NuMicro® 家族

1T 8051微控制器

CM1003系列

CM1003CJ3AE

CM1003DJ3AE

CM1003DL8AE

技术参考手册

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro® microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

目录

1 概述 9
1.1 主要特性和应用 9
2 特性 10
3 料号信息 14
3.1 CM1003系列封装型号 14
3.2 CM1003系列命名规则 14
3.3 CM1003系列选型指南 15
4 引脚配置 16
4.1 引脚配置 16
4.1.1 CM1003系列引脚图 16
4.1.2 CM1003系列多功能引脚分配图 17
4.2 CM1003系列引脚描述 22
4.2.1 CM1003系列引脚映射 22
4.2.2 CM1003系列引脚功能描述 24
5 功能框图 28
5.1 CM1003系列功能框图 28
6 功能描述 29
6.1 存储架构 29
6.1.1 概述 29
6.1.2 程序存储 (APROM / LDROM) 29
6.1.3 数据闪存(Data Flash) 31
6.1.4 安全保护存储器 (SPROM) 31
6.1.5 配置字节(CONFIG) 32
6.1.6 数据存储 (RAM) 38
6.1.7 特殊功能寄存器 (SFR) 41
6.2 系统管理 211
6.2.1 时钟系统 211
6.2.2 电源管理 219
6.2.3 电源检测和复位 221
6.2.4 中断系统 234
6.3 Flash存储控制 248
6.3.1 在应用编程 (IAP) 248
6.3.2 在线电路编程(ICP) 260
6.3.3 片上调试 (OCD) 261
6.3.4 96位唯一标识符 262

6.4 通用端口 I/O (GPIO) 263
6.4.1 功能描述 263
6.4.2 寄存器描述 268
6.5 定时器 276
6.5.1 定时器/计数器0 和 1 276
6.5.2 定时器2和输入捕获 284
6.5.3 定时器3 297
6.6 看门狗定时器 (WDT) 300
6.6.1 概述 300
6.6.2 功能描述 301
6.6.3 寄存器描述 303
6.7 自唤醒定时器 (WKT) 305
6.7.1 概述 305
6.7.2 寄存器描述 306
6.8 脉冲宽度调制 (PWM) 308
6.8.1 概述 308
6.8.2 基本配置 308
6.8.3 框图 310
6.8.4 功能描述 311
6.8.5 寄存器描述 317
6.9 串口 (UART0 & UART1) 335
6.9.1 概述 335
6.9.2 功能描述 335
6.9.3 寄存器描述 343
6.10 ISO 7816-3 接口 (SC0~2 & UART2 ~ 4) 353
6.10.1概述 353
6.10.2特点 353
6.10.3基本配置 353
6.10.4框图 354
6.10.5功能描述 354
6.10.6寄存器描述 360
6.11 内部集成电路 (I^2C) 373
6.11.1概述 373
6.11.2功能描述 373
6.11.3寄存器描述 386
6.12 串行外设接口 (SPI) 392
6.12.1概述 392
6.12.2结构框图 392

6.12.3功能描述 393
6.12.4寄存器描述 398
6.13 12位模数转换器 (ADC) 403
6.13.1概述 403
6.13.2基本配置 403
6.13.3框图 404
6.13.4功能描述 404
6.13.5寄存器描述 408
6.14 指令集 420
6.14.1指令集和寻址模式 420
6.14.2读-修改-写指令 421
6.14.3指令集列表 421
7 应用电路 425
7.1 供电方案 425
7.2 外设应用方案 426
8 电气特性 427
9 封装尺寸 428
9.1 LQFP 48引脚 (7.0 x 7.0 x 1.4 mm³) 428
9.2 QFN 33引脚(4.0 x 4.0 x 0.8 mm³) 429
10 缩写词 430
10.1 缩写词列表 430
11 版本历史 431

图集

图4.1-1 LQFP48引脚分配 16
图4.1-2 QFN33引脚分配 16
图4.1-3 LQFP48封装的引脚分配 17
图4.1-4 QFN33封装的引脚分配 20
图5.1-1 功能框图 28
图6.1-1 CM1003 64KB系列程序存储映射图 30
图6.1-2 CM1003 32KB 系列程序存储映射图 31
图6.1-3 SPROM存储区映射和SPROM保护模式 31
图6.1-4 CONFIG0 复位后自动重载 33
图6.1-5 CONFIG2 上电复位重载 35
图6.1-6 数据存储映射图 38
图6.1-7 内部256 字节RAM 寻址 39
图6.2-1 时钟系统框图 211
图6.2-2 欠压检测框图 223
图6.2-3 nRESET 复位波形 227
图6.2-4 启动选择结构图 232
图6.4-1 准双向模式结构 264
图6.4-2 推挽模式结构 264
图6.4-3 仅输入模式结构 265
图6.4-4 开漏模式结构 265
图6.4-5 引脚接口框图 267
图6.5-1 定时器/计数器 0和1 处于模式0 276
图6.5-2 定时器/计数器 0和1 处于模式1 277
图6.5-3 定时器/计数器 0和1 处于模式2 277
图6.5-4 定时器/计数器 0和1 处于模式3 278
图6.5-5 定时器2 框图 284
图6.5-6 定时器2自动重载模式和输入捕获模组功能框图 285
图 6.5-7 定时器2 比较模式和输入捕获模组功能框图 286
图6.5-8 定时器3 框图 297
图6.6-1 看门狗作为一个超时复位定时器 301
图 6.6-2 看门狗定时器框图 301
图6.7-1 自唤醒定时器框图 305
图6.8-1 PWM0 框图 310

图6.8-2 PWM1/2/3 框图 311
图6.8-3 PWM0 和故障刹车输出控制框图 312
图6.8-4 PWM1/2/3 控制框图 313
图6.8-5 PWM边沿对齐型波形 313
图6.8-6 PWM 中心对齐型波形 314
图6.8-7 带死区时间插入的PWM0互补模式 315
图6.8-8 故障刹车功能框图 316
图6.8-9 PWM中断类型 316
图6.9-1 串口模式0时序图 335
图6.9-2 串口模式1时序图 336
图6.9-3 串口模式2和模式3时序图 337
图6.10-1 SC控制器框图 354
图6.10-2 SC 接口连接 355
图6.10-3 SC 激活和冷复位序列 355
图6.10-4 SC 热复位序列 356
图6.10-5 SC 停用序列 357
图6.10-6 SC 数据字符 357
图6.10-7 初始字符 TS 358
图6.10-8 SC 错误信号 358
图6.10-9 发送方向区块保护时间操作 359
图6.10-10 接收方向区块保护时间操作 359
图6.10-11 额外保护时间操作 359
图6.11-1 I^2C 总线互联 373
图6.11-2 I^2C 总线协议 374
图6.11-3 START起始条件, Repeated START重复起始条件和STOP停止条件 374
图6.11-4 主机以7位数据格式向从机发送数据 375
图6.11-5 主机以7位数据格式向从机读取数据 375
图6.11-6 一次I^2C传输的数据格式 375
图6.11-7 应答位 376
图6.11-8 两台主机的仲裁过程 376
图6.11-9 根据当前I^2C 状态来控制I^2C总线 377
图6.11-10 主机发送模式流程和状态 378
图6.11-11 主机接收模式流程和状态 379
图6.11-12 从机接收模式流程和状态 380

图6.11-13 广播呼叫模式的流程和状态 381
圖6.11-14 I²C 超时计数器 383
图6.11-15 保持时间延长使能 387
图6.11-16 I²C 数据移位方向 388
图6.12-1 SPI 结构框图 392
图6.12-2 SPI 多主机, 多从机连接图 393
图6.12-3 SPI 单主机，单从机连接图 393
图6.12-4 SPI 时钟格式 395
图6.12-5 CPHA = 0 的SPI时钟和数据格式 396
图6.12-6 CPHA = 1 的SPI时钟和数据格式 396
图6.12-7 SPI 溢出波形 398
图6.12-8 SPI 中断请求 398
图6.13-1 12位 ADC框图 404
图6.13-2 外部触发 ADC电路 405
图6.13-3 ADC 结果比较器 406
图6.13-4 ADC 连续模式 407
图7.1-1 NuMicro CM1003 供电电路 425
图7.2-1 NuMicro CM1003 外设接口电路 426
图9.1-1 LQFP-48 封装尺寸 428
图9.2-1 QFN-33 封装尺寸 429

表格集

表1.1-1 NuMicro 系列主要功能支持表 9
表4.1-1 LQFP48封装的引脚多功能支持表 19
表4.1-2 QFN33封装的引脚多功能支持表 21
表6.1-1 特殊功能寄存器地址列表 50
表6.1-2 SFR定义和复位值 58
表6.2-1 BOF复位值 224
表6.2-2 最小欠压检测脉冲宽度 227
表6.2-3 中断向量 234
表6.2-4 中断优先级层级设置 238
表6.2-5 各中断源特性表 239
表6.3-1 IAP 模式和命令代码 249
表6.4-1 不同I/O模式的配置 263
表6.6-1 看门狗定时器在不同分频下的超时间隔 300
表6.8-1 PWM引脚定义和使能控制寄存器 309
表6.9-1 串口 UART0 模式/波特率说明 339
表6.9-2 串口UART1 模式/波特率说明 339
表6.10-1 UART智能卡引脚定义和使能控制寄存器 353
表6.12-1 从机选择引脚配置 397
表6.14-1 指令集和寻址模式 420
表6.14-2 指令影响标志位设置 420
表6.14-3 指令集 424
表10.1-1 缩写词列表 430

1 概述

NuMicro CM1003系列是一款增强型1T-8051内核微控制器。指令集与标准80C51完全兼容并具备更高效能。它支持16/24MHz工作频率，提供最高64KB 闪存，256字节的RAM和4KB的XRAM，可配置4KB LDROM引导代码区，工作电压范围为2.4V - 5.5V，并符合-40℃至105℃的工业级温度范围。

CM1003系列还提供大量外设，包括：两组16位定时器，一组带有3路输入捕获模式的16位定时器2，一组看门狗定时器，一组自唤醒定时器，一组带自动重载功能，可用于通用目的或波特率发生器的16位定时器3，两组具有帧错误检测和自动地址识别功能的UART，三组ISO 7816-3通讯接口，一组SPI，一组I²C，6路基础型PWM输出，6路增强型PWM输出，8通道共享所有I/O引脚中断，15路12位可达500kSPS的ADC。共支持24个中断源，并支持4级中断优先级配置。

支持封装类型包括： LQFP32引脚, QFN33引脚和LQFP48引脚。

开发资源方面，Nuvoton提供了NuMaker评估板和Nuvoton Nu-Link仿真器，还支持第三方IDE，如Keil C51、IAR EW8051和 NuEclipse SDCC编译平台。

1.1 主要特性和应用

产品线	UART	ISO 7816-3	I²C	SPI	Timer	PWM	ADC
CM1003CJ3AE CM1003DJ3AE CM1003DL8AE	2	3	1	1	4	12	15

表 1.1-1 NuMicro 系列主要功能支持表

The CM1003 适用于广泛的应用，例如:

- 家电
- LED照明控制
- 马达控制
- 工业自动化

2 特性

核心和系统	
8051	• 全静态设计8位高性能1T 8051内核CMOS微控制器 • 指令集完全兼容MCS-51 • 4级中断优先级配置 • 双数据指针(DPTRs)
上电复位 (POR)	• 上电复位门限1.15V
欠压检测 (BOD)	• 4级电压选择支持欠压中断和复位功能(4.4V / 3.7V / 2.7V / 2.2V)
低电压复位 (LVR)	• 复位电压门限2.0V
安全特性	• 96位唯一芯片标识符 (UID) • 128位唯一用户标识符 (UCID) • 128字节加密保护代码空间SPROM
存储器	
闪存Flash	• 高达64 KB应用代码空间(APROM) • 4/3/2/1KB加载代码空间(LDROM)，通过 APROM 配置，可通过在系统编程 (ISP) 设置大小 • 在应用编程(IAP)支持128字节的连续页更新，因此整个APROM可作为数据存储区 • 额外的 128 字节安全保护存储器SPROM • 通过 CONFIG进行代码锁定以确保数据安全
静态存储器SRAM	• 内嵌256字节RAM • 内嵌额外4 KB辅助RAM (XRAM)，通过MOVX 指令访问
时钟	
外部时钟源	• 4~24 MHz 高速外部晶体振荡器(HXT)，用于精确定时操作
内部时钟源	• 默认16 MHz 高速内部振荡器(HIRC)，精度为±1% (25°C,3.3V)， -20°C ~ +105°C时精度为±2%，-40°C ~ -20°C时精度为±4% • 可通过软件切换至可选24MHz高速内部振荡器(HIRC)，精度为±1% (25°C,3.3V)

	• 10 kHz低速内部振荡器(LIRC)，可通过高速内部振荡器用软件校正精度达 ±10%
定时器	
16位定时器	• 两组16位定时器/计数器0和1，与标准8051兼容 • 一组16位定时器2带有3路输入捕获功能，9个输入引脚可供选择 • 一组16位自动重载功能定时器3，可用于产生串行口UART0和UART1的波特率
看门狗	• 6位向上计数器，用于计算看门狗超时间隔。 • 由WDT_CLK = 10 kHz，将低速内部振荡器LIRC作为时钟源，超时时间间隔可选 6.40 ms ~ 1.638s • 可从掉电模式或空闲模式唤醒 • 看门狗超时可选择中断或复位
唤醒定时器	• 16位向上计数自唤醒功能定时器 • 内部LIRC作为时钟源 • 用于低功耗模式或者空闲模式下的自主唤醒，自动恢复计数 • 超时可触发中断
脉宽调制PWM	• 最多可达12路输出引脚选择 • 支持最高 24 MHz 时钟源频率 • 最多支持 4 个 PWM 模块，PWM0 模块提供 6 个输出通道。PWM1 ~ PWM3 模块提供 2 个输出通道。 • 支持每路独立输出模式输出 • 支持3对互补配对模式的PWM输出 • 支持16位分辨率PWM计数器 • 每个 PWM 引脚均支持屏蔽功能和三态使能功能 • PWM0 模块支持 8 位分辨率的死区插入 • PWM0 模块支持刹车功能 • PWM0 模块支持触发ADC事件
模拟接口	
模数转换 (ADC)	• 模拟输入电压范围: 0 ~ AV_{DD} • 保证12位分辨率和10位精度 • 最多 15 个单端模拟输入通道 • 1 个用于带隙电压(VBG) 输入的内部通道 • 高达 500 ksps 的采样率 • 软件向 ADCS 位写 1触发转换 • 外部引脚(STADC) 触发转换

	• PWM 触发转换 • 支持连续转换功能，将 A/D 转换结果自动存储在 XRAM 中
通讯接口	
通用异步收发器UART	• 支持最多 2 个 UART：UART0、UART1 • 最多支持三组配置为 UART的 ISO 7816-3 设备 • UART 波特率时钟来自 HIRC 或 HXT • 全双工异步通信 • 可编程第 9 位 • UART0 的 TXD 和 RXD 引脚可通过软件切换
I²C	• 1 组I²C 设备 • 主机/从机模式 • 主从设备之间双向数据传输 • 7位寻址模式 • 标准模式 (100 kbps) 和快速模式 (400 kbps) • 支持 8 位超时计数器，当 I²C 总线挂起且超时计数器溢出时，将产生I²C 中断请求 • 支持保持时间可改
串行外设接口SPI	• 一组 SPI 设备 • 支持主机或从机模式操作 • 支持 MSB 优先或 LSB 优先传输顺序 • 从机模式最高可达 12 MHz
ISO-7816	• 最多支持三组 ISO 7816-3 设备 • 支持符合ISO 7816-3 的 T=0, T=1 • 支持全双工 UART 模式
通用输入输出GPIO	• 4种 I/O 模式: – 准双向模式 – 推挽输出模式 – 开漏输出模式 – 高阻抗输入模式 • 可选施密特触发输入/TTL 模式 • 每个 I/O 引脚均可配置为边沿或电平触发的中断源 • 支持标准中断引脚 $\overline{INT0}$ and $\overline{INT1}$ • 支持高电流推和高灌驱动的I/O • 输入模式下 I/O 引脚内部上拉电阻启用 • 最大 I/O 速度为 24 MHz

	• 启用引脚中断也将使能唤醒功能
静电放电ESD &电快速放电EFT	
静电放电ESD	• HBM 达±7 kV
电气快速瞬变EFT	• > ±4.4 kV
闩锁效应Latch-up	• 200 mA 通过

3 料号信息

3.1 CM1003系列封装型号

包装不含卤素，符合RoHS和TSCA标准。

	TSSOP20	QFN20	QFN33	LQFP48
料号	CM1003BF2AE CM1003CF2AE	CM1003BF3AE CM1003CF3AE	CM1003CJ3AE CM1003DJ3AE	CM1003DL8AE

3.2 CM1003系列命名规则

C	M	1	003	C	J	3	A	E
芯唐南京	产品线	内核	系列	内存	引脚数	封装	版本	工作温度
C	M:MCU	1:8051	003	B: 16 KB C: 32 KB D: 64 KB	F: 20 J: 32/33 L: 48	2: TSSOP 3: QFN 8: LQFP		E: -40°C ~ +105°C

3.3 CM1003系列选型指南

料号[3]	Flash Memory (KB)	SRAM (B)	LDROM (KB) [1]	I/O	Timer	PWM	通讯接口				ADC(12-Bit)	封装
							ISO 7816-3 [2]	UART	SPI	I²C		
CM1003BF2AE	16	256+1K	4	18	4	6	-	2	1	1	8-ch	TSSOP20
CM1003BF3AE	16	256+1K	4	18	4	6	-	2	1	1	8-ch	QFN20
CM1003CF2AE	32	256+1K	4	18	4	6	-	2	1	1	8-ch	TSSOP20
CM1003CF3AE	32	256+1K	4	18	4	6	-	2	1	1	8-ch	QFN20
CM1003CJ3AE	32	256+4K	4	30	4	12	3	2	1	1	15-ch	QFN33
CM1003DJ3AE	64	256+4K	4	30	4	12	3	2	1	1	15-ch	QFN33
CM1003DL8AE	64	256+4K	4	46	4	12	3	2	1	1	15-ch	LQFP48

注意：

1. 用户加载程序 (LDROM) 由ARPOM独立出 1/2/3/4 KB Flash区域，大小由 ISP ROM编程确定。
2. ISO 7816-3 可配置为UART功能，GPIO 定义为 UART2 ~ UART4。
3. 本文档仅适用于 CM1003xJ/CM1003xL 4 KB RAM 系列产品。本表格中灰色背景型号的规格应参考 CM1003xF 1 KB RAM 系列产品技术参考手册。

4 引脚配置

4.1 引脚配置

用户使用 NuTool - PinConfigure 可以找到引脚配置信息。NuTool – PinConfigure 包含 NuMicro® 家族所有芯片系列的所有型号, 帮助用户方便正确的配置GPIO多功能。

4.1.1 CM1003系列引脚图

4.1.1.1 LQFP48封装

相关料号: CM1003DL8AE

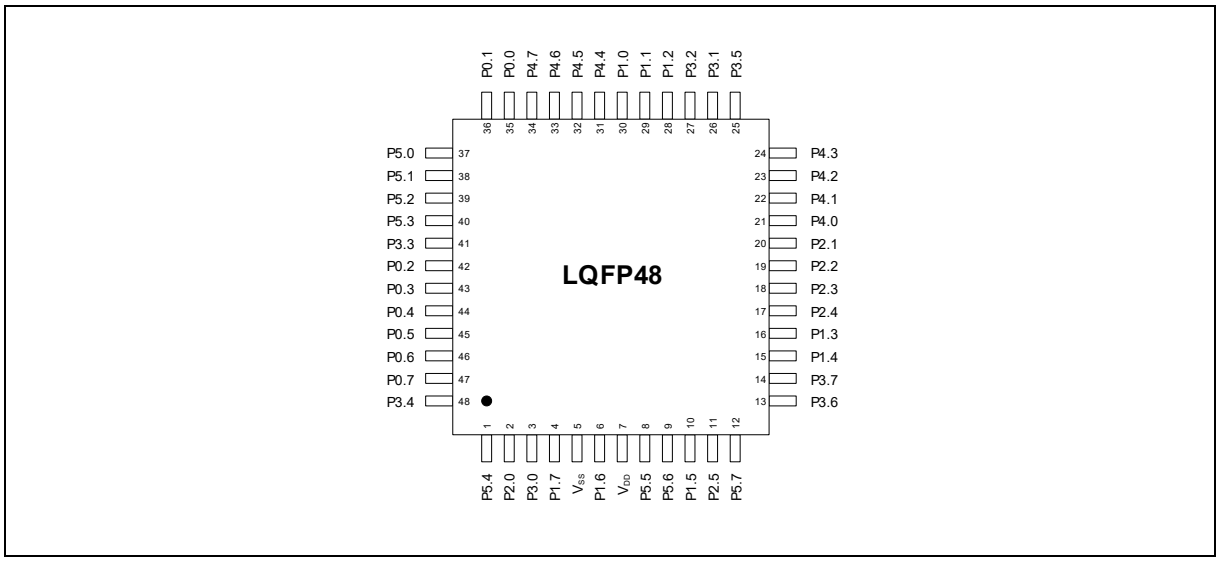


图 4.1-1 LQFP48 引脚分配

4.1.1.2 QFN33封装

相关料号: CM1003CJ3AE / CM1003DJ3AE

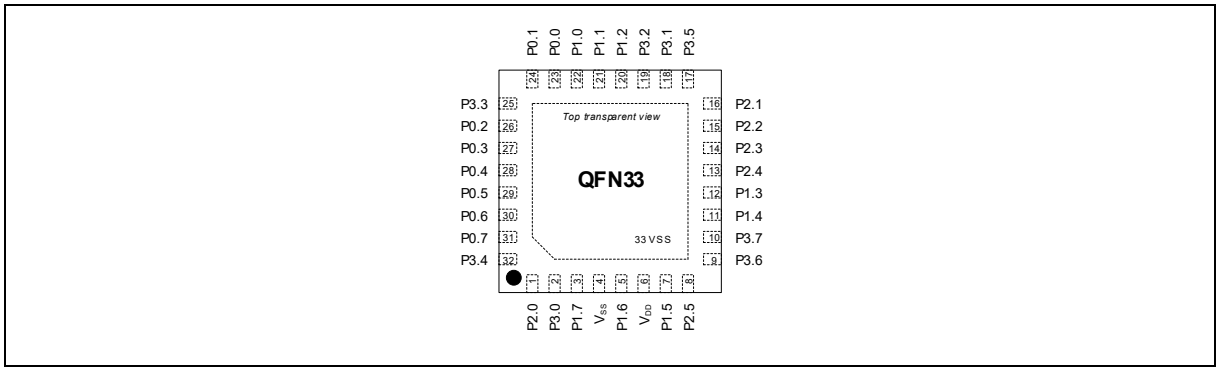


图 4.1-2 QFN33 引脚分配

4.1.2 CM1003系列多功能引脚分配图

4.1.2.1 LQFP48封装

相关料号: CM1003DL8AE

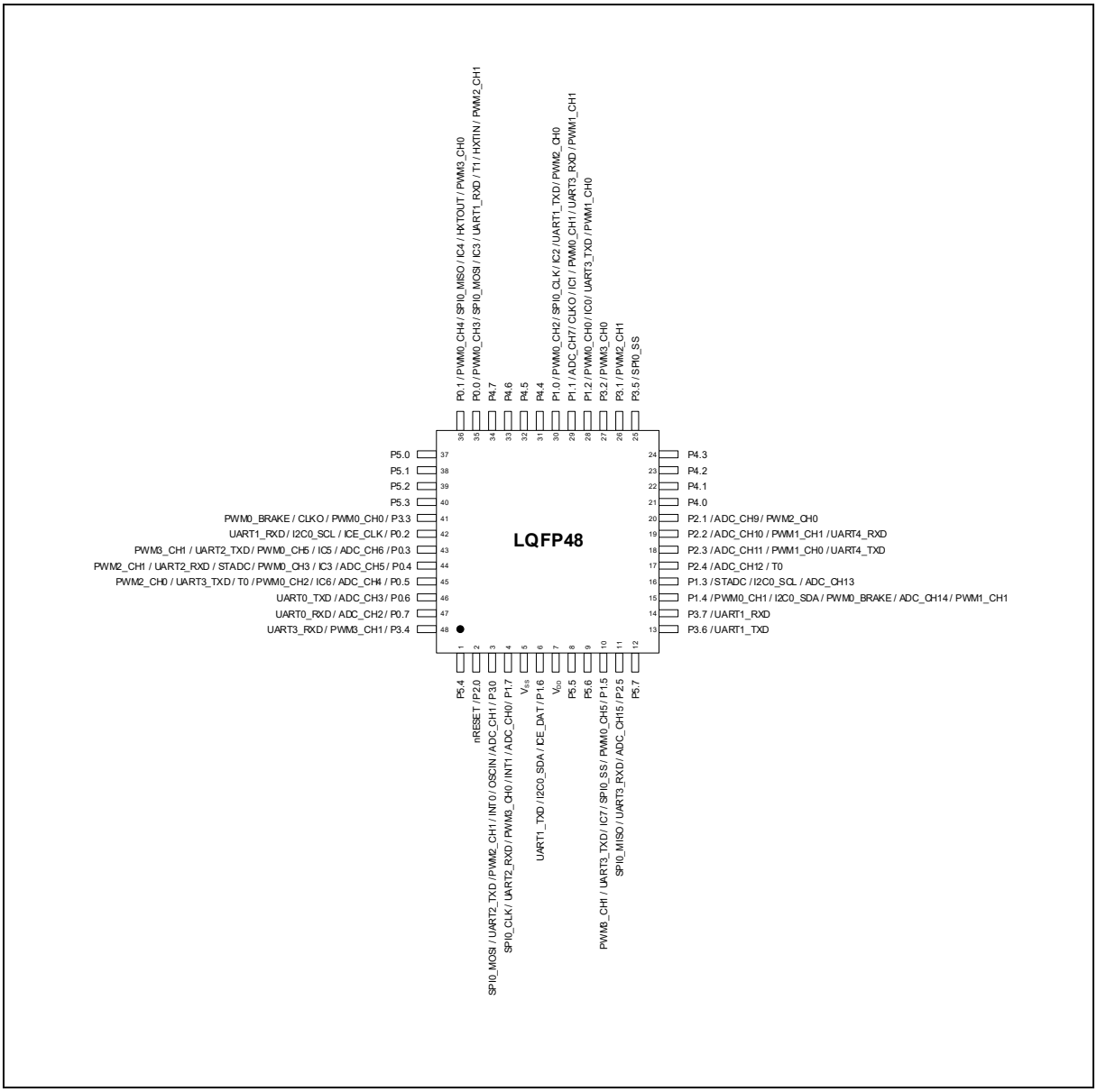


图 4.1-3 LQFP48 封装的引脚分配

引脚	类型	CM1003DL8AE 引脚功能
1	I/O	P5.4
2	I/O	P2.0 / nRESET
3	I/O	P3.0 / ADC_CH1 / OSCIN / INT0 / PWM2_CH1 / UART2_TXD / SPI0_MOSI
4	I/O	P1.7 / ADC_CH0 / INT1 / PWM3_CH0 / UART2_RXD / SPI0_CLK

引脚	类型	CM1003DL8AE 引脚功能
5	P	V_{SS}
6	I/O	P1.6 / ICE_DAT / I2C0_SDA / UART1_TXD
7	P	V_{DD}
8	I/O	P5.5
9	I/O	P5.6
10	I/O	P1.5 / PWM0_CH5 / SPI0_SS / IC7 / UART3_TXD / PWM3_CH1
11	I/O	P2.5 / ADC_CH15 / UART3_RXD / SPI0_MISO
12	I/O	P5.7
13	I/O	P3.6 / UART1_TXD
14	I/O	P3.7 / UART1_RXD
15	I/O	P1.4 / PWM0_CH1 / I2C0_SDA / PWM0_BRAKE / ADC_CH14 / PWM1_CH1
16	I/O	P1.3 / STADC / I2C0_SCL / ADC_CH13
17	I/O	P2.4 / ADC_CH12 / T0
18	I/O	P2.3 / ADC_CH11 / PWM1_CH0 / UART4_TXD
19	I/O	P2.2 / ADC_CH10 / PWM1_CH1 / UART4_RXD
20	I/O	P2.1 / ADC_CH9 / PWM2_CH0
21	I/O	P4.0
22	I/O	P4.1
23	I/O	P4.2
24	I/O	P4.3
25	I/O	P3.5 / SPI0_SS
26	I/O	P3.1 / PWM2_CH1
27	I/O	P3.2 / PWM3_CH0
28	I/O	P1.2 / PWM0_CH0 / IC0 / UART3_TXD / PWM1_CH0
29	I/O	P1.1 / ADC_CH7 / CLKO / IC1 / PWM0_CH1 / UART3_RXD / PWM1_CH1
30	I/O	P1.0 / PWM0_CH2 / SPI0_CLK / IC2 / UART1_TXD / PWM2_CH0
31	I/O	P4.4
32	I/O	P4.5
33	I/O	P4.6
34	I/O	P4.7
35	I/O	P0.0 / PWM0_CH3 / SPI0_MOSI / IC3 / UART1_RXD / T1 / HXTIN / PWM2_CH1
36	I/O	P0.1 / PWM0_CH4 / SPI0_MISO / IC4 / HXTOUT / PWM3_CH0
37	I/O	P5.0
38	I/O	P5.1

引脚	类型	CM1003DL8AE 引脚功能
39	I/O	P5.2
40	I/O	P5.3
41	I/O	P3.3 / PWM0_CH0 / CLKO / PWM0_BRAKE
42	I/O	P0.2 / ICE_CLK / I2C0_SCL / UART1_RXD
43	I/O	P0.3 / ADC_CH6 / IC5 / PWM0_CH5 / UART2_TXD / PWM3_CH1
44	I/O	P0.4 / ADC_CH5 / IC3 / PWM0_CH3 / STADC / UART2_RXD / PWM2_CH1
45	I/O	P0.5 / ADC_CH4 / IC6 / PWM0_CH2 / T0 / UART3_TXD / PWM2_CH0
46	I/O	P0.6 / ADC_CH3 / UART0_TXD
47	I/O	P0.7 / ADC_CH2 / UART0_RXD
48	I/O	P3.4 / PWM3_CH1 / UART3_RXD

表4.1-1 LQFP48封装的引脚多功能支持表

4.1.2.2 QFN33封装

相关料号: CM1003CJ3AE / CM1003DJ3AE

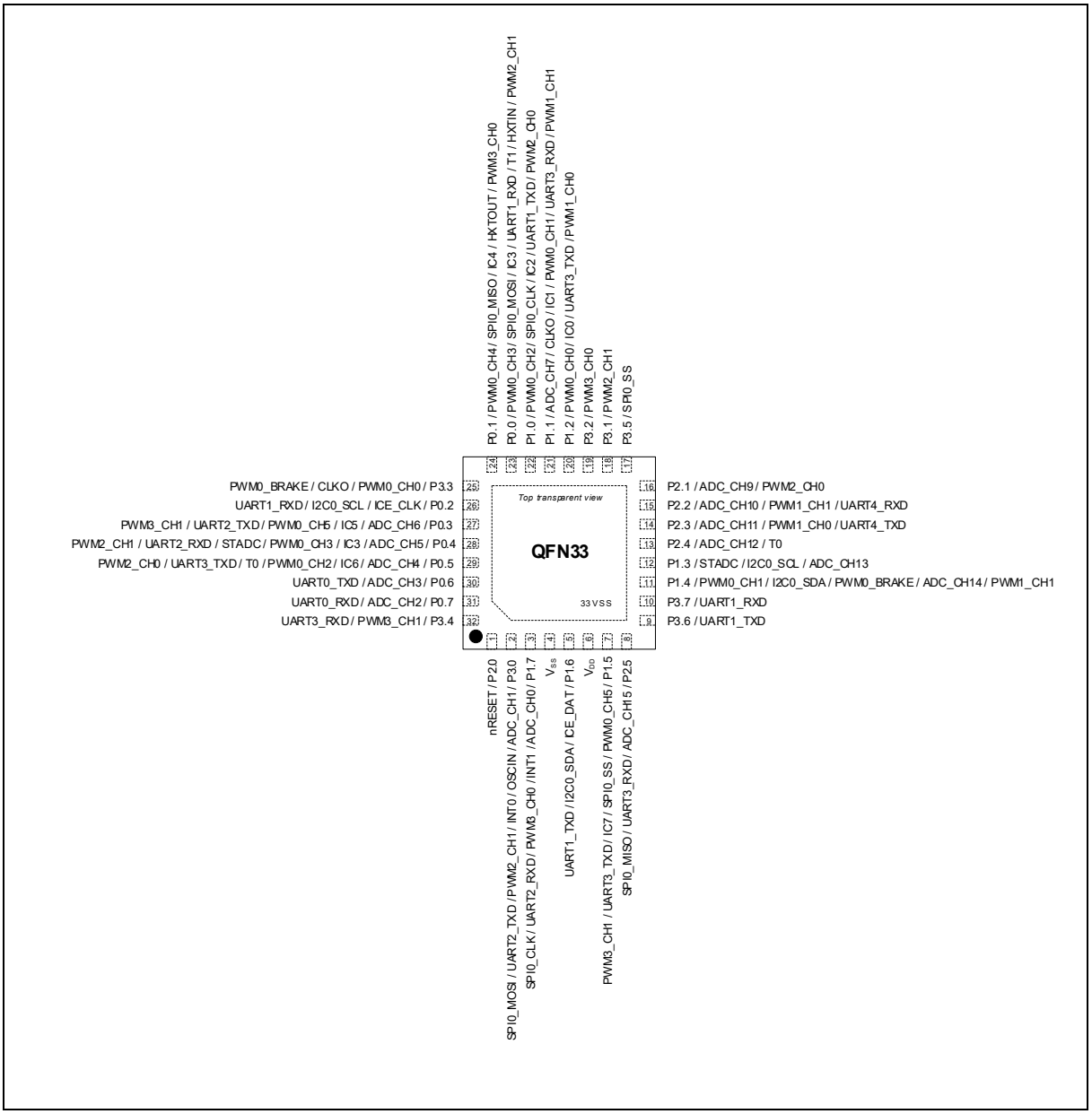


图 4.1-4 QFN33 封装的引脚分配

引脚	类型	CM1003CJ3AE / CM1003DJ3AE 引脚功能
1	I/O	P2.0 / nRESET
2	I/O	P3.0 / ADC_CH1 / OSCIN / INT0 / PWM2_CH1 / UART2_TXD / SPI0_MOSI
3	I/O	P1.7 / ADC_CH0 / INT1 / PWM3_CH0 / UART2_RXD / SPI0_CLK
4	P	V_{SS}
5	I/O	P1.6 / ICE_DAT / I2C0_SDA / UART1_TXD

引脚	类型	CM1003CJ3AE / CM1003DJ3AE 引脚功能
6	P	V_{DD}
7	I/O	P1.5 / PWM0_CH5 / SPI0_SS / IC7 / UART3_TXD / PWM3_CH1
8	I/O	P2.5 / ADC_CH15 / UART3_RXD / SPI0_MISO
9	I/O	P3.6 / UART1_TXD
10	I/O	P3.7 / UART1_RXD
11	I/O	P1.4 / PWM0_CH1 / I2C0_SDA / PWM0_BRAKE / ADC_CH14 / PWM1_CH1
12	I/O	P1.3 / STADC / I2C0_SCL / ADC_CH13
13	I/O	P2.4 / ADC_CH12 / T0
14	I/O	P2.3 / ADC_CH11 / PWM1_CH0 / UART4_TXD
15	I/O	P2.2 / ADC_CH10 / PWM1_CH1 / UART4_RXD
16	I/O	P2.1 / ADC_CH9 / PWM2_CH0
17	I/O	P3.5 / SPI0_SS
18	I/O	P3.1 / PWM2_CH1
19	I/O	P3.2 / PWM3_CH0
20	I/O	P1.2 / PWM0_CH0 / IC0 / UART3_TXD / PWM1_CH0
21	I/O	P1.1 / ADC_CH7 / CLKO / IC1 / PWM0_CH1 / UART3_RXD / PWM1_CH1
22	I/O	P1.0 / PWM0_CH2 / SPI0_CLK / IC2 / UART1_TXD / PWM2_CH0
23	I/O	P0.0 / PWM0_CH3 / SPI0_MOSI / IC3 / UART1_RXD / T1 / HXTIN / PWM2_CH1
24	I/O	P0.1 / PWM0_CH4 / SPI0_MISO / IC4 / HXTOUT / PWM3_CH0
25	I/O	P3.3 / PWM0_CH0 / CLKO / PWM0_BRAKE
26	I/O	P0.2 / ICE_CLK / I2C0_SCL / UART1_RXD
27	I/O	P0.3 / ADC_CH6 / IC5 / PWM0_CH5 / UART2_TXD / PWM3_CH1
28	I/O	P0.4 / ADC_CH5 / IC3 / PWM0_CH3 / STADC / UART2_RXD / PWM2_CH1
29	I/O	P0.5 / ADC_CH4 / IC6 / PWM0_CH2 / T0 / UART3_TXD / PWM2_CH0
30	I/O	P0.6 / ADC_CH3 / UART0_TXD
31	I/O	P0.7 / ADC_CH2 / UART0_RXD
32	I/O	P3.4 / PWM3_CH1 / UART3_RXD
33	EPAD	V_{SS}

表4.1-2 QFN33封装的引脚多功能支持表

4.2 CM1003系列引脚描述

4.2.1 CM1003系列引脚映射

	CM1003	
引脚名称	33引脚	48引脚
P5.4		1
P2.0	1	2
P3.0	2	3
P1.7	3	4
V_{SS}	4	5
P1.6	5	6
V_{DD}	6	7
P5.5		8
P5.6		9
P1.5	7	10
P2.5	8	11
P5.7		12
P3.6	9	13
P3.7	10	14
P1.4	11	15
P1.3	12	16
P2.4	13	17
P2.3	14	18
P2.2	15	19
P2.1	16	20
P4.0		21
P4.1		22
P4.2		23
P4.3		24
P3.5	17	25
P3.1	18	26
P3.2	19	27
P1.2	20	28
P1.1	21	29
P1.0	22	30
P4.4		31

P4.5		32
P4.6		33
P4.7		34
P0.0	23	35
P0.1	24	36
P5.0		37
P5.1		38
P5.2		39
P5.3		40
P3.3	25	41
P0.2	26	42
P0.3	27	43
P0.4	28	44
P0.5	29	45
P0.6	30	46
P0.7	31	47
P3.4	32	48

4.2.2 CM1003系列引脚功能描述

标号		多功能描述[1]	
LQFP48	QFN 33		
7	6	**VDD**	电源供电正极
5	4	**VSS**	电源地电位
35	23	**P0.0**	端口0管脚0
		PWM0_CH3	PWM0输出通道3
		PWM2_CH1	PWM2输出通道1
		IC3	输入捕获通道3
		SPI0_MOSI	SPI主机输出/从机输入脚
		UART1_RXD	UART1 接收输入脚
		XT1_IN	外部4~24 MHz (高速)振荡器输入脚
		OSCIN	如果EXTEN[1:0] = 10b, OSCIN是外部时钟输入引脚
		T1	定时器/计数器0或其切换输出的外部计数输入
36	24	**P0.1**	端口0管脚1
		PWM0_CH4	PWM0输出通道4
		PWM3_CH0	PWM3输出通道0
		IC4	输入捕获通道4
		SPI0_MISO	SPI主机输入/从机输出脚
		XT1_OUT	外部4~24 MHz (高速) 振荡器输出脚
42	26	**P0.2**	端口0管脚2
		I2C0_SCL	I^2C时钟
		UART1_RXD	UART1接收输入脚
		ICE_CLK	ICE / ICP时钟输入
43	27	**P0.3**	端口0管脚3
		ADC_CH6	ADC输入通道6
		PWM0_CH5	PWM0输出通道5
		PWM3_CH1	PWM3输出通道I1
		IC5	输入捕获通道5
		UART2_TXD	UART2发送数据输出
		SC0_CLK	智能卡0时钟脚位
44	28	**P0.4**	端口0管脚4
		ADC_CH5	ADC输入通道5
		PWM0_CH3	PWM0输出通道3
		PWM2_CH1	PWM2输出通道1
		IC3	输入捕获通道3
		UART2_RXD	UART2接收输入脚
		SC0_DAT	智能卡0数据脚位
		STADC	外部启动ADC触发器
45	29	**P0.5**	端口0管脚5
		ADC_CH4	ADC输入通道4
		PWM0_CH2	PWM0输出通道2
		PWM2_CH0	PWM2输出通道0
		IC6	输入捕获通道6
		UART3_TXD	UART3发送数据输出
		SC1_CLK	智能卡时钟脚位
		T0	定时器/计数器0或其切换输出的外部计数输入
46	30	**P0.6**	端口0管脚6
		ADC_CH3	ADC输入通道3
		UART0_TXD	UART0发送数据输出

标号		多功能描述[1]	
LQFP48	QFN 33		
47	31	**P0.7**	端口0管脚7
		ADC_CH2	ADC输入通道2.
		UART0_RXD	UART0 接收输入脚
30	22	**P1.0**	端口1管脚0
		PWM0_CH2	PWM0输出通道2
		PWM2_CH0	PWM2输出通道0
		IC2	输入捕获通道2
		SPI0_CLK	SPI0时钟
		UART1_TXD	UART1发送数据输出
29	21	**P1.1**	端口1管脚1
		ADC_CH7	ADC输入通道7
		PWM0_CH1	PWM0输出通道1
		PWM1_CH1	PWM1输出通道1
		IC1	输入捕获通道1
		UART3_RXD	UART3接收输入
		SC1_DAT	智能卡1数据脚
		CLKO	系统时钟输出
28	20	**P1.2**	端口1管脚2
		PWM0_CH0	PWM0输出通道0
		PWM1_CH0	PWM1输出通道0
		IC0	输入捕获通道0
		UART3_TXD	UART3发送数据输出
		SC1_CLK	智能卡1时钟脚
16	12	**P1.3**	端口1管脚3
		ADC_CH13	ADC输入通道13
		I2C0_SCL	I^2C0时钟
		STADC	外部启动ADC触发器
15	11	**P1.4**	端口1管脚4
		ADC_CH14	ADC输入通道14
		PWM0_CH1	PWM0输出通道1
		PWM1_CH1	PWM1输出通道1
		I2C0_SDA	I^2C0数据
		PWM0_BRAKE	PWM0故障刹车输入脚
10	7	**P1.5**	端口1管脚5
		PWM0_CH5	PWM0输出通道5
		PWM3_CH1	PWM3输出通道1
		IC7	输入捕获通道7
		SPI0_SS	SPI0从机选择输入脚
		UART3_TXD	UART3发送数据输出
		SC1_CLK	智能卡2 时钟脚
6	5	**P1.6**	端口1管脚6
		I2C0_SDA	I^2C0 数据脚
		UART1_TXD	UART1发送数据输出
		ICE_DAT	ICE数据输入或输出脚
4	3	**P1.7**	端口1管脚7
		ADC_CH0	ADC输入通道0
		PWM3_CH0	PWM3输出通道0
		SPI0_CLK	SPI0时钟脚
		UART2_RXD	UART2接收输入

标号		多功能描述[1]	
LQFP48	QFN 33		
		SC0_DAT	智能卡0数据脚
		INT1	外部中断1输入脚
2	1	**P2.0**	当RPD (CONFIG0.2)被编程为0时，端口2管脚0输入可用
		nRESET	它是用于硬件设备复位的施密特触发器输入引脚。此引脚的低电平会重置设备。nRESET引脚有一个内部上拉电阻器，只需将外部电容器连接到VSS即可进行上电复位。
20	16	**P2.1**	端口2管脚1
		ADC_CH9	ADC输入通道9
		PWM2_CH0	PWM2输出通道0
19	15	**P2.2**	端口2管脚2
		ADC_CH10	ADC输入通道10
		PWM1_CH1	PWM1输出通道1
		UART4_RXD	UART4接收输入
		SC2_DAT6	智能卡2数据脚
18	14	**P2.3**	端口2管脚3
		ADC_CH11	ADC输入通道11
		PWM1_CH0	PWM1输出通道0
		UART4_TXD	UART4发送数据输出
		SC2_CLK6	智能卡2时钟脚
17	13	**P2.4**	端口2管脚4
		ADC_CH12	ADC输入通道12
		T0	定时器/计数器0或其切换输出的外部计数输入
11	8	**P2.5**	端口2管脚5
		ADC_CH15	ADC输入通道15
		SPI0_MISO	SPI主机输入/从机输出脚
		UART3_RXD	UART3接收输入
		SC1_DAT6	智能卡1数据脚
3	2	**P3.0**	端口3管脚0
		ADC_CH1	ADC输入通道1
		PWM2_CH1	PWM2输出通道1.
		SPI0_MOSI	SPI主机输出/从机输入脚
		UART2_TXD	UART2发送数据输出
		SC0_CLK6	智能卡0时钟脚
		INT0	外部中断0输入脚
		OSCIN	如果EXTEN[1:0] = 11b, OSCIN是外部时钟输入引脚
26	18	**P3.1**	端口3管脚1
		PWM2_CH1	PWM2输出通道1.
27	19	**P3.2**	端口3管脚2
		PWM3_CH0	PWM3输出通道0.
41	25	**P3.3**	端口3管脚3
		PWM0_CH0	PWM3输出通道0.
		CLK_OUT	系统时钟输出
		PWM0_BRAKE	PWM0故障刹车输入脚
48	32	**P3.4**	端口3管脚4
		PWM3_CH1	PWM3输出通道0
		UART3_RXD	UART3接收输入
		SC1_DAT	智能卡0数据脚
25	17	**P3.5**	端口3管脚5
		SPI0_SS	SPI0从机选择输入脚

标号		多功能描述[1]	
LQFP48	QFN 33		
13	9	**P3.6**	端口3管脚6
		UART1_TXD	UART1发送数据输出
14	10	**P3.7**	端口3管脚7
		UART1_RXD	UART1接收输入
21	-	**P4.0**	端口4管脚0
22	-	**P4.1**	端口4管脚1
23	-	**P4.2**	端口4管脚2
24	-	**P4.3**	端口4管脚3
31	-	**P4.4**	端口4管脚4
32	-	**P4.5**	端口4管脚5
33	-	**P4.6**	端口4管脚6
34	-	**P4.7**	端口4管脚7
37	-	**P5.0**	端口5管脚0
38	-	**P5.1**	端口5管脚1
39	-	**P5.2**	端口5管脚2
40	-	**P5.3**	端口5管脚3
1	-	**P5.4**	端口5管脚4
8	-	**P5.5**	端口5管脚5
9	-	**P5.6**	端口5管脚6
12	-	**P5.7**	端口5管脚7

注意：

1. 所有I/O引脚都可以配置为中断引脚。此功能未在多功能说明中列出。
2. UART0_TXD和UART0_RXD引脚可通过UART0PX (AUXR1.2)进行软件交换。
3. [I2C] 备用功能重映射选项。 I^2C引脚由I2CPX (I2CON.0) 进行软件切换。
4. [STADC] 备用函数重映射选项。STADC引脚由STADCPX(ADCCON1.6)进行软件切换。
5. PIOx寄存器决定哪些引脚是PWM或GPIO。
6. UART2_TXD和UART2_RXD引脚由AUXR2寄存器定义。UART3_TXD, UART3_RXD, UART4_TXD 和UART4_RXD的引脚由AUXR3寄存器定义。

5 功能框图

5.1 CM1003系列功能框图

图5.1-1 展示了CM1003的功能框图，并给出了设备的概述。用户可以在图中找到设备的所有外设功能。

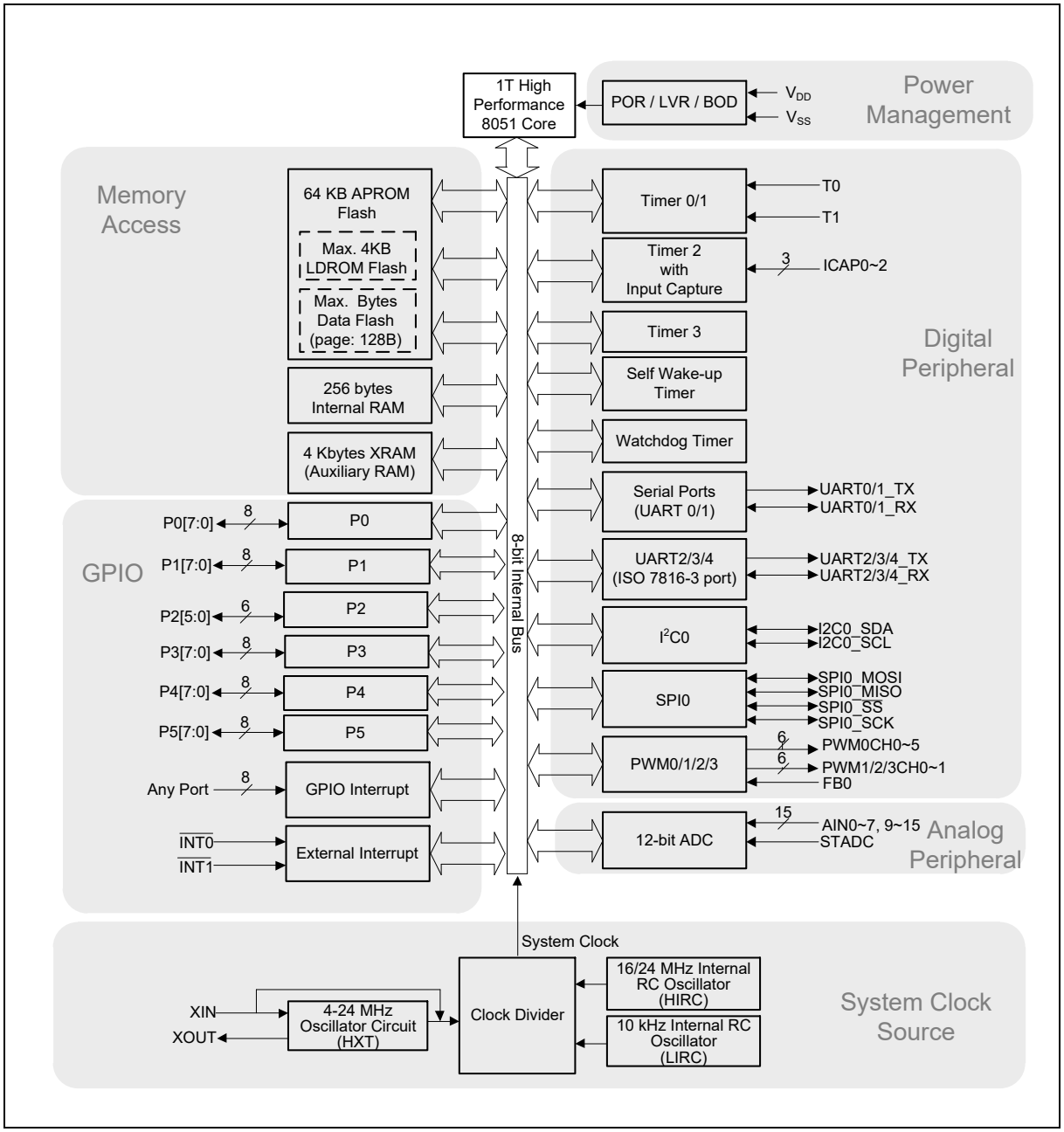


图 5.1-1 功能框图

6 功能描述

6.1 存储架构

6.1.1 概述

基于标准80C51的微控制器将存储分为两个不同的部分，程序存储和数据存储。程序存储用于存储指令代码，而数据存储用于存储程序执行过程中的数据或变量。

数据存储占用的地址空间独立于程序存储。在CM1003中，有256字节的内部RAM。对于许多需要更多RAM的应用，CM1003提供了片上另外4KB的RAM，称为XRAM，可通过MOVX指令访问。

整个嵌入式Flash作为程序存储，分为四块：应用ROM(APROM)通常用于存储用户代码，加载ROM(LDROM) 通常用于存储引导代码，安全保护存储器(SPROM) 通常用于存储安全应用的指令，CONFIG字节用于硬件初始化。事实上，APROM和LDROM的功能相同，但大小不同。每一块由一页一页组成，每页128字节。FLASH控制单元支持擦除、编程和读取模式。外部烧写器通过特定的I/O引脚烧写，在应用程序编程(IAP) 或在系统编程(ISP)都可以执行这些模式。SPROM包含128个字节，位于地址FF80H ~ FFFFH，不支持“整片擦除命令”。SPROM(地址：FFFFH)的最后一个字节用于识别SPROM 代码是非安全模式还是安全模式。

6.1.2 程序存储 (APROM / LDROM)

程序存储存储用于执行的程序代码。如图6.1-1 CM1003 64KB。任何复位之后CPU都从地址0000H开始执行。

为了处理中断，中断服务位置(称为中断向量)应位于程序存储中。每个中断在程序存储中都分配有一个固定位置。中断会导致 CPU 跳转到该位置，并开始执行中断服务例程 (ISR)。例如，外部中断 0 被分配到 0003H 位置。如果要使用外部中断 0，其中断服务例程应从 0003H 位置开始。如果不打算使用该中断，则其中断服务位置可用作通用的程序存储。

中断服务向量位置以八个字节为间隔：0003H用于外部中断0，000BH用于定时器0，0013H用于外部中断1，001BH用于定时器1等。 如果一个中断服务程序足够短(控制应用中的常见情况)，它可以完整地放在这8个字节的间隔中。而较长的中断服务程序应使用JMP指令跳过后面有被其它中断使用的中断地址。

CM1003系列提供两个内部程序存储块APROM和LDROM。虽然它们的行为与标准8051程序存储器相同，但根据ROM大小，它们遵循不同的规则。CM1003的APROM最高可达64KB。用户代码通常放在其中。CPU从此处获取指令以供执行。MOVC指令也可以读取此区域。

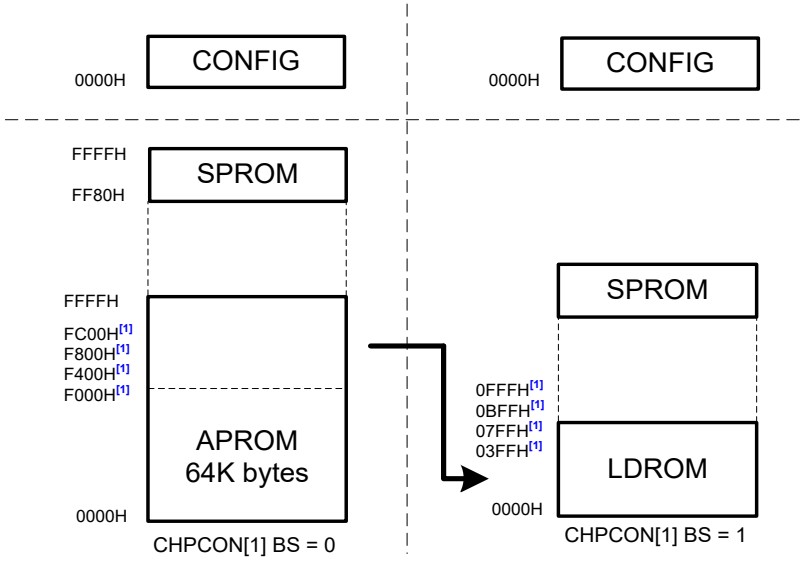
另一个独立的程序存储块称为LDROM。LDROM的常规功能是存储ISP的引导代码。它可以更新APROM空间和CONFIG配置字节。APROM中的代码也可以重新编程LDROM。有关 ISP 的详细信息以及与 APROM 和 LDROM 相关的配置位设置，请参阅章节6.3.1.5“在系统编程 (ISP)”。请注意，APROM和LDROM 是独立的硬件模块，如果CPU从LDROM重新启动，CPU会自动将程序计数器0000H重新向量化到LDROM的起始地址。因此CPU将LDROM视为独立的程序存储且所有中断向量独立于APROM。

CONFIG1

7	6	5	4	3	2	1	0
-	-	-	-	-	LDSIZE[2:0]		
-	-	-	-	-	读/写		

出厂默认值：1111 1111b

位	名称	描述
2:0	**LDSIZE[2:0]**	**LDROM容量选择** 此字段选择LDROM的大小。 针对CM1003Cx: 111 = 无LDROM。APROMis 32K 字节。 110 =LDROM为1K 字节。APROM为31K 字节。 101 =LDROM为2K 字节。APROM为30K 字节。 100 =LDROM为3K 字节。APROM为29K 字节。 0xx =LDROM为4K 字节。APROM为28K 字节。 针对CM1003Dx: 111 = 无LDROM。APROM 为64K 字节。 110 = LDROM为1K 字节。APROM为61K 字节。 101 = LDROM为2K 字节。APROM为60K 字节。 100 = LDROM为3K 字节。APROM为59K 字节。 0xx = LDROM为4K 字节。APROM为58K 字节。



[1] The logic boundary addresses of APROM and LDROM are defined by CONFIG1[2:0].

图 6.1-1 CM1003 64KB 系列程序存储映射图

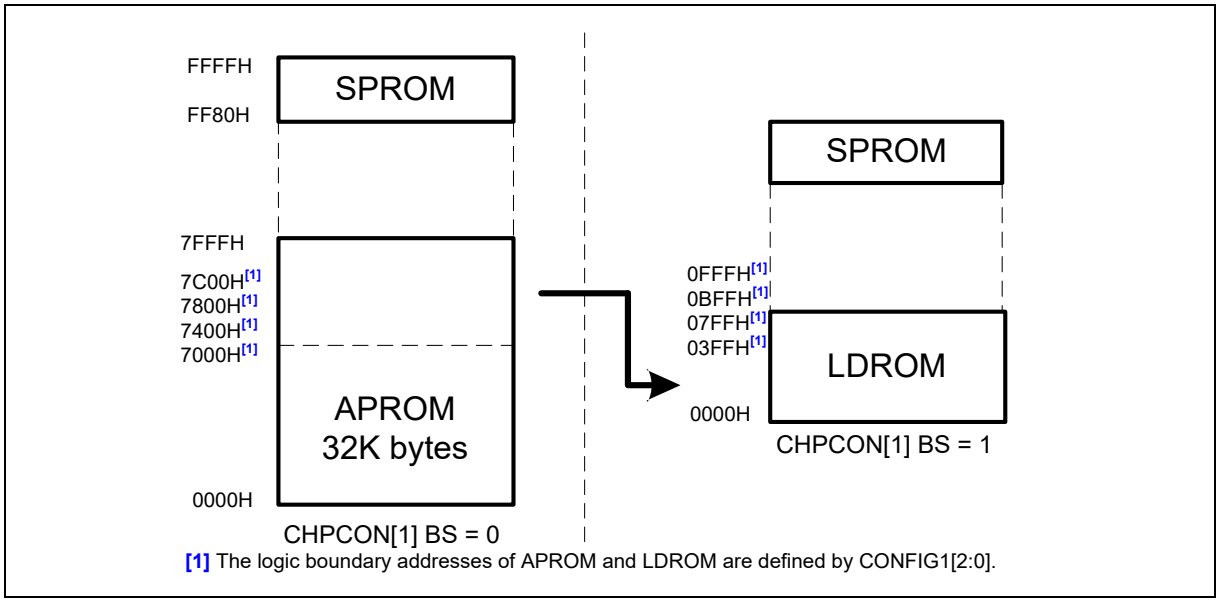


图 6.1-2 CM1003 32KB 系列程序存储映射图

6.1.3 数据闪存(Data Flash)

CM1003系列FLASH与APROM或LDROM共享。APROM或LDROM的任何页面都可以用作FLASH存储，大小不需要特殊配置。FLASH的基地址是通过应用IAP来确定的。有关IAP的详细信息，请参阅章节6.3.1在应用编程 (IAP) 。所有嵌入式闪存每页擦除128字节。

6.1.4 安全保护存储器 (SPROM)

安全保护存储器(SPROM)用于存储安全应用程序的指令。SPROM 包含 128字节 ，地址从FF80H至 FFFFH，不支持整片擦除命令。 图6.1-3 SPROM存储区映射和SPROM 表示了SPROM的最后一个字节(地址: FFFFH)，用于标识SPROM中的程序是非安全，还是安全模式。

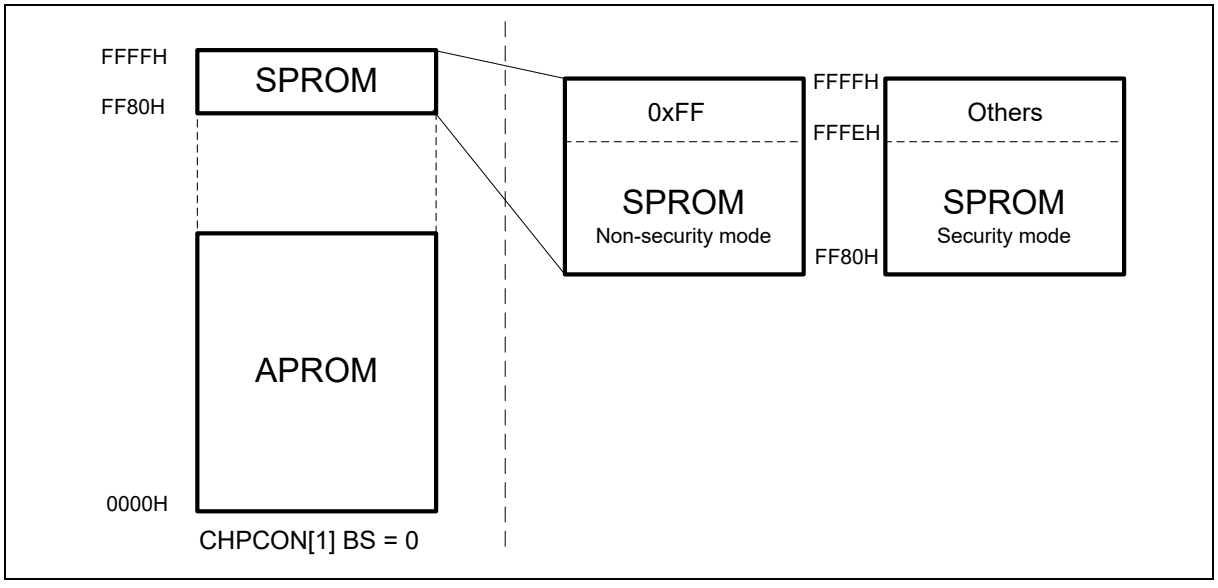


图 6.1-3 SPROM 存储区映射和 SPROM 保护模式

（1）SPROM非安全模式(最后一个字节为0xFF)。SPROM的访问行为与APROM和LDROM相同。

所有区域都可以通过CPU或ISP命令读取，也可以通过ISP命令擦除和编程。

（2）SPROM 安全模式(最后一个字节不为0xFF)。为了在安全模式下隐藏SPROM中的代码,CPU只有在SPROM区域运行时，才能从SPROM执行取指令和获取数据。否则，CPU的数据访问收到的将全是0。为了保护SPROM，即使在安全模式下 ICE(OCD)接口被连接上，CPU获取的指令也将全部为0。在此模式下，SPROM不支持ISP编程，读取或擦除命令。

6.1.5 配置字节(CONFIG)

CM1003有多个硬件配置字节，称为CONFIG，用于配置硬件选项，如安全位、系统时钟源等。这些硬件选项可以通过并行烧录器，在电路编程(ICP)或在应用编程(IAP)重新配置。 某些由特定的配置CONFIG位定义的功能，也可以通过特殊功能寄存器SFR重新配置。因此，复位之后需要加载这些配置CONFIG位到相应的SFR位。这些特殊功能寄存器位也可以通过用户的软件连续控制。

注意：标记为"-"的CONFIG位应始终保持未编程状态。

CONFIG0

7	6	5	4	3	2	1	0
CBS	-	OCDPWM	OCDEN	-	RPD	LOCK	-
读/写	-	读/写	读/写	-	读/写	读/写	-

出厂默认值：1111 1111b

位	名称	描述
7	CBS	**配置开机启动选择** 此位定义了除软件复位外的所有复位后，MCU选择从哪个块重新启动。 1 = 除软件复位外的所有复位后，MCU从APROM启动 0 = 除软件复位外的所有复位后，MCU从LDROM启动
5	OCDPWM	**OCD 调试停止时，PWM 的输出状态** 此位决定当OCD停止CPU时，PWM的输出状态 1 =PWM输出脚为三态模式(Tri-state) 0 =PWM持续输出. 请注意，只有当PWM通道的相应PIO位设置为1时，此位才有效。
4	OCDEN	**OCD 调试使能** 1 = 禁用OCD调试 0 = 使能OCD调试 注意: 当MCU运行在OCD调试模式下并且OCDEN=0,硬件故障复位将被禁用,仅仅HFIF 标志位有效。
3	-	保留
2	RPD	**复位脚禁用** 1=P2.0/nRST引脚的复位功能使能。P2.0/nRST用作外部复位引脚。 0=P2.0/nRST引脚的复位功能禁用。P2.0/nRST用作仅输入引脚。
1	LOCK	**芯片锁定使能** 1 =芯片已解锁。 Flash未被锁定，其内容可以通过并行烧录器/ ICP编程器读取。 0 =芯片已锁定。整个FLASH区域被锁定，通过并行烧录器/ICP编程器读取，读回全部为(FFH)，对FLASH进行编程无效。 请注意， CONFIG字节始终处于解锁状态，可以读取。因此，一旦芯片被锁定，CONFIG字节将无法单独擦除或编程。禁用芯片锁定的唯一方法是执行“整块芯片全擦除动作(whole chip erase)”，但执行此过程时，FLASH和CONFIG位中的所有数据都将被擦除。 如果芯片被锁定，不影响IAP功能。

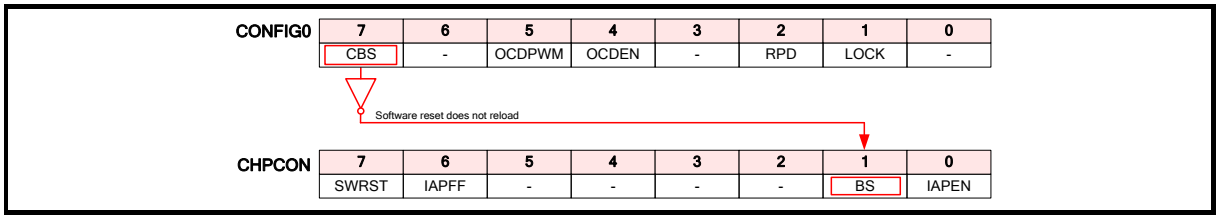


图 6.1-4 CONFIG0 复位后自动重载

CONFIG1

7	6	5	4	3	2	1	0
-	-	-	-	-	LDSIZE[2:0]		
-	-	-	-	-	读/写		

出厂默认值：1111 1111b

位	名称	描述
2:0	**LDSIZE[2:0]**	**LDROM容量选择** 此字段选择LDROM的大小。 针对CM1003Cx: 111 = 无LDROM。APROMis 32K 字节。 110 =LDROM为1K 字节。APROM为31K 字节。 101 =LDROM为2K 字节。APROM为30K 字节。 100 =LDROM为3K 字节。APROM为29K 字节。 0xx =LDROM为4K 字节。APROM为28K 字节。 针对CM1003Dx: 111 = 无LDROM。APROM 为64K 字节。 110 = LDROM为1K 字节。APROM为61K 字节。 101 = LDROM为2K 字节。APROM为60K 字节。 100 = LDROM为3K 字节。APROM为59K 字节。 0xx = LDROM为4K 字节。APROM为58K 字节。

CONFIG2

7	6	5	4	3	2	1	0
CBODEN	-	CBOV[1:0]		BOIAP	CBORST	-	-
读/写	-	读/写		读/写	读/写	-	-

出厂默认值：1111 1111b

位	名称	描述
7	CBODEN	**配置欠压检测使能** 1 = 欠压检测电路打开 0 = 欠压检测电路关闭
6	-	保留
5:4	CBOV[1:0]	**配置欠压检测电压选择** 11 = V_{BOD} 生效电压 2.2V. 10 = V_{BOD} 生效电压 2.7V. 01 = V_{BOD} 生效电压 3.7V. 00 = V_{BOD} 生效电压 4.4V.
3	BOIAP	**欠压抑制IAP** 此位决定欠压状态是否抑制IAP擦除及编程功能。此位仅当欠压检测功能使能后有效。 1 = V_{DD} 低于 V_{BOD} ,抑制IAP 擦除或编程功能。 0 = 任何可用V_{DD}下，都允许IAP擦除及编程功能。
2	CBORST	**配置欠压复位使能** 此位决定检测到电压低于V_{BOD}时，是否欠压复位 1 = 欠压复位功能使能。 0 = 欠压复位功能关闭。

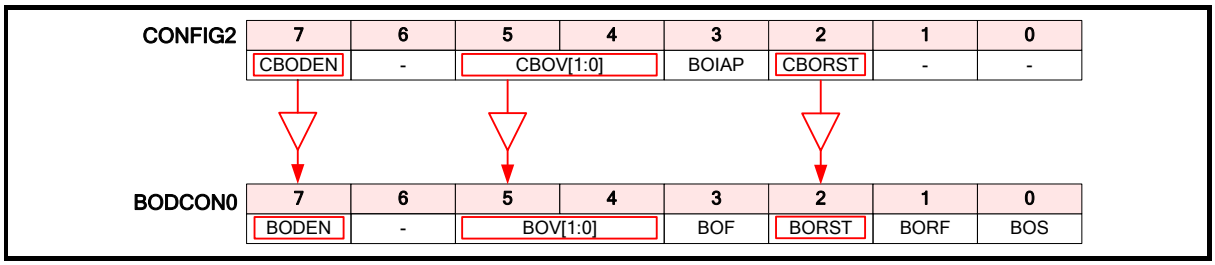


图 6.1-5 CONFIG2 上电复位重载

CONFIG3

7	6	5	4	3	2	1	0
CKFS[1:0]		-		ADCINTEN	-	-	-
读/写		-		读/写	-	-	-

出厂默认值：1111 1111b

位	名称	描述
7:6	**CKFS[1:0]**	**时钟滤波时间选择** 使能时钟滤波器。它可以提高抗噪能力和 EMC 性能。 11 = 15 ns. 10 = 15 ns. 01 = 15 ns. 00 = 时钟滤波器禁用。
5:4	-	保留
3	**ADCINTEN**	**ADC 比较模式下启用ADC中断标志位设置** 1 = 比较结果匹配时设置ADC标志。 0 = 每次ADC比较完成时设置ADC标志。
2:0	-	保留

CONFIG4

7	6	5	4	3	2	1	0
WDTEN[3:0]				-	-	-	-
读/写				-	-	-	-

出厂默认值：1111 1111b

位	名称	描述
7:4	**WDTEN[3:0]**	**看门狗定时器WDT使能** 此字段用于配置 MCU 执行后 WDT 的行为。 1111 = 看门狗定时器WDT禁用，WDT可用作软件控制通用定时器 0101 = WDT作为超时复位定时器使用，并在空闲或掉电模式下停止运行。 其他 = WDT作为超时复位定时器使用，并在空闲或掉电模式下继续运行。
3:0	-	保留

6.1.6 数据存储 (RAM)

6.1.6.1 内部数据存储

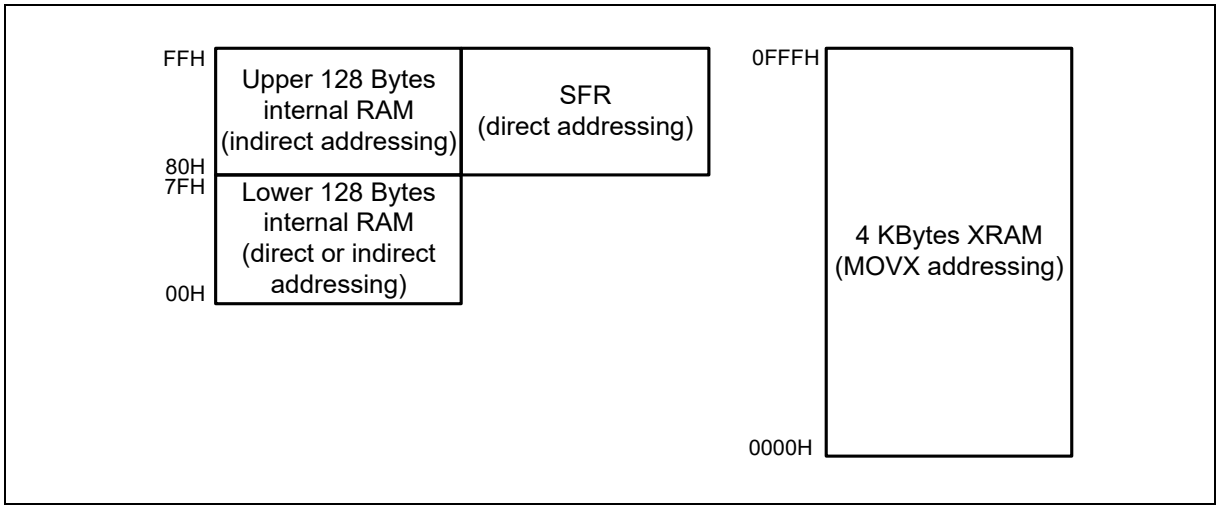


图 6.1-6 数据存储映射图

图6.1-6 数据存储映射图显示了 CM1003 上可用的数据存储空间。数据存储占用一块独立于程序存储的地址空间。数据存储可分为三块，分别是低 128 字节RAM、高 128 字节RAM和 128 字节 的特殊功能寄存器SFR 空间。数据存储地址 8 位宽度，这意味着地址空间只有 256 字节。高于7FH的直接寻址将访问特殊功能寄存器(SFR)空间，高于7FH的间接寻址将访问高128字节RAM。虽然SFR地址空间和高128字节RAM共享相同的逻辑地址80H 到 FFH，但事实上它们物理上是独立的实体。与高128字节RAM区分开的直接寻址只能访问SFR，SFR空间中的16个地址既可以字节寻址也可以位寻址。这些可位寻址的SFR是地址以 0H 或 8H 结尾的 SFR。

所有 80C51 设备都有内部低128字节的RAM。最低32字节作为通用寄存器，分为4组寄存器组，每组寄存器组包含8个寄存器。程序指令将这些寄存器称为R0至R7。程序状态字(PSW[3:4])中的两位RS0和RS1用于选择使用哪个寄存器组。由于寄存器指令比使用直接寻址的指令更短，因此可以提高代码空间的效率。通用寄存器上方的16个字节(字节地址 20H到2FH) 构成一块可位寻址的存储空间块(位地址 00H 到7FH)。80C51指令集有多种位指令，这些指令可以直接寻址该区域中的 128 位。该区域的位地址从00H到7FH。

直接寻址或间接寻址都可以访问低128字节空间。但高 128 字节只能通过间接寻址访问。

对于整个内部256字节RAM来说，另一个应用是堆栈。该区域由堆栈指针(SP)选择，它存储的是堆栈顶部的地址。每当调用 JMP、CALL 或中断时，返回地址都会放在堆栈里。堆栈在RAM中的起始位置没有限制。默认情况下，复位后堆栈指针为07H。用户可以将其更改为所需的任何值。SP会指向上次使用的值。因此SP会递增，然后地址保存到堆栈中。与之相反，内容从堆栈弹出时，将首先读取，然后 SP 会递减。

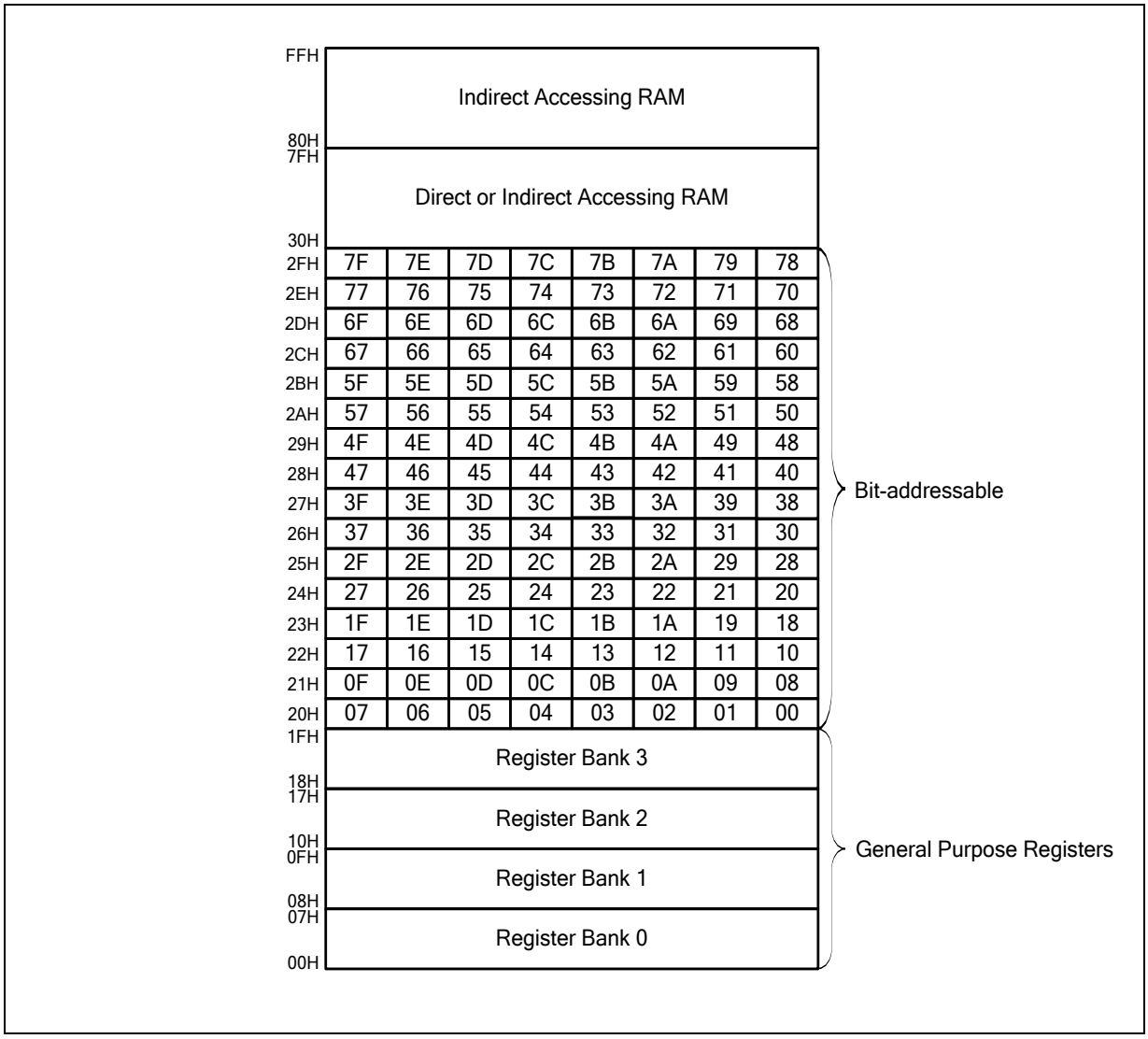


图 6.1-7 内部 256 字节 RAM 寻址

6.1.6.2 片上XRAM

CM1003 系列提供额外的4K字节片上辅助 RAM，称为 XRAM，用于扩展RAM 空间。它占用从 00H 到 FFH的地址空间。这 4K字节XRAM可以通过移动外部指令MOVX@DPTR 或 MOVX@Ri 间接访问。（参见下方的演示代码）请注意，堆栈指针不能位于XRAM的任何位置。

XRAM 演示代码：

汇编程序：

```
MOV R0,#23H          ;将#5AH 写入地址@23H 的 XRAM
MOV A,#5AH
MOVX  @R0,A
MOV R1,#23H          ;从XRAM读取地址@23H
MOVX  A,@R1
MOV DPTR,#0023H      ;将#5BH 写入地址@0023H 的XRAM
MOV A,#5BH
MOVX  @DPTR,A
MOV DPTR,#0023H      ;从XRAM读取地址@0023H
MOVX  A,@DPTR
```

C51:

```
unsigned char temp;       //定义数据变量
unsigned char xdata xtemp _at_ 0x23; //在 xdata 0x23 处定义变量
xtemp = 0x5B;           //将#5BH 写入地址@0023H 的 XRAM
xtemp++;
temp = xtemp;          //从地址@0023H的XRAM读取
```

6.1.7 特殊功能寄存器 (SFR)

CM1003系列使用特殊功能寄存器 (SFR) 来控制和监控外设及其模式。SFR 位于寄存器地址80至FFH，只能通过直接寻址访问。地址以0H或8H结尾的SFR可按位寻址。当用户希望直接修改特定位而不通过位域指令更改其他位时，此功能非常有用。所有其他SFR只能按字节寻址。CM1003系列包含标准 8051中的所有SFR。此外也内置了一些额外的SFR。因此，原始8051中一些未使用的字节被赋予了新的功能。SFR列表如下。

6.1.7.1 SFR页选择

为了在 0x80至0xFF地址空间中容纳超过128个SFR，我们实现了SFR分页。默认情况下，所有SFR访问都指向SFR页0。在设备初始化期间，可能需要访问位于SFR页1/2的某些SFR。寄存器SFRS用于切换特殊功能寄存器寻址页面。

注意：由于中断不可预测，建议在每个中断子程序中添加以下操作：在中断向量程序开始时将SFRS压入堆栈，并在退出中断时弹出SFRS。

SFRS – SFR 页选择

寄存器	特殊功能寄存器地址	复位值
SFRS	91H, 所有SFR页	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	**SFRPAGE1**	**SFRPAGE0**
-	-	-	-	-	-	读/写	读/写

位	名称	描述
1:0	**SFRPAGE[1:0]**	**SFR页选择** 0 = 指令访问 SFR 页0。 1 = 指令访问 SFR 页1。 2 = 指令访问 SFR 页2。

切换SFR页面演示代码：

```
    MOV SFRS,#01H
    MOV SFRS,#00H
```

6.1.7.2 定时访问保护 (TA)

CM1003 系列具有多项功能，例如看门狗定时器WDT和欠压检测，这些功能对于系统的正常运行至关重要。如果这些控制寄存器不受保护，错误的代码可能会将未确定的值写入其中，从而导致操作错误并失控。为了防止这种风险，CM1003系列有保护方案，可以限制对关键 SFR的写入访问。此保护方案通过定时访问 (TA) 实现。以下寄存器与 TA 过程相关。

TA – 定时访问

寄存器	特殊功能寄存器地址	复位值
TA	C7H, 所有SFR页	0000_0000 b

7	6	5	4	3	2	1	0
TA[7:0]							
只写							

位	名称	描述
7:0	TA[7:0]	**定时访问** 定时访问寄存器控制对受保护的SFR的访问。要访问受保护的位，用户应先将AAH写入TA，然后立即将55H写入TA。完成这两个步骤后，将打开一个写入权限窗口，持续4个时钟周期，在此期间用户可以写入受保护的SFR。

在定时访问方法中，受保护的位有写入使能窗口。仅当该窗口有效状态时，写入才能成功，否则写入将被丢弃。当软件往TA写入AAH，计数器开始运行。该计数器等待3个时钟周期查询是否往TA写入55H。如果写入55H在写入AAH后 3个时钟周期内，则定时访问窗口打开。它保持4个时钟周期的打开状态，在此期间用户可以写入受保护的位。4 个时钟周期后，此窗口自动关闭。一旦窗口关闭后，应重复该过程以写入另一个受保护的位。TA保护的SFR，需要定时访问才能写入，但读取不受保护。用户可以读取 TA保护的 SFR，而无需往TA寄存器写入AAH和55H。打开定时访问窗口的建议代码如下所示。

```
(CLR  EA)          ;如果任何中断被使能，则临时禁用
MOV   TA,#0AAH
MOV   TA,#55H
(写入 TA保护寄存器的指令)
(SETB EA)          ;恢复中断启用
```

在执行上述指令过程中，应禁用任何已启用的中断，以避免这三次写入操作之间出现延迟。 如果没有中断使能，则可以省略 CLR EA 和 SETB EA 指令。

以下是定时访问的范例，用以说明正确或错误的写入过程。

示例 1,

```
MOV   TA,#0AAH      ;3个时钟周期
MOV   TA,#55H       ;3个时钟周期
ORL   WDCON,#data   ;4个时钟周期
```

示例 2,

```
MOV   TA,#0AAH      ;3个时钟周期
MOV   TA,#55H       ;3个时钟周期
NOP                 ;1个时钟周期
```

```
ANL   BODCON0,#data   ;4个时钟周期
```

示例 3,

```
MOV   TA,#0AAH        ;3个时钟周期
MOV   TA,#55H         ;3个时钟周期
MOV   WDCON,#data1    ;3个时钟周期
ORL   BODCON0,#data2  ;4个时钟周期
```

示例 4,

```
MOV   TA,#0AAH        ;3个时钟周期
NOP                   ;1个时钟周期
MOV   TA,#55H         ;3个时钟周期
ANL   BODCON0,#data   ;4个时钟周期
```

第1个示例中，对受保护位的写入操作是在3个时钟周期窗口关闭之前完成的；在示例2中，对BODCON0 的写入操作并未在窗口打开期完成，因此BODCON0的值不会改变；在示例3中，WDCON成功写入，但对BODCON0的写入操作超出3个时钟周期窗口，因此BODCON0的值也不会变化；在示例4中，写55H到TA是在写AAH到TA的3个时钟周期之后完成的，因此定时访问窗口根本没有打开，对受保护字节的写入操作不成功。

EIPH – 扩展中断优先级高字节[4]

寄存器	特殊功能寄存器地址	复位值
EIPH	F7H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PT2H	**PSPIH**	**PFBH**	**PWDTH**	**PPWMH**	**PCAPH**	**PPIH**	**PI2CH**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**PT2H**	定时器2中断优先级高位
6	**PSPIH**	SPI中断优先级高位
5	**PFBH**	故障中断优先级高位
4	**PWDTH**	WDT看门狗中断优先级高位
3	**PPWMH**	PWM中断优先级高位
2	**PCAPH**	输入捕获中断优先级高位
1	**PPIH**	引脚中断优先级高位
0	**PI2CH**	I^2C中断优先级高位

[4]**注意：** EIPH 与 EIP 配合使用，用于确定每个中断源的优先级。有关正确的中断优先级配置，请参阅 表6.2-4 中断优先级层级设置。

6.1.7.3 双数据指针

传统8051架构仅有一组DPTR(数据指针)。使用单个 DPTR时，将数据从一个地址移动到另一个地址，程序非常冗长且效率低下。CM1003提供了两组数据指针。这样程序可以同时定义源地址和目标地址，直接进行数据移动。软件只需通过激活数据指针选择DPS(AUXR1.0)位在DPTR和DPTR1之间切换即可。

下面是采用双 DPTR移动64字节块的示例。通过把源地址和目标地址给到数据指针并激活循环，整组RAM数据移动比单DPTR更简单、更高效。 相比ORL或者ANL指令，INC AUXR1 指令是完成 DPTR 切换的最短(2 字节)指令。由于AUXR1.1始终为 0, 通过AUXR1递增来切换 DPS位，不会干扰该寄存器其它位。

```
    MOV   R0,#64          ;要移动的字节数
    MOV   DPTR,#D_Addr    ;加载目标地址
    INC   AUXR1           ;改变激活的DPTR
    MOV   DPTR,#S_Addr    ;加载源地址
  LOOP:
    MOVX  A,@DPTR         ;读取源数据字节
    INC   AUXR1           ;将 DPTR 更改为目标地址
```

```
MOVX  @DPTR,A          ;将数据写入目标
INC   DPTR             ;下一个目标地址
INC   AUXR1            ;将 DPTR 更改为源
INC   DPTR             ;下一个源地址
DJNZ  R0,LOOP
INC   AUXR1            ;(可选)恢复DPS
```

AUXR1的第3位还提供一个通用标志位 GF2，用户可以通过软件对其置位或清除。

DPL – 数据指针低字节

寄存器	特殊功能寄存器地址	复位值
DPL	82H, 所有SFR页	0000_0000b

7	6	5	4	3	2	1	0
DPL[7:0]							
读/写							

位	名称	描述
7:0	DPL[7:0]	**数据指针低字节** 这是16位数据指针的低字节，DPL 与DPH组合构成16位的数据指针DPTR，用于间接寻址访问RAM或程序存储。DPS(AUXR1.0)位决定激活哪个数据指针，DPTR 或 DPTR1。

DPH – 数据指针高字节

寄存器	特殊功能寄存器地址	复位值
DPH	83H, 所有SFR页	0000_0000b

7	6	5	4	3	2	1	0
DPH[7:0]							
读/写							

位	名称	描述
7:0	DPH[7:0]	**数据指针高字节** 这是16位数据指针的高字节，DPH 与 DPL 组合构成16位的数据指针DPTR，用于间接寻址访问RAM或程序存储。DPS(AUXR1.0)位决定激活哪个数据指针，DPTR 或 DPTR1 。

AUXR1 – 辅助寄存器 1

寄存器	特殊功能寄存器地址	复位值
AUXR1	A2H , SFR页0	上电复位: 0000 0000b, 软件复位: 1U00 0000b, 复位脚: U100 0000b, 其他: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	**RSTPINF**	**HardF**	**-**	**GF2**	**UART0PX**	**0**	**DPS**
读/写	读/写	读/写	-	读/写	读/写	只读	读/写

位	名称	描述
3	**GF2**	**通用标志位 2** 用户可通过软件置位或清除该位。
0	**DPS**	**数据指针选择** 0 = 数据指针0(DPTR)默认激活. 1 = 数据指针1(DPTR1)激活. 当通过 DPS 切换激活数据指针后，先前未激活的数据指针的值保持不变。

6.1.7.4 SFR 地址列表

页	地址	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0	F8	SCON_1	PWM0DTEN	PWM0DTCNT	PWM0MEN	PWM0MD	PORDIS	EIP1	EIPH1
1			-	-	-	-	-	-	LVRDIS
2			PIPS0	PIPS1	PIPS2	PIPS3	PIPS4	PIPS5	PIPS6
0	F0	B	CAPCON3	CAPCON4	SPCR	SPSR	SPDR	AINDIDS0	EIPH
1					SPCR2	-	-	-	-
2			SC0CR0	SC0CR1	SC1CR0	SC1CR1	SC2CR0	SC2CR1	PIPS7
0	E8	ADCCON0 P4 P5	PICON	PINEN	PIPEN	PIF	C2L	C2H	EIP
1			-	-	-	-	-	-	-
2			SC2DR	SC2EGT	SC2ETURD0	SC2ETURD1	SC2IE	SC2IS	SC2TSR
0	E0	ACC	ADCCON1	ADCCON2	ADCDLY	C0L	C0H	C1L	C1H
1			-	-	-	-	-	-	-
2			SC1DR	SC1EGT	SC1ETURD0	SC1ETURD1	SC1IE	SC1IS	SC1TSR
0	D8	PWM0CON0	PWM0PL	PWM0C0L	PWM0C1L	PWM0C2L	PWM0C3L	PIOCON0	PWM0CON1
1			-	-	-	-	-	-	-
2			SC0DR	SC0EGT	SC0ETURD0	SC0ETURD1	SC0IE	SC0IS	SC0TSR
0	D0	PSW	PWM0PH	PWM0C0H	PWM0C1H	PWM0C2H	PWM0C3H	PNP	PWM0FBD
1			-	-	-	-	-	-	-
2			PWM3PL	PWM3C0L	PWM3C1L	PWM3CON0	PWM3CON1	PWM3INTC	XTLCON
0	C8	T2CON	T2MOD	RCMP2L	RCMP2H	TL2	TH2	ADCMPL	ADCMPH
1			-	-	-	PWM0C4L	PWM0C5L	-	-
2			PWM3PH	PWM3C0H	PWM3C1H	PWM3MD	PWM3MEN	EIP2	EIPH2
0	C0	I2CON	I2ADDR	ADCRL	ADCRH	T3CON	RL3	RH3	
1			-	-	-	PWM0C4H	PWM0C5H	PIOCON1	TA
2			PWM2PL	PWM2C0L	PWM2C1L	PWM2CON0	PWM2CON1	PWM2INTC	
0	B8	IP	SADEN	SADEN_1	SADDR_1	I2DAT	I2STAT	I2CLK	I2TOC
1			-	-	-	-	-	-	-
2			PWM2PH	PWM2C0H	PWM2C1H	PWM2MD	PWM2MEN	-	-
0	B0	P3	P0M1	P0M2	P1M1	P1M2	TOE	-	IPH
1			P0S	P0SR	P1S	P1SR	-	-	PWM0INTC
2			PWM1PL	PWM1C0L	PWM1C1L	PWM1CON0	PWM1CON1	PWM1INTC	PIOCON2
0	A8	IE	SADDR	WDCON	BODCON1	P3M1	P3M2	IAPFD	IAPCN
1			-	-	-	P3S	P3SR	-	PWM0INTC
2			PWM1PH	PWM1C0H	PWM1C1H	PWM1MD	PWM1MEN	-	-
0	A0	P2	-	AUXR1	BODCON0	IAPTRG	IAPUEN	IAPAL	IAPAH
1			-	-	-	-	-	-	-
2			AUXR2	AUXR3	AUXR4	AUXR5	AUXR6	AUXR7	AUXR8
0	98	SCON	SBUF	SBUF_1	EIE	EIE1	RSR	-	CHPCON
1			P5M1	P5M2	P5SR	P5S	-	-	-
2			AINDIDS1	P0DW	P1DW	P2DW	P3DW	P4DW	P5DW
0	90	P1	SFRS	CAPCON0	CAPCON1	CAPCON2	CKDIV	CKSWT	CKEN
1				-	-	-	-	-	P5UP
2				P0UP	P1UP	P2UP	P3UP	P4UP	RWKH
0	88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	WKCON
1			P4M1	P4M2	P4SR	P4S	-	-	-
2			P2M1	P2M2	P2SR	P2S	ADCSN	ADCCN	ADCSR
0	80	P0	SP	DPL	DPH	RCTRIM0	RCTRIM1	RWKL	
1									PCON
2						ADCBAL	ADCBAH	ADCCON3	

