NuMicro® 系列

基于1T 8051微控制器

CM1003 系列

CM1003BF2AE
CM1003BF3AE
CM1003CF2AE
CM1003CF3AE

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 主要特点和应用 9

2 特性 10

3 料号信息 13

3.1 CM1003系列封装型号 13

3.2 CM1003系列命名规则 13

3.3 CM1003系列选型指南 13

4 引脚配置 14

4.1 CM1003 引脚配置 14

4.1.1 TSSOP 20引脚封装 14

4.1.2 QFN 20引脚封装 15

4.2 CM1003 引脚描述 16

5 功能框图 19

5.1 CM1003 系列功能框图 19

6 功能描述 20

6.1 存储架构 20

6.1.1 程序存储 20

6.1.2 数据闪存(Data Flash) 22

6.1.3 安全保护存储器(SPROM) 23

6.1.4 配置字节(CONFIG) 23

6.1.5 数据存储(RAM) 29

6.1.6 特殊功能寄存器(SFR) 31

6.2 系统管理 161

6.2.1 时钟系统 161

6.2.2 电源管理 166

6.2.3 重置系统 167

6.2.4 中断系统 176

6.3 Flash 存储控制 180

6.3.1 在应用编程(IAP) 180

6.3.2 在线电路编程(ICP) 191

6.3.3 片上调试(OCD) 191

6.3.4 96位唯一标识符(UCID) 192

6.4 通用端口(GPIO) 194

6.4.1 GPIO 模式 194

6.4.2 寄存器描述.....197
6.4.3 读取-修改-写指令.....201
6.4.4 外部中断引脚.....202
6.4.5 引脚中断.....212
6.5 定时器.....216
6.5.1 定时器/计数器0和1.....216
6.5.2 定时器2和输入捕获.....224
6.5.3 定时器3.....236
6.6 看门狗定时器(WDT).....239
6.6.1 概述.....239
6.6.2 功能描述.....239
6.7 自唤醒定时器(WKT).....243
6.7.1 概述.....243
6.7.2 框图.....243
6.7.3 寄存器描述.....243
6.8 串口(UART0 和 UART1).....245
6.8.1 概述.....245
6.8.2 功能描述.....245
6.8.3 寄存器描述.....253
6.9 串行外设接口(SPI).....265
6.9.1 概述.....265
6.9.2 结构框图.....265
6.9.3 功能描述.....265
6.9.4 寄存器描述.....272
6.10 内部集成电路(I^2C).....276
6.10.1 概述.....276
6.10.2 功能描述.....276
6.10.3 寄存器描述.....287
6.11 脉冲宽度调制(PWM).....295
6.11.1 概述.....295
6.11.2 功能描述.....295
6.11.3 寄存器描述.....302
6.12 12位模数转换器(ADC).....318
6.12.1 概述.....318
6.12.2 框图.....318
6.12.3 功能描述.....319
6.12.4 寄存器描述.....321
6.13 指令集.....331

7 应用电路 335
7.1 供电方案 335
7.2 外设应用方案 336
8 电气特性 337
9 封装信息 338
9.1 TSSOP 20 引脚(4.4 x 6.5 x 0.9 毫米) 338
9.2 QFN 20 引脚(3.0 x 3.0 x 0.8 毫米) 339
10 缩写词列表 340
10.1 缩写词列表 340
11 历史版本信息 341

图目录

图4.1-1 QFN-20封装的引脚图 15
图5.1-1 CM1003 系列功能框图 19
图6.1-1 CM1003 16KB 程序存储映射图和开机选择 21
图6.1-2 CM1003 32KB 程序存储映射图和开机选择 22
图6.1-3 SPROM存储区映射和SPROM保护模式 23
图6.1-4 CONFIG0 复位后自动重载 24
图6.1-5 CONFIG2 上电复位重载位 26
图6.1-6 数据存储映射图 29
图6.1-7 内部256 字节RAM 寻址 30
图6.2-1 时钟系统框图 161
图6.2-2 上电和复位时序 168
图6.2-3 启动选择结构图 174
图6.4-1 准双向模式结构 195
图6.4-2 推挽模式结构 195
图6.4-3 仅输入模式结构 196
图6.4-4 开漏模式结构 196
图6.4-5 引脚接口框图 212
图6.5-1 定时器/计数器 0和1 处于模式0 216
图6.5-2 定时器/计数器 0和1 处于模式1 217
图6.5-3 定时器/计数器 0和1 处于模式2 217
图6.5-4 定时器/计数器0和1处于模式3 218
图6.5-5 定时器2完整功能框图 224
图6.5-6 定时器2自动重载模式框图 225
图6.5-7 定时器2比较模式框图 226
图6.5-8 定时器3框图 236
图6.6-1 看门狗作为一个超时复位定时器 240
图6.6-2 看门狗定时器框图 240
图6.7-1 自唤醒定时器框图 243
图6.8-1 串口模式0时序图 245
图6.8-2 串口模式1时序图 246
图6.8-3 串口模式2模式3的传输时序 247
图6.9-1 SPI结构框图 265
图6.9-2 SPI 多主机，多从机连接图 266

图6.9-3 SPI 单主机，单从机连接图 266
图6.9-4 SPI 时钟格式 268
图6.9-5 CPHA = 0 的SPI时钟和数据格式 269
图6.9-6 CPHA = 1 的SPI时钟和数据格式 270
图6.9-7 SPI 溢出波形 271
图6.9-8 SPI 中断请求 272
图6.10-1 I²C 总线互联 276
图6.10-2 I²C 总线协议 277
图6.10-3 START起始条件, Repeated START重复起始条件和STOP停止条件 277
图6.10-4 主机以 7 位数据格式向从机发送数据 277
图6.10-5 主机以 7 位数据格式向从机读取数据 278
图6.10-6 一次I²C传输的数据格式 278
图6.10-7 应答位 279
图6.10-8 两台主机的仲裁过程 279
图6.10-9 根据当前I²C 状态来控制I²C总线 280
图6.10-10 主机发送模式流程和状态 281
图6.10-11 主机接收模式流程和状态 281
图6.10-12 从机接收模式流程和状态 282
图6.10-13 广播呼叫模式的流程和状态 283
图6.10-14 I²C 超时计数器 284
图6.10-15 保持时间延长使能 289
图6.11-1 PWM框图 296
图6.11-2 PWM和故障刹车输出控制框图 297
图6.11-3 PWM边沿对齐型波形 298
图6.11-4 PWM中心对齐型波形 299
图6.11-5 带死区时间插入的PWM互补模式 300
图6.11-6 故障刹车功能框图 300
图6.11-7 PWM中断类型 301
图6.12-1 12位ADC框图 318
图6.12-2 外部触发 ADC电路 320
图6.12-3 ADC结果比较器 320
图7.1-1 NuMicro CM1003供电电路 335
图7.2-1 NuMicro CM1003外设接口电路 336
图9.1-1 TSSOP-20 封装尺寸 338

图9.2-1 QFN-20 封装尺寸 339

表格目录

表1.1-1 NuMicro CM1003系列主要功能支持表 9
表4.1-1 TSSOP-20封装的引脚图 14
表4.2-1 脚位描述 18
表6.1-1 CM1003 特殊功能寄存器地址列表 39
表6.2-1 中断向量 176
表6.2-2 中断优先级层级设置 177
表6.2-3 各中断源特性表 178
表6.3-1 IAP模式和命令代码 181
表6.4-1 不同I/O模式的配置 194
表6.6-1 看门狗定时器在不同分频下的超时间隔 239
表6.8-1串口UART0模式/波特率描述 248
表6.8-2 串口UART1模式/波特率描述 249
表6.9-1 从机引脚配置选择 267
表6.9-2 SPI 主机时钟速率表 268
表6.9-3 SPI 时钟暂停间隔选择 268
表6.9-4 从机选择引脚配置 270
表6.13-1 影响标志位的指令 331
表6.13-2 指令集 334
表10.1-1 缩写词列表 340

1 概述

NuMicro CM1003系列是一款增强型1T-8051内核微控制器。指令集与标准的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32K闪存，256字节的RAM和1KB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系列还提供丰富的外设，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，一组SPI，一组I²C，6路增强型PWM输出，8通道共享所有I/O引脚中断，以及8路12位可达500kSPS的ADC。

支持小型封装包括两种类型，TSSOP-20引脚和QFN 20-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以及 NuEclipse SDCC编译平台。

1.1 主要特点和应用

产品线	UART	I²C	SPI	Timer	PWM	ADC
CM1003BF2AE CM1003BF3AE CM1003CF2AE CM1003CF3AE	2	1	1	4	6	8

表 1.1-1 NuMicro CM1003 系列主要功能支持表

CM1003系列适用于多种应用，例如:

- 家用电器
- 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8位高性能1T 8051内核CMOS微控制器 • 指令集与MCS-51完全兼容 • 4级优先级中断功能 • 双数据指针(DPTRs)
上电复位(POR)	• 上电复位阈值1.15V
欠压监测(BOD)	• 具有掉电中断和复位功能4阶BOD (4.4V / 3.7V / 2.7V / 2.2V)
低电压复位(LVR)	• LVR电压阈值2.0V
安全	• 96位唯一芯片标识符(UID) • 128位唯一客户标识符(UCID) • 128字节加密保护存储SPROM(仅适用于32KB/16KB APROM)
存储器 / 记忆体	
闪存(Flash)	• 16/32KB应用程序存储空间(APROM) • 4/3/2/1 KB 片上用户引导程序空间(LDROM)，通过 APROM 配置，用于在系统编程(ISP) • 在应用编程(IAP)功能支持APROM执行128字节的连续页更新，因此整个APROM可作为Data Flash使用 • 额外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256B的SRAM • 内嵌额外 1KB RAM(XRAM)，通过MOVX 指令访问
时钟	
内置时钟源	• 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UART的波特率
看门狗	• 一组6位看门狗定时器(WDT) • 由WDT_CLK = 10 kHz(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
WKT唤醒定时器	• 一组16位向上计数自唤醒功能定时器(WKT) • 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6路输出选择 • 支持最大时钟源频率可达 24 MHz • 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 • 支持每路PWM引脚的屏蔽功能和三态使能 • PWM0模块支持8位分辨率的死区插入时间 • PWM0模块支持刹车功能 • PWM0模块支持触发ADC
模拟接口	
模数转换(ADC)	• 模拟输入电压范围:0~ AV_{DD} • 12位分辨率和10位精度保证 • 多达8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 • 最快500 KSPS 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
通信接口	
通用异步收发器(UART)	• 支持2路标准UART: UART0, UART1 • 全双工异步通信

	• 支持可编程的第9位 • UART0可通过软件配置TXD与RXD引脚交换功能
I²C	• 1 组I²C 设备 • 主机/从机模式 • 主从机双向传输模式 • 7位寻址模式 • 标准模式(100 kbps)和快速模式(400 kbps) • 内嵌8位超时定时器，当I²C总线挂起且定时器超时溢出时，产生I²C中断 • 支持保持时间可改
串行外设接口(SPI)	• 支持一组 SPI设备 • 支持主机或从机模式操作 • 支持MSB或LSB传送顺序 • 从机模式最高可达 12 MHz
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IO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 $\overline{\text{INT0}}$ 及$\overline{\text{INT1}}$ • 支持电流推和灌驱动的 I/O • 最大 I/O 速度为 24 MHz • 支持唤醒功能
静电放电(ESD)&电快速放电(EFT)	
静电放电(ESD)	• HBM达 ± 7 kV
电快速放电(EFT)	• ± 4.4 kV
闩锁效应(Latch-up)	• ± 200 mA

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
料号	CM1003BF2AE CM1003CF2AE	CM1003BF3AE CM1003CF3AE

3.2 CM1003系列命名规则

C	M	1	003	B	F	2	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	Flash(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		
CM1003BF3AE	16	1K+256	4	18	4	6	-	2	1	1	8-ch	QFN20
CM1003BF2AE	16	1K+256	4	18	4	6	-	2	1	1	8-ch	TSSOP20
CM1003CF3AE	32	1K+256	4	18	4	6	-	2	1	1	8-ch	QFN20
CM1003CF2AE	32	1K+256	4	18	4	6	-	2	1	1	8-ch	TSSOP20

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. 本技术参考手册仅针对16/32 KB Flash的料号产品。

4 引脚配置

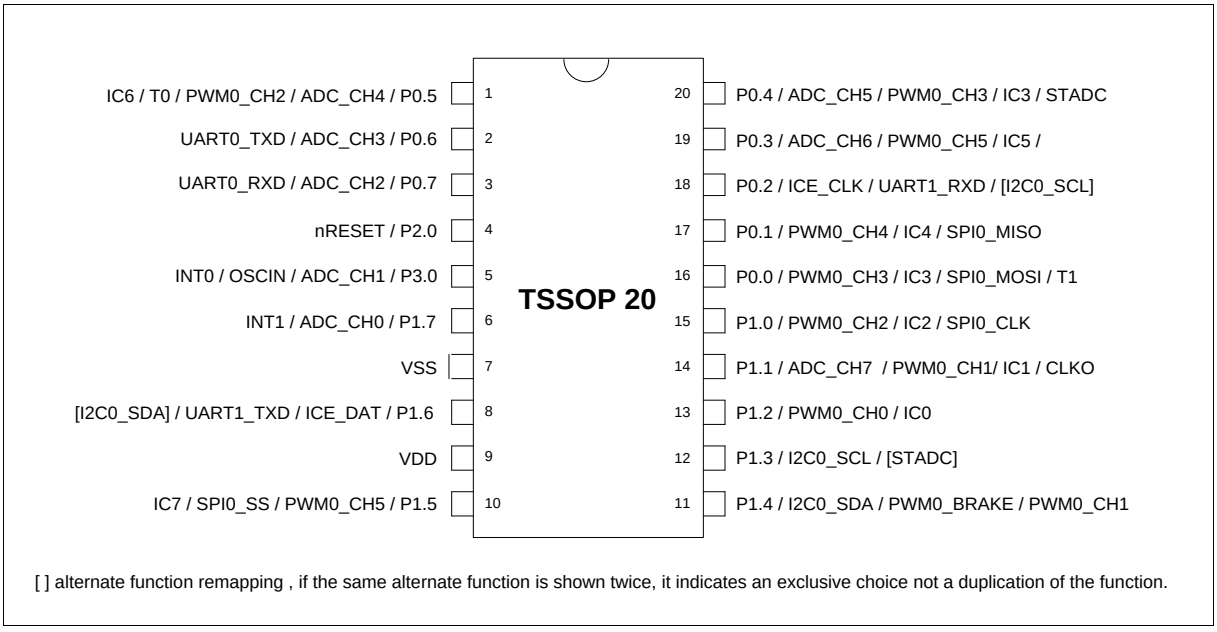
用户可以在CM1003多功能引脚图章节或使用 NuTool – PinConfigure找到引脚配置信息。NuTool - PinConfigure包含NuMicro® 家族所有芯片系列的所有型号, 帮助用户方便正确的配置GPIO多功能。

4.1 CM1003 引脚配置

4.1.1 TSSOP 20引脚封装

相关料号:

CM1003BF2AE, CM1003CF2AE



[ ] alternate function remapping , if the same alternate function is shown twice, it indicates an exclusive choice not a duplication of the function.

表 4.1-1 TSSOP-20 封装的引脚图

4.1.2 QFN 20引脚封装

相关料号:

CM1003BF3AE, CM1003CF3AE

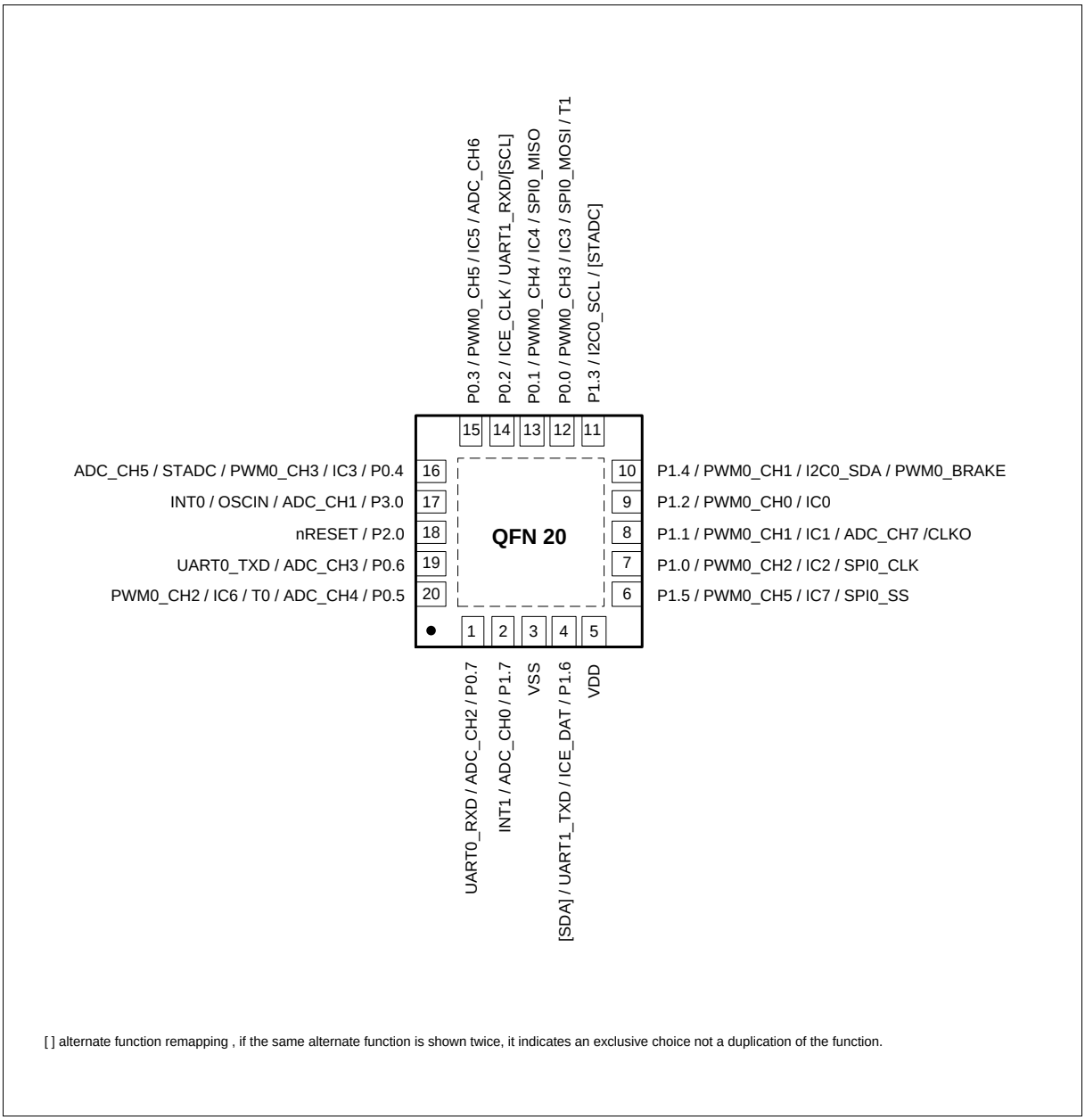


图 4.1-1 QFN-20 封装的引脚图

4.2 CM1003 引脚描述

引脚编号		符号	多功能描述[1]
CM1003BF2AE CM1003CF2AE	CM1003BF3AE CM1003CF3AE		
9	5	VDD	电源：提供工作电压VDD
7	3	VSS	电源地: 地电位
16	12	P0.0	端口0引脚0
		PWM0_CH3	PWM0 输出通道3
		SPI0_MOSI	SPI 主机输出/从机输入脚
		IC3	输入捕获通道3
		T1	定时器/计数器1，外部计数输入脚或翻转输出脚.
17	13	P0.1	端口0引脚1
		PWM0_CH4	PWM0 输出通道4
		IC4	输入捕获通道4
		SPI0_MISO	SPI 主机输入/从机输出脚
18	14	P0.2	端口0引脚2.
		ICE_CLK	调试及ICP编程时钟输入脚.
		UART1_RXD	串口1数据接收输入脚
		[I2C0_SCL] [3]	I^2C时钟脚.
19	15	P0.3	端口0引脚3
		ADC_CH6	ADC输入通道6
		PWM0_CH5	PWM0 输出通道5
		IC5	输入捕获通道5
20	16	P0.4	端口0引脚4
		PWM0_CH3	PWM0 输出通道3
		IC3	输入捕获通道3
		ADC_CH5	ADC 输入通道5
		STADC	外部启动ADC 触发脚
1	20	P0.5	端口0引脚5
		PWM0_CH2	PWM0 输出通道2
		IC6	输入捕获通道6
		T0	定时器/计数器0，外部计数输入脚或翻转输出脚
		ADC_CH4	ADC输入通道4
2	19	P0.6	端口0引脚6
		UART0_TXD [2]	串口0数据发送脚

引脚编号		符号	多功能描述[1]
CM1003BF2AE CM1003CF2AE	CM1003BF3AE CM1003CF3AE		
		ADC_CH3	ADC 输入通道3
3	1	P0.7	端口0引脚7
		UART0_RXD	串口0数据接收输入脚
		ADC_CH2	ADC 输入通道2
15	7	P1.0	端口1引脚0
		PWM0_CH2	PWM0 输出通道2
		IC2	输入捕获通道2
		SPI0_CLK	SPI 时钟脚
14	8	P1.1	端口1引脚1
		PWM0_CH1	PWM0 输出通道1
		IC1	输入捕获通道1
		ADC_CH7	ADC 输入通道7
		CLKO	系统时钟输出脚
13	9	P1.2	端口1引脚2
		PWM0_CH0	PWM0 输出通道0
		IC0	输入捕获通道0
12	11	P1.3	端口1引脚3
		I2C0_SCL	I^2C 时钟脚
		[STADC] [4]	外部启动ADC触发脚
11	10	P1.4	端口1引脚4
		PWM0_CH1	PWM0 输出通道1.
		I2C0_SDA	I^2C 数据脚
		PWM0_BRAKE	故障刹车输入脚
10	6	P1.5	端口1引脚5
		PWM0_CH5	PWM0 输出通道 5
		IC7	输入捕获通道7
		SPI0_SS	SPI 从机选择输入脚
8	4	P1.6	端口1引脚6
		ICE_DAT	ICP / 调试编程数据输入输出脚
		UART1_TXD/	串口1数据发送脚
		[I2C0_SDA] [3]	I^2C 数据脚
6	2	P1.7	端口1引脚7

引脚编号		符号	多功能描述[1]
CM1003BF2AE CM1003CF2AE	CM1003BF3AE CM1003CF3AE		
		$\overline{INT1}$	外部中断1输入脚
		ADC_CH0	ADC 输入通道0
4	18	P2.0/	端口2引脚0输入脚， RPD(CONFIG0.2)配置为0时可用
		nRESET	nRESET引脚是用于硬件设备复位的施密特触发器输入引脚。此引脚的低电平会复位芯片。nRESET引脚有一个内部上拉电阻，只需将外部电容器连接到地，即可进行通电复位。
5	17	P3.0	端口3引脚0，使用内部晶振当系统时钟时可用
		ADC_CH1	ADC 输入通道1
		$\overline{INT0}$	外部中断0输入
		OSCIN	使用 ECLK 模式时，OSCIN为外部时钟输入脚。

注意:

1. 所有引脚都可以配置为外部中断脚位，此功能未列入引脚描述列表。
2. UART0的 TXD 及 RXD 引脚可通过配置寄存器UART0PX(AUXR1.2)交换。
3. [I^2C] 备用功能重分配选项。 I^2C引脚可通过配置寄存器 I2CPX(I2CON.0)切换。
4. [STADC] 备用功能重分配选项。 STADC引脚可通过配置寄存器STADCPX(ADCCON1.6)切换。
5. PIOx 寄存器决定哪一个引脚是PWM 或 GPIO功能。

表 4.2-1 脚位描述

5 功能框图

5.1 CM1003 系列功能框图

图5.1-1 CM1003 CM1003 功能模块框图及设备的概述。用户可以在图中找到设备的所有外设功能。

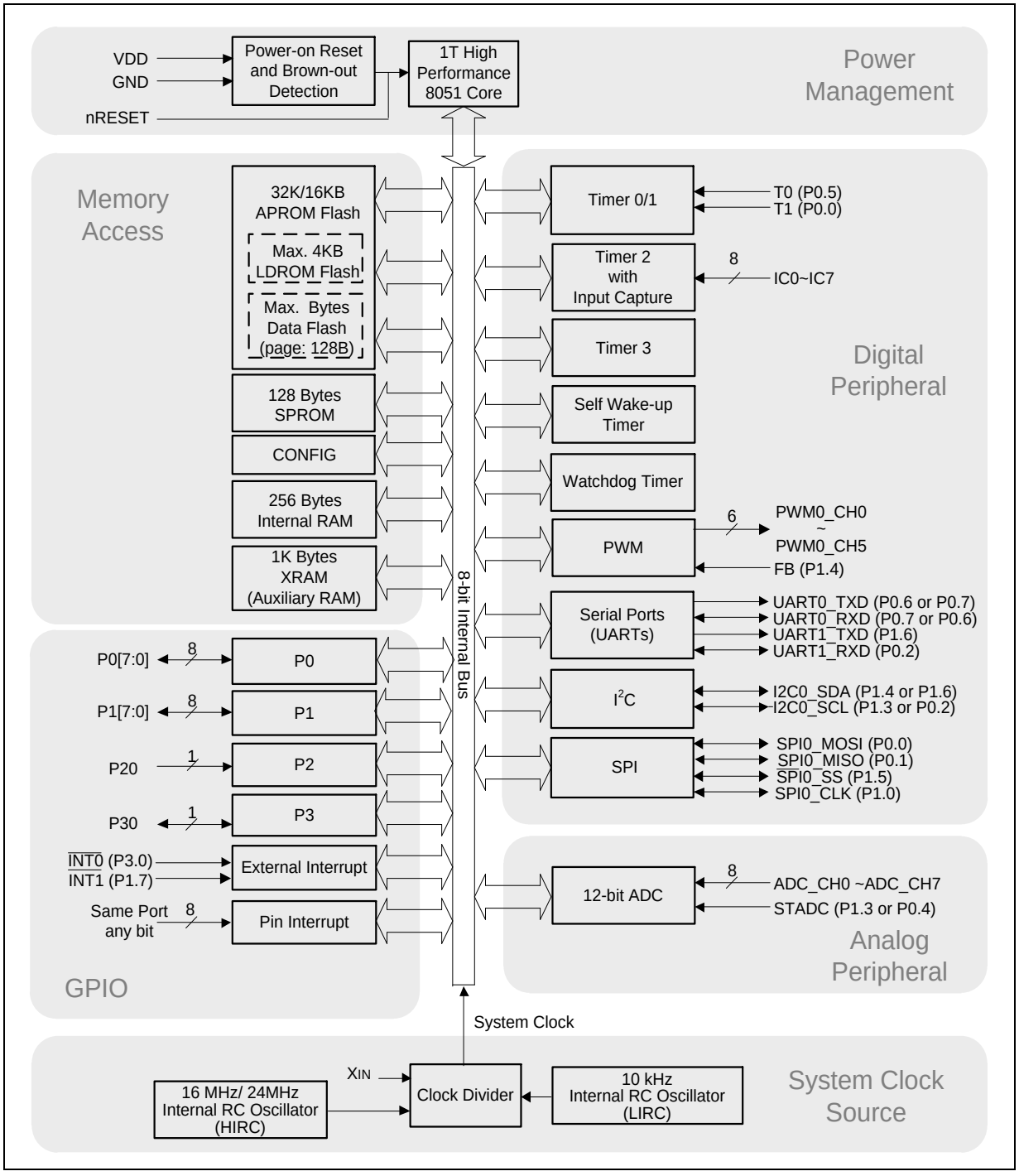


图 5.1-1 CM1003 系列功能框图

6 功能描述

6.1 存储架构

标准的基于80C51微控制器将存储分成两个不同的部分，程序存储和数据存储。程序存储用来存储指令代码。而数据存储用来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字节内部RAM。对于需要更多内部RAM的许多应用， CM1003 提供片上另外1K字节 的RAM，叫做XRAM，通过MOVX指令访问。

整个嵌入式FLASH作为程序存储器，被分成四块。应用ROM(APROM)通常存储用户代码，引导ROM(LDROM)通常存储引导代码，安全保护存储器(SPROM)通常用于存储安全应用的指令，CONFIG字节作用于硬件初始化。事实上，APROM和 LDROM功能相似，但是大小不一样，每一块由一页一页组成，每页128字节。FLASH控制单元支持擦除、编程和读模式。外部烧写器通过指定的I/O口烧写，在应用编程(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。

6.1.1 程序存储

程序存储存储用于执行的程序代码。如图6.1-1 CM1003 16KB 。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32/16KB。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。

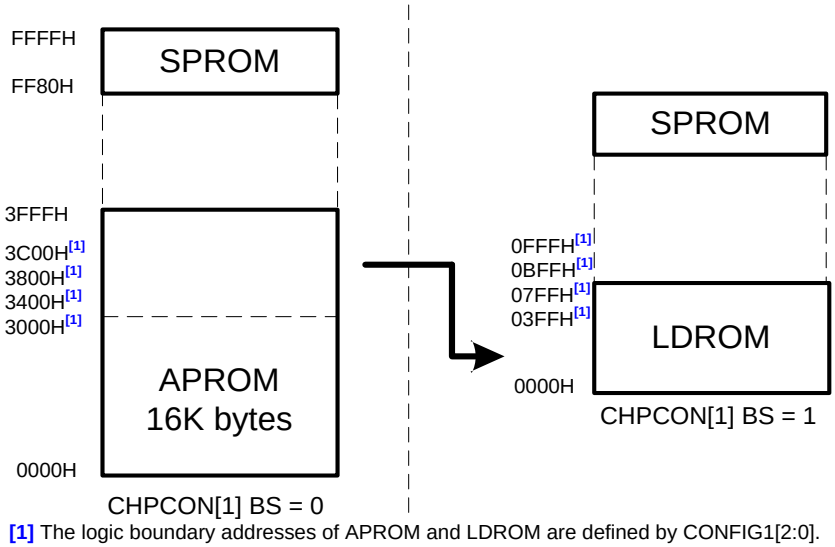
另一个独立的程序存储块称为LDROM。LDROM的常规功能是存储ISP的引导代码。它可以更新APROM空间和CONFIG配置字节。APROM中的代码也可以重新编程LDROM。有关 ISP 的详细信息以及与 APROM 和 LDROM 相关的配置位设置，请参阅第 6.3.1.5 章在系统编程(ISP). 请注意，APROM和LDROM 是独立的硬件模块，如果CPU从LDROM重新启动，CPU会自动将程序计数器0000H重新向量化到LDROM的起始地址。因此CPU将LDROM视为独立的程序存储且所有中断向量独立于APROM。

CONFIG1

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

出厂预设值: 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。 (CM1003BF2AE, CM1003BF3AE) 111 = 无 LDROM。APROM 为 16K 字节。 110 =LDROM为 1K 字节。APROM 为 15K 字节。 101 =LDROM为 2K 字节。APROM 为 14K 字节。 100 =LDROM为 3K 字节。APROM 为 13K 字节。. 0xx =LDROM为 4K 字节。APROM为 12K 字节。 (CM1003CF2AE, CM1003CF3AE) 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。



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

图 6.1-1 CM1003 16KB 程序存储映射图和开机选择

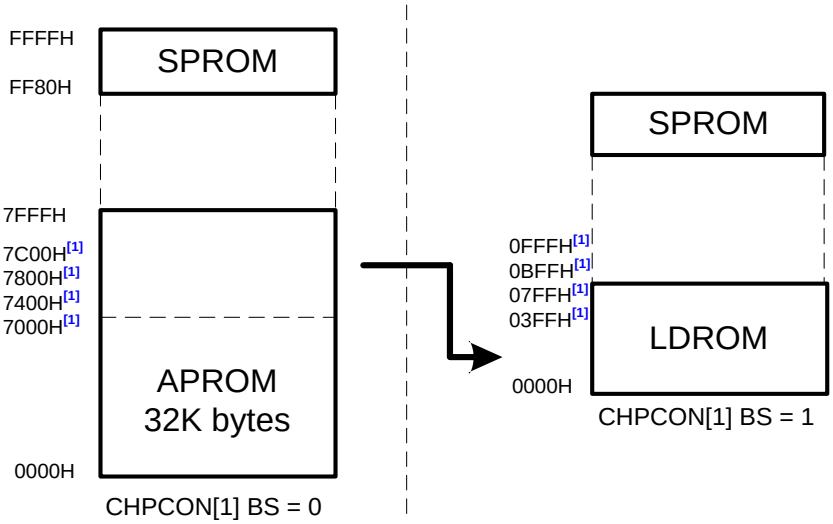


图 6.1-2 CM1003 32KB 程序存储映射图和开机选择

6.1.2 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，请参阅第 6.3 章节FLASH存储控制的在Flash 存储控制

在应用编程(IAP)。所有嵌入式FLASH页擦除为128字节。

6.1.3 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。

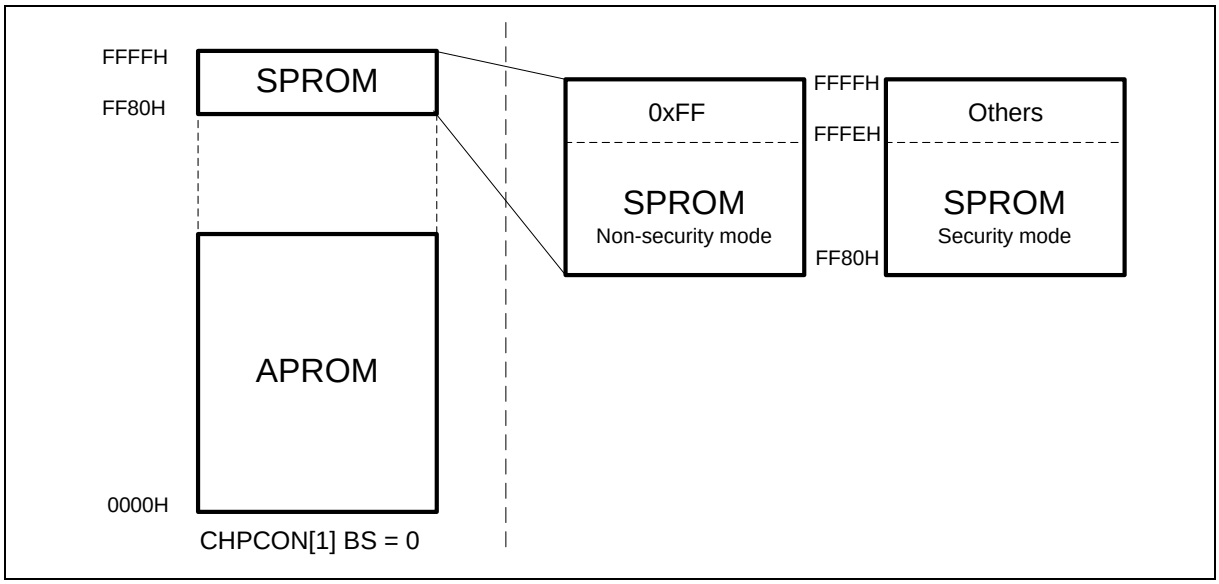


图 6.1-3 SPROM 存储区映射和 SPROM 保护模式

(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。

6.1.4 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	-

出厂预设值: 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。
[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P2.0。
[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。

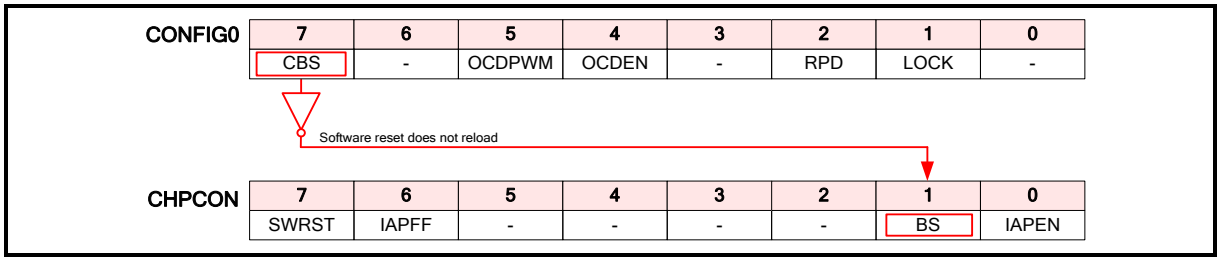


图 6.1-4 CONFIG0 复位后自动重载

CONFIG1

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

出厂预设值: 1111 1111b

位	名称	描述
[2:0]	**LDSIZE[2:0]**	**LDROM 容量选择** 适用于CM1003BF2AE, CM1003BF3AE 111 = 无 LDROM。APROM 为 16K 字节。 110 =LDROM为 1K 字节。APROM 为 15K 字节。 101 =LDROM为 2K 字节。APROM 为 14K 字节。 100 =LDROM为 3K 字节。APROM 为 13K 字节。. 0xx =LDROM为 4K 字节。APROM为 12K 字节。 适用于CM1003CF2AE, CM1003CF3AE 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。

