



CMS8S5885用户手册

增强型闪存8位1T 8051单片机

Rev. 1.03

请注意以下有关CMS知识产权政策

* 中微半导体（深圳）股份有限公司（以下简称本公司）已申请了专利，享有绝对的合法权益。与本公司MCU或其他产品有关的专利权并未被同意授权使用，任何经由不当手段侵害本公司专利权的公司、组织或个人，本公司将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨本公司因侵权行为所受的损失、或侵权者所得的不法利益。

* 中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

* 本公司保留对规格书中产品在可靠性、功能和设计方面的改进作进一步说明的权利。然而本公司对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，本公司不保证和不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本公司的产品不授权适用于救生、维生器件或系统中作为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考官方网站 www.mcu.com.cn。

目录

1. 产品概述	8
1.1 功能特性	8
1.2 系统结构框图	9
1.3 管脚分布	10
1.3.1 TSSOP20 引脚图	10
1.3.2 QFN20 引脚图	10
1.4 系统配置寄存器	13
1.5 在线串行编程	15
1.6 在线调试模式	15
2. 中央处理器 (CPU)	16
2.1 存储器	16
2.1.1 程序存储器	16
2.1.2 复位向量 (0000H)	18
2.1.3 数据存储器 (IRAM)	18
2.1.4 特殊功能寄存器表 (SFR)	20
2.1.5 外部数据存储器 (XRAM)	21
2.1.6 特殊功能寄存器表 (XSFR)	22
2.2 累加器 (ACC)	27
2.3 B 寄存器 (B)	27
2.4 堆栈指针寄存器 (SP)	27
2.5 数据指针寄存器 (DPTR0/DPTR1)	27
2.6 数据指针选择寄存器 (DPS)	27
2.7 程序状态寄存器 (PSW)	28
2.8 程序计数器 (PC)	29
2.9 时序存取寄存器 (TA)	29
2.10 看门狗计数器 (WDT)	30
2.10.1 WDT 溢出周期	30
2.10.2 看门狗控制寄存器 WDCON	30
2.10.3 WDT 溢出控制寄存器 CKCON	31
2.11 功能控制寄存器 (FUNCCR)	31
3. 系统时钟	32
3.1 系统振荡器	32
3.2 复位时间	32
3.3 时钟结构	33
3.3.1 系统时钟结构	33
3.3.2 振荡器控制寄存器 CLKDIV	34
3.3.3 功能时钟控制寄存器 CKCON	34
4. 复位	35
4.1 上电复位	35
4.2 外部复位	37
4.3 LVR 低压复位	37
4.4 看门狗复位 (WDT)	38
4.5 软件复位	39
4.6 CONFIG 状态保护复位	39

4.7	上电配置监控复位.....	39
5.	电源管理.....	40
5.1	电源管理寄存器 PCON.....	40
5.2	电源监控寄存器 LVDCON.....	41
5.3	IDLE 空闲模式.....	41
5.4	STOP 休眠模式.....	42
5.4.1	休眠模式功能唤醒.....	42
5.4.2	唤醒等待状态.....	42
5.4.3	休眠唤醒时间.....	42
5.4.4	复位操作重新启动系统.....	43
5.5	休眠模式应用举例.....	43
5.6	调试模式下的休眠功耗.....	43
6.	I/O 端口.....	44
6.1	GPIO 功能.....	44
6.1.1	PORTx 数据寄存器 Px.....	44
6.1.2	PORTx 方向寄存器 PxTRIS.....	44
6.1.3	PORTx 开漏控制寄存器 PxOD.....	45
6.1.4	PORTx 上拉电阻控制寄存器 PxUP.....	45
6.1.5	PORTx 下拉电阻控制寄存器 PxRD.....	45
6.1.6	PORTx 驱动电流控制寄存器 PxDR.....	45
6.1.7	PORTx 斜率控制寄存器 PxSR.....	46
6.1.8	PORTx 数据输入选择寄存器 PxDS.....	46
6.2	复用功能.....	47
6.2.1	端口复用功能配置寄存器.....	47
6.2.2	端口输入功能分配寄存器.....	50
6.2.3	端口外部中断控制寄存器.....	51
6.2.4	复用功能应用注意.....	51
7.	中断.....	52
7.1	中断概述.....	52
7.2	外部中断.....	53
7.2.1	INT0/INT1 中断.....	53
7.2.2	GPIO 中断.....	53
7.2.3	中断与休眠唤醒.....	53
7.3	中断寄存器.....	54
7.3.1	中断屏蔽寄存器.....	54
7.3.2	中断优先级控制寄存器.....	58
7.3.3	中断标志位寄存器.....	61
7.3.4	中断标志位的清除操作.....	67
7.3.5	调试模式下特殊中断标志位.....	68
8.	LSE 定时器.....	69
8.1	概述.....	69
8.2	寄存器.....	69
8.3	功能说明.....	70
8.4	中断与休眠唤醒.....	70
9.	定时计数器 0/1 (TIMER0/1).....	71

9.1	概述	71
9.2	定时器 0/1 寄存器	72
9.2.1	定时器 0/1 模式寄存器 TMOD	72
9.2.2	Timer0/1 控制寄存器 TCON, 可位寻址	73
9.2.3	定时器 0 数据寄存器低位 TL0	73
9.2.4	定时器 0 数据寄存器高位 TH0	74
9.2.5	定时器 1 数据寄存器低位 TL1	74
9.2.6	定时器 1 数据寄存器高位 TH1	74
9.2.7	功能时钟控制寄存器 CKCON	74
9.3	定时器 0/1 中断	75
9.4	定时器 0 工作模式	78
9.4.1	T0 -模式 0 (13 位定时/计数模式)	78
9.4.2	T0 -模式 1 (16 位定时/计数模式)	78
9.4.3	T0 -模式 2 (8 位自动重装定时/计数模式)	79
9.4.4	T0 -模式 3 (两个单独的 8 位定时/计数器)	80
9.5	定时器 1 工作模式	81
9.5.1	T1 -模式 0 (13 位定时/计数模式)	81
9.5.2	T1 -模式 1 (16 位定时/计数模式)	81
9.5.3	T1 -模式 2 (8 位自动重装定时/计数模式)	82
9.5.4	T1 -模式 3 (停止计数)	82
10.	定时计数器 2 (TIMER2)	83
10.1	概述	83
10.2	TIMER2 寄存器	84
10.2.1	T2 控制寄存器 T2CON	84
10.2.2	定时器 2 数据寄存器低位 TL2	85
10.2.3	定时器 2 数据寄存器高位 TH2	85
10.2.4	定时器 2 比较/捕获/自动重装寄存器低位 RLDL	85
10.2.5	定时器 2 比较/捕获/自动重装寄存器高位 RLDH	85
10.2.6	定时器 2 比较/捕获通道 1 寄存器低位 CCL1	85
10.2.7	定时器 2 比较/捕获通道 1 寄存器高位 CCH1	86
10.2.8	定时器 2 比较/捕获通道 2 寄存器低位 CCL2	86
10.2.9	定时器 2 比较/捕获通道 2 寄存器高位 CCH2	86
10.2.10	定时器 2 比较/捕获通道 3 寄存器低位 CCL3	86
10.2.11	定时器 2 比较/捕获通道 3 寄存器高位 CCH3	86
10.2.12	T2 比较捕获控制寄存器 CCEN	87
10.3	TIMER2 中断	88
10.3.1	定时中断	92
10.3.2	外部触发中断	92
10.3.3	比较中断	92
10.3.4	捕获中断	92
10.4	TIMER2 功能说明	93
10.4.1	T2 定时模式	93
10.4.2	T2 重装模式	93
10.4.3	Timer2 门控定时模式	93
10.4.4	Timer2 事件计数模式	93
10.5	比较功能/PWM 功能	94
10.5.1	比较模式 0	94

