

1T 8051
8位微控制器

NuMicro[®] 家族

MS51系列

MS51FC0AE

MS51XC0BE

MS51EC0AE

MS51TC0AE

MS51PC0AE

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 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 10**
- 2 特性 11**
- 3 料号信息 14**
 - 3.1 封装类型 14
 - 3.2 MS51 系列选型指南 14
 - 3.3 MS51 命名规则 15
- 4 引脚配置 16**
 - 4.1 MS51 32KB 系列引脚图 16
 - 4.1.1 QFN 33封装 16
 - 4.1.2 LQFP 32封装 17
 - 4.1.3 TSSOP 28封装 18
 - 4.1.4 TSSOP 20封装 18
 - 4.1.5 QFN 20封装 19
 - 4.2 MS51 32KB系列引脚描述 20
- 5 功能框图 24**
 - 5.1 MS51 32KB 系列框图 24
- 6 功能描述 25**
 - 6.1 内存架构 25
 - 6.1.1 概述 25
 - 6.1.2 程序内存编程内存 (APROM 及 LDROM) 25
 - 6.1.3 数据存储器数据内存 (Data Flash) 27
 - 6.1.4 安全保护存储器 (SPROM) 27
 - 6.1.5 用户配置区 (CONFIG) 27
 - 6.1.6 数据存储器 (RAM) 34
 - 6.1.7 特殊功能寄存器 (SFR) 37
 - 6.2 系统管理 205
 - 6.2.1 时钟系统 205
 - 6.2.2 电源管理 212
 - 6.2.3 电源监控与复位 213
 - 6.2.4 中断系统 227
 - 6.3 内存控制器 243
 - 6.3.1 在应用编程 (IAP) 243
 - 6.3.2 在电路编程 (ICP) 258
 - 6.3.3 片上调试器 (OCD) 258
 - 6.3.4 96位序列号 260

- 6.4 通用I/O(GPIO)..... 261
 - 6.4.1 I/O模式261
 - 6.4.2 引脚中断(PIT).....263
 - 6.4.3 控制寄存器.....264
- 6.5 定时器..... 274
 - 6.5.1 定时器/计数器0/1274
 - 6.5.2 定时器2 和输入捕获283
 - 6.5.3 定时器 3299
- 6.6 脉冲宽度调制(PWM) 303
 - 6.6.1 概述303
 - 6.6.2 引脚配置.....303
 - 6.6.3 结构框图.....304
 - 6.6.4 功能描述.....305
 - 6.6.5 PWM 中断.....310
 - 6.6.6 PWM 的控制寄存器.....312
- 6.7 看门狗(WDT)..... 334
 - 6.7.1 概述334
 - 6.7.2 WDT 溢出复位功能.....335
 - 6.7.3 WDT 通用定时器功能.....335
 - 6.7.4 WDT控制寄存器.....337
- 6.8 唤醒定时器(WKT) 339
 - 6.8.1 概述339
 - 6.8.2 WKT控制寄存器339
- 6.9 串口 (UART0 和 UART1)..... 343
 - 6.9.1 概述343
 - 6.9.2 操作模式.....343
 - 6.9.3 波特率.....346
 - 6.9.4 帧错误检测.....349
 - 6.9.5 多机通信.....349
 - 6.9.6 自动地址识别.....350
 - 6.9.7 串口控制寄存器.....351
- 6.10 ISO 7816-3 接口 (SC0~2 和 UART2 ~ 4) 367
 - 6.10.1概述367
 - 6.10.2操作模式.....368
 - 6.10.3ISO 7816-3 数据传输371
 - 6.10.4初始化字符 TS371
 - 6.10.5错误信号和字符重复.....372
 - 6.10.6块保护时间和扩展保护时间.....372
 - 6.10.7智能卡控制寄存器.....373

6.11	I ² C总线接口	388
6.11.1	概述	388
6.11.2	功能描述.....	388
6.11.3	I ² C 中断服务程序典型结构范例	397
6.11.4	I ² C 超时	400
6.11.5	I ² C 中断	401
6.11.6	I ² C控制寄存器	401
6.12	串行外围总线 (SPI).....	408
6.12.1	概述	408
6.12.2	操作模式.....	410
6.12.3	时钟格式和数据传输.....	410
6.12.4	片选管脚设置.....	412
6.12.5	模式错误检测.....	412
6.12.6	写冲突错误.....	412
6.12.7	覆盖错误.....	413
6.12.8	SPI 中断	413
6.12.9	SPI控制寄存器	413
6.13	12位模数转换器 (ADC)	419
6.13.1	概述	419
6.13.2	功能描述.....	419
6.13.3	ADC 连续转换	421
6.13.4	ADC控制寄存器	422
6.14	辅助功能.....	442
6.14.1	双 DPTR.....	442
7	应用电路	446
7.1	供电电路.....	446
7.2	外围应用电路.....	447
8	电气特性	448
9	封装尺寸	449
9.1	QFN 33-管脚 (4.0 x 4.0 x 0.8 毫米)	449
9.2	LQFP 32-管脚 (7.0 x 7.0 x 1.4 毫米).....	450
9.3	TSSOP 28-管脚 (4.4 x 9.7 x 1.0 毫米)	451
9.4	TSSOP 20-管脚 (4.4 x 6.5 x 0.9 毫米)	452
9.5	QFN 20-管脚 (3.0 x 3.0 x 0.5毫米).....	453
10	名词缩写	454
10.1	名词缩写.....	454

11 指令集..... 455

 11.1 指令集以及寻址模式..... 455

 11.2 读-修改-回写指令 456

 11.3 指令集列表..... 457

12 版本历史 460

图集

图 4.1-1 LQFP-33封装引脚配置..... 16

图 4.1-2 LQFP-32 封装引脚配置..... 17

图 4.1-3 TSSOP28封装引脚配置 18

图 4.1-4 TSSOP20封装引脚配置 18

图 4.1-5 QFN20封装引脚配置..... 19

图 5.1-1 功能框图 24

图 6.1-1 MS51 程序内存映像..... 26

图 6.1-2 SPROM内存映像和安全模式..... 27

图 6.1-3 CONFIG0 复位后自动重载..... 29

图 6.1-4 CONFIG2 上电复位自动重载..... 31

图 6.1-5 数据存储地址映像 34

图 6.1-6 内部256 字节RAM 地址 35

图 6.2-1 时钟系统框图..... 205

图 6.2-2 欠压检测框图..... 216

图 6.2-3 nRESET 复位波形 220

图 6.2-4启动选择框图..... 224

图 6.4-1 准双向结构图..... 262

图 6.4-2 推挽模式结构图 262

图 6.4-3仅输入模式结构图..... 263

图 6.4-4 开漏模式 263

图 6.4-5 引脚中断框图..... 264

图 6.5-1 定时器/计数器0/1模式0..... 274

图 6.5-2 定时器/计数器0/1 模式1..... 275

图 6.5-3 定时器/计数器0/1 模式2..... 275

图 6.5-4 定时器/计数器0 模式3..... 276

图 6.5-5 定时器2 框图 284

图 6.5-6 定时器2 自动装载和输入捕获框图..... 285

图 6.5-7 定时器2 比较模式和捕获模式框图..... 286

图 6.5-8 定时器 3 框图 299

图 6.6-1 PWM0 框图 304

图 6.6-2 PWM1/2/3 框图 305

图 6.6-3 PWM0 输出电路框图..... 306

图 6.6-4 PWM1/2/3 控制框图	307
图 6.6-5 PWM 边沿对齐波形图	307
图 6.6-6 PWM 中心对齐波形	308
图 6.6-7 PWM0 互补模式和死区插入	309
图 6.6-8 紧急停止框图	309
图 6.6-9 PWM 中断	311
图 6.7-1 WDT 配置为复位功能	335
图 6.7-2 WDT做定时器的框图	336
图 6.8-1 唤醒定时器框图	339
图 6.9-1 串口模式 0 时序图	343
图 6.9-2 串口模式 1 时序图	344
图 6.9-3 串口模式2 和 3 时序图.....	345
图 6.10-1 智能卡 控制器框图.....	367
图 6.10-2 智能卡接口连接	369
图 6.10-3 智能卡激活和冷复位流程.....	369
图 6.10-4 智能卡热复位流程.....	370
图 6.10-5 智能卡释放流程	370
图 6.10-6 智能卡数据格式	371
图 6.10-7 初始化字符 TS.....	372
图 6.10-8 智能卡错误信号	372
图 6.10-9 发送方向块保护时间操作.....	373
图 6.10-10 接收方向块保护时间操作.....	373
图 6.10-11 额外保护时间操作.....	373
图 6.11-1 I ² C 总线连接	388
图 6.11-2 I ² C 总线协议	388
图 6.11-3 启动、重复启动、停止条件	389
图 6.11-4 主机发送数据给一个7位地址从机.....	389
图 6.11-5 主机从一个7位地址从机读取数据.....	390
图 6.11-6 一次传输中的I2C数据格式.....	390
图 6.11-7 应答位.....	391
图 6.11-8 两台主机仲裁过程.....	391
图 6.11-9 根据当前I ² C 状态来控制I ² C总线.....	392
图 6.11-10 主机发送模式流程和寄存器状态.....	393

图 6.11-11 主机接收模式流程和寄存器状态..... 394

图 6.11-12 从机接收模式流程和状态..... 396

图 6.11-13 广播呼叫模式流程和状态..... 396

图 6.11-14 I²C 超时计数器 401

图 6.11-15 保持时间延长使能..... 402

图 6.12-1 SPI 框图 408

图 6.12-2 SPI多主机，多从机连接图..... 409

图 6.12-3 SPI 单主机单从机连接图 409

图 6.12-4 SPI 时钟格式 410

图 6.12-5 SPI CPHA = 0时时钟和数据格式..... 411

图 6.12-6 SPI CPHA = 1 时时钟和数据格式 412

图 6.12-7 SPI 覆盖波形 413

图 6.12-8 SPI 中断请求 413

图 6.13-1 12位ADC框图..... 419

图 6.13-2 外部触发 ADC 电路 421

图 6.13-3 ADC 结果比较器..... 421

图 6.13-4 ADC连续转换模式，包含 DMA 422

图 7.1-1 NuMicro[®] MS51 供电电路 446

图 7.2-1 NuMicro[®] MS51外围接口电路 447

图 9.1-1 QFN-33 封装尺寸 449

图 9.2-1 LQFP-32 封装尺寸 450

图 9.3-1 TSSOP-28 封装尺寸..... 451

图 9.4-1 TSSOP-20 封装尺寸..... 452

图 9.5-1 QFN-20 封装尺寸 453

表集

表 6.1-1 特殊功能寄存器地址映像 40

表 6.1-2 特殊功能寄存器定义和复位值 48

表 6.1-3 影响标志位设置的指令 135

表 6.2-1 BOF 复位值 217

表 6.2-2 最小欠压检测脉冲宽度 220

表 6.2-3 中断函数首地址表 227

表 6.2-4 中断优先级设置 231

表 6.2-5 各级中断源特性表 232

表 6.3-1 IAP 模式及命令 244

表 6.4-1 I/O 模式配置 261

表 6.6-1 PWM 引脚定义和使能寄存器表 303

表 6.7-1 WDT 定时时间表 334

表 6.9-1 串口 UART0 模式/波特率描述 347

表 6.9-2 串口 UART1 模式/波特率描述 347

表 6.10-1 智能卡或 UART 管脚定义以及使能控制寄存器 368

表 6.12-1 片选管脚配置 416

表 10.1-1 名词缩写列表 454

表 11.1-1 指令集寻址方式 455

表 11.1-2 指令影响标志位结果 455

表 11.3-1 指令集 459

1 概述

MS51为带有flash的增强型8位8051内核微控制器（1T工作模式），指令集与标准的80C51完全兼容并具备更高效能。

MS51系列内嵌32K字节的Flash存储区，通常称作APROM，用于存放用户程序代码。该Flash存储区支持在应用编程（IAP）功能，即可通过片内固件更新程序代码。IAP功能同时提供用户可自行配置程序区域或数据存储区。IAP功能可以对数据存储区进行读写操作，同时读数据也可以通过MOVC指令来实现。MS51有一个额外的存储区称作LDR0M，该区域通常存放用于执行在系统编程（ISP）的引导代码（boot code），LDR0M的大小最多可配置到4K字节。为了方便烧写和校验，整个flash区域支持并行烧录和ICP烧录。可通过加密位对Flash加密，保障程序代码无法被读出。MS51 32K系列内嵌256字节SRAM，2K字节XRAM。

MS51 32KB系列提供丰富的特殊功能模块，包括：256字节SRAM，2K字节XRAM。最多可达29个标准管脚。两组标准16位定时器/计数器：定时器0及1，一组带有3路管脚输入捕获模式的16位定时器：定时器2，一组看门狗定时器（WDT），一组自唤醒定时器（WKT），一组带自动重装载功能，可用于产生标准波特率的定时器：定时器3。两组标准串行口（UART），这两组串行口具有帧错误侦测及自动地址识别功能。一组SPI，一组I²C，3组ISO 7816-3标准界面，6通道标准型PWM输出，6路增强型PWM输出，16通道12位ADC。上述功能对应产生24个中断源，具有4级中断优先级配置。

MS51 32KB系列支持3组时钟源输入，所有时钟源支持软件切换立即生效功能（on-the-fly）。3组时钟源包括：外部时钟，10kHz内部RC振荡时钟和一个出厂时已校准到室温下精度达±1%的16MHz内部高速时钟。MS51提供额外的电源监控管理模块，例如上电复位和4级低电压检测，该模块用于保障芯片在上电及掉电时系统稳定工作。

MS51可运行在两种低功耗模式—空闲模式和掉电模式，可通过软件选择运行在哪种模式。空闲模式时，芯片主时钟关闭，但部分功能模块仍然运行。掉电模式下芯片全部时钟关闭确保芯片功耗达到最低。在正常工作模式下，也可选择主时钟除频方式工作，确保在功耗和性能之间灵活运用。高效能、丰富的功能模块及配置，MS51可灵活用于各种应用场合，家电产品，甚至是马达控制等高端需求控制系统。

2 特性

核心和系统

8051	<ul style="list-style-type: none"> 全静态8位1T 8051内核CMOS微控制器 指令集完全兼容 MCS-51 4级优先级中断配置 双数据指针(DPTRs)
上电复位 (POR)	<ul style="list-style-type: none"> POR 电压门限1.15V
欠压检测 (BOD)	<ul style="list-style-type: none"> 4级电压选择支持欠压中断和复位功能 (4.4V / 3.7V / 2.7V / 2.2V)
低电压复位 (LVR)	<ul style="list-style-type: none"> 电压门限 2.0V
安全	<ul style="list-style-type: none"> 96位唯一序列号(UID) 128位用户码(UCID) 128字节加密保护代码空间

存储

Flash	<ul style="list-style-type: none"> 32 KB应用代码空间(APROM) . 4/3/2/1 KB 加载代码空间(LDROM), 与APROM共享Flash空间, 可利用ISP来引导代码来更新APROM 支持内嵌内存128字节页擦除, 可作为Data Flash使用 128字节加密保护代码空间 通过安全加锁位保护整块内存不被外部读取 支持系统编程(ISP), 在应用编程(IAP) 更新内嵌内存 支持通过SWD/ICE接口的2线ICP更新
SRAM	<ul style="list-style-type: none"> 内嵌256KB SRAM 内嵌额外 2 KB字节 RAM (XRAM)通过MOVX 指令读写

时钟

外部时钟源	<ul style="list-style-type: none"> 4~24 MHz 外部高速晶振 (HXT) 为精确的时间操作
内置时钟源	<ul style="list-style-type: none"> 默认16 MHz高速内部RC振荡器 (HIRC) 校正精度±1% (25 °C, 3.3 V), ±2% (-20~105°C. 2.4 ~ 5.5 V) 高速内部RC振荡器可配置为24 MHz 10kHz低速内部振荡器(LIRC) ±1%精度等级 (25 °C, 3.3 V)

定时器

<p>16位定时器</p>	<ul style="list-style-type: none"> • 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个输入管脚可供选择 • 一组16位自动重装载功能定时器3，可用于配置串行口UART的波特率
<p>看门狗</p>	<ul style="list-style-type: none"> • 一组6位看门狗定时器(WDT) • 溢出时间间隔可选择 6.40 ms ~ 1.638s 由内部10 kHz独立时钟作为时钟源 • 可用作掉电模式或者空闲模式下的唤醒 • 看门狗溢出可触发中断或复位芯片
<p>WKT唤醒定时器</p>	<ul style="list-style-type: none"> • 一组16位上计数自唤醒功能定时器(WKT) • 内部10 kHz独立时钟作为时钟源 • 用于低功耗模式下自主唤醒 • 溢出可触发中断
<p>PWM</p>	<ul style="list-style-type: none"> • 最多可至12路输出选择 • 支持最大时钟源频率达 24 MHz • 四个 PWM 模块, 12路PWM输出 • 支持每路独立输出模式输出 • 支持3对互补模式PWM输出 • 8位分辨率的死区插入 • 最大16位分辨率的周期 • 每路PWM管脚支持掩码功能屏蔽输出功能和三态使能 • 支持刹车功能 • ADC比较事件产生刹车
<p>模拟接口</p>	
<p>模数转换 (ADC)</p>	<ul style="list-style-type: none"> • 模拟输入电压范围: 0 ~ AV_{DD}. • 12位分辨率和10位精度保证. • 单端输入模式 16 个信道 • 1 个内部通道, 带隙电压 (V_{BG}). • 最快ADC 转换频率1 MHz. • 500 KSPS 采样率 • 软件写 1 ADCS 位触发转换 • 外部管脚 (STADC) 触发转换 • PWM 触发转换
<p>通信接口</p>	
<p>UART</p>	<ul style="list-style-type: none"> • 支持2路标准UART: UART0, UART1

	<ul style="list-style-type: none"> • UART波特率时钟可选HIRC或HXT • 全双工异步通信 • 可发送或接收第9位. • 可通过软件配置TXD与RXD管脚对换位置
I ² C	<ul style="list-style-type: none"> • 1 组I²C 设备 • 主机/从机 模式 • 主从机双向传输模式 • 7位地址模式 • 标准模式 (100 kbps快速模式 (400 kbps) 和快速加模式 (1 Mbps) • 内建8位溢出定时器, 当I²C总线中止且定时器溢出时, 产生I²C中断 • 支持保持时间可编程
SPI	<ul style="list-style-type: none"> • 支持一组 SPI控制器 • 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
ISO-7816	<ul style="list-style-type: none"> • 支持三组ISO 7816-3 • 支持ISO 7816-3 协议 T=0, T=1 • 支持配置为全双工UART 模式.
GPIO	<ul style="list-style-type: none"> • 4种 I/O 模式: <ul style="list-style-type: none"> - 准双向模式 - 推挽输出模式 - 开漏模式 - 高阻态输入 • 施密特触发输入 / TTL 模式可选 • 每个GPIO均可配置为边沿 或电平触发的中断源 • 支持标准外部中断 $\overline{INT0}$ 及 $\overline{INT1}$. • 支持高驱动力和高翻转速率的 I/O • I/O 管脚可配置内部上拉 / 下拉功能 • 最大 I/O 速度为 24 MHz • 使能引脚中断功能将使能唤醒功能
ESD & EFT	
ESD	<ul style="list-style-type: none"> • HBM 达 8 kV
EFT	<ul style="list-style-type: none"> • > ± 4.4 kV
Latch-up	<ul style="list-style-type: none"> • 达150 mA

3 料号信息

3.1 封装类型

	MSOP10	TSSOP14	TSSOP20	QFN20	TSSOP28	LQFP32	QFN33
料号	MS51BA9AE	MS51DA9AE	MS51FB9AE MS51FC0AE	MS51XB9AE MS51XB9BE MS51XC0BE	MS51EC0AE	MS51PC0AE	MS51TC0AE

3.2 MS51 系列选型指南

料号	Flash (KB)	SRAM (KB)	LDROM (KB) [1]	I/O	Timer	PWM	通讯接口				ADC(12-Bit)	封装
							ISO 7816-3 [2]	UART	SPI	I2C		
MS51BA9AE	8	1	4	8	4	5	-	2	1	1	5-ch	MSOP10
MS51DA9AE	8	1	4	12	4	5	-	2	1	1	7-ch	TSSOP14
MS51XB9AE	16	1	4	18	4	6	-	2	1	1	8-ch	QFN20 [3]
MS51XB9BE	16	1	4	18	4	6	-	2	1	1	8-ch	QFN20 [3]
MS51FB9AE	16	1	4	18	4	6	-	2	1	1	8-ch	TSSOP20
MS51FC0AE	32	2	4	18	4	8	2	2	1	1	10-ch	TSSOP20
MS51XC0BE	32	2	4	18	4	8	2	2	1	1	10-ch	QFN20
MS51EC0AE	32	2	4	26	4	10	3	2	1	1	15-ch	TSSOP28
MS51TC0AE	32	2	4	30	4	12	3	2	2	1	15-ch	QFN33
MS51PC0AE	32	2	4	30	4	12	3	2	2	1	15-ch	LQFP32

注:

- LDROM 由 APROM 独立出 4/3/2/1KB Flash 区域, 可用于进行 ISP 动作。
- ISO 7816-3 可配置为 UART2~4。
- 封装具体尺寸请看章节9。
- 本技术参考手册仅针对 32KB 系列进行具体描述。

3.3 MS51 命名规则

MS	51	F	B	9	A	E
内核	产品线	封装	Flash	SRAM	保留	温度范围
1T 8051 Industry	51: Base	B: MSOP10 (3x3 mm) D: TSSOP14 (4.4x5.0 mm) E: TSSOP28 (4.4x9.7 mm) F: TSSOP20 (4.4x6.5 mm) O: SOP20 (300 mil) P: LQFP32 (7x7 mm) T: QFN33 (4x4 mm) U: SOP28 (300 mil) X: QFN20 (3x3mm)	A: 8 KB B: 16 KB C: 32 KB	0: 2 KB 1: 4 KB 2: 8/12 KB 3: 16 KB 6: 32 KB 8: 64 KB 9: 1 KB A: 96 KB		E: -40 ~ 105° C

4 引脚配置

用户可以在第四章找到引脚的配置信息或使用 **NuTool - PinConfigure**。NuTool – PinConfigure 包含所有 NuMicro® 家族芯片系列的所有型号，帮助客户正确方便的配置GPIO的多功能引脚。

4.1 MS51 32KB 系列引脚图

4.1.1 QFN 33封装

相关型号：MS51TC0AE

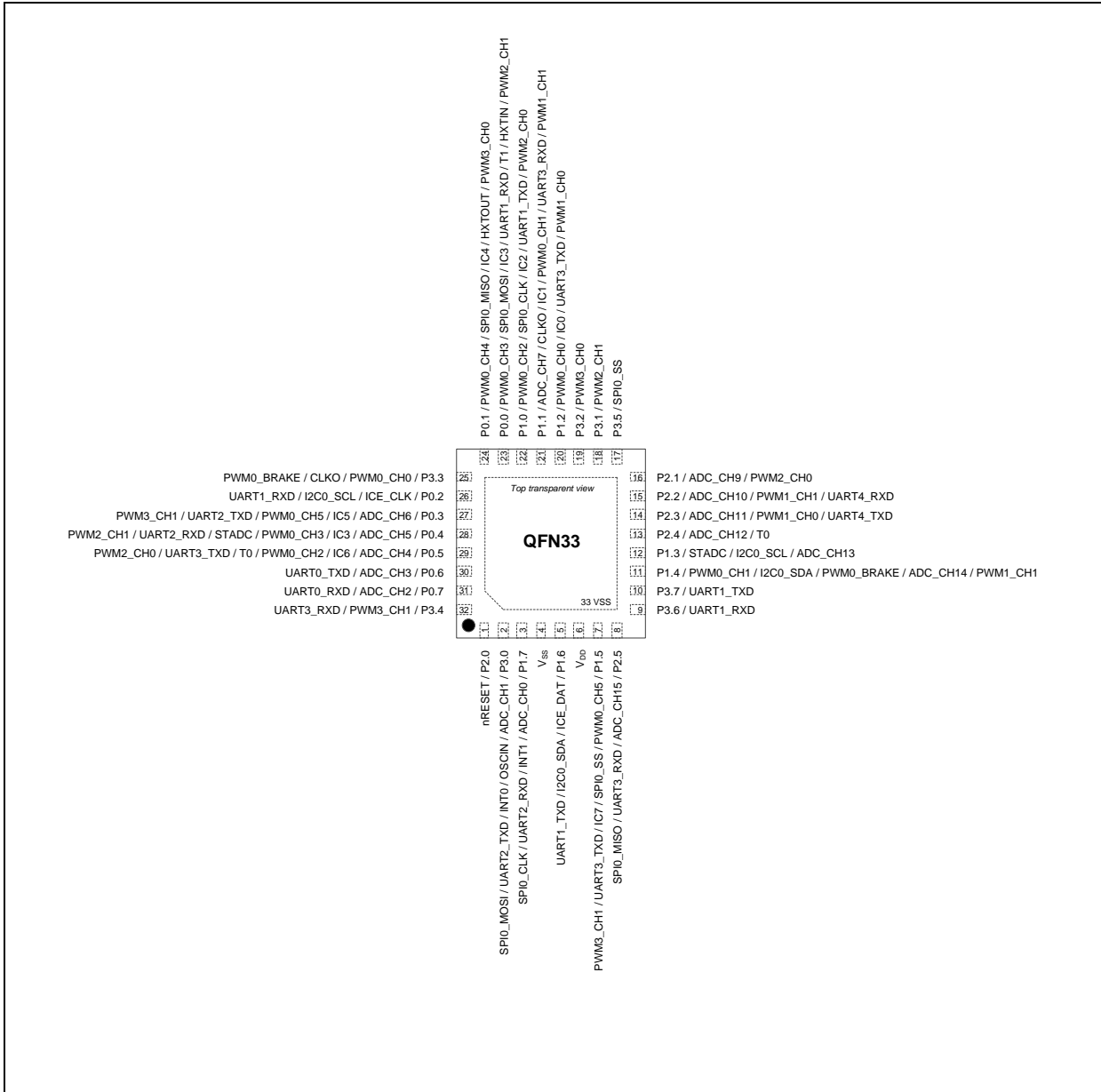


图 4.1-1 LQFP-33 封装引脚配置

4.1.2 LQFP 32封装

相关型号：MS51PC0AE

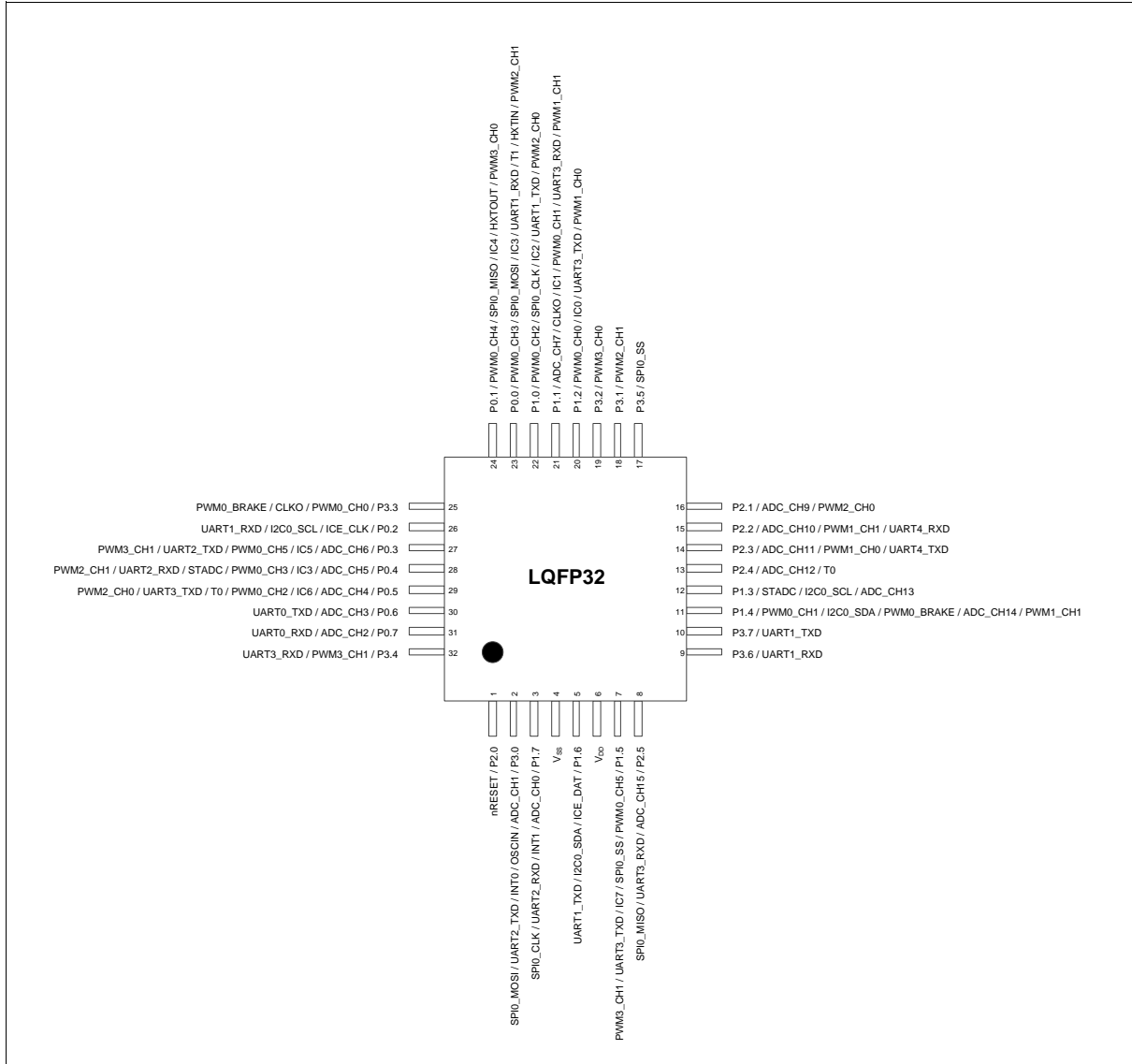


图 4.1- 2 LQFP-32 封装引脚配置

4.1.3 TSSOP 28封装

相关型号：MS51EC0AE

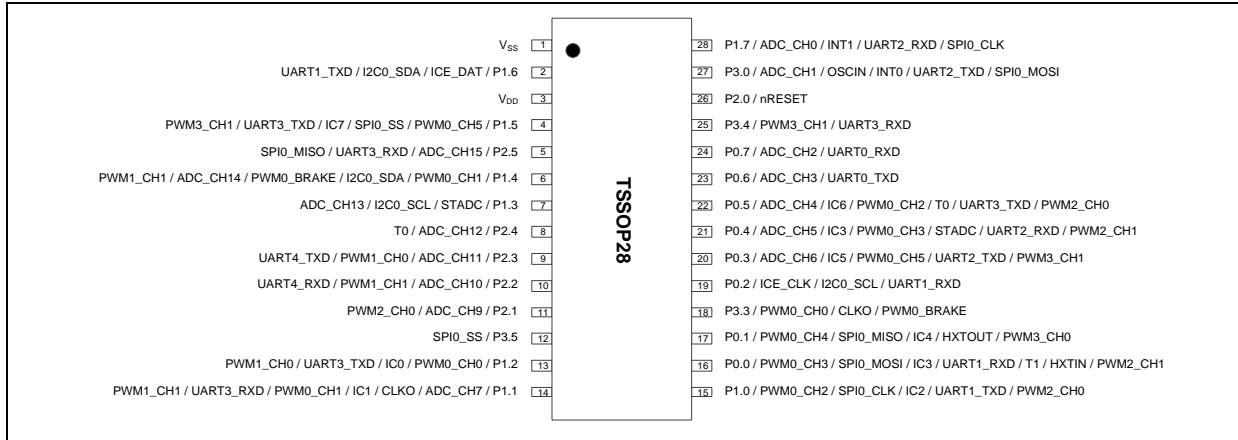


图 4.1- 3 TSSOP28 封装引脚配置

4.1.4 TSSOP 20封装

相关型号：MS51FC0AE

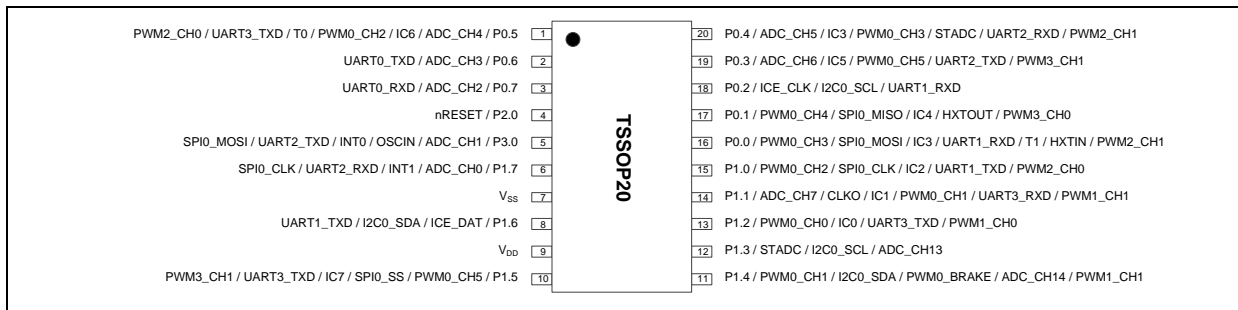


图 4.1- 4 TSSOP20 封装引脚配置

4.1.5 QFN 20封装

相关型号：MS51XC0BE

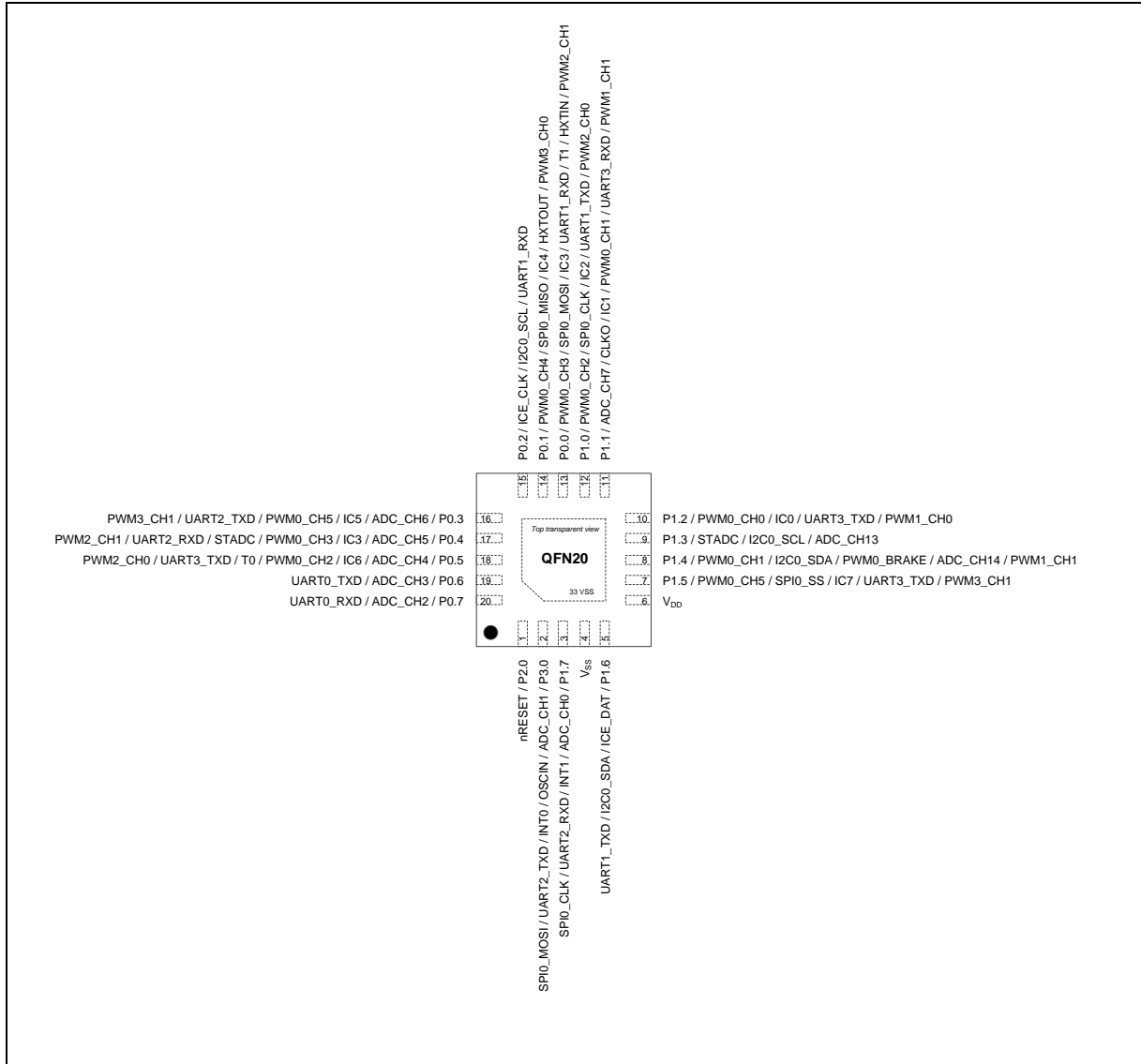


图 4.1- 5 QFN20 封装引脚配置

4.2 MS51 32KB系列引脚描述

管脚号				管脚名称	管脚描述 ^[1]
MS51XC0BE QFN 20	MS51FC0AE TSSOP20	MS51EC0AE TSSOP28	MS51PC0AE LQFP 32 MS51TC0AE QFN 33		
6	9	3	6	VDD	数字电源管脚.电源: 电源正端
4	7	1	4	VSS	数字地管脚 电源负端
13	16	16	23	P0.0	通用数字输入/输出管脚 P0.0: 端口0管脚0
				PWM0_CH3	PWM0通道3输出管脚.
				PWM2_CH1	PWM2通道1输出管脚.
				IC3	输入捕获通道 3管脚. 定时器输入捕获通道3
				SPI0_MOSI	SPI 主机输出/从机输入管脚.
				UART1_RX	UART1 输入管脚
				XT1_IN	外部4~24 MHz 晶振输入管脚.
				OSCIN	当 EXTEN[1:0] = 10b, 作为外部时钟输入管脚
14	17	17	24	T1	定时器/计数器1, 外部计数输入或翻转输出管脚
				P0.1	通用数字输入/输出管脚P0.1: 端口0管脚1
				PWM0_CH4	PWM0 输出通道 4管脚.
				PWM3_CH0	PWM3 输出通道 0 管脚.
				IC4	输入捕获通道4管脚.
				SPI0_MISO	SPI 主机输入/从机输出管脚
15	18	19	26	XT1_OUT	外部 4~24 MHz (高速) 晶振输出管脚
				P0.2	通用数字输入/输出管脚P0.2: 端口0管脚2
				I2C0_SCL	I ² C 时钟管脚
				UART1_RXD	UART1 数据接收管脚
16	19	20	27	ICE_CLK	ICE / ICP 时钟输入管脚
				P0.3	通用数字输入/输出管脚P0.3: 端口0管脚3
				ADC_CH6	ADC输出通道 6 管脚.
				PWM0_CH5	PWM0 输出通道5管脚
				PWM3_CH1	PWM3 输出通道1管脚
				IC5	输出捕获通道 5管脚.
17	20	21	28	UART2_TXD	UART2 数据发送管脚
				SC0_CLK	Smart Card 0时钟输入管脚
				P0.4	通用数字输入/输出管脚P0.4: 端口0管脚4
				ADC_CH5	ADC输入通道5管脚.
				PWM0_CH3	PWM0 输出通道3管脚.
				PWM2_CH1	PWM2 输出通道1管脚.
				IC3	输入捕获通道 3管脚.
				UART2_RXD	UART2 数据接收管脚.
18	1	22	29	SC0_DAT	Smart Card 0 数据管脚
				STADC	ADC外部触发管脚
				P0.5	通用数字输入/输出管脚
				ADC_CH4	ADC输出通道4管脚.
				PWM0_CH2	PWM0 输出通道2管脚.
				PWM2_CH0	PWM2 输出通道 0管脚.
				IC6	输入捕获通道 6管脚.
UART3_TXD	UART3 数据发送管脚.				
19	2	23	30	SC1_CLK	Smart card 时钟输入管脚.
				T0	定时器/计数器0, 外部计数输入或翻转输出管脚
				P0.6	通用数字输入/输出管脚P0.6: 端口0管脚6

管脚号				管脚名称	管脚描述 ^[1]
MS51XC0BE QFN 20	MS51FC0AE TSSOP20	MS51EC0AE TSSOP28	MS51PC0AE LQFP 32 MS51TC0AE QFN 33		
				ADC_CH3	ADC输入通道3管脚.
				UART0_TXD	UART0发送数据管脚
20	3	24	31	P0.7	通用数字输入/输出管脚P0.7: 端口0管脚7
				ADC_CH2	ADC 输入通道2管脚.
				UART0_RXD	UART0 数据发送管脚.
12	15	15	22	P1.0	通用数字输入/输出管脚P1.0: 端口1管脚0
				PWM0_CH2	PWM0输出通道 2管脚.
				PWM2_CH0	PWM2 输出通道 0管脚.
				IC2	输入捕获通道 2管脚.
				SPI0_CLK	SPI0 时钟管脚
				UART1_TXD	UART1 数据接收管脚
11	14	14	21	P1.1	通用数字输入/输出管脚
				ADC_CH7	ADC输入通道7管脚.
				PWM0_CH1	PWM0 输出通道1管脚.
				PWM1_CH1	PWM1 输出通道1管脚.
				IC1	输入捕获通道1管脚.
				UART3_RXD	UART3数据接收管脚
				SC1_DAT	Smart Card 1 数据管脚
				CLKO	系统时钟输出管脚
10	13	13	20	P1.2	通用数字输入/输出管脚P1.2: 端口1管脚2
				PWM0_CH0	PWM0 输出通道 0管脚.
				PWM1_CH0	PWM1输出通道0管脚.
				IC0	输入捕获通道 0管脚.
				UART3_TXD	UART3 数据发送管脚
				SC1_CLK	Smart Card 1 时钟输入管脚.
9	12	7	12	P1.3	通用数字输入/输出管脚P1.3: 端口1管脚3
				ADC_CH13	ADC 输入通道 13管脚.
				I ² C0_SCL	I ² C0 时钟管脚.
				STADC	ADC外部触发管脚外部启动ADC触发脚
8	11	6	11	P1.4	通用数字输入/输出管脚P1.4: 端口1管脚4
				ADC_CH14	ADC输入通道14管脚.
				PWM0_CH1	PWM0输出通道1管脚.
				PWM1_CH1	PWM1 输出通道1管脚.
				I ² C0_SDA	I ² C0 数据管脚
				PWM0_Brake	PWM0 刹车输入管脚.
7	10	4	7	P1.5	通用数字输入/输出管脚P1.5: 端口1管脚5
				PWM0_CH5	PWM0 输出通道5管脚.
				PWM3_CH1	PWM3输出通道1管脚.
				IC7	输入捕获通道 7管脚.
				SPI0_SS	SPI0 从机选择输入管脚.
				UART3_TXD	UART3数据发送管脚
				SC1_CLK ⁶	Smart card 2 时钟输入管脚
5	8	2	5	P1.6	通用数字输入/输出管脚P1.6: 端口1管脚6
				I ² C0_SDA	I ² C0 数据管脚.
				UART1_TXD	UART1数据发送管脚.
				ICE_DAT	ICE 数据输入/输出管脚.
3	6	28	3	P1.7	通用数字输入/输出管脚P1.7: 端口1管脚7

管脚号				管脚名称	管脚描述 ^[1]
MS51XC0BE QFN 20	MS51FC0AE TSSOP20	MS51EC0AE TSSOP28	MS51PC0AE LQFP 32 MS51TC0AE QFN 33		
				ADC_CH0	ADC 输入通道0管脚.
				SPIO_CLK	SPIO 时钟管脚.
				UART2_RXD	UART2 数据接收管脚.
				SC0_DAT	Smart Card 0 数据管脚
				INT1	外部中断1 输入管脚.
1	4	26	1	P2.0	通用数字输入/输出管脚, RPD (CONFIG0.2) 配置为0时可用
				nRESET	复位脚为施密特触发输入, 用以外部复位信号复位芯片. nRESET 内部带上拉电阻, 外部只需接下拉电容, 也可稳定工作。
-	-	11	16	P2.1	通用数字输入/输出管脚P2.1: 端口2管脚1
				ADC_CH9	ADC输入通道 9 管脚.
				PWM2_CH0	PWM2 输出通道0管脚.
				P2.2	通用数字输入/输出管脚
				ADC_CH10	ADC 输入通道10管脚.
				PWM1_CH1	PWM1 输出通道1管脚.
				UART4_RX6	UART4 数据接收管脚.
				SC2_DAT6	Smart card 2 数据管脚
				P2.3	通用数字输入/输出管脚
				ADC_CH11	ADC 输入通道11管脚.
				PWM1_CH0	PWM1 输出通道 0管脚.
				UART4_TXD	UART4数据发送管脚.
				SC2_CLK	Smart card 2 时钟管脚
				P2.4	通用数字输入/输出管脚
				ADC_CH12	ADC 输入通道 12管脚.
				T0	定时器/计数器1, 外部计数输入或翻转输出管脚
				P2.5	通用数字输入/输出管脚
				ADC_CH15	ADC 输入通道15管脚.
				SPIO_MISO	SPI 主机输入/从机输入管脚.
				UART3_RXD	UART3 数据接收管脚.
				SC1_DAT	Smart card 1 数据管脚
2	5	27	2	P3.0	通用数字输入/输出管脚
				ADC_CH1	ADC输入通道 1管脚.
				PWM2_CH1	PWM2 输出通道 1管脚.
				SPIO_MOSI	SPI 主机输出/从机输入管脚.
				UART2_TXD	UART2数据发送管脚.
				SC0_CLK	Smart card 0 时钟管脚
				INT0	外部中断0 输入管脚.
				OSCIN	当EXTEN[1:0] = 11b 用作外部时钟输入管脚
			18	P3.1	通用数字输入/输出管脚
				PWM2_CH1	PWM2 输出通道 1管脚.
			19	P3.2	通用数字输入/输出管脚
				PWM3_CH0	PWM3 输出通道 0管脚.
			25	P3.3	通用数字输入/输出管脚
				PWM0_CH0	PWM3 输出通道0管脚.
				CLK_OUT	系统时钟输出管脚
				PWM0_BRAKE	PWM0 刹车输入管脚.

管脚号				管脚名称	管脚描述 ^[1]
MS51XC0BE QFN 20	MS51FC0AE TSSOP20	MS51EC0AE TSSOP28	MS51PC0AE LQFP 32 MS51TC0AE QFN 33		
-	-	25	32	P3.4	通用数字输入/输出管脚
				PWM3_CH1	PWM3 输出通道1管脚.
				UART3_RXD	UART3 数据接收管脚.
				SC1_DAT	Smart card 0-1数据管脚
-	-	12	17	P3.5	通用数字输入/输出管脚
				SPI0_SS	SPI0从机选择输入管脚.
-	-	-	9	P3.6	通用数字输入/输出管脚
				UART1_TXD	UART1 数据发送管脚
-	-	-	10	P3.7	通用数字输入/输出管脚
				UART1_RXD	UART1数据接收管脚
<p>注:</p> <ol style="list-style-type: none"> 1. 所有管脚都可以被配置成中断引脚,此功能并没有在管脚描述中提到。详细请查看章节6.4.2引脚中断(PIT) 2. UART0_TXD和UART0_RXD管脚可以通过配置寄存器UART0PX (AUXR1.2)交换位置。 3. [I2C] 功能重映像选项。 I2C管脚可以通过配置寄存器I2CPX (I2CON.0)转换位置。 4. [STADC] 功能重映像选项。 STADC引脚通过配置寄存器STADCPX(ADCCON1.6)转换位置。 5. PIOx寄存器设置管脚是PWM或者GPIO功能。 6. UART2_TXD 和UART2_RXD管脚位置由AUXR2定义。UART3_TXD, UART3_RXD, UART4_TXD和UART4_RXD 管脚由AUXR3定义。 					

5 功能框图

5.1 MS51 32KB 系列框图

下图是MS51的功能框图。用户可以在框图中看到所有的外设。

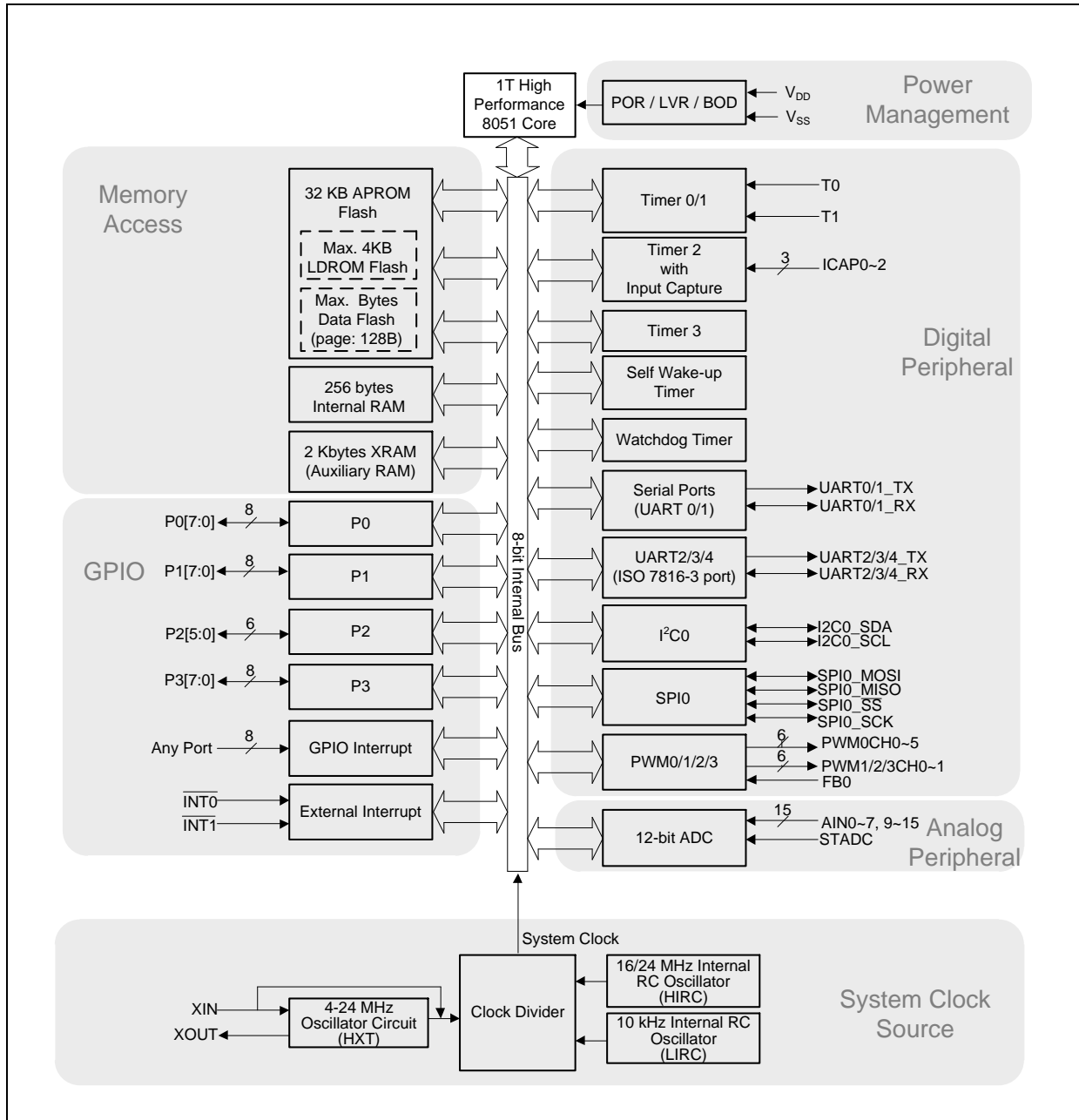


图 5.1- 1 功能框图

6 功能描述

6.1 内存架构

6.1.1 概述

标准的基于80C51微控制器将内存分成两个不同的部分，编程内存和数据内存。编程内存用来存储指令代码。而数据内存用来存储编程执行过程中的数据或变量。

数据内存占用的地址空间独立于编程内存。在MS51中，有256字节内部RAM。针对于需要更多内部RAM的应用，MS51提供另外一个2K字节的片上RAM，叫做XRAM，通过MOVX指令访问。

整个内置FLASH，作为编程存储器被分成了三块。应用ROM（APROM）通常存储用户代码，加载ROM（LDROM）通常存储启动代码，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，每页大小是128字节。FLASH控制单元支持擦除、编程和读模式。使用外部烧写器是通过指定的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。

6.1.2 程序内存编程内存 (APROM 及 LDROM)

程序内存用于存储执行代码，如图 6.1-1所示。任何复位后CPU从地址0000H开始执行。

关于服务中断，中断服务向量位置（叫做中断向量）位于编程内存。每一个中断被分配一个固定的编程内存地址。中断引起CPU跳到中断服务子程序（ISR）开始执行的地方。例如外部中断0被分配到地址0003H。如果外部中断0被使用，它的服务子程序应该从地址0003H开始。如果中断不使用，该地址可以作为通用的编程内存。

中断服务向量位置间隔为八个字节：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。如果一个中断服务子程序足够短，可以完整地放在这8个字节间隔中。而长的中断服务程序需要用JMP指令跳过后面的中断地址，如果此中断地址有被其它中断使用。

MS51提供两个内部编程内存块APROM和LDROM。虽然它们都和标准8051编程内存一样，但是根据它们ROM的大小不一样，扮演着不同的角色。MS51的APROM可以最大到32K字节。用户代码通常放在这里面。CPU从APROM获取指令来执行。MOVC指令也可以从这个区域读取。

另外单独的编程内存块叫做LDROM，它的功能通常是存储ISP的启动代码。它可以更新APROM和CONFIG。APROM里的代码可以重新编程LDROM。APROM和LDROM关于ISP的细节和配置位设置，请查看章节 6.3.1.5“在线系统编程 (ISP)”。注意APROM和LDROM是硬件独立的模块，如果CPU从LDROM启动，CPU会自动重映射PC指针0000H到LDROM开始的地址。因此CPU认为LDROM是单独的编程内存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		

出厂预设置: 1111 1111b

位	名称	描述
2:0	LDSIZE[2:0]	<p>LDROM 容量选择</p> <p>此区域选择LDROM的容量</p> <p>111 = 无 LDROM. APROM 为 32K 字节.</p> <p>110 = LDROM 为 1K 字节. APROM 为 4531K 字节.</p> <p>101 = LDROM 为 2K 字节. APROM 为 4430K 字节.</p> <p>100 = LDROM 为 3K 字节. APROM 为 4329K 字节.</p> <p>0xx = LDROM 为 4K 字节. APROM 为 4228K 字节.</p>

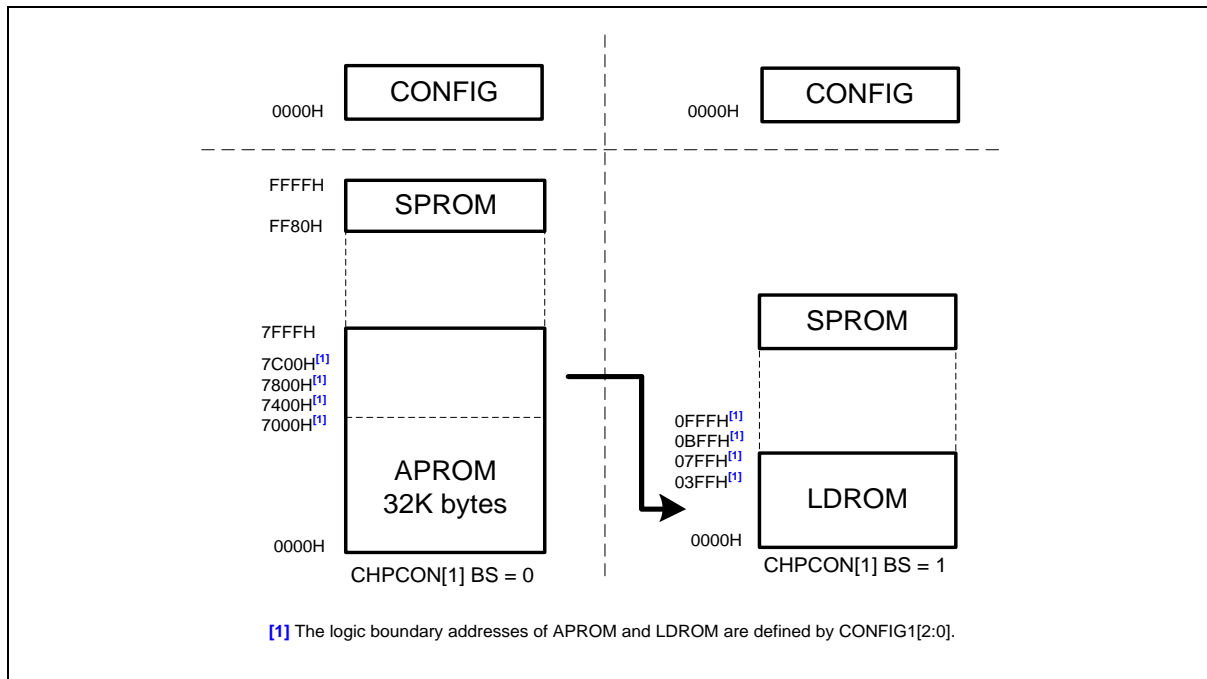


图 6.1- 1 MS51 程序内存映像

6.1.3 数据存储器的数据内存 (Data Flash)

MS51的数据存储器是和APROM和LDROM共享存储地址的。任何页的APROM或LDROM都可以用作非易失数据存储器，不需要专门设置大小。数据存储器的地址有IAP配置。关于IAP的细节请查看章节6.3.1在应用编程 (IAP).所有内置存储器的每次可擦除大小都是128字节。

6.1.4 安全保护存储器 (SPROM)

安全保护存储器 (SPROM)用于应用程序加密。SPROM包含128字节，起始地址从FF80H ~ FFFFH，不支持整片擦除命令。图 6.1-2 SPROM内存映像和安全模式 表示SPROM的最后一个自己字节(地址: FFFFH)用于标识SPROM中的程序是非安全模式还是安全模式。

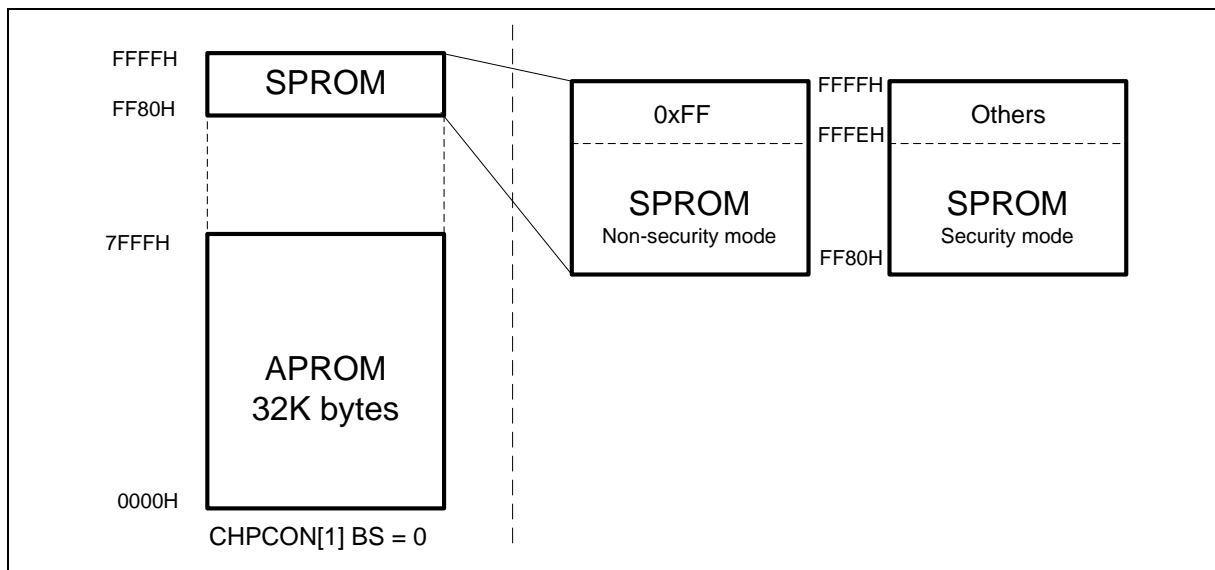


图 6.1-2 SPROM 内存映像和安全模式

(1) SPROM 非安全模式 (最后一个字节是 0xFF). 对 SPROM的操作跟 APROM和LDROM一样的。所有的区域都可以被CPU或者ISP命令读取，也能通过ISP命令擦除和编程。

(2) SPROM 安全模式 (最后一个字节不为 0xFF)。在安全模式下为了隐藏SPROM中的代码，只有当CPU运行在SPROM区域，CPU才能从SPROM执行取指令和获取数据。否则，CPU的操作收到的数据将全是0。在安全模式下，即使ICE接口被连接上，为了保护SPROM，CPU获取的指令也将全部为0。在这个模式下，SPROM不支持ISP/IAP编程和读flash的命令。

6.1.5 用户配置区 (CONFIG)

MS51有一部分硬件配置区。用于配置硬件的一些可选功能比如：安全锁位、系统时钟源等等。配置区可以用过并行烧写器、在电路编程器(ICP)、在应用编程器(ISP)重新编程。CONFIG某些位定义的功能也可以通过SFR修改。因此复位的时候，会将CONFIG位加载到SFR相应位。这些寄存器位SFR的位还可以通过用户软件控制。

CONFIG 位标记 “-”不允许更改。

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	配置启动选择位 该位定义，除软件复位外的所有复位后，MCU启动区域。 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	片上调试OCD模式下，PWM输出状态 该位决定当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 注：只有当PWM通道对应的PIO位设置为1时，该位才有效。
4	OCDEN	片上调试 使能 1 = 禁止片上调试。 0 = 使能片上调试。 注：当MCU运行在OCD调试模式并且 OCDEN = 0，硬件故障复位被禁止，仅仅HFIF标志位被置位。
3	-	保留
2	RPD	复位引脚使能 1 = 使能P2.0/nRST引脚复位功能。P2.0/nRST 作为复位引脚。 0 = 禁止P2.0/nRST引脚复位功能。P2.0/nRST 用作P2.0输入引脚。
1	LOCK	芯片加密使能 1 = 芯片不加密。Flash存储器不加密，用户可通过硬件编程器/ ICP编程器读取FLASH的值。 0 = 芯片加密。全芯片FLASH区域加密，通过硬件编程器/ ICP编程器读取FLASH的值，读回全部为 (FFH)，对FLASH进行编程无效。 注：配置字内容始终不加密，可以读出。当LOCK位配置为0对芯片加密后，配置字内容不能单独擦除或改写，解除芯片加密的唯一方式是执行全擦除动作 (whole chip erase)，一旦执行全擦除动作，FLASH内所有内容将被擦除且配置字内容也会被擦除。 芯片加密，不影响IAP功能。

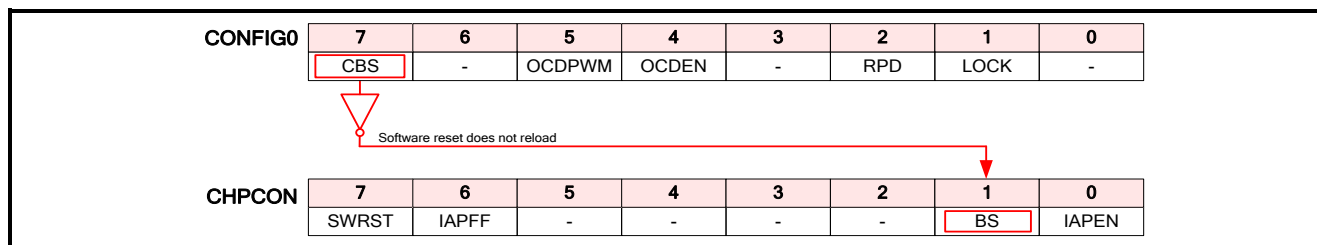


图 6.1-3 CONFIG0 复位后自动重载

CONFIG1

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

工厂预设值: 1111 1111b

位	名称	描述
2:0	LDSIZE[2:0]	LDROM 容量选择 型号 MS51: 111 = 无 LDROM. APROM 32 KB. 110 = LDROM 1 KB. APROM 31KB. 101 = LDROM 2 KB. APROM 30 KB. 100 = LDROM 3 KB. APROM 29 KB. 0xx = LDROM 4 KB. APROM 28 KB.

CONFIG2

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

工厂预设值: 1111 1111b

位	名称	描述
7	CBODEN	欠压检测使能 1 = 打开欠压检测功能. 0 = 关闭欠压检测功能.
6	-	保留
5:4	CBOV[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	BOIAP	欠压禁止IAP 该位决定当系统电压低于 V_{BOD} 时, IAP擦除、编程功能是否禁止。该位仅当欠压检测功能使能后生效。 1 = 当VDD 低于 V_{BOD} , IAP 擦除或编程功能禁止 0 = VDD 任何电压状态下, IAP擦除及编程功能都可执行
2	CBORST	欠压复位使能 该位决定当欠压检测电压低于 V_{BOD} 时, 芯片是否复位。 1 = 使能欠压复位功能. 0 = 禁止欠压复位功能.

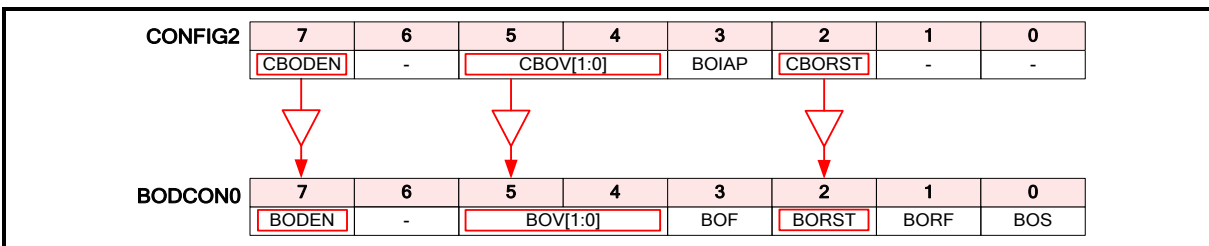


图 6.1-4 CONFIG2 上电复位自动重载

CONFIG3

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

工厂预设值： 1111 1111b

位	名称	描述
7:6	CKFS[1:0]	时钟滤波器时间选择 使能时钟滤波器。提高抗噪声和EMC能力。 00 = 禁止时钟滤波器 其它 = 15 ns
5:0	-	保留

CONFIG4

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

工厂预设值： 1111 1111b

位	名称	描述
7:4	WDTEN[3:0]	看门狗定时器使能 该区域设置MCU运行后，看门狗定时器的功能。 1111 = 禁止WDT。WDT可以通过软件设置成普通定时器。 0101 = 使能WDT超时复位功能。且待机和掉电模式下WDT停止。 其他 =使能WDT超时复位功能。且待机和掉电模式下WDT工作。
3:0	-	保留

6.1.6 数据存储器 (RAM)

6.1.6.1 内部数据存储器

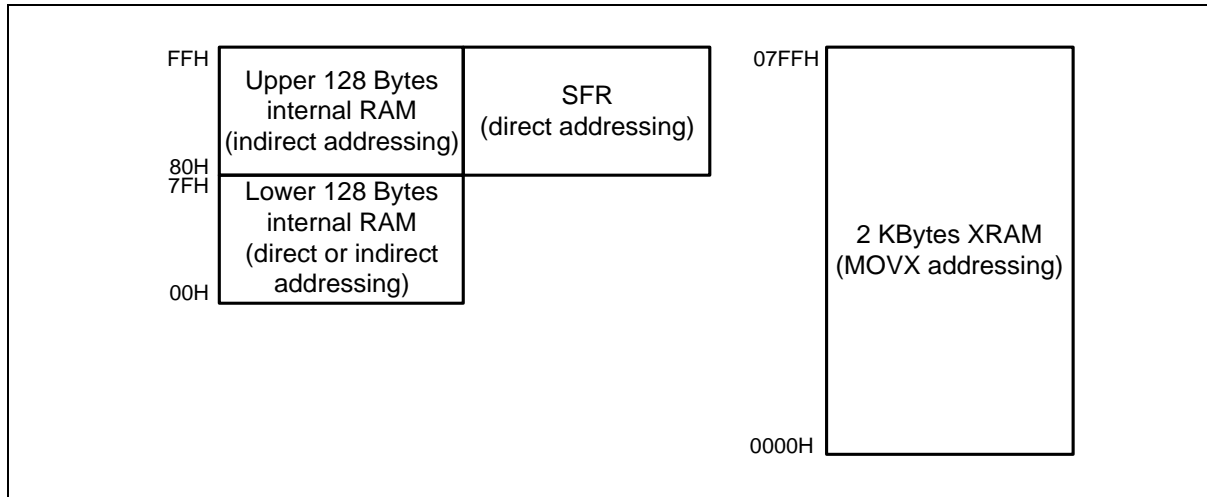


图 6.1-5 数据存储器地址映像

图 6.1-5所示MS51中可用内部数据存储器。内部数据内存占用一个独立的地址空间。内部数据存储器被划分为三块：RAM低128字节、RAM高128字节、128字节的SFR。内部数据内存的地址是8位的宽度，可用于256字节的地址空间。直接寻址高低于7FH的地址会访问特殊功能寄存器（SFR）空间，间接寻址高于7FH的地址会访问RAM的高128字节。虽然SFR地址空间和RAM高128字节共享相同的逻辑地址80H到FFH，事实上他们是物理独立的实体。区别于RAM的高128字节直接寻址仅可以访问SFR。SFR空间中的16个地址既可以字节寻址也可以位寻址。这些位寻址的SFR分布在地址以0H或8H结尾的地方。

内部RAM的低128字节在所有的80C51设备上都存在。最低的32字节作为通用寄存器分成四组8个寄存器，程序指令称呼这些寄存器为R0到R7。程序状态字(PSW[3:4])的两个位RS0和RS1用于选择哪一个寄存器组会被使用。这使代码空间更有效率，因为寄存器指令比其他直接寻址的指令更短。接下来的16个字节(字节地址20H到2FH)是可位寻址的内存空间(位地址00H到7FH)的一部分。80C51指令集包括各种的单位指令。这个域的128个位可以通过这些指令直接寻址。该域的位地址从00H到7FH。

直接或间接寻址都可以访问低128字节空间。但是访问高128字节空间必须采用间接寻址。

另一个应用是将整块内部256字节RAM用作栈空间。此区域用过堆栈指针(SP)来选择，SP保存栈顶地址。无论何时调用JMP、CALL和中断，返回地址都放在堆栈。堆栈在RAM里的起始位置没有限制。但是在默认情况下，复位后堆栈指针指向07H。用户可以根据需要修改这个值。堆栈指针SP会指向最后使用的值。因此，压栈时堆栈指针递增，弹栈时堆栈指针递减。

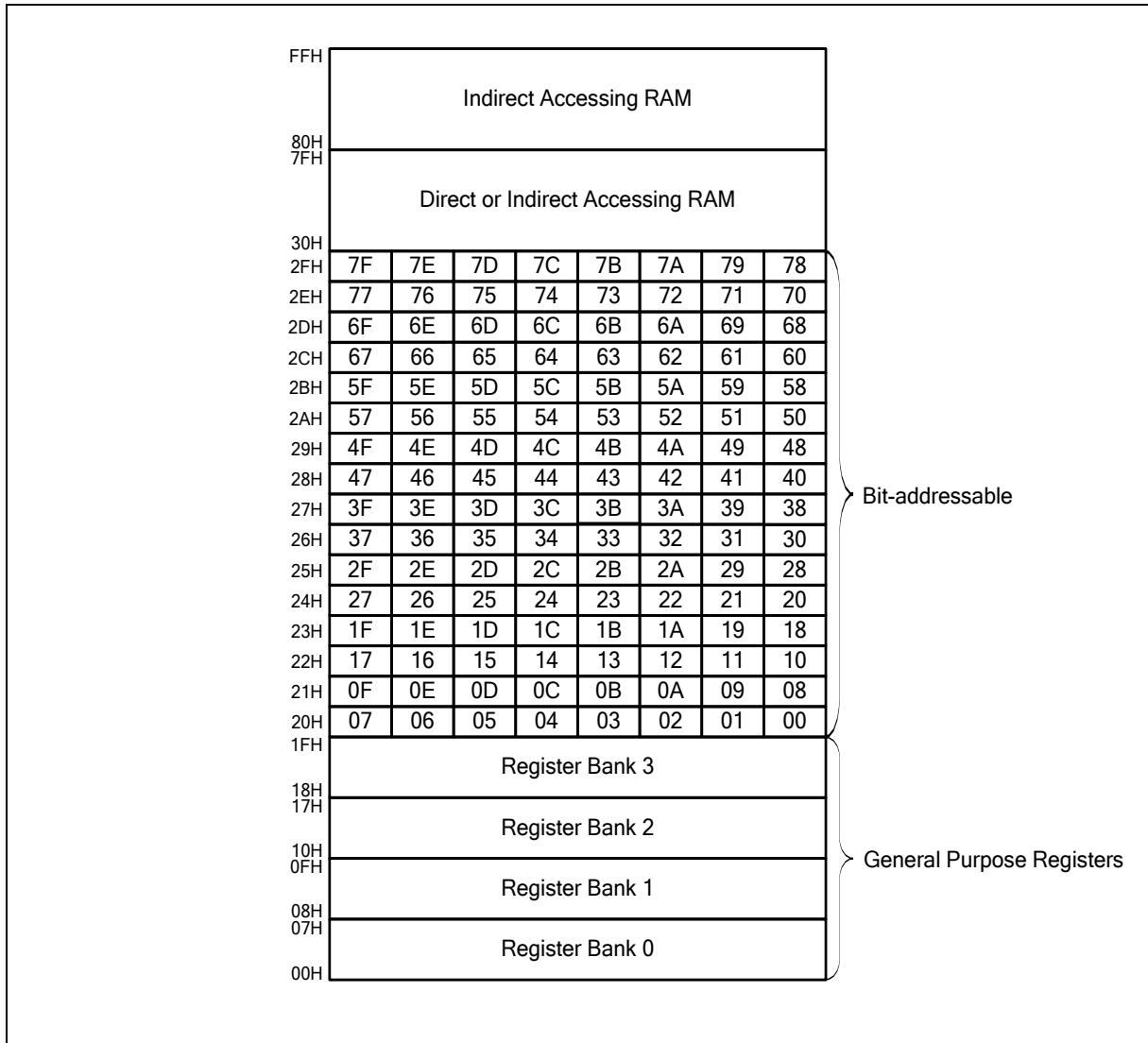


图 6.1-6 内部 256 字节 RAM 地址

6.1.6.2 片上 XRAM

MS51提供额外的片上2 K字节附加的RAM叫做XRAM用来扩展RAM空间。它占用的地址空间从00H 到7FFH。这2 K字节的XRAM通过调用外部指令MOVX @DPTR或 MOVX @Ri. (如下面示例代码.) 间接访问。注意堆栈指针不能位于XRAM的任何区域。

XRAM参考代码:

汇编:

```

MOV     R0,#23H           ;write #5AH to XRAM with address @23H
MOV     A,#5AH
MOVX    @R0,A
MOV     R1,#23H           ;read from XRAM with address @23H
MOVX    A,@R1
MOV     DPTR,#0023H       ;write #5BH to XRAM with address @0023H
MOV     A,#5BH
MOVX    @DPTR,A
MOV     DPTR,#0023H       ;read from XRAM with address @0023H
MOVX    A,@DPTR
    
```

C51:

```

unsigned char temp;           //define data variable
unsigned char xdata xtemp _at_ 0x23; //define variable at xdata 0x23;
xtemp = 0x5B;                 // write #5BH to XRAM with address @0023H
xtemp++;
temp = xtemp;                 //read from XRAM with address @0023H
    
```

6.1.7 特殊功能寄存器 (SFR)

MS51用特殊功能寄存器 (SFRs)来控制或监控外设和他们的模块。SFRs位于地址80到FFH地址空间，仅可以通过直接寻址访问。以0H或8H结尾的SFRs是可以位寻址的。当用户需要修改某一位而不改变其他位的情况下，位寻址是非常有用的。其它所有的 SFRs仅可以字节寻址。MS51包含标准8051中出现的所有SFRs，然而一些额外的 SFRs 也包含在内。因此在原始8051中一些没有使用的字节被给予了新的功能。SFRs 如下所列：

6.1.7.1 SFR页选择

为了在0x80 到 0xFF间容纳128字节的SFRs，增添了SFR页。默认情况下，所有SFR访问的都是SFR页0。在设备初始化期间，SFR页1、2上的一些SFR可能需要被访问。寄存器SFRS用于切换SFR寻址页。

注：特别是，由于中断不可预知，建议在每个中断服务子程序中添加以下操作。在中断子程序开始时将SFRS压入堆栈，推出中断时弹出SFRS。

SFRS – SFR 页选择

寄存器	SFR 地址	复位值
SFRS	91H, 所有页	0000_0000 b

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

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

切换 SFR 页参考代码：

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

6.1.7.2 时效访问保护 (TA)

MS51有几个特殊的功能，如WDT和掉电检测，对系统正常运行非常重要。如果这些控制寄存器不保护，错误的代码可能往它们写入不确定的值，导致运行在错误的状态且失控。为了避免这个风险，MS51有一个保护机制，关键的SFR限制写访问。该保护机制使用时效访问(TA)。下列寄存器与TA处理过程相关。

TA – 时钟控制访问保护

寄存器	地址, 页, 特别说明	复位值
TA	C7H, 所有页	0000_0000 b

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

位	名称	描述
7:0	TA[7:0]	<p>时效访问</p> <p>时效访问寄存器控制保护SFR的访问。为了访问保护位，用户必须首先往TA写入AAH，然后立即写入55H。这两步之后，写允许窗口将打开4个时钟周期，在这个时间内，用户可以写保护SFR。</p>

在时效访问方式，受保护的位有一个写使能时间窗口。只有在该窗口有效的状态下，写才能成功，否则写的直接丢弃。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，时效访问窗口打开。它保持4个时钟周期的打开状态，用户可以写保护位。4个时钟周期后，窗口自动关闭。一旦窗口关闭，写另一个保护位时，必须重新运行解保护程序。TA保护SFR，写入需要时效访问，但读不用。用户可以读TA保护SFR，而不需要往TA寄存器写入AAH和55H。打开时效访问窗口的建议代码如下所示。

```

(CLR EA)                ;if any interrupt is enabled, disable
temporarily
MOV TA, #0AAH
MOV TA, #55H
(Instruction that writes a TA保护 register)
(SETB EA)               ;resume interrupts enabled
    
```

在执行上述指令过程中，任何中断必须关闭，避免执行上述指令的过程中有延时。如果没有中断使能，CLR EA 和 SETB EA 指令可以拿掉。

以下是时效访问例子，展示正确或错误写过程。

例 1,

```

MOV TA, #0AAH           ;3 clock cycles
MOV TA, #55H            ;3 clock cycles
ORL WDCON, #data       ;4 clock cycles
    
```

例 2,

```

MOV TA, #0AAH           ;3 clock cycles
MOV TA, #55H            ;3 clock cycles
    
```

```
NOP                ;1 clock cycle
ANL  BODCON0,#data ;4 clock cycles
```

例 3,

```
MOV  TA,#0AAH      ;3 clock cycles
MOV  TA,#55H       ;3 clock cycles
MOV  WDCON,#data1  ;3 clock cycles
ORL  BODCON0,#data2 ;4 clock cycles
```

例 4,

```
MOV  TA,#0AAH      ;3 clock cycles
NOP                ;1 clock cycle
MOV  TA,#55H       ;3 clock cycles
ANL  BODCON0,#data ;4 clock cycles
```

第1个例子，写保护位在3个时钟周期窗口关闭之前已经完成；第2个例子，写BODCON0在窗口打开期间没有完成，BODCON0的值不会改变；第3个例子，WDCON成功写入，但是写BODCON0超出3个时钟周期窗口，因此BODCON0的值也不会变化；第4个例子，写55H到TA完成再往TA写入AAH 3个时钟周期以后，所以时效访问窗口没有打开，写保护字节不成功。

6.1.7.3 特殊功能寄存器SFR内存映射

页	地址	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0	F8	SCON_1	PWM0DTEN	PWM0DTCNT	PWM0MEN	PWM0MD	PORDIS	EIP1	EIPH1
1			-	-	-	-	-	-	LVRDIS
2			PIPS0	PIPS1	PIPS2	PIPS3	PIPS4	PIPS5	PIPS6
0	F0	B	CAPCON3	CAPCON4	SPCR	SPSR	SPDR	AINDIDS0	EIPH
1			-	-	-	-	-	-	-
2			SC0CR0	SC0CR1	SC1CR0	SC1CR1	SC2CR0	SC2CR1	PIPS7
0	E8	ADCCON0	PICON	PINEN	PIPEN	PIF	C2L	C2H	EIP
1			-	-	-	-	-	-	-
2			SC2DR	SC2EGT	SC2ETURD0	SC2ETURD1	SC2IE	SC2IS	SC2TSR
0	E0	ACC	ADCCON1	ADCCON2	ADCPLY	C0L	C0H	C1L	C1H
1			-	-	-	-	-	-	-
2			SC1DR	SC1EGT	SC1ETURD0	SC1ETURD1	SC1IE	SC1IS	SC1TSR
0	D8	PWM0CON0	PWM0PL	PWM0C0L	PWM0C1L	PWM0C2L	PWM0C3L	PIOCON0	PWM0CON1
1			-	-	-	-	-	-	-
2			SC0DR	SC0EGT	SC0ETURD0	SC0ETURD1	SC0IE	SC0IS	SC0TSR
0	D0	PSW	PWM0PH	PWM0C0H	PWM0C1H	PWM0C2H	PWM0C3H	PNP	PWM0FBD
1			-	-	-	-	-	-	-
2			PWM3PL	PWM3_CH0L	PWM3_CH1L	PWM3CON0	PWM3CON1	PWM3INTC	XTLCON
0	C8	T2CON	T2MOD	RCMP2L	RCMP2H	TL2	TH2	ADCMP	ADCMPL
1			-	-	-	PWM0C4L	PWM0C5L	-	-
2			PWM3PH	PWM3_CH0H	PWM3_CH1H	PWM3MD	PWM3MEN	EIP2	EIPH2
0	C0	I2CON	I2ADDR	ADCRL	ADCRH	T3CON	RL3	RH3	TA
1			-	-	-	PWM0C4H	PWM0C5H	PIOCON1	
2			PWM2PL	PWM2_CH0L	PWM2_CH1L	PWM2CON0	PWM2CON1	PWM2INTC	
0	B8	IP	SADEN	SADEN_1	SADDR_1	I2DAT	I2STAT	I2CLK	I2TOC
1			-	-	-	-	-	-	-
2			PWM2PH	PWM2_CH0H	PWM2_CH1H	PWM2MD	PWM2MEN	-	-
0	B0	P3	P0M1	P0M2	P1M1	P1M2	TOE	-	IPH
1			P0S	P0SR	P1S	P1SR	-	-	PWM0INTC
2			PWM1PL	PWM1_CH0L	PWM1_CH1L	PWM1CON0	PWM1CON1	PWM1INTC	PIOCON2
0	A8	IE	SADDR	WDCON	BODCON1	P3M1	P3M2	IAPFD	IAPCN
1			-	-	P3S	P3SR	-	PWM0INTC	
2			PWM1PH	PWM1_CH0H	PWM1_CH1H	PWM1MD	PWM1MEN	-	-
0	A0	P2	-	AUXR1	BODCON0	IAPTRG	IAPUEN	IAPAL	IAPAH
1			-	-	-	-	-	-	-
2			AUXR2	AUXR3	AUXR4	AUXR5	AUXR6	AUXR7	AUXR8
0	98	SCON	SBUF	SBUF_1	EIE	EIE1	RSR	-	CHPCON
1			-	-	-	-	-	-	-
2			AINDIDS1	P0DW	P1DW	P2DW	P3DW	-	-
0	90	P1	SFRS	CAPCON0	CAPCON1	CAPCON2	CKDIV	CKSWT	CKEN
1			-	-	-	-	-	-	-
2			P0UP	P1UP	P2UP	P3UP	-	-	RWKH
0	88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	WKCON
1			-	-	-	-	-	-	-
2			P2M1	P2M2	P2SR	P2S	ADCSN	ADCCN	ADCSR
0	80	P0	SP	DPL	DPH	RCTRIM0	RCTRIM1	RWK	PCON
1			-	-	-	-	-	-	-
2			-	-	-	ADCBAL	ADCBAL	ADCCON3	-

注:

- SFR空间中未使用的地址用“-”标记。访问这些地址会有不可预知的影响，请避免这种情况。
- 该类型标示为可位寻址字节。
- 该类型标示为任何页内都可直接读写。

