

## 混合信号 8KB ISP FLASH MCU 系列

### 模拟外设

- 逐次逼近式模数转换器 (SAR ADC)
  - 12 位分辨率 ('F206)
  - 8 位分辨率 ('F2201/1/6)
  - $\pm 1/4$  LSB(最低位) INL(8 位)和  $\pm 2$  LSB INL(12 位)
  - 最大速度可到 100ksps
  - 多至 32 通道的输入多路转换器, 每个 I/O 口端子均可作为模数转换器的输入
- 两个比较器
  - 16 位可编程滞后状态
  - 可配置的中断或复位
- VDD 监视器和节电降压检测器

### 片内 JTAG 调试

- 片内调试电路提供了全速、非插入式的系统在线调试 (不需要仿真器!)
- 提供断点, 单步执行, 观察点, 堆栈监视器
- 检查 / 修改存储器和寄存器
- 较使用 ICE 芯片, 目标仿真头和仿真插槽的仿真系统性能更加优越
- 完善、低成本的开发工具包:

### 高速 8051 微控制器内核

- 流水线式的指令结构; 70%的指令执行

只需 1 或 2 个系统时钟

- 在 25MHz 的时钟下, 速度可达 25MIPS (每秒百万条指令)
- 扩展的中断处理器

### 存储器

- 256 字节的内部数据 RAM
- 1024 字节 XRAM (适用于 F206 / 226 / 236)
- 8k 字节的 FLASH; 以 512 字节扇区进行系统在线编程

### 数字外设

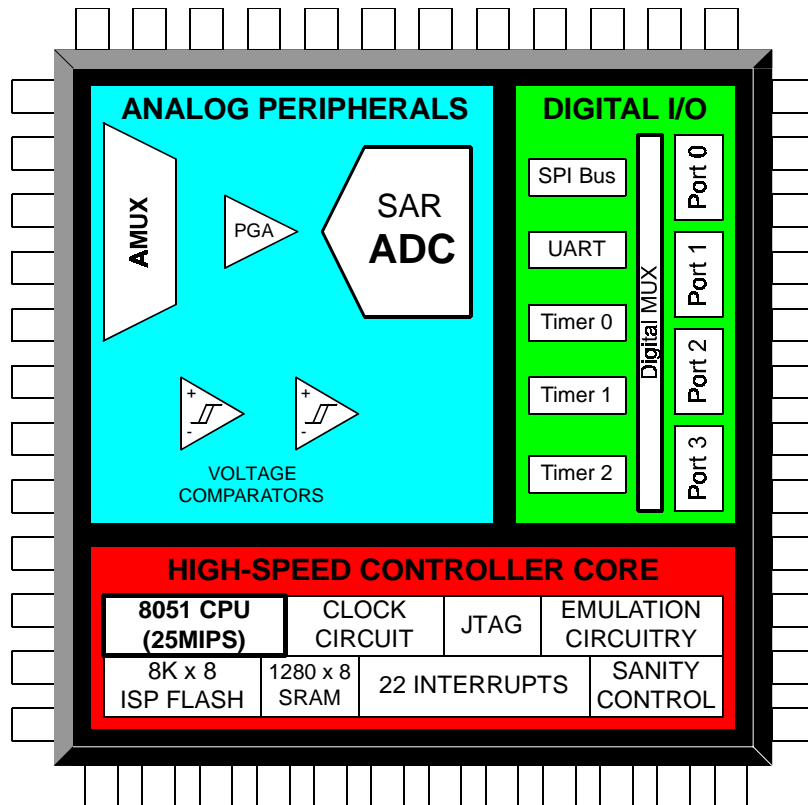
- 4 个字节宽的 I/O 端口, 耐压均为 5V
- 硬件通用异步收发器 (UART) 和串行外围设备接口 (SPI) 总线
- 3 个通用的 16 位计数器 / 定时器
- 专用的看门狗定时器
- 双向复位
- 系统时钟有: 内部可编程振荡器, 外部晶振, 外部阻容 (RC) 式或外部时钟

电源电压.....2.7V—3.6V

- 典型的操作电流: 10mA@25MHz
- 多种节能的睡眠和关断模式

(48 脚 TQFP 和 32 脚 LQFP 均可行)

温度范围: -40°C 至 +85°C



AMUX 模拟量多路转换器    SANITY CONTROL 控制正常    VOLTAGE COMPARATORS 电压比较器  
 CLOCK CIRCUIT 时钟电路    ANALOG PERIPHERALS 模拟外设    22 INTERRUPTS 22 个中断  
 EMULATION CIRCUITRY 仿真电路    HIGH-SPEED CONTROLLER CORE 高速控制器内核



目 录

1. 系统概述 .....7

表1.1.1. 产品选择指南 .....7

图1.1. C8051F206, C8051F220和 C8051F226框图 (48TQFP) .....8

图1.2. C8051F221框图 (32LQFP) .....9

图1.3. C8051F230和 C8051F236框图 (48TQFP) .....10

图1.4. C8051F231框图 (32LQFP) .....11

1.1. CIP-51TM 微控制器内核.....12

图1.5. MCU 处理能力峰值比较.....12

图1.6. 板内时钟和复位 .....13

1.2片内存储器.....14

图1.7. 片内存储器映象 .....14

1.2. JTAG .....15

图1.8. 调试环境框图 .....15

1.2. 数字量/模拟量可配置的 I/O .....16

图1.9. I / O 端口功能框图 .....16

1.2. 串行端口 .....16

1.3. 模数转换器 .....17

图1.10. 模数转换器框图.....17

1.4. 比较器 .....18

图1.11. 比较器框图.....18

2. 最大绝对额定值.....19

3. 全局直流电特性.....19

4. 输出端子和封装定义.....20

表4.1. 端子定义 .....20

图4.1. TQFP-48端子图.....22

图4.2. LQFP-32端子图.....23

图4.3. TQFP-48封装图.....24

图4.4. LQFP-32封装图.....25

5. 模数转换器 (8位, 仅限于 C8051F220/1/6) .....26

图5.1. 8位模数转换器 (ADC) 功能框图.....26

5.1. 模拟量多路转换器与可编程增益放大器 (PGA) .....26

5.2. 模数转换器 (ADC) 的操作模式.....26

图5.2. 12位 ADC 跟踪和转换实例时序.....27

图5.3. AMX0SL: AMUX 通道选择寄存器 (C8051F220/1/6和 C8051F206) .....28

图5.4. ADC0CF: ADC 配置寄存器 (C8051F220/1/6和 C8051F206) .....29

图5.5. ADC0CN: ADC 控制寄存器 (C8051F220/1/6和 C8051F206) .....30

图5.6. ADC0H: ADC 数据字寄存器 (C8051F220/1/6和 C8051F206) .....31

5.3. 模数转换器 (ADC) 可编程窗口检测器.....31

图5.7. ADC0GTH: ADC 大于数据寄存器 (C8051F220/1/6和 C8051F206) .....31

图5.8. ADC0LTH: ADC 小于数据字节寄存器 (C8051F220/1/6和 C8051F206) .....31

图5.9. 8位 ADC 窗口中断实例.....32

表5.1. 8位 ADC 电特性.....33

6. 模数转换器 (12位, 仅限于 C8051F206) .....34

图6.1. 12位模数转换器 (ADC) 功能框图.....34

6.1. 模拟量多路转换器与程控增益放大器 (PGA) .....34

6.2. 模数转换器(ADC)的操作模式.....	34
图6.2. 12位 ADC 跟踪和转换实例时序 .....	35
图6.3. AMX0SL: AMUX 通道选择寄存器 (C8051F220/1/6和 C8051F206) .....	36
图6.4. ADC0CF: ADC 配置寄存器 (C8051F220/1/6和 C8051F206) .....	37
图6.5. ADC0CN: ADC 控制寄存器 (C8051F220/1/6和 C8051F206) .....	38
图6.6. ADC0H: ADC 数据字最高有效位 (MSB) 寄存器 (C8051F206) .....	39
图6.7. ADC0L: ADC 数据字最低有效位 (LSB) 寄存器(C8051F206) .....	39
6.3. 模数转换器 (ADC) 可编程窗口检测器.....	39
图6.8. ADC0GTH: ADC 大于数据高字节寄存器 (C8051F206) .....	40
图6.9. ADC0GTL: ADC 大于数据低字节寄存器 (C8051F206) .....	40
图6.10. ADC0LTH: ADC 小于数据高字节寄存器 (C8051F206) .....	40
图6.11. ADC0LTL: ADC 小于数据低字节寄存器 (C8051F206) .....	40
图6.12. 12位 ADC 窗口中断例子, 右对齐数据 .....	41
图6.13. 12位 ADC 窗口中断例子, 左对齐数据 .....	42
表6.1. 12位 ADC 电特性 (仅限于 C8051F206) .....	43
7. 基准电压 (C8051F206/220/1/6) .....	44
图7.1. 基准电压功能框图.....	44
图7.2. REF0CN: 基准电压控制寄存器.....	44
表7.1. 基准电压电特性.....	44
8. 比较器 .....	45
图8.1. 比较器功能框图.....	45
图8.2. 比较器滞后曲线.....	46
图8.3. CPT0CN: 比较器0控制寄存器 .....	47
图8.4. CPT1CN: 比较器1控制寄存器 .....	48
表8.1. 比较器电特性.....	49
9. CIP-51微控制器 .....	50
图9.1. CIP-51框图.....	50
9.1. 指令集.....	52
表9.1. CIP-51指令集汇总.....	52
9.2. 存储器结构.....	56
图9.2. 存储器映象.....	57
9.3. 特殊功能寄存器.....	59
表9.2. 特殊功能寄存器的内存映象.....	59
表9.3. 特殊功能寄存器.....	59
图9.3. SP: 堆栈指针.....	62
图9.4. DPL: 数据指针低位字节 .....	62
图9.5. DPH: 数据指针高位字节 .....	62
图9.6. PSW: 程序状态字 .....	63
图9.7. ACC: 累加器 .....	64
图9.8. B: B 寄存器 .....	64
9.4. 中断处理器.....	65
表9.4. 中断汇总.....	66
图9.9. IE: 中断允许.....	67
图9.10. IP: 中断优先权 .....	68
图9.11. EIE1: 扩展中断允许1.....	69
图9.12. EIE2: 扩展中断允许2.....	70

图9.13. EIP1: 扩展中断优先级1 .....	71
图9.14. EIP2: 扩展中断优先级2 .....	72
9.5. 电源管理方式 .....	73
图9.15. PCON: 电源控制寄存器 .....	74
10. FLASH 存储器 .....	75
10.1. FLASH 存储器编程 .....	75
表10.1. FLASH 存储器电特性 .....	75
10.2. 非易失的数据存储 .....	76
10.3. 加密选项 .....	76
图10.1. FLASH 程序存储器的加密字节 .....	77
图10.2. PSCTL: 程序存储 RW 控制 .....	78
图10.3. FLSC: FLASH 存储器计时的预定标器 .....	79
图10.4. FLACL: FLASH 存取限 .....	79
11. 板内 XRAM (C8051F226 / 236 / 206) .....	80
图11.1. EMIOCN: 外部存储器接口控制 .....	80
12. 复位源 .....	81
图12.1. 复位源框图 .....	81
12.1. 上电复位 .....	82
12.2. 软件强制复位 .....	82
图12.2. VDD 监视器定时框图 .....	82
12.3. 电源失效复位 .....	82
12.4. 外部复位 .....	83
12.5. 时钟丢失检测器复位 .....	83
12.6. 比较器0复位 .....	83
12.7. 看门狗定时器复位 .....	83
图12.3. WDTCN: 看门狗定时器控制寄存器 .....	84
图12.4. RSTSRC: 复位源寄存器 .....	85
表12.1. VDD 监视器电特性 .....	86
13. 振荡器 .....	87
图13.1. 振荡器框图 .....	87
图13.2. OSCICN: 内部振荡器控制寄存器 .....	88
表13.1. 内部振荡器电特性 .....	88
图13.3. OSCXCN: 外部振荡器控制寄存器 .....	89
13.1. 外部晶体振荡器实例 .....	90
13.2. 外部阻容振荡器实例 .....	90
13.3. 外部电容振荡器实例 .....	90
14. 输入 / 输出 (I / O) 端口 .....	91
14.1. I / O 端口初始化 .....	91
图14.1. I / O 端口功能框图 .....	92
图14.2. I / O 端口单元框图 .....	92
图14.3. PRT0MX: I / O 端口多路转换器寄存器0 .....	93
图14.4. PRT1MX: I / O 端口多路转换器寄存器1 .....	94
图14.5. PRT2MX: I / O 端口多路转换器寄存器2 .....	94
14.2. 通用的 I / O 端口 .....	95
图14.6. P0: 端口0寄存器 .....	95
图14.7. PRT0CF: 端口0配置寄存器 .....	95

图14.8. POMODE: 端口0数 / 模输入模式 .....	96
图14.9. P1: 端口1寄存器 .....	96
图14.10. PRT1CF: 端口1配置寄存器 .....	96
图14.12. PRT1IF: 端口1中断标志寄存器 .....	97
图14.13. P2: 端口2寄存器 .....	98
图14.14. PRT2CF: 端口2配置寄存器 .....	98
图14.15. P2MODE: 端口2数 / 模输入模式 .....	98
图14.16. P3: 端口3寄存器 .....	99
图14.17. PRT3CF: 端口3配置寄存器 .....	99
图14.18. P3MODE: 端口3数 / 模输入模式 .....	99
表14.1. I / O 端口 DC 电特性 .....	100
15. 串行外设部件接口 (SPI) 总线 .....	101
图15.1. SPI 框图 .....	101
图15.2. 典型的 SPI 互连 .....	102
15.1 信号描述 .....	102
15.2 操作 .....	103
图15.3. 全双工操作 .....	103
15.3 串行时钟时序 .....	104
图15.4. 数据 / 时钟时序框图 .....	104
15.4. SPI 特殊功能寄存器 .....	105
图15.5. SPI0CFG: SPI 配置寄存器 .....	105
图15.6. SPI0CN: SPI 控制寄存器 .....	106
图15.7. SPI0CKR: SPI 时钟频率寄存器 .....	107
图15.8. SPI0DAT: SPI 数据寄存器 .....	107
16. 通用异步收发器 (UART) .....	108
图16.1. UART 框图 .....	108
16.1. UART 操作方式 .....	109
表16.1. UART 方式 .....	109
图16.2. UART 方式0互连 .....	109
图16.3. UART 方式0时序框图 .....	109
图16.4. UART 方式1时序框图 .....	110
图16.5. UART 方式1, 2, 3互连框图 .....	111
图16.6. UART 方式2, 3时序框图 .....	111
16.2 多处理器通信 .....	112
图16.7. UART 多处理器方式互连框图 .....	112
表16.2. 标准波特率时振荡器的频率 .....	113
图16.8. SBUF: 串行 (UART) 数据缓冲寄存器 .....	113
图16.9. SCON: 串行口控制寄存器 .....	114
17. 定时器 .....	115
17.1. 定时器0和定时器 1 .....	115
图17.1. T0方式0框图 .....	116
图17.2. T0方式2框图 .....	117
图17.3. T0方式3框图 .....	118
图17.4. TCON: 定时器控制寄存器 .....	119
图17.5. TMOD: 定时器方式寄存器 .....	120
图17.6. CKCON: 时钟控制寄存器 .....	121

图17.7.	TL0: 定时器0低字节 .....	122
图17.8.	TL1: 定时器1低字节 .....	122
图17.9.	TH0: 定时器0高字节 .....	122
图17.10.	TH1: 定时器1高字节 .....	122
17.2.	定时器2 .....	123
图17.11.	T2方式0框图 .....	124
图17.12.	T2方式1框图 .....	125
图17.13.	T2方式2框图 .....	126
图17.14.	T2CON: 定时器2控制寄存器 .....	127
图17.15.	RCAP2L: 定时器2捕捉寄存器低字节 .....	128
图17.16.	RCAP2H: 定时器2捕捉寄存器高字节 .....	128
图17.17.	TL2: 定时器2低字节 .....	128
图17.18.	TH2: 定时器2高字节 .....	128
18.	JTAG .....	129
图18.1.	IR: JTAG 指令寄存器 .....	129
18.1	快速编程命令 .....	130
图18.2.	FLASHCON: JTAGFLASH 控制寄存器 .....	131
图18.3.	FLASHADR: JTAGFLASH 地址寄存器 .....	131
图18.4.	FLASHDAT: JTAGFLASH 数据寄存器 .....	132
图18.5.	FLASHSCL: JTAGFLASH 比例寄存器 .....	132
18.1.	边界旁路扫描和标识码 .....	133
图18.6.	DEVICEID: JTAG 设备标志寄存器 .....	133
18.2.	调试支持 .....	133

## 1. 系统概述

C8051F2xx 系列是一种高集成度的混合信号系统，系统集成在一个 MCU 芯片上，可带有一个真正 8 位的多通道模数转换器（ADC）（F220/1/6 和 F206）或不含 ADC（F230/1/6）。每个模块有一个与 8051 兼容的微控制器内核，它具有 8k 字节的 FLASH 存储器，同时，也有用硬件实现的 UART 和 SPI 串行接口（用户软件中无“位绑定”）。此系列产品的特点是有 22 或 32 个通用的 I/O 端子，其中有一些能用于指定的数字外设部件接口，任意一个端子均可配置成为模拟量输入至模数转换器（仅限于 F220/1/6 和 F206）。（请看表 1.1.1—产品选择指南：每种 MCU 特性的快速参考）

片内还有 VDD 监视器，硬件看门狗定时器（WDT）和时钟振荡器，片内 FLASH 存储器可在线多次编程，且能用于非易失性数据的存储。在集成的外设部件中，可分别关断单个设备或所有外设部件，以节省电源。所有产品均有 256 字节的 SRAM，在 F226 / F236 中还附加有一个 1024 字节的 RAM。

片内 JTAG 调试支持非插入方式（不使用片内资源）、全速工作，电路调试使用安装了最终应用程序的 MCU 产品。系统调试支持存储器、寄存器的检查和修改，可设置断点，监视点，单步执行，运行以及停机命令。在用 JTAG 口仿真时，所有的模拟和数字量外设部件均具备完整的功能。

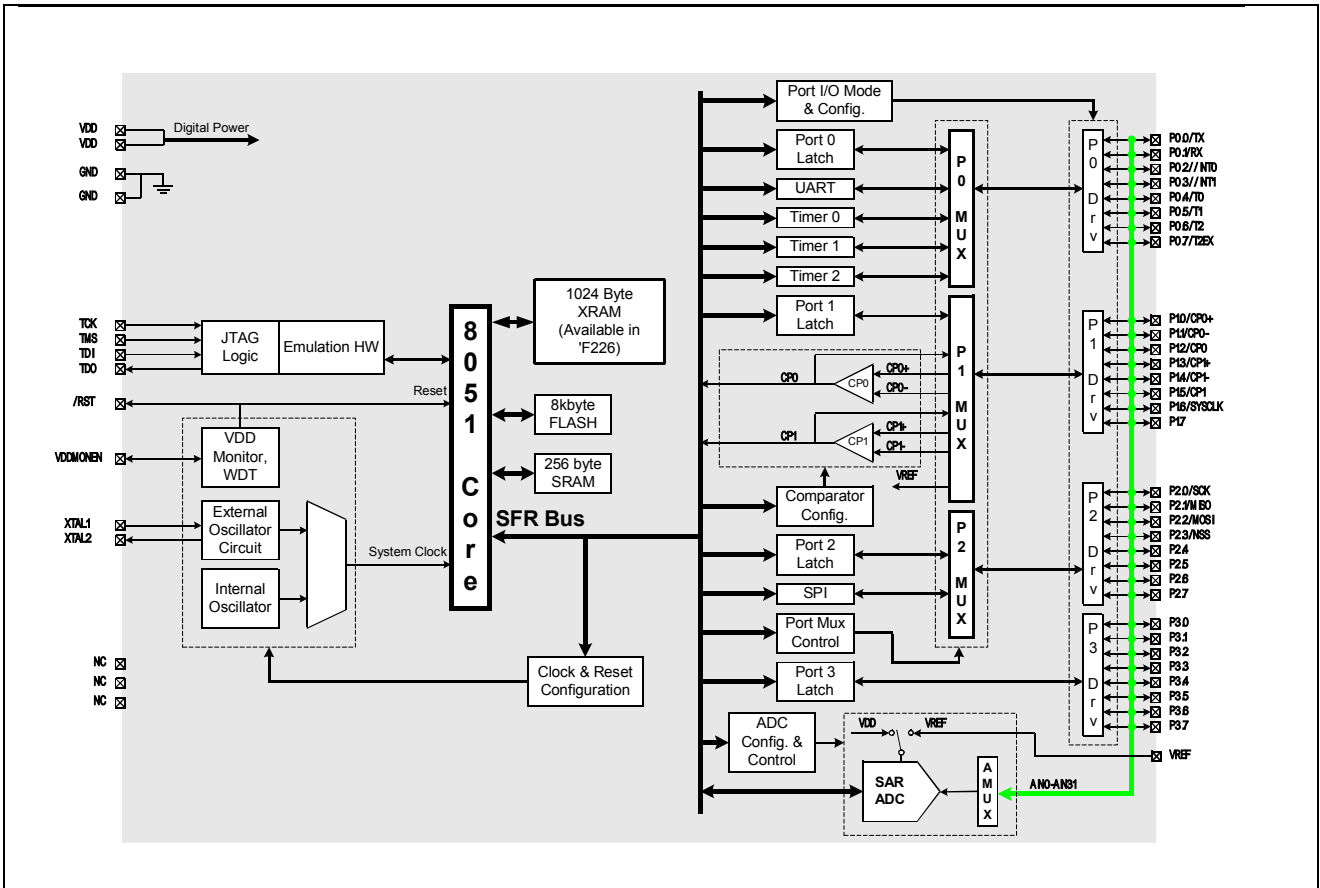
每种 MCU 在工业温度范围（-45℃—+85℃）下的操作电压是 2.7V—3.6V，可用 48 脚的 TQFP 和 32 脚的 LQFP。端口 I/O 对输入信号的耐压可达到 5V。

**表 1.1.1. 产品选择指南**

型号	每秒百万条指令 MIPS (峰值)	FLASH 存储器	RAM	串行外设部件接口 SPI	通用异步收发器 UART	计时器 (16 位)	数字量 I/O 端口	模数转换器分辨率 (位)	ADC 最大速度 (ksps)	ADC 输入	电压比较器	封装形式
C8051F206	25	8k	1280	有	有	3	32	12	100	32	2	48TQFP
C8051F220	25	8k	256	有	有	3	32	8	100	32	2	48TQFP
C8051F221	25	8k	256	有	有	3	32	8	100	32	2	32LQFP
C8051F226	25	8k	1280	有	有	3	32	8	100	32	2	48TQFP
C8051F230	25	8k	256	有	有	3	32	-	-	-	2	48TQFP
C8051F231	25	8k	256	有	有	3	32	-	-	-	2	32LQFP
C8051F236	25	8k	1280	有	有	3	32	-	-	-	2	48TQFP



图 1.1. C8051F206, C8051F220 和 C8051F226 框图 (48TQFP)



Digital Power 数字电源

JTAG logic JTAG 逻辑

Emulation HW 仿真 HW

Latch 锁存器

External Oscillator Circuit 外部振荡器电路

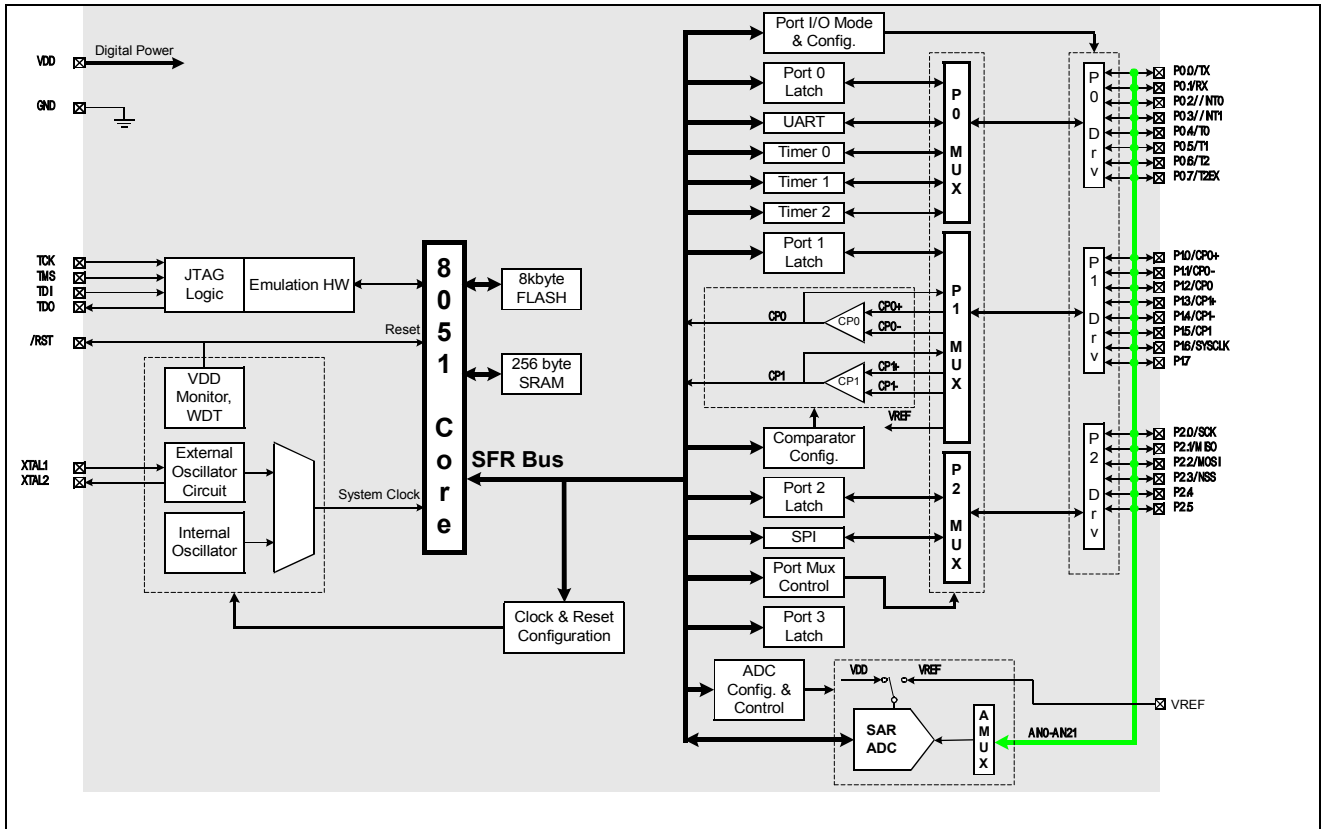
UART 通用异步收发器

SFR 特殊功能寄存器

Clock & Reset Configuration 时钟, 复位配置



图 1.2. C8051F221 框图 (32LQFP)



Digital Power 数字电源

JTAG logic JTAG 逻辑

Emulation HW 仿真 HW

Latch 锁存器

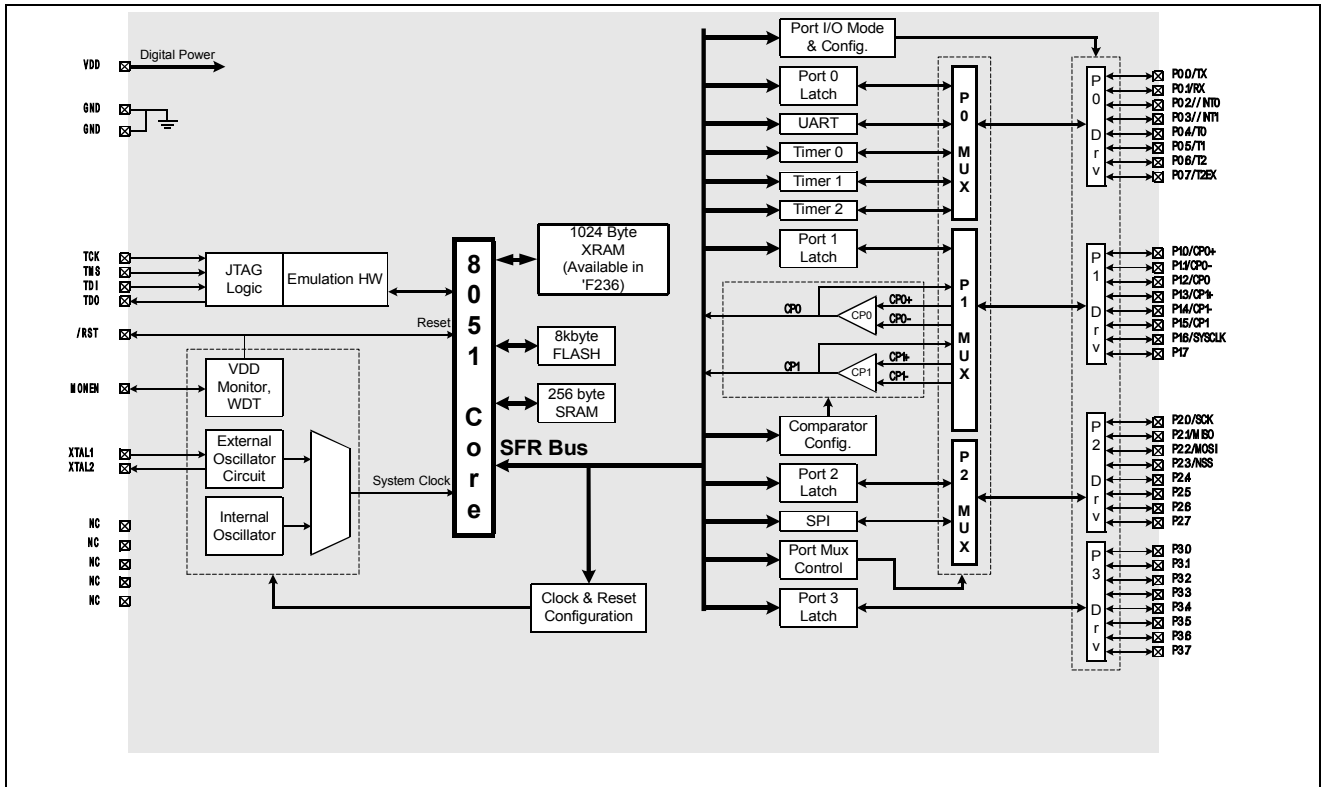
External Oscillator Circuit 外部振荡器电路

MUX 多路转换器

SFR 特殊功能寄存器

Clock & Reset Configuration 时钟, 复位配置

图 1.3. C8051F230 和 C8051F236 框图 (48TQFP)



Digital Power 数字电源

JTAG logic JTAG 逻辑

Emulation HW 仿真 HW

Latch 锁存器

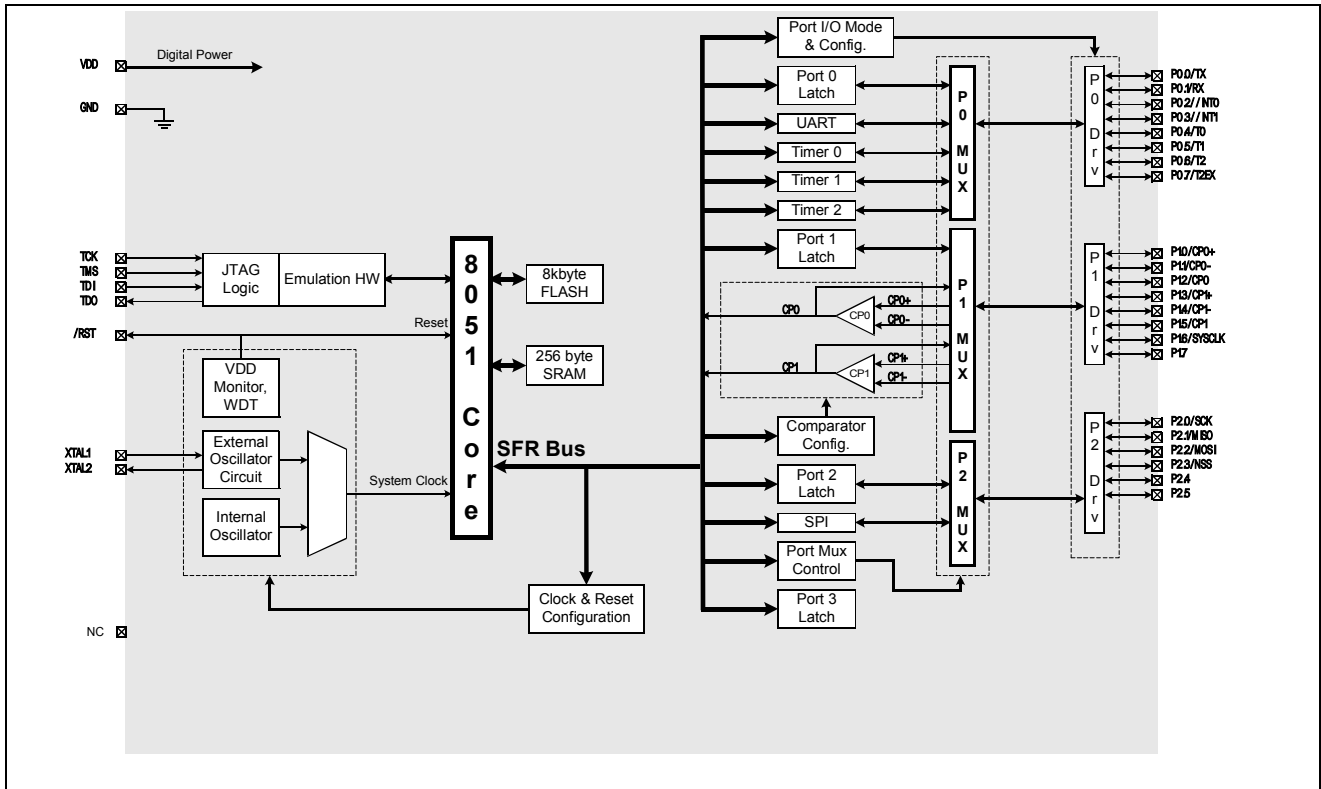
External Oscillator Circuit 外部振荡器电路

MUX 多路转换器

SFR 特殊功能寄存器

Clock & Reset Configuration 时钟, 复位配置

图 1.4. C8051F231 框图 (32LQFP)



Digital Power 数字电源

JTAG logic JTAG 逻辑

Emulation HW 仿真 HW

Latch 锁存器

External Oscillator Circuit 外部振荡器电路

MUX 多路转换器

SFR 特殊功能寄存器

Clock & Reset Configuration 时钟, 复位配置

## 1.1. CIP-51™ 微控制器内核

### 1.1.1. 与 8051 完全兼容

C8051F206, C8051F220/1/6 和 C8051F230/1/6 使用 CYGNAL 的专利 CIP-51 微控制器内核, CIP-51 与 MCS-51™ 的指令集完全兼容, 标准 803x/805x 的汇编和编译程序可用于软件开发。CIP-51 内核包含标准 8052 的一些外设部件, 有 3 个 16 位的计数器 / 定时器, 1 个全双工的通用异步收发器 (UART), 256 字节的内部 RAM, 可选用的 1024 字节的 XRAM, 128 字节的特殊功能寄存器 (SFR) 地址空间以及 4 字节宽的 I/O 端口。

### 1.1.2. 指令处理能力提高

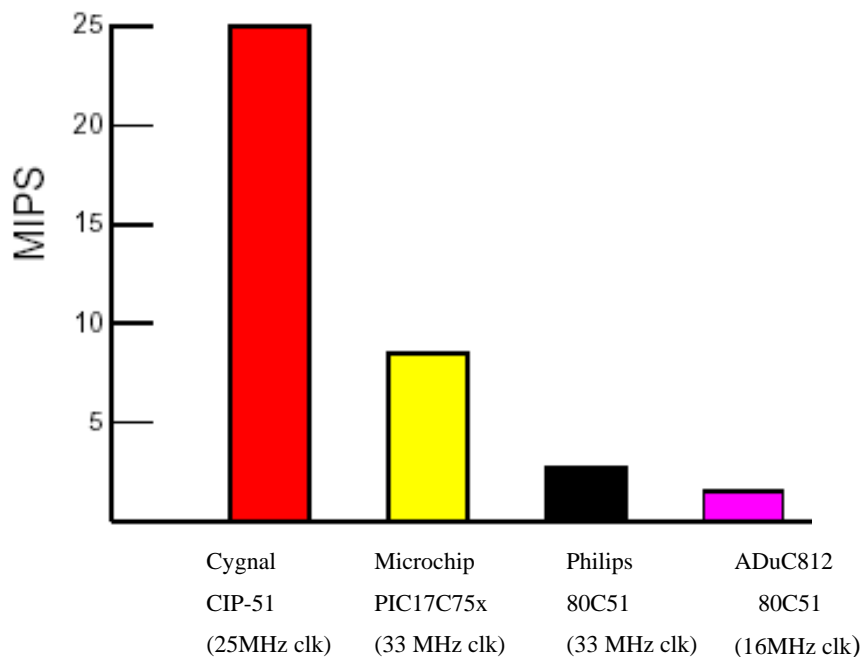
CIP-51 采用流水线式的结构, 大大增加了它的指令处理能力, 这一点超过了标准的 8051。在一个标准的 8051 中, 除了 MUL 和 DIV 两条指令外, 在 12MHz 的最大系统时钟下, 所有的指令均要占用 12 或 24 个系统时钟周期。作个对比, CIP-51 的内核执行它 70% 的指令只用 1 或 2 个系统时钟周期, 只有 4 条指令占用多于 4 个的系统时钟周期。

CIP-51 共有 109 条指令, 指令数目对应于执行这些指令所需的系统时钟周期如下表所示:

指令数	26	50	5	14	7	3	1	2	1
所需时钟	1	2	2/3	3	3/4	4	4/5	5	8

CIP-51 的最大系统时钟是 25MHz, 其指令处理能力峰值可达到 25MIPS (每秒百万条指令)。图 1.5 示出了几种不同的 8 位微控制器内核在其最大系统时钟情况下的指令处理能力峰值的比较。

**图 1.5. MCU 指令处理能力峰值的比较**



### 1.1.3. 其它特性

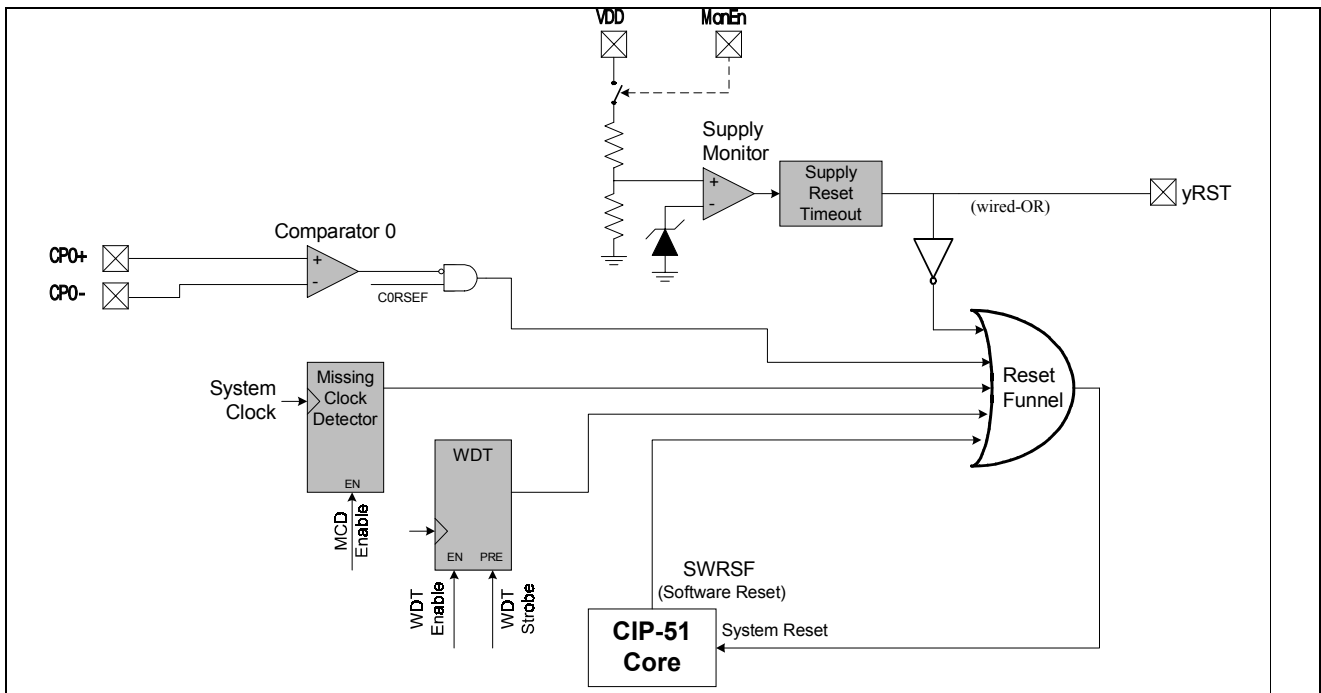
C8051F206, C8051F220/1/6 和 C8051F230/1/6 在 CIP-51 的核内外有几个关键性的改进, 提高了其在终端应用的总体性能及易用性。

扩展的中断处理器提供了 22 个中断源给 CIP-51 (相对于 8051 的 7 个中断源), 允许多个模拟、数字量外设部件去向控制器请求中断。(一个中断驱动系统不需 MCU 过多的干涉, 就会使之有更多有效的指令处理能力) 在建立一个多任务实时系统的时候, 附加的中断源非常有用。

MCU 多至 6 个复位源: 1 个片内 VDD 监视器, 1 个看门狗定时器, 1 个时钟丢失检测器, 1 个来自比较器 0 的电平检查, 1 个软件强制复位以及 1 个外部复位端子。/RST 端子是双向式的, 可引进外部复位信号, 也可允许内部发生的复位在 /RST 端子上输出。上拉 MONEN 端子至高 (数字 1) 时, 就给了片内 VDD 监视器一个使能信号。除了 VDD 监视器和由软件来的复位输入外, 用户可以禁止其它的每一个复位源。在 MCU 初始化期间上电复位之后, 看门狗定时器可以用软件使之永久地被使能。

MCU 有 1 个独立的内部时钟发生器, 它作为复位之后默认的系统时钟。若有必要, 时钟源能在程序运行过程中切换至外部振荡器以产生系统时钟, 外部振荡器可以是晶体式的, 陶瓷谐振式的, 电容式的, 阻容式的或外部时钟源。这一点在低功耗应用中很有用, 可以允许 MCU 在低频 (节电) 外部晶体源中运行起来, 需要时, 周期性地切换至高频 (频率可达 16MHz) 的内部振荡器。

图 1.6. 片内时钟和复位



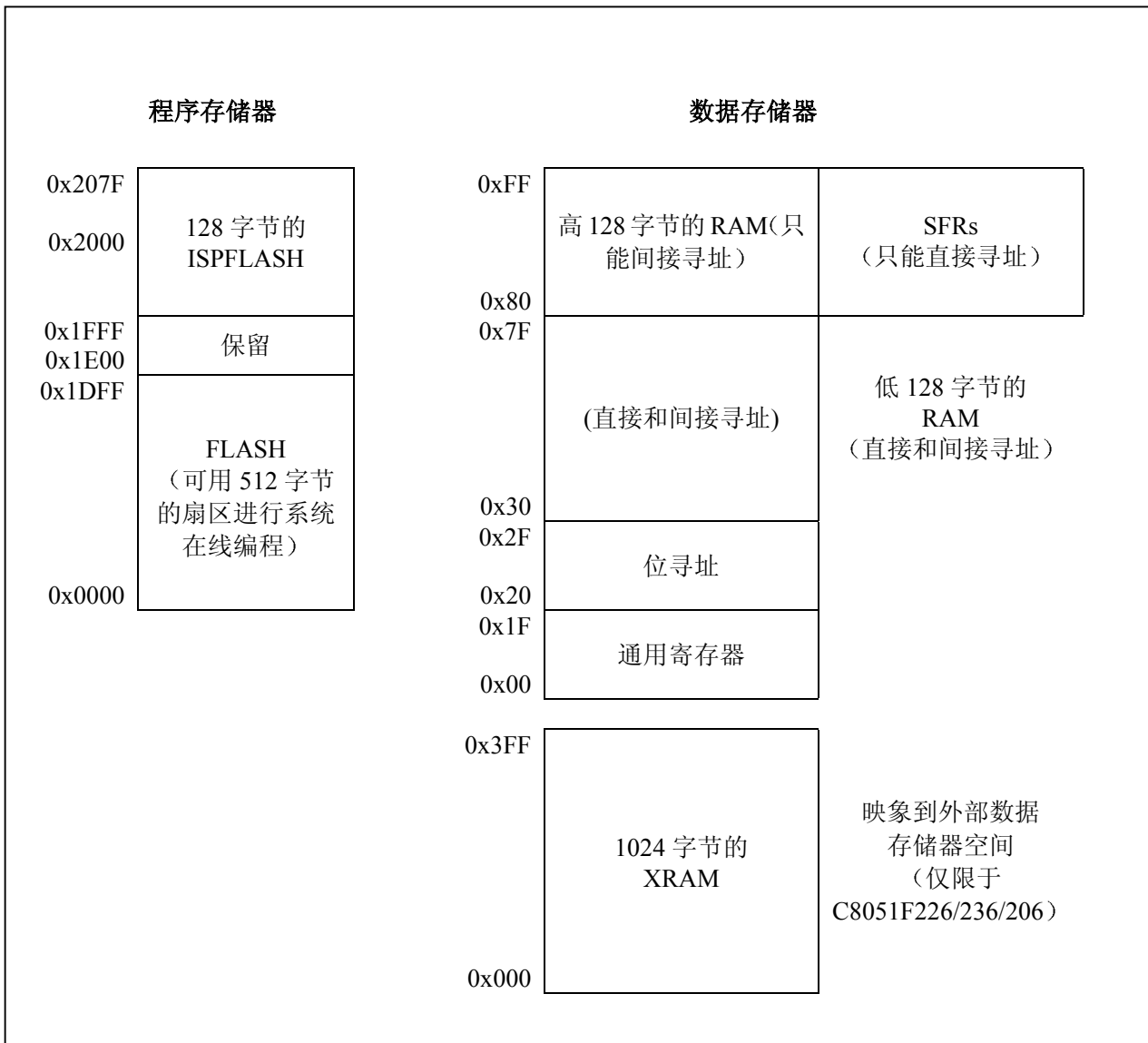
- Strobe 选通
- Reset Funnel 复位综合
- wired-OR “或”功能线
- Supply 供电
- Timeout 超时
- Supply Monitor 电源监视器
- Comparator 0 比较器 0
- Missing Clock Detector 时钟丢失检测器

## 1.2. 片内存储器

CIP-51 具有标准 8051 的程序和数据地址配置。包括 256 字节的数据 RAM，它高端的 128 字节是双映象的。在 F206, F226, F236 上有一个可选的 1024 字节的 XRAM。间接寻址访问通用 RAM 上层的 128 字节，而直接寻址则访问 128 字节的特殊功能寄存器 (SFR) 地址空间。通过直接寻址或间接寻址可访问 RAM 的低 128 字节，它的第一个 32 字节可作为 4 组通用寄存器的寻址，紧接着的 16 字节能用于字节寻址或位寻址。