CONFIG2

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

出厂预设值: 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.
[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。

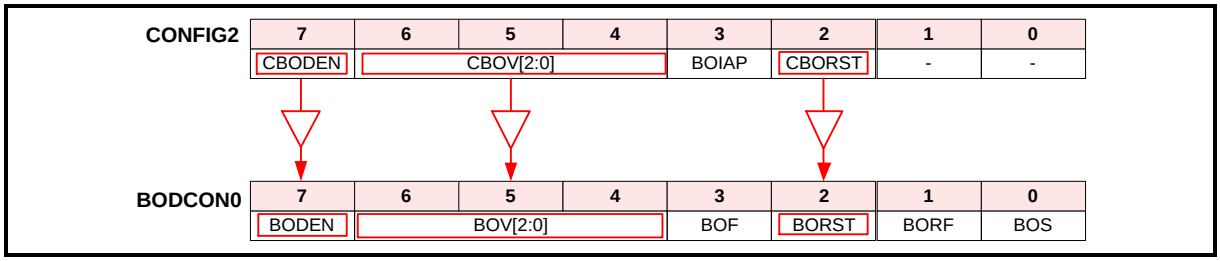


图 6.1-5 CONFIG2 上电复位重载位

CONFIG3

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

出厂预设值:1111 1111b

位	名称	描述
7:4	-	保留
3	ADCINTEN	**ADC 比较模式下启用ADC中断标志位设置** 1 = ADC 标志位 ADCF(ADCCON0.7)仅在比较结果匹配时设置为1。 选择此模式后，ADC的行为与MS51相同。 0 = 在每个ADC覆盖完成时设置ADC标志。 选择此模式后，ADC的行为与N76E003相同。
2:0	-	保留

CONFIG4

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

出厂预设值: 1111 1111b

位	名称	描述
[7:4]	**WDTEN[3:0]**	**看门狗定时器WDT使能** 此字段配置MCU执行后的WDT行为。 1111 = 看门狗定时器禁用，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保持运行。
[3:0]	-	保留

6.1.5 数据存储(RAM)

6.1.5.1 内部数据存储

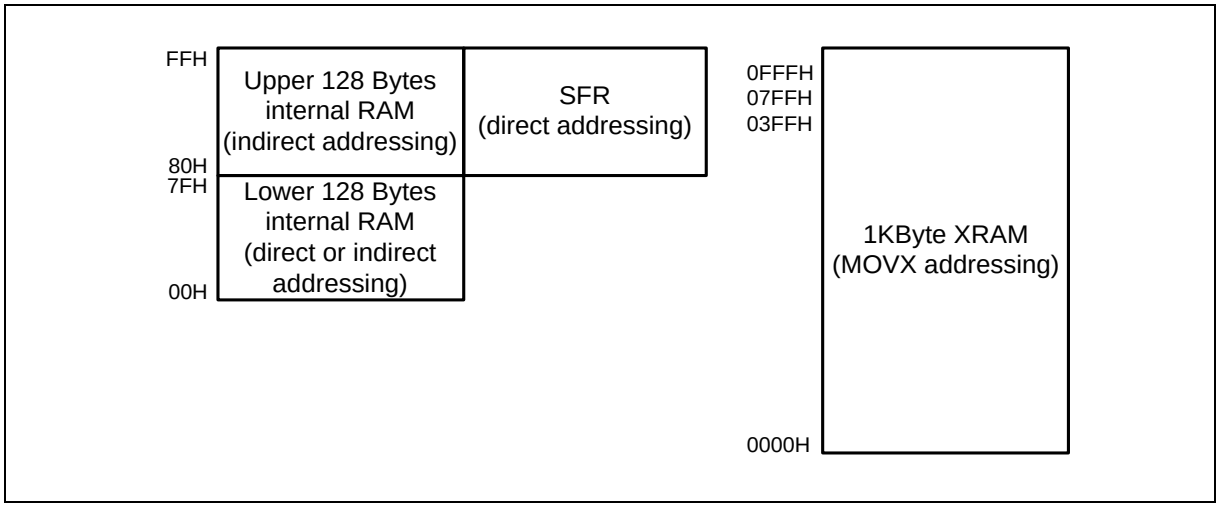


图 6.1-6 数据存储映射图

图6.1-6 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。

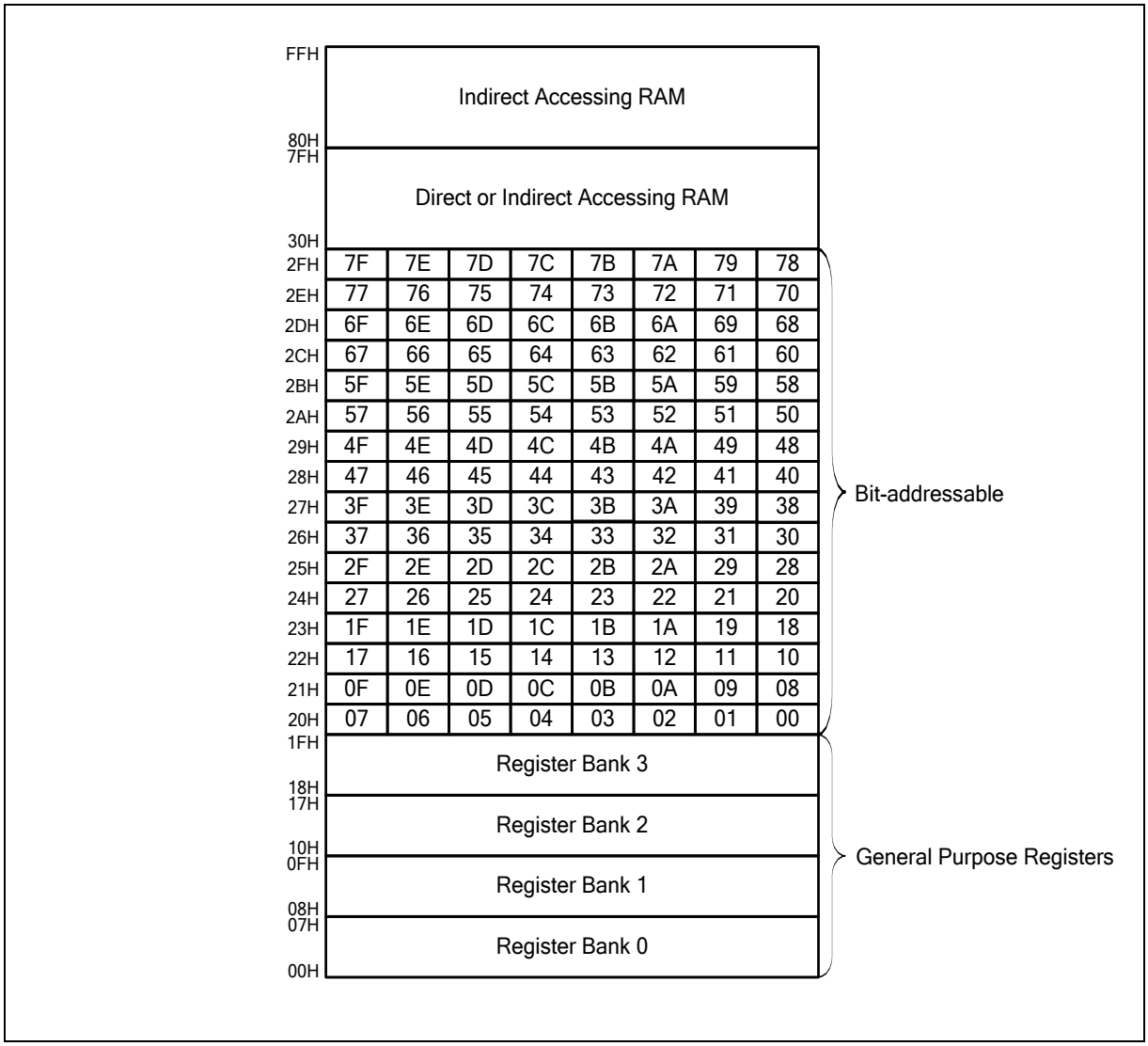


图 6.1-7 内部 256 字节 RAM 寻址

6.1.5.2 *片上 XRAM*

CM1003 提供额外的1K字节片上附加RAM叫做XRAM来扩大RAM的空间。它占用从 00H 到 FFH 的地址空间。这1K字节的 XRAM 可通过调用外部指令 MOVX @DPTR 或 MOVX @Ri 来间接访问。(参见下面的代码示例。)请注意，堆栈指针不能位于 XRAM 的任何部分。

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
```

6.1.6 特殊功能寄存器(SFR)

CM1003使用特殊功能寄存器(SFR)来控制和监视外设及其模式。SFR 位于地址80到FFH的地址空间，仅可以通过直接寻址访问。那些以0H或8H结尾的SFR是可位寻址的。当用户需要直接修改特定位而不通过位域指令更改其他位时，位寻址非常有用。其他所有的 SFR 只能字节寻址。CM1003包含标准 8051 中存在的所有 SFR，此外还内置了一些额外的 SFR。因此，原始 8051 中的一些未使用的字节被赋予了新功能。SFR 如下所列。

6.1.6.1 *SFR 页选择*

为了在 0x80 至 0xFF 地址空间中容纳超过 128 个 SFR，实现了 SFR 分页。默认情况下，所有 SFR 访问目标都是SFR页0 。在设备初始化期间，可能需要访问位于 SFR 页1 的某些 SFR。寄存器 SFRS 用于切换 SFR 寻址页。

SFRS – SFR 页选择

寄存器	地址	复位值
SFRS	91H，所有页，TA保护	0000_0000b

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

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

切换 SFR 页参考例程:

```
MOV     TA,#0AAH        ;切换到SFR第1页
MOV     TA,#55H
MOV     SFRS,#01H

MOV     TA,#0AAH        ;切换到SFR第0页
MOV     TA,#55H
MOV     SFRS,#00H
```

6.1.6.2 定时访问保护(TA)

CM1003 有几项功能，例如看门狗和欠压检测，这些功能对于系统的正常运行非常重要。如果对这些控制寄存器不进行保护，错误的代码可能往它们写入不确定的值，导致操作错误和失控。为了避免这种风险，CM1003 有一个保护机制，对关键 SFR 限制写访问。 该保护机制使用定时访问(TA)。下列寄存器与 TA 过程相关。

TA – 定时访问

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

地址: C7H，所有页 复位值: 0000 0000b

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。

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，则暂时禁用
MOV   TA,#0AAH
MOV   TA,#55H
(写入 TA保护寄存器的指令)
(SETBEA)                ;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。

6.1.6.3 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。

```
      MOVR0,#64          ;要移动的字节数
      MOVDPTR,#D_Addr    ;加载目标地址
      INCAUXR1           ;改变激活的DPTR
      MOVDPTR,#S_Addr    ;加载源地址
LOOP：
      MOVXA,@DPTR        ;读取源数据字节
      INCAUXR1           ;将 DPTR 更改为目标地址
      MOVX@DPTR,A        ;将数据写入目标
      INCDPTR            ;下一个目标地址
      INCAUXR1           ;将 DPTR 更改为源
      INCDPTR            ;下一个源地址
      DJNZR0,LOOP
      INCAUXR1           ;(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	地址	复位值
DPL	82H，所有页	0000_0000b

7	6	5	4	3	2	1	0
DPL[7:0]							
读写							

位	名字	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	地址	复位值
DPH	83H，所有页	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	地址	复位值
AUXR1	A2H，所有页	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。

6.1.6.4 SFR 地址列表

	Addr	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0	F8	SCON_1	PDTEN	PDTCNT	PMEN	PMD	PORDIS	EIP1	EIPH1
1							-		-
0	F0	B	CAPCON3	CAPCON4	SPCR SPCR2	SPSR	SPDR	AINDIDS	EIPH
1							-	-	-
0	E8	ADCCON0	PICON	PINEN	PIPEN	PIF	C2L	C2H	EIP
1		-							-
0	E0	ACC	ADCCON1	ADCCON2	ADCDLY	C0L	C0H	C1L	C1H
1			-	-					
0	D8	PWMCON0	PWMPL	PWM0L	PWM1L	PWM2L	PWM3L	PIOCON0	PWMCON1
1									
0	D0	PSW	PWMPH	PWM0H	PWM1H	PWM2H	PWM3H	PNP	FBD
1									
0	C8	T2CON	T2MOD	RCMP2L	RCMP2H	TL2 PWM4L	TH2 PWM5L	ADCMPL	ADCMPH
1								-	-
0	C0	I2CON	I2ADDR	ADCRL	ADCRH	T3CON PWM4H	RL3 PWM5H	RH3 PIOCON1	TA
1				-	-				
0	B8	IP	SADEN	SADEN_1	SADDR_1	I2DAT	I2STAT	I2CLK	I2TOC
1									
0	B0	P3	P0M1 P0S	P0M2 P0SR	P1M1 P1S	P1M2 P1SR	P2S	-	IPH PWMINTC
1									
0	A8	IE	SADDR	WDCON	BODCON1	P3M1 P3S	P3M2 P3SR	IAPFD	IAPCN
1									
0	A0	P2	-	AUXR1	BODCON0	IAPTRG	IAPUEN	IAPAL	IAPAH
1									
0	98	SCON	SBUF	SBUF_1	EIE	EIE1	-	-	CHPCON
1									
0	90	P1	SFRS	CAPCON0	CAPCON1	CAPCON2	CKDIV	CKSWT	CKEN
1									
0	88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	WKCON
1									
0	80	P0	SP	DPL	DPH	RCTRIM0	RCTRIM1	RWK	PCON
1									

注意: 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。

表 6.1-1 CM1003 特殊功能寄存器地址列表

6.1.6.5 SFR 位描述和复位值

CM1003

符号	定义	地址(页)	MSB							LSB[1]	复位值[2]
EIPH1	扩展中断优先级1高位	FFH(0)	-	-	-	-	-	PWKTH	PT3H	PSH_1	0000 0000b
EIP1	扩展中断优先级1	FEH(0)	-	-	-	-	-	PWKT	PT3	PS_1	0000 0000b
PMD	PWM 掩码数据	FCH	-	-	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0	0000 0000b
PMEN	PWM 掩码使能	FBH	-	-	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0	0000 0000b
PDTCNT [4]	PWM 空载时间计数器	FAH	PDTCNT[7:0]								0000 0000b
PDTEN[4]	PWM 空载时间使能	F9H	-	-	-	PDTCNT.8	-	PDT45EN	PDT23EN	PDT01EN	0000 0000b
SCON_1	串口1 控制	F8H	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	PT2H	PSPIH	PFBH	PWDTH	PPWMH	PCAPH	PPIH	PI2CH	0000 0000b
AINDIDS	ADC 数字输入禁用	F6H	P11DIDS	P03DIDS	P04DIDS	P05DIDS	P06DIDS	P07DIDS	P30DIDS	P17DIDS	0000 0000b
SPDR	SPI 数据	F5H(0)	SPDR[7:0]								0000 0000b
SPSR	SPI 状态	F4H	SPIF	WCOL	SPIOVF	MODF	DISMODF	TXBUF	-	-	0000 0000b
SPCR	SPI 控制	F3H(0)	SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR[1:0]		0000 0000b
SPCR2	SPI 控制2	F3H(1)	-	-	-	-	-	-	SPIS[1:0]		0000 0000b
CAPCON4	输入捕获控制4	F2H	-	-	-	-	CAP23	CAP22	CAP21	CAP20	0000 0000b
CAPCON3	输入捕获控制3	F1H	CAP13	CAP12	CAP11	CAP10	CAP03	CAP02	CAP01	CAP00	0000 0000b
B	B 寄存器	F0H	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	PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI2C	0000 0000b
C2H	输入捕获2高字节	EEH	C2H[7:0]								0000 0000b
C2L	I输入捕获2低字节	EDH	C2L[7:0]								0000 0000b
PIF	引脚中断标志	ECH	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	0000 0000b
PIPEN	引脚中断高电平/上升沿使能	EBH	PIPEN7	PIPEN6	PIPEN5	PIPEN4	PIPEN3	PIPEN2	PIPEN1	PIPEN0	0000 0000b
PINEN	引脚中断低电平/下降沿使能	EAH	PINEN7	PINEN6	PINEN5	PINEN4	PINEN3	PINEN2	PINEN1	PINEN0	0000 0000b
PICON	引脚中断控制	E9H	PIT67	PIT45	PIT3	PIT2	PIT1	PIT0ADC	PIPS[1:0]		0000 0000b
ADCCON0	ADC 控制0	E8H(0)	ADCF	ADCS	ETGSEL1	ETGSEL0	ADCHS3	ADCHS2	ADCHS1	ADCHS0	0000 0000b
C1H	输入捕获1高字节	E7H	C1H[7:0]								0000 0000b
C1L	输入捕获1低字节	E6H	C1L[7:0]								0000 0000b
C0H	输入捕获0高字节	E5H	C0H[7:0]								0000 0000b
C0L	输入捕获0低字节	E4H	C0L[7:0]								0000 0000b
ADCDLY	ADC 触发延迟	E3H	ADCDLY[7:0]								0000 0000b
ADCCON2	ADC 控制2	E2H(0)	ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	-	-	-	ADCDLY.8	0000 0000b
ADCCON1	ADC 控制1	E1H(0)	OCEN	STADCPX	ADCDIV[1:0]		ETGTYP[1:0]		ADCEX	ADCEN	0000 0000b
ACC	累加器	E0H	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	0000 0000b
PWMCON1	PWM 控制1	DFH	PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]			0000 0000b
PIOCON0	PWM I/O 开关0	DEH	-	-	PIO05	PIO04	PIO03	PIO02	PIO01	PIO00	0000 0000b
PWM3L	PWM0 3信道占空比低字节	DDH	PWM3[7:0]								0000 0000b
PWM2L	PWM0 2信道占空比低字节	DCH	PWM2[7:0]								0000 0000b
PWM1L	PWM0 1信道占空比低字节	DBH	PWM1[7:0]								0000 0000b
PWM0L	PWM0 0信道占空比低字节	DAH	PWM0[7:0]								0000 0000b
PWMPL	PWM 周期低字节	D9H	PWMP[7:0]								0000 0000b
PWMCON0	PWM0 控制寄存器0	D8H	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000b
FBD	Brake 刹车数据	D7H	FBF	FBINLS	FBD5	FBD4	FBD3	FBD2	FBD1	FBD0	0000 0000b

符号	定义	地址(页)	MSB							LSB[1]	复位值[2]
PNP	PWM 负极性	D6H	-	-	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0	0000 0000b
PWM3H	PWM0 3信道占空比高字节	D5H	PWM3[15:8]								0000 0000b
PWM2H	PWM0 2信道占空比高字节	D4H	PWM2[15:8]								0000 0000b
PWM1H	PWM0 1信道占空比高字节	D3H	PWM1[15:8]								0000 0000b
PWM0H	PWM0 0信道占空比高字节	D2H	PWM0[15:8]								0000 0000b
PWMPH	PWM 周期高字节	D1H	PWMP[15:8]								0000 0000b
PSW	程序状态字	D0H	CY	AC	F0	RS1	RS0	OV	-	P	0000 0000b
ADCMPH	ADC 比较位高字节	CFH (0)	ADCMP[11:4]								0000 0000b
ADCMPL	ADC比较位低字节	CEH	-	-	-	-	ADCMP[3:0]				0000 0000b
PWM5L	PWM0 5信道占空比低字节	CDH(1)	PWM5 [7:0]								0000 0000b
TH2	定时器2高字节	CDH(0)	TH2[7:0]								0000 0000b
PWM4L	PWM0 4信道占空比低字节	CCH(1)	PWM4[7:0]								0000 0000b
TL2	定时器2低字节	CCH(0)	TL2[7:0]								0000 0000b
RCMP2H	定时器2比较位高字节	CBH	RCMP2H[7:0]								0000 0000b
RCMP2L	定时器2比较位低字节	CAH(0)	RCMP2L[7:0]								0000 0000b
T2MOD	定时器2模式	C9H	LDEN	T2DIV[2:0]			CAPCR	CMPCR	LDTS[1:0]		0000 0000b
T2CON	定时器2控制	C8H	TF2	-	-	-	-	TR2	-	CM/$\overline{RL2}$	0000 0000b
TA	定时访问保护	C7H	TA[7:0]								0000 0000b
PIOCON1	PWM I/O 开关 1	C6H(1)	-	-	PIO15	-	PIO13	PIO12	PIO11	-	0000 0000b
RH3	定时器3重载高字节	C6H(0)	RH3[7:0]								0000 0000b
PWM5H	PWM0 信道5占空比高字节	C5H(1)	PWM5 [15:8]								0000 0000b
RL3	定时器3重载低字节	C5H(0)	RL3[7:0]								0000 0000b
PWM4H	PWM0 信道4占空比高字节	C4H(1)	PWM4[15:8]								0000 0000b
T3CON	定时器3控制	C4H(0)	SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS[2:0]			0000 0000b
ADCRH	ADC 结果高字节	C3H (0)	ADCR[11:4]								0000 0000b
ADCRL	ADC 结果低字节	C2H (0)	-	-	-	-	ADCR[3:0]				0000 0000b
I2ADDR	I²C 从机地址	C1H	I2ADDR[7:1]							GC	0000 0000b
I2CON	I²C 控制	C0H	**I2CHTEX**	I2CEN	STA	STO	SI	AA	-	I2CPX	0000 0000b
I2TOC	I²C 超时计数器	BFH	-	-	-	-	-	I2TOCEN	DIV	I2TOF	0000 0000b
I2CLK	I²C 时钟	BEH	I2CLK[7:0]								0000 1001b
I2STAT	I²C 状态	BDH	I2STAT[7:3]					0	0	0	1111 1000b
I2DAT	I²C 数据	BCH	I2DAT[7:0]								0000 0000b
SADDR_1	从机1 地址	BBH	SADDR_1[7:0]								0000 0000b
SADEN_1	从机1 地址掩码	BAH	SADEN_1[7:0]								0000 0000b
SADEN	从机0 地址掩码	B9H	SADEN[7:0]								0000 0000b
IP	中断优先级	B8H	-	PADC	PBOD	PS	PT1	PX1	PT0	PX0	0000 0000b
PWMINTC	PWM 中断控制	B7H(1)	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b
IPH	中断优先级高	B7H(0)	-	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H	0000 0000b
P2S	P20 设置	B5H	P20UP	-	-	-	T1OE	T0OE	-	P2S.0	0000 0000b

符号	定义	地址(页)	MSB							LSB[1]	复位值[2]
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
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
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
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
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
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
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
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
P3	端口3	B0H	0	0	0	0	0	0	0	P3.0	输出, 0000 0001b 输入 0000 000Xb[3]
IAPCN	IAP 控制	AFH	IAPA[17:16]		FOEN	FCEN	FCTRL[3:0]				0011 0000b
IAPFD	IAP flash 数据	AEH	IAPFD[7:0]								0000 0000b
P3SR	P3 转换速率	ADH(1)	-	-	-	-	-	-	-	P3SR.0	0000 0000b
P3M2	P3 模式选择2	ADH(0)	-	-	-	-	-	-	-	P3M2.0	0000 0000b
P3S	P3 施密特触发器输入	ACH(1)	-	-	-	-	-	-	-	P3S.0	0000 0000b
P3M1	P3 模式选择1	ACH(0)	-	-	-	-	-	-	-	P3M1.0	0000 0001b
BODCON1[4]	掉电检测控制1	ABH	-	-	-	-	-	LPBOD[1:0]		BODFLT	上电复位, 0000 0001b 其他, 0000 0UUUb
WDCON[4]	看门狗定时器控制	AAH	WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS[2:0]			上电复位, 0000 0111b 看门狗, 0000 1UUUb 其他, 0000 UUUUb
SADDR	从机0地址	A9H	SADDR[7:0]								0000 0000b
IE	中断使能	A8H	EA	EADC	EBOD	ES	ET1	EX1	ET0	EX0	0000 0000b
IAPAH	IAP 地址高字节	A7H	IAPA[15:8]								0000 0000b
IAPAL	IAP 地址低字节	A6H	IAPA[7:0]								0000 0000b
IAPUEN[4]	IAP 更新使能	A5H	-	-	-	SPMEN	SPUEN	CFUEN	LDUEN	APUEN	0000 0000b
IAPTRG[4]	IAP 触发器	A4H	-	-	-	-	-	-	-	IAPGO	0000 0000b
BODCON0 [4]	掉电检测控制0	A3H	BODEN [5]	-	BOV[1:0][5]		BOF[6]	BORST[5]	BORF	BOS[7]	上电复位, CCCC XC0Xb 掉电复位, UUUU XU1Xb 其他, UUUU XUUXb
AUXR1	辅助寄存器1	A2H	SWRF	RSTPINF	HardF	-	GF2	UART0PX	0	DPS	上电复位, 0000 0000b 软件, 1U00 0000b nRESET 复位脚, U100 0000b 其他, UUU0 0000b
P2	端口 2	A0H	0	0	0	0	0	0	0	P2.0	输出, 0000 000Xb 输入, 0000 000Xb[3]
CHPCON[4]	芯片控制	9FH	SWRST	IAPFF	-	-	-	-	BS[5]	IAPEN	软件, 0000 00U0b 其他, 0000 00C0b

符号	定义	地址 (页)	MSB							LSB[1]	复位值[2]
RSR	复位标志寄存器	9DH	-	-	HardF	POF	RSTPINF	BORF	WDTRF	SWRF	上电复位 0001_0000b 掉电复位, 000U_0100 软件 000U_0U01 复位脚 0U0U_ 其他 000U_0U0U
EIE1	扩展中断使能1	9CH	-	-	-	-	-	EWKT	ET3	ES_1	0000 0000b
EIE	扩展中断使能	9BH	ET2	ESPI	EFB	EWDT	EPWM	ECAP	EPI	EI2C	0000 0000b
SBUF_1	串口1数据缓冲	9AH	SBUF_1[7:0]								0000 0000b
SBUF	串口0数据缓冲	99H	SBUF[7:0]								0000 0000b
SCON	串口0控制	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000b
CKEN[4]	时钟使能	97H	EXTEN[1:0]		HIRCEN	-	-	-	-	CKSWTF	0011 0000b
CKSWT[4]	时钟开关	96H	-	-	HIRCST	-	ECLKST	OSC[1:0]		-	0011 0000b
CKDIV	时钟除频器	95H	CKDIV[7:0]								0000 0000b
CAPCON2	输入捕获控制2	94H	-	ENF2	ENF1	ENF0	-	-	-	-	0000 0000b
CAPCON1	输入捕获控制1	93H	-	-	CAP2LS[1:0]		CAP1LS[1:0]		CAP0LS[1:0]		0000 0000b
CAPCON0	输入捕获控制0	92H	-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0	0000 0000b
SFRS[4]	SFR 页面选择	91H	-	-	-	-	-	-	-	SFRPSEL	0000 0000b
P1	端口 1	90H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	输出, 1111 1111b 输入 XXXX XXXXb[3]
WKCON	自唤醒定时器控制	8FH	-	-	-	WKTF	WKTR	WKPS[2:0]			0000 0000b
CKCON	时钟控制	8EH	-	PWMCKS	-	T1M	T0M	-	CLOEN	-	0000 0000b
TH1	定时器1高字节	8DH	TH1[7:0]								0000 0000b
TH0	定时器0高字节	8CH	TH0[7:0]								0000 0000b
TL1	定时器1低字节	8BH	TL1[7:0]								0000 0000b
TL0	定时器0低字节	8AH	TL0[7:0]								0000 0000b
TMOD	定时器0和1模式	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	0000 0000b
TCON	定时器0和1控制	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000b
PCON	电源控制	87H	SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL	上电复位, 0001 0000b 其他, 000U 0000b
RWK	自唤醒定时器重载字节	86H	RWK[7:0]								0000 0000b
RCTRIM1	内部 RC 调整值低字节	85H	-	-	-	HIRC24	-	-	-	HIRCTRIM[0]	0000 0000b
RCTRIM0	内部 RC 调整值高字节	84H	HIRCTRIM[8:1]								0000 0000b
DPH	数据指针高字节	83H	DPTR[15:8]								0000 0000b
DPL	数据指针低字节	82H	DPTR[7:0]								0000 0000b
SP	堆栈指针	81H	SP[7:0]								0000 0111b
P0	端口 0	80H	(P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	输出, 1111 1111b 输入, XXXX XXXXb[3]

注：

[1]( )项意思是可位寻址SFRs

[2] 复位值符号描述。 0: 逻辑 0; 1: 逻辑 1; U: 不变; C: 详见 **[5]**; X: 详见 **[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SFRs有TA写保护。

[5] 在特定复位后，根据CONFIG值，SFRs有些位有初始化设置。

[6] BOF复位值取决于CONFIG2不同的设置和V_{DD} 电压值。请检查。

[7]当欠压检测使能，BOS是只读标志，由V_{DD}值来决定。

6.1.6.6 所有SFR 描述

下面列出了所有SFR 的描述。每个 SFR 的定义也在功能 IP 章节中列出。

Pn – 端口n

寄存器	地址	复位值
P0	80H，所有页, 可位寻址	1111_1111 b
P1	90H，所有页, 可位寻址	1111_1111 b
P2	A0H，所有页, 可位寻址	0000_0001 b
P3	B0H，所有页, 可位寻址	0000_0001 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

SP – 堆栈指针

寄存器	地址	复位值
SP	81H，所有页	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指令时，它会在存储数据之前递增。请注意，SP 的默认值为 07H。因此堆栈起始位置在08H。

DPL – 数据指针低字节

寄存器	地址	复位值
DPL	82H，所有页	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构成用作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	地址	复位值
DPH	83H，所有页	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 组合用作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)位决定哪个数据指针DPTR 或 DPTR1 被激活。

RCTRIM0 – 高速内部振荡器 16 MHz Trim 0

寄存器	地址	复位值
RCTRIM0	84H，所有页，TA保护	默认16MHz HIRC值

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

RCTRIM1 – 高速内部振荡器 16 MHz Trim 1

寄存器	地址	复位值
RCTRIM1	85H，所有页，TA保护	默认16MHz HIRC值

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

RWK – 自唤醒定时器重载字节

寄存器	地址	复位值
RWK	86H，所有页	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 重载字节** 用以存储WKT的8位重载值。注意如果预除频是1/1，RWK限制不能是FFH。

PCON – 电源控制

寄存器	地址	复位值
PCON	87H， 所有页	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。参见表6.8-1串口UART0模式/波特率描述 了解详情。
[6]	**SMOD0**	**串口0 帧错误访问使能** 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，此位被置1，用以标示当前冷复位的上电复位完成。其它任何复位后，此位值保持不变，建议通过软件清0。
[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	地址	复位值
TCON	88H， 所有页，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	地址	复位值
TMOD	89H，所有页	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	地址	复位值
TL0	8AH，所有页	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** 寄存器TL0是定时器0的16位数值的低字节

TL1 – 定时器1低字节

寄存器	地址	复位值
TL1	8BH，所有页	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** 寄存器TL1是定时器1的16位数值的低字节

TH0 – 定时器0 高字节

寄存器	地址	复位值
TH0	8CH，所有页	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** 寄存器TH0是定时器0 的16位数值的高字节

TH1 – 定时器1高字节

寄存器	地址	复位值
TH1	8DH,所有页	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** 寄存器TH1是定时器1的16位数值的高字节

CKCON – 时钟控制

寄存器	地址	复位值
CKCON	8EH，所有页	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

WKCON – 自唤醒定时器控制器

寄存器	地址	复位值
WKCON	8FH，所有页	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.

SFRS – SFR 页选择

寄存器	地址	复位值
SFRS	91H，所有页, TA保护	0000_0000b

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

位	名称	描述
0	SFRPAGE	**SFR 页选择** 0 = 指令访问SFR页0. 1 = 指令访问SFR页1.

CAPCON0 – 输入捕获控制 0

寄存器	地址	复位值
CAPCON0	92H，所有页	0000_0000 b

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。

CAPCON1 – 输入捕获控制 1

寄存器	地址	复位值
CAPCON1	93H，所有页	0000_0000 b

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	地址	复位值
CAPCON2	94H，所有页	0000_0000 b

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 – 时钟除频器

寄存器	地址	复位值
CKDIV	95H，所有页	0000_0000 b

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 – 时钟切换

寄存器	地址	复位值
CKSWT	96H，所有页，TA保护	0011_0000b

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

位	名称	描述
7:6	-	保留
5	HIRCST	**高速内部振荡器 24 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
4	-	保留
3	ECLKST	**外部时钟输入状态** 0 = 外部时钟输入不稳定或禁用 1 = 外部时钟输入使能且稳定
2:1	OSC[1:0]	**振荡器选择位** 此位段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	地址	复位值
CKEN	97H，所有页, TA保护	0011 _0000 b

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

位	名称	描述
7:6	EXTEN[1:0]	**外部时钟源使能** 11 =由XIN输入的外部时钟源使能 其它 = 外部时钟源禁用 。P30作为普通I/O使用。
5	HIRCEN	**高速内部振荡器 16 MHz 使能** 0 =高速内部振荡器禁止。 1 =高速内部振荡器使能。 注意：一旦通过设置 IAPEN(CHPCON.0)使能 IAP，内部高速 16MHz 振荡器将自动使能，硬件也会设置 HIRCEN 和 HIRCST 位。清除 IAPEN 后，HIRCEN 和 EHRCST 恢复原值。
4:1	-	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 – 串行端口控制

寄存器	SFR地址	复位值
SCON	98H，所有页, 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6.8-1串口UART0模式/波特率描述了解详情。 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	
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	地址	复位值
SBUF	99H，所有页	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	地址	复位值
SBUF_1	9AH，所有页	0000_0000b

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_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。

EIE – 扩展中断使能

寄存器	地址	复位值
EIE	9BH，所有页	0000_0000b

7	6	5	4	3	2	1	0
ET2	ESPI	EFB	EWDT	EPWM	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	EPWM	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	地址	复位值
EIE1	9CH，所有页	0000_0000b

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

位	名称	描述
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	地址	复位值
RSR	9DH，所有页	上电复位, 0001_0000b 掉电复位, 000U_0100 软件, 000U_0U01 复位脚, 0U0U_ 其他 000U_0U0U

7	6	5	4	3	2	1	0
-	-	HardF (来自AUXR1.5的镜像)	POF (来自 PCON.4的镜像)	RSTPINF (来自 AUXR1.6的镜像)	BORF (来自 BODCON0.1的镜像)	WDTRF (来自 WDCON.3 的镜像)	SWRF (来自AUXR1.7的镜像)
-	-	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** CM1003中写入AUXR1.5=0 或 RSR.5=0 来清除此位。
4	POF	**来自 PCON.4的镜像** CM1003中写入PCON.4=0 或 RSR.4=0 来清除此位。
3	RSTPINF	**来自 AUXR1.6的镜像** CM1003中写入AUXR1.6=0 或 RSR.3=0 来清除此位。
2	BORF	**来自 BODCON0.1的镜像** CM1003中写入BODCON0.1=0 或 RSR.2=0 来清除此位。
1	WDTRF	**来自 WDCON.3的镜像** CM1003中写入WDCON.3=0 或 RSR.1=0 来清除此位。
0	SWRF	**来自 AUXR1.7的镜像** CM1003中写入AUXR1.7=0 或 RSR.0=0 来清除此位。

CHPCON – 芯片控制

寄存器	地址	复位值
CHPCON	9FH，所有页，TA保护	软件复位：0000_00U0b 其他：0000_00C0b

7	6	5	4	3	2	1	0
SWRST	**IAPFF**	**-**	**-**	**-**	**-**	**BS**	**IAPEN**
可写	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	地址	复位值
AUXR1	A2H，所有页	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** 此位请保持为0。
3	**GF2**	**通用标志位2** 可通过软件置位或清除该位
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DPTR后，之前未激活的DPTR寄存器内的值保持不变。

BODCON0 – 欠压检测控制0

寄存器	地址	复位值
BODCON0	A3H，所有页，TA保护	上电复位,CCCC_XC0Xb 掉电复位, UUUU_XU1Xb 其他,UUUU_XUUXb

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。		

IAPTRG – IAP 触发

寄存器	地址	复位值
IAPTRG	A4H，所有页，TA保护	0000_0000b

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	IAPGO
-	-	-	-	-	-	-	只读

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	地址	复位值
IAPUEN	A5H，所有页，TA保护	0000_0000b

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
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	地址	复位值
IAPAL	A6H，所有页	0000_0000b

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	地址	复位值
IAPAH	A7H，所有页	0000_0000b

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	地址	复位值
IE	A8H，所有页,可位寻址	0000_0000b

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。

SADDR – 从机0地址

寄存器	地址	复位值
SADDR	A9H，所有页	0000_0000b

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	地址	复位值
WDCON	AAH，所有页, TA保护	上电复位 0000_0111b 看门狗 0000_1UUUb 其他 0000_UUUUb

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	地址	复位值
BODCON1	ABH，所有页, TA保护	上电复位 0000 0001b 其他:0000 0UUUb

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	地址	复位值
IAPFD	AEH，所有页	0000_0000b

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	地址	复位值
IAPCN	AFH，所有页	0011_0000b

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	地址	复位值
P0M1	B1H, 页0	1111_1111 b
P1M1	B3H, 页0	1111_1111 b
P3M1	ACH, 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 不同I/O.