表 6.1- 1 特殊功能寄存器地址映像

6.1.7.4 SFR 描述及复位值

寄存器	定义	地址	页	高位 7	6	5	4	3	2	1	低位 ^[1] 0	复位值 ^[2]	TA
EIPH1	扩展中断优先级高位 1	FFH	0	-	-	PPWM3H	PPWM2H	PPWM1H	PWKTH	PT3H	PSH_1	0000 0000b	
LVRDIS	低电压检测禁止	FFH	1	LVRDIS[7:0]								0000 0000b	Y
PIPS6	引脚中断控制6	FFH	2	-	BSEL2	BSEL1	BSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
-	-	FEH	0	-	-	-	-	-	-	-	-	-	
EIP1	扩展中断优先级 1	FEH	1	-	-	PPWM3	PPWM2	PPWM1	PWKTH	PT3	PS_1	0000 0000b	
PIPS5	引脚中断控制 5	FEH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PORDIS	上电复位禁止	FDH	0	PORDIS[7:0]								0000 0000b	Y
-	-	FDH	1	-	-	-	-	-	-	-	-	-	
PIPS4	引脚中断控制4	FDH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0MD	PWM0 屏蔽数据	FCH	0	-	-	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0	0000 0000b	
-	-	FCH	1	-	-	-	-	-	-	-	-	-	
PIPS3	引脚中断控制3	FCH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0MEN	PWM0 屏蔽使能	FBH	0	-	-	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0	0000 0000b	
-	-	FBH	1	-	-	-	-	-	-	-	-	-	
PIPS2	引脚中断控制 2	FBH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0DTCNT	PWM0 死区计数器	FAH	0	PWM0DTCNT[7:0]								0000 0000b	Y
-	-	FAH	1	-	-	-	-	-	-	-	-	-	
PIPS1	引脚中断控制1	FAH	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
PWM0DTCEN	PWM0 死区使能	F9H	0	-	-	-	PWM0DTCNT.8	-	PDT45EN	PDT23EN	PDT01EN	0000 0000b	Y
-	-	F9H	1	-	-	-	-	-	-	-	-	-	
PIPS0	引脚中断控制0	F9H	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
SCON_1	串口1控制	F8H	A	SM0_1 FE_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TL_1	RI_1	0000 0000b	
EIPH	扩展中断优先级高	F7H	0	PT2H	PSPIH	PFBH	PWDTH	PPWMH	PCAPH	PIIH	PI2CH	0000 0000b	
-	-	F7H	1	-	-	-	-	-	-	-	-	-	
PIPS7	引脚中断控制7	F7H	2	-	PSEL2	PSEL1	PSEL0	-	BSEL2	BSEL1	BSEL0	0000 0000b	
AINDIDS0	ADC 信道数字输入信道禁止	F6H	0	P11DIDS	P03DIDS	P04DIDS	P05DIDS	P06DIDS	P07DIDS	P30DIDS	P17DIDS	0000 0000b	
-	-	F6H	1	-	-	-	-	-	-	-	-	-	
SC2CR1	智能卡2 控制寄存器 1	F6H	2	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN		
SPDR	串行外设数据寄存器	F5H	0	SPDR[7:0]								0000 0000b	
-	-	F5H	1	-	-	-	-	-	-	-	-	-	
SC2CR0	智能卡2 控制寄存器 0	F5H	2	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	0000 0000b	
SPSR	串行外设状态寄存器	F4H	0	SPIF	WCOL	SPIOVF	MODF	DISMODF	TXBUF	-	-	0000 0000b	
-	-	F4H	1	-	-	-	-	-	-	-	-	-	
SC1CR1	智能卡1 控制寄存器1	F4H	2	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN	0000 0000b	
SPCR	串行外设控制寄存器	F3H	0	SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR1	SPR0	0000 0000b	
SPCR2	串行外设控制寄存器 2	F3H	1	-	-	-	-	-	-	SPIS1	SPIS0	0000 0000b	
SC1CR0	智能卡1 控制寄存器 0	F3H	2	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	0000 0000b	
CAPCON4	输入捕获控制 4	F2H	0	-	-	-	-	CAP23	CAP22	CAP21	CAP20	0000 0000b	
-	-	F2H	1	-	-	-	-	-	-	-	-	-	
SC0CR1	智能卡0 控制寄存器 1	F2H	2	OPE	PBOFF	WLS1	WLS0	TXDMAEN	RXDMAEN	CLKKEEP	UARTEN	0000 0000b	
CAPCON3	输入捕获控制 3	F1H	0	CAP13	CAP12	CAP11	CAP10	CAP03	CAP02	CAP01	CAP00	0000 0000b	
-	-	F1H	1	-	-	-	-	-	-	-	-	-	
SC0CR0	智能卡0控制寄存器 0	F1H	2	NSB	T	RXBGTEN	CONSEL	AUTOCEN	TXOFF	RXOFF	SCEN	0000 0000b	
B	B 累加器	F0H	A	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	0000 0000b	
EIP	扩展中断优先级	EFH	0	PT2	PSPI	PFB	PWDTH	PPWM	PCAP	PPI	PI2C	0000 0000b	
-	-	EFH	1	-	-	-	-	-	-	-	-	-	
SC2TSR	智能卡2 传输状态寄存器	EFH	2	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	0000 1010b	
C2H	输入捕获 2 高字节	EEH	0	C2H[7:0]								0000 0000b	
-	-	EEH	1	-	-	-	-	-	-	-	-	-	
SC2IS	智能卡2 中断状态寄存器	EEH	2	-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF	0000 0010b	
C2L	输入捕获 2 低字节	EDH	0	C2L[7:0]								0000 0000b	
-	-	EDH	1	-	-	-	-	-	-	-	-	-	

寄存器	定义	地址	页	高位 7	6	5	4	3	2	1	低位 ^[1] 0	复位值 ^[2]	TA
SC2IE	智能卡2 中断使能控制寄存器	EDH	2	-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN	0000 0000b	
PIF	引脚中断标志	ECH	0	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	0000 0000b	
-	-	ECH	1	-	-	-	-	-	-	-	-	-	
SC2ETURD1	智能卡2 ETU 除频器寄存器1	ECH	2	-	SCDIV2	SCDIV1	SCDIV0	ETURDIV1 1	ETURDIV1 0	ETURDIV9	ETURDIV8	0000 0000b	
PIPEN	引脚中断正相/上升边缘特性使能	EBH	0	PIPEN7	PIPEN6	PIPEN5	PIPEN4	PIPEN3	PIPEN2	PIPEN1	PIPEN0	0000 0000b	
-	-	EBH	1	-	-	-	-	-	-	-	-	-	
SC2ETURD0	智能卡2 ETU 除频器寄存器	EBH	2	ETURDIV[7:0]								0111 0011b	
PINEN	引脚中断反相/下降边缘特性使能	EAH	0	PINEN7	PINEN6	PINEN5	PINEN4	PINEN3	PINEN2	PINEN1	PINEN0	0000 0000b	
-	-	EAH	1	-	-	-	-	-	-	-	-	-	
SC2EGT	智能卡2 额外保护时间寄存器	EAH	2	SC2EGT[7:0]								0000 0000b	
PICON	引脚中断控制	E9H	0	PIT7	PIT6	PIT5	PIT4	PIT3	PIT2	PIT1	PIT0	0000 0000b	
-	-	E9H	1	-	-	-	-	-	-	-	-	-	
SC2DR	智能卡2 数据寄存器	E9H	2	SC2DR[7:0]								0000 0000b	
ADCCON0	ADC 控制 0	E8H	A	ADCF	ADCS	ETGSEL1	ETGSEL0	ADCHS3	ADCHS2	ADCHS1	ADCHS0	0000 0000b	
C1H	输入捕获1 高字节	E7H	0	C1H[7:0]								0000 0000b	
-	-	E7H	1	-	-	-	-	-	-	-	-	-	
SC1TSR	智能卡1 传输状态寄存器	E7H	2	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	0000 1010b	
C1L	输入捕获1 低字节	E6H	0	C1L[7:0]								0000 0000b	
-	-	E6H	1	-	-	-	-	-	-	-	-	-	
SC1IS	智能卡1 中断状态寄存器	E6H	2	-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF	0000 0010b	
C0H	输入捕获0 高字节	E5H	0	C0H[7:0]								0000 0000b	
-	-	E5H	1	-	-	-	-	-	-	-	-	-	
SC1IE	智能卡1 中断使能寄存器	E5H	2	-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN	0000 0000b	
C0L	输入捕获0 低字节	E4H	0	C0L[7:0]								0000 0000b	
-	-	E4H	1	-	-	-	-	-	-	-	-	-	
SC1ETURD1	智能卡1 ETU 除频器寄存器1	E4H	2	-	SCDIV2	SCDIV1	SCDIV0	ETURDIV1 1	ETURDIV1 0	ETURDIV9	ETURDIV8	0011 0001b	
ADCDLY	ADC 触发延时寄存器	E3H	0	ADCDLY[7:0]								0000 0000b	
-	-	E3H	1	-	-	-	-	-	-	-	-	-	
SC1ETURD0	智能卡1 ETU 除频器寄存器	E3H	2	ETURDIV[7:0]								0111 0011b	
ADCCON2	ADC 控制 2	E2H	0	ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	ADCAQT2	ADCAQT1	ADCAQT0	ADCDLY.8	0000 0000b	
-	-	E2H	1	-	-	-	-	-	-	-	-	-	
SC1EGT	智能卡1 额外保护时间寄存器	E2H	2	SC1EGT[7:0]								0000 0000b	
ADCCON1	ADC 控制 1	E1H	0	OCEN	STADCPX	ADCDIV1	ADCDIV0	ETGTYP1	ETGTYP0	ADCEX	ADCEN	0000 0000b	
-	-	E1H	1	-	-	-	-	-	-	-	-	-	
SC1DR	智能卡1 数据寄存器	E1H	2	SC1DR[7:0]								0000 0000b	
ACC	累加器	E0H	A	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	0000 0000b	
PWMOCN1	PWM0 控制 1	DFH	0	PWMMOD1	PWMMOD0	GP	PWMTYP	FBINEN	PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
-	-	DFH	1	-	-	-	-	-	-	-	-	-	
SC0TSR	智能卡0 传输状态寄存器	DFH	2	ACT	BEF	FEF	PEF	TXEMPTY	TXOV	RXEMPTY	RXOV	0000 1010b	
PIOCON0	PWM0 引脚功能0	DEH	0	-	-	PIO03	PIO01	PIO00	PIO10	PIO11	PIO12	0000 0000b	
-	-	DEH	1	-	-	-	-	-	-	-	-	-	
SC0IS	智能卡0 中断状态寄存器	DEH	2	-	-	-	ACERRIF	BGTIF	TERRIF	TBEIF	RDAIF	0000 0010b	
PWM0C3L	PWM0 通道3 占空比低字节	DDH	0	PWM0_CH3[7:0]								0000 0000b	
-	-	DDH	1	-	-	-	-	-	-	-	-	-	
SC0IE	智能卡0 中断使能控制寄存器	DDH	2	-	-	-	ACERRIEN	BGTIEN	TERRIEN	TBEIEN	RDAIEN	0000 0000b	
PWM0C2L	PWM0 通道2 占空比低字节	DCH	0	PWM0_CH2[7:0]								0000 0000b	
-	-	DCH	1	-	-	-	-	-	-	-	-	-	
SC0ETURD1	智能卡0 ETU 除频器寄存器	DCH	2	-	SCDIV2	SCDIV1	SCDIV0	ETURDIV1 1	ETURDIV1 0	ETURDIV9	ETURDIV8	0011 0001b	

寄存器	定义	地址	页	高位 7	6	5	4	3	2	1	低位 ^[1] 0	复位值 ^[2]	TA
PWM0C1L	PWM0 通道1占空比低字节	DBH	0	PWM0_CH1[7:0]								0000 0000b	
-	-	DBH	1	-	-	-	-	-	-	-	-	-	
SC0ETURD0	智能卡0 ETU除频器寄存器	DBH	2	ETURDIV[7:0]								0111 0011b	
PWM0C0L	PWM0通道0占空比低字节	DAH	0	PWM0C0[7:0]								0000 0000b	
-	-	DAH	1	-	-	-	-	-	-	-	-	-	
SC0EGT	智能卡0 额外时间保护寄存器	DAH	2	SC0EGT[7:0]								0000 0000b	
PWM0PL	PWM0 周期值低字节	D9H	0	PWM0P[7:0]								0000 0000b	
-	-	D9H	1	-	-	-	-	-	-	-	-	-	
SC0DR	智能卡0 数据寄存器	D9H	2	SC0DR[7:0]								0000 0000b	
PWM0CON0	PWM0控制0	D8H	0	PWM0RUN	LOAD	PWMF	CLRPWM	PWM3RUN	PWM2RUN	PWM1RUN	P33FBINEN	0000 0000b	
PWM0FBD	PWM急停控制	D7H	0	FBF	FBINLS	FBD5	FBD4	FBD3	FBD2	FBD1	R/W	0000 0000b	
-	-	D7H	1	-	-	-	-	-	-	-	-	-	
XTLCON	XLT 时钟控制	D7H	2	-	HXSG2	HXSG1	HXSG0	-	-	-	-	0111 0111b	Y
PNP	PWM反相控制	D6H	0	-	-	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0	0000 0000b	
-	-	D6H	1	-	-	-	-	-	-	-	-	-	
PWM3INTC	PWM3 中断控制	D6H	2	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
PWM0C3H	PWM0 通道 3 占空比高字节	D5H	0	PWM0_CH3[15:8]								0000 0000b	
-	-	D5H	1	-	-	-	-	-	-	-	-	-	
PWM3CON1	PWM3 控制 1	D5H	2	PWMMOD1	PWMMOD0	PWMTYP	-	-	PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
PWM0C2H	PWM0 通道 2 占空比高字节	D4H	0	PWM0_CH2[15:8]								0000 0000b	
-	-	D4H	1	-	-	-	-	-	-	-	-	-	
PWM3CON0	PWM3 控制 0	D4H	2	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000b	
PWM0C1H	PWM0 通道 1 占空比高字节	D3H	0	PWM0_CH1[15:8]								0000 0000b	
-	-	D3H	1	-	-	-	-	-	-	-	-	-	
PWM3_C H1L	PWM3 通道 1 占空比低字节	D3H	2	PWM3_CH1[7:0]								0000 0000b	
PWM0C0H	PWM0通道 0 占空比高字节	D2H	0	PWM0C0[15:8]								0000 0000b	
-	-	D2H	1	-	-	-	-	-	-	-	-	-	
PWM3_C H0L	PWM3 通道 0 占空比低字节	D2H	2	PWM3_CH0[7:0]								0000 0000b	
PWM0PH	PWM0 周期值高字节	D1H	0	PWM0P[15:8]								0000 0000b	
-	-	D1H	1	-	-	-	-	-	-	-	-	-	
PWM3PL	PWM3 周期值低字节	D1H	2	PWM3P[7:0]								0000 0000b	
PSW	程序状态字	D0H	A	CY	AC	F0	RS1	RS0	OV	-	P	0000 0000b	
ADCMPH	ADC 比较值高字节	CFH	0	ADCMPL[11:4]								0000 0000b	
-	-	CFH	1	-	-	-	-	-	-	-	-	-	
EIPH2	扩展中断优先级 高 2	CFH	2	-	-	-	-	-	PUART4H	PUART3H	PUART2H	0000 0000b	
ADCMPL	ADC 比较值低字节	CEH	0	-	-	-	-	ADCMP3	ADCMP2	ADCMP1	ADCMP0	0000 0000b	
-	-	CEH	1	-	-	-	-	-	-	-	-	-	
EIP2	扩展中断优先级 1	CEH	2	-	-	-	-	-	PUART4	PUART3	PUART2	0000 0000b	
TH2	定时器 2 高字节	CDH	0	TH2[7:0]								0000 0000b	
PWM0C5L	PWM0 通道 5 占空比低字节	CDH	1	PWM0C5[7:0]								0000 0000b	
PWM3MEN	PWM3 屏蔽使能	CDH	2	-	-	-	-	-	-	PMEN1	PMEN0	0000 0000b	
TL2	定时器 2 低字节	CCH	0	TL2[7:0]								0000 0000b	
PWM0C4L	PWM0 通道 4 占空比低字节	CCH	1	PWM0C4[7:0]								0000 0000b	
PWM3MD	PWM3 屏蔽数据	CCH	2	-	-	-	-	-	-	PMD1	PMD0	0000 0000b	
RCMP2H	定时器 2 比较值 高字节	CBH	0	RCMP2H[7:0]								0000 0000b	
-	-	CBH	1	-	-	-	-	-	-	-	-	-	
PWM3_C H1H	PWM3 通道 1 占空比高字节	CBH	2	PWM3_CH1[15:8]								0000 0000b	

寄存器	定义	地址	页	高位 7	6	5	4	3	2	1	低位 ^[1] 0	复位值 ^[2]	TA
RCMP2L	定时器 2 比较值低字节	CAH	0	RCMP2L[7:0]								0000 0000b	
-	-	CAH	1	-	-	-	-	-	-	-	-	-	
PWM3_C H0H	PWM3 通道 0 占空比 高字节	CAH	2	PWM3_CH0[15:8]								0000 0000b	
T2MOD	定时器 2 模式	C9H	0	LDEN	T2DIV2	T2DIV1	T2DIV0	CAPCR	CMPCR	LDTS1	LDTS0	0000 0000b	
-	-	C9H	1	-	-	-	-	-	-	-	-	-	
PWM3PH	PWM3周期值高字节	C9H	2	PWM3P[15:8]								0000 0000b	
T2CON	定时器 2 控制	C8H	A	TF2	-	-	-	-	TR2	-	CM/RL2	0000 0000b	
TA	时钟控制访问保护	C7H	A	TA[7:0]								0000 0000b	
RH3	定时器 3 重载值 高字节	C6H	0	RH3[7:0]								0000 0000b	
PIOCON1	PWM0引脚控制器	C6H	1	PIO17	-	PIO15	-	PIO04	PIO05	PIO14	-	-	
PWM2INT C	PWM2 中断控制	C6H	2	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
RL3	Timer 3 重载值低字节	C5H	0	RL3[7:0]								0000 0000b	
PWM0C5 H	PWM0 通道5占空比 高字节	C5H	1	PWM0C5[15:8]								0000 0000b	
PWM2CO N1	PWM2 控制 1	C5H	2	PWMMOD1	PWMMOD0	PWMTYP	-	-	PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
T3CON	定时器3 控制	C4H	0	SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS2	T3PS1	T3PS0	0000 0000b -	
PWM0C4 H	PWM0通道4占空比 高字节	C4H	1	PWM0C4[15:8]								0000 0000b	
PWM2CO N0	PWM2 控制 1	C4H	2	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000b	
ADCRH	ADC 结果 高字节	C3H	0	ADCR[11:4]								0000 0000b	
-	-	C3H	1	-	-	-	-	-	-	-	-	-	
PWM2_C H1L	PWM2 通道1占空比 低字节	C3H	2	PWM2_CH1[7:0]								0000 0000b	
ADCRL	ADC 结果低字节	C2H	0	-	-	-	-	ADCR3	ADCR2	ADCR1	ADCR0	0000 0000b	
-	-	C2H	1	-	-	-	-	-	-	-	-	-	
PWM2_C H0L	PWM2 通道0占空比 低字节	C2H	2	PWM2_CH0[7:0]								0000 0000b	
I2ADDR	I2C本机从机地址	C1H	0	I2ADDR7	I2ADDR6	I2ADDR5	I2ADDR4	I2ADDR3	I2ADDR2	I2ADDR1	GC	0000 0000b	
-	-	C1H	1	-	-	-	-	-	-	-	-	-	
PWM2PL	PWM2 周期值低字节	C1H	2	PWM2P[7:0]								0000 0000b	
I2CON	I2C控制器	C0H	A	-	I2CEN	STA	STO	SI	AA	-	I2CPX	0000 0000b	
I2TOC	I2C 超时计数器	BFH	0	-	-	-	-	-	I2TOCEN	DIV	I2TOF	0000 0000b	
-	-	BFH	1	-	-	-	-	-	-	-	-	-	
PWM2ME N	PWM2 屏蔽使能	BFH	2	-	-	-	-	-	-	PMEN1	PMEN0	0000 0000b	
I2CLK	I2C 时钟	BEH	0	I2CLK[7:0]								0000 1001b	
-	-	BEH	1	-	-	-	-	-	-	-	-	-	
PWM2MD	PWM2 屏蔽数据	BEH	2	-	-	-	-	-	-	PMD1	PMD0	0000 0000b	
I2STAT	I2C 状态	BDH	0	I2STAT7	I2STAT6	I2STAT5	I2STAT4	I2STAT3	0	0	0	1111 1000b	
-	-	BDH	1	-	-	-	-	-	-	-	-	-	
PWM2_C H1H	PWM2 通道1占空比 高字节	BDH	2	PWM2_CH1[15:8]								0000 0000b	
I2DAT	I2C 数据	BCH	0	I2DAT[7:0]								0000 0000b	
-	-	BCH	1	-	-	-	-	-	-	-	-	-	
PWM2_C H0H	PWM2 通道0占空比 高字节	BCH	2	PWM2_CH0[15:8]								0000 0000b	
SADDR_1	从机1地址	BBH	0	SADDR_1[7:0]								0000 0000b	
-	-	BBH	1	-	-	-	-	-	-	-	-	-	
PWM2PH	PWM2 周期值高字节	BBH	2	PWM2P[15:8]								0000 0000b	
SADEN_1	从机1 地址掩码	BAH	0	SADEN_1[7:0]								0000 0000b	
-	-	BAH	1	-	-	-	-	-	-	-	-	-	
-	-	BAH	2	-	-	-	-	-	-	-	-	-	
SADEN	从机0 地址掩码	B9H	0	SADEN[7:0]								0000 0000b	
-	-	B9H	1	-	-	-	-	-	-	-	-	-	

寄存器	定义	地址	页	高位 7	6	5	4	3	2	1	低位 ^[1] 0	复位值 ^[2]	TA
-	-	B9H	2	-	-	-	-	-	-	-	-	-	
IP	中断优先级	B8H	A	-	PADC	PBOD	PS	PT1	PX1	PT0	PX0	0000 0000b	
IPH	中断优先级 高	B7H	0	-	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H	0000 0000b	
-	-	B7H	1	-	-	-	-	-	-	-	-	-	
PIOCON2	PWM 引脚控制	B7H	2	PIO34	PIO33	PIO32	PIO31	-	PIO23	PIO22	PIO21	0000 0000b	
-	-	B6H	0	-	-	-	-	-	-	-	-	-	
-	-	B6H	1	-	-	-	-	-	-	-	-	-	
PWM1INTC	PWM1 中断控制	B6H	2	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSELO	0000 0000b	
TOE	定时器01 输出使能	B5H	0	-	-	-	-	T1OE	T0OE	-	-	0000 0000b	
-	-	B5H	1	-	-	-	-	-	-	-	-	-	
PWM1CON1	PWM1 控制 1	B5H	2	PWMMOD1	PWMMOD0	-	PWMTYP	-	PWMDIV2	PWMDIV1	PWMDIV0	0000 0000b	
P1M2	端口1 模式选择2	B4H	0	P1M2.7	P1M2.6	P1M2.5	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0	0000 0000b	
P1SR	端口1 转换速率	B4H	1	P1SR.7	P1SR.6	P1SR.5	P1SR.4	P1SR.3	P1SR.2	P1SR.1	P1SR.0	0000 0000b	
PWM1CON0	PWM1 控制 0	B4H	2	PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-	0000 0000b	
P1M1	端口1 模式选择1	B3H	0	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	1111 1111b	
P1S	端口1 施密特触发输入	B3H	1	P1S.7	P1S.6	P1S.5	P1S.4	P1S.3	P1S.2	P1S.1	P1S.0	0000 0000b	
PWM1_C H1L	PWM1 通道1 占空比低字节	B3H	2	PWM1_CH1[7:0]								0000 0000b	
P0M2	端口0 模式选择 2	B2H	0	P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0	0000 0000b	
P0SR	端口0 转换速率	B2H	1	P0SR.7	P0SR.6	P0SR.5	P0SR.4	P0SR.3	P0SR.2	P0SR.1	P0SR.0	0000 0000b	
PWM1_C H0L	PWM1通道0占空比低字节	B2H	2	PWM1_CH0[7:0]								0000 0000b	
P0M1	端口0 模式选择 1	B1H	0	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0	1111 1111b	
P0S	端口0 施密特触发输入t	B1H	1	P0S.7	P0S.6	P0S.5	P0S.4	P0S.3	P0S.2	P0S.1	P0S.0	0000 0000b	
PWM1PL	PWM1周期值低字节	B1H	2	PWM1P[7:0]								0000 0000b	
P3	端口 3	B0H	A	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	输出锁存, 1111 1111b 输入 xxxx xxxxb ^[3]	
IAPCN	IAP 控制寄存器	AFH	0	IAPA17	IAPA16	FOEN	FCEN	FCTRL3	FCTRL2	FCTRL1	FCTRL0	0011 0000b	
PWM0INTC	PWM0 中断控制	AFH	1	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSELO	0000 0000b	
-	-	AFH	2	-	-	-	-	-	-	-	-	-	
IAPFD	IAP 数据寄存器	AEH	0	IAPFD[7:0]								0000 0000b	
-	-	AEH	1	-	-	-	-	-	-	-	-	-	
PWM1MEN	PWM1 屏蔽使能	AEH	2	-	-	-	-	-	-	PMEN1	PMEN0	0000 0000b	
P3M2	端口3 模式选择 2	ADH	0	P3M2.7	P3M2.6	P3M2.5	P3M2.4	P3M2.3	P3M2.2	P3M2.1	P3M2.0	0000 0000b	
P3SR	端口3 转换速率	ADH	1	P3SR.7	P3SR.6	P3SR.5	P3SR.4	P3SR.3	P3SR.2	P3SR.1	P3SR.0	0000 0000b	
PWM1MD	PWM1 屏蔽数据	ADH	2	-	-	-	-	-	-	PMD1	PMD0	0000 0000b	
P3M1	端口3 模式选择 1	ACH	0	P3M1.7	P3M1.6	P3M1.5	P3M1.4	P3M1.3	P3M1.2	P3M1.1	P3M1.0	1111 1111b	
P3S	端口3 施密特触发输入	ACH	1	P3S.7	P3S.6	P3S.5	P3S.4	P3S.3	P3S.2	P3S.1	P3S.0	0000 0000b	
PWM1_C H1H	PWM1通道1占空比高字节	ACH	2	PWM1_CH1[15:8]								0000 0000b	
BODCON1	欠压检测控制1	ABH	0	-	-	-	-	-	LPBOD1	LPBOD0	BODFLT	FOR, 0000 0001b 其他, 0000 0UUUUb	Y
-	-	ABH	1	-	-	-	-	-	-	-	-	-	
PWM1_C H0H	PWM1通道0占空比高字节	ABH	2	PWM1_CH0[15:8]								0000 0000b	
WDCON	WTD控制	AAH	0	WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS2	WDPS1	WDPS0	FOR, 0000 0111b WDT, 0000 1UUUUb 其他, 0000 UUUUUb	Y
-	-	AAH	1	-	-	-	-	-	-	-	-	-	
PWM1PH	PWM1周期值高字节	AAH	2	PWM1P[15:8]								0000 0000b	

寄存器	定义	地址	页	高位 7	6	5	4	3	2	1	低位 ^[1] 0	复位值 ^[2]	TA
SADDR	从机0地址	A9H	0	SADDR[7:0]								0000 0000b	
-	-	A9H	1	-	-	-	-	-	-	-	-	-	
PWM1PH	PWM1周期值高字节	A9H	2	PWM1P[15:8]								0000 0000b	
IE	中断使能寄存器	A8H	A	EA	EADC	EBOD	ES	ET1	EX1	ET0	EX0	0000 0000b	
IAPAH	IAP 地址高字节	A7H	0	IAPA[15:8]								0000 0000b	
-	-	A7H	1	-	-	-	-	-	-	-	-	-	
AUXR8	辅助寄存器 8	A7H	2	CLODIV3	CLODIV2	CLODIV1	CLODIV0	CKTESTOE N3	CKTESTOE N2	CKTESTOE N1	CKTESTOE N0	0000 0000b	Y
IAPAL	IAP 地址低字节	A6H	0	IAPA[7:0]								0000 0000b	
-	-	A6H	1	-	-	-	-	-	-	-	-	-	
AUXR7	辅助寄存器 7	A6H	2	-	-	-	SPI0NSSP 1	SPI0NSSP 0	SPI0MOSIP	SPI0MISOP	SPI0CKP	0000 0000b	
IAPUEN	IAP 更新使能	A5H	0	-	-	-	SPMEN	SPUEN	CFUEN	LDUEN	APUEN	0000 0000b	Y
-	-	A5H	1	-	-	-	-	-	-	-	-	-	
AUXR6	辅助寄存器6	A5H	2	-	-	-	UART4DG	UART3DG	UART2DG	UART1DG	UART0DG	0000 0000b	
IAPTRG	IAP触发寄存器	A4H	0	-	-	-	-	-	-	-	IAPGO	0000 0000b	Y
-	-	A4H	1	-	-	-	-	-	-	-	-	-	
AUXR5	辅助寄存器5	A4H	2	CLOP	TOP	-	-	PWM3_CH 1P1	PWM3_CH 1P0	PWM3_CH 0P1	PWM3_CH 0P0	0000 0000b	
BODCON 0	欠压检测控制0	A3H	0	BODEN	-	BOV1	BOV0	BOF	BORST	BORF	BOS	POR, CCCC XC0Xb BOD, UUUU XU1Xb 其他, UUUU XU0Xb	Y
-	-	A3H	1	-	-	-	-	-	-	-	-	-	
AUXR4	辅助寄存器 4	A3H	2	PWM2_CH 1P1	PWM2_CH 1P0	PWM2_CH 0P1	PWM2_CH 0P0	PWM1_CH 1P1	PWM1_CH 1P0	PWM1_CH 0P1	PWM1_CH 0P0	0000 0000b	
AUXR1	辅助寄存器 1	A2H	0	SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS	POR, 0000 0000b 软件, 1U00 0000b nRESET 引脚, U100 0000b 其他, UUUU 0000b	
-	-	A2H	1	-	-	-	-	-	-	-	-	-	
AUXR3	辅助寄存器 3	A2H	2	UART4TXP 1	UART4TXP 0	UART4RXP 1	UART4RXP 0	UART3TXP 1	UART3TXP 0	UART3RXP 1	UART3RXP 0	0000 0000b	
-	-	A1H	0	-	-	-	-	-	-	-	-	-	
-	-	A1H	1	-	-	-	-	-	-	-	-	-	
AUXR2	辅助寄存器 2	A1H	2	UART2TXP 1	UART2TXP 0	UART2RXP 1	UART2RXP 0	UART1TXP 1	UART1TXP 0	UART1RXP 1	UART1RXP 0	0000 0000b	
P2	Port 2端口 2	A0H	A	-	-	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	输出锁存 0000 000Xb 输入, 0000 000Xb	
CHPCON	芯片控制	9FH	0	SWRST	IAPFF	-	-	-	-	BS[5]	IAPEN	软件, 0000 00U0b 其他, 0000 00C0b	Y
-	-	9FH	1	-	-	-	-	-	-	-	-	-	
-	-	9FH	2	-	-	-	-	-	-	-	-	-	
-	-	9EH	1	-	-	-	-	-	-	-	-	-	
-	-	9EH	1	-	-	-	-	-	-	-	-	-	
-	-	9EH	2	-	-	-	-	-	-	-	-	-	
RSR	复位标志寄存器	9DH	0	-	-	HardF(mirro red from AUXR1.5)	POF (mirrored from PCON.4)	RSTPINF (mirrored from AUXR1.6)	BORF (mirrored from BODCON. 1)	WDTRF (mirrored from WDCON.3)	SWRF (mirrored from AUXR1.7)	00XX XXXXb	
-	-	9DH	1	-	-	-	-	-	-	-	-	-	
P3DW	端口3下拉电阻控制	9DH	2	P3DW.7	P3DW.6	P3DW.5	P3DW.4	P3DW.3	P3DW.2	P3DW.1	P3DW.0	0000 0000b	
EIE1	扩展中断使能寄存 1	9CH	0	-	-	EPWM3	EPWM2	EPWM1	EWKT	ET3	ES_1	0000 0000b	
-	-	9CH	1	-	-	-	-	-	-	-	-	-	
P2DW	端口2下拉电阻控制	9CH	2	-	-	P2DW.5	P2DW.4	P2DW.3	P2DW.2	P2DW.1	P2DW.0	0000 0000b	
EIE	扩展中段使能寄存器	9BH	0	ET2	ESPI	EFB	EWDT	EPWM0	ECAP	EPI	EI2C	0000 0000b	
-	-	9BH	1	-	-	-	-	-	-	-	-	-	
P1DW	端口1下拉电阻控制	9BH	2	P1DW.7	P1DW.6	P1DW.5	P1DW.4	P1DW.3	P1DW.2	P1DW.1	P1DW.0	0000 0000b	
SBUF_1	串口1数据缓存	9AH	0	SBUF_1[7:0]								0000 0000b	

寄存器	定义	地址	页	高位 7	6	5	4	3	2	1	低位 ^[1] 0	复位值 ^[2]	TA
-	-	9AH	1	-	-	-	-	-	-	-	-	-	
P0DW	端口 0 下拉电阻控制	9AH	2	P0DW.7	P0DW.6	P0DW.5	P0DW.4	P0DW.3	P0DW.2	P0DW.1	P0DW.0	0000 000b	
SBUF	串口0数据缓存	99H	0	SBUF[7:0]								0000 000b	
-	-	99H	1	-	-	-	-	-	-	-	-	-	
AINDIDS1	ADC 信道数字输入信道关闭1	99H	2	P25DIDS	P14DIDS	P13DIDS	P24DIDS	P23DIDS	P22DIDS	P21DIDS	-	0000 000b	
SCON	串口0控制器	98H	A	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000 000b	
CKEN	时钟使能	97H	0	EXTEN1	EXTEN0	HIRCEN	-	-	-	-	CKSWTF	0011 000b	Y
-	-	97H	1	-	-	-	-	-	-	-	-	-	
RWKH	WKT加载值高字节	97H	2	RWK[15:8]								0000 000b	
CKSWT	时钟切换	96H	0	HXTST	ECKP00ST	HIRCST	-	ECKP30ST	OSC1	OSC0	-	0011 000b	Y
-	-	96H	1	-	-	-	-	-	-	-	-	-	
-	-	96H	2	-	-	-	-	-	-	-	-	0000 000b	
CKDIV	时钟除频器	95H	0	CKDIV[7:0]								0000 000b	
-	-	95H	1	-	-	-	-	-	-	-	-	-	
P3UP	端口 3 上拉电阻控制	95H	2	P3UP.7	P3UP.6	P3UP.5	P3UP.4	P3UP.3	P3UP.2	P3UP.1	P3UP.0	0000 000b	
CAPCON2	输入捕获控制 2	94H	0	-	ENF2	ENF1	ENF0	-	-	-	-	0000 000b	
-	-	94H	1	-	-	-	-	-	-	-	-	-	
P2UP	端口 2 上拉电阻控制	94H	2	-	-	P2UP.5	P2UP.4	P2UP.3	P2UP.2	P2UP.1	P2UP.0	0000 000b	
CAPCON1	输入捕获控制 1	93H	0	-	-	CAP2LS1	CAP2LS0	CAP1LS1	CAP1LS0	CAP0LS1	CAP0LS0	0000 000b	
-	-	93H	1	-	-	-	-	-	-	-	-	-	
P1UP	端口 1 上拉电阻控制	93H	2	P1UP.7	P1UP.6	P1UP.5	P1UP.4	P1UP.3	P1UP.2	P1UP.1	P1UP.0	0000 000b	
CAPCON0	输入捕获控制 0	92H	0	-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0	0000 000b	
-	-	92H	1	-	-	-	-	-	-	-	-	-	
P0UP	端口 0 上拉电阻控制	92H	2	P0UP.7	P0UP.6	P0UP.5	P0UP.4	P0UP.3	P0UP.2	P0UP.1	P0UP.0	0000 000b	
SFRS	SFR 页选择	91H	A	-	-	-	-	-	-	SFRPSEL1	SFRPSEL0	0000 000b	
P1	Port 4 端口 1	90H	A	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	输出锁存, 1111 1111b 输入 XXXX XXXXb	
WKCON	WTK控制寄存器	8FH	0	-	-	-	WKTF	WKTR	WKPS2	WKPS1	WKPS0	0000 000b	
-	-	8FH	1	-	-	-	-	-	-	-	-	-	
ADCSR	ADC 状态寄存器	8FH	2	-	-	-	-	-	CMPHIT	HDONE	FDONE	0000 000b	
CKCON	时钟控制器	8EH	0	-	PWMCKS	-	T1M	T0M	-	CLOEN	-	0000 000b	
-	-	8EH	1	-	-	-	-	-	-	-	-	-	
ADCCN	ADC目前采样个数	8EH	2	ADCCN[7:0]								0000 000b	
TH1	定时器 1 高字节	8DH	0	TH1[7:0]								0000 000b	
-	-	8DH	1	-	-	-	-	-	-	-	-	-	
ADCSN	ADC采样个数	8DH	2	ADCSN[7:0]								0000 000b	
TH0	定时器 0 高字节	8CH	0	TH0[7:0]								0000 000b	
-	-	8CH	1	-	-	-	-	-	-	-	-	-	
P2S	端口 2 施密特触发输入	8CH	2	P2S.7	P2S.6	P2S.5	P2S.4	P2S.3	P2S.2	P2S.1	P2S.0	0000 000b	
TL1	定时器 1 低字节	8BH	0	TL1[7:0]								0000 000b	
-	-	8BH	1	-	-	-	-	-	-	-	-	-	
P2SR	端口 2 转换速率	8BH	2	P2SR.7	P2SR.6	P2SR.5	P2SR.4	P2SR.3	P2SR.2	P2SR.1	P2SR.0	0000 000b	
TL0	定时器 0 低字节	8AH	0	TL1[7:0]								0000 000b	
-	-	8AH	1	-	-	-	-	-	-	-	-	-	
P2M2	端口2 模式选择 2	8AH	2	P2M2.7	P2M2.6	P2M2.5	P2M2.4	P2M2.3	P2M2.2	P2M2.1	P2M2.0	0000 000b	
TMOD	-	89H	0	-	-	-	-	-	-	-	-	0000 000b	
-	-	89H	1	-	-	-	-	-	-	-	-	-	
P2M1	端口模式选择 1	89H	2	P2M1.7	P2M1.6	P2M1.5	P2M1.4	P2M1.3	P2M1.2	P2M1.1	P2M1.0	0000 000b	
TCON	定时器0和1控制	88H	A	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 000b	
PCON	电源控制	87H	A	SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL	POR, 0001 000b 其他, 000U 000b	
RWKL	加载值低字节	86H	0	RWK[7:0]								0000 000b	
-	-	86H	1	-	-	-	-	-	-	-	-	-	

寄存器	定义	地址	页	高位 7	6	5	4	3	2	1	低位 ^[1] 0	复位值 ^[2]	TA
ADCCON3	ADC 控制3	86H	2	-	-	HIE	CONT	ADCAQT2	ADCAQT1	ADCAQT0	SLOW	0000 0000b	
RCTRIM1	高速内部振荡器 16MHz调整值1	85H	0	-	-	-	HIRC24	-	-	-	HIRCTRIM. 0	0000 0000b	Y
-	-	85H	1	-	-	-	-	-	-	-	-	-	
ADCBAH	ADC 内存基地址高字 节	85H	2	-	-	-	-	ADCBA3	ADCBA2	ADCBA1	ADCBA0	0000 0000b	
RCTRIM0	高速内部振荡器 16MHz调整值0	84H	0	HIRCTRIM[8:1]								0000 0000b	Y
-	-	84H	1	-	-	-	-	-	-	-	-	-	
ADCBAL	ADC 内存基地址低字 节	84H	2	ADCBA[7:0]								0000 0000b	
DPH	数据指针高字节	83H	A	DPTR[15:8]								0000 0000b	
DPL	数据指针低字节	82H	A	DPTR[7:0]								0000 0000b	
SP	堆栈指针	81H	A	SPI[7:0]								0000 0111b	
P0	端口 0	80H	A	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	输出锁存, 1111 1111b 输入, XXXX XXXXb	

注:

- () 意思是SFRs的可位寻址地址。
- 复位值符号描述。0: 逻辑 0; 1: 逻辑 1; U: 不变; C: 见 [5]; X见 [3], [6], 和 [7].
- 复位之后所有I/O引脚默认为悬浮输入模式。如果 RPD (CONFIG0.2) 一直保持1, 读取P2.0始终为0。复位后OCDDA和OCDCCK引脚将保持准双向带上拉电阻的模式 600个LIRC时钟周期之后变为输入模式。
- 这些SFRs有TA写保护。
- 在规定复位后, 根据CONFIG值, SFRs有些位初始化时设置。
- BOF复位值取决于CONFIG2不同的设置和VDD 电压值。
- 当欠压检测使能, BOS是只读标志, 由VDD值来决定。

表 6.1-2 特殊功能寄存器定义和复位值

6.1.7.5 所有SFR介绍

注:

1. 所有SFR的复位值意义为: U-不改变; C-由CONFIG初始化; X- 基于实际芯片状态。

Pn – 端口 (可位寻址)

寄存器	SFR 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	0011_1111 b
P3	B0H, 所有页, 可位寻址	1111_1111 b

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

位	名称	描述
7:0	Pn[7:0]	端口 端口是8位通用输入输出端口

SP – 堆栈指针

寄存器	SFR 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]	<p>堆栈指针</p> <p>堆栈指针存储的是RAM的地址，该地址是堆栈的起始地址。在执行PUSH 或 CALL指令的时候，在数据被存储之前，堆栈指针递增。注意：SP的默认值是07H。因此堆栈起始位置在08H。</p>

DPL – 数据指针低字节

寄存器	SFR 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]	<p>数据指针低字节</p> <p>这是16位数据指针的低字节，DPL 结合DPH作为16位的数据指针DPTR访问想要访问的RAM或编程内存地址。DPS (AUXR0.0) 位决定哪一个数据指针DPTR 或 DPTR1激活。</p>

DPH – 数据指针高字节

寄存器	SFR 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]	<p>数据指针高字节</p> <p>这是16位数据指针的低高字节， DPL 结合DPH作为16位的数据指针DPTR访问想要访问的RAM或编程内存地址。DPS (AUXR0.0) 位决定哪一个数据指针DPTR 或 DPTR1激活 。</p>

RWKL – 加载值低字节自唤醒定时器重载数据低字节寄存器

寄存器	SFR 地址	复位值
RWKL	86H, 页 0	0000 0000b

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

位	名称	描述
7:0	RWK[7:0]	WKT 重载数据低字节 加载WKT时16位数值的低字节。 注意：若不分频，加载值不能是 FFFFH

RWKH – 加载值高字节自唤醒定时器重装载数据高字节寄存器

寄存器	SFR 地址	复位值
RWKH	97H, 页 2	0000 0000b

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

位	名称	描述
7:0	RWK[15:8]	<p>WKT 重装载数据高字节</p> <p>加载WKT时16位数值的高字节。</p> <p>注意：若不分频，加载值不能是 FFFFH</p>

PCON – 电源控制

寄存器	SFR 地址	复位值
PCON	87H, 所有页	POR: 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
7	SMOD	串行 0 双波特率使能 当UART0 在模式2或定时器1的溢出用作串口模式1或3的波特率时，设置此位可加倍波特率。表 6.9-1 串口UART0 模式/波特率描述
6	SMOD0	串口 0 帧错误标志访问使能 0 = SCON.7 访问 SM0 位。 1 = SCON.7 访问 FE 位。
4	POF	上电复位标志 当上电复位后该位置1。表示冷复位、上电复位完成。其他任何复位不会影响该位。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0。
2	GF0	通用标志 0 通用标志可由用户通过软件置位和清0。
1	PD	掉电模式 设置该位使MCU进入掉电模式。在此模式下，CPU和外设时钟停止，程序计数器（PC）挂起，此时系统为最小功耗模式。CPU从掉电模式下唤醒后，该位自动由硬件清零，且程序继续执行唤醒系统的中断源对应的中断服务程序ISR。从ISR返回后，设备从让系统进入掉电模式的指令后续指令继续执行。 注：如果IDL位和PD位同时置位，MCU进入掉电模式。从掉电模式退出后不会进入空闲模式。
0	IDL	空闲模式 设置该位使MCU进入空闲模式。在此模式下，CPU时钟停止，且程序计数器（PC）挂起，但是所有外设继续工作。CPU从空闲模式唤醒后，该位自动由硬件清零，且程序继续执行唤醒系统的中断源对应的中断服务程序ISR。从ISR返回后，设备从让系统进入空闲模式的指令后续指令继续执行。

TCON – 定时器 0和1控制

寄存器	SFR 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	读 (电平) 读/写(边沿)	读/写	读 (电平) 读/写(边沿)	读/写

位	名称	描述
7	TF1	定时器 1 溢出标志 在定时器1溢出时该位置1。当程序相应定时器1中断执行相应的中断服务程序时，该位自动清0。软件 It is automatically cleared 也可对其写1或0。
6	TR1	定时器 1 启动控制 0 = 定时器1禁止。清该位将终止定时器1并且当前计数值将保存到TH1和TL1中。 1 = 使能定时器 1。
5	TF0	定时器 0 溢出标志 在定时器0溢出时该位置1。当程序响应定时器0中断执行相应的中断服务程序时，该位自动清0。软件 也可对其写1或写0。
4	TR0	定时器 0启动控制 0 = 定时器 0 禁止。清除该位将停止定时器0并且当前计数值将保存到TH0和TL0中。 1 = 使能定时器0。
3	IE1	外部中断 1 边沿标志 如果 IT1 = 1 (下降沿触发), 当检测到下降沿时该标志将被硬件置位。该位将保持置1直到软件清零或 在外部中断1服务程序中硬件清零。 如果 IT1 = 0 (低电平触发), 此标志是 $\overline{INT1}$ 输入信号逻辑电平的反转。软件不可控制。
2	IT1	外部中断 1 类型选择 该位选择 $\overline{INT1}$ 的中断触发类型。 0 = $\overline{INT1}$ 是低电平触发。 1 = $\overline{INT1}$ 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), 当检测到下降沿时该标志将被硬件置位。该位将保持置1直到软件清零或 在外部中断0服务程序中硬件清零。 如果 IT0 = 0 (低电平触发), 此标志是 $\overline{INT0}$ 输入信号逻辑电平的反转。软件不可控。

位	名称	描述
0	IT0	<p>外部中断 0 类型选择</p> <p>该位选择$\overline{INT0}$的中断触发类型。</p> <p>0 = $\overline{INT0}$ 是低电平触发</p> <p>1 = $\overline{INT0}$ 是下降沿触发</p>

TMOD – 定时器 0 和 1 模式

寄存器	SFR 地址	复位值
TMOD	89H, 页 0	0000_0000b

7	6	5	4	3	2	1	0
GATE	C/ \bar{T}	M1	M0	GATE	C/ \bar{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 = 不管 $\overline{INT1}$ 的逻辑电平, TR1 为1时, 定时器1时钟运行。 1 = 当 $\overline{INT1}$ 为1, TR1为1时, 定时器1时钟运行。															
6	C/ \bar{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 <table border="0"> <tr> <td><u>M1</u></td> <td><u>M0</u></td> <td>定时器 1 模式</td> </tr> <tr> <td>0</td> <td>0</td> <td>模式 0: 13位定时器/计数器</td> </tr> <tr> <td>0</td> <td>1</td> <td>模式 1: 16位定时器/计数器</td> </tr> <tr> <td>1</td> <td>0</td> <td>模式 2: TH1自动重载的8位定时器/计数器</td> </tr> <tr> <td>1</td> <td>1</td> <td>模式 3: 定时器1停止</td> </tr> </table>	<u>M1</u>	<u>M0</u>	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: TH1自动重载的8位定时器/计数器	1	1	模式 3: 定时器1停止
<u>M1</u>	<u>M0</u>		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: TH1自动重载的8位定时器/计数器															
1	1	模式 3: 定时器1停止															
4	M0																
3	GATE	定时器 0 门控制 0 = 不管 $\overline{INT0}$ 的逻辑电平, TR0为1时, 定时器0的时钟运行。 1 = 当 $\overline{INT0}$ 为1, TR0为1时, 定时0的时钟运行。															
2	C/ \bar{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 <table border="0"> <tr> <td><u>M1</u></td> <td><u>M0</u></td> <td>定时器 0 模式</td> </tr> <tr> <td>0</td> <td>0</td> <td>模式 0: 13位定时器/计数器</td> </tr> <tr> <td>0</td> <td>1</td> <td>模式 1: 16位 定时器/计数器</td> </tr> <tr> <td>1</td> <td>0</td> <td>模式 2: TH0自动重载的8位定时器/计数器</td> </tr> <tr> <td>1</td> <td>1</td> <td>模式 3: TL0 作为8位定时器/计数器, TH0 作为8位定时器。</td> </tr> </table>	<u>M1</u>	<u>M0</u>	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: TH0自动重载的8位定时器/计数器	1	1	模式 3: TL0 作为8位定时器/计数器, TH0 作为8位定时器。
<u>M1</u>	<u>M0</u>		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: TH0自动重载的8位定时器/计数器															
1	1	模式 3: TL0 作为8位定时器/计数器, TH0 作为8位定时器。															
0	M0																

TL0 – 定时器 0 低字节

寄存器	SFR 地址	复位值
TL0	8AH, 页 0	0000_0000b

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

位	名称	描述
7:0	TL0[7:0]	定时器 0 低字节 TL0是定时器0 16位计数器的低字节。

TL1 – 定时器 1 低字节

寄存器	SFR 地址	复位值
TL1	8BH, 页0	0000_0000b

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

位	名称	描述
7:0	TL1[7:0]	<p>定时器 1 低字节</p> <p>TL1是定时器1 16位计数器的低字节。</p>

TH0 – 定时器 0 高字节

寄存器	SFR 地址	复位值
TH0	8CH, 页0	0000_0000b

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

位	名称	描述
7:0	TH0[7:0]	定时器 0 高字节 TL0 TH0是定时器0 16位计数器的高字节。

TH1 – 定时器 1 高字节

寄存器	SFR 地址	复位值
TH1	8DH, 页0	0000_0000b

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

位	名称	描述
7:0	TH1[7:0]	定时器 1 高字节 TL4TH1是定时器1 16位计数器的高字节。

CKCON – 时钟控制

寄存器	SFR 地址	复位值
CKCON	8EH, 页0	0000_0000b

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

位	名称	
7	-	保留
6	PWMCKS	PWM 时钟源选择 0 = PWM时钟源选择系统时钟F _{sys} . 1 = PWM时钟源选择定时器1溢出.
5	-	保留
4	T1M	定时器 1 时钟源选择 0 = 定时器1时钟源选择系统时钟/12。兼容标准8051。 1 = 定时器1时钟源选择系统时钟。
3	T0M	定时器0时钟源选择 0 = 定时器0时钟源选择系统时钟/12。兼容标准8051。 1 = 定时器0 时钟源选择系统时钟。
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 引脚输出。 注意: CLO 由 AUXR5.7 设置.
0	-	保留

WKCON – WTK控制寄存器

寄存器	SFR 地址	复位值
WKCON	8FH, 页0	0000_0000b

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

位	名称	描述
7:5	-	保留
4	WKTF	<p>WKT 溢出标志</p> <p>WKT 溢出时此位置1, 若使能了中断, 会产生中断, 需软件清0</p> <p>如果WKT中断和全局中断使能, 置位该位会使CPU执行WKT中断服务程序。该位不会被硬件自动清零, 应该通过软件清零。</p>
3	WKTR	<p>WKT运行控制</p> <p>0 = WKT停止.</p> <p>1 = WKT开始计数</p> <p>注意: 只有在此位为0时, 才可以加载 WKT 的值, 在此位为1时加载WKT结果将不可预知</p>
2:0	WKPS[2:0]	<p>WKT 分频</p> <p>这些位决定WKT时钟的预分频.</p> <p>000 = 1/1.</p> <p>001 = 1/4.</p> <p>010 = 1/16.</p> <p>011 = 1/64.</p> <p>100 = 1/256.</p> <p>101 = 1/512.</p> <p>110 = 1/1024.</p> <p>111 = 1/2048.</p>

SFRS – SFR 页选择

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

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

位	名称	描述
7:2	-	保留
1:0	SFRPAGE[1:0]	SFR 页选择 0 = 指令访问SFR页 0。 1 =指令访问SFR 页 1。 2 =指令访问SFR 页 2。

CAPCON0 – 输入捕获控制 0

寄存器	SFR 地址	复位值
CAPCON0	92H, 页 0	0000_0000b

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

位	名称	描述
7	-	保留
6	CAPEN2	输入捕获通道2使能 0 = 输入捕获通道2禁止。 1 = 输入捕获通道2使能。
5	CAPEN1	输入捕获通道1使能 0 = 输入捕获通道1禁止。 1 = 输入捕获通道1使能。
4	CAPEN0	输入捕获通道0使能 0 = 输入捕获通道0禁止。 1 = 输入捕获通道0使能。
3	-	保留
2	CAPF2	输入捕获通道2标志 如果输入捕获通道2探测到边沿触发事件发生, 该位由硬件置位, 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, 该位由硬件置位, 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, 该位由硬件置位, 由软件清零。

CAPCON1 – 输入捕获控制 1

寄存器	SFR 地址	复位值
CAPCON1	93H, 页 0	0000_0000b

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

位	名称	描述
7:6	-	保留
5:4	CAP2LS[1:0]	输入捕获通道2电平选择 00 = 下降沿. 01 = 上升沿. 10 = 上升或下降沿. 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 = 上升或下降沿. 11 = 保留.

CAPCON2 – 输入捕获控制 2

寄存器	SFR 地址	复位值
CAPCON2	94H, 页0	0000_0000b

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

位	名称	描述
7	-	保留
6	ENF2	输入捕获通道2的噪声滤波器使能 0 = 输入捕获通道2的噪声滤波器禁止。 1 = 输入捕获通道2的噪声滤波器使能。
5	ENF1	输入捕获通道1的噪声滤波器使能 0 = 输入捕获通道1的噪声滤波器禁止。 1 = 输入捕获通道1的噪声滤波器使能。
4	ENF0	输入捕获通道0的噪声滤波器使能 0 = 输入捕获通道0的噪声滤波器禁止。 1 = 输入捕获通道0的噪声滤波器使能。
3:0	-	保留

CKDIV – 时钟除频器

寄存器	SFR 地址	复位值
CKDIV	95H, 页0	0000_0000b

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

位	名称	描述
7:0	CKDIV[7:0]	<p>时钟除频器</p> <p>根据CKDIV值，系统时钟频率FSYS如下式所示。</p> <p>当CKDIV = 00H 时， $F_{SYS} = F_{OSC}$</p> <p>当CKDIV = 01H to FFH时， $F_{SYS} = \frac{F_{OSC}}{2 \times CKDIV}$</p>

CKSWT – 时钟切换

寄存器	SFR 寄存器	复位值
CKSWT	96H, 页 0, TA 保护	0011_0000 b

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

位	名称	描述
7	HXTST	高速外部晶振 4 MHz 至 24 MHz 状态 0 = 高速外部晶振不稳定或被禁用 1 = 高速外部晶振已使能且稳定
6	ECKP00ST	外部时钟P00的输入状态 0 = 外部时钟P00输入不稳定或被禁用 1 = 外部时钟P00输入已使能且稳定
5	HIRCST	高速内部振荡器状态 0 = 高速内部振荡器不稳定或被禁用 1 = 高速内部振荡器已使能且稳定
-	-	保留
3	ECKP30ST	外部时钟P30的输入状态 0 = 外部时钟P30输入不稳定或被禁用 1 = 外部时钟P30输入已使能且稳定
2:1	OSC[1:0]	振荡器选择位 该位是用来选择系统时钟源 00 = 内部16MHz振荡器 01 = 外部时钟控制，通过EXTEN[1:0] (CKEN[7:6]) 设置 10 = 内部10kHz振荡器 11 = 保留 注意该位段只写，读回来的值可能与当前时钟源不一致
0	-	保留

CKEN – 时钟使能

寄存器	SFR 地址	复位值
CKEN	97H, 页 0, TA 保护	0011_0000 b

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

位	名称	描述
7:6	EXTEN[1:0]	<p>外部时钟源使能</p> <p>11 =外部时钟输入通过OSCIN (P30)使能 10 =外部时钟输入通过HXTIN (P00)使能 01 = 外部晶振 4~24 MHz 使能 00 = 禁用外部时钟输入。 P30/P00/P01 作为普通 I/O或其它功能使用</p>
5	HIRCEN	<p>高速内部振荡器16MHz使能</p> <p>0 = 高速内部振荡器禁用 1 = 高速内部振荡器使能</p> <p>注意一旦设置IAPEN (CHPCON. 0)位开启IAP功能, HIRC将会自动使能, 硬件也会设置HIRCEN和HIRCST位。 IAPEN被清除后, HIRCEN和EHCST位会恢复为原始值。</p>
4	LIRCEN	<p>Low speed internal oscillator 10 kHz enable</p> <p>0 = The low speed internal oscillator Disabled. 1 = The low speed internal oscillator Enabled.</p> <p>Note that when (1) WDT is enabled, (2) WKT is running by the clock source of the internal 10 kHz oscillator ,(3) BOD is enabled, or (4) LVR filter is enabled, a write 0 to LIRCEN will be ignored. LIRCEN is always 1 and the internal 10 kHz oscillator is always enabled.</p>
3:1	-	保留
0	CKSWTF	<p>时钟切换错误标志位</p> <p>0 = 系统时钟源切换成功 1 = 先前用户试图切换系统时钟的时钟源, 没有开启或不稳定。如果待切换的时钟源不稳定, 该位将一直保持为1, 直到时钟源稳定并切换成功为止。</p>

SCON – 串口控制

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	串口模式选择
6	SM1	SMOD0 (PCON.6) = 0: 参见 表 6.9-1 串口UART0 模式/波特率描述 SMOD0 (PCON.6) = 1: SM0/FE用作帧错误指示标志(FE), 可以被软件清除。 0 = 帧错误 (FE)没有发生 1 =检测到帧错误 (FE) 发生.
5	SM2	多机通信模式使能 该位功能取决于串口0模式。 模式 0: 该位选择波特率 FSYS/12 或 FSYS/2。 0 = 时钟运行在 FSYS/12 波特率, 与标志8051兼容。 1 = 时钟运行在 FSYS/2 波特率为更快的串口通信。 模式 1: 该位检查有效停止位。 0 = 无论停止位的逻辑电平, 接收总是有效的。 1 = 只有当接收的停止位是逻辑1并且接收的数据匹配“GIVEN”或“广播”地址时, 接收才有效。 模式 2 或 3: 为多机通讯 0 = 不管第九位的逻辑电平, 接收总是有效的。 1 =只有当接收的第九位是逻辑1并且接收的数据匹配“GIVEN”或“广播”地址时, 接收才有效。
4	REN	接收使能 0 = 串口0接收禁止。 1 = 工作模式为1, 2, 3时, 使能接收。工作模式为0时, 在条件 REN = 1 和 RI = 0时接收被初始化。
3	TB8	发送的第九位 串口0在模式2, 3时, 此位是发送的第9位, 模式0, 1不用此位。
2	RB8	接收的第九位 比位识别串行端口0模式2或3中的第九接收比特的逻辑电平。在模式1中, RB8是所接收的停止位的逻辑电平。SM2位逻辑电平1时例外。在模式0中不使用RB8。
1	TI	发送中断标志 模式0下该标志在发送完8位数据后由硬件置位, 其它模式下在发送到数据最后一位后由硬件置位。 当串口0中断使能, 将执行中断服务程序。该位必须由软件来清除

位	名称	描述
0	RI	接收中断标志 模式0下数据接收帧收到第八位后由硬件置位，模式1，2或3中接收到停止位后由硬件置位。SM2位逻辑电平1时例外。当串口0中断使能，将执行串口0中断服务程序。该位必须由软件清除。

SBUF – 串口 0 数据缓存

寄存器	SFR 地址	复位值
SBUF	99H, 页 0	0000_0000 b

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

位	名称	描述
7:0	SBUF[7:0]	<p>串口0 数据缓存</p> <p>这个寄存器实际上该地址上有2个独立寄存器. 一个用于接收数据, 一个用于发送数据。当数据写入 SBUF, 这是个发送寄存器并且移位进行串口发送。当数据从SBUF读出, 数据来自接收寄存器。每次向SBUF写入一字节数据, 将启动一次数据传输。</p>

SBUF_1 – 串口 1 数据缓存

寄存器	SFR 地址	复位值
SBUF_1	9AH, 页 0	0000_0000 b

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

位	名称	描述
7:0	SBUF_1[7:0]	<p>串口1数据缓存</p> <p>这个寄存器实际上该地址上有2个独立寄存器。一个用于接收数据，一个用于发送数据。当数据写入SBUF_1，这是个发送寄存器并且移位进行串口发送。当数据从SBUF_1读出，数据来自接收寄存器。</p> <p>每次向SBUF_1写入一字节数据，将启动一次数据传输。</p>

EIE – 扩展中断使能寄存器

寄存器	SFR 地址	复位值
EIE	9BH, 页 0	0000_0000 b

7	6	5	4	3	2	1	0
ET2	ESPI	EFB	EWDT	EPWM0	ECAP	EPI	EI ² C
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 = 定时器2中断使能. 当中断产生时TF2 (T2CON.7) 置1
6	ESPI	SPI中断使能 0 = SPI 中断禁止 1 = SPI 中断使能. 当中断产生时SPIF (SPSR.7), SPIOVF (SPSR.5), 或MODF (SPSR.4) 置 1
5	EFB	故障刹车中断使能 0 = 故障刹车中断禁止. 1 = 故障刹车中断使能. 当中断产生时FBF (PWM0FBD.7) 置 1.
4	EWDT	看门狗中断使能 0 = 看门狗中断禁止. 1 = 看门狗中断使能. 当中断产生时WDTF (WDCON.5) 置 1.
3	EPWM0	PWM0中断使能 0 = PWM 中断禁止. 1 = PWM 中断使能. 当中断产生时PWMF (PWM0CON0.5) 置 1
2	ECAP	输入捕获中断使能 0 = 输入捕获中断禁止. 1 = 输入中断使能. 当中断产生时CAPF[2:0] (CAPCON0[2:0]) 置 1.
1	EPI	引脚中断使能 0 = 引脚中断禁止. 1 = 引脚中断使能. 当中断产生时PIF 相关位置 1.
0	EI ² C	I²C 中断使能 0 = I ² C 中断禁止. 1 = I ² C 中断使能. 当中断产生时SI (I2CCON.3) 或 I2TOF (I2CTOC.0) 置 1.

EIE1 – 扩展中断使能寄存器 1

寄存器	SFR 地址	复位值
EIE1	9CH, 页 0	0000_0000 b

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

位	名称	描述
7:6	-	保留
5	EPWM3	PWM3 中断使能 0 = PWM3 中断关闭 1 = PWM3中断使能, 当中断产生时 PWM3F (PWM3CON0.5) 置1
4	EPWM2	PWM2 中断使能 0 = PWM2 中断关闭 1 = PWM2中断使能, 当中断产生时PWM2F (PWM2CON0.5) 置1
3	EPWM1	PWM1 中断使能 0 = PWM1 中断关闭 1 = PWM1中断使能, 当中断产生时 PWM1F (PWM1CON0.5) 置1
2	EWKT	WKT中断使能 0 = WKT 中断禁止. 1 = WKT 中断使能. 当中断产生时 WKTF (WKCON.4) 置 1
1	ET3	定时器3中断使能 0 = 定时器 3 中断禁止. 1 = 定时器 3中断使能. 当中断产生时 TF3 (T3CON.4) 置 1
0	ES_1	串口1中断使能 0 = 串口 1 中断禁止. 1 = 串口 1 中断使能. 当中断产生时 TI_1 (S1CON.1) 或 RI_1 (S1CON.0) 置 1

RSR – 复位标志寄存器

寄存器	SFR 地址	复位值
RSR	9DH, 页 0	00XX_XXXX b

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

位	名称	描述
7:6	-	保留
5	HardF	从 AUXR1.5 镜像 写 AUXR1.5=0 或 RSR.5=0清除
4	POF	从 PCON.4 镜像 写PCON.4=0 或 RSR.4=0清除
3	RSTPINF	从 AUXR1.6 镜像 写AUXR1.6=0 或 RSR.3=0清除
2	BORF	从 BODCON0.1 镜像 写 BODCON0.1=0 或 RSR.2=0清除
1	WDTRF	从 WDCON.3 镜像 写 WDCON.3=0 或 RSR.1=0清除
0	SWRF	从 AUXR1.7 镜像 写 AUXR1.7=0 或 RSR.0=0清除

CHPCON – 芯片控制

寄存器	SFR 地址	复位值
CHPCON	9FH, 页 0, TA 保护	软件复位 0000_00U0 b 其他 0000_00C0 b

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

位	名称	描述
7	SWRST	软件复位 对该位写1，芯片执行软件复位，复位完成后该位自动清零
6	IAPFF	IAP 错误标志 满足以下任意条件，硬件将在IAPGO (IAPTRG.0)置位后置位该位 (1) 访问地址越界。 (2) IAPCN 命令无效。 (3) IAP 擦除或者编程更新没有使能的区域。 (4) 当BOIAP (CONFIG2.5) =1, BODEN (BODCON0.7) =1, BORST (BODCON0.2) =0时，在V _{BOD} 电压下进行IAP 擦除和编程工作。 此位必须由软件清除。
5:2	-	保留
1	BS	启动选择 定义复位后MCU由哪块启动 0 =由APROM启动 1 =由LDROM启动
0	IAPEN	IAP 使能 0 = 禁止IAP。 1 =使能 IAP。 当打开IAP功能时，HIRC会被打开用于时序控制。为了降低功耗时停止内部晶振，IAP操作完后才执行清除IAPEN标志指令。

AUXR1 – 辅助寄存器 1

寄存器	SFR 地址	Reset Value
AUXR1	A2H, 页0	POR: 0000 0000b, 软件复位: 1U00 0000b, nRESET 外部复位: U100 0000b, 其他: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS
读/写	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, 该位硬件置1。通过软件清零。
6	RSTPINF	外部复位标志位 通过外部复位引脚复位MCU后, 该位将被硬件置1, 建议复位发生后通过软件清零
5	HardF	硬件故障复位标志 一旦程序计数器(PC)溢出flash地址空间EHFI (EIE1.4)=0, MCU将复位并且HardF硬件置位。通过软件清零 注意: 当MCU运行在OCD调试模式下并且OC DEN=0, 硬件故障复位将被禁用, 仅仅HardF置位
4	SLOW	ADC 低速选择 该位用于选择ADC的低速模式 0 = ADC 高速转换时间是1400ns, 转换频率 $F_{ADC} = 714$ ksp/s 1 = ADC 低速转换时间是4750ns, 转换频率 $F_{ADC} = 215$ ksp/s, 注1: 当VDD低于2.5V时, 采用低速ADC采样可以改善ADC转换性能。 注2: AUXR1.4或 ADCCON3 .0置 1, ADC 进入低速模式。
3	GF2	通用标志 2 可由用户软件置位或清0的通用标志。
2	UART0PX	串口0引脚交换 0 = 默认的RXD 分配到 P0.7引脚, TXD分配到P0.6引脚。 1 = RXD分配到 P0.6引脚, TXD 分配到 P0.7引脚。 注: 当此位置1或清0时, TXD 和 RXD 引脚立即交换。用户必须注传输或接收数据时不能交换引脚。否则可能产生不可预知的结果并且没有警告提醒。
1	0	保留 此位读出值一直为0。

位	名称	描述
0	DPS	数据指针选择 0 = 默认的数据指针 0(DPTR)是有效地。 1 = 数据指针1 (DPTR1) 是有效地。 DPS切换激活数据指针后，原来失效数据指针保持原始值不变。

AUXR2 – 辅助寄存器 2

寄存器	SFR 地址	复位值
AUXR2	A1H, 页 2	0000_0000 b

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

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

AUXR3 – 辅助寄存器 3

寄存器	SFR 地址	复位值
AUXR3	A2H, 页2	0000_0000 b

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

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

AUXR4 – 辅助寄存器 4

寄存器	SFR 地址	复位值
AUXR4	A3H, 页 2	0000_0000 b

7	6	5	4	3	2	1	0
PWM2_CH1P		PWM2_CH0P		PWM1_CH1P		PWM1_CH0P	
读/写		读/写		读/写		读/写	

位	名称	描述
7:6	PWM2_CH1P	PWM2通道1引脚选择 00 = PWM2_CH1 由 P3.0 引脚输出 01 = PWM2_CH1 由P3.1引脚输出 10 = PWM2_CH1 由P0.0引脚输出 11 = PWM2_CH1 由P0.4引脚输出
5:4	PWM2_CH0P	PWM2 通道0引脚选择 00 = 保留, 缺省值 01 = PWM2_CH0 由P2.1引脚输出 10 =PWM2_CH0 由P1.0引脚输出 11 =PWM2_CH0 由 P0.5引脚输出
3:2	PWM1_CH1P	PWM1通道1引脚选择 00 =保留, 缺省值 01 = PWM1_CH1 引脚选 P2.2引脚输出 10 = PWM1_CH1 由选 P1.4引脚输出 11 = PWM1_CH1 由选 P1.1引脚输出
1:0	PWM1_CH0P	PWM1通道0引脚选择 00 =保留, 缺省值 01 = PWM1_CH0 引脚选 P2.3引脚输出 10 = PWM1_CH0 由选P1.2引脚输出 11 =保留

AUXR5 – 辅助寄存器 5

寄存器	SFR 地址	复位值
AUXR5	A4H, 页 2	0000_0000 b

7	6	5	4	3	2	1	0
CLOP	T0P	-	-	PWM3_CH1P		PWM3_CH0P	
读/写	读/写	-	-	读/写		读/写	

位	名称	描述
7	CLOP	CLK输出引脚选择 0 = 当CLO输出使能时, CLO从P1.1引脚输出 1 = 当CLO输出使能时, CLO 从P3.引脚输出
6	T0P	T0引脚选择 0 = T0由P0.5引脚输出 1 = T0由P2.4引脚输出
5:4	-	保留
3:2	PWM3_CH1P	PWM3 通道1引脚选择 00 = 保留, 缺省值 01 = PWM3_CH1 由 P3.4 引脚输出 10 = PWM3_CH1 由 P1.5引脚输出 11 = PWM3_CH1 由 P0.3 引脚输出
1:0	PWM3_CH0P	PWM3通道0引脚选择 00 =保留, 缺省值 01 = PWM3_CH0由P3.2引脚输出 10 = PWM3_CH0由P0.1引脚输出 11 = 保留

AUXR6 – 辅助寄存器 6

寄存器	SFR 地址	复位值
AUXR6	A5H, 页 2	0000_0000 b

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

位	名称	描述
7:5	-	保留
4	UART4DG	UART4 RX 去毛刺控制 1: 去毛刺使能 0: 去毛刺禁止
3	UART3DG	UART3 RX去毛刺控制 1: 去毛刺使能 0: 去毛刺禁止
2	UART2DG	UART2 RX去毛刺控制 1: 去毛刺使能 0: 去毛刺禁止
1	UART1DG	UART1 RX去毛刺控制 1: 去毛刺使能 0: 去毛刺禁止
0	UART0DG	UART0 RX 去毛刺孔控制 1: 去毛刺使能 0: 去毛刺禁止

AUXR7 – 辅助寄存器 7

寄存器	SFR 地址	复位值
AUXR7	A6H, 页 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	SPI0NSSP		SPI0MOSIP	SPI0MISOP	SPI0CKP
-	-	-	读/写		读/写	读/写	读/写

位	名称	描述
7:5	-	保留
4:3	SPI0NSSP	SPI0_SS 引脚选择 00 = SPI0_SS 选择 P1.5 01 = 保留 10 = SPI0_SS选择P3.5 11 = 保留
2	SPI0MOSIP	SPI0_MOSI 引脚选择 0 = SPI0_MOSI 选择 P0.0 1 = SPI0_MOSI 选择 P3.0
1	SPI0MISOP	SPI0_MISO 引脚选择 0 = SPI0_MISO 选择 P0.1 1 = SPI0_MISO 选择 P2.5
0	SPI0CKP	SPI0_CLK 引脚选择 0 = SPI0_CLK 选择 P1.0 1 = SPI0_CLK 选择 P1.7

AUXR8 – 辅助寄存器 8

寄存器	SFR 地址	复位值
AUXR8	A7H, 页 2, TA 保护	0000_0000 b

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

位	名称	描述										
7:4	CLODIV[3:0]	时钟输出分频器 根据CLODIV的值，系统输出时钟计算公式如下： 当CLODIV = 00H时， $F_{CLO} = F_{SYS}$ 当CLODIV = 01H 到0FH时， $F_{CLO} = \frac{F_{SYS}}{2 \times CLODIV}$										
3-0	CKTESTOEN[3:0]	时钟输出CKO的时钟源 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>CKTESTOEN</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>HIRC (24 MHz) 输出到CKO 引脚</td> </tr> <tr> <td>[1]</td> <td>OSCIN 输出到CKO 引脚</td> </tr> <tr> <td>[2]</td> <td>LIRC (10 kHz) 输出到 CKO引脚</td> </tr> <tr> <td>[3]</td> <td>HXT (4~24 MHz)输出到CKO引脚</td> </tr> </tbody> </table>	CKTESTOEN	功能	[0]	HIRC (24 MHz) 输出到CKO 引脚	[1]	OSCIN 输出到CKO 引脚	[2]	LIRC (10 kHz) 输出到 CKO引脚	[3]	HXT (4~24 MHz)输出到CKO引脚
CKTESTOEN	功能											
[0]	HIRC (24 MHz) 输出到CKO 引脚											
[1]	OSCIN 输出到CKO 引脚											
[2]	LIRC (10 kHz) 输出到 CKO引脚											
[3]	HXT (4~24 MHz)输出到CKO引脚											

BODCON0 – 欠压检测控制0

寄存器	SFR 地址	复位值
BODCON0	A3H, 页 0, TA 保护	POR, CCCC XC0X b BOD, UUUU XU1X b 其他, UUUU XUUX b

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

位	寄存器	描述
7	BODEN	欠压检测使能 0 = 禁用欠压检测电路 1 = 使能欠压检测电路 注意在开启该功能后需要2到3个LIRC时钟BOD才能正常工作
6	-	保留
5:4	BOV[1:0]	配置欠压检测电压选择位 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，该标志由硬件设置为逻辑1。如果EBOD(EIE.2)和EA(IE.7)都置位，将请求欠压检测中断。该位必须由软件清零。
2	BORST	欠压检测复位使能 该位决定在电源电压跌到以V _{BOD} 下时是否产生欠压检测复位 0 = 禁用欠压检测复位 1 = 使能欠压检测复位
1	BORF	欠压复位标志 当MCU发生欠压复位时，该位被硬件值1。建议通过软件清除该位。
0	BOS	欠压状态标志 在BOD电路开启时，该位反应V _{DD} 与V _{BOD} 比较情况。BOD电路关闭时保持为0。 0 = V _{DD} 电压大于V _{BOD} 或是BOD电路关闭 1 = V _{DD} 电压小于V _{BOD} 注该位为只读位

注:

1. 所有复位后BODEN、BOV[1:0]和BORST初始化的值是直接通过加载CONFIG2位7位6~4和位2决定
2. BOF复位后的值依据CONFIG2的设置和VDD的电平. 请参考表 6.2-1 BOF 复位值

IAPTRG – IAP触发寄存器

寄存器	SFR 地址	复位值
IAPTRG	A4H, 页 0, TA 保护	0000 _0000 b

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

位	名称	描述
7:1	-	保留
0	IAPGO	<p>IAP 执行</p> <p>该位设置成1，开始执行IAP。此指令之后，CPU保持程序计数器(PC)，IAP硬件管理进程。IAP操作完成后，程序计数器继续执行下一条指令。IAPGO位将被自动清除并一直保持0。</p> <p>触发IAP操作前，由于硬件限制，中断（如果启用）应该临时禁止。</p>

IAPUEN – IAP更新使能

寄存器	SFR 地址	复位值
IAPUEN	A5H, 页 0, TA 保护	0000 _0000 b

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

位	名称	描述
7:3	-	保留
2	CFUEN	CONFIG 升级使能 0 = 禁止通过IAP擦除或编程CONFIG。 1 = 允许通过IAP擦除或编程CONFIG。
1	LDUEN	LDROM 升级允许使能 0 = 禁止通过IAP擦除或编程LDROM。 1 = 允许禁止通过IAP擦除或编程LDROM 。
0	APUEN	APROM 升级允许使能 0 = 禁止通过IAP擦除或编程APROM。 1 = 允许禁止通过IAP擦除或编程APROM。

IAPAL – IAP字节地址低字节

寄存器	SFR 地址	复位值
IAPAL	A6H, 页 0	0000_0000 b

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

位	名称	描述
7:0	IAPA[7:0]	IAP 地址地低位地址低字节 IAPAL 包含IAP操作地址的低八位

IAPAH – IAP地址高字节

Register	SFR Address	Reset Value
IAPAH	A7H, Page 0	0000_0000 b

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

位	名称	描述
7:0	IAPA[15:8]	IAP 地址高位 IAPAH 包含IAP操作地址的高八位 。

IE - 中断使能寄存器

寄存器	SFR 地址	复位值
IE	A8H, 所有页, 可位寻址	0000_0000 b

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

位	名称	描述
7	EA	使能所有中断 该位全局使能/禁止所有那些单独使能的中断 0 = 禁止所有中断源 1 = 每个中断使能依靠使能单个中断。 如果使能, 将会产生相应的单个中断
6	EADC	使能ADC中断 0 = ADC 中断禁止. 1 = ADC 中断使能. 当中断产生时ADCF (ADCCON0. 7) 置1
5	EBOD	使能欠压中断 0 = 欠压检测中断禁止. 1 = 欠压检测中断使能. 当中断产生时BOF (BODCON0. 3) 置 1
4	ES	使能串口0中断 0 = 串口 0 中断禁止. 1 = 串口 0 中断使能. 当中断产生时TI (SCON. 1) 或 RI (SCON. 0) 置1
3	ET1	使能定时器1中断 0 = 定时器 1 中断禁止. 1 = 定时器 1 中断使能. 当中断产生时TF1 (TCON. 7) 置 1
2	EX1	使能外部中断1 0 = 外部中断1中断禁止. 1 = 外部中断1 ($\overline{INT1}$ 管脚 P1. 7) 中断使能. 当中断产生时, IE1 (TCON. 3) 置1
1	ET0	使能定时器0中断 0 = 定时器 0 中断禁止. 1 = 定时器 0 中断使能. 当中断产生时 TF0 (TCON. 5) 置 1
0	EX0	使能外部中断0 0 = 外部中断0中断禁止. 1 = 外部中断0 ($\overline{INT0}$ 管脚 P3. 0) 中断使能. 当中断产生时, IE0 (TCON. 1) 置1

SADDR – 从机 0 地址

寄存器	SFR 地址	复位值
SADDR	A9H, 页 0	0000_0000 b

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

位	名称	描述
7:0	SADDR[7:0]	从机 0 地址 该字节定义微控器自身的从机地址以用于串口0多机通信

WDCON – WDT控制

寄存器	SFR 地址	复位值
WDCON	AAH, 页 0, TA 保护	POR 0000_0111 b WDT 0000_1UUU b 其他 0000_UUUU b

7	6	5	4	3	2	1	0
WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W		

位	名称	描述
7	WDTR	WDT 运行 此位仅当 WDTE[3:0] (CONFIG4[7:4]) 全 1、WDT 配置为通用定时器时有效。 0 = WDT 禁止 1 = WDT 使能。WDT开始计数。
6	WDCLR	WDT 清0 此位写1用于把 WDT 清0，读写效果不同： 设置该位复位WDT计数到00H。它使计数器到一个已知的状态，防止系统出现不可预知的复位。 写和读WDCLR位意思是不一样的 写： 0 = 无效。 1 = WDT计数器清0。 读： 0 = WDT清0完成。 1 = WDT 清0未完成。
5	WDTF	WDT溢出标志 指示 WDT溢出。需软件清0
4	WIDPD	WDT空闲或掉电状态运行使能位 仅WDTE[3:0] (CONFIG4[7:4])全 1配置为通用定时器晨，此位才有效。 0 = WDT在空闲或掉电状态停止运行 1 = WDT 在空闲或掉电状态保持运行。
3	WDTRF	WDT复位标志 此位指示 CPU 被 WDT 复位，建议每次复位后，软件对此位清0
2:0	WDPS[2:0]	WDT 分频选择 选择分频系数从 1/1 到 1/256。详见表 6.7-1 WDT 定时时间表 缺省最大分频

注:

1. WDTRF 上电后清0，WDT复位后置1，其它复位不变。
2. WDPS[2:0] 上电后全1，其它复位后不变。

BODCON1 – 欠压检测控制1

寄存器	SFR地址	复位值
BODCON1	ABH, 页 0, TA 保护	POR 0000 0001 b 其他 0000 0UUU b

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

	名称	描述
7:3	-	保留
2:1	LPBOD[1:0]	<p>低功耗BOD使能</p> <p>00 = BOD 正常模式. BOD 电路总是开启.</p> <p>01 = BOD 低功耗模式 1, 每隔1.6ms周期性开启BOD电路</p> <p>10 = BOD 低功耗模式 2, 每隔6.4ms周期性开启BOD电路</p> <p>11 = BOD 低功耗模式 3, 每隔25.6ms周期性开启BOD电路</p>
0	BODFLT	<p>BOD 滤波器控制</p> <p>BOD具有一个滤波器, 当MCU以HIRC或ECLK作为系统时钟且BOD没有在低功率模式下 (LPBOD[1:0] = [0, 0])工作时, 该滤波器采样32个FSYS 时钟用以过滤电源噪声。在其他条件下, 滤波器采样2个LIRC。</p> <p>注:当CPU停在掉电模式时, BOD滤波计数一直是2个LIRC时钟</p> <p>BOD滤波器有效地避免电源噪声误触发BOD事件发生。设置该位可以开启或关闭BOD滤波功能。</p> <p>0 = BOD 滤波禁止.</p> <p>1 = BOD 滤波使能. (上电复位默认值.)</p>

IAPFD – IAP 数据寄存器

寄存器	SFR 地址	复位值
IAPFD	AEH, 页 0	0000_0000 b

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

位	名称	描述
7:0	IAPFD[7:0]	IAP 数据 该字节包含将要读出或者写进flash的数据。在编程模式下，用户需要在触发ISPIAP之前将数据写入IAPFD里；在读/校验模式下，在IAP完成后从APFD里读出数据。

IAPCN – IAP 控制寄存器

寄存器	SFR 地址	复位值
IAPCN	AFH, 0页	0011_0000 b

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

位	名称	描述
7:6	IAPB[1:0]	IAP 控制 此字节是IAP控制命令。详细描述见 表6.3-1 IAP模式及命令
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

PnM1 – 端口 n 模式选择1

寄存器	SFR 数据	复位值
P0M1	B1H, 页 0	1111_1111 b
P1M1	B3H, 页 0	1111_1111 b
P2M1	89H, 页 2	0011_1111 b
P3M1	ACH, 页 0	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 0 模式选择1

PnM2 – 端口模式选择2

寄存器	SFR 数据	复位值
P0M2	B2H, 页 0	0000_0000 b
P1M2	B4H, 页 0	0000_0000 b
P2M2	8AH, 页 2	0000_0000 b
P3M2	ADH, 页 0	0000_0000 b

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

位	名称	描述
7:0	PnM2[7:0]	端口 0 模式选择 2 注: PxM1和PxM2结合使用, 用来决定端口上每个引脚的模式。表 6.4-1 I/O 模式配置

每个 I/O 引脚都可以单独配置成 TTL 输入或施密特触发输入。注: 访问所有的 PxS 寄存器都需要将 SFR 页切换到页 1。

PnS – Port n 施密特触发输出

寄存器	SFR 数据	复位值
P0S	B1H, 页 1	0000_0000 b
P1S	B3H, 页 1	0000_0000 b
P2S	8CH, 页 2	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]	Pn施密特触发输入 0 = Pn.x引脚TTL电平输入 1 = Pn.x引脚施密特触发输入

TOE – 定时器0和1 输出使能

寄存器	SFR 数据	复位值
TOE	B5H, 页 0	0000_0000 b

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

位	名称	描述
7:4	-	保留
3	T1OE	定时器 1 输出使能 0 = 定时器1输出禁止。 1 = 定时器1输出到T1引脚。 注：只有运行在定时器1模式时，输出才能被使能。
2	T0OE	定时器 0 输出使能 0 = 定时器0输出禁止。 1 = 定时器0输出到T40引脚。 注：只有运行在定时器0模式时，输出才能被使能。
1:0	-	保留

IPH – 中断优先级高位^[2]

寄存器	SFR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
7	-	保留
6	PADC	ADC中断优先级高位
5	PBOD	掉电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： IP与IPH结合使用，以确定每个中断源的优先级。见 [表 6.2-4 中断优先级设置](#)

PWMnINTC – PWM 中断控制

寄存器	SFR地址	复位值
PWM0INTC	B7H, 页 1	0000_0000 b
PWM1INTC	B6H, 页 2	0000_0000 b
PWM2INTC	C6H, 页 2	0000_0000 b
PWM3INTC	D6H, 页 2	0000_0000 b

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

位	名称	描述
7:6	-	保留
5:4	INTTYP[1:0]	PWM 中断类型选择 00 = PWM channel 0/1/2/3/4/5 引脚的下沿. 01 = PWM channel 0/1/2/3/4/5 引脚的上沿. 10 = 周期中点. 11 = 周期末. 注意中点和周期末, 只有互补模式中心对齐模式有效。
3	-	保留
2:0	INTSEL[2:0]	PWM 中断通道选择 在PWM0/1/2/3/4/5脚, 当PWM已选择中断类型为上升沿或下降沿时, 该位段用以选择中断响应所相对的PWM脚。 000 = PWM_CH0. 001 = PWM_CH1. 010 = PWM_CH2. 011 = PWM_CH3. 100 = PWM_CH4. 101 = PWM_CH5. 其它 = PWM_CH0.