MCU 的程序存储器包含 8k+128 字节的 FLASH, 可用 512 字节的扇区来对该存储器进行系统在线重复编程，且不需额外的编程电压。从地址 0x1E000 到 0x1FFF 共 512 字节的扇区为工厂使用而保留，从地址 0x2000 到 0x207F 共 128 字节的扇区空间是为用户编程用的，也可用于存储软件常量表、固定的配置信息或作为附加的程序空间。MCU 系统存储器映象如图 1.7 所示：

图 1.7. 片内存储器映象

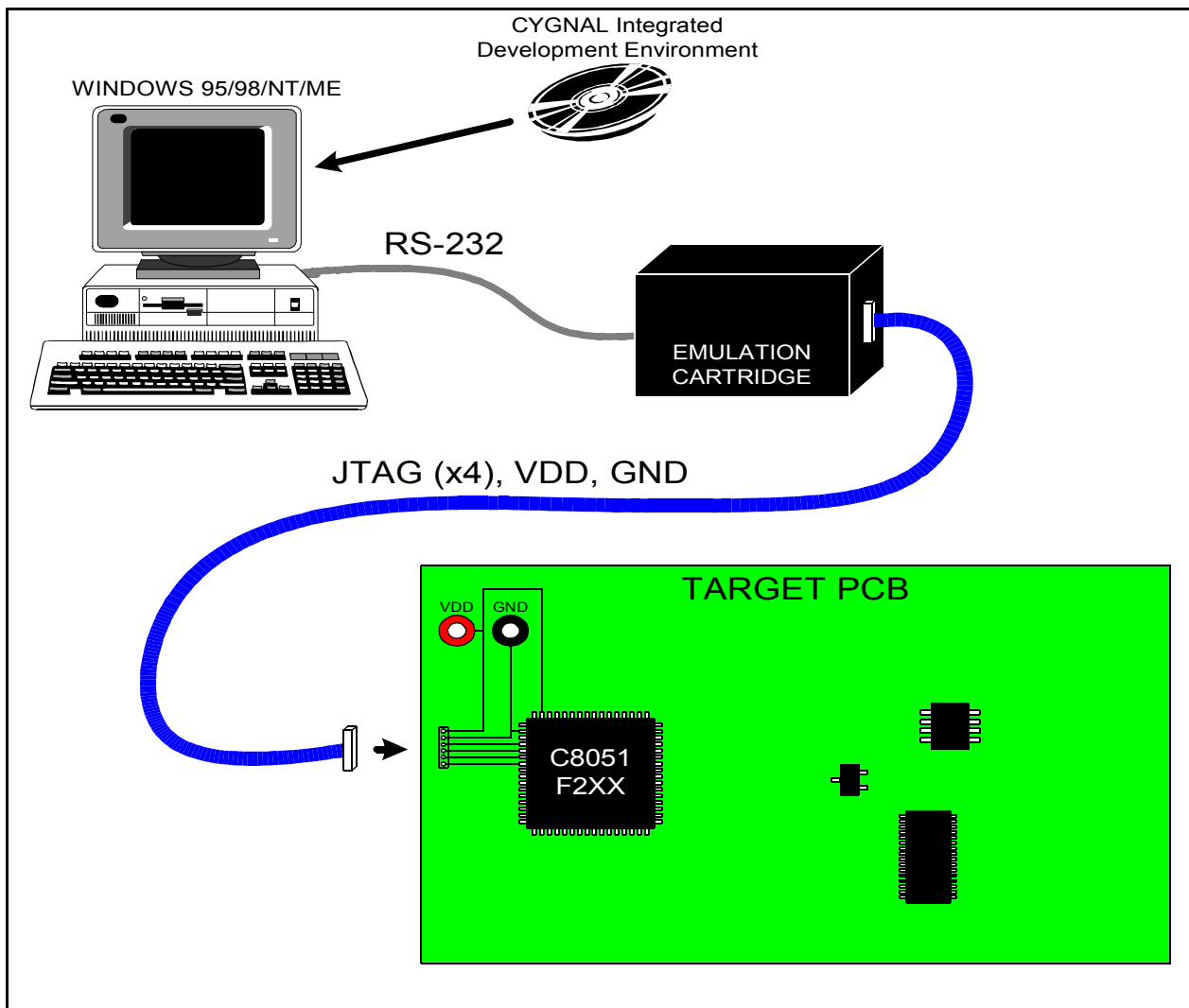


### 1.3. JTAG

C8051F2xx 有片内 JTAG 和调试逻辑, 提供非插入式、全速的工作, 电路调试使用安装了最终应用程序的产品, 调试时使用 4 脚 JTAG I/F。C8051F2xxDK 是一个开发工具包, 具有所有必需的软、硬件, 它和 C8051F2xx 一起开发应用程序代码, 实现在线调试。这个工具包包括一个开发者工作室所需的软件、调试程序、一个集成的 8051 汇编程序, 一个 RS-232 到 JTAG 的接口模块 EC。同时还具有一个带有 C8051F2xx 的目标应用板和较大的原型区, 加上 RS-232, JTAG 电缆和固定在墙上的电源。开发工具要求一台装有 Windows9x, NT 或 ME 并且有一个可用的 RS-232 串口的计算机。由图 1.8 可见, PC 机通过 RS-232 连接到 EC, EC 到用户应用程序电路板则通过 6 英寸的带状电缆连接, 它包括 4 根 JTAG 引脚, 1 根 VDD, 1 根 GND。EC 从应用程序电路板取电源, 电压大约是 2.7-3.6V, 电流 20mA。对应用来说, 并不是一定要从目标板上取电源, 也可以另行提供一个电源直接供给 EC。

与标准的 MCU 仿真器相比, 对开发和调试嵌入式的应用程序, 这是一个极其优越的配置。这种配置需用到电路板上的 ICE 芯片和目标电缆, 且需将 MCU 插入应用电路板中。CYGNAL 的调试环境不仅增加了其易用性, 且可保持精密模拟外设部件的性能

图 1.8. 调试环境框图



CYGNAL Integrated Development Environment CYGNAL集成开发环境  
EMULATION CARTRIDGE 仿真盒 (EC)



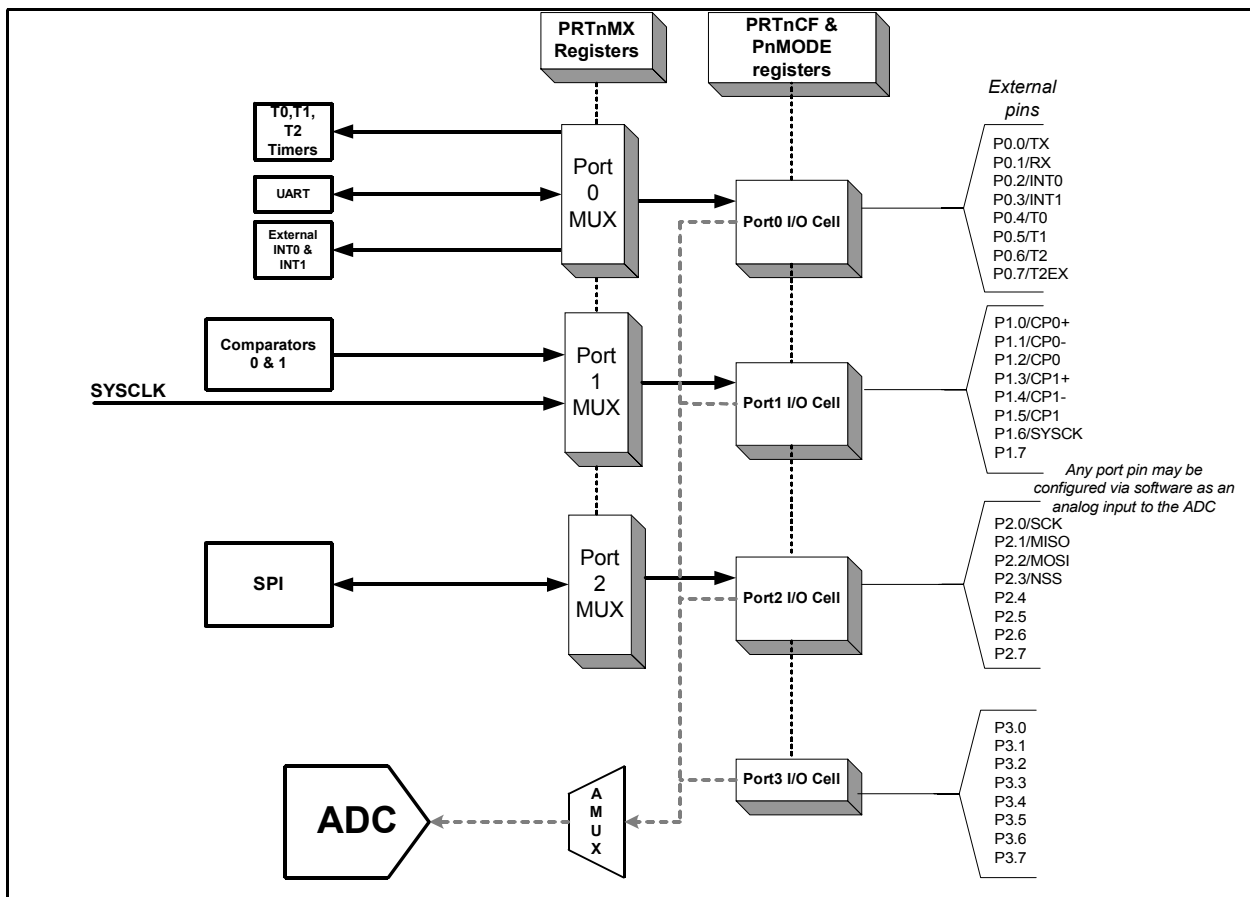
## 1.4. 数字/模拟可配置的 I/O

在设备上有标准的8051端口（0, 1, 2和3），这些端口是一些增强功能的标准8051端口。

每个端口的端子都能配置成1个推挽式输出或开漏极输出。可配置成模拟量输入的任一个输入端子，且有其对应的弱上拉关断功能。

数字信号源（定时器，串行外设接口，通用异步收发器，系统时钟和比较器）由配置的多路转换器端口发送至对应的 I/O 端子，而多路转换器的端口可由特殊功能寄存器 (SFR) 的置位功能来编程 (请看 14 节)。32 个外部端口端子的任意一个都可以配置成模拟量输入或数字量 I/O (见图 1.9)，这样，所有的端子都具有双向功能。

图 1.9. I / O 端口功能框图



*Any port pin may be configured via software as an analog input to the ADC*

任意一个端子都可用软件配置成模拟量输入至 ADC

## 1.5. 串行端口

C8051F206, C8051F220/1/6 和 C8051F230/1/6 包括一个全双工的通用异步收发器 (UART) 和串行外围设备接口 (SPI) 总线。串行总线都是用硬件实现，广泛使用 CIP-51 的中断，因而几乎不需 CPU 的介入。串行总线不必“共享”如定时器，中断，或 I/O 端口等资源，这样两种串行总线可以同时使用。(你可以用定时器 1，定时器 2 或系统时钟为 UART 产生波特率)

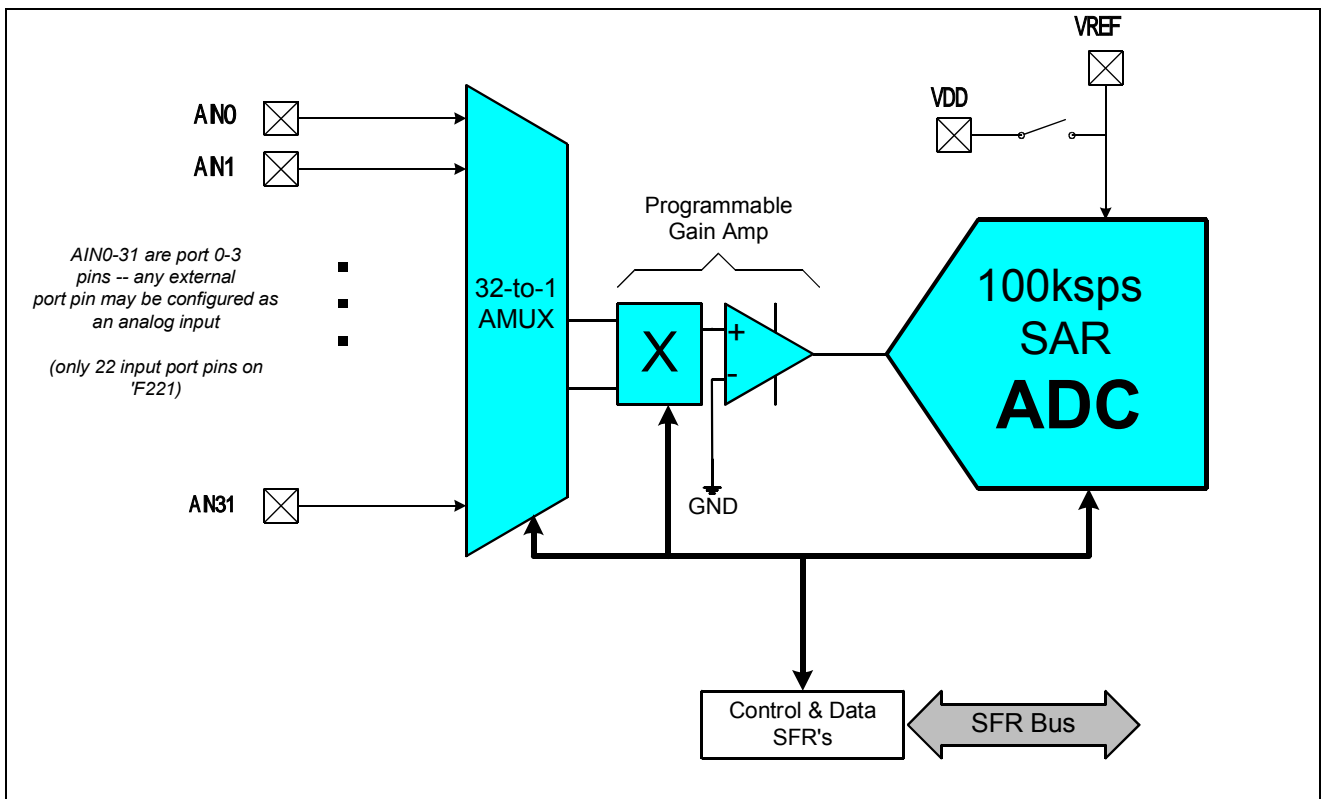
### 1.6. 模数转换器

C8051F220/1/6 有一个片内 8 位的 SAR 模数转换器 (ADC), C8051F206 有一个 12 位的 SAR ADC, 都带有一个可编程增益放大器 (PGA)。ADC 的最大指令处理能力是 100ksp/s, 有两种精度的 ADC: 一个具有  $\pm 1/4$  最低有效位 (LSB) INL 的真 8 位精度, 一个是具有  $\pm 2$ LSB 的 12 位精度。基准电压可以是电源 (VDD), 也可以是外部基准电压 (VREF), 不需使用 ADC 时, 系统控制器能将之置于一个节能的关断模式之中, 模拟多路转换器的输出到程控增益放大器 (PGA), PGA 的增益能用软件以 2 的幂从 0.5 到 16 来设置。

转换能通过两种方式初始化: 一种是软件命令, 一种是定时器 2 的溢出。这种灵活性允许用软件事件去触发转换的开始, 或者连续地进行转换。转换完成会触发中断, 或通过软件将状态位置位以确定转换的完成。转换完成后, 所得的 8 位数据被锁存至特殊功能寄存器中。

ADC 数据可通过一个可编程的窗口检测器进行连续地监视, 当数据处于用户程序窗口时, 检测器会中断 CPU。这样 ADC 就可不必使用 CPU 资源, 而以后台模式去监控系统的主要电压。

图 1.10. 模数转换器框图



*AIN0-31 are port 0-3 pins -- any external port pin may be configured as an analog input*

AIN0-31 是端口 0-3 的端子, 任意一个外部端口的端子都能配置为模拟量输入

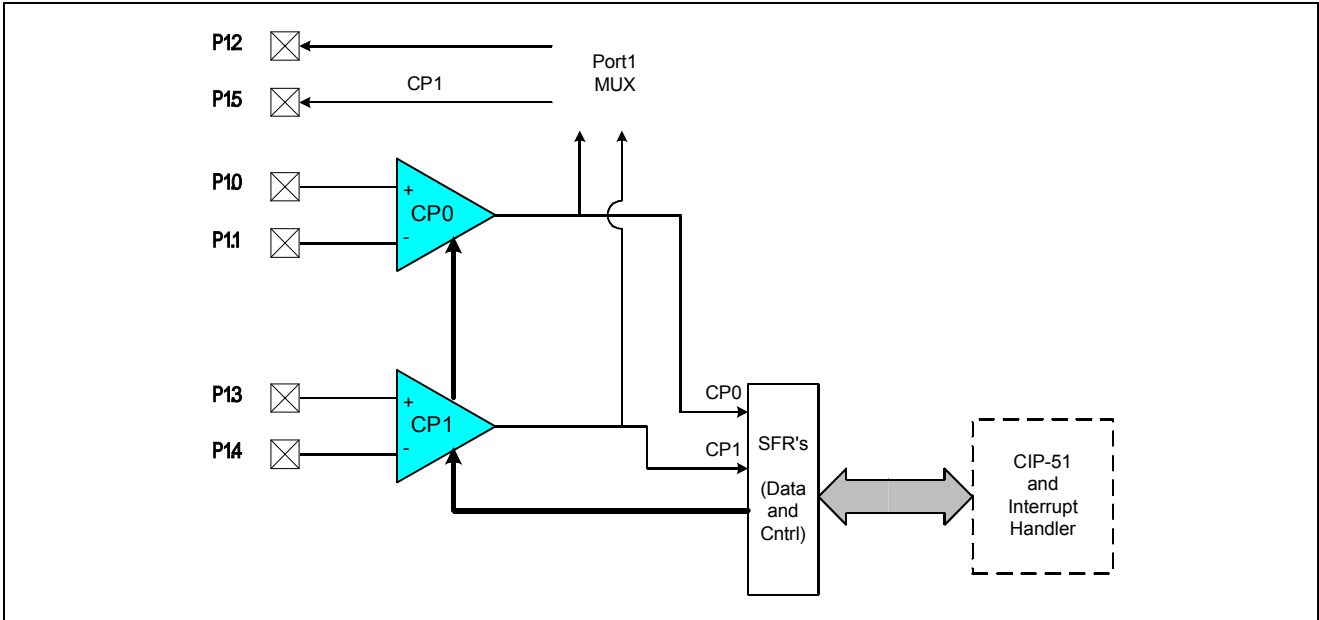
Programmable Gain Amp 可编程增益放大器

Control & Data SFR's SFR的控制和数据

### 1.7. 比较器

MCU 有 2 个片内电压比较器。作为例子，图 1.11 中比较器的输入引至封装的端子上。每个比较器的滞后通过特殊功能寄存器（SFR）用软件编程来实现。用户能容易地编程实现电压等级和正 / 负对称度。另外，比较器的中断可以在上升沿或下降沿输出跃变时发生。请看第 8 节的详述。

图 1.11. 比较器框图



### 2. 最大绝对额定值\*

倾斜情况下的环境温度	-55—125°C
储存温度	-65—150°C
任意一个端子（除 VDD 和 I/O 口外）对 DGND 的电压	-0.3V—(VDD+0.3V)
任意一个 I/O 口端子或 / RST 对 DGND 的电压	-0.3V—5.8V
VDD 相对 DGND 的电压	-0.3—4.2V
总功耗	1.0W
任意一个端口端子吸收的最大输出电流	200mA
其它任意一个 I/O 端子吸收的最大输出电流	25mA
任意一个端口端子发出的最大输出电流	200mA
其它任意一个 I/O 端子发出的最大输出电流	25mA

\*注：以上所列强调在“最大绝对额定值”下有可能对设备引起永久性的损害，这只是强调设备在上述条件下的额定值和功能操作，不是上述及以外条件下的操作列表。若长期在最大额定值下工作将影响设备的可靠性。

### 3. 全局直流电特性

温度范围：-40℃---+85℃（除了特别说明以外）

参量	条件	最小值	典型值	最大值	单位
电源电压	注 1	2.7		3.6	V
ADC、比较器和 CPU 激活时 VDD 电流	时钟是 25MHz		9.5		mA
	时钟是 1MHz		3.6		
	时钟是 32MHz		125		μ A
ADC、比较器激活而 CPU 未激活时（闲置模式）的 VDD 电流	时钟是 25MHz		5		mA
	时钟是 1MHz		1.8		
	时钟是 32MHz		125		μ A
ADC、比较器未激活而 CPU 激活时的 VDD 电流	时钟是 25MHz		9		mA
	时钟是 1MHz		1		
	时钟是 32MHz		20		μ A
CPU 未激活（闲置模式）时数字电源电流	时钟是 25MHz		4.5		mA
	时钟是 1MHz		0.1		
	时钟是 32MHz		10		μ A
数字电源电流（停止模式），VDD 监视器被使能	振荡器未运行		10		μ A
数字电源电流（停止模式）VDD 监视被禁止	振荡器未运行		0.1		μ A
RAM 数据保持的数字电源电压			1.5		V
指定的操作温度范围		-40		+85	℃

注 1：电源必须大于 1V，对 VDD 监视器的操作来说，MONEN 端子必须拉成高电平。

## 4. 输出端子和封装定义

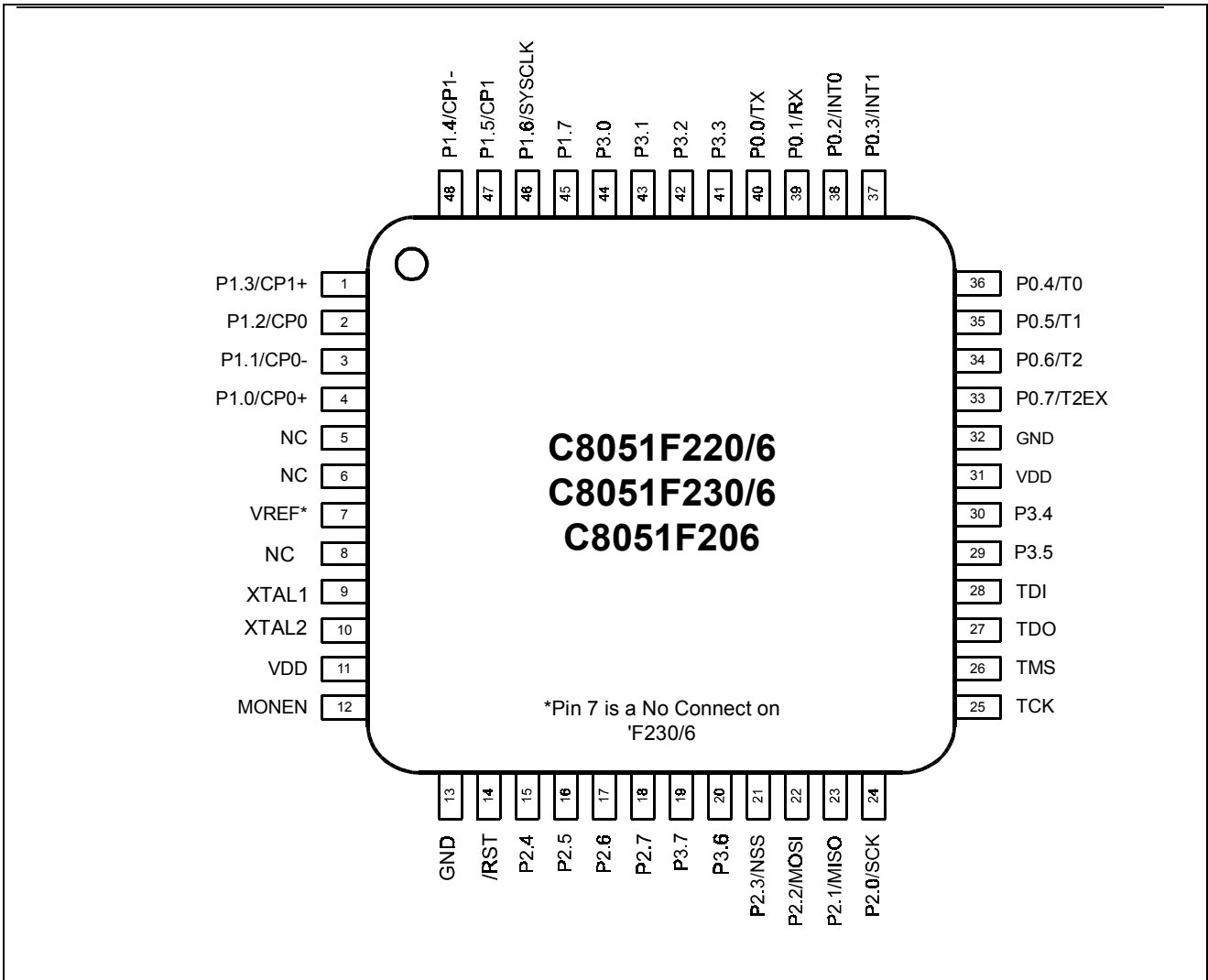
表 4.1. 端子定义

名称	F220, F226 F230, F236	F221, F231	类型	说明
	48 脚	32 脚		
VDD	11, 31	8		数字电压供电
GND	5, 6, 8, 13, 32	9		地（注：48 脚上的 5, 6, 8 脚未连接（NC），推荐将它们连接至地）
MONEN	12		D In	监视器使能（只对 48 脚封装）。当输入为高（逻辑 1）时，允许复位电压监视器的功能
TCK	25	17	D In	带内部上拉的 JTAG 测试时钟
TMS	26	18	D In	带内部上拉的 JTAG 测试模式选择
TDI	28	20	D In	带内部上拉的 JTAG 测试数据输入。TDI 被锁存在 TCK 的上升沿
TDO	27	19	D Out	JTAG 测试数据输出。在 TCK 的下降沿它移出数据，TDO 输出是一个三态驱动器
XTAL1	9	6	A In	晶体输入。此端子返回到一个晶体或陶瓷谐振器做的内部振荡器电路。对一个预先设定的内部时钟，从 XTAL1 连接一个晶体或陶瓷谐振器到 XTAL2。被一个外部 CMOS 时钟过度激励时可成为系统时钟
XTAL2	10	7	A Out	晶体输出。此端子是晶体或陶瓷谐振器的激发驱动器
/ RST	14	10	D I/O	芯片复位。内部电压供电监视器的开漏极输出，当 VDD < 2.7V 和 MONEN = 1，或者当 1 被写进 PORSF 时，该脚为低。一个外部信号源将此端子变成低时，能强制系统复位
VREF	7	5	A I/O	基准电压。当配置成一个输入时，该端子是 ADC 的基准电压，否则，VDD 是基准电压。注：此脚在 F230 / 1 / 6 上未连接（NC）
CP0+	4	4	A In	比较器 0 的非反相输入
CP0-	3	3	A In	比较器 0 的反相输入
CP0	2	2	D Out	比较器 0 的输出
CP1+	1	1	A In	比较器 1 的非相输入
CP1-	48	32	A In	比较器 1 的反相输入
CP1	47	31	D Out	比较器 1 的输出
P0. 0/ TX	40	28	D I/O A In	端口 0 的第 0 位（完整的说明见 I / O 端口子系统一节）
P0. 1/ RX	39	27	D I/O A In	端口 0 的第 1 位（完整的说明见 I / O 端口子系统一节）
P0. 2/ INT0	38	26	D I/O A In	端口 0 的第 2 位（完整的说明见 I / O 端口子系统一节）
P0. 3/ INT1	37	25	D I/O A In	端口 0 的第 3 位（完整的说明见 I / O 端口子系统一节）
P0. 4/ T0	36	24	D I/O A In	端口 0 的第 4 位（完整的说明见 I / O 端口子系统一节）
P0. 5/ T1	35	23	D I/O A In	端口 0 的第 5 位（完整的说明见 I / O 端口子系统一节）
P0. 6/ T2	34	22	D I/O A In	端口 0 的第 6 位（完整的说明见 I / O 端口子系统一节）
P0. 7/ T2EX	33	21	D I/O A In	端口 0 的第 7 位（完整的说明见 I / O 端口子系统一节）
P1. 0/ CP0+	4	4	D I/O A In	端口 1 的第 0 位（完整的说明见 I / O 端口子系统一节）



名称	F220, F226 F230, F236	F221, F231	类型	说明
	48脚	32脚		
P1.1/ CP0-	3	3	D I/O A In	端口1的第1位（完整的说明见I/O端口子第一节）
P1.2/ CP0+	2	2	D I/O A In	端口1的第2位（完整的说明见I/O端口子第一节）
P1.3/ CP0+	1	1	D I/O A In	端口1的第3位（完整的说明见I/O端口子第一节）
P1.4/ CP0+	48	32	D I/O A In	端口1的第4位（完整的说明见I/O端口子第一节）
P1.5/ CP0+	47	31	D I/O A In	端口1的第5位（完整的说明见I/O端口子第一节）
P1.6/ CP0+	46	30	D I/O A In	端口1的第6位（完整的说明见I/O端口子第一节）
P1.7/ CP0+	45	29	D I/O A In	端口1的第7位（完整的说明见I/O端口子第一节）
P2.0/ CP0+	24	16	D I/O A In	端口2的第0位（完整的说明见I/O端口子第一节）
P2.1/ CP0+	23	15	D I/O A In	端口2的第1位（完整的说明见I/O端口子第一节）
P2.2/ CP0+	22	14	D I/O A In	端口2的第2位（完整的说明见I/O端口子第一节）
P2.3/ CP0+	21	13	D I/O A In	端口2的第3位（完整的说明见I/O端口子第一节）
P2.4	15	11	D I/O A In	端口2的第4位（完整的说明见I/O端口子第一节）
P2.5	16	12	D I/O A In	端口2的第5位（完整的说明见I/O端口子第一节）
P2.6	17		D I/O A In	端口2的第6位（完整的说明见I/O端口子第一节）
P2.7	18		D I/O A In	端口2的第7位（完整的说明见I/O端口子第一节）
P3.0	44		D I/O A In	端口3的第0位（完整的说明见I/O端口子第一节）
P3.1	43		D I/O A In	端口3的第1位（完整的说明见I/O端口子第一节）
P3.2	42		D I/O A In	端口3的第2位（完整的说明见I/O端口子第一节）
P3.3	41		D I/O A In	端口3的第3位（完整的说明见I/O端口子第一节）
P3.4	30		D I/O A In	端口3的第4位（完整的说明见I/O端口子第一节）
P3.5	29		D I/O A In	端口3的第5位（完整的说明见I/O端口子第一节）
P3.6	20		D I/O A In	端口3的第6位（完整的说明见I/O端口子第一节）
P3.7	19		D I/O A In	端口3的第7位（完整的说明见I/O端口子第一节）

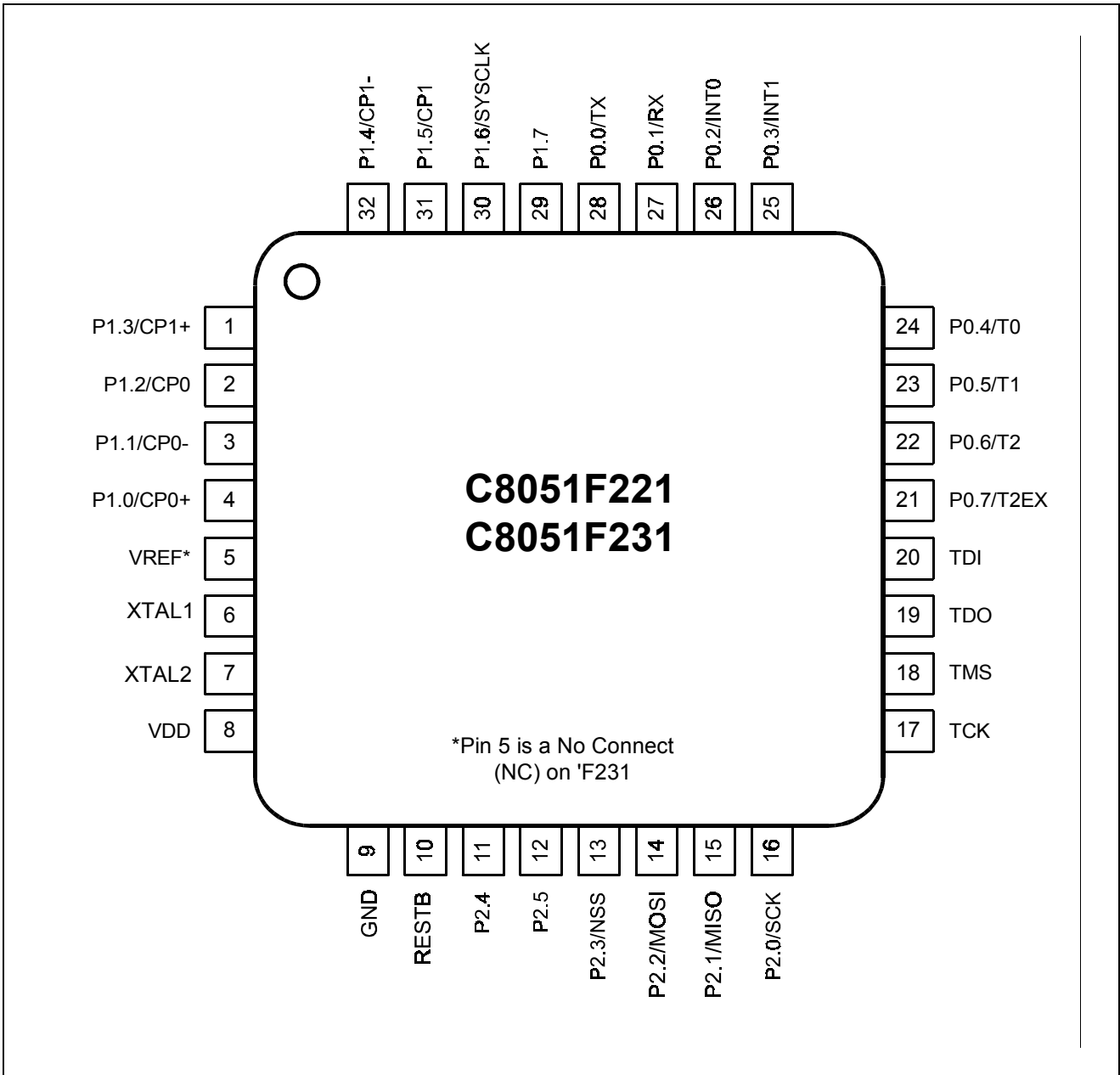
图 4.1. TQFP-48 端子图



Pin 7 is No Connect on 'F230/6 在 F230 / 6 上端子 7 未连接

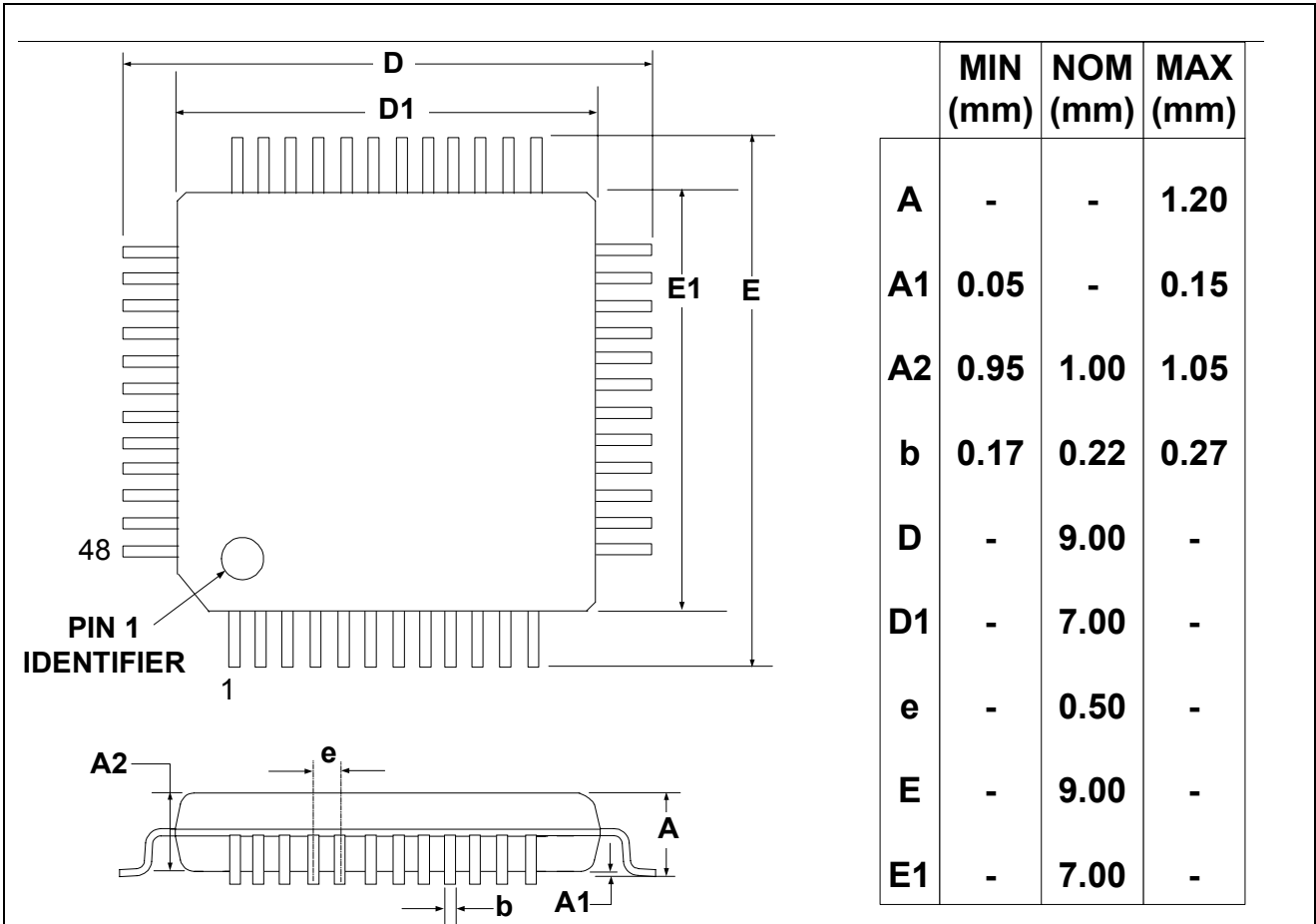


图 4.2. LQFP-32 端子图



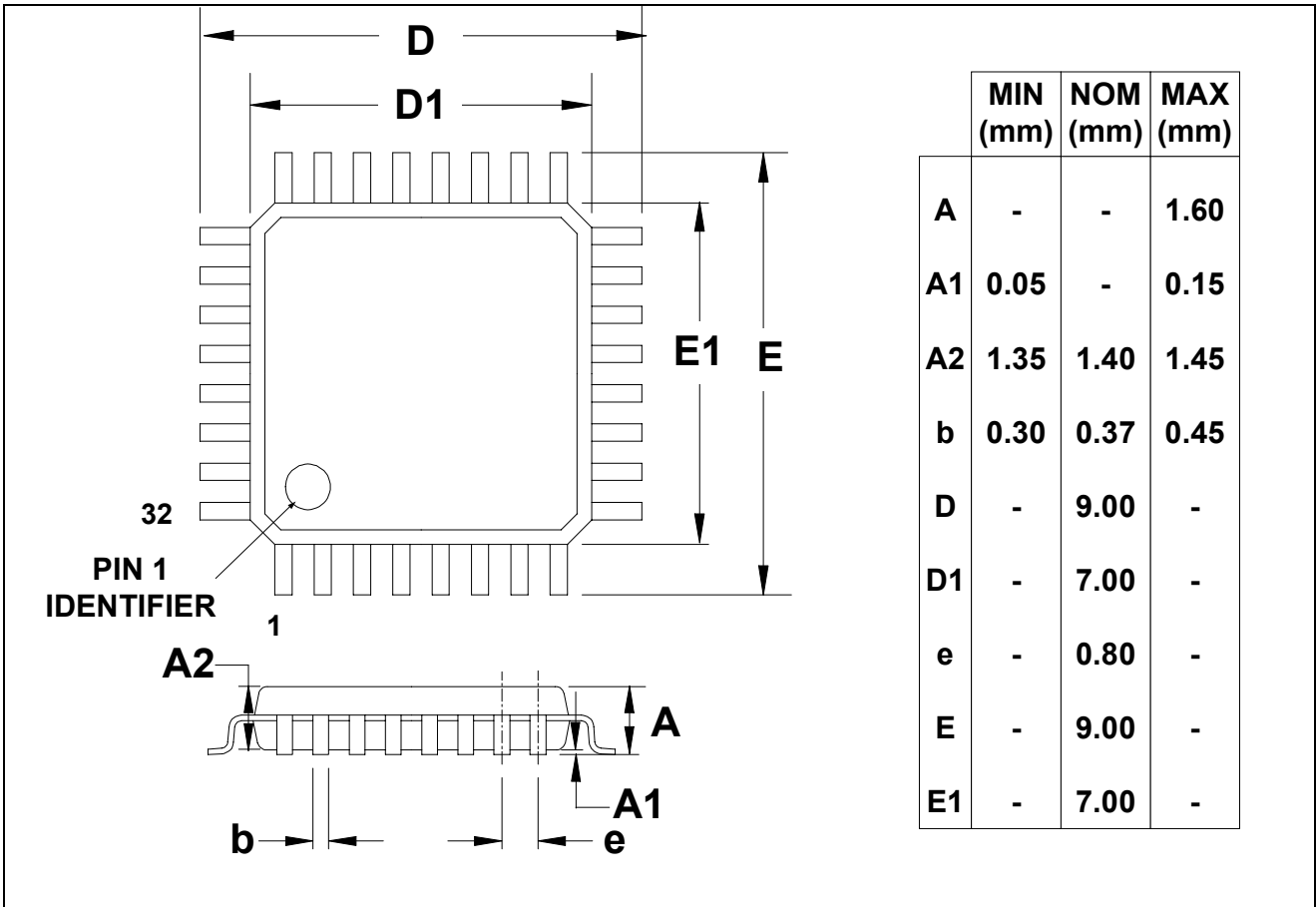
Pin 5 is a Connect on 'F231 在 F231 上端子 5 未连接

图 4.3. TQFP-48 封装绘图



Pin 1 IDENTIFIER 端子 1 标记

图 4. 4. LQFP-32 封装绘图



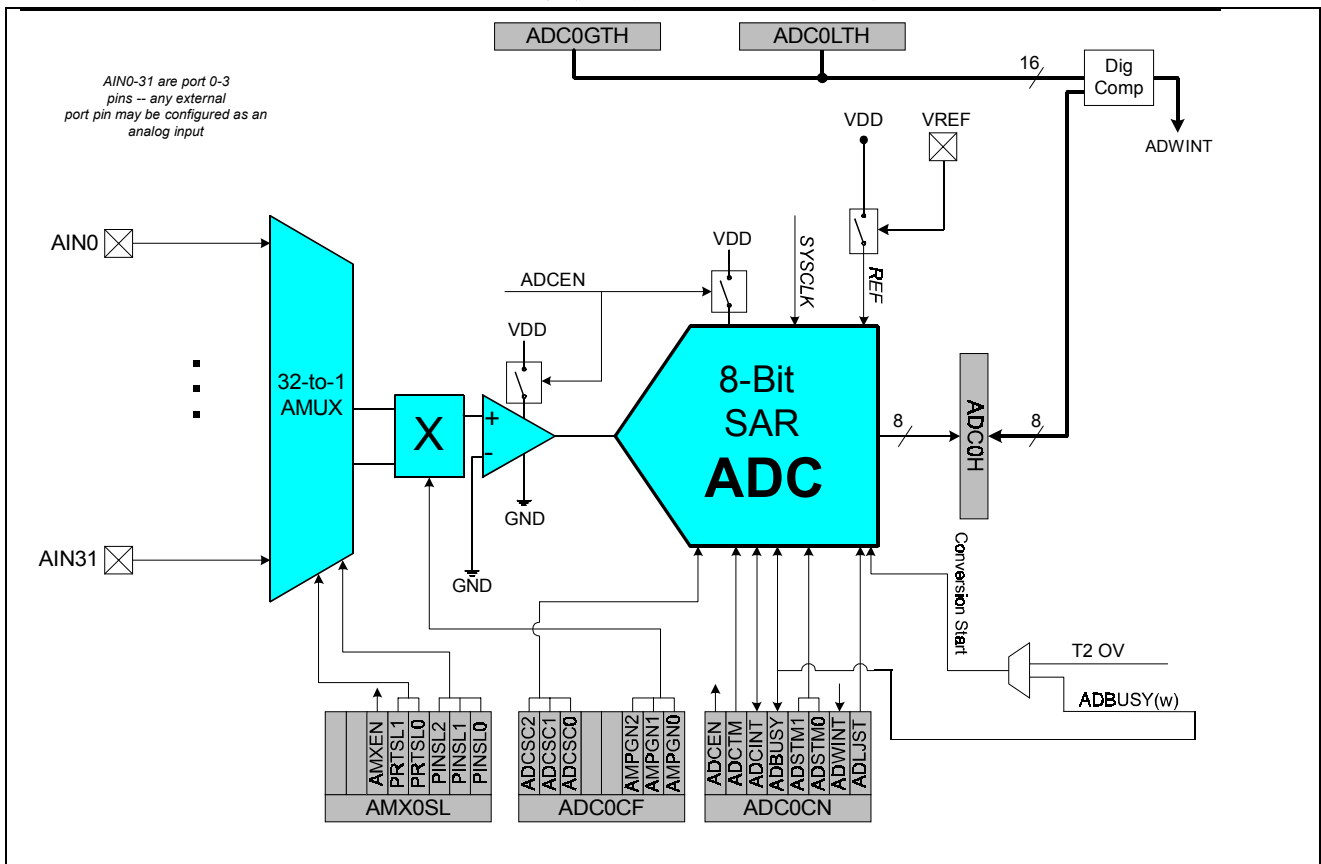
Pin 1 IDENTIFIER 端子 1 标记

## 5. 模数转换器 (ADC) (8 位, 仅限于 C8051F220/1/6)

### 说明:

C8051F220 / 1 / 6 的子系统 ADC 包含可配置的模拟量多路转换器 (AMUX), 1 个可编程增益放大器 (PGA) 和 1 个 100ksps 的 8 位逐次逼近式寄存器 ADC, 它具有集成的跟踪保持功能和可编程的窗口检测器 (见图 5. 1)。AMUX, PGA, 数据转换模式和窗口检测器都可通过图 5. 1 中的特殊功能寄存器由软件来配置。ADC 的子系统 (ADC, 跟踪保持和 PGA) 仅当 ADC 控制寄存器 (ADCCON, 图 5. 5) 的 ADCEN 位被置 1 时才被使能, 当该位是 0 时 ADC 子系统处于低功耗关断状态。

图 5. 1. 8 位模数转换器 (ADC) 功能框图



### 5. 1. 模拟量多路转换器 (AMUX) 和可编程增益放大器 (PGA)

任一个外部端子 (端口 0—3) 均可以通过软件来选择。AMX0SL 特殊功能寄存器可以选择期望的模拟量输入端子 (见图 5. 3)。当 AMUX 被使能时, 用户选择所要使用的端口 (位 PRTSL0—1), 然后被选择的端口 (位 PINSLO—2) 端子就是模拟量输入。

在外观上 AMUX 机能通过通道实现每种可能的配置。ADC 配置寄存器 ADC0CF 中 (图 5. 4), AMPGN2—0 的状态决定了 PGA 放大 AMUX 的输出信号的增益倍数。PGA 能用软件编程来实现其增益大小是 0.5, 1, 2, 4, 8, 或 16。复位后它的缺省增益是 1。

