

1T 8051

8位微控制器

NuMicro[®] 家族
MS51系列
MS51DA9AE
MS51BA9AE
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	11
2 特性	12
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 8K 系列	16
4.1.1 TSSOP 14脚封装引脚信息	16
4.1.2 MSOP 10脚封装引脚信息	16
4.2 MS51 8K 系列引脚描述	17
5 功能框图	19
5.1 MS51 系列框图	19
6 内存架构	20
6.1 程序内存	20
6.2 数据内存	21
6.3 片上XRAM	23
6.4 非易失性数据存储	23
6.5 安全保护存储器(SPROM)	24
7 特殊功能寄存器(SFR)	25
7.1 SFR页选择	25
7.2 SFR地址列表	26
7.3 SFR 位地址及复位值	27
7.4 所有SFR 介绍	34
8 普通 80C51 系统控制	87
9 I/O 端口结构和工作模式	91
9.1 GPIO 模式	91
9.2 准双向模式	91
9.3 推挽模式	91
9.4 输入模式	93
9.5 开漏模式	93

9.6 读-修改-写 指令	93
9.7 I/O 口控制寄存器	94
9.7.1 输入输出数据控制	94
9.7.2 GPIO 模式控制	95
9.7.3 输入类型	96
9.7.4 输出斜率控制	97
10 定时器/计数器 0 和 1	98
10.1 定时器模式	98
10.1.1 模式 0 (13-位 定时器)	98
10.1.2 模式 1 (16-位 定时器)	99
10.1.3 模式 2 (8-位 自动重载定时器)	99
10.1.4 模式 3 (两组独立 8-位 定时器)	100
10.2 定时器0和1控制寄存器	101
11 定时器 2 和输入捕获	105
11.1 定时器2结构框图	105
11.2 自动重载模式	106
11.3 比较模式	107
11.4 输入捕获模块	107
11.5 定时器2控制寄存器	108
12 定时器 3	115
12.1 定时器3结构描述	115
12.2 定时器3控制寄存器	115
13 看门狗定时器 (WDT)	117
13.1 看门狗定时器结构描述	117
13.2 超时复位定时器	118
13.3 通用定时器	118
13.4 看门狗控制寄存器	119
14 自唤醒定时器 (WKT)	121
14.1 WKT 结构描述	121
14.2 WKT 控制寄存器	121
15 串口 (UART0 & UART1)	123
15.1 功能描述	123
15.1.1 工作模式	123
15.1.2 波特率	125

15.1.3 帧错误检测.....	129
15.1.4 多机通讯.....	129
15.1.5 自动地址识别.....	129
15.2 串口控制寄存器.....	131
16 串行外围总线(SPI).....	137
16.1 结构功能图.....	137
16.2 功能描述.....	138
16.2.1 操作模式.....	139
16.2.2 时钟格式和数据传输.....	139
16.2.3 从机选择引脚SS配置	142
16.2.4 模式故障侦测.....	142
16.2.5 写冲突错误.....	142
16.2.6 溢出错误.....	142
16.2.7 SPI中断	143
16.3 SPI 控制寄存器	144
17 I²C 总线(I²C).....	147
17.1 功能描述.....	147
17.1.1 开始和停止条件.....	147
17.1.2 7位地址数据格式	148
17.1.3 应答Ack.....	149
17.1.4 仲裁	149
17.1.5 操作模式.....	150
17.1.6 从机模式.....	152
17.1.7 广播呼叫模式.....	153
17.1.8 I ² C超时溢出	154
17.1.9 I ² C中断	155
17.1.10 各状态码表述	155
17.2 I ² C控制寄存器	157
17.1 I ² C 中断服务程序典型结构范例	161
18 管脚中断	164
18.1 管脚中断概述.....	164
18.2 管脚中断控制寄存器.....	165
19 脉冲宽度调制 (PWM).....	167
19.1 功能描述.....	167
19.1.1 脉冲宽度调制发生器.....	167
19.1.2 PWM类型.....	170
19.1.3 工作模式.....	171

19.1.4屏蔽输出控制.....	172
19.1.5故障刹车.....	172
19.1.6极性控制.....	172
19.1.7PWM 中断.....	173
19.2 PWM 控制寄存器.....	174
20 12位模数转换器(ADC).....	181
20.1 功能描述.....	182
20.1.1ADC 工作方式.....	182
20.1.2外部触发ADC.....	183
20.1.3ADC 转换结果比较器.....	183
20.1.4带隙电压 (Band-gap) 输入应用.....	184
20.2 ADC控制寄存器.....	185
21 时效访问保护 (TA).....	190
22 中断系统.....	192
22.1 中断概述.....	192
22.2 中断使能.....	192
22.3 中断优先级.....	193
22.4 中断服务.....	194
22.5 中断延迟.....	195
22.6 外部中断.....	195
22.7 中断控制寄存器.....	195
23 在应用编程 (IAP).....	201
23.1 IAP 命令.....	201
23.2 IAP控制寄存器.....	202
23.3 IAP用户指南.....	206
23.4 使用Flash存储器作为数据存储.....	206
23.5 在线系统编程 (ISP).....	209
24 电源管理.....	214
24.1 空闲模式.....	215
24.2 掉电模式.....	215
25 时钟系统.....	216
25.1 时钟系统结构.....	216
25.2 系统时钟源.....	216

25.3	内部振荡器.....	217
25.4	系统时钟切换.....	218
25.5	系统时钟除频.....	220
25.6	系统时钟输出.....	221
26	电源监控.....	222
26.1	上电复位(POR).....	222
26.2	欠压检测(BOD).....	223
27	复位.....	227
27.1	上电复位.....	227
27.2	欠压复位.....	228
27.3	外部复位.....	229
27.4	看门狗复位.....	230
27.5	软件复位.....	231
27.6	启动选择.....	232
27.7	复位状态.....	233
28	辅助功能.....	234
28.1	双 DPTR.....	234
28.2	96位序列号(UID).....	235
29	片上调试 (OCD).....	236
29.1	功能描述.....	236
29.2	OCD限制条件.....	236
30	在线电路编程 (ICP).....	238
31	指令集.....	239
31.1	指令集概述.....	239
31.2	指令集表.....	240
32	配置字.....	243
32.1	CONFIG0.....	244
32.2	CONFIG1.....	245
32.3	CONFIG2.....	246
32.4	CONFIG4.....	247
33	封装信息.....	248
33.1	TSSOP 14-pin (4.4 x 5.5 x 1.2 mm).....	248

33.2 MSOP 10-pin (3.0 x 3.0 x 1.1 mm) 249

34 历史版本信息 250

图集

图 4.1- 1 TSSOP-14 封装引脚信息	16
图 4.1- 2 MSOP-10 封装引脚信息	16
图 5.1- 1 结构框图	19
图 6.1- 1 MS51 8K Flash 系列程序内存分配图	21
图 6.2- 1 数据内存分配图	21
图 6.2- 2 内部 256 字节 RAM 地址	22
图 6.5- 1 SPROM 存储器及 SPROM 保护模式	24
图 9.2- 1 准双向模式结构	91
图 9.3- 1 推挽模式结构	92
图 9.4- 1 输入模式结构	93
图 9.5- 1 开漏模式结构	93
图 10.1- 1 定时器/计数器 0 和 1 的模式 0	98
图 10.1- 2 定时器/计数器 0 和 1 的模式 1	99
图 10.1- 3 定时器/计数器 0 和 1 的模式 2	99
图 10.1- 4 定时器/计数器 0 的模式 3	100
图 11.1- 1 定时器 2 框图	105
图 11.2- 1 定时器 2 自动重载模式和输入捕获模块功能框图	106
图 11.3- 1 定时器 2 比较模式和输入捕获模块功能框图	107
图 12.1- 1 定时器 3 框图	115
图 13.2- 1 WDT 作为超时复位定时器	118
图 13.3- 1 看门狗定时器框图	118
图 14.1- 1 自唤醒定时器框图	121
图 15.1- 1 串口模式 0 时序图	123
图 15.1- 2 串口模式 1 的时序图	124
图 15.1- 3 串口模式 2 模式 3 的传输时序	125
图 16.1- 1 SPI 结构功能图	137
图 16.2- 1 SPI 多主机，多从机连接图	138
图 16.2- 2 SPI 单主机，单从机连接图	139
图 16.2- 3 SPI 时钟格式	140
图 16.2- 4 SPI 时钟和数据格式 (CPHA = 0)	141
图 16.2- 5 SPI 时钟和数据格式 (CPHA = 1)	141
图 16.2- 6 表示接收数据与溢出错误之间的关系	142
图 16.2- 7 SPI 中断请求	143

图 17.1-1 I ² C总线连接	147
图 17.1-2 I ² C总线协议	147
图 17.1-3 起始条件, 重复起始条件和停止条件	148
图 17.1-4 主机发送数据给一个7位地址从机	148
图 17.1-5 主机读取数据从一个7位地址从机	148
图 17.1-6 一个传输中的I2C数据格式	149
图 17.1-7 应答位	149
图 17.1-8 两台主机仲裁过程	150
图 17.1-9 根据当前I ² C 状态来控制I ² C总线	151
图 17.1-10 主机发送模式流程和寄存器状态	151
图 17.1-11 主机接收模式流程和寄存器状态	152
图 17.1-12 从机接收模式流程和寄存器状态	153
图 17.1-13 广播呼叫模式流程和寄存器状态	154
图 17.1-14 I ² C 超时溢出计数器	155
图 17.2-1 SCL保持时间延伸使能	158
图 18.1-1 管脚中断模块框图	164
图 19.1-1 PWM 结构框图	168
图 19.1-2 PWM和故障刹车输出控制方框图	169
图 19.1-3 PWM 边沿对齐类型波形	170
图 19.1-4 PWM中心对齐类型波形	171
图 19.1-5 带死区时间插入的PWM互补模式	172
图 19.1-6 故障刹车功能方框图	172
图 19.1-7 PWM中断类型	173
图 20.1-1 12-位 ADC 模块结构图	182
图 20.1-2 外部触发 ADC电路结构	183
图 20.1-3 ADC 结果比较器	184
图 25.1-1 时钟系统框图	216
图 26.2-1 欠压检测框图	223
图 27.6-1 启动选择结构图	232
图 32.1-1 CONFIG0 复位后自动重载	244
图 32.3-1 CONFIG2 上电复位重载位	246
图 33.1-1 TSSOP14 封装信息	248
图 33.2-1 MSOP10 封装信息	249

表集

表 7.2- 1 特殊功能寄存器地址..... 26

表 7.3- 1 SFR 定义及复位值 33

表 7.4- 1指令对标志位的影响..... 90

表 9.7- 1 配置不同的I/O模式 95

表 13.1- 1看门狗在不同分频下定时器溢出间隔列表 117

表 15.1- 1串口 Port 0 模式 / 波特率描述 126

表 15.1- 2串口 1 模式 / 波特率描述..... 127

表 16.3- 1从机选择脚定义 145

表 22.1- 1中断向量表 192

表 22.3- 1中断优先级设置 193

表 22.3- 2各级中断源特性表 194

表 23.1- 1 IAP 模式及命令表..... 201

表 26.2- 1 BOF 复位值 224

表 26.2- 2 最小欠压检测脉冲宽度 226

表 31.1- 1所有指令， 指令集及寻址模式注释..... 239

表 31.2- 1指令集..... 242

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 8K系列内嵌8K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LDROM，该区域通常存放用于执行在系统编程（ISP）的引导代码（boot code），LDROM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 8K系列提供丰富的特殊功能模块，包括：256字节SRAM，1K字节XRAM。最多可达18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，6 通道增强型PWM输出，8路12位ADC。上述功能对应产生18个中断源，具有4级中断优先级配置。

MS51 8K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

● CPU:

- 全静态8位1T 8051内核CMOS微控制器
- 指令集全兼容MCS-51
- 4级优先级中断配置
- 双数据指针(DPTRs)

● 工作条件:

- 宽电压工作范围2.4V至5.5V
- 宽工作频率最高至24MHz
- 工业级工作温度 -40°C 至 +105°C

● 存储器:

- 8K字节APROM用户程序代码区
- 可配置4K/3K/2K/1K字节LDROM引导代码区，用户可灵活配置用途
- 所有FLASH区域分隔为128字节一页
- 内建IAP编程功能
- 代码加密功能
- 256字节片内直接存取RAM
- 额外1K字节片内间接存取RAM(XRAM)通过MOVX指令读写

● 时钟源:

- 24MHz高速内部振荡器(HIRC) $\pm 1\%$ 精度等级 (25 °C, 3.3 V)全工作条件范围下精度 $\pm 2\%$ 精度等级
- 内部高速振荡器(HIRC)可配置为24MHz
- 10kHz低速内部振荡器(LIRC) $\pm 1\%$ 精度等级 (25 °C, 3.3 V)。
- 支持外部时钟输入.
- 支持系统时钟即时软件切换(On-the-fly)功能.
- 支持软件配置时钟除频最高至1/512.

● 功能:

- 标准外部中断脚 $\overline{\text{INT0}}$ 及 $\overline{\text{INT1}}$
- 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位自动重载功能定时器3，可用于配置串行口UART的波特率
- 一组16位PWM计数中断
- 一组看门狗(WDT)，由内部10 kHz独立时钟作为时钟源
- 一组自唤醒功能定时器(WKT)，用于低功耗模式下自主唤醒
- 两组全双工串口，带有帧错误检测及自动地址辨识功能。UART0的TXD及RXD脚可通过软件更换管脚位置
- 一组SPI总线，当系统时钟是16 MHz时，主机模式及从机模式最高传输速率皆可达到8 Mbps
- 一组I2C总线，主机模式及从机模式最高传输速率皆可达到400 kbps
- 三对，6通道脉宽调制器(PWM)，10个输出管脚可以选择，16位分辨率，带有不同的工作模式和故障刹车(Fault Brake)功能
- 最多可配置8信道管脚中断功能，所有的I/O端口都支持此功能，可通过软件配置边沿或电平触发
- 一组12位ADC，最高380ksps采样率
- 电源管理模块：
 - 两种省电模式：空闲模式及掉电模式
- 电源监控：
 - 欠压检测(BOD)用于侦测系统供电低电压，4级电压选择，可配置中断或复位响应
 - 上电复位(POR)
 - 低电压复位(LVR)
- 强效ESD及EFT能力.
 - ESD HBM 通过 8 kV
 - EFT $> \pm 4.4$ kV
 - 门锁测试通过150 mA
- 开发工具：
 - Nuvoton Nu-Link 基于 KEIL™ 和 IAR 开发环境.
 - Nuvoton 电路编程 (Nu-Link).
 - Nuvoton 在系统编程 (ISP) 通过 UART.

3 料号信息

3.1 封装类型

	MSOP10	TSSOP14	QFN20 ^[3]	TSSOP20	TSSOP28	LQFP32	QFN33
料号	MS51BA9AE	MS51DA9AE	MS51XB9AE MS51XB9BE MS51XC0AE	MS51FB9AE MS51FC0AE	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 ^[2]	UART	SPI	I ² C		
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	3	2	1	1	10-ch	TSSOP20
MS51XC0BE	32	2	4	18	4	8	3	2	1	1	10-ch	QFN20
MS51EC0AE	32	2	4	26	4	10	3	2	1	1	15-ch	TSSOP28
MS51PC0AE	32	2	4	30	4	12	3	2	2	1	15-ch	LQFP32
MS51TC0AE	32	2	4	30	4	12	3	2	2	1	15-ch	QFN33

注:

- LDROM 由 APROM 独立出 4/3/2/1KB Flash 区域，可用于进行 ISP 动作。
- ISO-7816 可配置为 UART2~4。
- QFN20 有两种封装尺寸，具体尺寸差异请看章节 33。
- 本技术参考手册仅针对 8K 及 8K Flash 料号进行具体描述。

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者使用 [NuTool - PinConfig](#). NuTool - PinConfigure 包含所有 NuMicro® 家族芯片系列的所有型号, 帮助用户方便正确的配置GPIO的多功能引脚.

4.1 MS51 8K 系列

4.1.1 TSSOP 14脚封装引脚信息

相关料号: MS51DA9AE

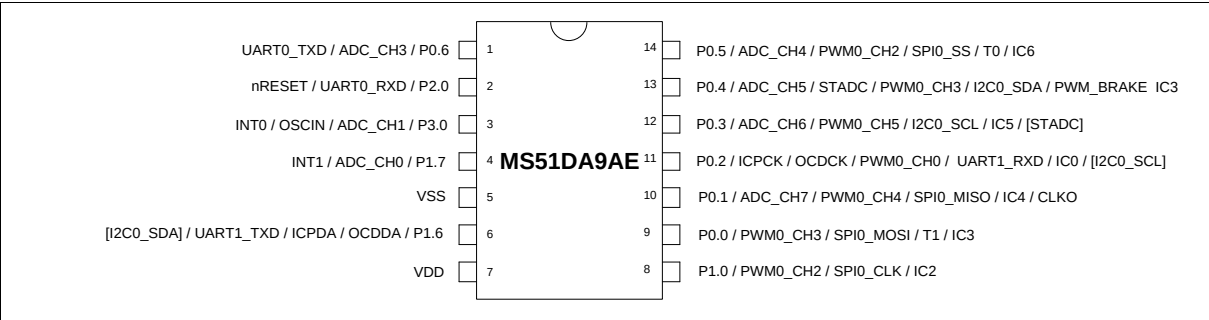


图 4.1- 1 TSSOP-14 封装引脚信息

4.1.2 MSOP 10脚封装引脚信息

相关料号: MS51BA9AE

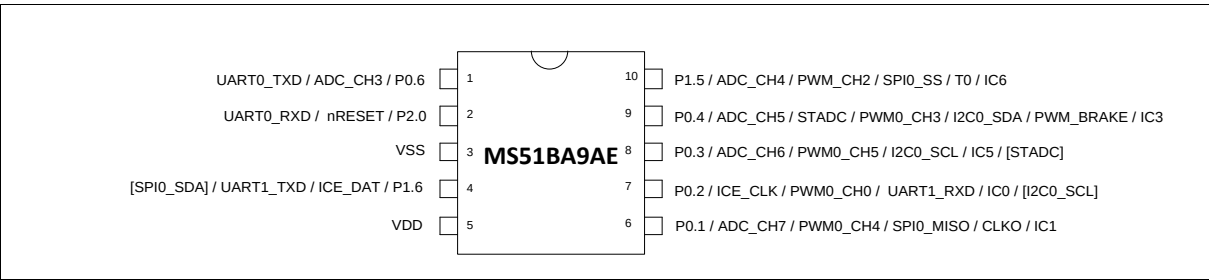


图 4.1- 2 MSOP-10 封装引脚信息

4.2 MS51 8K 系列引脚描述

引脚编号		符号	功能描述 ^[1]
MS51DA9AE	MS51BA9AE		
7	3	VDD	电源正端
5	5	VSS	电源负端
9	-	P0.0	端口0管脚0
		PWM0_CH3	PWM 输出通道3
		SPI0_MOSI	SPI 主机输出/从机输入脚
		IC3	定时器输入捕获通道3
		T1	定时器/计数器1，外部计数输入脚或翻转输出脚
10	6	P0.1	端口0管脚1
		ADC_CH7	ADC输入通道7
		PWM0_CH4	PWM 输出通道4
		SPI0_MISO	SPI 主机输入/从机输出脚
		IC4	定时器输入捕获通道4
		CLKO	系统时钟输出
11	7	P0.2	端口0管脚2
		ICE_CLK	仿真及ICP编程时钟输入脚。
		PWM0_CH0	PWM 输出通道0
		UART1_RXD ^[6]	串口1数据输入脚
		IC0	定时器输入捕获通道0
		[I2C0_SCL]	I2C 时钟脚
12	8	P0.3	端口0管脚3
		ADC_CH6	ADC输入通道6
		PWM0_CH5/	PWM 输出通道5
		I2C0_SCL	I2C 时钟脚
		IC5/	定时器输入捕获通道5
		[STADC]	外部启动ADC触发脚
13	9	P0.4	端口0管脚4
		ADC_CH5	ADC输入通道5
		PWM0_CH3	PWM 输出通道3
		PWM0_BRAKE	故障刹车输入脚
		I2C0_SDA	I2C 数据脚
		IC3	定时器输入捕获通道3
		STADC	外部启动ADC触发脚
14	10	P0.5	端口0管脚5
		ADC_CH4	ADC输入通道4
		PWM0_CH2	PWM 输出通道2
		SPI0_SS	SPI0 从机选择输入脚
		IC6	定时器输入捕获通道6
		T0	定时器/计数器0，外部计数输入脚或翻转输出脚
1	1	P0.6	端口0管脚6
		ADC_CH3	ADC 输入通道3
		UART0_TXD ^[2]	串口0数据发送脚
8	-	P1.0	端口1管脚0
		PWM0_CH2/	PWM 输出通道2

引脚编号		符号	功能描述 ^[1]
MS51DA9AE	MS51BA9AE		
6	4	SPI0_CLK	SPI 时钟脚
		IC2	定时器输入捕获通道2
		P1.6	端口1管脚6
		ICE_DAT	ICP 编程数据输入输出脚
		UART1_TXD ^[6]	串口1数据发送脚
4	-	[I2C0_SDA]	I2C 数据脚
		P1.7	端口1管脚7
		ADC_CH0	ADC 输入通道0
		INT1	外部中断1输入
2	2	P2.0	端口2管脚0， 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，即可稳定工作。
		UART0_RXD ^[2]	串口0数据接收脚
3	-	P3.0	端口3管脚0，使用内部晶振时可用
		ADC_CH1	ADC 输入通道1
		INT0	外部中断0输入
		OSCIN	使用 ECLK 模式，OSCIN为外部时钟输入脚。

注：

1. 所有管脚都可以配置为外部中断输入脚，该功能未列入管脚描述列表。详见章节16.管脚中断

2. UART0的 TXD 及 RXD 管脚可通过配置寄存器UART0PX (AUXR1.2)交换位置

3. [I2C]备用功能重分配选项，I2C管脚可通过配置寄存器 I2CPX (I2CON.0)转换位置

4. [STADC] 备用功能重分配选项。STADC 引脚可通过配置寄存器STADCPX(ADCCON1.6)转换位置

5. PIOx 寄存器决定哪一个管脚是PWM或GPIO功能

6 UART1的 TXD 及 RXD 管脚可通过配置寄存器UART1PX (AUXR1.1)交换位置

5 功能框图

5.1 MS51 系列框图

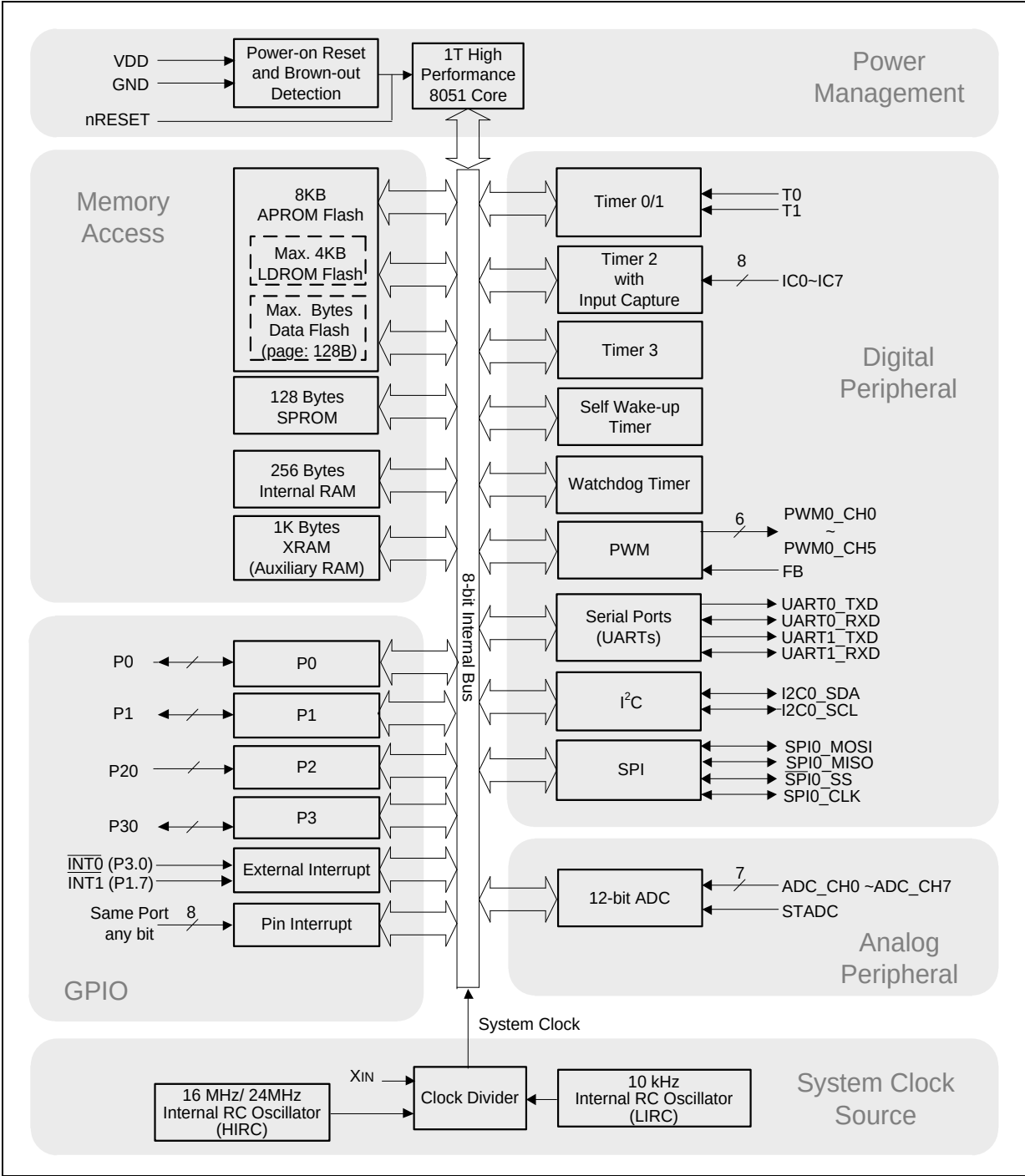


图 5.1- 1 结构框图

6 内存架构

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。对于需要更多内部RAM的许多应用，MS51提供另外片上768字节RAM，叫做XRAM，通过MOVX指令访问。

整个嵌入的FLASH，作为编程内存的功能，被分成三块。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 程序内存

程序内存存储用于执行的编程代码。如图 6.1- 1，在任何复位之后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8K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。

另外单独的编程块叫做LDROM，它的功能通常是存储启动代码用于ISP。它可以更新APROM空间和CONFIG字节。APROM中的代码也可以重新编程LDROM。APROM和LDROM关于IAP的细节和配置位设置，请看章节23在应用编程 (IAP)。注意APROM 和 LDROM 是硬件独立模块，因此如果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		

出厂预设值: 1111 1111b

位	名称	描述
2:0	LDSIZE[2:0]	LDROM 容量选择 111 = 无 LDROM. APROM 为 8K 字节. 110 = LDROM 为 1K 字节. APROM 为 7K 字节. 101 = LDROM 为 2K 字节. APROM 为 6K 字节. 100 = LDROM 为 3K 字节. APROM 为 5K 字节. 0xx = LDROM 为 4K 字节. APROM 为 4K 字节.

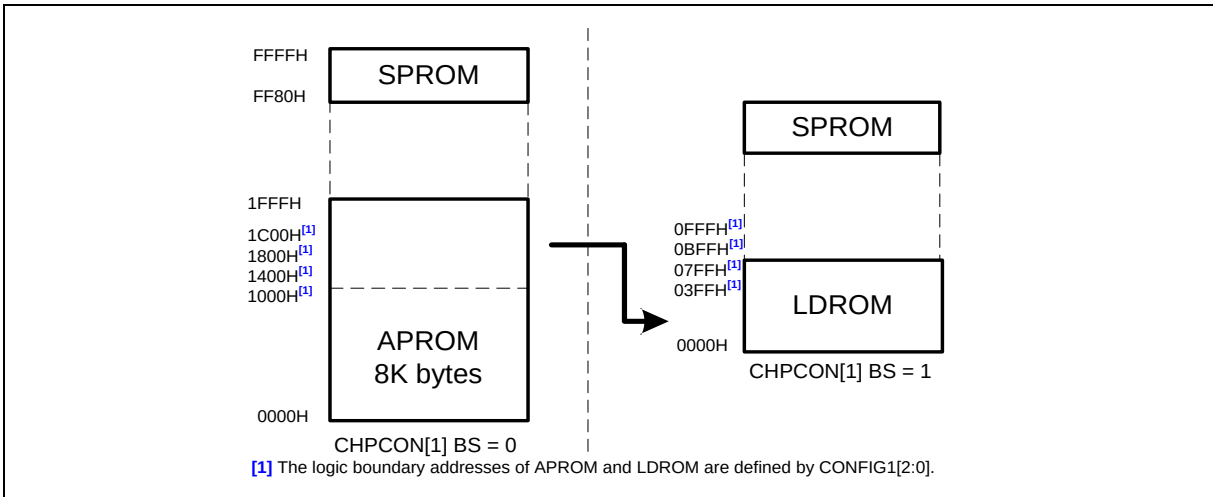


图 6.1- 1 MS51 8K Flash 系列程序内存分配图

6.2 数据内存

图 6.2- 1所示MS51中可用内部数据内存。内部数据内存占用一个独立于编程内存的地址空间。内部数据内存可以分割成三块。分别是RAM的低128字节，RAM的高128字节，和SFR空间的128字节。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。

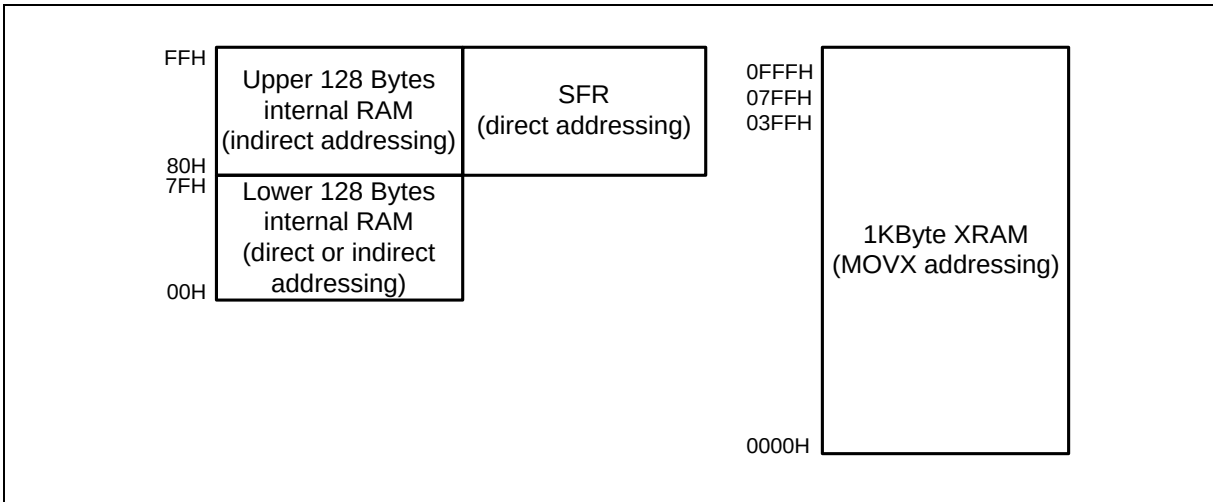


图 6.2- 1 数据内存分配图

对于整个256字节的内部RAM，另外一个应用是用于堆栈。这个区域通过堆栈指针(SP)来选择，SP存储

的是栈顶的地址。当CALL、JMP或中断被调用，返回的地址就存在堆栈里面。没有限制堆栈从RAM的什么地方开始。默认情况下，在复位后堆栈指针为07H。用户可以改变该地址为任何值。SP会指向最后使用的值。因此SP会增加，然后地址保存到堆栈中。相反的，当堆栈的内容出栈，SP会递减。

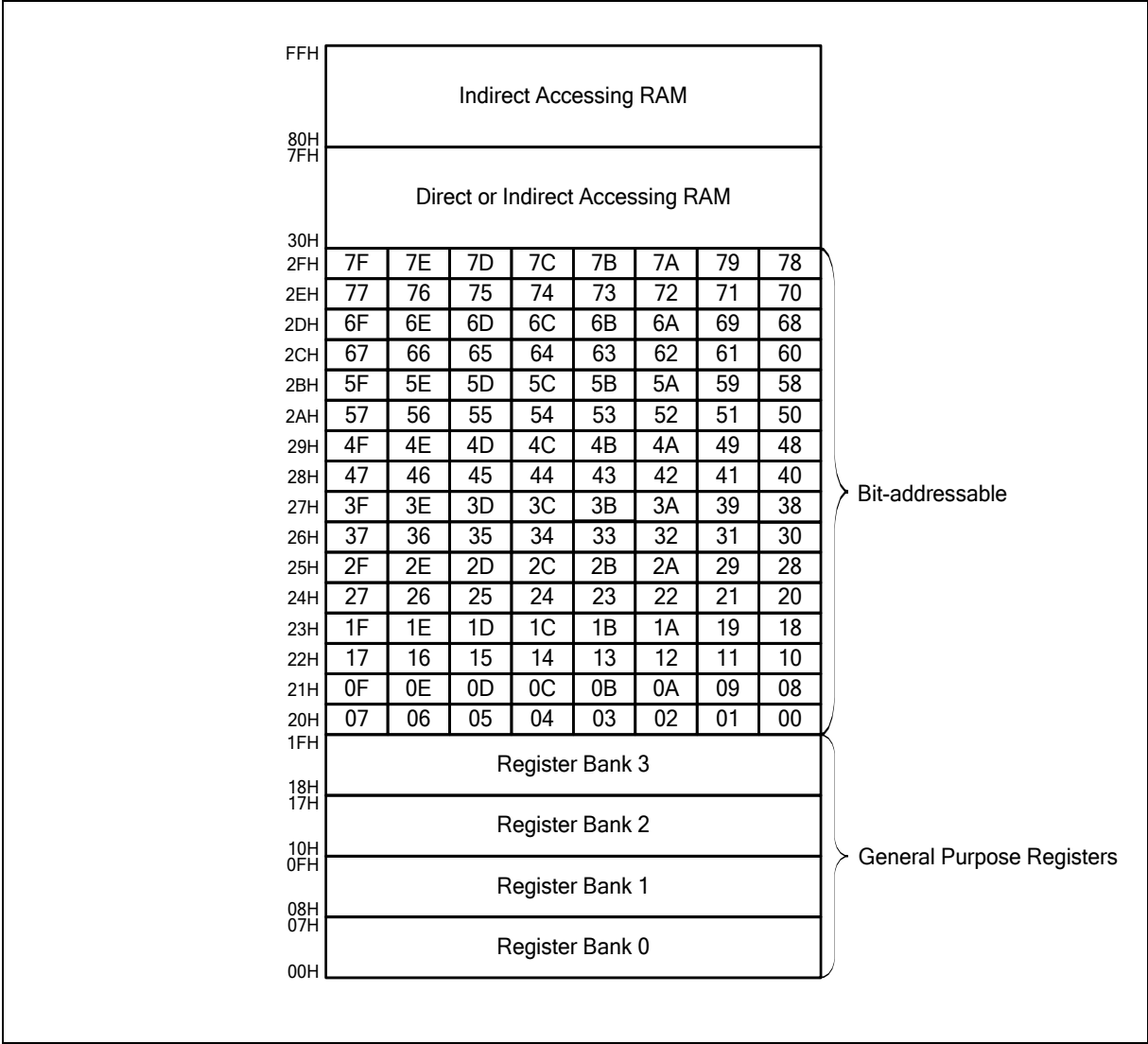


图 6.2- 2 内部 256 字节 RAM 地址

6.3 片上XRAM

MS51提供额外的片上768字节附加的RAM 叫做XRAM来扩大RAM的空间。它占用地址空间从00H到FFH。这768字节的XRAM通过调用外部指令MOVX @DPTR 或 MOVX @Ri. (看下面示例代码)间接访问。注意堆栈指针不能位于XRAM的任何区域。

XRAM 读写汇编范例:

```
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
```

6.4 非易失性数据存储

通过使用IAP, APROM或LDROM任何页都可以用来作为非可变数据存储, IAP的细节, 详见章节201在应用编程 (IAP)

6.5 安全保护存储器(SPROM)

安全保护存储器（SPROM）用于应用程序的保密。SPROM包含128字节，起始地址从FF80H ~ FFFFH，不支持整片擦除命令。图 6.5- 1表示SPROM的最后一个字节(地址: FFFFH)用于标识SPROM中的程序是非安全，还是安全模式。

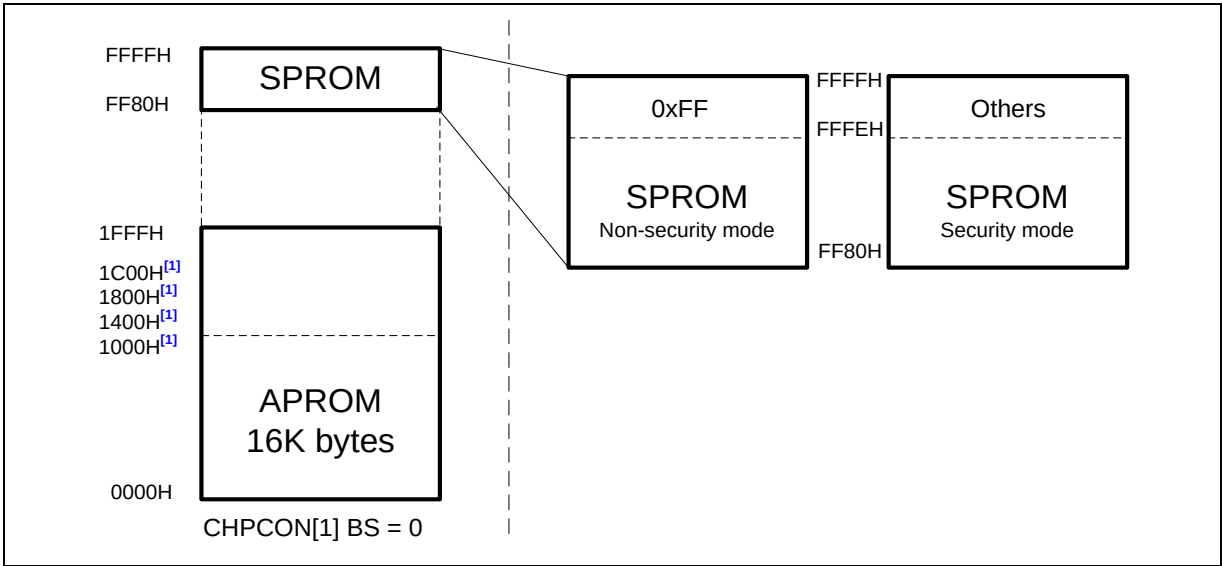


图 6.5- 1 SPROM 存储器及 SPROM 保护模式

1. SPROM 非安全模式 (最后一个字节是0xFF)， SPROM的操作跟APROM、LDROM是一样的。所有的区域都可以被CPU或者ISP读取，也能够通过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IAP读或写SPROM的命令, 仅支持用IAP整页擦除SPROM命令。

7 特殊功能寄存器(SFR)

MS51用特殊功能寄存器（SFRs）来控制或监视外设和其它模块。SFRs 位于地址80到FFH地址空间，仅可以通过直接寻址访问。那些地址以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：

7.1 SFR页选择

为了在地址0x80 到 0xFF之间提供多于128字节的SFRs，补充了SFR页。默认情况下，所有SFR访问目标都是SFR页0。在设备初始化过程中位于SFR页1的地址可能需要去访问。寄存器SFRS用来切换SFR页地址。注意这个寄存器是有TA写保护的，大部分可用的SFRs都在SFR页0和页1中。

SFRS – SFR页选择

寄存器	地址，页，特别说明	复位值
SFRS	91H, 所有页，TA保护	0000_0000b

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

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

切换 SFR 页参考例程:

MOV	TA, #0AAH	;switch to SFR page 1
MOV	TA, #55H	
MOV	SFRS, #01H	
MOV	TA, #0AAH	;switch to SFR page 0
MOV	TA, #55H	
MOV	SFRS, #00H	

7.2 SFR地址列表

	地址	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0 1	F8	SCON_1	PDTEN	PDTCNT	PMEN	PMD	PORDIS -	EIP1 -	EIPH1 -
0 1	F0	B	CAPCON3	CAPCON4	SPCR SPCR2	SPSR	SPDR -	AINDIDS -	EIPH -
0 1	E8	ADCCON0	PICON	PINEN	PIPEN	PIF	C2L	C2H	EIP -
0 1	E0	ACC	ADCCON1	A[7]DCCON2	ADCDLY	C0L	C0H	C1L	C1H
0 1	D8	PWMCON0	PWMPL	PWM0L	PWM1L	PWM2L	PWM3L	PIOCON0	PWMCON1
0 1	D0	PSW	PWMPH	PWM0H	PWM1H	PWM2H	PWM3H	PNP	FBD
0 1	C8	T2CON	T2MOD	RCMP2L	RCMP2H	TL2 PWM4L	TH2 PWM5L	ADCMPL	ADCMPL
0 1	C0	I2CON	I2ADDR	ADCRL	ADCRH	T3CON PWM4H	RL3 PWM5H	RH3 PIOCON1	TA
0 1	B8	IP	SADEN	SADEN_1	SADDR_1	I2DAT	I2STAT	I2CLK	I2TOC
0 1	B0	P3	P0M1 P0S	P0M2 P0SR	P1M1 P1S	P1M2 P1SR	P2S	-	IPH PWMINTC
0 1	A8	IE	SADDR	WDCON	BODCON1	P3M1 P3S	P3M2 P3SR	IAPFD	IAPCN
0 1	A0	P2	-	AUXR1	BODCON0	IAPTRG	IAPUEN	IAPAL	IAPAH
0 1	98	SCON	SBUF	SBUF_1	EIE	EIE1	RSR	-	CHPCON
0 1	90	P1	SFRS	CAPCON0	CAPCON1	CAPCON2	CKDIV	CKSWT	CKEN
0 1	88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	WKCON
0 1	80	P0	SP	DPL	DPH	RCTRIM0	RCTRIM1	RWK	PCON

注: SFR空间中未使用地址用‘-’标记, 访问改写这些地址字节会有不可预知的影响, 请避免这种情况。

表 7.2- 1 特殊功能寄存器地址

7.3 SFR 位地址及复位值

符号	定义	地址 (页)	MSB								LSB ^[1]		复位值 ^[2]
EIPH1	Extensive interrupt priority high 1	FFH (0)	-	-	-	-	-	PWKTH	PT3H	PSH_1	0000 0000b		
EIP1	Extensive interrupt priority 1	FEH (0)	-	-	-	-	-	PWKT	PT3	PS_1	0000 0000b		
PMD	PWM mask data	FCH	-	-	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0	0000 0000b		
PMEN	PWM mask enable	FBH	-	-	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0	0000 0000b		
PDTCNT ^[4]	PWM dead-time counter	FAH	PDTCNT[7:0]								0000 0000b		
PD TEN ^[4]	PWM dead-time enable	F9H	-	-	-	PDTCNT. 8	-	PDT45E N	PDT23E N	PDT01EN	0000 0000b		
SCON_1	Serial port 1 control	F8H	SM0_1/ FE_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1	0000 0000b		
EIPH	Extensive interrupt priority high	F7H	PT2H	PSPIH	PFBH	PWDTH	PPWMH	PCAPH	PPIH	PI2CH	0000 0000b		
AINDIDS	ADC channel digital input disable	F6H	P11DIDS	P03DIDS	P04DIDS	P05DIDS	P06DID S	P07DIDS	P30DIDS	P17DIDS	0000 0000b		
SPDR	SPI data	F5H (0)	SPDR[7:0]								0000 0000b		
SPSR	SPI status	F4H	SPIF	WCOL	SPIOVF	MODF	DISMOD F	TXBUF	-	-	0000 0000b		
SPCR	SPI control	F3H (0)	SSOE	SPIEN	LSBFE	MSTR	CPOL	CPHA	SPR[1:0]		0000 0000b		
SPCR2	SPI control 2	F3H (1)	-	-	-	-	-	-	SPIS[1:0]		0000 0000b		
CAPCON4	Input capture control 4	F2H	-	-	-	-	CAP23	CAP22	CAP21	CAP20	0000 0000b		
CAPCON3	Input capture control 3	F1H	CAP13	CAP12	CAP11	CAP10	CAP03	CAP02	CAP01	CAP00	0000 0000b		
B	B register	F0H	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	0000 0000b		
EIP	Extensive interrupt priority	EFH	PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI2C	0000 0000b		
C2H	Input capture 2 high byte	EEH	C2H[7:0]								0000 0000b		
C2L	Input capture 2 low byte	EDH	C2L[7:0]								0000 0000b		
PIF	Pin interrupt flag	ECH	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	0000 0000b		
PIPEN	Pin interrupt high level/rising edge enable	EBH	PIPEN7	PIPEN6	PIPEN5	PIPEN4	PIPEN3	PIPEN2	PIPEN1	PIPEN0	0000 0000b		
PINEN	Pin interrupt low level/falling edge enable	EAH	PINEN7	PINEN6	PINEN5	PINEN4	PINEN3	PINEN2	PINEN1	PINEN0	0000 0000b		
PICON	Pin interrupt	E9H	PIT67	PIT45	PIT3	PIT2	PIT1	PIT0ADC	PIPS[1:0]		0000 0000b		

符号	定义	地址 (页)	MSB								LSB ^[1]	复位值 ^[2]
	control											
ADCCON0	ADC control 0	E8H	ADCF	ADCS	ETGSEL1	ETGSEL0	ADCHS ₃	ADCHS2	ADCHS1	ADCHS0	0000 0000b	
C1H	Input capture 1 high byte	E7H	C1H[7:0]								0000 0000b	
C1L	Input capture 1 low byte	E6H	C1L[7:0]								0000 0000b	
C0H	Input capture 0 high byte	E5H	C0H[7:0]								0000 0000b	
C0L	Input capture 0 low byte	E4H	C0L[7:0]								0000 0000b	
ADCDLY	ADC trigger delay	E3H	ADCDLY[7:0]								0000 0000b	
ADCCON2	ADC control 2	E2H	ADFBEN	ADCMPO P	ADCMPE N	ADCMPO	-	-	-	ADCDLY.8	0000 0000b	
ADCCON1	ADC control 1	E1H	OCEN	STADCP X	ADCDIV[1:0]		ETGTYP[1:0]		ADCEX	ADCEN	0000 0000b	
ACC	Accumulator	E0H	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	0000 0000b	
PWMCON1	PWM control 1	DFH	PWMMOD[1:0]		GP	PWMTYP P	FBINEN	PWMDIV[2:0]			0000 0000b	
PIOCON0	PWM I/O switch 0	DEH	-	-	PIO05	PIO04	PIO03	PIO02	PIO01	PIO00	0000 0000b	
PWM3L	PWM0 channel 3 duty low byte	DDH	PWM3[7:0]								0000 0000b	
PWM2L	PWM0 channel 2 duty low byte	DCH	PWM2[7:0]								0000 0000b	
PWM1L	PWM0 channel 1 duty low byte	DBH	PWM1[7:0]								0000 0000b	
PWM0L	PWM0 channel 0 duty low byte	DAH	PWM0[7:0]								0000 0000b	
PWMPL	PWM period low byte	D9H	PWMP[7:0]								0000 0000b	
PWMCON0	PWM0 control register 0	D8H	PWMRUN	LOAD	PWMF	CLRPM M	-	-	-	-	0000 0000b	
FBD	Brake data	D7H	FBF	FBINLS	FBD5	FBD4	FBD3	FBD2	FBD1	FBD0	0000 0000b	
PNP	PWM negative polarity	D6H	-	-	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0	0000 0000b	
PWM3H	PWM0 channel 3 duty high byte	D5H	PWM3[15:8]								0000 0000b	
PWM2H	PWM0 channel 2 duty high byte	D4H	PWM2[15:8]								0000 0000b	
PWM1H	PWM0 channel 1 duty high byte	D3H	PWM1[15:8]								0000 0000b	
PWM0H	PWM0 channel 0 duty high byte	D2H	PWM0[15:8]								0000 0000b	
PWMPH	PWM period high	D1H	PWMP[15:8]								0000 0000b	