IP – 中断优先级^[1]

寄存器	SFR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
7	-	保留
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 中断优先级低位
1	PT0	定时器 0 中断优先级低位
0	PX0	外部中断 0 中断优先级低位

注: IP和 IPH结合使用来决定中断源的中断优先级。见 表 6.2-4 中断优先级设置。

SADEN – 从机0 地址掩码

寄存器	SFR地址	复位值
SADEN	B9H, 页 0	0000_0000 b

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

位	名称	描述
7:0	SADEN[7:0]	<p>从机0地址掩码.</p> <p>该字节为串口0掩码，为存储“Given”地址的“无关（定义为0）”位。无关位可使更多从机得以灵活运用。</p>

SADEN 1 – 从机 1 地址掩码

寄存器	SFR地址	复位值
SADEN_1	BAH, 页 0	0000_0000 b

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

位	名称	描述
7:0	SADEN_1[7:0]	<p>从机1地址掩码.</p> <p>该字节为串口1掩码为存储“Given”地址的“无关（定义为0）”位。无关位可使更多从机得以灵活运用。</p>

SADDR_1 – 从机 1 地址

寄存器	SFR地址	复位值
SADDR_1	BBH, 页 0	0000_0000 b

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

位	名称	描述
7:0	SADDR_1[7:0]	从机1地址 该字节定义微控器自身的从机地址以用于串口1多机通信.

I2DAT – I²C数据

寄存器	SFR地址	复位值
I2DAT	BCH, 页 0	0000_0000 b

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

位	名称	描述
7:0	I2DAT[7:0]	<p>I²C数据</p> <p>该寄存器存放准备发送的，或接收到的数据。只要SI = 1，此数据就有效。在I2C数据传输过程中，I2CnDAT读写的结果是不可预测的。</p> <p>当I2CnDAT移出时，总线上的数据同步移入更新I2CnDAT。I2CnDAT总是显示为从总线上取出的最后一个字节。所以总线仲裁失败时，I2CnDAT原来的值在传输后已经改变。</p>

I2STAT – I²C 状态

寄存器	SFR地址	复位值
I2STAT	BDH, 页 0	1111 1000 b

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

位	名称	描述
7:3	I2STAT[7:3]	I ² C 状态码 I2CnSTAT高5位为状态码，共有27种值。I2CnSTAT = F8H时，表示空闲，SI 将保持为0。其它26种状态，都会让SI置1，且产生中断请求。
2:0	0	保留 I2STAT 的低3位总是0

I2CLK – I²C 时钟

寄存器	SFR 地址	复位值
I2CLK	BEH, 页 0	0000_1001 b

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

位	名称	描述
7:0	I2CLK[7:0]	<p>I²C 时钟设定</p> <p><u>主机模式:</u> 该寄存器设定作主机时 I²C 总线时钟速率。算式如下:</p> $\frac{F_{SYS}}{4 \times (I2CLK + 1)}$ <p>注如果 I²C 的外设时钟频率是 16MHz，默认状态下，I²C 的总线频率为 400Kbps。I2CCLK 值写入 00H 及 01H 无效，实行了限制。</p> <p><u>从机模式:</u> 该字节无效，从机自动同步主机时钟，最高 400kbps。</p>

I2TOC – I²C 超时计数器

寄存器	SFR 地址	复位值
I2TOC	BFH, 页 0	0000_0000 b

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

位	名称	描述
7:3	-	保留
2	I2TOCEN	I ² C 超时计数器使能 0 = I ² C 超时计数器禁止. 1 = I ² C 超时计数器使能. 注意: 使能 I ² C 超时计数器功能后请使能 I ² C 中断
1	DIV	I ² C 超时计数器时钟除频 0 = I ² C 超时计数器时钟源是 F _{sys} /1. 1 = I ² C 超时计数器时钟源是 F _{sys} /4.
0	I2TOF	I ² C 超时标志 这个标志当 14-bit I ² C 超时计数器溢出时由硬件置1, 由软件清0

I2CON – I²C 控制器

寄存器	SFR 地址	复位值
I2CON	C0H, 所有页, 位可寻址	0000_0000 b

7	6	5	4	3	2	1	0
I	I2CEN	STA	STO	SI	AA	-	I2CPX
R/W	读/写	读/写	读/写	读/写	读/写	-	读/写

位	名称	描述
7	I	I2C0 保持时间延长使能 0 = 禁止I ² C DATA 到 I2C0_SCL的 保持时间延长 1 = 使能I ² C DATA 到 I2C0_SCL 的保持时间延长, 延长8个系统时钟
6	I2CEN	I2C0 保持时间延长使能 0 = 禁止I ² C DATA 到 I2C0_SCL的 保持时间延长 1 = 使能I ² C DATA 到 I2C0_SCL 的保持时间延长, 延长8个系统时钟 I2C 总线使能 0 = I2C 禁止. 1 = I2C 使能. 使能 I2C之前, SCL和SDA必须配置为输出1
5	STA	I ² C 总线使能 0 = I ² C 总线禁止. 1 = I ² C 总线使能. 在使能 I ² C之前, I2C0_SCL 和 I2C0_SDA 端口电平需要为高 起始标志START 当STA置1, 如果总线空闲, I ² C产生START信号, 如果总线忙, I ² C等待停止条件STOP, 然后产生START信号 如果总线已经在总机模式且已发送一个或多个字节, 此时再设定STA, I ² C 总线将产生重复开始信号 repeated START 注: STA可在任何时间置1, 包括从机模式。但硬件不会在发送START或repeat START信号后自动清0。用户需软件清除。

位	名称	描述
4	STO	<p>起始标志START 当STA置1，如果总线空闲，I²C产生START信号，如果总线忙，I²C等待停止条件STOP，然后产生START信号 如果总线已经在总机模式且已发送一个或多个字节，此时再设定STA，I²C总线将产生重复开始信号repeated-START 注：STA可在任何时间置1，包括从机模式。但硬件不会在发送START或repeat-START信号后自动清0。用户需软件清除。</p> <p>停止标志STOP I²C总线在主机模式下设定STO为1，将会向总线发送停止信号STOP。一旦总线上停止条件完成，STO由硬件自动清0。 当总线上产生错误状态(I2CnSTAT为00H) STO也会置1。这种情况下总线不会发送停止位。 如果STA和STO同时置1，且在主机模式下，I²C总线在发送STOP后马上发送START。如果在从机模式下，应避免STA及STO同时置1，以避免发出非法I²C帧</p>
3	SI	<p>停止标志STOP I²C总线在主机模式下设定STO为1，将会向总线发送停止信号STOP。一旦总线上停止条件完成，STO由硬件自动清0。 当总线上产生错误状态(I2CnSTAT为00H) STO也会置1。这种情况下总线不会发送停止位。 如果STA和STO同时置1，且在主机模式下，I²C总线在发送STOP后马上发送START。如果在从机模式下，应避免STA及STO同时置1，以避免发出非法I²C帧</p> <p>I²C中断标志 I²C所有26种状态中出现一种，硬件就会置1此位(F8H除外)，此时软件根据读取I2STAT值，来确认哪步已经执行和决定下一步动作。 SI由软件清0。在SI被清0之前，I2C0_SCL低电平周期延长，传输暂停，该状态对于从机处理接收到的数据非常有用，可以确保准确处理前一数据再接收下一个数据。 SI位被软件清0后，I²C外设才会继续下一步：SI清除后，总线将继续产生START和repeat-START条件，STOP条件，8位数据，或者其他控制字节或位。因此，在SI软件清除前，软件应该准备好合适的寄存器设置。</p>
2	AA	<p>I²C中断标志 I²C所有26种状态中出现一种，硬件就会置1此位(F8H除外)，此时软件根据读取I2STAT值，来确认哪步已经执行和决定下一步动作。 SI由软件清0。在SI被清0之前，I2C0_SCL低电平周期延长，传输暂停，该状态对于从机处理接收到的数据非常有用，可以确保准确处理前一数据再接收下一个数据。 SI位被软件清0后，I²C外设才会继续下一步：SI清除后，总线将继续产生START和repeat-START条件，STOP条件，8位数据，或者其他控制字节或位。因此，在SI软件清除前，软件应该准备好合适的寄存器设置。</p> <p>应答响应标志 若AA = 1，当I2C器件是一个接收器或者是个地址匹配的从机时，在I2C0_SCL上的应答时钟脉冲期间，I2C0_SDA将应答ACK（I2C0_SDA低电平） 若AA = 0，当I2C器件是一个接收器或者是个地址匹配的从机时，在I2C0_SCL上的应答时钟脉冲期间，I2C0_SDA将应答NACK（I2C0_SDA高电平）。若器件自我清除AA标志位，则会清除其从机地址或忽略广播呼叫，SI不会置位，中断不予产生。 注：若已被寻址的从机，在从机接收模式下未回复应答位或在从机发送模式下未接收到应答位，该从机将变为未寻址从机，无法接收数据直到其AA被置1，且重新被主机定址。 特殊情况注意：从机发送模式时，I2CnSTAT若为C8H，从机发送最后一个字节给主机之前，清除AA，发送完最后一个字节的位后，从机将变成未被定址的从机模式，和主机断开。主机若再从总线上读数据，将得到FFH。</p>

位	名称	描述
0	I2CPX	<p>应答响应标志</p> <p>若AA = 1, 当I2C器件是一个接收器或者是个地址匹配的从机时, 在I2C0_SCL上的应答时钟脉冲期间, I2C0_SDA将应答ACK (I2C0_SDA低电平)。</p> <p>若AA = 0, 当I2C器件是一个接收器或者是个地址匹配的从机时, 在I2C0_SCL上的应答时钟脉冲期间, I2C0_SDA将应答NACK (I2C0_SDA高电平)。若器件自我清除AA标志位, 则会清除其从机地址或忽略广播呼叫, SI不会置位, 中断不予产生。</p> <p>注: 若已被寻址的从机, 在从机接收模式下未回复应答位或在从机发送模式下未接收到应答位, 该从机将变为未寻址从机, 无法接收数据直到其AA被置1, 且重新被主机定址。</p> <p>特殊情况注意: 从机发送模式时, I2CnSTAT若为C8H, 从机发送最后一个字节给主机之前, 清除AA, 发送完最后一个字节的位后, 从机将变成未被定址的从机模式, 和主机断开。主机若再从总线上读数据, 将得到FFH。</p> <p>I2C 引脚选择</p> <p>0 = 分配 SCL 是 P1.3 和 SDA 是 P1.4.</p> <p>1 = 分配 SCL 是 P0.2 和 SDA 是 P1.6.</p> <p>注意: 一旦该位置位或清0, I2C引脚会立刻更换。</p>

I2ADDR – I²C 本机从机地址

寄存器	SFR 地址	复位值
I2ADDR	C1H, 页 0	0000_0000 b

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

位	名称	描述
7:1	I2ADDR[7:1]	<p>I²C 本机从机地址</p> <p>主机模式: 无效</p> <p>从机模式: 存放7位从机地址。主机需要定址该从机, 需通过在START或repeat START之后的第一个字节值地址信息与此地址相同。如果AA为1, 该从机响应主机, 成为被定址从机。否则主机呼叫地址会被忽略。</p> <p>注: I2CnADDRx [7:1] 不能写为全0, 因为0x00为广播呼叫方式寻址专用。.</p>
0	GC	<p>广播呼叫位</p> <p>主机模式: 无效</p> <p>从机模式: 0 = 广播呼叫模式忽略, 不响应。 1 = 如果AA置1, 参与广播呼叫模式, 若AA清0, 忽略广播呼叫。</p>

ADCRL – ADC转换结果低字节

寄存器	SFR 地址	复位值
ADCRL	C2H, 页 0	0000_0000 b

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

位	名称	描述
7:4	-	保留
3:0	ADCR[3:0]	ADC转换结果低字节 ADC 转换结果低4位存入这个寄存器

ADCRH – ADC转换结果高字节

寄存器	SFR 地址	复位值
ADCRH	C3H, 页 0	0000_0000 b

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

位	名称	描述
7:0	ADCR[11:4]	ADC转换 结果高字节 ADC 转换结果的高8位存入这个寄存器。

T3CON – 定时器 3 控制

寄存器	SFR 地址	复位值
T3CON	C4H, 页 0	0000_0000 b

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

位	名称	描述
7	SMOD_1	串行口1波特率加倍使能。 当串口1在模式2下波特率加倍使能。见表 6.9-2 串口 UART1 模式/波特率描述。
6	SMOD0_1	串口1帧错误访问使能 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 当串口0处于1、3模式时，该位选择哪个定时器用作波特率时钟源。 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，该位被置1。当程序执行定时器3中断服务子程序时，它被硬件自动被清0。该位也可以软件置位或清0。
3	TR3	定时器 3 运行孔子和控制 0 = 定时器 3停止。 1 = 定时器 3 开始运行。 注：重载寄存器R3H和R3L仅在定时器3停止（TR3为0）的时候才可以被写。如果TR3 为1时写入RH3或RI3，结果是不可预知的。
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 重载值低字节

寄存器	SFR 地址	复位值
RL3	C5H, 页 0	0000_0000 b

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

位	名称	描述
7:0	RL3[7:0]	定时器 3 重载值低字节 保持定时器3的重载值低字节。

RH3 – 定时器 3 重载值高字节

寄存器	SFR 地址	复位值
RH3	C6H, 页 0	0000_0000 b

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

位	名称	描述
7:0	RH3[7:0]	定时器 3 重载值高字节 保持定时器3重载值高字节

PIOCON1 – PWM 引脚控制器

寄存器	SFR 地址	复位值
PIOCON1	C6H, 页 1	0000_0000 b

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

位	名称	描述
7	PIO17	P1. 7/PWM引脚功能选择 0 = P1. 7/PWM3_CH0 引脚功能为 GPIO P1. 7. 1 = P1. 7/PWM3_CH0 引脚功能为 PWM3_CH0输出
6	-	保留
5	PIO15	P1. 5/PWM引脚功能选择 0 = P1. 5/PWM 引脚功能为 P1. 5. 1 = P1. 5/PWM 引脚功能为 PWM0_CH5 / PWM3_CH1输出 当 AUXR5[3:2].PWM3CH1P = 11, PIO15 = 1 , P1. 5定义为PWM3_CH1. 当 AUXR5[3:2].PWM3CH1P = 00 ,PIO15 = 1, P1. 5定义为PWM0_CH5.
4	-	保留
3	PIO04	P0. 4/PWM引脚功能选择 0 = P0. 4/PWM 引脚功能为 P0. 4. 1 = P0. 4/PWM 引脚功能为 PWM0_CH3 / PWM2_CH1输出 当 AUXR4[7:6].PWM2CH1P = 11, PIO04 = 1 , P1. 4 定义为 PWM2_CH1. 当 AUXR4[7:6].PWM2CH1P = 00 ,PIO04 = 1, P1. 4定义为PWM0_CH3.
2	PIO05	P0. 5/PWM pin function select 0 = P0. 5/PWM 引脚功能为 P0. 5. 1 = P0. 5/PWM引脚功能为 PWM0_CH2 / PWM2_CH0输出 (PWM2CH0P=11, 选 PWM2_CH0; 其它选 PWM0_CH32)
1	PIO14	P1. 4/PWM 引脚功能选择 0 = P1. 4/PWM 引脚功能为 P1. 4. 1 = P1. 4/PWM 引脚功能为 PWM0_CH1 / PWM1_CH1 输出. (PWM1_CH1P=10, 选 PWM1_CH1; 其它选PWM0_CH1)
0	-	保留

PIOCON2 – PWM引脚控制器

寄存器	SFR 地址	复位值
PIOCON2	B7H, 页 2	0000_0000 b

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

位	名称	描述
7	PIO34	P3. 4/PWM3_CH1 引脚功能选择 0 = P3. 4/PWM3_CH1引脚功能为P3. 4. 1 = P3. 4/PWM3_CH1引脚功能为PWM3_CH1 输出。
6	PIO33	P3. 3/PWM0_CH1 引脚功能选择 0 = P3. 3/PWM0_CH1 引脚功能为P3. 3. 1 = P3. 3/PWM0_CH1 引脚功能为PWM0_CH1输出。
5	PIO32	P3. 2/PWM3_CH0 引脚功能选择 0 = P3. 2/PWM3_CH0引脚功能为P3. 2. 1 = P3. 2/PWM3_CH0引脚功能为PWM3_CH0输出。
4	PIO31	P3. 1/PWM2_CH1 引脚功能选择 0 = P3. 1/PWM2_CH1 pin functions as 引脚功能为P3. 1. 1 = P3. 1/PWM2_CH1 引脚功能为PWM2_CH1输出。
3	PIO30	P3. 0/PWM2_CH1 引脚功能选择 0 = P3. 0/PWM2_CH1 pin functions as 引脚功能为 P3. 0 1 = P3. 0/PWM2_CH1 引脚功能为PWM2_CH1输出。
2	PIO23	P23/PWM1_CH0 引脚功能选择 0 = P2. 3/PWM1_CH0 pin functions as 引脚功能为 P2. 3. 1 = P2. 3/PWM1_CH0 引脚功能为PWM1_CH0输出。
1	PIO22	P2. 2/PWM1_CH1 引脚功能选择 0 = P2. 2/PWM1_CH1 引脚功能为 P2. 2. 1 = P2. 2/PWM1_CH1 引脚功能为 PWM1_CH1输出。
0	PIO21	P2. 1/ PWM2_CH0 引脚功能选择 0 = P2. 1/ PWM2_CH0 引脚功能为 P2. 1. 1 = P2. 1/ PWM2_CH0 引脚功能为PWM2_CH0输出。

TA – 时钟控制访问保护

寄存器	SFR 地址	复位值
TA	C7H, 页 0	0000_0000 b

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

位	名称	描述
7:0	TA[7:0]	时钟控制访问保护 TA寄存器控制着对被保护的SFRs的访问权限。当访问被保护位时，用户先对TA寄存器写入AAH，接着是55H，当写完这两条后，才可以有4个时钟周期的时间对具有时控保护的SFR寄存器写入数据。 .

T2CON – 定时器 2 控制

寄存器	SFR 地址	复位值
T2CON	C8H, 所有页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
TF2	-	-	-	-	TR2	-	CM/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溢出或发生比较匹配时，该位置1。如果定时器2中断和全局中断被启用，当该位被置1时，CPU将执行定时器2中断服务子程序。该位不能被硬件自动清0，需要软件清0。
6:3	-	保留
2	TR2	定时器 2运行控制 0 = 定时器 2 禁止。清除该位将停止定时器2，当前计数值保存在TH2和TL2里。 1 = 定时器 2 使能。
1	-	保留
0	CM/RL2	定时器 2比较模式或自动重载模式选择 该位选择定时器2模式。 0 = 自动重载模式 1 = 比较模式

T2MOD – 定时器 2 模式

寄存器	SFR 地址	复位值
T2MOD	C9H, 页 0	0000_0000 b

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

位	名称	描述
7	LDEN	使能自动重载 0 = 禁止重载 RCMP2H 和 RCMP2L 到 TH2 和 TL2. 1 = 使能重载 RCMP2H 和 RCMP2L 到 TH2 和 TL2.
6:4	T2DIV[2:0]	定时器 2 时钟分频器 000 = 定时器2 时钟分频器为 1/1. 001 = 定时器2 时钟分频器为 1/4. 010 = 定时器2 时钟分频器为 1/16. 011 = 定时器2 时钟分频器为 1/32. 100 = 定时器2 时钟分频器为 1/64. 101 = 定时器2 时钟分频器为 1/128. 110 = 定时器2 时钟分频器为 1/256. 111 = 定时器2 时钟分频器为 1/512.
3	CAPCR	捕获模式自动清除 此位仅在定时器2 自动重载模式下有效。当发生捕获事件时，它使硬件能够在 TH2 和 TL2 计数器寄存器被传输到 RCMP2H 和 RCMP2L 之后自动清除它们。 0 = 当发生捕获事件时，定时器 2 继续计数。 1 = 当发生捕获事件时，定时器 2 的计数值被自动清除。
2	CMPCR	比较匹配自动清除 该位仅当定时器2 在比较功能模式下有效。当比较符合后，硬件将自动清除 TH2 及 TL2 计数器值。 0 = 比较符合之后，定时器2 计数值按之前值继续计数。 1 = 比较符合之后，定时器2 计数值清0。
1:0	LDTS[1:0]	自动重载触发选择 这些位用来选择重载触发事件。 00 = 定时器2 溢出时重载 01 = 捕获0 事件发生时重载 10 = 捕获01 事件发生时重载 11 = 捕获02 事件发生时重载

RCMP2L – 定时器 2 重载/比较值低字节

寄存器	SFR 地址	复位值
RCMP2L	CAH, 页 0	0000_0000 b

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

位	名称	描述
7:0	RCMP2L[7:0]	定时器 2 重载/比较低字节 当定时器2配置成比较模式时，这些位保存比较值低字节。 当定时器2配置成重载模式时，这些位保存重载值低字节。

RCMP2H – 定时器 2 重载/比较值高字节

寄存器	SFR 地址	复位值
RCMP2H	CBH, 页 0	0000_0000 b

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

位	名称	描述
7:0	RCMP2H[7:0]	定时器 2 重载/比较值高字节 当定时器 2 配置成比较模式时，这些位保存比较值高字节。 当定时器 2 配置成重载模式时，这些位保存重载值高字节。

TL2 – 定时器 2 低字节

寄存器	SFR 地址	复位值
TL2	CCH, 页 0	0000_0000 b

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

位	名称	描述
7:0	TL2[7:0]	定时器2 低字节 TL2是定时器2 16位计数器的低字节。

TH2 – 定时器 2 高字节

寄存器	SFR 地址	复位值
TH2	CDH, 页 0	0000_0000 b

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

位	名称	描述
7:0	TH2[7:0]	定时器 2 高字节 TH2是定时器2 16位计数器的高字节。

ADCMP1 – ADC 表值比较值低字节

寄存器	SFR 地址	复位值
ADCMP1	CEH, 页 0	0000_0000 b

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

位	名称	描述
7:4	-	保留
3:0	ADCMP[3:0]	ADC 比较值高低字节 ADC 比较值高低4位字节存入这个寄存器

ADCMPPH – ADC比较值高低字节

寄存器	SFR 地址	复位值
ADCMPPH	CFH, 页 0	0000_0000 b

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

位	名称	描述
7:0	ADCMP[11:4]	ADC 比较值高低字节 ADC 比较值高8位低字节存入这个寄存器

PSW – 程序状态字

寄存器	SFR 地址	复位值
PSW	D0H, 所有页, 可位寻址	0000_0000 b

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

位	名称	描述																				
7	CY	高位进位标志 进行加法或减法操作时, 当前运算需要向高位进位或借位时, CY将置位, 否则清零。 在进行MUL 或 DIV运算时, CY始终为0。 CY受DA A指令影响, 用来表示是否初始BCD数大于100。 在CJNE指令中, 如果第一个无符号数的值小于第二个, 则CY置1, 否则清0																				
6	C	辅助进位标志 当前运算导致从半字节的低位序第4位进位或借位, 该位置位, 否则清0。																				
5	0	用户标志 0 可由用户置位或清零的通用标志。																				
4	RS1	寄存器块页选择位																				
3	RS0	这两位用来选择R0到R7位于四页中的哪一页。 <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>RS1</th> <th>RS0</th> <th>寄存器页</th> <th>RAM 地址</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>00H 到 07H</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>08H 到 0FH</td> </tr> <tr> <td>1</td> <td>0</td> <td>2</td> <td>10H 到 17H</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>18H 到 1FH</td> </tr> </tbody> </table>	RS1	RS0	寄存器页	RAM 地址	0	0	0	00H 到 07H	0	1	1	08H 到 0FH	1	0	2	10H 到 17H	1	1	3	18H 到 1FH
RS1	RS0	寄存器页	RAM 地址																			
0	0	0	00H 到 07H																			
0	1	1	08H 到 0FH																			
1	0	2	10H 到 17H																			
1	1	3	18H 到 1FH																			
2	V	溢出标志 OV用于标示发生溢出。对于加法指令 ADD或ADDC指令中, 如果位6有进位而位7没进位, 或者位7有进位而位6没有进位, 则溢出标志置“1”, 反之清“0”。OV也用于标示有符号数累加结果, 当两个正数相加, 或两个负数相加结果为负数时OV为1。对于减法指令SUBB, 当位6发生借位而位7没有, 或者位7发生借位而位6没有借位, 则溢出标志置“1”, 反之清“0”。OV也用于标示两个数相减时, 当一个正数加一负数结果为负, 或两个负数相减结果为负时。 对于MUL乘法指令, 当结果大于255 (00FFH)时, OV置1。反之清0 对于DIV除法指令, 通常情况下OV为0。除非当B设定值为00H, 则A和B的返回值为随机值, 同时OV置1。																				
1	1	用户标志 1 可由用户软件置位或清零的通用标志。																				

位	名称	描述
0	P	奇偶标志 当累加结果为奇数时，该标志置1，偶数时清0。其执行奇偶校验。

指令	CY	OV	AC	指令	CY	OV	AC
ADD	X ^[1]	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	0	X		ANL C, /bit	X		
DIV	0	X		ORL C, bit	X		
DA A	X			ORL C, /bit	X		
RRC A	X			MOV C, bit	X		
RLC A	X			CJNE	X		
SETB C	1						

Note: X 表示修改取决于指令执行结果

表 6.1-3 影响标志位设置的指令

PWMnPH – 周期值高字节, n = 0,1,2,3

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

7	6	5	4	3	2	1	0
PWM0P[15:8]							
读/写							

位	名称	描述
7:0	PWM0P[15:8]	PWM 周期值高字节 与PWM0PL共同决定 PWM 的周期值。

PNP – PWM 反相控制

寄存器	SFR 地址	复位值
PNP	D6H, 页 0	0000_0000 b

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

位	名称	描述
n	PNPn	PWMn 反相使能 0 = 直接输出. PWMn 输出按照设定直接输出到PWMn管脚 1 = 反相输出PWMn 输出按照设定反相输出到PWMn管脚

PWM0FBD – PWM 急停控制

寄存器	SFR 地址	复位值
PWM0FBD	D7H, 页 0	0000_0000 b

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

位	名称	描述
7	FBF	急停标志 FBINENT 置1时, FB引脚若有符合FBINLS (PWM0FBD.6) 定义的边沿发生时, 此位置1。需软件清0。清0后, 置位PWMORUN (PWMOCON0.7), PWM才会再次工作。
6	FBINLS	FB 引脚急停信号输入边沿选择 0 = 下沿. 1 = 上沿.
N	FBDn	PWMx 急停输出信号 0 = PWMOCx 输出低. 1 = PWMOCx 输出高

PWMnCON0 – PWMn 控制 0, n = 1,2,3

寄存器	SFR 地址	复位值
PWM1CON0	B4H, 页 2	0000_0000 b
PWM2CON0	C4H, 页 2	0000_0000 b
PWM3CON0	D4H, 页 2	0000_0000 b

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

位	名称	描述
7	PWMRUN	PWM 工作使能 0 = PWM 停止工作。 1 = PWM 开始工作。
6	LOAD	PWM 加载新的周期和占空比值 此位用于把缓存中的周期和占空比新值加载到周期和占空比寄存器，下个周期开始使用新值。写入缓存的周期和占空比，只有在周末才会加载，然后硬件清零此位。 <u>写:</u> 0 = 无效。 1 = 本次PWM周期结束后加载周期和占空比的新值。 <u>读:</u> 0 = 加载完成。 1 = 加载未完成。
5	PWMF	PWM 标志 按寄存器PWM0INTC中的 INTSEL[2:0] 和 INTTYP[1:0]的定义，此位置1，需软件清零。
4	CLRPWM	计数器清零 此位置1将对 PWM 16位计数器清零，清零完成后，硬件清零此位。 这特性会使得CLRPWM位写入和读出值不一致，意义也不相同。 <u>写:</u> 0 = 无效。 1 = PWM 16位计数器清零 <u>读:</u> 0 = 清零完成。 1 = 清零尚未完成。
3:0	-	保留

PWM0CON0 – PWM0 控制0 (可位寻址)

寄存器	SFR 复制	复位值
PWM0CON0	D8H, 页0	0000_0000 b

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

位	名称	描述
7	PWM0RUN	<p>PWM0 使能位</p> <p>0 = PWM0 停止工作.</p> <p>1 = PWM0 开始工作.</p> <p>注意: 仅适于 PWM0CON0。</p>
6	LOAD	<p>PWM 周期和占空比加载</p> <p>此位用于把缓存中的周期和占空比新值加载到周期和占空比寄存器，下个周期开始使用新值。写入缓存的周期和占空比，只有在周末才会加载，然后硬件清零此位。</p> <p><u>写:</u></p> <p>0 = 无效。</p> <p>1 = 本次PWM周期结束后加载周期和占空比的新值。</p> <p><u>读:</u></p> <p>0 = 加载完成。</p> <p>1 = 加载未完成。</p>
5	PWMF	<p>PWM 标志</p> <p>按寄存器PWM0INTC中的 INTSEL[2:0] 和 INTTYP[1:0]的定义，此位置1，需软件清零。</p>
4	CLRPWM	<p>计数器清零</p> <p>此位置1将对 PWM 16位计数器清零，清零完成后，硬件清零此位。</p> <p><u>写:</u></p> <p>0 = 无效。</p> <p>1 = PWM 16位计数器清零</p> <p><u>读:</u></p> <p>0 = 清零完成。</p> <p>1 = 清零尚未完成。</p>
3	PWM3RUN	<p>PWM3 工作使能</p> <p>0 = PWM3 停止工作.</p> <p>1 = PWM3 开始工作.</p> <p>注意: 仅适于 PWM0CON0。</p>

位	名称	描述
2	PWM2RUN	<p>PWM2 run enable</p> <p>0 = PWM2停止工作.</p> <p>1 = PWM2开始工作.</p> <p>注意: 仅适于 PWM0CON0。</p>
1	PWM1RUN	<p>PWM1 run enable</p> <p>0 = PWM1停止工作.</p> <p>1 = PWM1开始工作.</p> <p>注意: 仅适于 PWM0CON0。</p>
0	P33FBINEN	<p>P33 紧急停止</p> <p>0 = 禁止引脚 P33 FB 对PWM0的急停功能</p> <p>1 = 使能引脚 P33 FB 对PWM0的急停功能，一旦此引脚有符合 FBINLS (FBD.6)定义的跳变发生，PWM0C0~C5 output 输出寄存器 FBD 的值，PWMRUN (PWM0CON0.7) 将被硬件清0，PWM停止工作，PWMRUN 再次写1才开始工作。</p> <p>注意: 仅适于 PWM0CON0。</p>

PWMnPL – PWM 周期值低字节, n = 0,1,2,3

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

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

位	名称	描述
7:0	PWMnP[7:0]	PWM 周期值低字节 与PWM0PH共同决定 PWM 的周期值。

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

寄存器	SFR 地址	描述	复位值
PWM0C0H	D2H, 页 0	PWM0 通道0占空比高字节	0000_0000 b
PWM0C1H	D3H, 页0	PWM0通道1占空比高字节	0000_0000 b
PWM0C2H	D4H, 页0	PWM0 通道2占空比高字节	0000_0000 b
PWM0C3H	D5H, 页0	PWM0 通道3占空比高字节	0000_0000 b
PWM0C4H	C4H, 页1	PWM0 通道4占空比高字节	0000_0000 b
PWM0C5H	C5H, 页1	PWM0 通道5占空比高字节	0000_0000 b
PWM1_CH0H	AAH, 页2	PWM1通道0占空比高字节	0000_0000 b
PWM1_CH1H	ABH, 页2	PWM1通道1占空比高字节	0000_0000 b
PWM2_CH0H	BAH, 页2	PWM2 通道0占空比高字节	0000_0000 b
PWM2_CH1H	BBH, 页2	PWM2 通道1占空比高字节	0000_0000 b
PWM3_CH0H	CAH, 页2	PWM3 通道0占空比高字节	0000_0000 b
PWM3_CH1H	CBH, 页2	PWM3 通道1占空比高字节	0000_0000 b

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

位	名称	描述
7:0	PWMnCx [15:8]	PWMnCx 占空比高字节 与 PWMnCxL共同决定占空比值。

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

寄存器	SFR 地址	描述	复位值
PWM0C0L	DAH, 页 0	PWM0 通道0占空比低字节	0000_0000 b
PWM0C1L	DBH, 页 0	PWM0 通道1占空比低字节	0000_0000 b
PWM0C2L	DCH, 页 0	PWM0 通道2占空比低字节	0000_0000 b
PWM0C3L	DDH, 页 0	PWM0 通道3占空比低字节	0000_0000 b
PWM0C4L	CCH, 页 1	PWM0 通道4占空比低字节	0000_0000 b
PWM0C5L	CDH, 页1	PWM0 通道5占空比低字节	0000_0000 b
PWM1_CH0L	B2H, 页2	PWM0 通道0占空比低字节	0000_0000 b
PWM1_CH1L	B3H, 页2	PWM0 通道1占空比低字节	0000_0000 b
PWM2_CH0L	C2H, 页2	PWM0 通道0占空比低字节	0000_0000 b
PWM2_CH1L	C3H, 页2	PWM0 通道1占空比低字节	0000_0000 b
PWM3_CH0L	D2H, 页2	PWM0 通道0占空比低字节	0000_0000 b
PWM3_CH1L	D3H, 页2	PWM0 通道1占空比低字节	0000_0000 b

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

位	名称	描述
7:0	PWMnCx [7:0]	PWMnCx 占空比低字节 与 PWMnCxH 共同决定PWM的占空比。

PIOCON0 – PWM引脚控制

寄存器	SFR 地址	复位值
PIOCON0	DEH, 页 0	0000_0000 b

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

位	名称	描述
7:6	-	保留
5	PIO03	P0.3/PWM 引脚功能选择 0 = P0.3/PWM 引脚功能为 P0.3. 1 = P0.3/PWM 引脚功能为 PWM0_CH5/PWM3_CH1输出。 (PWM3_CH1P=11, 选 PWM3_CH1; 其它为 PWM0C5) 当AUXR5[7:6]. PWM3CH1P = 00, PIO03 = 1, , P0.3定义为PWM0_CH5. 当AUXR5[7:6]. PWM3CH1P =11, PIO03 = 1, , P0.3定义为PWM3_CH1.
4	PIO01	P0.1/PWM引脚功能选择 0 = P0.1/PWM引脚功能为P0.1 1 = P0.1/PWM引脚功能为PWM0C4/PWM3_CH0输出。 (PWM3_CH0P=10, 选PWM3_CH0; 其它选PWM0C4) 当AUXR5[5:4]. PWM3CH1P = 00, PIO01 = 1, , P0.1定义为PWM0_CH4. 当AUXR5[5:4] PWM3CH1P =10, PIO01 = 1, , P0.1定义为PWM3_CH0.
3	PIO00	P0.0/PWM引脚功能选择 0 = P0.0/PWM引脚功能为P0.0. 1 = P0.0/PWM引脚功能为PWM0_CH3/PWM2_CH1输出。 (PWM2_CH1P=10, 选PWM2_CH1; 其它选 PWM0_CH3) 当AUXR4[7:6]. PWM2CH1P = 00, PIO00 = 1, , P0.0定义为PWM0_CH3 当AUXR5[5:4] PWM2CH1P =10, PIO00 = 1, , P0.0定义为PWM2_CH1
2	PIO10	P1.0/PWM引脚功能选择 0 = P1.0/PWM引脚功能为P1.0. 1 = P1.0/PWM引脚功能为PWM0_CH2/PWM2_CH0输出。 (PWM2_CH0P=10, 选PWM2_CH0; 其它选 PWM0_CH2) 當 AUXR4[5:4]. PWM2CH0P = 00, PIO10 = 1, P1.0 定义为 PWM0_CH2. AUXR4[5:4]. PWM2CH0P = 10, PIO10 = 1, P1.0定义为PWM2_CH0.

位	名称	描述
1	PIO11	<p>P1. 1/PWM引脚功能选择</p> <p>0 = P1. 1/PWM引脚功能为P1. 1.</p> <p>1 = P1. 1/PWM引脚功能为PWM0_CH1/PWM1_CH1输出。</p> <p>(PWM1_CH1P=11, 选PWM1_CH1; 其它选 PWM0_CH1)</p> <p> 當AUXR4[3:2]. PWM1CH1P = 00, PIO11 = 1, P1.1定义为PWM0_CH1.</p> <p> AUXR4[3:2]. PWM1CH1P = 11, PIO11 = 1, P1.1 定义为PWM1_CH1.</p>
0	PIO12	<p>P1. 2/PWM引脚功能选择</p> <p>0 = P1. 2/PWM引脚功能为P1. 2.</p> <p>1 = P1. 2/PWM引脚功能为PWM0C0/PWM1_CH0 输出。</p> <p>(PWM1_CH0P=11, 选 PWM1_CH0; 其它选 PWM0_CH1)</p> <p> 當AUXR4[1:0]. PWM1CH0P = 00, PIO12 = 1, P1.2 定义为 PWM0_CH1.</p> <p> AUXR4[1:0]. PWM1CH0P = 10, PIO12 = 1, P1.2定义为PWM1_CH0.</p>

PWMnCON1 – PWM 控制 1

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

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

寄存器	名称	描述
7:6	PWMMOD[1:0]	PWM 模式选择 00 = 独立模式. 01 = 互补模式. 10 = 同步模式 11 = 保留.
5	GP	组模式使能 组模式使能后前三对 PWM 占空比由 PWM01H PWM01L 决定。 0 = 禁止组模式 1 = 使能组模式. 注意: 仅适用于 PWM0
4	PWMTYP	PWM 类型选择 0 = 边沿对齐. 1 = 中心对齐.
3	FBINEN	FB 急停引脚输入使能 0 = 功能禁止通过FB引脚输入触发PWM输出刹车故障禁止。 1 = 使能急停功能。一旦此引脚出现符合 FBINLS (PWM0FBD.6) 定义的边沿, PWM0 通道 0~5 输出 PWM0FBD 的值。PWM0RUN (PWM0CON0.7)被硬件清0。PWM0RUN置1后 PWM才再次开始工作。 注意: 仅适用于 PWM0

寄存器	名称	描述
2:0	PWMDIV[2:0]	PWM 时钟预分频系数 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 或 ACC – 累加器

寄存器	SFR 地址	复位值
ACC	E0H, 所有页, 可位寻址	0000_0000 b

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

位	名称	描述
7:0	ACC[7:0]	累加器 A或ACC是标准的80C51累加器，用于算数运算。

ADCCON1 – ADC控制 1

寄存器	SFR 地址	复位值
ADCCON1	E1H, 页 0	0000_0000 b

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

位	名称	描述
7	OCEN	ADC 偏移校准使能寄存器 这里使能偏移校准功能 0: ADC 偏移校准使能, ADC 硬件自动校准 1: ADC偏移校准禁止
6	STADCPX	外部触发ADC 触发管脚选择 0 = 选择 STADC 从 P0.4. 1 =选择 STADC 从 P1.3. 注意配置此位后 STADC 管脚立刻改变
5:4	ADCDIV[1:0]	ADC ADCAQT 时钟除频 此字段定义采样点采集时钟分频器值 00 =ADC时钟分频为 $F_{ADCAQT} = F_{SYS}/1$. 01 = ADC时钟分频为 $F_{ADCAQT} = F_{SYS}/2$. 10 = ADC时钟分频为 $F_{ADCAQT} = F_{SYS}/4$. 11 = ADC时钟分频为 $F_{ADCAQT} = F_{SYS}/8$.
3:2	ETGTYP[1:0]	外部触发模式选择 当 ADCEX (ADCCON1.1) 为1,这里选择何种条件触发ADC转换. 00 = PWM0/2/4或STADC管脚下沿 01 = PWM0/2/4 或STADC管脚上升沿. 10 = PWM 周期中点 11 = PWM 周期结束. 注意周期中点中断和周期结束中断只在PWM 中心对齐模式有效.
1	ADCEX	ADC 转换触发选择 这一位选择触发AD转换的条件. 0 = A/D 转换只由设置 ADCS 位触发 1 = A/D 转换通过设置 ADCS 位或做通过ETGSEL[1:0] 和 ETGTYP[1:0]设置的外部触发源触发。注意当 ADCS 为 1 (转换忙), ADC 会忽略后来的触发源直到 ADCS被硬件清除

位	名称	描述
0	ADCEN	ADC 使能 0 = ADC 电路关闭 1 = ADC 电路使能.

ADCCON2 – ADC控制2

寄存器	SFR 地址	复位值
ADCCON2	E2H, 页 0	0000_0000 b

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

位	名称	描述
7	ADFBEN	ADC 转换结果响应宣告故障刹车使能 0 = ADC 响应宣告故障刹车禁止. 1 = ADC 响应宣告故障刹车使能. 当比较结果 ADCMPO 为1触发故障刹车模块. 即符合PWM故障刹车输出值后, 硬件将清除 PWMRUN (PWMCON0.7), 并终止PWM输出. 当PWMRUN置1, PWM重新输出.
6	ADCMPOP	ADC 比较器输出极性 0 = 若ADCR[11:0]大于或等于ADCMP[11:0]., ADCMPO 为 1 1 = 若ADCR[11:0]小于ADCMP[11:0], ADCMPO 为 1
5	ADCMPEM	ADC 结果比较器使能 0 = ADC 结果比较器禁止. 1 = ADC 结果比较器使能
4	ADCMPO	ADC比较器输出值 该位输出ACMPOP设定比较输出的结果。每次AD转换结束都会更新输出。
3:1	ADCAQT1	ADC采集时间 1 这个3位字段决定ADC AIN0~AIN7和BG采样的捕获时间, 遵循下面的等式: $\text{ADC 采集时间} = \frac{4 * \text{ADCAQT} + 6}{F_{\text{ADC}}}$ F _{ADC} 在 ADCDIV (ADCCON2[3:1])定义 默认及最小捕获时间是6个ADC时钟周期。注意, 当ADC正在转换时, 这个字段不应该被改变
0	ADCDLY.8	ADC 外部触发延时计数器第8位 见 ADCDLY 寄存器

ADCCON3 – ADC 控制 3

寄存器	SFR 地址	复位值
ADCCON3	86H, 页 2	0000_0000 b

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

位	名称	描述
7:6	-	保留
5	HIE	ADC 一半完成中断使能 0 = 连续模式下, A/D 转换完成一半不会产生ADC中断 1 = 连续模式下, A/D 转换完成一半会产生ADC中断
4	CONT	ADC连续采样选择 0 = ADC 单次采样, 一次 A/D 转换完成后会产生ADC中断 1 = ADC 连续采样. 所有A/D 转换完成后会产生ADC中断
3:1	ADCAQT1	ADC 采样时间 1 这个3位字段决定ADC AIN9~AIN15采样的捕获时间, 遵循下面的等式: $\text{ADC 采集时间} = \frac{4 * \text{ADCAQT} + 6}{F_{\text{ADC}}} \cdot F_{\text{ADCAQT}}$ 在 ADCDIV (ADCCON2[3:1])定义 默认 $F_{\text{ADCAQT}} = F_{\text{SYS}}$ (ADCDIV=0), 因为最小取样时间必须超过370ns, 这意味当 $F_{\text{sys}} = 24\text{MHz}$ 时, ADCAQT应该设定为1. 默认及最小捕获时间是6个ADC时钟周期。注意, 当ADC正在转换时, 这个字段不应该被改变
0	SLOW	ADC慢速选择 此位选择 ADC慢速模式 0 = ADC 转换时间是高速 1400ns, $F_{\text{ADC}} = 714 \text{ ksps}$ 1 = ADC转换时间是低速 4750ns, $F_{\text{ADC}} = 215 \text{ ksps}$, 注意1: 慢速 ADC 转换可以提高 ADC的转换质量可以提高了转换质量当VDD低于2.5V时的转换质量 注意2: AUXR1.4 (SLOW) 或 ADCCON3 .0(SLOW) 设为 1, ADC 进入低速模式.

ADCDLY – ADC触发延时计数器

寄存器	SFR 地址	复位值
ADCDLY	E3H, 页 0	0000_0000 b

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

位	名称	描述
7:0	ADCDLY[7:0]	<p>ADC 外部触发启动延迟计数器低位</p> <p>该8位寄存器与ADCCON2.0组成9位计数器，用于在外部触发启动ADC之前加入一段延迟。延迟计数结束再开始ADC转换</p> $\text{外部延迟时间} = \frac{\text{ADCDLY}}{F_{\text{ADC}}}$ <p>注，该延迟仅当 ADCEX (ADCCON1.1) 置1时有效。如果启用PWM输出触发ADC功能，在PWM运行过程中不得更改ADCDLY计数值。</p>

CnL – 捕获通道低字节, n = 0,1,2

寄存器	SFR 地址	复位值
C0L	E4H, 页 0	0000_0000 b
C1L	E6H, 页 0	0000_0000 b
C2L	EDH, 页 0	0000_0000 b

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

位	名称	描述
7:0	CnL[7:0]	输入捕获通道n, 捕获结果低字节 CnL 是输入捕获n 16位捕获结果的低字节。

CnH – 捕获 n 高字节, n = 1,2,3

寄存器	SFR 地址	复位值
C0H	E5H, 页 0	0000_0000 b
C1H	E7H, 页 0	0000_0000 b
C2H	EEH, 页 0	0000_0000 b

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

位	名称	描述
7:0	CnH[7:0]	输入捕获n, 捕获结果高字节 CnL 是输入捕获n 16位捕获结果的高字节。

ADCCON0 – ADC控制 0 (可位寻址)

寄存器	SFR 地址	复位值
ADCCON0	E8H, 所有页	0000_0000b

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

位	名称	描述
7	ADCF	ADC标志位 当AD转换完成, 该位置1。可读取到当前AD转换结果。该位为1时无法开始新一轮转换, 需要软件清零
6	ADCS	A/D 转换软件启动位 该位置1启动AD转换。在AD转换过程中该位保持为1, 当转换结束硬件自动清0。这意味着写入ADCS的值和读出的不一定相符 <u>写:</u> 0 = 无动作。 1 = 开始AD转换 <u>读:</u> 0 = ADC 模块空闲状态 1 = ADC 模块工作中
5:4	ETGSEL[1:0]	外部触发源选择 当 ADCEX (ADCCON1.1) 为1, 该位选择外部触发ADC的来源 00 = PWM0. 01 = PWM2. 10 = PWM4. 11 = STADC 脚.

位	名称	描述
3:0	ADCHS[3:0]	<p>A/D转换通道选择</p> <p>该位用于选择连接ADC转换通道。当ADCEN为0所有输入不连接到ADC..</p> <p>0000 = ADC_CH0 0001 = ADC_CH1. 0010 = ADC_CH2. 0011 = ADC_CH3. 0100 = ADC_CH4. 0101 = ADC_CH5. 0110 = ADC_CH6. 0111 = ADC_CH7. 1000 = VBG (内部带隙电压). 1001 = ADC_CH9. 1010 = ADC_CH10. 1011 = ADC_CH11. 1100 = ADC_CH12. 1101 = ADC_CH13. 1110 = ADC_CH14. 1111 = ADC_CH15.</p>

PICON –引脚中断控制

寄存器	SFR 地址	复位值
PICON	E9H, 页 0,	0011_0100 b

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

位	名称	描述
7	PIT7	引脚中断通道7类型选择 该位选择引脚中断通道7触发类型 0 = 电平触发. 1 = 边沿触发.
6	PIT6	引脚中断通道6类型选择 该位选择引脚中断通道6触发类型 0 = 电平触发. 1 = 边沿触发.
5	PIT5	引脚中断通道5类型选择 该位选择引脚中断通道5触发类型. 0 = 电平触发. 1 = 边沿触发.
4	PIT4	引脚中断通道4类型选择 该位选择引脚中断通道4触发类型 0 = 电平触发. 1 = 边沿触发.
3	PIT3	引脚中断通道3类型选择 该位选择引脚中断通道3触发类型 0 = 电平触发. 1 = 边沿触发.
2	PIT2	引脚中断通道2类型选择 该位选择引脚中断通道2触发类型 0 = 电平触发. 1 = 边沿触发.

位	名称	描述
1	PIT1	引脚中断通道1类型选择 该位选择引脚中断通道1触发类型 0 = 电平触发. 1 = 边沿触发.
0	PIT0	引脚中断通道0类型选择 该位选择引脚中断通道0触发类型 0 = 电平触发. 1 = 边沿触发.

PINEN – 引脚中断反相特性使能.

寄存器	SFR 地址	复位值
PINEN	EAH, Page 0	0000_0000 b

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

位	名称	描述
n	PINENn	通道n管脚中断反相特性选择使能位 该位用以使能低电平/下降沿触发中断。至于是电平还是边沿，由PICON寄存器的PITn位决定 0 = 低电平/下降沿检测禁止。 1 = 低电平/下降沿检测使能

PIPEN – 引脚中断正向特性使能.

寄存器	SFR 地址	复位值
PIPEN	EBH, Page 0	0000_0000 b

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

位	名称	描述
n	PIPENn	管脚中断正相特性选择使能位 该位用以使能高电平/上升沿触发中断。至于是电平还是边沿，由PICON寄存器的PITn位决定。 0 = 高电平/上升沿检测禁止 1 = 高电平/上升沿检测使能

PIF – 引脚中断标志

寄存器	SFR 地址	复位值
PIF	ECH, 页 0	0000_0000 b

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

位	名称	描述
n	PIFn	<p>引脚中断通道n标志</p> <p>如果选择了边沿触发，则如果引脚中断的通道n检测到启用的边沿触发器，则硬件将设置此标志。这个标志应该用软件清除。</p> <p>如果选择了电平触发，则此标志为脚中断的通道n上输入信号的逻辑电平的反向。软件无法控制它。</p>

EIP – 扩展中断优先级^[3]

寄存器	SFR 地址	复位值
EIP	EFH, 页 0	0000_0000 b

7	6	5	4	3	2	1	0
PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI ² C
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	WDT中断优先级低位
3	PPWM	PWM中断优先级低位
2	PCAP	引脚捕获中断优先级低位
1	PPI	引脚中断优先级低位
0	PI ² C	I ² C中断优先级低位

注: EIP 和 EIPH 结合使用, 以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

LDOTRIM – LDO 调整

寄存器	SFR 地址	复位值
LDOTRIM	EFH, 页 1	0000_0000 b

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

位	名称	描述
7:5	Uldotrim[2:0]	LDO 输出电压调整 此位用于调整LDO输出电压 000 = 最大电压. 001 . . . 110 = 最小电压. 111 = 100.
4:0	LDOTRIM[4:0]	主LDO输出电压调整 此位用于调整主LDO输出电压 0 0000 = 最大输出电压. 0 0001 1 1110 = 最小输出电压. 1 1111 = 1 0000.

B – B累加器

寄存器	SFR 地址	复位值
B	F0H, 所有页, 可位寻址	0000_0000 b

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

位	名称	描述
7:0	B[7:0]	B 累加器 B累加器是标准8051的另一个累加器。主要用在MUL和DIV指令中。

CAPCON3 –输入捕获控制 3

寄存器	SFR 地址	复位值
CAPCON3	F1H, 页 0	0000_0000 b

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

位	名称	描述
[7:4]	CAP1[3:0]	输入捕获通道1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	SFR 地址	复位值
CAPCON4	F2H, Page 0	0000_0000 b

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

位	名称	描述
7:4	-	保留
[3:0]	CAP2[3:0]	输入捕获通道2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 – 串行外围控制寄存器

寄存器	SFR 地址	复位值
SPCR	F3H, Page 0	0000_0000 b

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

位	名称	描述
7	SSOE	<p>從機選擇片选输出使能</p> <p>.该位搭配DISMODF (SPInSR.3) 用于定义\overline{SS}管脚, 该位仅在MSTR=1和DISMODF=1的条件下有效。</p> <p>0 = \overline{SS}作为普通 I/O.</p> <p>1 = \overline{SS}选择的外部从机设备, 当数据传输时自动拉低, 当总线进入空闲模式时自动变高</p>
6	SPIEN	<p>SPI 使能</p> <p>0 = 关闭SPI功能.</p> <p>1 = 打开SPI功能.</p>
5	LSBFE	<p>LSB 优先使能</p> <p>0 = SPI优先传输最高位MSB数据</p> <p>1 = SPI优先传输最低位LSB数据</p>
4	MSTR	<p>使能主机模式</p> <p>该位用于切换SPI工作于主机与从机模式.</p> <p>0 = SPI 配置为从机模式.</p> <p>1 = SPI 配置为主机模式.</p>
3	CPOL	<p>SPI 时钟极性选择</p> <p>CPOL 位定义, SPI总线在空闲模式时时钟脚的电平状态.</p> <p>0 = SPI 时钟在空闲模式时低电平.</p> <p>1 = SPI 时钟在空闲模式时高电平.</p>
2	CPHA	<p>SPI 时钟相位选择</p> <p>CPHA位定义在采样时所用的时钟边沿. .</p> <p>0 = SPI 在时钟第一个边沿采样数据.</p> <p>1 = SPI 在时钟第二个边沿采样数据. .</p>

位	名称	描述																																								
1:0	SPR[1:0]	<p>SPI 时钟频率选择</p> <p>这两位设置4级SPI 时钟分频. 下面列出时钟频率</p> <p>F_{sys} = 16 MHz</p> <table border="1"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>除频</th> <th>SPI时钟频率</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2</td> <td>8M bit/s</td> </tr> <tr> <td>0</td> <td>1</td> <td>4</td> <td>4M bit/s</td> </tr> <tr> <td>1</td> <td>0</td> <td>8</td> <td>2M bit/s</td> </tr> <tr> <td>1</td> <td>1</td> <td>16</td> <td>1 M bit/s</td> </tr> </tbody> </table> <p>F_{sys} = 24 MHz</p> <table border="1"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>除频</th> <th>SPI 时钟频率</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2</td> <td>12M bit/s</td> </tr> <tr> <td>0</td> <td>1</td> <td>4</td> <td>6M bit/s</td> </tr> <tr> <td>1</td> <td>0</td> <td>8</td> <td>3M bit/s</td> </tr> <tr> <td>1</td> <td>1</td> <td>16</td> <td>1.5M bit/s</td> </tr> </tbody> </table> <p>SPR[1:0] 只在主机模式(MSTR = 1)下有效。从机模式下, 时钟自动同步主机通过 SPICLK 管脚发送来的时钟, 最高到F_{sys}/2</p>	SPR1	SPR0	除频	SPI时钟频率	0	0	2	8M bit/s	0	1	4	4M bit/s	1	0	8	2M bit/s	1	1	16	1 M bit/s	SPR1	SPR0	除频	SPI 时钟频率	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
SPR1	SPR0	除频	SPI时钟频率																																							
0	0	2	8M bit/s																																							
0	1	4	4M bit/s																																							
1	0	8	2M bit/s																																							
1	1	16	1 M bit/s																																							
SPR1	SPR0	除频	SPI 时钟频率																																							
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																																							

SPCR2 – 串行外围控制寄存器 2

寄存器	SFR 地址	复位值
SPCR2	F3H, Page 1	0000_0000 b

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

位	名称	描述																																													
7:2	-	保留																																													
0	SPIS[1:0]	<p>SPI 临近字节间隔时间选择</p> <p>SPIS[1:0] 和 CPHA 设置8级SPI 邻近字节接隔时间如下:</p> <table border="1"> <thead> <tr> <th>CPHA</th> <th>SPIS1</th> <th>SPIS0</th> <th>SPI clock</th> <th>SPI clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0.5</td> <td>0.0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1.0</td> <td>0.5</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1.5</td> <td>1.5</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>2.0</td> <td>2.0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1.0</td> <td>0.0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1.5</td> <td>1.0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>2.0</td> <td>2.0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>2.5</td> <td>2.5</td> </tr> </tbody> </table> <p>SPIS[1:0]只在主机模式 (MSTR = 1)时有效</p>	CPHA	SPIS1	SPIS0	SPI clock	SPI clock	0	0	0	0.5	0.0	0	0	1	1.0	0.5	0	1	0	1.5	1.5	0	1	1	2.0	2.0	1	0	0	1.0	0.0	1	0	1	1.5	1.0	1	1	0	2.0	2.0	1	1	1	2.5	2.5
CPHA	SPIS1	SPIS0	SPI clock	SPI clock																																											
0	0	0	0.5	0.0																																											
0	0	1	1.0	0.5																																											
0	1	0	1.5	1.5																																											
0	1	1	2.0	2.0																																											
1	0	0	1.0	0.0																																											
1	0	1	1.5	1.0																																											
1	1	0	2.0	2.0																																											
1	1	1	2.5	2.5																																											

SPSR – 串行外设状态寄存器

寄存器	SFR 地址	复位值
SPSR	F4H, 页 0	0000_0000 b

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

位	名称	描述
7	SPIF	SPI传输完成标志 在SPI数据传输完成或接收到的数据移入到SPI读缓冲时，该位通过硬件设置为1. 如果使能 ESPI (EIE .0)和 EA, SPI中断请求。该位必须由软件清零。如果SPIF置1，禁止向SPInDR写入
6	WCOL	写冲突位 该位表示写冲突事件. 一旦发生写冲突事件，该位被置位，必须通过软件清零。
5	SPIOVF	SPI 溢出标志 该位表示溢出事件，一旦发生溢出，该位置位，如果使能ESPI 和 EA, SPI请求中断。该位必须由软件清零
4	MODF	模式错误中断状态标志 该位表示模式错误事件。 如果 \overline{SS} 配置成模式错误输入(MSTR=1且DISMODF=0) 和 \overline{SS} 被外部器件拉低, 将产生模式错误， MODF将被置1。如果使能 ESPI 和 EA, SPI中断请求。 该位必须由软件清零
3	DISMODF	禁止模式错误检测 位结合SSOE (SPInCR.7) 位用于决定 \overline{SS} 的特征。DISMODF 仅在主机模式下有效(MSTR = 1) 0 = 使能模式错误检测。 \overline{SS} 为模式错误检测的输入脚，不管SSOE设置。 1 = 禁止模式错误检测。 \overline{SS} 的依赖SSOE 位配置。
2	TXBUF	SPI 写数据缓存状态 此位指示SPI发送缓存状态。 0 = SPI 写数据缓存空 1 = SPI 写数据缓存满。

SPDR – 串行外设数据寄存器

寄存器	SFR 地址	复位值
SPDR	F5H, 页 0	0000_0000 b

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

位	名称	描述
7:0	SPDR[7:0]	<p>串行外设数据</p> <p>该字节为SPI总线上传输或接收的数据。一个字节的写入实际是写入到移位寄存器内操作。一个字节的读取，实际上是一个缓冲区读取数据操作。在主机模式，写该寄存器将启动传输并同时接收一个字节。</p>

AINDIDS0 – ADC通道数字输入通道关闭

寄存器	SFR 地址	复位值
AINDIDS0	F6H, 页 0	0000_0000 b

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

位	名称	描述
n	PnnDIDS	ADC 通道数字通道关闭 0 = 使能 ADC通道的数字通道. 1 =禁止 ADC 通道数字输入， 读ADC 通道一直读到 0.

AINDIDS1 – ADC通道数字输入通道关闭

寄存器	SFR 地址	复位值
AINDIDS1	99H, 页 2	0000_0000 b

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

位	名称	描述
n	PnnDIDS	ADC 通道数字通道关闭 0 = 使能 ADC通道的数字通道. 1 =禁止 ADC 通道数字输入, 读ADC 通道一直读到 0.

EIPH – 扩展中断优先级高位^[4]

寄存器	SFR 地址	复位值
EIPH	F7H, 页 0	0000_0000 b

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

位	名称	描述
7	PT2H	定时器2中断优先级高位
6	PSPIH	SPI中断优先级高位
5	PFBH	故障刹车中断优先级高位
4	PWDTH	WDT中断优先级高位
3	PPWMH	PWM中断优先级高位
2	PCAPH	定时器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 ² C中断优先级高位

注： EIP与EIPH结合使用，以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

SCON_1 – 串口 1 控制

寄存器	SFR 地址	复位值
SCON_1	F8H, 所有页, 可位寻址	0000_0000 b

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

位	名称	描述
7	SM0_1/ FE_1	串口1模式选择 SMOD0_1 (T3CON.6) = 0:
6	SM1_1	详见 表 6.9-2 串口 UART1 模式/波特率描述 SMOD0_1 (T3CON.6) = 1: SM0_1/FE_1 用作帧错误状态标志(FE)，可以被软件清除 0 = 没有发生帧错误 (FE)。 1 = 侦测到帧错误(FE)
5	SM2_1	串口1多机通信模式使能 该位功能用于串口1模式 模式0: 无作用 模式1: 位检查有效停止位。 0 = 接收总是有效不管停止位的逻辑电平。 1 = 接收仅在接收停止位为逻辑1,同时接收数据与GIVEN或广播地址匹配时有效 模式2 或 3: 对于多机通信。 0 =接收总是有效不管第9位的逻辑电平。 1 =接收仅在第9位为逻辑1和接收数据与GIVEN或广播地址匹配时有效。
4	REN_1	串口1接收使能 0 = 关闭串口1接收功能。 1 = 打开串口1在模式1, 2或3模式下的接收功能。 模式0下, 打开接收, 需满足配置REN_1=1及RI_1=0
3	TB8_1	串口1第9位发送位 串口1在模式2和3中要被发送的第九位数据。在模式0和1中, 不支持该功能
2	RB8_1	串口1第9位接收位 串口1在模式2和3中接收到的第九位数据。模式1下, 此位是接收到的停止位逻辑电平。SM2_1为1时此位无效。模式0下该位无意义
1	TI_1	串口1发送中断标志位 发送中断标志: 模式0下该标志在发送完8位数据后由硬件置位, 而在其它模式下在发送完帧数据最后一位后硬件置位。当串口1中断使能, 将执行中断服务程序。该位必须由软件来清除。

位	名称	描述
0	RI_1	<p>串口1接收中断标志位</p> <p>该标志由硬件置位。在模式0中，数据帧接收到第8位；模式1，2或3中，数据帧接收到停止位(stop bit)，当SM2_1=1情况例外。当串口1中断使能，将执行中断服务程序。该位必须由软件来清除。</p>

PWM0DTEN – PWM 死区使能

寄存器	SFR 地址	复位值
PWM0DTEN	F9H, 页 0, TA 保护	0000_0000 b

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

位	名称	描述
7:5	-	保留
4	PWM0DTCNT.8	PWM0 死区计数器的第 8 位 见寄存器 PWM0DTCNT
3	-	保留
2	PDT45EN	PWM0_CH4/ PWM0_CH5 死区使能 此位仅当 PWM0_CH4/ PWM0_CH5 在互补模式有效。 0 = 无死区 1 = 在 POG4/POG5 上沿插入死区时间
1	PDT23EN	PWM0_CH2/ PWM0_CH3 死区使能 此位仅当 PWM0_CH2/ PWM0_CH3 在互补模式有效。 0 = 无死区。 1 = 在 POG2/POG3 上沿插入死区时间。
0	PDT01EN	PWM0_CH0/ PWM0_CH1 死区使能 此位仅当 PWM0_CH0/ PWM0_CH1 在互补模式有效。 0 = 无死区 1 = 在 POG0/POG1 上沿插入死区时间

PWM0DTCNT – PWM 死区计数器

寄存器	SFR 地址	复位值
PWM0DTCNT	FAH, 页 0, TA 保护	0000_0000 b

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

位	名称	描述
7:0	PWM0DTCNT[7:0]	<p>PWM0 死区计数器的低字节</p> <p>此8位与 PWM0DTEN.4组成 PWM0 的9位死区计数器 PWM0DTCNT。仅 PWM0 配置为互补模式并且 PWM0DTEN 的相关位为1时有效。</p> $\text{PWM0 死区时间} = \frac{\text{PDTCNT} + 1}{F_{\text{SYS}}}$ <p>注意在PWM0计数期间不要改 PWM0DTCNT 的值。</p>

PWMxMEN – PWMnCx 屏蔽使能, n=0,1,2,3;x=0,1,2,3,4,5

寄存器	SFR 地址	描述	复位值
PWM0MEN	FBH, 页 0	PWM0_CHn PWM0Cx 屏蔽使能	0000_0000 b
PWM1MEN	ADH, 页 2	PWM1_CHn-PWM1Cx 屏蔽使能	0000_0000 b
PWM2MEN	BDH, 页 2	PWM2_CHn PWM2Cx 屏蔽使能	0000_0000 b
PWM3MEN	CDH, 页 2	PWM3_CHn-PWM3Cx 屏蔽使能	0000_0000 b

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

位	名称	描述
x	PMENx	PWMnCx 屏蔽使能 0 = PWMnCx 信号从PWM发生器输出。 1 = PWMnCx 信号被PMDx屏蔽。 注: PMEN2~5 只能用作 PWM0。

PWMnMD – PWM屏蔽数据

寄存器	SFR 地址	描述	复位值
PWM0MD	FCH, 页 0	PWM0_CHn PWM0Cx 屏蔽数据	0000_0000 b
PWM1MD	ACH, 页 2	PWM1_CHn PWM1Cx 屏蔽数据	0000_0000 b
PWM2MD	BCH, 页 2	PWM2_CHn PWM2Cx 屏蔽数据	0000_0000 b
PWM3MD	CCH, 页 2	PWM3_CHn PWM3Cx 屏蔽数据	0000_0000 b

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

位	名称	描述
x	PMDx	<p>PWMnCx 屏蔽数据</p> <p>一旦正确输入PMENx, PWMnCx 输出数据屏蔽。</p> <p>0 = PWMnCx 信号输出0。</p> <p>1 = PWMnCx 信号输出 1。</p> <p>Note: PMD2~5 只能用作 PWM0.</p>

PORDIS – 上电复位禁止

寄存器	SFR 地址	复位值
PORDIS	FDH, 页 0, TA 保护	0000_0000 b

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

位	名称	描述
7:0	PORDIS[7:0]	上电复位（POR）禁止 首先写5AH到 PORDIS，然后在立即写入 A5H，禁止 POR功能。

EIP1 – 扩展中断优先级 1

寄存器	SFR 地址	复位值
EIP1	FEH, 页 0	0000_0000 b

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

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

注: EIP1 与 EIPH1 结合使用, 以确定每个中断源的优先级。见 [表 6.2-4 中断优先级设置](#)

EIPH1 – 扩展中断优先级高位^[6]

寄存器	SFR 地址	复位值
EIPH1	FFH, 页 0	0000_0000 b

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

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

注：： EIP1与EIPH1结合使用，以确定每个中断源的优先级。见 [表 6.2-4 中断优先级设置](#)。

LVRDIS – 低电压检测禁止

寄存器	SFR 地址	复位值
EIPH1	FFH, 页 1, TA 保护	0000_0000 b

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

位	名称	描述
7:0	LVRDIS[7:0]	低电压检测禁止 首先写5AH到LVRDIS，紧接着再写A5H将禁止LVR。

EIP2 – 扩展中断优先级 2

寄存器	SFR 地址	复位值
EIP2	CEH, Page 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	-	PUART4	PUART3	PUART2
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	PUART4	UART4 中断优先级低位
1	PUART3	UART3中断优先级低位
0	PUART2	UART2中断优先级低位

注EIP2与EIPH2结合使用，以确定每个中断源的优先级。见 表 6.2-4 中断优先级设置。

EIPH2 – 扩展中断优先级高 2

寄存器	SFR 地址	复位值
EIPH2	CFH, Page 2	0000_0000 b

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

位	名称	描述
2	PSC2H	SC2 / UART4 中断优先级高位
1	PSC1H	SC1 / UART3 中断优先级高位
0	PSC0H	SC0 / UART2 中断优先级高位

注：： EIP2与EIPH2结合使用，以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

PIPSn – 引脚中断控制

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

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

位	名称	描述
6:4	PSEL[2:0]	引脚中断通道端口选择 000 = P0 端口. 001 = P1 端口. 010 = P2 端口. 011 = P3 端口. 100 = 保留. 101 = 保留. 110 = 保留. 111 = 保留.
2:0	BSEL[2:0]	引脚中断通道选择 000 = Pn.0. 001 = Pn.1 010 = Pn.2 011 = Pn.3. 100 = Pn.4. 101 = Pn.5. 110 = Pn.6. 111 = Pn.7. n 是端口号, 在PSEL[2:0]里选择.

SCnCR0 – 智能卡控制寄存器 0

寄存器	SFR 地址	复位值
SC0CR0	F1H, 页 2	0000_0000 b
SC1CR0	F3H, 页 2	0000_0000 b
SC2CR0	F5H, 页 2	0000_0000 b

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

位	名称	描述
7	NSB	<p>停止位长度</p> <p>这一位指定停止位长度。</p> <p>0 = 停止位长度是 2 ETU。</p> <p>1 = 停止位长度是 1 ETU。</p> <p>注意: 默认的停止位长度是 2。SC 和 UART 都采用 NSB 设置停止位长度。</p>
6	T	<p>T 模式</p> <p>0 = T0 (ISO 7816-3 T = 0 模式)。</p> <p>1 = T1 (ISO 7816-3 T = 1 模式)。</p> <p>T 模式控制(快保护时间)。块保护时间是在不同传输方向之间的两个连续字符的第一位边缘之间的最短位长度。该域表示用于块保护时间的计数器。依据 ISO7816-3, 在 T=0 的模式下, 软件必须填充该域为 15 (实际的块保护时间=16.5), 在 T=1 的模式下, 软件必须填充该域位 21 (实际的块保护时间为 22.5)。</p> <p>注意:在 T = 0 模式, 接收端会产生奇偶校验标志 PEF(SC0TSR[4]), 如果检测到奇偶校验错误会产生奇偶校验错误信号给发送端。在 T = 1 模式, 接收端会产生奇偶校验标志 PEF(SC0TSR[4]), 如果检测到奇偶校验错误却不会产生奇偶校验错误信号给发送端。</p> <p>注意: 参见 6.10.5</p>
5	RXBGTEN	<p>接收块保护时间功能使能位</p> <p>0 = 接收块保护时间功能禁止。</p> <p>1 = 接收块保护时间功能使能。</p>
4	CONSEL	<p>约定选择</p> <p>0 = 正向直接约定。</p> <p>1 = 反向约定。</p> <p>注意 1:如果 AUTOZEN(SCnCR0[3]) 写“1”, 此位自动清0</p> <p>注意 2: 如果 AUTOZEN(SCnCR0[3]) 使能, 硬件决定约定选择, 在 SCEN (SCnCR0[0]) 为1后自动改变 CONSEL (SCnCR0[4]) 的值</p>

位	名称	描述
3	AUTOGEN	<p>自动约定使能位</p> <p>0 = 禁止自动约定.</p> <p>1 = 自动约定使能。当硬件ATR状态时接收到TS，当TS是正向直接约定, CONSEL(SCnCR0[4]) 会自动写 0，相反当TS是反向约定，CONSEL (SCnCR0[4]) 写1.</p> <p>注意:如果程序使能自动约定功能, 在ATR状态之前设置应该完成，且第一笔数据必须是0x3B或 0x3F. 当硬件接收到第一笔数据到缓存里，硬件会自动决定约定，并改变CONSEL (SCnCR0[4]) 位</p>
2	TXOFF	<p>TX 发送禁止位</p> <p>0 =发送端使能.</p> <p>1 = 发送端禁止</p>
1	RXOFF	<p>RX 接收端禁止位</p> <p>0 =接收端使能</p> <p>1 = 接收端禁止</p> <p>注意:如果AUTOGEN (SCnCR0[3])使能，这一位忽略.</p>
0	SCEN	<p>智能卡引擎使能位</p> <p>设置此位为1使能智能卡操作。如果这一位清0，智能卡会强迫所有传输进入空闲态.</p> <p>注意: SCEN 必须在写其他智能卡寄存器之前置1，不然ISO 7816-3 会工作不正常</p>

SCnCR1 – 智能卡控制寄存器

寄存器	SFR 地址	复位值
SC0CR1	F2H, 页 2	0000_0000 b
SC1CR1	F4H, 页 2	0000_0000 b
SC2CR1	F6H, 页Page 2	0000_0000 b

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

位	名称	描述
7	OPE	奇检验使能位 0 = 发送偶数个逻辑1或接收到相应的奇偶校验位。 1 = 发送奇数个逻辑1或接收到相应的奇偶校验位。 注意: 此位只在 PBOFF 位为 '0' 时有效
6	PBOFF	奇偶校验禁止控制 0 = 在最后一位数据和停止位之间插入奇偶校验位或检测奇偶校验位。 1 = 发送时不产生奇偶校验位, 接收时不检测奇偶校验位 注意: ISO 7816-3 模式下, 这一位要设为 '0' (默认值)
5:4	WLS	数据位数选择 00 = 数据位数 8 bits. 01 = 数据位数 7 bits. 10 = 数据位数 6 bits. 11 = 数据位数 5 bits. 注意: ISO 7816-3 模式下 WLS 要是 '00'
3	TXDMAEN	SC/UART 发送 DMA 使能 这一位使能 SC/UART 发送 PDMA 传输功能, 在SC/UART 开始前发送数据需要在 XRAM 里准备好 0 = SC/UART 发送 DMA 禁止 1 = SC/UART 发送 DMA 使能
2	RXDMAEN	SC/UART 接收 DMA 使能 这一位使能 SC/UART 通过 PDMA 接收数据, SC/UART 接收到数据后自动存入 XRAM 0 = SC/UART 接收 DMA 禁止 1 = SC/UART 接收 DMA 使能
1	CLKKEEP	智能卡 时钟使能位 0 = 禁止智能卡时钟。 1 = 产生智能卡时钟。

位	名称	描述
0	UARTEN	<p>UART 模式使能位</p> <p>0 = ISO 7816-3 模式.</p> <p>1 = UART 模式.</p> <p>注意1:当工作在 UART模式, 用户需要设置CONSEL (SCnCR0[4]) = 0 且 AUTOZEN(SCnCR0[3]) = 0.</p> <p>注意 2:当工作在 ISO 7816-3 模式, 用户需要设置 UARTEN(SCnCR1 [0]) = 0.</p> <p>注意 3:当 UART 使能后, 硬件会复位FIFO和内部状态机</p>

SCnDR – 智能卡数据寄存器

寄存器	SFR 地址	复位值
SC0DR	D9H, 页 2	0000_0000 b
SC1DR	E1H, 页 2	0000_0000 b
SC2DR	E9H, 页 2	0000_0000 b

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

位	名称	描述
7:0	SCnDR[7:0]	<p>SC / UART 缓存数据</p> <p>这个字节用来缓存SC / UART 总线的收发数据。写这个寄存器会写到移位寄存器，读这个寄存器会读到接收缓存的数据。</p> <p>注意: 如果SCEN(SCnCR0[0])没有使能, SCnDR 不可修改.</p>

SCnEGT – 智能卡额外扩展保护时间寄存器

寄存器	SFR 地址	复位值
SC0EGT	DAH, 页 2	0000_0000 b
SC1EGT	E2H, 页 2	0000_0000 b
SC2EGT	EAH, 页 2	0000_0000 b

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

位	名称	描述
7:0	SCnEGT[7:0]	智能卡额外扩展保护时间 这里指定额外扩展保护时间的值 注意: 计数器单位时 ETU .

SCnETURD0 –智能卡ETU 除频寄存器

寄存器	SFR 地址	复位值
SC0ETURD0	DBH, Page 2	0111_0011 b
SC1ETURD0	E3H, Page 2	0111_0011 b
SC2ETURD0	EBH, Page 2	0111_0011 b

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

位	名称	描述
7:0	ETURDIV[7:0]	ETU 除频寄存器的低位 实际的ETU 是 ETURDIV[11:0] + 1. 注 1: ETURDIV[11:0]必须大于 0x004. 注 2: SCnETURD0 必须首先配置, 然后是 SCnETUDR2.

SCnETURD1 –智能卡ETU除频寄存器

寄存器	SFR 地址	复位值
SC0ETURD1	DCH, 页 2	0011_0001 b
SC1ETURD1	E4H, 页 2	0011_0001 b
SC2ETURD1	ECH, 页 2	0011_0001 b

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

位	名称	描述
7	-	保留
6:4	SCDIV [2:0]	智能卡时钟除频 000 = F _{SC} 是 F _{SYS} /1. 001 = F _{SC} 是 F _{SYS} /2. 010 = F _{SC} 是 F _{SYS} /4. 011 = F _{SC} 是 F _{SYS} /8. (默认.) 100 = F _{SC} 是 F _{SYS} /16. 101 = F _{SC} 是 F _{SYS} /16. 110 = F _{SC} 是 F _{SYS} /16. 111 = F _{SC} 是 F _{SYS} /16. 注: 依照 ISO/IEC 7816-3 F _{SC} 时钟需要在 1Mhz ~ 5Mhz
3:0	ETURDIV [11:8]	ETU 除频寄存器的高位 这一位指定除频寄存器的高位. ETU 实际是 ETURDIV[11:0] + 1. 注 1: ETURDIV[11:0] 需要大于 0x004. 注 2: SCnETURD0 需要首先配置, 然后是 SCnETURD1 .

ScnIE –智能卡中断使能控制寄存器

寄存器	SFR 地址	复位值
SC0IE	DDH, 页 2	0000_0000 b
SC1IE	E5H, 页 2	0000_0000 b
SC2IE	EDH, 页 2	0000_0000 b

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

位	名称	描述
7:4	-	保留
4	ACERRIEN	自动约定错误中断使能位 这里使能自动约定错误中断 0 = 禁止自动约定错误中断 1 = 使能自动约定错误中断
3	BGTIEN	块保护时间中断使能位 这里使能快保护时间中断 0 = 禁止快保护时间中断 1 = 使能块保护时间中断
2	TERRIEN	传输错误中断使能位 这里使能传输错误中断。传输错误状态在 SC0TSR 寄存器，包含接收暂停错误 BEF(SC0TSR[6]), 帧错误 FEF(SC0TSR[5]), 奇偶校验错误 PEF(SC0TSR[4]), 接收缓存溢出错误 RXOV(SC0TSR[0]) 和发送缓存溢出错误 TXOV(SC0TSR[2]). 0 = 禁止传输错误中断 1 = 使能传输错误中断.
1	TBEIEN	发送缓存为空中断使能位 这里使能发送缓存空中断 0 = 禁止发送缓存空中断 1 = 使能发送缓存空中断
0	RDAIEN	接收数据到达中断使能位 这里使能接收数据中断 0 = 禁止接收数据中断 1 = 使能接收数据中断.

ScnIS – 智能卡中断状态寄存器

寄存器	SFR 地址	复位值
SC0IS	DEH, 页 2	0000_0010 b
SC1IS	E6H, 页 2	0000_0010 b
SC2IS	EEH, 页 2	0000_0010 b

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

位	名称	描述
7:5	-	保留
4	ACERRIF	自动约定错误中断状态标志（只读） 这里指示自动约定序列错误。如果在ATR状态接收到TS不是0x3B 或 0x3F, 这一位置位。 注意: 此位只读，写0清0
3	BGTIF	块保护时间中断状态标志（只读） 这里指示块保护时间状态标志 注意 1: 这一位当RXBG TEN (SCnCR0[5]) 使能时有效。 注意 2: 此位只读，写0清0。
2	TERRIF	传输错误中断状态标志（只读） 这里用作传输错误中断标志，传输错误状态在 SC0TSR 寄存器，包含接收暂停错误 BEF(SC0TSR[6]), 帧错误 FEF(SC0TSR[5]), 奇偶校验错误 PEF(SC0TSR[4]), 接收缓存溢出错误 RXOV(SC0TSR[0]) 和发送缓存溢出错误 TXOV(SC0TSR[2])。 注意: 这一位时 BEF(SC0TSR[6]), FEF(SC0TSR[5]), PEF(SC0TSR[4]), RXOV(SC0TSR[0]) 和 TXOV(SC0TSR[2])的状态标志，所以如果软件想要清除这一位，需要对这些位都写0
1	TBEIF	发送缓存空中断状态标志（只读） 这里用作发送缓存空中断状态标志 注意: 这里用作发送缓存空中断状态标志，如果软件想清除这一位，软件需要向.DAT(SCnDR[7:0]) 缓存写入数据，这一位会自动清0
0	RDAIF	接收数据到达中断状态标志（只读） 这里用作接收数据中断状态标志 注意: 这里是接收数据状态标志。如果软件从SC_DAT 管脚读取数据，此位自动清除

SCnTSR – 智能卡传输状态寄存器

寄存器	SFR 地址	复位值
SC0TSR	DFH, 页 2	0000_1010 b
SC1TSR	E7H, 页 2	0000_1010 b
SC2TSR	EFH, 页 2	0000_1010 b

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

位	名称	描述
7	ACT	发送接收忙碌状态标志（只读） 0 = 此位在发送接收完成后自动清0 1 = 当发送接收传输正在进行时此位由硬件置1
6	BEF	接收中断错误状态标志（只读） 此位在输入数据管脚上由超过一个完整的字长时间内保持为空闲态（逻辑0），完整字长时开始位加数据位，奇偶校验位和停止位。 注意: 此位只读，写0清0.
5	FET	接收帧错误状态标志（只读） 当接收字符没有有效的停止位时此位置1。（在最后一笔数据位或奇偶校验位后检测到逻辑0） 注意: 此位只读，写1清0.
4	PEF	接收奇偶校验错误状态标志（只读） 当接收到的字符没有有效的奇偶校验位时置1 注意: 此位只读，写0清0.
3	TXEMPTY	发送缓存空状态标志（只读） 此位指示发送缓存是否为空 注意: 当发送缓存的数据传输到发送移位寄存器时此位置1，当写数据到DAT(SCnDR[7:0])后清0 (发送缓存非空).
2	TXOVF	发送溢出错误状态标志（只读） 如果发送缓存已满，再写入数据到DAT(SCnDR[7:0]) 硬件会让此位置1。 注意: 此位只读，写0清0.
1	RXEMPTY	接收缓存空状态标志（只读） 此位指示jies后缓存是否为空 注意: 当接收缓存的数据被CPU读取之后，硬件会把此位置1，当智能卡接收到新的数据后此位清0

位	名称	描述
0	RXOV	接收缓存溢出错误状态标志（只读） 此位当接收缓存溢出时置1。 注意： 此位只读，写0清0

XTLCON – XLT 时钟控制 (TA 保护)

寄存器	SFR 地址	复位值
XTLCON	D7H, 页 2	0111_0111 b

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

位	名称	描述
7	-	保留
6:4	HXSG	HXT 增益值选择 000 = L0 模式 (最小值) 001 = L1 模式 010 = L2 模式 011 = L3 模式 100 = L4 模式 101 = L5 模式 110 = L6 模式 111 = L7 模式 (最大值)
3:0	-	保留

PnDW – 端口 n 下拉电阻控制

寄存器	SFR 地址	复位值
P0DW	9AH, 页 2	0000_0000 b
P1DW	9BH, 页 2	0000_0000 b
P2DW	9CH, 页 2	0000_0000 b
P3DW	9DH, 页 2	0000_0000 b

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

位	名称	描述
7:0	PnDW.x	Pn.x 下拉电阻使能 0 = Pn.x 下拉禁止. 1 = Pn.x 下拉使能.

PnUP – 端口 n 上拉电阻控制

寄存器	SFR 地址	复位值
P0UP	92H, 页 2	0000_0000 b
P1UP	93H, 页 2	0000_0000 b
P2UP	94H, 页 2	0000_0000 b
P3UP	95H, 页 2	0000_0000 b

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

位	名称	描述
7:0	PnUP.x	Pn.x 上拉使能 0 = Pn.x 上拉禁止. 1 = Pn.x 上拉使能.

