通过4脚JTAG接口并使用安装在最终应用系统中的产品器件就可以进行非侵入式、全速的在系统调试。该JTAG接口完全符合IEEE 1149.1标准，为生产和测试提供完全的边界扫描功能。

Cygnal的调试系统支持观察和修改存储器和寄存器、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除外）都停止运行，以便保持同步。

开发套件C8051F000DK、C8051F005DK、C8051F010DK和C8051F015DK具有开发应用代码所需要的全部硬件和软件，并可分别对C8051F000/1/2、F005/6/7、F010/1/2、F015/6/7 MCU进行在系统调试。开发套件中包括开发者工作室软件和调试器、一个集成的8051汇编器和一个被称为EC的RS-232至JTAG协议转换模块。套件中还有一个目标应用板，上面有对应的MCU和一大块样机区域。套件中还包括RS-232和JTAG电缆及一个墙装电源。开发套件需要一个运行Windows 95/98/Me/NT并有一个可用RS-232串口的计算机。如图1.7所示，PC机通过RS-232与EC连接。一条六英寸的扁平电缆将EC和用户的应用板连接起来，包括4个JTAG引脚和VDD及GND。EC从应用板获取电源，在2.7-3.6V时其供电电流大约为20 mA。对于不能从目标板上获取足够电流的应用，可以将套件中提供的电源直接连到EC上。

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

图1.7 调试环境示意图



1.4 可编程数字 I/O 和交叉开关

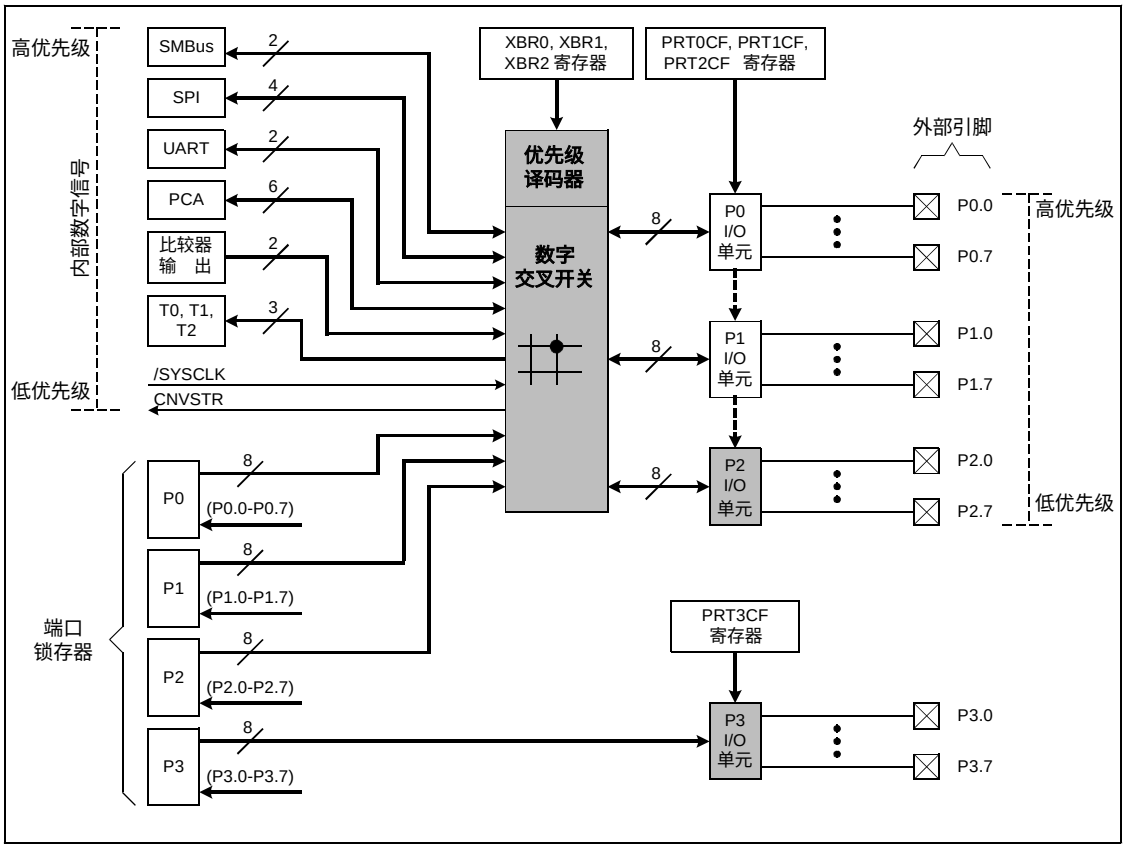
该系列MCU具有标准8051的端口（0、1、2和3）。在F000/05/10/15中这4个端口都有引出脚。在F001/06/10/16中端口0和1有引出脚，在F002/07/12/17中只有端口0有引出脚。没有引出脚的端口可用作通用寄存器。I/O端口的工作情况与标准8051相似，但有一些改进。

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

可能最突出的改进是引入了数字交叉开关。这是一个大的数字开关网络，允许将内部数字系统资源分配给P0、P1和P2的端口I/O引脚（见图1.8）。与具有标准复用数字I/O的微控制器不同，这种结构可支持所有的功能组合。

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

图1.8 数字交叉开关原理框图

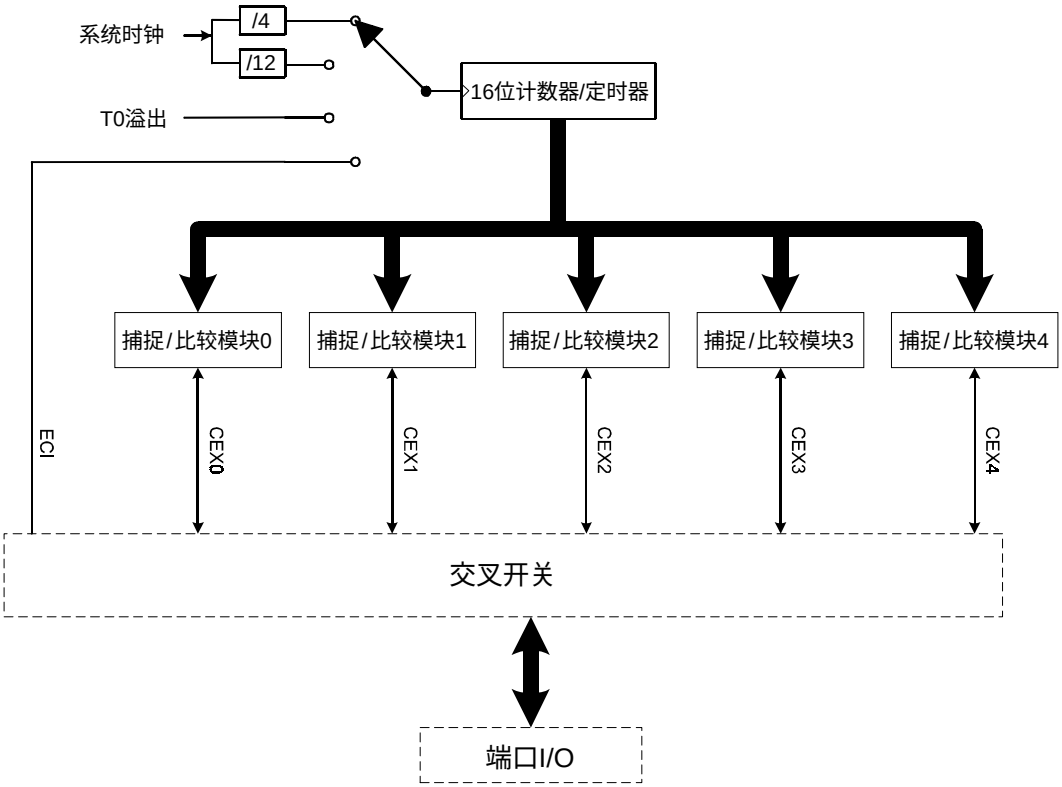


1.5 可编程计数器阵列

除了4个16位的通用计数器/定时器之外，C8051F000 MCU系列还有一个片内可编程计数器/定时器阵列（PCA）。PCA包括一个专用的16位计数器/定时器时间基准和5个可编程的捕捉/比较模块。时间基准的时钟可以是下面的4个时钟源之一：系统时钟/12、系统时钟/4、定时器0溢出或外部时钟输入（ECI）。

每个捕捉/比较模块都有4种工作方式：边沿触发捕捉、软件定时器、高速输出、脉冲宽度调制器。PCA捕捉/比较模块的I/O和外部时钟输入可以通过数字交叉开关连到MCU的端口I/O引脚。

图1.9 PCA原理框图



1.6 串行端口

C8051F000系列MCU内部有一个全双工UART、SPI总线和I2C/SMBus。每种串行总线都完全用硬件实现，都能向CIP-51产生中断，因此很少需要CPU的干预。这些串行总线不“共享”定时器、中断或端口I/O，所以可以使用任何一个或全部同时使用。

1.7 模数转换器

C8051F000/1/2/5/6/7有一个片内12位SAR ADC，一个9通道输入多路选择开关和可编程增益放大器。该ADC工作在100ksps的最大采样速率时可提供真正的12位精度，INL为±1LSB。C8051F010/1/2/5/6/7与C8051F000/1/2/5/6/7类似，但分辨率为10位。每个ADC的最大转换速率均为100ksps。C8051F00x可提供真正的12位精度，C8051F00x的ADC可提供真正的10位精度，每个ADC都具有±1LSB的INL。片内还有一个15ppm的电压基准，也可以通过VREF引脚使用外部基准。

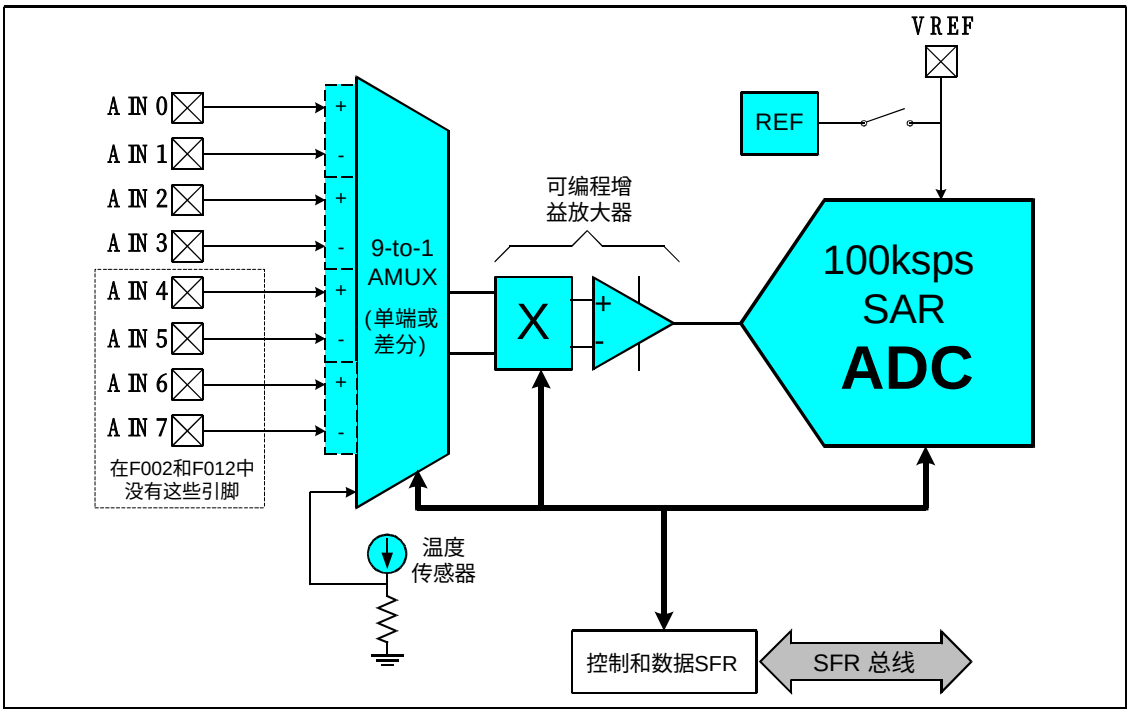
ADC完全由CIP-51通过特殊功能寄存器控制。有一个输入通道被连到内部温度传感器，其它8个通道接外部输入。8个外部输入通道的每一对都可被配置为两个单端输入或一个差分输入。系统控制器可以关断ADC以节省功耗。

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

A/D转换可以有4种启动方式：软件命令、定时器2溢出、定时器3溢出或外部信号输入。这种灵活性允许用软件事件、硬件信号触发转换或进行连续转换。一次转换完成可以产生一个中断，或者用软件查询一个状态位来判断转换结束。在转换完成时，10或12位转换结果数据字被锁存到两个特殊功能寄存器中。在软件控制下，这些数据字可以是左对齐或右对齐。

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

图1.10 ADC原理框图



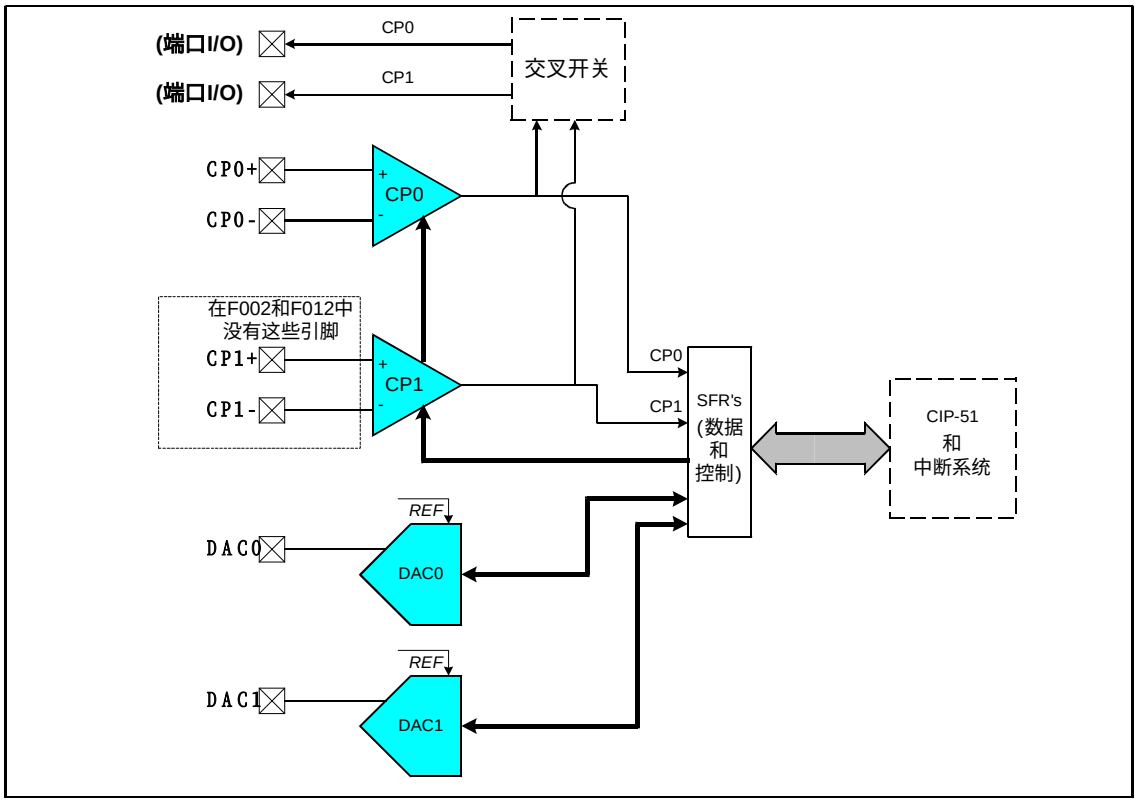
1.8 比较器和 DAC

C8051F000系列MCU内部有两个12位DAC和两个比较器（在F002、F007、F012、F017中没有第二个比较器CP1）。MCU与每个比较器和DAC之间的数据和控制接口通过特殊功能寄存器实现。MCU可以将任何一个DAC或比较器置于低功耗关断方式。

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

DAC为电压输出方式，与ADC使用同一个电压基准。DAC在作为比较器的参考电压或为ADC差分输入提供偏移电压时非常有用。

图1.11 比较器和DAC原理框图



2. 极限参数*

环境温度（通电情况下）..... -55到125℃
储存温度 ..... -65到150℃
任何引脚相对DGND的电压（VDD和端口I/O除外） ..... -0.3V到(VDD + 0.3V)
任何端口I/O引脚或/RST相对DGND的电压 ..... -0.3V到5.8V
VDD引脚相对DGND的电压 ..... -0.3V到4.2V
通过VDD、AV+、DGND和AGND的最大总电流..... 800mA
任何端口引脚的最大输出灌电流..... 100mA
任何其它I/O引脚的最大输出灌电流 25mA
任何端口引脚的最大输出拉电流..... 100mA
任何其它I/O引脚的最大输出拉电流 25mA

*注：超过这些列出的“极限参数”可能导致器件永久性损坏。

3. 总体直流电气特性

-40℃到+85℃（除非另有说明）。

参　数	条　件	最小值	典型值	最大值	单 位
模拟电源电压	(注1)	2.7	3.0	3.6	V
模拟电源电流	内部REF、ADC、 DAC、比较器都工作		1	2	mA
模拟电源电流 模拟子系统不工作	内部REF、ADC、 DAC、比较器都被禁止； 振荡器被禁止		5	20	μA
模拟与数字电源电压 之差（|VDD-AV+|）				0.5	V
数字电源电压		2.7	3.0	3.6	V
数字电源电流 CPU工作	VDD=2.7V, CLK=25MHz VDD=2.7V, CLK= 1MHz VDD=2.7V, CLK=32kHz		12.5 0.5 10		mA mA μA
数字电源电流 （停机方式）	振荡器不运行		5		μA
数字电源电压（RAM数 据保持电压）			1.5		V
额定工作温度范围		-40		+85	摄氏度

注 1: 模拟电源 AV+必须大于 1V 才能使 VDD 监视器工作。

4. 引脚和封装定义

表 4.1 引脚定义

引脚名称	引脚号			类 型	说 明
	F000 F010	F001 F011	F002 F012		
VDD	31，40，62	23，32	18，20		数字电源
DGND	30，41，61，	22，33，27，19	17，21		数字地
AV+	16，17	13，43	9，29		模拟电源
AGND	5，15	44，12	8，30		模拟地
TCK	22	18	14	数字输入	带内部上拉的 JTAG 测试时钟
TMS	21	17	13	数字输入	带内部上拉的 JTAG 测试模式选择
TDI	28	20	15	数字输入	带内部上拉的 JTAG 测试数据输入。TDI 在 TCK 上升沿被锁存
TDO	29	21	16	数字输出	带内部上拉的 JTAG 测试数据输出。数据在 TCK 的下降沿从 TDO 引脚输出。TDO 输出是一个三态驱动器。
XTAL1	18	14	10	模拟输入	晶体输入。该引脚为晶体或陶瓷谐振器的内部振荡器电路反馈输入。为了得到一个精确的内部时钟，可以在 XTAL1 和 XTAL2 之间接上一个晶体或陶瓷谐振器。如果被一个外部 CMOS 时钟驱动，则该引脚为系统时钟输入。
XTAL2	19	15	11	模拟输出	晶体输出。该引脚是晶体或陶瓷谐振器的激励驱动器。
/RST	20	16	12	数字 I/O	芯片复位。内部电压监视器的漏极开路输出，当 VDD＜2.7V 时为低电平。一个外部源可以通过将该引脚置为低电平使系统复位。
VREF	6	3	3	模拟 I/O	电压基准。当被配置为输入时，该引脚作为 MCU 的电压基准。否则，内部电压基准驱动该引脚。
CP0+	4	2	2	模拟输入	比较器 0 的同相输入端
CP0-	3	1	1	模拟输入	比较器 0 的反相输入端
CP1+	2	45		模拟输入	比较器 1 的同相输入端
CP1-	1	46		模拟输入	比较器 1 的反相输入端
DAC0	64	48	32	模拟输出	数模转换器输出口 0。DAC0 电压输出。（见第 7 节的 DAC 详细说明）
DAC1	63	47	31	模拟输出	数模转换器输出口 1。DAC1 端电压输出。（见第 7 节的 DAC 详细说明）
AIN0	7	4	4	模拟输入	模拟 MUX 通道输入 0（详见 ADC 指标说明）
AIN1	8	5	5	模拟输入	模拟 MUX 通道输入 1（详见 ADC 指标说明）
AIN2	9	6	6	模拟输入	模拟 MUX 通道输入 2（详见 ADC 指标说明）
AIN3	10	7	7	模拟输入	模拟 MUX 通道输入 3（详见 ADC 指标说明）
AIN4	11	8		模拟输入	模拟 MUX 通道输入 4（详见 ADC 指标说明）
AIN5	12	9		模拟输入	模拟 MUX 通道输入 5（详见 ADC 指标说明）
AIN6	13	10		模拟输入	模拟 MUX 通道输入 6（详见 ADC 指标说明）
AIN7	14	11		模拟输入	模拟 MUX 通道输入 7（详见 ADC 指标说明）

引脚名称	引脚号			类型	说 明
	F000 F010	F001 F011	F002 F012		
P0.0	39	31	19	数字 I/O	端口 0 位 0（详见端口 I/O 子系统部分）
P0.1	42	34	22	数字 I/O	端口 0 位 1（详见端口 I/O 子系统部分）
P0.2	47	35	23	数字 I/O	端口 0 位 2（详见端口 I/O 子系统部分）
P0.3	48	36	24	数字 I/O	端口 0 位 3（详见端口 I/O 子系统部分）
P0.4	49	37	25	数字 I/O	端口 0 位 4（详见端口 I/O 子系统部分）
P0.5	50	38	26	数字 I/O	端口 0 位 5（详见端口 I/O 子系统部分）
P0.6	55	39	27	数字 I/O	端口 0 位 6（详见端口 I/O 子系统部分）
P0.7	56	40	28	数字 I/O	端口 0 位 7（详见端口 I/O 子系统部分）
P1.0	38	30		数字 I/O	端口 1 位 0（详见端口 I/O 子系统部分）
P1.1	37	29		数字 I/O	端口 1 位 1（详见端口 I/O 子系统部分）
P1.2	36	28		数字 I/O	端口 1 位 2（详见端口 I/O 子系统部分）
P1.3	35	26		数字 I/O	端口 1 位 3（详见端口 I/O 子系统部分）
P1.4	34	25		数字 I/O	端口 1 位 4（详见端口 I/O 子系统部分）
P1.5	32	24		数字 I/O	端口 1 位 5（详见端口 I/O 子系统部分）
P1.6	60	42		数字 I/O	端口 1 位 6（详见端口 I/O 子系统部分）
P1.7	59	41		数字 I/O	端口 1 位 7（详见端口 I/O 子系统部分）
P2.0	33			数字 I/O	端口 2 位 0（详见端口 I/O 子系统部分）
P2.1	27			数字 I/O	端口 2 位 1（详见端口 I/O 子系统部分）
P2.2	54			数字 I/O	端口 2 位 2（详见端口 I/O 子系统部分）
P2.3	53			数字 I/O	端口 2 位 3（详见端口 I/O 子系统部分）
P2.4	52			数字 I/O	端口 2 位 4（详见端口 I/O 子系统部分）
P2.5	51			数字 I/O	端口 2 位 5（详见端口 I/O 子系统部分）
P2.6	44			数字 I/O	端口 2 位 6（详见端口 I/O 子系统部分）
P2.7	43			数字 I/O	端口 2 位 7（详见端口 I/O 子系统部分）
P3.0	26			数字 I/O	端口 3 位 0（详见端口 I/O 子系统部分）
P3.1	25			数字 I/O	端口 3 位 1（详见端口 I/O 子系统部分）
P3.2	24			数字 I/O	端口 3 位 2（详见端口 I/O 子系统部分）
P3.3	23			数字 I/O	端口 3 位 3（详见端口 I/O 子系统部分）
P3.4	58			数字 I/O	端口 3 位 4（详见端口 I/O 子系统部分）
P3.5	57			数字 I/O	端口 3 位 5（详见端口 I/O 子系统部分）
P3.6	46			数字 I/O	端口 3 位 6（详见端口 I/O 子系统部分）
P3.7	45			数字 I/O	端口 3 位 7（详见端口 I/O 子系统部分）

图 4.1 TQFP-64 引脚图

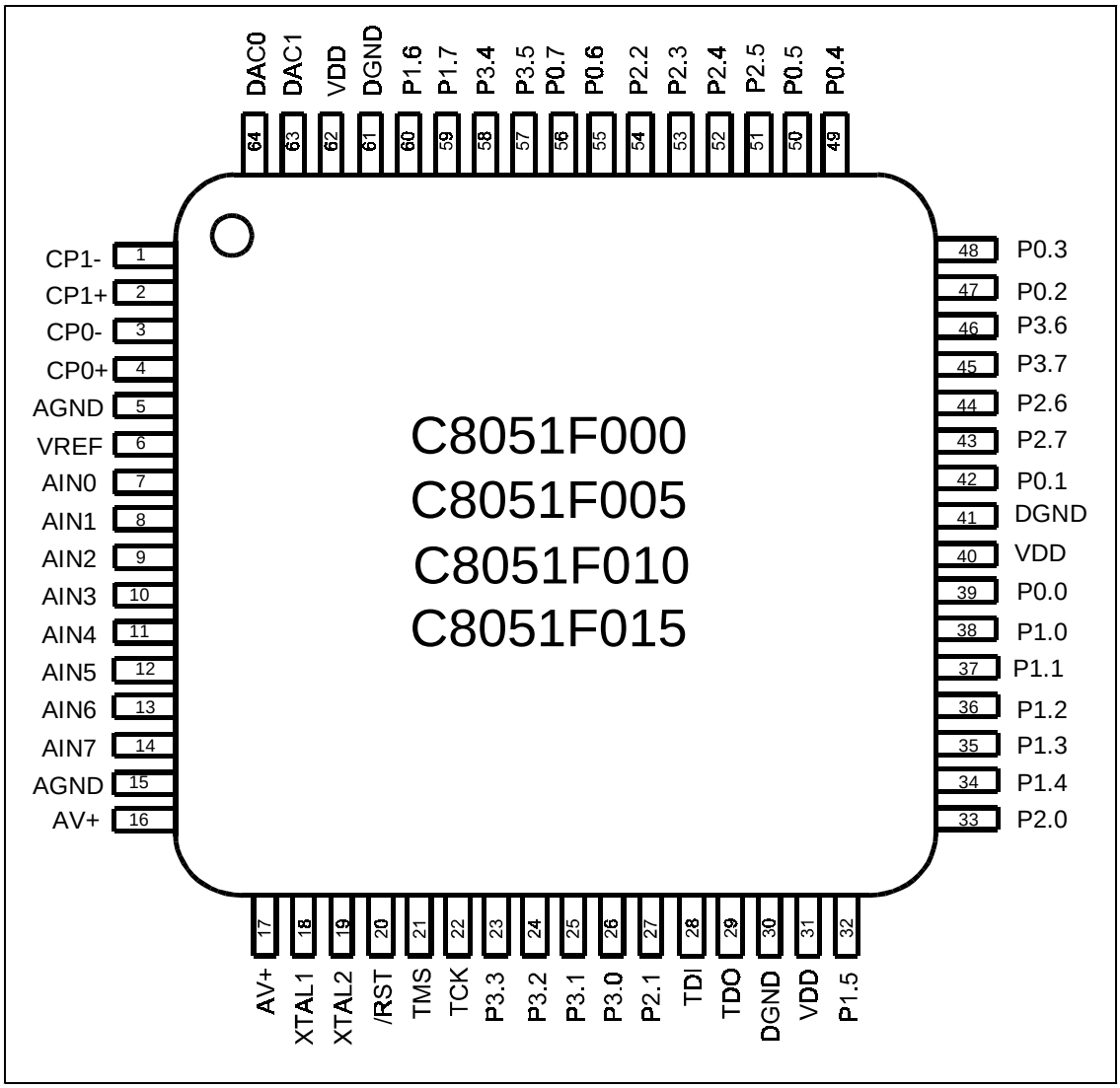
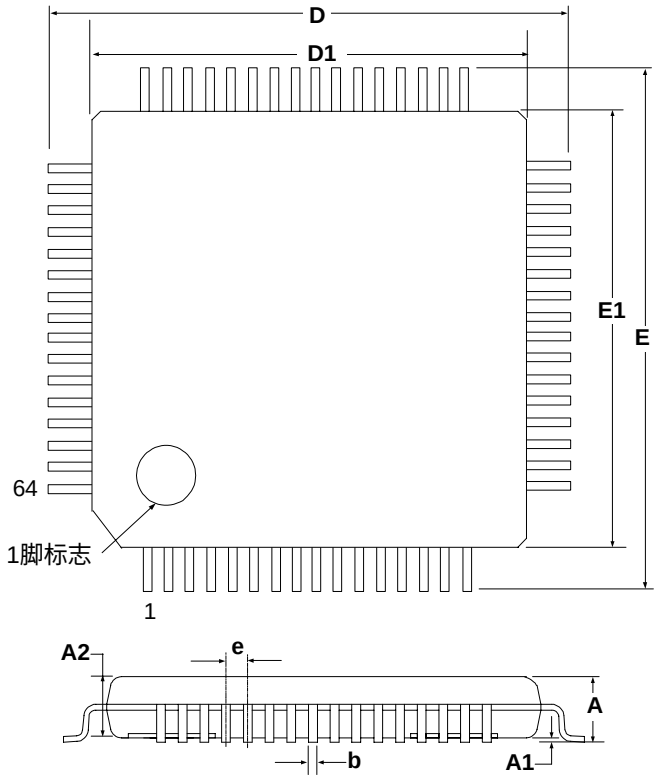


图 4.2 TQFP-64 封装图



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

图 4.3 TQFP-48 引脚图

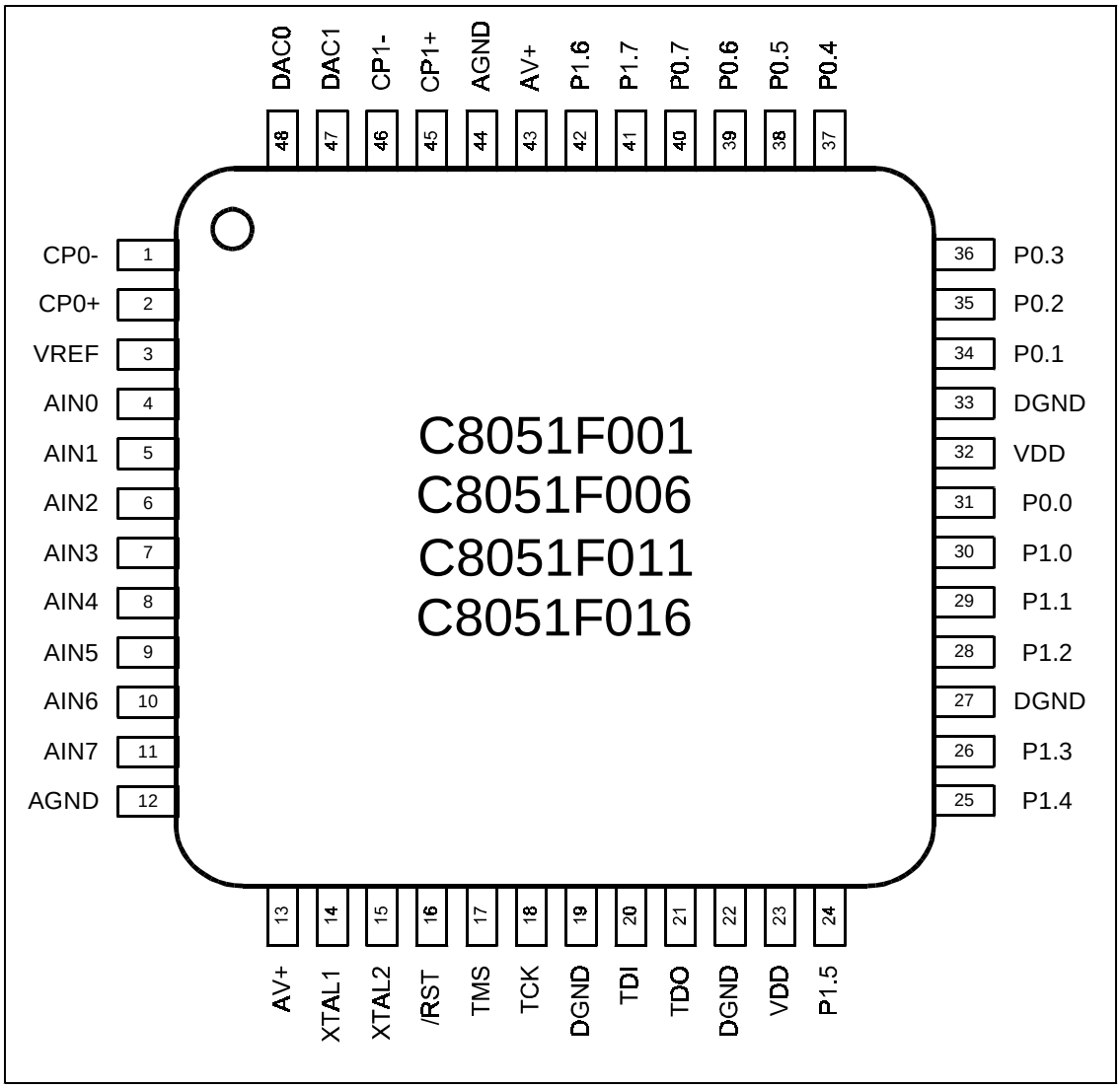
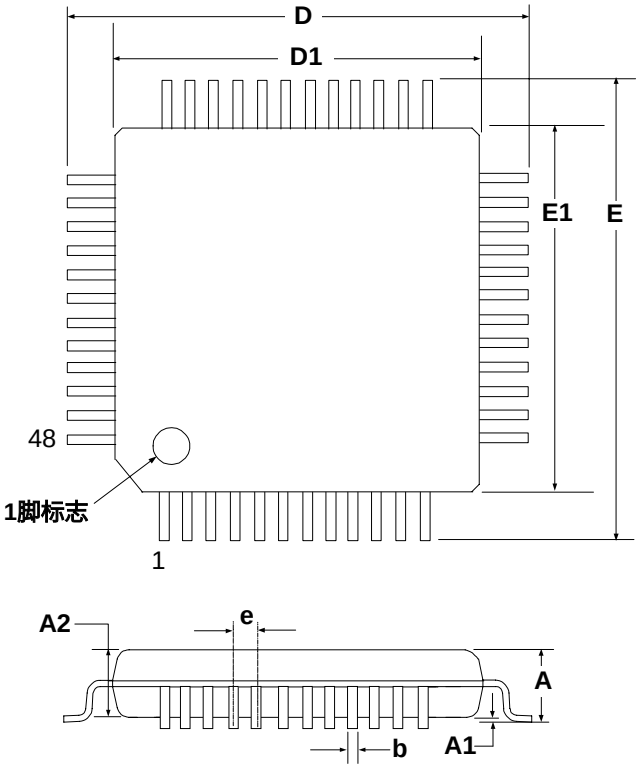


图 4.4 TQFP-48 封装图



	MIN (mm)	NOM (mm)	MAX (mm)
A	-	-	1.20
A1	0.05	-	0.15
A2	0.95	1.00	1.05
b	0.17	0.22	0.27
D	-	9.00	-
D1	-	7.00	-
e	-	0.50	-
E	-	9.00	-
E1	-	7.00	-

图 4.5 LQFP-32 引脚图

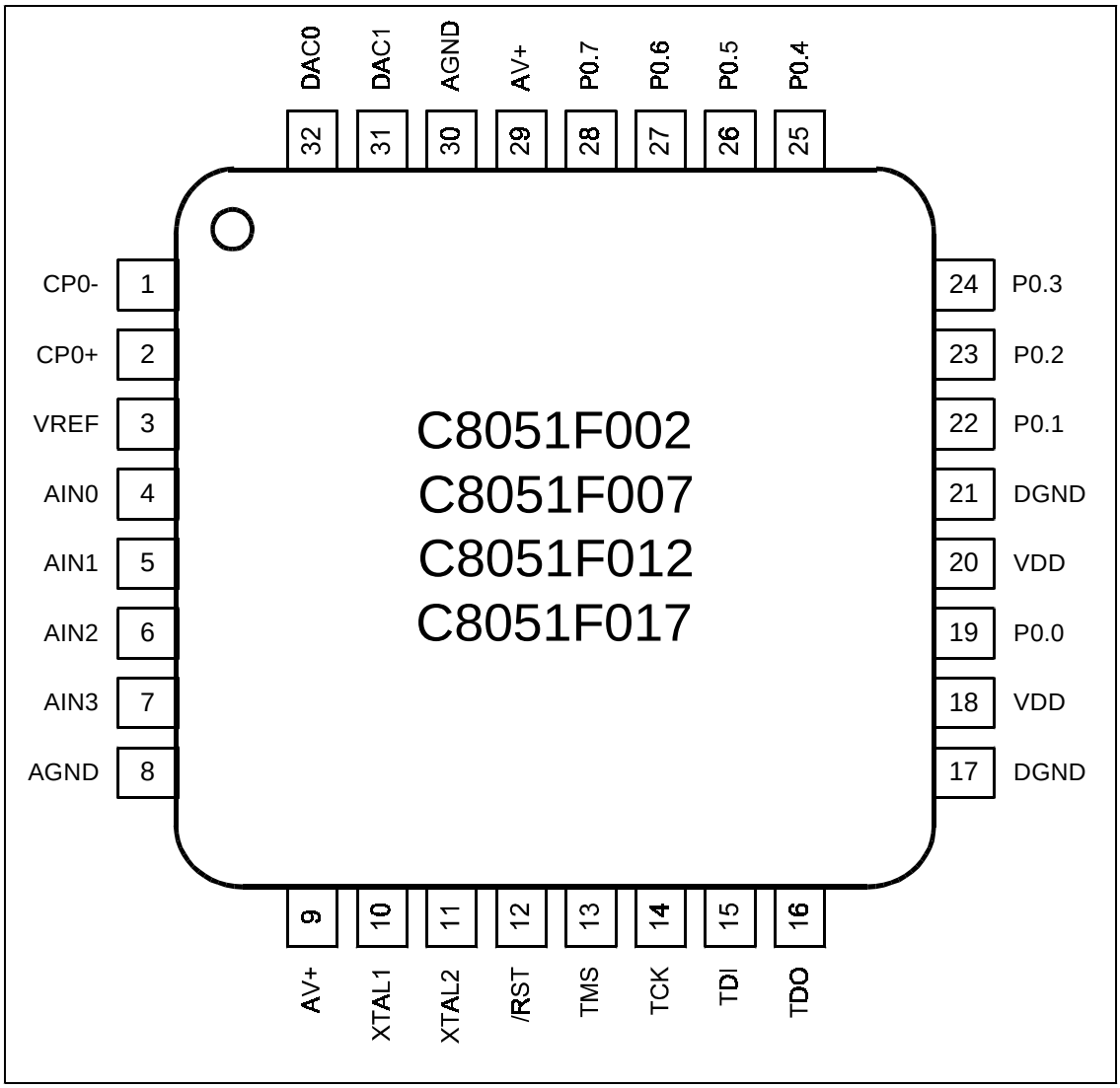
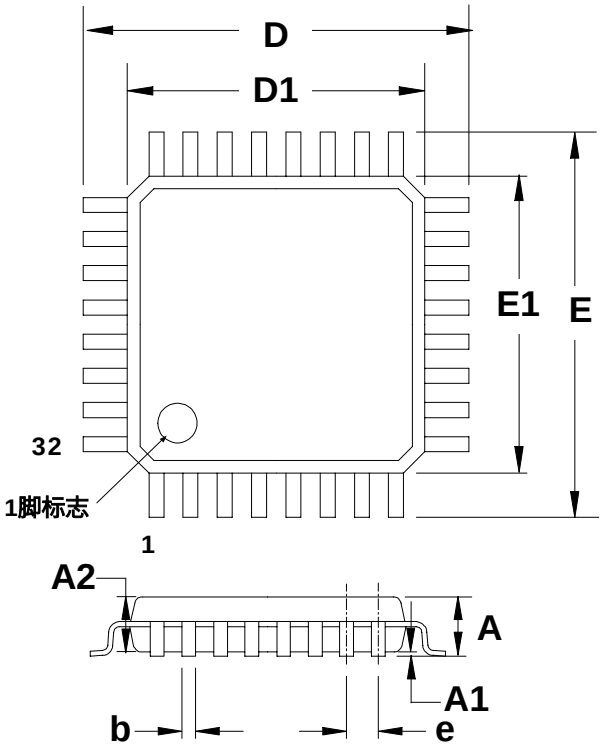


图 4.6 LQFP-32 封装图

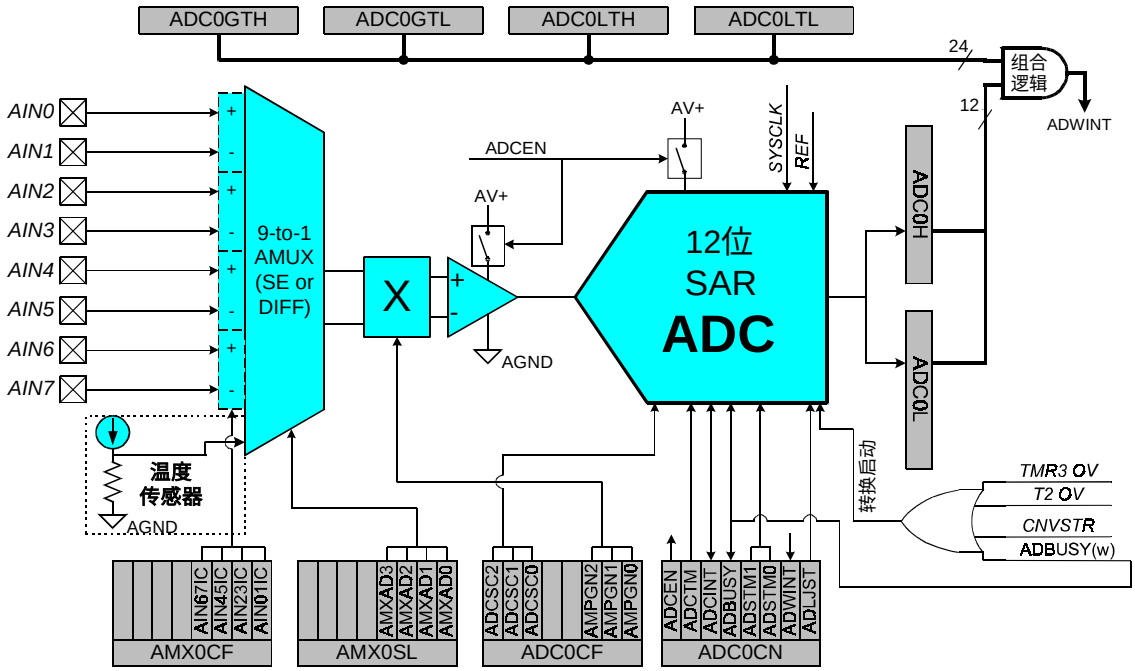


	MIN (mm)	NOM (mm)	MAX (mm)
A	-	-	1.60
A1	0.05	-	0.15
A2	1.35	1.40	1.45
b	0.30	0.37	0.45
D	-	9.00	-
D1	-	7.00	-
e	-	0.80	-
E	-	9.00	-
E1	-	7.00	-

