C8051F120/1/2/3/4/5/6/7
C8051F130/1/2/3 系列
混合信号 ISP FLASH 微控制器
数 据 手 册

潘 琢 金 译

Rev 1.3 2004.12

版 权 声 明

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

译者联系方式：

沈阳航空工业学院 计算机学院　　潘琢金

电话：024-86802267，13066535936
Email：panzhuojin@sina.com 或 panzhj@syiae.edu.cn

混合信号 ISP FLASH 微控制器

模拟外设

- **10 或 12 位 SAR ADC**
  - ±1LSB INL
  - 可编程转换速率，最大 100ksps
  - 可多达 8 个外部输入；可编程为单端输入或差分输入
  - 可编程放大器增益：16、8、4、2、1、0.5
  - 数据相关窗口中断发生器
  - 内置温度传感器
- **8 位 SAR ADC（仅 F12x）**
  - 可编程转换速率，最大 500ksps
  - 8 个外部输入（单端或差分）
  - 可编程放大器增益：4、2、1、0.5
- **两个 12 位 DAC（仅 F12x）**
  - 可用定时器触发同步输出，用于产生无抖动波形
- **两个模拟比较器**
- **电压基准**
- **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、非侵入式的在片/在系统调试
- 支持断点、单步、观察点、堆栈监视器；可以观察/修改存储器和寄存器
- 比使用仿真芯片、目标仿真头和仿真插座的仿真系统有更好的性能
- 符合 IEEE1149.1 边界扫描标准
- 完全的开发套件

100 脚 TQFP 和 64 脚 TQFP 封装

- 温度范围：-40°C - +85°C

高速 8051 微控制器内核

- 流水线指令结构；70%的指令的执行时间为一个或两个系统时钟周期
- 使用内部集成 PLL 时速度可达 100 或 50MIPS
- 2 周期 16 x 16 MAC 引擎（仅 C8051F120/1/2/3 和 C8051F130/1/2/3）

存储器

- 8448 字节内部数据 RAM（8K + 256）
- 128KB 或 64KB 分区 FLASH；可以在系统编程，扇区大小为 1024 字节
- 外部 64KB 数据存储器接口（可编程为复用方式或非复用方式）

数字外设

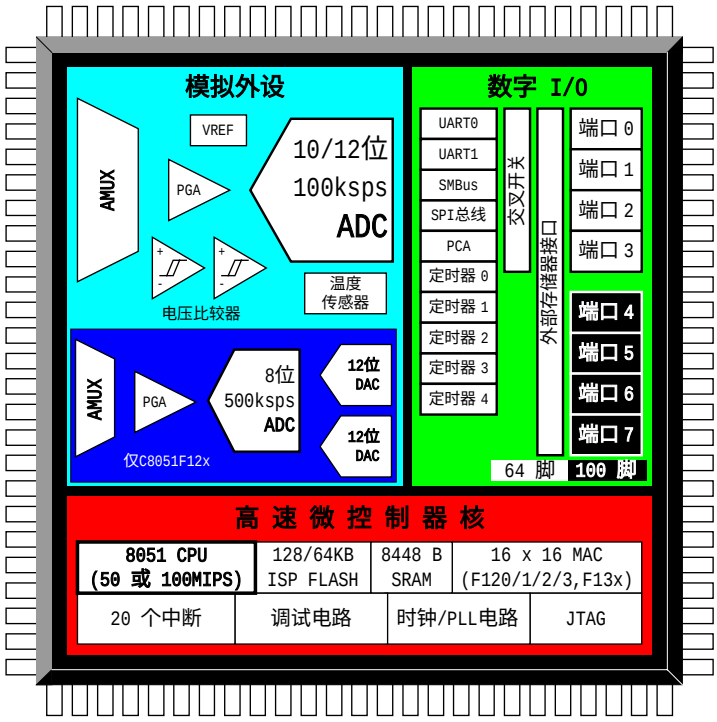
- 8 个 8 位宽端口 I/O（100TQFP），耐 5V
- 4 个 8 位宽端口 I/O（64TQFP），耐 5V
- 可同时使用的硬件 SMBus(I^2C^{TM} 兼容)、SPI^{TM} 及两个 UART 串行端口
- 可编程的 16 位计数器/定时器阵列，有 6 个捕捉/比较模块
- 5 个通用 16 位计数器/定时器
- 专用的看门狗定时器；双向复位引脚

时钟源

- 内部精确振荡器：24.5MHz
- 可灵活配置的 PLL
- 外部振荡器：晶体、RC、C、或外部时钟

供电电压

- 电源范围：2.7-3.6V(50MIPS) 3.0-3.6V(100MIPS)
- 节电休眠和停机方式



目 录

1．系统概述

C8051F12x 和 C8051F13x 系列器件是完全集成的混合信号片上系统型 MCU 芯片，具有 64 个数字 I/O 引脚（100 脚 TQFP 封装）或 32 个数字 I/O 引脚（64 脚 TQFP 封装）。下面列出了一些主要特性；有关某一产品的具体特性参见表 1.1。

- 高速、流水线结构的 8051 兼容的 CIP-51 内核（100MIPS 或 50MIPS）
- 全速、非侵入式的在系统调试接口（片内）
- 真正 12 位或 10 位、100 ksps 的 ADC，带 PGA 和 8 通道模拟多路开关
- 真正 8 位 500 ksps 的 ADC，带 PGA 和 8 通道模拟多路开关（仅 C8051F12x）
- 两个 12 位 DAC，具有可编程数据更新方式（仅 C8051F12x）
- 2 周期的 16 x 16 乘法和累加引擎（仅 C8051F120/1/2/3 和 C8051F130/1/2/3）
- 128KK 或 64KB 可在系统编程的 FLASH 存储器
- 8448（8K+256）字节的片内 RAM
- 可寻址 64KB 地址空间的外部数据存储器接口
- 硬件实现的 SPI、SMBus/ I^2C 和两个 UART 串行接口
- 5 个通用的 16 位定时器
- 具有 6 个捕捉/比较模块的可编程计数器/定时器阵列
- 片内看门狗定时器、VDD 监视器和温度传感器

具有片内 VDD 监视器、看门狗定时器和时钟振荡器的 C8051F12x 和 C8051F13x 器件是真正能独立工作的片上系统。所有模拟和数字外设均可由用户固件使能/禁止和配置。FLASH 存储器还具有在系统重新编程能力，可用于非易失性数据存储，并允许现场更新 8051 固件。

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

每个 MCU 都可在工业温度范围（-45℃到+85℃）工作。端口 I/O、/RST 和 JTAG 引脚都容许 5V 的输入信号电压。有 100 脚 TQFP 封装和 64 脚 TQFP 封装。表 1.1 列出了每个器件的特性和封装。图 1.1 ~ 图 1.6 给出了每种器件的功能框图。

表 1.1 产品选择指南

器件号	MIPS(峰值)	FLASH 存储器	RAM	2 周期 16 x 16 MAC	外部存储器接口	SMBus/I²C	SPI	UART	定时器(16 位)	可编程计数器阵列	数字端口 I/O	12 位 100ksps ADC 输入	10 位 100ksps ADC 输入	8 位 500ksps ADC 输入	电压基准	温度传感器	DAC 分辨率（位）	DAC 输出	模拟比较器	封装
C8051F120	100	128k	8448	√	√	√	√	2	5	√	64	8	-	8	√	√	12	2	2	100TQFP
C8051F121	100	128k	8448	√	√	√	√	2	5	√	32	8	-	8	√	√	12	2	2	64TQFP
C8051F122	100	128k	8448	√	√	√	√	2	5	√	64	-	8	8	√	√	12	2	2	100TQFP
C8051F123	100	128k	8448	√	√	√	√	2	5	√	32	-	8	8	√	√	12	2	2	64TQFP
C8051F124	50	128k	8448		√	√	√	2	5	√	64	8	-	8	√	√	12	2	2	100TQFP
C8051F125	50	128k	8448		√	√	√	2	5	√	32	8	-	8	√	√	12	2	2	64TQFP
C8051F126	50	128k	8448		√	√	√	2	5	√	64	-	8	8	√	√	12	2	2	100TQFP
C8051F127	50	128k	8448		√	√	√	2	5	√	32	-	8	8	√	√	12	2	2	64TQFP
C8051F130	100	128k	8448	√	√	√	√	2	5	√	64	-	8	-	√	√	-	-	2	100TQFP
C8051F131	100	128k	8448	√	√	√	√	2	5	√	32	-	8	-	√	√	-	-	2	64TQFP
C8051F132	100	64k	8448	√	√	√	√	2	5	√	64	-	8	-	√	√	-	-	2	100TQFP
C8051F133	100	64k	8448	√	√	√	√	2	5	√	32	-	8	-	√	√	-	-	2	64TQFP

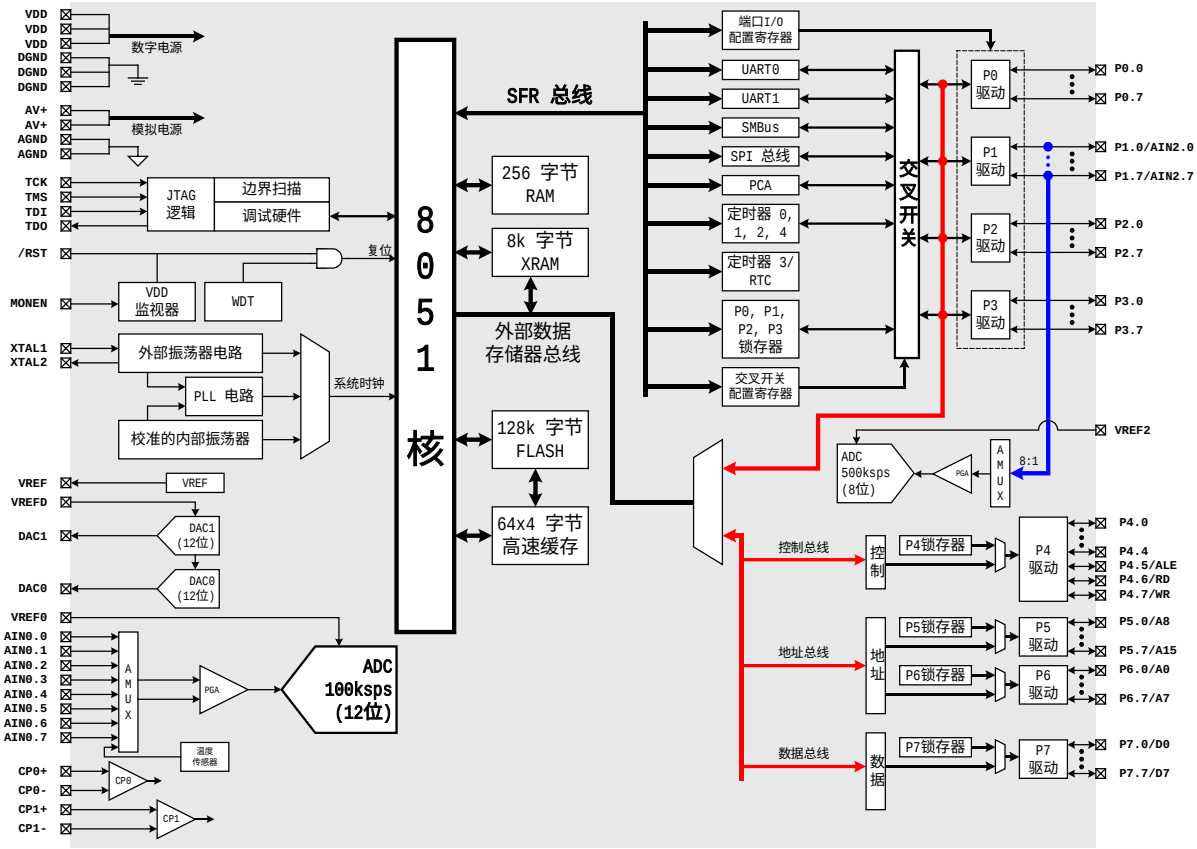


图 1.1 C8051F120/124 原理框图

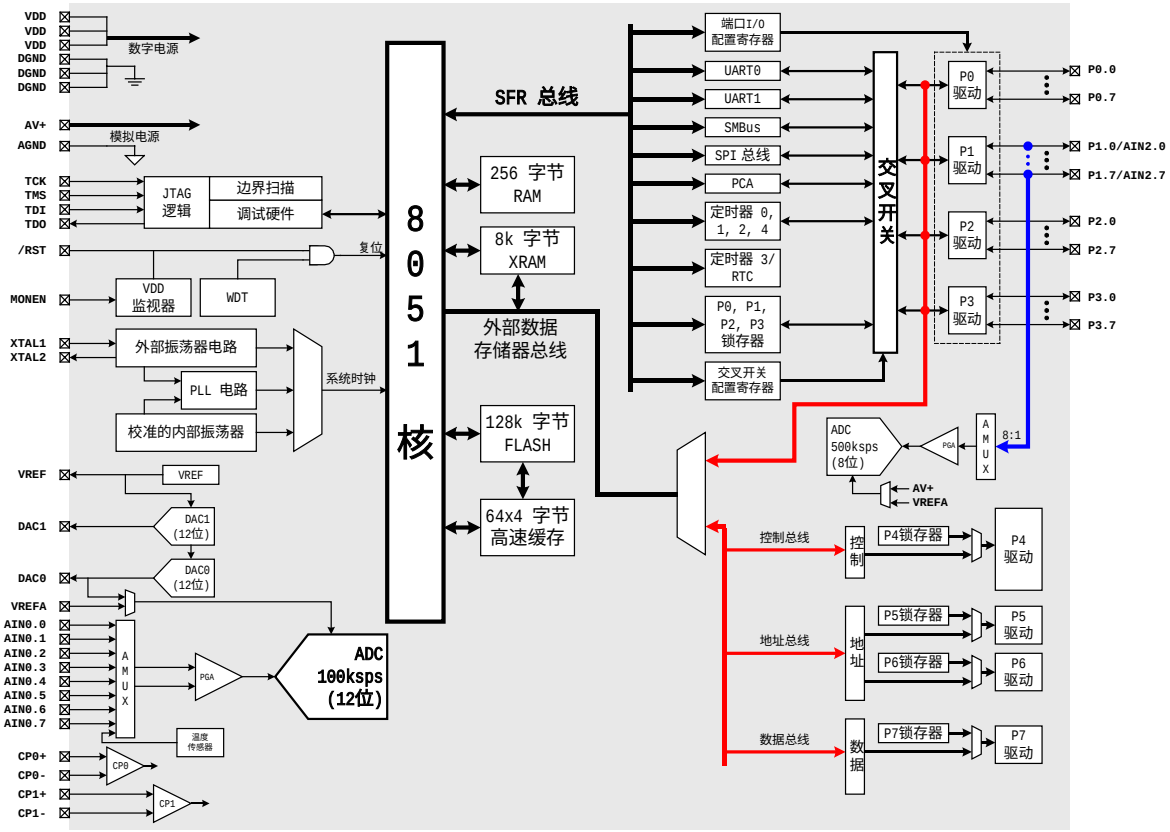


图 1.2 C8051F121/125 原理框图

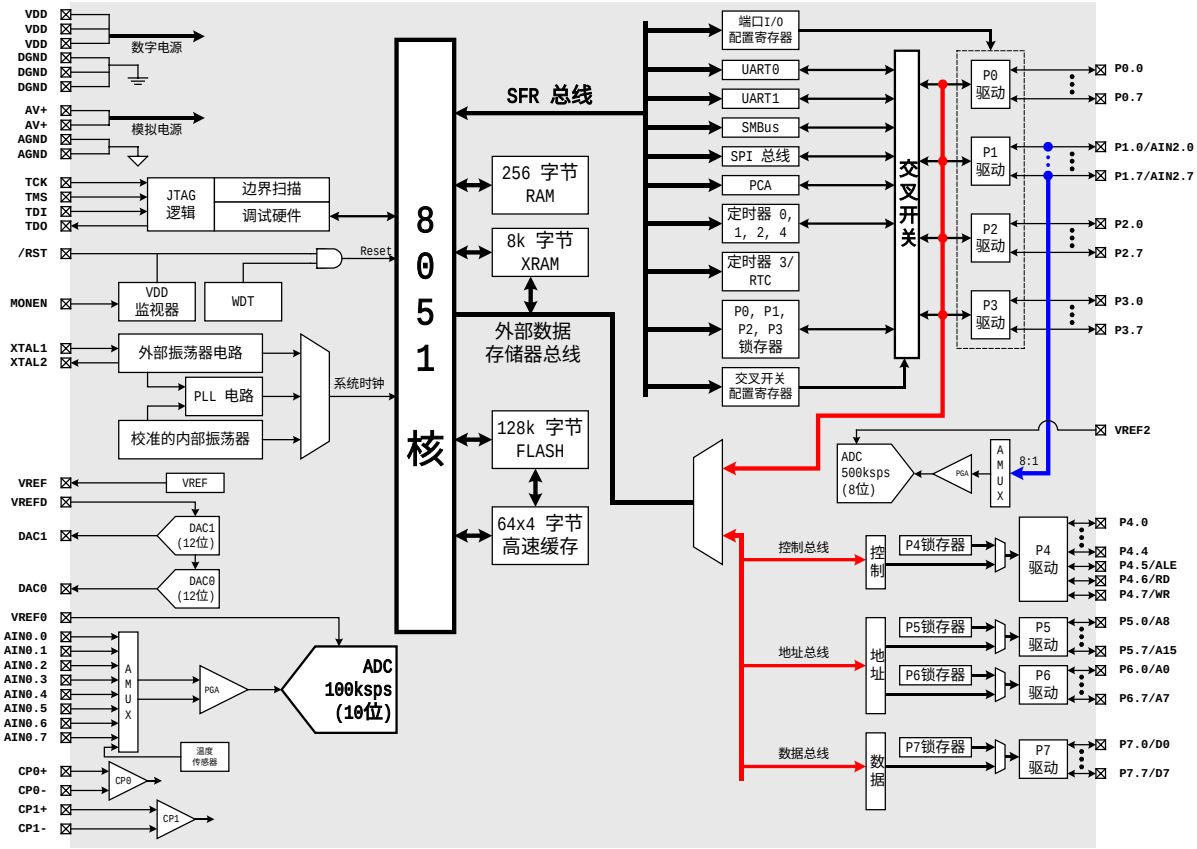


图 1.3 C8051F122/126 原理框图

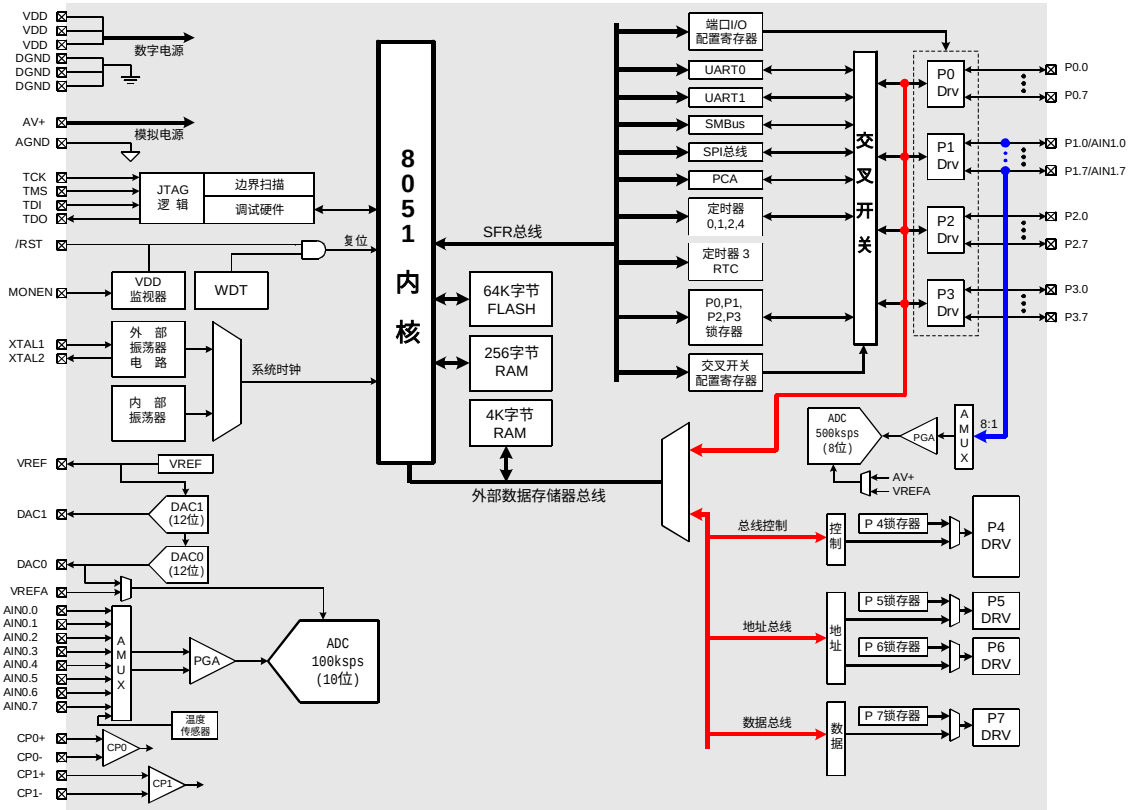


图 1.4 C8051F123/127 原理框图

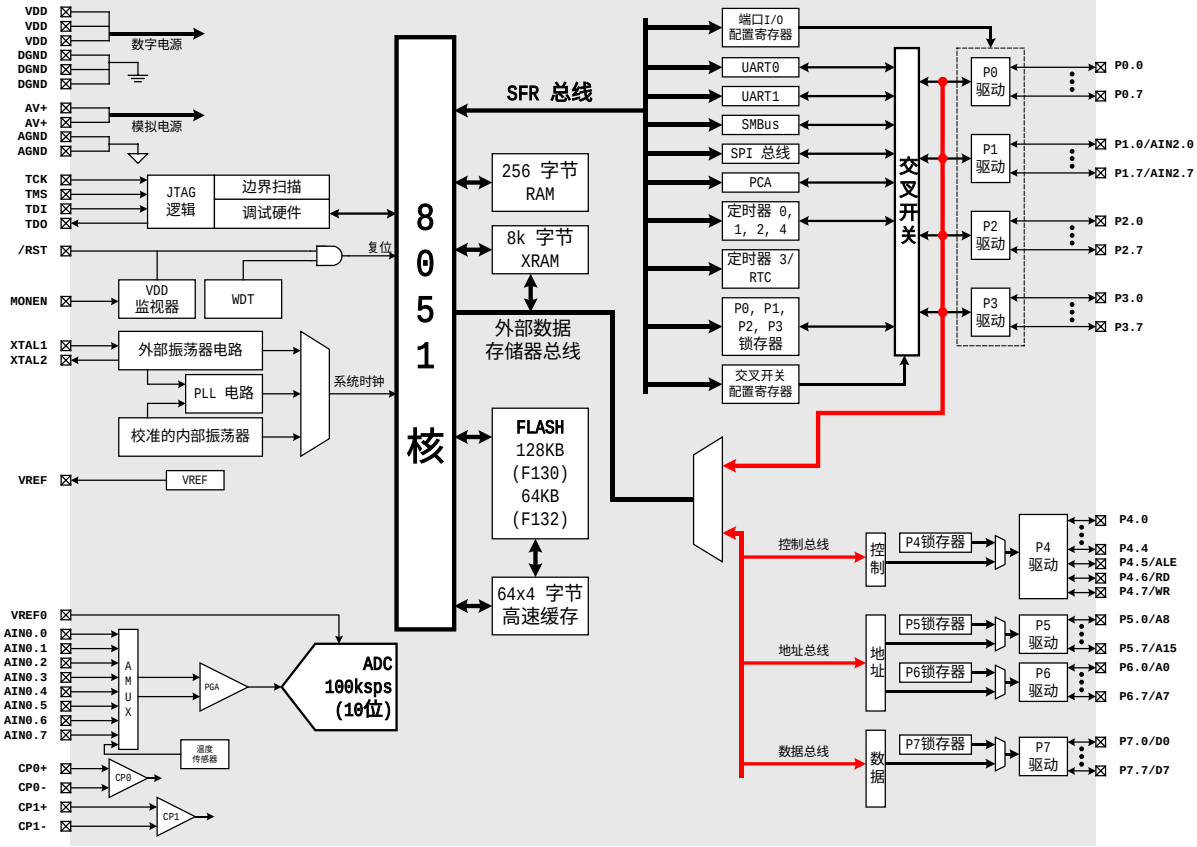


图 1.5 C8051F130/132 原理框图

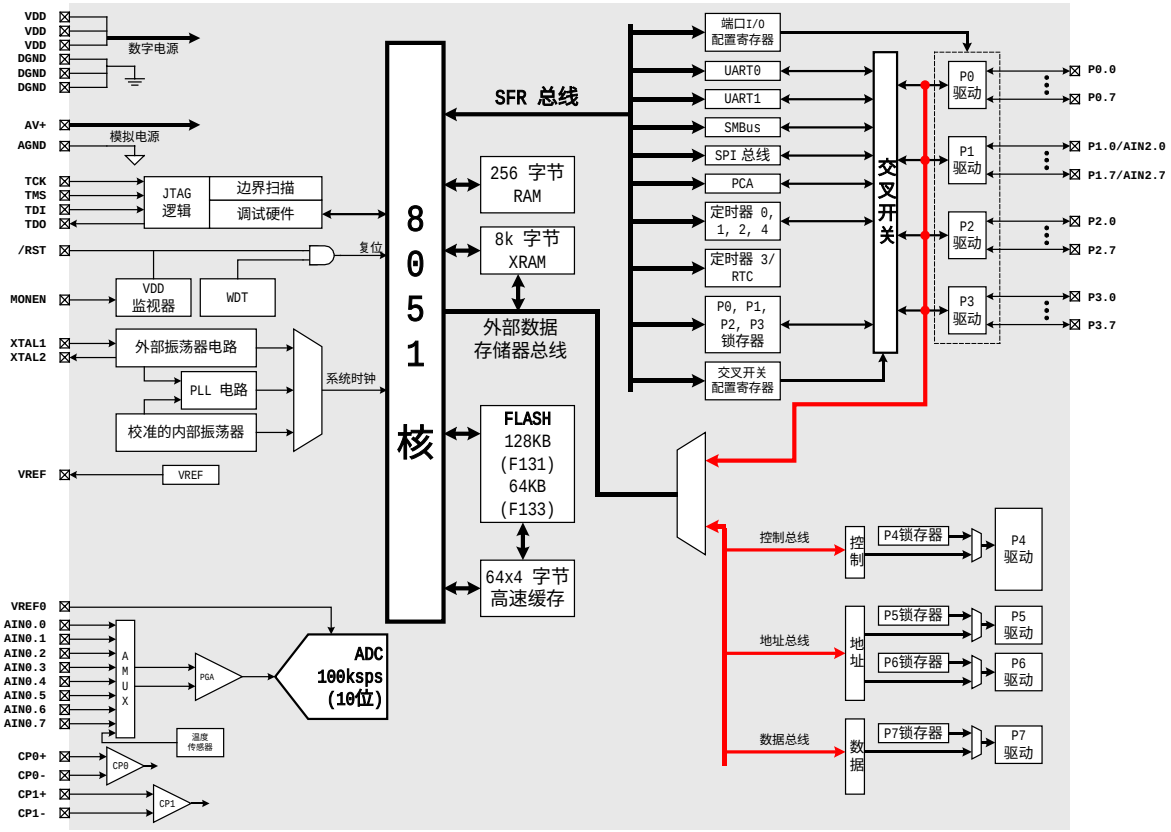


图 1.6 C8051F131/133 原理框图

1.1 CIP-51™ 微控制器核

1.1.1 与 8051 完全兼容

C8051F12x 和 C8051F13x 系列器件使用 Silicon Lab 的专利 CIP-51 微控制器内核。CIP-51 与 MCS-51™ 指令集完全兼容，可以使用标准 803x/805x 的汇编器和编译器进行软件开发。CIP-51 内核具有标准 8052 的所有外设部件，包括 5 个 16 位的计数器/定时器、两个全双工 UART、256 字节内部 RAM、128 字节特殊功能寄存器（SFR）地址空间及 8/4 个 8 位宽的 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。在标准 8051 中，除 MUL 和 DIV 以外所有指令都需要 12 或 24 个系统时钟周期，最大系统时钟频率为 12-24MHz。而对于 CIP-51 内核，70%的指令的执行时间为 1 或 2 个系统时钟周期，只有 4 条指令的执行时间大于 4 个系统时钟周期。

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

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

CIP-51 工作在最大系统时钟频率 100MHz 时，C8051F120/1/2/3 和 C8051F130/1/2/3 的峰值性能达到 100MIPS（C8051F124/5/6/7 的峰值性能达到 50MIPS）。

1.1.3 增加的功能

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

扩展的中断系统向 CIP-51 提供 20 个中断源（标准 8051 只有 7 个中断源），允许大量的模拟和数字外设中断微控制器。一个中断驱动的系统需要较少的 MCU 干预，因而有更高的执行效率。在设计一个多任务实时系统时，这些增加的中断源是非常有用的。

MCU 可有多达 7 个复位源：一个片内 VDD 监视器、一个看门狗定时器、一个时钟丢失检测器、一个由比较器 0 提供的电压检测器、一个软件强制复位、CNVSTR0 输入引脚及/RST 引脚。/RST 引脚是双向的，可接受外部复位或将内部产生的上电复位信号输出到/RST 引脚。除了 VDD 监视器和复位输入引脚以外，每个复位源都可以由用户用软件禁止；使用 MONEN 引脚使能/禁止 VDD 监视器。在一次上电复位之后的 MCU 初始化期间，可以用软件将 WDT 永久性使能。

MCU 内部有一个独立运行的时钟发生器，在复位后被默认为系统时钟。如果需要，时钟源可以在运行时切换到外部振荡器，外部振荡器可以使用晶体、陶瓷谐振器、电容、RC 或外部时钟源产生系统时钟。时钟切换功能在低功耗系统中是非常有用的，它允许 MCU 从一个低频率（节电）外部晶体源运行，当需要时再周期性地切换到 24.5MHz 的内部振荡器。另外，片内提供的 PLL 允许达到更高的系统时钟频率以提高运行速度。

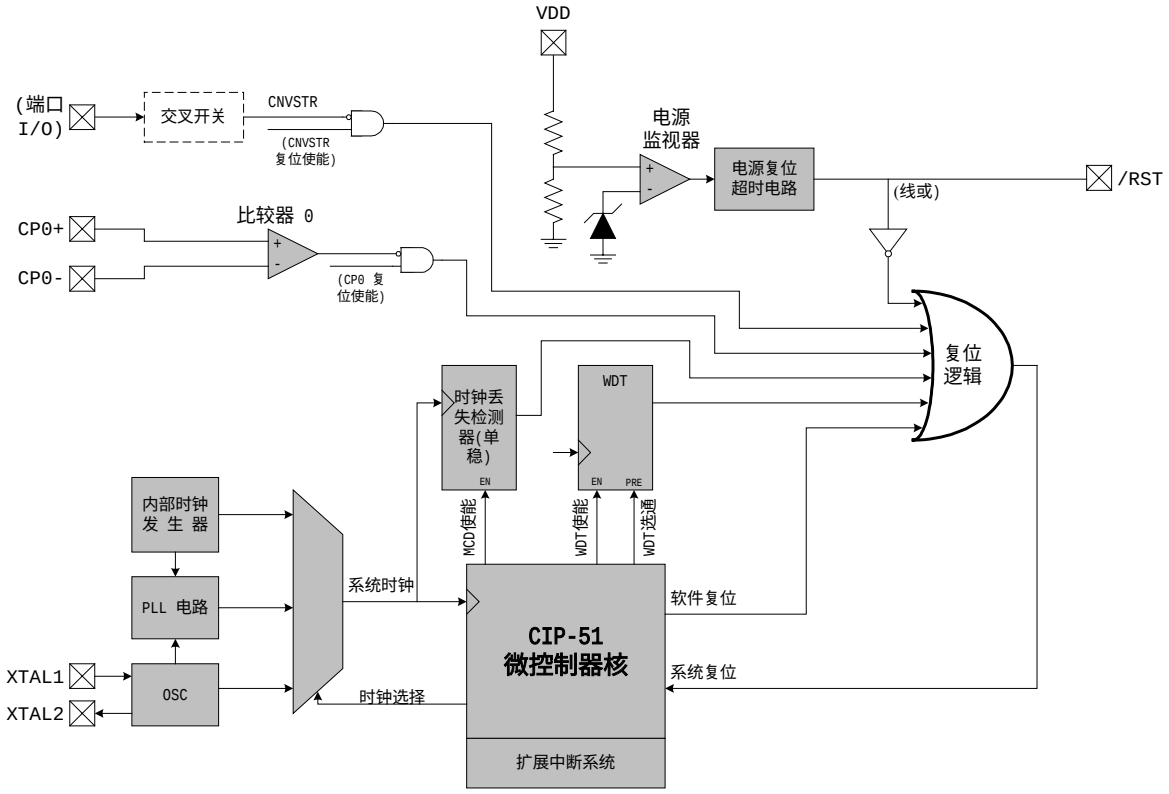


图 1.7 片内时钟和复位电路

1.2 片内存储器

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

C8051F12x 和 C8051F13x 器件还另有位于外部数据存储器地址空间的 8K 字节的 RAM 块和一个可用于访问外部数据存储器的外部存储器接口（EMIF）。这个片内的 8K 字节 RAM 块可以在整个 64K 外部数据存储器地址空间中被寻址（以 8K 为边界重叠）。外部数据存储器地址空间可以只映射到片内存储器、只映射到片外存储器、或两者的组合（8K 以下的地址指向片内，8K 以上的地址指向 EMIF）。EMIF 可以被配置为地址/数据线复用方式或非复用方式。

C8051F12x 和 C8051F13x 器件的程序存储器包含 128K 字节的分块 FLASH。该存储器以 1024 字节为一个扇区，可以在系统编程，且不需特别的外部编程电压。从 0x1FC00 到 0x1FFFF 的 1024 字节被保留。

还有两个位于地址 0x20000 - 0x200FF 的 128 字节扇区，这两个扇区可被软件用于数据存储。图 1.8 给出了 MCU 系统的存储器结构。

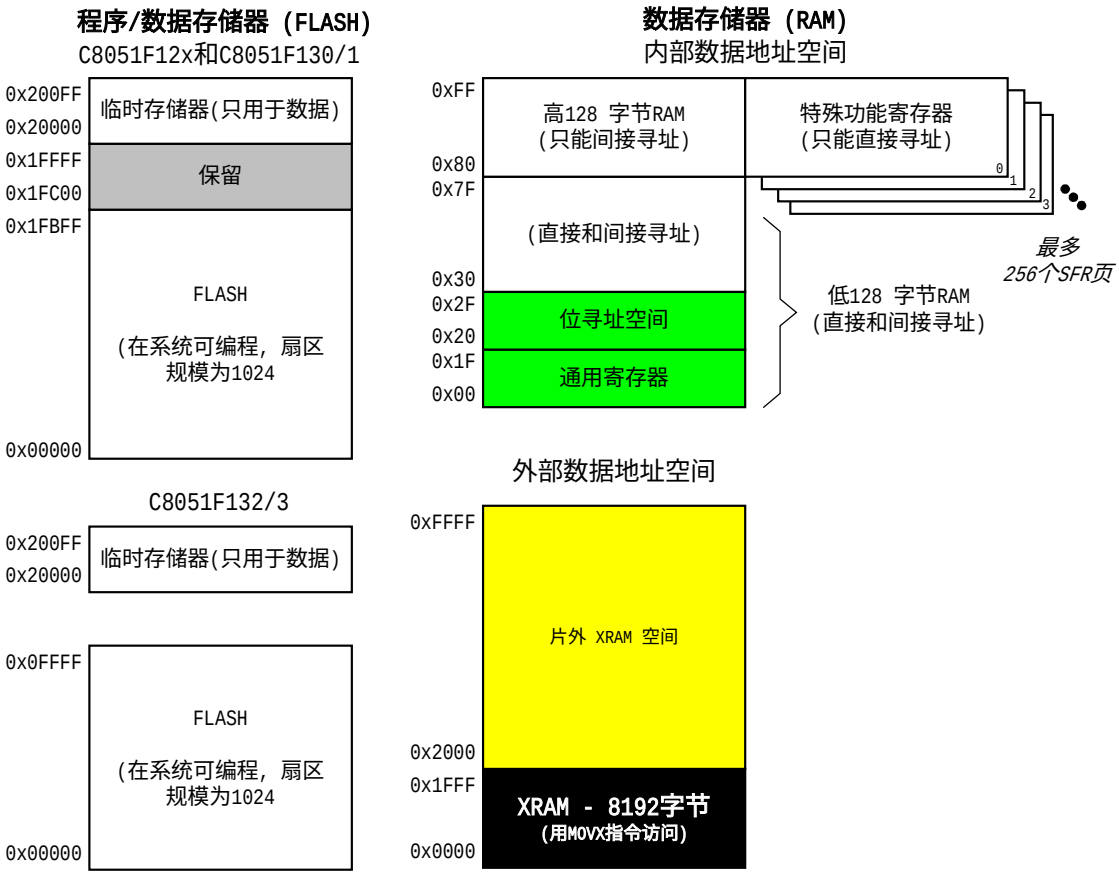


图 1.8 片内存储器组织

1.3 JTAG 调试和边界扫描

C8051F12x和C8051F13x系列器件具有片内JTAG边界扫描和调试电路，通过4脚JTAG接口并使用安装在最终应用系统中的产品器件就可以进行非侵入式、全速的在系统调试。该JTAG接口完全符合IEEE 1149.1规范，为生产和测试提供完全的边界扫描功能。

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

开发套件C8051F120DK具有开发应用代码所需要的全部硬件和软件，并可对C8051F12x MCU进行在系统调试。开发套件中包括开发者工作室软件和调试器、一个集成的8051汇编器和一个RS-232转换到JTAG的串行适配器。套件中还有一个目标应用板，上面有对应的MCU。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与串行适配器连接。一条六英寸的扁平电缆将串行适配器和用户的应用板连接起来，连接4个JTAG引脚和VDD及GND。串行适配器从应用板获取其电源。对于不能从目标板上获取足够电流的应用，可以将套件中提供的电源直接连到串行适配器上。

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

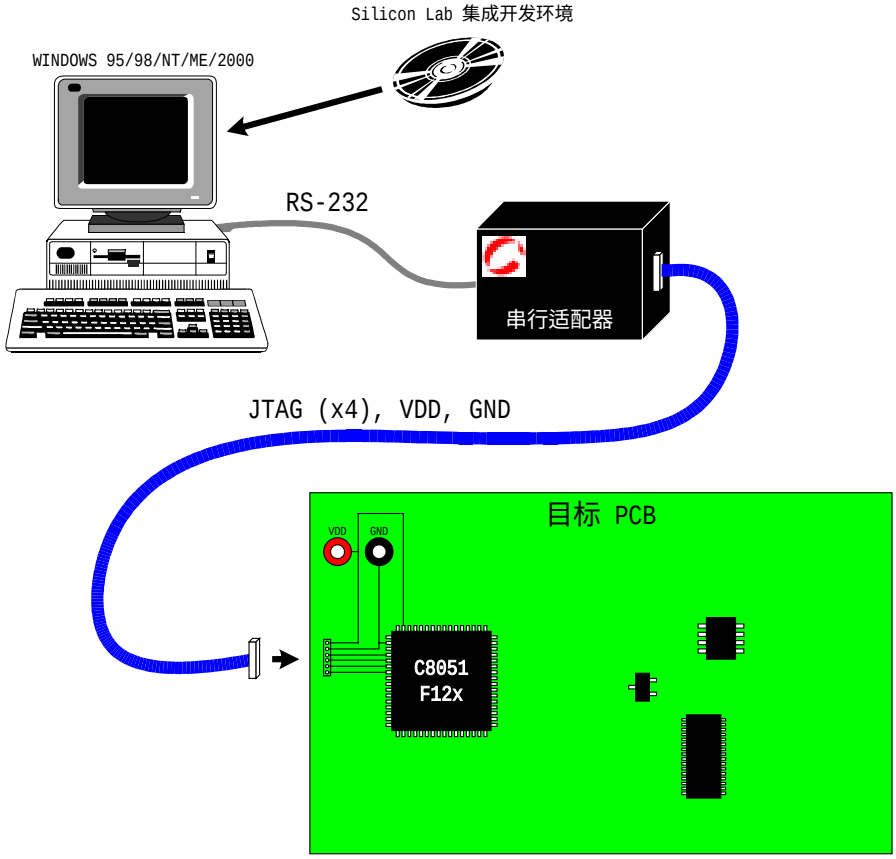


图1.9 调试环境示意图

1.4 16 x16 MAC（乘法和累加）引擎

C8051F120/1/2/3和C8051F130/1/2/3器件包含一个乘法和累加引擎MAC0，可用于加速很多数学运算。MAC0包含一个16 x16的乘法器和一个40位的加法器，它可以在两个SYSCLK周期内完成整数或小数乘法-累加，也可以对带符号的输入值执行乘法运算。一个舍入引擎可以在一个附加的（第三个）SYSCLK周期后提供提供舍入的16位小数结果。MAC0还包含一个1位算术移位器，可以在一个SYSCLK周期内对40位累加器中的内容进行左移或右移。

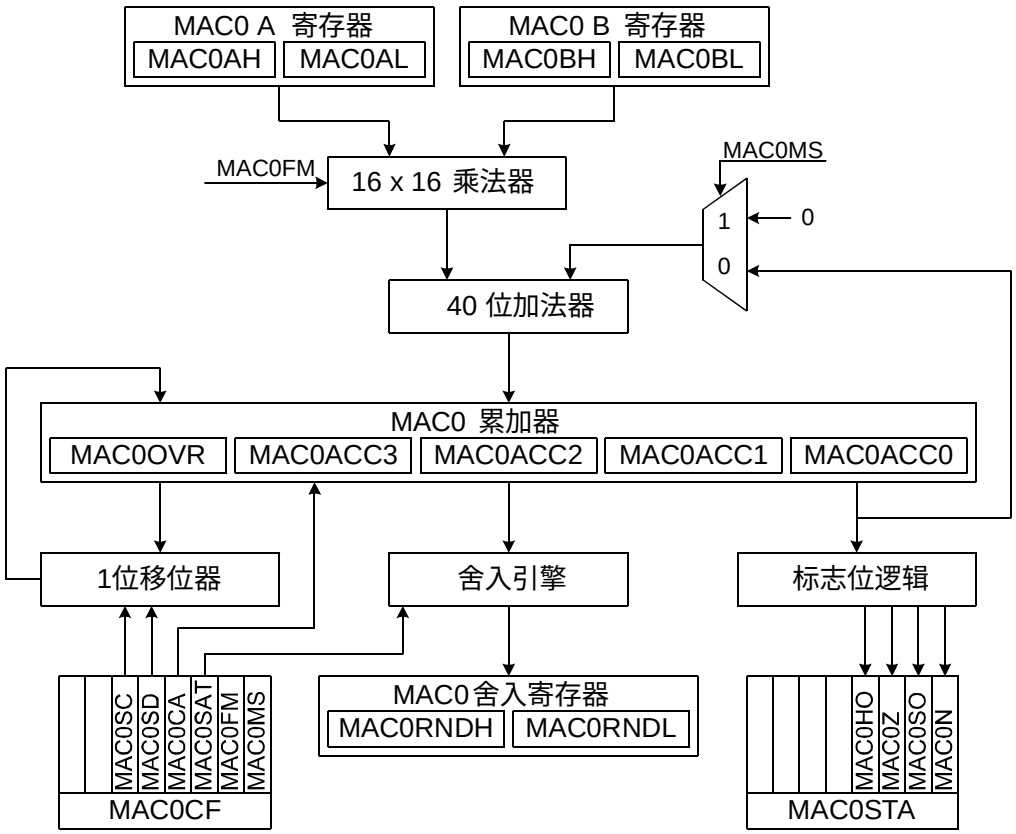


图1.10　MAC0原理框图

1.5 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100脚TQFP封装的器件中有4个附加的端口（4、5、6和7），因此共有64个通用端口I/O。这些端口I/O的工作情况与标准8051相似，但有一些改进。

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

可能最独特的改进是引入了数字交叉开关。这是一个大的数字开关网络，允许将内部数字系统资源映射到P0、P1、P2和P3的端口I/O引脚（见图1.11）。与具有标准复用数字I/O的微控制器不同，这种结构可支持所有的功能组合。

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

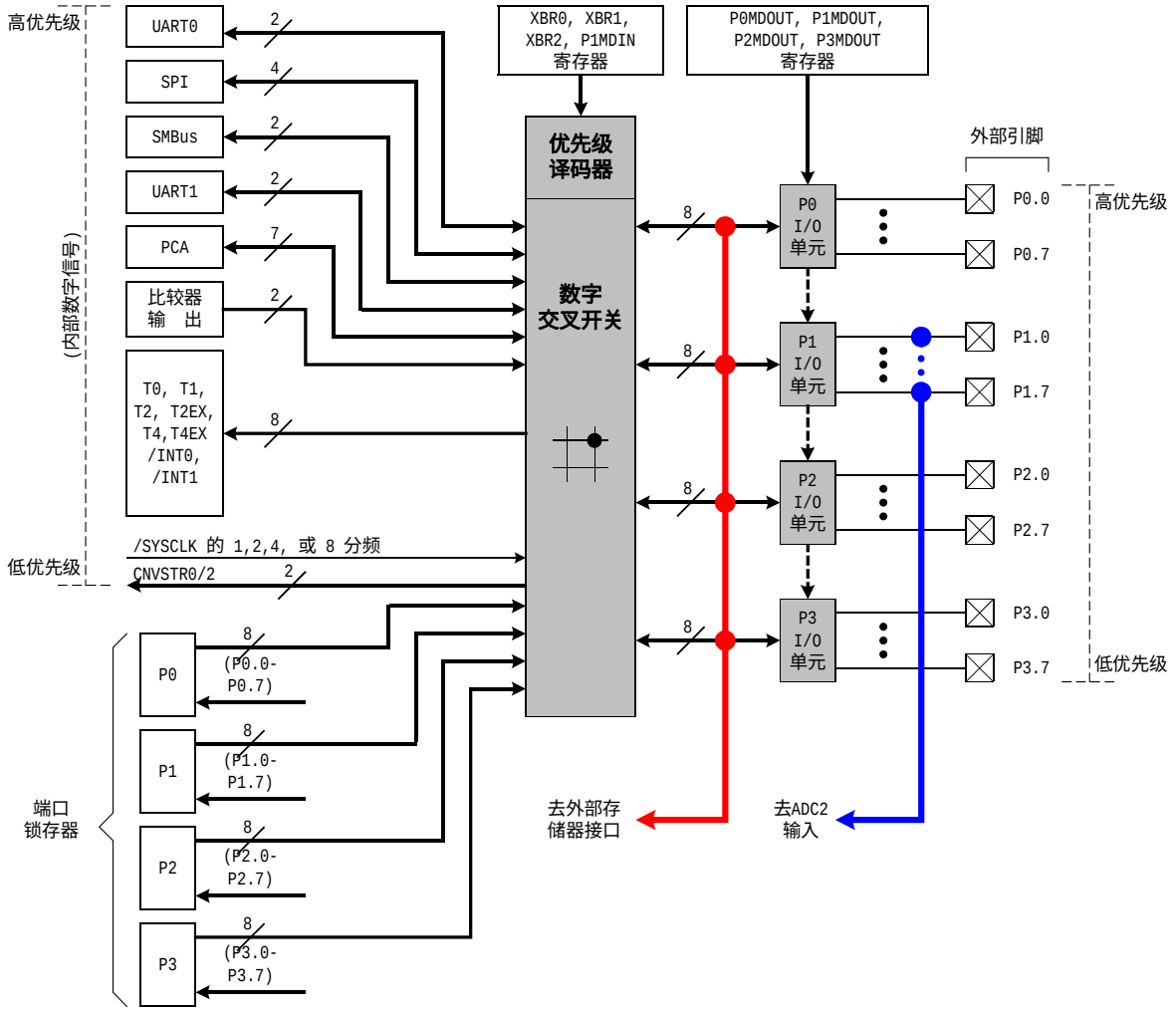


图1.11 数字交叉开关原理框图

1.6 可编程计数器阵列

除了5个16位的通用计数器/定时器之外，C8051F12x和C8051F13x器件中还有一个片内可编程计数器/定时器阵列（PCA）。PCA包括一个专用的16位计数器/定时器时间基准和6个可编程的捕捉/比较模块。时间基准的时钟可以是下面的六个时钟源之一：系统时钟/12、系统时钟/4、定时器0溢出、外部时钟输入（ECI）、系统时钟和外部振荡源/8。

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

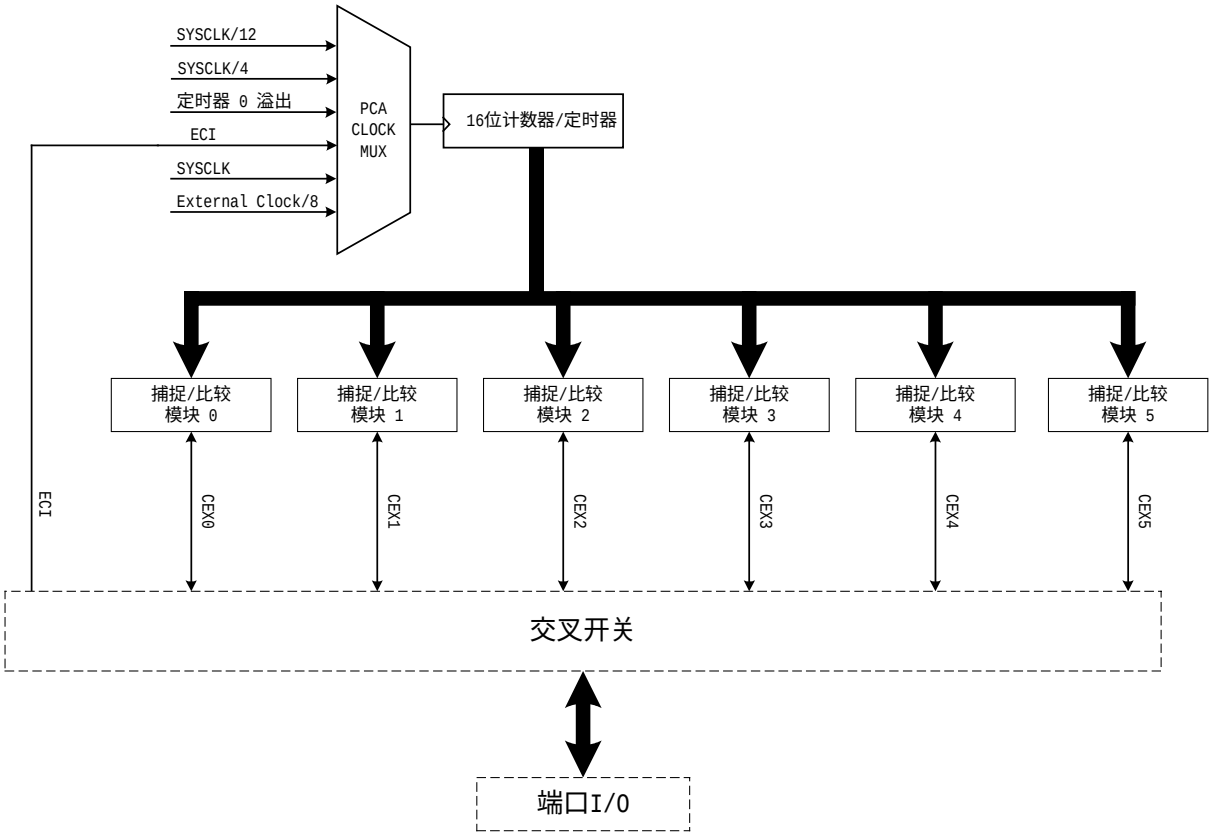


图1.12 PCA原理框图

1.7 串行端口

C8051F12x和C8051F13x系列MCU内部有两个增强型全双工UART、SPI总线和SMBus/I^2C。每种串行总线都完全用硬件实现，都能向CIP-51产生中断，因此需要很少的CPU干预。这些串行总线不“共享”定时器、中断或端口I/O等资源，所以可以使用任何一个或全部同时使用。

1.8 12 或 10 位模/数转换器

所有器件中都有一个片内12或10位SAR ADC（ADC0），一个9通道输入多路选择开关和可编程增益放大器。该ADC工作在100ksps的最大采样速率时可提供真正的12位或10位精度，INL为±1LSB。ADC0的电压基准可以在DAC0输出和一个外部VREF引脚之间选择（仅C8051F12x器件）。对于100脚TQFP器件，ADC0有其专用的VREF0输入引脚；对于64脚TQFP器件，ADC0与8位的ADC2共享VREFA输入引脚。片内15ppm/°C的电压基准可通过VREF输出引脚为其它系统部件或片内ADC产生基准电压。

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

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

A/D转换有4种启动方式：软件命令、定时器2溢出、定时器3溢出和外部信号输入。这种灵活性允许用软件事件、外部硬件信号或周期性的定时器溢出信号触发转换。转换结束由一个状态位指示，或者产生中断（如果中断被使能）。在转换完成后，10或12位转换结果数据字被锁存到两个特殊功能寄存器中。这些数据字可以用软件控制为左对齐或右对齐。

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

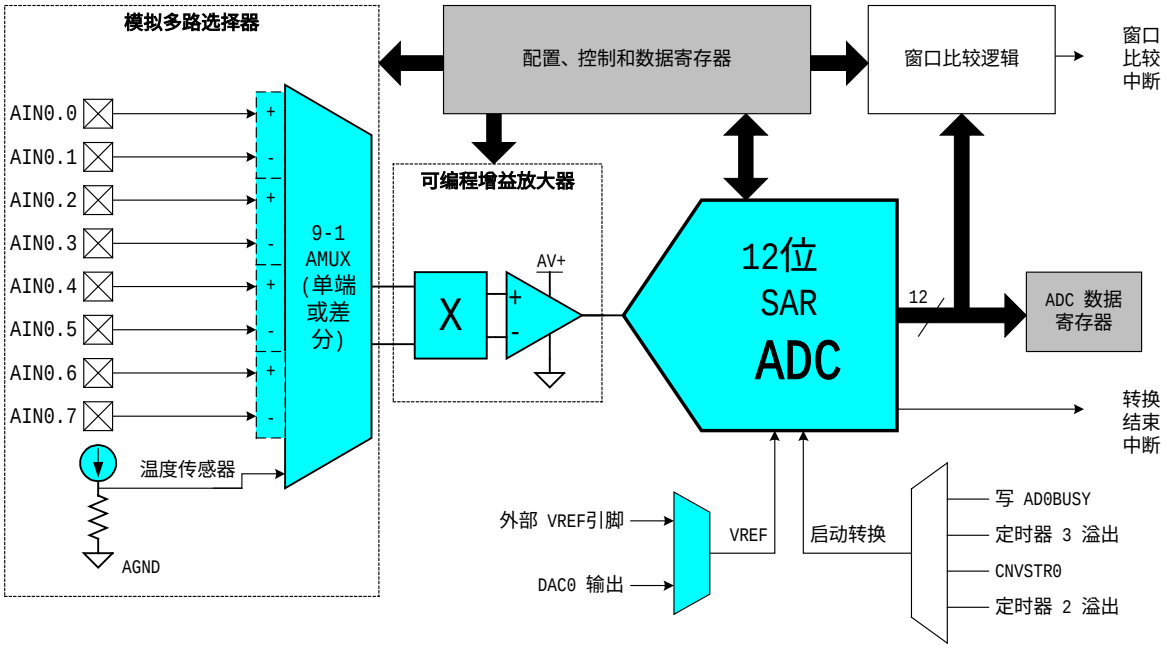


图1.13 12位ADC原理框图

1.9 8 位模/数转换器

C8051F12x系列器件有一个片内8位SAR ADC（ADC2），带有一个8通道输入多路选择器和可编程增益放大器。该ADC工作在500ksps的最大采样速率时可提供真正的8位精度，INL为±1LSB。有8个用于测量的输入端。ADC2完全由CIP-51通过特殊功能寄存器控制。ADC2的电压基准可以在模拟电源电压（AV+）和一个外部VREF引脚之间选择。对于100脚TQFP器件，ADC2有其专用的VREF2输入引脚；对于64脚TQFP器件，ADC2与12/10位的ADC0共享电压基准输入。用户软件可以将ADC2置于关断状态以节省功耗。

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

灵活的转换控制系统允许用软件命令、定时器溢出或外部信号启动ADC2转换。用软件命令可以使ADC2与ADC0同步转换。转换结束由一个状态位指示，或者产生中断（如果中断被使能）。在转换完成后，8位数据字被锁存到一个特殊功能寄存器中。

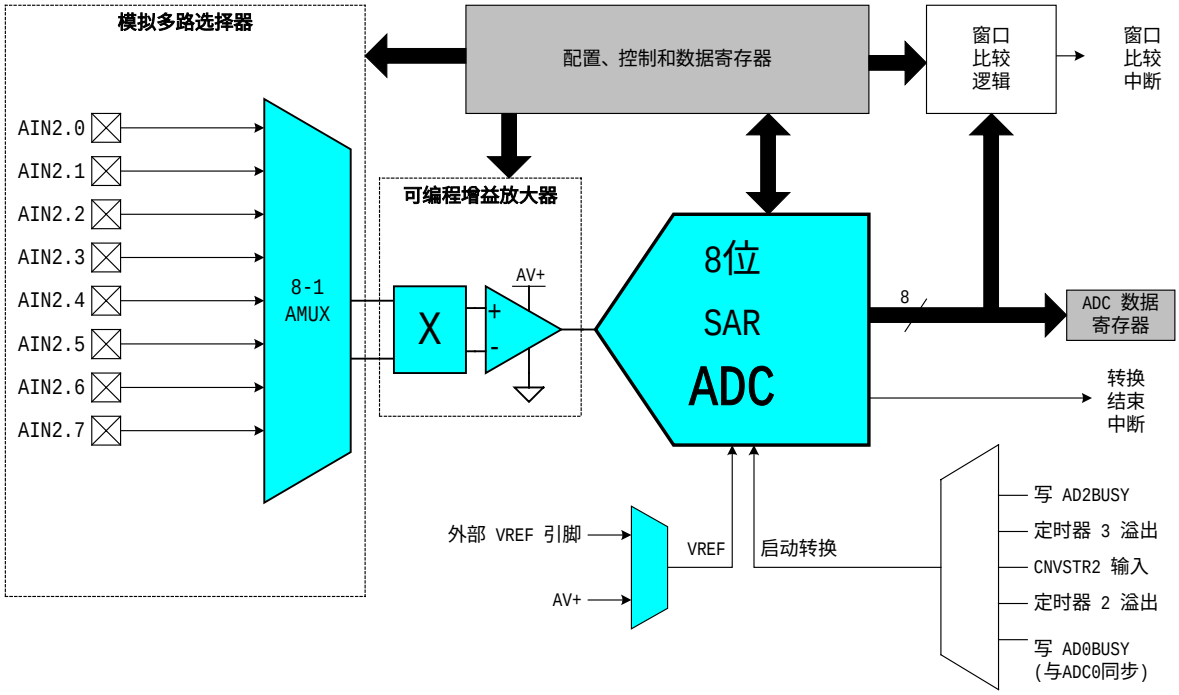


图1.14 8位ADC原理框图

1.10 12 位数/模转换器

C8051F12x系列MCU内部有两个12位数/模转换器（DAC）。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。

DAC为电压输出方式，有灵活的输出更新机制。这一机制允许用软件写和定时器2、定时器3及定时器4的溢出信号更新DAC输出。C8051F120/2/4/6的DAC之电压基准由专用的VREFD输入引脚提供，而C8051F121/3/5/7的DAC之电压基准由器件内部的电压基准提供。DAC在作为比较器的参考电压或为ADC差分输入提供偏移电压时非常有用。

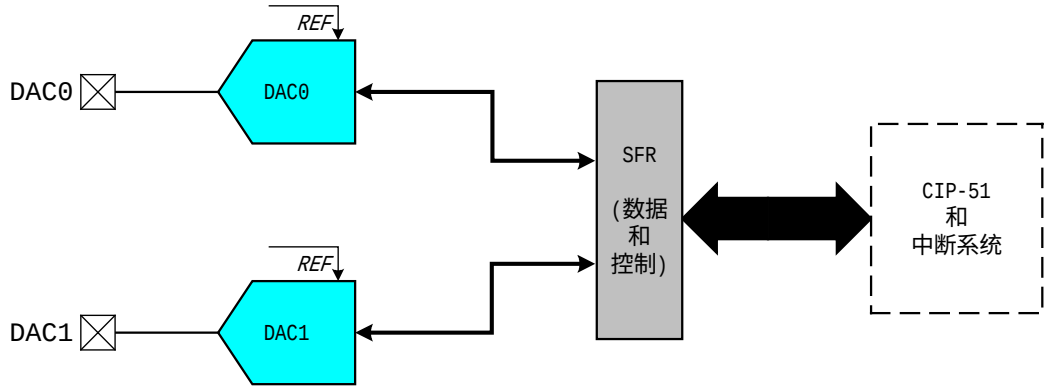


图1.15 DAC原理框图

1.11 比较器

比较器的回差电压可以用软件编程。可以通过调整比较器的响应时间使功耗最小或使速度最块。每个比较器都能在上升沿、下降沿或在两个边沿都产生中断。这些中断能将MCU从休眠方式唤醒。比较器的输出状态可以用软件查询。可通过设置交叉开关将比较器的输出接到端口I/O引脚。

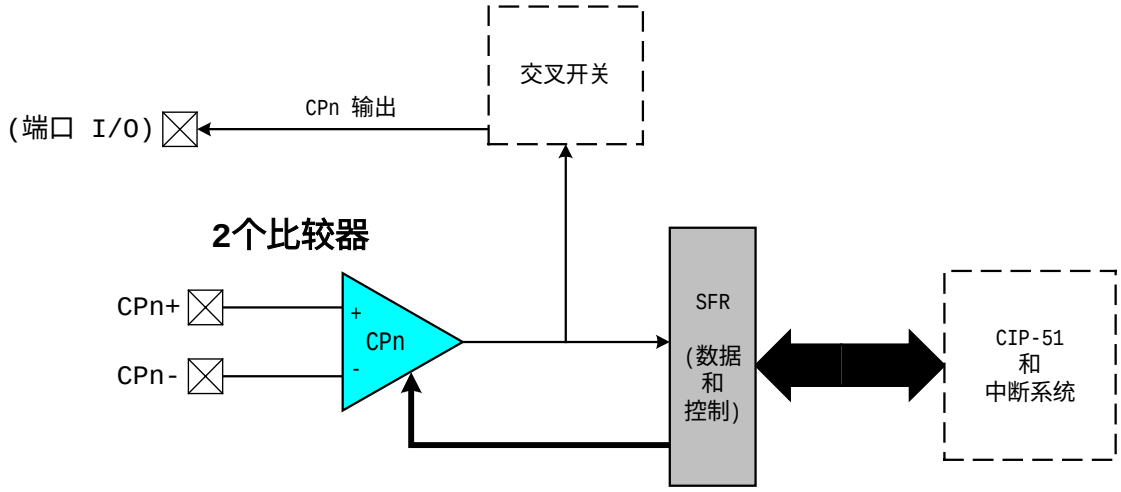


图1.16 比较器原理框图

2. 极限参数

表 2.1 极限参数*

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

*注：超过这些列出的“ 极限参数 ”可能导致器件永久性损坏。

3. 总体直流电气特性

表 3.1 总体直流电气特性（C8051F120/1/2/3 和 C8051F310/1/2/3）

-40℃ ~ 85℃，100MHz 系统时钟，除非特别说明

参　数	条　件	最小值	典型值	最大值	单 位
模拟电源电压（注1）	SYSCLK = 0 ~ 50 MHz SYSCLK > 50 MHz	2.7 3.0	3.0 3.3	3.6 3.6	V V
模拟电源电流	内部REF、ADC、DAC、比较器都活动		1.7		mA
模拟电源电流（模拟子系统不工作）	内部REF、ADC、DAC、比较器都被禁止；振荡器被禁止		0.2		µA
模拟与数字电源电压之差（|VDD-AV+|）				0.5	V
数字电源电压	SYSCLK = 0 ~ 50 MHz SYSCLK > 50 MHz	2.7 3.0	3.0 3.3	3.6 3.6	V V
数字电源电流 CPU处于活动状态	VDD=3.0V, CLK=100MHz VDD=3.0V, CLK=50MHz VDD=3.0V, CLK= 1MHz VDD=3.0V, CLK=32kHz		65 35 1 33		mA mA mA µA
数字电源电流 CPU不活动（不访问FLASH）	VDD=3.0V, CLK=100MHz VDD=3.0V, CLK=25MHz VDD=3.0V, CLK= 1MHz VDD=3.0V, CLK=32kHz		40 20 0.6 15		mA mA mA µA
数字电源电流（停机方式）	振荡器不运行		0.4		µA
数字电源（RAM数据保持电压）			1.5		V
SYSCLK（系统时钟）（注2和注3）	VDD, AV+ = 2.7V ~ 3.6V VDD, AV+ = 3.0V ~ 3.6V	0 0		50 100	MHz MHz
额定工作温度范围		-40		+85	℃

注 1: 模拟电源 AV+必须大于 1V 才能使 VDD 监视器工作。

注 2: SYSCLK 是器件内部时钟。运行速度超过 30 MHz 时，SYSCLK 必须通过锁相环（PLL）获得。

注 3: SYSCLK 至少应为 32KHz 才能使调试电路工作。

表 3.2 总体直流电气特性（C8051F124/5/6/7）

-40℃ ~ 85℃，50MHz 系统时钟，除非特别说明

参　数	条　件	最小值	典型值	最大值	单 位
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、比较器都活动		1.7		mA
模拟电源电流（模拟子系统不工作）	内部REF、ADC、DAC、比较器都被禁止；振荡器被禁止		0.2		μA
模拟与数字电源电压之差（|VDD-AV+|）				0.5	V
数字电源电压		2.7	3.0	3.6	V V
数字电源电流 CPU处于活动状态	VDD=3.0V, CLK=50MHz VDD=3.0V, CLK= 1MHz VDD=3.0V, CLK=32kHz		35 1 33		mA mA μA
数字电源电流 CPU不活动（不访问FLASH）	VDD=3.0V, CLK=25MHz VDD=3.0V, CLK= 1MHz VDD=3.0V, CLK=32kHz		27 0.4 15		mA mA μA
数字电源电流（停机方式）	振荡器不运行		0.4		μA
数字电源（RAM数据保持电压）			1.5		V
SYSCLK（系统时钟）（注2和注3）	VDD, AV+ = 2.7V ~ 3.6V	0		50	MHz
额定工作温度范围		-40		+85	℃

注 1: 模拟电源 AV+必须大于 1V 才能使 VDD 监视器工作。

注 2: SYSCLK 是器件内部时钟。运行速度超过 25 MHz 时，SYSCLK 必须通过锁相环（PLL）获得。

注 3: SYSCLK 至少应为 32KHz 才能使调试电路工作。

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
	F120 F122 F124 F126	F121 F123 F125 F127	F130 F132	F131 F133		
VDD	37,64,90	24,41,57	37,64,90	24,41,57		数字电源。必须接 +2.7V ~ +3.6V。
DGND	38,63,89	25,40,56	38,63,89	25,40,56		数字地，必须接地
AV+	11,14	6	11,14	6		模拟电源。必须接 +2.7V ~ +3.6V。
AGND	10,13	5	10,13	5		模拟地，必须接地
TMS	1	58	1	58	数字输入	JTAG 测试模式选择，带内部上拉
TCK	2	59	2	59	数字输入	JTAG 测试时钟，带内部上拉
TDI	3	60	3	60	数字输入	JTAG 测试数据输入，带内部上拉。TDI 在 TCK 上升沿被锁存。
TDO	4	61	4	61	数字输出	JTAG 测试数据输出，带内部上拉。数据在 TCK 的下降沿从 TDO 引脚输出。TDO 输出是一个三态驱动器。
/RST	5	62	5	62	数字 I/O	器件复位。内部 VDD 监视器的漏极开路输出。当 VDD < V_{RST} 并且 MONEN 为高时被驱动为低电平。一个外部源可以通过将该引脚置为低电平启动一次系统复位。
XTAL1	26	17	26	17	模拟输入	晶体输入。该引脚为晶体或陶瓷谐振器的内部振荡器电路的反馈输入。为了得到一个精确的内部时钟，可以在 XTAL1 和 XTAL2 之间接上一个晶体或陶瓷谐振器。如果被一个外部 CMOS 时钟驱动，则该引脚提供系统时钟。
XTAL2	27	18	27	18	模拟输出	晶体输出。该引脚是晶体或陶瓷谐振器的激励驱动器。
MONEN	28	19	28	19	数字输入	VDD 监视器使能。该引脚接高电平时允许内部 VDD 监视器工作，当 VDD < V_{RST} 时强制系统复位。该引脚接低电平时内部 VDD 监视器被禁止。 **该引脚必须接高电平或低电平。**

表 4.1 引脚定义（续）

引脚名称	引脚号				类 型	说 明
	F120 F122 F124 F126	F121 F123 F125 F127	F130 F132	F131 F133		
VREF	12	7	12	7	模拟 I/O	带隙电压基准输出（所有器件）。 DAC 电压基准输入（仅 F121/3/5/7）。
VREFA		8		8	模拟输入	ADC0 和 ADC2 的电压基准输入。
VREF0	16		16		模拟输入	ADC0 的电压基准输入。
VREF2	17				模拟输入	ADC2 的电压基准输入。
VRED	15				模拟输入	DAC 的电压基准输入。
AIN0.0	18	9	18	9	模拟输入	ADC0 输入通道 0（详见 ADC0 说明）
AIN0.1	19	10	19	10	模拟输入	ADC0 输入通道 1（详见 ADC0 说明）
AIN0.2	20	11	20	11	模拟输入	ADC0 输入通道 2（详见 ADC0 说明）
AIN0.3	21	12	21	12	模拟输入	ADC0 输入通道 3（详见 ADC0 说明）
AIN0.4	22	13	22	13	模拟输入	ADC0 输入通道 4（详见 ADC0 说明）
AIN0.5	23	14	23	14	模拟输入	ADC0 输入通道 5（详见 ADC0 说明）
AIN0.6	24	15	24	15	模拟输入	ADC0 输入通道 6（详见 ADC0 说明）
AIN0.7	25	16	25	16	模拟输入	ADC0 输入通道 7（详见 ADC0 说明）
CP0+	9	4	9	4	模拟输入	比较器 0 的同相输入端
CP0-	8	3	8	3	模拟输入	比较器 0 的反相输入端
CP1+	7	2	7	2	模拟输入	比较器 1 的同相输入端
CP1-	6	1	6	1	模拟输入	比较器 1 的反相输入端
DAC0	100	64			模拟输出	数模转换器 0 的电压输出。（见 DAC 说明）
DAC1	99	63			模拟输出	数模转换器 1 的电压输出。（见 DAC 说明）
P0.0	62	55			数字 I/O	P0.0。详见端口输入/输出部分。
P0.1	61	54			数字 I/O	P0.1。详见端口输入/输出部分。
P0.2	60	53			数字 I/O	P0.2。详见端口输入/输出部分。
P0.3	59	52			数字 I/O	P0.3。详见端口输入/输出部分。
P0.4	58	51			数字 I/O	P0.4。详见端口输入/输出部分。
ALE/P0.5	57	50			数字 I/O	外部存储器地址总线 ALE 选通（复用方式）。 P0.5。详见端口输入/输出部分。
/RD/P0.6	56	49			数字 I/O	外部存储器接口的/RD 选通。 P0.6。详见端口输入/输出部分。
/WR/P0.7	55	48			数字 I/O	外部存储器接口的/WR 选通。 P0.7。详见端口输入/输出部分。

表 4.1 引脚定义（续）

引脚名称	引脚号				类 型	说 明
	F120 F122 F124 F126	F121 F123 F125 F127	F130 F132	F131 F133		
AIN2.0/A8/P1.0	36	29	36	29	模拟输入	ADC2 输入通道 0（详见 ADC2 说明）。 外部存储器地址总线位 8（非复用方式）。 P1.0。详见端口输入/输出部分。
AIN2.1/A9/P1.1	35	28	35	28	模拟输入 数字 I/O	P1.1。详见端口输入/输出部分。
AIN2.2/A10/P1.2	34	27	34	27	模拟输入 数字 I/O	P1.2。详见端口输入/输出部分。
AIN2.3/A11/P1.3	33	26	33	26	模拟输入 数字 I/O	P1.3。详见端口输入/输出部分。
AIN2.4/A12/P1.4	32	23	32	23	模拟输入 数字 I/O	P1.4。详见端口输入/输出部分。
AIN2.5/A13/P1.5	31	22	31	22	模拟输入 数字 I/O	P1.5。详见端口输入/输出部分。
AIN2.6/A14/P1.6	30	21	30	21	模拟输入 数字 I/O	P1.6。详见端口输入/输出部分。
AIN2.7/A15/P1.7	29	20	29	20	模拟输入 数字 I/O	P1.7。详见端口输入/输出部分。
A8m/A0/P2.0	46	37	46	37	数字 I/O	外部存储器地址总线位 8（复用方式）。 外部存储器地址总线位 0（非复用方式）。 P2.0。详见端口输入/输出部分。
A9m/A1/P2.1	45	36	45	36	数字 I/O	P2.1。详见端口输入/输出部分。
A10m/A2/P2.2	44	35	44	35	数字 I/O	P2.2。详见端口输入/输出部分。
A11m/A3/P2.3	43	34	43	34	数字 I/O	P2.3。详见端口输入/输出部分。
A12m/A4/P2.4	42	33	42	33	数字 I/O	P2.4。详见端口输入/输出部分。
A13m/A5/P2.5	41	32	41	32	数字 I/O	P2.5。详见端口输入/输出部分。
A14m/A6/P2.6	40	31	40	31	数字 I/O	P2.6。详见端口输入/输出部分。
A15m/A7/P2.7	39	30	39	30	数字 I/O	P2.7。详见端口输入/输出部分。

表 4.1 引脚定义（续）

引脚名称	引脚号				类型	说 明
	F120 F122 F124 F126	F121 F123 F125 F127	F130 F132	F131 F133		
AD0/D0/P3.0	54	47	54	47	数字 I/O	外部存储器地址/数据总线位 0（复用方式）。 外部存储器数据总线位 0（非复用方式）。 P3.0。详见端口输入/输出部分。
AD1/D1/P3.1	53	46	53	46	数字 I/O	P3.1。详见端口输入/输出部分。
AD2/D2/P3.2	52	45	52	45	数字 I/O	P3.2。详见端口输入/输出部分。
AD3/D3/P3.3	51	44	51	44	数字 I/O	P3.3。详见端口输入/输出部分。
AD4/D4/P3.4	50	43	50	43	数字 I/O	P3.4。详见端口输入/输出部分。
AD5/D5/P3.5	49	42	49	42	数字 I/O	P3.5。详见端口输入/输出部分。
AD6/D6/P3.6	48	39	48	39	数字 I/O	P3.6。详见端口输入/输出部分。
AD7/D7/P3.7	47	38	47	38	数字 I/O	P3.7。详见端口输入/输出部分。
P4.0	98		98		数字 I/O	P4.0。详见端口输入/输出部分。
P4.1	97		97		数字 I/O	P4.1。详见端口输入/输出部分。
P4.2	96		96		数字 I/O	P4.2。详见端口输入/输出部分。
P4.3	95		95		数字 I/O	P4.3。详见端口输入/输出部分。
P4.4	94		94		数字 I/O	P4.4。详见端口输入/输出部分。
ALE/P4.5	93		93		数字 I/O	外部存储器地址总线 ALE 选通（复用方式）。 P4.5。详见端口输入/输出部分。
/RD/P4.6	92		92		数字 I/O	外部存储器接口的/RD 选通。 P4.6。详见端口输入/输出部分。
/WR/P4.7	91		91		数字 I/O	外部存储器接口的/WR 选通。 P4.7。详见端口输入/输出部分。

表 4.1 引脚定义（续）

引脚名称	引脚号				类型	说 明
	F120 F122 F124 F126	F121 F123 F125 F127	F130 F132	F131 F133		
A8/P5.0	88		88		数字 I/O	外部存储器地址总线位 8（非复用方式）。 P5.0。详见端口输入/输出部分。
A9/P5.1	87		87		数字 I/O	P5.1。详见端口输入/输出部分。
A10/P5.2	86		86		数字 I/O	P5.2。详见端口输入/输出部分。
A11/P5.3	85		85		数字 I/O	P5.3。详见端口输入/输出部分。
A12/P5.4	84		84		数字 I/O	P5.4。详见端口输入/输出部分。
A13/P5.5	83		83		数字 I/O	P5.5。详见端口输入/输出部分。
A14/P5.6	82		82		数字 I/O	P5.6。详见端口输入/输出部分。
A15/P5.7	81		81		数字 I/O	P5.7。详见端口输入/输出部分。
A8m/A0/P6.0	80		80		数字 I/O	外部存储器地址总线位 8（复用方式）。 外部存储器地址总线位 0（非复用方式）。 P6.0。详见端口输入/输出部分。
A9m/A1/P6.1	79		79		数字 I/O	P6.1。详见端口输入/输出部分。
A10m/A2/P6.2	78		78		数字 I/O	P6.2。详见端口输入/输出部分。
A11m/A3/P6.3	77		77		数字 I/O	P6.3。详见端口输入/输出部分。
A12m/A4/P6.4	76		76		数字 I/O	P6.4。详见端口输入/输出部分。
A13m/A5/P6.5	75		75		数字 I/O	P6.5。详见端口输入/输出部分。
A14m/A6/P6.6	74		74		数字 I/O	P6.6。详见端口输入/输出部分。
A15m/A7/P6.7	73		73		数字 I/O	P6.7。详见端口输入/输出部分。

表 4.1 引脚定义（续）

引脚名称	引脚号				类型	说 明
	F120 F122 F124 F126	F121 F123 F125 F127	F130 F132	F131 F133		
AD0/D0/P7.0	72		72		数字 I/O	外部存储器地址/数据总线位 0（复用方式）。 外部存储器数据总线位 0（非复用方式）。 P7.0。详见端口输入/输出部分。
AD1/D1/P7.1	71		71		数字 I/O	P7.1。详见端口输入/输出部分。
AD2/D2/P7.2	70		70		数字 I/O	P7.2。详见端口输入/输出部分。
AD3/D3/P7.3	69		69		数字 I/O	P7.3。详见端口输入/输出部分。
AD4/D4/P7.4	68		68		数字 I/O	P7.4。详见端口输入/输出部分。
AD5/D5/P7.5	67		67		数字 I/O	P7.5。详见端口输入/输出部分。
AD6/D6/P7.6	66		66		数字 I/O	P7.6。详见端口输入/输出部分。
AD7/D7/P7.7	65		65		数字 I/O	P7.7。详见端口输入/输出部分。
NC			15 17 99 100	63 64		未连接

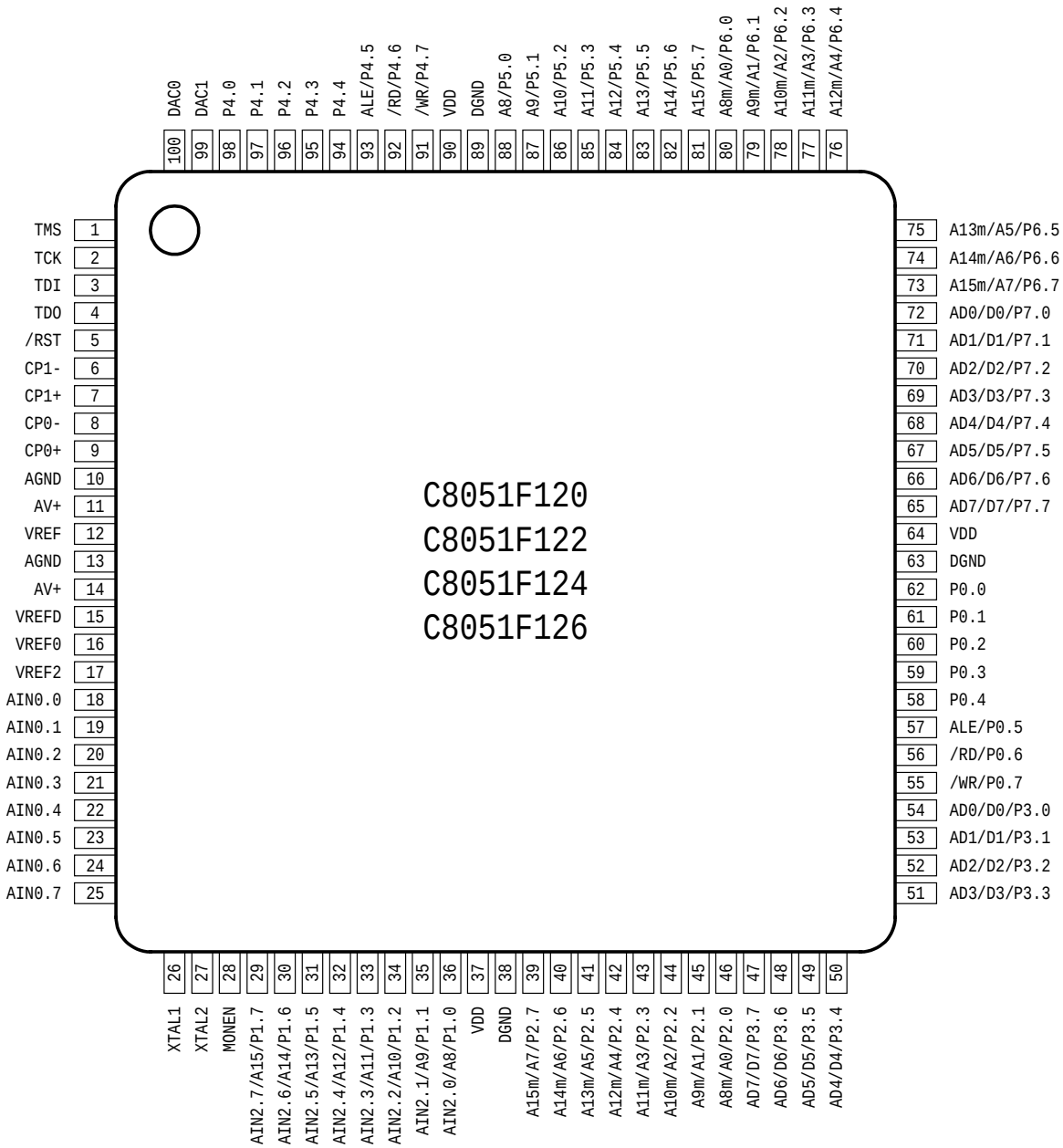


图 4.1 C8051F120/2/4/6 引脚图（TQFP-100）

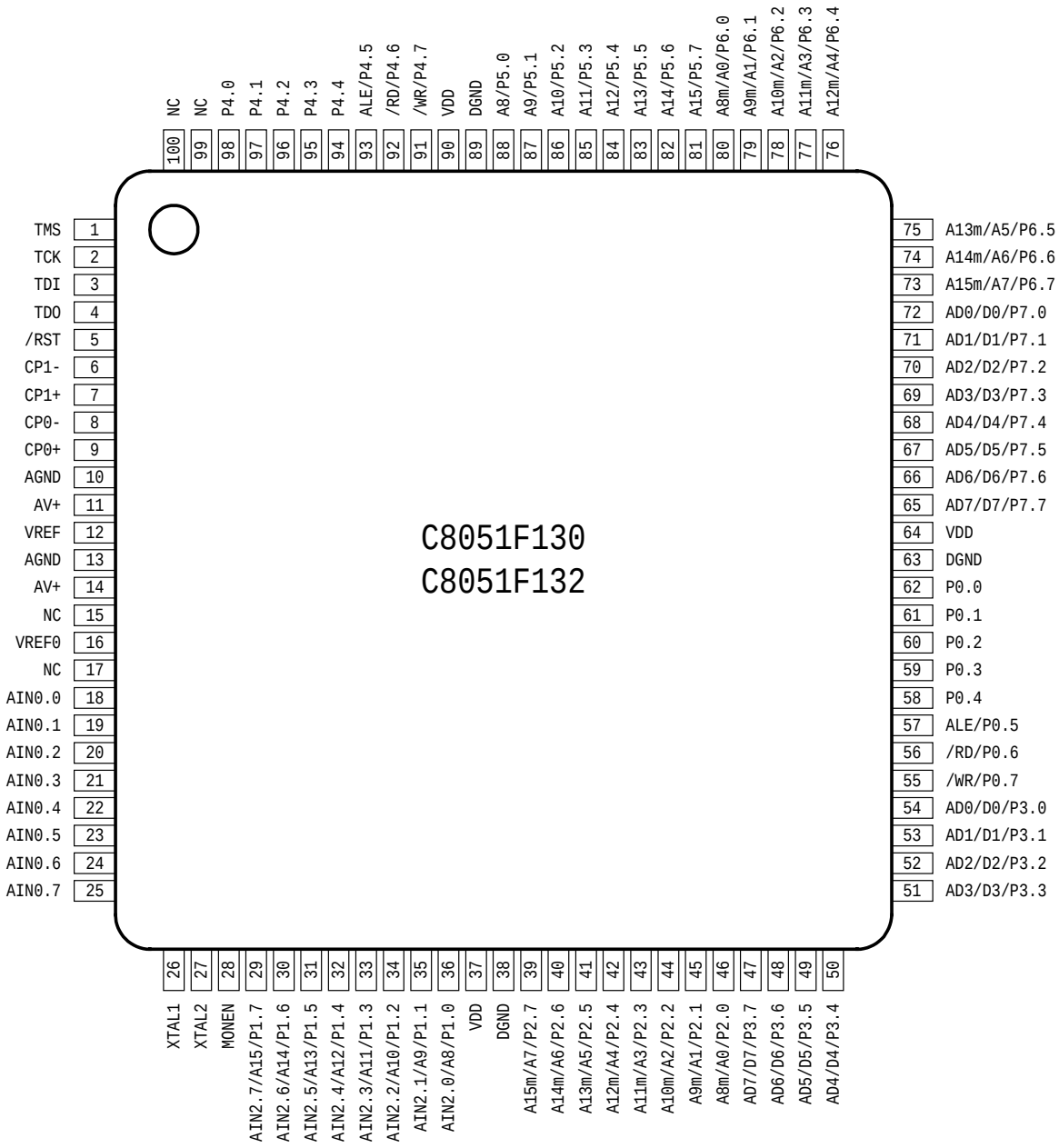
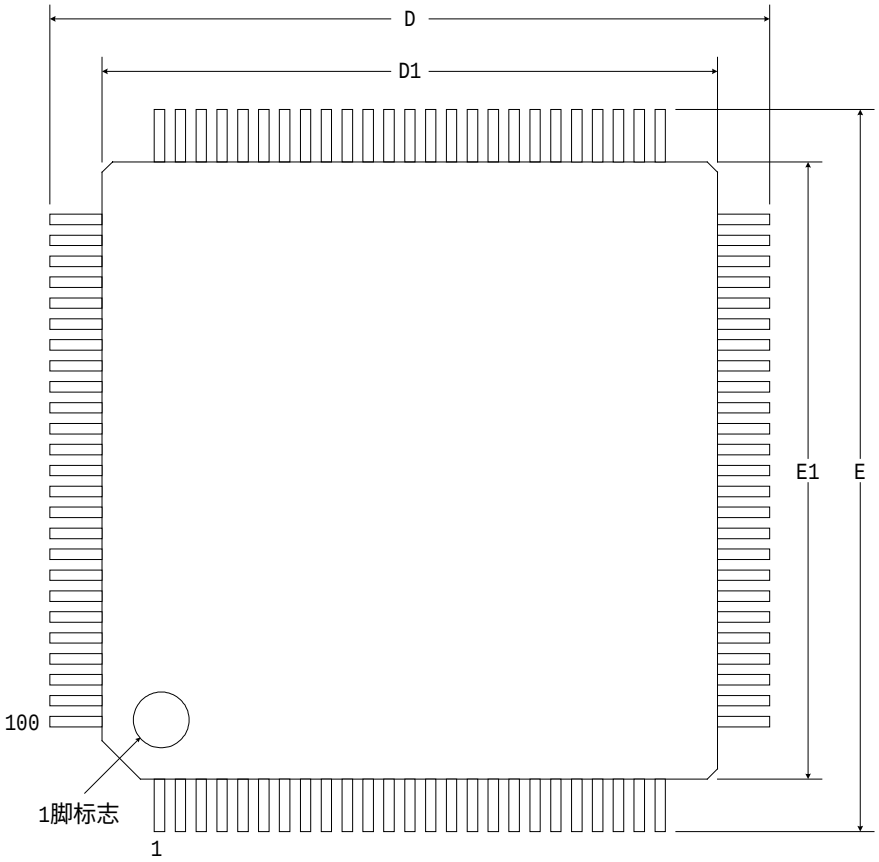