10.5.2 比较模式 1.....	95
10.6 捕获功能.....	96
10.6.1 捕获模式 0.....	96
10.6.2 捕获模式 1.....	97
11. 定时器 3/4 (TIMER3/4)	98
11.1 概述.....	98
11.2 定时器 3/4 寄存器.....	99
11.2.1 定时器 3/4 控制寄存器 T34MOD	99
11.2.2 定时器 3 数据寄存器低位 TL3	99
11.2.3 定时器 3 数据寄存器高位 TH3.....	100
11.2.4 定时器 4 数据寄存器低位 TL4	100
11.2.5 定时器 4 数据寄存器高位 TH4.....	100
11.3 定时器 3/4 中断.....	101
11.4 定时器 3 工作模式	103
11.4.1 T3 -模式 0 (13 位定时模式)	103
11.4.2 T3 -模式 1 (16 位定时模式)	103
11.4.3 T3 -模式 2 (8 位自动重装定时模式)	104
11.4.4 T3 -模式 3 (两个单独的 8 位定时器)	104
11.5 定时器 4 工作模式	105
11.5.1 T4 -模式 0 (13 位定时模式)	105
11.5.2 T4 -模式 1 (16 位定时模式)	105
11.5.3 T4- 模式 2 (8 位自动重装定时模式)	106
11.5.4 T4- 模式 3 (停止计数)	106
12. 蜂鸣器 (BUZZER)	107
13. 模数转换 (ADC)	108
13.1 ADC 概述.....	108
13.2 ADC 配置.....	109
13.2.1 端口配置.....	109
13.2.2 通道选择.....	109
13.2.3 ADC 参考电压	109
13.2.4 转换时钟.....	109
13.2.5 ADC 中断	110
13.2.6 结果格式化	110
13.3 ADC 硬件触发启动	110
13.3.1 外部端口边沿触发 ADC.....	110
13.3.2 PWM 触发 ADC.....	110
13.3.3 硬件触发启动前的延时	110
13.4 ADC 工作原理	111
13.4.1 启动转换.....	111
13.4.2 完成转换.....	111
13.4.3 终止转换.....	111
13.4.4 A/D 转换步骤.....	112
13.4.5 转换过程中进入休眠	112
13.5 ADC 相关寄存器	113
13.6 ADC 结果比较	118
14. 增强型 PWM 模块.....	119

14.1	概述	119
14.2	特性	119
14.3	端口配置	119
14.4	功能描述	120
14.4.1	加载更新模式	120
14.4.2	边沿对齐	121
14.4.3	中心对齐	122
14.4.4	互补模式与死区延时	125
14.4.5	刹车功能	125
14.5	PWM 相关寄存器	126
15.	I²C 模块	135
15.1	概述	135
15.2	I ² C 端口配置	136
15.3	I ² C 主控模式	137
15.3.1	I ² C 主控模式定时周期寄存器	137
15.3.2	I ² C 主控模式控制与状态寄存器	138
15.3.3	I ² C 从机地址寄存器	141
15.3.4	I ² C 主控模式发送与接收数据寄存器	141
15.4	I ² C 从动模式	142
15.4.1	I ² C 自有地址寄存器 I2CSADR	142
15.4.2	I ² C 从动模式控制与状态寄存器 I2CSCR/I2CSSR	142
15.4.3	I ² C 从动模式发送与接收缓冲寄存器 I2CSBUF	143
15.5	I ² C 中断	144
15.6	I ² C 从动模式传输方式	147
15.6.1	单次接收	147
15.6.2	单次发送	148
15.6.3	连续接收	149
15.6.4	连续发送	150
16.	BRT 模块	151
16.1	简介	151
16.2	功能描述	151
16.3	寄存器描述	152
17.	UARTN 模块 (N=0,1)	154
17.1	简介	154
17.2	UARTN 端口配置	154
17.3	UARTN 波特率	155
17.4	UARTN 寄存器	157
17.5	UARTN 中断	159
17.6	UARTN 模式 (N=0/1)	162
17.6.1	模式 0-同步模式	162
17.6.2	模式 1-8 位异步模式 (可变波特率)	162
17.6.3	模式 2-9 位异步模式 (固定波特率)	163
17.6.4	模式 3-9 位异步模式 (可变波特率)	163
18.	SPI 模块	164
18.1	SPI 简介	164

18.2	SPI 端口配置	165
18.3	SPI 硬件描述	166
18.4	SPI 相关寄存器	167
18.4.1	SPI 控制寄存器 SPCR	167
18.4.2	SPI 数据寄存器 SPDR	167
18.4.3	SPI 从器件选择控制寄存器 SSCR	168
18.4.4	SPI 状态寄存器 SPSR	168
18.5	SPI 主控模式	169
18.5.1	写冲突错误	170
18.6	SPI 从动模式	171
18.6.1	被寻址错误	171
18.6.2	写冲突错误	171
18.7	SPI 时钟控制逻辑	173
18.7.1	SPI 时钟相位与极性控制	173
18.7.2	SPI 传输格式	173
18.7.3	CPHA=0 传输格式	173
18.7.4	CPHA=1 传输格式	174
18.8	SPI 数据传输	175
18.8.1	SPI 传输启动	175
18.8.2	SPI 传输结束	175
18.9	SPI 时序图	176
18.9.1	主控模式传输	176
18.9.2	从动模式传输	176
18.10	SPI 中断	177
19.	FLASH 存储器的操作	180
20.	唯一 ID (UID)	183
20.1	概述	183
20.2	UID 寄存器说明	183
21.	休眠状态下定时唤醒	186
21.1	定时唤醒控制寄存器	186
21.2	定时唤醒原理	187
22.	指令	188
22.1	符号说明	188
22.2	指令一览表	189
23.	封装	192
23.1	TSSOP20	192
23.2	QFN20	193
24.	版本修订说明	194