6.2 系统管理

6.2.1 时钟系统

6.2.1.1 概述

MS51有多种时钟源可供选择，这样在应用中可以使系统性能发挥到最佳，并且功耗降到最低。软件3种系统时钟源可选择，包括内部振荡器和XIN引脚的外部时钟在内的。MS51内嵌2个内部RC振荡器一个10 kHz低速、一个16 MHz或24 MHz高速振荡器，高速误差在出厂时校准到±2%（全温度、全电压范围内）。CKDIV除频器可以调整MS51功耗与性能。

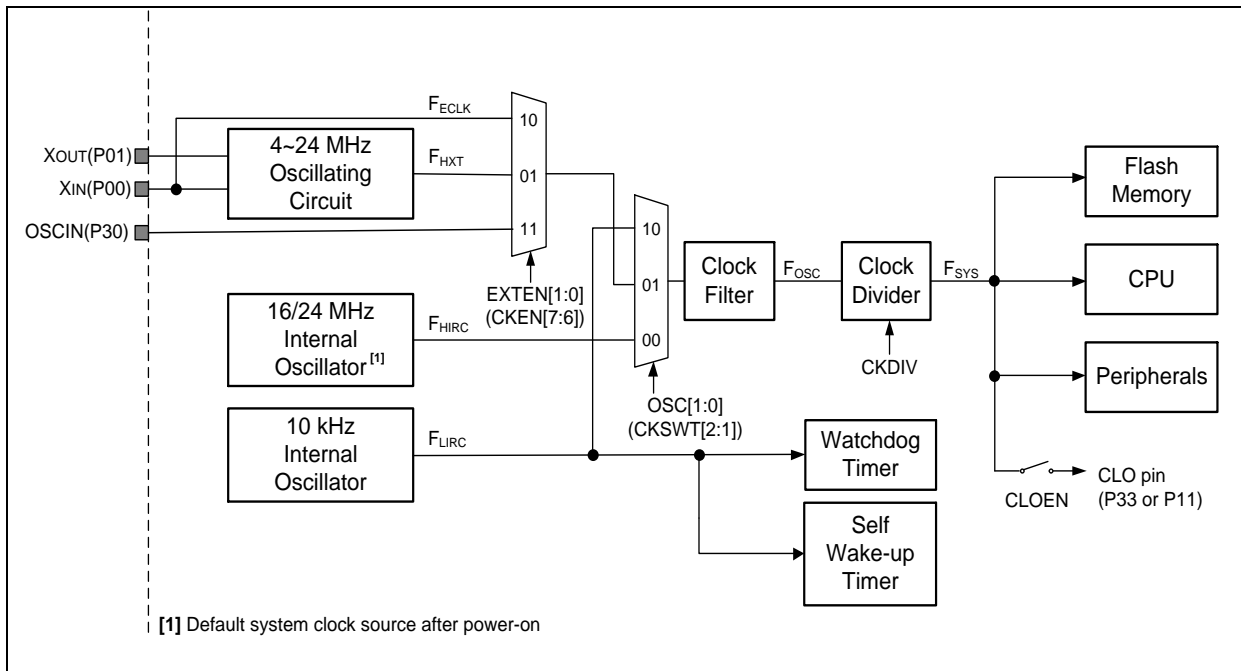


图 6.2-1 时钟系统框图

6.2.1.2 系统时钟源

MS51有3种时钟源：内部高低速振荡器和外部输入时钟。每个都可以作为MS51的系统时钟源。开启不同的时钟源可能会影响到多功能引脚P0.0/ XIN。

6.2.1.3 内部振荡器

MS51内部有高低速两个RC振荡器，高速16MHz或24MHz（由HIRC24(RCTRIM1.4)位选择），低速10 kHz（LIRC）。它们都可被选作系统时钟。通过设置HIRCEN(CKEN.5)位使能HIRC，设备上电时LIRC被使能。用户可设置OSC[1:0] (CKSWT [2:1])为[0,0]选择HIRC作为系统时钟，设置OSC[1:0](CKSWT [1:0])为[1,0]选择LIRC作为系统时钟。注意：MS51上电后HIRC和LIRC都被使能，并且系统默认HIRC为系统时钟。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 P0.0，P0M1和P0M2定义P0.0的输出模式。

有9位数据可更改HIRC频率，每更改一个位的数值，大约更改HIRC频率大约40kHz,值加1，HIRC频率变慢40KHz，更改HIRC频率的寄存器如下：

RCTRIM0 –高速内部振荡器16 MHz 调整值 0

寄存器	SFR地址	复位值
RCTRIM0	84H, 页0, TA保护	0000_0000b

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

RCTRIM1 –高速内部振荡器16 MHz 调整值1

寄存器	SFR地址	复位值
RCTRIM1	85H, All pages, 页0, TA 保护	0000_0000b

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

注意：由于 RCTRIM0 和 RCTRIM1 的默认值是基于 16MHz 的，如果基于这个值，那么修改 HIRC24(RCTIM1.4)使能24MHz的HIRC模式，那么真实的HIRC偏差将超过1%。建议从UID重新加载24MHz的HIRC值，并检查第4位的值，确认该值是否适合24MHz的应用。

6.2.1.4 外部晶振 / 谐振器或时钟输入

外部时钟源有三种—4MHz至24MHz高速晶体/谐振器（HXT），和XIN引脚的外部时钟输入（ECLK）。用户可以将OSC[21:0]设置为[0,1,*]然後設定EXTEN[1:0] (CKEN[7:6])去选择外部時鐘源ECLK为系统时钟。通过設定EXTEN[1:0] as [0,1]OSC[24:0]设为[1,10,0]，将选择HXT为系统时钟。用户可以設定EXTEN [1: 0]为[1,1]，將选择ECLK作为系统时钟，选择P3.0作為外部时钟输入源。或設定EXTEN [1: 0]為[1,0]，選擇 P0.0作為外部时钟输入源。

.当使用HXT作为系统时钟时，XIN和XOUT分别是内部反相放大器的输入和输出，晶体或谐振器应连接在XIN 和 XOUT引脚之间。

当启用和选择ECLK作为系统时钟源时，系统时钟通过Xin引脚提供，常見的應用是 有源晶振或其他主機設備提供時脈。请注意，在选择ECLK模式时，用户不应该向Xin馈送任何大于电压2.4V的时钟信号，并向XOUT管脚提供值为Xin的一半的直流电压。否则，可能会损坏设备。

XTLCON – XLT 时钟控制 (TA 保护)

寄存器	SFR地址	复位值
XTLCON	D7H, 页 2	0111_0111 b

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

位	名称	描述
7	-	保留
6:4	HXSG	HXT 增益值选择 000 = L0 模式 (最小值) 001 = L1模式 010 = L2模式 011 = L3模式 100 = L4模式 101 = L5模式 110 = L6模式 111 = L7模式(最大值)
3:0	-	保留

6.2.1.5 系统时钟切换

MS51 可以通过软件设置寄存器CKSWT 和 CKEN切换时钟源。这给应用带来了很大方便。注意，这些寄存器受TA保护。时钟源可以在外部时钟、内部时钟、内部高速与低速甚至内部高速与低速之间自由的切换。在切换前，必须确保待使用时钟源已稳定。因此，用户要遵循以下设置步骤。用户需要遵循以下设置步骤才能成功完成时钟源切换。首先配置CKEN寄存器打开目标时钟源，再查询CKSWT寄存器中对应的标志位，确定时钟源已稳定，然后写OSC[1:0](CKSWT[2:1])切换到目标时钟源。这些步骤过后，将会成功的切换时钟源。若用户关心功耗，可将原时钟源关闭。如不遵守以上步骤，硬件将会采取以下一些措施来应对这些违规的操作

1. 若用户试图改变CKEN的值来关闭当前时钟源，操作将被忽略。系统时钟保持不变并且CKEN保持原值。
2. 若用户试图改变OSC[1:0] 的值来切换系统时钟而待切换新时钟源未被打开，则OSC[1:0]值立即更新，但系统时钟将保持原来的时钟，CKSWTF标志将由硬件设置。
3. 一旦用户将系统时钟源切换到不稳定的时钟，硬件将等待其稳定。在此等待期间，设备将以原有时钟源继续执行程序，CKSWTF设置为1。目标时钟源稳定后，标志位(见CKSWT[7:3])被置位，时钟源切换成功后，CKSWTF会被硬件自动清0。

CKSWT – 时钟切换

寄存器	SFR地址	复位值
CKSWT	96H, 页 0, TA 保护	0011_0000 b

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

位	名称	描述
7	HXTST	高速外部晶振 4 MHz 至 24 MHz 状态 0 = 高速外部晶振不稳定或被禁用 1 = 高速外部晶振已使能且稳定
6	ECKP00ST	外部时钟P00的输入状态 0 = 外部时钟P00输入不稳定或被禁用 1 = 外部时钟P00输入已使能且稳定
5	HIRCST	高速内部振荡器状态 0 = 高速内部振荡器不稳定或被禁用 1 = 高速内部振荡器已使能且稳定
-	-	保留
3	ECKP30ST	外部时钟P30的输入状态 0 = 外部时钟P30输入不稳定或被禁用 1 = 外部时钟P30输入已使能且稳定
2:1	OSC[1:0]	振荡器选择位 该位是用来选择系统时钟源 00 = 内部16MHz振荡器 01 = 外部时钟控制，通过EXTEN[1:0] (CKEN[7:6]) 设置 10 = 内部10kHz振荡器 11 = 保留 注意该位段只写，读回来的值可能与当前时钟源不一致

CKEN – 时钟使能

寄存器	SFR地址	复位值
CKSWT	97H, 页 0, TA 保护	0011_0000 b

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

位	名称	描述
7:6	EXTEN[1:0]	外部时钟源使能 11 =外部时钟输入通过OSCIN (P30)使能 10 =外部时钟输入通过HXTIN (P00)使能 01 = 外部晶振 4~24 MHz 使能 00 = 禁用外部时钟输入。 P30/P00/P01 作为普通 I/O或其它功能使用
5	HIRCEN	高速内部振荡器16MHz使能 0 = 高速内部振荡器禁用 1 = 高速内部振荡器使能 注意一旦设置IAPEN (CHPCON.0)位开启IAP功能， HIRC将会自动使能，硬件也会设置HIRCEN和HIRCST位。 IAPEN被清除后， HIRCEN和EHRCST位会恢复为原始值。
4	LIRCEN	内部低速振荡器 10kHz 使能 0 = 内部低速振荡器禁用 1 = 内部低速振荡器使能 注意当启用(1)WDT使能， (2)WKT的时钟源是内部10kHz振荡器才能运行(3)BOD启用， 或(4)LVR滤波器启用时，对LIRCEN写0将被忽略。 LIRCEN总是1，内部10 kHz振荡器总是启用。
3:1	-	保留位
0	CKSWTF	时钟切换错误标志位 0 = 系统时钟源切换成功 1 = 先前用户试图切换系统时钟的时钟源，没有开启或不稳定。如果待切换的时钟源不稳定，该位将一直保持为1，直到时钟源稳定并切换成功为止

6.2.1.6 系统时钟除频

振荡频率(F_{OSC})通过配置除频寄存器CKDIV，整数倍（最大到1/510）除频后，再供给系统作为系统时钟(F_{SYS})。这一特征可以临时让MCU跑在很低的速度下来降低功耗。通过时钟除频，MCU可以通过以较低的速率执行其正常的程序来保留响应那些可能导致中断的事件(即允许退出空闲模式的事件)之外的事件的能力。这有可能比空闲模式还要省电。这样，可以避免掉电模式情况下，需要等待振荡器重新起振的时间。在不中断代码执行情况下，CKDIV的值可以在任何时间被程序改变。

CKDIV – 时钟除频器

寄存器	SFR地址	复位值
CKDIV	95H, 页0	0000_0000b

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

位	名称	描述
7:0	CKDIV[7:0]	<p>时钟除频器</p> <p>根据CKDIV值，系统时钟频率FSYS如下式所示</p> <p>当 CKDIV = 00H时， $F_{SYS} = F_{OSC}$</p> <p>当 CKDIV = 01H 至 FFH时， $F_{SYS} = \frac{F_{OSC}}{2 \times CKDIV}$，</p>

6.2.1.7 系统时钟输出

MS51提供一个CLO管脚可以输出系统时钟，该频率与F_{sys}频率相同。通过设置 CLOEN(CKCON.1)位打开这个功能。在掉电模式下CLO输出会停止，因为系统时钟已被关闭。注意当有干扰问题或是功耗问题时，用户最好关闭CLO输出。

CKCON – 时钟控制

寄存器	SFR地址	复位值
CKCON	8EH, 页0	0000_0000b

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

位	名称	描述
1	CLOEN	<p>系统时钟输出使能</p> <p>0 = 系统时钟输出禁止</p> <p>1 = 系统时钟输出使能，从CLO管脚输出</p> <p>注: CLO管脚取决于AUXR5.7</p>

6.2.2 电源管理

6.2.2.1 概述

MS51有几种工作模式可以帮助用户控制设备功耗，省电模式有两种模式包括掉电模式和空闲模式，用以节省功耗。控制设备功耗，必须处理好每个引脚的模式和状态。每个引脚状态需要外部上、下拉一致，比如上拉就要输出1，下拉就要输出0。如果引脚是悬浮的，建议用户配置端口为准双向模式。如果P2.0配置为输入引脚，必须外接上拉或下拉电阻或通过设置P20UP (P2UP.0)内部上拉

PCON – 电源控制

寄存器	SFR地址	复位值
PCON	87H, 所有页	POR: 0001_0000b 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
1	PD	<p>掉电模式</p> <p>设置该位使MCU进入掉电模式。在此模式下，CPU和外设时钟停止，程序计数器（PC）挂起，此时系统为最小功耗模式。CPU从掉电模式下唤醒后，该位自动由硬件清零，且程序继续执行唤醒系统的中断源对应的中断服务程序ISR。从ISR返回后，设备从让系统进入掉电模式的指令后续指令继续执行</p> <p>注: 如果IDL位和PD位同时置位，MCU进入掉电模式。从掉电模式退出后不会进入空闲模式。</p>

位	名称	描述
0	IDL	<p>空闲模式</p> <p>设置该位使MCU进入空闲模式。在此模式下，CPU时钟停止，且程序计数器（PC）挂起，但是所有外设继续工作。CPU从空闲模式唤醒后，该位自动由硬件清零，且程序继续执行唤醒系统的中断源对应的中断服务程序ISR。从ISR返回后，设备从让系统进入空闲模式的指令后续指令继续执行</p>

6.2.2.2 空闲模式

空闲模式下通过保持程序计数器使CPU挂起。在空闲模式下没有程序代码的取址指和运行，这迫使CPU处于待机状态。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值不变。端口引脚保持原有状态，在空闲模式下继续工作。通常空闲模式下的功耗约为工作状态下的一半。

既然在空闲模式下，外设电路，如定时器和串口仍然工作，则可以通过使能相应中断源来唤醒CPU。用户能通过向IDL (PCON.0)位写1，使设备进入空闲模式。这条指令是系统进入空闲模式前的最后一条指令。

有两种方法可以中止空闲模式，方法一、任何使能的中断发生都可以使系统退出空闲模式。中断发生自动清零IDL位，中止空闲模式，且将执行中断源对应的中断服务程序(ISR)，直到执行RETI返指令，退出中断后继续从让CPU进入空闲模式的指令后续指令执行。第二种方法是除软件复位外的所有复位，如果看门狗复位用来中止空闲模式，WIDPD (WDCON.4)需要设置为1，让WDT在空闲模式下继续运行。

6.2.2.3 掉电模式

掉电模式是MS51进入最低功耗状态的工作模式，通过停止系统时钟源保持功耗在“微安”级。CPU和外设，如定时器或UART都待机，Flash内存也停止，所有动作完全停止，功耗降到最低。可以通过向PD(PCON.1)写1进入掉电模式。这条指令是系统进入掉电模式前的最后一条指令。在掉电模式下，RAM保存其内容，端口引脚的值也保持掉电模式前的不变。

MS51有多种方法可以退出掉电模式。方法一，除软件复位外的所有复位。欠压检测复位也能使CPU从掉电模式唤醒，在系统进入掉电模式之前要确保使能欠压检测。即使为了降低功耗，我们还是建议在掉电模式下开启低功耗BOD欠压检测功能。当然外部引脚复位或上电复位也可以使CPU退出掉电模式。外部引脚复位或上电复位后，CPU初始化并从程序开始地址执行程序。

方法二，可以通过外部中断使MS51从掉电模式唤醒。触发外部中断管脚会重启系统时钟，在振荡器稳定后，设备执行外部中断对应的中断服务程序（ISR）。从ISR返回后，设备立即执行使系统进入掉电模式的指令接下来的指令。可以将芯片从掉电模式唤醒的中断有：外部中断INT0和INT1、引脚中断、WDT中断、WTK中断和欠压中断。

6.2.3 电源监控與复位

6.2.3.1 概述

为防止上电和掉电时，程序执行异常，MS51提供2种电源监控功能，上电复位检测，欠压检测。

MS51的复位条件有几中类型。通过寄存器标志位可以确定复位源。通常，大部分特殊功能寄存器复位后的值与复位条件无关，但是一些复位源的标志位的状态取决于复位源。有6种方法使芯片进入复位状态。他们是上电复位、欠压复位、外部复位、硬件故障复位、看门狗定时器复位以及软件复位。

6.2.3.2 上电检测 (POR) 與低电压复位 (LVR)

MS51包含内部上电复位 (POR) 和低电压复位 (LVR)。上电检测功能，用于检测电源上升到系统可以工作的电压。在上电过程中，当VDD低于参考电压门限值，上电复位将保持CPU为复位模式。这种设计使CPU在VDD 不满足执行读取存储器时，不访问程序存储器空间。如果从程序存储器读取并执行一

个不确定的操作码，可能会使CPU甚至是整个系统进入错误状态。VDD 上升到参考门限电压以上，系统工作，所选的振荡器起振，程序从0000H开始执行。同时，上电检测后，POF (PCON.4) 将置1，标志为冷复位，上电复位完成。POF标志可由软件清零。上电后，内部RAM的内容不确定。建议用户初始化RAM。

建议通过软件清除POF为0，以检测在下一次复位是冷复位还是热复位。如果是由掉电或上电引起的冷复位，POF 将再次置1。如果是由其他复位源引起的热复位，POF将保持为0。用户可以检测复位标志位，处理热复位事件。

PCON – 电源控制

寄存器	SFR地址	复位值
PCON	87H, 所有页	POR: 0001_0000b 其它: 000U_0000b

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

位	名称	描述
4	POF	上电复位标志 当上电后该位置1，用来标志当前是冷复位，上电复位完成。其它任何复位不会影响该位，建议通过软件清

PORDIS – 上电复位禁止

寄存器	SFR 地址	复位值
PORDIS	FDH, 页 0, TA 保护	0000_0000 b

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

位	名称	描述
7:0	PORDIS[7:0]	上电复位 (POR) 禁止 首先写5AH到 PORDIS，然后在立即写入 A5H，禁止 POR功能。

LVRDIS – 低电压检测禁止

寄存器	SFR 地址	复位值
EIPH1	FFH, 页 1, TA 保护	0000_0000 b

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

位	名称	描述
7:0	LVRDIS[7:0]	低电压检测禁止 首先写5AH到LVRDIS, 紧接着再写A5H将禁止LVR。

6.2.3.3 欠压检测 (BOD)

另一个电源监控功能是欠压检测电路，欠压检测电路是用于监测运行期间V_{DD}电平。有4个可编程的欠压检测触发电平，以适用于宽电压应用。4级电平 2.2V、2.7V、3.7V、4.4V，通过 CBOV[1:0] (CONFIG2[5:4])选择。当然在上电后也可以通过设置BOV[1:0] (BODCON0[6:4])来改变BOD电平。当VDD下降到所选择的欠压检测触发电平(VBOD)，欠压检测逻辑将复位MCU或请求欠压检测中断。用户可结合不同应用决定设备是欠压复位还是产生欠压中断。上电后也可以通过软件打开VBOD，注意在软件打开BOD后需要等待2到3个LIRC时钟才能正常工作。

当VDD下降到VBOD下并且 BORST (BODCON0.2) 为0时，BOD将会请求中断。此情况下，BOF(BODCON0.3)将被置1。用户清除该标志后，VDD依然保持在VBOD下，BOF不会被再次置1，BOF仅通知用户电源电压下降发生。当VDD上升到高于VBOD时，BORF将置1，以示电源恢复。BOD电路提供了一个很有用的状态位BOS (BODCON0.0)，可以用来指示当前是欠压还是电源已经恢复。设置BORST为1将开启欠压复位功能。欠压复位过后，BORF (BODCON0.1)将会被硬件置1，不会被其它复位重置除上电复位外，该位可以通过软件清除。注意BODCON0所有位的写入都受时效访问TA保护。

MS51支持低功耗BOD模式，在为了节省电流消耗的同时最大的发挥 BOD检测性能。通过设置LPBOD[1:0] (BODCON1[2:1])，BOD电路可以周期性的检测电源电压，通常是1.6ms、6.4ms或25.6ms。这样可以减少很多功耗，但同时电源电压检测速度将降低。注意在低功率BOD模式下，欠压检测迟滞特性将会消失。

对于噪声敏感的系统，MS51有一个BOD的滤波器可以避免电源噪声无意识地触发BOD事件。BOD滤波器上电默认开启，如果用户想要一个快速反应的BOD系统可以通过清BODFLT (BODCON1.0)为0来关闭。最小欠压检测脉冲宽度见LPBOD[1:0] (BODCON1)

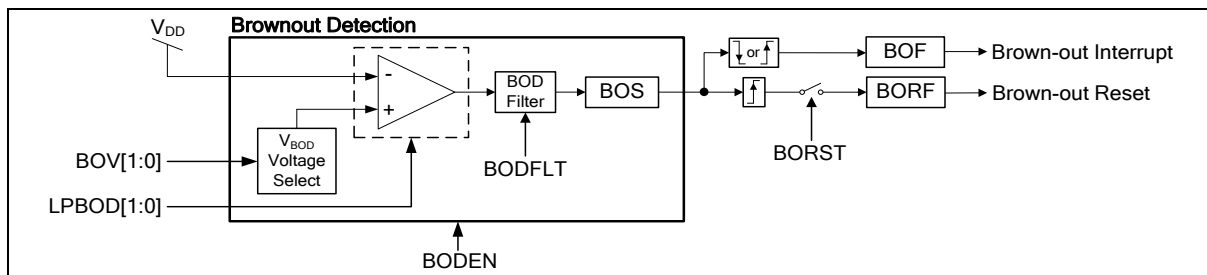


图 6.2-2 欠压检测框图

CONFIG2

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

出厂默认值: 1111 1111b

位	名称	描述
7	CBODEN	配置欠压检测使能位 1 = 欠压检测功能打开 0 = 欠压检测功能关闭
5:4	CBOV[1:0]	配置欠压检测电压选择位 11 = V_{BOD} 生效电压2.2V 10 = V_{BOD} 生效电压2.7V 01 = V_{BOD} 生效电压3.7V 00 = V_{BOD} 生效电压4.4V
2	CBORST	配置欠压检测复位使能 该位决定在电源电压跌到 V_{BOD} 以下时是否产生欠压检测复位 1 = 使能欠压检测复位 0 = 禁用欠压检测复位

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

表 6.2-1 BOF 复位值

BODCON0 – 欠压检测控制 0

寄存器	SFR地址	复位值
BODCON0	A3H, 页 0, TA 保护	POR,CCCC XC0X b BOD, UUUU XU1X b 其它,UUUU XUUX b

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

位	名称	描述
7	BODEN	欠压检测使能 0 = 禁用欠压检测电路 1 = 使能欠压检测电路 注意在开启该功能后需要2到3个LIRC时钟BOD才能正常工作
6:4	BOV[1:0]	配置欠压检测电压选择位 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, 该标志由硬件设置为逻辑1。如果EBOD(EIE.2)和EA(IE.7) 都置位, 将请求欠压检测中断。该位必须由软件清零。
2	BORST	欠压检测复位使能 该位决定在电源电压跌到以V _{BOD} 下时是否产生欠压检测复位 0 = 禁用欠压检测复位 1 = 使能欠压检测复位
1	BORF	欠压复位标志 当MCU发生欠压复位时, 该位被硬件值1。建议通过软件清除该位。
0	BOS	欠压状态标志 在BOD电路开启时, 该位反应V _{DD} 与V _{BOD} 比较情况。 BOD电路关闭时保持为0。 0 = V _{DD} 电压大于V _{BOD} 或是BOD电路关闭 1 = V _{DD} 电压小于V _{BOD} 注该位为只读位
注: 1. 所有复位后BODEN、 BOV[1:0]和 BORST初始化的值是直接通过加载CONFIG2位7位6~4和位2决定 2. BOF复位后的值依据CONFIG2的设置和VDD的电平. 请参考表 6.2-1 BOF		

BODCON1 – 欠压检测控制 1

寄存器	SFR地址	复位值
BODCON1	ABH, 页 0, TA 保护	POR 0000 0001 b 其它 0000 0UUU b

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

位	名称	描述
7:3	-	保留
2:1	LPBOD[1:0]	低功耗BOD使能 00 = BOD 正常模式. BOD 电路总是开启. 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周期性开启BOD电路
0	BODFLT	BOD 滤波器控制 BOD具有一个滤波器, 当MCU以HIRC或ECLK作为系统时钟且BOD没有在低功率模式下 (LPBOD[1:0] = [0, 0])工作时, 该滤波器采样32个FSYS 时钟用以过滤电源噪声。在其他条件下, 滤波器采样2个LIRC。 注:当CPU停在掉电模式时, BOD滤波计数一直是2个LIRC时钟 BOD滤波器有效地避免电源噪声误触发BOD事件发生。设置该位可以开启或关闭BOD滤波功能。 0 = BOD 滤波禁止。 1 = BOD 滤波使能. (上电复位默认值.)

BODFLT (BODCON1.1)	BOD 工作模式	系统时钟源	最小欠压检测脉冲宽度
0	正常模式 (LPBOD[1:0] = [0,0])	任意时钟源	Typ. 1 μ s
	低功耗模式 1 (LPBOD[1:0] = [0,1])	任意时钟源	16 (1/FLIRC)
	低功耗模式 2 (LPBOD[1:0] = [1,0])	任意时钟源	64 (1/FLIRC)
	低功耗模式 3 (LPBOD[1:0] = [1,1])	任意时钟源	256 (1/ FLIRC)
1	正常模式 (LPBOD[1:0] = [0,0])	HIRC/ECLK	正常工作模式: 32 (1/FSYS) 空闲模式: 32 (1/FSYS) 掉电模式: 2 (1/FLIRC)
		LIRC	2 (1/FLIRC)
	低功耗模式 1 (LPBOD[1:0] = [0,1])	任意时钟源	18 (1/FLIRC)
	低功耗模式 2 (LPBOD[1:0] = [1,0])	任意时钟源	66 (1/FLIRC)
	低功耗模式 3 (LPBOD[1:0] = [1,1])	任意时钟源	258 (1/ FLIRC)

表 6.2-2 最小欠压检测脉冲宽度

6.2.3.4 外部复位和硬件故障复位

外部复位引脚nRESET是带施密特触发器的输入引脚。外部RST引脚保持最少24个系统时钟周期的低电平，以确保能检测到有效的硬件复位信号，完成一次硬件复位动作。复位电路同步请求内部复位信号，因此，复位是同步运行，要求时钟在此期间运行来促使外部复位

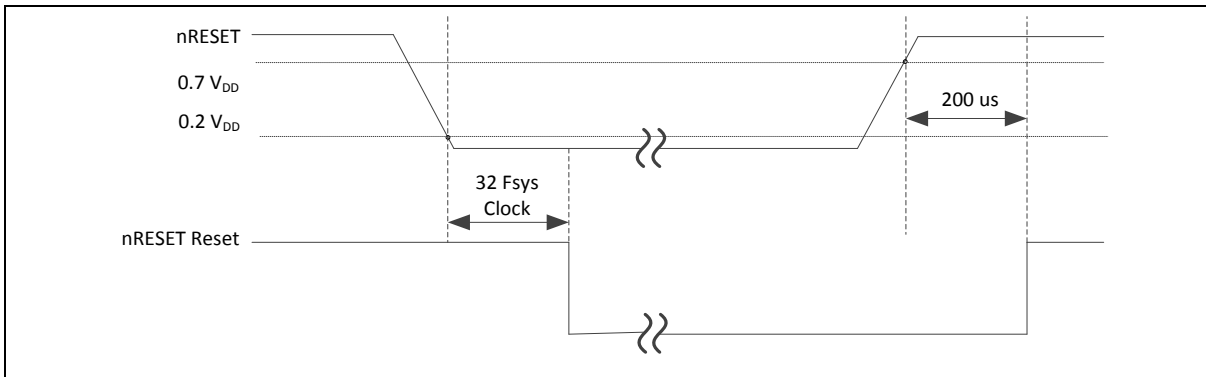


图 6.2-3 nRESET 复位波形

在复位条件下，只要nRESET引脚电平保持在低就一直处在复位状态直到变成高，CPU将退出复位状态，并从地址0000H处开始执行代码。如果CPU在掉电模式下，外部RST $\bar{}$ 引脚复位时，触发硬件复位的方法略有不同。因为掉电模式下系统时钟是停止的，复位信号将等待系统时钟恢复。在系统时钟稳定后，CPU将进入复位状态。

RSTPINF (AUXR1.6) 为复位标志位，用来标志发生了外部复位。当发生外部复位后，该位硬件置1。除上电复位或外部复位引脚复位外，该位不会置1，并通过软件清零。

程序计数器PC溢出flash地址空间，硬件故障将发生。硬件故障复位后辅助寄存器1 HardF(AUXR1.5)被硬件置位，辅助寄存器1 HardF除了会被上电复位或外部复位更改，不会被任何其他复位更改，这位能通过软件清零。当MCU运行在OCD调试模式并且OCDEN=0,硬件故障复位被禁用,仅仅HardF标志位置位。

AUXR1 – 辅助寄存器 1

寄存器	SFR地址	复位值
AUXR1	A2H, 页 0	POR: 0000 0000b, 软件复位: 1U00 0000b, nRESET 外部复位: U100 0000b, 其它: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS
读/写	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 通过外部复位引脚复位MCU后, 该位将被硬件置1, 建议复位发生后通过软件清零
5	HardF	硬件故障复位标志 一旦程序计数器(PC)溢出flash地址空间EHFI (EIE1.4)=0, MCU将复位并且HardF硬件置位。通过软件清零 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HardF置位

6.2.3.5 看门狗复位

看门狗定时器是一个自由运行的定时器, 带可编程溢出时间间隔和专用内部时钟源。用户可以在任何时候清除看门狗定时器, 使它重新开始计数。当选择的溢出时间间隔发生溢出后, 软件在一段时间内没有响应, 看门狗定时器将直接复位系统。复位完成后, 芯片从地址0000H开始运行。

如果看门狗定时器引起复位, 看门狗定时器复位标志WDTRF (WDCON.3)将置位。除上电复位或看门狗复位外该位保持不变, 用户可以通过软件清WDTRF。

WDCON – 看门狗定时器控制

寄存器	SFR地址	复位值
WDCON	AAH, 页 0, TA 保护	POR 0000_0111 b WDT 0000_1UUU b 其它 0000_UUUU b

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

位	名称	描述
---	----	----

位	名称	描述
3	WDTRF	看门狗复位标志 WDT 复位标志。当看门狗定时器溢出复位MCU时，该位由硬件置位。通过软件清零

6.2.3.6 软件复位

MS51提供软件复位功能，允许软件复位整个系统类似于外部复位，初使化MCU为复位状态。软件复位在ISP动作结束后非常有用。例如，如果通过ISP启动代码更新用户代码完成，软件复位能重启CPU立即执行用户代码。写 1 到 SWRST (CHPCON.7) 触发软件复位。注意，SWRST时效访问控制受TA保护，执行设置SWRST 位是设备复位之前的最后指令。见下面例程。

发生软件复位SWRF (AUXR1.7) 被硬件置1，用户可通过读取该位，来确定复位发生原因。除上电复位或软件复位外，SWRF不会被其它复位修改。通过软件清零。

CHPCON – 芯片控制

寄存器	SFR地址	复位值
CHPCON	9FH, 所有页, TA 保护	软件复位 0000_00U0 b 其它 0000_00C0 b

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

位	名称	描述
7	SWRST	软件复位 对该位写1，芯片执行软件复位，复位完成后该位自动清零

AUXR1 – 辅助寄存器 1

寄存器	SFR地址	复位值
AUXR1	A2H, 页 0	POR: 0000 0000b, 软件复位: 1U00 0000b, nRESET 外部复位: U100 0000b, 其它: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS
读/写	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, 该位硬件置1。通过软件清零

软件复位例程如下

```

ANL  AUXR0, #01111111b ;software reset flag clear
CLR  EA
MOV  TA, #0Aah
MOV  TA, #55h
ORL  CHPCON, #10000000b ;software reset
    
```

6.2.3.7 启动选择

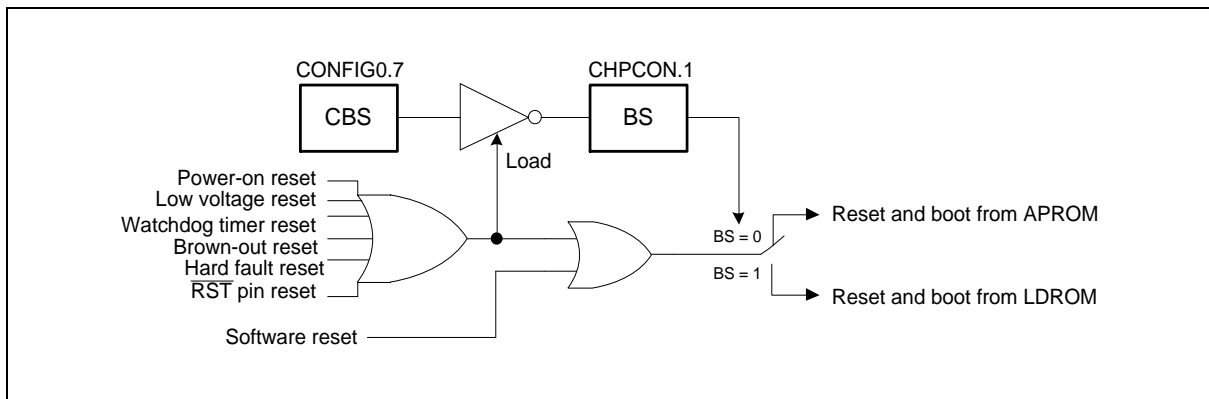


图 6.2-4 启动选择框图

MS51提供给用户灵活的启动选择, 适合不同的应用。BS(CHPCON.1)位用于决定复位后, CPU从 APROM 或LDROM 中启动。复位后, 如果BS = 0, CPU从APROM 中启动, 由0000H开始运行。反之 CPU 从 LDROM启动, 从0000H开始运行。注: 所有复位(除软件复位)后, BS加载CBS位 (CONFIG0.7)的相反值。

CONFIG0

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

出厂默认值: 1111 1111b

位	名称	描述
7	CBS	<p>配置启动选择位</p> <p>该位定义，除软件复位外的所有复位后，MCU选择FLASH区启动。</p> <p>1 = 除软件复位外的所有复位后，MCU从APROM启动</p> <p>0 = 除软件复位外的所有复位后，MCU从LDROM启动</p>

CHPCON – 芯片控制

寄存器	SFR地址	复位值
CHPCON	9FH, 所有页, TA 保护	软件复位 0000_00U0 b 其它 0000_00C0 b

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

位	名称	描述
1	BS	启动选择 定义复位后MCU由哪块启动 0 =由APROM启动 1 =由LDROM启动

[1] BS位是由复位后（除软件复位外）读取CONFIG0.7 CBS位内容相反的值并写入，软件复位后保持不变。CPU从所有复位状态释放后，硬件将检查BS位（非CBS）以决定是由APROM还是LDROM启动。

6.2.3.8 复位状态

除上电复位外复位状态不会影响片上RAM。在复位期间，RAM中的数据保留，上电复位后，RAM中内容不确定。

复位后，大多数寄存器恢复到默认值，有一些特殊功能寄存器初始值取决于不同复位事件。一旦复位，程序计数器强制切换至地址0000H。备注：堆栈指针复位至07H，同时堆栈内的数据可能丢失，即便RAM内的数据未改变。

复位状态下，所有外设及中断关闭，所有管脚值为FFH，并切换至输入模式。

6.2.4 中断系统

6.2.4.1 概述

中断的目的是让软件处理非常规或异步的事件。MS51有24个中断源，4个中断优先级。每个中断源有独立的优先级设置位、中断向量和使能位。中断可以全局使能或关闭。中断服务程序(ISR)被分配到预先指定地址，如表 6.2-3 中断函数首地址表所示。如果中断使能，中断发生时CPU 根据中断源跳到相应地址，执行此地址处的程序，保持中断服务状态直到执行完中断服务程序ISR。一旦ISR 开始执行，仅能被更高优先级的中断抢占。ISR 通过指令RETI返回，该指令强迫CPU回到中断发生前所执行指令的下一条指令。

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

表 6.2-3 中断函数首地址表

6.2.4.2 中断使能

每个中断源都可独自使能或禁止，这些位在寄存器IE和EIE中。中断全局使能位EA(IE.7)，清0将关闭所有中断，置位启用已单独使能了的的中断，清0不管单独的中断源，是否使能了都关闭所有中断。有中断请求也会被挂起直到EA恢复为1，才去执行该中断。所有中断标志位可以软件置位启动中断。

注意:中断产生时对应中断标志位会置1。大多数中断标志位都是写0清除。

IE – 中断使能寄存器

寄存器	SFR地址	复位值
IE	A8H, 所有页, 可位寻址	0000_0000 b

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

位	名称	描述
7	EA	使能所有中断 该位全局使能/禁止所有那些单独使能的中断 0 = 禁止所有中断源 1 = 每个中断使能依靠使能单个中断。如果使能, 将会产生相应的单个中断
6	EADC	使能ADC中断 0 = ADC 中断禁止. 1 = ADC 中断使能. 当中断产生时ADCF (ADCCON0.7) 置1
5	EBOD	使能欠压中断 0 = 欠压检测中断禁止. 1 = 欠压检测中断使能. 当中断产生时BOF (BODCON0.3) 置 1
4	ES	使能串口0中断 0 = 串口 0 中断禁止. 1 = 串口 0 中断使能. 当中断产生时TI (SCON.1) 或 RI (SCON.0) 置1
3	ET1	使能定时器1中断 0 = 定时器 1 中断禁止. 1 = 定时器 1 中断使能. 当中断产生时TF1 (TCON.7) 置 1
2	EX1	使能外部中断1 0 = 外部中断1中断禁止. 1 = 外部中断1(INT1 管脚 P1.7)中断使能. 当中断产生时, IE1(TCON.3)置1
1	ET0	使能定时器0中断 0 = 定时器 0 中断禁止. 1 = 定时器 0 中断使能. 当中断产生时 TF0 (TCON.5) 置 1
0	EX0	使能外部中断0 0 = 外部中断0中断禁止. 1 = 外部中断0(INT0 管脚 P3.0)中断使能. 当中断产生时, IE0(TCON.1)置1

EIE - 扩展中断使能寄存器

寄存器	SFR地址	复位值
EIE0	9BH, 页0	0000_0000 b

7	6	5	4	3	2	1	0
ET2	ESPI	EFB	EWDT	EPWM0	ECAP	EPI	EI ² C
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 = 定时器2中断使能. 当中断产生时TF2 (T2CON.7) 置 1
6	ESPI	SPI中断使能 0 = SPI 中断禁止. 1 = SPI 中断使能. 当中断产生时SPIF (SPSR.7), SPIOVF (SPSR.5), 或MODF (SPSR.4) 置 1 .
5	EFB	故障刹车中断使能 0 = 故障刹车中断禁止. 1 = 故障刹车中断使能. 当中断产生时FBF (PWM0FBD.7) 置 1.
4	EWDT	看门狗中断使能 0 = 看门狗中断禁止. 1 = 看门狗中断使能. 当中断产生时WDTF (WDCON.5) 置 1.
3	EPWM0	PWM0中断使能 0 = PWM 中断禁止. 1 = PWM 中断使能. 当中断产生时PWMF (PWM0CON0.5) 置 1.
2	ECAP	输入捕获中断使能 0 = 输入捕获中断禁止. 1 = 输入中断使能. 当中断产生时CAPF[2:0] (CAPCON0[2:0]) 置 1.
1	EPI	引脚中断使能 0 = 引脚中断禁止. 1 = 引脚中断使能. 当中断产生时PIF 相关位置 1.
0	EI ² C	I²C 中断使能 0 = I ² C 中断禁止. 1 = I ² C 中断使能. 当中断产生时SI (I2CCON.3) 或 I2TOF (I2CTOC.0) 置 1.

EIE1 – 扩展中断使能寄存器 1

寄存器	SFR地址	复位值
EIE1	9CH, 页0	0000_0000 b

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

位	名称	描述
5	EPWM3	PWM3 中断使能 0 = PWM3 中断关闭 1 = PWM3中断使能, 当中断产生时 PWM3F (PWM3CON0.5) 置1
4	EPWM2	PWM2 中断使能 0 = PWM2 中断关闭 1 = PWM2中断使能, 当中断产生时PWM2F (PWM2CON0.5) 置1
3	EPWM1	PWM1 中断使能 0 = PWM1 中断关闭 1 = PWM1中断使能, 当中断产生时 PWM1F (PWM1CON0.5) 置1
2	EWKT	WKT中断使能 0 = WKT 中断禁止. 1 = WKT 中断使能. 当中断产生时 WKTF (WKCON.4) 置 1
1	ET3	定时器3中断使能 0 = 定时器 3 中断禁止. 1 = 定时器 3中断使能. 当中断产生时 TF3 (T3CON.4) 置 1
0	ES_1	串口1中断使能 0 = 串口 1 中断禁止. 1 = 串口 1 中断使能. 当中断产生时 TI_1 (S1CON.1) 或 RI_1 (S1CON.0) 置 1

6.2.4.3 中断优先级

中断有4个优先级：3级最高，0级最低。中断源可以单独设置各自的优先级位来配置其优先级。表 6.2-4 中断优先级设置列举了4种优先级配置。低优先级中断可以被高优先级中断打断，不能被同等优先级、或更低的优先级打断。

在多个中断时，遵循以下规则：

1. 低优先级中断正在运行时，若一个高优先级中断产生，CPU转去执行高优先级中断，当高优先级中断执行完RETI后，再回到低优先级中断继续运行。当最低优先级中断执行完RETI后，控制器把运行权利交还给主程序。
2. 如果一个高优先级中断正在运行，不能被任何其他中断源打断——即使这高优先级中断，在默认优先级中比正在运行的中断优先级更高，也不能打断运行中的中断
3. 低优先级中断只有在其他中断没有执行的情况下才能被调用。然后同时，低优先级中断不能被另一个低优先级中断打断，即使这个低优先级中断，在默认优先级中比正在运行的中断优先级更高，也不能打断运行中的中断
4. 如果两个中断同时发生，优先级高的中断先执行。如果两个中断优先级相同，默认优先级高的中断先执行，这是符合默认优先级唯一的条件

默认优先级如表 6.2-4 中断优先级设置

CPU从掉电模式中唤醒，请参考章节 6.2.2.3掉电模式

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

表 6.2-4 中断优先级设置

中断源	向量地址	中断标志	使能位	默认优先级	优先级控制位	是否支持掉电模式唤醒
复位	0000H	-	总是使能	最高	-	Yes
外部中断0	0003H	IE0[1]	EX0	1	PX0, PX0H	Yes
欠压检测	0043H	BOF (BODCON0.3)	EBOD	2	PBOD, PBODH	Yes
看门狗定时器	0053H	WDTF (WDCON.5)	EWDT	3	PWDT, PWDTH	Yes
定时器 0	000BH	TF0[2]	ET0	4	PT0, PT0H	No
1°C 状态/超时	0033h	SI + I2TOF (I2TOC.0)	E1°C	5	PI2C, PI2CH	No

中断源	向量地址	中断标志	使能位	默认优先级	优先级控制位	是否支持掉电模式唤醒
ADC中断	005Bh	ADCF	EADC	6	PADC, PADCH	No
外部中断1	0013H	IE1[1]	EX1	7	PX1, PX1H	Yes
管脚中断	003BH	PIF0 to PIF7 (PIF)[3]	EPI	8	PPI, PPIH	Yes
定时器 1	001BH	TF1[2]	ET1	9	PT1, PT1H	No
串口 0	0023H	RI + TI	ES	10	PS, PSH	No
故障刹车中断	0073h	FBF (PWM0FBD.7)	EFB	11	PFB, PFBH	No
SPI	004Bh	SPIF (SPSR.7) + MODF (SPSR.4) + SPIOVF (SPSR.5)	ESPI	12	PSPI, PSPIH	No
定时器 2	002BH	TF2[2]	ET2	13	PT2, PT2H	No
定时器输入捕获	0063H	CAPF[2:0] (CAPCON0[2:0])	ECAP	14	PCAP, PCAPH	No
PWM0 中断	006BH	PWMF	EPWM0	15	PPWM, PPWMH	No
串口 1	007BH	RI_1 + TI_1	ES_1	16	PS_1, PSH_1	No
定时器 3	0083H	TF3[2] (T3CON.4)	ET3	17	PT3, PT3H	No
自唤醒定时器	008BH	WKTF (WKCON.4)	EWKT	18	PWKT, PWKTH	Yes
PWM1中断	0093H	PWM1F	EPWM1	19	PPWM1, PPWM1H	No
PWM2中断	009BH	PWM2F	EPWM2	20	PPWM2, PPWM2H	No
PWM3中断	00A3H	PWM3F	EPWM3	21	PPWM3, PPWM3H	No
SC0中断	00ABH	ACERR+BGT+TERR+TBE+ RDA (SC0IS[4:0])	SC0IE	22	PSC0, PSC0H	No
SC1中断	00B3H	ACERR+BGT+TERR+TBE+ RDA (SC1IS[4:0])	SC1IE	23	PSC1, PSC1H	No
SC2中断	00BBH	ACERR+BGT+TERR+TBE+ RDA (SC2IS[4:0])	SC2IE	24	PSC2, PSC2H	No

注:

1. 当外部中断引脚设置成边沿触发(ITx = 1), 在执行中断服务程序时中断标志位IEx 会被自动清除。当被设置成电平触发时(ITx = 0), IEx会跟随各自引脚反向电平状态变化一致, 不能通过软件控制。
2. 在执行中断服务程序时中断标志位TF0, TF1, 或TF3 会被自动清除。相反, 执行中断服务程序时TF2不会被自动清除。
3. 当引脚中断选择了电平触发, PIFn标志位反应各自通道的状态, 软件无法控制。

表 6.2-5 各级中断源特性表

IP – 中断优先级^[1]

寄存器	SFR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	BOD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中断优先级低位
1	PT0	定时器0中断优先级低位
0	PX0	外部中断0中断优先级低位

注： IP与IPH结合使用，以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

IPH – 中断优先级高位

寄存器	SFR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	掉电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： IP与IPH结合使用，以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

EIP – 扩展中断优先级

寄存器	SFR地址	复位值
EIP	EFH, 页 0	0000_0000 b

7	6	5	4	3	2	1	0
PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI ² C
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	WDT 中断优先级低位
3	PPWM	PWM 中断优先级低位
2	PCAP	引脚捕获中断优先级低位
1	PPI	引脚中断优先级低位
0	PI ² C	I ² C中断优先级低位

注： EIP与EIPH结合使用，以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

EIPH – 扩展中断优先级高位^[4]

寄存器	SFR地址	复位值
EIPH	F7H, 页 0	0000_0000 b

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

位	名称	描述
7	PT2H	定时器2中断优先级高位
6	PSPIH	SPI中断优先级高位
5	PFBH	故障刹车中断优先级高位
4	PWDTH	WDT中断优先级高位
3	PPWMH	PWM中断优先级高位
2	PCAPH	定时器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 ² C中断优先级高位

注： EIP与EIPH结合使用，以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

EIP1 –扩展中断优先级1

寄存器	SFR地址	复位值
EIP1	FEH, 页 0	0000_0000 b

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

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

注： EIP1与EIPH1结合使用，以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

EIPH1 –扩展中断优先级高位1

寄存器	SFR地址	复位值
EIPH1	FFH, 页 0	0000_0000 b

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

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

注： EIP1与EIPH1结合使用，以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

EIP2 –扩展中断优先级寄存器2

寄存器	SFR地址	复位值
EIP2	CEH, 页 2	0000_0000 b

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

位	名称	描述
2	PSC2	SC2 / UART4中断优先级低位
1	PSC1	SC1 / UART3中断优先级低位
0	PSC0	SC0 / UART2中断优先级低位

注： EIP2与EIPH2结合使用，以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

EIPH2 – 扩展中断优先级高2

寄存器	SFR地址	复位值
EIPH2	CFH, 页 2	0000_0000 b

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

位	名称	描述
2	PUART4H	UART4中断优先级高位
1	PUART3H	UART3中断优先级高位
0	PUART2H	UART2中断优先级高位

注：EIP2与EIPH2结合使用，以确定每个中断源的优先级。见表 6.2-4 中断优先级设置

6.2.4.4 中断服务

中断标志在每个系统时钟周期都会被采样，并解析它们的优先级。在同一个周期内，被采样到的中断和优先级都会被解决。如果满足执行条件将执行LCALL指令到中断代码首地址，目标地址是中断向量地址。能产生LCALL条件如下：

1. 没有相同或更高优先级中断服务程序在执行。
2. 当前查询中断标志周期正好是当前执行指令的最后一个周期。
3. 当前指令不是写任何中断使能位或优先级位且也不能是中断返回指令RETI。

如果以上有一个条件不满足，就不产生LCALL指令。在每一个指令周期都会重新检测中断标志。如果一个中断标志被置位，由于不满足上述条件而没有响应，即使后面满足上述条件，没有立即执行的中断仍然不会执行LCALL指令。这个中断标志生效，但没有进入中断服务程序，下一个指令周期需要重新检测中断标志。

处理器响应一个有效的中断，通过执行一个LCALL 指令将程序转移到中断入口地址执行中断服务程序时，不同中断源的标志可能被硬件清除，也可能不被清除。硬件LCALL与软件LCALL指令相同，执行LCALL指令时，保存程序计数器PC内容到堆栈，不保存程序状态字PSW，PC的值改为中断首地址继续执行程序直到RETI指令。在执行RETI指令时，栈顶内容加载到程序计数器PC。中断期间如果堆栈的内容被修改，处理器不会被通知。注，RET指令与RETI的不同在于，RET不会通知中断控制器中断服务已经完成，控制器认为中断服务仍在进行，中断不能再产生。

6.2.4.5 中断延迟

每一个中断源的响应时间取决于几个方面，如中断自身特点和指令的执行。在每个时钟周期，每一个中断标志和优先级都会被检测。如果有一个中断请求满足以上3个条件，硬件将自动产生LCALL指令，执行该指令需要4个机器周期。这样从中断标志置位，到执行中断服务程序最少需要5个机器周期。

如果三个条件不满足，很长的响应时间是可以预知的。如果高优先级和同等优先级中断正在执行，那么

中断延迟时间很明显取决于正在执行的中断服务程序。最大的响应时间（如果没有其他中断正在执行或没有更高优先级中断产生）是执行RET_I指令，然后下一条要执行的指令是最长的6个时钟周期的指令。从一个中断源被激活(没有检测到)，最长的反应时间是16个时钟周期。这些周期包括完成RET_I指令的5个时钟周期、完成最长指令的6个时钟周期、侦测中断1个时钟周期和完成硬件LCALL跳转到中断地址的4个时钟周期。

因此一个简单的中断系统响应时间，总是大于5个时钟周期并且不超过16个时钟周期。

6.2.4.6 外部中断管脚

外部中断引脚 $\overline{INT0}$ 和 $\overline{INT1}$ 可根据IT0 (TCON.0) 和 IT1 (TCON.2)选择边沿触发、还是电平触发。中断标志位是IE0 (TCON.1) 和 IE1 (TCON.3)。在边沿触发模式下，每个系统时钟采样 $\overline{INT0}$ 和 $\overline{INT1}$ 引脚电平。如果在一个周期中采样是高电平，下一个周期中为低电平，这个高到低的电平转换将会置位中断标志IE0或IE1。引脚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，IE0和IE1的值与 $\overline{INT0}$ 和 $\overline{INT1}$ 引脚电平相反。当中断服务程序结束后引脚若依然保持低电平，处理器会再次重头执行中断函数。 $\overline{INT0}$ 和 $\overline{INT1}$ 均支持将芯片从掉电模式唤醒。

TCON – 定时器 0 和 1 控制

寄存器	SFR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 如果 IT1 = 1 (下降沿触发), 当检测到下降沿时该标志将被硬件置位。该位将保持置1直到软件清零或在外部中断1服务程序中硬件清零。 如果 IT1 = 0 (低电平触发), 此标志是 $\overline{INT1}$ 输入信号逻辑电平的反转。软件不可控制。
2	IT1	外部中断 1 类型选择 该位选择 $\overline{INT1}$ 的中断触发类型。 0 = $\overline{INT1}$ 是低电平触发。 1 = $\overline{INT1}$ 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), 当检测到下降沿时该标志将被硬件置位。该位将保持置1直到软件清零或在外部中断0服务程序中硬件清零。 如果 IT0 = 0 (低电平触发), 此标志是 $\overline{INT0}$ 输入信号逻辑电平的反转。软件不可控。
0	IT0	外部中断 0 类型选择 该位选择 $\overline{INT0}$ 的中断触发类型。 0 = $\overline{INT0}$ 是低电平触发 1 = $\overline{INT0}$ 是下降沿触发

6.3 内存控制器

6.3.1 在应用编程 (IAP)

与RAM的实时操作不同,修改FLASH数据通常需要很长时间,不像RAM那样可以实时操作。而且擦除、编程或读取Flash数据需要遵循相当复杂的时序步骤。MS51提供了方便的FLASH编程方式,帮助用户通过IAP,重新编程FLASH内容。IAP就是通过软件实现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,通过设置IAPUEN里的相应位,使能需要升级的FLASH区域,将16位地址写入IAPAH和IAPAL,数据写入 IAPFD,命令写入 IAPCN。然后置1触发位IAPGO (IAPTRG.0受TA保护)执行IAP。此时,CPU程序计数器PC值保持不变,内建的IAP充电泵提升电压和信号时序。擦除和编程时间是内部控制的,与工作电压和频率无关。通常页擦除时间是5ms,字节编程时间是23.5 μs。IAP动作完成后,程序计数器PC继续运行后面的指令。IAPGO将被自动清零。IAPFF (CHPCON.6)是IAP错误标志,可以用来检查之前的IAP操作成功与否。通过纯软件的设置,用户可以方便的对FLASH存储器进行擦除、编程和校验。

6.3.1.1 IAP命令

MS51通过IAP可以操作APROM、LDROM或CONFIG。IAP操作模式和编程区域是由IAP的控制寄存器IAPCN决定的。

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 到 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 字节读取	00	0	0	0000	地址写入	数据读出
LDROM 字节读取	01	0	0	0000	地址写入	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 CONFIG6: 0005H	数据写入

IAP 模式	IAPCN				IAPA[15:0] {IAPAH, IAPAL}	IAPFD[7:0]
	IAPB [1:0]	FOEN	FCEN	FCTRL[3:0]		
CONFIG 字节读取	11	0	0	0000	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H CONFIG6: 0005H	数据读取
注意: 1、“X”表示无关项 2、每一页大小是128K字节。因此，地址应该是目标页的地址。						

表 6.3-1 IAP 模式及命令

6.3.1.2 IAP控制寄存器

IAP相关的寄存器。

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	IAP掉电影响 此位决定IAP的擦除和编程是否受掉电状态影响。此位只有在掉电检测使能的状态下才有效。 1 = 当V _{DD} 低于V _{BOD} 时，影响IAP擦除和编程。 0 = 在任何可工作的V _{DD} 下，IAP都可擦除和编程。

CHPCON – 芯片控制寄存器

寄存器	SFR 地址	复位值
CHPCON	9FH, 所有页TA保护	软件复位 0000_00U0 b 其他 0000_00C0 b

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

位	名称	描述
6	IAPFF	IAP 错误标志 满足以下任意条件，硬件将在IAPGO (IAPTRG.0)置位后置位该位 (1) 访问地址越界。 (2) IAPCN 命令无效。 (3) IAP 擦除或者编程更新没有使能的区域。 (4) 当BOIAP (CONFIG2.5) =1, BODEN (BODCON0.7) =1, BORST (BODCON0.2) =0时, 在V _{BOD} 电压下进行IAP 擦除和编程工作。 此位必须由软件清除。
0	IAPEN	IAP 使能 0 = 禁止IAP。 1 =使能 IAP。 当打开IAP功能时, HIRC会被打开用于时序控制。为了降低功耗时停止内部晶振, IAP操作完后才执行清除IAPEN标志指令。

IAPUEN – IAP 更新使能

寄存器	SFR 地址	复位值
IAPUEN	A5H, 0页, TA 保护	0000 _0000 b

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

位	名称	描述
7:5	-	保留
4	SPMEN	SPROM 存储空间映射使能 0 = CPU 内存地址 0xff80~0xffff 映射到APROM 1 = CPU 内存地址 0xff80~0xffff 映射到 SPROM
3	SPUEN	SPROM 升级使能(TA 保护) 0 = 禁止通过IAP擦除或编程SPROM。 1 = 使能通过IAP擦除或编程SPROM。
2	CFUEN	CONFIG 升级使能 0 = 禁止通过IAP擦除或编程CONFIG。 1 = 允许通过IAP擦除或编程CONFIG。
1	LDUEN	LDROM 升级允许 0 = 禁止通过IAP擦除或编程LDROM。 1 = 允许禁止通过IAP擦除或编程LDROM 。
0	APUEN	APROM 升级允许 0 = 禁止通过IAP擦除或编程APROM。 1 = 允许禁止通过IAP擦除或编程APROM。

IAPCN – IAP 控制寄存器

寄存器	SFR 地址	复位值
IAPCN	AFH, 0页	0011_0000 b

位	名称	描述
7:6	IAPB[1:0]	IAP 控制 此字节是IAP控制命令。详细描述见 表6.3-1 IAP模式及命令
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

IAPAH – IAP 地址 高字节

寄存器	SFR 地址	复位值
IAPAH	A7H, 0页	0000_0000 b

位	名称	描述
7:0	IAPA[15:8]	IAP 地址高位 IAPAH 包含IAP操作地址的高八位。

IAPAL – IAP 地址 低字节

寄存器	SFR 地址	复位值
IAPAL	A6H, 0页	0000_0000 b

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

位	名称	描述
7:0	IAPA[7:0]	IAP 地址低地位 IAPAL 包含IAP操作地址的低八位

IAPFD – IAP 数据寄存器

寄存器	SFR 地址	复位值
IAPFD	AEH, 0页	0000_0000 b

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

位	名称	描述
7:0	IAPFD[7:0]	IAP 数据 该字节包含将要读出或者写进flash的数据。在编程模式下，用户需要在触发ISP之前将数据写入IAPFD里；在读/校验模式下，在IAP完成后从APFD里读出数据。

IAPTRG – IAP 触发寄存器

寄存器	SFR 地址	复位值
IAPTRG	A4H, 0页, TA 保护	0000 _0000 b

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

位	名称	描述
7:1	-	保留
0	IAPGO	<p>IAP 执行</p> <p>该位设置成1, 开始执行IAP。此指令之后, CPU保持程序计数器 (PC), IAP硬件管理进程。IAP操作完成后, 程序计数器继续执行下一条指令。IAPGO位将被自动清除并一直保持0。</p> <p>触发IAP操作前, 由于硬件限制, 中断 (如果启用) 应该临时禁止。</p> <p>程序流程如下:</p> <pre> CLR EA MOV TA,#0AAH MOV TA,#55H ORL IAPTRG,#01H (SETB EA) </pre>

6.3.1.3 IAP 用户指南

IAP可以方便用户更新Flash内容。不过用户必须遵循一定的规则, 以确保IAP正确执行。否则可能引起不可预知的结果, 甚至器件损坏。此外, 下文是对正确执行IAP的一些有用建议。

1. 没有IAP操作时, 用户必须清除IAPEN (CHPCON.0)位。可以防止系统意外触发IAP。此外, IAP需要使用内部HIRC振荡器。如果选择外部时钟源, 停止IAP将会停止内部HIRC, 可以降低功耗。注: 写IAPEN受TA保护。
2. 当 LOCK 位(CONFIG0.1) 被激活, IAP 读、写或擦除仍然有效。

当进行IAP时, 中断 (如果打开)应该被临时关闭, 通过清除EA位。

当前代码执行的页不能被擦除或者编程。否则会出现不可预知的程序动作, 甚至破坏存储的程序数据。

6.3.1.4 使用Flash存储器作为数据存储器

在一般应用中, 有时一些数据需要断电后不丢失。以便读回或更改, 比如系统控制中的参数和常量。MS51支持IAP功能, 并且存储在Flash里的字节可以用MOVC指令读取, 所以很适合作为非易失数据存储器。IAP提供的擦除和编程函数, 在一个程序里擦除或编程一页中的一个或者多个字节非常容易。IAP在应用中由微控制器的软件控制。注意: Flash可写入100,000次。以下参考应用代码:

汇编例程如下:

```

;*****
; This code illustrates how to use IAP to make APROM 201h as a byte of
; Data Flash when user code is executed in APROM.
;*****
PAGE_ERASE_AP          EQU          00100010b
BYTE_PROGRAM_AP       EQU          00100001b

ORG    0000h

MOV    TA,#0Aah        ;CHPCON is TA protected
MOV    TA,#55h
ORL    CHPCON,#00000001b ;IAPEN = 1, enable IAP mode

MOV    TA,#0Aah        ;IAPUEN is TA protected
MOV    TA,#55h
ORL    IAPUEN,#00000001b ;APUEN = 1, enable APROM update

MOV    IAPCN,#PAGE_ERASE_AP ;Erase page 200h~27Fh
MOV    IAPAH,#02h
MOV    IAPAL,#00h
MOV    IAPFD,#0FFh
MOV    TA,#0Aah        ;IAPTRG is TA protected
MOV    TA,#55h
ORL    IAPTRG,#00000001b ;write '1' to IAPGO to trigger IAP process
MOV    IAPCN,#BYTE_PROGRAM_AP ;Program 201h with 55h
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, disable APROM update

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

MOV    DPTR,#201h
CLR    A
MOVC  A,@A+DPTR        ;Read content of address 201h
MOV    P0,A

SJMP  $

```

C语言例程如下:

```

//*****
//This code illustrates how to use IAP to make APROM 201h as a byte of
//Data Flash when user code is executed in APROM.
//*****
#define PAGE_ERASE_AP      0x22
#define BYTE_PROGRAM_AP    0x21

/*Data Flash, as part of APROM, is read by MOVC. Data Flash can be defined as
128-element array in "code" area from absolute address 0x0200          */

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

Main (void)
{
    TA = 0Xaa;                //CHPCON is TA protected
    TA = 0x55;
    CHPCON |= 0x01;          //IAPEN = 1, enable IAP mode

    TA = 0Xaa;                //IAPUEN is TA protected
    TA = 0x55;
    IAPUEN |= 0x01;          //APUEN = 1, enable APROM update

    IAPCN = PAGE_ERASE_AP;    //Erase page 200h~27Fh
    IAPAH = 0x02;
    IAPAL = 0x00;
    IAPFD = 0Xff;
    TA = 0Xaa;                //IAPTRG is TA protected
    TA = 0x55;
    IAPTRG |= 0x01;          //write '1' to IAPGO to trigger IAP process

    IAPCN = BYTE_PROGRAM_AP; // Program 201h with 55h
    IAPAH = 0x02;
    IAPAL = 0x01;
    IAPFD = 0x55;
    TA = 0Xaa;
    TA = 0x55;
    IAPTRG |= 0x01;          //write '1' to IAPGO to trigger IAP process

    TA = 0Xaa;                //IAPUEN is TA protected
    TA = 0x55;
    IAPUEN &= ~0x01;         //APUEN = 0, disable APROM update

    TA = 0Xaa;                //CHPCON is TA protected
    TA = 0x55;
    CHPCON &= ~0x01;         //IAPEN = 0, disable IAP mode

    P0 = Data_Flash[1];      //Read content of address 200h+1

    while(1);
}

```

6.3.1.5 在线系统编程 (ISP)

Flash存储器支持硬件编程和在应用编程(IAP)。如果产品在研发过程中或者终端用户的产品需要固件升级，采用硬件编程模式非常困难且不方便。采用ISP方式就比较简单和方便。且不需要将控制器从板上拆下来。通过软件可以对设备重新编程。因此，这使得更新应用程序固件的ISP得到广泛的应用。

用户可以开发自己的引导代码放在LDRROM里。LDRROM最大容量4KB。用户开发的引导代码可以通过并行烧录器或在电路编程器 (ICP) 下载到LRROM中。

一般来说，ISP是PC和MCU之间进行通讯。PC通过串口传输新的代码给MCU。然后引导程序接收这些数据，将这些数据通过IAP命令编程到用户代码区域。芯唐针对MS51提供ISP固件和PC端软件。这样用户可以很容易ISP通过UART端口升级代码。请访问芯唐8位微控制器网站：[芯唐 80C51 微控制器技术支持](#)。以下是简单的ISP参考代码：

汇编例程代码：

```

;*****
; This code illustrates how to do APROM and CONFIG IAP from LDRROM.
; APROM are re-programmed by the code to output P1 as 55h and P2 as aah.
; The CONFIG2 is also updated to disable BOD reset.
; User needs to configure 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                ;disable all interrupts
CALL  Enable_IAP

CALL  Enable_AP_Update
CALL  Erase_AP            ;erase AP data
CALL  Program_AP         ;programming AP data
CALL  Disable_AP_Update
CALL  Program_AP_Verify  ;verify Programmed AP data

CALL  Read_CONFIG        ;read back CONFIG2
CALL  Enable_CONFIG_Update
CALL  Erase_CONFIG       ;erase CONFIG bytes
CALL  Program_CONFIG     ;programming CONFIG2 with new data
CALL  Disable_CONFIG_Update
CALL  Program_CONFIG_Verify ;verify Programmed CONFIG2

CALL  Disable_IAP
MOV   TA, #0Aah          ;TA protection
MOV   TA, #55h           ;
ANL   CHPCON, #11111101b ;BS = 0, reset to APROM
MOV   TA, #0Aah
MOV   TA, #55h
    
```

```

    ORL    CHPCON, #80h                ;software reset and reboot from APROM

    SJMP   $

;*****
;       IAP Subroutine
;*****
Enable_IAP:
    MOV    TA, #0Aah                  ;CHPCON is TA protected
    MOV    TA, #55h
    ORL    CHPCON, #00000001b        ;IAPEN = 1, enable IAP mode
    RET

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

Enable_AP_Update:
    MOV    TA, #0Aah                  ;IAPUEN is TA protected
    MOV    TA, #55h
    ORL    IAPUEN, #00000001b        ;APUEN = 1, enable APROM update
    RET

Disable_AP_Update:
    MOV    TA, #0Aah
    MOV    TA, #55h
    ANL    IAPUEN, #11111110b        ;APUEN = 0, disable APROM update
    RET

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

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

Trigger_IAP:
    MOV    TA, #0Aah                  ;IAPTRG is TA protected
    MOV    TA, #55h
    ORL    IAPTRG, #00000001b        ;write '1' to IAPGO to trigger IAP process
    RET

;*****
;       IAP APROM Function
;*****

```

```

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 Function
;*****
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                ;disable BOD reset
    MOV    R6,A                ;temp data
    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 code
;*****
AP_code:
    DB     75h,0B1h, 00h          ;OPCODEs of "MOV    P0M1,#0"
    DB     75h,0ACh, 00h        ;OPCODEs of "MOV    P3M1,#0"

```

```

DB      75h, 90h, 55h          ;OPCODEs of "MOV    P1,#55h"
DB      75h,0A0h,0Aah        ;OPCODEs of "MOV    P2,#0Aah"
DB      80h,0Feh             ;OPCODEs of "SJMP  $"

END
    
```

6.3.2 在电路编程(ICP)

通过在线电路编程 (ICP) 编程Flash。如果产品在开发中,或在终端客户的产品需要固件升级,采用硬件编程模式非常困难且不方便。Flash存储器支持在电路编程(ICP)。不需要将控制器从板上拆下来。ICP方式同样允许客户在量产电路板上编程设备,在设备装配完成后再编程,这样允许设备编程最新的固件或定制化固件。

执行ICP功能,仅需要3个引脚 \overline{RST} 、ICP_DAT及ICP_CLK。 \overline{RST} 用于进入或者退出ICP模式,ICP_DAT为数据输入输出引脚,ICP_CLK为编程时钟输入引脚。用户必须在电路板上保留 \overline{RST} 、ICP_DAT、ICP_CLK V_{DD}及VSS。

芯唐提供MS51的ICP编程工具Nu_Link,通过新唐ICP编程器,用户可轻松使用ICP。。ICP编程器是芯唐根据MCU的电气特性专门设计的、高效稳定的编程器。更多细节,请访问芯唐8位微控制器网站: [芯唐 80C51 微控制器技术支持](#)。

6.3.3 片上调试器 (OCD)

6.3.3.1 概述

MS51内嵌片上调试 (OCD) 功能,这为用户提供低成本的调试方法,并且ML51的每一种封装都适用。OCD具有完整的调试过程控制流程,包括8个硬件断点、单步执行、全速运行、非侵入命令的内存访问。OCD系统并不占用任何本地内存,也不和偏上外设共享资源。

当OCDEN (CONFIG0.4)配置为0, LOCK (CONFIG0.1)为1时, OCD才能有效。如果芯片已经加密, OCD就不能工作。OCD系统使用两线串行接口, ICE_DAT和ICE_CLK, 让目标设备和调试控制器建立通讯。ICE_DAT是输入/输出引脚,调试时用作数据传输, ICE_CLK是输入引脚,调试时传输数据同步用的时钟。P2.0/nRESET引脚用来控制OCD模式进入和退出。MS51的OCD和ICP功能是共享这3个引脚。

MS51使用ICE_DAT、ICE_CLK和P2.0/nRESET引脚与OCD系统交互。在设计系统应用OCD时,必须遵循下面的一些限制条件:

1. P2.0/nRESET配置成外部复位引脚时,不能直接连接到VDD上,并且要连接外部电容。
2. P2.0/nRESET配置成P2.0输入引脚时,断开所有外部输入源。
3. 所有的外部复位源必须断开。
4. 所有与ICE_DAT和ICE_CLK连接的外围器件都必须断开。

6.3.3.2 OCD的限制条件

由于MS51功能丰富,而引脚有限,一个引脚上可能有多个功能。使用OCD系统时会牺牲掉一些功能。主要有有如下限制条件:

1. P2.0/nRESET引脚需要配置成OCD模式。因此,不管是P2.0输入还是外部复位源都不能使用。
2. ICE_DAT与 P1.6共享一个引脚。因此,不管是I/O功能还是其他功能都不能使用。
3. ICE_CLK 与P0.2共享一个引脚。因此,不管是I/O功能还是其他功能都不能使用。
4. 当系统处于待机或掉电模式时,因为部分外设时钟已经停止,所以任何访问都是无效的。读访

问可能返回无用数据，写访问可能失败。

5. 不能关闭HIRC，因为OCD需要这个时钟监视内部工作状态。在调试模式下，关闭HIRC指令将不起作用。CPU进入掉电模式时HIRC会继续运行。

MS51的OCD系统还有另外一个限制，就是正在运行用户程序时不能执行非侵入命令。非侵入指令允许用调试器访问MCU存储单元和状态控制寄存器。一个读或写存储器或控制寄存器必须在MCU停止条件下进行，产生停止条件是在与硬件断点匹配后或者不运行。

CONFIG0

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

出厂默认值：1111 1111b

位	名字	描述
5	OCDPWM	OCD停止状态下，PWM输出状态。 该位决定当OCD停止CPU时，PWM的输出状态。 1 = PWM 输出三态模式 (Tri-state)。 0 = PWM 持续输出。 注：只要当PWM通道对应PIO位设置为10时，这位才有效。
4	OCDEN	OCD 使能 1 = 禁止OCD. 0 = 使能OCD 注：当 MCU运行在OCD调试模式并且OCDEN = 0，硬件故障复位被禁止，仅仅HardF标志被置位。

6.3.4 96位序列号

出厂前，每颗MS51芯片都会预烧96位唯一序列号。用户只能通过IAP命令读取序列号。详见章节6.3.1.IAP命令。

6.4 通用I/O(GPIO)

6.4.1 I/O模式

MS51有30个I/O管脚，其中29个可位寻址的通用I/O，引脚分成4组通用输入输出端口 P0 ~ P3，P20只能输入。每组独立的控制寄存器，读写意义不同：写是写入输出锁存寄存器，读是读的引脚电平。可定义为双向(标准8051结构)，推挽，仅输入和开漏四种模式，由寄存器 Pxm1 和 Pxm2 配置，见下图。复位后为仅输入模式。

PnM1.X ^[1]	PnM2.X ^[1]	I/O 类型
0	0	准双向
0	1	推挽
1	0	输入
1	1	开漏输出

注 1: n = 0~3, X = 0~7

表 6.4-1 I/O 模式配置

所有引脚都可由寄存器PxS 配置为TTL电平或施密特输入，施密特触发输入有更好的抗干扰能力。所有的I/O引脚可通过软件选择位控制，斜率输出能力，由PxSR 配置输出速度：低速或高速，复位缺省为低速。

例:

```
P0M1 |= 0x40;
P0M2 &= 0xBF; // P0.6 配置为仅输入模式
```

6.4.1.1 准双向模式

准双向模式，作为8051的传统引脚模式，即可以输入、也可以输出数据。当引脚输出高时，内部上拉MOS仅能提供小电流，引脚电压可以被外部拉低。当引脚输出低时， \bar{P} MOS有很强的灌电流能力。引脚有两个上拉MOS，一个仅能提供小电流的弱上拉MOS，另一个是可以提供大电流的强上拉MOS。

强上拉MOS只在输出数由0变1时，导通两个系统时钟，然后关闭。这样做是为了加快电平上升时间。GPIO的结构，见下图。

第二种上拉为“强上拉”。这种上拉用于在准双向口引脚上，加速端口电平由逻辑0转为逻辑1的转换速度。当这种情况发生时，打开强上拉用两个CPU时钟的时间快速地将端口引脚拉高。然后就关闭，特弱上拉继续保持该端口引脚为高，准双向引脚结构如下所示。

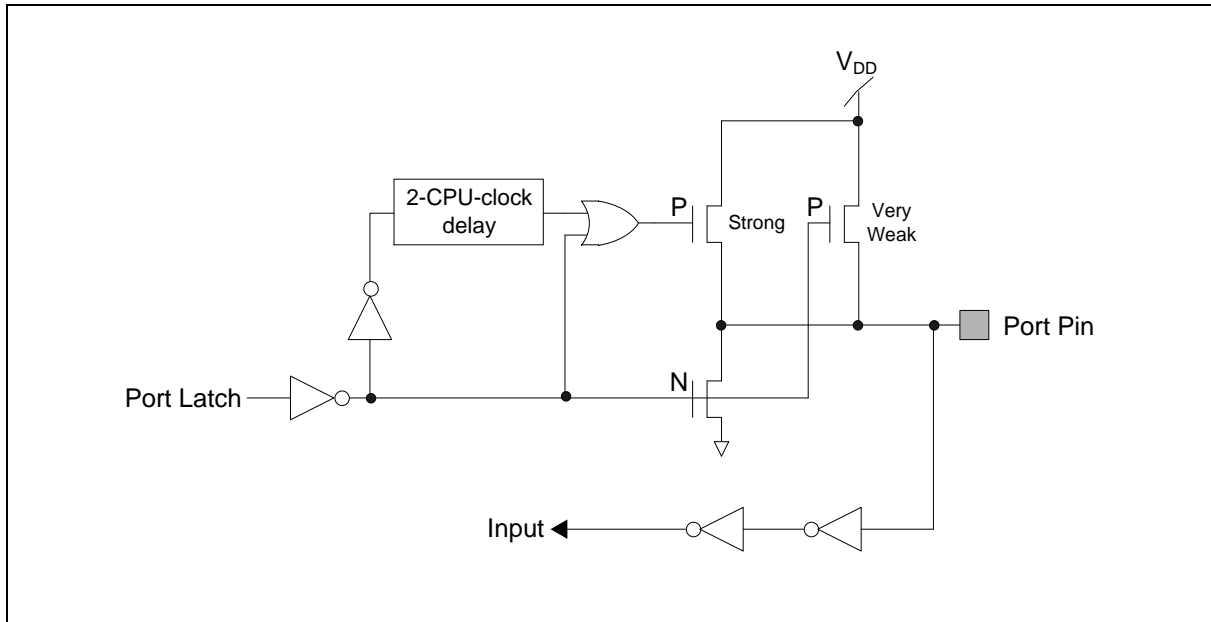


图 6.4-1 准双向结构图

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。当端口锁定为1时，提供持续的强上拉。推挽输出模式用于需要从端口输出大电流时的应用。

推挽模式的灌电流和拉电流能力都很强，结构见下图。

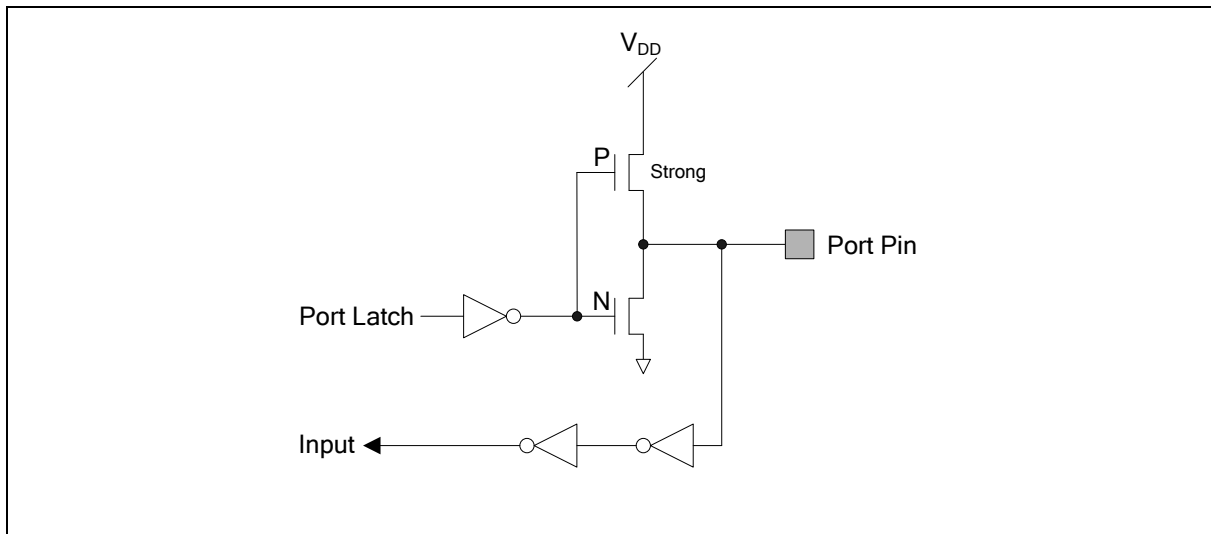


图 6.4-2 推挽模式结构图

6.4.1.3 仅输入模式

准双向模式引脚，提供真实的高阻输入路径，虽然可以在输出高时作为输入脚，但是它需要相对强的输入源。但是当外部输入为0时，毕竟要消耗电流，而仅输入模式，可以做到无电流消耗，如果是准双向模式，逻辑0时总是消耗来自VDD的电流。另外注意，仅输入模式引脚若悬空，会因电压飘呼不定增加电流消耗，所以仅输入模式的引脚，不能悬空，必须由外部驱动到高、或低电平。

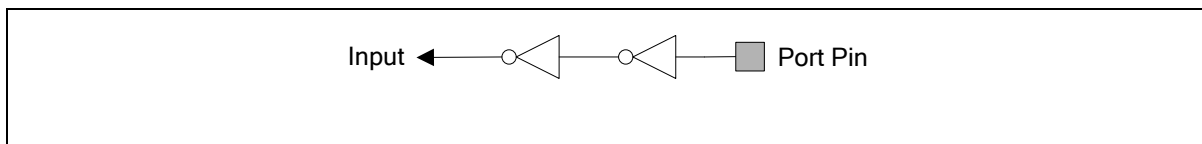


图 6.4-3 仅输入模式结构图

6.4.1.4 开漏模式

开漏输出配置关闭所有内部上拉，当端口锁定为逻辑0时，仅打开驱动端口的下拉晶体管。当端口锁存为逻辑1时，它就和输入模式一样。通常用于I2C输出线上，开漏引脚需要加一个外部上拉电阻，典型连一个电阻到VDD。用户需要注意的是，开漏模式输出逻辑1的时候，应该由外部设备或电阻提供一个确定的电平。悬浮的引脚在掉电状态下会引起漏电

开漏模式只有下MOS，输出0时，下MOS导通，输出1时，引脚就和输入一样成高阻状态了。这时外部电路必须让引脚电压为高，否则会增加额外的电流消耗。这种引脚常用于I2C通信。

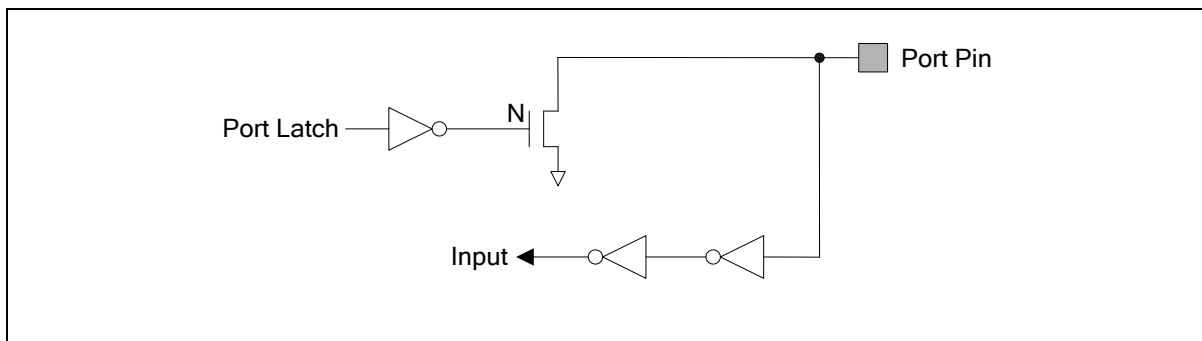


图 6.4-4 开漏模式

6.4.2 引脚中断(PIT)

MS51 每个引脚都可产生中断，用于检测管脚电平状态，如按键或键盘是否按下。但最多配置8个引脚中断，任何按键按下时通过边沿或电平触发产生一个管脚中断事件。可配置为电平或边沿中断，还可唤醒空闲或掉电状态的CPU。

每个管脚的中断使能和正反向特性通过寄存器 PIPEN 和 PINEN 可以配置每个中断的极性和使能，PICON 配置中断是边沿模式、还是电平模式，寄存器PIF的8位指示是哪个引脚了生了中断，PIF由硬件置位，只能软件清除。

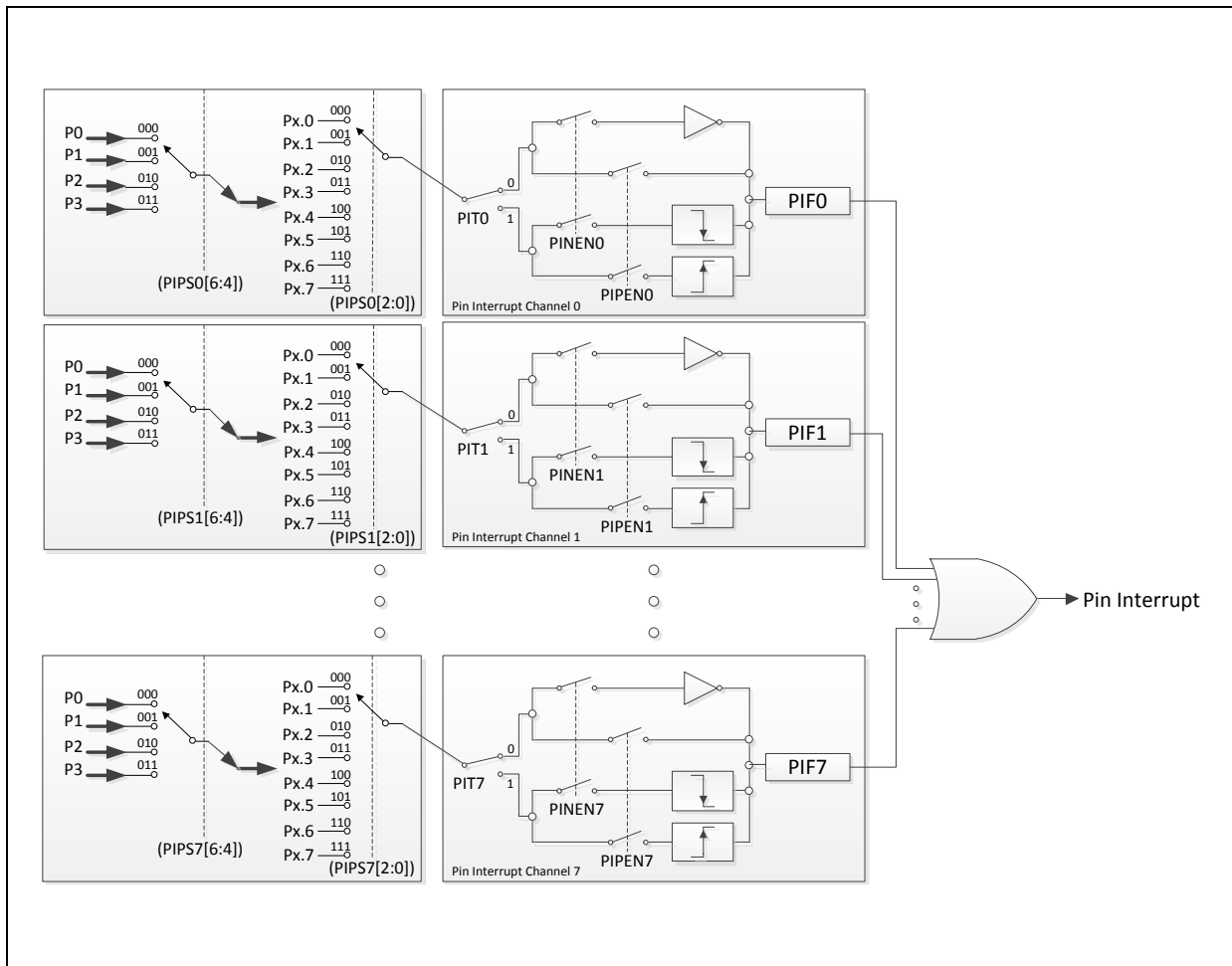


图 6.4-5 引脚中断框图

管脚中断一般用于检测接口设备（键盘板或键盘）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，管脚中断能将设备从掉电模式下唤醒。

6.4.3 控制寄存器

MS51有许多I/O控制寄存器提供灵活的各种应用。和I/O端口相关的SFRs可以分类成四组：

输入输出控制, 输出模式控制 输入类型和灌电流控制, 输出速度控制。

6.4.3.1 GPIO输入输出数据控制

这些寄存器是I/O输入输出数据缓存。读获取I/O输入的数据。读引脚电平，或写输出到引脚的电平。可位寻址。

Pn – 端口 (可位寻址)

寄存器	SFR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	0011_1111 b
P3	B0H, 所有页, 可位寻址	1111_1111 b

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

位	名称	描述
7:0	Pn[7:0]	端口 n n 最多为8.