### 5. 2. 模数转换器 (ADC) 的操作模式

ADC 的最大转换速度是 100ksps。ADC 的转换时钟来自于系统时钟, 大小是系统时钟的除运算结果, 除数是 1, 2, 4, 8, 或 16, ADC0CF 寄存器中的位 ADCSC 来设置这个除数是几。在调整转换器的速度以适应不同的系统时钟速度时, 这一点很有用。

一个转换的初始化可用两种方式中的一种，这取决于 ADC0CN 中转换模式位 (ADSTM1 和 ADSTM0) 的 ADC 启动的编程状态。转换器的初始化可以是：

- 1 将 1 写到 ADC0CN 中的 ADBUSY 位；
- 2 定时器 2 溢出 (即连续转换计时)。

将 1 写到 ADBUSY 位提供了这样一种可能，就是立刻执行 ADC 如何转换的控制软件。在转换期间，ADBUSY 位置 1，转换结束时，该位恢复到 0。在 ADBUSY 的下降沿触发一个中断 (允许时)，并将 ADC0CN 寄存器的 ADCINT 置中断标志。转换的数据存在于 ADC 的数据字寄存器 ADC0H 中。

寄存器 ADC0CN 中的 ADCTM 位控制 ADC 的跟踪保持模式，缺省状态时，ADC 的输入是连续跟踪的，但转换正在进行时除外。将 ADCTM 置 1 允许两个不同的低功耗跟踪保持模式，这由 ADSTM1—0 位的状态确定 (ADC0CN 也如此)：

- 1 将 1 写至 ADBUSY 时开始跟踪，并持续 3 个 SAR 时钟周期；
- 2 定时器 2 溢出时开始跟踪，并持续 3 个 SAR 时钟周期。

当整个芯片是低功耗备用或睡眠模式时跟踪可以被禁止 (关断)。

**图 5.2. 12 位 ADC 跟踪和转换实例时序**

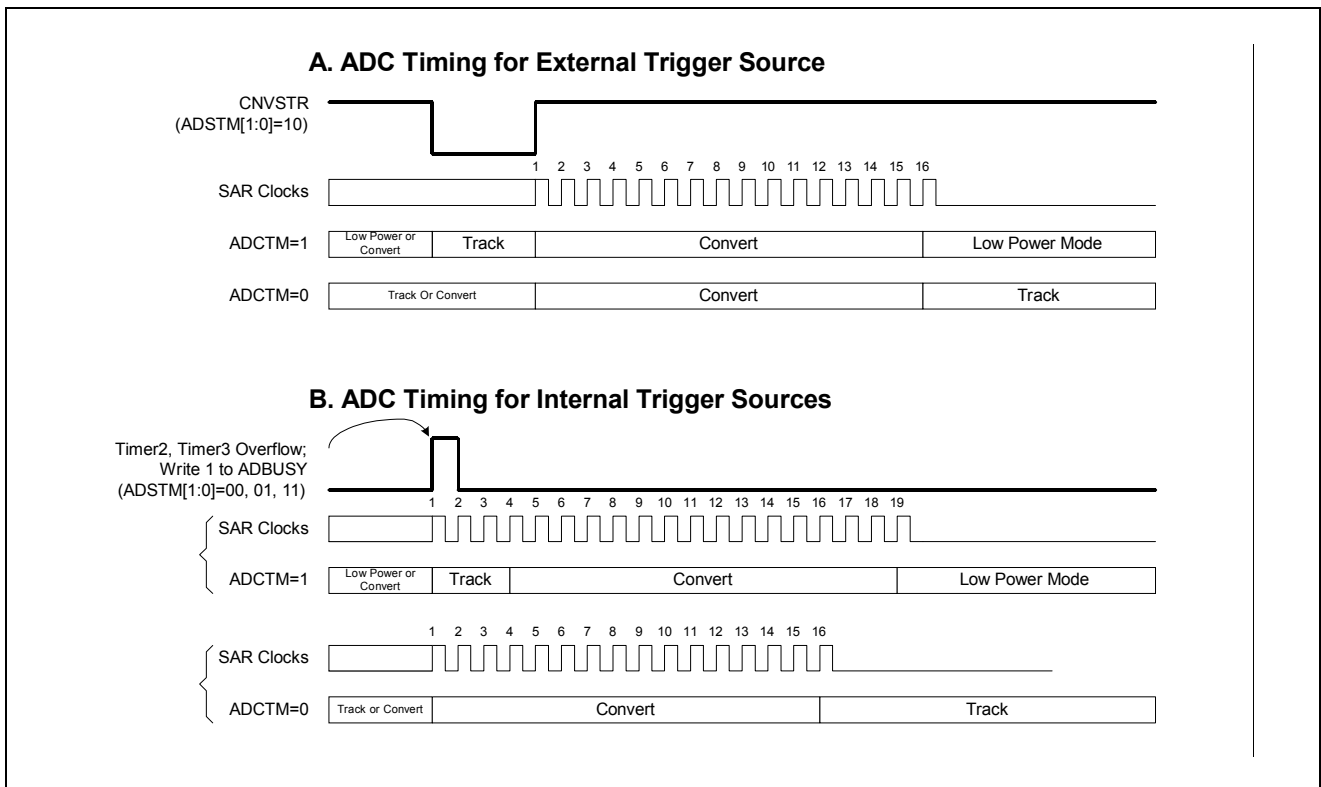


图 5.3. AMXOSL: AMUX 通道选择寄存器 (C8051F220/1/6 和 C8051F206)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值:00000000
-	-	AMXEN	PRTSL1	PRTSL0	PINSL2	PINSL1	PINSL0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址:0xBB

第 6, 7 位: 未用。读=00b;写=忽略

第 5 位: AMXEN 使能。

0: AMXEN 被禁止, 使用模拟量时, 端子无效。

1: AMXEN 被使能, 使用模拟量时, 可以使用 / 选择模拟量端子

第 3, 4 位: PRTSL1—0: 端口选择位 \*

00: 选择端口 0 去配置成模拟量输入端子;

01: 选择端口 1 去配置成模拟量输入端子;

10: 选择端口 2 去配置成模拟量输入端子;

11: 选择端口 3 去配置成模拟量输入端子。

第 0, 1, 2 位: PINSL2—0: 端子选择位

000: 上述所选择端口的端子 0 可用于模拟量输入;

001: 上述所选择端口的端子 1 可用于模拟量输入;

010: 上述所选择端口的端子 2 可用于模拟量输入;

011: 上述所选择端口的端子 3 可用于模拟量输入;

100: 上述所选择端口的端子 4 可用于模拟量输入;

101: 上述所选择端口的端子 5 可用于模拟量输入;

110: 上述所选择端口的端子 6 可用于模拟量输入;

111: 上述所选择端口的端子 7 可用于模拟量输入。

\* 选择一个端口作为模拟量输入并不是默认这个端口的所有端子均作模拟量输入用。在选择一个端口作为模拟量输入之后, 一个端子必须用端子选择位 (PINSL2—0) 去选择。例如: AMXEN 置 1 后, 置 PRTSL1—0 为 11, 再将 PINSL2—0 置 100, 这样端口 3 的第 4 个端子就被配置成为模拟量输入了。端口 3 的其它端子仍然是 GPIO 端子。应注意, 为了将一个端子作为模拟量输入使用, 它的输入方式应置为模拟量。请看 14.2 节。

**图 5.4. ADC0CF: ADC 配置寄存器 (C8051F220/1/6 和 C8051F206)**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值:01100000
ADCSC2	ADCSC1	ADCSC0	-	-	AMPGN2	AMPGN1	AMPGN0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址:0xBC

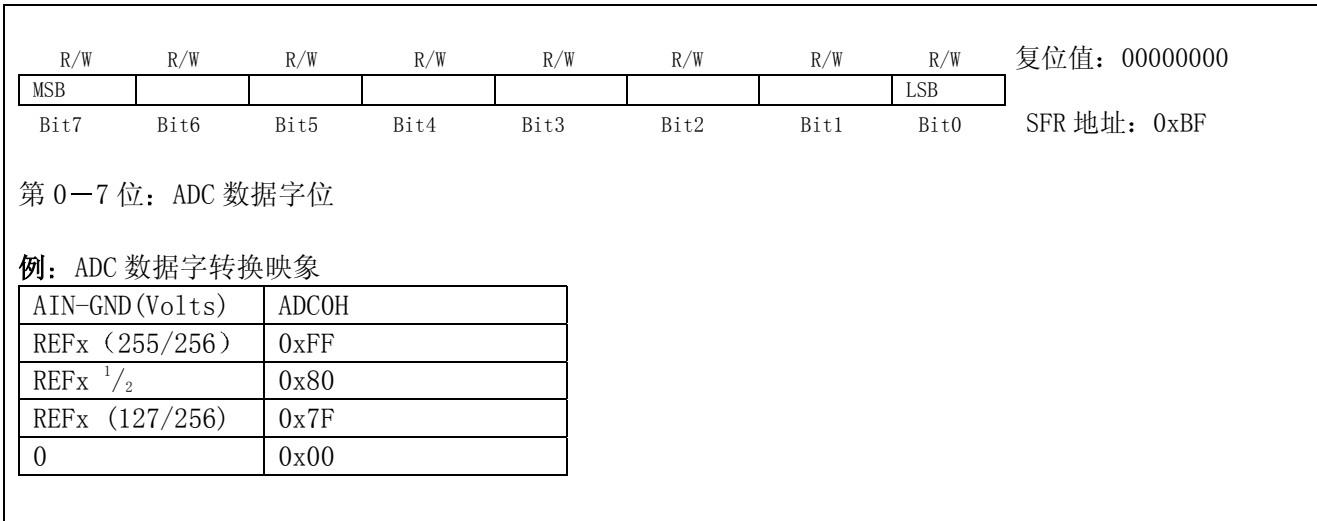
第 5, 6, 7 位: ADCSC2—0: ADC 逐次逼近式寄存器 (SAR) 转换时钟周期位  
     000: SAR 转换时钟=1 个系统时钟  
     001: SAR 转换时钟=2 个系统时钟  
     010: SAR 转换时钟=4 个系统时钟  
     011: SAR 转换时钟=8 个系统时钟  
     1xx: SAR 转换时钟=16 个系统时钟  
     注: SAR 转换时钟应该小于或等于 2MHz  
 第 3, 4 位: 未用。读=00b; 写=忽略  
 第 0, 1, 2 位: AMPGN2—0: ADC 内部放大器增益  
     000: 增益=1  
     001: 增益=2  
     010: 增益=4  
     011: 增益=8  
     10x: 增益=16  
     11x: 增益=0.5



图 5.5. ADC0CN: ADC 控制寄存器 (C8051F220/1/6 和 C8051F206)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值:00000000
ADCEN	ADCTM	ADCINT	ADBUSY	ADSTM1	ADSTMO	ADWINT	ADLIST	SFR 地址:0xE8
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	(位寻址)
<p>第 7 位: ADCEN: ADC 使能位            0: ADC 被禁止。ADC 处于低功耗关断模式            1: ADC 被使能。ADC 被激活准备数据转换</p> <p>第 6 位: ADCTM: ADC 的跟踪模式位            0: 当 ADC 被使能时, 跟踪是连续的, 除非转换正在进行之中            1: ADSTM1—0 位定义跟踪                ADSTM1—0:                00: 将 1 写至 ADBUSY 时开始跟踪, 并持续 SAR 的 3 个时钟                01: 保留                10: 保留                11: 定时器 2 溢出时开始跟踪, 并持续 SAR 的 3 个时钟</p> <p>第 5 位: ADCINT: ADC 转换结束中断标志 (由软件清除)            0: 从标志被清除的最后时刻开始, ADC 还未完成一个数据的转换            1: ADC 完成了数据的转换</p> <p>第 4 位: ADBUSY: ADC 忙碌位            读            0: ADC 转换完成, 或复位一次后, 无效数据被转换。ADBUSY 的下降沿在允许时产生一个中断            1: ADC 转换数据忙            写            0: 没有影响            1: 若 ADSTM1—0=00b, 则 ADC 开始转换</p> <p>第 2, 3 位: ADSTM1—0: ADC 转换模式的启动位            00: ADBUSY 置 1, 启动 ADC 转换            01: 保留            10: 保留            11: 定时器 2 溢出时 ADC 转换被初始化</p> <p>第 1 位: ADWINT: ADC 窗口比较中断标志            0: ADC 窗口比较数据不匹配            1: ADC 窗口比较数据匹配</p> <p>第 0 位: ADLJST: ADC 左校验数据位 (只在 C8051F206 上使用)            0: ADC0H: ADC0L 寄存器中的数据右对齐            1: ADC0H: ADC0L 寄存器中的数据左对齐</p>								

图 5.6. ADC0H: ADC 数据字寄存器 (C8051F220/1/6 和 C8051F206)



### 5.3. 模数转换器 (ADC) 可编程窗口检测器

在许多应用中, ADC 可编程窗口检测器都是非常有用的。它将 ADC 的输出和用户编程的限制量进行连续比较, 当检测到一个波段溢出时通知给系统。这一点在中断驱动系统中特别有效, 可以以较快的系统响应时间存储代码空间和 CPU 的带宽。窗口检测器中断标志 (ADC0CN 中的 ADWINT) 也能用于查询模式。参考字的高字节和低字节被载入 ADC 大于数据寄存器和 ADC 小于数据寄存器 (ADCOGTH 和 ADCOLTH) 之中。

图 5.7. ADCOGTH: ADC 大于数据寄存器 (C8051F220/1/6 和 C8051F206)

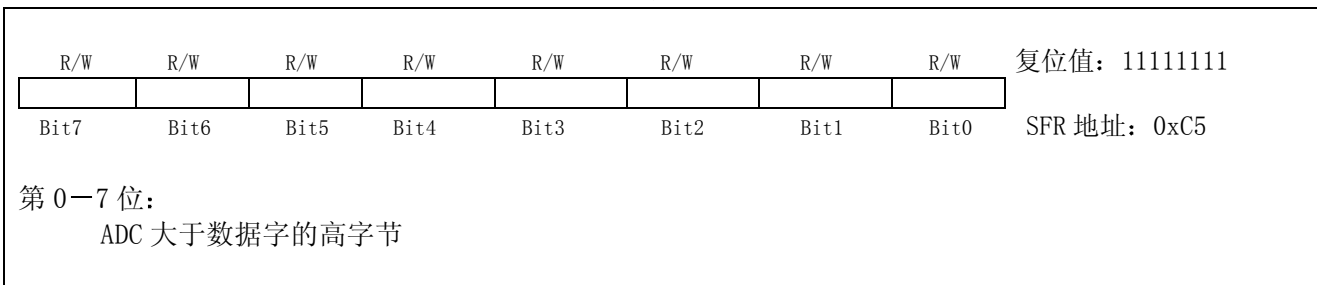


图 5.8. ADCOLTH: ADC 小于数据字节寄存器 (C8051F220/1/6 和 C8051F206)

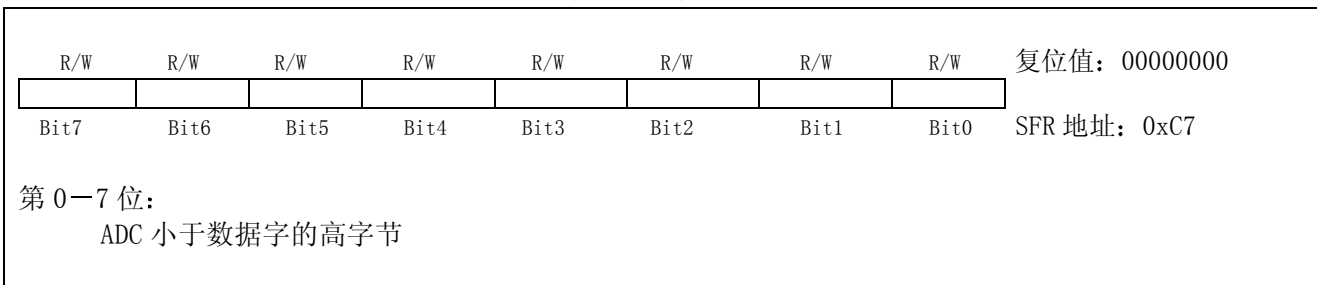


图 5.9. 8 位 ADC 窗口中断例子

Input Voltage (Analog Input-GND)	ADC Data Word	ADC Data Word	ADC Data Word
REFx (255/256)	0xFF	0xFF	} ADWINT=1
	0x21	0x21	
REF x (32/256)	0x20	0x20	ADCOLTH
	0x1F	0x1F	} ADWINT=1
	0x11	0x11	
REFx (16/256)	0x10	0x10	ADCOGTH
	0x0F	0x0F	} ADWINT=1
0	0x00	0x00	

Input Voltage (Analog Input-GND)	ADC Data Word	ADC Data Word	ADC Data Word
REFx (255/256)	0xFF	0xFF	} ADWINT=1
	0x21	0x21	
REFx (32/256)	0x20	0x20	ADCOGTH
	0x1F	0x1F	} ADWINT=1
	0x11	0x11	
REFx (16/256)	0x10	0x10	ADCOLTH
	0x0F	0x0F	} ADWINT=1
0	0x00	0x00	

给定：  
AMXOSL=0x00, AMXOCF=0x00,  
ADLJST=0, ADCOLTH= 0x20,  
ADCOGTH=0x10.

如果得到的 ADC 数据字小于 0x20 且大于 0x10，  
转换的结束将引起一个 ADC 窗口比较中断  
(ADWINT=1)

给定：  
AMXOSL=0x00, AMXOCF=0x00,  
ADLJST=0, ADCOLTH=0x10,  
ADCOGTH=0x20.

如果 ADC 数据字小于 0x10 或大于 0x20，则 ADC  
则 ADC 转换的结束将引起一个 ADC 窗口比  
较中断 (ADWINT=1)



表 5.1. 8 位 ADC 电特性

VDD=3.0V, VREF=2.4V (REFBE=0), PGA 增益=1, 除了特别说明外, 温度是-40°C到+85°C)

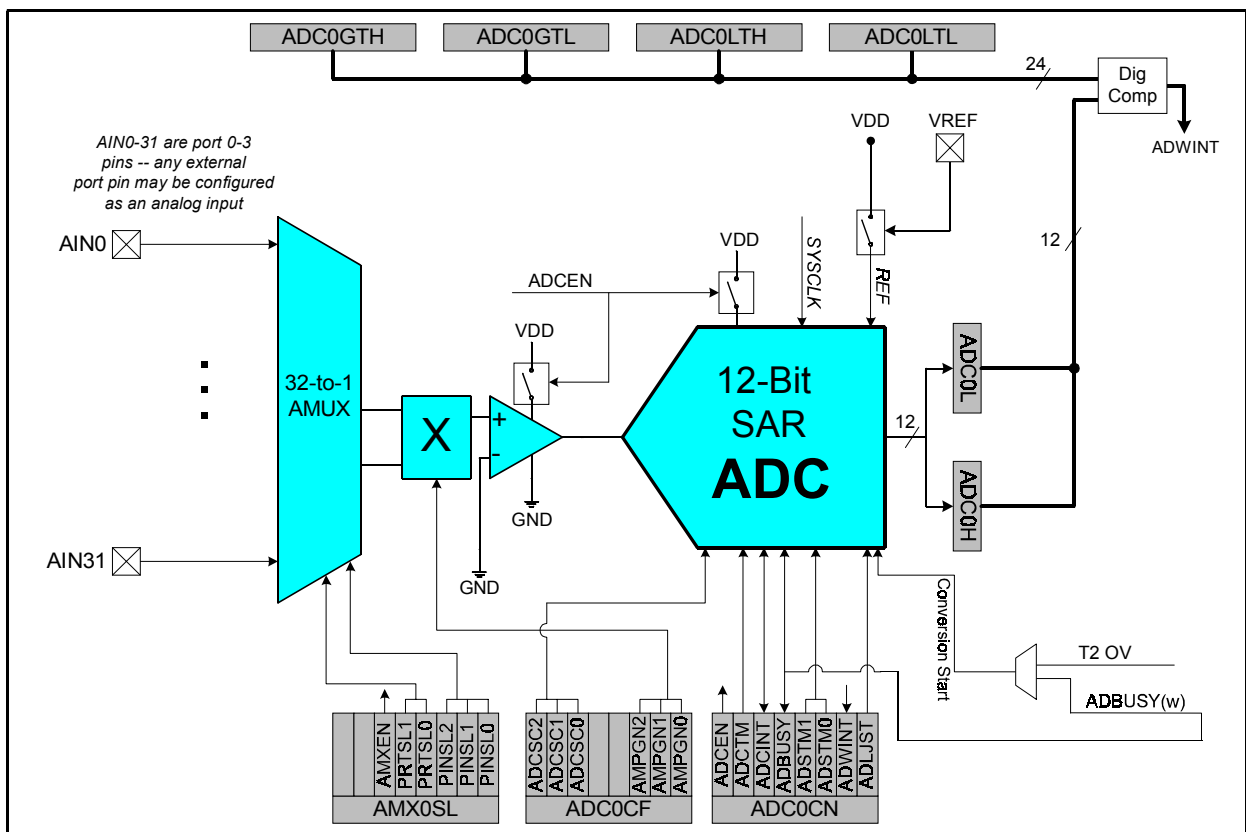
参量	条件	最小值	典型值	最大值	单位
<b>DC 精度</b>					
分辨率			8		位
非线性积分				±1/4	LSB
非线性微分	保证单调性			±1/2	LSB
偏移误差				±1/2	LSB
增益误差				±1/2	LSB
总的非调整误差				±1/2	LSB
<b>动态性能 (10kHz, 正弦波输入, 满刻度的 0-1dB, 100ksps)</b>					
信噪比加失真度			-49.5		dB
总的谐波失真度	到 5 次谐波		-65		dB
假自由动态范围			-65		dB
<b>转换速率</b>					
指令处理能力				100	ksps
<b>模拟量输入</b>					
输入电压		0		VDD	V
输入电容			10		pF
<b>电源规格</b>					
电源电流	操作模式, 100ksps			1.0	mA
关断时的电源电流			0.1	1	μA
电源抑制比			±0.3		mV/V

## 6. 模数转换器（12 位，仅限于 C8051F206）

### 说明：

C8051F206 的子系统 ADC 包含可配置的模拟量多路转换器（AMUX），1 个可编程增益放大器（PGA）和 1 个 100ksps 的 12 位逐次逼近式寄存器 ADC，它具有集成的跟踪保持功能和可编程的窗口检测器（见图 6.1）。AMUX，PGA，数据转换模式和窗口检测器都可通过图 6.1 中的特殊功能寄存器由软件来配置。ADC 的子系统（ADC，跟踪保持和 PGA）仅当 ADC 控制寄存器（ADCOCN，图 6.5）的 ADCEN 位被置 1 时才被使能，当该位是 0 时 ADC 子系统处于低功耗关断状态。

图 6.1. 12 位模数转换器（ADC）功能框图



### 6.1. 模拟量多路转换器（AMUX）和程控增益放大器（PGA）

任一个外部端子（端口 0—3）均可以通过软件来选择。AMX0SL 特殊功能寄存器可以选择期望的模拟量输入端子（见图 6.3）。当 AMUX 被使能时，用户选择所要使用的端口（位 PRTSL0—1），然后被选择的端口（位 PINSL0—2）端子就是模拟量输入。

外观上 AMUX 机能通过通道实现每个可能的配置。ADC 配置寄存器 ADC0CF 中（图 6.4），AMPGN2—0 的状态决定了 PGA 放大 AMUX 的输出信号的增益倍数。PGA 能用软件编程来实现其增益大小是 0.5，1，2，4，8，或 16。复位后它的缺省增益是 1。

### 6.2. 模数转换器(ADC)的操作模式

ADC 的最大转换速度是 100ksps。ADC 的转换时钟来自于系统时钟，大小是系统时钟的除运算结果，除数是 1，2，4，8，或 16，ADC0CF 寄存器中的位 ADCSC 来设置这个除数是几。在调整转换器的速度以适应不同的系统时钟速度时，这一点很有用。

一个转换器的初始化可用两种方式中的一种，这取决于 ADCOCN 中转换模式位 (ADSTM1 和 ADSTM0) 的 ADC 启动的编程状态。转换器的初始化可以是：

- 1 将 1 写到 ADCOCN 中的 ADBUSY 位；
- 2 定时器 2 溢出 (即连续转换计时)。

将 1 写到 ADBUSY 位提供了这样一种可能，就是立刻执行 ADC 如何转换的软件控制。在转换期间，ADBUSY 位置 1，转换结束时，该位恢复到 0。在 ADBUSY 的下降沿触发一个中断 (允许时)，并将 ADCOCN 寄存器的 ADCINT 置中断标志。转换的数据存在于 ADC 的数据字寄存器 ADCOH 中。

寄存器 ADCOCN 中的 ADCTM 位控制 ADC 的跟踪保持模式，缺省状态时，ADC 的输入是连续跟踪的，但转换正在进行时除外。将 ADCTM 置 1 允许两个不同的低功耗跟踪保持模式，这由 ADSTM1-0 位的状态确定 (ADCOCN 也如此)：

- 1 将 1 写至 ADBUSY 时开始跟踪，并持续 3 个 SAR 时钟周期；
- 2 定时器 2 溢出时开始跟踪，并持续 3 个 SAR 时钟周期。

当整个芯片是低功耗备用或睡眠模式时跟踪可以被禁止 (关断)。

**图 6.2. 12 位 ADC 跟踪和转换实例时序**

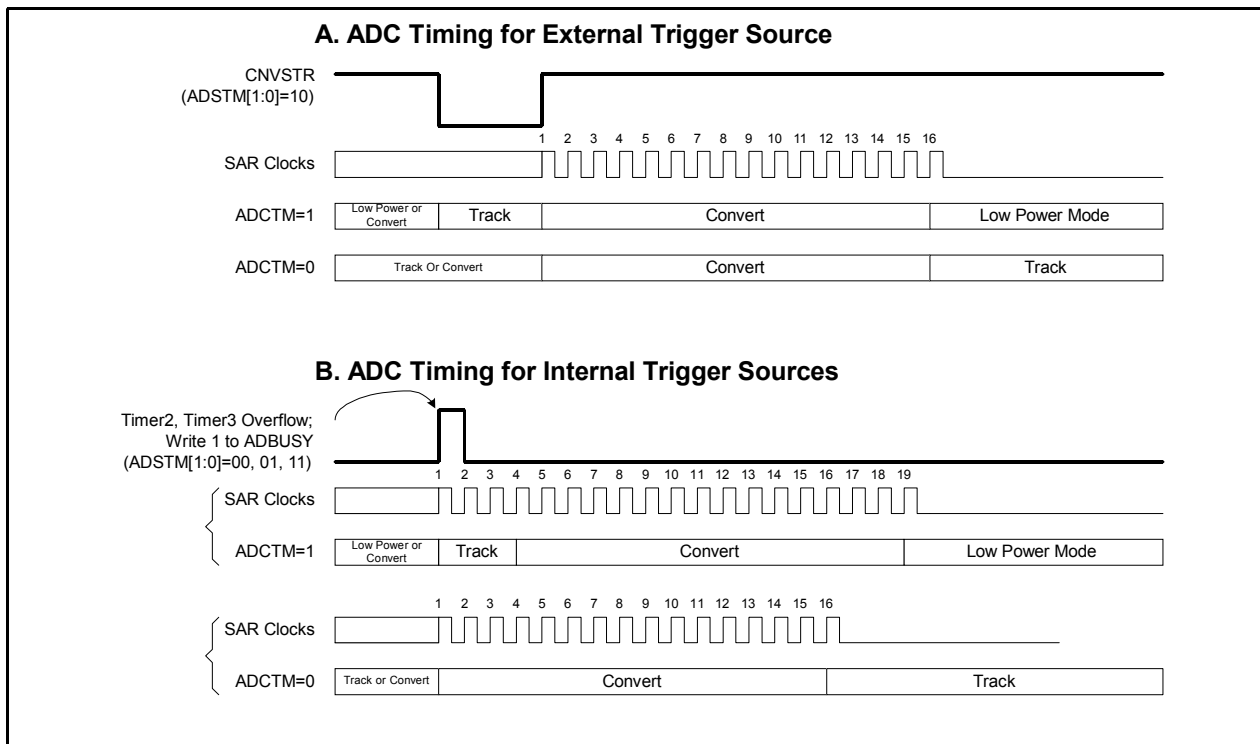


图 6.3. AMXOSL: AMUX 通道选择寄存器 (C8051F220/1/6 和 C8051F206)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
-	-	AMXEN	PRTSL1	PRTSLO	PINSL2	PINSL1	PINSLO	复位值: 00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xBB

第 6, 7 位: 未用。读=00b;写=忽略

第 5 位: AMXEN 使能。

0: AMXEN 被禁止, 使用模拟量时, 端子无效。

1: AMXEN 被使能, 可以使用 / 选择模拟量端子

第 3, 4 位: PRTSL1—0: 端口选择位 \*

00: 选择端口 0 去配置成模拟量输入端子;

01: 选择端口 1 去配置成模拟量输入端子;

10: 选择端口 2 去配置成模拟量输入端子;

11: 选择端口 3 去配置成模拟量输入端子。

第 0, 1, 2 位: PINSL2—0: 端子选择位

000: 上述所选择端口的端子 0 可用于模拟量输入;

001: 上述所选择端口的端子 1 可用于模拟量输入;

010: 上述所选择端口的端子 2 可用于模拟量输入;

011: 上述所选择端口的端子 3 可用于模拟量输入;

100: 上述所选择端口的端子 4 可用于模拟量输入;

101: 上述所选择端口的端子 5 可用于模拟量输入;

110: 上述所选择端口的端子 6 可用于模拟量输入;

111: 上述所选择端口的端子 7 可用于模拟量输入

\* 选择一个端口作为模拟量输入并不是默认这个端口的所有端子均作模拟量输入用。在选择一个端口作为模拟量输入之后, 一个端子必须用端子选择位 (PINSL2—0) 去选择。例如: AMXEN 置 1 后, 置 PRTSL1—0 为 11, 再将 PINSL2—0 置 100, 这样端口 3 的第 4 个端子就被配置成为模拟量输入了。端口 3 的其它端子仍然是 GPIO 端子。应注意, 为了将一个端子作为模拟量输入使用, 它的输入方式应置为模拟量。请看 14.2 节。



图 6.4. ADC0CF: ADC 配置寄存器 (C8051F220/1/6 和 C8051F206)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 01100000
ADCSC2	ADCSC1	ADCSC0	-	-	AMPGN2	AMPGN1	AMPGN0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xBC

第 5, 6, 7 位: ADCSC2—0: ADC 逐次逼近式寄存器 (SAR) 转换时钟周期位  
 000: SAR 转换时钟=1 个系统时钟  
 001: SAR 转换时钟=2 个系统时钟  
 010: SAR 转换时钟=4 个系统时钟  
 011: SAR 转换时钟=8 个系统时钟  
 1xx: SAR 转换时钟=16 个系统时钟  
 注: SAR 转换时钟应该小于或等于 2MHz

第 3, 4 位: 未用。读=00b; 写=忽略

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

图 6.5. ADC0CN: ADC 控制寄存器 (C8051F220/1/6 和 C8051F206)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
ADCEN	ADCTM	ADCINT	ADBUSY	ADSTM1	ADSTM0	ADWINT	ADLJST	SFR 地址: 0xE8
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

(位寻址)

第 7 位: ADCEN: ADC 使能位  
 0: ADC 被禁止。ADC 处于低功耗关断模式  
 1: ADC 被使能。ADC 被激活准备数据转换

第 6 位: ADCTM: ADC 的跟踪模式位  
 0: 当 ADC 被使能时, 跟踪是连续的, 除非转换正在进行之中  
 1: ADSTM1—0 位定义跟踪  
     ADSTM1—0:  
         00: 将 1 写至 ADBUSY 时开始跟踪, 并持续 SAR 的 3 个时钟  
         01: 保留  
         10: 保留  
         11: 定时器 2 溢出时开始跟踪, 并持续 SAR 的 3 个时钟

第 5 位: ADCINT: ADC 转换结束中断标志 (由软件清除)  
 0: 从标志被清除的最后时刻开始, ADC 还未完成一个数据的转换  
 1: ADC 完成了数据的转换

第 4 位: ADBUSY: ADC 忙碌位  
 读  
 0: ADC 转换完成, 或复位一次后, 无效数据被转换。ADBUSY 的下降沿在允许时产生一个中断  
 1: ADC 转换数据忙  
 写  
 0: 没有影响  
 1: 若 ADSTM1—0=00b, 则 ADC 开始转换

第 2, 3 位: ADSTM1—0: ADC 转换模式的启动位  
 00: ADBUSY 置 1, 启动 ADC 转换  
 01: 保留  
 10: 保留  
 11: 定时器 2 溢出时 ADC 转换被初始化

第 1 位: ADWINT: ADC 窗口比较中断标志  
 0: ADC 窗口比较数据不匹配  
 1: ADC 窗口比较数据匹配

第 0 位: ADLJST: ADC 左对齐数据位 (只在 C8051F206 上使用)  
 0: ADC0H: ADC0L 寄存器中的数据右对齐  
 1: ADC0H: ADC0L 寄存器中的数据左对齐

图 6.6. ADC0H: ADC 数据字最高有效位 (MSB) 寄存器 (C8051F206)

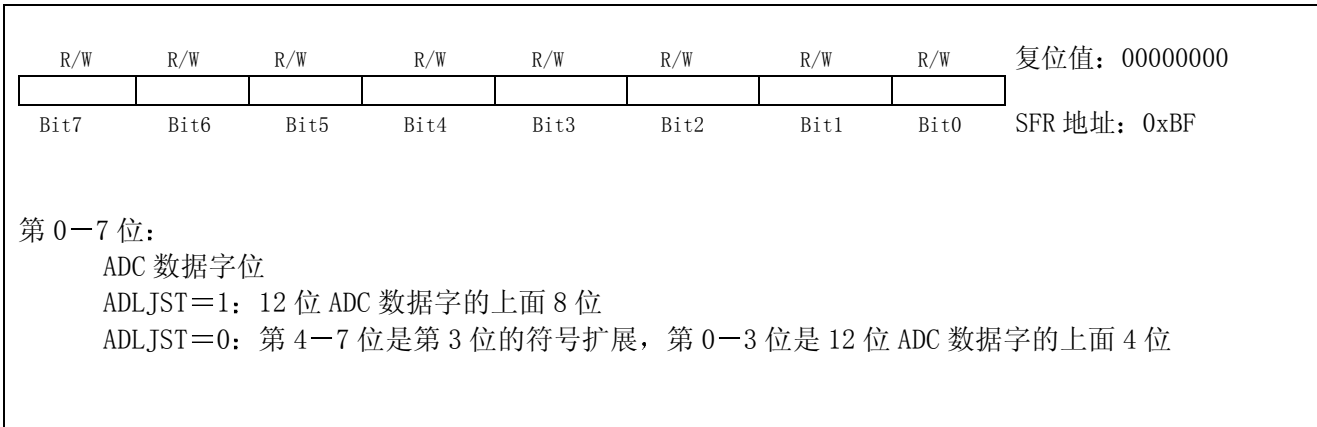
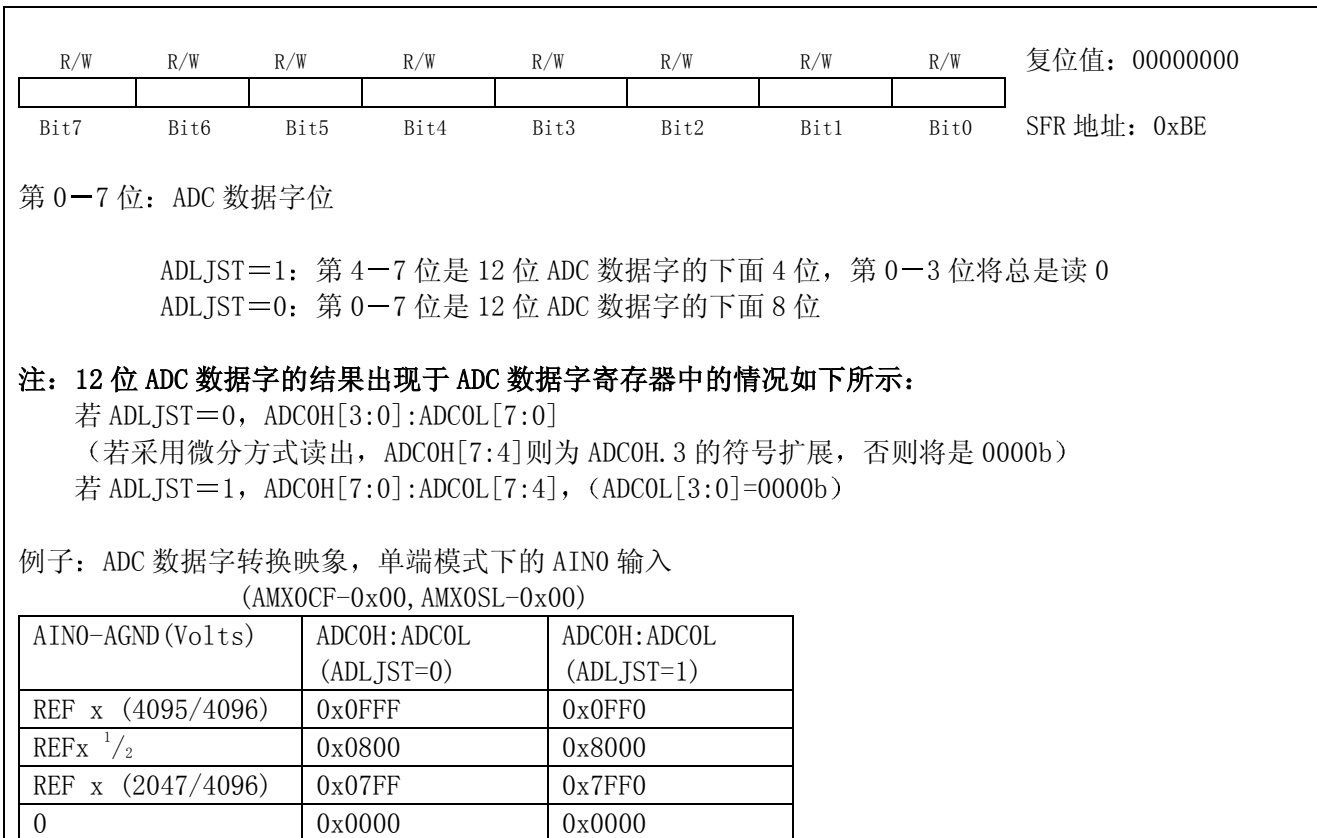


图 6.7. ADC0L: ADC 数据字最低有效位 (LSB) 寄存器 (C8051F206)



### 6.3. 模数转换器 (ADC) 可编程窗口检测器

在许多应用中, ADC 可编程窗口检测器都是非常有用的。它将 ADC 的输出和用户编程的限制量进行连续比较, 当检测到一个波段溢出时通知给系统。这一点在中断驱动系统中特别有效, 可以以较快的系统响应时间存储代码空间和 CPU 的带宽。窗口检测器中断标志 (ADCOCN 中的 ADWINT) 也能用于查询模式。参考字的高字节和低字节被载入 ADC 大于数据寄存器和 ADC 小于数据寄存器 (ADCOGTH, ADCOGTL, ADCOLTL 和 ADCOLTH) 之中。图 6.12 和图 6.13 举了个例子作为比较参考, 注意, 当测量数据在用户程序设置范围内或范围外时, 可根据对 ADCOGTx 和 ADCOLTx 寄存器的编程, 来设置窗口检测器的标志。

图 6.8. ADCOGTH: ADC 大于数据高字节寄存器 (C8051F206)

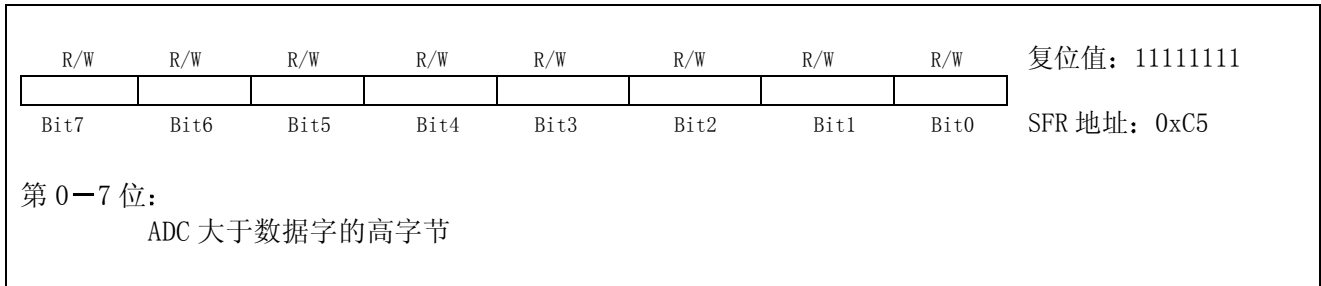


图 6.9. ADCOGTL: ADC 大于数据低字节寄存器 (C8051F206)

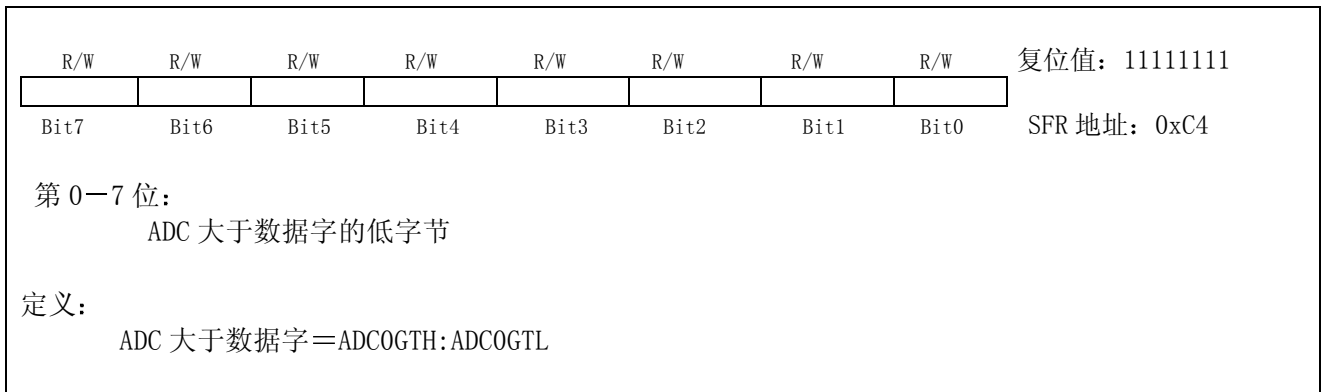


图 6.10. ADCOLTH: ADC 小于数据高字节寄存器 (C8051F206)

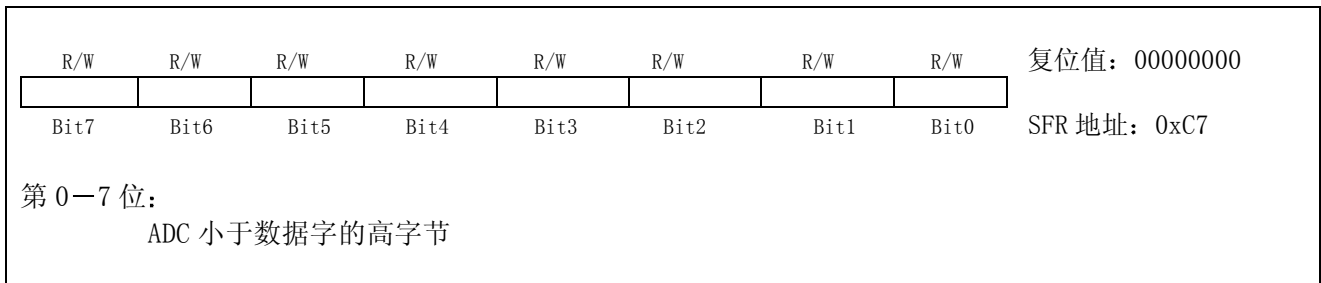


图 6.11. ADCOLT: ADC 小于数据低字节寄存器 (C8051F206)

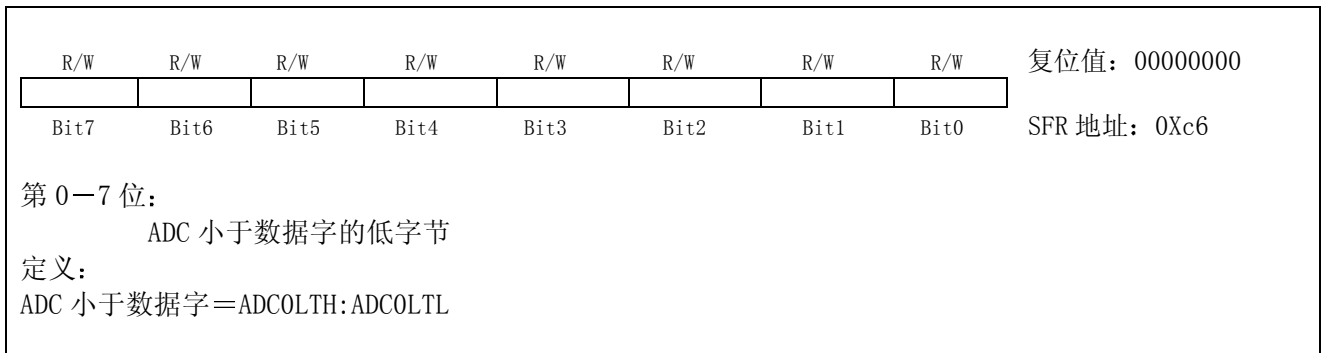


图 6.12. 12 位 ADC 窗口中断实例，右对齐数据

Input Voltage (Analog Input-GND)	ADC Data Word		Input Voltage (Analog Input-GND)	ADC Data Word	
REFx (4095/4096)	0x0FFF	ADWINT Not affected	REFx (4095/4096)	0x0FFF	ADWINT=1
	0x0201			0x0201	
REF x (512/4096)	0x0200	ADCOLTH:ADCOLTL	REF x (512/4096)	0x0200	ADCOGTH:ADCOGTL
	0x01FF	ADWINT=1		0x01FF	ADWINT not affected
	0x0101			0x0101	
REFx (256/4096)	0x0100	ADCOGTH:ADCOGTL	REFx (256/4096)	0x0100	ADCOLTH:ADCOLTL
	0x00FF	ADWINT not affected		0x00FF	ADWINT=1
0	0x0000			0	

<p>给定:</p> <p>AMXOSL=0x00, AMXOCF=0x00, DLJST=0, ADCOLTH:ADCOLTL=0x0200, ADCOGTH:ADCOGTL=0x0100.</p> <p>如果 ADC 数据字小于 0x0200 且大于 0x0100, 则 ADC 转换的结束将引起一个 ADC 窗口比较中断 (ADWINT=1)</p>	<p>给定:</p> <p>AMXOSL=0x00, AMXOCF=0x00, ADLJST=0, ADCOLTH:ADCOLTL=0x0100, ADCOGTH:ADCOGTL=0x0200.</p> <p>如果 ADC 数据字小于 0x0100 大于 0x0200, 则 ADC 转换的结束将引起一个 ADC 窗口比较中断 (ADWINT=1)</p>
--	--