注意：

1. SFR空间中未使用地址用 '-' 标记，为未来功能保留。访问这些区域会有不可预知的影响，应避免访问。
2. 在 ▭ 内的字节表示可位寻址的SFR，可全页访问。
3. 在 ▭ 内的字节表示此 SFR 可全页访问。

表 6.1-1 特殊功能寄存器地址列表

6.1.7.5 SFR 描述和复位值

寄存器	定义	地址	区	MSB 7	6	5	4	3	2	1	LSB[1] 0	复位值[2]	TA
EIPH1	扩展中断优先级1高字节	FFH	0	-	-	PPWM3H	PPWM2H	PPWM1H	PWKTH	PT3H	PSH_1	0000 0000b	
LVRDIS	LVR 禁用	FFH	1	LVRDIS[7:0]								0000 0000b	是
PIPS6	引脚中断控制6	FFH	2	-	BSEL2	BSEL1	BSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
-	-	FEH	0	-	-	-	-	-	-	-	-	-	
EIP1	扩展中断优先级1	FEH	1	-	-	PPWM3	PPWM2	PPWM1	PWKT	PT3	PS_1	0000 0000b	
PIPS5	引脚中断控制5	FEH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PORDIS	POR 禁用	FDH	0	PORDIS[7:0]								0000 0000b	是
-	-	FDH	1	-								-	-
PIPS4	引脚中断控制4	FDH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0MD	PWM0 掩码数据	FCH	0	-	-	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0	0000 0000b	
-	-	FCH	1	-	-	-	-	-	-	-	-	-	
PIPS3	引脚中断控制3	FCH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0MEN	PWM0 掩码使能	FBH	0	-	-	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0	0000 0000b	
-	-	FBH	1	-	-	-	-	-	-	-	-	-	
PIPS2	引脚中断控制2	FBH	2	-	PSEL2	PSEL1	PSEL0		BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0DTCNT	PWM0 死区时间计数器	FAH	0	PWM0DTCNT[7:0]								0000 0000b	是
-	-	FAH	1	-	-	-	-	-	-	-	-	-	
PIPS1	引脚中断控制1	FAH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0DTEN	PWM0 死区时间使能	F9H	0	-	-	-	PWM0DTCNT.8	-	PDT45EN	PDT23EN	PDT01EN	0000 0000b	是
-	-	F9H	1	-	-	-	-	-	-	-	-	-	
PIPS0	引脚中断控制 0	F9H	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
SCON_1	串口1控制	F8H	A	SM0_1 FE_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1	0000 0000b	
EIPH	扩展中断优先级高位	F7H	0	PT2H	PSPIH	PFBH	PWDTH	PPWMH	PCAPH	PPIH	PI2CH	0000 0000b	
-	-	F7H	1	-	-	-	-	-				-	-
PIPS7	引脚中断控制7	F7H	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
AINDIDS0	ADC通道数字输入禁用	F6H	0	P11DIDS	P03DIDS	P04DIDS	P05DIDS	P06DIDS	P07DIDS	P30DIDS	P17DIDS	0000 0000b	
-	-	F6H	1	-	-	-	-	-	-	-	-	-	
SC2CR1	SC2控制寄存器 1	F6H	2	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN		
SPDR	SPI 数据	F5H	0	SPDR[7:0]								0000 0000b	
-	-	F5H	1	-								-	-
SC2CR0	SC2 控制寄存器 0	F5H	2	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	0000 0000b	
SPSR	SPI 状态	F4H	0	SPIF	WCOL	SPIOVF	MODF	DISMODF	TXBUF	-	-	0000 0000b	
-	-	F4H	1	-	-	-	-	-	-	-	-	-	
SC1CR1	SC1 控制寄存器 1	F4H	2	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN	0000 0000b	
SPCR	SPI 控制	F3H	0	SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0	0000 0000b	
SPCR2	SPI 控制2	F3H	1	-	-	-	-	-	-	SPIS1	SPIS0	0000 0000b	
SC1CR0	SC1 控制寄存器 0	F3H	2	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	0000 0000b	
CAPCON4	输入捕获控制4	F2H	0	-	-	-	-	CAP23	CAP22	CAP21	CAP20	0000 0000b	
-	-	F2H	1	-	-	-	-	-	-	-	-	-	
SC0CR1	SC0 控制寄存器 1	F2H	2	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN	0000 0000b	
CAPCON3	输入捕获控制3	F1H	0	CAP13	CAP12	CAP11	CAP10	CAP03	CAP02	CAP01	CAP00	0000 0000b	
-	-	F1H	1	-	-	-	-	-	-	-	-	-	
SC0CR0	SC0 控制寄存器 0	F1H	2	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	0000 0000b	
B	B 寄存器	F0H	A	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	0000 0000b	
EIP	扩展中断优先级	EFH	0	PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI2C	0000 0000b	
-	-	EFH	1	-								-	-
SC2TSR	SC2 发送状态寄存器	EFH	2	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	0000 1010b	
C2H	输入捕获2高字节	EEH	0	C2H[7:0]								0000 0000b	
-	-	EEH	1	-	-	-	-	-	-	-	-	-	
SC2IS	SC2 中断状态寄存器	EEH	2	-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF	0000 0010b	
C2L	输入捕获2低字节	EDH	0	C2L[7:0]								0000 0000b	
-	-	EDH	1	-	-	-	-	-	-	-	-	-	
SC2IE	SC2 中断使能控制寄存器	EDH	2	-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN	0000 0000b	
PIF	引脚中断标志	ECH	0	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	0000 0000b	
-	-	ECH	1	-	-	-	-	-	-	-	-	-	

寄存器	定义	地址	区	MSB 7	6	5	4	3	2	1	LSB[1] 0	复位值[2]	TA
SC2ETURD1	SC2 ETU速率分频器寄存器	ECH	2	-	SCDIV2	SCDIV1	SCDIV0	ETURDIV11	ETURDIV10	ETURDIV9	ETURDIV8	0000 0000b	
PIPEN	引脚中断高电平/上升沿使能	EBH	0	PIPEN7	PIPEN6	PIPEN5	PIPEN4	PIPEN3	PIPEN2	PIPEN1	PIPEN0	0000 0000b	
-	-	EBH	1	-	-	-	-	-	-	-	-	-	
SC2ETURD0	SC2 ETU速率分频器寄存器	EBH	2	ETURDIV[7:0]								0111 0011b	
PINEN	引脚中断低电平/下降沿使能	EAH	0	PINEN7	PINEN6	PINEN5	PINEN4	PINEN3	PINEN2	PINEN1	PINEN0	0000 0000b	
-	-	EAH	1	-	-	-	-	-	-	-	-	-	
SC2EGT	SC2 额外保护时间寄存器	EAH	2	SC2EGT[7:0]								0000 0000b	
PICON	引脚中断控制	E9H	0	PIT7	PIT6	PIT5	PIT4	PIT3	PIT2	PIT1	PIT0	0000 0000b	
-	-	E9H	1	-	-	-	-	-	-	-	-	-	
SC2DR	SC2 数据寄存器	E9H	2	SC2DR[7:0]								0000 0000b	
ADCCON0	ADC 控制0	E8H	0	ADCF	ADCS	ETGSEL1	ETGSEL0	ADCHS3	ADCHS2	ADCHS1	ADCHS0	0000 0000b	
P4	端口4	E8H	1	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	输出锁存, 1111 1111b 输入 XXXX XXXXb	
P5	端口5	E8H	2	P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0	输出锁存, 1111 1111b 输入 XXXX XXXXb	
C1H	输入捕获1高字节	E7H	0	C1H[7:0]								0000 0000b	
-	-	E7H	1	-	-	-	-	-	-	-	-	-	
SC1TSR	SC1发送状态寄存器	E7H	2	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	0000 1010b	
C1L	输入捕获1低字节	E6H	0	C1L[7:0]								0000 0000b	
-	-	E6H	1	-	-	-	-	-	-	-	-	-	
SC1IS	SC1中断状态寄存器	E6H	2	-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF	0000 0010b	
C0H	输入捕获0高字节	E5H	0	C0H[7:0]								0000 0000b	
-	-	E5H	1	-	-	-	-	-	-	-	-	-	
SC1IE	SC1 中断使能控制寄存器	E5H	2	-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN	0000 0000b	
C0L	输入捕获0低字节	E4H	0	C0L[7:0]								0000 0000b	
-	-	E4H	1	-	-	-	-	-	-	-	-	-	
SC1ETURD1	SC1 ETU速率分频器寄存器	E4H	2	-	SCDIV2	SCDIV1	SCDIV0	ETURDIV11	ETURDIV10	ETURDIV9	ETURDIV8	0011 0001b	
ADCDLY	ADC触发延迟	E3H	0	ADCDLY[7:0]								0000 0000b	
-	-	E3H	1	-	-	-	-	-	-	-	-	-	
SC1ETURD0	SC1 ETU速率分频器寄存器	E3H	2	ETURDIV[7:0]								0111 0011b	
ADCCON2	ADC控制2	E2H	0	ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	ADCAQT1.2	ADCAQT1.1	ADCAQT1.0	ADCDLY.8	0000 0000b	
-	-	E2H	1	-	-	-	-	-	-	-	-	-	
SC1EGT	SC1 额外保护时间寄存器	E2H	2	SC1EGT[7:0]								0000 0000b	
ADCCON1	ADC 控制1	E1H	0	OCEN	STADCPX	ADCDIV1	ADCDIV0	ETGTYP1	ETGTYP0	ADCEX	ADCEN	0000 0000b	
-	-	E1H	1	-	-	-	-	-	-	-	-	-	
SC1DR	SC1 数据寄存器	E1H	2	SC1DR[7:0]								0000 0000b	
ACC	累加器	E0H	A	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	0000 0000b	
PWM0CON1	PWM0控制1	DFH	0	PWMMOD1	PWMMOD0	GP	PWMTYP	FBINEN	PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
-	-	DFH	1	-	-	-	-	-	-	-	-	-	
SC0TSR	SC0 发送状态寄存器	DFH	2	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	0000 1010b	
PIOCON0	PWM0 I/O 开关0	DEH	0	-	-	PIO03	PIO01	PIO00	PIO10	PIO11	PIO12	0000 0000b	
-	-	DEH	1	-	-	-	-	-	-	-	-	-	
SC0IS	SC0 中断状态寄存器	DEH	2	-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF	0000 0010b	
PWM0C3L	PWM0通道3占空比低字节	DDH	0	PWM0_CH3[7:0]								0000 0000b	
-	-	DDH	1	-	-	-	-	-	-	-	-	-	
SC0IE	SC0中断使能控制寄存器	DDH	2	-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN	0000 0000b	
PWM0C2L	PWM0通道2占空比低字节	DCH	0	PWM0_CH2[7:0]								0000 0000b	
-	-	DCH	1	-	-	-	-	-	-	-	-	-	
SC0ETURD1	SC0 ETU速率分频器寄存器	DCH	2	-	SCDIV2	SCDIV1	SCDIV0	ETURDIV11	ETURDIV10	ETURDIV9	ETURDIV8	0011 0001b	

寄存器	定义	地址	区	MSB 7	6	5	4	3	2	1	LSB[1] 0	复位值[2]	TA
PWM0C1L	PWM0通道1占空比低字节	DBH	0	PWM0_CH1[7:0]								0000 0000b	
-	-	DBH	1	-	-	-	-	-	-	-	-	-	
SC0ETURD0	SC0 ETU速率分频器寄存器	DBH	2	ETURDIV[7:0]								0111 0011b	
PWM0C0L	PWM0通道0占空比低字节	DAH	0	PWM0C0[7:0]								0000 0000b	
-	-	DAH	1	-	-	-	-	-	-	-	-	-	
SC0EGT	SC0Cn0 额外保护时间寄存器	DAH	2	SC0EGT[7:0]								0000 0000b	
PWM0PL	PWM0周期低字节	D9H	0	PWM0P[7:0]								0000 0000b	
-	-	D9H	1	-	-	-	-	-	-	-	-	-	
SC0DR	SC0 数据寄存器	D9H	2	SC0DR[7:0]								0000 0000b	
PWM0CON0	PWM0 控制寄存器 0	D8H	A	PWM0RUN	LOAD	PWMF	CLRPWM	PWM3RUN	PWM2RUN	PWM1RUN	P33FBINEN	0000 0000b	
PWM0FBD	刹车数据	D7H	0	FBF	FBINLS	FBD5	FBD4	FBD3	FBD2	FBD1	读/写	0000 0000b	
-	-	D7H	1	-	-	-	-	-	-	-	-	-	
XTLCON	XTL时钟控制	D7H	2	-	HXSG6	HXSG5	HXSG4	HXSG3	HXSG2	HXSG1	HXSG0	0010 1000b	是
PNP	PWM0 负极性	D6H	0	-	-	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0	0000 0000b	
-	-	D6H	1	-	-	-	-	-	-	-	-	-	
PWM3INTC	PWM3 中断控制	D6H	2	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
PWM0C3H	PWM0通道3占空比高字节	D5H	0	PWM0_CH3[15:8]								0000 0000b	
-	-	D5H	1	-	-	-	-	-	-	-	-	-	
PWM3CON1	PWM3 控制1	D5H	2	PWMMOD1	PWMMOD0	PWMTYP	-		PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
PWM0C2H	PWM0通道2占空比高字节	D4H	0	PWM0_CH2[15:8]								0000 0000b	
-	-	D4H	1	-	-	-	-	-	-	-	-	-	
PWM3CON0	PWM3 控制0	D4H	2	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000b	
PWM0C1H	PWM0通道1占空比高字节	D3H	0	PWM0_CH1[15:8]								0000 0000b	
-	-	D3H	1	-	-	-	-	-	-	-	-	-	
PWM3C1L	PWM3通道1占空比低字节	D3H	2	PWM3_CH1[7:0]								0000 0000b	
PWM0C0H	PWM0通道0占空比高字节	D2H	0	PWM0C0[15:8]								0000 0000b	
-	-	D2H	1	-	-	-	-	-	-	-	-	-	
PWM3C0L	PWM3通道0占空比低字节	D2H	2	PWM3_CH0[7:0]								0000 0000b	
PWM0PH	PWM0周期高字节	D1H	0	PWM0P[15:8]								0000 0000b	
-	-	D1H	1	-	-	-	-	-	-	-	-	-	
PWM3PL	PWM3周期低字节	D1H	2	PWM3P[7:0]								0000 0000b	
PSW	程序状态字	D0H	A	CY	AC	F0	RS1	RS0	OV	-	P	0000 0000b	
ADCMPH	ADC比较高字节	CFH	0	ADCMP[11:4]								0000 0000b	
-	-	CFH	1	-	-	-	-	-	-	-	-	-	
EIPH2	扩展中断优先级2高字节	CFH	2	-	-	-	-	-	PUART4H	PUART3H	PUART2H	0000 0000b	
ADCMPL	ADC 比较低字节	CEH	0	-	-	-	-	ADCMP3	ADCMP2	ADCMP1	ADCMP0	0000 0000b	
-	-	CEH	1	-	-	-	-	-	-	-	-	-	
EIP2	扩展中断优先级2	CEH	2	-	-	-	-	-	PUART4	PUART3	PUART2	0000 0000b	
TH2	定时器2 高字节	CDH	0	TH2[7:0]								0000 0000b	
PWM0C5L	PWM0通道5占空比低字节	CDH	1	PWM0_CH5[7:0]								0000 0000b	
PWM3MEN	PWM3 掩码使能	CDH	2	-	-	-	-	-	-	PMEN1	PMEN0	0000 0000b	
TL2	定时器2低字节	CCH	0	TL2[7:0]								0000 0000b	
PWM0C4L	PWM0通道4占空比低字节	CCH	1	PWM0_CH4[7:0]								0000 0000b	
PWM3MD	PWM3掩码使能	CCH	2	-	-	-	-	-	-	PMD1	PMD0	0000 0000b	
RCMP2H	定时器2比较高字节	CBH	0	RCMP2H[7:0]								0000 0000b	
-	-	CBH	1	-	-	-	-	-	-	-	-	-	
PWM3C1H	PWM3通道1占空比高字节	CBH	2	PWM3_CH1[15:8]								0000 0000b	

寄存器	定义	地址	区	MSB 7	6	5	4	3	2	1	LSB[1] 0	复位值[2]	TA
RCMP2L	定时器2比较低字节	CAH	0	RCMP2L[7:0]								0000 0000b	
-	-	CAH	1	-	-	-	-	-	-	-	-	-	
PWM3C0H	PWM3通道0占空比高字节	CAH	2	PWM3_CH0[15:8]								0000 0000b	
T2MOD	定时器2模式	C9H	0	LDEN	T2DIV2	T2DIV1	T2DIV0	CAPCR	CMPCR	LDTS1	LDTS0	0000 0000b	
-	-	C9H	1	-	-	-	-	-	-	-	-	-	
PWM3PH	PWM3周期高字节	C9H	2	PWM3P[15:8]								0000 0000b	
T2CON	定时器2控制	C8H	A	TF2	-	-	-	-	TR2	-	CM/RL2	0000 0000b	
TA	定时访问保护	C7H	A	TA[7:0]								0000 0000b	
RH3	定时器3重载高字节	C6H	0	RH3[7:0]								0000 0000b	
PIOCON1	PWM0 I/O 开关1	C6H	1	PIO17	-	PIO15	-	PIO04	PIO05	PIO14	-	-	
PWM2INTC	PWM2中断控制	C6H	2	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
RL3	定时器3重载低字节	C5H	0	RL3[7:0]								0000 0000b	
PWM0C5H	PWM0通道5占空比高字节	C5H	1	PWM0_CH5[15:8]								0000 0000b	
PWM2CON1	PWM2控制1	C5H	2	PWMMOD1	PWMMOD0	PWMTYP	-		PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
T3CON	定时器3控制	C4H	0	SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS2	T3PS1	T3PS0	0000 0000b -	
PWM0C4H	PWM0通道4占空比高字节	C4H	1	PWM0_CH4[15:8]								0000 0000b	
PWM2CON0	PWM2控制1	C4H	2	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000b	
ADCRH	ADC结果高字节	C3H	0	ADCR[11:4]								0000 0000b	
-	-	C3H	1	-	-	-	-	-	-	-	-	-	
PWM2C1L	PWM2通道1占空比低字节	C3H	2	PWM2_CH1[7:0]								0000 0000b	
ADCRL	ADC结果低字节	C2H	0	-	-	-	-	ADCR3	ADCR2	ADCR1	ADCR0	0000 0000b	
-	-	C2H	1	-	-	-	-	-	-	-	-	-	
PWM2C0L	PWM2通道0占空比低字节	C2H	2	PWM2_CH0[7:0]								0000 0000b	
I2ADDR	I²C从机地址	C1H	0	I2ADDR7	I2ADDR6	I2ADDR5	I2ADDR4	I2ADDR3	I2ADDR2	I2ADDR1	GC	0000 0000b	
-	-	C1H	1	-	-	-	-	-	-	-	-	-	
PWM2PL	PWM2周期低字节	C1H	2	PWM2P[7:0]								0000 0000b	
I2CON	I²C控制	C0H	A	I2CHTEX	I2CEN	STA	STO	SI	AA	-	I2CPX	0000 0000b	
I2TOC	I²C超时计数器	BFH	0	-	-	-	-	-	I2TOCEN	DIV	I2TOF	0000 0000b	
-	-	BFH	1	-	-	-	-	-	-	-	-	-	
PWM2MEN	PWM2掩码使能	BFH	2	-	-	-	-	-	-	PMEN1	PMEN0	0000 0000b	
I2CLK	I²C时钟	BEH	0	I2CLK[7:0]								0000 1001b	
-	-	BEH	1	-	-	-	-	-	-	-	-	-	
PWM2MD	PWM2掩码数据	BEH	2	-	-	-	-	-	-	PMD1	PMD0	0000 0000b	
I2STAT	I²C状态	BDH	0	I2STAT7	I2STAT6	I2STAT5	I2STAT4	I2STAT3	0	0	0	1111 1000b	
-	-	BDH	1	-	-	-	-	-	-	-	-	-	
PWM2C1H	PWM2通道1占空比高字节	BDH	2	PWM2_CH1[15:8]								0000 0000b	
I2DAT	I²C数据	BCH	0	I2DAT[7:0]								0000 0000b	
-	-	BCH	1	-	-	-	-	-	-	-	-	-	
PWM2C0H	PWM2通道0占空比高字节	BCH	2	PWM2_CH015:8]								0000 0000b	
SADDR_1	从机1地址	BBH	0	SADDR_1[7:0]								0000 0000b	
-	-	BBH	1	-	-	-	-	-	-	-	-	-	
PWM2PH	PWM2周期高字节	BBH	2	PWM2P[15:8]								0000 0000b	
SADEN_1	从机1地址掩码	BAH	0	SADEN_1[7:0]								0000 0000b	
-	-	BAH	1	-	-	-	-	-	-	-	-	-	
-	-	BAH	2	-	-	-	-	-	-	-	-	-	
SADEN	从机0地址掩码	B9H	0	SADEN[7:0]								0000 0000b	
-	-	B9H	1	-	-	-	-	-	-	-	-	-	
-	-	B9H	2	-	-	-	-	-	-	-	-	-	

寄存器	定义	地址	页	MSB 7	6	5	4	3	2	1	LSB[1] 0	复位值[2]	TA
IP	中断优先级	B8H	A	-	PADC	PBOD	PS	PT1	PX1	PT0	PX0	0000 0000b	
IPH	中断优先级高字节	B7H	0	-	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H	0000 0000b	
-	-	B7H	1	-	-	-	-	-	-	-	-	-	
PIOCON2	PWM或I/O选择2	B7H	2	PIO34	PIO33	PIO32	PIO31	-	PIO23	PIO22	PIO21	0000 0000b	
-	-	B6H	0	-	-	-	-	-	-	-	-	-	
-	-	B6H	1	-	-	-	-	-	-	-	-	-	
PWM1INTC	PWM1中断控制	B6H	2	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
TOE	Timer01输出使能	B5H	0	-	-	-	-	T1OE	T0OE	-	-	0000 0000b	
-	-	B5H	1	-	-	-	-	-	-	-	-	-	
PWM1CON1	PWM1控制1	B5H	2	PWMMOD1	PWMMOD0	-	PWMTYP	-	PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
P1M2	P1模式选择2	B4H	0	P1M2.7	P1M2.6	P1M2.5	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0	0000 0000b	
P1SR	P1转换速率	B4H	1	P1SR.7	P1SR.6	P1SR.5	P1SR.4	P1SR.3	P1SR.2	P1SR.1	P1SR.0	0000 0000b	
PWM1CON0	PWM1控制0	B4H	2	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000b	
P1M1	P1模式选择1	B3H	0	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	1111 1111b	
P1S	P1施密特触发器输入	B3H	1	P1S.7	P1S.6	P1S.5	P1S.4	P1S.3	P1S.2	P1S.1	P1S.0	0000 0000b	
PWM1C1L	PWM1通道1占空比低字节	B3H	2	PWM1_CH1[7:0]								0000 0000b	
P0M2	P0模式选择2	B2H	0	P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0	0000 0000b	
P0SR	P0转换速率	B2H	1	P0SR.7	P0SR.6	P0SR.5	P0SR.4	P0SR.3	P0SR.2	P0SR.1	P0SR.0	0000 0000b	
PWM1C0L	PWM1通道0占空比低字节	B2H	2	PWM1_CH07:0]								0000 0000b	
P0M1	P0模式选择1	B1H	0	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0	1111 1111b	
P0S	P0施密特触发器输入	B1H	1	P0S.7	P0S.6	P0S.5	P0S.4	P0S.3	P0S.2	P0S.1	P0S.0	0000 0000b	
PWM1PL	PWM1周期低字节	B1H	2	PWM1P[7:0]								0000 0000b	
P3	端口3	B0H	A	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	输出锁存, 1111 1111b 输入 xxxx xxxxb[3]	
IAPCN	IAP控制	AFH	0	IAPA17	IAPA16	FOEN	FCEN	FCTRL3	FCTRL2	FCTRL1	FCTRL0	0011 0000b	
PWM0INTC	PWM0中断控制	AFH	1	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
-	-	AFH	2	-	-	-	-	-	-	-	-	-	
IAPFD	IAP flash数据	AEH	0	IAPFD[7:0]								0000 0000b	
-	-	AEH	1	-	-	-	-	-	-	-	-	-	
PWM1MEN	PWM1掩码使能	AEH	2	-	-	-	-	-	-	PMEN1	PMEN0	0000 0000b	
P3M2	P3模式选择2	ADH	0	P3M2.7	P3M2.6	P3M2.5	P3M2.4	P3M2.3	P3M2.2	P3M2.1	P3M2.0	0000 0000b	
P3SR	P3转换速率	ADH	1	P3SR.7	P3SR.6	P3SR.5	P3SR.4	P3SR.3	P3SR.2	P3SR.1	P3SR.0	0000 0000b	
PWM1MD	PWM1掩码数据	ADH	2	-	-	-	-	-	-	PMD1	PMD0	0000 0000b	
P3M1	P3 模式选择1	ACH	0	P3M1.7	P3M1.6	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0	1111 1111b	
P3S	P3施密特触发器输入	ACH	1	P3S.7	P3S.6	P3S.5	P3S.4	P3S.3	P3S.2	P3S.1	P3S.0	0000 0000b	
PWM1C1H	PWM1通道1占空比高字节	ACH	2	PWM1_CH1[15:8]								0000 0000b	
BODCON1	欠压检测控制1	ABH	0	-	-	-	-	-	LPBOD1	LPBOD0	BODFLT	POR上电复位, 0000 0001b 其他, 0000 0UUUb	是
-	-	ABH	1	-	-	-	-	-	-	-	-	-	
PWM1C0H	PWM1通道0占空比高字节	ABH	2	PWM1_CH015:8]								0000 0000b	
WDCON	看门狗定时器控制	AAH	0	WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS2	WDPS1	WDPS0	POR上电复位, 0000 0111b WDT看门狗, 0000 1UUUb 其他, 0000 UUUUb	是
-	-	AAH	1	-	-	-	-	-	-	-	-	-	
PWM1PH	PWM1周期高字节	AAH	2	PWM1P[15:8]								0000 0000b	
SADDR	从机0 地址	A9H	0	SADDR[7:0]								0000 0000b	
-	-	A9H	1	-	-	-	-	-	-	-	-	-	
PWM1PH	PWM1周期高字节	A9H	2	PWM1P[15:8]								0000 0000b	

寄存器	定义	地址	区	MSB 7	6	5	4	3	2	1	LSB[1] 0	复位值[2]	TA
IE	中断使能	A8H	A	EA	EADC	EBOD	ES	ET1	EX1	ET0	EX0	0000 0000b	
IAPAH	IAP 地址高字节	A7H	0	IAPA[15:8]								0000 0000b	
-	-	A7H	1	-	-	-	-	-	-	-	-	-	
AUXR8	辅助寄存器 8	A7H	2	CLODIV3	CLODIV2	CLODIV1	CLODIV0	CKTESTOEN.3	CKTESTOEN.2	CKTESTOEN.1	CKTESTOEN.0	0000 0000b	是
IAPAL	IAP地址低字节	A6H	0	IAPA[7:0]								0000 0000b	
-	-	A6H	1	-	-	-	-	-	-	-	-	-	
AUXR7	辅助寄存器 7	A6H	2	-	-	-	SPI0NSSP1	SPI0NSSP0	SPI0MOSIP	SPI0MISOP	SPI0CKP	0000 0000b	
IAPUEN	IAP更新使能	A5H	0	-		-	SPMEN	SPUEN	CFUEN	LDUEN	APUEN	0000 0000b	是
-	-	A5H	1	-	-	-	-	-	-	-	-	-	
AUXR6	辅助寄存器 6	A5H	2	-	-	-	UART4DG	UART3DG	UART2DG	UART1DG	UART0DG	0000 0000b	
IAPTRG	IAP触发器	A4H	0	-		-	-	-	-	-	IAPGO	0000 0000b	是
-	-	A4H	1	-	-	-	-	-	-	-	-	-	
AUXR5	辅助寄存器 5	A4H	2	CLOP	T0P			PWM3CH1P1	PWM3CH1P0	PWM3CH0P1	PWM3CH0P0	0000 0000b	
BODCON0	欠压侦测控制0	A3H	0	BODEN	-	BOV1	BOV0	BOF	BORST	BORF	BOS	POR上电复位, CCCC XC0Xb BOD欠压检测, UUUU XU1Xb 其他, UUUU XUUXb	是
-	-	A3H	1	-	-	-	-	-	-	-	-	-	
AUXR4	辅助寄存器 4	A3H	2	PWM2CH1P1	PWM2CH1P0	PWM2CH0P1	PWM2CH0P0	PWM1CH1P1	PWM1CH1P0	PWM1CH0P1	PWM1CH0P0	0000 0000b	
AUXR1	辅助寄存器 1	A2H	0	SWRF	RSTPINF	HardF	-	GF2	UART0PX	0	DPS	POR上电复位, 0000 0000b 软件复位:, 1U00 0000b nRESET pin, U100 0000b 其他, UUU0 0000b	
-	-	A2H	1	-	-	-	-	-	-	-	-	-	
AUXR3	辅助寄存器 3	A2H	2	UART4TXP1	UART4TXP0	UART4RXP1	UART4RXP0	UART3TXP1	UART3TXP0	UART3RXP1	UART3RXP0	0000 0000b	
-	-	A1H	0	-	-	-	-	-	-	-	-	-	
-	-	A1H	1	-	-	-	-	-	-	-	-	-	
AUXR2	辅助寄存器 2	A1H	2	UART2TXP1	UART2TXP0	UART2RXP1	UART2RXP0	UART1TXP1	UART1TXP0	UART1RXP1	UART1RXP0	0000 0000b	
P2	端口2	A0H	A	-	-	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	输出锁存, 0000 000Xb 输入, 0000 000Xb	
CHPCON	芯片控制	9FH	0	SWRST	IAPFF	-	-	-	-	BS[5]	IAPEN	软件, 0000 00U0b 其他, 0000 00C0b	是
-	-	9FH	1	-	-	-	-	-	-	-	-	-	
P5DW	端口5下拉电阻控制	9FH	2	P5DW.7	P5DW.6	P5DW.5	P5DW.4	P5DW.3	P5DW.2	P5DW.1	P5DW.0	0000 0000b	
-	-	9EH	0	-	-			-	-	-	-	-	-
-	-	9EH	1	-	-	-	-	-	-	-	-	-	
P4DW	端口4下拉电阻控制	9EH	2	P4DW.7	P4DW.6	P4DW.5	P4DW.4	P4DW.3	P4DW.2	P4DW.1	P4DW.0	0000 0000b	
RSR	复位标志位寄存器	9DH	0	-	-	HardF(从AUXR1.5镜像)	POF (从PCON.4镜像)	RSTPINF (从AUXR1.6镜像)	BORF (从BODCON0.1镜像)	WDTRF (从WDCON.3镜像)	SWRF (从AUXR1.7镜像)	00XX XXXXb	
-	-	9DH	1	-	-	-	-	-	-	-	-	-	
P3DW	端口3下拉电阻控制	9DH	2	P3DW.7	P3DW.6	P3DW.5	P3DW.4	P3DW.3	P3DW.2	P3DW.1	P3DW.0	0000 0000b	
EIE1	扩展中断使能1	9CH	0	-	-	EPWM3	EPWM2	EPWM1	EWKT	ET3	ES_1	0000 0000b	
P5S	端口5施密特触发器输入	9CH	1	P5S.7	P5S.6	P5S.5	P5S.4	P5S.3	P5S.2	P5S.1	P5S.0	0000 0000b	
P2DW	端口2下拉电阻控制	9CH	2	-	-	P2DW.5	P2DW.4	P2DW.3	P2DW.2	P2DW.1	P2DW.0	0000 0000b	
EIE	扩展中断使能	9BH	0	ET2	ESPI	EFB	EWDT	EPWM0	ECAP	EPI	EI2C	0000 0000b	
P5SR	端口5转换率控制	9BH	1	P5SR.7	P5SR.6	P5SR.5	P5SR.4	P5SR.3	P5SR.2	P5SR.1	P5SR.0	0000 0000b	
P1DW	端口1下拉电阻控制	9BH	2	P1DW.7	P1DW.6	P1DW.5	P1DW.4	P1DW.3	P1DW.2	P1DW.1	P1DW.0	0000 0000b	
SBUF_1	串口1数据缓冲	9AH	0	SBUF_1[7:0]								0000 0000b	
P5M2	P5模式选择2	9AH	1	P5M2.7	P5M2.6	P5M2.5	P5M2.4	P5M2.3	P5M2.2	P5M2.1	P5M2.0	0000 0000b	
P0DW	端口0下拉电阻控制	9AH	2	P0DW.7	P0DW.6	P0DW.5	P0DW.4	P0DW.3	P0DW.2	P0DW.1	P0DW.0	0000 0000b	

寄存器	定义	地址	区	MSB 7	6	5	4	3	2	1	LSB[1] 0	复位值[2]	TA
SBUF	串口0数据缓冲	99H	0	SBUF[7:0]								0000 0000b	
P5M1	P5模式选择1	99H	1	P5M1.7	P5M1.6	P5M1.5	P5M1.4	P5M1.3	P5M1.2	P5M1.1	P5M1.0	0000 0000b	
AINDIDS1	ADC通道数字输入禁用	99H	2	P25DIDS	P14DIDS	P13DIDS	P24DIDS	P23DIDS	P22DIDS	P21DIDS	-	0000 0000b	
SCON	串口0控制	98H	A	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000b	
CKEN	时钟使能	97H	0	EXTEN1	EXTEN0	HIRCEN	LIRCEN	-	-	-	CKSWTF	0011 0000b	是
P5UP	端口5上拉电阻控制	97H	1	P5UP.7	P5UP.6	P5UP.5	P5UP.4	P5UP.3	P5UP.2	P5UP.1	P5UP.0	0000 0000b	
RWKH	自唤醒定时器重载高字节	97H	2	RWK[15:8]								0000 0000b	
CKSWT	时钟开关	96H	0	HXTST	ECKP00ST	HIRCST	-	ECKP30ST	OSC1	OSC0	-	0011 0000b	是
-	-	96H	1	-	-	-	-	-	-	-	-	-	
P4UP	端口4上拉电阻控制	96H	2	P4UP.7	P4UP.6	P4UP.5	P4UP.4	P4UP.3	P4UP.2	P4UP.1	P4UP.0	0000 0000b	
CKDIV	时钟分频器	95H	0	CKDIV[7:0]								0000 0000b	
-	-	95H	1	-	-	-	-	-	-	-	-	-	
P3UP	端口3 上拉电阻控制	95H	2	P3UP.7	P3UP.6	P3UP.5	P3UP.4	P3UP.3	P3UP.2	P3UP.1	P3UP.0	0000 0000b	
CAPCON2	输入捕获控制2	94H	0	-	ENF2	ENF1	ENF0	-	-	-	-	0000 0000b	
-	-	94H	1	-	-	-	-	-	-	-	-	-	
P2UP	端口2上拉电阻控制	94H	2	-	-	P2UP.5	P2UP.4	P2UP.3	P2UP.2	P2UP.1	P2UP.0	0000 0000b	
CAPCON1	输入捕获控制1	93H	0	-	-	CAP2LS1	CAP2LS0	CAP1LS1	CAP1LS0	CAP0LS1	CAP0LS0	0000 0000b	
-	-	93H	1	-	-	-	-	-	-	-	-	-	
P1UP	端口1上拉电阻控制	93H	2	P1UP.7	P1UP.6	P1UP.5	P1UP.4	P1UP.3	P1UP.2	P1UP.1	P1UP.0	0000 0000b	
CAPCON0	输入捕获控制0	92H	0	-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0	0000 0000b	
-	-	92H	1	-	-	-	-	-	-	-	-	-	
P0UP	端口0上拉电阻控制	92H	2	P0UP.7	P0UP.6	P0UP.5	P0UP.4	P0UP.3	P0UP.2	P0UP.1	P0UP.0	0000 0000b	
SFRS	SFR页选择	91H	A	-	-	-	-	-	-	SFRPSEL1	SFRPSEL0	0000 0000b	
P1	端口1	90H	A	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	输出锁存, 1111 1111b 输入 XXXX XXXXb	
WKCON	自唤醒定时器控制	8FH	0	-	-	-	WKTF	WKTR	WKPS2	WKPS1	WKPS0	0000 0000b	
-	-	8FH	1	-	-	-	-	-	-	-	-	-	
ADCSR	ADC状态寄存器	8FH	2	-	-	-	-	-	CMPHIT	HDONE	FDONE	0000 0000b	
CKCON	时钟控制	8EH	0	-	PWMCKS	-	T1M	T0M	-	CLOEN	-	POR,上电复位 0000 0000b 其他, 0000 00U0b	
-	-	8EH	1	-	-	-	-	-	-	-	-	-	
ADCCN	ADC电流采样数	8EH	2	ADCCN[7:0]								0000 0000b	
TH1	定时器1高字节	8DH	0	TH1[7:0]								0000 0000b	
-	-	8DH	1	-	-	-	-	-	-	-	-	-	
ADCSN	ADC采样数	8DH	2	ADCSN[7:0]								0000 0000b	
TH0	定时器0高字节	8CH	0	TH0[7:0]								0000 0000b	
P4S	端口4施密特触发器输入	8CH	1	P4S.7	P4S.6	P4S.5	P4S.4	P4S.3	P4S.2	P4S.1	P4S.0	0000 0000b	
P2S	端口2施密特触发器输入	8CH	2	P2S.7	P2S.6	P2S.5	P2S.4	P2S.3	P2S.2	P2S.1	P2S.0	0000 0000b	
TL1	定时器1低字节	8BH	0	TL1[7:0]								0000 0000b	
P4SR	端口4转换率控制	8BH	1	P4SR.7	P4SR.6	P4SR.5	P4SR.4	P4SR.3	P4SR.2	P4SR.1	P4SR.0	0000 0000b	
P2SR	端口2 转换率控制	8BH	2	P2SR.7	P2SR.6	P2SR.5	P2SR.4	P2SR.3	P2SR.2	P2SR.1	P2SR.0	0000 0000b	
TL0	定时器0低字节	8AH	0	TL1[7:0]								0000 0000b	
P4M2	P4模式选择2	8AH	1	P4M2.7	P4M2.6	P4M2.5	P4M2.4	P4M2.3	P4M2.2	P4M2.1	P4M2.0	0000 0000b	
P2M2	P2模式选择2	8AH	2	P2M2.7	P2M2.6	P2M2.5	P2M2.4	P2M2.3	P2M2.2	P2M2.1	P2M2.0	0000 0000b	
TMOD	定时器0和1模式	89H	0	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	0000 0000b	
P4M1	P4模式选择1	89H	1	P4M1.7	P4M1.6	P4M1.5	P4M1.4	P4M1.3	P4M1.2	P4M1.1	P4M1.0	0000 0000b	
P2M1	P2模式选择1	89H	2	P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0	0000 0000b	
TCON	定时器0和1控制	88H	A	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000b	
PCON	电源控制	87H	A	SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL	POR上电复位, 0001 0000b 其他, 000U 0000b	

寄存器	定义	地址	区	MSB 7	6	5	4	3	2	1	LSB[1] 0	复位值[2]	TA
RWKL	自唤醒定时器重载低字节	86H	0	RWK[7:0]								0000 0000b	
-	-	86H	1	-	-	-	-	-	-	-	-	-	
ADCCON3	ADC控制3	86H	2	-	-	HIE	CONT	ADCAQT2.2	ADCAQT2.1	ADCAQT2.0	-	0000 0000b	
RCTRIM1	内部RC调整值低字节	85H	0	-		-	HIRC24		-	-	HIRCTRIM.0	0000 0000b	是
-	-	85H	1	-	-	-	-	-	-	-	-	-	
ADCBAH	ADC RAM基地址高字节	85H	2	-	-	-	-	ADCBA3	ADCBA2	ADCBA1	ADCBA0	0000 0000b	
RCTRIM0	内部RC调整值高字节	84H	0	HIRCTRIM[8:1]								0000 0000b	是
-	-	84H	1	-	-	-	-	-	-	-	-	-	
ADCBAL	ADC RAM基地址低字节	84H	2	ADCBA[7:0]								0000 0000b	
DPH	数据指针高字节	83H	A	DPTR[15:8]								0000 0000b	
DPL	数据指针低字节	82H	A	DPTR[7:0]								0000 0000b	
SP	堆栈指针	81H	A	SP[7:0]								0000 0111b	
P0	端口0	80H	A	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	输出锁存, 1111 1111b 输入, XXXX XXXXb	

注意：

1. ( ) 项表示可位寻址特殊功能寄存器 (SFR) 中的位地址。

2. 复位值符号说明。0：逻辑 0；1：逻辑 1；U：保持不变；C：参见 [5]；X：参见 [3]、[6] 和 [7]。

3. 复位之后所有I/O引脚默认为仅输入模式(浮空)。如果 RPD(CONFIG0.2)保持未编程1，则读回 P2.0 始终为 0。复位后ICE_DAT和ICE_CLK引脚将保持准双向带上拉电阻的模式，等待 600个LIRC时钟周期之后才切换到输入模式。

4. 这些 SFR 具有 TA 写保护。

5.这些 SFR 中的某些位在特定的复位后会根据 CONFIG 值进行初始化。

6. BOF 复位值取决于CONFIG2和V_{DD}电压的不同设置。请检查

7. 当欠压检测使能，BOS是只读标志，由V_{DD}值来决定。

表6.1-2 SFR定义和复位值

6.1.7.6 所有SFR描述

注意：

1. 所有 SFR 的复位值分别表示为：U-保持不变；C-由 CONFIG 初始化；X-基于芯片实际状态。.

Pn – 端口 (可位寻址)

寄存器	特殊功能寄存器地址	复位值
P0	80H, 所有SFR页, 可位寻址	1111_1111 b
P1	90H, 所有SFR页, 可位寻址	1111_1111 b
P2	A0H, 所有SFR页, 可位寻址	0011_1111 b
P3	B0H, 所有SFR页, 可位寻址	1111_1111 b
P4	E8H, SFR页1, 可位寻址	1111_1111 b
P5	E8H, SFR页2, 可位寻址	1111_1111 b

7	6	5	4	3	2	1	0
Pn.7	**Pn.6**	**Pn.5**	**Pn.4**	**Pn.3**	**Pn.2**	**Pn.1**	**Pn.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**Pn[7:0]**	**端口n** 端口n是最大8 位通用 I/O 端口。

SP – 堆栈指针

寄存器	特殊功能寄存器地址	复位值
SP	81H, 所有SFR页	0000_0111b

7	6	5	4	3	2	1	0
SP[7:0]							
读/写							

位	名称	描述
7:0	SP[7:0]	**堆栈指针** 堆栈指针存储的是暂存器 RAM 中堆栈起始的地址。在执行 PUSH 或 CALL 指令期间，它会在数据存储之前递增。注意，SP 的默认值为 07H。这意味着堆栈起始于 08H 的位置。

DPL – 数据指针低字节

寄存器	特殊功能寄存器地址	复位值
DPL	82H, 所有SFR页	0000_0000b

7	6	5	4	3	2	1	0
DPL[7:0]							
读/写							

位	名称	描述
7:0	DPL[7:0]	**数据指针低字节** 这是16位数据指针的低字节，DPL 与DPH组合构成16位的数据指针DPTR，用于间接寻址访问RAM或程序存储。DPS(AUXR1.0)位决定激活哪个数据指针，DPTR 或 DPTR1。

DPH – 数据指针高字节

寄存器	特殊功能寄存器地址	复位值
DPH	83H, 所有SFR页	0000_0000b

7	6	5	4	3	2	1	0
DPH[7:0]							
读/写							

位	名称	描述
7:0	DPH[7:0]	**数据指针高字节** 这是16位数据指针的高字节，DPH 与 DPL 组合构成16位的数据指针DPTR，用于间接寻址访问RAM或程序存储。DPS(AUXR1.0)位决定激活哪个数据指针，DPTR 或 DPTR1 。

ADCBAH – ADC RAM 基地址高字节

寄存器	特殊功能寄存器地址	复位值
ADCBAH	84H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-				ADCBAH[3:0]			
-				读/写			

位	名称	描述
7:4	-	保留
3:0	ADCBAH[3:0]	**ADC RAM 基地址 (高字节)** RAM基地址的高4位用于存储ADC连续采样数据。 RAM基地址ADCBA[11:0] = {ADCBAH[3:0], ADCBAL[7:0]}

ADCBAL – ADC RAM基地址低字节

寄存器	特殊功能寄存器地址	复位值
ADCBAL	85H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
ADCBAL[7:0]							
读/写							

位	名称	描述
7:0	**ADCBAL[7:0]**	**ADC RAM基地址(低字节)** RAM基址的最低 8 位用于存储 ADC 连续采样数据。 RAM基地址 ADCBA[11:0] = {ADCBAH[3:0], ADCBAL[7:0]}

RWKL – 自唤醒定时器重新加载低字节

寄存器	特殊功能寄存器地址	复位值
RWKL	86H, SFR页0	0000 0000b

7	6	5	4	3	2	1	0
RWK[7:0]							
读/写							

位	名称	描述
7:0	**RWK[7:0]**	**WKT重载低字节** RWKL 寄存器是 WKT 16 位重载值的低字节。 注意，如果预分频比为 1/1，则 RWK 不应为 FFFFH，这是实现限制。

RWKH – 自唤醒定时器重新加载高字节

寄存器	特殊功能寄存器地址	复位值
RWKH	97H, SFR页2	0000 0000b

7	6	5	4	3	2	1	0
RWK[15:8]							
读/写							

位	名称	描述
7:0	RWK[15:8]	**WKT重载高字节** RWKH 寄存器是 WKT 16 位重载值的高字节。 注意，如果预分频比为 1/1，则 RWK 不应为 FFFFH，这是实现限制。

PCON – 电源控制

寄存器	特殊功能寄存器地址	复位值
PCON	87H, 所有SFR页	上电复位: 0001_000b, 其它: 000U_0000b

7	6	5	4	3	2	1	0
SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL
读/写	读/写	-	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	SMOD	**串口 0 双倍波特率使能** 当 UART0 处于模式 2， 或将定时器1溢出用作 UART0 模式1/3的波特率时钟源时，此位置位可使串口0 波特率加倍。详情请参见 表6. 9-1 串口 UART0 模式/波特率说明。
6	SMOD0	**串口0 帧错误标志访问使能** 0 = SCON.7访问 SM0 位。 1 = SCON.7访问 FE 位。
4	POF	**上电复位标志** 上电复位后，该位将被设置为1。表示冷复位，即上电复位完成。在任何其他复位后，该位将保持其值。建议通过软件清除该标志。
3	GF1	**通用标志1** 通用标志可由用户通过软件置位和清零。
2	GF0	**通用标志0** 通用标志可由用户通过软件置位和清零。
1	PD	**掉电模式** 置位此位将使CPU进入掉电模式。在此模式下，CPU和外设时钟均停止，程序计数器(PC)挂起。此时为最低功耗。CPU从掉电模式唤醒后，此位由硬件自动清除，程序继续执行唤醒系统中断源的中断服务程序(ISR)。从ISR返回后，设备继续执行使系统进入掉电模式的下一条指令。 注意：如果 IDL 位和 PD 位同时置位，CPU将进入掉电模式。退出掉电模式后，CPU不会进入空闲模式。
0	IDL	**空闲模式** 置位此位将使CPU进入空闲模式。在此模式下，CPU时钟停止，程序计数器(PC)挂起，但所有外设保持激活状态。CPU从空闲模式唤醒后，此位由硬件自动清除，程序继续执行唤醒系统中断源的ISR。从ISR返回后，设备继续执行使系统进入空闲模式的下一条指令。

TCON – 定时器 0和1控制

寄存器	特殊功能寄存器地址	复位值
TCON	88H, 所有SFR页, 可位寻址	0000_0000b

7	6	5	4	3	2	1	0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**
读/写	读/写	读/写	读/写	可读(电平) 读/写(边沿)	读/写	可读(电平) 读/写(边沿)	读/写

位	名称	描述
7	**TF1**	**定时器1溢出标志** 当定时器1溢出时，此位置位。程序执行定时器1中断服务程序时，硬件自动清零此位。此位也可由软件置位或清零。
6	**TR1**	**定时器1运行控制** 0 = 定时器1禁用。清除此位将暂停定时器 1，当前计数值将保存在 TH1 和 TL1 中。 1 = 定时器1使能。
5	**TF0**	**定时器0 溢出标志** 当定时器0溢出时，此位置位。程序执行定时器0中断服务程序时，硬件自动清除此位。此位也可由软件置位或清除。
4	**TR0**	**定时器0运行控制** 0 = 定时器0禁用。清除此位将暂停定时器 0，当前计数值将保存在 TH0 和 TL0 中。 1 = 定时器0使能。
3	**IE1**	**外部中断1边沿标志位** 如果 IT1 =1(下降沿触发), 当检测到下降沿时此标志位将被硬件置位。此位将保持置1直到软件清零或在其中断服务程序开始时被硬件清零。 如果 IT1 =0(低电平触发), 此标志位是$\overline{\text{INT1}}$输入信号逻辑电平的反相。软件无法控制该标志位。
2	**IT1**	**外部中断1类型选择** 此位选择$\overline{\text{INT1}}$的中断触发类型。 0 = $\overline{\text{INT1}}$ 是低电平触发. 1 = $\overline{\text{INT1}}$ 是下降沿触发.
1	**IE0**	**外部中断0边沿标志** 如果 IT0 =1(下降沿触发), 当检测到下降沿时此标志位将被硬件置位。此位将保持置1直到软件清零或在其中断服务程序开始时被硬件清零。 如果 IT0 =0(低电平触发), 此标志位是$\overline{\text{INT0}}$输入信号逻辑电平的反相。软件无法控制该标志位。
0	**IT0**	**外部中断0类型选择** 此位选择$\overline{\text{INT0}}$的中断触发类型。 0 = $\overline{\text{INT0}}$ 是低电平触发 1 = $\overline{\text{INT0}}$ 是下降沿触发

TMOD – 定时器 0和1模式

寄存器	特殊功能寄存器地址	复位值
TMOD	89H, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
GATE	**C/$\overline{T}$**	**M1**	**M0**	**GATE**	**C/$\overline{T}$**	**M1**	**M0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	GATE	**定时器 1门控制** 0 =无论$\overline{INT1}$的逻辑电平如何，定时器 1 都会在 TR1 为 1 时进行计时。 1 =仅当TR1为1且$\overline{INT1}$为逻辑1时，定时器 1 才会进行计时。
6	C/$\overline{T}$	**定时器1计数器/定时器选择** 0 = 定时器1随内部时钟而递增 1 = 定时器1随外部引脚T1的下降沿递增
5	M1	**定时器1模式选择** M1 M0 定时器1模式 0 0 模式 0: 13位定时器/计数器 0 1 模式1: 16位定时器/计数器 1 0 模式 2: 8位定时器/计数器，从TH1自动重载 1 1 模式 3: 定时器1暂停
4	M0	
3	GATE	**定时器0 门控制** 0 = 无论$\overline{INT0}$的逻辑电平如何，当 TR0 为 1 时，定时器 0 都会开始计时。 1 = 仅当TR0为1且$\overline{INT0}$为逻辑1时，定时器 0 才会开始计时。
2	C/$\overline{T}$	**定时器0 计数器/定时器选择** 0 = 定时器0 随内部时钟而递增 1 = 定时器0 随外部引脚T0的下降沿递增
1	M1	**定时器0 模式选择** M1 M0 定时器0模式 0 0 模式 0: 13位定时器/计数器 0 1 模式 1: 16位定时器/计数器 1 0 模式 2: 8位定时器/计数器，从TH0自动重载 1 1 模式 3: TL0 作为一个 8位定时器/计数器， TH0 作为一个8位定时器
0	M0	

TL0 – 定时器0低字节

寄存器	特殊功能寄存器地址	复位值
TL0	8AH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
TL0[7:0]							
读/写							

位	名称	描述
7:0	TL0[7:0]	**定时器0低字节** TL0 寄存器是定时器 0 的 16 位计数寄存器的低字节。

TL1 – 定时器1低字节

寄存器	特殊功能寄存器地址	复位值
TL1	8BH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
TL1[7:0]							
读/写							

位	名称	描述
7:0	TL1[7:0]	**定时器1低字节** TL1 寄存器是定时器1 的 16 位计数寄存器的低字节。

TH0 – 定时器0高字节

寄存器	特殊功能寄存器地址	复位值
TH0	8CH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
TH0[7:0]							
读/写							

位	名称	描述
7:0	TH0[7:0]	**定时器0高字节** TH0 寄存器是定时器 0 的 16 位计数寄存器的高字节。

TH1 – 定时器1高字节

寄存器	特殊功能寄存器地址	复位值
TH1	8DH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
TH1[7:0]							
读/写							

位	名称	描述
7:0	TH1[7:0]	**定时器1高字节** TH1 寄存器是定时器 1的 16 位计数寄存器的高字节。

CKCON – 时钟控制

寄存器	特殊功能寄存器地址	复位值
CKCON	8EH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	**PWMCKS**	-	**T1M**	**T0M**	-	**CLOEN**	-
-	读/写	-	读/写	读/写	-	读/写	-

位	名称	描述
6	**PWMCKS**	**PWM时钟源选择** 0 = PWM时钟源为系统时钟F_{SYS} 1 = PWM时钟源为定时器1的溢出时钟
4	**T1M**	**定时器1时钟模式选择** 0 = 定时器1的时钟源是系统时钟的12分频，它保持标准8051的兼容性 1 = 定时器1的时钟源直接是系统时钟
3	**T0M**	**定时器0时钟模式选择** 0 = 定时器0的时钟源是系统时钟的12分频，它保持标准8051的兼容性 1 = 定时器0的时钟源直接是系统时钟
1	**CLOEN**	**系统时钟输出使能** 0 = 系统时钟输出禁止 1 = 系统时钟从CLO脚位(P1.1)输出使能 **注意：** CLO 引脚由AUXR5.7决定。

ADCCN – ADC当前采样数

寄存器	特殊功能寄存器地址	复位值
ADCCN	8EH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
ADCCN[7:0]							
读/写							

位	名称	描述
7:0	**ADCCN[7:0]**	**ADC当前采样数** ADC 继续采样选择的当前采样数。 当前采样数 = ADCCN[7:0] + 1

WKCON – 自唤醒定时器控制

寄存器	特殊功能寄存器地址	复位值
WKCON	8FH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	-	-	WKTF	WKTR	WKPS[2:0]		
-	-	-	读/写	读/写	读/写		

位	名称	描述
4	**WKTF**	**WKT 溢出标志** 当WKT溢出，此位置位。如果WKT中断和全局中断使能，置位此位会使CPU执行WKT中断服务程序。此位不会被硬件自动清零，应通过软件清零。
3	**WKTR**	**WKT 运行控制** 0 = WKT 停止. 1 = WKT 开始运行. 注意：重载寄存器RWK仅在WKT停止(WKTR位为0)的时候可以写入。如果WKTR为1时写入WKT，结果不可预测。
2:0	**WKPS[2:0]**	**WKT 预分频** 这些位决定WKT时钟的预分频。 000 = 1/1. 001 = 1/4. 010 = 1/16. 011 = 1/64. 100 = 1/256. 101 = 1/512. 110 = 1/1024. 111 = 1/2048.

ADCSR – ADC 状态寄存器

寄存器	特殊功能寄存器地址	复位值
ADCSR	8FH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	CMPHIT	HDONE	FDONE
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
7:4	-	保留
2	CMPHIT	ADC比较器命中标志 此位仅在 ADCMPO (ADCCON2.4) 标志上升沿时由硬件置位。 注意：此位可通过写0清除。
1	HDONE	A/D转换半完成标志 此位仅在 ADCSN 在连续模式下完成一半 A/D 转换时由硬件置位。 注意：此位可通过写0清除。
0	FDONE	A/D转换完全完成标志 此位仅在 ADCSN 在连续模式下完成所有 A/D 转换或单次模式下完成单次转换时由硬件置位。 注意：此位可通过写0清除。

SFRS – SFR页选择

寄存器	特殊功能寄存器地址	复位值
SFRS	91H, 所有SFR页	0000_0000b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SFRPAGE1	SFRPAGE0
-	-	-	-	-	-	读/写	读/写

位	名称	描述
1:0	SFRPAGE[1:0]	**SFR页选择** 0 = 指令访问 SFR 页0。 1 = 指令访问 SFR 页1。 2 = 指令访问 SFR 页2。

CAPCON0 – 输入捕获控制 0

寄存器	特殊功能寄存器地址	复位值
CAPCON0	92H, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0
-	读/写	读/写	读/写	-	读/写	读/写	读/写

位	名称	描述
6	**CAPEN2**	**输入捕获通道2使能** 0 = 输入捕获通道2禁止. 1 = 输入捕获通道2使能.
5	**CAPEN1**	**输入捕获通道1使能** 0 = 输入捕获通道1禁止. 1 = 输入捕获通道1使能.
4	**CAPEN0**	**输入捕获通道0使能** 0 = 输入捕获通道0禁止. 1 = 输入捕获通道0使能.
2	**CAPF2**	**输入捕获通道 2标志** 如果输入捕获通道2探测到边沿触发事件发生，由硬件置位此位，此位应由软件清零。
1	**CAPF1**	**输入捕获通道1标志** 如果输入捕获通道1探测到边沿触发事件发生，由硬件置位此位，此位应由软件清零。
0	**CAPF0**	**输入捕获通道0标志** 如果输入捕获通道0探测到边沿触发事件发生，由硬件置位此位，此位应由软件清零。

PnUP – 端口n上拉寄存器控制

寄存器	特殊功能寄存器地址	复位值
P0UP	92H, 页2	0000_0000 b
P1UP	93H, 页2	0000_0000 b
P2UP	94H, 页2	0000_0000 b
P3UP	95H, 页2	0000_0000 b

7	6	5	4	3	2	1	0
PnUP.7	**PnUP.6**	**PnUP.5**	**PnUP.4**	**PnUP.3**	**PnUP.2**	**PnUP.1**	**PnUP.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PnUP.x**	**Pn.x 上拉使能** 0 = Pn.x 上拉禁用 1 = Pn.x 上拉选择。

CAPCON1 – 输入捕获控制 1

寄存器	特殊功能寄存器地址	复位值
CAPCON1	93H, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	-	CAP2LS[1:0]		CAP1LS[1:0]		CAP0LS[1:0]	
-	-	读/写		读/写		读/写	

位	名称	描述
5:4	**CAP2LS[1:0]**	**输入捕获通道2 电平选择** 00 = 下降沿 01 = 上升沿 10 = 上升沿或下降沿 11 = 保留
3:2	**CAP1LS[1:0]**	**输入捕获通道1 电平选择** 00 = 下降沿 01 = 上升沿 10 = 上升沿或下降沿 11 = 保留
1:0	**CAP0LS[1:0]**	**输入捕获通道0 电平选择** 00 = 下降沿 01 = 上升沿 10 = 上升沿或下降沿 11 = 保留

CAPCON2 – 输入捕获控制 2

寄存器	特殊功能寄存器地址	复位值
CAPCON2	94H, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	ENF2	ENF1	ENF0	-	-	-	-
-	读/写	读/写	读/写	-	-	-	-

位	名称	描述
6	**ENF2**	**输入捕获通道2 噪声滤波器使能** 0 = 输入捕获通道2 噪声滤波器禁用 1 = 输入捕获通道2 噪声滤波器使能
5	**ENF1**	**输入捕获通道1 噪声滤波器使能** 0 = 输入捕获通道1 噪声滤波器禁用 1 = 输入捕获通道1 噪声滤波器使能
4	**ENF0**	**输入捕获通道0 噪声滤波器使能** 0 = 输入捕获通道0 噪声滤波器禁用 1 = 输入捕获通道0 噪声滤波器使能

CKDIV – 时钟分频器

寄存器	特殊功能寄存器地址	复位值
CKDIV	95H, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
CKDIV[7:0]							
读/写							

位	名称	描述
7:0	**CKDIV[7:0]**	**时钟分频器** 系统时钟频率 F_{SYS}根据CKDIV的值遵循以下计算公式。 当CKDIV = 00H, $F_{SYS} = F_{OSC}$, 当CKDIV = 01H 至 FFH, $F_{SYS} = \frac{F_{OSC}}{2 \quad CKDIV}$。

CKSWT – 时钟开关

寄存器	特殊功能寄存器地址	复位值
CKSWT	96H, 页0, TA保护	0011 _0000 b

7	6	5	4	3	2	1	0
HXTST	ECKP00ST	HIRCST	LIRCST	ECKP30ST	OSC[1:0]		-
只读	只读	只读	只读	只读	只写		-

位	名称	描述
7	HXTST	**高速外部晶振 4 MHz 至 24 MHz 状态** 0 = 高速外部晶振不稳定或禁用。 1 = 高速外部晶振使能且稳定。
6	ECKP00ST	**外部时钟 P00 输入状态** 0 = 外部时钟 P00 输入不稳定或禁用。 1 = 外部时钟 P00 输入使能且稳定。
5	HIRCST	**高速内部振荡器 16 MHz 状态** 0 = 高速内部振荡器不稳定或禁用 1 = 高速内部振荡器使能且稳定
-	-	保留
3	ECKP30ST	**外部时钟 P30 输入状态** 0 = 外部时钟 P30 输入不稳定或禁用。 1 = 外部时钟 P30 输入使能且稳定。
2:1	OSC[1:0]	**振荡器选择位** 此位是用来选择系统时钟源 00 = 内部16MHz振荡器 01 = 根据 EXTEN[1:0](CKEN[7:6])设置选择外部时钟源。 10 = 内部10kHz振荡器 11 = 保留 注意:此位段为只写位段，读回的值可能与当前系统时钟源不一致

CKEN – 时钟使能

寄存器	特殊功能寄存器地址	复位值
CKEN	97H, 页0, TA保护	0011_0000 b

7	6	5	4	3	2	1	0
EXTEN[1:0]		HIRCEN	LIRCEN	-	-	-	CKSWTF
读/写		读/写	读/写	-	-	-	只读

位	名称	描述
7:6	**EXTEN[1:0]**	**外部时钟源使能** 11 = 通过 OSCIN P3.0 输入外部时钟使能 10 = 通过 OSCIN P0.0 输入外部时钟使能 01 = 外部晶振 4~24 MHz 使能 00 = 外部时钟输入禁用。P30/P00/P01 可用作通用 I/O 或其他功能。
5	**HIRCEN**	**16 MHz 内部高速振荡器使能** 0 = 禁用内部高速振荡器。 1 = 使能内部高速振荡器。 需要注意的是，一旦通过设置 IAPEN（CHPCON.0）使能 IAP，16 MHz 内部高速振荡器将自动使能。硬件也会设置 HIRCEN 和 HIRCST 位。IAPEN 清零后，HIRCEN 和 EHRCST 恢复原始值。
4	**LIRCEN**	**10 kHz 内部低速振荡器使能** 0 = 禁用内部低速振荡器。 1 = 使能内部低速振荡器。 请注意，当 (1) WDT 使能，(2) WKT 由内部 10 kHz 振荡器的时钟源运行时，向 LIRCEN 写入 0 将被忽略。LIRCEN 始终为 1，内部 10 kHz 振荡器始终处于使能状态。
3:1	**-**	保留
0	**CKSWTF**	**时钟切换故障标志** 0 =上一次系统时钟源切换成功 1 =用户在上一次的系统时钟源切换时尝试切换到禁用或是不稳定的时钟源。如果切换的时钟不稳定，此位将一直保持为1，直到时钟源稳定并切换成功为止。

SCON – 串口控制

寄存器	特殊功能寄存器地址	复位值
SCON	98H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
SM0/FE	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	SM0/FE	**串口模式选择** SMOD0 (PCON.6) = 0: 详情请参阅表6. 9-1 串口 UART0 模式/波特率说明 SMOD0 (PCON.6) = 1: SM0/FE 位用作帧错误(FE)状态标志位。由软件清零。 0 = 未发生帧错误(FE)。 1 = 发生并检测到帧错误(FE)。
6	SM1	
5	SM2	**多机通信模式使能** 此位功能取决于串口0模式。 模式0: 此位选择波特率 $F_{SYS}/12$ 或 $F_{SYS}/2$ 0 = 时钟以 $F_{SYS}/12$ 波特率运行， 保持与标准8051的兼容性。 1 = 时钟以 $F_{SYS}/2$ 波特率运行，以实现更快的串口通信。 模式1: 此位检查有效的停止位 0 = 无论停止位的逻辑电平，接收始终有效。 1 = 仅当接收的停止位是逻辑1并且接收的数据匹配“给定”或“广播”地址时，接收才有效。 模式 2 或 3: 用于多机通信。 0 = 不管第9位的逻辑电平，接收始终有效。 1 = 仅当接收的第9位是逻辑1并且接收的数据匹配“给定”或“广播”地址时，接收才有效。
4	REN	**接收使能** 0 = 串口0 接收禁用。 1 = 串口0 接收在模式1,2,3下使能。在模式0下， REN =1并且 RI =0时接收被启动初始化。
3	TB8	**第9个发送位** 此位定义串口0在模式2，3时第9个发送位的状态，模式0,1下不使用此位.
2	RB8	**第9个接收位** 此位标识串口0 模式2或3中的第9个接收位的逻辑电平。在模式1下，RB8为接收停止位的逻辑电平，SM2位为逻辑电平1 时，有异常限制。在模式0下RB8 不使用。

位	名称	描述
1	TI	**发送中断标志位** 模式0下串口0发送完第8位数据后，或其它模式下发送完数据帧最后一个数据位后，硬件会置位此标志位。当串口0中断使能，此位的置位将让CPU执行串口0 中断服务程序。此位应由软件手动清除。
0	RI	**接收中断标志位** 模式0下数据接收帧收到第8位后，或者模式1、2或3下采样到停止位后，硬件会置位此标志位。SM2位为逻辑电平1时，有异常限制。。当串口0中断使能，此位的置位将让CPU执行串口0 中断服务程序。此位应由软件手动清除。

SBUF – 串口0数据缓冲区

寄存器	特殊功能寄存器地址	复位值
SBUF	99H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SBUF[7:0]							

位	名称	描述
7:0	**SBUF[7:0]**	**串口0数据缓冲区** 这个字节实际上由两个独立的寄存器组成。一个是接收寄存器，另一个是发送缓冲区。当数据移入SBUF时，它会进入到发送缓冲区并被移位以进行串行传输。当数据从SBUF移出时，它来自接收寄存器。 通过向SBUF写入数据来启动传输。

SBUF_1 – 串口1数据缓冲区

寄存器	特殊功能寄存器地址	复位值
SBUF_1	9AH, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
SBUF_1[7:0]							
读/写							

位	名称	描述
7:0	**SBUF_1[7:0]**	**串口1数据缓冲区** 这个字节实际上由两个独立的寄存器组成。一个是接收寄存器，另一个是发送缓冲区。当数据移入SBUF_1 时，它进入到发送缓冲区并被移位以进行串行传输。当数据从SBUF_1 移出时，它来自接收寄存器。 通过向SBUF_1 写入数据来启动传输。

PnDW – 端口n下拉寄存器控制

寄存器	特殊功能寄存器地址	复位值
P0DW	9AH, 页2	0000_0000 b
P1DW	9BH, 页2	0000_0000 b
P2DW	9CH, 页2	0000_0000 b
P3DW	9DH, 页2	0000_0000 b
P4DW	9EH, 页2	0000_0000 b
P5DW	9FH, 页2	0000_0000 b

7	6	5	4	3	2	1	0
PnDW.7	**PnDW.6**	**PnDW.5**	**PnDW.4**	**PnDW.3**	**PnDW.2**	**PnDW.1**	**PnDW.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PnDW.x**	**Pn.x 下拉使能** 0 = Pn.x 下拉禁用。 1 = Pn.x 下拉使能。

EIE – 扩展中断使能

寄存器	特殊功能寄存器地址	复位值
EIE	9BH, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
ET2	ESPI	EFB	EWDT	EPWM0	ECAP	EPI	EI²C
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**ET2**	**定时器2中断使能** 0 = 定时器2中断禁止。 1 = 由 TF2(T2CON.7)产生的中断使能。
6	**ESPI**	**SPI 中断使能** 0 = SPI 中断禁止。 1 = 由 SPIF(SPSR.7), SPIOVF(SPSR.5)或 MODF(SPSR.4)产生的中断使能。
5	**EFB**	**故障刹车中断使能** 0 = 故障刹车中断禁止。 1 = 由FBF(FBD.7)产生的中断使能。
4	**EWDT**	**看门狗中断使能** 0 = 看门狗中断禁止。 1 = 由 WDTF(WDCON.5) 产生的中断使能。
3	**EPWM0**	**PWM中断使能** 0 = PWM中断禁止。 1 = 由 PWMF(PWMCON0.5) 产生的中断使能。
2	**ECAP**	**输入捕获中断使能** 0 = 输入捕获中断禁止。 1 = 由 CAPF[2:0](CAPCON0[2:0])中的任意标志位产生的中断使能。
1	**EPI**	**引脚中断使能** 0 = 引脚中断禁止。 1 = 由 PIF 寄存器中的任意标志位产生的中断使能。
0	**EI²C**	**I²C中断使能** 0 = I²C 中断禁止。 1 = 由 SI(I2CON.3)或 I2TOF(I2TOC.0) 产生的中断使能。

EIE1 – 扩展中断使能 1

寄存器	特殊功能寄存器地址	复位值
EIE1	9CH, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	EPWM3	EPWM2	EPWM1	EWKT	ET3	ES_1
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
5	**EPWM3**	**PWM3中断使能** 0 = PWM3中断禁止。 1 = 由PWM3F (PWM3CON0.5) 产生的中断使能。
4	**EPWM2**	**PWM2中断使能** 0 = PWM2中断禁止。 1 = 由PWM2F (PWM2CON0.5) 产生的中断使能。
3	**EPWM1**	**PWM1中断使能** 0 = PWM1中断禁止。 1 = 由PWM1F (PWM1CON0.5) 产生的中断使能。
2	**EWKT**	**WKT中断使能** 0 = WKT中断禁止。 1 = 由WKTF(WKCON.4) 产生的中断使能。
1	**ET3**	**定时器3中断使能** 0 = 定时器3中断禁止。 1 = 由TF3(T3CON.4) 产生的中断使能。
0	**ES_1**	**串口1中断使能** 0 = 串口1中断禁止。 1 = 由TI_1(SCON_1.1)或 RI_1(SCON_1.0)产生的中断使能。

RSR – 复位标志寄存器

寄存器	特殊功能寄存器地址	复位值
RSR	9DH, SFR页0	00XX_XXXX b

7	6	5	4	3	2	1	0
-	-	HardF	POF	RSTPINF	BORF	WDTRF	SWRF
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:6	-	保留
5	HardF	**来自 AUXR1.5的镜像** 写入AUXR1.5=0 或 RSR.5=0 来清除此位。
4	POF	**来自 PCON.4的镜像** 写入PCON.4=0 或 RSR.4=0 来清除此位。
3	RSTPINF	**来自 AUXR1.6的镜像** 写入AUXR1.6=0 或 RSR.3=0 来清除此位。
2	BORF	**来自 BODCON0.1的镜像** 写入BODCON0.1=0 或 RSR.2=0 来清除此位。
1	WDTRF	**来自 WDCON.3的镜像** 写入WDCON.3=0 或 RSR.1=0 来清除此位。
0	SWRF	**来自 AUXR1.7的镜像** 写入AUXR1.7=0 或 RSR.0=0 来清除此位。

CHPCON – 芯片控制

寄存器	特殊功能寄存器地址	复位值
CHPCON	9FH, SFR页0, TA保护	软件复位:0000_00U0 b 其他: 0000_00C0 b

7	6	5	4	3	2	1	0
SWRST	**IAPFF**	**-**	**-**	**-**	**-**	**BS**	**IAPEN**
只写	读/写	-	-	-	-	读/写	读/写

位	名称	描述
7	**SWRST**	**软件复位** 此位置位将执行软件复位，复位完成后此位由硬件自动清零
6	**IAPFF**	**IAP 故障标志** 在IAPGO(ISPTRG.0)置位后，如果满足以下任意条件，硬件将置位此位： (1)访问地址超过其大小的区域 (2)IAPCN 命令无效 (3)IAP擦除或编程更新未使能的区块 (4)当BOIAP(CONFIG2.5)在未编辑的1、 BODEN(BODCON0.7)为1以及BORST(BODCON0.2)为0 时，IAP在V_{BOD}模式下执行擦除或编程。 此位应由软件清零。
1	**BS**	**启动选择** 此位定义MCU在全部复位后由哪块重新启动 0 = 全部复位后，MCU将从APROM重新启动 1 = 全部复位后，MCU将从LDROM重新启动
0	**IAPEN**	**IAP 使能** 0 = IAP功能禁用 1 = IAP功能使能 一旦使能IAP功能，HIRC将会被打开作为时序控制。如果希望降低功耗，应始终在 IAP 操作后的最后一条指令清零 IAPEN，以停止内部振荡器。

AUXR1 – 辅助寄存器 1

寄存器	特殊功能寄存器地址	复位值
AUXR1	A2H , SFR页0	上电复位: 0000 0000b, 软件复位: 1U00 0000b, 复位脚: U100 0000b, 其他: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	**RSTPINF**	**HardF**	**-**	**GF2**	**UART0PX**	**0**	**DPS**
读/写	读/写	读/写	-	读/写	读/写	只读	读/写

位	名称	描述
7	**SWRF**	**软件复位标志位** 当MCU发生软件复位后，此位由硬件置位。建议由软件清0。
6	**RSTPINF**	**外部复位标志位** 当MCU发生外部复位引脚复位，此位由硬件置位。建议由软件清0。
5	**HardF**	**硬件故障复位标志** 一旦程序计数器(PC)的值超过flash的大小，MCU将会复位并且此位由硬件置位。建议通过软件清除此位。 注意：如果MCU运行在OCD调试模式并且OCDEN = 0，硬件故障复位将会被禁止，只有HardF标志位有效。
4	**-**	**保留**
3	**GF2**	**通用标志位2** 可通过软件置位或清除
2	**UART0PX**	**串口0引脚交换** 0 = 默认RXD 为 P0.7， TXD 为 P0.6。 1 = 交换RXD 为 P0.6， TXD 为 P0.7。 注意：设置或清除此位后，TXD及RXD的交换立即生效。用户需注意在串口发送或接收过程中不要交换引脚，否则可能引发不可预测的结果，且不会收到警告提醒。
1	**0**	**保留** 此位始终读到0
0	**DPS**	**数据指针选择** 0 = 数据指针0(DPTR)默认激活. 1 = 数据指针1(DPTR1)激活. 当通过 DPS 切换激活数据指针后，先前未激活的数据指针的值保持不变。

AUXR2 – 辅助寄存器 2

寄存器	特殊功能寄存器地址	复位值
AUXR2	A1H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
UART2TXP		UART2RXP		UART1TXP		UART1RXP	
读/写		读/写		读/写		读/写	

位	名称	描述
7:6	**UART2TXP**	**UART2 TX 引脚选择** 00 = 默认保留 01 = 将UART2 TX分配给P0.3 10 = 将UART2 TX分配给P3.0 11 = 保留
5:4	**UART2RXP**	**UART2 RX引脚选择** 00 = 默认保留 01 = 将UART2 RX分配给 P0.4 10 = 将UART2 RX 分配给 P1.7 11 = 保留
3:2	**UART1TXP**	**UART1 TX引脚选择** 00 = 默认保留 01 = 将UART1 TX 分配给 P1.6 10 = 将UART1 TX 分配给 P3.6 11 = 将UART1 TX 分配给 P1.0
1:0	**UART1RXP**	**UART1 RX引脚选择** 00 = 默认保留 01 = 将UART1 RX 分配给 P0.2 10 = 将UART1 RX 分配给 P3.7 11 = 将UART1 RX 分配给 P0.0

AUXR3 – 辅助寄存器 3

寄存器	特殊功能寄存器地址	复位值
AUXR3	A2H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
UART4TXP		UART4RXP		UART3TXP		UART3RXP	
读/写		读/写		读/写		读/写	

位	名称	描述
7:6	**UART4TXP**	**UART4 TX 引脚选择** 00 = 默认保留 01 = 将UART4 TX分配给P2.3 10 = 保留 11 = 保留
5:4	**UART4RXP**	**UART4 RX 引脚选择** 00 = 默认保留 01 = 将UART4 RX分配给P2.2 10 = 保留 11 = 保留
3:2	**UART3TXP**	**UART3 TX 引脚选择** 00 = 默认保留 01 = 将UART3 TX 分配给P1.2 10 = Assign UART3 TX 分配给 P1.5 11 = Assign UART3 TX 分配给 P0.5
1:0	**UART3RXP**	**UART3 RX 引脚选择** 00 = 默认保留 01 = 将UART3 RX 分配给 P1.1 10 = 将UART3 RX 分配给 P2.5 11 = 将UART3 RX 分配给 P3.4

AUXR4 – 辅助寄存器 4

寄存器	特殊功能寄存器地址	复位值
AUXR4	A3H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
PWM2CH1P		PWM2CH0P		PWM1CH1P		PWM1CH0P	
读/写		读/写		读/写		读/写	

位	名称	描述
7:6	**PWM2CH1P**	**PWM2 通道1引脚选择** 00 = 将PWM2_CH1分配给P3.0 01 = 将PWM2_CH1分配给P3.1 10 = 将PWM2_CH1分配给P0.0 11 = 将PWM2_CH1分配给P0.4
5:4	**PWM2CH0P**	**PWM2 通道0引脚选择** 00 = 默认保留 01 = 将PWM2_CH0分配给P2.1 10 = 将PWM2_CH0分配给P1.0 11 = 将PWM2_CH0分配给P0.5
3:2	**PWM1CH1P**	**PWM1 通道1引脚选择** 00 = 默认保留 01 = 将PWM1_CH1分配给P2.2 10 = 将PWM1_CH1分配给 P1.4 11 = 将PWM1_CH1分配给 P1.1
1:0	**PWM1CH0P**	**PWM1 通道0引脚选择** 00 = 默认保留 01 = 将PWM1_CH0分配给P2.3 10 = 将PWM1_CH0分配给P1.2 11 = 保留

AUXR5 – 辅助寄存器 5

寄存器	特殊功能寄存器地址	复位值
AUXR5	A4H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
CLOP	T0P	-		PWM3CH1P		PWM3CH0P	
读/写	读/写	-		读/写		读/写	

位	名称	描述
7	**CLOP**	**CLK 输出引脚选择** 0 = 当CLO输出使能，将CLO 分配给P1.1。 1 = 当CLO输出使能，将CLO 分配给P3.3。
6	**T0P**	**T0 引脚选择** 0 = 将T0分配给P0.5 1 = 将T0 分配给P2.4
5:4	**-**	保留
3:2	**PWM3CH1P**	**PWM3 通道1引脚选择** 00 = 默认保留 01 = 将 PWM3_CH1分配给 P3.4 10 = 将 PWM3_CH1分配给P1.5 11 = 将 PWM3_CH1分配给P0.3
1:0	**PWM3CH0P**	**PWM3 通道0引脚选择** 00 = 默认保留 01 = 将 PWM3_CH0分配给P3.2 10 = 将 PWM3_CH0分配给P0.1 11 = 保留

AUXR6 – 辅助寄存器 6

寄存器	特殊功能寄存器地址	复位值
AUXR6	A5H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	UART4DG	UART3DG	UART2DG	UART1DG	UART0DG
-	-	-	读/写	读/写	读/写	读/写	读/写

位	名称	描述
4	UART4DG	**UART4 RX 去毛刺控制** 1: 去毛刺使能 0: 去毛刺禁用
3	UART3DG	**UART3 RX 去毛刺控制** 1: 去毛刺使能 0: 去毛刺禁用
2	UART2DG	**UART2 RX 去毛刺控制** 1: 去毛刺使能 0: 去毛刺禁用
1	UART1DG	**UART1 RX 去毛刺控制** 1: 去毛刺使能 0: 去毛刺禁用
0	UART0DG	**UART0 RX 去毛刺控制** 1: 去毛刺使能 0: 去毛刺禁用

AUXR7 – 辅助寄存器 7

寄存器	特殊功能寄存器地址	复位值
AUXR7	A6H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	SPI0NSSP		SPIMOSIP	SPIMISOP	SPICKP
-	-	-	读/写		读/写	读/写	读/写

位	名称	描述
4:3	**SPI0NSSP**	**SPI0_SS 引脚选择** 00 = 将SPI0_SS分配给 P1.5 01 = 保留 10 = 将SPI0_SS分配给 P3.5 11 = 保留
2	**SPI0MOSIP**	**SPI0_MOSI 引脚选择** 0 = 将SPI0_MOSI分配给P0.0 1 = 将SPI0_MOSI分配给P3.0
1	**SPI0MISOP**	**SPI_MISO 引脚选择** 0 = 将SPI_MISO分配给P0.1 1 = 将SPI_MISO分配给P2.5
0	**SPI0CKP**	**SPI_CLK 引脚选择** 0 = 将SPI_CLK分配给P1.0 1 = 将SPI_CLK分配给P1.7

AUXR8 – 辅助寄存器 8

寄存器	特殊功能寄存器地址	复位值
AUXR8	A7H, SFR页2, TA保护	0000_0000 b

7	6	5	4	3	2	1	0
CLODIV[3:0]				CKTESTOEN[3:0]			
读/写				读/写			

位	名称	描述
7:4	**CLODIV[3:0]**	**时钟输出分频** 系统时钟输出根据CLODIV的值遵循以下计算公式。 当CLODIV = 00H,$F_{CLKO} = F_{SYS}$, 而当CLODIV = 01H至0FH,$F_{CLKO} = \frac{F_{SYS}}{2x\ CLODIV}$
3-0	**CKTESTOEN[3:0]**	**输出到CKO的时钟宏源** CKTESTOEN 功能 0000 HIRC (24 MHz) 输出至CKO功能引脚 0001 OSCIN输出至CKO功能引脚 0010 LIRC (10 kHz) 输出至CKO功能引脚 0011 HXT (4~24 MHz) 输出至CKO功能引脚

BODCON0 – 欠压检测控制0

寄存器	特殊功能寄存器地址	复位值
BODCON0	A3H, SFR页0, TA保护	上电复位POR：CCCC XC0X b 欠压复位BOD：UUUU XU1X b 其他：UUUU XUUX b

7	6	5	4	3	2	1	0
BODEN[1]	**-**	**BOV[1:0]**[1]		**BOF**[2]	**BORST**[1]	**BORF**	**BOS**
读/写		读/写		读/写	读/写	读/写	只读

位	名称	描述
7	**BODEN**	**欠压检测使能** 0 = 欠压检测电路关闭 1 = 欠压检测电路打开 注意在开启该功能后需要2到3个LIRC时钟BOD输出才能正常工作 。
6:4	**BOV[1:0]**	**欠压检测电压选择** 11 = V_{BOD} 生效电压 2.2V. 10 = V_{BOD} 生效电压 2.7V. 01 = V_{BOD} 生效电压 3.7V. 00 = V_{BOD} 生效电压 4.4V.
3	**BOF**	**欠压中断标志** 当V_{DD}下降到V_{BOD}以下或V_{DD}上升到V_{BOD}以上时，此该标志由硬件置位逻辑1。如果EBOD(EIE.2)和EA(IE.7)都置位，将生成欠压中断请求。此位应由软件清零。
2	**BORST**	**欠压检测复位使能** 此位决定在电源电压跌到V_{BOD}以下时是否产生欠压复位 0 = 当V_{DD}掉电到V_{BOD}以下，欠压复位禁用 1 =当V_{DD}掉电到V_{BOD}以下，欠压复位使能
1	**BORF**	**欠压复位标志** 当MCU发生欠压复位时，此位被硬件置位。建议通过软件清除此位。
0	**BOS**	**欠压状态标志** 在BOD电路开启时，此位指示V_{DD}与V_{BOD}的比较情况。BOD电路关闭时，此位保持为0. 0 = V_{DD} 电压高于 V_{BOD} 或BOD 电路关闭 1 = V_{DD} 电压低于V_{BOD} 注：此位为只读位
注意： 1.所有复位后，BODEN、BOV[1:0] 和 BORST 均通过直接从 CONFIG2的位7、[6:4] 和位2加载进行初始化。 2.BOF复位值取决于CONFIG2和V_{DD}电压电平的不同设置。请参阅表6.2-1 BOF复位值		

IAPTRG – IAP 触发

寄存器	特殊功能寄存器地址	复位值
IAPTRG	A4H, SFR页0, TA保护	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	**IAPGO**
-	-	-	-	-	-	-	只写

位	名称	描述
0	**IAPGO**	**IAP执行** 置位此位为逻辑1时开始执行IAP。在此指令之后，CPU保持程序计数器(PC)值不变，IAP硬件自动接管进程控制。IAP操作完成后，程序计数器继续执行下一条指令。IAPGO位将被自动清除并始终读取为逻辑0。 在触发IAP操作之前，应临时禁用中断(如果已使能)，以限制硬件执行。

IAPUEN – IAP更新使能

寄存器	特殊功能寄存器地址	复位值
IAPUEN	A5H, SFR页0, TA保护	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	-	SPMEN	SPUEN	CFUEN	LDUEN	APUEN
-	-	-	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:5	-	保留
4	SPMEN	**SPROM存储空间映射使能** 0 =CPU存储地址0xff80~0xffff 映射到APROM存储。 1 =CPU存储地址0xff80~0xffff 映射到SPROM存储。
3	SPUEN	**SPROM存储空间更新使能(TA保护)** 0 = 禁止通过IAP擦除或编程SPROM字节。 1 = 允许通过IAP擦除或编程SPROM字节。
2	CFUEN	**配置字节更新使能(TA保护)** 0 = 禁止通过IAP擦除或编程CONFIG字节。 1 = 允许通过IAP擦除或编程CONFIG字节。
1	LDUEN	**LDROM更新使能(TA保护)** 0 = 禁止通过IAP擦除或编程LDROM。 1 = 允许通过IAP擦除或编程LDROM。
0	APUEN	**APROM更新使能(TA保护)** 0 = 禁止通过IAP擦除或编程APROM。 1 = 允许通过IAP擦除或编程APROM。

IAPAL – IAP地址低字节

寄存器	特殊功能寄存器地址	复位值
IAPAL	A6H, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPA[7:0]							
读/写							

位	名称	描述
7:0	**IAPA[7:0]**	**IAP地址低字节** IAPAL包含用于IAP操作的地址 IAPA[7:0] 。

IAPAH – IAP地址高字节

寄存器	特殊功能寄存器地址	复位值
IAPAH	A7H, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPA[15:8]							
读/写							

位	名称	描述
7:0	**IAPA[15:8]**	**IAP地址高字节** IAPAH包含用于IAP操作的地址 IAPA[15:8] 。

IE – 中断使能

寄存器	特殊功能寄存器地址	复位值
IE	A8H, 所有SFR页, 可位寻址	0000 _0000 b

7	6	5	4	3	2	1	0
EA	**EADC**	**EBOD**	**ES**	**ET1**	**EX1**	**ET0**	**EX0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**EA**	**使能所有中断** 此位全局使能/禁用所有单独使能的中断。 0 = 所有中断源禁用。 1 = 每个中断根据其各自的屏蔽设置单独启用。如果启用，将产生相应的单独中断。
6	**EADC**	**ADC中断使能** 0 = ADC 中断禁用。 1 = 由ADCF(ADCCON0.7)产生的中断使能。
5	**EBOD**	**欠压中断使能** 0 = 欠压中断禁用。 1 = 由BOF(BODCON0.3)产生的中断使能。
4	**ES**	**串口0中断使能** 0 = 串口0中断禁用。 1 = 由TI(SCON.1)或RI(SCON.0)产生的中断使能。
3	**ET1**	**定时器1中断使能** 0 = 定时器1中断禁用。 1 = 由TF1(TCON.7)产生的中断使能。
2	**EX1**	**外部中断1使能** 0 = 外部中断1禁用。 1 = 由 $\overline{INT1}$ pin(P1.7)产生的中断使能。
1	**ET0**	**定时器0中断使能** 0 = 定时器0中断禁用。 1 = 由TF0(TCON.5)产生的中断使能。
0	**EX0**	**外部中断0使能** 0 = 外部中断0禁用。 1 = 由 $\overline{INT0}$ pin(P3.0)产生的中断使能。

SADDR – 从机0地址

寄存器	特殊功能寄存器地址	复位值
SADDR	A9H, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
SADDR[7:0]							
读/写							

位	名称	描述
7:0	**SADDR[7:0]**	**从机0地址** 此字节定义微控制器自身的从机地址，用于UART0 多机通讯。

WDCON – 看门狗定时器控制

寄存器	特殊功能寄存器地址	复位值
WDCON	AAH, SFR页0, TA保护	上电复位POR： 0000_0111 b 看门狗复位：0000_1UUU b 其他： 0000_UUUU b

7	6	5	4	3	2	1	0
WDTR	**WDCLR**	**WDTF**	**WIDPD**	**WDTRF**	**WDPS[2:0]**		
读/写	读/写	读/写	读/写	读/写	读/写		

位	名称	描述
7	**WDTR**	**WDT运行** 此位仅当WDTEN[3:0](CONFIG4[7:4]) 中的控制位全为1时有效。此时，WDT作为通用定时器工作。 0 = WDT禁止 1 = WDT使能，WDT计数器开始运行.
6	**WDCLR**	**WDT清零** 置位此位重置WDT计数为00H。它使计数器处于已知的状态，防止系统出现不可预知的复位。写入和读取WDCLR位的含义不同。 写入: 0 = 无影响 1 = 清除WDT计数器 读取: 0 = WDT计数器已完全清零 1 = WDT计数器尚未清零
5	**WDTF**	**WDT超时标志** 此位表示WDT计数器溢出。此标志位应由软件清零。
4	**WIDPD**	**WDT在空闲或掉电模式下运行** 此位仅当 WDTEN[3:0](CONFIG4[7:4])中的控制位全为1时才有效。当WDT作为通用定时器工作时，此位决定WDT在空闲或掉电模式下是否运行。 0 = WDT在空闲或掉电模式期间停止运行。 1 = WDT在空闲或掉电模式期间保持运行。
3	**WDTRF**	**WDT复位标志** 当MCU因WDT超时事件而复位时，此位由硬件置位。建议由软件清除此标志位。
2:0	**WDPS[2:0]**	**WDT时钟预分频选择** 这些位决定WDT时钟的预分频，范围从 1/1 到 1/256。

注意:

1. WDTRF上电复位后会被清0, WDT复位后会被置1, 其余复位后保持原有值不变。
2. WDPS[2:0] 上电复位后全部置位, 除此以外的任何其他复位，值均保持不变。

BODCON1 – 欠压检测控制1

寄存器	特殊功能寄存器地址	复位值
BODCON1	ABH, SFR页0, TA保护	上电复位POR： 0000 0001 b 其他： 0000 0UUU b

7	6	5	4	3	2	1	0
-	-	-	-	-	**LPBOD[1:0]**		**BODFLT**
-	-	-	-	-	读/写		读/写

位	名称	描述
7:3	-	保留
2:1	**LPBOD[1:0]**	**低功耗BOD使能** 00 = BOD正常模式，BOD电路始终启用。. 01 = BOD低功耗模式1， 每隔1.6ms周期性开启BOD电路 10 = BOD低功耗模式 2， 每隔6.4ms周期性开启BOD电路 11 = BOD低功耗模式 3， 每隔25.6ms周期开启BOD电路
0	**BODFLT**	**BOD滤波器控制** BOD内置一个滤波器，当MCU在HIRC或ECLK作为系统时钟源且BOD没有在低功率模式(LPBOD[1:0] = [0, 0]) 下工作时，该滤波器计数32个F_{SYS}时钟周期，用以过滤电源噪声。在其他情况下，滤波器计数2个LIRC时钟周期。 注意:当CPU停在掉电模式时，BOD将始终由2个LIRC时钟周期滤波输出。 BOD滤波器有效地避免电源噪声误触发BOD事件。设置此位控制BOD滤波功能开启或关闭。 0 = BOD滤波器禁用. 1 = BOD滤波器使能.(上电复位默认值)

IAPFD – IAP Flash数据

寄存器	特殊功能寄存器地址	复位值
IAPFD	AEH, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPFD[7:0]							
读/写							

位	名称	描述
7:0	**IAPFD[7:0]**	**IAP Flash数据** 此字节包含 Flash 数据，这些数据是从 Flash 中读取的，或者即将写入 Flash 的。编程模式下，用户应在触发 IAP 处理之前将数据写入 IAPFD ，读取/ 验证模式下，在 IAP 处理完成后从 IAPFD 读取数据。

IAPCN – IAP 控制

寄存器	特殊功能寄存器地址	复位值
IAPCN	AFH, SFR页0	0011_0000 b

7	6	5	4	3	2	1	0
IAPB[1:0]		FOEN	FCEN	FCTRL[3:0]			
读/写		读/写	读/写	读/写			

位	名称	描述
7:6	IAPB[1:0]	IAP控制 此字节用于 IAP 命令。 详细描述参见表6.3-1 IAP .
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

PnM1 – 端口n模式选择1

寄存器	特殊功能寄存器地址	复位值
P0M1	B1H, SFR页0	1111_1111 b
P1M1	B3H, SFR页0	1111_1111 b
P2M1	89H, SFR页2	0011_1111 b
P3M1	ACH, SFR页0	1111_1111 b
P4M1	89H, SFR页1	1111_1111 b
P5M1	99H, SFR页1	1111_1111 b

7	6	5	4	3	2	1	0
PnM1.7	**PnM1.6**	**PnM1.5**	**PnM1.4**	**PnM1.3**	**PnM1.2**	**PnM1.1**	**PnM1.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PnM1[7:0]**	**端口n模式选择1** **注意:** PnM1和PnM2组合使用来确定端口每个引脚的I/O模式。参见表6. 4-1 .