图 4.2 C8051F130/2 引脚图（TQFP-100）



	MIN (mm)	NOM (mm)	MAX (mm)
A	-	-	1.20
A1	0.05	-	0.15
A2	0.95	1.00	1.05
b	0.17	0.22	0.27
D	-	16.00	-
D1	-	14.00	-
e	-	0.50	-
E	-	16.00	-
E1	-	14.00	-
L	0.45	0.60	0.75

图 4.3　TQFP-100 封装图

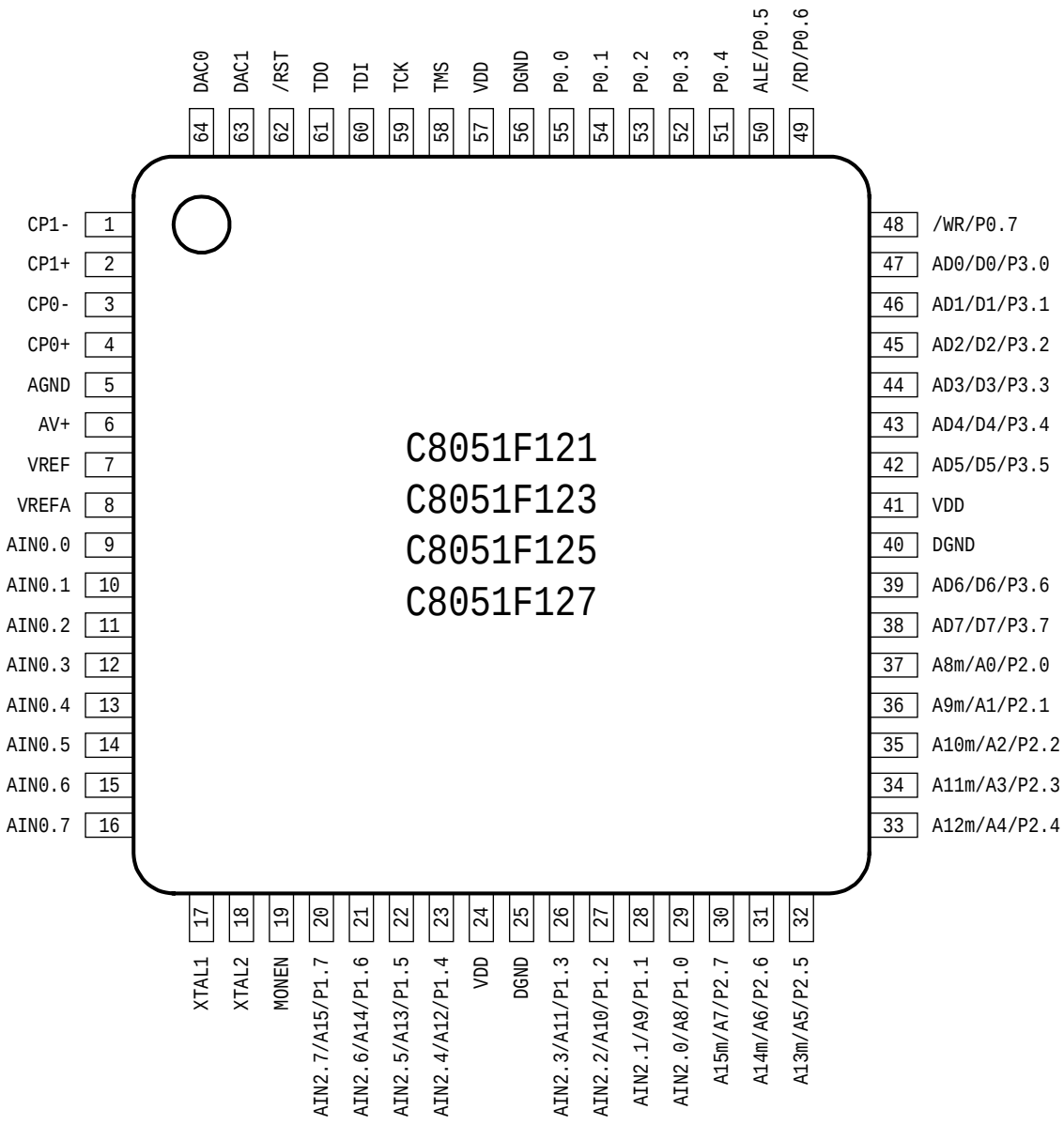


图 4.4 C8051F121/3/5/7 引脚图（TQFP-64）

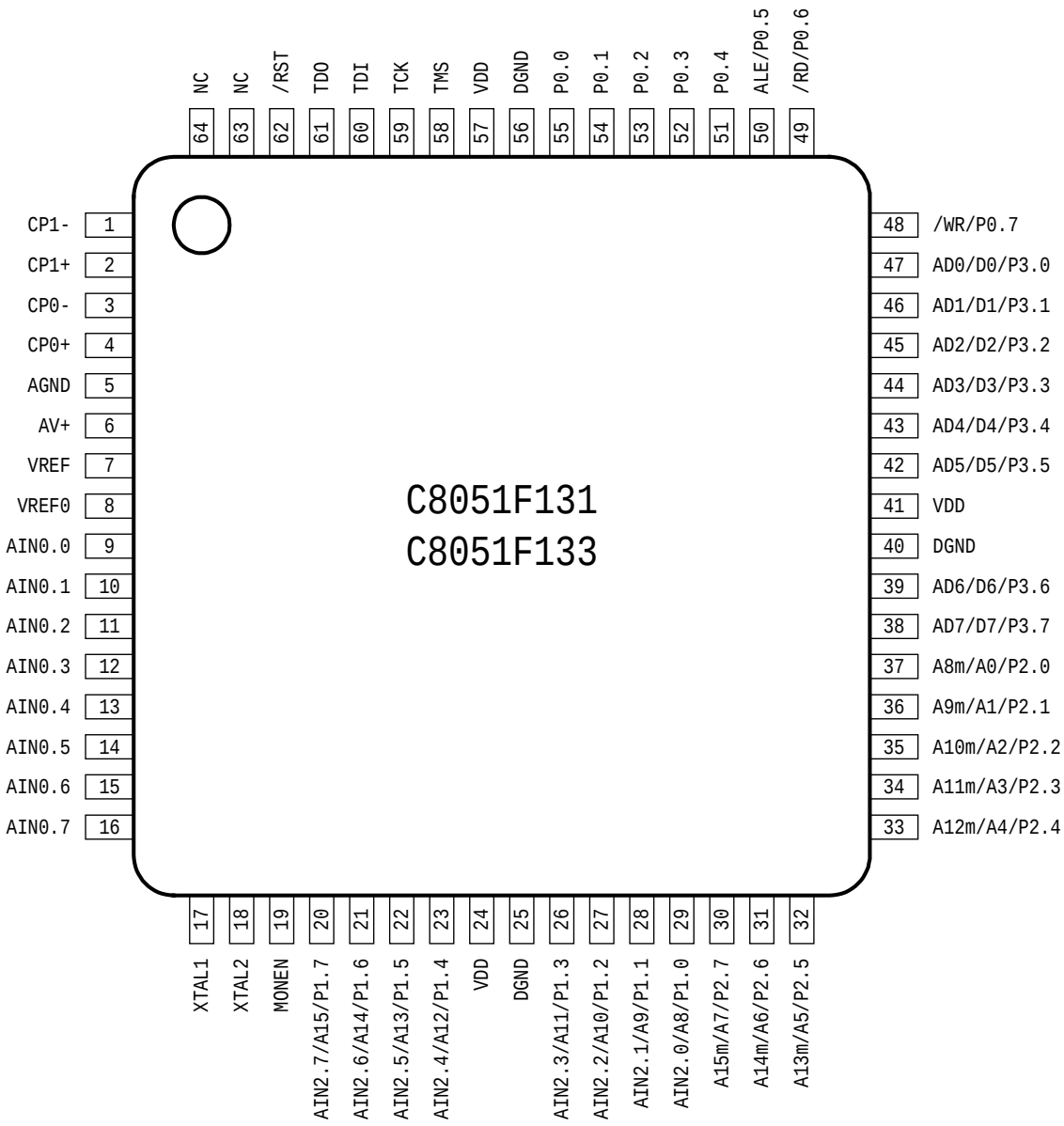
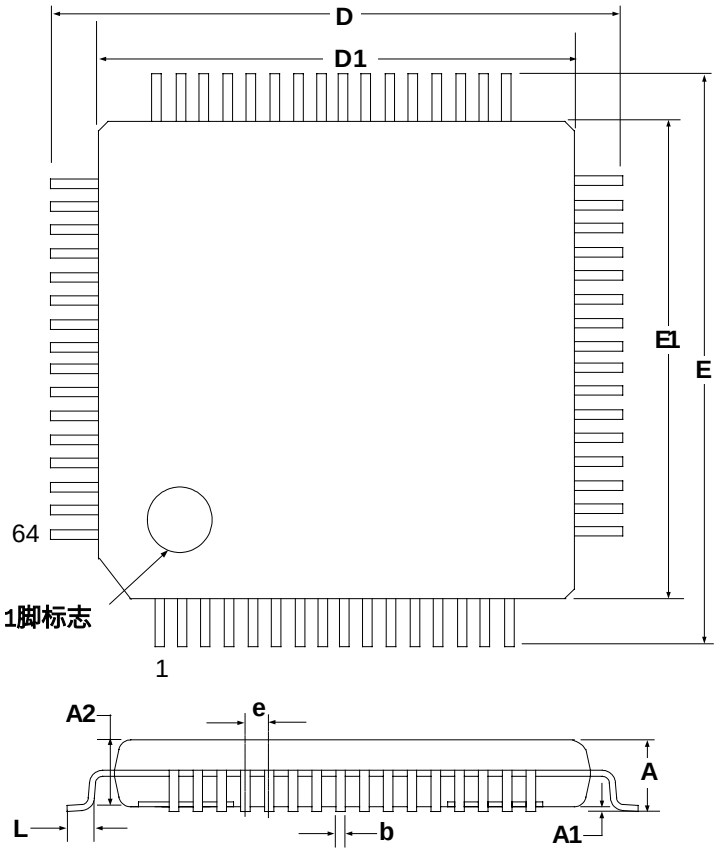


图 4.5 C8051F121/3/5/7 引脚图（TQFP-64）



	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	-
L	0.45	0.60	0.75

图 4.6 TQFP-64 封装图

5. ADC0（12 位，仅 C8051F120/1/4/5）

C8051F120/1/4/5 的 ADC0 子系统包括一个 9 通道的可编程模拟多路选择器（AMUX0），一个可编程增益放大器（PGA0）和一个 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）。AMUX0、PGA0、数据转换方式及窗口检测器都可用软件通过图 5.1 所示的特殊功能寄存器来控制。ADC0 所使用的电压基准按“9. 电压基准所述选择。只有当 ADC0 控制寄存器中的 AD0EN 位被置‘1’时 ADC0 子系统（ADC0、跟踪保持器和 PGA0）才被允许工作。当 AD0EN 位为‘0’时，ADC0 子系统处于低功耗关断方式。

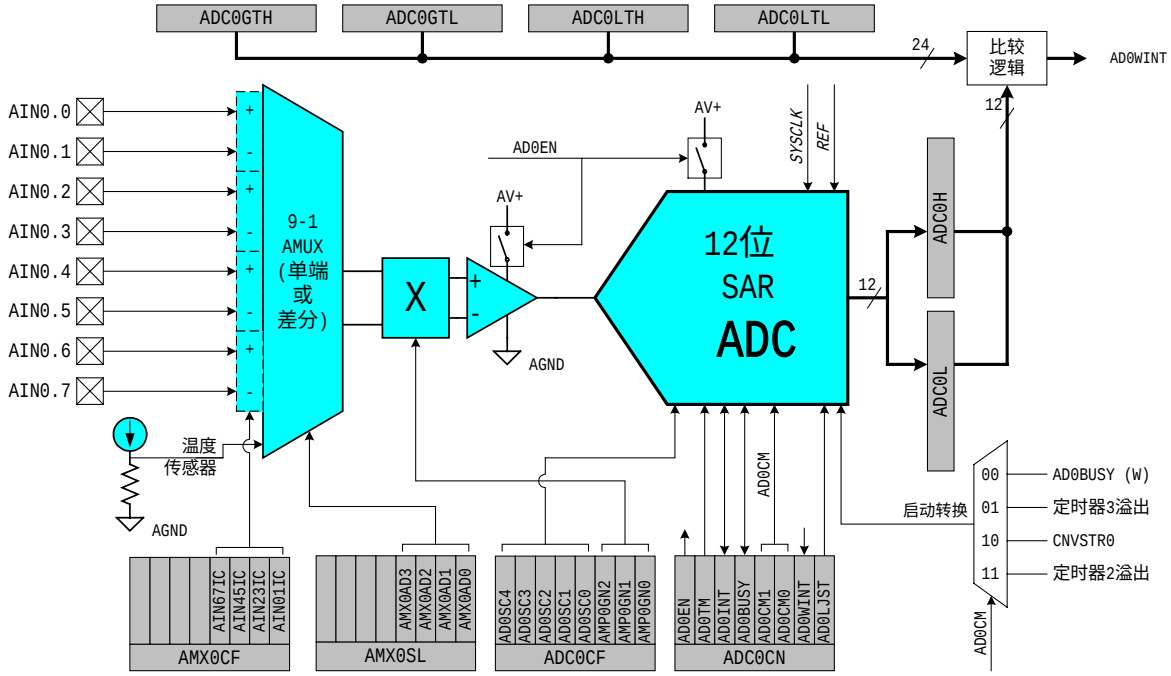


图 5.1 12 位 ADC0 功能框图

5.1 模拟多路开关和 PGA

AMUX 中的 8 个通道用于外部测量，而第九通道在内部被接到片内温度传感器（温度传输函数示于图 5.2）。注意，PGA0 的增益对温度传感器也起作用。可以将 AMUX 输入对编程为工作在差分或单端方式。这就允许用户对每个通道选择最佳的测量技术，甚至可以在测量过程中改变方式。在系统复位后 AMUX 的默认方式为单端输入。有两个与 AMUX 相关的寄存器：通道选择寄存器 AMX0SL（图 5.6）和配置寄存器 AMX0CF（图 5.5）。图 5.6 中的表给出了每种配置下各通道的功能。PGA 对 AMUX 输出信号的放大倍数由 ADC0 配置寄存器 ADC0CF（图 5.7）中的 AMP0GN2-0 确定。PGA 增益可以用软件编程为 0.5、1、2、4、8 或 16，复位后的默认增益为 1。

温度传感器的传输函数示于图 5.2。当温度传感器被选中（用 AMX0SL 中的 AMX0AD3-0）时，其输出电压（V_{TEMP}）是 PGA 的输入；PGA 对该电压的放大倍数由用户编程的 PGA 设置值决定。传输函数的斜率和偏移值见表 5.1

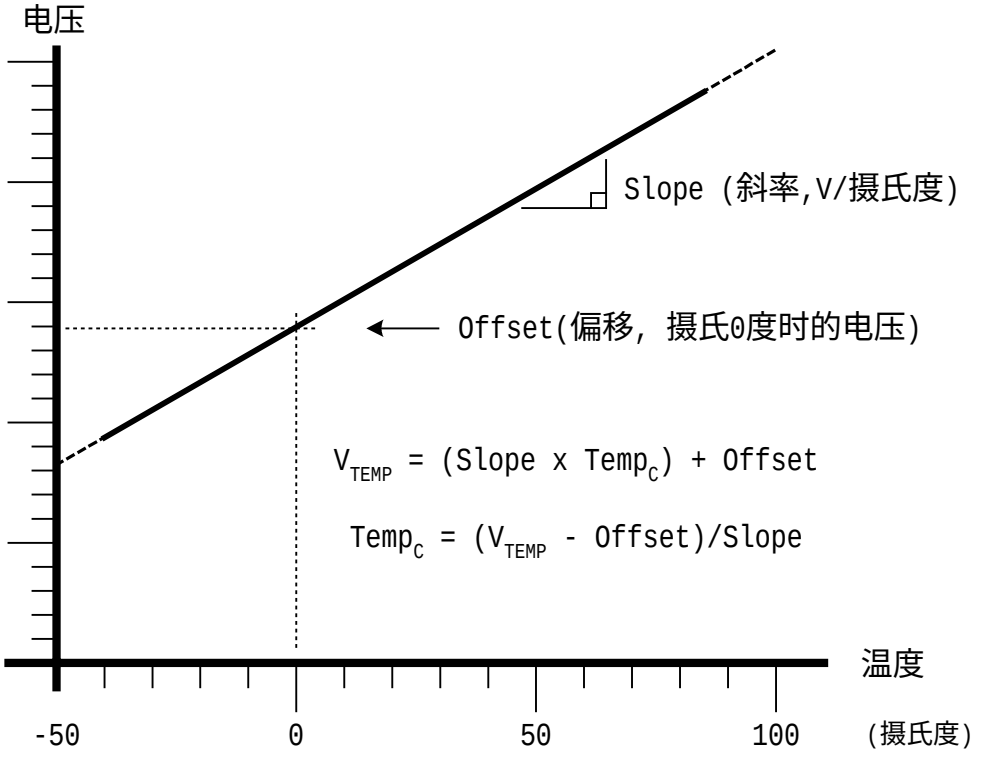


图 5.2 温度传感器传输函数

5.2 ADC 的工作方式

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

5.2.1 启动转换

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

1. 向 ADC0CN 的 AD0BUSY 位写 1；
2. 定时器 3 溢出（即定时的连续转换）；
3. 外部 ADC 转换启动信号的上升沿，CNVSTR0；
4. 定时器 2 溢出（即定时的连续转换）。

AD0BUSY 位在转换期间被置‘1’，转换结束后复‘0’。AD0BUSY 位的下降沿触发一个中断（当被允许时）并将中断标志 AD0INT（ADC0CN.5）置‘1’。转换数据被保存在 ADC 数据字的 MSB 和 LSB 寄存器：ADC0H 和 ADC0L。转换数据在寄存器对 ADC0H:ADC0L 中的存储方式可以是左对齐或右对齐（见图 5.11 中的例子），由 ADC0CN 寄存器中 AD0LJST 位的编程状态决定。

当通过向 AD0BUSY 写‘1’启动数据转换时，应查询 AD0INT 位以确定转换何时结束（也可

以使用 ADC0 中断)。建议的查询步骤如下：

1．写‘ 0 ’到 AD0INT；
2．向 AD0BUSY 写‘ 1 ’；
3．查询并等待 AD0INT 变‘ 1 ’；
4．处理 ADC0 数据

当 CNVSTR0 被用作转换启动源时，它必须在交叉开关中被使能，对应的引脚必须被配置为漏极开路、高阻方式（见 18. 端口输入/输出之端口 I/O 配置部分)。

5.2.2 跟踪方式

寄存器 ADC0CN 中的 AD0TM 位控制 ADC0 的跟踪保持方式。在缺省状态，除了转换期间之外 ADC0 输入被连续跟踪。当 AD0TM 位为逻辑‘ 1 ’时，ADC0 工作在低功耗跟踪保持方式。在该方式下，每次转换之前都有 3 个 SAR 时钟的跟踪周期（在启动转换信号有效之后)。当 CNVSTR0 信号用于在低功耗跟踪保持方式启动转换时，ADC0 只在 CNVSTR0 为低电平时跟踪；在 CNVSTR0 的上升沿开始转换（见图 5.3)。当整个芯片处于低功耗待机或休眠方式时，跟踪可以被禁止（关断)。当 AMUX 或 PGA 的设置频繁改变时，低功耗跟踪保持方式也非常有用，可以保证建立时间需求得到满足（见“ 5.2.3 建立时间要求 ”)。

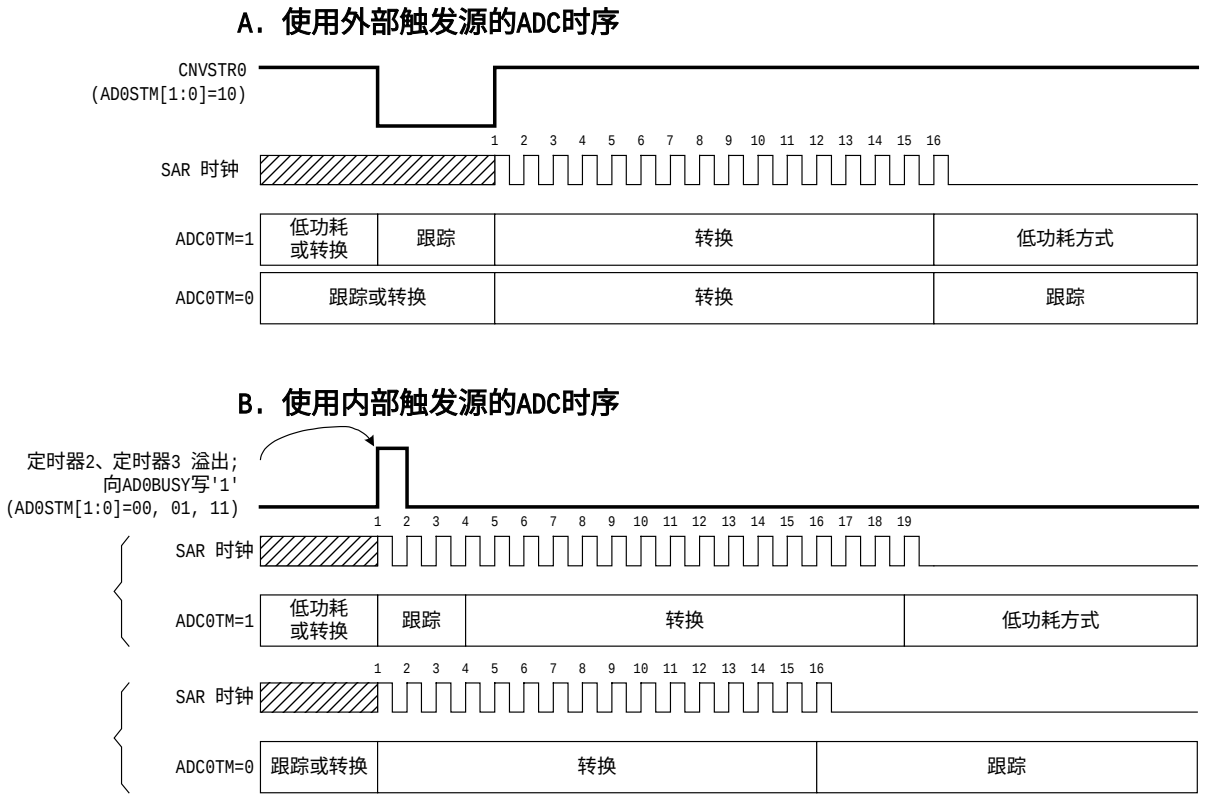


图 5.3 ADC0 的跟踪和转换时序示例

5.2.3 建立时间要求

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

方程 5.1 ADC0 建立时间要求

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

其中：

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

t 为所需要的建立时间，以秒为单位

R_{TOTAL} 为 ADC0 MUX 电阻与外部信号源电阻之和

n 为 ADC0 的分辨率，用比特表示（12）。

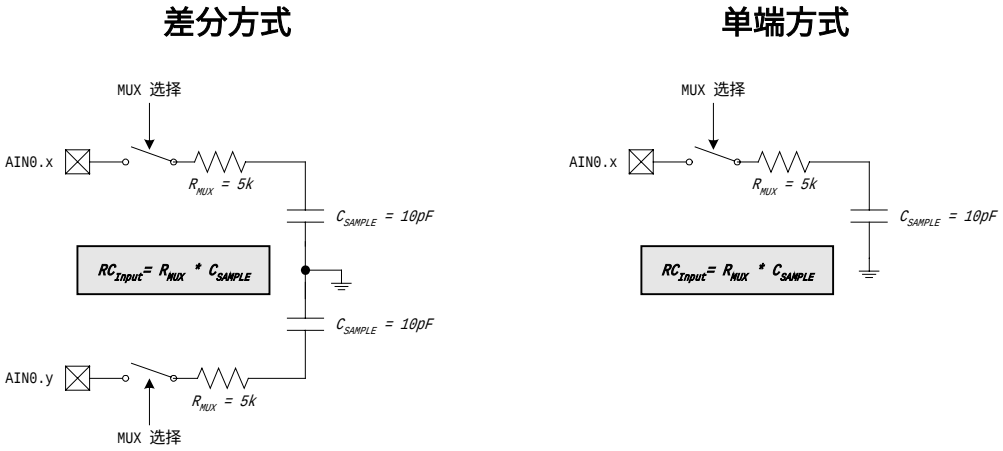


图 5.4 ADC0 等效输入电路

SFR 页： 0
SFR 地址：0xBA

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	

位 7-4： 未使用。读 = 0000b；写 = 忽略
位 3 AIN67IC：AIN0.6、AIN0.7 输入对配置位
0: AIN0.6 和 AIN0.7 为独立的单端输入
1: AIN0.6, AIN0.7 为（分别为）+, -差分输入对
位 2 AIN45IC：AIN0.4、AIN0.5 输入对配置位
0: AIN0.4 和 AIN0.5 为独立的单端输入
1: AIN0.4, AIN0.5 为（分别为）+, -差分输入对
位 1 AIN23IC：AIN0.2、AIN0.3 输入对配置位
0: AIN0.2 和 AIN0.2 为独立的单端输入
1: AIN0.2, AIN0.2 为（分别为）+, -差分输入对
位 0 AIN01IC：AIN0.0、AIN0.1 输入对配置位
0: AIN0.0 和 AIN0.1 为独立的单端输入
1: AIN0.0, AIN0.1 为（分别为）+, -差分输入对

注：对于被配置成差分输入的通道，ADC0 数据字格式为 2 的补码。

图 5.5 AMX0CF: AMUX0 配置寄存器

SFR 页： 0
SFR 地址：0xBB

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

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

AMX0CF 位 3-0	AMX0AD3-0								
	0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1xxx**
0000	AIN0.0	AIN0.1	AIN0.2	AIN0.3	AIN0.4	AIN0.5	AIN0.6	AIN0.7	温度传感器
0001	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	AIN0.4	AIN0.5	AIN0.6	AIN0.7	温度传感器
0010	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	AIN0.6	AIN0.7	温度传感器
0011	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	AIN0.6	AIN0.7	温度传感器
0100	AIN0.0	AIN0.1	AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	温度传感器
0101	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	温度传感器
0110	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	温度传感器
0111	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	温度传感器
1000	AIN0.0	AIN0.1	AIN0.2	AIN0.3	AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		温度传感器
1001	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		温度传感器
1010	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		温度传感器
1011	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		温度传感器
1100	AIN0.0	AIN0.1	AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		温度传感器
1101	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		温度传感器
1110	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		温度传感器
1111	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		温度传感器

图 5.6 AMX0SL: AMUX0 通道选择寄存器

SFR 页： 0
SFR 地址：0xBC

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AMP0GN2	AMP0GN1	AMP0GN0	11111000
位7	位6	位5	位4	位3	位2	位1	位0	

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

$$AD0SC = \frac{SYSCLK}{2 \times CLK_{SAR0}} - 1 \qquad (AD0SC > 00000b)$$

当 AD0SC 位= 00000b 时，SAR 转换时钟 = SYSCLK，可以在 SYSCLK 频率较低时获得较块的 ADC 转换速度。

位 2-0： AMP0GN2-0: ADC0 内部放大器增益(PGA)
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 ADC0CF: ADC0 配置寄存器

SFR 页： 0
SFR 地址：0xE8 （可位寻址）

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AD0EN	AD0TM	AD0INT	AD0BUSY	AD0CM1	AD0CM0	AD0WINT	AD0LJST	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7： AD0EN：ADC0 使能位
0：ADC0 禁止。ADC0 处于低耗停机状态。
1：ADC0 使能。ADC0 处于活动状态，并准备转换数据。

位 6： AD0TM：ADC 跟踪方式位
0：当 ADC 被使能时，除了转换期间之外一直处于跟踪方式。
1：由 ADSTM1-0 定义跟踪方式。

位 5： AD0INT：ADC0 转换结束中断标志
该标志必须用软件清‘0’。
0：从最后一次将该位清 0 后，ADC0 还没有完成一次数据转换。
1：ADC 完成了一次数据转换。

位 4： AD0BUSY：ADC0 忙标志位
读：
0：ADC0 转换结束或当前没有正在进行的数据转换。AD0INT 在 AD0BUSY 的下降沿被置‘1’。
1：ADC0 正在进行转换。
写：
0：无作用
1：若 ADSTM1-0＝00b 则启动 ADC0 转换。

位 3-2： AD0CM1-0：ADC0 转换启动方式选择位。
如果 AD0TM = 0：
00：每次向 AD0BUSY 写 1 时启动 ADC0 转换。
01：定时器 3 溢出启动 ADC0 转换。
10：CNVSTR0 上升沿启动 ADC0 转换。
11：定时器 2 溢出启动 ADC0 转换。
如果 AD0TM = 1：
00：向 AD0BUSY 写 1 时启动跟踪，持续 3 个 SAR 时钟，然后进行转换。
01：定时器 3 溢出启动跟踪，持续 3 个 SAR 时钟，然后进行转换。
10：只有当 CNVSTR0 输入为逻辑低电平时 ADC0 跟踪，在 CNVSTR0 的上升沿开始转换。
11：定时器 2 溢出启动跟踪，持续 3 个 SAR 时钟，然后进行转换。

位 1： AD0WINT：ADC0 窗口比较中断标志。
该位必须用软件清 0。
0：自该标志被清除后未发生过 ADC0 窗口比较匹配。
1：发生了 ADC0 窗口比较匹配。

位 0： AD0LJST：ADC0 数据左对齐选择位。
0：ADC0H:ADC0L 寄存器数据右对齐。
1：ADC0H:ADC0L 寄存器数据左对齐。

图 5.8 ADC0CN: ADC0 控制寄存器

SFR 页： 0
SFR 地址：0xBF

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

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

图 5.9 ADC0H: ADC 数据字 MSB 寄存器

SFR 页： 0
SFR 地址：0xBE

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

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

图 5.10 ADC0L: ADC0 数据字 LSB 寄存器

12 位 ADC 结果数据字在 ADC0 数据字寄存器中存放如下：

ADC0H[3:0]:ADC0L[7:0]，如果 AD0LJST＝0

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

ADC0H[7:0]:ADC0L[7:4]，如果 AD0LJST＝1

（ADC0L[3:0]=0000b）

例：ADC 数据字转换表，AIN0.0 为单端输入方式

（AMX0CF=0x00, AMX0SL=0x00）

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

例：ADC 数据字转换表，AIN0.0-AIN0.1 为差分输入对

（AMX0CF=0x01, AMX0SL=0x00）

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

对于 AD0LJST = 0：

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

图 5.11 ADC0 数据字示例

5.3 ADC0 可编程窗口检测器

ADC0 可编程窗口检测器不停地将 ADC0 的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 AD0WINT 位）也可被用于查询方式。参考字的高和低字节被装入到 ADC0 下限（大于）和 ADC0 上限（小于）寄存器（ADC0GTH、ADC0GTL、ADC0LTH 和 ADC0LTL）。图 5.16 - 5.19 给出了比较示例供参考。注意，窗口检测器标志既可以在测量数据位于用户编程的极限值以内时有效，也可以在测量数据位于用户编程的极限值以外时有效，这取决于 ADC0GTx 和 ADC0LTx 寄存器的编程值。

SFR 页： 0
SFR 地址：0xC5

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

位 7-0： ADC0 下限（大于）数据字的高字节

图 5.12 ADC0GTH: ADC0 下限数据高字节寄存器

SFR 页： 0
SFR 地址：0xC4

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

位 7-0： ADC0 下限（大于）数据字的低字节

图 5.13 ADC0GTL: ADC0 下限数据低字节寄存器

SFR 页： 0
SFR 地址：0xC7

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

位 7-0： ADC0 上限（小于）数据字的高字节

图 5.14 ADC0LTH: ADC0 上限数据高字节寄存器

SFR 页： 0
SFR 地址：0xC6

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

位 7-0： ADC0 上限（小于）数据字的低字节

图 5.15 ADC0LTL: ADC0 上限数据低字节寄存器

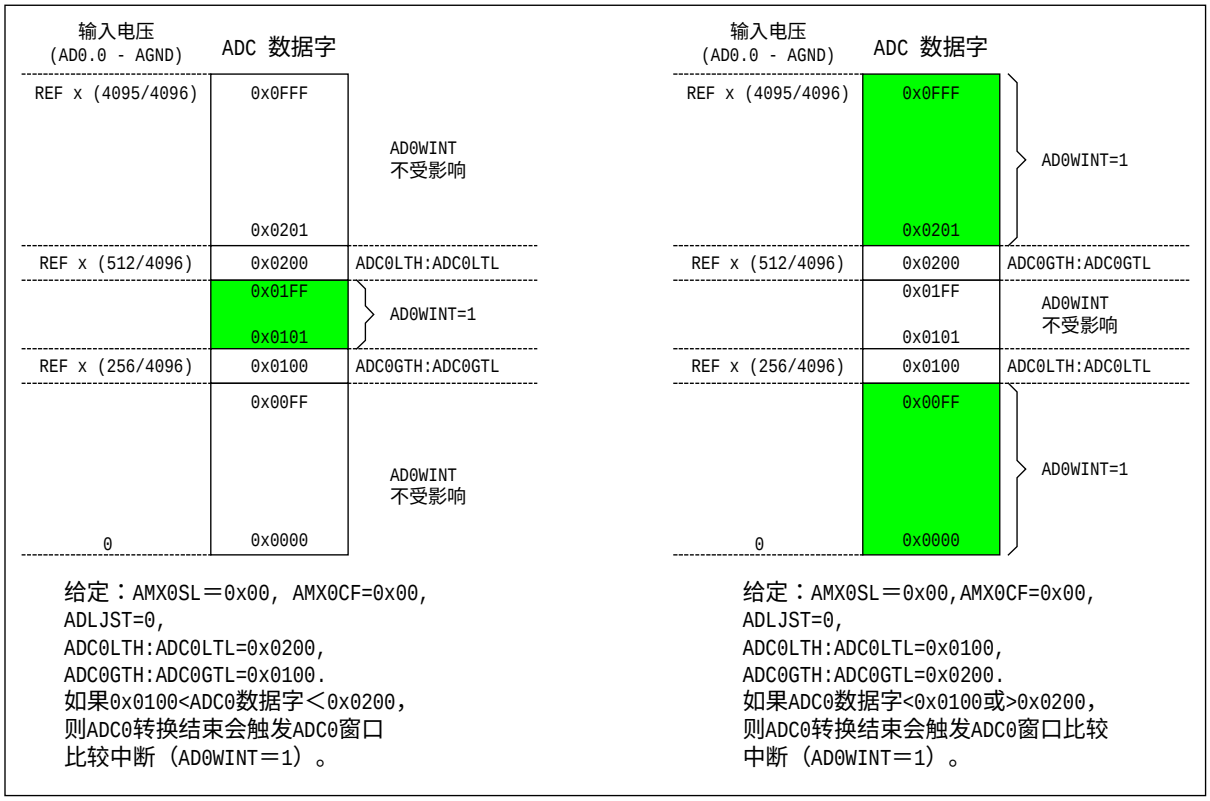


图 5.16 12 位 ADC0 窗口中断示例：右对齐的单端数据

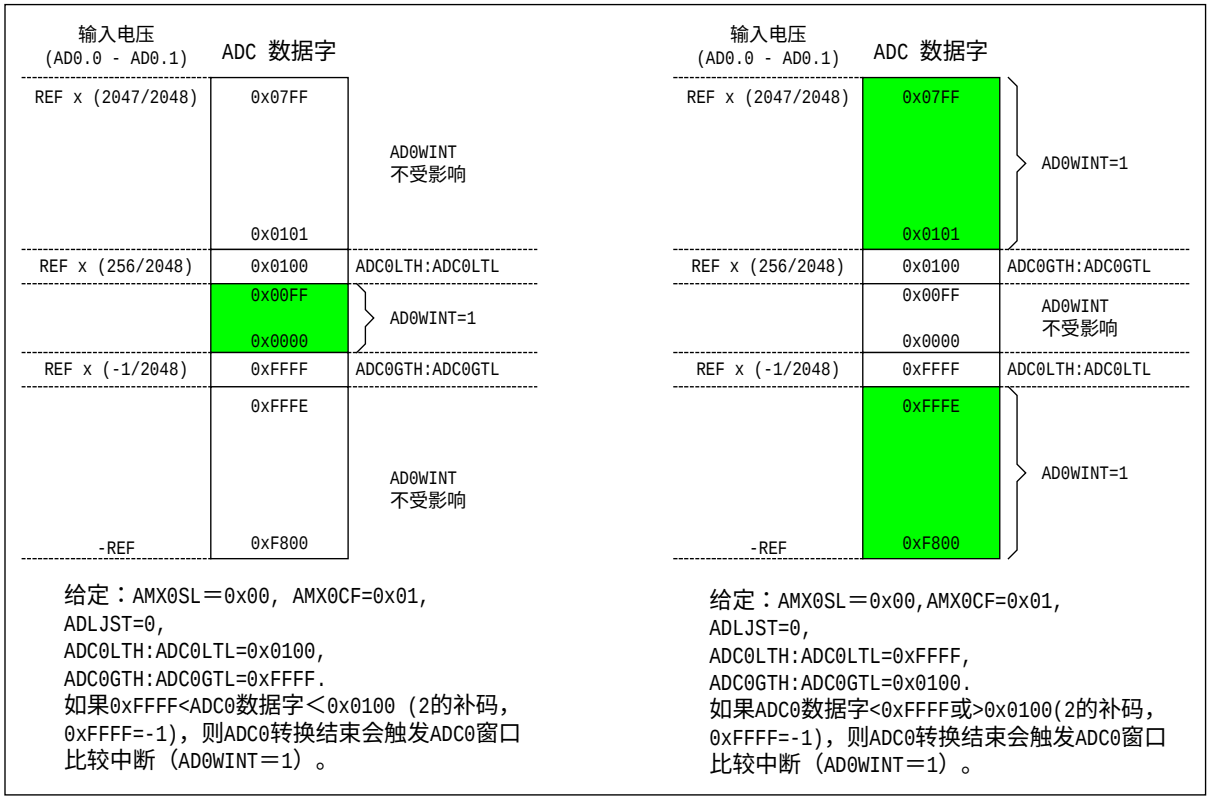


图 5.17 12 位 ADC0 窗口中断示例：右对齐的差分数据

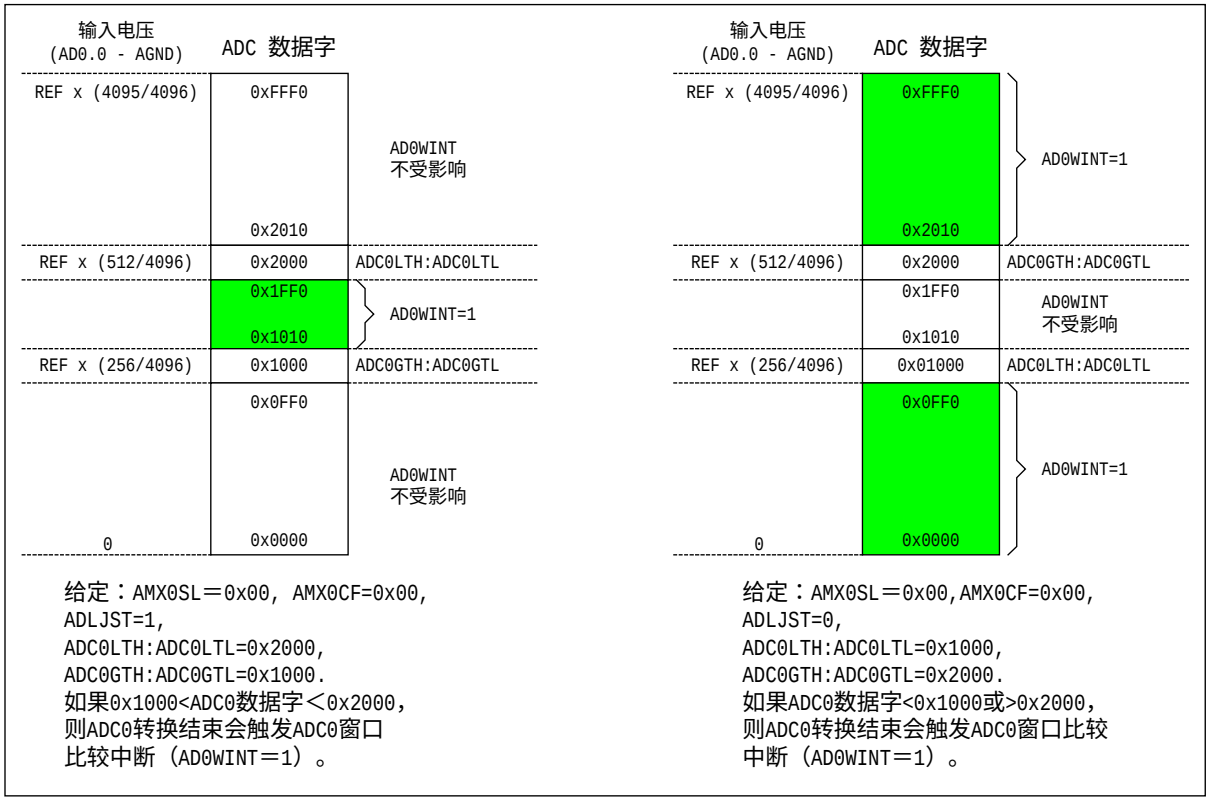


图 5.18 12 位 ADC0 窗口中断示例：左对齐的单端数据

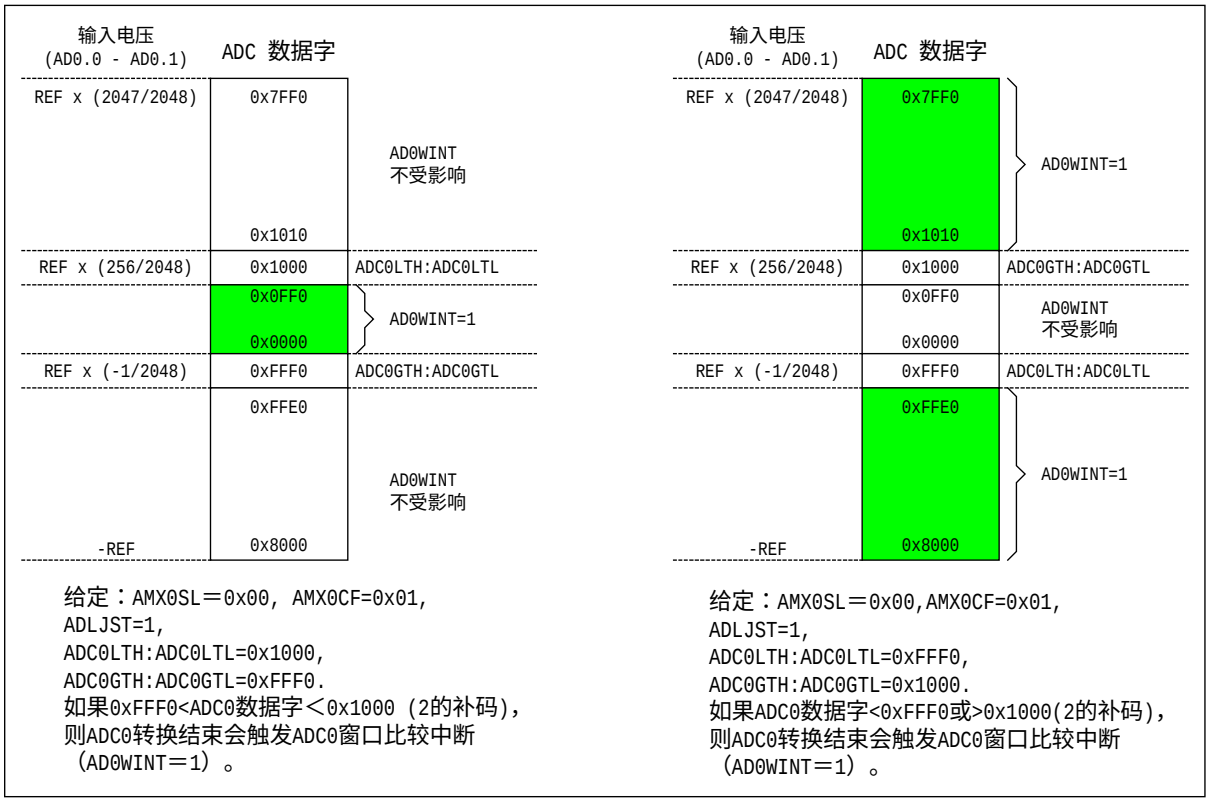


图 5.19 12 位 ADC0 窗口中断示例：左对齐的差分数据

表 5.1 12 位 ADC0 电气特性（C8051F120/1/4/5）

VDD=3.0V, AV+=3.0V, VREF=2.40V(REFBE=0), PGA 增益=1, -40°C ~ +85°C（除非特别说明）

参 数	条 件	最小值	典型值	最大值	单 位
直流精度					
分辨率			12		位
积分非线性				±1	LSB
微分非线性	保证单调			±1	LSB
偏移误差			-3±1		LSB
满度误差	差分方式		-7±3		LSB
偏移温度系数			±0.25		ppm/℃
动态性能（10kHz 正弦波输入，满度值的 0 到-1dB，100ksps）					
信号与噪声失真比		66			dB
总谐波失真	到 5 次谐波		-75		dB
有效动态范围			80		dB
转换速率					
SAR 时钟频率				2.5	MHz
转换占用 SAR 时钟数		16			周期
跟踪/保持捕获时间		1.5			μS
转换速率				100	ksps
模拟输入					
电压转换范围	单端方式	0		VREF	V
*共模电压范围	差分方式	AGND		AV+	V
输入电容			10		pF
温度传感器					
线性度（注 1）			±0.2		℃
偏移	温度 = 0℃		776		mV
偏移误差（注 1，注 2）	温度 = 0℃		±8.5		mV
斜率（增益）			2.86		mV/ ℃
斜率误差（注 2）			±0.034		mV/ ℃
电源指标					
电源电流（AV+给 ADC 供电）	工作状态，100ksps		450	900	μA
电源抑制比			±0.3		mV/V

注 1：包括 ADC 偏移、增益及线性度变化。

注 2：表示偏离平均值 1 个标准差。

6. ADC0（10 位，仅 C8051F122/3/6/7 和 C8051F13x）

C8051F122/3/6/7 和 C8051F13x 的 ADC0 子系统包括一个 9 通道的可编程模拟多路选择器（AMUX0），一个可编程增益放大器（PGA0）和一个 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）。AMUX0、PGA0、数据转换方式及窗口检测器都可用软件通过图 6.1 所示的特殊功能寄存器来控制。ADC0 所使用的电压基准按“9. 电压基准”所述选择。只有当 ADC0 控制寄存器中的 AD0EN 位被置‘1’时 ADC0 子系统（ADC0、跟踪保持器和 PGA0）才被允许工作。当 AD0EN 位为‘0’时，ADC0 子系统处于低功耗关断方式。

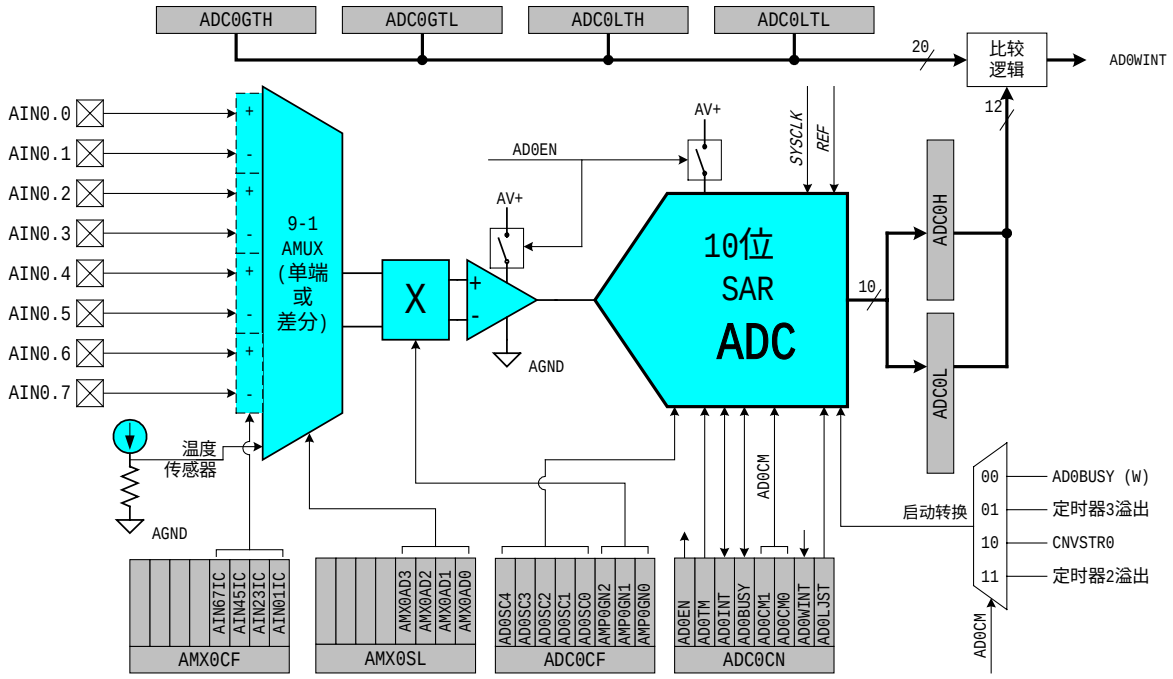


图 5.1 10 位 ADC0 功能框图

6.1 模拟多路开关和 PGA

AMUX 中的 8 个通道用于外部测量，而第九通道在内部被接到片内温度传感器（温度传输函数示于图 6.2）。注意，PGA0 的增益对温度传感器也起作用。可以将 AMUX 输入对编程为工作在差分或单端方式。这就允许用户对每个通道选择最佳的测量技术，甚至可以在测量过程中改变方式。在系统复位后 AMUX 的默认方式为单端输入。有两个与 AMUX 相关的寄存器：通道选择寄存器 AMX0SL（图 6.6）和配置寄存器 AMX0CF（图 6.5）。图 6.6 中的表给出了每种配置下各通道的功能。PGA 对 AMUX 输出信号的放大倍数由 ADC0 配置寄存器 ADC0CF（图 6.7）中的 AMP0GN2-0 确定。PGA 增益可以用软件编程为 0.5、1、2、4、8 或 16，复位后的默认增益为 1。

温度传感器的传输函数示于图 5.2。当温度传感器被选中（用 AMX0SL 中的 AMX0AD3-0）时，其输出电压（V_{TEMP}）是 PGA 的输入；PGA 对该电压的放大倍数由用户编程的 PGA 设置值决定。传输函数的斜率和偏移值见表 5.1

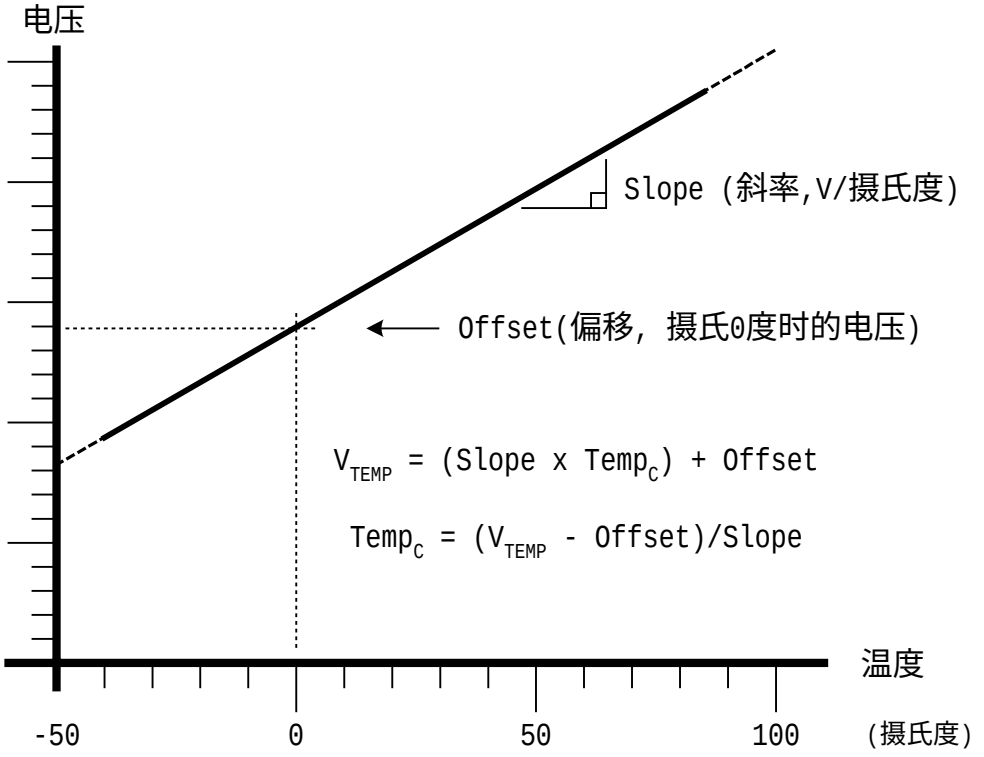


图 5.2 温度传感器传输函数

6.2 ADC 的工作方式

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

6.2.1 启动转换

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

1．向 ADC0CN 的 AD0BUSY 位写 1；
2．定时器 3 溢出（即定时的连续转换）；
3．外部 ADC 转换启动信号的上升沿，CNVSTR0；
4．定时器 2 溢出（即定时的连续转换）。

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

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

1．写‘ 0 ’到 AD0INT；
2．向 AD0BUSY 写‘ 1 ’；
3．查询并等待 AD0INT 变‘ 1 ’；
4．处理 ADC0 数据

当 CNVSTR0 被用作转化启动源时，它必须在交叉开关中被使能，对应的引脚必须被配置为漏极开路、高阻方式（见 18. 端口输入/输出之端口 I/O 配置部分）。

6.2.2 跟踪方式

ADC0CN 中的 AD0TM 位控制 ADC0 的跟踪保持方式。在缺省状态，除了转换期间之外 ADC0 输入被连续跟踪。当 AD0TM 位为逻辑‘ 1 ’时，ADC0 工作在低功耗跟踪保持方式。在该方式下，每次转换之前都有 3 个 SAR 时钟的跟踪周期（在启动转换信号有效之后）。当 CNVSTR0 信号用于在低功耗跟踪保持方式启动转换时，ADC0 只在 CNVSTR0 为低电平时跟踪；在 CNVSTR 的上升沿开始转换（见图 6.3）。当整个芯片处于低功耗待机或休眠方式时，跟踪可以被禁止（关断）。当 AMUX 或 PGA 的设置频繁改变时，低功耗跟踪保持方式也非常有用，可以保证建立时间需求得到满足（见“ 6.2.3 建立时间要求 ”）。

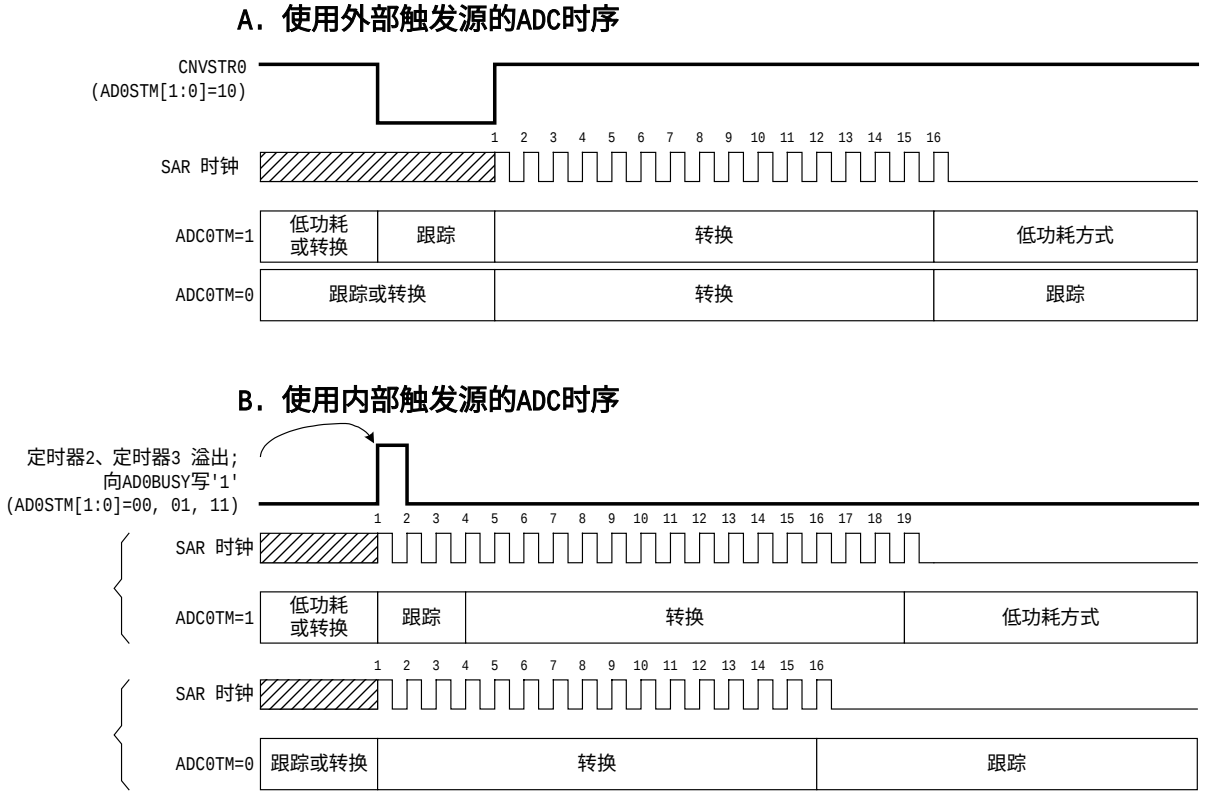


图 6.3 ADC0 的跟踪和转换时序举例

6.2.3 建立时间要求

当 ADC0 输入配置发生改变时（AMUX 或 PGA 的选择发生变化），在进行一次精确的转换之前需要有一个最小的跟踪时间。该跟踪时间由 ADC0 模拟多路器的电阻、ADC0 采样电容、外部信号源阻抗及所要求的转换精度决定。图 6.4 给出了单端和差分方式下等效的 ADC0 输入电路。注意：这两种等效电路的时间常数完全相同。对于一个给定的建立精度（*SA*），所需要的 ADC0 建立时间可以用方程 6.1 估算。当测量温度传感器的输出时，R_{TOTAL} 等于 R_{MUX}。在 MUX 或 PGA 选择发生变化之后，至少需要 1.5 μs 的建立时间。注意：在低功耗跟踪方式，每次转换需要用三个 SAR 时钟跟踪。对于大多数应用，三个 SAR 时钟可以满足跟踪需要。

方程 6.1 ADC0 建立时间要求

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

其中：

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

t 为所需要的建立时间，以秒为单位

R_{TOTAL} 为 ADC0 MUX 电阻与外部信号源电阻之和

n 为 ADC0 的分辨率，用比特表示（10）。

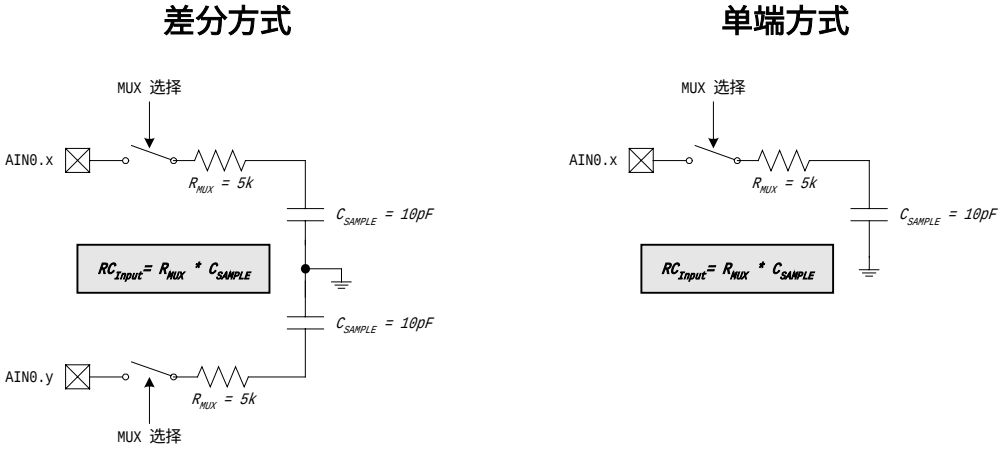


图 6.4 ADC0 等效输入电路

SFR 页： 0
SFR 地址：0xBA

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	

位 7-4： 未使用。读 = 0000b；写 = 忽略
位 3 AIN67IC：AIN0.6、AIN0.7 输入对配置位
0: AIN0.6 和 AIN0.7 为独立的单端输入
1: AIN0.6, AIN0.7 为（分别为）+, -差分输入对
位 2 AIN45IC：AIN0.4、AIN0.5 输入对配置位
0: AIN0.4 和 AIN0.5 为独立的单端输入
1: AIN0.4, AIN0.5 为（分别为）+, -差分输入对
位 1 AIN23IC：AIN0.2、AIN0.3 输入对配置位
0: AIN0.2 和 AIN0.2 为独立的单端输入
1: AIN0.2, AIN0.2 为（分别为）+, -差分输入对
位 0 AIN01IC：AIN0.0、AIN0.1 输入对配置位
0: AIN0.0 和 AIN0.1 为独立的单端输入
1: AIN0.0, AIN0.1 为（分别为）+, -差分输入对

注：对于被配置成差分输入的通道，ADC0 数据字格式为 2 的补码。

图 6.5 AMX0CF: AMUX0 配置寄存器

图 6.6 AMX0SL: AMUX0 通道选择寄存器

SFR 页： 0
SFR 地址：0xBB

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

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

AMX0CF 位 3-0	AMX0AD3-0								
	0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1xxx**
0000	AIN0.0	AIN0.1	AIN0.2	AIN0.3	AIN0.4	AIN0.5	AIN0.6	AIN0.7	温度传感器
0001	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	AIN0.4	AIN0.5	AIN0.6	AIN0.7	温度传感器
0010	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	AIN0.6	AIN0.7	温度传感器
0011	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	AIN0.6	AIN0.7	温度传感器
0100	AIN0.0	AIN0.1	AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	温度传感器
0101	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	温度传感器
0110	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	温度传感器
0111	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	温度传感器
1000	AIN0.0	AIN0.1	AIN0.2	AIN0.3	AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		温度传感器
1001	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		温度传感器
1010	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		温度传感器
1011	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		温度传感器
1100	AIN0.0	AIN0.1	AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		温度传感器
1101	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		温度传感器
1110	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		温度传感器
1111	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		温度传感器

SFR 页： 0
SFR 地址：0xBC

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AMP0GN2	AMP0GN1	AMP0GN0	11111000
位7	位6	位5	位4	位3	位2	位1	位0	

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

$$AD0SC = \frac{SYSCLK}{2 \times CLK_{SAR0}} - 1 \qquad (AD0SC > 00000b)$$

当 AD0SC 位= 00000b 时，SAR 转换时钟 = SYSCLK，可以在 SYSCLK 频率较低时获得较块的 ADC 转换速度。

位 2-0： AMP0GN2-0: ADC0 内部放大器增益(PGA)
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 ADC0CF: ADC0 配置寄存器

SFR 页： 0
SFR 地址：0xE8 （可位寻址）

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AD0EN	AD0TM	AD0INT	AD0BUSY	AD0CM1	AD0CM0	AD0WINT	AD0LJST	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7： AD0EN：ADC0 使能位
0：ADC0 禁止。ADC0 处于低耗停机状态。
1：ADC0 使能。ADC0 处于活动状态，并准备转换数据。

位 6： AD0TM：ADC 跟踪方式位
0：当 ADC 被使能时，除了转换期间之外一直处于跟踪方式。
1：由 ADSTM1-0 定义跟踪方式。

位 5： AD0INT：ADC0 转换结束中断标志
该标志必须用软件清‘0’。
0：从最后一次将该位清 0 后，ADC0 还没有完成一次数据转换。
1：ADC 完成了一次数据转换。

位 4： AD0BUSY：ADC0 忙标志位
读：
0：ADC0 转换结束或当前没有正在进行的数据转换。AD0INT 在 AD0BUSY 的下降沿被置‘1’。
1：ADC0 正在进行转换。
写：
0：无作用
1：若 ADSTM1-0＝00b 则启动 ADC0 转换。

位 3-2： AD0CM1-0：ADC0 转换启动方式选择位。
如果 AD0TM = 0：
00：每次向 AD0BUSY 写 1 时启动 ADC0 转换。
01：定时器 3 溢出启动 ADC0 转换。
10：外部 CNVSTR0 上升沿启动 ADC0 转换。
11：定时器 2 溢出启动 ADC0 转换。
如果 AD0TM = 1：
00：向 AD0BUSY 写 1 时启动跟踪，持续 3 个 SAR 时钟，然后进行转换。
01：定时器 3 溢出启动跟踪，持续 3 个 SAR 时钟，然后进行转换。
10：只有当 CNVSTR0 输入为逻辑低电平时 ADC0 跟踪，在 CNVSTR0 的上升沿开始转换。
11：定时器 2 溢出启动跟踪，持续 3 个 SAR 时钟，然后进行转换。

位 1： AD0WINT：ADC0 窗口比较中断标志。
该位必须用软件清 0。
0：自该标志被清除后未发生过 ADC0 窗口比较匹配。
1：发生了 ADC0 窗口比较匹配。

位 0： AD0LJST：ADC0 数据左对齐选择位。
0：ADC0H:ADC0L 寄存器数据右对齐。
1：ADC0H:ADC0L 寄存器数据左对齐。

图 6.8 ADC0CN: ADC0 控制寄存器

SFR 页： 0
SFR 地址：0xBF

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

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

图 6.9 ADC0H: ADC 数据字 MSB 寄存器

SFR 页： 0
SFR 地址：0xBE

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

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

图 6.10 ADC0L: ADC0 数据字 LSB 寄存器

10 位 ADC 结果数据字在 ADC0 数据字寄存器中存放如下：

ADC0H[1:0]:ADC0L[7:0]，如果 AD0LJST＝0
（如果是差分输入，ADC0H[7:2]为 ADC0H.1 的符号扩展位，否则=000000b）

ADC0H[7:0]:ADC0L[7:6]，如果 AD0LJST＝1
（ADC0L[5:0]=000000b）

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

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

例：ADC 数据字转换表，AIN0.0-AIN0.1 为差分输入对
（AMX0CF=0x01, AMX0SL=0x00）

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

对于 AD0LJST = 0：

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

图 6.11 ADC0 数据字示例

6.3 ADC0 可编程窗口检测器

ADC0 可编程窗口检测器不停地将 ADC0 的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 AD0WINT 位）也可被用于查询方式。参考字的高和低字节被装入到 ADC0 下限（大于）和 ADC0 上限（小于）寄存器（ADC0GTH、ADC0GTL、ADC0LTH 和 ADC0LTL）。图 6.16 - 6.19 给出了比较示例供参考。注意，窗口检测器标志既可以在测量数据位于用户编程的极限值以内时有效，也可以在测量数据位于用户编程的极限值以外时有效，这取决于 ADC0GTx 和 ADC0LTx 寄存器的编程值。

SFR 页： 0
SFR 地址：0xC5

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

位 7-0： ADC0 下限数据字的高字节

图 6.12 ADC0GTH: ADC0 下限数据高字节寄存器

SFR 页： 0
SFR 地址：0xC4

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

位 7-0： ADC0 下限数据字的低字节

图 6.13 ADC0GTL: ADC0 下限数据低字节寄存器

SFR 页： 0
SFR 地址：0xC7

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

位 7-0： ADC0 数据上限数据字的高字节

图 6.14 ADC0LTH: ADC0 上限数据高字节寄存器

SFR 页： 0
SFR 地址：0xC6

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

位 7-0： ADC0 上限数据字的低字节

图 6.15 ADC0LTL: ADC0 上限数据低字节寄存器

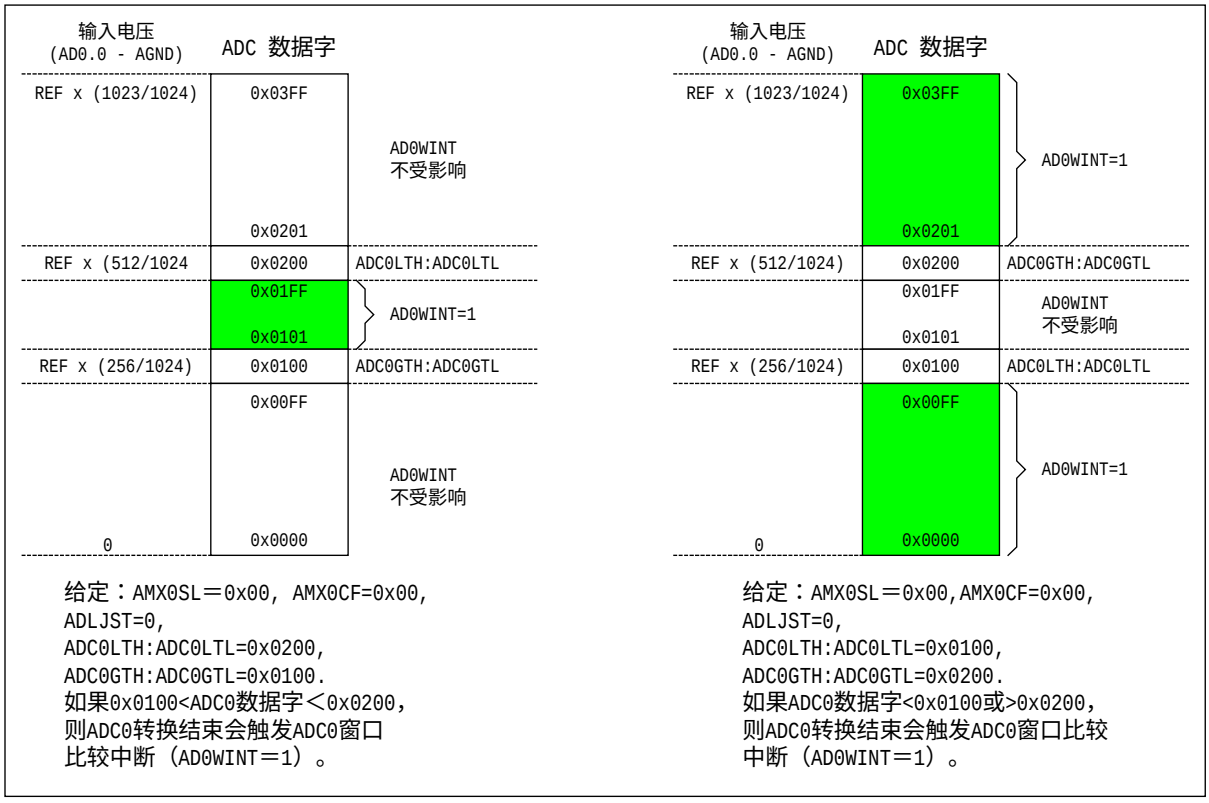


图 6.16 10 位 ADC0 窗口中断示例：右对齐的单端数据

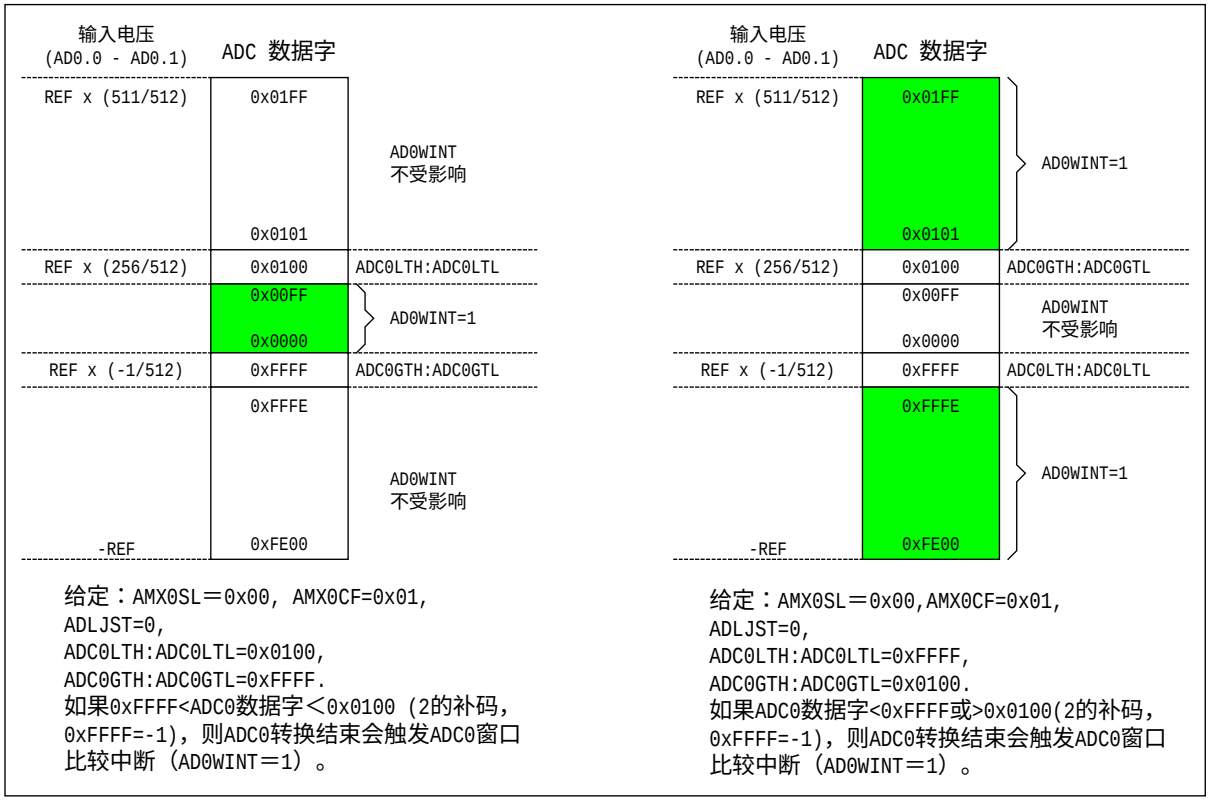


图 6.17 10 位 ADC0 窗口中断示例：右对齐的差分数据

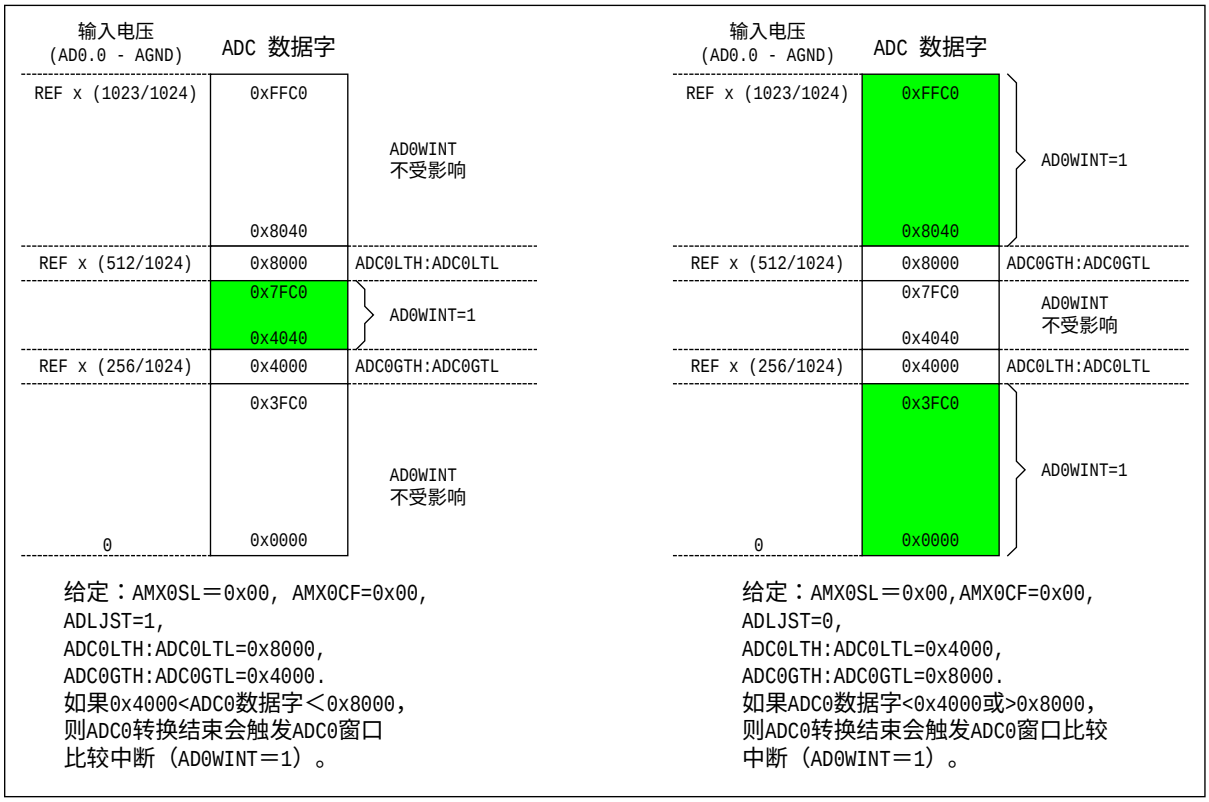


图 6.18 10 位 ADC0 窗口中断示例：左对齐的单端数据

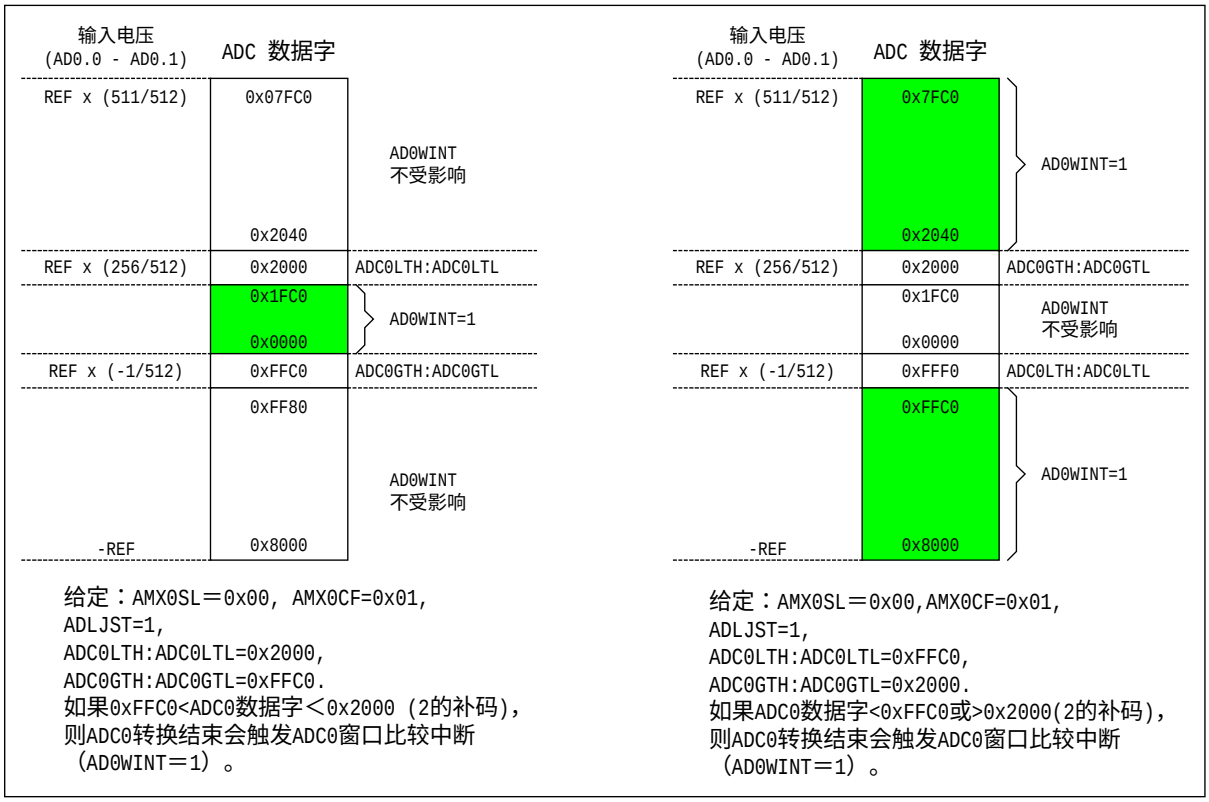


图 6.19 10 位 ADC0 窗口中断示例：左对齐的差分数据

表 6.1 10 位 ADC0 电气特性（C8051F122/3/6/7 和 C8051F13x）

VDD=3.0V, AV+=3.0V, VREF=2.40V(REFBE=0), PGA 增益=1, -40°C ~ +85°C（除非特别说明）

参 数	条 件	最小值	典型值	最大值	单 位
直流精度					
分辨率			10		位
积分非线性				±1	LSB
微分非线性	保证单调			±1	LSB
偏移误差			±0.5		LSB
满度误差	差分方式		-1.5±0.5		LSB
偏移温度系数			±0.25		ppm/℃
动态性能（10kHz 正弦波输入，满度值的 0 到-1dB，100ksps）					
信号与噪声失真比		59			dB
总谐波失真	到 5 次谐波		-70		dB
有效动态范围			80		dB
转换速率					
SAR 时钟频率				2.5	MHz
转换占用 SAR 时钟数		16			周期
跟踪/保持捕获时间		1.5			μS
转换速率				100	ksps
模拟输入					
电压转换范围	单端方式	0		VREF	V
*共模电压范围	差分方式	AGND		AV+	V
输入电容			10		pF
温度传感器					
线性度	注 1		±0.2		℃
偏移	温度 = 0℃		776		mV
偏移误差（注 1，注 2）	温度 = 0℃		±8.5		mV
斜率（增益）			2.86		mV/ ℃
斜率误差（注 2）			±0.034		mV/ ℃
电源指标					
电源电流（AV+给 ADC 供电）	工作状态，100ksps		450	900	μA
电源抑制比			±0.3		mV/V