PnS – 端口n施密特触发输入

寄存器	地址	复位值
P0S	99H, 页1	0000_0000 b
P1S	9BH, 页1	0000_0000 b
P3S	ACH, 页1	0000_0000 b

7	6	5	4	3	2	1	0
P0S.7	**P0S.6**	**P0S.5**	**P0S.4**	**P0S.3**	**P0S.2**	**P0S.1**	**P0S.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
n	**P0S.n**	**P0.n 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。

PnSR – 端口n电压转换速率

寄存器	地址	复位值
P0SR	B2H, 页1	0000_0000 b
P1SR	B4H, 页1	0000_0000 b
P3SR	ADH, 页1	0000_0000 b

7	6	5	4	3	2	1	0
P0SR.7	**P0SR.6**	**P0SR.5**	**P0SR.4**	**P0SR.3**	**P0SR.2**	**P0SR.1**	**P0SR.0**
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
n	**P0SR.n**	**P0.n 电压转换速率** 0 = P0.n 正常的输出电压转换速率 1 = P0.n 高速的输出电压转换速率

P2S – P20 设置和定时器0/1的输出使能

寄存器	地址	复位值
P2S	B5H，所有页	0000_0000 b

7	6	5	4	3	2	1	0
P20UP	**-**	**-**	**-**	**T1OE**	**T0OE**	**-**	**P2S.0**
读/写	-	-	-	读/写	读/写	-	读/写

位	名称	描述
7	**P20UP**	**P2.0上拉使能位** 0 = P2.0上拉禁用 1 = P2.0上拉使能 仅当RPD(CONFIG0.2)为0时，此位有效。如果作为$\overline{\text{nRESET}}$引脚，上拉始终启用。
3	**T1OE**	**定时器1输出使能** 0 = 定时器1输出禁用。 1 = 使能定时器1输出从T1引脚。 请注意，定时器1输出仅在“定时器”模式下运行时才有效。
2	**T0OE**	**定时器0输出使能** 0 = 定时器0输出禁用。 1 = 使能定时器0输出从T0引脚。 请注意，定时器0输出仅在“定时器”模式下运行时才有效。
0	**P2S.0**	**P2.0 施密特触发输入** 0 = P2.0 TTL电平输入 1 = P2.0 施密特触发输入

IPH – 中断优先级高位

寄存器	地址	复位值
IPH	B7H, 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6.2-2 。		

PWMINTC –PWM中断控制

寄存器	地址	复位值
PWMINTC	B7H, 页1	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 – 中断优先级低位

寄存器	地址	复位值
IP	B8H，所有页，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6.2-2 。		

SADEN – 从机0地址掩码

寄存器	地址	复位值
SADEN	B9H，所有页	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	地址	复位值
SADEN_1	BAH，所有页	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	地址	复位值
SADDR_1	BBH，所有页	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	地址	复位值
I2DAT	BCH，所有页	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。

I2STAT – I²C 状态

寄存器	地址	复位值
I2STAT	BDH，所有页	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状态字** 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**	I2STAT的低有效三位始终读到 0。

I2CLK – I²C时钟

寄存器	地址	复位值
I2CLK	BEH，所有页	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时钟设定** 主机模式： 此寄存器设定作主机时I²C总线的时钟速率。算式如下： $\frac{F_{SYS}}{4 \quad (I2CLK + 1)}$ 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： 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	地址	复位值
I2TOC	BFH，所有页	0000_1001 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	地址	复位值
I2CON	C0H，所有页，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**	**I²C保持时间延长使能** 0 = I²C数据传输 I2C0_SCL保持时间延长禁用. 1 = I²C数据传输 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。
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。

位	名称	描述
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。
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。

I2ADDR – I²C自身从机地址

寄存器	地址	复位值
I2ADDR	C1H，所有页	0000_0000b

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 转换结果低字节

寄存器	地址	复位值
ADCRL	C2H，页0	0000_0000b

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

位	名称	描述
3:0	ADCR[3:0]	**ADC转换结果低字节** 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	地址	复位值
ADCRH	C3H，页0	0000_0000b

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	地址	复位值
T3CON	C4H, 页0	0000_0000b

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6.8-2 串口UART1模式/波特率描述。
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	地址	复位值
RL3	C5H, 页0	0000_0000b

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	地址	复位值
RH3	C6H, 页0	0000_0000b

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I/O选择

寄存器	地址	复位值
PIOCON1	C6H, 页1	0000_0000b

7	6	5	4	3	2	1	0
-	-	PIO15	-	PIO13	PIO12	PIO11	-
-	-	读/写	-	读/写	读/写	读/写	-

位	名称	描述
5	**PIO15**	**P1.5/PWM0_CH5引脚功能选择** 0 = P1.5/PWM0_CH5引脚用作P1.5。 1 = P1.5/PWM0_CH5引脚用作PWM0通道5输出。
3	**PIO13**	**P0.4/PWM_CH3引脚功能选择** 0 = P0.4/PWM0_CH3引脚用作 P0.4。 1 = P0.4/PWM0_CH3引脚用作PWM0通道3输出。
2	**PIO12**	**P0.5/PWM_CH2引脚功能选择** 0 = P0.5/PWM0_CH2引脚用作 P0.5。 1 = P0.5/PWM0_CH2引脚用作PWM0通道2输出。
1	**PIO11**	**P1.4/PWM0_CH1引脚功能选择** 0 = P1.4/PWM0_CH1引脚用作 P1.4。 1 = P1.4/PWM0_CH1引脚用作PWM0通道1输出。

TA – 定时访问

寄存器	地址	复位值
TA	C7H，所有页	0000_0000b

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	地址	复位值
T2CON	C8H，所有页	0000_0000b

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	地址	复位值
T2MOD	C9H，所有页	0000_0000b

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 = 将自动重载功能禁用 。 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	地址	复位值
RCMP2L	CAH，所有页	0000_0000b

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	地址	复位值
RCMP2H	CBH，所有页	0000_0000b

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	地址	复位值
TL2	CCH，所有页	0000_0000b

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	地址	复位值
TH2	CDH, 页0	0000_0000b

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	地址	复位值
ADCMPL	CEH，页0	0000_0000b

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。

ADCMPH – ADC比较值高字节

寄存器	地址	复位值
ADCMPH	CFH，页0	0000_0000b

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比较值的最高有效8位存储在此寄存器中。

– 程序状态寄存器

寄存器	地址	复位值
PSW	D0H，所有页，可位寻址	0000_0000b

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

<table>
<tr><th>位</th><th>名称</th><th>描述</th></tr>
<tr><td>7</td><td>CY</td><td>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。</td></tr>
<tr><td>6</td><td>C</td><td>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。</td></tr>
<tr><td>5</td><td>F0</td><td>用户标志0
可由用户置位或清零的通用标志0。</td></tr>
<tr><td>4</td><td>RS1</td><td rowspan="2">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
<table><tr><th>RS1</th><th>RS0</th><th>寄存器组</th><th>RAM 地址</th></tr><tr><td>0</td><td>0</td><td>0</td><td>00H 至 07H</td></tr><tr><td>0</td><td>1</td><td>1</td><td>08H 至 0FH</td></tr><tr><td>1</td><td>0</td><td>2</td><td>10H 至 17H</td></tr><tr><td>1</td><td>1</td><td>3</td><td>18H 至 1FH</td></tr></table></td></tr>
<tr><td>3</td><td>RS0</td></tr>
<tr><td>2</td><td>V</td><td>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。</td></tr>
<tr><td>1</td><td>F1</td><td>用户标志1
可由用户置位或清零的通用标志1。</td></tr>
</table>

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。

PWMPH – PWM周期高字节

寄存器	地址	复位值
PWMPH	D1H，所有页	0000_0000b

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

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

PWMnH – PWM通道0~5占空比高字节 n =0,1,2,3,4,5

寄存器	地址	复位值
PWM0H	D1H，所有页	0000 _0000 b
PWM1H	D3H，所有页	0000 _0000 b
PWM2H	D4H，所有页	0000 _0000 b
PWM3H	D5H，所有页	0000 _0000 b
PWM4H	C4H，页1	0000 _0000 b
PWM5H	C5H，页1	0000 _0000 b

7	6	5	4	3	2	1	0
PWMn [15:8], n =0,1,2,3,4,5							
读/写							

位	名称	描述
7:0	**PWMn[15:8]** **n=0,1,2,3,4,5**	**通道n占空比高字节** 此字节与PWMnL一起控制PWM发生器的输出信号PGn的占空比。

PNP –PWM负极性

寄存器	地址	复位值
PNP	D6H，所有页	0000_0000b

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	**PWMn负极性输出使能** 0 = PWMn 信号直接输出到PWMn引脚。 1 = PWMn 信号反相输出到PWMn引脚。

FBD – PWM故障刹车数据

寄存器	地址	复位值
FBD	D7H，所有页	0000_0000b

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**	**PWMn故障刹车数据** 0 = 当故障刹车有效后，PWMn信号输出为0 1 = 当故障刹车有效后，PWMn信号输出为1

PWMCON0 – PWM控制0

寄存器	地址	复位值
PWMCON0	D7H，所有页，可位寻址	0000_0000b

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： 0 =PWM16 位计数器已完全清零。 1 =PWM16 位计数器尚未清零。

PWMPL –PWM周期低字节

寄存器	地址	复位值
PWMPL	D9H，所有页	0000_0000b

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

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

PWMnL – PWM通道 0~5 占空比低字节 n =0,1,2,3,4,5

寄存器	地址	复位值
PWM0L	DAH，所有页	0000 _0000 b
PWM1L	DBH，所有页	0000 _0000 b
PWM2L	DCH，所有页	0000 _0000 b
PWM3L	DDH，所有页	0000 _0000 b
PWM4L	CCH, 页1	0000 _0000 b
PWM5L	CCH, 页1	0000 _0000 b

7	6	5	4	3	2	1	0
PWMn [7:0], n =0,1,2,3,4,5							
读/写							

位	名称	描述
7:0	**PWMn[7:0]** **n=0,1,2,3,4,5**	**通道n占空比低字节** 此字节与PWMnH一起控制PWM发生器的输出信号PGn的占空比。

PIOCON0 – PWM或 I/O选择

寄存器	地址	复位值
PIOCON0	DEH，所有页	0000_0000b

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
5	PIO05	**P0.3/PWM0 通道5引脚功能选择** 0 = P0.3/PWM0_CH5引脚用作 P0.3. 1 = P0.3/PWM0_CH5引脚用作PWM0_CH5输出。
4	PIO04	**P0.1/PWM0 通道4引脚功能选择** 0 = P0.1/PWM0_CH4引脚用作 P0.1. 1 = P0.1/PWM0_CH4引脚用作PWM_CH4输出。
3	PIO03	**P0.0/PWM0 通道3引脚功能选择** 0 = P0.0/PWM0_CH3引脚用作 P0.0. 1 = P0.0/PWM0_CH3引脚用作PWM_CH3输出。
2	PIO02	**P1.0/PWM0 通道2引脚功能选择** 0 = P1.0/PWM0_CH2引脚用作 P1.0. 1 = P1.0/PWM0_CH2引脚用作PWM_CH2输出。
1	PIO01	**P1.1/PWM0 通道1引脚功能选择** 0 = P1.1/PWM0_CH1引脚用作 P1.1. 1 = P1.1/PWM0_CH1引脚用作PWM_CH1输出。
0	PIO00	**P1.2/PWM0 通道0引脚功能选择** 0 = P1.2/PWM0_CH0引脚用作 P1.2. 1 = P1.2/PWM0_CH0引脚用作PWM_CH0输出。

PWMCON1 – PWM控制1

寄存器	地址	复位值
PWMCON1	DFH，所有页	0000_0000b

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
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(FBD.6)选择匹配的边沿信号，PWM0~5输出FBD寄存器所设置的故障刹车数据， PWMRUN(PWMCON0.7)由硬件自动清除。当PWMRUN再次置位时，PWM输出恢复。
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	地址	复位值
ACC	E0H，所有页，可位寻址	0000_0000b

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	地址	复位值
ADCCON1	E1H，页0	0000_0000b

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

位	名称	描述
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]**	**ADC时钟分频** 00 = ADC 时钟源 = F_{SYS}/1 。 01 = ADC 时钟源 = F_{SYS}/2 。 10 = ADC 时钟源 = F_{SYS}/4 。 11 = ADC 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	地址	复位值
ADCCON2	E2H, 页0	0000_0000b

7	6	5	4	3	2	1	0
ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	ADCAQT[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。PWMRUN(PWMCON0.7) 也将由硬件自动清除。当PWMRUN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	**ADCAQT**	**ADC采样时间** 这3位字段决定ADC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。

ADCDLY – ADC外部触发延迟计数器

寄存器	地址	复位值
ADCDLY	E3H，所有页	0000_0000b

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= $\frac{ADCDLY}{F_{SYS}}$。 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 – 捕获通道n低字节

寄存器	地址	复位值
C0L	E4H，所有页	0000_0000b
C1L	E6H，所有页	0000_0000b
C2L	EDH，所有页	0000_0000b

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 –捕获通道n高字节

寄存器	地址	复位值
C0H	E4H，所有页	0000_0000b
C1H	E7H，所有页	0000_0000b
C2H	EEH，所有页	0000_0000b

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

寄存器	地址	复位值
ADCCON0	E8H，页0，可位寻址	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标志位** **当ADCMPEN(ADCCON2[5])禁用**，A/D 转换完成时，此标志位被置位；如果ADCCMPEN被使能，当比较器值与**CM1003的 ADC 比较值条件匹配时，此位置位**。可以读取ADC结果。当此标志为1时，ADC无法启动新的转换。此位由软件清除。
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.
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 = 内部带隙电压 其他= 保留.

PICON – 引脚中断控制

寄存器	地址	复位值
PICON	E9H，所有页	0000_0000b

7	6	5	4	3	2	1	0
PIT67	PIT45	PIT3	PIT2	PIT1	PIT0	PIPS[1:0]	
读/写	读/写	读/写	读/写	读/写	读/写	读/写	

位	名称	描述
7	**PIT67**	**引脚中断通道6和7类型选择** 此位选择引脚中断通道6和7触发哪种类型。 0 =电平触发。 1 =边沿触发。
6	**PIT45**	**引脚中断通道4和5类型选择** 此位选择引脚中断通道4和5触发的类型。 0 =电平触发。 1 =边沿触发。
5	**PIT3**	**引脚中断通道3类型选择** 此位选择引脚中断通道3触发的类型。 0 =电平触发。 1 =边沿触发。
4	**PIT2**	**引脚中断通道2类型选择** 此位选择引脚中断通道2触发的类型。 0 =电平触发。 1 =边沿触发。
3	**PIT1**	**引脚中断通道1类型选择** 此位选择引脚中断通道1触发的类型。 0 =电平触发。 1 =边沿触发。
2	**PIT0**	**引脚中断通道0类型选择** 此位选择引脚中断通道0触发的类型。 0 =电平触发。 1 =边沿触发。
1:0	**PIPS[:0]**	**引脚中断端口选择** 此位段选择哪个端口作为引脚中断所用的8位端口。 00 = 端口0。 01 = 端口1。 10 = 端口2。 11 = 端口3。

PINEN – 引脚中断负极性使能

寄存器	地址	复位值
PINEN	EAH，所有页	0000_0000b

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** 此位用于使能低电平/下降沿触发通道n的引脚中断。电平还是边沿，由PICON寄存器的PITn位决定。 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	地址	复位值
PIPEN	EBH，所有页	0000_0000b

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** 此位用于使能高电平/上升沿触发通道n的引脚中断。电平还是边沿，由PICON寄存器的PITn位决定。 0 = 高电平/上升沿检测禁用。 1 = 高电平/上升沿检测使能。

PIF – 引脚中断标志

寄存器	地址	复位值
PIF	ECH，所有页	0000_0000b

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，当侦测到通道n的边沿跳变触发引脚中断，此标志位由硬件置位，应由软件清除。 如果选择电平触发，此标志位根据引脚中断通道n上输入信号的逻辑电平的反转，软件无法控制此位。

EIP –扩展中断优先级

寄存器	地址	复位值
EIP	EFH，所有页	0000_0000b

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
注意:EIP和EIPH组合以确定每个中断源的优先级。关于正确的中断优先级配置，请参考表6.2-2 。		

B – B寄存器

寄存器	地址	复位值
B	F0H，所有页,可位寻址	0000_0000b

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，主要用于乘法和除法指令。

CAPCON3 – 输入捕获控制3

寄存器	地址	复位值
CAPCON3	F1H，所有页	0000_0000b

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	地址	复位值
CAPCON4	F2H，所有页	0000_0000b

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.

SPCR – 串行外设SPI控制寄存器

寄存器	地址	复位值
SPCR	F3H，页0	0000_0000b

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6.9-4 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6.9-4 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。 Fsys = 16MHz. SPR1 / SPR0 / 分频 / SPI 时钟速率 0 / 0 / 2 / 8M bit/s 0 / 1 / 4 / 4M bit/s 1 / 0 / 8 / 2M bit/s 1 / 1 / 16 / 1M bit/s Fsys = 24MHz. 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SPI控制寄存器2

寄存器	地址	复位值
SPCR2	F3H，页1	0000_0000b

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 相邻字节间的间隔时间选择** 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SPI状态寄存器

寄存器	地址	复位值
SPSR	F4H，所有页	0000_0000b

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。 如果使能ESPI和 EA，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** 此位指示写入冲突事件。一旦发生写入冲突事件，此位被置位，应由软件清零。
5	SPIOVF	**SPI 溢出标志** 此位指示溢出事件。一旦发生溢出，此位被置位。如果使能ESPI 和 EA， 产生SPI 中断请求。此位应由软件清零。
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6.9-1 选择所示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	地址	复位值
SPDR	F5H，所有页	0000_0000b

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。

AINDIDS – ADC通道数字输入功能关闭

寄存器	地址	复位值
AINDIDS	F6H，所有页	0000_0000b

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。

EIPH – 扩展中断优先级高位

寄存器	地址	复位值
EIPH	F7H，所有页	0000_0000b

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**	故障刹车中断优先级高位
4	**PWDTH**	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和EIP组合以确定每个中断源的优先级。关于正确的中断优先级配置，请参考表6.2-2 。		

SCON_1 – 串口1控制

寄存器	地址	复位值
SCON_1	F8H，所有页， 可位寻址	0000_0000b

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6.8-2 串口UART1模式/波特率描述。 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。

PDTEN – PWM死区时间使能

寄存器	地址	复位值
PDTEN	F9H，所有页，TA保护	0000_0000b

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

位	名称	描述
4	PDTCNT.8	**PWM死区计数器第8位** 参见PDTCNT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 = GP4/GP5对信号无延时。 1 = 在GP4/GP5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 = GP2/GP3对信号无延时。 1 =在GP2/GP3对信号的上升沿插入死区时间延时。
0	PDT01EN	**PWM0_CH0/PWM0_CH1组死区时间插入使能** 此位仅当PWM0_CH0/PWM0_CH1为互补模式时有效。 0 = GP0/GP1对信号无延时。 1 = 在GP0/GP1对信号的上升沿插入死区时间延时。

PDTCNT – PWM死区时间计数器

寄存器	地址	复位值
PDTCNT	FAH，所有页， TA保护	0000_0000b

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

位	名称	描述
7:0	**PDTCNT[7:0]**	**PWM死区计数器低字节** 此8位字段与PDTEN.4组合构成9位PWM死区计数器PDTCNT。仅当PWM处于互补模式且PWM对应的PDTEN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PDTCNT。

PMEN – PWM掩码使能

寄存器	地址	复位值
PMEN	FBH，所有页	0000_0000b

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
n	**PMENn**	**PWMn掩码使能** 0 = PWMn信号来自其PWM发生器输出。 1 = PWMn根据PMDn设定的电位值掩码。

PMD –PWM掩码数据

寄存器	地址	复位值
PMD	FCH，所有页	0000_0000b

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
n	PMDn	**PWMn掩码数据** 一旦相应的PMENn被设置，PWMn 信号输出掩码电准位值 0 = PWMn信号被0掩码。 1 = PWMn信号被1掩码。

PORDIS – 上电复位POR禁用

寄存器	地址	复位值
PORDIS	FDH，所有页， TA保护	0000_0000b

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	地址	复位值
EIP1	FEH，页0	0000_0000b

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

位	名称	描述
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和EIPH1组合以确定每个中断源的优先级。关于正确的中断优先级配置，请参考表6.2-2 。		

EIPH1 – 扩展中断优先级高位 1

寄存器	地址	复位值
EIPH1	FFH，页0	0000_0000b

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

位	名称	描述
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和EIP1组合以确定每个中断源的优先级。关于正确的中断优先级配置，请参考表6.2-2 。		

6.2 系统管理

6.2.1 时钟系统

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。共有三种系统时钟源可选择，包括内部振荡器，或通过软件设置来自XIN引脚的外部时钟。CM1003内嵌两个内部振荡器：一个10 kHz低速振荡器和一个16 MHz高速振荡器，误差在出厂时校准到±2%(全温度全电压范围内)。时钟除频器 CKDIV可灵活调整功耗和操作性能。

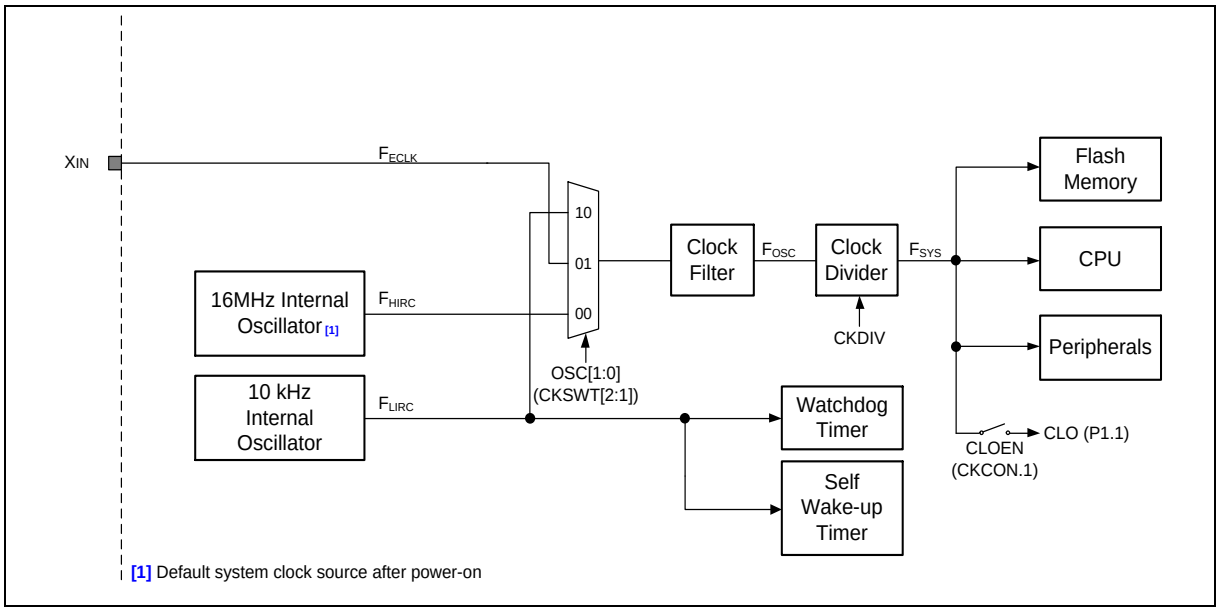


图 6.2-1 时钟系统框图

6.2.1.1 *系统时钟源*

CM1003共有3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P3.0/XIN的多功能。

6.2.1.2 *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10 kHz低速振荡器(LIRC)。它们都可以被选用成系统时钟。通过设置HIRCEN(CKEN.5)来使能 HIRC。设备上电后LIRC被使能。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，XIN自动作为普通I/O P3.0，用来扩展普通I/O数量。可以通过配置P3M1和P3M2寄存器来选择P3.0的输出模式。

由于HIRC的值有9位数据，因此用户可以通过调整SFR来更改HIRC的值，每更改一位的数值，大约调整HIRC频率40kHz。减小此值将使HIRC速度更快，增加1意味着HIRC速度变慢约40kHz。

下图显示两个字节组合成9 位内部 RC 调整值，以及HIRC 24MHz 定义位。

RCTRIM0 –高速内部振荡器 16 MHz Trim0(TA保护)

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

地址: 84H　复位值: 16 MHz HIRC 值

RCTRIM1 – 高速内部振荡器16MHz Trim1(TA保护)

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

地址: 85H　复位值: 16 MHz HIRC值

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。

6.2.1.3 *系统时钟切换*

CM1003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是定时访问写保护（TA）的。在时钟源控制下，时钟源可以在外部时钟源和内部振荡器之间切换，甚至可以在高速和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期来使切换时钟源稳定工作。因此，用户应遵循以下步骤以确保完成时钟源切换。用户可以先通过配置CKEN寄存器打开目标时钟源，再通过轮询CKSWT寄存器中的状态位等待时钟源稳定，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。如果用户关心功耗的话，可以关闭原时钟源。请注意，如果不遵循上述步骤，硬件将采取一些措施来处理此类违规操作，如下所示。

1. 如果用户试图通过改变CKEN值来关闭当前时钟源，设备将忽略此操作。系统时钟将保持现状，CKEN值不发生改变。

2. 如果用户试图通过改变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. 如果用户将系统时钟源切换到一个已启用但还不稳定的时钟源，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将设置为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 – 时钟切换(TA保护)

寄存器	地址	复位值
CKSWT	96H，所有页, TA保护	0011_0000b

7	6	5	4	3	2	1	0
-	-	HIRCST	-	ECLKST	OSC[1:0]		-
-	-	只读	-	只读	只写		-

位	名称	描述
7:6	-	保留
5	HIRCST	**高速内部振荡器24MHz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
4	-	保留
3	ECLKST	**外部时钟输入状态** 0 = 外部时钟输入不稳定或禁用 1 = 外部时钟输入使能且稳定

位	名称	描述
2:1	**OSC[1:0]**	**振荡器选择位** 此位段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	地址	复位值
CKEN	97H，所有页, TA保护	0011_0000b

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

位	名称	描述
7:6	**EXTEN[1:0]**	**外部时钟源使能** 11 = 由X_{IN}输入的外部时钟源使能 其它 = 外部时钟源禁用 。P30作为普通I/O使用。
5	**HIRCEN**	**高速内部振荡器16MHz使能** 0 = 高速内部振荡器禁止。 1 = 高速内部振荡器使能。 注意：一旦通过设置IAPEN(CHPCON.0)使能 IAP，高速内部16MHz 振荡器将自动使能，硬件也会设置HIRCEN和HIRCST位。清除IAPEN 后，HIRCEN和EHRCST恢复原值。
4:1	**-**	保留
0	**CKSWTF**	**时钟切换故障标志位** 0 = 上一次系统时钟源切换成功 1 = 用户试图将上一次的系统时钟源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。

6.2.1.4 *系统时钟除频*

通过配置除频寄存器 CKDIV，振荡器频率(F_{OSC})可以整数倍降频(最高可达 1/510)，再提供给系统时钟(F_{SYS})。此功能可临时以较低的速率运行MCU来降低功耗。通过时钟除频，MCU可以以低速执行正常程序，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 – 时钟除频器

寄存器	地址	复位值
CKDIV	95H，所有页	0000_0000 b

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

位	名称	描述
7:0	CKDIV[7:0]	**时钟除频器** 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, $F_{SYS} = \frac{F_{OSC}}{2 \quad CKDIV}$

6.2.1.5 *系统时钟输出*

CM1003提供一个CLO(P1.1)引脚用于输出系统时钟，其频率与F_{SYS}相同。输出使能位为CLOEN(CKCON.1)。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	地址	复位值
CKCON	8EH，所有页	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

6.2.2 电源管理

CM1003有多种功能，可帮助用户控制设备的功耗。省电功能有两种可选模式：空闲模式和掉电模式。为了实现稳定的功耗，应注意每个引脚的状态和模式。将引脚设为与外部上、下拉一致来可以实现最小功耗。比如，使用上拉则输出1；使用下拉则输出0。如果I/O引脚悬空，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，则应具有外部上拉或下拉，或者通过设置P20UP(P2S.7)启用其内部上拉。

6.2.2.1 *空闲模式*

空闲模式通过保持程序计数器来使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。通常空闲模式下的功耗约为全工作状态下的一半。

由于空闲模式下，提供给外设的逻辑电路，比如定时器或串口依然工作，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。

6.2.2.2 *掉电模式*

掉电模式是CM1003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。

PCON – 电源控制

寄存器	地址	复位值
PCON	87H，所有页	上电复位, 0001_0000b 其他，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
1	**PD**	**掉电模式** 设置此位将使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)暂停。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的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)暂停，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的中断源的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。

6.2.3 重置系统

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，上电检测和欠压检测。

CM1003的复位条件有几种类型。通过软件标志可以确定复位源。通常，大部分特殊功能寄存器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回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。

系统复位可以由下列事件之一引起。从寄存器中读取复位事件标志位来确定复位源。硬件复位源来自外设信号。软件复位可以通过设置控制寄存器来触发复位。

- 硬件复位源
  - 上电复位(POR)

- nRESET引脚的低电平
- 看门狗超时复位(WDT复位)
- 低电压复位(LVR)
- 掉电检测器复位(BOD 复位)

- 软件复位源
  - 芯片复位将通过向SWRST(CHPCON)写入1来重置整个芯片

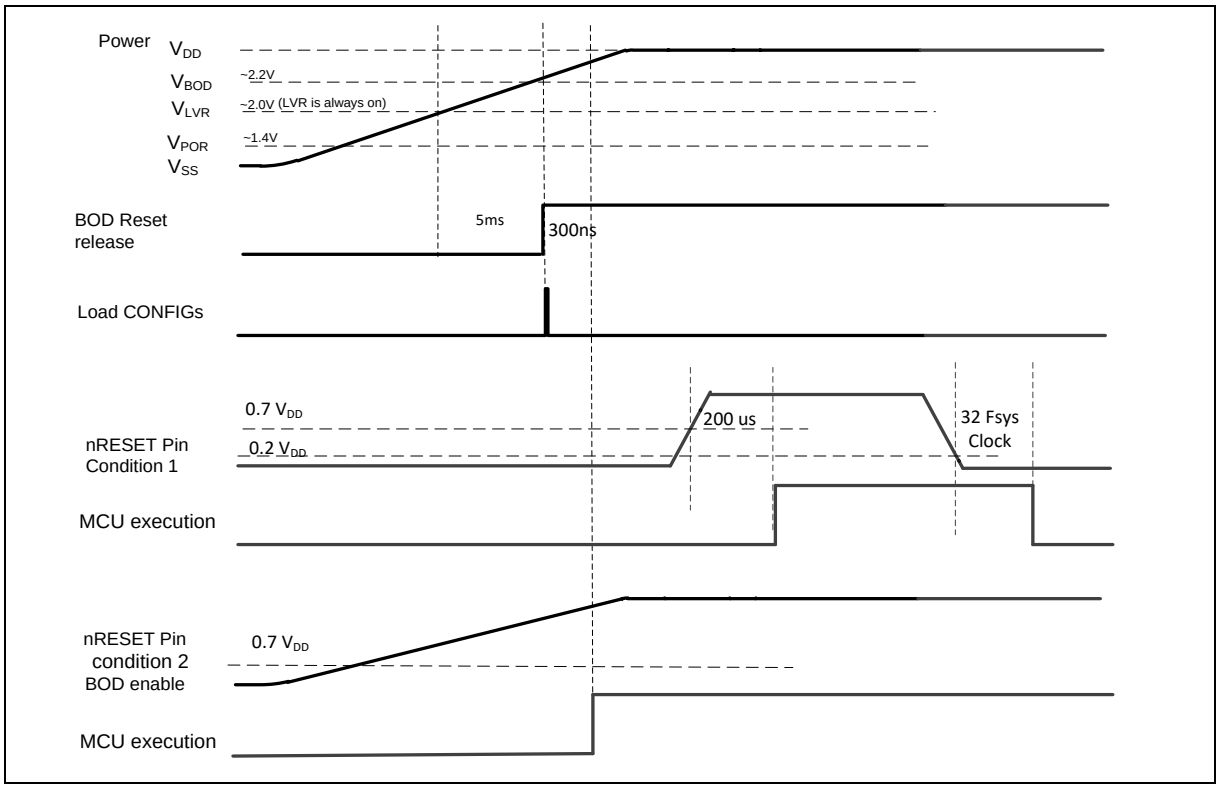


图 6.2-2 上电和复位时序

6.2.3.1 上电复位和低压复位

CM1003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，不访问程序存储空间。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。如果是电源关闭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	地址	复位值
PCON	87H，所有页	上电复位, 0001_0000b 其他，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，此位被置1，用以标示当前冷复位，上电复位完成。其它任何复位后，此位值保持不变，建议通过软件清0。

6.2.3.2 欠压检测和复位

欠压检测电路用于监测V_{DD}执行期间的电平。当V_{DD}降至选定的掉电触发电平(V_{BOD})时，如果BORST(BODCON0.2)设置为1，掉电检测逻辑将复位MCU。掉电复位后，BORF(BODCON0.1)将由硬件置位1。除上电复位或掉电复位本身外，任何其他复位都不会改变BORF。此位可以由软件设置或清除。

BODCON0 – 欠压检测控制0

寄存器	地址	复位值
BODCON0	A3H，所有页，TA保护	上电复位,CCCC_XC0Xb 掉电复位, UUUU_XU1Xb 其他，UUUU_XUUXb

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

位	名称	描述
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。

6.2.3.3 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。然后复位电路同步请求内部复位信号。因此，复位是同步操作，需要时钟在此期间运行才能促使外部复位。

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(AUXR1.6)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(AUXR1.5)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，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	地址	复位值
AUXR1	A2H，所有页	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**	**HardFInt**	**GF2**	**-**	**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 通过外部复位引发复位时，此位由硬件置位。建议由软件清0。
5	**HardF**	**硬件故障复位标志** 当 EHFI (EIE1.4) = 0 时，一旦 CPU 读取超过 Flash 大小的指令地址，MCU 将复位，并通过硬件置位此位。建议通过软件清除该标志。 **注意：**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。仅 HardF 标志有效。

6.2.3.4 *看门狗复位*

看门狗定时器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	地址	复位值
WDCON	AAH，所有页, TA保护	上电复位 0000_0111b 看门狗 0000_1UUUb 其他 0000_UUUUb

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。建议复位后由软件清零。

6.2.3.5 *软件复位*

CM1003提供软件复位，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。

如果发生软件复位，SWRF(AUXR1.7)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	地址	复位值
CHPCON	9FH，所有页，TA保护	软件复位： 0000_00U0b 其他：0000_00C0b

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	地址	复位值
AUXR1	A2H，所有页	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。建议由软件清零