1. 产品概述

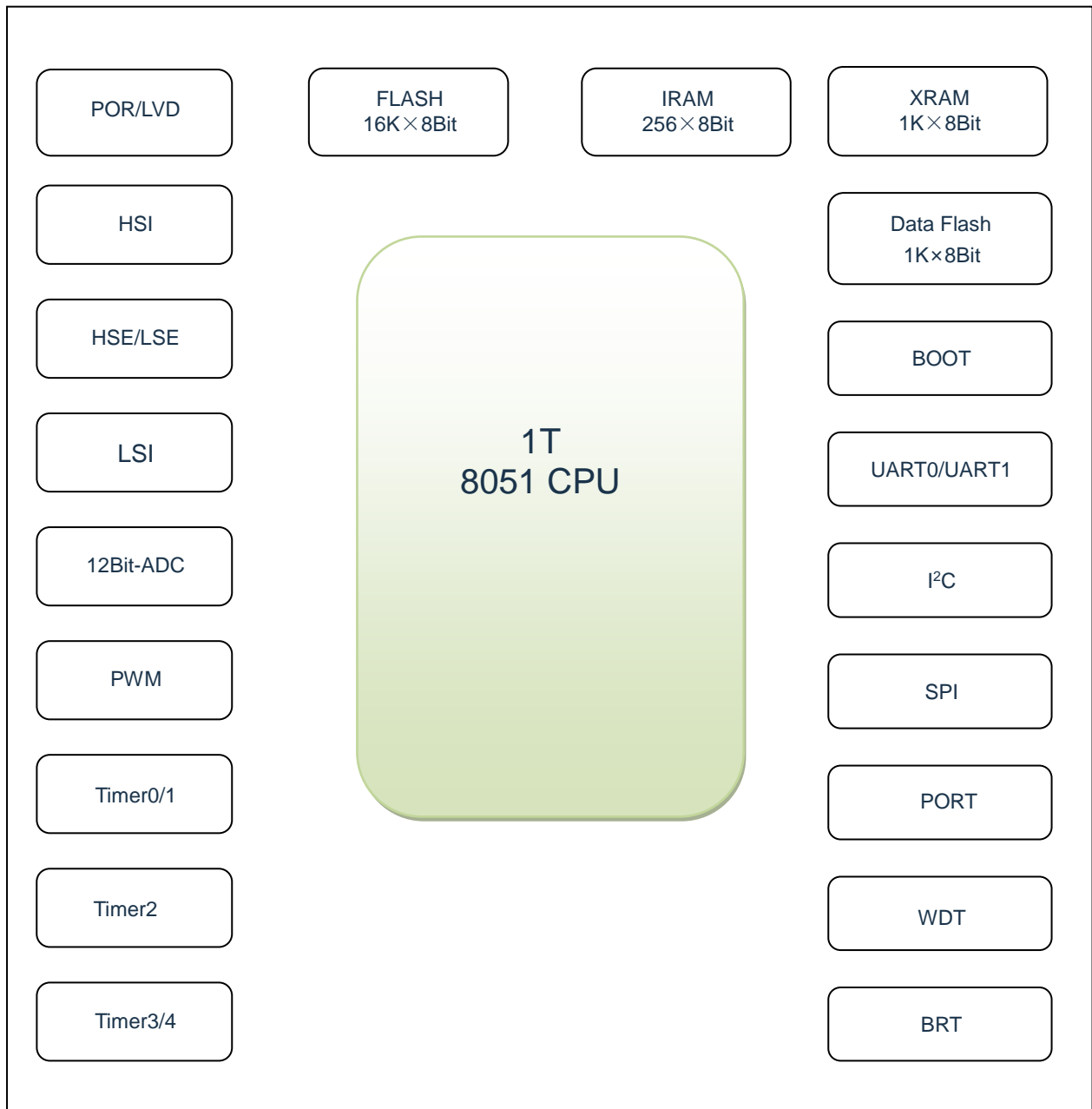
1.1 功能特性

- ◆ **兼容 MCS-51 的 1T 指令系统**
 - 系统时钟频率最高支持 $F_{sys}=F_{cpu}=24MHz$
 - 机器周期最快支持 $1T_{sys} @ F_{sys} \leq 24MHz$
- ◆ **内存**
 - 程序 FLASH: $16K \times 8Bit$ (APROM+BOOT)
 - Data FLASH: $1K \times 8Bit$
 - 通用 RAM: $256 \times 8Bit$
 - 通用 XRAM: $1K \times 8Bit$
 - 支持 BOOT 功能区, $1K/2K/4K$ 可选
 - 程序 FLASH 支持分区保护
- ◆ **4 种振荡方式**
 - HSI-内部 RC 振荡: 48MHz
 - HSE-外部晶体振荡: 8MHz/16MHz
 - LSE-外部晶体振荡: 32.768KHz
 - LSI-内部低功耗振荡: 125KHz
- ◆ **低压复位功能 (LVR)**
 - 1.8V/2.0V/2.5V/3.5V
- ◆ **低压检测功能 (LVD)**
 - 2.0V/2.2V/2.4V/2.7V/3.0V/3.7V/4.0V/4.3V
- ◆ **GPIO**
 - 最多可达 18 个 GPIO
 - 所有数字功能可分配到任意 GPIO
 - 均支持上/下拉电阻功能, 唤醒功能
 - 均支持边沿 (上升沿/下降沿/双沿) 中断/唤醒
- ◆ **中断源**
 - 支持所有的外部端口中断
 - 7 个定时器中断
 - 其它外设中断
- ◆ **定时器**
 - WDT 定时器 (看门狗定时器)
 - Timer0/1, Timer2, Timer3/4
 - LSE Timer (支持休眠唤醒功能)
 - WUT (专用唤醒定时器)
 - BRT (独立串口波特率时钟发生器)
- ◆ **工作电压范围**
 - 2.1V~5.5V
- ◆ **工作温度范围**
 - $-40^{\circ}C \sim 105^{\circ}C$
- ◆ **蜂鸣器驱动**
 - 50%占空比, 频率可自由设置
- ◆ **增强型 EPWM**
 - 6 通道增强型 PWM
 - 6 个相互独立周期计数器
 - 支持独立/互补/同步/成组模式
 - 支持边沿对齐/中心对齐方式
 - 支持互补模式死区延时功能
 - 支持掩码功能及刹车功能
- ◆ **高精度 12 位 ADC**
 - 所有 GPIO (18I/Os) 均支持 AD 通道
 - 参考电压可选 (1.2V/2.0V/2.4V/3.0V/VDD)
 - 可检测内部 1.2V 基准电压
 - 支持硬件触发启动转换功能
 - 支持一组结果数字比较功能
- ◆ **支持 96 位唯一 ID 号 (UID)**
 - 每颗芯片有独立的 ID 号
- ◆ **通信模块**
 - 1xSPI (通信速率最高可达 6Mb/s)
 - 1xI2C (通信速率最高可达 400Kb/s)
 - 2xUART (波特率最高可达 1Mb/s)
- ◆ **低功耗模式**
 - 空闲模式 (IDLE)
 - 休眠模式 (STOP, 功耗约 6uA)

型号说明:

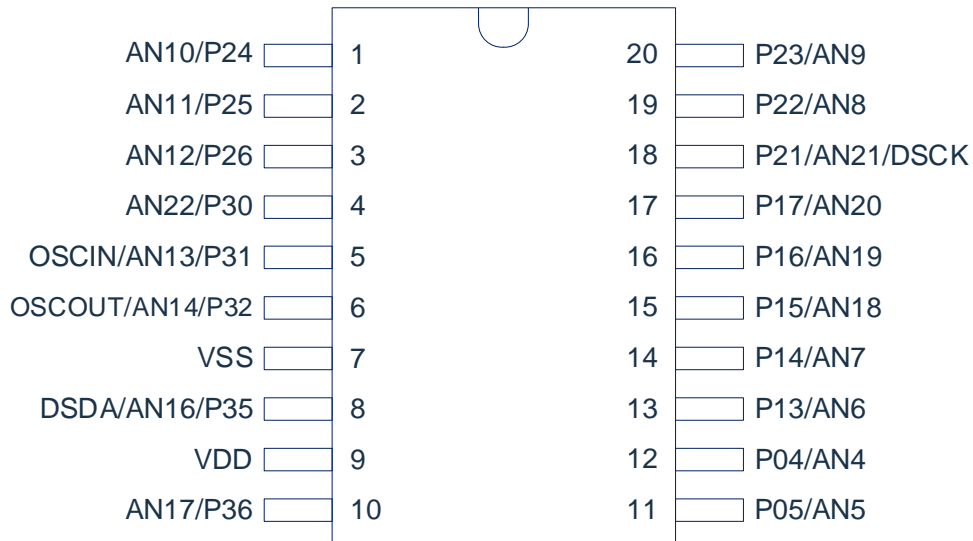
PRODUCT	FLASH	RAM	XRAM	DataFlash	I/O	UART	SPI	I2C	PWM	ADC	PACKAGE
CMS8S5885-TSSOP20	16KB	256B	1KB	1KB	18	2	1	1	6	12Bit ×18	TSSOP20
CMS8S5885-QFN20	16KB	256B	1KB	1KB	18	2	1	1	6	12Bit ×18	QFN20

1.2 系统结构框图

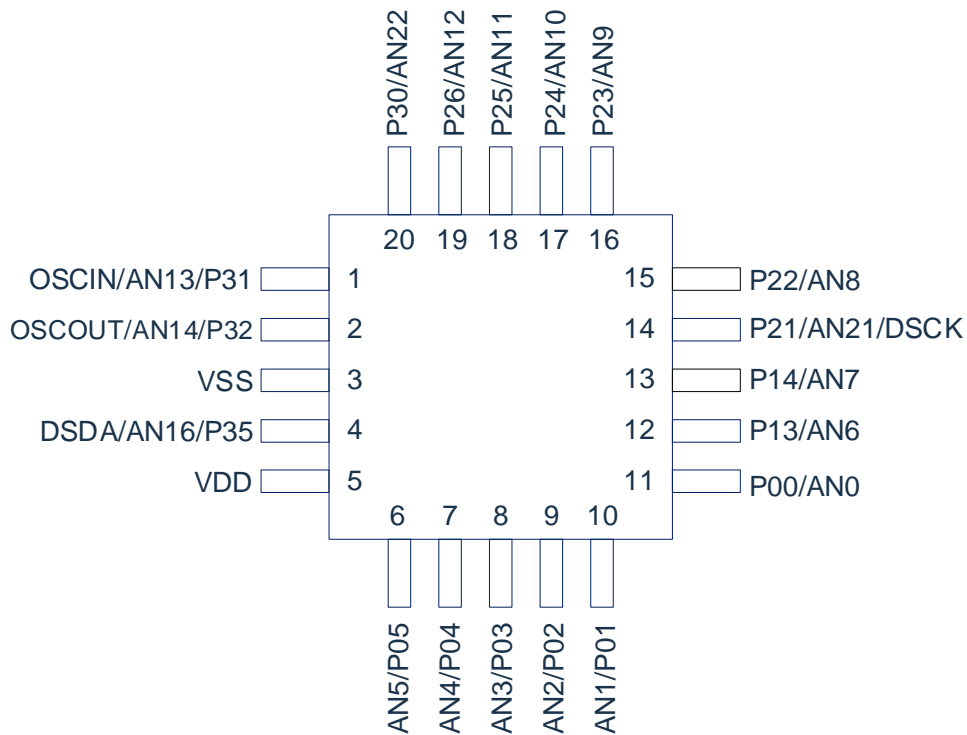


1.3 管脚分布

1.3.1 TSSOP20 引脚图



1.3.2 QFN20 引脚图



引脚说明:

引脚名称	功能	I/O 类型	说明
P00/AN0	P00	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN0	I	ADC 通道 0 输入
P01/AN1	P01	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN1	I	ADC 通道 1 输入
P02/AN2	P02	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN2	I	ADC 通道 2 输入
P03/AN3	P03	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN3	I	ADC 通道 3 输入
P04/AN4	P04	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN4	I	ADC 通道 4 输入
P05/AN5	P05	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN5	I	ADC 通道 5 输入
P13/AN6	P13	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN6	I	ADC 通道 6 输入
P14/AN7	P14	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN7	I	ADC 通道 7 输入
P15/AN18	P15	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN18	I	ADC 通道 18 输入
P16/AN19	P16	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN19	I	ADC 通道 19 输入
P17/AN20	P17	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN20	I	ADC 通道 20 输入
P21/AN21/DSCK	P21	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN21	I	ADC 通道 21 输入
	DSCK	I/O	编程、调试时钟输入输出
P22/AN8	P22	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN8	I	ADC 通道 8 输入
P23/AN9	P23	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN9	I	ADC 通道 9 输入
P24/AN10	P24	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN10	I	ADC 通道 10 输入
P25/AN11	P25	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN11	I	ADC 通道 11 输入
P26/AN12	P26	I/O	GPIO,通过寄存器配置输入输出, 上、下拉等功能
	AN12	I	ADC 通道 12 输入
P30/AN22	P30	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN22	I	ADC 通道 22 输入
P31/AN13/OSCIN	P31	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能

引脚名称	功能	I/O 类型	说明
	AN13	I	ADC 通道 13 输入
	OSCIN	I	外部振荡输入
P32/AN14/OSCOUT	P32	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN14	I	ADC 通道 14 输入
	OSCOUT	O	外部振荡输出
P35/AN16/DSDA	P35	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN16	I	ADC 通道 16 输入
	DSDA	I/O	编程、调试数据输入输出
P36/AN17	P36	I/O	GPIO 通过寄存器配置输入输出, 上、下拉等功能
	AN17	I	ADC 通道 17 输入
VDD		P	电源电压输入脚
VSS		P	接地脚