注 1：包括 ADC 偏移、增益及线性度变化。

注 2：表示偏离平均值 1 个标准差。

7. ADC2（8 位 ADC）

C8051F120/1/2/3/4/5/6/7 的 ADC2 子系统包括一个 8 通道的可配置模拟多路开关（AMUX2），一个可编程增益放大器（PGA2）和一个 500ksps、8 位分辨率的逐次逼近寄存器型 ADC，该 ADC 中集成了跟踪保持电路（见图 7.1 的原理框图）。AMUX2、PGA2 及数据转换方式都可用软件通过图 7.1 所示的特殊功能寄存器来配置。只有当 ADC2 控制寄存器（ADC2CN）中的 AD2EN 位被置‘1’时 ADC2 子系统（8 位 ADC、跟踪保持器和 PGA）才被使能。当 AD2EN 位为‘0’时，ADC2 子系统处于低功耗关断方式。关于 ADC2 电压基准的选择见“9. 电压基准”。

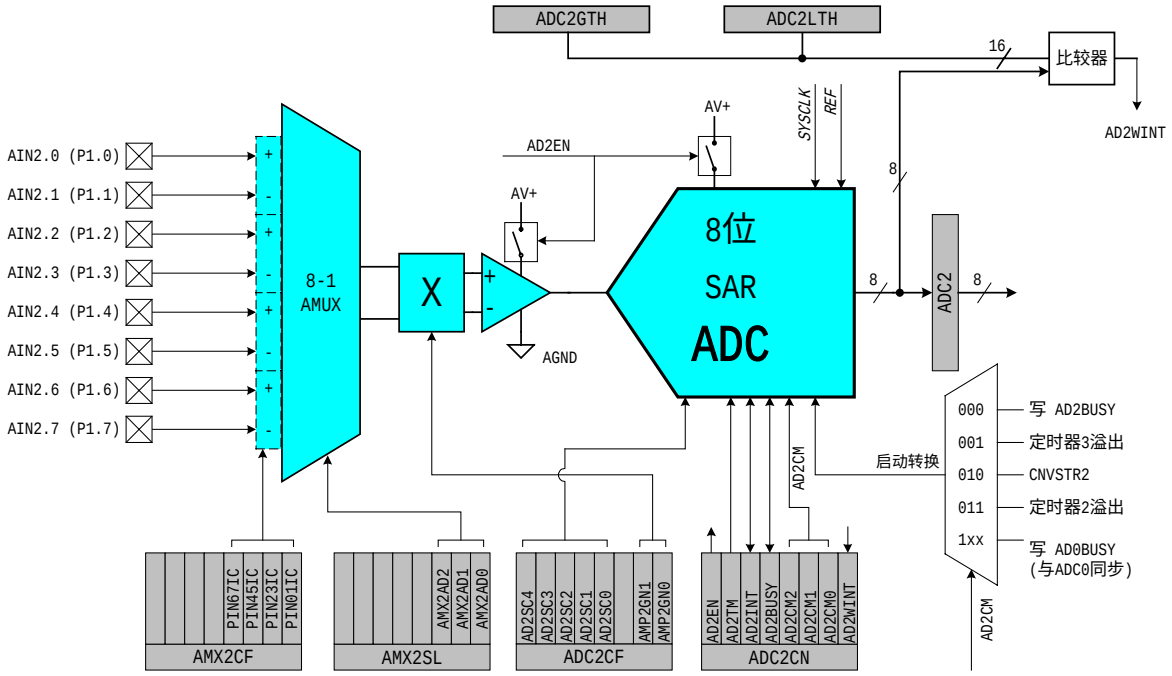


图 7.1 ADC2 功能框图

7.1 模拟多路开关和 PGA

ADC2 有 8 个通道用于测量，用寄存器 AMX2SL（图 7.5）选择通道。PGA 对 AMUX 输出信号的放大倍数由 ADC2 配置寄存器 ADC2CF（图 7.6）中的 AMP2GN2-0 确定。PGA 增益可以用软件编程为 0.5、1、2、4。复位时的默认增益为 0.5。

注意：AIN2 引脚也作为端口 1 的 I/O 引脚，当用作 ADC2 输入时必须被配置为模拟输入。为了将 AIN2 的某个引脚配置为模拟输入，要将寄存器 P1MDIN 中的对应位设置为‘0’。被选作模拟输入的端口 1 引脚被数字 I/O 交叉开关跳过。有关配置 AIN2 引脚的详细信息见“18.1.5. 配置端口 1 引脚为模拟输入）”。

7.2 ADC2 的工作方式

ADC2 的最高转换速度为 500ksps。ADC2 的转换时钟（SAR2 时钟）来源于系统时钟分频。由 ADC2CF 寄存器的 AD2SC 位决定。ADC2 转换时钟频率最大为 6 MHz。

7.2.1 启动转换

有 5 种 A/D 转换启动方式，由 ADC2CN 中的 ADC2 启动转换方式位（AD2CM2-0）的编程状态决定。转换启动源有：

1. 向 ADC2CN 的 AD2BUSY 位写‘ 1 ’；
2. 定时器 3 溢出（即定时的连续转换）；
3. 外部 ADC 转换启动信号 CNVSTR2 的上升沿；
4. 定时器 2 溢出（即定时的连续转换）。
5. 向 ADC0CN 的 AD0BUSY 位写 1（用一个软件命令启动 ADC2 和 ADC0）。

AD2BUSY 位在转换期间被置‘ 1 ’，转换结束后复‘ 0 ’。AD2BUSY 位的下降沿触发一个中断（当被允许时）并将 ADC2CN 中的中断标志置‘ 1 ’。转换结果保存在 ADC2 的数据字寄存器 ADC2 中。

当采用向 AD2BUSY 位写‘ 1 ’这一启动方式时，建议通过查询 AD2INT 来确定转换何时完成。建议的查询步骤如下：

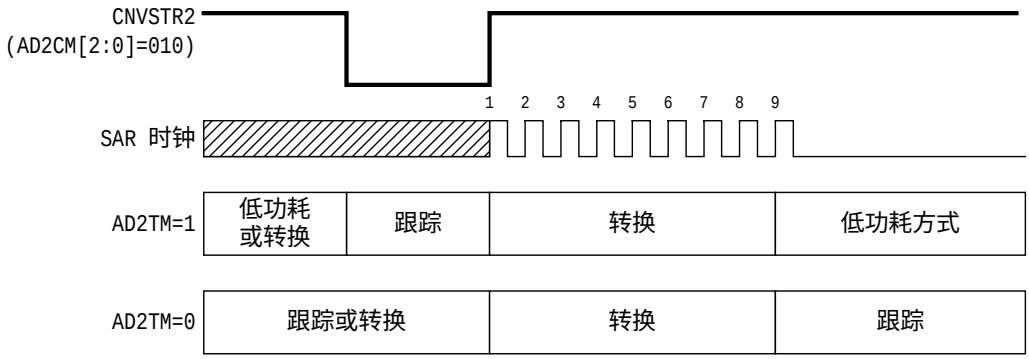
1. 向 AD2INT 写‘ 0 ’；
2. 向 AD2BUSY 写‘ 1 ’；
3. 查询并等待 AD2INT 变为‘ 1 ’；
4. 处理 ADC2 数据。

当 CNVSTR2 被当作转换启动源时，必须在交叉开关中被使能，对应的引脚必须被设置为漏极开路、高阻模式（有关端口 I/O 配置的详细信息，见“ 18. 端口输入/输出 ”）。

7.2.2 跟踪方式

ADC2CN 中的 AD2TM 位控制 ADC2 的跟踪保持方式。在缺省状态，ADC2 输入被连续跟踪（转换期间除外）。当 AD2TM 位被设置为逻辑‘ 1 ’时，ADC2 工作在低功耗跟踪方式。在该方式下，每次转换之前都要有三个 SAR 时钟的跟踪周期（在启动转换信号之后）。当在低功耗跟踪方式下用 CNVSTR2 信号作为转换启动源时，只在 CNVSTR2 为低电平时跟踪，从 CNVSTR2 的上升沿开始转换（见图 7.2）。当整个芯片处于低功耗停机或休眠方式时，跟踪被禁止。由于需要有建立时间，所以低功耗跟踪保持方式在需要频繁改变 AMUX 和 PGA 的场合也是非常有用的。关于建立时间要求，详见“ 7.2.3. 建立时间要求 ”。

A．使用外部触发源的ADC转换时序



A．使用内部触发源的ADC转换时序

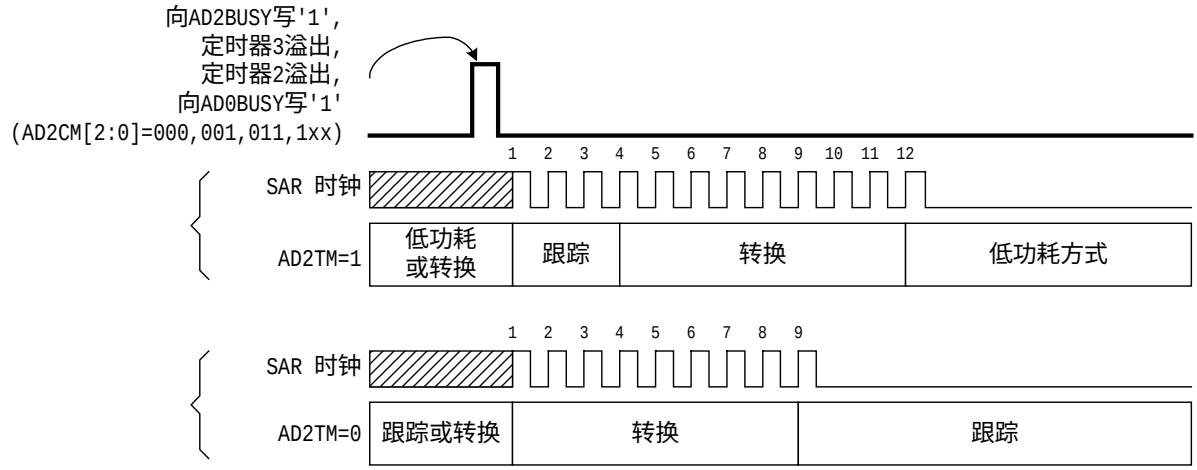


图 7.2　ADC2 跟踪和转换时序举例

7.2.3 建立时间要求

当 ADC2 输入配置发生改变时（即 AMUX 或 PGA 的选择发生变化），在进行一次精确的转换之前需要有一个最小的跟踪时间。该跟踪时间由 ADC2 模拟多路开关的电阻、ADC2 采样电容、外部信号源阻抗及所要求的转换精度决定。图 7.3 给出了等效的 ADC2 输入电路。对于一个给定的建立精度（SA），所需要的 ADC2 建立时间可以用方程 7.1 估算。注意：在 MUX 选择发生改变后，最少需要 0.8 μs 的建立时间；在低功耗跟踪方式，每次转换需要用三个 SAR2 时钟跟踪。对于大多数应用，三个 SAR 时钟可以满足跟踪需要。

方程 7.1 ADC2 建立时间要求

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

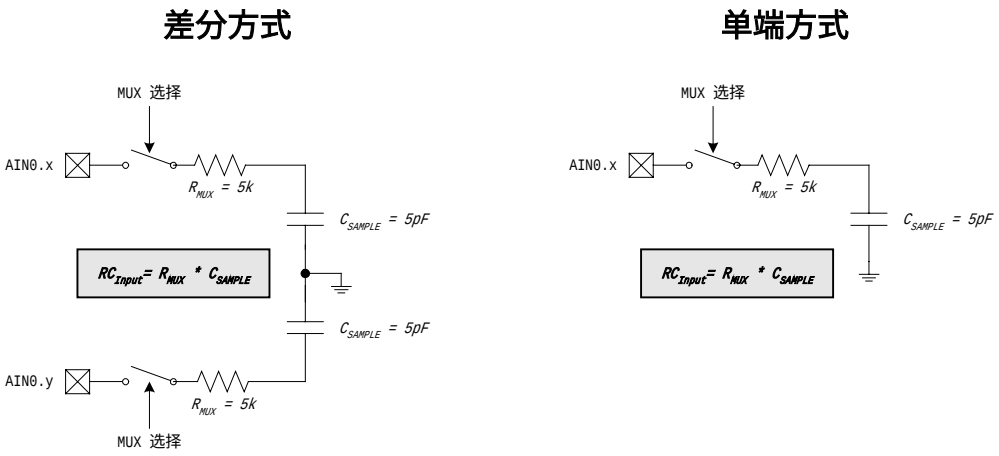
其中：

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

t 为所需要的建立时间，以秒为单位

R_{TOTAL} 为 ADC2 模拟开关电阻与外部信号源电阻之和

n 为 ADC 的分辨率，用比特表示（8）。



注：当 PGA 增益为 0.5 时，C_{SAMPLE} = 3 pF。

图 7.3 ADC2 等效输入电路

SFR 页： 2
SFR 地址：0xBA

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	

位 7-4： 未使用。读 = 0000b；写 = 忽略
位 3 AIN67IC：AIN2.6、AIN2.7 输入对配置位
0: AIN2.6 和 AIN2.7 为独立的单端输入
1: AIN2.6, AIN2.7 为（分别为）+, -差分输入对
位 2 AIN45IC：AIN2.4、AIN2.5 输入对配置位
0: AIN2.4 和 AIN2.5 为独立的单端输入
1: AIN2.4, AIN2.5 为（分别为）+, -差分输入对
位 1 AIN23IC：AIN2.2、AIN2.3 输入对配置位
0: AIN2.2 和 AIN2.3 为独立的单端输入
1: AIN2.2, AIN2.3 为（分别为）+, -差分输入对
位 0 AIN01IC：AIN2.0、AIN2.1 输入对配置位
0: AIN2.0 和 AIN2.1 为独立的单端输入
1: AIN2.0, AIN2.1 为（分别为）+, -差分输入对

注：对于被配置成差分输入的通道，ADC2 数据字格式为 2 的补码。

图 7.4 AMX2CF: AMUX2 配置寄存器

SFR 页： 0
SFR 地址：0xBB

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-		AMX2AD2	AMX2AD1	AMX2AD0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-3： 未使用。读 = 00000b；写 = 忽略
位 2-0： AMX2AD2-0: AMUX2 地址位
000-111b: 根据下表选择 ADC 输入

AMX2CF 位 3-0	AMX2AD2-0 000	001	010	011	100	101	110	111
0000	AIN2.0	AIN2.1	AIN2.2	AIN2.3	AIN2.4	AIN2.5	AIN2.6	AIN2.7
0001	+(AIN2.0) -(AIN2.1)		AIN2.2	AIN2.3	AIN2.4	AIN2.5	AIN2.6	AIN2.7
0010	AIN2.0	AIN2.1	+(AIN2.2) -(AIN2.3)		AIN2.4	AIN2.5	AIN2.6	AIN2.7
0011	+(AIN2.0) -(AIN2.1)		+(AIN2.2) -(AIN2.3)		AIN2.4	AIN2.5	AIN2.6	AIN2.7
0100	AIN2.0	AIN2.1	AIN2.2	AIN2.3	+(AIN2.4) -(AIN2.5)		AIN2.6	AIN2.7
0101	+(AIN2.0) -(AIN2.1)		AIN2.2	AIN2.3	+(AIN2.4) -(AIN2.5)		AIN2.6	AIN2.7
0110	AIN2.0	AIN2.1	+(AIN2.2) -(AIN2.3)		+(AIN2.4) -(AIN2.5)		AIN2.6	AIN2.7
0111	+(AIN2.0) -(AIN2.1)		+(AIN2.2) -(AIN2.3)		+(AIN2.4) -(AIN2.5)		AIN2.6	AIN2.7
1000	AIN2.0	AIN2.1	AIN2.2	AIN2.3	AIN2.4	AIN2.5	+(AIN2.6) -(AIN2.7)	
1001	+(AIN2.0) -(AIN2.1)		AIN2.2	AIN2.3	AIN2.4	AIN2.5	+(AIN2.6) -(AIN2.7)	
1010	AIN2.0	AIN2.1	+(AIN2.2) -(AIN2.3)		AIN2.4	AIN2.5	+(AIN2.6) -(AIN2.7)	
1011	+(AIN2.0) -(AIN2.1)		+(AIN2.2) -(AIN2.3)		AIN2.4	AIN2.5	+(AIN2.6) -(AIN2.7)	
1100	AIN2.0	AIN2.1	AIN2.2	AIN2.3	+(AIN2.4) -(AIN2.5)		+(AIN2.6) -(AIN2.7)	
1101	+(AIN2.0) -(AIN2.1)		AIN2.2	AIN2.3	+(AIN2.4) -(AIN2.5)		+(AIN2.6) -(AIN2.7)	
1110	AIN2.0	AIN2.1	+(AIN2.2) -(AIN2.3)		+(AIN2.4) -(AIN2.5)		+(AIN2.6) -(AIN2.7)	
1111	+(AIN2.0) -(AIN2.1)		+(AIN2.2) -(AIN2.3)		+(AIN2.4) -(AIN2.5)		+(AIN2.6) -(AIN2.7)	

图 7.5 AMX2SL: AMUX2 通道选择寄存器

图 7.6 ADC2CF: ADC2 配置寄存器

SFR 页： 2
SFR 地址：0xBC

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AD2SC4	AD2SC3	AD2SC2	AD2SC1	AD2SC0	-	AMP2GN1	AMP2GN0	11111000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-3： AD2SC4-0: ADC2 SAR 转换时钟周期控制位
SAR 转换时钟来源于系统时钟，由下面的方程给出，其中 *AD2SC* 表示 AD2SC4-0 中保持的 5 位数值，CLK_{SAR2} 表示所需要的 ADC2 SAR 时钟（注：ADC2 SAR 时钟应小于或等于 6MHz）。

$$AD2SC = \frac{SYSCLK}{CLK_{SAR2}} - 1$$

位 2： 未使用。读 = 0b；写 = 忽略

位 1-0： AMP2GN1-0: ADC2 内部放大器增益（PGA）
00: 增益 = 0.5
01: 增益 = 1
10: 增益 = 2
11: 增益 = 4

SFR 页： 2
SFR 地址：0xE8 （可位寻址）

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
AD2EN	AD2TM	AD2INT	AD2BUSY	AD2CM2	AD2CM1	AD2CM0	AD2WINT	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7： AD2EN：ADC2 使能位
0：ADC2 禁止。ADC2 处于低功耗停机状态。
1：ADC2 使能。ADC2 处于活动状态，并准备转换数据。

位 6： AD2TM：ADC2 跟踪方式位
0：一般跟踪方式。当 ADC2 被使能时，除了转换期间之外一直处于跟踪方式。
1：低功耗跟踪方式。由 AD2CM2-0 定义跟踪方式（见下面的说明）。

位 5： AD2INT：ADC2 转换结束中断标志
该标志必须用软件清‘0’。
0：从最后一次将该位清 0 后，ADC2 还没有完成一次数据转换。
1：ADC2 完成一次数据转换。

位 4： AD2BUSY：ADC2 忙标志位
读
0：ADC2 转换结束或不在进行数据转换。AD2INT 在 AD2BUSY 的下降沿被置‘1’。
1：ADC2 正在进行转换。
写
0：无效
1：若 AD2CM2-0＝000b 则启动 ADC2 转换。

位 3-1： AD2CM2-0：ADC2 转换启动方式选择
AD2TM = 0：
000：向 AD2BUSY 写 1 启动 ADC2 转换。
001：定时器 3 溢出启动 ADC2 转换。
010：外部 CNVSTR2 上升沿启动 ADC2 转换。
011：定时器 2 溢出启动 ADC2 转换。
1xx：向 AD0BUSY 写 1 启动 ADC2 转换（与 ADC0 软件命令转换同步）。
AD1TM = 1：
000：向 AD2BUSY 写 1 时启动跟踪并持续 3 个 SAR2 时钟，然后进行转换。
001：定时器 3 溢出启动跟踪并持续 3 个 SAR2 时钟，然后进行转换。
010：只有当 CNVSTR2 输入为逻辑低电平时才启动 ADC2 跟踪，在 CNVSTR2 上升沿开始转换。
011：定时器 2 溢出启动跟踪并持续 3 个 SAR2 时钟，然后进行转换。
1xx：向 AD0BUSY 写 2 启动跟踪并持续 3 个 SAR2 时钟，然后进行转换。

位 0： AD2WINT：ADC2 窗口比较中断标志。
该位必须用软件清 0。
0：自该标志被清除后未发生过 ADC2 窗口比较数据匹配。
1：发生了 ADC0 窗口比较数据匹配。

图 7.7 ADC2CN: ADC2 控制寄存器

SFR 页： 2
SFR 地址：0xBE

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

位 7-0： ADC2 数据字

图 7.8 ADC2: ADC2 数据字寄存器

单端方式示例

8位ADC数据字在ADC2数据字寄存器中的意义如下：

例：ADC2数据字转换映像，单端AIN2.0 输入

（AMX2CF = 0x00，AMX2SL = 0x00）

AIN2.0-AGND（伏）	ADC2
VREF*(255/256)	0xFF
VREF*(128/256)	0x80
VREF*(64/256)	0x40
0	0x00

$$转换结果 = Vin \times \frac{Gain}{VREF} \times 256$$

差分方式示例

8位ADC数据字在ADC2数据字寄存器中的意义如下：

例：ADC2数据字转换映像，差分AIN2.0- AIN2.1输入

（AMX2CF = 0x00，AMX2SL = 0x00）

AIN2.0- AIN2.1（伏）	ADC2
VREF*(127/128)	0x7F
VREF*(64/128)	0x40
0	0x00
-VREF*(64/128)	0xC0（-64d）
-VREF*(128/128)	0x80（-128d）

$$\textbf{转换结果} = \boldsymbol{Vin} \times \frac{\boldsymbol{Gain}}{\boldsymbol{2 \times VREF}} \times \boldsymbol{256}$$

图 7.9 ADC2 数据字示例

7.3 ADC2 可编程窗口检测器

ADC2 可编程窗口检测器不停地将 ADC2 的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（ADC2CN 中的 AD2WINT 位）也可被用于查询方式。ADC2 的下限（大于）和 ADC2 上限（小于）寄存器（分别为 ADC2GT 和 ADC2LT）保持比较值。图 7.11 和图 7.10 分别给出了差分和单端方式的比较示例。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 ADC2GT 和 ADC2LT 寄存器的编程值。

7.3.1 单端方式下的窗口检测器

图 7.10 给出了单端方式下窗口比较的两个例子，这里假设 ADC2LT=0x20，ADC2GT=0x10。注意，在单端方式，转换码为 8 位无符号整数，对应的信号变化范围是 0 ~ VREF*(255/256)。在左侧的例子中，如果 ADC2 转换字（ADC2）位于由 ADC2GT 和 ADC2LT 定义的范围之内（即 0x10<ADC2<0x20），则会产生 AD2WINT 中断。在右侧的例子中，如果 ADC2 转换字（ADC2）位于由 ADC2GT 和 ADC2LT 定义的范围之外（即 ADC2<0x10 或 ADC2>0x20），则会产生 AD2WINT 中断。

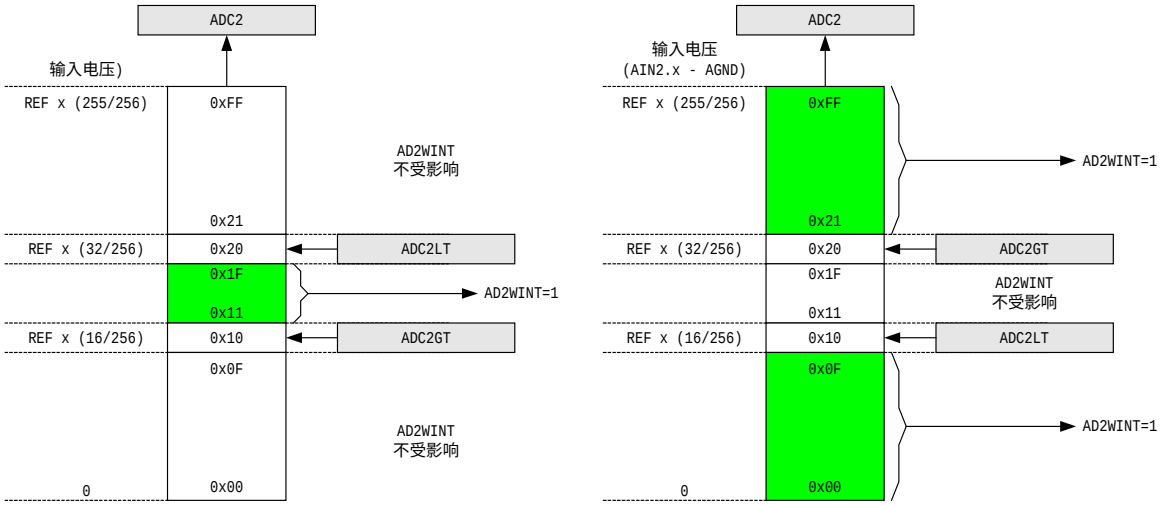


图 7.10 ADC2 窗口比较示例（单端方式）

7.3.2 差分方式下的窗口检测器

图 7.11 给出了差分方式下窗口比较的两个例子，这里假设 ADC2LT=0x10（+16d），ADC2GT=0xFF（-1d）。注意，在差分方式，转换码为 2 的补码表示的 8 位有符号整数，对应的信号变化范围是-VREF ~ VREF*(127/128)。在左侧的例子中，如果 ADC2 转换字（ADC2）位于由 ADC2GT 和 ADC2LT 定义的范围之内（即 0xFF<ADC2<0x0F），则会产生 AD2WINT 中断。在右侧的例子中，如果 ADC2 转换字（ADC2）位于由 ADC2GT 和 ADC2LT 定义的范围之外（即 ADC2<0xFF 或 ADC2>0x10），则会产生 AD2WINT 中断。

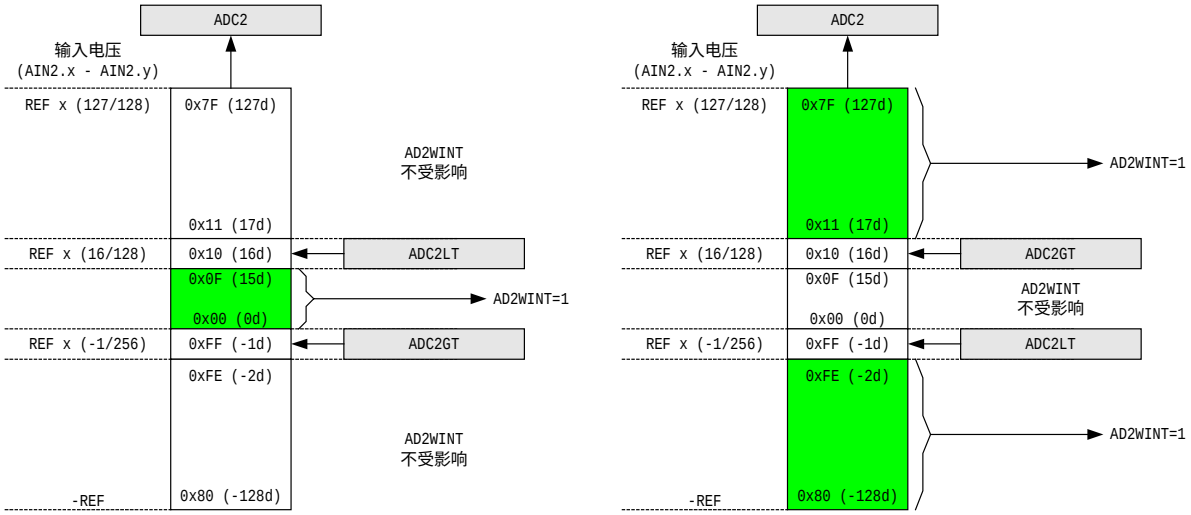


图 7.11 ADC2 窗口比较示例（差分方式）

SFR 页： 2
SFR 地址：0xC4

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

位 7-0： ADC2 下限（大于）数据字

图 7.12 ADC2GT: ADC2 下限数据寄存器

SFR 页： 2
SFR 地址：0xC6

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

位 7-0： ADC2 上限（小于）数据字

图 7.13 ADC2LT: ADC2 上限数据寄存器

表 7.1. ADC2 电气特性

VDD=3.0V, AV+=3.0V, VREF2=2.40V(REFBE=0), PGA2 增益=1, -40ºC ~ +85ºC（除非特别说明）

参　数	条　件	最小值	典型值	最大值	单 位
直流精度					
分辨率			8		位
积分非线性				±1	LSB
微分非线性	保证单调			±1	LSB
偏移误差			0.5±0.3		LSB
满度误差	差分方式		-1±02		LSB
偏移温度系数			10		ppm/℃
动态性能（10kHz 正弦波输入，满度值的 0 到-1dB，500ksps）					
信号与噪声失真比		45	47		dB
总谐波失真	到 5 次谐波		-51		dB
有效动态范围			52		dB
转换速率					
SAR 转换时钟				6	MHz
转换占用 SAR 时钟数		8			周期
跟踪/保持捕获时间		800			ns
最大转换速率				500	Ksps
模拟输入					
电压转换范围		0		VREF	V
输入电容			5		pF
电源指标					
电源电流（AV+给 ADC2 供电）	工作方式，500ksps		420	900	μA
电源抑制比			±0.3		mV/V

8. 12 位电压输出 DAC（仅 C8051F12x）

每个 C8051F12x 器件都有两个片内 12 位电压方式数/模转换器（DAC）。每个 DAC 的输出摆幅均为 0V 到（VREF-1LSB），对应的输入码范围是 0x000 到 0xFFF。可以用对应的控制寄存器 DAC0CN 和 DAC1CN 使能/禁止 DAC0 和 DAC1。在被禁止时，DAC 的输出保持在高阻状态，DAC 的供电电流降到 1μA 或更小。每个 DAC 的电压基准在 VREFD（C8051F120/2/4/6）或 VREF（C8051F121/3/5/7）引脚提供。注意：C8051F121/3/5/7 的 VREF 引脚可以由内部电压基准或一个外部源驱动。如果使用内部电压基准，为了使 DAC 输出有效，该基准必须被使能。有关配置 DAC 电压基准的详细信息，见“ 9. 电压基准 ”。

8.1 DAC 输出更新

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

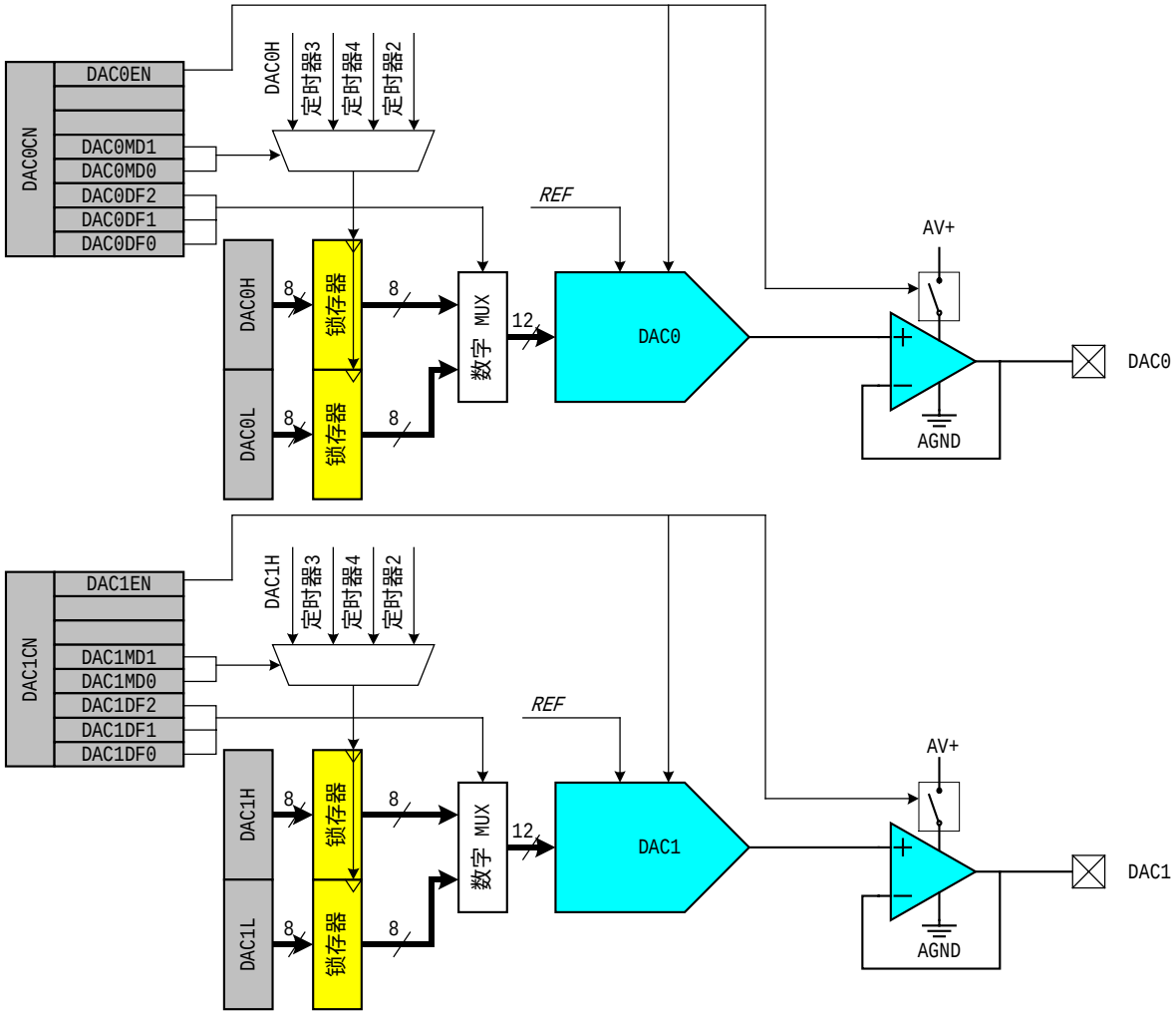


图 8.1 DAC 功能框图

8.1.1 根据软件命令更新输出

在缺省方式下（DAC0CN.[4:3] = ‘ 00 ’），DAC0 的输出在写 DAC0 数据寄存器高字节（DAC0H）时更新。注意：写 DAC0L 时数据被保持，对 DAC0 输出没有影响，直到对 DAC0H 的写操作发生。如果向 DAC 数据寄存器写入一个 12 位字，则 12 位的数据字被写到低字节（DAC0L）和高字节（DAC0H）数据寄存器。在写 DAC0H 寄存器后数据被锁存到 DAC0。因此，如果需要 12 位分辨率，应在写入 DAC0L 之后写 DAC0H。DAC 可被用于 8 位方式，这种情况是将 DAC0L 初始化一个所希望的数值（通常为 0x00），将数据只写入 DAC0H（有关在 16 位 SFR 空间内对 12 位 DAC 数据字格式化的说明见 8.2 节）。

8.1.2 基于定时器溢出的输出更新

在 ADC 转换操作中，ADC 转换可以由定时器溢出启动，不用处理器干预。与之类似，DAC 的输出更新也可以用定时器溢出事件触发。这一特点在用 DAC 产生一个固定采样频率的波形时尤其有用，可以消除中断延迟不同和指令执行时间不同对 DAC 输出时序的影响。当 DAC0MD 位（DAC0CN.[4:3]）被设置为 ‘ 01 ’、‘ 10 ’ 或 ‘ 11 ’ 时，对 DAC 两个数据寄存器（DAC0L 和 DAC0H）的写操作被保持，直到相应的定时器溢出事件（分别为定时器 3、定时器 4 或定时器 2）发生时 DAC0H:DAC0L 的内容才被复制到 DAC 输入锁存器，允许 DAC 数据改变为新值。

8.2 DAC 输出定标/调整

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

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

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
SFR 页： 0

位 7-0： DAC0 数据字高字节

图 8.2 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：0xD2
SFR 页： 0

位 7-0： DAC0 数据字低字节

图 8.3 DAC0L: DAC0 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
DAC0EN	-	-	DAC0MD1	DAC0MD0	DAC0DF2	DAC0DF1	DAC0DF0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xD4
SFR 页： 0

位 7： DAC0EN：DAC0 使能位
0：DAC0 禁止。DAC0 输出引脚被禁止，DAC0 处于低功耗停机方式
1：DAC0 使能。DAC0 输出有效；DAC0 处于工作状态。
位 6-5： 未用。读 = 00b；写 = 忽略。
位 4-3： DAC0MD1-0：DAC0 方式位。
00：DAC 输出更新发生在写 DAC0H 时。
01：DAC 输出更新发生在定时器 3 溢出时。
10：DAC 输出更新发生在定时器 4 溢出时。
11：DAC 输出更新发生在定时器 2 溢出时。
位 2-0: DAC0DF2-0：DAC0 数据格式位：

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

DAC0H								DAC0L							
				MSB											LSB

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

DAC0H								DAC0L							
			MSB											LSB	

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

DAC0H								DAC0L							
		MSB											LSB		

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

DAC0H								DAC0L							
	MSB											LSB			

1xx：高有效字节在 DAC0H[7:0]，低 4 位在 DAC0L[7:4]。

DAC0H								DAC0L							
MSB											LSB				

图 8.4 DAC0CN: DAC0 控制寄存器

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

SFR 地址：0xD3
SFR 页： 1

位 7-0： DAC1 数据字高字节

图 8.5 DAC1H: DAC1 高字节寄存器

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

SFR 地址：0xD2
SFR 页： 1

位 7-0： DAC1 数据字低字节

图 8.6 DAC1L: DAC1 低字节寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
DAC1EN	-	-	DAC1MD1	DAC1MD0	DAC1DF2	DAC1DF1	DAC1DF0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xD4
SFR 页： 1

位 7： DAC1EN：DAC1 使能位
0：DAC1 禁止。DAC1 输出引脚被禁止，DAC1 处于低功耗停机方式
1：DAC1 使能。DAC1 输出有效，DAC1 处于工作状态。
位 6-5： 未用。读 = 00b；写 = 忽略
位 4-3： DAC1MD1-0：DAC1 方式位。
00：DAC 输出更新发生在写 DAC1H 时。
01：DAC 输出更新发生在定时器 3 溢出时。
10：DAC 输出更新发生在定时器 4 溢出时。
11：DAC 输出更新发生在定时器 2 溢出时。
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[7:0]，低 4 位在 DAC1L[7:4]。

DAC1H								DAC1L							
MSB											LSB				

图 8.7 DAC1CN: DAC1 控制寄存器

表 8.1. DAC 电气特性。

VDD=3.0V, AV+=3.0V, VREF=2.40V(REFBE=0), 无输出负载（除非特别说明）

参 数	条 件	最小值	典型值	最大值	单 位
静态性能					
分辨率			12		位
积分非线性度			±1.5		LSB
微分非线性度				±1	LSB
输出噪声	无输出滤波器 100kHz 输出滤波器 10kHz 输出滤波器		250 128 41		μVrms
偏移误差	数据字= 0x014		±3	±30	mV
偏移误差温度系数			6		ppm/°C
满度误差			±20	±60	mV
满度误差温度系数			10		ppm/°C
VDD 电源抑制率			-60		dB
关断方式下的输出阻抗	DACnEN=0		100		kΩ
输出灌电流			300		μA
输出短路电流	数据字=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
输出电压摆幅		0		VREF-1LSB	V
启动时间			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

9. 电压基准

C8051F12x 和 C8051F13x 的电压基准电路选项与器件的功能和封装有关。

所有器件都集成了内部电压基准电路，由一个 1.2V、15ppm/°C（典型值）的带隙电压基准发生器和一个两倍增益的输出缓冲放大器组成。内部基准电压可以通过 VREF 引脚连到应用系统中的外部器件或图 9.1 所示的电压基准输入引脚。VREF 引脚对 AGND 的负载最大不能超过 200 μA。建议在 VREF 引脚与 AGND 之间接入 0.1 μF 和 4.7 μF 的旁路电容。

基准电压控制寄存器 REF0CN 使能/禁止内部基准发生器和内部温度传感器。REF0CN 中的 BIASE 位使能片内电压基准发生器，而 REFBE 位使能驱动 VREF 引脚的 2 倍增益缓冲放大器。当被禁止时，带隙基准和缓冲放大器消耗的电流小于 1μA（典型值），缓冲放大器的输出进入高阻状态。如果要使用内部带隙基准作为基准电压发生器，则 BIASE 和 REFBE 位必须被置‘ 1 ’。如果不使用内部基准，REFBE 位可以被清‘ 0 ’。注意：如果使用 ADC 或 DAC，则不管电压基准取自片内还是片外，BIASE 位必须被置为逻辑‘ 1 ’。如果既不使用 ADC 也不使用 DAC，则这两位都可以被清‘ 0 ’以节省功耗。

当被使能时，内部温度传感器连接到 ADC0 输入模拟多路器的最高序号输出。REF0CN 中的 TEMPE 位用于使能/禁止温度传感器。当被禁止时，温度传感器为缺省的高阻状态，此时对温度传感器的任何 A/D 测量结果都是无意义的。

表 9.1 给出了内部电压基准的电气特性。

9.1 C8051F120/2/4/6 的电压基准配置

对于 C8051F120/2/4/6 器件，REF0CN 寄存器还允许为 ADC0 和 ADC2 选择电压基准，如图 9.2 所示。REF0CN 寄存器中的 AD0VRS 和 AD2VRS 位分别用于选择 ADC0 和 ADC2 的电压基准源。电路为控制 ADC 和 DAC 模块工作提供了灵活性。有三个电压基准输入引脚，允许每个 ADC 和两个 DAC 使用外部电压基准或片内电压基准输出（通过外部连接）。通过配置 VREF 模拟开关，ADC0 还可以使用 DAC0 的内部输出作为基准，ADC2 可以使用模拟电源电压作为基准，如图 9.1 所示。

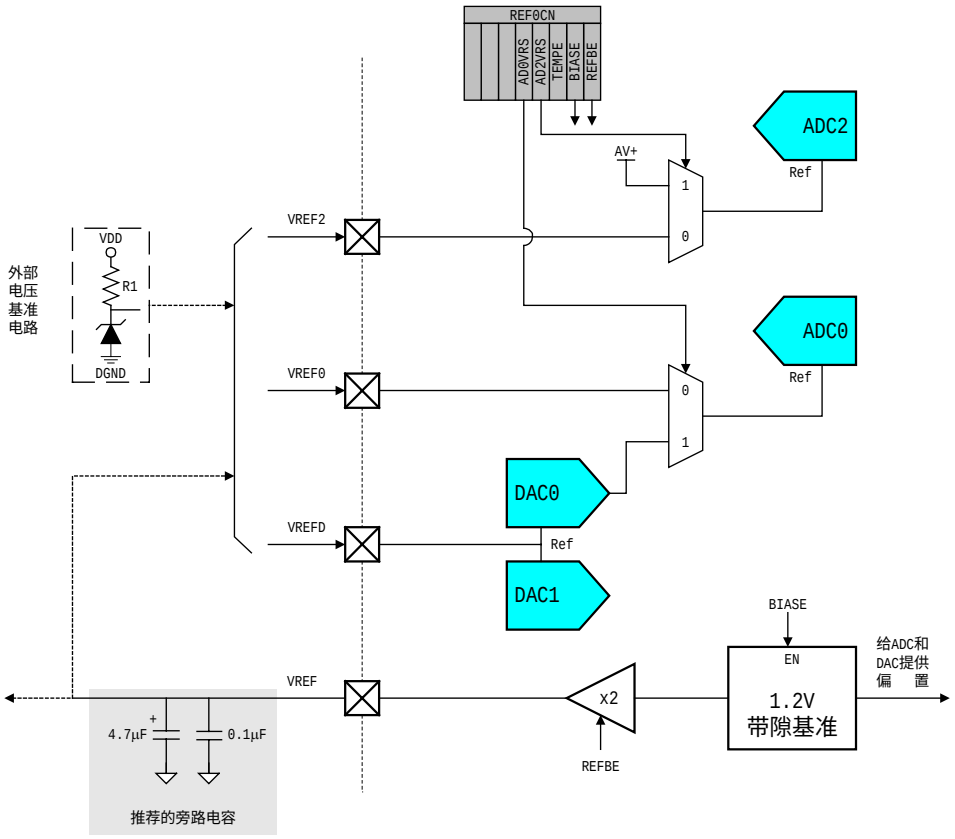


图 9.1 电压基准功能框图（C8051F120/2/4/6）

SFR 页： 0
SFR 地址：0xD1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	AD0VRS	AD2VRS	TEMPE	BIASE	REFBE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-5： 未用。读 = 000b，写 = 忽略。
位 4： AD0VRS：ADC0 电压基准选择位
0：ADC0 电压基准取自 VREF0 引脚。
1：ADC0 电压基准取自 DAC0 输出。
位 3： AD2VRS：ADC2 电压基准选择位
0：ADC2 电压基准取自 VREF2 引脚。
1：ADC2 电压基准取自 AV+。
位 2： TEMPE：温度传感器使能位
0：内部温度传感器关闭。
1：内部温度传感器工作。
位 1： BIASE：ADC/DAC 偏压发生器使能位（使用 ADC 和 DAC 时该位必须为 1）
0：内部偏压发生器关闭。
1：内部偏压发生器工作。
位 0： REFBE：内部电压基准缓冲器使能位
0：内部电压基准缓冲器关闭。
1：内部电压基准缓冲器工作。内部电压基准输出到 VREF 引脚。

图 9.2 REF0CN: 电压基准控制寄存器（C8051F120/2/4/6）

9.2 C8051F121/3/5/7 的电压基准

对于 C8051F121/3/5/7 器件，REF0CN 寄存器还允许为 ADC0 和 ADC2 选择电压基准，如图 9.4 所示。REF0CN 寄存器中的 AD0VRS 和 AD2VRS 位分别用于选择 ADC0 和 ADC2 的电压基准源。VREFA 引脚为 ADC0 和 ADC2 提供电压基准输入，该引脚可以被连接到一个外部精密基准或内部电压基准。通过配置 VREF 模拟开关，ADC0 还可以使用 DAC0 的内部输出作为基准，ADC2 可以使用模拟电源电压作为基准，如图 9.3 所示。

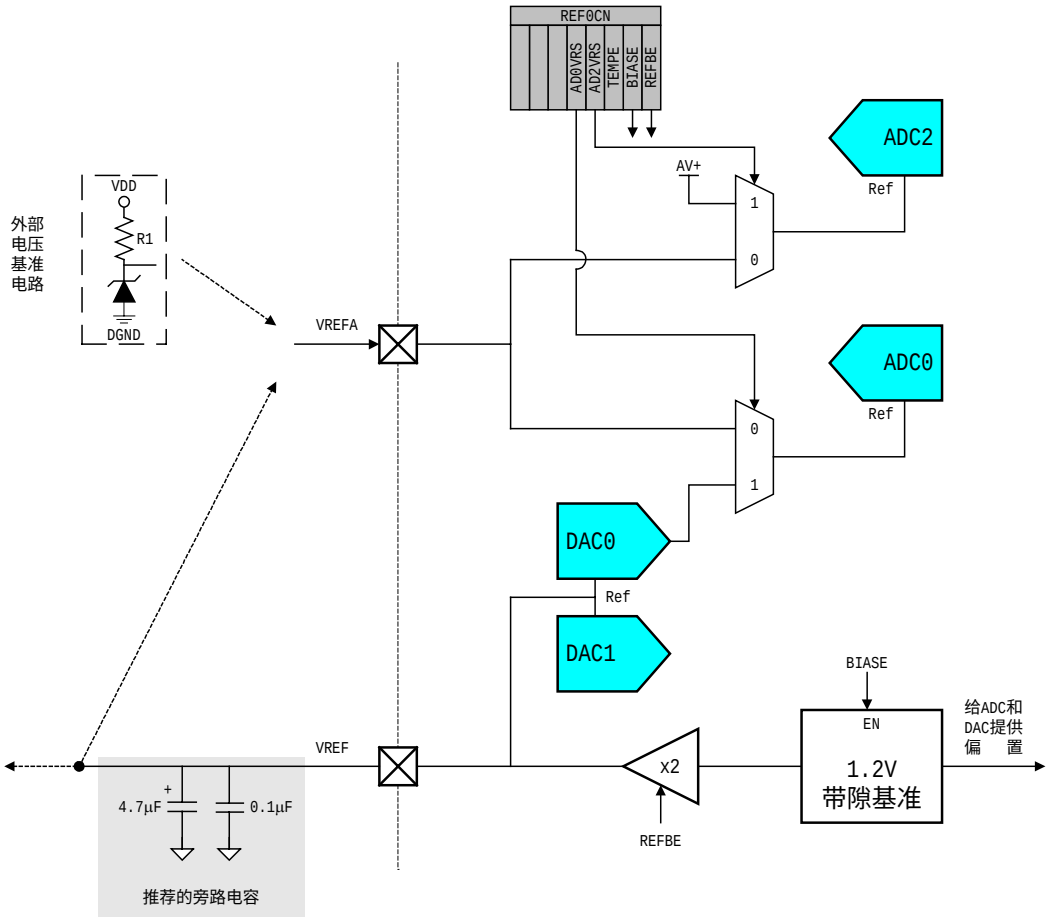


图 9.3 电压基准功能框图（C8051F121/3/5/7）

SFR 页： 0
SFR 地址：0xD1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	AD0VRS	AD1VRA	TEMPE	BIASE	REFBE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-5： 未用。读 = 000b，写 = 忽略。
位 4： AD0VRS：ADC0 电压基准选择位
0：ADC0 电压基准取自 VREFA 引脚。
1：ADC0 电压基准取自 DAC0 输出。
位 3： AD2VRS：ADC2 电压基准选择位
0：ADC2 电压基准取自 VREFA 引脚。
1：ADC2 电压基准取自 AV+。
位 2： TEMPE：温度传感器使能位
0：内部温度传感器关闭。
1：内部温度传感器工作。
位 1： BIASE：ADC/DAC 偏压发生器使能位（使用 ADC 或 DAC 时该位必须为 1）
0：内部偏压发生器关闭。
1：内部偏压发生器工作。
位 0： REFBE：内部电压基准缓冲器使能位
0：内部电压基准缓冲器关闭。
1：内部电压基准缓冲器工作。内部电压基准提供从 VREF 引脚输出。

图 9.4 REF0CN: 电压基准控制寄存器（C8051F121/3/5/7）

9.3 C8051F130/1/2/3 的电压基准

对于 C8051F130/1/2/3 器件，VREF0 引脚为 ADC0 提供电压基准输入，该引脚可以被连接到一个外部精密基准或内部电压基准，如图 9.5 所示。对 C8051F130/1/2/3 中的 REF0CN 寄存器的说明见图 9.6

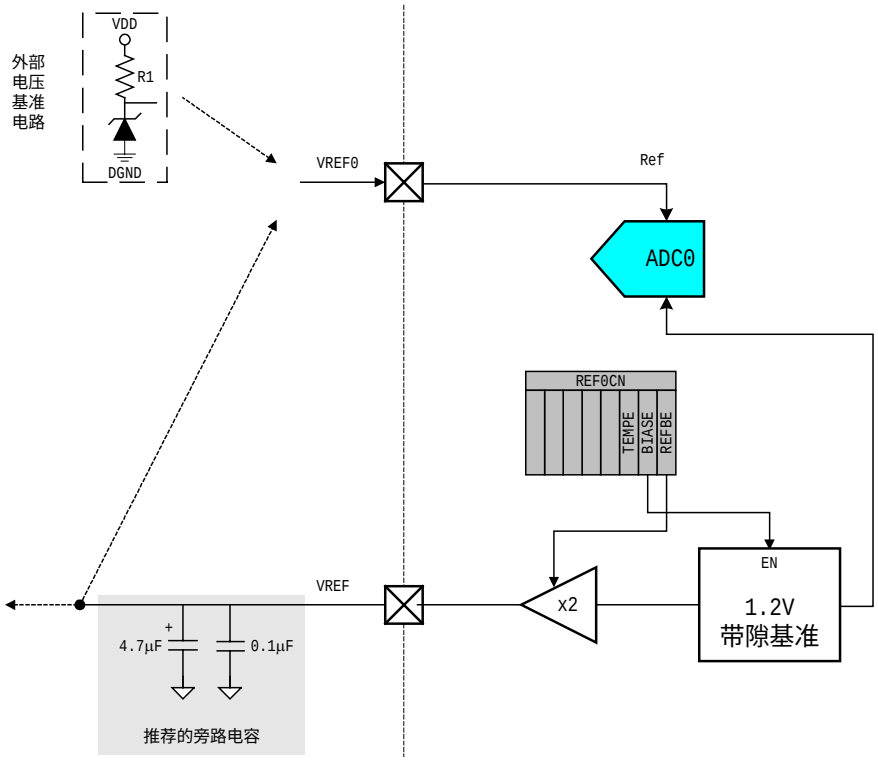


图 9.5 电压基准功能框图（C8051F130/1/2/3）

SFR 页： 0
SFR 地址：0xD1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	保留	保留	TEMPE	BIASE	REFBE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-5： 未用。读 = 000b，写 = 忽略。
位 4-3： 保留。必须写 0。
位 2： TEMPE：温度传感器使能位
0：内部温度传感器关闭。
1：内部温度传感器工作。
位 1： BIASE：ADC/DAC 偏压发生器使能位（使用 ADC 或 VREF 时该位必须为 1）
0：内部偏压发生器关闭。
1：内部偏压发生器工作。
位 0： REFBE：内部电压基准缓冲器使能位
0：内部电压基准缓冲器关闭。
1：内部电压基准缓冲器工作。内部电压基准提供从 VREF 引脚输出。

图 9.6 REF0CN: 电压基准控制寄存器（C8051F130/1/2/3）

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
模拟偏压发生器供电电流	BIASE = 1		100		µA
内部基准（REFBE＝1）					
输出电压	环境温度 25℃	2.36	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 ~ 200µA 到 AGND		0.5		ppm/µA
VREF 开启时间 1	4.7µF 钽电容和 0.1µF 陶瓷旁路电容		2		ms
VREF 开启时间 2	0.1µF 陶瓷旁路电容		20		µs
VREF 开启时间 3	无旁路电容		10		µs
基准缓冲器电源电流			40		µA
电源抑制比			140		ppm/V
外部基准（REFBE＝0）					
输入电压范围		1.00		(AV+)-0.3	V
输入电流			0	1	µA

表 10.1 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℃	2.36	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 ~ 200μA 到 AGND		0.5		ppm/μA
VREF 开启时间 1	4.7μF 钽电容和 0.1μF 陶瓷旁路电容		2		ms
VREF 开启时间 2	0.1μF 陶瓷旁路电容		20		μs
VREF 开启时间 3	无旁路电容		10		μs
基准缓冲器电源电流			40		μA
电源抑制比			140		ppm/V
外部基准（REFBE＝0）					
输入电压范围		1.00		(AV+)-0.3	V
输入电流			0	1	μA

10. 比较器

C8051F12x 和 C8051F13x 器件内部有两个可编程电压比较器，如图 10.1 所示。每个比较器都有专用的输入引脚。每个比较器的输出都可以经 I/O 交叉开关连到外部引脚。当被分配了封装引脚时，每个比较器输出都可以被编程为工作在漏极开路或推挽方式。关于交叉开关和端口初始化的详细信息见“ 18.1 端口 0 ~端口 3 优先权交叉开关译码器 ”。

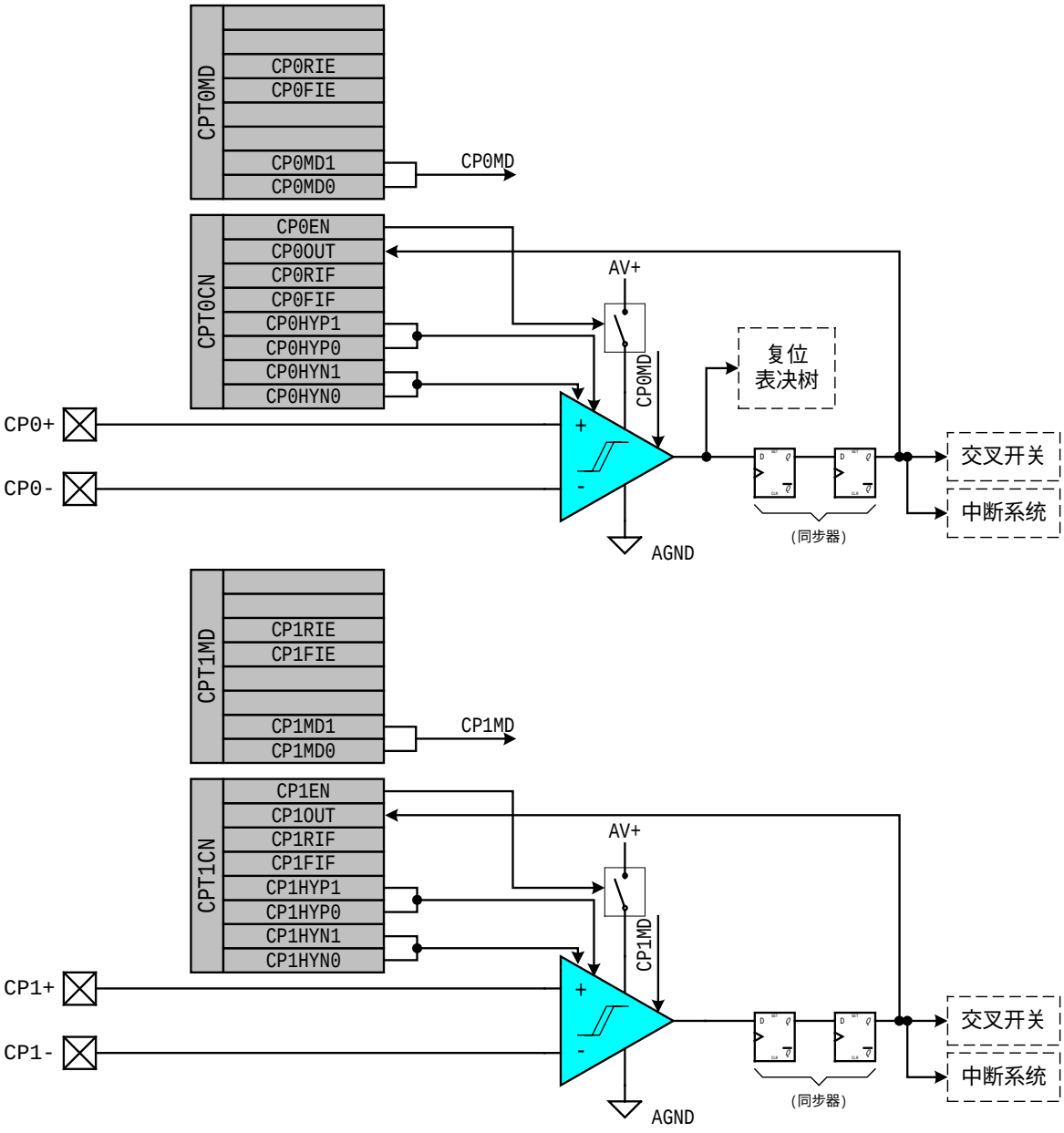


图 10.1 比较器功能框图

在比较器输出的上升沿和/或下降沿都可以产生中断。（有关中断允许和优先级控制的内容见“ 11.7 中断系统 ”）。比较器的下降沿中断置‘ 1 ’CP0FIF 标志，比较器 0 的上升沿中断置‘ 1 ’CP0RIF 标志。这些位一旦被置‘ 1 ’，将一直保持‘ 1 ’状态直到被软件清除。可以在任意时刻通过读取 CP0OUT 位得到比较器 0 的输出状态。通过置 ‘ 1 ’ CP0EN 位使能比较器 0，通过清除该位禁止比较器 0。比较器 0 还可被配置为复位源，详见“ 13.5 比较器 0 复位 ”。

注意，比较器在被使能后有一个上电时间（见表 10.1），在此时间后比较器输出达到稳定。在比较器上电后，上升沿和下降沿标志位的状态是不确定的，因此应在比较器中断被使能前或比较器被配置为复位源之前将这写标志位清 0。

比较器 0 的响应时间可以用软件通过 CPT0MD 寄存器的 CP0MD1-0 位编程（见图 10.4）。选择较长的响应时间可以减少比较器 0 消耗的电流。比较器的详细时序和电流消耗指标见表 10.1。

每个比较器的回差电压都可以通过对应的比较器控制寄存器（CPT0CN 和 CPT1CN 分别对应比较器 0 和比较器 1）用软件编程。用户既可以对回差电压值（这里指输入电压）编程，也可以对门限电压两侧的正向和负向回差对称度编程。比较器的输出可以被软件查询，也可以作为中断源。每个比较器可以被单独使能或禁止（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，它的中断能力被中止，电源电流降到小于 100 nA。比较器的输入可以承受-0.25V 到(AV+) + 0.25V 的外部驱动电压而不至损坏或发生工作错误。

使用比较器 0 控制寄存器 CPT0CN（见图 10.3）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。类似地，通过设置 CP0HYP 位决定正向回差电压值。

比较器 1 的操作与比较器 0 完全相同，只是比较器 1 不能被配置为复位源。比较器 1 受 CPT1CN 寄存器（图 10.4）控制。表 10.1 给出了比较器的详细电特性。

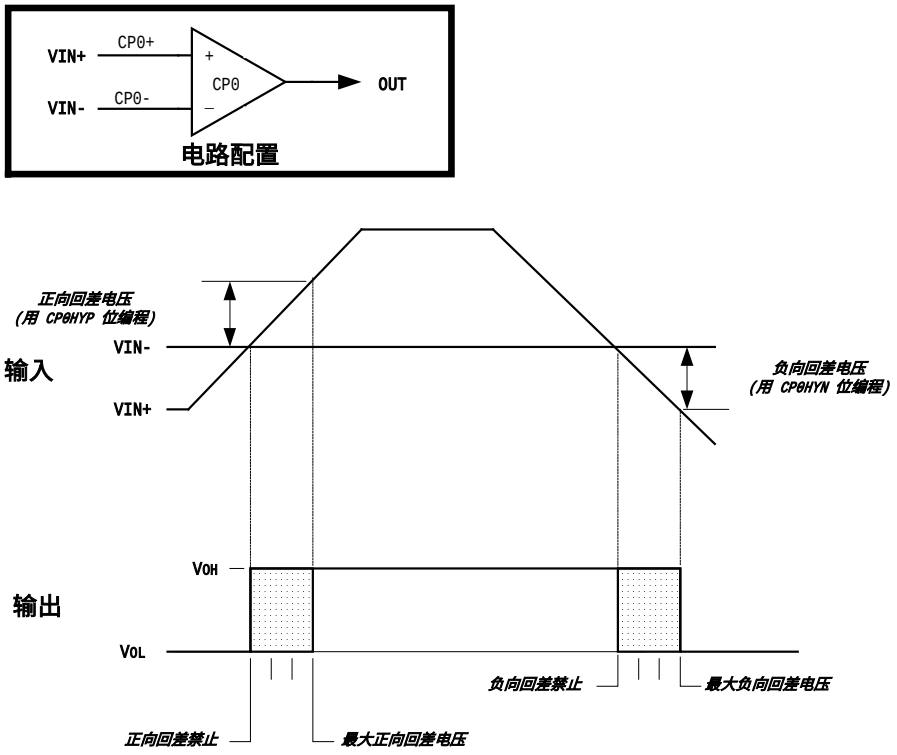


图 10.2 比较器回差电压曲线

SFR 页： 1
SFR 地址：0x88

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

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

位 6： CP0OUT：比较器 0 输出状态标志
0：电压值 CP0+ < CP0-
1：电压值 CP0+ > CP0-

位 5： CP0RIF：比较器 0 上升沿中断标志
0：自该标志位被清除后，没有发生过比较器 0 上升沿中断
1：自该标志位被清除后，发生了比较器 0 上升沿中断

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

位 3-2： CP0HYP1-0：比较器 0 正向回差电压控制位
00：禁止正向回差电压
01：正向回差电压=5mV
10：正向回差电压=10mV
11：正向回差电压=15mV

位 1-0： CP0HYN1-0：比较器 0 负向回差电压控制位
00：禁止负向回差电压
01：负向回差电压=5mV
10：负向回差电压=10mV
11：负向回差电压=15mV

图 10.3 CPT0CN: 比较器 0 控制寄存器

SFR 页： 1
SFR 地址：0x89

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

位 7-6： 未用。读 = 00b，写 = 忽略。
位 5： CP0RIE：比较器 0 上升沿中断使能位。
0：比较器 0 上升沿中断禁止。
1：比较器 0 上升沿中断使能。
位 4： CP0FIE：比较器 0 下降沿中断使能位。
0：比较器 0 下降沿中断禁止。
1：比较器 0 下降沿中断使能。
位 3-2： 未用。读 = 00b，写 = 忽略。
位 1-0： CP0MD1-CP0MD0：比较器 0 方式选择位。
这些位选择比较器 0 的响应时间。

方式	CP0MD1	CP0MD0	说 明
0	0	0	最快响应时间
1	0	1	-
2	1	0	-
3	1	1	最低功耗

图 10.4 CPT0MD: 比较器 0 方式选择寄存器

SFR 页： 2
SFR 地址：0x88

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

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

位 6： CP1OUT：比较器 1 输出状态位
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=5mV
10：正向回差电压=10mV
11：正向回差电压=15mV

位 1-0： CP1HYN1-0：比较器 1 负向回差电压控制位
00：禁止负向回差电压
01：负向回差电压=5mV
10：负向回差电压=10mV
11：负向回差电压=15mV

图 10.5 CPT1CN: 比较器 1 控制寄存器

SFR 页： 2
SFR 地址：0x89

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	CP1RIE	CP1FIE	-	-	CP1MD1	CP1MD0	00000010
位7	位6	位5	位4	位3	位2	位1	位0	

位 7-6： 未用。读 = 00b，写 = 忽略。
位 5： CP1RIE：比较器 1 上升沿中断使能位。
0：比较器 1 上升沿中断禁止。
1：比较器 1 上升沿中断使能。
位 4： CP0FIE：比较器 1 下降沿中断使能位。
0：比较器 1 下降沿中断禁止。
1：比较器 1 下降沿中断使能。
位 3-2： 未用。读 = 00b，写 = 忽略。
位 1-0： CP1MD1-CP1MD0：比较器 1 方式选择位。
这些位选择比较器 1 的响应时间。

方式	CP1MD1	CP1MD0	说 明
0	0	0	最快响应时间
1	0	1	-
2	1	0	-
3	1	1	最低功耗

图 10.6 CPT1MD: 比较器 1 方式选择寄存器

表 10.1. 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： 方式 0，Vcm† = 1.5V	(CPn+) - (CPn-) = 100mV		100		nS
	(CPn+) - (CPn-) = -100mV		250		nS
响应时间： 方式 1，Vcm† = 1.5V	(CPn+) - (CPn-) = 100mV		175		nS
	(CPn+) - (CPn-) = -100mV		500		nS
响应时间： 方式 2，Vcm† = 1.5V	(CPn+) - (CPn-) = 100mV		320		nS
	(CPn+) - (CPn-) = -100mV		1100		nS
响应时间： 方式 3，Vcm† = 1.5V	(CPn+) - (CPn-) = 100mV		1050		nS
	(CPn+) - (CPn-) = -100mV		5200		nS
共模抑制比			1.5	4	mV/V
正向回差电压 1	CPnHYP1-0 = 00		0	1	mV
正向回差电压 2	CPnHYP1-0 = 01	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）	方式 0		7.6		μA
	方式 1		3.2		
	方式 2		1.3		
	方式 3		0.4		

†Vcm 是 CPn+和 CPn-上的共模电压。

11. CIP-51 微控制器

MCU 系统控制器的内核是 CIP-51 微控制器。CIP-51 与 MCS-51™ 指令集完全兼容，可以使用标准 803x/805x 的汇编器和编译器进行软件开发。该系列 MCU 具有标准 8051 的所有外设部件，包括 5 个 16 位的计数器/定时器（详见第 23 章）、两个全双工 UART（详见第 21 章和第 22 章）、256 字节内部 RAM、128 字节特殊功能寄存器（SFR）地址空间及 8/4 个 8 位宽的 I/O 端口（详见第 19 章）。CIP-51 还包含片内调试硬件（详见第 25 章）和与 MCU 直接接口的模拟和数字子系统，在一个集成电路内提供了完整的数据采集或控制系统解决方案。

CIP-51 微控制器内核除了具有标准 8051 的组织结构和外设以外，另有增加的定制外设和功能，大大增强了它的处理能力（见图 11.1 的原理框图）。CIP-51 具有下列特点：

- 与 MCS-51 指令集完全兼容
- 使用片内 PLL 可获得 100 或 50MIPS 的峰值性能
- 256 字节内部 RAM
- 8/4 个 8 位 I/O 端口
- 扩展的中断处理系统
- 复位输入
- 电源管理方式
- 片内调试逻辑
- 程序和数据存储器安全

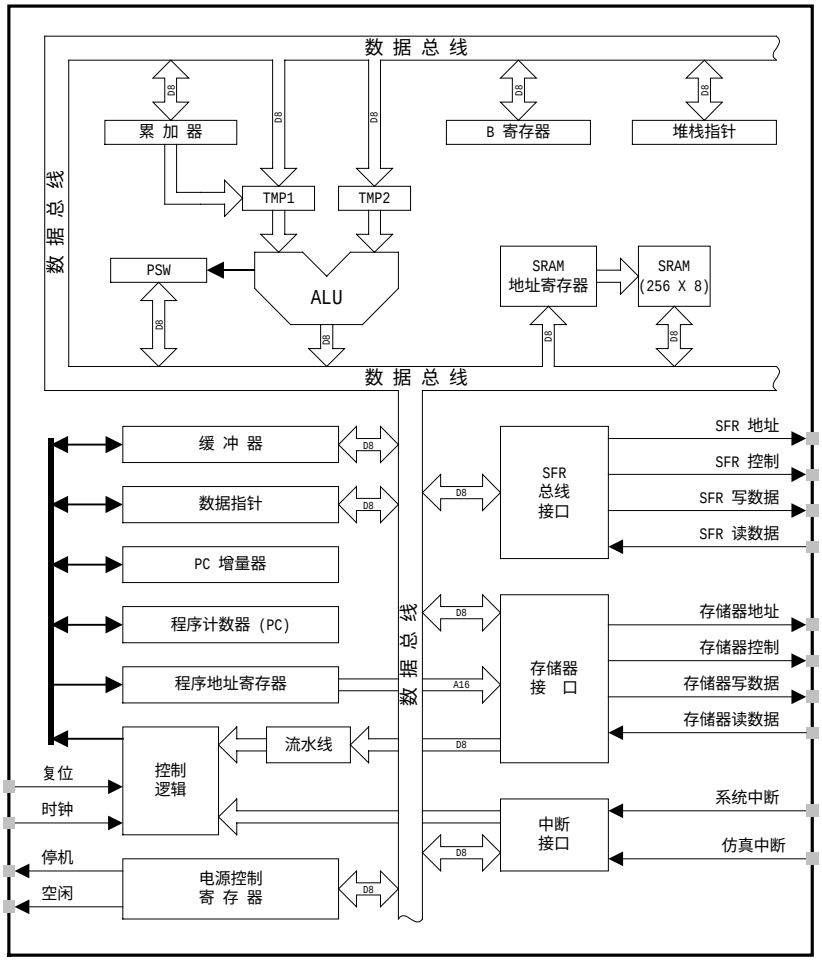


图 11.1 CIP-51 原理框图

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 100MHz 时，它的峰值速度达到 100MIPS。CIP-51 共有 111 条指令。下表列出了指令条数与执行时所需的系统时钟周期数的关系。

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

编程和调试支持

C8051F12x系列MCU提供了JTAG串行接口，通过该接口能对FLASH程序存储器进行在系统编程并可与片内调试支持电路通信。应用程序可以使用MOVC和MOVX指令对可再编程FLASH进行读或改写，每次只能读或写一个字节。这一特性允许将程序存储器用于非易失性数据存储以及在软件控制下更新程序代码。

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

CIP-51 有 Silicon Lab 集成产品公司和第三方供应商的开发工具支持。Silicon Lab 提供了一个集成开发环境（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。

11.1 指令集

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

11.1.1 指令和 CPU 时序

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

由于 CIP-51 采用了流水线结构，大多数指令执行所需的时钟周期数与指令的字节数一致。条件转移指令在不发生转移时的执行周期数比发生转移时少一个。表 11.1 给出了 CIP-51 指令一览表，包括每条指令的助记符、字节数和时钟周期数。

11.1.2 MOVX 指令和程序存储器

在 CIP-51 中，MOVX 指令有三种作用：访问片内 XRAM、访问片外 XRAM 和访问片内 FLASH 程序存储器。CIP-51 的 FLASH 访问特性提供了由用户程序更新程序代码和将程序存储器空间用于非易失性数据存储的机制（见“ 15. FLASH 存储器 ”）。通过外部存储器接口，可用 MOVX 指令快速访问片外 XRAM（或存储器编址的外设）。详见“ 17. 外部数据存储器接口和片内 XRAM ”。

表 11.1 CIP-51 指令集

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

表 11.1 CIP-51 指令集（续）

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

表 11.1 CIP-51 指令集（续）

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

寄存器、操作数和寻址方式说明：

Rn – 当前选择的寄存器区的寄存器 R0-R7。

@Ri – 通过寄存器 R0-R1 间接寻址的数据 RAM 地址。

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

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

#data – 8 位立即数

#data16 – 16 位立即数

bit – 数据 RAM 或 SFR 中的直接寻址位

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

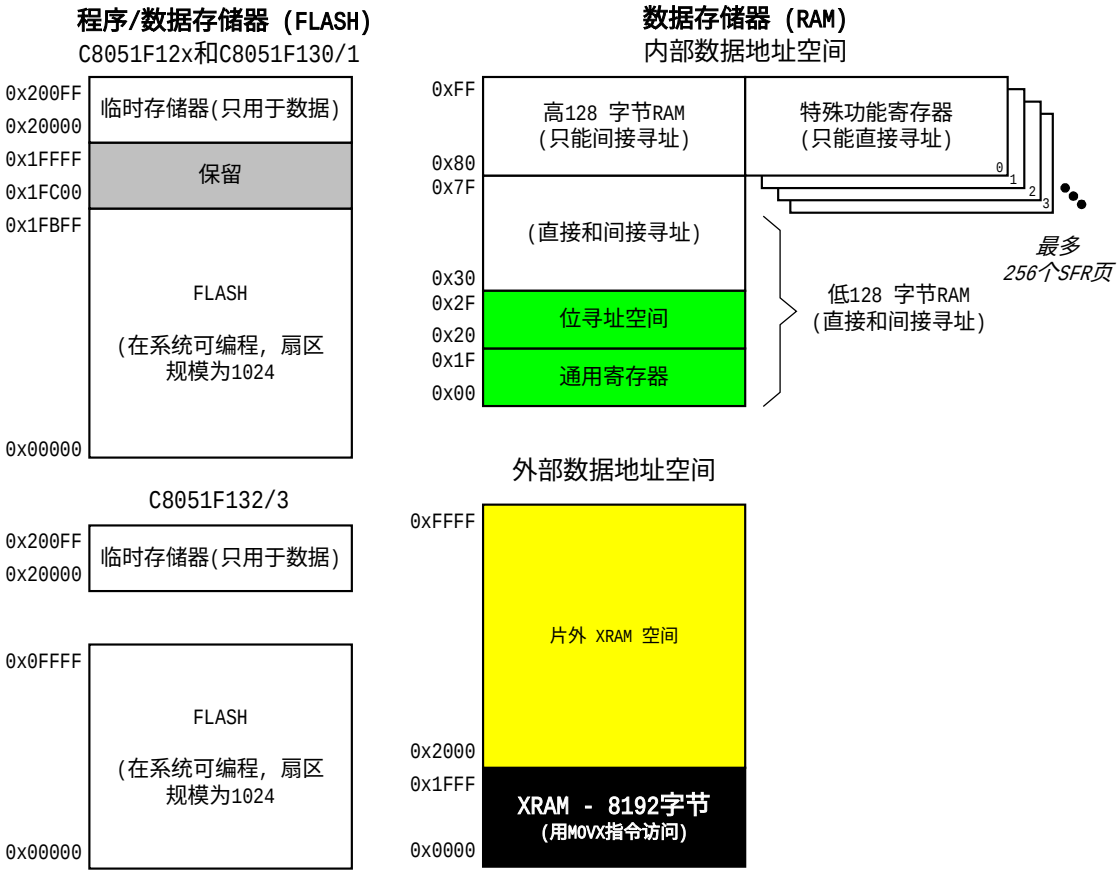
addr16 – LCALL 或 LJMP 使用的 16 位目的地址。目的地址可以是 64K 程序存储器空间内的任何位置。

有一个未使用的操作码（0xA5），它执行与 NOP 指令相同的功能。

11.2 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 128KB（C8051F12x 和 C8051F130/1）或 64KB（C8051F132/3）的内部程序存储器地址空间。CIP-51 的存储器组织如图 11.2 所示。

图 11.2 存储器结构图



11.2.1 程序存储器

C8051F12x 和 C8051F130/1 有 128KB 的程序存储器空间。MCU 在这个程序存储器空间中实现了可在系统编程的 FLASH 存储器，组织成 4 个 32KB 的程序存储块。地址在 0x0000 到 0x7FFF 之间的 32K 字节为公共程序存储块（块 0）。其它 3 个高地址程序存储块（块 1、块 2 和块 3）都映射到地址 0x8000 ~ 0xFFFF，由 PSBANK 寄存器中的块选择位选择当前存储块，见图 11.3。IFBANK 位选择哪一个高地址存储块用于程序执行，而 COBANK 位选择用于直接读写的存储块。注意：位于块 3 的 1024 个存储器字节（0x1FC00 ~ 0X1FFFF）为保留区，不能用于用户程序或数据存储。C8051F132/3 用可在系统编程 FLASH 存储器实现了 64KB 的程序存储器空间，按一个连续块组织，地址为 0x0000 ~ 0xFFFF。

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

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	COBANK		-	-	IFBANK		00010001
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xB1
SFR 页： 所有页

位 7-6： 保留

位 7： COBANK：常量操作存储块选择位。
这两位选择常量操作（MOVC 和 FLASH MOVX）地址在 0x8000 ~0xFFFF 范围的 FLASH 存储块。当访问临时存储区时，这些位被忽略（见 15. FLASH 存储器）。
00：常量操作指向存储块 0（注意，块 0 也映射到地址 0x0000 ~0x7FFF）。
01：常量操作指向存储块 1。
10：常量操作指向存储块 2。
11：常量操作指向存储块 3。

位 3-2： 保留

位 1-0： IFBANK：取指操作存储块选择位。
这两位选择取指操作（地址在 0x8000 ~0xFFFF 范围）的 FLASH 存储块。这两位只能由位于块 0 的程序改写（见图 11.4）。
00：从存储块 0 取指令（注意，块 0 也映射到地址 0x0000 ~0x7FFF）。
01：从存储块 1 取指令。
10：从存储块 2 取指令。
11：从存储块 3 取指令 3。

注意：对于 C8051F132/3，COBANK 和 IFBANK 位应保持默认设置‘ 01 ’，以保证器件工作正确。

图 11.3 PSBANK：程序存储器空间块选择寄存器

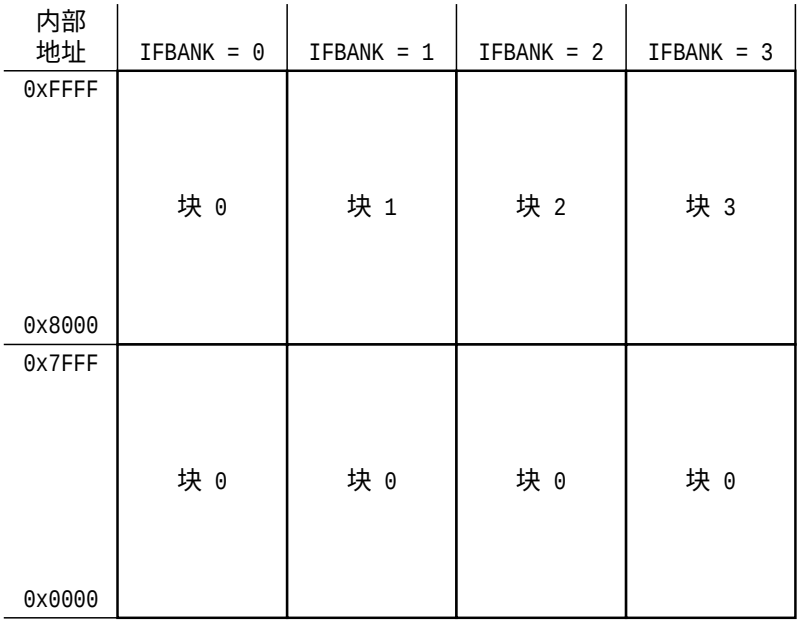


图 11.4　取指操作时的存储器映射（仅限于 128KB FLASH）

11.2.2 数据存储器

CIP-51 的数据存储器空间中有 256 字节的内部 RAM，位于地址 0x00 到 0xFF 的地址空间。数据存储器中的低 128 字节用于通用寄存器和临时存储器。可以用直接或间接寻址方式访问数据存储器的低 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 11.2 给出了 CIP-51 数据存储器组织的示意图。

11.2.3 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 11.18 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。