5. ADC（12 位，只适用于 C8051F000/1/2/5/6/7）

C8051F000/1/2/5/6/7 的 ADC 子系统包括一个 9 通道的可配置模拟多路开关（AMUX），一个可编程增益放大器（PGA）和一个 100ksps、12 位分辨率的逐次逼近寄存器型 ADC，ADC 中集成了跟踪保持电路和可编程窗口检测器（见图 5.1 的原理框图）。AMUX、PGA、数据转换方式及窗口检测器都可用软件通过图 5.1 所示的特殊功能寄存器来配置。只有当 ADC 控制寄存器（ADC0CN，图 5.7）中的 ADCEN 位被置‘1’时 ADC 子系统（ADC、跟踪保持器和 PGA）才被允许工作。当 ADCEN 位为‘0’时，ADC 子系统处于低功耗关断方式。如果要给 ADC 提供偏置，必须将 REF0CN 寄存器（见图 9.2）中的偏置允许位（BIASE）置‘1’。

图 5.1 12 位 ADC 功能框图



5.1 模拟多路开关和 PGA

AMUX 中的 8 个通道用于外部测量，而第九通道在内部被接到片内温度传感器（温度传输函数示于图 5.3）。注意，PGA 的增益对温度传感器也起作用。可以将 AMUX 输入对编程为工作在差分或单端方式。这就允许用户对每个通道选择最佳的测量技术，甚至可以在测量过程中改变方式。在系统复位后 AMUX 的默认方式为单端输入。有两个与 AMUX 相关的寄存器：通道选择寄存器 AMX0SL（图 5.5）和配置寄存器 AMX0CF（图 5.4）。图 5.5 中的表给出了每种配置下各通道的功能。PGA 对 AMUX 输出信号的放大倍数由 ADC 配置寄存器 ADC0CF（图 5.6）中的 AMPGN2-0 确定。PGA 增益可以用软件编程为 0.5、1、2、4、8 或 16。复位时的默认增益为 1。

5.2 ADC 的工作方式

ADC 使用 VREF 来确定它的满度电压，因此在进行一次转换之前必须正确设置这个参考电压（见第九章）。ADC 的最高转换速度为 100ksps，转换时钟来源于系统时钟。可以通过设置 ADC0CF 寄存器的 ADCSC 位将转换时钟的速度降为系统时钟的 1/2、1/4、1/8 或 1/16。这一功能用于根据不同的系统时钟速度调整转换速度。

有 4 种 A/D 转换启动方式，由 ADC0CN 的 ADC 启动转换方式位（ADSTM1，ADSTM0）的状态决定。转换触发源有：

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

向 ADBUSY 写 1 方式提供了在需要时用软件控制 ADC 启动转换的能力。ADBUSY 位在转换期间被置‘1’，转换结束后复‘0’。ADBUSY 位的下降沿触发一个中断（当被允许时）并置‘1’ADC0CN 中的中断标志。转换结果保存在 ADC 数据字的 MSB 和 LSB 寄存器：ADC0H 和 ADC0L。转换数据在寄存器对 ADC0H:ADC0L 中的存储方式可以是左对齐或右对齐，由 ADC0CN 寄存器中 ADLJST 位的编程状态决定。

ADC0CN 中的 ADCTM 位控制 ADC 的跟踪保持方式。在缺省状态，ADC 输入被连续跟踪（转换期间除外）。将 ADCTM 位设置为‘1’即可采用下面的 4 种低功耗跟踪保持方式，由 ADSTM1-0 位（也在 ADC0CN 中）的状态决定采用哪一种方式。

1. 从向 ADBUSY 写‘1’开始跟踪，持续 3 个 SAR 时钟；
2. 从定时器 3 溢出开始跟踪，持续 3 个 SAR 时钟；
3. 只在 CNVSTR 输入为低电平时跟踪；
4. 从定时器 2 溢出开始跟踪，持续 3 个 SAR 时钟；

上述的方式 1、2 和 4 在使用软件命令触发转换或 ADC 连续工作时有用。方式 3 用于需要使用外部硬件触发转换的场合。在方式 3，当 CNVSTR 输入信号为高电平时跟踪保持电路处于低功耗状态。当整个芯片处于低功耗停机或休眠方式时，可以禁止跟踪（关断）。

图 5.2　12 位 ADC 跟踪和转换时序举例

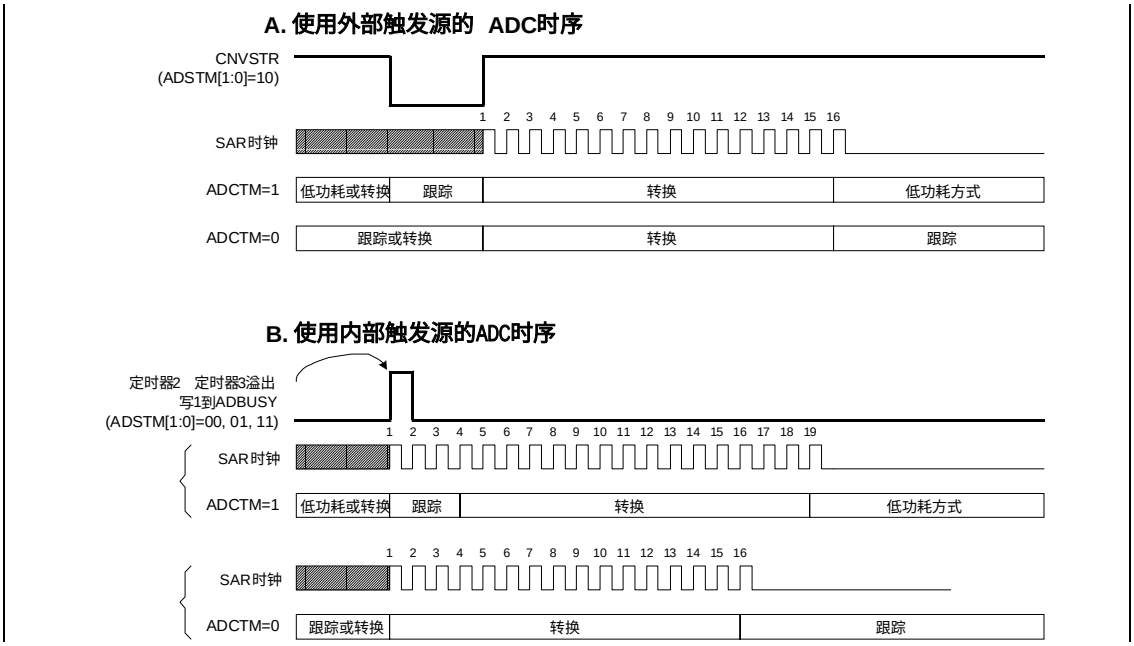


图 5.3 温度传感器传输函数

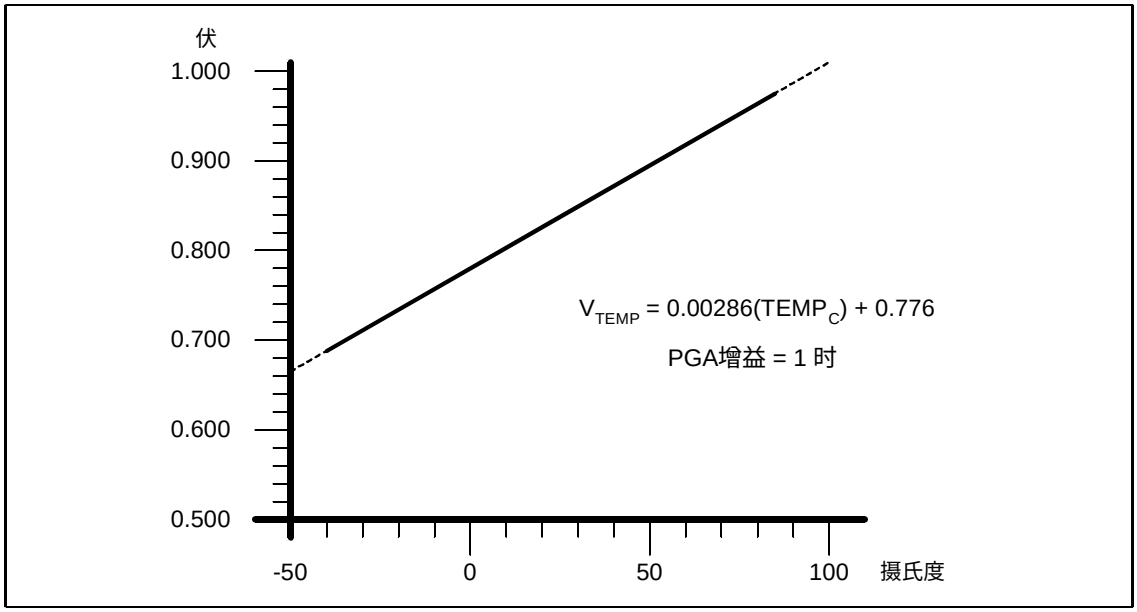


图 5.4 AMX0CF: AMUX 配置寄存器(C8051F00x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	AIN67IC	AIN45IC	AIN23IC	AIN01IC	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xBA

位 7-4： 未使用。读 = 0000b；写 = 忽略

位 3 AIN67IC：AIN6、AIN7 输入对配置位
0: AIN6 和 AIN7 为独立的单端输入
1: AIN6, AIN7 为（分别为）+, -差分输入对

位 2 AIN45IC：AIN4、AIN5 输入对配置位
0: AIN4 和 AIN5 为独立的单端输入
1: AIN4, AIN5 为（分别为）+, -差分输入对

位 1 AIN23IC：AIN2、AIN3 输入对配置位
0: AIN2 和 AIN3 为独立的单端输入
1: AIN2, AIN3 为（分别为）+, -差分输入对

位 0 AIN01IC：AIN0、AIN1 输入对配置位
0: AIN0 和 AIN1 为独立的单端输入
1: AIN0, AIN1 为（分别为）+, -差分输入对

注：对于被配置成差分输入的通道，ADC 数据字格式为 2 的补码。

图 5.5 AMUX0SL: AMUX 通道选择寄存器(C8051F00x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	AMXAD3	AMXAD2	AMXAD1	AMXAD0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xBB

位 7-4： 未使用。读 = 0000b；写 = 忽略
位 3-0： AMXAD3-0: AMUX 地址位
0000-1111: 按下表选择 ADC 输入

		AMXAD3-0								
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1xxx**
AMX0CF BITS 3-0	**0000**	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	温度传感器
	0001	+(AIN0) -(AIN1)		AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	温度传感器
	0010	AIN0	AIN1	+(AIN2) -(AIN3)		AIN4	AIN5	AIN6	AIN7	温度传感器
	0011	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		AIN4	AIN5	AIN6	AIN7	温度传感器
	0100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)		AIN6	AIN7	温度传感器
	0101	+(AIN0) -(AIN1)		AIN2	AIN3	+(AIN4) -(AIN5)		AIN6	AIN7	温度传感器
	0110	AIN0	AIN1	+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		AIN6	AIN7	温度传感器
	0111	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		AIN6	AIN7	温度传感器
	1000	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)		温度传感器
	1001	+(AIN0) -(AIN1)		AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)		温度传感器
	1010	AIN0	AIN1	+(AIN2) -(AIN3)		AIN4	AIN5	+(AIN6) -(AIN7)		温度传感器
	1011	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		AIN4	AIN5	+(AIN6) -(AIN7)		温度传感器
	1100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		温度传感器
	1101	+(AIN0) -(AIN1)		AIN2	AIN3	+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		温度传感器
	1110	AIN0	AIN1	+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		温度传感器
	1111	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		温度传感器

图 5.6 ADC0CF: ADC 配置寄存器(C8051F00x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
ADCSC2	ADCSC1	ADCSC0	-	-	AMPGN2	AMPGN1	AMPDN0	01100000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xBC

位 7-5： ADCSC2-0: ADC SAR 转换时钟周期控制位
000: SAR 转换时钟 = 1 个系统时钟
001: SAR 转换时钟 = 2 个系统时钟
010: SAR 转换时钟 = 4 个系统时钟
011: SAR 转换时钟 = 8 个系统时钟
1xx: SAR 转换时钟 = 16 个系统时钟
(注：SAR 转换时钟应 ≤ 2MHz)

位 4-3： 未使用。读 = 00b；写 = 忽略

位 2-0： AMPGN2-0: ADC 内部放大器增益
000: 增益 = 1
001: 增益 = 2
010: 增益 = 4
011: 增益 = 8
10x: 增益 = 16
11x: 增益 = 0.5

图 5.7 ADC0CN: ADC 控制寄存器(C8051F00x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
ADCEN	ADCTM	ADCINT	ADBUSY	ADCSTM1	ADCSTM0	ADWINT	ADLJST	00000000
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址: 0xE8

位 7： ADCEN：ADC 允许位
0：ADC 禁止。ADC 处于低耗停机状态。
1：ADC 允许。ADC 处于活动状态，并准备转换数据。

位 6： ADCTM：ADC 跟踪方式位
0：当 ADC 被允许时，除了转换期间之外一直处于跟踪方式。
1：由 ADSTM1-0 定义跟踪方式：
ADSTM1-0：
00：向 ADBUSY 写 1 时启动跟踪，持续 3 个 SAR 时钟
01：定时器 3 溢出启动跟踪，持续 3 个 SAR 时钟
10：只有当 CNVSTR 输入为逻辑低电平时启动 ADC 跟踪
11：定时器 2 溢出启动跟踪，持续 3 个 SAR 时钟

位 5： ADCINT：ADC 转换结束中断标志（必须用软件清 0）
0：从最后一次将该位清 0 后，ADC 还没有完成一次数据转换
1：ADC 完成一次数据转换

位 4： ADCBUSY：ADC 忙标志位
读
0：ADC 转换结束或复位以来没有有效的数据转换。当被允许时，ADCBUSY 的下降沿触发中断。
1：ADC 正在转换。
写
0：无效
1：若 ADSTM1-0＝00b 则启动 ADC 转换

位 3-2： ADSTM1-0：ADC 转换启动方式位
00：向 ADBUSY 写 1 启动 ADC 转换
01：定时器 3 溢出启动 ADC 转换
10：CNVSTR 上升沿启动 ADC 转换
11：定时器 2 溢出启动 ADC 转换

位 1： ADWINT：ADC 窗口比较中断标志（必须用软件清 0）
0：未发生 ADC 窗口比较匹配
1：发生 ADC 窗口比较匹配

位 0： ADLJST：ADC 数据左对齐控制位
0：ADC0H:ADC0L 寄存器数据右对齐
1：ADC0H:ADC0L 寄存器数据左对齐

图 5.8 ADC0H: ADC 数据字 MSB 寄存器(C8051F00x)

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： ADC 数据位
当 ADLJST＝1：12 位 ADC 数据字的高 8 位。
当 ADLJST＝0：位 7-4 为位 3 的符号扩展位。位 3-0 是 12 位 ADC 数据字的高 4 位。

图 5.9 ADC0L: ADC 数据字 LSB 寄存器(C8051F00x)

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

位 7-0： ADC 数据位
当 ADLJST＝1：位 7-4 是 12 位 ADC 数据字的低 4 位。位 3-0 读出值总是为 0。
当 ADLJST＝0：位 7-0 是 12 位 ADC 数据字的低 8 位。

注： 12 位 ADC 结果数据字在 ADC 数据字寄存器中存放如下：
ADC0H[3:0]:ADC0L[7:0]，如果 ADLJST＝0。
（如果是差分输入，ADC0H[7:4]为 ADC0H.3 的符号扩展位，否则=0000b）

ADC0H[7:0]:ADC0L[7:4]，如果 ADLJST＝1。
（ADC0L[3:0]=0000b）

例：ADC 数据字转换表，AIN0 为单端输入方式
（AMX0CF=0x00, AMX0SL=0x00）

AIN0-AGND(伏)	**ADC0H:ADC0L** (ADLJST=0)	**ADC0H:ADC0L** (ADLJST=1)
REF x (4095/4096)	0x0FFF	0xFFF0
REF x ½	0x0800	0x8000
REF x (2047/4096)	0x07FF	0x7FF0
0	0x0000	0x0000

例：ADC 数据字转换表，AIN0-AIN1 为差分输入对
（AMX0CF=0x01, AMX0SL=0x00）

AIN0-AIN1(伏)	**ADC0H:ADC0L** (ADLJST=0)	**ADC0H:ADC0L** (ADLJST=1)
REF x (2047/2048)	0x07FF	0x7FF0
0	0x0000	0x0000
-REF x (1/2048)	0xFFFF	0xFFF0
-REF	0xF800	0x8000

5.3 ADC 可编程窗口检测器

ADC 可编程窗口检测器在很多应用中非常有用。它不停地将 ADC 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中尤其有效，既可以节省代码空间和 CPU 带宽，又能提供快速响应时间。窗口检测器中断标志（ADC0CN 中的 ADWINT 位）也可被用于查询方式。参考字的高和低字节被装入到 ADC 下限（大于）和 ADC 上限（小于）寄存器（ADC0GTH、ADC0GTL、ADC0LTH 和 ADC0LTL）。图 5.14 和图 5.15 给出了比较示例供参考。注意，窗口检测器标志既可以在测量数据位于用户编程的极限值以内时有效，也可以在测量数据位于用户编程的极限值以外时有效，这取决于 ADC0GTx 和 ADC0LTx 的编程值。

图 5.10 ADC0GTH: ADC 下限数据高字节寄存器（C8051F00x)

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：0xC5

位 7-0：
ADC 下限数据字的高字节

图 5.11 ADC0GTL: ADC 下限数据低字节寄存器（C8051F00x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xC4

位 7-0：
ADC 下限数据字的低字节

定义：
ADC 下限数据字=ADC0GTH:ADC0GTL

图 5.12 ADC0LTH: ADC 上限数据高字节寄存器(C8051F00x)

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：0xC7

位 7-0：
ADC 数据上限数据字的高字节

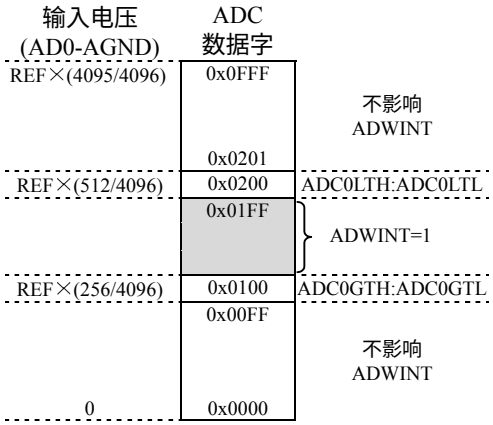
图 5.13 ADC0LTL: ADC 上限数据低字节寄存器(C8051F00x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xC6

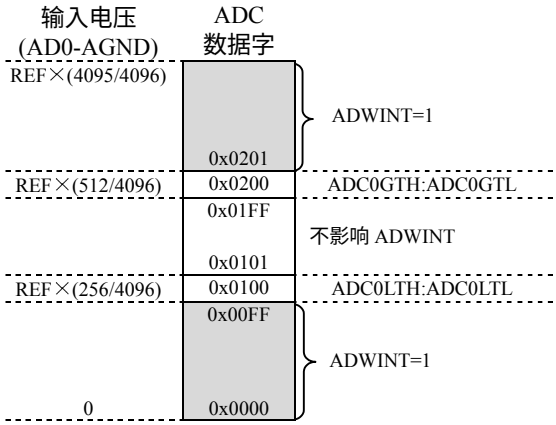
位 7-0：
ADC 上限数据字的低字节

定义：
ADC 上限数据字=ADC0LTH:ADC0LTL

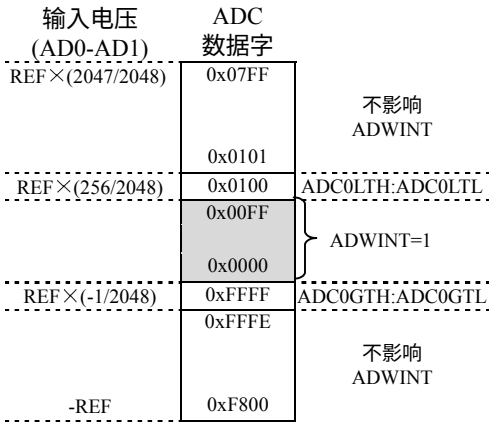
图 5.14 12 位 ADC 窗口中断示例，数据右对齐



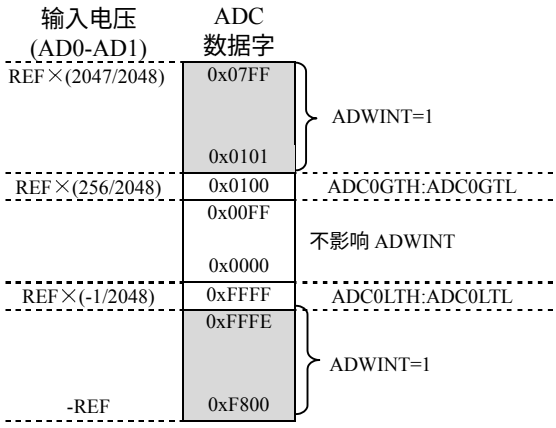
给定：
AMX0SL＝0x00, AMX0CF=0x00, ADLJST=0,
ADC0LTH:ADC0LTL=0x0200,
ADC0GTH:ADC0GTL=0x0100.
如果 0x0100<ADC 结果数据字＜0x0200，则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。



给定：
AMX0SL＝0x00,AMX0CF=0x00,ADLJST=0,
ADC0LTH:ADC0LTL=0x0100,
ADC0GTH:ADC0GTL=0x0200.
如果 ADC 数据字<0x0100 或>0x0200，则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。

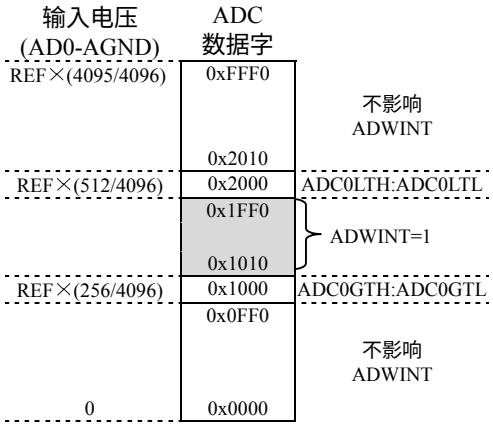


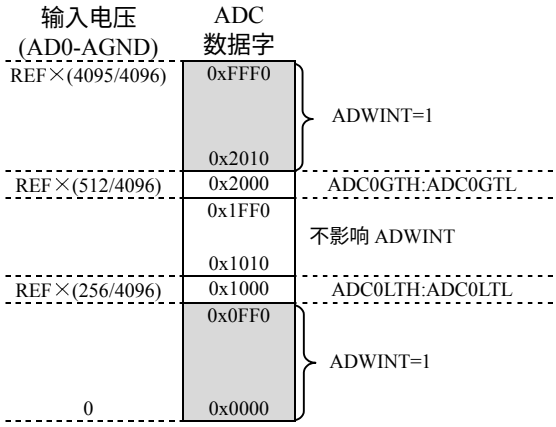
给定：
AMX0SL＝0x00, AMX0CF=0x01, ADLJST=0,
ADC0LTH:ADC0LTL=0x0100,
ADC0GTH:ADC0GTL=0xFFFF.
如果 0xFFFF<ADC 结果数据字＜0x0100（2 的补码，0xFFFF=-1），则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。



给定：
AMX0SL＝0x00, AMX0CF=0x01, ADLJST=0,
ADC0LTH:ADC0LTL=0xFFFF,
ADC0GTH:ADC0GTL=0x0100.
如果 ADC 结果数据字＜0xFFFF 或＞0x0100（2 的补码，0xFFFF=-1），则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。

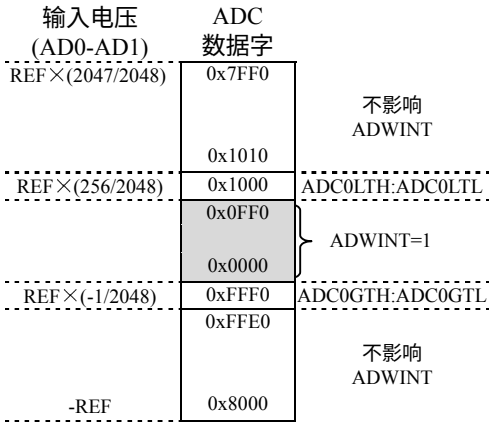
图 5.15 12 位 ADC 窗口中断示例，数据左对齐

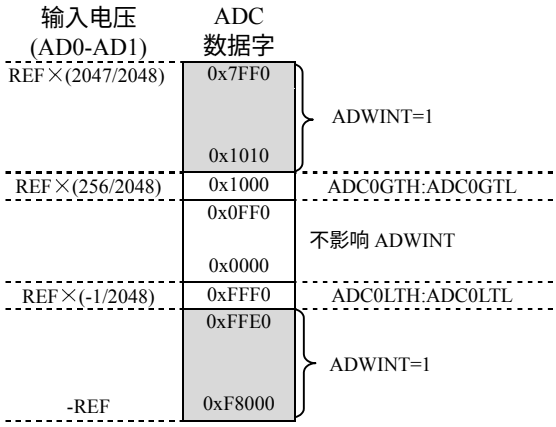




给定：
AMX0SL＝0x00, AMX0CF=0x00, ADLJST=1,
ADC0LTH:ADC0LTL=0x2000,
ADC0GTH:ADC0GTL=0x1000。
如果 0x100<ADC 数据字＜0x2000，则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。

给定：
AMX0SL＝0x00, AMX0CF=0x00, ADLJST=1,
ADC0LTH:ADC0LTL=0x1000,
ADC0GTH:ADC0GTL=0x2000。
如果 ADC 数据字<0x1000 或>0x02000，则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。





给定：
AMX0SL＝0x00,AMX0CF=0x01,ADLJST=1,
ADC0LTH:ADC0LTL=0x1000,
ADC0GTH:ADC0GTL=0xFFF0.
ADC 转换结束，如果 0xFFF0<ADC 数据字＜0x1000（2 的补码），则会触发 ADC 窗口比较中断（ADWINT＝1）。

给定：
AMX0SL＝0x00,AMX0CF=0x01,ADLJST=0,
ADC0LTH:ADC0LTL=0xFFF0,
ADC0GTH:ADC0GTL=0x1000.
ADC 转换结束，如果数据字＜0xFFF0 或＞0x1000（2 的补码），则会触发 ADC 窗口比较中断（ADWINT＝1）。

表 5.1　12 位 ADC 电气特性

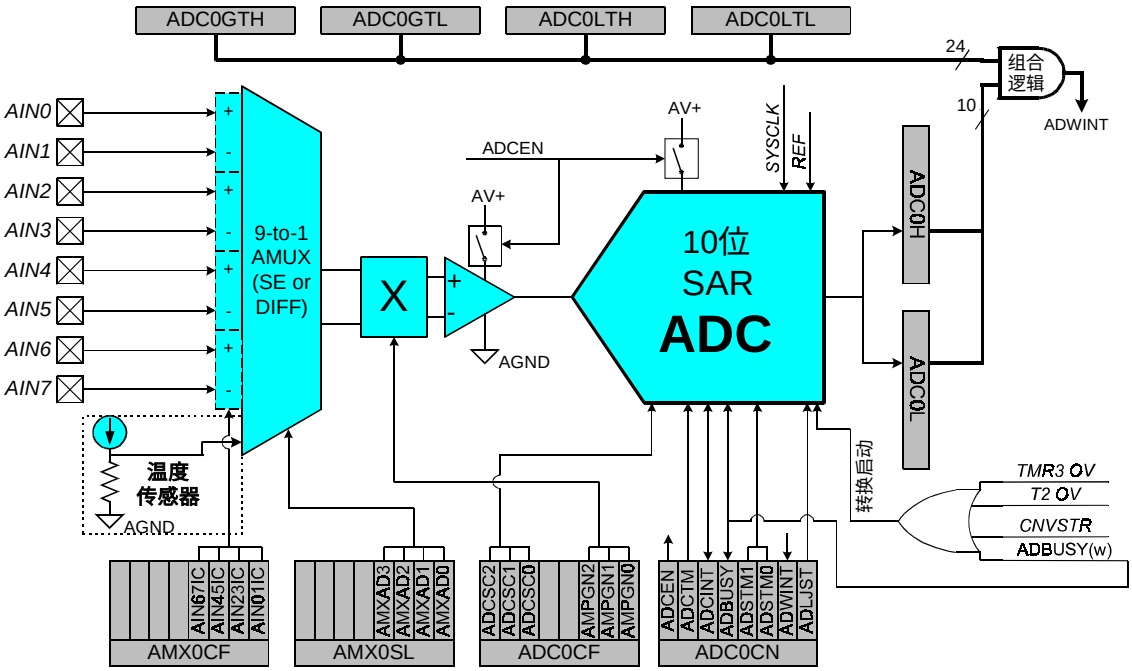
VDD=3.0V, AV+=3.0V, VREF=2.40V(REFBE=0), PGA 增益=1, -40°C 到+85°C（除非另有说明）

参　数	条　件	最小值	典型值	最大值	单 位
直流精度					
分辨率			12		位
积分非线性				±1	LSB
微分非线性	保证单调			±1	LSB
偏移误差			-3±1		LSB
满度误差	差分方式		-7±3		LSB
温度漂移系数			±0.25		ppm/℃
动态性能（10kHz 正弦波输入，满度值的 0 到-1dB，100ksps）					
信号与噪声加失真比		66	69		dB
总谐波失真	到 5 次谐波		-75		dB
有效动态范围			80		dB
转换速率					
转换时间（SAR 时钟数）		16			周期
SAR 时钟频率	C8051F000/001/002 C8051F005/006/007			2.0 2.5	MHz MHz
跟踪/保持捕获时间		1.5			μS
转换速率				100	ksps
模拟输入					
电压转换范围	单端方式（AINn-AGND） 差分方式|(AINn+)－(AINm-)|	0		VREF -1LSB	V
输入电压	所有 AINn 引脚	AGND		AV+	V
输入电容			10		pF
温度传感器					
线性度			±0.20		℃
绝对精度			±3		℃
增益	PGA 增益＝1		2.86		mV/ ℃
增益误差（±1σ）	PGA 增益＝1		±33.5		μV/ ℃
偏移	PGA 增益＝1，温度 ＝0℃		776		mV
偏移误差（±1σ）	PGA 增益＝1，温度 ＝0℃		±8.51		mV
电源指标					
电源电流（AV+给 ADC 供电）	工作方式，100ksps		450	900	μA
电源抑制比			±0.3		mV/V

6. ADC（10 位，只适用于 C8051F010/1/2/5/6/7）

C8051F010/1/2/5/6/7 的 ADC 子系统包括一个 9 通道的可配置模拟多路开关（AMUX），一个可编程增益放大器（PGA）和一个 100ksps、10 位分辨率的逐次逼近寄存器型 ADC，ADC 中集成了跟踪保持电路和可编程窗口检测器（见图 6.1 的原理框图）。AMUX、PGA、数据转换方式及窗口检测器都可用软件通过图 6.1 所示的特殊功能寄存器来配置。只有当 ADC 控制寄存器（ADC0CN，图 6.7）中的 ADCEN 位被置‘1’时 ADC 子系统（ADC、跟踪保持器和 PGA）才被允许工作。当 ADCEN 位为‘0’时，ADC 子系统处于低功耗关断方式。如果要给 ADC 提供偏置，必须将 REF0CN 寄存器（见图 9.2）中的偏置允许位（BIASE）置‘1’。

图 6.1 10 位 ADC 功能框图



6.1 模拟多路开关和 PGA

AMUX 中的 8 个通道用于外部测量，而第九通道在内部被接到片内温度传感器（温度传输函数示于图 6.3）。注意，PGA 的增益对温度传感器也起作用。可以将 AMUX 输入对编程为工作在差分或单端方式。这就允许用户对每个通道选择最佳的测量技术，甚至可以在测量过程中改变方式。在系统复位后 AMUX 的默认方式为单端输入。有两个与 AMUX 相关的寄存器：通道选择寄存器 AMX0SL（图 6.5）和配置寄存器 AMX0CF（图 6.4）。图 6.5 中的表给出了每种配置下各通道的功能。PGA 对 AMUX 输出信号的放大倍数由 ADC 配置寄存器 ADC0CF（图 6.6）中的 AMPGN2-0 确定。PGA 增益可以用软件编程为 0.5、1、2、4、8 或 16。复位时的默认增益为 1。

6.2 ADC 的工作方式

ADC 使用 VREF 来确定它的满度电压，因此在进行一次转换之前必须正确设置这个参考电压（见第九章）。ADC 的最高转换速度为 100ksps，转换时钟来源于系统时钟。可以通过设置 ADC0CF 寄存器的 ADCSC 位将转换时钟的速度降为系统时钟的 1/2、1/4、1/8 或 1/16。这一功能用于根据不同的系统时钟速度调整转换速度。

有 4 种 A/D 转换启动方式，由 ADC0CN 的 ADC 启动转换方式位（ADSTM1，ADSTM0）的状态决定。转换触发源有：

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

向 ADBUSY 写 1 方式提供了在需要时用软件控制 ADC 启动转换的能力。ADBUSY 位在转换期间被置‘1’，转换结束后复‘0’。ADBUSY 位的下降沿触发一个中断（当被允许时）并置‘1’ADC0CN 中的中断标志。转换结果保存在 ADC 数据字的 MSB 和 LSB 寄存器：ADC0H 和 ADC0L。转换数据在寄存器对 ADC0H:ADC0L 中的存储方式可以是左对齐或右对齐，由 ADC0CN 寄存器中 ADLJST 位的编程状态决定。

ADC0CN 中的 ADCTM 位控制 ADC 的跟踪保持方式。在缺省状态，ADC 输入被连续跟踪（转换期间除外）。将 ADCTM 位设置为‘1’即可采用下面的 4 种低功耗跟踪保持方式，由 ADSTM1-0 位（也在 ADC0CN 中）的状态决定采用哪一种方式。

1. 从向 ADBUSY 写‘1’开始跟踪，持续 3 个 SAR 时钟；
2. 从定时器 3 溢出开始跟踪，持续 3 个 SAR 时钟；
3. 只在 CNVSTR 输入为低电平时跟踪；
4. 从定时器 2 溢出开始跟踪，持续 3 个 SAR 时钟；

上述的方式 1、2 和 4 在使用软件命令触发转换或 ADC 连续工作时有用。方式 3 用于需要使用外部硬件触发转换的场合。在方式 3，当 CNVSTR 输入信号为高电平时跟踪保持电路处于低功耗状态。当整个芯片处于低功耗停机或休眠方式时，可以禁止跟踪（关断）。

图 6.2　10 位 ADC 跟踪和转换时序举例

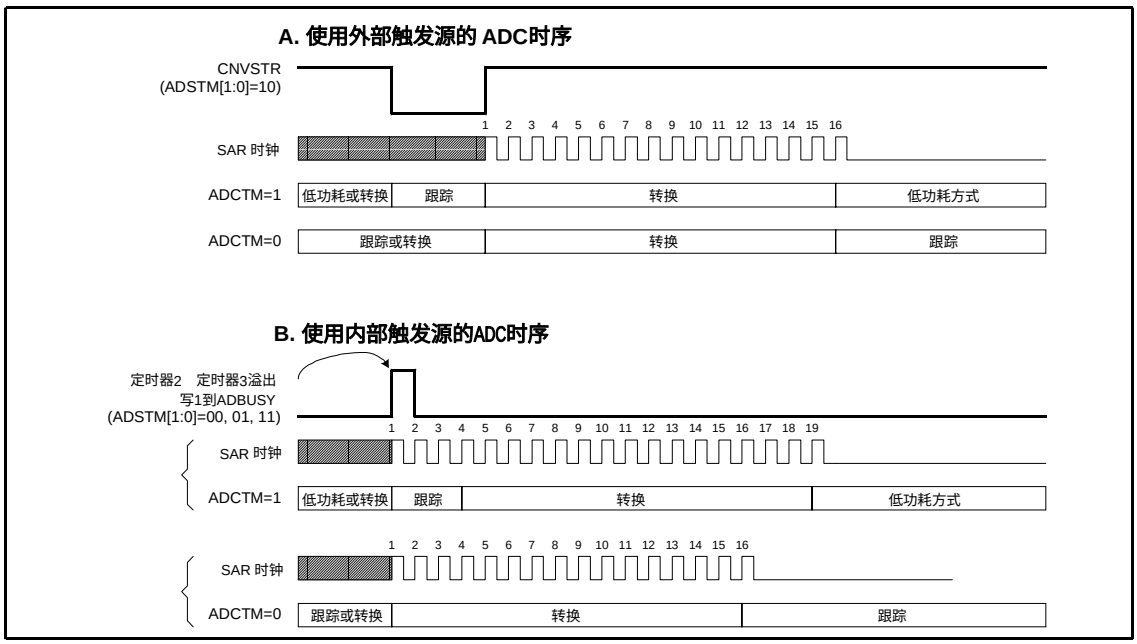


图 6.3 温度传感器传输函数

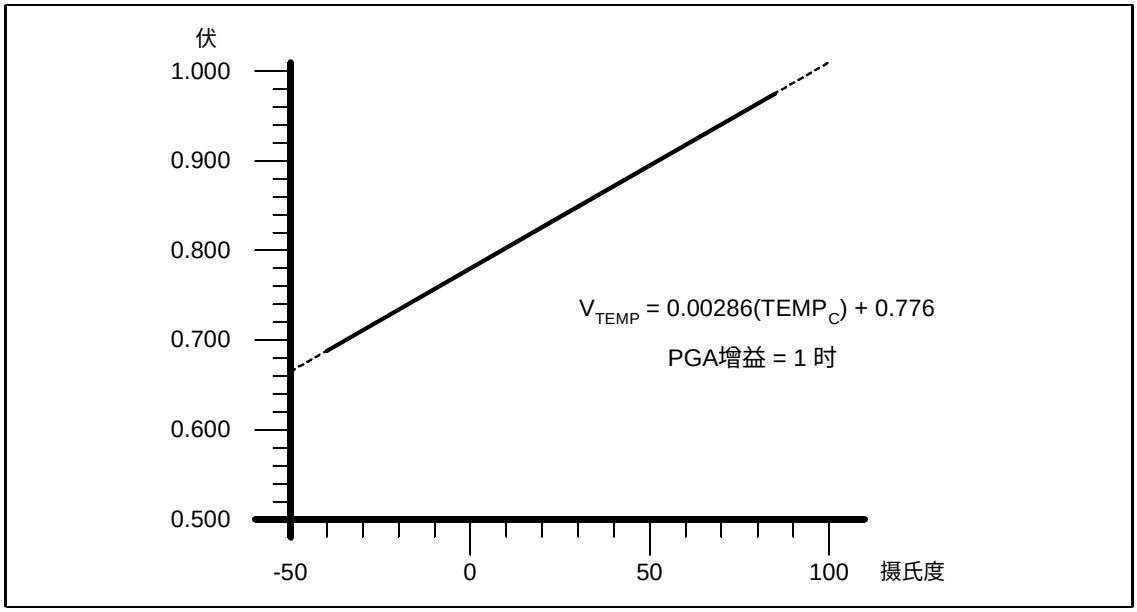


图 6.4 AMX0CF: AMUX 配置寄存器(C8051F01x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	AIN67IC	AIN45IC	AIN23IC	AIN01IC	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xBA

位 7-4： 未使用。读 = 0000b；写 = 忽略

位 3 AIN67IC：AIN6、AIN7 输入对配置位
0: AIN6 和 AIN7 为独立的单端输入
1: AIN6, AIN7 为（分别为）+, -差分输入对

位 2 AIN45IC：AIN4、AIN5 输入对配置位
0: AIN4 和 AIN5 为独立的单端输入
1: AIN4, AIN5 为（分别为）+, -差分输入对

位 1 AIN23IC：AIN2、AIN3 输入对配置位
0: AIN2 和 AIN3 为独立的单端输入
1: AIN2, AIN3 为（分别为）+, -差分输入对

位 0 AIN01IC：AIN0、AIN1 输入对配置位
0: AIN0 和 AIN1 为独立的单端输入
1: AIN0, AIN1 为（分别为）+, -差分输入对

注：对于被配置成差分输入的通道，ADC 数据字格式为 2 的补码。

图 6.5 AMUX0SL: AMUX 通道选择寄存器(C8051F01x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	AMXAD3	AMXAD2	AMXAD1	AMXAD0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xBB

位 7-4： 未使用。读 = 0000b；写 = 忽略
位 3-0： AMXAD3-0: AMUX 地址位
0000-1111: 按下表选择 ADC 输入

		AMXAD3-0								
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1xxx**
AMX0CF BITS 3-0	**0000**	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	温度传感器
	0001	+(AIN0) -(AIN1)		AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	温度传感器
	0010	AIN0	AIN1	+(AIN2) -(AIN3)		AIN4	AIN5	AIN6	AIN7	温度传感器
	0011	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		AIN4	AIN5	AIN6	AIN7	温度传感器
	0100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)		AIN6	AIN7	温度传感器
	0101	+(AIN0) -(AIN1)		AIN2	AIN3	+(AIN4) -(AIN5)		AIN6	AIN7	温度传感器
	0110	AIN0	AIN1	+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		AIN6	AIN7	温度传感器
	0111	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		AIN6	AIN7	温度传感器
	1000	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)		温度传感器
	1001	+(AIN0) -(AIN1)		AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)		温度传感器
	1010	AIN0	AIN1	+(AIN2) -(AIN3)		AIN4	AIN5	+(AIN6) -(AIN7)		温度传感器
	1011	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		AIN4	AIN5	+(AIN6) -(AIN7)		温度传感器
	1100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		温度传感器
	1101	+(AIN0) -(AIN1)		AIN2	AIN3	+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		温度传感器
	1110	AIN0	AIN1	+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		温度传感器
	1111	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		温度传感器