图 6.13. 12 位 ADC 窗口中断实例，左对齐数据

Input Voltage (Analog Input-GND)	ADC Data Word		Input Voltage (Analog Input-GND)	ADC Data Word	
REFx (4095/4096)	0xFFFF0	ADWINT Not affected	REFx (4095/4096)	0xFFFF0	} ADWINT=1
	0x2010			0x2010	
REF x (512/4096)	0x2000	ADCOLTH:ADCOLTL	REF x (512/4096)	0x2000	ADCOGTH:ADCOGTL
	0x1FF0	} ADWINT=1		0x1FF0	} ADWINT not affected
	0x1010				
REFx (256/4096)	0x1000	ADCOGTH:ADCOGTL	REFx (256/4096)	0x1000	ADCOLTH:ADCOLTL
	0x0FF0	} ADWINT not affected		0x0FF0	} ADWINT=1
0	0x0000			0	

<p>给定： AMXOSL=0x00, AMXOCF=0x00, ADLJST=1, ADCOLTH:ADCOLTL=0x2000, ADCOGTH:ADCOGTL=0x1000.</p> <p>如果 ADC 数据字小于 0x0200 且大于 0x0100， 则 ADC 转换的结束将引起一个 ADC 窗口比较 中断 (ADWINT=1)</p>	<p>给定： AMXOSL=0x00, AMXOCF=0x00, ADLJST=1, ADCOLTH:ADCOLTL=0x1000, ADCOGTH:ADCOGTL=0x2000.</p> <p>如果 ADC 数据字小于 0x0100 于 0x0200， 则 ADC 转换的结束将引起一个 ADC 窗口比较 中断 (ADWINT=1)</p>
---	---

表 6.1. 12 位 ADC 电特性 (仅限于 C8051F206)

VDD=3.0V, VREF=2.4V (REFBE=0), PGA 增益=1, 除了特别说明外, 温度是-40°C到+85°C)

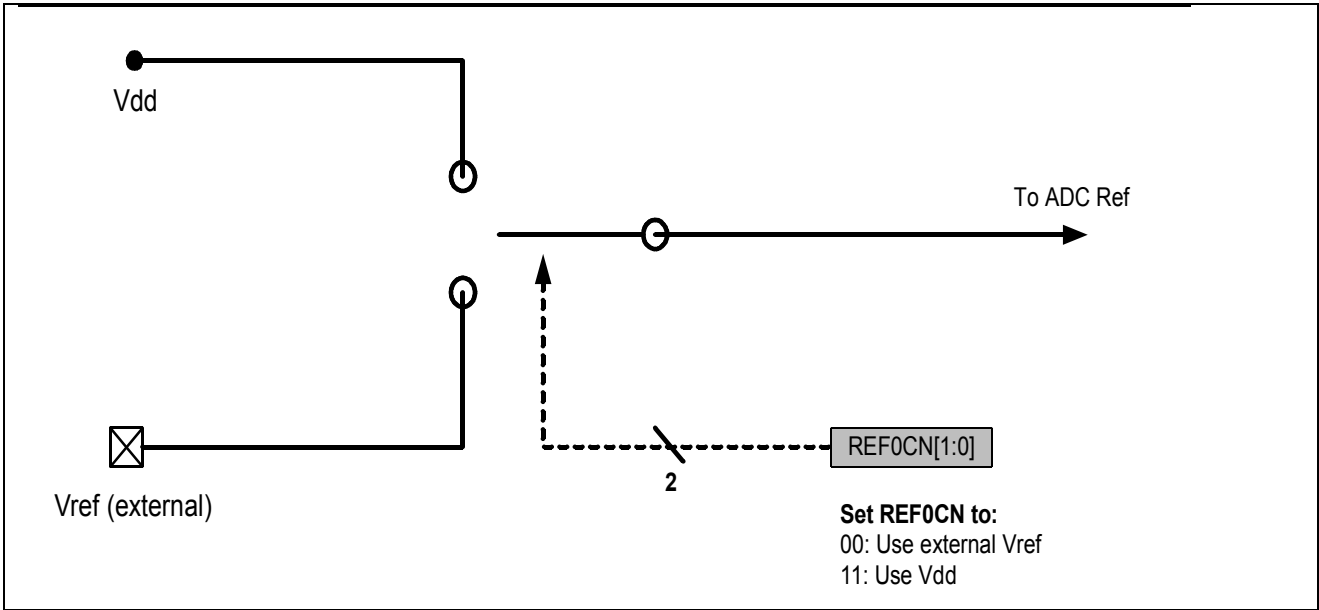
参量	条件	最小值	典型值	最大值	单位
<b>DC 精度</b>					
分辨率		12			位
非线性积分			±1	2	LSB
非线性微分	保证单调性			1	LSB
偏移误差			-3±2		LSB
满刻度误差	微分模式		-20±3		LSB
偏移温度系数			±0.25		ppm/°C
<b>动态性能 (10kHz, 正弦波输入, 满刻度的 0-1dB, 100ksps)</b>					
信噪比加失真度		64			dB
总的谐波失真度	到 5 次谐波		-75		dB
假自由动态范围			80		dB
<b>转换速率</b>					
用系统时钟表示的转换时间	ADCOCF=000xxxxxb	16			时钟
跟踪保持占用时间		1.5			μs
指令处理能力速率				100	ksps
<b>模拟量输入</b>					
电压转换范围		0		VREF	V
输入电压	任一个端子 (模拟量输入模式下)	GND		VDD	V
输入电容			10		pF
<b>电源规格</b>					
电源电流 (VDD 供给 ADC)	操作模式, 100ksps		450	900	μA
电源抑制			±0.3		mV/V

## 7. 基准电压 (C8051F206/220/1/6)

如图 7.1 所示, 基准电压电路在一个由外部连接的基准电压和电源电压 (VDD) 之间选择一个。

一个外部的基准电压可以连接至 VREF 端子, 通过设置特殊功能寄存器 REFOCN 来选择之, 设置方法见图 7.1。外部基准电压必须在 VDD-0.3V 和 1V 之间。VDD 也可以按图 7.2 中使用 REFOCN 的方法来进行选择。基准电压的电气规格参见表 7.1。

**图 7.1. 基准电压功能框图**



<b>Set REFOCN to:</b>	<b>设置 REFOCN:</b>
00: Use external Vref	00: 使用外部 VREF
11: Use Vdd	11: 使用 VDD

**图 7.2. REFOCN: 基准电压控制寄存器**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
-	-	-	-	-	-	REFSL1	REFSLO	SFR 地址: 0xD1
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

第 2—7 位: 未用。读=0000b,; 写=忽略

第 0, 1 位: REFSL1—REFSLO: 基准电压选择  
被选择的基准电压控制位。

- 00: 选择外部 VREF 源
- 01: 保留
- 11: 如同选择 VREF 源一样选择 VDD

**表 7.1. 基准电压电特性**

外部基准 ([REFSL1: REFSLO=00], VREF=2.4V)	最小值	典型值	最大值	单位
输入电压范围	1.00		(VDD)-0.3V	V
输入电流		0.1	10	μA
输入电阻	100			MΩ



## 8. 比较器

MCU有两个片内电压比较器，如图8.1所示。输入经封装端子到每个比较器，由配置好的端口1来任选输出（参看14节）。如果给每个比较器的输出分配封装端子，可通过编程使其工作于开漏极或推挽式的操作模式（参看14.2节）。

每个比较器的滞后可通过各自的比较器控制寄存器（CPT0CN，CPT1CN）进行软件编程。用户能编程实现滞后电压大小（参考输入电压值）以及这个滞后在阈值电压附近的正反相对称度。比较器的输出可用软件来查询，也可用作一个中断源。每个比较器能分别地使能或禁止（关断）。当被禁止的时候，比较器的输出（若通过端口1的MUX分配给了I/O端子）缺省为逻辑低状态，且其中断能力被暂停。比较器的输入可以是外部的驱动信号，从-0.25V到(VDD)+0.25V，没有损耗或翻转。

比较器0的滞后用它的控制寄存器CPT0CN(见图8.3)的第0—3位来编程实现。负的滞后电压值由CP0HYN的位设置来决定。在图8.2中，可以编程设置负滞后的值是10mV，4mV，2mV，也可以将其禁止。同理，正的滞后电压值由CP0HYP的位设置来决定。

在输出跳变时，它的上升沿或者下降沿都能产生比较器的中断。（中断允许和优先级的控制参看9.4节）。在一个比较器0的下降沿中断上，CP0FIF标志被置位；而CP0RIF标志的置位则发生在上升沿中断，置位一次后，一直保持到用户软件来清除。通过读CP0OUT位，可以随时得到比较器0的输出状态。通过设置CP0EN位，比较器0被允许，清除该位则比较器0被禁止。注意在设置CP0EN位和输出稳定之间有20μS的功率。比较器0也能用程序设计成一个复位源。参见11节的详述。比较器1的操作与比较器0相似，只是由CPT1CN寄存器来控制比较器1（图8.4），同时，比较器1不能用作复位源。表8.1给出了比较器完整的电气规格说明。

图 8.1. 比较器功能框图

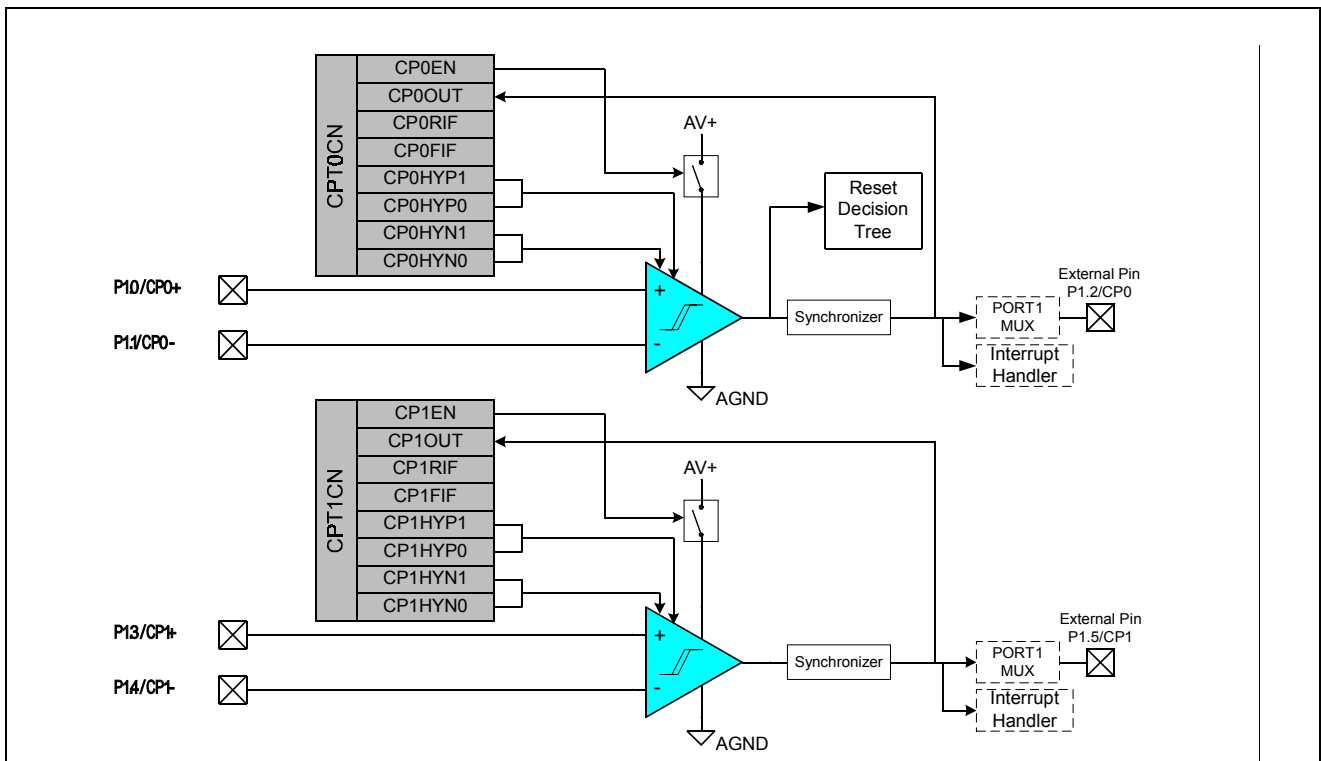


图 8.2. 比较器滞后曲线

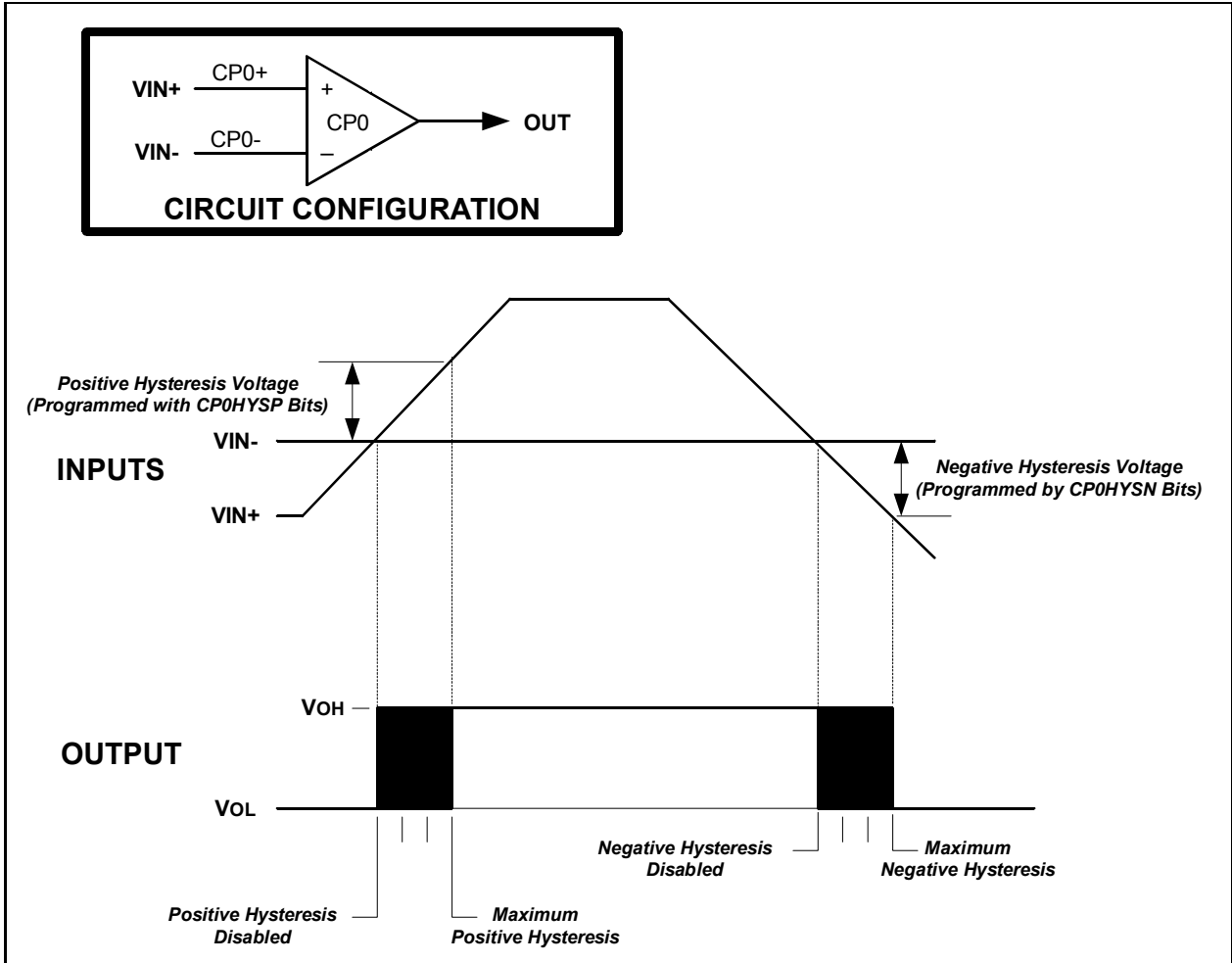


图 8.3. CPT0CN: 比较器 0 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
CPOEN	CPOOUT	CPORIF	CPOFIF	CPOHYP1	CPOHYPO	CPOHYN1	CPOHYNO	SFR 地址: 0x9E
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

第 7 位: CPOEN: 比较器 0 的允许位  
0: 比较器 0 禁止  
1: 比较器 0 允许

第 6 位: CPOOUT: 比较器 0 的输出状态标志  
0: 表示电压大小,  $CP0+ < CP0-$   
1: 表示电压大小,  $CP0+ > CP0-$

第 5 位: CPORIF: 比较器 0 上升沿中断标志  
0: 从这个标志位被清除以来, 比较器 0 上升沿中断未发生过  
1: 从这个标志位被清除后, 比较器 0 上升沿中断发生了

第 4 位: CPOFIF: 比较器 0 下降沿中断标志  
0: 从这个标志位被清除以来, 比较器 0 下降沿中断未发生过  
1: 从这个标志位被清除后, 比较器 0 下降沿中断发生了

第 3-2 位: CPOHYP1-0: 比较器 0 正滞后控制位  
00: 正滞后被禁止  
01: 正滞后 = 2mV  
10: 正滞后 = 4mV  
11: 正滞后 = 10mV

第 1-0 位: CPOHYN1-0: 比较器 0 负滞后控制位  
00: 负滞后被禁止  
01: 负滞后 = 2mV  
10: 负滞后 = 4mV  
11: 负滞后 = 10mV

图 8.4. CPT1CN: 比较器 1 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
CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP1	CP1HYPO	CP1HYN1	CP1HYNO	SFR 地址: 0x9F
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

第 7 位: CP1EN: 比较器 1 的允许位  
0: 比较器 1 禁止  
1: 比较器 1 允许

第 6 位: CP1OUT: 比较器 1 的输出状态标志  
0: 表示电压大小,  $CP1+ < CP1-$   
1: 表示电压大小,  $CP1+ > CP1-$

第 5 位: CP1RIF: 比较器 1 上升沿中断标志  
0: 从这个标志位被清除以来, 比较器 1 上升沿中断未发生过  
1: 从这个标志位被清除后, 比较器 1 上升沿中断发生了

第 4 位: CP1FIF: 比较器 1 下降沿中断标志  
0: 从这个标志位被清除以来, 比较器 1 下降沿中断未发生过  
1: 从这个标志位被清除后, 比较器 1 下降沿中断发生了

第 3-2 位: CP1HYP1-0: 比较器 1 正滞后控制位  
00: 正滞后被禁止  
01: 正滞后 = 2mV  
10: 正滞后 = 4mV  
11: 正滞后 = 10mV

第 1-0 位: CP1HYN1-0: 比较器 1 负滞后控制位  
00: 负滞后被禁止  
01: 负滞后 = 2mV  
10: 负滞后 = 4mV  
11: 负滞后 = 10mV

### 表 8.1. 比较器电特性

VDD=3.0V, 除了特别说明, 温度范围是: -40°C—+85°C

参量	条件	最小值	典型值	最大值	单位
响应时间 1	(CP+) - (CP-) = 100mV 注 1		4		μs
响应时间 2	(CP+) - (CP-) = 10mV 注 1		12		μs
共模抑制比			1.5	4	mV/V
正滞后 1	CPnHYP1-0=00		0	1	mV
正滞后 2	CPnHYP1-0=01	2	4.5	7	mV
正滞后 3	CPnHYP1-0=10	4	9	15	mV
正滞后 4	CPnHYP1-0=11	10	17	25	mV
负滞后 1	CPnHYN1-0=00		0	1	mV
负滞后 2	CPnHYN1-0=01	2	4.5	7	mV
负滞后 3	CPnHYN1-0=10	4	9	15	mV
负滞后 4	CPnHYN1-0=11	10	17	25	mV
反相或正相输入电压范围		-0.25		(VDD)+0.25	V
输入电容			7		pF
输入偏置电流		-5	0.001	+5	nA
输入偏移电压		-10		+10	mV
<b>电源</b>					
上电时间	CPnEN 从 0 到 1		20		μs
电源抑制比			0.1	1	mV/V
供电电流	DC 操作模式 (每个比较器)		1.5	4	μA

注 1: CPnHYP1-0=CPnHYN1-0=00

## 9. CIP-51 微控制器

### 总说明

MCU系统控制器的内核是CIP-51微控制器。CIP-51与MCS-51™的指令集完全兼容，标准的803x/805x的汇编和编译程序能用于开发软件。MCU是一个超集合体，这个集合体的所有外设部件均包含在标准8051中，它有3个16位的计数器 / 定时器（见17节的说明），1个全双工的UART（见16节的说明），256字节的内部RAM，128字节的特殊功能寄存器（SFR）地址空间（见9.3节）和4个字节宽的I / O端口（见14节的说明），CIP-51也包括片内调试硬件（见18节的说明）和接口，这个接口是MCU的模拟、数字子系统的直接接口，在单一的集成电路中，它提供了一个完整的数据采集或控制系统的解决方案

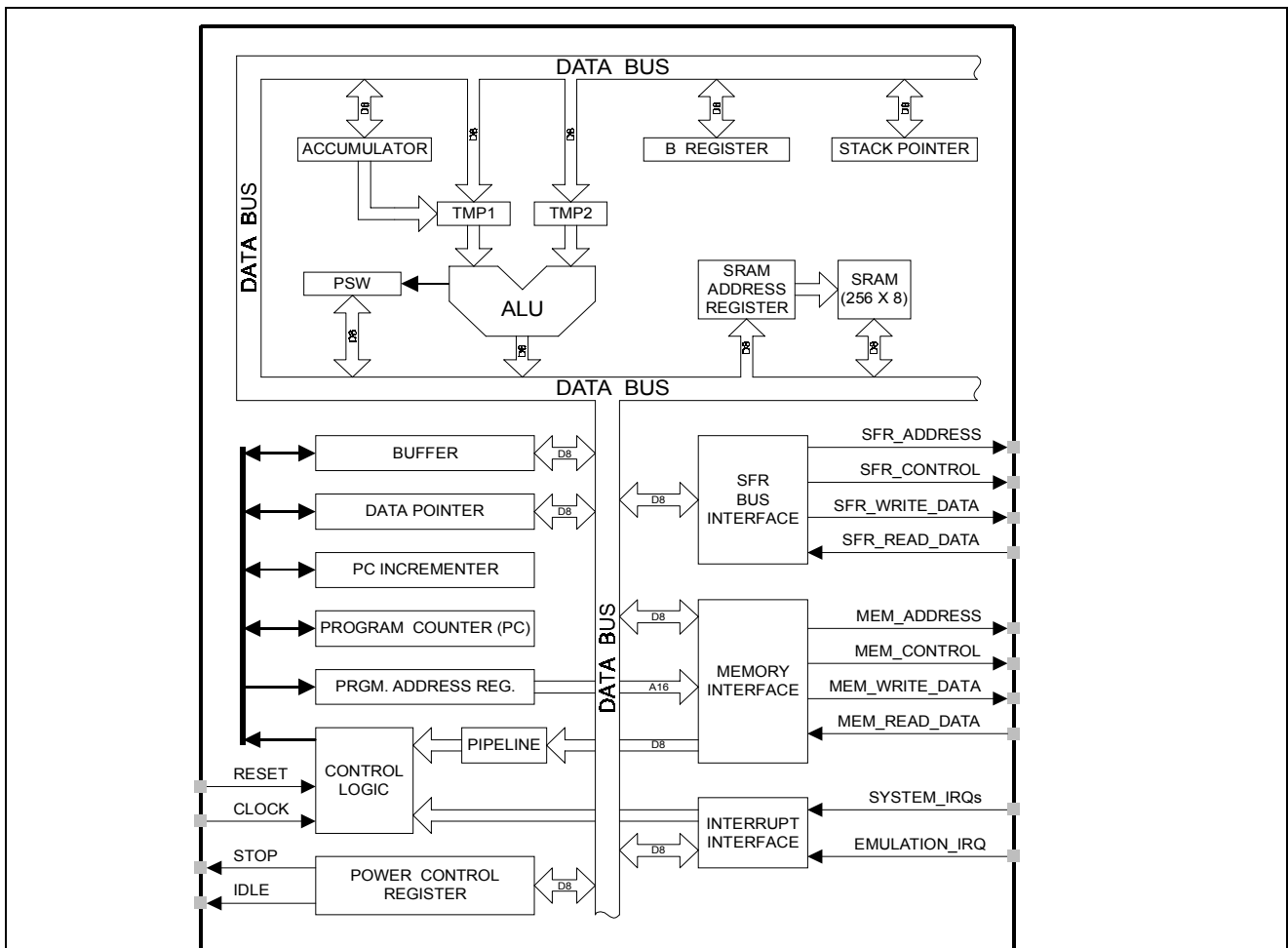
### 特性

CIP-51微控制器内核除了实现标准8051的构成和外设部件以外，还有附加的定制外设部件和功能，以扩展其功能（见图9.1的功能框图）。CIP-51具有下面一些特性：

与MCS-51的指令集完全兼容

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>— 在25MHz的时钟下，有25MIPS峰值的吞吐量</li> <li>— 0—25MHz的时钟频率</li> <li>— 256字节的内部RAM</li> <li>— 可选的1024字节的XRAM</li> <li>— 8k字节FLASH程序存储器</li> </ul> | <ul style="list-style-type: none"> <li>— 4个字节宽的I / O端口</li> <li>— 扩展的中断处理器</li> <li>— 复位输入</li> <li>— 电源管理模式</li> <li>— 片内调试电路</li> <li>— 程序和数据存储器加密</li> </ul> |
|---|---|

图 9.1. CIP-5 框图



## 性能

CIP-51 使用流水线式的体系结构，大大增加了它的指令处理能力，这一点超过了标准的 8051。在一个标准的 8051 中，除了 MUL 和 DIV 两条指令外，在 12MHz 的最大系统时钟下，所有的指令均要占用 12 或 24 个系统时钟周期。对比一下，CIP-51 的内核执行 70% 的指令只用 1 或 2 个系统时钟周期，没有超过 8 个系统时钟周期的指令。

CIP-51 的最大系统时钟是 25MHz，有 25MIPS 的指令处理能力峰值。CIP-51 共有 109 条指令，指令数目对应于执行这些指令所需的系统时钟周期如下表所示：

指令数	26	50	5	14	7	3	1	2	1
所需时钟	1	2	2/3	3	3/4	4	4/5	5	8

## 编程与调试支持

基于 JTAG 调试口的串行接口，可以实现对 FLASH 程序存储器的在线编程，及与芯片上的调试支持电路进行通信的功能。可通过应用软件的 MOVC 和 MOVX 指令对可重复编程的 FLASH 进行一次一个字节的读数和修改操作。这个特性允许程序存储器用于非易失性的数据存储，除此之外，还可以在软件的控制之下更新程序代码。

片内调试支持电路有利于电路中的高速调试，允许设置硬件断点和监视点，以及程序的开始，停止和单步执行（包括中断服务程序），检查程序调用堆栈的情况，读 / 写寄存器和存储器的内容。这种片内调试方法的特点是：完全非插入式和非逃避，不需要 RAM、堆栈、定时器或其它的片内资源。

CIP-51 得到了来自 CYGNAL 集成产品和销售方开发工具的支持。CYGNAL 提供一个集成的开发环境（IDE），包括编辑器，宏汇编程序，调试器和编程器。IDE 中的调试器和编程器与 CIP-51 之间的接口通过 JTAG 接口相连，这样，在系统中的编程和调试变得快速而有效。第三方的宏汇编程序和 C 编译程序也可使用。

## 9.1. 指令集

CIP-51 系统控制器的指令集与标准的 MCS-51™ 指令集完全兼容，标准 8051 的开发工具可用于开发 CIP-51 的软件。CIP-51 的所有指令都是二进制的，功能上和 MCS-51™ 的对应等价，包括操作码，寻址方式，PSW 标志的作用。然而，这种指令时序不同于标准的 8051 指令时序。

### 9.1.1. 指令和 CPU 计时

在多种 8051 的实现中，机器周期和时钟周期是有差别的。在长度上，机器周期于 2—12 个时钟周期中变化。而 CIP-51 的实现是基于时钟周期上的单独时序，所有的指令时序都是根据时钟周期来定的。

由于 CIP-51 指令具有流水线式的结构，大多数指令的执行周期与指令中的程序字节数一样。与用条件分支语句相比，不用条件分支语句时，条件分支指令将少占用一个时钟周期。表 9.1 列出了 CIP-51 的指令集汇总，包括助记符，字节数和每条指令的时钟周期数。

### 9.1.2. MOVX 指令和程序存储器

MOVX 是一个典型的访问外部数据存储器的指令，CIP-51 不支持外部数据或程序存储器。在 CIP-51 中，MOVX 指令通过可重编程的 FLASH 存储器和 1024 字节的 XRAM（在 F226 / 236 和 F206 上可选用）来访问片内程序存储器空间。该特点提供了这样一个机制，就是可为 CIP-51 更新程序代码，并且为了非易失性的数据存储而使用程序存储器空间。参考第 10 节（FLASH 存储器）和第 11 节（外部 RAM）更为详细的描述。

**表 9.1. CIP-51 指令集汇总**

助记符	说明	字节	时钟周期
<b>算术运算类指令</b>			
ADD A, Rn	寄存器加至 A	1	1
ADD A, direct	直接字节加至 A	2	2
ADD A, @Ri	间接 RAM 加至 A	1	2
ADD A, #data	立即数加至 A	2	2
ADDC A, Rn	寄存器加至 A 及进位	1	1
ADDC A, direct	直接字节加至 A 及进位	2	2
ADDC A, @Ri	间接 RAM 加至 A 及进位	1	2
ADDC A, #data	立即数加至 A 及进位	2	2
SUBB A, Rn	寄存器减自 A 及借位	1	1
SUBB A, direct	直接字节减自 A 及借位	2	2
SUBB A, @Ri	间接 RAM 减自 A 及借位	1	2
SUBB A, #data	立即数减自 A 及借位	2	2
INC A	A 增 1	1	1
INC Rn	寄存器增 1	1	1
INC direct	直接字节增 1	2	2
INC @Ri	间接 RAM 增 1	1	2
DEC A	A 减 1	1	1
DEC Rn	寄存器减 1	1	1
DEC direct	直接字节减 1	2	2
DEC @Ri	间接 RAM 减 1	1	2
INC DPTR	数据指针增 1	1	1
MUL AB	A 和 B 相乘	1	4
DIV AB	A 和 B 相除	1	8
DA A	十进制加法累加器调整	1	1
<b>逻辑运算指令</b>			
ANL A, Rn	寄存器与 A	1	1





助记符	说明	字节	时钟周期
ANL A, direct	直接字节与 A	2	2
ANL A, @Ri	间接 RAM 与 A	1	2
ANL A, #data	立即数与 A	2	2
ANL direct, A	A 与直接字节	2	2
ANL direct, #data	立即数与直接字节	3	3
ORL A, Rn	寄存器或 A	1	1
ORL A, direct	直接字节或 A	2	2
ORL A, @Ri	间接 RAM 或 A	1	2
ORL A, #data	立即数或 A	2	2
ORL direct, A	A 或直接字节	2	2
ORL direct, #data	立即数或直接字节	3	3
XRL A, Rn	寄存器异或 A	1	1
XRL A, direct	直接字节异或 A	2	2
XRL A, @Ri	间接 RAM 异或 A	1	2
XRL A, #data	立即数异或 A	2	2
XRL direct, A	A 异或直接字节	2	2
XRL direct, #data	立即数异或直接字节	3	3
CLR A	A 清零	1	1
CPL A	A 取反	1	1
RL A	A 循环左移	1	1
RLC A	A 连进位循环左移	1	1
RR A	A 循环右移	1	1
RRC A	A 连进位循环右移	1	1
SWAP A	A 的高、低 4 位交换	1	1
<b>数据传送指令</b>			
MOV A, Rn	寄存器传送至 A	1	1
MOV A, direct	直接字节传送至 A	2	2
MOV A, @Ri	间接 RAM 传送至 A	1	2
MOV A, #data	立即数传送至 A	2	2
MOV Rn, A	A 传送至寄存器	1	1
MOVRn, direct	直接字节传送至寄存器	2	2
MOVRn, #data	立即数传送至寄存器	2	2
MOV direct, A	A 传送至直接字节	2	2
MOV direct, Rn	寄存器传送至直接字节	2	2
MOV direct, direct	直接字节传送至直接字节	3	3
MOV direct, @Ri	间接 RAM 传送至直接字节	2	2
MOV direct, #data	立即数传送至直接字节	3	3
MOV @Ri, A	A 传送至间接 RAM	1	2
MOV @Ri, direct	直接字节传送至间接 RAM	2	2
MOV @Ri, #data	立即数传送至间接 RAM	2	2
MOV DPTR, #data16	16 位立即数传送至数据指针	3	3
MOVC A, @A+DPTR	相对 DPTR 的代码字节传送至 A	1	3
MOVC A, @A+PC	相对 PC 的代码字节传送至 A	1	3
MOVX A, @Ri	外部数据 RAM (8 位地址) 传送至 A	1	3
MOVX @Ri, A	A 传送至外部数据 RAM (8 位地址)	1	3
MOVX A, @DPTR	外部数据 RAM (16 位地址) 传送至 A	1	3
MOVX @DPTR, A	A 传送至外部数据 RAM (16 位地址)	1	3

助记符	说明	字节	时钟周期
PUSH direct	直接字节压栈	2	2
POP direct	直接字节弹栈	2	2
XCH A, Rn	A 和寄存器交换	1	1
XCH A, direct	A 和直接字节交换	2	2
XCH A, @Ri	A 和间接 RAM 交换	1	2
XCHD A, @Ri	A 和间接 RAM 的低 4 位交换	1	2
<b>布尔操作指令</b>			
CLR C	CY 清零	1	1
CLR bit	直接位清零	2	2
SETB C	CY 置位	1	1
SETB bit	直接位置位	2	2
CPL C	CY 取反	1	1
CPL bit	直接位取反	2	2
ANL C, bit	直接位与 CY	2	2
ANL C, /bit	直接位的反与 CY	2	2
ORL C, bit	直接位或 CY	2	2
ORL C, /bit	直接位的反或 CY	2	2
MOV C, bit	直接位传送到 CY	2	2
MOV bit, C	CY 传送到直接位	2	2
JC rel	CY 置位转移	2	2/3
JNC rel	CY 清零转移	2	2/3
JB bit, rel	直接位置位转移	3	3/4
JNB bit, rel	直接位清零转移	3	3/4
JBC bit, rel	直接位置位转移并清该位	3	3/4
<b>程序分支指令</b>			
ACALL addr11	绝对调用	2	3
LCALL addr16	长调用	3	4
RET	从子程序返回	1	5
RET1	中断返回	1	5
AJMP addr11	绝对转移	2	3
LJMP addr16	长转移	3	4
SJMP rel	短转移 (相对地址)	2	3
JMP @A+DPTR	相对 DPTR 的间址转移	1	3
JZ rel	A 为零转移	2	2/3
JNZ rel	A 非零转移	2	2/3
CJNE A, direct, rel	立即数与 A 比较不等转移	3	3/4
CJNE A, #data, rel	立即数与 A 比较不等转移	3	3/4
CJNE Rn, #data, rel	立即数与寄存器比较不等转移	3	3/4
CJNE @Ri, #data, rel	立即数与间接地址比较不等转移	3	4/5
DJNZ Rn, rel	寄存器减量不为零转移	2	2/3
DJNZ direct, rel	直接字节减量非零转移	3	3/4
NOP	空操作	1	1



## 关于寄存器、操作数和寻址方式的说明

**Rn** - 当前选中寄存器组的寄存器R0-R7

**@Ri** - 通过寄存器R0-R1间接寻址的数据RAM的位置

**rel** - 8位，相对于下一条指令首字节的符号偏移（2的补）。SJMP和所有的条件跳转指令用之

**direct** - 8位内部数据的位置地址。可以是直接访问数据RAM的地址（0x00-0x7F）或SFR（0x80-0xFF）

**#data** - 8位常量

**#data 1** - 16位常量

**bit** - 数据RAM或SFR的直接寻址位

**addr 11** - 被ACALL和AJMP使用的11位目的地址。目的地址必须在程序存储器的下一条指令首字节同一2k字节范围内。

**addr 16** - 被LCALL和LJMP使用的16位目的地址。目的地址可以是8k字节的程序存储器空间内的任一位置

有一个未使用的操作码（0xA5），和NOP有同样的功能。

所有的助记符版权属 Intel 公司（1980）

## 9.2. 存储器结构

CIP-51系统控制器的存储器结构类似于标准8051的存储器结构，有两个分隔的存储器空间，即程序存储器和数据存储器。程序和数据存储器共享同一地址空间，但它们通过不同类型的指令去访问。在CIP-51内部有256字节的内部数据存储器和8k字节的内部程序存储器地址空间。CIP-51的存储器结构图见图9.2。

### 9.2.1. 程序存储器

CIP-51有8k字节的程序存储器空间，MCU将这个程序存储器空间的8320个字节用于可重编程的FLASH存储器和系统中，起始地址是0x0000到0x207F，两个块相邻。注：这个存储器的512字节(0x1E00到0x1FFF)为工厂使用而保留，对用户存储程序来说没有效。

通常假定程序存储器为只读形式的，但CIP-51能通过设置程序存储写允许位(PSCTL.0)和使用MOVX指令将数据写至程序存储器。这一特点为CIP-51更新程序代码、使用程序存储器空间来存储非易失性的数据提供了一个机制。更进一步的叙述参看第10节FLASH存储器。

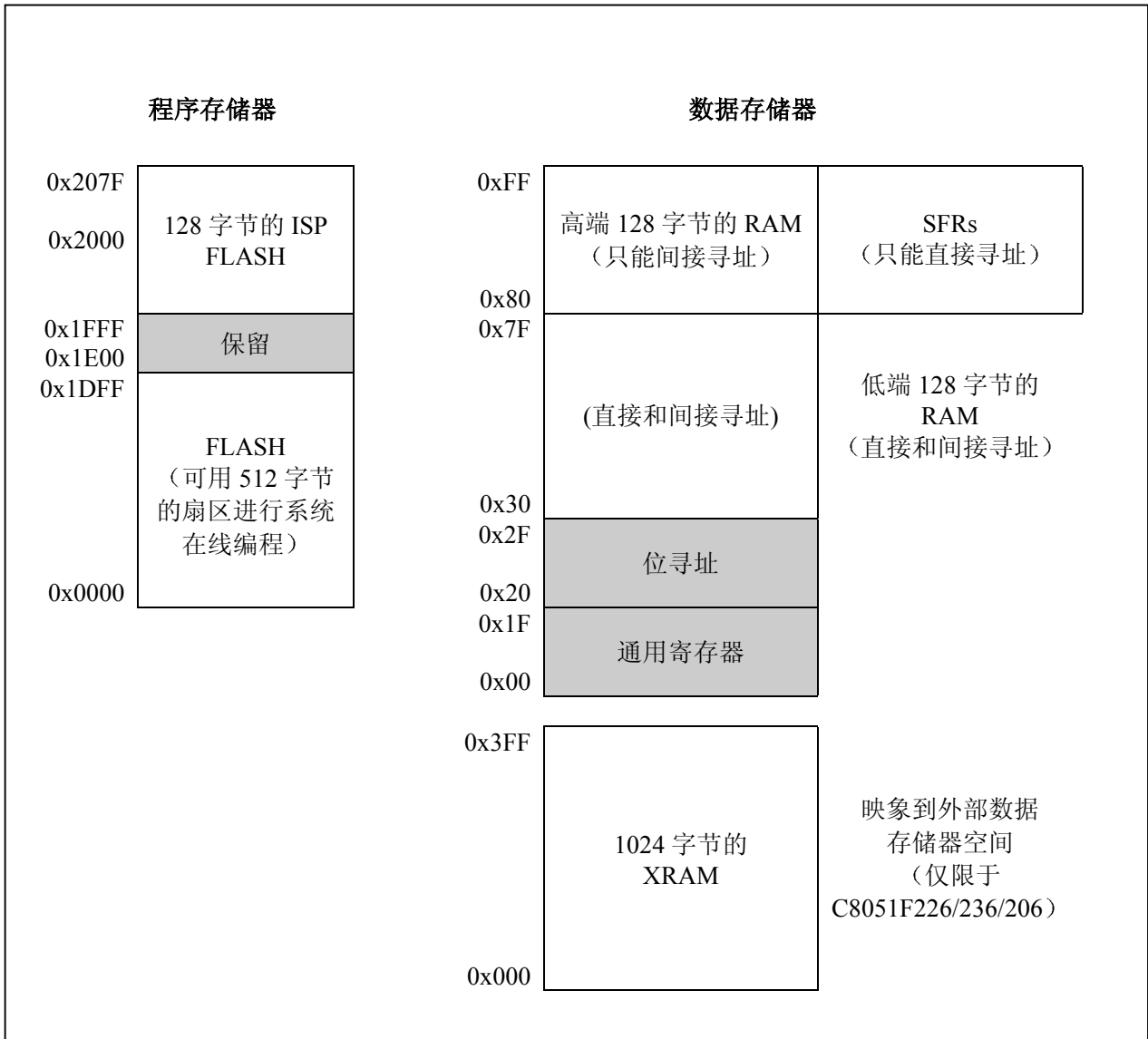
### 9.2.2. 数据存储器

CIP-51将256字节的内部RAM映象到从0x00至0xFF的数据存储器空间。数据存储器低端128字节被用作通用寄存器和缓冲区。直接寻址和间接寻址两者之一均可去访问数据存储器的低128字节。0x00到0x1F的位置可作为4组通用寄存器的寻址，每一组包含8字节宽的寄存器。紧接着的16字节，即0x20到0x2F的位置，可作为字节寻址区，或直接位寻址区。

只有间接寻址允许进入数据存储器的高端128字节，这一区域象SFR一样占用同样的地址空间，但在物理上它和SFR空间是分开的。当访问高于0x7F的地址空间时，指令的寻址方式决定了CPU是访问数据存储器的高128字节，还是SFR。用于直接寻址的指令将访问SFR空间，用于间接寻址0x7F的指令将访问数据存储器的高端128字节。图9.2举例说明了CIP-51数据存储器的结构

补充一点，C8051F206/226/236有1024字节的RAM映象到外部数据存储器，所有的地址区都可以用指令MOVX来访问。(参看第11节)。

图 9.2. 存储器映象



### 9.2.3. 通用寄存器

数据存储器的下面32个字节（从0x00到0x1F），可以为4组通用寄存器所寻址，其中的每一组含8个字节宽的寄存器（R0—R7），在某一时间只有1组可以允许。程序状态字的RS0（PSW. 3）和RS1（PSW. 4）选择寄存器组的激活（参见图9.6中PSW的说明），当进入子程序和中断服务程序时，就可以允许快速的切换，以选择哪一组寄存器。在间接寻址方式中，用寄存器R0和R1作为变址寄存器。

### 9.2.4. 位寻址区

数据存储器的16个字节（0x20—0x2F）除了以字节的形式直接访问以外，还能以128个单独的寻址位进行访问，每1位有一个位地址（0x00—0x7F），当0x20字节的第7位的位地址是0x07时，0x20字节的第0位的位地址是0x00。0x2F字节的第7位的位地址是0x7F。位访问和指令式的完全字节访问存在着区别（位及其位目的操作数不同于字节及其字节目标操作数）。

MCS-51的汇编语言允许用一个替代符号（XX.B）进行位寻址，其中XX是字节地址，B是位在字节中的位置。例如，指令MOV C, 22h.3的功能：

将0x13（0x22字节的第3位）上的布尔值送至用户的进位标志。

### 9.2.5. 堆栈

在256字节数据存储器的任何一个位置都可定位程序员的堆栈。堆栈指针（SP, 0x81）SFR指定堆栈区域，SP指向使用过的上一个位置，下一个值压入堆栈代替SP+1的内容，然后SP+1。复位后初始化堆栈指针为0x07，因此，最初压入堆栈的值被放到0x08（也是寄存器组1中的第一个寄存器R0），这样的话，如果使用了多个寄存器组，SP被初始化后指向数据存储器的某个区，这个区不能用于存储数据。堆栈的深度能扩展至256个字节。

MCU也有为堆栈记录嵌入的硬件，堆栈记录是一个32位的移动寄存器，每一次压入或SP增1，就压入一个记录位至寄存器上，每一次调用压入2个记录位到寄存器；每一次弹出或SP减1，就弹出一个记录位，每一次返回就弹出2个记录位。堆栈记录电路能检测32位移动寄存器的一个溢出或下溢，并通知仿真软件，甚至是MCU在高速调试运行的情况下也如此。

### 9.3. 特殊功能寄存器

直接访问从0x80到0xFF的数据存储器区，构成了特殊功能寄存器（SFRs），SFRs提供控制和CIP-51资源、外设部件的数据交换等功能。与用典型8051实现中相比，CIP-51具有双倍的SFRs，且可以用附加的SFRs去分别配置和访问只有MCU才有的子系统。这样就可在保持与MCS-51的指令系统兼容的同时，增加了新的功能。

直接寻址方式访问从 0x80 到 0xFF 的存储器区，SFR 的寄存器随时都可以被之访问。地址末端是 0x0 或 0x8 (P0, TCON, P1, SCON, IE, 等等)的 SFRs 是位寻址，除此之外，还可字节寻址，其它所有的 SFRs 均是字节寻址。SFR 空间中的空闲地址留为备用，访问这些区域将有不确定的影响，应当避免之。参考表 9.3 所列数据表中对应的页码，有每个寄存器详细的说明。

**表 9.2. 特殊功能寄存器的内存映象**

F8	SPI0CN						WDTCN
F0	B	POMODE	P1MODE	P2MODE	P3MODE <sup>2</sup>		EIP1 EIP2
E8	ADCOCN <sup>1</sup>						RSTSRC
E0	ACC	PRT0MX	PRT1MX	PRT2MX		EIE1 EIE2	
D8							
D0	PSW	REFOCN					
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2	
C0					ADCOGTL <sup>4</sup>	ADCOGTH <sup>1</sup>	ADCOLTL <sup>4</sup> ADCOLTH <sup>1</sup>
B8	IP			AMX0SL <sup>1</sup>	ADCOCF <sup>1</sup>		ADCOL <sup>4</sup> ADCOH <sup>1</sup>
B0	P3	OSCXCN	OSCXCN				FLSCL FLACL
A8	IE					PRT1IF	EMIOCN <sup>3</sup>
A0	P2				PRT0CF	PRT1CF	PRT2CF PRT3CF
98	SCON	SBUF	SPI0CFG	SPI0DAT		SPI0CKR	CPT0CN CPT1CN
90	P1						
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON PSCTL
80	P0	SP	DPL	DPH			PCON
	0 (8)	1 (9)	2 (A)	3 (B)	4 (C)	5 (D)	6 (E) 7 (F)

↑ 位寻址

<sup>1</sup>C8051F230/1/6 没有这些寄存器

<sup>2</sup>C8051F221/231 没有这个寄存器 (32端子封装)

<sup>3</sup>仅限于C8051F206和C8051F226/236.