11.2.4 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，  22 . 3 h
```

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。

11.2.5 堆栈

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

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

11.2.6 特殊功能寄存器

从 0x80 到 0xFF 的直接寻址存储器空间为特殊功能寄存器（SFR）。SFR 提供对 CIP-51 的资源和外设的控制及 CIP-51 与这些资源和外设之间的数据交换。CIP-51 具有标准 8051 中的全部 SFR，还增加了一些用于配置和访问专有子系统的 SFR。这就允许在保证与 MCS-51™ 指令集兼容的前提下增加新的功能。表 11.2 列出了 CIP-51 系统控制器中的全部 SFR。

任何时刻用直接寻址访问从 0x80 到 0xFF 的存储器空间将访问特殊功能寄存器（SFR）。地址以 0x0 或 0x8 结尾的 SFR（例如 P0、TCON、P1、SCON、IE 等）既可以按字节寻址也可以按位寻址，所有其它 SFR 只能按字节寻址。SFR 空间中未使用的地址保留为将来使用，访问这些地址会产生不确定的结果，应予避免。有关每个寄存器的详细说明请参见本数据表的相关部分（表 11.3 中已标明）。

SFR 分页机制

CIP-51 实现了 SFR 分页机制，允许器件将很多 SFR 映射到 0x80 ~ 0xFF 这个存储器地址空间。SFR 存储器空间有 256 页。0x80 ~ 0xFF 的每个存储器地址都可以访问多达 256 页。C8051F12x 器件使用 5 个 SFR 页：0、1、2、3 和 F。使用特殊功能寄存器页选择寄存器 SFRPAGE 来选择 SFR 页（见图 11.2）。读和写一个 SFR 的步骤如下：

1．用 SFRPAGE 寄存器选择相应的 SFR 页号。
2．用直接寻址方式读或写特殊功能寄存器（MOV 指令）。

中断和 SFR 分页

当一个中断发生时，SFR 页寄存器会自动切换到引起中断的那个标志位所在 SFR 页。这种自动 SFR 页切换功能减轻了从中断服务程序切换 SFR 页的负担。在执行 RETI 指令时，中断前使用的 SFR 页会被自动恢复。这是通过一个 3 字节的 *SFR 页堆栈*来实现的。位于该堆栈顶部的是 SFRPAGE，即当前 SFR 页；SFR 页堆栈的第二个字节是 SFRNEXT；第三个或者说 SFR 堆栈底部的字节是 SFRLAST。发生中断时，当前 SFR 页的值被压入到 SFRNEXT 字节，SFRNEXT 的值被压入到 SFRLAST。然后硬件将包含该中断对应标志位的 SFR 页装入 SFRPAGE。在中断返回时执行 SFR 页

堆栈出栈操作，SFRNEXT 的值返回到 SFRPAGE 寄存器，因此不需要软件干预即可恢复 SFR 页的上下文。SFRLAST 中的值（如果堆栈低部没有 SFR 页值，则该值为 0x00）被保存到 SFRNEXT 寄存器。如果需要，可以在中断服务程序中修改 SFRNEXT 和 SFRLAST 的值，以使 CPU 在执行 RETI 指令（中断返回）时返回到不同的 SFR 页。修改 SFR 页堆栈中的寄存器并不引起压栈或出栈操作。只有中断调用和返回才会导致对 SFR 页堆栈的压栈/出栈操作。

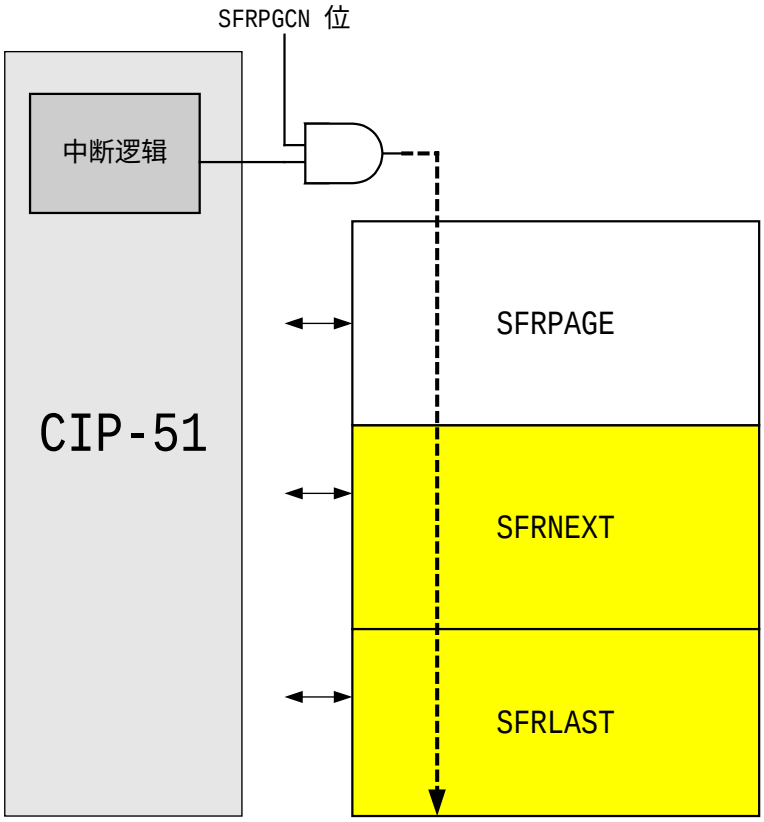


图 11.5　SFR 页堆栈

可以根据需要使能或禁止中断时的 SFR 页自动硬件切换功能，这是通过对位于 SFR 页控制寄存器（SFRPGCN）中的自动页控制使能位编程来实现的。系统复位后该功能缺省为使能状态。这样，自动切换功能被使能，除非用软件禁止。

表 11.2 以 SFR 存储器映像的形式给出了 SFR 单元（地址和 SFR 页）一览表。该表中的每个存储器单元都有指示其所在 SFR 页的列。注意：某些 SFR 可以从所有的 SFR 页访问，在表中表示为“所有页”。例如，端口 I/O 寄存器 P0、P1、P2 和 P3 都有“所有页”标志，表示可以从所有的 SFR 页访问这些 SFR 寄存器，而不管 SFRPAGE 寄存器的值为何。

SFR 页堆栈示例

下面给出了中断过程中 SFR 页堆栈操作的一个例子。

在本例中，SFR 页控制为缺省的使能状态（即 SFRPGEN = 1），CIP-51 执行向端口 5（SFR“ P5 ”，位于地址 0xD8，SFR 页为 0x0F）的写入操作。器件还使用可编程计数器阵列（PCA）并用 8 位 ADC（ADC2）窗口比较器监视一个电压值。系统用 PCA 中断服务程序（ISR）实现一个关键控制功能，因此 PCA 中断被使能并被设置为高优先级。ADC2 用于监视一个重要性稍差的电压，但为了减少软件开销，我们使用窗口检测器和相关的 ISR，窗口检测器中断被设置为低优先级。此时，SFR 页被设置为访问端口 5 SFR（SFRPAGE = 0x0F）。见下面的图 11.6。

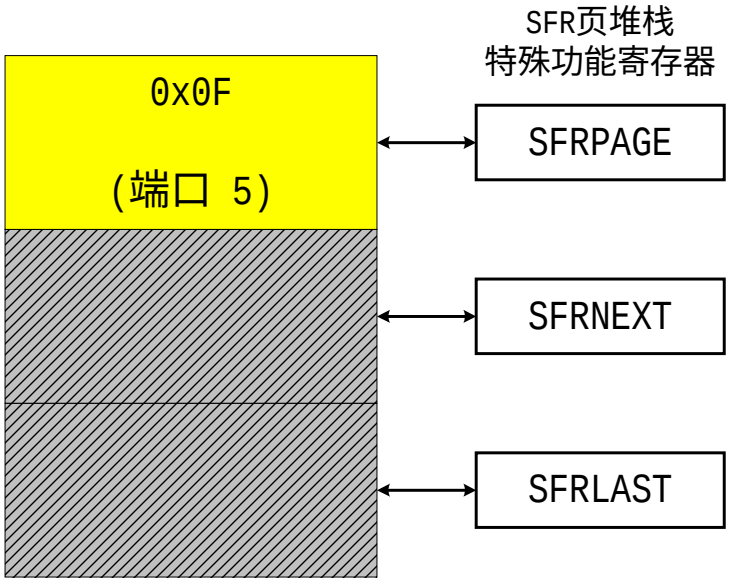


图 11.6　SFR 页堆栈：用 SFR 页 0x0F 访问端口 5

当 CIP-51 执行代码（在本例中为向端口 5 写值）时，发生了 ADC2 窗口比较器中断。CIP-51 转去执行 ADC2 窗口比较器 ISR，将当前 SFR 页值（SFR 页 0x0F）压入到 SFR 页堆栈的 SFRNEXT。然后，访问 ADC2 的 SFR 所需要的 SFR 页被自动装入 SFRPAGE 寄存器（SFR 页 0x02）。SFRPAGE 是 SFR 页堆栈的“ 栈顶 ”。软件现在可以访问 ADC2 的特殊功能寄存器。软件可以在执行 ADC2 ISR 的任何时刻通过向 SFRPAGE 寄存器写一个新值来切换到任何一个 SFR 页，以访问不在 SFR 页 0x02 的特殊功能寄存器。见下面的图 11.7。

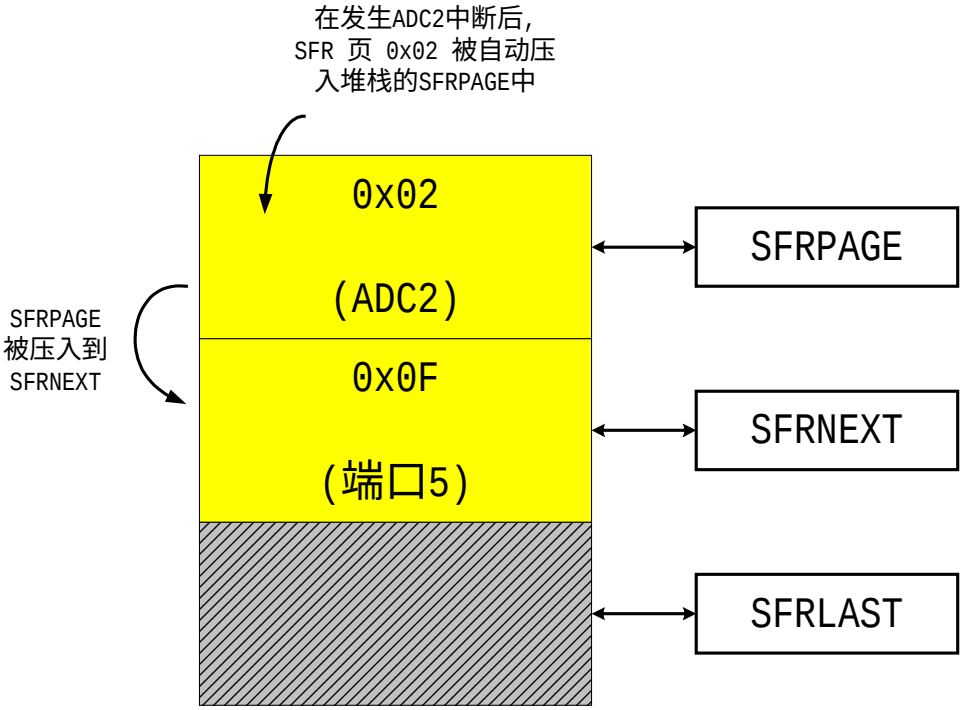


图 11.7　ADC2 窗口比较器中断发生后的 SFR 页堆栈

当 CIP-51 执行 ADC2 ISR 时，发生了 PCA 中断。回顾一下，PCA 中断被配置为高优先级，ADC2 中断被配置为低优先级。因此 CIP-51 现在转去执行高优先级的 PCA ISR。进入 PCA ISR 之时，CIP-51 自动将访问 PCA 的特殊功能寄存器所需要的 SFR 页（SFR 页 0x00）装入 SFRPAGE 寄存器。在 PCA 中断之前保存在 SFRPAGE 寄存器中的值（ADC2 所在的 SFR 页 0x02）被压入到堆栈的 SFRNEXT 中。同样，在 PCA 中断之前保存在 SFRNEXT 寄存器中的值（端口 5 所在的 SFR 页 0x05）被压入到 SFRLAST 寄存器中，即堆栈的底部。注意，先前保存在 SFRLAST 中的值（前一次软件写 SFRLAST 的操作）被覆盖。见下面的图 11.8。

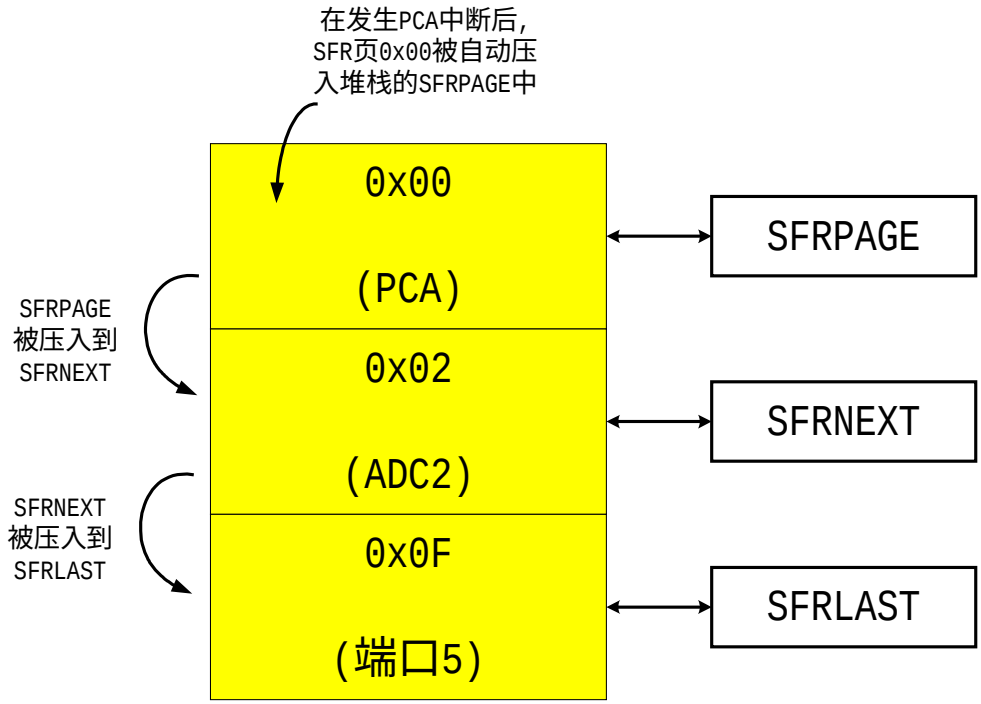


图 11.8　在 ADC2 ISR 期间发生 PCA 中断后的 SFR 页堆栈

从 PCA 中断服务程序返回时，CIP-51 将返回到 ADC2 窗口比较器 ISR。在执行 RETI 指令时，用于访问 PCA 寄存器的 SFR 页 0x00 被自动弹出 SFR 页堆栈，SFRNEXT 寄存器的内容被保存到 SFRPAGE 寄存器中。ADC2 ISR 中的软件可以像 PCA 中断发生之前那样继续访问 ADC2 的 SFR。同样，SFRLAST 寄存器中的值被保存到 SFRNEXT 寄存器中。这是在 ADC2 中断发生之前用于访问端口 5 的 SFR 页值 0x0F。见下面的图 11.9。

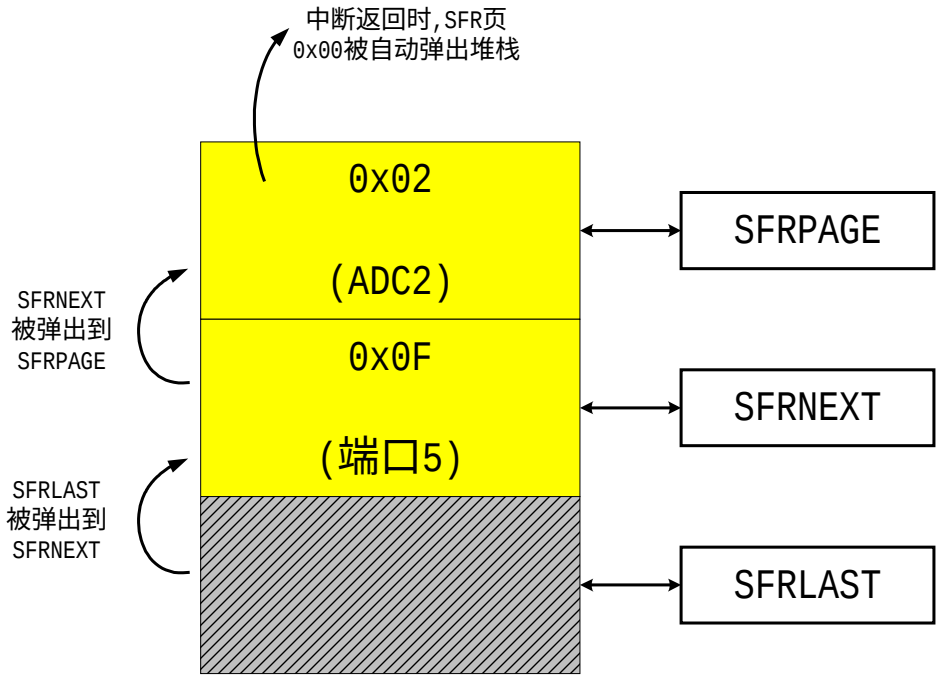


图 11.9 从 PCA 中断返回后的 SFR 页堆栈

在执行 ADC2 窗口比较器 ISR 中的 RETI 指令时，SFRPAGE 寄存器中的值被 SFRNEXT 的内容覆盖。CIP-51 现在可以像在中断发生之前那样访问端口 5 SFR 中的数据位。见下面的图 11.10。

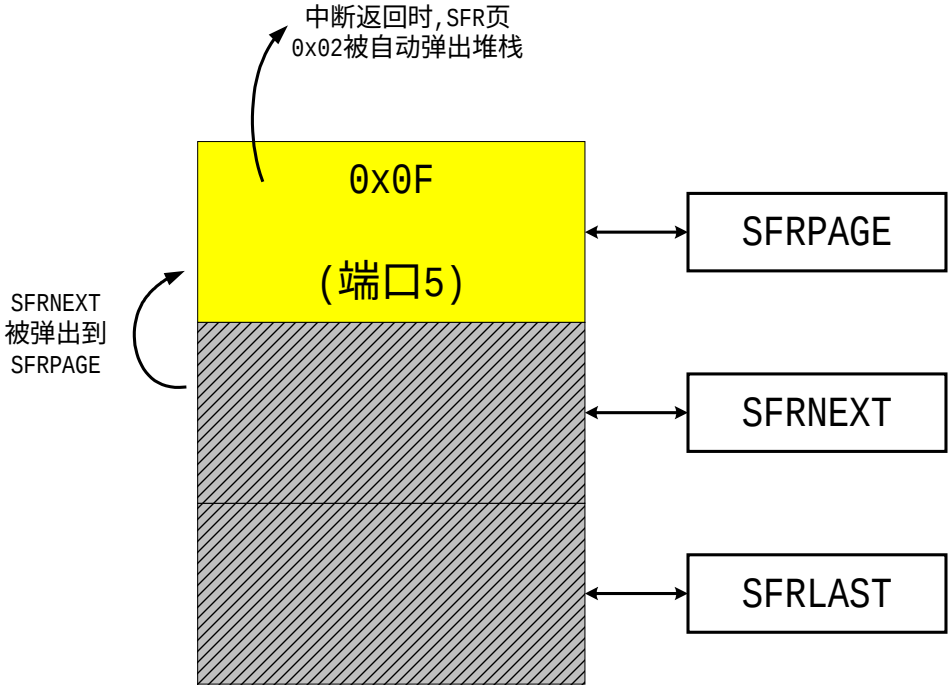


图 11.10 从 ADC2 窗口中断返回后的 SFR 页堆栈

注意 在上述例子中，SFR 页堆栈的全部 3 个字节都可以通过 SFRPAGE、SFRNEXT 和 SFRLAST 这几个特殊功能寄存器访问。如果在中断服务期间该堆栈被修改，中断返回后的当前 SFR 页可能与中断调用发生之前所选择的 SFR 页不同。在实时操作系统控制和管理多个任务间的上下文切换时，直接访问 SFR 页堆栈是很有用的。

SFR 页堆栈的压栈操作只能发生在中断服务之时，出栈操作只能发生在中断返回之际（执行 RETI 指令时）。上述的 SFRPAGE 自动切换和 SFR 页堆栈操作可以用软件禁止，这可以通过清除 SFR 页控制寄存器（SFRPGCN）中的 SFR 自动页使能位（SFRPGEN）来实现。见图 11.11。

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-			-	-	-	SFRPGCN	00000001
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x96
SFR 页： F

位 7-1： 保留

位 0： SFRPGEN：SFR 自动页控制使能位。
发生中断时，C8051 核将转向特定的中断服务程序，并自动将 SFR 页切换到相应外设或功能的 SFR 页。该位用于控制这种自动切换功能。
0：自动页控制功能被禁止。C8051 核不会自动切换到相应的 SFR 页（即包含中断源所在外设/功能的 SFR 之 SFR 页）。
1：自动页控制功能被使能。中断发生时，C8051 将 SFR 页切换到包含中断源所在外设/功能的 SFR 之 SFR 页）。

图 11.11 SFRPGCN: SFR 页控制寄存器

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

SFR 地址：0x84
SFR 页： 所有页

位 7-0： SFR 页：代表 C8051 MCU 读或修改 SFR 时所使用的 SFR 页。
写：设置 SFR 页。
读：C8051 MCU 正在使用的 SFR 页。
当 SFR 页控制寄存器（SFRPGCN）中的 SFRPGEN 被使能时，C8051 自动切换到包含引起中断的外设/功能的 SFR 之 SFR 页，并在中断返回时返回到中断前的 SFR 页（除非在中断返回前 SFR 栈被修改）。

SFRPAGE 是 SFR 页堆栈的顶部字节，对该堆栈的压栈/出栈操作是由中断引起的（而不是由读/写 SFRPAGE 寄存器引起的）。

图 11.12 SFRPAGE: SFR 页寄存器

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：0x85
SFR 页： 所有页

位 7-0： SFR 页堆栈位：在中断进入/返回时，SFR 页上下文被保存在一个 3 字节的 SFR 堆栈中：SFRPAGE 是第一个字节，SFRNEXT 是第二个字节，SFRLAST 是第三个字节。SFR 页堆栈字节可以用于改写 SFR 页堆栈的上下文，但不能引起 SFR 页堆栈的压栈和出栈操作。只有中断发生和中断返回才能导致 SFR 页堆栈的压栈和出栈操作。

写：设置 SFR 页堆栈的第二字节所包含的 SFR 页。这将导致在中断返回时 SFRPAGE 寄存器将含有该 SFR 页值。

读：返回 SFR 页堆栈的第二字节所包含的 SFR 值。这是中断返回时进入 SFRPAGE 寄存器的值。

图 11.13 SFRNEXT: SFR 后续寄存器

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：0x86
SFR 页： 所有页

位 7-0： SFR 页堆栈位：在中断进入/返回时，SFR 页上下文被保存在一个 3 字节的 SFR 堆栈中：SFRPAGE 是第一个字节，SFRNEXT 是第二个字节，SFRLAST 是第三个字节。SFR 页堆栈字节可以用于改写 SFR 页堆栈的上下文，但不能引起 SFR 页堆栈的压栈和出栈操作。只有中断发生和中断返回才能导致 SFR 页堆栈的压栈和出栈操作。

写：设置 SFR 页堆栈最后字节所包含的 SFR 页。这将导致在中断返回时 SFRNEXT 寄存器将含有该 SFR 页值。

读：返回 SFR 页堆栈的最后一个字节所包含的 SFR 值。

图 11.14 SFRLAST: SFR 栈底寄存器

表 11.2 特殊功能寄存器（SFR）存储器映象

地址	SFR 页	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8	0 1 2 3 F	SPI0CN P7	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	PCA0CPL1	PCA0CPH1	**WDTCN (所有页)**
F0	0 1 2 3 F	**B (所有页)**						**EIP1 (所有页)**	**EIP2 (所有页)**
E8	0 1 2 3 F	ADC0CN ADC2CN P6	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	PCA0CPL4	PCA0CPH4	RSTSRC
E0	0 1 2 3 F	**ACC (所有页)**	PCA0CPL5 XBR0	PCA0CPH5 XBR1	 XBR2			**EIE1 (所有页)**	**EIE2 (所有页)**
D8	0 1 2 3 F	PCA0CN P5	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	PCA0CPM5
D0	0 1 2 3 F	**PSW (所有页)**	REF0CN	DAC0L DAC1L	DAC0H DAC1H	DAC0CN DAC1CN			
C8	0 1 2 3 F	TMR2CN TMR3CN TMR4CN P4	TMR2CF TMR3CF TMR4CF	RCAP2L RCAP3L RCAP4L	RCAP2H RCAP3H RCAP4H	TMR2L TMR3L TMR4L	TMR2H TMR3H TMR4H	 MAC0RNDL	SMB0CR MAC0RNDH
C0	0 1 2 3 F	SMB0CN MAC0STA	SMB0STA MAC0AL	SMB0DAT MAC0AH	SMB0ADR MAC0CF	ADC0GTL ADC2GT	ADC0GTH	ADC0LTL ADC2LT	ADC0LTH
B8	0 1 2 3 F	**IP (所有页)**	SADEN0	AMX0CF AMX2CF	AMX0SL AMX2SL	ADC0CF ADC2CF		ADC0L ADC2	ADC0H
		0(8) 可位寻址	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

表 11.2 特殊功能寄存器（SFR）存储器映象（续）

地址	SFR 页	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
B0	0 1 2 3 F	**P3** **(所有页)**	**PSBANK** **(所有页)**						FLSCL FLACL
A8	0 1 2 3 F	**IE** **(所有页)**	SADDR0				P1MDIN		
A0	0 1 2 3 F	**P2** **(所有页)**	EMI0TC CCH0CN	EMI0CN CCH0TN	EMI0CF CCH0LC	P0MDOUT	P1MDOUT	P2MDOUT	P3MDOUT
98	0 1 2 3 F	SCON0 SCON1	SBUF0 SBUF1	SPI0CFG CCH0MA	SPI0DAT	P4MDOUT	SPI0CKR P5MDOUT	P6MDOUT	P7MDOUT
90	0 1 2 3 F	**P1** **(所有页)**	SSTA0 MAC0BL	MAC0BH	MAC0ACC0	MAC0ACC1	MAC0ACC2	MAC0ACC3 SFRPGCN	MAC0OVR CLKSEL
88	0 1 2 3 F	TCON CPT0CN CPT1CN FLSTAT	TMOD CPT0MD CPT1MD PLL0CN	TL0 OSCICN	TL1 OSCICL	TH0 OSCXCN	TH1 PLL0DIV	CKCON PLL0MUL	PSCTL PLL0FLT
80	0 1 2 3 F	**P0** **(所有页)**	**SP** **(所有页)**	**DPL** **(所有页)**	**DPH** **(所有页)**	**SFRPAGE** **(所有页)**	**SFRNEXT** **(所有页)**	**SFRLAST** **(所有页)**	**PCON** **(所有页)**
		0(8) 可位寻址	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

表 11.3 特殊功能寄存器

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

寄存器	地址	SFR 页	说　明	页码
ACC	0xE0	所有页	累加器	
ADC0CF	0xBC	0	ADC0 配置寄存器	
ADC0CN	0xE8	0	ADC0 控制寄存器	
ADC0GTH	0xC5	0	ADC0 下限（大于）数据字（高字节）	
ADC0GTL	0xC4	0	ADC0 下限（大于）数据字（低字节）	
ADC0H	0xBF	0	ADC0 数据字（高字节）	
ADC0L	0xBE	0	ADC0 数据字（低字节）	
ADC0LTH	0xC7	0	ADC0 上限（小于）数据字（高字节）	
ADC0LTL	0xC6	0	ADC0 上限（小于）数据字（低字节）	
ADC2	0xBE	2	ADC2 数据字	
ADC2CF	0xBC	2	ADC2 配置寄存器	
ADC2CN	0xE8	2	ADC2 控制寄存器	
ADC2GT	0xC4	2	ADC2 下限（大于）数据字	
ADC2LT	0xC6	2	ADC2 上限（小于）数据字	
AMX0CF	0xBA	0	ADC0 MUX 配置寄存器	
AMX0SL	0xBB	0	ADC0 MUX 通道选择寄存器	
AMX2CF	0xBA	2	ADC2 MUX 配置寄存器	
AMX2SL	0xBB	2	ADC2 MUX 通道选择寄存器	
B	0xF0	所有页	B 寄存器	
CCH0CN	0xA1	F	高速缓存（Cache）控制寄存器	
CCH0LC	0xA3	F	高速缓存（Cache）锁定寄存器	
CCH0MA	0x9A	F	高速缓存（Cache）遗漏（未命中）寄存器	
CCH0TN	0xA2	F	高速缓存（Cache）调节寄存器	
CKCON	0x8E	0	时钟控制寄存器	
CLKSEL	0x97	F	系统时钟选择寄存器	
CPT0CN	0x88	1	比较器 0 控制寄存器	
CPT0MD	0x89	1	比较器 0 配置寄存器	
CPT1CN	0x88	2	比较器 1 控制寄存器	
CPT1MD	0x89	2	比较器 1 配置寄存器	
DAC0CN	0xD4	0	DAC0 控制寄存器	
DAC0H	0xD3	0	DAC0 数据字高字节	
DAC0L	0xD2	0	DAC0 数据字低字节	
DAC1CN	0xD4	1	DAC1 控制寄存器	
DAC1H	0xD3	1	DAC1 数据字高字节	
DAC1L	0xD2	1	DAC1 数据字低字节	
DPH	0x83	所有页	数据指针（高字节）	
DPL	0x82	所有页	数据指针（低字节）	

表 11.3 特殊功能寄存器（续）

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

寄存器	地址	SFR 页	说 明	页码
EIE1	0xE6	所有页	扩展中断允许 1	
EIE2	0xE7	所有页	扩展中断允许 2	
EIP1	0xF6	所有页	扩展中断优先级 1	
EIP2	0xF7	所有页	扩展中断优先级 2	
EMI0CF	0xA3	0	外部存储器接口配置寄存器	
EMI0CN	0xA2	0	外部存储器接口控制寄存器	
EMI0TC	0xA1	0	外部存储器接口时序控制寄存器	
FLACL	0xB7	F	FLASH 访问极限寄存器	
FLSCL	0xB7	0	FLASH 存储器定时预分频器	
FLSTAT	0x88	F	FLASH 状态寄存器	
IE	0xA8	所有页	中断允许寄存器	
IP	0xB8	所有页	中断优先级控制寄存器	
MAC0ACC0	0x93	3	MAC0 累加器字节 0（LSB）	
MAC0ACC1	0x94	3	MAC0 累加器字节 1	
MAC0ACC2	0x95	3	MAC0 累加器字节 2	
MAC0ACC3	0x96	3	MAC0 累加器字节 3（MSB）	
MAC0AH	0xC2	3	MAC0 A 寄存器高字节	
MAC0AL	0xC1	3	MAC0 A 寄存器低字节	
MAC0BH	0x92	3	MAC0 B 寄存器高字节	
MAC0BL	0x91	3	MAC0 B 寄存器低字节	
MAC0CF	0xC3	3	MAC0 配置寄存器	
MAC0OVR	0x97	3	MAC0 累加器溢出寄存器	
MAC0RNDH	0xCF	3	MAC0 舍入寄存器高字节	
MAC0RNDL	0xCE	3	MAC0 舍入寄存器低字节	
MAC0STA	0xC0	3	MAC0 状态寄存器	
OSCICL	0x8B	F	内部振荡器校准寄存器	
OSCICN	0x8A	F	内部振荡器控制寄存器	
OSCXCN	0x8C	F	外部振荡器控制寄存器	
P0	0x80	所有页	端口 0 锁存器	
P0MDOUT	0xA4	F	端口 0 输出方式配置寄存器	
P1	0x90	所有页	端口 1 锁存器	
P1MDIN	0xAD	F	端口 1 输入方式寄存器	
P1MDOUT	0xA5	F	端口 1 输出方式配置寄存器	
P2	0xA0	所有页	端口 2 锁存器	
P2MDOUT	0xA6	F	端口 2 输出方式配置寄存器	
P3	0xB0	所有页	端口 3 锁存器	
P3MDOUT	0xA7	F	端口 3 输出方式配置寄存器	

表 11.3 特殊功能寄存器（续）

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

寄存器	地址	SFR 页	说 明	页码
P4	0xC8	F	端口 4 锁存器	
P4MDOUT	0x9C	F	端口 4 输出方式配置寄存器	
P5	0xD8	F	端口 5 锁存器	
P5MDOUT	0x9D	F	端口 5 输出方式配置寄存器	
P6	0xE8	F	端口 6 锁存器	
P6MDOUT	0x9E	F	端口 6 输出方式配置寄存器	
P7	0xF8	F	端口 7 锁存器	
P7MDOUT	0x9F	F	端口 7 输出方式配置寄存器	
PCA0CN	0xD8	0	PCA 控制寄存器	
PCA0CPH0	0xFC	0	PCA 模块 0 捕捉/比较高字节	
PCA0CPH1	0xFE	0	PCA 模块 1 捕捉/比较高字节	
PCA0CPH2	0xEA	0	PCA 模块 2 捕捉/比较高字节	
PCA0CPH3	0xEC	0	PCA 模块 3 捕捉/比较高字节	
PCA0CPH4	0xEE	0	PCA 模块 4 捕捉/比较高字节	
PCA0CPH5	0xE2	0	PCA 模块 5 捕捉/比较高字节	
PCA0CPL0	0xFB	0	PCA 模块 0 捕捉/比较低字节	
PCA0CPL1	0xFD	0	PCA 模块 1 捕捉/比较低字节	
PCA0CPL2	0xE9	0	PCA 模块 2 捕捉/比较低字节	
PCA0CPL3	0xEB	0	PCA 模块 3 捕捉/比较低字节	
PCA0CPL4	0xED	0	PCA 模块 4 捕捉/比较低字节	
PCA0CPL5	0xE1	0	PCA 模块 5 捕捉/比较低字节	
PCA0CPM0	0xDA	0	PCA 模块 0 方式寄存器	
PCA0CPM1	0xDB	0	PCA 模块 1 方式寄存器	
PCA0CPM2	0xDC	0	PCA 模块 2 方式寄存器	
PCA0CPM3	0xDD	0	PCA 模块 3 方式寄存器	
PCA0CPM4	0xDE	0	PCA 模块 4 方式寄存器	
PCA0CPM5	0xDF	0	PCA 模块 5 方式寄存器	
PCA0H	0xFA	0	PCA 计数器高字节	
PCA0L	0xF9	0	PCA 计数器低字节	
PCA0MD	0xD9	0	PCA 方式寄存器	
PCON	0x87	所有页	电源控制寄存器	
PLL0CN	0x89	F	PLL 控制寄存器	
PLL0DIV	0x8D	F	PLL 除数寄存器	
PLL0FLT	0x8F	F	PLL 滤波器寄存器	
PLL0MUL	0x8E	F	PLL 乘数寄存器	
PSBANK	0xB1	所有页	FLASH 块选择寄存器	
PSCTL	0x8F	0	FLASH 写/擦除控制寄存器	
PSW	0xD0	所有页	程序状态字	

表 11.3 特殊功能寄存器（续）

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

寄存器	地址	SFR 页	说 明	页码
RCAP2H	0xCB	0	定时器/计数器 2 捕捉（高字节）	
RCAP2L	0xCA	0	定时器/计数器 2 捕捉（低字节）	
RCAP3H	0xCB	1	定时器/计数器 3 捕捉（高字节）	
RCAP3L	0xCA	1	定时器/计数器 3 捕捉（低字节）	
RCAP4H	0xCB	2	定时器/计数器 4 捕捉（高字节）	
RCAP4L	0xCA	2	定时器/计数器 4 捕捉（低字节）	
REF0CN	0xD1	0	电压基准控制寄存器	
RSTSRC	0xEF	0	复位源寄存器	
SADDR0	0xA9	0	UART0 从地址寄存器	
SADEN0	0xB9	0	UART0 从地址允许寄存器	
SBUF0	0x99	0	UART0 数据缓冲器	
SBUF1	0x99	1	UART1 数据缓冲器	
SCON0	0x98	0	UART0 控制寄存器	
SCON1	0x98	1	UART1 控制寄存器	
SFRLAST	0x86	所有页	SFR 页堆栈最后字节	
SFRNEXT	0x85	所有页	SFR 页堆栈后续字节	
SFRPAGE	0x84	所有页	SFR 页选择	
SFRPGCN	0x96	F	SFR 页控制寄存器	
SMB0ADR	0xC3	0	SMBus 0 地址寄存器	
SMB0CN	0xC0	0	SMBus 0 控制寄存器	
SMB0CR	0xCF	0	SMBus 0 时钟频率寄存器	
SMB0DAT	0xC2	0	SMBus 0 数据寄存器	
SMB0STA	0xC1	0	SMBus 0 状态寄存器	
SP	0x81	所有页	堆栈指针	
SPI0CFG	0x9A	0	SPI 配置寄存器	
SPI0CKR	0x9D	0	SPI 时钟速率寄存器	
SPI0CN	0xF8	0	SPI 控制寄存器	
SPI0DAT	0x9B	0	SPI 数据寄存器	
SSTA0	0x91	0	UART0 状态寄存器	
TCON	0x88	0	定时器/计数器控制寄存器	
TH0	0x8C	0	定时器/计数器 0 高字节	
TH1	0x8D	0	定时器/计数器 1 高字节	
TL0	0x8A	0	定时器/计数器 0 低字节	
TL1	0x8B	0	定时器/计数器 1 低字节	
TMOD	0x89	0	定时器/计数器方式寄存器	
TMR2CF	0xC9	0	定时器 2 配置寄存器	
TMR2CN	0xC8	0	定时器 2 控制寄存器	
TMR2H	0xCD	0	定时器 2 高字节	
TMR2L	0xCC	0	定时器 2 低字节	

表 11.3 特殊功能寄存器（续）

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

寄存器	地址	SFR 页	说 明	页码
TMR3CF	0xC9	1	定时器 3 配置寄存器	
TMR3CN	0xC8	1	定时器 3 控制寄存器	
TMR3H	0xCD	1	定时器 3 高字节	
TMR3L	0xCC	1	定时器 3 低字节	
TMR4CF	0xC9	2	定时器 4 配置寄存器	
TMR4CN	0xC8	2	定时器 4 控制寄存器	
TMR4H	0xCD	2	定时器 4 高字节	
TMR4L	0xCC	2	定时器 4 低字节	
WDTCN	0xFF	所有页	看门狗定时器控制	
XBR0	0xE1	F	端口 I/O 交叉开关控制 0	
XBR1	0xE2	F	端口 I/O 交叉开关控制 1	
XBR2	0xE3	F	端口 I/O 交叉开关控制 2	

11.2.7 寄存器说明

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

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
SFR 页： 所有页

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

图 11.15 SP：堆栈指针

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
SFR 页： 所有页

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

图 11.16 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：0x83
SFR 页： 所有页

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

图 11.17 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
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
SFR 页： 所有页

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

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

位 5： F0：用户标志 0。
这是一个可位寻址、受软件控制的通用标志位。

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

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

位 2： OV：溢出标志。
该位在下列情况下被置 1：
- ADD、ADDC 或 SUBB 指令引起符号位变化溢出。
- MUL 指令引起溢出（结果大于 255）。
- DIV 指令的除数为 0。

ADD、ADDC、SUBB、MUL 和 DIV 指令的其它情况使该位清 0。

位 1： F1：用户标志 1。
这是一个可位寻址、受软件控制的通用标志位。

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

图 11.18 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
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
SFR 页： 所有页

位 7-0： ACC：累加器
该寄存器为算术操作用的累加器。

图 11.19 ACC：累加器

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
SFR 页： 所有页

位 7-0： B：B 寄存器
该寄存器为某些算术操作的第二累加器。

图 11.20 B：B 寄存器

11.3 中断系统

CIP-51 包含一个扩展的中断系统，支持 20 个中断源，每个中断源有两个优先级。中断源在片内外设与外部输入引脚之间的分配随器件的不同而变化。每个中断源可以在一个 SFR 中有一个或多个中断标志。当一个外设或外部源满足有效的中断条件时，相应的中断标志被置为逻辑‘ 1 ’。

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

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

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

11.3.1 MCU 中断源和中断向量

MCU 支持 20 个中断源。软件可以通过将任何一个中断标志设置为逻辑‘ 1 ’来模拟一个中断。如果中断标志被允许，系统将产生一个中断请求，CPU 将转向与该中断标志对应的 ISR 地址。表 11.4 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。

11.3.2 外部中断

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

表 11.4 中断一览表

中断源	中断向量	优先级	中断标志	位寻址？	硬件清除	使能位	优先级控制
复位	0x0000	最高	无	N/A	N/A	始终使能	总是最高
外部中断 0 (/INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
定时器 0 溢出	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
外部中断 1 (INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
定时器 1 溢出	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y		ES0 (IE.4)	PS0 (IP.4)
定时器 2	0x002B	5	TF2 (TMR2CN.7) EXF2 (TMR2CN.6)	Y		ET2 (IE.5)	PT2 (IP.5)
串行外设接口	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF(SPI0CN.5) RXOVRN(SPI0CN.4)	Y		ESPI0 (EIE1.0)	PSPI0 (EIP1.0)
SMBus 接口	0x003B	7	SI (SMB0CN.3)	Y		ESMB0(EIE1.1)	PSMB0 (EIP1.1)
ADC0 窗口比较	0x0043	8	AD0WINT (ADC0CN.1)	Y		EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
可编程计数器阵列	0x004B	9	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y		EPCA0 (EIE1.3)	PPCA0 (EIP1.3)
比较器 0 下降沿	0x0053	10	CP0FIF (CPT0CN.4)	Y		ECP0F (EIE1.4)	PCP0F (EIP1.4)
比较器 0 上升沿	0x005B	11	CP0RIF (CPT0CN.5)	Y		ECP0R (EIE1.5)	PCP0R (EIP1.5)
比较器 1 下降沿	0x0063	12	CP1FIF (CPT1CN.4)	Y		ECP1F (EIE1.6)	PCP1F (EIP1.6)
比较器 1 上升沿	0x006B	13	CP1RIF (CPT1CN.5)	Y		ECP1R (EIE1.7)	PCP1R (EIP1.7)
定时器 3	0x0073	14	TF3 (TMR3CN.7) EXF3 (TMR3CN.6)	Y		ET3 (EIE2.0)	PT3 (EIP2.0)
ADC0 转换结束	0x007B	15	AD0INT (ADC0CN.5)	Y		EADC0 (EIE2.1)	PADC0 (EIP2.1)
定时器 4	0x0083	16	TF4 (T4CON.7) EXF4 (TMR4CN.6)	Y		ET4 (EIE2.2)	PT4 (EIP2.2)
ADC2 窗口比较	0x008B	17	AD2WINT (ADC2CN.0)	Y		EWADC2 (EIE2.3)	PWADC2 (EIP2.3)
ADC2 转换结束	0x0093	18	AD2INT (ADC2CN.5)	Y		EADC2 (EIE2.4)	PADC2 (EIE2.4)
保留	0x009B	19	N/A	N/A	N/A	N/A	N/A
UART1	0x00A3	20	RI1 (SCON1.0) TI1 (SCON1.1)	Y		ES1(EIE2.6)	PS1(EIP2.6)

11.3.3 中断优先级

每个中断源都可以被独立地编程为两个优先级中的一个：低优先级或高优先级。一个低优先级的中断服务程序可以被高优先级的中断所中断，但高优先级的中断不能被中断。每个中断在 SFR（IP-EIP2）中都有一个配置其优先级的中断优先级设置位，缺省值为低优先级。如果两个中断同时发生，具有高优先级的中断先得到服务。如果这两个中断的优先级相同，则由固定的优先级顺序决定哪一个先得到服务（如表 11.4 所示）。

11.3.4 中断响应时间

中断响应时间取决于中断发生时 CPU 的状态。中断系统在每个系统时钟周期对中断标志采样并对优先级译码。最快的响应时间为 5 个系统时钟周期：一个周期用于检测中断，4 个周期完成对 ISR 的长调用（LCALL）。如果发生高速缓存遗漏（Cache miss），则还需几个附加周期。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 并且发生高速缓存遗漏的情况。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。

11.3.5 中断寄存器说明

下面介绍用于允许中断源和设置中断优先级的特殊功能寄存器。关于外设有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。

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

SFR 地址：0xA8
SFR 页： 所有页

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

位 6： IEGF0：通用标志位 0。
该位用作软件控制的通用标志位。

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

位 4： ES0：UART0 中断允许位。
该位设置 UART0 的中断屏蔽。
0：禁止 UART0 中断。
1：允许 UART0 中断。

位 3： ET1：定时器 1 中断允许位。
位6 该位用于设置定时器 1 的中断屏蔽。
0：禁止定时器 1 中断。
1：允许定时器 1 中断。

位 2： EX1：外部中断 1 允许位。
该位用于设置外部中断 1 的中断屏蔽。
0：禁止外部中断 1。
1：允许外部中断 1。

位 1： ET0：定时器 0 中断允许位。
该位用于设置定时器 0 的中断屏蔽。
0：禁止定时器 0 中断。
1：允许定时器 0 中断。

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：允许外部中断 0。

图 11.21 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
-	-	PT2	PS0	PT1	PX1	PT0	PX0	11000000
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：0xB8
SFR 页： 所有页

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 中断为低优先级。
1：定时器 2 中断为高优先级。
位 4： PS0： UART0 中断优先级控制。
该位设置 UART0 中断的优先级。
0：UART0 中断为低优先级。
1：UART0 中断为高优先级。
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 中断为低优先级。
1：定时器 1 中断为高优先级。
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 中断为低优先级。
1：外部中断 1 中断为高优先级。
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 中断为低优先级。
1：定时器 0 中断为高优先级。
位 0： PX0：外部中断 0 优先级控制
该位设置外部中断 0 的优先级。
0：外部中断 0 为低优先级。
1：外部中断 0 为高优先级。

图 11.22 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
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
SFR 页： 所有页

位 7： ECP1R：允许比较器 1（CP1）上升沿中断。
该位设置 CP1 的中断屏蔽。
0：禁止 CP1 上升沿中断。
1：允许 CP1 上升沿中断。

位 6： ECP1F：允许比较器 1（CP1）下降沿中断。
该位设置 CP1 的中断屏蔽。
0：禁止 CP1 下降沿中断。
1：允许 CP1 下降沿中断。

位 5： ECP0R：允许比较器 0（CP0）上升沿中断。
该位设置 CP0 的中断屏蔽。
0：禁止 CP0 上升沿中断。
1：允许 CP0 上升沿中断。

位 4： ECP0F：允许比较器 0（CP0）下降沿中断。
该位设置 CP0 的中断屏蔽。
0：禁止 CP0 下降沿中断。
1：禁止 CP0 下降沿中断。

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

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

位 1： ESMB0：允许 SMBus0 中断
该位设置 SMBus0 的中断屏蔽。
0：禁止 SMBus0 中断。
1：允许 SMBus0 中断。

位 0： ESPI0：允许串行外设接口 0（SPI0）中断
该位设置 SPI0 的中断屏蔽。
0：禁止 SPI0 中断。
1：允许 SPI0 中断。

图 11.23 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
-	ES1	-	EADC2	EWADC2	ET4	EADC0	ET3	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xE7
SFR 页： 所有页

位 7： 未用。读=0b，写=忽略。
位 6： ES1：允许 UART1 中断。
该位设置 UART1 的中断屏蔽。
0：禁止 UART1 中断。
1：允许 UART1 中断。
位 5： 未用。读=0b，写=忽略。
位 4： EADC2：允许 ADC2 转换结束中断。
该位设置 ADC2 转换结束的中断屏蔽。
0：禁止 ADC2 转换结束中断。
1：允许 ADC2 转换结束中断。
位 3： EWADC2：允许 ADC2 窗口比较中断。
该位设置 ADC2 窗口比较的中断屏蔽。
0：禁止 ADC2 窗口比较中断。
1：允许 ADC2 窗口比较中断。
位 2： ET4：允许定时器 4 中断。
该位设置定时器 4 的中断屏蔽。
0：禁止定时器 4 中断。
1：允许定时器 4 中断。
位 1： EADC0：允许 ADC0 转换结束中断。
该位设置 ADC0 转换结束的中断屏蔽。
0：禁止 ADC0 转换结束中断。
1：允许 ADC0 转换结束中断。
位 0： ET3：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：允许定时器 3 中断。

图 11.24 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
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：0xE7
SFR 页： 所有页

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 11.25 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
-	PS1	-	PADC2	PWADC2	PT4	PADC0	PT3	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xF7
SFR 页： 所有页

位 7： 未用。读=0b，写=忽略。
位 6： PS1：UART1 中断优先级控制
该位设置 UART1 中断的优先级。
0：UART1 中断为低优先级
1：UART1 中断为高优先级
位 5： 未用。读=0b，写=忽略。
位 4： PADC2：ADC2 转换结束中断优先级控制
该位设置 ADC2 转换结束中断的优先级。
0：ADC2 转换结束中断为低优先级
1：ADC2 转换结束中断为高优先级
位 3： PWADC2：ADC2 窗口比较中断优先级控制
该位设置 ADC2 窗口比较中断的优先级。
0：ADC2 窗口比较中断为低优先级
1：ADC2 窗口比较中断为高优先级
位 2： PT4：定时器 4 中断优先级控制
该位设置定时器 4 中断的优先级。
0：定时器 4 中断设置为低优先级
1：定时器 4 中断设置为高优先级
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

图 11.26 EIP2：扩展中断优先级 2

11.4 电源管理方式

CIP-51 有两种可软件编程的电源管理方式：空闲和停机。在空闲方式，CPU 停止运行，而外设和时钟处于活动状态。在停机方式，CPU 停止运行，所有的中断和定时器（时钟丢失检测器除外）都处于非活动状态，系统时钟停止。由于在空闲方式下时钟仍然运行，所以功耗与进入空闲方式之前的系统时钟频率和处于活动状态的外设数目有关。停机方式消耗最少的功率。图 11.27 对用于控制 CIP-51 电源管理方式的电源控制寄存器（PCON）作出了说明。

虽然 CIP-51 具有空闲和停机方式（与任何标准 8051 结构一样），但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，使其进入低功耗方式。象定时器、串行总线这样的数字外设在不使用时消耗很少的功率。关闭闪速存储器可以减小功耗，与进入空闲方式类似。关闭振荡器可以消耗更少的功率，但需要靠复位来重新启动 MCU。

11.4.1 空闲方式

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

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

如果被使能，WDT 将产生一个内部看门狗复位，从而结束空闲方式。这一功能可以保护系统不会因为对 PCON 寄存器的意外写入而导致永久性停机。如果不需要这种功能，可以在进入空闲方式之前禁止看门狗。这将进一步节省功耗，允许系统一直保持在空闲状态，等待一个外部激励唤醒系统。有关使用和配置 WDT 的详细信息，请参见第 14 章。

注：任何将 IDLE 位置 1 的指令后面应立即跟随一条具有 2 或多字节操作码的指令。例如：

```
// 用‘C’语言：
PCON |= 0x01    ;   //          将 IDLE 位置 1
PCON = PCON ;   //          … 跟随一条 3 字接操作码的指令

// 用汇编语言：
ORL   PCON, #01h        ;   将 IDLE 位置 1
MOV   PCON, PCON        ;   … 跟随一条 3 字接操作码的指令
```

如果在写 IDLE 位的指令之后是一条单字节指令，并且在执行将 IDLE 位置 1 的指令期间产生了一个中断，则将来发生中断时 CPU 可能不会被唤醒。

11.4.2 停机方式

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

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

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	-	-	STOP	IDLE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x87
SFR 页： 所有页

位 7-2： 保留。
位 1： STOP：停机方式选择。
向该位写‘ 1 ’将使 CIP-51 进入停机方式。该位读出值总是为 0。
1：CIP-51 被强制进入掉电方式（关闭振荡器）
位 0： IDLE：空闲方式选择。
向该位写‘ 1 ’将使 CIP-51 进入空闲方式。该位读出值总是为 0。
1：CIP-51 被强制进入空闲方式。（关闭供给 CPU 的时钟信号，但定时器、中断和所有外设保持活动状态。）

图 11.27 PCON：电源控制寄存器

12. 乘法和累加引擎（MAC0）

C8051F120/1/2/3和C8051F130/1/2/3器件包含一个乘法和累加引擎（MAC0），可以用于加速很多数学运算。MAC0包含一个16 x16的乘法器和一个40位的加法器，它可以在两个SYSCLK周期内完成整数或小数乘法-累加，或者对带符号的输入值执行乘法运算。一个舍入引擎可以在一个附加的（第三个）SYSCLK周期后提供提供舍入的16位小数结果。MAC0还包含一个1位算术移位器，可以在一个SYSCLK周期内对40位累加器中的内容进行左移或右移。图12.1给出了MAC0单元及其相关特殊功能寄存器的原理框图。

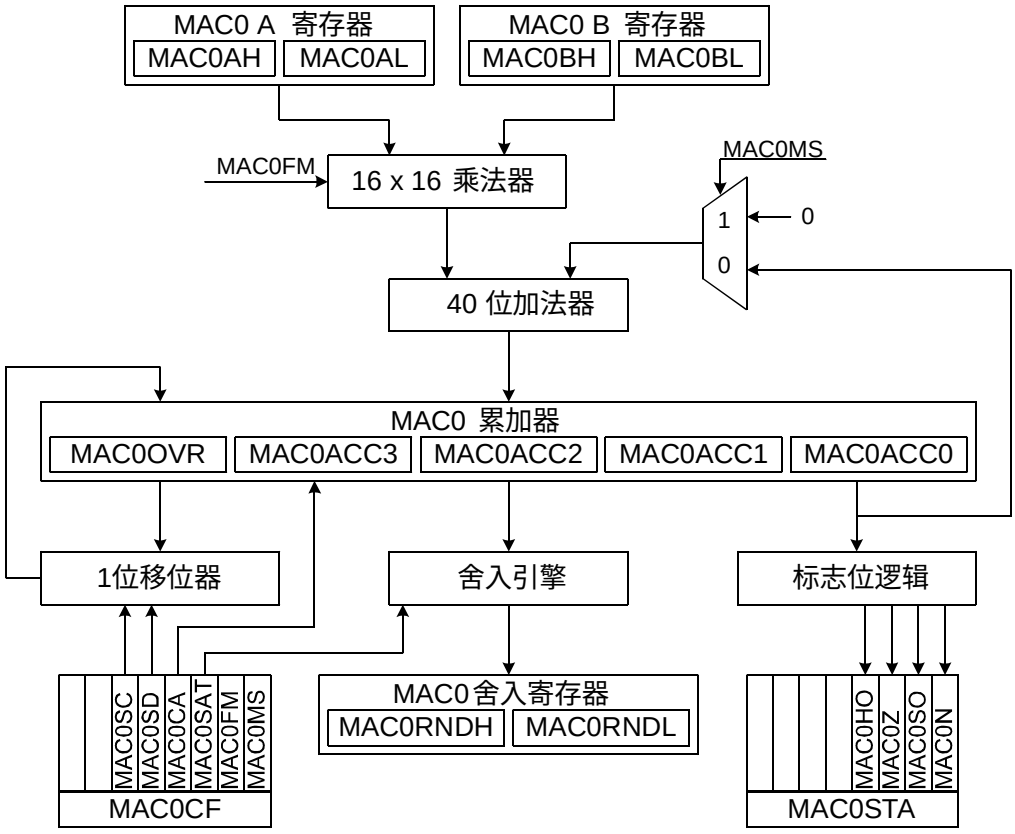


图12.1 MAC0原理框图

12.1 特殊功能寄存器

有13个与MAC0相关的特殊功能寄存器（SFR）。其中两个寄存器与配置和操作有关，其它11个寄存器用于保存MAC0的输入和输出数据。配置寄存器MAC0CF（见图12.8）用于配置和控制MAC0。状态寄存器MAC0STA（见图12.9）包含用于指示溢出条件以及零值、负值的标志位。16位的MAC0A（MAC0AH:MAC0AL）和MAC0B（MAC0BH:MAC0BL）寄存器为乘法器提供输入。MAC0累加器寄存器的长度为40位，包括5个SFR：MAC0OVR、MAC0ACC3、MAC0ACC2、MAC0ACC1和MAC0ACC0。MAC0操作的结果保存在MAC0累加器寄存器中。如果需要，舍入结果保存在舍入寄存器MAC0RND（MAC0RNDH:MAC0RNDL）中。

12.2 整数和小数运算

MAC0可以把MAC0A和MAC0B中的16位输入作为有符号整数或有符号小数处理。当MAC0FM位（MAC0CF.1）被清‘0’时，输入值被作为16位的整数（2的补码表示）。运算结束后，累加器中将包含40位长的整数（2的补码表示）。图12.2给出了整数在这些特殊功能寄存器中的表示格式。

MAC0A 和 MAC0B 位权

高字节								低字节							
$-(2^{15})$	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

MAC0 累加器位权

MAC0OVR					MAC0ACC3 : MAC0ACC2 : MAC0ACC1 : MAC0ACC0									
$-(2^{39})$	2^{38}		2^{33}	2^{32}	2^{31}	2^{30}	2^{29}	2^{28}		2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

图12.2 整型数据表示

当MAC0FM位（MAC0CF.1）被置‘1’时，输入值被作为16位的小数（2的补码表示）。小数点位于数据字的位15和位14之间。运算结束后，累加器中将包含40位长的整数（2的补码表示），小数点位于数据字的位31和位30之间。图12.3给出了小数在这些特殊功能寄存器中的表示格式。

MAC0A 和 MAC0B 位权

高字节								低字节							
-1	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}

MAC0 累加器位权

MAC0OVR					MAC0ACC3 : MAC0ACC2 : MAC0ACC1 : MAC0ACC0									
$-(2^{8})$	2^{7}		2^{2}	2^{1}	2^{0}	2^{-1}	2^{-2}	2^{-3}		2^{-27}	2^{-28}	2^{-29}	2^{-30}	2^{-31}

MAC0RND 位权

高字节								低字节							
* -2	1	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}

* MAC0RND 寄存器包含一个2的补码表示的数之16位LSB。 MAC0N 标志可用于确定 MAC0RND寄存器的符号。

图12.3 小数表示

12.3 乘法和累加工作方式

当MAC0MS位（MAC0CF.0）被清‘0’时，MAC0工作在乘法和累加（MAC）方式。当工作在MAC方式时，MAC0对MAC0A和MAC0B寄存器的内容执行16 x 16乘法运算，并将结果与40位MAC0累加器的内容相加。图12.4给出了MAC0流水线的示意图。

流水线分为三级，每一级正好需要一个SYSCLK周期。MAC0操作由对MAC0BL寄存器的写操作来启动。MAC0A和MAC0B的相乘操作在MAC0BL被写入后的第一个SYSCLK周期完成。在MAC0流水线的第二级，乘法结果与当前的累加器内容相加，加法结果保存在MAC0累加器中。MAC0STA寄存器中的标志位在流水线的第二级结束后被设置。如果需要，可以在流水线的第二级期间，通过写MAC0BL寄存器来启动下一次乘法操作。在流水线的第三级结束后，舍入（也可选则饱和）结果在MAC0RNDH和MAC0RNDL寄存器中可用。如果在MAC0操作启动时MAC0CA位（MAC0CF.3）被置‘1’，则在控制器时钟（SYSCLK）的下一周期MAC累加器和所有的MAC0STA标志位都将被清零。该清零操作结束后MAC0CA位也被清零。

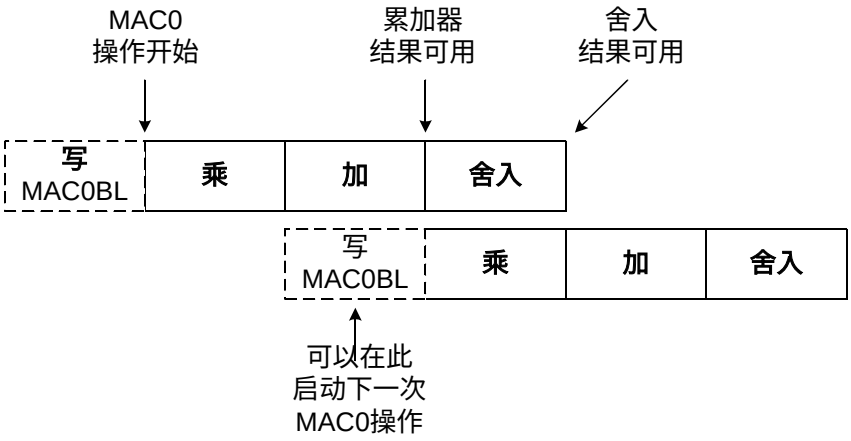


图12.4 MAC0流水线

12.4 乘法器工作方式

当MAC0MS位（MAC0CF.0）被置‘1’时，MAC0工作在乘法器方式。乘法器工作方式与乘法和累加工作方式的工作过程相同，只是乘法结果与0值相加，然后保存到MAC0累加器中（即当前累加器的内容被重写）。乘法结果在MAC0流水线的第二级结束后被保存在MAC0累加器中（MAC0BL被写入后两个SYSCLK周期）。与MAC方式一样，在流水线的第三级结束后，舍入结果在MAC0舍入寄存器中可用。

12.5 累加器移位操作

MAC0包含一个1位算术移位器，可用于对40位累加器中的内容执行左移或右移1位的操作。累加器移位操作通过向MAC0SC位（MAC0CF.5）写‘ 1 ’来启动，需一个SYSCLK周期（第二个SYSCLK周期之后，舍入结果在MAC0舍入寄存器中可用并且MAC0SC被清‘ 0 ’）。算术移位方向由MAC0SD位（MAC0CF.4）控制。当该位被清‘ 0 ’时，MAC0累加器将左移；当该位被置‘ 1 ’时，MAC0累加器将右移。右移操作是带符号扩展的（即位39的当前值）。注意，MAC0STA寄存器中的状态标志不受移位操作的影响。

12.6 舍入和饱和

MAC0包含一个舍入引擎，在进行小数运算时它可以提供舍入结果。MAC0使用无偏舍入算法对累加器中的位31~16进行舍入操作，如表12.1所示。舍入操作发生在下述情况：MAC0流水线的第三级、任何一次移位之后或写累加器的LSB时。舍入结果保存在舍入寄存器中：MAC0RNDH（见图12.19）和MAC0RNDL（见图12.20）。累加器寄存器不受舍入引擎的影响。尽管舍入操作主要针对小数运算，但当MAC0工作在整数方式时，舍入寄存器中的数据也以同样的方式被更新。

表12.1 MAC0舍入（MAC0SAT = 0）

累加器位15 ~ 0 (MAC0ACC1:MAC0ACC0)	累加器位31 ~ 16 (MAC0ACC3:MAC0ACC2)	舍入 方向	累加器位31 ~ 16 (MAC0RNDH:MAC0RNDL)
大于0x8000	任意值	向上	(MAC0ACC3:MAC0ACC2)+1
小于0x8000	任意值	向下	(MAC0ACC3:MAC0ACC2)
等于0x8000	奇数（LSB = 1）	向上	(MAC0ACC3:MAC0ACC2)+1
等于0x8000	偶数（LSB = 0）	向下	(MAC0ACC3:MAC0ACC2)

舍入引擎可用于对保存在舍入寄存器中的数据执行饱和操作。如果MAC0SAT位被置‘ 1 ’并且舍入寄存器溢出，则舍入寄存器将发生饱和。当发生正溢时，舍入寄存器的饱和值为0x7FFF；当发生负溢时，舍入寄存器的饱和值为0x8000。如果MAC0SAT位被清‘ 0 ’，则舍入寄存器不饱和。

12.7 用法举例

本节详细介绍几个使用MAC0的软件例子。图12.5给出了MAC0执行两个连续小数运算的例子，图12.6给出了MAC0工作在乘法器方式并执行一次运算的例子，图12.7的例子说明如何通过左移和右移操作修改累加器的内容。这些例子都假设MAC0STA寄存器中的所有标志均被初始化为‘0’值。

下面的例子实现方程：(0.5×0.25) + (0.5×-0.25) = 0.125 – 0.125 = 0

```
MOV  MAC0CF, #0Ah   ; 累加器清零，使用小数运算
MOV  MAC0AH, #40h   ; 向MAC0A寄存器装载 4000 hex = 0.5（十进制）
MOV  MAC0AL, #00h
MOV  MAC0BH, #20h   ; 向MAC0B寄存器装载 2000 hex = 0.25（十进制）
MOV  MAC0BL, #00h   ; 本行启动第一次MAC操作
MOV  MAC0BH, #E0h   ; 向MAC0B寄存器装载 E000 hex = -0.25（十进制）
MOV  MAC0BL, #00h   ; 本行启动第二次MAC操作
NOP
NOP                 ; 执行完本指令后，累加器中的值应为 0，
                    ; MAC0STA 寄存器应为 0x04，表示结果为0。
NOP                 ; 执行完本指令后，舍入寄存器被更新。
```

图12.5 乘法和累加示例

下面的例子实现方程：4660×-292 = – 1360720

```
MOV  MAC0CF, #01h   ; 使用整数，工作在乘法器方式（与0相加）
MOV  MAC0AH, #12h   ; 向MAC0A寄存器装载 1234 hex = 4660（十进制）
MOV  MAC0AL, #34h
MOV  MAC0BH, #FEh   ; 向MAC0B寄存器装载 FEDC hex = -292（十进制）
MOV  MAC0BL, #DCh   ; 本行启动乘法操作
NOP
NOP                 ; 执行完本指令后，累加器中的值应等于
                    ; FFFFEB3CB0 hex = -1360720（十进制）。
                    ; MAC0STA 寄存器应为 0x01， 表示结果为负数。
NOP                 ; 执行完本指令后，舍入寄存器被更新。
```

图12.6 乘法器方式示例

下面的例子将累加器左移一位，然后右移两位：