6.2.3.6 *启动选择*

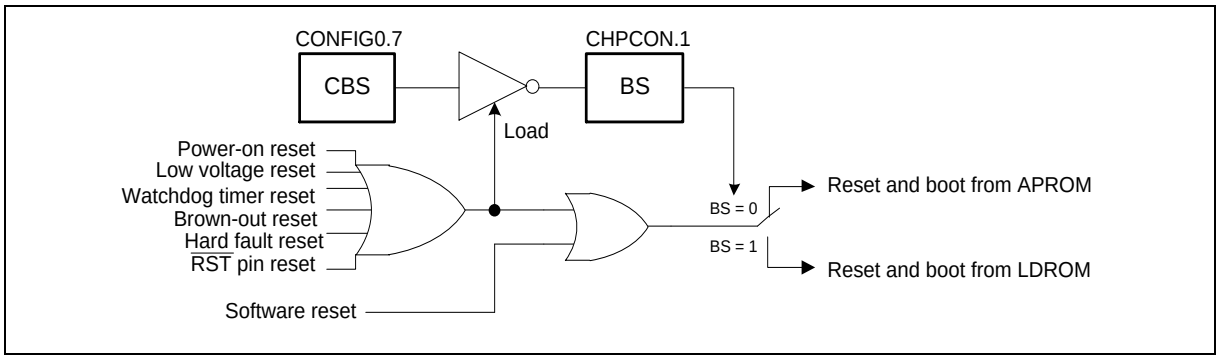


图 6.2-3 启动选择结构图

CM1003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位决定在复位后的MCU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加载CBS位(CONFIG0.7)的相反值。

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

CONFIG0

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

出厂预设值: 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从哪个模块重新启动。 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	地址	复位值
CHPCON	9FH，所有页，TA保护	软件复位： 0000_00U0b 其他：0000_00C0b

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** 此位定义所有复位后，MCU从哪个块重新启动 0 = 所有复位后，MCU从APROM重新启动 1 = 所有复位后，MCU从LDROM重新启动
注意: 除软件复位外，所有复位后 BS 加载 CONFIG0.7的CBS位 的相反值来初始化。软件复位后保持不变。		

6.2.3.7 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

中断的目的是使软件处理非计划或异步事件。CM1003具有四个中断优先级，30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 被分配到预先指定的地址，如表6.2-1 中断。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	-	SPI中断	004BH	9
外部中断0	0003H	0	看门狗中断	0053H	10
定时器0溢出	000BH	1	ADC中断	005BH	11
外部中断1	0013H	2	输入捕获中断	0063H	12
定时器1溢出	001BH	3	PWM中断	006BH	13
串口0中断	0023H	4	故障刹车中断	0073H	14
定时器2事件	002BH	5	串口1 中断	007BH	15
I^2C状态/超时中断	0033H	6	定时器3溢出	0083H	16
引脚中断	003BH	7	自唤醒定时器中断	008BH	17
掉电检测中断	0043H	8			

表 6.2-1 中断向量

6.2.4.1 中断使能

通过 IE 和 EIE0 特殊功能寄存器中的相应中断使能位，可以使能或禁用每个单独的中断源。还有一个全局中断使能 EA(IE.7)位，将其清0可一次性禁用所有中断。它被置位将使能所有单独使能的中断。将 EA 位设置为逻辑0会禁用所有中断源，而无论其单独的中断使能设置如何。请注意，当 EA 位设置为逻辑0时发生的中断将被挂起，直到 EA 位恢复为逻辑1时才会得到服务。所有中断标志位可以通过软件设置。从而可以生成软件启动的中断。

请注意，使能的每个中断都是通过将它的中断标志位设置为逻辑1来生成的，无论是通过硬件还是软件。用户应在其自己的中断服务程序(ISR)中处理每个中断标志位。大多数中断标志位应通过软件将其写入逻辑0来清除，以避免中断请求递归。

6.2.4.2 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6.2-2列举了四种优先级配置。当然，低优先级中断本身可以被高优先级中断打断，但不能被任何同级中断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。

如果有多个中断，遵循以下规则：

1. 当低优先级中断处理程序正在运行时，这时一个高优先级中断产生，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。当低优先级中断处理程序执行“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。

自然优先级定义如表6.2-2 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，请参阅第6.2.2.2“掉电模式”。

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 6.2-2 中断优先级层级设置

中断源	向量地址	中断标志位	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(FBD.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	EPWM	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

注：
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 6.2-3 各中断源特性表

6.2.4.3 中断服务

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。

6.2.4.4 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，用户可以轻松地向IAPAH和IAPAL中填入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**2.3ms**，字节编程时间为**27μs**。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
SPROM 页擦除	10	1	0	0010	0180H	FFH
SPROM 字节编程	10	1	0	0001	0180H~01FFH	数据写入
SPROM 字节读	10	0	0	0000	0180H~01FFH	数据读出
擦除所有CONFIG	11	1	0	0010	0000H	FFH
CONFIG 字节编程	11	1	0	0001	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H	数据写入
CONFIG 字节读	11	0	0	0000	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H	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 *控制寄存器*

下面的寄存器与IAP操作相关。

CONFIG2

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

出厂预设值: 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	地址	复位值
CHPCON	9FH，所有页，TA保护	软件复位： 0000_00U0b 其他：0000_00C0b

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	地址	复位值
IAPUEN	A5H，所有页，TA保护	0000_0000b

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擦除或者编程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	地址	复位值
IAPCN	AFH，所有页	0011_0000b

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]	

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

IAPAH – IAP 地址高字节

寄存器	地址	复位值
IAPAH	A7H，所有页	0000_0000b

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** IAPAHL包含用于IAP操作的地址 IAPA[15:8] 。

IAPAL – IAP 地址低字节

寄存器	地址	复位值
IAPAL	A6H，所有页	0000_0000b

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	地址	复位值
IAPFD	AEH，所有页	0000_0000b

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。用户应在触发 IAP 处理之前将数据写入 IAPFD 进行编程模式，并在 IAP 处理完成后从 IAPFD 读取数据进行读取/ 验证模式。

IAPTRG – IAP触发

寄存器	地址	复位值
IAPTRG	A4H，所有页，TA保护	0000_0000b

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。
		程式流程如下：
		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存储作为数据存储

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)

CM1003内嵌片上调试功能(OCD)，为开发人员提供了低成本的代码调试方法，每种封装都适用。OCD提供完整的程序流程控制调试功能，包括八个硬件断点、单步运行、全速运行和非侵入式存储访问命令。OCD系统并不占用任何本地存储，也不共享片上外设。

当 OCDEN(CONFIG0.4)被设置为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。nRESET引脚也是OCD模式进入和退出所必需的。CM1003通过ICP写入模式支持具有Flash存储控制路径的OCD，该模式与OCD 共用这三个引脚。

CM1003使用ICE_DAT、ICE_CLK 和nRESET引脚与OCD系统交互。在设计系统用到OCD时，必须考虑以下限制以确保正确运行：

1. nRESET不能直接连接到V_{DD}，并且必须移除连接的所有外部电容。

2.必须断开所有外部复位源。

3.连接在ICE_DAT和ICE_CLK上的任何外围器件都必须隔离。

6.3.3.1 OCD限制条件

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. nRESET引脚用于OCD模式选择。

2. ICE_DAT引脚物理上与P5.0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P5.1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**	**-**	**-**	**LOCK**	**-**
读/写	-	读/写	读/写	-	-	读/写	-

出厂预设值: 1111 1111b

位	名称	描述
5	**OCDPWM**	**OCD调试停止模式时，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,硬件故障复位被禁用,仅仅HFIF 标志位有效。

6.3.4 96位唯一标识符(UCID)

出厂前，每颗CM1003芯片都经过工厂预烧一个96位宽度的序列号，该序列号保证唯一。此序列号被称为唯一标识符UID(Unique Code)。用户只能通过IAP指令读取该唯一标识符，详见章节 6.3.1.1 IAP

IAP 模式	IAPCN				IAPA[15:0] {IAPAH, IAPAL}	IAPFD[7:0]
	IAPB [1:0]	FOEN	FCEN	FCTRL [3:0]		
96位标识符读取	XX	0	0	0100	0000H至000BH	数据输出

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。

6.4 通用端口(GPIO)

6.4.1 GPIO 模式

CM1003系列最多有43个通用I/O引脚，其中40个可位寻址的通用I/O引脚，分为5个端口P0 至P4，以及有7个通用I/O引脚的P5。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
注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
```

6.4.1.1 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。

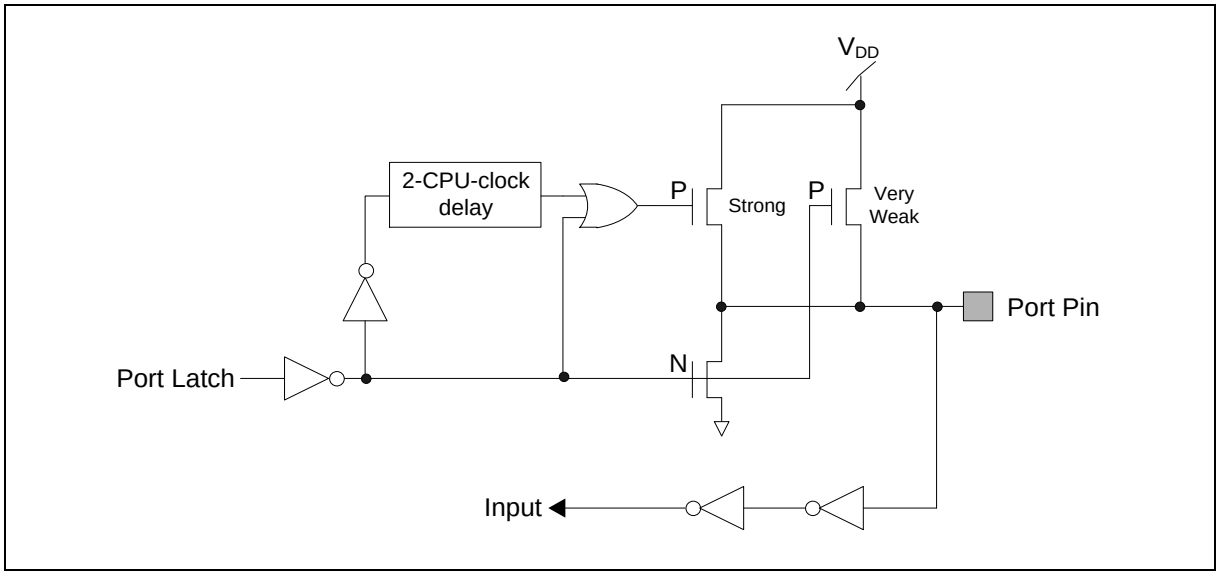


图 6.4-1 准双向模式结构

6.4.1.2 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。

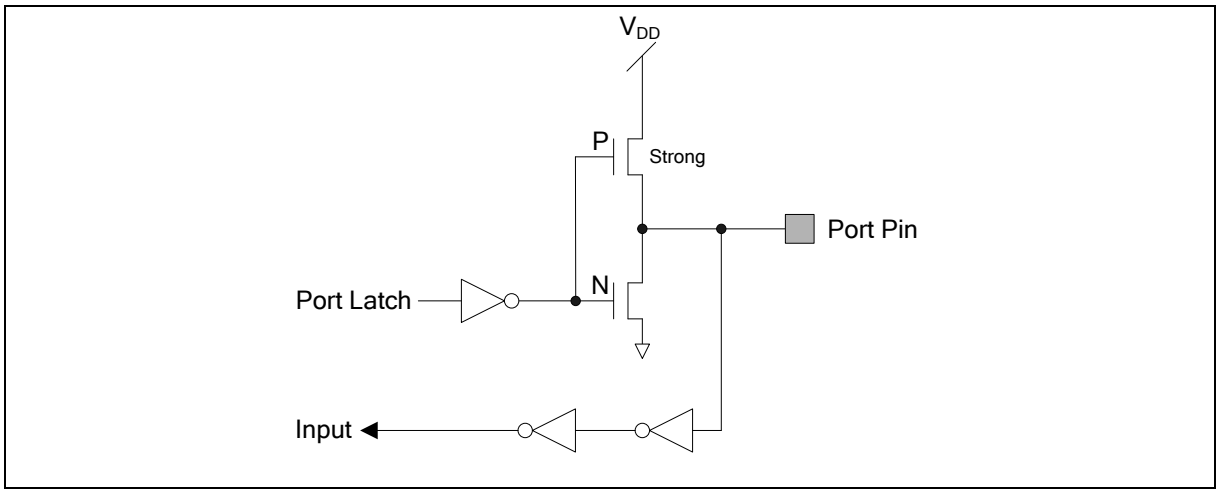


图 6.4-2 推挽模式结构

6.4.1.3 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。

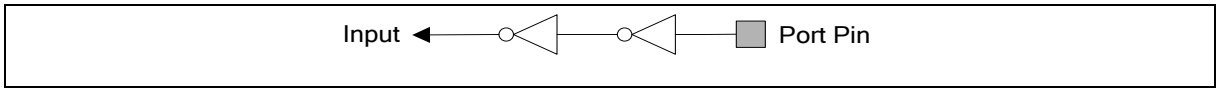


图 6.4-3 仅输入模式结构

6.4.1.4 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。

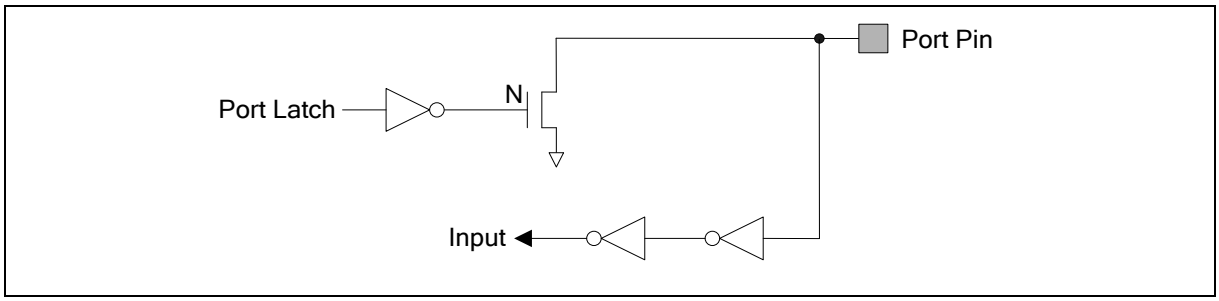


图 6.4-4 开漏模式结构

6.4.2 寄存器描述

CM1003具有大量的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可分为四组：输入和输出控制、输出模式控制、输入类型和灌电流控制，以及输出电压转换速率控制。所有特殊功能寄存器如下所列。

6.4.2.1 输入输出数据控制

这些寄存器是I/O输入输出数据缓冲区，读获取I/O输入的数据，写驱动数据输出。所有这些寄存器都可位寻址。

Pn – 端口n(可位寻址)

寄存器	地址	复位值
P0	80H，所有页, 可位寻址	1111_1111 b
P1	90H，所有页, 可位寻址	1111_1111 b
P2	A0H，所有页, 可位寻址	0000_0001 b
P3	B0H，所有页, 可位寻址	0000_0001 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

P0 / P1

位	名称	描述
7:0	**P0[7:0]** **P1[7:0]**	**端口 0/1** 端口 0/1是最大8 位通用 I/O 端口。

P2

位	名称	描述
0	**P2.0**	**端口 2 第0位** 当 RPD(CONFIG0.2)被编程为0时，P2.0 是仅输入引脚。当保持 RPD 未编程时，P2.0 始终读为 0。

P3

位	名称	描述
0	**P3.0**	**端口 3 第0位** P3.0仅当使用内部振荡器作为系统时钟时才可用，此时 P3.0 作为通用 I/O 口。 如果系统时钟没有选择内部振荡器，P3.0 引脚作为 OSCIN。对 P3.0 的写入无效，并且 P3.0 始终读为 0。

6.4.2.2 GPIO模式控制

这些寄存器控制GPIO模式，可配置为四种模式：仅输入、准双向、推挽或开漏。每个引脚都可以单独配置。

重置后所有 GPIO 恢复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	推拉
1	0	仅输入(高阻抗)
1	1	漏极开路

PnM1 – 端口n模式选择 1

寄存器	SFR 地址	复位值
P0M1	B1H, 页1	1111_1111 b
P1M1	B3H, 页1	1111_1111 b
P3M1	C2H, 页1	0000_0001 b

PnM2 – 端口模式选择 2

寄存器	SFR 地址	复位值
P0M2	B2H, 页1	0000_0000 b
P1M2	B4H, 页1	0000_0000 b
P3M2	C3H, 页1	0000_0000 b

PnM1 – 端口n模式选择 1

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

PnM2 – 端口n模式选择2

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	**PnMn[7:0]**	端口0模式选择

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，将 SFR切换到第1页，才可访问所有PxS寄存器。

PnS – 端口n施密特触发输入

寄存器	SFR 地址	复位值
P0S	B1H, 页1	0000_0000 b
P1S	83H, 页1	0000_0000 b
P3S	ACH, 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]**	**P0.n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。

P2S – P20 设置和定时器0/1的输出使能

寄存器	地址	复位值
P2S	B5H，所有页	0000_0000 b

7	6	5	4	3	2	1	0
P20UP	**-**	**-**	**-**	**T1OE**	**T0OE**	**-**	**P2S.0**
读/写	-	-	-	读/写	读/写	-	读/写

位	名称	描述
7	**P20UP**	**P2.0上拉使能位** 0 = P2.0 上拉禁用 1 = P2.0 上拉使能 仅当RPD(CONFIG0.2)为0时，此位有效。如果作为$\overline{\text{nRESET}}$引脚，上拉始终启用。
3	**T1OE**	**定时器1输出使能** 0 = 定时器1输出禁用。 1 = 从T1引脚启用定时器1输出。 请注意，定时器1输出仅在“定时器”模式下运行时才有效。
2	**T0OE**	**定时器0输出使能** 0 = 定时器0输出禁用。 1 = 从T0引脚启用定时器0 输出。 请注意，定时器0输出仅在“定时器”模式下运行时才有效。
0	**P2S.0**	**P2.0施密特触发输入** 0 = P2.0 TTL电平输入 1 = P2.0 施密特触发输入

6.4.2.4 *输出电压转换速率(输出斜率)控制*

每个I/O引脚的电压转换速率(输出斜率)均可单独配置。默认情况下，每个引脚处于正常斜率模式。用户可以设置每个控制寄存器位，以启用相应I/O引脚的高速斜率。请注意，将SFR切换到第1页，才可访问所有PxS寄存器。

PnSR – 端口n电压转换速率

寄存器	SFR 地址	复位值
P0SR	82H, 页1	0000_0000 b
P1SR	84H, 页1	0000_0000 b
P3SR	ADH, 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]	**P0.n电压转换速率** 0 = P0.n正常的输出电压转换速率 1 = P0.n高速的输出电压转换速率

6.4.3 读取-修改-写指令

从SFR或内部RAM读取一个字节、修改它并重新写回的指令称为“读取-修改-写入”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值、修改它并重新写回端口SFR。所有“读取-修改-写入”指令如下所列。

操作说明	描述
ANL	逻辑与。(ANL 直接地址、A 和 ANL 直接地址, #数据)
ORL	逻辑或。(ORL 直接地址，A 和 ORL 直接地址, #数据)
XRL	逻辑异或。(XRL 直接地址，A 和 XRL 直接地址, #数据)
JBC	如果位 =1则跳转并清除它。(JBC 位，rel)
CPL	补码位。(CPL 位)
INC	增量。(INC 直接地址)
DEC	减少。(DEC 直接地址)
DJNZ	如果不为零则减少并跳转。(DJNZ 直接地址，rel)
MOV	位，移动C进位标志位到位。(MOV 位，C)
CLR	位 清除。(CLR 位)
SETB	位 置位。(SETB 位)

最后三个指令看起来不像是“读-修改-写”指令，但实际上它们是。它们读取整个端口锁存器的值，修改要改变的位，然后将新值写回到端口锁存器。

6.4.4 外部中断引脚

外部中断$\overline{INT0}$和$\overline{INT1}$可用作中断源。它们可以根据IT0(TCON.0)和IT1(TCON.2)选择边沿触发或电平触发中断。检测IE0(TCON.1)和IE1(TCON.3)标志位判断是否产生中断。在边沿触发模式下，每个系统时钟周期都会采样$\overline{INT0}$或$\overline{INT1}$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，并且置位中断请求标志IE0或IE1。由于每个系统时钟周期都去采样外部中断，所以高电平或低电平至少保持一个系统时钟周期。当中断服务程序被执行时，IE0和IE1会被硬件自动清除。如果选择电平触发模式，那么必须保持引脚为低电平，直到进入中断服务，在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标志位与$\overline{INT0}$和$\overline{INT1}$引脚逻辑电平值相反。当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。$\overline{INT0}$和$\overline{INT1}$均支持将芯片从掉电模式唤醒。

IE – 中断使能

寄存器	地址	复位值
IE	A8H，所有页,可位寻址	0000_0000b

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	**掉电中断使能** 0 = 掉电中断禁用。 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。

位	名称	描述
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。

EIE – 扩展中断使能

寄存器	地址	复位值
EIE	9BH，所有页	0000_0000b

7	6	5	4	3	2	1	0
ET2	ESPI	EFB	EWDT	EPWM	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	**EPWM**	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**	I^2C**中断使能** 0 = I^2C 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	地址	复位值
EIE1	9CH，所有页	0000_0000b

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

位	名称	描述
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。

IP – 中断优先级

寄存器	地址	复位值
IP	B8H，所有页，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6.2-2 。		

IPH – 中断优先级高位

寄存器	地址	复位值
IPH	B7H, 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6.2-2 。		

EIP – 扩展中断优先级

寄存器	地址	复位值
EIP	EFH，所有页	0000_0000b

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
注意: EIP和EIPH组合以确定每个中断源的优先级。关于正确的中断优先级配置，请参考表6.2-2 。		

EIPH – 扩展中断优先级高位

寄存器	地址	复位值
EIPH	F7H，所有页	0000_0000b

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**	故障刹车中断优先级高位
4	**PWDTH**	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和EIP组合以确定每个中断源的优先级。关于正确的中断优先级配置，请参考表6.2-2 。		

EIP1 – 扩展中断优先级1

寄存器	地址	复位值
EIP1	FEH，页0	0000_0000b

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

位	名称	描述
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和EIPH1组合以确定每个中断源的优先级。关于正确的中断优先级配置，请参考表6.2-2 。		

EIPH1 – 扩展中断优先级高位 1

寄存器	地址	复位值
EIPH1	FFH，页0	0000_0000b

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

位	名称	描述
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和EIP1组合以确定每个中断源的优先级。关于正确的中断优先级配置，请参考表6.2-2 。		

TCON – 定时器0和1控制

寄存器	地址	复位值
TCON	88H，所有页, 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	只读(电平) 读/写(边沿)	读/写	只读(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$\overline{\text{INT1}}$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

6.4.5 引脚中断

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。当键盘或小键盘上的任何键被按下时，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。

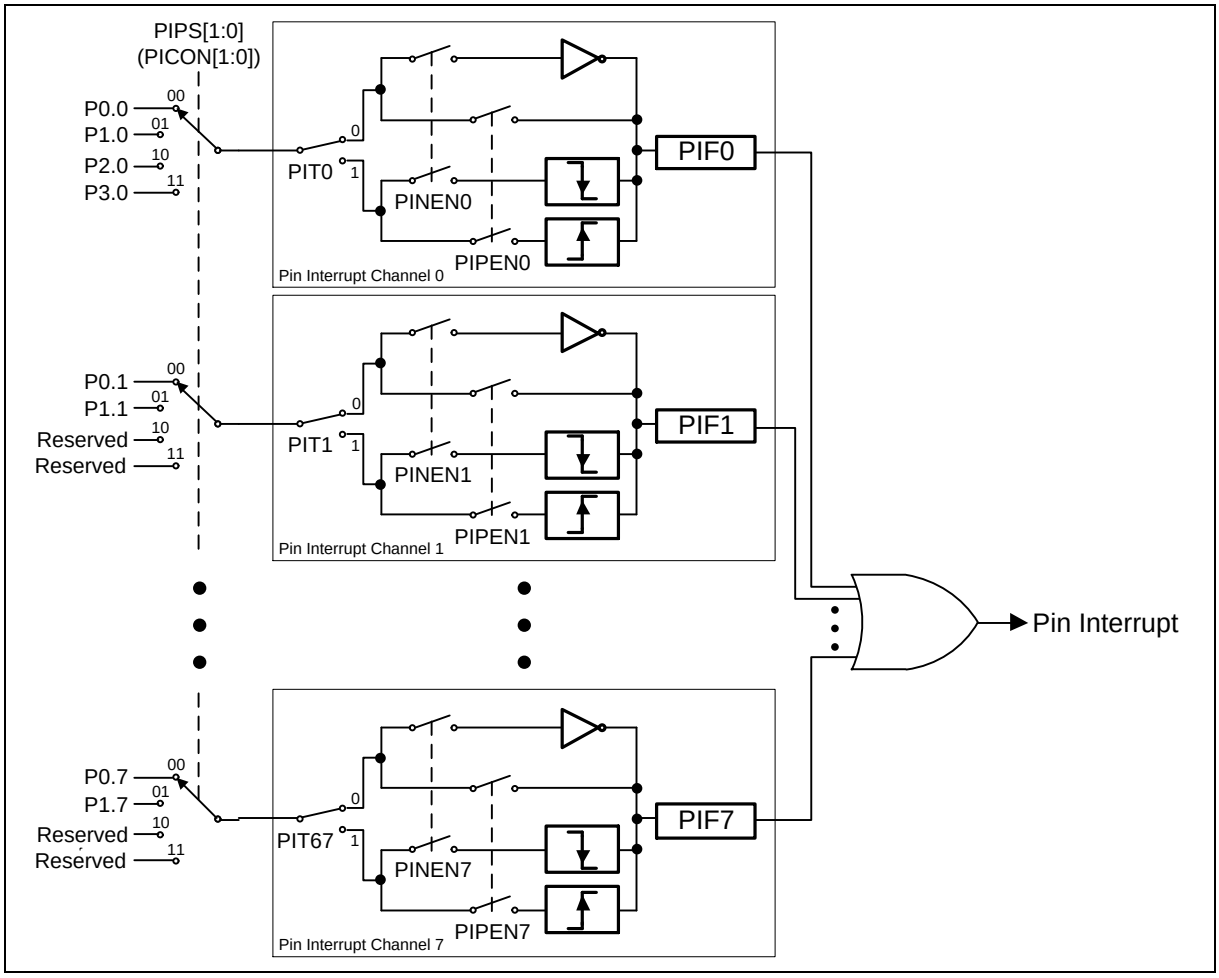


图 6.4-5 引脚接口框图

引脚中断通常用于检测外围设备(如键盘或小键盘)的边沿变化。在空闲状态，系统进入掉电模式下，降到最低功耗等待事件发生。引脚中断可以将设备从掉电模式唤醒。

PICON – 引脚中断控制

寄存器	地址	复位值
PICON	E9H，所有页	0000_0000b

7	6	5	4	3	2	1	0
PIT67	PIT45	PIT3	PIT2	PIT1	PIT0	PIPS[1:0]	
读/写	读/写	读/写	读/写	读/写	读/写	读/写	

位	名称	描述
7	**PIT67**	**引脚中断通道6和7类型选择** 此位选择引脚中断通道6和7触发哪种类型。 0 =电平触发。 1 =边沿触发。
6	**PIT45**	**引脚中断通道4和5类型选择** 此位选择引脚中断通道4和5触发的类型。 0 =电平触发。 1 =边沿触发。
5	**PIT3**	**引脚中断通道3类型选择** 此位选择引脚中断通道3触发的类型。 0 =电平触发。 1 =边沿触发。
4	**PIT2**	**引脚中断通道 2类型选择** 此位选择引脚中断通道2触发的类型。 0 =电平触发。 1 =边沿触发。
3	**PIT1**	**引脚中断通道1类型选择** 此位选择引脚中断通道1触发的类型。 0 =电平触发。 1 =边沿触发。
2	**PIT0**	**引脚中断通道0类型选择** 此位选择引脚中断通道0触发的类型。 0 =电平触发。 1 =边沿触发。
1:0	**PIPS[:0]**	**引脚中断端口选择** 此位段选择哪个端口作为引脚中断所用的8位端口。 00 = 端口0。 01 = 端口1。 10 = 端口2。 11 = 端口3。

PINEN – 引脚中断负极性使能

寄存器	地址	复位值
PINEN	EAH，所有页	0000_0000b

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** 此位用于使能低电平/下降沿触发通道n的引脚中断。电平还是边沿，由PICON寄存器的PITn位决定。 0 = 低电平/下降沿检测禁用。 1 = 低电平/下降沿检测使能。

PIPEN – 引脚中断正极性使能

寄存器	地址	复位值
PIPEN	EBH，所有页	0000_0000b

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** 此位用于使能高电平/上升沿触发通道n的引脚中断。电平还是边沿，由PICON寄存器的PITn位决定。 0 = 高电平/上升沿检测禁用。 1 = 高电平/上升沿检测使能。

PIF – 引脚中断标志

寄存器	地址	复位值
PIF	ECH，所有页	0000_0000b

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，当侦测到通道n的边沿跳变触发引脚中断，此标志位由硬件置位，应由软件清除。 如果选择电平触发，此标志位根据引脚中断通道n上输入信号的逻辑电平的反转，软件无法控制此位。

6.5 定时器

6.5.1 定时器/计数器0和1

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。

6.5.1.1 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。

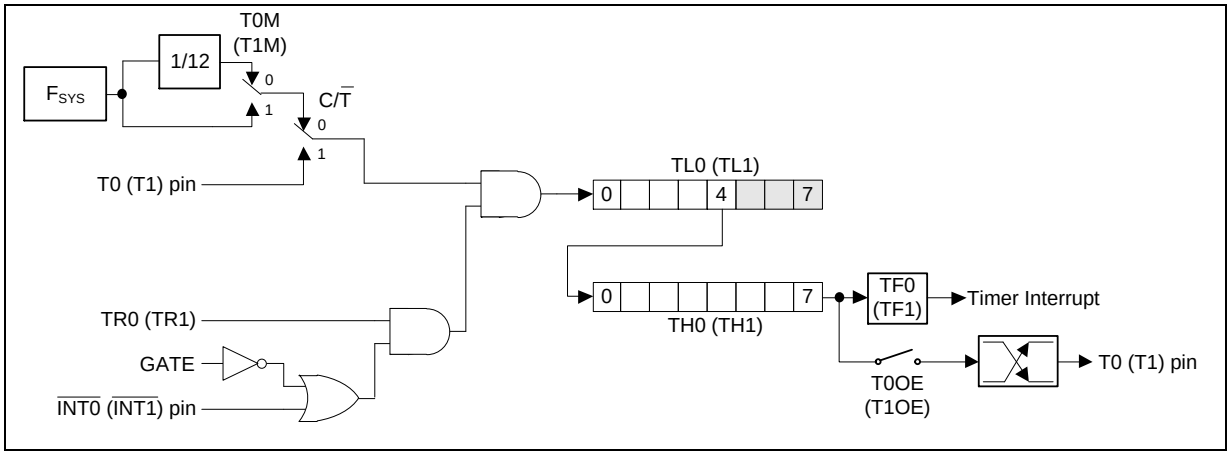


图 6.5-1 定时器/计数器 0 和 1 处于模式 0

6.5.1.2 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。

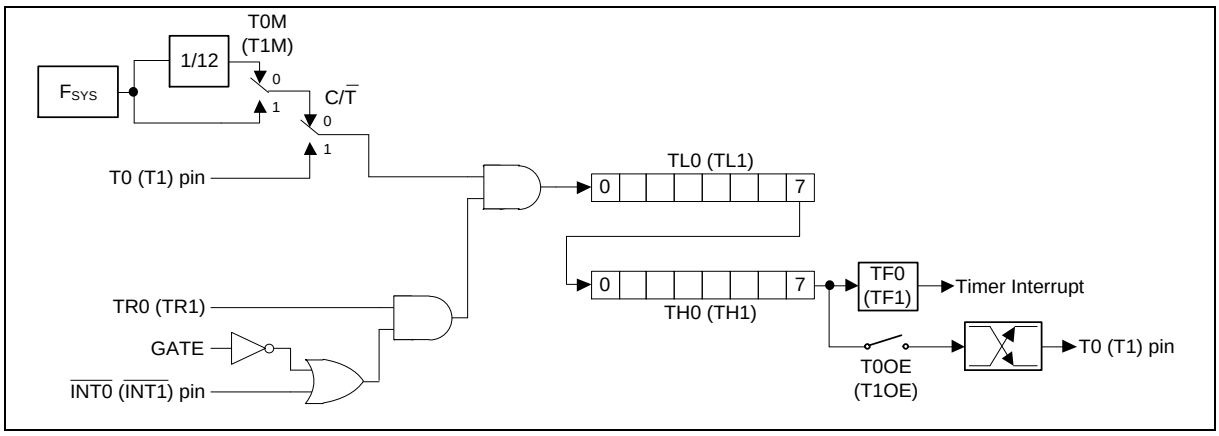


图 6.5-2 定时器/计数器 0 和 1 处于模式 1

6.5.1.3 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$\overline{INT0}$($\overline{INT1}$)引脚，可以启用计数。GATE和$\overline{INT0}$($\overline{INT1}$)引脚的设置与模式0和1相同。

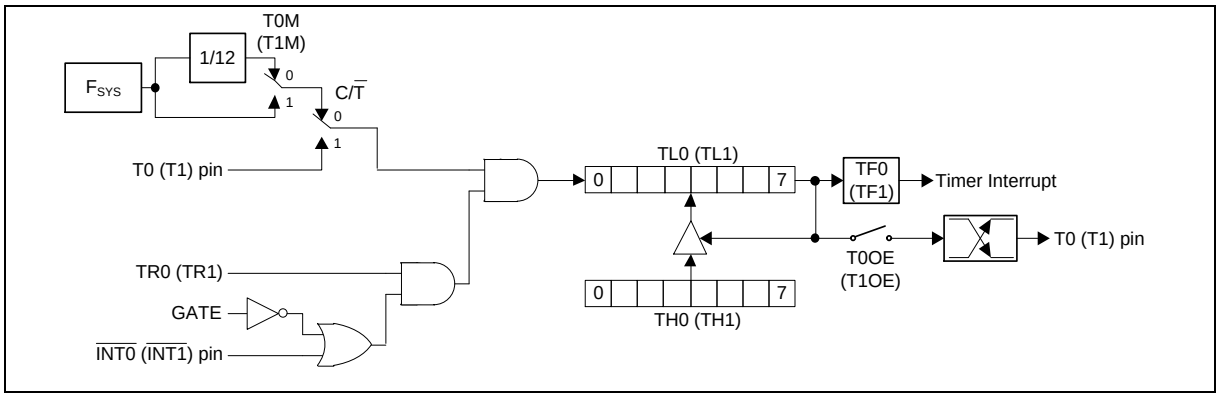


图 6.5-3 定时器/计数器 0 和 1 处于模式 2

6.5.1.4 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。

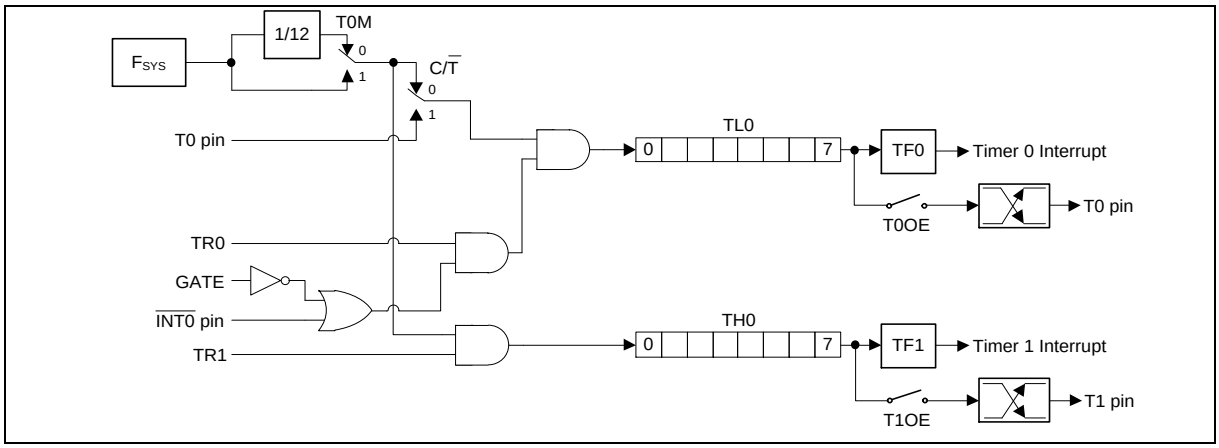


图 6.5-4 定时器/计数器 0 和 1 处于模式 3

6.5.1.5 寄存器描述

TMOD – 定时器0和1模式

寄存器	地址	复位值
TMOD	89H，所有页	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停止
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	地址	复位值
TCON	88H，所有页, 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	只读(电平) 读/写(边沿)	读/写	只读(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。软件不可控制。
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。软件不可控制。
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	地址	复位值
TL0	8AH，所有页	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** 寄存器TL0是定时器0的16位数值的低字节。

TH0 – 定时器0 高字节

寄存器	地址	复位值
TH0	8CH，所有页	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** 寄存器TH0是定时器0 的16位数值的高字节

TL1 – 定时器1低字节

寄存器	地址	复位值
TL1	8BH，所有页	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** 寄存器TL1是定时器1的16位数值的低字节。

TH1 – 定时器1高字节

寄存器	地址	复位值
TH1	8DH，所有页	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** 寄存器TH1是定时器1的16位数值的高字节

CKCON – 时钟控制

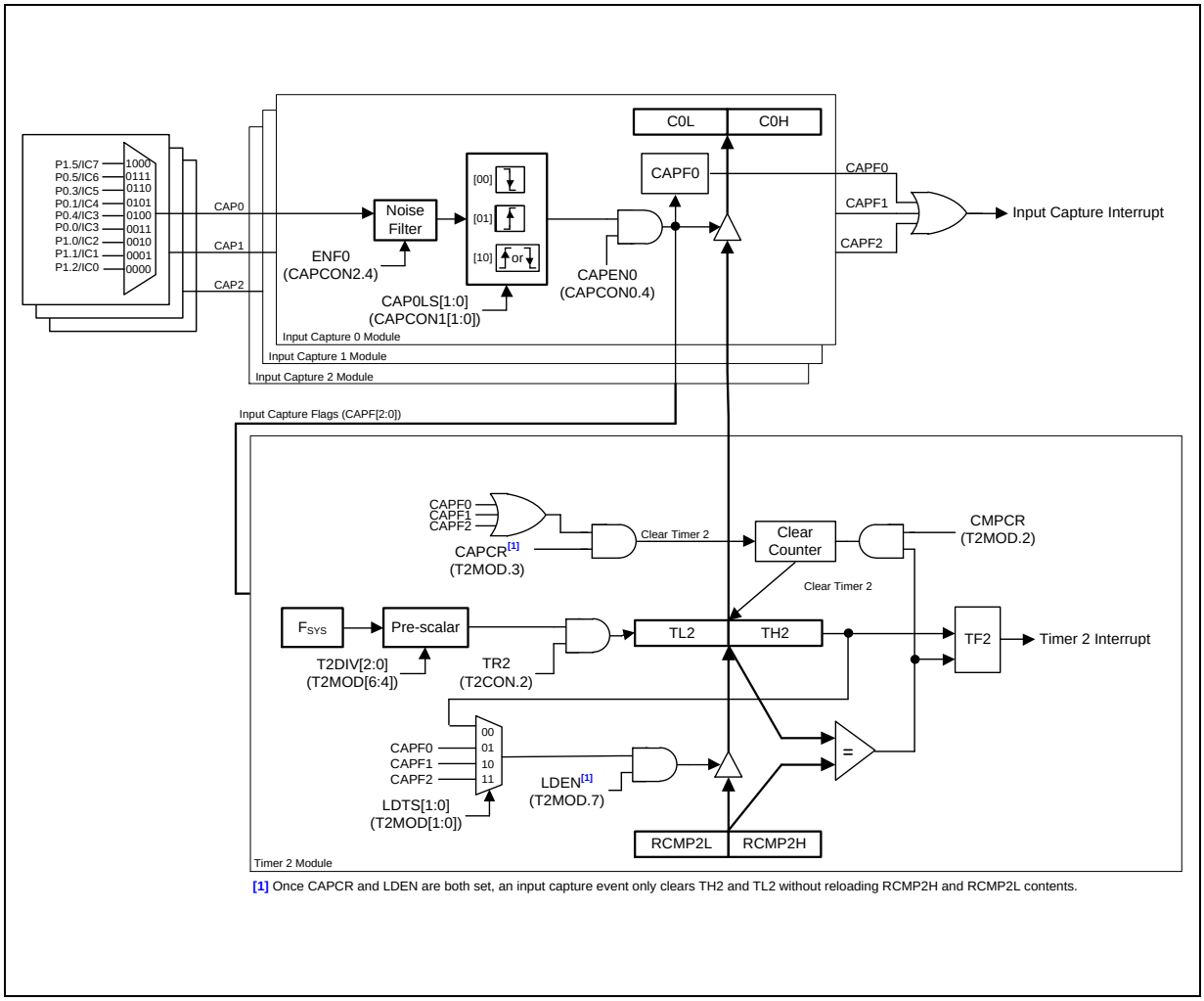
寄存器	地址	复位值
CKCON	8EH，所有页	0000_0000b

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

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。3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。



[1] Once CAPCR and LDEN are both set, an input capture event only clears TH2 and TL2 without reloading RCMP2H and RCMP2L contents.

图 6.5-5 定时器 2 完整功能框图

6.5.2.1 *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。

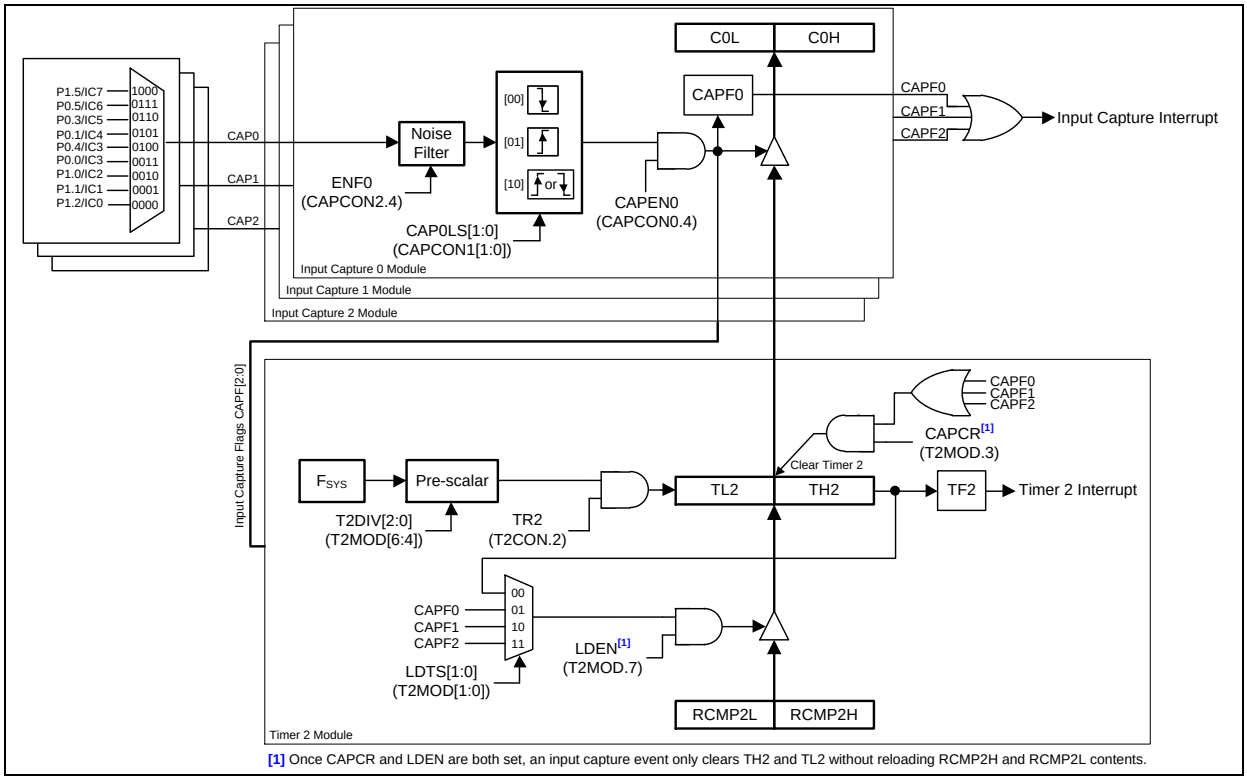


图 6.5-6 定时器 2 自动重载模式框图

6.5.2.2 *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。

6.5.2.3 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。

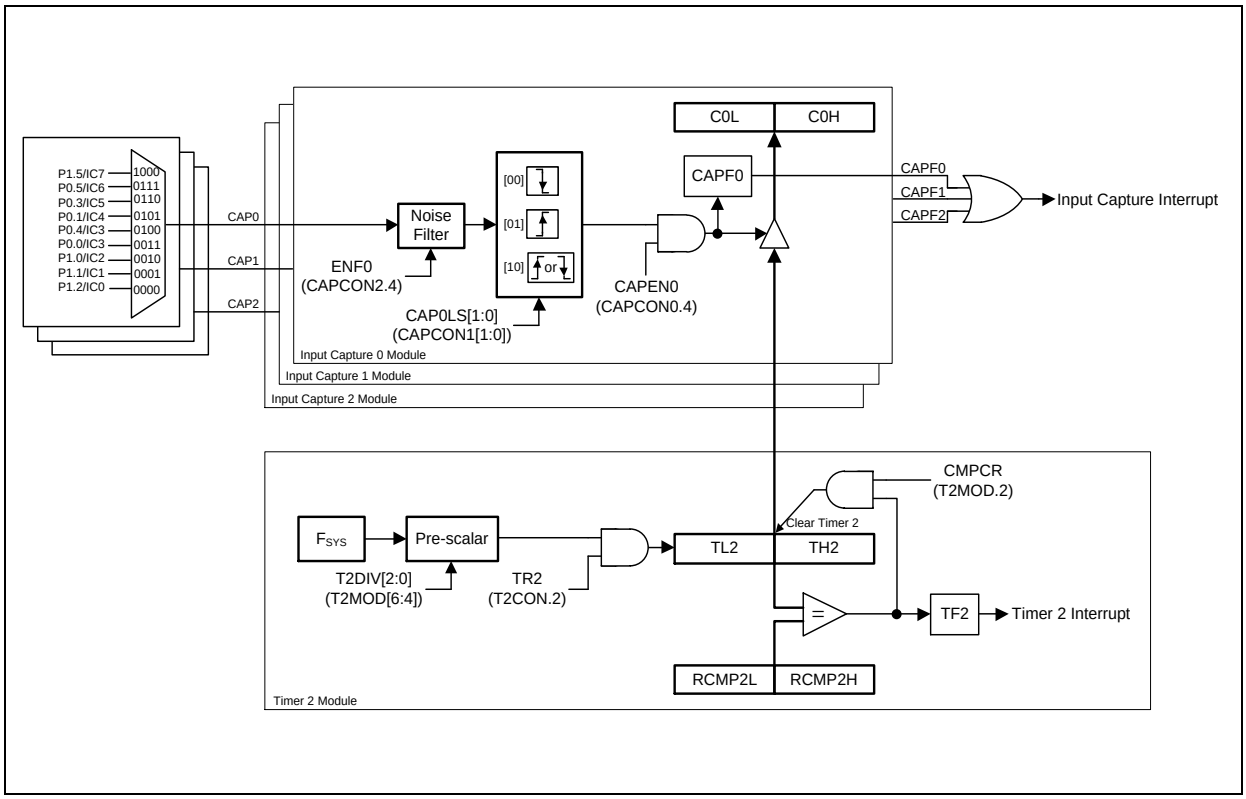


图 6.5-7 定时器 2 比较模式框图

6.5.2.4 寄存器描述

T2CON – 定时器2控制

寄存器	地址	复位值
T2CON	C8H，所有页	0000_0000b

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1。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	地址	复位值
T2MOD	C9H，所有页	0000_0000b

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 = 将重载功能禁用 。 1 = 将RCMP2H和RCMP2L重载至TH2和TL2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	地址	复位值
RCMP2L	CAH，所有页	0000_0000b

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	地址	复位值
RCMP2H	CBH，所有页	0000_0000b

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	地址	复位值
TL2	CCH，所有页	0000_0000b

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	地址	复位值
TH2	CDH, 页0	0000_0000b

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。

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	地址	复位值
CAPCON0	92H，所有页	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	地址	复位值
CAPCON1	93H，所有页	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	地址	复位值
CAPCON2	94H，所有页	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 – 捕获通道n低字节

寄存器	地址	复位值
C0L	E4H，所有页	0000_0000b
C1L	E6H，所有页	0000_0000b
C2L	EDH，所有页	0000_0000b

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 –捕获通道n高字节

寄存器	地址	复位值
C0H	E4H，所有页	0000_0000b
C1H	E7H，所有页	0000_0000b
C2H	EEH，所有页	0000_0000b

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	地址	复位值
CAPCON3	F1H，所有页	0000_0000b

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	地址	复位值
CAPCON4	F2H，所有页	0000_0000b

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

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**錯誤! 找不到參照來源。**“波特率”.

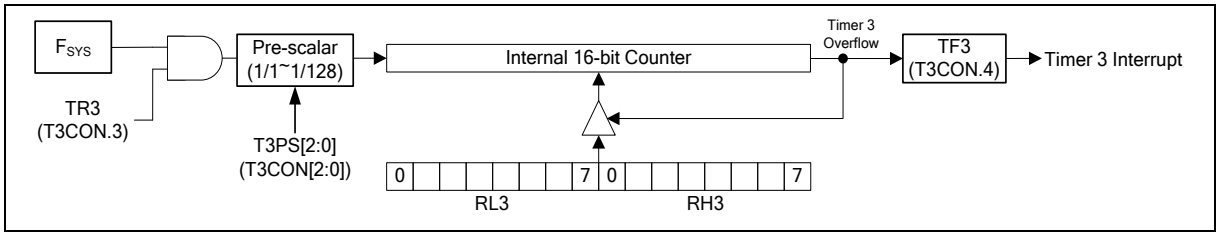


图 6.5-8 定时器 3 框图

6.5.3.1 寄存器描述

T3CON – 定时器3控制

寄存器	地址	复位值
T3CON	C4H, 页0	0000_0000b

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	地址	复位值
RL3	C5H, 页0	0000_0000b

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	地址	复位值
RH3	C6H, 页0	0000_0000b

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。

6.6.2 功能描述

CONFIG4

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

出厂预设值: 1111 1111b

位	名称	描述
7:4	**WDTEN[3:0]**	**看门狗定时器 WDT使能** 此字段配置MCU执行后的WDT行为。 1111 = 看门狗定时器禁用，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保持运行。

WDT带一个独立分频器，用于分频10kHz的低速内部振荡器时钟。分频输出是可选的，决定超时时间间隔。当达到超时间隔时，它将从空闲或掉电模式唤醒系统，如果使能了WDT中断，将发生中断事件。如果将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，经过一段时间的延迟后将发生系统复位。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)。

6.6.2.1 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。

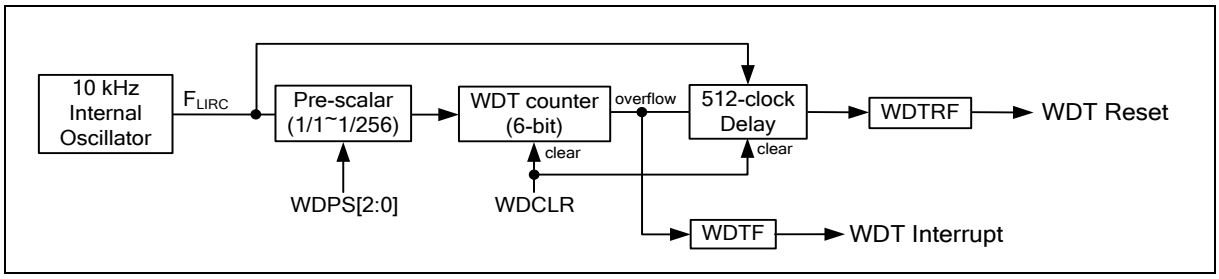


图 6.6-1 看门狗作为一个超时复位定时器

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。

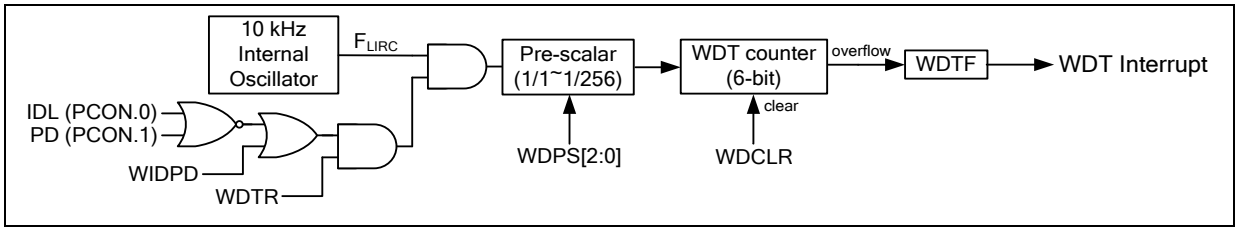


图 6.6-2 看门狗定时器框图

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。

WDCON – 看门狗定时器控制

寄存器	地址	复位值
WDCON	AAH，所有页, TA保护	上电复位 0000_0111b 看门狗 0000_1UUUb 其他 0000_UUUUb

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除。
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。

位	名称	描述
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。

6.7.2 框图

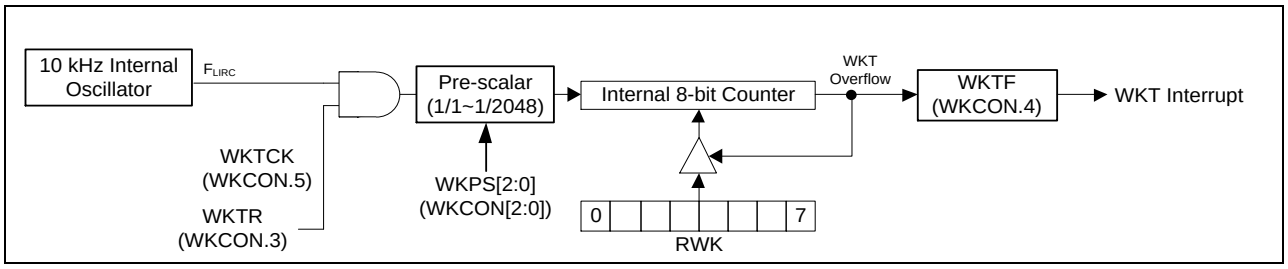


图 6.7-1 自唤醒定时器框图

6.7.3 寄存器描述

WKCON –自唤醒定时器控制器

寄存器	地址	复位值
WKCON	8FH，所有页	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。

位	名称	描述
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暂停(WKTR位为0)的时候可以写入。如果WKT在WKTR为1时被写入，结果不可预测。
2:0	**WKPS[2:0]**	**WKT 预除频** 这些位决定WKT时钟的预除频。 000 = 1/1. 001 = 1/4. 010 = 1/16. 011 = 1/64. 100 = 1/256. 101 = 1/512. 110 = 1/1024. 111 = 1/2048.

RWK – 自唤醒定时器重载字节

寄存器	地址	复位值
RWK	86H，所有页	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 重载字节** 用以存储WKT的8位重载值。注意如果预除频是1/1，RWK限制不能是FFH。

6.8 串口(UART0 和 UART1)

6.8.1 概述

CM1003包含两个增强型全双工串行端口，带有自动地址识别和帧错误检测功能。这两个串行端口的控制位实现相同。一般来说，下文中仅提及串口0，而不再提到串口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，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。

6.8.2 功能描述

6.8.2.1 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6.8-1显示了模式0下串行端口的相关时序。

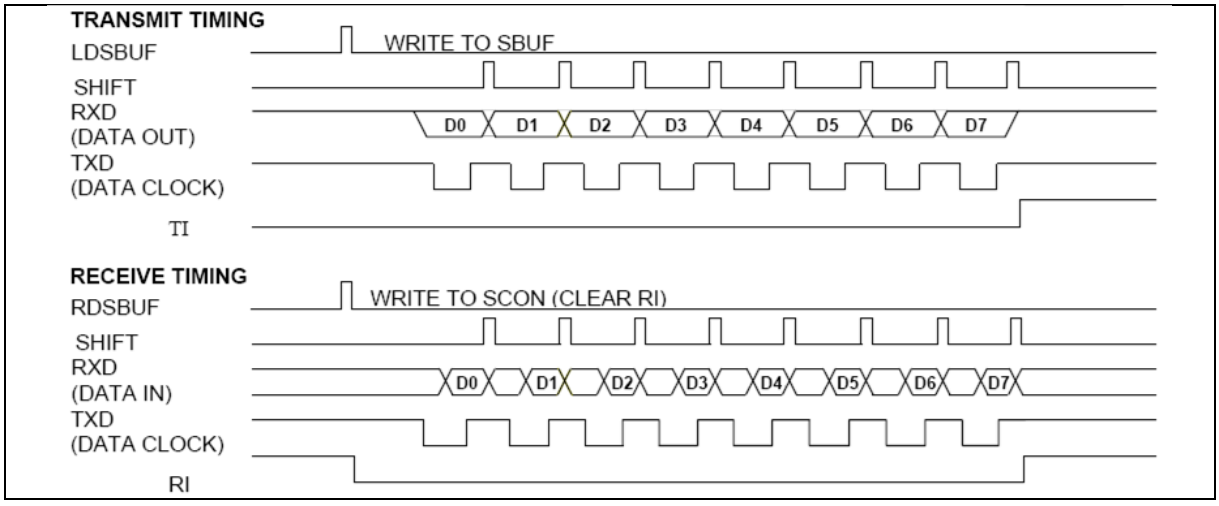


图 6.8-1 串口模式 0 时序图

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6.8-2显示了模式1下串口发送和接收的相关时序。

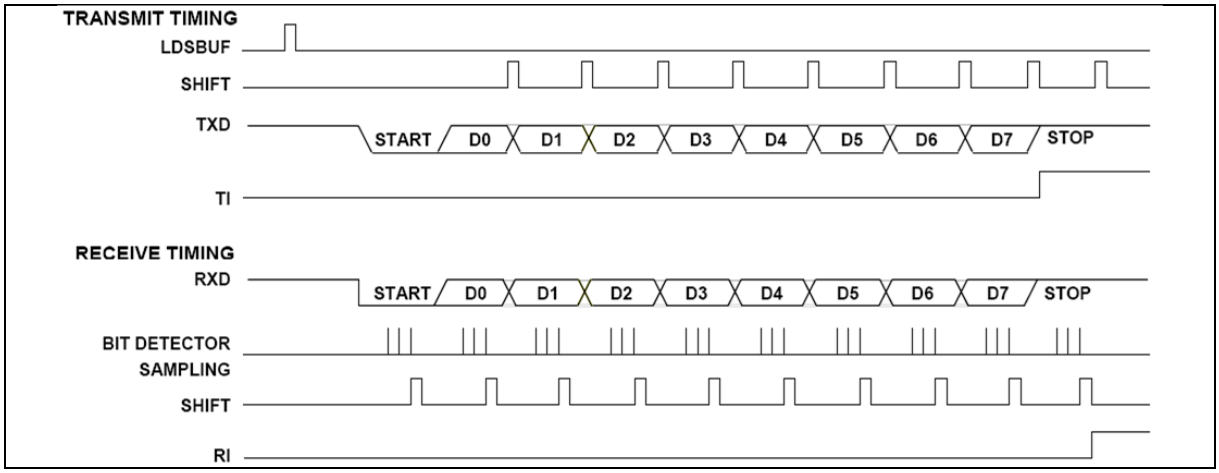


图 6.8-2 串口模式 1 时序图

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. SM2(SCON.5) = 0，或 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**錯誤! 找不到參照來源。** “**錯誤! 找不到參照來源。**” 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 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6.8-3显示了模式2下串行端口发送和接收的相关时序。

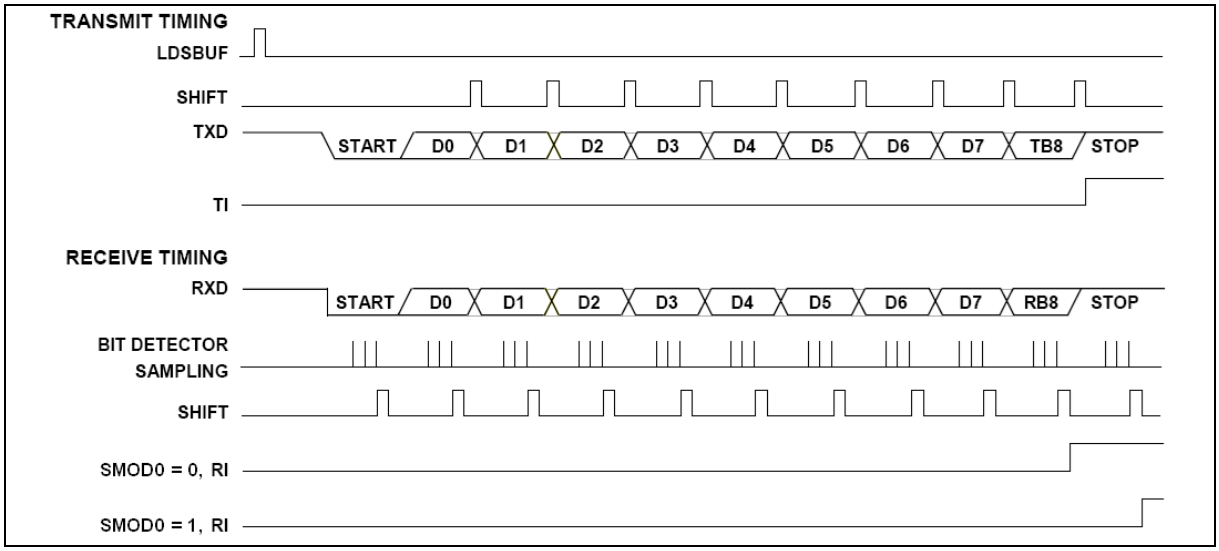


图 6.8-3 串口模式 2 模式 3 的传输时序

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. SM2(SCON.5) = 0，或当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**錯誤! 找不到參照來源。** “**錯誤! 找不到參照來源。**” 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的时序图请参见图图6.8-3它与模式2没有区别。

6.8.2.2 *波特率*

串口不同模式的波特率时钟源和速度差别很大，表6.8-1串口UART0模式/波特率描述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。

以下列出所有UART模式和波特率公式：

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 TM1 CKCON[3] = 0**	$\frac{1}{32}\times\frac{F_{SYS}}{12\times(256-TH1)}$
					Time1 TM1 CKCON[3] = 1	$\frac{1}{32}\times\frac{F_{SYS}}{(256-TH1)}$
					定时器3	$\frac{1}{32}\times\frac{F_{SYS}}{Pre\text{-}scale\times(65536-(256\times RH3+RL3))}$
				1	**Time1 TM1 CKCON[3] = 0**	$\frac{1}{16}\times\frac{F_{SYS}}{12\times(256-TH1)}$
					Time1 TM1 CKCON[3] = 1	$\frac{1}{16}\times\frac{F_{SYS}}{(256-TH1)}$
					定时器3	$\frac{1}{16}\times\frac{F_{SYS}}{Pre\text{-}scale\times(65536-(256\times RH3+RL3))}$
2	11	10	-	0	F_{SYS} 除以64	
				1	F_{SYS} 除以32	
3	11	11	-	0	**Time 1[1] TM1 CKCON[3] = 0**	$\frac{1}{32}\times\frac{F_{SYS}}{12\times(256-TH1)}$
					Time1[1] TM1 CKCON[3] = 1	$\frac{1}{32}\times\frac{F_{SYS}}{(256-TH1)}$
					定时器3	$\frac{1}{32}\times\frac{F_{SYS}}{Pre\text{-}scale\times(65536-(256\times RH3+RL3))}$
				1	**Time1 [1] TM1 CKCON[3] = 0**	$\frac{1}{16}\times\frac{F_{SYS}}{12\times(256-TH1)}$
					Time1[1] TM1 CKCON[3] = 1	$\frac{1}{16}\times\frac{F_{SYS}}{(256-TH1)}$
					定时器3	$\frac{1}{16}\times\frac{F_{SYS}}{Pre\text{-}scale\times(65536-(256\times RH3+RL3))}$
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 6.8-1 串口 UART0 模式/波特率描述

模式	帧位	SM0 / SM1 (SCON[7:6])	SM2 (SCON[5])	SMOD (PCON[7])	
0	8	00	-	F_{SYS} 分频12	
1	10	01	0	**定时器3**	$\frac{1}{32}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$
			1	**定时器3**	$\frac{1}{16}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$
2	11	10	0	F_{SYS} 分频64	
			1	F_{SYS} 分频32	
3	11	11	0	**定时器3**	$\frac{1}{32}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$
			1	**定时器3**	$\frac{1}{16}\times\frac{F_{SYS}}{\text{Pre-scale}\times(65536-(256\times RH3+RL3))}$

表 6.8-2 串口 UART1 模式/波特率描述

例程: 我们列出通用UART设置模式1初始化步骤如下：

串口0(**UART0**)用定时器**1**作为波特率发生器：公式为 $\frac{1}{16}\times\frac{F_{SYS}}{(256-TH1)}$