<sup>4</sup>仅限于 C8051F206 (12 位 ADC)

**表 9.3. 特殊功能寄存器**

SFR 的列表以字母为序

地址	寄存器	说明	页码
0xE0	ACC	累加器	64
0xBC	ADCOCF	ADC 配置	28
0xE8	ADCOCN	ADC 控制	29
0xC5	ADCOGTH <sup>1</sup>	ADC 大于数据字节 (高字节)	30
0xC4	ADCOGTL <sup>4</sup>	ADC 大于数据字节 (低字节)	39



地址	寄存器	说明	页码
0xBF	ADCOH <sup>1</sup>	ADC 数据字 (高字节)	30
0xBE	ADCOL <sup>4</sup>	ADC 数据字 (低字节)	38
0xC7	ADCOLTH <sup>1</sup>	ADC 小于数据字 (高字节)	30
0xCE	ADCOLTL <sup>4</sup>	ADC 小于数据字 (高字节)	39
0xBB	AMXOSL	ADC 的 MUX 通道选择	27
0xF0	B	B 寄存器	64
0x8E	CKCON	时钟控制	121
0x9E	CPT0CN	比较器 0 控制	46
0x9F	CPT1CN	比较器 1 控制	48
0x83	DPH	数据指针 (高位字节)	62
0x82	DPL	数据指针 (低位字节)	62
0xE6	EIE1	扩展中断允许 1	69
0xE7	EIE2	扩展中断允许 2	70
0xF6	EIP1	扩展中断优先级 1	71
0xF7	EIP2	扩展中断优先级 2	72
0xAF	EMIOCN <sup>3</sup>	外部存储器接口控制	80
0xB7	FLACL	FLASH 存储器读限制	79
0xB6	FLSCL	FLASH 存储器计时预定标器	79
0xA8	IE	中断允许	67
0xB8	IP	中断优先级控制	68
0xB2	OSCICN	内部振荡器控制	88
0xB1	OSCXCN	外部振荡器控制	89
0x80	P0	端口 0 锁存器	95
0x90	P1	端口 1 锁存器	96
0xA0	P2	端口 2 锁存器	98
0xB0	P3	端口 3 锁存器	98
0xF1	POMODE	端口 0 数字 / 模拟量输出模式	98
0xF2	P1MODE	端口 1 数字 / 模拟量输出模式	98
0xF3	P2MODE	端口 2 数字 / 模拟量输出模式	98
0xF4	P3MODE <sup>2</sup>	端口 3 数字 / 模拟量输出模式	88
0x87	PCON	电源控制	74
0xA4	PRT0CF	端口 0 配置	95
0xA5	PRT1CF	端口 1 配置	96
0xAD	PRT1IF	端口 1 中断标志	97
0xA6	PRT2CF	端口 2 配置	98
0xA7	PRT3CF	端口 3 配置	99
0xE1	PRTOMX	端口 0 多路转换器 I / O 配置	79
0xE2	PRT1MX	端口 1 多路转换器 I / O 配置	80
0xE3	PRT2MX	端口 2 多路转换器 I / O 配置	80
0x8F	PSCTL	程序存储 RW 控制	78
0xD0	PSW	程序状态字	63





地址	寄存器	说明	页码
0xCB	RCAP2H	计数器 / 定时器 2 捕捉 (高字节)	128
0xCA	RCAP2L	计数器 / 定时器 2 捕捉 (低字节)	128
0xD1	REF0CN	基准电压控制器	44
0xEF	RSTSRC	复位源寄存器	85
0x99	SBUF	串行数据缓冲器 (UART)	113
0x98	SCON	串口控制 (UART)	114
0x81	SP	栈指针	62
0x9A	SPIOCFG	串行外设接口配置	105
0x9D	SPIOCKR	SPI 时钟速率	107
0xF8	SPIOCN	SPI 总线控制	106
0x9B	SPIODAT	SPI 端口 1 数据	107
0xC8	T2CON	计数器 / 定时器 2 控制	127
0x88	TCON	计数器 / 定时器控制	119
0x8C	TH0	计数器 / 定时器 0 数据字 (高字节)	122
0x8D	TH1	计数器 / 定时器 1 数据字 (高字节)	122
0xCD	TH2	计数器 / 定时器 2 数据字 (高字节)	128
0x8A	TL0	计数器 / 定时器 0 数据字 (低字节)	122
0x8B	TL1	计数器 / 定时器 1 数据字 (低字节)	122
0xCC	TL2	计数器 / 定时器 2 数据字 (低字节)	128
0x89	TMOD	计数器 / 定时器模式	120
0xFF	WDTCN	看门狗定时器控制	84
0x84-86, 0x91-97, 0x9C, 0xA1-A3, 0xA9-AC, 0xAE, 0xB3-B5, 0xB9-BA, 0xBD-BE, 0xC0-C4, 0xC6, 0xCE-CF, 0xD2-DF, 0xE9-EE, 0xF5, 0xF9-FE		保留	

<sup>1</sup>C8051F230/1/6 没有这些寄存器<sup>2</sup>C8051F221/231 没有这个寄存器(32端子封装)<sup>3</sup>仅限于C8051F206和C8051F226/236.<sup>4</sup>仅限于 C8051F206(12 位 ADC)

### 9.3.1. 寄存器说明

以下是SFRs的说明，涉及到CIP-51系统控制器的操作。保留位应设置成逻辑0，如果这些位的复位值是逻辑0，就使得以后的产品可以使用之，以补充新的特性、选择以后的缺省状态。余下的SFRs更详细的说明在数据表一节中，有它们所对应的系统功能。

图 9.3. SP: 堆栈指针

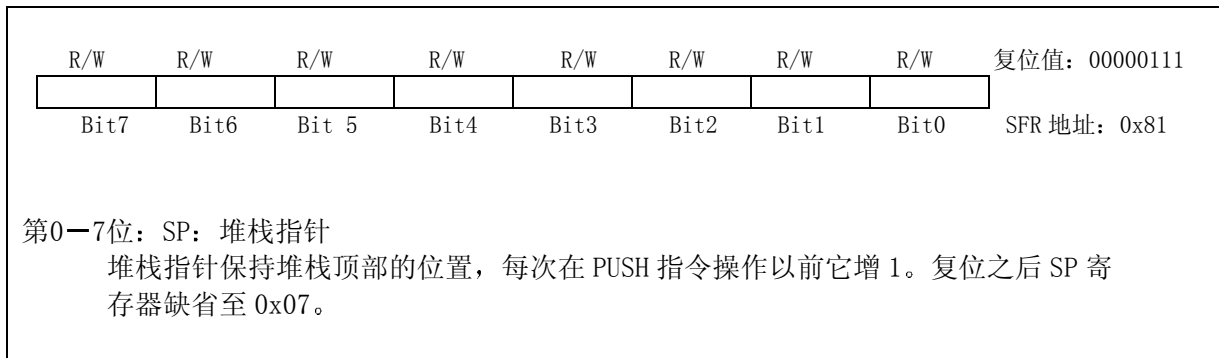


图 9.4. DPL: 数据指针低位字节

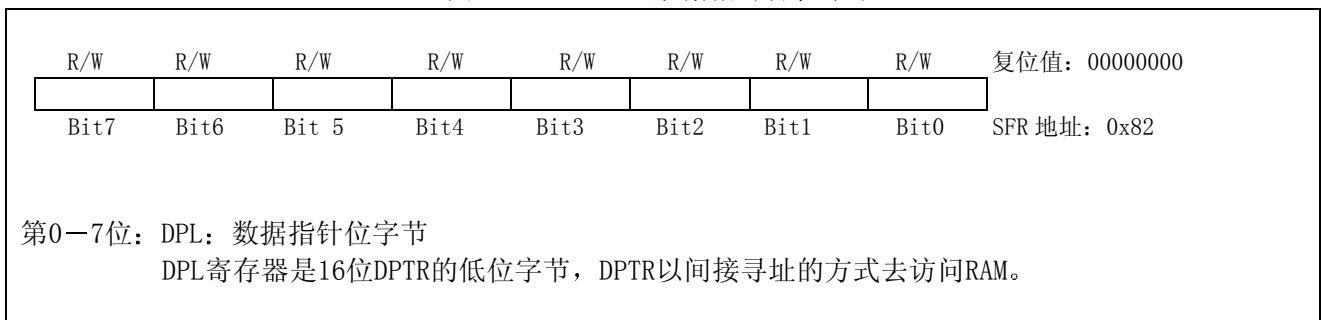


图 9.5. DPH: 数据指针高位字节

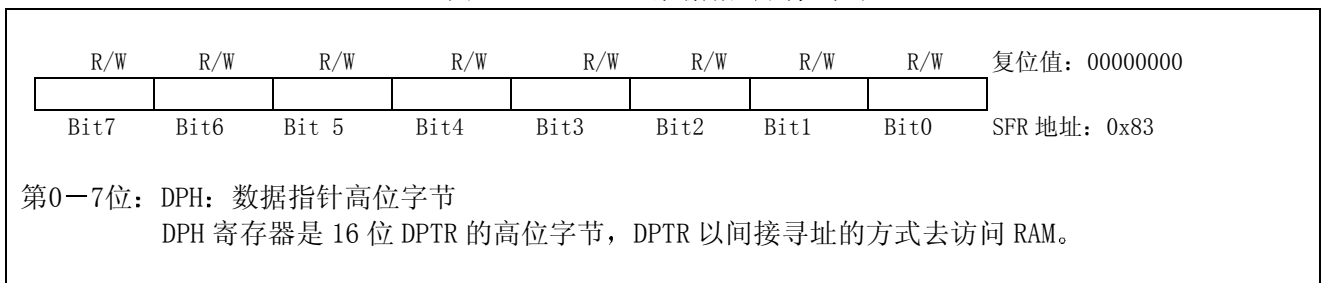


图 9. 6. PSW: 程序状态字

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

**第7位: CY: 进位标志位**  
 当上一次算术运算结果产生进位（加法）或借位（减法）时，该位被置位，其它所有的算术运算均可将之清零

**第6位: AC: 辅助进位标志位**  
 当上一次算术运算结果有向高阶半字节进位（加法）或从高阶半字节借位（减法）时，该位被置位，所有的其它算术运算均可将之清零

**第5位: F0: 用户标志0**  
 在软件的控制下，该位是一个通用的标志，用于位寻址

**第3, 4位: RS1-RS0: 寄存器组选择位**  
 在寄存器访问期间，这两个位选择寄存器组

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

**第2位: OV: 溢出标志位**  
 若上一次算术运算结果产生进位（加法），借位（减法）或溢出（乘或除）时，该位被置位，所有的其它算术运算均可将之清零。

**第1位: F1: 用户标志1**  
 在软件的控制下，该位是一个通用的标志，用于位寻址

**第0位: PARITY: 奇偶标志位**

**图 9.7. ACC: 累加器**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
ACC. 7	ACC. 6	ACC. 5	ACC. 4	ACC. 3	ACC. 2	ACC. 1	ACC. 0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xE0 (位寻址)

第 0—7 位: ACC: 累加器  
这个寄存器是用于算术运算的累加器。

**图 9.8. B: B 寄存器**

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

第 0—7 位: B: B 寄存器  
这个寄存器适合于作某些算术运算的第二个累加器。

## 9.4. 中断处理器

CIP-51包括一个具有两个优先级的扩展中断系统，它支持22个中断源。根据设备的具体情况，中断源可以是片内外设和外部输入端子的变化，每一个中断源都有一个或多个相应的SFR中断请求标志。当一个外设或外部源遇到一个有效的中断条件时，相应的中断请求标志被置位。

如果中断源被允许，当中断请求标志是1时，中断请求发生。此时，将很快完成现行指令的执行，CPU产生一个LCALL命令到预定的地址，开始执行中断服务程序（ISR）。每个ISR必须以RETI指令来结束，它返回程序去执行中断请求还未发生时已执行完的指令的下一条。如果中断未被允许，中断请求标志就被硬件和正常连续执行的程序所忽略。（不论中断是被允许，还是被禁止，中断请求标志均被设置成逻辑1。）

每个中断源能分别允许或禁止，这是通过使用SFR（IE—EIE2）中相关联的中断允许位来实现的。不论怎样，中断必须是首先识别出单个的中断允许之前，将EA位（IE.7）置逻辑1，中断才被全部允许。将EA的位设置成逻辑0后，不论单个中断允许的设置情况如何，所有的中断源将被禁止。

当CPU引导至ISR时，一些中断请求标志通过硬件自动地清除，但是，大多数不能通过硬件来清除，从ISR返回之前，必须由软件来清除。如果一个中断请求标志在CPU完成自中断返回（RETI）指令之后仍然处于置位状态，一个新的中断请求将立即产生，CPU执行完下一条指令之后，将再次进入ISR。

### 9.4.1. MCU中断源和中断向量

MCU分配了9个中断源给片内外设部件，将任意一个中断请求标志设置为逻辑1，软件就能模拟一个中断。如果中断标志被允许，就产生了一个中断请求，CPU将中断向量引导至与中断请求标志有关的ISR地址。MCU中断源、关联的向量地址、优先级顺序和控制位见表9.4的概述。有关片内外设部件的中断条件及其中断标志的行为，可通过查阅相应外设的数据资料得到。

### 9.4.2. 外部中断

将两个外部中断源（/INT0和/INT1）配置为电平触发或边沿触发，取决于IT0（TCON.0）和IT1（TCON.2）两个位的设置。IE0（TCON.1）和IE1（TCON.3）分别充当/INT0和/INT1的中断请求标志，如果/INT0或/INT1外部中断配置为边沿触发，当CPU向量引导至ISR时，对应的中断请求标志将由硬件自动地清除；若配置为电平触发，则中断请求标志将跟随外部输入端子的状态。外部中断源必须一直保持输入激活状态，直到识别出中断请求为止。在ISR的执行完成之前或产生另一个中断请求之前，必须释放中断请求。

端口1中断标志寄存器中的中断请求标志见图13.10。

表 9.4. 中断汇总

中断源	中断向量	优先权顺序	中断请求标志	允许
复位	0x0000	最高	没有	总被允许
外部中断 0 (/INT0)	0x0003	0	IE0 (TCON. 1)	EX0 (IE. 0)
定时器 0 溢出	0x000B	1	TF0 (TCON. 5)	ET0 (IE. 1)
外部中断 1 (/INT1)	0x0013	2	IE1 (TCON. 3)	EX1 (IE. 2)
定时器 1 溢出	0x001B	3	TF1 (TCON. 7)	ET1 (IE. 3)
串口 (UART)	0x0023	4	R1 (SCON. 0) T1 (SCON. 1)	ES (IE. 4)
定时器 2 溢出 (或 EXF2)	0x002B	5	TF2 (T2CON. 7)	ET2 (IE. 5)
串行外设部件接口	0x0033	6	SPIF (SPIOSTA. 7)	ESPIO (EIE1. 0)
ADC0 窗口比较	0x0043	8	ADWINT (ADCOCN. 2)	EWADCO (EIE1. 2)
比较器 0 下降沿	0x0053	10	CPOFIF (CPT0CN. 4)	ECPOF (EIE1. 4)
比较器 0 上升沿	0x005B	11	CPORIF (CPT0CN. 3)	ECPOR (EIE1. 5)
比较器 1 下降沿	0x0063	12	CP1FIF (CPT1CN. 4)	ECP1F (EIE1. 6)
比较器 1 上升沿	0x006B	13	CP1RIF (CPT1CN. 3)	ECP1R (EIE1. 7)
ADC0 转换结束	0x007B	15	ADCINT (ADCOCN. 5)	EADCO (EIE2. 1)
外部中断 4	0x0083	16	IE4 (PRT1IF. 4)	EX4 (EIE2. 2)
外部中断 5	0x008B	17	IE5 (PRT1IF. 5)	EX5 (EIE2. 3)
外部中断 6	0x0093	18	IE6 (PRT1IF. 6)	E X6 (EIE. 4)
外部中断 7	0x009B	19	IE7 (PRT1IF. 7)	E X7 (EIE2. 5)
未用的中断区	0x00A3	20	没有	保留 (EIE2. 6)
外部晶振就绪	0x00AB	21	XTLVLD (OSXCEN. 7)	EXVLD (EIE2. 7)

### 9.4.3. 中断优先级

每一个中断源能分别进行编程，以决定其中断优先级是低还是高，低优先级的ISR被高优先级的中断所阻止，一个高优先级的中断不能被阻止。每一个中断在SFR (IP-EIP2) 之中都有一个相应的中断优先级位，可用来配置优先级水平。低优先级是缺省状态，若两个中断被同时识别出来，那么首先服务更高一级的中断，如果两个中断有同样的优先级，则靠一个固定的优先级顺序去裁决。

### 9.4.4. 中断等待

当中断发生时，中断响应时间取决于 CPU 的状态，每一时钟周期采样一次未处理的中断信息并进行中断优先级解码，因此，最快的可能响应时间是 5 个系统时钟周期，其中 1 个时钟周期检测中断，4 个时钟周期完成 LCALL 指令到 ISR。当 RETI 执行后，如果一个中断正在请求，在 LCALL 服务于它之前，将执行完一条指令。因此，对一个中断来说（当前没有其它的中断需要服务或没有更大优先级的新中断），当 CPU 执行 RETI 指令后，下一条是 DIV 指令时，此时产生一个中断所需的最大响应时间是 18 个系统时钟周期：1 个时钟周期检测中断，5 个时钟周期执行 RETI，8 个时钟周期完成 DIV 指令，4 个时钟周期执行 LCALL 到 ISR。

**注：**若一个 FLASH 写或擦除完成后，MCU 被操作所耽误，直到操作完成为止，将不会响应中断。如果 CPU 正在执行一个中断的 ISR，那么对具有相等或低优先级的新中断来说，它也不会响应，直到当前的 ISR，包括 RETI 和下一条指令完成为止。

### 9.4.5 中断寄存器的描述

用于中断源使能及设置其优先级的 SFR，这些以下均有描述。有关外设的有效中断条件和中断悬挂标志行为，请参照数据表部分中与特定芯片外设相关的信息。

**图 9.9 IE: 中断允许**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值:00000000
<b>EA</b>	<b>-</b>	<b>ET2</b>	<b>ES</b>	<b>ET1</b>	<b>ES1</b>	<b>ET0</b>	<b>ES0</b>	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xA8

位 7: EA: 允许一切中断  
 这个位允许/禁止所有的中断。它的优先级高于各个单独的中断屏蔽设置。  
 0: 禁止一切中断源  
 1: 根据各自的屏蔽设置，允许各个中断

位 6: 不使用。读=0，写=忽略

位 5: ET2: 允许定时器 2 中断  
 这个位设置了定时器 2 中断的屏蔽  
 0: 禁止一切定时器 2 中断  
 1: 允许由 TF2 标志 (T2CON. 7) 产生的中断请求

位 4: ES: 允许串行口 (UART) 中断  
 这个位设置了串行口 (UART) 中断的屏蔽  
 0: 禁止一切 UART 中断  
 1: 允许由 R1 标志 (SCON. 0) 或 T1 标志 (SCON. 1) 产生的中断请求

位 3: ET1: 允许定时器 1 中断  
 这个位设置了定时器 1 中断的屏蔽  
 0: 禁止一切定时器 1 中断  
 1: 允许由 TF1 标志 (TCON. 7) 产生的中断请求

位 2: EX1: 允许外部中断 1  
 这个位设置了外部中断 1 的屏蔽  
 0: 禁止外部中断 1  
 1: 允许有 /INT1 引脚产生的中断请求

位 1: ET0: 允许定时器 0 中断  
 这个位设置了定时器中断的屏蔽  
 0: 禁止一切定时器 0 中断  
 1: 允许由 TF0 标志 (TCON. 5) 产生的中断请求

图 9.10 中断优先级

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
-	-	PT2	PS	PT1	PX1	PT0	PX0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xB8

位 7-6: 不使用 读=00b, 写=忽略

位 5: PT2 定时器 2 中断优先级控制  
这个位设置了定时器 2 中断的优先级  
0: 定时器 2 中断优先级由默认的优先级决定  
1: 定时器 2 中断产生高优先级

位 4: PS: 串行口 (UART) 中断优先级控制  
这个位设置了串行口 (UART) 中断的优先级  
0: UART 中断优先级由默认的优先级决定  
1: UART 中断产生高优先级

位 3: PT1: 定时器 1 中断优先级  
这个位设置了定时器 1 中断的优先级  
0: 定时器 1 优先级控制由默认的优先级决定  
1: 定时器 1 中断产生高优先级

位 2: PX1: 外部中断 1 优先级控制  
这个位设置了外部中断 1 的中断优先级  
0: 外部中断 1 优先级由默认的优先级决定  
1: 外部中断 1 产生高优先级

位 1: PT0: 定时器 0 中断优先级控制  
这个位设置了定时器 0 中断的优先级  
0: 定时器 0 中断优先级由默认的优先级决定  
1: 定时器 0 中断产生高优先级

位 0: PX0: 外部中断 0 优先级控制  
这个位设置了外部中断 0 中断的优先级  
0: 外部中断 0 优先级由默认的优先级决定  
1: 外部中断 0 产生高优先级



图 9.11 EIE1: 扩展中断允许 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
<b>ECP1R</b>	<b>ECP1F</b>	<b>ECP0R</b>	<b>ECP0F</b>	-	<b>EWADC0</b>	-	<b>ESP10</b>	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	FR 地址: 0xE6

位 7: ECP1R: 允许比较器 1 (CP1) 上跳沿中断  
 这个位设置了 CP1 中断的屏蔽  
 0: 禁止 CP1 上跳沿中断  
 1: 允许由 CP1RIF 标志 (CPT1CN. 3) 产生的中断请求

位 6: ECP1F: 允许比较器 1 (CP1) 下跳沿的中断  
 这个位设置了 CP1 中断的屏蔽  
 0: 禁止 CP1 下跳沿中断  
 1: 允许由 CP1FIF 标志产生的中断请求

位 5: ECP0R: 允许比较器 0 (CP0) 上跳沿中断  
 这个位设置了 CP0 中断的屏蔽  
 0: 禁止 CP0 上跳沿中断  
 允许由 CP0RIF 标志 (CPT0CN) 产生的中断请求

位 4: ECP0F: 允许比较器 0 下跳沿中断  
 这个位设置了 CP0 中断的屏蔽  
 0: 禁止 CP0 下跳沿中断  
 1: 允许由 CPOFIF 标志 (CPT0CN) 产生的中断请求

位 3: 保留 读=0 写=忽略

位 2: EWADC0: 允许窗口比较 ACDO 中断  
 这个位设置了 ACDO 窗口比较中断的屏蔽  
 0: 禁止 ADC0 窗口比较中断  
 1: 允许由 ADC0 窗口比较产生的中断请求

位 1: 保留 读=0 写=忽略  
 这个位设置了 SPI0 中断的屏蔽  
 0: 禁止一切 SPI0 中断  
 1: 允许由 SPI0 标志 (SPI0CN. 7) 产生的中断请求

图 9.12 EIE2: 扩展中断允许 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值:00000000
EXVLD	-	EX7	EX6	EX5	EX4	EADC0	-	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址 0xE7

位 7: EXVLD: 允许外部时钟源有效中断  
这个位设置了 XTLVLD 中断的屏蔽  
0: 禁止一切 XTLVLD 中断  
1: 允许由 XTLCLD 标志 (OSCXCN.7) 产生的中断请求

位 6: 保留 必写 0 读 0

位 5: EX7: 允许外部中断 7  
这个位设置了外部中断 7 的屏蔽  
0: 禁止外部中断 7  
1: 允许由外部中断输入引脚产生的中断请求

位 4: EX6: 允许外部中断 6  
这个位设置了外部中断 6 的屏蔽  
0: 禁止外部中断 6  
1: 允许由外部中断 6 输入引脚产生的中断请求

位 3: EX5: 允许外部中断 5  
这个位设置了外部中断 5 的屏蔽  
0: 禁止外部中断 5  
1: 允许由外部中断输入引脚产生的中断请求

位 2: EX4: 允许外部中断 4  
这个位设置了外部中断 4 的屏蔽  
0: 禁止外部中断 4  
1: 允许由外部中断 4 输入引脚产生的中断请求

位 1: EADC0: 允许 ADC0 转化结束中断  
这个位设置 ADC0 转化结束中断的屏蔽  
0: 禁止 ADC0 转化结束中断  
1: 允许由 ADC0 ADC0 转化结束中断产生的中断请求

位 0: 保留 读=0, 写=忽略

图 9.13 EIP1: 扩展中断优先级 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值:00000000
PCP1R	PCP1F	PCP0R	PCP0F	-	PWADC0	PSPI0	T0M0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xF6

位 7: PCP1R: 比较器 1 (CP1) 上跳沿中断优先级控制  
这个位设置了 CP1 中断的优先级  
0: CP1 上跳沿中断设为低优先级  
1: CP1 上跳沿中断设为高优先级

位 6: PCP1F: 比较器 1 (CP1) 下跳沿优先级控制  
这个位设置 CP1 中断的优先级  
0: CP1 下跳沿中断设为低优先级  
1: CP1 下跳沿中断设为高优先级

位 5: PCP0R: 比较器 0 (CP0) 上跳沿中断优先级控制  
这个位设置了 CP0 中断的优先级  
0: CP0 上跳沿中断设为低优先级  
1: CP0 上跳沿中断设为高优先级

位 4: PCP0F: 比较器 0 (CP0) 下跳沿中断优先级控制  
这个位设置了 CP0 中断的优先级  
0: CP0 下跳沿中断设为低优先级  
1: CP0 下跳沿中断设为高优先级

位 3: 保留 读=0 写=忽略

位 2: PWADC0: 模数转换器 0 窗口比较 (ADC0) 中断优先级控制  
这个位设置了 ADC0 窗口比较中断的优先级  
0: ADC0 窗口比较中断设为低优先级  
1: ADC0 窗口比较中断设为高优先级

位 1: 未使用 读=0 写=忽略

位 0: PSPI0: 串行外设接口 0 中断优先级控制  
这个位设置了 SPI0 中断的优先级  
0: SPI0 中断设为低优先级  
1: SPI0 中断设为高优先级

图 9.14 EIP2: 扩展中断优先级 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值:00000000
<b>PXVLD</b>	-	<b>PX7</b>	<b>PX6</b>	<b>PX5</b>	<b>PX4</b>	<b>PADC0</b>	-	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址 0xF7

位 7: PXVLD: 外部时钟源有效 (XTLVLD) 中断优先级控制  
 这个位设置了 XTLVLD 中断的优先级  
 0: XTLVLD 中断设为低优先级  
 1: XTLVLD 中断设为高优先级

位 6: 保留    必写 0    读 0

位 5: PX7: 外部中断 7 优先级控制  
 这个位设置了外部中断 7 的优先级  
 0: 外部中断 7 设为低优先级  
 1: 外部中断 7 设为高优先级

位 4: PX6: 外部中断 6 优先级控制  
 这个位设置了外部中断 6 的优先级  
 0: 外部中断 6 设为低优先级  
 1: 外部中断 6 设为高优先级

位 3: PX5: 外部中断 5 优先级控制  
 这个位设置了外部中断 5 的优先级  
 0: 外部中断 5 设为低优先级  
 1: 外部中断 5 设为高优先级

位 2: PX4: 外部中断 4 优先级控制  
 这个位设置了外部中断 4 的优先级  
 0: 外部中断 4 设为低优先级  
 1: 外部中断 4 设为高优先级

位 1: PADC0:    ADC 转化结束中断优先级控制  
 这个位设置了 ADC0 转化结束中断的优先级  
 0:    ADC0 转化结束中断设为低优先级  
 1:    ADC0 转化结束中断设为高优先级

位 0:    保留    读=0,    写=忽略

## 9.5 电源管理模式

CIP-51 核心有两种软件可编程电源管理模式：空闲和停机。在空闲模式外设及内部时钟继续运行而 CPU 停止工作。停机模式里，CPU 被停止运行，所有的中断和定时器（除了缺少的时钟检测器外）都暂停不用了。系统时钟也停止了，因为时钟在空闲模式中运行，电源消耗依赖系统时钟频率和进入空闲前的模式中外设的数量。停机模式耗电最少。图 9.15 描述了用来控制 CIP-51 电源管理模式的电源控制寄存器（PCON）。

尽管 CIP-51 具备了空闲和停机模式（同任何标准的 8051 体系结构一样），通过允许/禁止单独外设可以使整个 MCU 的电源管理更好地完成。每一个模拟外设暂停不用或输入低电源模式时都能被禁止。关掉振荡器能节省更多电源，但却要复位来重新启动 MCU。

### 9.5.1 空闲模式

置空闲模式选择位（PCON.0）在置位指令一完成就使 CIP-51 停止 CPU，并进入空闲模式。所有的内部寄存器及存储器都保存其原有数据。在空闲模式时所有的数字外设都保持活跃。在一个允许的中断或/RST 出现时，空闲模式就终止了。一个允许的中断的出现会复位空闲模式选择位（PCON.0）并重新运行 CPU。悬挂的中断将被继续执行，中断返回后执行的下一条指令将是紧随设置空闲模式选择位的指令。如果空闲模式被一个内部或外部复位终止，CIP-51 就按照正常的复位次序开始在地址 0x0000 执行程序。

如果允许的话，WDT 将最后总产生一个内部监督复位，并因此终止空闲模式。这一特征保护系统在给 PCON 寄存器一个疏忽写的情况下无意的持久关机。如果不希望出现这种情况，可以在最初允许 WDT 进行这种操作而进入空闲模式前软件禁止 WDT，这就可以额外省电，允许系统不继续空闲模式并等待外部召唤来唤醒系统。关于 WDT 的使用和配置详见 12.7 节监督定时器

### 9.5.2 停机模式

置停机模式选择位（PCON.1）在置位指令一完成就使 CIP-51 进入停机模式。CPU 和振荡器在停机模式里都停止了，有效地停止了一切数字外设，每一个模拟外设必须在进入停机模式之前单独关闭。只有内部和外部复位才能终止停机模式。复位后，CIP-51 就按照正常的复位次序开始在地址 0x0000 执行程序。

如果允许，缺少的时钟检测器将产生一个内部复位，并因此而终止停机模式，如果 CPU 的睡眠时间超过 MCD 的超时时间 100 $\mu$ sec 的话，缺少的时钟检测器应被禁止。

图 9.15 PCON: 电源控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值 00000000
<b>SMOD</b>	<b>GF4</b>	<b>GF3</b>	<b>GF2</b>	<b>GF1</b>	<b>GF0</b>	<b>STOP</b>	<b>IDLE</b>	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0x87

位 7: SMOD: 串行口波特率加倍允许  
 0: 串行口波特率由 SCON 中的串行口模式给出  
 1: 串行口波特率是 SCON 中给出的串行口模式的双倍。

位 6: GF4-GF0: 通用标志 4-0  
 这些都是软件控制下的通用标志

位 1: 停机: 停机模式选择  
 设置这一位将使 CIP-51 进入停机模式。这个位读时总为 0  
 1: 进入关闭电源模式 (关掉振荡器)

位 0: 空闲: 空闲模式选择  
 设置这个位将使 CIP-51 进入空闲模式。这个位读时总为 0  
 1: 进入空闲模式。(关掉 CPU 的时钟, 但定时器, 中断, 串行口和模拟外设的时钟仍在运行)

## 10. 闪存

这个 MCU 包括片内的 8K+128 字节的可重编程的闪存，可以存放程序代码以及非易失的数据。闪存可在系统内部编程，每次一个字节，通过 JTAG 接口或软件使用 MOVX 指令进行。一旦清除为 0，就必须把一个 FLASH 位擦为 1。这些字节在重编程之前必须被擦掉重写（设为 0xFF）。写和擦的操作都是在硬件执行下自动计时，并不要求数据查询来决定写/擦操作结束否。闪存至少能经受 10000 次的写/擦周期，参见表 10.1 闪存的电特性。

### 10.1 给闪存编程

给闪存编程的最简单方法就是通过 JTAG 接口，使用由 Cygnal 或第三方供应商提供的编程工具。这就是给唯一的未初始化设备编程的方法。有关给闪存编程的 JTAG 命令的详情，请见 18.1 节。

闪存可由软件通过使用 MOVX 指令进行编程，其所利用的地址和数据字节和利用常规操作数的编程一样。在通过 MOVX 向闪存写之前，FLASH 写操作必须要由设置 PSWE 编程储存写允许位（PSCTL.0）为逻辑 1 来允许。在 PSWE 位被软件清除之前，FLASH 写始终保持允许状态。

向闪存写能清除位但不能设置位。只有擦操作才可以在 FLASH 中设置位。因此，要编程的字节部分必须在输入新值前被擦掉。8K 字节闪存由 512 字节的扇区组成。擦的操作适用一个整个的扇区（设置扇区所有字节为 0xFF）。设置 PSEE 变成储存擦允许位（PSCTL.1 和 PSWE 编程储存写允许位（PSCTL.0）为逻辑 1，然后使用 MOVX 将一个数据字节写至扇区内的任何字节位置都将擦掉整个 512-字节扇区。这个数据字节可以是任何值因为事实上它并不写入 FLASH 中。FLASH 擦一直保持为允许，除非 PSEE 位被软件清除。接下来说明了由软件为闪存编程的算法：

1. 使用 FLASCL 位的 FLSCCL 寄存器闪存的写/擦除允许
2. 设置 PSEE（PSCTL.1）允许擦掉 FLASH 扇区
3. 设置 PSWE（PSCTL.0）允许 FLASH 写
4. 在擦掉的 512-字节扇区内部用 MOVX 来向任何一个位置写入数据字节
5. 清除 PSEE 来禁止 FLASH 扇区擦除
6. 在擦掉的 512 字节扇区内部，使用 MOVX 在要求的字节位置输入一个数据字节，反复重复至完成。（从一个单独的字节甚至一个整个扇区都可以输入）
7. 清除 PSWE 位来禁止向 FLASH 写

写/擦定时由硬件根据在闪存定时预标度寄存器（FLSCL）的预标度值自动控制。4 位预标度值 FLASCL 决定了写/擦操作的时间间隔。给出的系统时钟所要求的 FLASCL 值以及获得 FLASCL 值所需的公式见图 10.3。当 FLASCL 设置为 1111b 时，写/擦除操作就被禁止了。FLASH 正被编程或擦除时，8051 的程序执行就被停止了。

**表 10.1 闪存的电特性**

VDD=2.7---3.6V, -40---+85°C, 除非还有其他的说明

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Endurance		10k	100k		ERASE/Wr
Erase/write Cycle Time			10		ms

## 10.2. 非易失的数据存储

闪存可以贮藏非易失的数据和程序代码。这使得调整系数之类的数据可以在运行中计算和贮存。数据通过 MOVX 指令写入并通过 MOVC 指令读取。

MCU 在闪存的 0x2000–0x207F 位置有一个额外的 128-字节扇区。这个扇区可用于程序代码和数据存储。然而，它的较小扇区尺寸使它更适于常规用途，非易失的数据暂存器。尽管闪存每次能写一个单独的字节，整个扇区必须先擦除。为了改变多字节数据集中的单独字节，数据必须移到暂时贮存器。接下来，扇区被擦除，数据集被更新并返回到原来的扇区。128-字节扇区容量有助于不浪费程序存储器的情况下，从而可以使用内部数据 RAM 进行暂时存储（一个正常的 512-字节扇区过大，在 256 字节内部数据存储器中存不下）

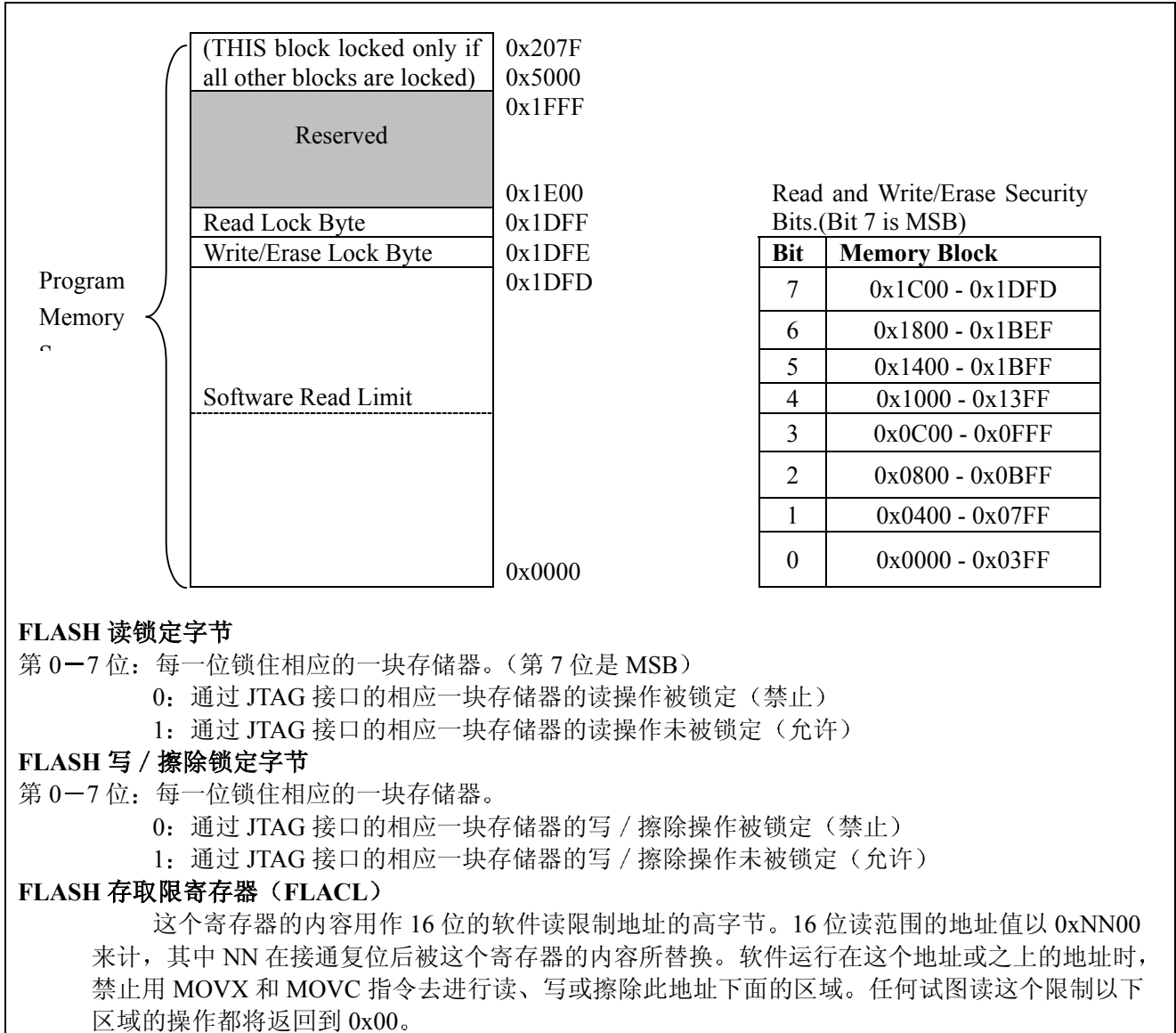
## 10.3 加密选项

CIP-51 提供了加密选项来保护闪存被软件不小心修改，或防止私有的程序代码或常量被查看。程序贮存写允许位 (PSCTL.0) 和编程贮存擦允许位 (PSCTL.1) 两位保护闪存被软件意外修改。这些位必须在软件能修改闪存前设置为逻辑 1。在 JTAG 接口中额外的加密特性避免了私有的程序代码和数据常量被系统控制器上运行的软件读或者修改。

贮存在 0x1DFF 和 0xDFE 的一套加密锁定字节防止了在 JTAG 接口中，读或修改闪存。加密锁定字节中的每一位都保护了存储器中的 1 个 1K 字节存储器块。将一个读锁定字节中的某位清 0 禁止读取相应的闪存块。在写/擦除锁定字节内清某位防止块被 JTAG 擦或是写。读锁定字节位置在 0x1DFF。写/擦除锁定字节位置在 0xDFE。图 10.1 说明了加密字节的位置及位定义。512-字节扇区含有锁定字节，不能被软件擦除。



图 10.1 FLASH 程序存储器的加密字节



**FLASH 读锁定字节**

第 0—7 位：每一位锁住相应的一块存储器。（第 7 位是 MSB）  
 0：通过 JTAG 接口的相应一块存储器的读操作被锁定（禁止）  
 1：通过 JTAG 接口的相应一块存储器的读操作未被锁定（允许）

**FLASH 写 / 擦除锁定字节**

第 0—7 位：每一位锁住相应的一块存储器。  
 0：通过 JTAG 接口的相应一块存储器的写 / 擦除操作被锁定（禁止）  
 1：通过 JTAG 接口的相应一块存储器的写 / 擦除操作未被锁定（允许）

**FLASH 存取限寄存器 (FLACL)**

这个寄存器的内容用作 16 位的软件读限制地址的高字节。16 位读范围的地址值以 0xNN00 来计，其中 NN 在接通复位后被这个寄存器的内容所替换。软件运行在这个地址或之上的地址时，禁止用 MOVX 和 MOVC 指令去进行读、写或擦除此地址下面的区域。任何试图读这个限制以下区域的操作都将返回到 0x00。

不论加密的设置情况如何，锁定位总能读和清零，这适用于包含加密字节的区域。这允许附加区域在含加密字节的块被锁住之后能被保护。然而，通过执行 JTAG 擦除操作，一旦设置成擦除整个程序存储器空间，这就是删除锁定位的唯一途径。**注：擦除包含加密字节的 FLASH 存储块时，将自动开始擦除整个程序存储器空间（除了保留区域以外）。这个擦除操作只能通过 JTAG 来执行。若 0x1C00-0x1DFD 页面内的非加密字节被进行写操作而执行该页面的擦除，那么包含加密字节的该页面将被擦除。**

CIP-51 上运行的软件要读取数据和程序代码，FLASH 存取限的加密特性可以保护所有的这些程序代码和数据，这一特性为 OEMs 提供了支持，即希望在分配之前编制具有所有增值固件的 MCU 程序。当允许用余下的程序存储器空间编制附加的代码时，增值固件能被保护。

软件读取限制 (SRL) 是一个 16 位的地址，它在程序存储器空间中建立了两个逻辑分区。第一个是上面的分区，它包含位于 SRL 地址或其上的所有程序存储器区域，第二个是下面的分区，它包含从 0x0000（不含该地址）到 SRL 地址的所有程序存储器区域。上面分区中的软件能执行下面分区中的代码，但禁止用 MOVC 指令读下面分区中的某一位置。（确定了下面分区中的一个源地址，在上面分区中执行 MOVC 指令时，将总是返回到数据值 0x00。）下面分区中的软件运行时，能无限制地访问上、下两个分区。

增值固件将放于下面的分区，接通复位后，控制通过复位向量被传送到增值固件。增值固件一旦完成其最初的执行后，就分支到上面分区中一个预定的位置。如果入口指针被公开，上面分区中的软件运行就可以执行下面分区中的程序代码，但是它不能读取下面分区中的内容。参数可以这样传送到下面分区中运行的程序代码：在调用这些参数或将这些参数放于上面分区中规定的存储器区域之前，通过将这些参数放置于堆栈上或寄存器中的典型方法来实现。

通过使用 FLASH 存取寄存器的内容规定 SRL 地址，16 位 SRL 地址以 0xNN00 来计，其中 NN 是 SRL 加密寄存器的内容，因而，SRL 能定位于 256 字节程序存储器空间中的任一边界。然而，512 字节的擦除扇区大小基本上需要 512 字节的边界。一个未初始化的 SRL 加密字节的内容是 0x00，藉此，在缺省情况下设置 SRL 地址为 0x0000，并且允许读取所有的程序存储器空间。

**图 10.2. PSCTL: 程序存储 RW 控制**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
-	-	-	-	-	-	RSEE	RSWE	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0x8F

第 2—7 位：未用。读=000000b,写=忽略

第 1 位：PSEE：程序存储擦除允许  
 设置这个位可以允许擦除 FLASH 程序存储器的整个页面（规定 PSWE 位为 1），设置这个位之后，使用 MOVX 指令写 FLASH 存储器时，将擦除 MOVX 指令寻址的整个页面。写数据字节的值没有关系。  
 0：FLASH 程序存储器擦除被禁止  
 1：FLASH 程序存储器擦除被允许

第 0 位：PSWE：程序存储写允许  
 设置这个位可以允许用 MOVX 指令写数据字节到 FLASH 程序存储器的某一位置，在进行写操作之前，这个位置必须已经被擦除。  
 0：FLASH 程序存储器写被禁止  
 1：FLASH 程序存储器写被允许

图 10.3. FLASCL: FLASH 存储器计时预定标器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 10001111
POSE	FRAE	-	-	FLASCL				
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xB6

第 7 位: FOSE: FLASH 单步定时器允许  
 0: FLASH 单步定时器禁止  
 1: FLASH 单步定时器允许

第 6 位: FRAE: FLASH 读总被允许  
 0: 按单步定时器进行 FLASH 的读操作  
 1: FLASH 总处于读操作方式

第 4, 5 位: 未用。读=00b, 写=忽略

第 0-3 位: FLASCL: FLASH 存储器计时预定标器  
 对一个给定的系统时钟, 当进行 FLASH 写 / 擦除操作时, 需产生正确的时序, 这个寄存器规定了预定标器的值, 若这个预定标器的值被设置成 111b, 则 FLASH 的写 / 擦除操作被禁止。

0000: 系统时钟 < 50KHz  
 0001: 50KHz ≤ 系统时钟 < 100KHz  
 0010: 100KHz ≤ 系统时钟 < 200KHz  
 0011: 200KHz ≤ 系统时钟 < 400KHz  
 0100: 400KHz ≤ 系统时钟 < 800KHz  
 0101: 800KHz ≤ 系统时钟 < 1.6MHz  
 0110: 1.6MHz ≤ 系统时钟 < 3.2MHz  
 0111: 3.2MHz ≤ 系统时钟 < 6.4MHz  
 1000: 6.4MHz ≤ 系统时钟 < 12.8MHz  
 1001: 12.8MHz ≤ 系统时钟 < 25.6MHz  
 1010: 25.6MHz ≤ 系统时钟 < 51.2MHz  
 1011, 1100, 1101, 1110: 保留值  
 1111: FLASH 存储器写 / 擦除被禁止

预定标器的值是满足以下等式的极小值:  
 $FLASCL > \log_2(\text{系统时钟} / 50\text{KHz})$

\* 测试的时候: 超过 25MHz 时不能保证 C8051F2xx 的操作。

图 10.4. FLACL: FLASH 存取限制

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xB7

第 0-7 位: FLACL: FLASH 存储器存取限  
 这个寄存器保持有 16 位程序存储器读 / 写 / 擦除范围地址的高字节, 所有的 16 位存取范围地址以 0xNN00 来计, 其中 NN 被 FLACL 的内容所替换。写这个寄存器可以设置 FLASH 的存取限, 这之后的任意一次写操作在复位之前均被忽略

### 11. 片内 XRAM (C8051F226 / 236 / 206)

C8051F226 / F236 / F206 的特点是有 1024 字节的 RAM 映象到外部数据存储器空间。所有的地址区域均可用外部传送指令 (MOVX) 和数据指针 (DPTR) 来访问, 或者用间接 MOVX 寻址方式。如果 MOVX 指令被用作 8 位操作数 (例如 @R1), 则在图 11.1 中示出了外部存储器的接口控制寄存器 (EMI0CN) 的高字节, 8 位寻址将被映象至 4 个 256 字节页面中的一个, 这些页面通过设置寄存器 EMI0CN 中的 PGSEL 位来选择。

**注: MOVX 指令也被用于 FLASH 存储器的写操作, 进一步的叙述参看第 10 节。缺省状态时 MOVX 指令将访问 XRAM。**

对任意一个寻址方式, 16 位外部数据存储器地址字的上面 6 位“忽略”。1024 字节的 RAM 被超过所有 64k 可能的地址值的模式 (“环结式处理”) 所映象, 例如: 在地址 0x0000 处的 XRAM 字节同时也在 0x0800, 0xC00, 0x1000 等等。当达到 RAM 块的边界, 地址指针不存在着复位时, 上述特性用于线性存储器填充的情况。

图 11.1. 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: 00000000
-	-	-	-	-	-	PGSEL1	PGSEL0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xAF