6.4.3.2 GPIO 模式控制

这些寄存器用于控制GPIO 模式，每个引脚都可单独配置为四种模式之一： 输入，准双向，推挽输出，开漏输出。

复位缺省为输入模式.

PnM1.X	PnM2.X	I/O 类型
0	0	准双向
0	1	推挽输出
1	0	输入
1	1	开漏输出

PnM1 – 模式选择 1

寄存器	SFR地址	复位值
P0M1	B1H, 0页	1111_1111 b
P1M1	B3H, 0页	1111_1111 b
P2M1	89H, 0页	0011_1111 b
P3M1	ACH, 0页	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 0 模式选择 1

PnM2 – 模式选择 2

寄存器	SFR地址	复位值
P0M2	B2H, 0页	0000_0000 b
P1M2	B4H, 0页	0000_0000 b
P2M2	8AH, 2页	0000_0000 b
P3M2	ADH, 0页	0000_0000 b

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

位	名称	描述
7:0	PnM2[7:0]	端口 0 模式选择2 注意: PxM1 和 PxM2 的组合决定了引脚的模式. 详见表 6.4-1 I/O 模式配置.

6.4.3.3 输入类型选择

引脚可单独配置为TTL或斯密特输入，注意：寄存器 PxS 要切换到页1

PnS – 端口 n 斯密特输入使能

寄存器	SFR地址	复位值
P0S	B1H, 1页	0000_0000 b
P1S	B3H, 1页	0000_0000 b
P2S	8CH, 2页	0000_0000 b
P3S	ACH, 1页	0000_0000 b

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

位	名称	描述
7:0	PnS[7:0]	Pn 斯密特输入使能 0 = Pn.x 是TTL输入电平 1 = Pn.x是斯密特输入电平.

6.4.3.4 输出速度控制

MS51可单独控制管脚输出斜率。默认情况下，管脚采用普通斜率模式。当用户切换到高速斜率模式时，每个管脚斜率可看到显著变化。注更改PxSR寄存器需要将SFR页切换到页1缺省为低速。

PnSR –引脚速度配置

寄存器	SFR地址	复位值
P0SR	B2H, 1页	0000_0000 b
P1SR	B4H, 1页	0000_0000 b
P2SR	8BH, 1页	0000_0000 b
P3SR	ADH, 1页	0000_0000 b

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

位	名称	描述
7:0	PnSR[7:0]	Pn 输出速度 0 = Pn.x 低速. 1 = Pn.x 高速

PICON – 引脚中断类型选择

寄存器	SFR地址	复位值
PICON	E9H, 0页,	0011_0100 b

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

位	名称	描述
7	PIT7	通道7 类型选择. 0 = 电平中断. 1 = 边沿中断
6	PIT6	通道6 类型选择. 0 = 电平中断. 1 = 边沿中断
5	PIT5	通道5 类型选择. 0 = 电平中断. 1 = 边沿中断
4	PIT4	通道4 类型选择. 0 = 电平中断. 1 = 边沿中断
3	PIT3	通道3 类型选择. 0 = 电平中断. 1 = 边沿中断
2	PIT2	通道2 类型选择. 0 = 电平中断. 1 = 边沿中断
1	PIT1	通道1 类型选择. 0 = 电平中断. 1 = 边沿中断
0	PIT0	通道0 类型选择. 0 = 电平中断. 1 = 边沿中断

PINEN – 负极性中断使能.

寄存器	SFR地址	复位值
PINEN	EAH, 0页	0000_0000 b

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

位	名称	描述
n	PINENn	引脚负极性中断使能 电平还是边沿依然由 PICON选择 0 = 低电平/下沿中断禁止. 1 =低电平/下沿中断使能.

PIPEN – 正极性中断使能.

寄存器	SFR地址	复位值
PIPEN	EBH, 0页	0000_0000 b

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

位	名称	描述
n	PIPENn	引脚正极性中断使能 电平还是边沿依然由 PICON选择 0 = 高电平/上沿中断禁止. 1 = 高电平/上沿中断使能.

PIF – 中断标志位

寄存器	SFR地址	复位值
PIF	ECH, 0页	0000_0000 b

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

位	名称	描述
n	PIFn	引脚中断标志位 若配置为边沿中断，此位由硬件置1，由软件清0。 若配置为电平中断，此位的值与引脚电平相反。软件写无效。

PIPSn – 中断控制

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

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

位	名称	描述
6:4	PSEL[2:0]	中断GPIO组选择 000 = P0. 001 = P1. 010 = P2. 011 = P3 其它 = 保留
2:0	BSEL[2:0]	引脚中断位选择 000 = Pn.0. 001 = Pn.1 010 = Pn.2 011 = Pn.3. 100 = Pn.4. 101 = Pn.5. 110 = Pn.6. 111 = Pn.7. n 是组号, 由 PSEL[2:0]选择。

6.5 定时器

6.5.1 定时器/计数器0/1

6.5.1.1 概述

定时器/计数器0/1 是两个16位定时计数器，每个都由两个8位的寄存器组成16计数寄存器。对于定时器/计数器0，TH0/1是高位值，TL0/1是低位值，计数模式由 TCON 和 TMOD 可以配置定时器/计数器0和1的模式。

寄存器TMOD中的位C/T配置来选择定时器或计数器功能。TMOD.2配置定时器器/计数器0，TMOD.6 配置定时器1。

将它们设置为定时器后，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计数能力定时器模式是对系统时钟或其12分频或增强型直接是系统时钟。计数，是否分频，定时器0 由T0M (CKCON.3) 配置，定时器1 由T1M (CKCON.4)配置。计数模式是对引脚 T0、T1 的下沿计数，上个系统时钟检测到T0/1是高，下一个时钟检测到为低，就认为引脚有一个下沿，计数一次。

定时器0/1 可配置为溢出时、引脚T0/1输出电平翻转功能，同一个管脚可以用于T0和T1的输入计数也可以用于定时器的翻转输出，由寄存器CKCON中的位T0OE 和 T1OE配置。首次使能此功能定时器未溢出时，引脚输出高。为确保此模式功能，1位应该被清除并且选择系统时钟作为定时器的时钟源

注意，由于定时器的高低字节是分开读写的，若读写高低字节中间，低字节进位高字节值改变，就会出错，所以，建议模式0和模式1读写前，先清0控制位TR0 (TR1)停止计数。否则，产生不可预料的结果。

6.5.1.2 模式 0 (13位一次定时模式)

在模式 0，定时器/计数器是13位的计数器。TH0 (TH1) 是计数的高8位，低5位在TL0 (TL1)的低5位。TL0 (TL1) 的高3位无效可忽略。TR0 (TR1) =1，并且GATE=0或者 INT0 (INT1)=1时，开始计数。Gate 可用于控制定时器测量引脚 INT0 (INT1)的脉冲时间。计数到 1FFFH 回 0000H时，TF0 (TF1)会置1，若使能了中断，会申请中断。

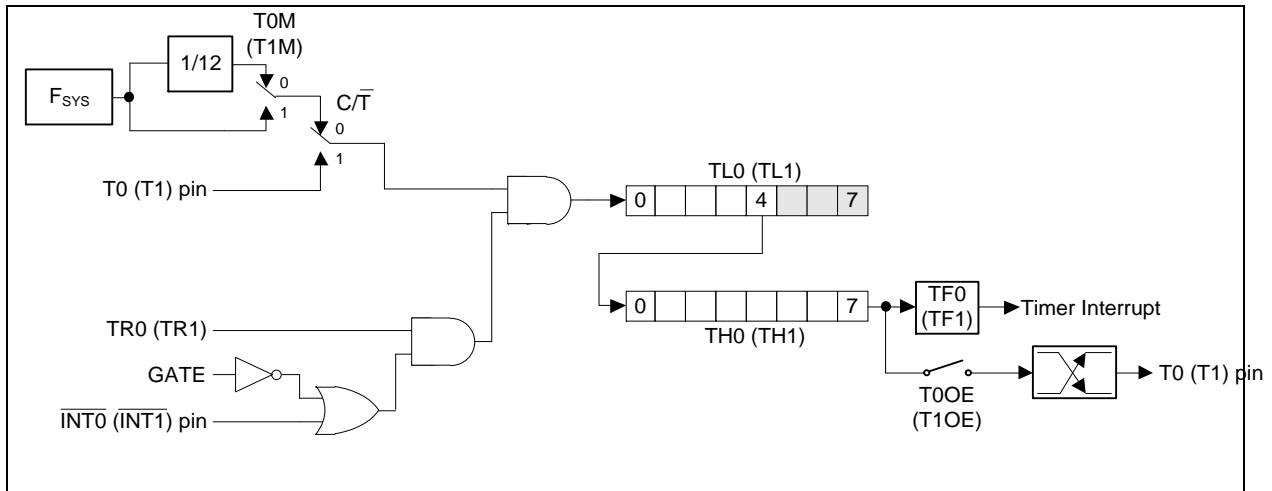


图 6.5-1 定时器/计数器 0/1 模式 0

6.5.1.3 模式 1 (16位一次定时模式)

模式 1 除了是16位计数外，其它与模式0相同。模式1计数到FFFFH回0，并置1标志位TF0 (TF1)会置1，若使能了中断，会申请中断。

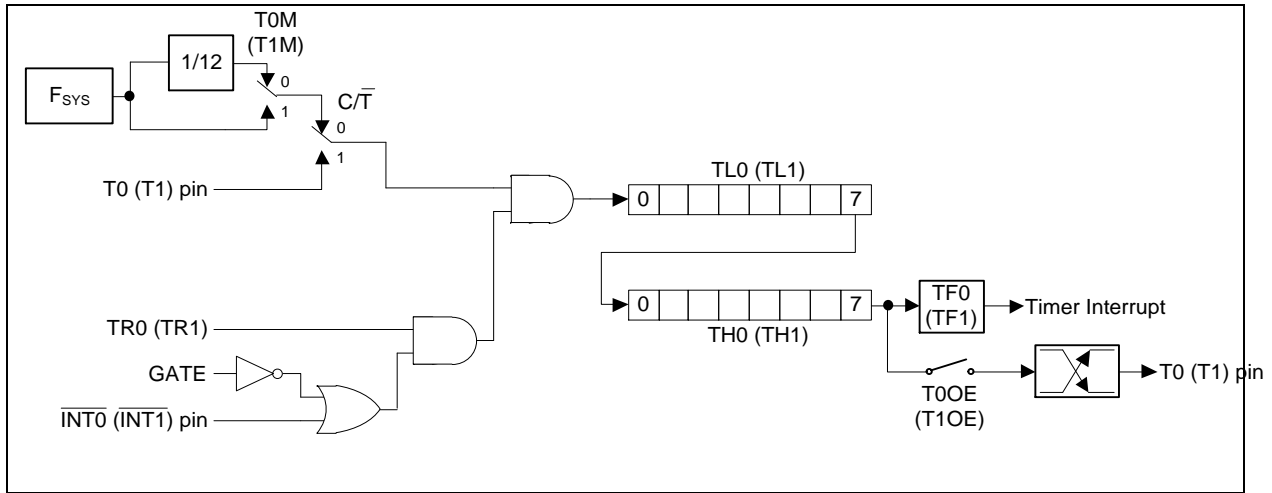


图 6.5-2 定时器/计数器 0/1 模式 1

6.5.1.4 模式 2 (8位自动装载、周期定时模式)

模式2下定时器/计数器为自动重装模式。这种模式，低字节TL0 (TL1) 做为一个8位计数器，高字节TH0 (TH1) 是重载的周期值，计数寄存器 TL0 (TL1) 溢出后，寄存器TCON中的位 TF0 (TF1) 置 1，并且 TL0 (TL1)被装入TH0 (TH1)的值，继续计数。这种模式适合作为 UART 的波特率产生器。注意：仅定时器1可做UART的波特率发生器。和模式0/1一样，TR0 (TR1) =1，并且GATE=0或者 INT0 (INT1)=1时，启动计数开始。

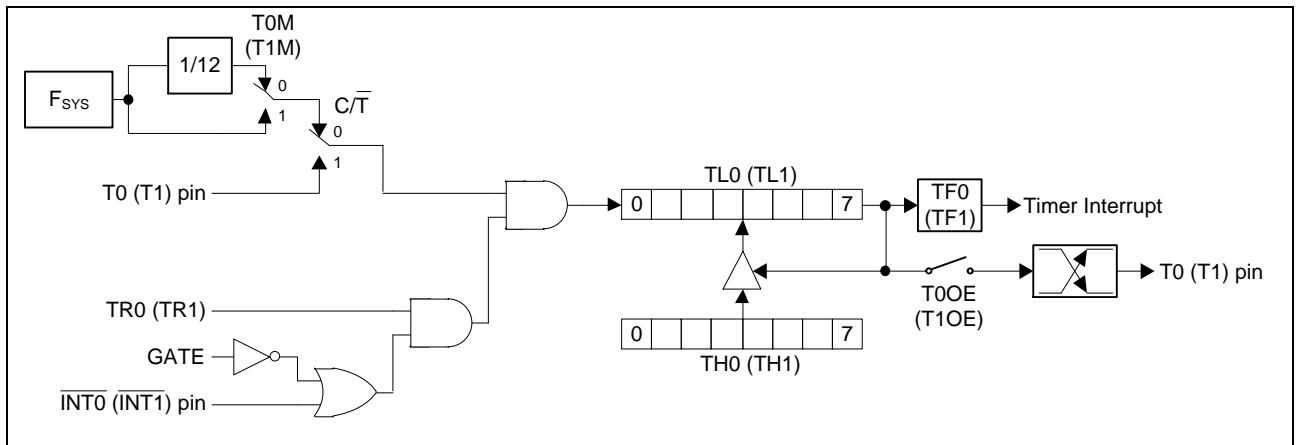


图 6.5-3 定时器/计数器 0/1 模式 2

6.5.1.5 模式 3 (双8位定时模式)

定时器0 和 定时器1的模式3操作方式不同。对定时器/计数器1来说模式3会将其停用；对定时器/计数器0来说，此模式下，TL0 和 TH0 被当做两个独立的8位计数器。TL0 用定时器0 的控制位C/T，GATE, TR0, INT0, 和 TF0。TL0 还可以由C/T (TMOD.2)控制对引脚T0的下沿计数。TH0 被强制为8位计数并由TR1 和 TF1控制。当需要额外的8位定时器时可以使用模式3。当定时器0配置为模式3时，定时器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自己。此时定时器1依然可以配置为模式0, 1或2工作，只是灵活性受到了限制，不能再用控制位TF1 和 TR1。可以使用 GATE, INT1 引脚和 T1M控制，可以当做串口波特率发生器，或者无需中断的定时器。

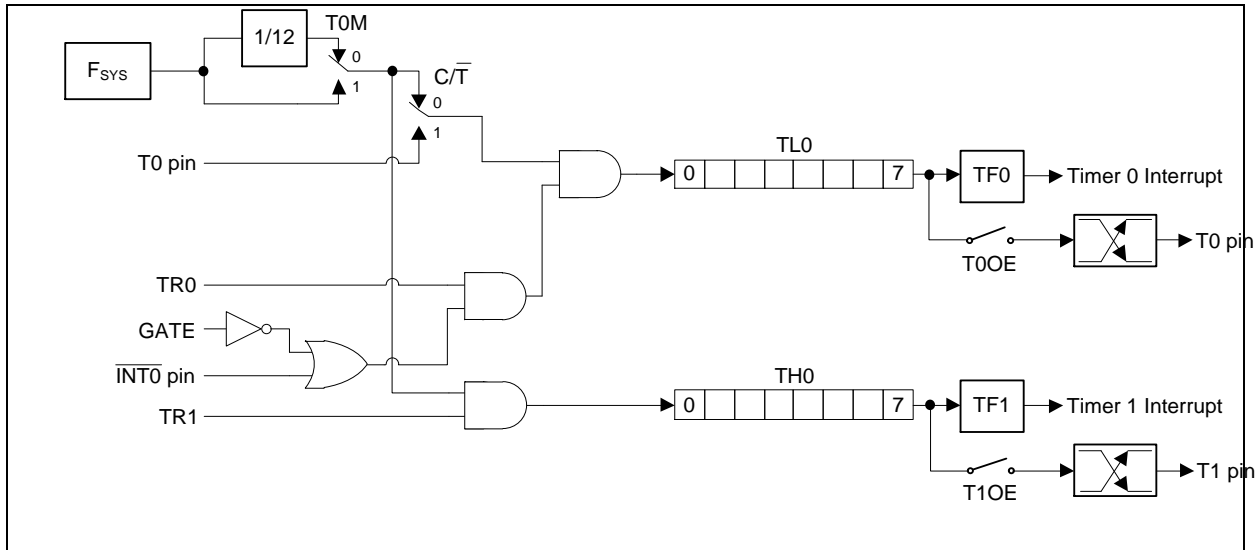


图 6.5-4 定时器/计数器 0 模式 3

6.5.1.6 与定时器有关的引脚控制寄存器

TMOD – 定时器0 模式1

寄存器	SFR地址	复位值
TMOD	89H, 0页	0000_0000b

7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/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 0 = 定时器1 在TR1 = 1 时计数, 与INT1 无关 1 = 定时器1 在TR1 = 1 并且INT1 = 1时开始计数.
6	C/T	定时器1 定时或计数选择 0 = 定时器1 对系统时钟计数 1 = 定时器1 对引脚T1的下沿计数.
5	M1	定时器1 模式选择

位	名称	描述		
4	M0	M1	M0	模式
		0	0	模式0: 13位定时计数模式
		0	1	模式1: 16位定时计数模式
		1	0	模式2: 8位从 TH1自动重载
		1	1	模式3: 定时器1 停止
3	GATE	定时器0 门控 0 = 定时器0 在TR0 = 1 时计数, 与 $\overline{INT0}$ 无关 1 = 定时器0 在TR0 = 1 并且 $\overline{INT0} = 1$ 时开始计数		
2	C/T	定时器0 定时或计数选择 0 = 定时器0 对系统时钟计数 1 = 定时器0 对引脚T0的下沿计数.		
1	M1	定时器0 模式选择		
0	M0	M1	M0	模式
		0	0	模式0: 13位定时计数模式
		0	1	模式1: 16位定时计数模式
		1	0	模式2: 8位从 TH0自动重载
		1	1	模式3: TLO和TH0做为两个8位定时器用

TCON – 定时器0/1 控制

寄存器	SFR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	读/写	读/写	读/写	读/写

位	名称	描述
7	TF1	定时器1 溢出标志 定时器溢出时置1, 执行中断函数后硬件清0, 可被软件清0或置1
6	TR1	定时器1计数控制 0 = 定时器1 停止计数, TH1 和 TL1保留当前值。 1 = 定时器1 使能计数
5	TF0	定时器0 溢出标志 定时器溢出时置1, 执行中断函数后硬件清0, 可被软件清0或置1
4	TR0	定时器 0计数控制 0 = 定时器1 停止计数, TH1 和 TL1保留当前值。 1 = 定时器1 使能计数。
3	IE1	INT1中断 标志 若 IT1 = 1 (下沿中断), 引脚有下沿就置1, 执行中断函数后由硬件清0, 或软件清0. 若 IT1 = 0 (低电平中断), 引脚 $\overline{INT1}$ 的反值, 软件无法控制
2	IT1	INT1 中断类型配置 0 = $\overline{INT1}$ 低电平中断. 1 = $\overline{INT1}$ 下沿中断
1	IE0	INT0 中断标志 若 IT0 = 1 (下沿中断), 引脚有下沿就置1, 执行中断函数后由硬件清0, 或软件清0. 若 IT0 = 0 (低电平中断), 引脚 $\overline{INT0}$ 的反值, 软件无法控制
0	IT0	INT0 中断类型配置 0 = $\overline{INT0}$ 低电平中断. 1 = $\overline{INT0}$ 下沿中断

TL0 – 定时器0低字节值

寄存器	SFR地址	复位值
TL0	8AH, 0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 低字节值

TH0 – 定时器0 高字节值

寄存器	SFR地址	复位值
TH0	8CH, 0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 高字节值

TL1 – 定时器1低字节值

寄存器	SFR地址	复位值
TL1	8BH, 0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低字节值

TH1 – 定时器1高字节值

寄存器	SFR地址	复位值
TH1	8DH, 0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 高字节值

CKCON – 计数时钟控制

寄存器	SFR地址	复位值
CKCON	8EH, 0页	0000_0000b

7	6	5	4	3	2	1	0
-	PWMCKS	-	T1M	T0M	T0OE	CLOEN	-
-	读/写	-	读/写	读/写	读/写	读/写	-

位	名称	描述
4	T1M	定时器1 时钟分频选择 0 = 对系统时钟的12分频计数 1 = 对系统时钟计数
3	T0M	定时器0时钟分频选择 0 = 对系统时钟的12分频计数 1 = 对系统时钟计数
2	T0OE	定时器0 输出使能 0 = 禁止引脚T0的输出. 1 = 使能引脚T0的输出, 对引脚计数模式无效.

6.5.2 定时器2 和输入捕获

6.5.2.1 概述

定时器2 是高低字节TH2和TL2级联的、有8种分频系数选择的、16位定时器，配备两个比较寄存器RCMP2H 和 RCMP2L。定时器2的比较模式和重装载模式由CM/RL2 (T2CON.0)配置。定时器2 还配备了三个通道的输入捕获模块，捕获时间分别存放于C0H/C0L，C1H/C1L和C2H/C2L中。TR2 (T2CON.2) = 1 使能计数时钟。见下图。

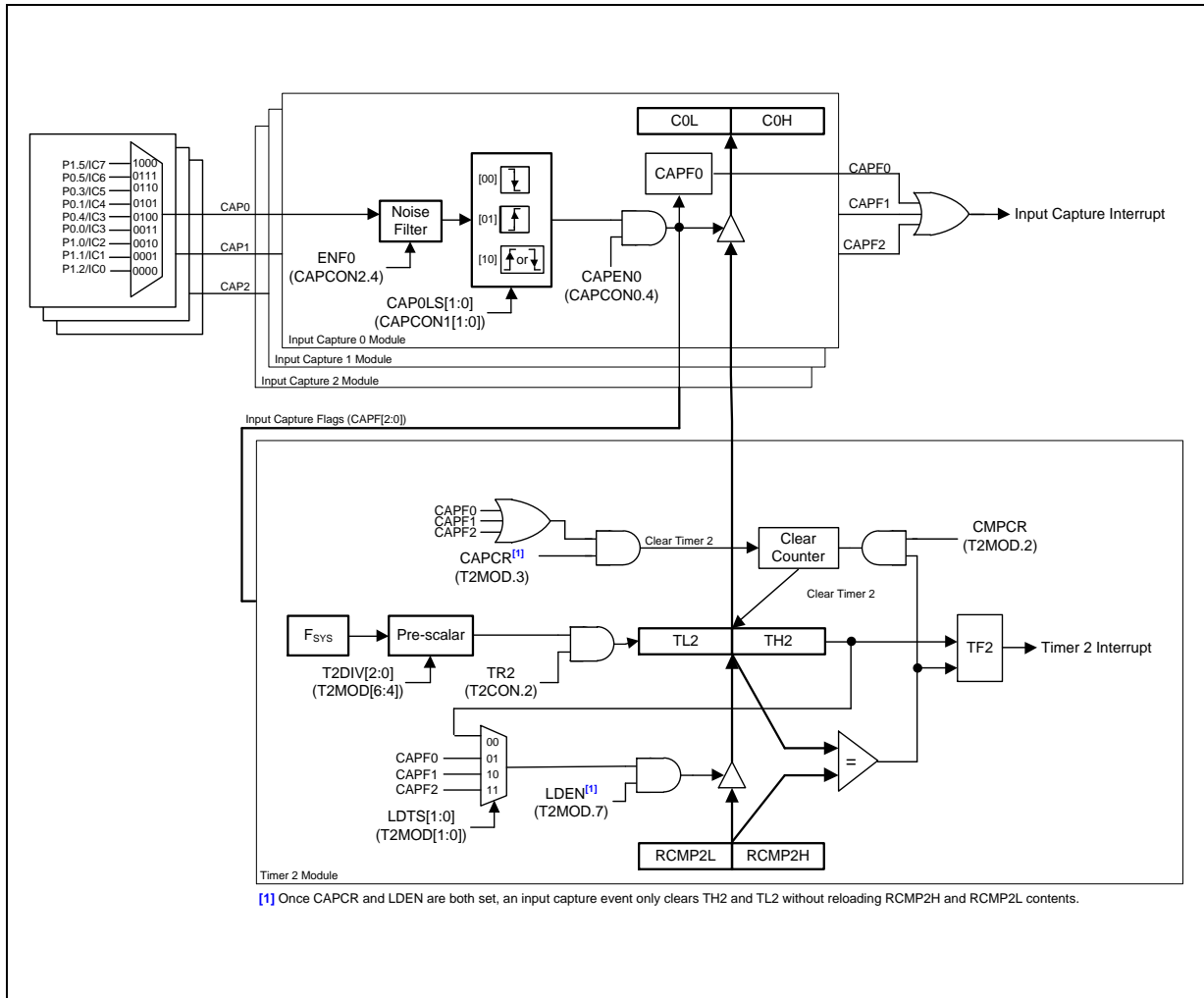


图 6.5-5 定时器 2 框图

6.5.2.2 自动装载模式

CM $\overline{RL2}$ =0，定时器2 就配置为自动装载模式，位LDEN置1时，一旦装载事件出现，RCMP2H和RCMP2L 的值就会装载到TH2和 TL2。装载事件可以是定时器2 溢出、或者是由LDTS[1:0] (T2MOD[1:0])配置的输入捕获事件之一。注意：CAPCR (T2MOD.3) 置1时，输入捕获事件只会清0寄存器TH2 和 TL2，不会加载 RCMP2H 和 RCMP2L 。

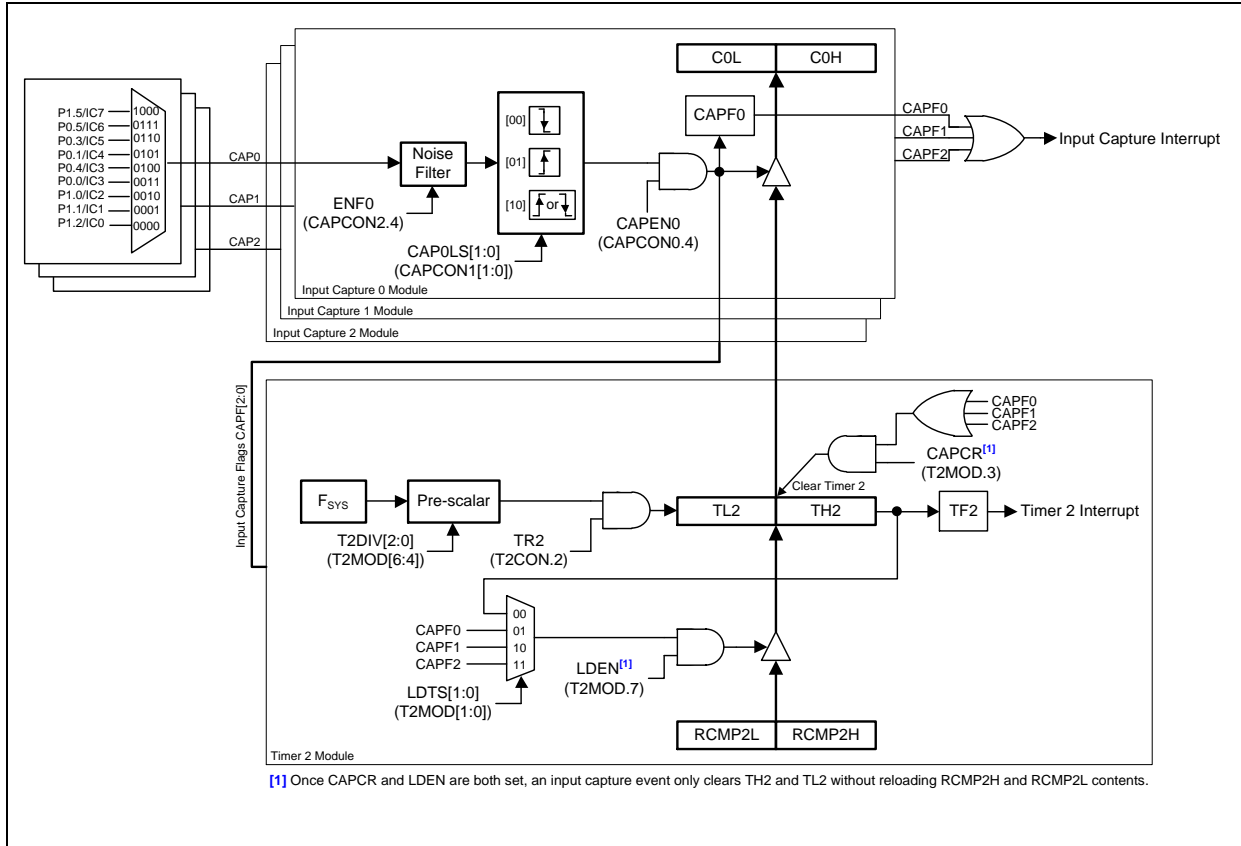


图 6.5-6 定时器 2 自动装载和输入捕获框图

6.5.2.3 比较模式

CM $\overline{RL2}$.=1时定时器2配置为比较模式，TH2/TL2计数值与RCMP2H/RCMP2L相同时，标志位TF2 (T2CON.7) 将置1。

CMPCR (T2MOD.2) 置1时，比较匹配事件会清0 定时器 2。

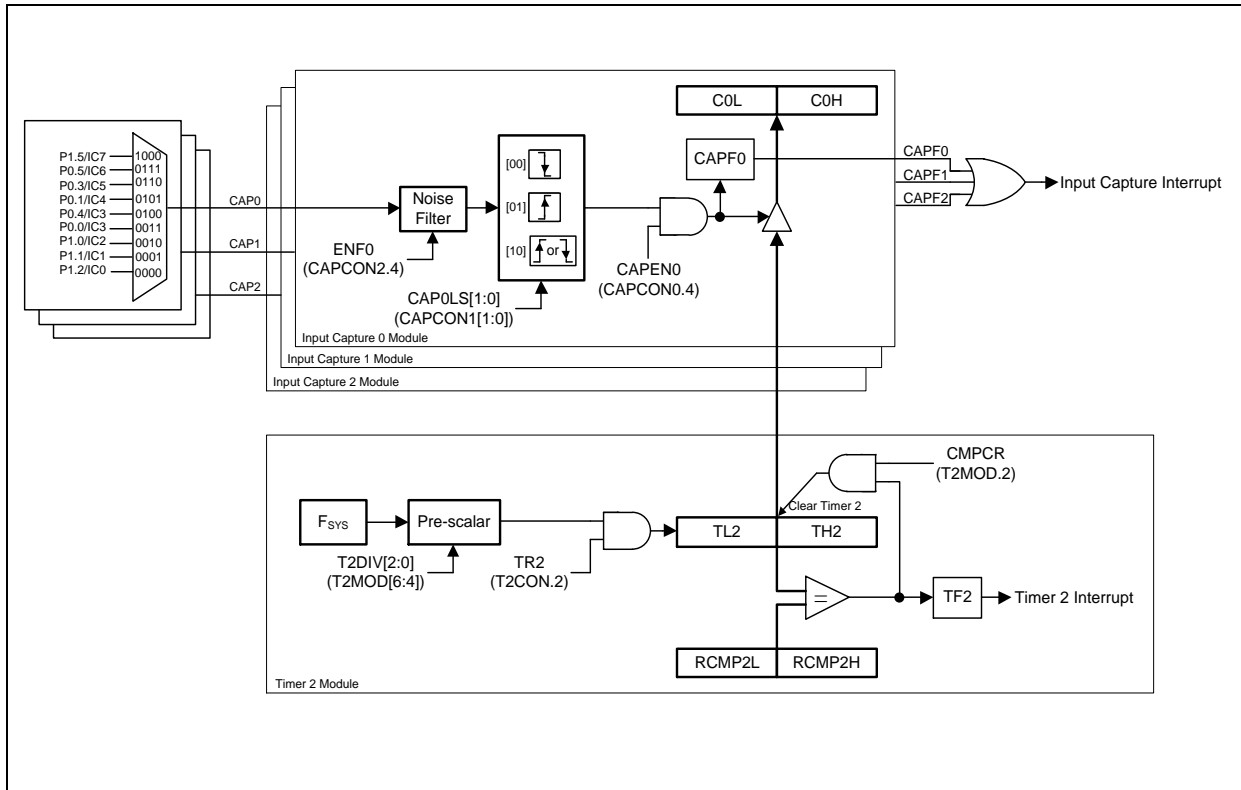


图 6.5-7 定时器 2 比较模式和捕获模式框图

6.5.2.4 输入捕获模块

三个输入捕获模块(CAP0, CAP1, 和 CAP2)由CAPCON0~2配置。各自独立的噪声滤波电路由 ENF0~2 (CAPCON2[6:4])使能, 滤除短于4个系统时钟的脉冲。每个通道都有独立的边沿检测电路, 由CAPCON1的相关位配置, 可以捕获上沿, 下沿, 和双沿都捕获。当然, 别忘了捕获使能位CAPEN0~2 (CAPCON0[6:4])。

当捕获通道使能, 并且发生了配置的边沿事件, 定时器2的计数值 TH2/TL2 的当前值, 就存入CnH/CnL, 同时置位标志 CAPFn (CAPCON0.n)。若ECAP (EIE0.2) 和EA 都置1了, 就会去执行中断函数, 三个捕获事件共用一个中断函数。哪个通道发生捕获, 可查寻标志位CAPFn, 此位要软件清0。

位CAPCR (CAPCON2.3)便于计算脉冲宽度, 此位若置1, 发生捕获时, TH2/TL2存入CnH/CnL后就清0, 这样就不必再做减法求脉宽了。

6.5.2.5 定时器2 控制寄存器

T2CON – 定时器2 控制

寄存器	SFR地址	复位值
T2CON	C8H, 所有页, 可位寻址	0000_0000 b

7	6	5	4	3	2	1	0
TF2	-	-	-	-	TR2	-	CM/RL2
读/写	-	-	-	-	读/写	-	读/写

位	名称	描述
7	TF2	定时器2 溢出标志 定时器 2 溢出或计数到比较值此位置1, 若全局中断使能, 会去执行中断函数。此位硬件不清0, 需由软件清0
5:3	-	保留
2	TR2	定时器 2 运行使能 0 = 禁止定时器 2计数, TH2 和 TL2保留当前值。 1 = 使能 定时器 2.
1	-	保留
0	CM/RL2	定时器2比较或自动装载模式选择 0 = 自动装载模式. 1 = 比较模式

T2MOD – 定时器2 模式

寄存器	SFR地址	复位值
T2MOD	C9H, 0页	0000_0000 b

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

位	名称	描述
7	LDEN	自动装载使能 0 = 禁止 RCMP2H/RCMP2L 往TH2/TL2 中装载 1 = 使能 RCMP2H/RCMP2L 往TH2/TL2 中装载
6:4	T2DIV[2:0]	定时器2 预分频系数选择 000 = 1/1. 001 = 1/4. 010 = 1/16. 011 = 1/32. 100 = 1/64. 101 = 1/128. 110 = 1/256. 111 = 1/512.
3	CAPCR	捕获清0使能 引位仅重装模式有效，发生捕获事件后，计数值 TH2/TL2 装入 RCMP2H/RCMP2L ，同时执行以下动作 0 =捕获事件发生后 定时器 2 保持计数。 1 =捕获事件发生后定时器 2 清0后开始计数
2	CMPCR	比较清0使能 此位仅比较模式有效 0 = 比较事件发生后 定时器2保持计数 1 =比较事件发生后 定时器 2 清0后开始计数.
1:0	LDTS[1:0]	重装触发选择 00 = 定时器2 溢出就重装. 01 = 输入捕获事件 0 触发重装 10 = 输入捕获事件 1 触发重装. 11 = 输入捕获事件 2 触发重装.

RCMP2L- 定时器2 重装/比较低字节

寄存器	SFR地址	复位值
寄存器	SFR地址	复位值

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 重装或比较的低字节 定时器2配置为比较模式时，此值是比较值的低字节，自动装载模式时是重装值的低字节。

RCMP2H – 定时器2重装/比较高字节

寄存器	SFR地址	复位值
RCMP2H	CBH, 0页	0000_0000 b

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

位	名称	描述
7:0	RCMP2H[7:0]	定时器2 重装或比较高字节 定时器2配置为比较模式时，此值是比较值的高字节， 自动装载模式时是重装值的高字节。

TL2 – 定时器2 低字节

寄存器	SFR地址	复位值
TL2	CCH, 0页	0000_0000 b

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

位	名称	描述
7:0	TL2[7:0]	定时器2 低字节 TL2 是定时器16位的低字节

TH2 – 定时器2 高字节

寄存器	SFR地址	复位值
TH2	CDH, 0页	0000_0000 b

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

位	名称	描述
7:0	TH2[7:0]	定时器2 高字节 TH2 是定时器16位的高字节

注意: TH2 TL2 是分开读写的, 为防止读写出错, 请在读写前, 置位TR2停止计数。

CAPCON0 – 捕获控制 0

寄存器	SFR地址	复位值
CAPCON0	92H, 0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	捕获通道2使能 0 = 禁止捕获通道2. 1 = 使能捕获通道2.
5	CAPEN1	捕获通道1使能 0 = 禁止捕获通道1. 1 = 使能捕获通道1.
4	CAPEN0	捕获通道0使能 0 = 禁止捕获通道0. 1 = 使能捕获通道0
3	-	保留
2	CAPF2	通道 2 捕获标志 通道2发生捕获事件时, 此位置1, 需软件清0.
1	CAPF1	通道 1 捕获标志 通道1发生捕获事件时, 此位置1, 需软件清0
0	CAPF0	通道 0 捕获标志 通道0发生捕获事件时, 此位置1, 需软件清0

CAPCON1 –捕获控制1

寄存器	SFR地址	复位值
CAPCON1	93H, 0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]	捕获通道2 触发边沿选择 00 = 下沿. 01 = 上沿. 10 = 双沿. 11 =保留.
3:2	CAP1LS[1:0]	捕获通道1 触发边沿选择 00 = 下沿. 01 = 上沿. 10 = 双沿. 11 =保留.
1:0	CAP0LS[1:0]	捕获通道0 触发边沿选择 00 = 下沿. 01 = 上沿. 10 = 双沿. 11 =保留.

CAPCON2 –捕获控制2

寄存器	SFR地址	复位值
CAPCON2	94H, 0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	通道2的噪声滤波器使能位 0 = 禁止通道2的噪声滤波 1 = 使能通道2的噪声滤波.
5	ENF1	通道1的噪声滤波器使能位 0 = 禁止通道1的噪声滤波 1 = 使能通道1的噪声滤波.
4	ENF0	通道0的噪声滤波器使能位 0 = 禁止通道0的噪声滤波 1 = 使能通道0的噪声滤波..

CnL –捕获低字节, n = 0,1,2

寄存器	SFR地址	复位值
C0L	E4H, 0页	0000_0000 b
C1L	E6H, 0页	0000_0000 b
C2L	EDH, 0页	0000_0000 b

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

位	名称	描述
7:0	CnL[7:0]	捕获16位时间值的低字节

CnH – 捕获高字节, n = 1,2,3

寄存器	SFR地址	复位值
C0H	E5H, 0页	0000_0000 b
C1H	E7H, 0页	0000_0000 b
C2H	EEH, 0页	0000_0000 b

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

位	名称	描述
7:0	CnH[7:0]	捕获16位时间值的高字节

CAPCON3 –捕获控制3

寄存器	SFR地址	复位值
CAPCON3	F1H, 0页	0000_0000 b

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

位	名称	描述
[7:4]	CAP1[3:0]	捕获通道1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 其它 = P1.2/IC0
[3:0]	CAP0[3:0]	捕获通道0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 其它 = P1.2/IC0

CAPCON4 –捕获控制4

寄存器	SFR地址	复位值
CAPCON4	F2H, 0页	0000_0000 b

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

位	名称	描述
[3:0]	CAP2[3:0]	捕获通道2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 其它 = P1.2/IC0

6.5.3 定时器 3

6.5.3.1 概述

定时器3 是一个16位自动装载、上计数的定时器，由T3PS[2:0] (T3CON[2:0]) 配置分频系数，重装载值 RH3 和 RL3 决定了溢出周期，设定好重装载值后，置1位 TR3 (T3CON.3) 开始计数。计到FFFFH再装载 RH3,RL3, 并且置1位 TF3 (T3CON.4) ，如果 ET3 (EIE1.1) 是1允许中断，将执行中断函数，TF3 在进入中断函数里被硬件清0.

定时器3 可做 UART波特率发生器。详见 章节6.9.3波特率。

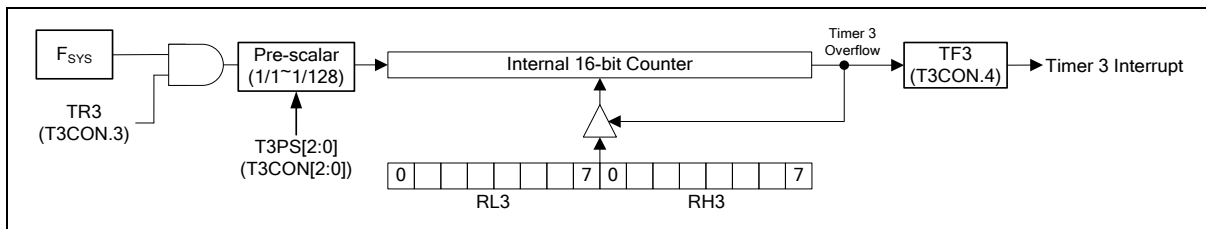


图 6.5-8 定时器 3 框图

6.5.3.2 定时器3 的控制寄存器

T3CON – 定时器3 控制寄存器

寄存器	SFR地址	复位值
T3CON	C4H, 0页	0000_0000 b

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

位	名称	描述
4	TF3	定时器3 溢出标志 计数溢出时此位置1，执行中断函数后硬件清0。此位也可由软件清0或置1
3	TR3	定时器3 运行控制 0 = 定时器3 停止状态。 1 = 定时器3 计数状态。 注意：只能在停止状态(TR3 = 0)写RH3 RL3，否则可能计数进位引起错误。
2:0	T3PS[2:0]	定时器3 预分频配置。 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 加载值低字节

寄存器	SFR地址	复位值
RL3	C5H, 0页	0000_0000 b

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

位	名称	描述
7:0	RL3[7:0]	定时器3加载值低字节 就是周期值低字节

RH3 – 定时器3 加载值高字节

寄存器	SFR地址	复位值
RH3	C6H, 0页	0000_0000 b

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

位	名称	描述
7:0	RH3[7:0]	定时器 3 加载值高字节 就是周期值高字节

6.6 脉冲宽度调制(PWM)

6.6.1 概述

PWM 应用广泛，可以控制电机，风扇，背光调节或LED亮度调节，低通滤波后还可以当做DAC使用。

PWM0 针对电机控制做了特别设计，三对输出，周期和占空比数值16位可调，控制单相或三相无刷直流电机、或三相交流电机非常方便。六路 PWM 可配置为独立模式，互补模式，或同步模式，互补模式死区可调。波形和中断可配置为中心对齐或边沿对齐模式。

PWM1/2/3，可分别配置16位的周期和占空比。这三对都可配置为独立、互补或同步模式，波形和中断点可配置为中心或边沿对齐模式。

6.6.2 引脚配置

PWM	引脚	控制寄存器 1			控制寄存器2		
		寄存器名	位名	值	寄存器名	位名	值
PWM0_CH0	P1.2	PIOCON0	PIO12	1	AUXR4[1:0]	PWM1C0P	00
	P3.3	PIOCON2	PIO33	1	-	-	-
PWM0_CH1	P1.1	PIOCON0	PIO11	1	AUXR4[3:2]	PWM1C1P	00
	P1.4	PIOCON1	PIO14	1	AUXR4[3:2]	PWM1C1P	00
PWM0_CH2	P0.5	PIOCON1	PIO05	1	AUXR4[5:4]	PWM2C0P	00
	P1.0	PIOCON0	PIO10	1	-	-	-
PWM0_CH3	P0.4	PIOCON1	PIO04	1	AUXR4[7:6]	PWM2C1P	00
	P0.0	PIOCON0	PIO00	1	AUXR4[7:6]	PWM2C1P	00
PWM0_CH4	P0.1	PIOCON0	PIO01	1	AUXR5[1:0]	PWM3C0P	00
PWM0_CH5	P0.3	PIOCON0	PIO03	1	AUXR5[3:2]	PWM3C1P	00
	P1.5	PIOCON1	PIO15	1	AUXR5[3:2]	PWM3C1P	00
PWM1_CH0	P2.3	PIOCON2	PIO23	1	AUXR4[1:0]	PWM1C0P	01
	P1.2	PIOCON0	PIO12	1			10
PWM1_CH1	P2.2	PIOCON2	PIO22	1	AUXR4[3:2]	PWM1C1P	01
	P1.4	PIOCON1	PIO14	1			10
	P1.1	PIOCON0	PIO11	1			11
PWM2_CH0	P2.1	PIOCON2	PIO21	1	AUXR4[5:4]	PWM2C0P	00
	P1.0	PIOCON0	PIO10	1			01
	P0.5	PIOCON1	PIO05	1			10
PWM2_CH1	P3.0	PIOCON2	PIO30	1	AUXR4[7:6]	PWM2C1P	00
	P3.1	PIOCON2	PIO31	1			01
	P0.0	PIOCON0	PIO00	1			10
	P0.4	PIOCON1	PIO04	1			11
PWM3_CH0	P3.2	PIOCON2	PIO32	1	AUXR5[1:0]	PWM3C0P	01
	P0.1	PIOCON0	PIO01	1			10
	P1.7	PIOCON1	PIO17	1			-
PWM3_CH1	P3.4	PIOCON2	PIO34	1	AUXR5[3:2]	PWM3C1P	01
	P1.5	PIOCON1	PIO15	1			10
	P0.3	PIOCON0	PIO03	1			11

表 6.6-1 PWM 引脚定义和使能寄存器表

6.6.3 结构框图

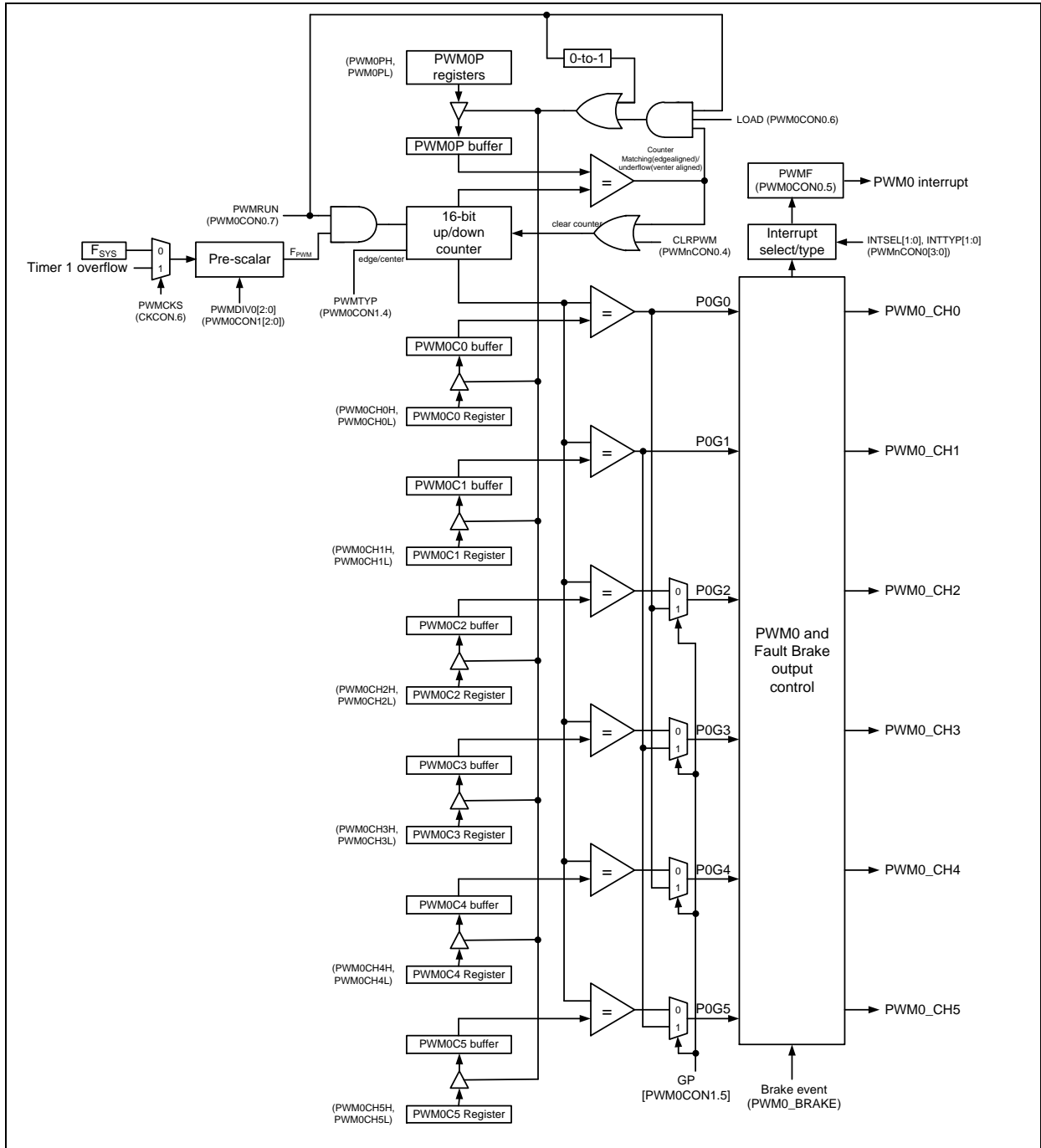


图 6.6-1 PWM0 框图

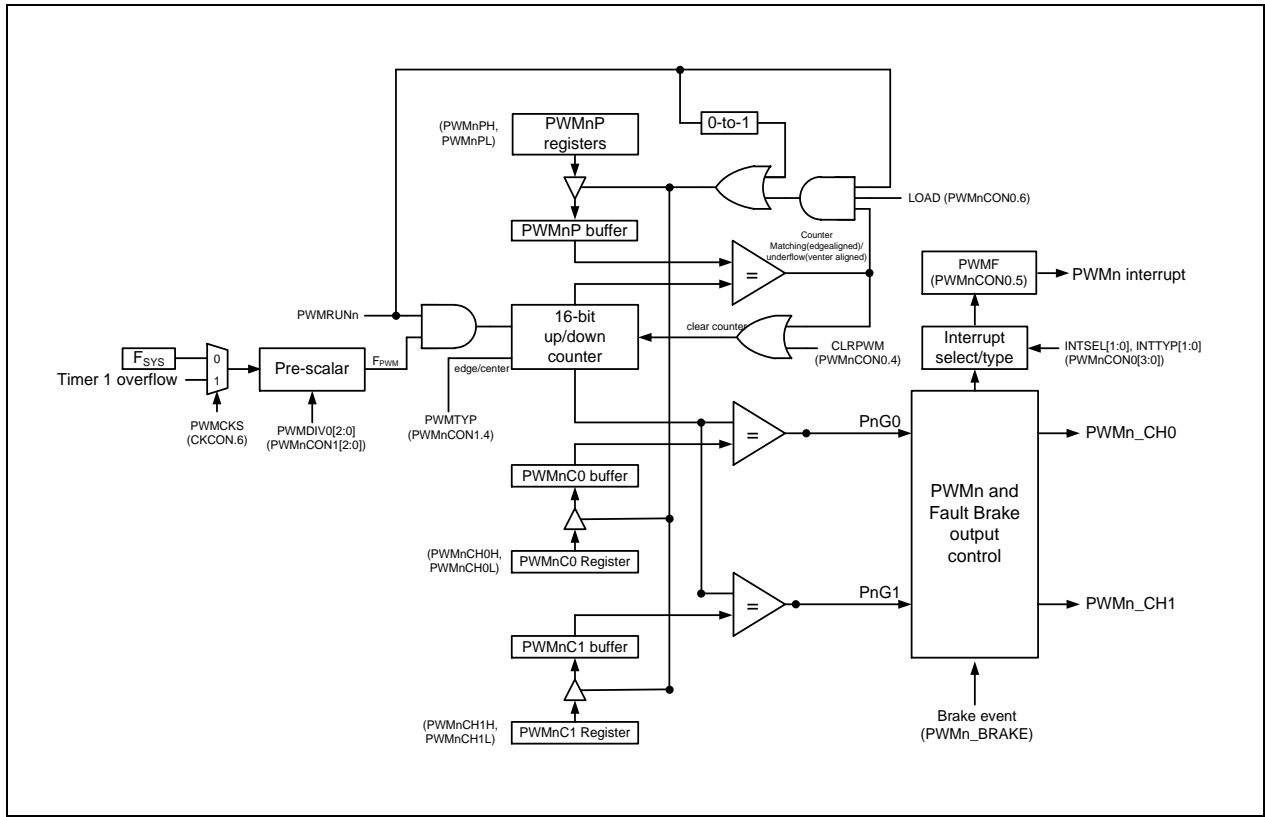


图 6.6-2 PWM1/2/3 框图

6.6.4 功能描述

6.6.4.1 PWM 发生器

PWM 的时钟可选系统时钟或 定时器 1 的溢出频率，再经 1/1~1/128 分频。周期由 {PWMnPH, PWMnPL} 配置，n=0,1,2,3，每组由一个周期计数器。PWM0 有六个占空比寄存器。

只有 PWM0 有方便电机控制的组模式，由 GP (PWM0CON1.5) 配置。这种模式 {PWM0C0H, PWM0C0L} 和 {PWM0C1H, PWM0C1L} 就决定了六路的占空比了。这种模式下 {PWM0C2H, PWM0C2L} 和 {PWM0C4H, PWM0C4L} 的值与 {PWM0C0H, PWM0C0L} 相同，{PWM0C3H, PWM0C3L} 和 {PWM0C5H, PWM0C5L} 的值与 {PWM0C1H, PWM0C1L} 相同。注意，使用 PWM0 别忘了把引脚配置为 PWM 功能。

PWM0 产生的六路 PWM 信号被命名为 P0G0, P0G1, P0G2, P0G3, P0G4 和 P0G5。信号的输出电路控制 PWM 的模式、插入死区，同时做输出消隐控制和紧急停止和极性控制，最后由引脚输出。

PWM1/2/3 各自输出两路 PWM 信号。输出电路决定 PWM 模式，输出消隐控制和极性控制，然后由引脚输出。

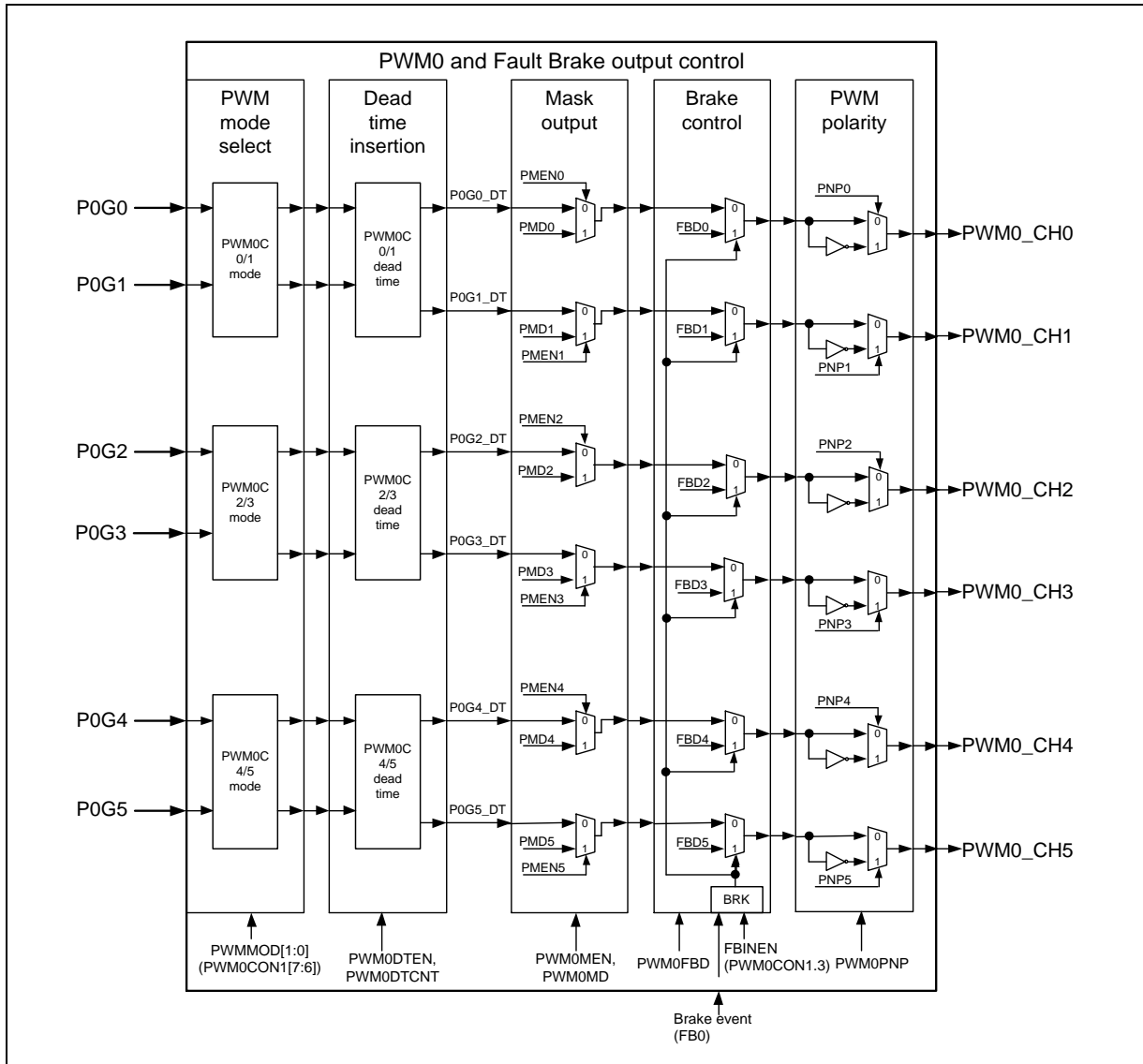


图 6.6-3 PWM0 输出电路框图

初始化PWM 首先置位 CLRPWM (PWMnCON0.4) 清0计数器，以确保上计数准确，然后配置 {PWMnPH, PWMnPL} 和 {PWMnCxH, PWMnCxL}，然后置位 PWMnRUN 计数开始。PWM引脚开始输出波形。周期值{PWMnPH, PWMnPL} 和 占空比值{PWMnCxH, PWMnCxL} 有双缓存，写入的数值，只有在 LOAD (PWMnCON0.6) 为0时才被更新，以防止输出不完整波形。

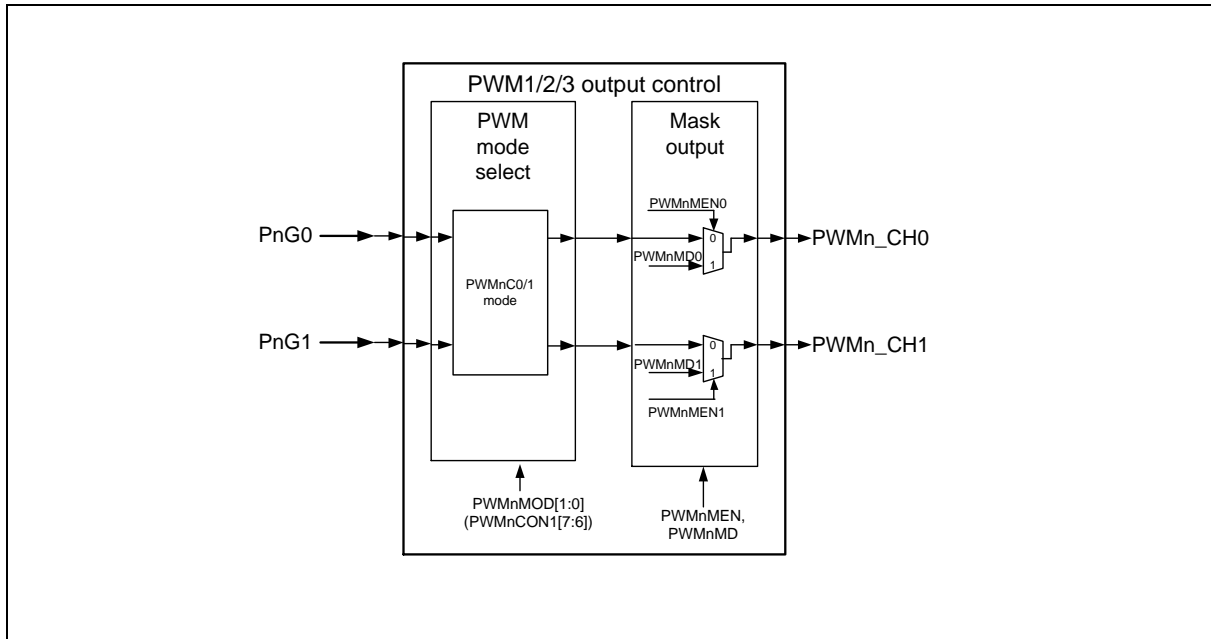


图 6.6-4 PWM1/2/3 控制框图

注意: 更新周期或占空比, 必须等LOAD=0指示更新完成后, 否则结果不可预测。

6.6.4.2 PWM 类型

两种类型: 边沿对齐和中心对齐, 由PWMTYP (PWMnCON1.4)配置。

6.6.4.3 边沿对齐

计数器从 0000H 上计数到 {PWMnPH, PWMnPL}, 然后再回0。当计数值与占空比值 {PWMnCxBH, PWMnCXL}相等时, 控制信号输出低(输出控制电路之前的信号)。波形如下

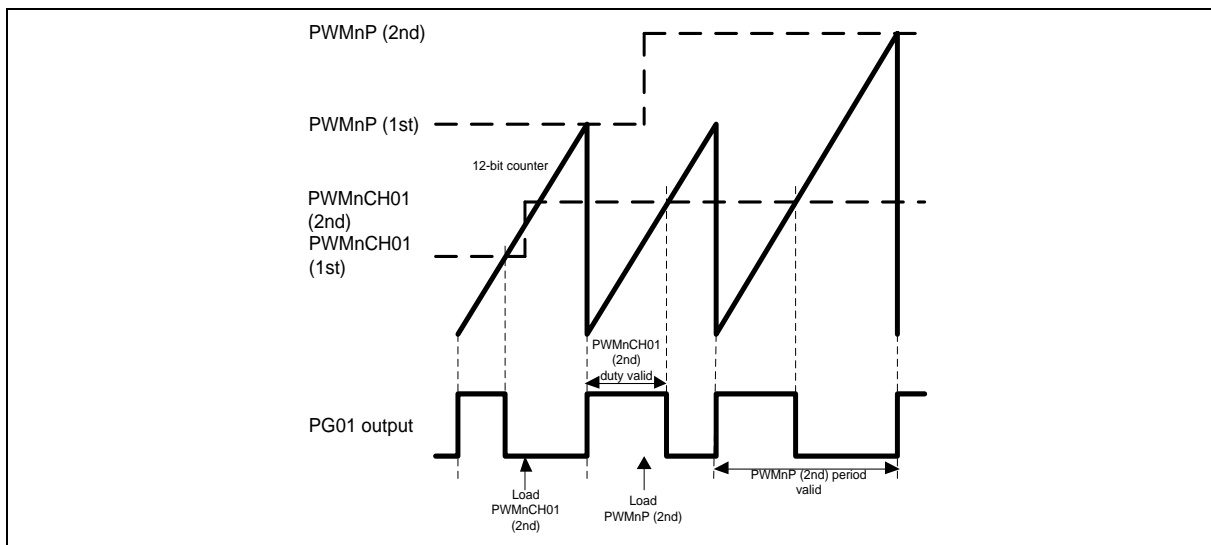


图 6.6-5 PWM 边沿对齐波形图

输出频率和占空比公式为:

$$\text{PWM 频率} = \frac{F_{\text{PWM}}}{\{PWMnPH, PWMnPL\} + 1} \quad (F_{\text{PWM}} \text{ 是时钟源经 PWMDIV分频后的频率}).$$

$$\text{PWM 高电平占空比} = \frac{\{PWMnCHxH, PWMnCHxL\}}{\{PWMnPH, PWMnPL\} + 1}.$$

6.6.4.4 中心对齐类型

计数器从 0000H 上计数到 {PWMnPH, PWMnPL} 然后再下计数到 0000H，上计数与占空比寄存器 {PWMnCxH, PWMnCxL} 的值相等时，输出低，下计数与占空比寄存器值相等时输出高。波形如下。

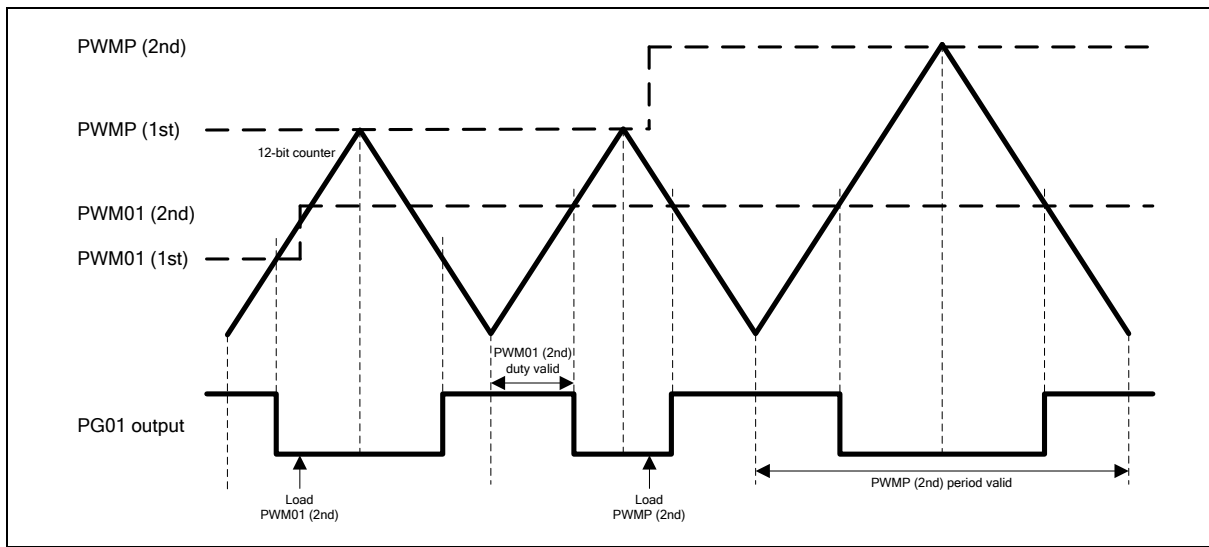


图 6.6-6 PWM 中心对齐波形

输出频率和占空比公式为：

$$\text{PWM 频率} = \frac{F_{\text{PWM}}}{2 \times \{PWMnPH, PWMnPL\}} \quad (F_{\text{PWM}} \text{ 是时钟源经 PWMDIV分频后的频率}).$$

$$\text{PWM 高电平占空比} = \frac{\{PWMnCHxH, PWMnCHxL\}}{\{PWMnPH, PWMnPL\}}.$$

6.6.4.5 操作模式

PnGx 经模式选择电路，输出不同模式的波形。有独立模式，互补模式和同步模式。

6.6.4.6 独立模式

当PWMMOD[1:0] (PWMnCON1[7:6]) = [0:0]时是独立模式，这是缺省模式，PnGx 各路独立输出波形。

6.6.4.7 带死区的互补模式

PWMMOD[1:0] = [0:1]时，是互补模式，PnG0/2/4 和独立模式一样，输出 PWM 波，PnG1/3/5 输出对应的互补波形，PnG1/3/5 占空比寄存器 {PWMnH, PWMnL} (n=1/3/5)被忽略。

电机控制应用，互补的波形要插入死区，防止一对半桥的上下功率管在开关切换时瞬间直通。PWM0的每一对PWM都有一个9位的下计数器 PWM0DTCNT，在输出从低到高跳变时插入死区时间，P0G0/P0G1 的死区插

入如下图，P0G2/P0G3 和 P0G4/P0G5 的死区时间插入与此相同，每一对PWM是否插入死区时间可由PWM0DTEN [3:0] 单独使能。

注意： PWM0DTCNT 和 PWM0DTEN 是 TA 写保护的，死区控制仅在 PWM0 配置为互补模式时有效。死区的插入也仅PWM0 这一组才有此功能。

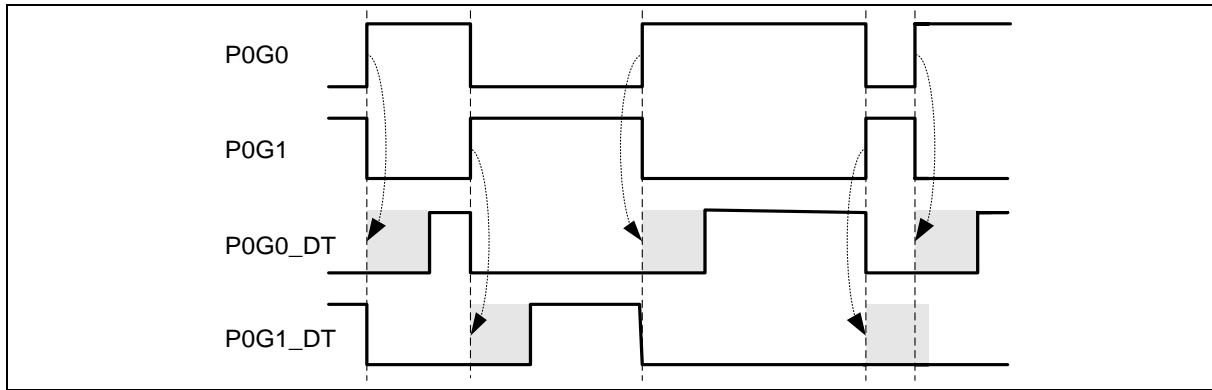


图 6.6-7 PWM0 互补模式和死区插入

6.6.4.8 同步模式

PWMMOD[1:0] = [1:0]使能同步模式。PnG0/2/4 和独立模式一样，PnG1/3/5 输出波形与 PnG0/2/4 一样。

6.6.4.9 输出消隐控制

每路PWM 都可以被 消隐掉而输出固定电平，控制BLDC时此功能非常有用。寄存器PWMnMEN决定哪一路 PWM 信号被消隐。寄存器PWMnMD 决定被消隐的通道输出的电平。PWMnMEN的缺省值是 00H，即不消隐，都输出PWM波。注意消隐电平可以由PWM0NP 配置为反相输出。

6.6.4.10 紧急停止

紧急停止常用于保护电机控制终统不被损坏。当FBINEN (PWMnCON1.3)置位并且引脚FB有效时，PWM 固定输出寄存器PWMnFBD 的值。PWM0RUN (PWM0CON0.7) 被硬件清0停止产生PWM信号。16位 PWM 计数器回到0000H值。急停标志FBF将被置位，如果使能了中断，将会发生中断。即使FBF标志被软件清0，PWM 引脚仍然输出固定值PWMnFBD。只有重新置位PWM0RUN后PWM才重新开始工作。引脚FB的信号还可用FBINLS (PWMnFBD.6) 选择有效电平。注意：引脚FB输入的信号必须长于8个系统时钟周期。因为有8/F_{sys} 防抖功能。另外ADC 的比较值也可触急停功能。机理与FB 引脚一样。

注意： 仅 PWM0 这一组有此功能。

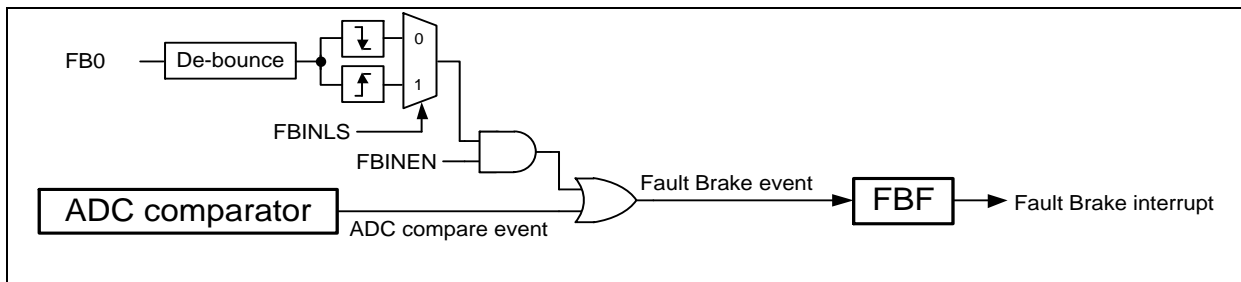


图 6.6-8 紧急停止框图

6.6.4.11 极性控制

PWM0 的每一路输出有独立的极性控制位 PNP0~PNP5。可控制波形反相后输出。

注意: 仅 PWM0 这一组有此功能。

6.6.5 PWM 中断

标志位PWMF (PWMnCON0.5) 指示特定点申请了中断, 这些点可由 INTSEL[2:0] 和 INTTYP[1:0] (PWMnINTC[2:0] 和 [5:4])选择。注意, 中心点和周期结束点中断, 只有中心对齐模式才有。PWMF 由软件清0。

PWMnINTC – PWM 中断控制

寄存器	SFR地址	复位值
PWM0INTC	B7H, 1页	0000_0000 b
PWM1INTC	B6H, 2页	0000_0000 b
PWM2INTC	C6H, 2页	0000_0000 b
PWM3INTC	D6H, 2页	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 00 = PWM 通道 0/1/2/3/4/5 引脚的下沿. 01 = PWM 通道0/1/2/3/4/5 引脚的上沿. 10 = 周期中点. 11 = 周期末. 注意中点和周期末, 只有互补模式有效。
2:0	INTSEL[2:0]	PWM 中断通道选择 000 = PWM_CH0. 001 = PWM_CH1. 010 = PWM_CH2. 011 = PWM_CH3. 100 = PWM_CH4. 101 = PWM_CH5. 其它 = PWM_CH0.

PWM 中断与 PWM 波形关系如下：

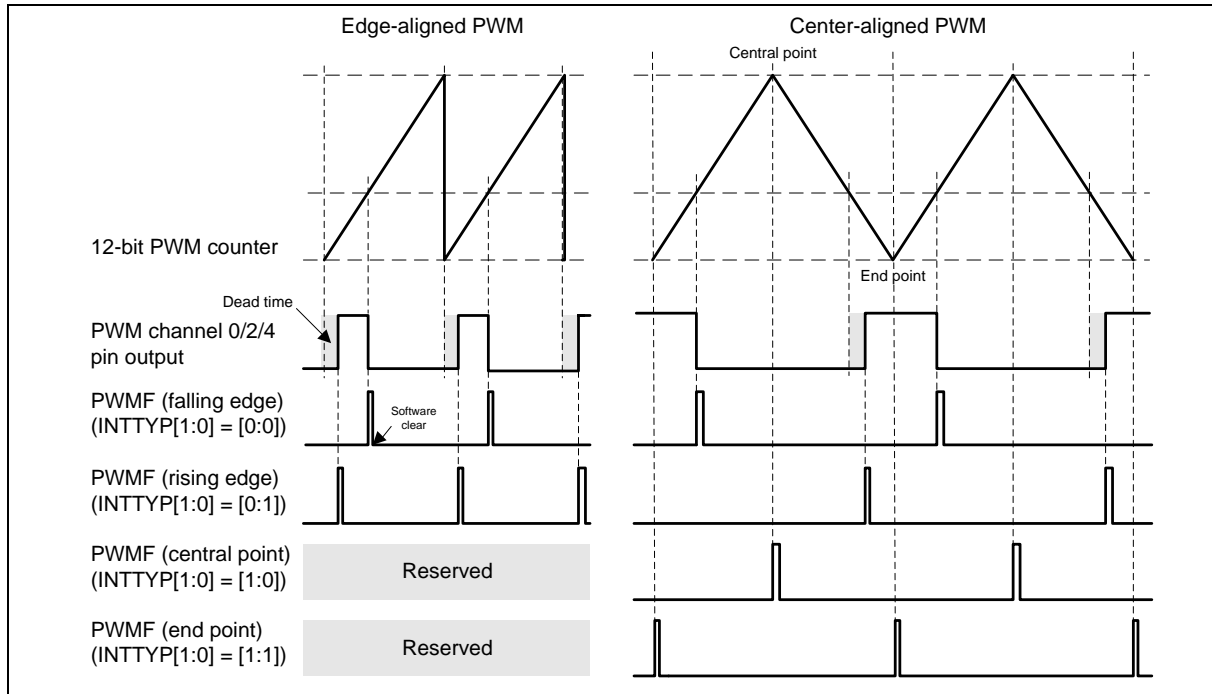


图 6.6-9 PWM 中断

FB 急停中断与 PWM 中断不是一个中断向量。当 FB 引脚或 ADC 比较产生急停中断时，FBF (PWM0FBD.7) 被置位，如果使能了中断就产生 FB 急停中断，急停中断的使能位是 EFB0 (EIE0.5)，中断标志 FBF 需软件清 0。

6.6.6 PWM 的控制寄存器

PWM0CON0 – PWM0 控制 0 (可位寻址)

寄存器	SFR地址	复位值
PWM0CON0	D8H, 0页	0000_0000b

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

位	名称	描述
7	PWM0RUN	<p>PWM0 使能位</p> <p>0 = PWM0 停止工作.</p> <p>1 = PWM0 开始工作.</p> <p>注意: 仅适于 PWM0CON0。</p>
6	LOAD	<p>PWM 周期和占空比加载</p> <p>此位用于把缓存中的周期和占空比新值加载到周期和占空比寄存器，下个周期开始使用新值。写入缓存的周期和占空比，只有在周末才会加载，然后硬件清零此位。</p> <p><u>写:</u></p> <p>0 = 无效。</p> <p>1 = 本次PWM周期结束后加载周期和占空比的新值。</p> <p><u>读:</u></p> <p>0 = 加载完成。</p> <p>1 = 加载未完成。</p>
5	PWMF	<p>PWM 标志</p> <p>按寄存器PWM0INTC中的 INTSEL[2:0] 和 INTTYP[1:0]的定义，此位置1，需软件清零。</p>
4	CLRPWM	<p>计数器清零</p> <p>此位置1将对 PWM 16位计数器清零，清零完成后，硬件清零此位。</p> <p><u>写:</u></p> <p>0 = 无效。</p> <p>1 = PWM 16位计数器清零</p> <p><u>读:</u></p> <p>0 = 清零完成。</p> <p>1 = 清零尚未完成。</p>
3	PWM3RUN	<p>PWM3 工作使能</p> <p>0 = PWM3 停止工作.</p> <p>1 = PWM3 开始工作.</p> <p>注意: 仅适于 PWM0CON0。</p>

位	名称	描述
2	PWM2RUN	<p>PWM2工作使能</p> <p>0 = PWM2停止工作.</p> <p>1 = PWM2开始工作.</p> <p>注意: 仅适于 PWM0CON0。</p>
1	PWM1RUN	<p>PWM1工作使能</p> <p>0 = PWM1停止工作.</p> <p>1 = PWM1开始工作.</p> <p>注意: 仅适于 PWM0CON0。</p>
0	P33FBINEN	<p>P33 紧急停止</p> <p>0 = 禁止引脚 P33 FB 对PWM0的急停功能</p> <p>1 = 使能引脚 P33 FB 对PWM0的急停功能，一旦此引脚有符合 FBINLS (FBD.6)定义的跳变发生，PWM0C0~C5输出寄存器 FBD 的值，PWMRUN (PWM0CON0.7) 将被硬件清0，PWM停止工作，PWMRUN 再次写1才开始工作。</p> <p>注意: 仅适于 PWM0CON0。</p>

PWMnCON0 – PWMn 控制 0, n = 1,2,3

寄存器	SFR地址	复位值
PWM1CON0	B4H, 2页	0000_0000 b
PWM2CON0	C4H, 2页	0000_0000 b
PWM3CON0	D4H, 2页	0000_0000 b

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

位	名称	描述
7	PWMRUN	PWM 工作使能 0 = PWM 停止工作. 1 = PWM 开始工作.
6	LOAD	PWM 加载新的周期和占空比值 此位用于把缓存中的周期和占空比新值加载到周期和占空比寄存器，下个周期开始使用新值。 写入缓存的周期和占空比，只有在周末才会加载，然后硬件清零此位。 <u>写:</u> 0 = 无效。 1 = 本次PWM周期结束后加载周期和占空比的新值。 <u>读:</u> 0 = 加载完成。 1 = 加载未完成。
5	PWMF	PWM 标志 按寄存器PWM0INTC中的 INTSEL[2:0] 和 INTTYP[1:0]的定义，此位置1，需软件清零。
4	CLRPWM	计数器清零 此位置1将对 PWM 16位计数器清零，清零完成后，硬件清零此位。 <u>写:</u> 0 = 无效。 1 = PWM 16位计数器清零 <u>读:</u> 0 = 清零完成。 1 = 清零尚未完成。

PWMnCON1 – PWM 控制 1

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

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

位	名称	描述
7:6	PWMMOD[1:0]	PWM 模式选择 00 = 独立模式. 01 = 互补模式. 10 = 同步模式 11 = 保留.
5	GP	组模式使能 组模式使能后前三对 PWM 占空比由 PWM01H PWM01L 决定. 0 = 禁止组模式 1 = 使能组模式. 注意: 仅适用于 PWM0
4	PWMTYP	PWM 类型选择 0 = 边沿对齐. 1 = 中心对齐.
3	FBINEN	FB 急停引脚输入使能 0 = 功能禁止 1 = 使能急停功能。一旦此引脚出现符合 FBINLS (PWM0FBD.6) 定义的边沿, PWM0 通道 0~5 输出 PWM0FBD 的值。PWM0RUN (PWM0CON0.7)被硬件清0。PWM0RUN置1后 PWM才再次开始工作。 注意: 仅适用于 PWM0

位	名称	描述
2:0	PWMDIV[2:0]	PWM 时钟预分频系数 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	SFR地址	复位值
CKCON	8EH, 0页	0000_0000b

7	6	5	4	3	2	1	0
-	PWMCKS	-	T1M	T0M	T0OE	CLOEN	-
-	读/写	-	读/写	读/写	读/写	读/写	-

位	名称	描述
6	PWMCKS	PWM 时钟源选择 0 = 选系统时钟 FSYS. 1 = 选 定时器 1 的溢出频率。

PWMnPL – PWM 周期值低字节, n = 0,1,2,3

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

7	6	5	4	3	2	1	0
PWM0P[7:0]							
读/写							

位	名称	描述
7:0	PWMnP[7:0]	PWMn 周期值低字节 与 PWMnPH 共同决定PWM的周期值。

PWMnPH – PWM 周期值高字节, n = 0,1,2,3

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

7	6	5	4	3	2	1	0
PWM0P[15:8]							
读/写							

位	名称	描述
7:0	PWM0P[15:8]	PWM 周期值高字节 与PWM0PL共同决定 PWM 的周期值。

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

位	名称	描述	复位值
PWM0C0H	D2H, 0页	PWM0 通道0占空比高字节	0000_0000 b
PWM0C1H	D3H, 0页	PWM0通道1占空比高字节	0000_0000 b
PWM0C2H	D4H, 0页	PWM0 通道2占空比高字节	0000_0000 b
PWM0C3H	D5H, 0页	PWM0 通道3占空比高字节	0000_0000 b
PWM0C4H	C4H, 1页	PWM0 通道4占空比高字节	0000_0000 b
PWM0C5H	C5H, 1页	PWM0 通道5占空比高字节	0000_0000 b
PWM1_CH0H	AAH, 2页	PWM1通道0占空比高字节	0000_0000 b
PWM1_CH1H	ABH, 2页	PWM1通道1占空比高字节	0000_0000 b
PWM2_CH0H	BAH, 2页	PWM2 通道0占空比高字节	0000_0000 b
PWM2_CH1H	BBH, 2页	PWM2 通道1占空比高字节	0000_0000 b
PWM3_CH0H	CAH, 2页	PWM3 通道0占空比高字节	0000_0000 b
PWM3_CH1H	CBH, 2页	PWM3 通道1占空比高字节	0000_0000 b

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

位	名称	描述
7:0	PWMnCx [15:8]	PWMnCx 占空比高字节 与 PWMnCxL共同决定占空比值。

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

位	名称	描述	复位值
PWM0C0L	DAH, 0页	PWM0 通道0占空比低字节	0000_0000 b
PWM0C1L	DBH, 0页	PWM0 通道1占空比低字节	0000_0000 b
PWM0C2L	DCH, 0页	PWM0 通道2占空比低字节	0000_0000 b
PWM0C3L	DDH, 0页	PWM0 通道3占空比低字节	0000_0000 b
PWM0C4L	CCH, 1页	PWM0 通道4占空比低字节	0000_0000 b
PWM0C5L	CDH, 1页	PWM0 通道5占空比低字节	0000_0000 b
PWM1_CH0L	B2H, 2页	PWM0 通道0占空比低字节	0000_0000 b
PWM1_CH1L	B3H, 2页	PWM0 通道1占空比低字节	0000_0000 b
PWM2_CH0L	C2H, 2页	PWM0 通道0占空比低字节	0000_0000 b
PWM2_CH1L	C3H, 2页	PWM0 通道1占空比低字节	0000_0000 b
PWM3_CH0L	D2H, 2页	PWM0 通道0占空比低字节	0000_0000 b
PWM3_CH1L	D3H, 2页	PWM0 通道1占空比低字节	0000_0000 b

7	6	5	4	3	2	1	0
PWMnCx [7:0]							
R/W							

位	名称	描述
7:0	PWMnCx [7:0]	PWMnCx 占空比低字节 与 PWMnCxH 共同决定PWM的占空比。

PWM0DTEN – PWM 死区使能

寄存器	SFR地址	复位值
PWM0DTEN	F9H, 0页, TA 保护	0000_0000 b

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

位	名称	描述
4	PDTCNT8	PWM0 死区计数器的第 8 位 见寄存器 PWM0DTCNT
2	PDT45EN	PWM0C4/5 死区使能 仅互补模式有效。 0 = 无死区 1 = 在 P0G4/P0G5 上沿插入死区时间
1	PDT23EN	PWM0_CH2/3死区使能 仅互补模式有效。 0 =无死区。 1 = 在 P0G2/P0G3上沿插入死区时间。
0	PDT01EN	PWM0C0/1死区使能 仅互补模式有效。 0 =无死区 1 = 在 P0G0/P0G1上沿插入死区时间

PWM0DTCNT – PWM 死区计数器

寄存器	SFR地址	复位值
PWM0DTCNT	FAH, 0页, TA 保护	0000_0000 b

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

位	名称	描述
7:0	PWM0DTCNT[7:0]	<p>PWM0 死区计数器的低字节</p> <p>此8位与 PWM0DTEN.4组成 PWM0 的9位死区计数器 PWM0DTCNT。仅 PWM0 配置为互补模式并且 PWM0DTEN 的相关位为1时有效。</p> $\text{PWM0 死区时间} = \frac{\text{PDTCNT} + 1}{F_{\text{SYS}}}$ <p>注意在PWM0计数期间不要改 PWM0DTCNT 的值。</p>

PWMxMEN – PWMnCx 消隐使能, n=0,1,2,3;x=0,1,2,3,4,5

寄存器	SFR地址	描述	复位值
PWM0MEN	FBH, 0页	PWM0Cx 消隐使能	0000_0000 b
PWM1MEN	ADH, 2页	PWM1Cx 消隐使能	0000_0000 b
PWM2MEN	BDH, 2页	PWM2Cx 消隐使能	0000_0000 b
PWM3MEN	CDH, 2页	PWM3Cx 消隐使能	0000_0000 b

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

位	名称	描述
x	PMENx	PWMnCx消隐使能 0 = PWMnCx 输出 PWM 波 1 = PWMnCx 输出 PMDx. 注意: PMEN2~5 仅适于PWM0.