PnM2 – 端口n模式选择2

寄存器	特殊功能寄存器地址	复位值
P0M2	B2H, SFR页0	0000_0000 b
P1M2	B4H, SFR页0	0000_0000 b
P2M2	8AH, SFR页2	0000_0000 b
P3M2	ADH, SFR页0	0000_0000 b
P4M2	8AH, SFR页1	0000_0000 b
P5M2	9AH, SFR页1	0000_0000 b

7	6	5	4	3	2	1	0
PnM2.7	**PnM2.6**	**PnM2.5**	**PnM2.4**	**PnM2.3**	**PnM2.2**	**PnM2.1**	**PnM2.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PnM2[7:0]**	**端口n模式选择2** **注意:** PnM1和PnM2组合使用来确定端口每个引脚的I/O模式。参见表6. 4-1 .

PnS – 端口n施密特触发输入

寄存器	特殊功能寄存器地址	复位值
P0S	B1H, SFR页1	0000_0000 b
P1S	B3H, SFR页1	0000_0000 b
P2S	8CH, SFR页2	0000_0000 b
P3S	ACH, SFR页1	0000_0000 b
P4S	8CH, SFR页1	0000_0000 b
P5S	9CH, SFR页1	0000_0000 b

7	6	5	4	3	2	1	0
PnS.7	**PnS.6**	**PnS.5**	**PnS.4**	**PnS.3**	**PnS.2**	**PnS.1**	**PnS.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PnS[7:0]**	**Pn施密特触发输入** 0 = P0.n的TTL电平输入。 1 = P0.n的施密特触发输入。 **注：**每个 I/O 引脚均可单独配置为 TTL 输入或施密特触发输入。请注意，所有 PxS 寄存器均可通过将 SFR 页面切换到第 1 页来访问。

TOE – 定时器01输出使能

寄存器	特殊功能寄存器地址	复位值
TOE	B5H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	T1OE	T0OE	-	-
-	-	-	-	读/写	读/写	-	-

位	名称	描述
3	T1OE	**定时器1输出使能** 0 = 定时器1 输出禁止。 1 = 定时器1 从T1引脚输出使能。 请注意，定时器 1 输出仅在“定时器”模式下启用。
2	T0OE	**定时器0输出使能** 0 = 定时器0 输出禁止。 1 = 定时器0 从T0引脚输出使能。 请注意，定时器 0 输出仅在“定时器”模式下启用。

IPH – 中断优先级高[2]

寄存器	特殊功能寄存器地址	复位值
IPH	B7H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H
-	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
6	**PADC**	ADC中断优先级高位
5	**PBOD**	欠压检测中断优先级高位
4	**PSH**	串口0中断优先级高位
3	**PT1H**	定时器1中断优先级高位
2	**PX1H**	外部中断1中断优先级高位
1	**PT0H**	定时器0 中断优先高位
0	**PX0H**	外部中断0中断优先级高位

[2]**注：** IPH与 IP 分别组合使用，决定每个中断源的优先级。有关正确的中断优先级配置，请参见 表6.2-4 中断优先级层级设置

PWMnINTC – PWM中断控制

寄存器	特殊功能寄存器地址	复位值
PWM0INTC	B7H, 页1	0000_0000 b
PWM1INTC	B6H, 页2	0000_0000 b
PWM2INTC	C6H, 页2	0000_0000 b
PWM3INTC	D6H, 页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0
-	-	读/写	读/写	-	读/写	读/写	读/写

位	名称	描述
5:4	**INTTYP[1:0]**	**PWM中断类型选择** 这些位用于选择PWM中断类型。 00 = PWM通道0/1/2/3/4/5 引脚的下降沿. 01 = PWM通道0/1/2/3/4/5 引脚的上升沿. 10 = PWM周期的中心点. 11 = PWM周期的终点. 注意：中心点中断方式或终点中断方式仅适用于PWM中心对齐模式
2:0	**INTSEL[2:0]**	**PWM中断组选择** 在PWM通道0~5引脚的PWM中断类型选择为上升沿或下降沿中断时，此位段用于选择哪个PWM通道发出PWM中断。 000 = PWM0_CH0. 001 = PWM0_CH1. 010 = PWM0_CH2. 011 = PWM0_CH3. 100 = PWM0_CH4. 101 = PWM0_CH5. 其他 = PWM0_CH0.

IP – 中断优先级

寄存器	特殊功能寄存器地址	复位值
IP	B8H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
-	**PADC**	**PBOD**	**PS**	**PT1**	**PX1**	**PT0**	**PX0**
-	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
6	**PADC**	ADC中断优先级低位
5	**PBOD**	欠压检测中断优先级低位
4	**PS**	串口0中断优先级低位
3	**PT1**	定时器1中断优先级低位
2	**PX1**	外部中断1优先级低位
1	**PT0**	定时器0中断优先级低位
0	**PX0**	外部中断0优先级低位

注： IP与 IPH 分别组合使用，决定每个中断源的优先级。有关正确的中断优先级配置，请参见 表6. 2-4 中断优先级层级设置

SADEN – 从机0地址掩码

寄存器	特殊功能寄存器地址	复位值
SADEN	B9H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SADEN[7:0]							
读/写							

位	名称	描述
7:0	**SADEN[7:0]**	**从机0地址掩码** 此字节为UART0的掩码，其中无关位用0表示，来组成设备的“给定”地址。无关位可以灵活地同时寻址一个或多个从机。

SADEN_1 – 从机1地址掩码

寄存器	特殊功能寄存器地址	复位值
SADEN_1	BAH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SADEN_1[7:0]							
读/写							

位	名称	描述
7:0	**SADEN_1[7:0]**	**从机1地址掩码** 此字节为UART1的掩码，其中无关位用0表示，来组成设备的“给定”地址。无关位可以灵活地同时寻址一个或多个从机。

SADDR_1 – 从机1地址

寄存器	特殊功能寄存器地址	复位值
SADDR_1	BBH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SADDR_1[7:0]							
读/写							

位	名称	描述
7:0	SADDR_1[7:0]	**从机1地址** 此字节定义微控制器自身的从机地址，用于串口UART1多机通讯。

I2DAT – I²C 数据

寄存器	特殊功能寄存器地址	复位值
I2DAT	BCH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
I2DAT[7:0]							
读/写							

位	名称	描述
7:0	I2DAT[7:0]	**I²C 数据** I2DAT包含I²C数据待发送的一个字节，或刚接收到的一个字节。只要SI 为逻辑 1，I2DAT内的数据就会一直保留。在I²C传输过程中读取或写入I2DAT，结果不可预测。 当 I2DAT 中的数据移出时，总线上的数据也会同时移入以更新I2DAT。I2DAT 始终显示I²C总线上出现的最后一个字节。所以总线仲裁失败时，I2DAT的值会在事务结束后发生变化。

I2STAT – I²C 状态

寄存器	特殊功能寄存器地址	复位值
I2STAT	BDH, SFR页0	1111 1000 b

7	6	5	4	3	2	1	0
I2STAT[7:3]					**0**	**0**	**0**
只读					只读	只读	只读

位	名称	描述
7:3	**I2STAT[7:3]**	**I²C 状态码** I2STAT的高5位为状态码，共有27种可能的状态值。 I2STAT = F8H时，没有相关状态信息，SI 标志保持为 0。其余26种状态码对应I²C的状态。当进入这些状态时，SI 将被置为逻辑1并且产生中断请求。
2:0	**0**	**保留** I2STAT的低有效三位始终读到 0。

I2CLK – I²C时钟

寄存器	特殊功能寄存器地址	复位值
I2CLK	BEH, SFR页0	0000_1001 b

7	6	5	4	3	2	1	0
I2CLK[7:0]							
读/写							

位	名称	描述
7:0	**I2CLK[7:0]**	**I²C 时钟设置** 主机模式下: 此寄存器设定作主机时I²C总线的时钟速率。时钟速率遵循公式如下： $\frac{F_{SYS}}{4 \quad (I2CLK+1)}$. 如果I²C的外设时钟频率是16MHz，默认状态下，I²C的总线的时钟频率为400Kbps。 注意：I2CLK 值 00H 及 01H 无效。这是一个限制。 从机模式下: 该字节无效，从机自动与给定的时钟速率同步，最高可达400k bps。

I2TOC – I²C超时计数器

寄存器	特殊功能寄存器地址	复位值
I2TOC	BFH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	I2TOCEN	DIV	I2TOF
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	I2TOCEN	**I²C超时计数器使能** 0 = I²C超时计数器禁用 1 = I²C超时计数器使能
1	DIV	**I²C超时计数器时钟分频** 0 = I²C超时计数器时钟为 F_{SYS}/1 1 = I²C超时计数器时钟为 F_{SYS}/4
0	I2TOF	**I²C超时标志** 如果14位I²C超时计数器溢出，此位由硬件置位。此位由软件清0。

I2CON – I²C控制

寄存器	特殊功能寄存器地址	复位值
I2CON	C0H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
I2CHTEX	**I2CEN**	**STA**	**STO**	**SI**	**AA**	**-**	**I2CPX**
读/写	读/写	读/写	读/写	读/写	读/写	-	读/写

位	名称	描述
7	**I2CHTEX**	**I2C0 保持时间延长使能** 0 = I²C 数据传输到I2C0_SCL保持时间延长禁用. 1 = I²C 数据传输到I2C0_SCL保持时间延长使能，延长8个系统时钟。
6	**I2CEN**	**I²C总线使能** 0 = I²C总线禁用. 1 = I²C总线使能. 使能 I²C前, I2C0_SCL和I2C0_SDA口应被设为逻辑1.
5	**STA**	**起始标志START** 当STA置1，如果总线空闲，I²C产生START起始条件，如果总线忙，I²C等待STOP停止条件，随后产生START起始条件 如果I²C总线已经在主机模式且已发送或接收了一个或多个字节，此时再设定STA，I²C 总线将产生repeated START重复起始条件 注意：STA可在任何时间置位，甚至是从机模式。但在检测到 START 起始条件或重复起始条件后，STA 不会由硬件自动清除。用户需自行注意手动清除STA。
4	**STO**	**停止标志STOP** 当I²C 在主机模式下，STO置1时，会向总线发送STOP停止条件。一旦总线上STOP被侦测，STO由硬件自动清0。 STO标志置位也用于将I²C设备从总线错误状态(I2STAT为00H)中恢复。在这种情况下，不会向I²C总线发送STOP停止条件。 如果STA和STO同时置1，且在主机模式下，I²C总线将产生STOP之后紧接着产生START。如果在从机模式下，应避免STA及STO同时置1，以免发出非法的I²C帧。
3	**SI**	**I²C中断标志位** 进入26种可能的I²C状态之一(F8H 状态除外)时，硬件就会置位此位。SI置位后，软件应读取I2STAT寄存器，以确定已完成哪一步，并采取下一步骤。 SI由软件清0。在SI清零之前，I2C0_SCL低电平时间延长，事务将被暂停。这有助于从机处理先前的数据字节，直到准备好接收下一个字节。 串行事务暂停，直到SI 位被软件清0。SI清除后， I²C总线将继续生成START条件或重复起始条件，STOP停止条件，8位数据等等，具体取决于软件配置的控制字节或者位段设定。因此用户应在软件清除 SI 之前给寄存器合适的设定。

位	名称	描述
2	**AA**	**确认响应标志位** I^2C设备为接收器或自身地址匹配的从机时 如果设置了 AA 标志，在 I2C0_SCL 线的确认时钟脉冲期间返回应答 ACK(I2C0_SDA 上的低电平)。 如果清除了 AA 标志，在I2C0_SCL 线的确认时钟脉冲期间返回非应答NACK(I2C0_SDA 上的高电平)。清除自身 AA 标志的设备将忽略其自身的从机地址和广播调用。因此，SI将不会被断言，也不会请求中断。 注意： 若已被寻址的从机，在从机接收模式下未返回ACK应答，或在从机发送模式下未接收到ACK应答，此从机将变为未寻址从机，无法接收任何数据直到其AA被置1，且主机再次对其进行寻址。 在从机发送模式下，I2STAT 值会变为 C8H，这是一种特殊情况。在从机向主机发送完最后一个数据字节之前，可以将 AA 标志位清零，这样在发送完最后一个数据字节后，从机就会主动切换到未寻址从机模式，断开与主机的连接。主机后续读取的数据将全部为 FFH。
0	**I2CPX**	**I^2C 引脚选择** 0 = 将I2C0_SCL分配给 P1.3，将I2C0_SDA分配给 P1.4。 1 = 将I2C0_SCL分配给 P0.2，将I2C0_SDA分配给 P1.6。 注意，一旦设置或清除此位，I^2C 引脚将立即更换。

I2ADDR – I²C 自身从机地址

寄存器	特殊功能寄存器地址	复位值
I2ADDR	C1H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
I2ADDR[7:1]							GC
读/写							读/写

位	名称	描述
7:1	I2ADDR[7:1]	**I²C 设备的从机地址** 主机模式: 此位段无效 从机模式: 这 7 位由用户定义此 I²C 设备的从机地址。主机应在起始条件或重复起始条件后的第一个字节数据中发送相同的地址来寻址 I²C设备。如果设置了 AA 标志，I²C设备接收到自己的地址后将响应主机，成为可寻址的从机。否则主机呼叫寻址会被忽略。 注意：I2ADDR[7:1] 不应保持全0的默认值，因为0x00保留用于广播呼叫。
6	GC	**广播呼叫位** 主机模式： 此位无效 从机模式： 0 =广播呼叫始终被忽略。 1 = 如果 AA 标志为 1，则识别广播呼叫；否则，如果 AA 为 0，则忽略广播呼叫。

ADCRL – ADC 转换结果低位

寄存器	特殊功能寄存器地址	复位值
ADCRL	C2H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	ADCR[3:0]			
-	-	-	-	只读			

位	名称	描述
3:0	**ADCR[3:0]**	**ADC结果低字节** ADC转换结果的最低有效 4 位存储在此寄存器中。

ADCRH – ADC 转换结果高字节

寄存器	特殊功能寄存器地址	复位值
ADCRH	C3H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
ADCR[11:4]							
只读							

位	名称	描述
7:0	**ADCR[11:4]**	**ADC 转换结果高字节** ADC 转换结果的最高有效8 位存储在此寄存器中。

T3CON – 定时器3控制

寄存器	特殊功能寄存器地址	复位值
T3CON	C4H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SMOD_1	**SMOD0_1**	**BRCK**	**TF3**	**TR3**	**T3PS[2:0]**		
读/写	读/写	读/写	读/写	读/写	读/写		

位	名称	描述
7	**SMOD_1**	**串口1双倍波特率使能** 当 UART1 处于模式 2 时，置位此位可使串口1波特率加倍。详细说明参见表6.9-2 串口UART1 模式/波特率。
6	**SMOD0_1**	**串口1帧错误访问使能位** 0 = SCON_1.7访问 SM0_1 位。 1 = SCON_1.7访问 FE_1 位。
5	**BRCK**	**串口0波特率时钟源** 当串行口0处于模式1或3时，此位选择使用哪个定时器作为波特率时钟源。 0 = 定时器1。 1 = 定时器3。
4	**TF3**	**定时器3溢出标志** 当定时器3溢出时，此位置位。程序执行定时器3中断服务程序时，硬件自动清零此位。此位可由软件置位或清零。
3	**TR3**	**定时器3运行控制** 0 = 定时器3停止。 1 = 定时器3开始运行。 注意，重载寄存器RH3和RL3仅能在定时器3停止(TR3 位为 0)时写入。如果在TR3为1时写入 RH3或RL3中的任何一个，结果将不可预测。
2:0	**T3PS[2:0]**	**定时器3预分频** 这些位段决定定时器3的时钟分频。 000 = 1/1. 001 = 1/2. 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

RL3 – 定时器3重载低字节

寄存器	特殊功能寄存器地址	复位值
RL3	C5H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
RL3[7:0]							
读/写							

位	名称	描述
7:0	**RL3[7:0]**	**定时器3重载低字节** 它保存了定时器3重载值的低字节。

RH3 – 定时器3重载高字节

寄存器	特殊功能寄存器地址	复位值
RH3	C6H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
RH3[7:0]							
读/写							

位	名称	描述
7:0	**RH3[7:0]**	**定时器3重载高字节** 它保存了定时器3重载值的高字节。

PIOCON1 – PWM或者I/O选择

寄存器	特殊功能寄存器地址	复位值
PIOCON1	C6H, SFR页1	0000_0000 b

7	6	5	4	3	2	1	0
PIO17	**-**	**PIO15**	**-**	**PIO04**	**PIO05**	**PIO14**	**-**
读/写	-	读/写	-	读/写	读/写	读/写	-

位	名称	描述
7	**PIO17**	**P1.7/PWM 引脚功能选择** 0 = P1.7/PWM3_CH0 引脚用作GPIO P1.7. 1 = P1.7/PWM3_CH0 引脚用作 PWM3_CH0输出。
6	**-**	保留
5	**PIO15**	**P1.5/PWM引脚功能选择** 0 = P1.5/PWM 引脚用作P1.5. 1 = P1.5/PWM 引脚用作PWM0_CH5 / PWM3_CH1输出。 当AUXR5[3:2].PWM3CH1P = 00, PIO15 = 1, P1.5 定义为 PWM0_CH5. AUXR5[3:2].PWM3CH1P = 11, PIO15 = 1, P1.5定义为 PWM3_CH1.
4	**-**	保留
3	**PIO04**	**P0.4/PWM引脚功能选择** 0 = P0.4/PWM 引脚用作P0.4. 1 = P0.4/PWM 引脚用作PWM0_CH3 / PWM2_CH1输出。 当AUXR4[7:6].PWM2CH1P = 00, PIO04 = 1, P0.4定义为 PWM0_CH3. AUXR4[7:6].PWM2CH1P = 11, PIO04 = 1, P0.4定义为 PWM2_CH1.
2	**PIO05**	**P0.5/PWM引脚功能选择** 0 = P0.5/PWM 引脚用作P0.5. 1 = P0.5/PWM 引脚用作PWM0_CH2 / PWM2_CH0输出。 当AUXR4[5:4].PWM2CH1P = 00, PIO05 = 1, P0.5定义为 PWM0_CH2. AUXR4[5:4].PWM2CH1P = 11, PIO05 = 1, P0.5定义为 PWM2_CH0.
1	**PIO14**	**P1.4/PWM引脚功能选择** 0 = P1.4/PWM 引脚用作P1.4. 1 = P1.4/PWM 引脚用作PWM0_CH1 / PWM1_CH1输出。 当AUXR4[3:2]. PWM1CH1P = 00, PIO14 = 1, P1.4定义为 PWM0_CH1. AUXR4[3:2]. PWM1CH1P = 10, PIO14 = 1, P1.4定义为 PWM1_CH1.
0	**-**	保留

PIOCON2 – PWM或者I/O选择

寄存器	特殊功能寄存器地址	复位值
PIOCON2	B7H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
PIO34	PIO33	PIO32	PIO31	PIO30	PIO23	PIO22	PIO21
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**PIO34**	**P3.4/PWM 引脚功能选择** 0 = P3.4/PWM 引脚用作P3.4. 1 = P3.4/PWM 引脚用作PWM3_CH1输出.
6	**PIO33**	**P3.3/PWM 引脚功能选择** 0 = P3.3/PWM 引脚用作P3.3. 1 = P3.3/PWM 引脚用作PWM0_CH0输出.
5	**PIO32**	**P3.2/PWM 引脚功能选择** 0 = P3.2/PWM 引脚用作P3.2. 1 = P3.2/PWM 引脚用作PWM3_CH0输出.
4	**PIO31**	**P3.1/PWM 引脚功能选择** 0 = P3.1/PWM 引脚用作P3.1. 1 = P3.1/PWM 引脚用作PWM2_CH1输出.
3	**PIO30**	**P3.0/PWM 引脚功能选择** 0 = P3.0/PWM 引脚用作P3.0 1 = P3.0/PWM 引脚用作PWM2_CH1输出.
2	**PIO23**	**P23/PWM 引脚功能选择** 0 = P2.3/PWM 引脚用作P2.3. 1 = P2.3/PWM 引脚用作PWM1_CH0输出.
1	**PIO22**	**P2.2/PWM 引脚功能选择** 0 = P2.2/PWM 引脚用作P2.2. 1 = P2.2/PWM 引脚用作PWM1_CH1输出.
0	**PIO21**	**P2.1/PWM 引脚功能选择** 0 = P2.1/PWM 引脚用作P2.1. 1 = P2.1/PWM 引脚用作PWM2_CH0输出.

TA – 定时访问

寄存器	特殊功能寄存器地址	复位值
TA	C7H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
TA[7:0]							
只写							

位	名称	描述
7:0	TA[7:0]	**定时访问** 定时访问寄存器控制对受保护的 SFR 的访问。要访问受保护的位，用户首先往TA写入AAH，然后立即写入55H。这两步之后，将打开一个持续 4 个时钟周期的写允许窗口，在此期间用户可以写入受保护的 SFR。

T2CON – 定时器2控制

寄存器	特殊功能寄存器地址	复位值
T2CON	C8H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
TF2	**-**	**-**	**-**	**-**	**TR2**	**-**	**CM/$\overline{RL2}$**
读/写	–	–	–	–	读/写	–	读/写

位	名称	描述
7	**TF2**	**定时器2溢出标志** 当定时器2溢出或者发生比较匹配时，此位置位。如果定时器2中断和全局中断都已使能，则置位此位将使CPU执行定时器 2的中断服务程序。此位不会通过硬件自动清除，应通过软件清除。
2	**TR2**	**定时器2运行控制** 0 = 定时器2禁用。清除此位将暂停定时器2，当前计数将保存在TH2和TL2中。 1 = 定时器2使能。
0	**CM/$\overline{RL2}$**	**定时器2比较或自动重载模式选择** 此位选择定时器2的工作模式。 0 = 自动重新加载模式。 1 = 比较模式。

T2MOD – 定时器2模式

寄存器	特殊功能寄存器地址	复位值
T2MOD	C9H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
LDEN	T2DIV[2:0]			CAPCR	CMPCR	LDTS[1:0]	
读/写	读/写			读/写	读/写	读/写	

位	名称	描述
7	LDEN	**自动重载使能** 0 = 将RCMP2H和RCMP2L重载至TH2和TL2功能禁用。 1 = 将RCMP2H和RCMP2L重载至TH2和TL2功能使能。
6:4	T2DIV[2:0]	**定时器2时钟分频** 000 = 定时器2时钟分频为1/1. 001 = 定时器2时钟分频为1/4. 010 = 定时器2时钟分频为1/16. 011 = 定时器2时钟分频为1/32. 100 = 定时器2时钟分频为1/64. 101 = 定时器2时钟分频为1/128. 110 = 定时器2时钟分频为1/256. 111 = 定时器2时钟分频为1/512.
3	CAPCR	**捕获自动清除** 此位仅当定时器2自动重载模式下有效。当发生捕获事件时，TH2和TL2计数器内数据被传输到RCMP2H 和RCMP2L后，该位使能硬件自动清除TH2和TL2这两个计数寄存器。 0 = 捕获事件发生，定时器2继续计数。 1 = 捕获事件发生，定时器2内数值被自动清0
2	CMPCR	**比较匹配自动清除** 此位仅当定时器2比较功能模式下有效。当发生比较匹配时，该位使能硬件自动清除TH2和TL2计数器值。 0 = 发生比较匹配时，定时器2继续计数。 1 = 发生比较匹配时，定时器2内数值被自动清0。
1:0	LDTS[1:0]	**自动重载触发选择** 这些位选择重载触发事件。 00 = 定时器2溢出时重载。 01 = 发生输入捕获 0 事件时重新加载。 10 = 发生输入捕获1 事件时重新加载。 11 = 发生输入捕获 2 事件时重新加载。

RCMP2L – 定时器2重载/比较低字节

寄存器	特殊功能寄存器地址	复位值
RCMP2L	CAH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
RCMP2L[7:0]							
读/写							

位	名称	描述
7:0	**RCMP2L[7:0]**	**定时器2重载/比较低字节** 当定时器2设定为比较模式，此寄存器存储比较值的低字节。 在自动重载模式下，此寄存器存储重载值的低字节。

RCMP2H – 定时器2重载/比较高字节

寄存器	特殊功能寄存器地址	复位值
RCMP2H	CBH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
RCMP2H[7:0]							
读/写							

位	名称	描述
7:0	**RCMP2H[7:0]**	**定时器2重载/比较高字节** 当定时器2设定为比较模式，此寄存器存储比较值的高字节。 在自动重载模式下，此寄存器存储重载值的高字节。

TL2 – 定时器2低字节

寄存器	特殊功能寄存器地址	复位值
TL2	CCH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
TL2[7:0]							
读/写							

位	名称	描述
7:0	TL2[7:0]	**定时器2低字节** TL2 寄存器为定时器2的16位计数寄存器的低字节。

TH2 – 定时器2高字节

寄存器	特殊功能寄存器地址	复位值
TH2	CDH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
TH2[7:0]							
读/写							

位	名称	描述
7:0	TH2[7:0]	**定时器2高字节** TH2 寄存器为定时器2的16位计数寄存器的高字节。

ADCMPL – ADC比较值低字节

寄存器	特殊功能寄存器地址	复位值
ADCMPL	CEH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	ADCMP[3:0]			
-	-	-	-	读/写			

位	名称	描述
3:0	**ADCMP[3:0]**	**ADC比较值低字节** ADC比较值的最低有效4位存储在此寄存器中。

EIP2 – 扩展中断优先级2

寄存器	特殊功能寄存器地址	复位值
EIP2	CEH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	PSC2	PSC1	PSC0
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	**PSC2**	SC2 / UART4 中断优先级低位
1	**PSC1**	SC1 / UART3 中断优先级低位
0	**PSC0**	SC0 / UART2 中断优先级低位
注意：EIP2 与 EIPH2 配合使用，用于确定每个中断源的优先级。有关正确的中断优先级配置，请参阅表6.2-4 中断优先级层级设置 。		

ADCMPH – ADC比较值高字节

寄存器	特殊功能寄存器地址	复位值
ADCMPH	CFH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
ADCMP[11:4]							
读/写							

位	名称	描述
7:0	**ADCMP[11:4]**	**ADC比较值高字节** ADC比较值的最低有效4位存储在此寄存器中。

EIPH2 – 扩展中断优先级高字节2

寄存器	特殊功能寄存器地址	复位值
EIPH2	CFH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	PSC2H	PSC1H	PSC0H
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	**PSC2H**	SC2 / UART4 中断优先级高位
1	**PSC1H**	SC1 / UART3 中断优先级高位
0	**PSC0H**	SC0 / UART2 中断优先级高位
注意：EIPH2与 EIP2 配合使用，用于确定每个中断源的优先级。有关正确的中断优先级配置，请参阅表6.2-4 中断优先级层级设置 。		

PSW – 程序状态寄存器

寄存器	特殊功能寄存器地址	复位值
PSW	D0H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
CY	**AC**	**F0**	**RS1**	**RS0**	**OV**	**F1**	**P**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	只读

位	名称	描述
7	CY	**进位标志** 对于加减法运算，如果前一次运算导致最高位发生进位或借位时，CY将被置位，否则清零。 对于乘除法运算，CY始终为0。 CY受DAA指令影响，该指令指示原始BCD和是否大于100。 对于CJNE分支，如果第一个无符号整数的值小于第二个无符号整数值，则CY置位，否则清0。
6	C	**辅助进位标志** 如果前一次运算导致低半字节的第4位发生进位或借位，此位置位，否则清零。
5	0	**用户标志0** 可由用户置位或清零的通用标志0。
4	RS1	**寄存器组选择位** 这两位用来选择R0到R7所在的四个寄存器组之一 RS1 RS0 寄存器组 RAM 地址 0 0 0 00H 至 07H 0 1 1 08H 至 0FH 1 0 2 10H 至 17H 1 1 3 18H 至 1FH
3	RS0	
2	V	溢出标志 OV用于有符号字符操作数。对于ADD或ADDC加法指令，如果位6有进位而位7没有进位，或者位7有进位而位6没有进位，则OV置位，反之OV 清0。OV 也用于两个正操作数相加得到的负数或两个负操作数相加得到的正数。对于SUBB减法指令，如果需要借位到位6而不是位7，或者需要借位到位7而不是位6，则OV置位，反之OV 清0。OV 也用于从正值中减去负值得到的负数，或者从负数中减去正数得到的正结果。 对于MUL乘法，如果乘积大于 255(00FFH)，则OV置位，反之OV 清0。 对于DIV除法，它通常为 0。但是，如果B最初包含00H，则A和B的返回值不确定。同时，OV将被置位。
1	1	**用户标志1** 可由用户置位或清零的通用标志1。

位	名称	描述
0	P	**奇偶校验标志** 置位表示累加器中 1 的个数为奇数。清零表示累加器中 1 的个数为偶数。执行偶校验。

PWMnPH – PWMn周期高字节, n = 0,1,2,3

寄存器	特殊功能寄存器地址	复位值
PWM0PH	D1H, SFR页0	0000_0000 b
PWM1PH	A9H, SFR页2	0000_0000 b
PWM2PH	B9H, SFR页2	0000_0000 b
PWM3PH	C9H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnP [15:8]							
读/写							

位	名称	描述
7:0	**PWMnP [15:8]**	**PWMn 周期高字节** 此字节与 PWMnPL一起控制PWMn发生器的信号周期。

PNP – PWM负极性

寄存器	特殊功能寄存器地址	复位值
PNP	D6H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
n	PNPn	**PWM0_CHn 负极性输出使能** 0 = PWM0_CHn 信号直接输出到 PWM0_CHn 引脚。 1 = PWM0_CHn 信号反相输出到PWM0_CHn引脚。

PWM0FBD – PWM 故障刹车数据

寄存器	特殊功能寄存器地址	复位值
PWM0FBD	D7H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
FBF	**FBINLS**	**FBD5**	**FBD4**	**FBD3**	**FBD2**	**FBD1**	**FBD0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**FBF**	**故障刹车标志位** 当FBINEN设置为1，且FB引脚检测到符合FBINLS(FBD.6)选择的边沿信号时，此位置1。此位需通过软件清0。FBF清除后，故障刹车数据输出不会被释放，直到PWMRUN(PWMCON0.7)被置位。
6	**FBINLS**	**PWM_BRAKE引脚输入电平选择** 0 = 下降沿。 1 = 上升沿。
N	**FBDn**	**PWM0_CHn故障刹车数据** 0 = 当故障刹车有效后， PWM0_CHn 信号将被 0 覆盖。 1 = 当故障刹车有效后， PWM0_CHn 信号将被 1 覆盖。

XTLCON – XLT 时钟控制 (TA 保护)

寄存器	特殊功能寄存器地址	复位值
XTLCON	D7H, SFR页2	0010_1000 b

7	6	5	4	3	2	1	0
-	HXSG						
-	读/写						

位	名称	描述
7	-	保留
6:0	**HXSG**	**HXT增益值选择** 增益控制用于增大晶体的增益，以确保晶体正常工作。如果启用增益控制，晶体的功耗将比关闭增益控制时更高。 建议： 000_0000 = L0。HXT频率为4 MHz (最小值). 000_1100 = L1。频率范围为4 MHz ~ 12 MHz. 001_1000 = L2。频率范围为12MHz ~ 16 MHz. 010_1000 = L3。频率范围为16MHz ~ 24 MHz. 111_1111 = 保留 (最大值). 其他 = 保留.

SCnDR – SC 数据寄存器

寄存器	特殊功能寄存器地址	复位值
SC0DR	D9H, SFR页2	0000_0000 b
SC1DR	E1H, SFR页2	0000_0000 b
SC2DR	E9H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
SCnDR[7:0]							
读/写							

位	名称	描述
7:0	SCnDR[7:0]	**SC / UART缓冲区数据** 此字节用于在SC / UART总线上发送或接收数据。写入此字节实际上是写入移位寄存器。读取此字节实际上是读取读数据缓冲区。 **注意：** 如果SCEN(SCnCR0[0]) 未使能，则SCnDR 不能被编程。

SCnEGT – SC 额外保护时间寄存器

寄存器	特殊功能寄存器地址	复位值
SC0EGT	DAH, SFR页2	0000_0000 b
SC1EGT	E2H, SFR页2	0000_0000 b
SC2EGT	EAH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
SCnEGT[7:0]							
读/写							

位	名称	描述
7:0	SCnEGT[7:0]	**SC额外保护时间** 此字段表示额外保护定时器的值。 **注意：**该计数器以 ETU 为基数。

SCnETURD0 – SCn ETU 速率分频器寄存器

寄存器	特殊功能寄存器地址	复位值
SC0ETURD0	DBH, SFR页2	0111_0011 b
SC1ETURD0	E3H, SFR页2	0111_0011 b
SC2ETURD0	EBH, SFR页2	0111_0011 b

7	6	5	4	3	2	1	0
ETURDIV[7:0]							
读/写							

位	名称	描述
7:0	**ETURDIV[7:0]**	**ETU 速率分频器的 LSB位** 该字段指示时钟速率分频器的 LSB。 实际 ETU为ETURDIV[11:0] + 1. **注1:** ETURDIV[11:0] 必须大于 0x004. **注2:** 必须先编程SCnETURD0，然后再编程 SCnETUDR2。

SCnETURD1 –SC ETU 速率分频器寄存器 1

寄存器	特殊功能寄存器地址	复位值
SC0ETURD1	DCH, SFR页2	0011_0001 b
SC1ETURD1	E4H, SFR页2	0011_0001 b
SC2ETURD1	ECH, SFR页2	0011_0001 b

7	6	5	4	3	2	1	0
-	SCDIV[2:0]			ETURDIV[11:8]			
-	读/写			读/写			

位	名称	描述
7	-	保留
6:4	SCDIV [2:0]	**SC 时钟分频器** 000 = F_{SC}为F_{SYS}/1. 001 = F_{SC}为F_{SYS}/2. 010 = F_{SC}为F_{SYS}/4. 011 = F_{SC}为F_{SYS}/8. (默认) 100 = F_{SC}为F_{SYS}/16. 101 = F_{SC}为F_{SYS}/16. 110 = F_{SC}为F_{SYS}/16. 111 = F_{SC}为F_{SYS}/16. **注意：**根据ISO/IEC 7816-3标准，F_{SC}时钟应为1Mhz ~ 5Mhz
3:0	ETURDIV [11:8]	**ETU 速率分频器的 MSB 位** 该字段表示时钟速率分频器的 MSB。 实际 ETU为ETURDIV[11:0] + 1. **注1:** ETURDIV[11:0] 必须大于 0x004. **注2:** 必须先编程SCnETURD0，然后再编程 SCnETUDR2。

ScnIE – SC中断使能控制寄存器

寄存器	特殊功能寄存器地址	复位值
SC0IE	DDH, SFR页2	0000_0000 b
SC1IE	E5H, SFR页2	0000_0000 b
SC2IE	EDH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN
-	-	-	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:4	-	保留
4	ACERRIEN	**自动约定错误中断使能位** 此字段用于使能自动约定错误中断。 0 = 自动约定错误中断禁用。 1 = 自动约定错误中断使能。
3	BGTIEN	**区块保护时间中断使能位** 此字段用于使能区块保护时间中断。 0 = 区块保护时间中断禁用。 1 = 区块保护时间中断使能。
2	TERRIEN	**传输错误中断使能位** 此字段用于使能传输错误中断。传输错误状态位于SC0TSR 寄存器中，包括接收器中断错误BEF(SC0TSR[6])，帧错误FEF(SC0TSR[5])，奇偶校验错误PEF(SC0TSR[4])，接收器缓冲区溢出错误RXOV(SC0TSR[0]) 和发送缓冲区溢出错误TXOV(SC0TSR[2]). 0 = 传输错误中断禁用。 1 = 传输错误中断使能。
1	TBEIEN	**发送缓冲区空中断使能位** 此字段用于使能发送缓冲区空中断。 0 = 禁用发送缓冲区空中断。 1 = 启用发送缓冲区空中断。
0	RDAIEN	**接收数据到达中断使能位** 此字段用于使能接收数据中断。 0 = 禁用接收数据中断。 1 = 启用接收数据中断。

ScnIS – SC 中断状态寄存器

寄存器	特殊功能寄存器地址	复位值
SC0IS	DEH, SFR页2	0000_0010 b
SC1IS	E6H, SFR页2	0000_0010 b
SC2IS	EEH, SFR页2	0000_0010 b

7	6	5	4	3	2	1	0
-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF
-	-	-	读/写	读/写	只读	只读	只读

位	名称	描述
7:5	-	保留
4	ACERRIF	**自动约定错误中断状态标志** 此字段指示自动约定序列错误。如果在 ATR 状态下接收到的 TS 既不是 0x3B 也不是 0x3F，则此位只能由硬件置位。 **注意：** 写入 0 即可清除。
3	BGTIF	**区块保护时间中断状态标志** 此字段用于指示区块保护时间中断状态，此位只能由硬件置位。 **注1:** 当 RXBGTEN (SCnCR0[5]) 使能时，此位有效。 **注2:** 写入 0 即可清除。
2	TERRIF	**传输错误中断状态标志（只读）** 此字段用于指示传输错误中断状态。传输错误状态位于 SC0TSR 寄存器中，包括接收器中断错误BEF(SC0TSR[6])，帧错误FEF(SC0TSR[5])，奇偶校验错误PEF(SC0TSR[4])，接收器缓冲区溢出错误RXOV(SC0TSR[0]) 和发送缓冲区溢出错误TXOV(SC0TSR[2]). **注意：** 此字段是BEF(SC0TSR[6]), FEF(SC0TSR[5]), PEF(SC0TSR[4]), RXOV(SC0TSR[0]) 和 TXOV(SC0TSR[2]) 的状态标志。因此，如果软件想要清除此位，必须向每个字段写入“0”。
1	TBEIF	**发送缓冲区空中断状态标志（只读）** 此字段用于发送缓冲区空中断状态标志。 **注意：** 此字段是发送缓冲区空状态的状态标志。如果软件要清除此位，必须将数据写入 DAT(SCnDR[7:0]) 缓冲区，然后此位将自动清除。
0	RDAIF	**接收数据到达中断状态标志（只读）** 此字段用于接收数据中断状态标志。 **注意：** 此字段是接收数据的状态标志。如果软件从 SC_DAT 引脚读取数据，此位将自动清除。

SCnTSR – SC 传输状态寄存器

寄存器	特殊功能寄存器地址	复位值
SC0TSR	DFH, SFR页2	0000_1010 b
SC1TSR	E7H, SFR页2	0000_1010 b
SC2TSR	EFH, SFR页2	0000_1010 b

7	6	5	4	3	2	1	0
ACT	**BEF**	**FEF**	**PEF**	**TXEMPTY**	**TXOV**	**RXEMPTY**	**RXOV**
只读	读/写	读/写	读/写	只读	读/写	只读	读/写

位	名称	描述
7	**ACT**	**发送/接收活动状态标志（只读）** 0 = 发送/接收传输完成后，此位自动清零 1 = 发送/接收传输处于活动状态时，此位由硬件置位
6	**BEF**	**接收器间隔错误状态标志** 仅当接收数据输入 (RX) 保持在“间隔状态”（逻辑 0）的时间长于一个完整字的传输时间（即“起始位”+数据位+奇偶校验位+停止位的总时间）时，此位才会设置为逻辑 1。 **注意：** 此位是只读的，但可以通过写入 0 来清除。
5	**FET**	**接收器帧错误状态标志** 仅当接收到的字符没有有效的“停止位”（即，检测到最后一个数据位或奇偶校验位后的停止位为逻辑0）时，此位才会设置为逻辑1。 **注意：** 此位是只读的，但可以通过写入0来清除。
4	**PEF**	**接收器奇偶校验错误状态标志** 仅当接收到的字符没有有效的“奇偶校验位”时，此位才会设置为逻辑1。 **注意：** 此位是只读的，但可以通过写入0来清除。
3	**TXEMPTY**	**发送缓冲区空状态标志（只读）** 此位指示发送缓冲区是否为空。 **注意：** 当发送缓冲区已传输至发送移位寄存器时，硬件会将此位置1。将数据写入 DAT(SCnDR[7:0])（发送缓冲区非空）时，此位置1。
2	**TXOVF**	**TX 溢出错误中断状态标志** 如果 TX 缓冲区已满，再次写入 DAT(SCnDR[7:0]) 将导致该位仅由硬件设置为“1”。 **注意：** 该位为只读，但可以通过写入 0 来清除。
1	**RXEMPTY**	**接收缓冲区空状态标志（只读）** 该位指示 RX 缓冲区是否为空。 **注意：** 当 CPU 读取 Rx 缓冲区后，硬件会将该位置高。当 SC 接收到任何新数据时，该位置高。

位	名称	描述
0	**RXOV**	**RX 溢出错误状态标志** 该位仅在 RX 缓冲区溢出时设置。 **注意：** 该位为只读，但可以通过写入 0 来清除。

PWMnCON0 – PWMn控制 0, n = 1,2,3

寄存器	特殊功能寄存器地址	复位值
PWM1CON0	B4H, SFR页2	0000_0000 b
PWM2CON0	C4H, SFR页2	0000_0000 b
PWM3CON0	D4H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
PWMRUN	**LOAD**	**PWMF**	**CLRPWM**	**-**	**-**	**-**	**-**
读/写	读/写	读/写	读/写	-	-	-	-

位	名称	描述
7	**PWMRUN**	**PWM运行使能** 0 = PWM保持空闲。 1 = PWM开始运行。
6	**LOAD**	**PWM新周期和占空比加载** 需要更新新的周期或占空比值时，此位用于将周期和占空比控制寄存器加载到其缓冲区中。加载动作将在PWM周期完成时进行。新的周期及占空比值将作用于下一个周期。加载完成后，硬件自动将LOAD位清0。写入和读取 LOAD 位的含义不同。 写入: 0 = 无影响 1 = 在 PWM 周期完成时，将新的周期和占空比加载到其缓冲区中。 读取: 0 = 新周期和占空比载入已完成 1 = 新周期和占空比载入尚未完成
5	**PWMF**	**PWM标志位** 此位根据PWMINTC的INTSEL[2:0]及 INTTYP[1:0] 的定义来设定。此位由软件清零。
4	**CLRPWM**	**清除PWM计数器** 设置此位将清除 PWM 16 位计数器的值，并将其重置为 0000H。计数器值清除后，此位会被硬件自动清0。写入和读取CLRPWM位的意义不同。 写入： 0 = 无影响。 1 = 清除PWM 16位计数器。 读取： 0 =PWM16 位计数器已完全清零。 1 =PWM16 位计数器尚未清零。

PWM0CON0 – PWM0 控制0 (可位寻址)

寄存器	特殊功能寄存器地址	复位值
PWM0CON0	D8H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PWM0RUN	**LOAD**	**PWMF**	**CLRPWM**	**PWM3RUN**	**PWM2RUN**	**PWM1RUN**	**P33FBINEN**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**PWM0RUN**	**PWM0运行使能** 0 = PWM0保持空闲。 1 = PWM0开始运行。 **注意：** 此位仅适用于 PWM0CON0。
6	**LOAD**	**PWM新周期和占空比加载** 需要更新新的周期或占空比值时，此位用于将周期和占空比控制寄存器加载到其缓冲区中。加载动作将在PWM周期完成时进行。新的周期及占空比值将作用于下一个周期。加载完成后，硬件自动将LOAD位清0。写入和读取 LOAD 位的含义不同。 写入: 0 = 无影响 1 = 在 PWM 周期完成时，将新的周期和占空比加载到其缓冲区中。 读取: 0 = 新周期和占空比载入已完成 1 = 新周期和占空比载入尚未完成
5	**PWMF**	**PWM标志位** 此位根据PWM0INTC的INTSEL[2:0]及 INTTYP[1:0] 的定义来设定。此位由软件清零。
4	**CLRPWM**	**清除PWM计数器** 设置此位将清除 PWM 16 位计数器的值，并将其重置为 0000H。计数器值清除后，此位会被硬件自动清0。写入和读取CLRPWM位的意义不同。 写入： 0 = 无影响。 1 = 清除PWM 16位计数器。 读取： 0 =PWM16 位计数器已完全清零。 1 =PWM16 位计数器尚未清零。
3	**PWM3RUN**	**PWM3 运行使能** 0 = PWM3 保持空闲状态。 1 = PWM3 开始运行。 **注意：**此位仅适用于 PWM0CON0。

位	名称	描述
2	**PWM2RUN**	**PWM2 运行使能** 0 = PWM2 保持空闲状态。 1 = PWM2 开始运行。 **注意：**此位仅适用于 PWM0CON0。
1	**PWM1RUN**	**PWM1 运行使能** 0 = PWM1 保持空闲状态。 1 = PWM1 开始运行。 **注意：**此位仅适用于 PWM0CON0。
0	**P33FBINEN**	**P33 FB 引脚输入使能** 0 = 由 P33 FB 引脚输入导致的PWM0 输出故障刹车禁用。 1 = 由 P33 FB 引脚输入导致的PWM0 输出故障刹车使能。一旦 FB 引脚出现 FBINLS (FBD.6) 选择的边沿，PWM0C0~C5 将在 FBD 寄存器中输出故障刹车数据。PWMRUN (PWM0CON0.7) 也将由硬件自动清零。再次设置 PWMRUN 后，PWM 输出将恢复。 **注意：**此位仅适用于 PWM0CON0。

PWMnPL – PWMn周期低字节, n = 0,1,2,3

寄存器	特殊功能寄存器地址	复位值
PWM0PL	D9H, SFR页0	0000_0000 b
PWM1PL	B1H, SFR页2	0000_0000 b
PWM2PL	C1H, SFR页2	0000_0000 b
PWM3PL	D1H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnP[7:0]							
读/写							

位	名称	描述
7:0	**PWMnP[7:0]**	**PWMn 周期低字节** 此字节与 PWMnPH一起控制PWMn发生器的信号周期。

PWMnCxH – PWMn 通道x 占空比高字节, n = 0,1,2,3; x = 0,1,2,3,4,5

寄存器	特殊功能寄存器地址	描述	复位值
PWM0C0H	D2H, SFR页0	PWM0 通道 0占空比高字节	0000_0000 b
PWM0C1H	D3H, SFR页0	PWM0 通道1占空比高字节	0000_0000 b
PWM0C2H	D4H, SFR页0	PWM0 通道2占空比高字节	0000_0000 b
PWM0C3H	D5H, SFR页0	PWM0 通道3占空比高字节	0000_0000 b
PWM0C4H	C4H, SFR页1	PWM0 通道4占空比高字节	0000_0000 b
PWM0C5H	C5H, SFR页1	PWM0 通道5占空比高字节	0000_0000 b
PWM1C0H	AAH, SFR页2	PWM1 通道0占空比高字节	0000_0000 b
PWM1C1H	ABH, SFR页2	PWM1 通道1占空比高字节	0000_0000 b
PWM2C0H	BAH, SFR页2	PWM2 通道0占空比高字节	0000_0000 b
PWM2C1H	BBH, SFR页2	PWM2 通道1占空比高字节	0000_0000 b
PWM3C0H	CAH, SFR页2	PWM3 通道0占空比高字节	0000_0000 b
PWM3C1H	CBH, SFR页2	PWM3 通道1占空比高字节	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [15:8]							
读/写							

位	名称	描述
7:0	**PWMnCx [15:8]**	**PWMnCx 占空比高字节** 该字节与PWMnCxL一起控制 PWM 发生器输出信号 PGx 的占空比。

PWMnCxL – PWMn 通道 x 占空比低字节, n = 0,1,2,3; x = 0,1,2,3,4,5

寄存器	特殊功能寄存器地址	描述	复位值
PWM0C0L	DAH, SFR页0	PWM0 通道0占空比低字节	0000_0000 b
PWM0C1L	DBH, SFR页0	PWM0 通道1占空比低字节	0000_0000 b
PWM0C2L	DCH, SFR页0	PWM0 通道2占空比低字节	0000_0000 b
PWM0C3L	DDH, SFR页0	PWM0 通道3占空比低字节	0000_0000 b
PWM0C4L	CCH, SFR页1	PWM0 通道4占空比低字节	0000_0000 b
PWM0C5L	CDH, SFR页1	PWM0 通道5占空比低字节	0000_0000 b
PWM1C0L	B2H, SFR页2	PWM1 通道0占空比低字节	0000_0000 b
PWM1C1L	B3H, SFR页2	PWM1 通道1占空比低字节	0000_0000 b
PWM2C0L	C2H, SFR页2	PWM2 通道0占空比低字节	0000_0000 b
PWM2C1L	C3H, SFR页2	PWM2 通道1占空比低字节	0000_0000 b
PWM3C0L	D2H, SFR页2	PWM3 通道0占空比低字节	0000_0000 b
PWM3C1L	D3H, SFR页2	PWM3 通道1占空比低字节	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [7:0]							
读/写							

位	名称	描述
7:0	**PWMnCx [7:0]**	**PWMnCx 占空比低字节** 该字节与PWMnCxH一起控制 PWM 发生器输出信号 PGx 的占空比。

PIOCON0 – PWM或者 I/O选择

寄存器	特殊功能寄存器地址	复位值
PIOCON0	DEH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PIO03	PIO01	PIO00	PIO10	PIO11	PIO12
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:6	-	保留.
5	PIO03	**P0.3/PWM 引脚功能选择** 0 = P0.3/PWM 引脚用作P0.3. 1 = P0.3/PWM 引脚用作PWM0_CH5 / PWM3_CH1输出. 当AUXR5[7:6]. PWM3CH1P = 00, PIO03 = 1, P0.3 定义为PWM0_CH5. AUXR5[7:6]. PWM3CH1P = 11, PIO03 = 1, P0.3 定义为PWM3_CH1.
4	PIO01	**P0.1/PWM 引脚功能选择** 0 = P0.1/PWM 引脚用作P0.1. 1 = P0.1/PWM 引脚用作PWM0_CH4 / PWM3_CH0输出. 当AUXR5[5:4]. PWM3CH0P = 00, PIO01 = 1, P0.3 定义为PWM0_CH4. AUXR5[5:4]. PWM3CH0P = 10, PIO01 = 1, P0.3 定义为PWM3_CH0.
3	PIO00	**P0.0/PWM 引脚功能选择** 0 = P0.0/PWM 引脚用作P0.0. 1 = P0.0/PWM 引脚用作PWM0_CH3 / PWM2_CH1输出. 当AUXR4[7:6]. PWM2CH1P = 00, PIO00 = 1, P0.0 定义为PWM0_CH3. AUXR4[7:6]. PWM2CH1P = 10, PIO00 = 1, P0.0 定义为PWM2_CH1.
2	PIO10	**P1.0/PWM 引脚功能选择** 0 = P1.0/PWM 引脚用作P1.0. 1 = P1.0/PWM 引脚用作PWM0_CH2 / PWM2_CH0输出. 当AUXR4[5:4]. PWM2CH0P = 00, PIO10 = 1, P1.0 定义为PWM0_CH2. AUXR4[5:4]. PWM2CH0P = 10, PIO10 = 1, P1.0 定义为PWM2_CH0.
1	PIO11	**P1.1/PWM 引脚功能选择** 0 = P1.1/PWM 引脚用作P1.1. 1 = P1.1/PWM 引脚用作PWM0_CH1 / PWM1_CH1输出. 当AUXR4[3:2]. PWM1CH1P = 00, PIO11 = 1, P1.1 定义为PWM0_CH1. AUXR4[3:2]. PWM1CH1P = 11, PIO11 = 1, P1.1 定义为PWM1_CH1.

位	名称	描述
0	**PIO12**	**P1.2/PWM 引脚功能选择** 0 = P1.2/PWM 引脚用作P1.2. 1 = P1.2/PWM 引脚用作PWM0_CH0 / PWM1_CH0输出. 当AUXR4[1:0]. PWM1CH0P = 00, PIO12 = 1, P1.2 定义为PWM0_CH1. AUXR4[1:0]. PWM1CH0P = 10, PIO12 = 1, P1.2 定义为PWM1_CH0.

PWMnCON1 – PWMn 控制1

寄存器	特殊功能寄存器地址	复位值
PWM0CON1	DFH, SFR页0	0000_0000 b
PWM1CON1	B5H, 页2	0000_0000 b
PWM2CON1	C5H, 页2	0000_0000 b
PWM3CON1	D5H, 页2	0000_0000 b

7	6	5	4	3	2	1	0
PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		
读/写		读/写	读/写	读/写	读/写		

位	名称	描述
7:6	**PWMMOD[1:0]**	**PWM 模式选择** 00 = 独立输出模式 01 = 互补模式 10 = 同步模式 11 = 保留.
5	**GP**	**分组模式使能** 此位使能PWM分组模式。一旦使能，前三对PWM的占空比由PWM01H和PWM01L决定，而不是由它们原本的占空比控制寄存器决定。 0 = 分组模式禁用 1 = 分组模式使能 **注意：**此位仅在 PWM0 中有效。
4	**PWMTYP**	**PWM类型选择** 0 = 边沿对齐型PWM 1 = 中心对齐型PWM
3	**FBINEN**	**FB引脚输入使能** 0= FB引脚输入导致PWM输出故障刹车禁用。 1= FB引脚输入导致PWM输出故障刹车使能。一旦FB引脚上出现与FBINLS (PWM0FBD.6)选择匹配的边沿信号，PWM0通道0~5输出PWM0FBD寄存器所设置的故障刹车数据，PWM0RUN (PWM0CON0.7)由硬件自动清除。当PWM0RUN再次置位时，PWM0输出恢复。 **注意：**此位仅在 PWM0 中有效。

位	名称	描述
2:0	**PWMDIV[2:0]**	**PWM 时钟分频** 此字段决定PWM时钟源的预分频。 000 = 1/1. 001 = 1/2. 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

A or ACC – 累加器

寄存器	特殊功能寄存器地址	复位值
ACC	E0H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**ACC[7:0]**	**累加器** A或ACC 寄存器是用于算术运算的标准 80C51 累加器。

ADCCON1 – ADC控制1

寄存器	特殊功能寄存器地址	复位值
ADCCON1	E1H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
OCEN	STADCPX	ADCDIV[1:0]		ETGTYP[1:0]		ADCEX	ADCEN
读/写	读/写	读/写		读/写		读/写	读/写

位	名称	描述
7	**OCEN**	**此字段用于启用失调校准功能。** 0：ADC 失调校准已启用，由 ADC 硬件自动校准。 1：ADC 失调校准已禁用。
6	**STADCPX**	**外部启动ADC触发引脚选择** 0 = 将STADC分配给P0.4. 1 = 将STADC分配给P1.3. 注意：一旦设置或清除此位后，STADC脚会立即切换。
5:4	**ADCDIV[1:0]**	**ADCAQT0 和 1 时钟分频** 00 = ADCAQT 时钟源 = F_{SYS}/1. 01 = ADCAQT 时钟源 = F_{SYS}/2. 10 = ADCAQT 时钟源 = F_{SYS}/4. 11 = ADCAQT 时钟源 = F_{SYS}/8.
3:2	**ETGTYP[1:0]**	**外部触发类型选择** 当ADCEX(ADCCON1.1)置位时，这些位决定ADC 转换的触发条件。 00 = PWM0/2/4或STADC引脚的下降沿。 01 = PWM0/2/4或STADC引脚的上升沿。 10 = PWM周期的中心点。 11 = PWM周期的结束点。 请注意，中心点中断或周期点中断仅适用于PWM中心对齐类型。
1	**ADCEX**	**ADC外部转换触发器选择** 此位选择触发 A/D 转换的方法。 0 = 仅通过设置ADCS位来启动 A/D 转换。 1 = 通过设置ADCS位或由外部触发源启动A/D 转换，具体取决于 ETGSEL[1:0] 和 ETGTYP[1:0]。注意，当 ADCS 为1(转换忙)时，ADC 将忽略后续的外部触发，直到 ADCS 被硬件清零。
0	**ADCEN**	**ADC使能** 0 = ADC电路关闭。 1 = ADC电路开启。

ADCCON2 – ADC 控制2

寄存器	特殊功能寄存器地址	复位值
ADCCON2	E2H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
ADFBEN	**ADCMPOP**	**ADCMPEN**	**ADCMPO**	**ADCAQT1[2:0]**			**ADCDLY.8**
读/写	读/写	读/写	只读	读/写			读/写

位	名称	描述
7	**ADFBEN**	**ADC比较结果响应故障刹车使能** 0 = ADC触发故障刹车禁用 1 = ADC触发故障刹车功能使能。一旦比较结果 ADCMPO 为1，故障刹车启动，同时PWM输出故障刹车数据。PWM0RUN (PWM0CON0.7)也将由硬件自动清除。当PWM0RUN再次置位时，PWM输出恢复。
6	**ADCMPOP**	**ADC比较器输出极性** 0 =如果ADCR[11:0]大于或等于ADCMP[11:0]，则ADCMPO 为1。 1 =如果ADCR[11:0]小于ADCMP[11:0]，则 ADCMPO 为1。
5	**ADCMPEN**	**ADC结果比较器使能** ADC 结果比较器触发 ADCF使能位。仅当比较器值与定义的 ADC 比较值条件匹配时，ADCF 才会设置为1。此条件基于 ADCMPH、ADCMPL 和 ADCMPOP 寄存器的定义。 仅当ADC比较结果与条件匹配时，ADCF寄存器才会变为 1。如果ADC中断使能，则进入中断向量。 0 = ADC结果比较器触发ADCF禁用。 1 = ADC结果比较器触发ADCF使能。 **注意：**此位置位并触发ADC启动后，ADC持续转换。寄存器ADCRH和ADCRL的值将根据ADC的结果而改变，也可以从寄存器中读出。此过程仅在 ADCF 设置为 1 后停止。
4	**ADCMPO**	**ADC比较器输出值** 此位是基于ACMPOP设置的ADC结果比较器的输出值。每次A/D转换完成后，此位都会更新。
3:1	**ADCAQT1**	**ADC 采样时间0** 这3位字段决定ADC AIN0~AIN7 和BG采样的采样时间，遵循以下公式： ADC 采样时间 = $\frac{4*ADCAQT+6}{F_{ADCCLK}}$，其中 ADCAQT 在 ADCCON2[3:1] 中定义， $F_{ADCCLK}=\frac{F_{SYS}}{ADCDIV}$，其中 ADCDIV 在 ADCCON1[5:4] 中定义。 注意： ADC 转换过程中，此字段不可更改。 默认情况下，$F_{ADCCLK} = F_{SYS}$ (ADCDIV = 0)，最小采样时间为 6 个 ADC 时钟周期。由于最小采样时间必须超过 370ns，因此当 FSYS = 24MHz 时，ADCAQT 必须至少设置为 1。
0	**ADCDLY.8**	**ADC外部触发延迟计数器 第8位** 参见ADCDLY寄存器。

ADCCON3 – ADC 控制3

寄存器	特殊功能寄存器地址	复位值
ADCCON3	86H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	OVPEN	HIE	CONT	ADCAQT2[2:0]			-
-	读/写	读/写	读/写	读/写			-

位	名称	描述
7	-	保留
6	OVPEN	**ADC 过压保护手动使能** 0 = ADC 过压 (VDD > 7V) 保护禁用。 1 = ADC 过压 (VDD > 7V) 保护使能。 **注意：** 设置此位并等待 2us 以上，之后 ADCEN (ADCCON1[0]) 才会置位。
5	HIE	**ADC 半完成中断使能** 0 = 在继续模式下，当一半 A/D 转换完成时，不设置 ADC 中断 1 = 在继续模式下，当一半 A/D 转换完成时，设置 ADC 中断
4	CONT	**ADC 继续采样选择** 0 = ADC 单次采样，当一次 A/D 转换完成时设置 ADC 中断 1 = ADC 继续采样。当所有 A/D 转换完成时，设置 ADC 中断
3:1	ADCAQT2	**ADC 采样时间1** 这 3 位字段决定ADC AIN9~AIN15采样的采样时间，遵循以下公式： ADC 采样时间 = $\frac{4*ADCAQT+6}{F_{ADCCLK}}$, 其中 ADCAQT 在 ADCCON3[3:1] 中定义， $F_{ADCCLK}=\frac{F_{SYS}}{ADCDIV}$, 其中 ADCDIV 在 ADCCON1[5:4] 中定义。 注意： ADC 转换过程中，此字段不可更改。 默认情况下，$F_{ADCCLK} = F_{SYS}$ (ADCDIV = 0) ，最小采样时间为 6 个 ADC 时钟周期。由于最小采样时间必须超过 370ns，因此当 FSYS = 24MHz 时，ADCAQT 必须至少设置为 1。
0	-	**保留** 此位应保持为 0。

ADCDLY – ADC 触发延迟计数器

寄存器	特殊功能寄存器地址	复位值
ADCDLY	E3H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
ADCDLY[7:0]							
读/写							

位	名称	描述
7:0	**ADCDLY[7:0]**	**ADC外部触发延迟计数器低字节** 此 8 位字段与 ADCCON2.0组成一个9位计数器。该计数器在检测到外部触发后插入一段延迟时间。A/D 转换在此延迟时间后再开始。 外部触发延迟时间 = $\frac{ADCDLY}{F_{ADC}}$. 注意，此字段仅当ADCEX(ADCCON1.1)被置位才有效。如果选择PWM输出作为外部ADC触发源，用户不应在PWM运行期间修改ADCDLY。

CnL – 捕获n低字节, n = 0,1,2

寄存器	特殊功能寄存器地址	复位值
C0L	E4H, SFR页0	0000_0000 b
C1L	E6H, SFR页0	0000_0000 b
C2L	EDH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
CnL[7:0]							
读/写							

位	名称	描述
7:0	CnL[7:0]	**输入捕获n结果低字节** CnL寄存器是输入捕获n所捕获的16 位结果的低字节。

CnH – 捕获n高字节, n = 1,2,3

寄存器	特殊功能寄存器地址	复位值
C0H	E5H, SFR页0	0000_0000 b
C1H	E7H, SFR页0	0000_0000 b
C2H	EEH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
CnH[7:0]							
读/写							

位	名称	描述
7:0	**CnH[7:0]**	**输入捕获n结果高字节** CnH寄存器是输入捕获n所捕获的16 位结果的高字节。

ADCCON0 – ADC 控制0 (可位寻址)

寄存器	特殊功能寄存器地址	复位值
ADCCON0	E8H, 所有页	0000_0000b

7	6	5	4	3	2	1	0
ADCF	ADCS	ETGSEL1	ETGSEL0	ADCHS3	ADCHS2	ADCHS1	ADCHS0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**ADCF**	**ADC标志** 此标志在 A/D 转换完成时置位。ADC 结果可读取。当此标志为 1 时，ADC 无法启动新的转换。此位由软件清零。
6	**ADCS**	**A/D转换软件启动触发器** 此位置1可触发AD转换。此位在 A/D 转换期间保持逻辑 1，并在转换完成后由硬件自动清零。写入和读取ADCS位的含义不同。 写入: 0 = 无影响 1 = 启动AD转换 读取: 0 = ADC处于空闲状态 1 = ADC转换中，忙状态
5:4	**ETGSEL[1:0]**	**外部触发源选择** 当ADCEX(ADCCON1.1)置位，这些位选择哪个引脚输出触发ADC转换。 00 = PWM0. 01 = PWM2. 10 = PWM4. 11 = STADC引脚.

位	名称	描述
3:0	**ADCHS[3:0]**	**A/D转换通道选择** 此字段选择ADC的激活模拟输入源。如果 ADCEN 为 0，则所有输入均断开。 0000 = ADC_CH0 0001 = ADC_CH1. 0010 = ADC_CH2. 0011 = ADC_CH3. 0100 = ADC_CH4. 0101 = ADC_CH5. 0110 = ADC_CH6. 0111 = ADC_CH7. 1000 = VBG (内部带隙电压) 1001 = ADC_CH9. 1010 = ADC_CH10. 1011 = ADC_CH11. 1100 = ADC_CH12. 1101 = ADC_CH13. 1110 = ADC_CH14. 1111 = ADC_CH15.

PICON – 引脚中断控制

寄存器	特殊功能寄存器地址	复位值
PICON	E9H, 页0,	0000 _0000 b

7	6	5	4	3	2	1	0
PIT7	PIT6	PIT5	PIT4	PIT3	PIT2	PIT1	PIT0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	PIT7	**引脚中断通道7的类型选择** 此位选择引脚中断通道7的触发类型。 0 =电平触发。 1 =边沿触发。
6	PIT6	**引脚中断通道6的类型选择** 此位选择引脚中断通道6的触发类型。 0 =电平触发。 1 =边沿触发。
5	PIT5	**引脚中断通道5 的类型选择** 此位选择引脚中断通道5的触发类型。 0 =电平触发。 1 =边沿触发。
4	PIT4	**引脚中断通道4的类型选择** 此位选择引脚中断通道4的触发类型。 0 =电平触发。 1 =边沿触发。
3	PIT3	**引脚中断通道3 的类型选择** 此位选择引脚中断通道3的触发类型。 0 =电平触发。 1 =边沿触发。
2	PIT2	**引脚中断通道2的类型选择** 此位选择引脚中断通道2的触发类型。 0 =电平触发。 1 =边沿触发。
1	PIT1	**引脚中断通道1的类型选择** 此位选择引脚中断通道1的触发类型。 0 =电平触发。 1 =边沿触发。
0	PIT0	**引脚中断通道0的类型选择** 此位选择引脚中断通道0的触发类型。 0 =电平触发。 1 =边沿触发。

PINEN – 引脚中断负极性使能

寄存器	特殊功能寄存器地址	复位值
PINEN	EAH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PINEN7	**PINEN6**	**PINEN5**	**PINEN4**	**PINEN3**	**PINEN2**	**PINEN1**	**PINEN0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
n	**PINENn**	**通道n的引脚中断负极性使能** 此位使能低电平/下降沿触发通道n的引脚中断。电平触发还是边沿触发，取决于 PICON 中的每个控制位 PITn的设定。 0 = 低电平/下降沿侦测禁用。 1 = 低电平/下降沿侦测使能。

PIPEN – 引脚中断正极性使能

寄存器	特殊功能寄存器地址	复位值
PIPEN	EBH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PIPEN7	**PIPEN6**	**PIPEN5**	**PIPEN4**	**PIPEN3**	**PIPEN2**	**PIPEN1**	**PIPEN0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
n	**PIPENn**	**通道n的引脚中断正极性使能** 此位使能高电平/上升沿触发通道n的引脚中断。电平触发还是边沿触发，取决于 PICON 中的每个控制位 PITn的设定。 0 = 高电平/上升沿侦测禁用。 1 = 高电平/上升沿侦测使能。

PIF – 引脚中断标志

寄存器	特殊功能寄存器地址	复位值
PIF	ECH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PIF7	**PIF6**	**PIF5**	**PIF4**	**PIF3**	**PIF2**	**PIF1**	**PIF0**
只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)

位	名称	描述
n	**PIFn**	**引脚中断通道 n 标志** 如果选择边沿触发，则当引脚中断通道 n 检测到已启用的边沿触发时，硬件将置位此标志。此标志应通过软件清除。 如果选择电平触发，则此标志跟随引脚中断通道 n 上输入信号的逻辑电平反转。软件无法控制它。

EIP – 扩展中断优先级[3]

寄存器	特殊功能寄存器地址	复位值
EIP	EFH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PT2	**PSPI**	**PFB**	**PWDT**	**PPWM**	**PCAP**	**PPI**	**PI²C**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**PT2**	定时器2中断优先级低位
6	**PSPI**	SPI 中断优先级低位
5	**PFB**	故障刹车中断优先级低位
4	**PWDT**	看门狗定时器WDT中断优先级低位
3	**PPWM**	PWM中断优先级低位
2	**PCAP**	输入捕获中断优先级低位
1	**PPI**	引脚中断优先级低位
0	**PI²C**	I^2C中断优先级低位

[3]**注意：** EIP与 EIPH 结合使用，确定每个中断源的优先级。有关正确的中断优先级配置，请参阅表6.2-4 中断优先级层级设置。

B – B 寄存器

寄存器	特殊功能寄存器地址	复位值
B	F0H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
B.7	**B.6**	**B.5**	**B.4**	**B.3**	**B.2**	**B.1**	**B.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**B[7:0]**	**B 寄存器** B 寄存器是标准 80C51 的另一个累加器，主要用于 MUL 和 DIV 指令。

CAPCON3 – 输入捕获控制3

寄存器	特殊功能寄存器地址	复位值
CAPCON3	F1H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
CAP13	**CAP12**	**CAP11**	**CAP10**	**CAP03**	**CAP02**	**CAP01**	**CAP00**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
[7:4]	**CAP1[3:0]**	**输入捕获通道1 输入引脚选择** 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 其他 = P1.2/IC0
[3:0]	**CAP0[3:0]**	**输入捕获通道0 输入引脚选择** 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 其他 = P1.2/IC0

SCnCR0 – SC 控制寄存器 0

寄存器	特殊功能寄存器地址	复位值
SC0CR0	F1H, SFR页2	0000_0000 b
SC1CR0	F3H, SFR页2	0000_0000 b
SC2CR0	F5H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**NSB**	**停止位长度** 此字段指示停止位的长度。 0 = 停止位长度为 2 ETU。 1 = 停止位长度为 1 ETU。 **注意：**默认停止位长度为 2。SC 和 UART 采用 NSB 编程设置停止位长度。
6	**T**	**T模式** 0 = T0 (ISO 7816-3 T = 0 模式). 1 = T1 (ISO 7816-3 T = 1 模式). T 模式控制 BGT（区块保护时间）。区块保护时间是指不同传输方向之间两个连续字符前沿之间的最小位长度。此字段指示区块保护时间位长度的计数器。根据 ISO 7816-3，在 T = 0 模式下，软件必须将 T 位清零，以获得实际区块保护时间 = 16.5。在 T = 1 模式下，软件必须将 T 位设置为 1，以获得实际区块保护时间 = 22.5。 **注意：**在 T = 0 模式下，如果检测到奇偶校验错误，接收器将在 PEF(SC0TSR[4]) 上生成奇偶校验标志，并将奇偶校验错误信号驱动至收发器。在 T = 1 模式下，如果检测到奇偶校验错误，接收器将在 PEF(SC0TSR[4]) 上生成奇偶校验标志，但不会将奇偶校验错误信号驱动至收发器。 **注意：**具体说明请参见章节6.10.5.4 错误信号和字符重复
5	**RXBGTEN**	**接收端区块保护时间功能使能位** 0 = 接收器块保护时间功能禁用。 1 = 接收器块保护时间功能启用。
4	**CONSEL**	**约定选择** 0 = 正向约定。 1 = 反向约定。 **注1:** 如果 AUTOCEN(SCnCR0[3]) 写入“1”，此位自动清零。 **注2:** 如果 AUTOCEN(SCnCR0[3]) 使能，硬件将在 SCEN (SCnCR0[0]) =“1”后自动决定约定并更改 CONSEL (SCnCR0[4]) 位。

位	名称	描述
3	**AUTOCEN**	**自动约定使能位** 0 = 自动约定禁用。 1 = 自动约定使能。当硬件在复位应答状态接收到 TS 时，如果该 TS 为正约定，则 CONSEL (SCnCR0[4]) 将自动设置为 0；否则，如果该 TS 为逆约定，则 CONSEL (SCnCR0[4]) 将设置为 1。 **注意：**如果软件启用自动约定功能，则必须在复位应答状态之前完成设置，且第一个数据必须为 0x3B 或 0x3F。硬件接收到第一个数据并将其存储在缓冲区后，硬件将决定转换方式并自动更改 CONSEL（SCnCR0[4]）位。
2	**TXOFF**	**TX 传输禁用位** 0 = 收发器启用。 1 = 收发器禁用。
1	**RXOFF**	**RX传输禁用位** 0 = 接收器启用。 1 = 接收器禁用。 **注意：**如果 AUTOCEN（SCnCR0[3]）已启用，则必须忽略这些字段。
0	**SCEN**	**SC引擎启用位** 将此位设置为 1 以启用 SC 操作。如果清除此位，SC 将强制所有转换到 IDLE 状态。 **注意：**在填充其他寄存器之前，必须将 SCEN 设置为 1，否则 ISO 7816-3 将无法正常工作。

CAPCON4 – 输入捕获控制4

寄存器	特殊功能寄存器地址	复位值
CAPCON4	F2H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	CAP23	CAP22	CAP21	CAP20
-	-	-	-	读/写	读/写	读/写	读/写

位	名称	描述
[3:0]	**CAP2[3:0]**	**输入捕获通道2 输入引脚选择** 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 其他 = P1.2/IC0

SCnCR1 – SC 控制寄存器

寄存器	特殊功能寄存器地址	复位值
SC0CR1	F2H, SFR页2	0000_0000 b
SC1CR1	F4H, SFR页2	0000_0000 b
SC2CR1	F6H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
OPE	PBOFF	WLS		TXDMAEN	RXDMAEN	CLKKEEP	UARTEN
读/写	读/写	读/写		读/写	读/写	读/写	读/写

位	名称	描述
7	OPE	**奇校验使能位** 0 = 接收模式下，发送偶数个逻辑1，或检查数据字和奇偶校验位。 1 = 接收模式下，发送奇数个逻辑1，或检查数据字和奇偶校验位。 **注意：** 此位仅在PBOFF位为0时有效。
6	PBOFF	**奇偶校验位禁用控制** 0 = 在串行数据的“最后一个数据字位”和“停止位”之间生成或检查奇偶校验位。 1 = 传输过程中不生成奇偶校验位（发送数据时）或检查奇偶校验位（接收数据时）。 **注意：** 在ISO 7816-3模式下，此字段必须为0（默认设置带奇偶校验位）。
5:4	WLS	**字长选择** 00 = 字长为8位。 01 = 字长为7位。 10 = 字长为6位。 11 = 字长为5位。 **注意：** 在 ISO 7816-3 模式下，此 WLS 必须为 00。
3	TXDMAEN	**SC/UART TX DMA 使能** 此位使能 SC/UART TX 通过 PDMA 传输进行操作，在 SC/UART TX 启动之前，TX 数据需要在 XRAM 中准备好。 0 = SPI TX DMA 禁用 1 = SPI TX DMA 使能
2	RXDMAEN	**SC/UART RX DMA 使能** 此位使能 SC/UART RX 通过 PDMA 传输进行操作，在 SC/UART RX 操作后，RX 数据保存在 XRAM 中。 0 = SC/UART RX DMA 禁用 1 = SC/UART RX DMA 使能
1	CLKKEEP	**SC 时钟使能位** 0 = SC 时钟生成禁用。 1 = SC 时钟始终保持自由运行。

位	名称	描述
0	**UARTEN**	**UART 模式使能位** 0 = ISO 7816-3 模式. 1 = UART 模式 **注 1：**在 UART 模式下工作时，用户必须设置 CONSEL (SCnCR0[4]) = 0 且 AUTOCEN(SCnCR0[3]) = 0。 **注 2：**在 ISO 7816-3 模式下工作时，用户必须设置 UARTEN(SCnCR1 [0]) = 0。 **注 3：**当 UART 使能时，硬件将产生复位信号以复位 FIFO 和内部状态机。

SPCR – 串行外设控制寄存器

寄存器	特殊功能寄存器地址	复位值
SPCR	F3H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**SSOE**	**从机选择输出使能位** 此位与DISMODF(SPSR.3)位配合使用，用于确定$\overline{SS}$引脚的功能，如下所示 0 = $\overline{SS}$. 引脚用作通用I/O引脚。 1 = $\overline{SS}$. 引脚在选择外部从机设备时，每次传输后自动变为低电平，并在空闲状态下变为高电平以取消选择从机设备。
6	**SPIEN**	**SPI使能** 0 = SPI功能禁用. 1 = SPI功能使能.
5	**LSBFE**	**LSB优先使能** 0 = SPI 数据按 MSB 优先传输。 1 = SPI 数据按 LSB 优先传输。
4	**MSTR**	**主机模式使能** 此位用于切换SPI的主机与从机模式. 0 = SPI 配置为从机模式. 1 = SPI 配置为主机模式.
3	**CPOL**	**SPI 时钟极性选择** CPOL 位决定 SPI 时钟的空闲状态电平。详见图6.12-4 SPI . 0 = SPI 时钟在空闲模式时低电平. 1 = SPI 时钟在空闲模式时高电平.
2	**CPHA**	**SPI 时钟相位选择** CPHA 位决定 SPI 时钟的数据采样沿。详见图6.12-4 SPI . 0 = 数据在 SPI 时钟的第一个沿采样。 1 = 数据在 SPI 时钟的第二个沿采样。
1:0	**SPR[1:0]**	**SPI 时钟速率选择** 这两位选择 SPI 时钟分频器的四种等级。以下时钟速率是在 F_{SYS} = 16 MHz 条件下计算的。 Fsys = 16 MHz SPR1 / SPR0 / 分频 / SPI 时钟速率 0 / 0 / 2 / 8M bit/s 0 / 1 / 4 / 4M bit/s 1 / 0 / 8 / WM bit/s 1 / 1 / 16 / 1 M bit/s Fsys = 24 MHz SPR1 / SPR0 / 分频 / SPI 时钟速率 0 / 0 / 2 / 12M bit/s 0 / 1 / 4 / 6M bit/s 1 / 0 / 8 / 3M bit/s 1 / 1 / 16 / 1.5M bit/s SPR[1:0] 仅在主机模式(MSTR = 1)下有效。如果处于从机模式，时钟将自动与SPICLK引脚上来自主机的外部时钟同步， 通信速度最高可达F_{SYS}/2。

SPCR2 – 串行外设控制寄存器寄存器 2

寄存器	特殊功能寄存器地址	复位值
SPCR2	F3H, SFR页1	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	**SPIS1**	**SPIS0**
-	-	-	-	-	-	读/写	读/写

位	名称	描述
7:2	-	保留
0	**SPIS[1:0]**	**SPI 相邻字节的间隔时间选择** SPIS[1:0]和CPHA在8级SPI相邻字节间的间隔时间做选择，如下表： CPHA / SPIS1 / SPIS0 / SPI 时钟 0 / 0 / 0 / 0.0 0 / 0 / 1 / 0.5 0 / 1 / 0 / 1.5 0 / 1 / 1 / 2.0 1 / 0 / 0 / 0.0 1 / 0 / 1 / 1.0 1 / 1 / 0 / 2.0 1 / 1 / 1 / 2.5 SPIS[1:0]仅在主机模式(MSTR = 1)下有效。

SPSR – 串行外设状态寄存器

寄存器	特殊功能寄存器地址	复位值
SPSR	F4H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SPIF	**WCOL**	**SPIOVF**	**MODF**	**DISMODF**	**TXBUF**	**-**	**-**
读/写	读/写	读/写	读/写	读/写	只读	-	-

位	名称	描述
7	**SPIF**	**SPI完成标志** 在SPI 数据传输完成或接收到的数据移入到SPI 读缓冲区时，此位由硬件置位逻辑1。 如果ESPI (EIE .0)和 EA使能，触发SPI 中断请求。此位应由软件清除。如果SPIF置位1，禁止向SPDR写入。
6	**WCOL**	**写入冲突错误位** 此位标示写入冲突事件。一旦发生写入冲突事件，此位被置位，应由软件清零。
5	**SPIOVF**	**SPI 溢出错误标志** 此位指示发生溢出事件。一旦发生溢出事件，此位置位。如果 ESPI 和 EA 使能，则触发 SPI 中断。此位应通过软件清零。
4	**MODF**	**模式故障错误标志** 此位指示发生模式故障错误事件。如果$\overline{SS}$引脚配置为模式故障输入(MSTR =1且 DISMODF = 0)并且$\overline{SS}$被外部设备拉低，则会发生模式故障错误。MODF 将立即被置位逻辑 1。如果ESPI 和 EA使能，则触发 SPI 中断请求。应通过软件清除此位。
3	**DISMODF**	**禁用模式故障错误检测** 此位与 SSOE(SPCR.7)位组合使用，用以确定引脚的特性，如表6. 12-1 所示DISMODF 仅在主机模式下有效(MSTR = 1)。 0 = 模式故障检测使能。$\overline{SS}$作为模式故障检测的输入引脚，忽略 SSOE设置。 1 = 模式故障检测禁用。$\overline{SS}$ 功能遵循 SSOE 位配置。
2	**TXBUF**	**SPI 写入数据缓冲区状态** 此位指示 SPI 传输缓冲区状态。 0 = SPI 写入器数据缓冲区为空 1 = SPI 写入器数据缓冲区已满。

SPDR – 串行外设数据寄存器

寄存器	特殊功能寄存器地址	复位值
SPDR	F5H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SPDR[7:0]							
读/写							

位	名称	描述
7:0	SPDR[7:0]	**串行外设数据** 此字节用于在 SPI 总线上发送或接收数据。写入此字节即写入移位寄存器。读取此字节实际上是读取数据缓冲区。在主机模式下，写入此寄存器可同时启动字节的发送和接收。

AINDIDS0 – ADC通道数字输入断开

寄存器	特殊功能寄存器地址	复位值
AINDIDS0	F6H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
P11DIDS	**P03DIDS**	**P04DIDS**	**P05DIDS**	**P06DIDS**	**P07DIDS**	**P30DIDS**	**P17DIDS**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PxxDIDS**	**ADC通道数字输入禁用** 0 = ADC通道n数字输入使能。 1 = ADC通道n数字输入禁用。ADC通道n读取始终为0。

PIPSn – 引脚中断控制

寄存器	特殊功能寄存器地址	复位值
PIPS7	F7H, 页2	0000_0000 b
PIPS6	FFH, 页2	0000_0000 b
PIPS5	FEH, 页2	0000_0000 b
PIPS4	FDH, 页2	0000_0000 b
PIPS3	FCH, 页2	0000_0000 b
PIPS2	FBH, 页2	0000_0000 b
PIPS1	FAH, 页2	0000_0000 b
PIPS0	F9H, 页2	0000_0000 b

7	6	5	4	3	2	1	0
-	PSEL[2:0]			-	BSEL[2:0]		
-	读/写			-	读/写		

位	名称	描述
6:4	**PSEL[2:0]**	**引脚中断通道端口选择** 000 = P0 端口. 001 = P1端口. 010 = P2端口. 011 = P3端口. 100 = P4端口. 101 = P5端口. 110 = 保留. 111 = 保留.
2:0	**BSEL[2:0]**	**引脚中断通道位选择** 000 = Pn.0. 001 = Pn.1 010 =.Pn.2 011 = Pn.3. 100 = Pn.4. 101 = Pn.5. 110 = Pn.6. 111 = Pn.7. n为端口编号, 由PSEL[2:0]选择.

AINDIDS1 – ADC 通道数字输入断开1

寄存器	特殊功能寄存器地址	复位值
AINDIDS1	99H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
P25DIDS	**P14DIDS**	**P13DIDS**	**P24DIDS**	**P23DIDS**	**P22DIDS**	**P21DIDS**	**-**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	-

位	名称	描述
7:1	**PxxDIDS**	**ADC通道数字输入禁用** 0 = ADC通道n数字输入使能。 1 = ADC通道n数字输入禁用。ADC通道n读取始终为0。

SCON_1 – 串口1控制

寄存器	特殊功能寄存器地址	复位值
SCON_1	F8H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
SM0_1/FE_1	**SM1_1**	**SM2_1**	**REN_1**	**TB8_1**	**RB8_1**	**TI_1**	**RI_1**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	SM0_1/ FE_1	**串口1模式选择** SMOD0_1(T3CON.6)= 0: 详见表6.9-2 串口UART1 模式/波特率 SMOD0_1(T3CON.6)= 1: SM0_1/FE_1用于帧错误(FE)状态标志。由软件清除. 0 = 未发生帧错误(FE) 1 = 发生并检测到帧错误(FE)
6	SM1_1	
5	SM2_1	**多机通信模式使能** 此位的功能取决于串口1模式。 模式 0： 没有效果。 模式 1： 该位检查有效的停止位。 0 = 无论停止位的逻辑电平如何，接收始终有效。 1 = 仅当接收到的停止位为逻辑1，并且接收到的数据与“给定”或“广播”地址匹配时，接收才有效。 模式 2 或 3： 用于多机通信。 0 = 无论第9位的逻辑电平如何，接收始终有效。 1 = 仅当接收到的第9位为逻辑1，且接收到的数据与“给定”或“广播”地址匹配时，接收才有效。
4	REN_1	**接收使能** 0 = 串口1接收禁用。 1 = 串口1接收在模式 1、2或3 下使能。在模式0下，接收由条件REN_1 =1和RI_1 = 0启动。
3	TB8_1	**第9位发送位** 串口1在模式2和3中第 9 个发送位的状态。在模式 0 或 1 下未使用。
2	RB8_1	**第9位接收位** 此位标识串口1 在模式 2 或 3 下第 9 个接收位的逻辑电平。模式1下，RB8_1为接收停止位的逻辑电平。SM2_1位为逻辑1时有例外限制。模式0下 RB8_1未使用。

位	名称	描述
1	TI_1	**发送中断标志位** 当串口1在模式0下发送完数据帧第8位后，或在其他模式下发送完一笔数据帧最后一个数据位后，此标志位由硬件置位。当串口1中断使能时，置位此位会导致CPU执行串口1中断服务程序。此位必须通过软件手动清除。
0	RI_1	**接收中断标志位** 当串口1在模式0下接收完数据帧第8位后，或在模式 1，2 或 3 下采样停止位后，此标志位由硬件置位。SM2_1位为逻辑1时有例外限制。当串口1中断使能时，此位置位会导致CPU执行串口1中断服务程序。此位必须通过软件手动清除。

PWM0DTEN – PWM死区时间使能

寄存器	特殊功能寄存器地址	复位值
PWM0DTEN	F9H, SFR页0, TA保护	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	PWM0DTCNT.8	-	PDT45EN	PDT23EN	PDT01EN
-	-	-	读/写	-	读/写	读/写	读/写

位	名称	描述
4	PWM0DTCNT.8	**PWM死区计数器第8位** 参见PWM0DTCNT 寄存器。
2	PDT45EN	**PWM0_CH4 / PWM0_CH5 对死区时间插入使能** 此位仅当PWM0_CH4/PWM0_CH5为互补模式时有效。 0 = P0G4/P0G5 对信号无延时。 1 = P0G4/P0G5 对信号的上升沿插入死区时间延时。
1	PDT23EN	**PWM0_CH2 / PWM0_CH3 对死区时间插入使能** 此位仅当PWM0_CH2 / PWM_CH3为互补模式时有效。 0 = P0G2/P0G3对信号无延时。 1 = P0G2/P0G3对信号的上升沿插入死区时间延时。
0	PDT01EN	**PWM0_CH0 / PWM0_CH1对死区时间插入使能** 此位仅当PWM0_CH0 / PWM0_CH1为互补模式时有效。 0 = P0G0/P0G1对信号无延时。 1 = P0G0/P0G1对信号的上升沿插入死区时间延时。

PWM0DTCNT – PWM 死区时间计数器

寄存器	特殊功能寄存器地址	复位值
PWM0DTCNT	FAH, SFR页0, TA保护	0000_0000 b

7	6	5	4	3	2	1	0
PWM0DTCNT[7:0]							
读/写							

位	名称	描述
7:0	**PWM0DTCNT[7:0]**	**PWM死区计数器低字节** 此8位字段与PWM0DTEN.4组合构成9位PWM死区计数器PWM0DTCNT。仅当PWM处于互补模式且PWM对应的PWM0DTEN位被置位时，此计数器才有效。 PWM死区时间= $\frac{PDTCNT+1}{F_{SYS}}$. 请注意，用户不应在PWM运行期间修改PWM0DTCNT。

PWMnMEN – PWMnCx 掩码使能, n=0,1,2,3;x=0,1,2,3,4,5

寄存器	特殊功能寄存器地址	描述	复位值
PWM0MEN	FBH, SFR页0	PWM0_CHx 掩码使能	0000_0000 b
PWM1MEN	ADH, SFR页2	PWM1_CHx 掩码使能	0000_0000 b
PWM2MEN	BDH, SFR页2	PWM2_CHx 掩码使能	0000_0000 b
PWM3MEN	CDH, SFR页2	PWM3_CHx 掩码使能	0000_0000 b

7	6	5	4	3	2	1	0
-	-	**PMEN5**	**PMEN4**	**PMEN3**	**PMEN2**	**PMEN1**	**PMEN0**
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
x	**PMENx**	**PWMn_CHx 掩码使能** 0 = PWMn_CHx 信号从其 PWM 发生器输出。 1 = PWMn_CHx 信号被 PMDx 屏蔽。 **注意：** PMEN2~5 仅适用于 PWM0。

PWMnMD – PWMnCx掩码数据

寄存器	特殊功能寄存器地址	描述	复位值
PWM0MD	FCH, SFR页0	PWM0_CHx掩码数据	0000_0000 b
PWM1MD	ACH, SFR页2	PWM1_CHx 掩码数据	0000_0000 b
PWM2MD	BCH, SFR页2	PWM2_CHx 掩码数据	0000_0000 b
PWM3MD	CCH, SFR页2	PWM3_CHx 掩码数据	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
x	PMDx	**PWMn_CHx Mask Data** 一旦设置了相应的PMENx, PWMn_CHx信号就会输出屏蔽数据。 0 = PWMn_CHx信号被 0 屏蔽。 1 = PWMn_CHx信号被 1 屏蔽。 **注意：** PMD2~5仅适用于 PWM0。

PORDIS – 上电复位POR 禁用

寄存器	特殊功能寄存器地址	复位值
PORDIS	FDH, SFR页0, TA保护	0000_0000 b

7	6	5	4	3	2	1	0
PORDIS[7:0]							
只写							

位	名称	描述
7:0	**PORDIS[7:0]**	**上电复位POR禁用** 首先将5AH写入PORDIS，然后立即写入A5H ，即可禁用POR功能。

EIP1 – 扩展中断优先级 1

寄存器	特殊功能寄存器地址	复位值
EIP1	FEH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PPWM3	PPWM2	PPWM1	PWKT	PT3	PS_1
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
5	**PPWM3**	PWM3 中断优先级低位
4	**PPWM2**	PWM2 中断优先级低位
3	**PPWM1**	PWM1 中断优先级低位
2	**PWKT**	WKT 中断优先级低位
1	**PT3**	定时器3 中断优先级低位
0	**PS_1**	串口1 中断优先级低位

注意： EIP1 与EIPH1 配合使用，用于确定每个中断源的优先级。有关正确的中断优先级配置，请参见 表6.2-4 中断优先级层级设置

EIPH1 – 扩展中断优先级高字节1[6]

寄存器	特殊功能寄存器地址	复位值
EIPH1	FFH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PPWM3H	PPWM2H	PPWM1H	PWKTH	PT3H	PSH_1
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
5	**PPWM3H**	PWM3 中断优先级高位
4	**PPWM2H**	PWM2 中断优先级高位
3	**PPWM1H**	PWM1 中断优先级高位
2	**PWKTH**	WKT 中断优先级高位
1	**PT3H**	定时器3 中断优先级高位
0	**PSH_1**	串口1 中断优先级高位

[6]**注意：** EIPH1与EIP1 配合使用，用于确定每个中断源的优先级。有关正确的中断优先级配置，请参见 表6.2-4 中断优先级层级设置

LVRDIS – LVR 禁用

寄存器	特殊功能寄存器地址	复位值
LVRDIS	FFH, SFR页1, TA保护	0000_0000 b

7	6	5	4	3	2	1	0
LVRDIS[7:0]							
只写							

位	名称	描述
7:0	LVRDIS[7:0]	**LVR 禁用** 首先向 LVRDIS 写入 5AH，然后立即写入 A5H，即可禁用 LVR。

6.2 系统管理

6.2.1 时钟系统

6.2.1.1 概述

CM1003具有多种时钟源可供选择，可用于各种应用，使系统性能发挥到最佳，把功耗降到最低。CM1003提供三种系统时钟源选项，包括内部振荡器或通过软件从X_{IN}引脚获取的外部时钟。CM1003内嵌了两个内部振荡器：一个10kHz低速振荡器和一个16MHz或24MHz高速振荡器，误差在出厂时校准到±2%(全温度全电压范围内)。CM1003上还提供时钟分频器CKDIV，可在功耗和工作性能间灵活调整。

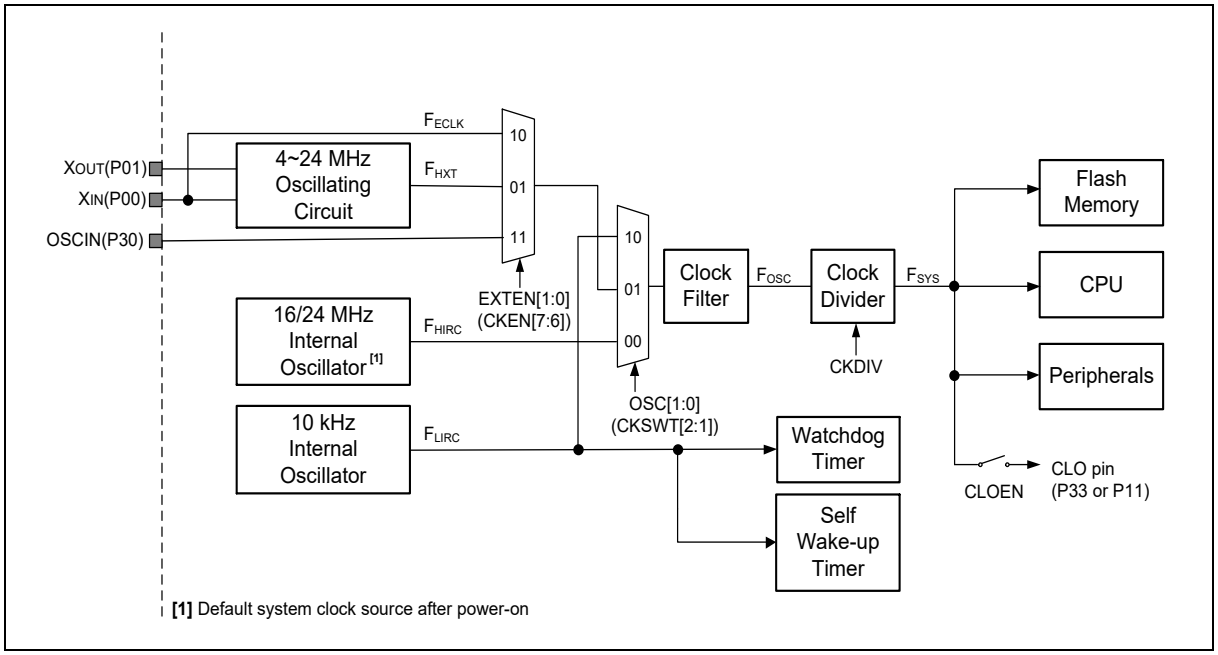


图 6.2-1 时钟系统框图

6.2.1.2 系统时钟源

CM1003共有三个可选的系统时钟源，包括高速内部振荡器、低速内部振荡器和外部时钟输入。每个时钟源都可以作为 CM1003的系统时钟源。开启不同的时钟源可能会影响到引脚P0.0/X_{IN}的多功能。

6.2.1.3 内部振荡器

CM1003有两个内部振荡器——一个高速内部振荡器(HIRC)，可根据HIRC24(RCTRIM1.4)配置成16MHz或24MHz，以及一个10kHz低速振荡器(LIRC)。两者都可以被选做系统时钟。通过设置HIRCEN(CKEN.5)来启动HIRC。设备上电后启用LIRC。用户可设置OSC[1:0](CKSWT [2:1])为 [1,1] 来选择 HIRC 作为系统时钟，设置OSC[1:0] 为[1,0]来选择LIRC作为系统时钟。请注意，CM1003上电后，HIRC和LIRC都将使能，并且默认选择HIRC作为系统时钟源。当系统使用内部振荡器作为时钟源时，X_{IN}会自动切为通过I/O P0.0，用来扩展通用I/O的数量。可以通过配置P0M1和P0M2寄存器来选择P0.0的I/O输出模式。

由于HIRC的值有9位数据，用户可以调整SFR来修改HIRC值，该值的每个1大约调整40kHz。减小此值将使HIRC更快，增加1意味着HIRC速度降低约40kHz。

下图显示了两个字节组合的9位内部RC调整值，以及HIRC 24MHz定义位。

RCTRIM0 – 高速内部振荡器 0

寄存器	特殊功能寄存器地址	复位值
RCTRIM0	84H, SFR页0, TA保护	0000_0000b

7	6	5	4	3	2	1	0
HIRCTRIM[8:1]							
读/写							

RCTRIM1 – 高速内部振荡器 1

寄存器	特殊功能寄存器地址	复位值
RCTRIM1	85H, 所有SFR页, TA保护	0000_0000b

7	6	5	4	3	2	1	0
-	-	-	HIRC24	-	-	-	HIRCTRIM.0
-	-	-	读/写	-	-	-	读/写

注意： 由于默认的RCTRIM0和RCTRIM1值是基于16MHz的，如果以此值为基础修改HIRC24(RCTIM1.4)以启用24MHz HIRC模式，则实际HIRC偏差将超过1%。建议在UID之后重新加载24MHz HIRC值，并检查值的第4位以确认此值适合24MHz应用。

6.2.1.4 外部晶振/谐振器或时钟输入

外部时钟源有三种可选的时钟源：4 MHz 至 24 MHz 高速晶振/谐振器 (HXT) 以及通过 X_{IN} 引脚的外部时钟输入 (ECLK)。用户可将 OSC[1:0] 设置为 [1,0]，然后设置 EXTEN[1:0]（CKEN[7:6]）以选择外部时钟源。要选择 HXT 作为输入源，请将 EXTEN[1:0] 定义为 [0,1]。要选择 ECLK 作为系统时钟，当 EXTEN[1:0] 设置为 [1,1] 时，选择由P3.0 输入的外部时钟。或者，当 EXTEN[1:0] 设置为 [1,0] 时，选择由P0.0 输入的外部时钟。

当使用 HXT 作为系统时钟时，X_{IN} 和 X_{OUT} 分别是内部反相放大器的输入和输出。应在 X_{IN} 和 X_{OUT} 引脚之间连接一个晶振或谐振器。

当使能并选择 ECLK 作为系统时钟源时，系统时钟由 X_{IN} 引脚提供。常见的应用是使用有源振荡器或来自其他主机设备的时钟驱动 X_{IN}。请注意，当选择 ECLK 模式时，切勿向 X_{IN}馈送任何高于 2.4V 电压的时钟信号，以及向 X_{OUT} 引脚施加电压为 X_{IN} 一半的直流电压，否则可能会损坏设备。

XTLCON – XLT 时钟控制 (TA 保护)

寄存器	特殊功能寄存器地址	复位值
XTLCON	D7H, SFR页2	0010_1000 b

7	6	5	4	3	2	1	0
-	HXSG						
-	读/写						

位	名称	描述
7	-	保留
6:0	HXSG	**HXT增益值选择** 增益控制用于增大晶体的增益，以确保晶体正常工作。如果启用增益控制，晶体的功耗将比关闭增益控制时更高。 建议： 000_0000 = L0。HXT频率为4 MHz (最小值). 000_1100 = L1。频率范围为4 MHz ~ 12 MHz. 001_1000 = L2。频率范围为12MHz ~ 16 MHz. 010_1000 = L3。频率范围为16MHz ~ 24 MHz. 111_1111 = 保留 (最大值). 其他 = 保留.