第 2—7 位: 未使用, 只读=000000b  
 第 0—1 位: XRAM 页面选择位 PGSEL[1: 0]  
 当使用 8 位 MOVX 命令, 有效地选择了 RAM256 字节的页面时, XRAM 页面选择位提供了 16 位外部存储器地址的高字节。上面 6 位“忽略”。这样 1k 的地址块被超过整个数据存储器地址空间的模所重复。

00: 0x000-0x0FF  
 01: 0x100-0x1FF  
 10: 0x200-0x2FF  
 11: 0x300-0x3FF

## 12. 复位源

MCU 的复位电路系统可以使控制器很容易地处于预先定义的缺省条件之中。一旦进入复位状态，CIP-51 就停止程序的执行，强制外部端口的端子为某个已知的状态，并且将 SFRs 初始化为被定义好的复位值，中断和定时器也被屏蔽。退出复位状态之后，程序计数器（PC）复位，程序从 0x0000 处开始执行。

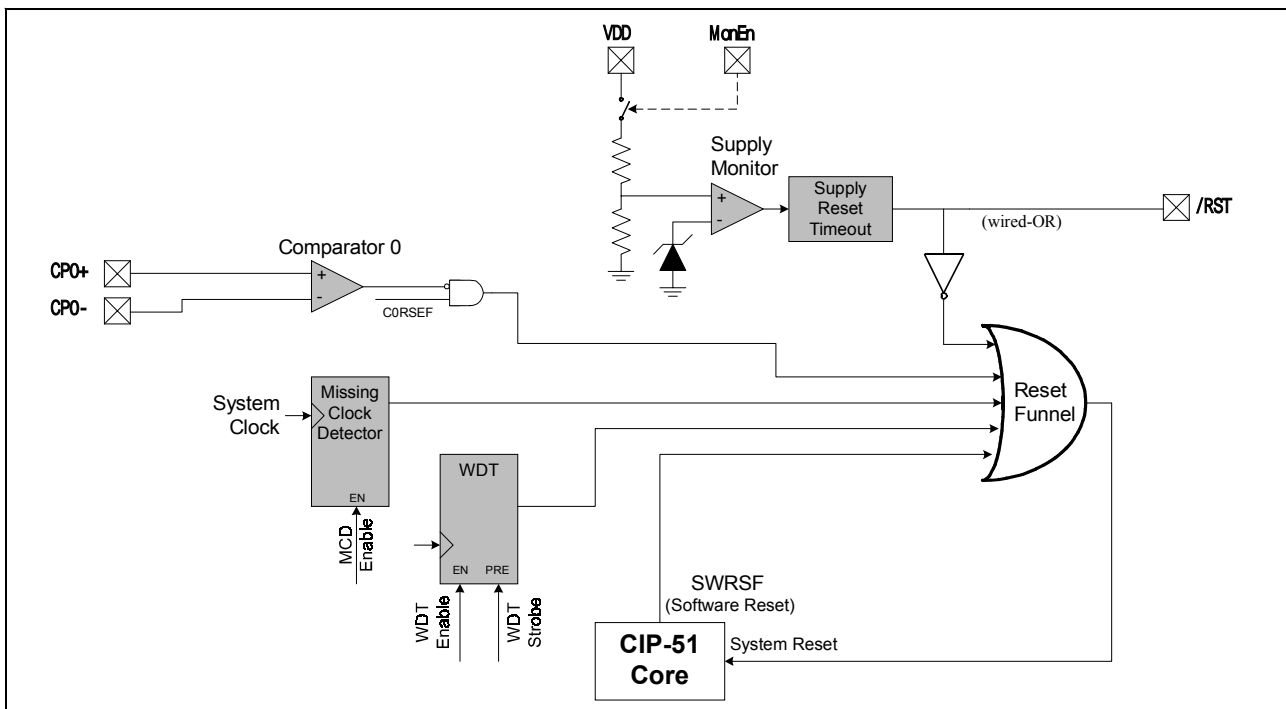
所有的 SFRs 复位到预先定义好的值，SFR 位的复位值通过 SFR 的详细描述来定义。内部数据存储器的内容在发生复位和保存任何先前的数据之间不会改变。但堆栈指针 SFR 复位时，即使堆栈上的数据未改变，堆栈还是会丢失。

I/O 端口锁存复位到 0xFF（全部为逻辑 1），激活内部弱上拉会使得外部 I/O 端子变成高状态。但外部 I/O 端子不会立即变成高，当进入复位状态后，在 4 个系统时钟周期内它将变为高。如果复位源是从 VDD 监视器或向 PORSF 位写“1”时来的，则 /RST 端子将变为低，一直到 VDD 复位超时结束为止。

退出复位状态后，MCU 运行于内部振荡器，其频率为 2MHz，它是缺省的系统时钟。参考第 13 节关于选择和配置系统时钟源的信息。看门狗定时器的允许是通过超时的间隔极大值来实现的（12.7 节有看门狗定时器的详细使用方法）。一旦系统时钟源稳定下来，程序就从 0x0000 处开始执行。

有 6 个复位源可将 MCU 导至复位状态，它们是：上电 / 电源失效（VDD 监视器），外部 / RST 端子，软件命令，比较器 0，时钟丢失检测器，看门狗定时器。每一个复位源的描述如下图：

图 12.1. 复位源框图



### 12.1. 上电复位

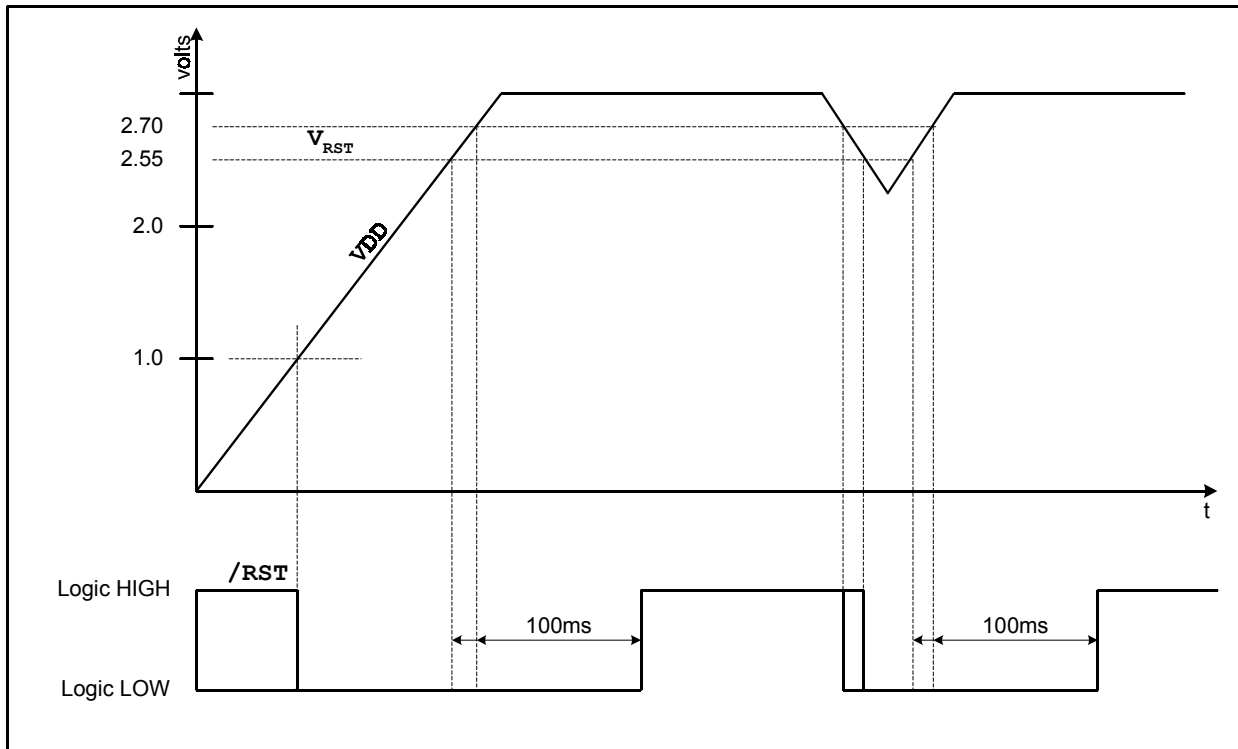
CIP-51 有一个电源监视器，它可以保持 MCU 的复位状态，一直到 VDD 在加电期间上升到  $V_{RST}$  以上时为止（参看图 12.2 的计时框图，参考表 12.1 中电源监视器电路的电特性）。 $\overline{RST}$  端子在 VDD 监视器超时满 100 毫秒之前是低电平，这是为了让 VDD 电源变得稳定。VDD 监视器可以被屏蔽，方法是将 MONEN 端子拉成低电平。

从上电复位退出后，PORSF 标志（PSTSRC.1）由硬件置成逻辑 1，PSTSRC 寄存器中的其它复位标志是不确定的，它可通过其它的所有复位来清除。所有复位均使程序在 0x0000 处开始执行，从此时起，软件能读取 PORSF 标志以确定加电是否引起了复位。假定内部数据存储器的内容在上电复位之后没有定义。

### 12.2. 软件强制复位

向 PORSF 位写 1 时，强制上电复位（参看 12.1 节）。

图 12.2. VDD 监视器定时框图



### 12.3. 电源失效复位

VDD 监视器的允许有这样几种方式：MONEN 端子（32 端子的 C8051F221 / F231 中没有）被拉高，断电转换或电源不规则引起 VDD 下降至  $V_{RST}$  以下。此时电源监视器将驱使  $\overline{RST}$  端子变低并且使 CIP-51 返回至复位状态（见图 12.2）。当 VDD 返回到  $V_{RST}$  以上时，CIP-51 将离开复位状态，这和上电复位是同一个方式。注意即使内部数据存储器的内容在电源失效时没有改变，它也不能决定 VDD 是否降到了数据保留需要的电平以下。如果 PORSF 标志已置位，数据就不再有效。

#### 12.4. 外部复位

外部 /RST 端子提供了这样一种手段，就是可用外部电路系统去强制 CIP-51 进入到复位状态。可以断定，/RST 端子上的低激活信号将使 CIP-51 处于复位状态，虽然有一个弱上拉，也可以禁止提供一个外部上拉和（或）/RST 端子的去藕，以避免不正确的噪声感应。在低激活 /RST 信号被删除后，至少有 12 个时钟周期的时间，此时 CIP-51 仍然处于复位状态。PINRSF 标志（RSR SRC.0）在从外部复位退出后被置位。/RST 端子的耐压为 5V。

#### 12.5. 时钟丢失检测器复位

时钟丢失检测器实质是一个单冲电路，它被 MCU 的系统时钟所触发，如果系统时钟大于 100 微秒，单冲将超时并产生复位。时钟丢失检测器复位后，MCDRSF 标志（RST SRC.2）将被置位，表示 MSD 是复位源；否则，该位是零。/RST 端子的状态不受这个复位的影响，通过设置 OSCICN 寄存器中的 MSCLKE 位（参见图 13.2），可以允许时钟丢失检测器。

#### 12.6. 比较器 0 复位

通过向 CORSEF 标志（RST SRC.5）写 1，比较器 0 能配置为复位输入。比较器使用 CPT0CN.7（图 8.3）来允许，这一点优先于写 CORSEF，当发生不期望的复位时，可避免所有的接通抖动。当配置为复位输入时，如果非反相端（CP0+）的输入电压小于反相端（CP0-）输入电压，MCU 就会进入到复位状态。比较器 0 复位之后，CORSEF 标志（RST SRC.5）将为 1，表示比较器 0 是复位源；否则，该位是 0。/RST 端子的状态不受这个复位的影响。

#### 12.7. 看门狗定时器复位

MCU 包含一个可编程的看门狗定时器（WDT），独立运行于系统时钟之外，当 WDT 溢出时，将强制 MCU 进入复位状态。若要防止复位，WDT 必须在发生溢出之前，由应用软件来重启。如果系统遇到软 / 硬件的不正常，在防止软件重启 WDT 的情况下，WDT 将溢出并且引起复位。这一点可以避免系统的运行处于不受控状态。

从总复位退出之后，WDT 自动允许和启动，且具有缺省的最大时间间隔。如果需要，WDT 可以被系统软件禁止，或者锁定“开”以避免意外的屏蔽发生。一旦被锁定，WDT 就不能被屏蔽，一直到下一个系统复位产生为止。/RST 端子的状态不受这个复位的影响。



## 12.7.1 看门狗的使用

WDT 包含一个 21 位的定时器，运行于程序控制的系统时钟。定时器测量专用的写控制寄存器的周期，如果这个周期超过了程序设置的范围，WDT 复位就会产生。作为需要，WDT 可以通过软件来允许和禁止，也可以根据需要使之永久地被允许。看门狗的特性是：可以通过看门狗定时器控制寄存器（WDTCN）来控制（图 12.3）。

### WDT 的允许 / 复位

看门狗定时器的允许和复位是通过向 WDTCN 寄存器写 0xA5 来实现的，用户的应用软件中，包括周期性的向 WDTCN 写 0xA5，这可以防止看门狗定时器的溢出。WDT 的允许和复位是任意一个系统复位的结果。

### 屏蔽 WDT

向 WDTCN 寄存器写 0xDE，紧接着写 0xAD，可以屏蔽掉 WDT，下面的一段代码举例说明了如何屏蔽 WDT：

```

CLR    EA                ; 屏蔽所有的中断
MOV    WDTCN, #0DEh     ; 屏蔽看门狗定时器
MOV    WDTCN, #0ADh     ;
SETB   EA                ; 中断再允许
    
```

写 0xDE 和 0xAD 时必须在 4 个系统时钟周期之内完成，或者是屏蔽的操作被忽略不计。在进行两次写操作时，中断将被屏蔽，以避免延迟。

### 屏蔽 WDT 锁定

向 WDTCN 写 0xFF 可以锁定 WDT 的屏蔽特性，一旦被锁定，屏蔽操作将被忽略不计，直到下一个系统复位来为止。写 0xFF 不能允许或复位看门狗定时器，应用程序总要使用看门狗，所以用初始化代码向 WDTCN 写 0xFF。

### 设置 WDT 间隔

WDTCN.[2:0]控制看门狗超时间隔，下面的等式给出了间隔的设置：

$$4^{3+WDTCN[2:0]} \times T_{SYSCLK}, \text{其中 } T_{SYSCLK} \text{ 是系统时钟周期。}$$

对于 2.0MHz 的系统时钟，这一等式给出了从 32 毫秒到 524 毫秒的间隔范围。当设置这个间隔时，WDTCN.7 必须被写为 0，读 WDTCN 返回程序控制的间隔。系统复位之后，WDTCN.[2:0]是 111b。

**图 12.3. WDTCN：看门狗定时器控制寄存器**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值：xxxx111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址：0xFF

:

第 0—7 位：WDT 控制  
 写 0xA5 可允许和重载 WDT  
 4 个时钟内，写 0xDE 和 0xAD，可屏蔽 WDT  
 写 0xFF 可锁定 WDT 的屏蔽特性

第 4 位：看门狗状态位（读的时候）。读 WDTCN.[4]位指示出 WDT 的状态  
 0：WDT 未激活  
 1：WDT 激活

第 0—2 位：看门狗超时间隔位。  
 WDTCN.[2:0]位用来设置看门狗的超时间隔，当写这些位时，WDTCN.7 必须要设置成 0



图 12.4. RSTSRC: 复位源寄存器

	R	R/W	R/W	R	R	R/W	R	复位值: xxxxxxxx
-		CORSEF	SWRSEF	WDTRSF	MCDRSF	PORSF	PINRSF	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xEF

(注: 对这个寄存器不要使用读—修改—写的操作。)

第 7 位: 保留

第 6 位: 未用。只读=0b

第 5 位: **CORSEF**: 比较器 0 复位允许和标志  
写

0: 比较器 0 不是复位源

1: 比较器 0 是复位源

读

0: 先前的复位源不是来自比较器 0

1: 先前的复位源来自比较器 0

第 4 位: **SWRSF**: 软件复位强制和标志  
写

0: 没有影响

1: 强制一个内部复位, /RST 端子不受影响

读

0: 先前的复位源不是来自于写 **SWRSF** 位

1: 先前的复位源来自于写 **SWRSF** 位

第 3 位: **WDTRSF**: WDT 复位标志 (只读)

0: 先前的复位源不是来自 WDT 超时

1: 先前的复位源来自于 WDT 超时

第 2 位: **MCDRSF**: 时钟丢失检测器标志 (只读)

0: 先前的复位源不是来自时钟丢失检测器超时

1: 先前的复位源来自于时钟丢失检测器超时

第 1 位: **PORSF**: 上电复位强制和标志  
写

0: 没有影响

1: 强制上电复位, /RST 端子变为低

读

0: 先前的复位源不是来自于 POR

1: 先前的复位源来自于 POR

第 0 位: **PINRSF**: HW 端子复位标志

0: 先前的复位源不是来自于 /RST 端子

1: 先前的复位源来自于 /RST 端子

表 12.1. VDD 监视器电特性

除非特别说明，温度范围是：-40°C到+85°C

参量	条件	最小值	典型值	最大值	单 位
/RST 输出低电压	$I_{OL}=8.5mA, VDD=2.7V-3.6V$				V
/RST 输入高电压		0.8x VDD			V
/RST 输入低电压				0.2x VDD	V
/RST 输入泄漏低电流	/RST=0.0V			50	$\mu A$
VDD 对 /RST 输出有效		1.0			V
复位阈值 ( $V_{rst}$ )		2.40	2.55	2.70	V
复位时间延迟	通过复位阈值后/RST 的上升沿	80	100	120	ms
时钟丢失检测器超时	从上一个系统时钟到发生复位的时间	100	220	500	$\mu S$

### 13. 振荡器

MCU 包括一个内部振荡器和一个外部振荡器驱动电路，这两个振荡器均可产生系统时钟。任意一个复位之后，MCU 将从内部振荡器引导。内部振荡器可以被允许 / 禁止，其振荡频率使用内部振荡器控制寄存器（OSCICN）来设置，参见图 13.2。表 13.1 给出了内部振荡器的电特性。

当 / RST 端子保持低时，这两类振荡器被禁止。MCU 能永久地运行于内部振荡器，如果想使用 OSCICN 寄存器中的 CLKSL 位，它也能切换至外部振荡器。外部振荡器要求有一个外部谐振器，晶振，电容或 RC 网络连接至 XTAL1 / XTAL2 端子（图 13.1），振荡器电路必须通过 OSCXCN 寄存器配置成这些振荡源中的一个。一个外部 CMOS 时钟也可以提供系统时钟，这是通过驱动 XTAL1 端子来实现的，XTAL1 和 XTAL2 端子的耐压不充许为 5V。

图 13.1. 振荡器框图

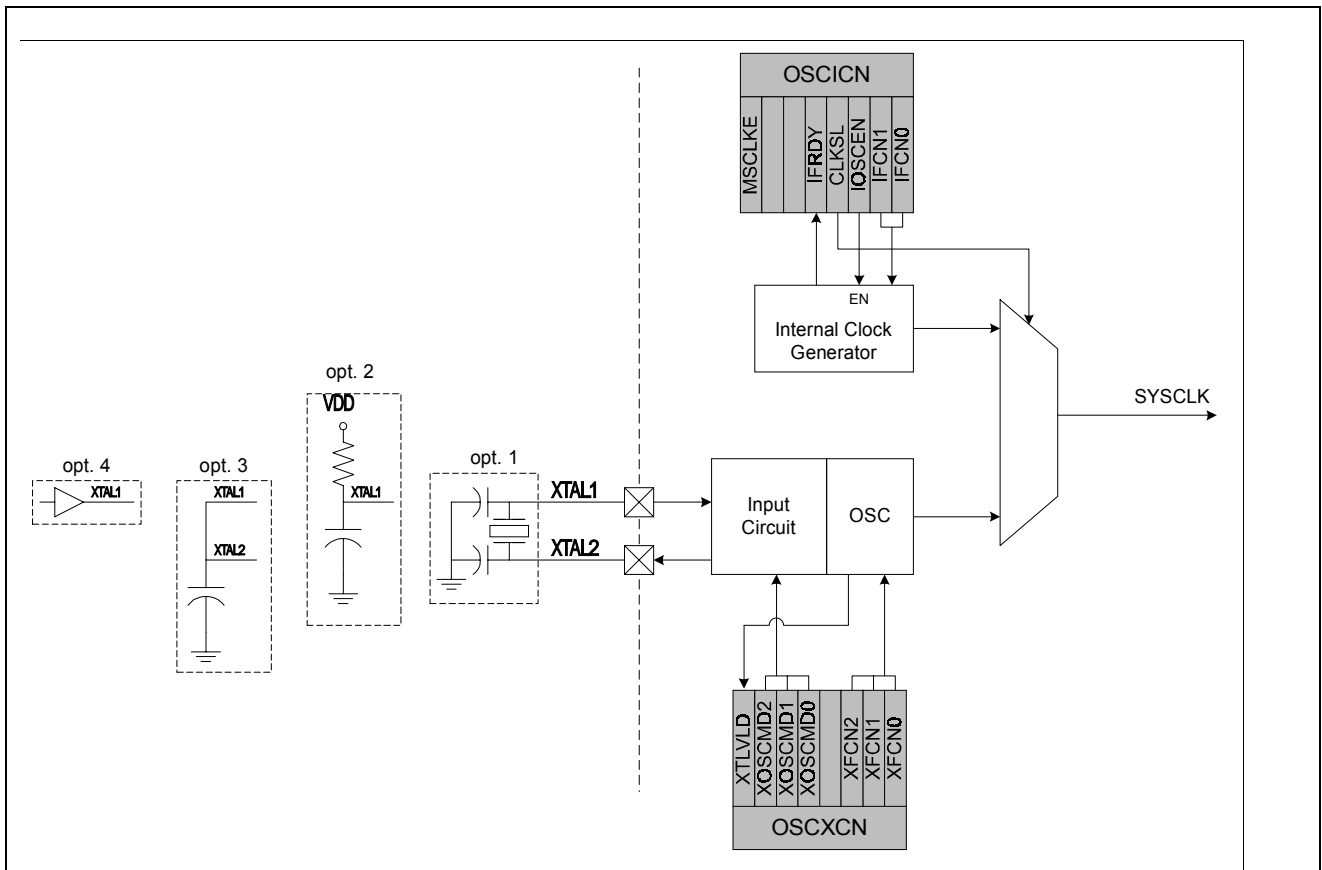


图 13.2. OSCICN: 内部振荡器控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000100
MSC_LKE	-	-	IFRDY	CLKSL	IOSCEN	IFCN1	IFCN0	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR 地址: 0xB2

第 7 位: MSC\_LKE: 时钟丢失允许位  
 0: 时钟丢失检测器禁止  
 1: 时钟丢失检测器允许; 若检测到时钟丢失, 则触发一个复位

第 5,6 位: 未用。读=00b, 写=忽略

第 4 位: IFRDY: 内部振荡器频率就绪标志  
 0: 内部振荡器频率未运行于 IFCN 位规定的速度上  
 1: 内部振荡器频率运行于 IFCN 位规定的速度上

第 3 位: CLKSL: 系统时钟源选择位  
 0: 使用内部振荡器作为系统时钟  
 1: 使用外部振荡器作为系统时钟

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

第 0,1 位: IFCN1—0: 内部振荡器频率控制位  
 00: 内部振荡器的典型频率是 2MHz  
 01: 内部振荡器的典型频率是 4MHz  
 10: 内部振荡器的典型频率是 8MHz  
 11: 内部振荡器的典型频率是 16MHz

表 13.1. 内部振荡器电特性

除非特别说明, 温度范围是: -40°C到+85°C

参量	条件	最小值	典型值	最大值	单位
内部振荡器频率	OSCICN.[1:0]=00	1.6	2	2.4	MHz
	OSCICN.[1:0]=01	3.2	4	4.8	
	OSCICN.[1:0]=10	6.4	8	9.6	
	OSCICN.[1:0]=11	12.8	16	19.2	
内部振荡器的电流损耗	OSCICN.2=1		200		μ A
内部振荡器的温度稳定性			4	0.2x VDD	ppm/°C
内部振荡器的电源 (VDD) 稳定见性	/RST=0.0V		6.4	50	%/V

图 13.3. OSCXCN: 外部振荡器控制寄存器

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 0110000
XTLVLD	XOSCND2	XOSCND1	XOSCND0	-	XFCN2	XFCN1	XFCN0	SFR 地址: 0XB1
Bit7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

第 7 位: XTLVLD: 晶体振荡器有效标志

( 只有在 XOSCND=1xx.才有效)

0: 晶体振荡器没有使用或还未稳定

1: 晶体振荡器正在运行且稳定

第 6-4 位: XOSCND2-0: 外部振荡器模式位

00x:关闭。XTAL1 引脚内部接地。

010: 来自 XTAL1 引脚的外部 CMOS 时钟的系统时钟。

011: 来自 XTAL1 引脚的外部 CMOS 时钟的系统时钟 2 除。

10x: 2 除 RC/C 振荡器模式。

110: 晶体振荡器模式

111: 2 除晶体振荡器模式。

第 3 位: 保留。读=未定义, 写=无关

第 2-0 位: XFCN2-0: 外部振荡器频率控制位

000-111: 参见下表

XFCN	Crystal(XOSCND=11x)	RC(XOSCND=10x)	C(XOSCND=10x)
000	功率因子 =30(10 <sup>3</sup> )	f≤25kHz	K 因子=0.44
001	功率因子 =90(10 <sup>3</sup> )	25kHz<f≤50kHz	K 因子=1.4
010	功率因子 =260(10 <sup>3</sup> )	50kHz<f≤100kHz	K 因子=4.4
011	功率因子 =740(10 <sup>3</sup> )	100kHz<f≤200kHz	K 因子=13
100	功率因子 =2.10(10 <sup>6</sup> )	200kHz<f≤400kHz	K 因子=38
101	功率因子 =5.80(10 <sup>6</sup> )	400kHz<f≤800kHz	K 因子=100
110	功率因子 =22.0(10 <sup>6</sup> )	800kHz<f≤1.6MHz	K 因子=420
111	功率因子 =65.0(10 <sup>6</sup> )	1.6MHz<f≤3.2MHz	K 因子=1400

晶体模式 (图 13.1 电路, 选项 1; XOSCND=11x)

根据晶体频率选择 XFCN 值。

RC 模式 (图 13.1 电路, 选项 2; XOSCND=10x)

选择振荡频率范围, 其中:

$f=1.23 (10^3)/(R*C)$ , 其中

f=振荡频率 (MHZ)

C=电容值 (pF)

R=上拉电阻值 (kΩ)

C 模式 (图 13.1 电路, 选项 3; XOSCND=10x)

根据期望振荡频率选择 K 因子 (KF)

$f=KF/(C*AV+)$ , 其中

f=振荡频率 (MHZ)

C=在 XTAL1, XTAL2 引脚上的电容值 (pF)

AV+=MCU 模拟电源 (伏)

### 13.1. 外部晶振示例

如果用晶体产生 MCU 的系统时钟，电路会如图 13.1 选项 1 所示。对于 ECS-110.5-20-4 晶体，共振频率是 11.0592MHz，固有电容是 7pF，ESR 是 60Ω。每个补偿电容 33pF，并且 PWB 寄生电容估计为 2pF。在图 13.3（OSCXCN 寄存器）表中晶体列的适当外部振荡器频率控制值（XFCN）应该是 111b。

当晶体振荡器使能时，会在晶体驱动器输出 XTAL2 上产生瞬间脉冲，这个脉冲足以使在晶振启动之前，OSCXCN 中的 XTLVLD 位变为 1。在使晶体振荡器使能和检验 XTLVLD 位之间引入 1 毫秒的消隐间隔可以防止过早地切换到外部振荡器。推荐的过程是：

1. 使能外部振荡器
2. 等待 1 毫秒
3. 查询 XTLVLD 是否由 0=>1
4. 转到外部振荡器

在晶体振荡器稳定之前转到外部振荡器会引起不可预知的行为。

**注：**晶体振荡器电路对 PCB 布线非常敏感。晶体应尽可能靠近设备的 XTAL 引脚，使其间轨迹越短越好，并进行接地以屏蔽任何可能引入噪声或干扰的其它线路。

### 13.2 外部 RC 示例

如果用外部 RC 网络为 MCU 产生系统时钟，电路如图 13.1 选项 2 所示。电容不得超过 100pF，但是电容过小会增加 PWB 寄生电容造成的频率漂移。为了确定 OSCXCN 寄存器中的外部振荡器频率控制值（XFCN），首先选择 RC 网络值来产生期望的振荡频率。如果期望的频率是 100KHz，令 R=246KΩ并且 C=50pF:

$$f=1.23(10^3)/RC=1.23(10^3)/[246*50]=0.1\text{MHz}=100\text{KHz}$$

$$XFCN \geq \log_2(f/25\text{KHz})$$

$$XFCN \geq \log_2(100\text{KHz}/25\text{KHz})=\log_2 4$$

$$XFCN \geq 2, \text{代码 } 010$$

### 13.3. 外部电容器示例

如果用外部电容器为 MCU 产生系统时钟，电路如图 13.1，选项 3 所示。电容不得超过 100pF，但是电容过小会增加 PWB 寄生电容造成的频率漂移。为确定在 OSCXCN 寄存器中的外部振荡器频率控制值（XFCN），选择所用的电容器并且从以下等式找出振荡器频率。假设 VDD=3V，C=50pF

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

$$f=KF/150$$

如果需要大约 90KHz 的频率，从图 13.3 的表中选择 K 因子为 KF=13。

$$f=13/150=0.087\text{MHz}, \text{或 } 87\text{KHz}$$

因此，在这个例子中使用的 XFCN 值是 011。

## 14. 端口输入/输出

### 描述

C8051F221/231 有 3 个 I/O 端口：端口 0，端口 1 和端口 2。C8051F206，C8051F220/6 和 C8051F230/6 有 4 个 I/O 端口：端口 0，端口 1，端口 2 和端口 3。可以通过对这些端口的相应多路转换器进行简单配置将数字资源宽数组分配给该端口。参见图 8.1。另外，所有外部端口引脚都可用作模拟输入。

### 14.1 端口 I/O 初始化

端口 I/O 的初始化比较直观。为了选择设计所要求的数字 I/O 功能，寄存器 PRT0MX，PRT1MX 和 PRT2MX 必须载入合适的值。I/O 引脚的输出驱动特性由端口配置寄存器 PRT0CF，PRT1CF，PRT2CF 和 PRT3CF 来定义。每个端口输出驱动器都可以配置为开漏或者推挽方式。即使对于在 PRTnMX 寄存器中选择的数字资源也是如此。

任意引脚都可以配置为数字 I/O 或模拟输入。缺省模式是数字 I/O。然后按本节所述用特殊功能寄存器 P0MODE、P1MODE、P2MODE 和 P3MODE 将端口引脚配置为数字或模拟方式。

最后一步是利用适当的设置寄存器对选择的各资源进行初始化。各种数字资源的初始化过程可以在每项可用功能的详细说明中找到。每个寄存器的复位状态示于其各自的描述图中。

**注：**配置用于定时器 0、1 或 2 的引脚的输入模式必须进行手工配置。

1. 无论该端口引脚是标准的通用 I/O 或由数字外设来控制，都必须对所有端口引脚的输出模式进行配置。
2. 对于所有作为定时器输入（P0.4/T0，P0.5/T1，P0.6/T2 和 P0.7/T2EX）的引脚，输出模式必须是开漏（复位状态），并且必须将“1”写到相关的端口引脚以避免引起端口引脚冲突和造成过流。例如，欲配置定时器 0，将 PRT0MX 的 T0E 定时器 0 使能位置为 1 以将定时器 0 路由到端口引脚 P0.4。然后将 P0.4/T0 配置为开漏模式（缺省在 PRT0CF 中设置），并作为数字外设输入在 P0.4 中写“1”将其输出变为高阻状态（在复位后端口引脚也缺省为逻辑高状态）。最后，确认 P0MODE4 是“1”对于数字输入模式（在复位后所有端口引脚也缺省为数字输入模式）。

图 14.1 端口 I/O 功能框图

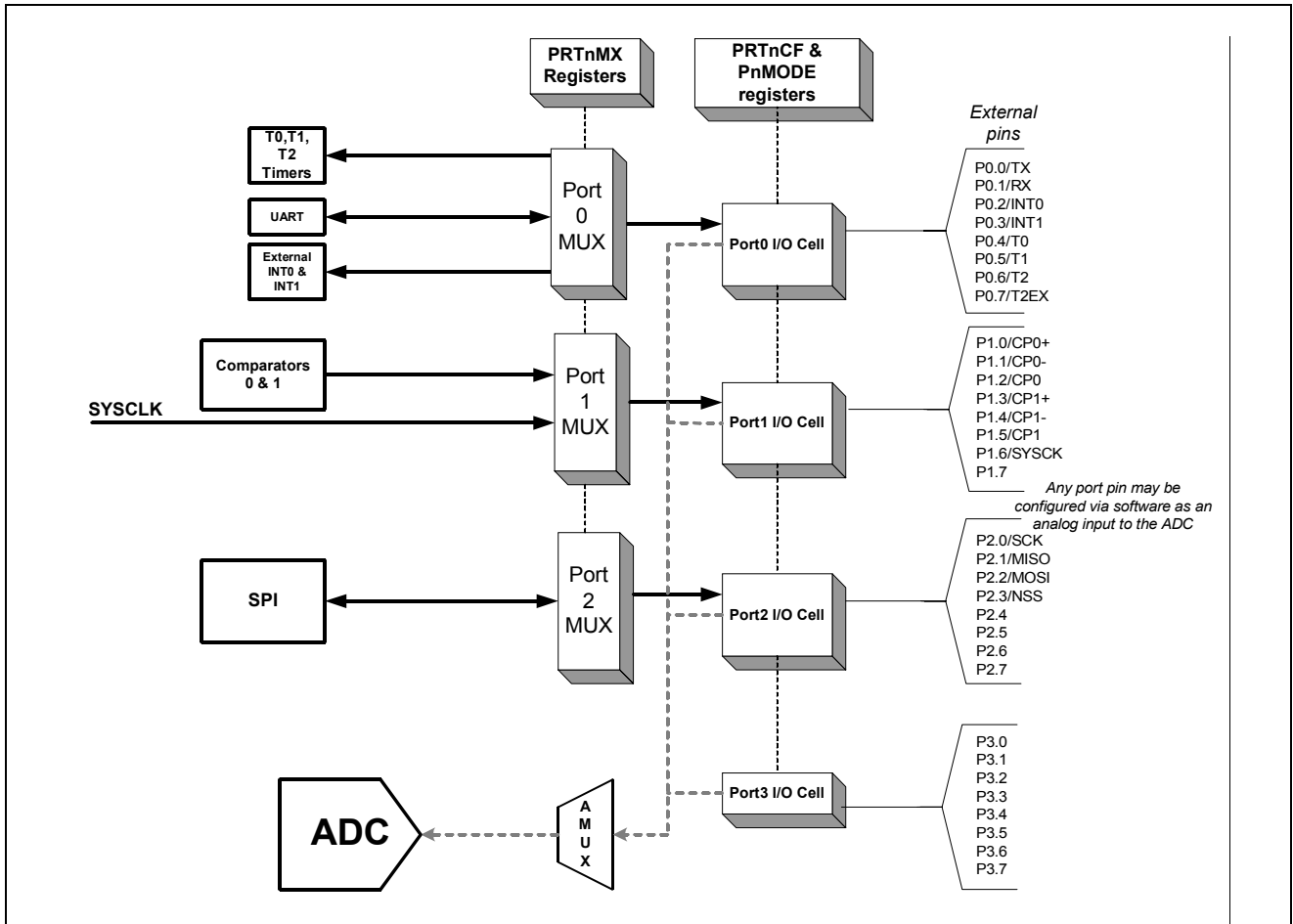


图 14.2 端口 I/O 单元框图

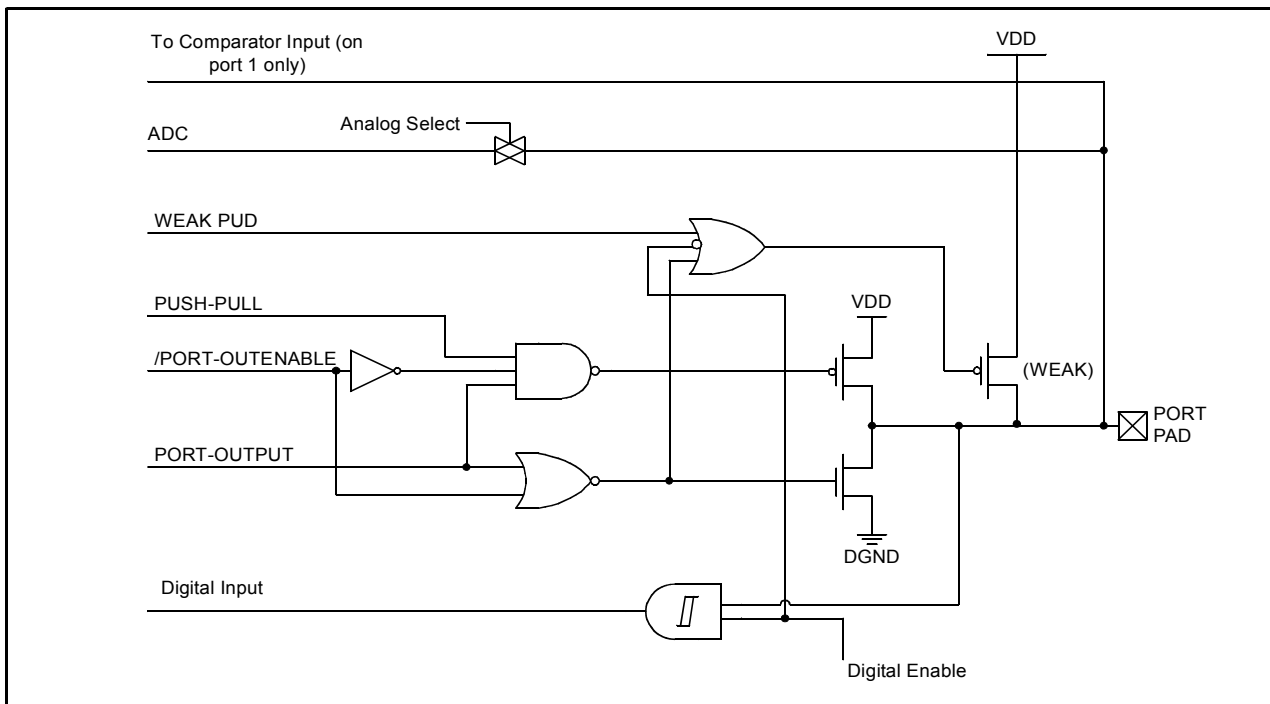




图 14.3 PRT0MX: 端口 I/O 多路转换寄存器 0

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	复位值: 0000000
T2EXE	T2E	T1E	T0E	INT1E	INT0E	-	UARTEN	SFR 地址: 0XE1
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

第 7 位: T2EXE: T2EX 使能位  
 0: T2EX 在端口引脚不可用  
 1: T2EX 路由至端口引脚。

第 6 位: T2E: T2 使能位  
 0: T2 在端口引脚不可用  
 1: T2 路由至端口引脚。

第 5 位: T1E: T1 使能位  
 0: T1 在端口引脚不可用  
 1: T1 路由至端口引脚。

第 4 位: T0E: T0 使能位  
 0: T0 在端口引脚不可用  
 1: T0 路由至端口引脚。

第 3 位: INT1E: /INT1 使能位  
 0: /INT1 在端口引脚不可用  
 1: /INT1 路由至端口引脚。

第 2 位: INT0E: /INT0 使能位  
 0: /INT0 在端口引脚不可用  
 1: /INT0 路由至端口引脚。

第 1 位: 未用, 读=0, 写=无关

第 0 位: UARTEN: UART I/O 使能  
 0: UART I/O 在端口引脚不可用  
 1: TX, RX 分别路由至引脚 P0.0 和 P0.1。

**图 14.4 PRT1MX: 端口 I/O 多路转换寄存器 1**

R	R/W	R	R	R	R	R/W	R/W	复位值: 0000000
-	SYSCKE	-	-	-	-	CPIOEN	CPO0EN	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XE2

第 7 位: 未用。读=0

第 6 位: **SYSCKE**: 系统时钟输出使能位  
 0: 在端口引脚系统时钟不能得到  
 1: 系统时钟输出指向 P1.6

第 5-2 位: 未用, 读=0000b, 写=无关

第 1 位: **CPIOEN**: 比较器 1 输出使能位  
 0: 在端口引脚 CPI 得不到  
 1: CPI 指向端口引脚 P1.5

第 0 位: **CPO0EN**: 比较器 0 输出使能位  
 0: 在端口引脚 CP0 得不到  
 1: CP0 指向端口引脚 P1.2

**图 14.5 PRT2MX: 端口 I/O 多路转换寄存器 2**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 0000000
GWPUD	P3WPUD	P2WPUD	P1WPUD	P0WPUD	-	-	SPIO0EN	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XE3

第 7 位: **GWPUD**: 全程端口 I/O 弱上拉禁止位  
 0: 对于所有的端口弱上拉使能。  
 1: 弱上拉禁止 (6-3 位无关)

第 6 位: **P3WPUD**: 端口 3 弱上拉禁止位  
 0: 对于端口 3 弱上拉使能。  
 1: 对于端口 3 弱上拉禁止

第 5 位: **P2WPUD**: 端口 2 弱上拉禁止位  
 0: 对于端口 2 弱上拉使能。  
 1: 对于端口 2 弱上拉禁止

第 4 位: **P1WPUD**: 端口 1 弱上拉禁止位  
 0: 对于端口 1 弱上拉使能。  
 1: 对于端口 1 弱上拉禁止

第 3 位: **P0WPUD**: 端口 0 弱上拉禁止位  
 0: 对于端口 0 弱上拉使能。  
 1: 对于端口 0 弱上拉禁止

2-1 位: 未用。读=00b, 写=无关

第 0 位: **SPIO0EN**: SPI 总线 I/O 使能位。  
 0: SPI I/O 在端口引脚不可用  
 1: SCK, MISO, MOSI, NSS 分别路由至 P2.0, P2.1, P2.2, P2.3 引脚。

### 14.2 一般目的端口 I/O

每一个 I/O 端口通过字节和位都可寻址的相应特殊功能寄存器访问。当向端口写时，写到 SFR 的值锁定以保持每一个引脚的输出数据值。读数据时，如果端口输入引脚配置为数字输入模式则无论 PRT n M X 设置如何都返回该引脚的逻辑电平（即使 MUX 将另外一个信号赋给该引脚，端口寄存器仍能读取其相应的端口 I/O 引脚）。一个例外是读-修改-写指令的执行。当在 SFR 端口上运行时，如果目的是 SFR 端口中的某一位则读-修改-写指令如下：ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ 和 MOV, CLR 或 SETB。对于这些指令，寄存器的值读、修改并写回到 SFR 中。

图 14.6 P0: 端口 0 寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 1111111
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0X80

位 7-0: P0.[7:0]  
 (写—输出通过 PRT0MX、PRT1MX 和 PRT2MX 寄存器出现在 I/O 引脚上)  
 0: 逻辑低输出  
 1: 逻辑高输出 (如果相应 PRT0CF.n 位=0)  
 (读—无论 PRT0MX、PRT1MX 和 PRT2MX 寄存器的设置如何)  
 0: P0.n 引脚逻辑低  
 1: P0.n 引脚逻辑高

图 14.7 PRT0CF: 端口 0 配置寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XA4

位 7-0: PRT0CF.[7:0]: P0.7- P0.0 的输出配置位 (分别)  
 0: 相应 P0.n 输出模式是开-漏  
 1: 相应 P0.n 输出模式是推挽

图 14.8 P0MODE: 端口 0 数字/模拟输入模式

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 11111111
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XF1

位 7-0: 端口 0 数字/模拟输入模式  
 0: 对应端口 0 引脚数字输入禁止 (用于模拟量, 即 ADC 或比较器)  
 1: 对应端口 0 引脚数字输入允许

图 14.9 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
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0X90

位 7-0: P1.[7:0]  
 (写—输出通过 PRT0MX、PRT1MX 和 PRT2MX 寄存器出现在 I/O 引脚上)  
 0: 逻辑低输出  
 1: 逻辑高输出 (如果相应 PRT1CF.n 位=0)  
 (读—无论 PRT0MX、PRT1MX 和 PRT2MX 寄存器的设置如何)  
 0: P1.n 引脚逻辑低  
 1: P1.n 引脚逻辑高

图 14.10 PRT1CF: 端口 1 配置寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XA5

位 7-0: PRT1CF.[7:0]: P1.7- P1.0 的输出配置位 (分别)  
 0: 相应的 P1.n 输出模式是开-漏  
 1: 相应 P1.n 输出模式是推-挽

图 14.11 P1MODE: 端口 1 数字/模拟输入模式

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 11111111
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XF2

位 7-0: 端口 0 数字/模拟输出模式  
 0: 对应端口 1 引脚数字输入禁止 (用于模拟量, 即 ADC 或比较器)  
 1: 对应端口 1 引脚数字输入允许

图 14.12 PRT1IF: 端口 1 中断标志寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
IE7	IE6	IE5	IE4	-	-	-	-	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XAD

第 7 位: IE7: 外部中断 7 悬挂标志  
 0: 在 P1.7 未检测到下跳沿  
 1: 在 P1.7 检测到下跳沿时硬件置位此标志  
 第 6 位: IE6: 外部中断 6 悬挂标志  
 0: 在 P1.6 未检测到下跳沿  
 1: 在 P1.6 检测到下跳沿时硬件置位此标志  
 第 5 位: IE5: 外部中断 5 悬挂标志  
 0: 在 P1.5 未检测到下跳沿  
 1: 在 P1.5 检测到下跳沿时硬件置位此标志  
 第 4 位: IE4: 外部中断 4 悬挂标志  
 0: 在 P1.4 未检测到下跳沿  
 1: 在 P1.4 检测到下跳沿时硬件置位此标志  
 位 3-0: 未用.读=0000b, 写=无关  
 注: 为了检测到下跳沿输入模式必须配置为数字模式。

图 14.13. P2: 端口 2 寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 1111111
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XA0 (可位寻址)

位 7-0: P2.[7:0]  
 (写—输出通过 PRT0MX、PRT1MX 和 PRT2MX 寄存器出现在 I/O 引脚上)  
 0: 逻辑低输出  
 1: 逻辑高输出 (如果相应 PRT2CF.n 位=0)  
 (读—无论 PRT0MX、PRT1MX 和 PRT2MX 寄存器的设置如何)  
 0: P2.n 引脚逻辑低  
 1: P2.n 引脚逻辑高

图 14.14 PRT2CF: 端口 2 配置寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XA6

7-0 位: PRT2CF.[7:0]: 对 P2.7-P2.0 的输出配置位 (分别的)  
 0: 相应 P2.n 输出模式是开漏  
 1: 相应 P2.n 输出模式是推挽

图 14.15 P2MODE 端口 2 数字/模拟输入模式

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 11111111
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XF3

位 7-0: 端口 0 数字/模拟输出模式  
 0: 对应端口 2 引脚数字输入禁止 (用于模拟量, 即 ADC 或比较器)  
 1: 对应端口 2 引脚数字输入允许

图 14.16. P3: 端口 3 寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 11111111
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XB0

7-0 位: P3.[7: 0]  
(写)  
0: 逻辑低输出  
1: 逻辑高输出 (如果相应 PRT3CF.n=0 则高阻抗)  
(读)  
0: P3.n 是逻辑低  
1: P3.n 是逻辑高