```
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**)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 = 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)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及和精度值：

Fsys 值	波特率	TH1值 (十六进制)	RH3,RL3 值(十六进制)	波特率精度
24MHz	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%
16MHz	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%

6.8.2.3 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。

6.8.2.4 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。

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。

6.8.2.5 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。

6.8.3 寄存器描述

SCON – 串口控制

寄存器	地址	复位值
SCON	98H，所有页，可位寻址	0000_0000b

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6.8-1串口UART0模式/波特率描述了解详情。 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	
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。

SCON_1 – 串口1 控制

寄存器	地址	复位值
SCON_1	F8H，所有页， 可位寻址	0000_0000b

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6.8-2 串口UART1模式/波特率描述。 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	地址	复位值
PCON	87H，所有页	上电复位, 0001_0000b 其他，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
7	**SMOD**	**串口0双倍波特率使能** 当 串口0处于模式2，或将定时器1溢出用作串口0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。参见表6.8-1串口UART0模式/波特率描述 了解详情。
6	**SMOD0**	**串口0帧错误访问使能** 0 = SCON.7访问SM0位。 1 = SCON.7访问FE位。

T3CON – 定时器3控制

寄存器	地址	复位值
T3CON	C4H, 页0	0000_0000b

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** 当串口1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6.8-2 串口UART1模式/波特率描述。
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	地址	复位值
SBUF	99H，所有页	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	地址	复位值
SBUF_1	9AH，所有页	0000_0000b

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(可位寻址)

寄存器	地址	复位值
IE	A8H，所有页, 可位寻址	0000_0000b

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	地址	复位值
EIE1	9CH，所有页	0000_0000b

7	6	5	4	3	2	1	0
-	-	-	-	-	EWKT	ET3	ES_1
-	-	-	-	-	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	地址	复位值
SADDR	A9H，所有页	0000_0000b

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	地址	复位值
SADEN	B9H，所有页	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** 此字节为串口0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	地址	复位值
SADDR_1	BBH，所有页	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	地址	复位值
SADEN_1	BAH，所有页	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** 此字节为串口1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	地址	复位值
AUXR1	A2H，所有页	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。

6.9 串行外设接口(SPI)

6.9.1 概述

CM1003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。它提供主机模式或从机模式，速度可高达F_{SYS}/4，支持传输完成标志位和写冲突标志位。对于多主机系统，SPI支持主机模式故障以防止多主机冲突。

6.9.2 结构框图

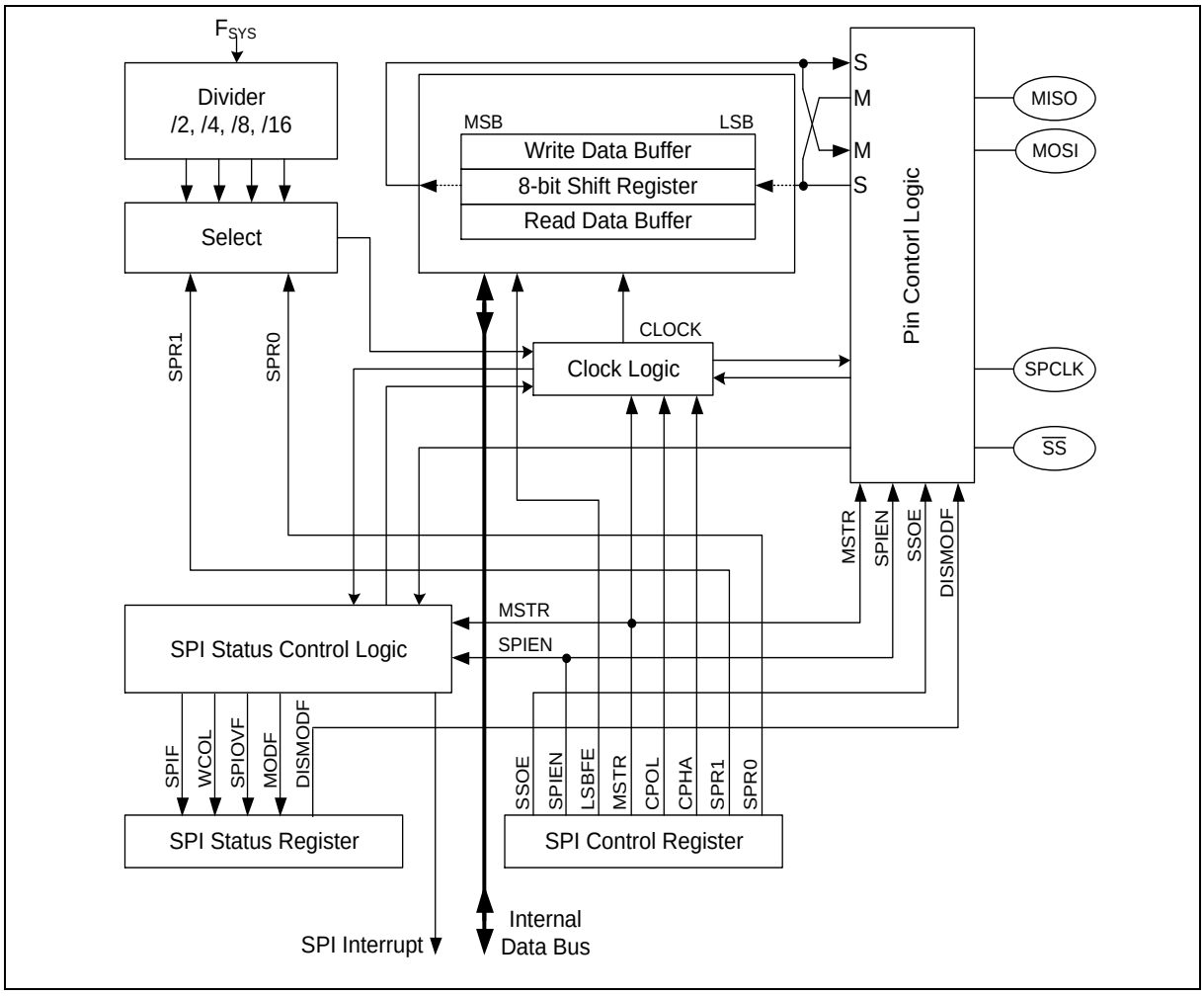


图 6.9-1 SPI 结构框图

图6.9-1 SPI 展示了SPI的结构功能。SPI的主要模块有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。为了传送数据和接收数据，SPI提供了一个写入数据缓冲区、一个移位寄存器和一个读数据缓冲区。数据接收和发送是双缓冲区的。前一个传输未完成时写入SPDR的新发送数据是会写入移位寄存器的（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。

6.9.3 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引脚在主机模式下是时钟输出，但在从机模式下是时钟输入。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，八个时钟周期在总线上交换一个字节数据。由于移位时钟始终由主机产生，因此系统不应存在多个处于主机模式的设备，以避免主机冲突。

每个从机外设都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。当SS为高，该从机将处于停用状态。若为多从机模式，则同一时间只能选择一个从机设备。对于主机模式MCU，该SS引脚不起作用，可配置为普通端口另做他用。多主机环境下，SS也可以由软件设置用作主机故障检测(参见章节6.9.3.5 模式故障检测)。CM1003也提供自动激活功能，在字节传输间切换SS。