图 6.6 ADC0CF: ADC 配置寄存器(C8051F01x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
ADCSC2	ADCSC1	ADCSC0	-	-	AMPGN2	AMPGN1	AMPDN0	01100000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xBC

位 7-5： ADCSC2-0: ADC SAR 转换时钟周期控制位
000: SAR 转换时钟 = 1 个系统时钟
001: SAR 转换时钟 = 2 个系统时钟
010: SAR 转换时钟 = 4 个系统时钟
011: SAR 转换时钟 = 8 个系统时钟
1xx: SAR 转换时钟 = 16 个系统时钟
(注：SAR 转换时钟应 ≤ 2MHz)

位 4-3： 未使用。读 = 00b；写 = 忽略

位 2-0： AMPGN2-0: ADC 内部放大器增益
000: 增益 = 1
001: 增益 = 2
010: 增益 = 4
011: 增益 = 8
10x: 增益 = 16
11x: 增益 = 0.5

图 6.7 ADC0CN: ADC 控制寄存器(C8051F01x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
ADCEN	ADCTM	ADCINT	ADBUSY	ADCSTM1	ADCSTM0	ADWINT	ADLJST	00000000
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址: 0xE8

位 7： ADCEN：ADC 允许位
0：ADC 禁止。ADC 处于低耗停机状态。
1：ADC 允许。ADC 处于活动状态，并准备转换数据。

位 6： ADCTM：ADC 跟踪方式位
0：当 ADC 被允许时，除了转换期间之外一直处于跟踪方式。
1：由 ADSTM1-0 定义跟踪方式：
ADSTM1-0：
00：向 ADBUSY 写 1 时启动跟踪，持续 3 个 SAR 时钟
01：定时器 3 溢出启动跟踪，持续 3 个 SAR 时钟
10：只有当 CNVSTR 输入为逻辑低电平时启动 ADC 跟踪
11：定时器 2 溢出启动跟踪，持续 3 个 SAR 时钟

位 5： ADCINT：ADC 转换结束中断标志（必须用软件清 0）
0：从最后一次将该位清 0 后，ADC 还没有完成一次数据转换
1：ADC 完成一次数据转换

位 4： ADCBUSY：ADC 忙标志位
读
0：ADC 转换结束或复位以来没有有效的数据转换。当被允许时，ADCBUSY 的下降沿触发中断。
1：ADC 正在转换。
写
0：无效
1：若 ADSTM1-0＝00b 则启动 ADC 转换

位 3-2： ADSTM1-0：ADC 转换启动方式位
00：向 ADBUSY 写 1 启动 ADC 转换
01：定时器 3 溢出启动 ADC 转换
10：CNVSTR 上升沿启动 ADC 转换
11：定时器 2 溢出启动 ADC 转换

位 1： ADWINT：ADC 窗口比较中断标志（必须用软件清 0）
0：未发生 ADC 窗口比较匹配
1：发生 ADC 窗口比较匹配

位 0： ADLJST：ADC 数据左对齐控制位
0：ADC0H:ADC0L 寄存器数据右对齐
1：ADC0H:ADC0L 寄存器数据左对齐

图 6.8　ADC0H: ADC 数据字 MSB 寄存器(C8051F01x)

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：　ADC 数据位
当 ADLJST＝1：10 位 ADC 数据字的高 8 位。
当 ADLJST＝0：位 7-2 为位 1 的符号扩展位。位 1-0 是 10 位 ADC 数据字的高 2 位。

图 6.9　ADC0L: ADC 数据字 LSB 寄存器(C8051F01x)

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

位 7-0：　ADC 数据位
当 ADLJST＝1：位 7-6 是 10 位 ADC 数据字的低 2 位。位 5-0 读出值总是为 0。
当 ADLJST＝0：位 7-0 是 10 位 ADC 数据字的低 8 位。

注：　10 位 ADC 结果数据字在 ADC 数据字寄存器中存放如下：
ADC0H[1:0]:ADC0L[7:0]，如果 ADLJST＝0。
（如果是差分输入，ADC0H[7:2]为 ADC0H.1 的符号扩展位，否则=0000b）

ADC0H[7:0]:ADC0L[7:6]，如果 ADLJST＝1。
（ADC0L[5:0]=0000b）

例：ADC 数据字转换表，AIN0 为单端输入方式
（AMX0CF=0x00, AMX0SL=0x00）

AIN0-AGND(伏)	**ADC0H:ADC0L** (ADLJST=0)	**ADC0H:ADC0L** (ADLJST=1)
REF x (1023/1024)	0x03FF	0xFFC0
REF x ½	0x0200	0x8000
REF x (511/1024)	0x01FF	0x7FC0
0	0x0000	0x0000

例：ADC 数据字转换表，AIN0-AIN1 为差分输入对
（AMX0CF=0x01, AMX0SL=0x00）

AIN0-AIN1(伏)	**ADC0H:ADC0L** (ADLJST=0)	**ADC0H:ADC0L** (ADLJST=1)
REF x (511/512)	0x01FF	0x7FC0
0	0x0000	0x0000
-REF x (1/512)	0xFFFF	0xFFC0
-REF	0xFE00	0x8000

6.3 ADC 可编程窗口检测器

ADC 可编程窗口检测器在很多应用中非常有用。它不停地将 ADC 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中尤其有效，既可以节省代码空间和 CPU 带宽又能提供快速响应时间。窗口检测器中断标志（ADC0CN 中的 ADWINT 位）也可被用于查询方式。参考字的高和低字节被装入到 ADC 下限（大于）和 ADC 上限（小于）寄存器（ADC0GTH、ADC0GTL、ADC0LTH 和 ADC0LTL）。图 6.12 和图 6.13 给出了比较示例供参考。注意，窗口检测器标志既可以在测量数据位于用户编程的极限值以内时有效，也可以在测量数据位于用户编程的极限值以外时有效，这取决于 ADC0GTx 和 ADC0LTx 的编程值。

图 6.10 ADC0GTH: ADC 下限数据高字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xC5

位 7-0：
ADC 下限数据字的高字节

图 6.11 ADC0GTL: ADC 下限数据低字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								11111111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xC4

位 7-0：
ADC 下限数据字的低字节

定义：
ADC 下限数据字=ADC0GTH:ADC0GTL

图 6.12 ADC0LTH: ADC 上限数据高字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xC7

位 7-0：
ADC 数据上限数据字的高字节

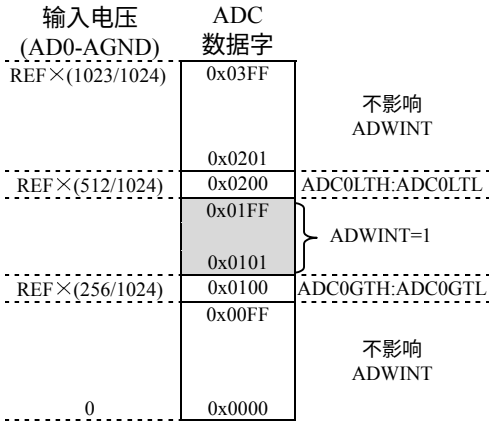
图 6.13 ADC0LTL: ADC 上限数据低字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xC6

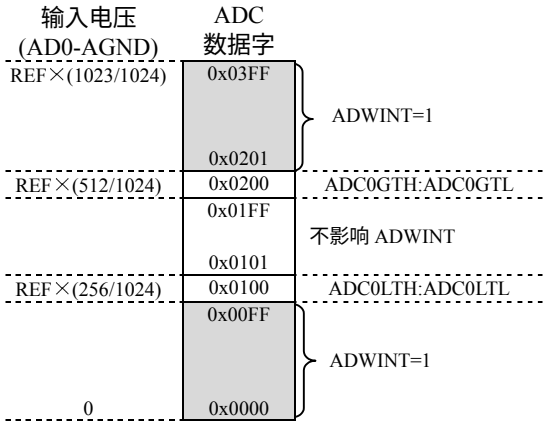
位 7-0：
ADC 上限数据字的低字节

定义：
ADC 上限数据字=ADC0LTH:ADC0LTL

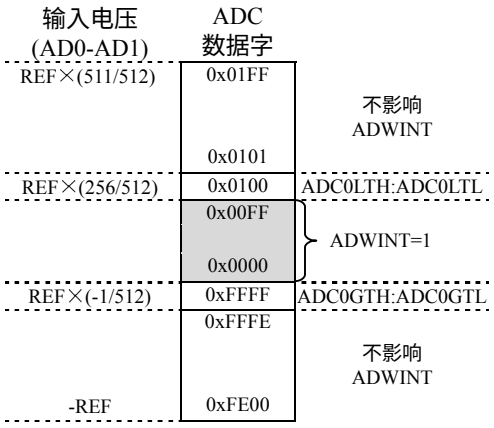
图 6.14　10 位 ADC 窗口中断示例，数据右对齐



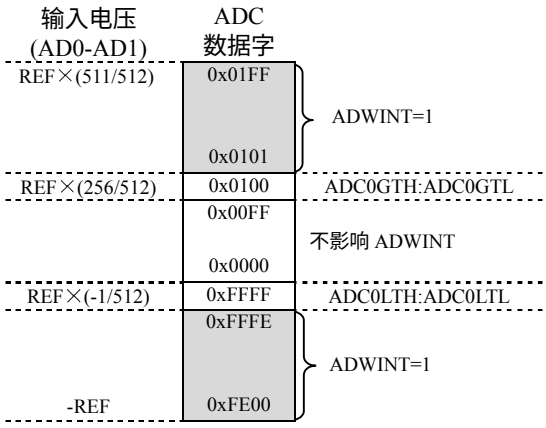
给定：
AMX0SL＝0x00, AMX0CF=0x00, ADLJST=0,
ADC0LTH:ADC0LTL=0x0200,
ADC0GTH:ADC0GTL=0x0100.
如果 0x0100<ADC 结果数据字＜0x0200，则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。



给定：
AMX0SL＝0x00,AMX0CF=0x00,ADLJST=0,
ADC0LTH:ADC0LTL=0x0100,
ADC0GTH:ADC0GTL=0x0200.
如果 ADC 数据字<0x0100 或>0x0200，则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。

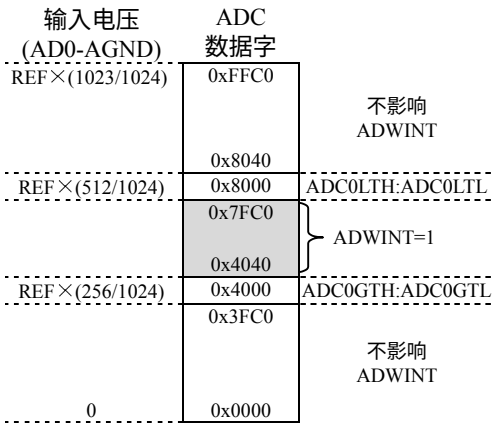


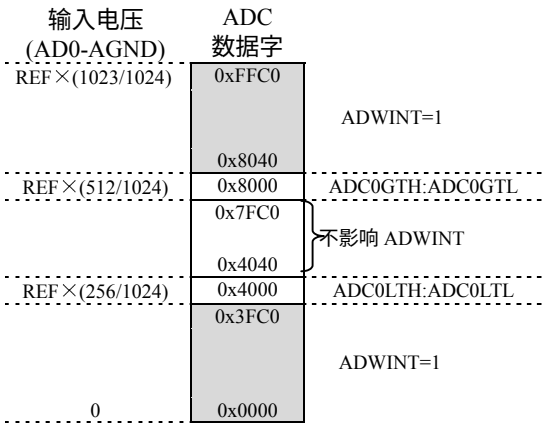
给定：
AMX0SL＝0x00, AMX0CF=0x01, ADLJST=0,
ADC0LTH:ADC0LTL=0x0100,
ADC0GTH:ADC0GTL=0xFFFF.
如果 0xFFFF<ADC 结果数据字＜0x0100（2 的补码，0xFFFF=-1），则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。



给定：
AMX0SL＝0x00, AMX0CF=0x01, ADLJST=0,
ADC0LTH:ADC0LTL=0xFFFF,
ADC0GTH:ADC0GTL=0x0100.
如果 ADC 结果数据字＜0xFFFF 或＞0x0100（2 的补码，0xFFFF=-1），则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。

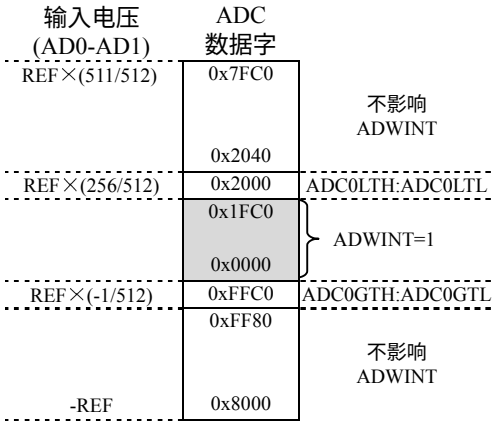
图 6.15　10 位 ADC 窗口中断示例，数据左对齐

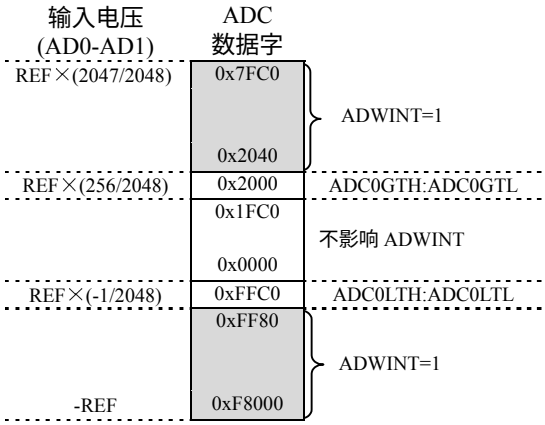




给定：
AMX0SL＝0x00, AMX0CF=0x00, ADLJST=1,
ADC0LTH:ADC0LTL=0x8000,
ADC0GTH:ADC0GTL=0x4000。
如果 0x4000<ADC 数据字＜0x8000，则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。

给定：
AMX0SL＝0x00, AMX0CF=0x00, ADLJST=1,
ADC0LTH:ADC0LTL=0x4000,
ADC0GTH:ADC0GTL=0x8000。
如果 ADC 数据字<0x4000 或>0x8000，则 ADC 转换结束会触发 ADC 窗口比较中断（ADWINT＝1）。





给定：
AMX0SL＝0x00,AMX0CF=0x01,ADLJST=1,
ADC0LTH:ADC0LTL=0x2000,
ADC0GTH:ADC0GTL=0xFFC0。
ADC 转换结束，如果 0xFFC0<ADC 数据字＜0x2000（2 的补码），则会触发 ADC 窗口比较中断（ADWINT＝1）。

给定：
AMX0SL＝0x00,AMX0CF=0x01,ADLJST=1,
ADC0LTH:ADC0LTL=0xFFC0,
ADC0GTH:ADC0GTL=0x2000。
ADC 转换结束，如果数据字＜0xFFC0 或＞0x2000（2 的补码），则会触发 ADC 窗口比较中断（ADWINT＝1）。

表 6.1. 10 位 ADC 电气特性

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

参　数	条　件	最小值	典型值	最大值	单 位
直流精度					
分辨率			10		位
积分非线性			±0.5	±1	LSB
微分非线性	保证单调		±0.5	±1	LSB
偏移误差			±0.5		LSB
满度误差	差分方式		-1.5±0.5		LSB
温度漂移系数			±0.25		ppm/℃
动态性能（10kHz 正弦波输入，满度值的 0 到-1dB，100ksps）					
信号与噪声加失真比		59	61		dB
总谐波失真	到 5 次谐波		-70		dB
有效动态范围			80		dB
转换速率					
转换时间（SAR 时钟数）		16			周期
SAR 时钟频率	C8051F010/011/012 C8051F015/016/017			2.0 2.5	MHz MHz
跟踪/保持捕获时间		1.5			μS
转换速率				100	ksps
模拟输入					
电压转换范围	单端方式（AINn-AGND） 差分方式|(AINn+)－(AINm-)|	0		VREF -1LSB	V
输入电压	所有 AINn 引脚	AGND		AV+	V
输入电容			10		pF
温度传感器					
线性度			±0.20		℃
绝对精度			±3		℃
增益	PGA 增益＝1		2.86		mV/ ℃
增益误差（±1σ）	PGA 增益＝1		±33.5		μV/ ℃
偏移	PGA 增益＝1，温度 ＝0℃		776		mV
偏移误差（±1σ）	PGA 增益＝1，温度 ＝0℃		±8.51		mV
电源指标					
电源电流（AV+给 ADC 供电）	工作方式，100ksps		450	900	μA
电源抑制比			±0.3		mV/V

7. 12 位电压输出 DAC

C8051F000 系列 MCU 有两个 12 位的电压方式 DAC。每个 DAC 的输出摆幅均为 0V 到 VREF-1LSB，对应的输入码范围是 0x000 到 0xFFF。以 DAC0 为例，12 位的数据字被写到低字节（DAC0L）和高字节（DAC0H）数据寄存器。在写 DAC0H 寄存器时数据被锁存到 DAC0，所以如果需要 12 位分辨率，应在写入 DAC0L 之后写 DAC0H。DAC 可被用于 8 位方式，这种情况是将数据左移后只写入 DAC0H，而在 DAC0L 中写入一个所希望的数值（通常为 0x00）。DAC0 控制寄存器（DAC0CN）提供允许/禁止 DAC0 和改变输入数据格式的手段。

DAC0 的允许/禁止功能由 DAC0EN 位控制（DAC0CN.7）。向 DAC0EN 写 1 允许 DAC0 工作，向 DAC0EN 写 0 则禁止 DAC0。在被禁止时，DAC0 的输出保持在高阻状态，DAC0 的供电电流降到 1μA 或更小。为了给 DAC0 提供偏置，必须将 REF0CN 寄存器（见图 9.2）中的偏置允许位（BIASE）置 1。还必须正确设置 DAC0 的电压基准（见第 9 章）。

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

DAC1 的功能与上述 DAC0 的功能完全相同。表 7.1 给出了 DAC0 和 DAC1 的电气特性。

图 7.1　DAC 功能框图

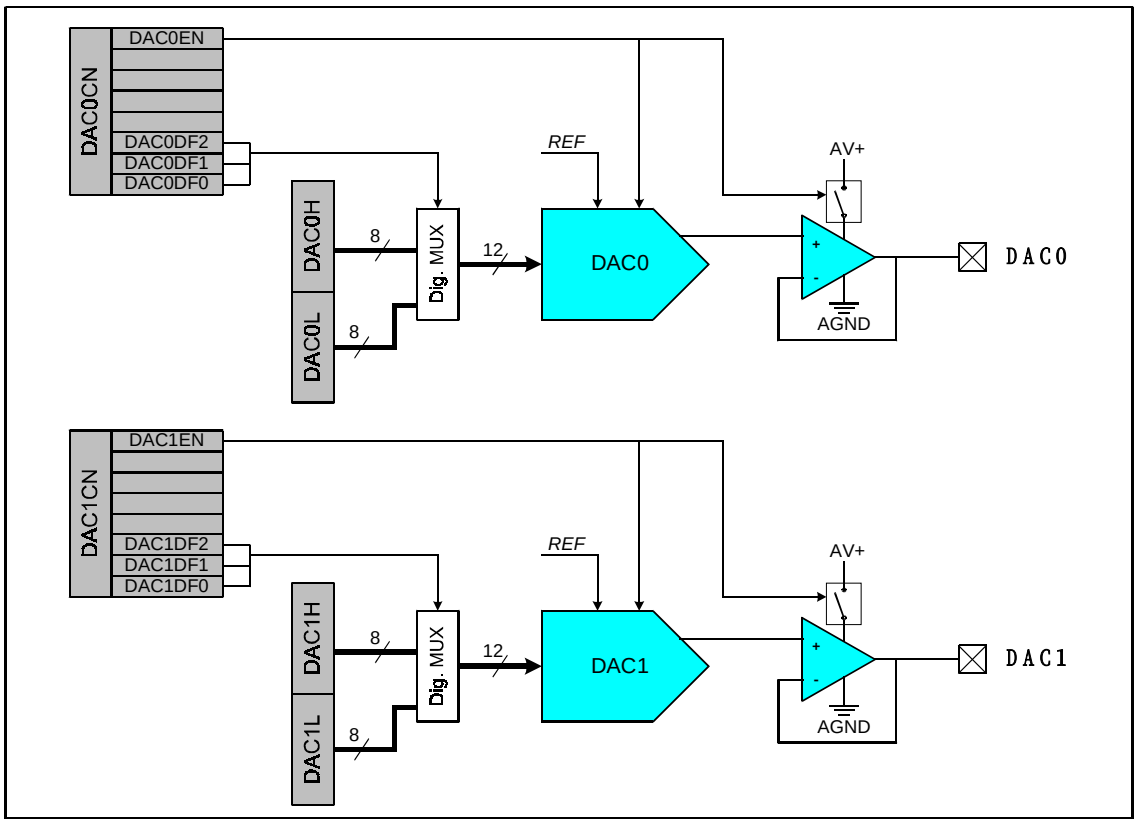


图 7.2　DAC0H: DAC0 高字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xD3

位 7-0：　DAC0 数据字高字节

图 7.3　DAC0L: DAC0 低位字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xD2

位 7-0：　DAC0 数据字低字节

图 7.4　DAC0CN: DAC0 控制寄存器

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

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

位 6-3：　未用。读 = 00b；写 = 忽略

位 2-0:　DAC0DF2-0：DAC0 数据格式位

000：DAC0 数据字的高 4 位在 DAC0H[3:0]，低字节在 DAC0L 中。

DAC0H								DAC0L							
				MSB											LSB

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

DAC0H								DAC0L							
			MSB											LSB	

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

DAC0H								DAC0L							
		MSB											LSB		

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

DAC0H								DAC0L							
	MSB											LSB			

1xx：高有效字节在 DAC0H 中，低 4 位在 DAC0L[7:4]。

DAC0H								DAC0L							
MSB											LSB				

图 7.5　DAC1H: DAC1 高字节寄存器

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

位 7-0：　DAC1 数据字高字节

图 7.6　DAC1L: DAC1 低字节寄存器

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

位 7-0：　DAC1 数据字低字节

图 7.7　DAC1CN: DAC1 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
DAC1EN					DAC1DF2	DAC1DF1	DAC1DF0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xD7

位 7：　DAC1EN：DAC1 允许位
0：DAC1 禁止。DAC1 输出引脚为高阻态，DAC1 处于节电停机方式
1：DAC1 允许。DAC1 引脚正常输出，DAC1 处于工作状态。

位 6-3：　未用。读 = 00b；写 = 忽略

位 2-0:　DAC1DF2-0：DAC1 数据格式位

000：DAC1 数据字的高 4 位在 DAC1H[3:0]，低字节在 DAC1L 中。

DAC1H								DAC1L							
				MSB											LSB

001：DAC1 数据字的高 5 位在 DAC1H[4:0]，低 7 位在 DAC1L[7:1]。

DAC1H								DAC1L							
			MSB											LSB	

010：DAC1 数据字的高 6 位在 DAC1H[5:0]，低 6 位在 DAC1L[7:2]。

DAC1H								DAC1L							
		MSB											LSB		

011：DAC1 数据字的高 7 位在 DAC1H[6:0]，低 5 位在 DAC1L[7:3]。

DAC1H								DAC1L							
	MSB											LSB			

1xx：DAC1 数据字的高字节在 DAC1H 中，低 4 位在 DAC1L[7:4]。

DAC1H								DAC1L							
MSB											LSB				

表 7.1. DAC 电气特性。

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

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

8. 比较器

该 MCU 系列有两个片内比较器，如图 8.1 所示。每个比较器都有输入引脚。每个比较器的输出都可以经 I/O 交叉开关连到外部引脚（见 15.1 节）。当被分配了封装引脚时，每个比较器输出都可以被编程为工作在漏极开路或推挽方式（见 15.3 节）。

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

使用比较器 0 控制寄存器 CPT0CN（见图 8.3）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 8.2 所示，可以设置 10、4 或 2mV 的负向回差电压值，或者禁止负向回差电压。类似地，通过设置 CP0HYP 位决定正向回差电压值。

在比较器的上升沿和下将沿都可以产生中断。（有关中断允许和优先级控制的内容见 10.4 节）。比较器的下降沿中断置‘1’CP0FIF 标志，比较器的上升沿中断置‘1’CP0RIF 标志。这些位一旦被置‘1’，将一直保持‘1’状态直到被 CPU 清除。可以在任意时间通过读取 CP0OUT 位得到比较器 0 的输出状态。注意，在上电后直到比较器能稳定工作之前应忽略比较器输出和中断。通过置‘1’CP0EN 位允许比较器 0 工作，通过清除该位禁止比较器 0。注意，在上电或置‘1’CP0EN 位后大约要经过 20 微秒的建立时间比较器输出才能稳定。比较器 0 还可被编程为复位源，详见第 13 章。

比较器 1 的工作过程与比较器 0 完全相同，只是比较器 1 受 CPT1CN 寄存器（图 8.4）控制。比较器 1 不能被编程为复位源。在 F002、F007、F012、F017 中，比较器 1 没有输入引脚。表 8.1 给出了比较器的详细电特性。

图 8.1 比较器功能框图

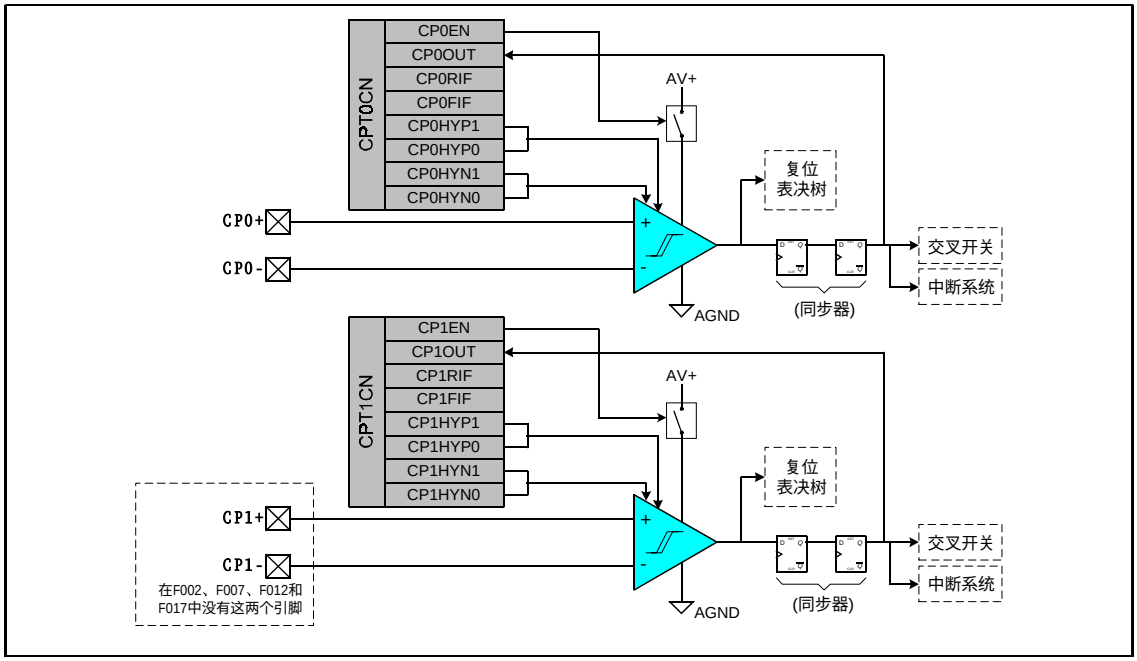


图 8.2 比较器回差电压曲线

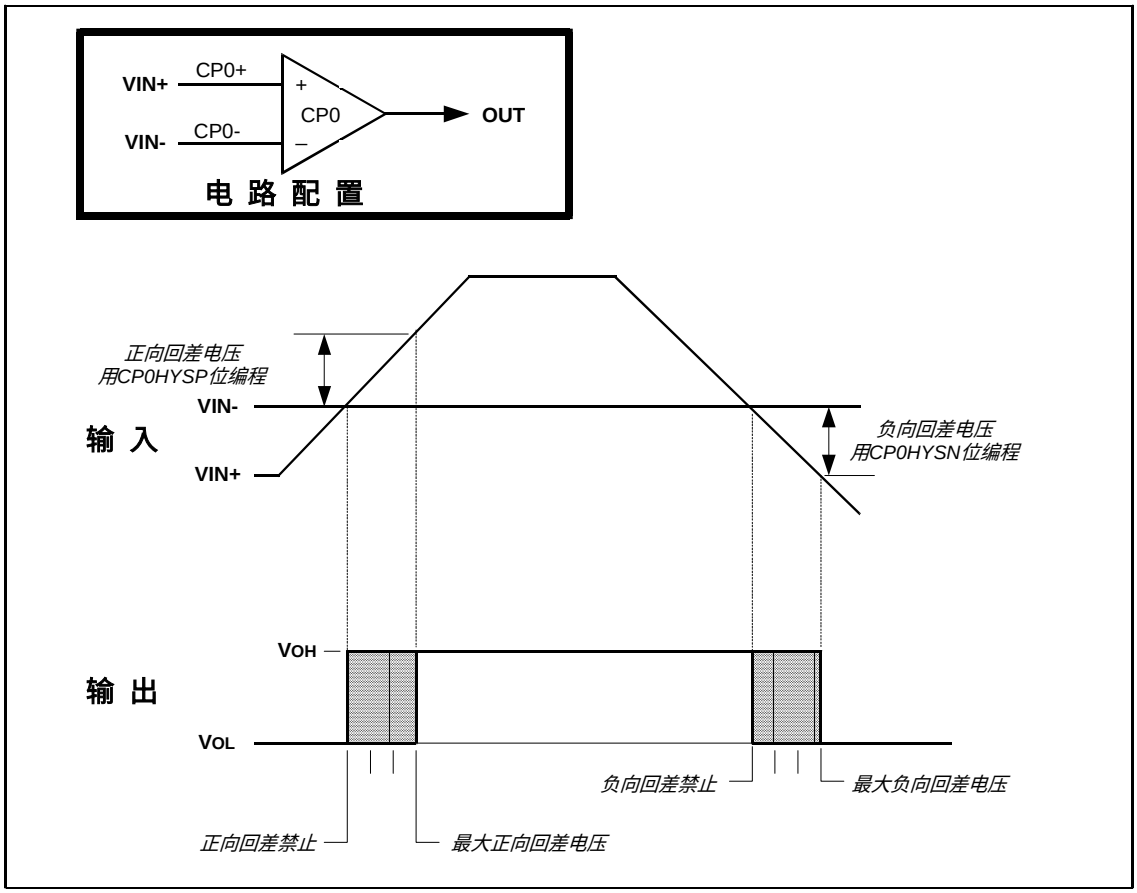


图 8.3　CPT0CN: 比较器 0 控制寄存器

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：0x9E

位 7：　CP0EN：比较器 0 允许位
0：比较器 0 禁止。
1：比较器 0 允许。

位 6：　CP0OUT：比较器 0 输出状态位
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，没有发生比较器 0 上升沿中断
1：自该标志位被清除后，发生了比较器 0 上升沿中断

位 4：　CP0FIF：比较器 0 下降沿中断标志
0：自该标志位被清除后，没有发生比较器 0 下降沿中断
1：自该标志位被清除后，发生了比较器 0 下降沿中断

位 3-2：　CP0HYP1-0：比较器 0 正向回差电压控制位
00：禁止正向回差电压
01：正向回差电压=2mV
10：正向回差电压=4mV
11：正向回差电压=10mV

位 1-0：　CP0HYN1-0：比较器 0 负向回差电压控制位
00：禁止负向回差电压
01：负向回差电压=2mV
10：负向回差电压=4mV
11：负向回差电压=10mV

图 8.4　CPT1CN: 比较器 1 控制寄存器

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP1	CP1HYP0	CP1HYN1	CP1HYN0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0x9F

位 7：　CP1EN：比较器 1 允许位
0：比较器 1 禁止。
1：比较器 1 允许。

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

位 5：　CP1RIF：比较器 1 上升沿中断标志
0：自该标志位被清除后，没有发生比较器 1 上升沿中断
1：自该标志位被清除后，发生了比较器 1 上升沿中断

位 4：　CP1FIF：比较器 1 下降沿中断标志
0：自该标志位被清除后，没有发生比较器 1 下降沿中断
1：自该标志位被清除后，发生了比较器 1 下降沿中断

位 3-2：　CP1HYP1-0：比较器 1 正向回差电压控制位
00：禁止正向回差电压
01：正向回差电压=2mV
10：正向回差电压=4mV
11：正向回差电压=10mV

位 1-0：　CP1HYN1-0：比较器 1 负向回差电压控制位
00：禁止负向回差电压
01：负向回差电压=2mV
10：负向回差电压=4mV
11：负向回差电压=10mV

表 8.1. 比较器电气特性

VDD=3.0V, AV+=3.0V, -40℃到+85℃（除非另有说明）

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

注 1：CPnHYP1-0 = CPnHYN1-0 = 00。

9. 电压基准

电压基准电路包含一个 1.2V、15ppm/℃（典型值）的带隙电压基准发生器和一个两倍增益的输出缓冲放大器。VREF 上的基准电压可以连到应用系统中的外部器件，条件是从 VREF 引脚看的对地最大负载小于 200μA（见 9.1 节）。

如果需要不同的基准电压，可以在 VREF 引脚接一个外部基准电压并用软件禁止内部带隙基准和缓冲放大器。外部基准电压必须小于 AV+ - 0.3V。基准电压控制寄存器 REF0CN（见图 9.2）提供了允许和禁止带隙基准和缓冲放大器的手段。REF0CN 中的 BIASE 位控制 ADC 和 DAC 的偏置电路工作，而 REFBE 位控制带隙基准和驱动 VREF 引脚的缓冲放大器。当被禁止时，带隙基准和缓冲放大器消耗的电流小于 1μA（典型值），缓冲放大器的输出进入高阻状态。如果要使用内部带隙基准作为电压基准发生器，则 BIASE 和 REFBE 位必须被置‘1’。如果使用外部基准，REFBE 位必须被清‘0’而 BIASE 位必须被置‘1’。如果既不使用 ADC 也不使用 DAC，则这两位都应被清‘0’以节省功耗。表 9.1 给出了电压基准的电气特性。

温度传感器接在 A/D 转换器输入多路开关的最后一个输入端（详见图 5.1 和图 5.5）。REF0CN 中的 TEMPE 位允许和禁止温度传感器。当被禁止时，温度传感器为缺省的高阻状态，此时对温度传感器的任何 A/D 测量结果都是无意义的。

图 9.1 电压基准功能框图

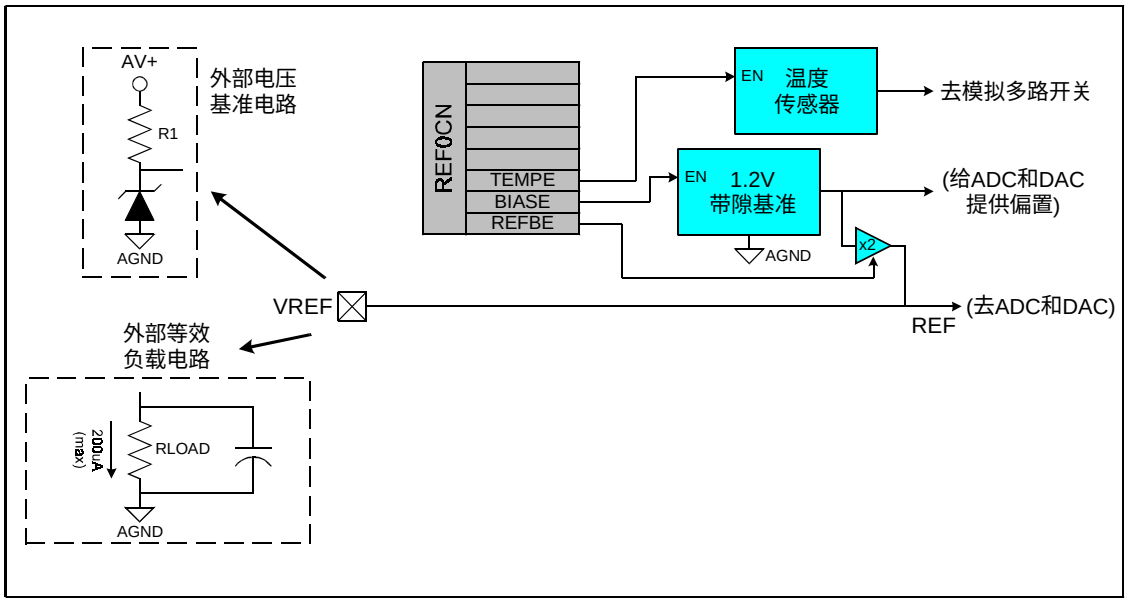


图 9.2 REF0CN: 基准电压控制寄存器

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

位 7-3： 未用。读 = 00000b，写 = 忽略
位 2： TEMPE：温度传感器允许位
0：内部传感器关闭
1：内部传感器工作
位 1： BIASE：ADC 和 DAC 偏压允许
0：内部偏压关闭
1：内部偏压允许（使用 ADC 和 DAC 时需要）
位 0： REFBE：内部电压基准缓冲器允许位
0：内部电压基准缓冲器关闭。系统基准电压取自 VREF 引脚的外部源。
1：电压基准缓冲器工作。系统基准电压由内部电压基准提供。

表 9.1 电压基准的电气特性

VDD=3.0V, AV+=3.0V, -40℃到+85℃（除非另有说明）

参 数	条 件	最小值	典型值	最大值	单 位
内部基准（REFBE＝1）					
输出电压	环境温度 25℃	2.34	2.43	2.48	V
VREF 短路电流				30	mA
VREF 温度系数			15		ppm/℃
负载调整	负载 = (0 至 200μA)到 AGND(注 1)		0.5		ppm/μA
VREF 开启时间 1	4.7μF 钽电容，0.1μF 陶瓷旁路电容		2		ms
VREF 开启时间 2	0.1μF 陶瓷旁路电容		20		μs
VREF 开启时间 3	没有旁路电容		10		μs
外部基准（REFBE＝0）					
输入电压范围		1.00		(AV+) -0.3V	V
输入电流			0	1	μA

注 1：电压基准只能提供电流。当驱动外部负载时，建议接一个负载电阻到 AGND。

10. CIP-51 CPU

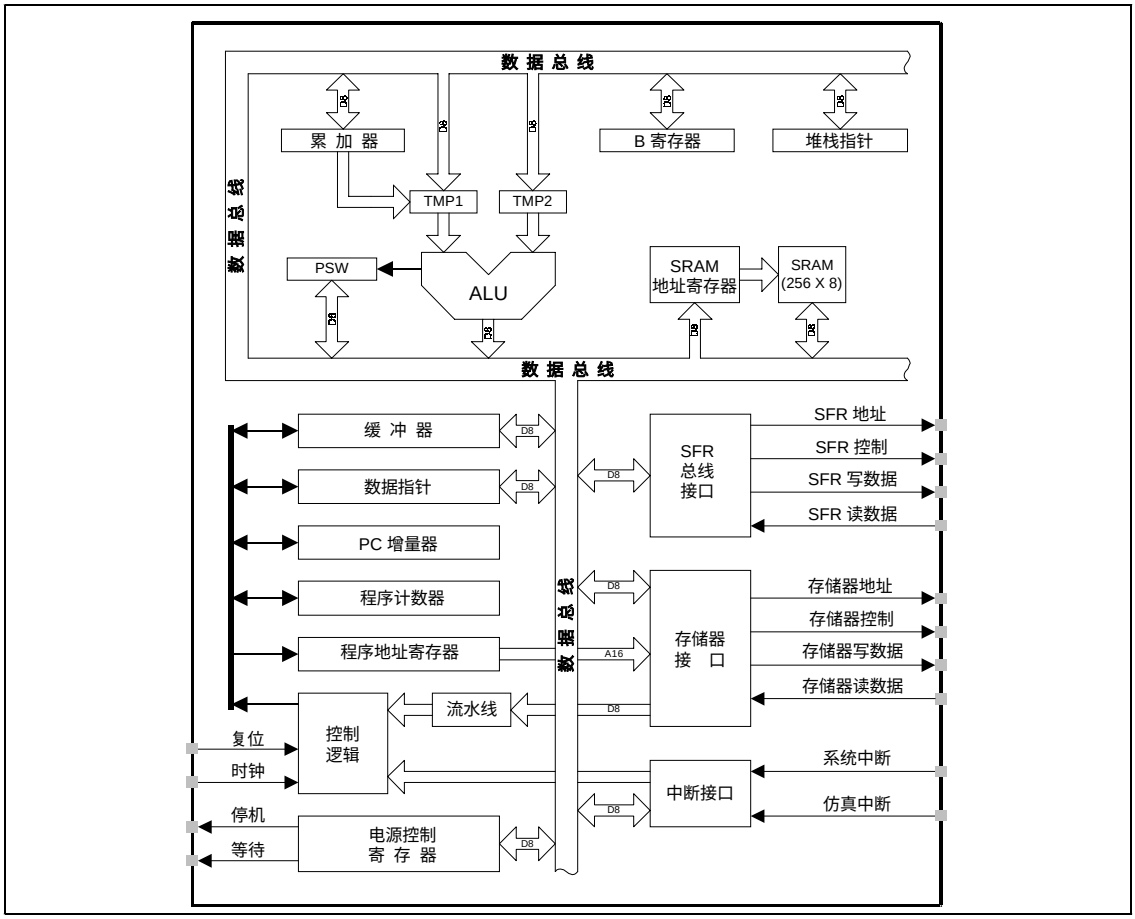
MCU 系统的 CPU 是 CIP-51。CIP-51 与 MCS-51™ 指令集完全兼容，可以使用标准 803x/805x 的汇编器和编译器进行软件开发。该系列 MCU 具有标准 8051 的所有外设部件，包括 4 个 16 位的计数器/定时器（详见第 19 章）、一个全双工 UART（详见第 18 章）、256 字节内部 RAM、128 字节特殊功能寄存器（SFR）地址空间及 4 字节宽的 I/O 端口（详见第 14 章）。CIP-51 还包含片内调试硬件（详见第 21 章）和与 MCU 直接接口的模拟和数字子系统，在一个集成电路内提供了完全的数据采集或控制系统解决方案。

特点

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 10.1 的原理框图）。CIP-51 具有下列特点：

- 与 MCS-51 指令集完全兼容
- 在时钟频率为 25MHz 时的峰值执行速度为 25MIPS
- 0 到 25MHz 的时钟频率
- 4 字节宽的 I/O 端口
- 扩展的中断处理系统
- 复位输入
- 电源管理方式
- 片内调试电路
- 程序和数据存储器安全