符号	定义	地址 (页)	MSB								LSB ^[1]	复位值 ^[2]
	byte											
PSW	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	-	P	0000 0000b	
ADCMPH	ADC compare high byte	CFH	ADCMP[11:4]									0000 0000b
ADCMPL	ADC compare low byte	CEH	-	-	-	-	ADCMP[3:0]					0000 0000b
PWM5L	PWM0 channel 5 duty low byte	CDH (1)	PWM5 [7:0]									0000 0000b
TH2	Timer 2 high byte	CDH (0)	TH2[7:0]									0000 0000b
PWM4L	PWM0 channel 4 duty low byte	CCH (1)	PWM4[7:0]									0000 0000b
TL2	Timer 2 low byte	CCH (0)	TL2[7:0]									0000 0000b
RCMP2H	Timer 2 compare high byte	CBH	RCMP2H[7:0]									0000 0000b
RCMP2L	Timer 2 compare low byte	CAH (0)	RCMP2L[7:0]									0000 0000b
T2MOD	Timer 2 mode	C9H	LDEN	T2DIV[2:0]			CAPCR	CMPCR	LDTS[1:0]			0000 0000b
T2CON	Timer 2 control	C8H	TF2	-	-	-	-	TR2	-	CM/RL2		0000 0000b
TA	Timed access protection	C7H	TA[7:0]									0000 0000b
PIOCON1	PWM I/O switch 1	C6H (1)	-	-	PIO15	-	PIO13	PIO12	PIO11	-		0000 0000b
RH3	Timer 3 reload high byte	C6H (0)	RH3[7:0]									0000 0000b
PWM5H	PWM0 channel 5 duty high byte	C5H (1)	PWM5 [15:8]									0000 0000b
RL3	Timer 3 reload low byte	C5H (0)	RL3[7:0]									0000 0000b
PWM4H	PWM4 duty high byte	C4H (1)	PWM4[15:8]									0000 0000b
T3CON	Timer 3 control	C4H (0)	SMOD_1	SMOD0_1	BRCK	TF3	TR3	T3PS[2:0]				0000 0000b
ADCRH	ADC result high byte	C3H	ADCR[11:4]									0000 0000b
ADCRL	ADC result low byte	C2H	-	-	-	-	ADCR[3:0]					0000 0000b
I2ADDR	I2C own slave address	C1H	I2ADDR[7:1]							GC		0000 0000b
I2CON	I2C control	C0H	-	I2CEN	STA	STO	SI	AA	-	I2CPX		0000 0000b
I2TOC	I2C time-out counter	BFH	-	-	-	-	-	I2TOCEN	DIV	I2TOF		0000 0000b

符号	定义	地址 (页)	MSB								LSB ^[1]	复位值 ^[2]
I2CLK	I2C clock	BEH	I2CLK[7:0]									0000 1001b
I2STAT	I2C status	BDH	I2STAT[7:3]				0	0	0		1111 1000b	
I2DAT	I2C data	BCH	I2DAT[7:0]									0000 0000b
SADDR_1	Slave 1 address	BBH	SADDR_1[7:0]									0000 0000b
SADEN_1	Slave 1 address mask	BAH	SADEN_1[7:0]									0000 0000b
SADEN	Slave 0 address mask	B9H	SADEN[7:0]									0000 0000b
IP	Interrupt priority	B8H	-	PADC	PBOD	PS	PT1	PX1	PT0	PX0	0000 0000b	
PWMINTC	PWM Interrupt Control	B7H (1)	-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0	0000 0000b	
IPH	Interrupt priority high	B7H (0)	-	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H	0000 0000b	
P2S	P20 Setting	B5H	P20UP	-	-	-	T1OE	T0OE	-	P2S.0	0000 0000b	
P1SR	P1 slew rate	B4H (1)	P1SR.7	P1SR.6	P1SR.5	P1SR.4	P1SR.3	P1SR.2	P1SR.1	P1SR.0	0000 0000b	
P1M2	P1 mode select 2	B4H (0)	P1M2.7	P1M2.6	P1M2.5	P1M2.4	P1M2.3	P1M2.2	P1M2.1	P1M2.0	0000 0000b	
P1S	P1 Schmitt trigger input	B3H (1)	P1S.7	P1S.6	P1S.5	P1S.4	P1S.3	P1S.2	P1S.1	P1S.0	0000 0000b	
P1M1	P1 mode select 1	B3H (0)	P1M1.7	P1M1.6	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0	1111 1111b	
P0SR	P0 slew rate	B2H (1)	P0SR.7	P0SR.6	P0SR.5	P0SR.4	P0SR.3	P0SR.2	P0SR.1	P0SR.0	0000 0000b	
P0M2	P0 mode select 2	B2H (0)	P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0	0000 0000b	
P0S	P0 Schmitt trigger input	B1H (1)	P0S.7	P0S.6	P0S.5	P0S.4	P0S.3	P0S.2	P0S.1	P0S.0	0000 0000b	
P0M1	P0 mode select 1	B1H (0)	P0M1.7	P0M1.6	P0M1.5	P0M1.4	P0M1.3	P0M1.2	P0M1.1	P0M1.0	1111 1111b	
P3	Port 3	B0H	0	0	0	0	0	0	0	P3.0	Output latch, 0000 0001b Input 0000 000Xb ^[3]	
IAPCN	IAP control	AFH	IAPA[17:16]		FOEN	FCEN	FCTRL[3:0]				0011 0000b	
IAPFD	IAP flash data	AEH	IAPFD[7:0]									0000 0000b
P3SR	P3 slew rate	ADH (1)	-	-	-	-	-	-	-	P3SR.0	0000 0000b	
P3M2	P3 mode select 2	ADH (0)	-	-	-	-	-	-	-	P3M2.0	0000 0000b	
P3S	P3 Schmitt trigger input	ACH (1)	-	-	-	-	-	-	-	P3S.0	0000 0000b	
P3M1	P3 mode select 1	ACH	-	-	-	-	-	-	-	P3M1.0	0000 0001b	

符号	定义	地址 (页)	MSB								LSB ^[1]		复位值 ^[2]
		(0)											
BODCON1 ^[4]	Brown-out detection control 1	ABH	-	-	-	-	-	LPBOD[1:0]		BODFLT	POR, 0000 0001b Others, 0000 0UUUb		
WDCON	Watchdog Timer control	AAH	WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS[2:0]			POR, 0000 0111b WDT, 0000 1UUUb Others, 0000 UUUUb		
SADDR	Slave 0 address	A9H	SADDR[7:0]										0000 0000b
IE	Interrupt enable	A8H	EA	EADC	EBOD	ES	ET1	EX1	ET0	EX0	0000 0000b		
IAPAH	IAP address high byte	A7H	IAPA[15:8]										0000 0000b
IAPAL	IAP address low byte	A6H	IAPA[7:0]										0000 0000b
IAPUEN ^[4]	IAP update enable	A5H	-	-	-	-	-	CFUEN	LDUEN	APUEN	0000 0000b		
IAPTRG ^[4]	IAP trigger	A4H	-	-	-	-	-	-	-	IAPGO	0000 0000b		
BODCON0 ^[4]	Brown-out detection control 0	A3H	BODEN ^[5]	-	BOV[1:0] ^[5]		BOF ^[6]	BORST ^[5]	BORF	BOS ^[7]	POR, CCCC XC0Xb BOD, UUUU XU1Xb Others, UUUU XUUXb		
AUXR1	Auxiliary register 1	A2H	SWRF	RSTPIN F	HardF	-	GF2	UART0P X	UART1PX	DPS	POR, 0000 0000b Software, 1U00 0000b nRESET pin, U100 0000b Others, UUU0 0000b		
P2	Port 2	A0H	0	0	0	0	0	0	0	P2.0	Output latch, 0000 000Xb Input, 0000 000Xb ^[3]		
CHPCON ^[4]	Chip control	9FH	SWRST	IAPFF	-	-	-	-	BS[5]	IAPEN	Software, 0000 00U0b Others, 0000 00C0b		
EIE1	Extensive interrupt enable 1	9CH	-	-	-	-	-	EWKT	ET3	ES_1	0000 0000b		
EIE	Extensive interrupt enable	9BH	ET2	ESPI	EFB	EWDT	EPWM	ECAP	EPI	EI2C	0000 0000b		
SBUF_1	Serial port 1 data buffer	9AH	SBUF_1[7:0]										0000 0000b
SBUF	Serial port 0 data buffer	99H	SBUF[7:0]										0000 0000b
SCON	Serial port 0 control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000b		

符号	定义	地址 (页)	MSB								LSB ^[1]	复位值 ^[2]
CKEN ^[4]	Clock enable	97H	EXTEN[1:0]		HIRCEN	-	-	-	-	CKSWTF	0011 0000b	
CKSWT ^[4]	Clock switch	96H	-	-	HIRCST	-	ECLKST	OSC[1:0]		-	0011 0000b	
CKDIV	Clock divider	95H	CKDIV[7:0]								0000 0000b	
CAPCON2	Input capture control 2	94H	-	ENF2	ENF1	ENF0	-	-	-	-	0000 0000b	
CAPCON1	Input capture control 1	93H	-	-	CAP2LS[1:0]		CAP1LS[1:0]		CAP0LS[1:0]		0000 0000b	
CAPCON0	Input capture control 0	92H	-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0	0000 0000b	
SFRS ^[4]	SFR page selection	91H	-	-	-	-	-	-	-	SFRPSEL	0000 0000b	
P1	Port 1	90H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	Output latch, 1111 1111b Input XXXX XXXXb ^[3]	
WKCON	Self Wake-up Timer control	8FH	-	-	-	WKTF	WKTR	WKPS[2:0]			0000 0000b	
CKCON	Clock control	8EH	-	PWMCK S	-	T1M	T0M	-	CLOEN	-	0000 0000b	
TH1	Timer 1 high byte	8DH	TH1[7:0]								0000 0000b	
TH0	Timer 0 high byte	8CH	TH0[7:0]								0000 0000b	
TL1	Timer 1 low byte	8BH	TL1[7:0]								0000 0000b	
TL0	Timer 0 low byte	8AH	TL0[7:0]								0000 0000b	
TMOD	Timer 0 and 1 mode	89H	GATE	C/ $\overline{T}$	M1	M0	GATE	C/ $\overline{T}$	M1	M0	0000 0000b	
TCON	Timer 0 and 1 control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000b	
PCON	Power control	87H	SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL	POR, 0001 0000b Others, 000U 0000b	
RWK	Self Wake-up Timer reload byte	86H	RWK[7:0]								0000 0000b	
RCTRIM1	Internal RC trim value low byte	85H	-	-	-	HIRC24	-	-	-	HIRCTRIM[0]	0000 0000b	
RCTRIM0	Internal RC trim value high byte	84H	HIRCTRIM[8:1]								0000 0000b	
DPH	Data pointer high byte	83H	DPTR[15:8]								0000 0000b	
DPL	Data pointer low byte	82H	DPTR[7:0]								0000 0000b	
SP	Stack pointer	81H	SP[7:0]								0000 0111b	
P0	Port 0	80H	(P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	Output latch, 1111 1111b Input.	

符号	定义	地址 (页)	MSB								LSB ^[1]	复位值 ^[2]
											XXXX XXXXb _[3]	

注：

[1] () 项意思是可位寻址SFRs

[2] 复位值符号描述。0: 逻辑 0; 1: 逻辑 1; U: 不变; C: 详见 [5]; X: 详见 [3], [6], 和 [7].

[3] 复位之后所有I/O引脚默认为悬浮输入模式。复位后ICE_DAT和ICE_CLK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。

[4] 这些SFRs有TA写保护。

[5] 在规定复位后，根据CONFIG值，SFRs有些位初始化时设置。

[6] BOF复位值取决于CONFIG2不同的设置和V_{DD} 电压值。

[7] 当欠压检测使能，BOS是只读标志，由V_{DD}值来决定。

表 7.3- 1 SFR 定义及复位值

7.4 所有SFR 介绍

如下重置值意义为：U - 不改变；C - 由CONFIG初始化；X - 基于实际芯片状态。

P0 – 端口

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	0000_0001 b
P3	B0H, 所有页, 可位寻址	0000_0001 b

7	6	5	4	3	2	1	0
Pn.7	Pn.6	Pn.5	Pn.4	Pn.3	Pn.2	Pn.1	Pn.0.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	P0[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]	堆栈指针 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。

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]	数据指针低字节 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。

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]	数据指针高字节 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

RCTRIM0 –内部高速HIRC设定值0

寄存器	SFR地址	复位值
RCTRIM0	84H, 所有页, TA保护	16MHz HIRC 预设值

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

RCTRIM1 –内部高速HIRC设定值1

寄存器	SFR地址	复位值
RCTRIM1	85H, 所有页, TA保护	16MHz HIRC 预设值

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

RWK – 自唤醒定时器重装载数据寄存器

寄存器	SFR地址	复位值
RWK	86H, 所有页	0000 _ 0000b

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

位	名称	描述
7:0	RWK[7:0]	WKT 重载字节 用以存储WKT的8位重载值。注意如果预分频是1/1，RWK限制不能是FFH。

PCON – 电源控制寄存器

寄存器	SFR地址	复位值
PCON	87H, 所有页	POR: 0001_000b, 其它: 000U_0000b

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

位	名称	描述
7	SMOD	串行口 0 波特率加倍使能 当UART0在模式2时或当定时器1溢出被用作UART0模式1或3的波特率源设置此位可加倍串口波特率。见 表 15.1-1 详细描述。
6	SMOD0	串行口 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	低功耗运行模式 0 = 禁止 1 = 使能 注： PD = 1 且 LPR = 1 时，LPR 无效，CPU 将进入掉电模式。
3	GF1	上电复位标志 当上电复位后该位置1，用以标示当前冷复位，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0。
2	GF0	通用标志 1 通用标志可由用户通过软件置位和清零。
1	PD	通用标志 0 通用标志可由用户通过软件置位和清零。

位	名称	描述
0	IDL	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。此时为最小功耗。CPU从掉电模式下唤醒后，该位自动由硬件清零，且在系统唤醒之前程序继续执行中断服务程序（ISR）。从ISR返回后，设备继续执行系统进入掉电模式时所处的指令。 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。

TCON – 定时器 0 和 1 控制位

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
7	TF1	定时器 1 溢出标志 在定时器1溢出时该位置1。当程序响应定时器1中断执行相应的中断服务程序时，该位自动清0。软件也可对其写1或写0
6	TR1	定时器 1 启动控制 0 = 定时器1 中止. 清该位将中止定时器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 中止. 清该位将中止定时器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$\overline{\text{INT1}}$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$\overline{\text{INT1}}$的中断触发类型。 0 = $\overline{\text{INT1}}$ 是低电平触发。 1 = $\overline{\text{INT1}}$ 是下降沿触发。

位	名称	描述
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1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

TMOD – 定时器0 和 1 模式寄存器

寄存器	SFR地址	复位值
TMOD	89H 所有页	0000_0000b

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

位	名称	描述															
7	GATE	定时器 1 门控制 0 = 不管 $\overline{INT1}$ 的逻辑电平, 当TR1为1时, 定时器1时钟运行 1 = 当TR1为1且 $\overline{INT1}$ 为逻辑1时, 定时器1时钟运行															
6	$C/\overline{T}$	定时器 1 计数器/定时器 选择 0 = 定时器1随内部时钟而递增 1 = 定时器1 随外部引脚T1的下降沿递增															
5	M1	定时器 1 模式选择 <table><tr><th>M1</th><th>M0</th><th>定时器 1 模式</th></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: 8位带从TH1自动重载功能的定时器/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>	M1	M0	定时器 1 模式	0	0	模式 0: 13位定时器/计数器	0	1	模式1: 16位定时器/计数器	1	0	模式 2: 8位带从TH1自动重载功能的定时器/计数器	1	1	模式 3: 定时器 1 停止
M1	M0		定时器 1 模式														
0	0		模式 0: 13位定时器/计数器														
0	1		模式1: 16位定时器/计数器														
1	0		模式 2: 8位带从TH1自动重载功能的定时器/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, 当TR0为1时, 定时器0时钟运行 1 = 当TR1为1且 $\overline{INT0}$ 为逻辑1时, 定时器0时钟运行															
2	$C/\overline{T}$	定时器0 计数器/定时器 选择 0 = 定时器0随内部时钟而递增 1 = 定时器1 随外部引脚T1的下降沿递增															

位	名称	描述
1	M1	定时器0 模式选择
0	M0	
		定时器 0 模式
		模式 0: 13位定时器/计数器
		模式 1: 16位定时器/计数器
		模式 2: 8位带从TH0自动重载功能的定时器/计数器
		模式 3: TL0 作为一个 8位 定时器/计数器 , TH0 作为一个 8位 定时器

TL0 – 定时器 0 低字节

寄存器	SFR地址	复位值
TL0	8AH 所有页	0000_0000b

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

位	名称	描述
7:0	TL0[7:0]	定时器0 低字节 寄存器TL0是定时器0的16位数值的低字节

TL1 – 定时器 1 低字节

寄存器	SFR地址	复位值
TL1	8BH, 所有页	0000_0000b

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

位	名称	描述
7:0	TL1[7:0]	定时器1低字节 寄存器TL1是定时器1的16位数值的低字节

TH0 – 定时器0 高字节

寄存器	SFR地址	复位值
TH0	8CH, 所有页	0000_0000b

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

TH0[7:0]
读/写

位	名称	描述
7:0	TH0[7:0]	定时器0 高字节 寄存器TH0是定时器0的16位数值的高字节

TH1 – 定时器1 高字节

寄存器	SFR地址	复位值
TH1	8DH, 所有页	0000_0000b

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

位	名称	描述
7:0	TH1[7:0]	定时器1 高字节 寄存器TH1是定时器1的16位数值的高字节

CKCON – 时钟控制

寄存器	SFR地址	复位值
CKCON	8EH, 所有页	0000_0000b

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

位	名称	描述
6	PWMCKS	PWM 时钟源选择 0 = PWM时钟源为系统时钟FSYS 1 = PWM时钟源为Timer 1的溢出
4	T1M	定时器1时钟模式选择 0 = 定时器 1 的时钟源是系统时钟的12除频, 它保留标准8051的特性 1 = 定时器 1 的时钟源直接是系统时钟
3	T0M	定时器0时钟模式选择 0 = 定时器 0 的时钟源是系统时钟的12除频, 它保留标准8051的特性 1 = 定时器 0 的时钟源直接是系统时钟

位	名称	描述
1	CLOEN	系统时钟输出使能 0 = 系统时钟输出禁止 1 = 系统时钟输出使能，从CLO输出

WKCON – 自唤醒定时器控制器

寄存器	SFR地址	复位值
WKCON	8FH, 所有页	0000_0000b

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

位	名称	描述
4	WKTF	WKT 溢出标志 当WKT溢出，该位置位。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。
3	WKTR	WKT 运行控制 0 = WKT 停止。 1 = WKT 开始运行。 注意重载寄存器RWK仅在WKT停止的时候可以写入（WKTR位为0）。否则结果是不可预知的。
2:0	WKPS[2:0]	WKT 预分频 这些位决定WKT时钟的预分频。 000 = 1/1. 001 = 1/4. 010 = 1/16. 011 = 1/64. 100 = 1/256. 101 = 1/512. 110 = 1/1024. 111 = 1/2048.

SFRS – SFR 页选择

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

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

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

CAPCON0 – 输入捕获控制 0

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

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

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

CAPCON1 – 输入捕获控制器 1

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

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

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

位	名称	描述
1:0	CAP0LS[1:0]	输入捕获通道0条件选择 00 = 下降沿 01 = 上升沿 10 = 双边沿或下降沿 11 = 保留

CAPCON2 – 输入捕获控制 2

寄存器	SFR地址	复位值
CAPCON2	94H, 所有页	0000_0000 b

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

位	名称	描述
6	ENF2	输入捕获通道2噪声滤波使能 0 = 输入捕获通道2噪声滤波关闭 1 = 输入捕获通道2噪声滤波使能
5	ENF1	输入捕获通道1噪声滤波使能 0 = 输入捕获通道1噪声滤波关闭 1 = 输入捕获通道1噪声滤波使能
4	ENF0	输入捕获通道0噪声滤波使能 0 = 输入捕获通道0噪声滤波关闭 1 = 输入捕获通道0噪声滤波使能

CKDIV – 时钟分频器

寄存器	SFR地址	复位值
CKDIV	95H, 所有页	0000_0000 b

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

位	名称	描述
7:0	CKDIV[7:0]	时钟分频器 系统时钟频率 F _{SYS} 根据CKDIV的值遵循以下计算公式。 $F_{SYS} = F_{OSC}, \text{ CKDIV} = 00H, \text{ 以及}$ $F_{SYS} = \frac{F_{OSC}}{2 \times \text{CKDIV}}, \text{ CKDIV} = 01H \text{ to } FFH.$

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
-	-	HIRCST	LIRCST	ECLKST	OSC[1:0]		-
-	-	只读	只读	只读	W		-

位	名称	描述
7:6	-	保留位
5	HIRCST	高速内部振荡器 24 MHz 状态 0 = 高速内部振荡器不稳定或禁用。 1 = 高速内部振荡器使能且稳定
4	-	保留位
3	ECLKST	外部时钟输入状态 0 = 外部时钟输入不稳定或禁用。 1 = 外部时钟输入使能且稳定
2:1	OSC[1:0]	振荡器选择位 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 = 保留位 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 (TA 保护)

寄存器	SFR地址	复位值
CKEN	97H, 所有页, TA保护	0011_0000 b

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

位	名称	描述
7:6	EXTEN[1:0]	外部时钟源使能 11 = 使能外部时钟，通过XIN输入外部时钟信号 其它 = 禁止外部时钟。P30作为普通I/O使用
5	HIRCEN	HIRC16MHz 使能位 0 = 关闭 HIRC 1 = 使能 HIRC 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
4:1	-	保留位
0	CKSWTF	时钟切换错误标志位 0 = 先前的系统时钟源切换成功 1 = 用户试图将先前的系统时钟源切换到没有开启或是不稳定的时钟源。如果待切换的时钟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。

SCON – 串行端口控制

寄存器	SFR地址	复位值
SCON	98H, 所有页, 可位寻址	0000_0000 b

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

位	名称	描述
7	SM0/FE	串行端口选择 SMOD0 (PCON.6) = 0: 见 表 15.1- 1 详细描述. SMOD0 (PCON.6) = 1: SM0/FE 位被用来做帧错误 (FE) 状态标志. 可被软件清除. 0 = 帧错误 (FE) 未发送错误. 1 = 检测到帧错误 (FE) .
6	SM1	
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“给定”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“给定”或“广播”地址时, 接收才有效。
4	REN	接收使能 0 = 串口0接收禁止。 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
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, 所有页	0000_0000 b
SBUF1	9AH, 所有页	0000_0000 b

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

位	名称	描述
7:0	SBUF[7:0]	串行数据缓存 这个字节实际上由两个单独的寄存器组成。一个是接收寄存器，另一个是发送缓冲器。当数据写到SBUF时，它转到发送缓冲器并被移位以进行串行传输。当数据从SBUF读时，它来自接收寄存器。通过向SBUF写数据来启动传输。

EIE – 扩展中断使能寄存器

寄存器	SFR地址	复位值
EIE0	9BH, 所有页	0000_0000 b

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

位	名称	描述
7	ET2	定时器2中断使能 0 = 定时器2中断禁止. 1 = 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 (SPInSR.7), SPIOVF (SPInSR.5), 或MODF (SPInSR.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	EPWM	PWM0中断使能 0 = PWM 中断禁止. 1 = PWM 中断使能. 当中断产生时PWMF (PWMnCON0.5) 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	EI2C	I²C0 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 (I2C0CON.3) 或 I2TOF (I2C0TOC.0) 置 1.

EIE1 – 拓展中断使能寄存器 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
-	-	-	-	-	EWKT	ET3	ES_1
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	EWKT	WKT中断使能 0 = WKT 中断禁止. 1 = 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.

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
W	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(ISPTRG.0)置位后置位该位: (1) 访问地址超过其大小的区域 (2) IAPCN 命令无效 (3) IAP擦除或编程更新没有使能的区域 (4) 当BOIAP(CONFIG2.5)为1、 BODEN (BODCON0.7)为1以及BORST(BODCON0.2) 为0时, 擦除或编程工作在V _{BOD} 下。 该位应该由软件清零
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

位	名称	描述
0	IAPEN	IAP 使能 0 = 禁用IAP功能 1 = 使能IAP功能 一旦使能IAP功能，HIRC将会被打开作为时序控制。清IAPEN应该在IAP操作最后一条指令，这样可以停止内部振荡器以减少功耗

AUXR1 – 辅助寄存器 1

寄存器	SFR地址	复位值
AUXR1	A2H, 所有页	POR 0000 0000b 软件复位 1U00 0000b, nRESET 复位脚 U100 0000b, 其它 UUUU 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	UART1PX	DPS
读/写	读/写	读/写	读/写	读/写	读/写	读/写	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。建议软件清0。
6	RSTPINF	外部复位标志位 当MCU发生外部复位脚复位，该位将被硬件置1，建议复位发生后软件清0。
5	HardF	硬件错误复位标志 一旦程序计数器（PC）的值超过flash的大小，MCU将会复位并且该位通过硬件置位。建议通过软件清除该位。 注意：如果MCU运行在OCD调试模式并且OCDEN = 0，硬件错误复位将会被禁止。
4	SLOW	ADC 慢速选择 此位用来选择ADC慢速 0 = 速度大于 500 ksps 1 = 速度慢于 200 ksps
3	GF2	通用标志位 2 可通过软件置1或清除
1	UART0PX	串口0管脚位置配置 0 = UART0_RXD 为 P0.7，UART0_TXD 为 P0.6（默认值）。 1 = 交换UART0_RXD 为 P0.6，UART0_TXD 为 P0.7。 注：更改此位后TXD及RXD的配置更改立即生效。软件需确保切换过程中没有进行串口传输，否则可能引发无法预测的结果。
1	UART1PX	串口0管脚位置配置 0 = UART1_RXD为 P0.2，UART1_TXD 为 P1.6（默认值）。 1 = 交换UART1_RXD为 P1.6，UART1_TXD 为 P0.2 注：更改此位后TXD及RXD的配置更改立即生效。软件需确保切换过程中没有进行串口传输，否则可能引发无法预测的结果。
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，之前有效DPTR寄存器内的值保持不变。

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_XC0Xb BOD: UUUU_XU1Xb 其它: UUUU_XUUXb
---------	---------------	--

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

位	名称	描述
7	BODEN	欠压检测使能 0 = 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]	欠压检测电压选择位 11 = VBOD 生效电压 2.2V. 10 = VBOD 生效电压 2.7V. 01 = VBOD 生效电压 3.7V. 00 = VBOD 生效电压 4.4V.
3	BOF	欠压中断标志 当VDD下降到VBOD以下或VDD上升到VBOD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落到以VBOD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VDD与VBOD比较情况。BOD电路关闭时保持为0。 0 = VDD电压大于VBOD或是BOD电路关闭 1 = VDD电压小于VBOD 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V _{DD} 的电平。请参照表 26.2-1		

IAPTRG – IAP 触发

寄存器	SFR地址	复位值
IAPTRG	A4H, 所有页, TA保护	0000_0000 b

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

位	名称	描述
---	----	----

位	名称	描述
0	IAPGO	IAP 执行 设置此位为1时开始执行IAP。在此指令之后，CPU保持程序计数器（PC），IAP硬件自动管理控制该进程。IAP操作完成后，程序计数器继续执行下一个指令。IAPGO位将被自动清除并保持为0。在触发IAP操作之前，由于硬件限制中断（如果启用）应该临时禁用。程序流程如下。 <pre> CLR EA MOV TA,#0AAH MOV TA,#55H ORL IAPTRG,#01H (SETB EA) </pre>

IAPUEN – IAP 更新使能

寄存器	SFR地址	复位值
IAPUEN	A5H, 所有页, 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
2	CFUEN	CONFIG 位升级使能 0 = 禁止通过IAP擦除或编程CONFIG字节 1 = 使能通过IAP擦除或编程CONFIG字节
1	LDUEN	LDROM 升级使能 0 = 禁止通过IAP擦除或编程LDROM 1 = 使能通过IAP擦除或编程LDROM
0	APUEN	APROM 升级使能 0 = 禁止通过IAP擦除或编程APROM 1 = 使能IAP擦除或编程APROM

IAPAL – IAP 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 地址地位 IAPAL 包含地址 IAPA[7:0] 为 IAP 操作。

IAPAH – IAP 地址低字节

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

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

位	名称	描述
7:0	IAPA[15:8]	IAP 地址高位 IAPAH 包含地址 IAPA[15:8] 为 IAP 操作.

IE – 中断使能

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/禁止所有中断, 单独使能. 0 = 禁止所有中断源 1 = 每个中断使能依靠使能单个中断, 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中断使能. 当中断产生时, IE0(TCON.1)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中断使能. 当中断产生时, IE1(TCON.3)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 该字节定义微控制器自身的从机地址用于串口0多机通信。

WDCON 看门狗定时器控制

寄存器	SFR地址	复位值
WDCON	AAH, 所有页, TA保护	POR: 0000 0111b WDT: 0000 1UUUb 其他: 0000 UUUUb

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

位	名称	描述
7	WDTR	WDT 运行 该位仅当控制位WDTEN[3:0] (CONFIG4[7:4])全为1时有效。这时WDT工作在通用定时器模式 0 = WDT 禁止. 1 = WDT 使能. WDT 计数器开始运行.
6	WDCLR	WDT 清除 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 = 无影响. 1 = 清除WDT计数器. 读: 0 = WDT 计数器完整清零. 1 = WDT 计数器还没有清零.
5	WDTF	WDT 超时标志 该位表示WDT计数器的溢出。该标志应该通过软件清零
4	WIDPD	WDT 工作在空闲或掉电模式 该位仅当控制位WDTEN[3:0] (CONFIG4[7:4])全为1时有效。它决定WDT作为通用定时器在空闲或掉电模式下是否保持工作。 0 = WDT 在空闲或掉电模式下停止工作 1 =在空闲或掉电模式下保持工作
3	WDTRF	WDT 复位标志 当WDT超时事件发生, MCU会复位, 该位由硬件置位。建议该位由软件清零
2:0	WDPS[2:0]	WDT 时钟预分频选择 这些位决定了WDT时钟的预分频, 从1/1到1/256.默认的是最大分频值。

位	名称	描述
注: 1. WDTRF上电复位后会被清0, 当发生WDT复位后置1, 其余复位保持原有值不变. 2. WDPS[2:0] 上电复位后全部为1, WDT复位发生不会改变该SFR值		

BODCON1 – 欠压检测控制 1

寄存器	SFR地址	复位值
BODCON1	ABH, 所有页, TA保护	POR: 0000_0001b Others:0000_0UUUb

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

位	名称	描述
7:3	-	保留
2:1	LPBOD[1:0]	低功耗BOD使能 00 = BOD 正常模式. BOD 电路总是开启. 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已HIRC或ECLK作为系统时钟且BOD没有在低功耗模式下(LPBOD[1:0] = [0, 0])工作时, 该滤波器采样32个FSYS 时钟以过滤电源噪声。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.)

IAPFD – IAP 内存数据

寄存器	SFR地址	复位值
IAPFD	AEH, 所有页	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 flash数据 该字节包含将要读或写进flash空间的数据。编程模式下, 用户需要在触发IAP之前写数据到IAPFD里, 读/校验模式下, 在IAP完成后从IAPFD读出数据。

IAPCN – IAP 控制

寄存器	SFR地址	复位值
-----	-------	-----

IAPCN	AFH, 页0	0011_0000 b
-------	---------	-------------

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

位	名称	描述
7:6	IAPB[1:0]	IAP 控制 此字节是IAP控制命令。详细描述见 表 23.1- 1
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

PnM1 – 端口模式选择寄存器 1

寄存器	SFR地址	复位值
P0M1	B1H, 页0	1111_1111 b
P1M1	B3H, 页0	1111_1111 b
P3M1	ACH, 页0	1111_1111 b

注: PnM1 和 PnM2 [n:0~7] 组合使用来确定P0引脚的I/O模式. 见表 9.7- 1

7	6	5	4	3	2	1	0
PnM1.7	PnM1.6	PnM1.5	PnM1.4	PnM1.3	P0M1.2	P0M1.1	P0M1.0
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 1NOTE: 详见表 9.7- 1

PnS – 端口 n 施密特触发输入

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

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

位	名称	描述
---	----	----

位	名称	描述
n	P0S.n	P0 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。

PnM2 – 端口模式选择2

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

7	6	5	4	3	2	1	0
P0M2.7	P0M2.6	P0M2.5	P0M2.4	P0M2.3	P0M2.2	P0M2.1	P0M2.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	P0M2[7:0]	端口 1 模式选择2

P2S – P20 设置及定时器0/1输出使能

寄存器	SFR地址	复位值
P2S	B5H, 所有页	0000_0000 b

7	6	5	4	3	2	1	0
P20UP	-	-	-	T1OE	T0OE	-	P2S.0
R/W	-	-	-	R/W	R/W	-	R/W

位	名称	描述
7	P20UP	P2.0 上拉使能位 0 = P2.0 禁止上拉 1 = P2.0 使能上拉 仅当RPD (CONFIG0.2)为0时该位有效。如果作为nRESET引脚，上拉总是被使能
3	T1OE	定时器 1 输出使能位 0 = 禁止定时器 1 输出 1 = 使能定时器 1 输出从T1引脚 注意：仅当运行在定时器模式时，定时器1输出才会被使能
2	T0OE	定时器 0 输出使能位 0 = 禁止定时器 0 输出 1 = 使能定时器 0 输出从T0引脚 注意：仅当运行在定时器模式时，定时器0输出才会被使能
0	P2S.0	P2.0 施密特触发输入选择 0 = P2.0 TTL电平输入 1 = P2.0 施密特触发输入

IPH –中断优先级高位

寄存器	SFR地址	复位值
IPH	B7H, 页0	0000_0000 b

注：IP与IPH结合使用，以确定每个中断源的优先级。见表 26-2. 中断优先级设定以正确配置中断优先级。

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	Timer 0 i中断优先高位
0	PX0H	外部中断0中断优先级高位

PWMINTC –PWM 中断控制寄存器

寄存器	SFR地址	复位值
PWMINTC	B7H, 页1	0000_0000 b

7	6	5	4	3	2	1	0
-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSELO
-	-	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 通过该位选择PWM的中断类型 00 = PWM0/1/2/3/4/5 脚下降沿. 01 = PWM0/1/2/3/4/5 脚上升沿. 10 = 每个PWM周期的中心店. 11 = 每个PWM周期的终点. 注：中心点中断方式或终点中断方式仅适用于PWM中心对齐模式
2:0	INTSEL[2:0]	PWM 中断对选择 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 = PWM0. 001 = PWM1. 010 = PWM2. 011 = PWM3. 100 = PWM4. 101 = PWM5. 其它 = PWM0.

IP – 中断优先级寄存器

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

SADEN – 从机 0 地址掩码

寄存器	SFR地址	复位值
PWMINTC	B9H, 所有页	0000_0000 b

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

位	名称	描述
7:0	SADEN[7:0]	从机0地址掩码 该字节为UART0的掩码, 其中无关位用0表示, 这样可以形成一个或多个更灵活的自身从机地址

SADEN_1 – 从机 1 地址掩码

寄存器	SFR地址	复位值
SADEN_1	BAH, 所有页	0000_0000 b

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

位	名称	描述
7:0	SADEN_1[7:0]	从机1 地址掩码。 该字节为UART1的掩码, 其中无关位用0表示, 这样可以形成一个或多个更灵活的自身从机地址

SADDR_1 –从机 1 地址

寄存器	SFR地址	复位值
SADDR_1	BBH, 所有页	0000_0000 b

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

位	名称	描述
7:0	SADDR_1[7:0]	从机1地址 该字节定义微控制器自身的从机地址用于串口1多机通信。

I2DAT – I²C 数据寄存器

寄存器	地址, 页, 特别说明	复位值
I2DAT	BCH, 所有页	0000_0000 b

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

位	名称	描述
7:0	I2DAT[7:0]	I²C 数据寄存器. 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传输过程中去读或写I2DAT, 其结果不可预知。 数据发送时, 数据移位到总线上, 同时总线上的数据会接收回来, 所以总线仲裁失败时, 再读这个数据, 可能与之前写入的值不一样。

I2STAT – I²C 状态寄存器

寄存器	SFR地址	复位值
I2STAT	BDH, 所有页	1111_1000 b

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

位	名称	描述
7:3	I2STAT[7:3]	I²C 状态字 I2STAT的高5位为状态码, 共有27种状态值。I2STAT = F8H 时, 表示空闲, SI 将保持为0。其它26种状态值与I ² C的状态一样, 进入每一种状态都会让SI置1, 且产生中断请求。

位	名称	描述
2:0	0	保留位 I2STAT的低3位最低有效位读取始终为0

I2CLK – I²C 时钟寄存器

寄存器	SFR地址	复位值
I2CLK	BEH, 所有页	0000 _1001 b

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

位	名称	描述
7:0	I2CLK[7:0]	I²C 时钟设定 <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: $\frac{F_{SYS}}{4 \times (I2CLK + 1)}$ 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。注意: I2CLK 值写入 00H 及 01H 无效。 <u>从机模式:</u> 该字节无效, 从机自动跟随主机时钟, 最高 400k bps。

I2TOC – I²C 超时计数器

寄存器	SFR地址	复位值
I2CLK	BFH, 所有页	0000 _0000 b

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

位	名称	描述
2	I2TOCEN	I²C 超时计数器使能位 0 = I²C 超时计数器关闭 1 = I²C 超时计数器使能
1	DIV	I²C 超时计数器计时除频 0 = I²C 超时计数器除频 F_{SYS}/1 1 = I²C 超时计数器除频 F_{SYS}/4
0	I2TOF	I²C 超时标志 如果14位I²C超时计数器溢出, 该位会由硬件置位。该位需要由软件清0

I2CON – I²C 控制寄存器

寄存器	SFR地址	复位值
I2CON	C0H, 所有页, 可位寻址	0000_0000 b

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

位	名称	描述
6	I2CEN	I²C 总线使能 0 = I ² C 禁止. 1 = I ² C 使能. 使能 I ² C 之前, SCL 和 SDA 必须配置为输出1
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 如果总线已经在总线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。
4	STO	停止标志STOP I ² C 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(I2STAT 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, I ² C 总线在发送 START 后马上发送 STOP。如果在从机模式下, 应避免 STA 及 STO 同时置1, 以避免引发帧错误。
3	SI	I²C 中断标志 I ² C 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 内值, 决定下一步骤。 SI 由软件清0。在 SI 被清0之前, 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, 可以确保准确处理再接收下一笔数据。 SI 位被软件清0后, I ² C 外设才会继续下一步: 根据软件控制位设定, 继续发数据, 或起始位, 或停止位等, 因此在清除 SI 位之前, 软件需确认适当的状态。
2	AA	应答应标志 若 AA = 1, 接收数据时, 会在第9位发出应答 ACK——拉低数据线。 若 AA = 0, 接收数据时, 将向总线发送不应答 (NACK), 第9位时不拉低数据线, 若器件自我清除 AA 标志位, 则会清除其从机地址或广播呼叫, SI 会被清0, 中断不予产生。所以 MCU 将不响应任何数据, 包括从机地址。 从机接收时若 AA=0 不返回应答, 或从机发送时没收到应答, 从机传输就结束了。 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, 且重新被主机寻址。 特殊情况: 注意: 从机发送时, 状态码若为 C8H, 从机发送最后一个字节之前, 让 AA= 0, 发送完最后一个字节, 不再回应答, 传输结束。主机若再从总线上读数据, 将得到 FFH。
0	I2CPX	I²C 引脚选择 0 = 分配 SCL 是 P1.3 和 SDA 是 P1.4. 1 = 分配 SCL 是 P0.2 和 SDA 是 P1.6. 注意: 一旦该位置位或清0, I2C 引脚会立刻更换。

I2ADDR – I²C 从机地址

寄存器	SFR地址	复位值
-----	-------	-----

I2ADDR	C1H, 所有页	0000 _0000 b
--------	----------	--------------

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

位	名称	描述
7:1	I2ADDR[7:1]	I²C 从机地址 主机模式: 该位段无效 从机模式: 存放7位从机地址。主机需在START或re-START之后写入该值, 发送从机地址。如果AA为1, I ² C设备接收到从地址后, 并响应主机。否则主机呼叫地址会被忽略。 注: I2ADDR[7:1] 不能写为全0, 因为0x00为广播呼叫方式寻址专用。
6	GC	广播呼叫位 主机模式: 无效 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。

ADCRL –转换结果低位寄存器

寄存器	SFR地址	复位值
ADCRL	C2H, 所有页	0000 _0000 b

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

位	名称	描述
3:0	ADCR[3:0]	ADC转换结果低位 ADC转换结果低4位

ADCRH –ADC转换结果高位寄存器

寄存器	SFR地址	复位值
ADCRH	C3H, 所有页	0000 _0000 b

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

位	名称	描述
---	----	----

位	名称	描述
7:0	ADCR[11:4]	ADC转换结果高位 ADC转换结果高8位

T3CON – 定时器3控制寄存器

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，时钟溢出波特率加倍使能。详见表 15.1-1 表 15.1-1 串口 Port 0 模式 / 波特率描述
6	SMOD0_1	串口1帧错误使能位 0 = 关闭帧错误检测功能，作为SM0_1用 1 = 使能帧错误检测，用作帧错误 (FE) 状态标志FE_1
5	BRCK	串口0波特率时钟源选择 该位用于设置串口0在模式1或模式3下，所使用的定时器 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，该位置位。当程序执行定时器3的中断服务程序，该位由硬件自动清零。该位可以通过软件置位或清零。
3	TR3	定时器3运行控制 0 = 定时器3停止。 1 = 定时器3开始计数 注意重载寄存器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位为1写入RH3或RL3，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自动重装载寄存器低字节

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 重载低字节 它保持着定时器3重载值的低字节

RH3 – Timer 3 Reload High Byte

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 重载高字节 它保持着定时器3重载值的高字节

PIOCON1 – PWM or I/O Select

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

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

位	名称	描述
5	PIO15	P1.5/PWM5 管脚功能选择位 0 = P1.5/PWM5 管脚用作 P1.5. 1 = P1.5/PWM5 管脚用作PWM5 输出.
3	PIO13	P0.4/PWM3 管脚功能选择位 0 = P0.4/PWM3 管脚用作 P0.4. 1 = P0.4/PWM3 管脚用作 PWM3 输出.
2	PIO12	P0.5/PWM2 管脚功能选择位 0 = P0.5/PWM2 管脚用作 P0.5. 1 = P0.5/PWM2 管脚用作 PWM2 输出.
1	PIO11	P1.4/PWM1 管脚功能选择位 0 = P1.4/PWM1 管脚用作 P1.4. 1 = P1.4/PWM1 管脚用作 PWM1 输出.