1.4 系统配置寄存器

系统配置寄存器（CONFIG）是 MCU 初始条件的 FLASH 选项。它只能被专用烧写器烧写，用户不能访问及操作。它包含了以下内容：

1. WDT（看门狗工作方式选择）
 - ENABLE 强制打开 WDT
 - SOFTWARE CONTROL(默认) WDT 工作方式由 WDCON 寄存器的 WDTRE 位控制
2. PROTECT
 - ENABLE FLASH 代码加密，读出的代码为 00H。且禁止进入调试模式
 - DISABLE（默认） FLASH 代码不加密
3. FLASH_DATA_PROTECT
 - DISABLE FLASH 数据区不加密
 - ENABLE（默认） FLASH 数据区加密，加密后烧写仿真器读出来的值为 00H
4. LVR（低压复位）
 - 1.8V（默认） ● 2.0V
 - 2.5V ● 3.5V
5. DEBUG（调试模式）
 - DISABLE（默认） 调试模式禁止，P21，P35 用做普通 IO 口
 - ENABLE 调试模式使能，P21，P35 配置成调试口（DSCK，DSDA），引脚对应的其他功能关闭
6. OSC（振荡方式）
 - HSI（默认） 48MHz
 - HSE OSCIN，OSCOOUT 配置成振荡口
 - LSE(32.768KHz) OSCIN，OSCOOUT 配置成振荡口
 - LSI(125KHz) 125KHz
7. OSC_PRESCALE（振荡输出预分频选择）
 - $F_{osc}/1$ （默认）
 - $F_{osc}/2$
 - $F_{osc}/4$
 - $F_{osc}/8$
8. HSI_FS（内部 RC 振荡器分频选择）
 - $F_{HSI}/2$ 24MHz
 - $F_{HSI}/3$ 16MHz
 - $F_{HSI}/6$ （默认） 8MHz
9. EXT_RESET（外部复位配置）
 - DISABLE（默认） 外部复位禁止
 - ENABLE 外部复位使能
 - ENABLE(OPEN PULLUP) 外部复位使能且打开复位口内部上拉电阻

10. EXT_RESETSEL (外部复位端口选择)

- P00
- P01
- P02
- P03
- P04
- P05
- P13
- P14
- P15
- P16
- P17
- P21
- P22
- P23
- P24
- P25
- P26
- P30
- P31
- P32
- P35
- P36

11. WAKE_UP_WAIT TIME (休眠唤醒等待振荡器稳定的时间默认为 1.0s)

- 50us
- 100us
- 500us
- 1ms
- 5ms
- 10ms
- 500ms
- 1.0s (默认)

12. CPU_WAITCLOCK (存储器等待时钟选择)

- 1*System Clock (1T) (默认)
- 2*System Clock (2T)
- 3*System Clock (3T)
- 4*System Clock (4T)
- 5*System Clock (5T)
- 6*System Clock (6T)
- 7*System Clock (7T)
- 8*System Clock (8T)

13. WRITE_PROTECT 程序分区保护 (可保护区间, 所有默认区间为不保护)

- 0-2K (0000H-07FFH 保护/不保护)
- 2-4K (0800H-0FFFH 保护/不保护)
- 4-6K (1000H-17FFH 保护/不保护)
- 6-8K (1800H-1FFFH 保护/不保护)
- 8-10K (2000H-27FFH 保护/不保护)
- 10-12K (2800H-2FFFH 保护/不保护)
- 12-14K (3000H-37FFH 保护/不保护)
- 14-16K (3800H-3FFFH 保护/不保护)

14. BOOT 空间选择

- BOOT_DIS (默认) BOOT 区禁止
- BOOT_1K BOOT 区空间为 1K
- BOOT_2K BOOT 区空间为 2K
- BOOT_4K BOOT 区空间为 4K

注:

- 1) 机器周期与存储器等待时钟选择 (CPU_WAITCLOCK) 相关: 机器周期= $T_{SYS}/CPU_WAITCLOCK$ 。
- 2) WRITE_PROTECT 程序分区保护在设置 0000H-07FFH 区间保护时, 同时对 Data Flash 生效。

1.5 在线串行编程

可在最终应用电路中对单片机进行串行编程。编程可以简单地通过以下 4 根线完成：

- 电源线
- 接地线
- 数据线
- 时钟线

这使用户可使用未编程的器件制造电路板，而仅在产品交付前才对单片机进行编程。从而可以将最新版本固件或者定制固件烧写到单片机中。

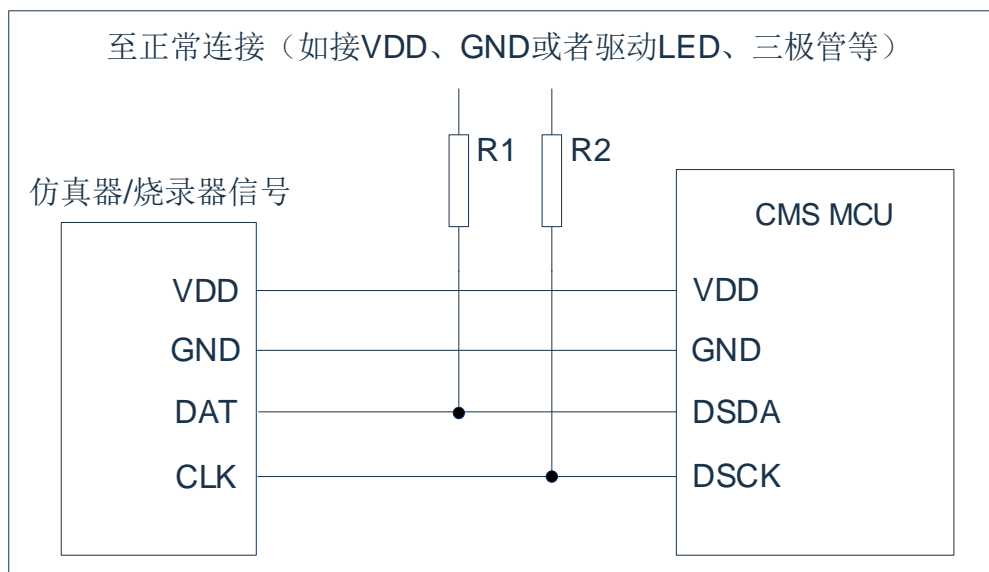


图 1-1：典型的在线串行编程连接方法

上图中，R1、R2 为电气隔离器件，常以电阻代替，其阻值如下： $R1 \geq 4.7K$ 、 $R2 \geq 4.7K$ 。

注意在编程和调试时，DSDA 禁止连接下拉电阻。如果实际电路需要接下拉电阻，建议利用跳线结构，在编程/调试时断开下拉电阻，完成之后再接入下拉电阻。

1.6 在线调试模式

芯片支持 2 线（DSCK，DSDA）在线调试功能，如使用该功能则需要将系统配置寄存器中的 DEBUG 设置为 ENABLE。使用调试模式时，需要注意以下几点：

- ◆ 在调试状态下，DSCK 与 DSDA 口作为专用调试口，不能实现其 GPIO 及复用功能。
- ◆ 在调试状态下，进入休眠模式/空闲模式（STOP/IDLE），系统电源与振荡器不会停止工作，在该状态下可仿真休眠唤醒功能。如需关注功耗，则建议关闭调试功能后再测试芯片实际的休眠电流。
- ◆ 在调试状态下暂停，其他功能外设继续运行，WDT，Timer0/1/2/3/4 计数器会停止，若 Timer1/4 作为 UART0/1 的波特率产生器，则暂停状态下 Timer1/4 也会继续运行。暂停状态下继续运行的外设可能会产生中断，调试时需要注意。