图 10.1 CIP-51 原理框图



性能

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

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

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

编程和调试支持

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

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

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

10.1 指令集

CIP-51 系统控制器的指令集与标准 MCS-51™ 指令集完全兼容。可以使用标准 8051 的开发工具开发 CIP-51 的软件。所有的 CIP-51 指令在二进制码和功能上与 MCS-51 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。

10.1.1 指令和 CPU 时序

在很多的 8051 实现中，机器周期和时钟周期之间是有差别的，机器周期的长度在 2 到 12 个时钟周期之间。但是 CIP-51 实现只基于时钟周期，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程序指令的字节数一致。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 10.1 给出了 CIP-51 指令一览表，包括助记符、字节数和时钟周期数。

10.1.2 MOVX 指令和程序存储器

MOVX 指令通常用于访问外部数据存储器。在 CIP-51 中，通过使用 PSCTL 寄存器中的控制位（见图 11.1），MOVX 指令可以访问用可重编程 FLASH 实现的片内程序存储器空间。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。对具有位于外部数据存储器空间的 RAM 的产品（C8051F005/06/07/15/16/17），MOVX 仍可用于读/写这个存储器，但 PSCTL 寄存器要配置为访问外部数据存储器空间。详见第 11 章（FLASH 存储器）。

表 10.1 CIP-51 指令集

助记符	功能说明	字节数	时钟周期数
算术操作类指令			
ADD A,Rn	寄存器加到累加器	1	1
ADD A,direct	直接寻址字节加到累加器	2	2
ADD A,@Ri	间址 RAM 内容加到累加器	1	2
ADD A,#data	立即数加到累加器	2	2
ADDC A,Rn	寄存器加到累加器(带进位)	1	1
ADDC A,direct	直接寻址字节加到累加器(带进位)	2	2
ADDC A,@Ri	间址 RAM 加到累加器(带进位)	1	2
ADDC A,#data	立即数加到累加器(带进位)	2	2
SUBB A,Rn	累加器减去寄存器(带借位)	1	1
SUBB A,direct	累加器减去间接寻址 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
逻辑操作类指令			
ANL A,Rn	寄存器“与”到累加器	1	1
ANL A,direct	直接寻址字节“与”到累加器	2	2
ANL A,@Ri	间址 RAM“与”到累加器	1	2
ANL A,#data	立即数“与”到累加器	2	2
ANL direct,A	累加器“与”到直接寻址字节	2	2
ANL direct,#data	立即数“与”到直接寻址字节	3	3
ORL A,Rn	寄存器“或”到累加器	1	1
ORL A,direct	直接寻址字节“或”到累加器	2	2
ORL A,@Ri	间址 RAM“或”到累加器	1	2
ORL A,#data	立即数“或”到累加器	2	2
ORL direct,A	累加器“或”到直接寻址字节	2	2
ORL direct,#data	立即数“或”到直接寻址字节	3	3
XRL A,Rn	寄存器“异或”到累加器	1	1
XRL A,direct	直接寻址数“异或”到累加器	2	2
XRL A,@Ri	间址 RAM“异或”到累加器	1	2
XRL A,#data	立即数“异或”到累加器	2	2
XRL direct,A	累加器“异或”到直接寻址字节	2	2
XRL direct,#data	立即数“异或”到直接寻址字节	3	3
CLR A	累加器清零	1	1
CPL A	累加器求反	1	1
RL A	循环循环左移	1	1
RLC A	经过进位位的累加器循环左移	1	1

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

助记符	功能说明	字节数	时钟周期数
控制转移类指令			
ACALL addr11	绝对调用子程序	2	3
LCALL addr16	长调用子程序	3	4
RET	从子程序返回	1	5
RETI	从中断返回	1	5
AJMP addr11	绝对转移	2	3
LJMP addr16	长转移	3	4
SJMP rel	短转移（相对偏移）	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。

10.2 存储器组织

CIP-51 系统控制器的存储器组织与标准 8051 的存储器组织类似。有两个独立的存储器空间：程序存储器和数据存储器。程序和数据存储器共享同一个地址空间，但用不同的指令类型访问。CIP-51 内部有 256 字节的内部数据存储器和 64K 字节的内部程序存储器地址空间。CIP-51 的存储器组织如图 10.2 所示。

10.2.1 程序存储器

CIP-51 有 64K 字节的程序存储器空间。MCU 在这个程序存储器空间中实现了 32896 字节可在系统编程的 FLASH 存储器，组织在一个连续的存储块内，从地址 0x0000 到 0x807F。注意：该存储器中有 512 字节（0x7E00 – 0x7FFF）保留给工厂使用，不能用于存储用户程序。

程序存储器通常被认为是只读的，但是 CIP-51 可以通过设置**程序存储写允许**位（PSCTL.0）用 MOVX 指令对程序存储器写入。这一特性为 CIP-51 提供了更新程序代码和将程序存储器空间用于非易失性数据存储的机制。更进一步的详细信息参见第 11 章（FLASH 存储器）。

10.2.2 数据存储器

CIP-51 的数据存储器空间中有 256 字节的内部 RAM，从地址 0x00 到 0xFF。数据存储器中的低 128 字节用于通用寄存器和临时存储器。可以用直接或间接寻址方式访问数据存储器的低 128 字节。从 0x00 到 0x1F 为 4 个通用寄存器区，每个区有 8 个寄存器。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 10.2 给出了 CIP-51 数据存储器组织的示意图。

C8051F005/06/07/15/16/17 还有 2048 字节的 RAM 位于 CIP-51 的外部数据存储器空间，需用 MOVX 指令访问。

10.2.3 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 10.6 中关于 PSW 的说明）。这允许在进入子程序或中断服务程序时进行快速现场切换。间接寻址方式使用 R0 和 R1 作为间址寄存器。

10.2.4 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

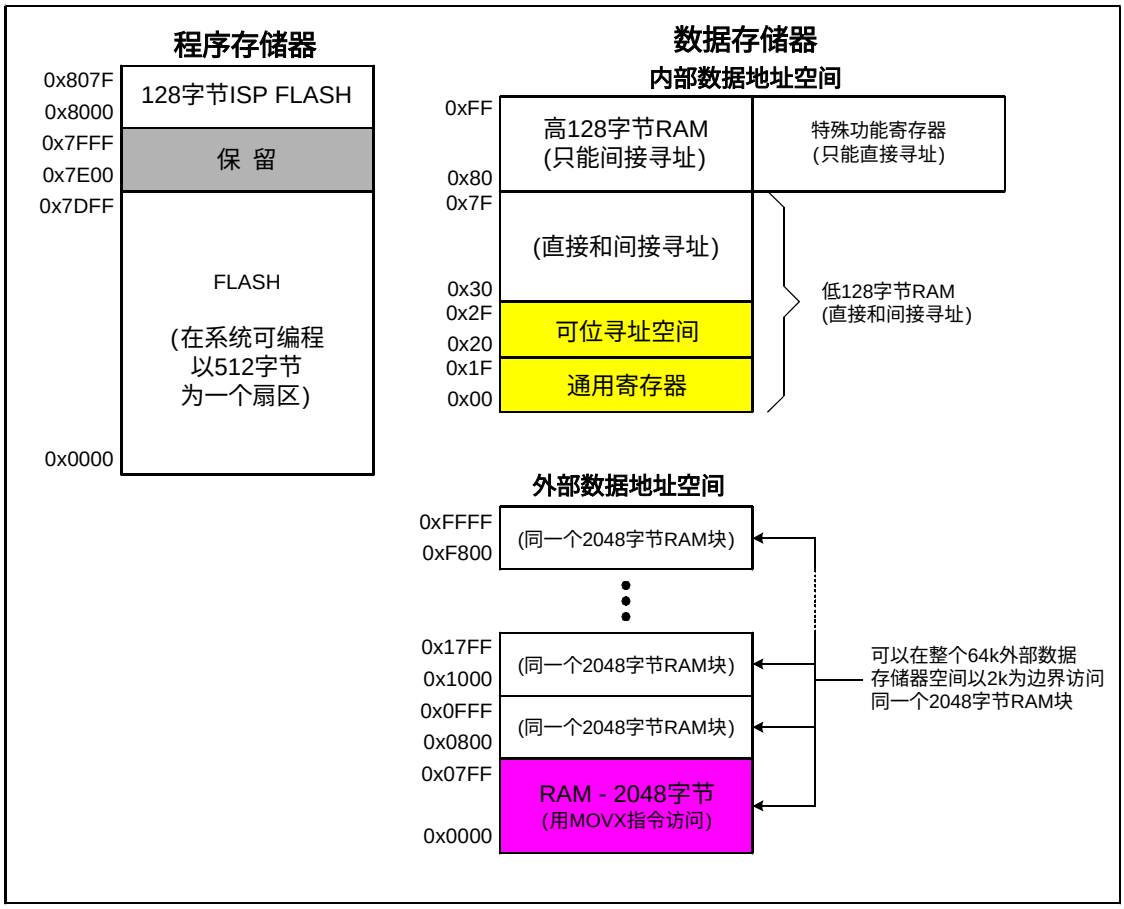
数据字节的位 7 具有位地址 0x7F。由所用指令的类型来区分是位寻址还是字节寻址。

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。

图 10.2 存储器结构图



10.2.5 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。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。

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

10.3 特殊功能寄存器

从 0x80 到 0xFF 的直接寻址存储器空间为特殊功能寄存器（SFR）。SFR 提供对 CIP-51 的资源和外设的控制及 CIP-51 与这些资源和外设之间的数据交换。CIP-51 具有标准 8051 中的全部 SFR，还增加了一些用于配置和访问专有子系统的 SFR。这就允许在保证与 MCS-51™ 指令集兼容的前提下增加新的功能。表 10.3 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）既可以按字节寻址也可以按位寻址。所有其它 SFR 只能按字节寻址。SFR 空间中未使用的地址保留为将来使用。访问这些地址会产生不确定的结果，应予避免。有关每个寄存器的详细说明请参见本数据表的相关部分（表 10.3 中已标明）。

表 10.2 特殊功能寄存器存储器映象

F8	SPI0CN	PCA0H	PCA0CPH0	PCA0CPH1	PCA0CPH2	PCA0CPH3	PCA0CPH4	WDTCN
F0	B						EIP1	EIP2
E8	ADC0CN	PCA0L	PCA0CPL0	PCA0CPL1	PCA0CPL2	PCA0CPL3	PCA0CPL4	RSTSRC
E0	ACC	XBR0	XBR1	XBR2			EIE1	EIE2
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	
D0	PSW	REF0CN	DAC0L	DAC0H	DAC0CN	DAC1L	DAC1H	DAC1CN
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2		SMB0CR
C0	SMB0CN	SMB0STA	SMB0DAT	SMB0ADR	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH
B8	IP		AMX0CF	AMX0SL	ADC0CF		ADC0L	ADC0H
B0	P3	OSCXCN	OSCICN				FLSCL	FLACL***
A8	IE					PRT1IF		EMI0CN***
A0	P2				PRT0CF	PRT1CF	PRT2CF	PRT3CF
98	SCON	SBUF	SPI0CFG	SPI0DAT		SPI0CKR	CPT0CN	CPT1CN
90	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H		
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH				PCON
	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

↑ 可位寻址

表 10.3 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。
*寄存器只在 C8051F000/1/2/5/6/7 中存在。
**寄存器只在 C8051F010/1/2/5/6/7 中存在。
***寄存器只在 C8051F005/06/07/15/16/17 中存在

地址	寄存器	说明	页码
0xE0	ACC	累加器	75
0xBC	ADC0CF	ADC 配置寄存器	33*,42**
0xE8	ADC0CN	ADC 控制寄存器	34*,45**
0xC5	ADC0GTH	ADC 下限数据字（高字节）	36*,47**
0xC4	ADC0GTL	ADC 下限数据字（低字节）	36*,47**
0xBF	ADC0H	ADC 数据字（高字节）	35*,46**
0xBE	ADC0L	ADC 数据字（低字节）	35*,46**
0xC7	ADC0LTH	ADC 上限数据字（高字节）	36*,47**
0xC6	ADC0LTL	ADC 上限数据字（低字节）	36*,47**
0xBA	AMX0CF	ADC MUX 配置寄存器	31*,42**
0xBB	AMX0SL	ADC MUX 通道选择寄存器	32*,43**
0xF0	B	B 寄存器	75
0x8E	CKCON	时钟控制寄存器	142
0x9E	CPT0CN	比较器 0 控制寄存器	56
0x9F	CPT1CN	比较器 1 控制寄存器	58
0xD4	DAC0CN	DAC0 控制寄存器	52
0xD3	DAC0H	DAC0 数据字（高字节）	52
0xD2	DAC0L	DAC0 数据字（低字节）	52
0xD7	DAC1CN	DAC1 控制寄存器	53
0xD6	DAC1H	DAC1 数据字（高字节）	53
0xD5	DAC1L	DAC1 数据字（低字节）	53
0x83	DPH	数据指针（高字节）	73
0x82	DPL	数据指针（低字节）	73
0xE6	EIE1	扩展中断允许 1	80
0xE7	EIE2	扩展中断允许 2	81
0xF6	EIP1	扩展中断优先级 1	82
0xF7	EIP2	扩展中断优先级 2	83
0xAF	EMI0CN	外部存储器接口控制寄存器	91***
0xB7	FLACL	Flash 访问限制	89***
0xB6	FLSCL	Flash 存储器时序预分频器	90
0xA8	IE	中断允许寄存器	78
0xB8	IP	中断优先级控制寄存器	79
0xB2	OSCICN	内部振荡器控制寄存器	99
0xB1	OSCXCN	外部振荡器控制寄存器	100

地址	寄存器	说明	页码
0x80	P0	端口 0 锁存器	108
0x90	P1	端口 1 锁存器	109
0xA0	P2	端口 2 锁存器	110
0xB0	P3	端口 3 锁存器	111
0xD8	PCA0CN	可编程计数器阵列 0 控制	158
0xFA	PCA0CPH0	PCA 捕捉模块 0 数据字（高字节）	161
0xFB	PCA0CPH1	PCA 捕捉模块 1 数据字（高字节）	161
0xFC	PCA0CPH2	PCA 捕捉模块 2 数据字（高字节）	161
0xFD	PCA0CPH3	PCA 捕捉模块 3 数据字（高字节）	161
0xFE	PCA0CPH4	PCA 捕捉模块 4 数据字（高字节）	161
0xEA	PCA0CPL0	PCA 捕捉模块 0 数据字（低字节）	161
0xEB	PCA0CPL1	PCA 捕捉模块 1 数据字（低字节）	161
0xEC	PCA0CPL2	PCA 捕捉模块 2 数据字（低字节）	161
0xED	PCA0CPL3	PCA 捕捉模块 3 数据字（低字节）	161
0xEE	PCA0CPL4	PCA 捕捉模块 4 数据字（低字节）	161
0xDA	PCA0CPM0	可编程计数器阵列 0 捕捉／比较 0	160
0xDB	PCA0CPM1	可编程计数器阵列 0 捕捉／比较 1	160
0xDC	PCA0CPM2	可编程计数器阵列 0 捕捉／比较 2	160
0xDD	PCA0CPM3	可编程计数器阵列 0 捕捉／比较 3	160
0xDE	PCA0CPM4	可编程计数器阵列 0 捕捉／比较 4	160
0xF9	PCA0H	PCA 计数器/定时器数据字（高字节）	161
0xE9	PCA0L	PCA 计数器/定时器数据字（低字节）	161
0xD9	PCA0MD	可编程计数器阵列 0 方式寄存器	159
0x87	PCON	电源控制寄存器	85
0xA4	PRT0CF	端口 0 配置寄存器	108
0xA5	PRT1CF	端口 1 配置寄存器	109
0xAD	PRT1IF	端口 1 中断标志寄存器	109
0xA6	PRT2CF	端口 2 配置寄存器	110
0xA7	PRT3CF	端口 3 配置寄存器	111
0x8F	PSCTL	程序存储读写控制寄存器	87
0xD0	PSW	程序状态字	74
0xCB	RCAP2H	计数器/定时器 2 捕捉（高字节）	149
0xCA	RCAP2L	计数器/定时器 2 捕捉（低字节）	149
0xD1	REF0CN	电压基准控制寄存器	61
0xEF	RSTSRC	复位源寄存器	96
0x99	SBUF	串口数据缓冲器（UART）	134
0x98	SCON	串口控制寄存器（UART）	135
0xC3	SMB0ADR	SMBus 0 地址寄存器	119
0xC0	SMB0CN	SMBus 0 控制寄存器	117
0xCF	SMB0CR	SMBus 0 时钟频率寄存器	118
0xC2	SMB0DAT	SMBus 0 数据寄存器	119
0xC1	SMB0STA	SMBus 0 状态寄存器	120

地址	寄存器	说明	页码
0x81	SP	堆栈指针	73
0x9A	SPI0CFG	串行外设接口配置	126
0x9D	SPI0CKR	SPI 时钟频率	128
0xF8	SPI0CN	SPI 总线控制	127
0x9B	SPI0DAT	SPI 端口 1 数据	128
0xC8	T2CON	计数器/定时器 2 控制	148
0x88	TCON	计数器/定时器控制	140
0x8C	TH0	计数器/定时器 0 数据字(高字节)	143
0x8D	TH1	计数器/定时器 1 数据字(高字节)	143
0xCD	TH2	计数器/定时器 2 数据字(高字节)	149
0x8A	TL0	计数器/定时器 0 数据字(低字节)	143
0x8B	TL1	计数器/定时器 1 数据字(低字节)	143
0xCC	TL2	计数器/定时器 2 数据字(低字节)	149
0x89	TMOD	计数器/定时器方式寄存器	141
0x91	TMR3CN	定时器 3 控制寄存器	150
0x95	TMR3H	定时器 3 高字节	151
0x94	TMR3L	定时器 3 低字节	151
0x93	TMR3RLH	定时器 3 重载值高字节	151
0x92	TMR3RLL	定时器 3 重载值低字节	151
0xFF	WDTCN	看门狗定时器控制	95
0xE1	XBR0	端口 I/O 交叉开关配置 1	104
0xE2	XBR1	端口 I/O 交叉开关配置 2	106
0xE3	XBR2	端口 I/O 交叉开关配置 3	107
0x84-86, 0x96-97, 0x9C, 0xA1-A3, 0xA9-AC, 0xAE, 0xB3-B5, 0xB9, 0xBD, 0xC9, 0xCE, 0xDF, 0xE4-E5, 0xF1-F5	保留		

*寄存器只在 C8051F000/1/2/5/6/7 中存在。
**寄存器只在 C8051F010/1/2/5/6/7 中存在。
***寄存器只在 C8051F005/06/07/15/16/17 中存在

10.3.1 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，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 的详细说明及它们对应的系统功能见本数据表的相关章节。

图 10.3　SP：堆栈指针

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								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 寄存器复位默认值为 0x07。

图 10.4　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 RAM 和 FLASH 存储器。

图 10.5　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 RAM 和 FLASH 存储器。

图 10.6　PSW：程序状态字

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

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

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

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

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

图 10.7　ACC：累加器

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

位 7-0：　ACC：累加器
该寄存器为算术操作用的累加器。

图 10.8　B：B 寄存器

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

位 7-0：　B：B 寄存器
该寄存器作为某些算术操作的第二个累加器。

10.4 中断系统

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

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

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

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

10.4.1 MCU 中断源和中断向量

MCU 有 12 个中断源分配给片内外设。根据器件 I/O 引脚的配置情况，最多可有 10 个附加的外部中断源。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，系统将产生中断，CPU 将转向与该中断标志对应的 ISR 地址。表 10.4 给出了 MCU 中断源、对应的向量地址、优先级和控制位一览表。关于外设有效中断条件和中断标志位工作状态方面的详细信息，请见与特定外设相关的章节。

10.4.2 外部中断

有两个外部中断源（/INT0 和/INT1）可被配置为低电平触发或下降沿触发输入，由 IT0（TCON.0）和 IT1（TCON.2）的设置决定。IE0（TCON.1）和 IE1（TCON.3）分别为外部中断/INT0 和/INT1 的中断标志。如果一个/INT0 或/INT1 外部中断被配置为边沿触发，CPU 在转向 ISR 时将自动清除相应的中断标志。当被配置为电平触发时，中断标志将跟随外部中断输入引脚的状态。外部中断源必须一直保持输入有效直到中断请求被响应。在 ISR 返回前必须使该中断请求无效，否则将产生另一个中断请求。

其余的 4 个外部中断（外部中断 4-7）为下降沿触发输入。这些中断的中断标志在**端口 1 中断标志寄存器**中，见图 15.10。

表 10.4 中断一览表

中断源	中断向量	优先权	中断标志	允许
复位	0x0000	最高	无	始终使能
外部中断 0 (/INT0)	0x0003	0	IE0 (TCON.1)	EX0 (IE.0)
定时器 0 溢出	0x000B	1	IE0 (TCON.5)	ET0 (IE.1)
外部中断 1 (INT1)	0x0013	2	IE1 (TCON.3)	EX1 (IE.2)
定时器 1 溢出	0x001B	3	TF1 (TCON.7)	ET1 (IE.3)
串口(UART)	0x0023	4	RI (SCON.0) TI (SCON.1)	ES (IE.4)
定时器 2 溢出（或 EXF2）	0x002B	5	TF2 (T2CON.7)	ET2 (IE.5)
串行外设接口	0x0033	6	SPIF (SPI0STA.7)	ESPI0 (EIE1.0)
SMBus 接口	0x003B	7	SI (SMB0CN.3)	ESMB0 (EIE1.1)
ADC0 窗口比较	0x0043	8	ADWINT(ADC0CN.2)	EWADC0(EIE1.2)
可编程计数器阵列 0	0x004B	9	CF (PCA0CN.7) CCFn (PCA0CN.n)	EPCA0 (EIE1.3)
比较器 0 下降沿	0x0053	10	CP0FIF (CPT0CN.4)	ECP0F (EIE1.4)
比较器 0 上升沿	0x005B	11	CP0RIF (CPT0CN.5)	ECP0R (EIE1.5)
比较器 1 下降沿	0x0063	12	CP1FIF (CPT1CN.4)	ECP1F (EIE1.6)
比较器 1 上升沿	0x006B	13	CP1RIF (CPT1CN.5)	ECP1R (EIE1.7)
定时器 3 溢出	0x0073	14	TF3 (TMR3CN.7)	ET3 (EIE2.0)
ADC0 转换结束	0x007B	15	ADCINT (ADC0CN.5)	EADC0 (EIE2.1)
外部中断 4	0x0083	16	IE4 (PRT1IF.4)	EX4 (EIE2.2)
外部中断 5	0x008B	17	IE5 (PRT1IF.5)	EX5 (EIE2.3)
外部中断 6	0x0093	18	IE6 (PRT1IF.6)	EX6 (EIE2.4)
外部中断 7	0x009B	19	IE7 (PRT1IF.7)	EX7 (EIE2.5)
未使用的中断位置	0x00A3	20	无	保留 (EIE2.6)
外部晶体振荡器准备好	0x00AB	21	XTLVLD(OSCXCN.7)	EXVLD (EIE2.7)

10.4.3 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-EIP2）中都有一个配置其优先级的中断优先级设置位，缺省值为低优先级。如果两个中断同时发生，具有高优先级的中断先得到服务。如果这两个中断的优先级相同，则由固定的优先级顺序决定哪一个先得到服务。

10.4.4 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 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。

10.4.5 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 10.9　IE：中断允许寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
EA	IEGF0	ET2	ES	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。单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 的所有中断。
1：允许 TF2 标志位（T2CON.7）的中断请求。

位 4：　ES：串行口（UART）中断允许位。
该位设置串行口（UART）中断屏蔽。
0：禁止所有 UART 中断。
1：允许 RI 标志位（SCON.0）或 TI 标志位（SCON.1）产生的中断。

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

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

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

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

图 10.10　IP：中断优先级寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	PT2	PS	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：　PS：串行口（UART）中断优先级控制。
该位设置串行口（UART）中断的优先级。
0：串行口（UART）为默认优先级。
1：串行口（UART）为高优先级。

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

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

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

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

图 10.11　EIE1：扩展中断允许 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
ECP1R	ECP1F	ECP0R	ECP0F	EPCA0	EWADC0	ESMB0	ESPI0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xE6

位 7：　ECP1R：允许比较器 1（CP1）上升沿中断。
该位设置 CP1 中断屏蔽。
0：禁止 CP1 上升沿中断。
1：允许 CP1RIF 标志位（CPT1CN.5）的中断请求。

位 6：　ECP1F：允许比较器 1（CP1）下降沿中断。
该位设置 CP1 中断屏蔽。
0：禁止 CP1 下降沿中断。
1：允许 CP1FIF 标志位（CPT1CN.4）的中断请求。

位 5：　ECP0R：允许比较器 0（CP0）上升沿中断。
该位设置 CP0 中断屏蔽。
0：禁止 CP0 上升沿中断。
1：允许 CP1RIF 标志位（CPT0CN.5）的中断请求。

位 4：　ECP0F：允许比较器 0（CP0）下降沿中断。
该位设置 CP0 中断屏蔽。
0：禁止 CP0 下降沿中断。
1：允许 CP0FIF 标志位（CPT0CN.4）的中断请求。

位 3：　EPCA0：允许可编程计数器阵列（PCA0）中断。
该位设置 PCA0 中断屏蔽。
0：禁止所有 PCA0 中断。
1：允许 PCA0 的中断请求。

位 2：　EWADC0：允许 ADC0 窗口比较中断
该位设置 ADC0 窗口比较中断屏蔽。
0：禁止 ADC0 窗口比较中断。
1：允许 ADC0 窗口比较中断请求。

位 1：　ESMB0：允许 SMBus0 中断
该位设置 SMBus0 中断屏蔽。
0：禁止 SMBus0 中断。
1：允许 SI 标志位（SMB0CN.3）的中断请求。

位 0：　ESPI0：允许串行外设接口 0 中断
该位设置 SPI0 的中断屏蔽。
0：禁止 SPI0 中断
1：允许 SPIF 标志位（SPI0CN.7）的中断请求。

图 10.12　EIE2：扩展中断允许 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
EXVLD	-	EX7	EX6	EX5	EX4	EADC0	ET3	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xE7

位 7：　EXVLD：允许外部时钟源有效（XTLVLD）中断。
该位设置 XTLVLD 中断屏蔽。
0：禁止 XTLVLD 中断。
1：允许 XTLVLD 标志位（OSCXCN.7）的中断请求。

位 6：　保留。必须写入 0。读出值为 0。

位 5：　EX7：允许外部中断 7。
该位设置外部中断 7 屏蔽。
0：禁止外部中断 7。
1：允许外部中断 7 输入引脚的中断请求。

位 4：　EX6：允许外部中断 6。
该位设置外部中断 6 屏蔽。
0：禁止外部中断 6。
1：允许外部中断 6 输入引脚的中断请求。

位 3：　EX5：允许外部中断 5。
该位设置外部中断 5 屏蔽。
0：禁止外部中断 5。
1：允许外部中断 5 输入引脚的中断请求。

位 2：　EX4：允许外部中断 4。
该位设置外部中断 4 屏蔽。
0：禁止外部中断 4。
1：允许外部中断 4 输入引脚的中断请求。

位 1：　EADC0：允许 ADC0 转换结束中断。
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 ADC0 转换结束的中断请求。

位 0：　EX3：允许定时器 3 中断。
该位设置定时器 3 中断屏蔽。
0：禁止定时器 3 中断。
1：允许 TF3 标志位（TMR3CN.7）的中断请求。

图 10.13　EIP1：扩展中断优先级 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
PCP1R	PCP1F	PCP0R	PCP0F	PPCA0	PWADC0	PSMB0	PSPI0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xF6

位 7：　PCP1R：比较器 1（CP1）上升沿中断优先级控制
该位设置 CP1 中断的优先级。
0：CP1 上升沿中断为低优先级。
1：CP1 上升沿中断为高优先级。

位 6：　PCP1F：比较器 1（CP1）下降沿中断优先级控制
该位设置 CP1 中断的优先级。
0：CP1 下降沿中断为低优先级。
1：CP1 下降沿中断为高优先级。

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

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

位 3：　PPCA0：可编程计数器阵列（PCA0）中断优先级控制
该位设置 PCA0 中断的优先级。
0：PCA0 中断设置为低优先级。
1：PCA0 中断设置为高优先级。

位 2：　PWADC0：ADC0 窗口比较器中断优先级控制
该位设置 ADC0 窗口中断的优先级。
0：ADC0 窗口中断为低优先级。
1：ADC0 窗口中断为高优先级。

位 1：　PSMB0：SMBus0 中断优先级控制
该位设置 SMBus0 中断的优先级。
0：SMBus 中断为低优先级。
1：SMBus 中断为高优先级。

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

图 10.14　EIP2：扩展中断优先级 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
PXVLD	-	PX7	PX6	PX5	PX4	PADC0	PT3	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xF7

位 7：　PXVLD：外部时钟源有效（XTLVLD）中断优先级控制
该位设置 XTLVLD 中断的优先级。
0：XTLVLD 中断为低优先级
1：XTLVLD 中断为高优先级

位 6：　保留。必须写 0。读出为 0。

位 5：　PX7：外部中断 7 优先级控制
该位设置外部中断 7 的优先级。
0：外部中断 7 为低优先级
1：外部中断 7 为高优先级

位 4：　PX6：外部中断 6 优先级控制
该位设置外部中断 6 的优先级。
0：外部中断 6 设置为低优先级
1：外部中断 6 设置为高优先级

位 3：　PX5：外部中断 5 优先级控制
该位设置外部中断 5 的优先级。
0：外部中断 5 设置为低优先级
1：外部中断 5 设置为高优先级

位 2：　PX4：外部中断 4 优先级控制
该位设置外部中断 4 的优先级。
0：外部中断 4 设置为低优先级
1：外部中断 4 设置为高优先级

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

位 0：　PT3：定时器 3 中断优先级控制
该位设置定时器 3 中断的优先级。
0：定时器 3 中断为低优先级
1：定时器 3 中断为高优先级

10.5 电源管理方式

CIP-51 有两种可软件编程的电源管理方式：等待和停机。在等待方式，CPU 停止运行，而外设和时钟处于活动状态。在停机方式，CPU 停止运行，所有的中断和定时器（时钟丢失检测器除外）都处于非活动状态，系统时钟停止。由于在等待方式下时钟仍然运行，所以功耗与进入等待方式之前的系统时钟频率和处于活动方式的外设数目有关。停机方式消耗最少的功率。图 10.15 对用于控制 CIP-51 电源管理方式的电源控制寄存器作出了说明。

虽然 CIP-51 具有等待和停机方式（与任何标准 8051 结构一样），但最好禁止不需要的外设，以使整个 MCU 的功耗最小。每个模拟外设在不用时都可以被禁止而进入低功耗方式。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。关闭振荡器可以消耗更少的功率，但需要复位来重新启动 MCU。

10.5.1 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在等待方式期间都可以保持活动状态。

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

如果被允许，WDT 将产生一个内部看门狗复位，从而结束等待方式。这一功能可以保护系统不会因为对 PCON 寄存器的意外写入而导致永久性停机。如果不需要这种功能，可以在进入等待方式之前禁止看门狗。这将进一步节省功耗，允许系统一直保持在等待状态，等待一个外部激励唤醒系统。有关使用和配置 WDT 的详细信息，请参见第 13.8 节 — 看门狗定时器。

10.5.1 停机方式

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

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

图 10.15　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
SMOD	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：　SMOD：串口波特率加倍允许。
0：串口波特率为 SCON 中的串口方式定义值。
1：串口波特率为 SCON 中的串口方式定义值的两倍。

位 6-2：　GF4-GF0：通用标志位 4-0。
这些位是用于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、串口和模拟外设处于活动状态。）

11. FLASH 存储器

该系列 MCU 内部有 32k ＋ 128 字节的可编程 FLASH 存储器，用于程序代码和非易失性数据存储。可以通过 JTAG 接口或由软件使用 MOVX 指令对 FLASH 存储器进行在系统编程，每次一个字节。一个 FLASH 位一旦被清‘0’，必须经过擦除才能再回到‘1’状态。在进行重新编程之前，一般要将数据字节擦除（置为 0xFF）。写和擦除操作由硬件自动定时，以保证操作正确。不需要进行数据查询来判断写/擦除操作何时结束。FLASH 存储器被设计为能承受至少 20,000 个擦/写周期。表 11.1 给出了 FLASH 存储器的电气特性。

11.1 FLASH 存储器编程

对 FLASH 存储器编程的最简单的方法是使用由 Cygnal 或第三方供应商提供的编程工具，通过 JTAG 接口编程。这是对未初始化器件的唯一的编程方法。有关 FLASH 存储器编程的 JTAG 命令方面的详细信息，见 21.2 节。

可以用软件使用 MOVX 指令对 FLASH 存储器编程，象一般的操作数一样为 MOVX 指令提供待编程的地址和数据字节。在使用 MOVX 指令对 FLASH 存储器写入之前，必须将程序存储写允许位 PSWE（PSCTL.0）设置为逻辑‘1’，以允许 FLASH 写操作，在用软件清除 PSWE 位之前将一直允许写操作。

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。32k 字节的 FLASH 存储器是以 512 字节的扇区为单位组织的。一次擦除操作将擦除整个扇区（将扇区内的所有字节设置为 0xFF）。将程序存储擦除允许位 PSEE（PSCTL.1）和 PSWE（PSCTL.0）设置为逻辑‘1’后，用 MOVX 命令写一个数据字节到扇区内的任何地址将擦除整个 512 字节的扇区。写入的数据字节可以是任何值，因为不是真正写入到 FLASH。在用软件清除 PSEE 位之前将一直允许擦除操作。下面的步骤说明了用软件对 FLASH 编程的过程：

1. 用 FLSCL 寄存器中的 FLASCL 位允许 FLASH 存储器的写/擦除操作。
2. 置位 PSEE（PSCTL.1）允许 FLASH 扇区擦除。
3. 置位 PSWE（PSCTL.0）允许 FLASH 写。
4. 用 MOVX 指令向待擦除扇区内的任何一个地址写入一个数据字节。
5. 清除 PSEE 以禁止 FLASH 扇区擦除。
6. 用 MOVX 指令向刚擦除的扇区中所希望的地址写入数据字节。重复该步直到结束。（写入的字节数可以是一个字节到整个扇区之间的任何值）。
7. 清除 PSWE 以禁止 FLASH 写。

写/擦除时序由硬件根据 FLASH 存储器定时预分频寄存器（FLSCL）中的预分频值自动控制。4 位预分频值 FLASCL 决定用于写/擦除操作的时间。对于一个给定的系统时钟，所要求的 FLASCL 值示于图 11.4，同时还给出了 FLASCL 值的计算公式。当 FLASCL 值被设置为 1111b 时，写/擦除操作被禁止。注意，在 FLASH 正在被编程或擦除期间，8051 中的程序停止执行。

表 11.1 FLASH 存储器电特性

VDD=2.7 - 3.6V, -40℃到+85℃（除非另有说明）

参 数	条 件	最小值	典型值	最大值	单 位
擦写寿命		20k	100k		擦/写
擦除时间		10			ms
写 周 期		40			μs

11.2 非易失性数据存储

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

MCU 的 FLASH 存储器中有一个附加的 128 字节的扇区，位于 0x8000 – 0x807F。该扇区可用于存储程序代码或数据。然而它较小的扇区容量使其尤其适于作为通用的非易失性临时存储器。尽管 FLASH 存储器可以每次写一个字节，但必须首先擦除整个扇区。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。128 字节的扇区规模使数据更新更加容易，可以使用内部数据 RAM 作为临时存储区而不浪费程序存储器空间。（一个正常的 512 字节的扇区无法保存到 256 字节的内部数据存储器中。）

11.3 安全选项

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

保存在地址 0x7DFE 和 0x7DFF 中的安全锁定字节集可以保护 FLASH 存储器，使得不能通过 JTAG 接口读取或修改其内容。安全锁定字节中的每一位保护一个 4k 字节的存储器块。将读锁定字节中的一位清‘0’可防止通过 JTAG 接口读对应的 FLASH 存储器块。将写/擦除锁定字节中的一位清‘0’可防止通过 JTAG 接口写/擦除对应的存储器块。读锁定字节位于 0x7DFF。写/擦除锁定字节位于 0x7DFE。图 11.2 给出了安全字节的地址和位定义。包含锁定字节的 512 字节扇区可以用软件写入，但不能用软件擦除。

图 11.1 PSCTL：程序存储读写控制

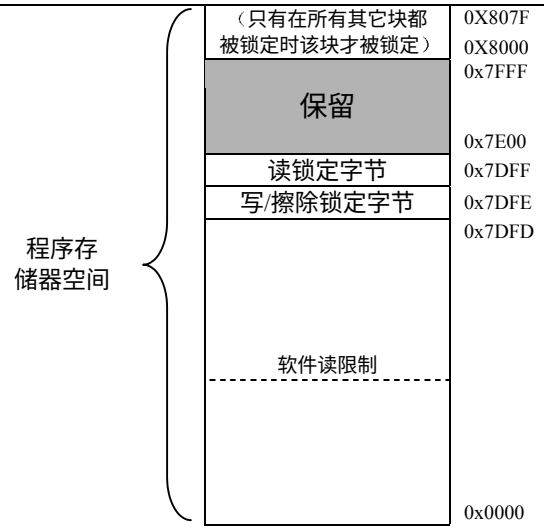
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	-	-	PSEE	PSWE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0x8F

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

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

图 11.2 FLASH 程序存储器安全字节



读和写/擦除安全位。
位7为MSB。

位	存储器块
7	0x7000 – 0x7FFD
6	0x6000 – 0x6FFF
5	0x5000 – 0x5FFF
4	0x4000 – 0x4FFF
3	0x3000 – 0x3FFF
2	0x2000 – 0x2FFF
1	0x1000 – 0x1FFF
0	0x0000 – 0x0FFF

FLASH读锁定字节

位7-0： 每一位锁定一个对应的存储器块。（位7为MSB）
0：锁定对应的存储器块，使不能通过JTAG接口进行读操作。
1：解锁对应的存储器块，可以通过JTAG接口进行读操作

FLASH写/擦除锁定字节

位7-0： 每一位锁定一个对应的存储器块。
0：锁定对应的存储器块，使不能通过JTAG接口进行写/擦除操作。
1：解锁对应的存储器块，可以通过JTAG接口进行写/擦除操作

FLASH访问限制寄存器（FLACL）

该寄存器的内容作为 16 位软件读限制地址的高字节。16 位读限制地址值按 0xNN00 计算，其中 NN 用该寄存器的内容替换。在该地址之上运行的软件不能用 MOVX 或 MOVC 指令读、写或擦除该地址以下的存储单元。读该极限值以下的地址将得到数值 0x00。

不管安全字节所在存储块的安全设置如何，锁定位总是可读的并可以被清‘0’。这就允许在锁定了安全字节所在的存储块以后还可以追加要保护的存储块。但是，一旦设置了锁定位，解锁的唯一办法是用 JTAG 擦除操作擦除整个程序存储器空间（即不能由用户固件解锁）。**注：使用任何一个安全字节地址执行 JTAG 擦除操作将自动启动对整个程序存储器空间的擦除（保留区除外）。如果擦除操作寻址的是 0x7C00-0x7DFF 页内的一个非安全字节，则只有该页（包括安全字节）被擦除。**

FLASH 访问限制这一安全功能（见图 11.3）保护产权程序代码和数据不被运行在 C8051F005/06/07/15/16/17 MCU 上的软件读取。该功能为那些想在产品发行前在 MCU 中加入增值产权固件的 OEM 生产商提供了支持。这一功能在使增值固件得到保护的同时允许以后在其余的程序存储器中写入代码。

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

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

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

图 11.3　FLACL: FLASH 访问限制（只限于 C8051F005/06/07/15/16/17）

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

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

图 11.4　FLSCL：FLASH 存储器定时预分频

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
FOSE	FRAE	-	-	FLASCL				10001111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xB6

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

位 6：　FRAE：闪存一直读允许。
0：每个单稳态定时器间隔读一次闪存。
1：闪存总是处于读方式。

位 5-4：　未用。读=00b，写=忽略。

位 3-0：　FLASCL：FLASH 存储器定时预分频位
这些位指定（对于给定的系统时钟）产生正确的 FLASH 写/擦除操作时间所需要的预分频值。如果预分频值被设为 1111b，则 FLASH 写/擦除操作被禁止。
0000：系统时钟 <50kHz
0001：50kHz ≤系统时钟 < 100kHz
0010：100kHz ≤系统时钟 < 200kHz
0011：200kHz ≤系统时钟 < 400kHz
0100：400kHz ≤系统时钟 < 800kHz
0101：800kHz ≤系统时钟 < 1.6MHz
0110：1.6MHz ≤系统时钟 < 3.2MHz
0111：3.2MHz ≤系统时钟 < 6.4MHz
1000：6.4MHz ≤系统时钟 < 12.8MHz
1001：12.8MHz ≤系统时钟 < 25.6MHz
1010：25.6MHz ≤系统时钟 < 51.2MHz*
1011, 1100, 1101, 1110：保留值。
1111：闪存写/擦除被禁止

预分频值是满足下式的最小值：
FLASCL > $\log_2$(系统时钟/50kHz)
*用于测试目的。C8051F000 系列不能保证超过 25MHz 时的操作。

12．外部 RAM（C8051F005/06/07/15/16/17）

C8051F005/06/07/15/16/17 MCU 内部有位于外部数据存储器空间的 2048 字节 RAM。可以用外部传送指令（MOVX）和数据指针（DPTR）访问这些地址单元，或者用 MOVX 间接寻址方式。如果 MOVX 指令使用一个 8 位地址操作数（例如 @R1），则 16 位地址的高字节由外部存储器接口控制寄存器（EMI0CN，如图 12.1 所示）提供。**注：MOVX 指令还用于写 FLASH 存储器，详见第 11 章。缺省情况下（即 PSTCL.0 =0）MOVX 指令访问 XRAM。**

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

图 12.1　EMI0CN：外部存储器接口控制

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	-	PGSEL2	PGSEL1	PGSEL0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xAF

位 7-3：　未用一读为 00000b

位 2-0：　PGSEL[2-0]：XRAM 页选择位

当使用 8 位的 MOVX 命令时，XRAM 页选择位提供 16 位外部数据存储器地址的高字节，实际上是选择一个 256 字节的 RAM 页。高 5 位被忽略，使得 2k 的存储块以取模的形式重复映射到 64k 的外部数据存储器地址空间。