TA –时控保护寄存器

寄存器	SFR地址	复位值
TA	C7H, 页1	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]	时控保护 TA寄存器控制着对被保护的SFRs的访问权限。当需要写特殊规定的寄存器时，必须先对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寄存器写入数据。

T2CON –定时器 2 控制寄存器

寄存器	SFR地址	复位值
T2CON	C8H, 所有页	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溢出或者比较数据符合，该位置1。如果已使能定时器2中断，该位置1会让芯片进入定时器2中断运行。该位硬件置1，但需要软件清0。
2	TR2	定时器2 运行控制 0 = 定时器计数关闭。清除该位关闭定时器计数，目前已计数值会保留在TH2及TL2中 1 = 定时器2计数使能。
0	CM/RL2	定时器 2 比较及自动重装载功能选择 0 = 自动重装载模式 1 = 比较器模式。

T2MOD –定时器2模式选择

寄存器	SFR地址	复位值
T2MOD	C9H, 所有页	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
---	----	----

位	名称	描述
7	LDEN	自动重载使能位 0 = 自动重载功能关闭 1 = 将RCMP2H及RCMP2L自动重载至TH2及TL2功能使能
6:4	T2DIV[2:0]	定时器 2时钟除频 000 = 定时器2时钟分频为 1/1. 001 = 定时器2时钟分频为 1/4. 010 = 定时器2时钟分频为 1/16. 011 = 定时器2时钟分频为 1/32. 100 = 定时器2时钟分频为 1/64. 101 = 定时器2时钟分频为 1/128. 110 = 定时器2时钟分频为 1/256. 111 = 定时器2时钟分频为 1/512.
3	CAPCR	捕获模式自动清除 该位仅当定时器2设定为自动重载模式下有效。该位使能，当捕获完成，TH2及TL2内数据移入RCMP2H 及 RCMP2L后，硬件自动清除TH2及TL2计数寄存器功能。 0 = 捕获完成后定时器2计数按之前计数值继续累加。 1 = 捕获完成后定时器2数据自动清0
2	CMPCR	比较完成自动清除 该位仅当定时器2比较功能模式下有效。当比较符合后，自动清除TH2及TL2计数器值。 0 = 比较符合之后，定时器2内数据按之前继续计数。 1 = 比较符合之后，定时器2内数据自动清0。
1:0	LDTS[1:0]	自动重载触发选择 00 = 当定时器2溢出自动重载 01 = 当捕获0通道事件完成，自动重载 10 = 当捕获1通道事件完成，自动重载 11 = 当捕获2通道事件完成，自动重载

RCMP2L –定时器 2 重载 / 比较数据低字节

寄存器	SFR地址	复位值
RCMP2L	CAH, 所有页	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/比较器低字节 当定时器2设定为比较模式，预存放低字节待比较数据。 当设定为自动重载，预存放低字节数据。

RCMP2H –定时器 2 重载 / 比较数据高字节

寄存器	SFR地址	复位值
RCMP2H	CBH, 所有页	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 重载/比较器高字节 当定时器2设定为比较模式，预存放高字节待比较数据。 当设定为自动重载，预存放高字节数据。

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 低字节数据 该寄存器存放16位定时器2实际计数的低8位字节数据。

TH2 – Timer 2 High Byte

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 高字节数据 该寄存器存放16位定时器2实际计数的高8位字节数据。

ADCMPL –ADC比较值低位寄存器

寄存器	SFR地址	复位值
ADCMPL	CEH,所有页	0000_0000 b

7	6	5	4	3	2	1	0
-	-	-	-	ADCMP[3:0]			
-	-	-	-	W/只读			

位	名称	描述
3:0	ADCMP[3:0]	ADC 比较值低位 ADC比较值低4位

ADCMPH –ADC比较值高位寄存器

寄存器	SFR地址	复位值
ADCMPH	CFH,所有页	0000 _0000 b

7	6	5	4	3	2	1	0
ADCMH[11:4]							
W/只读							

位	名称	描述
7:0	ADCMH[11:4]	ADC比较值高字节 ADC比较值高字节8位内容

PSW –编程状态字

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	只读

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序第4位进位或借位，该位置位，否则清零。																				
5	0	用户标志0. 可由用户置位或清零的通用标志。																				
4	RS1	寄存器页选择 这两位用来选择R0到R7位于四页中的哪一页：. <table><tr><th><u>RS1</u></th><th><u>RS0</u></th><th><u>寄存器页</u></th><th><u>RAM 地址</u></th></tr><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></table>	<u>RS1</u>	<u>RS0</u>	<u>寄存器页</u>	<u>RAM 地址</u>	0	0	0	00H 到 07H	0	1	1	08H 到 0FH	1	0	2	10H 到 17H	1	1	3	18H 到 1FH
<u>RS1</u>	<u>RS0</u>		<u>寄存器页</u>	<u>RAM 地址</u>																		
0	0	0	00H 到 07H																			
0	1	1	08H 到 0FH																			
1	0	2	10H 到 17H																			
1	1	3	18H 到 1FH																			
3	RS0																					
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。																				

位	名称	描述
1	1	用户标志1 可由用户置位或清零的通用标志。
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。

PWMPH – PWM Period High Byte

寄存器	SFR地址	复位值
PWMPH	D1H,所有页	0000_0000 b

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

位	名称	描述
7:0	PWMP[15:8]	PWM周期寄存器高字节 与PWMPL搭配组成PWM周期信号

PWMnH – PWM 通道 0~5 占空比寄存器高字节 n =0,1,2,3,4,5

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

7	6	5	4	3	2	1	0
PWM0[15:8]							
读/写							

位	名称	描述
7:0	PWM0[15:8]	PWM占空比高字节 这字节与PWMnL搭配确定PWM输出的占空比

PNP –PWM 负极性寄存器

寄存器	SFR地址	复位值
PNP	D6H,所有页	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
n	PNPn	PWMn 负极性输出使能 0 = PWMn 输出按照设定直接输出到PWMn管脚 1 = PWMn 输出按照设定取反输出到PWMn管脚

FBD – PWM F 故障刹车数据

寄存器	SFR地址	复位值
FBD	D7H,所有页	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	故障刹车标志位 当FBINEN设置为1, FB管脚上检测到符合FBINLS (FBD.6)设定的边沿信号后, 该位置1。该位需要通过软件清0。当FBF清0后, 故障刹车仍然不会释放PWM输出, 需要重新输出, 除非设置PWMRUN (PWMCON0.7)为1, 重新启动PWM输出。
6	FBINLS	FB 管脚输入选择 0 = 下降沿 1 = 上升沿
N	FBDn	PWMn 故障刹车数据 0 = 当故障发生时PWMn 信号输出为0 1 = 当故障发生时PWMn 信号输出为1

PWMCON0 – PWM PWM 控制寄存器 0

寄存器	SFR地址	复位值
PWMCON0	D8H,所有页, 可位寻址	0000 _0000 b

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

位	名称	描述
7	PWMRUN	PWM 运行使能位 0 = PWM 模块空闲 1 = PWM 开始运行

位	名称	描述
6	LOAD	PWM 载入新周期及占空比 该位用于载入周期及占空比所新设定的值。当前一个PWM周期输出结束，载入动作才会开始。更改的周期及占空比值将在下一个周期展现。当载入动作结束，硬件自动将LOAD位清0。这一特性会使得LOAD位写入及读出值可能不一致，意义也不相同。 <u>写:</u> 0 = 不动作。 1 = 载入之前在缓存内存入的周期及占空比值。 <u>读:</u> 0 = 载入动作已完成 1 = 正在执行载入动作
5	PWMF	PWM 标志位 该位根据PWMINTC的INTSEL[2:0]及 INTTYP[1:0] 位来设定。由软件清0。
4	CLRPWM	清除 PWM计数器 该位置1，会直接清零PWM16位计数器至0000H。当清除计数器动作完成，硬件会自动将该位清0。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 = 无动作。 1 = 清除16位PWM计数器。 <u>读:</u> 0 = PWM16位计数器已清除。 1 = 16位计数器内还有数值，未清除。

PWMPL – PWM 周期寄存器低字节

寄存器	SFR地址	复位值
PWMPL	D9H, 所有页	0000_0000 b

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

位	名称	描述
7:0	PWMP[7:0]	PWM 周期值低字节 该寄存器存储与PWMPH搭配存储PWM的周期值，该位段为低字节

PWMnL –PWM 通道 0~5占空比寄存器低字节 n=0,1; x=0,1,2,3,4,5

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

7	6	5	4	3	2	1	0
PWM0[7:0]							
读/写							

位	名称	描述
7:0	PWMn[7:0]	PWM 占空比低字节 该位与PWMnH搭配，配置由PWM输出的占空比数据

PIOCON0 –PWM 或 I/O选择寄存器

寄存器	SFR地址	复位值
PIOCON0	DEH, 所有页	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/PWM5管脚功能选择位 0 = P0.3/PWM5 管脚用作P0.3. 1 = P0.3/PWM5 管脚用作PWM5输出.
4	PIO04	P0.1/PWM4管脚功能选择位 0 = P0.1/PWM4 管脚用作P0.1. 1 = P0.1/PWM4 管脚用作PWM4输出.
3	PIO03	P0.0/PWM3管脚功能选择位 0 = P0.0/PWM3 管脚用作P0.0. 1 = P0.0/PWM3 管脚用作PWM3输出.
2	PIO02	P1.0/PWM2管脚功能选择位 0 = P1.0/PWM2 管脚用作P1.0. 1 = P1.0/PWM2 管脚用作PWM2输出.
1	PIO01	P1.1/PWM1管脚功能选择位 0 = P1.1/PWM1 管脚用作P1.1. 1 = P1.1/PWM1 管脚用作PWM1输出.
0	PIO00	P1.2/PWM0管脚功能选择位 0 = P1.2/PWM0 管脚用作P1.2. 1 = P1.2/PWM0 管脚用作PWM0输出.

PWMCON1 – PWM 控制寄存器 1

寄存器	SFR地址	复位值
PWMCON1	DFH, 所有页	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 = 独立输出模式 01 = 互补模式 10 = 同步模式 11 = 保留位
5	GP	群组模式使能位 该位使能PWM群组模式。一旦使能，三对PWM的占空比由PWM01H 和 PWM01L决定，原本配置的数据失效。 0 = 未组成群组模式。 1 = 群组模式使能
4	PWMTYP	PWM 类型选择 0 = 边沿对齐型 PWM 1 = 中心对齐型 PWM
3	FBINEN	FB管脚输入使能位 0 = 禁止 1 = 通过FB管脚输入使能PWM故障刹车功能。一旦在FB管脚输入的边沿信号与FBINLS (FBD.6)选择位相同。PWM0~5输出FBD寄存器所设置的信号电平， PWMRUN (PWMCN0.7) 位硬件清除。当PWMRUN信号重置1， PWM信号重新输出。
2:0	PWMDIV[2:0]	PWM 时钟除频 该寄存器段用于配置PWM时钟频率预分频。 000 = 1/1. 001 = 1/2 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

A or ACC –累加器

寄存器	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 标准80C51 累加器

ADCCON1 –ADC控制寄存器1

寄存器	SFR地址	复位值
ADCCON1	E1H, 所有页	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
6	STADCPX	外部启动ADC触发脚选择 0 = 分配 STADC 到 P0.4. 1 = 分配 STADC 到 P1.3. 注意：一旦该位置位或清0，STADC脚会立即转换。
5:4	ADCDIV[1:0]	外部触发信号类型选择 当 ADCEX (ADCCON1.1) 置1，该位决定响应外部触发的类型。 00 = PWM0/2/4 或 STADC 脚的下降沿. 01 = PWM0/2/4 或 STADC 脚的上升沿. 10 = 一个PWM周期的中点. 11 = 一个PWM周期的终点. 注PWM周期中点或终点触发仅适用于中心对齐模式的PWM输出。
3:2	ETGTYP[1:0]	ADC 触发启动信号选择位 该位决定启动ADC的触发条件 0 = 当软件设定ADCS位，启动AD转换 1 = 当软件设定ADCS位或外部触发信号启动AD转换。外部触发信号条件由寄存器 ETGSEL[1:0] 及 ETGTYP[1:0]决定。注，当ADCS为1时（正在转换），外部触发信号不会影响ADC直到ADC转换结束，ADCS被硬件清0。
1	ADCEX	ADC 使能位 0 = ADC 转换电路关闭 1 = ADC 转换电路打开
0	ADCEN	外部启动ADC触发脚选择 0 = 分配 STADC 到 P0.4. 1 = 分配 STADC 到 P1.3. 注意：一旦该位置位或清0，STADC脚会立即转换。

ADCCON2 –ADC控制寄存器 2

寄存器	SFR地址	复位值
ADCCON2	E2H, 所有页	0000 _0000 b

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

位	名称	描述
7	ADFBEN	ADC比较结果响应故障刹车使能寄存器 0 = 关闭 1 = ADC 触发故障刹车功能打开 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比较器输出极性选择位 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 结果比较使能位 0 = ADC 结果比较功能关闭。 1 = ADC 结果比较功能打开。

位	名称	描述
4	ADCMPO	ADC比较结果输出位 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	ADCAQT	ADC 采集时间 这个3位字段决定ADC采样的捕获时间，遵循下面的等式： $\text{ADC 采集时间} = \frac{4 * \text{ADCAQT} + 6}{F_{\text{ADC}}}$ 默认及最小捕获时间是10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。

ADCDLY –ADC外部触发延迟计数器

寄存器	SFR地址	复位值
ADCDLY	E3H, 所有页	0000_0000 b

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

位	名称	描述
7:0	ADCDLY[7:0]	ADC 外部触发启动延迟计数器低位 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 $\text{外部延迟时间} = \frac{\text{ADCDLY}}{F_{\text{ADC}}}$ 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。

CnL –捕获通道n低字节 n= 0,1,2

寄存器			SFR地址				复位值	
C0L			E4H, 所有页				0000_0000 b	
C1L			E6H, 所有页				0000_0000 b	
C2L			EDH, 所有页				0000_0000 b	
7	6	5	4	3	2	1	0	
C0L[7:0]								
读/写								

位	名称	描述
7:0	C0L[7:0]	捕获通道n 输入结果低字节 寄存器CnL是16位捕获通道输入结果的低字节值

CnH –捕获通道n高字节 n= 0,1,2

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

7	6	5	4	3	2	1	0
C0H[7:0]							
读/写							

位	名称	描述
7:0	C0H[7:0]	捕获通道n 输入结果高字节 寄存器CnL是16位捕获通道输入结果的高字节值

ADCCON0 –ADC控制寄存器0

寄存器	SFR地址	复位值
ADCCON0	E8H, 所有页, 可位寻址	0000 _0000 b

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]	A/D转换通道选择 该位用于选择ADC转换通道。当ADCEN 为 0所有输入无效。 0000 = AIN0. 0001 = AIN1. 0010 = AIN2. 0011 = AIN3. 0100 = AIN4. 0101 = AIN5. 0110 = AIN6. 0111 = AIN7. 1000 =内部带隙电压(band-gap)1.22V. Others = 保留.

PICON –管脚中断控制寄存器

寄存器	地址, 页, 特别说明	复位值
PICON	E9H, 所有页	0000_0000 b

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

位	名称	描述
7	PIT67	管脚中断通道6 及 7 类型选择 该位用以配置管脚中断6和7产生中断的信号类型 0 = 电平触发 1 = 边沿触发
6	PIT45	管脚中断通道4 及 5 类型选择 该位用以配置管脚中断4和5产生中断的信号类型 0 = 电平触发 1 = 边沿触发
5	PIT3	管脚中断通道3 类型选择 该位用以配置管脚中断3产生中断的信号类型 0 = 电平触发 1 = 边沿触发
4	PIT2	管脚中断通道2 类型选择 该位用以配置管脚中断2产生中断的信号类型 0 = 电平触发 1 = 边沿触发
3	PIT1	管脚中断通道1 类型选择 该位用以配置管脚中断1产生中断的信号类型 0 = 电平触发 1 = 边沿触发
2	PIT0	管脚中断通道0 类型选择 该位用以配置管脚中断0产生中断的信号类型 0 = 电平触发 1 = 边沿触发

位	名称	描述
1:0	PIPS[:0]	管脚中断端口选择 该位段选择管脚中断所用8位端口 00 = 端口0. 01 = 端口1. 10 = 端口2. 11 = 端口3.

PINEN –管脚中断反相特性使能

寄存器	SFR地址	复位值
PINEN	EAH, 所有页	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	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 = 关闭中断 1 = 低电平/下降沿触发.

PIPEN –管脚触发正相特性使能

寄存器	SFR地址	复位值
PIPEN	EBH, 所有页	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	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 = 关闭中断 1 = 高电平/上升沿触发.

PIF –管脚中断标志寄存器

寄存器	SFR地址	复位值
PIPEN	ECH, 所有页	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	通道n管脚中断标志位 如果选择边沿触发有效，当通道的产生边沿跳变信号进入中断，该标志置1，需要由软件清除。 如果选择电平触发有效，该标志根据管脚上对应的电平变换，软件无法控制该位。

EIP – 扩展中断优先级

寄存器	SFR地址	复位值
EIP0	EFH, 页0	0000_0000 b

注：EIP与EIPH结合使用，以确定每个中断源的优先级.见 表 26 2. 中断优先级设定 以正确配置中断优先级

7	6	5	4	3	2	1	0
PT2	PSPI0	PFB	PWDT	PPWM0	PCAP	PPI	PI2C0
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	PSPI0	SPI0 中断优先级低位
5	PFB	Fault Brake 中断优先级低位
4	PWDT	WDT 中断优先级低位
3	PPWM0	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	PI2C0	I2C 中断优先级低位

B – B 寄存器

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 寄存器 B寄存器是标准80C51的另一个累加器，主要用于乘法和除法指令。

CAPCON3 –输入捕获控制寄存器3

寄存器	SFR地址	复位值
CAPCON3	F1H, 所有页	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]	输入捕获通道1输入引脚选择 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 其它 = P1.2/IC0
[3:0]	CAP0[3:0]	输入捕获通道0输入引脚选择 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 其它 = P1.2/IC0

CAPCON4 –输入捕获控制寄存器

寄存器	SFR地址	复位值
CAPCON4	F2H, 所有页	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]	输入捕获通道2输入引脚选择 0000 = P1.2/IC0 0001 = P1.1/IC1 0010 = P1.0/IC2 0011 = P0.0/IC3 0100 = P0.4/IC3 0101 = P0.1/IC4 0110 = P0.3/IC5 0111 = P0.5/IC6 1000 = P1.5/IC7 其它 = P1.2/IC0

SPCR – SPI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	SSOE	从机选择输出使能位 该位搭配DISMODF (SPInSR.3) 用于定义 $\overline{SS}$ 管脚，如表 16.3- 1. 该位仅在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	SPIEN	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	MSTR	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。																																								
1:0	SPR[1:0]	SPI 时钟速率选择 Fsys = 16MHz <table><tr><th>SPR1</th><th>SPR0</th><th>Divider</th><th>SPI clock rate</th></tr><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>1M bit/s</td></tr></table> Fsys = 24MHz <table><tr><th>SPR1</th><th>SPR0</th><th>Divider</th><th>SPI clock rate</th></tr><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></table> SPR[1:0]只在主机模式有效 (MSTR = 1)。在从机模式下，时钟脚自动与主机同步最高速率为 Fsys/2	SPR1	SPR0	Divider	SPI clock rate	0	0	2	8M bit/s	0	1	4	4M bit/s	1	0	8	2M bit/s	1	1	16	1M bit/s	SPR1	SPR0	Divider	SPI clock rate	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	Divider	SPI clock rate																																							
0	0	2	8M bit/s																																							
0	1	4	4M bit/s																																							
1	0	8	2M bit/s																																							
1	1	16	1M bit/s																																							
SPR1	SPR0	Divider	SPI clock rate																																							
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 –SPI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																																				
0	SPIS[1:0]	SPI邻近字节的间隔时间选择 SPIS[1:0] 和 CPHA 在邻近字节的可选择8级SPI间隔时间项, 如下:: <table><thead><tr><th>CPHA</th><th>SPIS1</th><th>SPIS0</th><th>SPI 时钟</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td><td>0.0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0.5</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>0.0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1.0</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> SPIS[1:0] 在主机模式下有效(MSTR = 1).	CPHA	SPIS1	SPIS0	SPI 时钟	0	0	0	0.0	0	0	1	0.5	0	1	0	1.5	0	1	1	2.0	1	0	0	0.0	1	0	1	1.0	1	1	0	2.0	1	1	1	2.5
CPHA	SPIS1	SPIS0	SPI 时钟																																			
0	0	0	0.0																																			
0	0	1	0.5																																			
0	1	0	1.5																																			
0	1	1	2.0																																			
1	0	0	0.0																																			
1	0	1	1.0																																			
1	1	0	2.0																																			
1	1	1	2.5																																			

SPSR –SPI状态寄存器

寄存器	SFR地址	复位值
SPSR	F4H, 所有页	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 该位结合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	禁止 SPI 完成中断 该位用做禁止SPI完成中断, 当SPI发送完成或接收到数据移入SPI读缓存。特别用于SPI DMA操作 0 = SPI 完成中断使能当ESPI 和 EA 使能 1 = SPI 完成中断禁止

SPDR –SPI 数据寄存器

寄存器	SFR地址	复位值
SPDR	F5H, 所有页	0000_0000 b

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

位	名称	描述
7:0	SPDR[7:0]	串行外设数据寄存器 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。

AINDIDS –ADC 信道数字输入功能关闭

寄存器	SFR地址	复位值
AINDIDS	F6H, 所有页	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 信道数字输入功能关闭 0 = ADC 信道n数字输入功能打开 1 = ADC 信道n数字输入功能关闭。ADC通道n读取始终为0

EIPH –扩展中断优先级高位寄存器^[4]

寄存器	SFR地址	复位值
AINDIDS	F7H, 所有页	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

位	名称	描述
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

SCON_1 –串口1控制寄存器

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: 详见 表 15.1- 2 SMOD0_1 (T3CON.6) = 1: SM0_1/FE_1用于标示帧错误 (FE) 状态位。由软件清除。 0 = 没有帧错误 (FE) 1 = 检测到帧错误 (FE)
6	SM1_1	
5	SM2_1	串口1处理通讯模式选择 该位功能取决于串口1模式 模式0: 无效果 模式1: 该位检查有效停止位。 0 = 接收有效不管停止位是否有逻辑电平。 1 = 当接收到的数据帧停止位为1, 且接收地址与本机地址匹配或接收到的为广播地址时, 接收数据有效 模式2 或 3: 对于多机通信。 0 = 接收有效不管第9位是否有逻辑电平。 1 = 当接收到的数据帧第9位为1, 且接收地址与本机地址匹配或接收到的为广播地址时, 接收数据有效
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或3中接收到的第九位数据。模式1下, 若SM2_1 = 0则RB8_1是接收到的停止位。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下该标志由硬件在发送完8位数据后置1, 而在其它模式下, 串行发送到最后一位数据时置位。当该位中断使能, 发生中断后会转至中断子程序。该位必须由软件来清除。

位	名称	描述
0	RI_1	串口1接收中断标志 模式0下该标志由硬件置位，当串口发生帧错误接收到第8位，该位置位；模式1，2或3中接收到停止位(stop bit)，使该位置位。当SM2_1被限制的情况例外。当串口1中断使能，该位置位会跳到中断子程序运行。该位必须由软件来清除。

PDTEN –PWM 死区使能寄存器

寄存器	SFR地址	复位值
PDTEN	F9H, 所有页, TA保护	0000_0000 b

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

位	名称	描述
4	PDTCNT.8	PWM 8位死区时间计数器 详见 PDTCNT 寄存器。
2	PDT45EN	PWM4/5 死区时间使能位 仅当PWM4/5配置位互补模式，死区功能才会生效。 0 = GP4/GP5信号无延时 1 = 在GP4/GP5信号上升沿加入死区时间延时
1	PDT23EN	PWM2/3 死区时间使能位 仅当PWM2/3配置位互补模式，死区功能才会生效。 0 = GP2/GP3信号无延时 1 = 在GP2/GP3信号上升沿加入死区时间延时
0	PDT01EN	PWM0/1 死区时间使能位 仅当PWM0/1配置位互补模式，死区功能才会生效。 0 = GP0/GP1信号无延时 1 = 在GP0/GP1信号上升沿加入死区时间延时

PDTCNT –PWM死区时间计数器

寄存器	SFR地址	复位值
PWM0DTEN	FAH, 所有页,TA保护	0000_0000 b

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

位	名称	描述
7:0	PWM0DTCNT [7:0]	PWM死区时间计数器低字节 该8位与PWMnDTEN .4 组成一个9位PWM死区时间计数器PWM0DTCNT。该计数器只有当PWM在互补模式下且相应的PWMnDTEN 位设置时才有效。 $\text{PWM死区时间} = \frac{\text{PDTCNT} + 1}{F_{\text{sys}}}$ 注：在PWM运行过程中，用户不能更改PWM0DTCNT。

PMEN –PWM 输出掩码控制

寄存器	SFR地址	复位值
PMEN	FBH, 所有页	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
n	PMENn	PWMn 输出掩码使能位 0 = PWMn 信号输出 1 = PWMn 根据PMDn设定的电位值掩码

PMD –PWM 掩码数据寄存器

寄存器	SFR地址	复位值
PMD	FCH, 所有页	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
n	PMDn	PWMn 掩码数据寄存器 一旦相应的PMENn 被设置, PWMn 信号输出掩码电平值 0 = PWMn 掩码, 输出低电平 1 = PWMn 掩码, 输出高电平

PORDIS – 关闭POR

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]							
W							

位	名称	描述
7:0	PORDIS[7:0]	关闭POR 对PORDIS寄存器先写入5AH, 再写入A5H可以关闭POR功能

EIP1 –扩展中断优先级寄存器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
-	-	-	-	-	PWKT	PT3	PS_1
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	PWKT	WKT 中断优先级低位
1	PT3	定时器3 中断优先级低位
0	PS_1	串口1 中断优先级低位

EIPH1 –扩展中断优先级高位寄存器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
-	-	-	-	-	PWKTH	PT3H	PSH_1
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	PWKTH	WKT 中断优先级高位
1	PT3H	定时器3 中断优先级高位
0	PSH_1	串口1 中断优先级高位

8 普通 80C51 系统控制

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 寄存器是标准80C51累加器用于算数运算

B – B 寄存器

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 寄存器 B寄存器是标准80C51的另一个累加器，主要用于乘法和除法指令。

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]	堆栈指针 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。
-----	---------	---

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]	数据指针低字节 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。

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]	数据指针高字节 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

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	只读

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	AC	辅助进位标志 当前运算导致从半字节的低序第4位进位或借位，该位置位，否则清零。
5	F0	用户标志 0 可由用户置位或清零的通用标志。
4	RS1	寄存器块页择位 这两位用来选择R0到R7位于四页中的哪一页
3	RS0	
2	OV	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	F1	用户标志 1 可由用户软件置位或清零的通用标志。
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		

指令	CY	OV	AC	指令	CY	OV	AC
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标示根据指令的结果变化							

表 7.4- 1 指令对标志位的影响

PCON – 电源控制寄存器

寄存器	SFR地址	复位值
PCON	87H, 所有页s	POR: 0001_000b, other: 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
3	GF1	通用标志1 通用标志可由用户通过软件置位或清零
2	GF0	通用标志0 通用标志可由用户通过软件置位或清零

9 I/O 端口结构和工作模式

9.1 GPIO 模式

MS51最多有43个普通IO管脚，其中40个可位寻址的通用I/O引脚，分成5组 P0 到 P4，另外P5有7个普通IO管脚，每一个端口有它的端口控制寄存器(Px 寄存器)。端口控制寄存器的写和读有不同的意思。写端口控制寄存器设置输出锁存逻辑值，读获取端口引脚的逻辑状态。管脚有四种模式分别是准双向模式（标准8051端口结构）、推挽输出、输入和开漏模式。每一个端口通过两个特殊功能寄存器PxM1 和 PxM2来选择端口Px的I/O模式。下表指示如何选择Px.n的I/O模式。注意任何复位之后，默认的配置是高阻输入模式。

所有I/O引脚可以通过PxS寄存器里对应的位选择为TTL电平输入或施密特触发输入。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。输出斜率控制寄存器是PxSR。默认是慢斜率。如果用户想要增加I/O引脚输出速率，设置PxSR的相应位，将斜率设置成高速输出。

9.2 准双向模式

准双向模式作为标准8051的I/O结构，可以同时用作输入和输出。当端口输出逻辑高时，驱动能力较弱，同时允许外部器件将电平拉低。当引脚被拉低时有强驱动能力，会吸收大电流。在准双向I/O 结构中，有两个上拉三极管，适应不同的应用。其中一个上拉叫做特弱上拉，当端口锁定在逻辑1时，打开特弱上拉，特弱上拉有很小电流将引脚拉高

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。准双向端口结构如下所示。

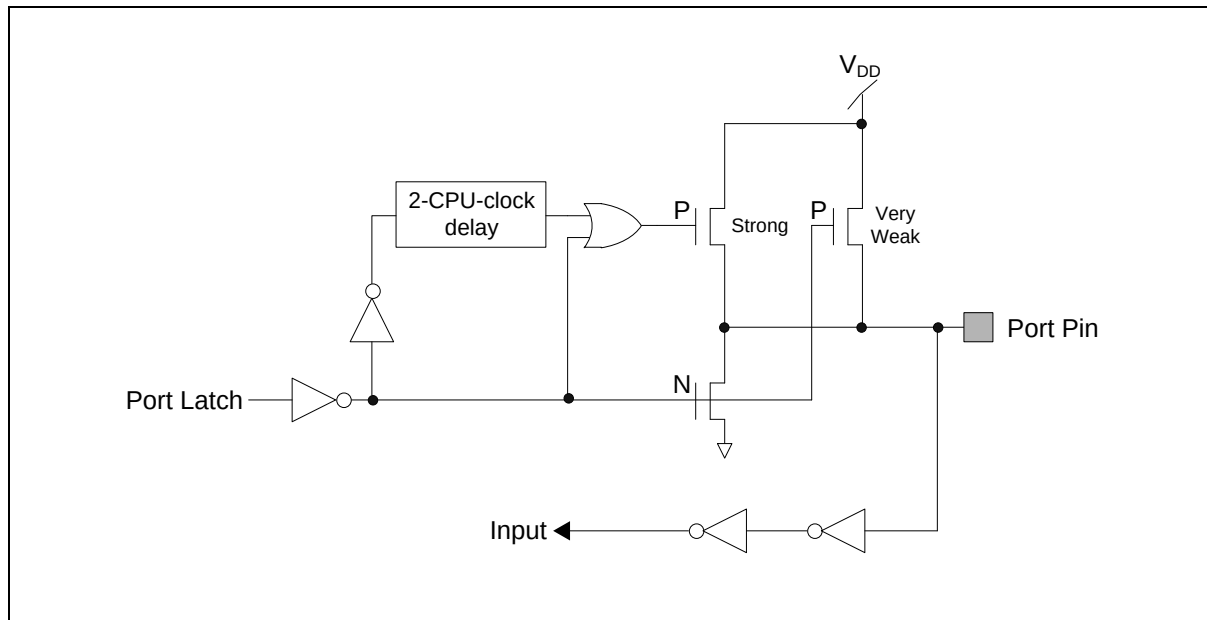


图 9.2- 1 准双向模式结构

9.3 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。

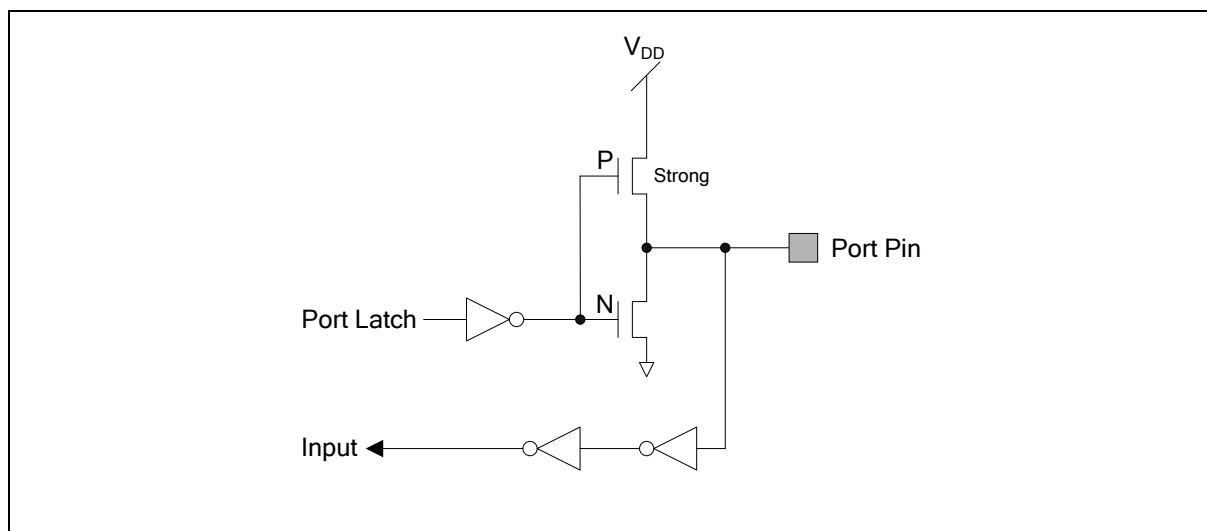


图 9.3- 1 推挽模式结构

9.4 输入模式

输入模式提供真实的高阻输入路径。虽然准双向模式也可以作为输入引脚，但是它需要相对强的输入源。输入模式的好处是减少在逻辑0时电流的消耗，如果是准双向模式，逻辑0时总是消耗来自V_{DD}的电流。用户需要注意的是，输入模式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。

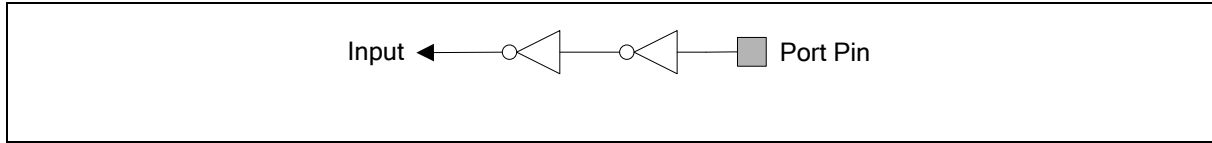


图 9.4- 1 输入模式结构

9.5 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V_{DD}。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。

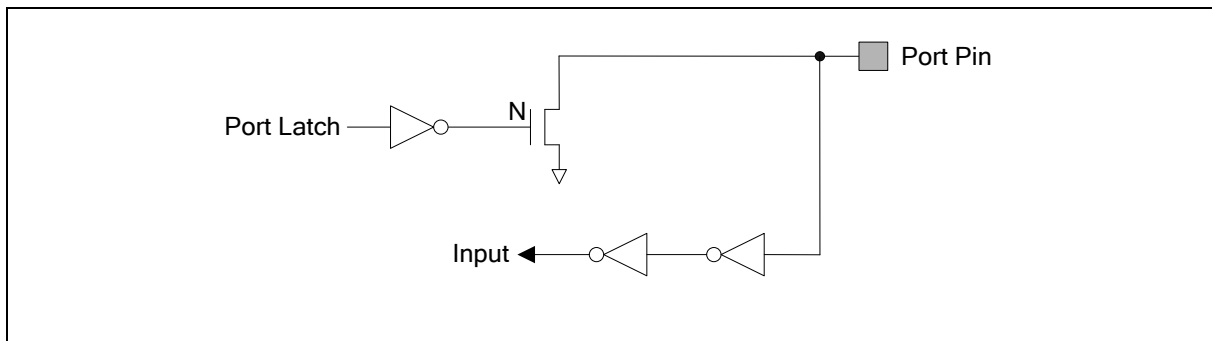


图 9.5- 1 开漏模式结构

9.6 读-修改-写 指令

从SFR或内部RAM读一个字节，修改它，并重新写回去的指令，叫做“读-修改-写指令”。当目标是一个I/O端口或一个端口位，这些指令读内部输出锁存而不是外部引脚的状态，这种指令读端口SFR的值，修改它并写回到SFR端口。所有“读-修改-写的指令”如下所列。

指令	描述
ANL	逻辑与 AND. (ANL direct, A and ANL direct, #data)
ORL	逻辑或 OR. (ORL direct, A and ORL direct, #data)
XRL	逻辑异或 OR. (XRL direct, A and XRL direct, #data)
JBC	如果位 = 1 转跳转指令并清除 (JBC bit, rel)
CPL	位取反. (CPL bit)
INC	加一指令. (INC direct)
DEC	减一指令. (DEC direct)
DJNZ	减一不为零跳转指令. (DJNZ direct, rel)
MOV	bit, C 移进位标志到位. (MOV bit, C)

CLR	bit	清位. (CLR bit)
SETB	bit	置位. (SETB bit)

最后三条指令看似不是明显的读-修改-写指令，实际它们就是读-修改-写指令。可以读整个端口锁定值，修改改变位，写入新的值。

9.7 I/O 口控制寄存器

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，输出斜率控制。所有SFRs如下所列

9.7.1 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。写驱动数据输出，所有这些寄存器都可位寻址。

P0 – 端口

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

7	6	5	4	3	2	1	0
Pn.7	Pn.6	Pn.5	Pn.4	Pn.3	Pn.2	Pn.1	Pn0.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
7:0	P0[7:0]	端口 端口是8位通用输入输出端口

9.7.2 GPIO 模式控制

这些寄存器用于控制GPIO 模式，可以配置四种模式中一种：输入，准双向，推挽或者开漏。每个管脚都可以独立配置。复位后所有管脚的默认设置都为输入模式

PnM1 – 端口模式选择1

寄存器	地址, 页, 特别说明	复位值
P0M1	B1H, 页 0	1111_1111 b
P1M1	B3H, 页0	1111_1111 b
P3M1	ACH, 页1	1111_1111 b

PnM2 – 端口模式选择2

寄存器	地址, 页, 特别说明	复位值
P0M2	B2H, 页0	0000_0000 b
P1M2	B4H, 页0	0000_0000 b
P3M2	ADH, 页0	0000_0000 b

PnM1 – 端口 n 模式选择 1

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

PnM2 – 端口 n 模式选择 2

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

PnM1.x ^[1]	PnM2.x ^[1]	I/O 类型
0	0	准双向
0	1	推挽
1	0	输入 (高阻)
1	1	开漏

注1: N = 0~5, x = 0~7

表 9.7- 1 配置不同的 I/O 模式

例如:

```
P0M1 |= 0x40;
P0M2 &= 0xBF; //Set P0.6 as input only mode
```

9.7.3 输入类型

每一个 I/O 引脚可以独立地配置成TTL输入或施密特触发输入，注意所有PxS 寄存器通过切换SFR页到页1来访问。

PnS – 端口 n 施密特触发输入

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

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

位	名称	描述
7:0	PnS[7:0]	P0 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 .

P2S – P20 设置及定时器0/1输出使能

寄存器	SFR 地址	复位值
P2S	B5H, 所有页	0000_0000 b

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

位	名称	描述
7	P20UP	P2.0 上拉使能位 0 = P2.0 禁止上拉 1 = P2.0 使能上拉 仅当RPD (CONFIG0.2)为0时该位有效。如果作为RST引脚，上拉总是被使能
3	T1OE	定时器 1 输出使能位 0 = 禁止定时器 1 输出 1 = 使能定时器 1 输出从T1引脚 注意：仅当运行在定时器模式时，定时器1输出才会被使能
2	T0OE	定时器 0 输出使能位 0 = 禁止定时器 0 输出 1 = 使能定时器 0 输出从T0引脚 注意：仅当运行在定时器模式时，定时器0输出才会被使能

位	名称	描述
0	P2S.0	P2.0 施密特触发输入选择 0 = P2.0 TTL电平输入 1 = P2.0 施密特触发输入

9.7.4 输出斜率控制

9.7.4 Output Slew Rate Control

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。

PnSR –Port n 输出斜率控制

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

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

位	名称	描述
7:0	PnSR[7:0]	P3.n 斜率控制 0 = P3.n 普通输出 1 = P3.n 高速输出

10 定时器/计数器 0 和 1

MS51定时器/计数器0 和 1 是两个16位定时器/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为高8位寄存器、TL0为低8位寄存器。同样定时器/计数器1也有两个8位寄存器，TH1 和TL1。TCON 和 TMOD 可以配置定时器/计数器0和1的模式。

通过TMOD中的C/T位来选择定时器或计数器功能。每个定时器/计数器都有选择位，TMOD的第2位选择定时器/计数器0功能，TMOD的第6位选择定时器/计数器1功能。

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计数能力系统时钟(FSYS)的12分频或增强型直接是系统时钟。在计数器模式下，每当检测到外部输入脚T0上的下降沿，计数寄存器的内容就会加一。如果在一个时钟周期采样到高电平，在下一个时钟周期采样到低电平，那么T0或T1引脚就会被认为一个由高到低的跳变

当有定时器溢出发生，定时器0和1能配置引脚自动翻转输出。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，这个功能通过CKCON寄存器的T0OE和T1OE控制位来分别使能定时器0和定时器1。当打开这个功能，输出端口在第一个定时溢出之前输出逻辑1。为确保此模式功能，C/T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：TH0(TH1)和TL0(TL1)是独立分开访问。需要特别注意，在模式0或模式1下时，当读/写TH0(TH1)和TL0(TL1)之前，必须清除TR0(TR1)来停止计时。否则将产生不可预料的结果。

10.1 定时器模式

10.1.1 模式 0 (13-位 定时器)

在模式 0, 定时器/计数器是13位的计数器。13位的计数器由TH0 (TH1) 和TL0 (TL1)的低五位组成。TL0 (TL1)的高三位被忽略。当TR0 (TR1)置位且GATE是0或INT0 (INT1) 是1时，定时器/计数器使能。GATE设置为1可以通过定时器来计算外部输入引脚INT0 (INT1)上输入脉冲的宽度。当13位的定时器计数值从1FFFH变为0000H后，定时器溢出标志TF0 (TF1) 置位，如果中断打开，此时会产生一个定时器中断。

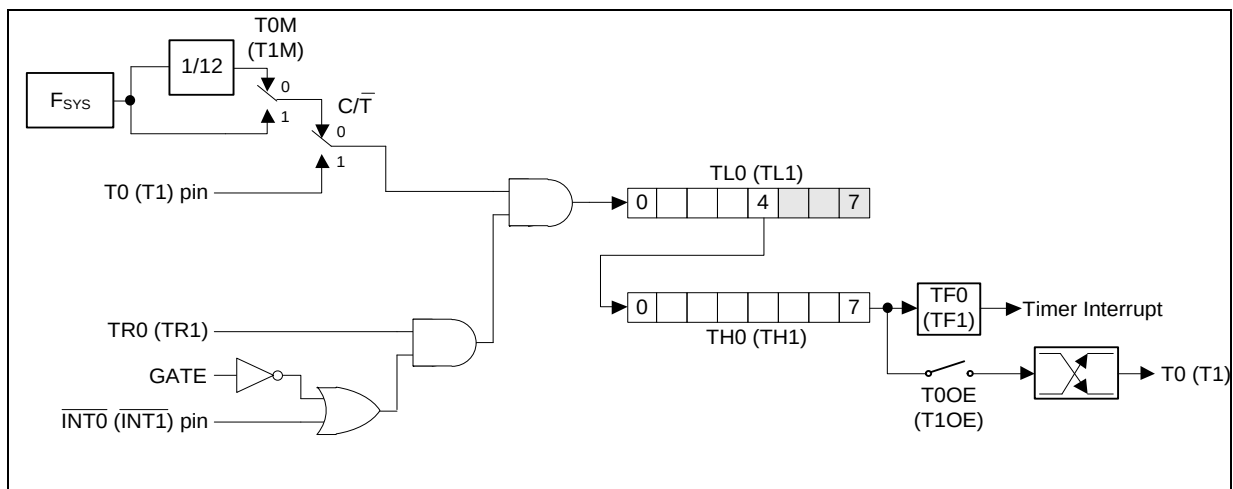


图 10.1- 1 定时器/计数器 0 和 1 的模式 0

10.1.2 模式 1 (16-位 定时器)

模式1与模式0 非常相似，只是模式1下定时器/计数器为16位的，就是说是用THx和TLx的全部16位用来计数。当计数值由FFFFH向0000H翻转后，定时器相应的溢出标志TF0（TF1）置1，如果中断使能则会产生中断。

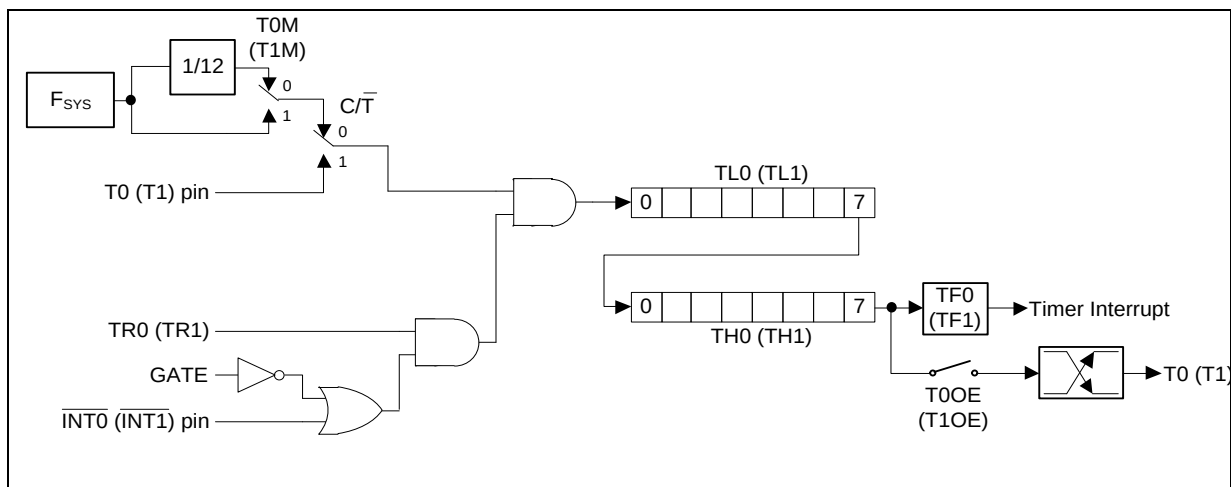


图 10.1-2 定时器/计数器 0 和 1 的模式 1

10.1.3 模式 2 (8-位 自动重载定时器)

模式2下定时器/计数器为自动重装模式。此模式下TL0 (TL1) 是一个8位的计数器, TH0(TH1)保存重装计数值。当TL0(TL1)溢出后, TCON中的TF0(TF1)标志置位且TH0(TH1)中内容重装至TL0(TL1), 然后继续计数过程。重装过程中TH0(TH1)内的值保持不变。该特征最好地适用于UART波特率发生器, 不需要连续软件介入。注意: 仅有定时器1可以用作UART的波特率源。正确设置GATE和INT0 (INT1)引脚及TR0 (TR1)位, 使能计数。GATE 和INT0 (INT1)引脚的设置与模式0和1相同。

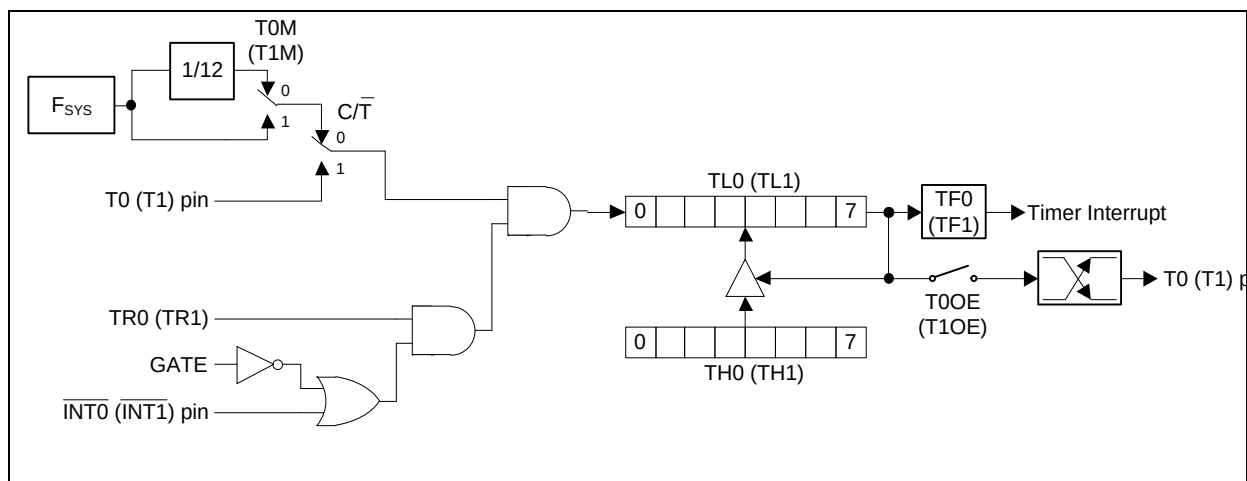


图 10.1-3 定时器/计数器 0 和 1 的模式 2

10.1.4 模式 3 (两组独立 8-位 定时器)

定时器0和定时器1的模式3有着不同的工作方式。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，模式3下TL0和TH0是2个独立的8位计数寄存器。模式3下TL0使用定时器0的控制位：如C/ $\overline{T}$, GATE, TR0, $\overline{INT0}$ 和TF0。TL0也可以用来对T0 脚上的1到0 跳变计数，由C/ $\overline{T}$ (TMOD.2)来决定。TH0 只能对时钟周期计数，并使用定时器/计数器1的控制位TR1和TF1。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。定时器1依然可以工作在模式0、1、2下，但它的灵活性受到限制。虽然基本功能得以维持，但已不能对TF1和TR1进行控制。此时定时器1 依然可以使用GATE, $\overline{INT1}$ 脚、T1M。它同样可以用作串行口的波特率发生器或其他不需要中断的应用。