```
MOV  MAC0OVR, #40h   ; 下面几条指令向累加器装入值 4088442211 Hex。
MOV  MAC0ACC3, #88h
MOV  MAC0ACC2, #44h
MOV  MAC0ACC1, #22h
MOV  MAC0ACC0, #11h
MOV  MAC0CF, #20h   ; 启动左移操作
NOP                 ; 执行完本指令后，累加器中的值应为 0x8110884422
NOP                 ; 执行完本指令后，舍入寄存器被更新
MOV MAC0CF, #30h   ; 启动右移操作
MOV MAC0CF, #30h   ; 启动第二次右移操作
NOP                 ; 执行完本指令后，累加器中的值应为 0xE044221108
NOP                 ; 执行完本指令后，舍入寄存器被更新
```

图12.7　MAC0累加器移位示例

R	R	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	MAC0SC	MAC0SD	MAC0CA	MAC0SAT	MAC0FM	MAC0MS	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xC3
SFR 页： 3

位 7-6： 未用：读 = 00b，写 = 忽略。
位 5： MAC0SC：累加器移位控制位
当被置‘ 1 ’时，40 位的 MAC0 累加器寄存器将在下一个 SYSCLK 周期被移位。
移位方向（左移或右移）由 MAC0SD 位控制。
该位在移位操作结束后被硬件清‘ 0 ’。
位 4： MAC0SD：累加器移位方向控制位
该位控制累加器移位的方向，移位操作由 MAC0SC 位启动。
0：MAC0 累加器将被左移。
1：MAC0 累加器将被右移。
位 3： MAC0CA：累加器清除位
该位用于在下一次操作前复位 MAC0。
当被置‘ 1 ’时， 在下一个 SYSCLK 周期，MAC0 累加器寄存器将被清零，MAC0 状态寄存器将被复位。
该位在 MAC0 复位操作结束后被硬件清‘ 0 ’。
位 2： MAC0SAT：饱和/舍入控制位
该位控制舍入寄存器是否执行饱和操作。
如果该位被置位并且发生了软件溢出，则舍入寄存器将执行饱和操作。该位不影响 MAC0 累加器的操作。详见 12.6 节。
0：舍入寄存器不执行饱和操作。
1：舍入寄存器执行饱和操作。
位 1： MAC0FM：小数方式选择位
该位选择 MAC0 工作方式为整数方式还是小数方式。
0：MAC0 以整数方式操作。
0：MAC0 以小数方式操作。
位 0： MAC0MS：工作方式选择位
该位在 MAC 方式和乘法器方式之间选择。
0：MAC（乘法和累加）方式。
0：乘法器方式。

注：不应在 MAC0 流水线的前两级用软件改变该寄存器的内容。

图 12.8 MAC0CF: MAC0 配置寄存器

R	R	R	R	R/W	R/W	R/W	R/W	复位值
-	-	-	-	MAC0HO	MAC0Z	MAC0SO	MAC0N	00000100
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：0xC0
SFR 页： 3

位 7-4： 未用：读 = 0000b，写 = 忽略。

位 3： MAC0HO：硬件溢出标志。
在 MAC 操作期间，如果发生 MAC0OVR 寄存器向外溢出，则该位被置‘ 1 ’(即当 MAC0OVR 从 0x7F 变化到 0x80 或从 0x80 变化到 0x7F 时)。
硬件溢出标志必须被软件清零（直接向该位写 0，或使用寄存器 MAC0CF 中的 MAC0CA 位使 MAC 逻辑复位。

位 2： MAC0Z：零标志
如果 MAC 操作结果是累加器中的值为 0，则该位被置‘ 1 ’。
结果不为 0，则该位被清‘ 0 ’。

位 1： MAC0SO：软件溢出标志
当 MAC 操作发生向 MAC0 累加器的符号位（位 31）溢出时，该位被置‘ 1 ’。如果在一次后续的 MAC 操作后，溢出条件被修正，则该位被清‘ 0 ’。

位 0： MAC0N：负数标志
如果 MAC 累加器的结果为负数，则该位被置‘ 1 ’。如果结果为正数或零，则该位被清‘ 0 ’。

注：不应在 MAC0 流水线的前两级用软件改变该寄存器的内容。

图 12.9 MAC0STA: MAC0 状态寄存器

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
SFR 页： 3

位 7-0： MAC0 A 寄存器的高字节（位 15 ~ 8)。

图 12.10 MAC0AH：MAC0 A 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：0xC1
SFR 页： 3

位 7-0： MAC0A 寄存器的低字节（位 7 ~ 0）。

图 12.11 MAC0AL：MAC0 A 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：0x92
SFR 页： 3

位 7-0： MAC0 B 寄存器的高字节（位 15 ~ 8）。

图 12.12 MAC0BH：MAC0 B 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：0x91
SFR 页： 3

位 7-0： MAC0B 寄存器的低字节（位 7 ~ 0）。
对该寄存器的写操作启动一次乘法或乘法和累加操作。

注：不应在 MAC0 流水线的第一级用软件改变该寄存器的内容。

图 12.13 MAC0BL：MAC0 B 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：0x96
SFR 页： 3

位 7-0： MAC0 累加器的字节 3（位 31 ~ 24）。

注：不应在 MAC0 流水线的前两级用软件改变该寄存器的内容。

图 12.14 MAC0ACC3：MAC0 累加器字节 3 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：0x95
SFR 页： 3

位 7-0： MAC0 累加器的字节 2（位 23 ~ 16）。

注：不应在 MAC0 流水线的前两级用软件改变该寄存器的内容。

图 12.15 MAC0ACC2：MAC0 累加器字节 2 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：0x94
SFR 页： 3

位 7-0： MAC0 累加器的字节 1（位 15 ~ 8）。

注：不应在 MAC0 流水线的前两级用软件改变该寄存器的内容。

图 12.16 MAC0ACC1：MAC0 累加器字节 1 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：0x93
SFR 页： 3

位 7-0： MAC0 累加器的字节 0（位 7 ~ 0）。

注：不应在 MAC0 流水线的前两级用软件改变该寄存器的内容。

图 12.17 MAC0ACC0：MAC0 累加器字节 0 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：0x97
SFR 页： 3

位 7-0： MAC0 累加器溢出位（位 39 ~ 32）。

注：不应在 MAC0 流水线的前两级用软件改变该寄存器的内容。

图 12.18 MAC0OVR：MAC0 累加器溢出寄存器

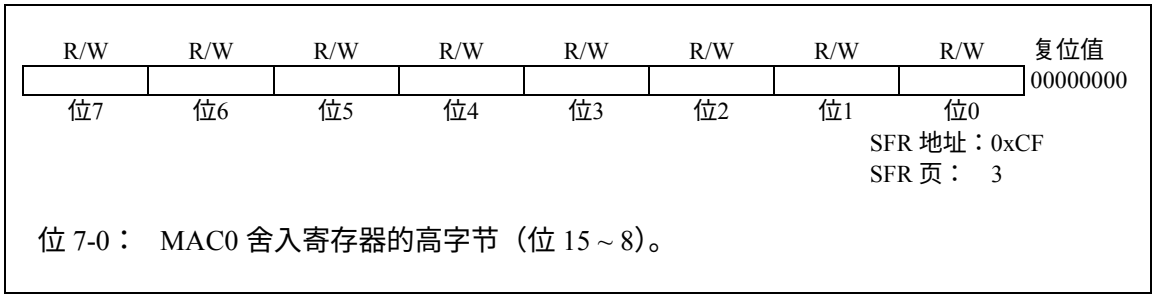


图 12.19 MAC0RNDH：MAC0 舍入寄存器高字节

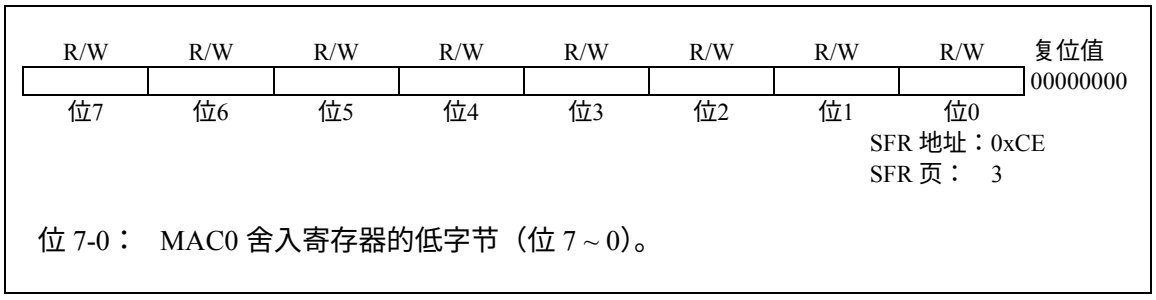


图 12.20 MAC0RNDL：MAC0 舍入寄存器低字节

13. 复位源

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

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

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

I/O 端口锁存器的复位值为 0xFF（全部为逻辑‘1’），复位期间和复位后内部弱上拉有效。对于 VDD 监视器复位，/RST 引脚被驱动为低电平，直到 VDD 复位超时结束。

在退出复位状态时，程序计数器（PC）被复位，MCU 使用内部振荡器的最低频率作为默认的系统时钟。有关选择和配置系统时钟源的详细说明见“14 振荡器”。看门狗定时器被使能，使用其最长的超时时间（见“13.7 看门狗定时器复位”）。一旦系统时钟源稳定，程序从地址 0x0000 开始执行。

有 7 个能使 MCU 进入复位状态的复位源：上电、掉电、外部/RST 引脚、外部 CNVSTR0 信号、软件命令、比较器 0、时钟丢失检测器及看门狗定时器。下面将对每个复位源进行说明。

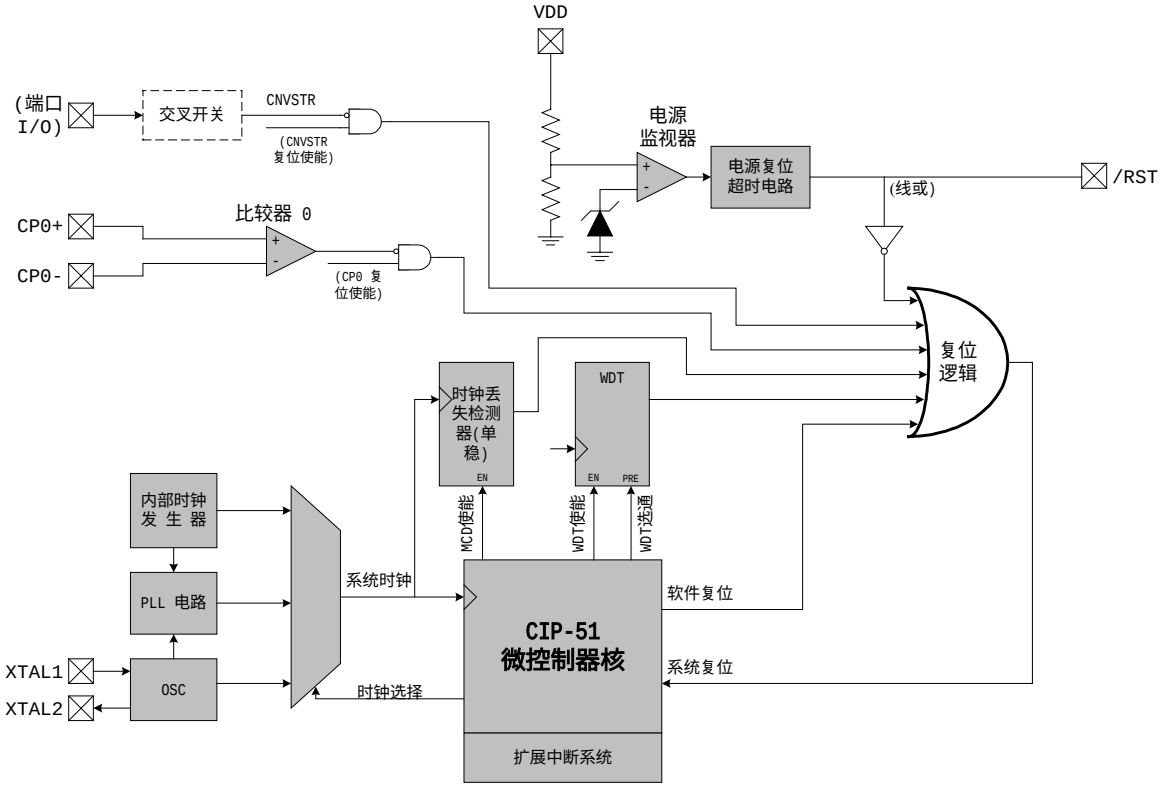


图 13.1 复位源框图

13.1 上电复位

C8051F12x 和 C8051F13x 器件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。使用外部 VDD 监视器使能引脚（MONEN）来使能和禁止 VDD 监视器复位。当 VDD 监视器被使能时，可以用 PORSF 位将其选择为复位源。要使 VDD 监视器有效，必须向 PORSF（RSTSRC.1）写‘1’。

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

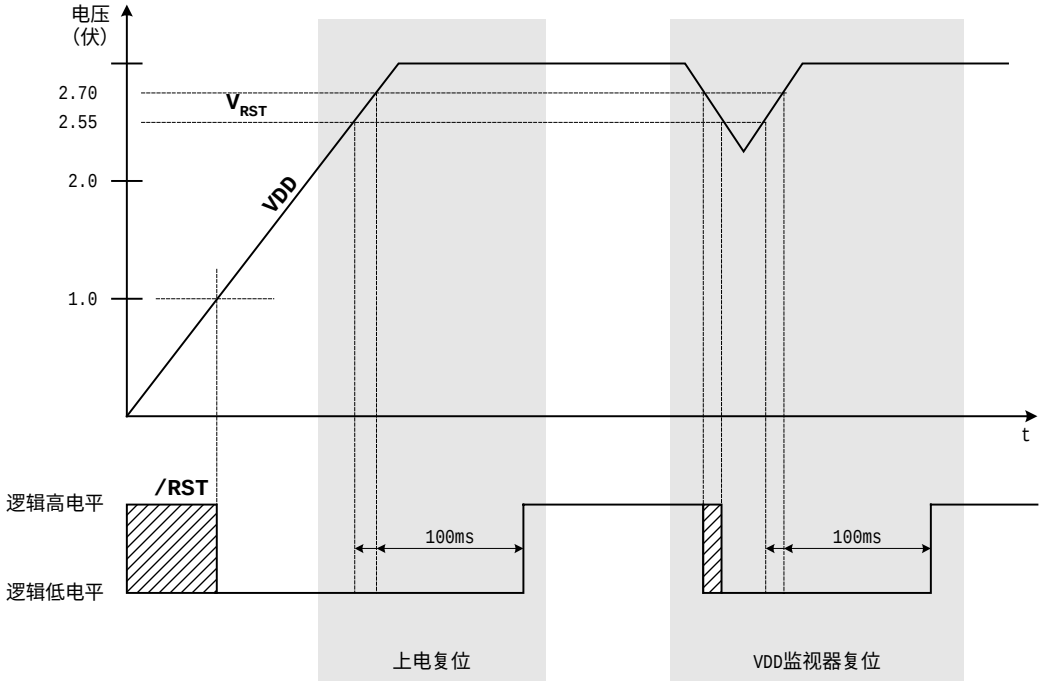


图 13.2 复位时序图

13.2 掉电复位

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

13.3 外部复位

外部/RST 引脚提供了使用外部电路强制 MCU 进入复位状态的手段。在/RST 引脚上加一个低电平有效信号将导致 MCU 进入复位状态。最好能提供一个外部上拉和/或对/RST 引脚去耦以防止强噪声引起复位。在低有效的/RST 信号撤出后，MCU 将保持在复位状态至少 12 个时钟周期。从外部复位状态退出后，PINRSF 标志（RSTSRC.0）被置位。

13.4 软件强制复位

向 SWRSEF 位写 1 将强制产生一个上电复位，如 13.1 节所述。

13.5 时钟丢失检测器复位

时钟丢失检测器实际上是由 MCU 系统时钟触发的单稳态电路。如果未收到系统时钟的时间大于 100 微秒，单稳态电路将超时并产生复位。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。置位 MCDRSF 标志—RSTSRC.2 将使能时钟丢失检测器。

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 之前用 CPT0CN.7（见 10. 比较器）使能比较器 0，以防止通电瞬间在输出端产生抖动，从而产生不希望的复位。比较器 0 复位是低电平有效：如果同相端输入电压（CP0+引脚）小于反相端输入电压（CP0-引脚），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；否则该位被清‘ 0 ’。/RST 引脚的状态不受该复位的影响。

13.7 外部 CNVSTR0 引脚复位

向 CNVRSEF 标志（RSTSRC.6）写‘ 1 ’可以将外部 CNVSTR0 号配置为复位源。CNVSTR0 号可以出现在 P0、P1、P2 或 P3 的任何 I/O 引脚，见“ 18.1 端口 0 – 端口 3 和优先权交叉开关译码器 ”。注意：交叉开关必许被配置为使 CNVSTR0 号接到正确的端口 I/O。应该在将 CNVRSEF 置‘ 1 ’之前配置并使能交叉开关。当被配置为复位源时，CNVSTR0 为低电平有效。在发生 CNVSTR0 复位之后，CNVRSEF 标志（RSTSRC.6）的读出值为‘ 1 ’，表示本次复位源为 CNVSTR；否则该位读出值为‘ 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 将溢出并产生一个复位，这可以防止系统失控。

在从任何一种复位退出时，WDT 被自动使能并使用缺省的最大时间间隔运行。系统软件可以根据需要禁止 WDT 或将其锁定为运行状态以防止意外产生的禁止操作。WDT 一旦被锁定，在下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。

WDT 是一个 21 位的使用系统时钟的定时器。该定时器测量对其控制寄存器的两次特定写操作

的时间间隔。如果这个时间间隔超过了编程的极限值，将产生一次 WDT 复位。可以根据需要用软件使能和禁止 WDT，或根据要求将其设置为永久性使能状态。看门狗的功能可以通过看门狗定时器控制寄存器（WDTCN）控制，见图 13.3。

13.8.1 使能/复位 WDT

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

13.8.2 禁止 WDT

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

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

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

13.8.3 禁止 WDT 锁定

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

13.8.4 设置 WDT 定时间隔

WDTCN.[2:0]控制看门狗超时间隔。超时间隔由下式给出：

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

对于 3MHz 的系统时钟，超时间隔的范围是 0.021ms 到 349.5ms。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，WDT.[2:0]为 111b。

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
SFR 页： 所有页

位 7-0： WDT 控制
写入 0xA5 将使能并重新装载 WDT。
写入 0xDE 后四个系统周期内写入 0xAD，将禁止 WDT。
写入 0xFF 将锁定禁止功能。

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

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

图 13.3 WDTCN: 看门狗定时器控制寄存器

R	R/W	R/W	R/W	R	R/W	R/W	R/W	复位值
-	CNVRSEF	C0RSEF	SWRSEF	WDTRSF	MCDRSF	PORSF	PINRSF	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xEF
SFR 页： 0

位 7： 保留

位 6： CNVRSEF：转换启动 0 复位源使能和标志。

写：0：CNVSTR0 不是复位源。
1：CNVSTR0 是复位源（低电平有效）。
读：0：前面的复位不是来自 CNVSTR0。
1：前面的复位来自 CNVSTR0。

位 5： C0RSEF：比较器 0 复位使能和标志

写：0：比较器 0 不是复位源。
1：比较器 0 是复位源（低电平有效）。
读：0：前面的复位不是来自比较器 0。
1：前面的复位来自比较器 0。

位 4： SWRSF：软件强制复位和标志

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

位 3： WDTRSF：看门狗定时器复位标志

0：前面的复位不是来自 WDT 超时。
1：前面的复位来自 WDT 超时。

位 2： MCDRSF：时钟丢失检测器标志

写：0：时钟丢失检测器禁止。
1：时钟丢失检测器使能。如果检测到时钟丢失条件，则触发复位。
读：0：前面的复位不是来自时钟丢失检测器超时。
1：前面的复位来自时钟丢失检测器超时。

位 1： PORSF：上电复位标志

写：如果 VDD 监视器被使能（MONEN 引脚接逻辑高电平），可以通过写该位来选择 VDD 监视器为复位源。
0：不选择 VDD 监视器为复位源。
1：选择 VDD 监视器为复位源。

重要：在上电时，通过外部 VDD 监视器使能引脚（MONEN）来使能/禁止 VDD 监视器。PORSF 位并不使能或禁止 VDD 监视器电路，它只是选择 VDD 监视器为复位源。

读：发生上电复位后该位被置‘1’。这可能是真正的上电复位，也可能是 VDD 监视器复位。无论哪一种情况，复位后数据存储器的内容都应被视为不确定。
0：前面的复位不是来自上电或 VDD 监视器复位。
1：前面的复位来自上电或 VDD 监视器复位。
注：当该标志位的读出值为‘1’时，所有其它复位标志都是不确定的。

位 0： PINRSF：硬件引脚复位标志

写：0：无影响。
1：强制产生一次上电复位。／RST 引脚被驱动为低电平。
读：0：前面的复位不是来自/RST 引脚。
1：前面的复位来自／RST 引脚。

图 13.4 RSTSRC：复位源寄存器

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		50		µA
/RST 输出有效 VDD		1.0			V
/RST 输出有效 AV+		1.0			V
VDD POR 门限(V_{RST})	注 1	2.40	2.55	2.70	V
产生系统复位的最小/RST 低电平时间		10			ns
复位时间延迟	从 VDD 超过复位门限（V_{RST}）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
注 1：当工作频率大于 50 MHz 时，最小 VDD 电源电压为 3.0 V。					

14. 振荡器

C8051F12x 和 C8051F13x 器件包含一个内部振荡器和一个外部振荡器驱动电路。可以使用 OSCICN 和 OSCICL 寄存器（如图 14.1 所示）来使能/禁止和校准内部振荡器。系统时钟可以由外部振荡器电路、内部振荡器或片内锁相环（PLL）提供。表 14.1 给出了振荡器电路的电气特性。

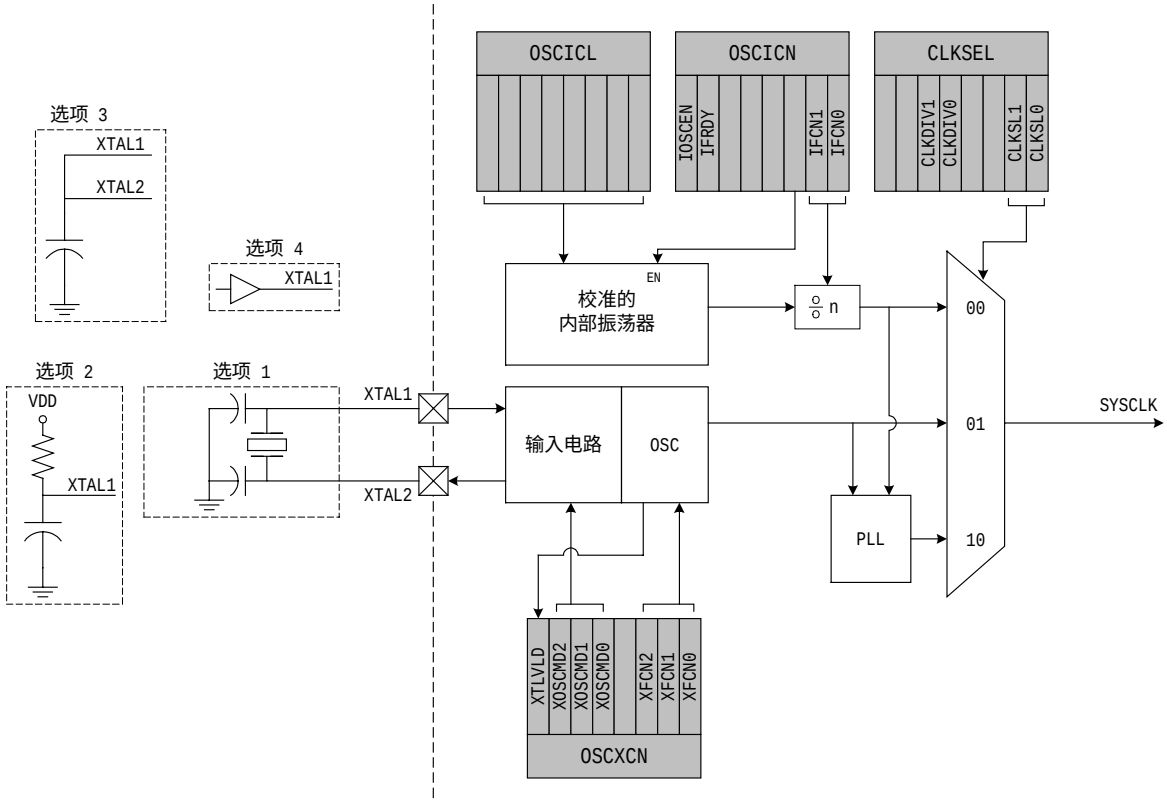


图 14.1 振荡器框图

表 14.1 振荡器电气特性

-40℃ ～+85℃（除非另有说明）。

参　数	条　件	最小值	典型值	最大值	单位
校准的内部振荡器频率		24	24.5	25	MHz
内部振荡器供电电流(从 VDD)	OSCICN.7=1		400		μA
外部时钟频率		0		30	MHz
T_{XCH}（外部时钟高电平时间）		15			ns
T_{XCL}（外部时钟低电平时间）		15			ns

14.1 可编程内部振荡器

所有器件都包含一个可编程内部振荡器，该振荡器在系统复位后被默认为系统时钟。内部振荡器的周期可以通过 OSCICL 寄存器调整，见图 14.2。OSCICL 在出厂前已被校准，对应 24.5 MHz 的振荡频率。

表 14.1 给出了精确内部振荡器的电气特性。注意：系统时钟可以从内部振荡器分频得到，分频数由寄存器 OSCICN 中的 IFCN 位设定，可为 1、2、4 或 8。

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

SFR 地址：0x8B
SFR 页： F

位 7-0： OSCICL：内部振荡器校准寄存器。
该寄存器校准内部振荡器的周期。OSCICL 的复位值定义内部振荡器的基频。复位值已经过工厂校准，对应的基频为 24.5MHz。

图 14.2 OSCICL：内部振荡器校准寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
IOSCEN	IFRDY	-	-	-	-	IFCN1	IFCN0	11000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x8A
SFR 页： F

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

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

位 5-2： 保留

位 1-0： IFCN1-0：内部振荡器频率控制位
00：内部振荡器 8 分频。
01：内部振荡器 4 分频。
10：内部振荡器 2 分频。
11：内部振荡器不分频。

图 14.3 OSCICN：内部振荡器控制寄存器

14.2 外部振荡器驱动电路

外部振荡器电路可以驱动外部晶体、陶瓷谐振器、电容或 RC 网络。也可以使用一个外部 CMOS 时钟提供系统时钟。对于晶体和陶瓷谐振器配置，晶体/陶瓷谐振器必须并接到 XTAL1 和 XTAL2 引脚（见图 14.1，选项 1）。对于 RC、电容或 CMOS 时钟配置，时钟源应接到 XTAL2 和/或 XTAL1 引脚（见图 14.1，选项 2、3、4）。必须在 OSCXCN 寄存器中选择外部振荡器类型，还必须正确选择频率控制位 XFCN（见图 14.5）。

14.3 系统时钟选择

寄存器 CLKSEL 中的 CLKSL1-0 位选择用于产生系统时钟的振荡源。如果要选择外部振荡器作为系统时钟，必须将 CLKSL1-0 设置为 ‘ 01 ’。当选择内部振荡器或 PLL 作为系统时钟时，外部振荡器仍然可以给某些外设（例如定时器、PCA）提供时钟。系统时钟可以在内部振荡器和外部振荡器或 PLL 之间自由切换，只要所选择的振荡器被使能并稳定运行。内部振荡器的起动时间很短，因此可以在同一个 OSCICN 写操作中使能和选择内部振荡器。外部晶体和陶瓷谐振器通常需要较长的起动时间，应待其稳定后方可用作系统时钟。当外部振荡器稳定后，晶体有效标志（寄存器 OSCXCN 中的 XTLVLD）被硬件置 ‘ 1 ’。在晶体方式，为了防止读到假 XTLVLD 标志，软件在使能外部振荡器和检查 XTLVLD 之间至少应延时 1ms。RC 和 C 方式通常不需要起动时间。PLL 也需要一定的时间才能锁定到所期望的频率，一旦 PLL 锁定到正确的频率，PLL 锁定标志（寄存器 PLL0CN 中的 PLLLCK 位）被硬件置 ‘ 1 ’。

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	CLKDIV1	CLKDIV0	-	-	CLKSL1	CLKSL0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x97
SFR 页： F

位 7-6： 保留。
位 5-4： CLKDIV1-0：输出 SYSCLK 除数
该位用于在 SYSCLK 经交叉开关输出到端口引脚之前对其预分频。
00：输出为 SYSCLK。
01：输出为 SYSCLK/2。
10：输出为 SYSCLK/4。
11：输出为 SYSCLK/8。
有关将该信号输出到端口引脚的详细信息，见 “ 18. 端口输入/输出 ”。
位 3-2： 保留。
位 1-0： CLKSL1-0：系统时钟源选择位
00：SYSCLK 源自内部振荡器，分频数由 OSCICN 寄存器中的 IFCN 位决定。
01：SYSCLK 源自外部振荡器。
10：SYSCLK 源自 PLL。
11：保留

图 14.4 CLKSEL：系统时钟选择寄存器

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

SFR 地址：0x8C
SFR 页： F

位 7： XTLVLD：晶体振荡器有效标志
（只在 XOSCMD＝11x 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。

位 6-4： XOSCMD2-0：外部振荡器方式位
00x：外部振荡器关闭。
010：外部 CMOS 时钟方式（外部 CMOS 时钟输入到 XTAL1 引脚）。
011：外部 CMOS 时钟二分频方式（外部 CMOS 时钟输入到 XTAL1 引脚）
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 = 0，写 = 忽略。

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

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

晶体方式（电路见图 14.1，选项 1；XOSCMD=11x）
选择 XFCN 值匹配晶体振荡器频率。

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

C 方式（电路见图 14.1，选项 3；XOSCMD=10x）
对于所需的振荡频率选择 K 因子（KF）：
f = KF/(C * VDD)，其中：
f = 以 MHz 为单位的振荡频率
C = XTAL1、XTAL2 引脚上的电容值，以 pF 为单位

图 14.3 OSCXCN：外部振荡器控制寄存器

14.4 外部晶体举例

如果使用晶体或陶瓷谐振器作为 MCU 的外部振荡器源，则电路应为图 14.1 中的选项 1。应从图 14.5（OSCXCN 寄存器）表中的晶体列选择外部振荡器频率控制值（XFCN）。例如，一个 11.0592MHz 的晶体要求的 XFCN 值为 111b。

外部晶体振荡器被使能后，振荡器幅值检测电路需要一段稳定时间才能达到正确的偏置。在使能振荡器工作和检测 XTLVLD 位之间至少等待 1 ms，以防止过早将外部振荡器切换为系统时钟。在外部振荡器稳定之前就切换到外部振荡器可能导致不可预见的后果。建议的步骤如下：

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

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

14.5 外部 RC 举例

如果使用 RC 网络作为 MCU 的外部振荡器源，则电路为图 14.1 中的选项 2。电容不应大于 100pF，但如果使用很小的电容，则总电容可能主要由 PCB 的寄生电容决定。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。如果所期望的频率是 100kHz，选 R=246kΩ，C=50pF：

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

参考图 14.5 中的表，得到所需要的 XFCN 值为 010。

14.6 外部电容举例

如果使用外部电容作为 MCU 的外部振荡器源，则电路为图 14.1 中的选项 3。电容不应大于 100pF，但如果使用很小的电容，则总电容将主要由 PCB 的寄生电容决定。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 VDD = 3.0V，C=50pF：

f = KF / (C * VDD) = KF / (50*3)
f = KF / 150

如果所需要的频率大约为 50kHz，从图 14.5 的表中选择 K 因子，得到 KF=7.7：

f = 7.7 / 150 = 0.051MHz，或 51kHz。

因此，本例中要用的 XFCN 值为 010。

14.7 锁相环（PLL）

C8051F12x 和 C8051F13x 系列器件包含一个锁相环（PLL），用于倍增内部振荡器或外部时钟源的频率，以获得较高的 CPU 工作频率。PLL 电路被设计为能从 5 MHz 和 30 MHz 之间的一个分频参考频率产生 25 MHz 和 100 MHz 之间的 CPU 工作频率。图 14.6 给出了 PLL 的原理框图。

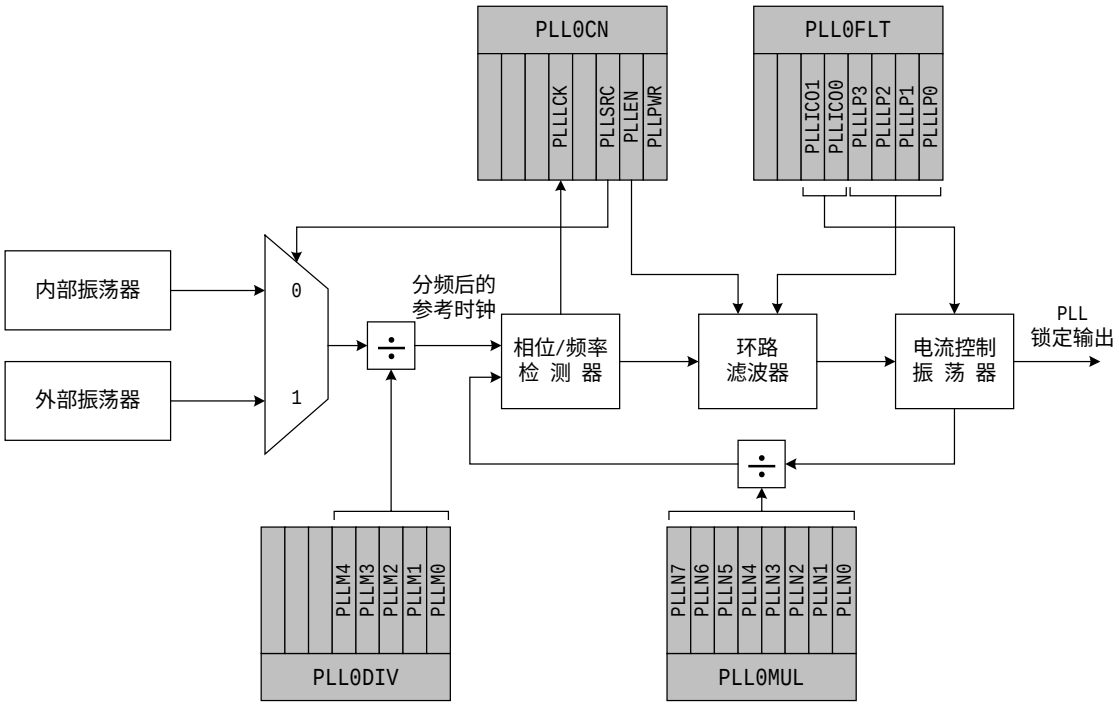


图 14.6 PLL 原理框图

14.7.1 PLL 输入时钟和预分频器

PLL 电路能从内部振荡器或外部时钟源获得参考时钟。PLLSRC 位（PLL0CN.2）控制参考时钟源的选择（见图 14.7）。如果 PLLSRC 为‘ 0 ’，则使用内部振荡器。注意，内部振荡器分频系数（由寄存器 OSCICN 中的 IFCN1-0 位决定）也适用于该时钟。当 PLLSRC 被置‘ 1 ’时，使用外部振荡源作为参考时钟。外部振荡器应在被选择为 PLL 电路参考时钟之前被使能并达到稳定。参考时钟在进入 PLL 电路之前被分频，分频系数由 PLL 预分频寄存器（PLL0DIV）中 PLLM4-0 位的内容决定，见图 14.8。

14.7.2 PLL 倍频和输出时钟

PLL 电路将分频后的参考时钟乘以保存在 PLL0MUL 寄存器（见图 14.9）中的倍频系数。为达此目的，它使用了由相位/频率检测器、环路滤波器、电流控制振荡器（ICO）组成的反馈环。根据期望的频率范围来配置环路滤波器和 ICO 是很重要的。应根据分频后的参考时钟频率设置 PLLLP3-0 位（PLL0FLT.3-0），根据所期望的输出频率范围来设置 PLLICO1-0 位（PLL0FLT.5-4）。图 14.10 对 PLLLP3-0 和 PLLICO1-0 位的正确设置作出了说明。当 PLL 锁定并稳定在所期望的频率时，PLLLCK 位（PLL0CN.5）被置‘ 1 ’。PLL 的输出频率由下面的方程设定：

$$PLL\text{频率} = \text{参考频率} \times \frac{PLLN}{PLLM}$$

其中“参考频率”所选时钟源的频率，PLLN 是 PLL 倍频系数，PLLM 是 PLL 预分频系数。

14.7.3 上电和 PLL 初始化

为了在器件上电后设置和使用 PLL 作为系统时钟，应遵循下列步骤：

1．确保要使用的参考时钟（内部或外部）处于稳定运行状态。
2．设置 PLLSRC 位（PLL0CN.2），为 PLL 选择时钟源。
3．将 FLASH 读定时控制位 FLRT（FLSCL.5-4）编程为适合于新时钟频率的值（见 15. FLASH 存储器）。
4．通过将 PLLPWR（PLL0CN.0）置‘1’给 PLL 上电。
5．对 PLL0DIV 寄存器编程，为 PLL 产生分频参考频率。
6．将 PLLLP3-0 位（PLL0FLT.3-0）编程为适合于分频参考频率的值。
7．将 PLLICO1-0 位（PLL0FLT.5-4）编程为适合于 PLL 输出频率的值。
8．将 PLL0MUL 寄存器编程为所期望的时钟倍频系数。
9．等待至少 5μS，以提供快速频率锁定。
10．通过将 PLLEN（PLL0CN.1）置‘1’来使能 PLL。
11．查询 PLLLCK（PLL0CN.4），直到该位从‘0’变到‘1’。
12．使用 CLKSEL 寄存器将系统时钟源切换到 PLL。

如果要在 PLL 已经运行时改变 PLL 的设置，则应应遵循下列步骤：

1．使用 CLKSEL 寄存器将系统时钟源切换到处于稳定运行状态的内部或外部时钟源。
2．确保新 PLL 设置要使用的参考时钟（内部或外部）处于稳定运行状态。
3．设置 PLLSRC 位（PLL0CN.2），为 PLL 选择时钟源。
4．如果要转向较高的频率，将 FLASH 读定时控制位 FLRT（FLSCL.5-4）编程为适合于新时钟频率的值（见 15. FLASH 存储器）。
5．通过将 PLLEN（PLL0CN.1）清‘0’来禁止 PLL。
6．对 PLL0DIV 寄存器编程，为 PLL 产生分频参考频率。
7．将 PLLLP3-0 位（PLL0FLT.3-0）编程为适合于分频参考频率的值。
8．将 PLLICO1-0 位（PLL0FLT.5-4）编程为适合于 PLL 输出频率的值。
9．将 PLL0MUL 寄存器编程为所期望的时钟倍频系数。
10．通过将 PLLEN（PLL0CN.1）置‘1’来使能 PLL。
11．查询 PLLLCK（PLL0CN.4），直到该位从‘0’变到‘1’。
12．使用 CLKSEL 寄存器将系统时钟源切换到 PLL。
13．如果要转向较低的频率，将 FLASH 读定时控制位 FLRT（FLSCL.5-4）编程为适合于新时钟频率的值（见 15. FLASH 存储器）。**注意：当 FLRT 位改变为较低值时，应禁止高速缓存读、高速缓存写和指令预取引擎。**

若要关闭 PLL，则首先应使用 CLKSEL 寄存器将系统时钟切换到内部振荡器或稳定的外部时钟源，然后通过将 PLLEN（PLL0CN.1）清‘0’来禁止 PLL，最后通过将 PLLPWR（PLL0CN.0）清‘0’给 PLL 断电。注意，PLLEN 和 PLLPWR 位可以被同时清‘0’。

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	复位值
-	-	-	PLLLCK	0	PLLSRC	PLLEN	PLLPWR	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x89
SFR 页： F

位 7-5： 未用：读 = 000b，写 = 忽略。
位 4： PLLLCK：PLL 锁定标志。
0：PLL 频率未锁定。
1：PLL 频率已锁定。
位 3： 保留。必须写‘ 0 ’。
位 2： PLLSRC：PLL 参考时钟源选择位。
0：PLL 参考时钟源为内部振荡器。
1：PLL 参考时钟源为外部振荡器。
位 1： PLLEN：PLL 使能位。
0：PLL 保持在复位状态。
1：PLL 被使能。PLLPWR 必须为‘ 1 ’。
位 0： PLLPWR：PLL 电源使能位。
0：PLL 偏置发生器被禁止。没有静态功耗
1：PLL 偏置发生器被使能。要使 PLL 工作，该位必须为‘ 1 ’。

图 14.7 PLL0CN：PLL 控制寄存器

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	复位值
-	-	-	PLLM4	PLLM3	PLLM2	PLLM1	PLLM0	00000001
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x8D
SFR 页： F

位 7-5： 未用：读 = 000b，写 = 忽略。
位 4-0： PLLM4-0：PLL 参考时钟预分频位。
这些位选择 PLL 参考时钟的预分频系数。当被设置为非 0 值时，参考时钟将被除以 PLLM4-0 中的值。当被设置为‘ 00000b ’时，参考时钟将被除以 32。

图 14.8 PLL0DIV：PLL 预分频寄存器

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	复位值
PLLN7	PLLN6	PLLN5	PLLN4	PLLN3	PLLN2	PLLN1	PLLN0	00000001
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x8E
SFR 页： F

位 7-0： PLLN7-0：PLL 倍频系数。
这些位选择分频 PLL 参考时钟的倍频系数。当被设置为非 0 值时，倍频系数等于 PLLN7-0 中的值。当被设置为 ‘ 00000000b ’ 时，倍频系数等于 256。

图 14.9 PLL0MUL：PLL 时钟倍频寄存器

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	PLLICO1	PLLICO0	PLLLP3	PLLLP2	PLLLP1	PLLLP0	00110001
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x8F
SFR 页： F

位 7-6： 未用：读 = 00b，写 = 忽略。
位 5-4： PLLICO1-0：PLL 电流控制振荡器控制位。
根据所期望的输出频率选择这两位的值，见下表：

PLL 输出时钟	PLLICO1-0
65 – 100 MHz	00
45 – 80 MHz	01
30 – 60 MHz	10
25 – 50 MHz	11

位 3-0： PLLLP3-0：PLL 环路滤波器控制位。
根据分频后的 PLL 参考时钟选择这两位的值，见下表：

分频后的 PLL 参考时钟	PLLLP3-0
19 – 30 MHz	0001
12.2 – 19.5 MHz	0011
7.8 – 12.5 MHz	0111
5 – 8 MHz	1111

图 14.10 PLL0FLT：PLL 滤波器寄存器

表 14.2 PLL 频率特性

-40℃ ～+85℃（除非另有说明）。

参　数	条　件	最小值	典型值	最大值	单位
输入频率 （分频后的参考频率）		5		30	MHz
PLL 输出频率 （C8051F120/1/2/3 和 C8051F130/1/2/3）		25		100	MHz
PLL 输出频率 （C8051F124/5/6/7）		25		50	MHz

表 14.3 PLL 锁定时间特性

-40℃ ～+85℃（除非另有说明）。

输入频率	倍频系数	PLL0FLT 设置	输出频率	最小值	典型值	最大值	单位
5 MHz	20	0x0F	100 MHz		202		μs
	13	0x0F	65 MHz		115		μs
	16	0x1F	80 MHz		241		μs
	9	0x1F	45 MHz		116		μs
	12	0x1F	60 MHz		258		μs
	6	0x1F	30 MHz		112		μs
	10	0x3F	50 MHz		263		μs
	5	0x3F	25 MHz		113		μs
25 MHz	4	0x01	100 MHz		42		μs
	2	0x01	50 MHz		33		μs
	3	0x11	75 MHz		48		μs
	2	0x11	50 MHz		17		μs
	2	0x21	50 MHz		42		μs
	1	0x21	25 MHz		33		μs
	2	0x31	50 MHz		60		μs
	1	0x31	25 MHz		25		μs

15. FLASH 存储器

C8051F120x 和 C8051F13x 系列 MCU 内部有 128KB（C8051F12x 和 C8051F130/1）或 64KB（C8051F132/3）的在系统可编程 FLASH 存储器，用于程序代码和非易失性数据存储。另外还有一个 256 字节的 FLASH 页，可以用于数据存储。可以通过 JTAG 接口对 FLASH 存储器进行在系统编程或由应用软件使用 MOVX 指令编程。一个 FLASH 位一旦被清‘ 0 ’，必须经过擦除才能再回到‘ 1 ’状态。在进行重新编程之前，应将数据字节擦除（置为 0xFF）。写和擦除操作由硬件自动定时，以保证操作正确。在一次擦除或写操作期间，FLSTAT 寄存器中的 FLBUSY 位被置‘ 1 ’（见图 16.8）。在此期间，位于指令预取缓冲器或转移地址高速缓存中的指令可以被执行，但如果必须从 FLASH 存储器中读取指令数据，则处理器将停止执行，直到擦除或写操作结束。如果当前执行的指令位于指令预取引擎或高速缓存，则被预装到转移地址高速缓存中的中断此时可以得到服务。任何没有被预装到高速缓存或在 CPU 停止执行时发生的中断都将在 FLASH 写/擦除操作期间被挂起，在 FLASH 操作结束后按优先级顺序得到服务。表 15.1 给出了 FLASH 存储器的电气特性。

15.1 FLASH 存储器编程

对 FLASH 存储器编程的最简单的方法是使用由 Silicon Labs 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“ 25. JTAG（IEEE 1149.1）”。

可以用软件使用 MOVX 指令对 FLASH 存储器编程，象一般的操作数一样为 MOVX 指令提供待编程的地址和数据字节。在使用 MOVX 指令对 FLASH 存储器写入之前，必须将程序存储写允许位 PSWE（PSCTL.0）设置为逻辑‘ 1 ’，以允许 FLASH 写操作。这将使 MOVX 指令执行对 FLASH 的写操作而不是对 XRAM 写入。在用软件清除之前 PSWE 位一直保持置位状态。为了避免对 FLASH 的误写，建议在 PSWE 为逻辑‘ 1 ’期间禁止中断。

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

对于含有 128KB FLASH 存储器的器件，在执行 FLASH 写、读、擦除操作时，PSBANK 寄存器（见图 11.3）中的 COBANK 位决定三个高地址 FLASH 块中的哪一个被映射到地址范围 0x08000 ~ 0x0FFFF。

对于含有 64KB FLASH 存储器的器件，在执行 FLASH 写、读、擦除操作时，COBANK 位应保持为‘ 01 ’，以保证 FLASH 写、擦除和读操作有效。

注意：为保证 FLASH 存储器内容的完整性，强烈建议在任何从应用软件写和/或擦除 FLASH 存储器的系统中使能 VDD 监视器（通过将 VDD 监视器使能引脚 MONEN 连接到 VDD 和将 RSTSRC 寄存器中的 PORSF 位置 1）。更详细的信息见“ 13. 复位源 ”。

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字节地址。

写/擦除操作的定时由硬件自动控制。注意：对于 128KB FLASH 器件，从 0x1FC00 开始的 1024 个单元被保留。应避免 FLASH 写和擦除操作指向保留区

表 15.1 FLASH 电气特性

VDD=2.7 - 3.6V, -40℃到+85℃

参 数	条 件	最小值	典型值	最大值	单 位
FLASH 容量*	C8051F12x 和 C8051F130/1		131328†		字节
FLASH 容量*	C8051F132/3		65792		字节
擦写寿命		20k	100k		擦/写
擦除时间		10	12	14	ms
写入时间		40	50	60	μs

*包括 256 字节的临时存储区
†位于 0x1FC00 ~ 0x1FFF 的 1024 个单元被保留。

15.1.1 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。数据写入和擦除使用 MOVX 指令（详见 15.1.2 和 15.1.3），读出用 MOVC 指令。在写或擦除地址大于 0x7FFF 时，由 PSBANK 寄存器（见图 11.3）中的 COBANK 位控制 FLASH 存储器中的哪一部分是目标操作块。对于含有 64KB FLASH 存储器的器件，在执行 FLASH 写、读、擦除操作时，COBANK 位应保持为 ‘ 01 ’，以保证 FLASH 写、擦除和读操作有效。

FLASH 存储器中有两个附加的 128 字节的扇区（共 256 字节），只能用于非易失性数据存储。它较小的扇区规模使其特别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。若要修改一个多字节数据集中的某一个字节，数据集必须被移动到临时存储区域。128 字节的扇区规模使数据更新更加容易，可以不浪费程序存储器或 RAM 空间。这两个 128 字节的扇区在 128K 字节 FLASH 存储器中是双映射的，只能用 MOVC 读和用 MOVX 写；它们的地址范围是 0x00 ~ 0x7F 和 0x80 ~ 0xFF（见图 15.2）。要访问这两个 128 字节的扇区，PSCTL 寄存器中的 SFLE 位必须被设置为逻辑 ‘ 1 ’。这两个 128 字节的扇区不能用于存储程序代码。两个 128 字节的扇区可以被分别擦除或同时擦除。要同时擦除这两个扇区，擦除操作的目标地址应为 0x0400，SFLE 位应被置‘ 1 ’。图 15.1 给出了对应不同 COBANK 和 SFLE 设置情况的存储器映射图。

<table>
<tr><th colspan="4">SFLE = 0</th><th>SFLE = 1</th><th>内部
地址</th></tr>
<tr><th>COBANK = 0</th><th>COBANK = 1</th><th>COBANK = 2</th><th>COBANK = 3</th><th></th><th></th></tr>
<tr><td>块 0</td><td>块 1</td><td>块 2</td><td>块 3</td><td rowspan="2">未定义</td><td>0xFFFF

0x8000</td></tr>
<tr><td>块 0</td><td>块 0</td><td>块 0</td><td>块 0</td><td>0x7FFF</td></tr>
<tr><td colspan="4" rowspan="2"></td><td>临时存储区
(2)</td><td>0x00FF

0x0000</td></tr>
</table>

图 15.1　MOVC 和 MOVX 操作的 FLASH 存储器映像

15.1.2 软件擦除 FLASH 页

一次 FLASH 擦除操作将擦除整个扇区（扇区内的所有字节被置为 0xFF）。128k 字节的 FLASH 存储器是以 1024 字节的扇区为单位组织的。256 字节的临时数据区（地址 0x20000 ~ 0x200FF）由两个 128 字节的扇区（页）组成。要擦除一个 FLASH 页，FLWE、PSWE 和 PSEE 位必须被置‘ 1 ’，然后用 MOVX 指令写一个数据字节到扇区内的任何一个地址。用软件擦除一个 FLASH 页的建议步骤如下：

1．禁止中断。
2．如果擦除块 1、块 2 或块 3 中的某一页，设置 COBANK 位（PSBANK.5-4）以选择正确的块。
3．如果擦除位于临时区的某一页，将 SFLE 位（PSCTL.2）置‘ 1 ’。
4．置位 FLWE（FLSCL.0），以允许用户软件写/擦除 FLASH。
5．置位 PSEE（PSCTL.1），以允许 FLASH 扇区擦除。
6．置位 PSWE（PSCTL.0），以允许 FLASH 写。
7．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。
8．清除 PSEE 以禁止 FLASH 扇区擦除。
9．清除 PSWE 位，使 MOVX 命令指向 XRAM 数据空间。
10．清除 FLWE 位，以禁止 FLASH 写/擦除。
11．如果被擦除的页位于临时区，清除 SFLE 位。
12．重新允许中断。

15.1.3 软件写 FLASH 存储器

可以每次向 FLASH 存储器写一个字节或一个小块。寄存器 CCH0CN（见图 16.4）中的 CHBLKW 位控制在 FLASH 写操作时写一个字节还是写一个字节块。当 CHBLKW 位被清‘ 0 ’时，对 FLASH 的写操作是每次写一个字节，当 CHBLKW 位被置‘ 1 ’时，对 FLASH 的程序空间每次写四个字节的存储块，对 FLASH 的临时区每次写两个字节的存储块。块写和单字节写需要相同的时间，当向 FLASH 存储器存储大量的数据时，块写可以节省时间。

FLASH 单字节写是每次写一个字节，FLASH 写操作在每条 MOVX 写指令后执行。对 FLASH 进行单字节写的建议步骤如下：

1. 禁止中断。
2. 清除 CHBLKW（CCH0CN.0）以选择单字节写方式。
3. 如果写位于块 1、块 2 或块 3 中的字节，设置 COBANK 位（PSBANK.5-4）以选择正确的块。
4. 如果写位于临时区的字节，将 SFLE 位（PSCTL.2）置‘ 1 ’。
5. 置位 FLWE（FLSCL.0），以允许用户软件写/擦除 FLASH。
6. 置位 PSWE（PSCTL.0），使 MOVX 指令写 FLASH。
7. 用 MOVX 指令向期望地址写入一个数据字节（如果需要，重复该步骤）。
8. 清除 PSWE 位，使 MOVX 命令指向 XRAM 数据空间。
9. 清除 FLWE 位，以禁止 FLASH 写/擦除。
10. 如果写操作的目的字节位于临时区，清除 SFLE 位。
11. 重新允许中断。

对于 FLASH 块写操作，真正的 FLASH 写过程是在用 MOVX 写指令写完存储块的最后一个字节之后开始的。当写位于程序存储块（四个 32KB 程序存储块中的一个）中的地址时，一次 FLASH 块写操作写四个字节，从末位地址 00b 到 11b。写操作必须按顺序进行（即必须按末位地址 00b, 01b, 10b, 11b 的顺序）。FLASH 内部写操作是在对末位地址为 11b 的目标单元执行完 MOVX 写指令后开始进行的。当写位于 FLASH 临时区中的地址时，一次 FLASH 块写操作写两个字节，从末位地址 0b 到 1b。FLASH 内部写操作是在对末位地址为 1b 的目标单元执行完 MOVX 写指令后开始进行的。块中不需要更新的任何字节都应被写入 0xFF。对 FLASH 进行块写的建议步骤如下：

1. 禁止中断。
2. 置位 CHBLKW（CCH0CN.0）以选择块写方式。
3. 如果写位于块 1、2 或 3 中的字节，设置 COBANK 位（PSBANK.5-4）以选择正确的块。
4. 如果写位于临时区的字节，将 SFLE 位（PSCTL.2）置‘ 1 ’。
5. 置位 FLWE（FLSCL.0），以允许用户软件写/擦除 FLASH。
6. 置位 PSWE（PSCTL.0），使 MOVX 指令写 FLASH。
7. 用 MOVX 指令向目的块写入数据字节。数据字节必须按顺序写，最后写入的字节必须是块中的高字节（详见上面的说明，如果需要，重复该步骤）。
8. 清除 PSWE 位，使 MOVX 命令指向 XRAM 数据空间。
9. 清除 FLWE 位，以禁止 FLASH 写/擦除。
10. 如果写操作的目的字节位于临时区，清除 SFLE 位。
11. 重新允许中断。

写/擦除定时由硬件自动控制。注意：从地址 0x1FC00 开始的 1024 个字节被保留。应避免对保留区进行写和擦除操作。

15.2 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。程序存储写允许位（PSCTL.0）、程序存储擦除允许位（PSCTL.1）和 FLASH 写/擦除允许位（FLACL.0）保护 FLASH 存储器不会被软件意外修改。在用软件写或擦除 FLASH 存储器之前，这些位必须被置‘1’。另外的安全功能是防止通过 JTAG 接口或通过运行在系统控制器上的软件读取或改写产权程序代码和数据常数。

安全锁定字节可以保护 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 16k 字节的存储器块。将读锁定字节中的一位清‘0’可防止通过 JTAG 接口读对应的 FLASH 存储器块。将写/擦除锁定字节中的一位清‘0’可防止通过 JTAG 接口写/擦除对应的存储器块。当对应的安全字节中的所有位都被清 0 后，临时存储扇区被读或写/擦除锁定。

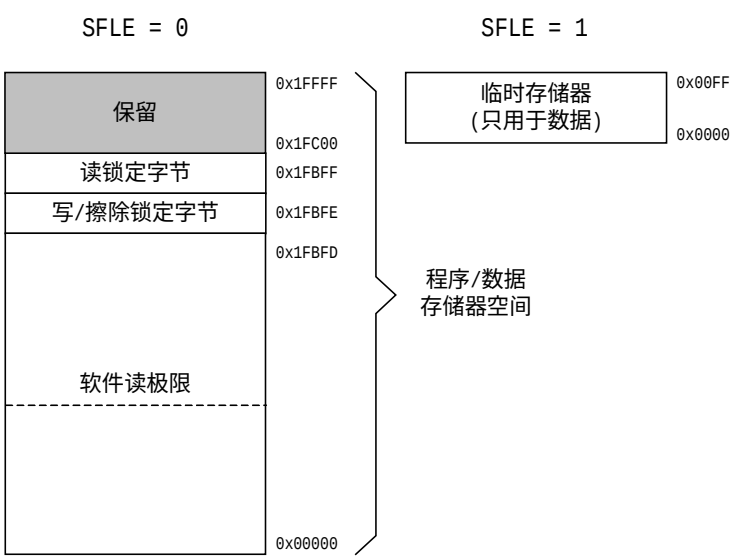
对于 C8051F12x 和 C8051F130/1，安全锁定字节位于 0x1FBFE（写/擦除锁定）和 0x1FBFF（读锁定），如图 15.2 所示。对于 C8051F132/3，安全锁定字节位于 0x0FFFE（写/擦除锁定）和 0x0FFFF（读锁定），如图 15.3 所示。包含锁定字节的 1024 字节扇区可以写入，但不能用软件擦除。对被读锁定的字节读取将返回无定义的数据。不能通过 JTAG 口调试位于读锁定扇区内的代码。不管安全字节所在存储块的安全设置如何，锁定位总是可读的并可以被清‘0’。这就允许在锁定了安全字节所在的存储块以后还可以追加要保护的存储块。

注意：为保护器件不被外部访问，包含锁定字节的块应被实施写/擦除锁定。对于 128KB FLASH 器件（C8051F12x 和 C8051F130/1），包含锁定字节的块是 0x18000 ~ 0x1BFFF，该块在写/擦除锁定字节的位 7 清 0 时被锁定。对于 64KB FLASH 器件（C8051F132/3），包含锁定字节的块是 0x0C000 ~ 0x0FFFF，该块在写/擦除锁定字节的位 3 清 0 时被锁定。如果包含安全字节的块未被写/擦除锁定，可以通过 JTAG 接口擦除该块，从而复位安全字节。

当包含安全字节的页被实施写/擦除锁定后，必须通过执行 JTAG 器件擦除操作才能对被安全字节锁定的任何一个区域解除锁定。使用任何一个安全字节地址执行 JTAG 擦除操作将自动启动对整个程序存储器空间的擦除（保留区除外）。该擦除操作只能通过 JTAG 口进行，而不能由运行在器件中的固件执行。

读和写/擦除安全位
(位 7 为 MSB.)

Bit	Memory Block
7	0x1C000 - 0x1FBFD
6	0x18000 - 0x1BFFF
5	0x14000 - 0x17FFF
4	0x10000 - 0x13FFF
3	0x0C000 - 0x0FFFF
2	0x08000 - 0x0BFFF
1	0x04000 - 0x07FFF
0	0x00000 - 0x03FFF



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，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

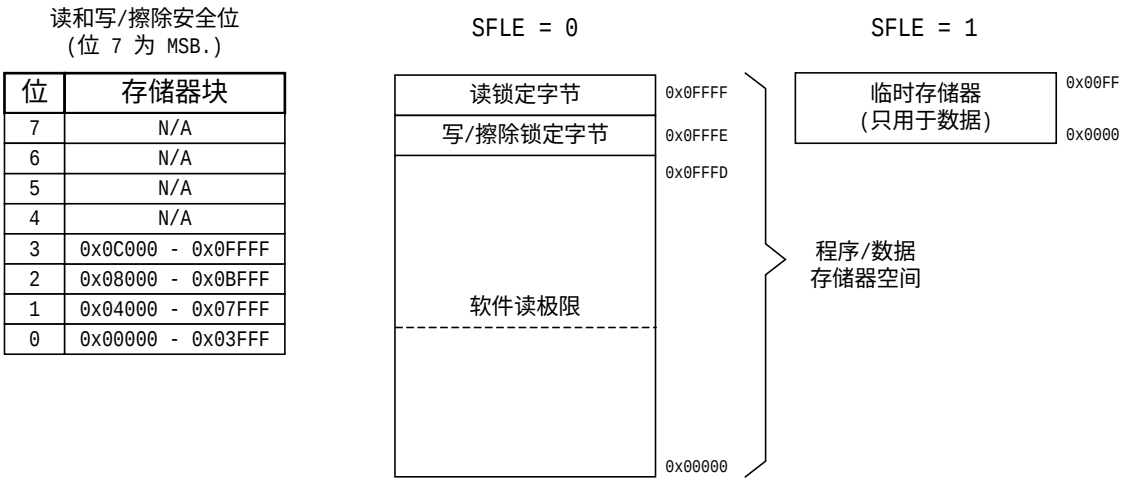
FLASH写/擦除锁定字节

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，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
注意：当最高块被锁定时，安全字节可以被写入，但不能被擦除。

FLASH访问极限寄存器（FLACL）

该寄存器的内容作为软件读极限地址的高 8 位。在该地址之上运行的固件不能用 MOVX 或 MOVC 指令读、写或擦除该地址以下的存储单元。

图 15.2 128KB FLASH 程序存储器组织和安全字节



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，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，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

注意：当最高块被锁定时，安全字节可以被写入，但不能被擦除。

FLASH访问极限寄存器（FLACL）

该寄存器的内容作为软件读极限地址的高 8 位。在该地址之上运行的固件不能用 MOVX 或 MOVC 指令读、写或擦除该地址以下的存储单元。

图 15.3 64KB FLASH 程序存储器组织和安全字节

FLASH 访问极限这一安全功能（见图 15.4）保护产权程序代码和数据不被运行在器件上的软件读取。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。

FLASH 访问极限（FAL）是一个 17 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 0x00000 到（但不包括）FAL 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 指令将返回不确定的数据。）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。复位后通过复位向量将控制转到增值固件。一旦增值固件完成初始化操作，程序将转到上分区中的预定位置。如果程序入口是公开的，运行在上分区中的软件就可以执行下分区中的代码，但不能读或改写下分区中的内容。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。

FAL 地址由 FLASH 访问寄存器中的内容指定。17 位 FAL 地址的高 8 位由 FLACL 寄存器的内容设定。因此，FAL 可位于程序存储器空间中以 512 字节为界的任何位置。然而 1024 字节的擦除扇区规模要求以 1024 字节为界。未初始化过的 FAL 安全字节的内容是 0x00，因此所设置的 SRL 地址为 0x00000，在这种缺省情况下允许读取全部程序存储器空间。

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
SFR 页： F

位 7-0： FLACL：FLASH 访问极限。
该寄存器保持 17 位程序存储器读／写／擦除极限地址的高 8 位。低 9 位读／写／擦除极限地址总是被设置为 0。向该地址写将设置 FLASH 访问极限地址。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。**为了完全保护位于该极限地址以下的所有地址，FLACL 的位 0 应为‘0’，以便与 FLASH 页边界对齐。**

图 15.4 FLACL: FLASH 访问极限寄存器

15.2.1 FLASH 安全选项小结

C8051F12x 和 C8051F13x 器件支持 3 种 FLASH 访问方法：

1．通过 JTAG 调试接口访问 FLASH；

2．从位于 FLASH 访问极限地址以下的固件访问 FLASH；

3．从位于 FLASH 访问极限地址之上（含该地址）的固件访问 FLASH。

通过 JTAG 调试接口访问 FLASH：

1. 读和写/擦除锁定字节（安全字节）提供对通过 JTAG 接口的 FLASH 访问保护。
2. 任何未被锁定的页都可以被读取、写入或擦除。
3. 被锁定的页不能被读取、写入或擦除。
4. 允许读安全字节。
5. 允许向安全字节写入以追加锁定页。
6. 如果包含安全字节的页未被锁定，则该页可以被直接擦除。**擦除该页将使安全字节复位，并解锁所有 FLASH 页。**
7. 如果包含安全字节的页被锁定，则该页不能被直接擦除。**要解锁包含安全字节所在页，需要执行 JTAG 器件擦除操作。**JTAG 器件擦除操作将擦除所有 FLASH 页，包括安全字节所在页和安全字节本身。
8. 任何时刻都不能对保留区进行读、写或擦除。

从位于 FLASH 访问极限地址以下的固件访问 FLASH：

1. 读和写/擦除锁定字节（安全字节）并不限制从用户固件访问 FLASH。
2. 除安全字节所在页外，任何 FLASH 页都可以被读取、写入或擦除。
3. **安全字节所在页不能被擦除。**解锁 FLASH 页只能通过 JTAG 接口执行。
4. 可以读或写安全字节所在页。可以通过写安全字节来锁定 FLASH 页，禁止通过 JTAG 接口访问。
5. 任何时刻都不能对保留区进行读、写或擦除。

从位于 FLASH 访问极限地址之上（含该地址）的固件访问 FLASH：

1. 读和写/擦除锁定字节（安全字节）并不限制从用户固件访问 FLASH。
2. 除安全字节所在页外，位于 FLASH 访问极限地址之上（含该地址）的任何 FLASH 页都可以被读取、写入或擦除。
3. 位于 FLASH 访问极限地址之下的任何 FLASH 页都不能被读取、写入或擦除。
4. 允许程序转移到 FLASH 访问极限地址之下的位置。
5. **安全字节所在页不能被擦除。**解锁 FLASH 页只能通过 JTAG 接口执行。
6. 安全字节所在页可以被读取或写入。可以通过写安全字节来锁定 FLASH 页，禁止通过 JTAG 接口访问。
7. 任何时刻都不能对保留区进行读、写或擦除。

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	FLRT		保留	保留	保留	FLWE	10000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xB7
SFR 页： 0

位 7-6： 未用。
位 5-4： FLRT：FLASH 读时间。
应根据系统时钟速度将这两位编程为最小允许值。
00：SYSCLK <= 25 MHz
01：SYSCLK <= 50 MHz
10：SYSCLK <= 75 MHz
11：SYSCLK <= 100 MHz
位 3-1： 保留。读 = 000b，写入值必须是 000b。
位 0： FLWE：FLASH 写/擦除允许。
该位必须置‘ 1 ’才能从用户软件写/擦除 FLASH。
0：禁止 FLASH 写/擦除。
1：允许 FLASH 写/擦除。

注意：当将 FLRT 位改变为较低值时（例如从 11b 变为 00b），应使用 CCH0CN 寄存器禁止高速缓存读、高速缓存写和指令预取（见图 16.4 CCH0CN：高速缓存控制寄存器）。

图 15.5 FLSCL：FLASH 存储器控制

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	-	SFLE	PSEE	PSWE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x8F
SFR 页： 0

位 7-3： 未使用。读 = 00000b，写 = 忽略。

位 2： SFLE：临时 FLASH 存储器访问允许。
当该位被置 1 时，用户软件对 FLASH 的读/写操作将指向两个 128 字节的 FLASH 临时存储扇区。当 SFLE 被设置为逻辑 ‘ 1 ’ 时，不应访问 0x00 – 0xFF 以外的地址范围（地址 0x400 例外，它可被用于同时擦除这两个临时区）。对该地址范围以外的地址进行读/写可能产生不可预料的结果。
0：从用户软件访问 FLASH 时将访问程序/数据 FLASH 扇区。
1：从用户软件访问 FLASH 时将访问两个 128 字节的临时存储器扇区。

位 1： PSEE：程序存储擦除允许。
将该位置 1 将允许擦除 FLASH 存储器中的一个页，前提是 PSWE 位也被置 1。在将该位置 1 后，用 MOVX 指令进行一次写操作将擦除包含 MOVX 指令寻址地址的那个 FLASH 页。用于写操作的数据可以是任意值。**注意：包含读锁定字节和写/擦除锁定字节的 FLASH 页不能用软件擦除。**
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擦除待写地址。
0：禁止写 FLASH 存储器。MOVX 操作指向外部 RAM。
1：允许写 FLASH 存储器。MOVX 操作指向 FLASH 存储器。

图 15.6 PSCTL：程序存储读写控制

16．转移地址高速缓存

C8051F12x 和 C8051F13x 系列器件包含一个 63 x 4 字节的转移地址高速缓存和一个 4 字节指令预取引擎。由于 FLASH 存储器的访问时间是 40 ns，而最短指令执行时间是 10 ns（C8051F120/1/2/3 和 C8051F130/1/2/3）或 20 ns（C8051F124/5/6/7），所以需要有转移地址高速缓存和预取引擎才能使程序全速执行。预取引擎每次从 FLASH 存储器读取 4 个指令字节，送给 CIP-51 处理器核执行。当运行线性代码时（程序没有任何转移），只需预取引擎就可使程序全速执行。当程序发生转移时，需要在转移地址高速缓存中查找目标地址。如果在高速缓存中找到转移地址信息（称为“高速缓存命中”），则从高速缓存读出指令数据并立即送给 CIP-51 执行，程序执行过程没有延时。如果在高速缓存中未找到转移地址（称为“高速缓存不命中”），则处理器暂时停止执行（最多 4 个时钟周期），从 FLASH 存储器中读取下一组 4 字节指令数据。每当发生高速缓存不命中时，所需要的指令数据便被写入高速缓存（如果当前的高速缓存设置允许学入）。CIP-51 与转移地址高速缓存和预取引擎之间的数据流图如图 16.1 所示。

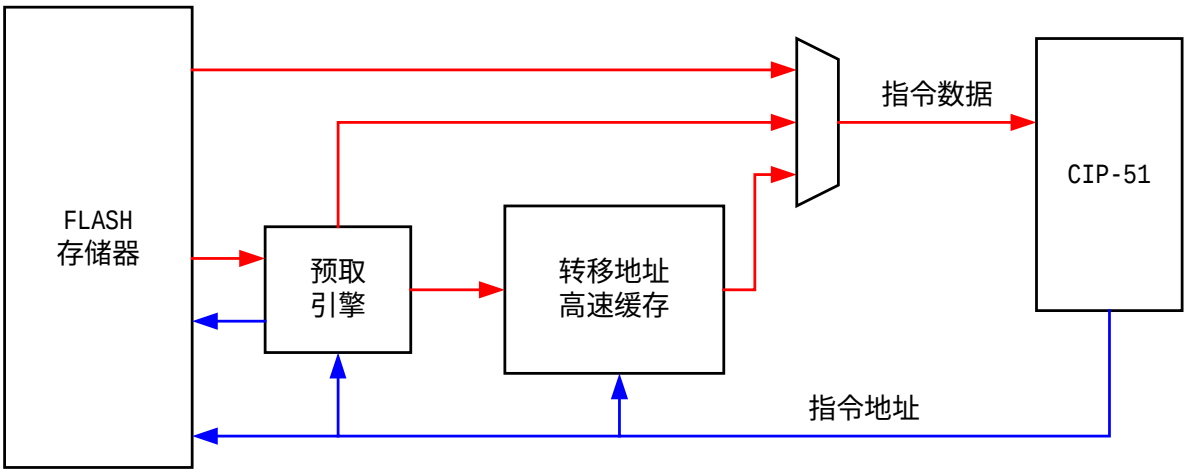


图 16.1　转移地址高速缓存数据流图

16.1 高速缓存和指令预取操作

转移地址高速缓存保存两类信息：“字块（slot）”和“标记（tag）”。字块保存从 FLASH 存储器读取的指令数据。每个字块包含 4 个连续的代码字节。标记含有一个 4 字节字块所对应 FLASH 地址的高 15 位。所以，指令数据总是以 4 字节为界被缓存。标记中还包含一个“有效位”，该位指示一个高速缓存字块中是否包含有效的指令数据。一个特殊的高速缓存位置（称为线性标记和字块）被保留，由预取引擎使用。高速缓存的组织如图 16.2 所示。每当需要进行 FLASH 读操作时，待读地址被与全部有效高速缓存标记地址比较（包括线性标记）。如果有任何一个标记地址与所要读的地址匹配，则对应字块中的数据被立即提供给 CIP-51。如果待读地址与预取引擎正在读取的指令中的一个地址匹配，则 CIP-51 暂停执行，直到预取操作完成。如果未发生匹配，则当前的预取操作被放弃，一次针对所需要的指令数据的预取操作被启动。当预取操作结束后，CIP-51 开始执行刚被读出的指令，同时预取引擎开始从 FLASH 存储器中读下一组 4 字节的指令数据。如果新读取的数据也符合被缓存的条件，则将被写入到高速缓存中由当前的替换算法给出的字块。

替换算法由高速缓存算法位 CHALGM（CCH0TN.3）选择。当 CHALGM 位被清‘ 0 ’时，高速缓存将使用回弹算法替换高速缓存中的数据。回弹算法按从高速缓存的开始到最后，然后再从高速缓存的最后到开始位置的顺序进行替换。当 CHALGM 位被置‘ 1 ’时，高速缓存将使用伪随机算法替换高速缓存中的数据。伪随机算法使用一个伪随机数来决定要被替换的高速缓存位置。可以通过向 CHFLUSH 位（CCH0CN.4）写‘ 1 ’来手动清空高速缓存。

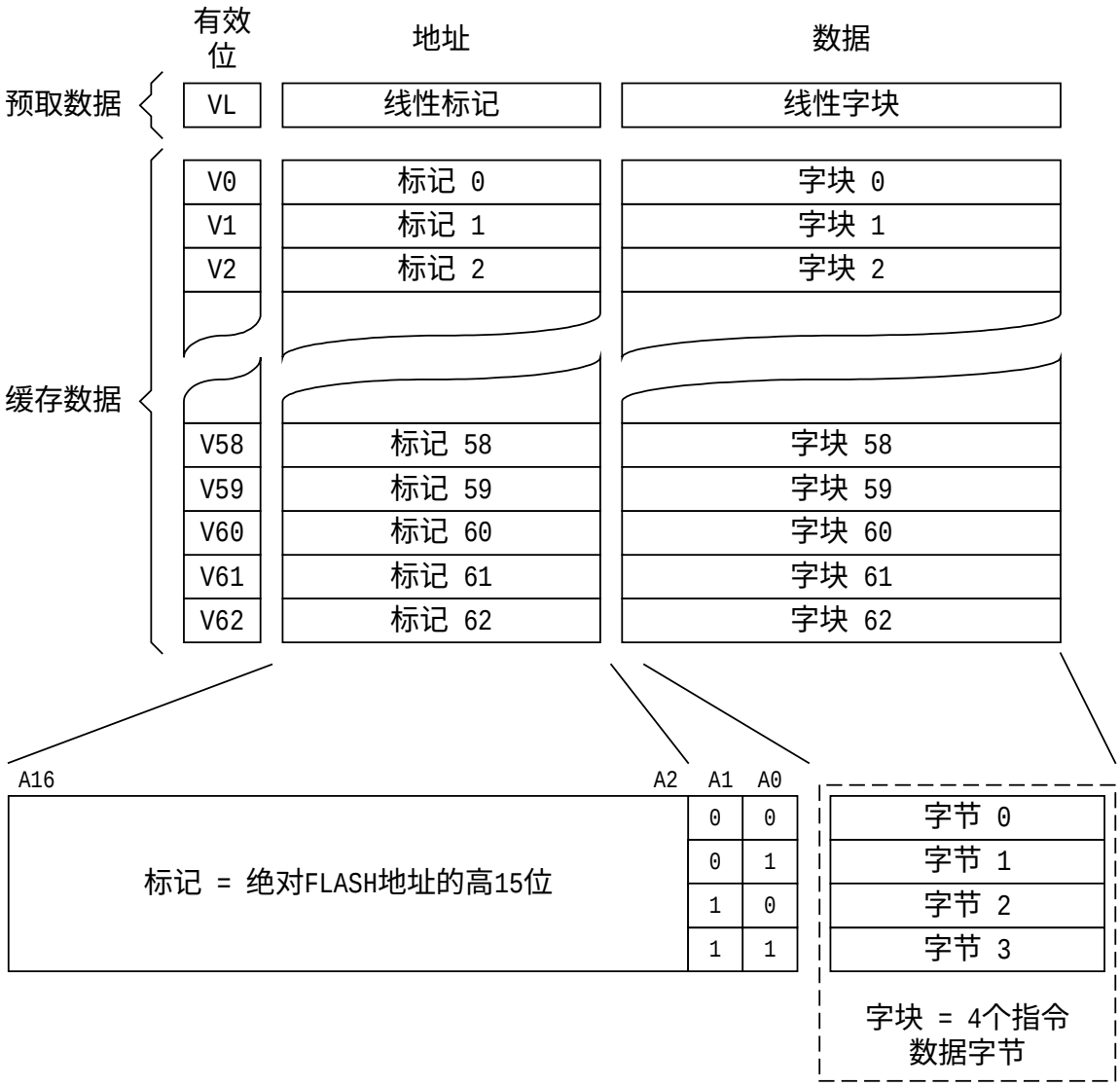


图 16.2 转移地址高速缓存组织

16.2 高速缓存和指令预取优化

转移地址高速缓存在缺省情况下被配置为对于多数情况可以提供代码执行速度的改善。**在大多数应用中，高速缓存控制寄存器应被保持在它们的复位状态。**有时也会希望优化一个特定例程或时间要求苛刻的循环程序的执行时间。转移地址高速缓存可以选择不缓存某些类型的数据或预装和锁定时间要求苛刻的转移地址以优化执行速度。