图 14.17. PRT3CF: 端口 3 配置寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XA7

7-0 位: PRT3CF.[7: 0]: P3.7-P3.0 的输出配置位  
0: 相应 P3.n 输出模式是开漏  
1: 相应 P3.n 输出模式是推挽

图 14.18. P3 模式: 端口 3 数字/模拟输入模式

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 11111111
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XF4

7-0 位: 端口 0 数字/模拟输出模式  
0: 端口 3 的相应引脚数字输入禁止。(用于模拟场合, 例如 ADC 或比较器)  
1: 相应端口 3 引脚数字输入允许。

\* (在 C8051F206, C8051F220/6 和 C8051F230/6 有此功能。)

表 14.1. 端口输入/输出 直流电气特性

VDD=2.7—3.6 伏，-40℃到 85℃除非特别说明

参数	条件	最小值	类型	最大值	单位
输出高电压	$I_{OH}=-10\mu A$ , 端口入/出推挽 $I_{OH}=-3mA$ , 端口入/出推挽 $I_{OH}=-10mA$ 端口入/出推挽	VDD-0.1 VDD-0.7	VDD-0.8		V
输出低电压	$I_{OL}=10\mu A$ $I_{OL}=8.5mA$ $I_{OL}=25mA$		1.0	0.1 0.6	V
输入高电压		0.7x VDD			V
输入低电压				0.3x VDD	V
输入泄露电 流	DGND<端口管脚<VDD,管脚三态弱上拉关闭 弱上拉打开		30	$\pm 1$	$\mu A$
电容装载			3		pF



### 15. 串行外设接口总线

串行外设接口提供对 4 芯，全双工串行总线的访问。SPI 支持一条总线上多个从设备到主设备的连接。一个单独的从-选择信号用来选择一个从设备并允许主从设备之间的数据传输。也支持一条总线上的多主设备。当两个或两个以上的主设备试图同时传输数据提供冲突检测。SPI 可以作为主设备或从设备运行。当 SPI 作为主设备时，最大数据传输率（位/秒）是系统时钟频率的一半。

当 SPI 配置为从设备时，如果主设备与系统时钟同步发出 SCK，NSS 和串行输入数据，全双工操作最大数据传输率（字节/秒）是系统时钟频率的 1/10。如果主设备同时发出 SCK、NSS 和串行输入数据，最大数据传输率必须小于系统时钟频率的 1/10。当出现主设备只输出数据而不需要从从设备输入的特殊情况时（例如，半双工操作），SPI 从设备可以以最大为系统时钟频率 1/4 的速率接收数据。这是在主设备与系统时钟同时发出 SCK、NSS 和输入数据的情况下。

图 15.1 SPI 方框图

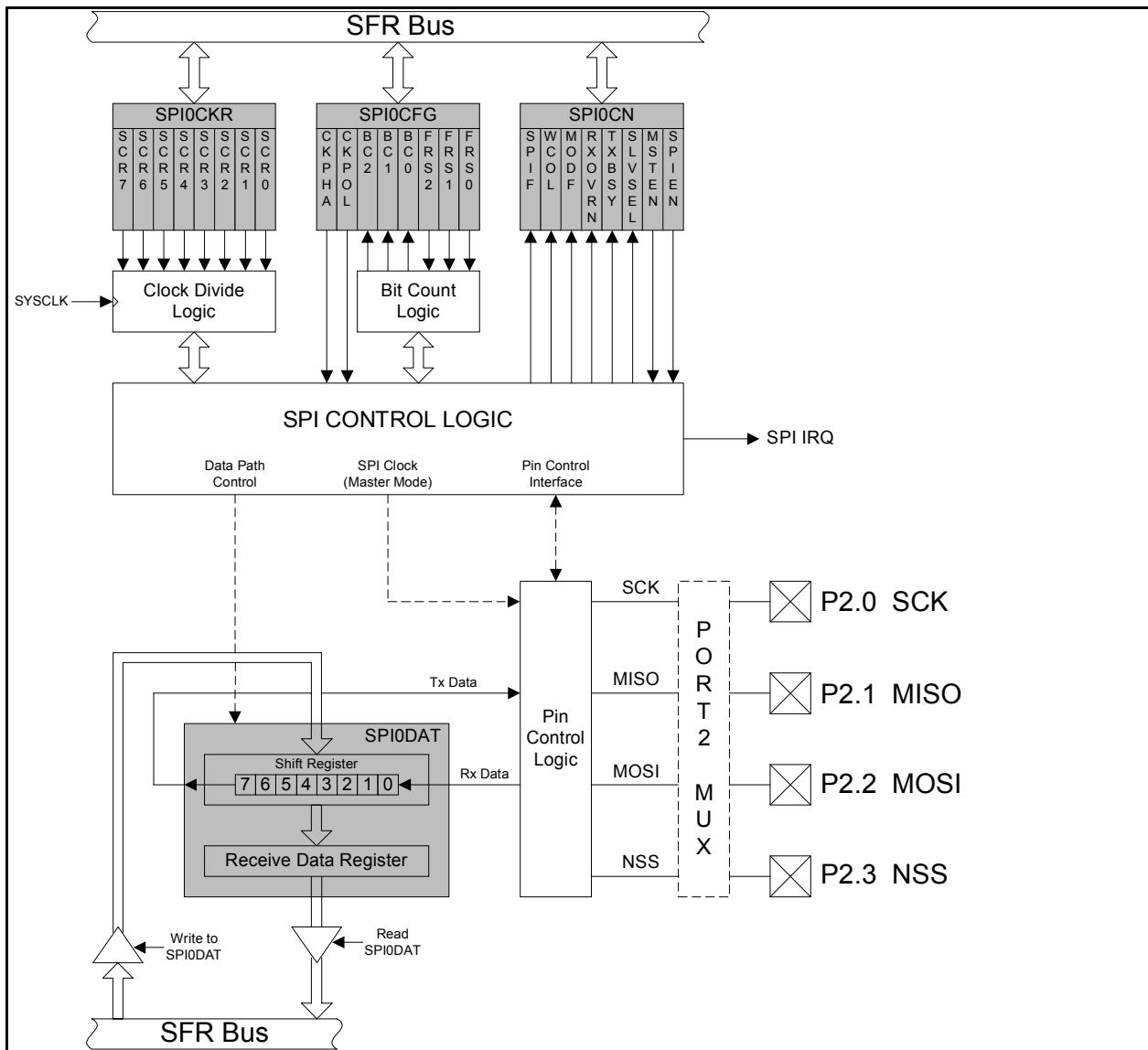
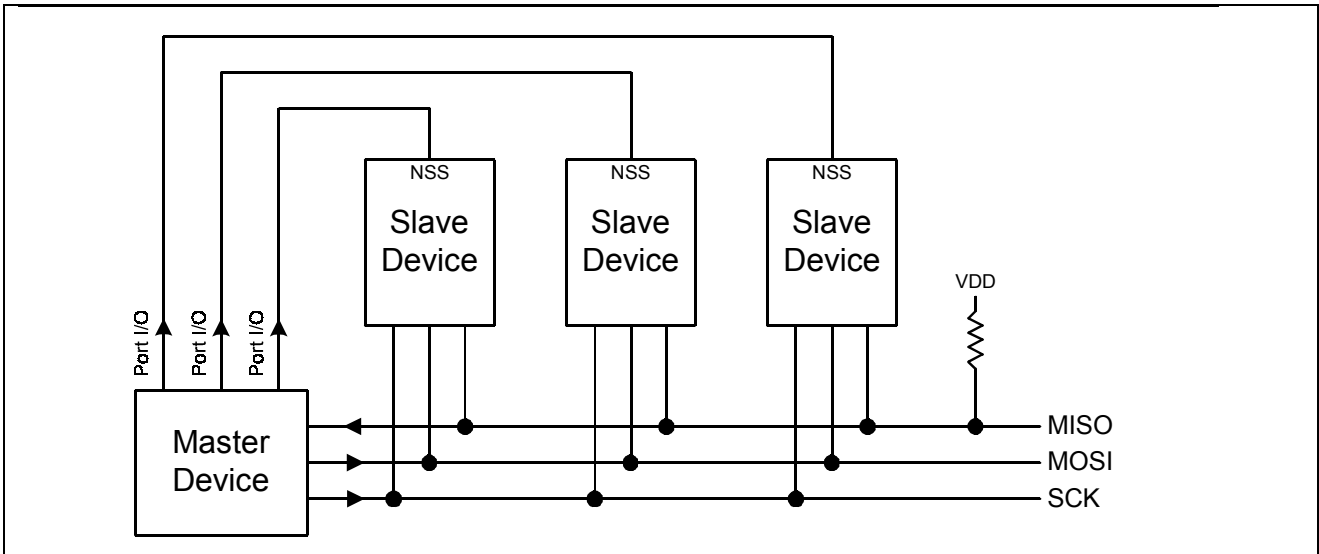


图 15.2 典型的 SPI 连接



## 15.1 信号描述

SPI 使用的 4 个信号（MOSI，MISO，SCK，NSS）说明如下。

### 15.1.1 主出从入

MOSI 信号是从主设备的输出和从设备的输入。它用来从主设备到从设备串行传输数据。数据传输时最高位先传。

### 15.1.2 主入从出

MISO 信号是从从设备的输出并输入到主设备。它用来串行地从从设备到主设备传输数据。数据传输时最高位先传。当从设备没有被选择时，SPI 从设备置 MISO 引脚于高阻状态。

### 15.1.3 串行时钟

串行时钟信号（SCK）是主设备的输出和从设备的输入。用来同步在 MOSI 和 MISO 线上主和从设备之间的数据传输。

### 15.1.4 从设备选择

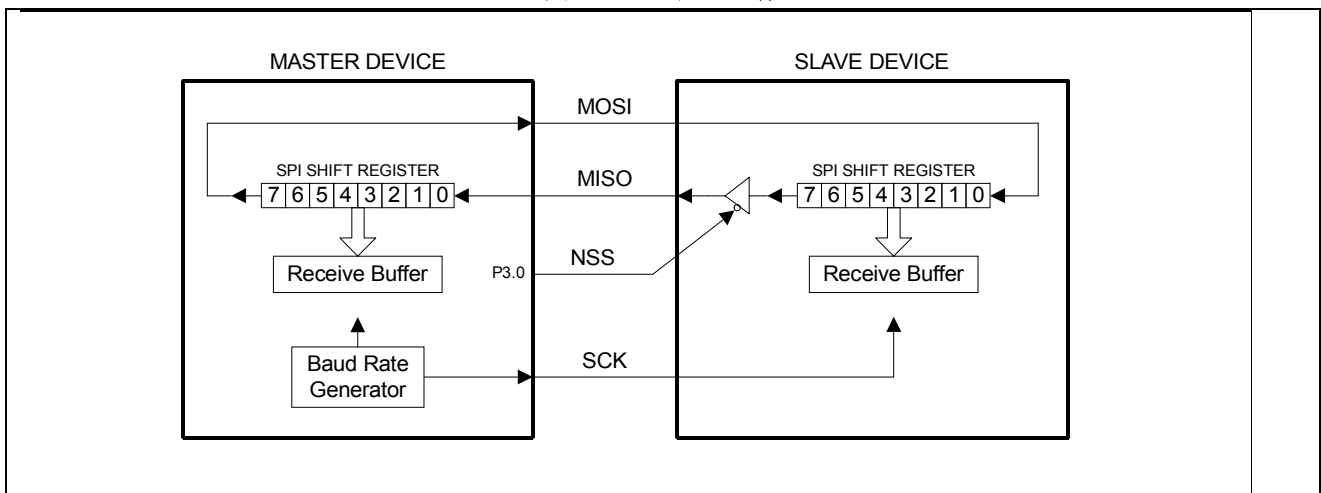
从选择信号（NSS）是处于从模式时主设备用来选择 SPI 模块或处于主模式中禁止 SPI 模块的输入信号。在从模式中时，它为低以启动数据传输并且在传输过程中一直保持为低。

## 15.2. 操作

只有 SPI 主设备才能启动数据传输。通过设定主设备使能标志 (MSTEN, SPI0CN1) 将 SPI 置于主设备模式。处于主设备模式写一字节数据到 SPI 数据寄存器时启动数据传输。当提供串行时钟 SPI 主设备立即在 MOSI 线上顺序移出数据, 同时在 SCK 上提供时钟。在传输的最后 SPIF 设定为逻辑 1。如果中断允许, 当 SPIF 标志被置位, 产生中断请求。在传输操作中为了容纳不同字长的从设备, SPI 主设备可以配置为一至八位移入/移出。在 SPI 配置寄存器 (SPI0CFG[2: 0]) 中的 SPIFRS 位用来选择在传输操作中移入/出的位数。

在全双工操作中, 当 SPI 主设备在 MOSI 线上传输数据到从设备时, 被寻址的 SPI 从设备同时将移位寄存器的内容传到 SPI 主设备。从设备收到的数据字节取代了主设备数据寄存器中的数据。所以, SPIF 标志既作为传输完成又作为接收数据准备好标志。双向数据传输由主设备产生的串行钟来同步。图 15.3 描述了 SPI 主设备和所寻址的从设备的全双工操作。

图 15.3 全双工工作



SPI 数据寄存器在读时双缓冲, 但是在写时不是。如果在数据传输时试图往 SPI0DAT 上写, WCOL 标志将被置为逻辑 1 并且写被忽略。当前数据传输会继续进行。系统控制器实际上读了接收缓冲器而不是 SPI 数据寄存器。如果接收缓冲器还存有上次未读取的数据, 一个接收溢出发生并且 RXOVRN 标志被置为逻辑 1。新数据没有被传输到接收缓冲器, 从而可以读出以前接收的数据字节。引起溢出的数据字节将丢失。

当 SPI 被激活并且不被作为主设备, 它将作为 SPI 从设备工作。另一个作为主设备的 SPI 设备通过使 NSS 信号低启动。把数据输出移位寄存器 MOSI 使用它的串行时钟。(当 NSS 信号走高时) 数据传输结束时 SPI 标志被置为逻辑 1。为了下一次数据传输, 通过写到 SPI 数据寄存器从设备可以释放移动寄存器。从设备必须在主设备开始下一次传输之前至少提前一个 SPI 串行时钟循环写数据到数据寄存器。否则, 已在从设备移位寄存器中的数据将被传输。

在一条总线上可以存在多个主设备。当 SPI 被配置为主设备 (MSTEN=1) 并且它的从设备选择信号 NSS 为低时, 一个模式错误标志 (MODF, SPI0CN5) 被置为逻辑 1。当模式错误标志被置位时, SPI 控制寄存器中的 MSTEN 和 SPIEN 位由硬件复位, 从而将 SPI 模块置于“离线”状态。在有多个主设备的环境中, 在将 MSTEN 置位和启动数据传输之前, 系统控制器应该检查 SLVSEL 标志的状态以确保总线是空闲的。

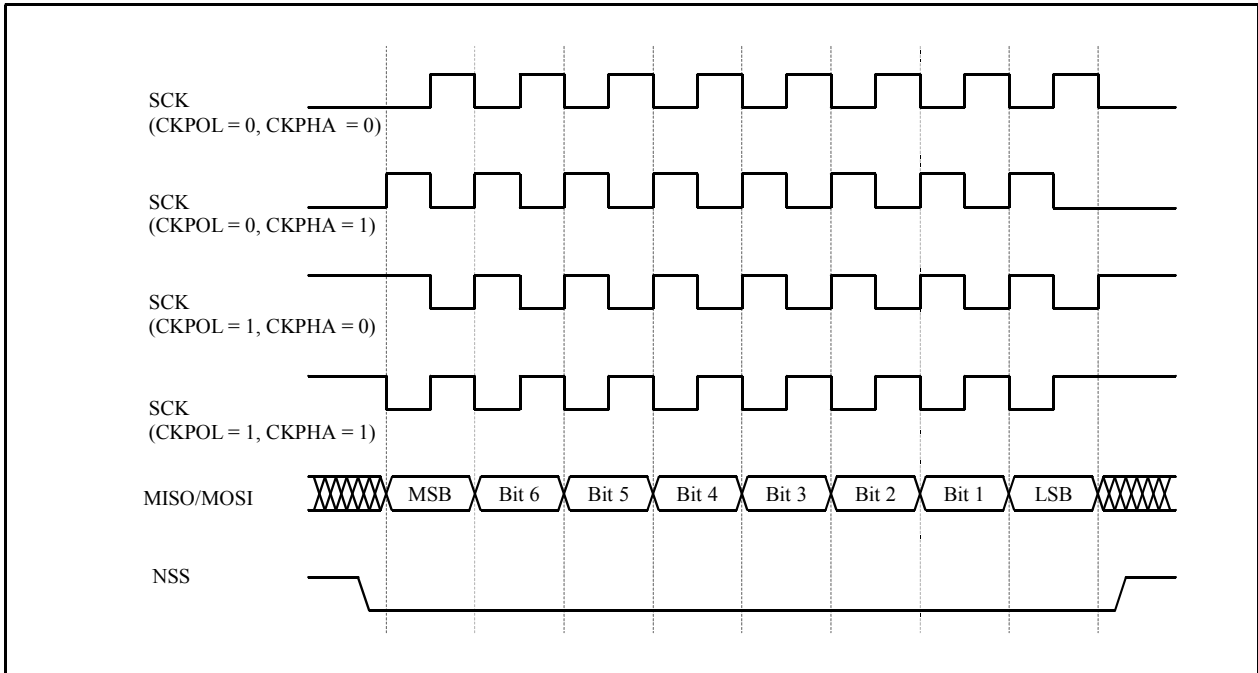
## 15.3. 串行时钟时序

如图 15.4 所示, 使用 SPI 配置寄存器 (SPI0CFG) 中的时钟控制位可以选择串行时钟相位和极性的四种组合。CKPHA 位 (SPI0CFG.7) 选择两个时钟相位之一 (用来锁住数据的边缘)。CKPOL 位 (SPI0CFG.6) 用来选择活跃-高或活跃-低时钟。必须将主设备和从设备配置为使用相同的时钟相位和极性。注: 当改变时

钟相位和极性时应该禁止 SPI (通过复位 SPIEN 位, SPI0CN.0)

如图 15.7 所示的 SPI 时钟频率寄存器 (SPI0CKR) 控制主设备模式串行时钟频率。当以从设备模式工作时这个寄存器被忽略。

图 15.4 数据/时钟时序图



### 15.4. SPI 特殊功能寄存器

在系统控制器中 SPI 通过 4 个特殊功能寄存器访问控制：SPI0CN 控制寄存器，SPI0DAT 数据寄存器，SPI0CFG 配置寄存器和 SPI0CKR 时钟频率寄存器。有关 SPI 总线操作的 4 个特殊功能寄存器如下面所述。

图 15.5 SPI0CFG: SPI 配置寄存器

R/W	R/W	R	R	R	R/W	R/W	R/W	
CKPHA	CKPOL	BC2	BC1	BC0	SPIFRS2	SPIFRS1	SPIFRS0	复位值：00000111B
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址：0x9A

7 位：CKPHA：SPI 时钟相位。  
此位控制 SPI 时钟相位。  
0：在 SCK 周期第一个边缘采集的数据  
1：在 SCK 周期第二个边缘采集的数据

6 位：CKPOL：SPI 时钟极性。  
这一节控制 SPI 时钟极性。  
0：SCK 线为低，处于空闲状态  
1：SCK 线为高，处于空闲状态

5-3 位：BC2-BC0：SPI 位计数  
指示 SPI 字 8 个位中的哪几位被传输。

BC2-BC0			传输位
0	0	0	0 位（最低位）
0	0	1	1 位
0	1	0	2 位
0	1	1	3 位
1	0	0	4 位
1	0	1	5 位
1	1	0	6 位
1	1	1	7 位（最高位）

2-0 位：SPIFRS2-SPIFRS0：SPI 帧尺寸。  
这三位决定在主设备模式中移入/移出 SPI 移位寄存器的位数。在从模式中其被忽略。

SPIFRS			移动的位
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

图 15.6 SPI0CN: SPI 控制寄存器

R/W	R/W	R/W	R/W	R	R	R/W	R/W	复位值: 00000000
SPIF	WCOL	MODF	RXOVRN	TXBSY	SLVSEL	MSTEN	SPIEN	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XF8
<p>7 位: SPIF: SPI 中断标志 在数据传输结束后, 此位被硬件置为逻辑 1。如果允许中断, 置这 1 位使 CPU 执行 SPI0 中断服务程序。这 1 位不会被硬件自动清除, 它必须由软件清除。</p> <p>6 位: WCOL: 写冲突标志 当数据传输正在进行时, 此位被硬件置为逻辑 1 (并且产生一个 SPI 中断) 以指示正在尝试进行写到 SPI 数据寄存器。此位由软件清除。</p> <p>5 位: MODF: 模式错误标志 当检测到主设备模式冲突时, 此位被硬件置为逻辑 1 (NSS 低和 MSTEN=1) 此位不会被硬件自动清除, 它必须被软件清除。</p> <p>4 位: RXOVRN: 接收溢出标志 当接收缓冲器中还存有上一次传输的未被读取的数据并且当前传输的最后一位移进了 SPI 移位寄存器时此位被硬件置为逻辑 1 (产生一个 SPI 中断)。它不会被硬件自动清除, 必须被软件清除。</p> <p>3 位: TXBSY: 传输忙标志 当主模式数据传输正在进行时此位被硬件置为逻辑 1。在传输结束时由硬件清除。</p> <p>2 位: SLVSEL: 从选择标志 当 NSS 引脚为低时此位被置为逻辑 1 指示它允许为从。当 NSS 高时它被清除到逻辑 0 (从禁止)</p> <p>1 位: MSTEN: 主模式允许 0: 主设备模式禁止。工作于从模式 1: 主设备模式允许。工作于主模式</p> <p>0 位: SPIEN: SPI 使能 此位允许/禁止 SPI 0: SPI 禁止 1: SPI 允许</p>								

图 15.7 SPI0CKR: SPI 时钟频率寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0X9D

7-0 位: SCR7-SCR0: SPI 时钟频率  
 当 SPI 模块配置为主设备模式时这些位决定 SCK 输出频率。SCK 时钟频率是由系统时钟分频时产生，以如下等式给出：  
 $f_{SCK} = 0.5 * f_{SYSCLK} / (SPI0CKR + 1)$        $0 \leq SPI0CKR \leq 255$

图 15.8 SPI0DAT: SPI 数据寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
-	-	-	-	-	-	-	-	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0X9B

位 7-0: SPI0DAT: SPI0 发送和接收数据  
 SPI0DAT 寄存器用来发送和接收 SPI 数据。当在主设备模式中时，向 SPI0DAT 写数据使数据立刻进入到移位寄存器并且启动数据传输。读 SPI0DAT 返回接收缓冲区的内容。

## 16. UART

### 描述

CIP-51 包括一能够异步传输的串行端口 (UART)。URAT 能以全双工模式工作。在所有模式中, 接收数据在保存寄存器中缓冲。这使 URAT 有可能在软件读完前一个数据字节之前, 就接收下一个数据字节。UART 在 SBF 中有一个相关的串行控制寄存器 (SCON) 和一个串行数据缓冲器 (SBUF)。SBUF 对发送和接收寄存器都可访问。读自动访问接收寄存器, 写自动访问发送寄存器。

如果处于使能状态, UART 能够产生中断。UART 有两种中断源, 发送中断标志 TI (SCON.1) 当一个数据字节的发送完成时置位, 和一个接收中断标志, RI (SCON.0) 当一个数据字节的接收完成时置位。当 CPU 进入中断服务程序时 UART 中断标志不被硬件清除。它们必须通过软件手工清除。这使得可以通过软件得到 UART 中断的原因 (发送结束或接收结束)。

### 16.1. UART 工作方式

通过在 SCON 寄存器设置配置位, UART 提供了 4 种工作方式 (一种同步和三种异步) 选择。这 4 种方式提供不同的波特率和通讯方式。这 4 种方式总结如下表 16.1, 后面是详尽的描述。

图 16.1. UART 方块图

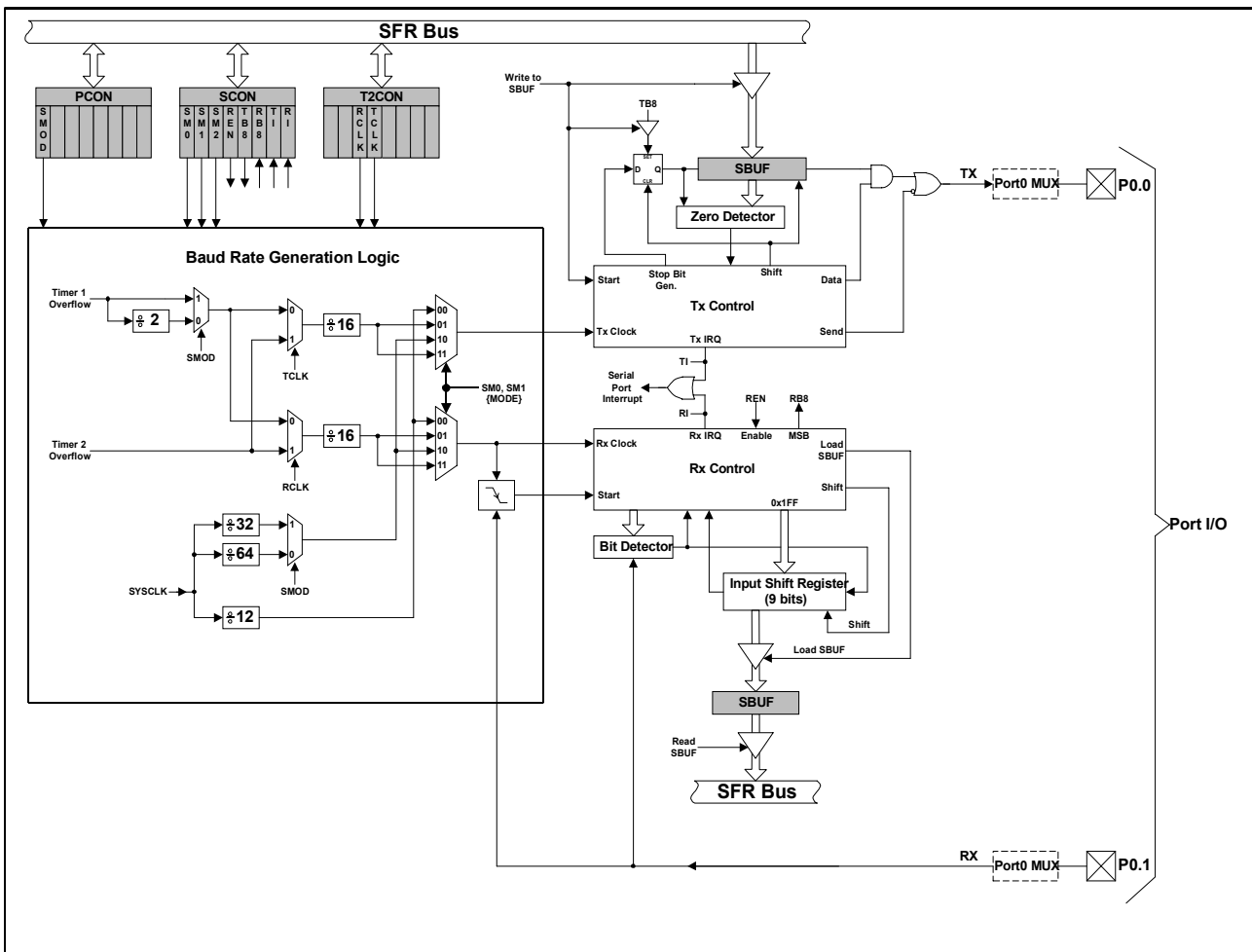




表 16.1 UART 方式

模式	同步	波特 时钟	数据位	开始/停止位
0	同步	系统时钟/12	8	没有
1	异步	定时器 1 或定时器 2 溢出	8	1 个开始位, 1 个停止位
2	异步	系统时钟/32 或系统时钟/64	9	1 个开始位, 1 个停止位
3	异步	定时器 1 或定时器 2 溢出	9	1 个开始位, 1 个停止位

16.1.1 方式 0: 同步方式

方式 0 提供同步, 半双工通讯。在 RX 引脚上发送和接收串行数据。TX 引脚提供发送和接收的移位时钟。MCU 必须为主设备因为在双向传输的移位时钟都是它产生的 (参见连接图 16.2)

共发送/接收 8 个数据位, 最低位优先 (参见 16.3 定时图)。当指令向 SBUF 寄存器写 1 个数据字节时数据发送开始。TI 发送中断标志 (SCON.1) 在第 8 位时间的末尾被置位。当接收使能位 REN 置为逻辑 1 并且接收中断标志 (SCON.0) 被清除时数据接收开始。 八位移入的一个周期后, RI 标志被置位并且接收停止直到软件清除了 RI 位。当 TI 或 RI 置位时, 如果允许中断将产生中断。

方式 0 波特率是系统时钟频率除以 12。

图 16.2 UART 方式 0 连接

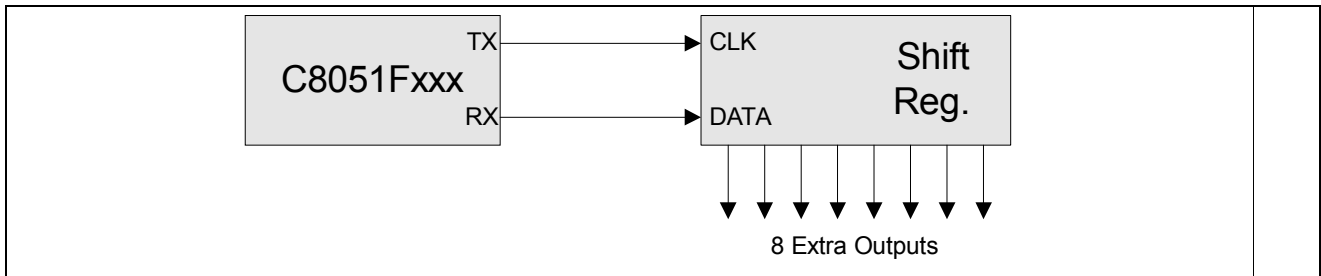
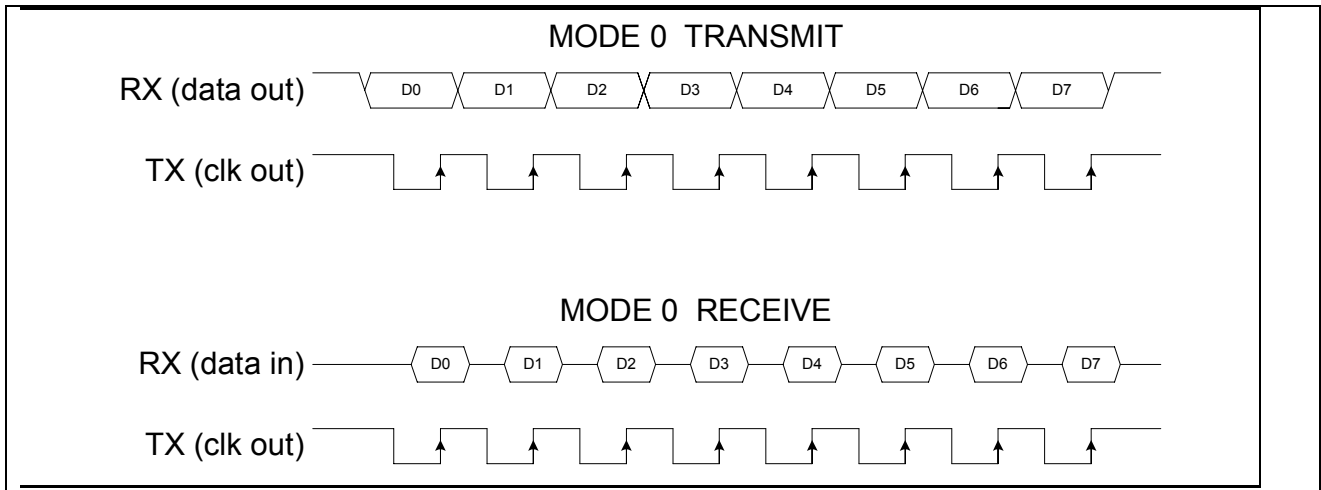


图 16.3 UART 方式 0 时序



方式 1 提供标准的异步全双工通讯，使用每数据字节共 10 位：一个起始位，八个数据位，（最低位优先）和一个停止位。数据从 TX 引脚发送在 RX 引脚接收。接收时 8 个数据位储存在 SBUF 而停止位存入 RB8（SCON.2）。

当向 SBUF 寄存器写数据字节时，数据发送开始。TI（发送中断）标志（SCON.1）在传输结束时被置位（停止位时间的开始）。在接收使能位 REN（SCON.4）被置为逻辑 1 后的任何时间数据接收都可以开始。在收到停止位后，如果满足以下条件：RI 是逻辑 0，并且如果 SM2 是逻辑 1，停止位必须是逻辑 1，则数据字节将被装入 SBUF 接收寄存器。

如果这些条件满足，8 个数据位被储存在 SBUF 中，停止位储存在 RB8，并且 RI 标志被置位。如果这些条件不满足，SBUF 和 RB8 将不会被装载并且 RI 标志不会被置位。当 TI 或 RI 置位时，如果允许中断将产生中断。

方式 1 中产生的波特率是定时器溢出的函数。UART 可以使用定时器 1 或者定时器 2 以自动重装方式来产生波特率。每发生一次溢出事件（所有的 1 都变为 0，对定时器 1 是 0xFF，对定时器 2 是 0xFFFF）1 个时钟都会被送到波特率电路。这个时钟被 16 除来产生波特率。

定时器 1 应当配置为对于 8 位计数器/定时器为自动重载模式，且其作为波特率发生器时中断应禁止。储存在 TH1 中的系统时钟频率和重装值的组合如下确定波特率：

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

SMOD 位（PCON.7）选择是否把定时器 1 溢出率被 2 除。清除后，SMOD 位是逻辑 0，这样就缺省选择了低波特率。选择定时器 1 使用的时基允许进一步控制产生波特率。系统时钟被 1 除（在 CKCON 里置位 TIM）使上面公式分母中的 12 变为 1。

为使用定时器 2 产生波特率，配置定时器的波特率发生器方式并将 RCLK 和/或 TCLK 置为逻辑 1。将 RCLK 和/或 TCLK 置位自动禁止了定时器 2 的中断并且定时器 2 使用系统时钟被 2 除作为它的时基。如果需要不同的时基，将 C/T2 位置为逻辑 1 将在外部输入脚 T2 提供的外部时钟作为时基。时钟频率和储存在捕捉寄存器中的重装值共同决定了波特率：

$$\text{方式 1 波特率} = \text{系统时钟} / [32 * (65536 - [\text{RCAP2H} : \text{RCAP2L}])] ]$$

[RCAP2H: RCAP2L]是存在捕捉寄存器中的 16 位值.

**图 16.4 UART 方式 1 时序图**

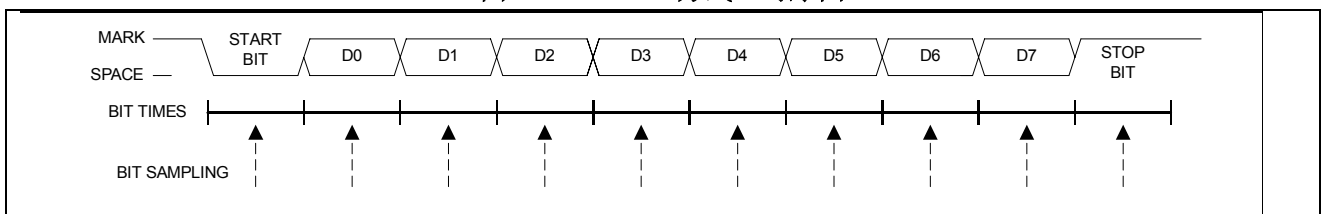
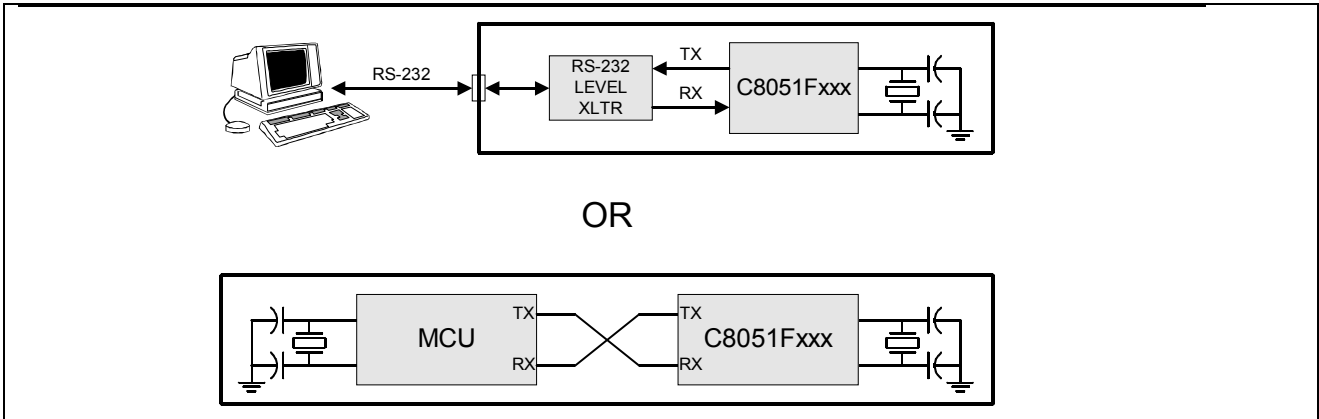


图 16.5 UART 方式 1, 2, 和 3 接口图



### 16.1.3. 方式 2: 9 位 UART, 固定波特率

方式 2 提供了异步, 全双工通讯每个数据字节总共使用了 11 位: 1 个开始位, 8 个数据位 (最低位优先), 1 个可编程的第 9 个数据位和 1 个停止位。传输时, 第 9 数据位被 TB8 (SCON.3) 中的值所决定。可以赋给它 PSW 中的奇偶标志 P 的值或者在多处理器通讯中使用它。接收时, 第 9 数据位存入 RB8 (SCON2) 并且停止位被忽略。

当指令写 1 个数据字节到 SBUF 寄存器时数据传输开始。TI 发送中断标志 (SCON1) 在传输结束时被置位 (停止位时间的开始)。数据接收可以在 REN (SCON4) 被置为逻辑 1 以后的任何时间开始。接到停止位之后, 如果满足以下条件, 数据字节将被装载入 SBUF 接收寄存器: RI 必须是逻辑 0, 并且如果 SM2 是逻辑 1, 第 9 位必须是逻辑 1。

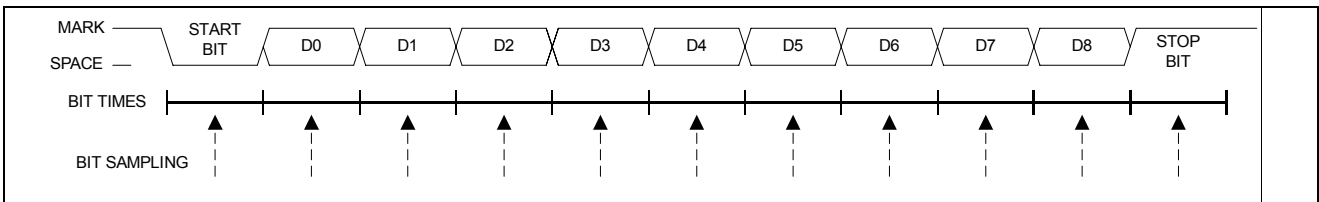
如果满足这些条件, 数据的 8 个字节储存在 SBUF 中, 第 9 个字节储存在 RB8 并且 RI 标志被置位。如果这些条件不满足, SBUF 和 RB8 将不会被装载并且 RI 标志不会被置位。当 TI 或 RI 被置位时, 如果允许中断将产生中断。

方式 2 中的波特率是系统时钟频率的直接函数:

$$\text{方式 2 波特率} = 2^{\text{SMOD}} * (\text{SYSCLK} / 64)$$

SMOD 位 (PCON7) 选择以 32 还是 64 来除 SYSCLK。在公式中, 2 是底数, SMOD 是指数, 使波特率是系统时钟频率的 1/32 或 1/64。清除时, SMOD 位是逻辑 0, 因而缺省选择低波特率。

图 16.6. UART 方式 2 和 3 时序图



### 16.1.4 方式 3: 9 位 UART, 变化的波特率

除了波特率是变化的之外, 方式 3 在各个方面与方式 2 都是一样的。波特率的确定方式与方式 1 相同。方式 3 传输 11 位: 1 个开始位, 8 个数据位 (最低位优先), 1 个可编程的第 9 数据位, 和 1 个停止位。定时器 1 和定时器 2 溢出所产生的波特率与方式 1 完全一样。总之, 方式 3 传输与方式 2 使用同样的协议, 但是使用方式 1 产生波特率。

## 16.2 多处理器通讯

方式 2 和 3 支持在 1 个主处理器和 1 个或多个从处理器之间的通过特殊使用第 9 个数据位的通讯。当 1 个主处理器欲向 1 个或多个从处理器传输数据时，它首先送出 1 个地址字节来选择目标。1 个地址字节与 1 个数据字节不同的是，它第 9 位是逻辑 1 而数据字节第 9 位总是设为逻辑 0。

将从处理器的 SM2 位置为 1 (SCON5) 将使得它的 UART 在接收到 1 个停止位时，只有第 9 位是逻辑 1 时 UART 才产生中断 (RB8=1) 以表明 1 个地址字节被收到。在 UART 的中断处理程序中，软件将所接收到的地址和从设备自己分配的 8 位地址进行比较。如果地址相符，从设备将清除它的 SM2 位使余下数据字节的接收能产生中断。没有被寻址的从设备其 SM2 位置位并且对余下数据字节的接收不产生中断，因而会将此数据忽略。一旦接收到完整的消息，被寻址的从设备将它的 SM2 位复位从而忽略所有的数据传输，直到它收到下一个地址字节为止。

对于一个从设备可以分配多个地址和/或 1 个地址可以分配给多个从设备，这样就可以将数据传输同时“广播”到不止 1 个从设备。可以将主处理器配置为可以接收所有的数据传输，或者实施一个协议，使得主/从设备的关系可以暂时地互换从而使原来的主设备/从设备之间的半-双工传输成为可能。

图 16.7 UART 多处理器方式接点图

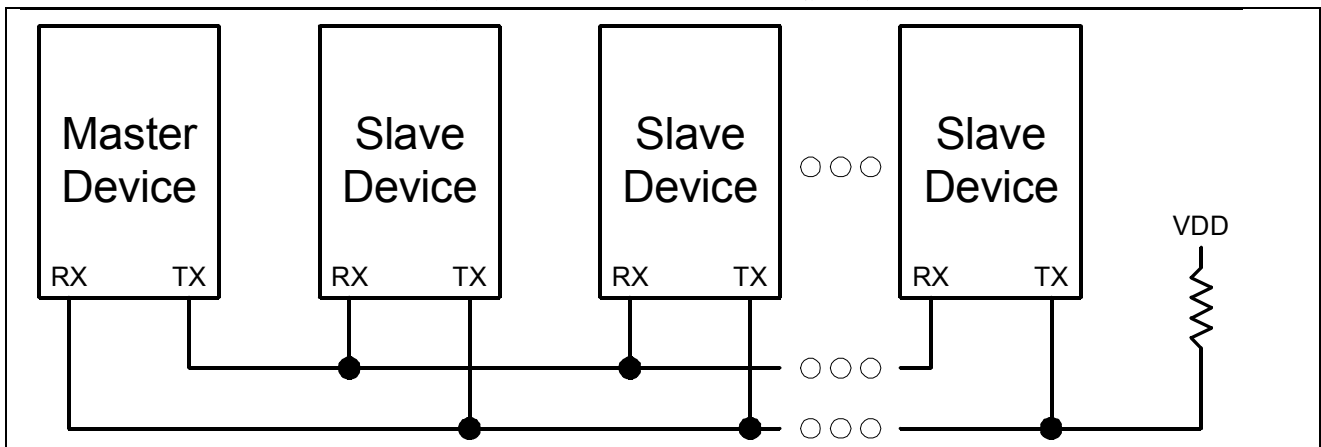


表 16.2 标准波特率的振荡器频率

振荡器频率	除因子	定时器 1 装载值*	得到的波特率**
24.0	208	0xF3	115200(115384)
23.592	205	0xF3	115200(13423)
22.1184	192	0xF4	115200
18.432	160	0xF6	115200
16.5888	144	0xF7	115200
14.7456	128	0xF8	115200
12.9024	112	0xF9	115200
11.0592	96	0xFA	115200
9.216	80	0xFB	115200
7.3728	64	0xFC	115200
5.5296	48	0xFD	115200
3.6864	32	0xFE	115200
1.8432	16	0xFF	115200
24.576	320	0xEC	76800
25.0	432	0xE5	57600 (57870)
25.0	868	0xCA	28800
24.576	848	0xCB	28800 (28921)
24.0	833	0xCC	28800(28846)
23.592	819	0xCD	28800(28911)
22.1184	768	0xD0	28800
18.432	640	0xD8	28800
16.5888	576	0xDC	28800
14.7456	512	0xE0	28800
12.9024	448	0xE4	28800
11.0592	348	0xE8	28800
9.216	320	0xEC	28800
7.3728	256	0xF0	28800
5.5296	192	0xF4	28800
3.6864	128	0xF8	28800
1.8432	64	0xFC	28800

\*假设 SMOD=1, TIM=1

\*\*括号内的数表示实际的波特率

图 16.8. SBUF: 串行 (UART) 数据缓冲寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0X99

7-0 位: SBUF: 串行数据缓冲位 7-0 (最高位-最低位)

这实际上是 2 个寄存器; 1 个发送缓冲寄存器, 1 个接收缓冲寄存器。当数据移动到 SBUF 时, 它进入到发送缓冲器等待串行传输。将 1 个字节发给 SBUF 启动数据传输。当数据从 SBUF 移出时, 它来自于接收缓冲器。

图 16.9. SCON 串行端口控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0X98

6-7 位: SM0-SM1: 串行端口操作方式  
这些位选择串行端口操作方式

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

5 位: SM2: 多处理器通讯使能  
此位的功能依赖于串行端口操作方式位  
方式 0: 无效  
方式 1: 检查合法停止位  
0: 停止位的逻辑级被忽略  
1: 只有在停止位是逻辑级 1 时, RI 才能被激活  
方式 2 和 3: 多处理器通讯  
0: 第 9 位的逻辑级被忽略  
1: RI 被置位只有当第 9 位是逻辑 1 时才产生中断

4 位: REN: 接收使能  
此位允许/禁止 UART 接收器  
0: UART 接收禁止  
1: UART 接收使能

3 位: TB8: 第 9 传输位  
在方式 2 和 3 中此位的逻辑级将被赋到第 9 传输位。它不在方式 0 和 1 中使用。根据需要由软件置位或复位

2 位: RB8: 第 9 接收位  
此位被赋予方式 2, 3 中收到的第 9 位的逻辑级。在方式 1 中, 如果 SM2 是逻辑 0, RB8 被赋予接收到的停止位的逻辑级。在方式 0 中 RB8 没有被使用。

1 位: TI: 发送中断标志  
当 1 字节数据已经被 UART 传输后由硬件置位, (在方式 0 中时, 在第 8 位之后, 在别的方式中时在停止位的开始)。当 UART 中断使能时, 将此位置位使 CPU 进入 UART 中断服务程序。此位必须由软件手工清除

0 位: RI: 接收中断标志  
当 1 字节数据已经被 UART 接收后由硬件置位 (在方式 0 中时, 在第 8 位之后, 在别的方式中时在停止位之后, SM2 位是例外)。当 UART 允许中断时, 将此位置位使 CPU 指到 UART 中断服务程序。此位必须由软件手工清除

## 17. 定时器

CIP-51 包括 3 个与在标准 8051MCU 中使用类似的 16 位计数器/定时器。它们可用来测量时间间隔，对外部事件计数和产生周期性的中断请求。定时器 0 和 1 非常类似并且有 4 种主要工作方式。定时器 2 提供定时器 1, 0 没有的附加功能，例如捕捉和产生波特率。

定时器 0 和 1	定时器 2
13 位计数器/定时器	带自动重装的 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
带自动重装的 8 位计数器/定时器	波特率发生器
2 个 8 位计数器/定时器（只是定时器 0）	

当作为定时器，计数器/定时器寄存器在每一个时钟脉冲下都增长。时钟敲系统时钟被 1 或 12 除如在 CKCON 中被定时器时钟选择位 (T2M-T0M) 所强调的那样。12 钟每下选项提供了和 8051 家族老一代兼容。如果需要 1 个更快的定时器可以使用每下 1 个时钟选项。

当作为计数器时，计数器/定时器寄存器在选择的输入引脚 (P04/T0, P05/T1, P06/T2) 每 1 次从高到低传输都增长 1。大到系统时钟频率 1/4 的事件能被数。输入信号不需要是阶段性的，但是它应该被保持在给定的电平至少 2 个整的系统时钟循环来确保这个级是取样。???