6.2.1.5 系统时钟切换

CM1003系列支持通过软件设置CKSWT和CKEN寄存器来实时切换时钟源。这给应用带来了极大的灵活性。请注意，这些特殊功能寄存器SFR是定时访问写保护的（TA）。通过这种时钟源控制，时钟源可以在外部时钟源和内部振荡器之间切换，甚至可以在高低速内部振荡器之间切换。但是，在时钟源切换期间，设备需要一定的预热时间来让切换后的时钟源稳定工作。因此，用户应遵循以下步骤以确保完成时钟源切换。用户可以先通过向CKEN寄存器写入适当的值来打开目标时钟源，再通过向CKSWT寄存器轮询时钟源的状态位来等待时钟源稳定，然后通过更改OSC[1:0](CKSWT[2:1])切换到目标时钟源。完成这些步骤后，时钟源切换成功。如果考虑到功耗，用户可以关闭原时钟源。请注意，如果不遵循上述步骤，硬件将采取以下措施来处理此类非法操作。

1. 如果用户试图通过更改CKEN值来关闭当前时钟源，设备将忽略此操作。系统时钟和 CKEN 的值将保持不变。
2. 如果用户试图通过更改OSC[1:0]值将系统时钟源切换为未被启用的时钟源，OSC[1:0]值将被立即更新。但系统时钟仍保持原样，而CKSWTF标志位会被硬件置位。
3. 一旦用户将系统时钟源切换到一个已启用但仍不稳定的时钟源，那么硬件将等待目标时钟源稳定后再切换到该时钟源。在等待期间，设备将继续使用原时钟源工作，CKSWTF被置位1。在目标时钟源的稳定标志 (参见CKSWT[7:3]) 置位且时钟源切换成功后，硬件将自动把CKSWTF清零。

CKSWT – 时钟开关

寄存器	特殊功能寄存器地址	复位值
CKSWT	96H, 页0, TA保护	0011_0000 b

7	6	5	4	3	2	1	0
HXTST	ECKP00ST	HIRCST	LIRCST	ECKP30ST	OSC[1:0]		-
只读	只读	只读	只读	只读	只写		-

位	名称	描述
7	HXTST	**高速外部晶振 4 MHz 至 24 MHz 状态** 0 = 高速外部晶振不稳定或禁用。 1 = 高速外部晶振使能且稳定。
6	ECKP00ST	**外部时钟 P00 输入状态** 0 = 外部时钟 P00 输入不稳定或禁用。 1 = 外部时钟 P00 输入使能且稳定。
5	HIRCST	**高速内部振荡器 16 MHz 状态** 0 = 高速内部振荡器不稳定或禁用 1 = 高速内部振荡器使能且稳定
-	-	保留
3	ECKP30ST	**外部时钟 P30 输入状态** 0 = 外部时钟 P30 输入不稳定或禁用。 1 = 外部时钟 P30 输入使能且稳定。
2:1	OSC[1:0]	**振荡器选择位** 此位是用来选择系统时钟源 00 = 内部16MHz振荡器 01 = 根据 EXTEN[1:0](CKEN[7:6])设置选择外部时钟源。 10 = 内部10kHz振荡器 11 = 保留 注意:此位段为只写位段，读回的值可能与当前系统时钟源不一致

CKEN – 时钟使能

寄存器	特殊功能寄存器地址	复位值
CKEN	97H, 页0, TA保护	0011_0000 b

7	6	5	4	3	2	1	0
EXTEN[1:0]		HIRCEN	LIRCEN	-	-	-	CKSWTF
读/写		读/写	读/写	-	-	-	只读

位	名称	描述
7:6	**EXTEN[1:0]**	**外部时钟源使能** 11 = 通过 OSCIN P3.0 输入外部时钟使能 10 = 通过 OSCIN P0.0 输入外部时钟使能 01 = 外部晶振 4~24 MHz 使能 00 = 外部时钟输入禁用。P30/P00/P01 可用作通用 I/O 或其他功能。
5	**HIRCEN**	**16 MHz 内部高速振荡器使能** 0 = 禁用内部高速振荡器。 1 = 使能内部高速振荡器。 需要注意的是，一旦通过设置 IAPEN（CHPCON.0）使能 IAP，16 MHz 内部高速振荡器将自动使能。硬件也会设置 HIRCEN 和 HIRCST 位。IAPEN 清零后，HIRCEN 和 EHRCST 恢复原始值。
4	**LIRCEN**	**10 kHz 内部低速振荡器使能** 0 = 禁用内部低速振荡器。 1 = 使能内部低速振荡器。 请注意，当 (1) WDT 使能，(2) WKT 由内部 10 kHz 振荡器的时钟源运行时，向 LIRCEN 写入 0 将被忽略。LIRCEN 始终为 1，内部 10 kHz 振荡器始终处于使能状态。
3:1	**-**	保留
0	**CKSWTF**	**时钟切换故障标志** 0 =上一次系统时钟源切换成功 1 =用户在上一次的系统时钟源切换时尝试切换到禁用或是不稳定的时钟源。如果切换的时钟不稳定，此位将一直保持为1，直到时钟源稳定并切换成功为止。

6.2.1.6 *系统时钟除频*

通过配置分频寄存器CKDIV，振荡器频率(F_{OSC})可以整数倍分频(最高可达 1/510)，再提供给系统时钟(F_{SYS})。此功能可使MCU临时以较低的速率运行来降低功耗。通过分频时钟，MCU可以以较低的速率执行正常程序，也能响应事件，除了那些可能导致中断的事件(比如允许退出空闲模式的事件)。这通常比空闲模式还要省电。可以在原本要使用掉电模式的情况下绕过振荡器启动时间。程序可以随时更改CKDIV 的值，而不会中断代码执行。

CKDIV – 时钟分频器

寄存器	特殊功能寄存器地址	复位值
CKDIV	95H, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
CKDIV[7:0]							
读/写							

位	名称	描述
7:0	**CKDIV[7:0]**	**时钟分频器** 系统时钟频率 F_{SYS}根据CKDIV的值遵循以下计算公式。 当CKDIV = 00H, $F_{SYS} = F_{OSC}$, 当CKDIV = 01H 至 FFH, $F_{SYS} = \frac{F_{OSC}}{2 \quad CKDIV}$ 。

6.2.1.7 系统时钟输出

CM1003提供一个CLO引脚用于输出系统时钟，其频率与F_{SYS}相同。CLOEN(CKCON.1)为输出使能位。当设备因系统时钟关闭而进入掉电模式时，CLO输出停止。请注意，当噪声或功耗成为重要问题时，用户最好不要启用CLO输出。

CKCON – 时钟控制

寄存器	特殊功能寄存器地址	复位值
CKCON	8EH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	PWMCKS	-	T1M	T0M	-	CLOEN	-
-	读/写	-	读/写	读/写	-	读/写	-

位	名称	描述
1	CLOEN	**系统时钟输出使能** 0 = 系统时钟输出禁止 1 = 系统时钟从CLO脚位(P1.1)输出使能 **注意：** CLO 引脚由AUXR5.7决定。

6.2.2 电源管理

6.2.2.1 概述

CM1003系列有多种功能，可帮助用户控制设备的功耗。低功耗功能有两种可选模式：空闲模式和掉电模式，以节省功耗。为了保持稳定的电流消耗，应注意每个引脚的状态和模式。将引脚状态设为与外部上拉、下拉一致来可以实现最小功耗。比如，使用上拉电阻则输出1，使用下拉电阻则输出0。如果I/O引脚浮空，建议用户将其设置为准双向模式。如果P2.0配置为仅输入引脚，它应该使用外部上拉或下拉，或者通过设置P30UP(P3UP.0)启用其内部上拉电阻。

PCON – 电源控制

寄存器	特殊功能寄存器地址	复位值
PCON	87H, 所有SFR页	上电复位: 0001_000b, 其它: 000U_0000b

7	6	5	4	3	2	1	0
SMOD	SMOD0	保留	POF	GF1	GF0	PD	IDL
读/写	读/写	-	读/写	读/写	读/写	读/写	读/写

位	名称	描述
1	PD	**掉电模式** 置位此位将使CPU进入掉电模式。在此模式下，CPU和外设时钟均停止，程序计数器(PC)挂起。此时为最低功耗。CPU从掉电模式唤醒后，此位由硬件自动清除，程序继续执行唤醒系统中断源的中断服务程序(ISR)。从ISR返回后，设备继续执行使系统进入掉电模式的下一条指令。 注意：如果 IDL 位和 PD 位同时置位，CPU将进入掉电模式。退出掉电模式后，CPU不会进入空闲模式。
0	IDL	**空闲模式** 置位此位将使CPU进入空闲模式。在此模式下，CPU时钟停止，程序计数器(PC)挂起，但所有外设保持激活状态。CPU从空闲模式唤醒后，此位由硬件自动清除，程序继续执行唤醒系统中断源的ISR。从ISR返回后，设备继续执行使系统进入空闲模式的下一条指令。

6.2.2.2 空闲模式

空闲模式通过保持程序计数器的值来使CPU挂起。空闲模式下不会读取和运行任何程序代码。它强制冻结CPU状态。程序计数器(PC)、堆栈指针(SP)、程序状态字(PSW)、累加器(ACC)和其他寄存器在空闲模式下保持其内容不变。端口引脚保持空闲模式激活时的逻辑状态。通常空闲模式可以节省相当多的功耗，一般功耗约为全工作状态下的一半。

由于空闲模式下提供给外设比如定时器或串口的逻辑电路依然工作，因此CPU可以通过任意使能的中断源从空闲模式中释放出来。用户可以通过向 IDL位(PCON.0)写入1来使设备进入空闲模式。设置IDL 位的指令是设备进入空闲模式之前将执行的最后一条指令。

空闲模式可以通过两种方式终止。首先，如前所述，任意使能的中断都会导致系统退出空闲模式。它将自动清除IDL位，终止空闲模式，并执行中断服务例程(ISR)。使用RETI指令跳出ISR后，程序将返回使CPU进入空闲模式之后的下一条指令继续执行。终止空闲模式的第二种方法是使用除软件复位以外的任何复位。请记住，如果使用看门狗复位退出空闲模式，则需要将WIDPD(WDCON.4)设置为1，以使WDT在空闲模式下继续运行。

6.2.2.3 掉电模式

掉电模式是CM1003系列可以进入的最低功耗状态。通过停止系统时钟源，功耗保持在“微安”级别。CPU和外设功能，比如定时器或UART，均被冻结。Flash存储进入停止模式，所有动作完全停止，功耗降至最低。通过向PD(PCON.1)写1可让设备进入掉电模式。这条指令将是设备进入掉电模式之前的最后一条指令。在掉电模式下，RAM 保持其内容。端口引脚分别输出掉电前其自身状态所保持的值。

有几种方法可以让CM1003从掉电模式中退出。第一种是使用除软件复位之外的所有复位。掉电复位也会将CPU从掉电模式中唤醒。确保在系统进入掉电模式之前开启掉电检测。即使为了降低功耗，还是建议在掉电模式下启用低功耗BOD欠压检测功能。当然，外部引脚复位或上电复位也将退出掉电状态。外部引脚复位或上电复位后，CPU被初始化并从头开始执行程序代码。

将CM1003从掉电模式唤醒的第二种方法是通过启用外部中断。外部引脚上的触发器将异步重新启动系统时钟。振荡器稳定后，设备将执行相应外部中断的中断服务程序(ISR)。ISR完成后，程序执行返回到使设备进入掉电模式的下一条指令，继续执行。允许将CPU从掉电模式唤醒的中断包括外部中断$\overline{INT0}$和$\overline{INT1}$、引脚中断、WDT中断、WKT中断和掉电中断。

6.2.3 电源检测和复位

6.2.3.1 概述

为了防止上电和断电期间的错误执行， CM1003提供了三种电源监控功能，包括上电复位，低电复位和欠压检测。

CM1003系列提供多种复位方式，通过软件标志还可以确定复位源。通常，大部分特殊功能寄存器的复位后的值与复位条件无关，但有一些标志位的状态取决于复位源。用户可以使用软件读取这些标志位以确定复位原因。有五种方式会让设备复位，分别是上电复位、欠压复位、外部复位、看门狗定时器复位和软件复位。

6.2.3.2 上电复位和低电复位

CM1003系列集成了内部上电复位(POR)和低电压复位(LVR)。在V_{DD}上升的上电过程中，当V_{DD}低于参考电压门限值，POR或LVR将使MCU处于复位模式。这种设计使CPU在V_{DD} 不满足执行读取存储器时，无法访问程序存储空间。如果从程序存储中读取并执行不确定的操作代码，可能会使CPU甚至是整个系统进入错误状态。一段时间后，V_{DD}上升到系统可以工作的参考门限电压以上，选定的振荡器将启动，然后程序代码将从0000H开始执行。同时，上电标志POF(PCON.4)将置为1表示冷复位，上电过程完成。请注意，上电后内部RAM的内容将不确定。建议用户初始化RAM。

建议通过软件将POF清零，以检测下次复位是冷复位还是热复位。如果是电源断电再打开引起的冷复位，POF将再次设置为1。如果是其他复位源引起的热复位，POF将保持为0。用户可以采取不同的方法检测其他复位标志位来处理热复位事件。

PCON – 电源控制

寄存器	特殊功能寄存器地址	复位值
PCON	87H, 所有SFR页	上电复位: 0001_000b, 其它: 000U_0000b

7	6	5	4	3	2	1	0
SMOD	SMOD0	LPR	POF	GF1	GF0	PD	IDL
读/写	读/写	RW	读/写	读/写	读/写	读/写	读/写

位	名称	描述
4	POF	**上电复位标志** 上电复位后，该位将被设置为1。表示冷复位，即上电复位完成。在任何其他复位后，该位将保持其值。建议通过软件清除该标志。

PORDIS – 上电复位POR禁用

寄存器	特殊功能寄存器地址	复位值
PORDIS	FDH, SFR页0, TA保护	0000_0000 b

7	6	5	4	3	2	1	0
PORDIS[7:0]							
只写							

位	名称	描述
7:0	**PORDIS[7:0]**	**上电复位POR禁用** 首先将5AH写入PORDIS，然后立即写入A5H ，即可禁用POR功能。

LVRDIS – LVR 禁用

寄存器	特殊功能寄存器地址	复位值
EIPH1	FFH, SFR页1, TA保护	0000_0000 b

7	6	5	4	3	2	1	0
LVRDIS[7:0]							
只写							

位	名称	描述
7:0	**LVRDIS[7:0]**	**LVR 禁用** 首先向 LVRDIS 写入 5AH，然后立即写入 A5H，即可禁用 LVR。

6.2.3.3 欠压检测和复位

另一项电源监控功能是欠压检测电路(BOD)，用于执行期间监测V_{DD}的电平。针对宽电压应用，CONFIG提供了8个可选的欠压触发级别。通过设置CBOV[2:0](CONFIG2[6:4])来选择4个标称电压2.2V、2.7V、3.7V和4.4V。上电后也可以在通过设置BOV[2:0] (BODCON0[6:4])来改变BOD电平。当V_{DD}降至选定的欠压触发电平(V_{BOD})时，BOD逻辑将复位MCU或请求欠压中断。用户可以根据不同的应用来选择复位或生成欠压中断。V_{BOD}也可以在通电后由软件设置。请注意，BOD 输出在软件使能后需要2~3个LIRC时钟后才能输出。

当BORST(BODCON0.2)为0时，如果V_{DD}降至V_{BOD}以下，BOD将请求中断。在这种情况下，BOF(BODCON0.3)将设置为1。用户清除此标志，如果V_{DD}仍低于V_{BOD}，BOF将不会再次设置。BOF仅用于通知用户电压下降发生。在V_{DD}高于V_{BOD}后，BOF也将置位，以指示电源恢复。BOD电路提供了一个有用的状态指示器BOS(BODCON0.0)，有助于告知发生欠压事件或电源恢复事件。如果BORST位被置位，这将启用欠压复位功能。欠压复位后，BORF(BODCON0.1) 将由硬件置位1。除上电复位外，任何其他复位都不会改变BORF。此位可以由软件清除。请注意，BODCON0中的所有位都受到定时访问(TA)的写保护。

CM1003系列提供低功耗BOD模式，可在降低电流消耗的同时保留响应有限的BOD功能。通过设置LPBOD[1:0] (BODCON1[2:1])，可以周期性地启用BOD电路，以每1.6 ms、6.4 ms或25.6 ms检测一次电源电压。这节省了大量功耗，还能提供低速电源电压检测。请注意，在低功耗BOD模式下，迟滞功能将消失。

对于对噪声敏感的系统，CM1003系列内置BOD滤波器，可滤除电源噪声，以避免BOD事件的意外触发。BOD滤波器默认启用，如果用户需要快速BOD响应，可以通过将BODFLT (BODCON1.0)设置为0来禁用BOD过滤器。LPBOD[1:0] (BODCON1)中列出了最小欠压检测脉冲宽度。

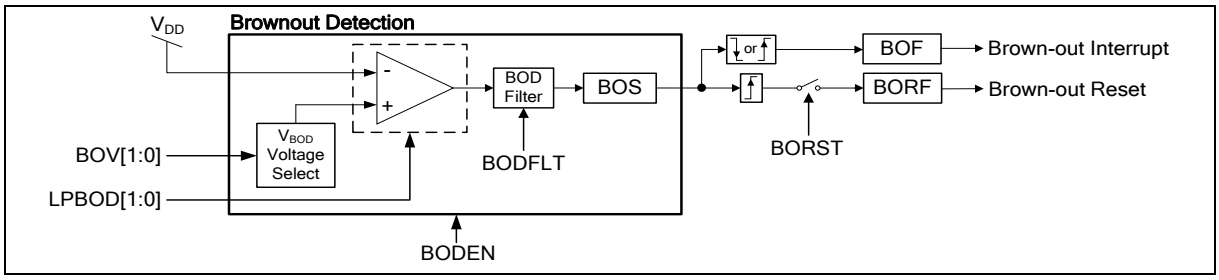


图 6.2-2 欠压检测框图

CONFIG2

7	6	5	4	3	2	1	0
CBODEN	-	CBOV[1:0]		BOIAP	CBORST	-	-
读/写	-	读/写		读/写	读/写	-	-

出厂默认值：1111 1111b

位	名称	描述
7	CBODEN	**配置欠压检测使能** 1 = 欠压检测电路打开 0 = 欠压检测电路关闭
5:4	CBOV[1:0]	**配置欠压检测电压选择** 11 = V_{BOD} 生效电压 2.2V. 10 = V_{BOD} 生效电压 2.7V. 01 = V_{BOD} 生效电压 3.7V. 00 = V_{BOD} 生效电压 4.4V.
2	CBORST	**配置欠压复位使能** 此位决定检测到电压低于V_{BOD}时，是否欠压复位 1 = 欠压复位功能使能。 0 = 欠压复位功能关闭。

CBODEN (CONFIG2.7)	CBORST (CONFIG2.2)	V_{DD} 电平	BOF
1	1	总是> V_{BOD}	0
1	0	< V_{BOD}	1
1	0	> V_{BOD}	0
0	X	X	0

表6.2-1 BOF复位值

BODCON0 – 欠压检测控制0

寄存器	特殊功能寄存器地址	复位值
BODCON0	A3H, SFR页0, TA保护	上电复位POR：CCCC XC0X b 欠压复位BOD： UUUU XU1X b 其他：UUUU XUUX b

7	6	5	4	3	2	1	0
BODEN[1]	**BOV[2:0]** [1]			**BOF**[2]	**BORST**[1]	**BORF**	**BOS**
读/写	读/写			读/写	读/写	读/写	只读

位	名称	描述
7	**BODEN**	**欠压检测使能** 0 = 欠压检测电路关闭 1 = 欠压检测电路打开 注意在开启该功能后需要2到3个LIRC时钟BOD输出才能正常工作 。
6:4	**BOV[1:0]**	**欠压检测电压选择** 11 = V_{BOD} 生效电压 2.2V. 10 = V_{BOD} 生效电压 2.7V. 01 = V_{BOD} 生效电压 3.7V. 00 = V_{BOD} 生效电压 4.4V.
3	**BOF**	**欠压中断标志** 当V_{DD}下降到V_{BOD}以下或V_{DD}上升到V_{BOD}以上时，此该标志由硬件置位逻辑1。如果EBOD(EIE.2)和EA(IE.7)都置位，将生成欠压中断请求。此位应由软件清零。
2	**BORST**	**欠压检测复位使能** 此位决定在电源电压跌到V_{BOD}以下时是否产生欠压复位 0 = 当V_{DD}掉电到V_{BOD}以下，欠压复位禁用 1 =当V_{DD}掉电到V_{BOD}以下，欠压复位使能
1	**BORF**	**欠压复位标志** 当MCU发生欠压复位时，此位被硬件置位。建议通过软件清除此位。
0	**BOS**	**欠压状态标志** 在BOD电路开启时，此位指示V_{DD}与V_{BOD}的比较情况。BOD电路关闭时，此位保持为0. 0 = V_{DD} 电压高于 V_{BOD} 或BOD 电路关闭 1 = V_{DD} 电压低于V_{BOD} 注：此位为只读位

注意：

1.所有复位后，BODEN、BOV[1:0] 和 BORST 均通过直接从 CONFIG2的位7、[6:4] 和位2加载进行初始化。

2.BOF复位值取决于CONFIG2和V_{DD}电压电平的不同设置。请参阅表6. 2-1 BOF复位值

BODCON1 – 欠压检测控制 1

寄存器	特殊功能寄存器地址	复位值
BODCON1	ABH, SFR页0, TA保护	上电复位POR： 0000 0001 b 其他： 0000 0UUU b

7	6	5	4	3	2	1	0
-	-	-	-	-	**LPBOD[1:0]**		**BODFLT**
-	-	-	-	-	读/写		读/写

位	名称	描述
7:3	-	保留
2:1	**LPBOD[1:0]**	**低功耗BOD使能** 00 = BOD正常模式，BOD电路始终启用。. 01 = BOD低功耗模式1， 每隔1.6ms周期性开启BOD电路 10 = BOD低功耗模式 2， 每隔6.4ms周期性开启BOD电路 11 = BOD低功耗模式 3， 每隔25.6ms周期开启BOD电路
0	**BODFLT**	**BOD滤波器控制** BOD内置一个滤波器，当MCU在HIRC或ECLK作为系统时钟源且BOD没有在低功率模式(LPBOD[1:0] = [0, 0]) 下工作时，该滤波器计数32个F_{SYS}时钟周期，用以过滤电源噪声。在其他情况下，滤波器计数2个LIRC时钟周期。 注意:当CPU停在掉电模式时，BOD将始终由2个LIRC时钟周期滤波输出。 BOD滤波器有效地避免电源噪声误触发BOD事件。设置此位控制BOD滤波功能开启或关闭。 0 = BOD滤波器禁用. 1 = BOD滤波器使能.(上电复位默认值)

BODFLT (BODCON1.1)	BOD 操作模式	系统时钟源	最小欠压检测脉冲宽度
0	正常模式 (LPBOD[1:0] = [0,0])	任意时钟源	Typ. 1μs
	低功耗模式1 (LPBOD[1:0] = [0,1])	任意时钟源	16 (1/FLIRC)
	低功耗模式2 (LPBOD[1:0] = [1,0])	任意时钟源	64 (1/FLIRC)
	低功耗模式3 (LPBOD[1:0] = [1,1])	任意时钟源	256 (1/ FLIRC)
1	正常模式 (LPBOD[1:0] = [0,0])	HIRC/ECLK	正常操作: 32 (1/FSYS) 空闲模式: 32 (1/FSYS) 掉电模式: 2 (1/FLIRC)
		LIRC	2 (1/FLIRC)
	低功耗模式1 (LPBOD[1:0] = [0,1])	任意时钟源	18 (1/FLIRC)

BODFLT (BODCON1.1)	BOD 操作模式	系统时钟源	最小欠压检测脉冲宽度
	低功耗模式2 (LPBOD[1:0] = [1,0])	任意时钟源	66 (1/FLIRC)
	低功耗模式3 (LPBOD[1:0] = [1,1])	任意时钟源	258 (1/ FLIRC)

表 6.2-2 最小欠压检测脉冲宽度

6.2.3.4 *外部复位和硬故障复位*

外部复位引脚nRESET是一个带有施密特触发器的输入引脚。外部复位的实现方法是将nRESET引脚保持低电平至少24个系统时钟周期，以确保检测到有效的硬件复位信号。然后复位电路同步施加内部复位信号。因此，复位是一个同步操作，需要时钟运行才能触发外部复位。

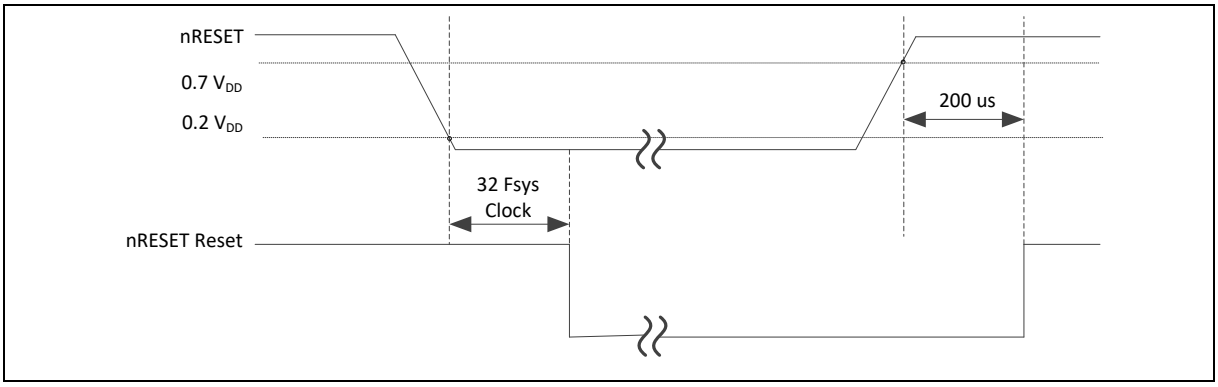


图 6.2-3 nRESET 复位波形

一旦设备处于复位状态，只要nRESET引脚为低电平，复位状态就会一直保持。nRESET引脚高电平后，MCU将退出复位状态，并从地址0000H开始执行代码。如果CPU处于掉电模式时发生外部复位，则触发硬件复位的方式略有不同。由于掉电模式会停止系统时钟，复位信号将异步地恢复系统时钟。系统时钟稳定后，MCU将进入复位状态。

RSTPINF(AUXR0.6)标志位用来指示发生了外部复位。外部复位后，此位将由硬件置1。除上电复位或外部复位本身外，任何复位后RSTPINF都不会改变。此位可通过软件清零。

如果CPU读取的指令地址超过 Flash 大小，将发生硬故障复位， HardF(AUXR0.5)标志位将由硬件置位。除上电复位或外部复位本身外，任何其他复位后HardF都不会改变。此位可通过软件清零。如果MCU运行在OCD调试模式下且OCDEN = 0，则硬故障复位将被禁用。只有HardF标志有效。

AUXR1 –辅助寄存器 1

寄存器	特殊功能寄存器地址	复位值
AUXR1	A2H , SFR页0	上电复位: 0000 0000b, 软件复位: 1U00 0000b, 复位脚: U100 0000b, 其他: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	**RSTPINF**	**HardF**	**-**	**GF2**	**UART0PX**	**0**	**DPS**
读/写	读/写	读/写	-	读/写	读/写	只读	读/写

位	名称	描述
6	**RSTPINF**	**外部复位标志位** 当MCU发生外部复位引脚复位，此位由硬件置位。建议由软件清0。
5	**HardF**	**硬件故障复位标志** 一旦程序计数器(PC)的值超过flash的大小，MCU将会复位并且此位由硬件置位。建议通过软件清除此位。 注意：如果MCU运行在OCD调试模式并且OCDEN = 0，硬件故障复位将会被禁止，只有HardF标志位有效。

6.2.3.5 *看门狗定时器复位*

门狗定时器WDT是一个自由运行的定时器，带可编程的超时间隔和专用的内部时钟源。用户可以随时清零WDT，使其重新开始计数。当发生所选超时，且一段时间内没有软件响应时，WDT将直接复位系统，CPU将从0000H开始执行。

一旦发生由WDT引起的复位，WDT复位标志WDTRF(WDCON.3)将被置位。除了上电复位和WDT本身复位，任何其他复位后此位保持不变。用户可以通过软件清除WDTRF。

WDCON – 看门狗定时器控制

寄存器	特殊功能寄存器地址	复位值
WDCON	AAH, SFR页0, TA保护	上电复位POR： 0000_0111 b WDT 0000_1UUU b 其他： 0000_UUUU b

7	6	5	4	3	2	1	0
WDTR	**WDCLR**	**WDTF**	**WIDPD**	**WDTRF**	**WDPS[2:0]**		
读/写	读/写	读/写	读/写	读/写	读/写		

位	名称	描述
3	**WDTRF**	**WDT复位标志** 当MCU因WDT超时事件而复位时，此位由硬件置位。建议由软件清除此标志位。

6.2.3.6 *软件复位*

CM1003系列提供软件复位功能，允许软件复位整个系统，类似于外部复位，在复位状态下初始化MCU。软件复位在ISP进程结束时非常有用。例如，如果引导代码更新用户代码的 ISP 完成，则可以触发软件复位，重新启动CPU以立即执行新的用户代码。向SWRST(CHPCON.7)写入1将触发软件复位。请注意，此位受写入TA保护。设置SWRST位的指令是设备复位前执行的最后一条指令。请参阅下面的演示代码。

如果发生软件复位，SWRF(AUXR0.7)将由硬件自动置位。用户可以将其作为复位源指示来进行检测。除上电复位或软件复位本身外，任何其他复位后SWRF保持不变。SWRF可以通过软件清零。

CHPCON – 芯片控制

寄存器	特殊功能寄存器地址	复位值
CHPCON	9FH, 所有SFR页, TA保护	软件复位: 0000_00U0 b 其他:0000_00C0 b

7	6	5	4	3	2	1	0
SWRST	**IAPFF**	**-**	**-**	**-**	**-**	**BS**	**IAPEN**
只写	读/写	-	-	-	-	读/写	读/写

位	名称	描述
7	**SWRST**	**软件复位** 此位置位将执行软件复位，复位完成后此位由硬件自动清零

AUXR1 – 辅助寄存器 1

寄存器	特殊功能寄存器地址	复位值
AUXR1	A2H , SFR页0	上电复位: 0000 0000b, 软件复位: 1U00 0000b, 复位脚: U100 0000b, 其他: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	**RSTPINF**	**HardF**	**-**	**GF2**	**UART0PX**	**0**	**DPS**
读/写	读/写	读/写	-	读/写	读/写	只读	读/写

位	名称	描述
7	**SWRF**	**软件复位标志位** 当MCU发生软件复位后，此位由硬件置位。建议由软件清0。

软件演示代码如下。

```
ANL   AUXR1,#01111111b      ;软件复位标志位清除
CLR   EA
MOV   TA,#0Aah
MOV   TA,#55h
ORL   CHPCON,#10000000b     ;软件复位
```

6.2.3.7 启动选择

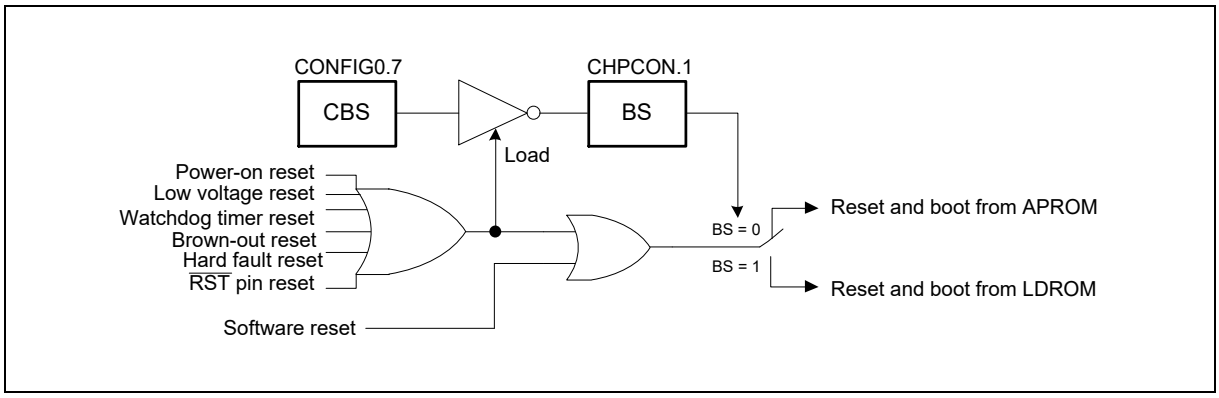


图 6.2-4 启动选择结构图

CM1003系列为用户提供了灵活的启动选择，适用于各种应用。CHPCON.1中的BS位决定 MCU 在任何复位后从APROM还是LDROM启动。如果发生复位且BS为0，MCU将从APROM的0000H地址重新启动。否则，MCU将从LDROM的0000H地址重新启动。请注意，除软件复位外，所有复位后BS均加载CONFIG0.7中CBS位的相反值。

CONFIG0

7	6	5	4	3	2	1	0
CBS	**-**	**OCDPWM**	**OCDEN**	**-**	**-**	**LOCK**	**-**
读/写	-	读/写	读/写	-	-	读/写	-

出厂默认值：1111 1111b

位	名称	描述
7	**CBS**	**配置开机启动选择** 此位定义了除软件复位外的所有复位后，MCU选择从哪个块重新启动。 1 = 除软件复位外的所有复位后，MCU从APROM启动 0 = 除软件复位外的所有复位后，MCU从LDROM启动

CHPCON – 芯片控制

寄存器	特殊功能寄存器地址	复位值
CHPCON	9FH, 所有SFR页, TA保护	软件复位: 0000_00U0 b 其他:0000_00C0 b

7	6	5	4	3	2	1	0
SWRST	**IAPFF**	**-**	**-**	**-**	**-**	**BS**[1]	**IAPEN**
只写	读/写	-	-	-	-	读/写	读/写

位	名称	描述
1	**BS**	**启动选择** 此位定义MCU在全部复位后由哪块重新启动 0 = 全部复位后，MCU将从APROM重新启动 1 = 全部复位后，MCU将从LDROM重新启动
注意： 除软件复位外的其他复位后，BS 的初始化方式为加载 CONFIG0.7 中 CBS 位的反转值。软件复位后，BS 保持不变。		

MCU从复位状态释放后，硬件将始终检查BS位(非CBS位)来确定设备从哪个区块重新启动。

6.2.3.8 复位状态

除上电复位外，其他复位状态均不影响片内RAM。复位期间RAM中的数据将被保留。上电复位后，RAM内容将不确定。

复位后，大多数特殊功能寄存器 (SFR)会恢复其初始值，除了某些会被不同复位事件影响的位。程序计数器会被强制设置为0000H，并在复位期间一直保持该值。需要注意的是，堆栈指针也会被复位为07H，因此即使RAM内容没有改变，堆栈内容也可能会在复位事件期间丢失。

复位后，所有外设和中断均被禁用。I/O端口引脚恢复FFH，并恢复为仅输入模式。

6.2.4 中断系统

6.2.4.1 概述

中断的目的是使软件处理非计划或异步事件。CM1003系列具有四个中断优先级，24个中断源。每个中断源都有单独的优先级设置位、中断向量和使能位。此外，中断可以全局使能或禁用。发生中断时，CPU应执行中断服务程序(ISR)。ISR 位于预定地址，如表6.2-3 。如果中断使能，则发生中断时，CPU将根据中断源跳转到相应的中断向量地址，执行此处程序代码，保持中断服务状态直到ISR完成。一旦ISR开始执行，它只能被更高优先级的中断打断。ISR 通过中断返回指令RETI来终止。此指令将强制CPU返回到中断发生时的下一条指令。

中断源	向量地址	向量号	中断源	向量地址	向量号
复位	0000H	-	输入捕获中断	0063H	12
外部中断0	0003H	0	PWM0中断	006BH	13
定时器0溢出	000BH	1	故障刹车中断	0073H	14
外部中断1	0013H	2	串口1 中断	007BH	15
定时器1溢出	001BH	3	定时器3溢出	0083H	16
串口0中断	0023H	4	自唤醒定时器中断	008BH	17
定时器2事件	002BH	5	PWM1中断	0093H	18
I^2C状态/超时中断	0033H	6	PWM2中断	009BH	19
引脚中断	003BH	7	PWM3中断	00A3H	20
欠压检测中断	0043H	8	SC0中断	00ABH	21
SPI中断	004BH	9	SC1中断	00B3H	22
WDT中断	0053H	10	SC2中断	00BBH	23
ADC中断	005BH	11			

表6.2-3 中断向量

6.2.4.2 使能中断

每个中断源都可以通过使用 IE 和 EIE0 特殊功能寄存器中的相关中断使能位来使能或禁用。此外，还有一个全局使能位 EA 位 (IE.7)，清零该位即可同时禁用所有中断。EA 位设置为使能所有单独使能的中断。将 EA 位设置为逻辑 0 即可禁用所有中断源，无论各个中断使能设置如何。请注意，当 EA 位设置为逻辑 0 时发生的中断将处于待处理状态，直到 EA 位重新设置为逻辑 1 时才会得到处理。所有产生中断的中断标志也可以通过软件设置。从而可以生成由软件触发的中断。

请注意，每个中断，如果使能，都是通过将其中断标志设置为逻辑 1 来生成的，无论是通过硬件还是软件实现的。用户应在各自的中断服务程序 (ISR) 中处理每个中断标志。大多数中断标志应通过软件写入逻辑 0 来清除，以避免递归中断请求。

IE – 中断使能

寄存器	特殊功能寄存器地址	复位值
IE	A8H, 所有SFR页, 可位寻址	0000 _0000 b

7	6	5	4	3	2	1	0
EA	**EADC**	**EBOD**	**ES**	**ET1**	**EX1**	**ET0**	**EX0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**EA**	**使能所有中断** 此位全局使能/禁用所有单独使能的中断。 0 = 所有中断源禁用。 1 = 每个中断根据其各自的屏蔽设置单独启用。如果启用，将产生相应的单独中断。
6	**EADC**	**ADC中断使能** 0 = ADC 中断禁用。 1 = 由ADCF(ADCCON0.7)产生的中断使能。
5	**EBOD**	**欠压中断使能** 0 = 欠压中断禁用。 1 = 由BOF(BODCON0.3)产生的中断使能。
4	**ES**	**串口0中断使能** 0 = 串口0中断禁用。 1 = 由TI(SCON.1)或RI(SCON.0)产生的中断使能。
3	**ET1**	**定时器1中断使能** 0 = 定时器1中断禁用。 1 = 由TF1(TCON.7)产生的中断使能。
2	**EX1**	**外部中断1使能** 0 = 外部中断1禁用。 1 = 由 $\overline{\text{INT1}}$ pin(P1.7)产生的中断使能。
1	**ET0**	**定时器0中断使能** 0 = 定时器0中断禁用。 1 = 由TF0(TCON.5)产生的中断使能。
0	**EX0**	**外部中断0使能** 0 = 外部中断0禁用。 1 = 由 $\overline{\text{INT0}}$ pin(P3.0)产生的中断使能。

EIE – 扩展中断使能

寄存器	特殊功能寄存器地址	复位值
EIE	9BH, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
ET2	ESPI	EFB	EWDT	EPWM0	ECAP	EPI	EI²C
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**ET2**	**定时器2中断使能** 0 = 定时器2中断禁止。 1 = 由 TF2(T2CON.7)产生的中断使能。
6	**ESPI**	**SPI 中断使能** 0 = SPI 中断禁止。 1 = 由 SPIF(SPSR.7), SPIOVF(SPSR.5)或 MODF(SPSR.4)产生的中断使能。
5	**EFB**	**故障刹车中断使能** 0 = 故障刹车中断禁止。 1 = 由FBF(FBD.7)产生的中断使能。
4	**EWDT**	**看门狗中断使能** 0 = 看门狗中断禁止。 1 = 由 WDTF(WDCON.5) 产生的中断使能。
3	**EPWM0**	**PWM中断使能** 0 = PWM中断禁止。 1 = 由 PWMF(PWMCON0.5) 产生的中断使能。
2	**ECAP**	**输入捕获中断使能** 0 = 输入捕获中断禁止。 1 = 由 CAPF[2:0](CAPCON0[2:0])中的任意标志位产生的中断使能。
1	**EPI**	**引脚中断使能** 0 = 引脚中断禁止。 1 = 由 PIF 寄存器中的任意标志位产生的中断使能。
0	**EI²C**	**I²C中断使能** 0 = I²C 中断禁止。 1 = 由 SI(I2CON.3)或 I2TOF(I2TOC.0) 产生的中断使能。

EIE1 – 扩展中断使能 1

寄存器	特殊功能寄存器地址	复位值
EIE1	9CH, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	EPWM3	EPWM2	EPWM1	EWKT	ET3	ES_1
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
5	**EPWM3**	**PWM3中断使能** 0 = PWM3中断禁止。 1 = 由PWM3F (PWM3CON0.5) 产生的中断使能。
4	**EPWM2**	**PWM2中断使能** 0 = PWM2中断禁止。 1 = 由PWM2F (PWM2CON0.5) 产生的中断使能。
3	**EPWM1**	**PWM1中断使能** 0 = PWM1中断禁止。 1 = 由PWM1F (PWM1CON0.5) 产生的中断使能。
2	**EWKT**	**WKT中断使能** 0 = WKT中断禁止。 1 = 由WKTF(WKCON.4) 产生的中断使能。
1	**ET3**	**定时器3中断使能** 0 = 定时器3中断禁止。 1 = 由TF3(T3CON.4) 产生的中断使能。
0	**ES_1**	**串口1中断使能** 0 = 串口1中断禁止。 1 = 由TI_1(SCON_1.1)或 RI_1(SCON_1.0)产生的中断使能。

6.2.4.3 中断优先级

中断有四种优先级，分别为最高、高、低和最低，用3级、2 级、1级和0级表示。中断源可以单独设置各自的优先级位来配置其优先级。表6.2-4 列出了四种优先级设置。当然，低优先级中断本身可以被高优先级中断打断，但不能被任何同级或低优先级中断打断。此外，中断之间存在预定义的自然优先级。当中断控制器需要处理具有相同优先级的同时请求时，自然优先级就会发挥作用。

对于多个中断，适用以下规则：

1. 当低优先级中断处理程序正在运行时，如果高优先级中断到达，则当前处理程序将被打断，高优先级处理程序将运行。当高优先级处理程序执行“RETI”时，低优先级处理程序将恢复运行。当该处理程序执行“RETI”时，控制权将交还给主程序。
2. 如果高优先级中断正在运行，则它不能被任何其他中断源中断——即使该中断的自然优先级更高。
3. 仅当没有其他中断正在执行时，才会调用低优先级中断处理程序。同样，低优先级中断不能抢占另一个低优先级中断，即使后者的自然优先级更高。
4. 如果两个中断同时发生，则优先级较高的中断将首先执行。如果两个中断的优先级相同，则自然优先级较高的中断将首先执行。这是自然优先级唯一起作用的情况。

自然优先级的定义如表 6.2-4 中断优先级层级设置

。它还总结了中断源、标志位、向量地址、使能位、优先级位、自然优先级以及允许 CPU 从掉电模式唤醒的权限。有关将 CPU 从掉电模式唤醒的详细信息，请参阅章节**錯誤! 找不到參照來源。**“**錯誤! 找不到參照來源。**”

中断优先级控制位		中断优先级层级
IPH / EIPH / EIPH1	IP / EIP / EIP2	
0	0	0级(最低)
0	1	1级
1	0	2级
1	1	3级(最高)

表 6.2-4 中断优先级层级设置

中断源	向量地址	中断标志位(S)	使能位	自然优先级	优先级控制位	掉电唤醒
复位	0000H	-	始终使能	最高	-	支持
外部中断0	0003H	IE0[1]	EX0	1	PX0, PX0H	支持
欠压	0043H	BOF (BODCON0.3)	EBOD	2	PBOD, PBODH	支持
看门狗定时器	0053H	WDTF (WDCON.5)	EWDT	3	PWDT, PWDTH	支持
定时器0	000BH	TF0[2]	ET0	4	PT0, PT0H	不支持
I^2C 状态/超时	0033h	SI + I2TOF (I2TOC.0)	EI2C	5	PI2C, PI2CH	不支持
ADC	005Bh	ADCF	EADC	6	PADC, PADCH	不支持
外部中断1	0013H	IE1[1]	EX1	7	PX1, PX1H	支持
引脚中断	003BH	PIF0 to PIF7 (PIF)[3]	EPI	8	PPI, PPIH	支持
定时器1	001BH	TF1[2]	ET1	9	PT1, PT1H	不支持
串口0	0023H	RI + TI	ES	10	PS, PSH	不支持
故障刹车事件	0073h	FBF (PWM0FBD.7)	EFB	11	PFB, PFBH	不支持
SPI	004Bh	SPIF (SPSR.7) + MODF (SPSR.4) + SPIOVF (SPSR.5)	ESPI	12	PSPI, PSPIH	不支持
定时器2	002BH	TF2[2]	ET2	13	PT2, PT2H	不支持
输入捕获	0063H	CAPF[2:0] (CAPCON0[2:0])	ECAP	14	PCAP, PCAPH	不支持
PWM 中断	006BH	PWMF	EPWM0	15	PPWM, PPWMH	不支持
串口1	007BH	RI_1 + TI_1	ES_1	16	PS_1, PSH_1	不支持
定时器3	0083H	TF3[2] (T3CON.4)	ET3	17	PT3, PT3H	不支持
自唤醒定时器	008BH	WKTF (WKCON.4)	EWKT	18	PWKT, PWKTH	支持
PWM1中断	0093H	PWM1F	EPWM1	19	PPWM1, PPWM1H	不支持
PWM2中断	009BH	PWM2F	EPWM2	20	PPWM2, PPWM2H	不支持
PWM3中断	00A3H	PWM3F	EPWM3	21	PPWM3, PPWM3H	不支持
SC0中断	00ABH	ACERR+BGT+TERR+TBE+RDA (SC0IS[4:0])	SC0IE	22	PSC0, PSC0H	不支持
SC1中断	00B3H	ACERR+BGT+TERR+TBE+RDA (SC1IS[4:0])	SC1IE	23	PSC1, PSC1H	不支持
SC2中断	00BBH	ACERR+BGT+TERR+TBE+RDA (SC2IS[4:0])	SC2IE	24	PSC2, PSC2H	不支持

注意：
1. .当外部中断引脚设置为边沿触发(Itx = 1)时，执行中断服务程序(ISR)时，其自身中断标志位 Iex 会被自动清除。当设置为电平触发(Itx = 0)时，Iex的值将与相应引脚的状态相反。不受软件控制。
2. 执行中断服务程序(ISR)时，TF0、TF1 或 TF3 会被自动清除。与之相反， TF2 不会被清除。
3. 如果引脚中断通道 n 选择电平触发，则 PIFn 标志反映相应通道的状态，它不受软件控制。.

表 6.2-5 各中断源特性表

IP – 中断优先级

寄存器	特殊功能寄存器地址	复位值
IP	B8H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
-	**PADC**	**PBOD**	**PS**	**PT1**	**PX1**	**PT0**	**PX0**
-	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
6	**PADC**	ADC中断优先级低位
5	**PBOD**	欠压检测中断优先级低位
4	**PS**	串口0中断优先级低位
3	**PT1**	定时器1中断优先级低位
2	**PX1**	外部中断1优先级低位
1	**PT0**	定时器0中断优先级低位
0	**PX0**	外部中断0优先级低位
注： IP与 IPH 分别组合使用，决定每个中断源的优先级。有关正确的中断优先级配置，请参见 表6.2-4 中断优先级层级设置		

IPH – 中断优先级高

寄存器	特殊功能寄存器地址	复位值
IPH	B7H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H
-	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
6	PADC	ADC中断优先级高位
5	PBOD	欠压检测中断优先级高位
4	PSH	串口0中断优先级高位
3	PT1H	定时器1中断优先级高位
2	PX1H	外部中断1中断优先级高位
1	PT0H	定时器0 中断优先高位
0	PX0H	外部中断0中断优先级高位
注： IPH与 IP 分别组合使用，决定每个中断源的优先级。有关正确的中断优先级配置，请参见 表6.2-4 中断优先级层级设置		

EIP – 扩展中断优先级

寄存器	特殊功能寄存器地址	复位值
EIP	EFH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PT2	**PSPI**	**PFB**	**PWDT**	**PPWM**	**PCAP**	**PPI**	**PI²C**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**PT2**	定时器2中断优先级低位
6	**PSPI**	SPI 中断优先级低位
5	**PFB**	故障刹车中断优先级低位
4	**PWDT**	看门狗定时器WDT中断优先级低位
3	**PPWM**	PWM中断优先级低位
2	**PCAP**	输入捕获中断优先级低位
1	**PPI**	引脚中断优先级低位
0	**PI²C**	I^2C中断优先级低位
Note : EIP与 EIPH 结合使用，确定每个中断源的优先级。有关正确的中断优先级配置，请参阅表6.2-4 中断优先级层级设置。		

EIPH – 扩展中断优先级高字节[4]

寄存器	特殊功能寄存器地址	复位值
EIPH	F7H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PT2H	**PSPIH**	**PFBH**	**PWDTH**	**PPWMH**	**PCAPH**	**PPIH**	**PI2CH**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**PT2H**	定时器2中断优先级高位
6	**PSPIH**	SPI中断优先级高位
5	**PFBH**	故障中断优先级高位
4	**PWDTH**	WDT看门狗中断优先级高位
3	**PPWMH**	PWM中断优先级高位
2	**PCAPH**	输入捕获中断优先级高位
1	**PPIH**	引脚中断优先级高位
0	**PI2CH**	I^2C中断优先级高位
注意： EIPH 与 EIP 配合使用，用于确定每个中断源的优先级。有关正确的中断优先级配置，请参阅 表6.2-4 中断优先级层级设置。		

EIP1 – 扩展中断优先级 1

寄存器	特殊功能寄存器地址	复位值
EIP1	FEH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	**PPWM3**	**PPWM2**	**PPWM1**	**PWKT**	**PT3**	**PS_1**
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
5	**PPWM3**	PWM3 中断优先级低位
4	**PPWM2**	PWM2 中断优先级低位
3	**PPWM1**	PWM1 中断优先级低位
2	**PWKT**	WKT 中断优先级低位
1	**PT3**	定时器3 中断优先级低位
0	**PS_1**	串口1 中断优先级低位
注意： EIP1 与EIPH1 配合使用，用于确定每个中断源的优先级。有关正确的中断优先级配置，请参见 表6.2-4 中断优先级层级设置		

EIPH1 – 扩展中断优先级高字节1

寄存器	特殊功能寄存器地址	复位值
EIPH1	FFH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	**PPWM3H**	**PPWM2H**	**PPWM1H**	**PWKTH**	**PT3H**	**PSH_1**
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
5	**PPWM3H**	PWM3 中断优先级高位
4	**PPWM2H**	PWM2 中断优先级高位
3	**PPWM1H**	PWM1 中断优先级高位
2	**PWKTH**	WKT 中断优先级高位
1	**PT3H**	定时器3 中断优先级高位
0	**PSH_1**	串口1 中断优先级高位
注意： EIPH1与EIP1 配合使用，用于确定每个中断源的优先级。有关正确的中断优先级配置，请参见 表6.2-4 中断优先级层级设置		

EIP2 – 扩展中断优先级 2

寄存器	特殊功能寄存器地址	复位值
EIP2	CEH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	**PSC2**	**PSC1**	**PSC0**
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	**PSC2**	SC2 / UART4 中断优先级低位
1	**PSC1**	SC1 / UART3 中断优先级低位
0	**PSC0**	SC0 / UART2 中断优先级低位
注意：EIP2 与 EIPH2 配合使用，用于确定每个中断源的优先级。有关正确的中断优先级配置，请参阅表6.2-4 中断优先级层级设置 。		

EIPH2 – 扩展中断优先级高字节 2

寄存器	特殊功能寄存器地址	复位值
EIPH2	CFH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	**PUART4H**	**PUART3H**	**PUART2H**
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	**PUART4H**	SC2 / UART4 中断优先级高位
1	**PUART3H**	SC1 / UART3 中断优先级高位
0	**PUART2H**	SC0 / UART2 中断优先级高位
注意：EIPH2与 EIP2 配合使用，用于确定每个中断源的优先级。有关正确的中断优先级配置，请参阅表6.2-4 中断优先级层级设置 。		

6.2.4.4 中断源

每个系统时钟周期都会对中断标志进行采样。在同一周期内，系统会轮询采样到的中断并确定其优先级。如果满足某些条件，硬件将执行内部生成的LCALL指令，将进程引导至对应的中断向量地址。生成LCALL的条件如下:

1. 当前没有同等或更高优先级未处理。
2. 当前轮询中断标志周期是当前正在执行指令的最后一个周期。
3. 当前指令不涉及写入任何中断使能位或优先级置位，并且也不是中断返回指令RETI。

上述条件任一不满足，则不会生成LCALL指令。在每个指令周期都会轮询检测中断标志一次。如果中断标志在一个周期内处于活动状态，但由于上述条件不满足而未得到响应，即使后面满足上述条件该标志不再处于活动状态，则被拒绝的中断仍将不会得到处理。这意味着不会记住曾经处于活动状态但未得到服务的中断标志位。每个轮询周期都需要重新检测中断标志位。

处理器通过执行LCALL指令来响应有效中断，该指令将中断转发到相应的服务程序。根据不同的中断源，此操作可能会清除中断标志位，也有可能不会清除。硬件LCALL的行为与软件LCALL指令完全相同。该指令将程序计数器内容保存到堆栈RAM中，但不保存程序状态字(PSW)。PC 指针会重新加载产生LCALL的中断的中断向量地址。程序会从向量地址继续执行，直到执行RETI指令。执行RETI指令时，处理器弹出堆栈并将堆栈顶部的内容加载到PC。用户应注意堆栈的状态。如果堆栈内容被修改，处理器不会察觉变化，仍将会从堆栈加载PC的地址继续执行。请注意，简单的RET指令与RETI指令表现完全相同，但它不会通知中断控制器中断服务程序已完成。RET会让控制器认为中断服务程序仍然在进行中，使后面的中断不能再被执行。

6.2.4.5 中断延迟

每个中断源的响应时间取决于多种因素，例如中断的自身性质和正在执行的指令。每个系统时钟周期都会轮询每个中断标志并对其进行优先级解析。如果一个中断请求处于活动状态并且满足先前3个条件，则执行硬件生成的LCALL指令。LCALL本身需要4个时钟周期才能完成。因此，从设置中断标志到执行中断服务程序，最少需要5个时钟周期的反应时间。

如果三个条件中的任一不满足，则应预知更长的响应时间。如果更高优先级或同等优先级中断正在执行中，则中断延迟时间显然取决于当前正在执行的中断服务程序。如果轮询周期不是正在执行指令的最后一个时钟周期，则会有额外的延迟。如果设备正在执行 RETI，然后执行最长的 6 个时钟周期指令作为下一条指令，则会出现最大响应时间（如果当前没有其他中断正在服务或新中断的优先级更高）。从、中断源被激活(未被检测到)开始，最长反应时间为16个时钟周期。此时间段包括 5 个时钟周期用于完成 RETI，6 个时钟周期用于完成最长指令，1 个时钟周期用于检测中断，以及 4 个时钟周期用于完成硬件 LCALL 到中断向量位置的指令。

因此在单中断系统中，中断响应时间始终大于5个时钟周期，但不超过16个时钟周期。

6.3 Flash存储控制

6.3.1 在应用编程 (IAP)

与RAM的实时操作不同，更新FLASH数据通常需要很长时间。此外，擦除、编程或读取FLASH数据的时序过程也相当复杂。CM1003系列提供了便捷的FALSH操作机制，方便用户通过在应用编程(IAP)对FLASH内容重新编程。IAP是一种通过软件进行在线电擦除和编程的方法。

通过设置IAPEN (CHPCON.0，受TA保护)使能IAP，并设置IAPUEN中的相应位，使能允许更新的FLASH区块(CONFIG、LDROM、APROM)，用户可以轻松地向IAPAH和IAPAL填入16位目标地址，向IAPFD写入数据，向IAPCN写入命令。然后通过设置触发位IAPGO(IAPTRG.0)，IAP即可开始。请注意，IAPTRG也受TA保护。此时，CPU保持程序计数器，内置IAP自动控制内部电压和信号时序。擦除和编程时间由内部控制，不受工作电压和频率的影响。通常页擦除时间为5ms，字节编程时间为23.5μs。IAP操作完成后，程序计数器继续运行后续指令。IAPGO位将自动清零。IAP失败标志位IAPFF(CHPCON.6)可用于检查上一次IAP 操作是否成功。通过此过程，用户只需用纯软件即可轻松擦除、编程和验证FLASH。

6.3.1.1 IAP 命令

CM1003系列提供多种应用程序来执行对APROM、LDROM 或 CONFIG 字节的IAP操作。IAP操作模式和FLASH编程的目标区域由IAP 控制寄存器IAPCN 决定。.

IAP 模式	IAPCN				IAPA[15:0] {IAPAH, IAPAL}	IAPFD[7:0]
	IAPB [1:0]	FOEN	FCEN	FCTRL [3:0]		
读公司ID	XX[1]	0	0	1011	X	DAH
读器件ID	XX	0	0	1100	低字节 DID: 0000H 高字节 DID: 0001H	低字节 DID 高字节 DID
读96位UID	XX	0	0	0100	0000H ~ 000BH	数据读出
APROM 页擦除	00	1	0	0010	地址写入[2]	FFH
LDROM 页擦除	01	1	0	0010	地址写入[2]	FFH
APROM 字节编程	00	1	0	0001	地址写入	数据写入
LDROM 字节编程	01	1	0	0001	地址写入	数据写入
APROM 字节读	00	0	0	0000	地址写入	数据读出
LDROM 字节读	01	0	0	0000	地址写入	数据读出
擦除所有CONFIG	11	1	0	0010	0000H	FFH

IAP 模式	IAPCN				IAPA[15:0] {IAPAH, IAPAL}	IAPFD[7:0]
	IAPB [1:0]	FOEN	FCEN	FCTRL [3:0]		
CONFIG 字节编程	11	1	0	0001	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H CONFIG6: 0005H	数据写入
CONFIG 字节读	11	0	0	0000	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H CONFIG6: 0005H	数据读出

注意：

1. “X” 表示无关项。
2. 每一页是 128字节，因此，该地址应该是指向目标页面的地址。

表6.3-1 IAP 模式和命令代码

6.3.1.2 IAP控制寄存器

以下寄存器与IAP处理有关。

CONFIG2

7	6	5	4	3	2	1	0
CBODEN	CBOV[2:0]			BOIAP	CBORST	-	-
读/写	读/写			读/写	读/写	-	-

出厂默认值：1111 1111b

位	名称	描述
3	BOIAP	**欠压抑制IAP** 此位决定欠压状态是否抑制IAP擦除及编程功能。此位仅当欠压检测功能使能后有效。 1 = V_{DD} 低于 V_{BOD} ,抑制IAP 擦除或编程功能。 0 = 任何可用V_{DD}下，都允许IAP擦除及编程功能。

CHPCON – 芯片控制

寄存器	特殊功能寄存器地址	复位值
CHPCON	9FH, 所有SFR页, TA保护	软件复位: 0000_00U0 b 其他:0000_00C0 b

7	6	5	4	3	2	1	0
SWRST	IAPFF	-	-	-	-	BS	IAPEN
只写	读/写	–	–	–	–	读/写	读/写

位	名称	描述
6	IAPFF	**IAP 故障标志** 在IAPGO(ISPTRG.0)置位后，如果满足以下任意条件，硬件将置位此位： (1)访问地址超过其大小的区域 (2)IAPCN 命令无效 (3)IAP擦除或编程更新未使能的区块 (4)当BOIAP(CONFIG2.5)在未编辑的1、 BODEN(BODCON0.7)为1以及BORST(BODCON0.2)为0 时，IAP在V_{BOD}模式下执行擦除或编程。 此位应由软件清零。
0	IAPEN	**IAP 使能** 0 = IAP功能禁用 1 = IAP功能使能 一旦使能IAP功能，HIRC将会被打开作为时序控制。如果希望降低功耗，应始终在 IAP 操作后的最后一条指令清零 IAPEN，以停止内部振荡器。

IAPUEN – IAP更新使能

寄存器	特殊功能寄存器地址	复位值
IAPUEN	A5H, SFR页0, TA保护	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	-	**SPMEN**	**SPUEN**	**CFUEN**	**LDUEN**	**APUEN**
-	-	-	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:5	-	保留
4	**SPMEN**	**SPROM存储空间映射使能** 0 =CPU存储地址0xff80~0xffff 映射到APROM存储。 1 =CPU存储地址0xff80~0xffff 映射到SPROM存储。
3	**SPUEN**	**SPROM存储空间更新使能(TA保护)** 0 = 禁止通过IAP擦除或编程SPROM字节。 1 = 允许通过IAP擦除或编程SPROM字节。
2	**CFUEN**	**配置字节更新使能(TA保护)** 0 = 禁止通过IAP擦除或编程CONFIG字节。 1 = 允许通过IAP擦除或编程CONFIG字节。
1	**LDUEN**	**LDROM更新使能(TA保护)** 0 = 禁止通过IAP擦除或编程LDROM。 1 = 允许通过IAP擦除或编程LDROM。
0	**APUEN**	**APROM更新使能(TA保护)** 0 = 禁止通过IAP擦除或编程APROM。 1 = 允许通过IAP擦除或编程APROM。

IAPCN – IAP 控制

寄存器	特殊功能寄存器地址	复位值
IAPCN	AFH, SFR页0	0011_0000 b

位	名称	描述
7:6	IAPB[1:0]	IAP控制 此字节用于 IAP 命令。 详细描述参见表6.3-1 IAP.
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

IAPAH – IAP地址高字节

寄存器	特殊功能寄存器地址	复位值
IAPAH	A7H, SFR页0	0000 _0000 b

位	名称	描述
7:0	**IAPA[15:8]**	**IAP地址高字节** IAPAH包含用于IAP操作的地址 IAPA[15:8] 。

IAPAL – IAP地址低字节

寄存器	特殊功能寄存器地址	复位值
IAPAL	A6H, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPA[7:0]							
读/写							

位	名称	描述
7:0	**IAPA[7:0]**	**IAP地址低字节** IAPAL包含用于IAP操作的地址 IAPA[7:0] 。

IAPFD – IAP Flash数据

寄存器	特殊功能寄存器地址	复位值
IAPFD	AEH, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
IAPFD[7:0]							
读/写							

位	名称	描述
7:0	**IAPFD[7:0]**	**IAP Flash数据** 此字节包含 Flash 数据，这些数据是从 Flash 中读取的，或者即将写入 Flash 的。编程模式下，用户应在触发 IAP 处理之前将数据写入 IAPFD，读取/ 验证模式下，在 IAP 处理完成后从 IAPFD 读取数据。

IAPTRG – IAP触发

寄存器	特殊功能寄存器地址	复位值
IAPTRG	A4H, SFR页0, TA保护	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	**IAPGO**
-	-	-	-	-	-	-	只写

位	名称	描述
7:1	-	保留
0	**IAPGO**	**IAP执行** 置位此位为逻辑1时开始执行IAP。在此指令之后，CPU保持程序计数器(PC)值不变，IAP硬件自动接管进程控制。IAP操作完成后，程序计数器继续执行下一条指令。IAPGO位将被自动清除并始终读取为逻辑0。 在触发IAP操作之前，应临时禁用中断(如果已使能)，以限制硬件执行。 程序流程如下。 CLR EA MOV TA,#0AAH MOV TA,#55H ORL IAPTRG,#01H (SETB EA)

6.3.1.3 IAP用户指南

IAP可以方便用户更新FLASH内容，但是用户应遵循一些限制规则，以确保IAP正常运行。未注意到警告可能会导致未知结果，甚至严重损坏器件。此外，下文对于正确执行IAP还将提供实用的建议。

1. 如果不再需要IAP操作，用户应清除IAPEN(CHPCON.0)。可以防止系统意外触发IAP。此外，IAP需要内部振荡器HIRC运行。如果选择了外部时钟源，禁用IAP将停止HIRC以节省功耗。请注意，IAPEN 的写入受TA保护。
2. 当LOCK位(CONFIG0.1)激活时，IAP读、写或擦除仍然有效。

在IAP过程中，应通过清除EA位临时禁用中断(如果已使能)，以实施限制。

请勿尝试擦除或编程当前正在执行代码的页面。这将导致不可预测的程序行为，可能损坏程序数据。

6.3.1.4 使用Flash存储用于数据Flash

在一般应用中，需要非易失性数据存储，即使断电后其内容仍会保留，用户可以读回或更新。这些数据作为系统控制的参数或常量。CM1003的Flash支持IAP功能，并可以使用MOVC指令读取Flash中的任何字节，因此适合用作非易失性数据存储。IAP提供擦除和编程功能，可轻松擦除和编程页面中的一个或多个字节。在微控制器固件的控制下，IAP在应用程序中执行。请注意Flash写入寿命为 100,000 次。以下为参考代码。

汇编例程如下：

```
;*****************************************************************************
;此代码演示用户代码在APROM中执行时
;如何使用 IAP 将APROM201h 变为1字节的数据Flash
;*****************************************************************************
PAGE_ERASE_AP       EQU     00100010b
BYTE_PROGRAM_AP     EQU     00100001b

 ORG   0000h

 MOV   TA,#0Aah              ;CHPCON 受TA保护
 MOV   TA,#55h
 ORL   CHPCON,#00000001b     ;IAPEN = 1, 使能 IAP模式

 MOV   TA,#0Aah              ;IAPUEN 受TA保护
 MOV   TA,#55h
 ORL   IAPUEN,#00000001b     ;APUEN = 1, 使能APROM更新

 MOV   IAPCN,#PAGE_ERASE_AP ;擦除页面 200h~27Fh
 MOV   IAPAH,#02h
 MOV   IAPAL,#00h
 MOV   IAPFD,#0FFh
 MOV   TA,#0Aah              ;IAPTRG 受TA保护
 MOV   TA,#55h
 ORL   IAPTRG,#00000001b     ;向IAPGO写入'1'，触发IAP过程
 MOV   IAPCN,#BYTE_PROGRAM_AP ;用55h编程 201h
 MOV   IAPAH,#02h
 MOV   IAPAL,#01h
 MOV   IAPFD,#55h
 MOV   TA,#0Aah
```

```
MOV   TA,#55h
ORL   IAPTRG,#00000001b

MOV   TA,#0Aah
MOV   TA,#55h
ANL   IAPUEN,#11111110b     ;APUEN = 0, 禁用APROM更新

MOV   TA,#0Aah
MOV   TA,#55h
ANL   CHPCON,#11111110b     ;IAPEN = 0, 禁用IAP 模式

MOV   DPTR,#201h
CLR   A
MOVC  A,@A+DPTR             ;读取201h的内容
MOV   P0,A

SJMP  $
```

C语言例程:

```
//*************************************************************************************
//    此代码演示用户代码在APROM中执行时
//    如何使用 IAP 将APROM201h 变为1字节的数据Flash.
//*************************************************************************************
*
#define  PAGE_ERASE_AP   0x22
#define  BYTE_PROGRAM_AP    0x21

/*Data Flash 作为APROM的一部分，由 MOVC 读取。Data Flash 可以定义为
“代码”区域中从绝对地址 0x0200 开始的 128 个元素的数组 */

volatile unsigned char code Data_Flash[128] _at_ 0x0200;

Main (void)
{
 TA = 0Xaa;              //CHPCON受TA保护
 TA = 0x55;
 CHPCON |= 0x01;         //IAPEN = 1, 使能IAP模式

 TA = 0Xaa;              //IAPUEN 受TA保护
 TA = 0x55;
 IAPUEN |= 0x01;         //APUEN = 1, 使能APROM 更新

 IAPCN = PAGE_ERASE_AP;  //擦除页面200h~27Fh
 IAPAH = 0x02;
 IAPAL = 0x00;
 IAPFD = 0Xff;
 TA = 0Xaa;              //IAPTRG 受TA保护
 TA = 0x55;
 IAPTRG |= 0x01;         //向IAPGO写入'1'，触发IAP过程
```

```
    IAPCN = BYTE_PROGRAM_AP;    //用55h编程 201h
    IAPAH = 0x02;

    IAPAH = 0x02;
    IAPAL = 0x01;
    IAPFD = 0x55;
    TA = 0Xaa;
    TA = 0x55;
    IAPTRG |= 0x01;             //向IAPGO写入'1'，触发IAP过程

    TA = 0Xaa;                  //IAPUEN 受TA保护
    TA = 0x55;
    IAPUEN &= ~0x01;            //APUEN = 0, 禁用APROM更新

    TA = 0Xaa;                  //CHPCON 受TA保护
    TA = 0x55;
    CHPCON &= ~0x01;            //IAPEN = 0, 禁用IAP模式

    P0 = Data_Flash[1];         //读取地址200h+1的内容

    while(1);
}
```

6.3.1.5 *在系统编程 (ISP)*

Flash存储支持硬件编程和在应用编程(IAP)。如果产品在研发阶段或终端产品需要由最终客户更新固件时，硬件编程模式会使重复再编程变得非常困难且不方便。在系统编程(ISP)使这过程简单易行。ISP更新Flash存储不需要将微控制器从系统板上拆下来。它允许在软件控制下对设备进行重新编程。此外，更新应用程序固件的功能使广泛的应用成为可能。

用户可以开发一个自己的引导代码放在LDROM中。LDROM最大为4K字节。用户开发的引导代码可以通过并行烧录器或在线电路编程(ICP)工具进行重新编程。

一般而言，ISP是通过PC和MCU之间的通信来实现的。PC通过串口将新的用户代码传输给MCU。然后引导代码接收数据并通过IAP命令重新编程到客户代码区。Nuvoton为CM1003提供了ISP固件和PC端应用程序。它使用户能够轻松地通过UART端口执行ISP。更多信息请访问Nuvoton 8位微控制器网站：Nuvoton 80C51 微控制器技术支持。以下是简单ISP的参考例程。

汇编例程代码：

```
;*****************************************************************************
;此代码演示了如何从LDROM执行APROM和CONFIG IAP。
;APROM通过代码重新编程，将P1输出为 55h，将P2输出为 aah。
;CONFIG2也被更新以禁用BOD复位。
;用户需要配置 CONFIG0 = 0x7F、CONFIG1 = 0Xfe、CONFIG2 = 0Xff。
;*****************************************************************************
PAGE_ERASE_AP       EQU     00100010b
BYTE_PROGRAM_AP     EQU     00100001b
BYTE_READ_AP        EQU     00000000b
ALL_ERASE_CONFIG    EQU     11100010b
BYTE_PROGRAM_CONFIG   EQU     11100001b
BYTE_READ_CONFIG    EQU     11000000b
```

```
 ORG   0000h

 CLR   EA                    ;禁用所有中断
 CALL  Enable_IAP

 CALL  Enable_AP_Update
 CALL  Erase_AP              ;擦除AP数据
 CALL  Program_AP            ;编程AP数据
 CALL  Disable_AP_Update
 CALL  Program_AP_Verify     ;校验已编程的AP数据

 CALL  Read_CONFIG           ;读回CONFIG2
 CALL  Enable_CONFIG_Update
 CALL  Erase_CONFIG          ;擦除CONFIG字节
 CALL  Program_CONFIG        ;用新数据编程CONFIG2
 CALL  Disable_CONFIG_Update
 CALL  Program_CONFIG_Verify   ;校验已编程的CONFIG2

 CALL  Disable_IAP
 MOV   TA,#0Aah              ;TA保护
 MOV   TA,#55h       ;
 ANL   CHPCON,#11111101b     ;BS = 0, 复位至APROM
 MOV   TA,#0Aah
 MOV   TA,#55h
 ORL   CHPCON,#80h           ;软件复位并从APROM重启

 SJMP  $

;*****************************************************************
;      IAP 子程序
;*****************************************************************
Enable_IAP:
 MOV   TA,#0Aah              ;CHPCON 受TA保护
 MOV   TA,#55h
 ORL   CHPCON,#00000001b     ;IAPEN = 1,使能IAP模式
 RET

Disable_IAP:
 MOV   TA,#0Aah
 MOV   TA,#55h
 ANL   CHPCON,#11111110b     ;IAPEN = 0, 禁用IAP模式
 RET

Enable_AP_Update:
 MOV   TA,#0Aah              ;IAPUEN 受TA保护
 MOV   TA,#55h
 ORL   IAPUEN,#00000001b     ;APUEN = 1, 使能APROM更新
 RET
```

```
Disable_AP_Update:
 MOV   TA,#0Aah
 MOV   TA,#55h
 ANL   IAPUEN,#11111110b     ;APUEN = 0, 禁用APROM更新
 RET

Enable_CONFIG_Update:
 MOV   TA,#0Aah
 MOV   TA,#55h
 ORL   IAPUEN,#00000100b     ;CFUEN = 1, 使能CONFIG更新
 RET

Disable_CONFIG_Update:
 MOV   TA,#0Aah
 MOV   TA,#55h
 ANL   IAPUEN,#11111011b     ;CFUEN = 0, 禁用CONFIG更新
 RET

Trigger_IAP:
 MOV   TA,#0Aah              ;IAPTRG 受TA保护
 MOV   TA,#55h
 ORL   IAPTRG,#00000001b     ; 向IAPGO写入'1'，触发IAP过程
 RET

;*************************************************************************
;       IAPAPROM函数
;*************************************************************************
Erase_AP:
 MOV   IAPCN,#PAGE_ERASE_AP
 MOV   IAPFD,#0FFh
 MOV   R0,#00h
Erase_AP_Loop:
 MOV   IAPAH,R0
 MOV   IAPAL,#00h
 CALL  Trigger_IAP
 MOV   IAPAL,#80h
 CALL  Trigger_IAP
 INC   R0
 CJNE  R0,#44h,Erase_AP_Loop
 RET

Program_AP:
 MOV   IAPCN,#BYTE_PROGRAM_AP
 MOV   IAPAH,#00h
 MOV   IAPAL,#00h
 MOV   DPTR,#AP_code

Program_AP_Loop:
 CLR   A
 MOVC  A,@A+DPTR
 MOV   IAPFD,A
```

```
    CALL  Trigger_IAP
    INC   DPTR
    INC   IAPAL
    MOV   A,IAPAL
    CJNE  A,#14,Program_AP_Loop
    RET

Program_AP_Verify:
    MOV   IAPCN,#BYTE_READ_AP
    MOV   IAPAH,#00h
    MOV   IAPAL,#00h
    MOV   DPTR,#AP_code
Program_AP_Verify_Loop:
    CALL  Trigger_IAP
    CLR   A
    MOVC  A,@A+DPTR
    MOV   B,A
    MOV   A,IAPFD
    CJNE  A,B,Program_AP_Verify_Error
    INC   DPTR
    INC   IAPAL
    MOV   A,IAPAL
    CJNE  A,#14,Program_AP_Verify_Loop
    RET

Program_AP_Verify_Error:
    CALL  Disable_IAP
    MOV   P0,#00h
    SJMP  $

;*********************************************************************
;       IAP CONFIG 函数
;*********************************************************************
Erase_CONFIG:
    MOV   IAPCN,#ALL_ERASE_CONFIG
    MOV   IAPAH,#00h
    MOV   IAPAL,#00h
    MOV   IAPFD,#0FFh
    CALL  Trigger_IAP
    RET

Read_CONFIG:
    MOV   IAPCN,#BYTE_READ_CONFIG
    MOV   IAPAH,#00h
    MOV   IAPAL,#02h
    CALL  Trigger_IAP
    MOV   R7,IAPFD
    RET

Program_CONFIG:
    MOV   IAPCN,#BYTE_PROGRAM_CONFIG
    MOV   IAPAH,#00h
```

```
    MOV   IAPAL,#02h
    MOV   A,R7
    ANL   A,#11111011b
    MOV   IAPFD,A              ;禁用BOD 复位
    MOV   R6,A                 ;临时数据
    CALL  Trigger_IAP
    RET

Program_CONFIG_Verify:
    MOV   IAPCN,#BYTE_READ_CONFIG
    MOV   IAPAH,#00h
    MOV   IAPAL,#02h
    CALL  Trigger_IAP
    MOV   B,R6
    MOV   A,IAPFD
    CJNE  A,B,Program_CONFIG_Verify_Error
    RET

Program_CONFIG_Verify_Error:
    CALL  Disable_IAP
    MOV   P0,#00h
    SJMP  $

;*********************************************************************
;      APROM代码
;*********************************************************************
AP_code:
    DB 75h,0B1h, 00h       ; "MOV     P0M1,#0" 的操作码
    DB 75h,0Ach, 00h       ; "MOV     P3M1,#0" 的操作码
    DB 75h, 90h, 55h       ; "MOV     P1,#55h" 的操作码
    DB 75h,0A0h,0Aah       ; "MOV     P2,#0Aah" 的操作码
    DB 80h,0Feh            ; "SJMP    $" 的操作码

    END
```

6.3.2 在线电路编程(ICP)

Flash可以通过在线电路编程(ICP)进行编程。如果产品处于研发阶段或最终产品需要更新固件，硬件编程模式会使重复编程变得困难且不方便。采用ICP方式将简单而可行，无需将微控制器从板上拆下来。ICP模式还允许客户使用未编程器件制造电路板，编程可以在组装过程结束后进行，从而可以使用最新固件或定制化固件对设备进行编程。

ICP功能涉及3个信号引脚，nRESET 、ICE_DAT和ICE_CLK。nRESET用于进入或退出 ICP模式。ICE_DAT是数据输入输出引脚，ICE_CLK是时钟输入引脚，用于同步正在编程的MCU数据输入或输出。用户需要在电路板上保留这三个引脚以及V_{DD}、V_{SS}引脚，以便进行 ICP 操作。

Nuvoton为CM1003提供ICP工具，用户可以通过Nuvoton ICP烧录器轻松进行ICP烧录。Nuvoton自主研发的ICP烧录器针对MCU的电气特性进行了优化，并满足生产过程中的稳定性与效率。更多详细信息，请浏览Nuvoton8 位微控制器网站：Nuvoton 80C51微控制器技术支持。

6.3.3 片上调试 (OCD)

6.3.3.1 概述

CM1003系列内置片上调试器(OCD)，为开发人员提供了低成本的代码调试方法，每种封装均提供。OCD提供完整的程序流程控制调试功能，包括八个硬件地址断点、单步运行、全速运行和非侵入式存储访问命令。OCD系统并不占用存储映射中的任何位置，也不共享片上外设。

当 OCDEN(CONFIG0.4)设置为0且LOCK(CONFIG0.1)保持未设置为1时，OCD被激活。如果芯片被锁定，OCD 将无法运行。OCD系统使用两线串行接口 ICE_DAT和ICE_CLK，在目标设备和调试控制器主机之间建立通信。ICE_DAT是用于调试数据传输的输入/输出引脚，ICE_CLK是用于与ICE_DAT数据同步的输入引脚。P2.0/nRESET引脚也是OCD模式进入和退出所必需的。CM1003系列通过ICP写入模式支持带有Flash存储控制路径的OCD，该模式与OCD 共用这三个引脚。

CM1003 系列使用 ICE_DAT、ICE_CLK 和 P2.0/nRESET 引脚与 OCD 系统连接。在设计使用 OCD 的系统时，必须考虑以下限制以确保正确操作：

1. 如果 P2.0/nRESET 配置为外部复位引脚，则不能将其直接连接到 V_{DD}，并且必须移除所有连接的外部电容。
2. 如果 P2.0/nRESET 配置为输入引脚 P2.0，则必须隔离所有外部输入源。
3. 必须断开所有外部复位源。
4. 必须隔离连接在 ICE_DAT 和 ICE_CLK 上的任何外部组件。

6.3.3.2 OCD的限制

CM1003系列是一款功能齐全的微控制器，其有限的I/O引脚复用了多种功能。为了给 OCD 系统提供资源，必须牺牲部分设备功能。OCD具有以下限制：

1. 需要使用 P2.0/nRESET 引脚进行 OCD 模式选择。因此，无法模拟 P2.0 输入或外部复位源。
2. ICE_DAT引脚物理上与P1.6是一个引脚，因此无法模拟该引脚的I/O功能和其他共享的多功能。
3. ICE_CLK引脚物理上与P0.2是一个引脚，因此无法模拟该引脚的I/O功能和其他共享的多功能。
4. 当系统处于空闲或掉电模式时，因为部分外设时钟已经停止，所以任何访问可能无效的。读访问可能返回一个无用的数据，写访问可能也无法成功。
5. HIRC不能关闭，因为OCD使用这个时钟来监控其内部状态。在调试模式下执行关闭HIRC的指令将不起作用。CPU在调试模式下进入掉电模式时，HIRC会保持开启状态。

CM1003 OCD系统还有另一个限制，即在用户程序运行时无法随时执行非侵入命令。非侵入命令允许用户使用调试控制器读取或写入MCU存储位置或访问状态和控制寄存器。仅当MCU在硬件断点地址匹配或单步运行后处于暂停状态时，才允许读写存储或控制寄存器。

CONFIG0

7	6	5	4	3	2	1	0
CBS	-	OCDPWM	OCDEN	-	RPD	LOCK	-
读/写	-	读/写	读/写	-	读/写	读/写	-

出厂默认值：1111 1111b

位	名称	描述
5	**OCDPWM**	**OCD 调试停止时，PWM 的输出状态** 此位决定当OCD停止CPU时，PWM的输出状态 1 =PWM输出脚为三态模式(Tri-state) 0 =PWM持续输出. 请注意，只有当PWM通道的相应PIO位设置为1时，此位才有效。
4	**OCDEN**	**OCD 调试使能** 1 = 禁用OCD调试 0 = 使能OCD调试 注意: 当MCU运行在OCD调试模式下并且OCDEN=0,硬件故障复位将被禁用,仅仅HFIF 标志位有效。

6.3.4 96位唯一标识符

每颗CM1003芯片出厂前都预编程了一个96位的序列号，该序列号保证芯片的唯一性。该序列号称为唯一码（Unique Code）。用户只能通过IAP命令读取该唯一码。请参阅章节 6.3.1.1 IAP .