PWMnMD –PWMnCx 消隐数据

寄存器	SFR地址	描述	复位值
PWM0MD	FCH, 0页	PWM0Cx 消隐数据	0000_0000 b
PWM1MD	ACH, 2页	PWM1Cx 消隐数据	0000_0000 b
PWM2MD	BCH, 2页	PWM2Cx 消隐数据	0000_0000 b
PWM3MD	CCH, 2页	PWM3Cx 消隐数据	0000_0000 b

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

位	名称	描述
x	PMDx	PWMnCx 消隐数据 一旦对应的PMENx置1，PWMnCx 将输出消隐数据。 0 = PWMnCx 输出 0。 1 = PWMnCx 输出 1。 注意: PMD2~5 仅适于 PWM0.

PWM0CON1 – PWM 控制 1

寄存器	SFR地址	复位值
PWM0CON1	DFH, 0页	0000_0000 b

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

位	名称	描述
3	FBINEN	FB 急停引脚输入使能 0 = 禁止FB引脚的急停功能。 1 = 使能FB引脚的急停功能。引脚FB一旦出现符合FBINLS (PWM0FBD.6)定义的信号边沿时, PWM0 通道 0~5输出寄存器PWM0FBD的值, PWM6/7保持。PWM0RUN (PWM0CON0.7) 被硬件清0, PWM0RUN 被软件写1后, PWM功能恢复

PWM0FBD – PWM 急停控制

寄存器	SFR地址	复位值
PWM0FBD	D7H, 0页	0000_0000 b

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

位	名称	描述
7	FBF	故障急停标志 FBINENT 置1时，FB引脚若有符合FBINLS (PWM0FBD.6) 定义的边沿发生时，此位置1。需软件清0。清0后，置位PWM0RUN (PWM0CON0.7)，PWM才会再次工作。
6	FBINLS	FB 引脚急停信号输入边沿选择 0 = 下沿. 1 = 上沿.
5:0	FBDx	PWMx 急停输出信号 0 = PWM0Cx 输出低. 1 = PWM0Cx 输出高

PNP – PWM 反相控制

寄存器	SFR地址	复位值
PNP	D6H, 0页	0000_0000 b

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

位	名称	描述
5:0	PNPn	PWMn 反相使能 0 = 直接输出. 1 = 反相输出

PIOCON0 – PWM引脚选择

寄存器	SFR地址	复位值
PIOCON0	DEH, 0页	0000_0000 b

7	6	5	4	3	2	1	0
-	-	PIO05	PIO04	PIO03	PIO02	PIO01	PIO00
-	-	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
5	PIO05	P0.3/PWM 引脚功能选择 0 = P0.3/PWM 引脚功能为 P0.3. 1 = P0.3/PWM 引脚功能为 PWM0C5/PWM3_CH1输出。 (PWM3_CH1P=11, 选 PWM3_CH1; 其它为 PWM0C5)
4	PIO04	P0.1/PWM引脚功能选择 0 = P0.1/PWM引脚功能为P0.1 1 = P0.1/PWM引脚功能为PWM0C4/PWM3_CH0输出。 (PWM3_CH0P=10, 选PWM3_CH0; 其它选PWM0C4)
3	PIO03	P0.0/PWM引脚功能选择 0 = P0.0/PWM引脚功能为P0.0。 1 = P0.0/PWM引脚功能为PWM0_CH3/PWM2_CH1输出。 (PWM2_CH1P=10, 选PWM2_CH1; 其它选 PWM0_CH3)
2	PIO02	P1.0/PWM引脚功能选择 0 = P1.0/PWM引脚功能为P1.0。 1 = P1.0/PWM引脚功能为PWM0_CH2/PWM2_CH0输出。 (PWM2_CH0P=10, 选PWM2_CH0; 其它选 PWM0_CH2)
1	PIO01	P1.1/PWM引脚功能选择 0 = P1.1/PWM引脚功能为P1.1。 1 = P1.1/PWM引脚功能为PWM0_CH1/PWM1_CH1输出。 (PWM1_CH1P=11, 选PWM1_CH1; 其它选 PWM0_CH1)
0	PIO00	P1.2/PWM引脚功能选择 0 = P1.2/PWM引脚功能为P1.2。 1 = P1.2/PWM引脚功能为PWM0C0/PWM1_CH0 输出。 (PWM1_CH0P=11, 选 PWM1_CH0; 其它选 PWM0_CH1)

PIOCON1 – PWM引脚选择

寄存器	SFR地址	复位值
PIOCON1	C6H, 1页	0000_0000 b

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

位	名称	描述
7	PIO17	P1.7/PWM引脚功能选择 0 = P1.7/PWM3_CH0 引脚功能为 GPIO P1.7. 1 = P1.7/PWM3_CH0 引脚功能为 PWM3_CH0输出
5	PIO15	P1.5/PWM引脚功能选择 0 = P1.5/PWM 引脚功能为 P1.5. 1 = P1.5/PWM 引脚功能为 PWM0_CH5 / PWM3_CH1输出 当 AUXR5[3:2].PWM3CH1P = 11, PIO15 = 1 , P1.5定义为PWM3_CH1. 当 AUXR5[3:2].PWM3CH1P = 00 ,PIO15 = 1, P1.5定义为PWM0_CH5.
3	PIO04	P0.4/PWM引脚功能选择 0 = P0.4/PWM 引脚功能为 P0.4. 1 = P0.4/PWM 引脚功能为 PWM0_CH3 / PWM2_CH1输出 当 AUXR4[7:6].PWM2CH1P = 11, PIO04 = 1 , P1.4 定义为 PWM2_CH1. 当 AUXR4[7:6].PWM2CH1P = 00 ,PIO04 = 1, P1.4定义为PWM0_CH5.
2	PIO05	P0.5/PWM 引脚功能选择 0 = P0.5/PWM 引脚功能为 P0.5. 1 = P0.5/PWM引脚功能为 PWM0_CH2 / PWM2_CH0输出 (PWM2CH0P=11, 选 PWM2_CH0; 其它选 PWM0_CH3)
1	PIO14	P1.4/PWM 引脚功能选择 0 = P1.4/PWM 引脚功能为 P1.4. 1 = P1.4/PWM 引脚功能为 PWM0_CH1 / PWM1_CH1 输出. (PWM1_CH1P=10, 选 PWM1_CH1; 其它选PWM0_CH1)

PIOCON2 – PWM 引脚选择

寄存器	SFR地址	复位值
PIOCON2	B7H, 2页	0000_0000 b

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

位	名称	描述
7	PIO34	P3.4/PWM3_CH1 引脚功能选择 0 = P3.4/PWM3_CH1 引脚功能为P3.4. 1 = P3.4/PWM3_CH1 引脚功能为PWM3_CH1 输出。
6	PIO33	P3.3/PWM0_CH1 引脚功能选择 0 = P3.3/PWM0C0 引脚功能为P3.3. 1 = P3.3/PWM0C0 引脚功能为PWM0_CH1 输出。
5	PIO32	P3.2/PWM3_CH0 引脚功能选择 0 = P3.2/PWM3_CH0 引脚功能为P3.2. 1 = P3.2/PWM3_CH0 引脚功能为PWM3_CH0 输出。
4	PIO31	P3.1/PWM2_CH1 引脚功能选择 0 = P3.1/PWM2_CH1 引脚功能为P3.1. 1 = P3.1/PWM2_CH1 引脚功能为PWM2_CH1 输出。
3	PIO30	P3.0/PWM2_CH1 引脚功能选择 0 = P3.0/PWM2_CH1 引脚功能为P3.0 1 = P3.0/PWM2_CH1 引脚功能为PWM2_CH1 输出。
2	PIO23	P23/PWM1_CH0 引脚功能选择 0 = P2.3/PWM1_CH0 引脚功能为P2.3. 1 = P2.3/PWM1_CH0 引脚功能为PWM1_CH0 输出。
1	PIO22	P2.2/PWM1_CH1 引脚功能选择 0 = P2.2/PWM1_CH1 引脚功能为 P2.2. 1 = P2.2/PWM1_CH1 引脚功能为 PWM1_CH1 输出。

AUXR4 –辅助寄存器4

寄存器	SFR地址	复位值
AUXR4	A3H, 2页	0000_0000 b

7	6	5	4	3	2	1	0
PWM2_CH1P		PWM2_CH0P		PWM1_CH1P		PWM1_CH0P	
读/写		读/写		读/写		读/写	

位	名称	描述
7:6	PWM2_CH1P	PWM2通道1引脚选择 00 = PWM2_CH1 由 P3.0 引脚输出 01 = 选 P3.1 10 = 选 P0.0 11 = 选 P0.4
5:4	PWM2_CH0P	PWM2 通道0引脚选择 00 = 保留, 缺省值 01 = 选 P2.1 10 = 选 P1.0 11 = 选 P0.5
3:2	PWM1_CH1P	PWM1通道1引脚选择 00 =保留, 缺省值 01 = PWM1_CH1 引脚选 P2.2 10 = 选 P1.4 11 = 选 P1.1
1:0	PWM1_CH0P	PWM1通道0引脚选择 00 =保留, 缺省值 01 = PWM1_CH0 引脚选 P2.3 10 = 选 P1.2 11 =保留

AUXR5 – 辅助寄存器 5

寄存器	SFR地址	复位值
AUXR5	A4H, 2页	0000_0000 b

7	6	5	4	3	2	1	0
CLOP	T0P	-		PWM3_CH1P		PWM3_CH0P	
读/写	读/写	-		读/写		读/写	

位	名称	描述
3:2	PWM3_CH1P	PWM3 通道1引脚选择 00 = 保留, 缺省值 01 = PWM3_CH1 由 P3.4 引脚输出 10 = PWM3_CH1 由 P1.5引脚输出 11 = PWM3_CH1 由 P0.3 引脚输出
1:0	PWM3_CH0P	PWM3通道0引脚选择 00 =保留, 缺省值 01 = P3.2 10 = P0.1 11 = 保留

6.7 看门狗(WDT)

6.7.1 概述

WDT可配置为溢出后复位芯片，所以对于干扰较强的应用，代码一旦跑飞，WDT可以复位芯片，恢复系统到正常运行状态。WDT 也可配置为在空闲或掉电模式下运行的通用定时器，WDTEN[3:0] (CONFIG4[7:4]) 可配置 WDT 为通用定时器，还是复位芯片的功能。

WDT的定时时间为 $\frac{1}{F_{LIRC} \times \text{clockdividerscalar}} \times 64$ ，此 F_{LIRC} 是内部 10 kHz时钟。下表是不同定时时间的配置

WDPS.2	WDPS.1	WDPS.0	分频系数	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: LIRC 误差较大，此为时间期望值。

表 6.7-1 WDT 定时时间表

由于 WDT 定时的最大值有限，从空闲或掉电中唤醒 MS51 可使用唤醒定时器 WKT。

```

WDT的配置见 章节 0      ORG    0000H
LJMP  START

ORG    0053H
LJMP  WDT_ISR

ORG    0100H
;*****
;WDT interrupt service routine
;*****
WDT_ISR:
CLR    EA
MOV    TA, #0AAH
MOV    TA, #55H
ANL    WDCON, #11011111B    ;clear WDT interrupt flag
SETB   EA
RETI
    
```

```

;*****
;Start here
;*****
START:
MOV    TA,#0AAH
MOV    TA,#55H
ORL    WDCON,#00010111B    ;choose interval length and enable WDT
                                ;running during Power-down

SETB   EWDT                ;enable WDT interrupt
SETB   EA

MOV    TA,#0AAH
MOV    TA,#55H
ORL    WDCON,#10000000B    ; WDT run
;*****
;Enter Power-down mode
;*****
LOOP:
ORL    PCON,#02H
LJMP   LOOP
    
```

WDT控制寄存器

6.7.2 WDT 溢出复位功能

WDTEN[3:0] (CONFIG4[7:4]) 非 FH 时, WDT 配置为溢出复位功能。若WDTEN[3:0]的值不是 5H, 则空闲或掉电状态下, 保持计数。注意配置为复位功能时, 写 WDTR 和 WIDPD 无效。

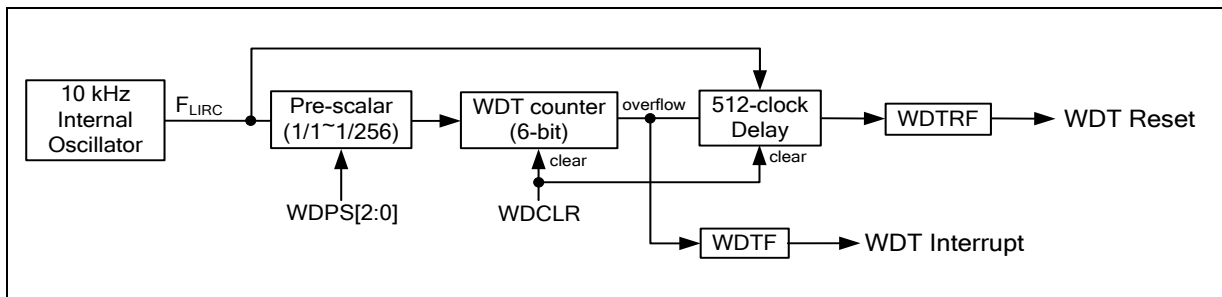


图 6.7-1 WDT 配置为复位功能

芯片上电开始执行代码后, WDT连续计数。计数时间由 WDPSS[2:0] (WDCON[2:0])配置。计数期满会置位中断标志 WDTF (WDCON.5), 若中断使能位 EWDT (EIE0.4) 和全局中断使能位 EA 都置1了, 就去执行 WDT 中断代码, 同时512个10KHz时钟后复位芯片, 所以若不想不复位, 就必须在512个时钟内, 通过对 WDCLR 写1清零 WDT计数器。WDCLR 位会自动清0。WDT复位芯片后, 标志位 WDRF (WDCON.3) 会置1, WDT复位标志 WDRF 除上电复位会被清除外, 其它复位不会修改此位的值, 当然此位可被软件清0。注意, 代码中必须安排时间写 WDCON 。

WDT主要用于系统监测, 在干扰比较强的场和, 强干扰可能导致代码跑飞, 这时WDT就可以让芯片复位, 避免系统崩溃, 造成更大故障。如果不想WDT 复位芯片, 代码中要有“喂狗”代码: 写WDCLR 让WDT清 0 重新计数。当代码跑飞时, 执行不到喂狗代码, WDT 计数期满, 就会复位芯片。

6.7.3 WDT 通用定时器功能

当 WDTEN[3:0] (CONFIG4[7:4]) 为 FH, WDT 就是一个通用定时器, 软件可读写 WDTR 和 WIDPD 。

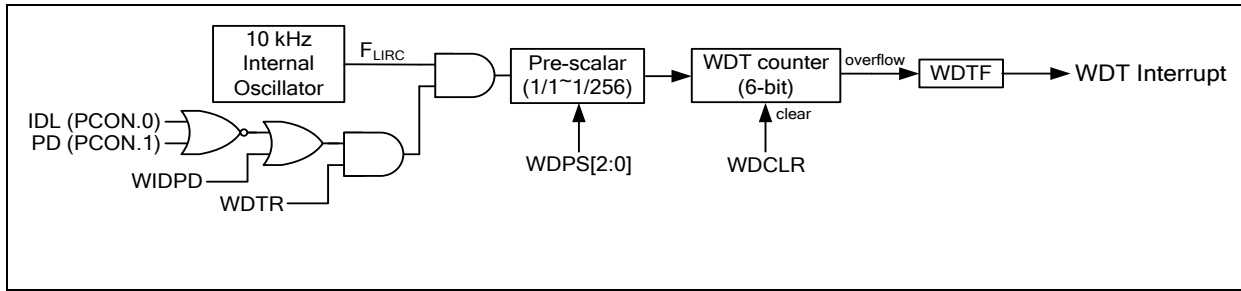


图 6.7-2 WDT 做定时器的框图

WDTR的值写0/1，决定WDT 停止/开始计数。计数期满标志 WDTF 硬件置1，用于软件查寻。若 EWDT (EIE0.4) 和全局中断 EA 被使能，将去执行中断代码。WDTF 由软件清0。

WDT可在CPU空闲或掉电状态下，以很低的功耗对10KHz保持计数，计数期满可将CPU唤醒。这对降低功耗非常有效。示例代码如下：

```

ORG    0000H
LJMP   START

ORG    0053H
LJMP   WDT_ISR

ORG    0100H
;*****
;WDT interrupt service routine
;*****
WDT_ISR:
    CLR    EA
    MOV    TA,#0AAH
    MOV    TA,#55H
    ANL    WDCON,#11011111B    ;clear WDT interrupt flag
    SETB   EA
    RETI

;*****
;Start here
;*****
START:
    MOV    TA,#0AAH
    MOV    TA,#55H
    ORL    WDCON,#00010111B    ;choose interval length and enable WDT
                                ;running during Power-down
    SETB   EWDT                ;enable WDT interrupt
    SETB   EA

    MOV    TA,#0AAH
    MOV    TA,#55H
    ORL    WDCON,#10000000B    ; WDT run
;*****
;Enter Power-down mode
;*****
LOOP:

```


ORL	PCON, #02H
LJMP	LOOP

6.7.4 WDT控制寄存器

CONFIG4

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

缺省值: 1111 1111b

位	名称	描述
7:4	WDTEN[3:0]	WDT使能 配置 WDT 功能. 1111 = WDT 关闭. WDT 配置为通用定时器. 0101 = WDT 配置为空闲或掉电时不工作的看门狗. 其它 = WDT配置为空闲或掉电时保持工作的看门狗.

WDT对 10 kHz分频，定时时间可选，溢出后可把芯处从空闲或掉电状态唤醒到正常工作，若使能了中断，还可产生中断。如果配置为溢出时复位芯片，芯片就会复位，除非软件不停的喂狗。

WDCON – WDT 控制

寄存器	SFR地址	复位值
WDCON	AAH, 0页, TA 保护	POR 0000_0111 b WDT 0000_1UUU b 其它 0000_UUUU b

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

位	名称	描述
7	WDTR	WDT 运行 此位仅当 WDTEN[3:0] (CONFIG4[7:4]) 全 1、WDT 配置为通用定时器时有效。 0 = WDT 禁止 1 = WDT 使能. WDT开始计数.
6	WDCLR	WDT 清0 此位写1用于把 WDT 清0, 读写效果不同: 写: 0 = 无效. 1 = WDT计数器清0. 读: 0 = WDT清0完成. 1 = WDT 清0未完成.
5	WDTF	WDT溢出标志 指示 WDT溢出.需软件清0
4	WIDPD	WDT空闲或掉电状态运行使能位 仅WDTEN[3:0] (CONFIG4[7:4])全 1配置为通用定时器晨, 此位才有效。 0 = WDT在空闲或掉电状态停止运行 1 = WDT 在空闲或掉电状态保持运行.
3	WDTRF	WDT复位标志 此位指示 CPU 被 WDT 复位, 建议每次复位后, 软件对此位清0
2:0	WDPS[2:0]	WDT 分频选择 选择分频系数从 1/1 到 1/256。详见 表 6.7-1. 缺省最大分频
<p>注意:</p> <p>1. WDTRF 上电后清0, WDT复位后置1, 其它复位不变。</p> <p>2. WDPS[2:0] 上电后全1, 其它复位后不变</p>		

6.8 唤醒定时器(WKT)

6.8.1 概述

WKT以片内10KHz为计数时钟，在空闲或掉电状态下仍能保持计数，可周期性地唤醒CPU。注意系统时钟必须大于WKT时钟的两倍。使能WKT不会自动使能时钟，别忘了配置并使能WKT的时钟。若想WKT在空闲或掉电状态保持计数，也要让其时钟源在空闲或掉电时保持工作。

WKT是一个自动重载、16位、上计数定时器，有一个分频系数在1/1~1/2048之间的预分频器。分频系数由WKPS[2:0] (WKCON[2:0])配置。位WKTR (WKCON.3)可启动或停止计数，计数到0xFFFF后，硬件从RWK中装载初值到计数器，并置位EWKT (EIE1.2)，若使能了中断，会产生中断。

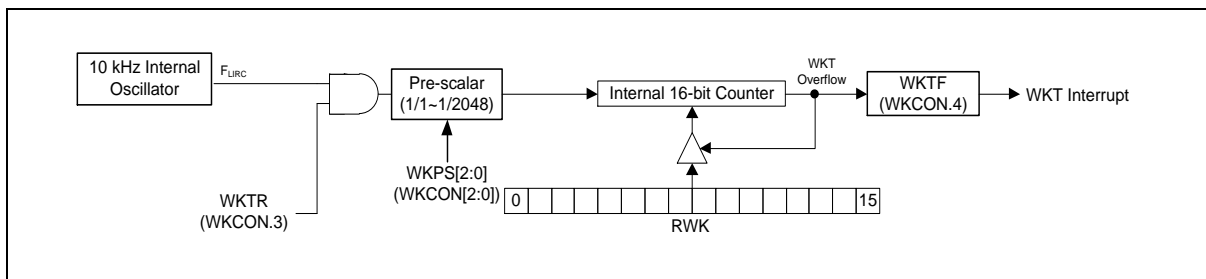


图 6.8-1 唤醒定时器框图

6.8.2 WKT控制寄存器

WKCON – 控制寄存器

寄存器	SFR地址	复位值
WKCON	8FH, 0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 WKT 溢出时此位置1，若使能了中断，会产生中断，需软件清0
3	WKTR	WKT运行控制 0 = WKT停止. 1 = WKT开始计数 注意：只有在此位为0时，才可以加载 WKT 的值，在此位为1时加载WKT结果将不可预知

位	名称	描述
2:0	WKPS[2:0]	WKT 分频 000 = 1/1. 001 = 1/4. 010 = 1/16. 011 = 1/64. 100 = 1/256. 101 = 1/512. 110 = 1/1024. 111 = 1/2048.

RWKL – 加载值低字节

寄存器	SFR地址	复位值
RWKL	86H, 0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低字节 加载WKT时16位数值的低字节。 注意：若不分频，加载值不能是 FFFFH

RWKH – 加载值高字节

寄存器	SFR地址	复位值
RWKH	97H, 2页	0000 0000b

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

位	名称	描述
7:0	RWK[15:8]	WKT 高字节 加载WKT时16位数值的高字节。 注意：若不分频，加载值不能是 FFFFH

6.9 串口 (UART0 和 UART1)

6.9.1 概述

MS51包含两个具备增强了自动地址识别和帧错误检测功能的全双工串口。由于两个串口的控制位是一样的。一般来说，在以下内容中，没有对串口1介绍，只对串口0介绍。

每个串口都有一种同步工作模式，模式0和三种全双工异步模式(通用异步接收器和发送器)，模式1, 2, 3。这意味着可以连续收发。串口带有接收缓存，意味着接收的前一个数据未被读取时，串口就能接收第二个数据。接收和发送都是对SBUF进行操作访问。写入SBUF数据将直接传到发送寄存器，而读取SBUF是访问一个具有独立物理地址的接收寄存器。串口共有4种操作模式，任何一种模式，任何操作SBUF的指令都将开始一次传输。

6.9.2 操作模式

6.9.2.1 模式0

模式0是与外部设备进行同步通信的方式。在该模式下，串行数据由RXD脚进行收发，而TXD脚用于产生移位时钟。每帧接收或发送8位数据。模式0提供半双工的通讯方式，它通过RXD收发数据。波特率设置为 $F_{SYS}/12$ (SM2 (SCON.5)为0) 或 $F_{SYS}/2$ (SM2为1)。注意无论发送或接收数据，串行时钟将一直由MCU产生。因此模式0下，连接在串口上的任何外设都以MCU为主机。图 6.9-1 串口模式0时序图显示串口模式0传输时序图。

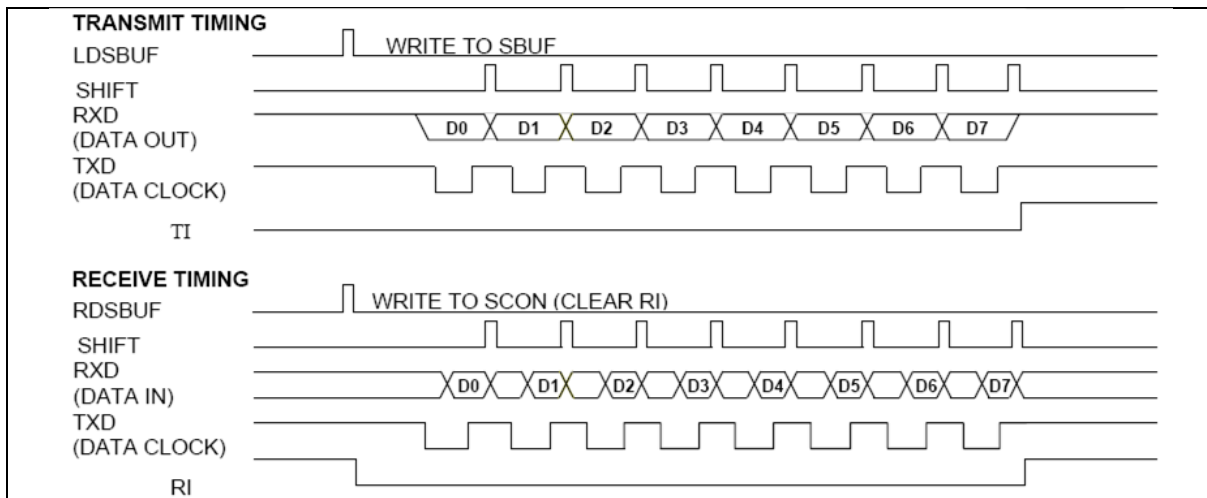


图 6.9-1 串口模式0时序图

如图所示，双向RXD引脚进行数据收发，TXD引脚用来输出移位时钟。串行通讯时，用移位时钟一位一位接收/发送数据。数据移入移出由最低位开始，波特率等于TXD的移位时钟频率。

向SBUF写入数据将会开始发送。控制模块将产生移位时钟和启动数据传输，直至8位数据传输完成。然后传输标志位TI (SCON.1) 置1，表示1个字节数据传输完成。

当REN(SCON.4)=1且RI(SCON.0)=0时串口开始接收数据。该条件告诉串口控制器有数据要移位进入。这个过程将持续到8位数据接收完毕，然后接收标志RI置1。用户可以清零RI，以触发接收下一字节数据。

6.9.2.2 模式1

模式1为异步全双工的工作方式。异步通讯模式通常用于和PC，调制解调器和其它类似接口间通讯。模式1下，10位数据通过TXD发送，通过RXD接收。10位数据组成如下：起始位（逻辑0），8位数据（最

低位在前)，停止位（逻辑1）。波特率由定时器1决定，SMOD(PCON.7)置1使波特率加倍。图 6.9-2 串口模式 1 时序图为串口模式1发送和接收的时序图。

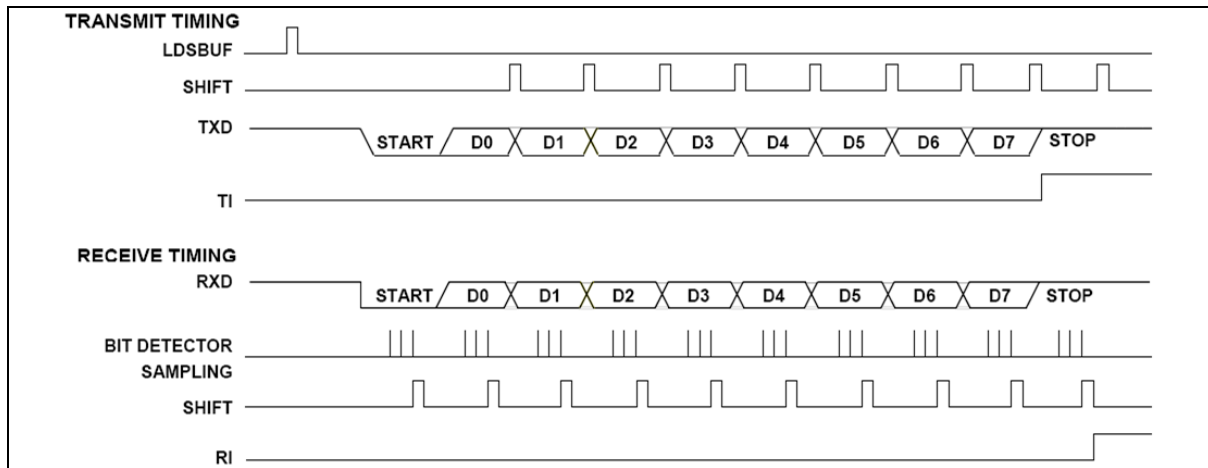


图 6.9-2 串口模式 1 时序图

向SBUF写入数据开始传输，数据通过TXD引脚发送。首先是开始位，随后是8位数据位，最后是停止位。停止位出现后，TI（SCON.1）将置1，表示一个字节传输完成。波特率发生器决定位传输速度。

当波特率发生器激活且REN(SCON.4) =1时，系统可以随时开始接收操作，当RXD脚上检测到1到0的跳变时，数据将开始被采样并根据波特率的时钟频率接收，停止位必须符合一定的条件，接收到数据才能装载到SBUF：

1. RI (SCON.0) = 0,
2. SM2 (SCON.5) = 0, 或当SM2 = 1且接收数据匹配“Given”或“广播地址”地址时停止位STOP=1。(详见 章节6.9.5多机通信”和 章节6.9.6自动地址识别.)

如果上述条件满足，接收到的数据将加载到SBUF里，RB8（SCON.2）为停止位，RI将被置。如果条件不满足，RI保持为0，没有数据加载。完成接收过程后，串口控制器将等待RXD脚上出现另一个1-0传输以开始新的数据接收。

6.9.2.3 模式 2

模式2为全双工异步通信，与模式1不同的是，模式2是11位收发。数据由起始位（逻辑0），8位数据（最低位在前），第9位数据（TB8或RB8）和停止位(逻辑1)组成。第9位做奇偶校验位或多机通信时用来区分数据和地址。波特率是系统时钟频率的1/32 或1/64，由 SMOD位(PCON.7)来配置。图 6.9-3 串口模式2 和 3 时序图指示串口模式2的传输时序。

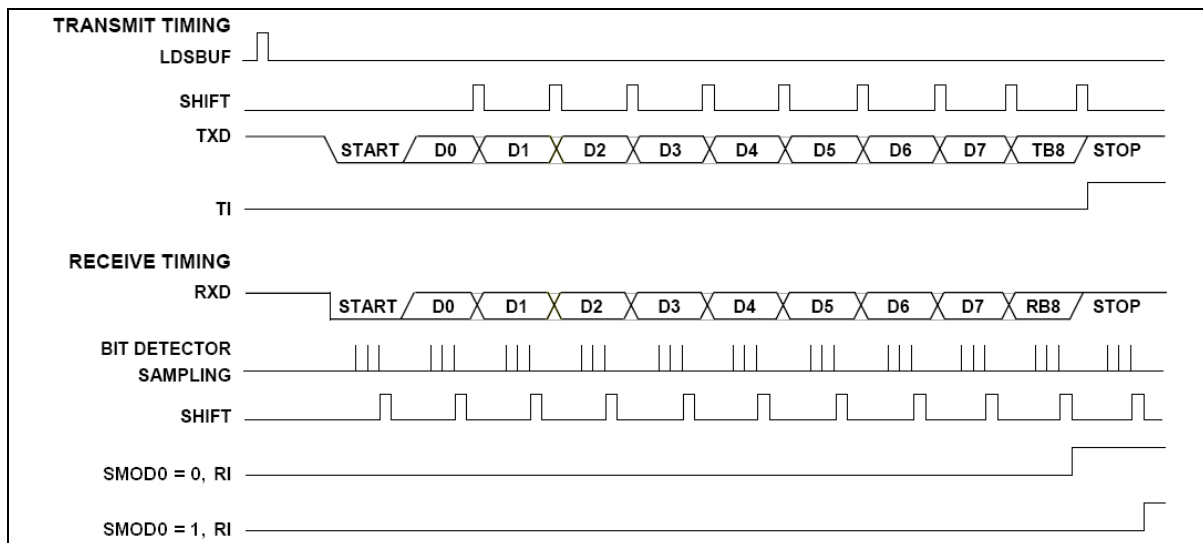


图 6.9-3 串口模式 2 和 3 时序图

向SBUF写入数据开始传输，数据发送在TXD引脚上。首先是开始位，然后是8位数据和TB8（SCON.3），最后是停止位，停止位发送后，TI将置位标志传输完成。

当REN=1时，串口可进行接收操作。RXD上的下降沿表示接收过程开始，数据根据所配置波特率进行采样和接收。停止位必须符合一定的条件，接收到数据才能装载到SBUF：

1. RI (SCON.0) = 0,
2. SM2(SCON.5) = 0, 或9th位 = 1同时 SM2 = 1且接收数据匹配“Given”或“广播地址”地址时。(详见(详见 章节6.9.5多机通信”和 章节6.9.6自动地址识别.)

如果上述条件满足，则第9位数据进入RB8(SCON.2)，8位数据进入SBUF，且RI置位。否则数据将不会装载，且RI保持为0。完成接收过程后，串口控制器等待RXD脚上的另一个1-0跳变以开始新的数据接收

6.9.2.4 模式 3

除波特率外模式 3与模式 2相同。模式3采用定时器1的溢出率作为波特率时钟。图 6.9-3 串口模式2 和 3 时序图模式3的传输时序，与模式2没有不同。

6.9.3 波特率

串口不同模式的波特率时钟源和速度是完全不同的。表 6.9-1 串口UART0 模式/波特率描述列出了所有选项。用户可以根据系统配置来计算。

在模式1或模式3，串口0的波特率时钟源可通过BRCK (T3CON.5)选择定时器1或定时器3。对于串口1，定时器3是唯一的时钟源。

当采用定时器1作为波特率发生器时，需要关闭定时器1中断。定时器1可配置为计数器或定时器，在三种工作模式下都可以。典型应用中，会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，同样也需要关闭定时器3中断。

下面为所有UART模式和波特率公式：

模式	帧位数	SM0 / SM1 (SCON[7:6])	SM2 (SCON[5])	SMOD (PCON[7])	波特率	
0	8	00	0	-	FSYS 除以 12	
			1		FSYS 除以 2	
1	10	01	-	0	Time1 T1M CKCON[4] = 0	$\frac{1}{32} \times \frac{F_{sys}}{12 \times (256 - TH1)}$
					Time1 T1M CKCON[4] = 1	$\frac{1}{32} \times \frac{F_{sys}}{(256 - TH1)}$
					Timer 3	$\frac{1}{32} \times \frac{F_{sys}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$
				1	Time1 T1M CKCON[4] = 0	$\frac{1}{16} \times \frac{F_{sys}}{12 \times (256 - TH1)}$
					Time1 T1M CKCON[4] = 1	$\frac{1}{16} \times \frac{F_{sys}}{(256 - TH1)}$
					Timer 3	$\frac{1}{16} \times \frac{F_{sys}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$
2	11	10	-	0	FSYS 除以 64	
				1	FSYS 除以 32	
3	11	11	-	0	Time 1 ^[1] T1M CKCON[4] = 0	$\frac{1}{32} \times \frac{F_{sys}}{12 \times (256 - TH1)}$
					Time 1 ^[1] T1M CKCON[4] = 1	$\frac{1}{32} \times \frac{F_{sys}}{(256 - TH1)}$
					Timer 3	$\frac{1}{32} \times \frac{F_{sys}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$

模式	帧位数	SM0 / SM1 (SCON[7:6])	SM2 (SCON[5])	SMOD (PCON[7])	波特率	
				1	Time1 ^[1] TM1 CKCON[3] = 0	$\frac{1}{16} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$
					Time1 ^[1] TM1 CKCON[3] = 1	$\frac{1}{16} \times \frac{F_{SYS}}{(256 - TH1)}$
					Timer 3	$\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$
注意: 定时器1应配置为定时器在自动重载模式 (模式2)						

表 6.9-1 串口 UART0 模式/波特率描述

模式	帧位数	SM0_1 / SM1_1 (S1CON[7:6])	SMOD_1 (T3CON[7])	波特率	
0	8	00	-	FSYS 除以 12	
1	10	01	0	Timer 3	$\frac{1}{32} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$
			1	Timer 3	$\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$
2	11	10	0	FSYS 除以 64	
			1	FSYS 除以 32	
3	11	11	0	Timer 3	$\frac{1}{32} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$
			1	Timer 3	$\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$

表 6.9-2 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 Mode1,REN=1,TI=1
TMOD |= 0x20;    //Timer1 set to Mode2 auto reload mode (must)
PCON |= 0x80;    //UART0 Double Rate Enable
CKCON |= 0x10;   //Timer 1 as clock source
T3CON &= 0xDF;   //Timer1 as UART0 clock source
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 Mode1,REN=1,TI=1
PCON |= 0x80;    //UART0 Double Rate Enable
T3CON &= 0xF8;   //(Prescale=1)
T3CON |= 0x20;   //UART0 baud rate clock source = Timer3
RH3 = value high byte
RL3 = value low byte
T3CON|= 0x08;    //Trigger Timer3
    
```

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 Mode1,REN_1=1,TI_1=1
T3CON = 0xF8;    //T3PS2=0,T3PS1=0,T3PS0=0 (Prescale=1),
RH3 = value high byte
RL3 = value low byte
T3CON|= 0x08;
    
```

下面列出一些通用波特率值根据不同Fsys和精度值:

Fsys 值	波特率	TH1 值 (Hex)	RH3,RL3 值(Hex)	波特率精度
24 MHz	4800	64 (SMOD=0)	FEC8	0.160256%
	9600	64	FF64	0.160256%
	19200	B2	FFB2	0.160256%
	38400	D9	FFD9	0.160256%
	57600	E6	FFE6	0.160256%
	115200	F3	FFF3	0.160256%
	150000	F6	FFF6	0.000000%
	166666	F7	FFF7	0.000400%

Fsys 值	波特率	TH1 值 (Hex)	RH3,RL3 值(Hex)	波特率精度
	187500	F8	FFF8	0.000000%
	214285	F9	FFF9	0.000333%
	250000	FA	FFFA	0.000000%
	300000	FB	FFFB	0.000000%
	375000	FC	FFFC	0.000000%
	500000	FD	FFFD	0.000000%
	750000	FE	FFFE	0.000000%
	1500000	FF	FFFF	0.000000%
16 MHz	4800	30	FF30	0.160256%
	9600	98	FF98	0.160256%
	19200	CC	FFCC	0.160256%
	38400	E6	FFE6	0.160256%
	57600	EF	FFE6	2.124183%
	115200	F7	FFF7	-3.549383%
	200000	FB	FFFB	0.000000%
	250000	FC	FFFC	0.000000%
	333333	FD	FFFD	0.000100%
	500000	FE	FFFE	0.000000%
	1000000	FF	FFFF	0.000000%

6.9.4 帧错误检测

帧错误检测用于异步模式 (模式 1、2 和 3)。由于总线噪音或争夺，导致没有检测到有效的停止位时，将发生帧错误。串口可以检测帧错误，并通过软件提示出错。

FE为帧错误标志，位于SCON第7位，这个位正常被用作为SM0。当SMOD0 (PCON.6)置1时，帧错误检测功能打开，它作为FE标志。SM0和FE其实是相互独立的标志位。

当帧错误发生时，FE标志由硬件置位。如果必要，FE可在串口中断程序中检测。注意在对FE标志位进行读写时，同时SMOD0必须为1。如果FE被置位，那么接下来即使接收到的正确数据帧也不会将其清除。对该位的清除必须由软件来完成。

6.9.5 多机通信

MS51串口支持多机通讯，可让一个主机向多从机系统中一个从机发送多帧序列信息。在同一串行线上使用该功能过程中不需要中断其它从机设备工作。该功能只能在模式2或模式3下进行。用户设置SM2(SCON.5)为1打开这个功能，以便当一个数据帧接收后，当第9位为1时，串口中断将产生（模式2下，第9位为停止位）。当SM2为1时，如果第9位为0，不会发生中断。在该情况下，第9位能简单的把从机地址和数据分开。

当主机需要向多个从机中的一个发送数据时，首先需要发送目标从机的地址。注意地址字节与数据字节是不同的：在地址字节中第9位为1，而数据字节中第9位为0。地址字节会触发所有从机，而每台从机检查接收到的地址是否与自身匹配。地址匹配的从机，清除SM2，准备接收数据；未被寻址到的从机的

SM2 必须保持，从而系统会持续工作，同时忽略接收数据。

配置多机通信步骤如下：

1. 设置所有设备(主机与从机)为串口模式2或3；
2. 所有从机的 SM2 位置为1；
3. 主机传输协议：
 - 第一个字节：地址，目标从机地址(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，因为第9位数据为1所有从机将中断。目标从机比较自身地址并且清除SM2 位等待接收后面的数据。其它从机则继续正常运行。
5. 接收到所有数据后，置SM2为1，等待下一地址。
6. SM2 在模式0下无效。若SM2 置 1，模式1可用于检测有效的停止位。模式1接收，如果SM2 是 1，将不会产生中断除非有效停止位已经接收。

6.9.6 自动地址识别

自动地址识别功能提高了多机通讯功能，允许UART通过硬件比较，在接收的比特流中，来识别特别的地址信息。该功能可以节省软件识别地址而所占用的程序空间，仅当串口识别到自身地址时，接收器置位RI位并请求中断。当多机通信功能使能时（SM2置位），自动地址识别功能也被使能。

如果需要，用户可以在模式1下使能自动地址识别功能。在这种配置下，停止位取代第九位的数据位。仅当接收命令的帧地址与器件地址匹配和有效的停止位时，RI置位。

使用自动地址识别功能，允许一个主机通过从机地址或“Given”地址选择性与一个或几个从机通信。所有从机可以通过“广播”地址联系。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的哪些位被用，哪些位不必关心。SADEN掩码可以与SADDR以“逻辑与”的方式以创建每个从机的“Given”地址。使用“Given”地址允许多从机被识别同时排除其它从机。

下列范例用以说明该功能的灵活应用：

例子 1，从机 0：

```
SADDR = 11000000b
SADEN = 11111101b
Given = 110000X0b
```

例子 2，从机 1：

```
SADDR = 11000000b
SADEN = 11111110b
Given = 110000Xb
```

在上面的例子中SADDR是相同的，SADEN的数据用于区分两个从机。从机0第0位为“0”而忽略第1位，从机1第1位为“0”而忽略第0位。当从机1第1位为0时，11000010B是从机0唯一的地址。当第0位为1，

排除从机0时，11000001b是从机1唯一的地址。当第0位=0（从机0）和第1位=0（从机1）时，两个从机可以通过同一个地址同时被选中。因此，使用广播地址(Boadcast address) 1100000b就可以同时定址。

更复杂应用可用于排除从机0之后，选择从机1或2:

例子 1, 从机 0:

```
SADDR = 11000000b
SADEN = 11111001b
Given = 1100XX0b
```

例子 2, 从机 1:

```
SADDR = 11100000b
SADEN = 11111010b
Given = 11100X0b
```

例子 3, 从机 2:

```
SADDR = 11000000b
SADEN = 11111100b
Given = 11000XXb
```

在上面的例子中，3个从机的区别在于地址的低3位。从机0需要第0位=0，它的唯一地址是11100110b。从机1需要第1位=0，它的唯一地址是11100101b。从机2需要第2位=0，它的唯一地址是11100011b。要只选择从机0和1，可使用地址11100100b，因为从机2需要第2位=1来排除从机2。

每个从机的“广播”地址的计算是通过逻辑或SADDR和SADEN。结果中的零位被视为“无关”位。例如:

```
SADDR = 01010110b
SADEN = 11111100b
Broadcast = 111111Xb
```

使用“无关”位可在广播模式下，提供更灵活的应用。不过在大部分应用条件下，广播地址全部使用FFH。

复位后，SADDR和SADEN初始化为00H。对于所有“无关”地址产生一个“Given”地址，以及一个“广播”地址对应所有XXXXXXXXb地址（所有“无关”位）。这样有效地禁止了自动寻址模式，允许微控制器保持标准串口模式而不使用这个功能。

6.9.7 串口控制寄存器

SCON – 串口控制

寄存器	SFR 地址	复位值
SCON	98H, 所有页, 位操作	0000_0000 b

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

位	名称	描述
7	SM0/FE	串口模式选择
6	SM1	SMOD0 (PCON.6) = 0: 参见 表 6.9-1 串口UART0 模式/波特率描述 SMOD0 (PCON.6) = 1: SM0/FE用作帧错误指示标志(FE), 可以被软件清除。 0 = 帧错误 (FE)没有发生 1 =检测到帧错误 (FE) 发生.
5	SM2	多机通信模式使能 该位功能取决于串口0模式。 模式 0: 该位选择波特率 FSYS/12 或 FSYS/2。 0 = 时钟运行在 FSYS/12 波特率. 与标志8051兼容。 1 = 时钟运行在 FSYS/2 波特率为更快的串口通信。 模式 1: 该位检查有效停止位。 0 = 无论停止位的逻辑电平, 接收总是有效的。 1 = 只有当接收的停止位是逻辑1并且接收的数据匹配 “GIVEN”或“广播”地址时, 接收才有效。 模式 2 或 3: 为多机通讯 0 = 不管第九位的逻辑电平, 接收总是有效的。 1 =只有当接收的第九位是逻辑1并且接收的数据匹配“GIVEN”或“广播”地址时, 接收才有效。
4	REN	接收使能 0 = 串口0接收禁止。 1 = 工作模式为1,2,3时, 使能接收。工作模式为0时, 在条件 REN = 1 和 RI = 0时接收被初始化。
3	TB8	发送的第九位 串口0在模式2, 3时, 此位是发送的第9位, 模式0,1不用此位..
2	RB8	接收的第九位 比位识别串行端口0模式2或3中的第九接收比特的逻辑电平。在模式1中, RB8是所接收的停止位的逻辑电平。SM2位逻辑电平1时例外。在模式0中不使用RB8。
1	TI	发送中断标志 模式0下该标志在发送完8位数据后由硬件置位, 其它模式下在发送到数据最后一位后由硬件置位。当串口0中断使能, 将执行中断服务程序。该位必须由软件来清除

位	名称	描述
0	RI	接收中断标志 模式0下数据接收帧收到第八位后由硬件置位，模式1，2或3中接收到停止位后由硬件置位。SM2位逻辑电平1时例外。当串口0中断使能，将执行串口0中断服务程序。该位必须由软件清除。

SCON_1 – 串口 1 控制

寄存器	SFR 地址	复位值
SCON_1	F8H, 所有页, 位寻址	0000_0000 b

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

位	名称	描述
7	SM0_1/FE_1	串口1模式选择
6	SM1_1	SMOD0_1 (T3CON.6) = 0: 详见 表 6.9-2 串口 UART1 模式/波特率描述 SMOD0_1 (T3CON.6) = 1: SM0_1/FE_1 用作帧错误状态标志(FE)，可以被软件清除 0 = 没有发生帧错误 (FE)。 1 = 侦测到帧错误(FE)
5	SM2_1	串口1多机通信模式使能 该位功能用于串口1模式 模式0: 无作用 模式1: 位检查有效停止位。 0 = 接收总是有效不管停止位的逻辑电平。 1 = 接收仅在接收停止位为逻辑1,同时接收数据与GIVEN或广播地址匹配时有效 模式2 或 3: 对于多机通信。 0 =接收总是有效不管第9位的逻辑电平。 1 =接收仅在第9位为逻辑1和接收数据与GIVEN或广播地址匹配时有效。
4	REN_1	串口1接收使能 0 = 关闭串口1接收功能。 1 = 打开串口1在模式1, 2或3模式下的接收功能。 模式0下, 打开接收, 需满足配置REN_1=1及RI_1=0
3	TB8_1	串口1第9位发送位 串口1在模式2和3中要被发送的第九位数据。在模式0和1中, 不支持该功能
2	RB8_1	串口1第9位接收位 串口1在模式2和3中接收到的第九位数据。模式1下, 此位是接收到的停止位逻辑电平。SM2_1为1时此位无效。模式0下该位无意义
1	TI_1	串口1发送中断标志位 发送中断标志: 模式0下该标志在发送完8位数据后由硬件置位, 而在其它模式下在发送完帧数据最后一位后硬件置位。当串口1中断使能, 将执行中断服务程序。该位必须由软件来清除。

位	名称	描述
0	RI_1	<p>串口1接收中断标志位</p> <p>该标志由硬件置位。在模式0中，数据帧接收到第8位；模式1，2或3中，数据帧接收到停止位 (stop bit)，当SM2_1=1情况例外。当串口1中断使能，将执行中断服务程序。该位必须由软件来清除。</p>

PCON – 电源控制

寄存器	SFR 地址	复位值
PCON	87H,所有页	POR: 0001_0000b 其它: 000U_0000b

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

位	名称	描述
7	SMOD	串口0波特率加倍使能. 串口0工作在模式2, 或在模式1或3时, 串口0用定时器1溢出作为波特率时钟源时, 设置此位波特率加倍
6	SMOD0	串口0帧错误标志访问使能 0 = SCON.7 访问SM0位 1 = SCON.7访问FE位

T3CON – 定时器 3 控制

寄存器	SFR 地址	复位值
T3CON	C4H, 页 0	0000_0000 b

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

位	名称	描述
7	SMOD_1	串行口1波特率加倍使能. 当串口1在模式2下波特率加倍使能.
6	SMOD0_1	串口1帧错误访问使能 0 = SCON_1.7 访问SM0_1位 1 = SCON_1.7访问FE_1位

SBUF – 串口 0 数据缓存

寄存器	SFR 地址	复位值
SBUF	99H, 页 0	0000_0000 b

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

位	名称	描述
7:0	SBUF[7:0]	串口0 数据缓存 这个寄存器实际上该地址上有2个独立寄存器. 一个用于接收数据, 一个用于发送数据. 当数据写入 SBUF, 这是个发送寄存器并且移位进行串口发送. 当数据从SBUF读出, 数据来自接收寄存器. 每次向SBUF写入一字节数据, 将启动一次数据传输.

SBUF_1 –串口 1 数据缓存

寄存器	SFR地址	复位值
SBUF_1	9AH, 页 0	0000_0000 b

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

位	名称	描述
7:0	SBUF_1[7:0]	<p>串口1数据缓存</p> <p>这个寄存器实际上该地址上有2个独立寄存器. 一个用于接收数据, 一个用于发送数据。当数据写入SBUF_1, 这是个发送寄存器并且移位进行串口发送。当数据从SBUF_1读出, 数据来自接收寄存器。</p> <p>每次向SBUF_1写入一字节数据, 将启动一次数据传输。</p>

IE – 中断使能

寄存器	SFR 地址	复位值
IE	A8H, 所有页, 位寻址	0000_0000 b

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

位	名称	描述
4	ES	使能串口 0 中断 0 = 串口 0 中断禁止. 1 =使能来自 TI (SCON.1) 或 RI (SCON.0) 的中断

EIE1 – 扩展中断使能 1

寄存器	SFR 地址	复位值
EIE1	9CH, 页 0	0000_0000 b

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

位	名称	描述
0	ES1	使能串口 1 中断 0 = 禁止串口 1 中断. 1 = 使能来自 TI_1 (S1CON.1) 或 RI_1 (S1CON.0) 为1.时的中断

SADDR –从机 0 地址

寄存器	SFR 地址	复位值
SADDR	A9H, 页 0	0000_0000 b

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

位	名称	描述
7:0	SADDR[7:0]	从机 0 地址 该字节定义微控器自身的从机地址以用于串口0多机通信

SADEN – 从机 0 地址掩码

寄存器	SFR 地址	复位值
SADEN	B9H, 页0	0000_0000 b

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

位	名称	描述
7:0	SADEN[7:0]	<p>从机0地址掩码.</p> <p>该字节为串口0掩码，为存储"Given"地址的"无关（定义为0）"位。无关位可使更多从机得以灵活运用。</p>

SADDR_1 – 从机 1 地址

寄存器	SFR 地址	复位值
SADDR_1	BBH, 页 0	0000_0000 b

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

位	名称	描述
7:0	SADDR_1[7:0]	从机1地址 该字节定义微控器自身的从机地址以用于串口1多机通信.

SADEN 1 – 从机 1 地址掩码

寄存器	SFR 地址	复位值
SADEN_1	BAH, 页 0	0000_0000 b

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

位	名称	描述
7:0	SADEN_1[7:0]	<p>从机1地址掩码。</p> <p>该字节为串口1掩码为存储“Given”地址的“无关（定义为0）”位。无关位可使更多从机得以灵活运用。</p>

AUXR1 – 辅助寄存器 1

寄存器	SFR地址	复位值
AUXR1	A2H, 页 0	POR: 0000 0000b, 软件复位: 1U00 0000b, nRESET 管脚: U100 0000b, 其它: UUU0 0000b

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

位	名称	描述
2	UART0PX	串口0 管脚切换 0 = 默认RXD 在 P0.7且TXD 在P0.6 1 =切换到RXD在 P0.6 且TXD在 P0.7. 注意 TXD 和 RXD 在设置此位后会立刻切换。请注意不要再收发数据的过程中切换，不然会导致不可预知的情形或告警。

AUXR2 – 辅助寄存器 2

寄存器	SFR 地址	复位值
AUXR2	A1H, 页 2	0000_0000 b

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

位	名称	描述
3:2	UART1TXP	UART1 TX管脚选择 00 = 默认保留 01 = UART1 TX 选择 P1.6 10 = UART1 TX 选择 P3.6 11 = UART1 TX 选择 P1.0
1:0	UART1RXP	UART1 RX 管脚选择 00 = 默认保留 01 = UART1 RX 选择 P0.2 10 = UART1 RX 选择 P3.7 11 = UART1 RX 选择 P0.0

6.10 ISO 7816-3 接口 (SC0~2 和 UART2 ~ 4)

6.10.1 概述

MS51提供基于 ISO/IEC 7816-3 标准异步协议的提ISO 7816-3 接口控制器 (SC 控制器)。程序可以控制 GPIO作为智能卡复位功能或做卡片检测功能。同时也可以模拟高精度波特率通信串口功能。

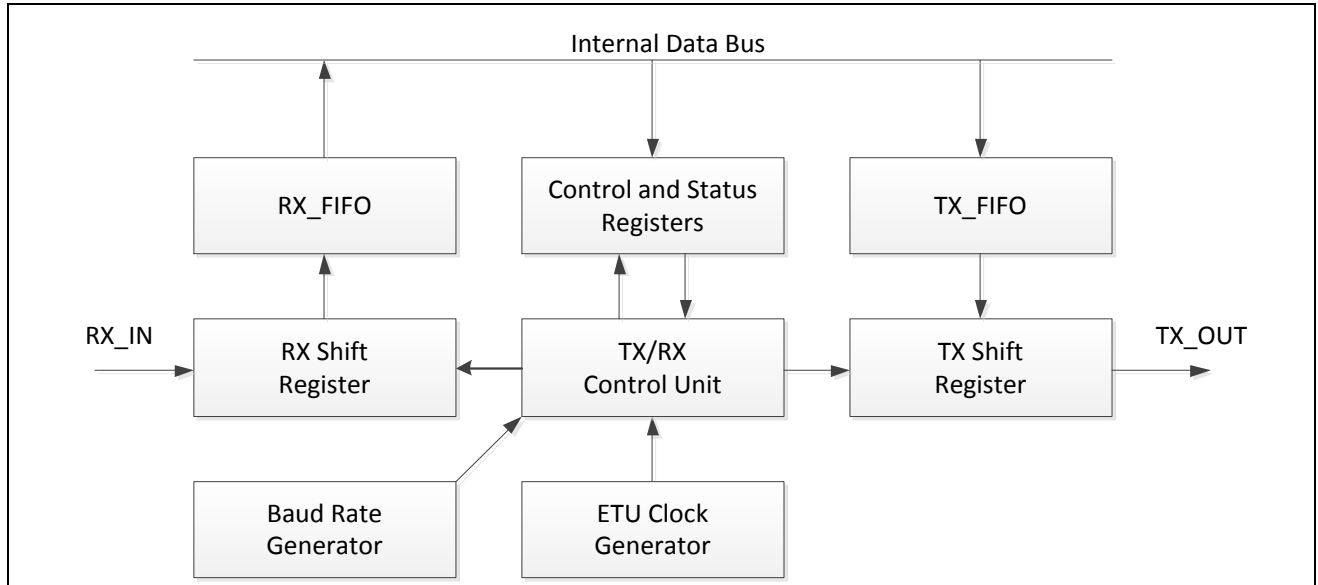


图 6.10-1 智能卡 控制器框图

- 兼容 ISO-7816-3 T=0, T=1
- 可编程的发送时钟频率
- 可编程的块保护时间选择
- 支持自动反向功能
- 支持 UART 模式
 - 全双工异步通信
 - 支持每个通道的可编程的波特率发生器
 - 通过设置SCnEGT，可编程的发送数据延时时间 (最后一个停止位从 TX-FIFO 离开到释放的时间)。
 - 可编程的偶，奇或者无校验位的生成和检测
 - 可编程的停止位，1 位或 2 位的停止位生成

下面列出ISO 7816-3 多功能管脚定义

URAT 管脚	SC 管脚	管脚名称	SFR 定义		
			SFR 字节名称	SFR 位名称	值
UART2_TXD	SC0_CLK	P0.3	AUXR2[7:6]	UART2TXP	01
		P3.0			10
UART2_RXD	SC0_DAT	P0.4	AUXR2[5:4]	UART2RXP	01
		P1.7			10

URAT 管脚	SC 管脚	管脚名称	SFR 定义		
			SFR 字节名称	SFR 位名称	值
UART3_TXD	SC1_CLK	P1.2	AUXR3[3:2]	UART3TXP	01
		P1.5			10
		P0.5			11
UART3_RXD	SC1_DAT	P1.1	AUXR3[1:0]	UART3RXP	01
		P2.5			10
		P3.4			11
UART4_TXD	SC2_CLK	P2.3	AUXR3[7:6]	UART4TXP	01
UART4_RXD	SC2_DAT	P2.2	AUXR3[5:4]	UART4RXP	01

表 6.10-1 智能卡或 UART 管脚定义以及使能控制寄存器

6.10.2 操作模式

6.10.2.1 ISO 7816-3 模式

ISO 7816-3 接口可以由软件控制产生硬件激活，热复位和释放序列。后面会对硬件激活，热复位和释放序列作详细说明。

智能卡接口连接

智能卡接口连接如图 6.10-2 智能卡接口连接

1. SCn_CLK/UARTn_TXD: 智能卡接口时钟 (MCU输出)
2. SCn_DAT/UARTn_RXD: 智能卡数据 (双向管脚)
3. SCn_RST: 智能卡复位脚 (MCU输出, 软件指定GPIO)
4. SCn_PWR: 智能卡电源控制脚 (MCU输出, 软件指定 GPIO)
5. *: SCn_PWR 是用来控制 ISO 7816-3电源的开关, 请不要将SCn_PWR作为ISO 7816-3的供电管脚。
6. SCn_CD: 智能卡插入检测管脚 (MCU输入, 检测卡片插入)

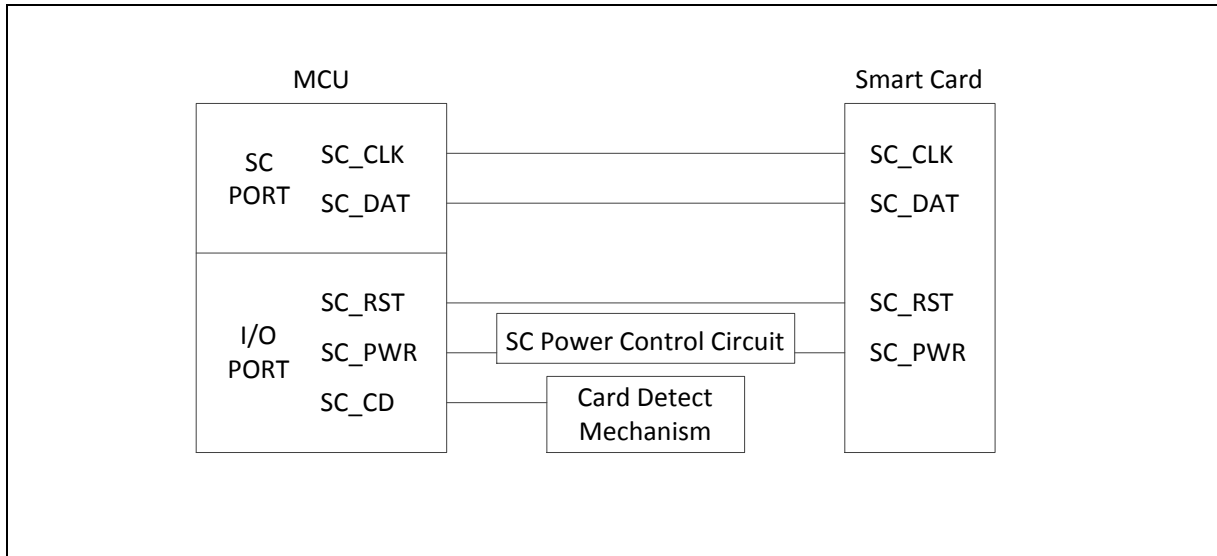


图 6.10-2 智能卡接口连接

激活和冷复位

激活和冷复位的流程见图 6.10-3 智能卡激活和冷复位流程

1. 软件设置SCn_RST为低
2. 在T1和SCn_DAT为高（接收模式）之前设置SCn_PWR为高
3. T1之后设置CLKKEEP (SCCR2[1]) 为 ‘1’使能SCn_CLK时钟
4. T2后释放SCn_RST为高
5. ISO 7816-3主机控制器在T3时读卡片ATR

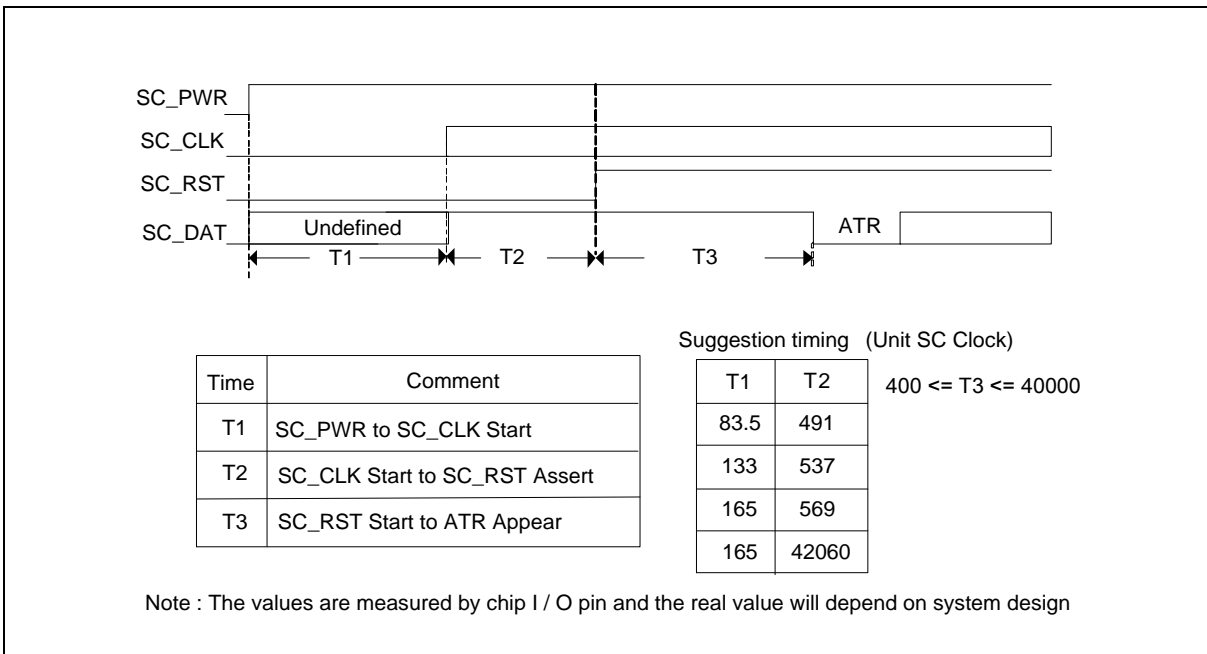


图 6.10-3 智能卡激活和冷复位流程

热复位

热复位流程见图 6.10-4 智能卡热复位流程

1. T4前设置SCn_RST为低
2. T4时设置Set SCn_DAT为高
3. T5之后设置SCn_RST为高
4. T6时ISO 7816-3 主机控制器读卡片ATR

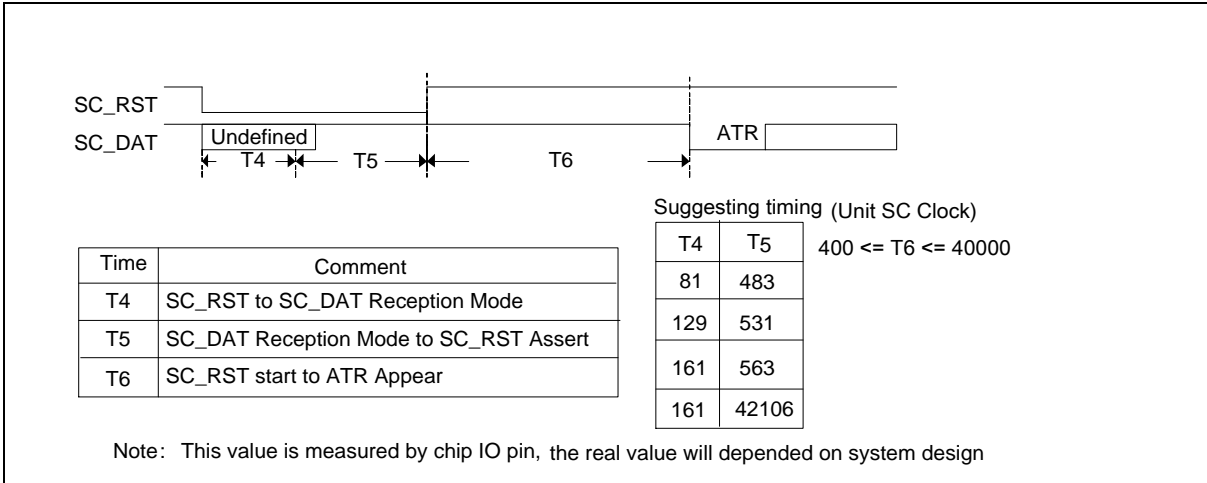


图 6.10-4 智能卡热复位流程

释放

释放流程见图 6.10-5 智能卡释放流程

1. T7时设置SCn_RST为低
2. T8时设置CLKKEEP(SCCR2[1])为 '0' 停止SCn_CLK时钟
3. T8时设置SCn_DAT为低
4. T9时拉低SCn_PWR.

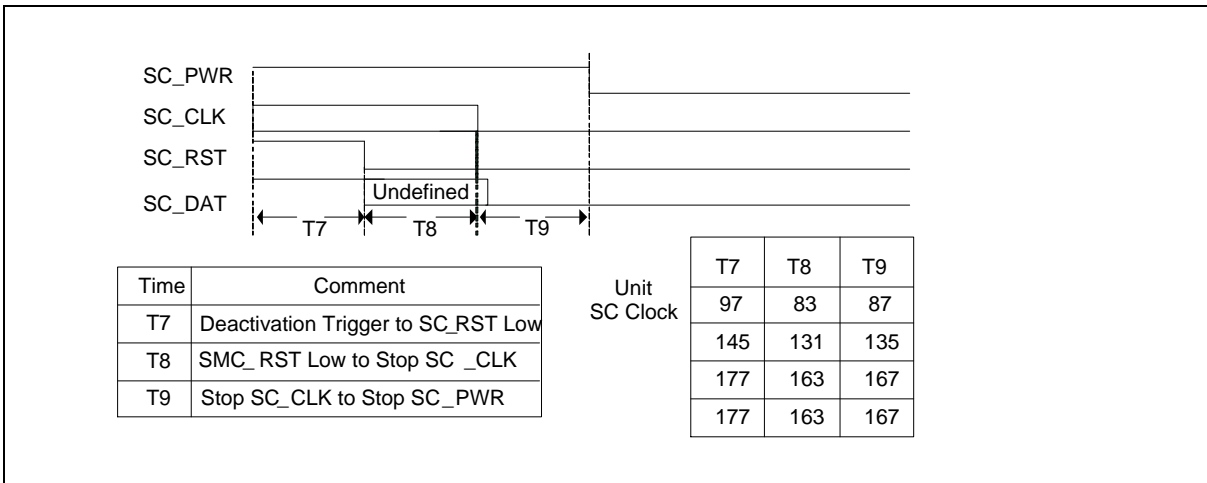


图 6.10-5 智能卡释放流程

6.10.2.2 UART 模式

当 UARTEN (SCnCR2[0]) 被置位，智能卡控制器也可以作为一个基本的 UART 功能来使用。下面是一个 UART 模式的编程例子。

编程示例：

1. 设置UARTEN(SCnCR2[0])位进入UART模式
2. CONSEL(SCnCR1[4])和AUTOCEM(SCnCR1[3])清0。(在UART模式下，这些区域要填0)
3. 设置 ETURDIV[11:0]({SCnETURD1[3:0]:SCnETURD0[7:0]})确定UART波特率。例如：如果智能卡时钟是12MHz，目标波特率是115200bps，ETURDIV需要填入 $(12000000 / 115200 - 1)$ 。
4. 设置数据格式，包括数据位数(设置WLS(SCCR2[5:4])、奇偶校验模式(设置OPE (SCCR2[7])和 PBOFF (SCCR2[6]))以及停止位长度(设置NSB(SCnCR1[7]))。
5. 写SCnDR (SCnDR[7:0])(TX) 寄存器或读SCnDR(SCnDR[7:0])(RX)寄存器姐可以操作UART功能。

6.10.3 ISO 7816-3 数据传输

智能卡ISO 7816-3接口扮演的就是一个半双工异步通信端口的角色，它的数据格式如下图所示的 10 个连续位组成。

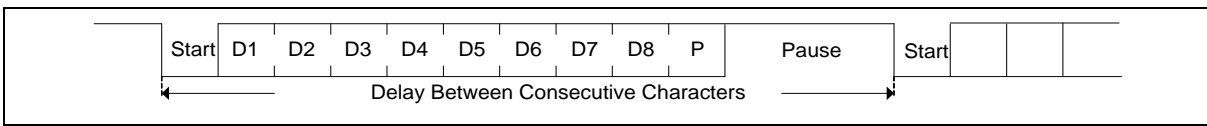


图 6.10-6 智能卡数据格式

6.10.4 初始化字符 TS

依据 ISO 7816-3 协议，初始化字符 TS 有两种可能的模式，如图6.10-3所示。如果TS模式是 1100_0000，则是反向约定。当按反向约定进行解码后，则传送的字节等于0x3F。如果TS模式是 1101_1100，则是直接约定。当按直接约定解码后，则传送的字节等于 0x3B。用户可以设置 AUTOCEM (SCnCR1[3])让硬件来决定操作的约定。用户也可以设置CONSEL(SCnCR1[4])为' 00' 或者' 11'，在收到 ATR 的 TS 后才去改变操作的约定。

如果软件通过设置 AUTOCEM (SCnCR1[3])使能自动约定功能，则设置步骤必须在 ATR 状态之前完成，而且第一个数据必须是 0x3B 或 0x3F。在硬件收到第一个数据并放到SCn_DAT中之后，硬件将决定约定模式并自动改变CONSEL(SCnCR1[4])。如果第一个数据既不是 0x3B也不是0x3F，硬件将产生一个 INT_ACON_ERR中断给CPU(如果ACERRIEN(ScnIE[4] = '1'))。

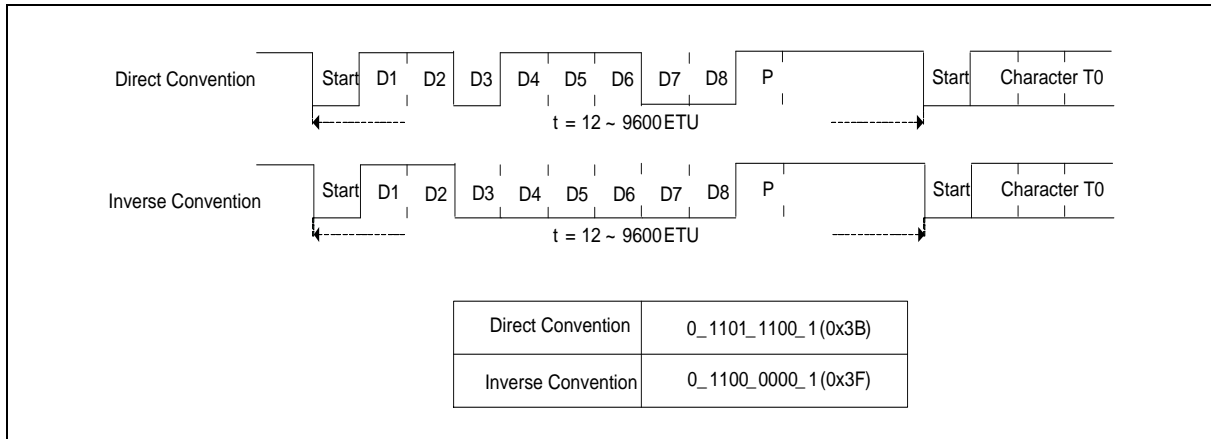


图 6.10-7 初始化字符 TS

6.10.5 错误信号和字符重复

依据 ISO 7816-3 T=0模式描述，如图 6.10-8 智能卡错误信号所示，如果接收端收到错误的奇偶校验位，会拉低SC_DAT 1.5个位宽，提示发送端奇偶校验错误。然后发送端会重复发送字符。智能卡控制器接口提供接收端硬件错误检测功能(SC0TSR[4])，智能卡控制器会产生传输错误中断给CPU (如果 TERRIEN(SCnIE[2] = '1'))。

在 T=1 模式，接收端不会拉低SC_DAT 1.5位宽提示发送端奇偶校验错误。

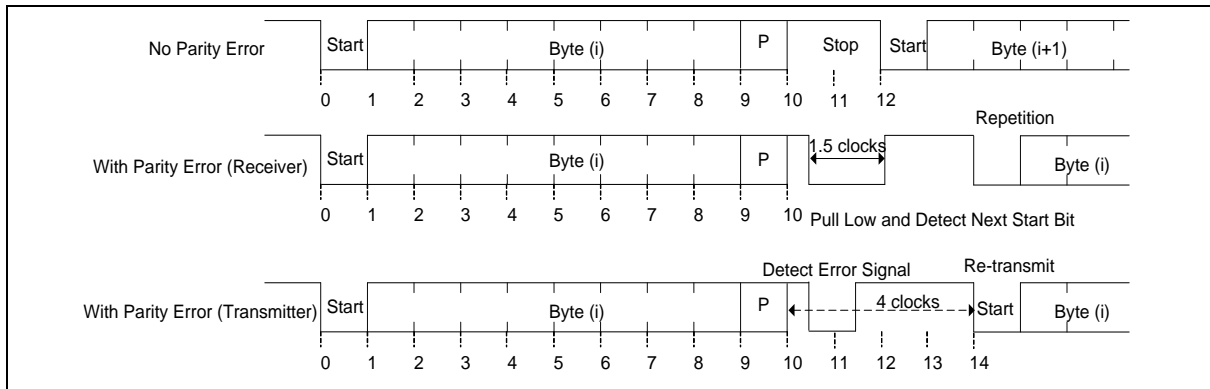


图 6.10-8 智能卡错误信号

6.10.6 块保护时间和扩展保护时间

块保护时间是在不同传输方向之间的两个连续字符的第一位边缘之间的最短位长度。表示用于块保护时间的计数器。

依据 ISO7816-3，在 T=0 的模式下，软件必须填充该域为 15 (实际的块保护时间=16.5)，在 T=1 的模式下，软件必须填充该域位 21 (实际的块保护时间为 22.5)。

发送时，智能卡先发送数据到智能卡主控制器。在一段时间 (16.5 or 22.5, 由 T 位设置) 后，智能卡主控制器开始发送数据。

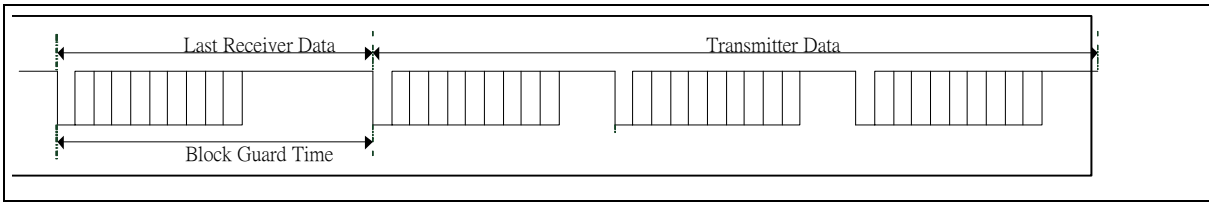


图 6.10-9 发送方向块保护时间操作

接收时，智能卡主控制器先发送数据给智能卡。如果智能卡发送数据给智能卡主控制器的时间小于 16.5或 22.5（由T位设置），当RXBGTEN(SCnCR1[5])和BGTIEN(ScnIE[3])都使能时，块保护时间中断BGTIF(ScnIS[3])会产生。

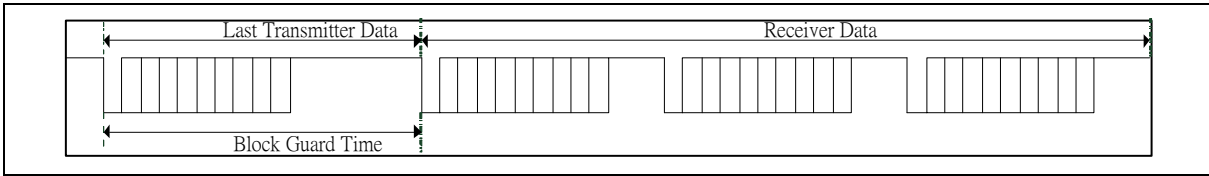


图 6.10-10 接收方向块保护时间操作

额外保护时间是EGT (SCnEGT[7:0])，它只影响智能卡接口发送的数据，具体格式如图 6.10-11额外保护时间操作

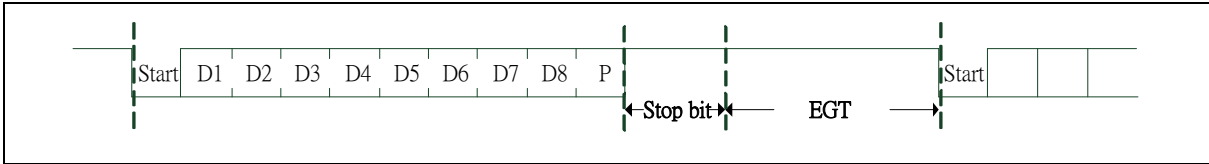


图 6.10-11 额外保护时间操作

6.10.7 智能卡控制寄存器

SCnCR0 – 智能卡控制寄存器 0

寄存器	SFR 地址	复位值
SC0CR0	F1H, 页 2	0000_0000 b
SC1CR0	F3H, 页 2	0000_0000 b
SC2CR0	F5H, 页 2	0000_0000 b

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

位	名称	描述
7	NSB	<p>停止位长度</p> <p>这一位指定停止位长度。</p> <p>0 = 停止位长度是 2 ETU.</p> <p>1= 停止位长度是 1 ETU.</p> <p>注意: 默认的停止位长度是 2. SC 和 UART 都采用NSB设置停止位长度.</p>
6	T	<p>T 模式</p> <p>0 = T0 (ISO 7816-3 T = 0 模式).</p> <p>1 = T1 (ISO 7816-3 T = 1 模式).</p> <p>T 模式控制(快保护时间). 块保护时间是在不同传输方向之间的两个连续字符的第一位边缘之间的最短位长度。该域表示用于块保护时间的计数器。依据 ISO7816-3, 在 T=0 的模式下, 软件必须填充该域为 15 (实际的块保护时间=16.5), 在 T=1 的模式下, 软件必须填充该域位 21 (实际的块保护时间为 22.5)。</p> <p>注意:在 T = 0模式, 接收端会产生奇偶校验标志PEF(SC0TSR[4]),如果检测到奇偶校验错误会产生奇偶校验错误信号给发送端。在 T = 1 模式, 接收端会产生奇偶校验标志 PEF(SC0TSR[4]), 如果检测到奇偶校验错误却不会产生奇偶校验错误信号给发送端。</p> <p>注意: 参见 6.10.5</p>
5	RXBGTEN	<p>接收块保护功能使能位</p> <p>0 = 接收块保护功能禁止.</p> <p>1 = 接收块保护功能使能.</p>
4	CONSEL	<p>约定选择</p> <p>0 = 直接约定.</p> <p>1 = 反向约定.</p> <p>注意 1:如果 AUOCEN(SCnCR0[3]) 写 “1”, 此位自动清0</p> <p>注意 2: 如果 AUOCEN(SCnCR0[3]) 使能, 硬件决定约定选择, 在SCEN (SCnCR0[0]) 为1后自动改变CONSEL (SCnCR0[4]) 的值</p>

位	名称	描述
3	AUTOGEN	<p>自动约定使能位</p> <p>0 = 禁止自动约定.</p> <p>1 = 自动约定使能。当硬件ATR状态时接收到TS，当TS是直接约定，CONSEL(SCnCR0[4])会自动写0，相反当TS是反向约定，CONSEL(SCnCR0[4])写1.</p> <p>注意:如果程序使能自动约定功能，在ATR状态之前设置应该完成，且第一笔数据必须是0x3B或0x3F.当硬件接收到第一笔数据到缓存里，硬件会自动决定约定，并改变CONSEL(SCnCR0[4])位</p>
2	TXOFF	<p>TX 发送禁止位</p> <p>0 =发送端使能.</p> <p>1 = 发送端禁止</p>
1	RXOFF	<p>RX 接收端禁止位</p> <p>0 =接收端使能</p> <p>1 = 接收端禁止</p> <p>注意:如果AUTOGEN(SCnCR0[3])使能，这一位忽略.</p>
0	SCEN	<p>智能卡引擎使能位</p> <p>设置此位为1使能智能卡操作。如果这一位清0，智能卡会强迫所有传输进入空闲态.</p> <p>注意: SCEN 必须在写其他智能卡寄存器之前置1，不然ISO 7816-3 会工作不正常</p>

SCnCR1 – SC 控制寄存器

寄存器	SFR 地址	复位值
SC0CR1	F2H, 页 2	0000_0000 b
SC1CR1	F4H, 页 2	0000_0000 b
SC2CR1	F6H, 页 2	0000_0000 b

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

位	名称	描述
7	OPE	奇检验使能位 0 = 发送偶数个逻辑1或接收到相应的奇偶校验位. 1 = 发送奇数个逻辑1或接收到相应的奇偶校验位. 注意: 此位只在 PBOFF 位为 '0' 时有效
6	PBOFF	奇偶校验禁止控制 0 = 在最后一位数据和停止位之间插入奇偶校验位或检测奇偶校验位。 1 = 发送时不产生奇偶校验位, 接收时不检测奇偶校验位 注意: ISO 7816-3 模式下, 这一位要设为 '0' (默认值)
5:4	WLS	数据位数选择 00 = 数据位数 8 bits. 01 = 数据位数 7 bits. 10 = 数据位数 6 bits. 11 = 数据位数 5 bits. 注意: ISO 7816-3 模式下 WLS 要是 '00'
3	TXDMAEN	SC/UART 发送 DMA 使能 这一位使能 SC/UART 发送 PDMA 传输功能, 在 SC/UART 开始前发送数据需要在 XRAM 里准备好 0 = SC/UART 发送 DMA 禁止 1 = SC/UART 发送 DMA 使能
2	RXDMAEN	SC/UART 接收 DMA 使能 这一位使能 SC/UART 通过 PDMA 接收数据, SC/UART 接收到数据后自动存入 XRAM 0 = SC/UART 接收 DMA 禁止 1 = SC/UART 接收 DMA 使能
1	CLKKEEP	智能卡 时钟使能位 0 = 禁止智能卡时钟. 1 = 产生智能卡时钟.