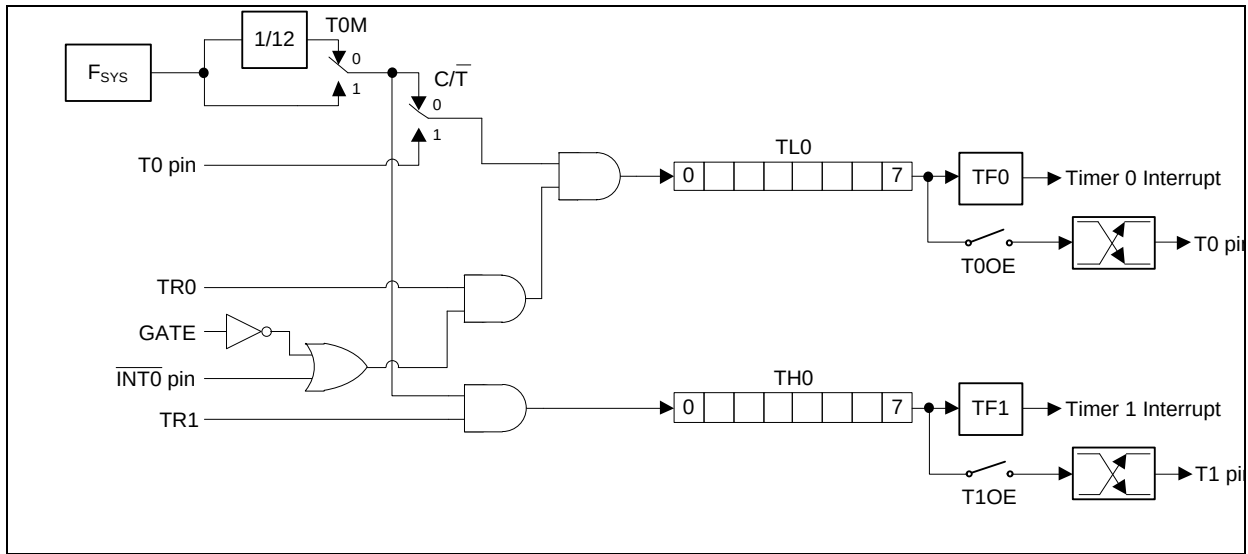


图 10.1- 4 定时器/计数器 0 的模式 3

10.2 定时器0和1控制寄存器

TMOD – 定时器0 和 1 模式寄存器

寄存器	地址, 页, 特别说明	复位值
TMOD	89H 所有页	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 门控制 0 = 不管INT1的逻辑电平, 当TR1为1时, 定时器1时钟运行 1 = 当TR1为1且INT1为逻辑1时, 定时器1时钟运行															
6	C/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><tr><th>M1</th><th>M0</th><th>定时器 1 模式</th></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: 8位带从TH1自动重载功能的定时器/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>	M1	M0	定时器 1 模式	0	0	模式 0: 13位定时器/计数器	0	1	模式1: 16位定时器/计数器	1	0	模式 2: 8位带从TH1自动重载功能的定时器/计数器	1	1	模式 3: 定时器 1 停止
M1	M0		定时器 1 模式														
0	0		模式 0: 13位定时器/计数器														
0	1		模式1: 16位定时器/计数器														
1	0	模式 2: 8位带从TH1自动重载功能的定时器/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INT0的逻辑电平, 当TR0为1时, 定时器0时钟运行 1 = 当TR1为1且INT0为逻辑1时, 定时器0时钟运行															
2	C/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1 随外部引脚T1的下降沿递增															
1	M1	定时器0 模式选择 <table><tr><th>M1</th><th>M0</th><th>定时器 0 模式</th></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: 8位带从TH0自动重载功能的定时器/计数器</td></tr><tr><td>1</td><td>1</td><td>模式 3: TL0 作为一个 8位 定时器/计数器 , TH0 作为一个 8位 定时器</td></tr></table>	M1	M0	定时器 0 模式	0	0	模式 0: 13位定时器/计数器	0	1	模式 1: 16位定时器/计数器	1	0	模式 2: 8位带从TH0自动重载功能的定时器/计数器	1	1	模式 3: TL0 作为一个 8位 定时器/计数器 , TH0 作为一个 8位 定时器
M1	M0		定时器 0 模式														
0	0		模式 0: 13位定时器/计数器														
0	1		模式 1: 16位定时器/计数器														
1	0	模式 2: 8位带从TH0自动重载功能的定时器/计数器															
1	1	模式 3: TL0 作为一个 8位 定时器/计数器 , TH0 作为一个 8位 定时器															
0	M0																

TCON – 定时器 0 和 1 控制位

寄存器	地址, 页, 特别说明	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。当执行定时器1中断服务程序时，该位由硬件自动清0。该位可由软件写1或写0
6	TR1	定时器 1 运行控制 0 = 定时器1停止。清除该位将停止定时器1并将当前计数保存在TH1 和 TL1 1 = 定时器1启动
5	TF0	定时器 0 溢出标志 在定时器0溢出时该位置1。当执行定时器0中断服务程序时，该位由硬件自动清0。该位可由软件写1或写0
4	TR0	定时器 0 运行控制 0 = 定时器0停止。清除该位将停止定时器1并将当前计数保存在TH0 和 TL0 1 = 定时器0启动
3	IE1	外部中断 1 边沿标志 当下降沿被检测到时，如果IT1=1(下降沿触发)，该标志位将被硬件置位。该位可以通过软件清零或者在进入中断服务程序后硬件清零。 如果IT1=0(低电平触发)，该标志位的电平将与INT1输入信号的逻辑电平相反。软件不能控制该位。
2	IT1	外部中断1类型选择 该位用于选择 INT1的触发类型 0 = INT1 低电平触发。 1 = INT1 下降沿触发。
1	IE0	外部中断 0 边沿标志 当下降沿被检测到时，如果IT0=1(下降沿触发)，该标志位将被硬件置位。该位可以通过软件清零或者在进入中断服务程序后硬件清零。 如果IT0=0(低电平触发)，该标志位的电平将与INT0输入信号的逻辑电平相反。软件不能控制该位。
0	IT0	外部中断0类型选择 该位用于选择 INT0的触发类型 0 = INT0 低电平触发。 1 = INT0 下降沿触发。

TL0 – 定时器 0 低字节

寄存器	地址，页，特别说明	复位值
TL0	8AH 所有页	0000_0000b

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

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

TH0 – 定时器0 高字节

寄存器	地址，页，特别说明	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 寄存器TH0是定时器0的16位计数寄存器的高字节

TL1 – 定时器 1 低字节

寄存器	地址, 页, 特别说明	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]	定时器1低字节 寄存器TL1是定时器1的16位计数寄存器的低字节。

TH1 – 定时器1 高字节

寄存器	地址, 页, 特别说明	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 寄存器TH1是定时器1的16位计数寄存器的高字节

CKCON – 时钟控制

寄存器	地址, 页, 特别说明	复位值
CKCON	8EH, 页0	1000_0000b

7	6	5	4	3	2	1	0
FASTWK	PWMCKS	T1OE	T1M	T0M	T0OE	CLOEN	-
读/写	读/写	读/写	读/写	读/写	读/写	读/写	-

位	名称	描述
5	T1OE	定时器 1 输出使能 0 = 定时器 1 输出禁止. 1 = 定时器 1 从 T1 引脚输出使能 . 注意: 定时器 1 输出, 只有配置为定时器模式才可由 T1 输出。
4	T1M	定时器 1 时钟模式选择 0 = 定时器 1 的时钟选择为 1/12 系统时钟 , 保持标准的 8051 能力 1 = 定时器 1 的时钟选择为系统时钟
3	T0M	定时器 0 时钟模式选择 0 = 定时器 0 的时钟选择为 1/12 系统时钟 , 保持标准的 8051 能力 1 = 定时器 0 的时钟选择为系统时钟
2	T0OE	定时器 0 输出使能 0 = 定时器 0 输出禁止. 1 = 定时器 0 从 T0 引脚输出使能 . 注意: 定时器 1 输出, 只有配置为定时器模式才可由 T0 输出。

11 定时器 2 和输入捕获

定时器2是一个16位的向上计数器，由高8位寄存器(TH2)和低8位寄存器(TL2)组成。通过配置寄存器RCMP2H和RCMP2L，设置CM/RL2 (T2CON.0)后，定时器2能工作在比较模式和自动重载模式下。定时器2具有3通道输入捕获模块，可用于测量输入脉冲宽度或周期。3通道捕获结果分别存放在寄存器C0H 及 C0L, C1H 及 C1L, C2H 及 C2L中。定时器2的时钟来自系统时钟的分频，总共具有8级分频，可适用于更多应用需求。当TR2 (T2CON.2) 置 1，定时器使能；TR2置0时，定时器关闭。下列寄存器用于控制定时器2功能。

11.1 定时器2结构框图

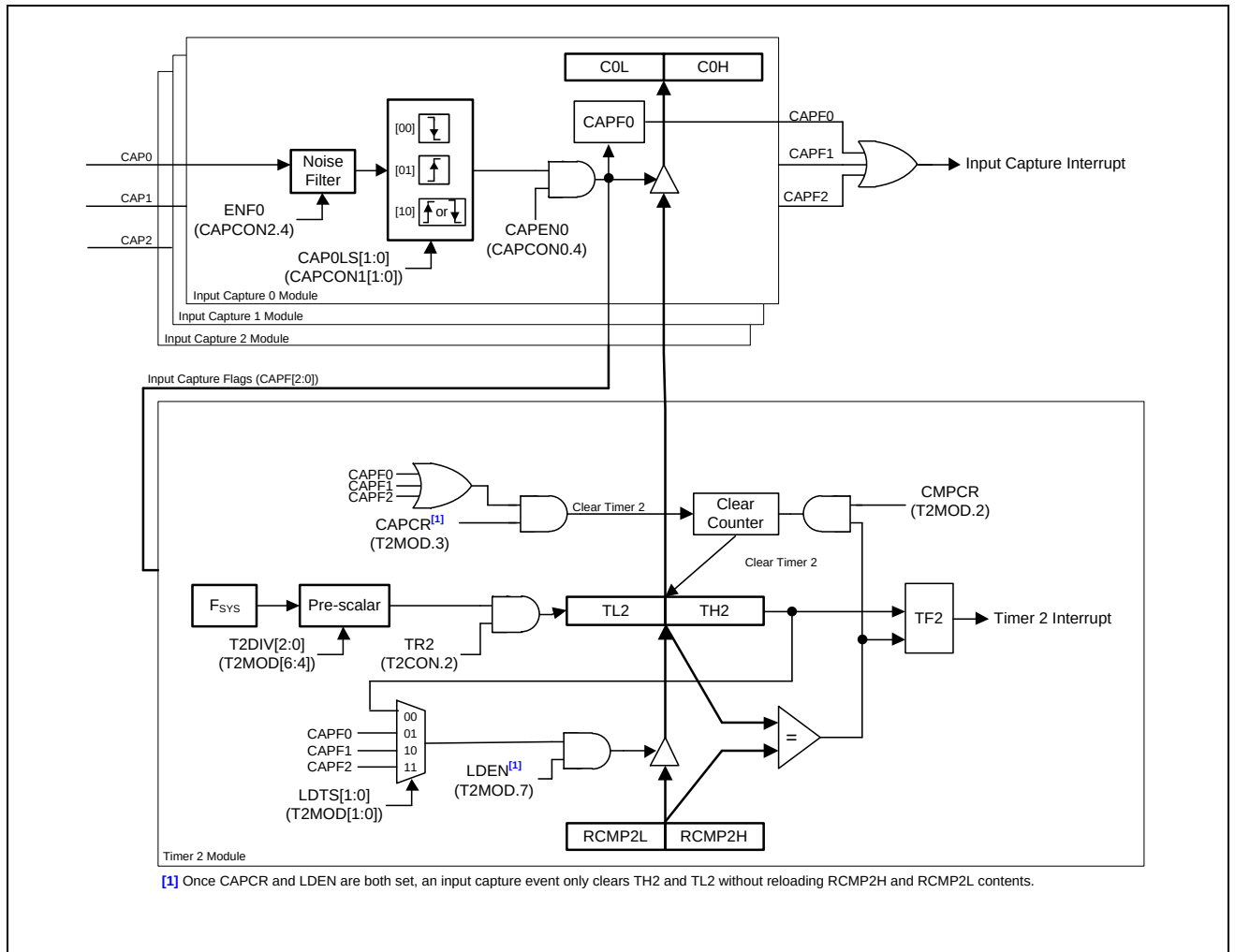


图 11.1- 1 定时器 2 框图

11.2 自动重载模式

当CM/RL2清0，定时器2配置为自动重载模式。在该模式下，RCMP2H 及 RCMP2L保存重载的数值。当LDEN置位后，一旦有触发事件发生，硬件将自动把 RCMP2H 及 RCMP2L 寄存器内的值写入 TH2 及 TL2 中。触发事件可以是定时器2溢出或是一个所配置的捕获信道有触发事件发生根据 LDTS[1:0] (T2MOD[1:0])配置。注意，一旦CAPCR (T2MOD.3) 置1，如有一个捕获事件发生，仅清除 TH2及 TL2 内的值，不会将 RCMP2H 及 RCMP2L的值载入。

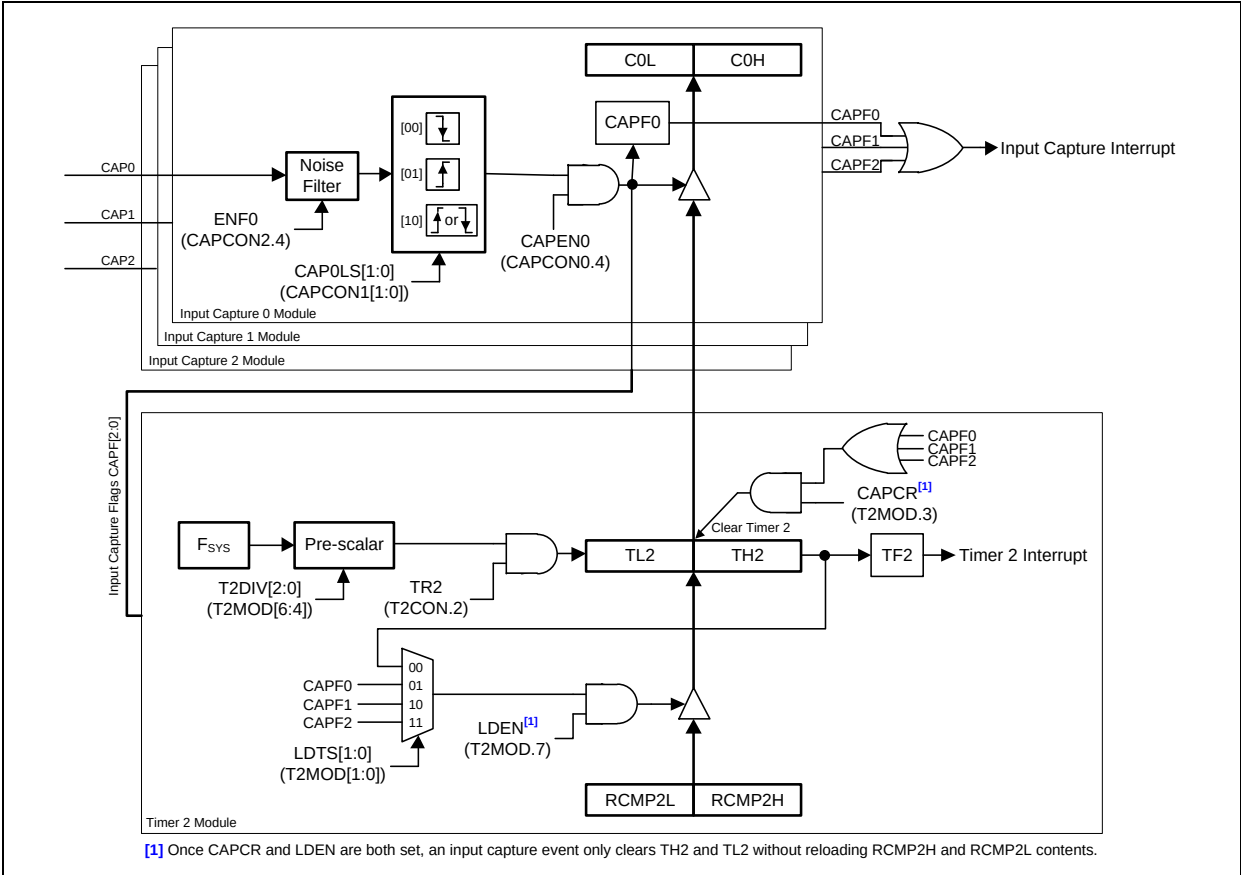


图 11.2- 1 定时器 2 自动重载模式 和输入捕获模块功能框图

11.3 比较模式

当CM/RL2置1，定时器2配置为比较器模式。在该模式下，RCMP2H 及 RCMP2L 预存待比较数据。由于定时器 2向上计数，一旦TH2 和 TL2 匹配RCMP2H 和 RCMP2L的设定值，TF2 (T2CON.7) 将会由硬件置1，用以标示发生了比较匹配事件。

如果 CMPCR (T2MOD.2) 置1，当发生比较匹配事件后，定时器2 计数器将硬件自动清0。

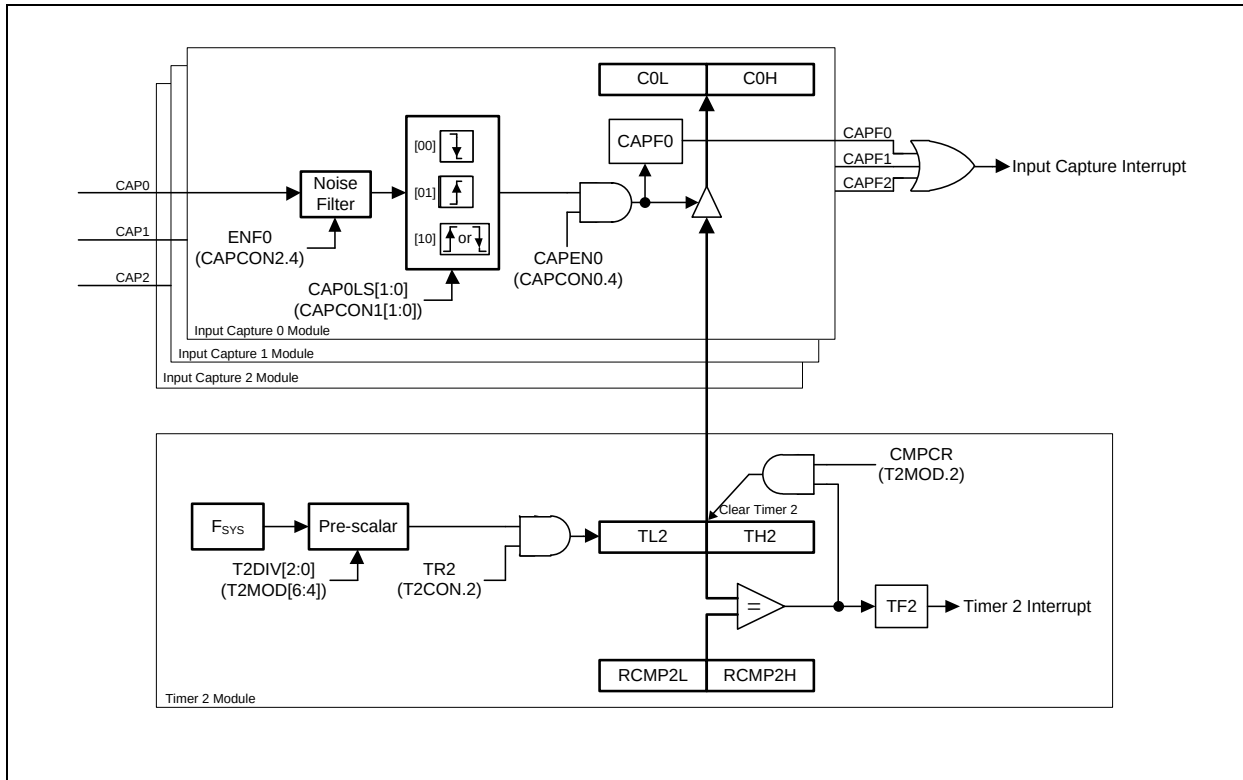


图 11.3- 1 定时器 2 比较模式和输入捕获模块功能框图

11.4 输入捕获模块

输入捕获模块依靠定时器2实现输入捕获功能。输入捕获模块通过寄存器CAPCON0~2配置来支持3组信道输入(CAP0, CAP1和CAP2)，每个输入通道的噪声滤波器可通过设置ENF0~2 (CAPCON2[6:4])使能，可滤除小于4个系统时钟的输入毛刺。每组输入捕获通道共享定时器2计数，但有自己独立的边沿检测。每个触发边沿检测可由寄存器CAPCON1的相关位独立配置，支持正边沿捕获，负边沿捕获，或双边沿捕获。在使用前，必须设置通道使能位CAPEN0~2 (CAPCON0[6:4])。

当输入捕获通道使能且所选择的边沿触发发生时，定时器2的计数值TH2和TL2将被捕获、传输并存储到捕获寄存器CnH 和 CnL。边沿触发也可硬件置位CAPFn (CAPCON0.n)，如果ECAP (EIE.2)和EA都打开，将产生中断。三组输入捕获共享一个中断向量，用户可通过检查CAPFn来确定具体哪个通道有输入捕获。这些标志必须由软件清零。

CAPCR (CAPCON2.3)用于周期计算。当设置CAPCR为1，一个捕获边沿事件发生后，TH2和TL2的值将被保存，然后硬件将自动清除定时器2的值为0000H。这样可以避免常规软件写16位计数或者算法开销。

11.5 定时器2控制寄存器

T2CON – 定时器2控制

寄存器	地址, 页, 特别说明	复位值
T2CON	C8H, 所有页	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溢出或者比较数据匹配符合, 该位置1。如果定时器2中断和全局中断使能, 该位置1将使CPU执行定时器2中断服务程序。该位硬件不能自动清除, 需要软件清0。
2	TR2	定时器 2 运行控制 0 = 定时器2计数关闭。清除该位将关闭定时器2计数, 当前计数值会保留在TH2及TL2中 1 = 定时器2计数使能。
0	CM/RL2	定时器 2 比较或者自动重载模式选择 定时器2功能模式选择位 0 = 自动重载模式 1 = 比较器模式。

T2MOD – 定时器2模式

寄存器	地址, 页, 特别说明	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 = 重载 RCMP2H 和 RCMP2L 到 TH2 和 TL2 禁用。 1 = 重载RCMP2H 和 RCMP2L 到 TH2 和 TL2 使能。

位	名称	描述
6:4	T2DIV[2:0]	定时器 2 时钟分频 000 = 定时器 2 时钟分频为1/1. 001 = 定时器2时钟分频为1/4. 010 = 定时器2时钟分频为1/16. 011 = 定时器2时钟分频为1/32. 100 = 定时器2时钟分频为1/64. 101 = 定时器2时钟分频为1/128. 110 = 定时器2时钟分频为1/256. 111 = 定时器2时钟分频为1/512.
3	CAPCR	捕获模式自动清除 该位仅当定时器2设定为自动重载模式时有效。该位使能，当捕获事件发生，在TH2和TL2内的数据移入RCMP2H 和 RCMP2L后，硬件将自动清除TH2及TL2计数寄存器的值。 0 = 捕获事件发生后定时器2计数值按之前计数值继续累加 1 = 捕获事件发生后定时器2计数值自动清0
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2个位用来选择重载触发事件。 00 = 当定时器2溢出时，自动重载 01 = 当捕获0通道事件发生，自动重载 10 = 当捕获1通道事件发生，自动重载 11 = 当捕获2通道事件发生，自动重载

RCMP2L- 定时器2重载/比较低字节

寄存器	地址，页，特别说明	复位值
RCMP2L	CAH, 页1	0000_0000 b

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

位	名称	描述
7:0	RCMP2L[7:0]	定时器 2 重载/比较 低字节 当定时器2设定为比较模式，存放比较值的低字节数据。 当设定为自动重载模式，存放重载值的低字节数据。

RCMP2H – 定时器2重载/比机器高字节

寄存器	地址, 页, 特别说明	复位值
RCMP2H	CBH, 页1	0000_0000 b

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

位	名称	描述
7:0	RCMP2H[7:0]	定时器 2 重载/比较 高字节 当定时器2设定为比较模式, 存放比较值的高字节数据。 当设定为自动重载模式, 存放重载值的高字节数据。

TL2 – 定时器2低字节

寄存器	地址, 页, 特别说明	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寄存器是定时器2的16位寄存器的低8位字节数据。

TH2 – 定时器2高字节

寄存器	地址, 页, 特别说明	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寄存器是定时器2的16位寄存器的高8位字节数据

注: TH2和TL2是独立分别访问, 强烈建议软件在清除TR2停止定时器2工作之后再读写这两个寄存器的

值。如果在运行中对寄存器直接读写，可能引发无法预测的结果。

CAPCON0 – 输入捕获控制 0

寄存器	SFR 地址	复位值
CAPCON0	E1H, 页1	0000_0000 b

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

位	名称	描述
7	-	保留
6	CAPEN2	输入捕获 通道2 使能 0 = 输入捕获通道2禁用。 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 禁用。 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禁用。 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 – 输入捕获控制器 1

寄存器	SFR 地址	复位值
CAPCON1	E2H, 页1	0000_0000 b

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

位	名称	描述
7:6	-	保留

位	名称	描述
5:4	CAP2LS[1:0]	输入捕获通道2条件选择 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条件选择 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条件选择 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
ADCDLY	E3H, 页1	0000_0000 b

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

位	名称	描述
6	ENF2	输入捕获通道2 噪声滤波使能位 0 = 关闭输入捕获通道2的噪声滤波 1 = 打开输入捕获通道2的噪声滤波
5	ENF1	输入捕获通道1 噪声滤波使能位 0 = 关闭输入捕获通道1的噪声滤波 1 = 打开输入捕获通道1的噪声滤波
4	ENF0	输入捕获通道0 噪声滤波使能位 0 = 关闭输入捕获通道0的噪声滤波 1 = 打开输入捕获通道0的噪声滤波.

C0L – 捕获通道0低字节

寄存器	SFR 地址	复位值
C0L	E4H, 页1	0000_0000 b

7	6	5	4	3	2	1	0
C0L[7:0]							
读/写							

位	名称	描述
7:0	C0L[7:0]	输入捕获通道0结果低字节 寄存器C0L是输入捕获通道0的16位结果的低字节值

C0H – 捕获通道0高字节

寄存器	SFR 地址	复位值
C0H	E5H, 页1	0000_0000 b

7	6	5	4	3	2	1	0
C0H[7:0]							
读/写							

位	名称	描述
7:0	C0H[7:0]	输入捕获通道0结果高字节 寄存器C0H是输入捕获通道0的16位结果的高字节值

C1L – 捕获通道1低字节

寄存器	SFR 地址	复位值
C1L	E6H, 页1	0000_0000 b

7	6	5	4	3	2	1	0
C1L[7:0]							
读/写							

位	名称	描述
7:0	C1L[7:0]	输入捕获通道1结果低字节 寄存器C1L是输入捕获通道1的16位结果的低字节值

C1H – 捕获通道1高字节

寄存器	SFR 地址	复位值
C1H	E7H, 页1	0000_0000 b

7	6	5	4	3	2	1	0
C1H[7:0]							
读/写							

位	名称	描述
7:0	C1H[7:0]	输入捕获通道1结果高字节 寄存器C1H是输入捕获通道1的16位结果的高字节值

C2L – 捕获通道2低字节

寄存器	SFR 地址	复位值
C2L	EDH, 页1	0000_0000 b

7	6	5	4	3	2	1	0
C2L[7:0]							
读/写							

位	名称	描述
7:0	C2L[7:0]	输入捕获通道2结果低字节 寄存器C2L是输入捕获通道2的16位结果的低字节值。

C2H – 捕获通道2高字节

寄存器	SFR 地址	复位值
C2H	EEH, 页1	0000_0000 b

7	6	5	4	3	2	1	0
C2H[7:0]							
读/写							

位	名称	描述
7:0	C2H[7:0]	输入捕获通道2结果高字节 寄存器C2H是输入捕获通道2的16位结果的高字节值。

12 定时器 3

12.1 定时器3结构描述

定时器3是一个16位自动重装载，向上计数定时器。用户可以通过配置T3PS[2:0] (T3CON[2:0])选择预分频，并写入重载值到R3H 和R3L寄存器来决定它的溢出速率。用户可以设置TR3 (T3CON.3)来开始计数。当计数跨过FFFFH，TF3 (T3CON.4)置为1，且R3H 和R3L寄存器的内容重载到内部16位计数器。如果ET3 (EIE1.1)置为1，定时器3中断服务程序被执行。当进入中断服务程序，TF3会被硬件自动清零。

定时器3同时也用作串口波特率产生定时器，详细内容见章节15.1.2波特率。

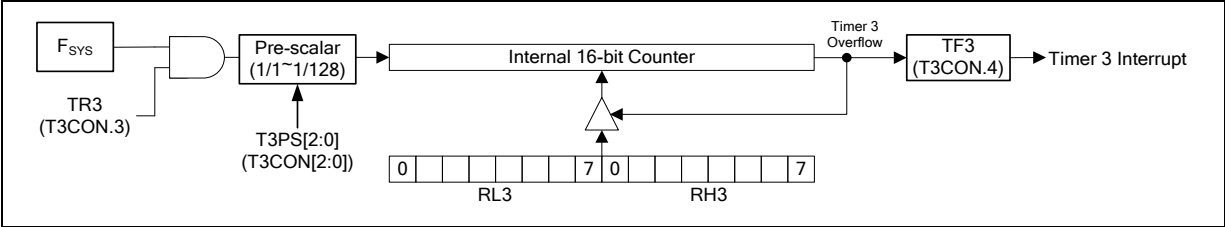


图 12.1- 1 定时器 3 框图

12.2 定时器3控制寄存器

T3CON – 定时器3控制寄存器

寄存器	地址，页，特别说明	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
4	TF3	定时器 3 溢出标志 当定时器3溢出，该位置位。当程序执行定时器3的中断服务程序，该位由硬件自动清零。该位可以通过软件置位或清零。
3	TR3	定时器 3 运行控制 0 = 定时器 3 停止. 1 = Timer 3 开始计数. 注意重载寄存器R3H 和 R3L仅在定时器3停止(TR3 位为 0)的时候才可以被写。如果TR3 位为1时取写，读写结果是不可预知的。
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	地址, 页, 特别说明	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 自动重载寄存器低字节 定时器3重载值的低字节

RH3 – 定时器3重载值高字节

寄存器	地址, 页, 特别说明	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 自动重载寄存器高字节 定时器3重载值的高字节

13 看门狗定时器 (WDT)

13.1 看门狗定时器结构描述

MS51提供一个看门狗定时器(WDT)，它可以配置成一个超时复位定时器用于复位整个设备。一旦由于外界干扰设备进入非正常状态或挂起，看门狗可以复位恢复系统。这有助于监测系统运行以提高系统可靠性。对于容易受到噪声，电源干扰或静电放电干扰的系统，是十分有用的。看门狗也可以配置成通用定时器，可以工作在空闲模式或掉电模式，用于周期中断服务作为事件定时器或连续系统监测。WDTEN[3:0] (CONFIG4[7:4])初始化WDT工作在超时复位定时器或通用定时器模式。

配置4

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

工厂默认值: 1111 1111b

位	名称	描述
7:4	WDTEN[3:0]	WDT 使能 该寄存器配置MCU执行后，WDT的动作。 1111 = WDT 禁止，WDT 可以通过软件控制用于通用定时器。 0101 = WDT 使能，作为一个超时复位定时器，且在空闲或掉电模式会停止运行。 其他 = WDT 使能，作为一个超时复位定时器，且在空闲或掉电模式会保持运行。

WDT带一个独立的分频器用于分频38.4K 内部低速时钟。分频器的时钟分频可选，来决定超时间间隔。当达到超时间隔，系统会被从空闲或掉电模式唤醒，且如果WDT中断使能会产生一个中断事件。如果WDT初始化为一个超时复位定时器，在经过一个延时周期而软件没有任何动作后会产生系统复位。

WDPS.2	WDPS.1	WDPS.0	时钟除频	WDT 溢出时间 ^[1]
0	0	0	1/1	1.66 ms
0	0	1	1/4	6.64 ms
0	1	0	1/8	13.31 ms
0	1	1	1/16	26.62 ms
1	0	0	1/32	53.25 ms
1	0	1	1/64	106.66 ms
1	1	0	1/128	213.12 ms
1	1	1	1/256	426.64 ms

注1: 该值是估算值会根据LIRC的误差有些偏差。
由于WDT定时器延时最大值的限制，对于将MS51从空闲模式或掉电模式唤醒建议用WKT功能，详见章节14自唤醒定时器（WKT）

表 13.1- 1 看门狗在不同分频下定时器溢出间隔列表

13.2 超时复位定时器

当配置 CONFIG 位 WDTEN[3:0] (CONFIG4[7:4]) 不是 FH 时, WDT 将初始化为一个超时复位定时器。如果 WDTEN[3:0] 不是 5H, WDT 在系统进入空闲或掉电模式后允许继续运行。注意当 WDT 初始化为超时复位定时器时, WDTR 和 WIDPPD 没有作用。

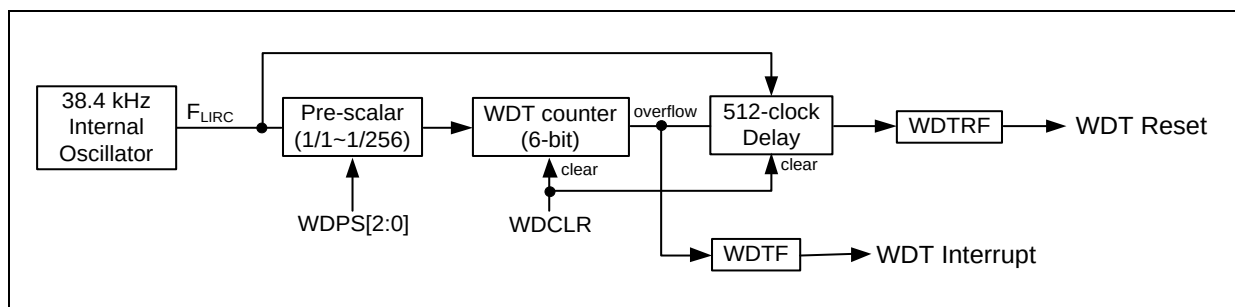


图 13.2-1 WDT 作为超时复位定时器

在设备上电后，开始执行软件代码，同时WDT开始计数。超时间隔时间通过WDPS[2:0] (WDCON[2:0])配置。当配置的超时事件发生，WDT会置位中断标志WDTF (WDCON.5)。如果WDT中断使能位EWDTE (EIE.4)和全局中断使能EA都置位，WDT中断程序被执行。同时如果系统在正常运行中，在512个LIRC时钟延时期间，系统可通过置位WDCLR来清零计数器来避免系统被WDT复位。如果在这512个LIRC时钟内没有写1到WDCLR，WDT复位将会发生。置位WDCLR位用来清零WDT计数器。如系统正常运行，该位是自我清零。一旦WDT复位发生，WDT复位标志WDTRF (WDCON.3)将会被置位，除上电复位之外的其他任何复位后，该位都保持不变。用户可以通过软件清零WDTRF。注意WDCON的所有位有写入时序要求。

看门狗定时器复位的主要应用是系统监测，这对于实时控制系统来说这很重要。在一些电源干扰，电磁干扰，CPU可能执行错误代码，或进入不可控制的状态。如果发生这些情况不加以控制，系统有可能崩溃。使用看门狗定时器用户可选择理想的“喂狗”时间来清除WDT计数器。通过指令置位WDCLR，可使程序继续运行而不看门狗定时器复位。如果干扰引起代码运行在错误的状态下，导致无法及时清除看门狗定时器，将引起芯片复位，使系统从错误的状态恢复过来。

13.3 通用定时器

看门狗定时器的另一个应用是用作简单的，长周期定时器。当CONFIG 位 WDTEN[3:0] (CONFIG4[7:4]) 是 FH，WDT初始化为通用定时器。在这种模式下，WDTR 和 WIDPD 是可以通过软件进行访问操作。

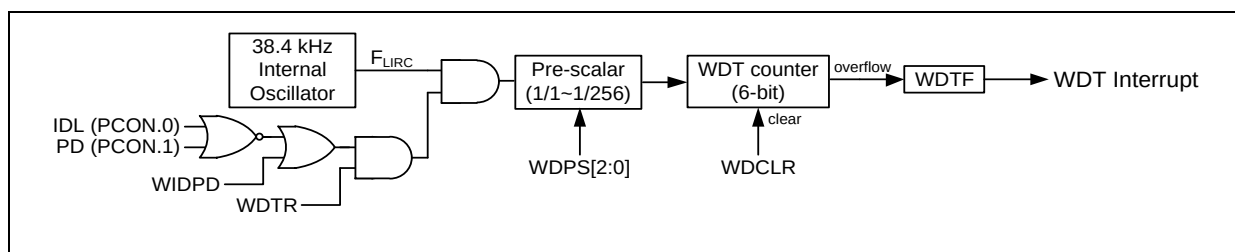


图 13.3-1 看门狗定时器框图

看门狗定时器通过设置WDTR为1开始运行，通过清零WDTR停止。当WDT配置的时间间隔到后，WDTF标志会置位。可通过软件查询WDTF标志来侦测是否超时。如果EWDT (EIE.4)和EA置位，WDT会产生中断，接着WDT 会继续计数。用户必须清零WDTF并等待下一次溢出，通过查询WDTF标志或等待中断发生。

13.4 看门狗控制寄存器

WDCON 看门狗定时器控制

寄存器	地址, 页, 特别说明	复位值
WDCON	AAH, 页0, TA保护	POR: 0000 0111b WDT: 0000 1UUUb Others: 0000 UUUUb

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

位	名称	描述
7	WDTR	WDT 运行 该位仅当控制位WDTEN[3:0] (CONFIG4[7:4])全为1时有效。这时WDT工作在通用定时器模式。 0 = WDT 禁止. 1 = WDT 使能. WDT 计数器开始运行
6	WDCLR	WDT 清除 设置该位将复位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 = 无影响. 1 = 清除 WDT 计数器. 读: 0 = WDT 计数器完全清零. 1 = WDT 计数器还没清零.
5	WDTF	WDT 超时标志 该位表示WDT计数器的溢出。该标志必须通过软件清零。
4	WIDPD	WDT 运行在空闲模式或者掉电模式 该位仅当控制位WDTEN[3:0] (CONFIG4[7:4])全为1时有效。它决定WDT作为通用定时器在空闲或掉电模式下是否保持工作。 0 = WDT 在空闲或掉电模式下停止工作 1 = WDT 在空闲或掉电模式下保持工作.
3	WDTRF	WDT 复位标志 当CPU通过WDT超时事件复位后, 该位会由硬件置位。该位建议复位之后通过软件清零。
2:0	WDPS[2:0]	WDT 时钟预分频选项 这些位决定了WDT时钟的预分频, 从1/1 到 1/256。
注 1: WDTRF在上电复位之后会被清零。在WDT复位之后置位, 在其他任何复位之后保持不变。 2: WDPS[2:0] 在上电复位之后全部置位。除上电复位外, 在其他任何复位都保持不变。		

看门狗定时器溢出时间间隔算式为
$$\frac{1}{F_{LIRC} \times \text{clock divider scalar}} \times 64$$
, FLIRC 是内部震荡的 38.4 kHz 频率. 下表所示不同预分频的超时间隔的例子

在一些低功耗的应用中, 为节省功耗, CPU在没有处理事件时常处于空闲模式。通过运行定时器0~3, 周期性的唤醒察看是否有需求响应, 然后这种空闲模式下耗电量将达到毫安(mA)级, 为了减少耗电量达到微安(μA)级的要求, 当没有需求需要相应的时候, CPU应该停留在掉电模式, 并且可以通过编程的时候间隔来唤醒。MS51配备了很有用的WDT唤醒功能, 由于基于内部38.4kHz的RC时钟源, 看门狗定时

器功耗非常低，它能够在掉电模式下计数并唤醒CPU。例程如下：

```

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

```

14 自唤醒定时器 (WKT)

14.1 WKT 结构描述

MS51有一个专用的自唤醒定时器（WKT），用于低功耗模式下的周期唤醒芯片，也可用作通用定时器。WKT保持计数在空闲或掉电模式。当WKT用作唤醒定时器时，WKT要在进入省电模式之前开启。WKT能配置两种时钟源片内38.4kHz时钟源LIRC和32.768kHz外部时钟源LXT。注意系统时钟频率必须大于WKT时钟两倍以上。如果WKT开始计数，在设备进入空闲或掉电模式下，选择的时钟源会也要保持工作。注意选择的WKT时钟源不会连同WKT的配置自动使能，用户应该手动使能选择的时钟源并等待它稳定确保操作的成功。

WKT配备了一个简单的8位自动重载向上计数定时器。它的预分频可选择从1/1 到 1/2048，通过WKPS[2:0] (WKCON[2:0])来设置。用户填重装载值到RWK寄存器来决定它的溢出速率。当计数器发生溢出RWK可以重载。CWK寄存器可以读出当前的计数值。WKTR (WKCON.3)置位开始计数。当计数器跨过FFH，WKTF (WKCON.4)置为1，将发生重载 RWK寄存器的值会重载到内部8为计数器。如果EWKT (EIE1.2) 置为1，WKT中断服务程序将被执行。

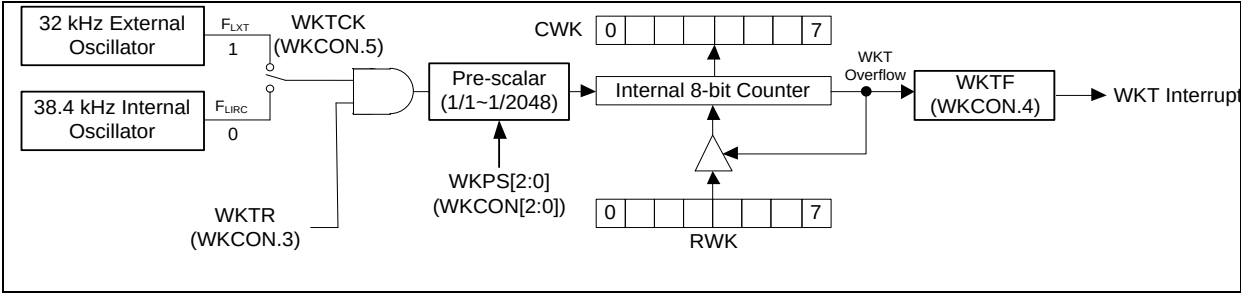


图 14.1- 1 自唤醒定时器框图

14.2 WKT 控制寄存器

WKCON – 自唤醒定时器控制器

寄存器	地址，页，特别说明	复位值
WKCON	8FH, 页0	0000_0000b

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

位	名称	描述
7:6	-	保留
	WKTCK	WKT 时钟源 该位用于选择WKT时钟源 0 = LIRC 1 = LXT
4	WKTF	WKT 溢出标志 当WKT溢出，该位置位。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，必须通过软件清零。

位	名称	描述
3	WKTR	WKT 运行控制 0 = WKT 停止. 1 = WKT 开始运行. 注意重载寄存器RWK仅在WKT停止(WKTR位为 0)的时候可以写入。否则结果是不可预知的。
2:0	WKPS[2:0]	WKT 预分频 这些位决定WKT时钟的预分频 000 = 1/1. 001 = 1/4. 010 = 1/16. 011 = 1/64. 100 = 1/256. 101 = 1/512. 110 = 1/1024. 111 = 1/2048.

RWK – 自唤醒定时器重载数据寄存器

寄存器	地址, 页, 特别说明	复位值
RWK	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 重载值 用以保存WKT的8位重载值。注意如果预分频是1/1, RWK限制不能是FFH。

CWK – 自唤醒定时器当前计数值

寄存器	地址, 页, 特别说明	复位值
CWK	86H, 页1	0000 _ 0000b

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

位	名称	描述
7:0	CWK[7:0]	WKT 当前计数值 用以保存WKT当前计数值

15 串口 (UART0 & UART1)

MS51包含两个具备增强的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。三种全双工异步模式(通用异步接收器和发送器)，模式1, 2, 和3，这意味着收发可以同时连续进行。串口接收带有接收缓存，意味着在接收的前一个数据在被读取之前，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，任何以操作SBUF的指令都将开始一次传输。

15.1 功能描述

15.1.1 工作模式

15.1.1.1 模式0

模式0是与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 发送和接收数据通过数据线RXD，因此提供为半双工通讯。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，因此任何连接串口外设模式0下都以MCU为主机模式。图 15.1- 1 串口模式0 时序图显示串口模式0传输时序图。

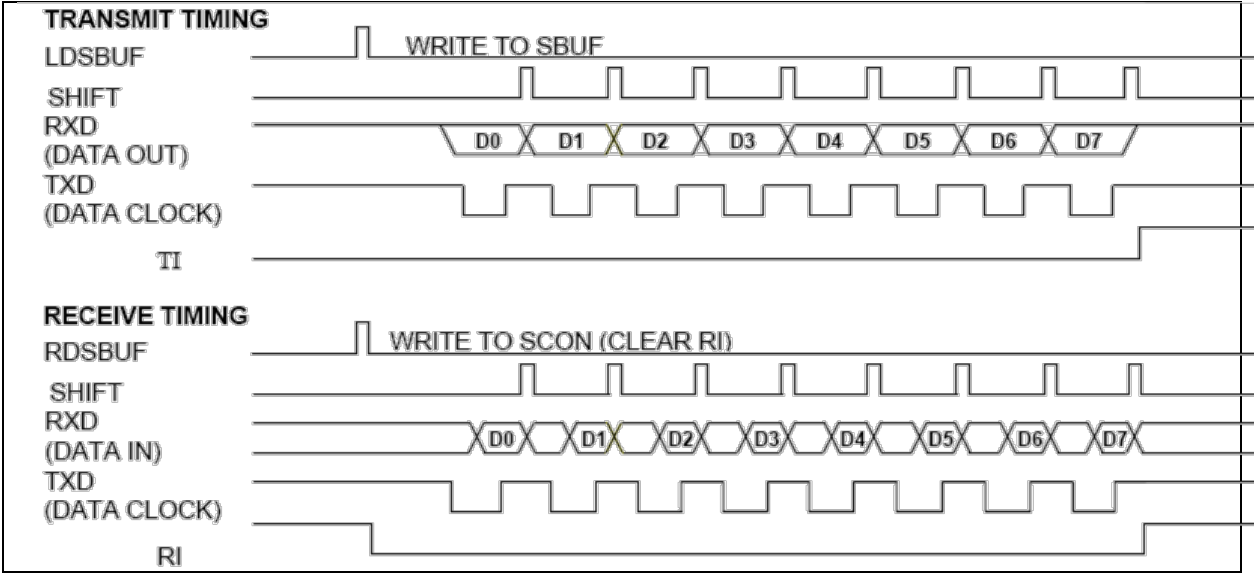


图 15.1- 1 串口模式0 时序图

如图所示，数据由双向RXD引脚进行收发，TXD引脚用来输出移位时钟。串口用移位时钟来一位位接收/发送数据与其他串口通讯。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的写入数据将会开启发送，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。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将置1。用户可以清零RI，以触发接收下一字节数据。

15.1.1.2 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PC间，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) 设置为1可使波特率加倍。图15.1- 2为串口模式1发送和接收的时序图。

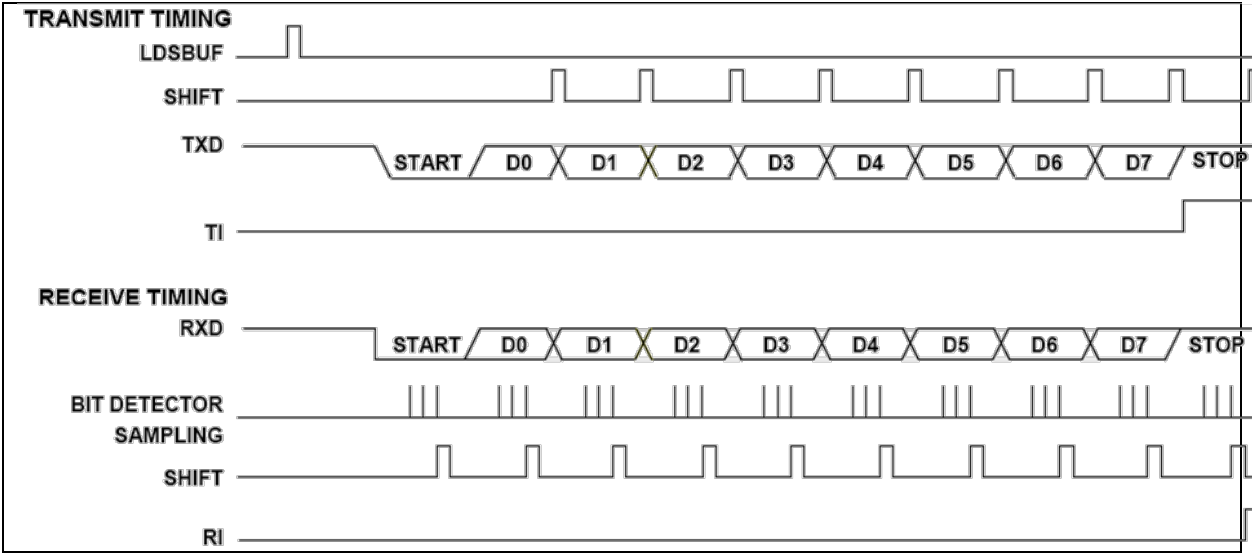


图 15.1- 2 串口模式 1 的时序图

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，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，所有位的传输速度取决于波特率。

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. 要么SM2 (SCON.5) = 0, 要么接收到停止位STOP= 1, 同时SM2 = 1且接收数据匹配“Given”或“Broadcast address”地址时。(详见 章节“多机通讯”和 章节“自动地址识别”).