000：xxxxx000b
001：xxxxx001b
010：xxxxx010b
011：xxxxx011b
100：xxxxx100b
101：xxxxx101b
110：xxxxx110b
111：xxxxx111b

13. 复位源

MCU 的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，CIP-51 停止程序执行，将外部端口引脚置于一个已知状态，将 SFR 初始化为缺省值，禁止中断和定时器。在退出复位状态时，程序计数器被复位，程序从地址 0x0000 开始执行。

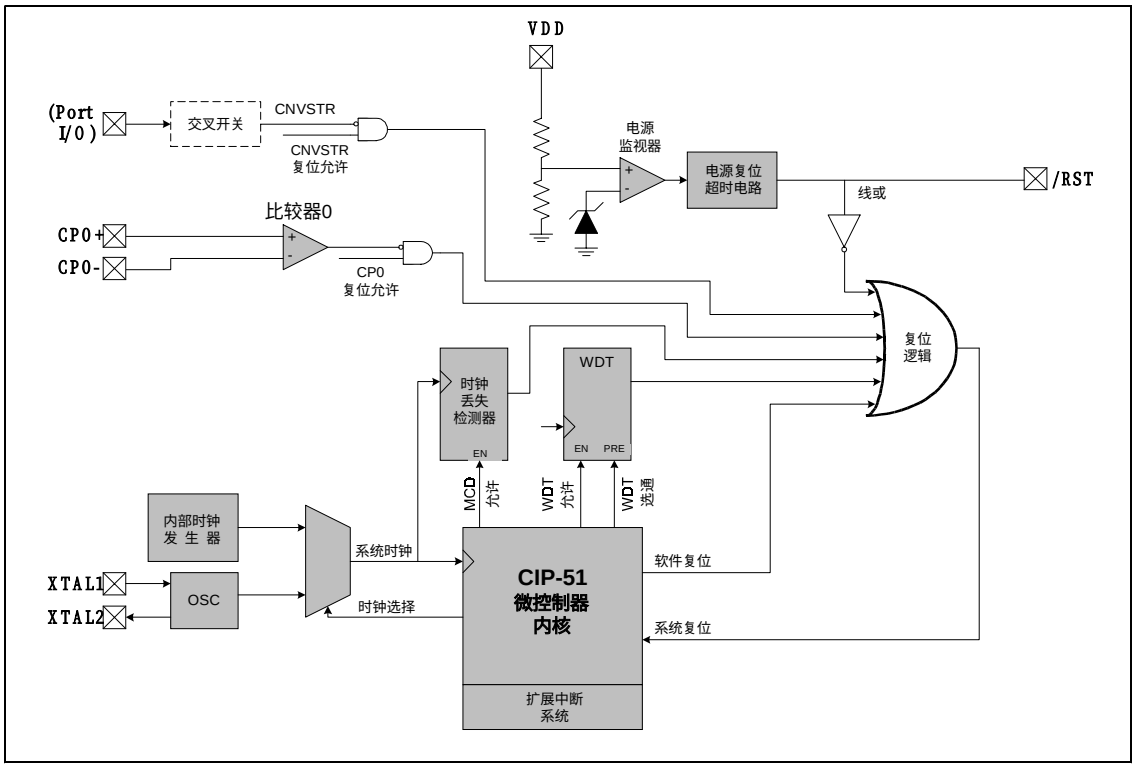
所有的 SFR 在复位后都被初始化为预定值。SFR 中各位的复位值在 SFR 的详细说明中定义。在复位期间内部数据存储器的内容不发生改变，复位前存储的数据保持不变。但由于堆栈指针 SFR 被复位，堆栈实际上已丢失，尽管堆栈中的数据未发生变化。

I/O 端口锁存器的复位值为 0xFF，内部弱上拉有效，使外部 I/O 引脚处于高电平状态。如果复位源是 VDD 监视器或是一个向 PORSF 的写 1 操作，则/RST 引脚被驱动为低电平，直到 VDD 复位结束。

在退出复位状态时，MCU 使用内部振荡器运行在 2MHz 作为默认的系统时钟。有关选择和配置系统时钟源的详细说明见第 14 章。看门狗定时器被允许，使用其最长的超时时间。（有关使用看门狗定时器的详细信息见第 13.8 节）

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

图 13.1 复位源框图



13.1 上电复位

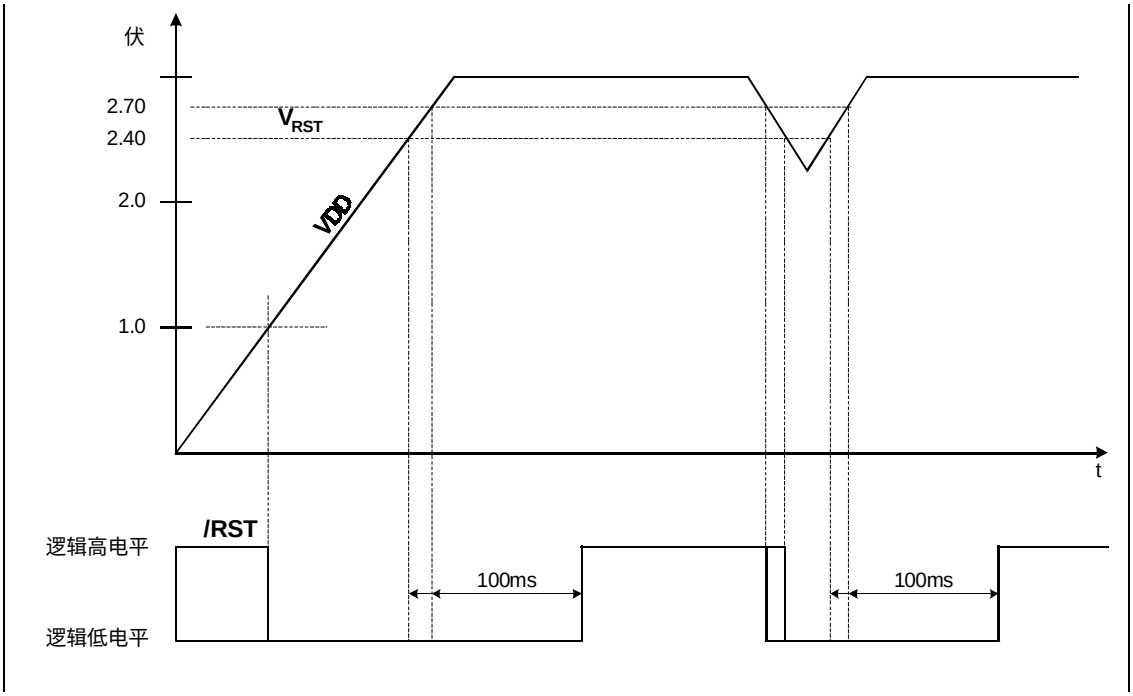
C8051F000 系列 MCU 内部有一个电源监视器，在上电期间该监视器使 MCU 保持在复位状态，直到 VDD 上升到超过 V_{RST} 电平。（见图 13.2 的时序图，有关电源监视器电路的电气特性见表 13.1。）/RST 引脚一直被置为低电平，直到 100 毫秒的 VDD 监视器超时时间结束，这 100 毫秒的等待时间是为了使 VDD 电源稳定。

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

13.2 软件强制复位

向 PORSF 位写 1 将强制产生一个上电复位，如 13.1 节所述。

图 13.2 VDD 监视器时序图



13.3 掉电复位

当发生掉电或因电源波动导致 VDD 降到 V_{RST} 以下时，电源监视器将/RST 引脚驱动为低电平并使 CIP-51 回到复位状态（见图 13.2）。当 VDD 又回到高于 V_{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 的数据不再有效。

13.4 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。在低有效的/RST 信号撤出后，MCU 将保持在复位状态至少 12 个时钟周期。从外部复位状态退出后，PINRSF 标志（RSTSRC.0）被置‘1’。/RST 引脚耐 5V 电压。

13.5 时钟丢失检测器复位

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

13.6 比较器 0 复位

向 C0RSEF 标志（RSTSRC.5）写‘1’可以将比较器 0 配置为复位源。应在写 C0RSEF 之前至少 20 微秒用 CPT0CN.7（见图 8.3）允许比较器 0，以防止通电瞬间在输出端产生抖动，从而产生不希望的复位。当被配置为复位源时，如果同相端输入电压（在 CP0+）小于反相端输入电压（在 CP0-），则 MCU 被置于复位状态。在发生比较器 0 复位之后，C0RSEF 标志（RSTSRC.5）的读出值为‘1’，表示本次复位源为比较器 0；否则该位读出值为‘0’。/RST 引脚的状态不受该复位的影响。比较器 0 在有或没有系统时钟的情况下都可产生复位。

13.7 外部 CNVSTR 引脚复位

向 CNVRSEF 标志（RSTSRC.6）写‘1’可以将外部 CNVSTR 信号配置为复位源。CNVSTR 信号可以出现在 P0、P1 或 P2 的任何 I/O 引脚，如 15.1 节所述。（注意，交叉开关必许被配置为使 CNVSTR 信号接到正确的端口 I/O。）应该在将 CNVSTR 配置为复位源之前配置并允许交叉开关。当被配置为复位源时，CNVSTR 为低电平有效。在发生 CNVSTR 复位之后，CNVRSEF 标志（RSTSRC.6）的读出值为‘1’，表示本次复位源为 CNVSTR；否则该位的读出值为‘0’。/RST 引脚的状态不受该复位的影响。

13.8 看门狗定时器复位

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

在从任何一种复位退出时，WDT 被自动允许并使用缺省的最大时间间隔。系统软件可以根据需要禁止 WDT 或锁定为运行状态以防止意外产生的禁止命令。WDT 一旦被锁定，在下一次系统复位之前将不能被禁止。/RST 引脚的状态不受该复位的影响。

13.8.1 看门狗的使用

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

允许/复位 WDT

向 WDTCN 寄存器写入 0xA5 将允许看门狗定时器并使其开始减 1 计数。用户的应用软件应周期性地向 WDTCN 写入 0xA5，以防止看门狗定时器溢出。每次系统复位都将允许并启动 WDT。

禁止 WDT

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

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

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

锁定 WDT

向 WDTCN 写入 0xFF 将使禁止功能无效。一旦锁定，在下一次复位之前禁止操作将被忽略。写 0xFF 并不允许或复位**看门狗定时器。如果应用程序想一直使用看门狗，则应在初始化代码中向 WDTCN 写入 0xFF。**

设置 WDT 定时间隔

WDTCN.[2:0]控制看门狗超时间隔。超时间隔由下式给出：

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

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

图 13.3 WDTCN：看门狗定时器控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								xxxxx111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xFF

位 7-0： WDT 控制
写入 0xA5 将允许并重装载 WDT。
写入 0xDE 后四个系统周期内写入 0xAD，将禁止 WDT。
写入 0xFF 锁定禁止功能。

位 4： 看门狗状态位（读）
读 WDTCN.[4]得到看门狗定时器的状态。
0：WDT 处于不活动状态。
1：WDT 处于活动状态。

位 2-0： 看门狗超时间隔位
位 WDTCN.[2:0]设置看门狗超时间隔位。在写这些位时，WDTCN.7 必须被置为 0。

图 13.4　RSTSRC：复位源寄存器

R	R/W	R/W	R/W	R	R	R/W	R	复位值
JTAGRST	CNVRSEF	C0RSEF	SWRSEF	WDTRSF	MCDRSF	P0RSF	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）

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

位 6：　CNVRSEF：转换启动复位源允许和标志。
写
0：CNVSTR 不是复位源。
1：CNVSTR 是复位源（低电平有效）。
读：
0：最后一次复位不是来自 CNVSTR。
1：最后一次复位来自 CNVSTR。

位 5：　C0RSEF：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：强制产生一个内部复位。/RST 引脚不受影响。
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
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：强制上电复位和标志
写
0：无作用
1：强制产生一个上电复位。/RST 引脚被驱动为低电平。
读：
0：最后一次复位不是来自 POR。
1：最后一次复位来自 POR。

位 0：　PINRSF：HW 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。

表 13.1. 复位源电气特性

-40℃到+85℃（除非另有说明）

参　数	条　件	最小值	典型值	最大值	单 位
/RST 输出低电平	I_{OL}=8.5mA, VDD=2.7 到 3.6V			0.6	V
/RST 输入高电平		0.7x VDD			V
/RST 输入低电平				0.3x VDD	V
/RST 输入漏电流	/RST=0.0V		20		µA
/RST 输出有效 VDD		1.0			V
/RST 输出有效 AV+		1.0			V
VDD 上电门限(V_{RST})		2.40	2.55	2.70	V
复位时间延迟	从 VDD 超过复位门限到/RST 的上升沿	80	100	120	ms
时钟丢失检测器超时	从最后一个系统时钟到产生复位	100	220	500	µs

14. 振荡器

每个 MCU 都有一个内部振荡器和一个外部振荡器驱动电路，每个驱动电路都能产生系统时钟。MCU 在复位后从内部振荡器启动。内部振荡器的启动是瞬间完成的。内部振荡器可以被允许/禁止，其振荡频率可以用内部振荡器控制寄存器（OSCICN）改变，如图 14.2 所示。表 14.1 给出了内部振荡器的电气特性。

当/RST 引脚为低电平时，两个振荡器都被禁止。MCU 可以从内部振荡器或外部振荡器运行，可使用 OSCICN 寄存器中的 CLKSL 位在两个振荡器之间随意切换。外部振荡器需要一个外部谐振器、并行方式的晶体、电容或 RC 网络连接到 XTAL1/XTAL2 引脚（见图 14.1）。必须在 OSCXCN 寄存器中为这些振荡源中的某一个配置振荡器电路。一个外部 CMOS 时钟也可以通过驱动 XTAL1 引脚提供系统时钟。XTAL1 和 XTAL2 引脚的耐压值是 3.6V（不是 5V）。即使在 MCU 已经切换到内部振荡器时，外部振荡器仍可保持允许状态并运行。

图 14.1 振荡器框图

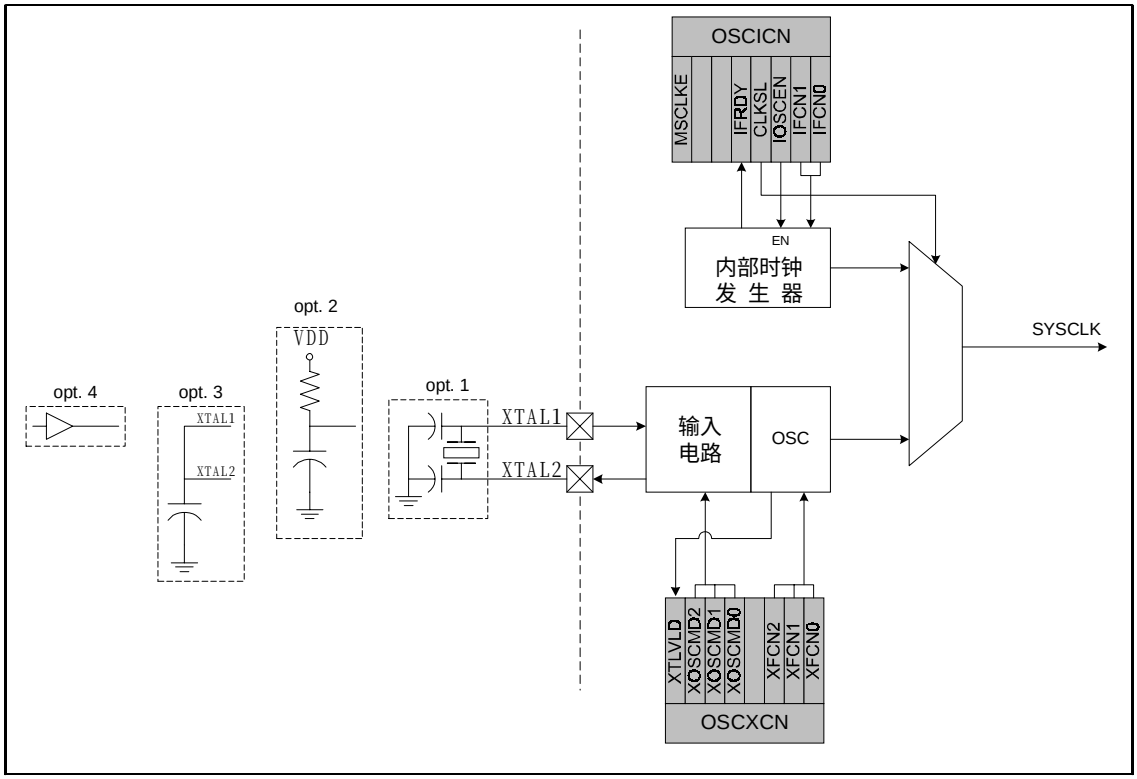


图 14.2　OSCICN：内部振荡器控制寄存器

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	复位值
MSCLKE	-	-	IFRDY	CLKSL	IOSCEN	IFCN1	IFCN0	00000100
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xB2

位 7：　MSCLKE：时钟丢失允许位
0：禁止时钟丢失检测器。
1：允许时钟丢失检测器；检测到时钟丢失将触发复位。

位 6-5：　未用。读=00b，写=忽略。

位 4：　IFRDY：内部振荡器频率准备好标志
0：内部振荡器频率不是按 IFCN 位指定的速度运行。
1：内部振荡器频率按照 IFCN 位指定的速度运行。

位 3：　CLKSL：系统时钟源选择位
0：选择内部时钟源作为系统时钟。
1：选择外部时钟源作为系统时钟。

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：内部振荡器典型频率为 2MHz。
01：内部振荡器典型频率为 4MHz。
10：内部振荡器典型频率为 8MHz。
11：内部振荡器典型频率为 16MHz。

表 14.1 内部振荡器电气特性

-40℃to+85℃（除非另有说明）。

参　数	条　件	最小值	典型值	最大值	单位
内部振荡器频率	OSCICN.[1:0]=00 OSCICN.[1:0]=01 OSCICN.[1:0]=10 OSCICN.[1:0]=11	1.5 3.1 6.2 12.3	2 4 8 16	2.4 4.8 9.6 19.2	MHz
内部振荡器电流消耗(从 VDD)	OSCICN.2=1		200		μA
内部振荡器温度稳定性			4		ppm/℃
内部振荡器电源（VDD）稳定性			6.4		%/V

图 14.3　OSCXCN：外部振荡器控制寄存器

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	-	XFCN2	XFCN1	XFCN0	00110000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xB1

位 7：　XTLVLD：晶体振荡器有效标志
（只有当 XOSCMD＝1xx 时有效）
0：晶体振荡器未用或未稳定。
1：晶体振荡器正在运行并且工作稳定（为了避免瞬态条件，应在允许晶体振荡器工作 1ms 后开始读）。

位 6-4：　XOSCMD2-0：外部振荡器方式位
00x：关闭。XTAL1 引脚内部接地。
010：系统时钟为来自 XTAL1 引脚的外部 CMOS 时钟。
011：系统时钟为来自 XTAL1 引脚的外部 CMOS 时钟的二分频。
10x：RC/C 振荡器方式二分频。
110：晶体振荡器方式。
111：晶体振荡器方式二分频。

位 3：　保留。读=无定义，写=忽略。

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

XFCN	晶体（XOSCMD=11x）	RC(XOSCMD=10x)	C(XOSCMD=10x)
000	F ≤ 12.5kHz	F ≤ 25kHz	K 因子= 0.44
001	12.5kHz < f ≤ 30.3kHz	25kHz < f ≤50kHz	K 因子= 1.4
010	30.35kHz < f ≤93.8kHz	50kHz < f ≤100kHz	K 因子= 4.4
011	93.8kHz < f ≤267kHz	100kHz < f ≤200kHz	K 因子= 13
100	267kHz < f ≤722kHz	200kHz < f ≤400kHz	K 因子= 38
101	722kHz < f ≤2.23MHz	400kHz < f ≤800kHz	K 因子= 100
110	2.23MHz < f≤6.74MHz	800kHz < f ≤1.6MHz	K 因子= 420
111	F > 6.74MHz	1.6MHz < f ≤3.2MHz	K 因子= 1400

晶体方式（电路见图 14.1，选项 1；XOSCMD=11x）
选择 XFCN 值，以匹配晶体或陶瓷谐振器频率。

RC 方式（电路见图 14.1，选项 2；XOSCMD=10x）
选择振荡器频率范围：
$f=1.23(10^3)/(R*C)$，其中：
f=以 MHz 为单位的振荡器频率
C=以 pF 为单位的电容值
R=以 kΩ为单位的上拉电阻值

C 方式（电路见图 14.1，选项 3；XOSCMD=10x）
选择振荡器频率由下式决定：
f=KF/(C*AV+)，其中：
f=以 MHz 为单位的振荡器频率
C=XTAL1、XTAL2 引脚上的电容值，以 pF 为单位
AV＋=供给 MCU 的模拟电源电压值，以伏特为单位

14.1 外部晶体举例

如果使用晶体或陶瓷谐振器产生 MCU 的时钟，则电路为图 14.1 中的选项 1。对于一个 ECS-110.5-20-4 的晶体，其谐振频率为 11.0592MHz，固有电容为 7.0 pF，ESR 为 60Ω。每个补偿电容应为 33pF，PWB 寄生电容约为 2pF。从图 14.3 的表中的晶体列查出合适的外部振荡器频率控制值（XFCN）应为 111b。

在晶体振荡器被允许时，晶体驱动器的输出端 XTAL2 脚会出现一个瞬时脉冲，该脉冲足以在晶体实际启动前将 OSCXCN 中的 XTLVLD 位置‘1’。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切换到外部振荡器。建议的过程为：

1. 允许外部振荡器
2. 等待 1ms
3. 查询 XTLVLD ‘0’➔’1’
4. 切换到外部振荡器

在外部振荡器稳定之前就切换到外部振荡器可能导致不可预料的后果。

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

14.2 外部 RC 举例

如使用 RC 网络产生 MCU 的时钟，则电路为图 14.1 中的选项 2。电容不能大于 100pF，但由于 PWB 寄生电容的影响，使用很小的电容将增加频率偏差。为了确定 OSCXCN 寄存器中所需要的外部振荡器频率控制值（XFCN），首先选择能产生所要求的振荡频率的 RC 网络值。如果所希望的频率是 100kHz，选 R=246kΩ，C=50pF：

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

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

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

$XFCN \geq 2$，或代码 010

14.3 外部电容举例

如使用外部电容产生 MCU 的时钟，则电路为图 14.1 中的选项 3。电容不能大于 100pF，但由于 PWB 寄生电容的影响，使用很小的电容将增加频率偏差。为了确定 OSCXCN 寄存器中所需要的外部振荡器频率控制值（XFCN），选择要用的电容并利用下面的方程计算振荡频率。假设 AV+ = 3.0V，C=50pF：

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

$f = KF / 150$

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

$f = 13 / 150 = 0.087MHz$，或 87kHz。

因此，本例中要用的 XFCN 值为 011。

15. 端口输入/输出

MCU 中有大量的数字资源需要通过数字 I/O 端口 P0、P1、P2 和 P3 才能使用。端口 0、1 和 2 中的每个引脚即可定义为对应的端口 I/O，又可以分配给一个内部数字资源，如图 15.1 所示。设计者对功能分配有完全的控制，只受所选封装的可用引脚数的限制（C8051F000/05/15 有 4 个端口具有对应的引脚，F001/06/16 有 P0 和 P1 具有对应的引脚，F002/07/17 只有 P0 具有对应的引脚）。这种资源分配的灵活性是通过使用优先权交叉开关译码器实现的。（注意，不管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 15.1 将所选择的内部数字资源分配到 I/O 引脚。寄存器 XBR0、XBR1 和 XBR2（分别在图 15.3、图 15.4 和图 15.5 中定义）用于选择内部数字功能或让 I/O 引脚默认为端口 I/O。每个 MCU 的交叉开关的功能完全一样，只是 P2 在 F001/06/16 中没有对应的引脚，P1 和 P2 在 F002/07/17 中没有对应的引脚。数字资源如果被分配到没有引脚的端口，则它不能被访问。

所有的端口 I/O 都耐 5V 电压（参见图 15.2 的 I/O 单元电路）。可以用端口配置寄存器（PRT0CF、PRT1CF、PRT2CF、PRT3CF）将端口 I/O 单元配置为推挽或漏极开路方式。表 15.2 给出了端口 I/O 的电气特性。

15.1 优先权交叉开关译码器

该系列 MCU 的设计目标之一是使所有的数字资源都能为设计者所用，即使在使用引脚少的封装时也是如此。优先权交叉开关译码器为连接内部数字资源到物理 I/O 引脚提供了很好的解决方案。

优先权交叉开关译码表（表 15.1）为每个 I/O 功能分配一个优先权，从顶端 SMBus 开始。如该表所示，当被选择时，它的两个信号将被分配到 I/O 端口 0 的引脚 0 和引脚 1。译码器总是从端口 0 开始，按 LSB 到 MSB 的顺序填充 I/O 位，然后是端口 1，如果需要的话最后填充端口 2。如果你选择不用某个资源，则表中的下一个功能将填充这个优先权位置。这样就可以只选择设计中用到的功能，充分利用可用的 I/O 引脚。另外，任何未分配的端口 I/O 被组合到一起，更便于在应用代码中使用。

寄存器 XBR0、XBR1 和 XBR2 用于将数字 I/O 资源分配到物理 I/O 端口引脚。当选择了 SMBus、SPI 总线或 UART 时，交叉开关将为选择的总线分配所有对应的引脚，理解这一点是很重要的。例如，不能为 UART 功能只分配 RX 引脚而不分配 TX 引脚。在优先功能都被分配之后，标准端口 I/O 是连续的。例如，如果你选择的功能使用前 14 个端口 I/O（P0.[7:0], P1.[5:0]），则将有 18 个端口 I/O 未被交叉开关使用（P1.[7:6]，P2 和 P3）。

15.2 端口 I/O 初始化

端口 I/O 初始化是很简单的。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’XBAR2 中的 XBARE 位将允许交叉开关。**不管 XBRn 寄存器的设置为何，在交叉开关被允许之前，外部引脚保持标准端口的输入方式。**对于给定

的 XBRn 设置，可以使用优先权表确定 I/O 引脚分配；另一种方法是使用 IDE 软件的代码配置向导功能来确定基于 XBRn 寄存器设置的 I/O 引脚分配。

I/O 引脚的输出驱动器特性是用端口配置寄存器 PRT0CF、PRT1CF、PRT2CF 和 PRT3CF 定义的（见图 15.7、图 15.9、图 15.12、图 15.14）。每个端口输出驱动器都可被配置为漏极开路或推挽方式。即使对于在 XBRn 寄存器中选择的数字资源也需要定义，而不是自动定义的。例外的情况是 SMBus（SDA、SCL）和 UART 接收（RX，在方式 0 时）引脚，这些引脚是漏极开路的，与 PRTnCF 的设置无关。当 XBR2 中的 WEAKUD 位为 0 时，被配置为漏极开路的所有端口 I/O 都被允许为弱上拉。WEAKUD 不影响推挽方式的端口 I/O。另外，输出为 0 的漏极开路单元的弱上拉被关闭，以避免不必要的功耗。

第三步和最后一步是用相应的设置寄存器对所选择的每个资源进行初始化。对不同数字资源的初始化过程的详细说明见相关章节。每个寄存器的复位状态示于描述每个寄存器的图中。

图 15.1 端口 I/O 功能框图

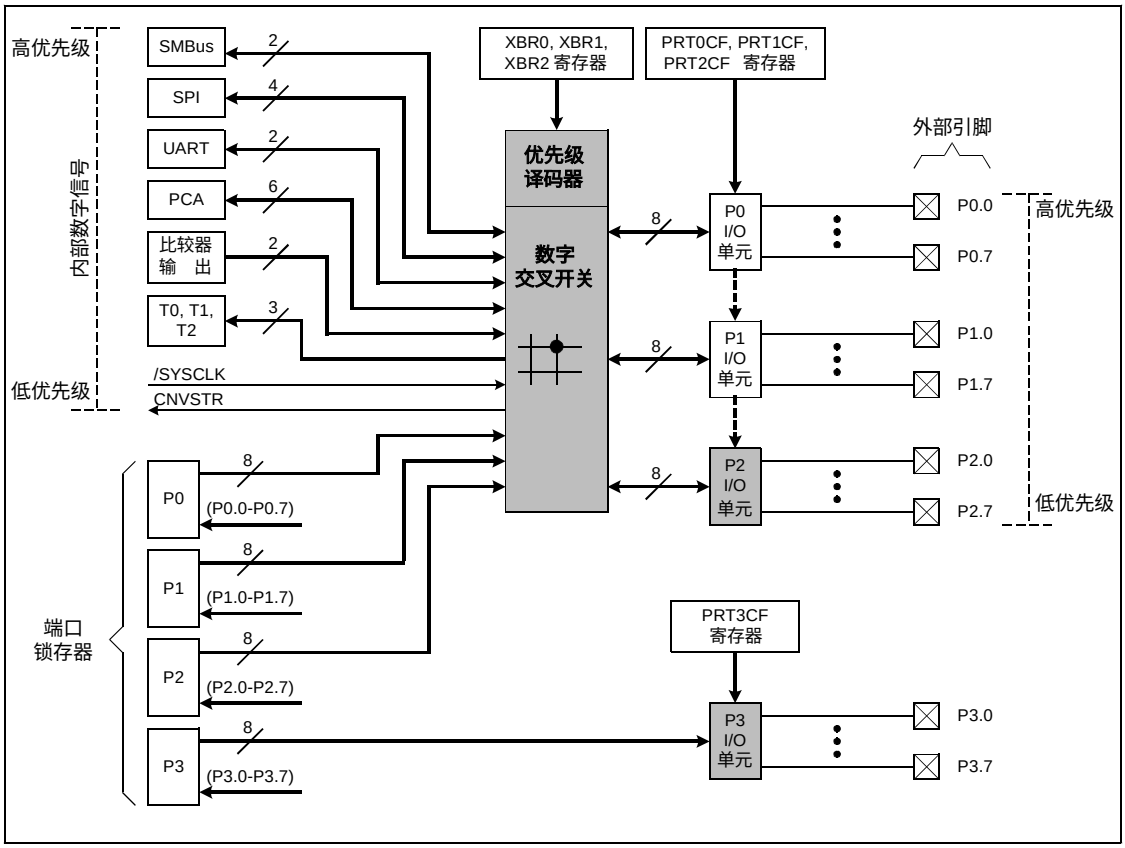


图 15.2 端口 I/O 单元框图

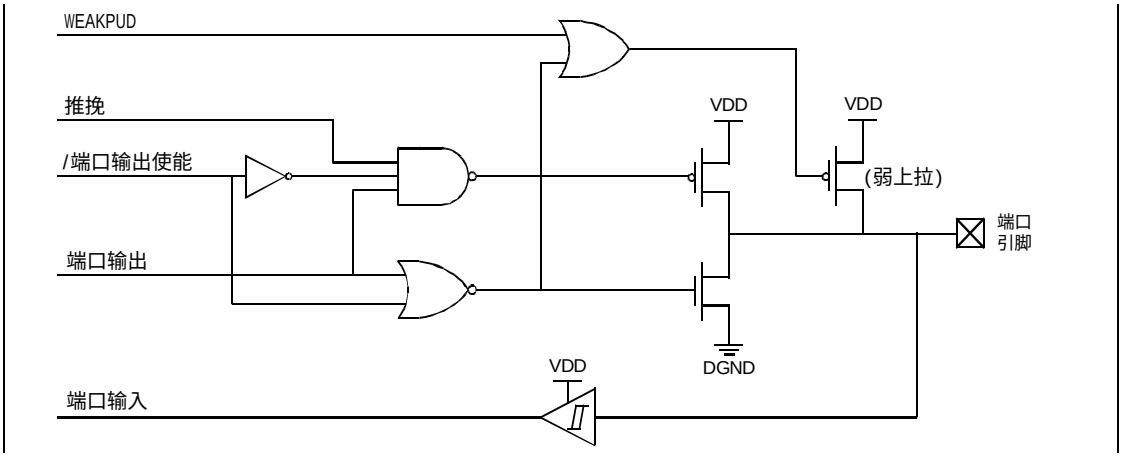


表 15.1 交叉开关优先权译码表

	P0								P1								P2							
I/O 引脚	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
SDA	●																							
SCL		●																						
SCK	●		●																					
MISO		●		●																				
MOSI			●		●																			
NSS				●		●																		
TX	●		●		●		●																	
RX		●		●		●		●																
CEX0	●		●		●		●		●															
CEX1		●		●		●		●		●														
CEX2			●		●		●		●		●													
CEX3				●		●		●		●		●												
CEX4					●		●		●		●		●											
ECI	●	●	●	●	●	●	●	●	●	●	●	●	●	●										
CP0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									
CP1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
T0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							
/INT0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						
T1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					
/INT1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				
T2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			
T2EX	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		
/SYSCLK	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
CNVSTR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

在优先权交叉开关译码表中，黑圆点（●）用于指示每个信号（行）可以被用户应用代码分配（通过对寄存器XBR2、XBR1和XBR0编程）的外部端口I/O引脚（列）。

图 15.3　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
CP0OEN	ECIE	PCA0ME			UARTEN	SPI0EN	SMB0OEN	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xE1

位 7：　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 6：　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。

位 5-3：　PCA0ME：PCA 模块 I/O 允许位
000：所有的 PCA I/O 不连到端口引脚。
001：CEX0 连到端口引脚。
010：CEX0、CEX1 连到 2 个端口引脚。
011：CEX0、CEX1、CEX2 连到 3 个端口引脚。
100：CEX0、CEX1、CEX2、CEX3 连到 4 个端口引脚。
101：CEX0、CEX1、CEX2、CEX3、CEX4 连到 5 个端口引脚。
110：保留
111：保留

位 2：　UARTEN：UART I/O 允许位。
0：UART I/O 不连到端口引脚。
1：RX、TX 连到 2 个端口引脚。

位 1：　SPI0OEN：SPI 总线 I/O 允许位。
0：SPI I/O 不连到端口引脚。
1：MISO、MOSI、SCK 和 NSS 连到 4 个端口引脚。

位 0：　SMB0OEN：SMBus 总线 I/O 允许位
0：SMBus I/O 不连到端口引脚。
1：SDA 连到 P0.0 ，SCL 连到 P0.1。

图 15.4　XBR1：端口 I/O 交叉开关寄存器 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SYSCKE	T2EXE	T2E	INT1E	T1E	INT0E	T0E	CP1OEN	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xE2

位 7：　SYSCKE：SYSCLK 输出允许位
0：SYSCLK 不连到端口引脚。
1：SYSCLK 连到端口引脚。

位 6：　T2EXE：T2EX 允许位
0：T2EX 不连到端口引脚。
1：T2EX 连到端口引脚。

位 5：　T2E：T2 允许位
0：T2 不连到端口引脚。
1：T2 连到端口引脚。

位 4：　INT1E：/INT1 允许位。
0：/INT1 不连到端口引脚。
1：/INT1 连到端口引脚。

位 3：　T1E：T1 允许位
0：T1 不连到端口引脚。
1：T1 连到端口引脚。

位 2：　INT0E：/INT0 允许位
0：/INT0 不连到端口引脚。
1：/INT0 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：　CP1OEN：比较器 1 输出允许位
0：CP1 不连到端口引脚。
1：CP1 连到端口引脚。

图 15.5 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	-	-	-	-	-	CNVSTE	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（除了 I/O 被配置为推挽方式的端口）
　　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 5-1：　未用。读＝00000b，写=忽略。

位 0：　CNVSTE：ADC 转换启动输入允许位
　　0：CNVSTR 不连到端口引脚。
　　1：CNVSTR 连到端口引脚。

XBR0、XBR1、XBR2用法示例：

当被选中时，数字资源按顺序（如优先权交叉开关译码表中所示，从上往下）分配端口I/O引脚，从P0.0到P0.7，然后从P1.0到P1.7，最后从P2.0到P2.7。如果数字资源没有被连到I/0端口引脚，则这些端口引脚按缺省状态对应于内部端口寄存器位。

例1：如果XBR0 = 0x11，XBR1 = 0x00，XBR2 = 0x40：
P0.0=SDA，P0.1=SCL，P0.2=CEX0，P0.3=CEX1，P0.4 – P2.7 分配给相应的端口I/O。

例2：如果XBR0 = 0x80，XBR1 = 0x04，XBR2 = 0x41：
P0.0=CP0，P0.1=/INT0，P0.2 = CNVSTR，P0.3 – P2.7 分配给相应的端口 I/O。

15.3 通用端口 I/O

每个 MCU 有 4 个 8 位的双向并行端口，这些端口可用作通用 I/O。通过相应的特殊功能寄存器（SFR）访问每个端口，而这些 SFR 既可以按字节寻址又可以按位寻址。当写一个端口时，写到 SFR 的数据被锁存，以保持引脚的输出数据值不变。当进行读操作时，端口输入引脚的逻辑电平被读入，与 XBRn 的设置无关（即使该引脚被交叉开关分配给另一个信号，端口寄存器总是读它对应的端口 I/O 引脚）。例外的情况是执行*读-修改-写*指令。对一个端口 SFR 操作的*读-修改-写*指令如下：ANL、ORL、XRL、JBC、CPL、INC、DEC、DJNZ 和 MOV、CLR、SET（当目的地址是一个端口 SFR 中的一个位地址时）。这些指令读取端口寄存器的值，对其修改，再写回到 SFR。

15.4 配置无对应引脚的端口

P2 和 P3 在 F001/06/16 中没有对应的引脚。P1、P2 和 P3 在 F002/02/17 中没有对应的引脚。在这些缩减引脚的 MCU 中，这些没有对应引脚的端口寄存器（和对应的中断）仍然可为软件所用。不管软件中是否用到，建议不要使这些端口驱动器进入高阻状态。为防止发生这种情况，可以在复位后用 XBR2 将它们配置为弱上拉。建议用相应的 PRTnCF 寄存器将没有对应引脚的端口输出驱动器配置为推挽方式。这样一来，即使弱上拉被禁止，驱动器也不可能进入高阻状态。

图 15.6　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 PRT0CF.n 位=0，则为高阻状态）
（读—与 XBR0、XBR1 和 XBR2 寄存器的设置无关）
0：P0.n 为逻辑低电平。
1：P0.n 为逻辑高电平。

图 15.7 PRT0CF：端口 0 配置寄存器

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

位 7-0：　PRT0CF.[7:0]：P0.7-P0.0 输出配置位（分别对应）
0：对应的 P0.n 输出方式为漏极开路。
1：对应的 P0.n 输出方式为推挽。
（注：当 SDA、SCL 和 RX 出现在 P0 I/O 的任何引脚时，均为开漏输出，与 PRT0CF 的设置无关）。

图 15.8　P1：端口 1 寄存器

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

位 7-0：　P1.[7:0]
（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 PRT1CF.n 位=0，则为高阻状态）
（读 － 与 XBR0、XBR1 和 XBR2 寄存器配置无关）
0：P1.n 为逻辑低电平。
1：P1.n 为逻辑高电平。

图 15.9　PR1CF：端口 1 配置寄存器

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

位 7-0：　PRT1CF.[7:0]：P1.7-P1.0 输出配置位（分别对应）
0：对应的 P1.n 输出方式为漏极开路。
1：对应的 P1.n 输出方式为推挽

图 15.10　PRT1IF：端口 1 中断标志位寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
IE7	IE6	IE5	IE4	-	-	-	-	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xAD

位 7：　IE7：外部中断 7 标志位
0：P1.7 引脚没有检测到下降沿。
1：当检测到 P1.7 引脚的下降沿时，该标志由硬件置位。
位 6：　IE6：外部中断 6 标志位
0：P1.6 引脚没有检测到下降沿。
1：当检测到 P1.6 引脚的下降沿时，该标志由硬件置位。
位 5：　IE5：外部中断 5 标志位
0：P1.5 引脚没有检测到下降沿。
1：当检测到 P1.5 引脚的下降沿时，该标志由硬件置位。
位 4：　IE4：外部中断 4 标志位
0：P1.4 引脚没有检测到下降沿。
1：当检测到 P1.4 引脚的下降沿时，该标志由硬件置位。
位 3-0：　未用。读=0000b，写=忽略。

图 15.11 P2：端口 2 寄存器

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

位 7-0： P2.[7:0]
（写 — 输出出现在 I/O 引脚，根据 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 PRT2CF.n 位=0，则为高阻状态）
（读—与 XBR0、XBR1 和 XBR2 寄存器配置无关）
0：P2.n 为逻辑低电平。
1：P2.n 为逻辑高电平。

图 15.12 PR2CF：端口 2 配置寄存器

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

位 7-0： PRT2CF.[7:0]：P2.7-P2.0 输出配置位（分别对应）
0：对应的 P2.n 输出方式为漏极开路。
1：对应的 P2.n 输出方式为推挽。

图 15.13　P3：端口 3 寄存器

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

位 7-0：　P3.[7:0]
（写 一 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 PRT3CF.n 位=0，则为高阻状态）
（读一与 XBR0、XBR1 和 XBR2 寄存器配置无关）
0：P3.n 为逻辑低电平。
1：P3.n 为逻辑高电平。

图 15.14　PR3CF：端口 3 配置寄存器

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

位 7-0：　PRT3CF.[7:0]：P3.7-P3.0 输出配置位（分别对应）
0：对应的 P3.n 输出方式为漏极开路。
1：对应的 P3.n 输出方式为推挽。

表 15.2　端口 I/O DC 电气特性

VDD = 2.7V – 3.6V, -40℃到+85℃（除非另有说明）。

参　数	条　件	最小值	典型值	最大值	单 位
输出高电压	I_{OH} = -10μA，端口 I/O 为推挽方式 I_{OH} = -3mA，端口 I/O 为推挽方式 I_{OH} = -10mA，端口 I/O 为推挽方式	VDD-0.1 VDD-0.7	 VDD-0.8		V
输出低电压	I_{OL} = 10μA I_{OL} = 8.5mA I_{OL} = 25mA		 1.0	0.1 0.6	V
输入高电压		0.7×VDD			V
输入低电压				0.3×VDD	V
输入漏电流	DGND < 端口引脚 < VDD，高阻态 弱上拉关闭 弱上拉打开		 30	±1	μA
容性负载			5		pF

16. SMBus

SMBus 串行 I/O 接口完全符合系统管理总线规范 1.1 版。它是一个双线的双向串行总线，与 I^2C 串行总线兼容。系统控制器对总线的读写操作都是以字节为单位的，由 SMBus 接口自动控制数据的串行传输。数据传输的最大速率可达系统时钟的八分之一（这可能比 SMBus 的规定速度要快，取决于所使用的系统时钟）。可以采用延长低电平时间的方法适应同一总线上不同速度的器件。

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和被寻址的从发送器到主接收器。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。