6.4 通用端口 I/O (GPIO)

6.4.1 功能描述

6.4.1.1 GPIO 模式

CM1003系列最多有46个通用I/O引脚，其中45个可位寻址的通用I/O引脚，分为6个端口P0至P5，以及1个仅输入引脚P20。每个端口都有其端口控制寄存器(Px 寄存器)。端口控制寄存器的写入和读取具有不同的含义。写入端口控制寄存器将设置端口输出锁存逻辑值，而读取将获取端口引脚逻辑状态。引脚有四种模式分别是准双向(标准 8051 端口结构)、推挽输出、仅输入和开漏模式。每个端口使用两个特殊功能寄存器PxM1和PxM2 来选择端口Px的I/O模式。下表说明了如何选择Px.n的I/O模式。请注意，任何复位后，默认配置为仅输入(高阻)模式。

PnM1.X[1]	PnM2.X[1]	I/O类型
0	0	准双向
0	1	推挽
1	0	仅输入(高阻)
1	1	开漏
Note 1: n = 0~5, X = 0~7		

表6.4-1 不同I/O模式的配置

所有I/O引脚均可通过PxS寄存器中的相应位选择为TTL电平输入或施密特触发输入 。施密特触发输入具有更好的抗干扰能力。所有的I/O引脚具有位可控，通过软件选择电压转换速率(输出斜率)的能力。电压转换速率控制寄存器为PxSR。默认是慢斜率。如果用户想要提高I/O输出速率，可通过设置PxSR中的相应位，来选择更快的速率。

例如：

```
    P0M1 |= 0x40;
    P0M2 &= 0xBF;     //设置P0.6为仅输入模式
```

准双向模式

准双向模式作为标准的8051 I/O结构，可以同时用作输入和输出。当端口输出逻辑高时，端口驱动较弱，允许外部器件将引脚拉低。当引脚被拉低时，端口驱动较强，能够吸收大电流。在准双向I/O结构中，有两个上拉晶体管。每个晶体管都有不同的用途。其中一个上拉称为“特弱”上拉晶体管，当端口锁定在逻辑1时，它就会导通。如果引脚悬空，“特弱”上拉晶体管会提供非常小的电流，将引脚拉高。

第二个上拉是“强”上拉晶体管。这种上拉用于在准双向口引脚上，加速端口引脚上由逻辑0到逻辑1的转换速度。发生这种情况时，“强”上拉晶体管会开启两个 CPU 时钟周期，快速地将端口引脚拉高。然后它会关闭“特弱”上拉，并继续保持端口引脚为高。准双向端口结构如下所示。

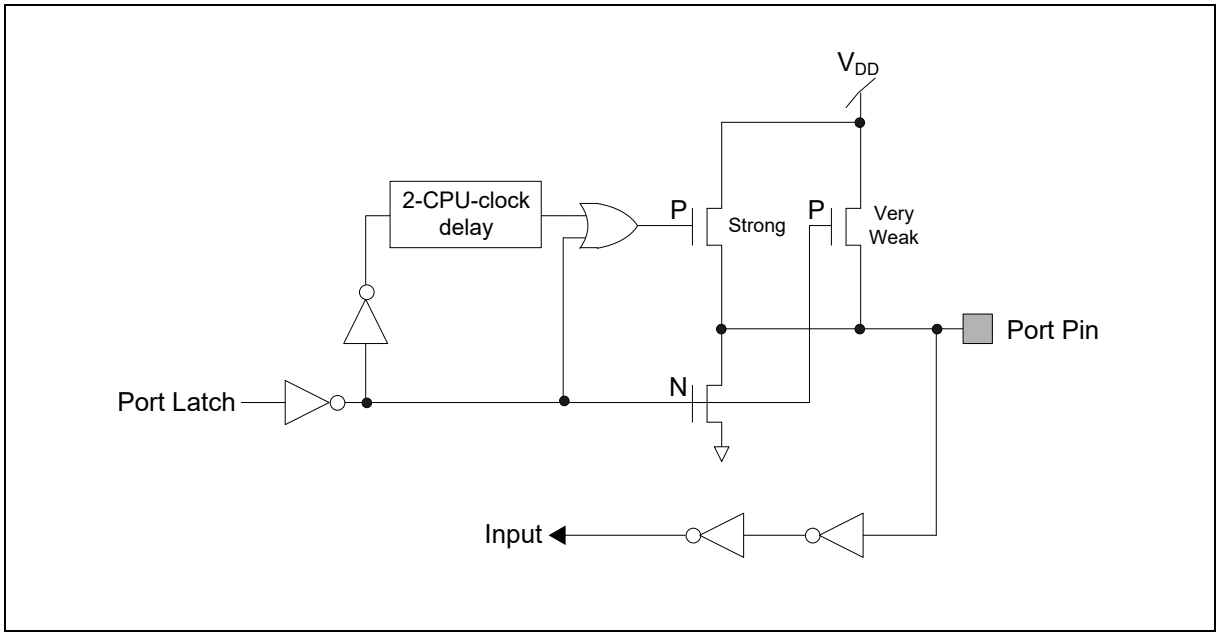


图 6.4-1 准双向模式结构

推挽模式

推挽输出模式与准双向输出模式有相同的下拉结构。但当端口写入逻辑1时，提供持续的强上拉。推挽输出模式通常用于需要从端口输出大电流时的应用。

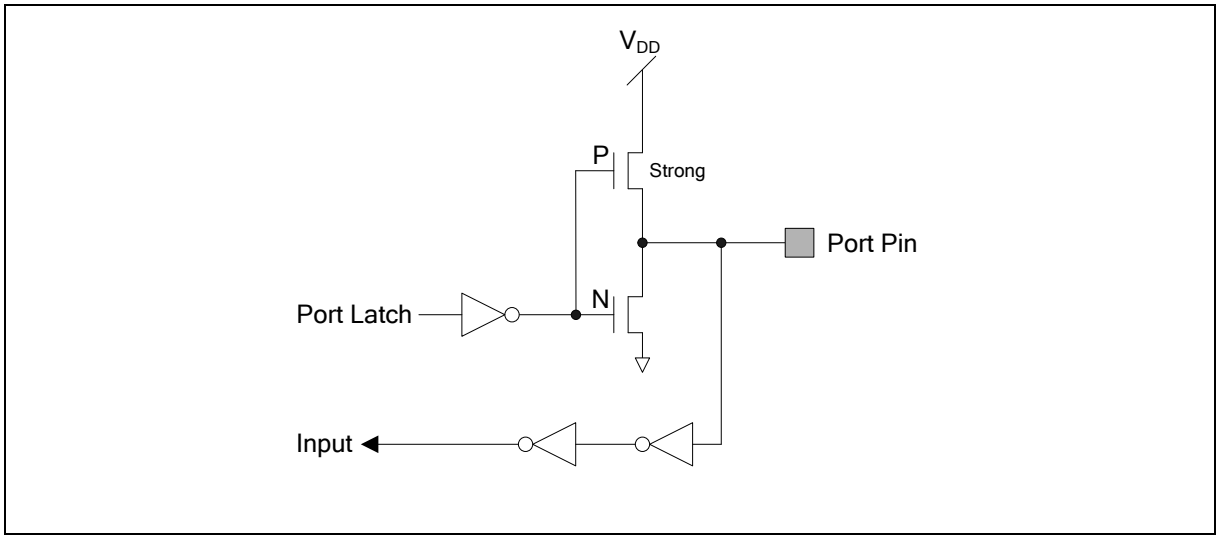


图 6.4-2 推挽模式结构

仅输入模式

仅输入模式提供真实的高阻输入路径。虽然准双向模式也可以作为输入引脚，但是它需要相对较强的输入源。仅输入模式有利于降低功耗，因为准双向模式下逻辑0输入始终会从V_{DD}消耗电流。用户需要注意的是，仅输入模式应该由外部设备或电阻提供一个确定的电平。悬空的引脚将引起漏电流，尤其是在掉电模式下。

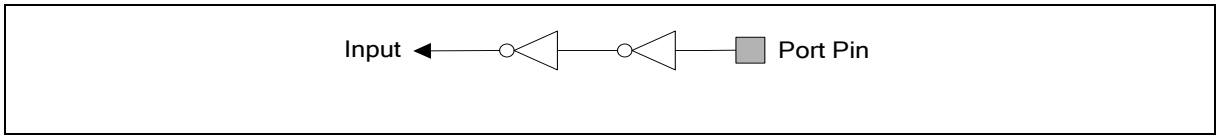


图 6.4-3 仅输入模式结构

开漏模式

开漏模式关闭所有上拉晶体管，仅当端口锁存器为逻辑 0 时驱动端口引脚的下拉电阻。当端口锁存器为逻辑1时，它的行为就和仅输入模式一样。要用作输出引脚(通常为I^2C线路)，开漏引脚应添加一个外部上拉电阻，通常是连一个电阻到V_{DD}。用户需要注意的是，开漏模式端口锁存器为逻辑1的时候，应该由外部设备或电阻提供一个确定的电平。悬空的引脚将引起漏电流，尤其是在掉电模式下。

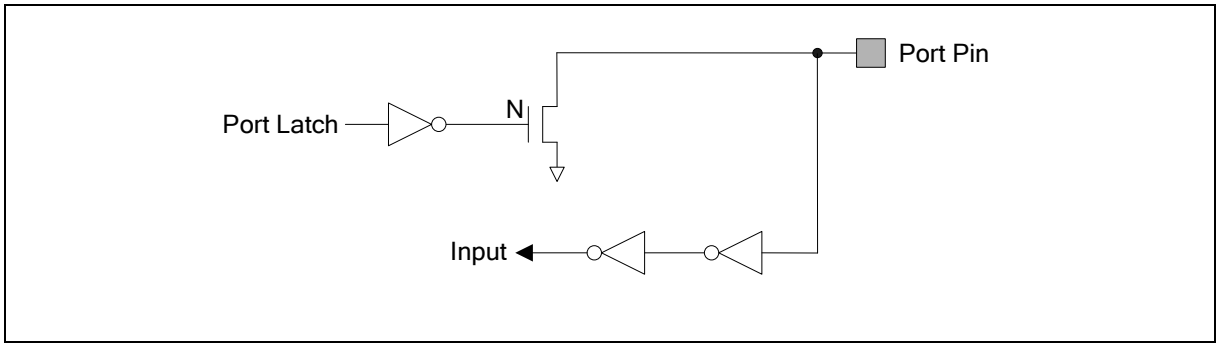


图 6.4-4 开漏模式结构

6.4.1.2 外部中断引脚 (INT0 和 INT1)

外部中断INT0 和 INT1可用作中断源，它们可以通过IT0(TCON.0)和IT1(TCON.2)来选择边沿触发或电平触发。 IE0 (TCON.1) 和 IE1 (TCON.3) 位是用于检查中断生成的标志位。在边沿触发模式下，每个系统时钟周期都会采样 INT0 或 INT1的输入状态。如果在一个周期中采样是高电平，然后在下一个周期中为低电平，那么这个高到低的电平转换将会被检测到，并设置中断请求标志 IE0 或IE1。由于每个系统时钟周期都会采样外部中断，因此高电平或低电平至少保持一个系统时钟周期。调用中断服务程序时， IE0 IE1会自动清零。如果选择电平触发模式，则必须将引脚保持低电平，直到中断服务完成。进入中断服务程序时， IE0和IE1不会被硬件清除。在电平触发模式下，IE0和IE1的值会跟随 INT0 和 INT1 引脚的反相值。如果中断服务程序结束后引脚依然保持低电平，处理器将响应来自同一中断源的另一个中断请求。INT0 和 INT1 均可将设备从掉电模式唤醒。

TCON – 定时器 0和1控制

寄存器	特殊功能寄存器地址	复位值
TCON	88H, 所有SFR页, 可位寻址	0000_0000b

7	6	5	4	3	2	1	0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**
读/写	读/写	读/写	读/写	可读(电平) 读/写(边沿)	读/写	可读(电平) 读/写(边沿)	读/写

位	名称	描述
3	**IE1**	**外部中断1边沿标志位** 如果 IT1 =1(下降沿触发), 当检测到下降沿时此标志位将被硬件置位。此位将保持置1直到软件清零或在其中断服务程序开始时被硬件清零。 如果 IT1 =0(低电平触发), 此标志位是INT1输入信号逻辑电平的反相。软件无法控制该标志位。
2	**IT1**	**外部中断1类型选择** 此位选择INT1的中断触发类型。 0 = INT1 是低电平触发. 1 = INT1 是下降沿触发.
1	**IE0**	**外部中断0边沿标志** 如果 IT0 =1(下降沿触发), 当检测到下降沿时此标志位将被硬件置位。此位将保持置1直到软件清零或在其中断服务程序开始时被硬件清零。 如果 IT0 =0(低电平触发), 此标志位是INT0输入信号逻辑电平的反相。软件无法控制该标志位。
0	**IT0**	**外部中断0类型选择** 此位选择INT0的中断触发类型。 0 = INT0 是低电平触发 1 = INT0 是下降沿触发

6.4.1.3 *引脚中断(PIT)*

CM1003系列为每个I/O引脚提供引脚中断输入，以便在使用按钮或键盘时检测引脚状态。通过I/O端口共享可以配置最多8个通道的引脚中断检测。当键盘或按键的任何键被按下时，产生边沿或电平触发事件，会产生引脚中断。引脚中断可用于将CPU从空闲或掉电模式唤醒。

每个引脚中断通道均可通过 PIPEN和PINEN寄存器独立使能和控制极性。PICON选择引脚中断在哪个端口有效，还设置引脚中断的触发方式：电平或边沿检测。每个通道也有自己的中断标志位。PIF寄存器中总共有八个引脚中断标志位，进入中断服务程序后轮询这些标志位来确定中断发生在哪个引脚上。PIF寄存器中的所有标志位均由硬件置位，应由软件清除。

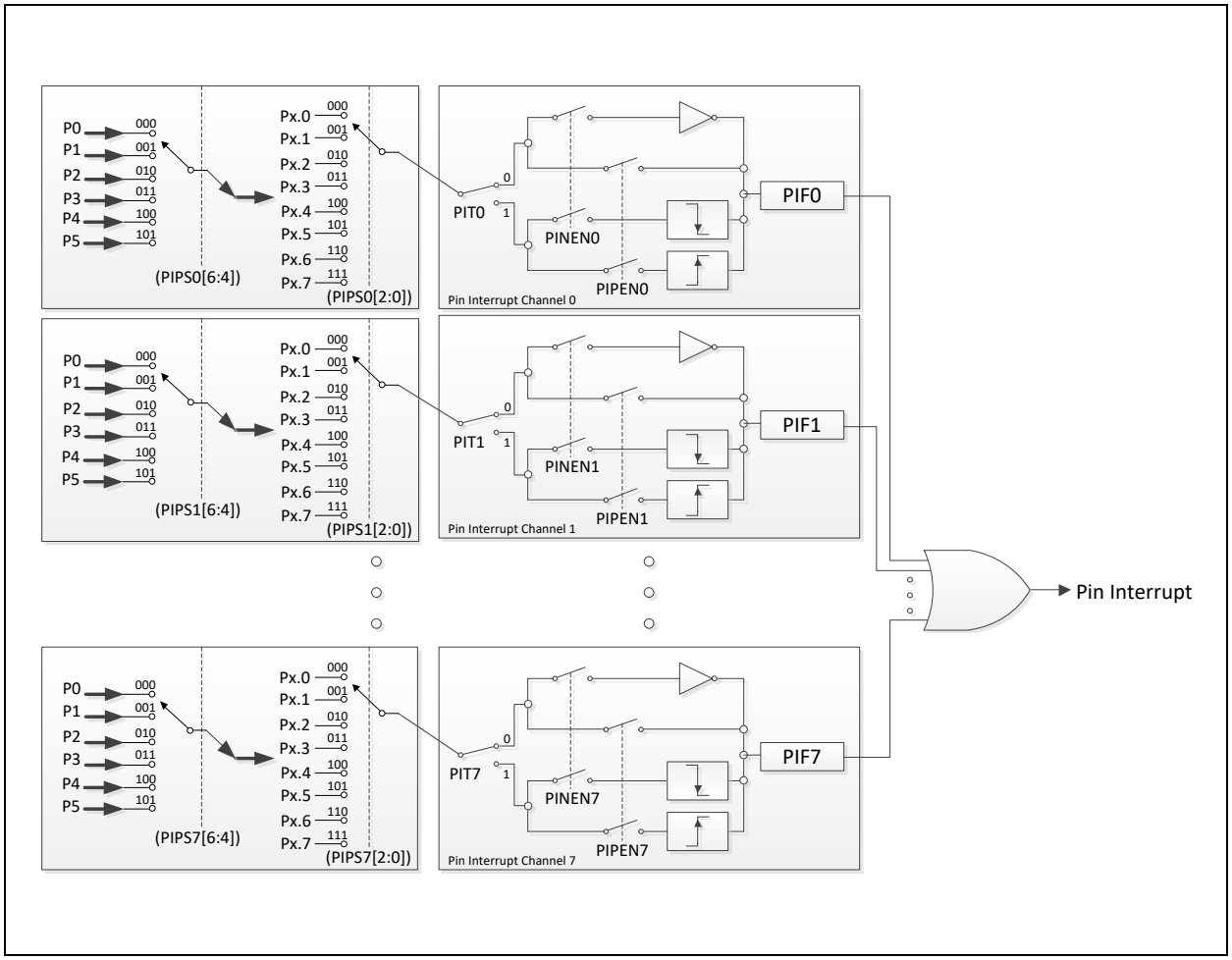


图 6.4-5 引脚接口框图

引脚中断通常用于检测来自键盘或按键等外设的边沿瞬变。在空闲状态下，系统会优先进入掉电模式以最大限度地降低功耗，并等待事件触发。引脚中断可以将设备从掉电模式唤醒。

6.4.2 寄存器描述

CM1003 系列拥有丰富的 I/O 控制寄存器，可为各种应用提供灵活性。与 I/O 端口相关的特殊功能寄存器 (SFR) 可分为四类：输入和输出控制、输出模式控制、输入类型和灌电流控制以及输出斜率控制。所有 SFR 列表如下。

6.4.2.1 输入和输出数据控制

这些寄存器是 I/O 输入和输出数据缓冲区。读取操作可获取 I/O 输入数据。写入操作可强制输出数据。所有这些寄存器都是可位寻址的。

Pn – 端口 (可位寻址)

寄存器	特殊功能寄存器地址	复位值
P0	80H, 所有SFR页, 可位寻址	1111_1111 b
P1	90H, 所有SFR页, 可位寻址	1111_1111 b
P2	A0H, 所有SFR页, 可位寻址	0011_1111 b
P3	B0H, 所有SFR页, 可位寻址	1111_1111 b
P4	D8H, SFR页1, 可位寻址	1111_1111 b
P5	D8H, SFR页2, 可位寻址	1111_1111 b

7	6	5	4	3	2	1	0
Pn.7	**Pn.6**	**Pn.5**	**Pn.4**	**Pn.3**	**Pn.2**	**Pn.1**	**Pn.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**Pn[7:0]**	**端口n** 端口n是最大8 位通用 I/O 端口。

6.4.2.2 GPIO 模式控制

这些寄存器控制 GPIO 模式，可在四种模式之间配置：仅输入、准双向、推挽或开漏。每个引脚均可单独配置。

复位后，所有 GPIO 默认设置为仅输入模式。

PnM1.X	PnM2.X	I/O类型
0	0	准双向
0	1	推挽
1	0	仅输入(高阻)
1	1	开漏

PnM1 – 端口n模式选择1

寄存器	特殊功能寄存器地址	复位值
P0M1	B1H, SFR页0	1111_1111 b
P1M1	B3H, SFR页0	1111_1111 b
P2M1	89H, SFR页2	0011_1111 b
P3M1	ACH, SFR页0	1111_1111 b
P4M1	89H, SFR页1	1111_1111 b
P5M1	99H, SFR页1	1111_1111 b

7	6	5	4	3	2	1	0
PnM1.7	**PnM1.6**	**PnM1.5**	**PnM1.4**	**PnM1.3**	**PnM1.2**	**PnM1.1**	**PnM1.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PnM1[7:0]**	**端口n模式选择1** **注意:** PnM1和PnM2组合使用来确定端口每个引脚的I/O模式。参见表6. 4-1

PnM2 – 端口n模式选择 2

寄存器	特殊功能寄存器地址	复位值
P0M2	B2H, SFR页0	0000_0000 b
P1M2	B4H, SFR页0	0000_0000 b
P2M2	8AH, SFR页2	0000_0000 b
P3M2	ADH, SFR页0	0000_0000 b
P4M2	8AH, SFR页1	0000_0000 b
P5M2	9AH, SFR页1	0000_0000 b

7	6	5	4	3	2	1	0
PnM2.7	**PnM2.6**	**PnM2.5**	**PnM2.4**	**PnM2.3**	**PnM2.2**	**PnM2.1**	**PnM2.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PnM2[7:0]**	**端口n模式选择2** **注意:** PnM1和PnM2组合使用来确定端口每个引脚的I/O模式。参见表6. 4-1 .

6.4.2.3 输入类型选择

每个 I/O 引脚均可单独配置为 TTL 输入或施密特触发输入。请注意，所有 PxS 寄存器均可通过将 SFR 页切换到第 1 页来访问。

PnS – 端口n施密特触发输入

寄存器	特殊功能寄存器地址	复位值
P0S	B1H, SFR页1	0000_0000 b
P1S	B3H, SFR页1	0000_0000 b
P2S	8CH, SFR页2	0000_0000 b
P3S	ACH, SFR页1	0000_0000 b
P4S	8CH, SFR页1	0000_0000 b
P5S	9CH, SFR页1	0000_0000 b

7	6	5	4	3	2	1	0
PnS.7	**PnS.6**	**PnS.5**	**PnS.4**	**PnS.3**	**PnS.2**	**PnS.1**	**PnS.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PnS[7:0]**	**Pn施密特触发输入** 0 = P0.n的TTL电平输入。 1 = P0.n的施密特触发输入。 **注：**每个 I/O 引脚均可单独配置为 TTL 输入或施密特触发输入。请注意，所有 PxS 寄存器均可通过将 SFR 页面切换到第 1 页来访问。

6.4.2.4 输出速率控制

每个 I/O 引脚的电压转换速率均可单独配置。默认情况下，每个引脚处于正常速率模式。用户可以设置每个控制寄存器位，以使能相应 I/O 引脚的高速斜率。请注意，所有 PxSR 寄存器均可通过将 SFR 页面切换到第 1 页来访问。

PnSR – 端口n 速率控制

寄存器	特殊功能寄存器地址	复位值
P0SR	B2H, SFR页1	0000_0000 b
P1SR	B4H, SFR页1	0000_0000 b
P2SR	8BH, SFR页2	0000_0000 b
P3SR	ADH, SFR页1	0000_0000 b
P4SR	8BH, SFR页1	0000_0000 b
P5SR	9BH, SFR页1	0000_0000 b

7	6	5	4	3	2	1	0
PnSR.7	**PnSR.6**	**PnSR.5**	**PnSR.4**	**PnSR.3**	**PnSR.2**	**PnSR.1**	**PnSR.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PnSR[7:0]**	**Pn速率** 0 = Pn.x 正常输出速率。 1 = Pn.x 高速输出速率。

PICON – 引脚中断控制

寄存器	特殊功能寄存器地址	复位值
PICON	E9H, 页0,	0011 _0100 b

7	6	5	4	3	2	1	0
PIT7	PIT6	PIT5	PIT4	PIT3	PIT2	PIT1	PIT0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	PIT7	**引脚中断通道7的类型选择** 此位选择引脚中断通道7的触发类型。 0 =电平触发。 1 =边沿触发。
6	PIT6	**引脚中断通道6的类型选择** 此位选择引脚中断通道6的触发类型。 0 =电平触发。 1 =边沿触发。
5	PIT5	**引脚中断通道5 的类型选择** 此位选择引脚中断通道5的触发类型。 0 =电平触发。 1 =边沿触发。
4	PIT4	**引脚中断通道4的类型选择** 此位选择引脚中断通道4的触发类型。 0 =电平触发。 1 =边沿触发。
3	PIT3	**引脚中断通道3 的类型选择** 此位选择引脚中断通道3的触发类型。 0 =电平触发。 1 =边沿触发。
2	PIT2	**引脚中断通道2的类型选择** 此位选择引脚中断通道2的触发类型。 0 =电平触发。 1 =边沿触发。
1	PIT1	**引脚中断通道1的类型选择** 此位选择引脚中断通道1的触发类型。 0 =电平触发。 1 =边沿触发。
0	PIT0	**引脚中断通道0的类型选择** 此位选择引脚中断通道0的触发类型。 0 =电平触发。 1 =边沿触发。

PINEN – 引脚中断负极性使能

寄存器	特殊功能寄存器地址	复位值
PINEN	EAH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PINEN7	PINEN6	PINEN5	PINEN4	PINEN3	PINEN2	PINEN1	PINEN0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
n	**PINENn**	**通道n的引脚中断负极性使能** 此位使能低电平/下降沿触发通道n的引脚中断。电平触发还是边沿触发，取决于 PICON 中的每个控制位 PITn的设定。 0 = 低电平/下降沿侦测禁用。 1 = 低电平/下降沿侦测使能。

PIPEN – 引脚中断正极性使能

寄存器	特殊功能寄存器地址	复位值
PIPEN	EBH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PIPEN7	PIPEN6	PIPEN5	PIPEN4	PIPEN3	PIPEN2	PIPEN1	PIPEN0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
n	**PIPENn**	**通道n的引脚中断正极性使能** 此位使能高电平/上升沿触发通道n的引脚中断。电平触发还是边沿触发，取决于 PICON 中的每个控制位 PITn的设定。 0 = 高电平/上升沿侦测禁用。 1 = 高电平/上升沿侦测使能。

PIF – 引脚中断标志

寄存器	特殊功能寄存器地址	复位值
PIF	ECH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)	只读(电平) 读/写(边沿)

位	名称	描述
n	PIFn	**引脚中断通道 n 标志** 如果选择边沿触发，则当引脚中断通道 n 检测到已启用的边沿触发时，硬件将置位此标志。此标志应通过软件清除。 如果选择电平触发，则此标志跟随引脚中断通道 n 上输入信号的逻辑电平反转。软件无法控制它。

PIPSn – 引脚中断控制

寄存器	特殊功能寄存器地址	复位值
PIPS7	F7H, 页2	0000_0000 b
PIPS6	FFH, 页2	0000_0000 b
PIPS5	FEH, 页2	0000_0000 b
PIPS4	FDH, 页2	0000_0000 b
PIPS3	FCH, 页2	0000_0000 b
PIPS2	FBH, 页2	0000_0000 b
PIPS1	FAH, 页2	0000_0000 b
PIPS0	F9H, 页2	0000_0000 b

7	6	5	4	3	2	1	0
-	PSEL[2:0]			-	BSEL[2:0]		
-	读/写			-	读/写		

位	名称	描述
6:4	**PSEL[2:0]**	**引脚中断通道端口选择** 000 = P0 端口. 001 = P1端口. 010 = P2端口. 011 = P3端口. 100 = P4端口. 101 = P5端口. 110 = 保留. 111 = 保留.
2:0	**BSEL[2:0]**	**引脚中断通道位选择** 000 = Pn.0. 001 = Pn.1 010 =.Pn.2 011 = Pn.3. 100 = Pn.4. 101 = Pn.5. 110 = Pn.6. 111 = Pn.7. n为端口编号, 由PSEL[2:0]选择.

6.5 定时器

6.5.1 定时器/计数器0 和 1

6.5.1.1 概述

CM1003上的定时器/计数器0和1是两个16位定时器/计数器。它们各自由两个8位的寄存器组成16计数寄存器。对于定时器/计数器 0，TH0为高8位寄存器、TL0为低8位寄存器。同样定时器/计数器1也有两个8位寄存器，TH1和TL1。TCON和TMOD可以配置定时器/计数器0和1的模式。

定时器或计数器功能由TMOD中的C/$\overline{T}$位来选择。每个定时器/计数器都有自己的选择位。TMOD.2选择定时器/计数器0的功能，TMOD.6 选择定时器/计数器1的功能。

当配置为“定时器”时，定时器将对系统时钟周期计数。定时器0通过设置T0M(CKCON.3)位，定时器1通过设置T1M(CKCON.4)位，来选择定时器时钟是标准的8051功能时钟(系统时钟F_{SYS}的1/12分频)还是增强型时钟(直接是系统时钟)。在“计数器”模式下，计数寄存器在外部输入引脚T0的下降沿加一。如果采样值在一个时钟周期为高，在下一个时钟周期为低，那么T0或T1引脚就会识别出有效的1到0跳变。

定时器0和1可以配置为在发生定时器溢出时自动切换输出。同一个引脚可以用于T0和T1的计数输入也可以用于定时器的切换输出，这个功能通过CKCON寄存器的T0OE和T1OE控制位来分别使能定时器0和定时器1。打开此模式时，在第一次定时器溢出之前，端口输出将为逻辑1。为确保此模式功能正常工作，C/$\overline{T}$位应该被清除并且选择系统时钟作为定时器的时钟源。

请注意，TH0(TH1)和TL0(TL1)是单独访问的。强烈建议用户在模式0或1下，在读/写TH0(TH1)和TL0(TL1)之前，通过清除TR0(TR1) 暂时停止定时器，自由运行的读写可能导致不可预测的结果。

6.5.1.2 模式 0 (13位定时器)

在模式0, 定时器/计数器是13位的计数器。该13位的计数器由TH0(TH1)和TL0(TL1)的低五位组成。TL0(TL1)的高三位被忽略。当TR0(TR1)置位，且GATE是0或$\overline{INT0}$($\overline{INT1}$)是1时，定时器/计数器使能。GATE设置为1允许定时器用来计算外部输入引脚$\overline{INT0}$($\overline{INT1}$)上的脉冲宽度。当13位的定时器计数值从1FFFH变为0000H后，定时器溢出标志TF0(TF1)被置位，如果中断使能则将产生中断。

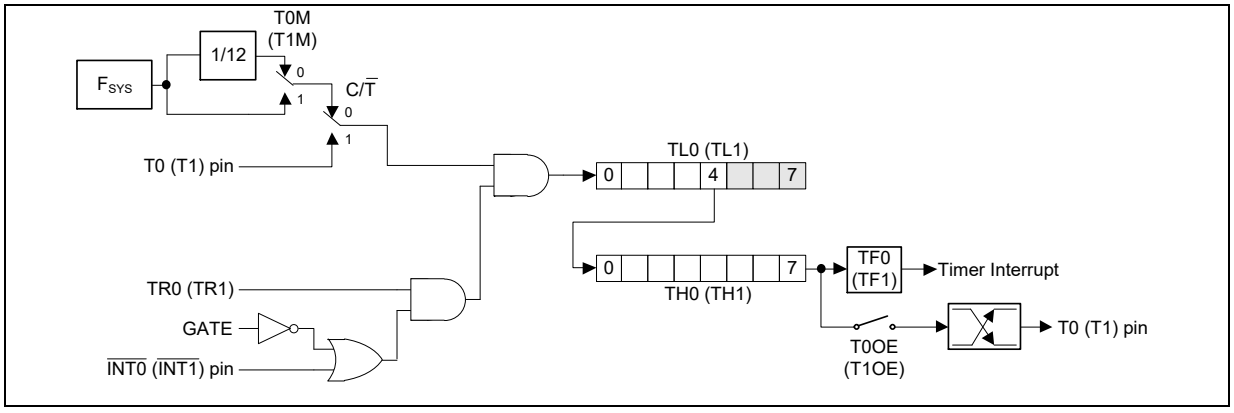


图 6.5-1 定时器/计数器 0 和 1 处于模式 0

6.5.1.3 *模式 1 (16位定时器)*

模式1与模式0类似，只是计数寄存器完全用作16 位计数器。计数从FFFFH变到0000H时发生翻转。相应的定时器溢出标志TF0(TF1)被置位，如果中断使能则将产生中断。

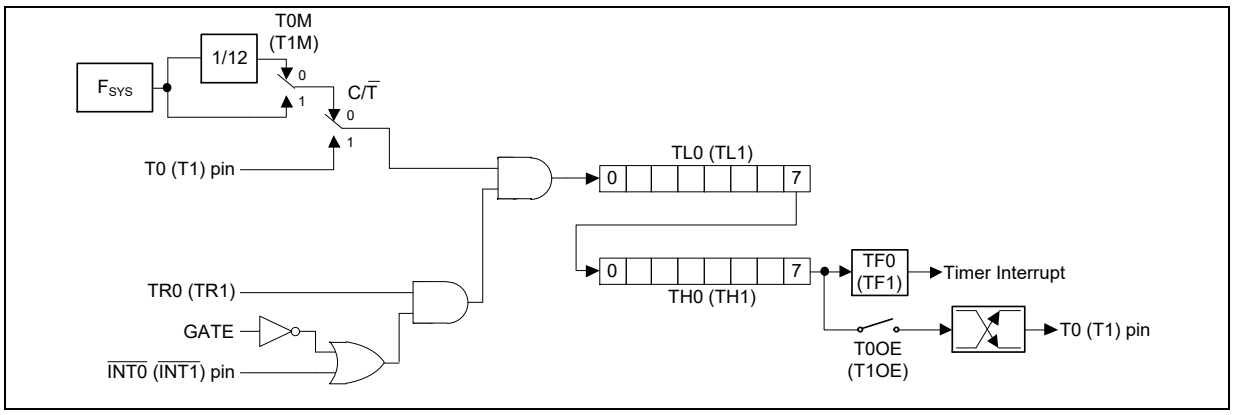


图 6.5-2 定时器/计数器 0 和 1 处于模式 1

6.5.1.4 *模式 2 (8位自动重载定时器)*

模式2下定时器/计数器为自动重载模式。此模式下TL0(TL1)用作 8 位计数寄存器，而TH0(TH1)保存重载值。当TL0(TL1)溢出后，TCON中的TF0(TF1)标志位被置位，并将TH0(TH1)中内容重载至TL0(TL1)，然后继续计数过程。重载操作不会改变TH0(TH1)寄存器的内容。此功能最适合UART波特率发生器，因为它无需持续的软件干预即可运行。请注意，只有定时器1可以用作UART的波特率源。通过将TR0(TR1)设置为1并正确设置GATE和INT0(INT1)引脚，可以启用计数。GATE和INT0(INT1)引脚的设置与模式0和1相同。

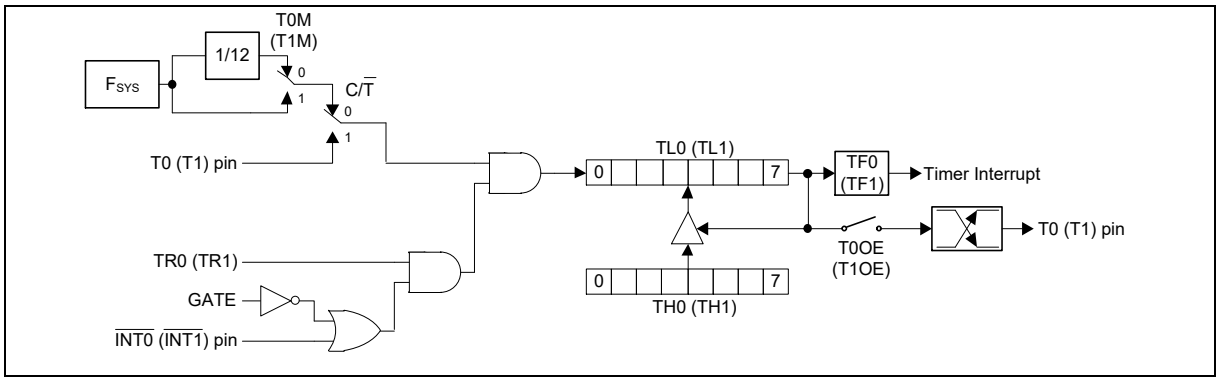


图 6.5-3 定时器/计数器 0 和 1 处于模式 2

6.5.1.5 模式 3 (两组独立的 8位定时器)

模式3对定时器0和定时器1的操作方式有所不同。对于定时器/计数器1来说模式3只是冻结计数器；而对定时器/计数器0来说，模式3下TL0和TH0配置为2个独立的8位计数寄存器。TL0使用定时器/计数器0控制位C/$\overline{T}$、GATE、TR0、$\overline{INT0}$和 TF0。TL0也可以用来对引脚T0上的1到0跳变计数，具体由C/$\overline{T}$(TMOD.2)来决定。TH0 被强制用作时钟周期计数器，并从定时器/计数器 1 接管 TR1 和 TF1 的使用。如果需要额外的8位定时器时可以使用模式3。当定时器/计数器0配置为模式3时，定时器/计数器1可以通过配置其进入或离开模式3的方式来打开或关闭定时器/计数器 1。定时器1依然可以工作在模式0、1、2下，但它的灵活性受到限制。它不再控制其溢出标志位TF1和使能位TR1。此时定时器1仍可用作定时器/计数器，并依然可以使用GATE、$\overline{INT1}$引脚和T1M。它可用作串行端口的波特率生成器或其他不需要中断的应用。

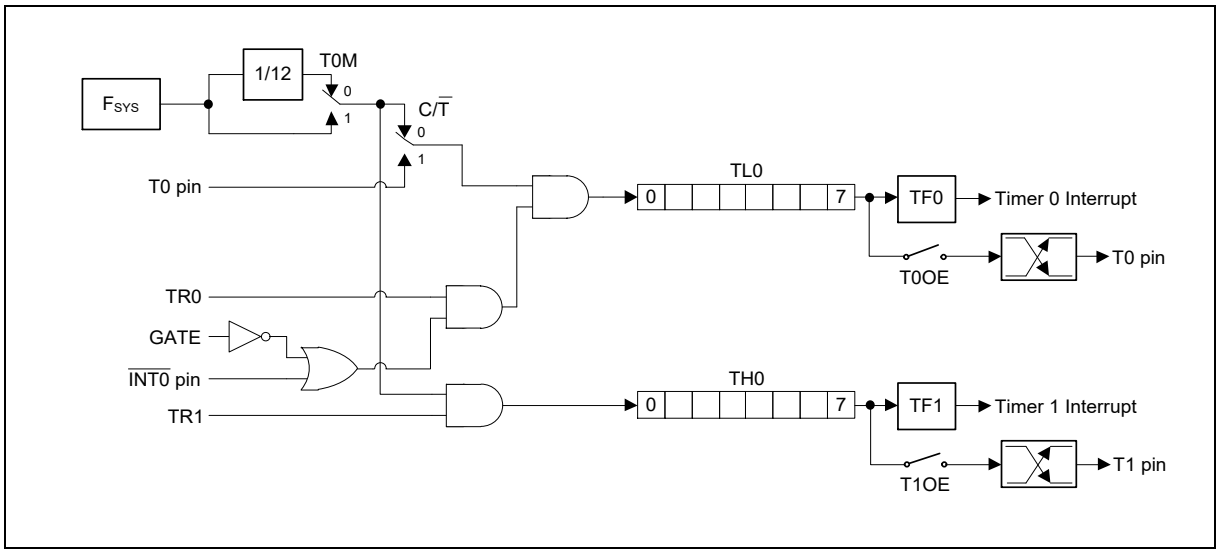


图 6.5-4 定时器/计数器 0 和 1 处于模式 3

6.5.1.6 I/O端口的控制寄存器

TMOD – 定时器 0和1模式

寄存器	特殊功能寄存器地址	复位值
TMOD	89H, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
GATE	**C/$\overline{T}$**	**M1**	**M0**	**GATE**	**C/$\overline{T}$**	**M1**	**M0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	GATE	**定时器 1门控制** 0 =无论$\overline{INT1}$的逻辑电平如何，定时器 1 都会在 TR1 为 1 时进行计时。 1 =仅当TR1为1且$\overline{INT1}$为逻辑1时，定时器 1 才会进行计时。
6	C/$\overline{T}$	**定时器1计数器/定时器 选择** 0 = 定时器1随内部时钟而递增 1 = 定时器1随外部引脚T1的下降沿递增
5	M1	**定时器1模式选择** M1 M0 定时器1模式 0 0 模式 0: 13位定时器/计数器 0 1 模式1: 16位定时器/计数器 1 0 模式 2: 8位定时器/计数器，从TH1自动重载 1 1 模式 3: 定时器1暂停
4	M0	
3	GATE	**定时器0 门控制** 0 = 无论$\overline{INT0}$的逻辑电平如何，当 TR0 为 1 时，定时器 0 都会开始计时。 1 = 仅当TR0为1且$\overline{INT0}$为逻辑1时，定时器 0 才会开始计时。
2	C/$\overline{T}$	**定时器0 计数器/定时器 选择** 0 = 定时器0 随内部时钟而递增 1 = 定时器0 随外部引脚T0的下降沿递增
1	M1	**定时器0 模式选择** M1 M0 定时器0模式 0 0 模式 0: 13位定时器/计数器 0 1 模式 1: 16位定时器/计数器 1 0 模式 2: 8位定时器/计数器，从TH0自动重载 1 1 模式 3: TL0 作为一个 8位 定时器/计数器， TH0 作为一个 8位 定时器
0	M0	

TCON – 定时器 0和1控制

寄存器	特殊功能寄存器地址	复位值
TCON	88H, 所有SFR页, 可位寻址	0000_0000b

7	6	5	4	3	2	1	0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**
读/写	读/写	读/写	读/写	可读(电平) 读/写(边沿)	读/写	可读(电平) 读/写(边沿)	读/写

位	名称	描述
7	**TF1**	**定时器1溢出标志** 当定时器1溢出时，此位置位。程序执行定时器1中断服务程序时，硬件自动清零此位。此位也可由软件置位或清零。
6	**TR1**	**定时器1运行控制** 0 = 定时器1禁用。清除此位将暂停定时器 1，当前计数值将保存在 TH1 和 TL1 中。 1 = 定时器1使能。
5	**TF0**	**定时器0 溢出标志** 当定时器0溢出时，此位置位。程序执行定时器0中断服务程序时，硬件自动清除此位。此位也可由软件置位或清除。
4	**TR0**	**定时器0运行控制** 0 = 定时器0禁用。清除此位将暂停定时器 0，当前计数值将保存在 TH0 和 TL0 中。 1 = 定时器0使能。
3	**IE1**	**外部中断1边沿标志位** 如果 IT1 =1(下降沿触发), 当检测到下降沿时此标志位将被硬件置位。此位将保持置1直到软件清零或在其中断服务程序开始时被硬件清零。 如果 IT1 =0(低电平触发), 此标志位是$\overline{\text{INT1}}$输入信号逻辑电平的反相。软件无法控制该标志位。
2	**IT1**	**外部中断1类型选择** 此位选择$\overline{\text{INT1}}$的中断触发类型。 0 = $\overline{\text{INT1}}$ 是低电平触发. 1 = $\overline{\text{INT1}}$ 是下降沿触发.
1	**IE0**	**外部中断0边沿标志** 如果 IT0 =1(下降沿触发), 当检测到下降沿时此标志位将被硬件置位。此位将保持置1直到软件清零或在其中断服务程序开始时被硬件清零。 如果 IT0 =0(低电平触发), 此标志位是$\overline{\text{INT0}}$输入信号逻辑电平的反相。软件无法控制该标志位。

位	名称	描述
0	**IT0**	**外部中断0类型选择** 此位选择$\overline{\text{INT0}}$的中断触发类型。 0 = $\overline{\text{INT0}}$ 是低电平触发 1 = $\overline{\text{INT0}}$ 是下降沿触发

TL0 – 定时器0低字节

寄存器	特殊功能寄存器地址	复位值
TL0	8AH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
TL0[7:0]							
读/写							

位	名称	描述
7:0	**TL0[7:0]**	**定时器0低字节** TL0 寄存器是定时器 0 的 16 位计数寄存器的低字节。

TH0 – 定时器0高字节

寄存器	特殊功能寄存器地址	复位值
TH0	8CH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
TH0[7:0]							
读/写							

位	名称	描述
7:0	**TH0[7:0]**	**定时器0高字节** TH0 寄存器是定时器 0 的 16 位计数寄存器的高字节。

TL1 – 定时器1低字节

寄存器	特殊功能寄存器地址	复位值
TL1	8BH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
TL1[7:0]							
读/写							

位	名称	描述
7:0	**TL1[7:0]**	**定时器1低字节** TL1 寄存器是定时器1 的 16 位计数寄存器的低字节。

TH1 – 定时器1高字节

寄存器	特殊功能寄存器地址	复位值
TH1	8DH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
TH1[7:0]							
读/写							

位	名称	描述
7:0	TH1[7:0]	**定时器1高字节** TH1 寄存器是定时器 1的 16 位计数寄存器的高字节。

CKCON – 时钟控制

寄存器	特殊功能寄存器地址	复位值
CKCON	8EH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	PWMCKS	-	T1M	T0M	-	CLOEN	-
-	读/写	-	读/写	读/写	-	读/写	-

位	名称	描述
4	T1M	**定时器1时钟模式选择** 0 = 定时器1的时钟源是系统时钟的12分频，它保持标准8051的兼容性 1 = 定时器1的时钟源直接是系统时钟
3	T0M	**定时器0时钟模式选择** 0 = 定时器0的时钟源是系统时钟的12分频，它保持标准8051的兼容性 1 = 定时器0的时钟源直接是系统时钟

6.5.2 定时器2和输入捕获

6.5.2.1 概述

定时器2是一个16位的向上计数器，由高8位寄存器(TH2)和低8位寄存器(TL2)组成。通过配置寄存器RCMP2H和RCMP2L，定时器2可设置CM/$\overline{RL2}$(T2CON.0)，选择工作在比较模式还是自动重载模式下。定时器2具有3通道输入捕获模块，可用于测量输入脉冲宽度或周期。通道捕获结果分别存放在寄存器C0H和C0L,C1H和C1L,C2H和C2L中。定时器2的时钟来自系统时钟经过分频器8级的预分频，可适用于广泛的应用需求。当TR2(T2CON.2)置 1，定时器使能；TR2置0时，定时器禁用。以下寄存器与定时器 2 功能相关。以下寄存器与定时器 2 功能相关。

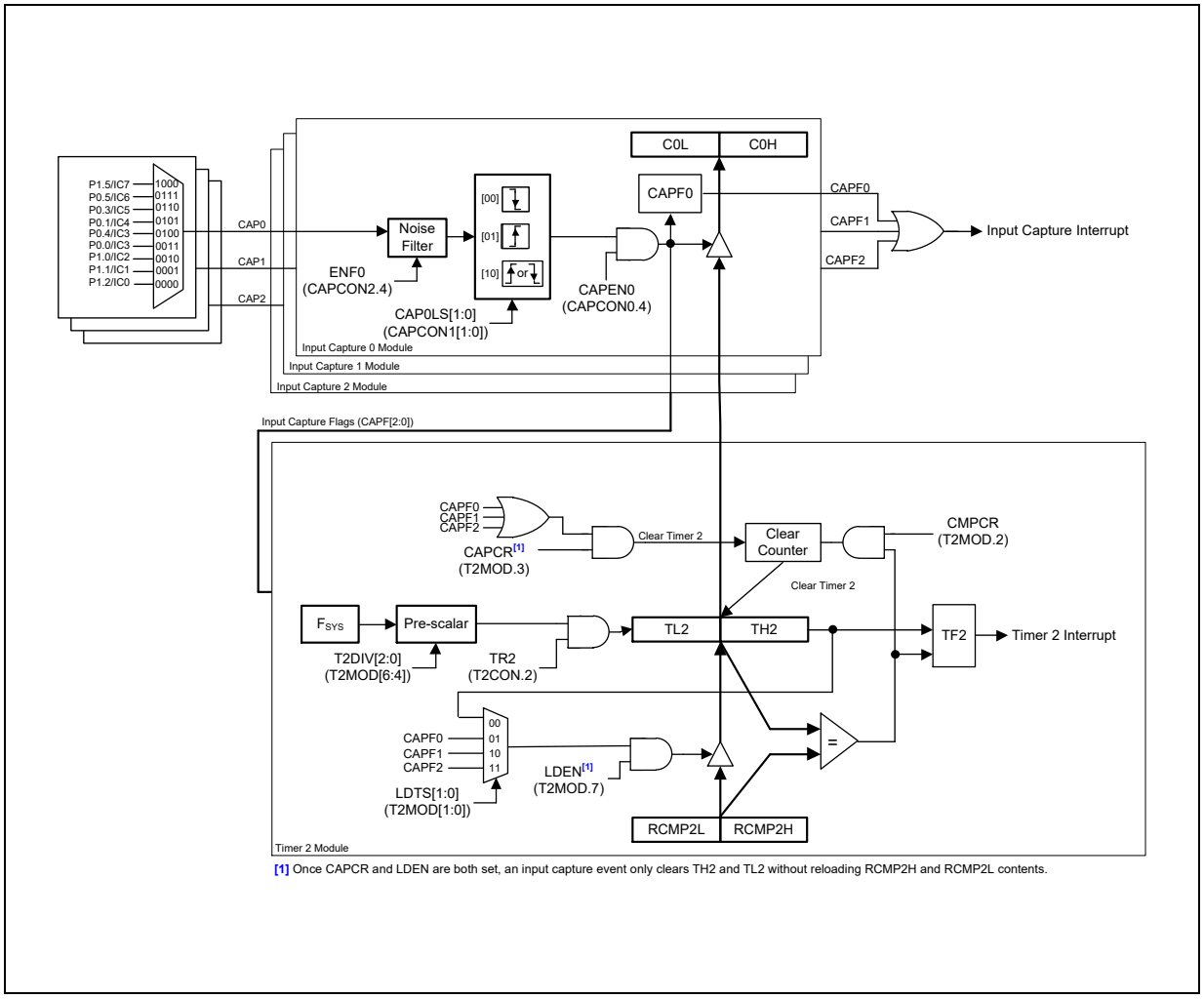


图 6.5-5 定时器 2 框图

6.5.2.2 *自动重载模式*

当CM/$\overline{RL2}$清0，定时器2配置为自动重装载模式。在此模式下，RCMP2H及RCMP2L保存重载的数值。当LDEN置位后，一旦发生自动重载事件，RCMP2H和RCMP2L中的内容将写入TH2和TL2中。该事件可以是定时器2溢出，也可以是任何启用的输入捕获通道上的触发事件之一，具体取决于LDTS[1:0](T2MOD[1:0])的配置。请注意，一旦CAPCR(T2MOD.3)被置位，输入捕获事件只会清除TH2和TL2，而不会重载RCMP2H和RCMP2L的内容。

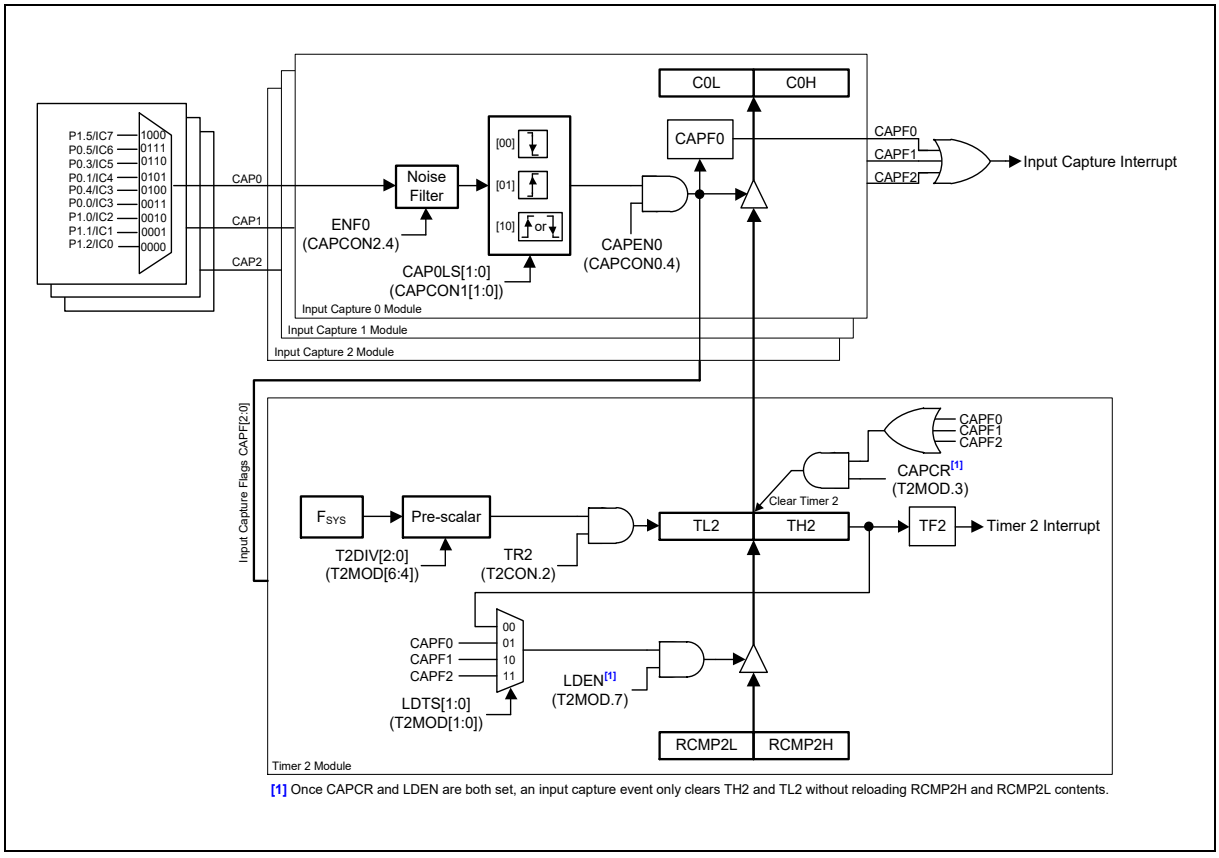


图 6.5-6 定时器 2 自动重载模式和输入捕获模组功能框图

6.5.2.3 比较模式

定时器2也可以通过置位CM/$\overline{RL2}$配置为比较模式。在此模式下，RCMP2H和RCMP2L寄存器用作比较值寄存器。当定时器2向上计数， TH2和TL2与RCMP2H和RCMP2L匹配，TF2(T2CON.7)将会由硬件置位，用以标示发生了比较匹配事件。

设置CMPCR(T2MOD.2) 会使硬件在发生比较匹配后自动把定时器2计数器清零为 0000H。

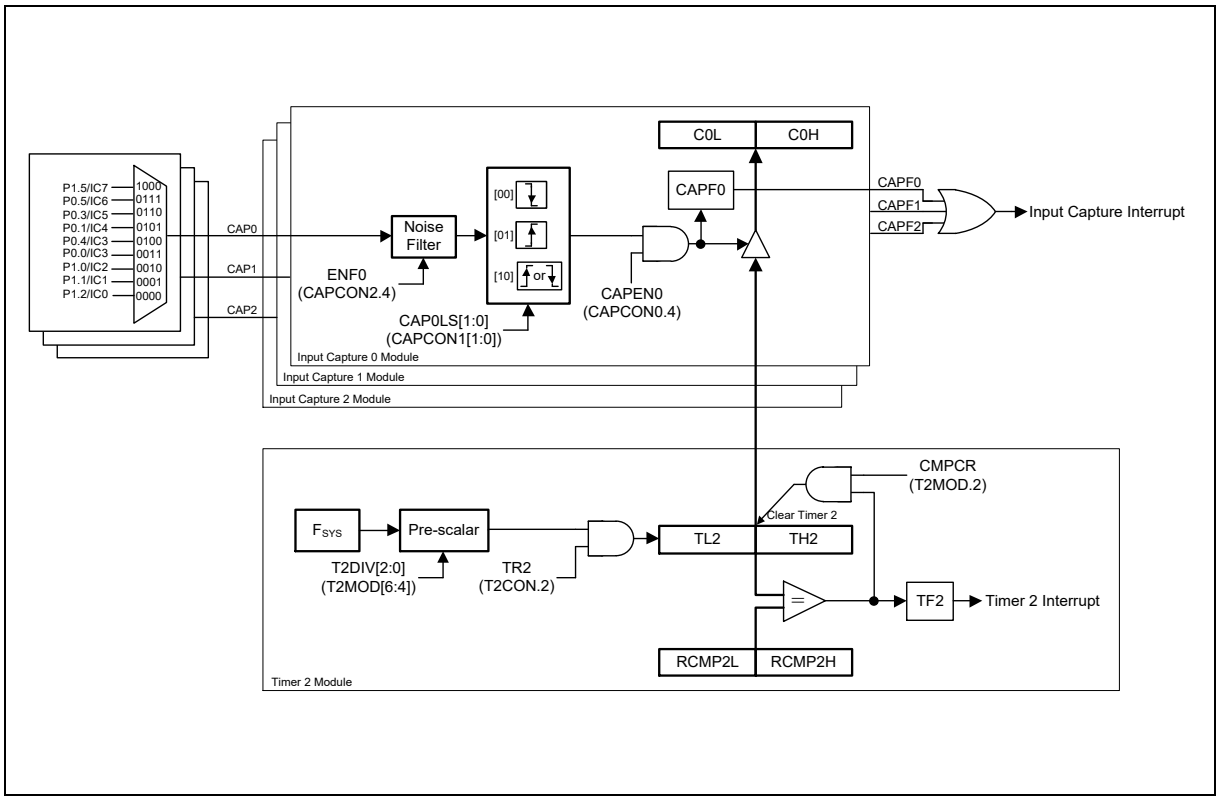


图 6.5-7 定时器 2 比较模式和输入捕获模组功能框图

6.5.2.4 输入捕获模块

输入捕获模块搭配定时器2实现输入捕获功能。输入捕获模块通过寄存器CAPCON0~2配置，支持 3 通道输入(CAP0、CAP1 和 CAP2)。每个输入通道的噪声滤波器可通过设置ENF0~2(CAPCON2[6:4])使能，过滤小于四个系统时钟周期的输入毛刺讯号。输入捕获通道有自己独立的边沿检测器，但共用唯一的定时器 2。每个触发边沿检测器可由寄存器CAPCON1的相关位独立配置，支持上升沿捕获、下降沿捕获或任意边沿捕获。每个输入捕获通道在使用前都必须设置各自的使能位CAPEN0~2(CAPCON0[6:4])。

当输入捕获通道使能且发生所选边沿触发时，定时器2的计数器TH2和TL2的内容将被捕获、传输并存储到捕获寄存器CnH和CnL中。边沿触发还会使硬件置位CAPFn(CAPCON0.n)。如果ECAP(EIE0.2)和EA位都已设置，将会产生中断。由于三组输入捕获标志位共享同一个中断向量，用户应检查CAPFn来确认哪个通道触发了输入捕获。这些标志位应由软件清除。

CAPCR(CAPCON2.3) 位有助于实现周期计算。置位CAPCR使得输入捕获边沿事件发生后，TH2和TL2的值被捕获， 硬件自动把定时器2清零为 0000H。这样可以避免常规软件写16位计数或者算法减法运算开销。

6.5.2.5 定时器2的控制寄存器

T2CON – 定时器2控制

寄存器	特殊功能寄存器地址	复位值
T2CON	C8H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
TF2	**-**	**-**	**-**	**-**	**TR2**	**-**	**CM/$\overline{RL2}$**
读/写	-	-	-	-	读/写	-	读/写

位	名称	描述
7	**TF2**	**定时器2溢出标志** 当定时器2溢出或者发生比较匹配时，此位置位。如果定时器2中断和全局中断都已使能，则置位此位将使CPU执行定时器 2的中断服务程序。此位不会通过硬件自动清除，应通过软件清除。
5:3	**-**	保留
2	**TR2**	**定时器2运行控制** 0 = 定时器2禁用。清除此位将暂停定时器2，当前计数将保存在TH2和TL2中。 1 = 定时器2使能。
1	**-**	保留
0	**CM/$\overline{RL2}$**	**定时器2比较或自动重载模式选择** 此位选择定时器2的工作模式。 0 = 自动重新加载模式。 1 = 比较模式。

T2MOD – 定时器2模式

寄存器	特殊功能寄存器地址	复位值
T2MOD	C9H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
LDEN	T2DIV[2:0]			CAPCR	CMPCR	LDTS[1:0]	
读/写	读/写			读/写	读/写	读/写	

位	名称	描述
7	LDEN	**自动重载使能** 0 = 将RCMP2H和RCMP2L重载至TH2和TL2功能禁用。 1 = 将RCMP2H和RCMP2L重载至TH2和TL2功能使能。
6:4	T2DIV[2:0]	**定时器2时钟分频** 000 = 定时器2时钟分频为1/1. 001 = 定时器2时钟分频为1/4. 010 = 定时器2时钟分频为1/16. 011 = 定时器2时钟分频为1/32. 100 = 定时器2时钟分频为1/64. 101 = 定时器2时钟分频为1/128. 110 = 定时器2时钟分频为1/256. 111 = 定时器2时钟分频为1/512.
3	CAPCR	**捕获自动清除** 此位仅当定时器2自动重载模式下有效。当发生捕获事件时，TH2和TL2计数器内数据被传输到RCMP2H 和RCMP2L后，该位使能硬件自动清除TH2和TL2这两个计数寄存器。 0 = 捕获事件发生，定时器2继续计数。 1 = 捕获事件发生，定时器2内数值被自动清0
2	CMPCR	**比较匹配自动清除** 此位仅当定时器2比较功能模式下有效。当发生比较匹配时，该位使能硬件自动清除TH2和TL2计数器值。 0 = 发生比较匹配时，定时器2继续计数。 1 = 发生比较匹配时，定时器2内数值被自动清0。
1:0	LDTS[1:0]	**自动重载触发选择** 这些位选择重载触发事件。 00 = 定时器2溢出时重载。 01 = 发生输入捕获 0 事件时重新加载。 10 = 发生输入捕获1 事件时重新加载。 11 = 发生输入捕获 2 事件时重新加载。

RCMP2L– 定时器2重载/比较低字节

寄存器	特殊功能寄存器地址	复位值
RCMP2L	CAH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
RCMP2L[7:0]							
读/写							

位	名称	描述
7:0	**RCMP2L[7:0]**	**定时器2重载/比较低字节** 当定时器2设定为比较模式，此寄存器存储比较值的低字节。 在自动重载模式下，此寄存器存储重载值的低字节。

RCMP2H – 定时器2重载/比较高字节

寄存器	特殊功能寄存器地址	复位值
RCMP2H	CBH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
RCMP2H[7:0]							
读/写							

位	名称	描述
7:0	**RCMP2H[7:0]**	**定时器2重载/比较高字节** 当定时器2设定为比较模式，此寄存器存储比较值的高字节。 在自动重载模式下，此寄存器存储重载值的高字节。

TL2 – 定时器2低字节

寄存器	特殊功能寄存器地址	复位值
TL2	CCH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
TL2[7:0]							
读/写							

位	名称	描述
7:0	TL2[7:0]	**定时器2低字节** TL2 寄存器为定时器2的16位计数寄存器的低字节。

TH2 – 定时器2高字节

寄存器	特殊功能寄存器地址	复位值
TH2	CDH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
TH2[7:0]							
读/写							

位	名称	描述
7:0	TH2[7:0]	**定时器2高字节** TL2 寄存器为定时器2的16位计数寄存器的高字节。

请注意，TH2和TL2是分开访问的，强烈建议用户在读取或写入TH2和TL2之前通过清除TR2位暂停定时器2。如果在运行中对寄存器直接读写，可能导致不可预测的结果。

CAPCON0 – 输入捕获控制 0

寄存器	特殊功能寄存器地址	复位值
CAPCON0	92H, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0
-	读/写	读/写	读/写	-	读/写	读/写	读/写

位	名称	描述
7	-	保留
6	CAPEN2	**输入捕获通道2使能** 0 = 输入捕获通道2禁止. 1 = 输入捕获通道2使能.
5	CAPEN1	**输入捕获通道1使能** 0 = 输入捕获通道1禁止. 1 = 输入捕获通道1使能.
4	CAPEN0	**输入捕获通道0使能** 0 = 输入捕获通道0禁止. 1 = 输入捕获通道0使能.
3	-	保留
2	CAPF2	**输入捕获通道 2标志** 如果输入捕获通道2探测到边沿触发事件发生，由硬件置位此位，此位应由软件清零。
1	CAPF1	**输入捕获通道1标志** 如果输入捕获通道1探测到边沿触发事件发生，由硬件置位此位，此位应由软件清零。
0	CAPF0	**输入捕获通道0标志** 如果输入捕获通道0探测到边沿触发事件发生，由硬件置位此位，此位应由软件清零。

CAPCON1 – 输入捕获控制 1

寄存器	特殊功能寄存器地址	复位值
CAPCON1	93H, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	-	CAP2LS[1:0]		CAP1LS[1:0]		CAP0LS[1:0]	
-	-	读/写		读/写		读/写	

位	名称	描述
7:6	-	保留
5:4	CAP2LS[1:0]	**输入捕获通道2 电平选择** 00 = 下降沿 01 = 上升沿 10 = 上升沿或下降沿 11 = 保留
3:2	CAP1LS[1:0]	**输入捕获通道1 电平选择** 00 = 下降沿 01 = 上升沿 10 = 上升沿或下降沿 11 = 保留
1:0	CAP0LS[1:0]	**输入捕获通道0 电平选择** 00 = 下降沿 01 = 上升沿 10 = 上升沿或下降沿 11 = 保留

CAPCON2 – 输入捕获控制 2

寄存器	特殊功能寄存器地址	复位值
CAPCON2	94H, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	ENF2	ENF1	ENF0	-	-	-	-
-	读/写	读/写	读/写	-	-	-	-

位	名称	描述
6	**ENF2**	**输入捕获通道2 噪声滤波器使能** 0 = 输入捕获通道2 噪声滤波器禁用 1 = 输入捕获通道2 噪声滤波器使能
5	**ENF1**	**输入捕获通道1 噪声滤波器使能** 0 = 输入捕获通道1 噪声滤波器禁用 1 = 输入捕获通道1 噪声滤波器使能
4	**ENF0**	**输入捕获通道0 噪声滤波器使能** 0 = 输入捕获通道0 噪声滤波器禁用 1 = 输入捕获通道0 噪声滤波器使能

CnL – 捕获n低字节, n = 0,1,2

寄存器	特殊功能寄存器地址	复位值
C0L	E4H, SFR页0	0000_0000 b
C1L	E6H, SFR页0	0000_0000 b
C2L	EDH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
CnL[7:0]							
读/写							

位	名称	描述
7:0	**CnL[7:0]**	**输入捕获n结果低字节** CnL寄存器是输入捕获n所捕获的16 位结果的低字节。

CnH – 捕获n 高字节, n = 1,2,3

寄存器	特殊功能寄存器地址	复位值
C0H	E5H, SFR页0	0000_0000 b
C1H	E7H, SFR页0	0000_0000 b
C2H	EEH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
CnH[7:0]							
读/写							

位	名称	描述
7:0	**CnH[7:0]**	**输入捕获n结果高字节** CnH寄存器是输入捕获n所捕获的16 位结果的高字节。

CAPCON3 – 输入捕获控制3

寄存器	特殊功能寄存器地址	复位值
CAPCON3	F1H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
CAP13	**CAP12**	**CAP11**	**CAP10**	**CAP03**	**CAP02**	**CAP01**	**CAP00**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
[7:4]	**CAP1[3:0]**	**输入捕获通道1 输入引脚选择** 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 其他 = P1.2/IC0
[3:0]	**CAP0[3:0]**	**输入捕获通道0 输入引脚选择** 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 其他 = P1.2/IC0

CAPCON4 – 输入捕获控制 4

寄存器	特殊功能寄存器地址	复位值
CAPCON4	F2H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	**CAP23**	**CAP22**	**CAP21**	**CAP20**
-	-	-	-	读/写	读/写	读/写	读/写

位	名称	描述
[3:0]	**CAP2[3:0]**	**输入捕获通道2 输入引脚选择** 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 其他 = P1.2/IC0

6.5.3 定时器3

6.5.3.1 概述

定时器3是一个16位自动重载、向上计数的定时器。用户可以使用T3PS[2:0](T3CON[2:0])选择预分频，并将重载值填入RH3和RL3寄存器以确定其溢出速率。用户可以设置TR3(T3CON.3)来开始计数。当计数器超过FFFFH时，TF3(T3CON.4)设置为1，并将R3H和R3L寄存器的内容重载到内部16位计数器。如果ET3(EIE1.1)被置为1，则将执行定时器3中断服务程序。进入中断服务程序后，TF3由硬件自动清除。

定时器3也可作为两个UART的波特率时钟源。详细信息请参阅 章节**錯誤! 找不到參照來源。**“**錯誤! 找不到參照來源。**”。

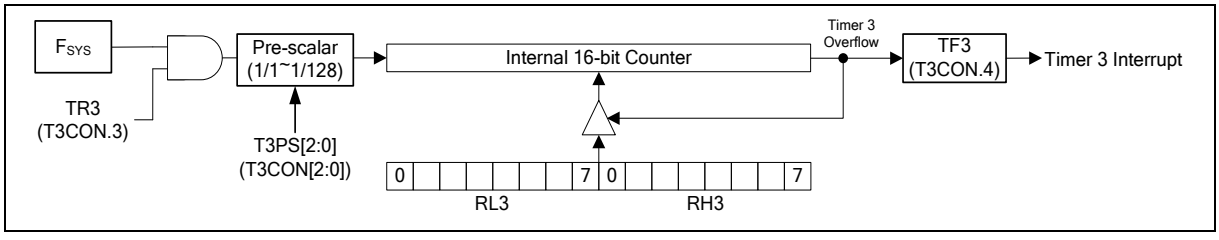


图 6.5-8 定时器 3 框图

6.5.3.2 定时器3的控制寄存器

T3CON – 定时器3控制

寄存器	特殊功能寄存器地址	复位值
T3CON	C4H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS[2:0]		
读/写	读/写	读/写	读/写	读/写	读/写		

位	名称	描述
4	TF3	**定时器3溢出标志** 当定时器3溢出时，此位置位。程序执行定时器3中断服务程序时，硬件自动清零此位。此位可由软件置位或清零。
3	TR3	**定时器3运行控制** 0 = 定时器3停止。 1 = 定时器3开始运行。 注意，重载寄存器RH3和RL3仅能在定时器3停止(TR3 位为 0)时写入。如果在TR3为1时写入RH3或RL3中的任何一个，结果将不可预测。
2:0	T3PS[2:0]	**定时器3预分频** 这些位段决定定时器3的时钟分频。 000 = 1/1. 001 = 1/2. 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

RL3 – 定时器3重载低字节

寄存器	特殊功能寄存器地址	复位值
RL3	C5H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
RL3[7:0]							
读/写							

位	名称	描述
7:0	**RL3[7:0]**	**定时器3重载低字节** 它保存了定时器3重载值的低字节。

RH3 – 定时器3重载高字节

寄存器	特殊功能寄存器地址	复位值
RH3	C6H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
RH3[7:0]							
读/写							

位	名称	描述
7:0	**RH3[7:0]**	**定时器3重载高字节** 它保存了定时器3重载值的高字节。

6.6 看门狗定时器 (WDT)

6.6.1 概述

CM1003提供一个看门狗定时器(WDT)。它可以配置为超时复位定时器，用以复位整个设备。一旦设备运行在异常状态或因外部干扰而挂起，WDT复位可以恢复系统。它提供了一个系统监视，提高了系统的可靠性。因此，WDT对于易受噪声干扰、电源故障或静电放电干扰的系统特别有用。WDT还可以配置为通用定时器，它的周期性中断可用作事件定时器或监控系统中的持续系统监测，可以在空闲模式或掉电模式下工作，WDTEN[3:0](CONFIG4[7:4])初始化WDT，使其作为超时复位定时器或通用定时器运行。

看门狗超时间隔由公式确定 $\frac{1}{F_{LIRC}\ \text{clock divider scalar}}$ 64，其中F_{LIRC}是内部10kHz振荡器的频率。下表显示了具有不同预分频的看门狗超时间隔的示例。

WDPS.2	WDPS.1	WDPS.0	时钟除频	WDT溢出时间[1]
0	0	0	1/1	6.40 ms
0	0	1	1/4	25.60 ms
0	1	0	1/8	51.20 ms
0	1	1	1/16	102.40 ms
1	0	0	1/32	204.80 ms
1	0	1	1/64	409.60 ms
1	1	0	1/128	819.20 ms
1	1	1	1/256	1.638 s
注意： 由于 LIRC 存在偏差，这是一个近似值。				

表6.6-1 看门狗定时器在不同分频下的超时间隔

由于WDT定时器延迟最大值的限制，要将CM1003从空闲模式或掉电模式唤醒，建议使用WKT功能，参见章节 6.7 自唤醒定时器 (WKT)。

6.6.2 功能描述

6.6.2.1 超时复位定时器

当CONFIG位WDTEN[3:0](CONFIG4[7:4])不为FH时，WDT被初始化为超时复位定时器。如果WDTEN[3:0]不为5H，则系统进入空闲或掉电模式后，允许WDT继续运行。请注意，WDT被初始化为超时复位定时器时，WDTR和WIDPD不起作用。

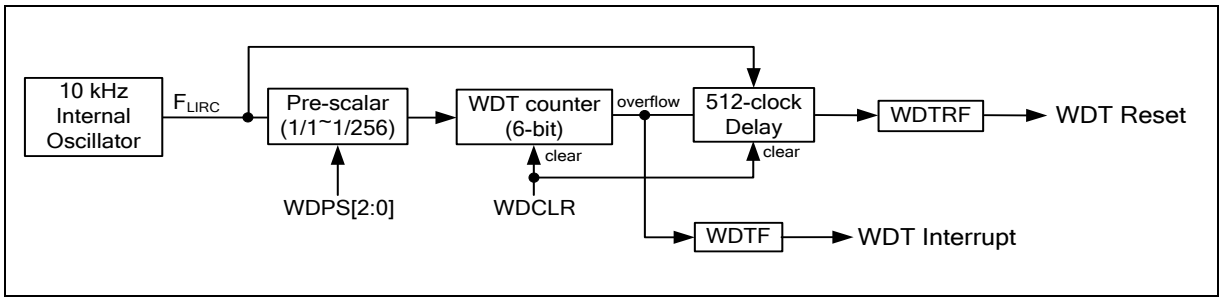


图 6.6-1 看门狗作为一个超时复位定时器

设备上电后，开始执行软件代码，同时WDT开始计数。超时间隔时间由WDPS[2:0](WDCON[2:0])三位选择。当所选超时发生时，WDT将置位中断标志WDTF(WDCON.5)。如果WDT中断使能位EWDT(EIE0.4)和全局中断使能位EA都设置，则将执行WDT中断程序。同时低速内部振荡器会延迟 512 个时钟周期，通过置位WDCLR来清零计数器，以避免在设备正常工作的情况下 WDT 导致系统复位。如果在此512个时钟周期内没有向WDCLR写入1来复位计数器，则将发生WDT复位。置位WDCLR位用来清零WDT计数器。此位会自动清除以供用户监控。一旦WDT复位发生，WDT复位标志WDTRF(WDCON.3)将会被置位，除上电复位之外的其他任何复位后，该位都保持不变。用户可以通过软件清零WDTRF。请注意，WDCON中的所有位都需要定时访问写入。

具有超时复位功能的WDT主要应用是系统监视器。这在实时控制应用中很重要。如果出现某些电源故障或电磁干扰，CPU可能会开始执行错误代码并以不可预测的状态运行。如果不加以控制，整个系统可能会崩溃。在软件开发过程中使用WDT的用户需要选择适当的“喂狗”时间来清除WDT计数器。通过插入设置 WDCLR 的指令，代码可以在不复位 WDT 的情况下运行。如果干扰引起代码运行在错误的状态下，导致无法及时清除看门狗定时器，将引起芯片复位，使系统从错误的状态恢复过来。

6.6.2.2 通用定时器

WDT还有另一种应用，用作简单的长周期定时器。当CONFIG位WDTEN[3:0](CONFIG4[7:4])为FH时，WDT被初始化为通用定时器。在此模式下，WDTR和WIDPD完全通过软件访问。

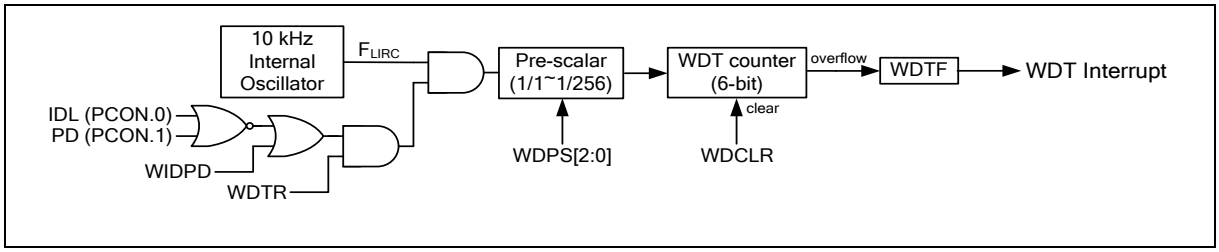


图 6.6-2 看门狗定时器框图

看门狗定时器通过设置WDTR为1开始运行，通过清零WDTR来停止运行。当WDT配置的时间间隔到后，WDTF标志位会被置位。软件将轮询WDTF标志位以检测超时。如果独立中断EWDT(EIE0.4)和全局中断使能EA都被置位，则将发生中断。WDT将继续计数。用户应清零 WDTF，并通过轮询WDTF标志位或等待中断发生来等待下一次溢出。

在一些低功耗应用中，无任何服务需求时，CPU通常会进入空闲模式以节省功耗。一段时间后，CPU将被唤醒，以定时器0~3设定的超时周期间隔检查是否有任何服务需求。但是，空闲模式的电流消耗仍然保持在“mA”级别。为了进一步将电流消耗降低到“uA”级别，无任何服务需求时，CPU应进入掉电模式，并能够以可编程的时间间隔唤醒。CM1003系列通过WDT唤醒功能实现了此实用功能。它提供了一个非常低功耗的10kHz内部振荡器作为WDT的时钟源。它还能够在掉电模式下计数并唤醒CPU。实现此功能的演示代码如下所示。

```
      ORG   0000H
      LJMP  START

      ORG   0053H
      LJMP  WDT_ISR

      ORG   0100H
     ;*********************************************************************
     ;WDT中断服务程序
     ;*********************************************************************
     WDT_ISR:
      CLR   EA
      MOV   TA,#0AAH
      MOV   TA,#55H
      ANL   WDCON,#11011111B   ;清除WDT中断标志位
      SETB  EA
      RETI
     ;*********************************************************************
     ;开始
     ;*********************************************************************
     START:
      MOV   TA,#0AAH
      MOV   TA,#55H
      ORL   WDCON,#00010111B    ;选择间隔长度并启用WDT
                               ;掉电时运行
      SETB  EWDT                ;WDT中断使能
      SETB  EA

      MOV   TA,#0AAH
      MOV   TA,#55H
      ORL   WDCON,#10000000B    ;WDT运行
     ;*********************************************************************
     ;进入掉电模式
     ;*********************************************************************
     LOOP:
      ORL   PCON,#02H
      LJMP  LOOP
```

6.6.3 寄存器描述

CONFIG4

7	6	5	4	3	2	1	0
WDTEN[3:0]				-	-	-	-
读/写				-	-	-	-

出厂默认值：1111 1111b

位	名称	描述
7:4	**WDTEN[3:0]**	**看门狗定时器WDT使能** 此字段用于配置 MCU 执行后 WDT 的行为。 1111 = 看门狗定时器WDT禁用，WDT可用作软件控制通用定时器 0101 = WDT作为超时复位定时器使用，并在空闲或掉电模式下停止运行。 其他 = WDT作为超时复位定时器使用，并在空闲或掉电模式下继续运行。

WDT 由一组分频器实现，用于对10 kHz 的低速内部振荡器时钟进行分频。分频器输出可选，并决定超时间隔。当达到超时间隔时，系统将从空闲或掉电模式唤醒，并且如果启用了 WDT 中断，则会产生中断事件。如果 WDT 初始化为超时复位定时器，则在没有任何软件操作的情况下，经过一段时间延迟后系统将复位。

WDCON – 看门狗定时器控制

寄存器	特殊功能寄存器地址	复位值
WDCON	AAH, SFR页0, TA保护	上电复位POR： 0000_0111 b 看门狗复位WDT： 0000_1UUU b 其他： 0000_UUUU b

7	6	5	4	3	2	1	0
WDTR	**WDCLR**	**WDTF**	**WIDPD**	**WDTRF**	**WDPS[2:0]**		
读/写	读/写	读/写	读/写	读/写	读/写		

位	名称	描述
7	**WDTR**	**WDT运行** 此位仅当WDTEN[3:0](CONFIG4[7:4]) 中的控制位全为1时有效。此时，WDT作为通用定时器工作。 0 = WDT禁止 1 = WDT使能，WDT计数器开始运行.
6	**WDCLR**	**WDT清零** 置位此位重置WDT计数为00H。它使计数器处于已知的状态，防止系统出现不可预知的复位。写入和读取WDCLR位的含义不同。 写入: 0 = 无影响 1 = 清除WDT计数器 读取: 0 = WDT计数器已完全清零 1 = WDT计数器尚未清零
5	**WDTF**	**WDT超时标志** 此位表示WDT计数器溢出。此标志位应由软件清零。
4	**WIDPD**	**WDT在空闲或掉电模式下运行** 此位仅当 WDTEN[3:0](CONFIG4[7:4])中的控制位全为1时才有效。当WDT作为通用定时器工作时，此位决定WDT在空闲或掉电模式下是否运行。 0 = WDT在空闲或掉电模式期间停止运行。 1 = WDT在空闲或掉电模式期间保持运行。
3	**WDTRF**	**WDT复位标志** 当MCU因WDT超时事件而复位时，此位由硬件置位。建议由软件清除此标志位。
2:0	**WDPS[2:0]**	**WDT时钟预分频选择** 这些位决定WDT时钟的预分频，范围从 1/1 到 1/256。
注意: 1. WDTRF上电复位后会被清0, WDT复位后会被置1, 其余复位后保持原有值不变。 2. WDPS[2:0] 上电复位后全部置位, 除此以外的任何其他复位，值均保持不变。		

6.7 自唤醒定时器 (WKT)

6.7.1 概述

CM1003具有专用的自唤醒定时器(WKT)，可在低功耗模式下用作周期性唤醒定时器或通用定时器。WKT在空闲或掉电模式下保持计数。当WKT用作唤醒定时器时，WKT可以在进入省电模式之前启动。WKT有一个内部10kHz的时钟源。请注意，系统时钟频率必须是WKT时钟的两倍以上。如果WKT开始计数，一旦设备进入空闲或掉电模式，所选时钟源也要保持运行状态。请注意，WKT的选定时钟源不会随WKT配置自动使能，用户应手动使能所选时钟源并等待它稳定以确保正常运行。

WKT配备一个简单的8位自动重载的向上计数定时器，预分频范围为1/1至1/2048，由WKPS[2:0](WKCON[2:0])选择。用户将重载值填入RWK寄存器以确定其溢出速率。WKTR(WKCON.3)置位开始计数。当计数器超过FFH时，WKTF(WKCON.4)被置为1，并生成重载，RWK 寄存器的内容重新加载到内部 8 位计数器中。如果 EWKT(EIE1.2)被置为1，将执行WKT中断服务程序。

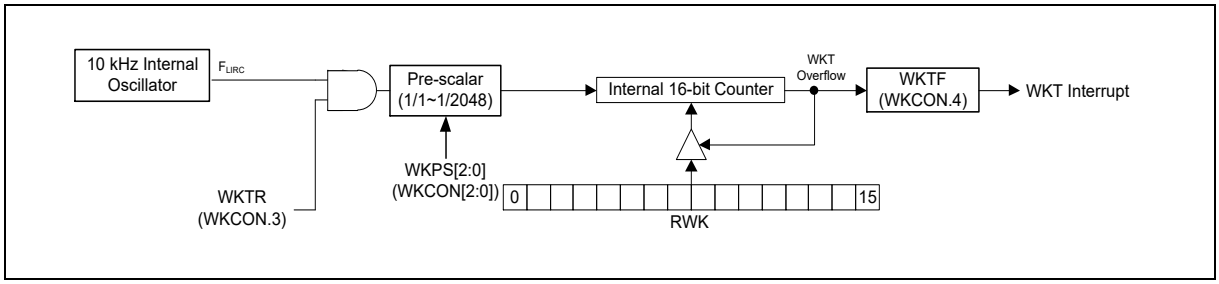


图 6.7-1 自唤醒定时器框图

6.7.2 寄存器描述

WKCON – 自唤醒定时器控制

寄存器	特殊功能寄存器地址	复位值
WKCON	8FH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	-	-	WKTF	WKTR	WKPS[2:0]		
-	-	-	读/写	读/写	读/写		

位	名称	描述
7:5	-	保留
4	WKTF	**WKT 溢出标志** 当WKT溢出，此位置位。如果WKT中断和全局中断使能，置位此位会使CPU执行WKT中断服务程序。此位不会被硬件自动清零，应通过软件清零。
3	WKTR	**WKT 运行控制** 0 = WKT 停止. 1 = WKT 开始运行. 注意：重载寄存器RWK仅在WKT停止(WKTR位为0)的时候可以写入。如果WKTR为1时写入WKT，结果不可预测。
2:0	WKPS[2:0]	**WKT 预分频** 这些位决定WKT时钟的预分频。 000 = 1/1. 001 = 1/4. 010 = 1/16. 011 = 1/64. 100 = 1/256. 101 = 1/512. 110 = 1/1024. 111 = 1/2048.

RWKL – 自唤醒定时器重新加载低字节

寄存器	特殊功能寄存器地址	复位值
RWKL	86H, SFR页0	0000 0000b

7	6	5	4	3	2	1	0
RWK[7:0]							
读/写							

位	名称	描述
7:0	**RWK[7:0]**	**WKT重载低字节** RWKL 寄存器是 WKT 16 位重载值的低字节。 注意，如果预分频比为 1/1，则 RWK 不应为 FFFFH，这是实现限制。

RWKH – 自唤醒定时器重新加载高字节

寄存器	特殊功能寄存器地址	复位值
RWKH	97H, SFR页2	0000 0000b

7	6	5	4	3	2	1	0
RWK[15:8]							
读/写							

位	名称	描述
7:0	**RWK[15:8]**	**WKT重载高字节** RWKH 寄存器是 WKT 16 位重载值的高字节。 注意，如果预分频比为 1/1，则 RWK 不应为 FFFFH，这是实现限制。

6.8 脉冲宽度调制 (PWM)

6.8.1 概述

PWM(脉冲宽度调制)信号是一种应用广泛的控制解决方案。它可用于电机驱动、风扇控制、背光亮度调节、LED灯调光，或通过低通滤波器电路模拟为简单的数模转换器输出。

CM1003 PWM0专为电机控制而设计，提供三对PWM0输出，最大16位分辨率，并具有可编程周期和占空比。该架构使得用户可以轻松驱动单相或三相无刷直流电机(BLDC)或三相交流感应电机。六个PWM均可配置为独立模式、互补模式或同步模式之一。如果使用互补模式，通过插入可编程的死区时间，来保护MOS管同时导通。PWM波形可配置成边沿对齐或中心对齐，具有可变的中断点。

CM1003 PWM1/2/3 提供可单独配置的周期和占空比，最高 16 位分辨率输出。PWM1/2/3各自2个PWM均可配置为独立模式、互补模式或同步模式。PWM1/2/3波形可以边沿对齐或中心对齐，并具有可变的中断点。

6.8.2 基本配置

PWM输出引脚定义和使能控制寄存器表。

PWM 通道	输出脚	控制寄存器 1			控制寄存器2		
		SFR字节名称	位名称	Value	SFR Byte Name	SFR字节名称	位名称
PWM0_CH0	P1.2	PIOCON0	PIO12	1	AUXR4[1:0]	PWM1C0P	00
	P3.3	PIOCON2	PIO33	1	-	-	-
PWM0_CH1	P1.1	PIOCON0	PIO11	1	AUXR4[3:2]	PWM1C1P	00
	P1.4	PIOCON1	PIO14	1	AUXR4[3:2]	PWM1C1P	00
PWM0_CH2	P0.5	PIOCON1	PIO05	1	AUXR4[5:4]	PWM2C0P	00
	P1.0	PIOCON0	PIO10	1	-	-	-
PWM0_CH3	P0.4	PIOCON1	PIO04	1	AUXR4[7:6]	PWM2C1P	00
	P0.0	PIOCON0	PIO00	1	AUXR4[7:6]	PWM2C1P	00
PWM0_CH4	P0.1	PIOCON0	PIO01	1	AUXR5[1:0]	PWM3C0P	00
PWM0_CH5	P0.3	PIOCON0	PIO03	1	AUXR5[3:2]	PWM3C1P	00
	P1.5	PIOCON1	PIO15	1	AUXR5[3:2]	PWM3C1P	00
PWM1_CH0	P2.3	PIOCON2	PIO23	1	AUXR4[1:0]	PWM1C0P	01
	P1.2	PIOCON0	PIO12	1			10
PWM1_CH1	P2.2	PIOCON2	PIO22	1	AUXR4[3:2]	PWM1C1P	01
	P1.4	PIOCON1	PIO14	1			10
	P1.1	PIOCON0	PIO11	1			11
PWM2_CH0	P2.1	PIOCON2	PIO21	1	AUXR4[5:4]	PWM2C0P	00
	P1.0	PIOCON0	PIO10	1			01
	P0.5	PIOCON1	PIO05	1			10
PWM2_CH1	P3.0	PIOCON2	PIO30	1	AUXR4[7:6]	PWM2C1P	00
	P3.1	PIOCON2	PIO31	1			01
	P0.0	PIOCON0	PIO00	1			10
	P0.4	PIOCON1	PIO04	1			11
PWM3_CH0	P3.2	PIOCON2	PIO32	1	AUXR5[1:0]	PWM3C0P	01

PWM 通道	输出脚	控制寄存器 1			控制寄存器2		
		SFR字节名称	位名称	Value	SFR Byte Name	SFR字节名称	位名称
	P0.1	PIOCON0	PIO01	1			10
	P1.7	PIOCON1	PIO17	1			-
PWM3_CH1	P3.4	PIOCON2	PIO34	1	AUXR5[3:2]	PWM3C1P	01
	P1.5	PIOCON1	PIO15	1			10
	P0.3	PIOCON0	PIO03	1			11

表 6.8-1 PWM 引脚定义和使能控制寄存器

6.8.3 框图

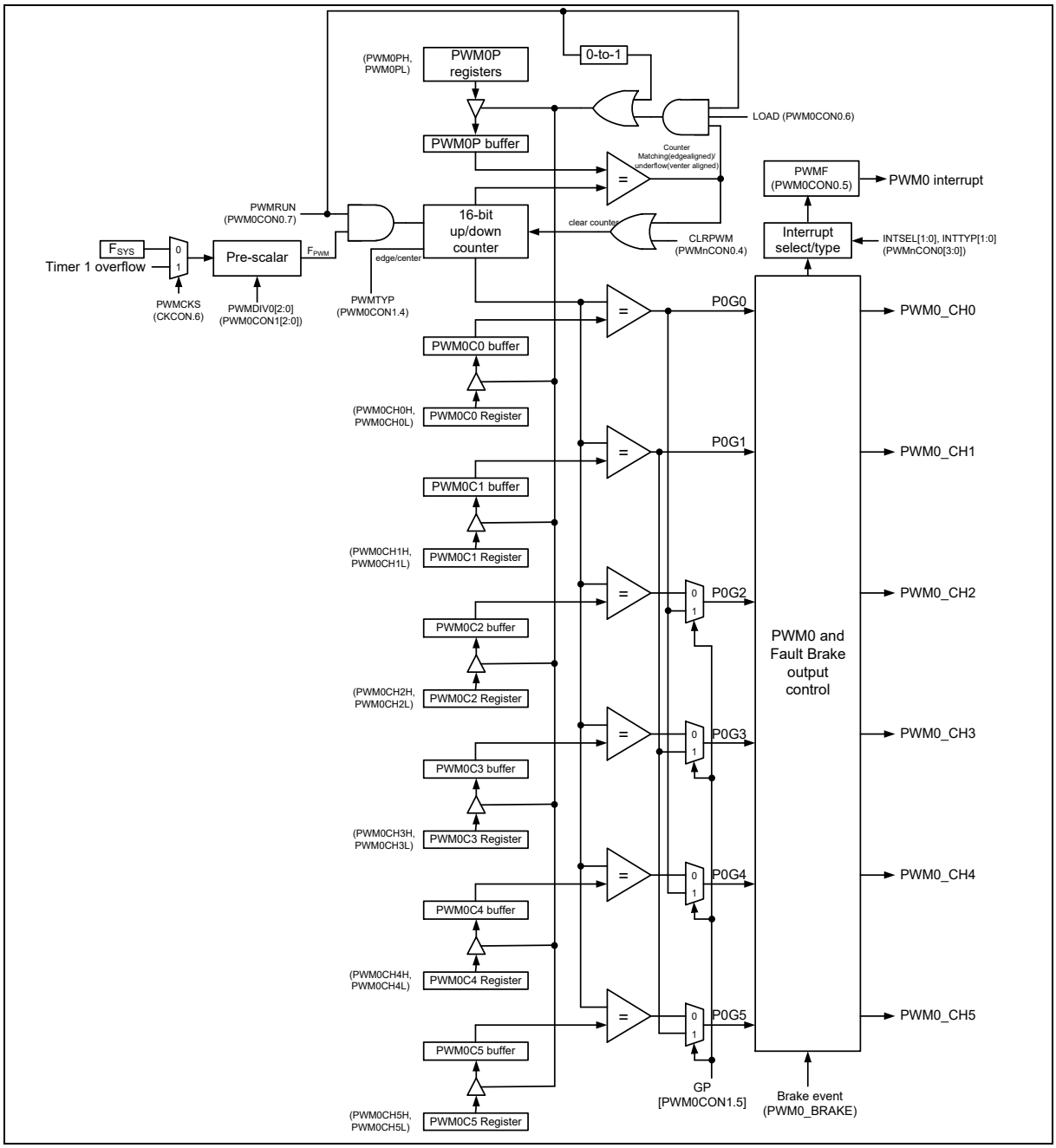


图 6.8-1 PWM0 框图

6.8.4 功能描述

6.8.4.1 脉冲宽度调制发生器

PWM发生器的时钟来源为系统时钟或定时器 1 溢出，除以 PWM 时钟预分频器（可选范围为 1/1 至 1/128）。PWM0/1/2/3 周期由有效的 16 位周期寄存器 {PWMnPH, PWMnPL} 定义。所有 PWM0/1/2/3 通道的周期相同，因为它们共享相同的 16 位周期计数器。每个 PWM 的占空比由占空比寄存器的值独立决定。PWM0 有六个占空比寄存器。PWM1/2/3 有两个占空比寄存器。这些 PWM 输出可以独立生成，并具有不同的占空比。PWM0/1/2/3 信号的间隔和占空比由 16 位计数器与周期和占空比寄存器比较产生。

只有 PWM0 具有分组模式以方便三相电机控制，可以通过设置 GP（PWM0CON1.5）来使用分组模式，这使得 {PWM0C0H, PWM0C0L} 和 {PWM0C1H, PWM0C1L} 占空比寄存器决定 PWM 输出的占空比。在三相电机控制应用中，两组 PWM 输出通常具有相同的占空比。当分组模式启用时，{PWM0C2H, PWM0C2L}、{PWM0C3H, PWM0C3L}、{PWM0C4H, PWM0C4L} 和 {PWM0C5H, PWM0C5L} 寄存器不起作用。这意味着 {PWM0C2H, PWM0C2L} 和 {PWM0C4H, PWM0C4L} 都与 {PWM0C0H, PWM0C0L} 相同。此外，{PWM0C3H, PWM0C3L} 和 {PWM0C5H, PWM0C5L} 与 {PWM0C1H, PWM0C1L} 相同。请注意，启用 PWM0 并不会自动将 I/O 引脚配置为输出模式。用户应通过软件手动配置 I/O 输出模式。

PWM0 计数器生成六个 PWM0 信号，分别为 P0G0、P0G1、P0G2、P0G3、P0G4 和 P0G5。这些信号将经过 PWM0 和故障刹车输出控制电路，产生实际的PWM0 输出至I/O引脚。输出控制电路决定 PWM 模式，死区插入，屏蔽输出，故障刹车控制和 PWM 极性。最后一步是 PWM0 输出或 I/O 功能的多路转换器。

PWM1/2/3 计数器生成两个 PWM1/2/3 信号。这些信号将经过 PWM1/2/3，产生实际的PWM1/2/3 输出至I/O引脚。输出控制电路决定 PWM 模式，屏蔽输出和 PWM 极性。最后一步是 PWM1/2/3 输出或 I/O 功能的多路转换器。

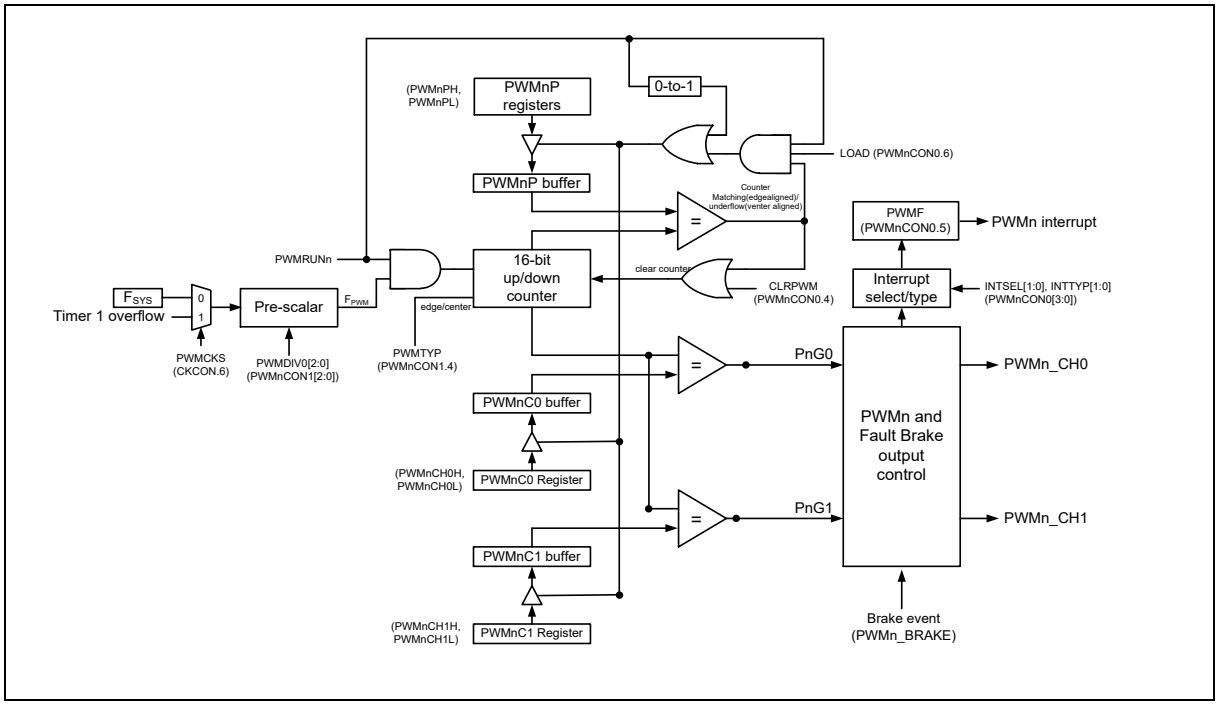


图 6.8-2 PWM1/2/3 框图

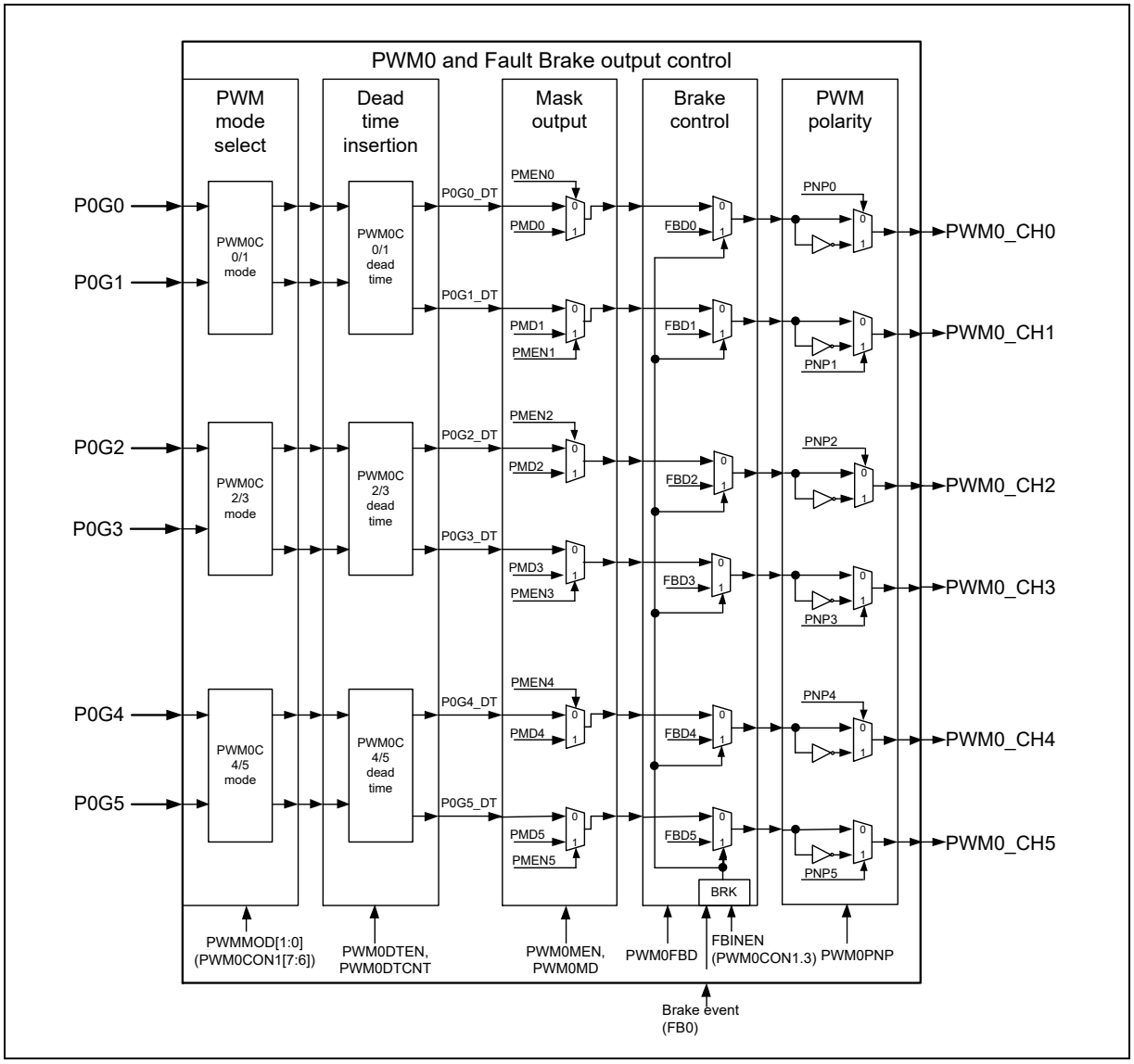


图 6.8-3 PWM0 和故障刹车输出控制框图

用户应按照以下步骤初始化PWM信号输出。第一步，设置CLRPWM (PWMnCON0.4)，以确保16位向上计数器复位来达到第一个周期的精度。初始化并设置 {PWMnPH, PWMnPL} 和所有{PWMnCxH, PWMnCxL} 寄存器后，PWMnRUN可设置为逻辑1，以触发16位计数器运行。PWM开始在其输出引脚上产生波形。所有周期和占空比控制寄存器的硬件均采用双缓冲设计。因此 {PWMnPH, PWMnPL}和所有 {PWMnCxH, PWMnCxL} 寄存器可以随时写入，但PWM的周期和占空比不会立即更新，直到LOAD (PWMnCON0.6) 被置1且前一个周期完成才生效。这可避免在更新PWM周期或占空比时出现毛刺信号。

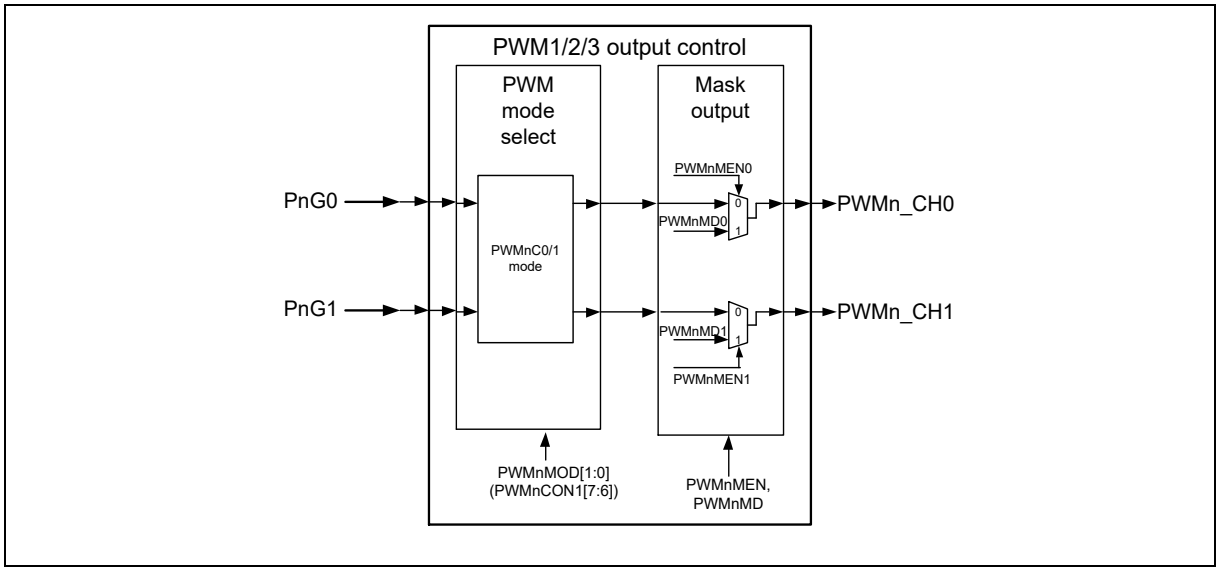


图 6.8-4 PWM1/2/3 控制框图

注意：通过设置LOAD来加载新的周期和占空比，应通过监控并等待硬件自动清除LOAD位来确保加载完成。在LOAD位为逻辑1期间对PWM控制寄存器的任何更新都将导致不可预测的输出。

6.8.4.2 PWM 类型

PWM发生器提供两种PWM类型：边沿对齐或中心对齐。PWM类型通过PWMTYP (PWMnCON1.4)来选择。

边沿对齐型

边沿对齐模式下，16位计数器采用单斜率操作，从0000H向上计数至 {PWMnPH, PWMnPL}，然后重新从0000H开始计数。PWM发生器信号(在PWM和故障刹车输出控制前的PGn信号)在16位计数器与占空比寄存器 {PWMnCxH, PWMnCxL} 比较匹配时清零，并且设16位计数器为0000H。PWM输出波形为左边沿对齐。

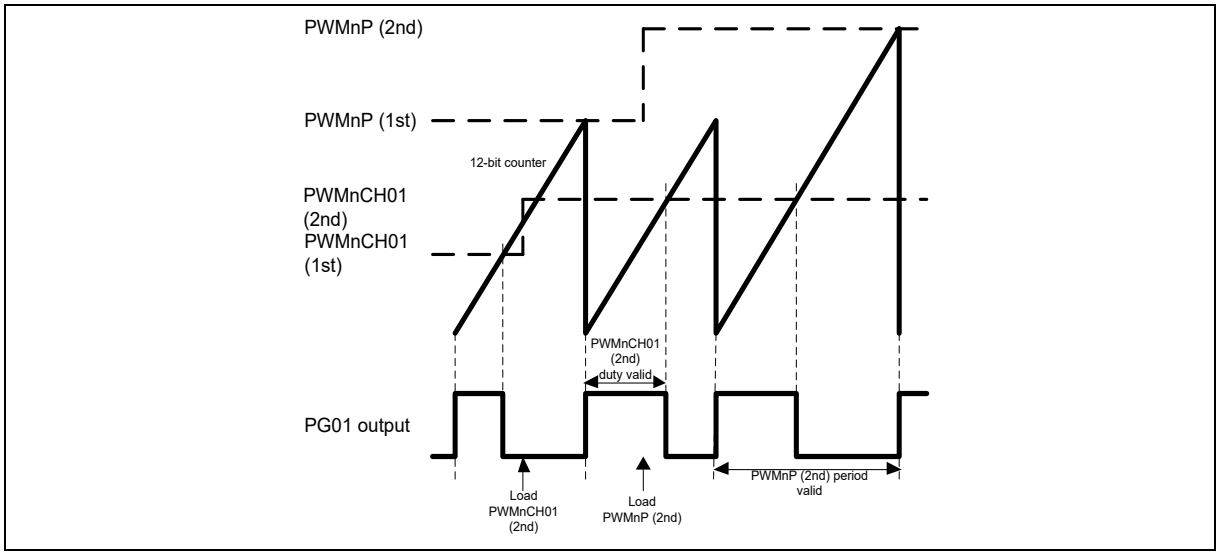


图 6.8-5 PWM 边沿对齐型波形

边沿对齐型PWM的输出频率及占空比算式如下:

$PWM\ 频率 = \frac{F_{PWM}}{\{PWMnPH,PWMnpl\}+1}$ (F_{PWM} 是PWM 时钟源频率除以PWMDIV).