如果上述条件满足，SBUF将加载到接收数据，RB8 (SCON.2) 为停止位，RI将被置1, 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。

15.1.1.3 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15.1- 3指示串口模式2的传输时序。

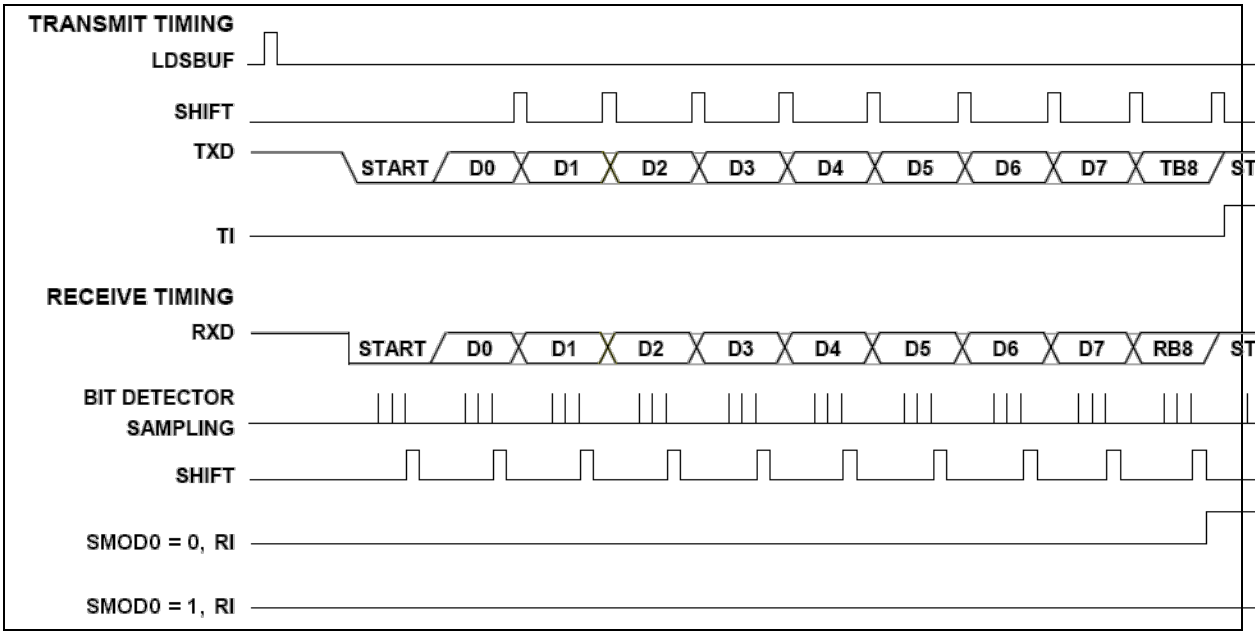


图 15.1- 3 串口模式 2 模式 3 的传输时序

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. 要么SM2(SCON.5) = 0, 要么9th位 = 1同时 SM2 = 1且接收数据匹配“Given”或“Broadcast address”地址时。(详见“多机通讯”和 章节“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。

15.1.1.4 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15.1-3 模式3的传输时序，与模式2没有不同。

15.1.2 波特率

串口的不同模式的波特率时钟源和速度是完全不同的。所有选项列表在 表 15.1- 1。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，只有采用定时器3作为唯一的时钟源。

当采用定时器1作为波特率发生器，需要关闭定时器1中断。定时器1可配置为计数器或是定时器，三种工作模式都可以。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：

描述	SM0 / SM1 (SCON[7:6])	SM2 (SCON[5])	SMOD (PCON[7])	帧 位	波特率	
0	00	0	-	8	F _{SYS} 分频 12	
		1			F _{SYS} 分频 2	
1	01	-	0	10	Time1 TM1 CKCON[3] = 0 $\frac{1}{32} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$	
					Time1 TM1 CKCON[3] = 1 $\frac{1}{32} \times \frac{F_{SYS}}{(256 - TH1)}$	
					Timer 3 $\frac{1}{32} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$	
			Time1 TM1 CKCON[3] = 0 $\frac{1}{16} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$			
			Time1 TM1 CKCON[3] = 1 $\frac{1}{16} \times \frac{F_{SYS}}{(256 - TH1)}$			
			Timer 3 $\frac{1}{16} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$			
2	10	-	0	11	F _{SYS} 分频64	
			1		F _{SYS} 分频32	
3	11	-	0	11	Time 1 ^[1] TM1 CKCON[3] = 0 $\frac{1}{32} \times \frac{F_{SYS}}{12 \times (256 - TH1)}$	
					Time 1 ^[1] TM1 CKCON[3] = 1 $\frac{1}{32} \times \frac{F_{SYS}}{(256 - TH1)}$	
					Timer 3 $\frac{1}{32} \times \frac{F_{SYS}}{\text{Pre-scale} \times (65536 - (256 \times RH3 + RL3))}$	
			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: 定时器1应配置为定时器在自动重载模式（模式2）

表 15.1- 1 串口 Port 0 模式 / 波特率描述

描述	SM0_1 / SM1_1 (S1CON[7:6])	SMOD_1 (T3CON[7])	帧位	波特率	
0	00	-	8	F _{SYS} 分频12	
1	01	0	10	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	10	0	11	F _{SYS} 分频64	
		1		F _{SYS} 分频32	
3	11	0	11	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 15.1- 2 串口 1 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
24000000	4800	64(SMOD=0)	FEC8	0.160256%
	9600	64	FF64	0.160256%
	19200	B2	FFB2	0.160256%
	38400	D9	FFD9	0.160256%
	57600	E6	FFE6	0.160256%
	115200	F3	FFF3	0.160256%
	150000	F6	FFF6	0.000000%
	166666	F7	FFF7	0.000400%
	187500	F8	FFF8	0.000000%
	214285	F9	FFF9	0.000333%
	250000	FA	FFFA	0.000000%
	300000	FB	FFFB	0.000000%
	375000	FC	FFFC	0.000000%
	500000	FD	FFFD	0.000000%
	750000	FE	FFFE	0.000000%
	1500000	FF	FFFF	0.000000%
16000000	4800	30	FF30	0.160256%
	9600	98	FF98	0.160256%
	19200	CC	FFCC	0.160256%
	38400	E6	FFE6	0.160256%
	57600	EF	FFEF	2.124183%
	115200	F7	FFF7	-3.549383%
	200000	FB	FFFB	0.000000%
	250000	FC	FFFC	0.000000%
	333333	FD	FFFD	0.000100%
	500000	FE	FFFE	0.000000%
	1000000	FF	FFFF	0.000000%

15.1.3 帧错误检测

帧错误检测用于异步模式 (模式 1, 2 和 3)。当由于总线干扰或争夺, 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。

15.1.4 多机通讯

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, 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 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。

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。

15.1.5 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。当多机通信特征使能时 (SM2置位), 就使能自动地址识别。

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 1, 从机 0:

SADDR = 11000000b

SADEN = 11111101b

Given = 110000X0b

例 2, 从机 1:

SADDR = 11000000b

SADEN = 11111110b

Given = 1100000Xb

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要求位0为“0”而忽略位1，从机1要求位1为“0”而位0被忽略。一个从机0唯一的地址11000010B，由于从机1要求位1为0。一个从机1唯一的地址将自1位11000001b将排除从机0。这两个从机可以选择在同一时间，地址位0 = 0（从机0）和第1位= 0（从机1）。因此，使用广播地址(Boadcast address) 11000000b就可以同时寻址。

更复杂应用可用于排除从机0之后，选择从机1或2:

范例 1, 从机 0:

SADDR = 11000000b

SADEN = 11111001b

Given = 11000XX0b

范例 2, 从机 1:

SADDR = 11100000b

SADEN = 11111010b

Given = 11100X0Xb

范例 3, 从机 2:

SADDR = 11000000b

SADEN = 11111100b

Given = 110000XXb

在上面的例子中，3个从机的分别是在地址的低3位。从机0要求位0 = 0，它可用11100110b解决。从机1要求位1= 0，它可用11100101b识别。从机2要求位2= 0，其独立的地址是11100011b。要选择从机0和1，去除从机2，可使用地址11100100b，因为它是必要的第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 = 1111111Xb

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。这将对所有“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。

15.2 串口控制寄存器

SCON – 串行端口控制

寄存器	地址, 页, 特别说明	复位值
SCON	98H, 所有页, 可位寻址	0000_0000 b

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

位	名称	描述
7	SM0/FE	串行端口选择 SMOD0 (PCON.6) = 0: 见 表 15.1-1 详细描述. SMOD0 (PCON.6) = 1: SM0/FE 位被用来做帧错误 (FE) 状态标志. 可被软件清除.
6	SM1	0 = 帧错误 (FE) 未发送错误. 1 = 检测到帧错误 (FE).
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“给定”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“给定”或“广播”地址时, 接收才有效。
4	REN	接收使能 0 = 串口0接收禁止。 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
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。

S1CON – 串行口1控制 (可位寻址)

寄存器	SFR 地址	复位值
S1CON	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 详见 表 15.1-2
6	SM1_1	SMOD0_1 (T3CON.6) = 1; SM0_1/FE_1 用于标示帧错误 (FE) 状态。由软件清除。 0 = 没有帧错误 (FE) 1 = 检测到帧错误 (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 = 接收仪在接收停止位为逻辑1,同时接收数据与GIVEN或BROADCAST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 =接收仪在第9位为逻辑1和接收数据与GIVEN或BROADCAST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, 若SM2=0则RB8是接收到的停止位。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。
0	RI_1	串口1接收中断标志位 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。

PCON – 电源控制寄存器

寄存器	地址, 页, 特别说明	复位值
PCON	87H, 所有页s	POR: 0001_000b, 其它: 000U_0000b

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

位	名称	描述
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 详见 表 15.1- 1
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

寄存器	地址，页，特别说明	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。 详见 表 15.1- 1串口 Port 0 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

SBUF – 串口0数据缓存

寄存器	地址，页，特别说明	复位值
SBUF	99H, 页0	0000_0000 b
SBUF1	9AH, 页0	0000_0000 b

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

位	名称	描述
---	----	----

位	名称	描述
7:0	SBUF[7:0]	串口n数据缓存寄存器 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, 这是个接收寄存器。 每次向SBUF写入一字节数据, 将启动一次数据传输。

SADDR – 从机0地址

寄存器	地址, 页, 特别说明	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 该字节定义微控制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	地址, 页, 特别说明	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]	从机0地址掩码。 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。

SADDR1 – 从机1地址

寄存器	地址, 页, 特别说明	复位值
SADDR1	BBH, 页0	0000_0000 b

7	6	5	4	3	2	1	0
SADDR1[7:0]							
读/写							

位	名称	描述
7:0	SADDR1[7:0]	从机1地址 该字节定义微控制器自身的从机地址以用于串口1多机通信.

SADEN1 – 从机1地址掩码

寄存器	地址, 页, 特别说明	复位值
SADEN1	BAH, 页0	0000_0000 b

7	6	5	4	3	2	1	0
SADEN1[7:0]							
读/写							

位	名称	描述
7:0	SADEN1[7:0]	从机1地址掩码. 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.

IE – 中断使能

寄存器	地址, 页, 特别说明	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
4	ES	使能串口0中断 0 = 串口0中断禁止 1 = 串口0通过 TI (SCON.1) 或 RI (SCON.0)置1可产生中断使能

EIE1 – 拓展中断使能寄存器 1

寄存器	地址, 页, 特别说明	复位值
EIE1	9CH, 页0	0000_0000 b

7	6	5	4	3	2	1	0
EFB1	EPWM1	EI2C1	ESPI1	EHFI	EWKT	ET3	ES1

读/写	读/写	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 = 串口1中断禁止 1 = 串口1通过TI_1 (S1CON.1) 或 RI_1 (S1CON.0) 置1可产生中断使能

16 串行外围总线(SPI)

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 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 $F_{SYS}/4$ ，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。

16.1 结构功能图

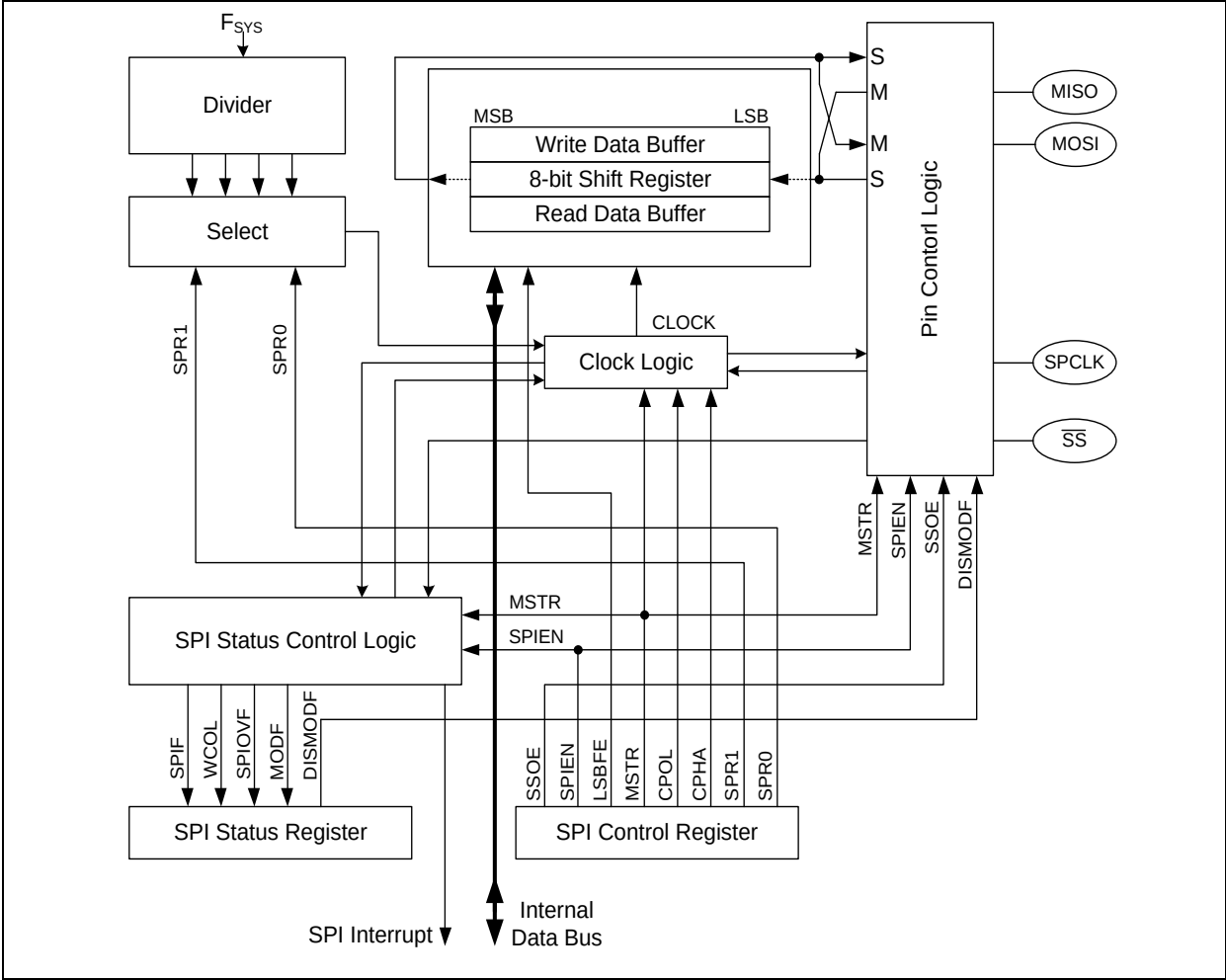


图 16.1- 1 SPI 结构功能图

16.2 功能描述

图 16.1- 1展示了SPI的结构功能。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。无论传送数据或是接收数据都是双缓存。在传输数据时，在前一个数据发送完成前，不可以写入下一笔数据到移位寄存器。接收端有并行读数据缓存,所以在移位寄存器接收第二个数据时，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是一个主机设备的输出引脚，从机设备的输入引脚。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个移位时钟周期，在总线上交换一个字节数据。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。当需要访问任何从机时，该从机的此信号脚必须保持低。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，(详见 章节)，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。

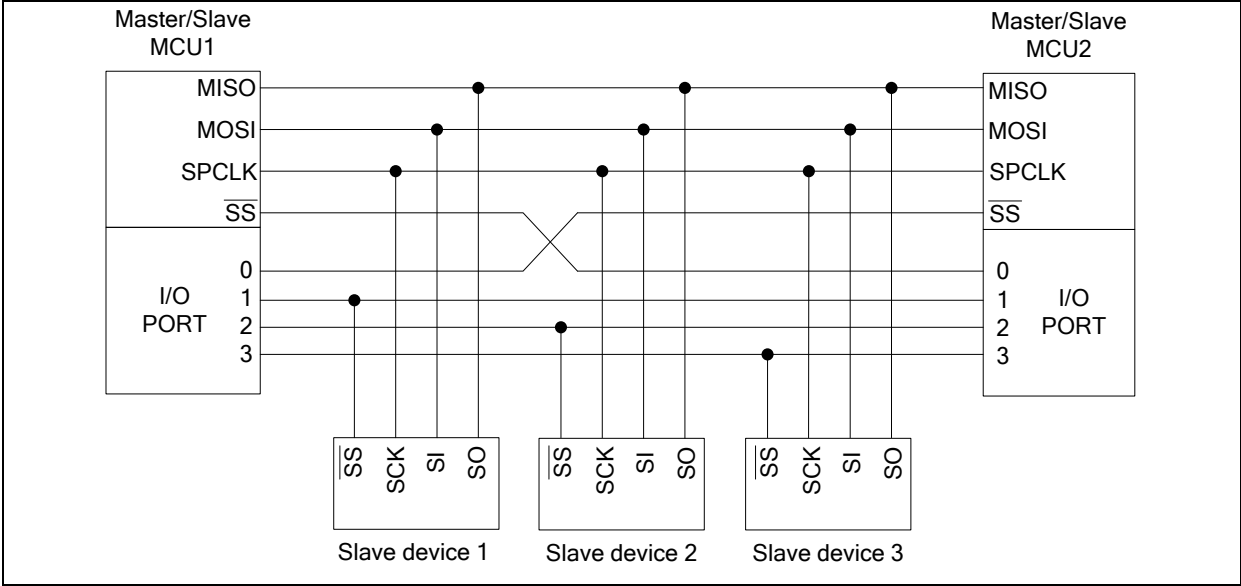


图 16.2- 1 SPI 多主机，多从机连接图

图16.2- 1 SPI多主机，多从机连接图典型的SPI设备连接图。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 可以任意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。

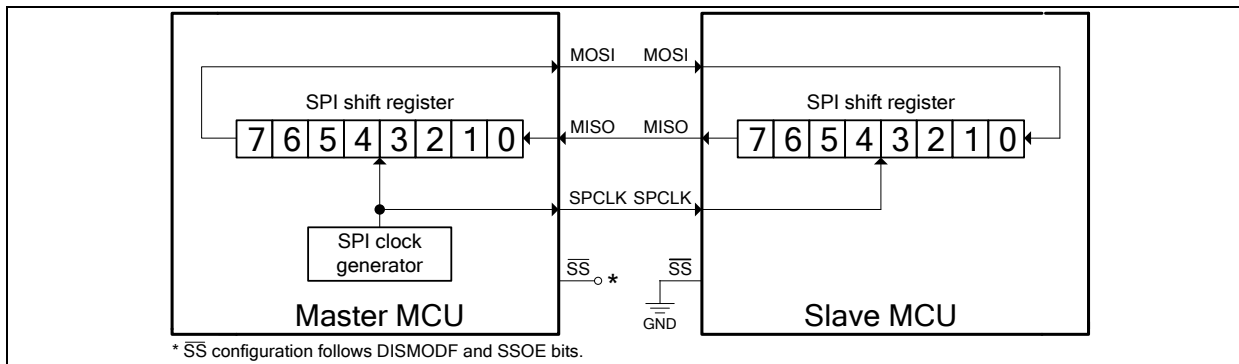


图 16.2- 2 SPI 单主机，单从机连接图

图16.2- 2 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 (SPInCR.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 (SPInCR), SPI 状态寄存器 (SPInSR), SPI 数据寄存器 (SPInDR) 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。

16.2.1 操作模式

16.2.1.1 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。

16.2.1.2 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，SS管脚也变为输入脚。当从机设备的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，SS管脚都需要保持低电平状态。如果SS变为高电平，SPI将被迫进入闲置状态。如果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，数据进入位移寄存器。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。

16.2.2 时钟格式和数据传输

为了适应各种各样的同步串行外设，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 16.2- 3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。

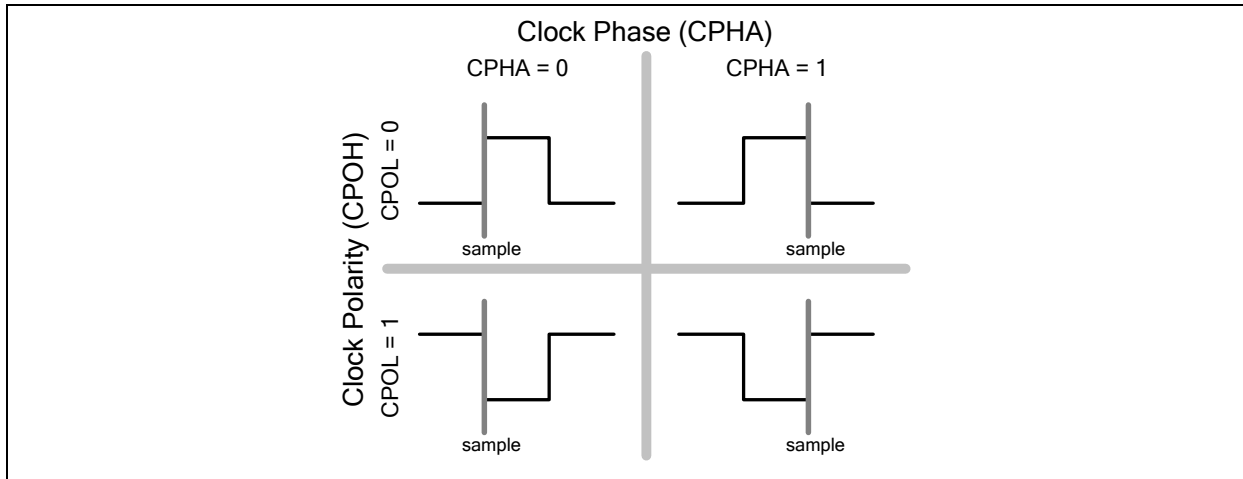


图 16.2- 3 SPI 时钟格式

在SPI传输中，总是由主机启动传输。如果SPI被选定作为主模模式（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。传出一个字节的会接收一个字节的的内容，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。

关于从机模式下， $\overline{SS}$ 信号需要注意。如图 16.2- 3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。

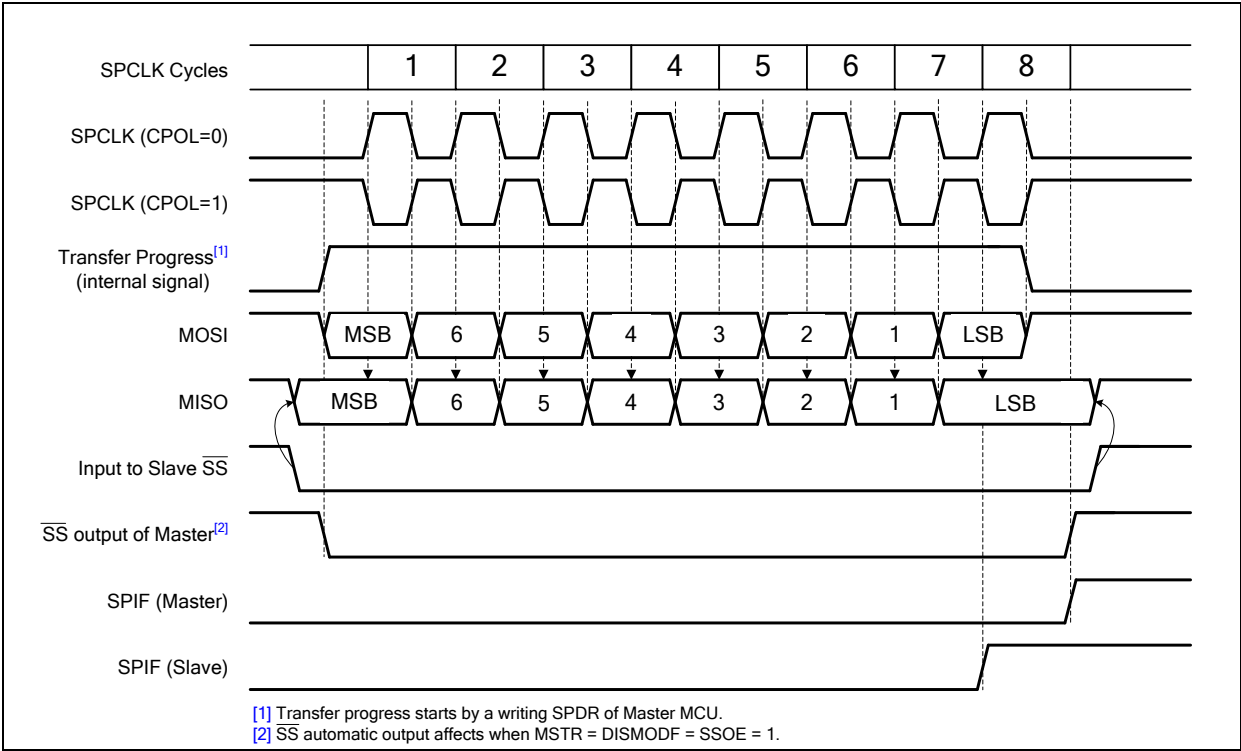


图 16.2- 4 SPI时钟和数据格式（CPHA = 0）

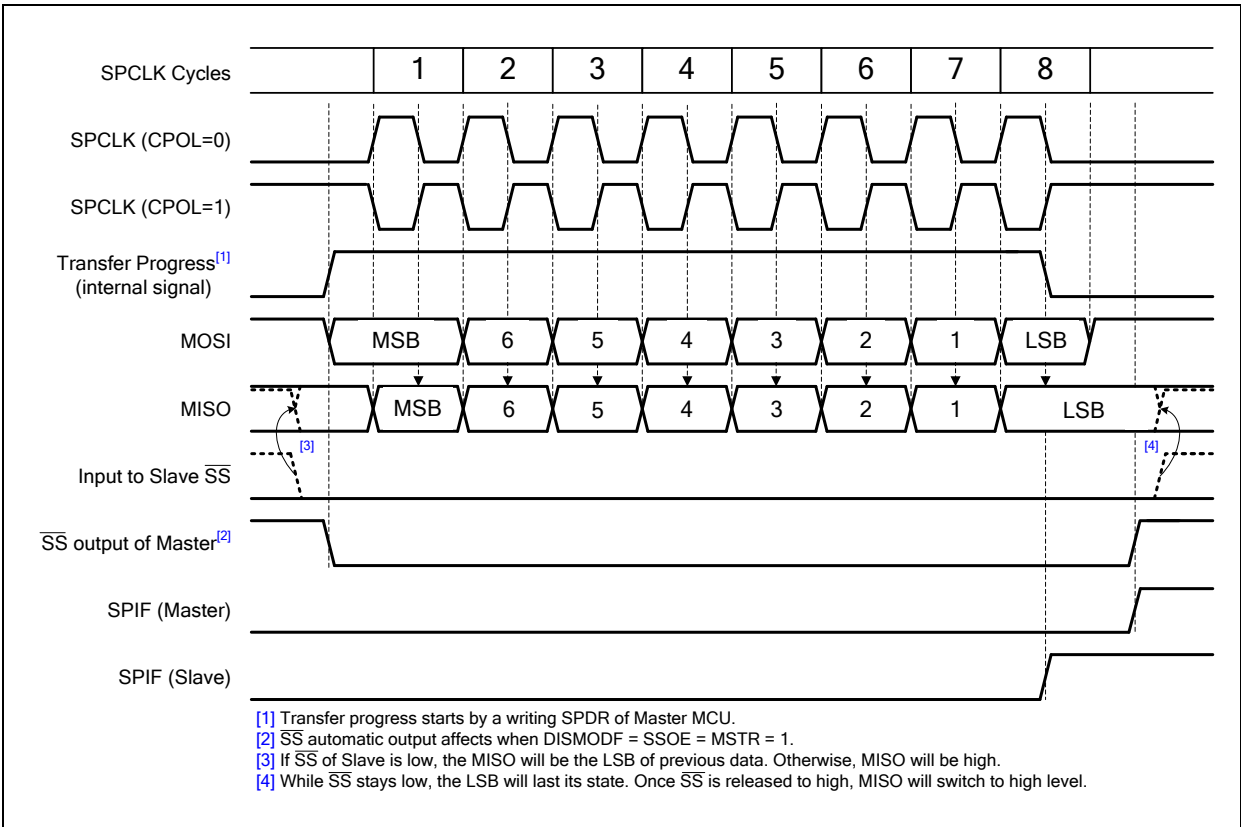


图 16.2- 5 SPI 时钟和数据格式（CPHA = 1）

16.2.3 从机选择引脚SS配置

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 (SPInSR.3) 和SSOE (SPInCR.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。

16.2.4 模式故障侦测

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SPInCR0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 (SPInSR.4)置1，如果之前已打开中断ESPI (EIE .6) 和EA置1，则会进入中断向量。

16.2.5 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SPInDR，将发生写冲突错误。发送数据时SPInDR是一个双缓存，任何写入SPInDR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(SPInSR.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，但写冲突通常会是一个从机错误，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。

16.2.6 溢出错误

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，这样的才不会产生溢出错误，反之，将产生溢出错误。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。当发生溢出错误时，SPIOVF (SPInSR.5)会被硬件置1。如果中断使能，会进入SPIOVF中断请求。图16.2- 6表示接收数据与溢出错误之间的关系。

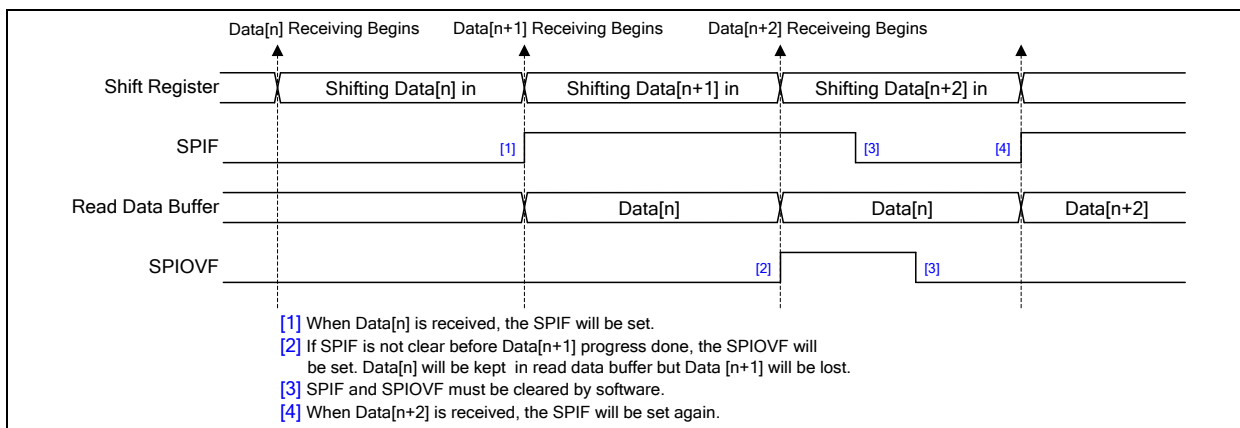


图 16.2- 6 表示接收数据与溢出错误之间的关系

16.2.7 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。

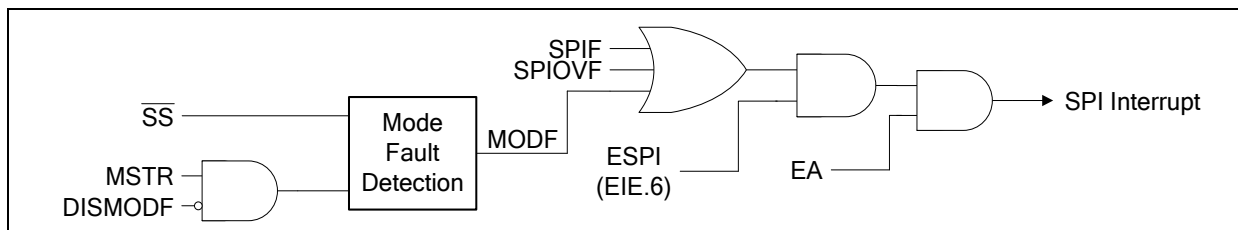


图 16.2- 7 SPI 中断请求

16.3 SPI 控制寄存器

SPCR – SPI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	SSOE	从机选择输出使能位 该位搭配DISMODF (SPInSR.3) 用于定义 $\overline{SS}$ 管脚，如表 16.3- 1. 该位仅在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	SPIEN	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	MSTR	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。																																								
1:0	SPR[1:0]	SPI 时钟速率选择 Fsys = 16MHz <table><tr><th>SPR1</th><th>SPR0</th><th>Divider</th><th>SPI clock rate</th></tr><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>1M bit/s</td></tr></table> Fsys = 24MHz <table><tr><th>SPR1</th><th>SPR0</th><th>Divider</th><th>SPI clock rate</th></tr><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></table> SPR[1:0]只在主机模式有效 (MSTR = 1)。在从机模式下，时钟脚自动与主机同步最高速率为 Fsys/2	SPR1	SPR0	Divider	SPI clock rate	0	0	2	8M bit/s	0	1	4	4M bit/s	1	0	8	2M bit/s	1	1	16	1M bit/s	SPR1	SPR0	Divider	SPI clock rate	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	Divider	SPI clock rate																																							
0	0	2	8M bit/s																																							
0	1	4	4M bit/s																																							
1	0	8	2M bit/s																																							
1	1	16	1M bit/s																																							
SPR1	SPR0	Divider	SPI clock rate																																							
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																																							

SPIInCR1 – SPI控制寄存器1

寄存器	SFR 地址	复位 值
SPCR2	F3H, 页0	0000_0000 b

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

位	名称	描述																																				
7:2	-	保留																																				
1:0	SPIS[1:0]	SPI邻近字节的间隔时间选择 SPIS[1:0] 和 CPHA 在邻近字节的可选择8级SPI间隔时间项, 如下: <table><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> SPIS[1:0] 在主机模式下有效(MSTR = 1).	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																																			

DISMODF	SSOE	主机模式(MSTR = 1)	从机 模式 (MSTR = 0)
0	X	SS作为模式错误输入脚	SS作为从机选择输入脚
1	0	通用I/O	
1	1	自动 SS 输出	

表 16.3- 1 从机选择脚定义

SPSR – SPI状态寄存器

寄存器	SFR 地址	复位 值
SPSR	F4H, 所有页	0000_0000 b

7	6	5	4	3	2	1	0
SPIF	WCOL	SPIOVF	MODF	DISMODF	TXBFF	-	-
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SPIInDR写入

位	名称	描述
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 该位结合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) 详见表 16.3- 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	TXBFF	SPI TX 缓存满标志 0 = SPI TX 缓存为空 1 = SPI TX 缓存已满

SPDR – SPI数据寄存器

寄存器	SFR 地址	复位 值
SPI0DR	F5H, 页 0	0000_0000 b

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

位	名称	描述
7:0	SPInDR[7:0]	串行外设数据寄存器 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。

17 I²C 总线(I²C)

MS51 提供了两个内部集成电路(I²C)总线作为一种串行通信方式，用在 MCU 与 EEPROM，LCD 模块，温度传感器等等之间控制。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)。

17.1 功能描述

对于双向传输操作，SDA 及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SCL，SDA的输出锁存在逻辑1的状态。

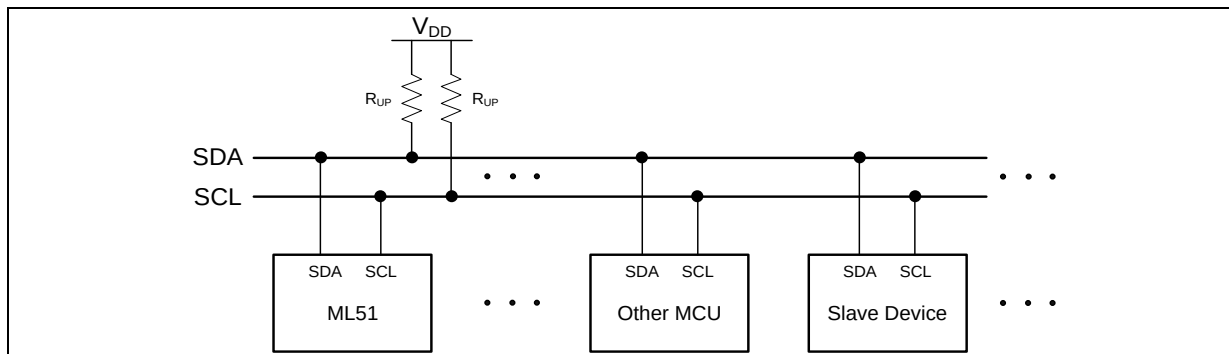


图 17.1- 1 I²C 总线连接

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SDA脚输出数据后，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，可以拉住时钟线保持低，让传输挂起。接收端释放时钟线SCL以后，传输继续。

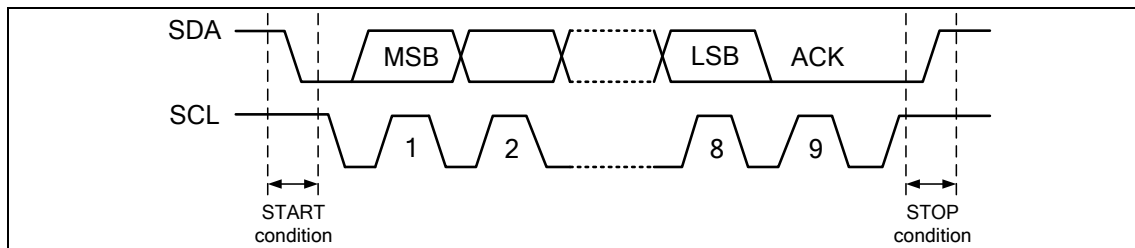


图 17.1- 2 I²C 总线协议

17.1.1 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SCL为高时，数据线SDA由高电平至

低电平的跳变被认为是起始START标志。时钟SCL为高时，数据线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。²I²C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信号（Sr）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。

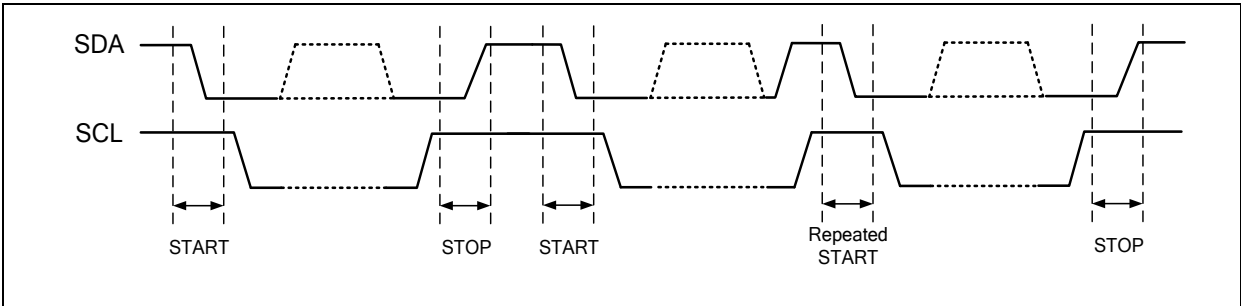


图 17.1- 3 起始条件, 重复起始条件和停止条件

17.1.2 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。当一个从机已被寻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 17.1- 4为主机发送数据给一个7位地址从机。主机寻址一个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。

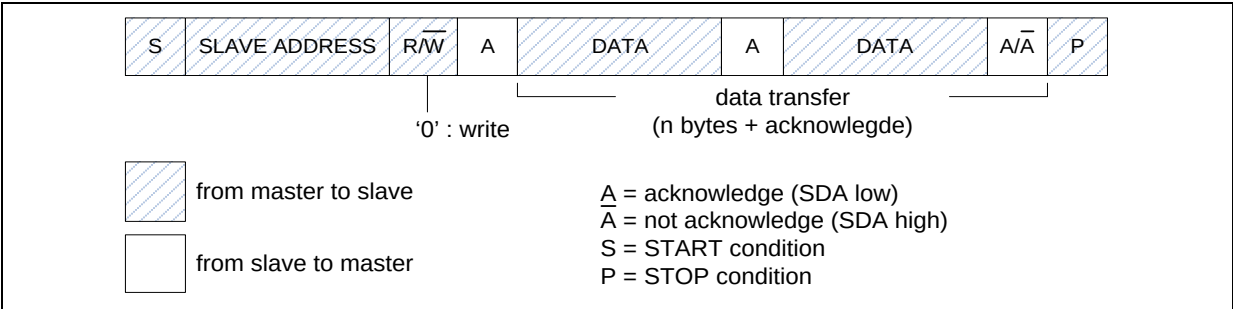


图 17.1- 4 主机发送数据给一个 7 位地址从机

图 17.1- 5为主机读取数据从一个7位地址从机。主机寻址一个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。

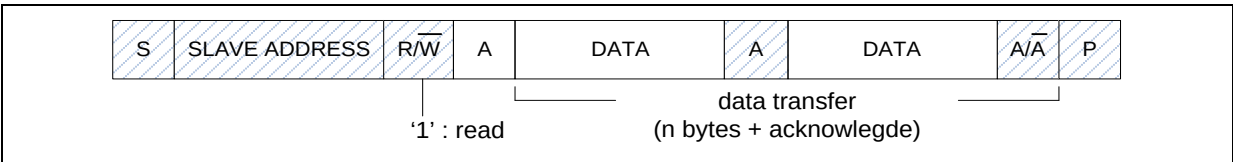


图 17.1- 5 主机读取数据从一个 7 位地址从机

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。

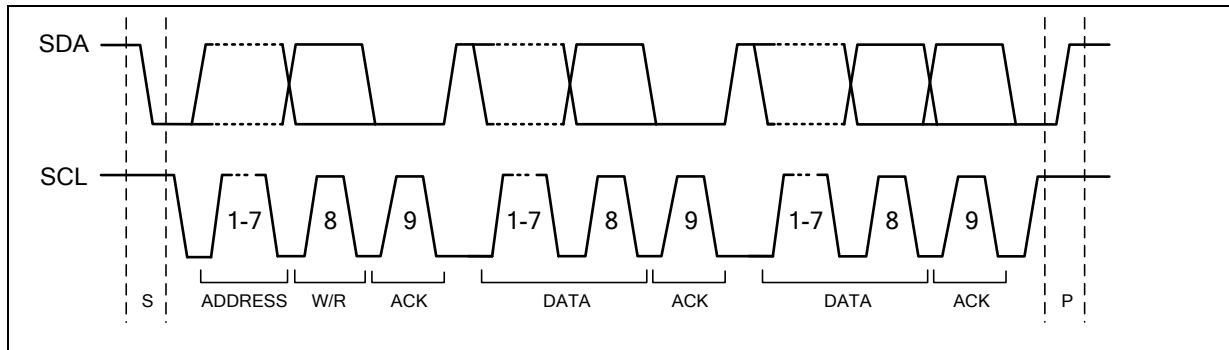


图 17.1- 6 一个传输中的 I2C 数据格式

在数据传输过程中，在时钟高电平时，SDA需要保持数据内容不能更改。只有在SCL为低时，SDA内容可以改变。

17.1.3 应答Ack

每字节传输SCL第9个脉冲用于传输应答位 (ACK)。通过将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，将自身切换到未寻址从机模式，从而无法接收更多数据字节。这时从机将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，从机端将切换为未寻址从机模式，并释放SDA线，以便主机直接发停止位STOP 或重开始位START。