图 16.1　SMBus 原理框图

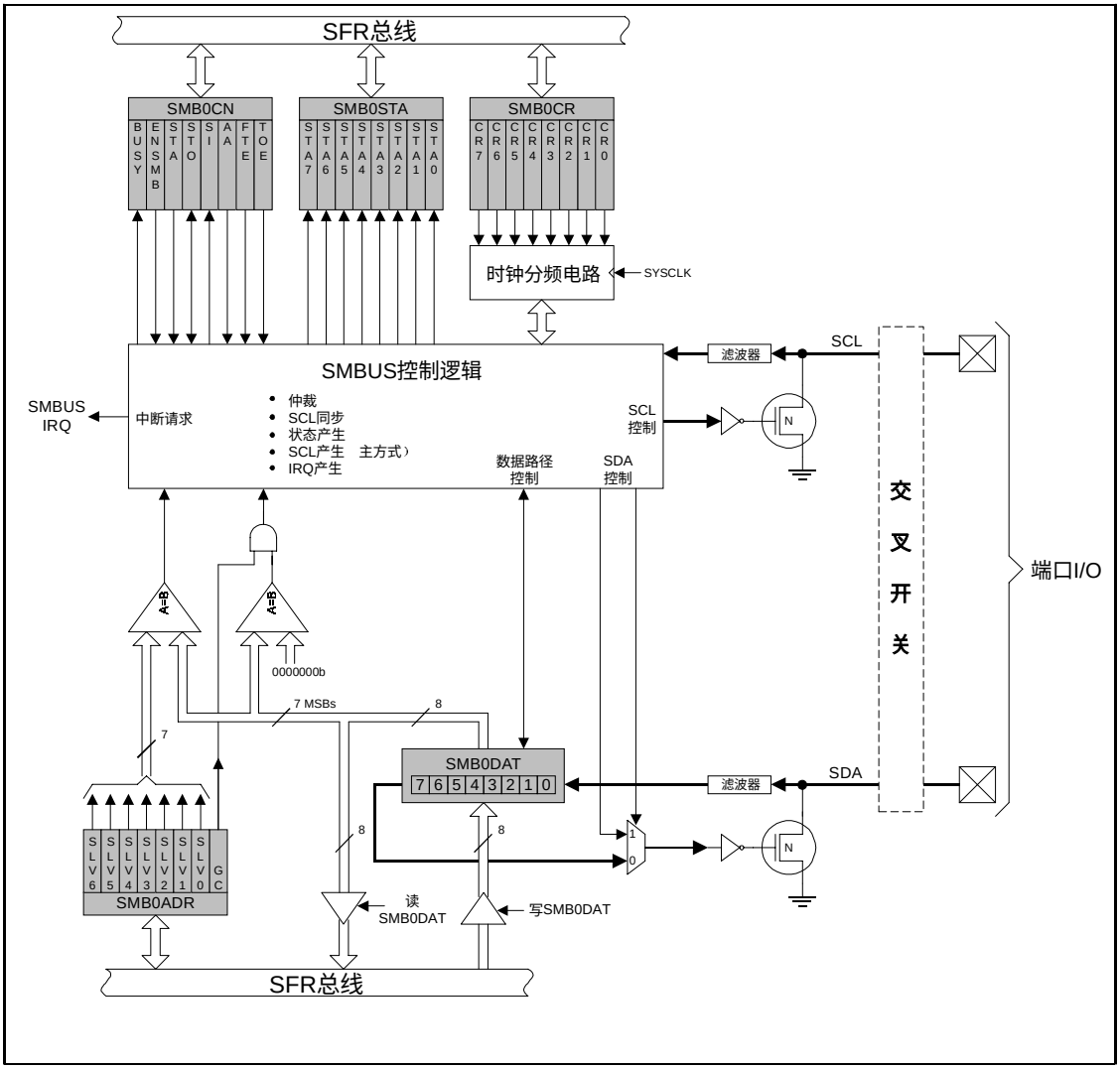
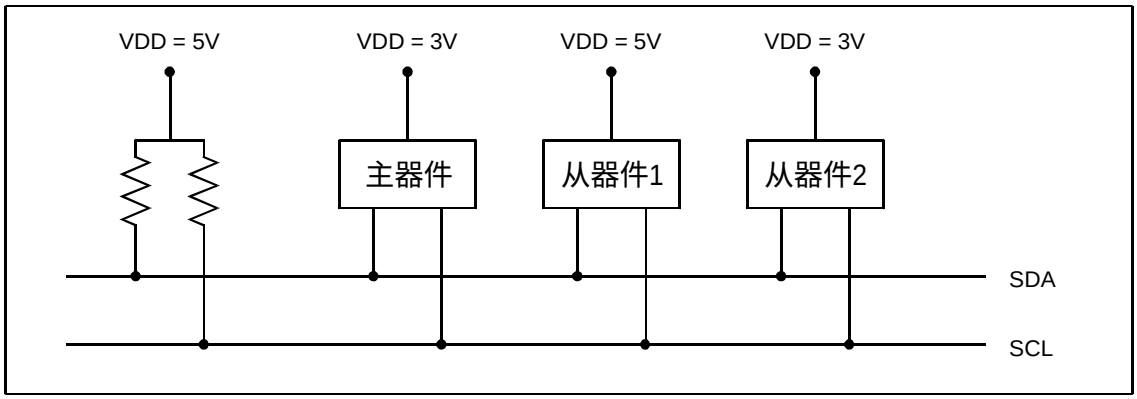


图 16.2 给出了一个典型的 SMBus 配置。SMBus 接口的工作电压可以在 3.0V 和 5.0V 之间，总线上不同器件的工作电压可以不同。SCL（串行时钟）和 SDA（串行数据）线是双向的。必须通过上拉电阻或类似电路将它们连到电源电压。当总线空闲时，这两条线都被拉到高电平。连接在总线上的每个器件的 SCL 和 SDA 都必须是漏极开路或集电极开路的。总线上的最大器件数只受所要求的上升和下降时间的限制，上升和下降时间分别不能超过 300ns 和 1000ns。

图 16.2 典型 SMBus 配置



16.1 支持文档

假设读者熟悉或有条件阅读下列支持文档：

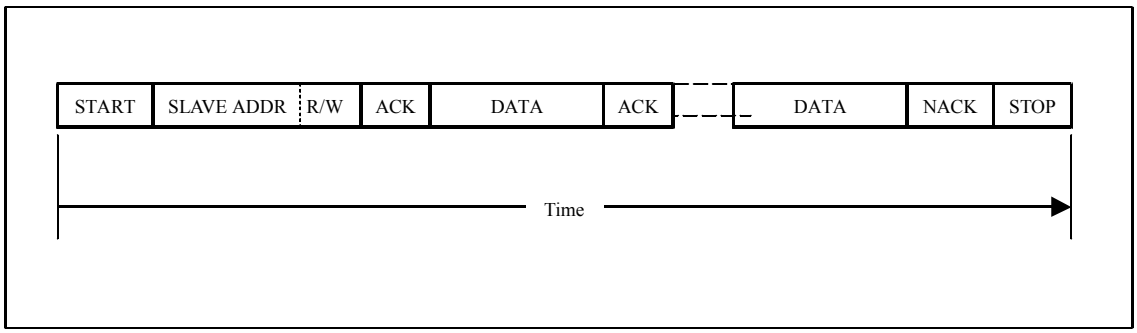
1. I^2C 总线及使用（包含规范），菲力浦半导体。
2. I^2C 总线规范—2.0 版，菲力浦半导体。
3. 系统管理总线规范—1.1 版，SBS

16.2 工作过程

一次典型的 SMBus 数据交换包括一个起始条件（START）、一个地址字节、一个或多个字节数据和一个停止（STOP）条件。每个地址字节和每个数据字节后面都跟随一个来自接收器的确认（ACKNOWLEDGE）位。地址字节包含一个 7 位的地址和一个方向位（R/W）。方向位占据地址字节的最低位。方向位被设置为逻辑‘1’时表示这是一个“读”（READ）操作，方向位为逻辑‘0’表示这是一个“写”（WRITE）操作。所有从器件都能识别一个通用呼叫地址（0x00 + R/W），这就允许一个主器件同时访问多个从器件。

所有的数据交换都由主器件启动，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。如果本次数据交换是一个从主器件到从器件的写操作，则主器件每发送一个数据字节后等待来自从器件的确认。如果是一个读操作，则由从器件发送数据并等待主器件的确认。在数据传输结束时，主器件产生一个停止条件，结束数据交换并释放总线。图 16.3 给出了一次典型 SMBus 数据传输示意图。

图 16.3 SMBus 数据传输



SMBus 接口可以被配置为工作在主方式或从方式。在某一时刻，它将工作在下述 4 种方式之一：

16.2.1 主发送器方式

在 SDA 上发送串行数据，在 SCL 上输出串行时钟。发送的第一个字节含有目标从器件的地址和数据方向位。在这种情况下数据方向位（R/W）应为逻辑 0，表示这是一个“写”操作。主发送器接着发送一个或多个字节的串行数据。在发送完每个字节后，由从器件产生一个确认位。为了指示串行传输的开始和结束，主器件输出起始和停止条件。

16.2.2 主接收器方式

在 SDA 上接收串行数据，在 SCL 上输出串行时钟。主器件发送第一个字节，该字节含有目标从器件的地址和数据方向位。在这种情况下数据方向位（R/W）应为逻辑 1，表示这是一个“读”操作。接着从 SDA 接收来自从器件的串行数据并在 SCL 上输出串行时钟。从器件发送一个或多个字节的串行数据。每收到一个字节后，由主器件发送一个确认位。主器件输出起始和停止条件指示串行传输的开始和结束。

16.2.3 从发送器方式

在 SDA 上发送串行数据，在 SCL 上接收串行时钟。首先收到一个含有地址和数据方向位的字节。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。如果收到的地址与从器件本身的从地址一致（或收到一个通用呼叫地址），则向主器件发送一个或多个字节的串行数据。在收到每个字节后，由主器件发送一个确认位。主器件输出起始和停止条件指示串行传输的开始和结束。

16.2.4 从接收器方式

在 SDA 上接收串行数据，在 SCL 上接收串行时钟。首先收到一个含有地址和数据方向位的字节。在这种情况下数据方向位（R/W）应为逻辑 0，表示这是一个“写”操作。如果收到的地址与从器件本身的从地址一致（或收到一个通用呼叫地址），则接着从主器件接收一个或多个字节的串行数据。每收到一个字节后，发送一个确认位。主器件输出起始和停止条件指示串行传输的开始和结束。

16.3 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，所以使用仲裁机制解决这一问题。这些主器件继续发送起始条件，直到其中一个主器件发送一个高电平而其它主器件在 SDA 上发送低电平。第一个在 SDA 上发送高电平的器件在竞争中失败并被要求放弃总线。

16.4 时钟低电平扩展

SMBus 提供一种与 I^2C 类似的同步机制，允许不同速度的器件在总线上共存。为了使低速从器件能与高速主器件通信，在传输期间使用低电平扩展。从器件可以保持 SCL 为低电平以扩展时钟低电平时间，实际上是降低了串行时钟频率。

16.5 超时

16.5.1 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条件。为了解决这一问题，SMBus 协议规定：参加一次数据传输的器件必须检查时钟低电平时间，若超过 25ms 则认为是“超时”，检测到超时条件的器件必须在 10ms 以内对通信复位。

可以用 MCU 的一个通用定时器工作在 16 位自动重装载方式来监视 SCL 线的超时条件。定时器 3 就是为此目的特别设计的。（有关定时器 3 的详细信息见 19.3 节）

16.5.2 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，则可认为总线处于空闲状态。当 SMB0CN 寄存器中的 FTE 位被置 1 时，SMB0CR 寄存器用于检测这一条件。

16.6 SMBus 特殊功能寄存器

对 SMBus 串行接口的访问和控制是通过 5 个特殊功能寄存器来实现的：控制寄存器 SMB0CN、时钟速率寄存器 SMB0CR、地址寄存器 SMB0ADR、数据寄存器 SMB0DAT 和状态寄存器 SMB0STA。系统器件可以有一个或多个 SMBus 串行接口。下面将对这 5 个与 SMBus 接口操作有关的特殊功能寄存器进行详细说明。

16.6.1 控制寄存器

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

设置 ENSMB 标志为逻辑 1 将允许 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。然而不应使用 ENSMB 从总线临时移出一个器件，因为总线状态信息将丢失。应使用确认标志（AA）从总线临时移出器件（见下面对 AA 标志的说明）。

设置起始标志（STA，SMB0CN.5）为逻辑 1 将 SMBus 设置为主方式。如果总线空闲，SMBus 硬件将产生一个起始条件。如果总线不空闲，SMBus 硬件等待停止条件释放总线，然后根据 SMB0CR 的值在经过 5 微秒的延时后产生一个起始条件。（根据 SMBus 协议，如果总线处于等待状态的时间超过 50 微秒而没有检测到停止条件，SMBus 接口可以认为总线是空闲的。）如果 STA 被设置为逻辑 1，而此时 SMBus 处于主方式并且已经发送了一个或多个字节，则将产生一个重复起始条件。为保证操作正确，应在对 STA 位置 1 之前，将 STO 标志清 0。

当停止标志（STO，SMB0CN.4）被设置为逻辑 1，而此时 SMBus 接口处于主方式，则硬件将在 SMBus 上产生一个停止条件。在从方式，STO 标志可以用于恢复一个错误条件。在这种情况下，SMBus 上不产生停止条件，但 SMBus 硬件的表现就象是收到了一个停止条件，并进入“未寻址”的从接收器状态。当检测到总线上的停止条件时，SMBus 硬件自动将 STO 标志清为逻辑 0。

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。当 SI 标志被设置为逻辑 1 时，串行时钟的低电平时间将被延长，串行传输暂时停止。

确认标志（AA，SMB0CN.2）用于在 SCL 线的确认周期中设置 SDA 线的电平。如果器件被寻址，设置 AA 标志为逻辑 1 将在确认周期发送一个确认位（SDA 上的低电平）。设置 AA 标志为逻辑 0 将在确认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。从器件自身地址或通用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 以允许识别其从地址。

设置 SMBus 空闲定时器允许位（FTE，SMB0CN.1）为逻辑 1 将允许 SMBus 空闲超时功能。如果 SCL 和 SDA 保持高电平的时间等于在 SMBus 时钟速率寄存器（图 16.5）中给定的 SMBus 空闲超时间隔，则可认为总线是空闲的。总线空闲时间应大于 50 微秒。

当 SCL 线为低电平且定时器 3 被允许时，设置 SMBus 超时允许位为逻辑 1 将允许定时器 3 向上计数。当定时器 3 溢出时将产生一个定时器 3 中断，通知 CPU 发生了 SMBus SCL 低电平超时。

图 16.4 SMB0CN：SMBus 控制寄存器

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
BUSY	ENSMB	STA	STO	SI	AA	FTE	TOE	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： BUSY：忙状态标志
0：SMBus 空闲
1：SMBus 忙

位 6： ENSMB：SMBus 允许
该位允许/禁止 SMBus 串行接口
0：禁止 SMBus
1：允许 SMBus

位 5： STA：SMBus 起始标志
0：不发送起始条件。
1：当作为主器件时，若总线空闲，则发送出一个起始条件。（如果总线不空闲，在收到停止条件后再发送起始条件。）如果 STA 被置 1，而此时已经发送或接收了一个或多个字节并且没有收到停止条件，则发送一个重复起始条件。

位 4： STO：SMBus 停止标志
0：不发送停止条件。
1：将 STO 置为逻辑 1 将发送一个停止条件。当收到停止条件时，硬件将 STO 清为逻辑 0。如果 STA 和 STO 都被置位，则发送一个停止条件后再发送一个起始条件。在从方式，置位 STO 标志将导致 SMBus 的行为象收到了停止条件一样。

位 3： SI：SMBus 串行中断标志
当 SMBus 进入 27 种状态之一时该位被硬件置位。（状态码 0xF8 不使 SI 置位。）当 SI 中断被允许时，该位置 1 将导致 CPU 转向 SMBus 中断服务程序。该位不能被硬件自动清 0，必须用软件清除。

位 2： AA：SMBus 确认标志
该位定义在 SCL 线应答周期内返回的应答类型。
0：在应答周期内返回“非确认”（SDA 线高电平）。
1：在应答周期内返回“确认”（SDA 线低电平）。

位 1： ETE：SMBus 空闲定时器允许位
0：无 SCL 高电平超时。
1：当 SCL 高电平时间超过由 SMB0CR 规定的极限值时发生超时。

位 0： TOE：SMBus 超时允许位
0：SCL 低电平未发生超时。
1：当 SCL 处于低电平的时间超过由定时器 3（如果被允许）定义的极限值时发生超时。

16.6.2 时钟频率寄存器

图 16.5　SMB0CR：SMBus 时钟频率寄存器

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：0xCF

位 7-0：　SMB0CR.[7:0]：SMBus 时钟频率预设值

SMB0CR 时钟频率寄存器用于控制主方式下串行时钟 SCL 的频率。存储在 SMB0CR 寄存器中的 8 位字预装一个专用的 8 位定时器。定时器向上计数，当计满回到 0x00 时，SCL 改变逻辑状态。SCL 时钟的周期由下式给定：

$$T_H = T_L = T_{SYSCLK} * [(256 - SMB0CR) + 2.5] \geq 5\mu s$$

使用与上式相同的 SMB0CR 值，总线空闲超时周期由下式给出：

$$T_{BFT} = T_{SYSCLK} * [10 * (256 - SMB0CR) + 1] \geq 50\mu s$$

SMBus 时钟频率寄存器的值可以从 0x00 到 0xFE，但不能使用 0xFF。

16.6.3 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 被设置为逻辑 1 时数据是稳定的。当 SI 标志置位时，软件可以安全地读/写数据寄存器。当 SMBus 被允许且 SI 标志被清为逻辑 0 时软件不应访问 SMB0DAT 寄存器，因为硬件可能正在对该寄存器中的数据字节进行移入或移出操作。

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

图 16.6 SMB0DAT：SMBus 数据寄存器

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

位 7-0： SMB0DAT：SMBus 数据
SMB0DAT 寄存器保存要发送到 SMBus 串行接口上的一个数据字节，或刚从 SMBus 串行接口接收到的一个字节。一旦 SI 串行中断标志（SMB0CN.3）被置为逻辑 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。

16.6.4 地址寄存器

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

图 16.7 SMB0ADR：SMBus 地址寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SLV6	SLV5	SLV4	SLV3	SLV2	SLV1	SLV0	GC	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xC3

位 7-1： SLV6-SLV0：SMBus 从地址
这些位用于存放 7 位从地址，当器件工作在从发送器或从接收器方式时，SMBus 将应答该地址。SLV6 是地址的最高位，对应从 SMBus 收到的地址字节的第一位。

位 0： GC：通用呼叫地址允许
该位用于允许通用呼叫地址（0x00）识别。
0：忽略通用呼叫地址。
1：识别通用呼叫地址。

16.6.5 状态寄存器

状态寄存器 SMB0STA 保存一个 8 位的状态码，用于指示 SMBus 的当前状态。共有 28 个可能的 SMBus 状态，每个状态对应一个唯一的状态码。状态码的高 5 位是可变的，而一个有效状态码的低 3 位固定为 0（当 SI=1 时）。因此所有有效的状态码都是 8 的整数倍。这使我们可以很容易地在软件中用状态码作为转移到正确的中断服务程序的索引（允许用 8 字节的代码对状态服务或转到更长的中断服务程序）。

对于用户软件而言，SMB0STA 的内容只在 SI 标志为 1 时有定义。软件不应向 SMB0STA 寄存器写入。如果写入，则会产生不确定的结果。表 16.1 列出了 28 个 SMBus 状态和对应的状态码。

图 16.8　SMB0STA：SMBus 状态寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
STA7	STA6	STA5	STA4	STA3	STA2	STA1	STA0	11111000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xC1

位 7-3：　STA7-STA3：SMBus 状态代码
这些位包含了 SMBus 状态代码。共有 28 个可能的状态码。每个状态码对应一个 SMBus 状态。在 SI 标志（SMB0CN.3）置位时，SMB0STA 中的状态码有效。当 SI 标志为逻辑 0 时，SMB0STA 中的内容不确定。任何时候向 SMB0STA 寄存器写入将导致不确定的结果。

位 2-0：　STA2-STA0：当 SI 标志位为逻辑 1 时，这三个 SMB0STA 最低位的读出值总是为逻辑 0。

表 16.1. SMBus 状态码

状态码 (SMB0STA)	工作方式	SMBus 状态
0x00	所有方式	总线错误（非法起始条件、停止条件、…）
0x08	主发送器/接收器	起始条件已发出
0x10	主发送器/接收器	重复起始条件已发出
0x18	主发送器	从地址+W 已发出。收到 ACK。
0x20	主发送器	从地址+W 已发出。收到 NACK。
0x28	主发送器	数据字节已发出。收到 ACK。
0x30	主发送器	数据字节已发出。收到 NACK。
0x38	主发送器	竞争失败
0x40	主接收器	从地址+R 已发出。收到 ACK。
0x48	主接收器	从地址+R 已发出。收到 NACK。
0x50	主接收器	收到数据字节。ACK 已发出
0x58	主接收器	收到数据字节。NACK 已发出
0x60	从接收器	收到自己的从地址+W。ACK 已发出。
0x68	从接收器	在作为主器件发送 SLA+R/W 时竞争失败。收到自身地址+W。ACK 已发出。
0x70	从接收器	收到通用呼叫地址。ACK 已发出。
0x78	从接收器	在作为主器件发送 SLA+R/W 时竞争失败。收到通用呼叫地址+W。ACK 已发出。
0x80	从接收器	收到自己的从地址+W。收到数据字节。ACK 已发出。
0x88	从接收器	收到自己的从地址+W。收到数据字节。NACK 已发出。
0x90	从接收器	收到通用呼叫地址。收到数据字节。ACK 已发出。
0x98	从接收器	收到通用呼叫地址。收到数据字节。NACK 已发出。
0xA0	从接收器	收到停止条件或重复起始条件。
0xA8	从发送器	收到自己的从地址+R。ACK 已发出。
0xB0	从发送器	在作为主器件发送 SLA+R/W 时竞争失败。收到自身从地址+R。ACK 已发出。
0xB8	从发送器	数据字节已发送。收到 ACK。
0xC0	从发送器	数据字节已发送。收到 NACK。
0xC8	从发送器	最后字节已发送（AA=0）。收到 ACK。
0xD0	从发送器/接收器	SCL 时钟高电平定时器超时（根据 SMB0CR）
0xF8	所有方式	空闲

17. 串行外设接口总线

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

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

图 17.1　SPI 原理框图

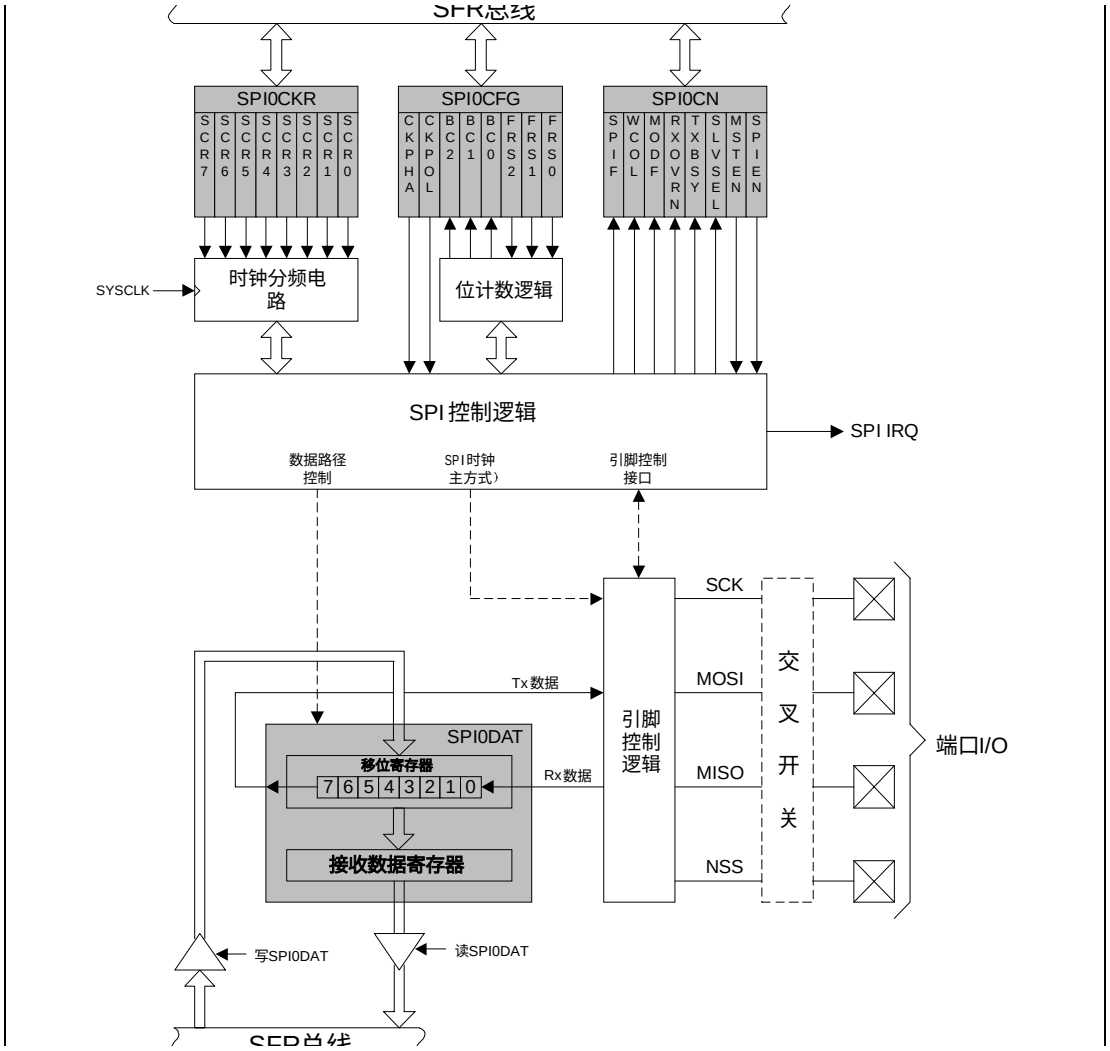
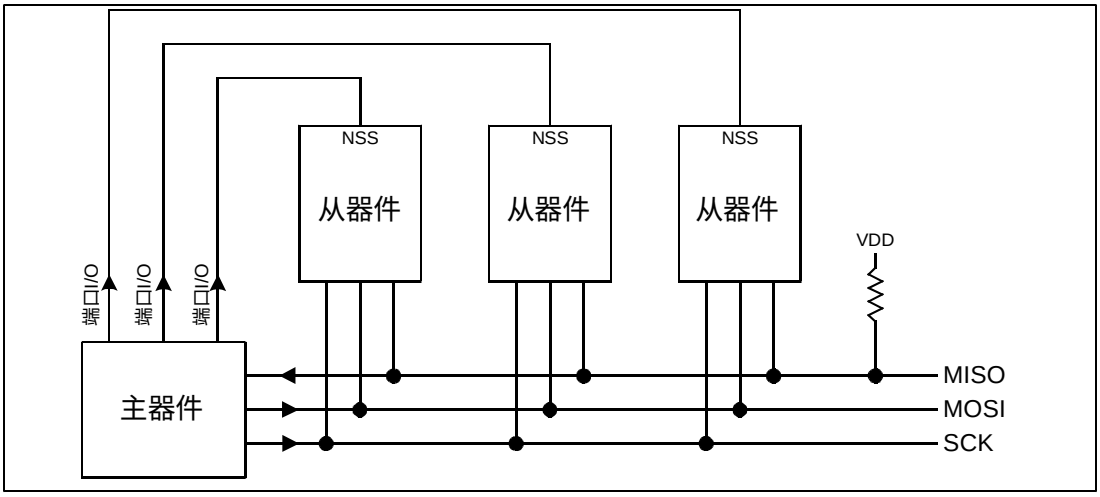


图 17.2　典型 SPI 互连



17.1 信号说明

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

17.1.1 主输出、从输入

主出从入（MOSI）信号是主器件的输出和从器件的输入。用于从主器件到从器件的串行数据传输。数据传输时最高位在先。

17.1.2 主输入、从输出

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

17.1.3 串行时钟

串行时钟（SCK）信号是主器件的输出和从器件的输入。用于同步主器件和从器件之间在 MOSI 和 MISO 线上的串行数据传输。

17.1.4 从选择

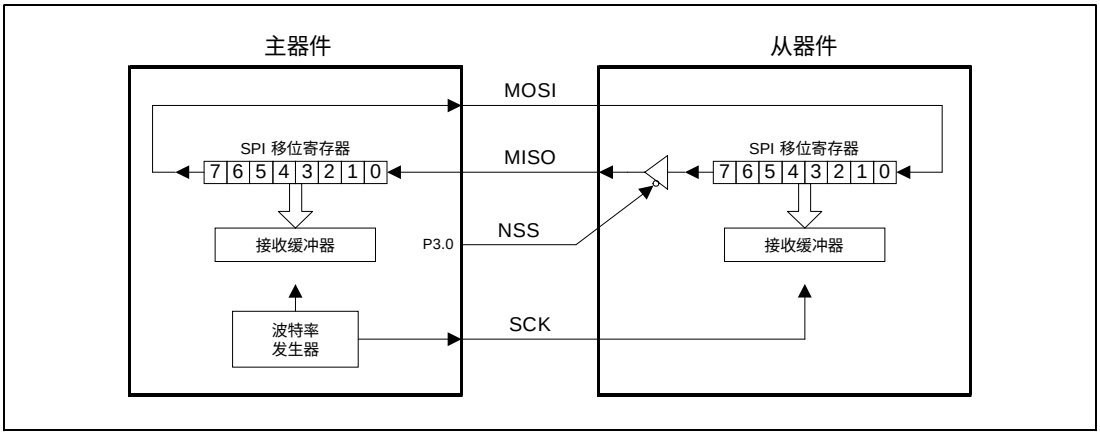
从选择（NSS）信号是一个输入信号，主器件用它来选择处于从方式的 SPI 模块，在主方式时用于禁止 SPI 模块。当处于从方式时，它被拉为低电平以启动一次数据传输并在传输期间保持低电平。

17.2 操作

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

在全双工操作中，SPI 主器件在 MOSI 线上向从器件发送数据，被寻址的从器件可以同时在 MISO 线上向主器件发送其移位寄存器中的内容。所接收到的来自从器件的数据替换主器件数据寄存器中的数据。因此，SPIF 标志既作为发送完成标志又作为接收数据准备好标志。两个方向上的数据传输由主器件产生的串行时钟同步。图 17.3 例示了一个 SPI 主器件和一个 SPI 从器件的全双工操作。

图 17.3　全双工操作



SPI 数据寄存器对读操作是双缓冲的，但写操作时不是。如果在一次数据传输期间试图写 SPI0DAT，则 WCOL 标志（SPI0CN.6）将被设置为逻辑 1，写操作被忽略，当前的数据传输不受影响。系统控制器读 SPI 数据寄存器时，实际上是读接收缓冲器。如果当前传输的最后一位已经移入 SPI 移位寄存器，而接收缓冲器中仍保存着前一次传输未被读取的数据，则发生接收溢出，RXOVRN 标志（SPI0CN.4）被设置为逻辑 1。新数据不传送到接受缓冲器，允许前面接收的数据等待读取，引起溢出的数据字节丢失。

当 SPI 被允许而未被配置为主器件时，它将作为从器件工作。另一个 SPI 主器件通过将 NSS 信号驱动为低电平启动一次数据传输。主器件用其串行时钟将移位寄存器中的数据移出到 MOSI 引脚。在一次数据传输结束后（当 NSS 信号变为高电平时），SPIF 标志被设置为逻辑 1。从器件可以通过写 SPI 数据寄存器来为下一次数据传输装载它的移位寄存器。从器件必须在主器件开始下一次数据传输之前至少一个 SPI 串行时钟周期写数据寄存器。否则，已经位于从器件移位寄存器中的数据字节将被传输。

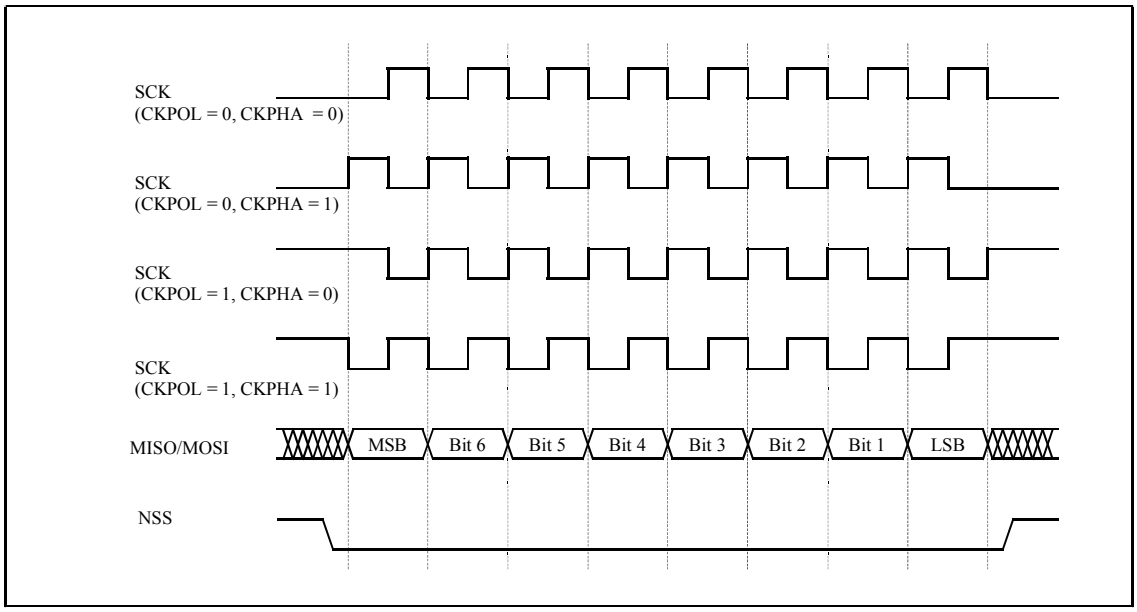
多个主器件可以共存于同一总线。当 SPI 被配置为主器件（MSTEN=1）而其从选择信号 NSS 被拉为低电平时，方式错误标志（MODF，SPI0CN.5）被设置为逻辑 1。当方式错误标志被置 1 时，SPI 控制寄存器中的 MSTEN 和 SPIEN 位被硬件清除，从而将 SPI 模块置于“离线”状态。在一个多主环境，系统控制器应检查 SLVSEL 标志（SPI0CN.2）,以保证在置‘1’MSTEN 位和启动一次数据传输之前总线是空闲的。

17.3 串行时钟时序

如图 17.4 所示，使用 SPI 配置寄存器（SPI0CFG）中的时钟控制选择位可以在串行时钟相位和极性的 4 种组合中选择其一。CKPHA 位（SPI0CFG.7）选择两种时钟相位（用于锁存数据的边沿）中的一种。CKPOL 位（SPI0CFG.6）在高电平有效和低电平有效的时钟之间选择。主器件和从器件必须被配置为使用相同的时钟相位和极性。注：在改变时钟相位和极性期间应禁止 SPI（通过清除 SPIEN 位，SPI0CN.0）。

图 17.7 所示的 SPI 时钟频率寄存器（SPI0CKR）控制主方式的串行时钟频率。当工作于从方式时该寄存器被忽略。

图 17.4　数据/时钟时序图



17.4 SPI 特殊功能寄存器

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

图 17.5 SPI0CFG：SPI 配置寄存器

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

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

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

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

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

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

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

图 17.6　SPI0CN：SPI 控制寄存器

R/W	R/W	R/W	R/W	R	R	R/W	R/W	复位值
SPIF	WCOL	MODF	RXOVRN	TXBSY	SLVSEL	MSTEN	SPIEN	00000000
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址: 0xF8

位 7：　SPIF：SPI 中断标志
该位在传输结束后被硬件置为逻辑 1。如果中断被允许，置 1 该位将会使 CPU 转到 SPI0 中断处理服务程序。该位不能被硬件自动清 0，必须用软件清 0。

位 6：　WCOL：写冲突标志
该位由硬件置为逻辑 1（并产生一个 SPI 中断），表示数据传送期间对 SPI 数据寄存器进行了写操作。该位用软件清 0。

位 5：　MODF：方式错误标志
当检测到主方式冲突（NSS 为低电平，MSTEN＝1）时，该位由硬件置为逻辑 1（并产生一个 SPI 中断）。该位不能被硬件自动清 0，必须用软件清 0。

位 4：　RXOVRN：接收覆盖标志
当前传输的最后一位已经移入 SPI 移位寄存器，而接收缓冲器中仍保存着前一次传输未被读取的数据时该位由硬件置为逻辑 1（并产生一个 SPI 中断）。该位不会被硬件自动清 0，必须用软件清 0。

位 3：　TXBSY：发送忙标志
当一个主方式传输正在进行时，该位被硬件置为逻辑 1。在传输结束后由硬件清 0。

位 2：　SLVSEL：从选择标志
该位在 NSS 引脚为低电平时置 1，说明它被允许为从方式。它在 NSS 变为高电平时清 0（从方式被禁止）

位 1：　MSTEN：主方式允许位
0：禁止主方式。以从方式操作。
1：允许主方式。以主方式操作。

位 0：　SPIEN：SPI 允许位
该位允许 / 禁止 SPI。
0：禁止 SPI
1：允许 SPI

图 17.7　SPI0CKR：SPI 时钟频率寄存器

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

位 7-0：　SCR7-SCR0：SPI 时钟频率

当 SPI 模块被配置为主方式操作时，这些位决定 SCK 输出的频率。SCK 时钟频率是从系统时钟分频得到的，由下式给出：

$$f_{SCK} = 0.5 * f_{SYSCLK} / (SPI0CKR + 1) \qquad (0 \leqslant SPI0CKR \leqslant 255)$$

图 17.8　SPI0DAT：SPI 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：0x9B

位 7-0：　SPI0DAT：SPI0 发送和接收数据。

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

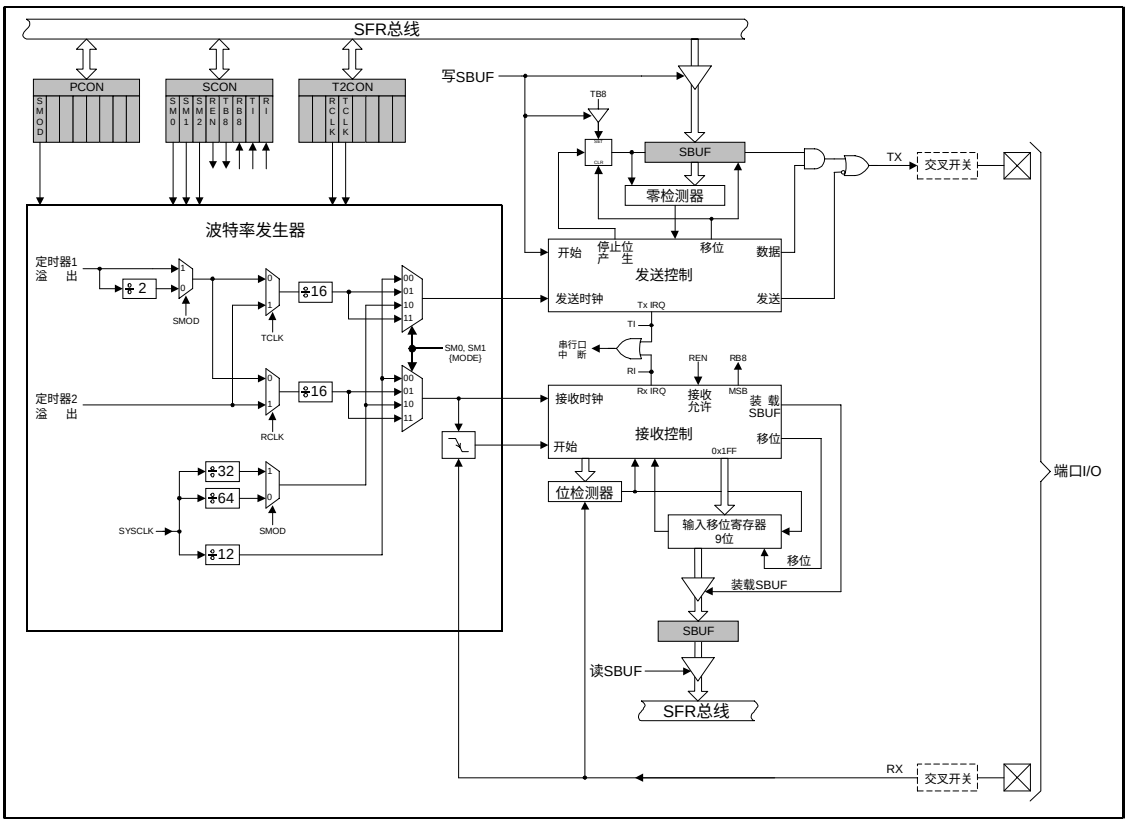
18. UART

UART 是一个能进行异步传输的串行口。UART 可以工作在全双工方式。在所有方式下，接收数据被放入一个保持寄存器。这就允许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。

UART 在特殊功能寄存器中有一个相关的串行控制寄存器（SCON）和一个串行数据缓冲器（SBUF）。用一个 SBUF 地址可以访问发送寄存器和接收寄存器。读操作访问接收寄存器，写操作访问发送寄存器。

如果被允许，UART 能产生中断。UART 有两个中断源：一个发送中断标志 TI（SCON.1）（数据字节发送结束时置位）和一个接收中断标志 RI（SCON.0）（接收完一个数据字节后置位）。当 UART 转向中断服务程序时硬件不清除 UART 中断标志，中断标志必须用软件清除。这就允许软件查询 UART 中断的原因（发送完成或接收完成）。

图 18.1　UART 原理框图



18.1 UART 工作方式

UART 提供四种工作方式（一种同步方式，三种异步方式），通过设置 SCON 寄存器中的配置位选择。这四种方式提供不同的波特率和通信协议。下面的表 18.1 概述了这四种方式。后面将详细说明。

表 18.1 UART 方式

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

18.1.1 方式 0：同步方式

方式 0 提供同步、半双工通信。在 RX 引脚上发送和接收数据。TX 引脚提供发送和接收的移位时钟。MCU 必须是主器件，因为它要为两个方向的数据传输产生移位时钟（见图 18.2 中的连接图）。