用高速缓存不命中记录阈值位 CHMSTH（CCH0TN.1-0）来实现对高速缓存的最基本控制。如果处理器在预取操作期间暂停执行的时钟周期数大于保存在 CHMSTH 中的数值，则所要求的数据在可用之后将被缓存。CHMSTH 位在缺省情况下被设置为 0，表示在任何时刻只要处理器暂停执行，新数据就将被缓存。如果（例如）CHMSTH 等于 2，任何导致 3 或 4 个时钟周期延时的高速缓存不命中事件都会使新数据被缓存，而引起 1 或 2 个时钟周期延时的高速缓存不命中事件不会使新数据被缓存。

某些类型的指令数据或某些代码块也可以被排除在高速缓存之外。缺省情况下，RETI 指令的目的地址不被缓存。为了能使 RETI 的目的地址被缓存，可以将 CHRETI 位（CCH0CN.3）设置为‘ 1 ’。一般来说，缓存 RETI 的目的地址没有意义，除非同一条指令经常被中断（例如，等待中断发生的代码循环）。中断服务程序（ISR）中的指令也可以被排除在高速缓存之外。缺省情况下，ISR 指令可以进入高速缓存，但可以通过将 CHISR 位（CCH0CN.2）清‘ 0 ’来禁止。另一种可以被明确排除在高速缓存之外的信息是由 MOVC 指令返回的数据。将 CHMOV 位（CCH0CN.1）清‘ 0 ’可以禁止 MOVC 的数据被缓存。如果 MOVC 缓存被允许，可以限制 MOVC 信息只使用字块 0（不允许高速缓存的压栈操作）。CHFIXM 位（CCH0TN.2）控制这一行为。

通过禁止所有的高速缓存写操作可以实现对高速缓存的进一步控制。可以通过将 CHWREN 位（CCH0CN.7）清‘ 0 ’来禁止高速缓存写操作。尽管正常的高速缓存写操作（在发生高速缓不命中之后的那些写操作）被禁止，仍然可以通过高速缓存压栈操作向高速缓存写入数据。禁止高速缓存写操作可以防止非关键代码改变高速缓存的内容。注意，不管 CHWREN 的值如何，FLASH 写或擦除操作自动将受影响的字节从高速缓存中排除。高速缓存读操作和预取引擎还可以被单独禁止。禁止高速缓存读操作强制所有指令数据都从 FLASH 存储器或预取引擎中执行。将 CHRDEN 位（CCH0CN.6）清‘ 0 ’可以禁止高速缓存读操作。注意，当高速缓存读操作被禁止时，仍然可以进行高速缓存写操作（如果 CHWREN 被置‘ 1 ’）。用 CHPFEN 位（CCH0CN.5）来禁止预取引擎。当该位被清‘ 0 ’时，预取引擎被禁止。如果 CHPFEN 和 CHRDEN 均为‘ 0 ’，程序将以固定的速度执行，因为此时指令都取自 FLASH 存储器。

可以向高速缓存预装和锁定时间要求苛刻的转移目的地址。例如，在一个具有需要尽快响应的 ISR 的系统中，ISR 的入口可以被锁定在一个高速缓存地址以使 ISR 的响应延迟最小。一次可以锁定最多 61 个高速缓存位置。通过对 CHPUSH 位（CCH0LC.7）编程来使能高速缓存的压栈操作，从而将指令锁定在高速缓存中。当 CHPUSH 被置‘ 1 ’时，MOVC 指令将使包含所读数据字节的 4 字节代码段被写入到由 CHSLOT（CCH0LC.5-0）指示的高速缓存字块。然后，CHSLOT 被减 1，指向下一个可锁定的高速缓存位置。该过程被称为高速缓存的压栈操作。位于 CHSLOT 之上的高速缓存位置被“ 锁定 ”，不能被处理器核修改，如图 16.3 所示。使用高速缓存出栈操作可以对被锁定的高速缓存位置解锁。通过向 CHPOP 位（CCH0LC.6）写‘ 1 ’来执行高速缓存出栈操作。当一个高速缓存出栈操作被启动后，CHSLOT 的值被加 1。该操作对最后被锁定的高速缓存位置解锁，但并不将数据从高速缓存中移出。注意，如果 CHSLOT 等于 111110b，则不应启动高速缓存出栈操作，否则可能对高速缓存性能有不利的影响。**重要提示：尽管硬件未明确禁止锁定高速缓存位置 1，但当 CHSLOT 等于 000000b 时，整个高速缓存被解锁。因此，高速缓存位置 1 和 0 在所有时间内都保持未锁定状态。**

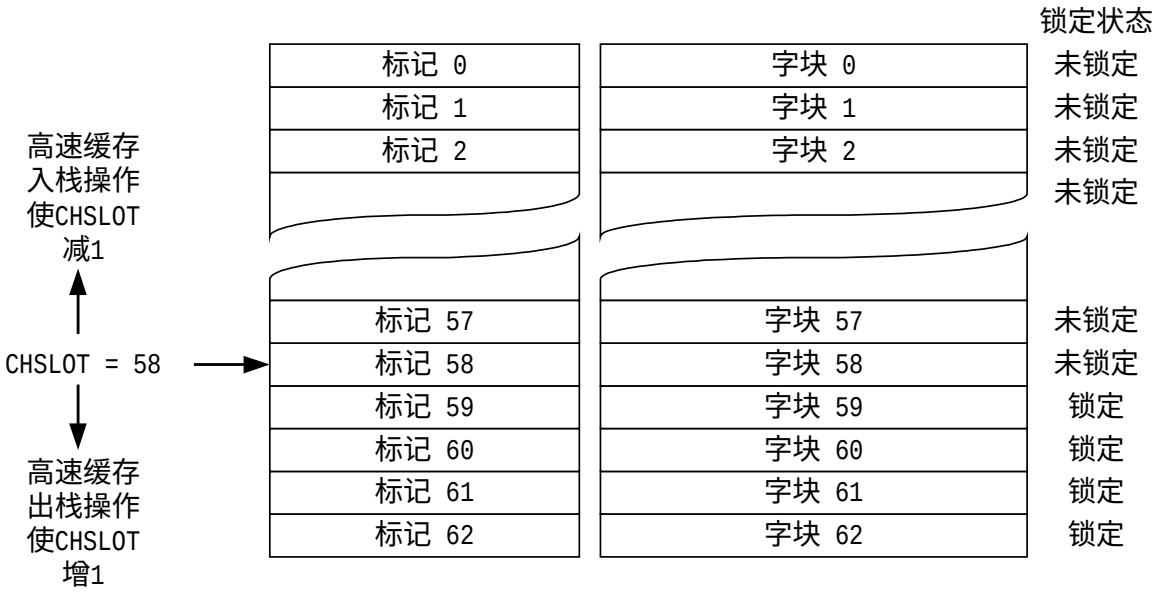


图 16.3　高速缓存锁定操作

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CHWREN	CHRDEN	CHPFEN	CHFLSH	CHRETI	CHISR	CHMOVC	CHBLKW	11100110
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xA1
SFR 页： F

位 7： CHWREN：高速缓存写操作使能位
该位使能处理器对高速缓存的写操作。
0：除了在 FLASH 写/擦除或高速缓存锁定期间，高速缓存内容不允许改变。
1：允许对高速缓存写入。

位 6： CHRDEN：高速缓存读操作使能位
该位使能处理器对高速缓存的读操作。
0：所有指令数据都取自 FLASH 存储器或预取引擎。
1：指令数据取自高速缓存（当可用时）。

位 5： CHPFEN：高速缓存预取使能位
该位使能预取引擎。
0：禁止预取引擎。
1：使能预取引擎。

位 4： CHFLSH：高速缓存清除位
向该位写‘ 1 ’时清除高速缓存内容。该位的读出值总是为‘ 0 ’。

位 3： CHRETI：高速缓存 RETI 目的地址使能位
该位控制 RETI 目的地址是否可以进入高速缓存。
0：RETI 指令的目的地址不能被缓存。
1：RETI 指令的目的地址可以被缓存。

位 2： CHISR：高速缓存 ISR 使能位。
该位控制中断服务程序（ISR）中的指令数据是否可以进入高速缓存。
0：ISR 中的指令不能进入高速缓存。
1：ISR 中的指令可以进入高速缓存。

位 1： CHMOVC：高速缓存 MOVC 使能位。
该位控制 MOVC 指令返回的数据是否可以进入高速缓存。
0：MOVC 指令返回的数据不能进入高速缓存。
1：MOVC 指令返回的数据可以进入高速缓存。

位 0： CHBLKW：块写使能位。
该位控制软件对 FLASH 存储器的块写操作。
0：软件 FLASH 写操作的每个字节都被单独写入。
1：FLASH 字节按 4 字节为一组（代码空间写）或 2 字节为一组（临时存储区写）写入。

图 16.4 CCH0CN：高速缓存控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CHMSCTL				CHALGM	CHFIXM	CHMSTH		00000100
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xA2
SFR 页： F

位 7-4： CHMSCTL：高速缓存未命中事件累加器（位 4~1）
这些位是高速缓存未命中事件累加器的位 4~1。要读这些位之前，它们必须被先锁存（通过读 CCH0MA 寄存器中的 CHMSCTH 位实现，见图 16.7）。

位 3： CHALGM：高速缓存替换算法选择位
该位高速缓存的替换算法。
0：高速缓存使用回弹算法。
1：高速缓存使用伪随机算法。

位 2： CHFIXM：高速缓存固定 MOVC 使能位。
该位强制 MOVC 数据写入到高速缓存的字块 0。
0：根据 CHALGM 位选择的当前算法写 MOVC 数据。
1：MOVC 数据总是被写到高速缓存的字块 0。

位 1-0： CHMSTH：高速缓存未命中阈值。
这两位决定何时未命中的指令数据进入高速缓存。
如果获得数据的时间多于 CHMSTH 个时钟，该数据将进入高速缓存。

图 16.5 CCH0TN：高速缓存调整寄存器

R/W	R/W	R	R	R	R	R	R	复位值
CHPUSH	CHPOP	CHSLOT						00111110
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xA3
SFR 页： F

位 7： CHPUSH：高速缓存压栈使能位
该位使能高速缓存的压栈操作，该操作用 MOVC 指令锁盯高速缓存字块中的信息。
0：禁止高速缓存的压栈操作。
1：允许高速缓存的压栈操作。当执行 MOVC 读指令时，含有要读取数据的 4 字节代码段被锁定在高速缓存中由 CHSLOT 指示的位置，然后 CHSLOT 减 1。
注意：一次锁定的高速缓存字块数不能大于 61，因为当 CHSLOT 等于 0 时整个高速缓存将被解锁。

位 6： CHPOP：高速缓存出栈
向该位写‘ 1 ’使 CHSLOT 加 1，然后解锁对应的字块。该位的读出值总是为‘ 0 ’。注意：当 CHSLOT 等于 111110b 时不应执行高速缓存出栈操作，出栈的字块数若大于入栈的字块数将会对高速缓存性能造成不可确定的影响。

位 5-0： CHSLOT：高速缓存字块指针
这些只读位是高速缓存锁定堆栈的指针。在 CHSLOT 之上的位置被锁定，并且不能被处理器改写，除非 CHSLOT 等于 0。

图 16.6 CCH0LC：高速缓存锁定控制寄存器

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CHMSOV	CHMSCTH							00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x9A
SFR 页： F

位 7： CHMSOV：高速缓存未命中溢出标志
该位指示高速缓存未命中累加器自上一次被写入后是否发生溢出。
0：自上一次被写入后，高速缓存未命中累加器未发生溢出。
1：自上一次被写入后，高速缓存未命中累加器发生了溢出。

位 6-0： CHMSCTH：高速缓存未命中事件累加器（位 11~5）
这些位是高速缓存未命中累加器的位 11~5。接下来的 4 位（位 4~1）保存在 CCH0TN 寄存器中的 CHMSCTL。
在因高速缓存未命中导致的处理器延时的每个时钟周期该累加器加 1。该功能主要用于优化代码执行速度的调试阶段。
写 CHMSCTH 时清除高速缓存未命中累加器的低 5 位。
读 CHMSCTH 时返回 CHMSCTH 的值，并将位 4~1 锁存到 CHMSCTL 以便读取。因为高速缓存未命中累加器的位 0 不可见，所以高速缓存未命中累加值等于 2*（CCHMSCTH:CCHMSCTL）。

图 16.7 CCH0MA：高速缓存不命中累加器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
							FLBUSY	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
SFR 页： F

位 7-1： 保留。

位 0： FLBUSY：FLASH 忙标志
该位指示是否正在进行 FLASH 写或擦除操作。
0：FLASH 处于空闲过读状态。
1：正在进行 FLASH 写或擦除操作。

图 16.8 FLSTAT：FLASH 状态

17．外部数据存储器接口和片内 XRAM

C8051F12x 和C8051F13x MCU内部有位于外部数据存储器空间的8K字节片上RAM（XRAM），还有可用于访问片外存储器和存储器映射 I/O 器件的外部数据存储器接口（EMIF）。外部存储器空间可以用外部传送指令（MOVX）和数据指针（DPTR）访问，或者通过使用 R0 或 R1 用间接寻址方式访问。如果 MOVX 指令使用一个 8 位地址操作数（例如 @R1），则 16 位地址的高字节由外部存储器接口控制寄存器（EMI0CN，如图 17.1 所示）提供。**注意：MOVX 指令还用于写 FLASH 存储器，详见“ 15. FLASH 存储器 ”。缺省情况下 MOVX 指令访问 XRAM。**EMIF 可被配置为使用低 I/O 端口（P0-P3）或高 I/O 端口（P4-P7）。

17.1 访问 XRAM

XRAM 存储器空间用 MOVX 指令访问。MOVX 指令有两种形式，这两种形式都使用间接寻址方式。第一种方法使用数据指针 DPTR，该 16 位寄存器中含有待读或写的 XRAM 单元的有效地址。第二种方法使用 R0 或 R1，与 EMI0CN 寄存器一起形成有效 XRAM 地址。下面举例说明这两种方法。

17.1.1 16 位 MOVX 示例

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

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

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

17.1.2 8 位 MOVX 示例

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

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

17.2 配置外部存储器接口

配置外部存储器接口的过程包括下面 6 个步骤：

1. 将 EMIF 选到低端口（P3、P2、P1 和 P0）或选到高端口（P7、P6、P5 和 P4）。
2. 配置端口引脚的输出方式为推挽或漏极开路（最常用的是推挽方式）。
3. 配置对应 EMIF 引脚的端口锁存器为休眠态（通常将它们设置为逻辑‘1’）。
4. 选择复用方式或非复用方式。
5. 选择存储器模式（只用片内存储器、不带块选择的分片方式、带块选择的分片方式或只用片外存储器）。
6. 设置与片外存储器或外设接口的时序。

下面将对上述 6 个步骤作出详细说明。端口选择、复用方式选择和存储器模式位都位于 EMI0CN 寄存器中，如图 17.2 所示。

17.3 端口选择和配置

外部存储器接口可以位于端口 3、2、1 和 0（所有器件）或端口 7、6、5 和 4（只限于 100 脚 TQFP 封装器件），由 PRTSEL 位（EMI0CF.5）的状态决定。如果选择低端口，则 EMIFLE 位（XBR2.1）必须被置‘1’，以使交叉开关跳过 P0.7(W/R)、P0.6(R/D)和 P0.5(ALE，如果选择复用方式)。有关配置交叉开关的详细信息见“18.1 端口 0 ~ 3 和优先权交叉开关译码器”。

外部存储器接口只在执行片外 MOVX 指令期间使用相关的端口引脚。一旦 MOVX 指令执行完毕，端口锁存器或交叉开关又重新恢复对端口引脚的控制（端口 3、2、1 和 0），有关交叉开关和端口操作及控制的详细信息见“18. 端口输入/输出”。**端口锁存器应被明确地配置为使外部存储器接口引脚处于休眠状态，通常是通过将它们设置为逻辑‘1’。**

在执行 MOVX 指令期间，外部存储器接口将禁止所有作为输入的那些引脚的驱动器（例如，读操作期间的 Data[7:0]）。端口引脚的输出方式（无论引脚被配置为漏极开路或是推挽方式）不受外部存储器接口操作的影响，始终受 PnMDOUT 寄存器的控制。在大多数情况下，所有 EMIF 引脚的输出方式都应被配置为推挽方式。有关端口输出方式配置的详细信息见“18.2.2 配置端口引脚的输出方式”。

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
PGSEL7	PGSEL6	PGSEL5	PGSEL4	PGSEL3	PGSEL2	PGSEL1	PGSEL0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xA2
SFR 页： 0

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

图 17.1 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
-	-	PRTSEL	EMD2	EMD1	EMD0	EALE1	EALE0	00000011
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xA3
SFR 页： 0

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

图 17.2 EMI0CF：外部存储器接口配置

17.4 复用和非复用选择

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

17.4.1 复用方式配置

在复用方式，数据总线和地址总线的低 8 位共享相同的端口引脚：AD[7:0]。在该方式下，要用一个外部锁存器（74HC373 或相同功能的逻辑门）保持 RAM 地址的低 8 位。外部锁存器由 ALE（地址锁存使能）信号控制，ALE 信号由外部存储器接口逻辑驱动。图 17.3 给出了复用方式配置的一个例子。

在复用方式，可以根据 ALE 信号的状态将外部 MOVX 操作分成两个阶段。在第一个阶段，ALE 为高电平，地址总线的低 8 位出现在 AD[7:0]。在该阶段，地址锁存器的‘ Q ’输出与‘ D ’输入的状态相同。ALE 由高变低时标志第二阶段开始，地址锁存器的输出保持不变，即与锁存器的输入无关。在第二阶段稍后，当/RD 或/WR 有效时，数据总线控制 AD[7:0]端口的状态。更详细的信息见“ 17.6.2 复用方式 ”。

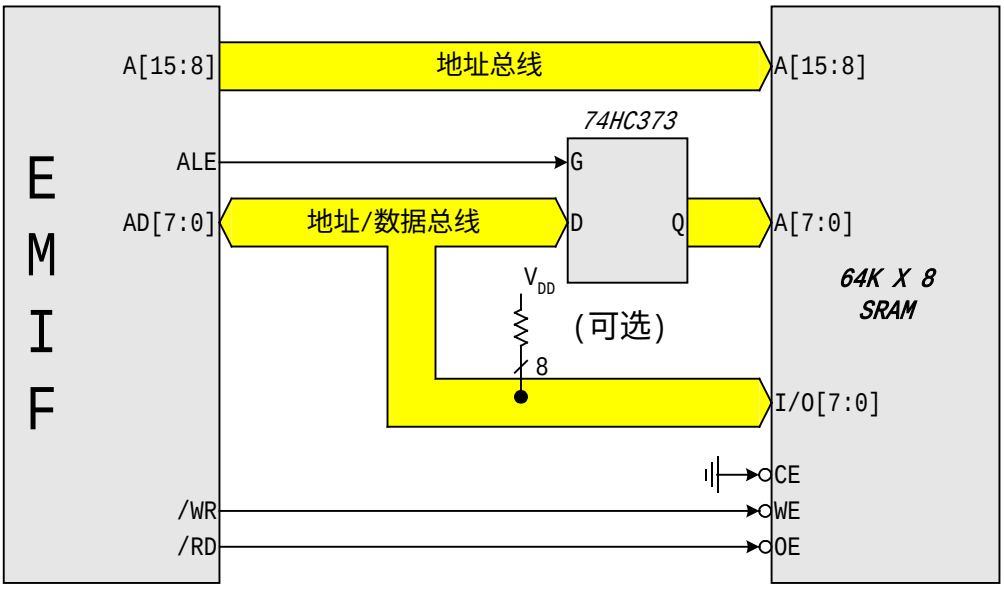


图 17.3 复用方式配置示例

17.4.2 非复用方式配置

在非复用方式，数据总线和地址总线是分开的。图 17.4 给出了非复用方式配置的一个例子。关于非复用方式操作的更详细的信息见“ 17.6.1 非复用方式 ”。

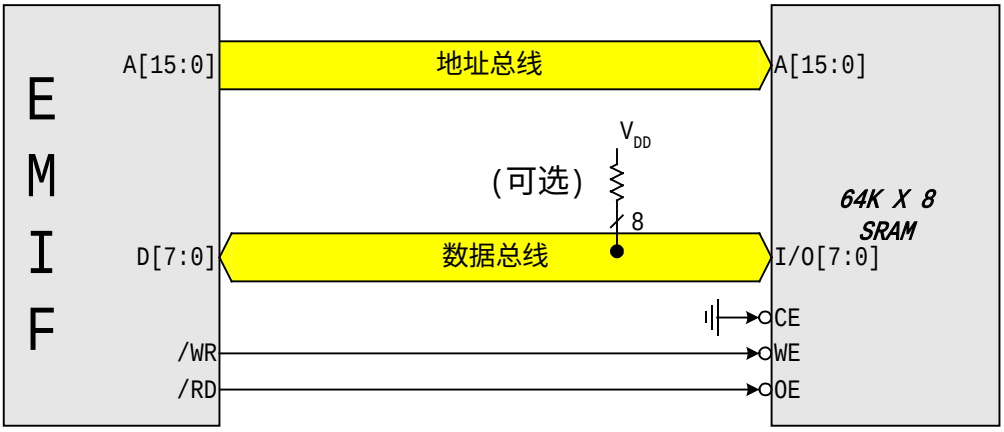


图 17.4 非复用方式配置示例

17.5 存储器模式选择

可以用 EMI0CF 寄存器（图 17.2）中 EMIF 模式选择位将外部数据存储器空间配置为图 17.5 所示的四种工作模式之一。下面简要介绍这些模式。有关不同模式的更详细信息见“ 17.6 EMIF 时序 ”。

17.5.1 只用内部 XRAM

当 EMI0CF.[3:2]被设置为‘ 00 ’时，所有 MOVX 指令都将访问器件内部的 XRAM 空间。存储器寻址的地址大于实际地址空间时将以 8K 为边界回绕。例如：地址 0x2000 和 0x4000 都指向片内 XRAM 空间的 0x0000 地址。

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

17.5.2 无块选择的分片模式

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

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

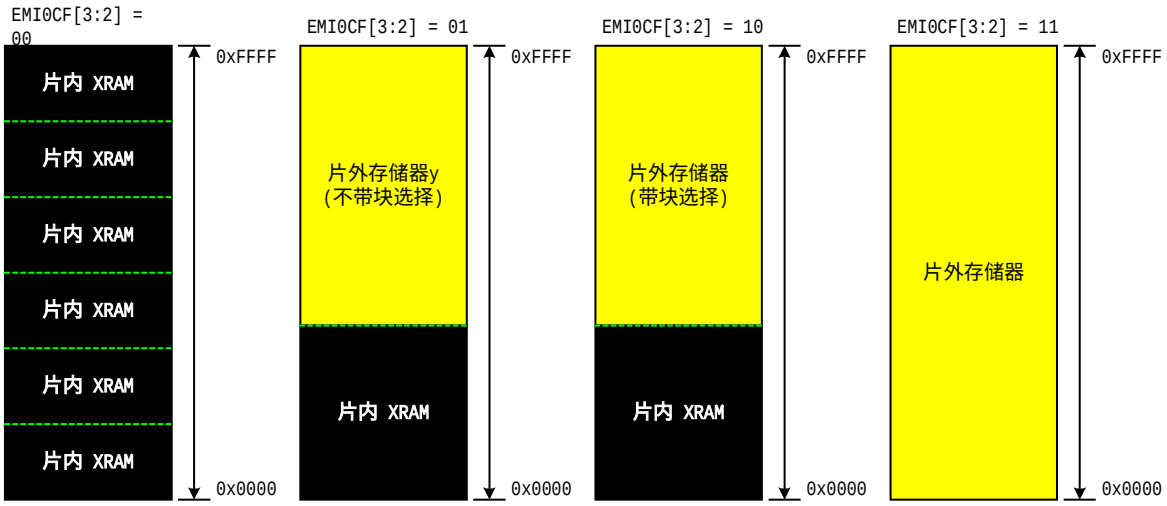


图 17.5 EMIF 工作模式

17.5.3 带块选择的分片模式

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

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

17.5.4 只用外部存储器

当 EMI0CF.[3:2]被设置为‘ 11 ’时，所有 MOVX 指令都将访问器件外部 XRAM 空间。片内 XRAM 对 CPU 为不可见。该方式在访问从 0x0000 开始的 8K 片外存储器时有用。

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

17.6 时序

外部存储器接口的时序参数是可编程的，这就允许连接具有不同建立时间和保持时间要求的器件。地址建立时间、地址保持时间、/RD 和/WR 选通脉冲的宽度以及复用方式下 ALE 脉冲的宽度都可以通过 EM0TC（见图 17.6）和 EMI0CF[1:0]编程，编程单位为 SYSCLK 周期。

片外 MOVX 指令的时序可以通过将 EMI0TC 寄存器中定义的时序参数加上 4 个 SYSCLK 周期来计算。在非复用方式，一次片外 XRAM 操作的最小执行时间为 5 个 SYSCLK 周期（用于/RD 或/WR 脉冲的 1 个 SYSCLK + 4 个 SYSCLK）。对于复用方式，地址锁存使能信号至少需要 2 个附加的 SYSCLK 周期。因此，在复用方式，一次片外 XRAM 操作的最小执行时间为 7 个 SYSCLK 周期（用于 ALE 的 2 个 SYSCLK + 用于/RD 或/WR 脉冲的 1 个 SYSCLK + 4 个 SYSCLK）。在器件复位后，可编程建立和保持时间的缺省值为最大延迟设置。

表 17.1 列出了外部存储器接口的 AC 参数，图 17.7 到图 17.12 给出了对应不同外部存储器接口模式和 MOVX 操作的时序图。

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
EAS1	EAS0	EWR3	EWR2	EWR1	EWR0	EAH1	EAH0	11111111
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xA1
SFR 页： 0

位 7-6： EAS1-0：EMIF 地址建立时间位。
00：地址建立时间 = 0 个 SYSCLK 周期。
01：地址建立时间 = 1 个 SYSCLK 周期。
10：地址建立时间 = 2 个 SYSCLK 周期。
11：地址建立时间 = 3 个 SYSCLK 周期。

位 5-2： EWR3-0：EMIF /WR 和/RD 脉冲宽度控制位。
0000：/WR 和/RD 脉冲宽度 = 1 个 SYSCLK 周期。
0001：/WR 和/RD 脉冲宽度 = 2 个 SYSCLK 周期。
0010：/WR 和/RD 脉冲宽度 = 3 个 SYSCLK 周期。
0011：/WR 和/RD 脉冲宽度 = 4 个 SYSCLK 周期。
0100：/WR 和/RD 脉冲宽度 = 5 个 SYSCLK 周期。
0101：/WR 和/RD 脉冲宽度 = 6 个 SYSCLK 周期。
0110：/WR 和/RD 脉冲宽度 = 7 个 SYSCLK 周期。
0111：/WR 和/RD 脉冲宽度 = 8 个 SYSCLK 周期。
1000：/WR 和/RD 脉冲宽度 = 9 个 SYSCLK 周期。
1001：/WR 和/RD 脉冲宽度 = 10 个 SYSCLK 周期。
1010：/WR 和/RD 脉冲宽度 = 11 个 SYSCLK 周期。
1011：/WR 和/RD 脉冲宽度 = 12 个 SYSCLK 周期。
1100：/WR 和/RD 脉冲宽度 = 13 个 SYSCLK 周期。
1101：/WR 和/RD 脉冲宽度 = 14 个 SYSCLK 周期。
1110：/WR 和/RD 脉冲宽度 = 15 个 SYSCLK 周期。
1111：/WR 和/RD 脉冲宽度 = 16 个 SYSCLK 周期。

位 1-0： EAH1-0：EMIF 地址保持时间位。
00：地址保持时间 = 0 个 SYSCLK 周期。
01：地址保持时间 = 1 个 SYSCLK 周期。
10：地址保持时间 = 2 个 SYSCLK 周期。
11：地址保持时间 = 3 个 SYSCLK 周期。

图 17.6 EMI0TC：外部存储器时序控制

17.6.1 非复用方式

17.6.1.1 16 位 MOVX：EMI0CF[4:2] = ‘101’, ‘110’, ‘111’

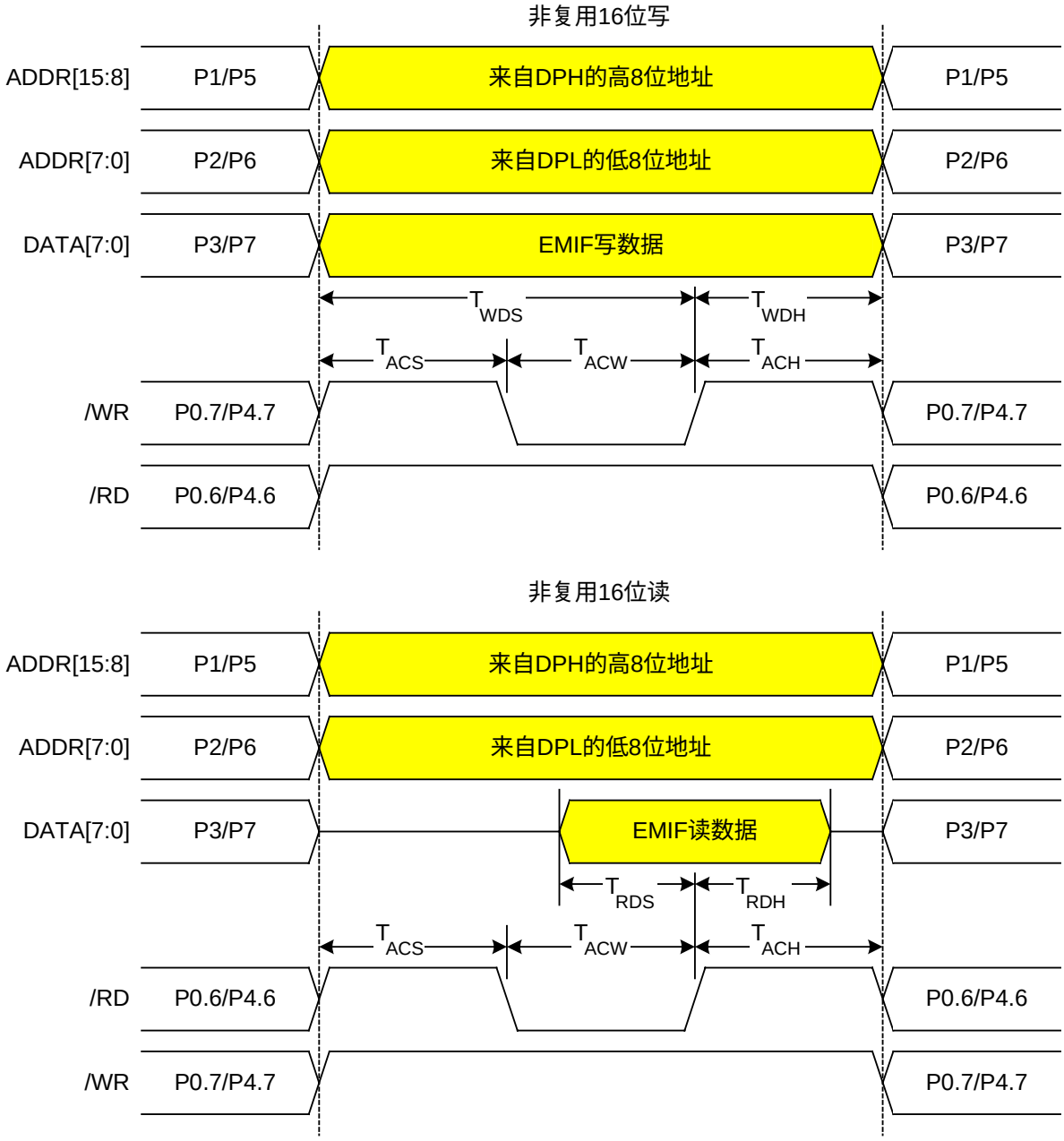


图 17.7 非复用 16 位 MOVX 时序

17.6.1.2 无块选择的 8 位 MOVX：EMI0CF[4:2] = ‘101’或‘111’

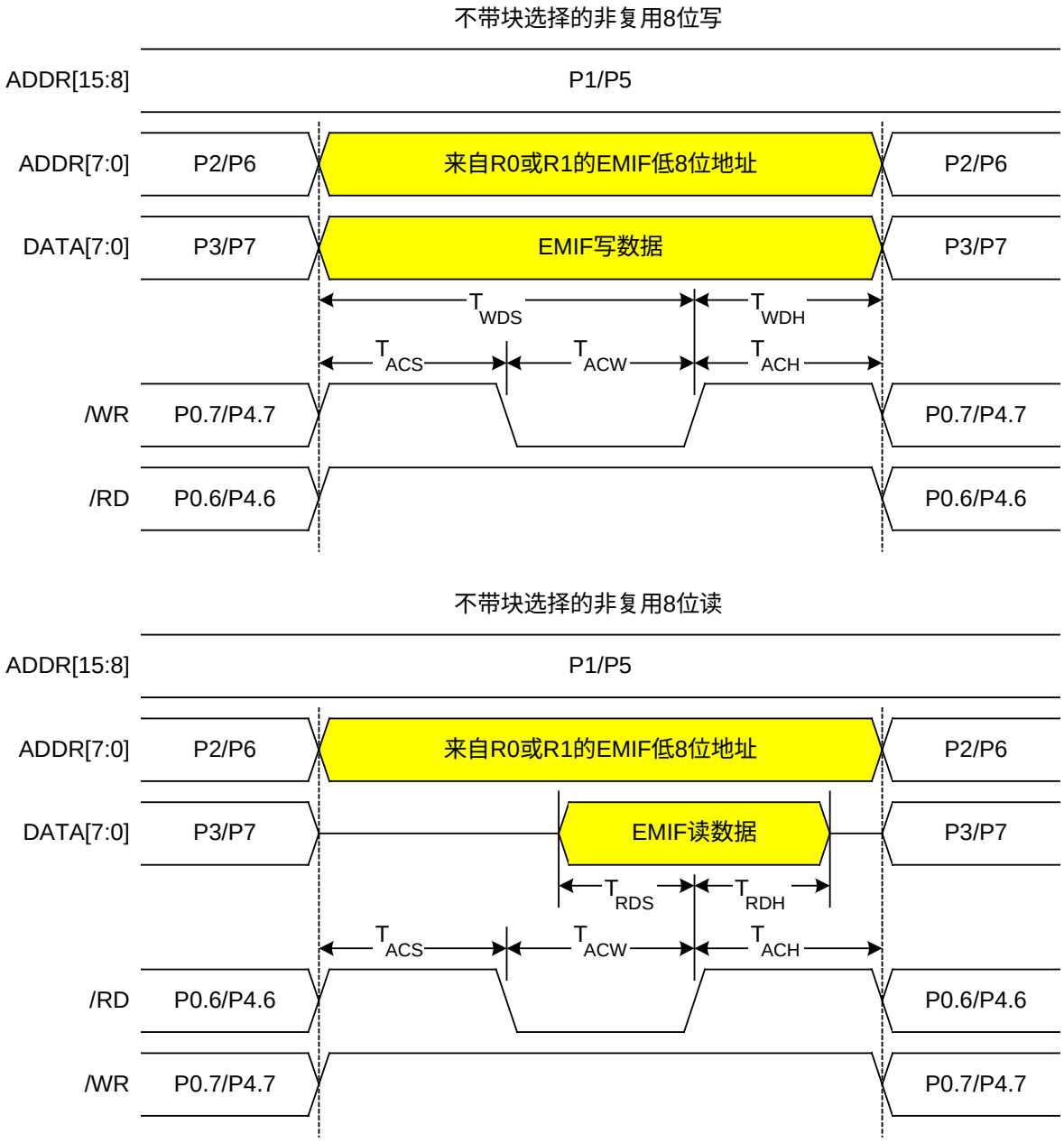


图 17.8 无块选择的非复用 8 位 MOVX 时序

17.6.1.3 带块选择的 8 位 MOVX：EMI0CF[4:2] = ‘110’

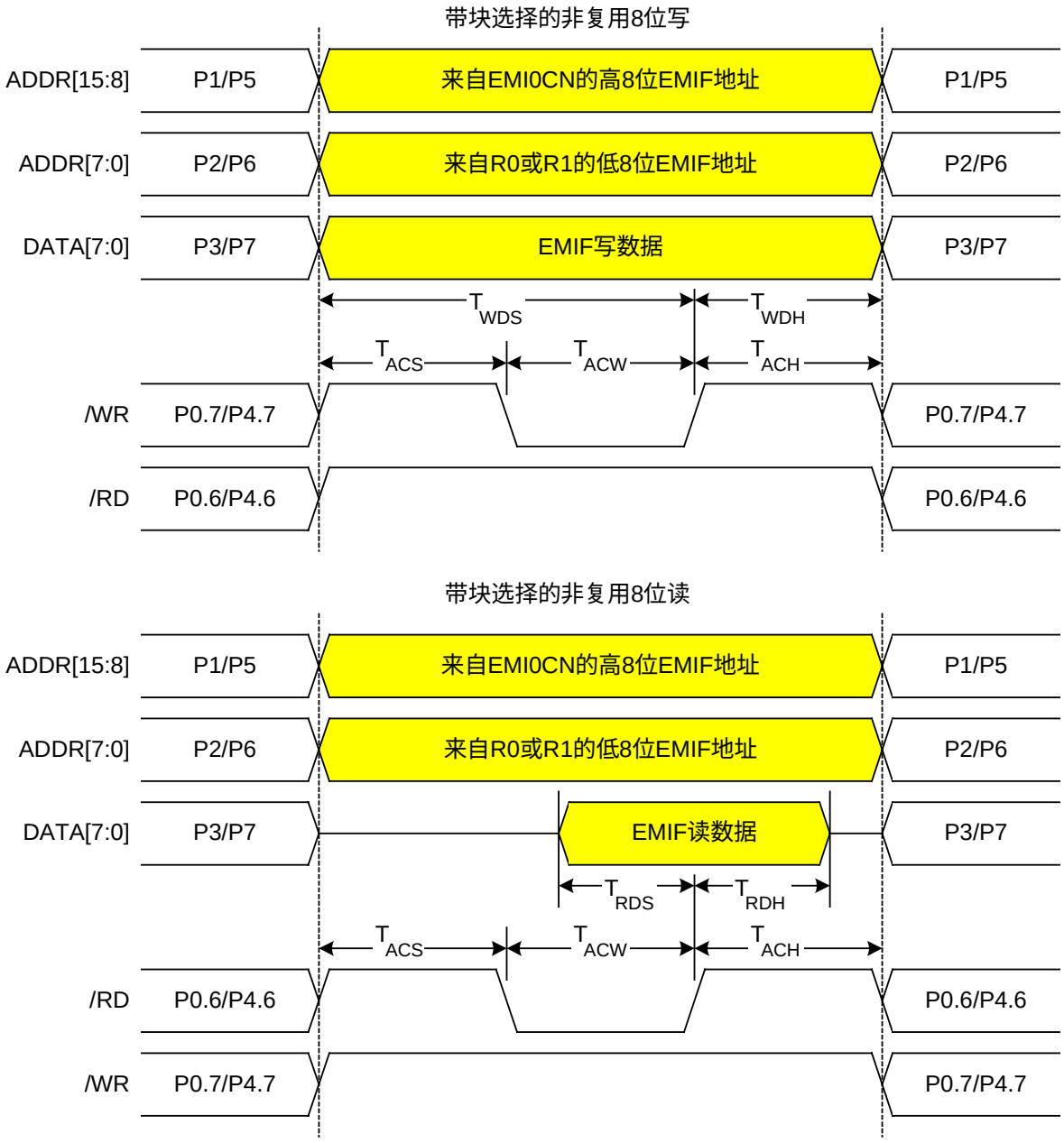


图 17.9 带块选择的非复用 8 位 MOVX 时序

17.6.2 复用方式

17.6.2.1 16 位 MOVX：EMI0CF[4:2] = ‘ 001 ’, ‘ 010 ’, ‘ 011 ’

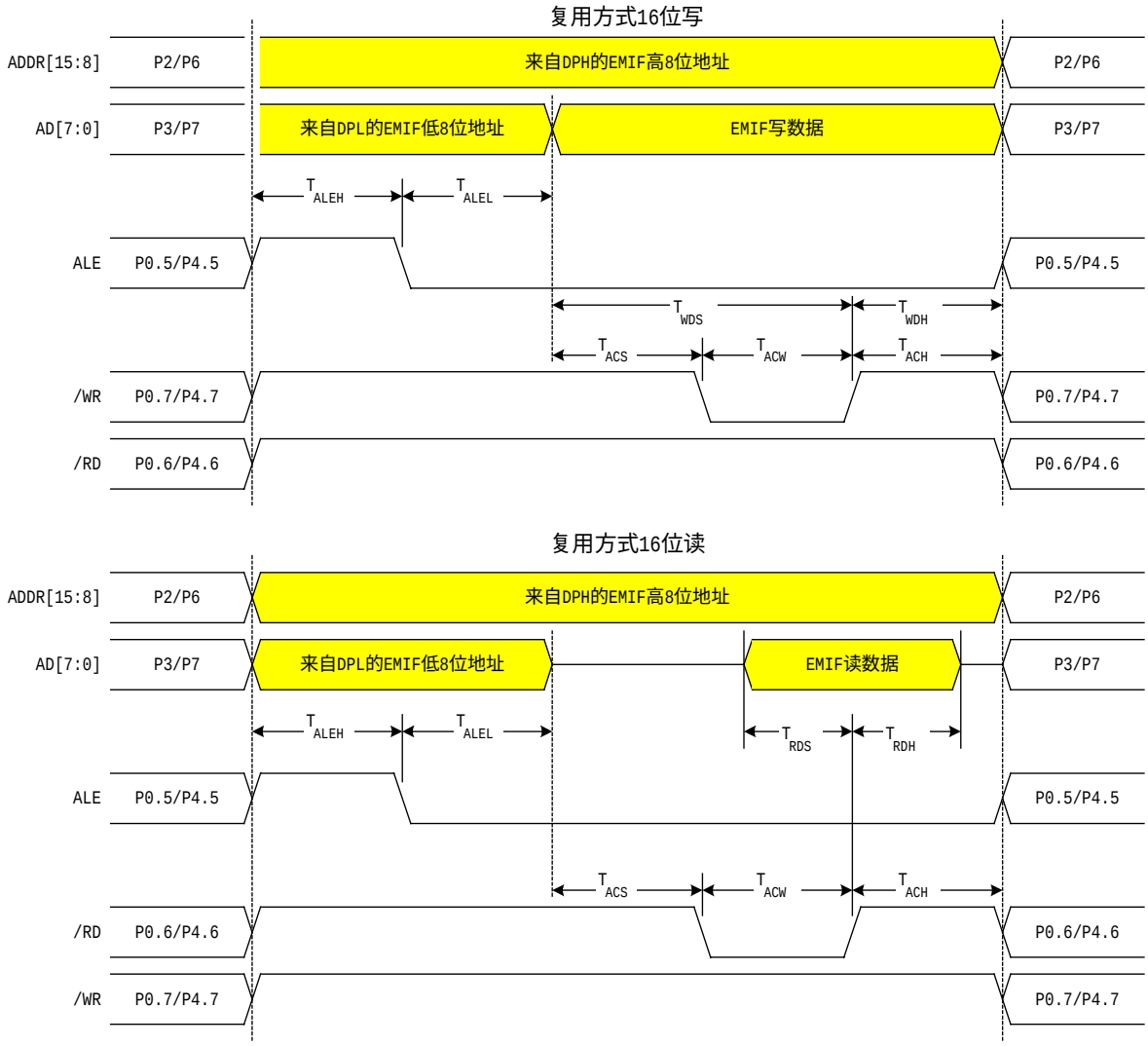


图 17.10 复用方式 16 位 MOVX 时序

17.6.2.2 无块选择的 8 位 MOVX：EMI0CF[4:2] = ‘ 001 ’ 或 ‘ 011 ’

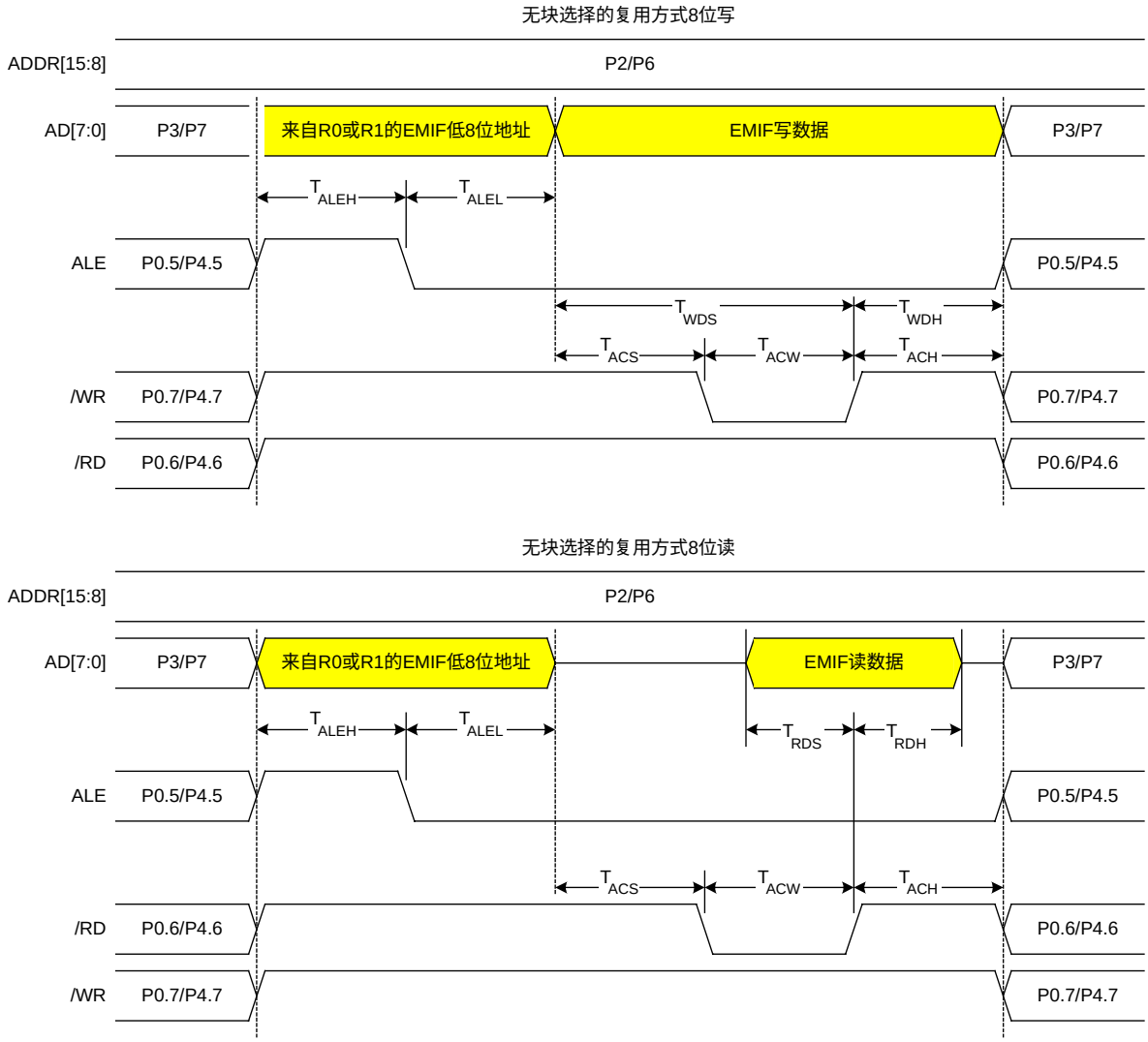


图 17.11 无块选择的复用方式 8 位 MOVX 时序

17.6.2.3 带块选择的 8 位 MOVX：EMI0CF[4:2] = ‘ 010 ’

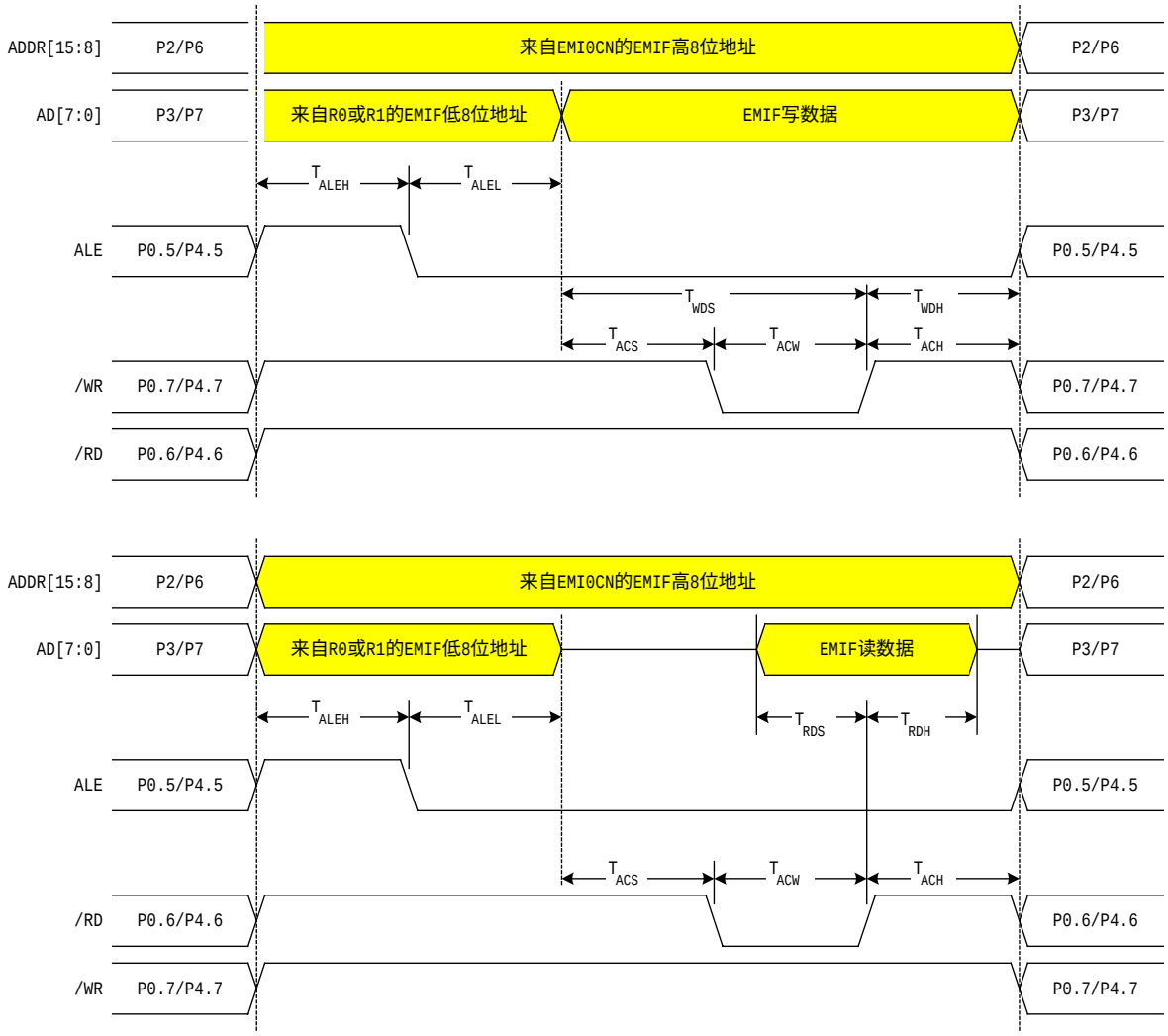


图 17.12 带块选择的复用方式 8 位 MOVX 时序

表 17.1 外部存储器接口的 AC 参数

参 数	说 明	最小值	最大值	单 位
T_{ACS}	地址/控制信号建立时间	0	3* T_{SYSCLK}	ns
T_{ACW}	地址/控制信号脉冲宽度	1* T_{SYSCLK}	16* T_{SYSCLK}	ns
T_{ACH}	地址/控制信号保持时间	0	3* T_{SYSCLK}	ns
T_{ALEH}	地址锁存使能信号高电平时间	1* T_{SYSCLK}	4* T_{SYSCLK}	ns
T_{ALEL}	地址锁存使能信号低电平时间	1* T_{SYSCLK}	4* T_{SYSCLK}	ns
T_{WDS}	写数据建立时间	1* T_{SYSCLK}	19* T_{SYSCLK}	ns
T_{WDH}	写数据保持时间	0	3* T_{SYSCLK}	ns
T_{RDS}	读数据建立时间	20		ns
T_{RDH}	读数据保持时间	0		ns
注：T_{SYSCLK} 等于器件系统时钟（SYSCLK）的一个周期。				

18. 端口输入/输出

C8051F12x 和 C8051F13x 系列器件有按 8 位端口组织的 64 个数字 I/O 引脚（100 脚 TQFP 封装）或 32 个数字 I/O 引脚（64 脚 TQFP 封装）。所有端口都可以通过对应的端口数据寄存器按位寻址和按字节寻址。所有端口引脚都耐 5V 电压，都可以被配置为漏极开路或推挽输出方式和弱上拉。端口 I/O 单元的原理框图示于图 18.1。表 18.1 给出了端口 I/O 引脚的电气特性。

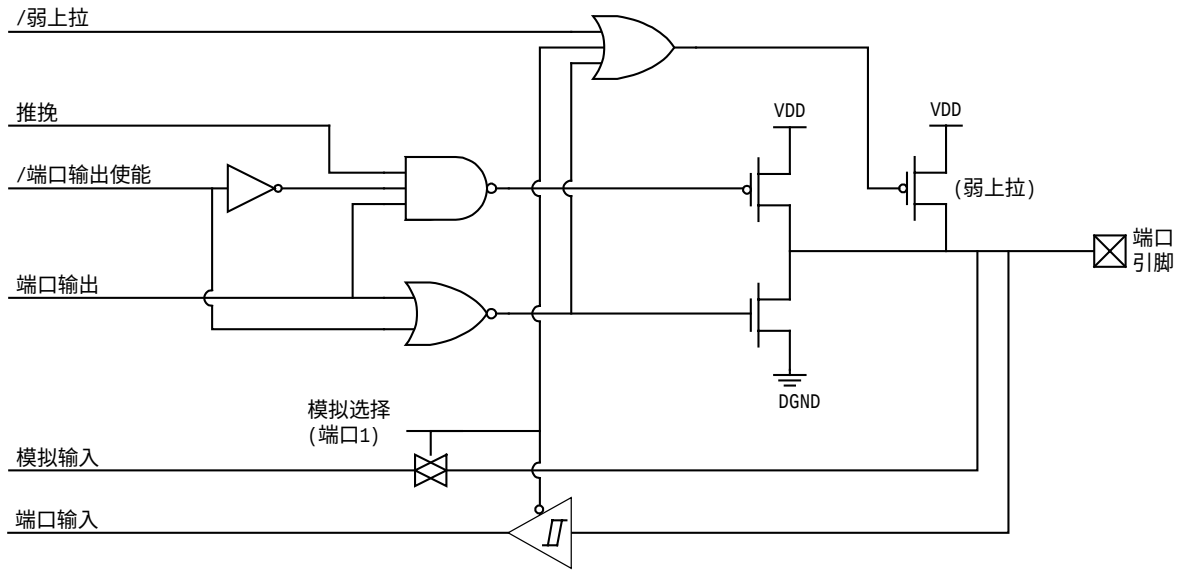


图 18.1 端口 I/O 单元功能框图

表 18.1 端口 I/O 直流电气特性

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（V_{OH}）	I_{OH} = -10μA，端口 I/O 为推挽方式 I_{OH} = -3mA，端口 I/O 为推挽方式 I_{OH} = -10mA，端口 I/O 为推挽方式	VDD-0.1 VDD-0.7	 VDD-0.8		V
输出低电压（V_{OL}）	I_{OL} = 10μA I_{OL} = 8.5mA I_{OL} = 25mA		 1.0	0.1 0.6	V
输入高电压（V_{IH}）		0.7×VDD			V
输入低电压（V_{IL}）				0.3×VDD	V
输入漏电流	DGND < 端口引脚 < VDD，高阻态 弱上拉禁止 弱上拉使能		 10	±1	μA
输入电容			5		pF

大量的数字资源需要通过 4 个低端 I/O 端口 P0、P1、P2 和 P3 才能使用。P0、P1、P2 和 P3 中的每个引脚既可定义为通用的端口 I/O（GPIO）引脚，又可以分配给一个数字外设或功能（例如：UART0 或/INT1），如图 18.2 所示。系统设计者控制数字功能的引脚分配，只受可用引脚数的限制。这种资源分配的灵活性是通过使用优先权交叉开关译码器实现的。注意，不管引脚被分配给一个数字外设或是作为通用 I/O，总是可以通过读相应的数据寄存器得到端口 I/O 引脚的状态。端口 1 的引脚可以用做 ADC2 的模拟输入。

在执行目标地址为片外 XRAM 的 MOVX 指令时，外部存储器接口可以在低端口或高端口有效。有关外部存储器接口的详细信息见“ 17. 外部数据存储器接口和片内 XRAM ”。

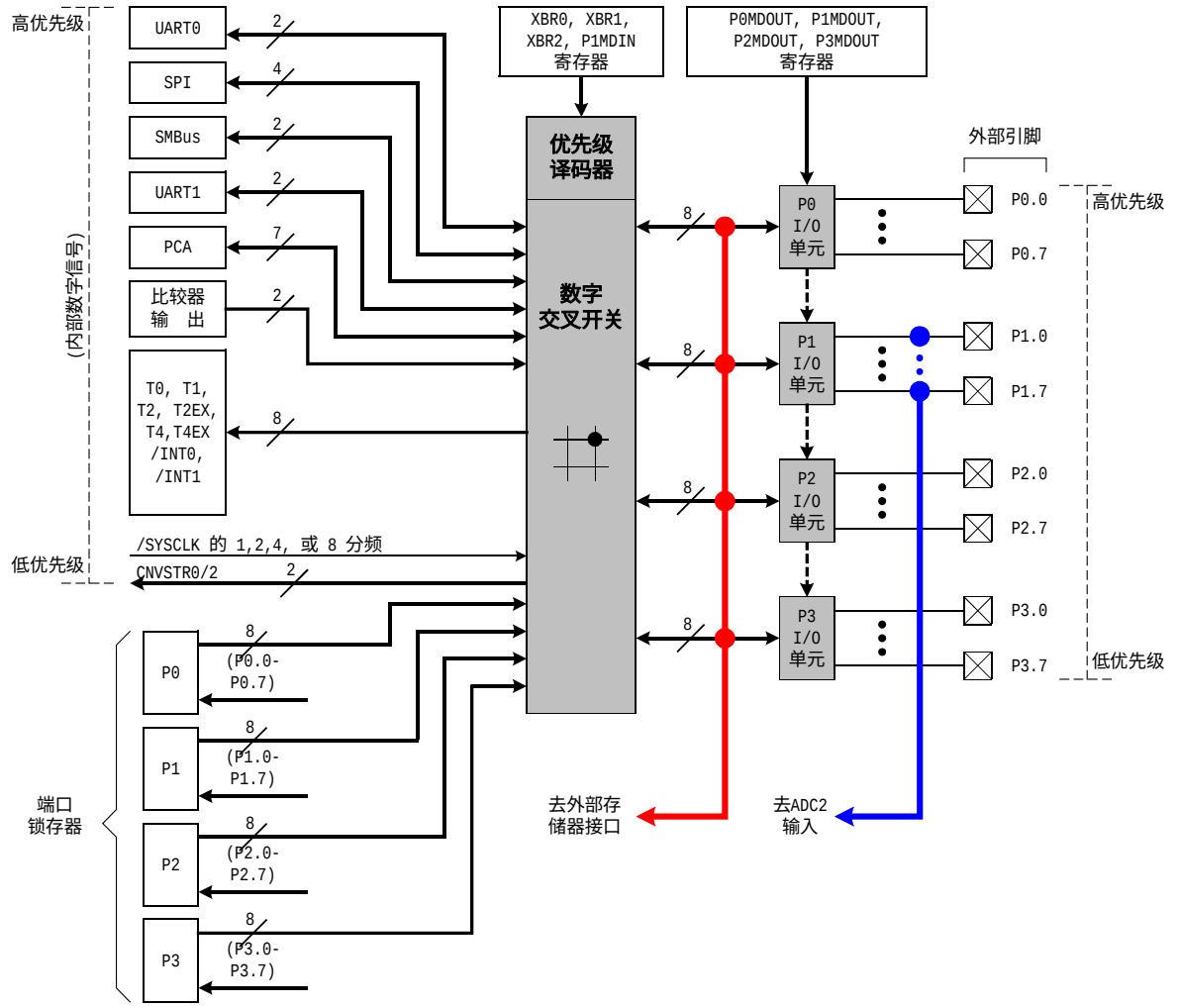


图 18.2 端口 I/O 功能框图

18.1 端口 0 – 3 和优先权交叉开关译码器

优先权交叉开关译码器，或称为“交叉开关”，按优先权顺序将端口 0 – 3 的引脚分配给器件上的数字外设（UART、SMBus、PCA、定时器等）。端口引脚的分配顺序是从 P0.0 开始，可以一直分配到 P3.7。为数字外设分配端口引脚的优先权顺序列于图 18.3，UART0 具有最高优先权，而 CNVSTR2 具有最低优先权。

18.1.1 交叉开关引脚分配

当交叉开关配置寄存器 XBR0、XBR1 和 XBR2 中外设的对应使能位被设置为逻辑‘1’时，交叉开关将端口引脚分配给外设，如图 18.7、图 18.8 和图 18.9 所示。例如，如果 UART0EN 位（XBR0.2）被设置为逻辑‘1’，则 TX0 和 RX0 引脚将分别被分配到 P0.0 和 P0.1。因为 UART0 有最高优先权，所以当 UART0EN 位被设置为逻辑‘1’时其引脚将总是被分配到 P0.0 和 P0.1。如果一个数字外设的使能位未被设置为逻辑‘1’，则其端口将不能通过器件的端口引脚被访问。注意：当选择了串行通信外设（即 SMBus、SPI 或 UART）时，交叉开关将为所有相关功能分配引脚。例如，不能为 UART0 功能只分配 TX0 引脚而不分配 RX0 引脚。被使能的外设的每种组合导致唯一的器件引脚分配。

（EMIFLE = 0；P1MDIN = 0xFF）

	P0								P1								P2								P3								交叉开关寄存器位
引脚 I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
TX0	●																																UART0EN:XBR0.2
RX0		●																															
SCK	●		●																														SPI0EN:XBR0.1
MISO		●		●																													
MOSI			●		●																												
NSS				●		●	当 SPI 被设置为 3 线方式时，NSS 不被分配端口引脚																												
SDA	●		●	●	●	●	●																										SMB0EN:XBR0.0
SCL		●		●	●	●	●																										
TX1	●		●	●	●	●	●	●																									UART1EN:XBR2.2
RX1		●		●	●	●	●	●																									
CEX0	●		●	●	●	●	●	●																									PCA0ME:XBR0.[5:3]
CEX1		●		●	●	●	●	●																									
CEX2			●		●	●	●	●																									
CEX3				●		●	●	●																									
CEX4					●		●	●																									
CEX5						●		●																									
ECI	●	●	●	●	●	●	●	●									●																ECI0E:XBR0.6
CP0	●	●	●	●	●	●	●	●									●	●															CP0E:XBR0.7
CP1	●	●	●	●	●	●	●	●									●	●	●														CP1E:XBR1.0
T0	●	●	●	●	●	●	●	●									●	●	●	●													T0E:XBR1.1
/INT0	●	●	●	●	●	●	●	●									●	●	●	●	●												INT0E:XBR1.2
T1	●	●	●	●	●	●	●	●									●	●	●	●	●	●											T1E:XBR1.3
/INT1	●	●	●	●	●	●	●	●									●	●	●	●	●	●	●										INT1E:XBR1.4
T2	●	●	●	●	●	●	●	●									●	●	●	●	●	●	●	●									T2E:XBR1.5
T2EX	●	●	●	●	●	●	●	●									●	●	●	●	●	●	●	●	●								T2EXE:XBR1.6
T4	●	●	●	●	●	●	●	●									●	●	●	●	●	●	●	●	●	●							T4E:XBR2.3
T4EX	●	●	●	●	●	●	●	●									●	●	●	●	●	●	●	●	●	●	●						T4EXE:XBR2.4
/SYSCLK	●	●	●	●	●	●	●	●									●	●	●	●	●	●	●	●	●	●	●	●					SYSCKE:XBR1.7
CNVSTR0	●	●	●	●	●	●	●	●									●	●	●	●	●	●	●	●	●	●	●	●	●				CNVSTE0:XBR2.0
CNVSTR2	●	●	●	●	●	●	●	●									●	●	●	●	●	●	●	●	●	●	●	●	●	●			CNVSTE2:XBR2.5
						ALE	/RD	/WR	AIN2.0/A8	AIN2.1/A9	AIN2.2/A10	AIN2.3/A11	AIN2.4/A12	AIN2.5/A13	AIN2.6/A14	AIN2.7/A15	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7	
									AIN2 输入/非复用地址高								复用地址高/非复用地址低								复用数据/非复用数据								

图 18.3 优先权交叉开关译码表

端口 0 – 3 中所有未被交叉开关分配的引脚都可以作为通用 I/O（GPI/O）引脚，通过读或写相应的端口数据寄存器（见图 18.10、图 18.12、图 18.15 和图 18.17）访问，这是一组既可以按位寻址也可以按字节寻址的 SFR。被交叉开关分配的那些端口引脚的输出状态受使用这些引脚的数字外设的控制。向端口数据寄存器（或相应的端口位）写入时对这些引脚的状态没有影响。

不管交叉开关是否将引脚分配给外设，读一个端口数据寄存器（或端口位）将总是返回引脚本身的逻辑状态。唯一的例外发生在执行读-修改-写指令（ANL、ORL、XRL、CPL、INC、DEC、DJNZ、JBC、CLR、SET 和位传送操作）期间。在读-修改-写指令的读周期，所读的值是端口数据寄存器的内容，而不是端口引脚本身的状态。

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

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

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

18.1.2 配置端口引脚的输出方式

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

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

端口 0-3 引脚的输出方式由 PnMDOUT 寄存器中的对应位决定（见图 18.11、图 18.14、图 18.16 和图 18.18）。例如，P3MDOUT.7 为逻辑‘ 1 ’时将 P3.7 配置为推挽方式；P3MDOUT.7 为逻辑‘ 0 ’时将 P3.7 配置为漏极开路方式。所有端口引脚的缺省方式均为漏极开路。

不管交叉开关是否将一个端口引脚分配给某个数字外设，端口引脚的输出方式都受 PnMDOUT 寄存器控制。例外情况是：连接到 SDA、SCL、RX0（如果 UART0 工作于方式 0）、RX1（如果 UART1 工作于方式 0）的端口引脚总是被配置为漏极开路输出，而与 PnMDOUT 寄存器中的对应位的设置值无关。

18.1.3 配置端口引脚为数字输入

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

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

18.1.4 弱上拉

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

18.1.5 配置端口 1 的引脚为模拟输入（AIN.[7:0]）

对于 C8051F12x 器件，端口 1 的引脚可以用作 ADC2 模拟多路开关的模拟输入。通过向 P1MDIN 寄存器（见图 18.13）中的对应位写‘0’即可将端口引脚配置为模拟输入。缺省情况下所有端口引脚均为数字输入方式。将一个端口引脚配置为模拟输入的过程如下：

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

注意：被配置为模拟输入的引脚的输出驱动器并没有被明确地禁止。因此被配置为模拟输入的引脚所对应的 P1MDOUT 位应被明确地设置为逻辑‘0’(漏极开路方式)，对应的端口数据位应被设置为逻辑‘1’(高阻态)。还需要注意的是，将一个端口引脚用作 ADC2 模拟多路开关的输入时并不要求将其配置为模拟输入，但强烈建议这样做。有关 ADC2 的更详细信息见“7. ADC2(8 位 ADC)”。

18.1.6 外部存储器接口引脚分配

如果外部存储器接口（EMIF）被设置在低端口（端口 0-3），EMIFLE（XBR2.1）位应被设置为逻辑‘ 1 ’，以使交叉开关不将 P0.7 (/WR)、P0.6 (/RD)和 P0.5 (/ALE)（如果外部存储器接口使用复用方式）分配给外设。图 18.4 给出了 EMIFLE=1 并且 EMIF 工作在复用方式时的交叉开关译码表的示例。图 18.5 给出了 EMIFLE=1 并且 EMIF 工作在非复用方式时的交叉开关译码表的示例。

如果外部存储器接口被设置在低端口并且发生一次片外 MOVX 操作，则在该 MOVX 指令执行期间外部存储器接口将控制有关端口引脚的输出状态，而不管交叉开关寄存器和端口数据寄存器的设置如何。端口引脚的输出配置不受 EMIF 操作的影响，但读操作将禁止数据总线上的输出驱动器。有关外部存储器接口的详细信息见“ 17. 外部数据存储器接口和片内 XRAM ”。

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

	P0								P1								P2								P3								交叉开关寄存器位
引脚 I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
TX0	●																																UART0EN:XBR0.2
RX0		●																															
SCK	●		●																														SPI0EN:XBR0.1
MISO		●		●																													
MOSI			●		●																												
NSS				●					● 当 SPI 被设置为 3 线方式时，NSS 不被分配端口引脚																								
SDA	●		●	●	●				●	●																							SMB0EN:XBR0.0
SCL		●		●	●				●	●	●																						
TX1	●		●	●	●				●	●	●	●																					UART1EN:XBR2.2
RX1		●		●	●				●	●	●	●	●																				
CEX0	●		●	●	●				●	●	●	●	●	●																			PCA0ME:XBR0.[5:3]
CEX1		●		●	●				●	●	●	●	●	●	●																		
CEX2			●		●				●	●	●	●	●	●	●	●																	
CEX3				●					●	●	●	●	●	●	●	●	●																
CEX4					●					●	●	●	●	●	●	●		●															
CEX5									●		●	●	●	●	●	●	●	●	●														
ECI	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●													ECI0E:XBR0.6
CP0	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●												CP0E:XBR0.7
CP1	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●											CP1E:XBR1.0
T0	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										T0E:XBR1.1
/INT0	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									INT0E:XBR1.2
T1	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								T1E:XBR1.3
/INT1	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							INT1E:XBR1.4
T2	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						T2E:XBR1.5
T2EX	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					T2EXE:XBR1.6
T4	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				T4E:XBR2.3
T4EX	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			T4EXE:XBR2.4
/SYSCLK	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		SYSCKE:XBR1.7
CNVSTR0	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	CNVSTE0:XBR2.0
CNVSTR2	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	CNVSTE2:XBR2.5
						ALE	/RD	/WR	AIN2.0/A8	AIN2.1/A9	AIN2.2/A10	AIN2.3/A11	AIN2.4/A12	AIN2.5/A13	AIN2.6/A14	AIN2.7/A15	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7	
									AIN2 输入/非复用地址高								复用地址高/非复用地址低								复用数据/非复用数据								

图 18.4 优先权交叉开关译码表

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

	P0								P1								P2								P3								交叉开关寄存器位
引脚 I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	
TX0	●																																UART0EN:XBR0.2
RX0		●																															
SCK	●		●																														SPI0EN:XBR0.1
MISO		●		●																													
MOSI			●		●																												
NSS				●		●			当 SPI 被设置为 3 线方式时，NSS 不被分配端口引脚																								
SDA	●		●	●	●	●			●																								SMB0EN:XBR0.0
SCL		●		●	●	●			●	●																							
TX1	●		●	●	●	●			●	●	●																						UART1EN:XBR2.2
RX1		●		●	●	●			●	●	●	●																					
CEX0	●		●	●	●	●			●	●	●	●	●																				PCA0ME:XBR0.[5:3]
CEX1		●		●	●	●			●	●	●	●	●	●																			
CEX2			●		●	●			●	●	●	●	●	●	●																			
CEX3				●		●			●	●	●	●	●	●	●	●																		
CEX4					●				●	●	●	●	●	●	●	●	●																	
CEX5						●				●	●	●	●	●	●	●	●	●																
ECI	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●														ECI0E:XBR0.6
CP0	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●													CP0E:XBR0.7
CP1	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●												CP1E:XBR1.0
T0	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●											T0E:XBR1.1
/INT0	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										INT0E:XBR1.2
T1	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									T1E:XBR1.3
/INT1	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								INT1E:XBR1.4
T2	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							T2E:XBR1.5
T2EX	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						T2EXE:XBR1.6
T4	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					T4E:XBR2.3
T4EX	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				T4EXE:XBR2.4
/SYSCLK	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			SYSCKE:XBR1.7
CNVSTR0	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		CNVSTE0:XBR2.0
CNVSTR2	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	CNVSTE2:XBR2.5
						ALE	/RD	/WR	AIN2.0/A8	AIN2.1/A9	AIN2.2/A10	AIN2.3/A11	AIN2.4/A12	AIN2.5/A13	AIN2.6/A14	AIN2.7/A15	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7		
									AIN2 输入/非复用地址高								复用地址高/非复用地址低								复用数据/非复用数据									

图 18.5 优先权交叉开关译码表

18.1.7 交叉开关引脚分配示例

在本例中（图 18.6），我们将配置交叉开关，为 UART0、SMBus、UART1、/INT0 和/INT1 分配端口引脚（共 8 个引脚）。另外，我们将外部存储器接口配置为复用方式并使用低端口。我们还将 P1.2、P1.3 和 P1.4 配置为模拟输入，以便用 ADC2 测量加在这些引脚上的电压。配置步骤如下：

1. 按 UART0EN = 1、SMB0EN = 1、INT0E = 1、INT1E = 1 和 EMIFLE =1 设置 XBR0、XBR1 和 XBR2，则有：XBR0 = 0x05，XBR1 = 0x14，XBR2 = 0x02。
2. 将外部存储器接口配置为复用方式并使用低端口，有：PRTSEL = 0，EMD2 = 0。
3. 将作为模拟输入的端口 1 引脚配置为模拟输入方式：设置 P1MDIN 为 0xE3（P1.4、P1.3 和 P1.2 为模拟输入，所以它们的对应 P1MDIN 位被设置为逻辑‘ 0 ’）。
4. 设置 XBARE = 1 以使能交叉开关：XBR2= 0x42。
   - UART0 有最高优先权，所以 P0.0 被分配给 TX0，P0.1 被分配给 RX0。
   - SMBus 的优先权次之，所以 P0.2 被分配给 SDA，P0.3 被分配给 SCL。
   - 接下来是 UART1，所以 P0.4 被分配给 TX1。由于外部存储器接口选在低端口（EMIFLE = 1），所以交叉开关跳过 P0.6(/RD)和 P0.7(/WR)。又因为外部存储器接口被配置为复用方式，所以交叉开关也跳过 P0.5(ALE)。下一个未被跳过的引脚 P1.0 被分配给 RX1。
   - 接下来是/INT0，被分配到引脚 P1.1。
   - 将 P1MDIN 设置为 0xE3，使 P1.2、P1.3 和 P1.4 被配置为模拟输入，导致交叉开关跳过这些引脚。
   - 下面优先权高的是/INT1，所以下一个未跳过的引脚 P1.5 被分配给/INT1。
   - 在执行对片外操作的 MOVX 指令期间，外部存储器接口将驱动端口 2 和端口 3（由图 18.6 中的红点表示）。
5. 我们将 UART0 的 TX 引脚（TX0，P0.0）、UART1 的 TX 引脚（TX1，P0.4）的输出设置为推挽方式，通过设置 P0MDOUT = 0x11 来实现。
6. 我们通过设置 P0MDOUT |= 0xE0，P2MDOUT = 0xFF 和 P3MDOUT = 0xFF 将 EMIF 控制的所有引脚的输出方式都配置为推挽方式。
7. 我们通过设置 P1MDOUT = 0x00（配置输出为漏极开路）和 P1 = 0xFF（逻辑‘ 1 ’选择高阻态）禁止 3 个模拟输入引脚的输出驱动器。

（EMIFLE = 1；EMIF 工作在复用方式；P1MDIN = 0xE3
XBR0 = 0x05; XBR1 = 0x14; XBR2 = 0x42）

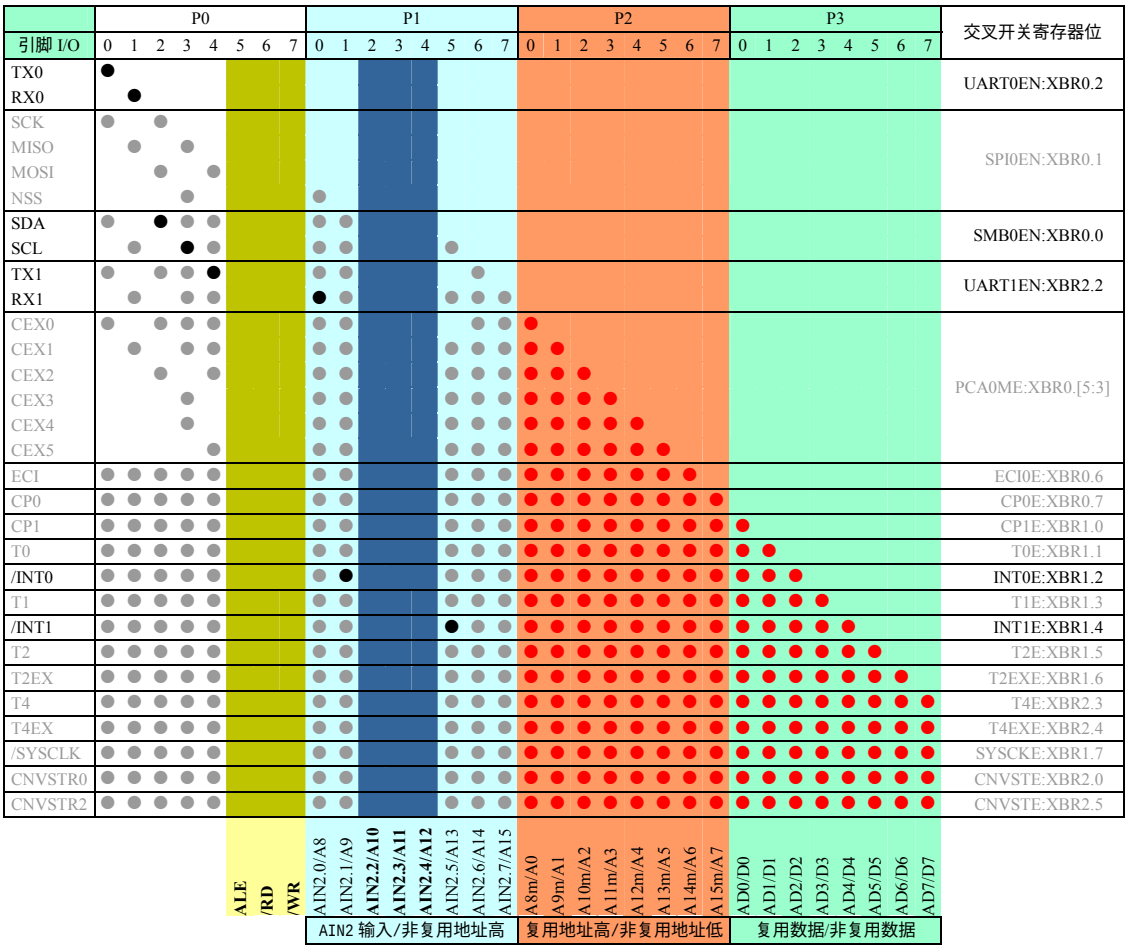


图 18.6 交叉开关示例

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CP0E	ECI0E	PCA0ME			UART0EN	SPI0EN	SMB0EN	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xE1
SFR 页： F

位 7： CP0E：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： ECI0E：PCA0 外部计数器输入允许位
0：PCA0 外部计数器输入不连到端口引脚。
1：PCA0 外部计数器输入（ECI0）连到端口引脚。

位 5-3： PCA0ME：PCA0 模块 I/O 允许位
000：所有的 PCA0 I/O 都不连到端口引脚。
001：CEX0 连到端口引脚。
010：CEX0、CEX1 连到 2 个端口引脚。
011：CEX0、CEX1、CEX2 连到 3 个端口引脚。
100：CEX0、CEX1、CEX2、CEX3 连到 4 个端口引脚。
101：CEX0、CEX1、CEX2、CEX3、CEX4 连到 5 个端口引脚。
110：CEX0、CEX1、CEX2、CEX3、CEX4、CEX5 连到 6 个端口引脚。

位 2： UART0EN：UART0 I/O 允许位。
0：UART0 I/O 不连到端口引脚。
1：UART0 的 TX0 连到 P0.0，RX0 连到 P0.1。

位 1： SPI0EN：SPI 总线 I/O 允许位。
0：SPI0 I/O 不连到端口引脚。
1：SPI0 的 SCK、MISO、MOSI 和 NSS 连到 4 个端口引脚。
注意：当 SPI 被设置为 3 线方式时，NSS 信号不被分配端口引脚。

位 0： SMB0EN：SMBus 总线 I/O 允许位
0：SMBus0 I/O 不连到端口引脚。
1：SMBus0 的 SDA 和 SCL 连到 2 个端口引脚。

图 18.7 XBR0：端口 I/O 交叉开关寄存器 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SYSCKE	T2EXE	T2E	INT1E	T1E	INT0E	T0E	CP1E	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xE2
SFR 页： F

位 7： SYSCKE：/SYSCLK 输出允许位
0：/SYSCLK 不连到端口引脚。
1：/SYSCLK（1、2、4 或 8 分频）连到端口引脚。分频系数由寄存器 CLKSEL 中的 CLKDIV1-0 位决定（见 14. 振荡器）。

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： CP1E：比较器 1 输出允许位
0：CP1 不连到端口引脚。
1：CP1 连到端口引脚。

图 18.8 XBR1：端口 I/O 交叉开关寄存器 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
WEAKPUD	XBARE	CNVST2E	T4EXE	T4E	UART1E	EMIFLE	CNVST0E	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0xE3
SFR 页： F

位 7： WEAKPUD：弱上拉禁止位。
0：弱上拉全局允许。
1：弱上拉全局禁止。

位 6： XBARE：交叉开关使能位
0：交叉开关禁止。端口 0、1、2 和 3 的所有引脚被强制为输入方式。
1：交叉开关使能。

位 5： CNVST2E：外部转换启动输入 2 允许位
0：CNVSTR2 不连到端口引脚。
1：CNVSTR2 连到端口引脚。

位 4： T4EXE：T4EX 输入允许位
0：T4EX 不连到端口引脚。
1：T4EX 连到端口引脚。

位 3： T4E：T4 输入允许位
0：T4 不连到端口引脚。
1：T4 连到端口引脚。

位 2： UART1E：UART1 I/O 允许位
0：UART1 I/O 不连到端口引脚。
1：UART1 的 TX 和 RX 连到两个端口引脚。

位 1： EMIFLE：外部存储器接口低端口使能位
0：P0.7、P0.6 和 P0.5 的功能由交叉开关或端口锁存器决定。
1：如果 EMI0CF.4 = ‘ 0 ’（外部存储器接口为复用方式）
则 P0.7 (/WR)、P0.6 (/RD)和 P0.5 (/ALE)被交叉开关跳过，它们的输出状态由端口锁存器和外部存储器接口决定。
1：如果 EMI0CF.4 = ‘ 1 ’（外部存储器接口为非复用方式）
则 P0.7 (/WR)和 P0.6 (/RD)被交叉开关跳过，它们的输出状态由端口锁存器和外部存储器接口决定。

位 0： CNVST0E：ADC0 外部转换启动输入允许位
0：CNVSTR0 不连到端口引脚。
1：CNVSTR0 连到端口引脚。

图 18.9 XBR2：端口 I/O 交叉开关寄存器 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
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
SFR 页： 所有页

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

注：P0.7 (/WR)、P0.6 (/RD)和 P0.5 (/ALE)可由外部数据存储器接口驱动。更详细的信息见“ 17. 外部数据存储器接口和片内 XRAM ”。关于为外部存储器访问配置交叉开关的信息见图 18.9。

图 18.10 P0：端口 0 寄存器

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

SFR 地址：0xA4
SFR 页： F

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

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

图 18.11 P0MDOUT：端口 0 输出方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
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
SFR 页： 所有页

位 7-0： P1.[7:0]：端口 1 输出锁存器位。
（写 － 输出出现在 I/O 引脚，根据 XBR0、XBR1 和 XBR2 寄存器的设置）
0：逻辑低电平输出。
1：逻辑高电平输出。（若相应的 P1MDOUT.n 位 = 0，则为漏极开路）。
（读－与 XBR0、XBR1 和 XBR2 寄存器的设置无关）
0：P1.n 为逻辑低电平。
1：P1.n 为逻辑高电平。

注：
1．P1.[7:0]可以被配置为 ADC2 的输入 AIN2.[7:0]。在这种情况下，交叉开关的引脚分配将跳过这些引脚，它们的数字输入通路被禁止，由 P1MDIN（见图 18.13）决定。注意：在模拟方式，引脚的输出方式由端口 1 锁存器和 P1MDOUT（图 18.14）决定。有关 ADC2 的更详细信息见“ 7. ADC2（8 位 ADC）”。
2．P1.[7:0]可以由外部数据存储器接口驱动（在非复用方式作为地址 A[15:8]）。有关外部存储器接口的更详细信息见“ 17. 外部数据存储器接口和片内 XRAM ”。

图 18.12 P1：端口 1 寄存器

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

SFR 地址：0xAD
SFR 页： F

位 7-0： P1MDIN.[7:0]：端口 1 输入方式位。
0：端口引脚被配置为模拟输入方式。数字输入通路被禁止（读端口位将总是返回‘ 0 ’）。引脚的弱上拉被禁止。
1：端口引脚被配置为数字输入方式。读端口位将返回引脚的逻辑电平。当被配置为数字输入方式时，弱上拉状态由 WEAKPUD 位（XBR2.7，见图 18.9）决定。

图 18.13 P1MDIN：端口 1 输入方式寄存器

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

SFR 地址：0xA5
SFR 页： F

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

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

图 18.14 P1MDOUT：端口 1 输出方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
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
SFR 页： 所有页

位 7-0： P2.[7:0]：端口 2 输出锁存器位。
（写 － 输出出现在 I/O 引脚，根据 XBR0、XBR1 和 XBR2 寄存器的设置）
0：逻辑低电平输出。
1：逻辑高电平输出。（若相应的 P2MDOUT.n 位 = 0，则为漏极开路）。
（读－与 XBR0、XBR1 和 XBR2 寄存器的设置无关）
0：P2.n 为逻辑低电平。
1：P2.n 为逻辑高电平。

注：P2.[7:0] 可以由外部数据存储器接口驱动（在复用方式作为地址 A[15:8] 或在非复用方式作为地址 A[7:0]）。有关外部存储器接口的更详细信息见“ 17. 外部数据存储器接口和片内 XRAM ”。

图 18.15 P2：端口 2 数据寄存器

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

SFR 地址：0xA6
SFR 页： F

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

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

图 18.16 P2MDOUT：端口 2 输出方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
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
SFR 页： 所有页

位 7-0： P3.[7:0]：端口 3 输出锁存器位。
（写 － 输出出现在 I/O 引脚，根据 XBR0、XBR1 和 XBR2 寄存器的设置）
0：逻辑低电平输出。
1：逻辑高电平输出。（若相应的 P3MDOUT.n 位 = 0，则为漏极开路）。
（读－与 XBR0、XBR1 和 XBR2 寄存器的设置无关）
0：P3.n 为逻辑低电平。
1：P3.n 为逻辑高电平。

注：P3.[7:0] 可以由外部数据存储器接口驱动（在复用方式作为 AD[7:0]或在非复用方式作为 D[7:0]）。有关外部存储器接口的更详细信息见“ 17. 外部数据存储器接口和片内 XRAM ”。

图 18.17 P3：端口 3 数据寄存器

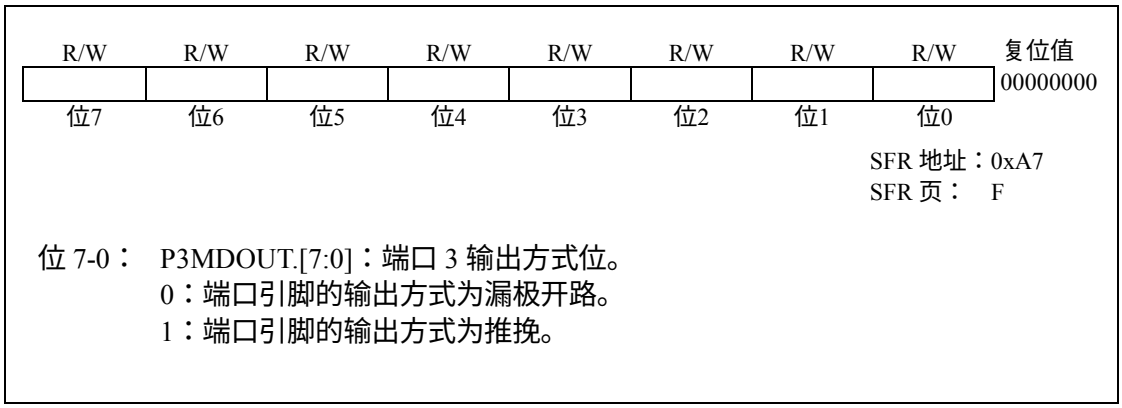


图 18.18 P3MDOUT：端口 3 输出方式寄存器

18.2 端口 4-7（仅限于 100 脚 TQFP 器件）

端口 4-7 的所有端口引脚都可用作通用 I/O（GPIO），通过读和写相应的端口数据寄存器（见图 18.19、图 18.21、图 18.23 和图 18.25）访问每个端口，这些端口数据寄存器是一组即可以按位寻址又可以按字节寻址的特殊功能寄存器。注意：端口 4、5、6、7 寄存器均位于 SFR 页 F，要访问这些寄存器时必须将 SFRPAGE 寄存器设置为 0x0F。

读端口数据寄存器时（或端口位）时，返回的是端口引脚本身的逻辑状态。例外的情况发生在执行*读-修改-写*指令（ANL、ORL、XRL、JBC、CPL、INC、DEC、DJNZ、JBC、CLR、SETB 和位传送操作）期间。在*读-修改-写*指令的读周期，读入的是端口数据寄存器的内容，而不是端口引脚本身的状态。注意：当时钟频率大于 50 MHz 时，向一个引脚写入后立即读引脚时（即写指令后面紧跟一条读指令），由于端口驱动器存在传输延迟，读指令可能返回该引脚在写操作之前的逻辑电平值。

18.2.1 配置无引出脚的端口

尽管 P4、P5、P6 和 P7 在 64 脚 TQFP 器件中没有对应的引脚，但端口数据寄存器仍然存在并可为软件所用。由于数字输入通路保持活动状态，所以建议不要将这些引脚处于“浮空”状态，以避免因输入浮空为一个无效逻辑电平而导致不必要的功率消耗。下面的任何一种措施都可以防止这种情况出现：

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

18.2.2 配置端口引脚的输出方式

每个端口引脚的输出方式都可被配置为漏极开路或推挽方式。在推挽方式，向端口数据寄存器中的相应位写逻辑‘0’将使端口引脚被驱动到 GND，写逻辑‘1’将使端口引脚被驱动到 VDD。在漏极开路方式，向端口数据寄存器中的相应位写逻辑‘0’将使端口引脚被驱动到 GND，写逻辑‘1’将使端口引脚处于高阻状态。当系统中不同器件的端口引脚有共享连接，即多个输出连接到同

一个物理线时（例如 SMBus 连接中的 SDA 信号），使用漏极开路方式可以防止不同器件之间的冲突。

端口 4-7 引脚的输出方式由相应的 PnMDOUT 寄存器中的位决定。PnMDOUT 中的每一位控制相应端口引脚的输出方式（见图 18.20、图 18.22、图 18.24 和图 18.26）。例如，要将端口引脚 P4.3 配置为推挽方式（数字输出），则应将 P4MDOUT.3 设置为逻辑‘1’。器件复位后所有端口引脚均默认为漏极开路方式。

18.2.3 配置端口引脚为数字输入