$PWM\ 高电平占空比 = \frac{\{PWMnCHxH, PWMnCHxL\}}{\{PWMnPH, PWMnPL\}+1}$

中心对齐型

中心对齐模式下，16位计数器采用双斜率操作，从0000H向上计数至{PWMnPH, PWMnPL}，然后从{PWMnPH, PWMnPL}向下计数至0000H。PGn信号在16位计数器与占空比寄存器{PWMnCxH, PWMnCxL} 向上计数比较匹配时清零，在向下计数比较匹配时置位。中心对齐PWM可用于生成非重叠波形。

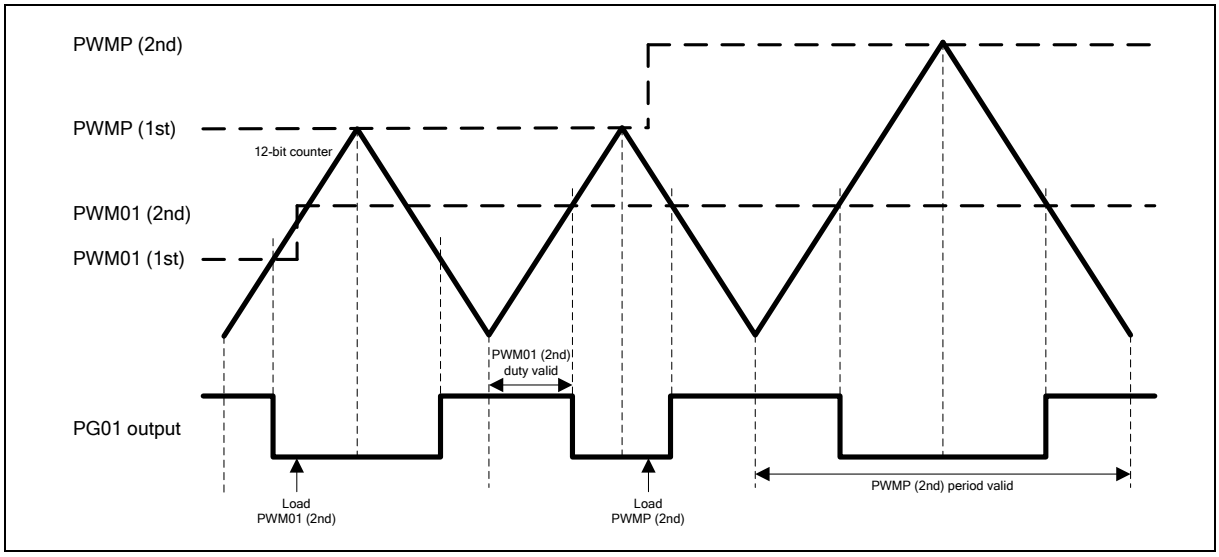


图 6.8-6 PWM 中心对齐型波形

中心对齐型PWM输出频率及占空比算式如下:

$PWM\ 频率 = \frac{F_{PWM}}{2\times\{PWMnPH,PWMnPL\}}$ (F_{PWM} 是PWM 时钟源频率除以PWMDIV).

PWM 高电平占空比 $= \frac{\{PWMnCHxH,PWMnCHxL\}}{\{PWMnPH,PWMnPL\}}$.

6.8.4.3 工作模式

PnGx 信号经过 PWM 的第一级后，PWM 模式选择电路会生成不同类型的 PWM 输出模式。它支持独立模式、互补模式和同步模式。

独立模式

当PWMMOD[1:0] (PWMnCON1[7:6]) 为 [0:0] 时，使能独立模式。这是 PWM 的默认模式。PnGx独立输出PWM信号。

带死区插入的互补模式

当PWMMOD[1:0] = [0:1]，互补模式使能。在此模式下，PnG0/2/4输出与独立模式相同的PWM信号。但是PnG1/3/5会相应地输出与PnG0/2/4反相的PWM信号，并忽略PnG1/3/5 的占空比寄存器{PWMnH, PWMnL} (n:1/3/5)。此模式使PnG0/PnG1成为一对互补的PWM输出，P0G2/P0G3和P0G4/P0G5也是如此。

在实际电机应用中，互补PWM0输出通常需要插入“死区”，用来防止半桥上下开关同时导通（即使在只是“us”级的时间）而损坏像GPIBs这样的电源开关器件。因为电源开关器件的物理状态无法瞬间打开/关闭。对于 CM1003 PWM0，每对 PWM0 共用一个 9 位死区递减计数器 PWM0DTCNT，用于产生同一对中两个 PWM0 信号之间的关断时间。在实现上，PWM0DTCNT 定时器下溢后，会在信号沿从0变为1的边缘加入一段延时等待。下图展示了带死区插入的P0G0/P0G1对互补模式的时序图。P0G2/P0G3 和 P0G4/P0G5 对具有相同的死区电路。每对在 PWM0DTEN [3:0] 字段中都有各自的死区使能位。

请注意，PWM0DTCNT 和 PWM0DTEN 寄存器均受 TA 写保护。死区控制仅在 PWM0 配置为互补模式时有效。死区插入仅在 PWM0 中有效。

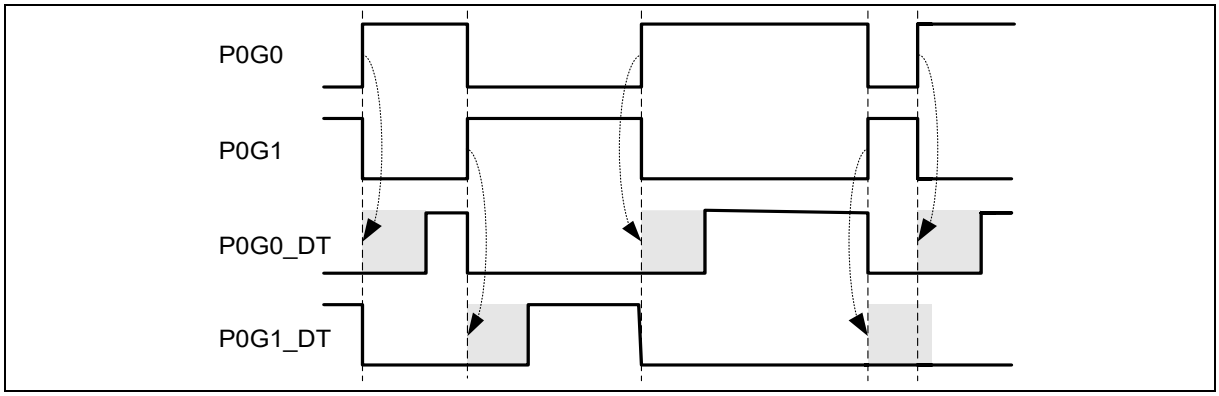


图 6.8-7 带死区时间插入的 PWM0 互补模式

同步模式

当PWMMOD[1:0] = [1:0] 时，同步模式使能。此模式下PnG0/2/4输出与独立模式相同的PWM信号，PnG1/3/5则输出与PnG0/2/4相同的同相PWM信号。

6.8.4.4 屏蔽输出控制

每一个PWM信号可以通过软件屏蔽输出一个指定的电平。PWM屏蔽输出功能在控制无刷直流电机(BLDC)等电子换向电机时非常有用。PWMnMEN用于决定屏蔽哪个PWM通道。PWMnMD分别设置每个 PWM 通道的屏蔽电平。PWMnMEN的默认值为 00H，这使得所有 PWM 通道的输出都遵循 PWM 发生器的信号。需要注意的是，屏蔽电平的反相与否取决于 PWM 输出引脚上 PWM0NP 的设置。

6.8.4.5 故障刹车

故障刹车功能通常与增强型PWM电路配合使用。它作为故障检测输入，保护电机系统免受损坏。FBINEN (PWMnCON1.3) 置位时，故障刹车引脚输入(FB)有效。当故障刹车有效时，PWM信号将被PWMnFBD相应位逐一覆盖。PWM0RUN（PWM0CON0.7）也将由硬件自动清零，以停止PWM生成。PWM 16位计数器也将复位为0000H。如果故障刹车中断使能，硬件将置位指示标志FBF，以触发故障刹车中断。即使FBF被软件清零，PWMnFBD数据输出仍然保留。用户只需再次置位PWM0RUN即可恢复PWM输出。同时，故障刹车状态将被解除，引脚上的PWM波形将恢复正常输出。请注意，由于FB引脚输入有永久的8/F_{SYS}去抖动时间，用于避免输入杂讯导致的虚假故障刹车事件，所以反馈到FB引脚的故障刹车信号输入时间应大于8个系统时钟时间。另一个触发故障刹车事件是ADC比较事件，它会触发与 FB 引脚输入相同的故障刹车行为。

请注意，故障刹车功能仅在 PWM0 中有效。

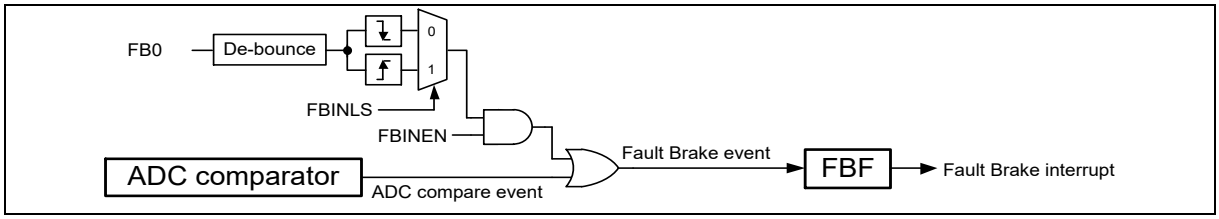


图 6.8-8 故障刹车功能框图

6.8.4.6 *极性控制*

每个 PWM0 输出通道都有独立的极性控制位，分别为 PNP0~PNP5。所有控制字段默认为高电平有效，采用正逻辑实现。这意味着 PWM 输出高电平时，电源开关开启；输出低电平时，电源开关关闭。用户可以轻松地将所有设置配置为正逻辑，然后设置 PNP 位，使 PWM0 实际按照负逻辑输出。

请注意，极性控制功能仅在 PWM0 中有效。

6.8.4.7 *PWM中断*

PWM 模块有一个标志位PWMF (PWMnCON0.5) ，用来指示每个完整PWM周期的特定点。指定的PWM通道和点可以通过INTSEL[2:0] 和INTTYP[1:0] (PWMnINTC[2:0] 和[5:4]) 选择。请注意，中心点和终点中断仅在PWM工作在中心对齐类型时才可用。PWMF由软件清除。

与PWM波形相关的PWM中断如下图所示。

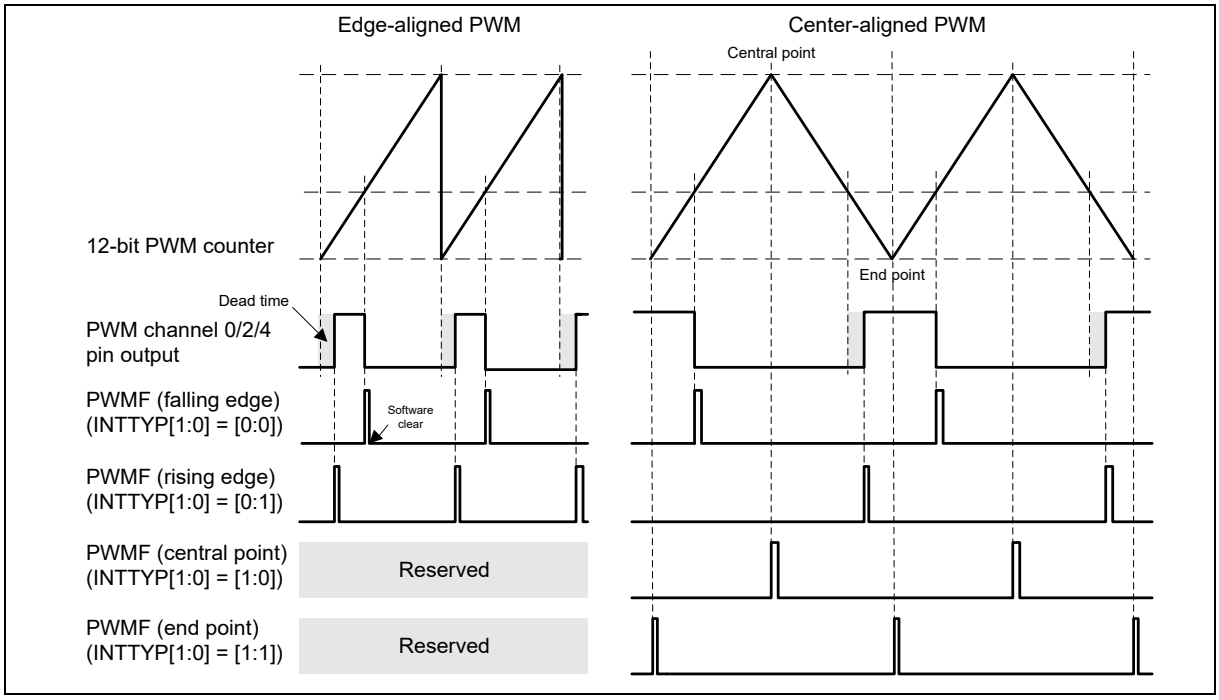


图 6.8-9 PWM 中断类型

故障刹车事件请求另一个中断，即故障刹车中断。它的中断向量与 PWM 中断不同。当发生故障刹车引脚输入事件或ADC比较事件时，FBF (PWM0FBD.7) 将由硬件置位。如果中断使能，将产生故障刹车中断。故障刹车中断使能位为EFB0 (EIE0.5) 。FBF由软件清除。

6.8.5 寄存器描述

PWM0CON0 – PWM0控制0(可位寻址)

寄存器	特殊功能寄存器地址	复位值
PWM0CON0	D8H, SFR页0, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
PWM0RUN	LOAD	PWMF	CLRPWM	PWM3RUN	PWM2RUN	PWM1RUN	P33FBINEN
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**PWM0RUN**	**PWM0运行使能** 0 = PWM0保持空闲。 1 = PWM0开始运行。 **注意：** 此位仅适用于 PWM0CON0。
6	**LOAD**	**PWM新周期和占空比加载** 需要更新新的周期或占空比值时，此位用于将周期和占空比控制寄存器加载到其缓冲区中。加载动作将在PWM周期完成时进行。新的周期及占空比值将作用于下一个周期。加载完成后，硬件自动将LOAD位清0。写入和读取 LOAD 位的含义不同。 写入: 0 = 无影响 1 = 在 PWM 周期完成时，将新的周期和占空比加载到其缓冲区中。 读取: 0 = 新周期和占空比载入已完成 1 = 新周期和占空比载入尚未完成
5	**PWMF**	**PWM标志位** 此位根据PWM0INTC的INTSEL[2:0]及 INTTYP[1:0] 的定义来设定。此位由软件清零。
4	**CLRPWM**	**清除PWM计数器** 设置此位将清除 PWM 16 位计数器的值，并将其重置为 0000H。计数器值清除后，此位会被硬件自动清0。写入和读取CLRPWM位的意义不同。 写入： 0 = 无影响。 1 = 清除PWM 16位计数器。 读取： 0 =PWM16 位计数器已完全清零。 1 =PWM16 位计数器尚未清零。
3	**PWM3RUN**	**PWM3 运行使能** 0 = PWM3 保持空闲状态。 1 = PWM3 开始运行。 **注意：**此位仅适用于 PWM0CON0。

位	名称	描述
2	**PWM2RUN**	**PWM2 运行使能** 0 = PWM2 保持空闲状态。 1 = PWM2 开始运行。 **注意：**此位仅适用于 PWM0CON0。
1	**PWM1RUN**	**PWM1 运行使能** 0 = PWM1 保持空闲状态。 1 = PWM1 开始运行。 **注意：**此位仅适用于 PWM0CON0。
0	**P33FBINEN**	**P33 FB 引脚输入使能** 0 = 由 P33 FB 引脚输入导致的PWM0 输出故障刹车禁用。 1 = 由 P33 FB 引脚输入导致的PWM0 输出故障刹车使能。一旦 FB 引脚出现 FBINLS (FBD.6) 选择的边沿，PWM0C0~C5 将在 FBD 寄存器中输出故障刹车数据。PWMRUN (PWM0CON0.7) 也将由硬件自动清零。再次设置 PWMRUN 后，PWM 输出将恢复。 **注意：**此位仅适用于 PWM0CON0。

PWMnCON0 – PWMn 控制 0, n = 1,2,3

寄存器	特殊功能寄存器地址	复位值
PWM1CON0	B4H, SFR页2	0000_0000 b
PWM2CON0	C4H, SFR页2	0000_0000 b
PWM3CON0	D4H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
PWMRUN	**LOAD**	**PWMF**	**CLRPWM**	**-**	**-**	**-**	**-**
读/写	读/写	读/写	读/写	-	-	-	-

位	名称	描述
7	**PWMRUN**	**PWM运行使能** 0 = PWM保持空闲。 1 = PWM开始运行。
6	**LOAD**	**PWM新周期和占空比加载** 需要更新新的周期或占空比值时，此位用于将周期和占空比控制寄存器加载到其缓冲区中。加载动作将在PWM周期完成时进行。新的周期及占空比值将作用于下一个周期。加载完成后，硬件自动将LOAD位清0。写入和读取 LOAD 位的含义不同。 写入: 0 = 无影响 1 = 在 PWM 周期完成时，将新的周期和占空比加载到其缓冲区中。 读取: 0 = 新周期和占空比载入已完成 1 = 新周期和占空比载入尚未完成
5	**PWMF**	**PWM标志位** 此位根据PWMINTC的INTSEL[2:0]及 INTTYP[1:0] 的定义来设定。此位由软件清零。
4	**CLRPWM**	**清除PWM计数器** 设置此位将清除 PWM 16 位计数器的值，并将其重置为 0000H。计数器值清除后，此位会被硬件自动清0。写入和读取CLRPWM位的意义不同。 写入： 0 = 无影响。 1 = 清除PWM 16位计数器。 读取： 0 =PWM16 位计数器已完全清零。 1 =PWM16 位计数器尚未清零。

PWMnCON1 – PWMn 控制1

寄存器	特殊功能寄存器地址	复位值
PWM0CON1	DFH, page 0	0000_0000 b
PWM1CON1	B5H, 页2	0000_0000 b
PWM2CON1	C5H, 页2	0000_0000 b
PWM3CON1	D5H, 页2	0000_0000 b

7	6	5	4	3	2	1	0
PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		
读/写		读/写	读/写	读/写	读/写		

位	名称	描述
7:6	**PWMMOD[1:0]**	**PWM 模式选择** 00 = 独立输出模式 01 = 互补模式 10 = 同步模式 11 = 保留.
5	**GP**	**分组模式使能** 此位使能PWM分组模式。一旦使能，前三对PWM的占空比由PWM01H和PWM01L决定，而不是由它们原本的占空比控制寄存器决定。 0 = 分组模式禁用 1 = 分组模式使能 **注意：**此位仅在 PWM0 中有效。
4	**PWMTYP**	**PWM类型选择** 0 = 边沿对齐型PWM 1 = 中心对齐型PWM
3	**FBINEN**	**FB引脚输入使能** 0= FB引脚输入导致PWM输出故障刹车禁用。 1= FB引脚输入导致PWM输出故障刹车使能。一旦FB引脚上出现与FBINLS (PWM0FBD.6)选择匹配的边沿信号，PWM0通道0~5输出PWM0FBD寄存器所设置的故障刹车数据，PWM0RUN (PWM0CON0.7)由硬件自动清除。当PWM0RUN再次置位时，PWM0输出恢复。 **注意：**此位仅在 PWM0 中有效。
2:0	**PWMDIV[2:0]**	**PWM 时钟分频** 此字段决定PWM时钟源的预分频。 000 = 1/1. 001 = 1/2. 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

CKCON – 时钟控制

寄存器	特殊功能寄存器地址	复位值
CKCON	8EH, SFR页0	0000_0000b

7	6	5	4	3	2	1	0
-	PWMCKS	-	T1M	T0M	-	CLOEN	-
-	读/写	-	读/写	读/写	-	读/写	-

位	名称	描述
6	PWMCKS	**PWM时钟源选择** 0 = PWM时钟源为系统时钟F_{SYS} 1 = PWM时钟源为定时器1的溢出时钟

PWMnPL – PWMn周期低字节, n = 0,1,2,3

寄存器	特殊功能寄存器地址	复位值
PWM0PL	D9H, SFR页0	0000_0000 b
PWM1PL	B1H, SFR页2	0000_0000 b
PWM2PL	C1H, SFR页2	0000_0000 b
PWM3PL	D1H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnP[7:0]							
读/写							

位	名称	描述
7:0	**PWMnP[7:0]**	**PWMn 周期低字节** 此字节与 PWMnPH一起控制PWMn发生器的信号周期。

PWMnPH – PWMn周期高字节, n = 0,1,2,3

寄存器	特殊功能寄存器地址	复位值
PWM0PH	D1H, SFR页0	0000_0000 b
PWM1PH	A9H, SFR页2	0000_0000 b
PWM2PH	B9H, SFR页2	0000_0000 b
PWM3PH	C9H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
PWMnP [15:8]							
读/写							

位	名称	描述
7:0	**PWMnP [15:8]**	**PWMn 周期高字节** 此字节与 PWMnPL一起控制PWMn发生器的信号周期。

PWMnCxH – PWMn 通道x 占空比高字节, n = 0,1,2,3; x = 0,1,2,3,4,5

寄存器	特殊功能寄存器地址	描述	复位值
PWM0C0H	D2H, SFR页0	PWM0 通道 0占空比高字节	0000_0000 b
PWM0C1H	D3H, SFR页0	PWM0 通道1占空比高字节	0000_0000 b
PWM0C2H	D4H, SFR页0	PWM0 通道2占空比高字节	0000_0000 b
PWM0C3H	D5H, SFR页0	PWM0 通道3占空比高字节	0000_0000 b
PWM0C4H	C4H, SFR页1	PWM0 通道4占空比高字节	0000_0000 b
PWM0C5H	C5H, SFR页1	PWM0 通道5占空比高字节	0000_0000 b
PWM1C0H	AAH, SFR页2	PWM1 通道0占空比高字节	0000_0000 b
PWM1C1H	ABH, SFR页2	PWM1 通道1占空比高字节	0000_0000 b
PWM2C0H	BAH, SFR页2	PWM2 通道0占空比高字节	0000_0000 b
PWM2C1H	BBH, SFR页2	PWM2 通道1占空比高字节	0000_0000 b
PWM3C0H	CAH, SFR页2	PWM3 通道0占空比高字节	0000_0000 b
PWM3C1H	CBH, SFR页2	PWM3 通道1占空比高字节	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [15:8]							
读/写							

位	名称	描述
7:0	**PWMnCx [15:8]**	**PWMnCx 占空比高字节** 该字节与PWMnCxL一起控制 PWM 发生器输出信号 PGx 的占空比。

PWMnCxL – PWMn 通道x 占空比低字节, n = 0,1,2,3; x = 0,1,2,3,4,5

寄存器	特殊功能寄存器地址	描述	复位值
PWM0C0L	DAH, SFR页0	PWM0 通道0占空比低字节	0000_0000 b
PWM0C1L	DBH, SFR页0	PWM0 通道1占空比低字节	0000_0000 b
PWM0C2L	DCH, SFR页0	PWM0 通道2占空比低字节	0000_0000 b
PWM0C3L	DDH, SFR页0	PWM0 通道3占空比低字节	0000_0000 b
PWM0C4L	CCH, SFR页1	PWM0 通道4占空比低字节	0000_0000 b
PWM0C5L	CDH, SFR页1	PWM0 通道5占空比低字节	0000_0000 b
PWM1C0L	B2H, SFR页2	PWM1 通道0占空比低字节	0000_0000 b
PWM1C1L	B3H, SFR页2	PWM1 通道1占空比低字节	0000_0000 b
PWM2C0L	C2H, SFR页2	PWM2 通道0占空比低字节	0000_0000 b
PWM2C1L	C3H, SFR页2	PWM2 通道1占空比低字节	0000_0000 b
PWM3C0L	D2H, SFR页2	PWM3 通道0占空比低字节	0000_0000 b
PWM3C1L	D3H, SFR页2	PWM3 通道1占空比低字节	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [7:0]							
读/写							

位	名称	描述
7:0	**PWMnCx [7:0]**	**PWMnCx 占空比低字节** 该字节与PWMnCxH一起控制 PWM 发生器输出信号 PGx 的占空比。

PWM0DTEN – PWM 死区时间使能

寄存器	特殊功能寄存器地址	复位值
PWM0DTEN	F9H, SFR页0, TA保护	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	PWM0DTCNT.8	-	PDT45EN	PDT23EN	PDT01EN
-	-	-	读/写	-	读/写	读/写	读/写

位	名称	描述
4	PWM0DTCNT.8	**PWM死区计数器第8位** 参见PWM0DTCNT 寄存器。
2	PDT45EN	**PWM0_CH4 / PWM0_CH5 对死区时间插入使能** 此位仅当PWM0_CH4/PWM0_CH5为互补模式时有效。 0 = P0G4/P0G5 对信号无延时。 1 = P0G4/P0G5 对信号的上升沿插入死区时间延时。
1	PDT23EN	**PWM0_CH2 / PWM0_CH3 对死区时间插入使能** 此位仅当PWM0_CH2 / PWM_CH3为互补模式时有效。 0 = P0G2/P0G3对信号无延时。 1 = P0G2/P0G3对信号的上升沿插入死区时间延时。
0	PDT01EN	**PWM0_CH0 / PWM0_CH1对死区时间插入使能** 此位仅当PWM0_CH0 / PWM0_CH1为互补模式时有效。 0 = P0G0/P0G1对信号无延时。 1 = P0G0/P0G1对信号的上升沿插入死区时间延时。

PWM0DTCNT – PWM 死区时间计数器

寄存器	特殊功能寄存器地址	复位值
PWM0DTCNT	FAH, SFR页0, TA保护	0000_0000 b

7	6	5	4	3	2	1	0
PWM0DTCNT[7:0]							
读/写							

位	名称	描述
7:0	**PWM0DTCNT[7:0]**	**PWM死区计数器低字节** 此8位字段与PWM0DTEN.4组合构成9位PWM死区计数器PWM0DTCNT。仅当PWM处于互补模式且PWM对应的PWM0DTEN位被置位时，此计数器才有效。 PWM死区时间= $\frac{PDTCNT+1}{F_{SYS}}$. 请注意，用户不应在PWM运行期间修改PWM0DTCNT。

PWMnMEN – PWMnCx 掩码使能, n=0,1,2,3;x=0,1,2,3,4,5

寄存器	特殊功能寄存器地址	描述	复位值
PWM0MEN	FBH, SFR页0	PWM0_CHx 掩码使能	0000_0000 b
PWM1MEN	ADH, SFR页2	PWM1_CHx 掩码使能	0000_0000 b
PWM2MEN	BDH, SFR页2	PWM2_CHx 掩码使能	0000_0000 b
PWM3MEN	CDH, SFR页2	PWM3_CHx 掩码使能	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
x	PMENx	**PWMn_CHx 掩码使能** 0 = PWMn_CHx 信号从其 PWM 发生器输出。 1 = PWMn_CHx 信号被 PMDx 屏蔽。 **注意：** PMEN2~5 仅适用于 PWM0。

PWMnMD – PWMnCx掩码数据

寄存器	特殊功能寄存器地址	描述	复位值
PWM0MD	FCH, SFR页0	PWM0_CHx掩码数据	0000_0000 b
PWM1MD	ACH, SFR页2	PWM1_CHx 掩码数据	0000_0000 b
PWM2MD	BCH, SFR页2	PWM2_CHx 掩码数据	0000_0000 b
PWM3MD	CCH, SFR页2	PWM3_CHx 掩码数据	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
x	PMDx	**PWMn_CHx Mask Data** 一旦设置了相应的PMENx, PWMn_CHx信号就会输出屏蔽数据。 0 = PWMn_CHx信号被 0 屏蔽。 1 = PWMn_CHx信号被 1 屏蔽。 **注意：** PMD2~5仅适用于 PWM0。

PWM0FBD – PWM 故障刹车数据

寄存器	特殊功能寄存器地址	复位值
PWM0FBD	D7H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
FBF	**FBINLS**	**FBD5**	**FBD4**	**FBD3**	**FBD2**	**FBD1**	**FBD0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**FBF**	**故障刹车标志位** 当FBINEN设置为1，且FB引脚检测到符合FBINLS(FBD.6)选择的边沿信号时，此位置1。此位需通过软件清0。FBF清除后，故障刹车数据输出不会被释放，直到PWMRUN(PWMCON0.7)被置位。
6	**FBINLS**	**PWM_BRAKE引脚输入电平选择** 0 = 下降沿。 1 = 上升沿。
5:0	**FBDx**	**PWM0_CHn故障刹车数据** 0 = 当故障刹车有效后， PWM0_CHn 信号将被 0 覆盖。 1 = 当故障刹车有效后， PWM0_CHn 信号将被 1 覆盖。

PNP – PWM 负极性

寄存器	特殊功能寄存器地址	复位值
PNP	D6H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	**PNP5**	**PNP4**	**PNP3**	**PNP2**	**PNP1**	**PNP0**
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
5:0	**PNPn**	**PWM0_CHn 负极性输出使能** 0 = PWM0_CHn 信号直接输出到PWM0_CHn引脚。 1 = PWM0_CHn 信号反相输出到PWM0_CHn引脚。

PIOCON0 – PWM或者 I/O选择

寄存器	特殊功能寄存器地址	复位值
PIOCON0	DEH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PIO05	PIO04	PIO03	PIO02	PIO01	PIO00
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:6	-	保留.
5	PIO03	**P0.3/PWM 引脚功能选择** 0 = P0.3/PWM 引脚用作P0.3. 1 = P0.3/PWM 引脚用作PWM0_CH5 / PWM3_CH1输出. 当AUXR5[7:6]. PWM3CH1P = 00, PIO03 = 1, P0.3 定义为PWM0_CH5. AUXR5[7:6]. PWM3CH1P = 11, PIO03 = 1, P0.3 定义为PWM3_CH1.
4	PIO01	**P0.1/PWM 引脚功能选择** 0 = P0.1/PWM 引脚用作P0.1. 1 = P0.1/PWM 引脚用作PWM0_CH4 / PWM3_CH0输出. 当AUXR5[5:4]. PWM3CH0P = 00, PIO01 = 1, P0.3 定义为PWM0_CH4. AUXR5[5:4]. PWM3CH0P = 10, PIO01 = 1, P0.3 定义为PWM3_CH0.
3	PIO00	**P0.0/PWM 引脚功能选择** 0 = P0.0/PWM 引脚用作P0.0. 1 = P0.0/PWM 引脚用作PWM0_CH3 / PWM2_CH1输出. 当AUXR4[7:6]. PWM2CH1P = 00, PIO00 = 1, P0.0 定义为PWM0_CH3. AUXR4[7:6]. PWM2CH1P = 10, PIO00 = 1, P0.0 定义为PWM2_CH1.
2	PIO10	**P1.0/PWM 引脚功能选择** 0 = P1.0/PWM 引脚用作P1.0. 1 = P1.0/PWM 引脚用作PWM0_CH2 / PWM2_CH0输出. 当AUXR4[5:4]. PWM2CH0P = 00, PIO10 = 1, P1.0 定义为PWM0_CH2. AUXR4[5:4]. PWM2CH0P = 10, PIO10 = 1, P1.0 定义为PWM2_CH0.
1	PIO11	**P1.1/PWM 引脚功能选择** 0 = P1.1/PWM 引脚用作P1.1. 1 = P1.1/PWM 引脚用作PWM0_CH1 / PWM1_CH1输出. 当AUXR4[3:2]. PWM1CH1P = 00, PIO11 = 1, P1.1 定义为PWM0_CH1. AUXR4[3:2]. PWM1CH1P = 11, PIO11 = 1, P1.1 定义为PWM1_CH1.
0	PIO12	**P1.2/PWM 引脚功能选择** 0 = P1.2/PWM 引脚用作P1.2. 1 = P1.2/PWM 引脚用作PWM0_CH0 / PWM1_CH0输出. 当AUXR4[1:0]. PWM1CH0P = 00, PIO12 = 1, P1.2 定义为PWM0_CH1. AUXR4[1:0]. PWM1CH0P = 10, PIO12 = 1, P1.2 定义为PWM1_CH0.

PIOCON1 – PWM或者 I/O选择

寄存器	特殊功能寄存器地址	复位值
PIOCON1	C6H, SFR页1	0000_0000 b

7	6	5	4	3	2	1	0
PIO17	**-**	**PIO15**	**-**	**PIO04**	**PIO05**	**PIO14**	**-**
读/写	-	读/写	-	读/写	读/写	读/写	-

位	名称	描述
7	**PIO17**	**P1.7/PWM 引脚功能选择** 0 = P1.7/PWM3_CH0 引脚用作GPIO P1.7. 1 = P1.7/PWM3_CH0 引脚用作 PWM3_CH0输出。
6	**-**	保留
5	**PIO15**	**P1.5/PWM引脚功能选择** 0 = P1.5/PWM 引脚用作P1.5. 1 = P1.5/PWM 引脚用作PWM0_CH5 / PWM3_CH1输出。 当AUXR5[3:2].PWM3CH1P = 00, PIO15 = 1, P1.5 定义为 PWM0_CH5. AUXR5[3:2].PWM3CH1P = 11, PIO15 = 1, P1.5定义为 PWM3_CH1.
4	**-**	保留
3	**PIO04**	**P0.4/PWM引脚功能选择** 0 = P0.4/PWM 引脚用作P0.4. 1 = P0.4/PWM 引脚用作PWM0_CH3 / PWM2_CH1输出。 当AUXR4[7:6].PWM2CH1P = 00, PIO04 = 1, P0.4定义为 PWM0_CH3. AUXR4[7:6].PWM2CH1P = 11, PIO04 = 1, P0.4定义为 PWM2_CH1.
2	**PIO05**	**P0.5/PWM引脚功能选择** 0 = P0.5/PWM 引脚用作P0.5. 1 = P0.5/PWM 引脚用作PWM0_CH2 / PWM2_CH0输出。 当AUXR4[5:4].PWM2CH1P = 00, PIO05 = 1, P0.5定义为 PWM0_CH2. AUXR4[5:4].PWM2CH1P = 11, PIO05 = 1, P0.5定义为 PWM2_CH0.
1	**PIO14**	**P1.4/PWM引脚功能选择** 0 = P1.4/PWM 引脚用作P1.4. 1 = P1.4/PWM 引脚用作PWM0_CH1 / PWM1_CH1输出。 当AUXR4[3:2]. PWM1CH1P = 00, PIO14 = 1, P1.4定义为 PWM0_CH1. AUXR4[3:2]. PWM1CH1P = 10, PIO14 = 1, P1.4定义为 PWM1_CH1.
0	**-**	保留

PIOCON2 – PWM或者I/O选择

寄存器	特殊功能寄存器地址	复位值
PIOCON2	B7H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
PIO34	PIO33	PIO32	PIO31	PIO30	PIO23	PIO22	PIO21
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**PIO34**	**P3.4/PWM 引脚功能选择** 0 = P3.4/PWM 引脚用作P3.4. 1 = P3.4/PWM 引脚用作PWM3_CH1输出.
6	**PIO33**	**P3.3/PWM 引脚功能选择** 0 = P3.3/PWM 引脚用作P3.3. 1 = P3.3/PWM 引脚用作PWM0_CH0输出.
5	**PIO32**	**P3.2/PWM 引脚功能选择** 0 = P3.2/PWM 引脚用作P3.2. 1 = P3.2/PWM 引脚用作PWM3_CH0输出.
4	**PIO31**	**P3.1/PWM 引脚功能选择** 0 = P3.1/PWM 引脚用作P3.1. 1 = P3.1/PWM 引脚用作PWM2_CH1输出.
3	**PIO30**	**P3.0/PWM 引脚功能选择** 0 = P3.0/PWM 引脚用作P3.0 1 = P3.0/PWM 引脚用作PWM2_CH1输出.
2	**PIO23**	**P23/PWM 引脚功能选择** 0 = P2.3/PWM 引脚用作P2.3. 1 = P2.3/PWM 引脚用作PWM1_CH0输出.
1	**PIO22**	**P2.2/PWM 引脚功能选择** 0 = P2.2/PWM 引脚用作P2.2. 1 = P2.2/PWM 引脚用作PWM1_CH1输出.
0	**PIO21**	**P2.1/PWM 引脚功能选择** 0 = P2.1/PWM 引脚用作P2.1. 1 = P2.1/PWM 引脚用作PWM2_CH0输出.

AUXR4 – 辅助寄存器 4

寄存器	特殊功能寄存器地址	复位值
AUXR4	A3H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
PWM2CH1P		PWM2CH0P		PWM1CH1P		PWM1CH0P	
读/写		读/写		读/写		读/写	

位	名称	描述
7:6	**PWM2CH1P**	**PWM2 通道1引脚选择** 00 = 将PWM2_CH1分配给P3.0 01 = 将PWM2_CH1分配给P3.1 10 = 将PWM2_CH1分配给P0.0 11 = 将PWM2_CH1分配给P0.4
5:4	**PWM2CH0P**	**PWM2 通道0引脚选择** 00 = 默认保留 01 = 将PWM2_CH0分配给P2.1 10 = 将PWM2_CH0分配给P1.0 11 = 将PWM2_CH0分配给P0.5
3:2	**PWM1CH1P**	**PWM1 通道1引脚选择** 00 = 默认保留 01 = 将PWM1_CH1分配给P2.2 10 = 将PWM1_CH1分配给 P1.4 11 = 将PWM1_CH1分配给 P1.1
1:0	**PWM1CH0P**	**PWM1 通道0引脚选择** 00 = 默认保留 01 = 将PWM1_CH0分配给P2.3 10 = 将PWM1_CH0分配给P1.2 11 = 保留

AUXR5 – 辅助寄存器 5

寄存器	特殊功能寄存器地址	复位值
AUXR5	A4H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
CLOP	T0P	-		PWM3CH1P		PWM3CH0P	
读/写	读/写	-		读/写		读/写	

位	名称	描述
3:2	**PWM3CH1P**	**PWM3 通道1引脚选择** 00 = 默认保留 01 = 将 PWM3_CH1分配给 P3.4 10 = 将 PWM3_CH1分配给P1.5 11 = 将 PWM3_CH1分配给P0.3
1:0	**PWM3CH0P**	**PWM3 通道0引脚选择** 00 = 默认保留 01 = 将 PWM3_CH0分配给P3.2 10 = 将 PWM3_CH0分配给P0.1 11 = 保留

PWMnINTC – PWM中断控制

寄存器	特殊功能寄存器地址	复位值
PWM0INTC	B7H, 页1	0000_0000 b
PWM1INTC	B6H, 页2	0000_0000 b
PWM2INTC	C6H, 页2	0000_0000 b
PWM3INTC	D6H, 页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	**INTTYP1**	**INTTYP0**	-	**INTSEL2**	**INTSEL1**	**INTSEL0**
-	-	读/写	读/写	-	读/写	读/写	读/写

位	名称	描述
5:4	**INTTYP[1:0]**	**PWM中断类型选择** 这些位用于选择PWM中断类型。 00 = PWM通道0/1/2/3/4/5 引脚的下降沿. 01 = PWM通道0/1/2/3/4/5 引脚的上升沿. 10 = PWM周期的中心点. 11 = PWM周期的终点. 注意：中心点中断方式或终点中断方式仅适用于PWM中心对齐模式
2:0	**INTSEL[2:0]**	**PWM中断组选择** 在PWM通道0~5引脚的PWM中断类型选择为上升沿或下降沿中断时，此位段用于选择哪个PWM通道发出PWM中断。 000 = PWM0_CH0. 001 = PWM0_CH1. 010 = PWM0_CH2. 011 = PWM0_CH3. 100 = PWM0_CH4. 101 = PWM0_CH5. 其他 = PWM0_CH0.

6.9 串口 (UART0 & UART1)

6.9.1 概述

CM1003 包含两个增强型全双工串口，并支持自动地址识别和帧错误检测。这两个串口的控制位实现方式相同。一般而言，下文仅提及串口 0，不再提到串口1.

每个串口支持一种同步通信模式（模式 0）和三种全双工 UART（通用异步收发器）模式（模式 1、模式 2 和模式 3）。这意味着它可以同时发送和接收数据。串口还带有接收缓冲区，这意味着在接收的前一个字节被读取之前，串口就能开始接收第二个字节。接收和发送都是对SBUF进行操作访问，写入SBUF会加载到发送寄存器，而读取SBUF是访问物理上独立的接收寄存器。串口共有4种操作模式，在这四种模式下，任何使用 SBUF 作为目标寄存器的指令都可以启动传输。

6.9.2 功能描述

6.9.2.1 操作模式

模式 0

模式0是与外部设备同步通信的方式。串行数据通过RXD引脚收发，TXD输出移位时钟。每帧接收或发送8位数据，因为发送或接收数据是通过同一条数据线RXD进行的，因此模式0提供半双工通信。如果SM2(SCON.5)为0，则波特率为F_{SYS}/12，如果SM2为1，则波特率增强为F_{SYS}/2。请注意，无论发送或接收，串行时钟始终由MCU生成。因此，模式0串行端口上的任何设备都都应接受MCU主机模式。图 6.9-1 显示了模式0下串行端口的相关时序。

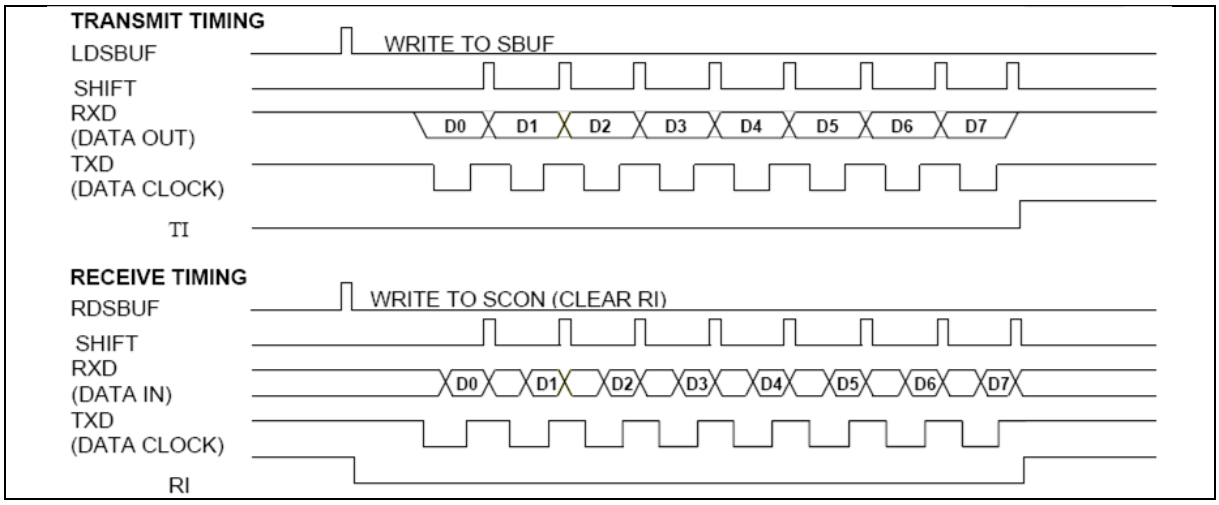


图 6.9-1 串口模式 0 时序图

如图所示，有一条双向数据线(RXD)和一条移位时钟线(TXD)。移位时钟用于逐位将数据移入或移出串口控制器，以进行串行通信。数据移入移出由LSB开始，波特率等于移位时钟频率。

任何写入SBUF的指令都会启动传输。控制模块将产生移位时钟并开始传输数据，直到8位都完成。然后传输标志TI(SCON.1)将设置为1，表示一个字节传输完成。

当REN(SCON.4)=1且RI(SCON.0)=0时串口开始接收数据。此条件告诉串口控制器有数据需要移入。此过程将持续到接收到8位数据为止。然后接收标志RI将被置1。用户可以清除RI以触发下一个字节的接收。

模式 1

模式1支持异步全双工串行通信。异步通讯模式通常用于与PC、调制解调器或其他类似接口通信。模式1下，10位数据由TXD发送，或由RXD接收，10位数据包括1个起始位(逻辑0)，8个数据位(最低有效位 (LSB) 优先)和1个停止位(逻辑1)。波特率由定时器1决定。SMOD(PCON.7)设置为1可使波特率加倍。图6.9-2 s显示了模式1下串口发送和接收的相关时序。

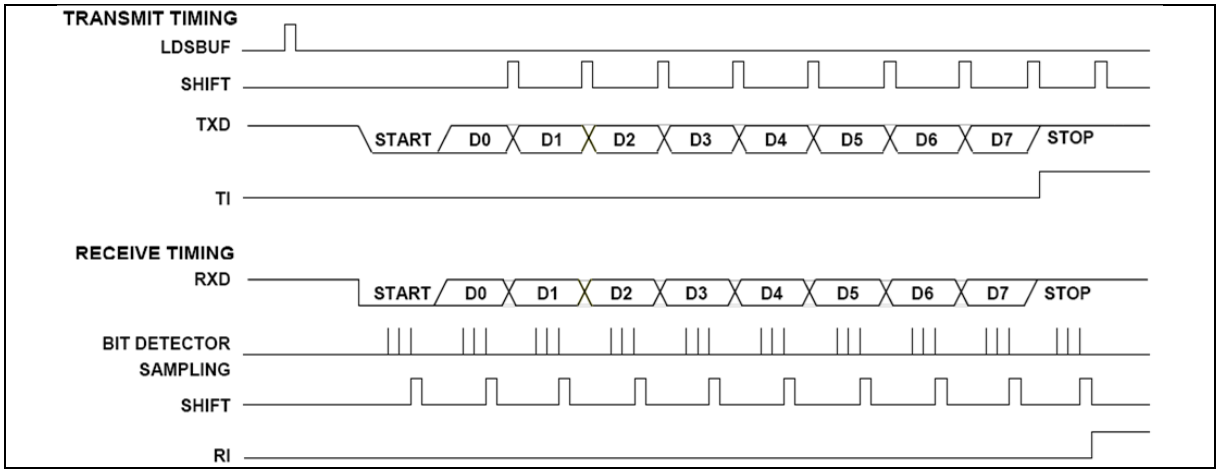


图 6.9-2 串口模式 1 时序图

任何写入SBUF的指令都会启动传输。传输在TXD引脚上进行。首先输出起始位，然后移出8位数据，最后以停止位结束。出现停止位后，TI(SCON.1)将置位以指示一个字节传输完成。所有位的移出速度由波特率发生器决定。

一旦激活波特率发生器且REN(SCON.4)为1，接收便可随时开始。当RXD脚上侦测到1到0的跳变时，接收启动。数据将以选定的波特率进行采样并移入。停止位应满足以下条件才能将接收到的数据加载到SBUF中：

1. RI (SCON.0) = 0，并且
2. 要么SM2 (SCON.5) = 0，或者 SM2 =1且接收到的数据与“给定”或“广播”地址匹配的情况下，接收到停止位 = 1。 (有关增强功能，请参阅章节 6.9.2.4 “多机” 和章节 **錯誤! 找不到參照來源。** “**錯誤! 找不到參照來源。**”。)

如果上述条件满足，SBUF将加载接收到的数据，RB8(SCON.2) 将加载停止位，并且RI将被置1，如果条件不满足，则不会加载任何数据，RI将保持为0。完成接收过程后，串口控制器将等待RXD引脚上的另一个 1 到 0 的跳变，以开始新的数据接收。

模式 2

模式2支持异步全双工串行通信。与模式1不同，模式 2有11位数据需要发送或接收，包括1个起始位(逻辑0)，8个数据位(最低有效位 (LSB) 优先)，1个可编程的第9位TB8或RB8，和1个停止位(逻辑1)。第9位最常见的用途是做奇偶校验位或多机通讯时用来区分数据和地址帧。波特率固定为系统时钟频率的1/32或1/64，具体取决于SMOD(PCON.7)位设定。图6.9-3 显示了模式2下串行端口发送和接收的相关时序。

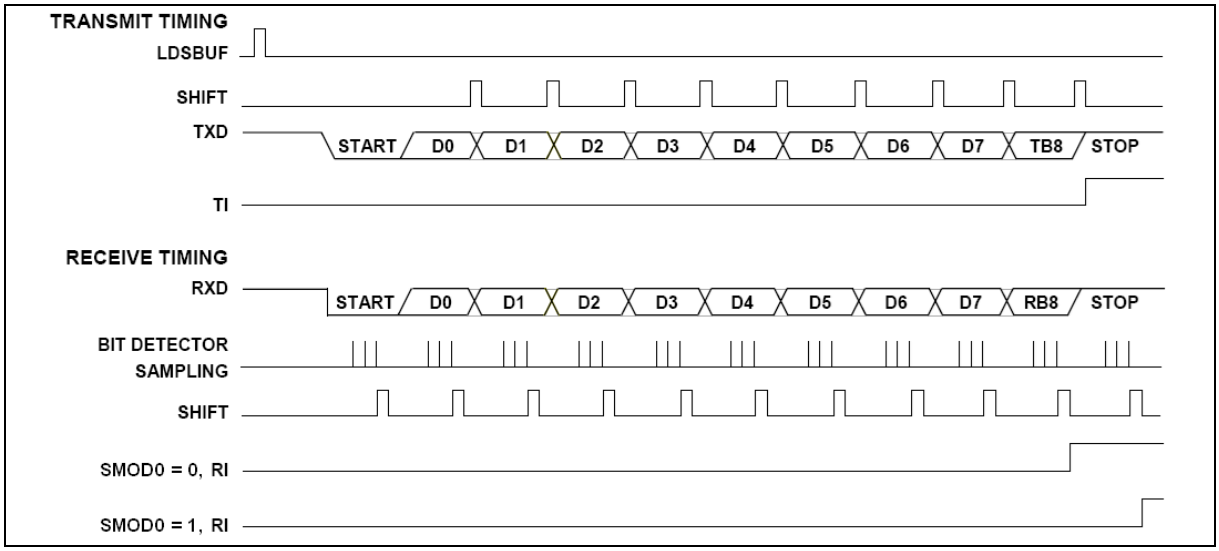


图 6.9-3 串口模式 2 和模式 3 时序图

任何写入SBUF的指令都会启动传输。传输在TXD引脚上进行。首先输出起始位，然后移出8位数据和TB8位(SCON.3)，最后输出停止位。停止位出现后，TI将置位以指示传输完成。

当REN置位后，可随时接收数据。RXD上的起始位的下降沿将启动接收进程。数据将以选定的波特率进行采样和移入。停止位应满足某些条件才能将接收到的数据加载到SBUF中：

1. RI (SCON.0) = 0，并且
2. 要么SM2 (SCON.5) = 0，或者 当SM2 =1且接收到的数据与“给定”或“广播”地址匹配的情况下，接收到的第9位 = 1。(有关增强功能，请参阅章节 6.9.2.4 “多机” 和章节 **錯誤! 找不到參照來源。** “**錯誤! 找不到參照來源。**”)

如果上述条件满足，SBUF 将加载接收到的数据，RB8(SCON.2) 将加载接收到的第9位，并且RI将置位。如果条件不满足，则不会加载任何数据，RI将保持为0。在完成上述接收过程后，串口控制器将等待RXD引脚上的另一个1到0的跳变，以开始新的数据接收。

模式 3

模式3的操作与模式2相同，只是其采用定时器1的溢出作为波特率时钟。模式3的时序图请参见图6.9-3 。它与模式2没有区别。

6.9.2.2 波特率

串口不同模式的波特率时钟源和速度差别很大，表6.9-1 串口 UART0 模式/波特率说明中列出了所有选项情况，用户应根据系统配置计算波特率。

在模式1或模式3下，UART0的波特率时钟源可通过BRCK(T3CON.5)在定时器1或定时器3中选择。对于UART1，只有采用定时器3作为唯一的时钟源。

当使用定时器1作为波特率时钟源时，请注意需要禁用定时器1中断。定时器1本身可配置为“计数器”或“定时器”操作，它可以处于三种运行模式中的任何一种。典型应用中，它被配置为“定时器”工作在自动重装载模式(模式2)。如果采用定时器3作为波特率发生器，同样也需要禁用其中断。

模式	帧位	SM0 / SM1 (SCON[7:6])	SM2 (SCON[5])	SMOD (PCON[7])	波特率	
0	8	00	0	-	F_{SYS} 除以12	
			1		F_{SYS} 除以2	
1	10	01	-	0	**Time1 T1M CKCON[4] = 0**	$\frac{1}{32}\times\frac{F_{SYS}}{12\times(256-TH1)}$
					Time1 T1M CKCON[4] = 1	$\frac{1}{32}\times\frac{F_{SYS}}{(256-TH1)}$
					Timer 3	$\frac{1}{32}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$
				1	**Time1 T1M CKCON[4] = 0**	$\frac{1}{16}\times\frac{F_{SYS}}{12\times(256-TH1)}$
					Time1 T1M CKCON[4] = 1	$\frac{1}{16}\times\frac{F_{SYS}}{(256-TH1)}$
					Timer 3	$\frac{1}{16}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$
2	11	10	-	0	F_{SYS} 除以64	
				1	F_{SYS} 除以32	
3	11	11	-	0	**Time 1 T1M CKCON[4] = 0**	$\frac{1}{32}\times\frac{F_{SYS}}{12\times(256-TH1)}$
					Time 1 T1M CKCON[4] = 1	$\frac{1}{32}\times\frac{F_{SYS}}{(256-TH1)}$
					Timer 3	$\frac{1}{32}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$
				1	**Time1 T1M CKCON[4] = 0**	$\frac{1}{16}\times\frac{F_{SYS}}{12\times(256-TH1)}$

					Time1 **T1M CKCON[4] = 1**	$\frac{1}{16} \times \frac{F_{SYS}}{(256\text{-TH1})}$
					Timer 3	$\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536\text{-}(256 \times \text{RH3} + \text{RL3}))}$
注意：定时器1应配置为定时器工作在自动重载模式(模式2)。						

表 6.9-1 串口 UART0 模式/波特率说明

模式	帧位	SM0_1 / SM1_1 (S1CON[7:6])	SMOD_1 (T3CON[7])	波特率	
0	8	00	-	F_{SYS} 除以12	
1	10	01	0	**Timer 3**	$\frac{1}{32} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536\text{-}(256 \times \text{RH3} + \text{RL3}))}$
			1	**Timer 3**	$\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536\text{-}(256 \times \text{RH3} + \text{RL3}))}$
2	11	10	0	F_{SYS} 除以64	
			1	F_{SYS} 除以32	
3	11	11	0	**Timer 3**	$\frac{1}{32} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536\text{-}(256 \times \text{RH3} + \text{RL3}))}$
			1	**Timer 3**	$\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536\text{-}(256 \times \text{RH3} + \text{RL3}))}$

表 6.9-2 串口 UART1 模式/波特率说明

示例代码: 我们列出了最常用的UART 设置模式 1 初始化步骤，如下所示：

串口0 (**UART0**) 使用定时器 1 作为波特率发生器: 公式为 $\frac{1}{16} \times \frac{F_{SYS}}{(256\text{-TH1})}$

```
SCON = 0x50;        //UART0模式1,REN=1,TI=1
TMOD |= 0x20;       //定时器1设置为模式2自动重载模式(必须)
PCON |= 0x80;       //UART0双倍速率使能
CKCON |= 0x10;      //定时器1作为时钟源
T3CON &= 0xDF;      //定时器1作为UART0时钟源
TH1 = value;
TR1=1;
```

串口0 (**UART0**) 使用定时器 3 作为波特率发生器: 公式为 $\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale}\times(65536\text{-}(256\times\text{RH3}+\text{RL3}))}$

```
SCON = 0x50;       //UART0 模式1,REN=1,TI=1
PCON |= 0x80;      //UART0 双倍速率使能
T3CON &= 0xF8;     //(预分频=1)
T3CON |= 0x20;     //UART0波特率时钟源 = 定时器3
```

```
    RH3 = value high byte
    RL3 = value low byte
    T3CON|= 0x08;          //触发定时器3
```

串口1 (UART1) 使用定时器 3 作为波特率发生器: 公式为 $\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536-(256 \times RH3+RL3))}$

```
    SCON_1 = 0x52;     //UART1模式1,REN_1=1,TI_1=1
    T3CON = 0xF8;      //T3PS2=0,T3PS1=0,T3PS0=0(预分频=1),
    RH3 = value 高字节
    RL3 = value 低字节
    T3CON|= 0x08;
```

下面根据不同的Fsys列出了一些常用的波特率值和精度值:

Fsys值	波特率	TH1值 (十六进制)	RH3,RL3值 (十六进制)	波特率精度
24 MHz	4800	64 (SMOD=0)	FEC8	0.160256%
	9600	64	FF64	0.160256%
	19200	B2	FFB2	0.160256%
	38400	D9	FFD9	0.160256%
	57600	E6	FFE6	0.160256%
	115200	F3	FFF3	0.160256%
	150000	F6	FFF6	0.000000%
	166666	F7	FFF7	0.000400%
	187500	F8	FFF8	0.000000%
	214285	F9	FFF9	0.000333%
	250000	FA	FFFA	0.000000%
	300000	FB	FFFB	0.000000%
	375000	FC	FFFC	0.000000%
	500000	FD	FFFD	0.000000%
	750000	FE	FFFE	0.000000%
	1500000	FF	FFFF	0.000000%
16 MHz	4800	30	FF30	0.160256%
	9600	98	FF98	0.160256%
	19200	CC	FFCC	0.160256%
	38400	E6	FFE6	0.160256%
	57600	EF	FFEF	2.124183%
	115200	F7	FFF7	-3.549383%
	200000	FB	FFFB	0.000000%
	250000	FC	FFFC	0.000000%
	333333	FD	FFFD	0.000100%
	500000	FE	FFFE	0.000000%
	1000000	FF	FFFF	0.000000%

6.9.2.3 帧错误检测

异步模式(模式1、2或3)提供帧错误检测。当由于总线干扰或争用而未检测到有效的停止位时，就会发生帧错误。UART可以检测到帧错误并通知软件。

帧错误位FE位于SCON.7，此位通常用作SM0，当帧错误访问使能位SMOD0(PCON.6)置1时，它将作为FE标志使用，实际上，SM0和FE位于不同的寄存器中。

发生帧错误时，FE位由硬件置位1。如有必要，可以在UART中断服务程序中检查FE。请注意，在读取或写入FE时，SMOD0应为1。如果FE被置位，那么后续接收的正确数据帧也不会清除FE标志。清除操作必须通过软件完成。

6.9.2.4 多机通讯

CM1003支持多机通讯，允许一个主机向多从机系统中一个从机发送多帧串行消息。这样做不会中断在同一串行线路上的其他从机设备的工作。此功能只能在UART模式2或3中使用。用户可以通过将SM2(SCON.5)设置为逻辑1来启用此功能，这样当接收到一个帧字节时，只有第9位为1时才会生成串行中断。(模式2下，第9位是停止位。)当SM2位为1时，第9位为0，接收的串行数据帧不会生成中断。在这种情况下，第9位能简单的把从机地址和数据区分开。

当主机想要通过串行线路向多个从机中的某一个发送数据块时，它会首先发出一个地址字节来标识目标从机。请注意，在这种情况下地址字节与数据字节不同。地址字节的第9位为1，而数据字节的第9位为0。地址字节会中断所有从机，以便每台从机检查接收到的地址是否与自身匹配。然后，被寻址的从机，会清除SM2，准备接收传入的数据字节；未被寻址的从机的SM2位保持置位，它们会继续正常运行，同时忽略传入的数据字节。

请按照以下步骤配置多机通讯：

1. 将所有设备(主机与从机)设置为UART模式2或3。
2. 将所有从机的SM2位写入1。
3. 主机的传输协议为：
   - 第一个字节：地址，用于标识目标从机，(第9位 = 1)。
   - 下一个字节：数据，(第9位 = 0)。
4. 当目标从机接收到第一个字节时，由于第9个数据位为1，因此所有从机都会中断。目标从机会将地址字节与自身地址进行比较，然后清除其SM2位以接收后续数据。其他从机继续正常运行。
5. 接收到所有数据字节后，将SM2重新设置为1以等待下一个地址。
6. SM2在模式0下无效，在模式 1 下可用于检查停止位的有效性。对于模式1接收，若SM2为1，则除非接收到有效的停止位，否则不会发出接收中断。

6.9.2.5 自动地址识别

自动地址识别增强了多机通讯的性能，它允许UART使用硬件进行比较，来识别串行比特流中的特定地址。此功能无需软件检查每个经过串行端口的串行地址，从而节省了大量的软件开销。仅当串口识别到自身地址时，接收器才会设置RI位以请求中断。当多机通讯功能被使能且SM2置位时，自动地址识别功能才会启用。

如果需要，用户可以在模式1中使能自动地址识别功能。在此配置中，停止位取代了第九个数据位。仅当收到的命令帧地址与设备地址匹配并由有效停止位终止时，RI 才会置位。

使用自动地址识别功能，主机可以通过调用“给定”从机地址来选择与一个或多个从机通信。可以使用“广播”地址联系所有从机。有两个特殊功能寄存器用于定义从机地址SADDR和从机地址掩码SADEN。SADEN用于定义SADDR中的哪些位要使用以及哪些位“无关”。SADEN掩码可以与SADDR进行逻辑

与运算以生成“给定”地址，主机将使用该地址寻址每个从机。使用“给定”地址可以识别多个从机，同时排除其他从机。

下列范例有助于展示此方案的多功能性：

例1, 从机 0:

```
SADDR = 11000000b
SADEN = 11111101b
给定地址 = 110000X0b
```

例2, 从机 1:

```
SADDR = 11000000b
SADEN = 11111110b
给定地址 = 1100000Xb
```

在上面的例子中，SADDR是相同的，SADEN数据用于区分两个从机。从机0要求位0为“0”，并忽略位 1。从机1要求位1为“0”，并忽略位0。从机0的唯一地址是11000010b，因为从机1要求位1为0。从机1的唯一地址是11000001b，因为位0为1将排除从机0。两个从机可以通过一个地址同时被选择，该地址的位0=0(对于从机0)和位1=0(对于从机1)。因此，两者都可以使用11000000b作为其“广播”地址进行寻址。

在更复杂的系统中，可以使用以下方法来选择从机1和2，同时排除从机0：

例1, 从机 0:

```
SADDR = 11000000b
SADEN = 11111001b
给定地址 = 11000XX0b
```

例2, 从机 1:

```
SADDR = 11100000b
SADEN = 11111010b
给定地址 = 11100X0Xb
```

例3, 从机 2:

```
SADDR = 11000000b
SADEN = 11111100b
给定地址 = 110000XXb
```

在上面的例子中，3个从机之间的区别在于地址的低3位。从机0要求位0= 0，其唯一寻址方式为

11100110b。从机1要求位1= 0，其唯一寻址方式为11100101b。从机2要求位2 = 0，其唯一寻址方式为11100011b。要选择从机0和从机1并排除从机2，请使用地址11100100b，因为必须使位 2 =1才能排除从机2。

每个从机的“广播”地址由SADDR和SADEN的逻辑或运算得出。结果中的零被视为“无关”，例如：

```
SADDR    = 01010110b
SADEN    = 11111100b
广播地址 = 1111111Xb
```

使用无关位可以灵活地定义广播地址，但是在大多数应用中，将“无关位”解释为全 1，则广播地址将为FFH。

复位时，SADDR和SADEN被初始化为00H。这将为所有“无关位”生成一个“给定”地址，并为所有 XXXXXXXXb（所有“无关位”）生成一个“广播”地址。这确保了串行端口能够响应任何地址，并使其向后兼容不支持自动地址识别的标准 80C51 微控制器。

6.9.3 寄存器描述

SCON – 串口控制

寄存器	特殊功能寄存器地址	复位值
SCON	98H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	SM0/FE	**串行端口选择** SMOD0(PCON.6)= 0: 参见表6.9-1 串口 UART0 模式/波特率说明了解详情。 SMOD0(PCON.6)= 1: SM0/FE 位用作帧错误(FE)状态标志位。由软件清零。 0 = 未发生帧错误(FE)。 1 = 发生并检测到帧错误(FE)。
6	SM1	

位	名称	描述
5	SM2	**多机通信模式使能** 此位功能取决于串口0模式。 模式0: 此位选择波特率 F_{SYS}/12 或 F_{SYS}/2 0 = 时钟以 F_{SYS}/12 波特率运行， 保持与标准8051的兼容性。 1 = 时钟以 F_{SYS}/2 波特率运行，以实现更快的串口通信。 模式1: 此位检查有效的停止位 0 = 无论停止位的逻辑电平，接收始终有效。 1 = 仅当接收的停止位是逻辑1并且接收的数据匹配“给定”或“广播”地址时，接收才有效。 模式 2 或 3: 用于多机通信。 0 = 不管第9位的逻辑电平，接收始终有效。 1 = 仅当接收的第9位是逻辑1并且接收的数据匹配“给定”或“广播”地址时，接收才有效。
4	REN	**接收使能** 0 = 串口0 接收禁用。 1 = 串口0 接收在模式1,2,3下使能。在模式0下， REN =1并且 RI =0时接收被启动初始化。
3	TB8	**第9个发送位** 此位定义串口0在模式2，3时第9个发送位的状态，模式0,1下不使用此位.
2	RB8	**第9个接收位** 此位标识串口0 模式2或3中的第9个接收位的逻辑电平。在模式1下，RB8为接收停止位的逻辑电平，SM2位为逻辑电平1 时，有异常限制。在模式0下RB8 不使用。
1	TI	**发送中断标志位** 模式0下串口0发送完第8位数据后，或其它模式下发送完数据帧最后一个数据位后，硬件会置位此标志位。当串口0中断使能，此位的置位将让CPU执行串口0 中断服务程序。此位应由软件手动清除。
0	RI	**接收中断标志位** 模式0下数据接收帧收到第8位后，或者模式1、2或3下采样到停止位后，硬件会置位此标志位。SM2位为逻辑电平1时，有异常限制。。当串口0中断使能，此位的置位将让CPU执行串口0 中断服务程序。此位应由软件手动清除。

SCON_1 – 串口1控制

寄存器	特殊功能寄存器地址	复位值
SCON_1	F8H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
SM0_1/FE_1	**SM1_1**	**SM2_1**	**REN_1**	**TB8_1**	**RB8_1**	**TI_1**	**RI_1**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	SM0_1/FE_1	**串口1模式选择** SMOD0_1(T3CON.6)= 0: 详见表6.9-2 串口UART1 模式/波特率 SMOD0_1(T3CON.6)= 1: SM0_1/FE_1用于帧错误(FE)状态标志。由软件清除. 0 = 未发生帧错误(FE) 1 = 发生并检测到帧错误(FE)
6	SM1_1	
5	SM2_1	**多机通信模式使能** 此位的功能取决于串口1模式。 模式 0： 没有效果。 模式 1： 该位检查有效的停止位。 0 = 无论停止位的逻辑电平如何，接收始终有效。 1 = 仅当接收到的停止位为逻辑1，并且接收到的数据与“给定”或“广播”地址匹配时，接收才有效。 模式 2 或 3： 用于多机通信。 0 = 无论第9位的逻辑电平如何，接收始终有效。 1 = 仅当接收到的第9位为逻辑1，且接收到的数据与“给定”或“广播”地址匹配时，接收才有效。
4	REN_1	**接收使能** 0 = 串口1接收禁用。 1 = 串口1接收在模式 1、2或3 下使能。在模式0下，接收由条件REN_1 =1和RI_1 = 0启动。
3	TB8_1	**第9位发送位** 串口1在模式2和3中第 9 个发送位的状态。在模式 0 或 1 下未使用。
2	RB8_1	**第9位接收位** 此位标识串口1 在模式 2 或 3 下第 9 个接收位的逻辑电平。模式1下，RB8_1为接收停止位的逻辑电平。SM2_1位为逻辑1时有例外限制。模式0下 RB8_1未使用。

位	名称	描述
1	TI_1	**发送中断标志位** 当串口1在模式0下发送完数据帧第8位后，或在其他模式下发送完一笔数据帧最后一个数据位后，此标志位由硬件置位。当串口1中断使能时，置位此位会导致CPU执行串口1中断服务程序。此位必须通过软件手动清除。
0	RI_1	**接收中断标志位** 当串口1在模式0下接收完数据帧第8位后，或在模式 1，2 或 3 下采样停止位后，此标志位由硬件置位。SM2_1位为逻辑1时有例外限制。当串口1中断使能时，此位置位会导致CPU执行串口1中断服务程序。此位必须通过软件手动清除。

PCON – 电源控制

寄存器	特殊功能寄存器地址	复位值
PCON	87H, 所有SFR页	上电复位: 0001_000b, 其它: 000U_0000b

7	6	5	4	3	2	1	0
SMOD	SMOD0	LPR	POF	GF1	GF0	PD	IDL
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	SMOD	**串口 0 双倍波特率使能** 当 UART0 处于模式 2， 或将定时器1溢出用作 UART0 模式1/3的波特率时钟源时，此位置位可使串口0 波特率加倍。详情请参见 表6.9-1 串口 UART0 模式/波特率说明。
6	SMOD0	**串口0 帧错误标志访问使能** 0 = SCON.7访问 SM0 位。 1 = SCON.7访问 FE 位。

T3CON – 定时器3控制

寄存器	特殊功能寄存器地址	复位值
T3CON	C4H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS[2:0]		
读/写	读/写	读/写	读/写	读/写	读/写		

位	名称	描述
7	SMOD_1	**串口1双倍波特率使能** 当 UART1 处于模式 2 时，置位此位可使串口1波特率加倍。详细说明参见表6.9-2 串口UART1 模式/波特率。
6	SMOD0_1	**串口1帧错误访问使能位** 0 = SCON_1.7访问 SM0_1 位。 1 = SCON_1.7访问 FE_1 位。

SBUF – 串口0数据缓冲区

寄存器	特殊功能寄存器地址	复位值
SBUF	99H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SBUF[7:0]							
读/写							

位	名称	描述
7:0	**SBUF[7:0]**	**串口0数据缓冲区** 这个字节实际上由两个独立的寄存器组成。一个是接收寄存器，另一个是发送缓冲区。当数据移入SBUF时，它会进入到发送缓冲区并被移位以进行串行传输。当数据从SBUF移出时，它来自接收寄存器。 通过向SBUF写入数据来启动传输。

SBUF_1 – 串口1数据缓冲区

寄存器	特殊功能寄存器地址	复位值
SBUF_1	9AH, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
SBUF_1[7:0]							
读/写							

位	名称	描述
7:0	**SBUF_1[7:0]**	**串口1数据缓冲区** 这个字节实际上由两个独立的寄存器组成。一个是接收寄存器，另一个是发送缓冲区。当数据移入SBUF_1 时，它进入到发送缓冲区并被移位以进行串行传输。当数据从SBUF_1 移出时，它来自接收寄存器。 通过向SBUF_1 写入数据来启动传输。

IE – 中断使能

寄存器	特殊功能寄存器地址	复位值
IE	A8H, 所有SFR页, 可位寻址	0000 _0000 b

7	6	5	4	3	2	1	0
EA	EADC	EBOD	ES	ET1	EX1	ET0	EX0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
4	ES	**串口0中断使能** 0 = 串口0中断禁用。 1 = 由TI(SCON.1)或RI(SCON.0)产生的中断使能。

EIE1 – 扩展中断使能1

寄存器	特殊功能寄存器地址	复位值
EIE1	9CH, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
-	-	EPWM3	EPWM2	EPWM1	EWKT	ET3	ES_1
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
0	ES_1	**串口1中断使能** 0 = 串口1中断禁止。 1 = 由TI_1(SCON_1.1)或 RI_1(SCON_1.0)产生的中断使能。

SADDR – 从机0地址

寄存器	特殊功能寄存器地址	复位值
SADDR	A9H, SFR页0	0000 _0000 b

7	6	5	4	3	2	1	0
SADDR[7:0]							
读/写							

位	名称	描述
7:0	**SADDR[7:0]**	**从机0地址** 此字节定义微控制器自身的从机地址，用于UART0 多机通讯。

SADEN – 从机0地址掩码

寄存器	特殊功能寄存器地址	复位值
SADEN	B9H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SADEN[7:0]							
读/写							

位	名称	描述
7:0	**SADEN[7:0]**	**从机0地址掩码** 此字节为UART0的掩码，其中无关位用0表示，来组成设备的“给定”地址。无关位可以灵活地同时寻址一个或多个从机。

SADDR_1 – 从机1地址

寄存器	特殊功能寄存器地址	复位值
SADDR_1	BBH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SADDR_1[7:0]							
读/写							

位	名称	描述
7:0	**SADDR_1[7:0]**	**从机1地址** 此字节定义微控制器自身的从机地址，用于串口UART1多机通讯。

SADEN_1 – 从机1 地址掩码

寄存器	特殊功能寄存器地址	复位值
SADEN_1	BAH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SADEN_1[7:0]							
读/写							

位	名称	描述
7:0	**SADEN_1[7:0]**	**从机1地址掩码** 此字节为UART1的掩码，其中无关位用0表示，来组成设备的“给定”地址。无关位可以灵活地同时寻址一个或多个从机。

AUXR1 – 辅助寄存器 1

寄存器	特殊功能寄存器地址	复位值
AUXR1	A2H , SFR页0	上电复位: 0000 0000b, 软件复位: 1U00 0000b, 复位脚: U100 0000b, 其他: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	**RSTPINF**	**HardF**	**-**	**GF2**	**UART0PX**	**0**	**DPS**
读/写	读/写	读/写	-	读/写	读/写	只读	读/写

位	名称	描述
2	**UART0PX**	**串口0引脚交换** 0 = 默认RXD 为 P0.7， TXD 为 P0.6。 1 = 交换RXD 为 P0.6， TXD 为 P0.7。 注意：设置或清除此位后，TXD及RXD的交换立即生效。用户需注意在串口发送或接收过程中不要交换引脚，否则可能引发不可预测的结果，且不会收到警告提醒。

AUXR2 – 辅助寄存器 2

寄存器	特殊功能寄存器地址	复位值
AUXR2	A1H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
UART2TXP		**UART2RXP**		**UART1TXP**		**UART1RXP**	
读/写		读/写		读/写		读/写	

位	名称	描述
3:2	**UART1TXP**	**UART1 TX引脚选择** 00 = 默认保留 01 = 将UART1 TX 分配给 P1.6 10 = 将UART1 TX 分配给 P3.6 11 = 将UART1 TX 分配给 P1.0
1:0	**UART1RXP**	**UART1 RX引脚选择** 00 = 默认保留 01 = 将UART1 RX 分配给 P0.2 10 = 将UART1 RX 分配给 P3.7 11 = 将UART1 RX 分配给 P0.0

6.10 ISO 7816-3 接口 (SC0~2 & UART2 ~ 4)

6.10.1 概述

CM1003系列提供符合ISO 7816-3标准的接口控制器（SC控制器），其异步协议基于ISO/IEC 7816-3标准。软件控制GPIO引脚，用于智能卡复位和卡片检测。该控制器还提供UART仿真，以实现高精度波特率通信。

6.10.2 特点

- 符合ISO-7816-3 T = 0, T = 1 标准
- 可编程传输时钟频率
- 可编程额外保护时间选择
- 支持自动反转约定功能
- 支持UART模式
  - 全双工异步通信
  - 支持每个通道可编程波特率发生器
  - 通过设置 SCnEGT 寄存器，可编程发送数据延迟时间（从最后一个停止位离开 TX-FIFO 到解除断言之间的时间）
  - 可编程奇偶校验或无校验位生成和检测
  - 可编程停止位，1 位或 2 位停止位的生成

6.10.3 基本配置

以下是 ISO 7816-3 多功能引脚定义

URAT 引脚	SC 引脚	引脚名称	SFR 定义		
			SFR 字节名称	SFR 位名称	数值
UART2_TXD	SC0_CLK	P0.3	AUXR2[7:6]	UART2TXP	01
		P3.0			10
UART2_RXD	SC0_DAT	P0.4	AUXR2[5:4]	UART2RXP	01
		P1.7			10
UART3_TXD	SC1_CLK	P1.2	AUXR3[3:2]	UART3TXP	01
		P1.5			10
		P0.5			11
UART3_RXD	SC1_DAT	P1.1	AUXR3[1:0]	UART3RXP	01
		P2.5			10
		P3.4			11
UART4_TXD	SC2_CLK	P2.3	AUXR3[7:6]	UART4TXP	01
UART4_RXD	SC2_DAT	P2.2	AUXR3[5:4]	UART4RXP	01

表 6.10-1 UART 智能卡引脚定义和使能控制寄存器

6.10.4 框图

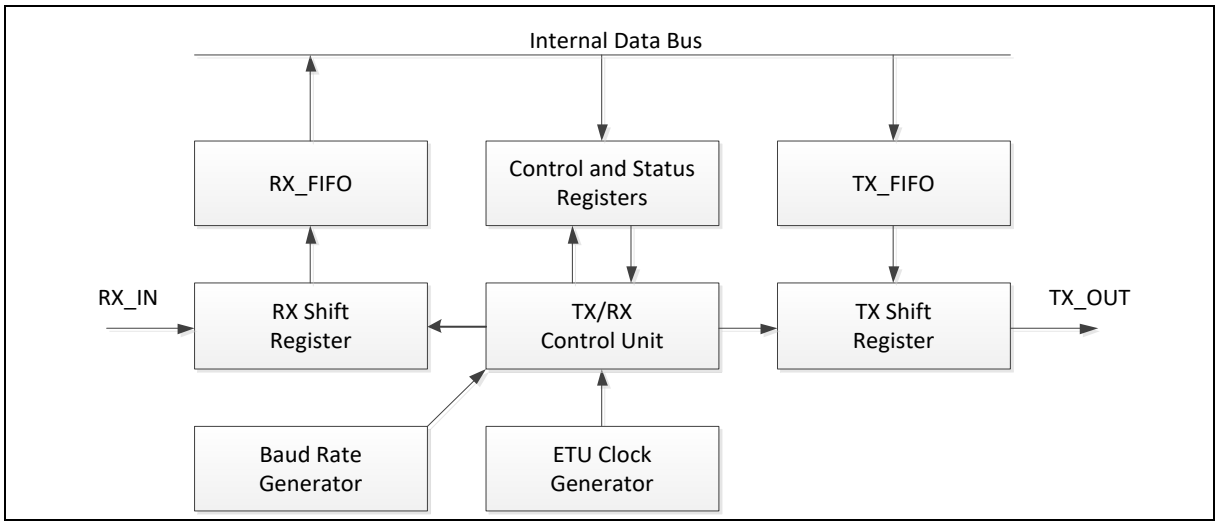


图 6.10-1 SC 控制器框图

6.10.5 功能描述

6.10.5.1 工作模式

ISO 7816-3 模式

ISO 7816-3 接口控制器支持软件控制激活、冷复位、热复位和停用时序。激活、冷复位、热复位和停用时序如下所示。

SC 接口连接

1. SCn_CLK / UARTn_TXD : SC 时钟引脚 (由MCU输出)
2. SCn_DAT / UARTn_RXD : SC 数据引脚 (双向)SCn_RST: SC 复位引脚 (由 MCU 输出，固件指定的 GPIO)
3. SCn_PWR: SC 电源引脚 (由 MCU 输出，固件指定的 GPIO)
4. *: SCn_PWR 用于电源控制功能，用于打开/关闭 ISO 7816-3的电源。请勿将 SCn_PWR用作 ISO 7816-3的直接电源。
5. SCn_CD: SC 卡检测引脚（输入至MCU，通过卡插入机制检测卡）

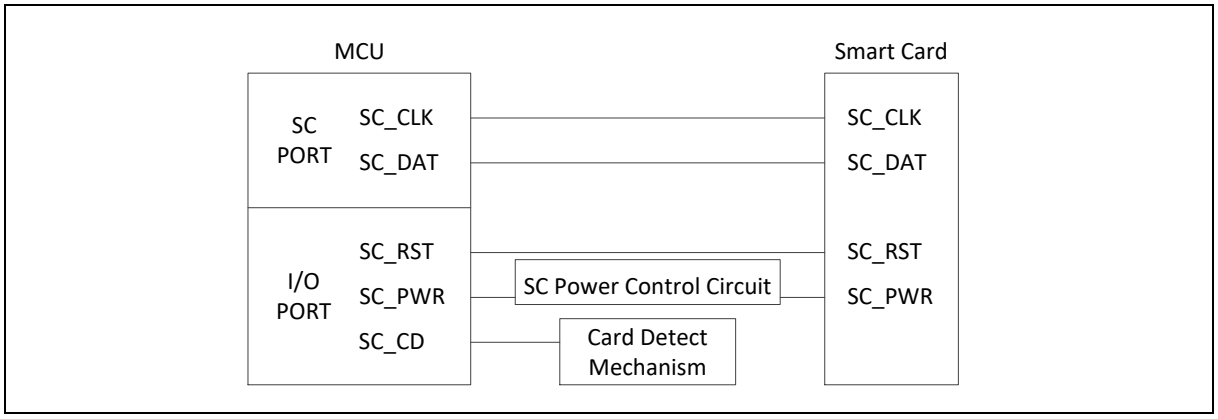
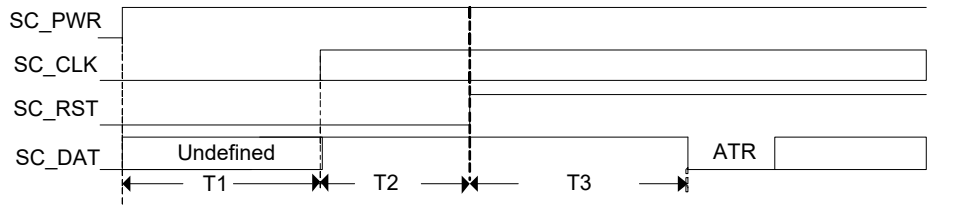