发送/接收的数据为 8 位，LSB 在先（见图 18.3 中的时序图）。执行一条写 SBUF 寄存器的指令时开始数据传输。在第 8 个位时间结束后发送中断标志 TI（SCON.1）置位。当接收允许位 REN（SCON.4）被设置为逻辑 1 并且接收中断标志 RI（SCON.0）被清 0 时开始数据接收。在第 8 位被移入后一个周期 RI 标志置位，接收过程停止，直到软件清除 RI 位。如果中断被允许，在 TI 或 RI 置位后，将发生一次中断。

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

图 18.2 UART 方式 0 连接

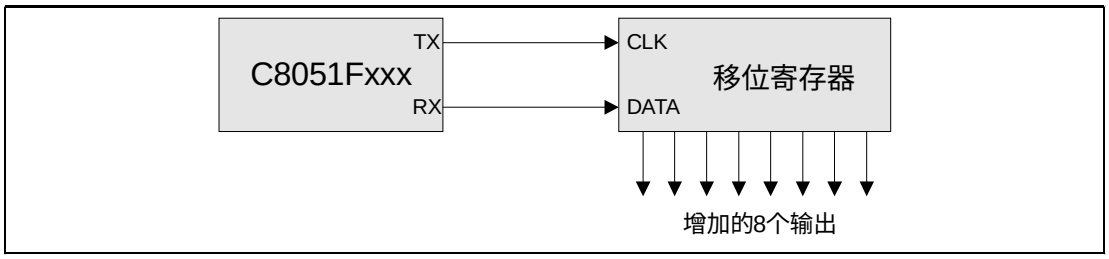
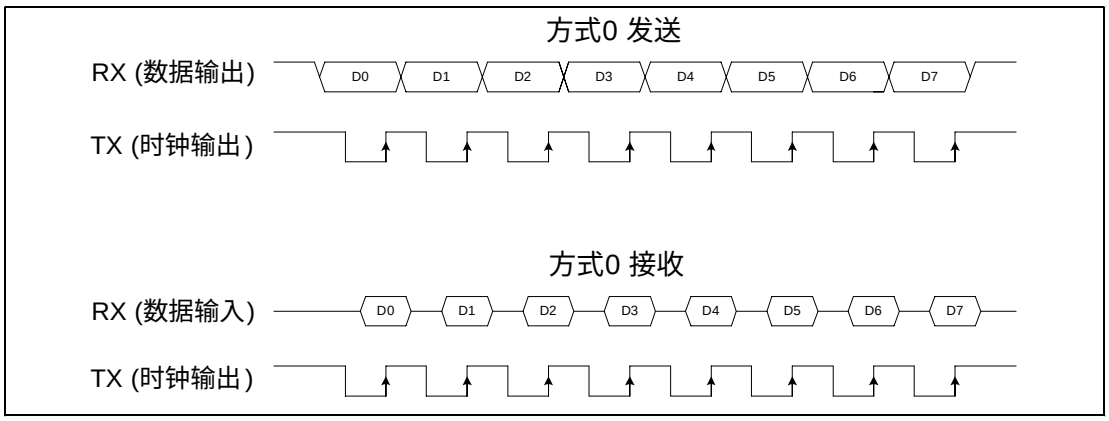


图 18.3 UART 方式 0 时序图



18.1.2 方式 1：8 位 UART，可变波特率

方式 1 提供标准的异步、全双工通信，每个数据字节共使用 10 位：一个起始位、8 个数据位（LSB 在先）和一个停止位。数据从 TX 引脚发送，在 RX 引脚接收。在接收时，8 个数据位存入 SBUF，停止位进入 RB8（SCON.2）。

当执行一条向 SBUF 寄存器写入一个字节的指令时开始数据发送。在发送结束时（停止位开始）发送中断标志 TI（SCON.1）置位。在接收允许位 REN（SCON.4）被设置为逻辑 1 时开始数据接收。收到停止位后如果满足下述条件则数据字节将被装入接收寄存器 SBUF：RI 为逻辑 0，并且如果 SM2 为逻辑 1 则停止位必须为 1。

如果这些条件满足，则 8 位数据被存入 SBUF，停止位被存入 RB8，RI 标志被置位。如果这些条件不满足，则不装入 SBUF 和 RB8，RI 标志也不被置 1。如果中断被允许，在 TI 或 RI 置位时将产生一个中断。

方式 1 的波特率是定时器溢出周期的函数。UART 可以使用定时器 1 或定时器 2 工作在自动重装载方式产生波特率。每次定时器发生溢出（从全 1（对定时器 1 为 0xFF，对定时器 2 为 0xFFFF）返回到 0）时向波特率电路发送一个时钟脉冲。该时钟的 16 分频即为波特率。

定时器 1 在用于波特率发生器时应被配置为自动重装载的 8 位计数器/定时器。系统时钟频率和 TH1 中的重载值的组合决定波特率：

方式 1 波特率=（2^{SMOD}/32）（SYSCLK）/12$^{(T1M-1)}$*（256-TH1））*

SMOD 位（PCON.7）选择是否将定时器溢出率除以 2。复位后 SMOD 位为逻辑 0，因此缺省情况下选择低速波特率。合理选择定时器 1 的时间基准可以进一步控制波特率产生。使用系统时钟除以 1 方式（在 CKCON 中置位 T1M）将使上式分母中的 12 变为 1。

如果使用定时器 2 作为波特率发生器，应将定时器配置为波特率发生器方式并将 RCLK 和/或 TCLK 设置为逻辑 1。置位 RCLK 和/或 TCLK 将自动禁止定时器 2 中断并使用系统时钟的二分频作为时间基准。如果需要使用不同的时间基准，将 C/T2 位设置为逻辑 1 将允许从外部输入引脚 T2 引入时间基准。系统时钟频率和保存在捕捉寄存器中的重载值的组合决定波特率：

方式 1 波特率= SYSCLK/ [32（65536- [RCAP2H:RCAP2L]）]*

其中[RCAP2H:RCAP2L]是捕捉寄存器中的数值。

图 18.4　UART 方式 1 时序图

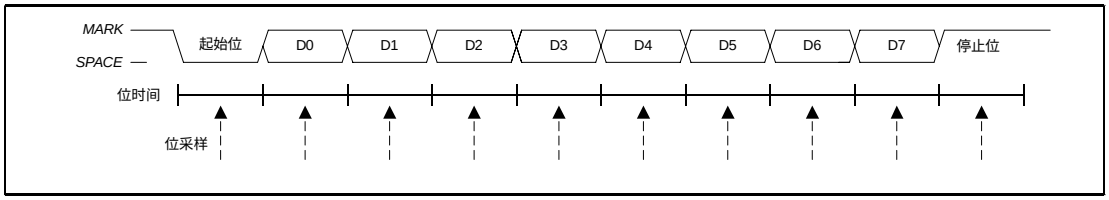
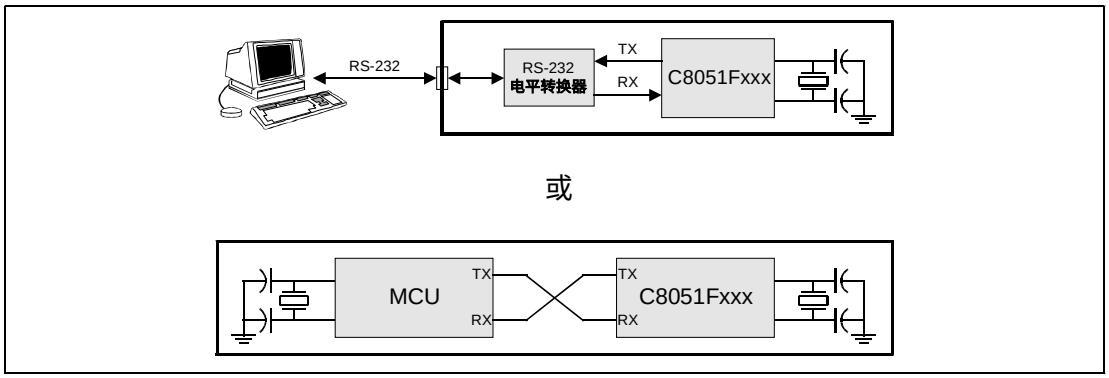


图 18.5　UART 方式 1、2 和 3 连接图



18.1.3 方式 2：9 位 UART，固定波特率

方式 2 提供异步、全双工通信，每个数据字节共使用 11 位：一个起始位、8 个数据位（LSB 在先）、一个可编程的第九位和一个停止位。在发送时，第九数据位由 TB8（SCON.3）中的值决定。它可以被赋值为 PSW 中的奇偶位 P，或用于多处理器通信。在接收时，第九数据位进入 RB8（SCON.2），停止位被忽略。

当执行一条向 SBUF 寄存器写入一个字节的指令时开始数据发送。在发送结束时（停止位开始）发送中断标志 TI（SCON.1）置位。在接收允许位 REN（SCON.4）被设置为逻辑 1 时开始数据接收。收到停止位后，如果满足下述条件则数据字节将被装入到接收寄存器 SBUF：RI 为逻辑 0，并且如果 SM2 为逻辑 1 则第九位必须为 1。

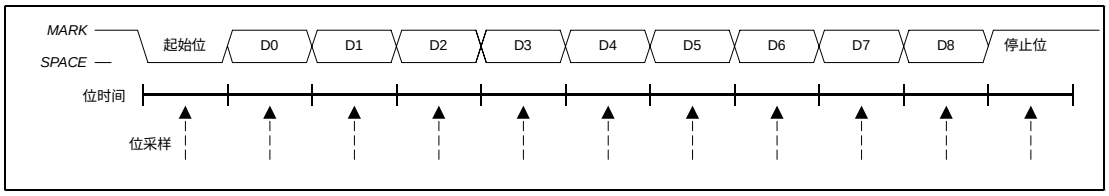
如果这些条件满足，则 8 位数据被存入 SBUF，第九位被存入 RB8，RI 标志被置位。如果这些条件不满足，则不装入 SBUF 和 RB8，RI 标志也不被置 1。如果中断被允许，在 TI 或 RI 置位时将产生一个中断。

方式 2 的波特率是系统时钟频率的直接函数：

方式 2 波特率$=2^{SMOD}*$*（SYSCLK/64）*

SMOD 位（PCON.7）选择是将 SYSCLK 除以 32 还是 64。波特率为系统时钟频率的 1/32 或 1/64。复位后 SMOD 位为逻辑 0，因此缺省情况下选择低速波特率。

图 18.6　UART 方式 2 和 3 时序图



18.1.4 方式 3：9 位 UART，可变波特率

除了波特率可变之外，方式 3 与方式 2 完全一样。波特率的确定方式与方式 1 相同。方式 3 的数据字节共使用 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。与方式 1 一样用定时器 1 和定时器 2 产生波特率。简言之，方式 3 使用与方式 2 相同的协议，而波特率产生方法与方式 1 相同。

18.2 多机通信

方式 2 和方式 3 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 SM2 位（SCON.5）被置‘1’，则只有当接收到的第九位为逻辑 1（RB8=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 SM2 位以允许后面接收数据字节时产生中断。地址不匹配的从机仍保持其 SM2 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 SM2 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 18.7　UART 多机方式连接图

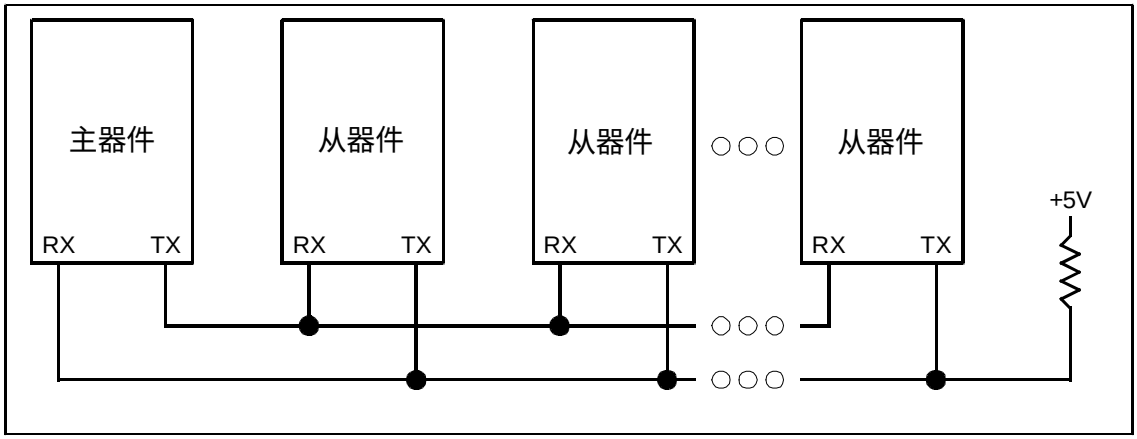


图 18.2. 产生标准波特率的振荡器频率

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

*假定 SMOD=1 且 TIM=1。

**括号里的数是实际波特率。

图 18.8 SBUF：串行（UART）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： SBUF.[7:0]：串行数据缓冲器位 7-0（MSB-LSB）
实际上是两个寄存器：一个发送和一个接收缓冲寄存器。当数据被送到 SBUF 时，它进入发送缓冲器等待串行发送。向 SBUF 写入一个字节即启动发送过程。当从 SBUF 读出数据时，数据来自接收缓冲器。

图 18.9　SCON：串行口控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00000000
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址：0x98

位 7-6：　SM0-SM1：串行口工作方式
这些位选择串行口的工作方式。

SM0	SM1	方式
0	0	方式 0：同步方式
0	1	方式 1：8 位 UART，可变波特率
1	0	方式 2：9 位 UART，固定波特率
1	1	方式 3：9 位 UART，可变波特率

位 5：　SM2：多处理器通信允许
该位的功能取决于串行口工作方式。
方式 0：无影响。
方式 1：检查有效停止位
　　0：停止位的逻辑电平被忽略。
　　1：只有当停止位为逻辑电平 1 时 RI 激活。
方式 2 和方式 3：多机通信允许
　　0：第九位的逻辑电平被忽略。
　　1：只有当第九位为逻辑电平 1 时 RI 才被置位并产生中断。

位 4：　REN：接收允许
该位允许/禁止 UART 接收。
0：UART 接收禁止
1：UART 接收允许

位 3：　TB8：第九发送位
该位的逻辑电平被赋值给方式 2 和 3 的第九发送位。在方式 0 和 1 中未用。跟据需要用软件置位或清 0。

位 2：　RB8：第九接收位
该位被赋值为方式 2 和 3 中第九接收位的逻辑电平。在方式 1，如果 SM2 为逻辑 0，则 RB8 被赋值为所接收到的停止位的逻辑电平。RB8 在方式 0 中未用。

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

位 0：　RI：接收中断标志
当 UART 接收到一个字节数据时（方式 0 时是在发送第 8 位后，其它方式在停止位后－例外情况见 SM2 位说明）该位被硬件置 1。在 UART 中断被允许时，置 1 该位将会使 CPU 转到 UART 中断服务程序。该位必须用软件手动清 0。

19. 定时器

每个 MCU 内部有 4 个计数器/定时器：其中三个 16 位计数器/定时器与标准 8051 中的计数器/定时器兼容，另一个 16 位定时器用于 ADC、SMBus 或作为通用定时器使用。这些计数器/定时器可以用于测量时间间隔，对外部事件计数或产生周期性的中断请求。定时器 0 和定时器 1 几乎完全相同，有四种工作方式。定时器 2 增加了一些定时器 0 和定时器 1 中所没有的功能。定时器 3 与定时器 2 类似，但没有捕捉或波特率发生器方式。

定时器 0 和定时器 1：	定时器 2：	定时器 3：
13 位计数/定时器	自动重装载的 16 位计数器/定时器	自动重装载的 16 位计数器/定时器
16 位计数器/定时器	带捕捉的 16 位计数/时器	
自动重装载的 8 位计数数/定时器	波特率发生器	
双 8 位计数器/定时器（只限于定时器 0）		

当工作在定时器方式时，计数数/定时器寄存器在每个时钟滴答加 1。时钟滴答为系统时钟除以 1 或系统时钟除以 12，由 CKCON 中的定时器时钟选择位（T2M-T0M）指定。每滴答为 12 个时钟的选项提供了与标准 8051 系列的兼容性。需要更快速定时器的应用可以使用每滴答 1 个时钟的选项。

当作为计数器使用时，在为 T0、T1 或 T2 所选择的引脚上出现负跳变时计数器/定时器寄存器加 1。对事件计数的最大频率可达到系统时钟频率的四分之一。输入信号不需要是周期性的，但在一个给定电平上的保持时间至少应为两个完整的系统时钟周期，以保证该电平能够被采样。

19.1 定时器 0 和定时器 1

对定时器 0 和定时器 1 的访问和控制是通过 SFR 实现的。每个计数器/定时器都是一个 16 位的寄存器，在被访问时以两个字节的形式出现：一个低字节（TL0 或 TL1）和一个高字节（TH0 或 TH1）。计数器/定时器控制（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。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 M1-M0 来选择工作方式。每个定时器都可以被独立配置。下面对每种工作方式进行详细说明。

19.1.1 方式 0：13 位计数器/定时器

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

TH0 寄存器保持 13 位计数器/定时器的高 8 位，TL0 在 TL0.4-TL0.0 位置保持 13 位计数器/定时器的低 5 位。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（如果被允许）。

C/T0 位（TMOD.2）选择计数器/定时器的时钟源。清除 C/T 将选择系统时钟作为定时器的输入。当 C/T0 被设置为逻辑 1 时，出现在所选输入引脚上的负跳变使定时器寄存器加 1。（有关选择和配置外部 I/O 引脚的详细信息见 15.1 节对端口 I/O 的介绍。）

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

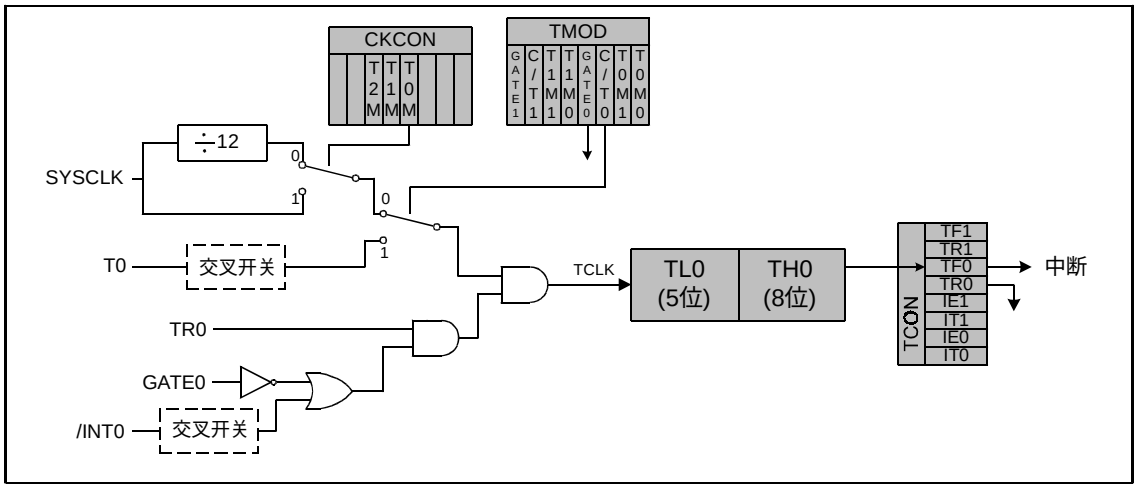
TR0	GATE0	/INT0	计数器/定时器
0	X	X	禁止
1	0	X	允许
1	1	0	禁止
1	1	1	允许

X=任意

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

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

图 19.1　T0 方式 0 原理框图



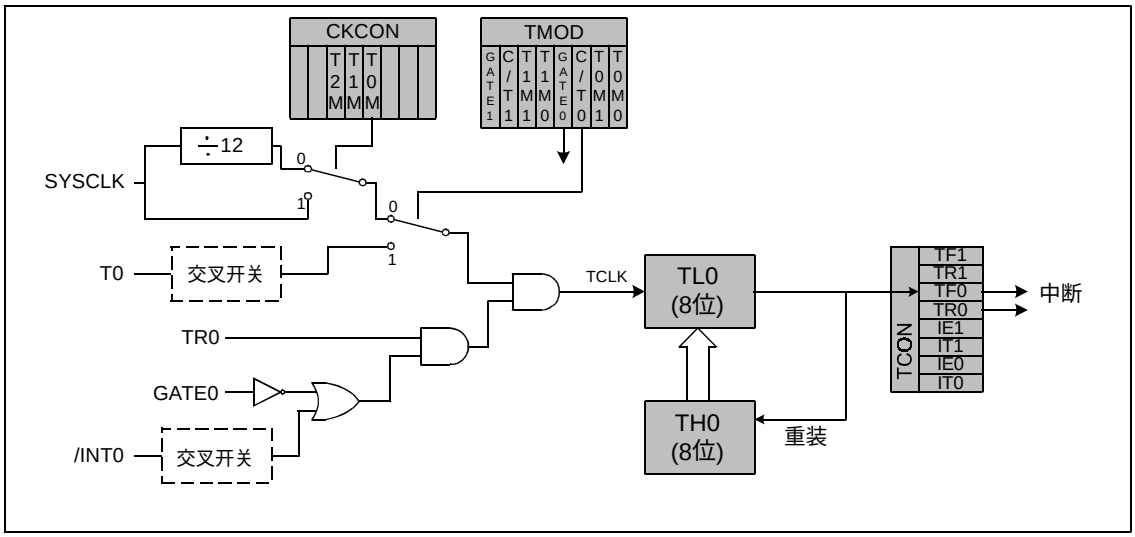
19.1.2 方式 1：16 位计数器/定时器

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

19.1.3 方式 2：8 位自动重装载的计数器/定时器

方式 2 将定时器 0 和定时器 1 配置为具有自动重新装入计数初值能力的 8 位计数器/定时器。TL0 保持计数值，而 TH0 保持重载值。当 TL0 中的计数值发生溢出（从全‘1’到 0x00）时，定时器溢出标志 TF0（TCON.5）被置位，TH0 中的重载值被重新装入到 TL0。如果中断被允许，在 TF0 被置位时将产生中断。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 时，定时器 0 和 1 的允许和配置方法与方式 0 一样。

图 19.2　T0 方式 2 原理框图



19.1.4 方式 3：两个 8 位计数器/定时器（只限于定时器 0）

在方式 3 时，定时器 0 和定时器 1 的表现不同。定时器 0 被配置两个独立的 8 位定时器/计数器，计数值在 TL0 和 TH0 中。在 TL0 中的计数器/定时器使用 TCON 和 TMOD 中定时器 0 的控制/状态位：TR0、C/T0、GATE0 和 TF0。它既可以使用系统时钟也可以使用一个外部输入信号作为时间基准。TH0 寄存器只能作为定时器使用，由系统时钟提供时间基准。TH0 使用定时器 1 的运行控制位 TR1。TH0 在发生溢出时将定时器 1 的溢出标志位 TF1 置‘1’，所以它控制定时器 1 的中断。

定时器 1 在方式 3 时停止运行。当定时器 0 工作于方式 3 时，可以通过将定时器 1 切换到方式 3 使其停止运行。在定时器 0 工作于方式 3 时，定时器 1 可以工作在方式 0、1 或 2，但不能用外部信号作为时钟，也不能设置 TF1 标志和产生中断。但是定时器 1 溢出可以用于产生波特率。有关将定时器 1 配置为波特率源的详细信息见第 18 章（UART）。

图 19.3　T0 方式 3 原理框图

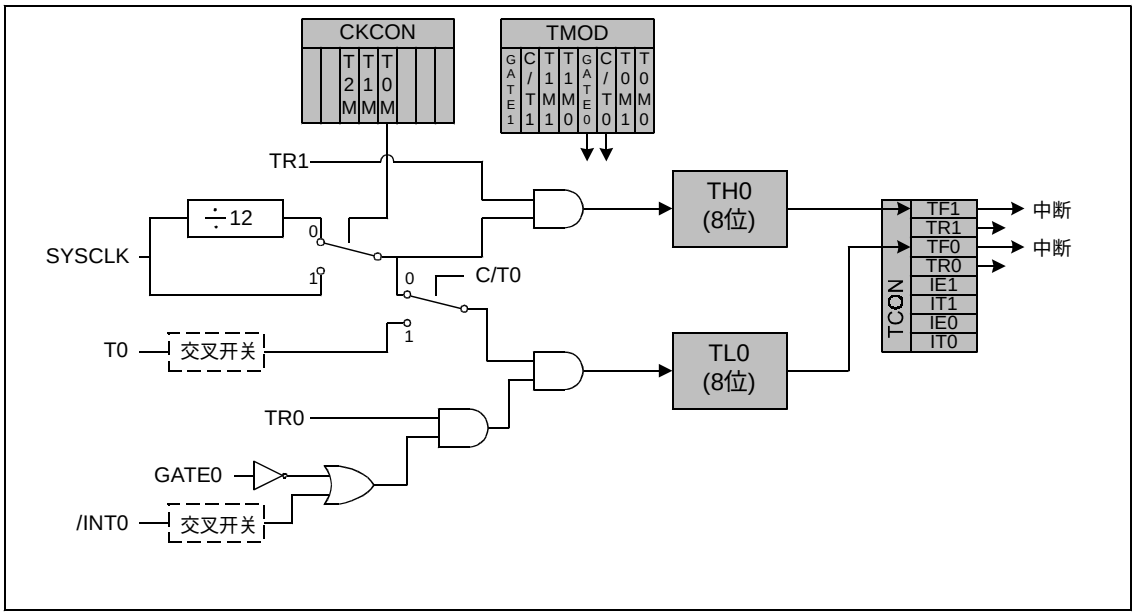


图 19.4　TCON：定时器控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址：0x88

位 7：　TF1：定时器 1 溢出标志
当定时器 1 溢出时由硬件置位。该位可以用软件清 0，但当 CPU 转向定时器 1 中断服务程序时该位被自动清 0。
0：未检测到定时器 1 溢出
1：定时器 1 发生溢出

位 6：　TR1：定时器 1 运行控制
0：定时器 1 禁止
1：定时器 1 允许

位 5：　TF0：定时器 0 溢出标志
当定时器 0 溢出时由硬件置位。该位可以用软件清 0，但当 CPU 转向定时器 0 中断服务程序时该位被自动清 0。
0：未检测到定时器 0 溢出
1：定时器 0 发生溢出

位 4：　TR0：定时器 0 运行控制
0：定时器 0 禁止
1：定时器 0 允许

位 3：　IE1：外部中断 1
当检测到一个由 IT1 定义的边沿/电平时，该标志由硬件置位。该位可以用软件清 0，但当 CPU 转向外部中断 1 中断服务程序时该位被自动清 0（如果 IT1=1）。当 IT1=0 时，该标志是/INT1 输入信号的逻辑电平取反。

位 2：　IT1：中断 1 类型选择
该位选择/INT1 信号检测下降沿中断还是检测低电平有效中断。
0：/INT1 为电平触发
1：/INT1 为边沿触发

位 1：　IE0：外部中断 0
当检测到一个由 IT0 定义的边沿/电平时，该标志由硬件置位。该位可以用软件清 0，但当 CPU 转向外部中断 0 中断服务程序时该位被自动清 0（如果 IT0=1）。当 IT0=0 时，该标志是/INT0 输入信号的逻辑电平取反。

位 0：　IT0：中断 0 类型选择
该位选择/INT0 信号检测下降沿中断还是检测低电平有效中断。
0：/INT0 为电平触发
1：/INT0 为边沿触发

图 19.5　TMOD：定时器方式寄存器

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

位 7：　GATE1：定时器 1 门控位
0：当 TR1=1 时定时器 1 被允许，与/INT1 的逻辑电平无关。
1：只有当 TR1=1 并且/INT1=逻辑 1 时定时器 1 才被允许。

位 6：　C/T1：计数器/定时器 1 功能选择。
0：定时器功能：定时器 1 由 T1M 位（CKCON.4）定义的时钟加 1。
1：计数器功能：定时器 1 由外部输入引脚（T1）的负跳变加 1。

位 5-4：　T1M1-T1M0：定时器 1 方式选择
这些位选择定时器 1 的工作方式。

T1M1	T1M0	方式
0	0	方式 0：13 位计数器/定时器
0	1	方式 1：16 位计数器/定时器
1	0	方式 2：自动重装载的 8 位计数器/定时器
1	1	方式 3：定时器 1 停止运行

位 3：　GATE0：定时器 0 门控位
0：当 TR0=1 时定时器 0 被允许，与/INT0 的逻辑电平无关。
1：只有当 TR0=1 并且/INT0=逻辑 1 时定时器 1 才被允许。

位 2：　C/T0：计数器/定时器 0 功能选择。
0：定时器功能：定时器 0 由 T0M 位（CKCON.3）定义的时钟加 1。
1：计数器功能：定时器 0 由外部输入引脚（T0）的负跳变加 1。

位 1-0：　T0M1-T0M0：定时器 0 方式选择
这些位选择定时器 0 的工作方式。

T0M1	T0M0	方式
0	0	方式 0：13 位计数器/定时器
0	1	方式 1：16 位计数器/定时器
1	0	方式 2：自动重装载的 8 位计数器/定时器
1	1	方式 3：双 8 位计数器/定时器

图 19.6　CKCON：时钟控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	T2M	T1M	T0M	保留	保留	保留	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0x8E

位 7-6：　未用。读=00b，写=忽略。

位 5：　T2M：定时器 2 时钟选择。
该位控制提供给定时器 2 的系统时钟的分频数。当定时器工作于波特率发生器方式或计数器方式（即 C/T2=1）时该位被忽略。
0：定时器 2 使用系统时钟的 12 分频
1：定时器 2 使用系统时钟

位 4：　T1M：定时器 1 时钟选择。
该位控制提供给定时器 1 的系统时钟的分频数。
0：定时器 1 使用系统时钟的 12 分频
1：定时器 1 使用系统时钟

位 3：　T0M：定时器 0 时钟选择。
该位控制提供给定时器 0 的系统时钟的分频数。
0：计数器/定时器使用系统时钟的 12 分频
1：计数器/定时器使用系统时钟

位 2-0：　保留。读=000b，必须写入 000。

图 19.7　TL0：定时器 0 低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0x8A

位 7-0：　TL0：定时器 0 低字节
TL0 寄存器是 16 位定时器 0 的低字节。

图 19.8　TL1：定时器 1 低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	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 的位字节。

图 19.9　TH0：定时器 0 高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0x8C

位 7-0：　TH0：定时器 0 高字节
TH0 寄存器是 16 位定时器 0 的高字节。

图 19.10　TH1：定时器 1 高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0x8D

位 7-0：　TH1：定时器 1 高字节
TH1 寄存器是 16 位定时器 1 的高字节。

19.2 定时器 2

定时器 2 是一个 16 位的计数器/定时器，由两个 8 位的 SFR 组成：TL2（低字节）和 TH2（高字节）。与定时器 0 和定时器 1 一样，它既可以使用系统时钟也可以使用一个外部输入引脚上的状态变化作为时钟源。计数器/定时器选择位 C/T2（T2CON.1）选择定时器 2 的时钟源。清除 C/T2 将选择系统时钟作为定时器的输入（由 CKCON 中的定时器时钟选择位 T2M 指定不分频或 12 分频）。当 C/T2 被置‘1’时，T2 输入引脚上的负跳变使计数器/定时器寄存器加 1。（有关选择和配置外部输入引脚的详细信息见第 14 章。）定时器 2 还可用于启动 ADC 数据转换。

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

RCLK	TCLK	CP／RL2	TR2	方式
0	0	1	1	带捕捉的 16 位计数器/定时器
0	0	0	1	自动重装载的 16 位计数器/定时器
0	1	X	1	TX 波特率发生器
1	0	X	1	RX 波特率发生器
1	1	X	1	TX 和 RX 波特率发生器
X	X	X	0	关闭

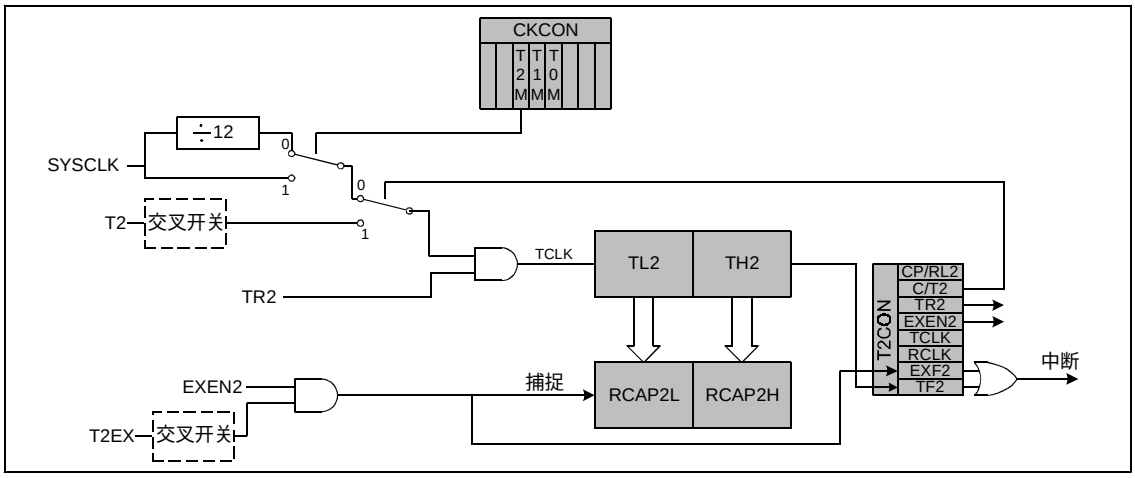
19.2.1 方式 0：带捕捉的 16 位计数器/定时器

在该方式，定时器 2 被作为具有捕捉能力的 16 位计数器/定时器使用。T2EX 引脚上的负跳变将定时器 2（TH2，TL2）中的 16 位计数值装入到捕捉寄存器（RCAP2H，RCAP2L）。

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

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

图 19.11　T2 方式 0 原理框图

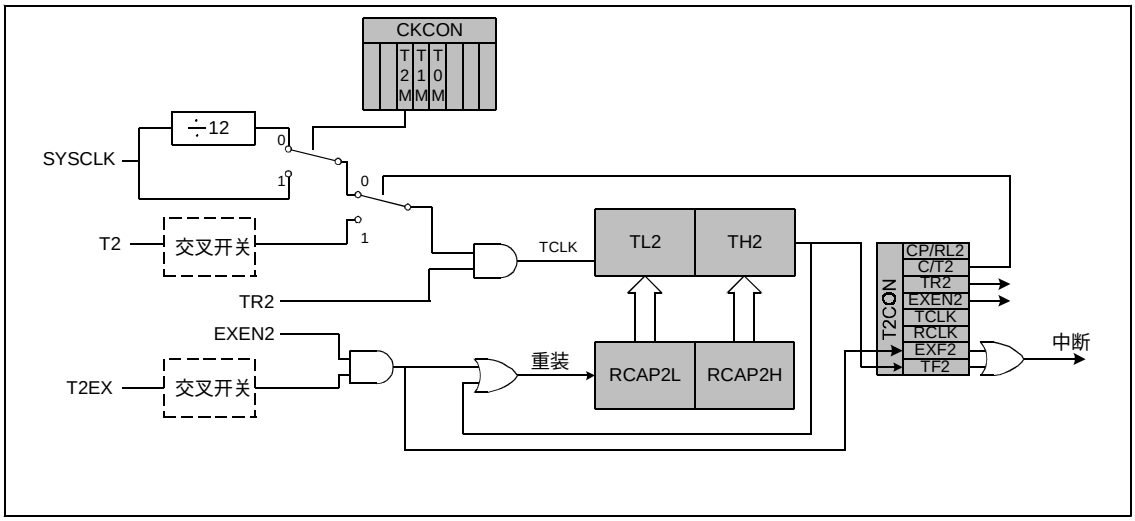


19.2.2 方式 1：自动重装载的 16 位计数器/定时器

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

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

图 19.12　T2 方式 1 原理框图



19.2.3 方式 2：波特率发生器

当串行口（UART）工作于方式 1 或方式 3 时，定时器 2 可以用作波特率发生器（有关 UART 工作方式的详细信息见 18.1 节）。定时器 2 的波特率发生器方式与自动重装载方式相似。在溢出时，两个捕捉寄存器（RCAP2H，RCAP2L）中的 16 位计数初值被自动装入到计数器/定时器寄存器。但是 TF2 溢出标志不置位，也不产生中断。溢出事件用作 UART 的移位时钟输入，定时器 2 溢出可以用于产生独立的发送和/或接收波特率。

设置 RCLK（T2CON.5）和/或 TCLK（T2CON.4）为逻辑 1 将选择波特率发生器方式。当 RCLK 或 TCLK 被设置为逻辑 1 时，定时器 2 工作在自动重装载方式，与 CP/RL2 位的状态无关。UART 工作在方式 1 或方式 3 的波特率由定时器 2 的溢出率决定：

波特率 = 定时器 2 溢出率/16

注意：在所有其它方式，定时器的时基信号为系统时钟或系统时钟的 12 分频，由 CKCON 中的 T2M 位选择。但是在波特率发生器方式时基信号为系统时钟的 2 分频，不能选择其它分频系数。如果需要不同的时基信号，可以通过将 C/T2 位设置为逻辑‘1’选择外部引脚 T2 上的输入作为时基。在这种情况下，UART 的波特率计算公式为：

波特率 = FCLK / [32 (65536 - [RCAP2H:RCAP2L])]*

其中，FCLK 为加在 T2 引脚上的信号的频率，而 RCAP2H:RCAP2L 为捕捉寄存器中的 16 位数值。

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

图 19.13　T2 方式 2 原理框图

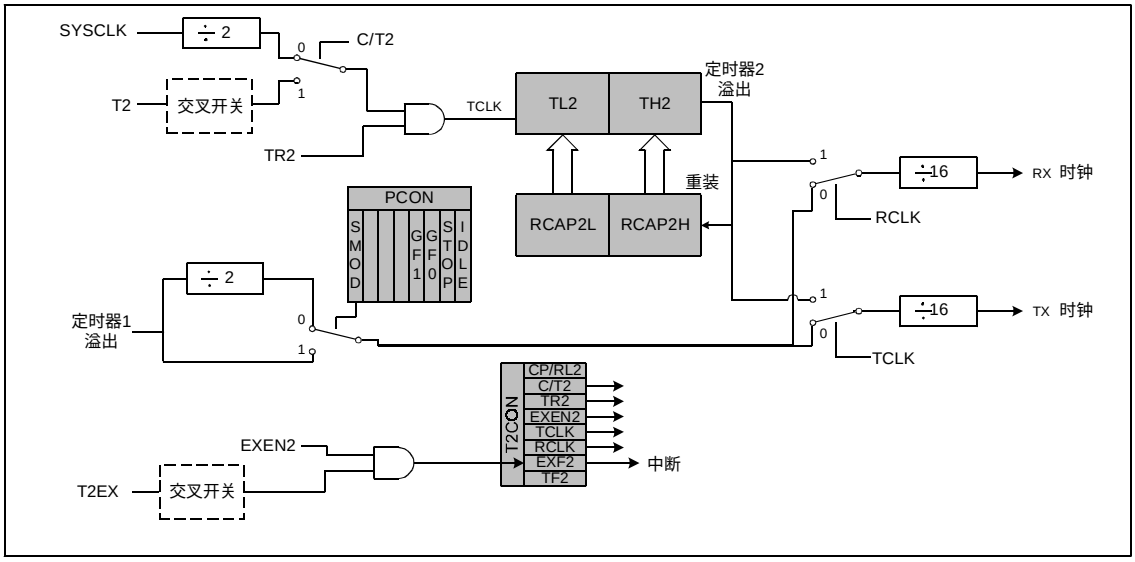


图 19.14　T2CON：定时器 2 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00000000
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址：0xC8

位 7：　TF2：定时器 2 溢出标志
当定时器 2 从 0xFFFF 到 0x0000 溢出时由硬件置位。当定时器 2 中断被允许时，该位置 1 导致 CPU 转向定时器 2 的中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。当 RCLK 和 / 或 TCLK 为逻辑 1 时，TF2 不会被置位。

位 6：　EXF2：定时器 2 外部标志
当 T2EX 输入引脚的负跳变导致发生捕捉或重载并且 EXEN2 为逻辑 1 时，该位由硬件置位。在定时器 2 中断被允许时，该位置‘1’使 CPU 转向定时器 2 的中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 5：　RCLK：接收时钟标志
选择 UART 工作在方式 1 或 3 时接收时钟使用的定时器。
0：定时器 1 溢出作为接收时钟。
1：定时器 2 溢出作为接收时钟。

位 4：　TCLK：发送时钟标志
选择 UART 工作在方式 1 或 3 时发送时钟使用的定时器。
0：定时器 1 溢出作为发送时钟。
1：定时器 2 溢出作为发送时钟。

位 3：　EXEN2：定时器 2 外部允许
当定时器 2 不是工作在波特率发生器方式时，允许 T2EX 上的负跳变触发捕捉或重载。
0：T2EX 上的负跳变被忽略
1：T2EX 上的负跳变导致一次捕捉或重载

位 2：　TR2：定时器 2 运行控制
该位允许/禁止定时器 2。
0：定时器 2 禁止
1：定时器 2 允许

位 1：　C/T2：计数器/定时器选择
0：定时器功能：定时器 2 由 T2M（CKCON.5）定义的时钟触发加 1。
1：计数器功能：定时器 2 由外部输入引脚（T2）的负跳变触发加 1。

位 0：　CP/RL2：捕捉/重载选择
该位选择定时器 2 为捕捉还是自动重装载方式。EXEN2 必须为逻辑 1 才能使 T2EX 上的负跳变能够被识别并用于触发捕捉和重载。若 RCLK 或 TCLK 被置位，该位将被忽略，定时器 2 将工作在自动重装载方式。
0：当定时器 2 溢出或 T2EX 上发生负跳变时将自动重装载（EXEN2=1）。
1：在 T2EX 发生负跳变时捕捉（EXEN2=1）。

图 19.15　RCAP2L：定时器 2 捕捉寄存器低字节

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

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

图 19.16 RCAP2H：定时器 2 捕捉寄存器高字节

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

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

图 19.17　TL2：定时器 2 低字节

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

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

图 19.18　TH2：定时器 2 高字节

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

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

19.3 定时器 3

定时器 3 是一个 16 位的计数器/定时器，由两个 8 位的 SFR 组成：TMR3L（低字节）和 TMR3H（高字节）。定时器 3 的输入为系统时钟（不分频或 12 分频，由定时器 3 控制寄存器 TMR3CN 中的定时器 3 时钟选择位 T3M 指定）。定时器 3 总是被配置为自动重装载方式定时器，重载值保存在 TMR3RLL（低字节）和 TMR3RLH（高字节）中。定时器 3 可用于启动 ADC 数据转换、SMBus 定时（见 16.5 节）或作为通用定时器使用。定时器 3 没有计数器方式。