2. 中央处理器（CPU）

2.1 存储器

2.1.1 程序存储器



图 2-1：程序存储器空间

一个 16-Bit 程序计数器寻址最大可达 64K bytes，但是此芯片只有 16K bytes 程序存储空间。

上图所示为程序存储器较低区域，复位后 CPU 从 0000H 开始执行。每个中断在程序存储器中都分配有固定的地址，中断使 CPU 跳到该地址开始执行服务程序。例如外部中断 1，被分配地址为 0013H，如果使用外部中断 1，它的服务程序必须从 0013H 位置开始。如果没有用到该中断，其服务地址作为普通程序存储地址使用。

2.1.1.1 BOOT 分区

程序区空间大小为 16K*8Bit，其中程序区分为 BOOT 区和 APROM 区，BOOT 区大小由系统配置寄存器分配，详见第 1.4 章。BOOT 区和 APROM 区空间地址分配见下表 2.1 所示。

表 2.1 程序存储区空间地址分配

16K（程序存储区 0000H-3FFFFH）				
地址空间分配方式	APROM 区		BOOT 区	
方式 0	16K	0000H-3FFFFH	--	--
方式 1	15K	0000H-3BFFFH	1K	3C00H-3FFFFH
方式 2	14K	0000H-37FFFH	2K	3800H-3FFFFH
方式 3	12K	0000H-2FFFFH	4K	3000H-3FFFFH

芯片上电，若程序从 BOOT 区启动需满足：地址空间分配方式为 1/2/3（通过 CONFIG 设置 BOOT_1K/BOOT_2K/BOOT_4K），否则程序将从 APROM 区启动。

以 BOOT 区 1K 空间举例：CONFIG 配置 BOOT_1K,，芯片上电配置后，程序从地址 3C00 开始运行。若程序需要在 BOOT 区和 APROM 区相互切换，需向 BOOT 区控制寄存器 BOOTCON 写入 0xAA/0x55（详见寄存器描述），之后执行软件复位或者产生看门狗复位。

上电复位、外部复位、电压复位时，BOOTCON 复位值为 0x00，软件复位和看门狗复位不能清除该寄存器。

BOOT 控制寄存器 (BOOTCON)

F691H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BOOTCON	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7-Bit0 D7-D0 BOOT 区控制位（该寄存器只在芯片配置为 BOOT_1K/BOOT_2K/BOOT_4K 时，才能写操作）
 0x55= 若从 APROM 区切换至 BOOT 区，需向其写入 0x55,再执行软件复位或者产生看门狗复位
 0xAA= 若从 BOOT 区切换至 APROM 区，需向其写入 0xAA,再执行软件复位或者产生看门狗复位
 其它值= 无效

例如：芯片上电从 BOOT 区启动后，使用软件复位方式，切换到 APROM 区，配置如下：

1) BOOT_ID 寄存器需要写 AAH

```
MOV DPTR, #BOOT_ID
MOV A, #0AAH
MOVX @DPTR, A
```

2) 执行软件复位

```
MOV TA, #0AAH
MOV TA, #055H
MOV WDCON, #080H
```

例如：使用软件复位方式，再从 APROM 区切换到 BOOT 区，配置如下：

1) BOOT_ID 寄存器需要写 55H

```
MOV DPTR, #BOOT_ID
MOV A, #055H
MOVX @DPTR, A
```

2) 执行软件复位

```
MOV TA, #0AAH
MOV TA, #055H
MOV WDCON, #080H
```

注：在BOOT功能有效时，APROM的中程序需要保证PC不会溢出（溢出是指PC超出APROM的地址范围），如果出现PC溢出的情况，则系统有可能运行异常。

2.1.2 复位向量 (0000H)

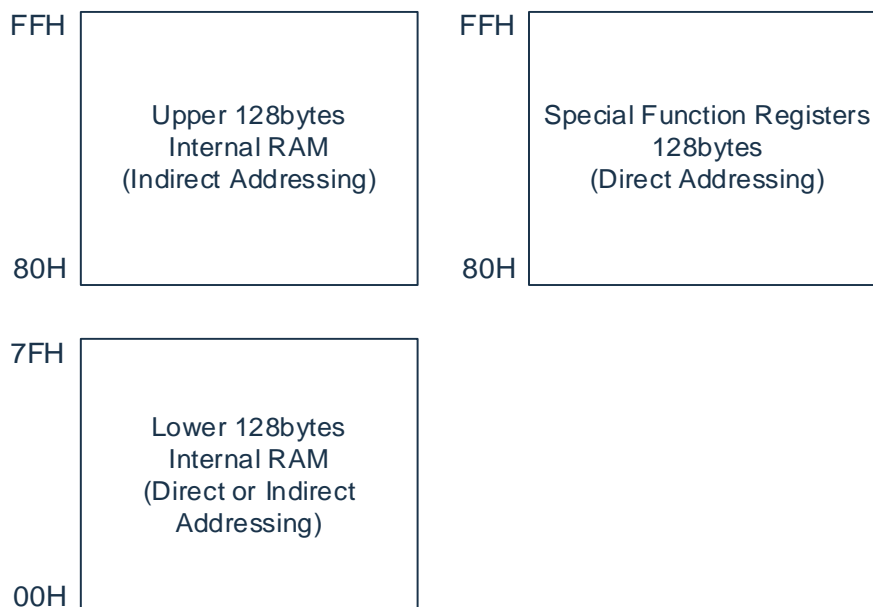
单片机具有一个字长的系统复位向量 (0000H)。

发生复位后，程序将从 0000H 处重新开始执行，系统寄存器也都将恢复为默认值。下面一段程序演示了如何定义 FLASH 中的复位向量。

例：定义复位向量

	ORG	0000H	;系统复位向量
	LJMP	START	
	ORG	0010H	;用户程序起始
START:			
	...		;用户程序
	...		
	END		;程序结束

2.1.3 数据存储 (IRAM)