位	名称	描述
0	UARTEN	<p>UART 模式使能位</p> <p>0 = ISO 7816-3 模式.</p> <p>1 = UART 模式.</p> <p>注意1:当工作在 UART模式, 用户需要设置CONSEL (SCnCR0[4]) = 0 且 AUTOZEN(SCnCR0[3]) = 0.</p> <p>注意 2:当工作在 ISO 7816-3 模式, 用户需要设置 UARTEN(SCnCR1 [0]) = 0.</p> <p>注意 3:当 UART 使能后, 硬件会复位FIFO和内部状态机</p>

SCnDR –智能卡数据寄存器

寄存器	SFR 地址	复位值
SC0DR	D9H, 页 2	0000_0000 b
SC1DR	E1H, 页 2	0000_0000 b
SC2DR	E9H, 页 2	0000_0000 b

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

位	名称	描述
7:0	SCnDR[7:0]	<p>SC / UART 缓存数据</p> <p>这个字节用来缓存SC / UART 总线的收发数据。写这个寄存器会写到移位寄存器，读这个寄存器会读到接收缓存的数据。</p> <p>注意: 如果SCEN(SCnCR0[0])没有使能, SCnDR 不可修改.</p>

SCnEGT – 智能卡额外保护时间寄存器

寄存器	SFR 地址	复位值
SC0EGT	DAH, 页 2	0000_0000 b
SC1EGT	E2H, 页 2	0000_0000 b
SC2EGT	EAH, 页 2	0000_0000 b

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

位	名称	描述
7:0	SCnEGT[7:0]	智能卡额外保护时间 这里指定额外保护时间的值 注意: 计数器单位时 ETU .

SCnETURD0 –智能卡 ETU 除频寄存器

寄存器	SFR 地址	复位值
SC0ETURD0	DBH, 页 2	0111_0011 b
SC1ETURD0	E3H, 页 2	0111_0011 b
SC2ETURD0	EBH, 页 2	0111_0011 b

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

位	名称	描述
7:0	ETURDIV[7:0]	ETU 除频寄存器的低位 实际的ETU 是 ETURDIV[11:0] + 1. 注意 1: ETURDIV[11:0]必须大于 0x004. 注意 2: SCnETURD0 必须首先配置, 然后是 SCnETUDR2.

SCnETURD1 –SC ETU除频寄存器

寄存器	SFR 地址	复位值
SC0ETURD1	DCH, 页 2	0011_0001 b
SC1ETURD1	E4H, 页 2	0011_0001 b
SC2ETURD1	ECH, 页 2	0011_0001 b

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

位	名称	描述
7	-	保留
6:4	SCDIV [2:0]	智能卡时钟除频 000 = F_{SC} 是 $F_{SYS}/1$. 001 = F_{SC} 是 $F_{SYS}/2$. 010 = F_{SC} 是 $F_{SYS}/4$. 011 = F_{SC} 是 $F_{SYS}/8$. (默认.) 100 = F_{SC} 是 $F_{SYS}/16$. 101 = F_{SC} 是 $F_{SYS}/16$. 110 = F_{SC} 是 $F_{SYS}/16$. 111 = F_{SC} 是 $F_{SYS}/16$. 注意: 依照 ISO/IEC 7816-3 F_{SC} 时钟需要在 1Mhz ~ 5Mhz
3:0	ETURDIV [11:8]	ETU 除频寄存器的高位 这一位指定除频寄存器的高位. ETU 实际是 $ETURDIV[11:0] + 1$. Note 1: ETURDIV[11:0] 需要大于 0x004. Note 2: SCnETURD0 需要首先配置, 然后是 SCnETURD1 .

ScnIE – 智能卡中断使能控制寄存器

寄存器	SFR 地址	复位值
SC0IE	DCH, 页 2	0000_0000 b
SC1IE	E4H, 页 2	0000_0000 b
SC2IE	ECH, 页 2	0000_0000 b

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

位	名称	描述
7:4	-	保留
4	ACERRIEN	自动约定错误中断使能位 这里使能自动约定错误中断 0 = 禁止自动约定错误中断 1 = 使能自动约定错误中断
3	BGTIEN	块保护时间中断使能位 这里使能快保护时间中断 0 = 禁止快保护时间中断 1 = 使能块保护时间中断
2	TERRIEN	传输错误中断使能位 这里使能传输错误中断。传输错误状态在 SC0TSR 寄存器，包含接收暂停错误 BEF(SC0TSR[6]), 帧错误 FEF(SC0TSR[5]), 奇偶校验错误 PEF(SC0TSR[4]), 接收缓存溢出错误 RXOV(SC0TSR[0]) 和发送缓存溢出错误 TXOV(SC0TSR[2]). 0 = 禁止传输错误中断 1 = 使能传输错误中断.
1	TBEIEN	发送缓存空中断使能位 这里使能发送缓存空中断 0 = 禁止发送缓存空中断 1 = 使能发送缓存空中断
0	RDAIEN	接收数据到达中断使能位 这里使能接收数据中断 0 = 禁止接收数据中断 1 = 使能接收数据中断.

ScnIS – 智能卡中断状态寄存器

寄存器	SFR 地址	复位值
SC0IS	DEH, 页 2	0000_0010 b
SC1IS	E6H, 页 2	0000_0010 b
SC2IS	EEH, 页 2	0000_0010 b

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

位	名称	描述
7:5	-	保留
4	ACERRIF	自动约定错误中断状态标志（只读） 这里指示自动约定序列错误。如果在ATR状态接收到TS不是0x3B 或 0x3F, 这一位置位。 注意: 此位只读，写0清0
3	BGTIF	块保护时间中断状态标志（只读） 这里指示块保护时间状态标志 注意 1: 这一位当RXBG TEN (SCnCR0[5]) 使能时有效。 注意 2: 此位只读，写0清0。
2	TERRIF	传输错误中断状态标志（只读） 这里用作传输错误中断标志，传输错误状态在 SC0TSR 寄存器，包含接收暂停错误 BEF(SC0TSR[6]), 帧错误 FEF(SC0TSR[5]), 奇偶校验错误 PEF(SC0TSR[4]), 接收缓存溢出错误 RXOV(SC0TSR[0]) 和发送缓存溢出错误 TXOV(SC0TSR[2])。 注意: 这一位时 BEF(SC0TSR[6]), FEF(SC0TSR[5]), PEF(SC0TSR[4]), RXOV(SC0TSR[0]) 和 TXOV(SC0TSR[2])的状态标志，所以如果软件想要清除这一位，需要对这些位都写0
1	TBEIF	发送缓存空中断状态标志（只读） 这里用作发送缓存空中断状态标志 注意: 这里用作发送缓存空中断状态标志，如果软件想清除这一位，软件需要向.DAT(SCnDR[7:0]) 缓存写入数据，这一位会自动清0
0	RDAIF	接收数据到达中断状态标志（只读） 这里用作接收数据中断状态标志 注意: 这里是接收数据状态标志。如果软件从SC_DAT 管脚读取数据，此位自动清除

SCnTSR – 智能卡传输状态寄存器

寄存器	SFR 地址	复位值
SC0TSR	DFH, 页 2	0000_1010 b
SC1TSR	E7H, 页 2	0000_1010 b
SC2TSR	EFH, 页 2	0000_1010 b

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

位	名称	描述
7	ACT	发送接收忙碌状态标志（只读） 0 = 此位在发送接收完成后自动清0 1 = 当发送接收传输正在进行时此位由硬件置1
6	BEF	接收中断错误状态标志（只读） 此位在输入数据管脚上由超过一个完整的字长时间内保持为空闲态（逻辑0），完整字长时开始位加数据位，奇偶校验位和停止位。 注意: 此位只读，写0清0.
5	FET	接收帧错误状态标志（只读） 当接收字符没有有效的停止位时此位置1。（在最后一笔数据位或奇偶校验位后检测到逻辑0） 注意: 此位只读，写1清0.
4	PEF	接收奇偶校验错误状态标志（只读） 当接收到的字符没有有效的奇偶校验位时置1 注意: 此位只读，写0清0.
3	TXEMPTY	发送缓存空状态标志（只读） 此位指示发送缓存是否为空 注意: 当发送缓存的数据传输到发送移位寄存器时此位置1，当写数据到DAT(SCnDR[7:0])后清0 (发送缓存非空).
2	TXOV	发送溢出错误状态标志（只读） 如果发送缓存已满，再写入数据到DAT(SCnDR[7:0]) 硬件会让此位置1。 注意: 此位只读，写0清0.
1	RXEMPTY	接收缓存空状态标志（只读） 此位指示接收缓存是否为空 注意: 当接收缓存的数据被CPU读取之后，硬件会把此位置1，当智能卡接收到新的数据后此位清0

位	名称	描述
0	RXOV	接收缓存溢出错误状态标志（只读） 此位当接收缓存溢出时置1。 注意： 此位只读，写0清0

AUXR2 – 辅助寄存器 2

寄存器	SFR 地址	复位值
AUXR2	A1H, Page 2	0000_0000 b

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

位	名称	描述
7:6	UART2TXP	串口2 TX 引脚选择 00 = 默认保留 01 = TX 选择 P0.3 10 = TX 选择 P3.0 11 = 保留
5:4	UART2RXP	串口2 RX 引脚选择 00 = 默认保留 01 = RX 选择 P0.4 10 = RX 选择 P1.7 11 = 保留

AUXR3 – 辅助寄存器 3

寄存器	SFR 地址	复位值
AUXR3	A2H, Page 2	0000_0000 b

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

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

6.11 I²C总线接口

6.11.1 概述

MS51提供了两个内部集成的I²C总线作为一种串行通信方式，用在MCU与I²C设备之间通讯。例如：EEPROM、LCD模块、温度传感器等等。I²C用两条线(数据线SDA和时钟线SCL)在设备间传输数据。

I²C总线用作主机与从机之间双向数据传输。可以用于多主机系统，支持无中央主机及多主机系统，主机与主机之间的总线仲裁传输，同步时钟SCL的存在，允许设备间使用不同比特率的数据传输。支持四种传输模式：主发、主收、从发、从收。I²C总线仅支持7位地址。支持广播呼叫、支持标准速率传输(100kbps)和快速传输(400k bps)。

6.11.2 功能描述

对于双向传输操作，I2C0_SDA及I2C0_SCL引脚必须配置成开漏模式，形成逻辑线与功能：总线上当有一个器件输出0，总线上就是0电平，所有器件全输出1，总线上才是高电平，需要通过外接上拉电阻把电平拉高。MS51在设置I2CEN使能I2C功能之前，必须把I2C0_SCL，I2C0_SDA的输出锁存在逻辑1的状态

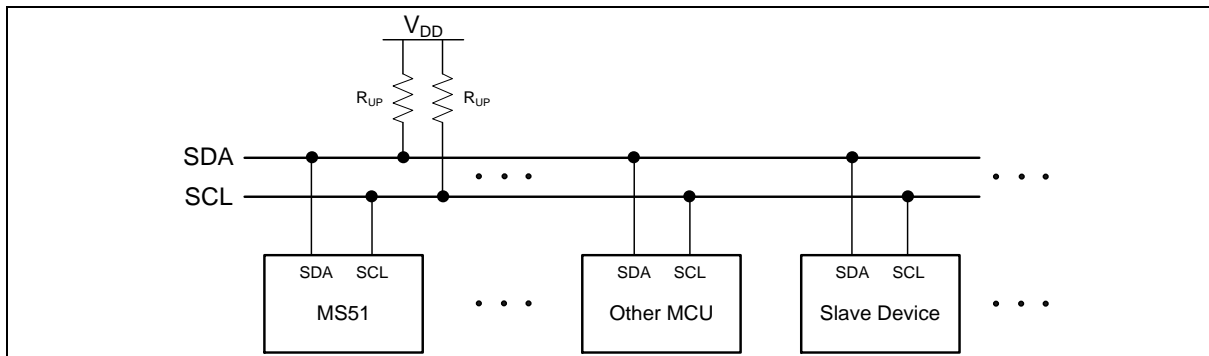


图 6.11-1 I²C 总线连接

I²C空闲时，两条线都为高。这时任一设备都可以做为主机发个起始位START开始数据传输，在停止位STOP出现之前，总线被认为处于忙状态。主机产生串行时钟脉冲以及起始位和停止位。如果总线上没有START起始信号，则所有总线设备被认为未被寻址从机，硬件自动匹配自己的从机地址或广播呼叫地址，(广播地址可由GC(I2CnADDRx.0)使能或禁止.)。若地址匹配，就产生中断。

I2C总线上传输的每个字节都包含8个数据位(MSB优先)和一个应答位，共9位。但每次传输的字节个数没有明确界定(起始位START和停止位STOP之间的字节个数)，但每个字节都应有一个应答位。主机产生8个时钟脉冲，以传输8位数据。在第8个时钟SCL下沿后，由I2C0_SDA脚输出数据后，I2C0_SDA转为输入模式在第9个时钟脉冲以读取第9位应答位。在第9个时钟脉冲后，数据接收端若没准备好接收下一个字节，可以拉住时钟线I2C0_SCL保持低，让传输挂起。接收端释放时钟线I2C0_SCL以后，传输继续。

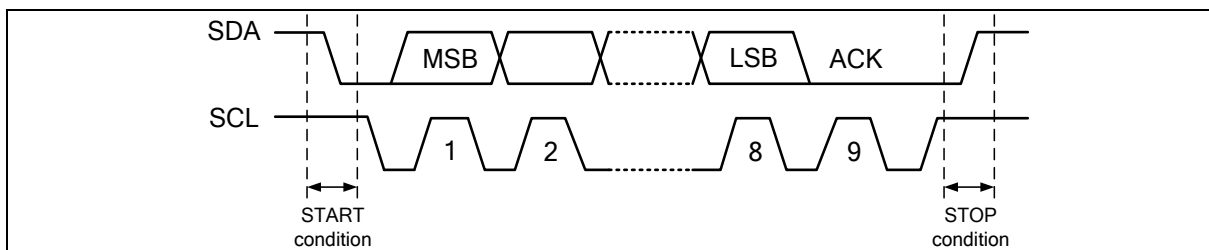


图 6.11-2 I²C 总线协议

6.11.2.1 开始和停止条件

I²C 总线时序定义了起始START (S)和结束STOP (P)的条件。时钟I2C0_SCL为高时，数据线I2C0_SDA由高电平至低电平的跳变被认为是起始START标志。时钟I2C0_SCL为高时，数据线I2C0_SDA由低电平至高电平的跳变被认为是结束STOP标志。起始和结束都由主机产生，起始和结束之间被认为是总线忙状态。当成功判定结束条件以后，主机释放总线，所有设备都回到监听总线起始位状态，之前被呼叫从机也转为未寻址从机。I²C总线进入空闲状态等待下一个起始START信号，开始下一次传输。

主机若发出停止位STOP，传输就停止了。然而，主机也可以不发停止位，而是再次发出起始START (Sr) 信号继续和上个地址通信，或者换个地址继续通信。各种读/写组合格式操作可能存在于一个传输中。

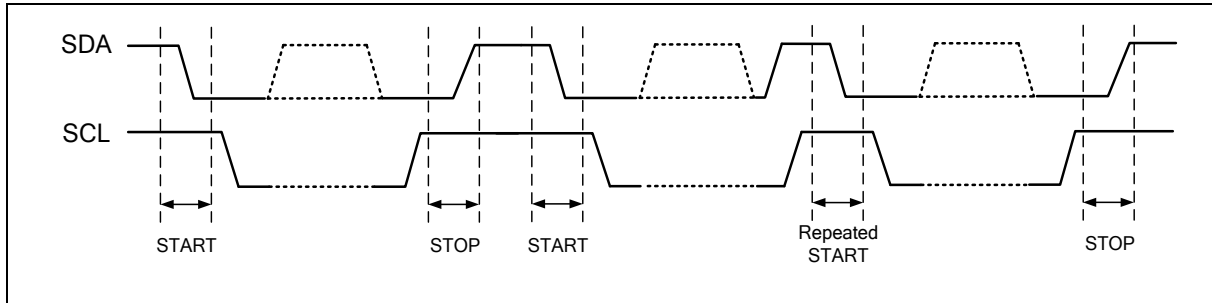


图 6.11-3 启动、重复启动、停止条件

6.11.2.2 7位地址数据格式

起始位 START 之后，第一个字节必须是“7位从机地址SLA+第8位读写方向位(读/写)”，用以定义目标从机地址以及从机写入或读出数据。若第8位是0，即SLA+W，表示下个字节开始主机向从机写数据；若是1，即SLA+只读，就表示下个字节开始，主机向从机读数据。一个地址包包含一个从机地址和一个读或写位叫SLA+只读 或 SLA+W。所以，一个典型数据包包含起始位 START，SLA+W/只读，一个或多个字节数据，最后是停止位 STOP。当一个从机已被定址及读写方向通过字节SLA+W/只读，随后的8位数据就跟随之前的读写设定进行传输。

图 6.11-4 主机发送数据给一个7位地址从机，主机定址一个7位地址从机和1位写方向来表明主机想发送数据给从机。在从机返回一个应答信号后，主机将发送数据。

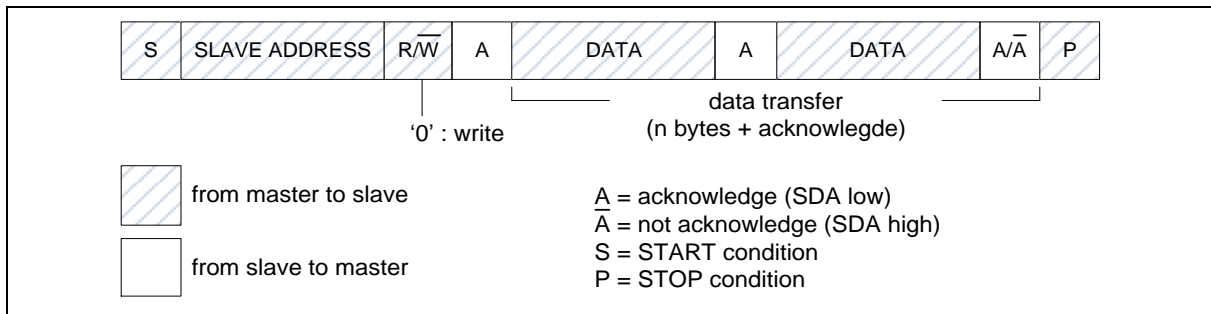


图 6.11-4 主机发送数据给一个 7 位地址从机

图 6.11-4 主机发送数据给一个7位地址从机所示主机从一个7位地址从机读取数据。主机定址一个7位地址从机和1位读方向来表明主机想从从机读取数据。在从机返回一个应答信号后，从机将发送数据。

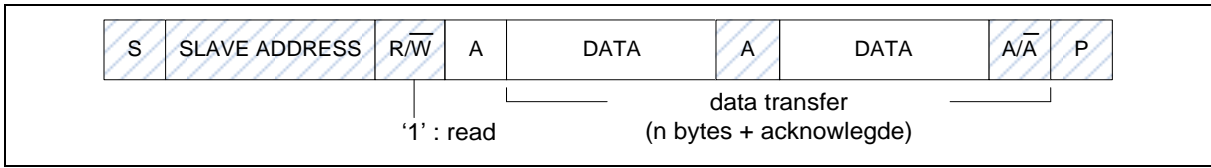


图 6.11-5 主机从一个 7 位地址从机读取数据

I²C总线还有一种特殊寻址方式，广播呼叫寻址。在该模式下，发送的首字节数据为0。广播呼叫模式应用于主机希望向所有从机传输相同数据。当此寻址方式启用。收到广播要不要发应答由软件决定。若某个从机发了应答，这个从机就收发后续数据，和标准从机接收方式相同。注意：地址0x00默认用于广播呼叫方式，不能用于普通从机地址。因此理论上，总共7位地址I²C总线，共可以连接127个设备，地址由1至127。

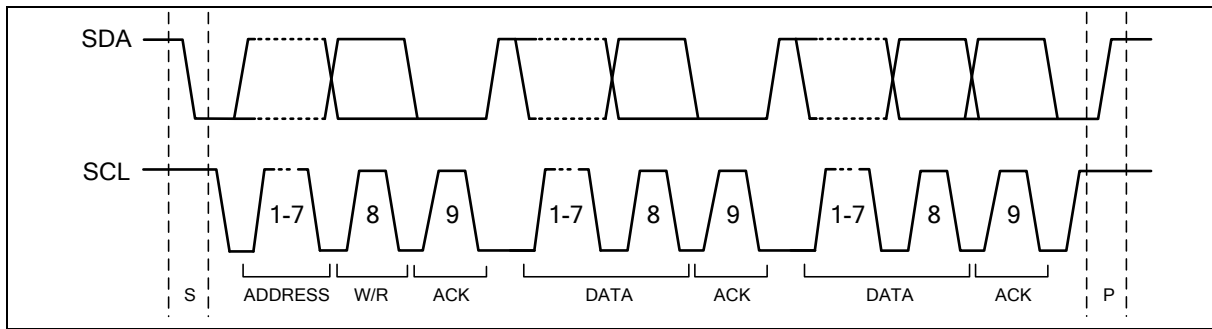


图 6.11-6 一次传输中的 I2C 数据格式

在数据传输过程中，在时钟I2C0_SCL高电平时，I2C0_SDA需要保持数据内容不能更改。只有在I2C0_SCL为低时，I2C0_SDA内容可以改变。

6.11.2.3 应答

每字节传输I2C0_SCL第9个脉冲用于传输应答位 (ACK)。通过将I2C0_SDA拉低，来允许接收端（可以是主机或是从机）回应发送端（可以是主机或从机）。应答位时钟由主机产生。发送端设备在应答位时钟高电平周期内，需释放对I2C0_SDA的控制。ACK 为一个低电平信号。在应答位时钟周期的高电平时，SDA保持低电平用以表示接收端已成功接收到发送端的数据。通常被寻址的从机在整个传输过程中每字节都需要回复应答位。当该从机接收无法应答 (NACK) 从机地址时，从机将SDA线保持高以便主机产生停止(STOP)或发送重复开始(START)信号。

若从机接收没有应答从机地址后，将自身切换到未定址从机模式，从而无法接收更多数据字节。这时从机将I2C0_SDA线拉高，此时主机应发送停止STOP信号或重复起始(repeated START)信号。

若是主机接收，主机控制着收发字节个数，主机在最后一个字节收发结束后不发应答位信号，从机端将切换为未定址从机模式，并释放I2C0_SDA线，以便主机直接发停止位STOP 或重新开始位START。

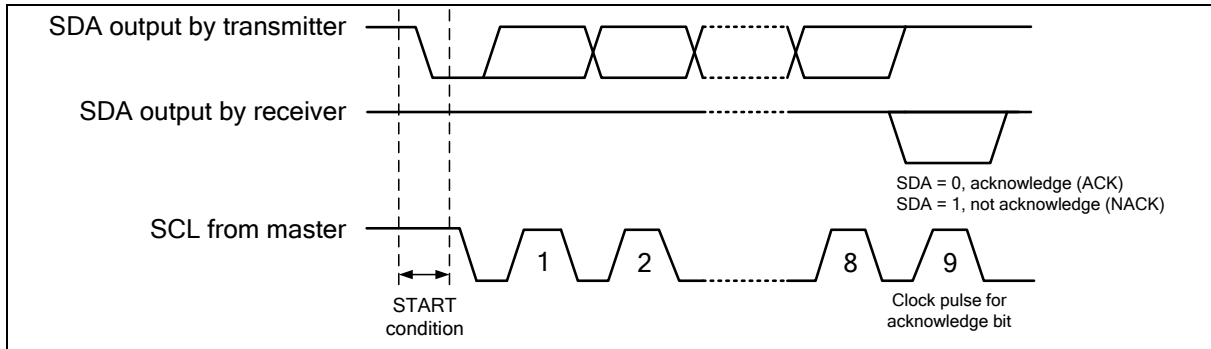


图 6.11-7 应答位

6.11.2.4 仲裁

主机仅可在总线空闲时发起传输。可能有多个器件同时发开始位试图发起数据传输，这时就会出现总线仲裁。在该状态下，当I2C0_SCL为高时，I2C0_SDA上呈现仲裁信号。在仲裁过程中，第一个发起主机对I2C0_SDA线置1（高电平）而另一个主机发送0（低电平），发送后主机会对I2C0_SDA线上信号与自己发出的信号进行比较，由于“线与”的原因，时钟I2C0_SCL为高时，发送0的主机会成功，而发送1的主机会失败。发送失败的主机立刻切换自身到未被寻址的从机状态，以确保自身能被仲裁胜利的主机寻址到。同时也释放数据线，并回到地址侦测状态，仲裁失败的主机，仍会发送时钟，直到当前字节结束。

仲裁机制让每个主机发送数据时，都会同时比较总线上的数据是否与自己发送的一致。注：如果其它主机发送0，发送1的主机会在仲裁中失败，仲裁会持续到总线上只有一个主机。如两个主机同时向一个从机发数据时，地址相同，仲裁会持续比较数据位或应答位。

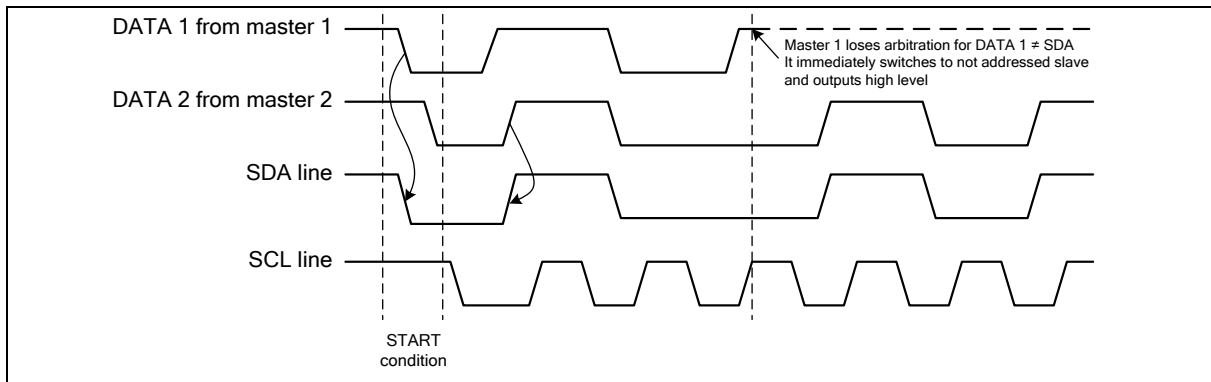


图 6.11-8 两台主机仲裁过程

I²C 总线的这种仲裁机制，让总线上的设备可以有多个主机，而且没有优先等级。从机不介入仲裁。

6.11.2.5 操作模式

I²C 接口支持三种操作模式：主机，从机和广播模式。

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。如果检测到这些地址中的一个，并且从机愿意与主机接收或发送数据(通过设置AA位)，在第9个时钟时发送应答脉冲，如果使能了中断，主机和从机都会请求中断。当微控制器想要成为总线主机时，硬件要等到总线空闲时才进入主模式，这样就不会中断可能的从机操作。如果总线仲裁在主机下失败，I²C端口会立即切换到从机模式，并能在串行传输中检测自己的从机地址。

为控制每种模式下的I2C总线传输，用户需要根据当前状态码I²C _STATUS0寄存器来设置I2C_CTL0、

I2C_DAT寄存器。换句话说，对于每个I2C总线操作，用户需要通过I2C_STATUS0寄存器检查当前状态，然后设置I2C_CTL0, I2C_DAT寄存器来执行总线操作。最后，使用I2C_STATUS0检查响应状态。

I2C_CTL0寄存器中的STA、STO和AA标志位用于在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。新动作完成后，在I2C_STATUS0寄存器中更新新的状态码，且I2C_CTL0寄存器的SI标志被设置。但I2C停止动作时不会设置SI标志。如果设置了I2C中断使能位INTEN (I2C_CTL0[7])，则在中断服务例程中将执行新状态码或软件。

图 6.11-9 根据当前I²C 状态来控制I²C总线显示当前 I2C 状态码为 0x08，然后设置I2C_DATA=SLA+W 和 (STA,STO,SI,AA) = (0,0,1,x) 来发送到I²C总线。如果总线上一个从机匹配地址并应答，I2C_STATUS0 将更新状态码0x18。

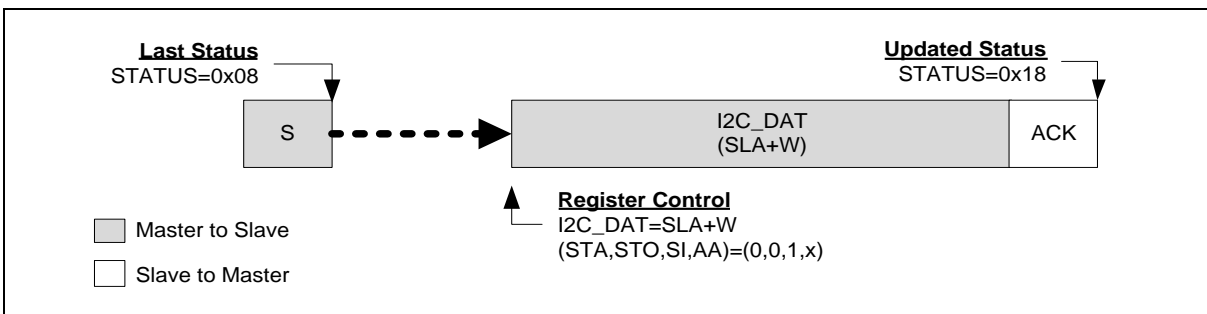


图 6.11-9 根据当前 I²C 状态来控制 I²C 总线

主机发送模式

主机发送模式，主机发送多个字节到从机。主机需要在I2CnCLK里设定时钟频率。主机发送模式需要将STA (I2CnCON.5) 置1。此时，一旦检测到总线空闲，主机硬件就会发出一个起始位START，若成功，SI(I2CnCON.3) 标志位将被置1，状态码I2CnSTAT置为08H。接下来应把从机地址和操作码“写”(SLA+W)写入I2CnDAT，然后清0 SI标志位，总线上发出SLA+W。

主机发出SLA+W 收到从机应答位ACK后，SI被置1，状态码I2CnSTAT = 18H。接下来将按照用户定义格式发送数据。所有的数据发送完以后，STO(I2CnCON.4)标志位置1，并清除SI位以发出停止信号STOP，或者也可以发送重复起始信号repeat START，而不发送STOP，直接开始新一轮数据传输。

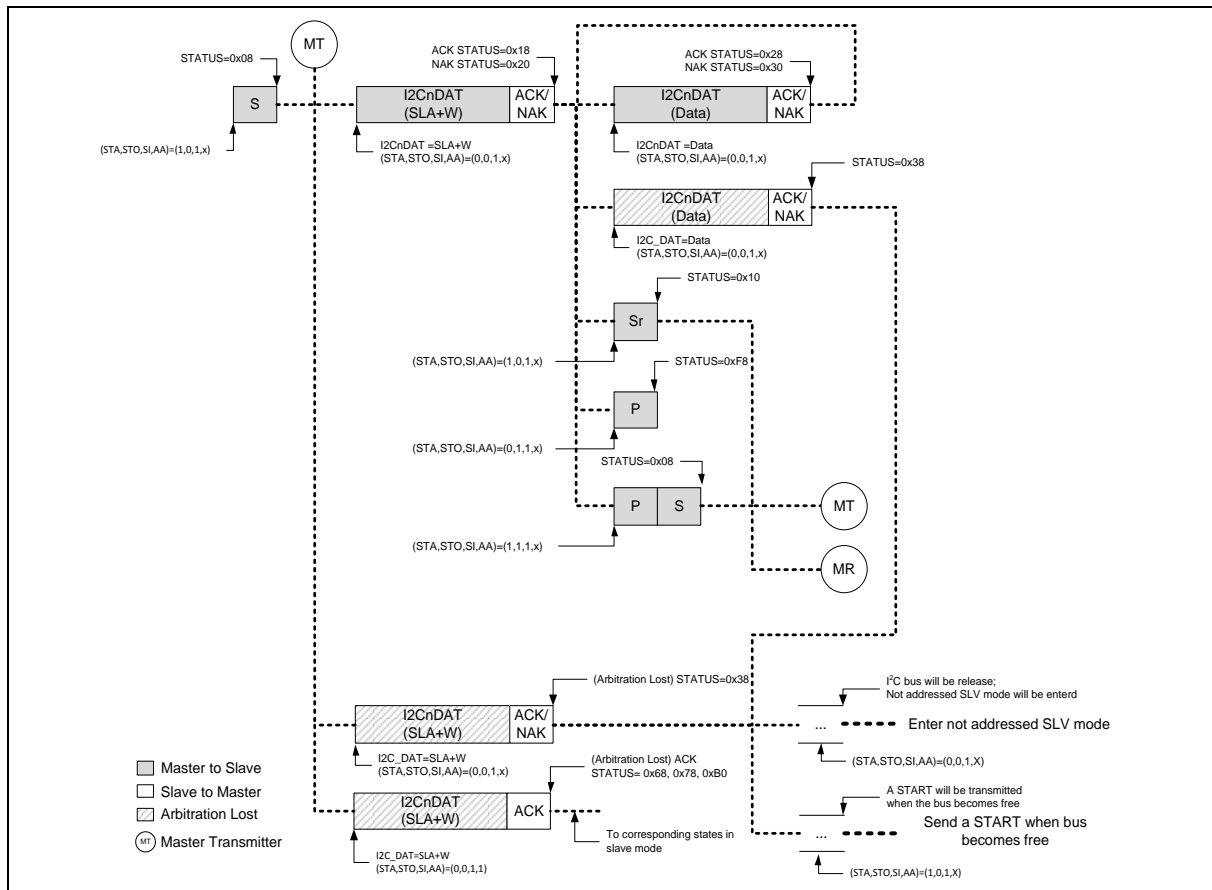


图 6.11-10 主机发送模式流程和寄存器状态

主机接收模式

主机接收模式，主机接收由从机传输的数据。初始化设置与主机发送模式相同，主机发送起始位以后，I2CnDAT应写入从机地址和“读”（SLA+只读）。收到从机应答位ACK后，SI 被置1且状态码I2CnSTAT = 40H。SI 清0后开始接收从机数据，若 AA 位 (I2CnCON.2) =1，主机收到数据后回应答位；若 AA =0 主机收到数据后不回应答，从机释放从机为一个未定址从机。然后主机可以发停止位或重起始开始新一轮传输。

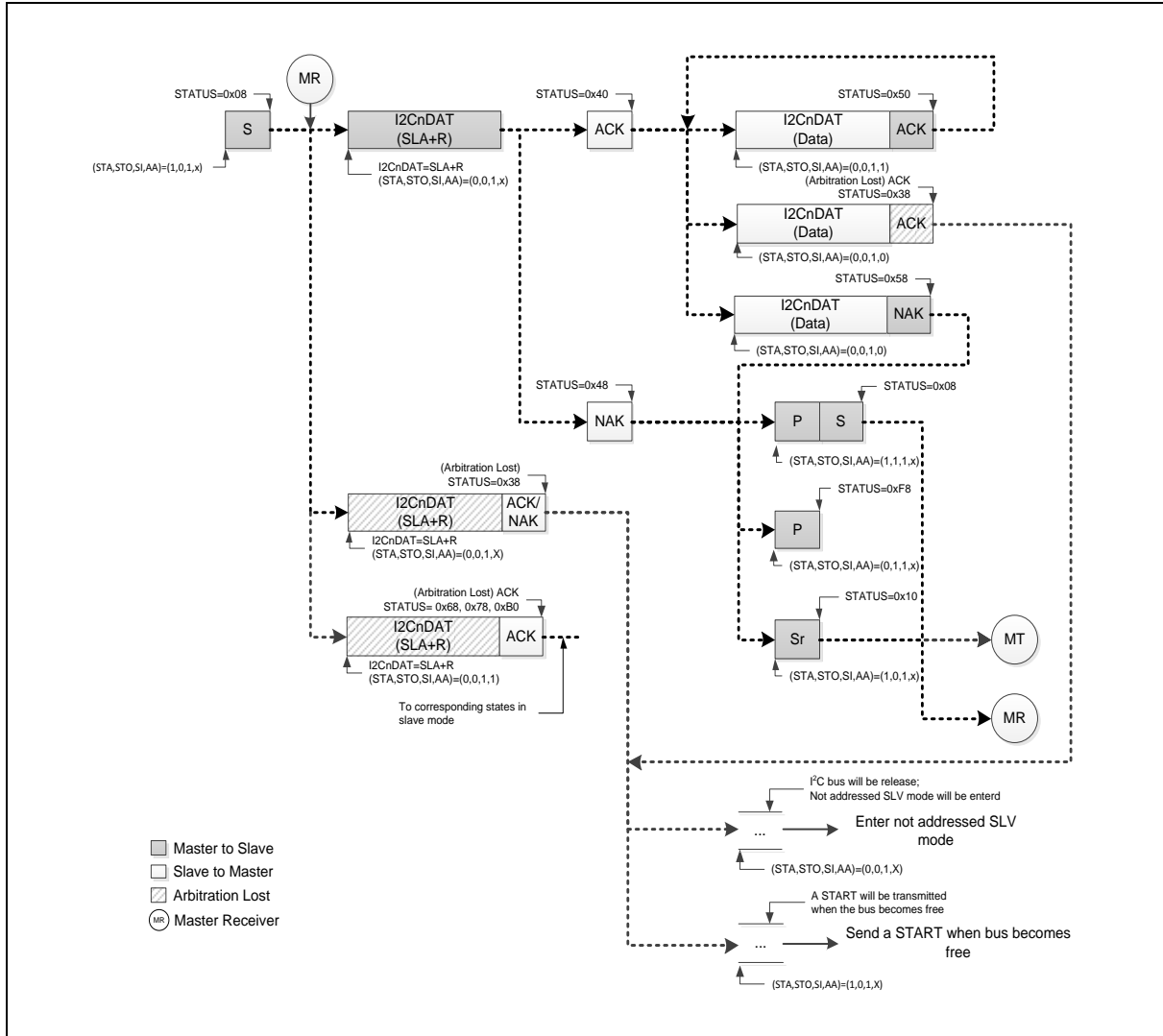


图 6.11-11 主机接收模式流程和寄存器状态

从机接收

在从机接收模式下，从机接收主机发来的数据。在传输开始前，I2CnADDRx应写入从机地址，I2CnCLK内容无效，AA置1用以响应主机的定址。上述初始化后，从机进入空闲模式，除非通过自己的地址和“写”信号（SLA+W）被定址。若主机仲裁失败，也会直接进入从机接收模式。

当从机被“写”信号SLA+W寻址到后，需要清0 SI位，以便从主机接收数据。如果在传输过程中AA=0，从机将在下一字节返回无应答位NACK，从机也将转为未定址从机，与主机通讯终止，不再接收数据，且I2CnDAT保持之前接收到的数据。

从机发送

I²C 接口有四个从机地址寄存器I2CnADDRx (x=0~3)。主机模式下，这些寄存器里的内容是无效的。从机发送模式，数据由从机发送给主机。当初始化I2CnADDRx及I2CnCON值后，设备等待直到自身地址被“读”信号(SLA+只读)定址。若主机仲裁失败，也可进入从机发送模式。

当从机被“读”信号SLA+R定址，需要将SI信号清0用以向主机发送数据。通常主机接收每字节数据后会返回应答位，如果没有接收到应答位，接下去的传输中，从机将发送全1数据。并变为未定址从机。如果传输过程中AA清0，从机将发送最后一个字节数据，并在接下去的传输中发送全1数据，并将自身变为未定址从机。

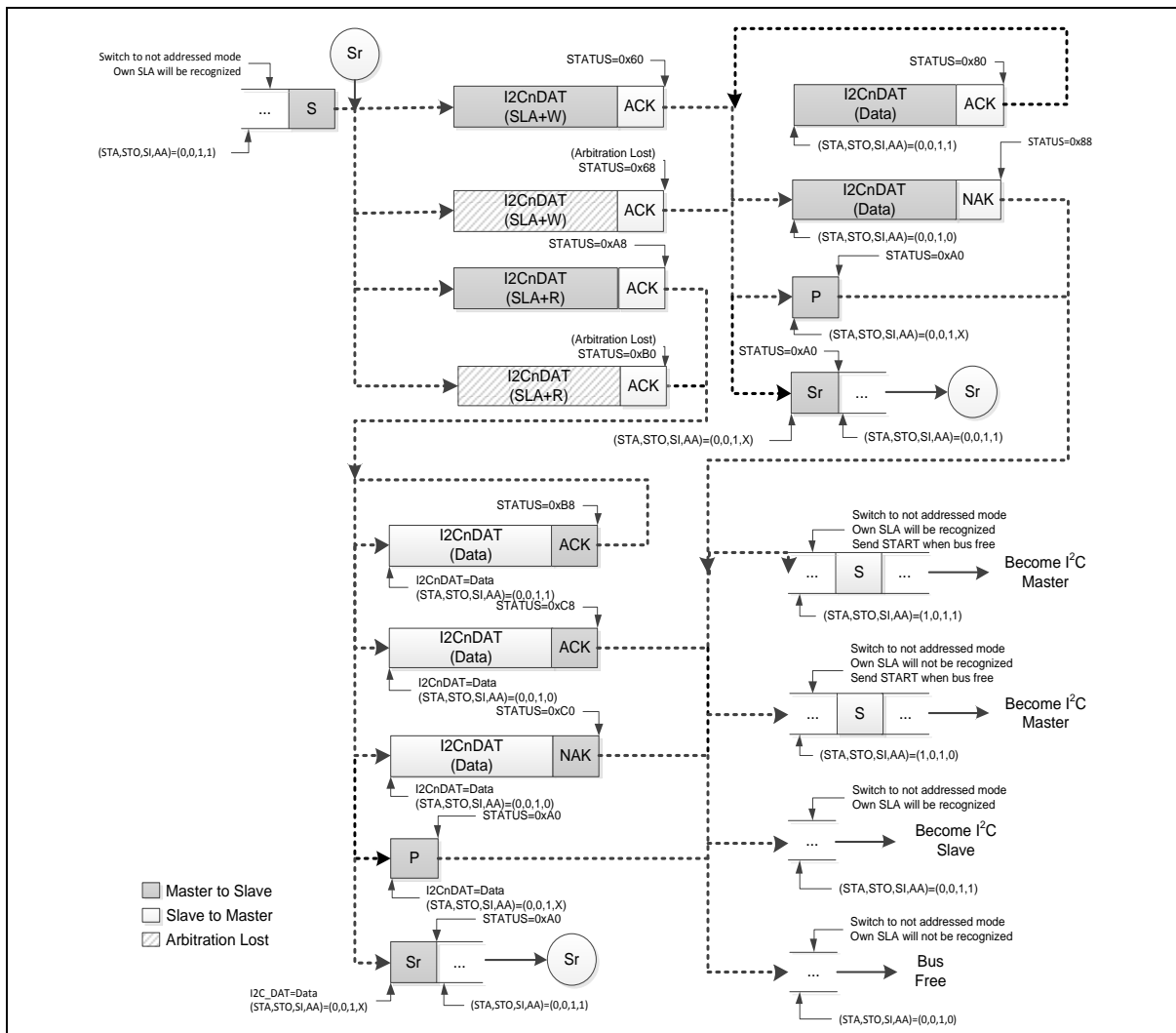


图 6.11-12 从机接收模式流程和状态

广播模式

广播呼叫模式是一种特殊的从机接收模式，定址为0x00，从机地址和读写都为0。当GC (I2CADDR.0)位及AA位都置1时，使能接收广播呼叫模式。在该模式下I2STAT值与普通从机接收模式I2CnSTAT值不同。仲裁失败也可能进入广播呼叫模式。

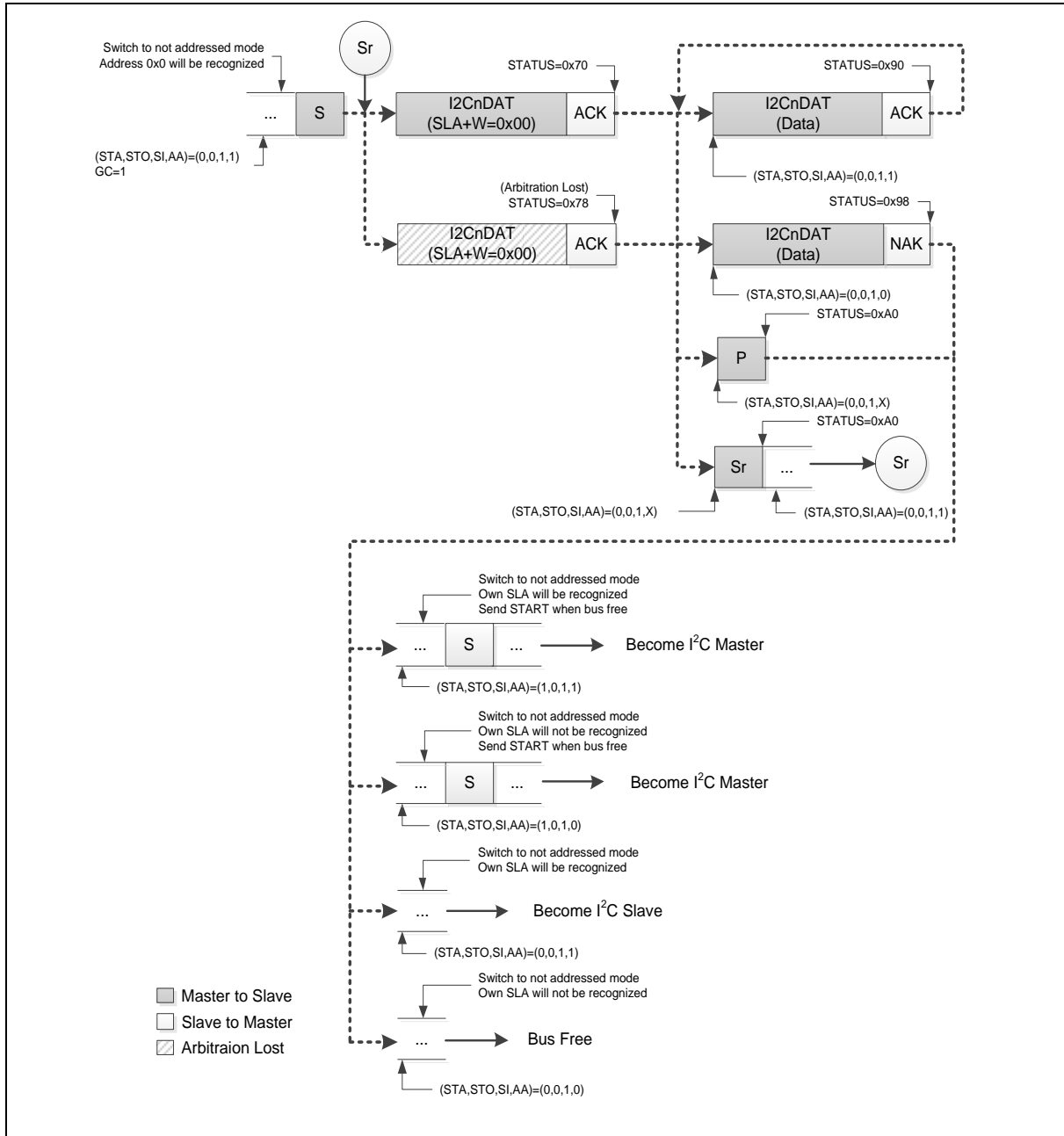


图 6.11-13 广播呼叫模式流程和状态

6.11.2.6 状态码

I2CSTA状态寄存器中,有两种状态码不归属25种之前所述状态，第一个状态码是F8H 用以标示在每次的传输中没有有效信息，意味着，SI标志位为0且无I²C 中断产生。另一个状态码00H 标示在传输过程中，

有总线错误发生。总线错误是指START或STOP在传输的过程中出现在错误的位置，例如在地址或数据的第二位或应答位。当总线发生错误，SI标志立即被置1，工作中的节点设备立即切换到未被定址从机模式，释放I2C0_SDA和I2C0_SCL，并将I2CnSTAT寄存器清0。要恢复总线状态，需要置位STO位并清除SI位，然后STO会由硬件清0，而不需要真正的STOP信号波形，释放总线恢复到正常空闲状态。

有一种特殊情况，当I2C0_SDA线被强制拉低导致阻塞，START信号或重复起始(repeat START)信号无法成功产生时，从机将失去同步。解决方法是在I2C0_SCL线上额外多送一个时钟脉冲。通过将STA位置1，总线上产生额外的时钟脉冲，由于SDA始终拉低，I2C0_SDA线上不会产生START信号。一旦I2C0_SDA线被释放，正常的START信号送出，状态寄存器上会显示08H，串行传输继续。相同的状况，如果需要发送重复开始 (repeated START)信号受阻，也可以采用上述方式。在上述方式完成传输后，状态寄存器显示10H，而不是显示08H。注：软件无法介入上述总线问题传输。

下表为I2C的I2STAT寄存器状态值和描述：

主机模式		从机模式	
状态	描述	状态	描述
0x08	开始信号	0Xa0	从机发送重复开始或停止信号
0x10	主机重复开始信号	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	主机发送地址无应答	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	从机接收地址无应答
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	广播呼叫模式数应答
		0x98	广播呼叫模式数据无应答
0xF8	总线释放 注意：“0xF8”退出主机/从机模式，不会触发中断		

6.11.3 I²C 中断服务程序典型结构范例

下例为在Keil™ C51编译器中的C程序范例，是I²C中断服务程序范例，包含26中状态子程处理。用户可根据实际应用选取所需状态处理。删除状态时，请确认该状态不会产生。

```

Void I2C_ISR (void) interrupt 6
{
    switch (I2STAT)
    {
        //=====
        //Bus Error, always put in ISR for noise handling
        //=====
        case 0x00: /*00H, bus error occurs*/
    
```

```

        STO = 1; //recover from bus error
        break;
//=====
//Master Mode
//=====
case 0x08: /*08H, a START transmitted*/
software STA = 0; //STA bit should be cleared by
        I2DAT = SLA_ADDR1; //load SLA+W/R
        break;
case 0x10: /*10H, a repeated START transmitted*/
        STA = 0;
        I2DAT = SLA_ADDR2;
        break;
//=====
//Master Transmitter Mode
//=====
case 0x18: /*18H, SLA+W transmitted, ACK received*/
        I2DAT = NEXT_SEND_DATA1; //load DATA
        break;
case 0x20: /*20H, SLA+W transmitted, NACK received*/
General Call STO = 1; //transmit STOP
        AA = 1; //ready for ACK own SLA+W/R or
        break;
case 0x28: /*28H, DATA transmitted, ACK received*/
        if (Conti_TX_Data) //if continuing to send DATA
        I2DAT = NEXT_SEND_DATA2;
        else //if no DATA to be sent
        {
            STO = 1;
            AA = 1;
        }
        break;
case 0x30: /*30H, DATA transmitted, NACK received*/
        STO = 1;
        AA = 1;
        break;
//=====
//Master Mode
//=====
case 0x38: /*38H, arbitration lost*/
        STA = 1; //retry to transmit START if bus free
        break;
//=====
//Master Receiver Mode
//=====
case 0x40: /*40H, SLA+R transmitted, ACK received*/
        AA = 1; //ACK next received DATA
        break;
case 0x48: /*48H, SLA+R transmitted, NACK received*/
        STO = 1;
        AA = 1;
        break;
case 0x50: /*50H, DATA received, ACK transmitted*/
        DATA_RECEIVED1 = I2DAT; //store received DATA
        if (To_RX_Last_Data1) //if last DATA will be received
            AA = 0; //not ACK next received DATA
        else //if continuing receiving DATA
            AA = 1;
        break;
case 0x58: /*58H, DATA received, NACK transmitted*/
        DATA_RECEIVED_LAST1 = I2DAT;
        STO = 1;

```

```

        AA = 1;
        break;
//=====
//Slave Receiver and General Call Mode
//=====
case 0x60:                                /*60H, own SLA+W received, ACK returned*/
        AA = 1;
        break;
case 0x68:                                /*68H, arbitration lost in SLA+W/R
        own SLA+W received, ACK returned */
        AA = 0;                            //not ACK next received DATA after
        STA = 1;                            //arbitration lost
        break;                            //retry to transmit START if bus free
case 0x70:                                /*70H, General Call received, ACK
returned*/
        AA = 1;
        break;
case 0x78:                                /*78H, arbitration lost in SLA+W/R
        General Call received, ACK
returned*/
        AA = 0;
        STA = 1;
        break;
case 0x80:                                /*80H, previous own SLA+W, DATA received,
        ACK returned*/
        DATA_RECEIVED2 = I2DAT;
        if (To_RX_Last_Data2)
            AA = 0;
        else
            AA = 1;
        break;
case 0x88:                                /*88H, previous own SLA+W, DATA received,
        NACK returned, not addressed SLAVE
mode
        entered*/
        DATA_RECEIVED_LAST2 = I2DAT;
        AA = 1;                            //wait for ACK next Master addressing
        break;
case 0x90:                                /*90H, previous General Call, DATA
received,
        ACK returned*/
        DATA_RECEIVED3 = I2DAT;
        if (To_RX_Last_Data3)
            AA = 0;
        else
            AA = 1;
        break;
case 0x98:                                /*98H, previous General Call, DATA
received,
        NACK returned, not addressed SLAVE
mode
        entered*/
        DATA_RECEIVED_LAST3 = I2DAT;
        AA = 1;
        break;
//=====
//Slave Mode
//=====
case 0xA0:                                /*A0H, STOP or repeated START received
while
        still addressed SLAVE mode*/
        AA = 1;
        break;

```

```

//=====
//Slave Transmitter Mode
//=====
case 0Xa8: /*A8H, own SLA+R received, ACK returned*/
    I2DAT = NEXT_SEND_DATA3;
    AA = 1; //when AA is "1", not last data to be
           //transmitted
    break;
case 0Xb0: /*B0H, arbitration lost in SLA+W/R
           own SLA+R received, ACK returned */
    I2DAT = DUMMY_DATA;
    AA = 0; //when AA is "0", last data to be
           //transmitted
    STA = 1; //retry to transmit START if bus free
    break;
case 0Xb8: /*B8H, previous own SLA+R, DATA
transmitted,
           ACK received*/
    I2DAT = NEXT_SEND_DATA4;
    if (To_TX_Last_Data) //if last DATA will be transmitted
        AA = 0;
    else
        AA = 1;
    break;
case 0Xc0: /*C0H, previous own SLA+R, DATA
transmitted,
           NACK received, not addressed SLAVE
mode
           entered*/
    AA = 1;
    break;
case 0Xc8: /*C8H, previous own SLA+R, last DATA trans-
SLAVE
           mitted, ACK received, not addressed
           mode entered*/
    AA = 1;
    break;
} //end of switch (I2STAT)

SI = 0; //SI should be the last command of
I2C ISR
while(STO); //wait for STOP transmitted or bus
error //free, STO is cleared by hardware
}

```

6.11.4 I²C 超时

MS51带一组14位超时计数器，用于防止I²C总线故障。一旦使能了超时定时器，计数器开始计数直至溢出，即如果开启中断，I2TOF位会被硬件置1。当使能计数器，SI置1复位计数器，SI清0后重新开始计数。若I²C总线出现故障，SI位长时间不能清0，14位的超时定时器就会溢出，并进入中断。

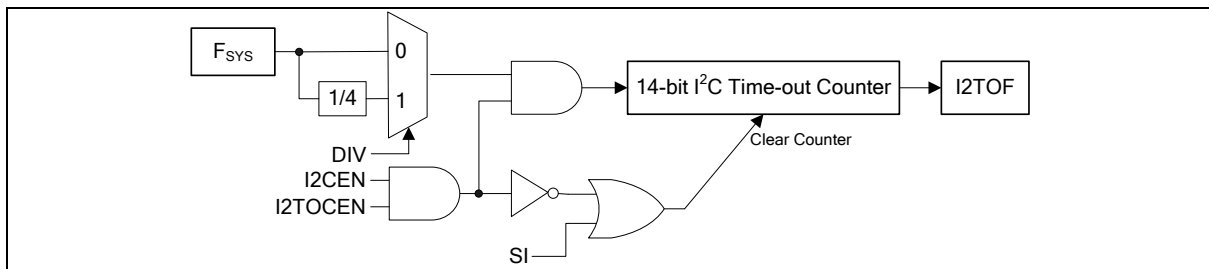


图 6.11-14 I²C 超时计数器

6.11.5 I²C 中断

I²C的两个标志位：SI 和 I2TOF。这两个标志都会产生I2C事件中断请求。如果 EI²C = 1且 EA = 1,当两个标志中的任意一个发生时，CPU就会去执行中断代码。用户可以读取这两个标志位，来确定中断产生的原因。这两个标志需软件清0。

6.11.6 I²C控制寄存器

I²C 总线接口包含5个控制寄存器，I2CON、I2STAT、I2DAT、I2ADDR和I2CLK。这些寄存器提供总线控制、状态、数据传输和接收功能。为了应用的灵活性，I2C0_SDA和I2C0_SCL管脚可以由I2CPX (I2CON.0)来配置。下面是寄存器的说明。

I2CON – I²C 控制

寄存器	SFR 地址	复位值
I2CON	C0H, 所有页, 位寻址	0000_0000 b

7	6	5	4	3	2	1	0
I	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	I	I2C0 保持时间延长使能 0 = 禁止I ² C DATA 到 I2C0_SCL的 保持时间延长 1 = 使能I ² C DATA 到 I2C0_SCL 的保持时间延长, 延长8个系统时钟
6	I2CEN	I ² C 总线使能 0 = I ² C 总线禁止. 1 = I ² C 总线使能. 在使能 I ² C之前, I2C0_SCL 和 I2C0_SDA 端口电平需要为高
5	STA	起始标志START 当STA置1, 如果总线空闲, I ² C产生START信号, 如果总线忙, I ² C等待停止条件STOP, 然后产生START信号 如果总线已经在总机模式且已发送一个或多个字节, 此时再设定STA, I ² C 总线将产生重复开始信号 repeated START 注: STA可在任何时间置1, 包括从机模式。但硬件不会在发送START或repeat START信号后自动清0。用户需软件清除。

位	名称	描述
4	STO	<p>停止标志STOP</p> <p>I²C 总线在主机模式下设定STO为1, 将会向总线发送停止信号STOP。一旦总线上停止条件完成, STO由硬件自动清0。</p> <p>当总线上产生错误状态(I2CnSTAT为 00H) STO 也会置1。这种情况下总线不会发送停止位。</p> <p>如果STA和STO同时置1, 且在主机模式下, I²C总线在发送STOP后马上发送START。如果在从机模式下, 应避免STA及STO同时置1, 以避免发出非法I²C帧</p>
3	SI	<p>I²C中断标志</p> <p>I²C所有26种状态中出现一种, 硬件就会置1此位 (F8H 除外), 此时软件根据读取I2STAT值, 来确认哪步已经执行和决定下一步动作。</p> <p>SI由软件清0。在SI被清0之前, I2C0_SCL低电平周期延长, 传输暂停, 该状态对于从机处理接收到的数据非常有用, 可以确保准确处理前一数据再接收下一个数据。</p> <p>SI位被软件清0后, I²C外设才会继续下一步: SI清除后, 总线将继续产生START和repeat START条件, STOP条件, 8位数据, 或者其他控制字节或位。因此, 在SI软件清除前, 软件应该准备好合适的寄存器设置。</p>
2	AA	<p>应答应标志</p> <p>若AA = 1, 当I2C器件是一个接收器或者是个地址匹配的从机时, 在I2C0_SCL上的应答时钟脉冲期间, I2C0_SDA将应答ACK (I2C0_SDA低电平)</p> <p>若AA = 0, 当I2C器件是一个接收器或者是个地址匹配的从机时, 在I2C0_SCL上的应答时钟脉冲期间, I2C0_SDA将应答NACK (I2C0_SDA高电平)。若器件自我清除AA标志位, 则会清除其从机地址或忽略广播呼叫, SI不会置位, 中断不予产生。</p> <p>注: 若已被寻址的从机, 在从机接收模式下未回复应答位或在从机发送模式下未接收到应答位, 该从机将变为未寻址从机, 无法接收数据直到其AA被置1, 且重新被主机定址。</p> <p>特殊情况注意: 从机发送模式时, I2CnSTAT若为C8H, 从机发送最后一个字节给主机之前, 清除AA, 发送完最后一个字节的位后, 从机将变成未被定址的从机模式, 和主机断开。主机若再从总线上读数据, 将得到FFH。</p>
1	-	保留
0	I2CPX	<p>I²C 管脚选择</p> <p>0 = 设置 I2C0_SCL 在 P1.3 且 I2C0_SDA 在 P1.4.</p> <p>1 = 设置 I2C0_SCL 在 P0.2 且 I2C0_SDA 在 P1.6.</p> <p>注意配置此位后 I²C 管脚会立刻改变</p>

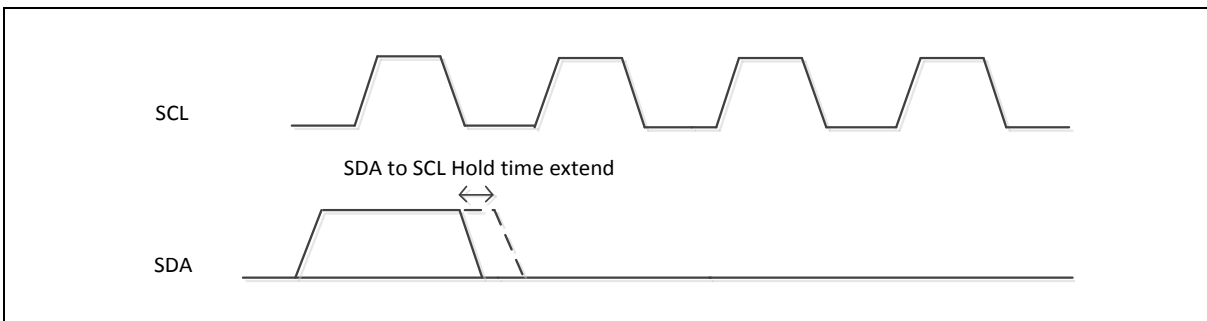


图 6.11-15 保持时间延长使能

I2STAT – I²C 状态

寄存器	SFR 地址	复位值
I2STAT	BDH, 页 0	1111 1000 b

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

位	名称	描述
7:3	I2STAT[7:3]	I ² C 状态码 I2CnSTAT高5位为状态码，共有27种值。I2CnSTAT = F8H时，表示空闲，SI 将保持为0。其它26种状态，都会让SI置1，且产生中断请求。
2:0	0	保留 I2STAT 的低3位总是0

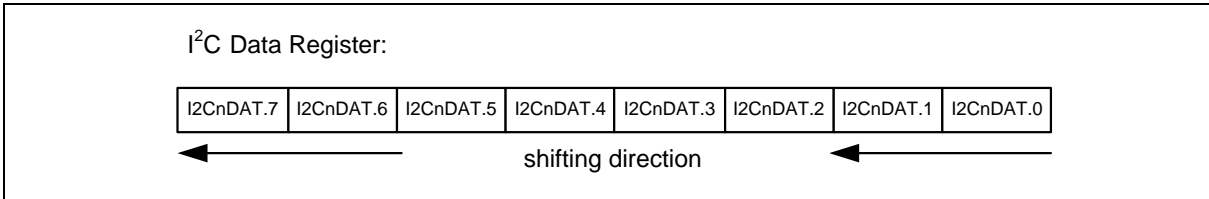
I2DAT – I²C 数据

寄存器	SFR 地址	复位值
I2DAT	BCH, 页 0	0000_0000 b

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

位	名称	描述
7:0	I2DAT[7:0]	<p>I²C 数据</p> <p>该寄存器存放准备发送的，或接收到的数据。只要 SI = 1，此数据就有效。在 I²C 数据传输过程中，I2CnDAT 读写的结果是不可预测的。</p> <p>当 I2CnDAT 移出时，总线上的数据同步移入更新 I2CnDAT。I2CnDAT 总是显示为从总线上取出的最后一个字节。所以总线仲裁失败时，I2CnDAT 原来的值在传输后已经改变。</p>

I²C 数据移位方向.



I2ADDR – I²C 本机从机地址

寄存器	SFR 地址	复位值
I2ADDR	C1H, 页 0	0000_0000 b

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

位	名称	描述
7:1	I2ADDR[7:1]	<p>I²C 本机从机地址</p> <p>主机模式: 无效</p> <p>从机模式: 存放7位从机地址。主机需要定址该从机, 需通过在START或repeat START之后的第一个字节值地址信息与此地址相同。如果AA为1, 该从机响应主机, 成为被定址从机。否则主机呼叫地址会被忽略。</p> <p>注: I2CnADDRx [7:1] 不能写为全0, 因为0x00为广播呼叫方式寻址专用。.</p>
0	GC	<p>广播呼叫位</p> <p>主机模式: 无效</p> <p>从机模式: 0 = 广播呼叫模式忽略, 不响应。 1 = 如果AA置1, 参与广播呼叫模式, 若AA清0, 忽略广播呼叫。</p>

I2CLK – I²C 时钟

寄存器	SFR 地址	复位值
I2CLK	BEH, 页 0	0000_1001 b

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

位	名称	描述
7:0	I2CLK[7:0]	<p>I²C 时钟设定</p> <p><u>主机模式:</u> 该寄存器设定作主机时 I²C 总线时钟速率。算式如下:</p> $\frac{F_{SYS}}{4 \times (I2CLK + 1)}$ <p>注 I2CCLK 值写入 00H 及 01H 无效, 实行了限制。</p> <p><u>从机模式:</u> 该字节无效, 从机自动同步主机时钟, 最高 400kbps.</p>

I2TOC – I²C 超时计数器

寄存器	SFR 地址	复位值
I2TOC	BFH, 页 0	0000_0000 b

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

位	名称	描述
2	I2TOCEN	I ² C 超时计数器使能 0 = I ² C 超时计数器禁止. 1 = I ² C 超时计数器使能. 注意: 使能 I ² C 超时计数器功能后请使能 I ² C 中断
1	DIV	I ² C 超时计数器时钟除频 0 = I ² C 超时计数器时钟源是 F _{sys} /1. 1 = I ² C 超时计数器时钟源是 F _{sys} /4.
0	I2TOF	I ² C 超时标志 T 这个标志当 14-bit I ² C 超时计数器溢出时由硬件置1, 由软件清0

6.12 串行外围总线 (SPI)

6.12.1 概述

MS51系列提供支持高速串行通信的SPI模块。SPI为微控制与外设如：EEPROM、LCD 驱动、D/A 转换之间提供全双工、高速、同步传输的总线。可提供主机从机模式传输，速度可达到时钟频率 $F_{sys}/2$ ，支持传输完成标志位和“写”冲突标志位。在多主机系统中，SPI支持主机模式错误用以防止主机冲突。

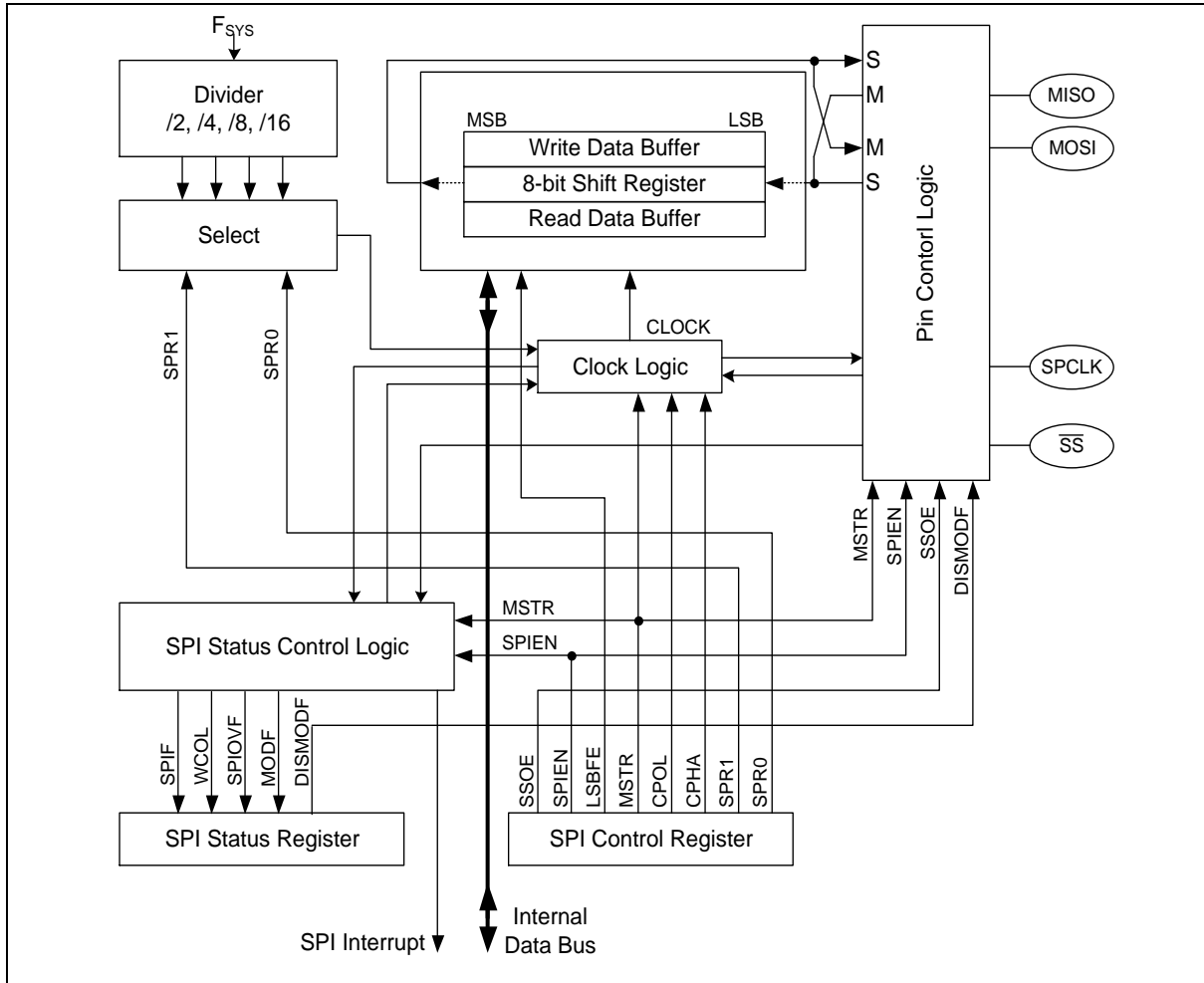


图 6.12-1 SPI 框图

图 6.12-1 SPI 框图展示了SPI的结构框图。SPI的主要模块有SPI控制寄存器、SPI状态逻辑、时钟率控制和管脚逻辑控制。为了传送数据和接收数据，SPI 提供了移位寄存器和读数据缓冲器。无论传送或接收数据都是双缓存。在传输数据时，前一个数据发送完成前，不可以写入下一个数据到移位寄存器。接收端有并行读数据缓存，移位寄存器接收第二个数据的同时，前一个接收的数据将被传送到读数据缓存。

SPI 接口有四个管脚，分别是主进/从出(MISO)、主出/从进(MOSI)、移位时钟(SPCLK)和从机选择(\overline{SS})。MOSI脚用于传输主机到从机的8位数据，所以MOSI是一个主机的输出、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引脚为主机模式下的时钟输出，从机的输入时钟。移位时钟用于MOSI和MISO脚之间数据传输

的同步时钟。主机模式发送8个移位时钟，总线上可以完成1个字节的数据交换。位移时钟由主机输出，所以一组SPI从机设备通过设定从机选择脚 (\overline{SS})选择。当需要访问从机时，该从机的片选脚必须保持低。当 \overline{SS} 为高，该从机访问将被禁止。若为多从机模式，在同一时刻必须保持只有一个从机被选定。对于主机， \overline{SS} 脚不做任何用途，可配置为普通端口另做他用。 \overline{SS} 可用于多主机模式下主机模式错误侦测功能，如存在多主机环境通过软件设置。MS51也提供自动激活片选脚功能，通过自动触发 \overline{SS} 来传输字节。

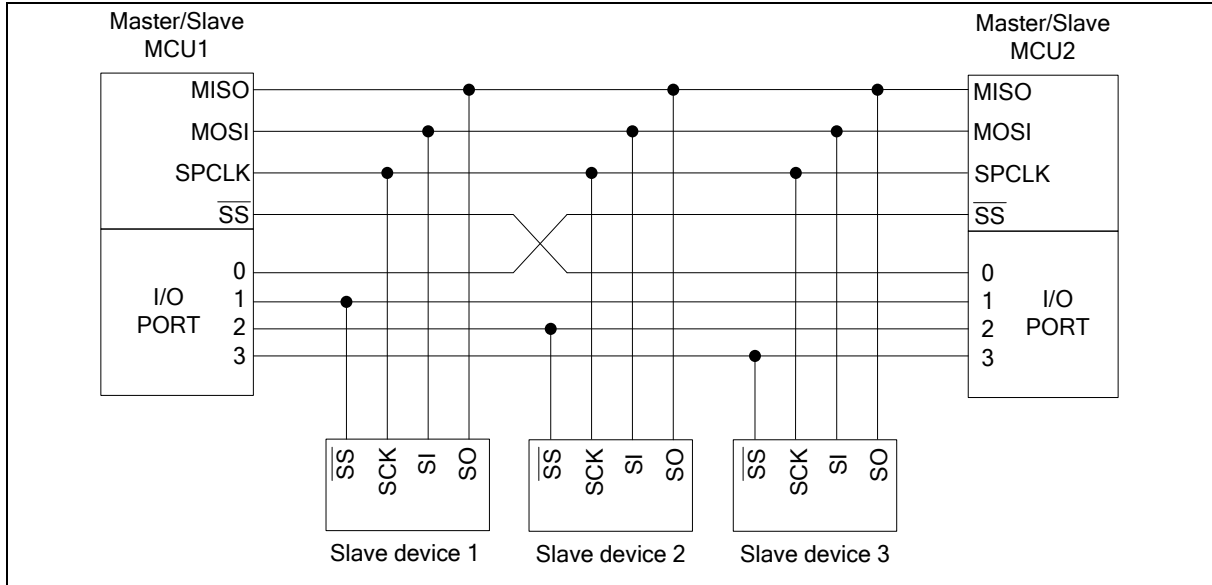


图 6.12-2 SPI 多主机，多从机连接图

图 6.12-2 SPI多主机，多从机连接图。通信总线通过3根信号线相连，MOSI ~ MOSI, MISO ~ MISO, 和 SPCLK ~ SPCLK。主机通过一个并口的4个管脚来控制4个 \overline{SS} 脚，从而实现每个 \overline{SS} 线分别控制每个从机。MCU1和MCU2可以定义为主机或从机模式。 \overline{SS} 需配置为主机模式侦测功能避免多主机冲突。

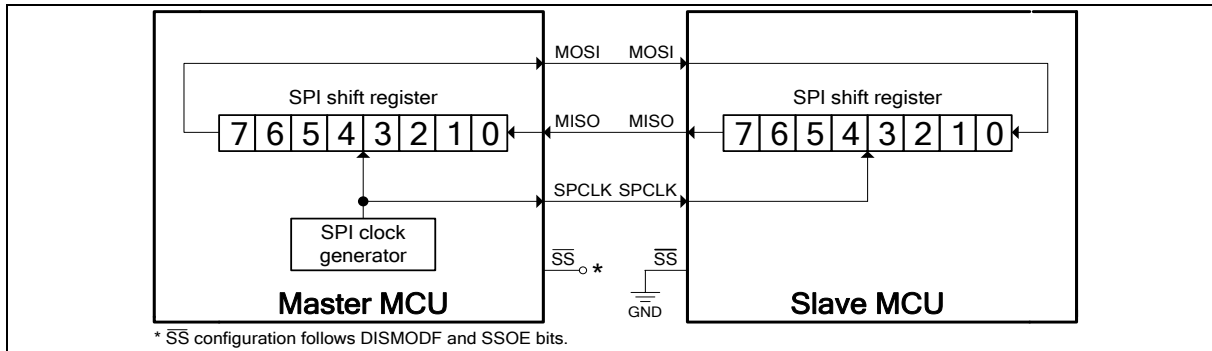


图 6.12-3 SPI 单主机单从机连接图

图 6.12-3 SPI 单主机单从机连接图表示SPI模块单主机/从机互连简图。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。这样通过两MCU的SPI移位寄存器，就完成了交换数据。

默认情况下，SPI先发送MSB。当LSBFE (SPIInCR.5) 置1，SPI首先发送LSB，该位不会影响寄存器内MSB/LSB的排列顺序。注：下述全部是基于LSBFE为0的情况，MSB 首先被发送和接收。

控制寄存器 (SPIInCR)、SPI 状态寄存器 (SPIInSR)、SPI 数据寄存器 (SPIInDR) 这三个寄存器用于SPI传

输。这些寄存器提供控制、状态检测、数据存储以及时钟设置。下面描述SPI寄存器的相关功能。

6.12.2 操作模式

6.12.2.1 主机模式

MSTR (SPInCR.4)位置1, SPI工作在主机模式。整个SPI系统中只允许一个主机启动传输。每次传输总是由主机发起, 对主机SPInDR寄存器的写开始传送。在SPCLK控制下在MOSI管脚传送数据。同时, MISO管脚接收数据。在8位数据传输完毕后, SPIF (SPInSR.7)由硬件自动置位以示完成一个字节数据传输。同时接收到的数据也会传送到SPInDR。用户可以从SPInDR读出数据, 并清除SPIF。

6.12.2.2 从机模式

设定MSTR为0, SPI将工作在从机模式。当作为从机模式时, SPCLK管脚变为输入脚, 它将作为时钟输入被另外一个主机SPI设备的输出时钟控制, \overline{SS} 管脚也变为输入脚。当从机设备的 \overline{SS} 管脚不为低时, 主机设备不能与从机交换数据。在数据传输开始前和数据传输完成前, \overline{SS} 管脚都需要保持低电平状态。如果 \overline{SS} 变为高电平, SPI将被迫进入闲置状态。如果 \overline{SS} 管脚在传输的过程被置高, 那么传输将被取消, 接收移位缓存区里剩下的位数将变高, 同时也将进入闲置状态。

在从机模式下, 数据通过MOSI管脚从主机向从机传输, 通过MISO管脚从从机向主机传输。通过主机SPCLK的时钟控制, 数据进入移位寄存器。在移位寄存器接收到一个字节后, 数据将移到读数据缓存, 同时SPIF置1。对SPInDR的读操作实际上就是对读缓冲器的一次读操作。为了防止缓冲器溢出或因溢出导致数据丢失, 从设备必须在数据第二次从移位寄存器向读缓冲器传送前, 把数据从SPInDR读出和把SPIF清零。

6.12.3 时钟格式和数据传输

为了适应各种同步串行外设, SPI提供时钟极性位CPOL (SPInCR.3)和时钟相位位CPHA (SPInCR.2)寄存器用以控制。如图 6.12-4所示, CPOL和CPHA组合出四种不同的时钟格式。CPOL位表示空闲状态时SPCLK脚电平。CPHA位定义表示是MOSI和MISO上时钟的哪个边沿用来采样。在同一系统上的主从设备中, CPOL和CPHA的配置应该是相同的。传输不同的数据格式, 将产生随机错误结果。

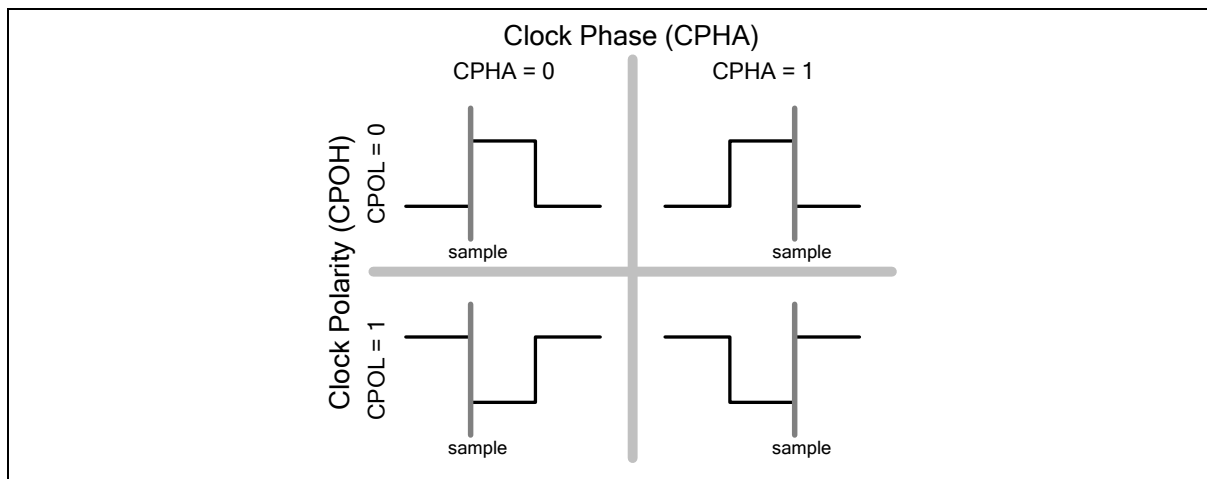


图 6.12-4 SPI 时钟格式

在SPI传输中, 总是由主机启动传输。如果SPI被选定作为主模机式 (MSTR = 1) 并且打开传输 (SPIEN =1), 对主机的SPI数据寄存器 (SPInDR) 写入数据将启动SPI时钟和数据传输。发送一个字节的同时会接收一个字节的数据, 此后SPI时钟停止, 主机和从机的SPIF (SPInSR.7) 同时被置1。如果SPI中断使能位ESPI设置为1, 全局中断使能 (EA= 1), 将执行SPI的 (ISR) 中断服务程序。

从机模式下，需要注意 \overline{SS} 信号。如图 6.12-5 SPI CPHA = 0时时钟和数据格式所示，CPHA=0时，SPCLK第一个边沿为MSB的采样点（LSBFE= 0，MSB优先发送为例）。因此，从机必须在SPCLK第一个采样边沿出现之前先把MSB传出。 \overline{SS} 的下降沿可用于准备MISO的MSB。因此，每次成功串行传输一个字节后， \overline{SS} 引脚必须切换先高然后低。此外，如果从机将数据写入SPI数据寄存器（SPInDR）时，如果 \overline{SS} 为低电位，则会发生写冲突错误。

当CPHA = 1，采样边沿位于SPCLK时钟的第二个边沿。从机使用的第一个SPCLK时钟转移的MSB，而不是 \overline{SS} 的下降沿。因此，在每次成功传输时 \overline{SS} 可以始终保持低电位保持低之间的转移。此格式更适合单主机单从机的结构使用。CPHA =1模式，从机的 \overline{SS} 可以不连接在SPI系统中，直接接地。

注：在SPI传输使能(SPIEN = 1)前，必须先对SPI传输进行配置，否则传输过程中对LSBFE、MSTR、CPOL、CPHA 及 SPR[1:0] 的任一更改，将会停止SPI传输并强迫总线进入空闲模式。所以在任何配置位更改前，请先关闭SPIEN使能位。