图 19.19　定时器 3 原理框图

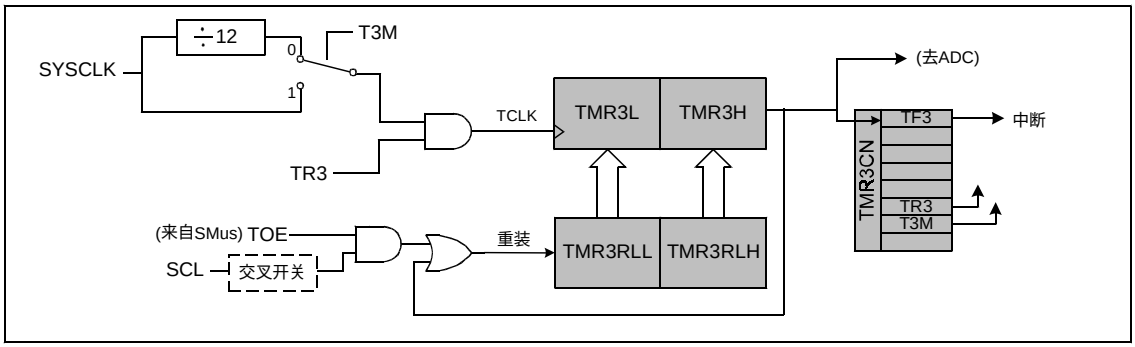


图 19.20　TMR3CN：定时器 3 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
TF3	-	-	-	-	TR3	T3M	-	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0x91

位 7：　TF3：定时器 3 溢出标志
当定时器 3 从 0xFFFF 到 0x0000 溢出时由硬件置位。当定时器 3 中断被允许时，该位置‘1’使 CPU 转向定时器 3 的中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 6-3：　未用。读=0000b，写=忽略。

位 2：　TR3：定时器 3 运行控制
该位允许/禁止定时器 3
0：定时器 3 禁止
1：定时器 3 允许

位 1：　T3M：定时器 3 时钟选择
该位控制提供给计数器/定时器 3 的系统时钟的分频数。
0：计数器/定时器 3 使用系统时钟的 12 分频。
1：计数器/定时器 3 使用系统时钟。

位 0：　未用。读=0，写=忽略。

图 19.21　TMR3RLL：定时器 3 重载寄存器低字节

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

位 7-0：　TMR3RLL：定时器 3 重载寄存器的低字节
定时器 3 被配置为自动重装载定时器。该寄存器保存重载值的低字节。

图 19.22　TMR3RLH：定时器 3 重载寄存器高字节

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

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

图 19.23　TMR3L：定时器 3 低字节

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

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

图 19.24　TMR3H：定时器 3 高字节

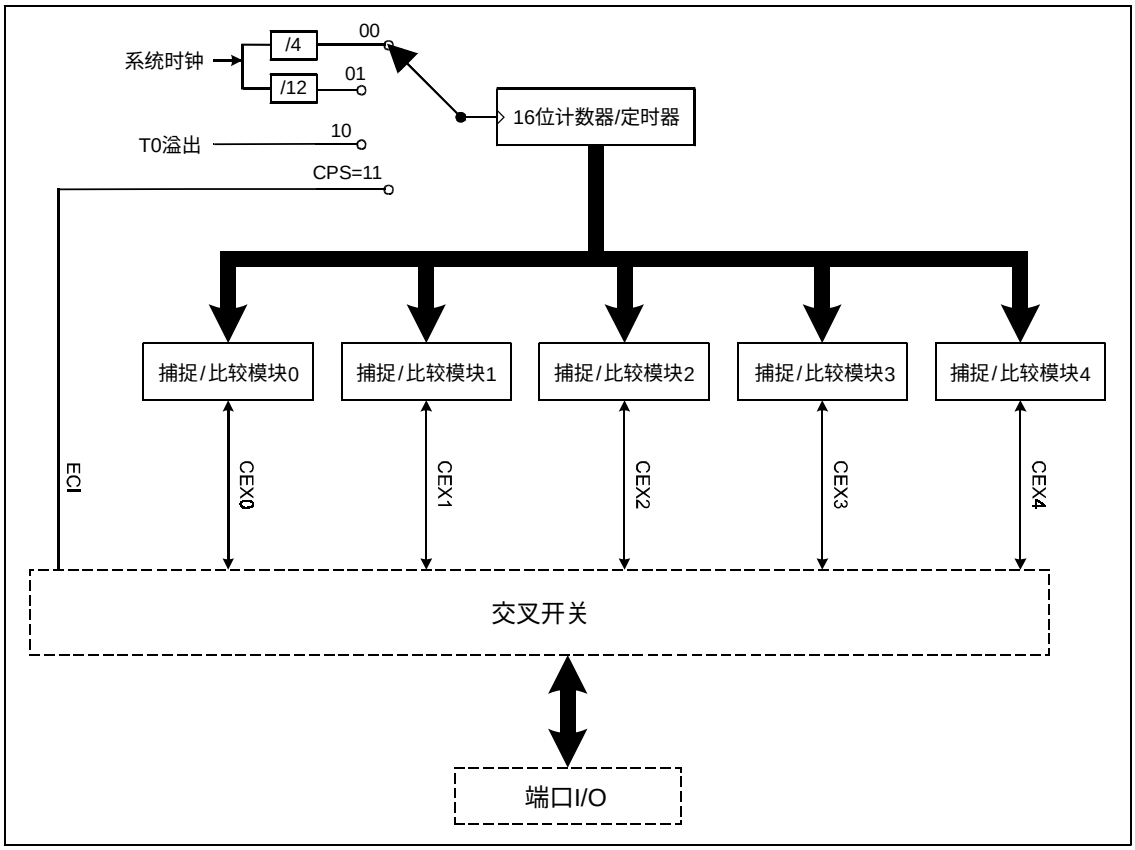
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0x95

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

20. 可编程计数器阵列

可编程计数器阵列（PCA）提供增强的定时器功能，与标准8051计数器/定时器相比，它需要较少的CPU干预。PCA包含一个专用的16位计数器/定时器和5个16位捕捉/比较模块。每个捕捉/比较模块有其自己的I/O线（CEXn）。当被允许时，I/O线通过交叉开关连到端口I/O（有关配置交叉开关的详细信息见15.1节）。计数器/定时器由一个可配置的时基信号驱动，可以在四个输入源中选择时基信号：系统时钟12分频、系统时钟4分频、定时器0溢出或ECI线上的外部时钟信号。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的基本原理框图示于图20.1。

图 20.1　PCA 原理框图



20.1 捕捉/比较模块

每个模块都可被配置为独立工作，有四种工作方式：边沿触发捕捉、软件定时器、高速输出和脉宽调制器。每个模块在CIP-51系统控制器中都有属于自己的特殊功能寄存器（SFR）。这些寄存器用于配置模块的工作方式和与模块交换数据。

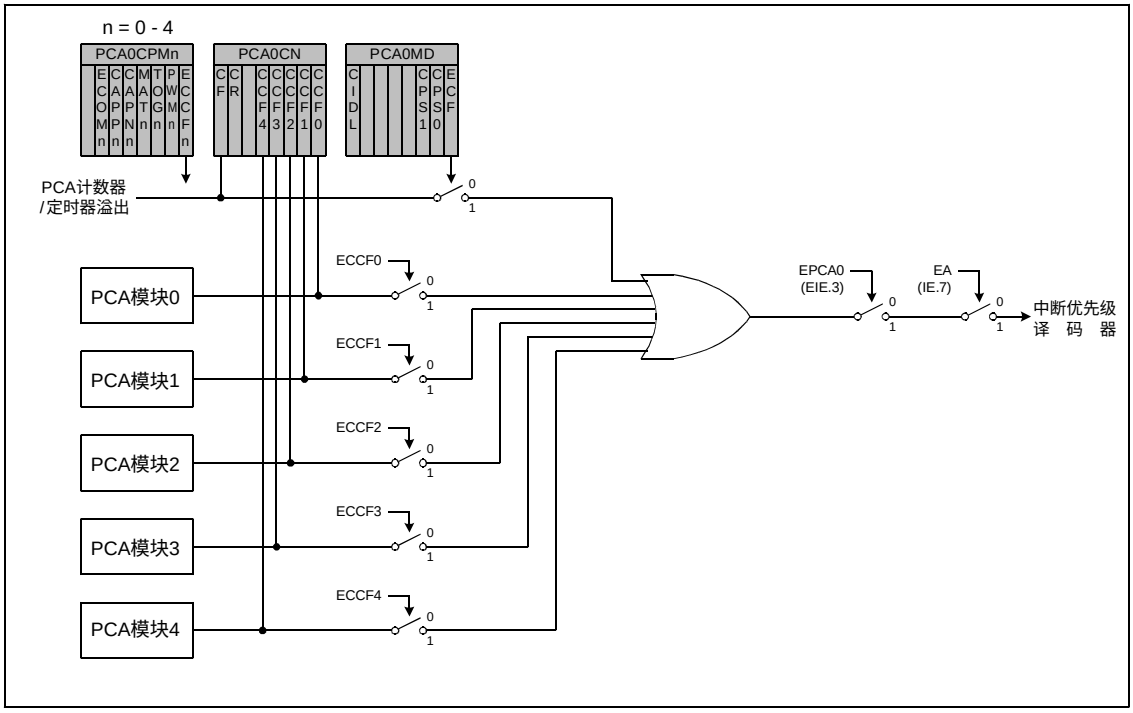
PCA0CPMn寄存器用于配置PCA捕捉/比较模块的工作方式，表20.1概述了模块工作在不同方式时该寄存器各位的设置情况。置‘1’ PCA0CPMn寄存器中的ECCFn位将允许模块的CCFn中断。注意：要使单独的CCFn中断得到响应，必须先整体允许PCA0中断。通过将EA位（IE.7）和EPCA0位（EIE1.3）设置为逻辑1来整体允许PCA0中断。PCA0中断配置的详细信息见图20.2。

表 20.1. PCA 捕捉/比较模块的 PCA0CPM 寄存器设置

ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	工作方式
X	1	0	0	0	0	X	用 CEXn 的正沿触发捕捉
X	0	1	0	0	0	X	用 CEXn 的负沿触发捕捉
X	1	1	0	0	0	X	用 CEXn 的电平变化触发捕捉
1	0	0	1	0	0	X	软件定时器
1	0	0	1	1	0	X	高速输出
1	0	0	X	0	1	0	脉冲宽度调制器

X = 忽略

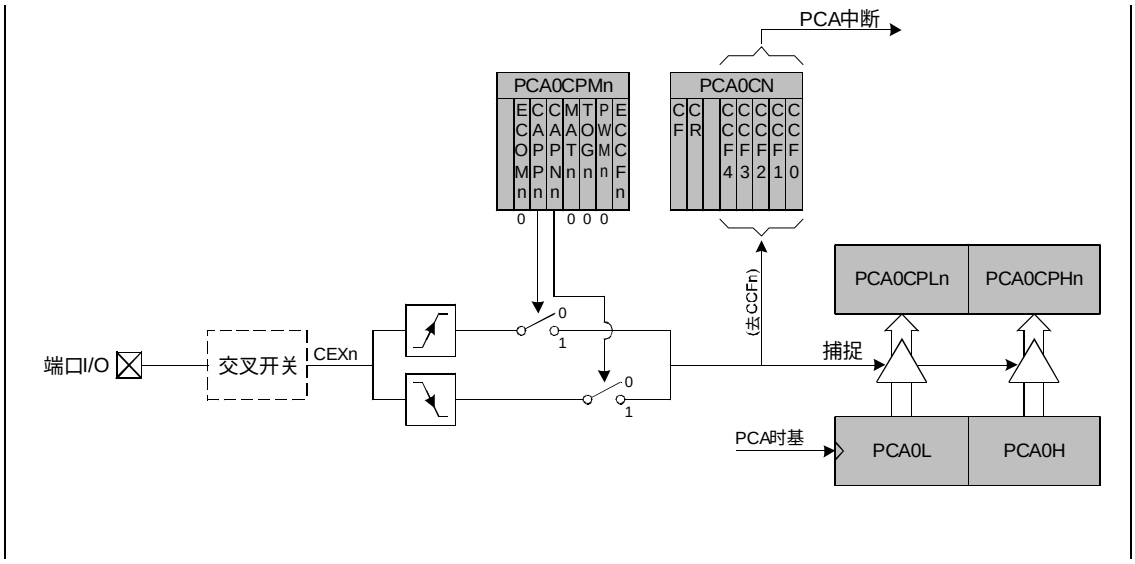
图 20.2 PCA 中断原理框图



20.1.1 边沿触发的捕捉方式

在该方式，CEXn引脚上出现的有效电平变化导致PCA捕捉PCA计数器/定时器的值并将其装入到对应模块的16位捕捉/比较寄存器（PCA0CPLn和PCA0CPHn）。PCA0CPMn寄存器中的CAPNn位用于选择触发捕捉的电平变化类型：低电平到高电平（正沿）、高电平到低电平（负沿）或任何变化（正沿或负沿）。当捕捉发生时，PCA0CN中的捕捉/比较标志（CCFn）被置为逻辑1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。

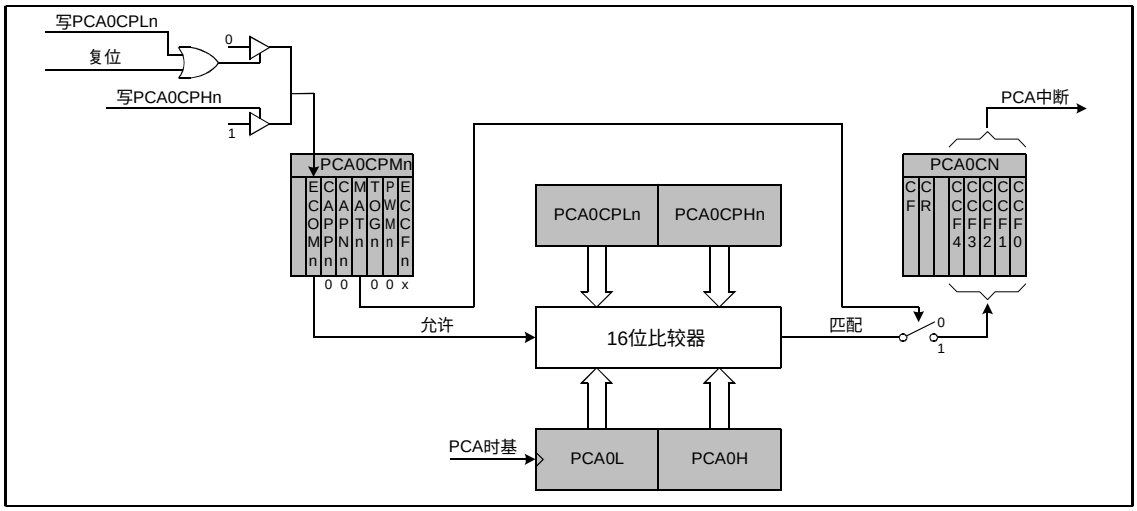
图 20.3　PCA 捕捉方式原理框图



20.1.2 软件定时器（比较）方式

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

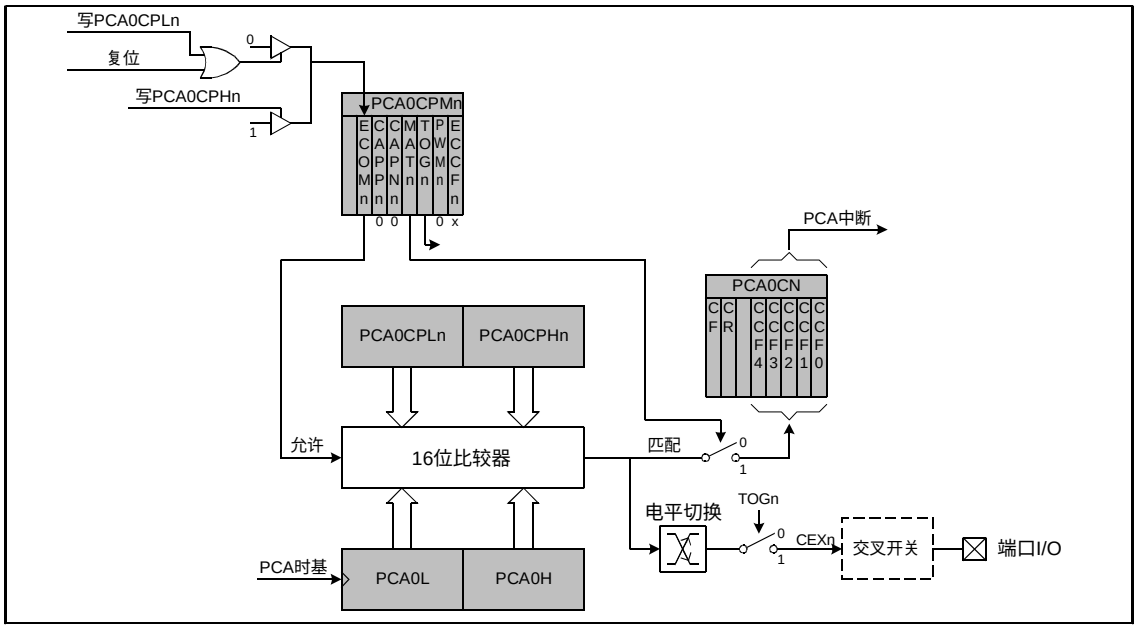
图 20.4　PCA 软件定时器方式原理框图



20.1.3 高速输出方式

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

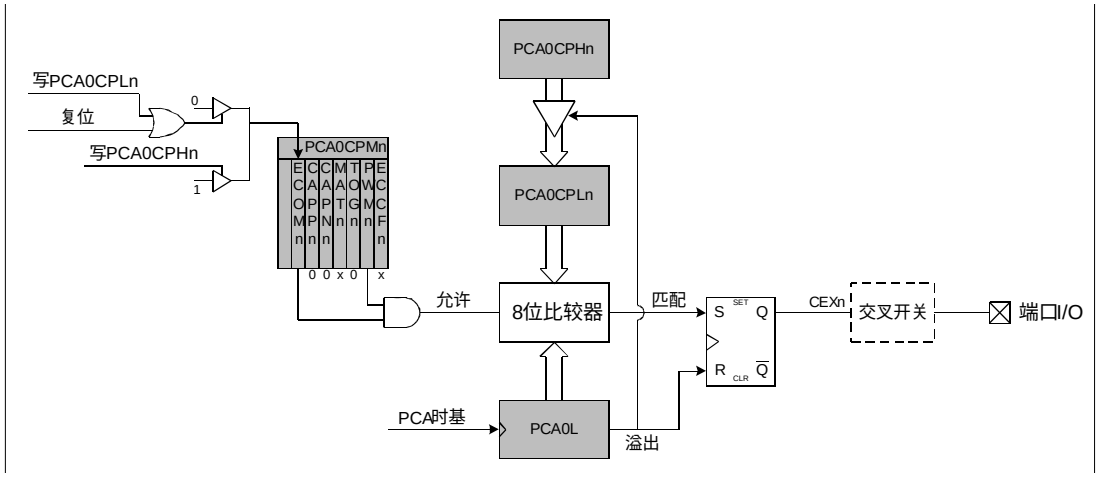
图 20.5　PCA 高速输出方式原理框图



20.1.4 脉宽调制器方式

所有模块都可以独立地用于在对应的CEXn引脚产生脉宽调制输出（PWM）。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PCA0CPLn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PCA0CPLn中的值相等时，CEXn引脚上的输出被置‘1’。当PCA0L中的计数值溢出时，CEXn输出被复位（见图20.6）。当计数器/定时器的低字节PCA0L溢出时（从0xFF到0x00），保存在PCA0CPHn中的值被自动装入PCA0CPLn，不需软件干预。最好不直接写PCA0CPLn，而是对PCA0CPHn写入，以避免数字比较器中发生瞬时错误。置‘1’ PCA0CPMn寄存器中的ECOMn和PWMn位将允许脉冲宽度调制器方式。

图 20.6　PCA 的 PWM 方式原理框图



20.2 PCA 计数器定时器

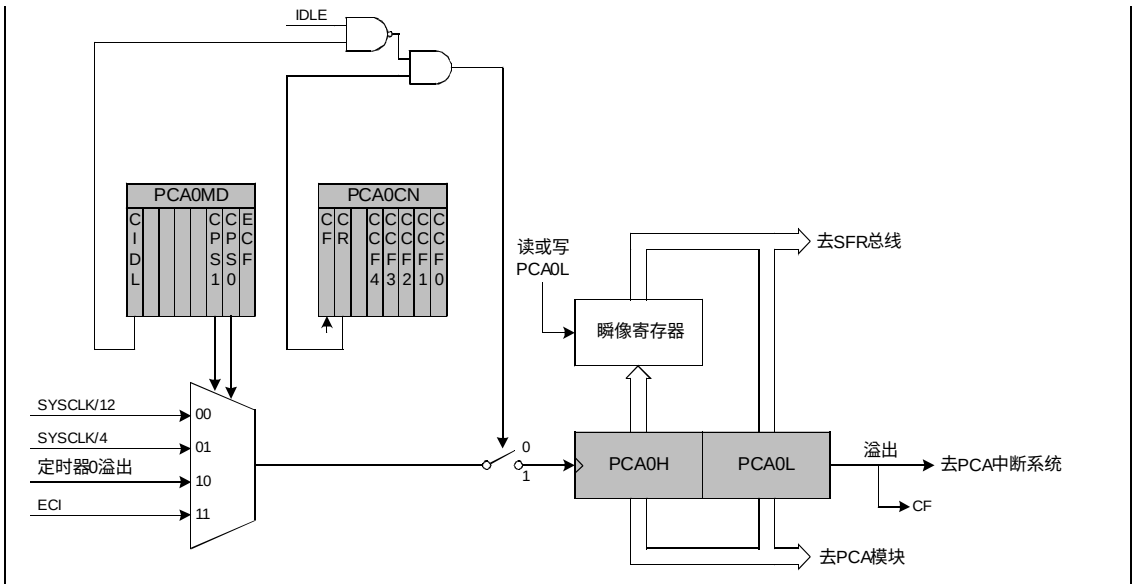
16 位的 PCA 计数器/定时器由两个 8 位的 SFR 组成：PCA0L 和 PCA0H。PCA0H 是 16 位计数器/定时器的高字节（MSB），而 PCA0L 是低字节（LSB）。在读 PCA0L 的同时自动锁存 PCA0H 的值。先读 PCA0L 寄存器将使 PCA0H 的值得到保持（在读 PCA0L 的同时），直到用户读 PCA0H 寄存器为止。读 PCA0H 或 PCA0L 不影响计数器工作。PCA0MD 寄存器中的 CPS1 和 CPS0 位用于选择 PCA 计数器/定时器的时基，如表 20.2 所示。

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 中的 ECF 位设置为逻辑 1 即可允许 CF 标志产生中断请求。当 CPU 转向中断服务程序时，CF 位不能被硬件自动清除，必须用软件清 0。（注意：要使 CF 中断得到响应，必须先总体允许 PCA0 中断。通过将 EA 位（IE.7）和 EPCA0 位（EIE1.3）设置为逻辑 1 来总体允许 PCA0 中断。）清除 PCA0MD 寄存器中的 CIDL 位将允许 PCA 在微控制器内核处于等待方式时继续正常工作。

表 20.2. PCA 时基输入选择

CPS1	CPS0	时间基准
0	0	系统时钟的 12 分频
0	1	系统时钟的 4 分频
1	0	定时器 0 溢出
1	1	ECI 下降沿（最大速率 ＝ 系统时钟频率/4）

图 20.7 PCA 计数器/定时器原理框图



20.3 PCA 寄存器说明

系统器件可以实现一个或多个可编程计数器阵列。下面对与 PCA 工作有关的特殊功能寄存器进行详细说明。更多关于 SFR 及其使用的信息见本手册的 CIP-51 系统控制器部分。

图 20.8 PCA0CN：PCA 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0	00000000
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址: 0xD8

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 CF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 6： CR：PCA 计数器 / 定时器运行控制
该位允许 / 禁止 PCA 计数器 / 定时器。
0：禁止 PCA 计数器 / 定时器
1：允许 PCA 计数器 / 定时器

位 5： 未用。读=0，写=忽略。

位 4： CCF4：PCA 模块 4 捕捉 / 比较标志
在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时，该位置‘1’将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 3： CCF3：PCA 模块 3 捕捉 / 比较标志
在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时，该位置‘1’将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

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 CCF 中断被允许时，该位置‘1’将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

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 CCF 中断被允许时，该位置‘1’将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

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 CCF 中断被允许时，该位置‘1’将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

图 20.9　PCA0MD：PCA 方式选择寄存器

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

位 7：　CIDL：PCA 计数器 / 定时器等待控制
规定 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。

位 6-3：　未用。读=0000b，写=忽略。

位 2-1：　CPS1-CPS0：PCA 计数器 / 定时器脉冲选择
这些位选择 PCA 计数器的时钟源。

CPS1	CPS0	时间基准
0	0	系统时钟的 12 分频
0	1	系统时钟的 4 分频
1	0	定时器 0 溢出
1	1	ECI 负跳变（最大速率 ＝ 系统时钟频率/4）

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）置位时，允许 PCA 计数器 / 定时器溢出（CF）中断请求。

图 20.10　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
-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xDA-0xDE

PCA0CPMn 地址：　PCA0CPM0＝0xDA(n=0)
PCA0CPM0＝0xDB(n=1)
PCA0CPM0＝0xDC(n=2)
PCA0CPM0＝0xDD(n=3)
PCA0CPM0＝0xDE(n=4)

位 7：　未用。读 = 0，写 = 忽略。

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 位置位。
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 引脚的逻辑电平切换。
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 比较器功能。如果被允许，CEXn 引脚输出脉冲宽度调制信号。
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。

图 20.11　PCA0L：PCA 计数器 / 定时器低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址：0xE9

位 7-0：　PCA0L：PCA 计数器 / 定时器的低字节
PCA0L 寄存器保存 16 位 PCA 计数器 / 定时器的低字节（LSB）。

图 20.12　PCA0H：PCA 计数器 / 定时器高字节

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

位 7-0：　PCA0H：PCA 计数器 / 定时器高字节
PCA0H 寄存器保存 16 位 PCA 计数器 / 定时器的高字节（MSB）。注意，读出来的值实际上是从瞬像（Snapshot）寄存器读出，以便与 PCA0L 同步。

图 20.13　PCA0CPLn：PCA 捕捉模块低字节

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

PCA0CPLn：　PCA0CPL0 = 0xEA (n=0)
PCA0CPL1 = 0xEB (n=1)
PCA0CPL2 = 0xEC (n=2)
PCA0CPL3 = 0xED (n=3)
PCA0CPL4 = 0xEE (n=4)

位 7-6：　PCA0CPLn：PCA 捕捉模块低字节
PCA0CPLn 寄存器保存 16 位捕捉模块 n 的低字节（LSB）。

图 20.14 PCA0CPHn：PCA 捕捉模块高字节

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

PCA0CPHn：　PCA0CPH0=0xEA (n=0)
PCA0CPH1=0xEB (n=1)
PCA0CPH2=0xEC (n=2)
PCA0CPH3=0xED (n=3)
PCA0CPH4=0xEE (n=4)

位 7-6：　PCA0CPHn：PCA 捕捉模块高字节
PCA0CPHn 寄存器保存 16 位捕捉模块 n 的高字节（MSB）。

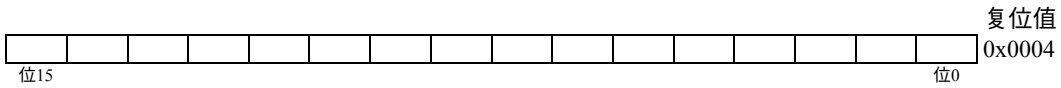
21. JTAG (IEEE 1149.1)

每个 MCU 都有一个片内 JTAG 接口和逻辑，提供生产和在系统测试所需要的边界扫描功能，支持闪存的读和写操作以及非侵入式在系统调试。MCU 中的 JTAG 接口完全符合 IEEE 1149.1 规范。关于测试接口和边界扫描结构方面的详细信息请参见该规范。在 IEEE 1149.1 规范中的测试接口和操作部分介绍了如何访问 JTAG 指令寄存器（IR）和数据寄存器（DR）。

JTAG 接口使用 MCU 上的四个专用引脚，它们是：TCK、TMS、TDI 和 TDO。这些引脚都耐 5V 电压。

通过 16 位 JTAG 指令寄存器（IR）可以发出图 21.1 所示的 8 种指令。MCU 中有三个与 JTAG 边界扫描相关的数据寄存器和四个与 FLASH 读/写操作相关的寄存器。

图 21.1　IR：JTAG 指令寄存器



IR 值	指令	描述
0x0000	EXTEST	选择边界数据寄存器，以控制和观察所有的器件引脚。
0x0002	SAMPLE/ PRELOAD	选择边界数据寄存器，以观察和预设扫描路径锁存器。
0x0004	IDCODE	选择器件的 ID 寄存器
0xFFFF	BYPASS	选择旁路数据寄存器
0x0082	闪存控制	选择 FLASHCON 寄存器，以便控制接口逻辑对读和写 FLASHDAT 寄存器如何响应。
0x0083	闪存数据	选择 FLASHDAT 寄存器，以便读/写 FLASH 存储器。
0x0084	闪存地址	选择 FLASHADR 寄存器，该寄存器保持 FLASH 读、写和擦除操作的地址。
0x0085	闪存定标	选择 FLASHSCL 寄存器，该寄存器控制用于产生 FLASH 操作定时信号的预分频器。

21.1 边界扫描

边界扫描路径中的数据寄存器是一个 87 位的移位寄存器。通过执行 EXTEST 和 SAMPLE 命令，边界数据寄存器能提供对所有器件引脚以及 SFR 总线和弱上拉功能的控制和观察。

表 21.1 边界数据寄存器位定义

EXTEST 支持捕捉和更新两种操作，而采样只执行捕捉操作。

位	操作	目的
0	捕捉	来自 MCU 的复位允许
	更新	去/RST 引脚的复位允许
1	捕捉	来自/RST 引脚的复位输入
	更新	去/RST 引脚的复位输出
2	捕捉	来自 XTAL1 引脚的外部时钟
	更新	未用
3	捕捉	来自 MCU 的弱上拉允许
	更新	去端口引脚的弱上来允许
4-11	捕捉	来自 CIP-51 的 SFR 地址总线位(如 Bit4=SFRA0, Bit5=SFRA1…)
	更新	去 SFR 地址总线的 SFR 地址总线位(如 Bit4=SFRA0, Bit5=SFRA1)
12-19	捕捉	从 SFR 读取的 SFR 数据总线位(如 Bit12=SFRD0, Bit13=SFRD1…)
	更新	向 SFR 写入的 SFR 数据总线位(如 Bit12=SFRD0, Bit13=SFRD1…)
20	捕捉	来自 CIP-51 的 SFR 写选通
	更新	去 SFR 总线的 SFR 写选通
21	捕捉	来自 CIP-51 的 SFR 读选通
	更新	去 SFR 总线的 SFR 读选通
22	捕捉	来自 CIP-51 的 SFR 读/修改/写选通信号
	更新	去 SFR 总线的 SFR 读/修改/写选通信号
23,25,27,29, 31,33,35,37	捕捉	来自 MCU 的 P0.n 输出允许（如 Bit23=P0.0, Bit25=P0.1 等）
	更新	去引脚的 P0.n 输出允许（如 Bit23=P0.0oe, Bit25=P0.1oe 等）
24,26,28,30, 32,34,36,38	捕捉	来自引脚的 P0.n 输入（如 Bit24=P0.0, Bit26=P0.1 等）
	更新	P0.n 输出到引脚（如 Bit24=P0.0, Bit26=P0.1 等）
39,41,43,45, 47,49,51,53	捕捉	来自 MCU 的 P1.n 输出允许（如 Bit39=P1.0, Bit41=P1.1 等）
	更新	去引脚的 P1.n 输出允许（如 Bit39=P1.0oe, Bit41=P1.1oe 等）
40,42,44,46, 48,50,52,54	捕捉	来自引脚的 P1.n 输入（如 Bit40=P1.0, Bit42=P1.1 等）
	更新	P1.n 输出到引脚（如 Bit40=P1.0, Bit42=P1.1 等）
55,57,59,61, 63,65,67,69	捕捉	来自 MCU 的 P2.n 输出允许（如 Bit55=P2.0, Bit57=P2.1 等）
	更新	去引脚的 P2.n 输出允许（如 Bit55=P2.0oe, Bit57=P2.1oe 等）
56,58,60,62, 64,66,68,70	捕捉	来自引脚的 P2.n 输入（如 Bit56=P2.0, Bit58=P2.1 等）
	更新	P2.n 输出到引脚（如 Bit56=P2.0, Bit58=P2.1 等）
71,73,75,77, 79,81,83,85	捕捉	来自 MCU 的 P3.n 输出允许（如 Bit71=P3.0, Bit73=P3.1 等）
	更新	去引脚的 P3.n 输出允许（如 Bit71=P3.0oe, Bit73=P3.1oe 等）
72,74,76,78, 80,82,84,86	捕捉	来自引脚的 P3.n 输入（如 Bit72=P3.0, Bit74=P3.1 等）
	更新	P3.n 输出到引脚（如 Bit72=P3.0, Bit74=P3.1 等）

21.1.1 EXTEST 指令

通过 IR 进入 EXTEST 指令。边界数据寄存器提供对所有器件引脚以及 SFR 总线和弱上拉功能的控制和观察。片内逻辑的所有输入都被设置为 1。

21.1.2 SAMPLE 指令

通过 IR 进入 SAMPLE 指令。边界数据寄存器提供对扫描路径锁存器的观察和预置。

21.1.3 BYPASS 指令

通过 IR 进入 BYPASS 指令。它提供对标准 1 位 JTAG 旁路数据寄存器的访问。

21.1.4 IDCODE 指令

通过 IR 进入 IDCODE 指令。它提供对 32 位器件 ID 寄存器的访问。

图 21.2 DEVICEID：JTAG 器件 ID 寄存器

版本	器件编号	制造商 ID	1	复位值 (取决于器件)
位31 位28	位27 位12	位11 位1	位0	

版本=0000b（版本 A）
=0001b（版本 B）
器件编号= 0000 0000 0000 0000b（C8051F000/01/02/10/11/12）
= 0000 0000 0000 0010b（C8051F005/06/07/15/16/17）
制造商 ID= 0010 0100 001b（Cygnal 集成产品公司）

21.2 FLASH 编程命令

通过 JTAG 接口可以直接对 FLASH 存储器编程，编程时要使用 FLASH 控制、FLASH 数据、FLASH 地址和 FLASH 预分频寄存器。这些间接寄存器是通过 JTAG 指令寄存器访问的。对间接数据寄存器进行读和写操作时，首先要要在 IR 寄存器中设置正确的数据寄存器地址。每次读或写都是通过向所选择的数据寄存器写入适当的间接操作码（IndOpCode）来启动的。输入到该寄存器的命令具有如下格式：

19:18	17:0
IndOpCode	WriteData（待写数据）

IndOpCode：这些位根据下表来设置要执行的操作：

IndOpCode	操作
0x	查询
10	读
11	写

查询操作用于检查下面将要介绍的 Busy（忙）位。查询操作执行一次 Capture_DR（数据寄存器捕捉），但不执行 Update_DR（数据寄存器更新）。由于更新操作被禁止，查询操作只进行一个二进制位的移入/移出。

读操作启动一次对由 IR 寻址的寄存器的读取。只需向间接寄存器移入两位即可启动读操作。在读操作被启动后，必须通过查询 Busy 位来确定该操作何时完成。

写操作启动一次对由 IR 寻址的寄存器的写入。可以写任何数据长度不超过 18 位的寄存器。如果待写寄存器的数据不足 18 位，应对位于 WriteData 字段的数据进行左对齐，即 MSB 应占据位 17。这就允许以较少的 JTAG 时钟周期对较短的寄存器进行写入。例如，只需进行 10 次移位即可完成对一个 8 位寄存器的写入。在写操作被启动后，必须通过查询 Busy 位来确定何时能启动下一次操作。在读或写操作正在进行时，不应改变指令寄存器中的内容。

间接数据寄存器输出的数据具有如下格式：

19	18:1	0
0	ReadData（读出的数据）	Busy（忙）

Busy 位指示当前操作是否完成。它在操作被启动后变高，在操作完成后变低。在 Busy 为高时，读和写命令均被忽略。实际上，如果需要在查到 Busy 位为低电平之后进行另一次读或写操作，则下一操作的 JTAG 写可以在查询 Busy 位是否为低电平时进行。该操作将被忽略，直到 Busy 位变低为止，此时将启动新操作。该位处于最低位（位 0），只需移位一次即可对其查询。当等待一个读操作完成而 Busy 位为 0 时，可以移出后面的 18 位以得到结果数据。ReadData（读出的数据）总是右对齐的，这就允许以较少的移位次数读取长度小于 18 位的寄存器。例如，只需进行 9 次移位（Busy + 8 位）即可得到一次字节读的结果。

图 21.3　FLASHCON：JTAG FLASH 控制寄存器

WRMD3	WRMD2	WRMD1	WRMD0	RDMD3	RDMD2	RDMD1	RDMD0	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000

该寄存器决定 FLASH 接口逻辑对读/写 FLASHDAT 寄存器的操作如何响应。

位 7-4：　WRMD3-0：写方式选择
写方式选择位控制 FLASH 接口逻辑对写 FLASHDAT 寄存器的操作如何响应，其值意义如下：
0000：　一个 FLASHDAT 写操作替换 FLASHDAT 寄存器中的数据，否则忽略。
0001：　一个 FLASHDAT 写操作启动对存储器的写入，写入地址由 FLASHADR 寄存器给出。FLASHADR 在操作完成后增 1。
0010：　一个 FLASHDAT 写操作启动对一个 FLASH 页的擦除，FLASHADR 寄存器给出待擦页内的一个地址。对于擦除操作，FLASHDAT 中的值必须是 0xA5。FLASHADR 不受影响。如果 FLASHADR = 0x7DFE – 0x7DFF，则整个用户空间将被擦除（即除保留区外的整个 FLASH 存储器）。
（所有其它 WRMD3-0 值保留。）

位 3-0：　RDMD3-0：读方式选择位
读方式选择位控制 FLASH 接口逻辑对读 FLASHDAT 寄存器的操作如何响应，其值意义如下：
0000：　一个 FLASHDAT 读操作提供 FLASHDAT 寄存器中的数据，否则忽略。
0001：　如果没有其它操作正在执行，一个 FLASHDAT 读操作启动对存储器的字节读，待读字节的地址由 FLASHADR 寄存器给出。该方式用于块读。
0010：　在没有其它操作正在执行和前面的读操作的数据已经被读出的情况下，一个 FLASHDAT 读操作启动对存储器的读，待读字节的地址由 FLASHADR 寄存器给出。该方式允许读一个字节（或一个存储块内的最后一个字节）。
（所有其它 RDMD3-0 值保留）

图 21.4　FLASHADR：JTAG FLASH 地址寄存器

																复位值
																0x0000
位15															位0	

该寄存器保持所有 JTAG FLASH 读、写和擦除操作的地址。该寄存器在每次读或写操作后自动加 1，不管操作成功或是失败。

位 15-0：FLASH 操作的 16 位地址。

图 21.5　FLASHDAT：JTAG FLASH 数据寄存器

DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	FAIL	FBUSY	复位值
位9	位8	位7	位6	位5	位4	位3	位2	位1	位0	0000000000

该寄存器用于通过 JTAG 接口读或写 FLASH 存储器。

位 9-2：　DATA7-0：FLASH 数据字节
位 1：　FAIL：FLASH 操作失败标志
0：FLASH 存储器操作成功。
1：FLASH 存储器操作失败。通常指示相关存储器单元被锁定。

位 0：　FBUSY：FLASH 忙标志
0：FLASH 接口逻辑不忙。
1：FLASH 接口逻辑正在处理一个请求。当 FBUSY= 1 时，读或写都不会启动另一次操作。

图 21.6　FLASHSCL：JTAG FLASH 定时预分频寄存器

FOSE	FRAE	-	-	FLSCL3	FLSCL2	FLSCL1	FLSCL0	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000

该寄存器控制 FLASH 读时序电路和预分频器，预分频器用于产生 FLASH 操作所要求的正确定时信号。

位 7：　FOSE：FLASH 单稳态定时器允许位
0：闪存读选通信号宽度为一个完整的定时周期。
1：闪存读选通信号宽度为 50 ns。

位 6：　FRAE：FLASH 总读位
0：FLASH 输出允许和读出放大器允许信号只在读 FLASH 期间有效。
1：FLASH 输出允许和读出放大器允许信号一直有效。这可以用于限制因切换读出放大器而引起的数字电源电流的变化，从而减小数字感应噪声。

位 5-4：　未用。读=00b，写=忽略。

位 3-0：　FLSCL3-0：FLASH 定时预分频控制位
FLSCL3-0 控制用于产生 FLASH 操作定时信号的预分频器。其值应在启动任何 FLASH 操作之前写入，写入值应是满足下式的最小整数：

$$FLSCL[3:0] > \log_2(f_{SYSCLK}/50kHz)$$

其中 f_{SYSCLK} 是系统时钟频率。当 FLSCL[3:0] = 1111b 时，所有对 FLASH 的读/写/擦除操作都被禁止。

21.3 调试支持

每个 MCU 内部都有 JTAG 和调试电路。通过 4 脚的 JTAG 接口，可以使用安装在最终应用系统上的产品 MCU 进行非侵入式、全速、在系统调试。Cygnal 的调试系统支持观察和修改存储器和寄存器，设置断点、观察点，支持单步及运行和停机命令；不需要额外的目标 RAM、程序存储器或通信通道。在调试时，所有的模拟和数字外设都全功能正确运行（保持同步）。当 MCU 因单步执行或执行到断点而停机时，WDT 被禁止。

开发套件 C8051F000DK、C8051F005DK、C8051F010DK 和 C8051F015DK 具有开发应用代码和进行在系统调试所需要的全部硬件和软件。每个套件包括一个具有调试器和 8051 汇编器的集成开发环境（IDE）、一个被称为 EC 的 RS232 到 JTAG 的协议转换模块和一个安装有 C8051F000、F005、F010 或 F015 的目标应用板（板上有大块样机区）。每个套件还包括 RS-232 和 JTAG 电缆及墙装电源。