通过设置输出方式为“漏极开路”并向端口数据寄存器中的相应位写‘1’将端口引脚配置为数字输入。例如，设置 P7MDOUT.7 为逻辑‘0’并设置 P7.7 为逻辑‘1’即可将 P7.7 配置为数字输入。

18.2.4 弱上拉

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

18.2.5 外部存储器接口

如果外部存储器接口（EMIF）被设置在高端口（端口 4-7），EMIFLE（XBR2.1）位应被设置为逻辑‘0’。

如果外部存储器接口被设置在高端口并且发生一次片外 MOVX 操作，则在该 MOVX 指令执行期间外部存储器接口将控制有关端口引脚的输出状态，而不管端口数据寄存器的设置如何。端口引脚的输出配置不受 EMIF 操作的影响，但读操作将禁止数据总线上的输出驱动器。有关外部存储器接口的详细信息见“17. 外部数据存储器接口和片内 XRAM”。

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	11111111
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：0xC8
SFR 页： F

位 7-0： P4.[7:0]：端口 4 输出锁存器位。
写 － 输出出现在 I/O 引脚。
0：逻辑低电平输出。
1：逻辑高电平输出。（若相应的 P4MDOUT.n 位 = 0，则为漏极开路）。见图 18.20。
读 － 返回 I/O 引脚的状态。
0：P4.n 为逻辑低电平。
1：P4.n 为逻辑高电平。

注：P4.7（/WR）、P4.6（/RD）和 P4.5（ALE） 可以由外部数据存储器接口驱动。有关外部存储器接口的更详细信息见“ 17. 外部数据存储器接口和片内 XRAM ”。

图 18.19 P4：端口 4 数据寄存器

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

SFR 地址：0x9C
SFR 页： F

位 7-0： P4MDOUT.[7:0]：端口 4 输出方式位。
0：端口引脚的输出方式为漏极开路。
1：端口引脚的输出方式为推挽。

图 18.20 P4MDOUT：端口 4 输出方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0	11111111
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：0xD8
SFR 页： F

位 7-0： P5.[7:0]：端口 5 输出锁存器位。
写 － 输出出现在 I/O 引脚。
0：逻辑低电平输出。
1：逻辑高电平输出。（若相应的 P5MDOUT.n 位 = 0，则为漏极开路）。见图 18.22。
读 － 返回 I/O 引脚的状态。
0：P5.n 为逻辑低电平。
1：P5.n 为逻辑高电平。

注：P5.[7:0] 可以由外部数据存储器接口驱动（在非复用方式作为地址 A[15:8]）。有关外部存储器接口的更详细信息见“ 17. 外部数据存储器接口和片内 XRAM ”。

图 18.21 P5：端口 5 数据寄存器

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

SFR 地址：0x9D
SFR 页： F

位 7-0： P5MDOUT.[7:0]：端口 5 输出方式位。
0：端口引脚的输出方式为漏极开路。
1：端口引脚的输出方式为推挽。

图 18.22 P5MDOUT：端口 5 输出方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
P6.7	P6.6	P6.5	P6.4	P6.3	P6.2	P6.1	P6.0	11111111
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：0xE8
SFR 页： F

位 7-0： P6.[7:0]：端口 6 输出锁存器位。
写 － 输出出现在 I/O 引脚。
0：逻辑低电平输出。
1：逻辑高电平输出。（若相应的 P6MDOUT.n 位 = 0，则为漏极开路）。见图 18.24。
读 － 返回 I/O 引脚的状态。
0：P6.n 为逻辑低电平。
1：P6.n 为逻辑高电平。

注：P6.[7:0] 可以由外部数据存储器接口驱动（在复用方式作为地址 A[15:8] 或在非复用方式作为地址 A[7:0]）。有关外部存储器接口的更详细信息见“ 17. 外部数据存储器接口和片内 XRAM ”。

图 18.23 P6：端口 6 数据寄存器

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

SFR 地址：0x9E
SFR 页： F

位 7-0： P6MDOUT.[7:0]：端口 6 输出方式位。
0：端口引脚的输出方式为漏极开路。
1：端口引脚的输出方式为推挽。

图 18.24 P6MDOUT：端口 6 输出方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
P7.7	P7.6	P7.5	P7.4	P7.3	P7.2	P7.1	P7.0	11111111
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：0xF8
SFR 页： F

位 7-0： P7.[7:0]：端口 7 输出锁存器位。
写 － 输出出现在 I/O 引脚。
0：逻辑低电平输出。
1：逻辑高电平输出。（若相应的 P7MDOUT.n 位 = 0，则为漏极开路）。见图 18.26。
读 － 返回 I/O 引脚的状态。
0：P7.n 为逻辑低电平。
1：P7.n 为逻辑高电平。

注：P7.[7:0] 可以由外部数据存储器接口驱动（在复用方式作为 AD[7:0]，在非复用方式作为 D[7:0]）。有关外部存储器接口的更详细信息见“ 17. 外部数据存储器接口和片内 XRAM ”。

图 18.25 P7：端口 7 数据寄存器

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

SFR 地址：0x9F
SFR 页： F

位 7-0： P7MDOUT.[7:0]：端口 7 输出方式位。
0：端口引脚的输出方式为漏极开路。
1：端口引脚的输出方式为推挽。

图 18.26 P7MDOUT：端口 7 输出方式寄存器

19. SMBus

SMBus0 I/O 接口是一个双线的双向串行总线。SMBus0 完全符合系统管理总线规范 1.1 版，与 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。可以采用延长低电平时间的方法协调同一总线上不同速度的器件。

SMBus 可以工作在主和/或从方式，一个总线上可以有多个主器件。SMBus 提供了 SDA（串行数据）控制、SCL（串行时钟）产生和同步、仲裁逻辑以及起始/停止的控制和产生电路。

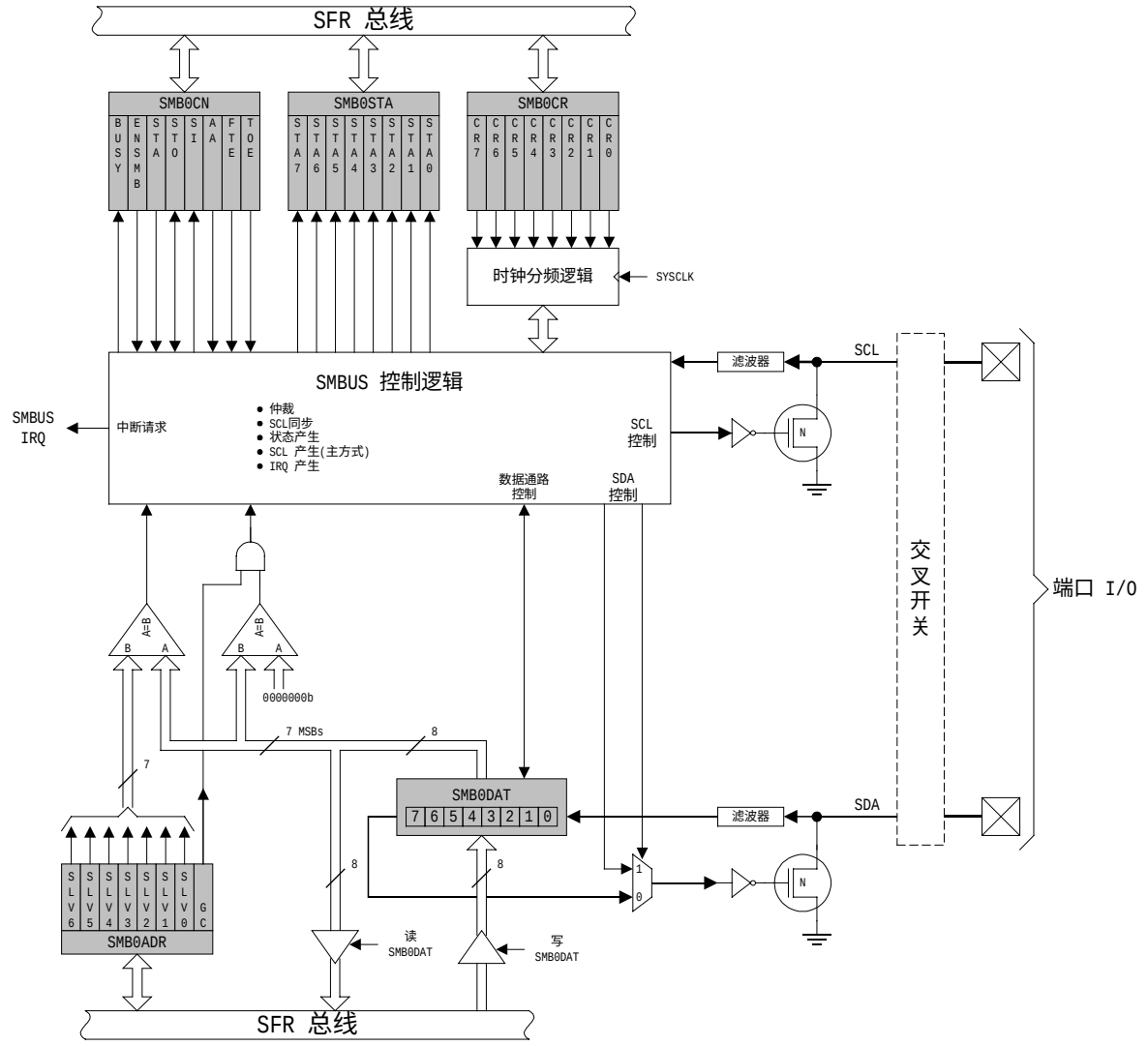


图 19.1　SMBus 原理框图

图 19.2 给出了一个典型的 SMBus 配置。SMBus 接口的工作电压可以在 3.0V 和 5.0V 之间，总线上不同器件的工作电压可以不同。SCL（串行时钟）和 SDA（串行数据）线是双向的，必须通过一个上拉电阻或类似电路将它们连到电源电压。连接在总线上的每个器件的 SCL 和 SDA 都必须是漏极开路或集电极开路的，当总线空闲时，这两条线都被拉到高电平。总线上的最大器件数只受所要求的上升和下降时间的限制，上升和下降时间分别不能超过 300ns 和 1000ns。

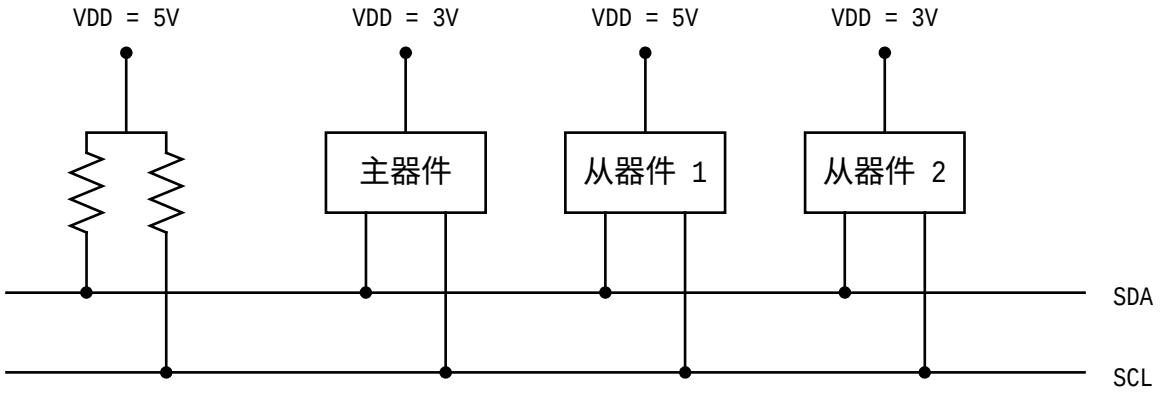


图 19.2 典型 SMBus 配置

19.1 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

19.2 SMBus 协议

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

一次典型的 SMBus 数据传输包括一个起始条件（START）、一个地址字节（位 7-1：7 位从地址；位 0：R/W 方向位）、一个或多个字节的数据和一个停止条件（STOP）。每个接收的字节（由一个主器件或从器件）都必须用 SCL 高电平期间的 SDA 低电平（见图 19.3）来确认（ACK）。如果接收器件不确认，则发送器件将读到一个“非确认”（NACK），这用 SCL 高电平期间的 SDA 高电平表示。

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

所有的数据传输都由主器件启动，可以寻址一个或多个目标从器件。主器件产生一个起始条件，然后发送地址和方向位。如果本次数据传输是一个从主器件到从器件的写操作，则主器件每发送一个数据字节后等待来自从器件的确认。如果是一个读操作，则由从器件发送数据并等待主器件的确认。在数据传输结束时，主器件产生一个停止条件，结束数据交换并释放总线。图 19.3 示出了一次典型的 SMBus 数据传输过程。

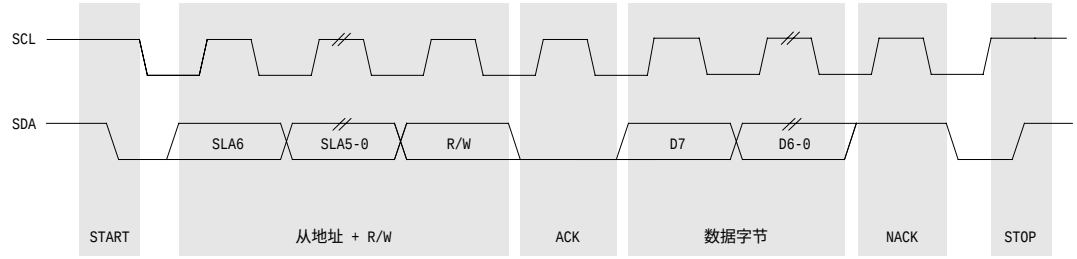


图 19.3 SMBus 数据传输

19.2.1 总线仲裁

一个主器件只能在总线空闲时启动一次传输。在一个停止条件之后或 SCL 和 SDA 保持高电平已经超过了指定时间（见 19.2.4 节），则总线是空闲的。当两个或多个器件在同一时刻启动数据传输时，仲裁机制迫使一个主器件放弃总线。这些主器件继续发送起始条件，直到其中一个主器件发送高电平而另一个在 SDA 上发送低电平。由于总线是漏极开路的，总线将被拉为低电平。发送高电平的主器件将检测到这个 SDA 低电平并放弃总线。赢得总线的器件继续其数据传输过程，而未赢得总线的器件成为从器件。该仲裁机制是非破坏性的：总会有一个器件赢得总线，不会发生数据丢失。

19.2.2 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共存于一个总线上。为了使低速从器件能与高速主器件通信，在传输期间采取低电平扩展。从器件可以保持 SCL 为低电平以扩展时钟低电平时间，这实际上相当于降低了串行时钟频率。

19.2.3 SCL 低电平超时

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

19.2.4 SCL 高电平（SMBus 空闲）超时

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

19.3 SMBus 数据传输方式

SMBus 接口可以被配置为工作在主方式和/或从方式。在某一时刻，它将工作在下述 4 种方式之一：主发送器、主接收器、从发送器或从接收器。传输方式状态译码（使用 SMB0STA 状态寄存器）的情况见表 19.1。下面以中断驱动的 SMBus0 应用为例来说明这四种工作方式；SMBus0 也可以工作在查询方式。

19.3.1 主发送器方式

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

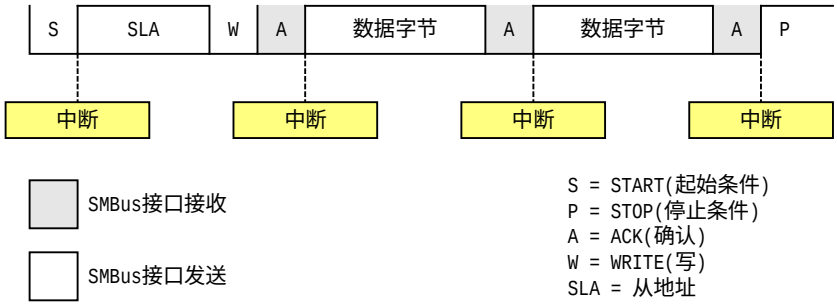


图 19.4 典型的主发送器时序

19.3.2 主接收器方式

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

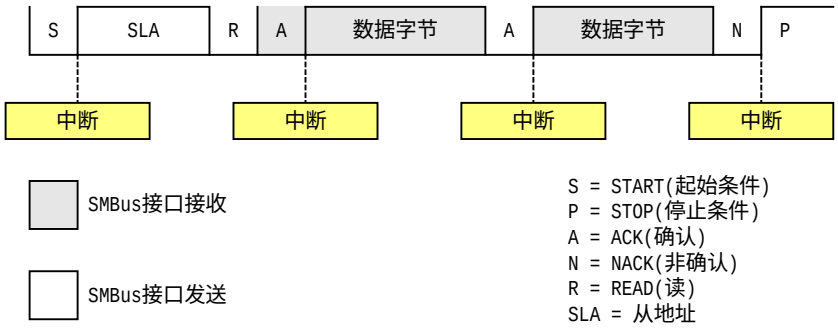


图 19.5 典型的主接收器时序

19.3.3 从发送器方式

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

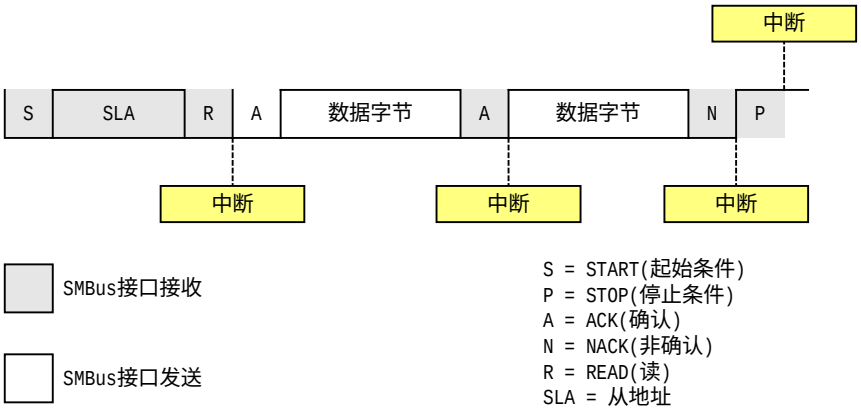


图 19.6　典型的从发送器时序

19.3.4 从接收器方式

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

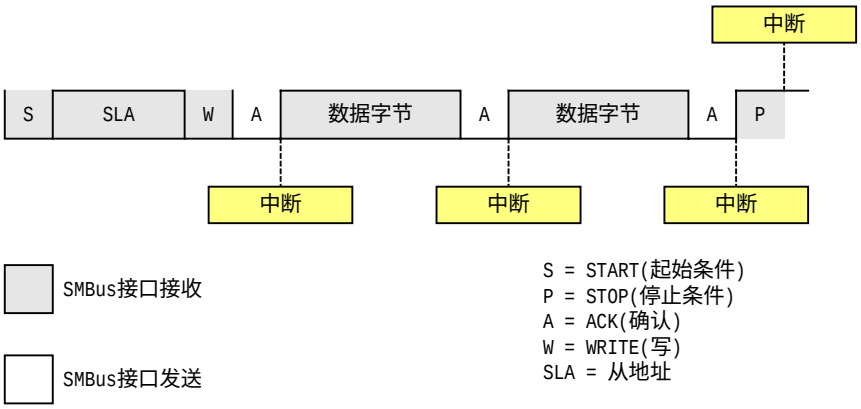


图 19.7　典型的从接收器时序

19.4 SMBus 特殊功能寄存器

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

19.4.1 控制寄存器

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

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

设置起始标志（STA，SMB0CN.5）为逻辑 1 将使 SMBus0 工作于主方式。如果总线空闲，SMBus0 硬件将产生一个起始条件。如果总线不空闲，SMBus0 硬件将等待停止条件释放总线，然后根据 SMB0CR 的值在经过 5 微秒的延时后产生一个起始条件。（根据 SMBus 协议，如果总线处于等待状态的时间超过 50 微秒而没有检测到停止条件，SMBus0 接口可以认为总线是空闲的。）如果 STA 被设置为逻辑 1，而此时 SMBus 处于主方式并且已经传送了一个或多个字节，则将产生一个重复起始条件。

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

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

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

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

设置 SMBus0 空闲定时器允许位（FTE，SMB0CN.1）为逻辑 1 将使能 SMB0CR 中的定时器。当 SCL 变高时，SMB0CR 的定时器向上计数。定时器溢出指示总线空闲超时：如果 SMBus0 等待产生一个起始条件，则将在超时发生后进行。总线空闲周期应小于 50 微秒（见图 19.9，SMBus0 时钟速率寄存器）。

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

R/W	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
SFR 页： 0

位 7： BUSY：忙状态标志
0：SMBus0 空闲
1：SMBus0 忙

位 6： ENSMB：SMBus0 使能
该位使能/禁止 SMBus0 串行接口
0：禁止 SMBus0
1：使能 SMBus0

位 5： STA：SMBus0 起始标志
0：不发送起始条件。
1：当作为主器件时，若总线空闲，则发送出一个起始条件。（如果总线不空闲，在收到停止条件后再发送起始条件。）如果 STA 被置 1，而此时已经发送或接收了一个或多个字节并且没有收到停止条件，则发送一个重复起始条件。

位 4： STO：SMBus0 停止标志
0：不发送停止条件。
1：将 STO 置为逻辑 1 将发送一个停止条件。当收到停止条件时，硬件将 STO 清为逻辑 0。如果 STA 和 STO 都被置位，则发送一个停止条件后再发送一个起始条件。在从方式，置位 STO 标志将导致 SMBus 的行为象收到了停止条件一样。

位 3： SI：SMBus0 串行中断标志
当 SMBus0 进入 27 种可能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：SMBus0 有效确认标志
该位定义在 SCL 线应答周期内返回的应答类型。
0：在应答周期内返回“非确认”(SDA 线高电平)。
1：在应答周期内返回“确认”(SDA 线低电平)。

位 1： FTE：SMBus0 空闭定时器允许位
0：无 SCL 高电平超时。
1：当 SCL 高电平时间超过由 SMB0CR 规定的极限值时发生超时。

位 0： TOE：SMBus0 超时允许位
0：无 SCL 低电平超时。
1：当 SCL 处于低电平的时间超过由定时器 3（如果被允许）定义的极限值时发生超时。

图 19.8 SMB0CN：SMBus0 控制寄存器

19.4.2 时钟速率寄存器

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

SFR 地址：0xCF
SFR 页： 0

位 7-0： SMB0CR.[7:0]：SMBus0 时钟速率预设值
SMB0CR 时钟速率寄存器用于控制主方式下串行时钟 SCL 的频率。存储在 SMB0CR 寄存器中的 8 位字预装一个专用的 8 位定时器。该定时器向上计数，当计满回到 0x00 时 SCL 改变逻辑状态。

SMB0CR[7:0]的值根据下面的方程设置，其中 *SMB0CR* 是 SMB0CR 寄存器中的 8 位无符号数值。*SYSCLK* 是系统时钟频率（单位为 Hz）。

$$SMB0CR < \left(288 - 0.85 \times \frac{SYSCLK}{4} \right) / 1.125$$

SCL 信号的高电平和低电平时间由下式给出，其中 SYSCLK 是系统时钟频率（以 MHz 为单位）。

$$T_{LOW} = 4 \times (256 - SMB0CR) / SYSCLK$$

$$T_{HIGH} \cong 4 \times (258 - SMB0CR) / SYSCLK + 625ns$$

使用相同的 SMB0CR 值，总线空闲超时周期由下式给出。

$$T_{BFT} \cong 10 \times \frac{4 \times (256 - SMB0CR) + 1}{SYSCLK}$$

图 19.9 SMB0CR：SMBus0 时钟速率寄存器

19.4.3 数据寄存器

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

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

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
SFR 页： 0

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

图 19.10 SMB0DAT：SMBus0 数据寄存器

19.4.4 地址寄存器

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

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
SFR 页： 0

位 7-1： SLV6-SLV0：SMBus0 从地址
这些位用于存放 7 位从地址，当器件工作在从发送器或从接收器方式时，SMBus0 将应答该地址。SLV6 是地址的最高位，对应从 SMBus0 收到的地址字节的第一位。

位 0： GC：全局呼叫地址使能
该位用于使能全局呼叫地址（0x00）识别。
0：忽略全局呼叫地址。
1：识别全局呼叫地址。

图 19.11 SMB0ADR：SMBus0 地址寄存器

19.4.5 状态寄存器

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

对于用户软件而言，SMB0STA 的内容只在 SI 标志为逻辑 1 时有定义。软件不应向 SMB0STA 寄存器写入；如果写入，将会产生不确定的结果。表 19.1 列出了 28 个 SMBus0 状态和对应的状态码。

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
SFR 页： 0

位 7-3： STA7-STA3：SMBus0 状态代码
这些位含有 SMBus0 状态代码。共有 28 个可能的状态码，每个状态码对应一个 SMBus 状态。在 SI 标志（SMB0CN.3）置位时，SMB0STA 中的状态码有效。当 SI 标志为逻辑 0 时，SMB0STA 中的内容无定义。任何时候向 SMB0STA 寄存器写入将导致不确定的结果。

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。

图 19.12 SMB0STA：SMBus 状态寄存器

表 19.1. SMB0STA 状态码和状态

方式	状态码	SMBus 状态	典型操作
主发送器/主接收器	0x08	起始条件已发出	将从地址+R/W 装入到 SMB0DAT。清‘0’STA。
	0x10	重复起始条件已发出	将从地址+R/W 装入到 SMB0DAT。清‘0’STA。
主发送器	0x18	从地址+W 已发出。收到 ACK。	将要发送的数据装入到 SMB0DAT。
	0x20	从地址+W 已发出。收到 NACK。	确认查询重试。置位 STO+STA。
	0x28	数据字节已发出。收到 ACK。	1） 将下一字节装入到 SMB0DAT，或 2） 置位 STO，或 3） 置位 STO，然后置位 STA 以发送重复起始条件。
	0x30	数据字节已发出。收到 NACK。	1） 重试传输或 2） 置位 STO。
	0x38	竞争失败	保存当前数据。
主接收器	0x40	从地址+R 已发出。收到 ACK。	如果只接收一个字节，清 AA 位（收到字节后发送 NACK）。等待接收数据。
	0x48	从地址+R 已发出。收到 NACK。	确认查询重试。置位 STO+STA
	0x50	数据字节收到。ACK 已发出	读 SMB0DAT。等待下一字节。 如果下一字节是最后字节，清除 AA。
	0x58	数据字节收到。NACK 已发出	置位 STO。

表 18.1 SMB0STA 状态码和状态（续）

方式	状态码	SMBus 状态	典型操作
从接收器	0x60	收到自身的从地址+W。ACK 已发出。	等待数据
	0x68	在作为主器件发送 SLA+R/W 时竞争失败。收到自身地址+W。ACK 已发出。	保存当前数据以备总线空闲时重试。等待数据
	0x70	收到全局呼叫地址。ACK 已发出。	等待数据
	0x78	在作为主器件发送 SLA+R/W 时竞争失败。收到全局呼叫地址+W。ACK 已发出。	保存当前数据以备总线空闲时重试。
	0x80	收到数据字节。ACK 已发出。	读 SMB0DAT。等待下一字节或停止条件。
	0x88	收到数据字节。NACK 已发出。	置位 STO 以复位 SMBus
	0x90	在全局呼叫地址之后收到数据字节。ACK 已发出。	读 SMB0DAT。等待下一字节或停止条件。
	0x98	在全局呼叫地址之后收到数据字节。NACK 已发出。	置位 STO 以复位 SMBus
	0xA0	收到停止条件或重复起始条件。	不需操作
从发送器	0xA8	收到自己的从地址+R。ACK 已发出。	将要发送的数据装入到 SMB0DAT。
	0xB0	在作为主器件发送 SLA+R/W 时竞争失败。收到自身地址+R。ACK 已发出。	保存当前数据以备总线空闲时重试。将要发送的数据装入到 SMB0DAT。
	0xB8	数据字节已发送。收到 ACK。	将要发送的数据装入到 SMB0DAT。
	0xC0	数据字节已发送。收到 NACK。	等待停止条件
	0xC8	最后一个字节已发送（AA=0）。收到 ACK。	置位 STO 以复位 SMBus
从器件	0xD0	SCL 时钟高电平定时器超时（根据 SMB0CR）。	置位 STO 以复位 SMBus
所有方式	0x00	总线错误（非法起始条件或停止条件）。	置位 STO 以复位 SMBus
	0xF8	空闲状态。	该状态不置位 SI

20. 增强型串行外设接口（SPI0）

串行外设接口（SPI0）提供访问一灵活的全双工串行总线。SPI0 可以作为主器件或从器件，有 3 线工作方式和 4 线工作方式，并支持在同一总线上连接多个主器件和从器件。从选择信号（NSS）可以被配置为输入以择选从方式下的 SPI0，或在多主环境中禁止主器件方式操作，以避免两个以上主器件试图同时进行数据传输时产生冲突。NSS 还可以被配置为主方式下的片选输出，或在 3 线操作时被禁止。在主方式，可以用通用端口 I/O 引脚选择多个从器件。

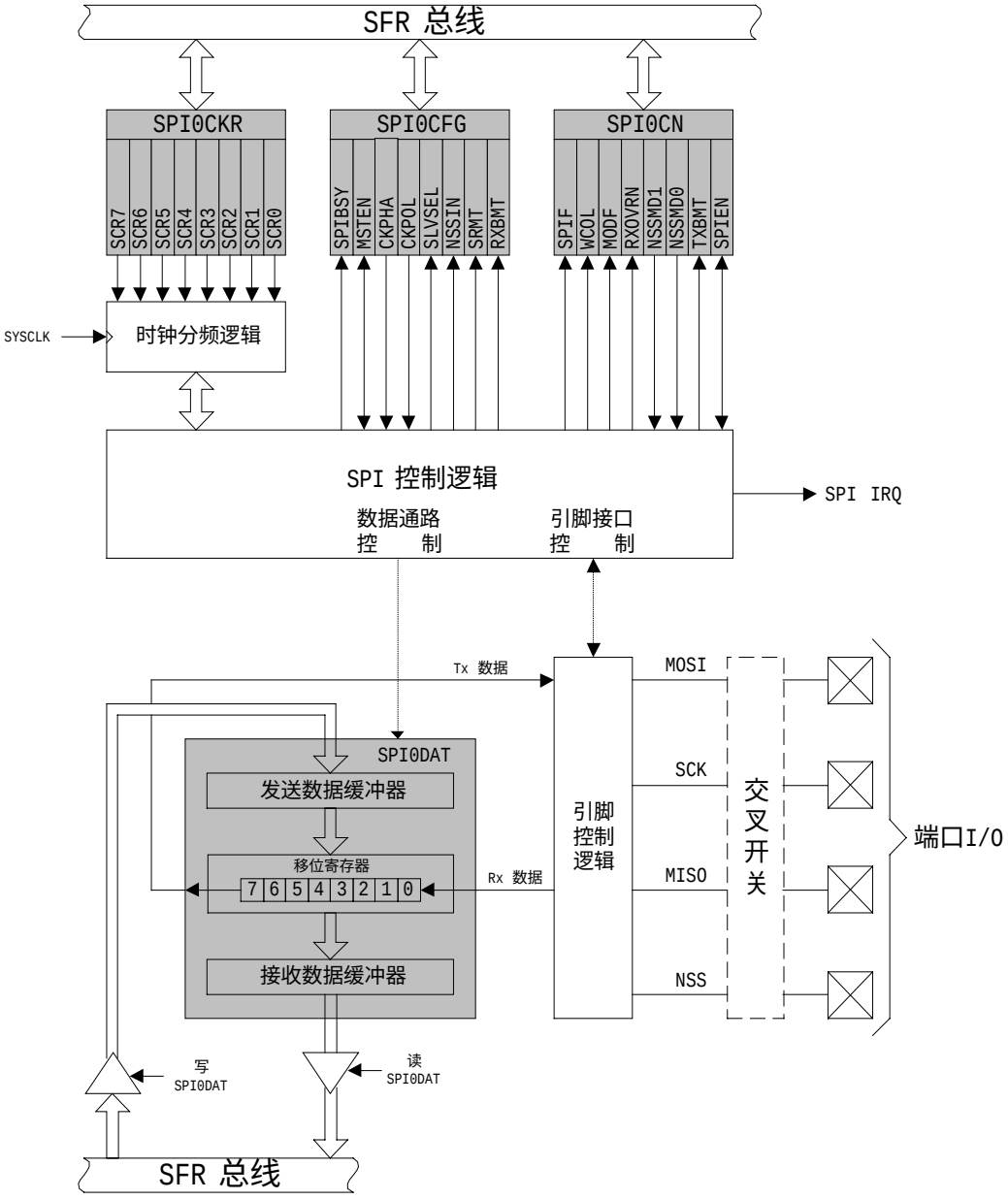


图 20.1 SPI 原理框图

20.1 信号说明

下面介绍 SPI0 所使用的 4 个信号（MOSI、MISO、SCK、NSS）。

20.1.1 主输出、从输入（MOSI）

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

20.1.2 主输入、从输出（MISO）

主入从出（MISO）信号是从器件的输出和主器件的输入，用于从从器件到主器件的串行数据传输。当 SPI0 作为主器件时，该信号是输入；当 SPI0 作为从器件时，该信号是输出。数据传输时最高位在先。当 SPI 模块被禁止或 SPI 工作在 4 线从方式但未被选中时，MISO 引脚为高阻状态。当作为从器件工作在 3 线方式时，MISO 总是由移位寄存器的 MSB 驱动。

20.1.3 串行时钟（SCK）

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

20.1.4 从选择（NSS）

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

1. NSSMD[1:0] = 00：3 线主方式或 3 线从方式：SPI0 工作在 3 线方式，NSS 被禁止。当作为从器件时，SPI0 总是被选择为 3 线方式。由于没有选择信号，SPI0 必须是 3 线总线上唯一的从器件。这种情况用于一个主器件和一个从器件之间点对点通信。
2. NSSMD[1:0] = 01：4 线从方式或多主方式：SPI0 工作在 4 线方式，NSS 作为输入。当作为从器件时，NSS 选择从 SPI0 器件。当作为主器件时，NSS 信号的负跳变禁止 SPI0 的主器件功能，因此可以在同一个 SPI 总线上使用多个主器件。
3. NSSMD[1:0] = 1x：4 线主方式：SPI0 工作在 4 线方式，NSS 作为输出。NSSMD0 的设置值决定 NSS 引脚的输出电平。这种配置只能在 SPI0 作为主器件时使用。

图 20.2、图 20.3 和图 20.4 给出了不同方式下的典型连接图。注意：NSSMD 位的设置影响器件的引脚分配。当工作在 3 线主或从方式时，NSS 不被交叉开关分配引脚。在所有其它方式，NSS 必须被映射到器件引脚。有关通用端口 I/O 和交叉开关的详细信息见“ 18.1 端口输入/输出 ”。

20.2 SPI0 主方式

SPI 主器件启动 SPI 总线上所有的数据传输。通过将主允许标志（MSTEN，SPI0CFG.6）置 1 将 SPI0 置于主方式。当处于主方式时，向 SPI0 数据寄存器（SPI0DAT）写入一个字节时是写发送缓冲器。如果 SPI 移位寄存器为空，发送缓冲器中的数据字节被传送到移位寄存器，数据传输开始。SPI0 主器件立即在MOSI线上串行移出数据,同时在SCK上提供串行时钟。在传输结束后SPIF（SPI0CN.7）标志被置为逻辑 1。如果中断被允许，在 SPIF 标志置位时将产生一个中断请求。在全双工操作中，当 SPI 主器件在 MOSI 线向从器件发送数据时，被寻址的 SPI 从器件同时在 MISO 线上向主器件发送其移位寄存器中的内容。因此，SPIF 标志既作为发送完成标志又作为接收数据准备好标志。从从器件接收的数据字节以 MSB 在先的形式传送到主器件的移位寄存器。当一个数据字节被完全移入移位寄存器时，便被传送到接收缓冲器，处理器通过读 SPI0DAT 来读该字节。

当被配置为主器件时，SPI0 可以工作在下面的三种方式之一：多主方式、3 线单主方式或 4 线单主方式。当 NSSMD1（SPI0CN.3）=0 且 NSSMD0（SPI0CN.2）=1 时，是默认的多主方式。在该方式，NSS 是器件的输入，用于禁止主 SPI0，以允许另一主器件访问总线。在该方式，当 NSS 被拉为低电平时，MSTEN（SPI0CN.6）和 SPIEN（SPI0CN.0）位被硬件清 0，以禁止 SPI 主器件，且方式错误标志（MODF，SPI0CN.5）被置 1。如果中断被允许，将产生方式错误中断。在这种情况下，必须用软件重新使能 SPI0。在多主系统中，当器件不作为系统主器件使用时，一般被默认为从器件。在多主方式，可以用通用 I/O 引脚对从器件单独寻址（如果需要）。图 20.2 给出了两个主器件在多主方式下的连接图。

当 NSSMD1（SPI0CN.3）=0 且 NSSMD0（SPI0CN.2）=0 时，SPI0 工作在 3 线单主方式。在该方式，NSS 未被使用，也不被交叉开关映射到外部端口引脚。在该方式，应使用通用 I/O 引脚选择要寻址的从器件。图 20.3 给出了一个 3 线主方式主器件和一个从器件的连接图。

当 NSSMD1（SPI0CN.3）=1 时，SPI0 工作在 4 线单主方式。在该方式，NSS 被配置为输出引脚，可被用作从选择信号去选中一个 SPI 器件。在该方式，NSS 的输出值由 NSSMD0（SPI0CN.2）控制（用软件）。可以用通用 I/O 引脚选择另外的从器件。图 20.4 给出了一个 4 线主方式主器件和两个从器件的连接图。

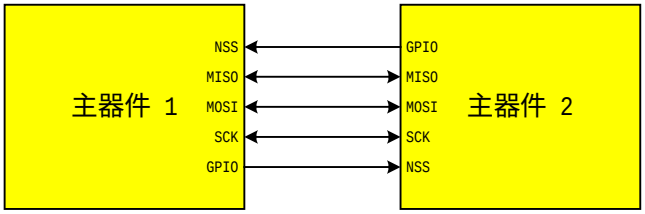


图 20.2　多主方式连接图

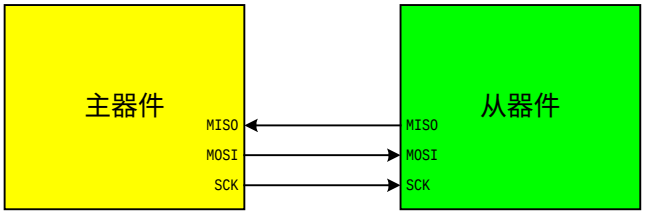


图 20.3　3 线单主方式和 3 线单从方式连接图

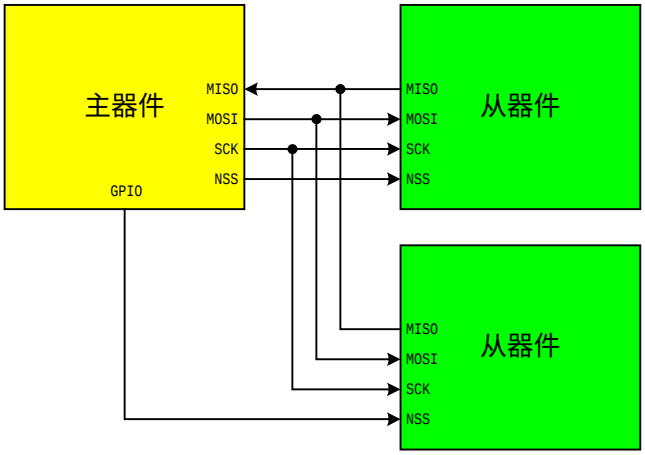


图 20.4　4 线单主方式和 4 线从方式连接图

20.3 SPI0 从方式

当 SPI0 被使能而未被配置为主器件时，它将作为 SPI 从器件工作。作为从器件，由主器件控制串行时钟，从 MOSI 移入数据，从 MISO 引脚移出数据。SPI0 逻辑中的位计数器对 SCK 边沿计数。当 8 位数据经过移位寄存器后，SPIF 标志被置为逻辑 1，接收到的字节被传送到接收缓冲器。通过读 SPI0DAT 来读取接收缓冲器中的数据。从器件不能启动数据传送。通过写 SPI0DAT 将要发送给主器件的数据预装到移位寄存器。写往 SPI0DAT 的数据是双缓冲的，首先被放在发送缓冲器。如果移位寄存器为空，发送缓冲器中的数据会立即被传送到移位寄存器。当移位寄存器中已经有数据时，SPI 在下一次（或当前）SPI 传输的最后一个 SCK 边沿将发送缓冲器的内容装入移位寄存器。

当被配置为从器件时，SPI0 可以工作 4 线或 3 线方式。当 NSSMD1（SPI0CN.3）=0 且 NSSMD0（SPI0CN.2）=1 时，是默认的 4 线方式。在 4 线方式，NSS 被分配端口引脚并被配置为数字输入。当 NSS 为逻辑 0 时，SPI0 被使能；当 NSS 为逻辑 1 时，SPI0 被禁止。在 NSS 的下降沿，位计数器被复位。注意，对应每次字节传输，在第一个有效 SCK 边沿到来之前，NSS 信号必须被驱动到低电平至少两个系统时钟周期。图 20.4 给出了两个 4 线方式从器件和一个主器件的连接图。

当 NSSMD1（SPI0CN.3）=0 且 NSSMD0（SPI0CN.2）=0 时，SPI0 工作在 3 线从方式。在该方式，NSS 未被使用，也不被交叉开关映射到外部端口引脚。由于在 3 线从方式无法唯一地寻址从器件，所以 SPI0 必须是总线上唯一的从器件。需要注意的是，在 3 线从方式，没有外部手段对位计数器复位以判断是否收到一个完整的字节。只能通过用 SPIEN 位禁止并重新使能 SPI0 来复位位计数器。图 20.3 给出了一个 3 线从器件和一个主器件的连接图。

20.4 SPI0 中断源

如果 SPI0 中断被允许，在下述 4 个标志位被置 1 时将产生中断。

注意：这 4 个标志位都必须用软件清 0。

1. 在每次字节传输结束，SPI 中断标志 SPIF（SPI0CN.7）被置为逻辑 1。该标志在所有 SPI 方式都能发生。
2. 如果在发送缓冲器中的数据尚未被传送到移位寄存器时写 SPI0DAT，则写冲突标志 WCOL（SPI0CN.6）被置 1。发生这种情况时，写 SPI0DAT 的操作被忽略，不会对发送缓冲器写入。该标志在所有 SPI 方式都能发生。
3. 当 SPI0 被配置为工作于多主方式的主器件而 NSS 被拉为低电平时，方式错误标志 MODF（SPI0CN.5）被置 1。当发生方式错误时，SPI0CN 中的 MSTEN 和 SPIEN 位被清 0，以禁止 SPI0 并允许另一个主器件访问总线。
4. 当 SPI0 被配置为从器件并且一次传输结束，而接收缓冲器中还保持着上一次传输的数据未被读取时，接收溢出标志 RXOVRN（SPI0CN.4）被置 1。新接收的字节将不被传送到接收缓冲器，允许前面接收的字节被读取。引起溢出的数据字节丢失。

20.5 串行时钟时序

使用 SPI0 配置寄存器（SPI0CFG）中的时钟控制选择位可以在串行时钟相位和极性的 4 种组合中选择其一。CKPHA 位（SPI0CFG.5）选择两种时钟相位（锁存数据所用的边沿）中的一种。CKPOL 位（SPI0CFG.4）在高电平有效和低电平有效的时钟之间选择。主器件和从器件必须被配置为使用相同的时钟相位和极性。在改变时钟相位和极性期间应禁止 SPI0（通过清除 SPIEN 位，SPI0CN.0）。主方式下时钟和数据线的时序关系示于图 20.5；从方式下时钟和数据线的时序关系示于图 20.6 和图 20.7。注意：当 C8051F04x, C8051F06x, C8051F12x, C8051F31x, C8051F32x, C8051F33x 中的两个器件通信时，主器件和从器件的 CKPHA 必须被置 0。

图 20.10 所示的 SPI0 时钟速率寄存器（SPI0CKR）控制主方式的串行时钟频率。当工作于从方式时该寄存器被忽略。当 SPI 被配置为主器件时，最大数据传输率（位/秒）是系统时钟频率的二分之一或 12.5MHz，取两者的较低值。当 SPI 被配置为从器件时，全双工操作的最大数据传输率（位/秒）是系统时钟频率的十分之一，前提是主器件与从器件的系统时钟同步发出 SCK、NSS（在 4 线从方式）和串行输入数据。如果主器件发出的 SCK、NSS 及串行输入数据不同步，则最大数据传输率（位/秒）必须小于系统时钟频率的十分之一。在主器件只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 SCK、NSS 和串行输入数据与从器件系统时钟同步的情况下。

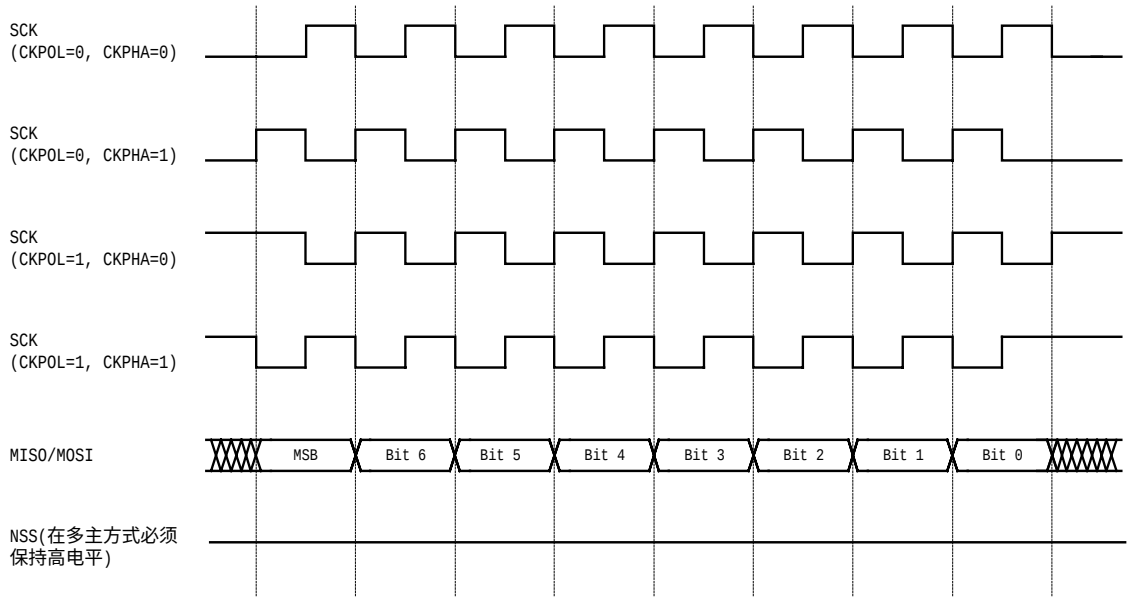


图 20.5　主方式数据/时钟时序图

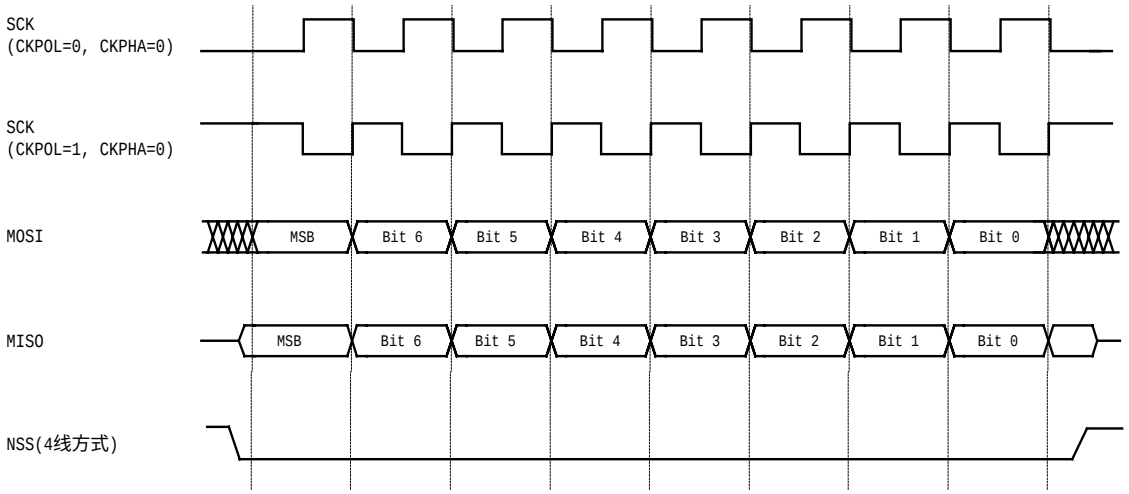


图 20.6　从方式数据/时钟时序图（CKPHA = 0）

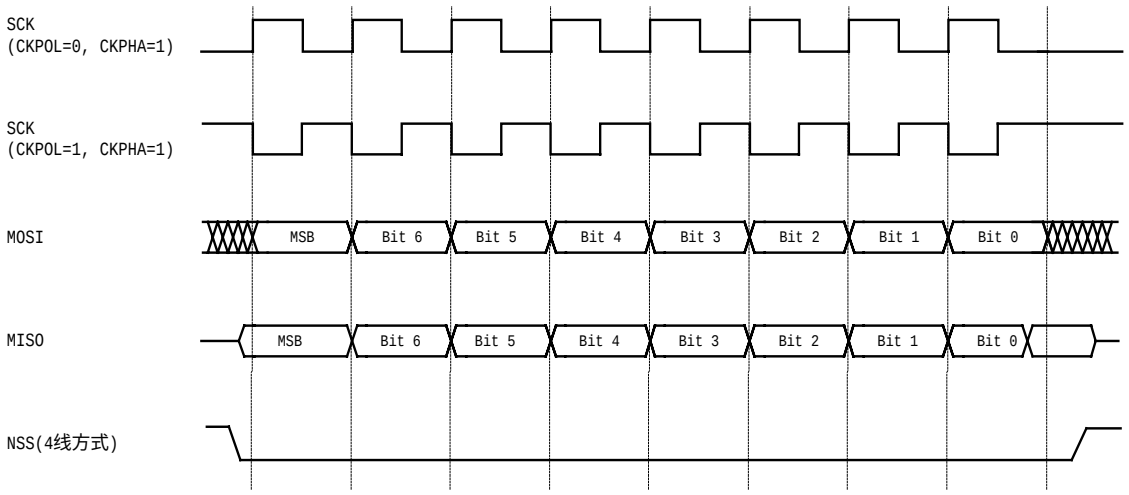


图 20.7　从方式数据/时钟时序图（CKPHA = 1）

20.6 SPI 特殊功能寄存器

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

R	R/W	R/W	R/W	R	R	R	R	复位值
SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT	00000111
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x9A
SFR 页： 0

位 7： SPIBSY：SPI 忙标志（只读）
当一次 SPI 传输正在进行时（主或从方式），该位被置为逻辑 1。

位 6： MSTEN：主方式使能位
0：禁止主方式，工作在从方式。
1：使能主方式，工作在主器件方式。

位 5： CKPHA：SPI0 时钟相位。
该位控制 SPI0 时钟的相位。
0：在 SCK 周期的第一个边沿采样数据。†
1：在 SCK 周期的第二个边沿采样数据。†

位 4： CKPOL：SPI0 时钟极性
该位控制 SPI0 时钟的极性。
0：SCK 在空闲状态时处于低电平。
1：SCK 在空闲状态时处于高电平。

位 3： SLVSEL：从选择标志（只读）。
当 NSS 引脚为低电平时该位被置 1，表示 SPI0 是被选中的从器件。当 NSS 引脚为高电平时（未被选中为从器件）该位被清 0。该位不指示 NSS 引脚的即时值，而是该引脚输入的去噪信号。

位 2： NSSIN：NSS 引脚瞬时值（只读）
该位指示读该寄存器时 NSS 引脚的即时值。该信号未被去噪。

位 1： SRMT：移位寄存器空标志（只在从方式有效，(只读)）。
当所有数据都被移入/移出移位寄存器并且没有新数据可以从发送缓冲器读出或向接收缓冲器写入时，该位被置 1。当数据字节被从发送缓冲器传送到移位寄存器时，该位被清 0。
注：在主方式，SRMT = 1。

位 0： RXBMT：接收缓冲器空（只在从方式有效，只读）
当接收缓冲器被读取且没有新数据时，该位被置 1。如果在接收缓冲器中有新数据未被读取，则该位被清 0。
注：在主方式，RXBMT = 1。

†在从方式，MOSI 上的数据在每个数据位的中间被采样。在主方式，MISO 上的数据在每个数据位结束前一个 SYSCLK 被采样，以便为从器件提供最大的建立时间。见表 20.1 的时序参数。

图 20.8 SPI0CFG：SPI0 配置寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	复位值
SPIF	WCOL	MODF	RXOVRN	NSSMD1	NSSMD0	TXBMT	SPIEN	00000110
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：0xF8
SFR 页： 0

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

位 6： WCOL：写冲突标志
该位由硬件置为逻辑 1（并产生一个 SPI0 中断），表示数据传送期间对 SPI0 数据寄存器进行了写操作。该位必须用软件清 0。

位 5： MODF：方式错误标志
当检测到主方式冲突（NSS 为低电平，MSTEN＝1，NSSMD[1:0] = 01）时，该位由硬件置为逻辑 1（并产生一个 SPI0 中断）。该位不能被硬件自动清 0，必须用软件清 0。

位 4： RXOVRN：接收溢出标志（只适用于从方式）
当前传输的最后一位已经移入 SPI0 移位寄存器，而接收缓冲器中仍保存着前一次传输未被读取的数据时该位由硬件置为逻辑 1（并产生一个 SPI0 中断）。该位不会被硬件自动清 0，必须用软件清 0。

位 3-2： NSSMD1-NSSMD0：从选择方式位
在下面的 NSS 操作方式中选择。见“ 20.2 SPI0 主方式 ”和 “ 20.3 SPI0 从方式 ”。
00：3 线从方式或 3 线主方式。NSS 信号不连到端口引脚。
01：4 线从方式或多主方式（默认）。NSS 总是器件的输入。
1x：4 线单主方式。NSS 被分配一个输出引脚并输出 NSSMD0 的值。

位 1： TXBMT：发送缓冲器空标志
当新数据被写入发送缓冲器时，该位被清 0。当发送缓冲器中的数据被传送到 SPI 移位寄存器时，该位被置 1，表示可以向发送缓冲器写新数据。

位 0： SPIEN：SPI0 使能位
该位允许／禁止 SPI0。
0：禁止 SPI0
1：使能 SPI0

图 20.9 SPI0CN：SPI0 控制寄存器

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
SFR 页： 0

位 7-0： SCR7-SCR0：SPI0 时钟频率设置位。
当 SPI0 模块被配置为工作于主方式时，这些位决定 SCK 输出的频率。SCK 时钟频率是从系统时钟分频得到的，由下面的方程给出，其中：*SYSCLK* 是系统时钟频率，*SPI0CKR* 是 SPI0CKR 寄存器中的 8 位值。

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

（0 ≤SPI0CKR≤255）

例如：如果 SYSCLK = 2MHz，SPI0CKR = 0x04，则

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

$$f_{SCK} = 200kHz$$

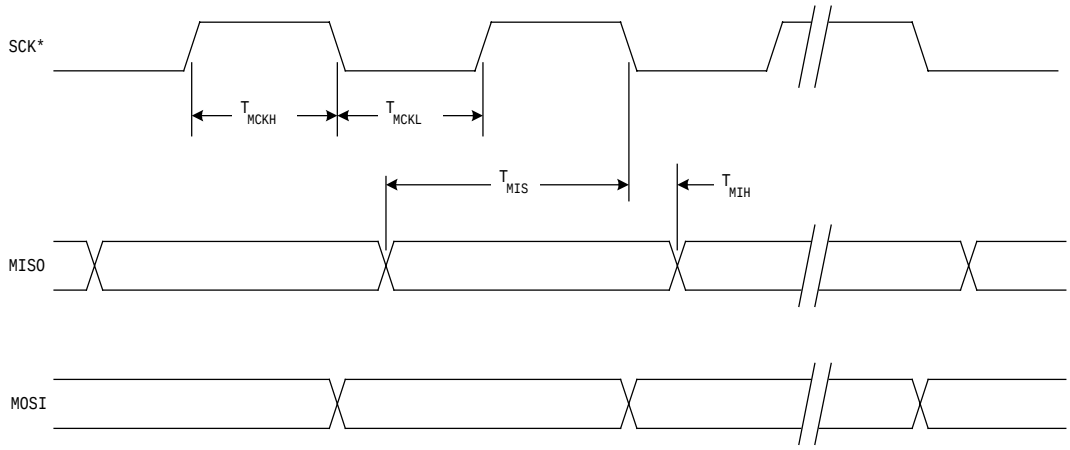
图 20.10 SPI0CKR：SPI0 时钟速率寄存器

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

SFR 地址：0x9B
SFR 页： 0

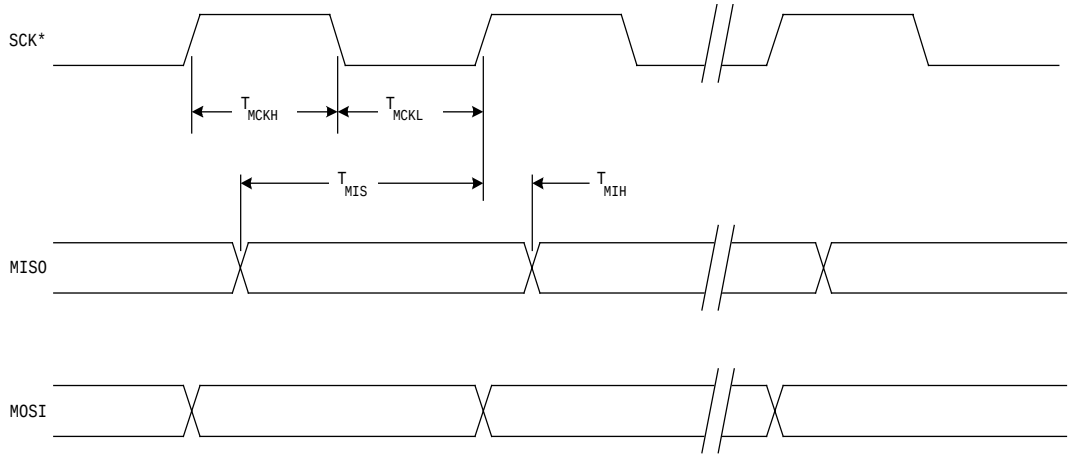
位 7-0： SPI0DAT：SPI0 发送和接收数据寄存器。
SPI0DAT 寄存器用于发送和接收 SPI0 数据。在主方式下，向 SPI0DAT 写入数据时，数据被放到发送缓冲器并启动发送。读 SPI0DAT 返回接收缓冲器的内容。

图 20.11 SPI0DAT：SPI0 数据寄存器



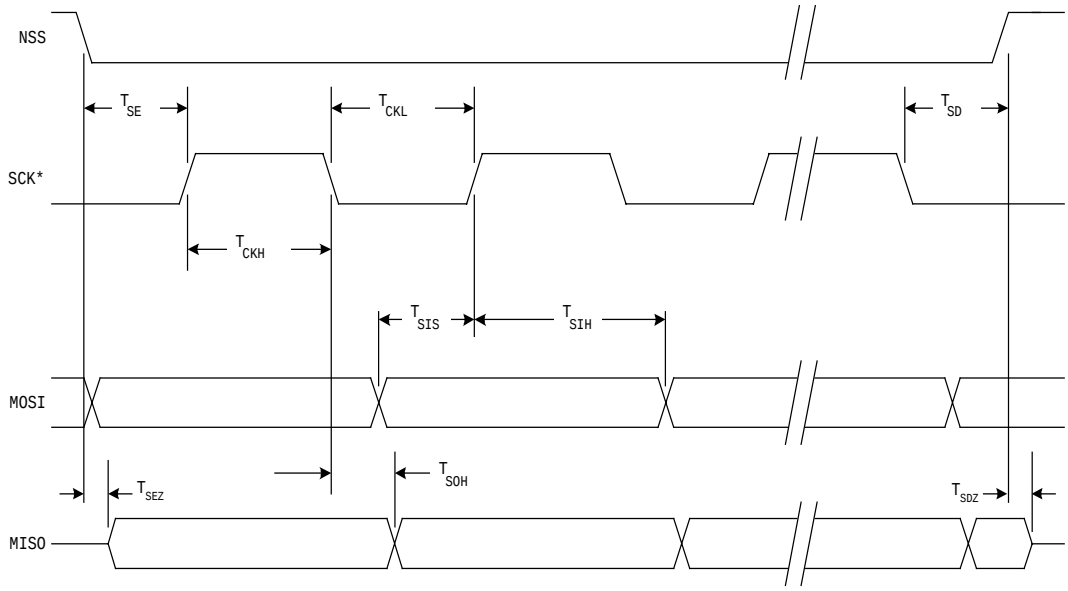
* 这是对应 CKPOL = 0时的 SCK 波形。 对于 CKPOL = 1，SCK波形的极性反向。

图 20.12 SPI 主方式时序（CKPHA = 0）



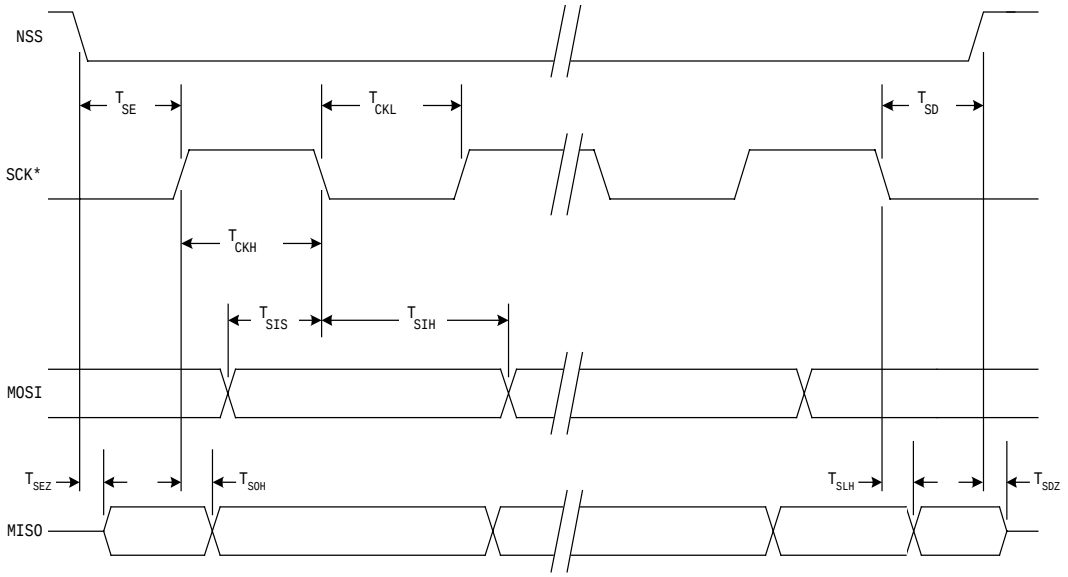
* 这是对应 CKPOL = 0时的 SCK 波形。对于 CKPOL = 1，SCK波形的极性反向。

图 20.13 SPI 主方式时序（CKPHA = 1）



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

图 20.14　SPI 从方式时序（CKPHA = 0）



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

图 20.15　SPI 从方式时序（CKPHA = 1）

表 20.1 SPI 从方式时序参数

参 数	说 明	最小值	最大值	单位
主方式时序（见图 20.12 和图 20.13）				
T_{MCKH}	SCK 高电平时间	$1*T_{SYSCLK}$		ns
T_{MIS}	SCK 低电平时间	$1*T_{SYSCLK}$		ns
T_{MCKH}	MISO 有效到 SCK 移位边沿	$1*T_{SYSCLK}+20$		ns
T_{MIH}	SCK 移位边沿到 MISO 发生改变	0		ns
从方式时序（见图 20.14 和图 20.15）				
T_{SE}	NSS 下降沿到第一个 SCK 边沿	$2*T_{SYSCLK}$		ns
T_{SD}	最后一个 SCK 边沿到 NSS 上升沿	$2*T_{SYSCLK}$		ns
T_{SEZ}	NSS 下降沿到 MISO 有效		$4*T_{SYSCLK}$	ns
T_{SDZ}	NSS 上升沿到 MISO 变为高阻态		$4*T_{SYSCLK}$	ns
T_{CKH}	SCK 高电平时间	$5*T_{SYSCLK}$		ns
T_{CKL}	SCK 低电平时间	$5*T_{SYSCLK}$		ns
T_{SIS}	MOSI 有效到 SCK 采样边沿	$2*T_{SYSCLK}$		ns
T_{SIH}	SCK 采样边沿到 MOSI 发生改变	$2*T_{SYSCLK}$		ns
T_{SOH}	SCK 移位边沿到 MISO 发生改变		$4*T_{SYSCLK}$	ns
T_{SLH}	最后一个 SCK 边沿到 MISO 发生改变（只限于 CKPHA = 1）	$6*T_{SYSCLK}$	$8*T_{SYSCLK}$	ns
注：T_{SYSCLK} 为一个系统时钟（SYSCLK）周期。				

21. UART0

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

对 UART0 的控制和访问是通过相关的特殊功能寄存器即串行控制寄存器（SCON0）和串行数据缓冲器（SBUF0）来实现的。用同一个 SBUF0 地址可以访问发送寄存器和接收寄存器。读操作将自动访问接收寄存器，而写操作自动访问发送寄存器。

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

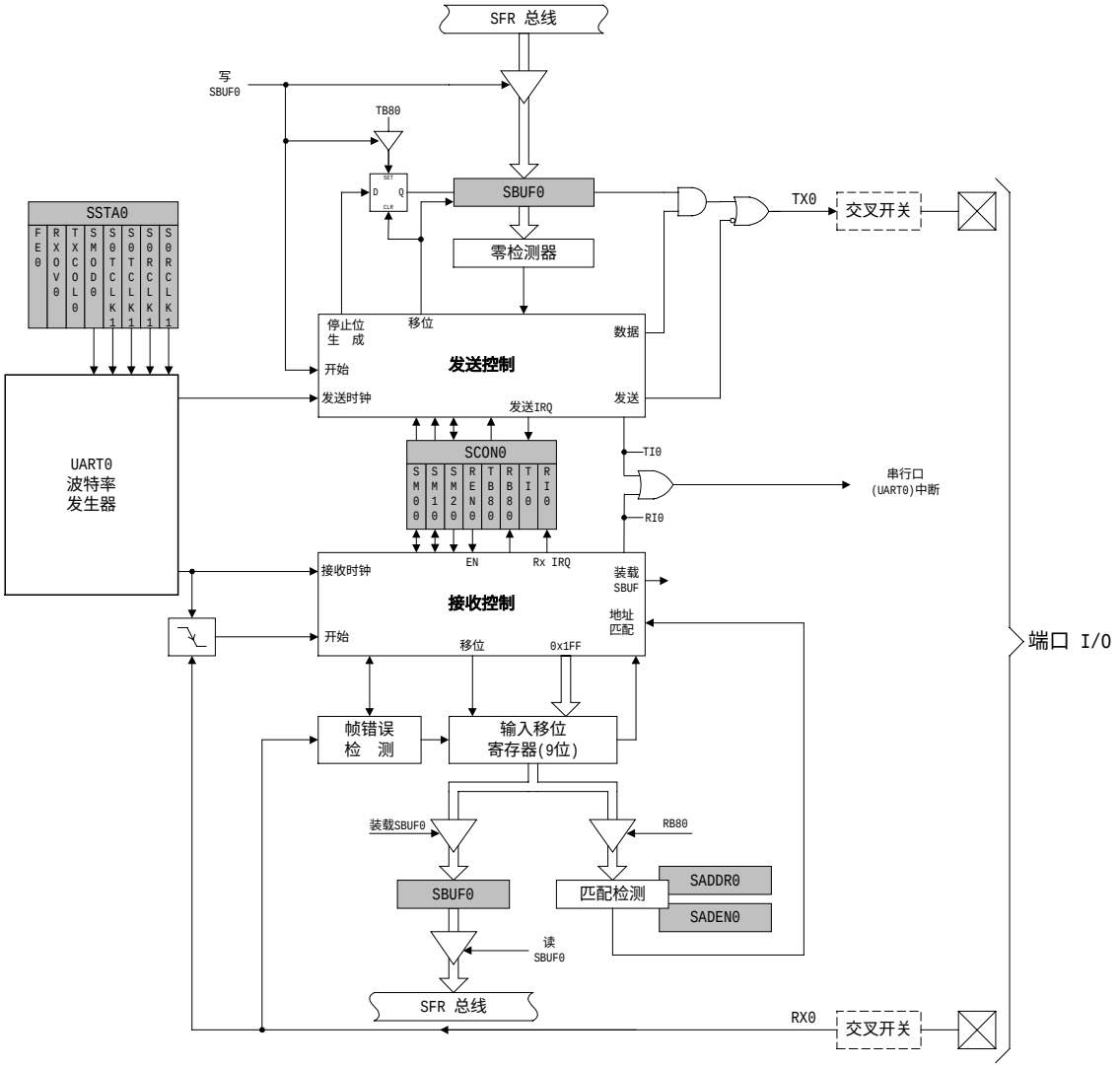


图 21.1　UART0 原理框图

21.1 UART0 工作方式

UART0 提供四种工作方式（一种同步方式和三种异步方式），通过设置 SCON0 寄存器中的配置位选择。这四种方式提供不同的波特率和通信协议。下面的表 21.1 概述了这四种方式。

表 21.1 UART0 工作方式

方式	同步性	波特率时钟	数据位	起始/停止位
0	同步	SYSCLK/12	8	无
1	异步	定时器 1、2、3 或 4 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、2、3 或 4 溢出	9	一个起始位，一个停止位

21.1.1 方式 0：同步方式

方式 0 提供同步、半双工通信。在 RX0 引脚上发送和接收数据，TX0 引脚提供发送和接收的移位时钟。MCU 必须是主器件，因为它要为两个方向的数据传输产生移位时钟（见图 21.3 中的连接图）。

执行一条写 SBUF0 寄存器的指令时开始数据发送。发送/接收的数据为 8 位，LSB 在先（见图 21.2 中的时序图），在第 8 个位时间结束后发送中断标志 TI0（SCON0.1）置位。当接收允许位 REN0（SCON0.4）被设置为逻辑 1 并且接收中断标志 RI0（SCON0.0）被清 0 时开始数据接收。在第 8 位被移入后一个周期 RI0 标志置位，接收过程停止，直到软件清除 RI0 位。如果中断被允许，在 TI0 或 RI0 置位后将发生一次中断。

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

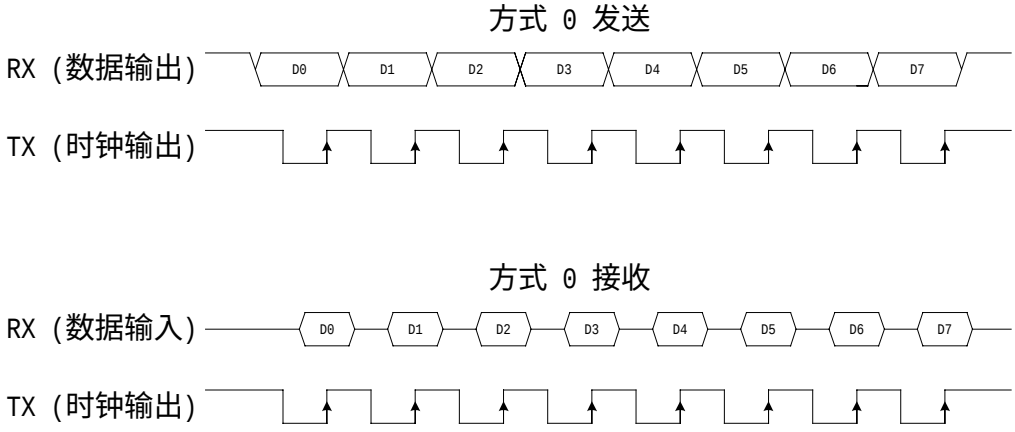


图 21.2 UART 方式 0 时序图

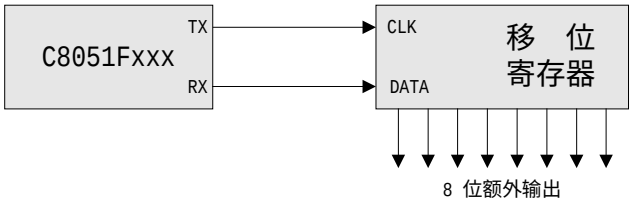


图 21.3 UART0 方式 0 连接

21.1.2 方式 1：8 位 UART，可变波特率

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

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

如果这些条件满足，则 8 位数据被存入 SBUF0，停止位被存入 RB80，RI0 标志被置位。如果这些条件不满足，则不装入 SBU0F 和 RB80，RI0 标志也不被置 1。如果中断被允许，在 TI0 或 RI0 置位时将产生中断。

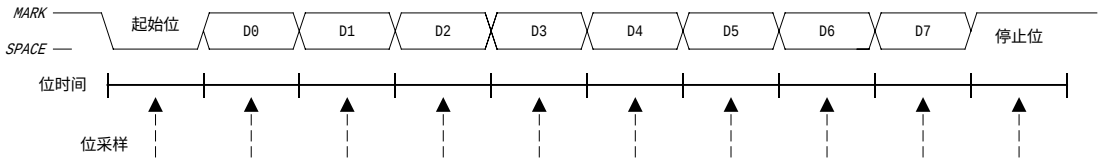


图 21.4 UART0 方式 1 时序图

方式 1 的波特率是定时器溢出时间的函数，如方程 21.1 和方程 21.3 所示。UART0 可以使用定时器 1 工作在 8 位自动重装载方式或定时器 2、3、4 工作在自动重装载方式产生波特率（注意，TX 和 RX 时钟可以分别选择）。每次定时器发生溢出，即从全 1（对定时器 1 为 0xFF，对定时器 2、3、4 为 0xFFFF）返回到 0 时向波特率电路发送一个时钟脉冲。

用 SSTA0 寄存器（见图 21.9）选择定时器 1、2、3 或 4 作为波特率发生源。发送波特率时钟用 S0TCLK1 和 S0TCLK0 位选择，接收波特率时钟用 S0RCLK1 和 S0RCLK0 位选择。

下面给出了方式 1 的波特率方程，其中：T1M 为定时器 1 时钟选择位（CKCON.4），TH1 是定时器 1 的 8 位重装载寄存器，[RCAPnH:RCAPnL]是定时器 2、3、4 的 16 位重装载寄存器。

方程 21.1 使用定时器 1 的方式 1 波特率

当 SMOD0 = 0 时：

方式 1 波特率 = 1/32 × 定时器 1 溢出率

当 SMOD0 = 1 时：

方式 1 波特率 = 1/16 × 定时器 1 溢出率

定时器 1 溢出率由定时器 1 的时钟源（T1CLK）和重载值（TH1）确定。T1CLK 的频率选择见“23.1 定时器 0 和定时器 1”。定时器的溢出率用下面的方程 21.2 计算。

方程 21.2 定时器溢出率

定时器 1 溢出率 = T1CLK /（256 — TH1）

当定时器 2、3 或 4 被选择为波特率源时，波特率用方程 21.3 计算。

方程 21.3 使用定时器 2、3 或 4 的方式 1 波特率

方式 1 波特率 = 1/16 × 定时器 2、3 或 4 溢出率

定时器 2、3 或 4 的溢出率由定时器的时钟源（TnCLK）和保存在 RCAPn 寄存器（n = 2、3 或 4）中的 16 位重载值确定。用下面的方程 21.4 计算。

方程 21.4 定时器 2、3 或 4 溢出率

定时器 2、3 或 4 溢出率 = TnCLK /（65536 — RCAPn）

21.1.3 方式 2：9 位 UART，固定波特率

方式 2 提供异步、全双工通信，每个数据字节共使用 11 位：一个起始位、8 个数据位（LSB 在先）、一个可编程的第九位和一个停止位。方式 2 支持多处理器通信和硬件地址识别（见“21.2 多处理器通信”）。在发送时，第九数据位由 TB80（SCON0.3）中的值决定。它可以被赋值为 PSW 中的奇偶标志 P，或用于多处理器通信。在接收时，第九数据位进入 RB80（SCON0.2），停止位被忽略。

当执行一条向 SBUF0 寄存器写入一个字节的指令时开始数据发送。在发送结束时（停止位开始）发送中断标志 TI0（SCON0.1）置位。在接收允许位 REN0（SCON0.4）被设置为逻辑 1 后的任何时间都可以开始数据接收。收到停止位后如果 RI0 为逻辑 0 并且满足下述条件之一则数据字节将被装入到接收寄存器 SBUF0：

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

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

方式 2 波特率为 SYSCLK/32 或 SYSCLK/64，由寄存器 SSTA0 中的 SMOD0 位决定。

方程 21.5 方式 2 波特率

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

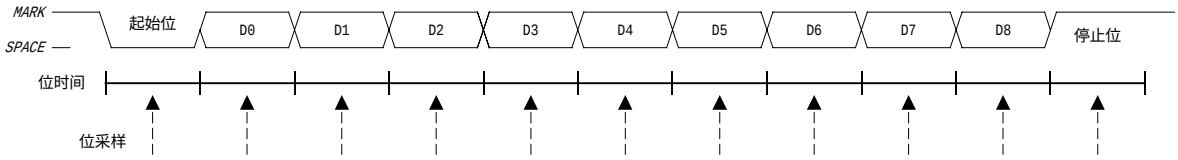


图 21.5　UART0 方式 2 和 3 时序图

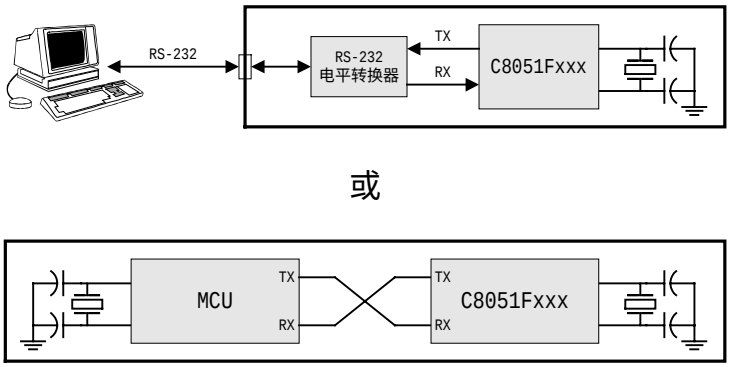


图 21.6　UART0 方式 1、2 和 3 连接图

21.1.4 方式 3：9 位 UART，可变波特率

方式 3 使用方式 2 的传输协议，波特率的确定方式与方式 1 相同。方式 3 操作使用 11 位：一个起始位、8 个数据位（LSB 在先）、一个可编程的第九位和一个停止位。用定时器 1 或定时器 2、3、4 溢出产生波特率，见方程 21.1 和方程 21.3。方式 3 支持多机通信和硬件地址识别。

21.2 多机通信

方式 2 和方式 3 通过使用第九数据位和内置 UART0 地址识别硬件支持一个主处理器与一个或多个从处理器之间的多机通信。当主机想要向一个或多个从机发送数据数据时，它先发送一个用于选择目标从机的地址字节。地址字节与数据字节的区别是：地址字节的第九位为逻辑 1；数据字节的第九位总是设置为逻辑 0。UART0 可以在任何时刻识别（即能产生中断）两种类型的“ 有效 ”地址：（1）掩码地址和（2）广播地址。下面对这两种地址方式进行详细说明。

21.2.1 掩码地址设置

UART0 地址由两个特殊功能寄存器配置：SADDR0（串口地址）和 SADEN0（串口地址使能）。SADEN0 设置 SADDR0 中的地址的屏蔽位：SADEN0 中设置为逻辑‘ 1 ’的位对应于 SADDR0 中那些用来检查接收到的地址字节的位；SADEN0 中被设置为逻辑‘ 0 ’的位对应于 SADDR0 中那些“ 无关 ”位。

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

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

21.2.2 广播寻址

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

例 4，从机 1	例 5，从机 2	例 6，从机 3
SADDR0 = 00110101	SADDR0 = 00110101	SADDR0 = 00110101
SADEN0 = 00001111	SADEN0 = 11110011	SADEN0 = 11000000
广播地址 = 00111111	广播地址 = 11110111	广播地址 = 11110101

广播地址中所有的 0 都被视为无关位。

注意：在上面的例 4、例 5 和例 6 中，每个从机都将地址 0xFF 视为“ 有效 ”广播地址。还应注意到，例 4、例 5 和例 6 使用与例 1、例 2 和例 3 中相同的 SADDR0 和 SADEN0 寄存器值。因此，主机可以用掩码地址单独寻址每个器件，也可以向所有这三个从机进行广播。例如，如果主器件发送一个掩码地址“11110101”，则只有从机 1 会识别该地址有效；如果主器件再发送一个地址“11111111”，则三个从机都会识别该地址为有效广播地址。

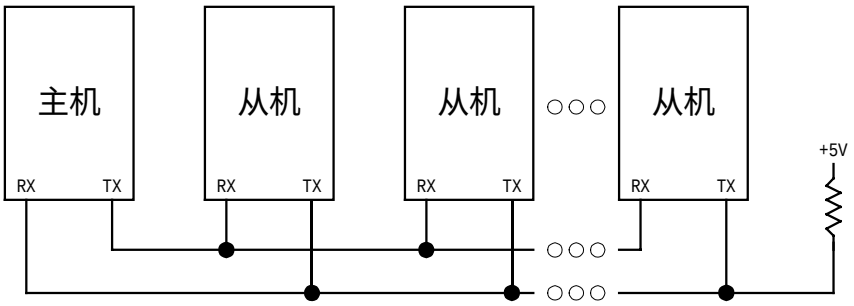


图 21.7 UART 多机方式连接图

21.3 帧错误和传输错误检测

所有方式：

当一次发送过程正在进行时，如果用户软件向 SBUF0 寄存器写数据，则发送冲突位（寄存器 SSSTA0 中的 TXCOL0）被置‘ 1 ’。

方式 1、2 和 3：

如果一个新的数据字节被锁存到接收缓冲器而前面接收的字节尚未被读取，则接收覆盖位（寄存器 SSTA0 中的 RXOVR0）被置‘ 1 ’。如果检测到一个无效（低电平）停止位，则帧错误位（寄存器 SSTA0 中的 FE0）被置‘ 1 ’。

表 21.2. 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*	定时器 2、3 或 4 装载值*	波特率（Hz）**
100	864	0xCA	0xFFCA	115200(115741)
99.5328	864	0xCA	0xFFCA	115200
50.0	432	0xE5	0xFFE5	115200(115741)
49.7664	432	0xE5	0xFFE5	115200
24.0	208	0xF3	0xFFF3	115200(115384)
22.1184	192	0xF4	0xFFF4	115200
18.432	160	0xF6	0xFFF6	115200
11.0592	96	0xFA	0xFFFA	115200
3.6864	32	0xFE	0xFFFE	115200
1.8432	16	0xFF	0xFFFF	115200
100	3472	0x27	0xFF27	28800(28802)
99.5328	3456	0x28	0xFF28	28800
50.0	1744	0x93	0xFF93	28800(28670)
49.7664	1728	0x94	0xFF94	28800
24.0	832	0xCC	0xFFCC	28800(28846)
22.1184	768	0xD0	0xFFD0	28800
18.432	640	0xD8	0xFFD8	28800
11.0592	348	0xE8	0xFFE8	28800
3.6864	128	0xF8	0xFFF8	28800
1.8432	64	0xFC	0xFFFC	28800
100	10416	-	0xFD75	9600(9601)
99.5328	10368	-	0xFD78	9600
50.0	5216	-	0xFEBA	9600(9586)
49.7664	5184	-	0xFEBC	9600
24.0	2496	0x64	0xFF64	9600(9615)
22.1184	2304	0x70	0xFF70	9600
18.432	1920	0x88	0xFF88	9600
11.0592	1152	0xB8	0xFFB8	9600
3.6864	384	0xE8	0xFFE8	9600
1.8432	192	0xF4	0xFFF4	9600

*假定 SMOD0=1 且 T1M=1。

**括号里的数是实际波特率。

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SM00	SM10	SM20	REN0	TB80	RB80	TI0	RI0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：0x98
SFR 页： 0

位 7-6： SM00-SM10：串行口工作方式。
写：当被写入时，这两位按下表选择串行口工作方式：

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

读：读这两位时返回 UART0 的当前工作方式。

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

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

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

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

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

位 0： RI0：接收中断标志
当 UART0 接收到一个字节数据时（根据 SM20 位的选择）该位被硬件置 1。在 UART0 中断被允许时，置 1 该位将会使 CPU 转到 UART0 中断服务程序。该位必须用软件手动清 0。

图 21.8 SCON0：UART0 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
FE0	RXOV0	TXCOL0	SMOD0	S0TCLK1	S0TCLK0	S0RCLK1	S0RCLK0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x91
SFR 页： 0

位 7： FE0：帧错误标志。†
该位指示是否检测到无效（低电平）停止位。
0：未检测到帧错误。
1：检测到帧错误。

位 6： RXOV0：接收溢出标志。
该位指示新数据已经被锁存到接收缓冲器，而前一字节未被读取。
0：未检测到接收溢出。
1：检测到接收溢出。

位 5： TXCOL0：发送冲突标志。
该位指示发送过程正在进行时用户软件向 SBUF0 寄存器写入数据。
0：未检测到发送冲突。
1：检测到发送冲突。

位 4： SMOD0：UART0 波特率加倍使能。
该位使能/禁止 UART0 波特率逻辑的波特率 2 分频功能。
0：使能 UART0 的波特率 2 分频功能。
1：禁止 UART0 的波特率 2 分频功能。

位 3-2： UART0 发送波特率时钟选择位

S0TCLK1	S0TCLK0	串行发送波特率时钟源
0	0	定时器 1 溢出产生 UART0 发送波特率
0	1	定时器 2 溢出产生 UART0 发送波特率
1	0	定时器 3 溢出产生 UART0 发送波特率
1	1	定时器 4 溢出产生 UART0 发送波特率

位 1-0： UART0 接收波特率时钟选择位

S0RCLK1	S0RCLK0	串行接收波特率时钟源
0	0	定时器 1 溢出产生 UART0 接收波特率
0	1	定时器 2 溢出产生 UART0 接收波特率
1	0	定时器 3 溢出产生 UART0 接收波特率
1	1	定时器 4 溢出产生 UART0 接收波特率

†注：FE0、RXOV0 和 TXCOL0 只作为标志位使用，不用于产生中断。

图 21.9 SSTA0：UART0 状态和时钟选择寄存器

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

SFR 地址：0x99
SFR 页： 0

位 7-0： SBUF0.[7:0]：UART0 缓冲器位 7-0（MSB-LSB）
该 SFR 实际上是两个寄存器：一个发送缓冲器和一个接收缓冲器。当数据被写入 SBUF0 时，它进入发送移位寄存器等待串行发送。向 SBUF0 写入一个字节即启动发送过程。读 SBUF0 时返回数据来自接收缓冲器。

图 21.10 SBUF0： UART0 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：0xA9
SFR 页： 0

位 7-0： SADDR0.[7:0]：UART0 从地址
该寄存器的内容用于定义 UART0 的从地址。寄存器 SADEN0 中是地址掩码，它决定 SADDR0 中哪些位用于检查接收到的地址：与 SADEN0 中被置‘ 1 ’的那些位相对应的位被检查，与 SADEN0 中被置‘ 0 ’的那些位相对应的位被忽略。

图 21.11 SADDR0： UART0 从地址寄存器

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

SFR 地址：0xB9
SFR 页： 0

位 7-0： SADEN0.[7:0]：UART0 从地址使能
该寄存器中的位用于使能寄存器 SADDR0 中的对应位，以确定 UART0 的从地址。
0：SADDR0 中的对应位被视为“ 无关 ”。
1：SADDR0 中的对应位用于检查接收到的地址。

图 21.12 SADEN0： UART0 从地址使能寄存器

22. UART1

UART1 是一个异步、全双工串口，它提供标准 8051 串行口的方式 1 和方式 3。UART1 具有增强的波特率发生器电路，有多个时钟源可用于产生标准波特率。接收数据缓冲机制允许 UART1 在软件尚未读取前一个数据字节的情况下开始接收第二个输入数据字节。

UART1 有两个相关的特殊功能寄存器：串行控制寄存器（SCON1）和串行数据缓冲器（SBUF1）。用同一个 SBUF1 地址可以访问发送寄存器和接收寄存器。写 SBUF1 时自动访问发送寄存器；读 SBUF1 时自动访问接收寄存器。

如果 UART1 中断被允许，则每次发送完成（SCON1 中的 TI1 位被置‘1’）或接收到数据字节（SCON1 中的 RI1 位被置‘1’）时将产生中断。当 CPU 转向中断服务程序时硬件不清除 UART1 中断标志。中断标志必须用软件清除，这就允许软件查询 UART1 中断的原因（发送完成或接收完成）。

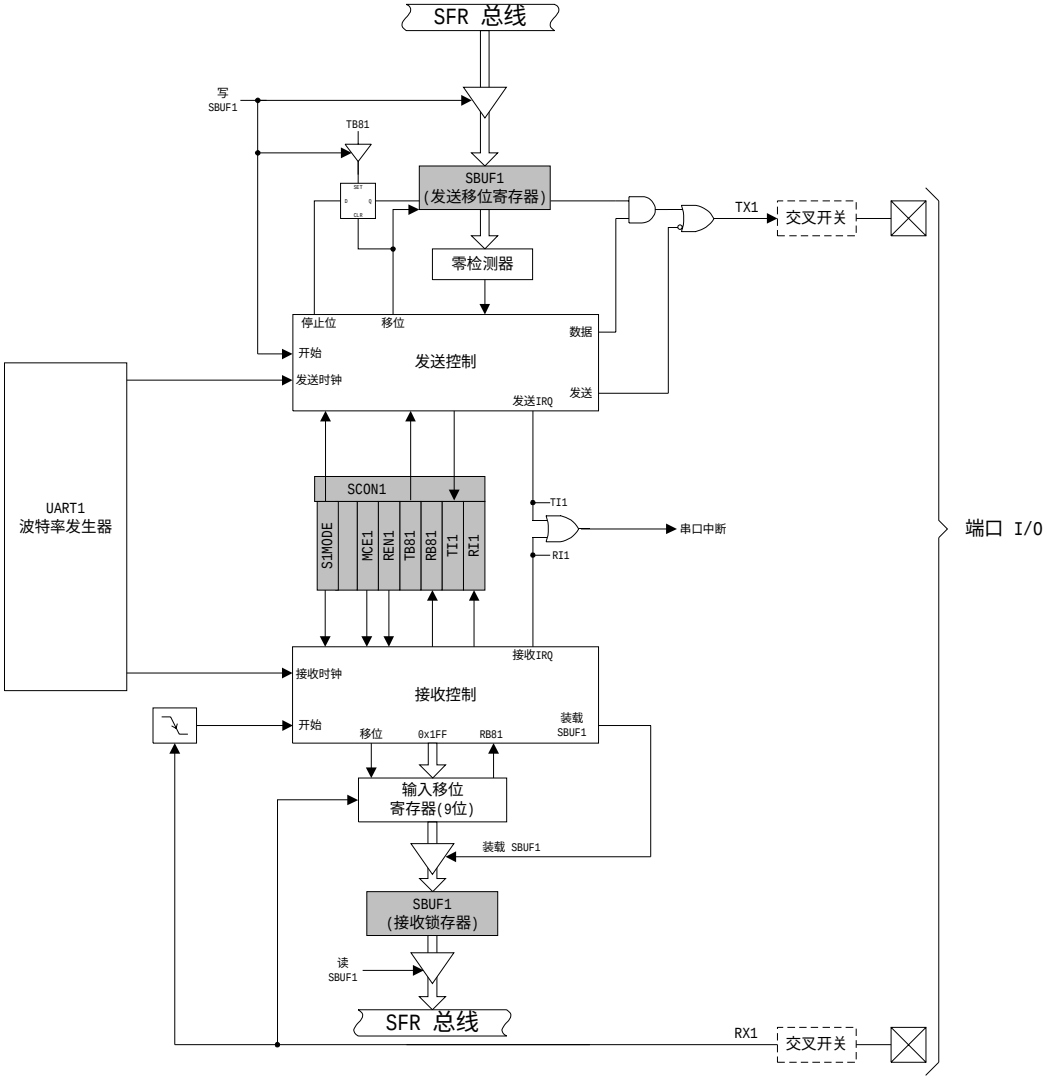


图 22.1　UART1 原理框图

22.1 增强的波特率发生器

UART1 波特率由定时器 1 工作在 8 位自动重装载方式产生。发送（TX）时钟由 TL1 产生；接收（RX）时钟由 TL1 的拷贝寄存器（图 22.2 中的 RX 定时器）产生，该寄存器不能被用户访问。TX 和 RX 定时器的溢出信号经过二分频后用于产生 TX 和 RX 波特率。当定时器 1 被允许时，RX 定时器运行并使用与定时器 1 相同的重载值（TH1）。在检测到 RX 引脚上的起始条件时 RX 定时器被强制重载，这允许在检测到起始位时立即开始接收过程，而与 TX 定时器的状态无关。

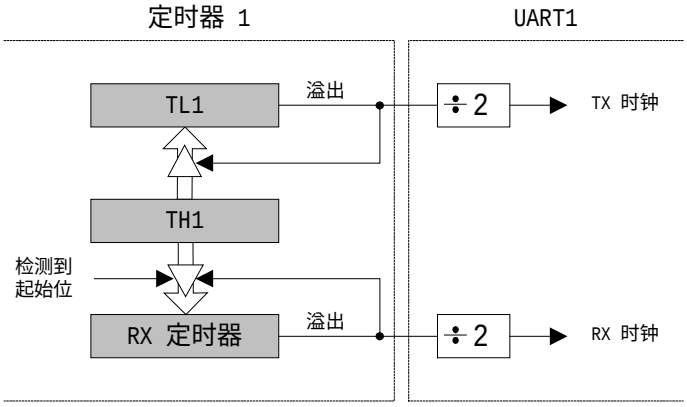


图 22.2 UART1 波特率逻辑

定时器 1 应被配置为方式 2，即 8 位自动重装载方式。定时器 1 的重载值应设置为使其溢出频率为所期望的波特率的两倍。注意，定时器 1 的时钟可以在 5 个时钟源中选择：SYSCLK、SYSCLK/4、SYSCLK/12、SYSCLK/48、外部振荡器时钟/8。对于任何给定的定时器 1 时钟源，UART0 的波特率由方程 22.1 决定：

方程 22.1 UART0 波特率

$$UART\text{波特率} = \frac{T1_{CLK}}{\left(256 - T1H\right)} \times \frac{1}{2}$$

其中 $T1_{CLK}$ 是定时器 1 的时钟频率，$T1H$ 是定时器 1 的高字节（重载值）。

定时器 1 时钟频率的选择方法见“ 23.1 定时器 0 和定时器 1 ”。表 22.1 – 22.5 给出了典型波特率和系统时钟频率的对照表。注意，当外部振荡器驱动定时器 1 时，内部振荡器或 PLL 仍可产生系统时钟。

22.2 工作方式

UART1 提供标准的异步、全双工通信，其工作方式（8 位或 9 位）通过 S1MODE 位（SCON1.7）来选择。典型的 UART 连接方式如图 22.3 所示。

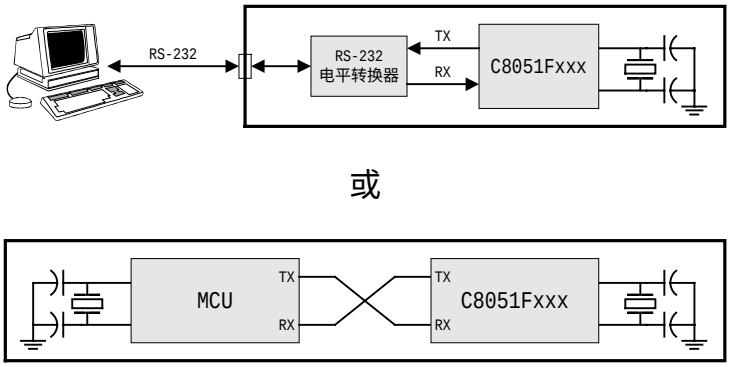


图 22.3　UART 连接图

22.2.1　8 位 UART

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

当软件向 SBUF1 寄存器写入一个字节时开始数据发送。在发送结束时（停止位开始）发送中断标志 TI1（SCON1.1）被置‘ 1 ’。在接收允许位 REN1（SCON1.4）被置‘ 1 ’后，数据接收可以在任何时刻开始。收到停止位后，如果满足下述条件则数据字节将被装入到接收寄存器 SBUF1：RI1 必须为逻辑‘ 0 ’；如果 MCE1 为逻辑‘ 1 ’，则停止位必须为‘ 1 ’。在发生接收数据溢出的情况下，先接收到的 8 位数据被锁存到 SBUF1 接收寄存器，而后面的溢出数据被丢弃。

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