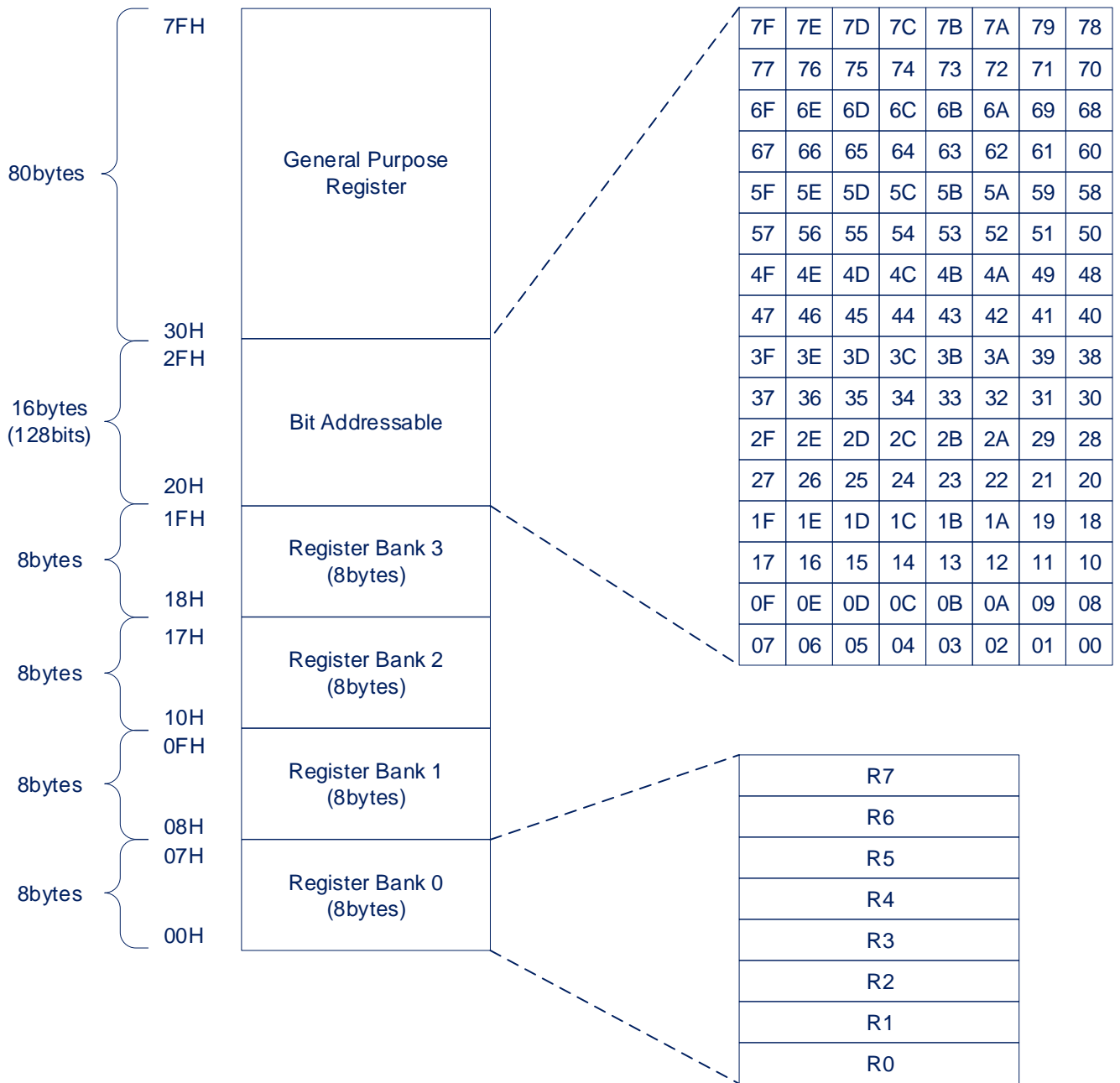
内部数据存储分为 3 个部分：低 128bytes、高 128bytes、SFR。

内部数据存储地址是一个字节宽度，这就意味着地址空间只有 256 bytes。但是，事实上通过内部 RAM 寻址方式调节可提供 384bytes。直接寻址高于 7FH 的存储空间和间接寻址高于 7FH 的存储空间进入到不同的存储空间，上图所示高 128bytes 和 SFR 占用相同的区域，80H 到 FFH，但他们本身却是独立的。

上图显示的是所有 8051 系列的低 128 字节 RAM。最低的 32 字节组成了 4 个寄存器组，程序指令可以调用从 R0 到 R7 的寄存器，程序状态字中的两位决定选用哪个寄存器组。这样可以更有效的使用代码空间，因为寄存器指令要比直接寻址指令更短。

寄存器组后边的 16bytes 组成了可位寻址的存储空间。8051 指令设置包括了位指令宽度的操作，但是这个区域的 128bits 可以通过这些指令直接寻址，该区域的地址是 20H 到 2FH。

通用寄存器低 128 字节的所有字节均可以直接访问或间接访问，高 128 字节只能间接寻址访问，这些区域用作数据 RAM 和堆栈。



2.1.4 特殊功能寄存器表（SFR）

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	--	--	--	MLOCK	MADRL	MADRH	MDATA	MCTRL
0xF0	B	I2CSADR	I2CSCR	I2CSBUF	I2CMSA	I2CMCR	I2CMBUF	I2CMTP
0xE8	--	ADCON2	SCON1	SBUF1	SPCR	SPSR	SPDR	SSCR
0xE0	ACC	--	TL4	TH4	--	--	--	--
0xD8	--	--	TL3	TH3	ADRESL	ADRESH	ADCON1	ADCON0
0xD0	PSW	ADCMPC	T34MOD	ADDLYL	ADCMPL	ADCMPLH	--	--
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2	CCEN	T2IE
0xC0	--	--	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3
0xB8	IP	EIP1	EIP2	--	WUTCRL	WUTCRLH	BUZDIV	BUZCON
0xB0	P3	--	EIF2	--	P0EXTIF	P1EXTIF	P2EXTIF	P3EXTIF
0xA8	IE	--	EIE2	--	P0EXTIE	P1EXTIE	P2EXTIE	P3EXTIE
0xA0	P2	P1TRIS	P2TRIS	P3TRIS	--	--	--	--
0x98	SCON0	SBUF	P0TRIS	--	--	--	--	--
0x90	P1	FUNCCR	--	DPX0	--	DPX1	TA	WDCON
0x88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	CLKDIV
0x80	P0	SP	DPL0	DPH0	DPL1	DPH1	DPS	PCON

说明：SFR 地址的低四位为 0000 或 1000 可进行位寻址，如 P0、TCON、P1。

表格中标注为“-”的地址禁止访问。

2.1.5 外部数据存储 (XRAM)

芯片内部有 1K bytes XRAM 该区域与 RAM/FLASH 没有联系，可通过 8 位的 SFR 进行读写。



XRAM/XSFR 的操作：

通过 DPTR 数据指针操作，DPTR 包括两组指针：DPTR0，DPTR1，通过 DPS 寄存器选择。

每组指针包括两个 8 位寄存器：DPTR0={DPH0,DPL0}；DPTR1={DPH1,DPL1}；

汇编代码如下：

MOV	DPTR,#0001H	
MOV	A,#5AH	
MOVX	@DPTR,A	;将 A 中的数据写入 XRAM 地址 0001H 中

通过 MOVX 间接寻址操作，汇编代码如下：

MOV	R0,#01H	
MOV	A,#5AH	
MOVX	@R0,A	;将 A 中的数据写入 XRAM 地址 01H 中，高 8 位地址由 DPH0/1 决定

在 Keil51 中将 Target-->Memory Model 设置为 Large 后，C 编译器将采用 XRAM 作为变量地址。一般用 DPTR 进行 XRAM/XSFR 的操作。