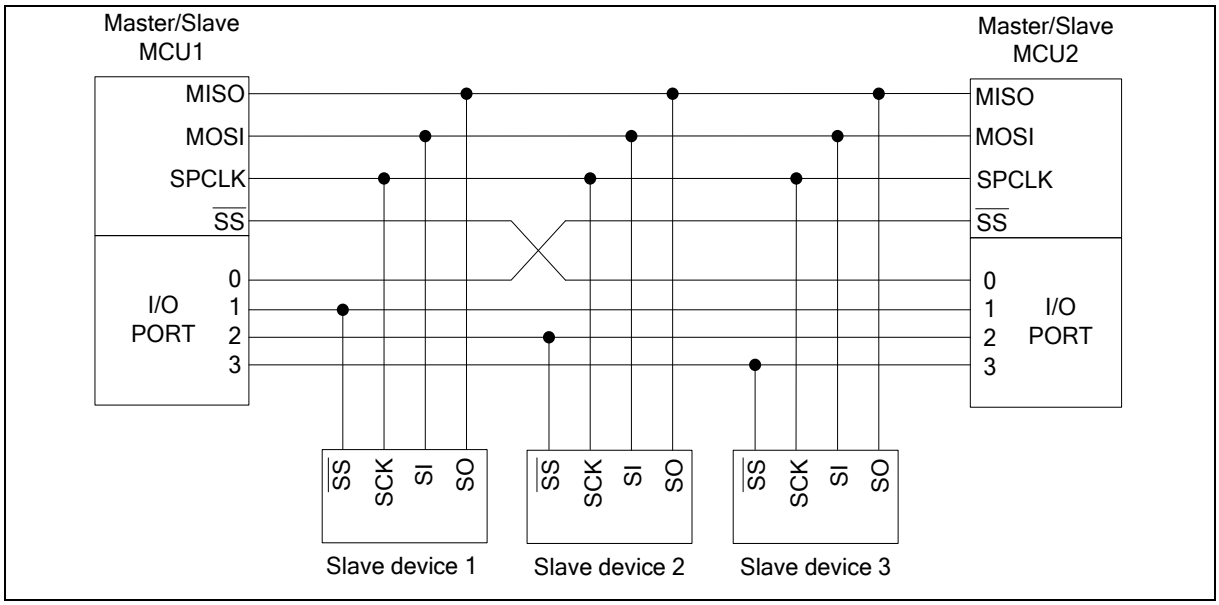


图 6.9-2 SPI 多主机，多从机连接图

图6.9-2 SPI 显示了SPI设备间的典型互连。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。。主机设备使用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可以是主机或从机模式。应将SS引脚配置为主机模式故障检测，以避免多主机冲突。

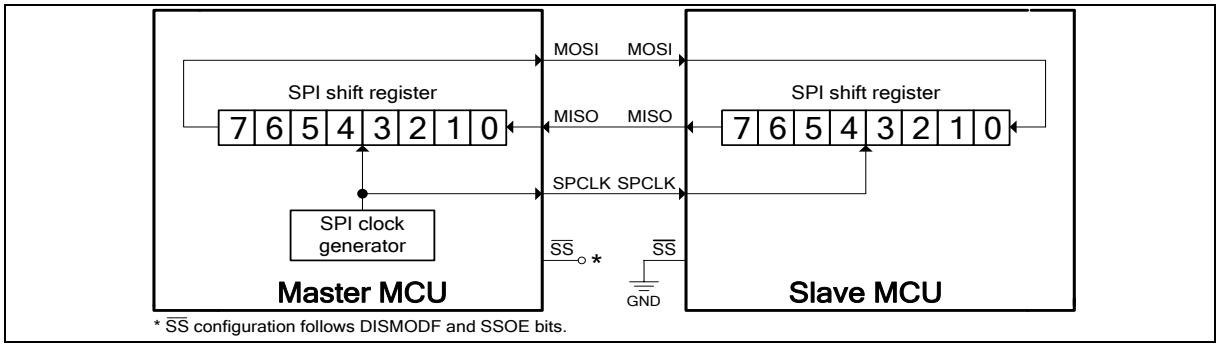


图 6.9-3 SPI 单主机，单从机连接图

图6.9-3 SPI 显示了最简单的SPI系统互连，即单主机和单从机。在传输时，主机通过MOSI线向从

机发送数据。同时，主机也通过MISO线由从机接收数据。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，当主机向从机发送数据时，从机数据也同时推向主机。该传输实际上交换了两个 MCU 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。如果设置了LSBFE(SPCR.5)，SPI数据将先传输低位(LSB)。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，状态检测，数据存储功能和时钟速率选择。以下寄存器与SPI功能相关。

DISMODF	SSOE	主机模式(MSTR = 1)	从机模式(MSTR = 0)
0	X	$\overline{SS}$ 为模式故障输入	$\overline{SS}$ 为从属选择输入
1	0	通用输入/输出	
1	1	自动$\overline{SS}$输出	

表 6.9-1 从机引脚配置选择

6.9.3.1 *工作模式*

主机模式

当MSTR(SPCR.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(SPSR.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。

6.9.3.2 *SPI 主机时钟配置*

这两位选择SPI的四个时钟除频。以下时钟速率是在FSYS = 16/24MHz条件下说明的。SPR[1:0] 仅在主机模式(MSTR = 1)下有效。如果在从机模式下，时钟将自动与主机设备SPICLK引脚上的外部时钟同步，最高可达F_{SYS}/4 通信速度。以下是SPI时钟速率表

F_{SYS} = 16 MHz				F_{SYS} = 24 MHz			
SPR[1] SPCR.1	SPR[0] SPCR.0	除频	SP时钟速率	SPR[1] SPCR.1	SPR[0] SPCR.0	除频	SP时钟速率
0	0	2	8M bit/s	0	0	2	12M bit/s

0	1	4	4M bit/s	0	1	4	6M bit/s
1	0	8	2M bit/s	1	0	8	3M bit/s
1	1	16	1M bit/s	1	1	16	1.5M bit/s

表 6.9-2 SPI 主机时钟速率表

SPIS[1:0](SPI0CR1[1:0])提供在主机模式下两笔连续的传输字节之间可设置的挂起时间间隔，0.5 ~ 2.5个SPI时钟周期。挂起时间间隔的是指前一个传输字的最后一个时钟沿与下一个传输字的第一个时钟沿之间的间隔。以下显示了挂起时间间隔的选择。

CPHA SPCR0.2	SPIS[1] SPCR1.1	SPIS[0] SPCR1.0	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

表 6.9-3 SPI 时钟暂停间隔选择

6.9.3.3 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(SPCR.3)和时钟相位位CPHA(SPCR.2)。图6.9-4 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。

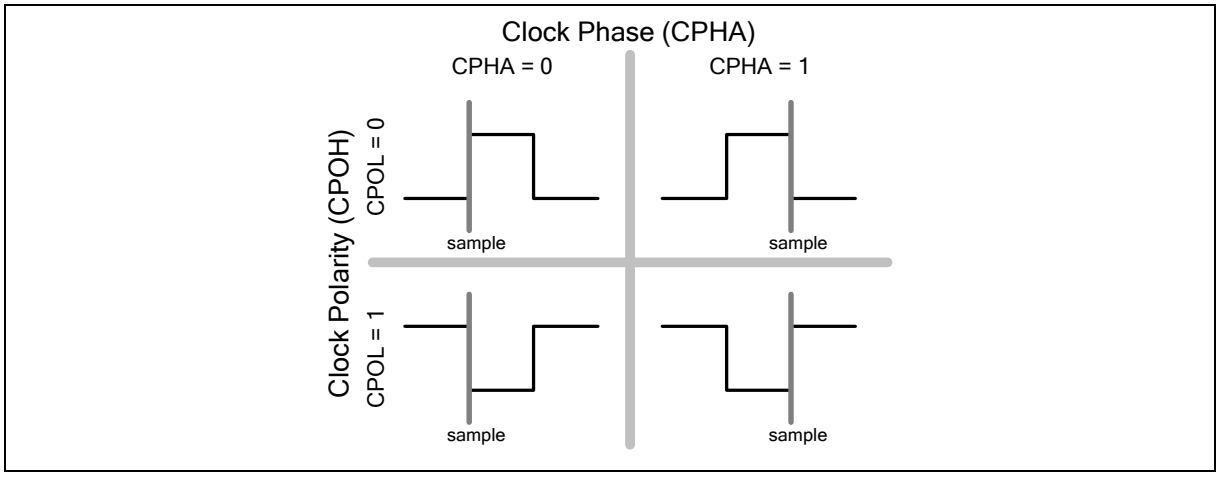


图 6.9-4 SPI 时钟格式

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(SPSR.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。如图图6.9-4 SPI ，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。

请注意：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。

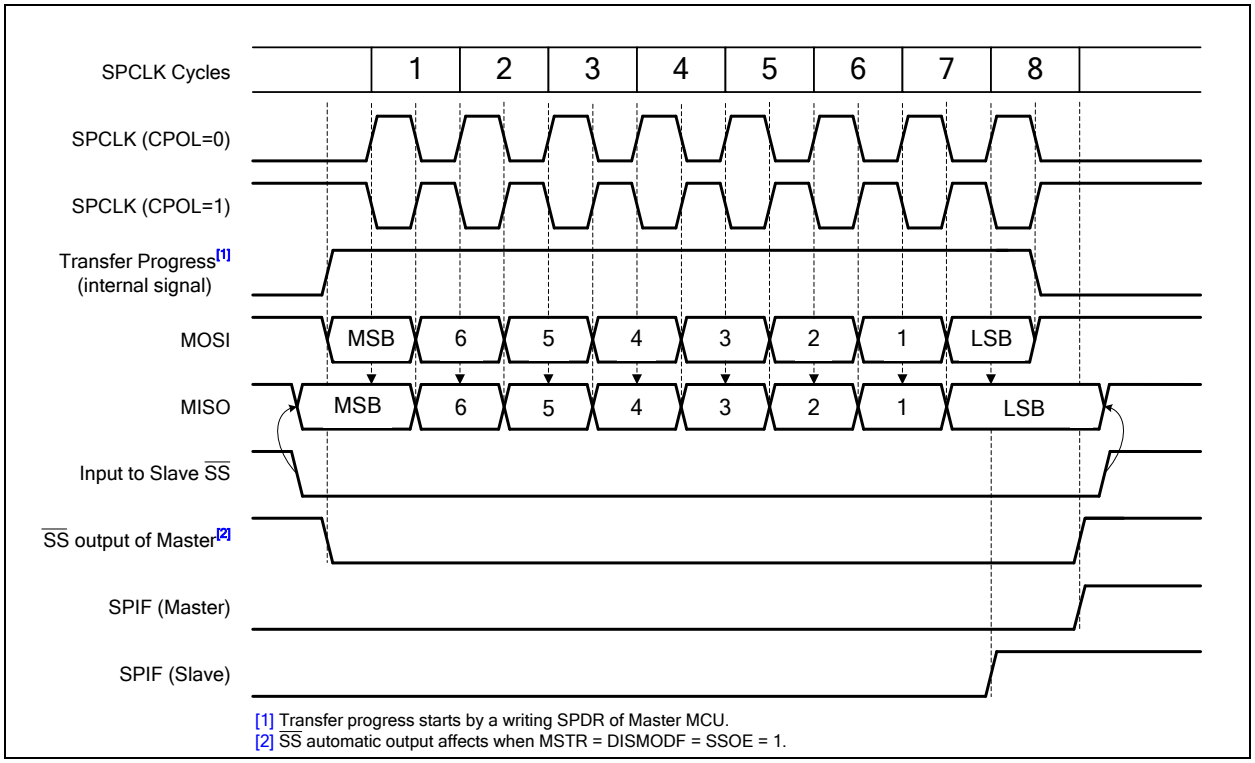


图 6.9-5 CPHA = 0 的 SPI 时钟和数据格式

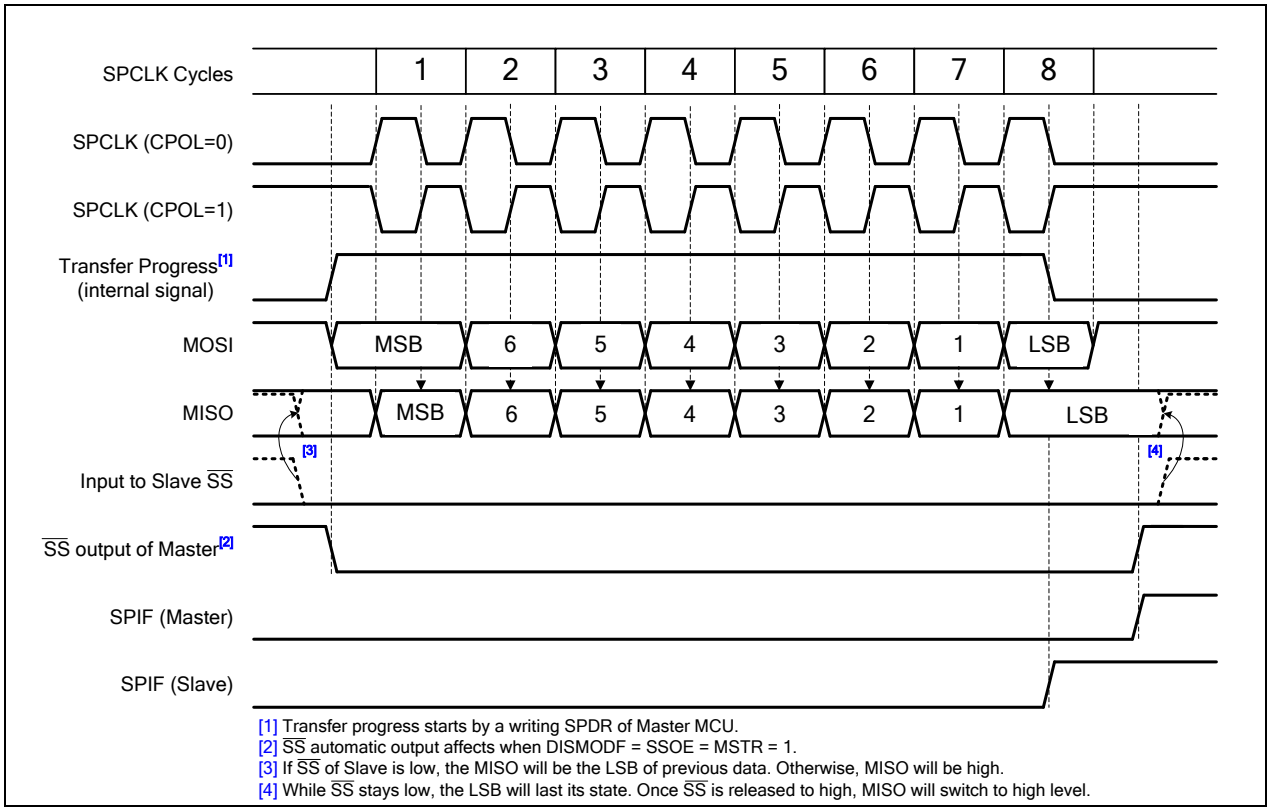


图 6.9-6 CPHA = 1 的 SPI 时钟和数据格式

6.9.3.4 从机选择引脚配置

CM1003 SPI提供灵活的SS引脚功能，可满足不同的系统要求。当SPI作为从机运行时，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，SS引脚根据DISMODF(SPSR.3)和SSOE(SPCR.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，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 SS 作为输出引脚通常与从机的SS输入引脚相连。在选择外部从机时，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， 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	SS故障模式输入	SS从机选择输入引脚
1	0	通用I/O	
1	1	自动SS输出	

表 6.9-4 从机选择引脚配置

6.9.3.5 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SS被配置为模式故障输入引脚(具体参考表6.9-1 ，一旦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SPCR中的MSTR和SPIEN控制位以禁用 SPI，设置模式故障标志位 MODF(SPSR.4)，并在ESPI和EA都使能时生成中断。

6.9.3.6 *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(SPSR.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。

6.9.3.7 *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(SPSR.5)被硬件置位1。如果中断使能，SPIOVF置位会产生中断请求。图6.9-7 SPI 显示了数据接收与溢出错误之间的关系。

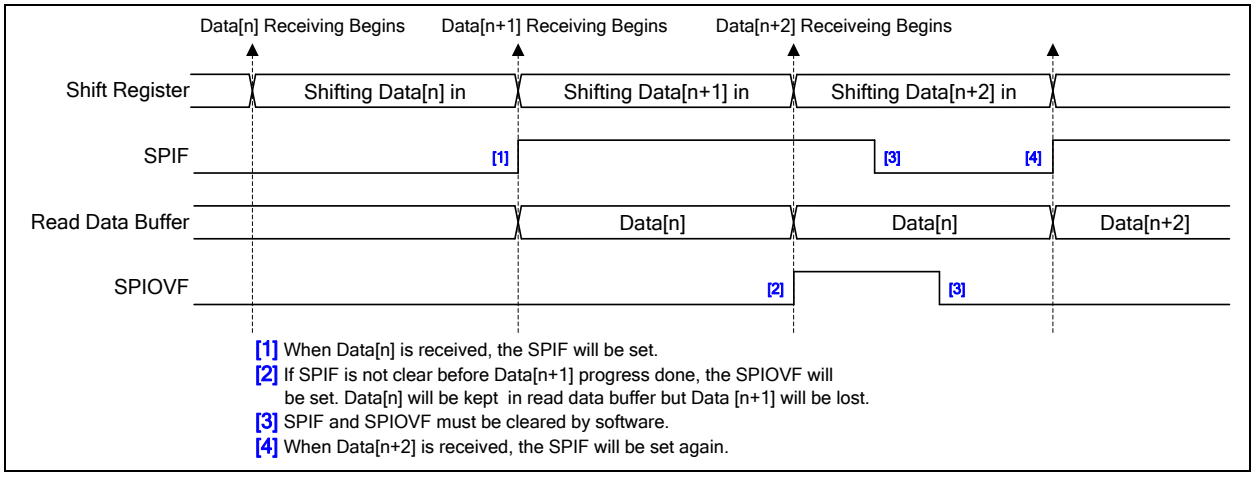


图 6.9-7 SPI 溢出波形

6.9.3.8 *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。

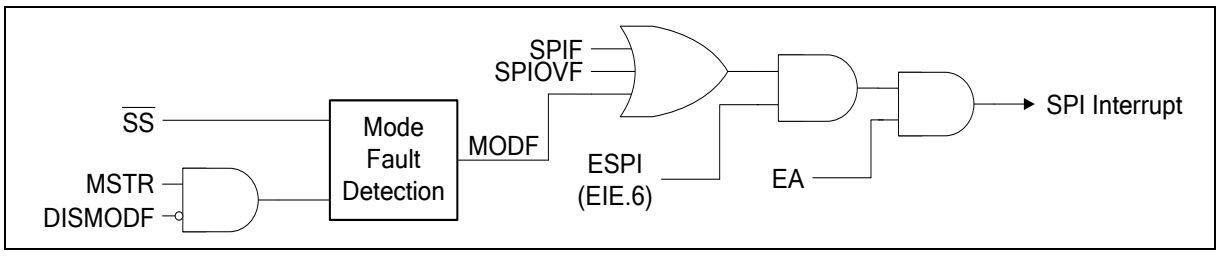


图 6.9-8 SPI 中断请求

6.9.4 寄存器描述

SPCR – 串行外设SPI控制寄存器

寄存器	地址	复位值
SPCR	F3H，页0	0000_0000b

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6.9-4 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6.9-4 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。 Fsys = 16MHz. SPR1 / SPR0 / 分频 / SPI 时钟速率 0 / 0 / 2 / 8M bit/s 0 / 1 / 4 / 4M bit/s 1 / 0 / 8 / 2M bit/s 1 / 1 / 16 / 1M bit/s Fsys = 24MHz. 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SPI控制寄存器 2

寄存器	地址	复位值
SPCR2	F3H，页1	0000_0000b

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 相邻字节间的间隔时间选择** 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]仅在机主机模式(MSTR = 1)下有效。

SPSR – 串行外设SPI状态寄存器

寄存器	地址	复位值
SPSR	F4H，所有页	0000_0000b

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。 如果使能ESPI和 EA，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** 此位指示写入冲突事件。一旦发生写入冲突事件，此位被置位，应由软件清零。
5	SPIOVF	**SPI 溢出标志** 此位指示溢出事件。一旦发生溢出，此位被置位。如果使能ESPI 和 EA， 产生SPI 中断请求。此位应由软件清零。
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6.9-1 选择所示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	地址	复位值
SPDR	F5H，所有页	0000_0000b

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。

6.10 内部集成电路(I²C)

6.10.1 概述

CM1003提供两条内部集成电路(I²C)总线，用作微控制器和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)在设备之间传输数据。

I²C总线在主机和从机之间双向数据传输。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。同步时钟SCL的存在，允许具有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。支持广播呼叫，支持标准速率传输(100kbps)和快速传输(400k bps)。

6.10.2 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。

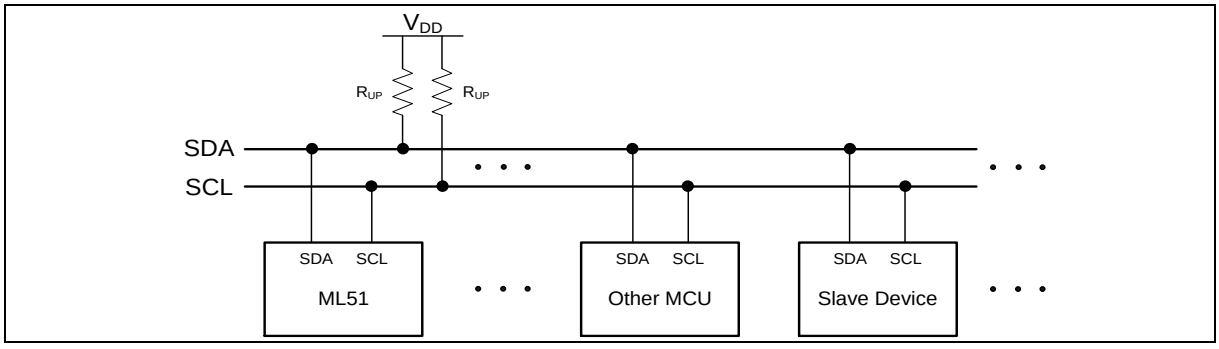


图 6.10-1 I²C 总线互联

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。

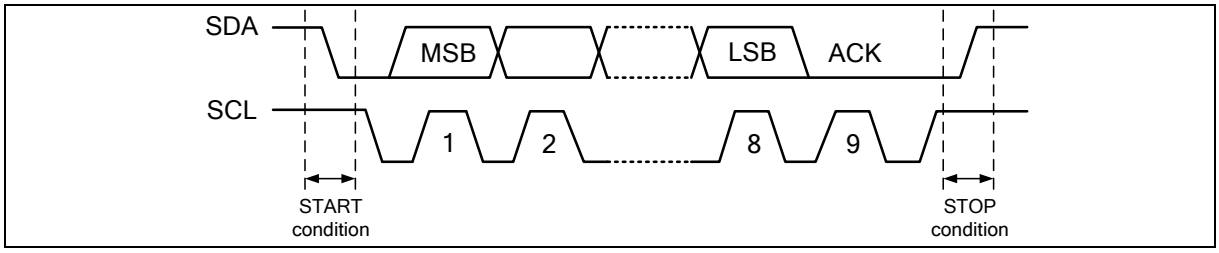


图 6.10-2 I²C 总线协议

6.10.2.1 开始和停止条件

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。

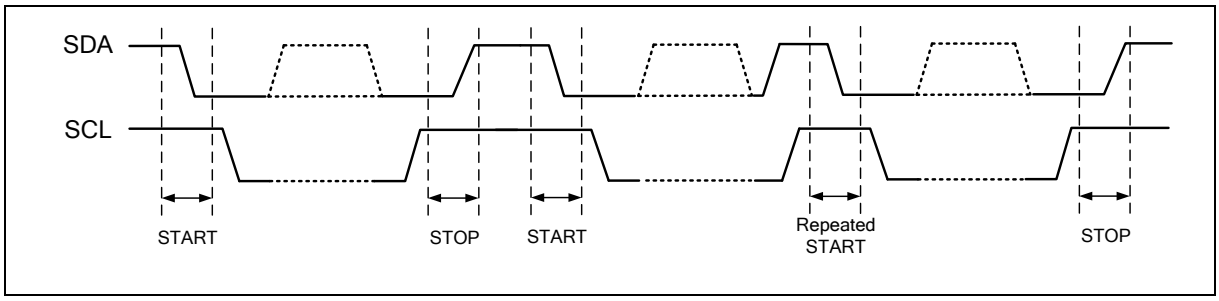


图 6.10-3 START 起始条件, Repeated START 重复起始条件和 STOP 停止条件

6.10.2.2 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起始条件START，一个SLA+W/R、一个或多个数据包和一个停止条件STOP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6.10-4 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。

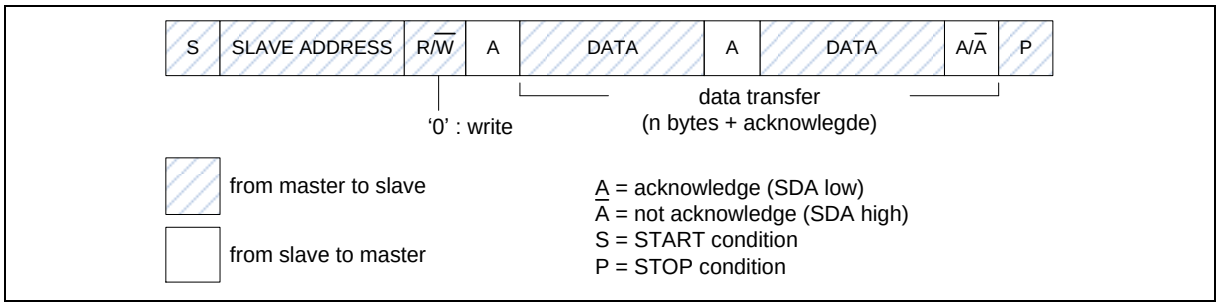


图 6.10-4 主机以 7 位数据格式向从机发送数据

图6.10-5 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。

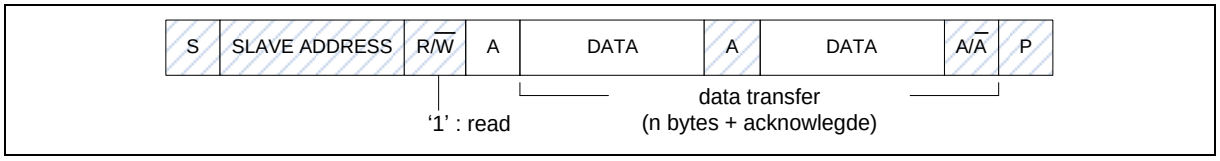


图 6.10-5 主机以 7 位数据格式向从机读取数据

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。

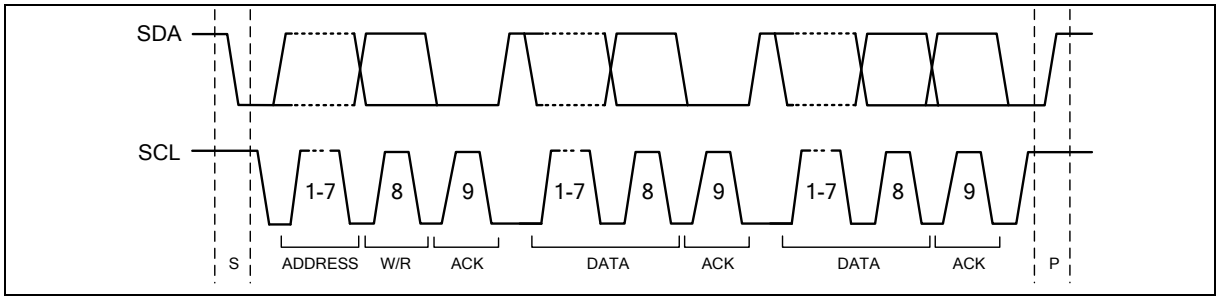


图 6.10-6 一次 I²C 传输的数据格式

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。

6.10.2.3 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。

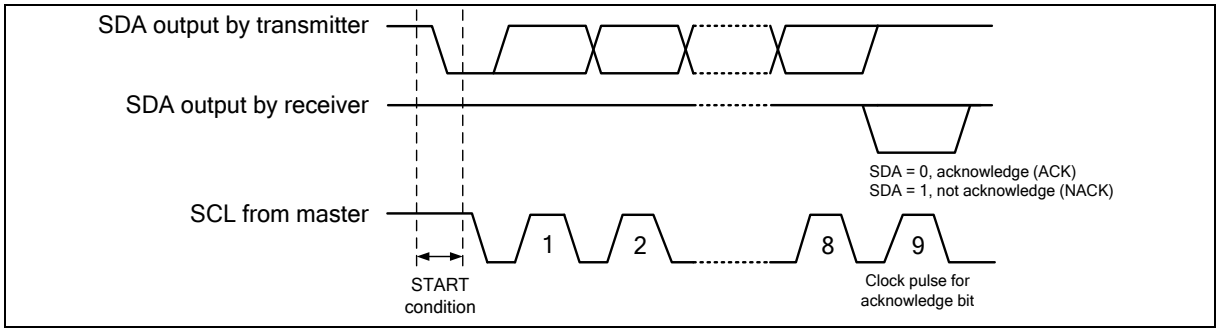


图 6.10-7 应答位

6.10.2.4 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。

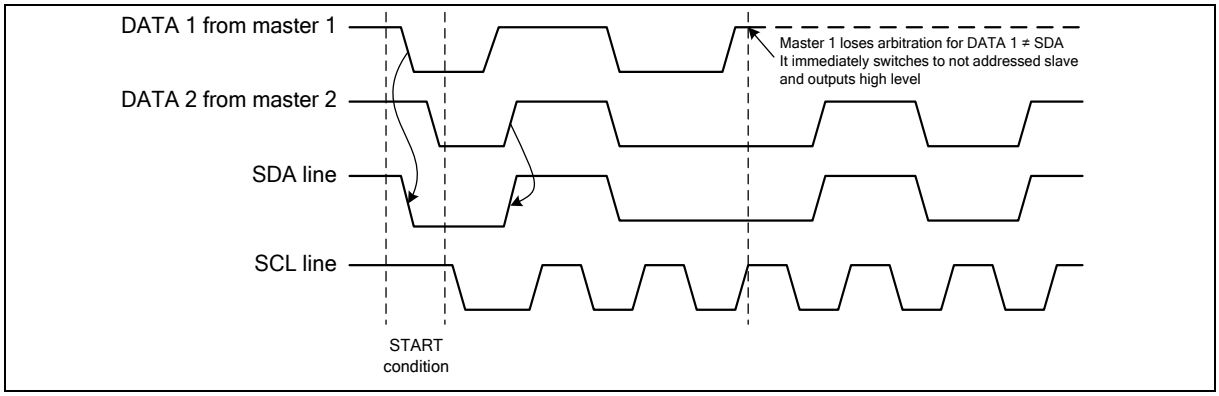


图 6.10-8 两台主机的仲裁过程

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。

6.10.2.5 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6.10-9 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。