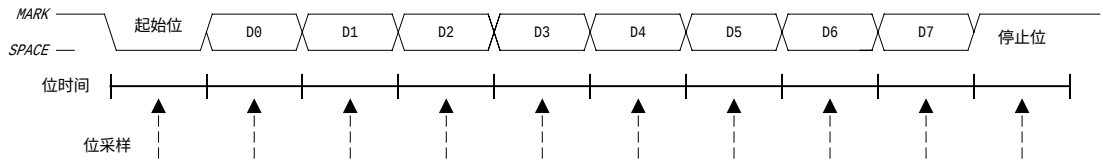


图 22.4　8 位 UART 时序图

22.2.2 9 位 UART

在 9 位 UART 方式，每个数据字节共使用 11 位：一个起始位、8 个数据位（LSB 在先）、一个可编程的第九位和一个停止位。在发送时，第九数据位由 TB81（SCON1.3）中的值决定,它可以被赋值为 PSW 中的奇偶位 P（用于错误检测），或用于多处理器通信。在接收时，第九数据位进入 RB81（SCON1.2），停止位被忽略。

当执行一条向 SBUF1 寄存器写一个数据字节的指令时开始数据发送。在发送结束时（停止位开始）发送中断标志 TI1 被置‘ 1 ’。在接收允许位 REN1 被置‘ 1 ’后，数据接收可以在任何时刻开始。收到停止位后如果满足下述条件则数据字节将被装入到接收寄存器 SBUF1：（1）RI1 必须为逻辑‘ 0 ’；（2）如果 MCE1 为逻辑‘ 1 ’，则第九位必须为逻辑‘ 1 ’（当 MCE1 为逻辑‘ 0 ’时，第九位数据的状态并不重要）。如果这些条件满足，则 8 位数据被存入 SBUF1，第九位被存入 RB81，RI1 标志被置位。如果这些条件不满足，则不装入 SBUF1 和 RB81，RI1 标志也不会被置‘ 1 ’。如果中断被允许，在 TI1 或 RI1 置位时将产生中断。

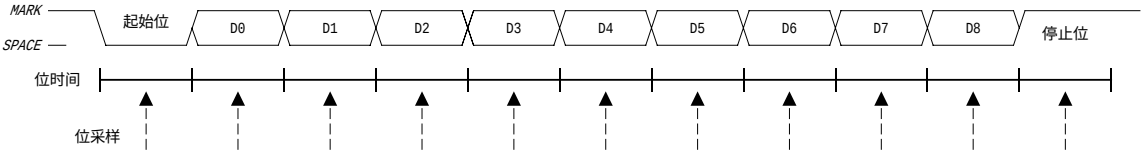


图 22.5 9 位 UART 时序图

22.3 多机通信

9 位 UART 方式通过使用第 9 数据位可以支持一个主处理器与一个或多个从处理器之间的多机通信。当主机要发送数据给一个或多个从机时，它先发送一个用于选择目标的地址字节。地址字节与数据字节的区别是：地址字节的第 9 位为逻辑‘ 1 ’；数据字节的第 9 位总是设置为逻辑‘ 0 ’。

如果从机的 MCE1 位（SCON1.5）被置‘ 1 ’，则只有当 UART 接收到的第九位为逻辑‘ 1 ’（RB81 = 1）并收到有效的停止位后 UART 才会产生中断。第 9 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 MCE1 位以允许后面接收数据字节时产生中断。未被寻址的从机仍保持其 MCE1 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 MCE1 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和/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。

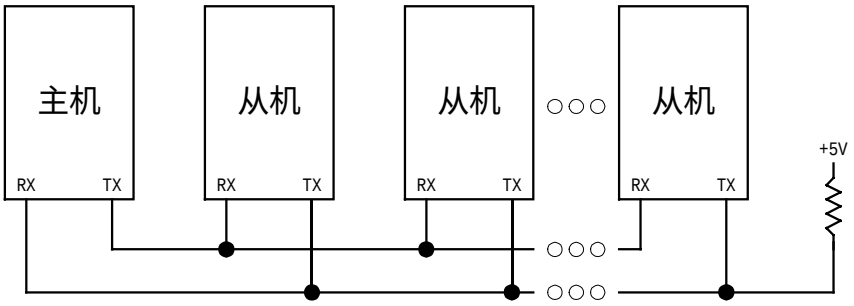


图 22.6　UART 多机方式连接图

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
S1MODE	-	MCE1	REN1	TB81	RB81	TI1	RI1	01000000
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：0x98
SFR 页： 1

位 7： S1MODE：UART1 工作方式选择位
该位选择 UART1 的工作方式。
0：方式 0：波特率可编程的 8 位 UART。
1：方式 1：波特率可编程的 9 位 UART。

位 6： 未使用。读 = 1b。写 = 忽略。

位 5： MCE1：多处理器通信使能
该位的功能取决于串行口工作方式。
方式 0：检查有效停止位。
0：停止位的逻辑电平被忽略。
1：只有当停止位为逻辑‘ 1 ’时 RI1 激活。
方式 1：多处理器通信使能位。
0：第 9 位的逻辑电平被忽略。
1：只有当第 9 位为逻辑‘ 1 ’时 RI1 才被置位并产生中断。

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

位 3： TB81：第 9 发送位
该位的逻辑电平被赋值给 9 位 UART 方式的第 9 发送位。在 8 位 UART 方式中未用。跟据需要用软件置‘ 1 ’或清‘ 0 ’。

位 2： RB81：第 9 接收位
该位被赋值为 9 位 UART 方式中第 9 数据位的值。在方式 0，则 RB8 被赋值为停止位的值。

位 1： TI1：发送中断标志
当 UART 发送完一个字节数据后该位被硬件置‘ 1 ’(在 8 位 UART 方式时，是在发送第 8 位后；在 9 位 UART 方式时，是在停止位开始)。当 UART1 中断被允许时，置‘ 1 ’该位将导致 CPU 转到 UART1 中断服务程序。该位必须用软件清‘ 0 ’。

位 0： RI1：接收中断标志
当 UART1 接收到一个字节数据时该位被硬件置‘ 1 ’(在停止位采样后)。当 UART1 中断被允许时，置‘ 1 ’该位将会使 CPU 转到 UART1 中断服务程序。该位必须用软件清‘ 0 ’。

图 22.7 SCON1： UART1 控制寄存器

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

SFR 地址：0x99
SFR 页： 1

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

图 22.8 SBUF1：UART1 串行数据缓冲寄存器

表 22.1 对应标准波特率的定时器设置（使用内部振荡器）

	频率：24.5MHz						
	目标波特率（bps）	波特率误差（%）	振荡器分频系数	定时器时钟源	SCA1-SCA0（预分频选择）	T1M	定时器 1 重载值（hex）
SYSCLK 源自内部振荡器	230400	-0.32%	106	SYSCLK	XX	1	0xCB
	115200	-0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	-0.32%	848	SYSCLK/4	01	0	0x96
	14400	0.15%	1704	SYSCLK/12	00	0	0xB9
	9600	-0.32%	2544	SYSCLK/12	00	0	0x96
	2400	-0.32%	10176	SYSCLK/48	10	0	0x96
	1200	0.15%	20448	SYSCLK/48	10	0	0x2B

X = 忽略
†SCA1-SCA0 和 T1M 位的定义见 23.1 节。

表 22.2 对应标准波特率的定时器设置（使用外部振荡器）

	频率：25.0 MHz						
	目标波特率（bps）	波特率误差（%）	振荡器分频系数	定时器时钟源	SCA1-SCA0† （预分频选择）	T1M†	定时器 1 重载值（hex）
SYSCLK 源自外部振荡器	230400	-0.47%	108	SYSCLK	XX	1	0xCA
	115200	0.45%	218	SYSCLK	XX	1	0x93
	57600	-0.01%	434	SYSCLK	XX	1	0x27
	28800	0.45%	872	SYSCLK/4	01	0	0x93
	14400	-0.01%	1736	SYSCLK/4	01	0	0x27
	9600	0.15%	2608	EXTCLK/8	11	0	0x5D
	2400	0.45%	10464	SYSCLK/48	10	0	0x93
	1200	-0.01%	20832	SYSCLK/48	10	0	0x27
SYSCLK 源自内部振荡器	57600	-0.47%	432	EXTCLK/8	11	0	0xE5
	28800	-0.47%	864	EXTCLK/8	11	0	0xCA
	14400	0.45%	1744	EXTCLK/8	11	0	0x93
	9600	0.15%	2608	EXTCLK/8	11	0	0x5D

X = 忽略

†SCA1-SCA0 和 T1M 位的定义见 23.1 节。

表 22.3 对应标准波特率的定时器设置（使用外部振荡器）

	频率：22.1184MHz						
	目标波特率（bps）	波特率误差（%）	振荡器分频系数	定时器时钟源	SCA1-SCA0† （预分频选择）	T1M†	定时器 1 重载值（hex）
SYSCLK 源自外部振荡器	230400	0.00%	96	SYSCLK	XX	1	0xD0
	115200	0.00%	192	SYSCLK	XX	1	0xA0
	57600	0.00%	384	SYSCLK	XX	1	0x40
	28800	0.00%	768	SYSCLK/12	00	0	0xE0
	14400	0.00%	1536	SYSCLK/12	00	0	0xC0
	9600	0.00%	2304	SYSCLK/12	00	0	0xA0
	2400	0.00%	9216	SYSCLK/48	10	0	0xA0
	1200	0.00%	18432	SYSCLK/48	10	0	0x40
SYSCLK 源自内部振荡器	230400	0.00%	96	EXTCLK/8	11	0	0xFA
	115200	0.00%	192	EXTCLK/8	11	0	0xF4
	57600	0.00%	384	EXTCLK/8	11	0	0xE8
	28800	0.00%	768	EXTCLK/8	11	0	0xD0
	14400	0.00%	1536	EXTCLK/8	11	0	0xA0
	9600	0.00%	2304	EXTCLK/8	11	0	0x70

X = 忽略

†SCA1-SCA0 和 T1M 位的定义见 23.1 节。

表 22.4 对应标准波特率的定时器设置（使用 PLL）

频率：50MHz						
目标波特率（bps）	波特率误差（%）	振荡器分频系数	定时器时钟源	SCA1-SCA0† (预分频选择)	T1M†	定时器 1 重载值（hex）
230400	0.45%	218	SYSCLK	XX	1	0x93
115200	-0.01%	434	SYSCLK	XX	1	0x27
57600	0.45%	872	SYSCLK/4	01	0	0x93
28800	-0.01%	1736	SYSCLK/4	01	0	0x27
14400	0.22%	3480	SYSCLK/12	00	0	0x6F
9600	-0.01%	5208	SYSCLK/12	00	0	0x27
2400	-0.01%	20832	SYSCLK/48	10	0	0x27

X = 忽略

†SCA1-SCA0 和 T1M 位的定义见 23.1 节。

表 22.5 对应标准波特率的定时器设置（使用 PLL）

频率：100MHz						
目标波特率（bps）	波特率误差（%）	振荡器分频系数	定时器时钟源	SCA1-SCA0† (预分频选择)	T1M†	定时器 1 重载值（hex）
230400	-0.01%	434	SYSCLK	XX	1	0x27
115200	0.45%	872	SYSCLK/4	01	0	0x93
57600	-0.01%	1736	SYSCLK/4	01	0	0x27
28800	0.22%	3480	SYSCLK/12	00	0	0x6F
14400	-0.47%	6912	SYSCLK/48	10	0	0xB8
9600	0.45%	10464	SYSCLK/48	10	0	0x93

X = 忽略

†SCA1-SCA0 和 T1M 位的定义见 23.1 节。

23. 定时器

C8051F12x 和 C8051F13x MCU 内部有 5 个计数器/定时器 其中定时器 0 和定时器 1 与标准 8051 中的计数器/定时器兼容。定时器 2、定时器 3 和定时器 4 是 16 位自动重装载并具有捕捉功能的定时器，可用于 ADC、DAC、方波发生器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 3 具有自动重装载和捕捉功能。定时器 2 和定时器 4 完全相同，不但提供了自动重装载和捕捉功能，还具有在外部端口引脚上产生 50%占空比的方波的能力（电平切换输出）。

定时器 0 和定时器 1 方式	定时器 2、3、4 方式
13 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 位计数器/定时器
8 位自动重装载的计数器/定时器	电平切换输出（只限于定时器 2 和 4）
两个 8 位计数器/定时器（只限于定时器 0）	

定时器 0 和定时器 1 的时钟可以在 5 个时钟源中选择，由定时器方式选择位（T1M-T0M）和时钟预分频位（SCA1-SCA0）决定。时钟预分频位为定时器 0 和/或定时器 1 定义一个预分频时钟（见图 23.6）。定时器 0/1 可以被配置为使用这个预分频时钟或系统时钟。定时器 2、3、4 可以使用系统时钟、系统时钟/12 或外部振荡器时钟/8。

定时器 0 和定时器 1 还可以作为计数器使用。当作为计数器使用时，在所分配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。

23.1 定时器 0 和定时器 1

每个计数器/定时器都是一个 16 位的寄存器，在被访问时以两个字节的形式出现：一个低字节（TL0 或 TL1）和一个高字节（TH0 或 TH1）。计数器/定时器控制寄存器（TCON）用于允许定时器 0 和定时器 1 以及指示它们的状态。通过设置 IE 寄存器中的 ET0 位使能定时器 0 中断，通过设置 IE 寄存器中的 ET1 位使能定时器 1 中断（见“ 11.7.5 中断寄存器说明 ”）。这两个计数器/定时器都有四种工作方式，通过设置计数器/定时器方式寄存器（TMOD）中的方式选择位 T1M1-T0M0 来选择工作方式。每个定时器都可以被独立编程。

23.1.1 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。这两个定时器在工作上完全相同，定时器 1 的配置过程与定时器 0 一样。

TH0 寄存器保持 13 位计数器/定时器的 8 个 MSB。TL0 在 TL0.4-TL0.0 位置保持 5 个 LSB。TL0 的高 3 位（TL0.7-TL0.5）是不确定的，在读计数值时应屏蔽掉或忽略这 3 位。作为 13 位定时器寄存器，计到 0x1FFF（全 1）后再计一次将发生溢出，使计数值回到 0x0000，此时定时器溢出标志 TF0（TCON.5）被置位并产生一个中断（如果被允许）。

C/T0 位（TMOD.2）选择计数器/定时器的时钟源。当 C/T0 被设置为逻辑 1 时，出现在所选输入引脚（T0）上的负跳变使定时器寄存器加 1。清除 C/T0 将选择 T0M 位（CKCON.3）定义的时钟。有关选择和配置外部 I/O 引脚的详细信息见“ 18.1 端口 0 - 端口 3 和优先级交叉开关译码器 ”。当 T0M 被置 1 时，定时器 0 使用系统时钟；当 T0M 被清 0 时，定时器 0 使用由 CKCON（见图 23.6）中的时钟预分频位所选择的时钟源。

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

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

X=任意

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

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

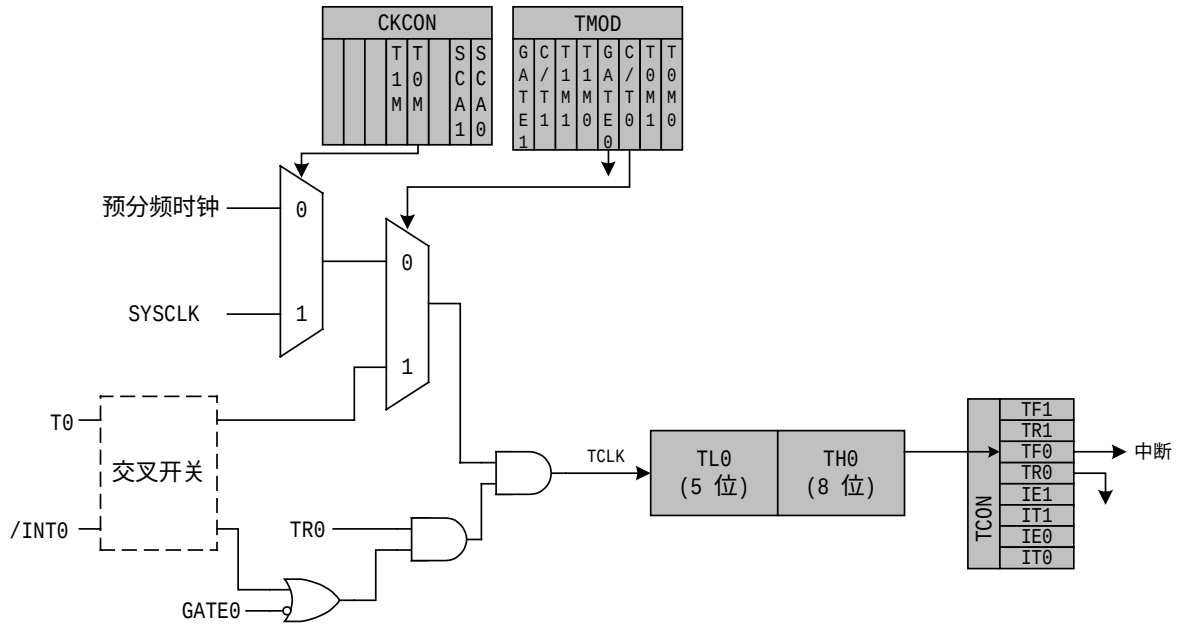


图 23.1　T0 方式 0 原理框图

23.1.2 方式 1：16 位计数器/定时器

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

23.1.3 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 0xFF 到 0x00）时，定时器溢出标志 TF0（TCON.5）被置位，TH0 中的重载值被重新装入到 TL0。如果中断被允许，在 TF0 被置位时将产生一个中断。TH0 中的重载值保持不变。为了保证第一次计数正确，必须在允许定时器之前将 TL0 初始化为所希望的计数初值。当工作于方式 2 时，定时器 1 的操作与定时器 0 完全相同。

在方式 2，两个定时器的允许和配置方法与方式 0 一样。当 GATE0（TMOD.3）为逻辑 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。

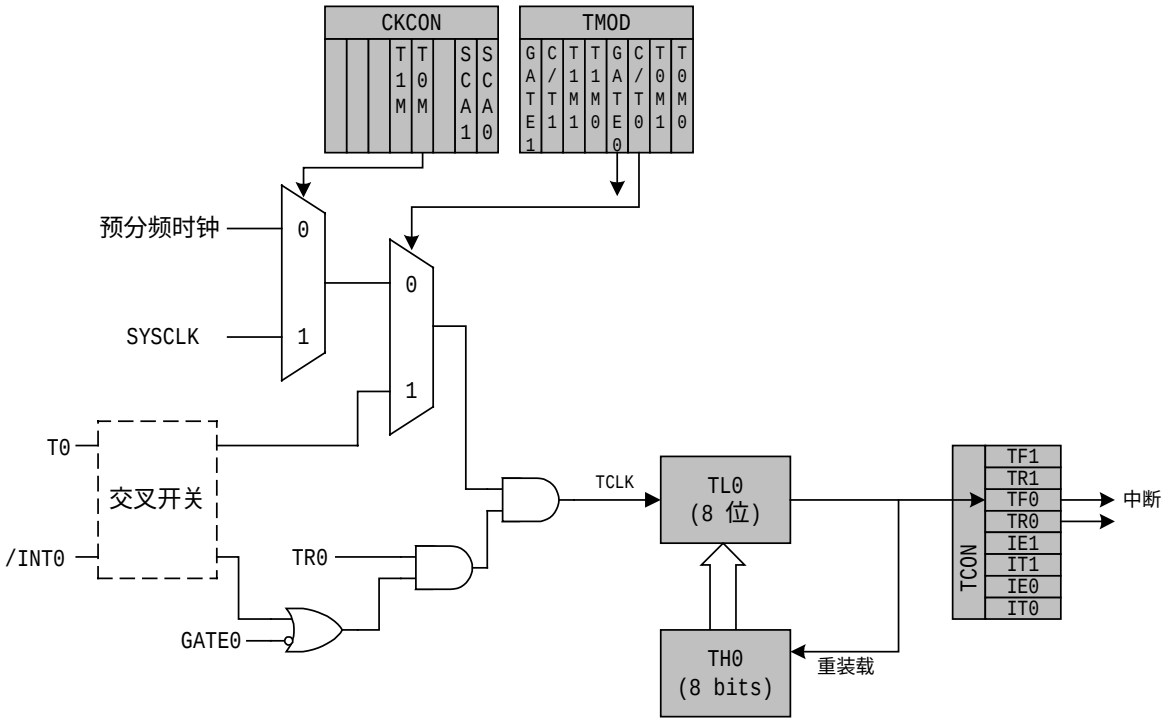


图 23.2　T0 方式 2（8 位重装载）原理框图

23.1.4 方式 3：两个 8 位计数器/定时器（仅定时器 0）

在方式 3 时，定时器 0 被配置为两个独立的 8 位定时器/计数器，计数值在 TL0 和 TH0 中。在 TL0 中的计数器/定时器使用 TCON 和 TMOD 中定时器 0 的控制/状态位：TR0、C/T0、GATE0 和 TF0。TL0 既可以使用系统时钟也可以使用一个外部输入信号作为时间基准。TH0 寄存器只能作为定时器使用，由系统时钟或预分频时钟提供时间基准。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，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 SMBus 和/或 UART 的波特率时钟，和/或启动 ADC 转换。要在定时器 0 工作于方式 3 时使用定时器 1，应将定时器 1 工作方式设置为 0、1 或 2。要禁止定时器 1，可将其配置为方式 3。

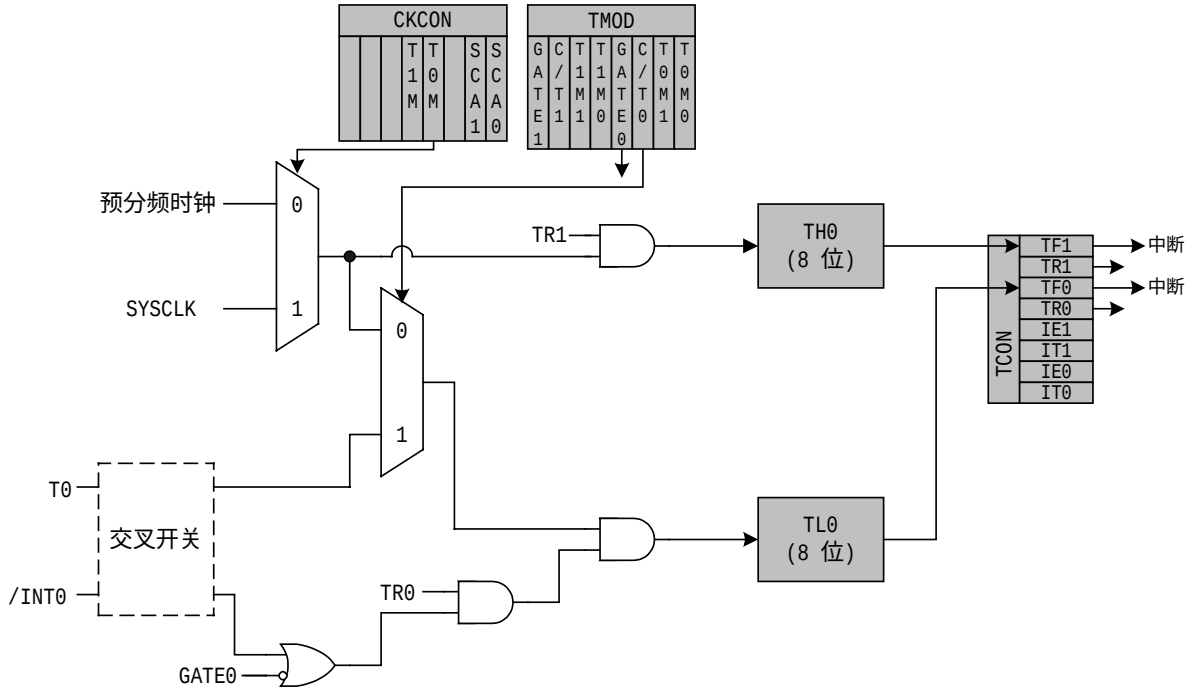


图 23.3　T0 方式 3（两个 8 位定时器）原理框图

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
SFR 页： 0

位 7： TF1：定时器 1 溢出标志
当定时器 1 溢出时由硬件置位。该位可以用软件清 0，但当 CPU 转向定时器 1 中断服务程序时该位被自动清 0。
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，低电平有效。
1：/INT1 为边沿触发，下降沿有效。

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，低电平有效。
1：/INT0 为边沿触发，下降沿有效。

图 23.4 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
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
SFR 页： 0

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 = 1 时定时器 1 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 = 1 时定时器 1 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 23.5 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
-	-	-	T1M	T0M	-	SCA1	SCA0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x8E
SFR 页： 0

位 7-5： 未用。读=000b，写=忽略。
位 4： T1M：定时器 1 时钟选择。
该位选择定时器 1 的时钟源。当 C/T1 被置‘ 1 ’时，该位被忽略。
0：定时器 1 使用预分频位 SCA1-SCA0 定义的时钟。
1：定时器 1 使用系统时钟
位 3： T0M：定时器 0 时钟选择。
该位选择定时器 0 的时钟源。当 C/T0 被置‘ 1 ’时，该位被忽略。
0：计数器/定时器使用预分频位 SCA1-SCA0 定义的时钟。
1：计数器/定时器使用系统时钟

位 2： 保留。读=0b，写=忽略。
位 1-0： SCA1-SCA0：定时器 1/0 预分频位。
如果定时器 0 和/或定时器 1 被配置为使用预分频时钟输入，则这两位控制预分频时钟的分频系数。

SCA1	SCA0	预分频时钟
0	0	系统时钟/12
0	1	系统时钟/4
1	0	系统时钟/48
1	1	外部时钟/8†

†注：外部时钟/8 与系统时钟同步，定时器工作在该方式时外部时钟必须小于或等于系统时钟频率。

图 23.6 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
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：0x8A
SFR 页： 0

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 23.7 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：0x8B
SFR 页： 0

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 23.8 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：0x8C
SFR 页： 0

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 23.9 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：0x8D
SFR 页： 0

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。

图 23.10 TH1：定时器 1 高字节

23.2 定时器 2、定时器 3 和定时器 4

定时器 2、定时器 3 和定时器 4 是 16 位的计数器/定时器，每个定时器由两个 8 位的 SFR 组成：TMRnL（低字节）和 TMRnH（高字节），其中 n = 2、3 或 4。定时器 2 和定时器 4 具有自动重装载、捕捉和电平切换输出功能，可以向上或向下计数。定时器 3 具有自动重装载和捕捉功能，也可以向上或向下计数。用定时器 2、3 或 4 控制寄存器（TMRnCN）中的位选择捕捉方式和自动重装载方式。用定时器 2 或 4 配置寄存器（TMRnCF）中的位选择电平切换输出方式。这些定时器还可以在外部引脚上产生方波信号。与定时器 0 和定时器 1 一样，定时器 2、3 和 4 可以使用系统时钟（1、2 或 12 分频）、外部时钟（8 分频）或外部输入引脚上的跳变作为时钟源。计数器/定时器选择位 C/Tn（TMRnCN.1）将定时器配置为计数器方式或定时器方式。清除 C/Tn 将定时器配置为定时器方式（即系统时钟或外部输入引脚上的跳变作为定时器的输入）。当 C/Tn 位被置‘ 1 ’时，将定时器配置为计数器方式（即在 Tn 输入引脚上的负跳变使计数器/定时器的寄存器加 1 或减 1）。定时器 3 和定时器 2 共享 T2 输入引脚。有关为外设选择和配置外部 I/O 引脚的详细信息见“ 18.1 端口 0 – 端口 3 和优先级交叉开关译码器 ”。定时器 2 和定时器 3 可用于启动 ADC 数据转换，定时器 2、定时器 3 和定时器 4 可用于更新 DAC 输出。只有定时器 1 能用于为 UART1 产生波特率，定时器 1、定时器 2、定时器 3 和定时器 4 均可用于为 UART0 产生波特率。

当定时器 2、定时器 3 和定时器 4 工作在捕捉模式时，可以在下列时钟源中选择时钟：SYSCK、SYSCLK/2、SYSCLK/12、外部时钟/8、Tn 输入引脚上的负跳变。清除 C/Tn 位（TMRnCN.1）选择系统时钟/外部时钟作为定时器的输入。可以用 TMRnCF 中的时钟选择位 TnM0 和 TnM1 在系统时钟、系统时钟/12 或 XTAL1/XTAL2 引脚上的外部时钟/8 中选择（见图 23.14）。当 C/Tn 位被置‘ 1 ’时，Tn 输入引脚上的负跳变使计数器/定时器的寄存器加 1 或减 1（即配置为计数器方式）。

23.2.1 配置定时器 2、3 和 4 向下计数

定时器 2、定时器 3 和定时器 4 具有向下计数的能力。当定时器配置寄存器（见图 23.14）中的减 1 使能位（DCEN）被置‘ 1 ’时，定时器可以向上或向下计数。当 DCEN = 1 时，定时器的计数方向受 TnEX 引脚上的逻辑电平的控制（定时器 3 与定时器 2 共享 T2EX 引脚）。当 TnEX = 1 时，计数器/定时器向上计数；当 TnEX = 0 时，计数器/定时器向下计数。如果要使用该功能，TnEX 必须在数字交叉开关中被使能并且被配置为数字输入。

注：当 DCEN = 1 时，TnEX 输入的其它功能（即捕捉和重装载）不可用。当 DCEN =1 时，TnEX 只控制定时器的计数方向。

23.2.2 捕捉方式

在捕捉方式，定时器 2、3 和 4 被作为具有捕捉能力的 16 位计数器/定时器使用。当定时器外部使能位（在 TMRnCN 中）被置‘ 1 ’时，TnEX 输入引脚（定时器 3 与定时器 2 共享 T2EX 引脚）上的负跳变导致相应定时器中的 16 位计数值(THn,TLn)被装入到捕捉寄存器(RCAPnH,RCAPnL)。如果捕捉被触发，则定时器外部标志外（TMRnCN.6）被置‘ 1 ’，并产生中断（如果中断被允许)。有关中断源配置的详细信息见“ 11.7 中断系统 ”。

当 16 位的定时器寄存器（TMRnH:TMRnL）加 1 后发生上溢时，定时器上溢/下溢标志 TFn（TMRnCN.7）被置‘ 1 ’，并产生中断（如果中断被允许)。通过将减 1 使能位（TMRnCF.0）置‘ 1 ’，可以将定时器配置为向下计数。在这种情况下，定时器在每个时钟/计数事件到来时进行减 1 计数，并在发生从 0x0000 到 0xFFFF 的变化时产生下溢。与上溢时一样，上溢/下溢标志 TFn 被置‘ 1 ’，并产生中断（如果中断被允许)。

通过置‘ 1 ’捕捉/重装选择位 CP/RLn（TMRnCN.0）和定时器的运行控制位 TRn（TMRnCN.2）来选择带捕捉的计数器/定时器方式。定时器的外部使能位 EXENn（TMRnCN.3）也必须被设置为逻辑 1 以使能捕捉方式。如果 EXENn 被清除，TnEX 上的电平变化将被忽略。

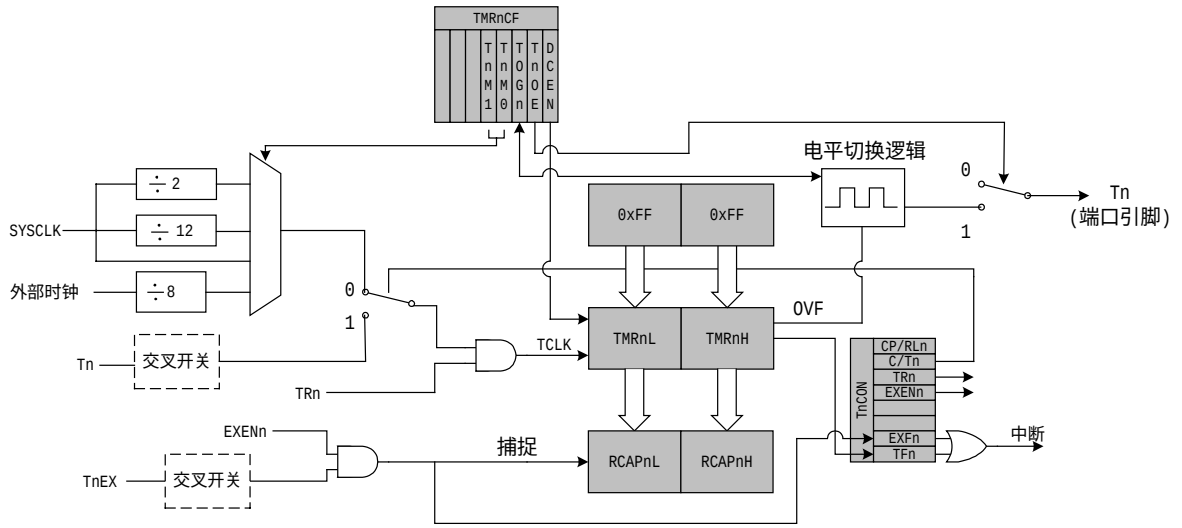


图 23.11 T2、3 和 4 捕捉方式原理框图

23.2.3 自动重装载方式

在自动重装载方式，计数器/定时器可以被配置为向上计数或向下计数，并在发生上溢/下溢事件时产生中断。当向上计数并发生溢出时，计数器/定时器将其上溢/下溢标志（TFn）置‘ 1 ’并产生中断（如果中断被允许），同时重载/捕捉寄存器（RCAPnH 和 RCAPnL）中的值被装入到定时器寄存器，定时器重新开始计数。当定时器的外部使能位（EXENn）被置‘ 1 ’并且 减 1 使能位（DCEN）被清‘ 0 ’时，TnEX 引脚上的下降沿将导致定时器重装载。注意，定时器溢出也会导致自动重装载。当 DCEN 被置‘ 1 ’时，TnEX 引脚的状态控制计数器/定时器是向上计数（加 1）还是向下计数（减 1），不会导致自动重装载或产生中断（定时器 3 与定时器 2 共享 T2EX 引脚）。有关配置定时器向下计数的详细信息见 23.2.1 节。

当向下计数并且 TMRnH 和 TMRnL 寄存器中的值与重载/捕捉寄存器（RCAPnH 和 RCAPnL）中的值相等时，计数器/定时器将其上溢/下溢标志（TFn）置‘ 1 ’并产生中断（如果中断被允许）。这被认为是一次下溢事件，该事件使定时器重新装载 0xFFFF。当发生下溢时，定时器自动重新开始计算数。

清除 CP/RLn 位将选择计数器/定时器的自动重装载方式。设置 TRn 为逻辑 1 允许并启动定时器。

在自动重装载方式，外部标志（EXFn）在每次发生上溢或下溢时改变电平，但不引起中断。EXFn 标志可以被用作 17 位计数器的最高位（MSB）。

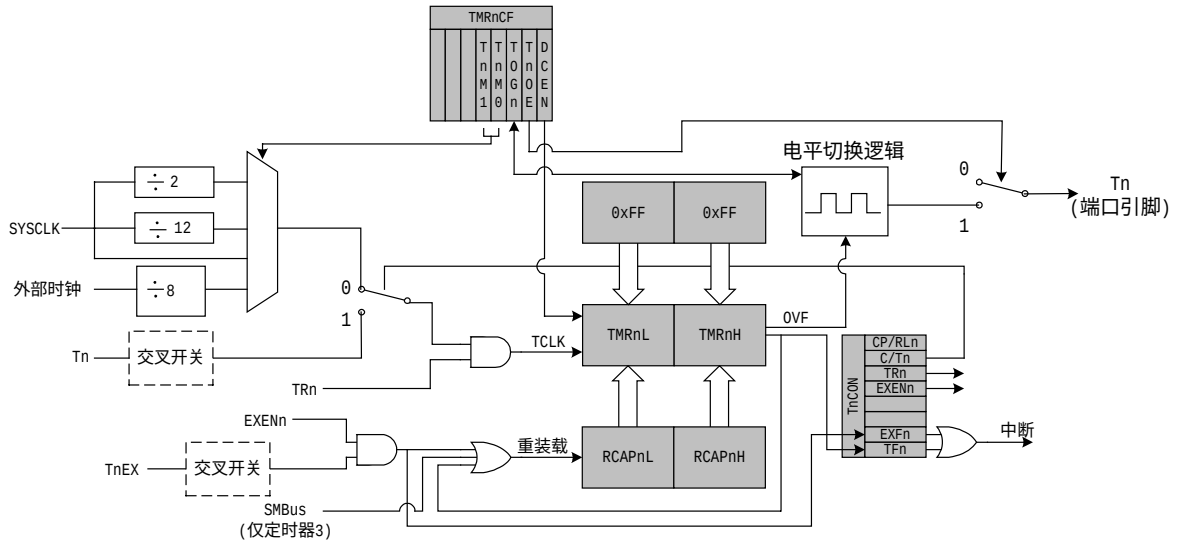


图 23.12　T2、3 和 4 自动重装载方式原理框图

23.2.4 电平切换方式（仅限于定时器 2 和定时器 4）

定时器 2 和定时器 4 具有切换对应输出引脚（T2 或 T4）电平的能力，可以产生 50%占空比的输出波形。定时器每发生一次上溢或下溢（取决于定时器是向上计数还是向下计数）时，对应引脚的输出状态发生改变。切换频率由定时器的时钟源和 RCAPnH:RCAPnL 中的值决定。当向下计数时，定时器的自动重装载值为 0xFFFF，下溢发生在定时器的值与 RCAPnH:RCAPnL 中的值相等时。当向上计数时，定时器的自动重装载值为 RCAPnH:RCAPnL，上溢发生在定时器的值从 0xFFFF 变化到重载值时。

为了输出一个方波，将定时器配置为自动重装载方式（TMRnCN 中的捕捉/重装选择位和定时器/计数器选择位均被清‘0’）。通过将 TMRnCF 中的定时器输出使能位设置为‘1’来使能定时器输出功能。应正确配置定时器时钟源和重装/下溢值，使定时器的上溢/下溢频率为期望输出频率的 2 倍。由交叉开关分配的定时器输出引脚应被配置为数字输出（见“18. 端口输入/输出”）。设置定时器的运行控制位为‘1’将启动引脚电平切换。读定时器的切换输出状态位（TMRnCF.2）返回切换输出的状态，写定时器的切换输出状态位将产生强制输出。这在希望引脚从一个已知状态开始切换或在切换被停止时强制引脚进入所期望的状态时是很有用的。

方程 23.1 方波频率（仅限于定时器 2 和定时器 4）

$$F_{sq} = \frac{F_{TCLK}}{2 \times (65536 - RCAPn)}$$

方程 23.1 适用于定时器被配置为向上或向下计数两种情况。

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
TFn	EXFn	-	-	EXENn	TRn	C/Tn	CP/RLn	00000000
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：TMR2CN:0xC8; TMR3CN:0xC8; TMR4CN:0xC8
SFR 页： TMR2CN:页 0; TMR3CN: 页 1; TMR4CN: 页 2

位 7： TFn：定时器 2、3 和 4 上溢/下溢标志
当定时器发生上溢（从 0xFFFF 到 0x0000）、下溢（从 RCAPnH:RCAPnL 中的值到 0xFFFF，自动重装载方式）或下溢（从 0x0000 到 0xFFFF，捕捉方式）时由硬件置位。当定时器中断被允许时，该位置 1 导致 CPU 转向定时器的中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 6： EXFn：定时器 2、3 和 4 外部标志
当 TnEX 输入引脚的负跳变导致发生捕捉或重装载并且 EXENn 为逻辑 1 时，该位由硬件置位。在定时器中断被允许时，该位置‘ 1 ’使 CPU 转向定时器的中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 5-4： 保留。

位 3： EXENn：定时器 2、3 和 4 外部使能
使能 TnEX 上的负跳变触发捕捉、重载或控制计数器/定时器的计数方向（向上或向下计数）。如果 DCENn = 1，TnEX 决定定时器是向上计数还是向下计数（在自动重装载方式）。如果 EXENn = 1，TnEX 应被配置为数字输入。
0：TnEX 上的跳变被忽略。
1：TnEX 上的跳变导致一次捕捉、重装载或控制计数器/定时器的计数方向（向上或向下计数），具体如下：
捕捉方式：TnEX 上的负跳变导致 RCAPnH:RCAPnL 捕捉定时器的值。
自动重装载方式：
DCEN = 0：TnEX 上的负跳变导致定时器重装载并置位 EXFn 标志。
DCEN = 1：TnEX 上的逻辑电平控制定时器的计数方向（向上或向下）。

位 2： TRn：定时器 2、3 和 4 运行控制
该位允许/禁止相应的定时器。
0：定时器禁止。
1：定时器允许并运行/计数。

位 1： C/Tn：计数器/定时器功能选择
0：定时器功能：定时器由 TnM1:TnM0（TMRnCF.4: TMRnCF.3）定义的时钟加 1。
1：计数器功能：定时器由外部输入引脚的负跳变加 1。

位 0： CP/RLn：捕捉/重载选择
该位选择定时器工作在捕捉方式还是自动重装载方式。
0：定时器工作在自动重装载方式。
1：定时器工作在捕捉方式。

注：定时器 3 和定时器 2 共享 T2 和 T2EX 引脚。

图 23.13 TMRnCN：定时器 2、3 和 4 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	TnM1	TnM0	TOGn	TnOE	DCEN	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：TMR2CF:0xC9; TMR3CF:0xC9; TMR4CF:0xC9
SFR 页： TMR2CF:页 0; TMR3CF: 页 1; TMR4CF: 页 2

位 7-5： 保留。

位 4-3： TnM1 和 TnM0：定时器时钟方式选择位
这些位用于选择定时器的时钟源。时钟源可以是：系统时钟（SYSCLK）、SYSCLK/2、SYSCLK/12 或外部时钟/8。时钟源选择如下：
00：SYSCLK/12
01：SYSCLK
10：外部时钟/8（与系统时钟同步）
11：SYSCLK/2

位 2： TOGn：切换输出状态位
当定时器被用于切换一个端口引脚电平时，该位用于读引脚输出状态，或向该位写时产生强制输出（仅限于定时器 2 和定时器 4）。

位 1： TnOE：定时器输出使能位
该位使能定时器在对应外部端口引脚上输出 50%占空比信号的功能。
注：定时器按如下配置产生方波输出：
CP/RLn = 0
C/Tn = 0
TnOE = 1
装载 RCAPnH:RCAPnL。
配置端口引脚以输出方波。
0：电平切换输出在为定时器被分配的端口引脚不可用。
1：电平切换输出在为定时器被分配的端口引脚可用。

位 0： DCEN：减计数使能位
该位使能定时器的向上或向下计数功能，计数方向由 TnEX 的状态决定。
0：定时器向上计数，与 TnEX 的状态无关。
1：定时器可以向上计数，也可以向下计数，取决于 TnEX 的状态。
如果 TnEX = 0，定时器向下计数。
如果 TnEX = 1，定时器向上计数

注：定时器 3 和定时器 2 共享 T2 和 T2EX 引脚。

图 23.14 TMRnCF：定时器 2、3 和 4 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：RCAP2L:0xCA; RCAP3L:0xCA; RCAP4L:0xCA
SFR 页： RCAP2L:页 0; RCAP3L: 页 1; RCAP4L: 页 2

位 7-0： RCAPnL：定时器 2、3 和 4 捕捉寄存器的低字节
当定时器被配置为捕捉方式时，RCAPnL 寄存器捕捉定时器的低字节。当定时器被配置为自动重装载方式时，它保存重载值的低字节。

图 23.15 RCAPnL：定时器 2、3 和 4 捕捉寄存器低字节

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

SFR 地址：RCAP2H:0xCB; RCAP3H:0xCB; RCAP4H:0xCB
SFR 页： RCAP2H:页 0; RCAP3H: 页 1; RCAP4H: 页 2

位 7-0： RCAPnH：定时器 2、3 和 4 捕捉寄存器的高字节
当定时器被配置为捕捉方式时，RCAPnH 寄存器捕捉定时器的高字节。当定时器被配置为自动重装载方式时，它保存重载值的高字节。

图 23.16 RCAPnH：定时器 2、3 和 4 捕捉寄存器高字节

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

SFR 地址：TMR2L:0xCC; TMR3L:0xCC; TMR4L:0xCC
SFR 页： TMR2L:页 0; TMR3L: 页 1; TMR4L: 页 2

位 7-0： TMRnL：定时器 2、3 和 4 的低字节
TMRnL 寄存器保存 16 位定时器 2、3 和 4 的低字节。

图 23.17 TMRnL：定时器 2、3 和 4 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：TMR2H:0xCD; TMR3H:0xCD; TMR4H:0xCD
SFR 页： TMR2H:页 0; TMR3H: 页 1; TMR4H: 页 2

位 7-0： TMRnH：定时器 2、3 和 4 的高字节
TMRnH 寄存器保存 16 位定时器 2、3 和 4 的高字节。

图 23.18 TMRnH：定时器 2、3 和 4 高字节

24. 可编程计数器阵列

可编程计数器阵列（PCA0）提供增强的定时器功能，与标准8051计数器/定时器相比，它需要较少的CPU干预。PCA0包含一个专用的16位计数器/定时器和6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（见“ 18.1 端口0 – 端口3和优先级交叉开关译码器 ”）。计数器/定时器由一个可编程的时基信号驱动，时基信号有六个输入源：系统时钟、系统时钟/4、系统时钟/12、外部振荡器时钟源8分频、定时器0溢出、ECI线上的外部时钟信号。每个捕捉/比较模块可以被编程为独立工作在下面的6种工作方式之一：边沿触发捕捉、软件定时器、高速输出、频率输出、8位PWM或16位PWM（24.2节对每种方式进行说明）。对PCA的编程和控制是通过系统控制器的特殊功能寄存器来实现的。PCA的基本原理框图示于图24.1。

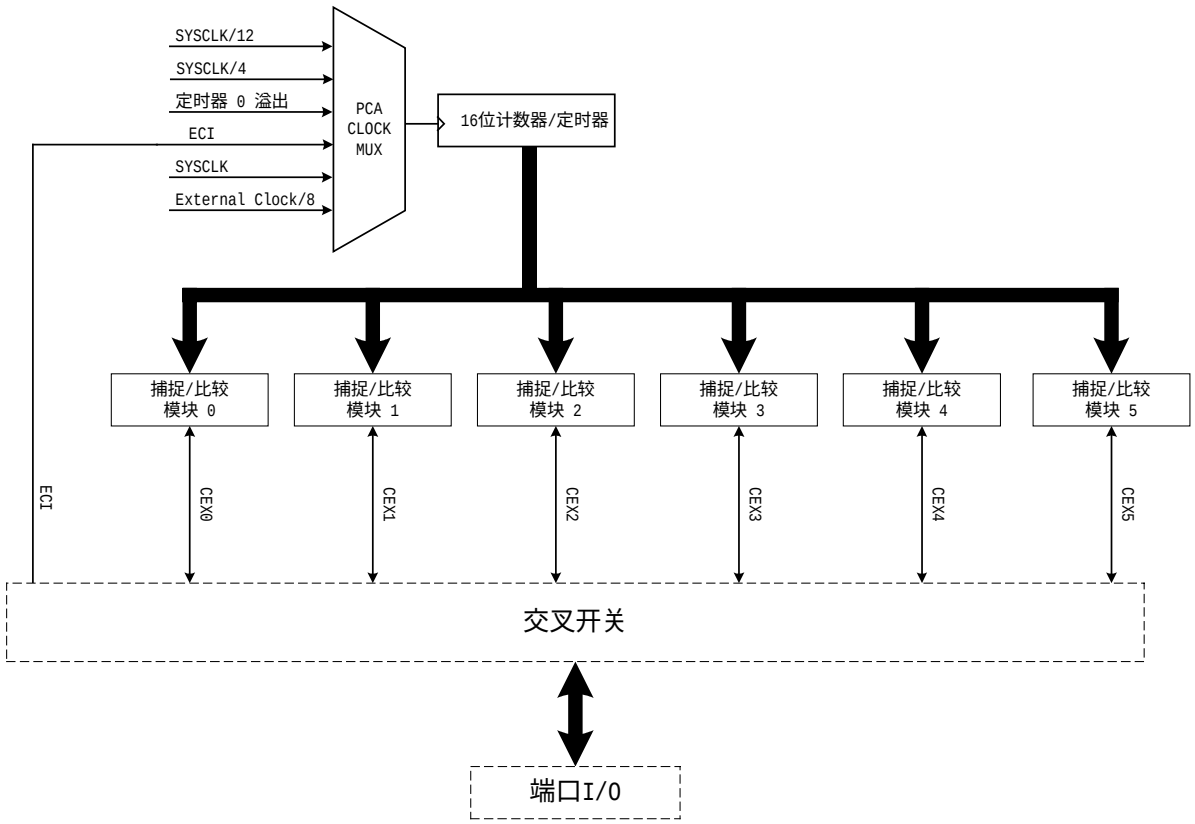


图 24.1 PCA 原理框图

24.1 PCA 计数器/定时器

16 位的 PCA 计数器/定时器由两个 8 位的 SFR 组成：PCA0L 和 PCA0H。PCA0H 是 16 位计数器/定时器的高字节（MSB），而 PCA0L 是低字节（LSB）。在读 PCA0L 的同时自动将 PCA0H 的值锁存到一个“瞬像（或快照）”寄存器，然后可以访问“瞬像”寄存器读 PCA0H 的值。先读 PCA0L 寄存器可以保证正确地读取整个 16 位 PCA0 的计数值。读 PCA0H 或 PCA0L 不影响计数器工作。PCA0MD 寄存器中的 CPS2-CPS0 位用于选择 PCA 计数器/定时器的时基信号，如表 24.1 所示。

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

表 24.1. PCA 时基输入选择

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

†注：外部时钟 8 分频与系统时钟同步。

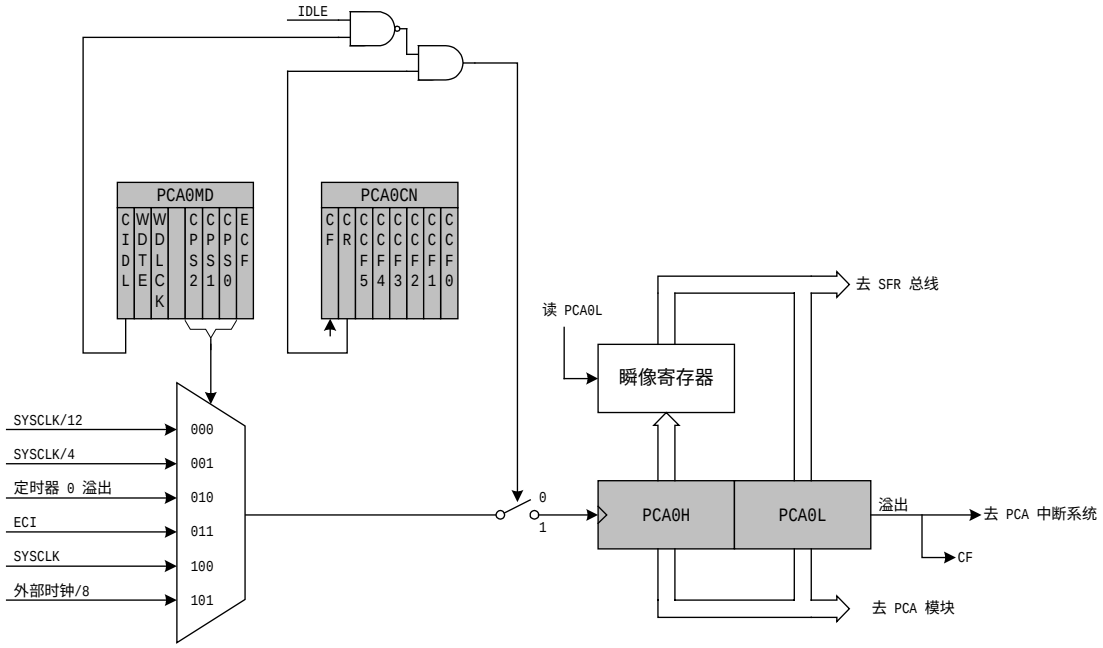


图 24.2 PCA 计数器/定时器原理框图

24.2 捕捉/比较模块

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

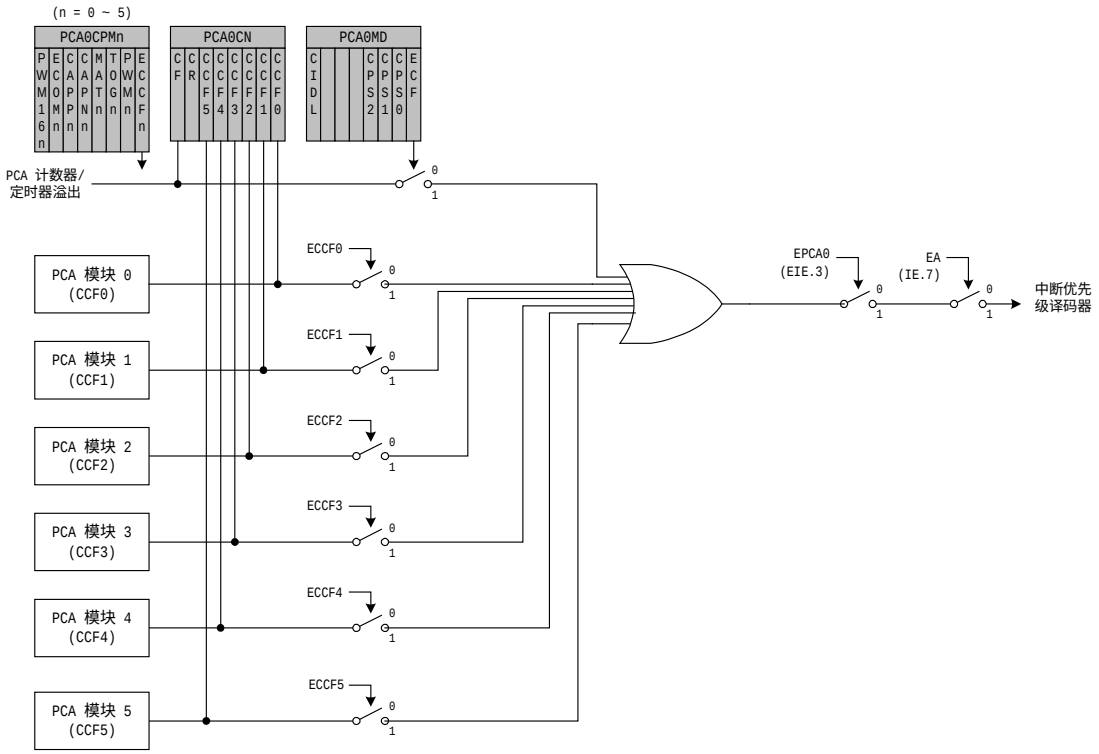
PCA0CPMn寄存器用于配置PCA捕捉/比较模块的工作方式，表24.2概述了模块工作在不同方式时这些寄存器各个位的设置情况。置‘ 1 ’ PCA0CPMn寄存器中的ECCFn位将允许模块的CCFn中断。注意：要使单独的CCFn中断得到响应，必须先整体允许PCA0中断。通过将EA位（IE.7）和EPCA0位（EIE1.3）设置为逻辑1来整体允许PCA0中断。PCA0中断配置的详细信息见图24.3。

表 24.2. PCA 捕捉/比较模块的 PCA0CPM 寄存器设置

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

X = 忽略

图 24.3 PCA 中断原理框图



24.2.1 边沿触发捕捉方式

在该方式，CEXn引脚上出现的有效电平变化导致PCA0捕捉PCA0计数器/定时器的值并将其装入到对应模块的16位捕捉/比较寄存器（PCA0CPLn和PCA0CPHn）。PCA0CPMn寄存器中的CAPPn和CAPNn位用于选择触发捕捉的电平变化类型：低电平到高电平（正沿）、高电平到低电平（负沿）或任何一种变化（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。

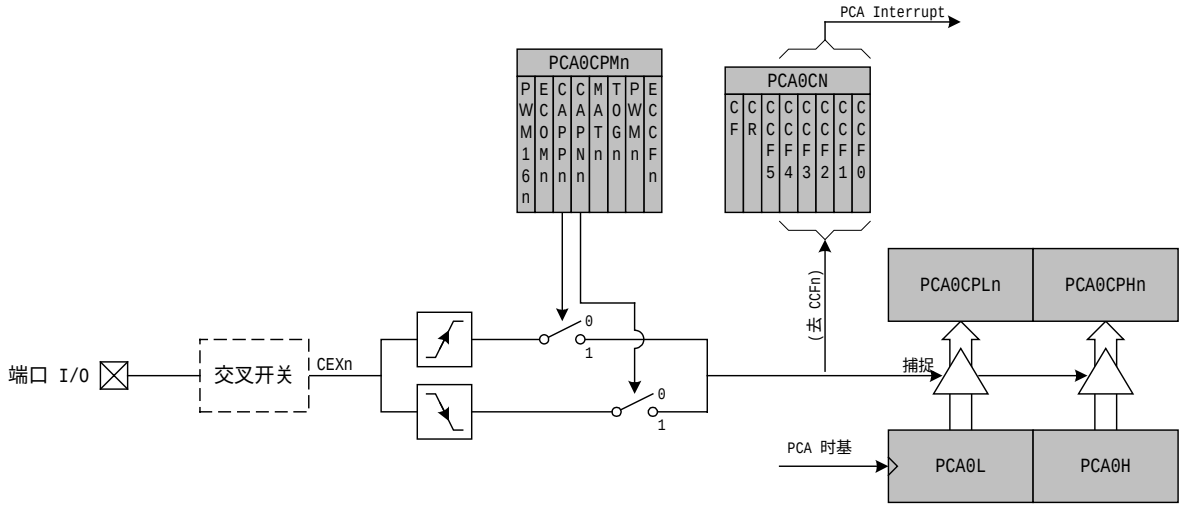


图 24.4 PCA 捕捉方式原理框图

注：CEXn引脚上输入信号的高电平或低电平至少要持续两个系统时钟周期才能确保被采样到。

24.2.2 软件定时器（比较）方式

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

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

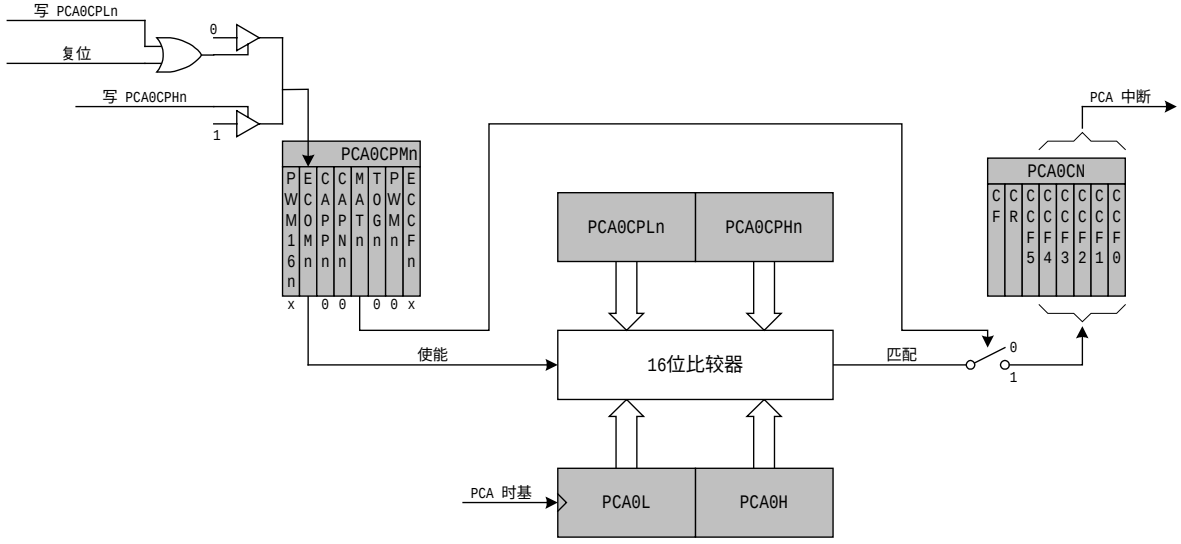


图 24.5 PCA 软件定时器方式原理框图

24.2.3 高速输出方式

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

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

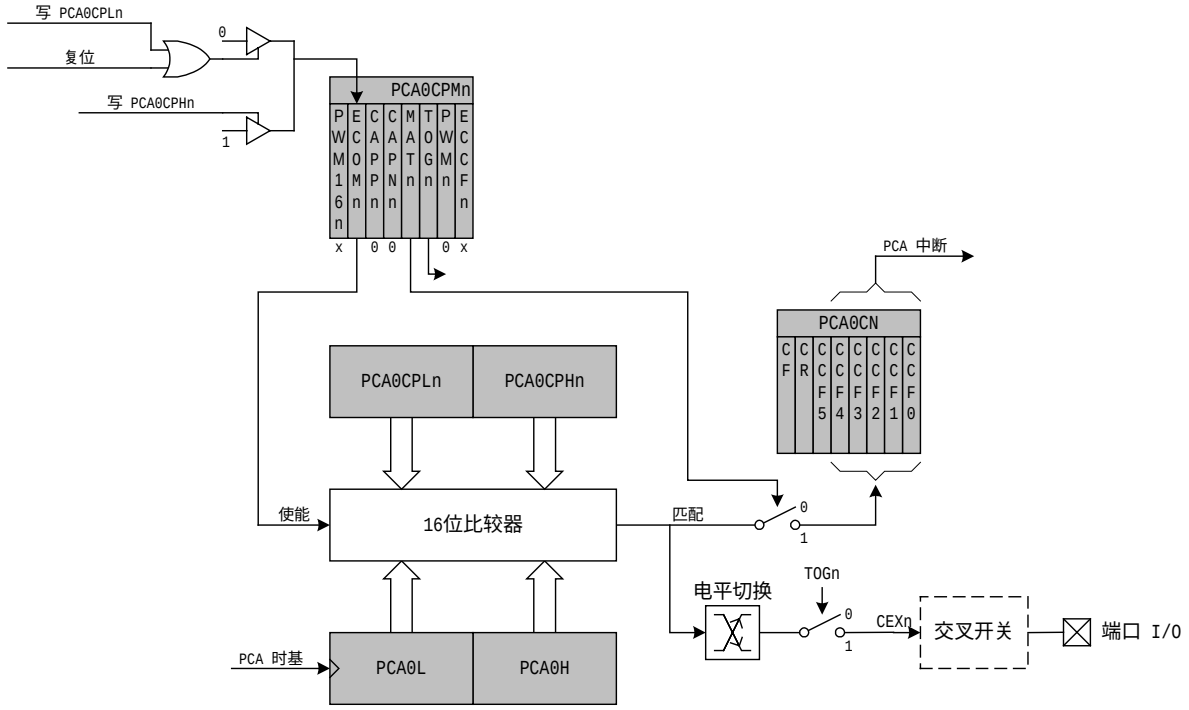


图 24.6 PCA 高速输出方式原理框图

24.2.4 频率输出方式

频率输出方式在对应的CEXn引脚产生可编程频率的方波。捕捉/比较寄存器的高字节保持着输出电平改变前要计的PCA时钟数。所产生的方波的频率由方程24.1定义。

方程24.1 方波频率输出

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

注：PCA0CPHn寄存器中的值为0x00时，对于该方程则相等价于256。

其中：F_{PCA}是由PCA方式寄存器PCA0MD中的CPS2-0位选择的PCA时钟的频率。捕捉/比较模块的低字节与PCA0计数器的低字节比较；两者匹配时，CEXn的电平发生改变，高字节中的偏移值被加到PCA0CPLn。通过置位PCA0CPMn寄存器中ECOMn、TOGn和PWMn位来使能频率输出方式。

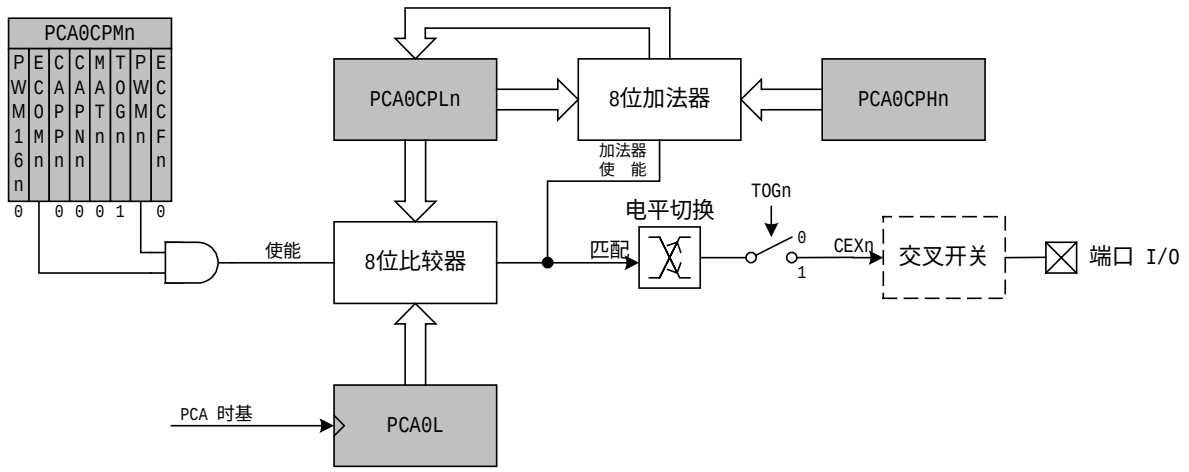


图 24.7 PCA 频率输出方式原理框图

24.2.5 8 位脉宽调制器方式

每个模块都可以独立地用于在对应的CEXn引脚产生脉宽调制（PWM）输出。PWM输出信号的频率取决于PCA0计数器/定时器的时基。使用模块的捕捉/比较寄存器PCA0CPLn改变PWM输出信号的占空比。当PCA0计数器/定时器的低字节（PCA0L）与PCA0CPLn中的值相等时，CEXn的输出为高电平。当PCA0L中的计数值溢出时，CEXn输出被置为低电平（见图24.8）。当计数器/定时器的低字节PCA0L溢出时（从0xFF到0x00），保存在计数器/定时器高字节（PCA0H）中的值被自动装入PCA0CPLn，不需软件干预。置‘1’PCA0CPMn寄存器中的ECOMn和PWMn位将使能8位脉冲宽度调制器方式。8位PWM方式的占空比由方程24.2给出。

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

方程24.2 8位PWM的占空比

$$\text{占空比} = \frac{(256 - PCA0CPHn)}{256}$$

由方程24.2可知，最大占空比为100%（PCA0CPHn = 0），最小占空比为0.39%（PCA0CPHn = 0xFF）。可以通过清‘0’ECOMn位产生0%的占空比。

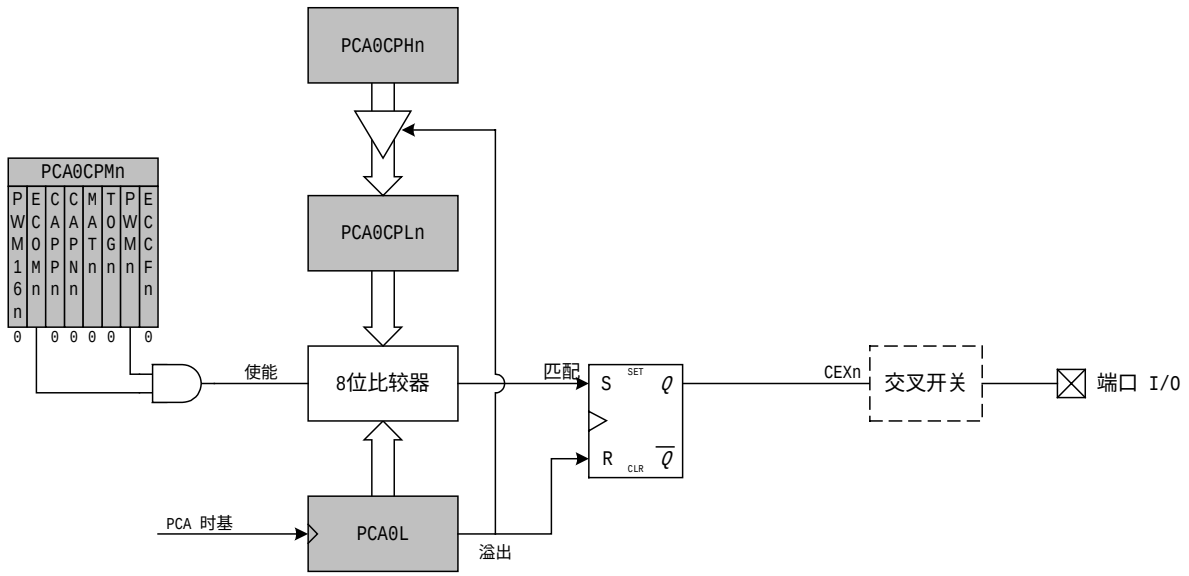


图 24.8 PCA 的 8 位 PWM 方式原理框图

24.2.6 16 位脉宽调制器方式

每个PCA0模块都可以工作在16位PWM方式。在该方式下，16位捕捉/比较模块定义PWM信号低电平时间的PCA0时钟数。当PCA0计数器与模块的值匹配时，CEXn的输出被置为高电平；当计数器溢出时，CEXn输出被置为低电平。为了输出一个占空比可变的波形，新值的写入应与PCA0 CCFn匹配中断同步。置‘1’ PCA0CPMn寄存器中的ECOMn、PWMn和PWM16n位将使能16位脉冲宽度调制器方式。为了输出一个占空比可变的波形，应将CCFn设置为逻辑‘1’以允许匹配中断。16位PWM方式的占空比由方程24.3给出。

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

方程24.3 16位PWM的占空比

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

由方程24.3可知，最大占空比为100%（PCA0CPn = 0），最小占空比为0.0015%（PCA0CPn = 0xFFFF）。可以通过清‘0’ECOMn位产生0%的占空比。

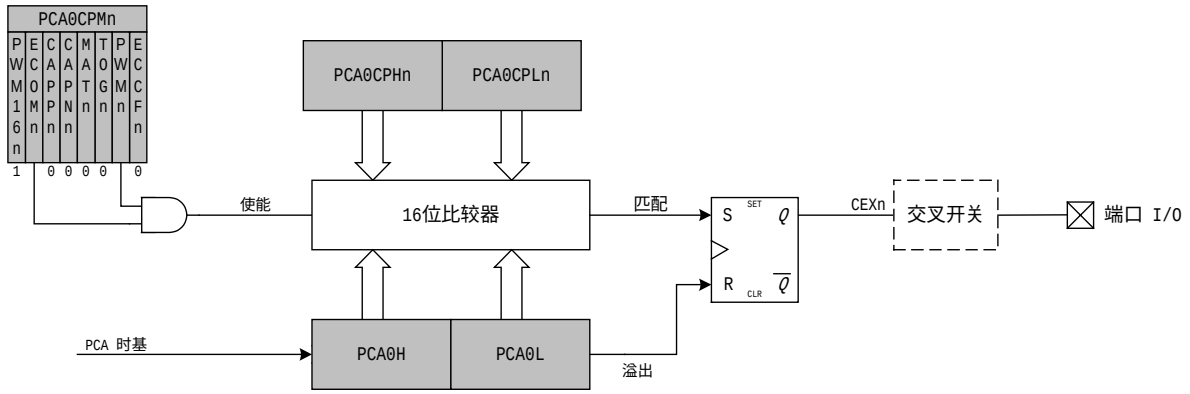


图 24.9 PCA 的 16 位 PWM 方式原理框图

24.3 PCA0 的寄存器说明

下面对与 PCA0 工作有关的特殊功能寄存器进行详细说明。

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	可位寻址

SFR 地址：0xD8
SFR 页： 0

位 7： CF：PCA 计数器／定时器溢出标志
当 PCA0 计数器／定时器从 0xFFFF 到 0x0000 溢出时由硬件置位。在计数器／定时器溢出（CF）中断被允许时，该位置‘ 1 ’将导致 CPU 转向 CF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 6： CR：PCA0 计数器／定时器运行控制
该位允许／禁止 PCA0 计数器／定时器。
0：禁止 PCA0 计数器／定时器
1：允许 PCA0 计数器／定时器

位 5： CCF5：PCA0 模块 5 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 4： CCF4：PCA0 模块 4 捕捉／比较标志
在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时，该位置‘ 1 ’将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 3： CCF3：PCA0 模块 3 捕捉／比较标志
在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时，该位置‘ 1 ’将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 2： CCF2：PCA0 模块 2 捕捉／比较标志
在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时，该位置‘ 1 ’将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 1： CCF1：PCA0 模块 1 捕捉／比较标志
在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时，该位置‘ 1 ’将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

位 0： CCF0：PCA0 模块 0 捕捉／比较标志
在发生一次匹配或捕捉时该位由硬件置位。当 CCF 中断被允许时，该位置‘ 1 ’将导致 CPU 转向 CCF 中断服务程序。该位不能由硬件自动清 0，必须用软件清 0。

图 24.10 PCA0CN：PCA 控制寄存器

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

SFR 地址：0xD9
SFR 页： 0

位 7： CIDL：PCA0 计数器／定时器空闲控制
规定 CPU 空闲方式下的 PCA0 工作方式。
0：当系统控制器处于空闲方式时，PCA0 继续正常工作。
1：当系统控制器处于空闲方式时，PCA0 停止工作。

位 6-4： 未用。读=000b，写=忽略。

位 3-1： CPS2-CPS0：PCA0 计数器／定时器脉冲选择
这些位选择 PCA0 计数器的时基。

CPS2	CPS1	CPS0	时间基准
0	0	0	系统时钟 12 分频
0	0	1	系统时钟 4 分频
0	1	0	定时器 0 溢出
0	1	1	ECI 负跳变（最大速率 = 系统时钟/4）
1	0	0	系统时钟
1	0	1	外部时钟 8 分频
1	1	0	保留
1	1	1	保留

位 0： ECF：PCA 计数器／定时器溢出中断允许
该位是 PCA0 计数器／定时器溢出（CF）中断的屏蔽位。
0：禁止 CF 中断。
1：当 CF（PCA0CN.7）置位时，允许 PCA0 计数器／定时器溢出中断请求。

图 24.11 PCA0MD：PCA0 方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	00000000
位7	位6	位5	位4	位3	位2	位1	位0	

SFR 地址：PCA0CMP0:0xDA; PCA0CMP1:0xDB; PCA0CMP2:0xDC; PCA0CMP3:0xDD; PCA0CMP4:0xDE; PCA0CMP5:0xDF
SFR 页： PCA0CMP0:页 0; PCA0CMP1:页 0; PCA0CMP2:页 0; PCA0CMP3:页 0; PCA0CMP4:页 0; PCA0CMP5:页 0;

位 7： PWM16n：16 位脉冲宽度调制使能
当脉冲宽度调制方式被使能时（PWMn = 1），该位选择 16 位方式。
0：选择 8 位 PWM。
1：选择 16 位 PWM。

位 6： ECOMn：比较器功能使能
该位使能／禁止 PCA0 模块 n 的比较器功能。
0：禁止。
1：使能。

位 5： CAPPn：正沿捕捉功能使能
该位使能／禁止 PCA0 模块 n 的正边沿捕捉。
0：禁止。
1：使能。

位 4： CAPNn：负沿捕捉功能使能
该位使能／禁止 PCA0 模块 n 的负边沿捕捉。
0：禁止。
1：使能。

位 3： MATn：匹配功能使能
该位使能／禁止 PCA0 模块 n 的匹配功能。如果被允许，当 PCA0 计数器与一个模块的捕捉／比较寄存器匹配时，PCA0MD 寄存器中的 CCFn 位置位。
0：禁止。
1：使能。

位 2： TOGn：电平切换功能允许
该位使能／禁止 PCA0 模块 n 的电平切换功能。如果被使能，当 PCA0 计数器与一个模块的捕捉／比较寄存器匹配时，CEXn 引脚的逻辑电平切换。如果 PWMn 位也被置为逻辑‘ 1 ’，则模块工作在频率输出方式。
0：禁止。
1：使能。

位 1： PWMn：脉宽调制方式允许
该位使能／禁止 PCA0 模块的 PWM 功能。如果被使能，CEXn 引脚输出脉冲宽度调制信号。如果 PWM16n 为逻辑‘ 0 ’，使用 8 位 PWM 方式；如果 PWM16n 为逻辑‘ 1 ’，使用 16 位方式。如果 TOGn 位也被置为逻辑‘ 1 ’，则模块工作在频率输出方式。
0：禁止。
1：使能。

位 0： ECCFn：捕捉／比较标志中断允许
该位设置捕捉／比较标志（CCFn）的中断屏蔽。
0：禁止 CCFn 中断
1：当 CCFn 位被置 1 时，允许捕捉／比较标志的中断请求。

图 24.12 PCA0CPMn：PCA0 捕捉／比较寄存器

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

SFR 地址：0xF9
SFR 页： 0

位 7-0： PCA0L：PCA0 计数器／定时器的低字节
PCA0L 寄存器保存 16 位 PCA0 计数器／定时器的低字节（LSB）。

图 24.13 PCA0L：PCA0 计数器／定时器低字节

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

SFR 地址：0xFA
SFR 页： 0

位 7-0： PCA0H：PCA0 计数器／定时器高字节
PCA0H 寄存器保存 16 位 PCA0 计数器／定时器的高字节（MSB）。

图 24.14 PCA0H：PCA0 计数器／定时器高字节

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

SFR 地址：PCA0CPL0:0xFB; PCA0CPL1:0xFD; PCA0CPL2:0xE9; PCA0CPL30:0xEB; PCA0CPL4:0xED; PCA0CPL5:0xE1
SFR 页： PCA0CPL0:页 0; PCA0CPL1:页 0; PCA0CPL2:页 0; PCA0CPL3:页 0; PCA0CPL4:页 0; PCA0CPL5:页 0;

位 7-0： PCA0CPLn：PCA0 捕捉模块低字节
PCA0CPLn 寄存器保存 16 位捕捉模块 n 的低字节（LSB）。

图 24.15 PCA0CPLn：PCA 捕捉模块低字节

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

SFR 地址：PCA0CPH0:0xFC; PCA0CPH1:0xFD; PCA0CPH2:0xEA; PCA0CPH30:0xEC; PCA0CPH4:0xEE; PCA0CPH5:0xE2
SFR 页： PCA0CPH0:页 0; PCA0CPH1:页 0; PCA0CPH2:页 0; PCA0CPH3:页 0; PCA0CPH4:页 0; PCA0CPH5:页 0;

位 7-0： PCA0CPHn：PCA0 捕捉模块高字节
PCA0CPHn 寄存器保存 16 位捕捉模块 n 的高字节（MSB）。

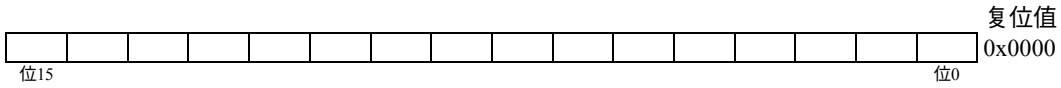
图 24.16 PCA0CPHn：PCA0 捕捉模块高字节

25. JTAG (IEEE 1149.1)

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

JTAG 接口使用 MCU 上的四个专用引脚，它们是：TCK、TMS、TDI 和 TDO。

通过 16 位的 JTAG 指令寄存器（IR）可以发出图 25.1 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。



IR 值	指令	说明
0x0000	EXTEST	选择用于控制和观察所有器件引脚的边界数据寄存器。
0x0002	SAMPLE/ PRELOAD	选择用于观察和预设扫描路径锁存器的边界数据寄存器。
0x0004	IDCODE	选择器件的 ID 寄存器。
0xFFFF	BYPASS	选择旁路数据寄存器。
0x0082	闪存控制	选择 FLASHCON 寄存器，以控制接口逻辑对读和写 FLASHDAT 寄存器如何响应。
0x0083	闪存数据	选择用于读和写闪速存储器的 FLASHDAT 寄存器。
0x0084	闪存地址	选择 FLASHADR 寄存器，该寄存器保持 FLASH 读、写和擦除操作的地址。
0x0085	闪存定标	选择 FLASHSCL 寄存器，该寄存器控制 FLASH 单稳态定时和常读允许。

图 25.1 IR：JTAG 指令寄存器

25.1 边界扫描

边界扫描路径中的数据寄存器是一个 134 位的移位寄存器。通过执行 EXTEST 和 SAMPLE 命令，边界数据寄存器能提供对所有器件引脚以及 SFR 总线和弱上拉功能的控制和观察。

表 25.1 边界数据寄存器位定义

EXTEST 支持捕捉和更新两种操作，而 SAMPLE 只执行捕捉操作。

位	操作	目标
0	捕捉	来自 MCU 的复位使能（64 脚 TQFP）
	更新	去/RST 引脚的复位使能（64 脚 TQFP）
1	捕捉	来自/RST 引脚的复位输入（64 脚 TQFP）
	更新	去/RST 引脚的复位输出（64 脚 TQFP）
2	捕捉	来自 MCU 的复位使能（100 脚 TQFP）
	更新	去/RST 引脚的复位使能（100 脚 TQFP）
3	捕捉	来自/RST 引脚的复位输入（100 脚 TQFP）
	更新	去/RST 引脚的复位输出（100 脚 TQFP）
4	捕捉	来自 XTAL1 引脚的外部时钟
	更新	未用
5	捕捉	来自 MCU 的弱上拉使能
	更新	去端口引脚的弱上拉使能
6,8,10,12, 14,16,18,20	捕捉	来自 MCU 的 P0.n 输出使能（如 Bit6=P0.0, Bit8=P0.1 等）
	更新	去引脚的 P0.n 输出使能（如 Bit6=P0.0oe, Bit8=P0.1oe 等）
7,9,11,13, 15,17,19,21	捕捉	来自引脚的 P0.n 输入（如 Bit7=P0.0, Bit9=P0.1 等）
	更新	P0.n 输出到引脚（如 Bit7=P0.0, Bit9=P0.1 等）
22,24,26,28, 30,32,34,36	捕捉	来自 MCU 的 P1.n 输出使能
	更新	去引脚的 P1.n 输出使能
23,25,27,29, 31,33,35,37	捕捉	来自引脚的 P1.n 输入
	更新	P1.n 输出到引脚
38,40,42,44, 46,48,50,52	捕捉	来自 MCU 的 P2.n 输出使能
	更新	去引脚的 P2.n 输出使能
39,41,43,45, 47,49,51,53	捕捉	来自引脚的 P2.n 输入
	更新	P2.n 输出到引脚
54,56,58,60, 62,64,66,68	捕捉	来自 MCU 的 P3.n 输出使能
	更新	去引脚的 P3.n 输出使能
55,57,59,61, 63,65,67,69	捕捉	来自引脚的 P3.n 输入
	更新	P3.n 输出到引脚
70,72,74,76, 78,80,82,84	捕捉	来自 MCU 的 P4.n 输出使能
	更新	去引脚的 P4.n 输出使能
71,73,75,77, 79,81,83,85	捕捉	来自引脚的 P4.n 输入
	更新	P4.n 输出到引脚
86,88,90,92, 94,96,98,100	捕捉	来自 MCU 的 P5.n 输出使能
	更新	去引脚的 P5.n 输出使能
87,89,91,93, 95,97,99,101	捕捉	来自引脚的 P5.n 输入
	更新	P5.n 输出到引脚

表 25.1 边界数据寄存器位定义（续）

102,104,106,108, 110,112,114,116	捕捉	来自 MCU 的 P6.n 输出使能
	更新	去引脚的 P6.n 输出使能
103,105,107,109, 111,113,115,117	捕捉	来自引脚的 P6.n 输入
	更新	P6.n 输出到引脚
118,120,122,124, 126,128,130,132	捕捉	来自 MCU 的 P7.n 输出使能
	更新	去引脚的 P7.n 输出使能
119,121,123,125, 127,129,131,133	捕捉	来自引脚的 P7.n 输入
	更新	P7.n 输出到引脚

25.1.1 EXTEST 指令

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

25.1.2 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。

25.1.3 BYPASS 指令

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

25.1.4 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。

				复位值
版本	器件编号	制造商 ID	1	0xn0007243
位31 位28	位27 位12	位11 位1	位0	

版本=0000b
器件编号= 0000 0000 0000 0111b（C8051F120/1/2/3/4/5/6/7）
制造商 ID= 0010 0100 001b（Silicon Labs）

图 25.2 DEVICEID：JTAG 器件 ID 寄存器

25.2 闪存编程命令

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

19:18	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），但不允许 Update_DR（数据寄存器更新）操作。由于更新操作被禁止，查询操作只进行一个二进制位的移入/移出。

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

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

从间接数据寄存器输出的数据具有如下格式：

19	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。

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

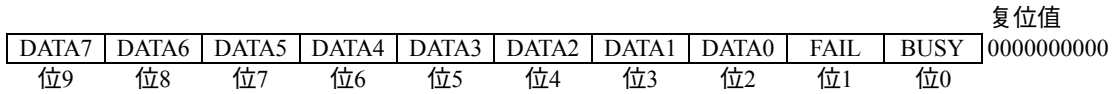
该寄存器决定闪存接口逻辑对读/写 FLASHDAT 寄存器的操作如何响应。

位 7： SFLE：临时 FLASH 存储器访问允许。
当该位被置 1 时， FLASH 的读和写操作将指向两个 128 字节的 FLASH 临时存储扇区。当 SFLE 被设置为逻辑 ‘ 1 ’ 时，不应访问 0x00 – 0xFF 以外的地址范围（地址 0x400 例外，它可被用于同时擦除这两个临时区）。对该地址范围以外的地址进行读/写可能产生不可预料的结果。
0：访问 FLASH 时将访问 128K 字节的程序/数据 FLASH 扇区。
1：访问 FLASH 时将访问两个 128 字节的临时存储器扇区。

位 6-4： WRMD2-0：写方式选择
写方式选择位控制闪存接口逻辑对写 FLASHDAT 寄存器的操作如何响应，其值意义如下：
000： 一个 FLASHDAT 写操作替换 FLASHDAT 寄存器中的数据，否则忽略。
001： 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。
010： 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 = 0xFDFE – 0xFDFF，则整个用户空间将被擦除（即除保留区 0x1FC00 ~ 0x1FFFF 外的整个 FLASH 存储器）。
（所有其它 WRMD2-0 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 读操作提供位于 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： 如果没有其它操作正在执行，并且前一次读操作的数据已经被从 FLASHDAT 中读出，则 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。该方式用于单个字节读（或块中的最后一个字节）。
（所有其它 RDMD2-0 值保留。）

图 25.3 FLASHCON：JTAG FLASH 控制寄存器

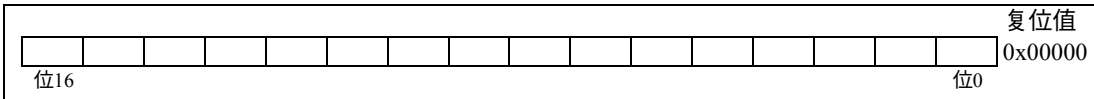


该寄存器用于通过 JTAG 接口从 FLASH 中读数据或写数据到 FLASH。

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： BUSY：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 BUSY= 1 时，读或写都不会启动另一个操作。

图 24.4 FLASHDAT：JTAG FLASH 数据寄存器



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 16-0：FLASH 操作的 17 位地址。

图 25.4 FLASHADR：JTAG FLASH 地址寄存器

25.3 调试支持

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

开发套件 C8051F120DK 适用于 C8051F12x 和 C8051F13x 系列的所有 MCU，具有开发应用代码和进行在系统调试所需要的全部硬件和软件。每个套件包括一个具有调试器和 8051 汇编器的集成开发环境（IDE）。套件中还包括一个被称为串口适配器的 RS232 到 JTAG 的接口模块，还有一个安装有 C8051F120 的目标应用板。每个套件还包括所需电缆及墙装电源。

联系信息

Silicon Laboratories Inc.

4635 Boston Lane
Austin, TX 78735
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032
Email: mcuinfo@silabs.com
Internet: www.silabs.com

新华龙电子有限公司

电话：0755-83645240　83645242　83645244　83645251
技术支持：0755-83645259
传真：0755-83645243
地址：深圳市福田区华强北路现代之窗大厦 A 座 13F C 室(518013)
Email：sales@xhl.com.cn
Email：shenzhen@xhl.com.cn
网站： www.xhl.com.cn
技术支持：support-sz@xhl.com.cn