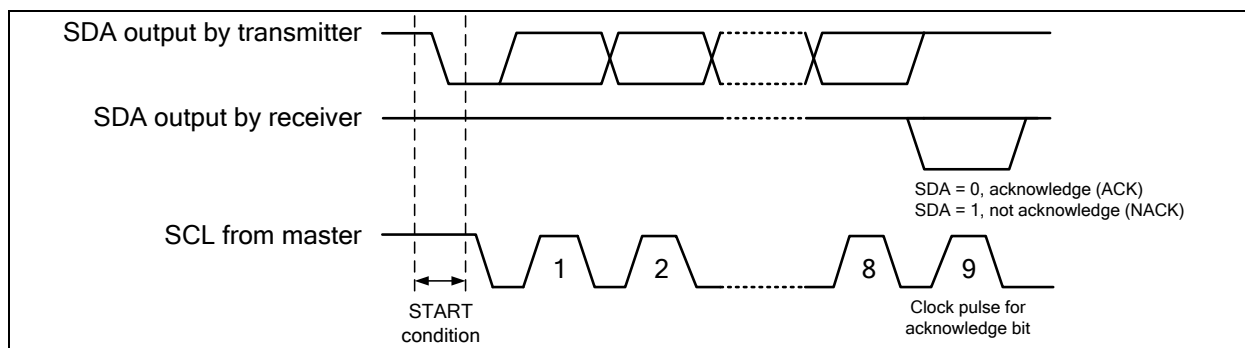


图 17.1- 7 应答位

17.1.4 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START试图发起数据传输，这时就会出现总线仲裁。在该状态下，当SCL为高时，SDA上呈现仲裁信号。在仲裁过程中，第一个发起主机对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。

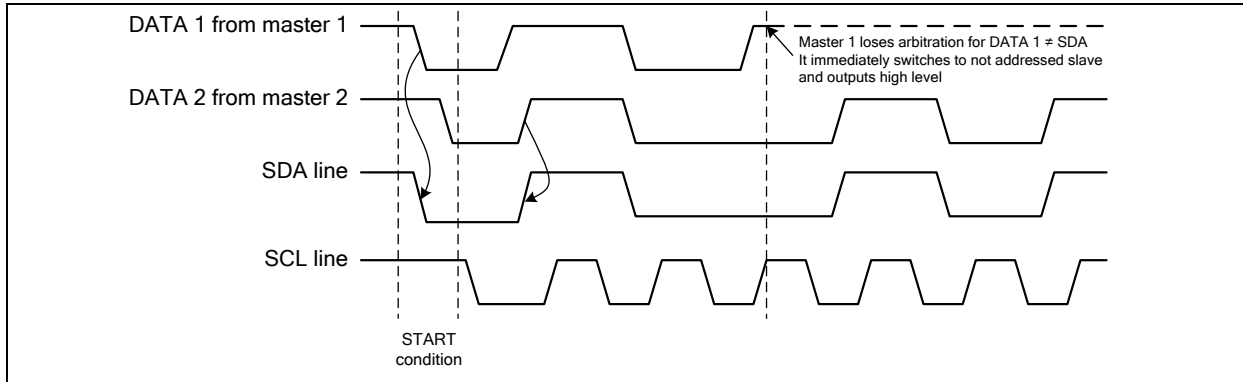


图 17.1- 8 两台主机仲裁过程

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。

17.1.5 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，如果从机愿意与主机接收或发送数据(通过设置AA位)，在第9个时钟将发送应答脉冲，因此如果使能了中断，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寄存器中的位STA、STO和AA用于在I2C_CTL0[3]寄存器的SI标志被清除后来控制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停止动作时将不设置SI标志。如果设置了I2C中断控制位INTEN (I2C_CTL0[7])，则在中断服务例程中将执行新状态码或软件。

图 17.1- 9 显示当前 I2C 状态码为 0x08，然后设置I2C_DATA=SLA+W 和 (STA,STO,SI,AA) = (0,0,1,x) 来发送到I²C总线。如果一个总线上的从机匹配地址并且应答， I2C_STATUS0 将更新状态码0x18。

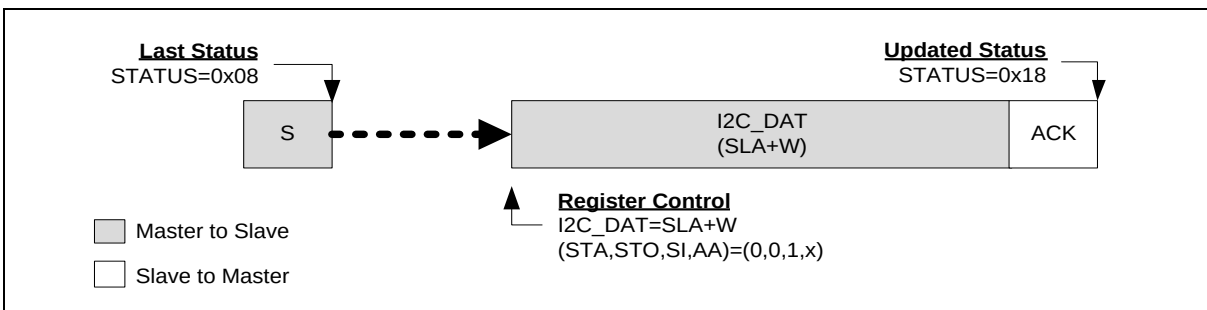


图 17.1- 9 根据当前 I²C 状态来控制 I²C 总线

17.1.5.1 主机发送模式

主机发送模式，主机发送多个字节到从机，主机产生时钟，故需要在I2CnCLK内填入设定值。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) 将被置1，状态码I2CnSTAT置为08H。接下来应把从机地址和写位(SLA+W)写入I2CnDAT，然后清0位SI，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，位 STO (I2CnCON.4)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。

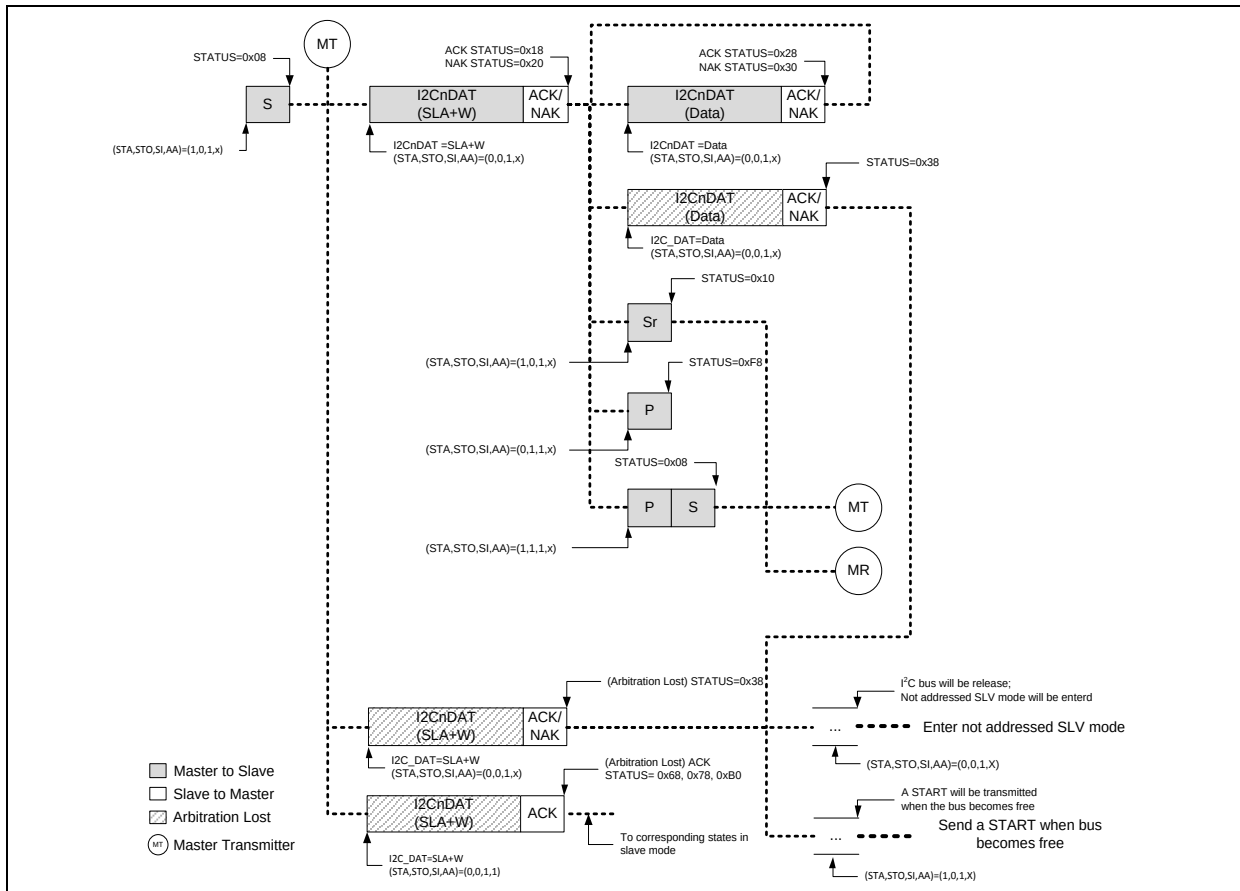


图 17.1- 10 主机发送模式流程和寄存器状态

17.1.5.2 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“读位” (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并释放从机为一个未寻址从机。然后主机可以发停止位或重起始开始新一轮传输。

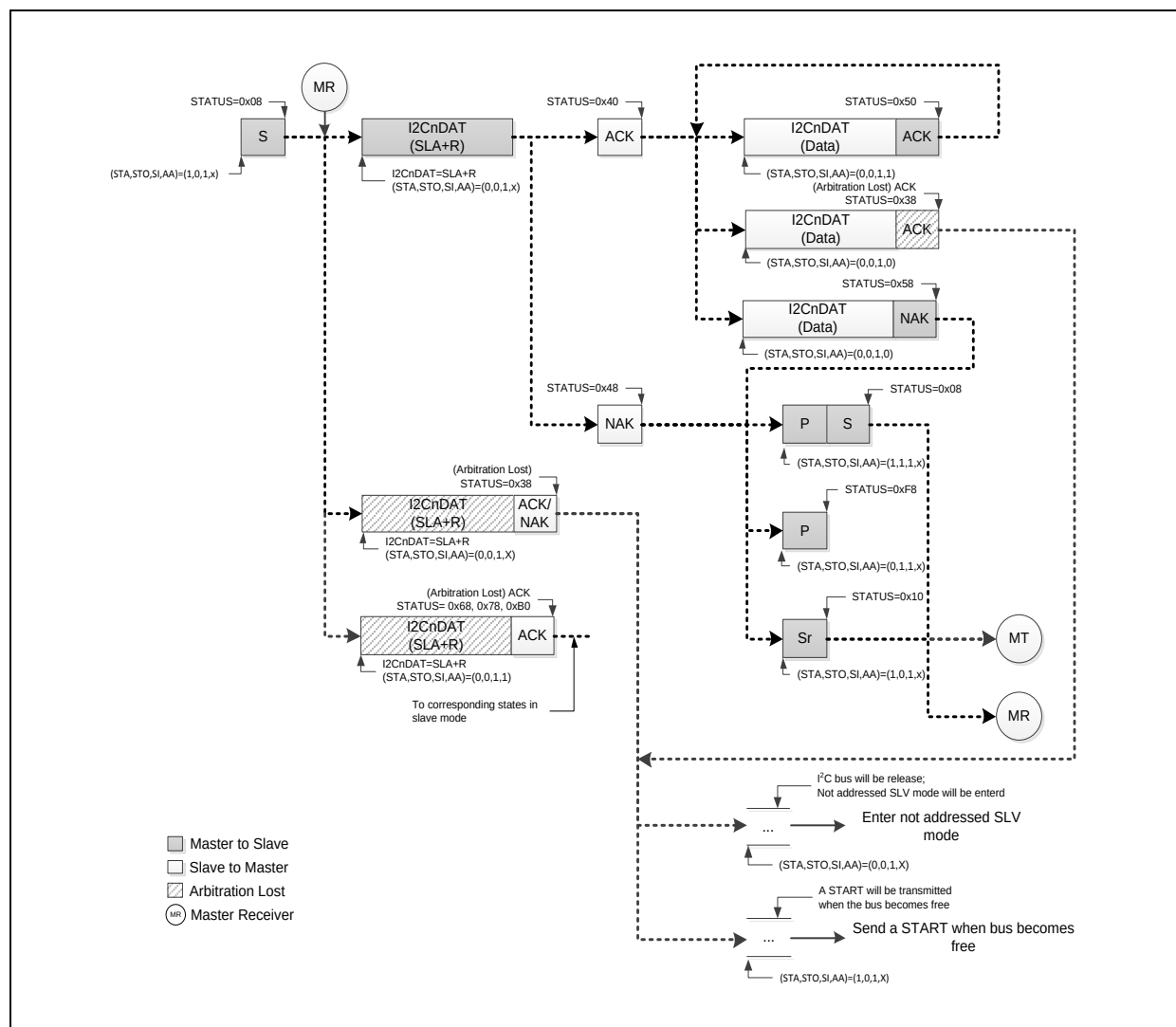


图 17.1-11 主机接收模式流程和寄存器状态

17.1.6 从机模式

17.1.6.1 从机接收模式

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用以响应主机的寻址。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）被寻址。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，从机也将转为未寻址从机，与主机联系终止，不再接收数据，且I2CnDAT保持之前接收到的数据。

17.1.6.2 从机发送模式

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，器件等待直到自身地址被“读”信号(SLA+只读)寻址。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寻址，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。并变为未寻址从机。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，并将自身变为未寻址从机。

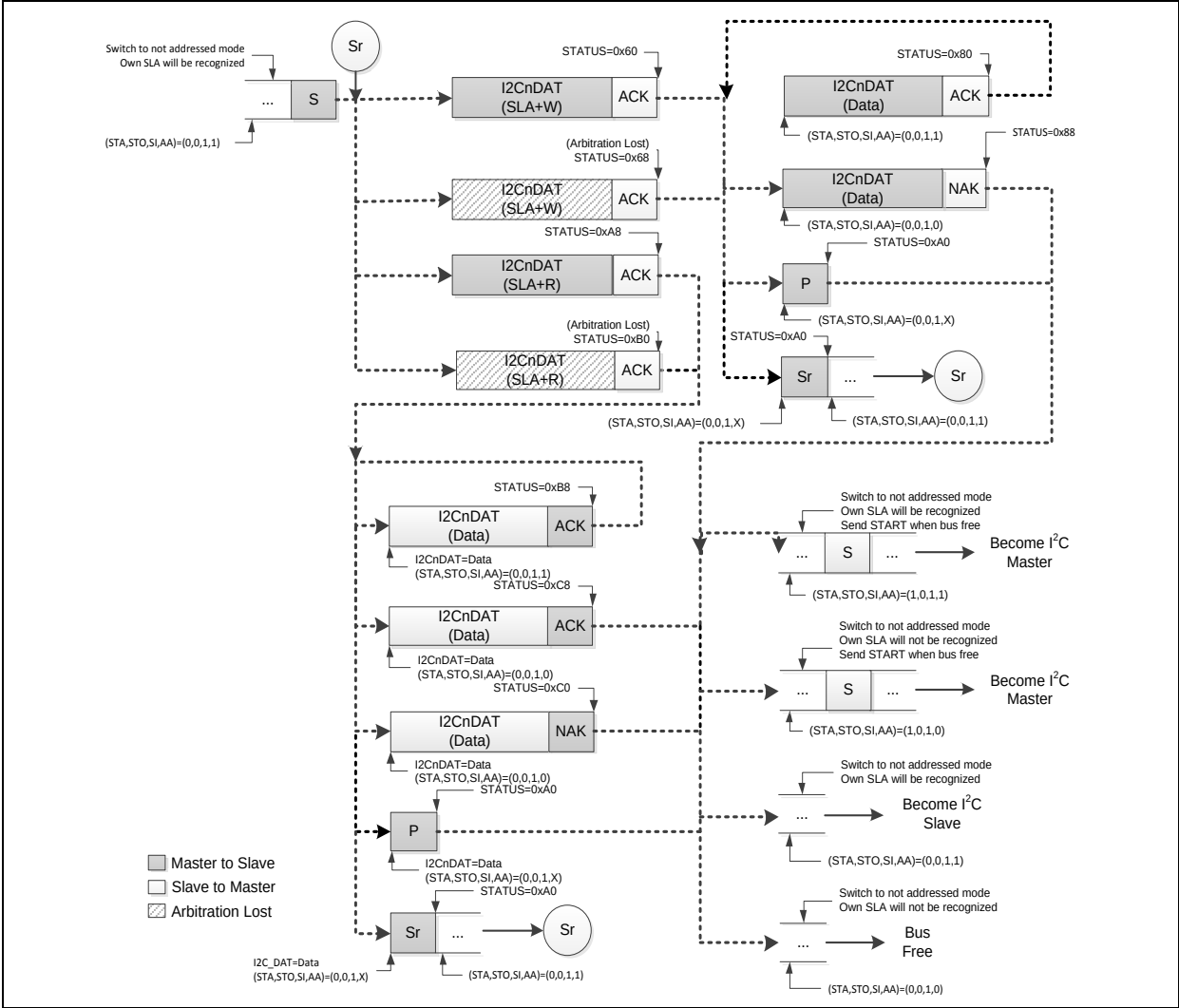


图 17.1- 12 从机接收模式流程和寄存器状态

17.1.7 广播呼叫模式

广播呼叫模式是一种特殊的从机接收模式，寻址为0x00，从机地址和读写都为0。当GC (I2CADDRx.0) 位及AA 都为置1，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。

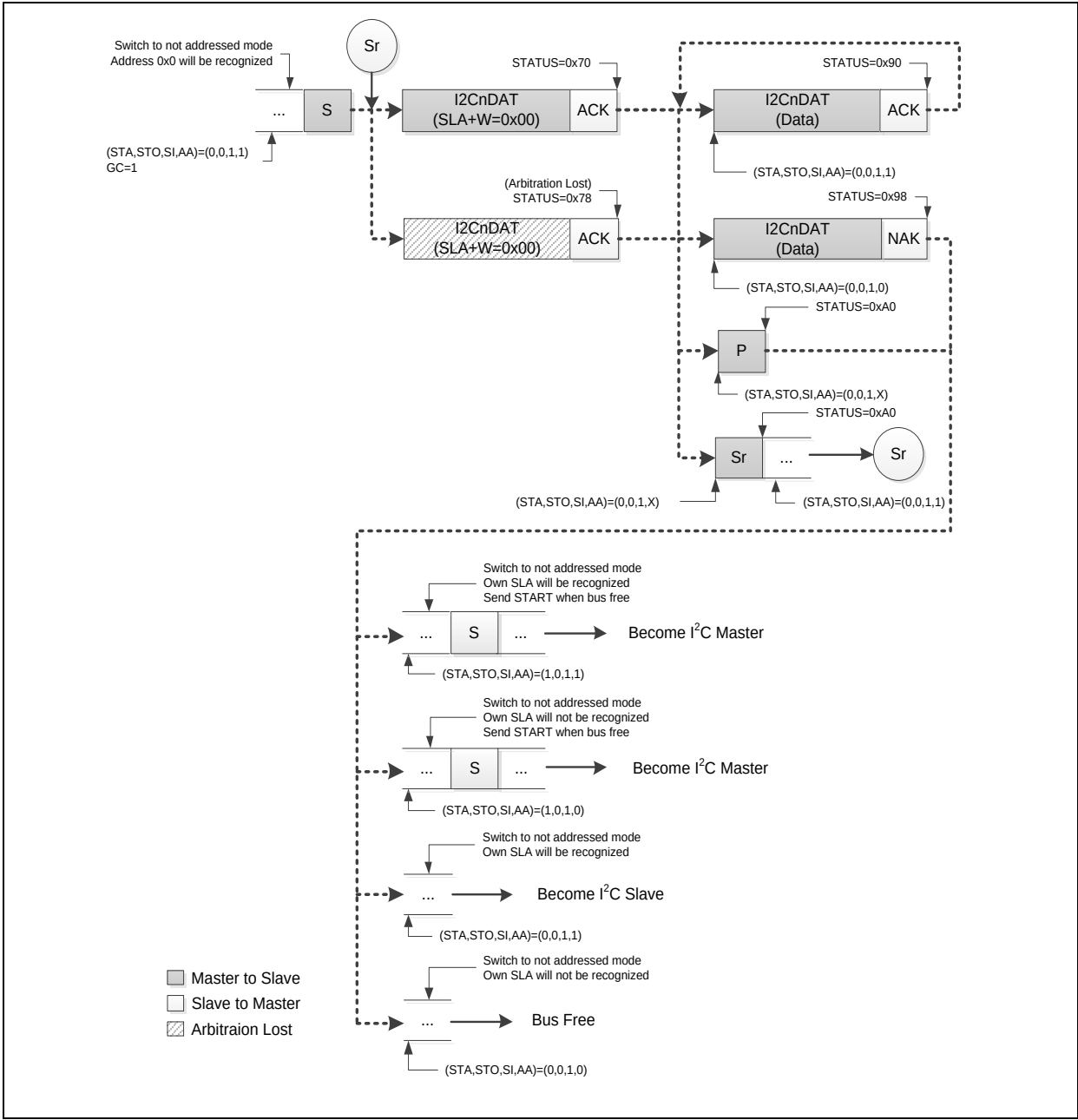


图 17.1- 13 广播呼叫模式流程和寄存器状态

17.1.8 I²C超时溢出

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。

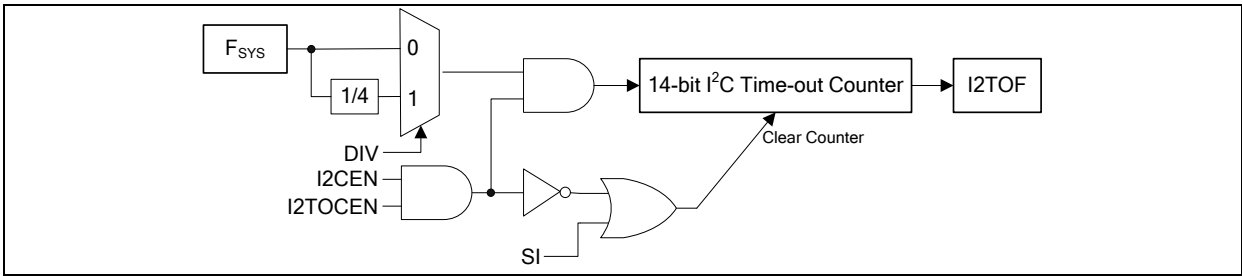


图 17.1- 14 I²C 超时溢出计数器

17.1.9 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。

17.1.10 各状态码表述

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标志马上会被置1，工作中的节点设备马上切换到未被寻址从机模式，释放SDA 和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， SDA线上不会产生START信号。一旦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		

17.2 I²C控制寄存器

I2CCON – I²C控制寄存器

寄存器	SFR 地址	复位 值
I2CCON	C0H, 所有页, 可位寻址	0000_0000 b

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

位	名称	描述
7	I	I²C0保持时间延伸使能 0 = I ² C 数据传输SCL保持时间延伸禁止。 1 = I ² C数据传输SCL保持时间延伸使能, 延伸8个系统时间。
6	I2CEN	I²C 总线使能 0 = I ² C 禁止。 1 = I ² C 使能。 使能 I ² C之前, SDA和SCL必须配置为输出1
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。
4	STO	停止标志STOP I ² C 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, I ² C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I ² C帧。
3	SI	I²C0 中断标志 I ² C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, 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, I ² C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。
2	AA	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SCL上的应答时钟脉冲期间, SDA将应答ACK (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SCL上的应答时钟脉冲期间, SDA将应答NACK (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, 且重新被主机寻址。 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, 从机将变成未被寻址的从机模式, 和主机断开。主机若再从总线上读数据, 将得到FFH。

位	名称	描述
1:0	-	保留

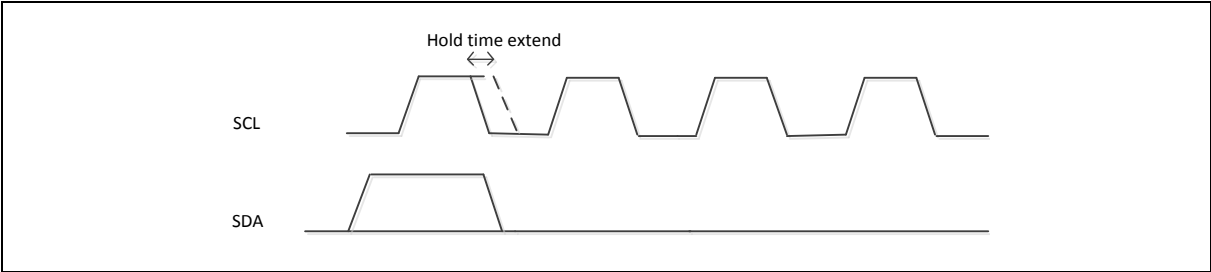


图 17.2- 1 SCL 保持时间延伸使能

I2CSTAT – I²C 状态寄存器

寄存器	SFR 地址	复位 值
I2CSTAT	BDH, 所有页	1111_1000 b

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

位	名称	描述
7:0	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。

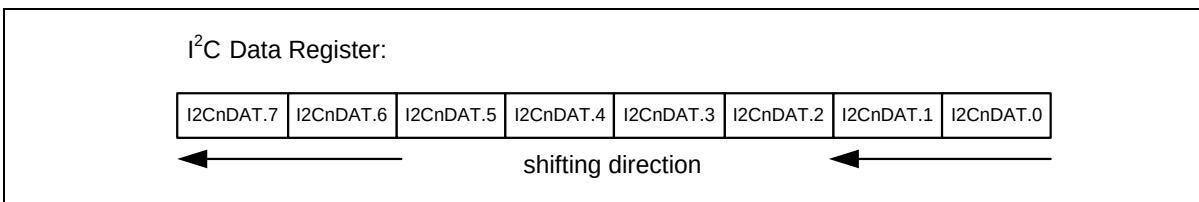
I2CDAT – I²C数据寄存器

寄存器	SFR 地址	复位 值
I2CDAT	BCH, 所有页	0000_0000 b

7	6	5	4	3	2	1	0
I2C0DAT[7:0]							
读/写							

位	名称	描述
7:0	I2CnDAT[7:0]	I ² C0 数据 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。 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。

I²C 数据移位方向



I2CADDR – I²Cn 从机地址

寄存器	SFR 地址	复位 值
I2CADDR	C1H, 所有页	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	I2C0ADDRx[7:1]	I²C0从机地址 主机模式: 无效 从机模式: 存放7位从机地址。主机需要寻址该从机, 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, 成为被寻址从机。否则主机呼叫地址会被忽略。 注: I2CnADDRx [7:1] 不能写为全0, 因为0x00为广播呼叫方式寻址专用。
6	GC	广播呼叫位 主机模式: 无效 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。

I2CLK – I²C时钟寄存器

寄存器	SFR 地址	复位 值
I2CLK	BEH, 所有页	0000_1001 b

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

Bit	Name	Description
-----	------	-------------

Bit	Name	Description
7:0	I2CLK[7:0]	I²C 时钟设定 <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: $\frac{F_{sys}}{4 \times (I2CLK + 1)}$ 注I2CCLK值写入00H 及 01H 无效, 实行了限制。 <u>从机模式:</u> 该字节无效, 从机自动同步主机时钟, 最高400kbps.

17.1 I²C 中断服务程序典型结构范例

下例为在Keil™ C51 中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*/
                                //recover from bus error
                                STO = 1;
                                break;
        //=====
        //Master Mode
        //=====
        case 0x08:                /*08H, a START transmitted*/
                                //STA bit should be cleared by software
                                I2DAT = SLA_ADDR1;        //load SLA+W/只读
                                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*/
                                STO = 1;                    //transmit STOP
                                AA = 1;                      //ready for ACK own SLA+W/只读 or General
                                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*/
                                //retry to transmit START if bus free
                                STA = 1;
                                break;
        //=====
        //Master Receiver Mode
        //=====
        case 0x40:                /*40H, SLA+只读 transmitted, ACK received*/
```