图 6.10-2 SC 接口连接

激活和冷复位

激活和冷复位序列如图6.10-3所示

1. 通过软件编程将 SCn_RST 设置为低电平“0”。
2. 在 T1 时刻之前，通过软件编程将 SCn_PWR 设置为高电平“1”，并在 T1 时刻期间将 SCn_DAT 设置为高电平（接收模式）。
3. 在 T1 时刻之后，通过将 CLKKEEP (SCCR2[1]) 设置为“1”，使能 SCn_CLK 时钟。
4. 在 T2 时刻之后，通过软件编程将 SCn_RST 设置为高电平“1”。
5. ISO 7816-3 主机控制器在 T3 时刻读取卡片的 ATR 数据。



Time	Comment
T1	SC_PWR to SC_CLK Start
T2	SC_CLK Start to SC_RST Assert
T3	SC_RST Start to ATR Appear

Suggestion timing (Unit SC Clock)

T1	T2
83.5	491
133	537
165	569
165	42060

400 <= T3 <= 40000

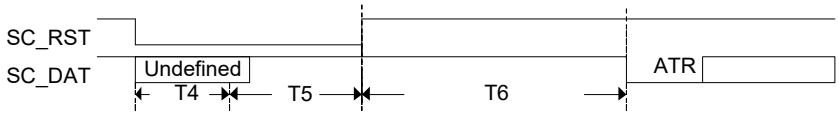
Note : The values are measured by chip I / O pin and the real value will depend on system design

图 6.10-3 SC 激活和冷复位序列

热复位

热复位序列如图6.10-4所示。

1. 在 T4 时刻之前，通过软件编程将 SCn_RST 设置为低电平。
2. 在 T4 时刻期间，通过软件编程将 SCn_DAT 设置为高电平。
3. 在 T5 时刻之后，通过软件编程将 SCn_RST 设置为高电平。
4. ISO 7816-3 主控制器在 T6 时刻读取卡片的 ATR 值。



Time	Comment
T4	SC_RST to SC_DAT Reception Mode
T5	SC_DAT Reception Mode to SC_RST Assert
T6	SC_RST start to ATR Appear

Suggesting timing (Unit SC Clock)

T4	T5
81	483
129	531
161	563
161	42106

400 <= T6 <= 40000

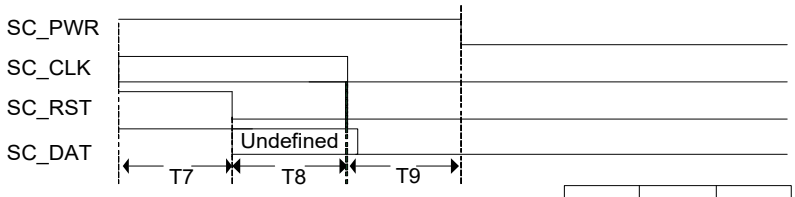
Note: This value is measured by chip IO pin, the real value will depended on system design

图 6.10-4 SC 热复位序列

停用

停用序列如图6.10-5所示

1. 通过软件编程将 SCn_RST 设置为低电平，在 T7 时序周期内将其置位。
2. 通过编程 CLKKEEP (SCCR2[1]) 为 0，在 T8 时序周期内将其置位，停止 SCn_CLK。
3. 通过软件编程将 SCn_DAT 设置为低电平，在 T8 时序周期内将其置位。
4. 通过软件编程将 SCn_PWR 设置为 0，在 T9 时序周期内将其停用。



Time	Comment
T7	Deactivation Trigger to SC_RST Low
T8	SMC_ RST Low to Stop SC _CLK
T9	Stop SC_CLK to Stop SC_PWR

Unit SC Clock

T7	T8	T9
97	83	87
145	131	135
177	163	167
177	163	167

图 6.10-5 SC 停用序列

UART 模式

当UARTEN (SCnCR2[0]) 置位时，ISO 7816-3 接口控制器也可用作基本 UART 功能。以下是 UART 模式的编程示例。

编程示例：

1. 置位UARTEN (SCnCR2[0]) 位以进入UART 模式。
2. 将CONSEL (SCnCR1[4]) 和 AUTOCEN (SCnCR1[3]) 字段填入“0”。 (在 UART 模式下，这些字段必须为 “0”)
3. 通过设置ETURDIV[11:0] ({SCnETURD1[3:0]:SCnETURD0[7:0]}) 字段来选择 UART 波特率。例如，如果智能卡模块时钟为12 MHz，目标波特率为115200bps, 则ETURDIV应填入 (12000000 / 115200 – 1).
4. 选择数据格式，包括数据长度 (通过设置WLS (SCCR2[5:4])，奇偶校验格式 (通过设置OPE (SCCR2[7]) ，PBOFF (SCCR2[6])) 和停止位长度 (通过设置NSB (SCnCR1[7])).
5. 写入SCnDR (SCnDR[7:0]) (TX) 寄存器或读取SCnDR (SCnDR[7:0]) (RX) 寄存器寄存器可执行 UART 功能。

6.10.5.2 ISO 7816-3 数据传输

ISO 7816-3接口基本用作半双工异步通信端口，其数据格式由 10 个连续的位组成，如下所示。

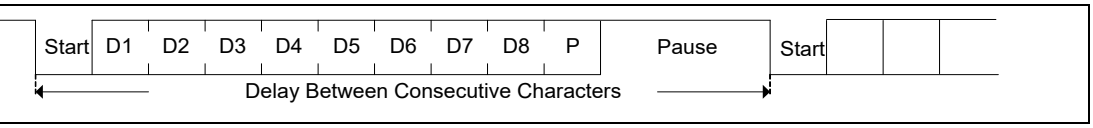


图 6.10-6 SC 数据字符

6.10.5.3 初始字符TS

根据7816-规范，初始字符TS有两种可能的模式，如图6.10-7 初始字符 TS所示。如果TS模式为1100_0000，则为反相约定。按反相约定后，传送的字节等于0x3F。如果TS模式为1101_1100，则为正向约定。按正向约定后，传送的字节等于0x3B。软件可以设置AUTOCEN (SCnCR1[3])，然后由硬件决定操作规则。软件也可以设置CONSEL (SCnCR1[4]) 寄存器 (设置为‘0’ 或‘1’)，以便在 SC 收到应答请求(ATR) 的 TS 后更改操作规则。

如果通过设置AUTOCEN (SCnCR1[3]) 寄存器寄存器使能自动转换功能，则设置步骤必须在应答请求状态之前完成，且第一个数据必须为0x3B或0x3F硬件接收到第一个数据并将其存储在缓冲区后，将自动决定转换方式并更改CONSEL (SCnCR1[4])寄存器。如果第一个数据既不是0x3B也不是0x3F，则硬件将向 CPU 发出中断 (如果ACERRIEN (ScnIE[4] = ‘1’) 。

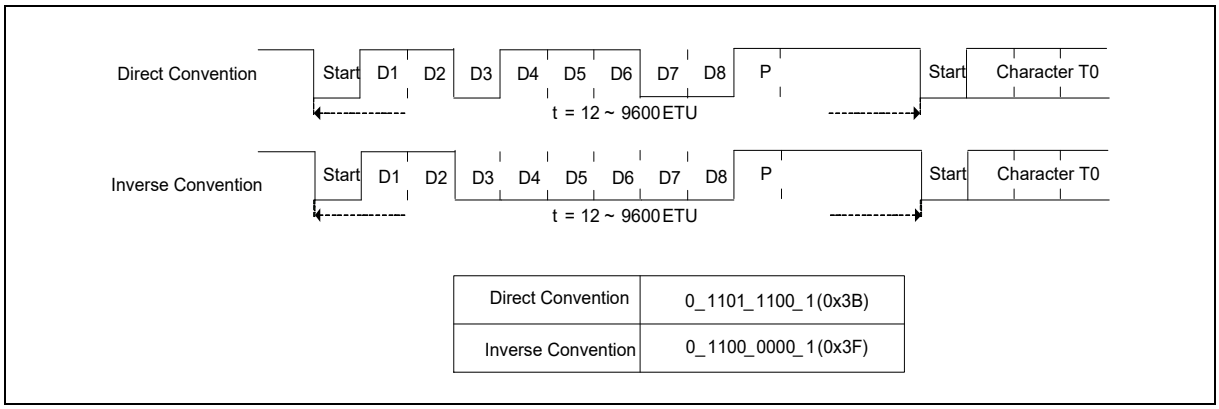


图 6.10-7 初始字符 TS

6.10.5.4 错误信号和字符重复

根据ISO 7816-3 T=0模式的描述，如图6.10-8 SC 所示，如果接收器接收到错误的奇偶校验位，它会将SC_DAT拉低 1.5 个位周期，以告知发送端奇偶校验错误。然后，发送端将重新发送该字符。SC 接口控制器在接收端中支持硬件错误检测功能(SC0TSR[4])，(如果TERRIEN(ScnIE[2] = ‘1’)，SC 控制器将向 CPU 发出传输错误中断。

在T=1 模式下，接收端不会将 SC_DAT拉低 1.5 个位周期来告知发送器奇偶校验错误。

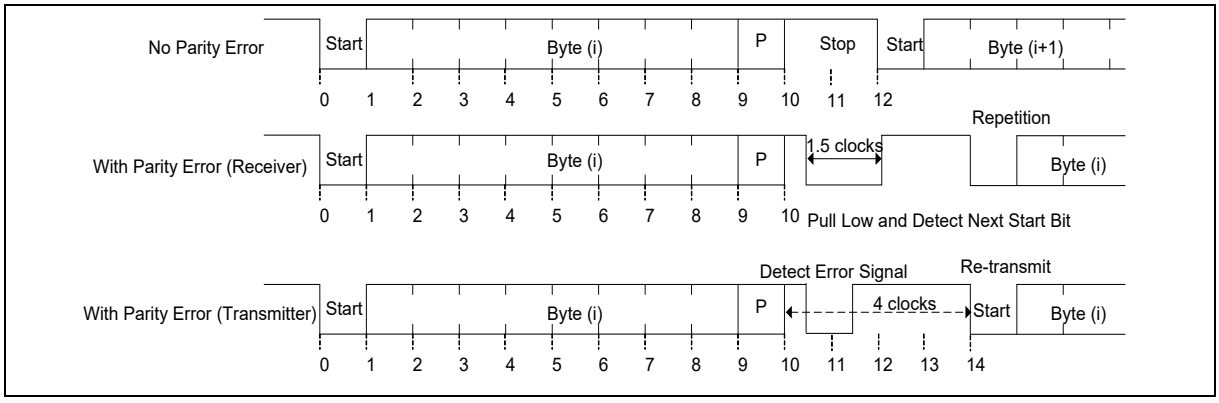


图 6.10-8 SC 错误信号

6.10.5.5 区块保护时间和额外保护时间

区块保护时间是指不同传输方向之间两个连续字符前沿之间的最小位长度。该字段表示块保护时间位长度的计数器。

根据ISO 7816-3的规定，在T = 0模式下，软件必须将T位填充为0 (实际区块保护时间 = 16.5)；在T = 1模式下，软件必须将 T 位填充为1 (实际区块保护时间 = 22.5)。

在发送方向上，ISO 7816首先将数据发送到ISO 7816-3主控制器。当周期大于(16.5或22.5，由 T 位设置)后，ISO 7816-3主控制器开始发送数据。

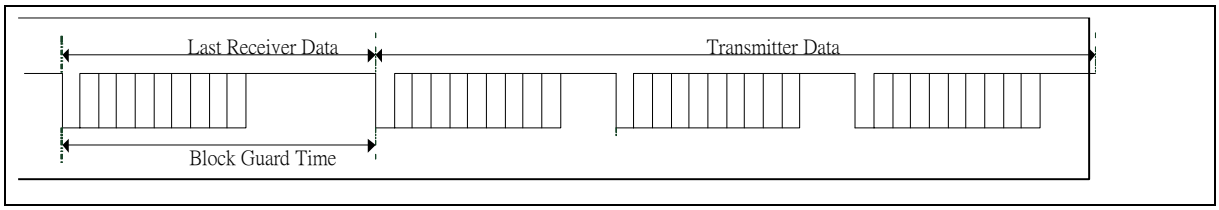


图 6.10-9 发送方向区块保护时间操作

在接收方向，ISO 7816-3主控制器首先向 ISO 7816-3发送数据。如果ISO 7816-3向主控制器发送数据的时间小于(16.5或22.5，由 T 位设置)，则在RXBGTEN (SCnCR1[5])和 BGTIEN (ScnIE[3]) 使能时，会产生区块保护时间中断BGTIF (ScnIS[3])。

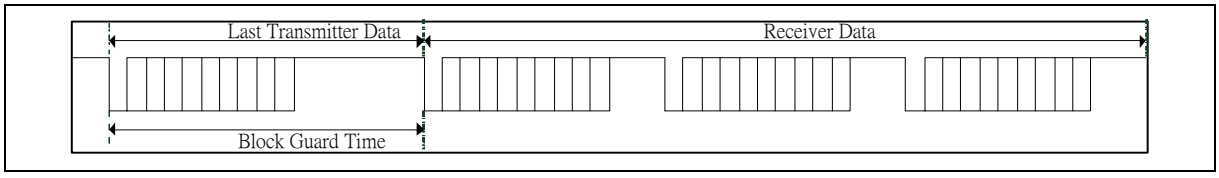


图 6.10-10 接收方向区块保护时间操作

额外保护时间即EGT (SCnEGT[7:0])，只影响ISO 7816-3接口传输的数据，格式如图6.10-11所示。

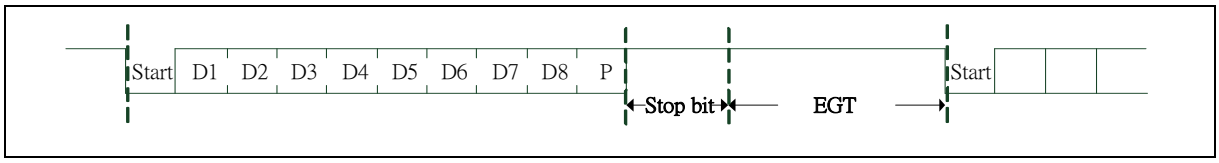


图 6.10-11 额外保护时间操作

6.10.6 寄存器描述

SCnCR0 – SC 控制寄存器 0

寄存器	特殊功能寄存器地址	复位值
SC0CR0	F1H, SFR页2	0000_0000 b
SC1CR0	F3H, SFR页2	0000_0000 b
SC2CR0	F5H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
NSB	**T**	**RXBGTEN**	**CONSEL**	**AUTOCEN**	**TXOFF**	**RXOFF**	**SCEN**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**NSB**	**停止位长度** 此字段指示停止位的长度。 0 = 停止位长度为 2 ETU。 1 = 停止位长度为 1 ETU。 **注意：**默认停止位长度为 2。SC 和 UART 采用 NSB 编程设置停止位长度。
6	**T**	**T模式** 0 = T0 (ISO 7816-3 T = 0 模式). 1 = T1 (ISO 7816-3 T = 1 模式). T 模式控制 BGT（区块保护时间）。区块保护时间是指不同传输方向之间两个连续字符前沿之间的最小位长度。此字段指示区块保护时间位长度的计数器。根据 ISO 7816-3，在 T = 0 模式下，软件必须将 T 位清零，以获得实际区块保护时间 = 16.5。在 T = 1 模式下，软件必须将 T 位设置为 1，以获得实际区块保护时间 = 22.5。 **注意：**在 T = 0 模式下，如果检测到奇偶校验错误，接收器将在 PEF(SC0TSR[4]) 上生成奇偶校验标志，并将奇偶校验错误信号驱动至收发器。在 T = 1 模式下，如果检测到奇偶校验错误，接收器将在 PEF(SC0TSR[4]) 上生成奇偶校验标志，但不会将奇偶校验错误信号驱动至收发器。 **注意：**具体说明请参见章节6.10.5.4 错误信号和字符重复
5	**RXBGTEN**	**接收端区块保护时间功能使能位** 0 = 接收端块保护时间功能禁用。 1 = 接收端块保护时间功能启用。
4	**CONSEL**	**约定选择** 0 = 正相约定。 1 = 反相约定。 **注1:** 如果 AUTOCEN(SCnCR0[3]) 写入“1”，此位自动清零。 **注2:** 如果 AUTOCEN(SCnCR0[3]) 使能，硬件将在 SCEN (SCnCR0[0]) =“1”后自动决定约定并更改 CONSEL (SCnCR0[4]) 位。

位	名称	描述
3	**AUTOCEN**	**自动约定使能位** 0 = 自动约定禁用。 1 = 自动约定使能。当硬件在复位应答状态接收到 TS 时，如果该 TS 为正约定，则 CONSEL (SCnCR0[4]) 将自动设置为 0；否则，如果该 TS 为逆约定，则 CONSEL (SCnCR0[4]) 将设置为 1。 **注意：**如果软件启用自动约定功能，则必须在复位应答状态之前完成设置，且第一个数据必须为 0x3B 或 0x3F。硬件接收到第一个数据并将其存储在缓冲区后，硬件将决定转换方式并自动更改 CONSEL（SCnCR0[4]）位。
2	**TXOFF**	**TX 传输禁用位** 0 = 收发器启用。 1 = 收发器禁用。
1	**RXOFF**	**RX传输禁用位** 0 = 接收器启用。 1 = 接收器禁用。 **注意：**如果 AUTOCEN（SCnCR0[3]）已启用，则必须忽略这些字段。
0	**SCEN**	**SC引擎启用位** 将此位设置为 1 以启用 SC 操作。如果清除此位，SC 将强制所有转换到 IDLE 状态。 **注意：**在填充其他寄存器之前，必须将 SCEN 设置为 1，否则 ISO 7816-3 将无法正常工作。

SCnCR1 – SC 控制寄存器

寄存器	特殊功能寄存器地址	复位值
SC0CR1	F2H, SFR页2	0000_0000 b
SC1CR1	F4H, SFR页2	0000_0000 b
SC2CR1	F6H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
OPE	PBOFF	WLS		TXDMAEN	RXDMAEN	CLKKEEP	UARTEN
读/写	读/写	读/写		读/写	读/写	读/写	读/写

位	名称	描述
7	OPE	**奇校验使能位** 0 = 接收模式下，发送偶数个逻辑1，或检查数据字和奇偶校验位。 1 = 接收模式下，发送奇数个逻辑1，或检查数据字和奇偶校验位。 **注意：** 此位仅在PBOFF位为0时有效。
6	PBOFF	**奇偶校验位禁用控制** 0 = 在串行数据的“最后一个数据字位”和“停止位”之间生成或检查奇偶校验位。 1 = 传输过程中不生成奇偶校验位（发送数据时）或检查奇偶校验位（接收数据时）。 **注意：** 在ISO 7816-3模式下，此字段必须为0（默认设置带奇偶校验位）。
5:4	WLS	**字长选择** 00 = 字长为8位。 01 = 字长为7位。 10 = 字长为6位。 11 = 字长为5位。 **注意：** 在 ISO 7816-3 模式下，此 WLS 必须为 00。
3	TXDMAEN	**SC/UART TX DMA 使能** 此位使能 SC/UART TX 通过 PDMA 传输进行操作，在 SC/UART TX 启动之前，TX 数据需要在 XRAM 中准备好。 0 = SPI TX DMA 禁用 1 = SPI TX DMA 使能
2	RXDMAEN	**SC/UART RX DMA 使能** 此位使能 SC/UART RX 通过 PDMA 传输进行操作，在 SC/UART RX 操作后，RX 数据保存在 XRAM 中。 0 = SC/UART RX DMA 禁用 1 = SC/UART RX DMA 使能
1	CLKKEEP	**SC 时钟使能位** 0 = SC 时钟生成禁用。 1 = SC 时钟始终保持自由运行。

位	名称	描述
0	**UARTEN**	**UART 模式使能位** 0 = ISO 7816-3 模式. 1 = UART 模式 **注 1：**在 UART 模式下工作时，用户必须设置 CONSEL (SCnCR0[4]) = 0 且 AUTOCEN(SCnCR0[3]) = 0。 **注 2：**在 ISO 7816-3 模式下工作时，用户必须设置 UARTEN(SCnCR1 [0]) = 0。 **注 3：**当 UART 使能时，硬件将产生复位信号以复位 FIFO 和内部状态机。

SCnDR – SC 数据寄存器

寄存器	特殊功能寄存器地址	复位值
SC0DR	D9H, SFR页2	0000_0000 b
SC1DR	E1H, SFR页2	0000_0000 b
SC2DR	E9H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
SCnDR[7:0]							
读/写							

位	名称	描述
7:0	SCnDR[7:0]	**SC / UART缓冲区数据** 此字节用于在SC / UART总线上发送或接收数据。写入此字节实际上是写入移位寄存器。读取此字节实际上是读取读数据缓冲区。 **注意：** 如果SCEN(SCnCR0[0]) 未使能，则SCnDR 不能被编程。

SCnEGT – SC 额外保护时间寄存器

寄存器	特殊功能寄存器地址	复位值
SC0EGT	DAH, SFR页2	0000_0000 b
SC1EGT	E2H, SFR页2	0000_0000 b
SC2EGT	EAH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
SCnEGT[7:0]							
读/写							

位	名称	描述
7:0	SCnEGT[7:0]	**SC额外保护时间** 此字段表示额外保护定时器的值。 **注意：** 该计数器以 ETU 为基数。

SCnETURD0 – SCn ETU 速率分频器寄存器

寄存器	特殊功能寄存器地址	复位值
SC0ETURD0	DBH, SFR页2	0111_0011 b
SC1ETURD0	E3H, SFR页2	0111_0011 b
SC2ETURD0	EBH, SFR页2	0111_0011 b

7	6	5	4	3	2	1	0
ETURDIV[7:0]							
读/写							

位	名称	描述
7:0	**ETURDIV[7:0]**	**ETU 速率分频器的 LSB位** 该字段指示时钟速率分频器的 LSB。 实际 ETU = ETURDIV[11:0] + 1. **注1:** ETURDIV[11:0] 必须大于 0x004. **注2:** 必须先编程SCnETURD0，然后再编程 SCnETUDR2。

SCnETURD1 –SC ETU 速率分频器寄存器 1

寄存器	特殊功能寄存器地址	复位值
SC0ETURD1	DCH, SFR页2	0011_0001 b
SC1ETURD1	E4H, SFR页2	0011_0001 b
SC2ETURD1	ECH, SFR页2	0011_0001 b

7	6	5	4	3	2	1	0
-	SCDIV[2:0]			ETURDIV[11:8]			
-	读/写			读/写			

位	名称	描述
7	-	保留
6:4	SCDIV [2:0]	**SC 时钟分频器** 000 = F_{SC}为F_{SYS}/1. 001 = F_{SC}为F_{SYS}/2. 010 = F_{SC}为F_{SYS}/4. 011 = F_{SC}为F_{SYS}/8. (默认) 100 = F_{SC}为F_{SYS}/16. 101 = F_{SC}为F_{SYS}/16. 110 = F_{SC}为F_{SYS}/16. 111 = F_{SC}为F_{SYS}/16. **注意：**根据ISO/IEC 7816-3标准，F_{SC}时钟应为1Mhz ~ 5Mhz
3:0	ETURDIV [11:8]	**ETU 速率分频器的 MSB 位** 该字段表示时钟速率分频器的 MSB。 实际 ETU为ETURDIV[11:0] + 1. **注1:** ETURDIV[11:0] 必须大于 0x004. **注2:** 必须先编程SCnETURD0，然后再编程 SCnETUDR2。

ScnIE – SC 中断使能控制寄存器

寄存器	特殊功能寄存器地址	复位值
SC0IE	DCH, SFR页2	0000_0000 b
SC1IE	E4H, SFR页2	0000_0000 b
SC2IE	ECH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN
-	-	-	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:4	-	保留
4	ACERRIEN	**自动约定错误中断使能位** 此字段用于使能自动约定错误中断。 0 = 自动约定错误中断禁用。 1 = 自动约定错误中断使能。
3	BGTIEN	**区块保护时间中断使能位** 此字段用于使能区块保护时间中断。 0 = 区块保护时间中断禁用。 1 = 区块保护时间中断使能。
2	TERRIEN	**传输错误中断使能位** 此字段用于使能传输错误中断。传输错误状态位于SC0TSR 寄存器中，包括接收器中断错误BEF(SC0TSR[6])，帧错误FEF(SC0TSR[5])，奇偶校验错误PEF(SC0TSR[4])，接收器缓冲区溢出错误RXOV(SC0TSR[0]) 和发送缓冲区溢出错误TXOV(SC0TSR[2]). 0 = 传输错误中断禁用。 1 = 传输错误中断使能。
1	TBEIEN	**发送缓冲区空中断使能位** 此字段用于使能发送缓冲区空中断。 0 = 禁用发送缓冲区空中断。 1 = 启用发送缓冲区空中断。
0	RDAIEN	**接收数据到达中断使能位** 此字段用于使能接收数据中断。 0 = 禁用接收数据中断。 1 = 启用接收数据中断。

ScnIS – SC中断状态寄存器

寄存器	特殊功能寄存器地址	复位值
SC0IS	DEH, SFR页2	0000_0010 b
SC1IS	E6H, SFR页2	0000_0010 b
SC2IS	EEH, SFR页2	0000_0010 b

7	6	5	4	3	2	1	0
-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF
-	-	-	读/写	读/写	只读	只读	只读

位	名称	描述
7:5	-	保留
4	ACERRIF	**自动约定错误中断状态标志** 此字段指示自动约定序列错误。如果在 ATR 状态下接收到的 TS 既不是 0x3B 也不是 0x3F，则此位只能由硬件置位。 **注意：** 写入 0 即可清除。
3	BGTIF	**区块保护时间中断状态标志** 此字段用于指示区块保护时间中断状态，此位只能由硬件置位。 **注1:** 当 RXBGTEN (SCnCR0[5]) 使能时，此位有效。 **注2:** 写入 0 即可清除。
2	TERRIF	**传输错误中断状态标志（只读）** 此字段用于指示传输错误中断状态。传输错误状态位于 SC0TSR 寄存器中，包括接收器中断错误BEF(SC0TSR[6]) ，帧错误FEF(SC0TSR[5])，奇偶校验错误PEF(SC0TSR[4])，接收器缓冲区溢出错误RXOV(SC0TSR[0]) 和发送缓冲区溢出错误TXOV(SC0TSR[2]). **注意：** 此字段是BEF(SC0TSR[6]), FEF(SC0TSR[5]), PEF(SC0TSR[4]), RXOV(SC0TSR[0]) 和TXOV(SC0TSR[2]) 的状态标志。因此，如果软件想要清除此位，必须向每个字段写入“0”。
1	TBEIF	**发送缓冲区空中断状态标志（只读）** 此字段用于发送缓冲区空中断状态标志。 **注意：** 此字段是发送缓冲区空状态的状态标志。如果软件要清除此位，必须将数据写入DAT(SCnDR[7:0]) 缓冲区，然后此位将自动清除。
0	RDAIF	**接收数据到达中断状态标志（只读）** 此字段用于接收数据中断状态标志。 **注意：** 此字段是接收数据的状态标志。如果软件从 SC_DAT 引脚读取数据，此位将自动清除。

SCnTSR – SC 传输状态寄存器

寄存器	特殊功能寄存器地址	复位值
SC0TSR	DFH, SFR页2	0000_1010 b
SC1TSR	E7H, SFR页2	0000_1010 b
SC2TSR	EFH, SFR页2	0000_1010 b

7	6	5	4	3	2	1	0
ACT	**BEF**	**FEF**	**PEF**	**TXEMPTY**	**TXOV**	**RXEMPTY**	**RXOV**
只读	读/写	读/写	读/写	只读	读/写	只读	读/写

位	名称	描述
7	**ACT**	**发送/接收活动状态标志（只读）** 0 = 发送/接收传输完成后，此位自动清零 1 = 发送/接收传输处于活动状态时，此位由硬件置位
6	**BEF**	**接收器间隔错误状态标志** 仅当接收数据输入 (RX) 保持在“间隔状态”（逻辑 0）的时间长于一个完整字的传输时间（即“起始位”+数据位+奇偶校验位+停止位的总时间）时，此位才会设置为逻辑 1。 **注意：** 此位是只读的，但可以通过写入 0 来清除。
5	**FET**	**接收器帧错误状态标志** 仅当接收到的字符没有有效的“停止位”（即，检测到最后一个数据位或奇偶校验位后的停止位为逻辑0）时，此位才会设置为逻辑1。 **注意：** 此位是只读的，但可以通过写入0来清除。
4	**PEF**	**接收器奇偶校验错误状态标志** 仅当接收到的字符没有有效的“奇偶校验位”时，此位才会设置为逻辑1。 **注意：** 此位是只读的，但可以通过写入0来清除。
3	**TXEMPTY**	**发送缓冲区空状态标志（只读）** 此位指示发送缓冲区是否为空。 **注意：** 当发送缓冲区已传输至发送移位寄存器时，硬件会将此位置1。将数据写入 DAT(SCnDR[7:0])（发送缓冲区非空）时，此位置1。
2	**TXOVF**	**TX 溢出错误中断状态标志** 如果 TX 缓冲区已满，再次写入 DAT(SCnDR[7:0]) 将导致该位仅由硬件设置为“1”。 **注意：** 该位为只读，但可以通过写入 0 来清除。
1	**RXEMPTY**	**接收缓冲区空状态标志（只读）** 该位指示 RX 缓冲区是否为空。 **注意：** 当 CPU 读取 Rx 缓冲区后，硬件会将该位置高。当 SC 接收到任何新数据时，该位置高。

位	名称	描述
0	**RXOV**	**RX 溢出错误状态标志** 该位仅在 RX 缓冲区溢出时设置。 **注意：** 该位为只读，但可以通过写入 0 来清除。

AUXR2 – 辅助寄存器 2

寄存器	特殊功能寄存器地址	复位值
AUXR2	A1H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
UART2TXP		UART2RXP		UART1TXP		UART1RXP	
读/写		读/写		读/写		读/写	

位	名称	描述
7:6	**UART2TXP**	**UART2 TX 引脚选择** 00 = 默认保留 01 = 将UART2 TX分配给P0.3 10 = 将UART2 TX分配给P3.0 11 = 保留
5:4	**UART2RXP**	**UART2 RX引脚选择** 00 = 默认保留 01 = 将UART2 RX分配给 P0.4 10 = 将UART2 RX 分配给 P1.7 11 = 保留

AUXR3 – 辅助寄存器 3

寄存器	特殊功能寄存器地址	复位值
AUXR3	A2H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
UART4TXP		UART4RXP		UART3TXP		UART3RXP	
读/写		读/写		读/写		读/写	

位	名称	描述
7:6	**UART4TXP**	**UART4 TX 引脚选择** 00 = 默认保留 01 = 将UART4 TX分配给P2.3 10 = 保留 11 = 保留
5:4	**UART4RXP**	**UART4 RX 引脚选择** 00 = 默认保留 01 = 将UART4 RX分配给P2.2 10 = 保留 11 = 保留
3:2	**UART3TXP**	**UART3 TX 引脚选择** 00 = 默认保留 01 = 将UART3 TX 分配给P1.2 10 = Assign UART3 TX 分配给 P1.5 11 = Assign UART3 TX 分配给 P0.5
1:0	**UART3RXP**	**UART3 RX 引脚选择** 00 = 默认保留 01 = 将UART3 RX 分配给 P1.1 10 = 将UART3 RX 分配给 P2.5 11 = 将UART3 RX 分配给 P3.4

6.11 内部集成电路 (I²C)

6.11.1 概述

CM1003 系列提供两条内部集成电路 (I²C) 总线，用作微控制器与 I²C 设备（例如 EEPROM、LCD 模块、温度传感器等）之间的串行接口。I²C 总线采用双线设计（一条串行数据线 I2C0_SDA 和一条串行时钟线 I2C0_SCL）在设备之间传输信息。

I²C 总线在主机和从机之间使用双向数据传输。它没有中央主机，通过同时传输数据的主机之间的仲裁来实现多主机系统。串行时钟同步允许不同比特率的设备通过一条串行总线进行通信。I²C 总线支持四种传输模式，包括主机发送、主机接收、从机接收和从机发送。I²C 接口仅支持 7 位寻址模式。此外，还提供特殊模式“广播呼叫”。I²C 支持标准速率传输（高达 100kbps）和快速传输（高达 400kbps）。

6.11.2 功能描述

对于双向传输操作，I2C0_SDA及I2C0_SCL引脚应为开漏焊盘。实现了逻辑线与功能，这对于接口的操作至关重要。当一个或多个I²C设备输出“0”时，I²C总线上会产生低电平。当所有I²C设备全输出1，总线上才是高电平，允许上拉电阻把电平拉高。在CM1003中，用户应在通过设置I2CEN使能I²C功能之前，将I2C0_SCL和I2C0_SDA的输出锁存设置为逻辑1。

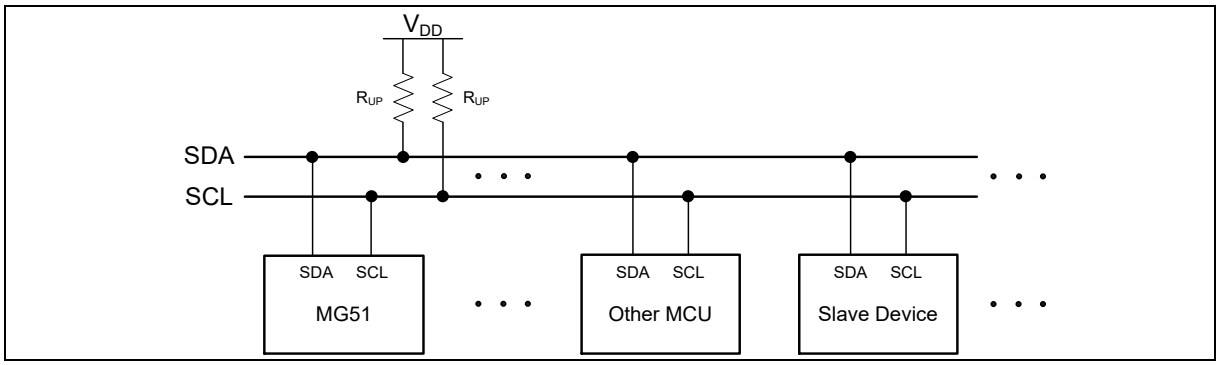


图 6.11-1 I²C 总线互联

当两条线都处于高电平时，I²C被视为空闲。这时任一设备都可以做为主机占用总线，并在生成START条件后开始一次数据传输。在发送STOP停止条件结束传输前，总线被认为处于忙状态。主机产生所有串行时钟脉冲以及START和STOP条件。如果总线上没有START条件，则所有总线设备都充当未被寻址的从机，硬件会查找自己的从机地址或广播呼叫地址，(广播地址检测可由 GC(I2CnADDRx.0)使能或禁止)。若收到匹配的地址，则请求中断。

I²C总线上的每笔传输事务长度为9位，由8个数据位(MSB在前)和1个应答位组成。每次传输的字节数(定义为有效 START 和 STOP 条件之间的时间)不受限制，但每个字节后面必须有一个应答位。主机产生8个时钟脉冲来发送8位数据。在I2C0_SCL线的第8个下降沿之后，在I2C0_SDA上输出数据的设备将该引脚更改为输入，并在第9个时钟脉冲上读取应答值。在第9个时钟脉冲之后，如果下一个接收操作尚未准备好，数据接收端可以拉住I2C0_SCL线保持为低电平，强制暂停下一个字节事务。当接收端释放I2C0_SCL线时，数据传输将继续。

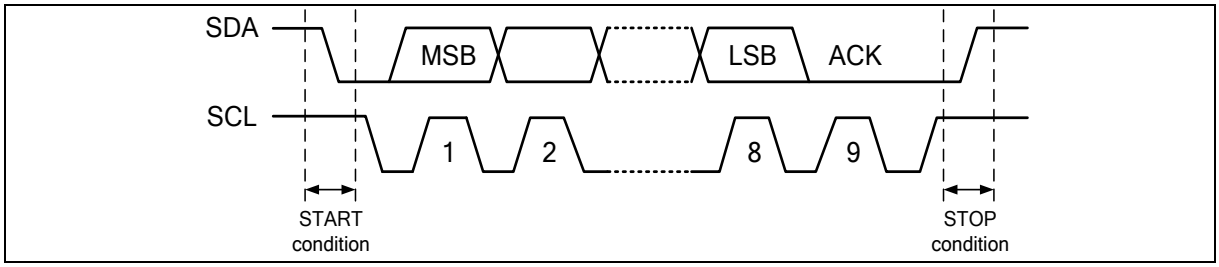


图 6.11-2 I²C 总线协议

6.11.2.1 起始和停止条件

I²C总线协议定义了两种状态来开始和结束传输：起始START(S)和停止STOP(P)条件。起始条件定义为I2C0_SCL线为高电平时I2C0_SDA线上的电平从高到低的转换。停止条件的定义是I2C0_SCL线为高电平时I2C0_SDA 线上的电平从低到高的转换。起始或停止条件都由主机生成，起始条件后I²C总线被视为忙，停止条件后I²C总线被视为空闲。成功发出停止条件后，原主机将释放总线控制权并恢复为未寻址的从机。因此，原寻址的从机将成为未寻址的从机。I²C总线处于空闲状态并监听下一次传输的起始条件。

数据传输始终由主机生成的停止条件终止。但是，如果主机仍希望在总线上通信，它可以生成repeated START重复起始条件，并寻址前一个或另一个从机继续通信，而无需先生成停止条件。在这样的传输中，就可以实现各种读/写格式的组合。

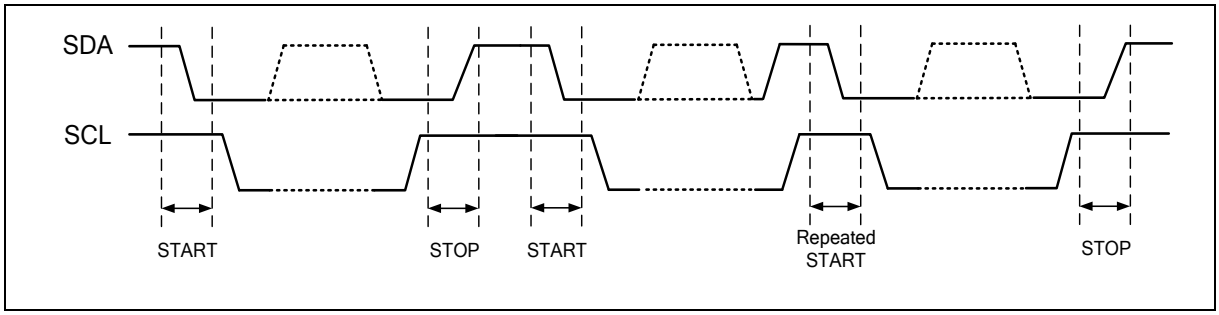


图 6.11-3 START 起始条件, Repeated START 重复起始条件和 STOP 停止条件

6.11.2.2 7位地址数据格式

生成START条件后，主机应传输一个字节的特殊数据。该数据包括一个 7 位长的从机地址 (SLA)+第8位数据方向位(读/写)”，用以寻址目标从机设备并确定数据流的方向。如果 读/写 位为0，则表示主机将向选定的从机写入信息。如果 读/写 位为1，则表示主机将从被寻址的从机读取信息。由从机地址和读取(R)或写入(W)位组成的地址包分别称为SLA+R或SLA+W。一次传输基本上由一个START起始条件，一个SLA+W/R、一个或多个数据包和一个STOP停止条件组成。通过 SLA+W/R 寻址到指定的从机后，主机或从机设备将根据读/写位的配置发出第二个及后续的 8 位数据字节。

图6.11-4 为主机以 7 位数据格式向从机发送数据。主机通过7位地址和1位写方向向从机发送寻址，以表示主机想要向从机发送数据。在从机返回一个应答信号后，主机将继续发送数据。

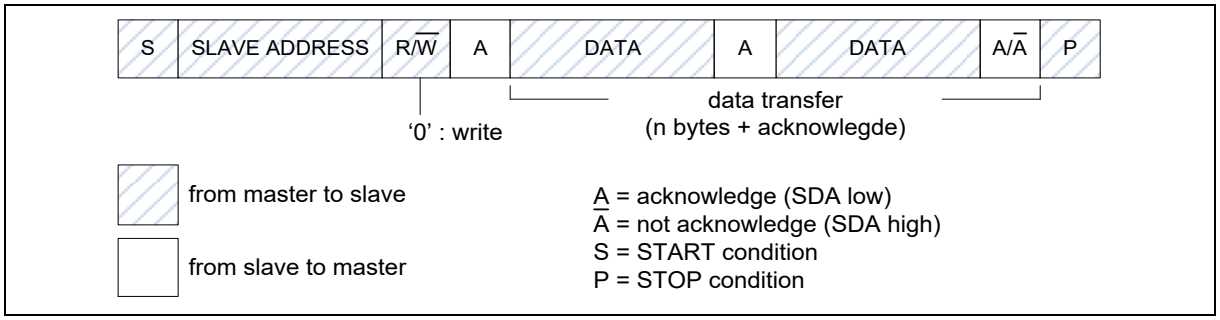


图 6.11-4 主机以 7 位数据格式向从机发送数据

图6.11-5 为主机以 7 位数据格式从机读取数据。主机使用7位地址和1位读方向来寻址从机，以表示主机想要从从机读取数据。从机向主机返回应答信号后，从机将开始传输数据。

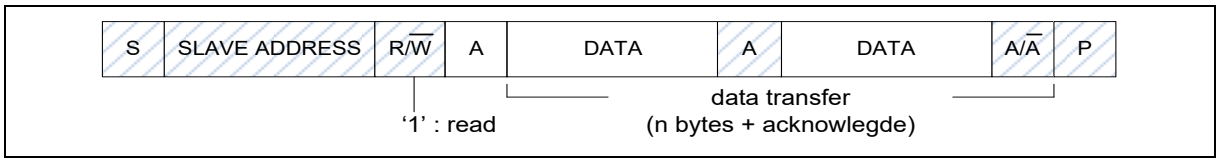


图 6.11-5 主机以 7 位数据格式向从机读取数据

有一种例外情况称为“广播呼叫”寻址，它可以通过给数据的第一个字节全0来寻址所有设备。当主机希望将相同的消息传输给系统中的多个从机时，使用广播呼叫。收到广播呼叫要不要发应答或忽略该消息由其他设备的软件配置决定。如果某个设备响应了广播呼叫，则其操作方式和标准从机接收模式相同。请注意，地址0x00是为广播呼叫保留的，不能用作从机地址，因此理论上，7位寻址I²C总线可以接受 127 个从机地址为 1 至 127 的设备。

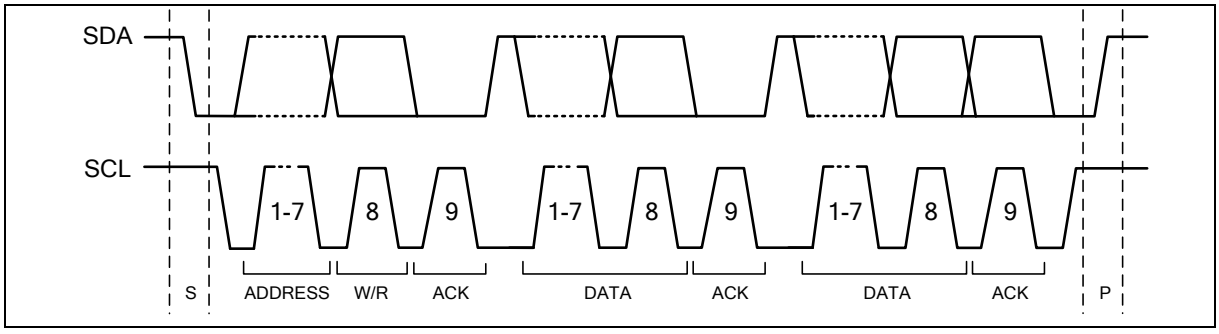


图 6.11-6 一次 I²C 传输的数据格式

在数据传输期间，I2C0_SDA线上的数据在时钟高电平期间应保持稳定，只有当I2C0_SCL为低电平时，数据线内容才能发生变化。

6.11.2.3 应答Ack

字节传输的第9个I2C0_SCL脉冲专用于应答(ACK)。它允许接收端(可以是主机或从机)通过将I2C0_SDA 线拉低来响应发送端(也可以是主机或从机)。应答位时钟脉冲由主机产生。发送端应在应答时钟脉冲期间释放对I2C0_SDA线的控制。ACK是一个低电平有效信号，在时钟脉冲高电平期间将I2C0_SDA线拉低，向发送端表明已收到传输的数据。通常，已寻址的从机需要在收到每个字节后生成一个ACK。当该从机未应答(NACK)从机地址时，从机应将I2C0_SDA线保持高电平，以便主机可以生成STOP停止条件或repeated START重复起始条件。

如果从机接收端应答了从机地址，它将自行切换为未寻址从机模式，从而无法再接收任何数据字节。此从机会将I2C0_SDA线保持高电平。主机应生成STOP停止条件或repeated START重复起始条

件。

若是主机接收端参与传输，由于主机控制传输的字节数，因此它应通过不在最后一个字节生成应答来向从机发送端发出数据结束信号。然后，从机发射端会切换到非寻址模式并释放I2C0_SDA线，以允许主机生成STOP停止条件或repeated START重复起始条件。

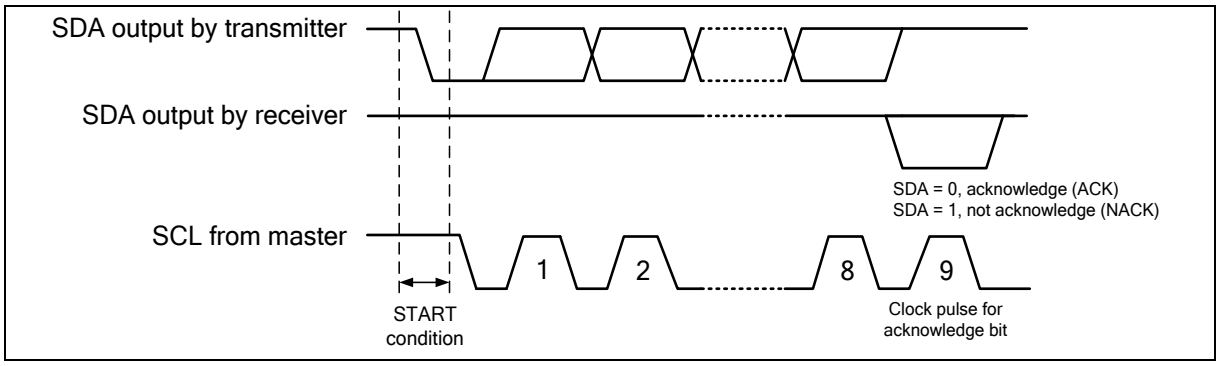


图 6.11-7 应答位

6.11.2.4 仲裁

主机只有在总线空闲时才能启动传输。可能有两个或多个主机生成START条件。在这种情况下，仲裁机制会在 I2C0_SDA 线上进行，此 I2C0_SCL 为高电平。仲裁期间，第一个竞争主机对I2C0_SDA置“a”1(高电平)，而另一个主机发送“a”0(低电平)。发送0的主机关闭其数据输出，因为总线上的电平与其自身电平不匹配。仲裁失败的主机立即切换到未寻址的从机状态，以在同一串行传输中检测其自身的从机地址是否被仲裁胜利的主机寻址。仲裁失败的主机还会释放I2C0_SDA数据线到高电平，以免影响仲裁胜利的主机继续进行的数据传输。但是，仲裁失败的主机继续在I2C0_SCL 线上生成时钟脉冲，直到仲裁失败的字节结束。

仲裁机制由所有主机在输出数据后持续监控I2C0_SDA线来执行。如果从I2C0_SDA线读取的值与主机刚输出的值不匹配，则主机仲裁失败。请注意，只有当主机输出I2C0_SDA高值而另一个主机输出低值时，主机才会仲裁失败。仲裁将持续进行，直到总线上只剩下一个主机，这可能需要很多位。仲裁的第一阶段是地址位比较，如果两个主机都试图寻址同一个设备，则仲裁将继续进行，直到比较数据位或确认位。

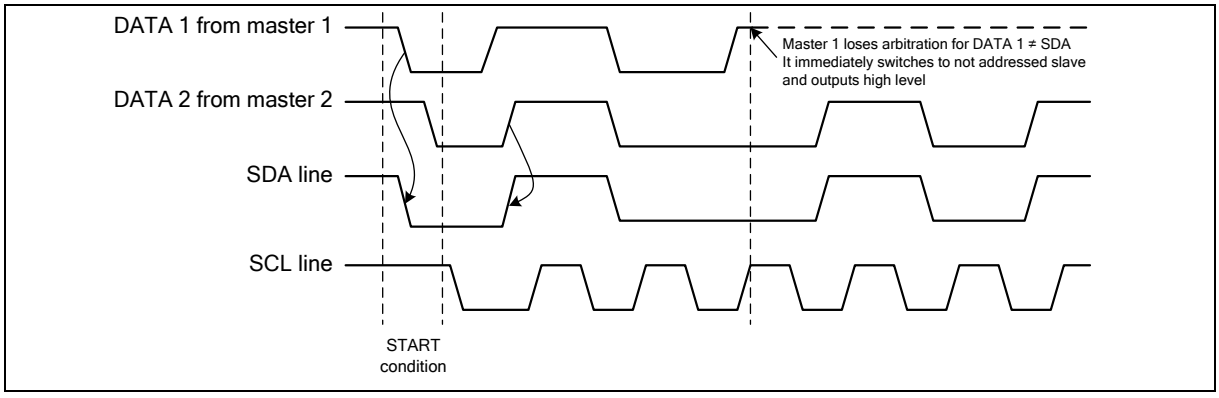


图 6.11-8 两台主机的仲裁过程

由于I²C总线的控制权完全取决于竞争主机发送的地址或主机代码和数据，因此总线上不存在中央主机，也没有任何优先级顺序。 从机不参与仲裁程序。

6.11.2.5 操作模式

片上I²C端口支持三种操作模式：主机模式，从机模式和广播呼叫模式。

在应用程序中，I²C端口可以作为主机或从机。从机模式下，I²C端口硬件寻找自己的从机地址和广播地址。如果检测到其中一个地址，并且从机愿意从主机接收或向主机发送数据(通过设置AA位)，则在第9个时钟上发送应答信号，因此如果中断使能，主机和从机都会请求中断。当微控制器希望成为总线主机时，硬件会等到总线空闲后再进入主机模式，以免中断可能的从机操作。如果在主机模式下总线仲裁失败，I²C端口会立即切换到从机模式，并能在同一串行传输中检测自己的从机地址。

为了控制每种模式下的I²C总线传输，用户需要根据I2C_STATUS0状态寄存器的当前状态码来设置I2C_CTL0和I2C_DAT寄存器。换句话说，对于每个I²C总线动作，用户需要通过I2C_STATUS0寄存器检查当前状态，然后设置I2C_CTL0和I2C_DAT寄存器以执行总线动作。最后通过I2C_STATUS0检查响应状态。

I2C_CTL0寄存器中的位STA、STO和AA用于在I2C_CTL0[3]寄存器的SI标志被清除后，控制I²C硬件的下一个状态。新操作完成后，I2C_STATUS0 寄存器中的状态码将被更新，并且I2C_CTL0 寄存器的SI标志位将被置位。但I²C停止时， SI标志不会被置位。如果设置了I²C中断控制位INTEN(I2C_CTL0[7])，则可以在中断服务程序中执行新状态码的相应操作或软件分支。

图6.11-9 显示当前I²C状态码为0x08，然后设置I2C_DATA=SLA+W和(STA,STO,SI,AA)=(0,0,1,x)，将地址发送到I²C总线。如果总线上有从机匹配地址并响应ACK，则I2C_STATUS0将更新为状态码0x18。

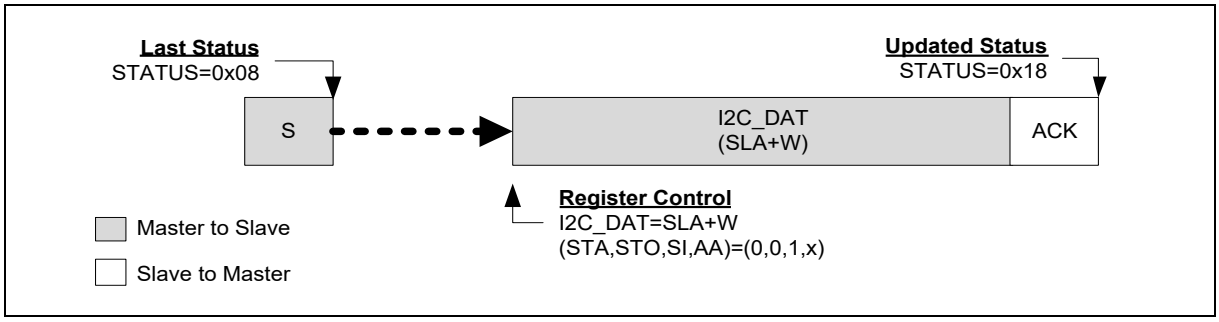


图 6.11-9 根据当前 I²C 状态来控制 I²C 总线

主机发送模式

主机发送模式下，会向从机接收端发送多个字节的数据。主机应在I2CnCLK中设置所需的时钟速率，以做好准备。通过将STA(I2CnCON.5)位设置为1来进入主机发送模式。硬件将检测总线，并在总线空闲时立即生成START起始条件。成功生成START条件后，SI标志(I2CnCON.3)将被置位，I2CnSTAT中的状态代码将显示08H。接下来向I2CDAT写入目标从机地址和数据方向位“写入”(SLA+W)，然后应清除SI位以开始SLA+W事务 。

SLA+W字节发送完毕且被寻址的从机返回应答(ACK)后，SI标志再次被置位，I2CnSTAT读取结果为18H。根据用户定义的通信协议，通过连续发送数据来采取相应的操作。所有数据发送完成后，主机可以通过置位STO(I2CnCON.4)并清除SI以发送STOP条件来结束传输，也可以生成repeated START重复起始条件而不发送STOP条件，立即启动另一次传输。

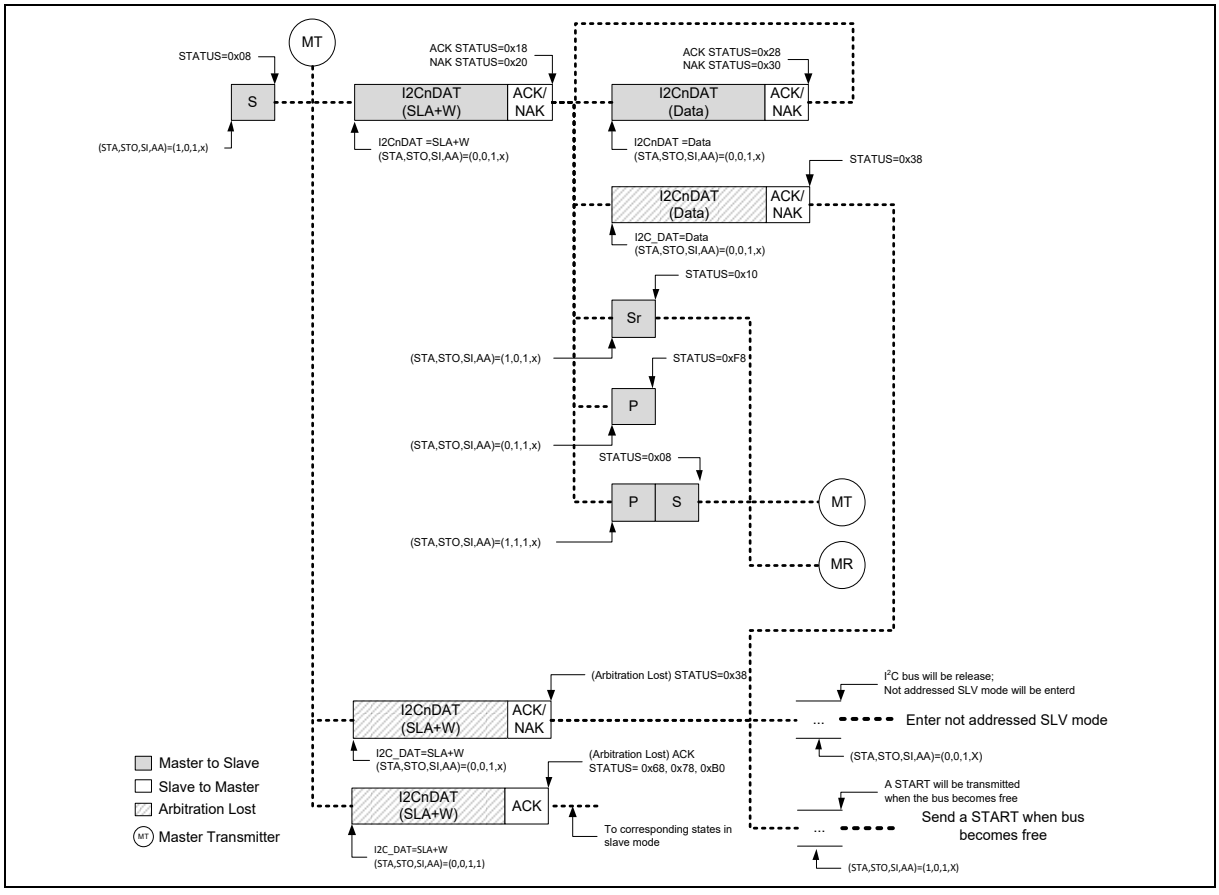


图 6.11-10 主机发送模式流程和状态

主机接收模式

在主机接收模式下，接收由从机发送的多个字节的数据。初始化方式与主机发送模式相同，主机发送START起始条件后，应向I2CnDAT写入目标从机地址和数据方向位“读取”(SLA+R)。发送 SLA+R 字节并收到从机应答位ACK后，SI标志再次被置1，状态寄存器I2CSTAT读取结果为40H。然后应清除 SI 标志开始接收从机数据，如果AA标志(I2CnCON.2)置位，主机接收端收到数据后回复应答位；如果AA标志清零，则主机接收端将不应答从机，并将从机释放为一个未寻址的从机。然后主机可以生成STOP停止条件或repeated START重复起始条件来停止传输或启动另一个传输。

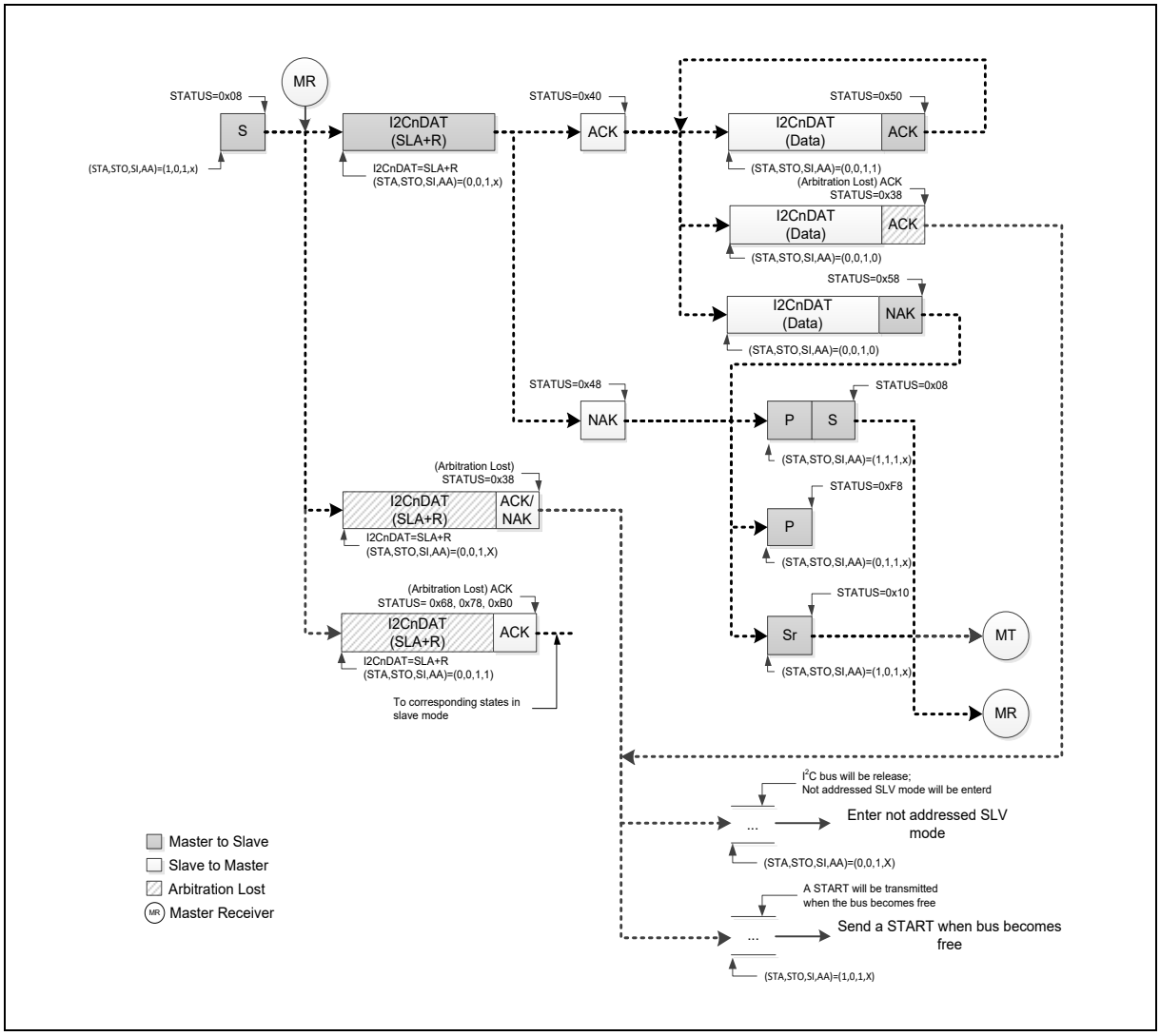


图 6.11-11 主机接收模式流程和状态

从机接收模式

从机接收模式下，会从主机发送端接收多个字节的数据。在开始传输前，应将设备在被主机寻址时响应的地址加载到I2CnADDRx中，I2CnCLK在从机模式下无效。应设置AA位以启用确认其自身的从机地址。完成上述初始化后，I²C处于空闲状态，直到被自身地址和数据方向位“写入”(SLA+W)寻址到。如果仲裁失败，也可以进入从机接收模式。

当从机被SLA+W寻址到后，应清除其SI标志位，以接收来自主机发送端的数据。如果在传输期间AA位为0，则从机将在下一字节返回非应答NACK，从机也将变为未寻址状态并与主机脱离。它无法接收任何数据字节，而I2CnDAT仍保留刚收到的上一个数据字节。

从机发送模式

I²C端口配备4个从机地址寄存器I2CnADDRx(x=0~3)。当I²C处于主机模式时，该寄存器的内容无效。在从机发送模式下，会向主机接收端发送几个字节的数据。在设置I2CnADDRx和I2CnCON 的值后，I²C会等待直到被自身地址和数据方向位“读取”(SLA+R)寻址到。如果仲裁失败，也可以进入从机发送模式。

当从机被“SLA+R寻址到，应清除其SI标志位以向主机发送数据。通常主机接收端会在从机发送完每个字节的数据后返回一个应答位，如果没有接收到应答位，则继续进行传输，从机将发送全1数据。它变为未寻址。如果在传输期间清除AA标志，则从机将发送最后一个字节的数据，并在接下去的传输中发送全1数据，并且从机变为无寻址。

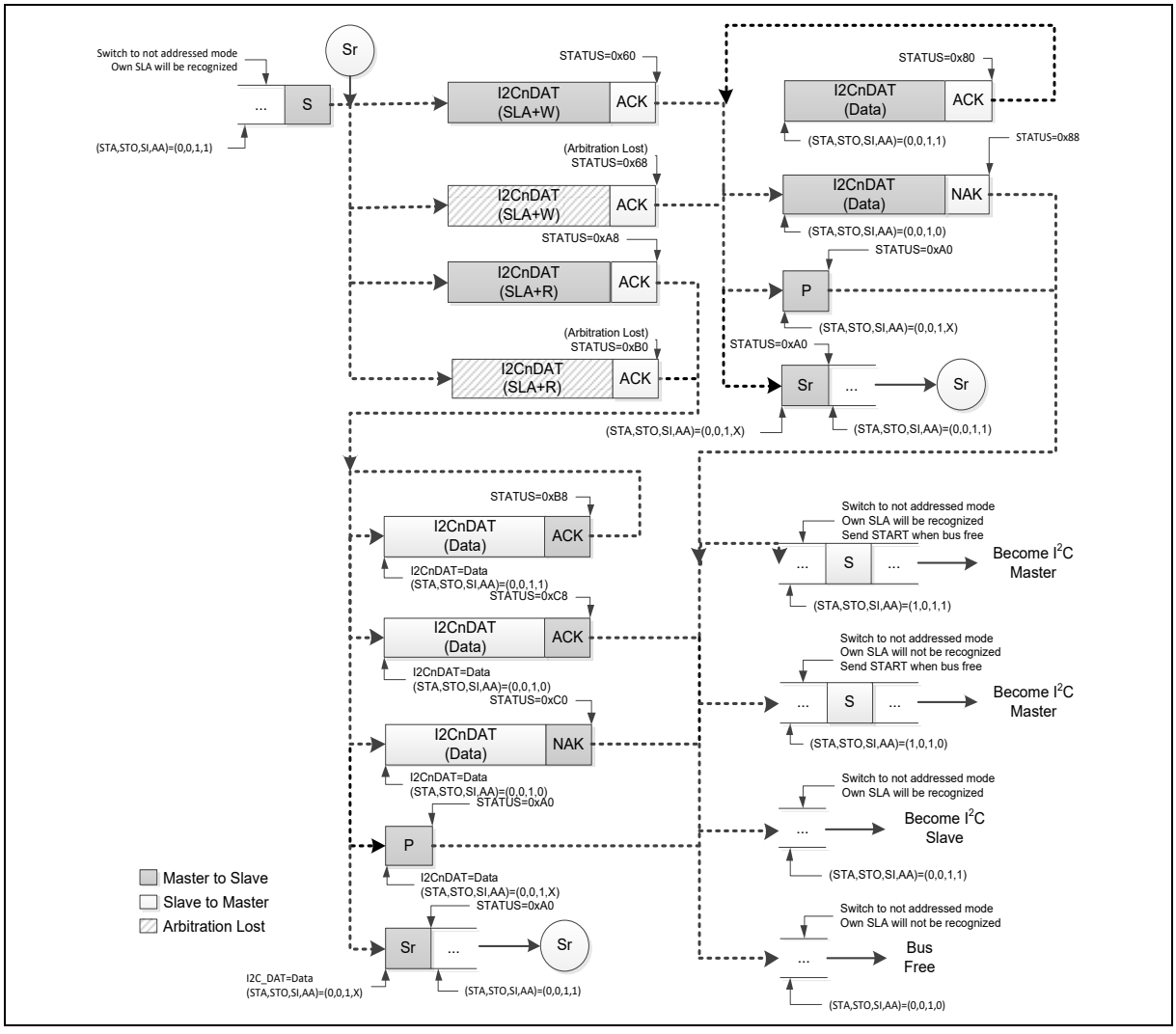


图 6.11-12 从机接收模式流程和状态

广播呼叫模式

广播呼叫模式是从机接收模式的一种特殊情况，通过从机地址和数据方向位全“0”寻址。GC(I2CnADDRx.0)位和AA位均应设置为1，使能广播呼叫的应答。通过广播呼叫寻址的从机在I2CnSTAT中的状态码与普通从机接收模式不同。如果仲裁失败，也可以进入广播呼叫模式。

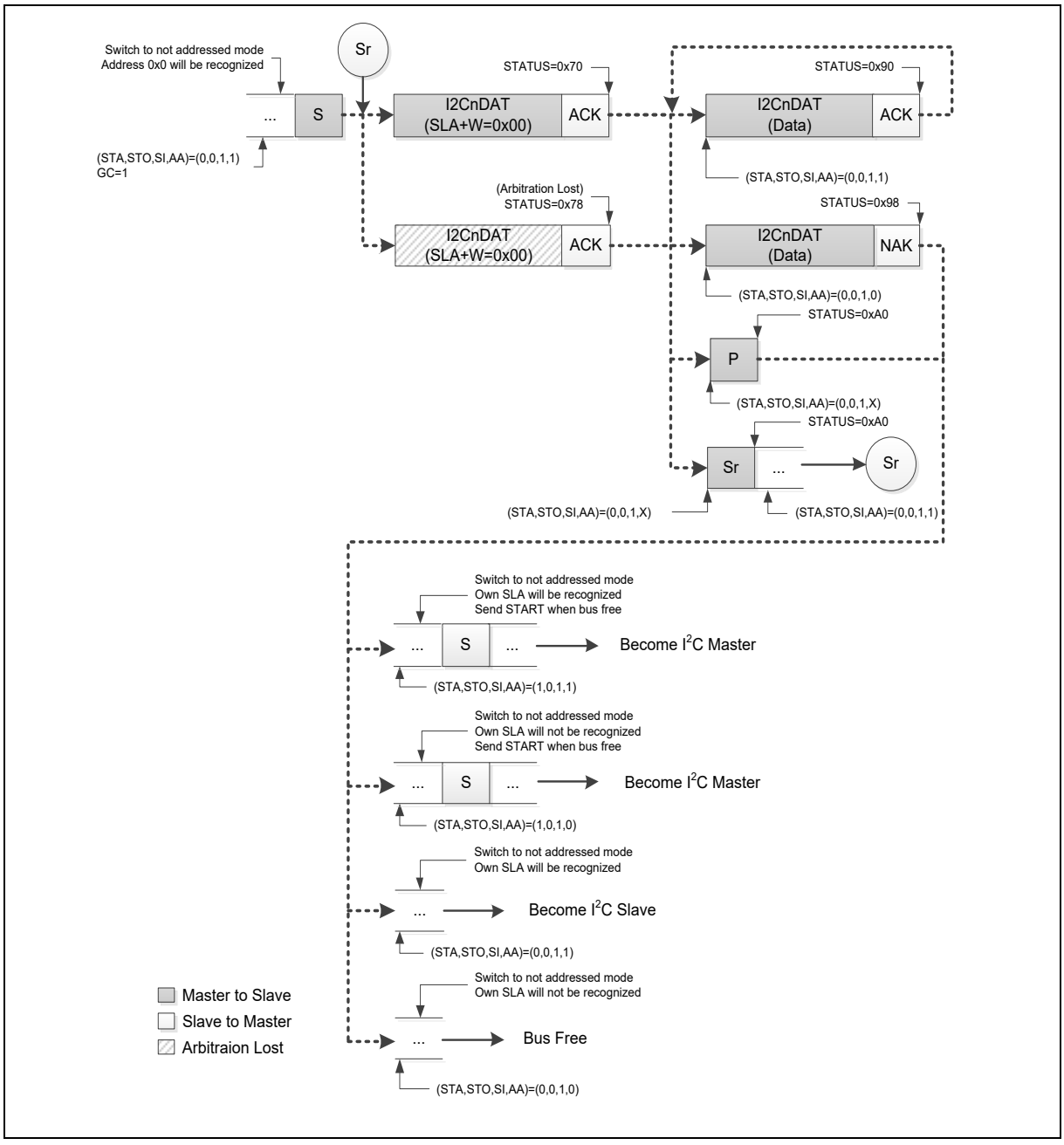


图 6.11-13 广播呼叫模式的流程和状态

6.11.2.6 其他状态

I2CnSTAT状态寄存器中,有两种状态码不属于之前所说的25种状态，第一个状态码F8H表示每次的传输中没有有效信息，同时SI标志位为0且无I²C中断产生。另一个状态代码00H表示在传输期间发生了总线错误。总线错误是指START起始条件或STOP停止条件暂时出现在非法位置，例如地址或数据的第二至第八位或应答位。发生总线错误时，SI标志会被立即置位。当在I²C总线上检测到总线错误时，操作设备会立即切换到未寻址从机位模式，释放I2C0_SDA和I2C0_SCL线，设置SI标志，并将I2CnSTAT清0。要从总线错误中恢复，应设置STO位并清除SI位。然后STO会被硬件清0，并释放总线，而无需在I²C总线上产生真正的STOP条件波形。

有一种特殊情况：如果I²C总线因I2C0_SDA 线被低电平阻塞（例如从机同步不一致），而无法成功生成 START起始条件或repeat START重复起始条件，则可以通过在I2C0_SCL线路上发送额外的时钟脉冲来解决此问题。当STA置位时，I²C硬件会发送额外的时钟脉冲，但由于I2C0_SDA线被拉低，因此无法生成START条件。当I2C0_SDA 线最终被释放时，会发送正常的START 条件，状态寄存器上会显示08H，串行传输继续。相同的状况，I2C0_SDA被低电平阻塞时，无法正常发送repeated START重复起始条件，I²C硬件也会执行与上述相同的操作。在这种情况下，成功发送起始条件后，将进入状态08H，而不是10H。请注意，软件不参与解决这些总线问题。

下表为I2STAT寄存器中I²C的状态编号及说明：

主机模式		从机模式	
状态	描述	状态	描述
0x08	起始条件	0Xa0	从机发送重复起始或停止条件
0x10	主机重复起始条件	0Xa8	从机发送地址应答
0x18	主机发送地址应答	0Xb0	从机发送仲裁失败
0x20	主机发送地址无应答	0Xb8	从机发送数据应答
0x28	主机发送数据应答	0Xc0	从机发送数据无应答
0x30	主机发送数据无应答	0Xc8	从机发送最后数据应答
0x38	主机仲裁失败	0x60	从机接收地址应答
0x40	主机接收地址应答	0x68	从机接收仲裁失败
0x48	主机接收地址无应答	0x80	从机接收数据应答
0x50	主机接收数据应答	0x88	从机接收数据无应答
0x58	主机接收数据无应答	0x70	广播呼叫模式地址应答
0x00	总线错误	0x78	广播呼叫模式仲裁失败
		0x90	广播呼叫模式数剧应答
		0x98	广播呼叫模式数据无应答
0xF8	总线释放 **注意:** 状态“0xF8 ”在主/从机模式下都存在，并且不会引发中断。		

6.11.2.7 I²C超时

有一个14位超时计数器，可用于处理I²C总线挂起。如果超时计数器使能，计数器开始计数直到溢出，此时I2TOF将由硬件置位并请求I²C中断。当超时计数器使能时，将标志位SI设置为高可重置计数器，并在SI清零後重新开始计数。如果I²C总线挂起，会导致SI标志位一段时间内不能置位。14位超时计数器

将溢出并请求中断服务。

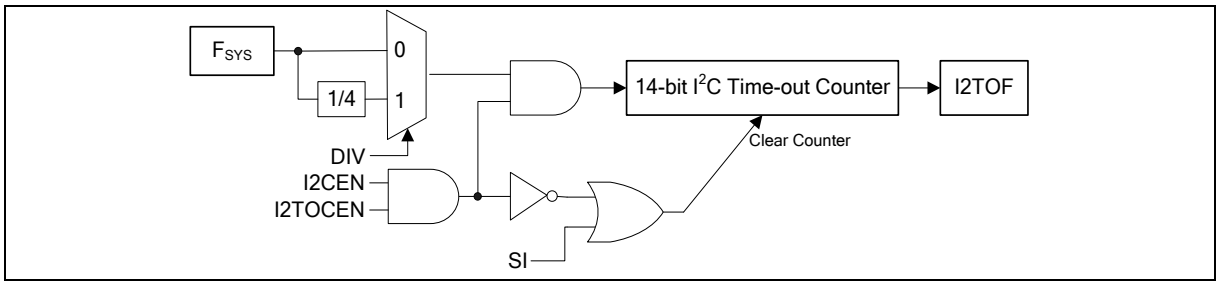


圖 6.11-14 I²C 超时计数器

6.11.2.8 I²C中断

I²C 有两个标志位，SI和I2TOF。它们都可以产生I²C事件中断请求。如果通过将EI2C和EA设置为1来使能I²C中断屏蔽，则只要这两个标志位中的任何一个置位，CPU就会执行I²C中断服务程序。用户需要检查标志位以确定导致中断发生的事件。这两个I²C标志位都由软件清除。

6.11.2.9 I²C中断服务程序的典型结构

以下是KEIL™ C51编译器下用C语言例程展示了I²C中断服务程序的典型结构，包括26个状态服务程序，可用作客户应用程序的基础。用户可以参考或修改例程，用于自己的应用程序。如果五种模式中的一种或多种未使用，则可以删除相关的状态服务程序，但应注意，不能调用已删除的程序。

```
Void I2C_ISR (void) interrupt 6
{
 switch (I2STAT)
 {
    //================================================
    //总线错误，始终放入ISR中进行毛刺处理
    //================================================
 case 0x00:              /*00H, 总线错误发生*/
      STO = 1;          //从总线错误中恢复
      break;
    //============
    //主机模式
    //============
 case 0x08:              /*08H, 一个起始条件START被发送*/
      STA = 0;          //STA位应由软件清零
      I2DAT = SLA_ADDR1;    //加载SLA+W/R
      break;
 case 0x10:            /*10H, 一个重复起始条件repeated START被发送*/
      STA = 0;
      I2DAT = SLA_ADDR2;
      break;
    //========================
    //主机发送模式
    //========================
 case 0x18:          /*18H, SLA+W 被发送, ACK 被接收*/
      I2DAT = NEXT_SEND_DATA1;   //加载数据
      break;
 case 0x20:          /*20H, SLA+W 被发送, NACK 被接收*/
      STO = 1;       //传输STOP停止条件
      AA = 1;        //准备好 ACK 自己SLA+W/R 或广播呼叫
      break;
 case 0x28:          /*28H, 数据被发送, ACK 被接收*/
```

```
            if (Conti_TX_Data)    //如果继续发送数据
            I2DAT = NEXT_SEND_DATA2;
            else          //如果没有数据要发
            {
               STO = 1;
               AA = 1;
            }
            break;
      case 0x30:          /*30H, 数据被发送, NACK 被接收*/
            STO = 1;
            AA = 1;
            break;
         //===========
         //主机模式
         //===========
      case 0x38:           /*38H,仲裁失败*/
            STA = 1;       //如果总线空闲，重传START起始条件
            break;
         //====================
         //主机接收模式
         //====================
      case 0x40:          /*40H, SLA+R 被发送, ACK 被接收*/
            AA = 1;       //ACK next received DATA
            break;
      case 0x48:          /*48H, SLA+R 被发送, NACK 被接收*/
            STO = 1;
            AA = 1;
            break;
      case 0x50:          /*50H, 数据被接收, ACK 被发送*/
            DATA_RECEIVED1 = I2DAT; //存储接收到的数据
            if (To_RX_Last_Data1)   //如果收到最后一个数据
               AA = 0;    //不 ACK 下一个收到的数据
            else          //如果继续接收数据
               AA = 1;
            break;
      case 0x58:          /*58H, 数据被接收, NACK 被发送*/
            DATA_RECEIVED_LAST1 = I2DAT;
            STO = 1;
            AA = 1;
            break;
         //====================================
         //从机接收和广播呼叫模式
         //====================================
      case 0x60:          /*60H, 自己的 SLA+W 被接收, ACK 返回*/
            AA = 1;
            break;
      case 0x68:          /*68H, SLA+W/R 仲裁失败，收到自身 SLA+W, ACK 返回*/
            AA = 0;       //不 ACK 下一个接收到的数据
                          //仲裁失败
            STA = 1;     //如果总线空闲，重传START起始条件
            break;
       case 0x70:         /*70H, 广播呼叫被接收, ACK 返回*/
            AA = 1;
            break;
      case 0x78:          /*78H, SLA+W/R仲裁失败，广播呼叫被接收，ACK 返回*/
            AA = 0;
            STA = 1;
            break;
      case 0x80:          /*80H, 先前自己的 SLA+W,数据被接收, ACK 返回*/
            DATA_RECEIVED2 = I2DAT;
```

```
            if (To_RX_Last_Data2)
               AA = 0;
            else
               AA = 1;
            break;
      case 0x88:          /*88H, 先前自己的 SLA+W, 数据被接收,
                        NACK 返回, 进入未寻址的从机模式*/
            DATA_RECEIVED_LAST2 = I2DAT;
            AA = 1;       //等待 ACK 下一个主机寻址
            break;
      case 0x90:          /*90H, 先前广播呼叫, DATA 被接收, ACK 返回*/
            DATA_RECEIVED3 = I2DAT;
            if (To_RX_Last_Data3)
               AA = 0;
            else
               AA = 1;
            break;
      case 0x98:          /*98H, 先前广播呼叫, DATA 被接收,
                        NACK 返回, 进入未寻址的从机模式*/
            DATA_RECEIVED_LAST3 = I2DAT;
            AA = 1;
            break;
         //==========
         //从机模式
         //==========
      case 0Xa0:          /*A0H, STOP 停止条件或 repeated START重复起始条件被接收,
                        仍可被寻址的从机模式*/
            AA = 1;
            break;
         //======================
         //从机发送模式
         //======================
      case 0Xa8:          /*A8H, 自己 SLA+R 被接收, ACK 返回*/
            I2DAT = NEXT_SEND_DATA3;
            AA = 1;       //当 AA 是 "1", 不是最后一个要传输的数据
            break;
      case 0Xb0:          /*B0H, SLA+W/R 仲裁失败,
                        自己的 SLA+R 被接收, ACK 返回*/
            I2DAT = DUMMY_DATA;
            AA = 0;       //当AA 是"0", 最后一个被传输的数据
            STA = 1;      //如果总线空闲, 重传START起始条件
            break;
      case 0Xb8:          /*B8H, 先前自己 SLA+R, 数据被传输, ACK 返回*/
            I2DAT = NEXT_SEND_DATA4;
            if (To_TX_Last_Data) //如果传输最后一个数据
               AA = 0;
            else
               AA = 1;
            break;
      case 0Xc0:          /*C0H, 先前自己 SLA+R, 数据被传输,
                        NACK 返回, 进入未寻址的从机模式*/
            AA = 1;
            break;
      case 0Xc8:          /*C8H, 先前自己 SLA+R, 最后一个数据被传输
                         ACK 返回, 进入未寻址的从机模式*/
            AA = 1;
            break;
      }// switch结束 (I2STAT)

      SI = 0;             //SI应该是I²C ISR的最后一个命令
```

```
    while(STO);          //等待STOP停止条件或总线错误
                        //空闲, STO 由硬件清除
}
```

6.11.3 寄存器描述

I²C总线接口控制寄存器共有5个，包括I2CON、I2STAT、I2DAT、I2ADDR 和 I2CLK，提供协议控制，状态，数据发送和接收功能以及时钟速率配置。为了提高应用灵活性，可以通过I2CPX(I2CON.0)交换I2C0_SDA和I2C0_SCL引脚。以下寄存器与I²C功能相关。

I2CON – I²C控制

寄存器	特殊功能寄存器地址	复位值
I2CON	C0H, 所有SFR页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
I2CHTEX	**I2CEN**	**STA**	**STO**	**SI**	**AA**	**-**	**I2CPX**
读/写	读/写	读/写	读/写	读/写	读/写	-	读/写

位	名称	描述
7	**I2CHTEX**	**I2C0保持时间延长使能** 0 = I²C 数据传输到I2C0_SCL保持时间延长禁用. 1 = I²C 数据传输到I2C0_SCL保持时间延长使能，延长8个系统时钟。
6	**I2CEN**	**I²C总线使能** 0 = I²C总线禁用. 1 = I²C总线使能. 使能 I²C前, I2C0_SCL和I2C0_SDA口应被设为逻辑1.
5	**STA**	**起始标志START** 当STA置1，如果总线空闲，I²C产生START起始条件，如果总线忙，I²C等待STOP停止条件，随后产生START起始条件 如果I²C总线已经在主机模式且已发送或接收了一个或多个字节，此时再设定STA，I²C 总线将产生repeated START重复起始条件 注意：STA可在任何时间置位，甚至是从机模式。但在检测到 START 或repeated START重复起始条件后，STA 不会由硬件自动清除。用户需自行注意手动清除STA。
4	**STO**	**停止标志STOP** 当I²C 在主机模式下，STO置1时，会向总线发送STOP停止条件。一旦总线上STOP被侦测，STO由硬件自动清0。 STO标志置位也用于将I²C设备从总线错误状态(I2STAT为00H)中恢复。在这种情况下，不会向I²C总线发送STOP停止条件。 如果STA和STO同时置1，且在主机模式下，I²C总线将产生STOP之后紧接着产生START。如果在从机模式下，应避免STA及STO同时置1，以免发出非法的I²C帧。

位	名称	描述
3	**SI**	**I²C中断标志位** 进入26种可能的I²C状态之一(F8H 状态除外)时，硬件就会置位此位。SI置位后，软件应读取I2STAT寄存器，以确定已完成哪一步，并采取下一步骤。 SI由软件清0。在SI清零之前，I2C0_SCL低电平时间延长，事务将被暂停。这有助于从机处理先前的数据字节，直到准备好接收下一个字节。 串行事务暂停，直到SI 位被软件清0。SI清除后， I²C总线将继续生成START条件或重复起始条件，STOP条件，8位数据等等，具体取决于软件配置的控制字节或者位段设定。因此用户应在软件清除 SI 之前给寄存器合适的设定。
2	**AA**	**确认响应标志位** I²C设备为接收器或自身地址匹配的从机时 如果设置了 AA 标志，在 I2C0_SCL 线的确认时钟脉冲期间返回应答 ACK(I2C0_SDA 上的低电平)。 如果清除了 AA 标志，在I2C0_SCL 线的确认时钟脉冲期间返回非应答NACK(I2C0_SDA 上的高电平)。清除自身 AA 标志的设备将忽略其自身的从机地址和广播调用。因此，SI将不会被断言，也不会请求中断。 注意： 若已被寻址的从机，在从机接收模式下未返回ACK应答，或在从机发送模式下未接收到ACK应答，此从机将变为未寻址从机，无法接收任何数据直到其AA被置1，且主机再次对其进行寻址。 在从机发送模式下，I2STAT 值会变为 C8H，这是一种特殊情况。在从机向主机发送完最后一个数据字节之前，可以将 AA 标志位清零，这样在发送完最后一个数据字节后，从机就会主动切换到未寻址从机模式，断开与主机的连接。主机后续读取的数据将全部为 FFH。
1	-	保留
0	**I2CPX**	**I²C 引脚选择** 0 = 将I2C0_SCL分配给 P1.3，将I2C0_SDA分配给 P1.4。 1 = 将I2C0_SCL分配给 P0.2，将I2C0_SDA分配给 P1.6。 注意，一旦设置或清除此位，I²C 引脚将立即更换。

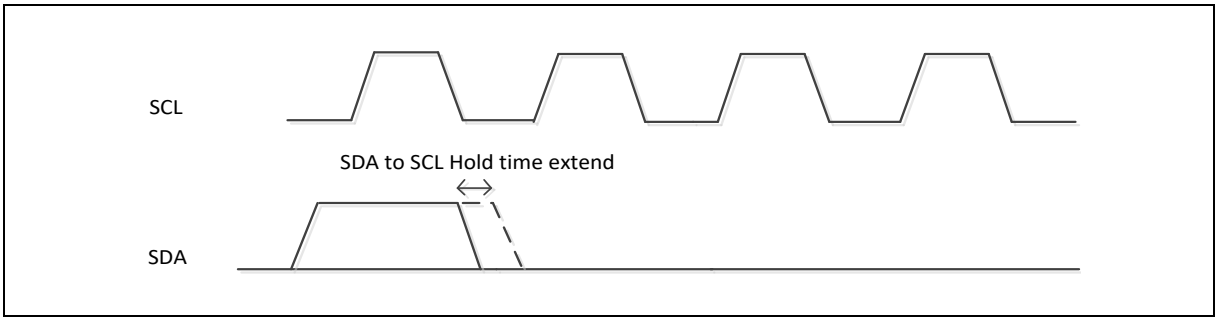


图 6.11-15 保持时间延长使能

I2STAT – I²C状态

寄存器	特殊功能寄存器地址	复位值
I2STAT	BDH, SFR页0	1111 1000 b

7	6	5	4	3	2	1	0
I2STAT[7:3]					**0**	**0**	**0**
只读					只读	只读	只读

位	名称	描述
7:3	**I2STAT[7:3]**	**I²C 状态码** I2STAT的高5位为状态码，共有27种可能的状态值。 I2STAT = F8H时，没有相关状态信息，SI 标志保持为 0。其余26种状态码对应I²C的状态。当进入这些状态时，SI 将被置为逻辑1并且产生中断请求。
2:0	**0**	**保留** I2STAT的低有效三位始终读到 0。

I2DAT – I²C 数据

寄存器	特殊功能寄存器地址	复位值
I2DAT	BCH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
I2DAT[7:0]							
读/写							

位	名称	描述
7:0	**I2DAT[7:0]**	**I²C 数据** I2DAT包含I²C数据待发送的一个字节，或刚接收到的一个字节。只要SI 为逻辑 1，I2DAT内的数据就会一直保留。在I²C传输过程中读取或写入I2DAT，结果不可预测。 当 I2DAT 中的数据移出时，总线上的数据也会同时移入以更新I2DAT。I2DAT 始终显示I²C总线上出现的最后一个字节。所以总线仲载失败时，I2DAT的值会在事务结束后发生变化。

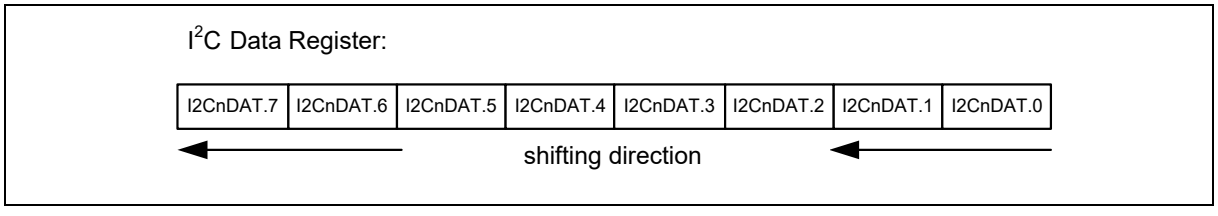


图 6.11-16 I²C 数据移位方向

I2ADDR – I²C 自身从机地址

寄存器	特殊功能寄存器地址	复位值
I2ADDR	C1H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
I2ADDR[7:1]							GC
读/写							读/写

位	名称	描述
7:1	**I2ADDR[7:1]**	**I²C 设备的从机地址** 主机模式: 此位段无效 从机模式: 这 7 位由用户定义此 I²C 设备的从机地址。主机应在起始条件或重复起始条件后的第一个字节数据中发送相同的地址来寻址 I²C设备。如果设置了 AA 标志，I²C设备接收到自己的地址后将响应主机，成为可寻址的从机。否则主机呼叫寻址会被忽略。 注意：I2ADDR[7:1] 不应保持全0的默认值，因为0x00保留用于广播呼叫。
0	**GC**	**广播呼叫位** 主机模式： 此位无效 从机模式： 0 =广播呼叫始终被忽略。 1 = 如果 AA 标志为 1，则识别广播呼叫；否则，如果 AA 为 0，则忽略广播呼叫。

I2CLK – I²C时钟

寄存器	特殊功能寄存器地址	复位值
I2CLK	BEH, SFR页0	0000_1001 b

7	6	5	4	3	2	1	0
I2CLK[7:0]							
读/写							

位	名称	描述
7:0	**I2CLK[7:0]**	**I²C 时钟设置** 主机模式下: 此寄存器设定作主机时I²C总线的时钟速率。时钟速率遵循公式如下： $\frac{F_{SYS}}{4 \quad (I2CLK+1)}$. 如果I²C的外设时钟频率是16MHz，默认状态下，I²C的总线的时钟频率为400Kbps。 注意：I2CLK 值 00H 及 01H 无效。这是一个限制。 从机模式下: 该字节无效，从机自动与给定的时钟速率同步，最高可达400k bps。

I2TOC – I²C超时计数器

寄存器	特殊功能寄存器地址	复位值
I2TOC	BFH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	I2TOCEN	DIV	I2TOF
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	I2TOCEN	**I²C超时计数器使能** 0 = I²C超时计数器禁用 1 = I²C超时计数器使能
1	DIV	**I²C超时计数器时钟分频** 0 = I²C超时计数器时钟为 F_{SYS}/1 1 = I²C超时计数器时钟为 F_{SYS}/4
0	I2TOF	**I²C超时标志** 如果14位I²C超时计数器溢出，此位由硬件置位。此位由软件清0。

6.12 串行外设接口 (SPI)

6.12.1 概述

CM1003 系列提供两个串行外设接口 (SPI) 模块，用于支持高速串行通信。SPI 是一种全双工、高速、同步通信总线，可连接微控制器或其他外设，例如串行 EEPROM、LCD 驱动器或 D/A 转换器。它支持主机或从机模式，速率高达 $F_{SYS}/2$，并带有传输完成标志和写入冲突标志。对于多主机系统，SPI 支持主机模式故障，以防止多主机冲突。