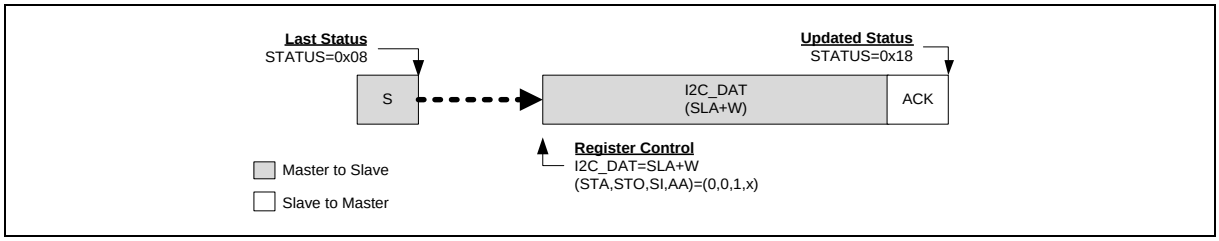


图 6.10-9 根据当前 I²C 状态来控制 I²C 总线

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。

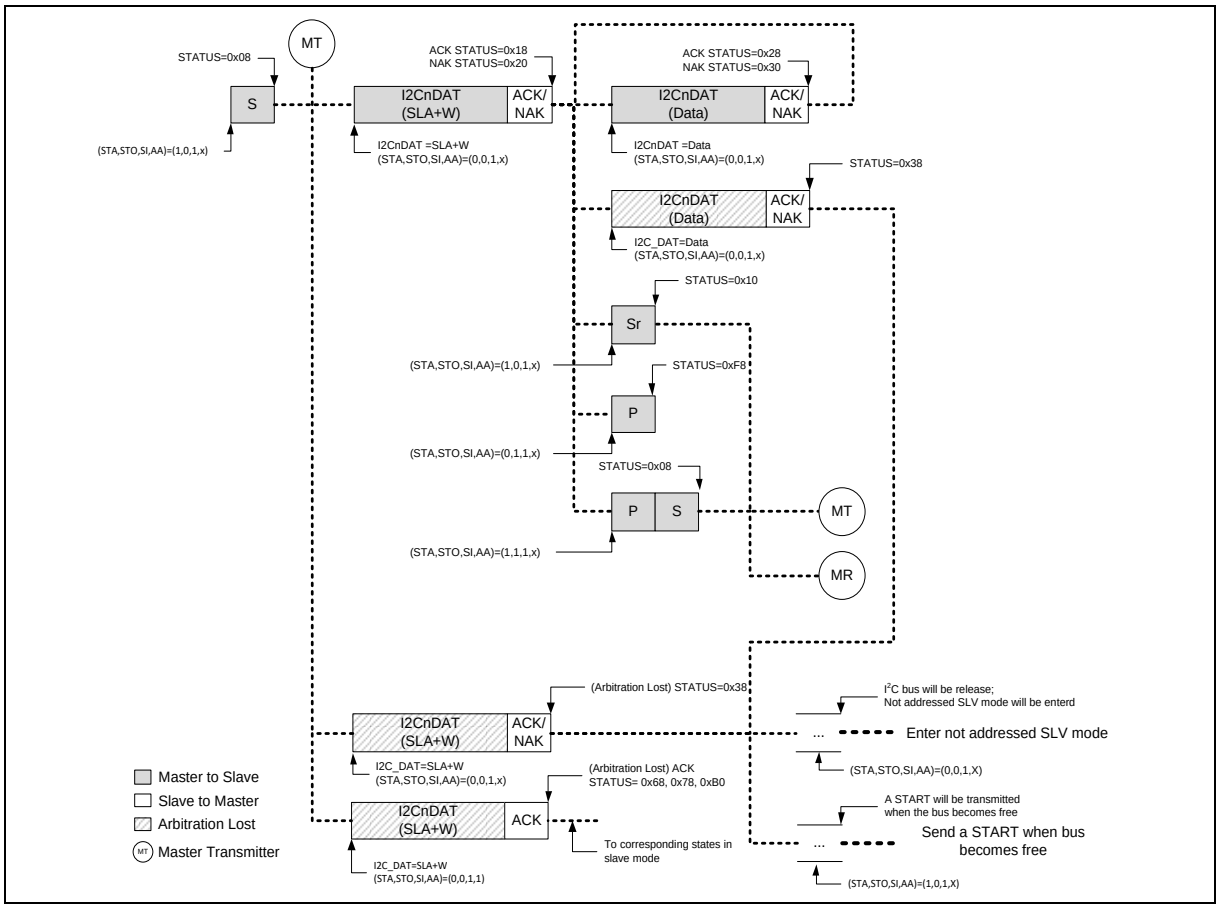


图 6.10-10 主机发送模式流程和状态

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。

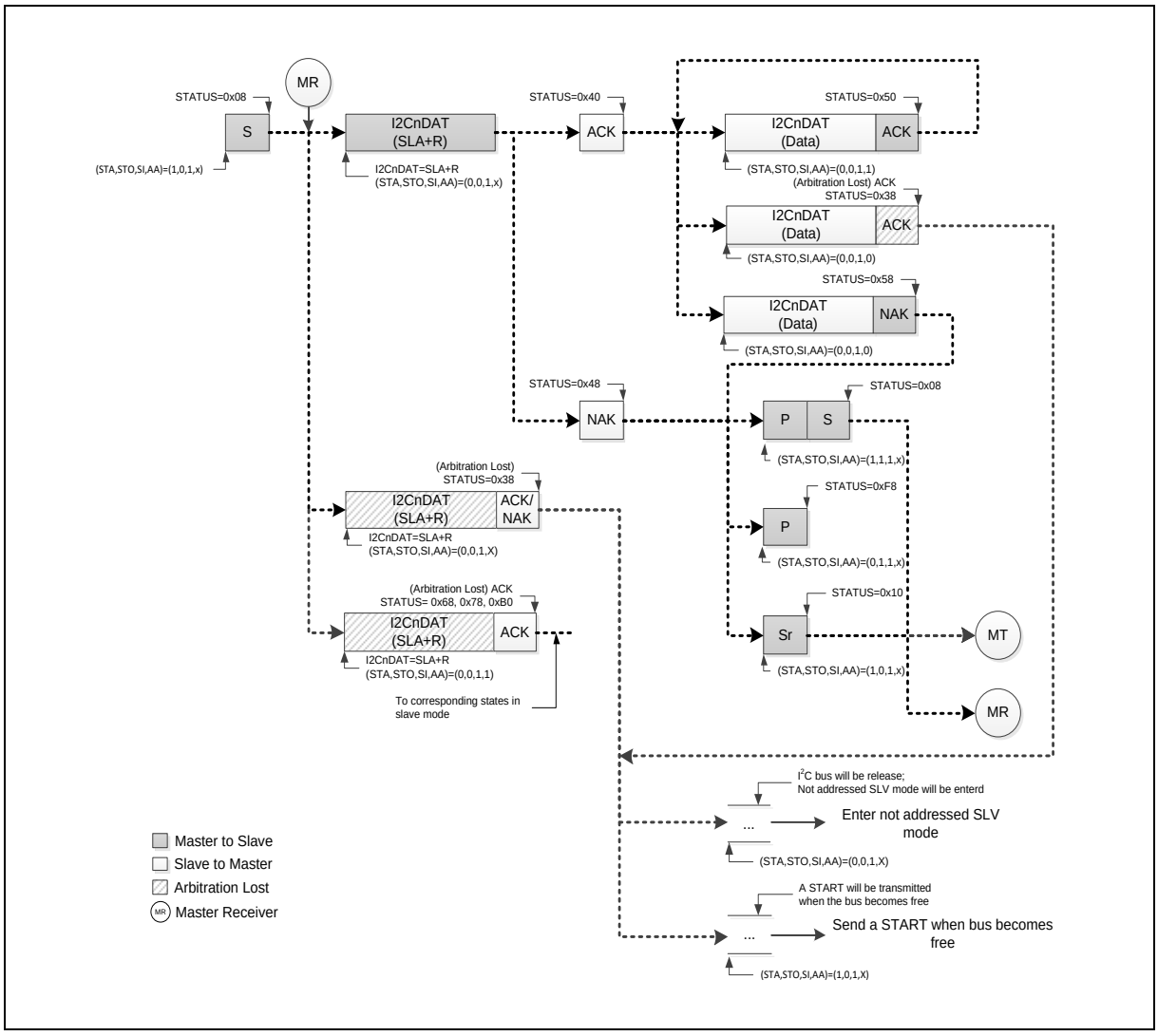


图 6.10-11 主机接收模式流程和状态

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。

从机发射模式

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。

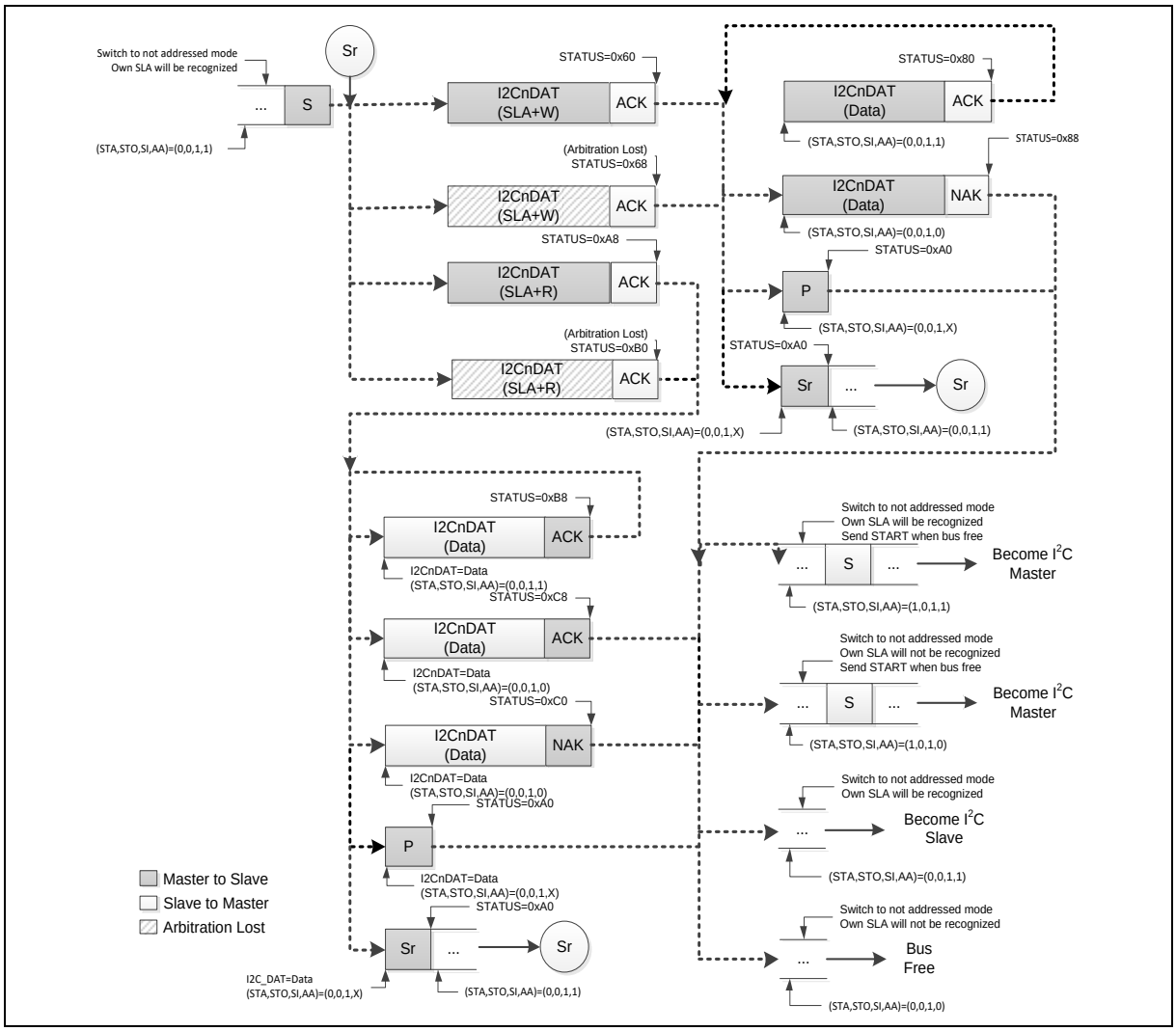


图 6.10-12 从机接收模式流程和状态

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。

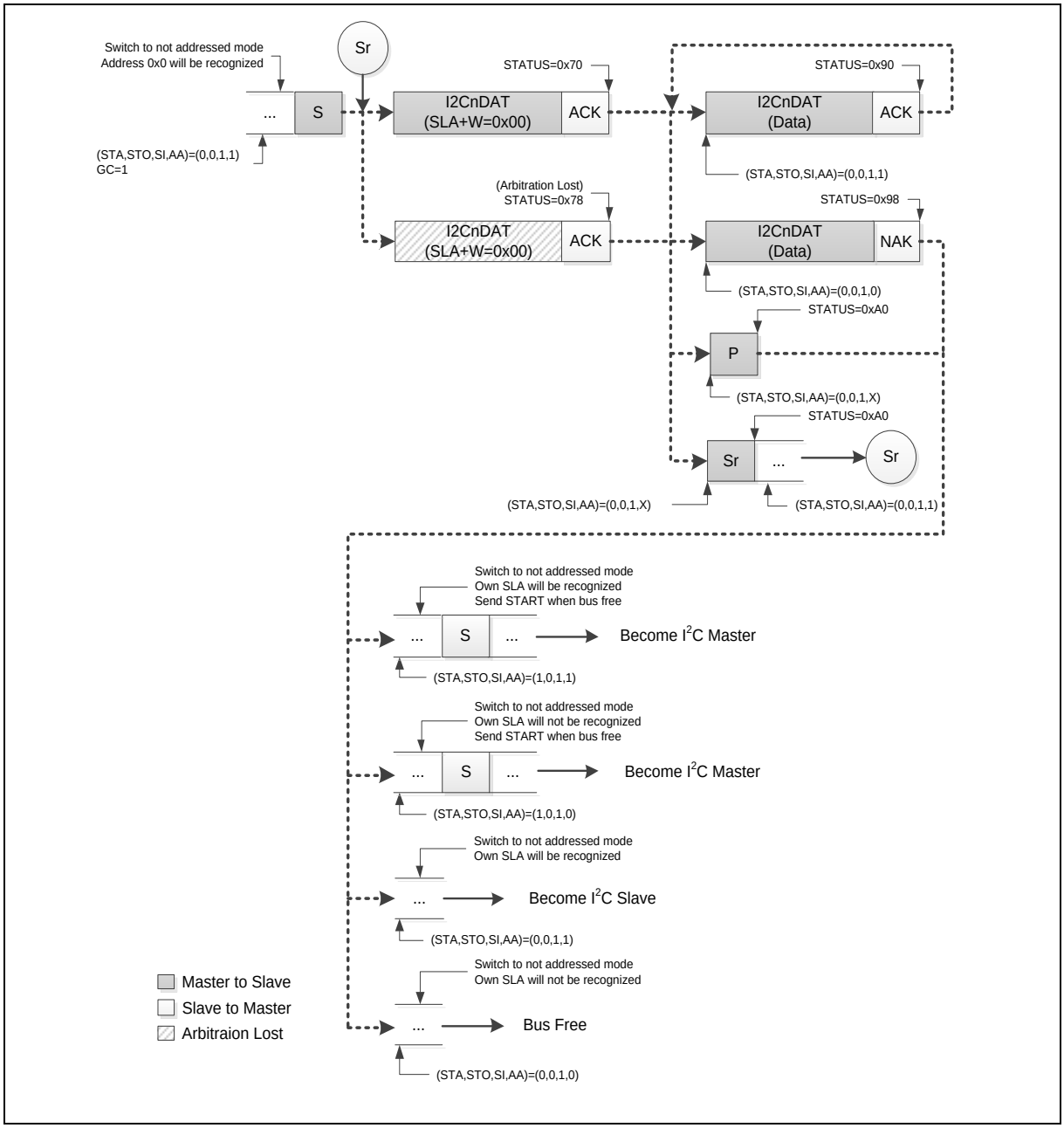


图 6.10-13 广播呼叫模式的流程和状态

6.10.2.6 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。		

6.10.2.7 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。

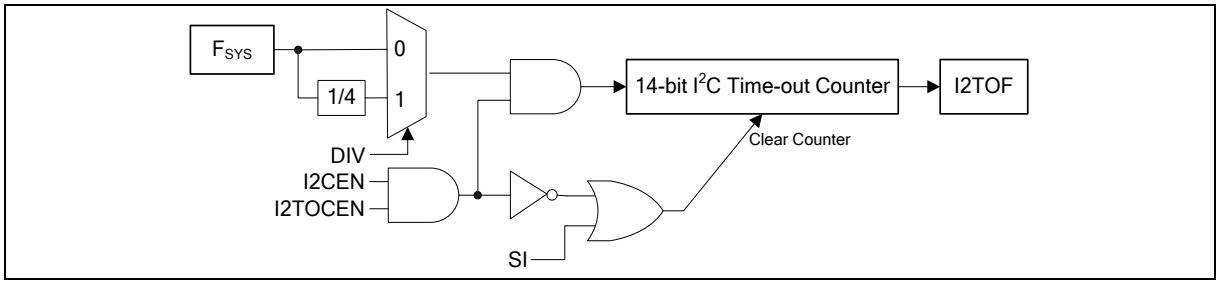


图 6.10-14 I²C 超时计数器

6.10.2.8 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。

6.10.2.9 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repeated START重复起始条件被发送*/
      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
    }
```

6.10.3 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	地址	复位值
I2CON	C0H，所有页，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**	**I²C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。
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。

位	名称	描述
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。
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。

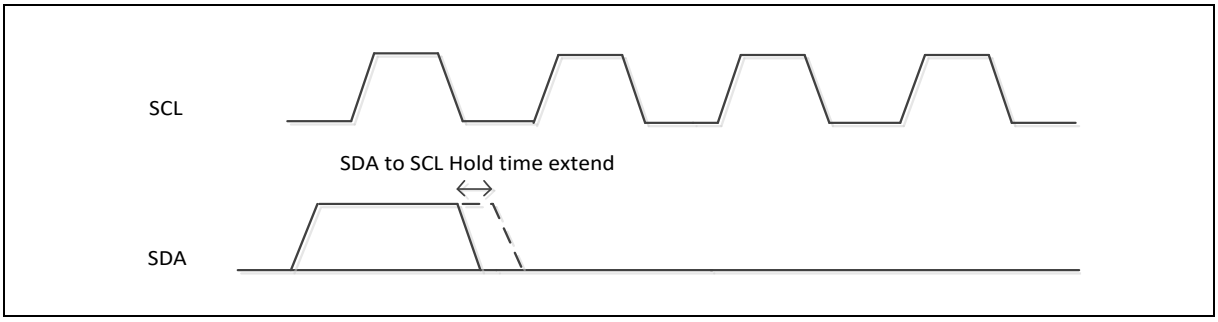


图 6.10-15 保持时间延长使能

I2STAT – I²C 状态

寄存器	地址	复位值
I2STAT	BDH，所有页	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状态字** 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**	I2STAT的最低有效三位有效位始终读到0。

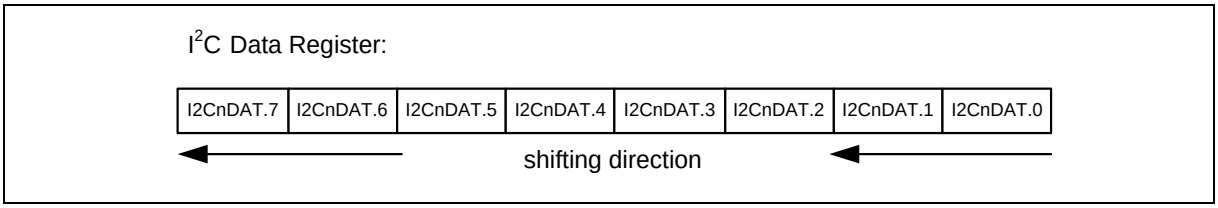
I2DAT – I²C 数据

寄存器	地址	复位值
I2DAT	BCH，所有页	0000_0000b

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。

I²C 数据移位方向。



I2ADDR – I²C自身从机地址

寄存器	地址	复位值
I2ADDR	C1H，所有页	0000_0000b

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。

I2CLK – I²C 时钟

寄存器	地址	复位值
I2CLK	BEH，所有页	0000_1001b

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

位	名称	描述
7:0	I2CLK[7:0]	**I²C时钟设定** 主机模式： 此寄存器设定作主机时I²C总线的时钟速率。算式如下： $\frac{F_{SYS}}{4 \quad (I2CLK+1)}$ 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： 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	地址	复位值
I2TOC	BFH，所有页	0000_1001 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。

6.11 脉冲宽度调制(PWM)

6.11.1 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 PWM专为电机控制而设计，提供三对PWM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。

6.11.2 功能描述

6.11.2.1 脉冲宽度调制发生器

脉冲宽度调制发生器时钟由系统时钟或定时器1计数溢出产生，可通过PWM时钟预分频调整1/1~1/128除频。PWM周期由16位周期寄存器 {PWMPH,PWMPL}组合预先设置。所有PWM通道共享同一个16位周期计数器，所以周期是相同的。每个PWM通道都有独立占空比寄存器分别为{PWM0H,PWM0L}，{PWM1H,PWM1L}，{PWM2H,PWM2L}，{PWM3H,PWM3L}，{PWM4H,PWM4L}和{PWM5H,PWM5L}。通过6个占空比寄存器，可以独立产生6个不同占空比的PWM输出。PWM信号的间隔和占空比通过16位计数器跟周期和占空比寄存器比较产生。

为了方便三相电机控制，可以通过设置GP(PWMCON1.5)来使用分组模式，使用{PWM0H,PWM0L}和{PWM1H,PWM1L}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输出一般情况下都赋予相同的占空比。当分组模式使能时，{PWM2H,PWM2L}，{PWM3H,PWM3L}，{PWM4H,PWM4L}和{PWM5H,PWM5L}寄存器无效。意味着{PWM2H,PWM2L}和{PWM4H,PWM4L}都与{PWM0H,PWM0L}相同。同时，{PWM3H,PWM3L}和{PWM5H,PWM5L}与{PWM1H,PWM1L}相同。请注意，启用PWM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。

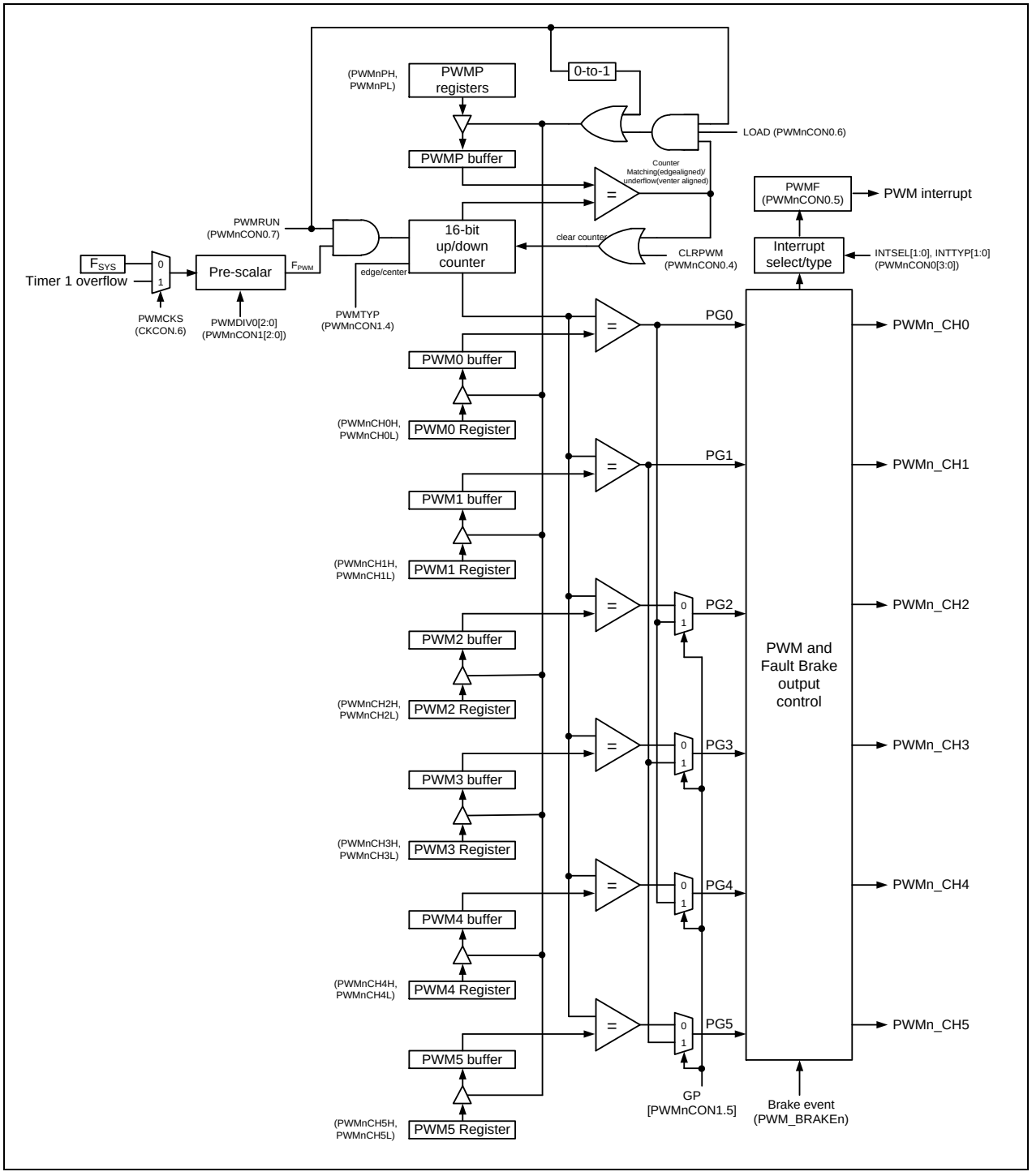


图 6.11-1 PWM 框图

PWM计数器产生六个PWM信号，分别是PG0，PG1，PG2，PG3，PG4和PG5。这些信号通过PWM及故障刹车输出控制电路，产生实际PWM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，以及PWM极性。最后一步是PWM输出或I/O功能的多路转换器。

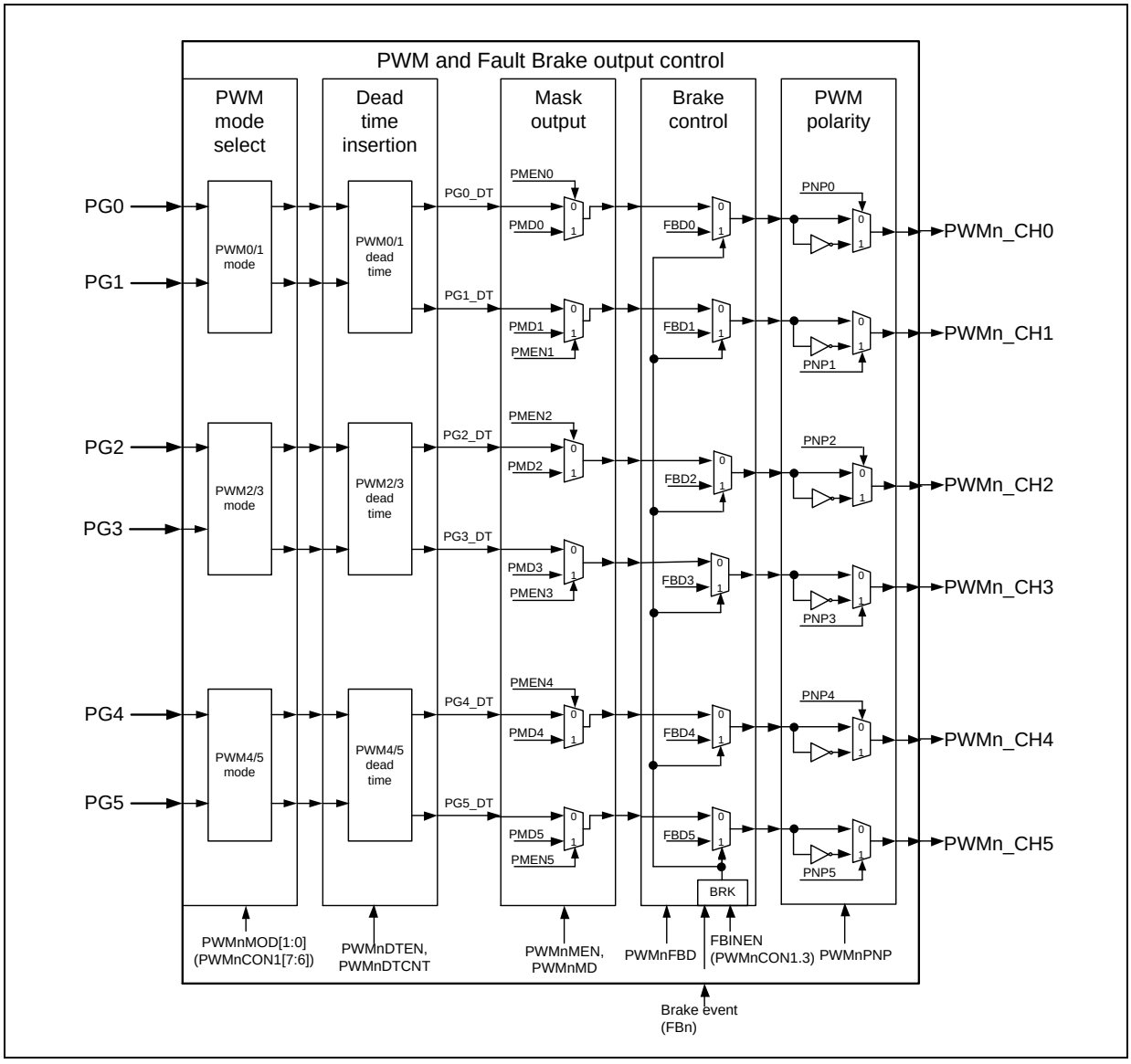


图 6.11-2 PWM 和故障刹车输出控制框图

用户应按照以下初始化PWM信号输出。第一步，设置CLRPWM(PWMCON0.4)，以确保16位向上计数器复位来达到第一个周期的准确性。初始化并设置{PWMPH,PWMPL}和所有{PWMnH,PWMnL}寄存器后，对PWMRUN(PWMCON0.7)置位逻辑1，以触发16位计数器运行。PWM开始在其输出引脚上生成波形。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，{PWMPH,PWMPL} 和所有{PWMnH,PWMnL} 寄存器可随时写入，但PWM的周期和占空比不会立即更新，直到LOAD(PWMCON0.6)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故障。

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。

6.11.2.2 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(PWMCON1.4)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{PWMPH,PWMPL}，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{PWMnH,PWMnL}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。

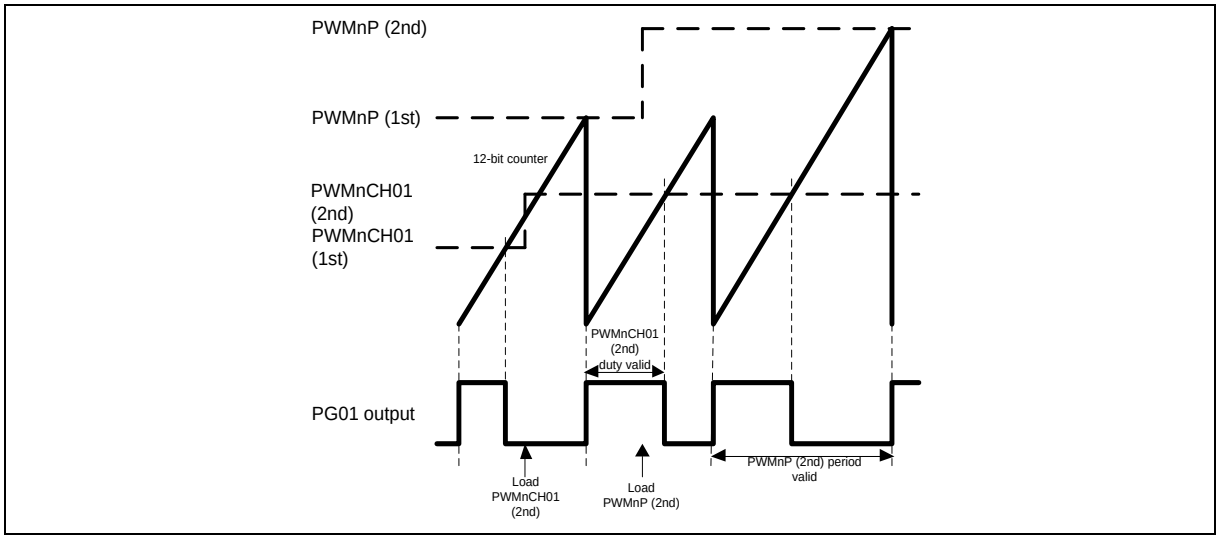


图 6.11-3 PWM 边沿对齐型波形

边沿对齐型PWM的输出频率及占空比算式如下:

PWM频率= $\frac{F_{PWM}}{\{PWMPH,PWMPL\}+1}$($F_{PWM}$为PWM时钟源除以PWMDIV)。

PWM高电平占空比= $\frac{\{PWMnH,PWMnL\}}{\{PWMPH,PWMPL\}+1}$.

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{PWMPH,PWMPL}，然后从{PWMPH,PWMPL}向下计数至0000H。PGn信号在16位计数器与占空比寄存器{PWMnH,PWMnL}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。

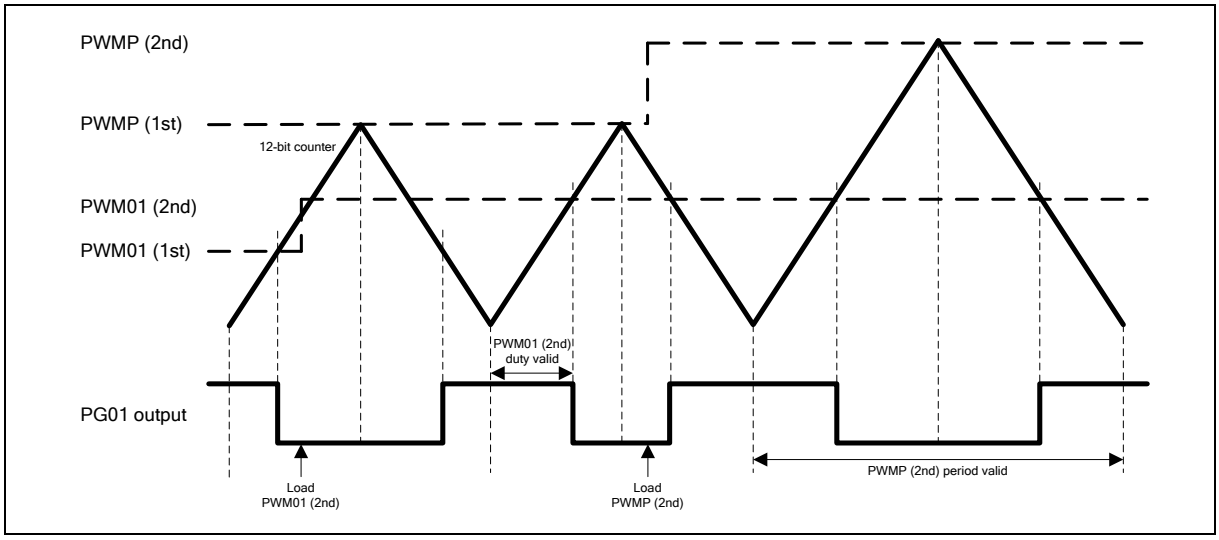


图 6.11-4 PWM 中心对齐型波形

中心对齐型PWM输出频率及占空比算式如下:

PWM频率= $\frac{F_{PWM}}{2\times\{PWMPH,PWMPL\}}$ (F_{PWM}为PWM时钟源除以 PWMDIV)。

PWM高电平占空比 = $\frac{\{PWMnH,PWMnL\}}{\{PWMPH,PWMPL\}}$.

6.11.2.3 工作模式

PGn信号经过第一级PWM及和故障刹车输出控制电路后，PWM模式选择电路将生成不同类型的PWM输出模式，总共6个通道三组 PG0~PG5信号，支持独立模式、互补模式和同步模式。

独立模式

当PWMMOD[1:0](PWMCON1[7:6])设定为[0:0]，独立模式使能。这是PWM默认模式，PG0，PG1，PG2，PG3，PG4及PG5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，PG0/2/4输出与独立模式相同的PWM信号。PG1/3/5则会相应地输出与PG0/2/4反相的PWM信号，并忽略PG1/3/5占空比寄存器{PWMnH，PWMnL}(n:1/3/5)。此模式使PG0/PG1成为一对互补的PWM输出，PG2/PG3和PG4/PG5也是如此。

在实际电机应用中，互补PWM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。CM1003每对PWM共用一个9位死区递减计数器PDTCNT，用于产生同一对中两个PWM信号之间的关闭时间。在实现上，PDTCNT定时器下溢后会在0要跳到1的信号边缘加入一段延时等待。下图展示了带死区插入的PG0/PG1对互补模式的时序图。对PG2/PG3 和 PG4/PG5有同样的死区时间电路。每对有各自的死区时间使能位(PWMnDTEN[3:0])。

请注意，PDTCNT和PDTEN 寄存器都是TA写保护。死区时间控制也仅在PWM配置为互补模式时有效。

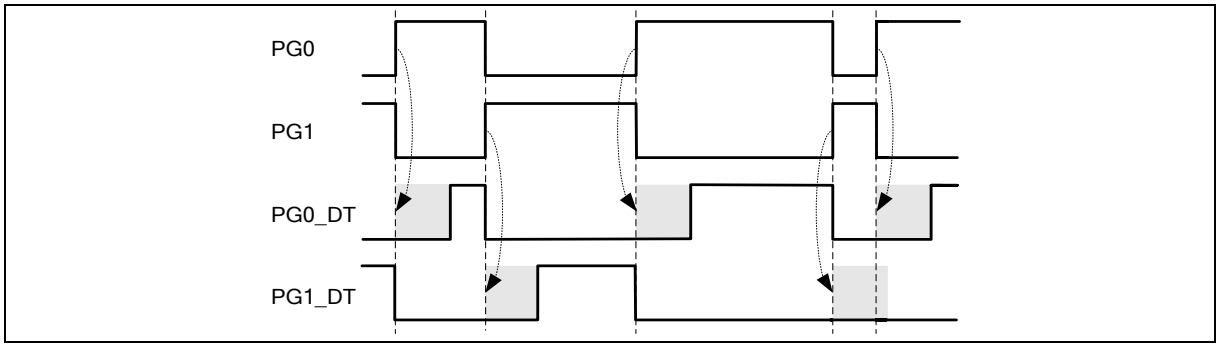


图 6.11-5 带死区时间插入的 PWM 互补模式

同步模式

当PWMMOD[1:0] = [1:0]时，同步模式使能。此模式下PG0/2/4输出与独立模式相同的PWM信号，PG1/3/5则对应输出与PG02/4相同的同相PWM信号。

6.11.2.4 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BLDC等电子换向电机时非常有用。PMEN包含6个位，用于决定屏蔽哪个PWM通道。PMD设置每个PWM通道的单独屏蔽电平。PMEN的默认值为00H，这使得所有PWM通道的输出都遵循来自PWM发生器的信号。请注意，PWM屏蔽电平是否反相取决于输出引脚上的PNP设置。

6.11.2.5 故障刹车

故障刹车功能通常与增强型PWM电路配合使用。它带故障检测输入用于保护电机系统免受损坏。当FBINEN(PWMCON1.3)置位时，故障刹车引脚输入(FB)有效。当故障刹车有效时，PWM信号将根据FBD相应位重写，PWMRUN(PWMCON0.7)将被硬件自动清除以停止PWM生成。PWM 16位计数器也将重置为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被软件清除，FBD数据输出仍会保留。用户只需再次设置PWMRUN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。故障刹车输入的极性可通过FBINLS(FBD.6)位来选择。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。请参见章节6.12.3.4“ ADC”。

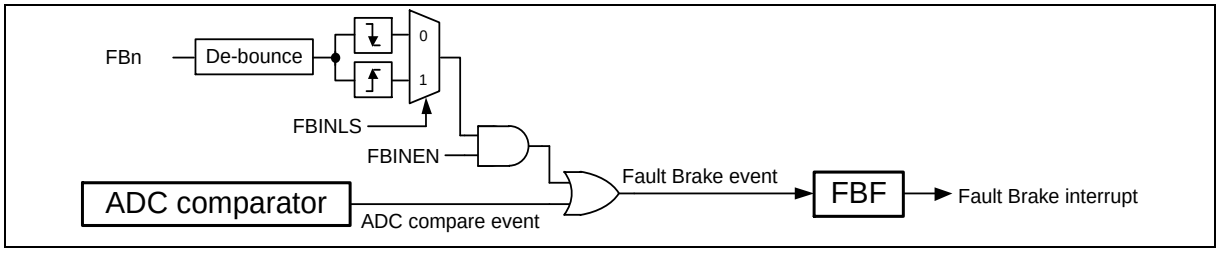


图 6.11-6 故障刹车功能框图

6.11.2.6 极性控制

每个PWM输出通道都有独立的极性控制位PNP0~PNP5。所有控制字段默认高电平有效，采用正逻辑实现。这意味着当PWM输出高电平时，电源开关打开，输出低电平时，电源开关关闭。用户可以轻松地用正逻辑配置所有设置，然后设置PNP位使PWM实际上按照负逻辑反相输出。

6.11.2.7 PWM中断

PWM模块有一个标志位PWMF(PWMCON0.5)，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](PWMINTC[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。

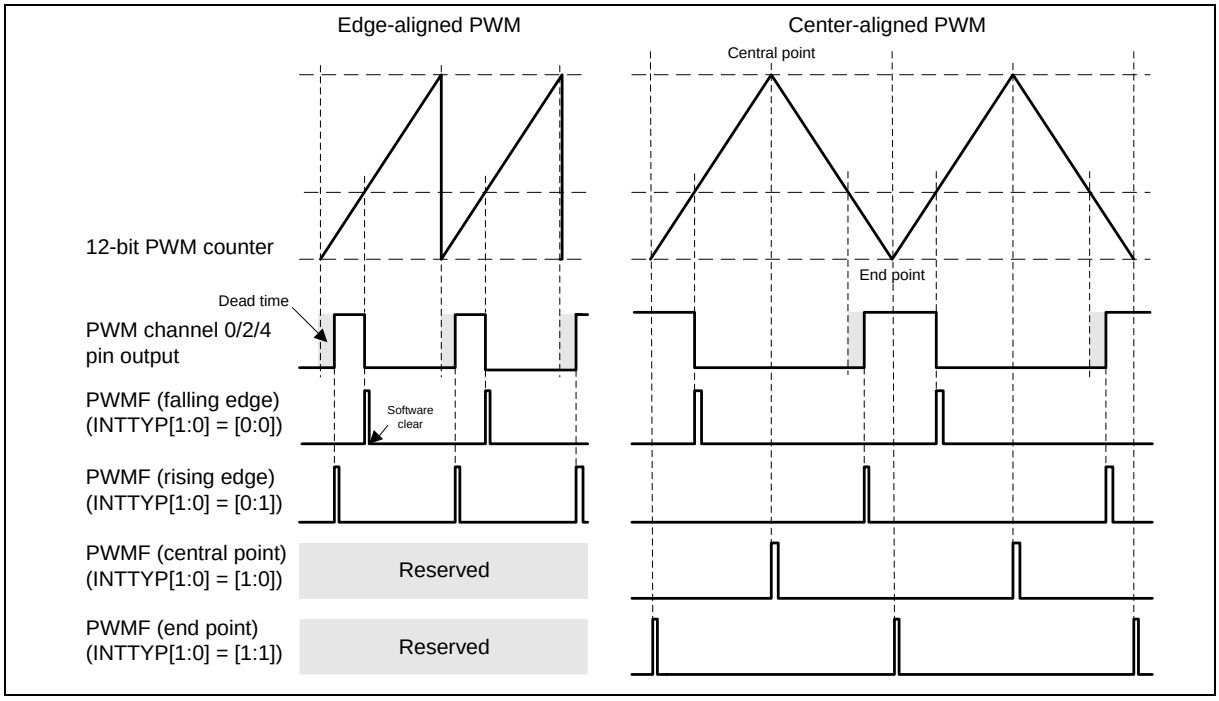


图 6.11-7 PWM 中断类型

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(FBD.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。

6.11.3 寄存器描述

PWMCON0 –PWM控制0

寄存器	地址	复位值
PWMCON0	D7H，所有页，位寻址	0000_0000b

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： 0 =PWM16 位计数器已完全清零。 1 =PWM16 位计数器尚未清零。

PWMCON1 – PWM控制1

寄存器	地址	复位值
PWMCON1	DFH，所有页	0000_0000b

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**	**群组模式使能** 此位使能PWM群组模式。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 = 群组模式禁用 1 = 群组模式使能
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(FBD.6)选择匹配的边沿信号，PWM0~5输出FBD寄存器所设置的故障刹车数据， PWMRUN(PWMCON0.7)由硬件自动清除。当PWMRUN再次置位时，PWM输出恢复。
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	地址	复位值
CKCON	8EH，所有页	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

PWMPL –PWM周期低字节

寄存器	地址	复位值
PWMPL	D9H，所有页	0000_0000b

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

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

PWMPH – PWM周期高字节

寄存器	地址	复位值
PWMPH	D1H，所有页	0000_0000b

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

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

PWMnH –PWM通道 0~5 占空比高字节 n =0,1,2,3,4,5

寄存器	地址	复位值
PWM0H	D1H，所有页	0000 _0000 b
PWM1H	D3H，所有页	0000 _0000 b
PWM2H	D4H，所有页	0000 _0000 b
PWM3H	D5H，所有页	0000 _0000 b
PWM4H	C4H，页1	0000 _0000 b
PWM5H	C5H，页1	0000 _0000 b

7	6	5	4	3	2	1	0
PWMn [15:8], n =0,1,2,3,4,5							
读/写							

位	名称	描述
7:0	**PWMn[15:8]** **n=0,1,2,3,4,5**	**通道 n占空比高字节** 此字节与 PWMnL一起控制PWM 发生器的输出信号PGn的占空比。

PWMnL –PWM通道 0~5 占空比低字节 n =0,1,2,3,4,5

寄存器	SFR 地址	描述	复位值
PWM0L	DAH，所有页	PWM Channel0Duty 低字节	0000_0000 b
PWM1L	DBH，所有页	PWM Channel1Duty 低字节	0000_0000 b
PWM2L	DCH，所有页	PWM Channel 2 Duty 低字节	0000_0000 b
PWM3L	DDH，所有页	PWM Channel 3 Duty 低字节	0000_0000 b
PWM4L	CCH, 页1	PWM Channel 4 Duty 低字节	0000_0000 b
PWM5L	CDH, 页1	PWM Channel 5 Duty 低字节	0000_0000 b

7	6	5	4	3	2	1	0
PWMn [7:0], n =0,1,2,3,4,5							
读/写							

位	名称	描述
7:0	**PWMn[7:0]** **n=0,1,2,3,4,5**	**通道n占空比低字节** 此字节与 PWMnH一起控制PWM 发生器的输出信号PGn的占空比。

PIOCON0 – PWM或 I/O选择

寄存器	地址	复位值
PIOCON0	DEH，所有页	0000_0000b

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
5	PIO05	**P0.3/PWM0 通道5引脚功能选择** 0 = P0.3/PWM0_CH5引脚用作P0.3. 1 = P0.3/PWM0_CH5引脚用作PWM0_CH5输出。
4	PIO04	**P0.1/PWM0 通道4引脚功能选择** 0 = P0.1/PWM0_CH4引脚用作P0.1. 1 = P0.1/PWM0_CH4引脚用作PWM_CH4输出。
3	PIO03	**P0.0/PWM0 通道3引脚功能选择** 0 = P0.0/PWM0_CH3引脚用作P0.0. 1 = P0.0/PWM0_CH3引脚用作PWM_CH3输出。
2	PIO02	**P1.0/PWM0 通道2引脚功能选择** 0 = P1.0/PWM0_CH2引脚用作P1.0. 1 = P1.0/PWM0_CH2引脚用作PWM_CH2输出。
1	PIO01	**P1.1/PWM0 通道1引脚功能选择** 0 = P1.1/PWM0_CH1引脚用作P1.1. 1 = P1.1/PWM0_CH1引脚用作PWM_CH1输出。
0	PIO00	**P1.2/PWM0 通道0引脚功能选择** 0 = P1.2/PWM0_CH0引脚用作P1.2. 1 = P1.2/PWM0_CH0引脚用作PWM_CH0输出。