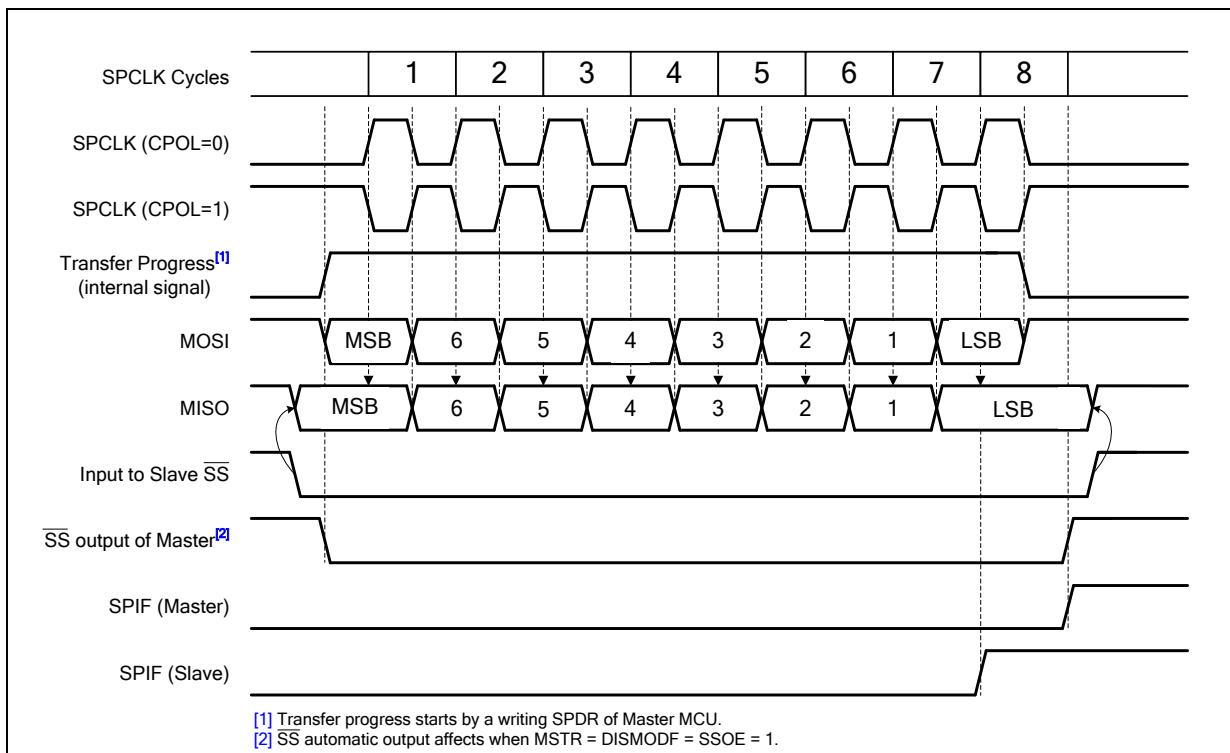


图 6.12-5 SPI CPHA = 0 时时钟和数据格式

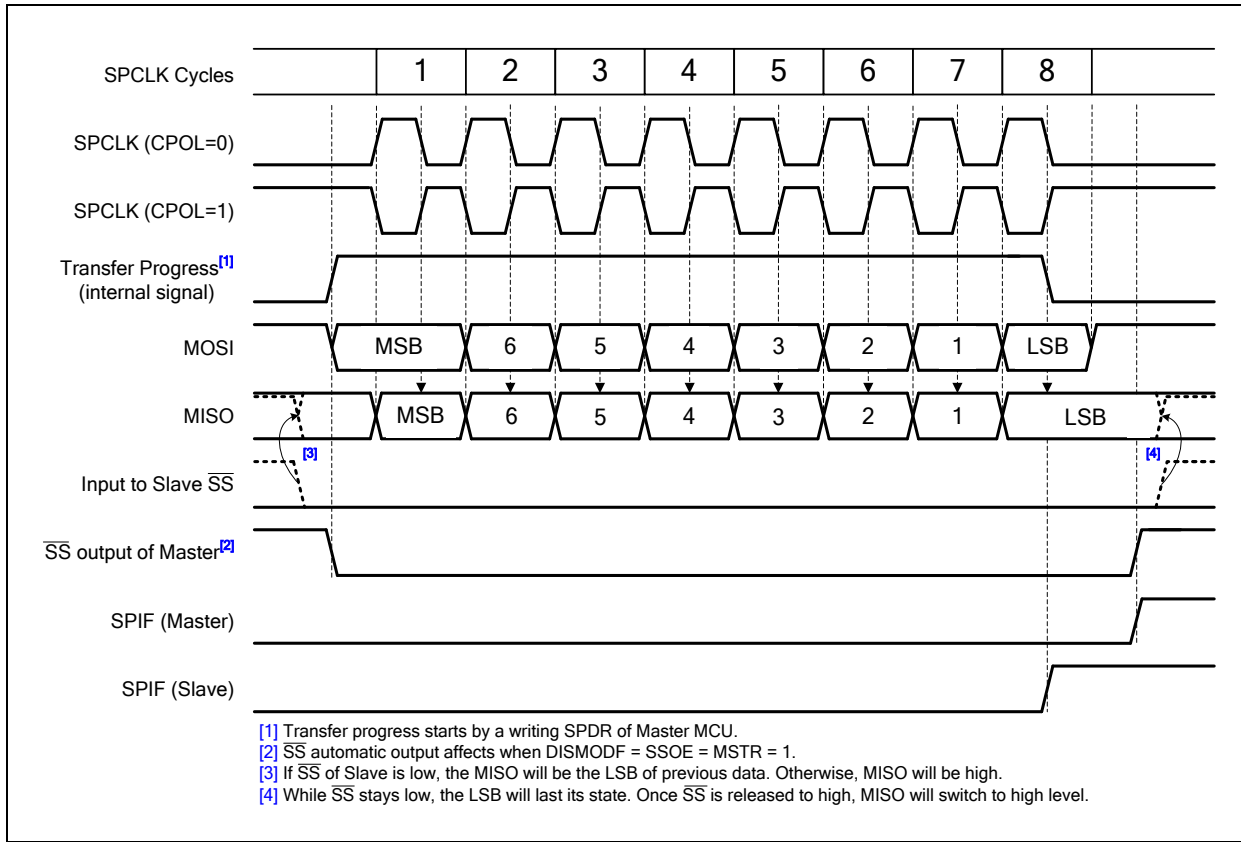


图 6.12-6 SPI CPHA = 1 时时钟和数据格式

6.12.4 片选管脚设置

MS51 SPI提供灵活的 \overline{SS} 配置用于不同系统。当作为从机时， \overline{SS} 始终定义为选择输入脚。当作为主机时， \overline{SS} 有三种不同的功能定义，可以通过DISMODF(SPIInSR.3)和SSOE(SPIInCR.7)来配置。默认情况DISMODF=0，故障侦测功能打开， \overline{SS} 配置为输入脚并检测是否发生故障。反之，如果DISMODF=1，故障侦测功能关闭，SSOE寄存器定义控制 \overline{SS} 管脚。当SSOE=1，从机选择信号自动生成，主机的 \overline{SS} 管脚直接与从机的 \overline{SS} 脚连接，当选择外部从机进行传输时 \overline{SS} 自动拉低，当进入闲置状态或者没有选择从机时，自动拉高。当SSOE=0且DISMODF=1时， \overline{SS} 不再用作SPI管脚，而完全配置为普通端口状态。

6.12.5 模式错误检测

在一个SPI网络中，当不止一个设备有可能成为主机时，为减少数据传输错误，模式故障侦测功能是非常有用的。当一个SPI设备配置为主机，通过SPIInCR0把 \overline{SS} 配置为打开模式故障侦测输入脚，发现 \overline{SS} 由其它设备拉低，发生模式故障发生。说明系统上有一个从机试图寻找主机地址并把主机认为从机。此时，硬件会自动将SPCR的MSTR和SPIEN清除，从而SPI功能关闭，并使能错误侦测标志MODF(SPIInSR.4)置1，如果之前已打开中断ESPI(EIE.6)和EA置1，则会进入中断。

6.12.6 写冲突错误

SPI在传输方向上缓冲信号，在接收和发射方向上是双缓冲。除非前一个数据传输完，否则新的数据不能写入移位寄存器。当正在进行一次传送时，如果设备同时又写数据到SPIInDR，将发生写冲突错误。发送数据时SPIInDR是一个双缓存，任何写入SPIInDR数据将直接写入SPI的移位寄存器。一旦发生一个写冲突错误，WCOL(SPIInSR.6)会被硬件置1指示发生一个写冲突。这种情况下，当前的传输继续不停

，然而引起写冲突的新数据将丢失。尽管SPI逻辑可以在主机和从机之间进行写冲突检测，但写冲突通常是一个从机错误，原因是当主机开始一次传送时，从机是无法预知。在从机接收过程中，写SPInDR也将产生写冲突错误。WCOL标志用软件清除。

6.12.7 覆盖错误

对于接收数据，SPI是双缓存。接收到的数据移入到一个并行读数据缓存中，同时能接收第二个数据。然而，在下一个数据移入之前，必须确保已从SPInDR中读取已接收数据。在下一个数据被移入前，要把前一个数据从读缓冲区内读出，并且清除SPIF，这样的才不会产生覆盖错误，反之，将产生覆盖错误。这种情况下，第二个字节的数据不会正常移入读数据缓存，缓存区内仍保留有前一个数据。当发生覆盖错误时，SPIOVF (SPInSR.5)会被硬件置1。如果中断使能，会进入SPIOVF中断请求。图 6.12-7 SPI 覆盖波形表示接收数据与移出错误之间的关系。

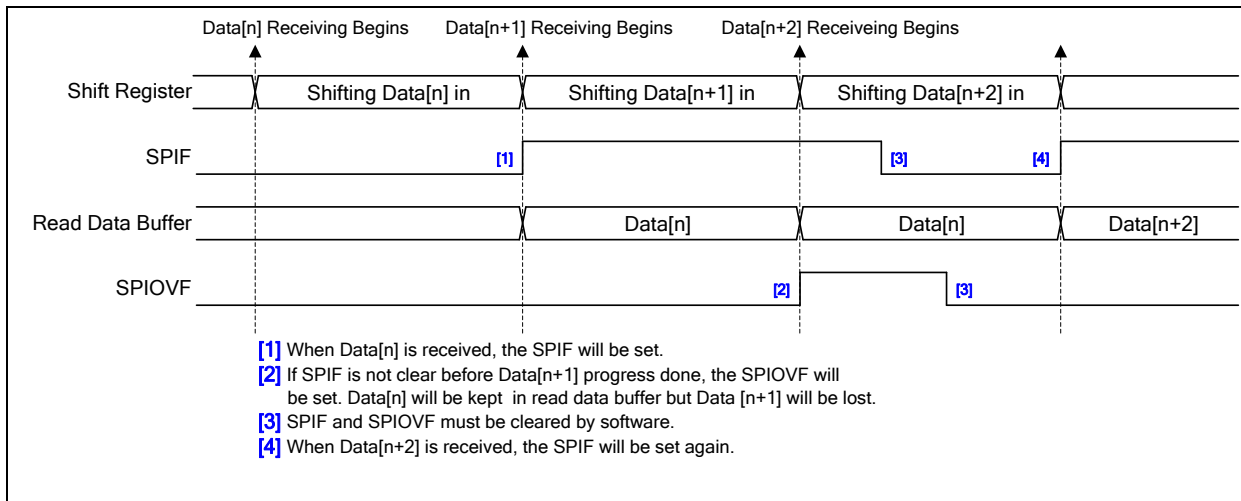


图 6.12-7 SPI 覆盖波形

6.12.8 SPI 中断

SPI三个中断状态标志包括SPIF、MODF 和 SPIOVF，用于产生SPI事件中断请求。这些位都放在SPInSR寄存器中。当有外部数据传入SPInDR或自身完成数据传输后，SPIF标志将被置位。MODF置1时，表示SS进入模式错误状态，SPIOVF表示接收发生数据溢出错误。当SPI中断打开时（ESPI 和EA置1），当这3个标志中的任意一个置1，CPU会执行SPI中断服务程序。用户若需要了解是由何种标志引起中断，必须检查相应的标志位。这三个标志必须由用户软件清除。

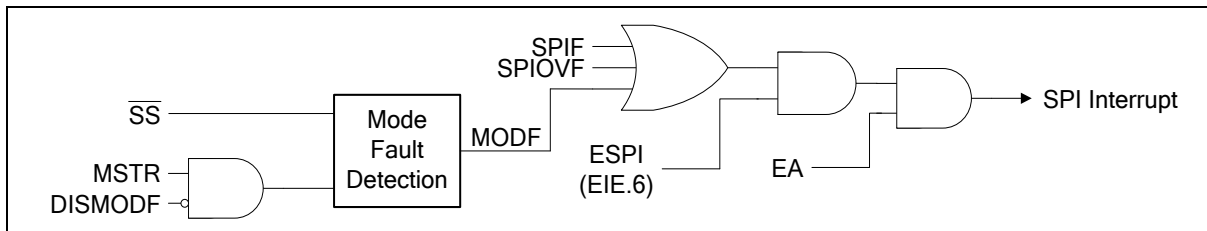


图 6.12-8 SPI 中断请求

6.12.9 SPI控制寄存器

SPCR – 串行外围控制寄存器

寄存器	SFR 地址	复位值
SPCR	F3H, 页 0	0000_0000 b

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

位	名称	描述
7	SOE	片选输出使能 该位搭配DISMODF (SPIInSR.3) 用于定义 \overline{SS} 管脚，如位仅在MSTR=1和DISMODF=1的条件下有效。 0 = \overline{SS} 作为普通 I/O。 1 = \overline{SS} 选择的外部从机设备，当数据传输时自动拉低，当总线进入空闲模式时自动变高
6	PIEN	SPI 使能 0 = 关闭SPI功能。 1 = 打开SPI功能。
5	LSBFE	LSB 优先使能 0 = SPI优先传输最高位MSB数据 1 = SPI优先传输最低位LSB数据
4	STR	使能主机模式 该位用于切换SPI工作于主机与从机模式。 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 位定义，SPI总线在空闲模式时时钟脚的电平状态。 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位定义在采样时所用的时钟边沿。 0 = SPI在时钟第一个边沿采样数据。 1 = SPI在时钟第二个边沿采样数据。

位	名称	描述																																								
1:0	SPR[1:0]	<p>SPI 时钟频率选择</p> <p>这两位设置4级SPI 时钟分频. 下面列出时钟频率</p> <p>$F_{sys} = 16 \text{ MHz}$</p> <table border="1"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>除频</th> <th>SPI时钟频率</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2</td> <td>8M 位/秒</td> </tr> <tr> <td>0</td> <td>1</td> <td>4</td> <td>4M 位/秒</td> </tr> <tr> <td>1</td> <td>0</td> <td>8</td> <td>2M 位/秒</td> </tr> <tr> <td>1</td> <td>1</td> <td>16</td> <td>1 M位/秒</td> </tr> </tbody> </table> <p>$F_{sys} = 24 \text{ MHz}$</p> <table border="1"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>除频</th> <th>SPI 时钟频率</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>2</td> <td>12M 位/秒</td> </tr> <tr> <td>0</td> <td>1</td> <td>4</td> <td>6M 位/秒</td> </tr> <tr> <td>1</td> <td>0</td> <td>8</td> <td>3M 位/秒</td> </tr> <tr> <td>1</td> <td>1</td> <td>16</td> <td>1.5M 位/秒</td> </tr> </tbody> </table> <p>SPR[1:0] 只在主机模式(MSTR = 1)下有效。从机模式下，时钟自动同步主机通过 SPICLK 管脚发送来的时钟，最高到$F_{sys}/2$</p>	SPR1	SPR0	除频	SPI时钟频率	0	0	2	8M 位/秒	0	1	4	4M 位/秒	1	0	8	2M 位/秒	1	1	16	1 M位/秒	SPR1	SPR0	除频	SPI 时钟频率	0	0	2	12M 位/秒	0	1	4	6M 位/秒	1	0	8	3M 位/秒	1	1	16	1.5M 位/秒
SPR1	SPR0	除频	SPI时钟频率																																							
0	0	2	8M 位/秒																																							
0	1	4	4M 位/秒																																							
1	0	8	2M 位/秒																																							
1	1	16	1 M位/秒																																							
SPR1	SPR0	除频	SPI 时钟频率																																							
0	0	2	12M 位/秒																																							
0	1	4	6M 位/秒																																							
1	0	8	3M 位/秒																																							
1	1	16	1.5M 位/秒																																							

SPCR2 – 串行外设控制寄存器 2

寄存器	SFR 地址	复位值
SPCR2	F3H, 页 1	0000_0000 b

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

位	名称	描述																																				
7:2	-	保留																																				
1:0	SPIS[1:0]	<p>SPI 临近字节间隔时间选择</p> <p>SPIS[1:0] 和 CPHA 设置8级SPI 邻近字节接隔时间如下:</p> <table border="1"> <thead> <tr> <th>CPHA</th> <th>SPIS1</th> <th>SPIS0</th> <th>SPI clock</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0.5</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1.0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1.5</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>2.0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1.0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1.5</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>2.0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>2.5</td></tr> </tbody> </table> <p>SPIS[1:0]只在主机模式 (MSTR = 1)时有效</p>	CPHA	SPIS1	SPIS0	SPI clock	0	0	0	0.5	0	0	1	1.0	0	1	0	1.5	0	1	1	2.0	1	0	0	1.0	1	0	1	1.5	1	1	0	2.0	1	1	1	2.5
CPHA	SPIS1	SPIS0	SPI clock																																			
0	0	0	0.5																																			
0	0	1	1.0																																			
0	1	0	1.5																																			
0	1	1	2.0																																			
1	0	0	1.0																																			
1	0	1	1.5																																			
1	1	0	2.0																																			
1	1	1	2.5																																			

表 6.12-1 片选管脚配置

DISMODF	SSOE	主机模式 (MSTR = 1)	从机模式 (MSTR = 0)
0	X	\overline{SS} 作为模式错误输入脚	\overline{SS} 作为片选输入
1	0	普通 I/O	
1	1	自动 \overline{SS} 输出	

SPSR – 串行外设状态寄存器

寄存器	SFR 地址	复位值
SPSR	F4H, 页 0	0000_0000 b

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

位	名称	描述
7	SPIF	SPI传输完成标志 在SPI数据传输完成或接收到的数据移入到SPI读缓冲时，该位通过硬件设置为1. 如果使能 ESPI和 EA， SPI中断请求。该位必须由软件清零。如果SPIF置1，禁止向SPInDR写入
6	WCOL	写冲突位 该位表示写冲突事件. 一旦发生写冲突事件，该位被置位，必须通过软件清零。
5	SPIOVF	SPI 溢出标志 该位表示溢出事件，一旦发生溢出，该位置位，如果使能ESPI 和 EA, SPI请求中断。该位必须由软件清零
4	MODF	模式错误中断状态标志 该位表示模式错误事件。 如果 \overline{SS} 配置成模式错误输入(MSTR=1且DISMODF=0) 和 \overline{SS} 被外部器件拉低， 将产生模式错误， MODF将被置1。如果使能 ESPI 和 EA， SPI中断请求。 该位必须由软件清零
3	DISMODF	禁止模式错误检测 位结合SSOE (SPInCR.7) 位用于决定 \overline{SS} 的特征。DISMODF 仅在主机模式下有效(MSTR = 1) 0 = 使能模式错误检测。 \overline{SS} 为模式错误检测的输入脚， 不管SSOE设置。 1 = 禁止模式错误检测。 \overline{SS} 的依赖SSOE 位配置。
2	TXBUF	SPI 写数据缓存状态 此位指示SPI发送缓存状态。 0 = SPI 写数据缓存空 1 = SPI 写数据缓存满。

SPDR –串行外设数据寄存器

寄存器	SFR 地址	复位值
SPDR	F5H, 页 0	0000_0000 b

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

位	名称	描述
7:0	SPDR[7:0]	<p>串行外设数据</p> <p>该字节为SPI总线上传输或接收的数据。一个字节的写入实际是写入到移位寄存器内操作。一个字节的读取，实际上是一个缓冲区读取数据操作。在主机模式，写该寄存器将启动传输并同时接收一个字节。</p>

6.13 12位模数转换器 (ADC)

6.13.1 概述

MS51内嵌12位逐次逼近寄存器型(SAR)的模拟数字转换器(ADC)。模数转换模块负责将管脚上的模拟信号转换为12位二进制数据。MS51支持16通道单端输入模式。内部带隙电压(band-gap voltage)为1.22V，同时也可用作内部ADC输入端。所有模拟电路复用同一组采样电路和同一组采样保持电容。该组采样保持电容为转换电路的输入端。然后转换器通过逐次逼近的方式得到有效结果并存放在结果寄存器中。ADC 控制器页支持连续转换模式，结果存入XRAM中。

6.13.2 功能描述

6.13.2.1 ADC 操作

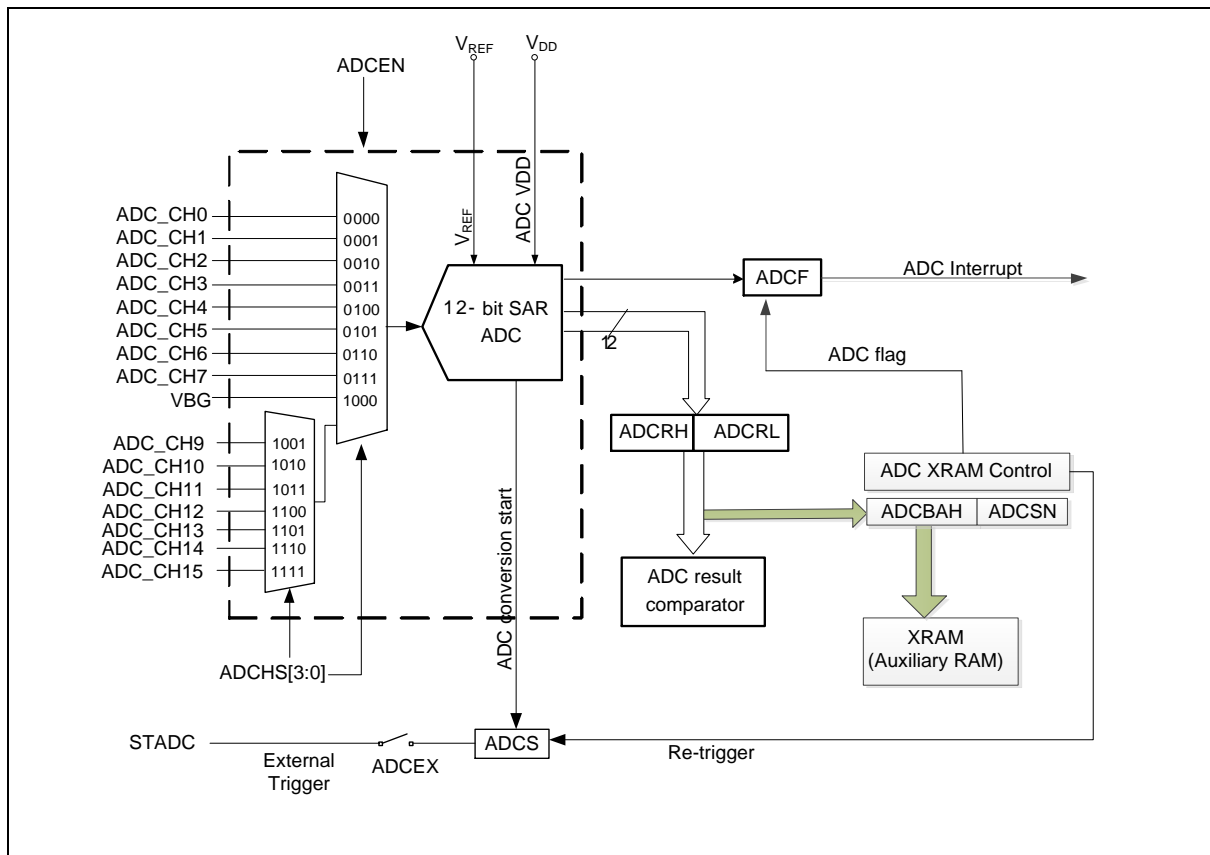


图 6.13-1 12 位 ADC 框图

在开始ADC转换前，通过设置ADCEN(ADCCON1.0)位使能ADC电路，从而激活ADC电路，由于ADC模块需要额外功耗，一旦ADC转换模块不再使用，建议清零ADCEN位关闭ADC模块电路以节省功耗。

ADC转换输入管脚需要特别配置，通过ADCHS[2:0] 来选择采样所需要的ADC通道连接到采样电路上。同时，用户需要通过PxMn寄存器把采样所用的管脚配置为“输入高阻模式”(input-only high impedance)。配置后用于ADC管脚与数字输出电路将断开，但数字输入电路仍然可以工作，因此数字输入将可能产生漏电流。所以还需要通过配置AINDIDS寄存器相应位来关闭数字输入缓冲区。如上配置后，ADC输入脚将变成纯模拟输入电路。同样ADC采样时钟也需要认真考虑。ADC最高时钟频率500kpsps 当采样时钟设置超过最大值时，采样结果数据为不可预测。

通过置ADCS位(ADCCON0.6)开启AD转换。当转换完成后，硬件会自动清除该位，同时置ADCF(ADCCON0.7)位，如果之前ADC中断已使能，则会产生ADC中断。转换结果存放在ADCRH (高8位) 及

$$4095 \times \frac{V_{AIN}}{V_{REF}}$$

ADCRL (低4位)中。12位转换结果值为

内部及外部数字电路，可能影响采样结果的准确度。所以如果需要高精度的转换结果，请参考如下应用，以降低噪声电平干扰。

1. 模拟输入脚尽量离芯片越近越好，避免管脚附近有高速数字电路经过，并离高速数字电路越远越好。
2. 在转换过程中，将芯片进入空闲模式。
3. 如果模拟输入脚 AIN 在系统中同时需要切换做数字管脚，请确保在转换过程中不要做数字/模拟切换动作。

下面列出ADC多功能管脚定义

组	管脚名称	GPIO	描述
ADC	ADC_CH0	P1.7	ADC通道模拟输入
	ADC_CH1	P3.0	ADC通道模拟输入
	ADC_CH2	P0.7	ADC通道模拟输入
	ADC_CH3	P0.6	ADC通道模拟输入
	ADC_CH4	P0.5	ADC通道模拟输入
	ADC_CH5	P0.4	ADC通道模拟输入
	ADC_CH6	P0.3	ADC通道模拟输入
	ADC_CH7	P1.1	ADC通道模拟输入
	ADC_CH9	P2.1	ADC通道模拟输入
	ADC_CH10	P2.2	ADC通道模拟输入
	ADC_CH11	P2.3	ADC通道模拟输入
	ADC_CH12	P2.4	ADC通道模拟输入
	ADC_CH13	P1.3	ADC通道模拟输入
	ADC_CH14	P1.4	ADC通道模拟输入

组	管脚名称	GPIO	描述
	ADC_CH15	P2.5	ADC通道模拟输入

6.13.2.2 外部触发ADC转换

除了通过软件启动AD转换外，MS51提供硬件触发方式启动AD转换。一旦ADCEX (ADCCON1.1)置1，选择PWM通道的边沿或周期，STADC管脚边沿自动触发启动AD转换(由硬件设置自动ADCS信号)。通过STADCPX(ADCCON1.6)可以灵活配置STADC的输入管脚，通过ETGSEL(ADCCON0[5:4])和ETGTYP (ADCCON1[3:2])设置来选择触发源和触发类型。同时，还可以在外部触发信号与启动AD转换之间插入触发延时(触发延时计数器)。MS51该功能将非常适用于高精度电机控制。注意，在AD模块转换过程中(ADCS = 1)，任何软件或硬件触发信号都是无效的。

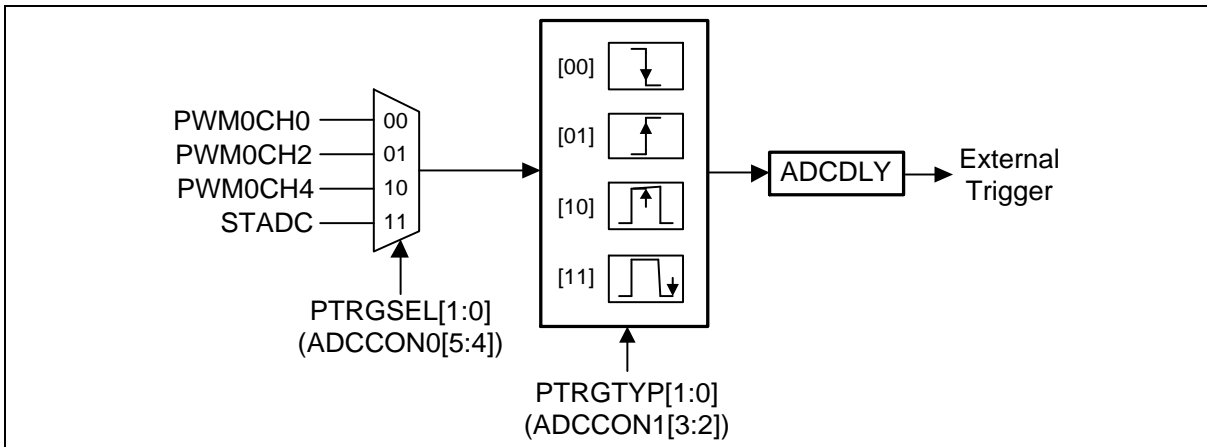


图 6.13-2 外部触发 ADC 电路

6.13.2.3 ADC 转换结果比较器

MS51 ADC 提供一组数字比较器，用于比较AD 12位转换结果与预先填入寄存器ACMPH 及 ACMPH 的内容是否一致。ADC 比较器使能位为ADCM PEN (ADCCON2.5)一旦设定，每次AD转换结束都会进行比较。ADCMPO(ADCCON2.4)显示根据ADCMPOP (ADCCON2.6)设定的比较结果。当ADFBEN (ADCCON2.7)设置后，ADC比较结果可触发PWM故障刹车。请参考章节 18.1.5“故障刹车”。

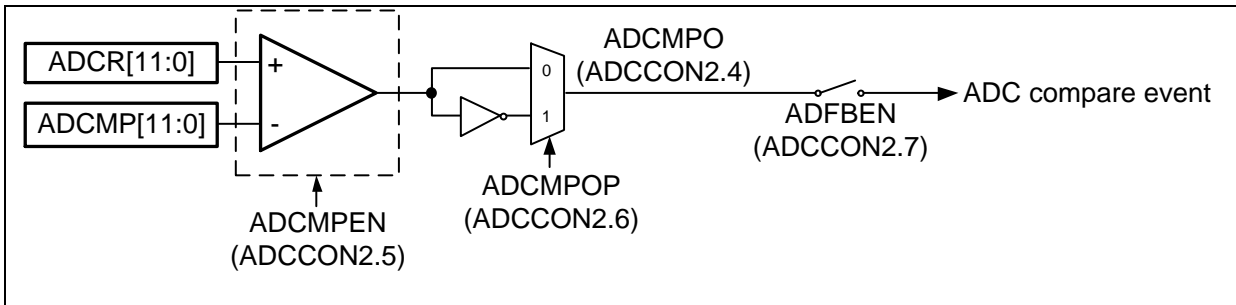


图 6.13-3 ADC 结果比较器

6.13.3 ADC 连续转换

ADC 控制器支持连续转换功能，自动存储A/D 转换结果。ADC 连续转换模式把 12-位 ADC 转换结果存入 XRAM 缓存, 12-位 ADC 数据自动分为 8-位 高字节和 4-位 低半字节两部分。考虑到降低 XRAM 内

存大小, 两笔 4-位 半字节数据 (连续 ADC 转换结果) 自动合并成一个字节存入XRAM。

存储方式如下图列出, 先存入 8-位 转换高字节数据, 然后存入合并的数据。分隔点是ADC 连续转换长度, 由ADCCN定义。

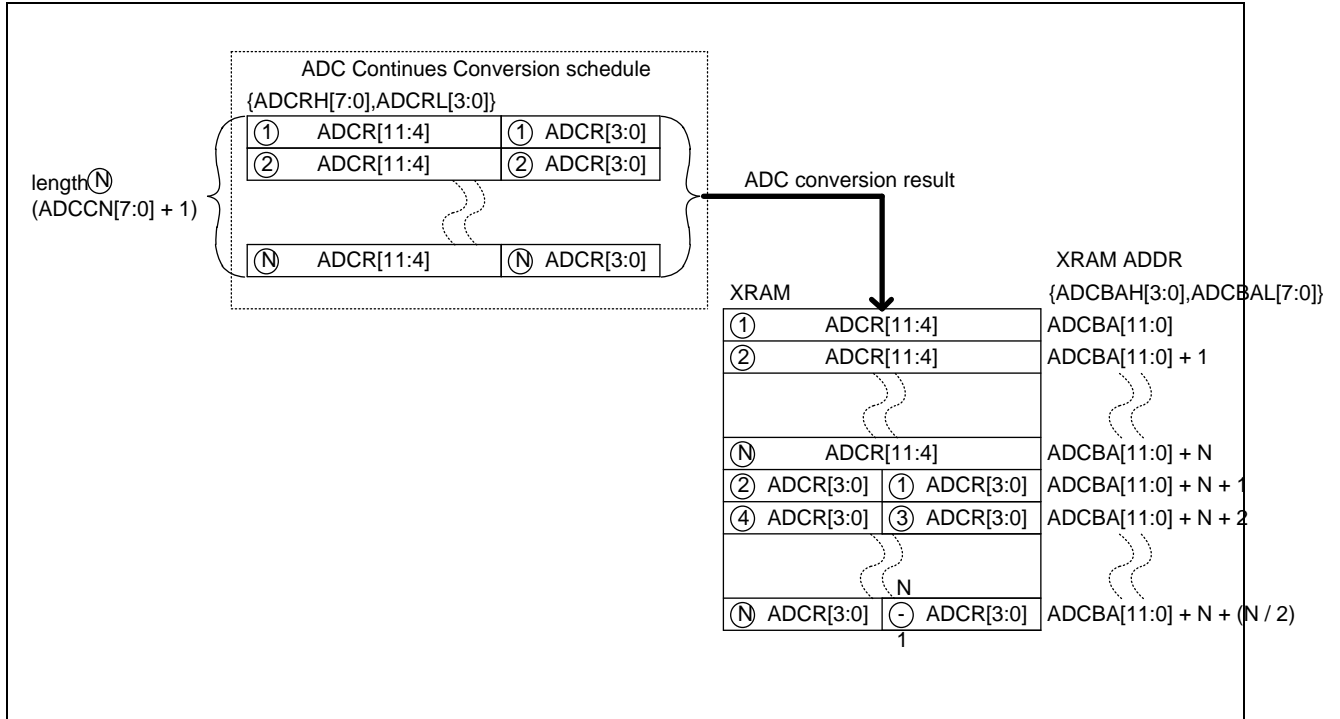


图 6.13-4 ADC 连续转换模式, 包含 DMA

编程流程描述如下:

1. 设置ADC通道、使能ADC，像普通ADC设置方法一样
2. 设置CONT(ADCCON1.4)位1设置ADC进入连续转换模式
3. 设置ADCBAH和ADCBAL寄存器配置转换结果存放地址
4. 设置ADCCN寄存器配置ADC转换数量.
5. 设置HIE/FIE(ADCCON1[5])使能ADC转换完成一半中断(可选)
6. 由软件触发(ADCS=1)或者外部触发(ADCEX=1)，开始 ADC 工作

6.13.4 ADC控制寄存器

ADCCON0 – ADC 控制 0 (位寻址)

寄存器	SFR 地址	复位值
ADCCON0	E8H, 所有页	0000_0000b

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

位	名称	描述
7	ADCF	ADC标志位 当AD转换完成, 该位置1. 可读取到当前AD转换结果. 该位为1时无法开始新一轮转换, 需要软件清零
6	ADCS	A/D 转换软件启动位 该位置1启动AD转换. 在AD转换过程中该位保持为1, 当转换结束硬件自动清0. 这意味着写入ADCS的值和读出的不一定相符 写: 0 = 无动作. 1 = 开始AD转换 读: 0 = ADC 模块空闲状态 1 = ADC 模块工作中
5:4	ETGSEL[1:0]	外部触发源选择 当 ADCEX (ADCCON1.1) 为1, 该位选择外部触发ADC的来源 00 = PWM0. 01 = PWM2. 10 = PWM4. 11 = STADC 脚.

位	名称	描述
3:0	ADCHS[3:0]	<p>A/D转换通道选择</p> <p>该位用于选择连接ADC转换通道。当ADCEN为0所有输入不连接到ADC..</p> <p>0000 = ADC_CH0 0001 = ADC_CH1. 0010 = ADC_CH2. 0011 = ADC_CH3. 0100 = ADC_CH4. 0101 = ADC_CH5. 0110 = ADC_CH6. 0111 = ADC_CH7. 1000 = VBG (内部带隙电压). 1001 = ADC_CH9. 1010 = ADC_CH10. 1011 = ADC_CH11. 1100 = ADC_CH12. 1101 = ADC_CH13. 1110 = ADC_CH14. 1111 = ADC_CH15.</p>

ADCCON1 – ADC 控制 1

寄存器	SFR 地址	复位值
ADCCON1	E1H, 页 0	0000_0000 b

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

位	名称	描述
7	OCEN	ADC 偏移校准使能寄存器 这里使能偏移校准功能 0: ADC 偏移校准使能, ADC 硬件自动校准 1: ADC 偏移校准禁止
6	STADCPX	外部触发ADC 触发管脚选择 0 = 选择 STADC 从 P0.4. 1 = 选择 STADC 从 P1.3. 注意配置此位后 STADC 管脚立刻改变
5:4	ADCDIV[1:0]	ADC ADCAQT 时钟除频 此字段定义采样点采集时钟分频器值 00 = $F_{ADCAQT} = F_{SYS}/1$. 01 = $F_{ADCAQT} = F_{SYS}/2$. 10 = $F_{ADCAQT} = F_{SYS}/4$. 11 = $F_{ADCAQT} = F_{SYS}/8$.
3:2	ETGTYP[1:0]	外部触发模式选择 当 ADCEX (ADCCON1.1) 为1,这里选择何种条件触发ADC转换. 00 = PWM0/2/4或STADC管脚下沿 01 = PWM0/2/4 或STADC管脚上升沿. 10 = PWM 周期中点 11 = PWM 周期结束. 注意周期中点中断和周期结束中断只在PWM 中心对齐模式有效.
1	ADCEX	ADC 转换触发选择 这一位选择触发AD转换的条件. 0 = A/D 转换只由设置 ADCS 位触发 1 = A/D 转换通过设置 ADCS 位或做通过ETGSEL[1:0] 和 ETGTYP[1:0]设置的外部触发源触发。注意当 ADCS 为 1 (转换正忙), ADC 会忽略后来的触发源直到 ADCS被硬件清除

位	名称	描述
0	ADCEN	ADC 使能 0 = ADC 电路关闭 1 = ADC 电路使能.

ADCCON2 – ADC控制 2

寄存器	SFR 地址	复位值
ADCCON2	E2H, 页 0	0000_0000 b

7	6	5	4	3	2	1	0
ADFBEN	ADCMPOP	ADCMPEM	ADCMPO	ADCAQT0[2:0]			ADCDLY.8
读/写	读/写	读/写	只读	读/写			读/写

位	名称	描述
7	ADFBEN	ADC 转换结果宣告故障刹车使能 0 = ADC 宣告故障刹车禁止。 1 = ADC 宣告故障刹车使能。当比较结果 ADCMPO 为1触发故障刹车模块。即符合PWM故障刹车输出值后，硬件将清除 PWMRUN (PWMCON0.7)，并终止PWM输出。当PWMRUN置1，PWM重新输出。
6	ADCMPOP	ADC 比较器输出极性 0 = 若ADCR[11:0]大于或等于ADCMP[11:0]， ADCMPO 为 1 1 = 若ADCR[11:0]小于ADCMP[11:0]， ADCMPO 为 1
5	ADCMPEM	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ADC比较值定义的条件匹配时，ADCF才会设置为1。该条件基于ADCMPLH，ADCMPL和ADCMPOP寄存器的设定值。 仅当ADC比较结果符合条件时，ADCF寄存器才会变为1；如果使能ADC中断，则进入中断向量。 0 = ADC 结果比较器禁止。 1 = ADC 结果比较器使能 註:使能该位并触发ADC启动后，ADC继续转换。寄存器ADCRH和ADCRL的值将根据ADC设置的结果而改变，也可以从寄存器中读出。仅在ADCF设置为1后，此过程才停止
4	ADCMPO	ADC比较器输出值 该位输出ACMPOP设定比较输出的结果，基於ACMPOP設定值。每次AD转换结束都会更新输出。
3:1	ADCAQT0[2:0]	ADC采集时间 1 这个3位字段决定ADC AIN0~AIN7和BG采样的捕获时间，遵循下面的等式： $\text{ADC 采集时间} = \frac{4 * \text{ADCAQT} + 6}{F_{\text{ADC}}} \cdot F_{\text{ADCAQT}}$ 在 ADCDIV (ADCCON2[3:1])定义 默认 $F_{\text{ADCAQT}} = F_{\text{SYS}}$ (ADCDIV=0)，因为最小取样时间必须超过370ns，这意味着 $F_{\text{sys}} = 24\text{MHz}$ 时，ADCAQT应该设定为1。 默认及最小捕获时间是6个ADC时钟周期。注意，当ADC正在转换时，这个字段不应该被改变
0	ADCDLY.8	ADC 外部触发延时计数器第8位 见 ADCDLY 寄存器

ADCCON3 – ADC 控制 3

寄存器	SFR 地址	复位值
ADCCON3	86H, 页 2	0000_0000 b

7	6	5	4	3	2	1	0
-	-	HIE	CONT	ADCAQT1[2:0]			SLOW
-	-	读/写	读/写	读/写			读/写

位	名称	描述
7:6	-	Reserved
5	HIE	ADC 一半完成中断使能 0 = 连续模式下, A/D 转换完成一半不会产生ADC中断 1 = 连续模式下, A/D 转换完成一半会产生ADC中断
4	CONT	ADC连续采样选择 0 = ADC 单次采样, 一次 A/D 转换完成后会产生ADC中断 1 = ADC 连续采样. 所有A/D 转换完成后会产生ADC中断
3:1	ADCAQT1	ADC 采样时间 1 这个3位字段决定ADC AIN9~AIN15采样的捕获时间, 遵循下面的等式: $\text{ADC 采集时间} = \frac{4 * \text{ADCAQT} + 6}{F_{\text{ADC}}}$ F _{ADC} 在 ADCDIV (ADCCON2[3:1])定义 默认及最小捕获时间是6个ADC时钟周期。注意, 当ADC正在转换时, 这个字段不应该被改变
0	SLOW	ADC慢速选择 此位选择 ADC慢速模式 0 = ADC 转换时间是高速 1400ns, F _{ADC} = 714 ksps 1 = ADC转换时间是低速 4750ns, F _{ADC} = 215 ksps, 注意1: 慢速 ADC 转换可以提高 ADC可以提高当VDD低于2.5V时的转换质量 注意2: AUXR1.4 (SLOW) 或 ADCCON3 .0(SLOW) 设为 1, ADC 进入低速模式.

ADCDLY – ADC触发延时计数器

寄存器	SFR 地址	复位值
ADCDLY	E3H, 页 0	0000_0000 b

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

位	名称	描述
7:0	ADCDLY[7:0]	<p>ADC 外部触发启动延迟计数器低位</p> <p>该8位寄存器与ADCCON2.0组成9位计数器，用于在外部触发启动ADC之前加入一段延迟。延迟计数结束再开始ADC转换</p> $\text{外部延迟时间} = \frac{\text{ADCDLY}}{F_{\text{ADC}}}$ <p>注，该延迟仅当 ADCEX (ADCCON1.1) 置1时有效。如果启用PWM输出触发ADC功能，在PWM运行过程中不得更改ADCDLY计数值。</p>

AINDIDS0 – ADC 通道数字通道关闭

寄存器	SFR 地址	复位值
AINDIDS0	F6H, 页 0	0000_0000 b

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

位	名称	描述
7:0	AINnDIDS	ADC 通道数字通道关闭 0 = 使能 ADC通道的数字通道. 1 =禁止 ADC 通道数字输入, 读ADC 通道一直读到 0.

AINDIDS1 – ADC 通道数字通道关闭

寄存器	SFR 地址	复位值
AINDIDS1	99H, 页 2	0000_0000 b

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

位	名称	描述
n	PnnDIDS	ADC 通道数字通道关闭 0 = 使能 ADC通道的数字通道. 1 =禁止 ADC 通道数字输入, 读ADC 通道一直读到 0.

ADCRH – ADC 结果高字节

寄存器	SFR 地址	复位值
ADCRH	C3H, 页 0	0000_0000 b

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

位	名称	描述
7:0	ADCR[11:4]	ADC 结果高字节 ADC 转换结果的高8位存入这个寄存器。

ADCRL – ADC 结果低字节

寄存器	SFR 地址	复位值
ADCRL	C2H, 页 0	0000_0000 b

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

位	名称	描述
7:4	-	保留
3:0	ADCR[3:0]	ADC结果低字节 ADC 转换结果低4位存入这个寄存器

ADCM PH – ADC 比较值高字节

寄存器	SFR 地址	复位值
ADCM PH	CFH, 页 0	0000_0000 b

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

位	名称	描述
7:0	ADCM P[11:4]	ADC 比较值高字节 ADC 比较值高字节存入这个寄存器.

ADCMPL – ADC 比较值低字节

寄存器	SFR 地址	复位值
ADCMPL	CEH, 页 0	0000_0000 b

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

位	名称	描述
7:4	-	保留
3:0	ADCMP[3:0]	ADC 比较值低字节 ADC 比较值低4位存入这个寄存器

ADCBAH – ADC内存基地址高字节

寄存器	SFR 地址	复位值
ADCBAH	84H, 页 2	0000_0000 b

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

位	名称	描述
7:4	-	保留
3:0	ADCBAH[3:0]	ADC 内存基地址 (高字节) 内存基地址的高4位，用来存放ADC连续采样数据。 内存基地址ADCBA[11:0] = {ADCBAH[3:0], ADCBAL[7:0]}

ADCBAL – ADC 内存基地址低字节

寄存器	SFR地址	复位值
ADCBAL	85H, 页 2	0000_0000 b

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

位	名称	描述
7:0	ADCBAL[7:0]	ADC 内存基地址 (低字节) 内存基地址的低8位，用来存放ADC连续采样数据。 内存基地址ADCBA[11:0] = {ADCBAL[3:0], ADCBAL[7:0]}

ADCSN – ADC 采样个数

寄存器	SFR 地址	复位值
ADCSN	8DH, 页 2	0000_0000 b

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

位	名称	描述
7:0	ADCSN[7:0]	ADC 采样数目 设置ADC连续采样的采样数目 总的采样数= ADCSN[7:0] + 1

ADCCN – ADC 目前采样数目

寄存器	SFR 地址	复位值
ADCCN	8EH, 页 2	0000_0000 b

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

位	名称	描述
7:0	ADCCN[7:0]	ADC 目前采样数目 目前ADC 连续采样模式的采样数目 目前采样数目 = ADCCN[7:0] + 1

ADCSR – ADC 状态寄存器

寄存器	SFR 地址	复位值
ADCSR	8FH, 页 2	0000_0000 b

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

位	名称	描述
[7:3]	保留	保留
[2]	CMPHIT	ADC 比较条件满足标志 这一位当 ADCMPO (ADCCON2.4) 标志变1后由硬件置1 注意: 此位写0清0.
[1]	HDONE	A/D 转换半完成标志 这一位当连续模式下 ADCSN A/D 转换完成一半后由硬件置位. 注意: 此位写0清0.
[0]	FDONE	A/D 转换完全完成标志 此位在连续模式下 ADCSN A/D 转换全部完成或做单次模式下完成一次转换后由硬件置1. 注意: 此位写0清0.

AUXR1 – 辅助寄存器 1

寄存器	SFR 地址	复位值
AUXR1	A2H , 页 0	POR: 0000 0000b, 软件复位 1U00 0000b, nRESET 管脚: U100 0000b, 其他: UUU0 0000b

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

位	名称	描述
4	SLOW	ADC 慢速选择 此位用来选择 ADC 慢速模式。 0 = ADC 转换时间为高速 1400ns, $F_{ADC} = 714$ ksp/s 1 = ADC 转换时间是低速 4750ns, $F_{ADC} = 215$ ksp/s, 注意1: 慢速 ADC 转换可以提升 VDD 低于 2.5V 时的转换结果 注意2: AUXR1.4 (SLOW) 或 ADCCON3 .0(SLOW) 设为1, ADC 进入慢速模式.

6.14 辅助功能

6.14.1 双 DPTR

传统8051架构仅有一组DPTR（数据指针）。在一组DPTR的结构中，当需要从某个绝对地址移动数据至另一个绝对地址时，程序非常冗长。MS51提供两组数据指针，这样程序可以分别同时定义源地址和目标地址，直接进行数据移动。程序通过DPS(AUXR0.0)位切换DPTR及DPTR1。

下例为64字节双DPTR。给定源地址和目标地址，然后采用循环指令可以简单移动整组数据，比单DPTR指令效率高很多。相对ORL 或 ANL指令，使用INC AUXR1 指令是最简短（2字节）指令。AUXR1.1始终为零，所以每次执行加一指令即可，不会影响该寄存器其它位。

```

MOV    R0,#64                ;number of bytes to move
MOV    DPTR,#D_Addr          ;load destination address
INC    AUXR1                  ;change active DPTR
MOV    DPTR,#S_Addr          ;load source address
LOOP:
MOVX   A,@DPTR               ;read source data byte
INC    AUXR1                  ;change DPTR to destination
MOVX   @DPTR,A               ;write data to destination
INC    DPTR                   ;next destination address
INC    AUXR1                  ;change DPTR to source
INC    DPTR                   ;next source address
DJNZ   R0,LOOP
INC    AUXR1                  ;(optional) restore DPS
    
```

AUXR1第3位用于提供用户一个通用标志位GF2，可以通过软件置1或清0

DPL – 数据指针低字节

寄存器名称	SFR 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]	<p>数据指针低字节</p> <p>这是16位数据指针的低字节，DPL 结合DPH作为16位的数据指针DPTR访问想要访问的RAM或编程内存地址。DPS (AUXR0.0) 位决定哪一个数据指针DPTR 或 DPTR1激活。</p>

DPH – 数据指针高字节

寄存器	SFR 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]	<p>数据指针高字节</p> <p>这是16位数据指针的高字节， DPL 结合DPH作为16位的数据指针DPTR访问想要访问的RAM或编程内存地址。DPS (AUXR0.0) 位决定哪一个数据指针DPTR 或 DPTR1激活</p>

AUXR1 – 辅助寄存器 1

寄存器	SFR 地址	复位值
AUXR1	A2H, 页 0	POR: 0000 0000b, 软件复位: 1U00 0000b, nRESET管脚: U100 0000b, 其他: UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	0	DPS
读/写	读/写	读/写	读/写	读/写	读/写	R	读/写

位	名称	描述
3	GF2	通用标志位 2 可通过软件置位或清除.
0	DPS	数据指针选择 0 = 数据指针0 (DPTR) 默认有效. 1 = 数据指针1 (DPTR1)有效. 当通过 DPS 切换当前有效DPTR后, 之前生效的DPTR寄存器内的值保持不变

7 应用电路

7.1 供电电路

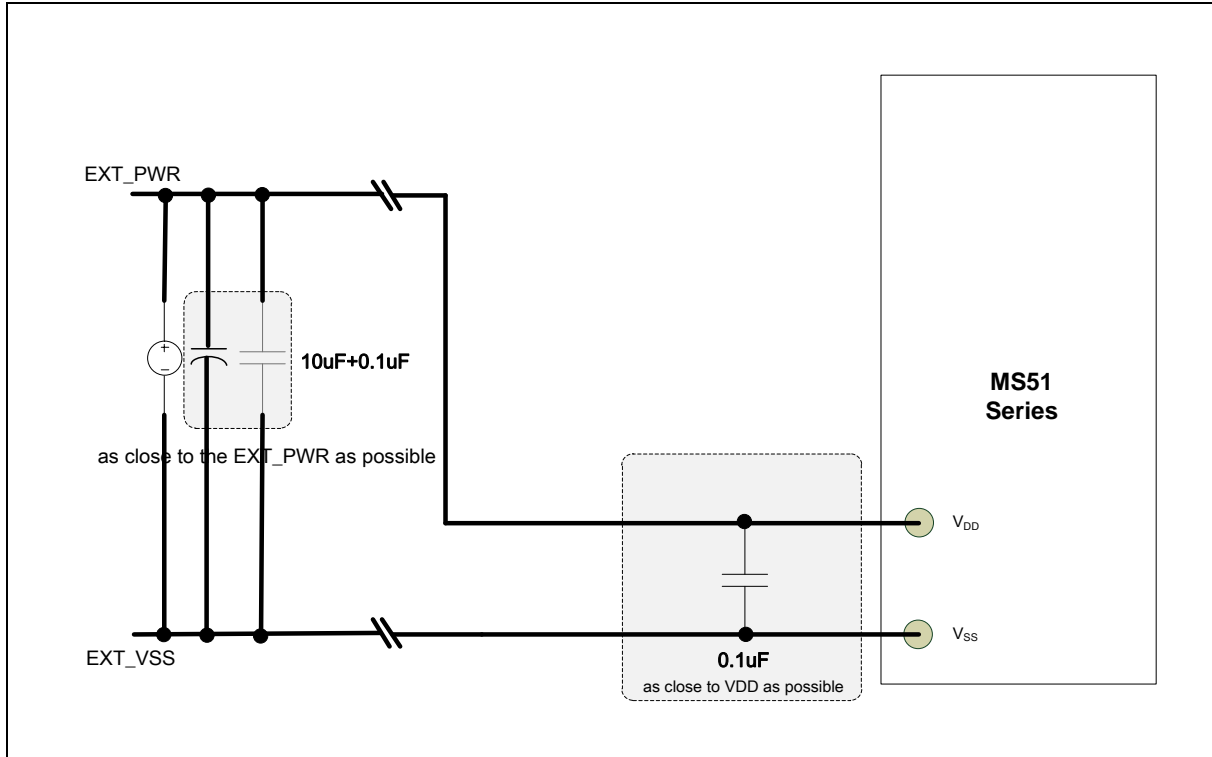


图 7.1-1 NuMicro® MS51 供电电路

7.2 外围应用电路

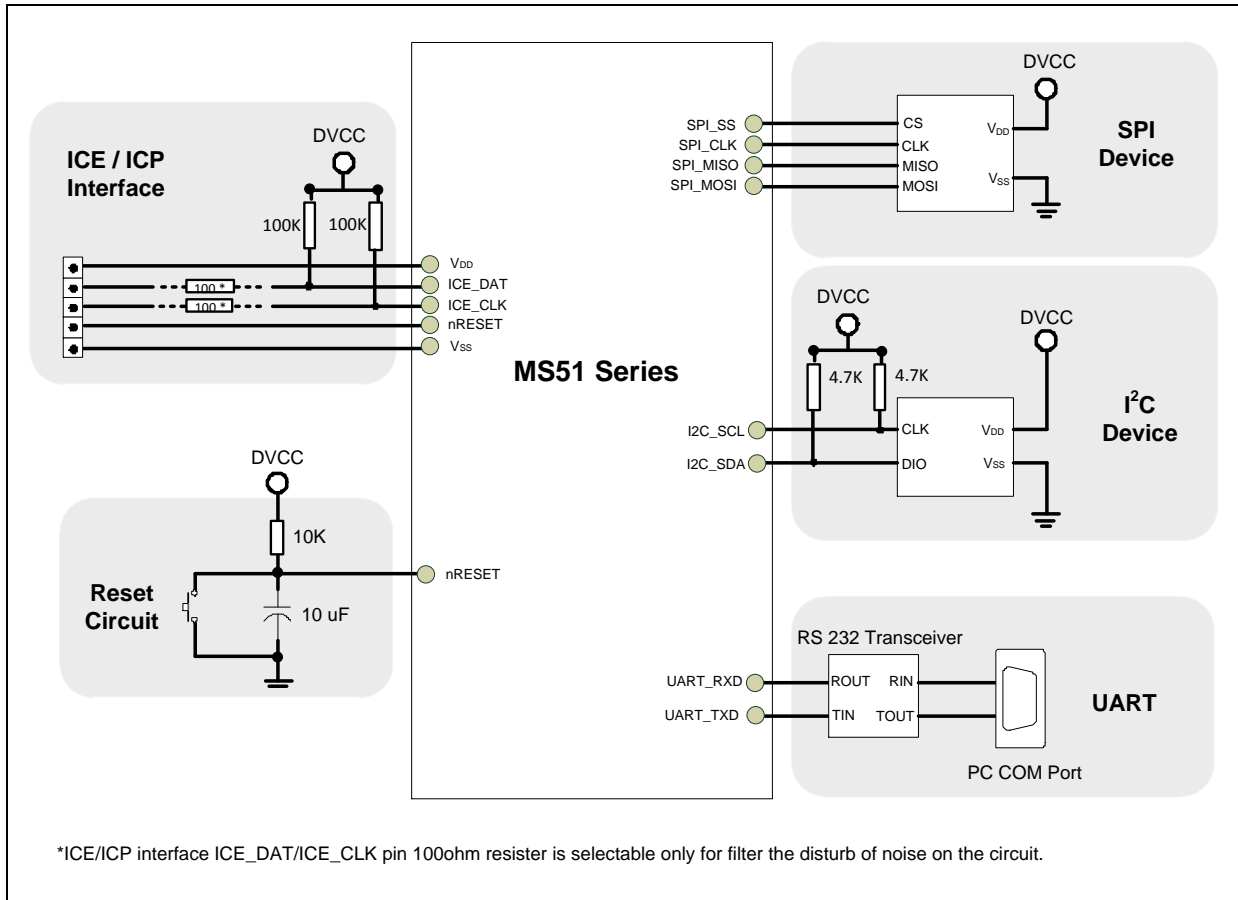


图 7.2-1 NuMicro® MS51 外围接口电路

8 电气特性

请参考MS51 相关规格数据文件了解电气特性参数

9 封装尺寸

9.1 QFN 33-管脚 (4.0 x 4.0 x 0.8 毫米)

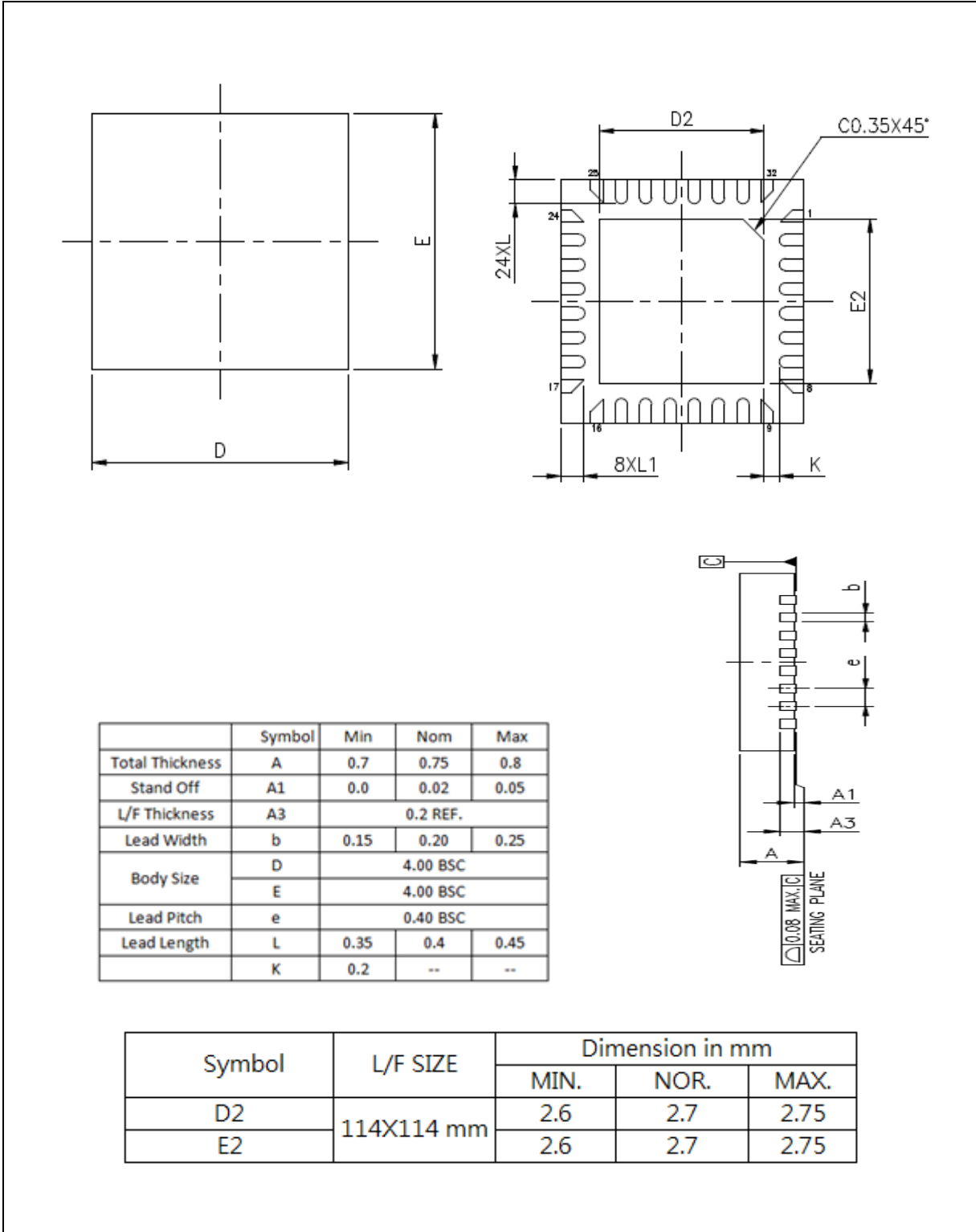


图 9.1-1 QFN-33 封装尺寸

9.2 LQFP 32-管脚 (7.0 x 7.0 x 1.4 毫米)

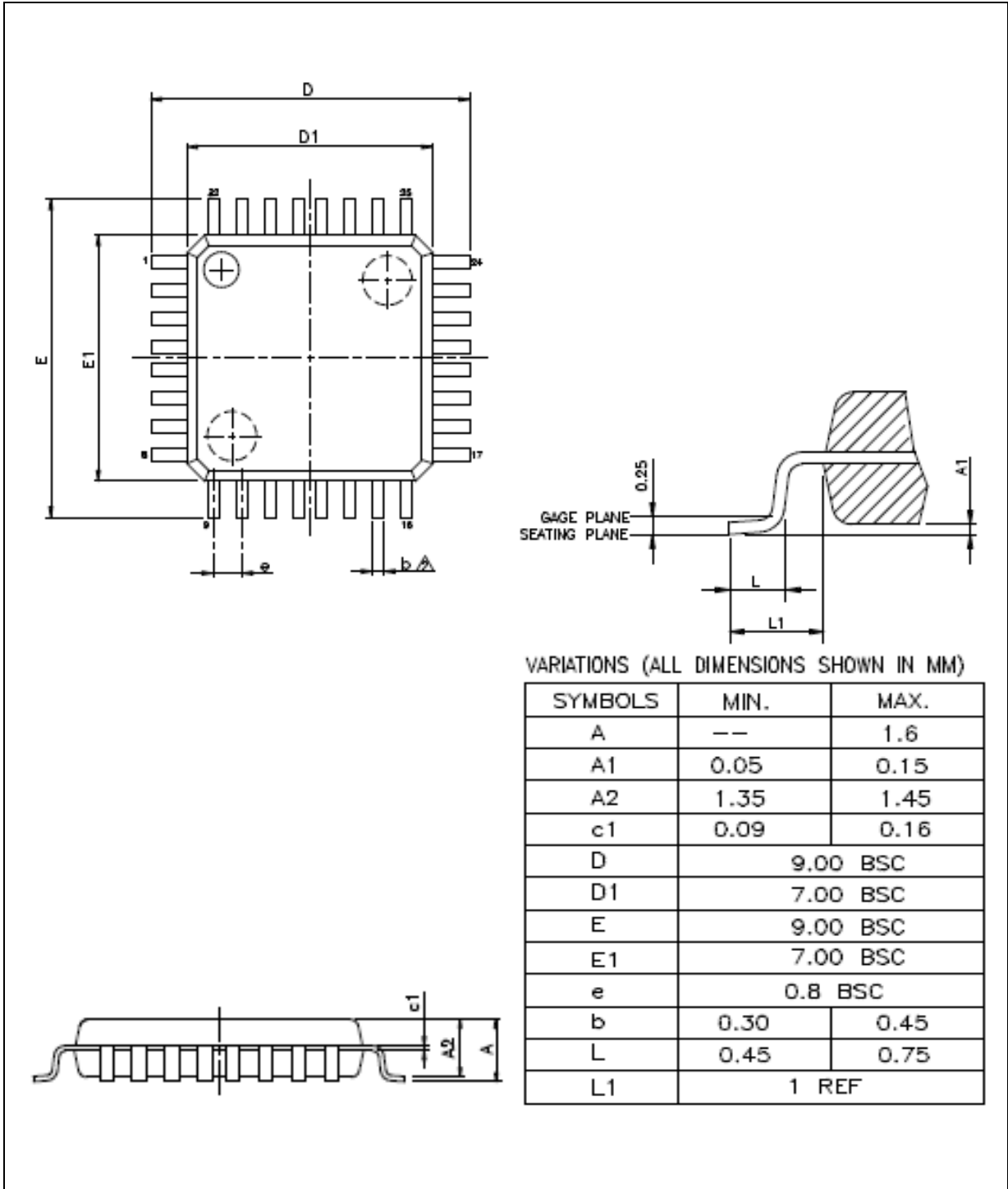


图 9.2-1 LQFP-32 封装尺寸

9.3 TSSOP 28-管脚 (4.4 x 9.7 x 1.0 毫米)

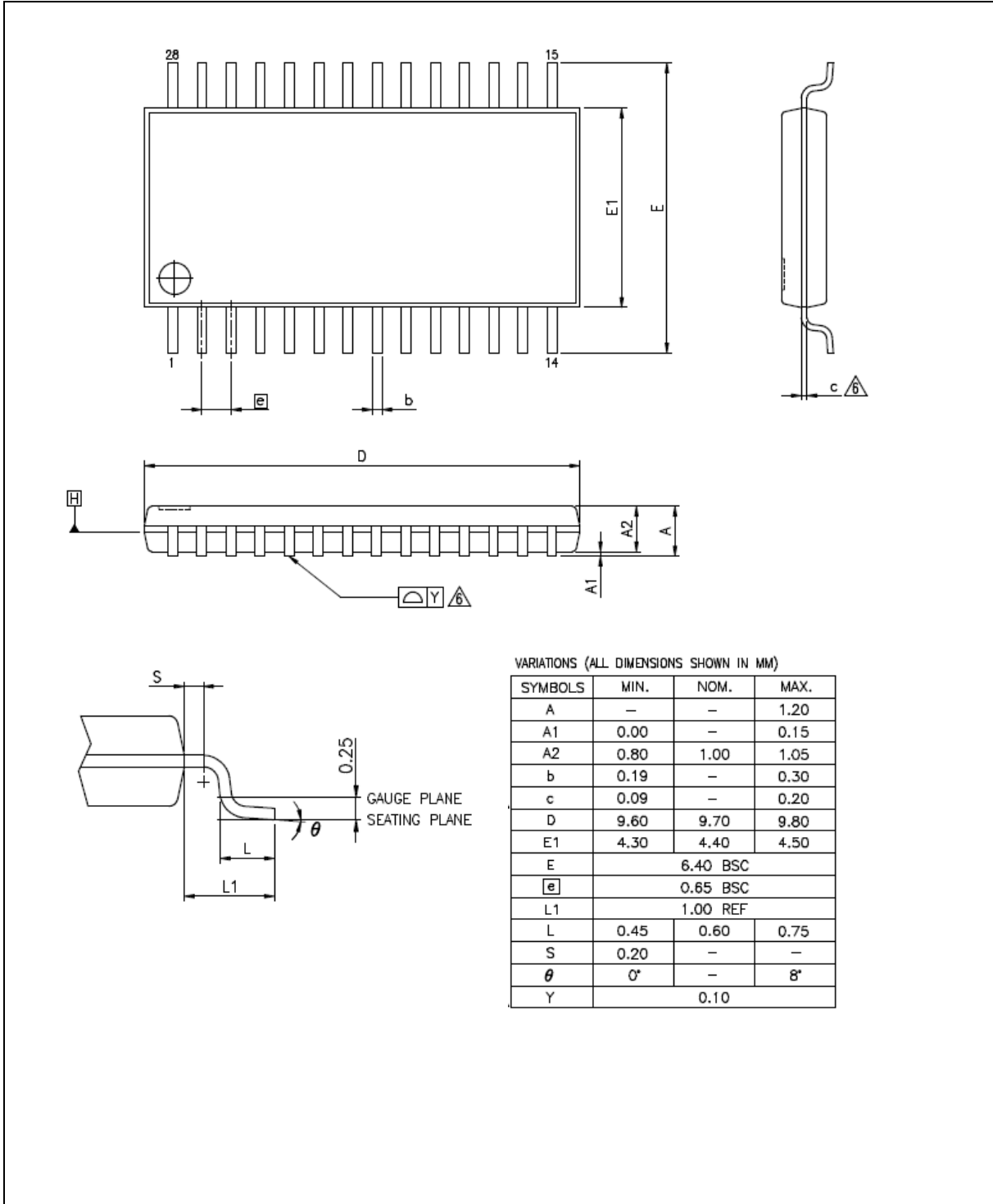


图 9.3-1 TSSOP-28 封装尺寸

9.4 TSSOP 20-管脚 (4.4 x 6.5 x 0.9 毫米)

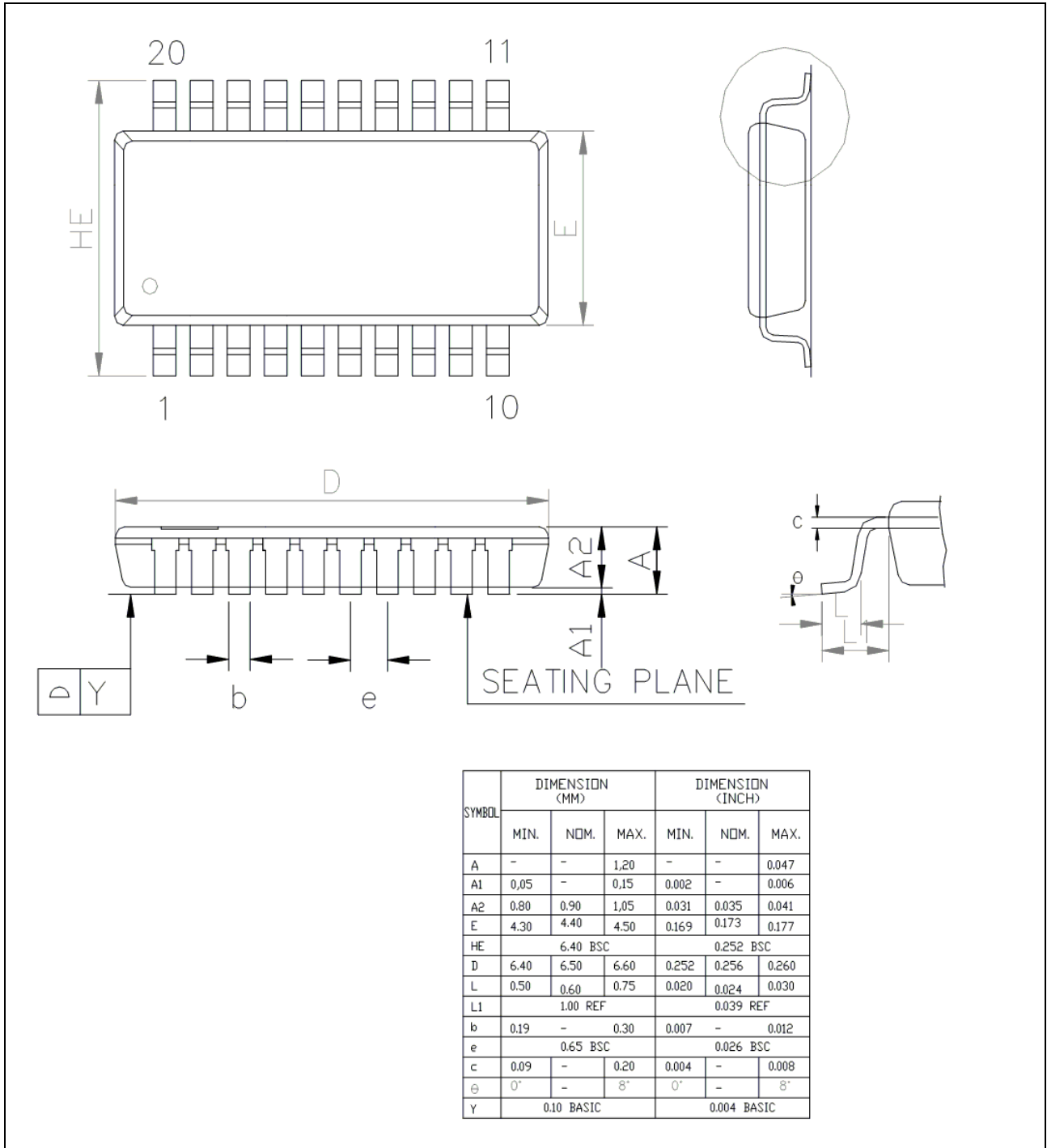


图 9.4-1 TSSOP-20 封装尺寸

9.5 QFN 20-管脚 (3.0 x 3.0 x 0.5毫米)

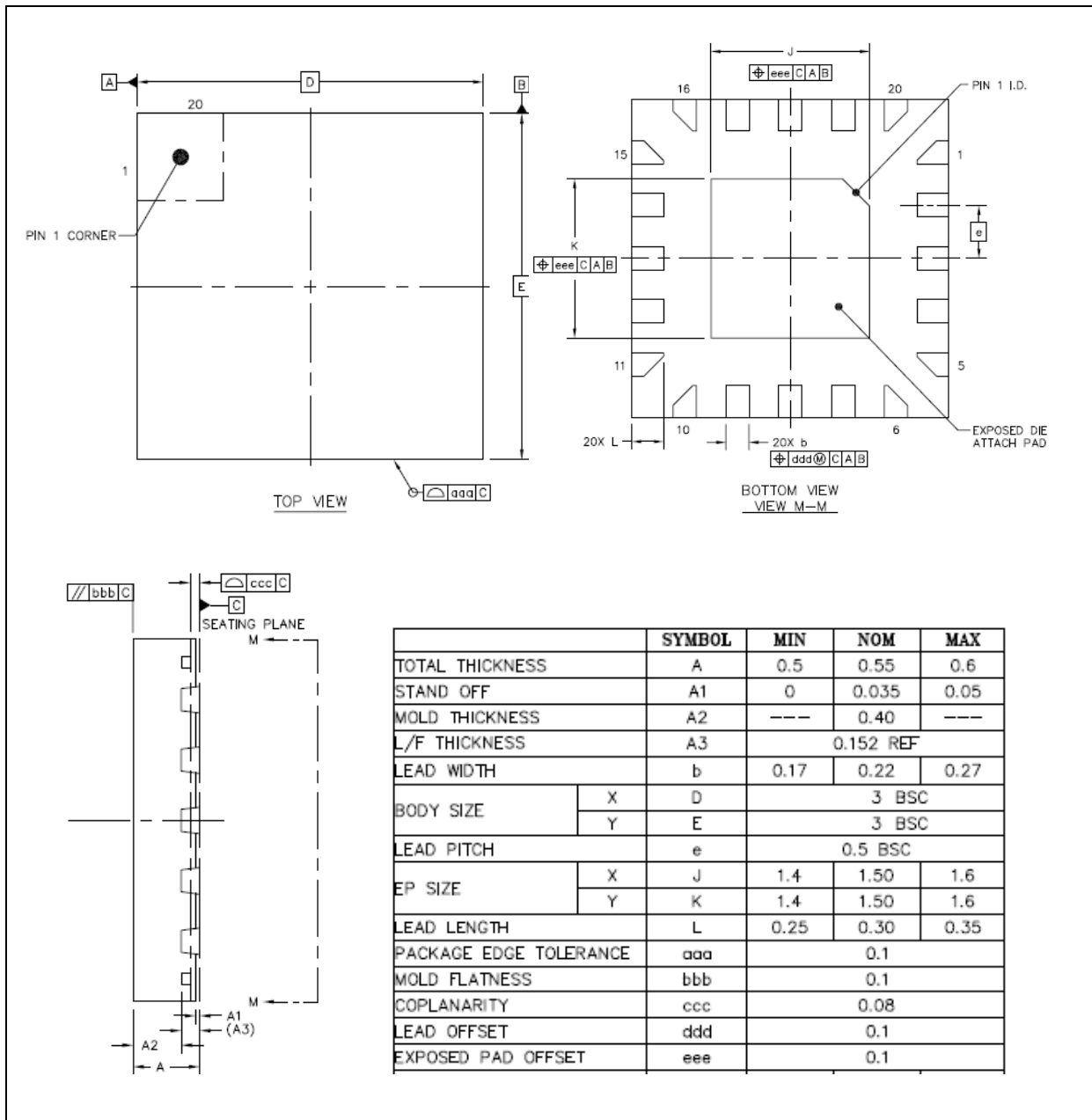


图 9.5-1 QFN-20 封装尺寸

10 名词缩写

10.1 名词缩写

首字母缩写词	描述
ADC	Analog-to-Digital Converter
BOD	Brown-out Detection
GPIO	General-Purpose Input/Output
Fsys	Frequency of system clock
HIRC	12 MHz Internal High Speed RC Oscillator
IAP	In Application Programming
ICP	In Circuit Programming
ISP	In System Programming
LDO	Low Dropout Regulator
LIRC	10 kHz internal low speed RC oscillator (LIRC)
LVR	Low Voltage \$eset
PDMA	Peripheral Direct Memory Access
POR	Power On Reset
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
UCID	Unique Customer ID
WKT	Wakeup Timer
WDT	Watchdog Timer

表 10.1-1 名词缩写列表

11 指令集

11.1 指令集以及寻址模式

MS51执行所有标准8051的指令集与MCS-51全兼容，每条指令执行周期根据1T 8051内核标准执行。该架构消除了冗余总线状态，且实现了取指、译码和运行并行执行。MS51每个机器周期占用一个时钟周期，对比传统12T 80C51系统，同样的系统时钟频率，提升性能至8.1倍（基于MIPS）。当然实际的速率还是需要基于指令编译的结果。

所有指令代码为 8-位 OPCODE，单字节来源于程序存储器，OPCODE 通过CPU译码。决定系统的工作情况以及从存储器中运行数据。如果没有其它数据请求，为单字节指令。在一些应用中，需要更多的数据，使用2或3字节指令。

表 11.1-1 指令集详细列举了所有的指令集，指令注释以及寻址模式如下所示

指令	寻址模式
Rn (N = 0~7)	当前选择寄存器区域的寄存器R0~R7。
Direct	8位内部数据地址。可作为内部RAM地址(00H 到 7FH) 或者 SFR (80H 到 FFH)。
@RI (I = 0, 1)	通过R0和R1间接寻址的8位内部RAM地址(00H 到 FFH)。
#data	指令包含8位常量。
#data16	指令包含16位常量。
Addr16	LCALL和LJMP使用的16位目标地址。分支可达程序空间的任何位置。
Addr11	ACALL 和 AJMP 使用的11位目标地址。分支可达2K字节程序内存。
Rel	带符号 (2's 互补) 8位偏移字节。被SJMP和所有条件分支使用。范围为 -128 to +127 字节。
Bit	对内存数据RAM或SFR直接寻址。

表 11.1-1 指令集寻址方式

指令	CY	OV	AC	指令	CY	OV	AC
ADD	X ⁽¹⁾	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	0	X		ANL C, /bit	X		
DIV	0	X		ORL C, bit	X		
DA A	X			ORL C, /bit	X		
RRC A	X			MOV C, bit	X		
RLC A	X			CJNE	X		
SETB C	1						

注: X 代表根据指令产生修改结果

表 11.1- 2 指令影响标志位结果

11.2 读-修改-回写指令

从特殊功能寄存器或RAM读一个字节，修改值，然后再写回去，这叫“读-修改-回写”指令。这种指令用于引脚时，读和回写都是输出锁存器的值，不是引脚电平，这些指令如下：

<u>指令</u>	<u>描述</u>
ANL	逻辑与 (ANL direct, A 和 ANL direct, #data)
ORL	逻辑或(ORL direct, A 和 ORL direct, #data)
XRL	逻辑异或(XRL direct, A 和 XRL direct, #data)
JBC	位检测清0跳转，如果 bit = 1 (JBC bit, rel)
CPL	位取反, (CPL bit)
INC	增1, (INC direct)
DEC	减1, (DEC direct)
DJNZ	减1若为0就跳转 (DJNZ direct, rel)
MOV	位操作, (MOV bit, C)
CLR	位操作, (CLR bit)
SETB	位操作, (SETB bit)

最后三个貌似非“读-修改-回写”，实际会先读整组引脚值，修改后回写。

11.3 指令集列表

指令	OPCODE	字节	时钟周期	MS51与传统 80C51 速率比值
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
INC A	04	1	1	12
INC Rn	08~0F	1	3	4
INC direct	05	2	4	3
INC @Ri	06, 07	1	5	2.4
INC DPTR	A3	1	1	24
DEC A	14	1	1	12
DEC Rn	18~1F	1	3	4
DEC direct	15	2	4	3
DEC @Ri	16, 17	1	5	2.4
MUL AB	A4	1	4	12
DIV AB	84	1	4	12
DA A	D4	1	1	12
ANL A, Rn	58~5F	1	2	6
ANL A, direct	55	2	3	4
ANL A, @Ri	56, 57	1	4	3
ANL A, #data	54	2	2	6
ANL direct, A	52	2	4	3
ANL direct, #data	53	3	4	6
ORL A, Rn	48~4F	1	2	6
ORL A, direct	45	2	3	4
ORL A, @Ri	46, 47	1	4	3
ORL A, #data	44	2	2	6
ORL direct, A	42	2	4	3
ORL direct, #data	43	3	4	6
XRL A, Rn	68~6F	1	2	6
XRL A, direct	65	2	3	4
XRL A, @Ri	66, 67	1	4	3

指令	OPCODE	字节	时钟周期	MS51与传统 80C51 速率比值
XRL A, #data	64	2	2	6
XRL direct, A	62	2	4	3
XRL direct, #data	63	3	4	6
CLR A	E4	1	1	12
CPL A	F4	1	1	12
RL A	23	1	1	12
RLC A	33	1	1	12
RR A	03	1	1	12
RRC A	13	1	1	12
SWAP A	C4	1	1	12
MOV A, Rn	E8~EF	1	1	12
MOV A, direct	E5	2	3	4
MOV A, @Ri	E6, E7	1	4	3
MOV A, #data	74	2	2	6
MOV Rn, A	F8~FF	1	1	12
MOV Rn, direct	A8~AF	2	4	6
MOV Rn, #data	78~7F	2	2	6
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

指令	OPCODE	字节	时钟周期	MS51与传统 80C51 速率比值
SETB bit	D2	2	4	3
CPL C	B3	1	1	12
CPL bit	B2	2	4	3
ANL C, bit	82	2	3	8
ANL C, /bit	B0	2	3	8
ORL C, bit	72	2	3	8
ORL C, /bit	A0	2	3	8
MOV C, bit	A2	2	3	4
MOV bit, C	92	2	4	6
ACALL addr11	11, 31, 51, 71, 91, B1, D1, F1 ^[2]	2	4	6
LCALL addr16	12	3	4	6
RET	22	1	5	4.8
RETI	32	1	5	4.8
AJMP addr11	01, 21, 41, 61, 81, A1, C1, E1 ^[3]	2	3	8
LJMP addr16	02	3	4	6
SJMP rel	80	2	3	8
JMP @A+DPTR	73	1	3	8
JZ rel	60	2	3	8
JNZ rel	70	2	3	8
JC rel	40	2	3	8
JNC rel	50	2	3	8
JB bit, rel	20	3	5	4.8
JNB bit, rel	30	3	5	4.8
JBC bit, rel	10	3	5	4.8
CJNE A, direct, rel	B5	3	5	4.8
CJNE A, #data, rel	B4	3	4	6
CJNE Rn, #data, rel	B8~BF	3	4	6
CJNE @Ri, #data, rel	B6, B7	3	6	4
DJNZ Rn, rel	D8~DF	2	4	6
DJNZ direct, rel	D5	3	5	4.8

注意：
 1. MS51 没有外部内存总线，MOVX 指令用来访问内部 XRAM。
 2. 11-位地址 [A10:A8] 高3位决定ACALL 16进制码。为[A10,A9,A8,1,0,0,0,1]。
 3. 11-位地址 [A10:A8] 高3位决定AJMP 16进制码，为 [A10,A9,A8,0,0,0,0,1]。

表 11.3-1 指令集

12 版本历史

日期	版本	描述
2019.11.28	1.00	初始版本
2020.02.21	1.01	章节 3.2 更改选型指南表内 MS51FC0AE/MS51XC0BE ISO 7816-3 数量为 2 章节 4.2 引脚描述表内P3.0新增 PWM2_CH1引脚定义描述.

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [32-bit Microcontrollers - MCU category](#):

Click to view products by [Nuvoton manufacturer](#):

Other Similar products are found below :

[MCF51AC256AVFUE](#) [MCF51AC256BCFUE](#) [MCF51AC256BVFUE](#) [MB91F464AAPMC-GSE2](#) [R5S726B0D216FP#V0](#) [MB91F248PFV-GE1](#) [MB91243PFV-GS-136E1](#) [SAK-TC1782F-320F180HR BA](#) [TC364DP64F300WAAKXUMA1](#) [R5F566NNDDFP#30](#)
[R5F566NNDDFC#30](#) [R5F566NNDDBD#20](#) [MC96F8216ADBN](#) [A96G181HDN](#) [A96G140KNN](#) [A96G174FDN](#) [A31G213CL2N](#)
[A96G148KNN](#) [A96G174AEN](#) [AC33M3064TLBN-01](#) [V3s](#) [T3](#) [A40i-H](#) [V526](#) [A83T](#) [R11](#) [V851s](#) [A133](#) [V833](#) [F1C100S](#) [T3L](#) [T507](#) [A33](#)
[A63](#) [T113-i](#) [H616](#) [V853](#) [V533](#) [R16-J](#) [V536-H](#) [A64-H](#) [V831](#) [V3LP](#) [T113-S3](#) [F1C200S](#) [F133-A](#) [R128-S2](#) [D1-H](#) [ADUCM360BCPZ128-TR](#)
[APT32S003F8PT](#)