### 17.1. 定时器 0 和定时器 1

定时器 0 和定时器 1 通过 SFR 进行访问和控制。每个计数器/定时器都是 16 位寄存器，作为 2 个分开的字节进行访问：1 个低字节 (TL0 或 TL1) 和 1 个高字节 (TH0 或 TH1)。计数器/定时器控制寄存器用来给出定时器 0 和定时器 1 的使能信号以及指示其状态。定时器/计数器都工作在通过设置在定时器/计数器方式寄存器中的方式选择位 M1-M0 所确定的四种主要方式之一。每个定时器都可以独立进行配置。以下是每种工作方式的详尽描述。

#### 17.1.1. 方式 0: 13 位计数器/定时器

在方式 0 中定时器 0 和定时器 1 作为 13 位计数器/定时器工作。以下描述了定时器 0 的配置和操作。然而，2 个定时器操作完全一致并且定时器 1 的配置与定时器 0 的相同。

TH0 寄存器存放 13 位计数器/定时器的 8 个最高位。TL0 存放 TL0.4-TL0.1 的 5 个最低位。TL0 (TL07-TL05) 的高 3 位不用，在读取时应该屏蔽掉或忽略。当 13 位定时器寄存器递增并且从 0x1FFF 溢出而变为 0x0000 时，定时器溢出标志 TF0 (TCON5) 被置位，并且如果允许中断，将产生中断。

C/T0 位 (TMOD2) 选择计数器/定时器的时钟源。将 C/T 复位选择系统时钟作为定时器的输入。当 C/T0 被置为逻辑 1 时，所选定输入引脚的从高到低的跳变使定时器寄存器递增。(关于外部 I/O 引脚的选择和配置请参看 14 节)。



当 GATE0 (TMOD3) 是 0 或者输入信号 /INT0 是逻辑 1 时, 置 TR0 位 (TCON4) 激活定时器。置 GATE0 为逻辑 1 允许外部输入信号控制定时器 /INT0, 有助于脉冲宽度测量。

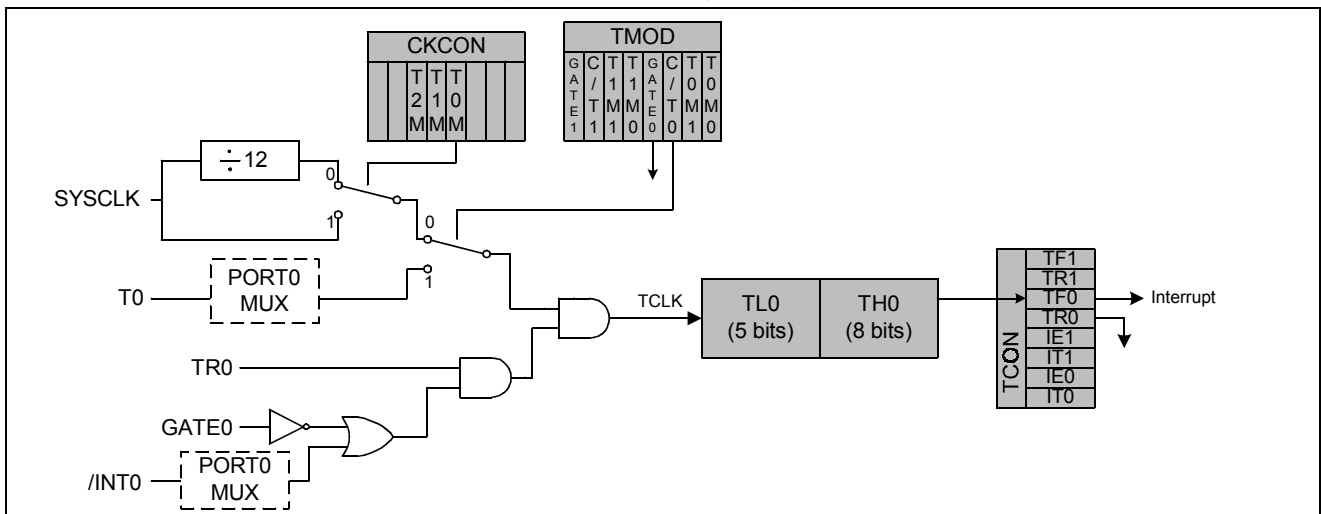
TR0	GATE0	/INT0	计数器/定时器
0	X	X	禁止
1	0	X	使能
1	1	0	禁止
1	1	1	使能

X=无关

置 TR0 并不会复位定时器寄存器。在激活定时器之前应该将定时器寄存器初始化为希望的值。

定时器 1 的 TL1 和 TH1 组成了 13 位寄存器, 与 TL0 和 TH0 相同。定时器 1 使用有关的 TCON 和 TMOD 位进行配置和控制, 这与定时器 0 是一样的。

图 17.1 T0 方式 0 方块图



### 17.1.2 方式 1: 16 位计数器/定时器

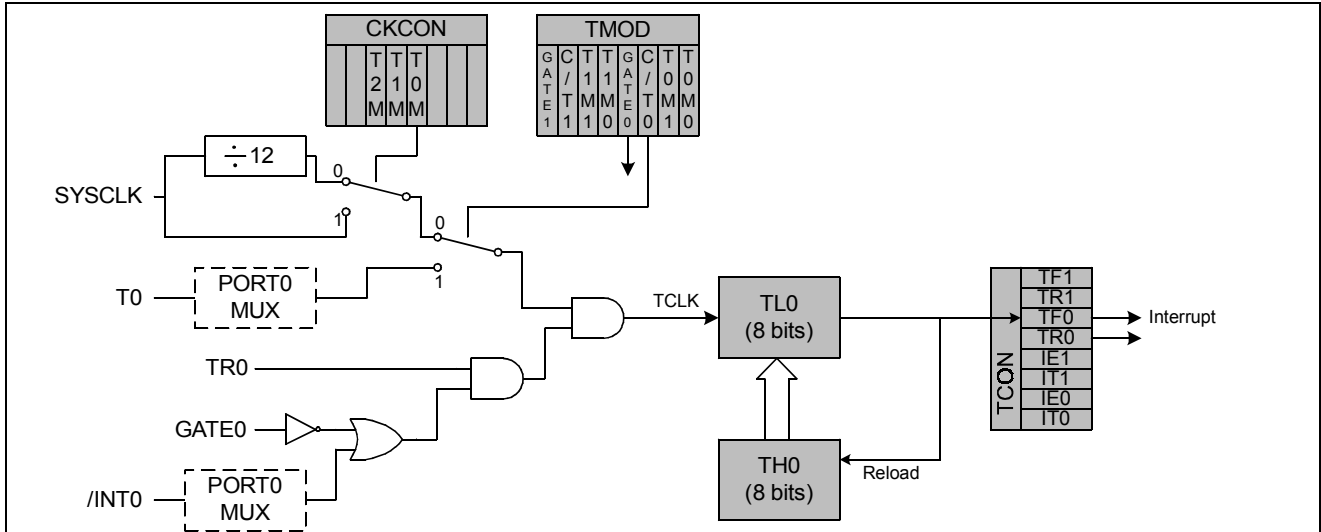
除了计数器/定时器寄存器使用所有的 16 位之外, 方式 1 与方式 0 完全一样。在方式 1 中计数器/定时器以与方式 0 同样的方式被激活和配置。



### 17.1.3 方式 2：带自动重装的 8-位计数器/定时器

方式 2 将定时器 0 和定时器 1 配置为自动重装初始值的 8 位计数器/定时器。TL0 保存计数值，TH0 保存重装值。当 TL0 中的计数器溢出，从全 1 变为 0x00 时，定时器溢出标志 TF0 (TCON5) 被置位并且 TL0 中的计数器从 TH0 中重装。如果允许中断，当 TF0 标志置位时将产生中断。TH0 中的重装值不变。启动定时器之前，TL0 必须被初始化到设定值。这样第 1 次计数才是正确的。在方式 2 中定时器 1 和定时器 0 的工作方式完全相同。在方式 2 中计数器和定时器以与方式 0 同样的方式被启动和配置。

图 17.2. T0 方式 2 方块图



### 17.1.4 方式 3：2 个 8 位计数器/定时器（仅定时器 0）

在方式 3 中定时器 0 和定时器 1 功能不同。定时器 0 被配置为 2 个独立的 8 位计数器/定时器，存于 TL0 和 TH0 中。使用在 TCON 和 TMOD 中的定时器 0 的控制/状态位：TR0, C/T0, GATE0 和 TF0 来控制 TL0 中的计数器/定时器。它可以使用系统时钟或者外部输入信号作为它的时基。TH0 寄存器必须使用系统时钟的函数作为时间源。通过定时器 1 的控制位 TR1 来允许 TH0。TH0 在溢出时置位定时器 1 溢出标志 TF1 这样就控制了定时器 1 的中断。

在方式 3 中，定时器 1 是禁止的，定时器 0 在方式 3 中也是这样，可以通过将定时器 1 切入或切出方式来开启或关闭它。定时器 1 可以在方式 0, 1 或 2 中工作，但是不能接收外部时钟信号 也不能置 TF1 标志或产生中断。然而，定时器 1 的溢出可以用来产生波特率。参看 16 节（UART）关于定时器 1 产生波特率的配置。

图 17.3 T0 方式 3 方块图

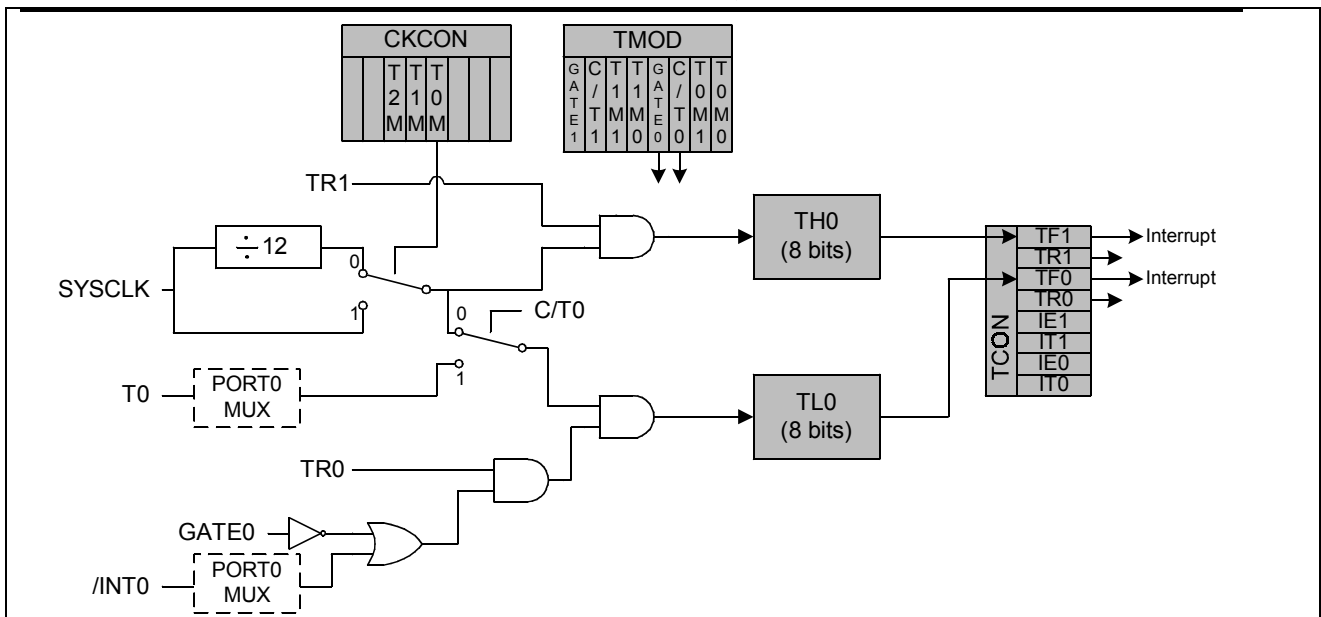


图 17.4 TCON: 定时器控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值: 00000000
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0X88

7 位: TF1: 定时器 1 溢出标志。  
当定时器 1 溢出时, 由硬件置位。这个标志能被软件复位, 当 CPU 进入定时器 1 的中断服务程序时它被自动复位。  
0: 没有发现定时器 1 溢出。  
1: 定时器 1 已经溢出

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

5 位: TF0: 定时器 0 溢出标志  
当定时器 0 溢出时, 由硬件置位。这个标志能被软件复位, 当 CPU 进入定时器 0 的中断服务程序时它被自动复位。  
0: 没有发现定时器 0 溢出。  
1: 定时器 0 已经溢出。

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

3 位: IE1: 外部中断 1  
当检测到 IT1 所定义类型的边缘/电平时这个标志由硬件置位。它能被软件复位但是当 IT1=1 时如果 CPU 进入外部中断 1 服务程序则自动复位。当 IT 1=0 时这个标志是/INT1 输入信号的逻辑反。

2 位: IT1: 中断 1 类型选择  
此位选择/INT1 信号是否检测下跳沿或者活跃低级敏感中断。  
0: /INT1 是电平触发。  
1: /INT1 是边缘触发。

1 位: IE0: 外部中断 0  
当检测到 IT0 所定义类型的边缘/电平时, 这个标志由硬件置位。它能被软件复位但是当 IT0=1 时如果 CPU 进入外部中断 1 服务程序则自动复位。当 IT0=0 时这个标志是/INT0 输入信号的逻辑反

0 位: IT0: 中断 0 类型选择  
此位选择/INT0 信号是否检测落下跳沿或者活跃低级敏感中断。  
0: /INT0 是电平触发。  
1: /INT0 是边缘触发。

**图 17.5 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: 00000000
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0X89

7 位: 门 1: 定时器 1 门控制  
 0: 当 TR1=1 时, 无论/INT1 如何, 定时器 1 允许  
 1: 只有在 TR1=1 并且/INT1=逻辑 1 时定时器 1 允许。

6 位: C/T1: 计数器/定时器 1 选择。  
 0: 定时器功能: T1M 位 (CKCON.4) 定义的时钟使定时器 1 递增  
 1: 计数器功能: 在外部输入引脚 P0.5/T1 上的从高到低的跳变使定时器 1 递增

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

T1M1	T1M0	模式
0	0	模式 0: 13 位计数器/定时器
0	1	模式 1: 16 位计数器/定时器
1	0	模式 2: 带自动重装的 8 位计数器/定时器
1	1	模式 3: 定时器 1 非工作/停止状态

3 位: 门 0: 定时器 0 门控制  
 0: 当 TR0=1 时, 无论/INT0 如何, 定时器 0 允许  
 1: 只有在 TR0=1 并且/INT0=逻辑 1 时定时器 0 允许。

2 位: C/T0: 计数器/定时器选择。  
 0: 定时器功能: T0M 位 (CKCON.3) 定义的时钟使定时器 0 递增  
 1: 计数器功能: 在外部输入引脚 P0.4/T0 上的从高到低的跳变使定时器 0 递增

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

T0M1	T0M0	模式
0	0	模式 0: 13 位计数器/定时器
0	1	模式 1: 16 位计数器/定时器
1	0	模式 2: 带自动重装的 8 位计数器/定时器
1	1	模式 3: 2 个 8 位计数器/定时器

图 17.6 CKCON: 时钟控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
-	-	T2M	T1M	T0M	-	-	-	复位值: 00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0X8E

7-6 位: 未用。读=00b, 写=无关

5 位: T2M: 定时器 2 时钟选择  
此位控制定时器 2 的系统时钟的分频。当定时器是波特率发生方式或计数器方式 (即 C/T2=1) 此位被忽略  
0: 定时器 2 使用系统时钟被 12 除  
1: 定时器 2 使用系统时钟。

4 位: T1M: 定时器 1 时钟选择  
此位控制定时器 1 的系统时钟的分频。  
0: 定时器 1 使用系统时钟被 12 除  
1: 定时器 1 使用系统时钟。

3 位: T0M: 定时器 0 时钟选择  
此位控制计数器/定时器 0 的系统时钟的分频。  
0: 计数器/定时器使用系统时钟被 12 除  
1: 计数器/定时器使用系统时钟。

0-2 位: 未用。读=00b, 写=无关

图 17.7 TL0: 定时器 0 低字节

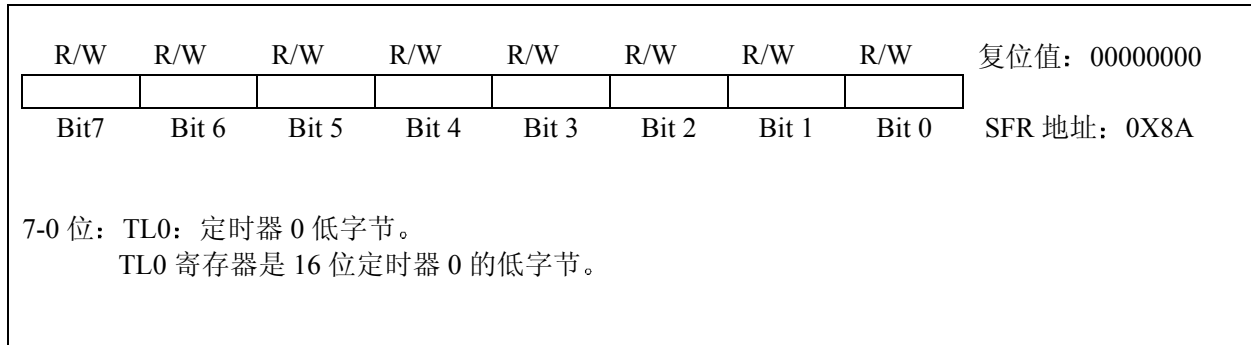


图 17.8 TL1: 定时器 1 低字节

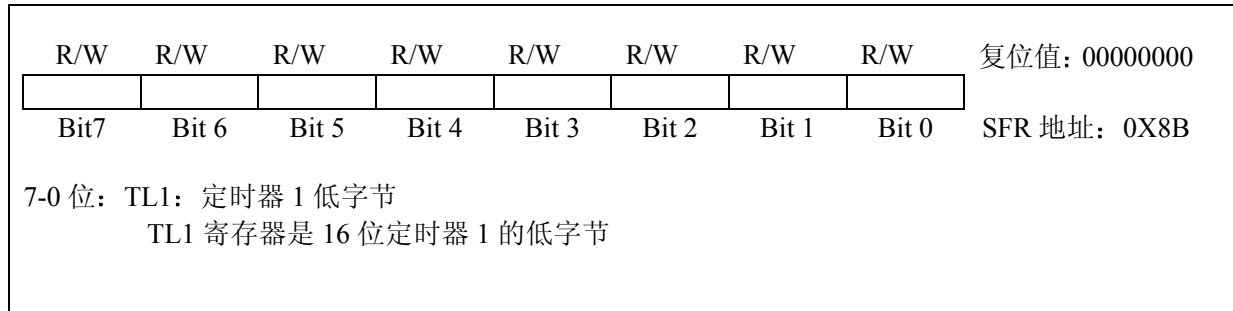


图 17.9 TH0: 定时器 0 高字节

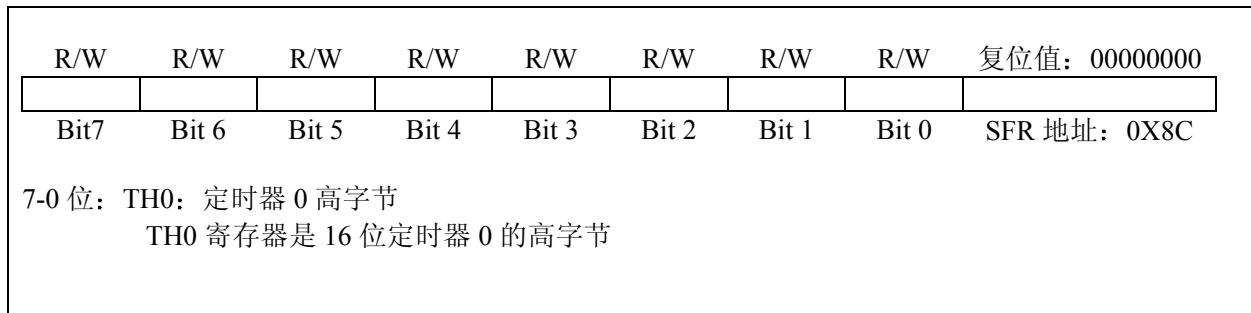
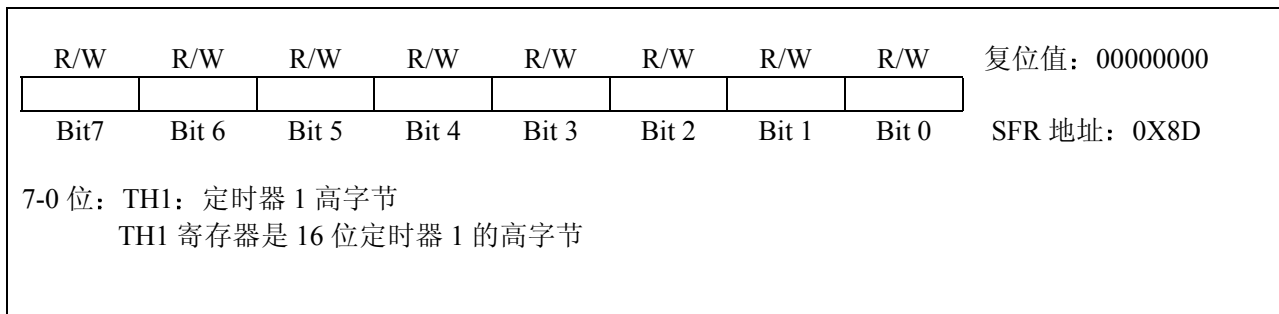


图 17.10 TH1 定时器 1 高字节



## 17.2 定时器 2

定时器 2 是 2 个 8 位 SFR：TL2（低字节）和 TH2（高字节）形成的 16 位计数器/定时器。对于定时器 0 和 1，2 可以使用系统时钟或者外部输入引脚的跳变作为时钟源。计数器/定时器选择位 C/T2（T2CON1）选择定时器 2 的时钟源。复位 C/T2 选择系统时钟作为定时器的输入（按照 CKCON 中的定时器时钟选择位 T2M 的指定被 1 或 12 除）。当 C/T2 被设为 1 时，在 T2 输入引脚从高到低的跳变使计数器/定时器寄存器递增。（参看 14 节关于外部 I/O 引脚的选择和配置问题）。定时器 2 也可以被用来启动 1 个 ADC 数据转换（参看第 5 节）

定时器 2 提供定时器 0，1 不具备的功能。它运行于三种方式之一：带捕捉的 16 位计数器/定时器，带自动重装的 16 位计数器/定时器或者波特率发生器方式。定时器 2 的操作方式通过在定时器 2 控制寄存器中设置配置位来选择。以下是定时器 2 操作方式的总结和用来配置计数器/定时器的 T2CON 位。然后将对每种方式进行详细的描述。

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

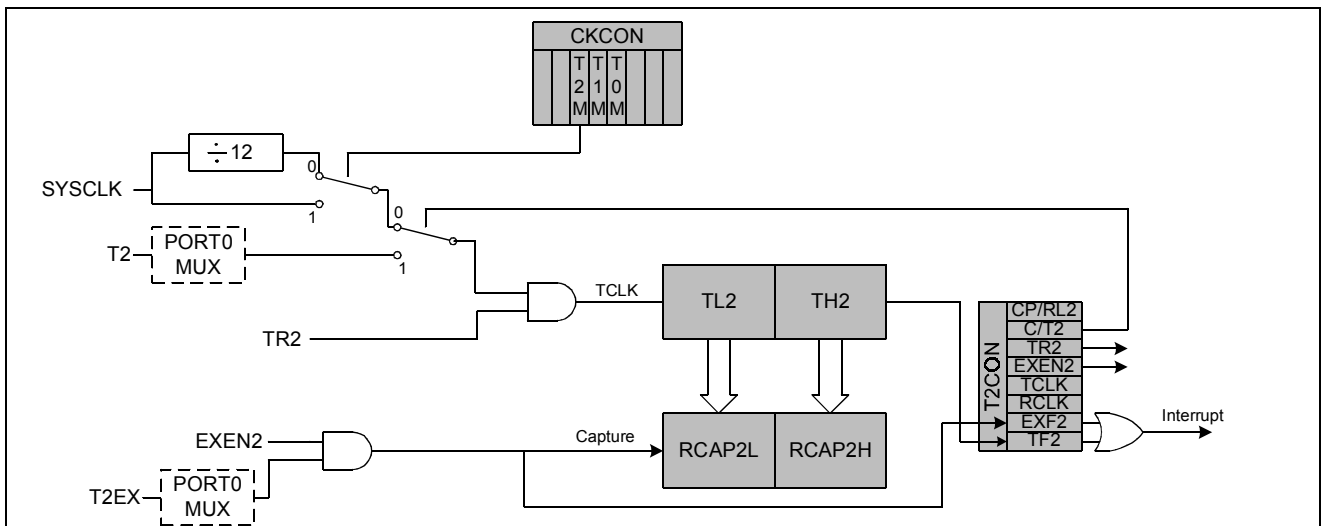
### 17.2.1. 方式 0: 带捕捉的 16 位计数器/定时器

在这个方式中，定时器 2 作为带捕捉的 16 位计数器/定时器工作。在 T2EX 输入引脚上从高到低的跳变使定时器 2 (TH2, TL2) 中的 16 位数值装入捕捉寄存器 (RCAP2H, RCAP2L)。

当工作在计数器/定时器和捕捉方式时，定时器 2 的时钟源可以使用系统时钟或者系统时钟被 12 除，或者在外部 T2 输入引脚从高到低的跳变。复位 C/T2 位 (T2CON1) 将选择系统时钟作为定时器的输入 (按照 CKCON 中的定时器时钟选择位 T2M 决定被 1 或 12 除)。当 C/T2 被置为逻辑 1，T2 输入引脚的从高到低的跳变使计数器/定时器寄存器递增。当 16-位计数器/定时器寄存器不断增加并且从 0xFFFF 溢出到 0x0000 时，TF2 定时器溢出标志 (T2CON7) 被置位并且如果允许中断，将产生中断。

通过设定捕捉/重装选择位 CP/RL2 (T2CON.0) 和定时器 2 控制位 TR2 (T2CON.2) 为逻辑 1 选择带捕捉方式的计数器/定时器。为了允许捕捉，定时器 2 外部允许 EXEN2 (T2CON.3) 也必须置为逻辑 1。如果 EXEN2 被复位，在 T2EX 上的跳变将被忽略。

图 17.11. T2 模式 0 方块图



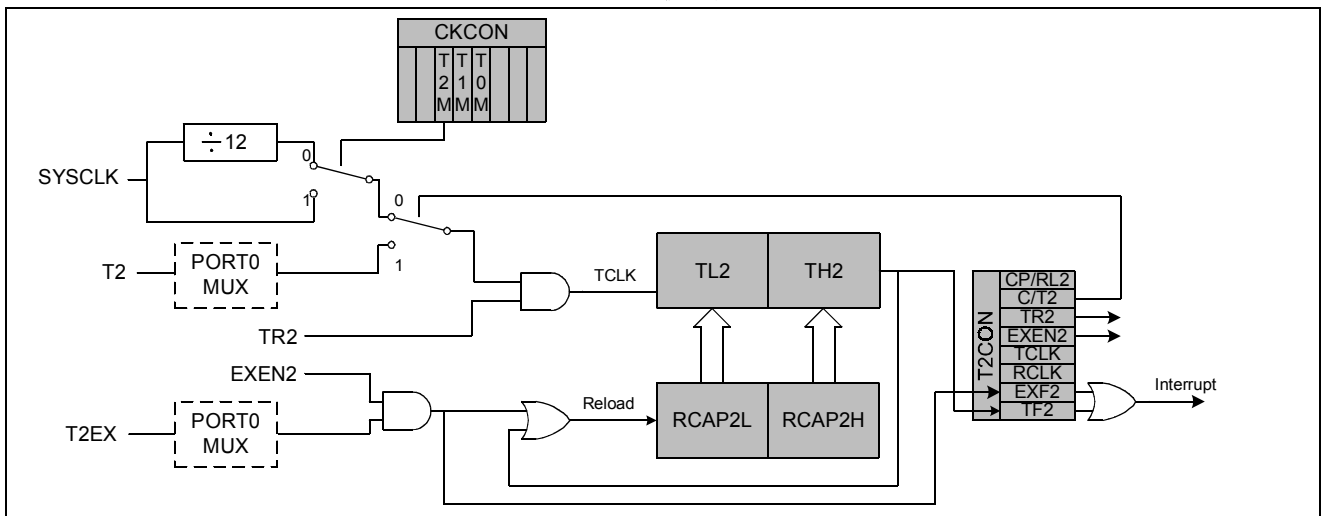


### 17.2.2. 模式 1: 带自动重载的 16 位计数器/定时器

当计数器/定时器寄存器从 0xFFFF 溢出到 0x0000 时, 带自动重装模式的计数器/定时器置位 TF2 定时器溢出标志。如果允许则产生中断。溢出时, 在 2 个捕捉寄存器 (RCAP2H, RCAP2L) 中的 16 位数值被自动装载到计数器/定时器寄存器并且定时器重新开始。

通过复位 CP/RL2 位来选择带自动重装模式的计数器/定时器。置 TR2 为逻辑 1 允许和启动定时器。定时器 2 可以使用系统时钟或在外部输入引脚上的跳变作为它的时钟源, 这些由 C/T2 位确定。如果 EXEN2 被置为逻辑 1, 在 T2EX 上从高到低的跳变也将使定时器 2 被重装。如果 EXEN2 被复位, 在 T2EX 上的跳变将被忽略。

图 17.12 T2 模式 1 方块图



### 17.2.3 模式 2: 波特率发生器

当 UART 工作在模式 1 或 3 中时, 定时器 2 可以作为串行端口 (UART) 的波特率发生器 (参看 161 章关于 UART 操作模式的更多信息), 在波特率发生器模式, 定时器 2 的工作方式和自动重装模式类似。溢出时, 在 2 个捕捉寄存器 (RCAP2H, RCAP2L) 中的 16 位值被自动装载入计数器/定时器寄存器。然而, TF2 溢出标志不被置位并且不产生中断。相反, 溢出事件被作为 UART 移位时钟的输入。定时器 2 的溢出可以被用来独立地为发送和/或接收产生波特率。

通过将 RCLK (T2CON5) 和/或 TCLK (T2CON4) 置为逻辑 1 选择波特率发生器模式。当将 RCLK 或 TCLK 置为逻辑 1 时, 无论 CP/RL2 位的状态如何, 定时器 2 都工作在自动重装模式。当工作在模式 1 或 3 中时, UART 的波特率由定时器 2 的溢出频率决定:

$$\text{波特率} = \text{定时器 2 溢出频率} / 16$$

注意, 在所有其他模式中, 定时器的时基是系统时钟被 1 或 12 除, 这由 CKCON 中的 T2M 位决定。然而, 在波特率发生器模式中, 时基是系统时钟被 2 除, 没有别的除数。如果需要不同的时基, 设 C/T2 位为逻辑 1 将从外部输入引脚 T2 得到时基。在这种情况下, UART 的波特率计算如下:

$$\text{波特率} = \text{FCLK} / [32 * (65536 - [\text{RCAP2H: RCAP2L}])] ]$$

FCLK 是提供给 T2 的信号频率并且 [RCAP2H: RCAP2L] 是捕捉寄存器中的 16 位值。如上所述, 在波特率发生器模式中, 定时器 2 并未置 TF2 溢出标志因此不能产生中断。然而, 如果 EXEN2 被置为逻辑 1, 在 T2EX 输入引脚上的从高到低的跳变将置 EXF2 标志并且如果允许中断, 定时器 2 将产生中断。那时, T2EX 输入可以作为附加的外部中断源。

图 17.13. T2 模式 2 方块图

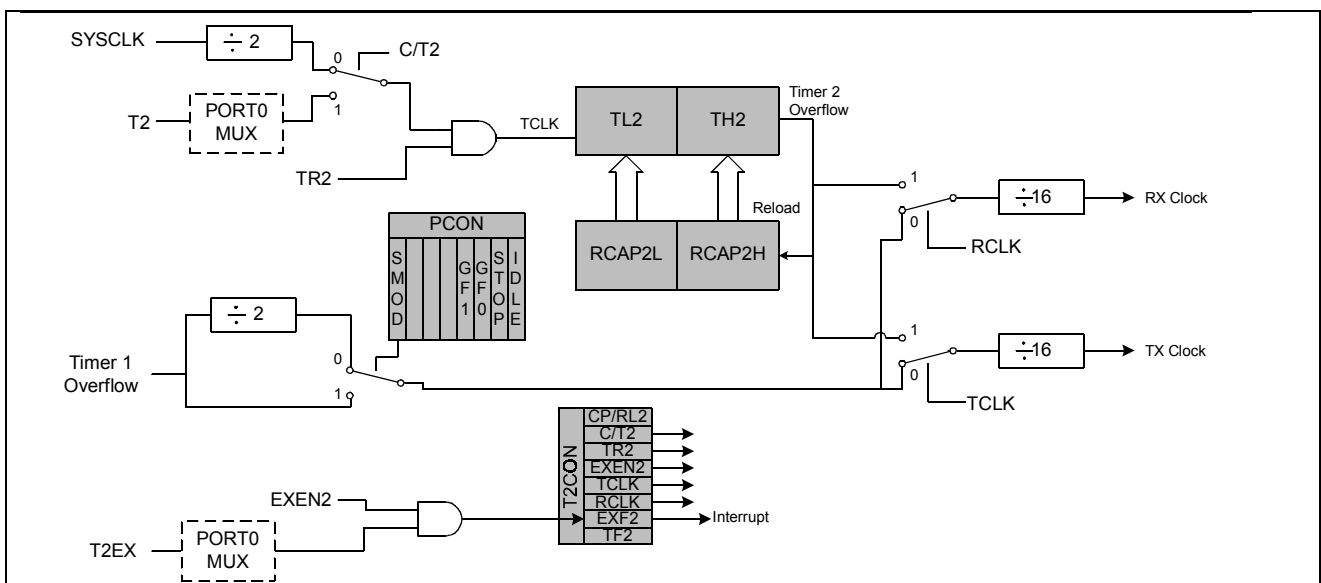


图 17.14. T2CON: 定时器 2 控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	复位值: 00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址: 0XC8

7 位: TF2: 定时器 2 溢出标志  
当定时器 2 从 0xFFFF 溢出到 0x0000 时由硬件置位。当定时器 2 中断允许, 将此位置位使 CPU 转到定时器 2 中断服务程序。此位不被硬件自动复位必须由软件复位。当 RCLK 和/或 TCLK 是逻辑 1 时 TF2 将不会被置位。

6 位: EXF2: 定时器 2 外部标志  
当在 T2EX 输入引脚上的从高到低的跳变引起捕捉或重装且 EXEN2 是逻辑 1 时由硬件置位。当定时器 2 允许中断时, 置此位使 CPU 转到定时器 2 中断服务程序。此位不是被硬件自动复位, 它必须被软件复位。

5 位: RCLK: 接收时钟标志  
选择在模式 1 或 3 中使用哪个定时器作为 UART 的接收时钟。  
0: 定时器 1 溢出用作接收时钟。  
1: 定时器 2 溢出用作接收时钟。

4 位: TCLK: 发送时钟标志  
选择在模式 1 或 3 中使用哪个定时器作为 UART 的发送时钟。  
0: 定时器 1 溢出用作发送时钟。  
1: 定时器 2 溢出用作发送时钟。

3 位: EXEN2: 定时器 2 外部使能  
当定时器 2 未工作在波特率发生器模式下时, 允许在 T2EX 上从高到低的跳变触发捕捉或重装  
0: T2EX 上从高到低的跳变被忽略  
1: T2EX 上从高到低的跳变引起捕捉或重装

2 位: TR2: 定时器 2 控制  
此位允许/禁止定时器 2  
0: 定时器 2 禁止  
1: 定时器 2 允许

1 位: C/T2: 计数器/定时器选择  
0: 定时器功能: T2M (CKCON5) 所定义的时钟使定时器 2 递增。  
1: 计数器功能: 在外部输入引脚 P06T2 上从高到低的跳变使定时器 2 递增

0 位: CP/RL2: 捕捉/重装选择  
此位选择定时器 2 工作在捕捉还是重装模式。EXEN2 必须是逻辑 1, 这样在 T2EX 上从高到低的跳变才能被检测并用来触发捕捉或重装。如果 RCLK 或 TCLK 被置位, 此位被忽略并且定时器 2 将工作在自动重装模式。  
0: 定时器 2 溢出或 T2EX (EXEN2=1) 从高到低跳变时自动重装  
1: 在 T2EX (EXEN2=1) 从高到低跳变时进行捕捉

图 17.15 RCAP2L 定时器 2 捕捉寄存器低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值：00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址：0XCA

7-0 位：RCAP2L 定时器 2 捕捉寄存器低字节  
 当定时器 2 配置在捕捉模式时 RCAP2L 寄存器捕捉定时器 2 的低字节。当定时器 2 配置在重装模式时，它保存重装值的低字节

图 17.16 RCAP2H：定时器 2 捕捉寄存器高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值：00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址：0XCB

7-0 位：RCAP2L 定时器 2 捕捉寄存器高字节  
 当定时器 2 配置在捕捉模式时 RCAP2L 寄存器捕捉定时器 2 的高字节。当定时器 2 配置在重装模式中时，它保存重装值的高字节

图 17.17 TL2：定时器 2 低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值：00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址：0XCC

7-0 位：TL2：定时器 2 低字节  
 TL2 寄存器包含 16 位定时器 2 的低字节

图 17.18 TH2：定时器 2 高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值：00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SFR 地址：0XCD

7-0 位：TL2：定时器 2 高字节  
 TL2 寄存器包含 16 位定时器 2 的高字节

## 18. JTAG

### 描述

MCU 有集成的 JTAG 接口和逻辑来支持 FLASH 读写操作和非侵入的在线调试。C8051F2xx 可以放在 JTAG 测试环中，以保证在 1 个系统中仅有 1 个 JTAG 接口，这样可以对其他部分进行边界扫描并且仍然可以使用 C8051F2xx 调试和闪程序。然而，C8051F2xx 不支持边界扫描，它的工作符合 IEEE1149.1 标准。JTAG 接口包括在 MCU 上的 4 个引脚，它们是 TCK，TMS，TDI，TDO。这些引脚耐压都是 5 伏。通过 16 位的 JTAG 指令寄存器 (IR)，可以发出图 18.1 中的 5 条指令。这些命令可以选择设备 ID 码，或为闪编程操作选择寄存器。通过 BYPASS 来说明它的默认设置。在 MCU 上有 4 个数据寄存器与闪读写操作有关。

图 18.1 IR: JTAG 指令寄存器



### 18.1 闪编程命令

可以利用闪控制，闪数据，闪地址和闪标度寄存器直接对 JTAG 接口编程。这些间接的数据寄存器通过 JTAG 指令寄存器来访问。在间接数据寄存器上的读写操作通过首先设定 IR 寄存器中的合适 DR 地址来进行。然后通过向选定的数据寄存器写适当的间接操作码启动读写操作。进入到这个寄存器的指令有如下格式：

19: 18	17: 0
非直接操作码	写数据

IndOpCode: 这些位根据下面的表设定进行的操作：

非直接操作码	操作
0x	查询
10	读
11	写

如下所述，查询操作用来检查忙位。尽管执行了 1 个 Capture-DR，查询操作不需要 Capture-DR 操作。既然不能更新，可以通过移入/出 1 位来完成查询。

读操作启动对 DRAddress 所寻址的寄存器的读操作。可以通过仅将 2 个字节移入间接寄存器启动读操作。读操作启动后，必须对忙位进行查询以确定操作何时结束。

写操作将 WriteData 写到被 DRAddress 所寻址的寄存器中。位数达到 18 位的寄存器都可以进行写操作。如果被写的寄存器少于 18 位，WriteData 中的数据应该左对齐，即它的最高位应该在 17 位以上。这使得较短的寄存器可以在较少的 JTAG 时钟周期中写完。例如，8 位的寄存器可以通过仅移动 10 位写完。启动写操作之后，应该对忙位进行查询以决定下一次操作何时可以开始。当读或写操作正在进行时指令寄存器的内容不允许改变。

从间接数据寄存器的输出数据格式如下：

19	18: 1	0
0	读数据	忙

忙位指示当前操作没有结束。当启动操作时它变高，操作结束后变低。当忙是高时读写命令被忽略。事实上，如果忙位查询为低，将开始另 1 次读写操作，而当忙位查询为低时下一轮的 JTAG 写操作可以开始。在忙线查询为低之前，各种操作的命令将被忽略，直到忙线查询为低才能开始新一轮的操作。此位位于 0 位，从而可以通过单位移位来进行查询。当正在等待读操作完成并且忙位是 0 时，下面的 18 位可以被移出以得到数据。ReadData 总是右对齐。这使得少于 18 位寄存器的读取可以使用少一些的移位来完成。例如，读 1 个字节的的结果需要 9 次移位。



图 18.4 FLASHDAT: JTAG 闪数据寄存器

DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	FAIL	BUSY	复位值: 000000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit7	Bit 6	

这个寄存器用来通过 JTAG 接口读或写数据到闪内存。

9-2 位: 数据 7-0: 闪数据字节

1 位: 失败: 闪失败位

0: 前一次闪内存操作成功

1: 前一次闪内存操作失败。通常指示相关的内存位置被锁定。

0 位: BUSY: 闪忙位

0: 闪接口逻辑不忙

1: 闪接口逻辑正在处理请求。在 BUSY=1 时进行读或写操作不会启动另一个操作。

图 18.5 FLASHASCL: JTAG 闪标度寄存器

FOSE	FRAE	-	-	FLSCL3	FLSCL2	FLSCL1	FLSCL0	复位值: 00000000
Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

这个寄存器控制闪读的定时电路和用来为闪操作进行正确定时的预标度器。

7 位: FOSE: 闪一次允许位

0: 闪读选通是完整时钟周期宽

1: 闪读选通是 50n 秒

6 位: FRAE: 闪始终读位

0: 只有在需要读闪内存时闪输出允许和传感放大器允许。

1: 闪输出允许和传感放大器允许始终打开。这可以用来限制由于传感放大器的切换造成的数字电流的波动, 从而减少了数字噪声。

5-4 位: 未用。读=00b, 写=无关

3-0 位: FLSCL3-0: 闪予级控制位

FLSCL3-0 位控制为闪操作产生定时信号的预标度器。它的值应该在任何闪写或闪删除操作开始之前写入。所写的值应该是满足下面条件的最小整数:

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

$f_{SYSCLK}$  是系统时钟频率。当 FLSCL[3: 0]=1111b 时禁止所有的读/写/删除操作。



## 18.2 边界扫描旁路和 ID 码

MCU 不支持边界扫描 (IEEE 11491)，然而，它支持旁路和 ID 码功能。因为 MCU 利用了 JTAG 的闪存编程和调试支持，而且在一个系统中别的设备也可以使用 JTAG 边界扫描，MCU 支持放在 BYPASS 中，从而用户可以为一个系统保证 1 个 JTAG 端口。另外，MCU 还支持 ID 码。

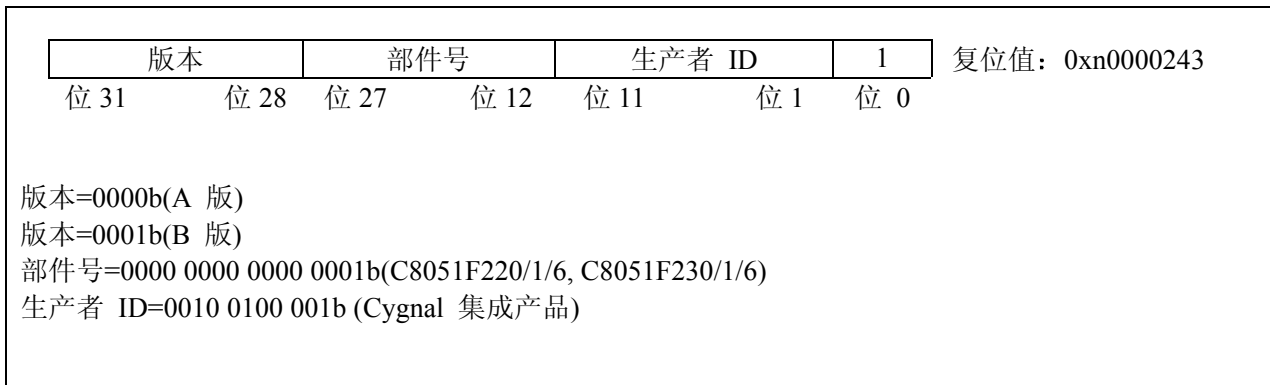
### 18.2.1. BYPASS 指令

BYPASS 指令通过 IR 进行访问。它提供到标准的 1 位 JTAG Bypass 数据寄存器的访问。

### 18.2.2. IDCODE 指令

IDCODE 指令通过 IR 进行访问。通过它可以访问 32 位的设备 ID 寄存器。

**图 18.6. DEVICEID: JTAG 设备 ID 寄存器**



## 18.3. 调试支持

MCU 的内置 JTAG 和调试电路利用安装在终端应用程序中的生产部件提供非插入、全速的在线调试，这是使用 4 脚的 JTAG I/F 来完成的。CygnaI 的调试系统支持对内存和寄存器的检查、修改，支持断点、堆栈跟踪和单步。不需要另外的目标 RAM，程序内存或通讯通道。当模拟时所有的数字和模拟外设都是实用的并且运行正确。当 MCU 在单步调试或处于断点时 WDT 被禁止。

C8051F2xxDK 开发包具有所有利用 C8061F206, C8051F220/1/6 和 C8051F230/1/6 开发应用程序和进行在线调试所必需的软硬件。这个开发包包括了带有调试器和集成 8051 汇编器的集成开发环境 (IDE)。它还有一个 RS232 到 JTAG 的接口模块称为 EC，另外还包括 RS232 和 JTAG 电缆，以及安装在墙中的电源。