PIOCON1 – PWM或I/O选择

寄存器	地址	复位值
PIOCON1	C6H, 页1	0000_0000b

7	6	5	4	3	2	1	0
-	-	PIO15	-	PIO13	PIO12	PIO11	-
-	-	读/写	-	读/写	读/写	读/写	-

位	名称	描述
5	**PIO15**	**P1.5/PWM0_CH5引脚功能选择** 0 = P1.5/PWM0_CH5引脚用作P1.5。 1 = P1.5/PWM0_CH5引脚用作PWM0通道5输出。
3	**PIO13**	**P0.4/PWM_CH3引脚功能选择** 0 = P0.4/PWM0_CH3引脚用作P0.4。 1 = P0.4/PWM0_CH3引脚用作PWM0通道3输出。
2	**PIO12**	**P0.5/PWM_CH2引脚功能选择** 0 = P0.5/PWM0_CH2引脚用作P0.5。 1 = P0.5/PWM0_CH2引脚用作PWM0通道2输出。
1	**PIO11**	**P1.4/PWM0_CH1引脚功能选择** 0 = P1.4/PWM0_CH1引脚用作P1.4。 1 = P1.4/PWM0_CH1引脚用作PWM0通道1输出。

PDTEN –PWM死区时间使能

寄存器	地址	复位值
PDTEN	F9H，所有页，TA保护	0000_0000b

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

位	名称	描述
4	PDTCNT.8	**PWM死区计数器第8位** 参见PDTCNT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 = GP4/GP5对信号无延时。 1 = 在GP4/GP5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 = GP2/GP3对信号无延时。 1 =在GP2/GP3对信号的上升沿插入死区时间延时。
0	PDT01EN	**PWM0_CH0/PWM0_CH1组死区时间插入使能** 此位仅当PWM0_CH0/PWM0_CH1为互补模式时有效。 0 = GP0/GP1对信号无延时。 1 = 在GP0/GP1对信号的上升沿插入死区时间延时。

PDTCNT –PWM死区时间计数器

寄存器	地址	复位值
PDTCNT	FAH，所有页，TA保护	0000_0000b

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

位	名称	描述
7:0	**PDTCNT[7:0]**	**PWM死区计数器低字节** 此8位字段与PDTEN.4 组合构成9 位PWM死区计数器PDTCNT。仅当PWM处于互补模式且PWM对应的PDTEN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PDTCNT。

PMEN – PWM掩码使能

寄存器	地址	复位值
PMEN	FBH，所有页	0000_0000b

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
n	**PMENn**	**PWMn掩码使能** 0 = PWMn信号来自其PWM发生器输出。 1 = PWMn根据PMDn设定的电位值掩码。

PMD –PWM掩码数据

寄存器	地址	复位值
PMD	FCH，所有页	0000_0000b

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
n	PMDn	**PWMn掩码数据** 一旦相应的PMENn被设置，PWMn信号输出掩码电准位值 0 = PWMn掩码，输出低电平0。 1 = PWMn掩码，输出高电平1。

FBD –PWM故障刹车数据

寄存器	地址	复位值
FBD	D7H，所有页	0000_0000b

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，故障刹车输出数据不会被释放，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**	**PWMn故障刹车数据** 0 = 当故障刹车有效后，PWMn 信号输出为0 1 = 当故障刹车有效后，PWMn 信号输出为1

PNP – PWM负极性

寄存器	地址	复位值
PNP	D6H，所有页	0000_0000b

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**	**PWMn负极性输出使能** 0 = PWMn 信号直接输出到PWMn引脚。 1 = PWMn 信号反相输出到PWMn引脚。

PWMINTC –PWM中断控制

寄存器	地址	复位值
PWMINTC	B7H, 页1	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.

6.12 12位模数转换器(ADC)

6.12.1 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位二进制数据。CM1003系列在单端模式下可选择 8通道输入。内部带隙电压0.814V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DMA(直接存储访问)功能，用于ADC连续转换并将结果数据存储到XRAM中，无需特殊启用 PDMA 模块。

6.12.2 框图

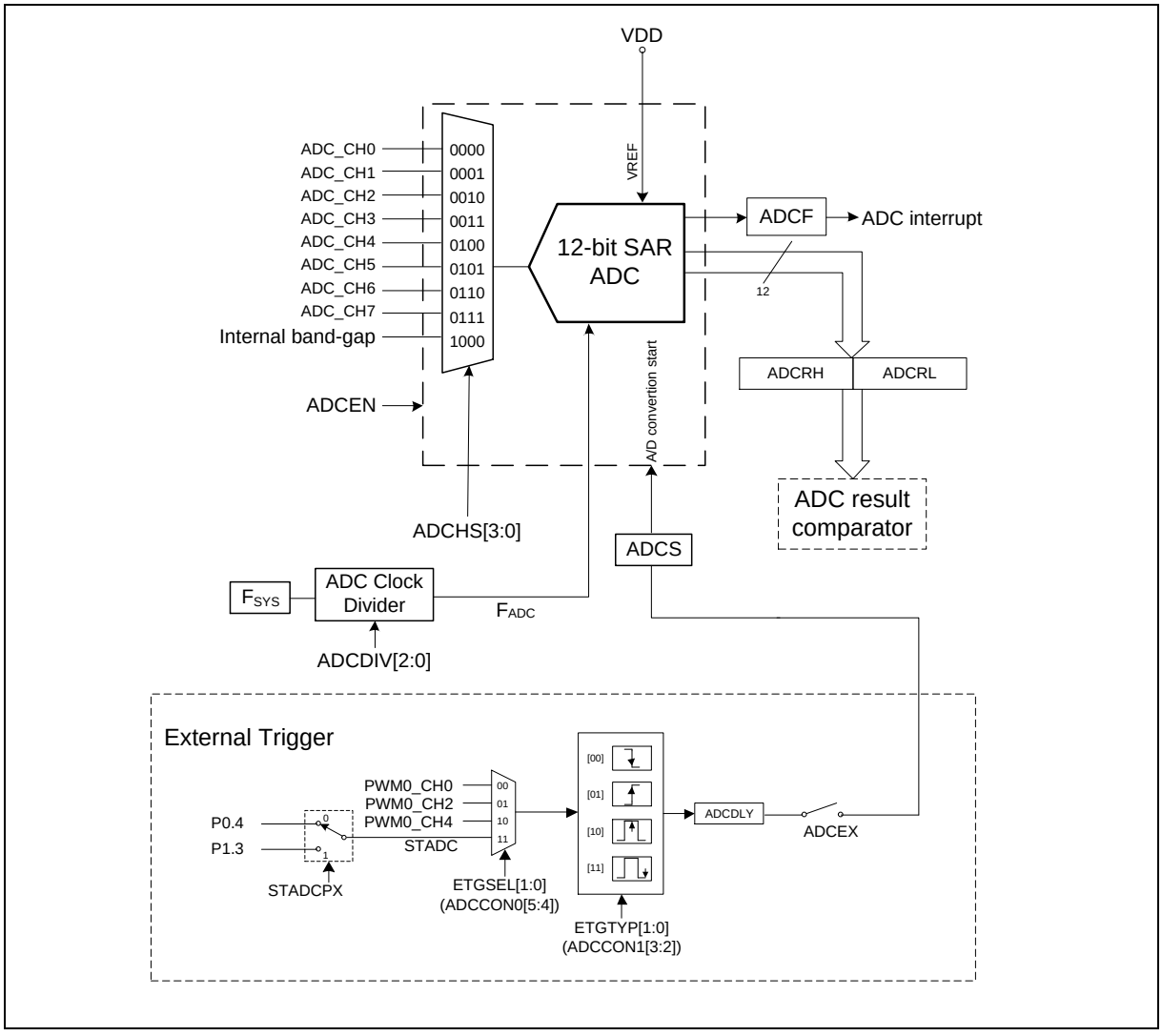


图 6.12-1 12 位 ADC 框图

6.12.3 功能描述

6.12.3.1 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AINDIDS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。

6.12.3.2 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 编码时间：$T_{ENCODING} = \frac{1}{500k}(s)$= 固定是2μs.

ADC 采样时间：$T_{SAMPLING} = \frac{4 \times ADCAQT + 6}{F_{SYS} \div ADCDIV}$

默认的最小采集时间为6个ADC时钟周期。

当F_{SYS} = 16MHz时，ADC采样时间最小值为$\frac{6}{16MHz}$ (ADCAQT = 0, ADCDIV = 0)，ADC采样时间最大值为$\frac{4*7+6}{16MHz/8}$ (ADCAQT = 7, ADCDIV = 8)

当F_{SYS} = 24MHz时，由于采样时序的限制，我们建议将ADCAQT至少设为1。ADC采样时间最小值为$\frac{4*1+6}{24MHz}$ (ADCAQT = 1, ADCDIV = 0)，ADC采样时间最大值为$\frac{4*7+6}{24MHz/8}$ (ADCAQT = 7, ADCDIV = 8)

请注意，ADC 正在转换时，ADCAQT值不能改变。

6.12.3.3 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。

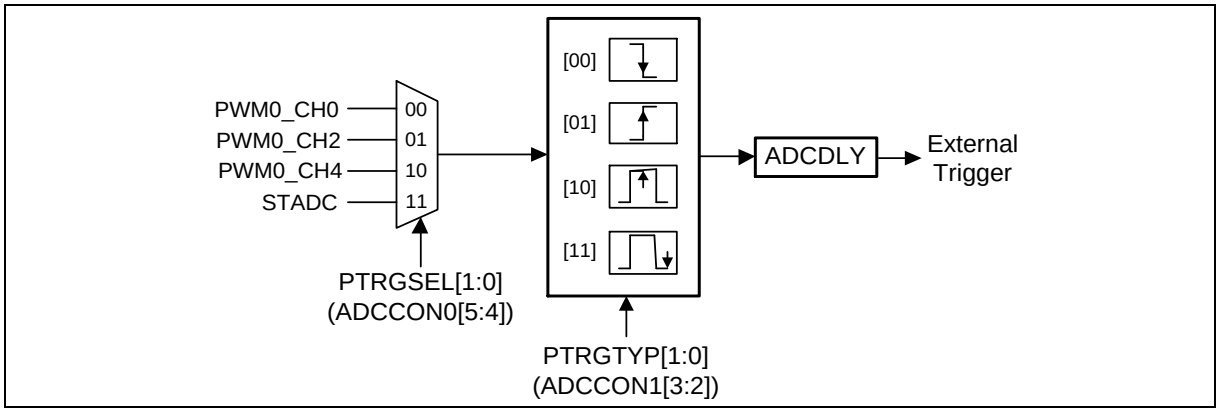


图 6.12-2 外部触发 ADC 电路

6.12.3.4 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 **錯誤! 找不到參照來源。** 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。

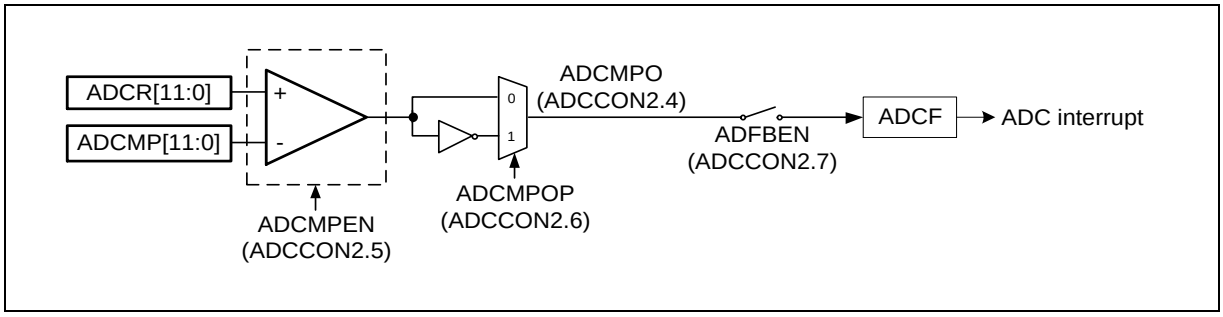


图 6.12-3 ADC 结果比较器

6.12.3.5 内部带隙电压(Band-gap)

室温下，所有CM1003的带隙电压值都会校准在1.17V至1.30V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;

– 使用下列公式换算成实际的电压值

计算公式:

$$Bandgap_Voltage = \frac{Bandgap_Value \times 3}{4}(\text{mV})$$

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：

$$Bandgap_Voltage = \frac{1614 \times 3}{4} = 1210.5(\text{mV})$$

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输入来计算V_{DD}值：

$$VDD_Voltage = \frac{Bandgap_Value \times 3072}{ADC_Convert_Bandgap_Value}(\text{mV})$$

例如:

ADC转换输入通道设置为带隙电压且转换结果为1100:

$$VDD_Voltage = \frac{1614 \times 3072}{1100} = 4496(\text{mV})$$

6.12.4 寄存器描述

ADCCON0 – ADC控制0

寄存器	地址	复位值
ADCCON0	E8H，页0，可位寻址	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标志位** **当ADCMPEN(ADCCON2[5])禁用**，A/D转换完成时，此标志位被置位；如果ADCCMPEN 被使能，当比较器值与**CM1003的ADC比较值条件匹配时，此位置位**。可以读取 ADC 结果。当此标志为1时，ADC 无法启动新的转换。此位由软件清除。
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

位	名称	描述
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.
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 = 内部带隙电压 其他= 保留.

ADCCON1 – ADC控制1

寄存器	地址	复位值
ADCCON1	E1H，页0	0000_0000b

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

位	名称	描述
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]**	**ADC时钟除频器** 00= ADC 时钟源 = F_{SYS}/1 。 01= ADC 时钟源 = F_{SYS}/2 。 10= ADC 时钟源 = F_{SYS}/4 。 11= ADC 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周期的终点。 请注意，中心点中断或周期中断仅适用于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	地址	复位值
ADCCON2	E2H,页0	0000_0000b

7	6	5	4	3	2	1	0
ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	ADCAQT[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。PWMRUN(PWMCON0.7) 也将由硬件自动清除。当PWMRUN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	**ADCAQT**	**ADC采样时间** 这3位字段决定ADC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。

AINDIDS – ADC 通道数字输入功能关闭

寄存器	地址	复位值
AINDIDS	F6H，所有页	0000_0000b

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
n	**PnDIDS**	**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。

ADCDLY – ADC外部触发延迟计数器

寄存器	地址	复位值
ADCDLY	E3H，所有页	0000_0000b

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= $\frac{ADCDLY}{F_{SYS}}$。 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。

ADCRH – ADC 转换结果高字节

寄存器	地址	复位值
ADCRH	C3H，页0	0000_0000b

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	地址	复位值
ADCRL	C2H，页0	0000_0000b

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

位	名称	描述
3:0	ADCR[3:0]	**ADC转换结果低字节** 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	地址	复位值
ADCMPH	CFH，页0	0000_0000b

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比较值的最高有效8位存储在此寄存器中。

ADCMPL – ADC 比较值低字节

寄存器	地址	复位值
ADCMPL	CEH，页0	0000_0000b

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。

6.13 指令集

6.13.1.1 指令集和寻址模式

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。

指令	CY	OV	AC	指定	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 6.13-1 影响标志位的指令

表6.13-2 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。

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。

6.13.1.2 指令集列表

指令	操作码 OPCODE	字节	时钟周期	CM1003 对比 传统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
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
DIVAB	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

指令	操作码 OPCODE	字节	时钟周期	CM1003 对比 传统80C51 速率
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
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

指令	操作码 OPCODE	字节	时钟周期	CM1003 对比 传统80C51 速率
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
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:
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 6.13-2 指令集

7 应用电路

7.1 供电方案

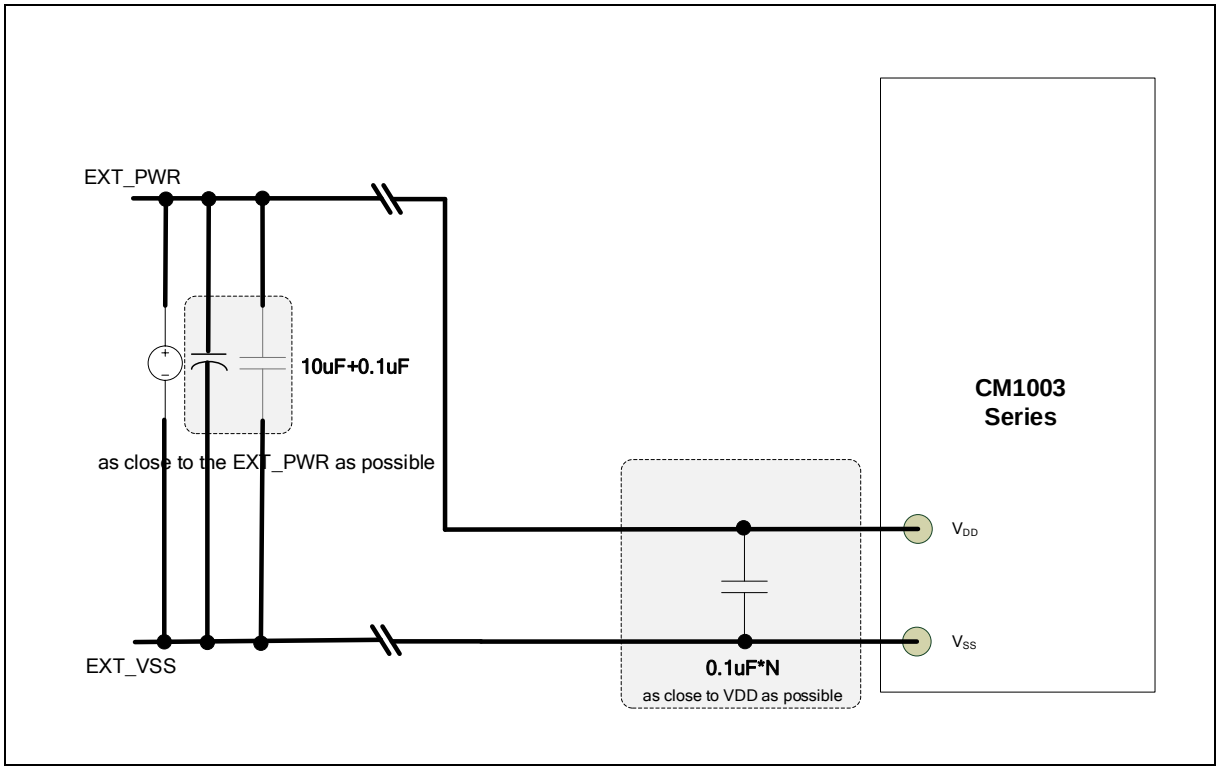
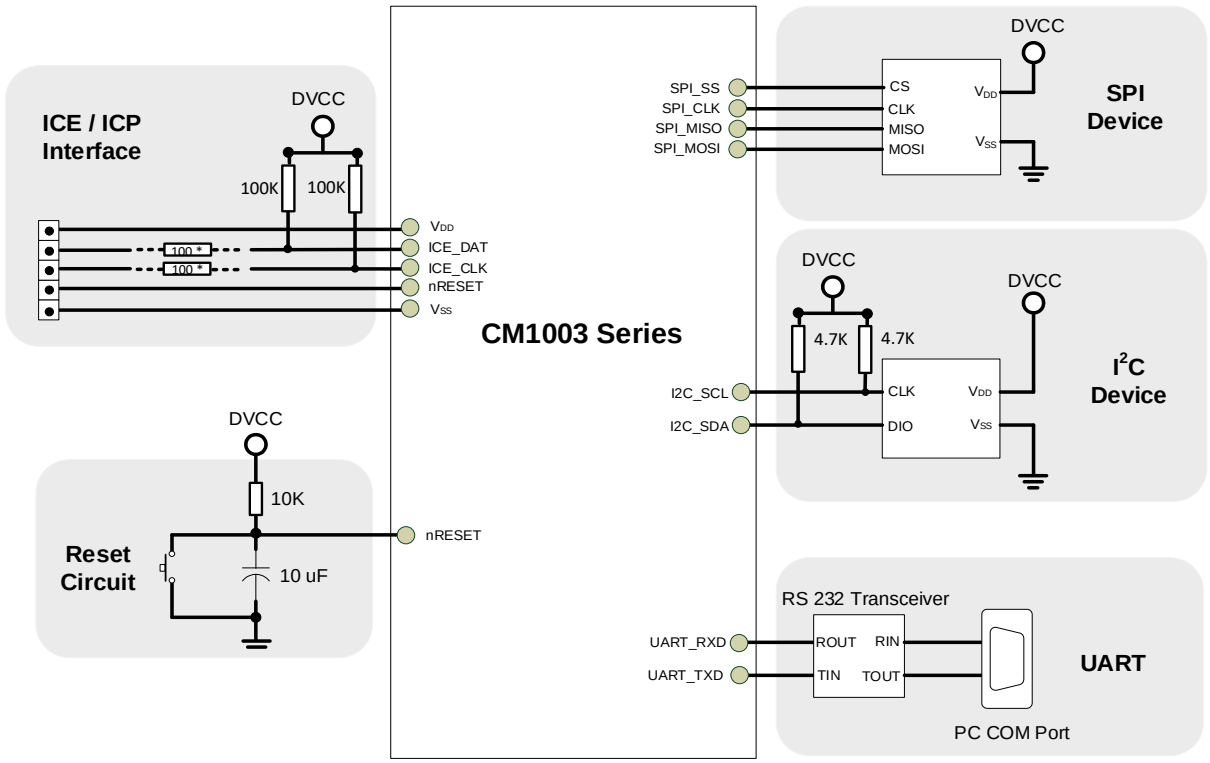


图 7.1-1 NuMicro CM1003 供电电路

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

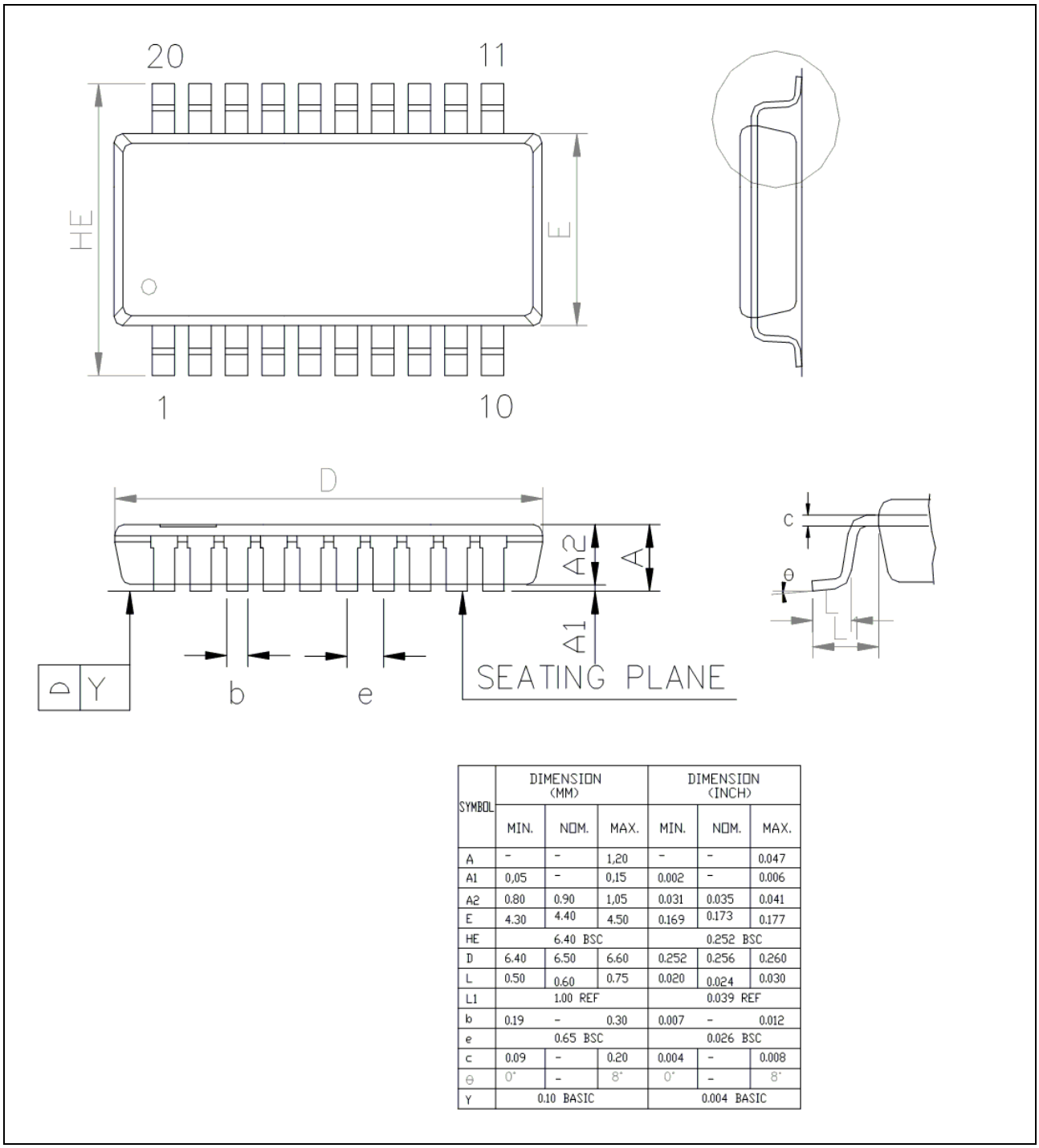
8 电气特性

CM1003电气特性的详细信息，请参阅相关的规格书。

9 封装信息

封装不含卤素，符合 RoHS 和 TSCA 标准。

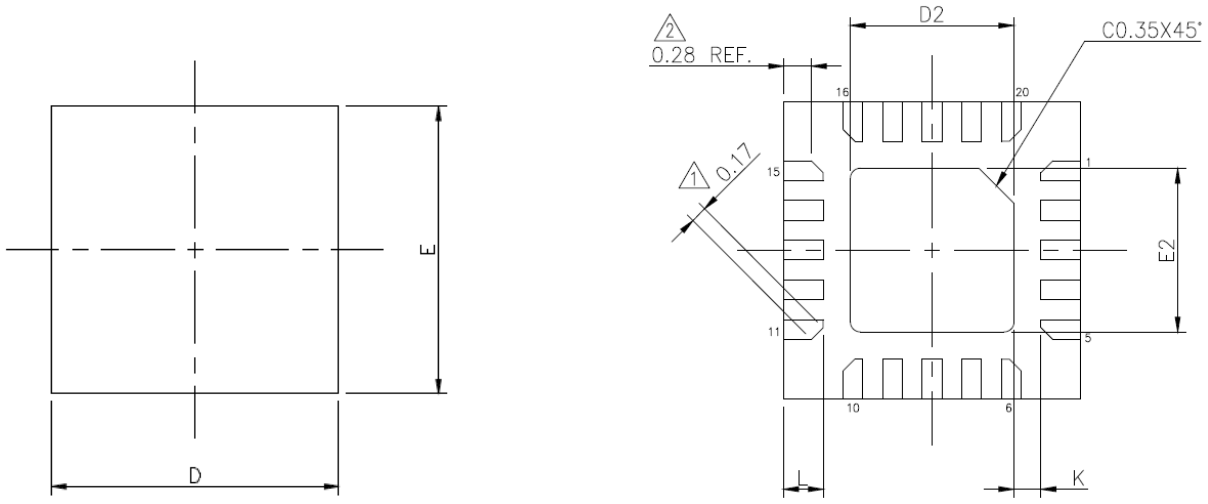
9.1 TSSOP 20 引脚(4.4 x 6.5 x 0.9 毫米)

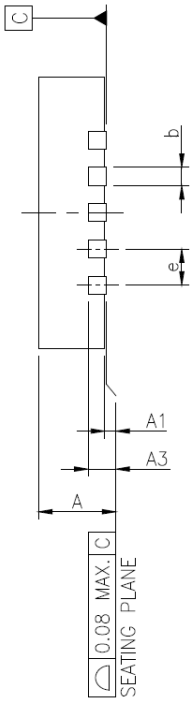


SYMBOL	DIMENSION (MM)			DIMENSION (INCH)		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	-	-	1,20	-	-	0.047
A1	0,05	-	0,15	0.002	-	0.006
A2	0.80	0.90	1,05	0.031	0.035	0.041
E	4.30	4.40	4.50	0.169	0.173	0.177
HE		6.40 BSC			0.252 BSC	
D	6.40	6.50	6.60	0.252	0.256	0.260
L	0.50	0.60	0.75	0.020	0.024	0.030
L1		1.00 REF			0.039 REF	
b	0.19	-	0.30	0.007	-	0.012
e		0.65 BSC			0.026 BSC	
c	0.09	-	0.20	0.004	-	0.008
θ	0°	-	8°	0°	-	8°
Y		0.10 BASIC			0.004 BASIC	

图 9.1-1 TSSOP-20 封装尺寸

9.2 QFN 20 引脚(3.0 x 3.0 x 0.8 毫米)





JEDEC OUTLINE	MO-220		
PKG CODE	WQFN(X319)		
SYMBOLS	MIN.	NOM.	MAX.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	0.203 REF.		
b	0.15	0.20	0.25
D	2.90	3.00	3.10
E	2.90	3.00	3.10
e	0.40 BSC		
K	0.20	—	—

PAD SIZE	D2			E2			L		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
75*X75* MIL	1.60	1.65	1.70	1.60	1.65	1.70	0.30	0.40	0.50

图 9.2-1 QFN-20 封装尺寸

10 缩写词列表

10.1 缩写词列表

Acronym	描述
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	16MHz内部高速RC振荡器 16MHz Internal High Speed RC Oscillator
IAP	应用程序内编程 In Application Programming
ICP	在线编程 In Circuit Programming
ISP	在系统编程 In System Programming
LIRC	10kHz内部低速RC振荡器 10kHz internal low speed RC oscillator(LIRC)
LVR	低电压复位 Low Voltage Reset
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	通用异步接收器/发射器 Universal Asynchronous Receiver/Transmitter
UCID	客户唯一标识符 Unique 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 历史版本信息

日期	版本	修订章节	描述
2024-05-02	1.00		初始版本。
2024-12-09	1.01		修正RAM大小的书写错误
2025-05-13	1.02		更新文法和表达
		章节 6.1.6.6 章节 6.12	更新ADC最大最小采样时间的描述 P133,P319,P324
		章节 2	更新HIRC测试温度
			将CM1003改为CM1003xFxAE
		章节 3.2,3.3	更新3.2CM1003系列命名规则和3.3CM1003系列选型指南
		章节 6.1	更新6.1存储架构
		章节 6.1.6.6 章节 6.10.3	更新I2CON第7位的名称
		章节 9.2	更新图9.2-1 QFN-20 封装尺寸
2025-10-11	1.03	章节 2	更新章节2特性中的时钟信息

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.