```

        AA = 1;                                //ACK next received DATA
        break;
    case 0x48:                                  /*48H, SLA+只读 transmitted, NACK received*/
        ST0 = 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;
        ST0 = 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/只读
                                                own SLA+W received, ACK returned */
        AA = 0;                                //not ACK next received DATA after
                                                //arbitration lost
        STA = 1;                                //retry to transmit START if bus free
        break;
    case 0x70:                                  /*70H, General Call received, ACK
returned*/
        AA = 1;
        break;
    case 0x78:                                  /*78H, arbitration lost in SLA+W/只读
                                                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+只读 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/只读
                                own SLA+只读 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+只读, 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+只读, DATA transmitted,
                                NACK received, not addressed SLAVE mode
                                entered*/
        AA = 1;
        break;
case 0xC8:                      /*C8H, previous own SLA+只读, last DATA trans-
                                mitted, ACK received, not addressed SLAVE
                                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
}

```

18 管脚中断

18.1 管脚中断概述

MS51每个管脚都提供管脚输入中断功能，用于检测管脚电平状态，如按键或键盘是否按下。最多配置8个I/O管脚用于管脚中断。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。管脚中断能使MCU从空闲模式或掉电模式唤醒。

每个管脚的中断使能和正反向特性通过PIPEN寄存器、PINEN寄存器分别设置。通过PICON寄存器选择端口用于管脚中断，设置管脚中断的触发方式（电平或边沿检测）。每个中断通道都有自己的中断标志，总共8个中断标志，存放在寄存器 PIF 中，进入中断后判断该寄存器确定中断发生具体管脚。PIF寄存器中的所有标志位都是通过硬件置位，通过软件清0。

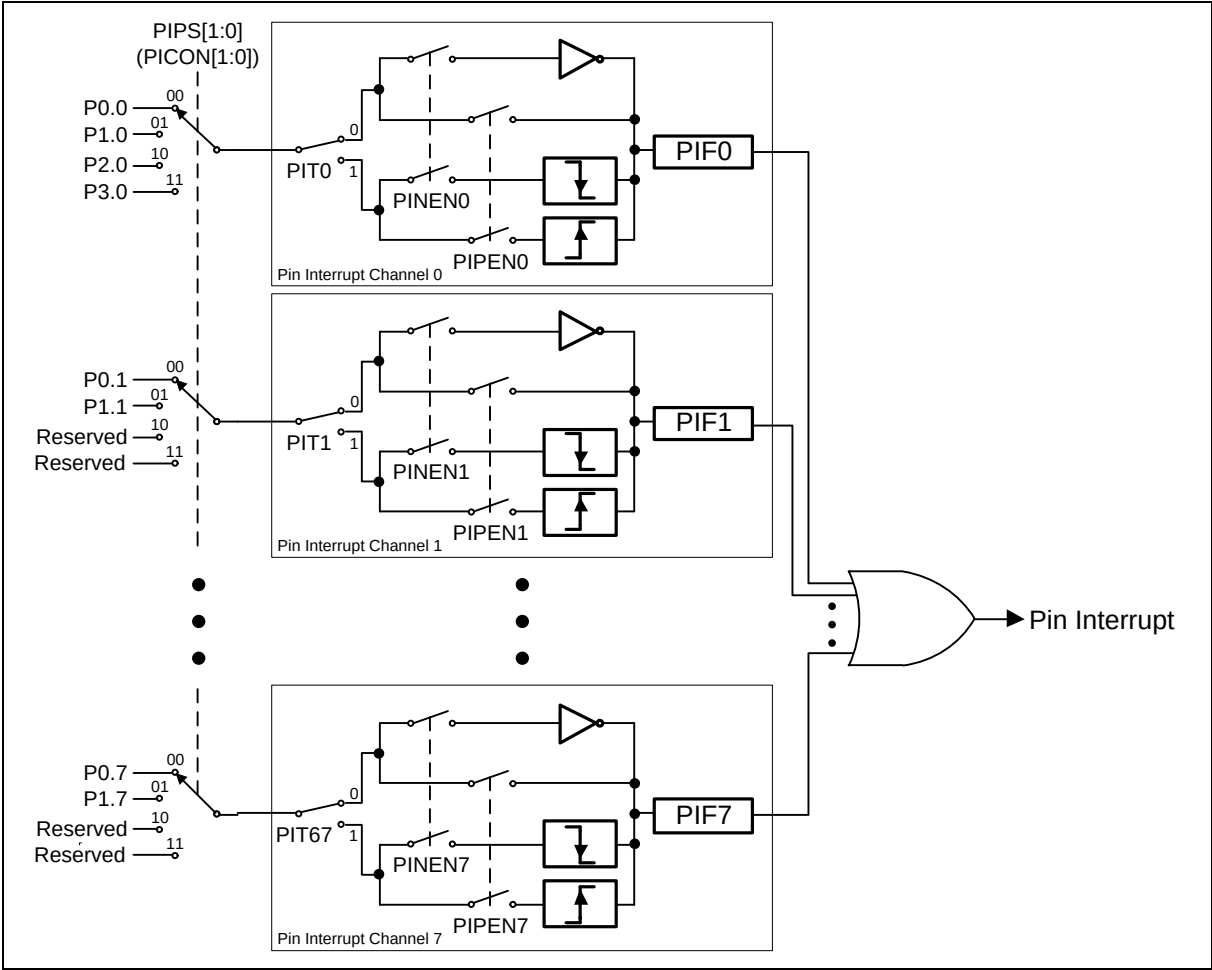


图 18.1- 1 管脚中断模块框图

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。

18.2 管脚中断控制寄存器

PICON – 引脚中断寄存器

寄存器	SFR 地址	复位值
PICON	E9H, 所有页	0000_0000 b

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

位	名称	描述
7	PIT67	管脚中断通道6 及 7 类型选择 该位用来配置管脚中断6和7产生中断的信号类型 0 = 电平触发 1 = 边沿触发
6	PIT45	管脚中断通道4 及 5 类型选择 该位用来配置管脚中断4和5产生中断的信号类型 0 = 电平触发 1 = 边沿触发
5	PIT3	管脚中断通道3 类型选择 该位用来配置管脚中断3产生中断的信号类型 0 = 电平触发 1 = 边沿触发
4	PIT2	管脚中断通道2 类型选择 该位用来配置管脚中断2产生中断的信号类型 0 = 电平触发 1 = 边沿触发
3	PIT1	管脚中断通道1 类型选择 该位用来配置管脚中断1产生中断的信号类型 0 = 电平触发 1 = 边沿触发
2	PIT0	管脚中断通道0 类型选择 该位用来配置管脚中断0产生中断的信号类型 0 = 电平触发 1 = 边沿触发
1:0	PIPS[0]	管脚中断端口选择 该位段选择管脚中断所用8位端口 00 = 端口 0 01 = 端口 1 10 = 端口 2 11 = 端口 3

PINEN – 管脚中断反向特性使能

寄存器	SFR 地址	复位值
PINEN	EAH, 所有页	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
7:0	PIENn	管脚中断通道n负极性使能 该位使能低电平/下降沿触发管脚中断通道n。电平触发还是边沿触发取决于PICON的每个控制位PITn。 0 = 低电平/下降沿检测禁用。 1 = 低电平/下降沿检测使能。

PIPEN – 管脚触发正相特性使能

寄存器	SFR 地址	复位值
PIPEN	EBH, 所有页	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	管脚中断通道n正极性使能 该位使能高电平/上升沿触发管脚中断通道n。电平还是边沿触发取决于PICON的每个控制位PITn。 0 = 高电平/上升沿检测禁用。 1 = 高电平/上升沿检测使能。

PIF – 引脚中断标志

寄存器	地址, 页, 特别说明	复位值
PIF	ECH, 所有页	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	管脚中断通道n标志 如果选择边沿触发, 如果管脚中断的通道n检测到使能的边沿触发, 该标志位将被硬件置1, 该标志由软件清零。 如果选择电平触发, 该标志位根据管脚中断的通道n的输入信号的逻辑电平的反向决定, 软件不可以控制它。

19 脉冲宽度调制 (PWM)

PWM (脉冲宽度调制) 信号在控制方案中应用非常广泛。可用于电机驱动、风扇控制、背光调节、LED 光源调光或通过低通滤波器电路模拟一个简单的数模转换器的输出。

MS51包含三对（6个PWM通道）16位精度、可调周期和占空比的PWM输出，非常适合用于电机控制。该模块架构适用于驱动单相或三相无刷直流电机(BLDC)，或三相交流感应电机。每个信道PWM输出可配置为独立输出模式、互补模式或同步模式。当设定为互补模式时，通过插入可配置的死区时间，保护MOS管同时导通。PWM波形可配置边沿对齐或中心对齐来选择中断响应位置。

19.1 功能描述

19.1.1 脉冲宽度调制发生器

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

为了更好适用于三相电机控制，通过设置GP (PWMCON1.5)位来使用组群模式，{PWM0H, PWM0L}和{PWM1H, PWM1L}占空比寄存器决定PWM输出的占空比。在三相电机控制应用中，另二对PWM输出产生完全相同的占空比信号，一旦组群模式启用{PWM2H, PWM2L}、{PWM3H, PWM3L}、{PWM4H, PWM4L} and {PWM5H, PWM5L} 寄存器失效。意味着寄存器{PWM2H, PWM2L}和{PWM4H, PWM4L}与{PWM0H, PWM0L}相同，{PWM3H, PWM3L}和{PWM5H, PWM5L}与{PWM1H, PWM1L}相同。

注：启用PWM输出，芯片不会自动配置管脚为“输出模式”，用户需要通过软件配置。

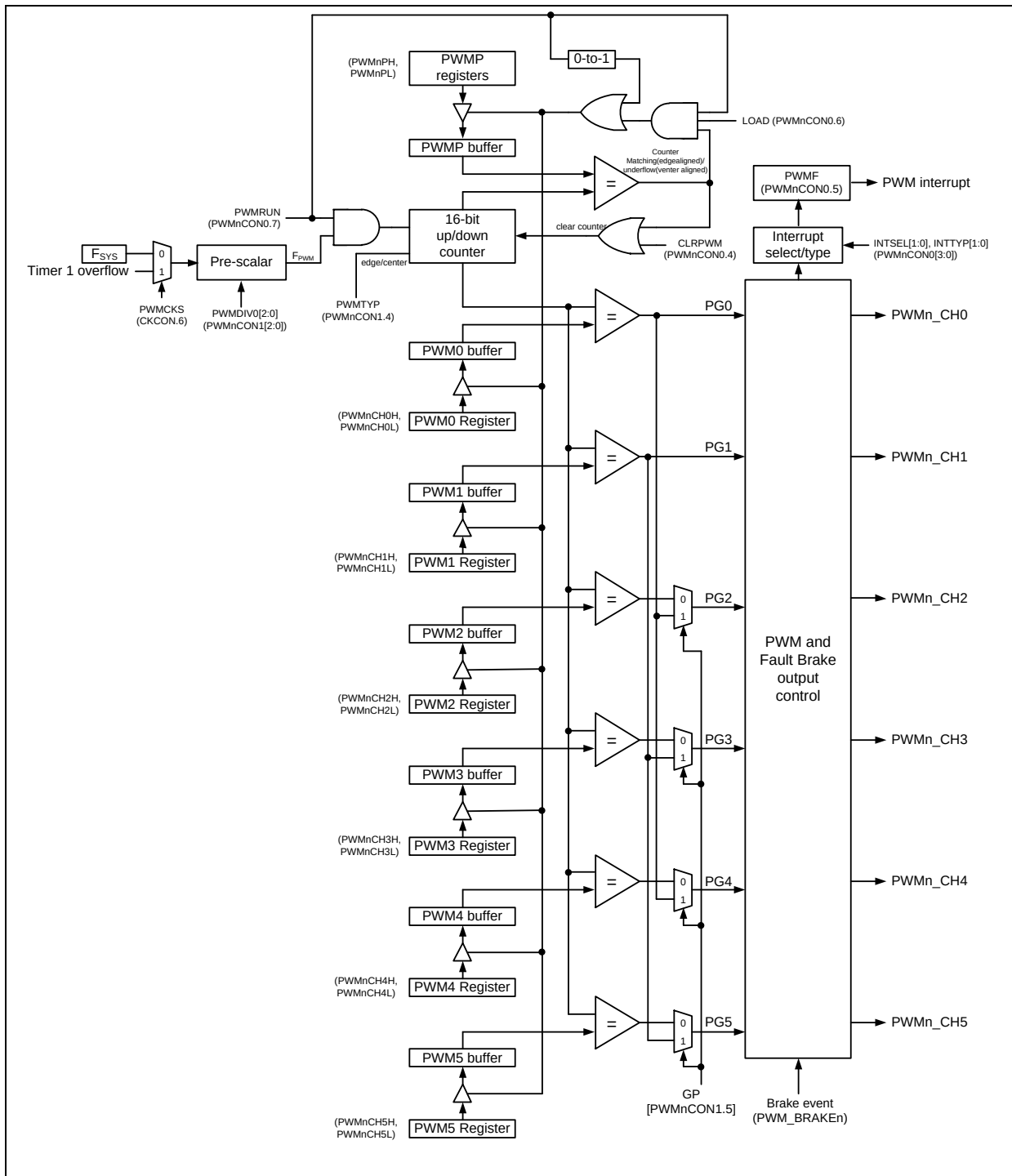


图 19.1- 1 PWM 结构框图

PWM计数器产生六个PWM信号，分别是 PG0, PG1, PG2, PG3, PG4, 和 PG5。这些信号通过PWM及故障刹车输出控制电路，产生实际PWM输出至I/O管脚。输出控制电路决定PWM模式，死区插入，屏蔽输出，故障刹车控制，以及PWM极性。最后是PWM输出或I/O功能的多路转换器。

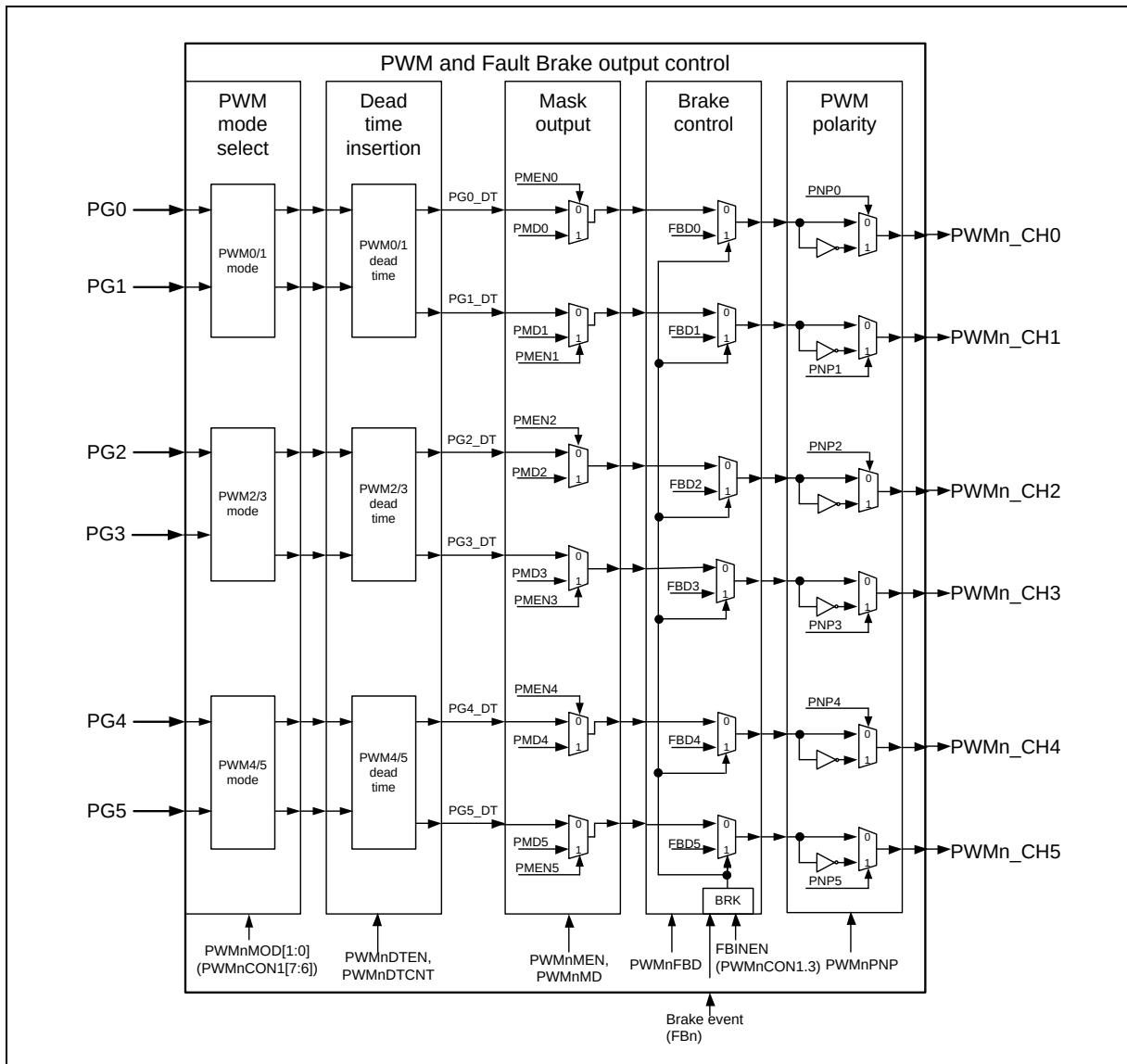


图 19.1- 2 PWM 和故障刹车输出控制方框图

请按照如下初始化步骤来产生PWM信号。第一步，设定CLRPWM (PWMnCON0.4)，以确保16位向上计数器计数准确。初始化和设置{PWMnPH, PWMnPL}和所有{PWMnH, PWMnL}寄存器之后，对PWMRUN (PWMnCON0.7)置1触发16位计数器运行。PWM开始在输出管脚产生波形。所有的周期及占空比寄存器，具有硬件双缓存设计。因此，{PWMnPH, PWMnPL} 和 所有 {PWMnH, PWMnL} 寄存器可随时被写，但不会立即更新PWM周期和占空比，直到LOAD (PWMnCON0.6)置1且前一个周期完成。避免当更新PWM周期或占空比时，产生故障。

注：通过设置LOAD加载新的周期和占空比，必须通过监控LOAD位，等待硬件自动清除LOAD位，确保完成。当LOAD位为1时，任何对PWM控制寄存器的更新，将可能引起不可预知的输出。

19.1.2 PWM类型

PWM发生器包含两种类型，边沿对齐或中心对齐。PWM类型通过设定PWMTYP (PWMnCON1.4)选择。

19.1.2.1 边沿对齐类型

边沿对齐模式，16位计数器为单溢出模式，从0000H计数到{PWMnPH, PWMnPL}，然后重新从0000H开始开始计数。PWM发生器信号（PGn信号在PWM和故障刹车输出控制模块之前）在16位计数器与周期寄存器和占空比寄存器{PWMnH, PWMnL}设置值匹配时清除，并且设置16位计数器为0000H。PWM输出波形为左边沿对齐方式。

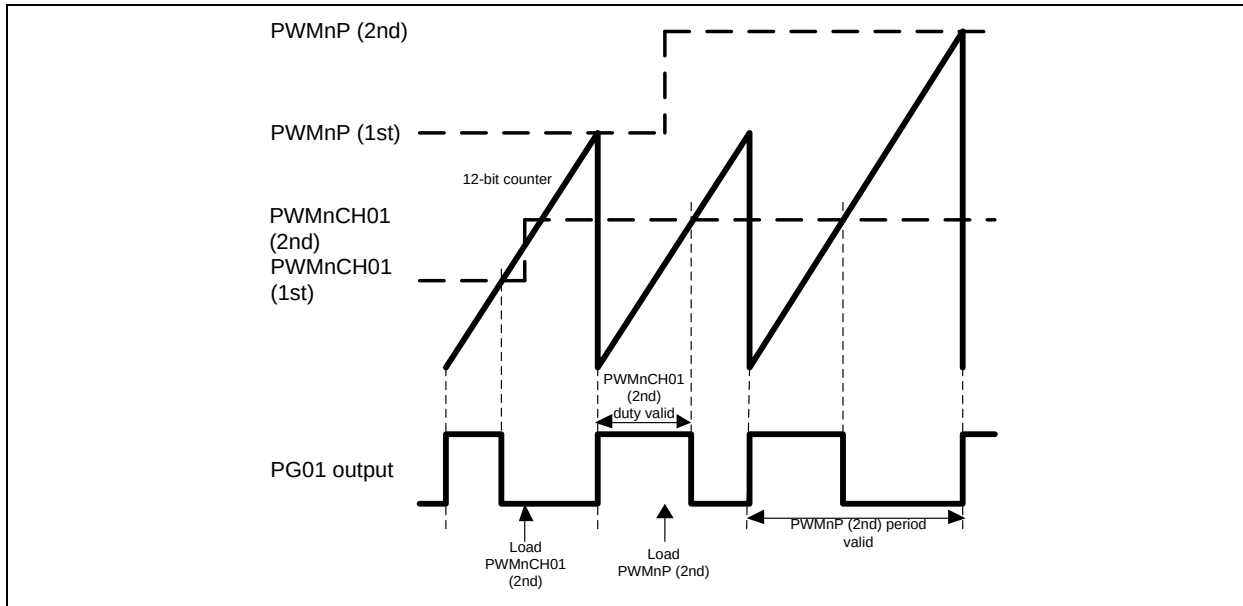


图 19.1- 3 PWM 边沿对齐类型波形

PWM边沿对齐方式输出频率及占空比算式如下：

$$\text{PWM 频率} = \frac{F_{\text{PWM}}}{\{\text{PWMnPH}, \text{PWMnPL}\} + 1} \quad (F_{\text{PWM}} \text{为PWM时钟源除以PWMDIV}).$$

$$\text{PWM 占空比高电平} = \frac{\{\text{PWMnCHxH}, \text{PWMnCHxL}\}}{\{\text{PWMnPH}, \text{PWMnPL}\} + 1}.$$

19.1.2.2 中心对齐类型

中心对齐模式，16位计数器采用双溢出模式，从0000H向上计数到{PWMnPH, PWMnPL}，然后由{PWMnPH, PWMnPL}向下计数至0000H。PGn信号在16位计数器与占空比寄存器{PWMnH, PWMnL}在向上计数匹配时清零，在向下计数匹配时置位。

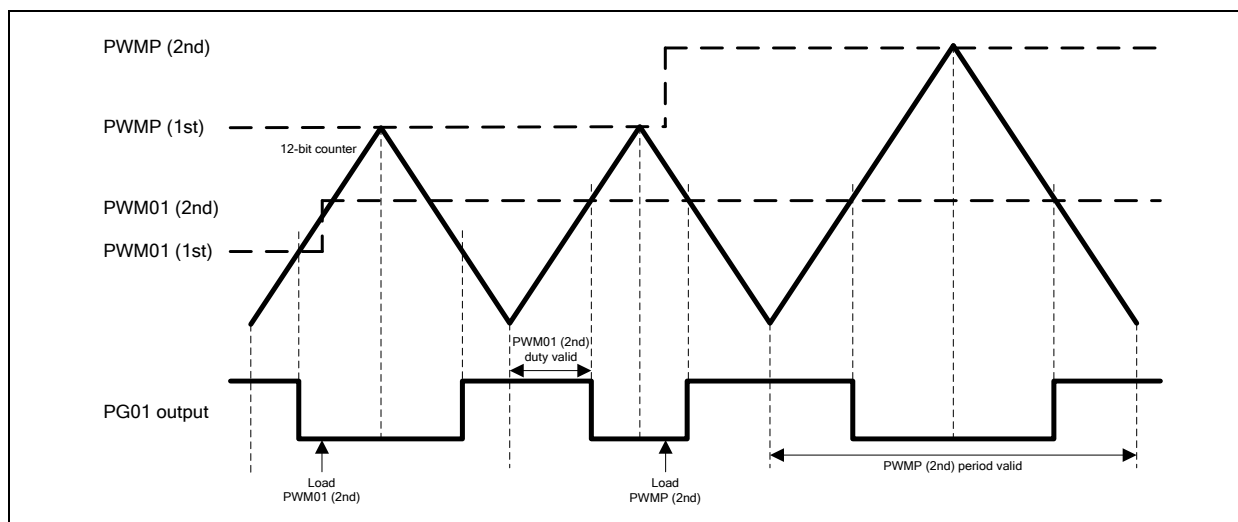


图 19.1- 4 PWM 中心对齐类型波形

中心对齐模式输出频率及占空比算式如下：

$$\text{PWM 频率} = \frac{F_{\text{PWM}}}{2 \times \{ \text{PWMnPH}, \text{PWMnPL} \}} \quad (F_{\text{PWM}} \text{ 为 PWM 时钟源除以 PWMDIV}).$$

$$\text{PWM 占空比高电平} = \frac{\{ \text{PWMnCHxH}, \text{PWMnCHxL} \}}{\{ \text{PWMnPH}, \text{PWMnPL} \}}.$$

19.1.3 工作模式

PGn信号通过PWM和故障刹车输出控制电路后，PWM模式选择电路将产生不同类型的PWM输出模式，总共6个通道、三组，分别是PG0~PG5。支持独立输出模式，互补模式及同步模式。

19.1.3.1 独立模式

当PWMMOD[1:0] (PWMnCON1[7:6])设定为 [0:0]，独立模式使能。该模式为PWM默认模式，PG0, PG1, PG2, PG3, PG4 及 PG5 独立输出PWM信号。

19.1.3.2 带死区插入的互补模式

当PWMMOD[1:0] = [0:1]，互补模式使能。在该模式中PG0/2/4输出信号与独立模式下输出信号相同，但PG1/3/5 输出与PG0/2/4输出的信号互补。同时忽略PG1/3/5占空比寄存器{PWMnH, PWMnL} (n:1/3/5)。该模式可使PG0/PG1形成一对互补的PWM输出。同样 PG2/PG3, PG4/PG5也一样。

在实际马达应用中，互补PWM输出经常需要插入“死区”，用来防止半桥在上下开关同时打开时损坏GPIBs之类的电源开关器件，尽管只是“us”级的时间。用来控制不能同时开关的电源开关器件。MS51每组PWM共享一个9位死区时间向下计数器PWM0DTCNT，用于在同组中两个PWM信号之间产生关闭时间。在实现上，是在0要跳到1的信号之后加入一段延时等待PWM0DTCNT溢出，再跳转。下图展示了带死区插入的PG0/PG1组互补模式的时序图。组PG2/PG3 和 PG4/PG5有同样的死区时间电路。每组有各自的死区时间使能位（PWMnDTEN [3:0]）。

注： PWM0DTCNT 和 PWMnDTEN寄存器都为TA写保护。死区时间控制只有在PWM配置为互补模式时才有效。

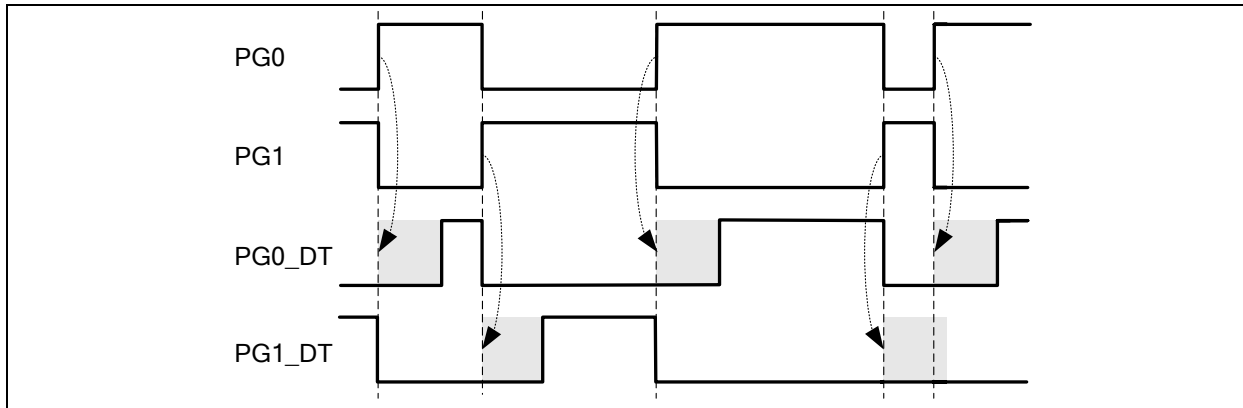


图 19.1- 5 带死区时间插入的 PWM 互补模式

19.1.3.3 同步模式

当PWMMOD[1:0] = [1:0]时，同步模式使能。在该模式， PG0/2/4输出跟独立模式一样的PWM信号。PG1/3/5输出跟PG0/2/4一样的PWM信号。

19.1.4 屏蔽输出控制

每一个PWM信号可以通过软件屏蔽输出一个指定的电平。PWM屏蔽输出功能在控制电子换相电机如BLDC非常有用。PWMnMEN包含6个位，用于决定屏蔽哪个PWM通道。PWMnMD设置PWM通道的屏蔽电平。PWMnMEN 默认值为00H，PWM输出不屏蔽。注：PWM屏蔽电平是否反向取决于PWMONP的设置。

19.1.5 故障刹车

故障刹车功能通常用在增强型PWM电路中。它带故障检测输入用于保护马达系统，防止损坏。当FBINEN (PWMnCON1.3)置1后，故障刹车引脚输入 (FB) 有效。当故障刹车产生，PWM信号根据PWMnFBD相应位重写，PWMRUN (PWMnCON0.7)将被硬件自动清除，停止PWM产生。PWM 16位计数器将被复位为0000H。如果中断使能，相应的FBF标志将被硬件置1并产生硬件刹车中断。PWMnFBD输出保留，尽管FBF由软件清除。用户必须通过重新设置PWMRUN恢复PWM输出。同时，故障刹车状态将释放，PWM波形像往常一样输出到相应的引脚。故障刹车输入有一个极性选择，可通过FBINLS (PWMnFBD.6)位选择。注：由于FB引脚输入有一个 $8/F_{SYS}$ de-bouncing，用于避免输入噪声产生故障刹车事件，所以反馈到FB引脚的故障刹车信号必须8个系统时钟时间。另一个触发故障刹车事件的路径是ADC比较事件。其产生故障刹车行为与FB引脚输入一样。

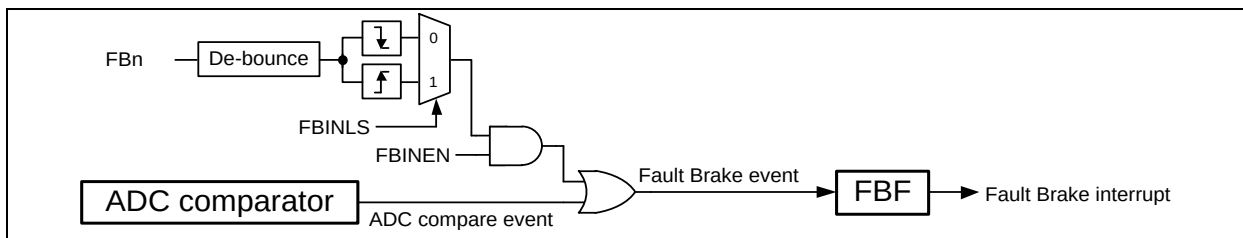


图 19.1- 6 故障刹车功能方框图

19.1.6 极性控制

每一个PWM输出通道有独立的极性控制位PNP0~PNP5。默认正逻辑为高电平有效，即当PWM输出高电平，电源开关开，输出低电平，电源开关关。用户可以设置PWMnNP位，使PWM实际输出反向。

19.1.7 PWM 中断

PWM模块有一个标志位PWMF (PWMnCON0.5)，用来表明每一个PWM周期指定的点。指定的PWM通道和点可以通过NTSEL[2:0] 和 INTTYP[1:0] (PWMnINTC[2:0] 和 [5:4])选择。注，中心点中断或结束点中断只有在PWM运行在中心对齐类型时有效。PWMF由软件清0。

PWM波形相关PWM中断如下图所示

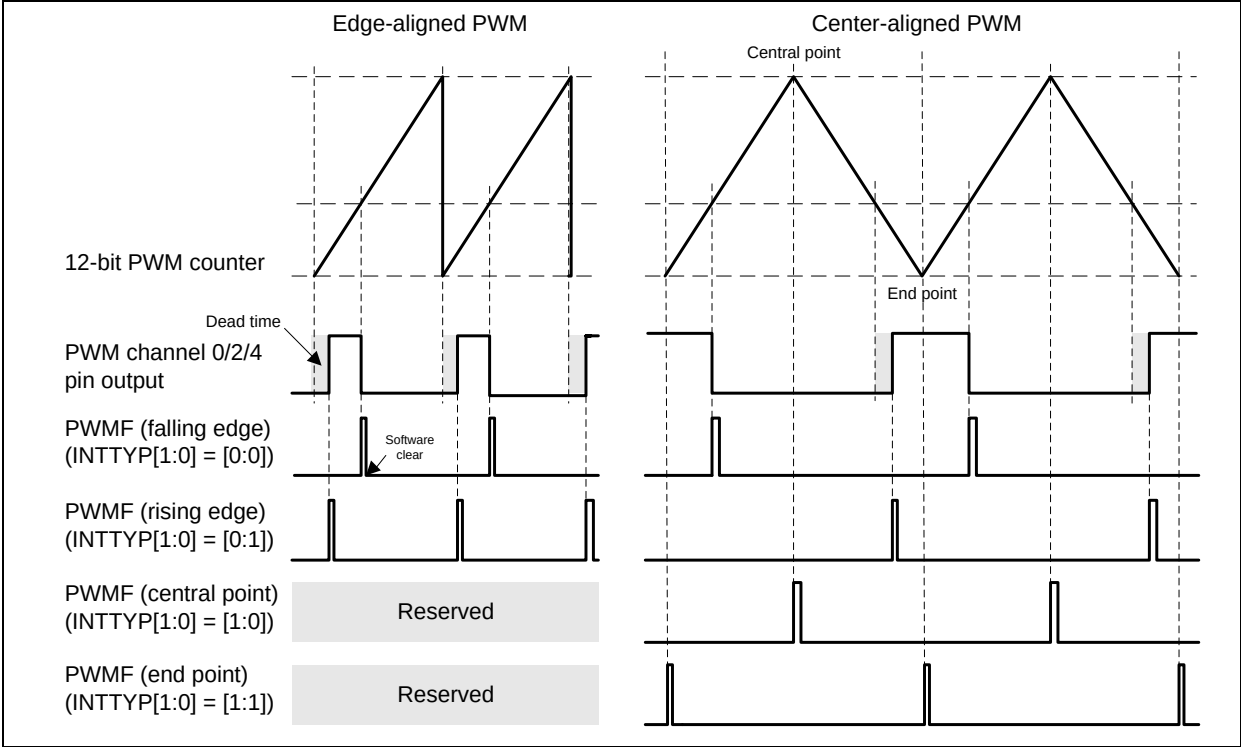


图 19.1- 7 PWM 中断类型

故障刹车事件需要另一个中断，故障刹车中断。其跟PWM中断有不一样的中断向量。当故障刹车引脚输入事件或ADC比较事件发生，FBF (PWMnFBD.7)将被硬件置位。如果中断使能，将产生故障刹车中断。故障刹车中断使能位为EFB0 (EIE0.5). FBF由软件清零。

19.2 PWM 控制寄存器

PWMCON0 – PWM 控制寄存器 0

寄存器	SFR 地址	复位值
PWM0CON0	D8H, 所有页	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 运行使能 0 = PWM处于空闲模式 1 = PWM 开始运行
6	LOAD	PWM新周期和占空比载入 该位用于如果新的周期或占空比值需要被更新时，载入周期和占空比控制寄存器到它们的缓存中。当前一个PWM周期输出结束，载入动作才会开始。新的周期及占空比值将在下一个周期生效。当载入动作结束，硬件自动清除LOAD位，也就是说LOAD位写入及读出是不一样的。 <u>写:</u> 0 = 不影响。 1 = 当PWM周期完成，载入新的周期和占空比到它们的缓存。 <u>读:</u> 0 = 载入新周期和占空比完成。 1 = 载入新周期和占空比还没完成。
5	PWMF	PWM 标志 该位根据PWMnINTC寄存器的INTSEL[2:0]和INTTYP[1:0]的定义来设置。该位通过软件清零。
4	CLRPWM	清除PWM计数器 设置该位将清除PWM 16位计数器为0000H，当计数器的值清除完成后，CLRPWM将由硬件自动清除。这就意味着，写和读CLRPWM位是不一样的。 <u>写:</u> 0 = 不影响。 1 = 清PWM 16位计数器。 <u>读:</u> 0 = PWM 16位计数器完成清除。 1 = PWM 16位计数器未完成清除。

PWMCON1 – PWM 控制器 1

寄存器	SFR 地址	复位值
PWM0CON1	DFH, 所有页	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
---	----	----

位	名称	描述
7:6	PWMMOD [1:0]	PWM 模式选择 00 = 独立输出模式 01 = 互补模式 10 = 同步模式 11 = 保留
5	GP	群组模式使能位 该位使能PWM群组模式。一旦使能，三对PWM的占空比由PWM01H 和 PWM01L决定，原本配置的数据失效。 0 = 群组模式禁用 1 = 群组模式使能
4	PWMTYP	PWM 类型选择 0 = PWM边沿对齐模式。 1 = PWM中心对齐模式。
3	FBINEN	FB 引脚输入使能 0 = 通过FB引脚输入触发PWM输出刹车故障禁止。 1 = 通过FB引脚输入触发PWM输出刹车故障使能。一旦在FB管脚输入的边沿信号与FBINLS (FBD.6)选择位相同。PWM0~5输出FBD寄存器所设置的信号电平， PWMRUN (PWMCON0.7) 位硬件清除。当PWMRUN信号重置1，PWM信号重新输出。
2:0	PWMDIV[2:0]	PWM 时钟除频 此域决定PWM时钟源预分频。 000 = 1/1. 001 = 1/2 010 = 1/4. 011 = 1/8. 100 = 1/16. 101 = 1/32. 110 = 1/64. 111 = 1/128.

CKCON – 时钟控制

寄存器	SFR 地址	复位值
CKCON	8EH, 页0	1000_0000 b

7	6	5	4	3	2	1	0
FASTWK	PWMCKS	T1OE	T1M	T0M	T0OE	CLOEN	-
读/写	读/写	读/写	读/写	读/写	读/写	读/写	-

位	名称	描述
6	PWMCKS	PWM时钟源选择 0 = PWM时钟源为系统时钟FSYS. 1 = PWM时钟源为定时器1的溢出。

PWMnPL – PWM 周期低字节

寄存器	SFR 地址	复位值
PWM0PL	D9H, 所有页	0000_0000 b

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

PWMP[7:0]		
读/写		

位	名称	描述
7:0	PWMnP[7:0]	PWMn周期低字节 该字节与PWMnPH控制PWM产生信号的周期.

PWMnPH – PWM 周期高字节

寄存器	SFR 地址	复位值
PWM0PH	D1H, 所有页	0000_0000 b

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

位	名称	描述
7:0	PWMP[15:8]	PWM周期高字节 该字节与PWMnPL控制PWM产生信号的周期.

PWMnH – PWM 通道 0~5占空比寄存器高字节 n=0,1,2,3,4,5

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

7	6	5	4	3	2	1	0
PWM0[15:8]							
读/写							

位	名称	描述
7:0	PWM0[15:8]	PWM占空比高字节 这字节与PWMnL搭配确定PWM输出的占空比

PWMnL –PWM 通道 0~5占空比寄存器低字节 n=0,1; x=0,1,2,3,4,5

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

7	6	5	4	3	2	1	0
PWM0[7:0]							
读/写							

位	名称	描述
7:0	PWMn[7:0]	PWM 占空比低字节 该位与PWMnH搭配, 配置由PWM输出的占空比数据

PIOCON0 –PWM或I/O 选择寄存器

寄存器	SFR 地址	复位值
PIOCON0	DEH, 所有页	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/PWM5管脚功能选择位 0 = P0.3/PWM5 管脚用作 P0.3 1 = P0.3/PWM5 管脚用作 PWM5输出
4	PIO04	P0.1/PWM4管脚功能选择位 0 = P0.1/PWM4 管脚用作 P0.1 1 = P0.1/PWM4 管脚用作 PWM4输出
3	PIO03	P0.0/PWM3管脚功能选择位 0 = P0.0/PWM3 管脚用作 P0.0 1 = P0.0/PWM3 管脚用作 PWM3输出
2	PIO02	P1.0/PWM2管脚功能选择位 0 = P1.0/PWM2 管脚用作 P1.0 1 = P1.0/PWM2 管脚用作 PWM2输出

位	名称	描述
1	PIO01	P1.1/PWM1管脚功能选择位 0 = P1.1/PWM1 管脚用作P1.1 1 = P1.1/PWM1 管脚用作 PWM1输出
0	PIO00	P1.2/PWM0管脚功能选择位 0 = P1.2/PWM0 管脚用作 P1.2 1 = P1.2/PWM0 管脚用作 PWM0输出

PIOCON1 –PWM或I/O 选择寄存器1

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

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

位	名称	描述
5	PIO15	P1.5/PWM5管脚功能选择位 0 = P1.5/PWM5 管脚用作 P1.5 1 = P1.5/PWM5 管脚用作 PWM5输出
3	PIO13	P0.4/PWM3管脚功能选择位 0 = P0.4/PWM3 管脚用作 P0.4 1 = P0.4/PWM3 管脚用作 PWM3输出
2	PIO12	P0.5/PWM2管脚功能选择位 0 = P0.5/PWM2 管脚用作 P0.5 1 = P0.5/PWM2 管脚用作 PWM2输出
1	PIO11	P1.4/PWM1管脚功能选择位 0 = P1.4/PWM1 管脚用作 P1.4 1 = P1.4/PWM1 管脚用作PWM1输出

PWMDTEN – PWM死区时间使能

寄存器	SFR地址	复位值
PWMDTEN	F9H, 所有页,TA保护	0000_0000 b

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

位	名称	描述
4	PDTCNT8	PWM 8位死区时间计数器 见 PWMnDTCNT 寄存器.

位	名称	描述
2	PDT45EN	PWM4/5 配对插入死区时间使能 该位仅当PWM4/5在配对模式下有效。 0 = GP4/GP5配对信号无延时 1 = GP4/GP5配对信号上升沿时插入死区时间延时。
1	PDT23EN	PWM2/3 配对插入死区时间使能 该位仅当PWM2/3在配对模式下有效。 0 = GP2/GP3 配对信号无延时 1 = GP2/GP3配对信号上升沿时插入死区时间延时。
0	PDT01EN	PWM0/1 配对插入死区时间使能 该位仅当PWM2/3在配对模式下有效。 0 = GP0/GP1 配对信号无延时 1 = GP0/GP1配对信号上升沿时插入死区时间延时。

PDTCNT –PWM死区时间计数器 (TA protected)

寄存器	SFR地址	复位值
PWM0DTEN	FAH, 所有页,TA保护	0000_0000 b

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

位	名称	描述
7:0	PWM0DTCNT [7:0]	PWM死区时间计数器低字节 该8位与PWMnDTEN .4 组成一个9位PWM死区时间计数器PWM0DTCNT。该计数器只有当PWM在互补模式下且相应的PWMnDTEN 位设置时才有效。 $\text{PWM死区时间} = \frac{\text{PDTCNT} + 1}{F_{\text{SYS}}}$ 注：在PWM运行过程中，用户不能更改PWM0DTCNT。

PMEN –PWM 输出掩码控制

寄存器	SFR地址	复位值
PMEN	FBH, 所有页	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

Address: FBH Reset value: 0000 0000b

位	名称	描述
---	----	----

位	名称	描述
n	PMENn	PWMn 输出掩码使能位 0 = PWMn 信号输出 1 = PWMn 根据PMDn设定的电位值掩码

PMD –PWM 掩码数据寄存器

寄存器	SFR地址	复位值
PMD	FCH, 所有页	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
n	PMDn	PWMn 掩码数据寄存器 一旦相应的PMENn 被设置，PWMn 信号输出掩码电平值 0 = PWMn 掩码，输出低电平 1 = PWMn 掩码，输出高电平

20 12位模数转换器(ADC)

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8信道单端输入模式。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ADC结果寄存器中。

20.1 功能描述

20.1.1 ADC 工作方式

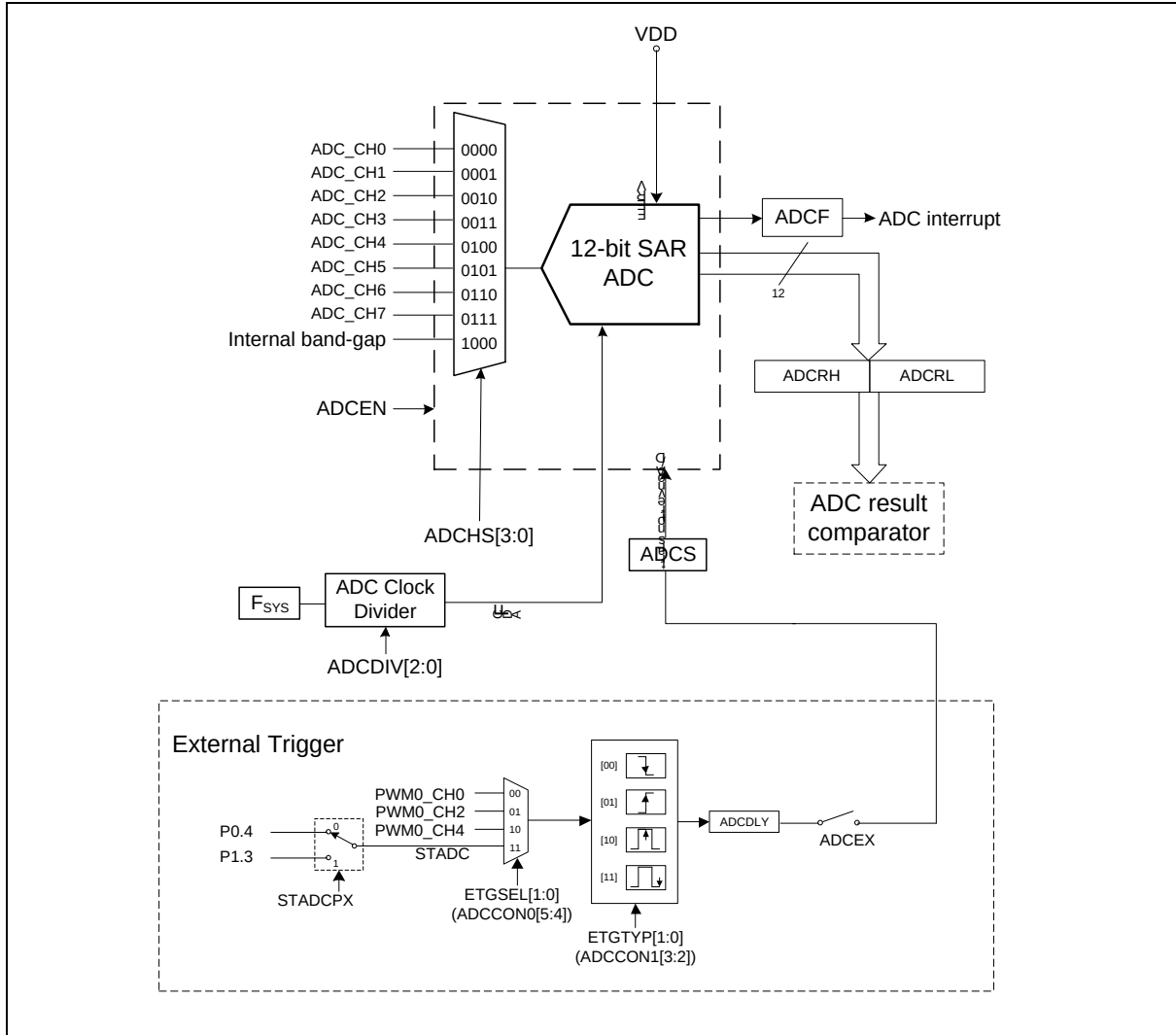


图 20.1- 1 12-位 ADC 模块结构图

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500ksps 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

ADCRL (低4位)中。12位转换结果值为 $4095 \times \frac{V_{AIN}}{V_{REF}}$ 。

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。

20.1.2 外部触发ADC

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 (触发延時計数器)。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。。

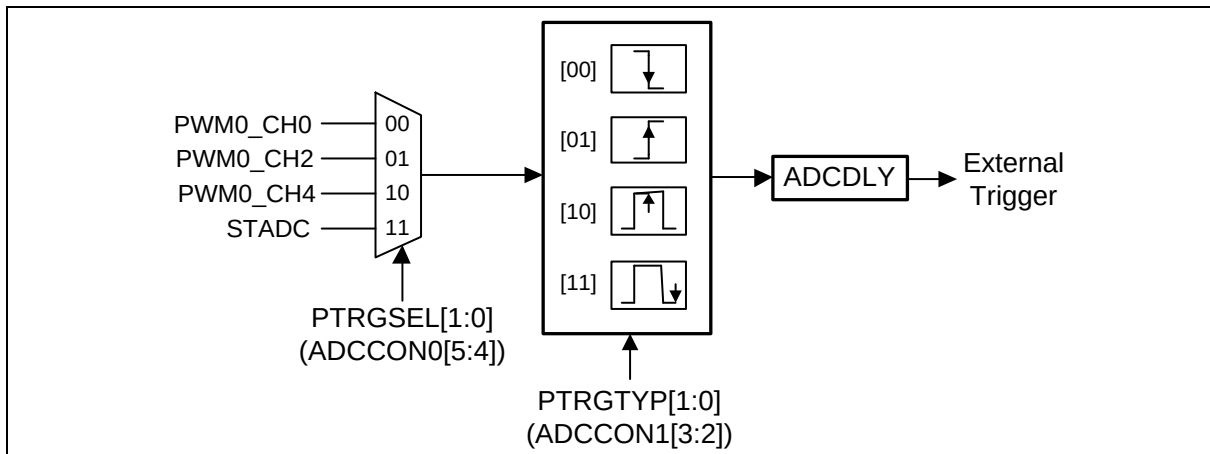


图 20.1- 2 外部触发 ADC 电路结构

20.1.3 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 ACMPL 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。。

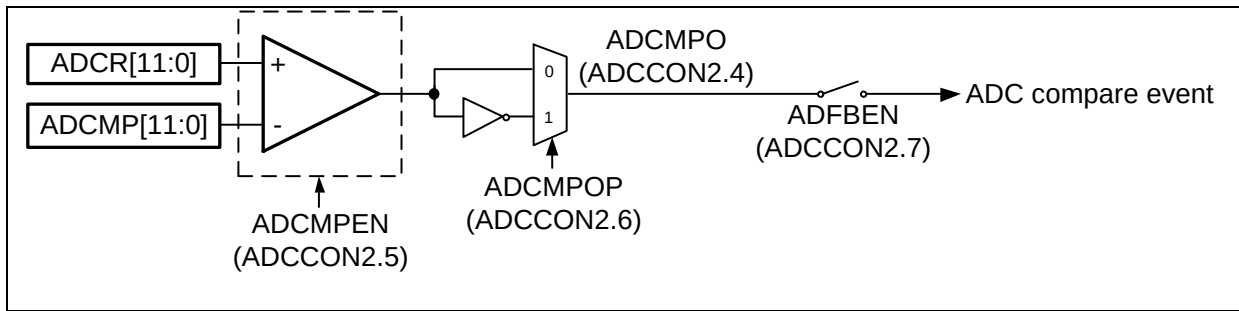


图 20.1- 3 ADC 结果比较器

20.1.4 带隙电压（Band-gap）输入应用

MS51的ADC可以选择内部Band-gap电压作为输入脚，能用来反推ADC的参考电压，得到更精准的ADC结果。

常温下，所有MS51的Band-gap电压值会校准在1.17V ~ 1.27 V内。如果想要知道每颗MS51的Band-gap实际电压值，可以通过读取UID地址后的2个bytes值来确认。UID 后的2个bytes储存Band-gap的实际值，总共12个有效bit。第一个byte是高8位，第二个byte的低4位为12 bit的低4位。

读取和计算步骤：

1. 用读取UID的方式，以IAP 读取Band-gap的值为例
2. 合并高8位和低4位
3. 使用下列公式换算成实际的电压值：

计算公式：

$$Bandgap_Voltage = \frac{Bandgap_Value \times 3072}{4096} (mV)$$

计算范例：

读取UID地址后的2个字节，第一个字节值为0x64，第二个字节值为0x0E，合并为0x64E = 1614，换算电压值结果为：

$$Bandgap_Voltage = \frac{1614 \times 3072}{4096} = 1210.5 (mV)$$

20.2 ADC控制寄存器

ADCCON0 –ADC控制寄存器0

寄存器	SFR地址	复位值
ADCCON0	E8H, 所有页, 可位寻址	0000_0000 b

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.
3:0	ADCHS[3:0]	A/D转换通道选择 该位用于选择ADC转换通道。当ADCEN 为 0所有输入无效。 0000 = AIN0. 0001 = AIN1. 0010 = AIN2. 0011 = AIN3. 0100 = AIN4. 0101 = AIN5. 0110 = AIN6. 0111 = AIN7 1000 =内部带隙电压(band-gap)1.22V. Others = 保留.

ADCCON1 –ADC控制寄存器1

寄存器	SFR地址	复位值
ADCCON1	E1H, 所有页	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
6	STADCPX	外部启动ADC触发脚选择 0 = 分配 STADC 到 P0.4. 1 = 分配 STADC 到 P1.3. 注意：一旦该位置位或清0，STADC脚会立即转换。
5:4	ADCDIV[1:0]	外部触发信号类型选择 当 ADCEX (ADCCON1.1) 置1，该位决定响应外部触发的类型。 00 = PWM0/2/4 或 STADC 脚的下降沿. 01 = PWM0/2/4 或 STADC 脚的上升沿. 10 = 一个PWM周期的中点. 11 = 一个PWM周期的终点. 注PWM周期中点或终点触发仅适用于中心对齐模式的PWM输出。
3:2	ETGTYP[1:0]	ADC 触发启动信号选择位 该位决定启动ADC的触发条件 0 = 当软件设定ADCS位，启动AD转换 1 = 当软件设定ADCS位或外部触发信号启动AD转换。外部触发信号条件由寄存器 ETGSEL[1:0] 及 ETGTYP[1:0]决定。注，当ADCS为1时（正在转换），外部触发信号不会影响ADC直到ADC转换结束，ADCS被硬件清0。
1	ADCEX	ADC 使能位 0 = ADC 转换电路关闭 1 = ADC 转换电路打开
0	ADCEN	外部启动ADC触发脚选择 0 = 分配 STADC 到 P0.4. 1 = 分配 STADC 到 P1.3. 注意：一旦该位置位或清0，STADC脚会立即转换。

ADCCON2 –ADC控制寄存器 2

寄存器	SFR地址	复位值
ADCCON2	E2H, 所有页	0000 _0000 b

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

位	名称	描述
7	ADFBEN	ADC比较结果响应故障刹车使能寄存器 0 = 关闭 1 = ADC 触发故障刹车功能打开 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 (PWMCN0.7)，并终止PWM输出。当PWMRUN置1，PWM重新输出。
6	ADCMPOP	ADC比较器输出极性选择位 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	ADCMPEN	ADC 结果比较使能位 0 = ADC 结果比较功能关闭。 1 = ADC 结果比较功能打开。
4	ADCMPO	ADC比较结果输出位 该位输出ACMPPOP设定比较输出的结果。每次AD转换结束都会更新输出。

位	名称	描述
3:1	ADCAQT	ADC 采集时间 这个3位字段决定ADC采样的捕获时间，遵循下面的等式： $\text{ADC 采集时间} = \frac{4 * \text{ADCAQT} + 6}{F_{\text{ADC}}}$ 默认及最小捕获时间是10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 外部触发延迟计数器第8位 见 ADCDLY 寄存器。

AINDIDS –ADC 信道数字输入功能关闭

寄存器	SFR地址	复位值
AINDIDS	F6H, 所有页	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 信道数字输入功能关闭 0 = ADC 信道n数字输入功能打开 1 = ADC 信道n数字输入功能关闭。ADC通道n读取始终为0

ADCDLY –ADC外部触发延迟计数器

寄存器	SFR地址	复位值
ADCDLY	E3H, 所有页	0000_0000 b

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

位	名称	描述
7:0	ADCDLY[7:0]	ADC 外部触发启动延迟计数器低位 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 $\text{外部延迟时间} = \frac{\text{ADCDLY}}{F_{\text{ADC}}}$ 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。

ADCRH –ADC转换结果高位寄存器

寄存器	SFR地址	复位值
ADCRH	C3H, 所有页	0000 _0000 b

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

Bit	Name	Description
7:0	ADCR[11:4]	ADC转换结果高位 ADC转换结果高8位

ADCRL –转换结果低位寄存器

寄存器	SFR地址	复位值
ADCRL	C2H, 所有页	0000 _0000 b

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

位	名称	描述
3:0	ADCR[3:0]	ADC转换结果低位 ADC转换结果低4位

ADCMPL –ADC比较值高位寄存器

寄存器	SFR地址	复位值
ADCMPL	CFH,所有页	0000 _0000 b

7	6	5	4	3	2	1	0
ADCMPL[11:4]							
W/只读							

位	名称	描述
7:0	ADCMPL[11:4]	ADC比较值高字节 ADC比较值高字节8位内容

ADCMPL –ADC比较值低位寄存器

寄存器	SFR地址	复位值
-----	-------	-----

ADCMPL	CEH,所有页	0000_0000 b
--------	---------	-------------

7	6	5	4	3	2	1	0
-	-	-	-	ADCMP[3:0]			
-	-	-	-	W/只读			

位	名称	描述
3:0	ADCMP[3:0]	ADC 比较值低位 ADC比较值低4位

AUXR1 – 辅助寄存器 1

寄存器	SFR地址	复位值
AUXR1	A2H, 所有页	POR 0000 0000b 软件复位 1U00 0000b, nRESET 复位脚 U100 0000b, 其它 UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	UART1PX	DPS
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
4	SLOW	ADC 慢速选择 此位用来选择ADC慢速 0 = 速度大于 500 ksp/s 1 = 速度慢于 200 ksp/s

21 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]	时效访问 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。

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)	
(SETBEA)	;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。

22 中断系统

22.1 中断概述

中断的目的是让软件处理非常规或异步的事件。MS51有4个中断优先级，30个中断源。每一个中断源有一个独立的优先级设置位、中断向量和使能位。此外，中断可以全局使能或关闭。当中断发生时，CPU将执行对应的中断服务程序(ISR)。ISR被分配到预先指定的地址，如表 22.1- 1所示。如果中断使能，当中断发生时，CPU 将根据中断源跳转到相应的中断向量地址，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	-	SPI0 中断	004BH	9
外部中断0	0003H	0	WDT 中断	0053H	10
定时器0溢出	000BH	1	ADC 中断	005BH	11
外部中断1	0013H	2	输入捕获中断	0063H	12
定时器1溢出	001BH	3	PWM 中断	006BH	13
串口0中断	0023H	4	故障刹车中断	0073H	14
定时器2事件	002BH	5	串口1中断	007BH	15
I2C0状态/超时中断	0033H	6	定时器3溢出	0083H	16
引脚中断	003BH	7	自唤醒定时器中断	008BH	17
掉电检测中断	0043H	8			

表 22.1- 1 中断向量表

22.2 中断使能

每一个中断源都可以通过各自的中断使能位开启或关闭，这些位在IE和EIE特殊功能寄存器SFRs中。有一个全局使能中断EA(IE.7)位，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。注意：当EA为0时有中断请求，所有中断会被挂起直到EA恢复为1，才去执行该中断。所有中断标志位可以用软件置位，也可以用软件启动中断。

注意:每一个中断产生时对应中断标志位都会被置1，不管是通过硬件还是软件。用户在中断服务程序里应该小心处理中断标志位，大多数中断标志位都是写0清除，这样可以避免递归中断请求

22.3 中断优先级

系统为所有中断源提供4种优先级：最高（3级）、高（2级）、低（1级）、最低（0级）。中断源可以单独设置各自的优先级位来配置其优先级。表 22.3- 1列举了4种优先级配置。相对来说，低优先级中断可以被高优先级中断打断，但不能被同等优先级或更低的优先级打断。默认优先级可以帮助中断控制器解决同等优先级同时请求中断的状况。

在多个中断时，遵循以下规则：

1. 当一个低优先级中断正在运行，这时一个高优先级产生，该中断会被打断去执行高优先级中断。当高优先级中断执行完RETI后，低优先级中断恢复继续运行。当低优先级中断执行完RETI后，控制器把运行权利交回给主程序。
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 22.3- 2所示，总结了中断源、标志位、向量地址、使能位、优先级位和允许CPU可以从掉电模式中唤醒。CPU从掉电模式中唤醒。

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 22.3- 1 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	-	Always Enabled	Highest	-	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
I ² C 状态/超时	0033h	SI + I2TOF (I2TOC.0)	EI2C	5	PI2C, PI2CH	No
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

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
串口 0	0023H	RI + TI	ES	10	PS, PSH	No
故障刹车中断	0073h	FBF (FBD.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
PWM中断	006BH	PWMF	EPWM	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
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, 在定时器 2x工作在自动加载模式下, 执行中断服务程序时TF2x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 22.3- 2 各级中断源特性表

22.4 中断服务

中断标志位在每个系统时钟周期都会被采样。在同一个周期内, 被采样到的中断和优先级都会被解决。如果满足特定的条件硬件将执行内部产生的LCALL指令, 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。

如果以上任何一个条件不满足, 就不能产生LCALL指令。在每一个指令周期都会重新检测中断标志。当某个中断标志位被置起, 但没有满足上述条件都不会被响应, 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。对应的中断标志根据不同的中断源在执行中断服务程序时, 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, 保存程序计数器PC内容到堆栈, 但不保存程序状态字PSW, PC指针重新装载产生中断的中断向量地址, 从向量地址继续执行程序直到执行RETI指令。在执行RETI指令时, 处理器弹出堆栈, 将栈顶内容加载到程序计数器PC。用户必须注意堆栈的状态, 如果堆栈的内容被修改, 处理器不会被通知, 将会从堆栈加载的地址继续执行。注, RET指令与RETI指令表现相同, 但它不会通知中断控制器中断服务已经完成, 致使控制器认为中断服务仍在进行。使中断不可能再产生。

22.5 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是也没有更高优先级中断产生）是执行RETI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RETI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。

22.6 外部中断

外部中断 $\overline{INT0}$ 和 $\overline{INT1}$ 被作为中断源。它们可以根据IT0 (TCON.0) 和 IT1 (TCON.2)选择边沿触发，还是电平触发中断。检测IE0 (TCON.1) 和 IE1 (TCON.3)标志位判断是否产生中断。在边沿触发模式下，每个系统时钟周期都会采样 $\overline{INT0}$ 或 $\overline{INT1}$ 输入状态。如果在一个周期中采样是高电平，然后在下一个周期中为低电平，那么这个高到低的电平转换将会被检测到，并且置位中断请求标志IE0或IE1。由于每个系统时钟周期都去采样外部中断，所以高电平或低电平至少保持一个系统时钟周期。当中断服务程序被执行时，IE0和IE1会被硬件自动清除。如果选择电平触发模式，那么必须保持引脚为低电平，直到进入中断服务，在进入中断服务程序时IE0和IE1不会被硬件清除。在电平触发模式下，IE0和IE1标志位与 $\overline{INT0}$ 和 $\overline{INT1}$ 引脚逻辑电平值相反。当中断服务程序结束后引脚依然保持低电平，处理器会响应另一个来自同一中断源的中断。 $\overline{INT0}$ 和 $\overline{INT1}$ 均支持将芯片从掉电模式唤醒。

22.7 中断控制寄存器

IE – 中断使能

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/禁止所有中断，单独使能。 0 = 禁止所有中断源 1 = 每个中断使能依靠使能单个中断，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。

位	名称	描述
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中断使能. 当中断产生时, IE0(TCON.1)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中断使能. 当中断产生时, IE1(TCON.3)置1

EIE – 扩展中断使能寄存器

寄存器	SFR地址	复位值
EIE0	9BH, 所有页	0000_0000 b

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

位	名称	描述
7	ET2	定时器2中断使能 0 = 定时器2中断禁止. 1 = 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 (SPInSR.7), SPIOVF (SPInSR.5), 或 MODF (SPInSR.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 WDTE (WDCON.5) 置 1.
3	EPWM	PWM0中断使能 0 = PWM 中断禁止. 1 = PWM 中断使能. 当中断产生时 PWMF (PWMnCON0.5) 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.

位	名称	描述
0	EI2C	I ² C0 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 (I2C0CON.3) 或 I2TOF (I2C0TOC.0) 置 1.

EIE1 – 拓展中断使能寄存器 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
-	-	-	-	-	EWKT	ET3	ES_1
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	EWKT	WKT中断使能 0 = WKT 中断禁止. 1 = 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.

IP – 中断优先级寄存器

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

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
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	Timer 0 i中断优先级高位
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 22.3- 1

EIP – 扩展中断优先级

寄存器	SFR地址	复位值
EIP0	EFH, 页0	0000_0000 b

7	6	5	4	3	2	1	0
PT2	PSPI0	PFB	PWDT	PPWM0	PCAP	PPI	PI2C0
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	PSPI0	SPI0 中断优先级低位
5	PFB	Fault Brake 中断优先级低位
4	PWDT	WDT 中断优先级低位
3	PPWM0	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	PI2C0	I2C 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 22.3- 1

EIPH –扩展中断优先级高位寄存器

寄存器	SFR地址	复位值
AINDIDS	F7H, 所有页	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

EIP1 –扩展中断优先级寄存器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
-	-	-	-	-	PWKT	PT3	PS_1
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	PWKT	WKT 中断优先级低位
1	PT3	定时器3 中断优先级低位
0	PS_1	串口1 中断优先级低位

EIPH1 –扩展中断优先级高位寄存器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
-	-	-	-	-	PWKTH	PT3H	PSH_1
-	-	-	-	-	读/写	读/写	读/写

位	名称	描述
2	PWKTH	WKT 中断优先级高位
1	PT3H	定时器3 中断优先级高位
0	PSH_1	串口1 中断优先级高位

TCON – 定时器 0 和 1 控制位

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1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

23 在应用编程 (IAP)

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提供方便FALSH编程方式，可以帮助用户通过IAP方式，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，并且设置IAPUEN寄存器的相应位，使能需要升级的FLASH区域（CONFIG、LDROM、APROM），用户将16位操作地址写入IAPAH和IAPAL，数据写入IAPFD，命令写入IAPCN。然后通过设置触发位IAPGO(IAPTRG.0)，去执行IAP。注意：IAPTRG也受TA保护。此时，CPU保持程序计数器，内嵌IAP自动控制内部充电泵提高电压和信号时序。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23.5μs。IAP动作完成后，程序计数器继续运行之后的指令，IAPGO位将自动清零。IAPFF（CHPCON.6）是IAP错误标志，可以用来检查之前IAP操作成功与否。通过这些纯软件的设置，用户可以很方便对FLASH存储器进行擦除、编程和校验。

23.1 IAP 命令

MS51 通过IAP可操作 APROM、LDROM或 CONFIG。IAP操作模式和编程区域是由IAP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: 50H 高字节 DID: 36H
读96位UID	XX	0	0	0100	0000H to 000BH	数据读出
APROM 页擦除	00	1	0	0010	地址 写入 ^[2]	FFH
LDROM 页擦除	01	1	0	0010	地址 写入 ^[2]	FFH
APROM 字节编程	00	1	0	0001	地址 写入	数据写入
LDROM 字节编程	01	1	0	0001	地址 写入	数据写入
APROM 字节读	00	0	0	0000	地址 写入	数据读出
LDROM 字节读	01	0	0	0000	地址 写入	数据读出
擦除所有CONFIG	11	1	0	0010	0000H	FFH
CONFIG 字节编程	11	1	0	0001	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H	数据写入
CONFIG 字节读	11	0	0	0000	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H	数据读出
注: 1.表示无关项 2.每一页是 128字节，所以地址应该是目标页的地址						

表 23.1- 1 IAP 模式及命令表

23.2 IAP控制寄存器

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	Brown-out inhibiting IAP This bit decide whether IAP erasing or programming is inhibited by brown-out status. This bit is valid only when brown-out detection is enabled. 1 = IAP erasing or programming is inhibited if V_{DD} is lower than V_{BOD} . 0 = IAP erasing or programming is allowed under any workable V_{DD} .

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
W	读/写	-	-	-	-	读/写	读/写

位	名称	描述
6	IAPFF	软件复位 对此位写1将导致软件复位. 复位完成后将通过硬件自动清除。
0	IAPEN	IAP 错误标志 满足以下任意条件, 硬件将在IAPGO(ISPTRG.0)置位后置位该位: (1) 访问地址越界. (2) IAPCN 命令无效. (3) IAP 擦除或编程更新没有使能的区域. (4) 当BOIAP(CONFIG2.5)为1、BODEN (BODCON0.7)为1以及BORST(BODCON0.2) 为0时, 擦除或编程工作在VBOD下。 此位必须由软件清除。

IAPUEN – IAP 更新使能

寄存器	SFR地址	复位值
IAPUEN	A5H, 所有页, TA保护	0000_0000 b

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

-	-	-	-	SPUEN	CFUEN	LDUEN	APUEN
-	-	-	-	读/写	读/写	读/写	读/写

位	名称	描述
7:4	-	保留
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 位升级使能 0 = 禁止通过IAP擦除或编程CONFIG字节 1 = 使能通过IAP擦除或编程CONFIG字节
1	LDUEN	LDROM 升级使能 0 = 禁止通过IAP擦除或编程LDROM 1 = 使能通过IAP擦除或编程LDROM
0	APUEN	APROM 升级使能 0 = 禁止通过IAP擦除或编程APROM 1 = 使能IAP擦除或编程APROM

IAPCN – IAP 控制

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

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

位	名称	描述
7:6	IAPB[1:0]	IAP 控制 此字节是IAP控制命令。详细描述见 表 23.1- 1
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

IAPAH – IAP 地址低字节

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

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

位	名称	描述
7:0	IAPA[15:8]	IAP 地址高位 IAPAH 包含地址 IAPA[15:8] 为 IAP 操作.

IAPAL – IAP 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 地址低位 IAPAL 包含地址 IAPA[7:0] 为 IAP 操作.

IAPFD – IAP 内存数据

寄存器	SFR地址	复位值
IAPFD	AEH, 所有页	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 flash数据 该字节包含将要读或写入flash空间的数据。编程模式下，用户需要在触发IAP之前写数据到IAPFD里，读/校验模式下，在IAP完成后从IAPFD读出数据。

IAPTRG – IAP 触发

寄存器	SFR地址	复位值
IAPTRG	A4H, 所有页, TA保护	0000_0000 b

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

位	名称	描述
0	IAPGO	IAP 执行 设置此位为1时开始执行IAP。在此指令之后，CPU保持程序计数器（PC），IAP硬件自动管理控制该进程。IAP操作完成后，程序计数器继续执行下一个指令。IAPGO位将被自动清除并保持为0。在触发IAP操作之前，由于硬件限制中断（如果启用）应该临时禁用。程序流程如下。 <pre> CLR EA MOV TA,#0AAH MOV TA,#55H ORL IAPTRG,#01H (SETB EA) </pre>

23.3 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规则，以确保IAP正确执行，否则可能引起不确定的结果，甚至损坏器件。此外，下文对于正确执行IAP有很好建议。

(1)没有IAP操作时，用户必须清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，禁止IAP将停止内部HIRC，可以达到省电的目的。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。

注：

1. 在进行IAP时，如果中断打开应该临时清除EA位
2. 擦除或编程的页不能是当前代码执行的页。否则会出现不可预计程序动作，甚至破坏存储的数据

23.4 使用Flash存储器作为数据存储

在一般应用中，有时需要一些数据在断电情况下不能丢失，以使用户读回或更新，作为系统控制的参数。MS51支持IAP功能并且存储在flash中的字节都可以用MOVC指令读取，所以很适合作为非易失数据存储。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 IAPG0 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:

```

//*****
// 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 IAPG0 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 IAPG0 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);
}

```

23.5 在线系统编程 (ISP)

Flash存储器支持硬件编程和应用编程（IAP）。如果产品在研发阶段或产品需要更新软固件时，硬件编程就显得不太方便，采用在系统编程（ISP）方式，可使这一过程变得方便。执行ISP不需要将控制器从系统板上拆下来。通过软件控制可以重新编程设备。因此这使得更新应用程序固件ISP得到广泛的应用。

用户可以开发自己的引导代码放在LDROM中。LDROM最大为4KB。用户开发的引导代码可以通过并行烧录器或是电路编程器（ICP）下载到LDROM中去。

一般来说，ISP是PC与MCU之间进行通讯。PC通过串口传输新的用户代码给MCU。然后引导代码接收这些数据，将这些数据通过IAP命令编程到用户代码区域。新唐针对MS51提供ISP固件和PC端软件，这样可以很容易实现ISP通过UART端口升级代码。更多信息请访问新唐8位微控制器网站：[新唐80C51微控制器技术支持](#)。

以下是简单ISP参考代码

汇编例程代码:

```

;*****
;
;This code illustrates how to do APROM and CONFIG IAP from LDROM.
;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

24 电源管理

MS51有几种工作模式可以帮助用户控制设备功耗，省电模式包括掉电模式和空闲模式。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 (P2S.7)内部上拉

PCON – 电源控制寄存器

寄存器	SFR地址	复位值
PCON	87H, 所有页	POR: 0001_000b, 其它: 000U_0000b

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

位	名称	描述
1	PD	通用标志 0 通用标志可由用户通过软件置位和清零。
0	IDL	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。此时为最小功耗。CPU从掉电模式下唤醒后，该位自动由硬件清零，且在系统唤醒之前程序继续执行中断服务程序（ISR）。从ISR返回后，设备继续执行系统进入掉电模式时所处的指令。 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。

24.1 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。用户可以通过向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，并将执行中断源对应的中断服务程序(ISR)，直到执行RETI返回指令，返回后立即执行使CPU进入空闲模式的指令，使CPU进入空闲模式。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。

24.2 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BOD欠压检测功能。当然nRESET引脚的复位或上电复位也可以使CPU退出掉电模式。nRESET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。

25 时钟系统

MS51拥有多种时钟源可供选择，这样在应用中可以有多种选择，使系统性能发挥到最佳，并且功耗降到最低。共有3种系统时钟源可供选择，包括:内部振荡器、来自 XIN 引脚的外部时钟。可以通过软件设置选择。MS51内嵌2个内部RC振荡器一个10 kHz低速、一个 16MHz 高速RC振荡器，高速16MHz误差在出厂时校准到 $\pm 2\%$ （全温度、全电压范围内）。CKDIV除频器可以灵活地调整MS51功耗与性能。

25.1 时钟系统结构

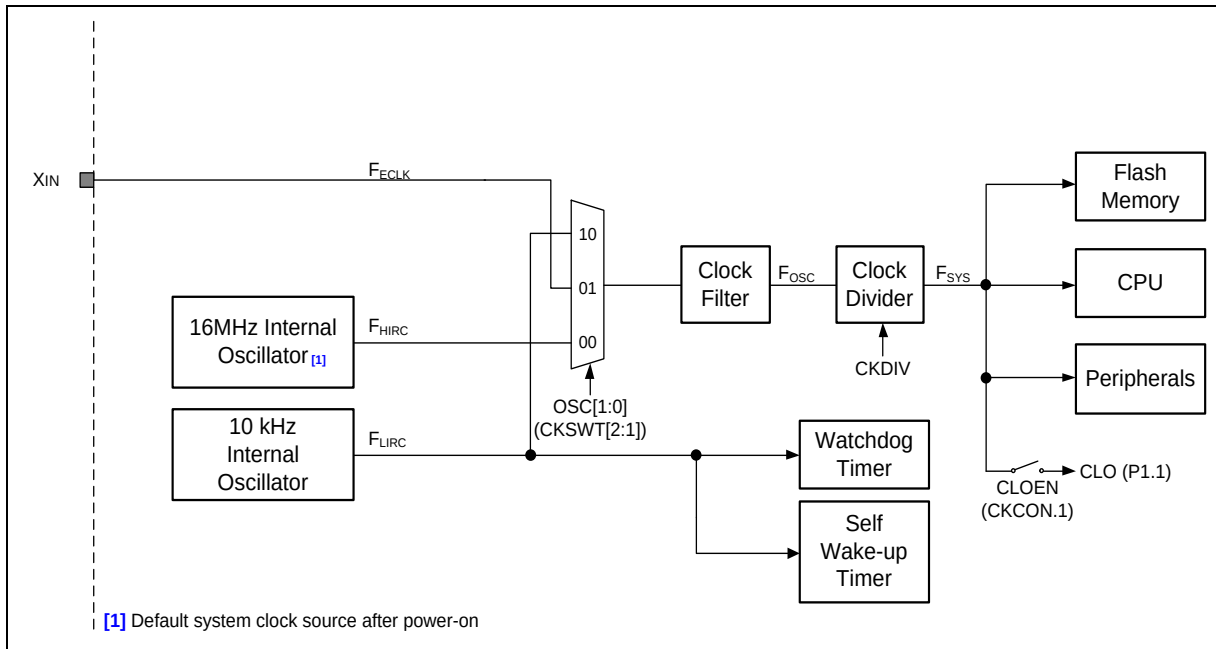


图 25.1- 1 时钟系统框图

25.2 系统时钟源

MS51共有3种系统时钟源，包括: 内部高速/低速振荡器、外部输入时钟。它们每一个都可以作为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P3.0/XIN。

25.3 内部振荡器

MS51内部有两个RC振荡器，一个高速16MHz或24MHz（HIRC），可通过HIRC24(RCTRIM1.4)位进行选择配置，以及一个低速10 kHz（LIRC）。它们都可被选择用作系统时钟。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 P3.0，用来扩展普通I/O数量。可以通过配置P3M1和P3M2寄存器来选择P3.0的输出模式。

因为HIRC数据共有9位，用户可通过更改SFR直接对HIRC值进行更改，每更改一个位的数值，大约更改HIRC频率大约40kHz。HIRC数值减小，HIRC频率增快，也就是RCTRIM的值减1，系统频率增加40Khz。下列寄存器定义9位的RC值。

注： 对于24MHz的HIRC预设值，建议不要直接更改HIRC24位，而是采用IAP读取HIRC预设值，否则HIRC误差可能会偏差超过1%， IAP读取预设值的方式可参考BSP。

RCTRIM0 –内部高速HIRC设定值0

寄存器		SFR地址	复位值
RCTRIM0		84H, 所有页, TA保护	16MHz HIRC 预设值

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

RCTRIM1 –内部高速HIRC设定值1

寄存器		SFR地址	复位值
RCTRIM1		85H, 所有页, TA保护	16MHz HIRC 预设值

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

25.4 系统时钟切换

MS51可以通过软件设置CKSWT 和 CKEN寄存器切换时钟源。这给应用带来了很大方便。注意，这些特殊寄存器是写时效访问控制受TA保护寄存器。在时钟源控制下，时钟源可以在外部时钟、内部时钟，甚至内部高速与低速之间自由的切换。然而在切换时钟源时，必须确保待切换时钟源已稳定。因此，用户需要遵循以下设置步骤才能成功完成时钟源切换。用户首先要通过配置CKEN寄存器打开目标时钟源，再通过查询CKSWT寄存器中对应的标志位，确定时钟源是否稳定，并且通过写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。如果用户关心功耗的话，可以将原先时钟源关闭了。如果不遵守以上步骤，硬件将会采取以下一些措施来应对这些违规的操作。

- 1.如果用户试图改变CKEN的值来关闭当前时钟源，设备将忽略这个操作。系统时钟维持现状，CKEN值不变。
- 2.如果用户试图改变OSC[1:0]的值来切换系统时钟，而待切换新时钟源未被打开，OSC[1:0]值将会被立即更新，但是系统时钟保持不变，CKSWTF (CLKEN.0)会被硬件置位。
- 3.如果用户切换系统时钟源，但是目标时钟源已经打开还没稳定，那么硬件会等待目标稳定后，再切换过去。在等待期间，设备继续以原来时钟源工作，并且CKSWTF会被置1。等到目标时钟源稳定后，标志位(见 CKSWT[7:3])被置位，时钟将会成功切换，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
-	-	HIRCST	LIRCST	ECLKST	OSC[1:0]		-
-	-	只读	只读	只读	W		-

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

位	名称	描述
2:1	OSC[1:0]	振荡器选择位 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 = 保留位 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 (TA 保护)

寄存器	SFR地址	复位值
CKEN	97H, 所有页, TA保护	0011_0000 b

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

位	名称	描述
7:6	EXTEN[1:0]	外部时钟源使能 11 = 使能外部时钟，通过XIN输入外部时钟信号 其它 = 禁止外部时钟。P30作为普通I/O使用
5	HIRCEN	HIRC16MHz 使能位 0 = 关闭 HIRC 1 = 使能 HIRC 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
4:1	-	保留位
0	CKSWTF	时钟切换错误标志位 0 = 先前的系统时钟源切换成功 1 = 用户试图将先前的系统时钟源切换到没有开启或是不稳定的时钟源。如果待切换的时钟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。

25.5 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，可以让MCU在正常工作模式下，很低的速度运行，确保其及时能够响应中断事件（比如空闲模式只能通过中断事件退出）。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。CKDIV的值可以在任何时间被程序改变，除了不能在中断服务程序里改变。

CKDIV – 时钟分频器

寄存器	SFR地址	复位值
CKDIV	95H, 所有页	0000_0000 b

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

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

25.6 系统时钟输出

MS51 提供一个 CLO(P1.1) 引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, 所有页	0000_0000b

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

位	名称	描述
1	CLOEN	系统时钟输出使能 0 = 系统时钟输出禁止 1 = 系统时钟输出使能，从 CLO 输出

26 电源监控

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。

26.1 上电复位(POR)

上电检测功能，用于检测电源上升到系统可以工作的电压。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。

PCON – 电源控制寄存器

寄存器	SFR地址	复位值
PCON	87H, 所有页	POR: 0001_000b, 其它: 000U_0000b

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

位	名称	描述
4	POF	低功耗运行模式 0 = 禁止 1 = 使能 注： PD = 1 且LPR = 1时，LPR 无效, CPU 将进入掉电模式.

26.2 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V_{DD}下降到所选择的欠压检测触发电平(V_{BOD})，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V_{BOD}，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V_{DD}下降到V_{BOD}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，V_{DD}依然保持在V_{BOD}下，BOF不会被再次置1，BOF仅通知用户电源电压下降发生。当V_{DD}上升到高于V_{BOD}时，BOF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。它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支持低功率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。注，BODCON0带有时效访问TA保护。

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表 26.2- 1

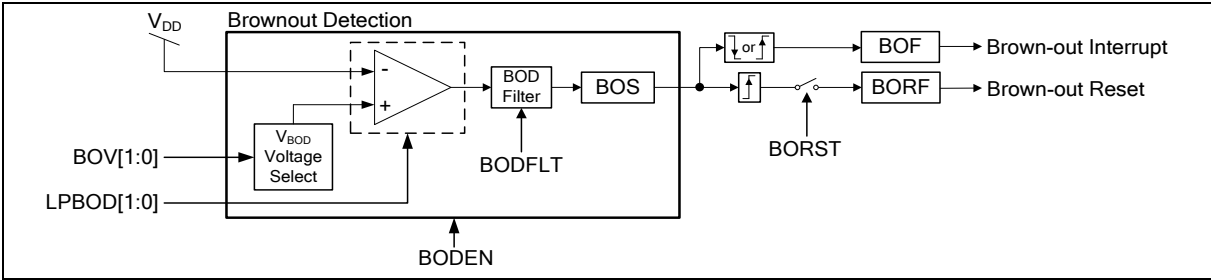


图 26.2- 1 欠压检测框图

CONFIG2

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

出厂预设值: 1111 1111b

位	名称	描述
7	CBODEN	配置欠压侦测使能位 1 = 欠压检测功能打开. 0 = 欠压检测功能关闭
5:4	CBOV[1:0]	配置欠压侦测电压选择位 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.

位	名称	描述
2	CBORST	配置欠压检测复位使能 该位决定在电源电压跌到VBOD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 26.2- 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_XC0Xb BOD: UUUU_XU1Xb 其它: UUUU_XUUXb

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]	欠压侦测电压选择位 11 = VBOD 生效电压 2.2V. 10 = VBOD 生效电压 2.7V. 01 = VBOD 生效电压 3.7V. 00 = VBOD 生效电压 4.4V.
3	BOF	欠压中断标志 当VDD下降到VBOD以下或VDD上升到VBOD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VBOD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。

位	名称	描述
0	BOS	欠压状态标志 在BOD电路开启时，该位反应VDD与VBOD比较情况。BOD电路关闭时保持为0。 0 = VDD电压大于VBOD或是BOD电路关闭 1 = VDD电压小于VBOD 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V _{DD} 的电平。请参照表 26.2- 1		

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_0001b Others:0000_0UUUb

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

位	名称	描述
7:3	-	保留
2:1	LPBOD[1:0]	低功率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已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	最小电压检测脉冲宽度
0	正常工作模式 (LPBOD[1:0] = [0,0])	任意时钟源	Typ. 1μs
	低功耗模式 1 (LPBOD[1:0] = [0,1])	任意时钟源	16 (1/F _{LIRC})
	低功耗模式 2 (LPBOD[1:0] = [1,0])	任意时钟源	64 (1/F _{LIRC})
	低功耗模式 3 (LPBOD[1:0] = [1,1])	任意时钟源	256 (1/ F _{LIRC})
1	正常工作模式 (LPBOD[1:0] = [0,0])	HIRC/ECLK	正常工作模式: 32 (1/F _{SYS}) 空闲模式: 32 (1/F _{SYS}) 低功耗模式: 2 (1/F _{LIRC})
		LIRC	2 (1/F _{LIRC})
	低功耗模式 1 (LPBOD[1:0] = [0,1])	任意时钟源	18 (1/F _{LIRC})
	低功耗模式 2 (LPBOD[1:0] = [1,0])	任意时钟源	66 (1/F _{LIRC})
	低功耗模式 3 (LPBOD[1:0] = [1,1])	任意时钟源	258 (1/ F _{LIRC})

表 26.2- 2 最小欠压检测脉冲宽度

27 复位

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。

27.1 上电复位

MS51包含内部上电参考电压复位。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，上电标志 POF (PCON.4)置1表示冷复位，上电复位完成。注：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

寄存器	SFR地址	复位值
PCON	87H, 所有页	POR: 0001_000b, 其它: 000U_0000b

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

位	名称	描述
4	POF	上电复位标志 当上电后该位置1，用以标示当前冷复位，上电复位完成。其它任何复位不会影响该位，建议通过软件清零

27.2 欠压复位

欠压检测电路用于监测系统运行时V_{DD}电平。当V_{DD}下降到所选的欠压触发电平 (V_{BOD})，如果BORST (BODCON0.2) 置 1，CPU将欠压复位。发生欠压复位后，BORF (BODCON0.1)通过硬件自动置1，除上电复位或欠压复位，该位不会置1，该位可通过软件设置或清除。

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_XC0Xb BOD: UUUU_XU1Xb 其它: UUUU_XUUXb

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

位	名称	描述
1	BORF	欠压复位标志 当MCU发生欠压复位，该位将被硬件置1，建议复位发生后通过软件清零

27.3 外部复位

外部复位引脚nRESET是带史密特触发器的输入引脚。外部nRESET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。

在复位条件下，只要nRESET引脚电平从低到高，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，然后退出，并从地址0000H处开始执行程序。

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。

AUXR1 – 辅助寄存器 1

寄存器	SFR地址	复位值
AUXR1	A2H, 所有页	POR 0000 0000b 软件复位 1U00 0000b, nRESET 复位脚 U100 0000b, 其它 UUUU 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	UART1PX	DPS
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位	名称	描述
6	RSTPINF	外部复位标志位 当MCU发生外部复位脚复位，该位将被硬件置1，建议复位发生后软件清0。
5	HardF	硬件错误复位标志 一旦程序计数器（PC）的值超过flash的大小，MCU将会复位并且该位通过硬件置位。建议通过软件清除该位。 注意：如果MCU运行在OCD调试模式并且OCDEN = 0，硬件错误复位将会被禁止。

27.4 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，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 (WDCON0.3)将置位。除上电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, 所有页, TA保护	POR: 0000 0111b WDT: 0000 1UUUb 其他: 0000 UUUUb

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

位	名称	描述
3	WDTRF	看门狗复位标志 WDT 复位标志。当MCU复位时，该位由硬件置位。通过软件清零

27.5 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

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
W	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, 所有页	POR 0000 0000b 软件复位 1U00 0000b, nRESET 复位脚 U100 0000b, 其它 UUUU 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	UART1PX	DPS
读/写	读/写	读/写	读/写	读/写	读/写	读/写	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。建议软件清0。

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

27.6 启动选择

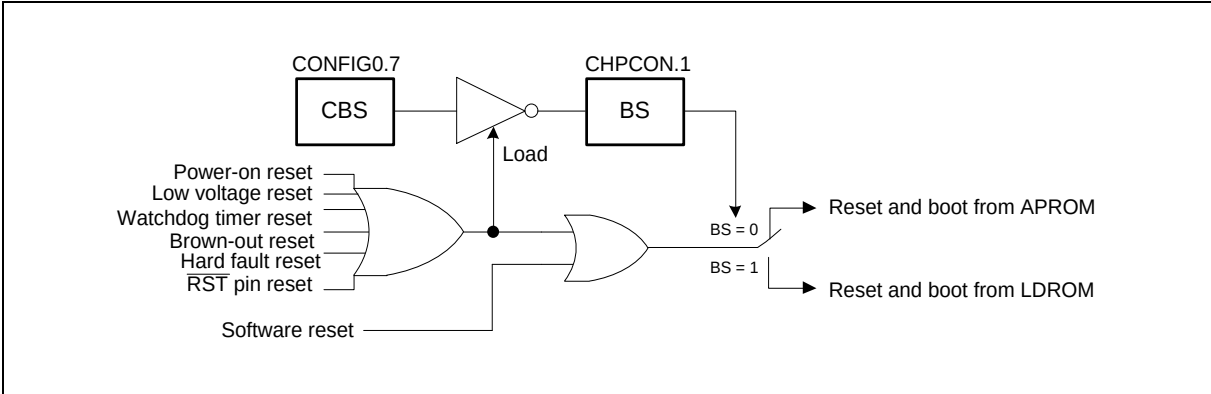


图 27.6- 1 启动选择结构图

MS51提供给用户灵活的启动选择，适合不同的应用。CHPCON.1 BS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APPROM 中启动，由0000H开始运行。反之CPU 从 LDROM启动，从0000H开始运行。注：复位（除软件复位）后，BS加载CBS位（CONFIG0.7）的相反值。

注：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。

CONFIG0

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

出厂预设值: 1111 1111b

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

CHPCON – 芯片控制寄存器

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 ^[1]	IAPEN
W	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
注： 1该位由复位后（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。		

27.7 复位状态

复位状态，除上电复位不会影响片上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。

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。见表 7.3- 1，一旦复位，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。

复位后，除了ICE_DAT 及 ICE_CLK脚外默认的配置是高阻输入模式。ICE_DAT和 ICE_CLK 脚复位后先保持 600 个 LIRC时钟周期的准双向带上拉电阻的模式，然后再转为输入高阻模式。

28 辅助功能

28.1 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 (AUXR1.0)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  AUXR0              ;change active DPTR
MOV  DPTR, #S_Addr     ;load source address
LOOP:
MOVX A, @DPTR          ;read source data byte
INC  AUXR0              ;change DPTR to destination
MOVX @DPTR, A          ;write data to destination
INC  DPTR               ;next destination address
INC  AUXR0              ;change DPTR to source
INC  DPTR               ;next source address
DJNZ R0, LOOP
INC  AUXR0              ;(optional) restore DPS
```

AUXR1 位3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

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]	数据指针高字节 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

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]	数据指针低字节 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。

AUXR1 – 辅助寄存器 1

寄存器	SFR地址	复位值
AUXR1	A2H, 所有页	POR 0000 0000b 软件复位 1U00 0000b, nRESET 复位脚 U100 0000b, 其它 UUU0 0000b

7	6	5	4	3	2	1	0
SWRF	RSTPINF	HardF	SLOW	GF2	UART0PX	UART1PX	DPS
读/写	读/写	读/写	读/写	读/写	读/写	读/写	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，前有效DPTR寄存器内的值保持不变。

28.2 96位序列号(UID)

出厂前，每颗 MS51都会预烧一个96位的序列号，用以确保该芯片的唯一性，这个唯一代码被称为序列号UID (Unique Code)。用户获得序列号的唯一方式是通过IAP指令读取，详见 章节

29 片上调试 (OCD)

29.1 功能描述

MS51内嵌片上调试功能（OCD），这为软件开发者提供了低成本调试方法，并且MS51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、单步运行、全速运行和非侵入命令的内存访问。OCD系统并不占用任何本地内存，也不和片上外设共享资源。

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是输入口，调试时传输数据同步用的时钟。P2.0/Nreset 引脚也是必不可少的，它是用来控制OCD模式进入和退出的。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。在设计系统用到OCD时，必须考虑下面一些限制条件：

1. P2.0/nRESET 配置成外部复位引脚时，它不能直接连接到V_{DD}上并且要和所有外部复位设备断开
2. 如果 P2.0/ nRESET配置成输入引脚时，必须和外部输入源断开
3. 所有外部复位源必须断开
- 4.所有与ICE_DAT和ICE_CLK连接的外围器件必须断开

29.2 OCD限制条件

由于MS51功能比较丰富，而引脚比较有限，所以一个引脚上可能多个功能。使用OCD系统时肯定会牺牲一些功能，主要有以下一些限制条件：

1. OCD模式用到P2. 0/nRESET引脚，因此该引脚既不能作为输入也不能作为外部复位
2. ICE_DAT与P1.6共享一个引脚，因此该引脚I/O功能或其他功能都不能使用
3. ICE_CLK与P0.2共享一个引脚，因此该引脚I/O功能或其他功能都不能使用
- 4.当系统处在空闲或掉电模式时，因为部分外设时钟已经停止，所以任何访问可能无效的。读访问可能返回一个无用的数据，写访问可能不成功。
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的指令将不起作用，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。非侵入命令可以用调试器访问MCU的存储单元、状态或控制寄存器，一个读或写控制寄存器必须在MCU停止的条件下进行，产生停止条件是在与硬件断点匹配后或单步运行后。

CONFIG0

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

出厂预设值: 1111 1111b

位	名称	描述
---	----	----

位	名称	描述
5	OCDPWM	片上调试OCD占用模式下，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。 备注：该位仅当PWM管脚的PIO 位设置为1后有效
4	OCDEN	OCD 使能 1 = 禁用OCD 0 = 使能OCD 注意：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 (AUXR1.5) 标志位置位。

30 在线电路编程 (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。采用ICP方式将很简单,且不需要将微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 nRESET、ICPDA及 ICPCK。nRESET用于进入或退出ICP模式,ICPDA为数据输入输出脚,ICPCK为编程时钟输入脚。用户需要在系统板上预留VDD、GND以及这三个脚。

新唐提供MS51的ICP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Nuvoton根据MCU的电气特性专门设计的,是很高效稳定的编程方式。

31 指令集

31.1 指令集概述

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。

Rn (N = 0~7)	Register R0 To R7 Of The Currently Selected Register Bank.
Direct	8-bit internal data location's address. It could be an internal data RAM location (00H to 7FH) or an SFR (80H to FFH).
@RI (I = 0, 1)	8-bit internal data RAM location (00H to FFH) addressed indirectly through register R0 or R1.
#data	8-bit constant included in the instruction.
#data16	16-bit constant included in the instruction.
Addr16	16-bit destination address. Used by LCALL and LJMP. A branch can be any-where within the Program Memory address space.
Addr11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2K-Byte page of Program Memory as the first byte of the following instruction.
Rel	Signed (2's complement) 8-bit offset Byte. Used by SJMP and all conditional branches. The range is -128 to +127 bytes relative to first byte of the following instruction.
Bit	Direct addressed bit in internal data RAM or SFR.

表 31.1- 1 所有指令， 指令集及寻址模式注释

31.2 指令集表

指令	OPCODE	字节	时钟周期	MS51 对比传统80C51速率
NOP	00	1	1	12
ADD A, Rn	28~2F	1	2	6
ADD A, direct	25	2	3	4
ADD A, @Ri	26, 27	1	4	3
ADD A, #data	24	2	2	6
ADDC A, Rn	38~3F	1	2	6
ADDC A, direct	35	2	3	4
ADDC A, @Ri	36, 37	1	4	3
ADDC A, #data	34	2	2	6
SUBB A, Rn	98~9F	1	2	6
SUBB A, direct	95	2	3	4
SUBB A, @Ri	96, 97	1	4	3
SUBB A, #data	94	2	2	6
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
XRL A, #data	64	2	2	6
XRL direct, A	62	2	4	3
XRL direct, #data	63	3	4	6
CLR A	E4	1	1	12
CPL A	F4	1	1	12
RL A	23	1	1	12

指令	OPCODE	字节	时钟周期	MS51 对比传统80C51速率
RLC A	33	1	1	12
RR A	03	1	1	12
RRC A	13	1	1	12
SWAP A	C4	1	1	12
MOV A, Rn	E8~EF	1	1	12
MOV A, direct	E5	2	3	4
MOV A, @Ri	E6, E7	1	4	3
MOV A, #data	74	2	2	6
MOV Rn, A	F8~FF	1	1	12
MOV Rn, direct	A8~AF	2	4	6
MOV Rn, #data	78~7F	2	2	6
MOV direct, A	F5	2	2	6
MOV direct, Rn	88~8F	2	3	8
MOV direct, direct	85	3	4	6
MOV direct, @Ri	86, 87	2	5	4.8
MOV direct, #data	75	3	3	8
MOV @Ri, A	F6, F7	1	3	4
MOV @Ri, direct	A6, A7	2	4	6
MOV @Ri, #data	76, 77	2	3	6
MOV DPTR, #data16	90	3	3	8
MOVC A, @A+DPTR	93	1	4	6
MOVC A, @A+PC	83	1	4	6
MOVX A, @Ri ^[1]	E2, E3	1	5	4.8
MOVX A, @DPTR ^[1]	E0	1	4	6
MOVX @Ri, A ^[1]	F2, F3	1	6	4
MOVX @DPTR, A ^[1]	F0	1	5	4.8
PUSH direct	C0	2	4	6
POP direct	D0	2	3	8
XCH A, Rn	C8~CF	1	2	6
XCH A, direct	C5	2	3	4
XCH A, @Ri	C6, C7	1	4	3
XCHD A, @Ri	D6, D7	1	5	2.4
CLR C	C3	1	1	12
CLR bit	C2	2	4	3
SETB C	D3	1	1	12
SETB bit	D2	2	4	3
CPL C	B3	1	1	12
CPL bit	B2	2	4	3
ANL C, bit	82	2	3	8
ANL C, /bit	B0	2	3	8
ORL C, bit	72	2	3	8
ORL C, /bit	A0	2	3	8
MOV C, bit	A2	2	3	4
MOV bit, C	92	2	4	6
ACALL addr11	11, 31, 51, 71, 91, B1, D1, F1 ^[2]	2	4	6
LCALL addr16	12	3	4	6
RET	22	1	5	4.8

指令	OPCODE	字节	时钟周期	MS51 对比传统80C51速率
RETI	32	1	5	4.8
AJMP addr11	01, 21, 41, 61, 81, A1, C1, E1 ^[3]	2	3	8
LJMP addr16	02	3	4	6
SJMP rel	80	2	3	8
JMP @A+DPTR	73	1	3	8
JZ rel	60	2	3	8
JNZ rel	70	2	3	8
JC rel	40	2	3	8
JNC rel	50	2	3	8
JB bit, rel	20	3	5	4.8
JNB bit, rel	30	3	5	4.8
JBC bit, rel	10	3	5	4.8
CJNE A, direct, rel	B5	3	5	4.8
CJNE A, #data, rel	B4	3	4	6
CJNE Rn, #data, rel	B8~BF	3	4	6
CJNE @Ri, #data, rel	B6, B7	3	6	4
DJNZ Rn, rel	D8~DF	2	4	6
DJNZ direct, rel	D5	3	5	4.8
注 1. MS51 没有外部存储器总线结构。MOVX指令仅用于读写内部XRAM用 2.11位地址[A10:A8]的最高三位决定ACALL hex 码。代码为[A10,A9,A8,1,0,0,0,1] 3.11位地址[A10:A8]的最高三位决定AJMP hex 码。代码为 [A10,A9,A8,0,0,0,0,1]				

表 31.2- 1 指令集

32 配置字

MS51 具有几个硬件配置字节，设置这些配置位配置硬件选项，如安全位，系统时钟位等等。这些硬件配置位可通过编程器/烧录器或ICP、IAP来配置。有些特定的配置位定义的功能，也可以通过特殊寄存器位重新配置。因此，需要加载这些配置位到相应的寄存器位。这些加载发生在复位之后。软件复位会加载部分配置位至特殊功能寄存器相应的位，这些寄存器位也可以通过用户的软件控制及修改。其他复位将不改变这些寄存器位的值

32.1 CONFIG0

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

出厂预设值: 1111 1111b

位	名称	描述
7	CBS	配置启动选择位 该位定义，除软件复位外的所有复位后，MCU选择FLASH区启动。 1 = 除软件复位外的所有复位后，MCU从APROM启动 0 = 除软件复位外的所有复位后，MCU从LDROM启动
5	OCDPWM	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。
4	OCDEN	OCD使能 1 = 禁用OCD 0 = 使能OCD 注意: 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HFIF标志位置位。
3	-	保留
2	RPD	复位脚禁用位 1 = P2.0/nRESET 复位功能使能，管脚用作外部复位脚 0 = P2.0/ nRESET 复位功能关闭，管脚用作输入脚 P2.0
1	LOCK	芯片加密使能位 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。 备注：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。

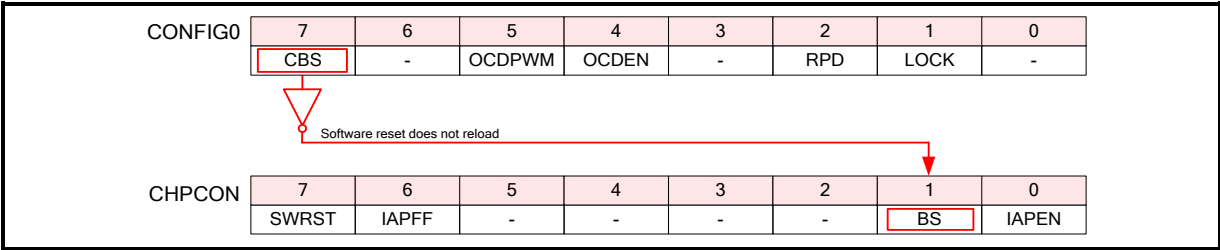


图 32.1- 1 CONFIG0 复位后自动重载

32.2 CONFIG1

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

出厂预设值: 1111 1111b

位	名称	描述
2:0	LDSIZE[2:0]	LDROM 区域选择 111 = 无 LDROM. APROM 16 KB. 110 = LDROM 1 KB. APROM 15 KB. 101 = LDROM 2 KB. APROM 14 KB. 100 = LDROM 3 KB. APROM 13 KB. 0xx = LDROM 4 KB. APROM 12 KB.

32.3 CONFIG2

CONFIG2

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

出厂预设值: 1111 1111b

位	名称	描述
7	CBODEN	配置欠压侦测使能位 1 = 欠压检测功能打开 0 = 欠压检测功能关闭
6:4	CBOV[2:0]	配置欠压侦测电压选择位 111 = VBOD 生效电压 1.8V. 110 = VBOD 生效电压 1.8V. 101 = VBOD 生效电压 2.0V. 100 = VBOD 生效电压 2.4V. 011 = VBOD 生效电压 2.7V. 010 = VBOD 生效电压 3.0V. 001 = VBOD 生效电压 3.7V. 000 = VBOD 生效电压 4.4V.
3	BOIAP	欠压禁止IAP 位 该位决定当系统电压低于欠压侦测设定值时，IAP擦除及编程功能是否禁止。该位仅当欠压检测功能使能后有效。 1 = 当VDD 低于VBOD ,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	CONFIG 欠压复位使能位 该为决定当侦测到电压低于VBOD 时，芯片是否复位 1 = 欠压复位功能使能 0 = 欠压复位功能关闭

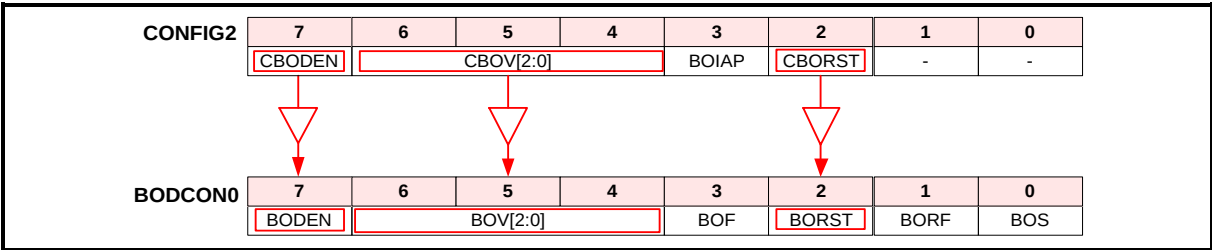


图 32.3- 1 CONFIG2 上电复位重载位

32.4 CONFIG4

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

出厂预设值: 1111 1111b

位	名称	描述
7:4	WDTEN[3:0]	看门狗定时器WDT 使能 该为决定复位后，看门狗定时器功能 1111 = 看门狗定时器关闭，WDT可用作普通定时器用 0101 = 使能WDT超时复位功能，且WDT在空闲或掉电模式下停止运行 其他 = 使能WDT超时复位功能，且WDT在空闲或掉电模式下保持运行
3:0	-	保留

33 封装信息

33.1 TSSOP 14-pin (4.4 x 5.5 x 1.2 mm)

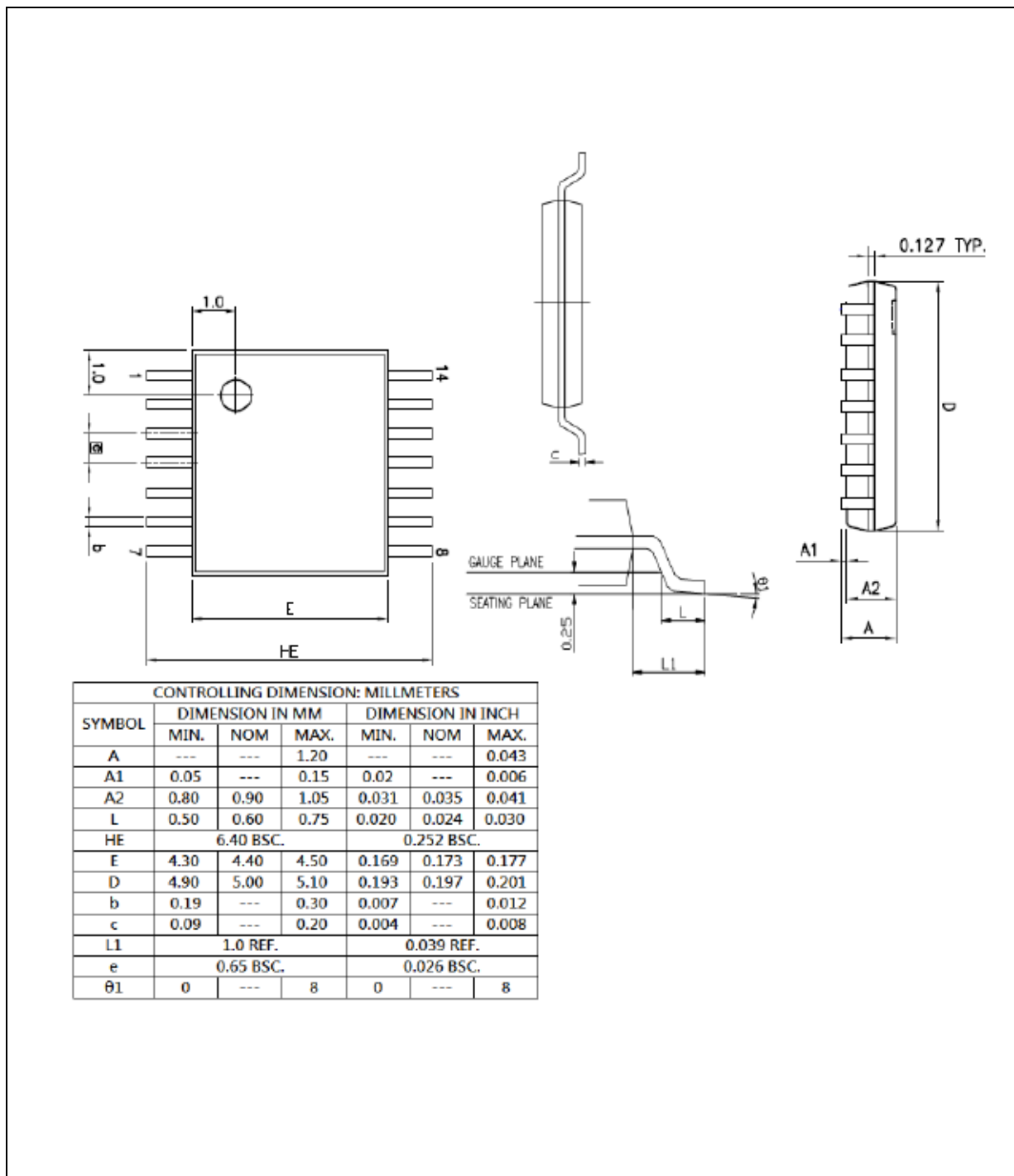


图 33.1-1 TSSOP14 封装信息

33.2 MSOP 10-pin (3.0 x 3.0 x 1.1 mm)

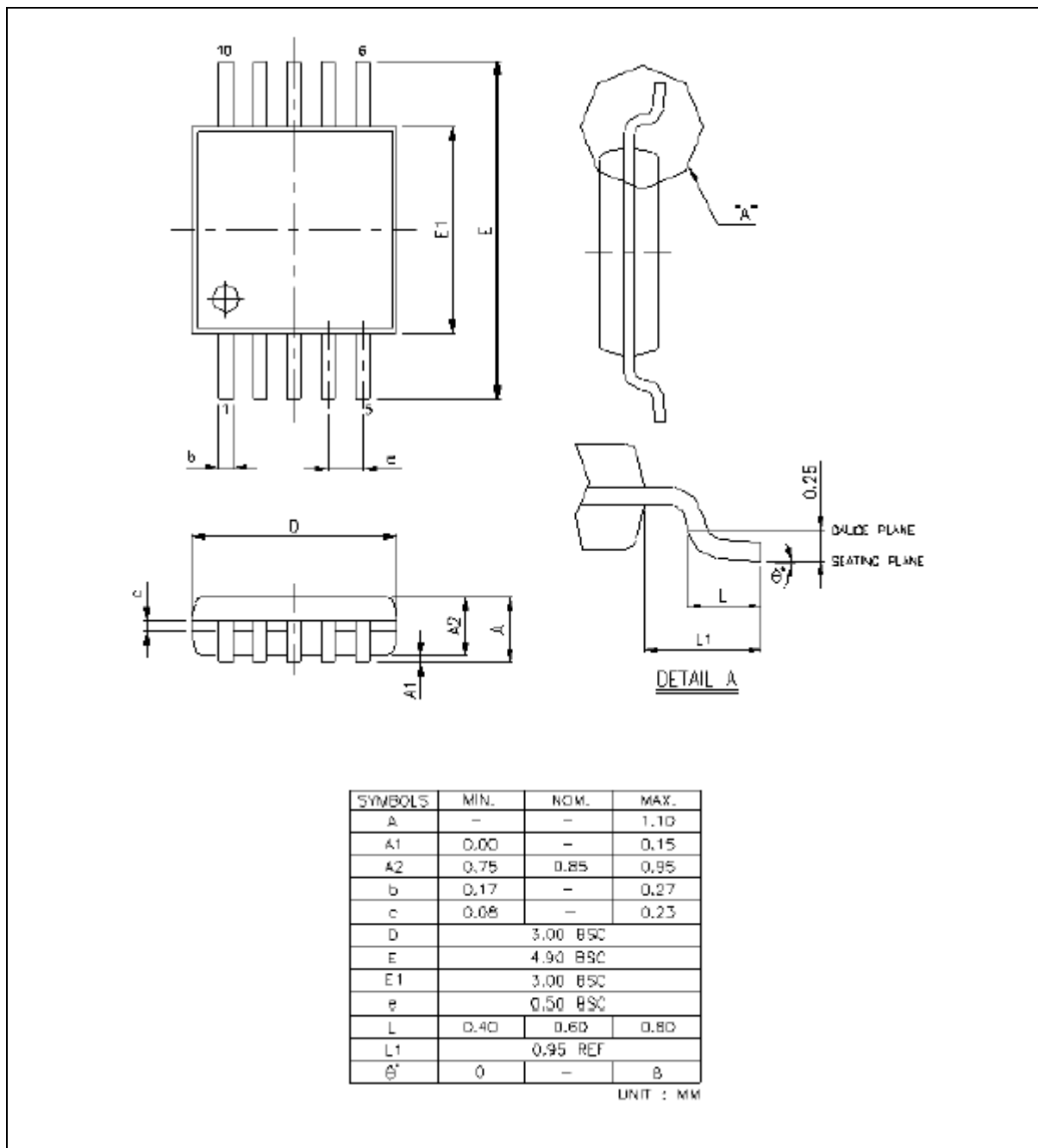


图 33.2- 1 MSOP10 封装信息

34 历史版本信息

日期	版本	描述
2019.6.19	1.00	初版发布

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.