6.12.2 结构框图

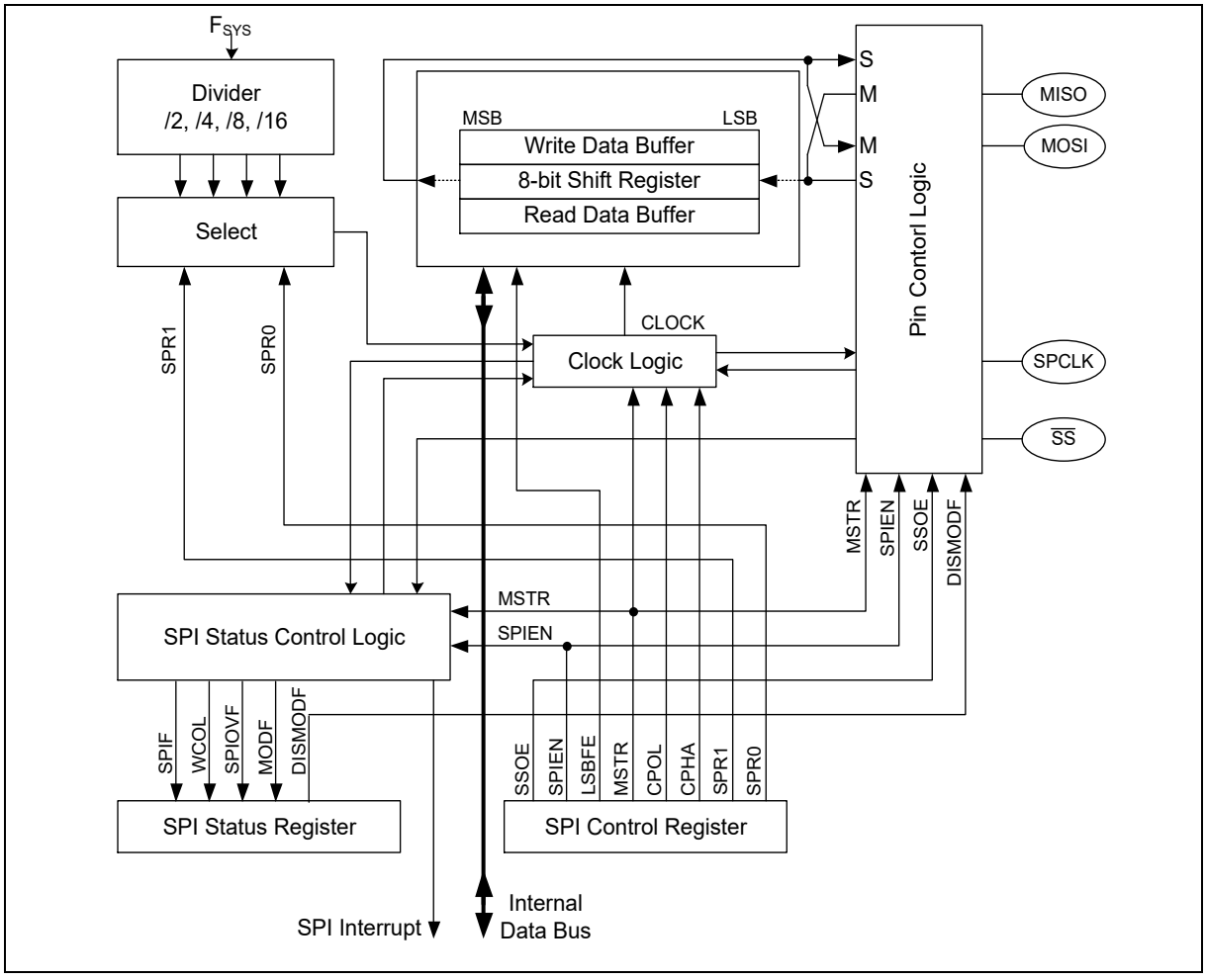


图 6.12-1 SPI 结构框图

图6.12-1 SPI 展示了SPI的架构。SPI 的主要模块包括 SPI 控制寄存器逻辑、SPI 状态逻辑、时钟速率控制逻辑和引脚控制逻辑。对于串行数据传输或接收，SPI 模块包含一个写入数据缓冲区、一个移位寄存器和一个读取数据缓冲区。它在接收和发送方向上采用双缓冲。前一个传输未完成时写入SPDR的新发送数据是会写入移位器的（不建议这样操作）。接收端由并行读取数据缓冲区组成，因为第一个接收到的数据将被传输到读取数据缓冲区，因此移位寄存器可以自由接收第二个数据。

6.12.3 功能描述

SPI 接口的四个引脚分别是主输入/从输出 (MISO)、主输出/从输入 (MOSI)、移位时钟 (SPCLK) 和从机选择(SS) 。MOSI 引脚用于将 8 位数据以串行方式从主机传输到从机。因此，MOSI 是主机的输出引脚，也是从机的输入引脚。相应的，MISO用于接收从机到主机的串行数据。

SPCLK 引脚在主机模式下为时钟输出，但在从机模式下为时钟输入。移位时钟用于同步通过 MOSI 和 MISO 引脚进出设备的数据移动。移位时钟由主机模式设备驱动八个时钟周期，八个时钟周期在串行总线上交换一个字节的数据。由于移位时钟始终由主机设备产生，因此系统中不应存在多个处于主机模式的设备，以避免设备冲突。

每个从机外设均由一个从机选择引脚(SS) 选择。当需要访问任何从机时，该从机的此信号脚都应保持低电平。当SS被驱动为高电平时，该从机将处于停用状态。如果系统为多从机，则同一时间只能选择一个从机设备。对于主机模式MCU， SS 引脚不起作用，可以将其配置为通用 I/O。但是，如果存在多主机环境，则可以通过软件设置将 SS 用作主机模式故障检测(参见章节**錯誤! 找不到參照來源。 錯誤! 找不到參照來源。**) 。CM1003还提供自动激活功能，可在每次字节传输之间切换 SS 。

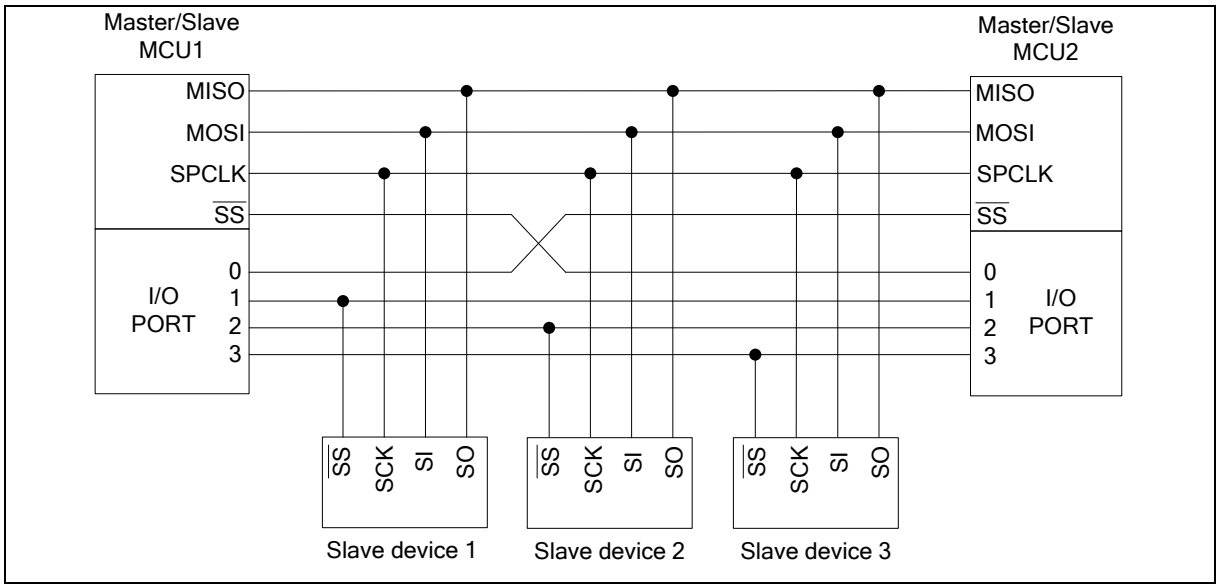


图 6.12-2 SPI 多主机, 多从机连接图

图6.12-2 SPI 显示了SPI设备的典型互连。该总线通过三条信号线将设备连接在一起：即MOSI到MOSI，MISO到MISO和SPCLK到SPCLK。主机设备通过并行端口的四个引脚控制四个 SS引脚来选择各个从机设备。MCU1和MCU2可以处于主机或从机模式。应将 SS引脚配置为主机模式故障检测，以避免多主机设备冲突。

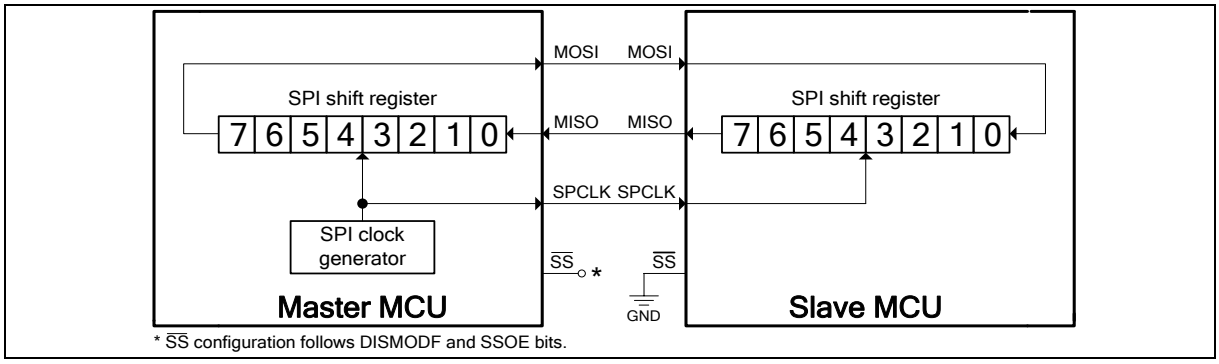


图 6.12-3 SPI 单主机，单从机连接图

图6.12-3 SPI 显示了最简单的SPI系统互连，即单主机和单从机。传输过程中，主机设备通过MOSI 线将数据移出到从机。同时，主机设备通过 MISO 线将数据从从机设备移入。此时主机和从机的两个移位寄存器可被视为一个16位的循环移位寄存器。因此，当主机将数据推送到从机时，从机数据也同时推向主机。该传输实际上交换了两个 MCU的SPI 移位寄存器中的数据。

默认情况下，SPI数据传输高位(MSB)优先。如果LSBFE(SPCR.5) 被置位，则SPI数据传输时采用低位(LSB)优先。此位不影响数据寄存器内MSB/LSB的排列顺序。请注意，以下所有描述和图片均基于LSBFE为逻辑0的情况下，MSB优先传输和接收。

SPI控制寄存器(SPCR)、SPI状态寄存器(SPSR)和SPI数据寄存器(SPDR)这三个寄存器用于SPI传输。这些寄存器提供控制，状态，数据存储功能和时钟速率选择。以下寄存器与SPI功能相关。

6.12.3.1 操作模式

主机模式

当MSTR(SPInCR.4)设置为1时，SPI可以在主机模式下运行。只有一个主机SPI设备可以发起传输。传输始终由主机写入SPDR开始。写入SPDR的字节在SPCLK的控制下开始从MOSI引脚移出。同时，另一个字节从从机的MISO引脚上移入。8位数据传输完成后，SPIF(SPInSR.7)将由硬件自动置位以指示一个字节的数据传输完成。同时，从从机接收的数据也会传输到SPDR中。用户可以清除SPIF并从SPDR中读取数据。

从机模式

当MSTR为0时，SPI工作在从机模式。SPCLK引脚变为输入，并由另一个主机SPI设备为其提供时钟。引脚$\overline{SS}$也变为输入。当从机设备的$\overline{SS}$引脚不为低时，主机设备不能与从机交换数据。在数据传输发生之前，应将从机的$\overline{SS}$拉低并保持低电平，直到传输完成。如果$\overline{SS}$变为高，SPI将被强制进入空闲状态。如果在传输过程中$\overline{SS}$被强制为高，传输将被中止，接收移位缓冲器的其余位将变为高，并进入空闲状态。

在从机模式下，数据通过MOSI引脚从主机向从机传输，通过MISO引脚从从机向主机传输。数据在主机设备的SPCLK控制下进入移位寄存器。移位寄存器接收到一个字节后，立即将其移入读数据缓冲区，同时SPIF置位。读取SPDR实际上是读取读数据缓冲区。为防止缓冲区溢出和溢出导致的字节丢失，在主机设备第二笔传输数据进入读数据缓冲区之前，从机应读出SPDR并清除第一个SPIF。

6.12.3.2 时钟格式和数据传输

为了兼容各种同步串行外设，SPI 具有时钟极性位CPOL (SPInCR.3)和时钟相位位CPHA (SPInCR.2)。 图6.12-4 SPI 显示了CPOL和CPHA 组成出的四种不同的时钟格式。CPOL位表示空闲状态下的SPCLK线路电平。CPHA位定义MOSI 和MISO线路上采样边沿用。同一系统上的主从设备的CPOL和CPHA设置应该相同。不同的数据格式相互通信将导致不确定的结果。

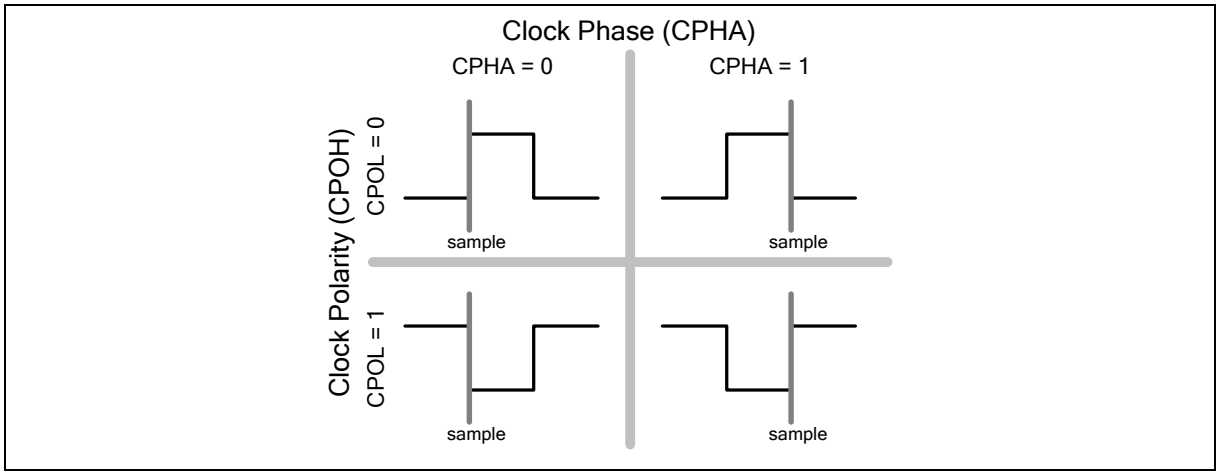


图 6.12-4 SPI 时钟格式

在SPI传输中，始终由主机发起传输。如果SPI被选择为主机模式(MSTR = 1)并使能(SPIEN = 1)，则主机向SPI数据寄存器(SPDR) 写入数据将启动SPI时钟和数据传输。移出一个字节并接收一个字节后，SPI时钟停止，主机和从机的SPIF(SPInSR.7)均被置位。如果SPI中断使能位被置1且全局中断使能(EA = 1)，则将执行SPI的中断服务程序(ISR)。

对于从机模式，需要注意$\overline{SS}$信号。如图6.12-4 SPI 所示，当CPHA=0时，SPCLK第一个边沿为MSB的采样选通(例如，LSBFE= 0，即MSB优先)。因此，从机应在第一个SPCLK边沿之前移出其 MSB 数据。$\overline{SS}$的下降沿可用于MISO线上准备MSB。因此，每个连续的串行字节之间，$\overline{SS}$引脚应切换为高电平，然后切换为低电平。此外，如果从机在$\overline{SS}$为低电平时将数据写入SPI 数据寄存器(SPDR)，则会发生写冲突错误。

当CPHA =1时，用SPCLK时钟的第二个边沿来采样。从机使用第一个SPCLK时钟移出MSB，而不是$\overline{SS}$的下降沿。因此，$\overline{SS}$线路可以在连续传输之间保持低电平。此格式更适合单主机单从机的结构使用。只要仅使用CPHA =1模式，就可以将唯一从机的$\overline{SS}$直接接地。

注意：SPI应在使能(SPIEN = 1)之前进行配置，否则LSBFE、MSTR、CPOL、CPHA和SPR[1:0]的任一更改将中止正在进行的传输并强制SPI系统进入空闲状态。在更改任何配置位之前，必须先禁用SPIEN。

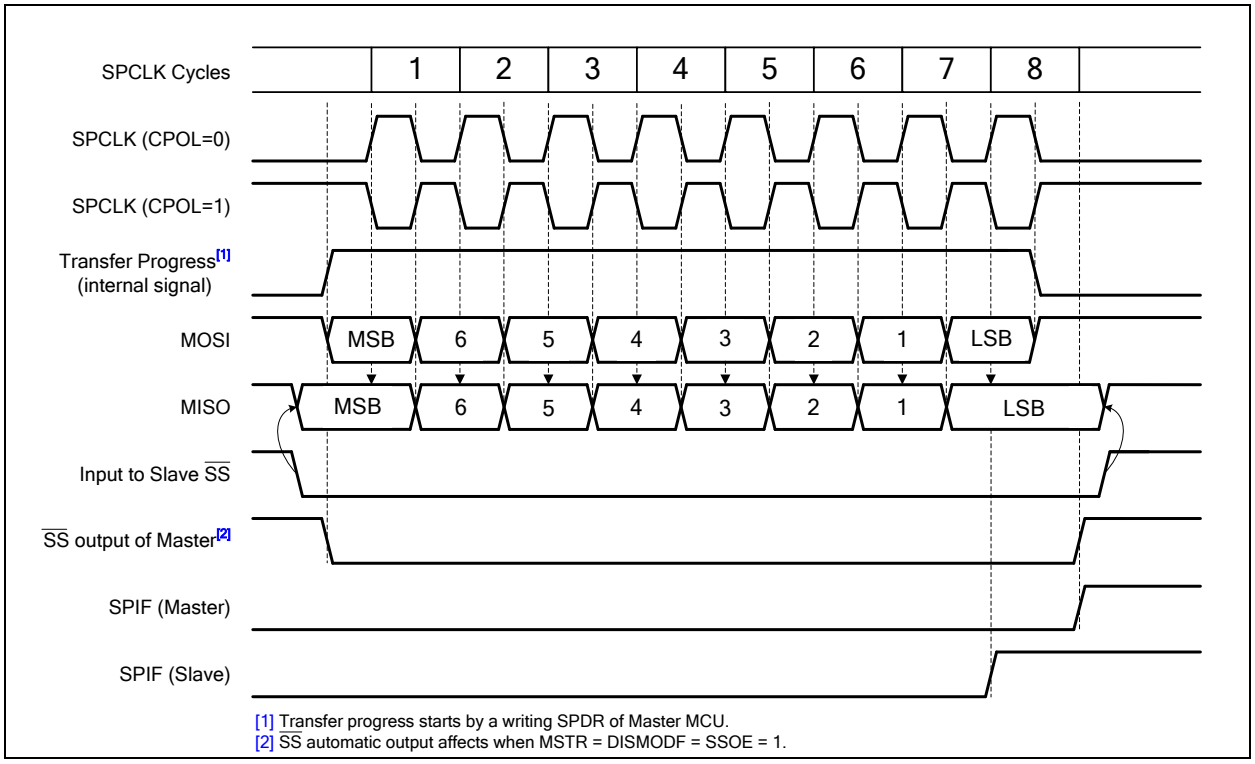


图 6.12-5 CPHA = 0 的 SPI 时钟和数据格式

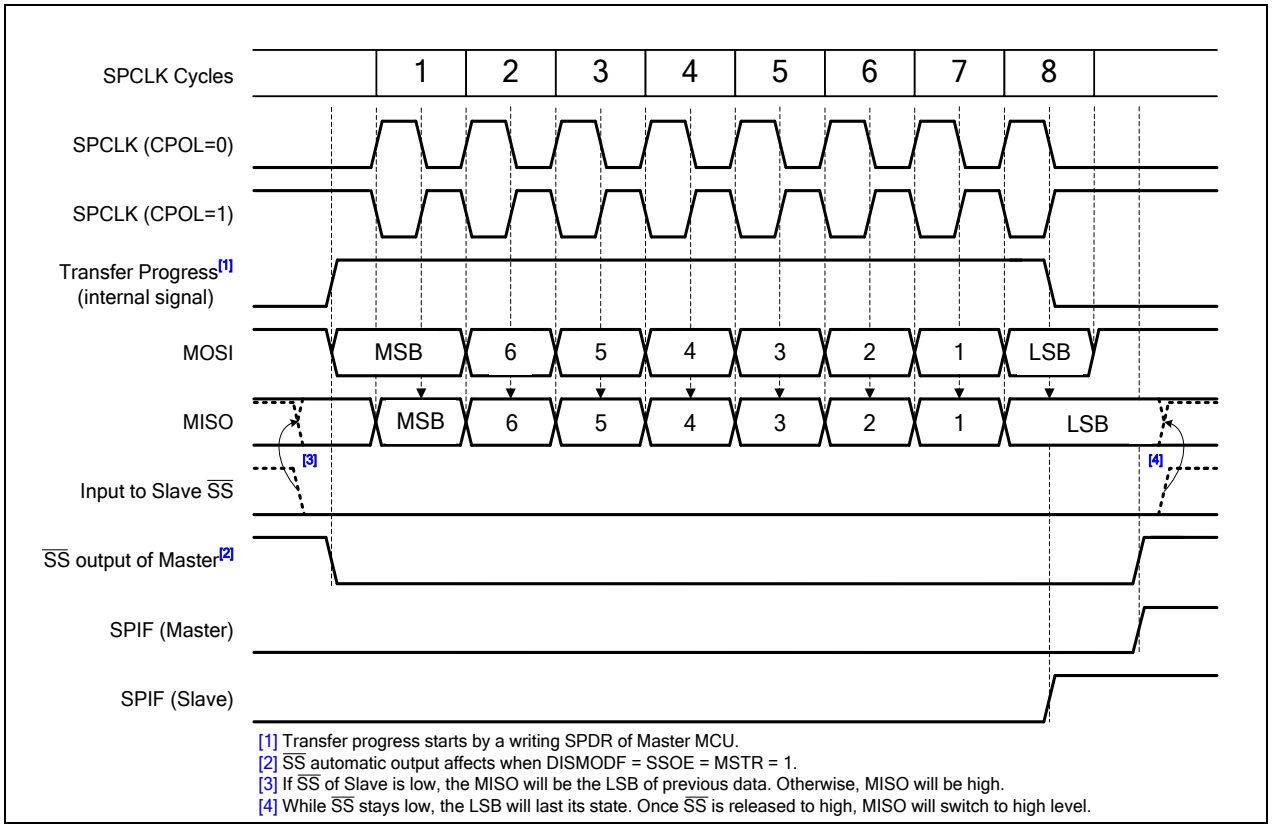


图 6.12-6 CPHA = 1 的 SPI 时钟和数据格式

6.12.3.3 从机选择引脚配置

CM1003 SPI提供灵活的$\overline{SS}$引脚功能，可满足不同的系统要求。当SPI作为从机运行时，$\overline{SS}$引脚始终作为从机选择输入引脚。当作为主机模式使能时，$\overline{SS}$引脚根据DISMODF(SPInSR.3)和SSOE(SPInCR.7)配置，具有三种不同的功能。默认情况下，DISMODF为0，故障检测功能已激活，$\overline{SS}$配置为输入引脚用于检查是否出现模式故障。相反，如果DISMODF为1，则故障检测功能关闭，SSOE位接管$\overline{SS}$引脚的功能。当SSOE为1时，意味着将自动生成从机选择信号以选择从机，主机的$\overline{SS}$作为输出引脚通常与从机的$\overline{SS}$输入引脚相连。在选择外部从机时，$\overline{SS}$输出会在每次传输时自动变为低电平，并在每次空闲时变为高电平以取消选择从机。当SSOE为0且DISMODF为1时，$\overline{SS}$不再被SPI使用并恢复为通用I/O引脚。

DISMODF	SSOE	主机模式 (MSTR = 1)	从机模式 (MSTR = 0)
0	X	$\overline{SS}$ 为故障模式输入引脚	$\overline{SS}$ 从机选择输入引脚
1	0	通用I/O	
1	1	自动$\overline{SS}$ 输出	

表 6.12-1 从机选择引脚配置

6.12.3.4 模式故障检测

在多个SPI设备可能同时成为主机的系统中，很有可能会出现数据争用，此时模式故障检测就很有用。当SPI设备配置为主机，并且输入$\overline{SS}$被配置为模式故障输入引脚时，(具体参考 表6.12-1) 一旦$\overline{SS}$被其他设备拉低，就会发生模式故障错误。这表明其他某个SPI设备正在尝试将此主机作为从机进行寻址。此时硬件立即会清除SPInCR中的MSTR和SPIEN控制位以禁用 SPI，设置模式故障标志位MODF(SPInSR.4)，并在ESPI和EA都使能时生成中断。

6.12.3.5 写冲突错误

SPI在传输方向采用信号缓冲，在接收和发送方向采用双缓冲。新的待传输数据必须等到前一个事务完成才能写入移位寄存器。如果在传输过程中多次写入SPDR，则会发生写冲突。SPDR在发送方向是双缓冲。对SPDR的任何写入都会导致数据直接写入SPI移位寄存器。一旦发生写入冲突错误，WCOL(SPInSR.6)将由硬件置位1，以指示发生了写冲突。在这种情况下，当前传输的数据将继续传输，而引起写冲突的新数据将丢失。虽然SPI逻辑可以在主机和从机之间检测写冲突，但写冲突通常是从机错误，因为当主机发起传输时，从机没有指示可以知道。在从机接收数据期间，对SPDR的写入会导致从机模式下的写冲突。WCOL标志位需要通过软件清除。

6.12.3.6 溢出错误

对于接收数据，SPI在接收方向是双缓冲的。接收的数据被传输到并行读数据缓冲区中，因此移位寄存器可以自由接收第二个串行字节。但是，在下一个数据完全移入之前，应从SPDR读取出已接收的数据。只要第一个字节从读数据缓冲区中读出，并且在下一个字节被移入之前清除SPIF，就不会发生溢出错误，否则会发生溢出错误。在这种情况下，第二个字节数据将无法成功移入到读数据寄存器中，缓冲区内仍保留有前一个数据。当发生溢出错误时，SPIOVF(SPInSR.5)被硬件置位1。如果中断使能，SPIOVF置位会产生中断请求。图6.12-7 SPI 显示了数据接收与溢出错误之间的关系。

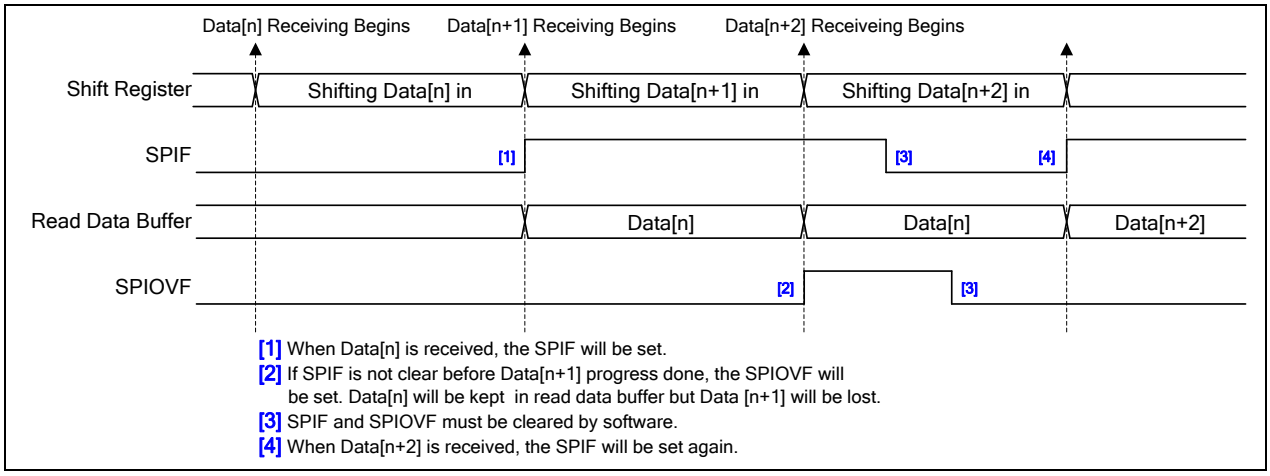


图 6.12-7 SPI 溢出波形

6.12.3.7 SPI中断

三个SPI状态标志SPIF、MODF和SPIOVF用于生成SPI事件中断请求。它们都位于SPSR寄存器中。当与外部设备的数据传输完成或新数据已收到并复制到SPDR后，SPIF标志将被置位。MODF置1时表示$\overline{SS}$上的低电平导致模式故障状态。SPIOVF表示接收数据发生溢出错误。如果通过设置 ESPI 和EA置1让SPI屏蔽中断使能时，则只要这3个标志中的任意一个置1，CPU会执行SPI中断服务程序。用户需要检查标志位以确定导致中断的事件。这三个标志位是软件清除的。

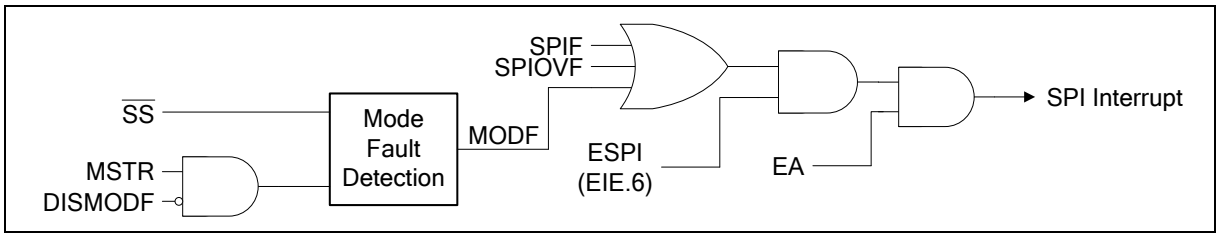


图 6.12-8 SPI 中断请求

6.12.4 寄存器描述

SPCR – 串行外设控制寄存器寄存器

寄存器	特殊功能寄存器地址	复位值
SPCR	F3H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**SOE**	**从机选择输出使能位** 此位与DISMODF(SPSR.3)位配合使用，用于确定$\overline{SS}$引脚的功能，如下所示 0 = $\overline{SS}$. 引脚用作通用I/O引脚。 1 = $\overline{SS}$. 引脚在选择外部从机设备时，每次传输后自动变为低电平，并在空闲状态下变为高电平以取消选择从机设备。
6	**PIEN**	**SPI使能** 0 = SPI功能禁用. 1 = SPI功能使能.
5	**LSBFE**	**LSB优先使能** 0 = SPI 数据按 MSB 优先传输。 1 = SPI 数据按 LSB 优先传输。
4	**STR**	**主机模式使能** 此位用于切换SPI的主机与从机模式. 0 = SPI 配置为从机模式. 1 = SPI 配置为主机模式.
3	**OL**	**SPI 时钟极性选择** CPOL 位决定 SPI 时钟的空闲状态电平。详见图6.12-4 SPI . 0 = SPI 时钟在空闲模式时低电平. 1 = SPI 时钟在空闲模式时高电平.
2	**CPHA**	**SPI 时钟相位选择** CPHA 位决定 SPI 时钟的数据采样沿。详见图6.12-4 SPI . 0 = 数据在 SPI 时钟的第一个沿采样。 1 = 数据在 SPI 时钟的第二个沿采样。
1:0	**SPR[1:0]**	**SPI 时钟速率选择** 这两位选择 SPI 时钟分频器的四种等级。以下时钟速率是在 F_{SYS} = 16 MHz 条件下计算的。 Fsys = 16 MHz SPR1 / SPR0 / 分频 / SPI 时钟速率 0 / 0 / 2 / 8M bit/s 0 / 1 / 4 / 4M bit/s 1 / 0 / 8 / WM bit/s 1 / 1 / 16 / 1 M bit/s Fsys = 24 MHz SPR1 / SPR0 / 分频 / SPI 时钟速率 0 / 0 / 2 / 12M bit/s 0 / 1 / 4 / 6M bit/s 1 / 0 / 8 / 3M bit/s 1 / 1 / 16 / 1.5M bit/s SPR[1:0] 仅在主机模式(MSTR = 1)下有效。如果处于从机模式，时钟将自动与SPICLK引脚上来自主机的外部时钟同步， 通信速度最高可达F_{SYS}/2。

SPCR2 – 串行外设控制寄存器寄存器 2

寄存器	特殊功能寄存器地址	复位值
SPCR2	F3H, SFR页1	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	**SPIS1**	**SPIS0**
-	-	-	-	-	-	读/写	读/写

位	名称	描述
7:2	-	保留
1:0	**SPIS[1:0]**	**SPI 相邻字节的间隔时间选择** SPIS[1:0]和CPHA在8级SPI相邻字节间的间隔时间做选择，如下表： CPHA / SPIS1 / SPIS0 / SPI 时钟 0 / 0 / 0 / 0.0 0 / 0 / 1 / 0.5 0 / 1 / 0 / 1.5 0 / 1 / 1 / 2.0 1 / 0 / 0 / 0.0 1 / 0 / 1 / 1.0 1 / 1 / 0 / 2.0 1 / 1 / 1 / 2.5 SPIS[1:0]仅在主机模式(MSTR = 1)下有效。

SPSR – 串行外设状态寄存器

寄存器	特殊功能寄存器地址	复位值
SPSR	F4H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SPIF	WCOL	SPIOVF	MODF	DISMODF	TXBUF	-	-
读/写	读/写	读/写	读/写	读/写	只读	-	-

位	名称	描述
7	SPIF	**SPI完成标志** 在SPI 数据传输完成或接收到的数据移入到SPI 读缓冲区时，此位由硬件置位逻辑1。 如果ESPI (EIE .0)和 EA使能，触发SPI 中断请求。此位应由软件清除。如果SPIF置位1，禁止向SPDR写入。
6	WCOL	**写入冲突错误位** 此位标示写入冲突事件。一旦发生写入冲突事件，此位被置位，应由软件清零。
5	SPIOVF	**SPI 溢出错误标志** 此位指示发生溢出事件。一旦发生溢出事件，此位置位。如果 ESPI 和 EA 使能，则触发 SPI 中断。此位应通过软件清零。
4	MODF	**模式故障错误标志** 此位指示发生模式故障错误事件。如果$\overline{SS}$引脚配置为模式故障输入(MSTR =1且 DISMODF = 0)并且$\overline{SS}$被外部设备拉低，则会发生模式故障错误。MODF 将立即被置位逻辑 1。如果ESPI 和 EA使能，则触发 SPI 中断请求。应通过软件清除此位。
3	DISMODF	**禁用模式故障错误检测** 此位与 SSOE(SPCR.7)位组合使用，用以确定引脚的特性，如表6. 12-1 所示DISMODF 仅在主机模式下有效(MSTR = 1)。 0 = 模式故障检测使能。$\overline{SS}$作为模式故障检测的输入引脚，忽略 SSOE设置。 1 = 模式故障检测禁用。$\overline{SS}$ 功能遵循 SSOE 位配置。
2	TXBUF	**SPI 写入数据缓冲区状态** 此位指示 SPI 传输缓冲区状态。 0 = SPI 写入器数据缓冲区为空 1 = SPI 写入器数据缓冲区已满。

SPDR – 串行外设数据寄存器

寄存器	特殊功能寄存器地址	复位值
SPDR	F5H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
SPDR[7:0]							
读/写							

位	名称	描述
7:0	SPDR[7:0]	**串行外设数据** 此字节用于在 SPI 总线上发送或接收数据。写入此字节即写入移位寄存器。读取此字节实际上是读取数据缓冲区。在主机模式下，写入此寄存器可同时启动字节的发送和接收。

6.13 12位模数转换器 (ADC)

6.13.1 概述

CM1003系列内嵌12位SAR(逐次逼近寄存器)的ADC。ADC(模拟数字转换器)可将模拟输入信号转换为12位二进制表示。CM1003 系列在单端模式下可选择15个通道输入。内部带隙电压也可作为内部ADC输入。模拟输入复用至一个采样保持电路，对采样保持电容充电。此采样保持电容的输出作为转换器的输入。然后转换器通过逐次逼近生成此模拟电平的数字结果，并将结果存储在结果寄存器中。ADC控制器还支持连续转换并将结果数据存储到 XRAM 中。

6.13.2 基本配置

以下显示了ADC的多功能定义。

组	引脚名称	GPIO	描述
ADC	ADC_CH0	P1.7	ADC通道0模拟输入
	ADC_CH1	P3.0	ADC通道1模拟输入
	ADC_CH2	P0.7	ADC通道2模拟输入
	ADC_CH3	P0.6	ADC通道3模拟输入
	ADC_CH4	P0.5	ADC通道4模拟输入
	ADC_CH5	P0.4	ADC通道5模拟输入
	ADC_CH6	P0.3	ADC通道6模拟输入
	ADC_CH7	P1.1	ADC通道7模拟输入
	ADC_CH9	P2.1	ADC通道9模拟输入
	ADC_CH10	P2.2	ADC通道10模拟输入
	ADC_CH11	P2.3	ADC通道11模拟输入
	ADC_CH12	P2.4	ADC通道12模拟输入
	ADC_CH13	P1.3	ADC通道13模拟输入
	ADC_CH14	P1.4	ADC通道14模拟输入
	ADC_CH15	P2.5	ADC通道15模拟输入

6.13.3 框图

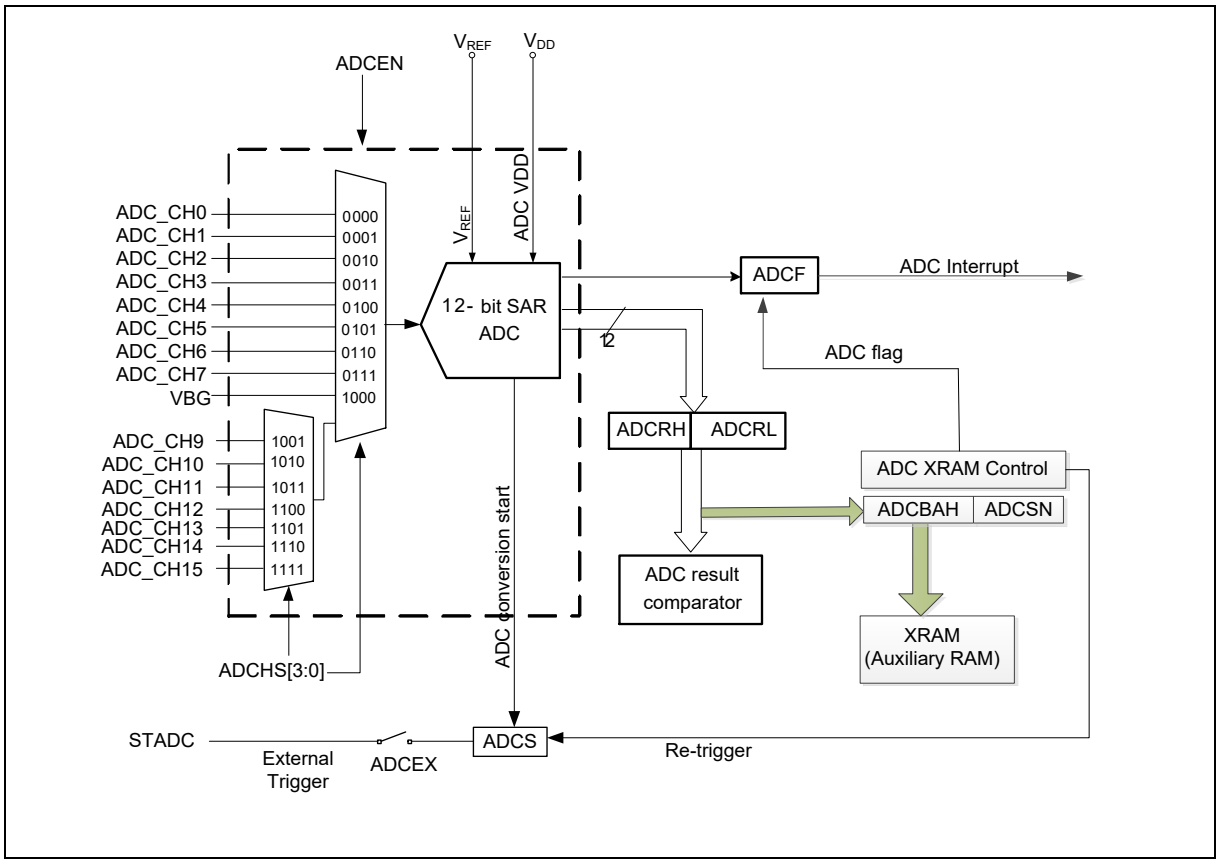


图 6.13-1 12 位 ADC 框图

6.13.4 功能描述

6.13.4.1 ADC 操作

ADC操作前，应通过设置ADCEN(ADCCON1.0)来使能ADC电路。这将激活ADC电路，但会消耗额外的功耗，一旦ADC转换模块不再使用，建议清零ADCEN位关闭ADC电路可以节省功耗。

ADC模拟输入引脚需要特别注意。ADCHS[2:0] 是通道选择位，用于控制哪个通道连接到采样保持电路。用户需要通过PxMn寄存器中的相应位，将选定的ADC输入引脚配置为仅输入(高阻抗)模式。此配置会断开每个选定 ADC 输入引脚的数字输出电路，但数字输入电路仍然工作。由于数字输入可能会导致输入缓冲产生漏电流，所以应配置AINDIDS0中的相应位来禁用数字输入缓冲区。上述配置使选定的ADC输入引脚设置为纯模拟输入，允许外部输入模拟电压信号。此外，还需要认真考虑ADC采样时钟速率。ADC最高时钟频率列在ADC模拟电气特性中。当采样时钟超过最大值时，会不可预测地降低 ADC 性能。

通过置位ADCS(ADCCON0.6)来开启A/D转换。转换完成后，硬件将自动清除ADCS，置位ADCF(ADCCON0.7)，并在中断使能的情况下生成中断。新的转换结果存储在ADCR(最高8位)和ADCRL(低有效4位)中。12位转换结果值为 $4095 \times \frac{V_{AIN}}{V_{REF}}$。

设备内部及外部的数字电路会产生噪声，这可能会影响ADC测量的精度。如果需要高精准的转换结果，则可以应用以下技术来降低噪声干扰：

1. 尽可能缩短模拟信号路径。确保模拟信号走线远离高速数字信号走线。
2. 转换期间将设备进入空闲模式。

3. 如果任何模拟输入ADC_CHn引脚也用作数字输出，则务必确保这些引脚在转换过程中不要切换。

6.13.4.2 ADC 转换时序

以下是总ADC转换时间T_{ADC}，包括以下时间：

ADC 转换时间： $T_{ADC} = T_{SAMPLING} + T_{ENCODING}$

ADC 采样时间：$T_{SAMPLING} = \frac{4\times ADCAQT+6}{F_{SYS}\div ADCDIV}$

ADC 编码时间：$T_{ENCODING} = \frac{1}{500k}(s)$= 固定是2μs.

6.13.4.3 外部触发ADC转换

除了通过软件设置ADCS之外，CM1003还支持硬件触发方法来启动A/D转换。如果置位了ADCEX(ADCCON1.1)，则所选PWM通道上的边沿或周期点或STADC引脚的边沿将自动触发A/D转换(硬件触发器硬件置位ADCS)。

有效条件由ETGSEL(ADCCON0[5:4])和ETGTYP(ADCCON1[3:2])选择触发源和触发类型。还可以在外部触发点与A/D转换之间插入触发延迟。外部触发ADC硬件具有可控的触发延迟，使CM1003适用于高性能电机控制。请注意，在ADC处于转换忙状态(ADCS = 1)，任何由软件或硬件触发的转换都将被忽略，并且不会出现任何警告。

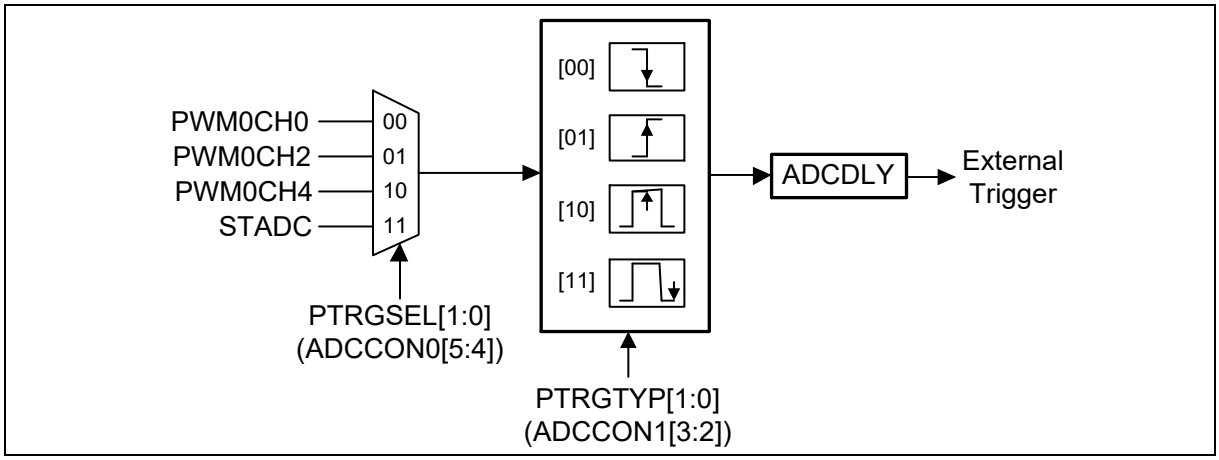


图 6.13-2 外部触发 ADC 电路

6.13.4.4 ADC 转换结果比较器

CM1003系列ADC带有一组数字比较器，用于将A/D转换结果与ACMPH和ACMPL寄存器中给定的12位常数值进行比较。置位ADCMPEN(ADCCON2.5)使能 ADC 比较器，每次A/D转换完成时都会进行比较。ADCMPO(ADCCON2.4)根据其输出极性设置位ADCMPOP(ADCCON2.6)显示比较结果。ADC比较结果可直接触发PWM故障刹车输出。此功能在ADFBEN(ADCCON2.7)位使能。ADCMPO置位后，会产生ADC比较事件并触发故障刹车。详细请参阅章节6.8.4.5故障刹车。

注意： 对于CM1003系列，使能结果比较功能后，仅当ADC比较结果与条件匹配时，ADCF寄存器才会变为1。如果ADC中断使能，则进入中断向量。使能此位并触发ADC启动后，ADC将持续转换。寄存器ADCRH和ADCRL值将根据ADC设置的结果而改变，也可以从寄存器中读取。此过程仅在ADCF设置为1后才会停止。

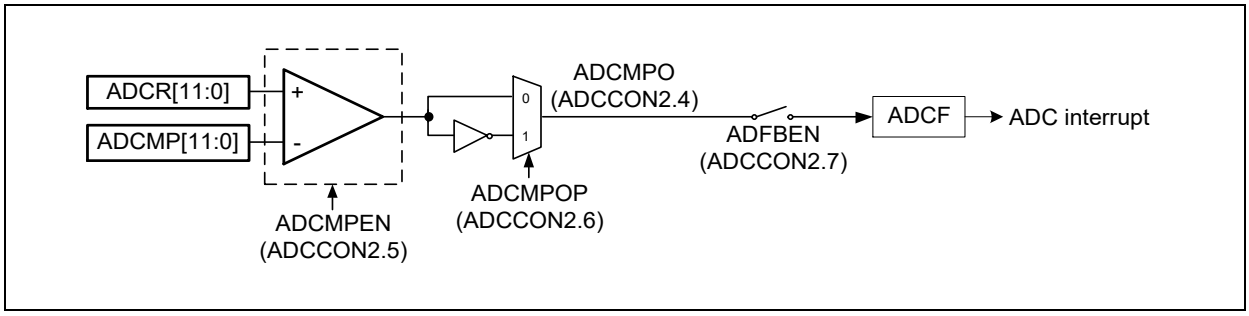


图 6.13-3 ADC 结果比较器

6.13.4.5 内部带隙电压(Band-gap)

室温下，所有CM1003的带隙电压值都会在校准范围内。如需获取CM1003的实际带隙电压值，请读取UID地址后的2个字节值，实际12位有效。第一个字节高8位，第二个字节的低4位为12位的低4位。

读取和计算步骤：

- 用IAP读取UID，读到Band-gap的值;
- 合并高8位和低4位，定义为READ_BANDGAP;
- 使用以下公式换算为实际的电压值。

公式如下

$$Bandga\_Voltage = \frac{READ\_BANDGAP \times 3072}{4096}(mV)$$

例如:

读取UID地址后的2个字节，第一个字节值为0x64，第二个字节值为0x0E，合并为0x64E = 1614，

换算电压值结果为：

$$Bandgap\_Voltage = \frac{1614 \times 3072}{4096} = 1210.5(mV)$$

CM1003内部带隙电压也可以作为内部ADC输入。此输入对于测量V_{REF}值很有用，意味着可以从ADC转换结果计算出V_{DD}值。带隙电压作为 ADC 输入计算 VDD 值:

$$V_{DD} = \frac{4096}{ADC\_Bandga\_Result} \times Bandgap\_Voltage \text{ 或 } V_{DD} = \frac{READ\_BANDGAP \times 3072}{ADC\_Bandgap\_Result}(mV)$$

6.13.4.6 ADC 连续转换

ADC 控制器支持连续转换功能，可自动存储 A/D 转换结果。ADC 连续转换模式可将 12 位 ADC 结果存储到 XRAM 缓冲区中，12 位 ADC 数据会自动分成 8 位高字节和 4 位低字节两部分。为了减少 XRAM 存储空间，两个 4 位低字节数据（连续 ADC 转换结果）会自动合并为一个字节大小并存储在 XRAM 中。

存储方式如下图所示。首先存储 8 位转换结果的高字节数据，然后再存储合并后的数据，分割点为 ADC 连续转换长度，该长度由 ADCCN 定义。

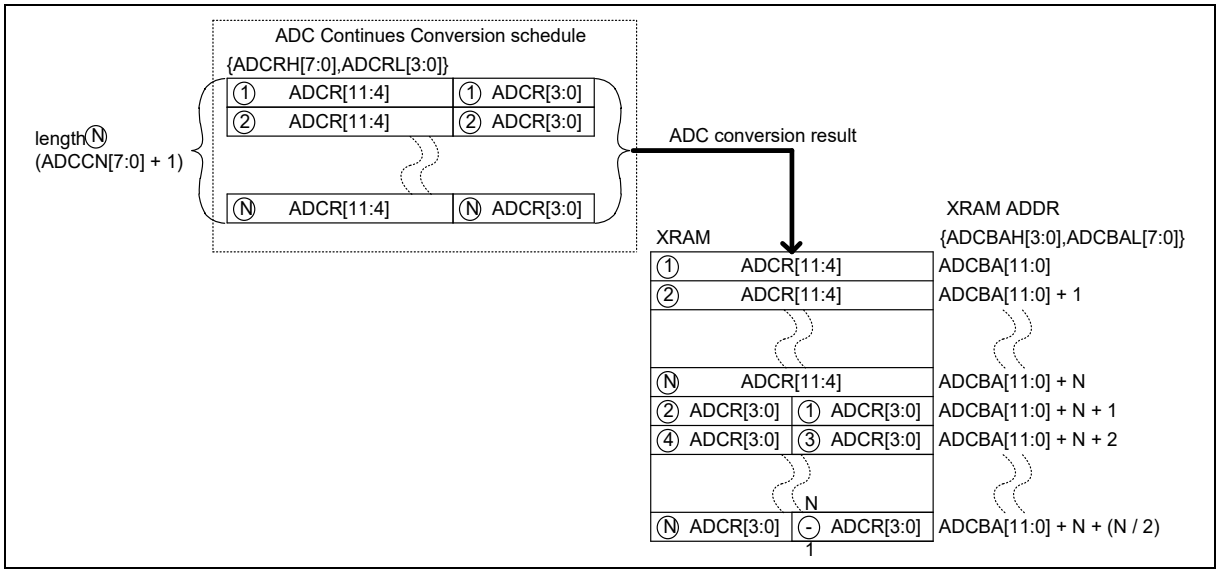


图 6.13-4 ADC 连续模式

编程序列如下：

1. 设置 ADC 通道并使能，方法与普通 ADC 设置相同。
2. 置位 CONT（ADCCON1.4）为 1，将 ADC 设置为连续转换模式。
3. 设置 ADCBAH 和 ADCBAL 寄存器，配置转换结果的存储地址。
4. 设置 ADCCN 寄存器，配置 ADC 转换次数。
5. 设置 HIE/FIE（ADCCON1[5]）使能 ADC 转换半完成中断。（可选）
6. 通过软件触发（ADCS=1）或外部触发（ADCEX=1）启动 ADC 运行。

6.13.5 寄存器描述

ADCCON0 – ADC控制0 (可位寻址)

寄存器	特殊功能寄存器地址	复位值
ADCCON0	E8H, 所有页	0000_0000b

7	6	5	4	3	2	1	0
ADCF	ADCS	ETGSEL1	ETGSEL0	ADCHS3	ADCHS2	ADCHS1	ADCHS0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7	**ADCF**	**ADC标志** 此标志在 A/D 转换完成时置位。ADC 结果可读取。当此标志为 1 时，ADC 无法启动新的转换。此位由软件清零。
6	**ADCS**	**A/D转换软件启动触发器** 此位置1可触发AD转换。此位在 A/D 转换期间保持逻辑 1，并在转换完成后由硬件自动清零。写入和读取ADCS位的含义不同。 写入: 0 = 无影响 1 = 启动AD转换 读取: 0 = ADC处于空闲状态 1 = ADC转换中，忙状态
5:4	**ETGSEL[1:0]**	**外部触发源选择** 当ADCEX(ADCCON1.1)置位，这些位选择哪个引脚输出触发ADC转换。 00 = PWM0. 01 = PWM2. 10 = PWM4. 11 = STADC引脚.

位	名称	描述
3:0	**ADCHS[3:0]**	**A/D转换通道选择** 此字段选择ADC的激活模拟输入源。如果 ADCEN 为 0，则所有输入均断开。 0000 = ADC_CH0 0001 = ADC_CH1. 0010 = ADC_CH2. 0011 = ADC_CH3. 0100 = ADC_CH4. 0101 = ADC_CH5. 0110 = ADC_CH6. 0111 = ADC_CH7. 1000 = VBG (内部带隙电压) 1001 = ADC_CH9. 1010 = ADC_CH10. 1011 = ADC_CH11. 1100 = ADC_CH12. 1101 = ADC_CH13. 1110 = ADC_CH14. 1111 = ADC_CH15.

ADCCON1 – ADC 控制1

寄存器	特殊功能寄存器地址	复位值
ADCCON1	E1H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
OCEN	STADCPX	ADCDIV[1:0]		ETGTYP[1:0]		ADCEX	ADCEN
读/写	读/写	读/写		读/写		读/写	读/写

位	名称	描述
7	**OCEN**	**此字段用于启用失调校准功能。** 0：ADC 失调校准已启用，由 ADC 硬件自动校准。 1：ADC 失调校准已禁用。
6	**STADCPX**	**外部启动ADC触发引脚选择** 0 = 将STADC分配给P0.4. 1 = 将STADC分配给P1.3. 注意：一旦设置或清除此位后，STADC脚会立即切换。
5:4	**ADCDIV[1:0]**	**ADCAQT0 和 1 时钟分频** 00 = ADCAQT 时钟源 = F_{SYS}/1. 01 = ADCAQT 时钟源 = F_{SYS}/2. 10 = ADCAQT 时钟源 = F_{SYS}/4. 11 = ADCAQT 时钟源 = F_{SYS}/8.
3:2	**ETGTYP[1:0]**	**外部触发类型选择** 当ADCEX(ADCCON1.1)置位时，这些位决定ADC 转换的触发条件。 00 = PWM0/2/4或STADC引脚的下降沿。 01 = PWM0/2/4或STADC引脚的上升沿。 10 = PWM周期的中心点。 11 = PWM周期的结束点。 请注意，中心点中断或周期点中断仅适用于PWM中心对齐类型。
1	**ADCEX**	**ADC外部转换触发器选择** 此位选择触发 A/D 转换的方法。 0 = 仅通过设置ADCS位来启动 A/D 转换。 1 = 通过设置ADCS位或由外部触发源启动A/D 转换，具体取决于 ETGSEL[1:0] 和 ETGTYP[1:0]。注意，当 ADCS 为1(转换忙)时，ADC 将忽略后续的外部触发，直到 ADCS 被硬件清零。
0	**ADCEN**	**ADC使能** 0 = ADC电路关闭。 1 = ADC电路开启。

ADCCON2 – ADC 控制2

寄存器	特殊功能寄存器地址	复位值
ADCCON2	E2H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
ADFBEN	**ADCMPOP**	**ADCMPEN**	**ADCMPO**	**ADCAQT1[2:0]**			**ADCDLY.8**
读/写	读/写	读/写	只读	读/写			读/写

位	名称	描述
7	**ADFBEN**	**ADC比较结果响应故障刹车使能** 0 = ADC触发故障刹车禁用 1 = ADC触发故障刹车功能使能。一旦比较结果 ADCMPO 为1，故障刹车启动，同时PWM输出故障刹车数据。PWM0RUN (PWM0CON0.7)也将由硬件自动清除。当PWM0RUN再次置位时，PWM输出恢复。
6	**ADCMPOP**	**ADC比较器输出极性** 0 =如果ADCR[11:0]大于或等于ADCMP[11:0]，则ADCMPO 为1。 1 =如果ADCR[11:0]小于ADCMP[11:0]，则 ADCMPO 为1。
5	**ADCMPEN**	**ADC结果比较器使能** ADC 结果比较器触发 ADCF使能位。仅当比较器值与定义的 ADC 比较值条件匹配时，ADCF 才会设置为1。此条件基于 ADCMPH、ADCMPL 和 ADCMPOP 寄存器的定义。 仅当ADC比较结果与条件匹配时，ADCF寄存器才会变为 1。如果ADC中断使能，则进入中断向量。 0 = ADC结果比较器触发ADCF禁用。 1 = ADC结果比较器触发ADCF使能。 **注意：**此位置位并触发ADC启动后，ADC持续转换。寄存器ADCRH和ADCRL的值将根据ADC的结果而改变，也可以从寄存器中读出。此过程仅在 ADCF 设置为 1 后停止。
4	**ADCMPO**	**ADC比较器输出值** 此位是基于ACMPOP设置的ADC结果比较器的输出值。每次A/D转换完成后，此位都会更新。
3:1	**ADCAQT1[2:0]**	**ADC 采样时间0** 这3位字段决定ADC AIN0~AIN7 和BG采样的采样时间，遵循以下公式： ADC 采样时间 = $\frac{4*ADCAQT+6}{F_{ADCCLK}}$，其中 ADCAQT 在 ADCCON2[3:1] 中定义， $F_{ADCCLK}=\frac{F_{SYS}}{ADCDIV}$，其中 ADCDIV 在 ADCCON1[5:4] 中定义。 注意： ADC 转换过程中，此字段不可更改。 默认情况下，$F_{ADCCLK} = F_{SYS}$ (ADCDIV = 0)，最小采样时间为 6 个 ADC 时钟周期。由于最小采样时间必须超过 370ns，因此当 FSYS = 24MHz 时，ADCAQT 必须至少设置为 1。
0	**ADCDLY.8**	**ADC外部触发延迟计数器 第8位** 参见ADCDLY寄存器。

ADCCON3 – ADC 控制3

寄存器	特殊功能寄存器地址	复位值
ADCCON3	86H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	OVPEN	HIE	CONT	ADCAQT2[2:0]			-
-	读/写	读/写	读/写	读/写			0

位	名称	描述
7	-	保留
6	OVPEN	**ADC 过压保护手动使能** 0 = ADC 过压 (VDD > 7V) 保护禁用。 1 = ADC 过压 (VDD > 7V) 保护使能。 **注意：** 设置此位并等待 2us 以上，之后 ADCEN (ADCCON1[0]) 才会置位。
5	HIE	**ADC 半完成中断使能** 0 = 在继续模式下，当一半 A/D 转换完成时，不设置 ADC 中断 1 = 在继续模式下，当一半 A/D 转换完成时，设置 ADC 中断
4	CONT	**ADC 继续采样选择** 0 = ADC 单次采样，当一次 A/D 转换完成时设置 ADC 中断 1 = ADC 继续采样。当所有 A/D 转换完成时，设置 ADC 中断
3:1	ADCAQT2	**ADC 采样时间1** 这 3 位字段决定ADC AIN9~AIN15采样的采样时间，遵循以下公式： ADC 采样时间 = $\frac{4*ADCAQT+6}{F_{ADCCLK}}$, 其中 ADCAQT 在 ADCCON3[3:1] 中定义， $F_{ADCCLK}=\frac{F_{SYS}}{ADCDIV}$, 其中 ADCDIV 在 ADCCON1[5:4] 中定义。 注意： ADC 转换过程中，此字段不可更改。 默认情况下，$F_{ADCCLK} = F_{SYS}$ (ADCDIV = 0)，最小采样时间为 6 个 ADC 时钟周期。由于最小采样时间必须超过 370ns，因此当 FSYS = 24MHz 时，ADCAQT 必须至少设置为 1。
0	-	**保留** 此位应保持为 0。

ADCDLY – ADC 触发延迟计数器

寄存器	特殊功能寄存器地址	复位值
ADCDLY	E3H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
ADCDLY[7:0]							
读/写							

位	名称	描述
7:0	**ADCDLY[7:0]**	**ADC外部触发延迟计数器低字节** 此 8 位字段与 ADCCON2.0组成一个9位计数器。该计数器在检测到外部触发后插入一段延迟时间。A/D 转换在此延迟时间后再开始。 外部触发延迟时间 = $\frac{ADCDLY}{F_{ADC}}$. 注意，此字段仅当ADCEX(ADCCON1.1)被置位才有效。如果选择PWM输出作为外部ADC触发源，用户不应在PWM运行期间修改ADCDLY。

AINDIDS0 – ADC通道数字输入断开

寄存器	特殊功能寄存器地址	复位值
AINDIDS0	F6H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
P11DIDS	P03DIDS	P04DIDS	P05DIDS	P06DIDS	P07DIDS	P30DIDS	P17DIDS
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	**PxxDIDS**	**ADC通道数字输入禁用** 0 = ADC通道n数字输入使能。 1 = ADC通道n数字输入禁用。ADC通道n读取始终为0。

AINDIDS1 – ADC 通道数字输入断开1

寄存器	特殊功能寄存器地址	复位值
AINDIDS1	99H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
P25DIDS	P14DIDS	P13DIDS	P24DIDS	P23DIDS	P22DIDS	P21DIDS	-

位	名称	描述
7:1	**PxxDIDS**	**ADC通道数字输入禁用** 0 = ADC通道n数字输入使能。 1 = ADC通道n数字输入禁用。ADC通道n读取始终为0。

ADCRH – ADC 转换结果高字节

寄存器	特殊功能寄存器地址	复位值
ADCRH	C3H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
ADCR[11:4]							
只读							

位	名称	描述
7:0	**ADCR[11:4]**	**ADC 转换结果高字节** ADC 转换结果的最高有效8 位存储在此寄存器中。

ADCRL – ADC 转换结果低字节

寄存器	特殊功能寄存器地址	复位值
ADCRL	C2H, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	ADCR[3:0]			
-	-	-	-	只读			

位	名称	描述
7:4	-	保留
3:0	**ADCR[3:0]**	**ADC结果低字节** ADC转换结果的最低有效 4 位存储在此寄存器中。

ADCMPH – ADC比较值高字节

寄存器	特殊功能寄存器地址	复位值
ADCMPH	CFH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
ADCMP[11:4]							
写/读							

位	名称	描述
7:0	**ADCMP[11:4]**	**ADC比较值高字节** ADC比较值的最低有效4位存储在此寄存器中。

ADCMPL – ADC比较值低字节

寄存器	特殊功能寄存器地址	复位值
ADCMPL	CEH, SFR页0	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	**ADCMP[3:0]**			
-	-	-	-	写/读			

位	名称	描述
7:4	-	保留
3:0	**ADCMP[3:0]**	**ADC比较值低字节** ADC比较值的最低有效4位存储在此寄存器中。

ADCBAH – ADC RAM基地址高字节

寄存器	特殊功能寄存器地址	复位值
ADCBAH	84H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-				ADCBAH[3:0]			
-				读/写			

位	名称	描述
7:4	-	保留
3:0	**ADCBAH[3:0]**	**ADC RAM 基地址 (高字节)** RAM基地址的高4位用于存储ADC连续采样数据。 RAM基地址ADCBA[11:0] = {ADCBAH[3:0], ADCBAL[7:0]}

ADCBAL – ADC RAM基地址低字节

寄存器	特殊功能寄存器地址	复位值
ADCBAL	85H, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
ADCBAL[7:0]							
读/写							

位	名称	描述
7:0	**ADCBAL[7:0]**	**ADC RAM基地址(低字节)** RAM基址的最低 8 位用于存储 ADC 连续采样数据。 RAM基地址 ADCBA[11:0] = {ADCBAH[3:0], ADCBAL[7:0]}

ADCSN – ADC 采样数

寄存器	特殊功能寄存器地址	复位值
ADCSN	8DH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
ADCSN[7:0]							
读/写							

Bit	Name	Description
7:0	**ADCSN[7:0]**	**ADC采样数** ADC 继续采样选择的总采样数。 总采样数= ADCSN[7:0] + 1

ADCCN – ADC当前采样数

寄存器	特殊功能寄存器地址	复位值
ADCCN	8EH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
ADCCN[7:0]							
读/写							

位	名称	描述
7:0	**ADCCN[7:0]**	**ADC当前采样数** ADC 继续采样选择的当前采样数。 当前采样数 = ADCCN[7:0] + 1

ADCSR – ADC状态寄存器

寄存器	特殊功能寄存器地址	复位值
ADCSR	8FH, SFR页2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	CMPHIT	HDONE	FDONE
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
7:4	-	保留
2	CMPHIT	ADC比较器命中标志 此位仅在 ADCMPO (ADCCON2.4) 标志上升沿时由硬件置位。 注意：此位可通过写0清除。
1	HDONE	A/D转换半完成标志 此位仅在 ADCSN 在连续模式下完成一半 A/D 转换时由硬件置位。 注意：此位可通过写0清除。
0	FDONE	A/D转换完全完成标志 此位仅在 ADCSN 在连续模式下完成所有 A/D 转换或单次模式下完成单次转换时由硬件置位。 注意：此位可通过写0清除。

6.14 指令集

CM1003系列可执行标准80C51系列的所有指令，且完全兼容 MCS-51。但由于采用高性能1T 8051内核，因此每条指令的执行周期略有不同。该架构消除了冗余总线状态，并实现了读取、解码和运行的并行执行。CM1003每个机器周期使用一个时钟周期。对比同样时钟频率工作的传统12T 80C51器件，它的性能提高了8.1倍(以 MIPS 为单位)。当然，实际的速度提升将取决于指令组合编译的结果。

所有指令都编码成一个8位字段，称为操作码OPCODE。这个单字节从程序存储器中获取。OPCODE由CPU解码。它决定CPU将采取什么操作以及是否需要从存储器中获取更多操作数据。如果不需要其他数据，则只需要一个字节。这样的指令称为单字节指令。在某些应用下需要更多数据，即两字节或三字节指令。

6.14.1 指令集和寻址模式

表6.14-3 列出了所有指令的详细信息。指令集和寻址方式的注释如下所示。

指令	寻址模式
Rn (N = 0~7)	当前选定的寄存器组的寄存器R0至R7
Direct	8位内部数据单元的地址。它可以是内部数据RAM位置(00H至7FH)或者特殊功能寄存器SFR(80H至FFH).
@RI (I = 0, 1)	通过寄存器R0或R1间接寻址的8位内部数据RAM位置(00H至FFH)。
#data	指令中的8位常数
#data16	指令中的16位常数
Addr16	16位目标地址，由LCALL和LJMP使用。分支可以在程序存储地址空间的任意位置。
Addr11	11位目标地址，由ACALL和AJMP使用。 分支将位于与后续指令的第一个字节在相同的2K字节程序存储页面中。
Rel	有符号的(2的补码)8位偏移字节. 由SJMP和所有条件分支指令使用。范围是后续指令的第一个字节的-128到+127字节区域内。
Bit	内部数据RAM 或特殊功能寄存器 SFR的直接寻址位。

表 6.14-1 指令集和寻址模式

指令	CY	OV	AC	指令	CY	OV	AC
ADD	X[1]	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	0	X		ANL C, /bit	X		
DIV	0	X		ORL C, bit	X		
DA A	X			ORL C, /bit	X		
RRC A	X			MOV C, bit	X		
RLC A	X			CJNE	X		
SETB C	1						
注意： X表示根据指令的执行结果变化。							

表 6.14-2 指令影响标志位设置

6.14.2 读-修改-写指令

从特殊功能寄存器 (SFR) 或内部 RAM 读取一个字节，修改该字节，然后将其重写回的指令称为“读-修改-写”指令。当目标是 I/O 端口或端口位时，这些指令读取内部输出锁存器，而不是外部引脚状态。此类指令读取端口 SFR 的值，修改该值，然后将其写回端口 SFR。所有“读-修改-写”指令列表如下。

指令	说明
ANL	逻辑与。 (ANL 直接寻址, A和ANL 直接寻址, #立即数)
ORL	逻辑或。 (ORL 直接寻址, A 和ORL 直接寻址, #立即数)
XRL	逻辑异或。 (XRL 直接寻址, A 和XRL 直接寻址, #立即数)
JBC	如果位 = 1 ，则跳转并清除该位。 (JBC 位寻址, rel)
CPL	对位求反。 (CPL 位寻址)
INC	递增。 (INC 直接寻址)
DEC	递减。(DEC 直接寻址)
DJNZ	递减，如果不为零则跳转。 (DJNZ 直接寻址, rel)
MOV	bit, C 将进位移动到位。 (MOV 位寻址, C)
CLR	bit 清除位。 (CLR 位寻址)
SETB	bit 设置位。 (SETB 位寻址)

最后三条指令虽然看起来不像是“读-修改-写”指令，但实际上是的。它们读取整个端口锁存器的值，修改要改变的位，然后将新值写回端口锁存器。

6.14.3 指令集列表

指令	操作码	字节	时钟周期	CM1003 V.S. 传统的 80C51 速率
NOP	00	1	1	12
ADD A, Rn	28~2F	1	2	6
ADD A, direct	25	2	3	4
ADD A, @Ri	26, 27	1	4	3
ADD A, #data	24	2	2	6
ADDC A, Rn	38~3F	1	2	6
ADDC A, direct	35	2	3	4
ADDC A, @Ri	36, 37	1	4	3
ADDC A, #data	34	2	2	6
SUBB A, Rn	98~9F	1	2	6
SUBB A, direct	95	2	3	4
SUBB A, @Ri	96, 97	1	4	3
SUBB A, #data	94	2	2	6
INCA	04	1	1	12
INCRn	08~0F	1	3	4
INCdirect	05	2	4	3

指令	操作码	字节	时钟周期	CM1003 V.S. 传统的 80C51 速率
INC@Ri	06, 07	1	5	2.4
INCDPTR	A3	1	1	24
DEC A	14	1	1	12
DEC Rn	18~1F	1	3	4
DEC direct	15	2	4	3
DEC @Ri	16, 17	1	5	2.4
MUL AB	A4	1	4	12
DIV AB	84	1	4	12
DA A	D4	1	1	12
ANL A, Rn	58~5F	1	2	6
ANL A, direct	55	2	3	4
ANL A, @Ri	56, 57	1	4	3
ANL A, #data	54	2	2	6
ANL direct, A	52	2	4	3
ANL direct, #data	53	3	4	6
ORL A, Rn	48~4F	1	2	6
ORL A, direct	45	2	3	4
ORL A, @Ri	46, 47	1	4	3
ORL A, #data	44	2	2	6
ORL direct, A	42	2	4	3
ORL direct, #data	43	3	4	6
XRL A, Rn	68~6F	1	2	6
XRL A, direct	65	2	3	4
XRL A, @Ri	66, 67	1	4	3
XRL A, #data	64	2	2	6
XRL direct, A	62	2	4	3
XRL direct, #data	63	3	4	6
CLR A	E4	1	1	12
CPL A	F4	1	1	12
RL A	23	1	1	12
RLC A	33	1	1	12
RR A	03	1	1	12
RRC A	13	1	1	12
SWAP A	C4	1	1	12
MOV A, Rn	E8~EF	1	1	12
MOV A, direct	E5	2	3	4
MOV A, @Ri	E6, E7	1	4	3
MOV A, #data	74	2	2	6
MOV Rn, A	F8~FF	1	1	12
MOV Rn, direct	A8~AF	2	4	6
MOV Rn, #data	78~7F	2	2	6

指令	操作码	字节	时钟周期	CM1003 V.S. 传统的 80C51 速率
MOV direct, A	F5	2	2	6
MOV direct, Rn	88~8F	2	3	8
MOV direct, direct	85	3	4	6
MOV direct, @Ri	86, 87	2	5	4.8
MOV direct, #data	75	3	3	8
MOV @Ri, A	F6, F7	1	3	4
MOV @Ri, direct	A6, A7	2	4	6
MOV @Ri, #data	76, 77	2	3	6
MOV DPTR, #data16	90	3	3	8
MOVC A, @A+DPTR	93	1	4	6
MOVC A, @A+PC	83	1	4	6
MOVX A, @Ri[1]	E2, E3	1	5	4.8
MOVX A, @DPTR[1]	E0	1	4	6
MOVX @Ri, A[1]	F2, F3	1	6	4
MOVX @DPTR, A[1]	F0	1	5	4.8
PUSH direct	C0	2	4	6
POP direct	D0	2	3	8
XCH A, Rn	C8~CF	1	2	6
XCH A, direct	C5	2	3	4
XCH A, @Ri	C6, C7	1	4	3
XCHD A, @Ri	D6, D7	1	5	2.4
CLR C	C3	1	1	12
CLR bit	C2	2	4	3
SETB C	D3	1	1	12
SETB bit	D2	2	4	3
CPL C	B3	1	1	12
CPL bit	B2	2	4	3
ANL C, bit	82	2	3	8
ANL C, /bit	B0	2	3	8
ORL C, bit	72	2	3	8
ORL C, /bit	A0	2	3	8
MOV C, bit	A2	2	3	4
MOV bit, C	92	2	4	6
ACALL addr11	11, 31, 51, 71, 91, B1, D1, F1[2]	2	4	6
LCALL addr16	12	3	4	6
RET	22	1	5	4.8
RETI	32	1	5	4.8
AJMP addr11	01, 21, 41, 61, 81, A1, C1, E1[3]	2	3	8
LJMP addr16	02	3	4	6

指令	操作码	字节	时钟周期	CM1003 V.S. 传统的 80C51 速率
SJMP rel	80	2	3	8
JMP @A+DPTR	73	1	3	8
JZ rel	60	2	3	8
JNZ rel	70	2	3	8
JC rel	40	2	3	8
JNC rel	50	2	3	8
JB bit, rel	20	3	5	4.8
JNB bit, rel	30	3	5	4.8
JBC bit, rel	10	3	5	4.8
CJNE A, direct, rel	B5	3	5	4.8
CJNE A, #data, rel	B4	3	4	6
CJNE Rn, #data, rel	B8~BF	3	4	6
CJNE @Ri, #data, rel	B6, B7	3	6	4
DJNZ Rn, rel	D8~DF	2	4	6
DJNZ direct, rel	D5	3	5	4.8

注意：
[1] CM1003没有外部存储器总线结构。使用MOVX指令来访问内部XRAM。
[2] 11位地址[A10:A8]的最高三位决定ACALL十六进制码。代码为[A10,A9,A8,1,0,0,0,1]。
[3] 11位地址[A10:A8]的最高三位决定AJMP 十六进制码。代码为 [A10,A9,A8,0,0,0,0,1]。

表 6.14-3 指令集

7 应用电路

7.1 供电方案

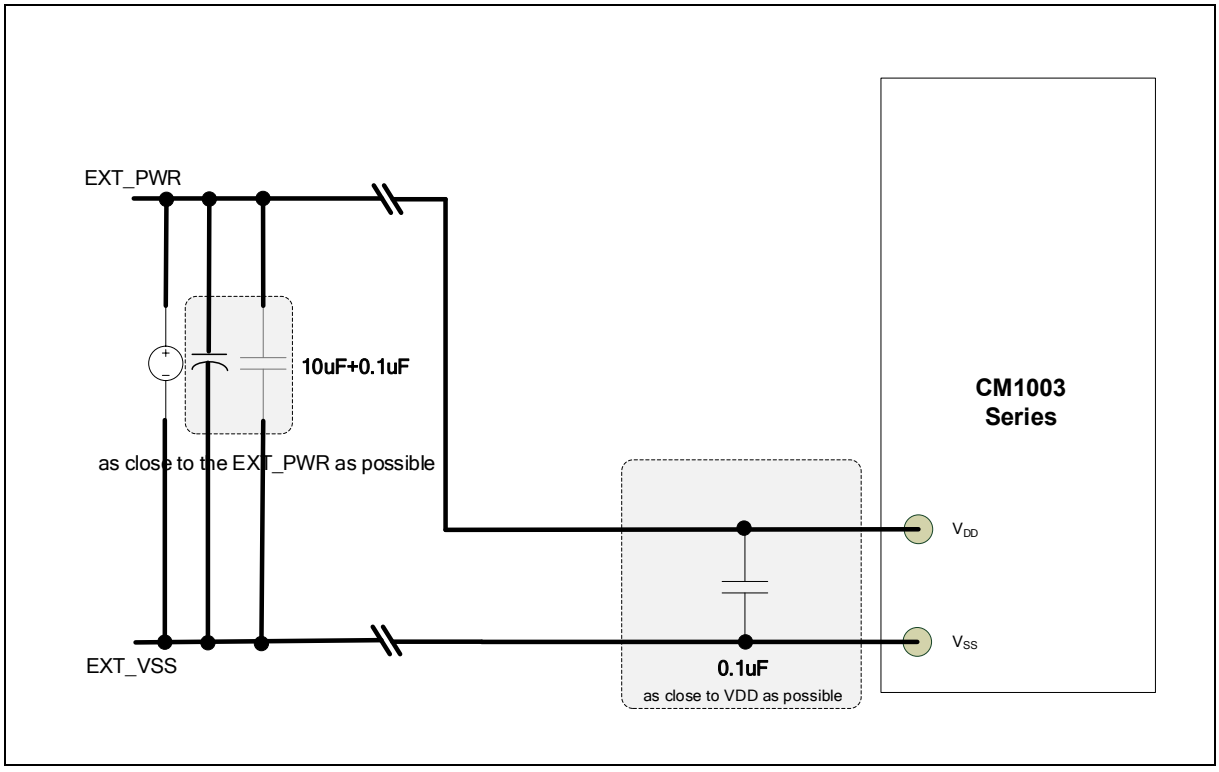
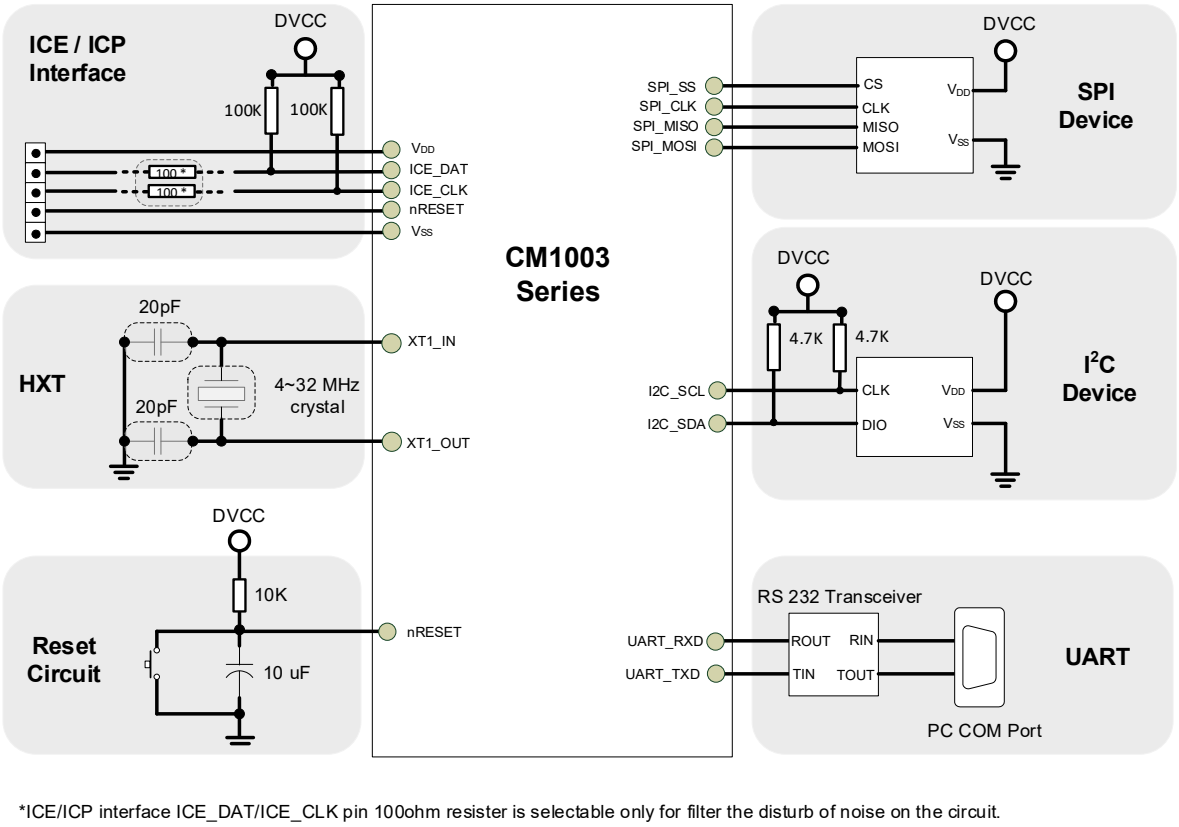


图 7.1-1 NuMicro CM1003 供电电路

7.2 外设应用方案



注1: 建议在ICE_DAT和ICE_CLK引脚上都用100kΩ上拉电阻。

注2: 建议在nRESET复位脚上使用10kΩ上拉电阻和 10uF电容。

注3: 可选择在ICE_DAT/ICE_CLK和烧写引脚间添加100ohm串联电阻以滤除噪声干扰。

图 7.2-1 NuMicro CM1003 外设接口电路

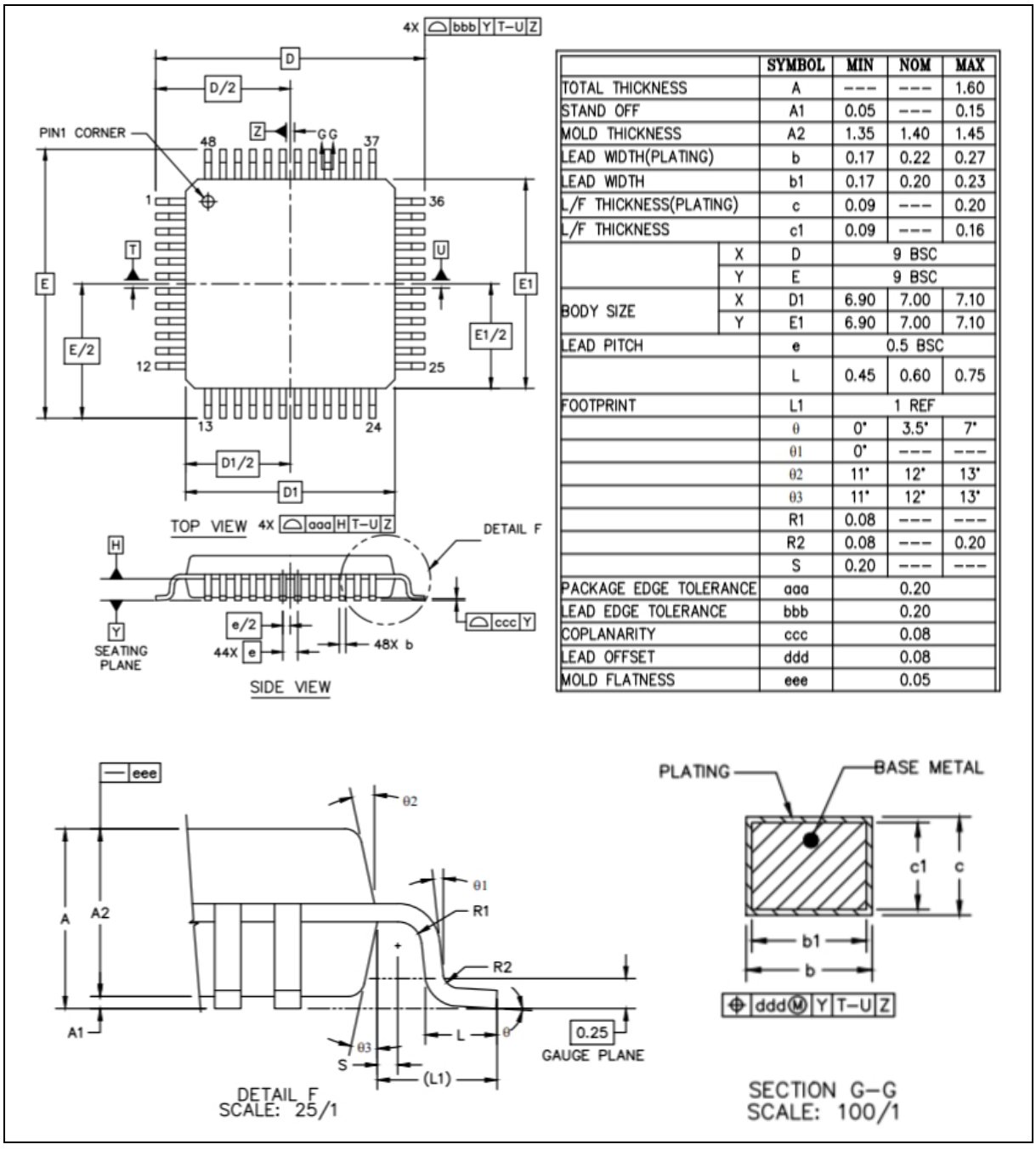
8 电气特性

有关 CM1003 电气特性的详细信息，请参阅相关数据手册。

9 封装尺寸

封装不含卤素，符合 RoHS 和 TSCA 标准。

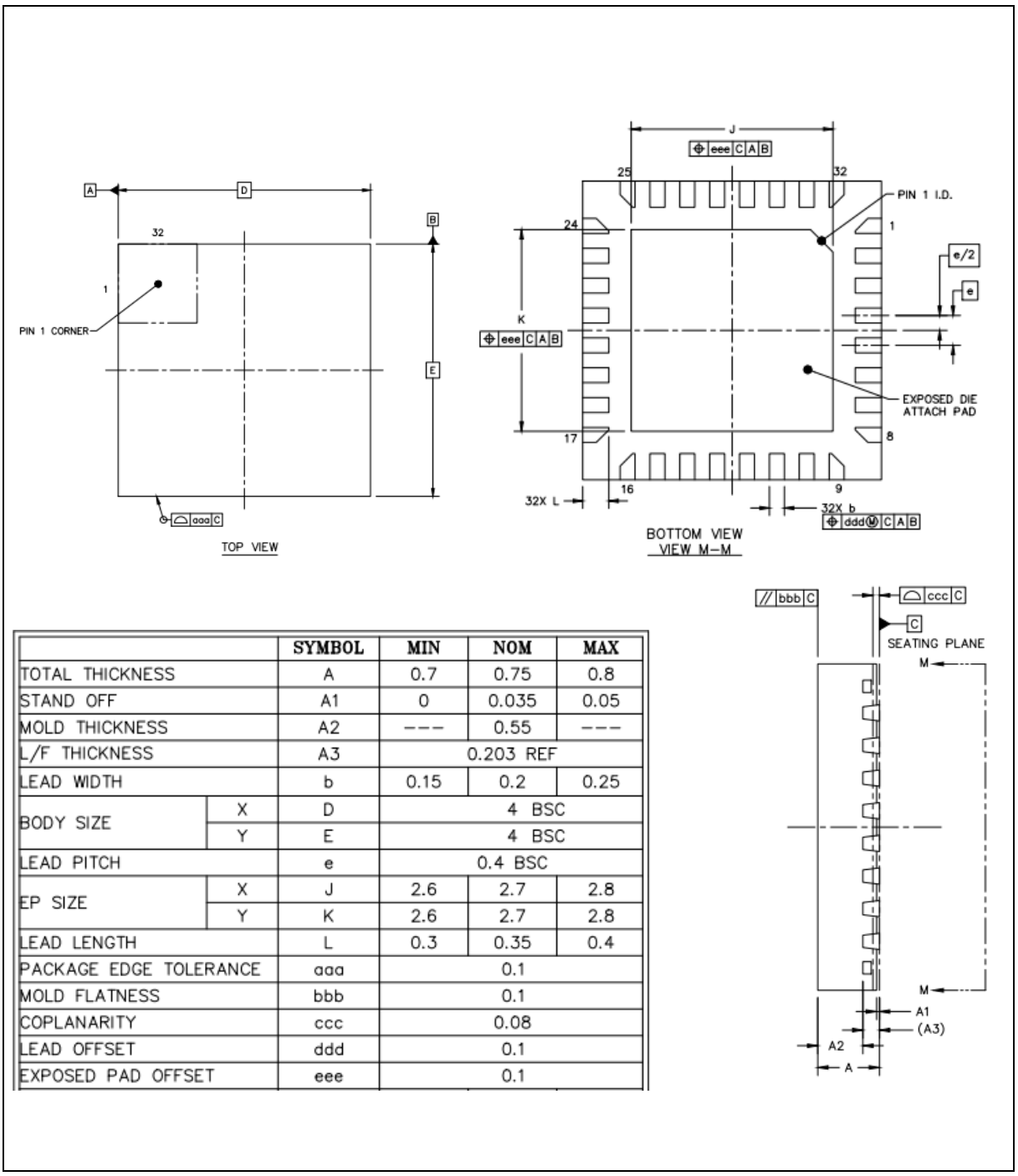
9.1 LQFP 48引脚 (7.0 x 7.0 x 1.4 mm³)



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	---	---	1.60
STAND OFF		A1	0.05	---	0.15
MOLD THICKNESS		A2	1.35	1.40	1.45
LEAD WIDTH(PLATING)		b	0.17	0.22	0.27
LEAD WIDTH		b1	0.17	0.20	0.23
L/F THICKNESS(PLATING)		c	0.09	---	0.20
L/F THICKNESS		c1	0.09	---	0.16
BODY SIZE	X	D	9 BSC		
	Y	E	9 BSC		
	X	D1	6.90	7.00	7.10
	Y	E1	6.90	7.00	7.10
LEAD PITCH		e	0.5 BSC		
		L	0.45	0.60	0.75
FOOTPRINT		L1	1 REF		
		θ	0°	3.5°	7°
		θ1	0°	---	---
		θ2	11°	12°	13°
		θ3	11°	12°	13°
		R1	0.08	---	---
		R2	0.08	---	0.20
		S	0.20	---	---
PACKAGE EDGE TOLERANCE		aaa	0.20		
LEAD EDGE TOLERANCE		bbb	0.20		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.08		
MOLD FLATNESS		eee	0.05		

图 9.1-1 LQFP-48 封装尺寸

9.2 QFN 33引脚(4.0 x 4.0 x 0.8 mm³)



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.55	---
L/F THICKNESS		A3		0.203 REF	
LEAD WIDTH		b	0.15	0.2	0.25
BODY SIZE	X	D		4 BSC	
	Y	E		4 BSC	
LEAD PITCH		e		0.4 BSC	
EP SIZE	X	J	2.6	2.7	2.8
	Y	K	2.6	2.7	2.8
LEAD LENGTH		L	0.3	0.35	0.4
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		bbb		0.1	
COPLANARITY		ccc		0.08	
LEAD OFFSET		ddd		0.1	
EXPOSED PAD OFFSET		eee		0.1	

图 9.2-1 QFN-33 封装尺寸

10 缩写词

10.1 缩写词列表

Acronym	Description
ADC	模数转换器Analog-to-Digital Converter
BOD	掉电检测Brown-out Detection
GPIO	通用输入/输出General-Purpose Input/Output
Fsys	系统时钟频率Frequency of system clock
HIRC	12MHz内部高速RC振荡器12 MHz Internal High Speed RC Oscillator
IAP	在应用编程In Application Programming
ICP	在电路编程In Circuit Programming
ISP	在系统编程In System Programming
LIRC	10kHz内部低速RC振荡器10 kHz internal low speed RC oscillator (LIRC)
LVR	低压复位Low Voltage Reset
PDMA	外设直接存储器访问Peripheral Direct Memory Access
POR	上电复位Power On Reset
PWM	脉冲宽度调制Pulse Width Modulation
SPI	串行外设接口Serial Peripheral Interface
UART	通用异步收发器Universal Asynchronous Receiver/Transmitter
UCID	唯一客户标识符 IDUnique Customer ID
WKT	唤醒定时器Wakeup Timer
WDT	看门狗定时器Watchdog Timer

表 10.1-1 缩写词列表

11 版本历史

日期	版本	修订章节	描述
2025-06-24	1.00		初始版本
2025-10-11	1.01	章节 2	更新章节2中的时钟描述
		章节3.2	更新3.2 CM1003系列命名规则
2026-01-19	1.02	章节4.2.2	更新CM1003系列引脚功能描述
		章节1，3.1，3.3 4.1，4.1.2，9	移除 CM1003DJ8AE 和 LQFP32
2026-06-17	1.03	章节3.3	修正了 CM1003DL8AE 的 SRAM 容量

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.