2.1.6 特殊功能寄存器表 (XSFR)

XSFR 是寻址空间与 XRAM 共用的特殊寄存器，主要包括：端口控制寄存器、其他功能控制寄存器。其寻址范围如下：



XSFR 列表：

地址	符号	说明
F000H	P00CFG	P0.0 口配置寄存器
F001H	P01CFG	P0.1 口配置寄存器
F002H	P02CFG	P0.2 口配置寄存器
F003H	P03CFG	P0.3 口配置寄存器
F004H	P04CFG	P0.4 口配置寄存器
F005H	P05CFG	P0.5 口配置寄存器
F006H	--	--
F007H	--	--
F009H	P0OD	P0 口开漏控制寄存器
F00AH	P0UP	P0 口上拉电阻控制寄存器
F00BH	P0RD	P0 口下拉电阻控制寄存器
F00CH	P0DR	P0 口驱动电流控制寄存器
F00DH	P0SR	P0 口斜率控制寄存器
F00EH	P0DS	P0 口数据输入选择寄存器
--	--	--
F013H	P13CFG	P1.3 口配置寄存器
F014H	P14CFG	P1.4 口配置寄存器
F015H	P15CFG	P1.5 口配置寄存器
F016H	P16CFG	P1.6 口配置寄存器
F017H	P17CFG	P1.7 口配置寄存器
F019H	P1OD	P1 口开漏控制寄存器
F01AH	P1UP	P1 口上拉电阻控制寄存器
F01BH	P1RD	P1 口下拉电阻控制寄存器
F01CH	P1DR	P1 口驱动电流控制寄存器
F01DH	P1SR	P1 口斜率控制寄存器
F01EH	P1DS	P1 口数据输入选择寄存器
--	--	--
F020H	--	--
F021H	P21CFG	P2.1 口配置寄存器

地址	符号	说明
F022H	P22CFG	P2.2 口配置寄存器
F023H	P23CFG	P2.3 口配置寄存器
F024H	P24CFG	P2.4 口配置寄存器
F025H	P25CFG	P2.5 口配置寄存器
F026H	P26CFG	P2.6 口配置寄存器
F027H	--	--
F029H	P2OD	P2 口开漏控制寄存器
F02AH	P2UP	P2 口上拉电阻控制寄存器
F02BH	P2RD	P2 口下拉电阻控制寄存器
F02CH	P2DR	P2 口驱动电流控制寄存器
F02DH	P2SR	P2 口斜率控制寄存器
F02EH	P2DS	P2 口数据输入选择寄存器
--	--	--
F030H	P30CFG	P3.0 口配置寄存器
F031H	P31CFG	P3.1 口配置寄存器
F032H	P32CFG	P3.2 口配置寄存器
F033H	--	--
F034H	--	--
F035H	P35CFG	P3.5 口配置寄存器
F036H	P36CFG	P3.6 口配置寄存器
F037H	--	--
F039H	P3OD	P3 口开漏控制寄存器
F03AH	P3UP	P3 口上拉电阻控制寄存器
F03BH	P3RD	P3 口下拉电阻控制寄存器
F03CH	P3DR	P3 口驱动电流控制寄存器
F03DH	P3SR	P3 口斜率控制寄存器
F03EH	P3DS	P3 口数据输入选择寄存器
--	--	--
F080H	P00EICFG	P0.0 口中断控制寄存器
F081H	P01EICFG	P0.1 口中断控制寄存器
F082H	P02EICFG	P0.2 口中断控制寄存器
F083H	P03EICFG	P0.3 口中断控制寄存器
F084H	P04EICFG	P0.4 口中断控制寄存器
F085H	P05EICFG	P0.5 口中断控制寄存器
--	--	--
F08BH	P13EICFG	P1.3 口中断控制寄存器
F08CH	P14EICFG	P1.4 口中断控制寄存器
F08DH	P15EICFG	P1.5 口中断控制寄存器
F08EH	P16EICFG	P1.6 口中断控制寄存器
F08FH	P17EICFG	P1.7 口中断控制寄存器
F090H	--	--
F091H	P21EICFG	P2.1 口中断控制寄存器
F092H	P22EICFG	P2.2 口中断控制寄存器
F093H	P23EICFG	P2.3 口中断控制寄存器
F094H	P24EICFG	P2.4 口中断控制寄存器
F095H	P25EICFG	P2.5 口中断控制寄存器
F096H	P26EICFG	P2.6 口中断控制寄存器

地址	符号	说明
F097H	--	--
F098H	P30EICFG	P3.0 口中断控制寄存器
F099H	P31EICFG	P3.1 口中断控制寄存器
F09AH	P32EICFG	P3.2 口中断控制寄存器
F09BH	--	--
F09CH	--	--
F09DH	P35EICFG	P3.5 口中断控制寄存器
F09EH	P36EICFG	P3.6 口中断控制寄存器
F09FH	--	--
--	--	--
F0C0H	PS_INT0	外部中断 0 输入端口分配寄存器
F0C1H	PS_INT1	外部中断 1 输入端口分配寄存器
F0C2H	PS_T0	Timer0 外部时钟输入端口分配寄存器
F0C3H	PS_T0G	Timer0 门控输入端口分配寄存器
F0C4H	PS_T1	Timer1 外部时钟输入端口分配寄存器
F0C5H	PS_T1G	Timer1 门控输入端口分配寄存器
F0C6H	PS_T2	Timer2 外部事件或门控输入端口分配寄存器
F0C7H	PS_T2EX	Timer2 下降沿自动重装输入端口分配寄存器
F0C8H	PS_CAP0	Timer2 输入捕获通道 0 端口分配寄存器
F0C9H	PS_CAP1	Timer2 输入捕获通道 1 端口分配寄存器
F0CAH	PS_CAP2	Timer2 输入捕获通道 2 端口分配寄存器
F0CBH	PS_CAP3	Timer2 输入捕获通道 3 端口分配寄存器
F0CCH	PS_ADET	ADC 的外部触发输入端口分配寄存器
F0CDH	PS_FB	PWM 的外部刹车信号端口分配寄存器
--	--	--
F120H	PWMCON	PWM 控制寄存器
F121H	PWMOE	PWM 输出使能寄存器
F122H	PWMPINV	PWM 输出极性选择寄存器
F123H	PWM0PSC	PWM0/PWM1 预分频控制寄存器
F124H	PWM2PSC	PWM2/PWM3 预分频控制寄存器
F125H	PWM4PSC	PWM4/PWM5 预分频控制寄存器
F126H	PWMCNTE	PWM 计数启动控制寄存器
F127H	PWMCNTM	PWM 计数模式选择寄存器
F128H	PWMCNTCLR	PWM 计数器清零控制寄存器
F129H	PWMLoadEN	PWM 加载使能控制寄存器
F12AH	PWM0DIV	PWM0 分频控制寄存器
F12BH	PWM1DIV	PWM1 分频控制寄存器
F12CH	PWM2DIV	PWM2 分频控制寄存器
F12DH	PWM3DIV	PWM3 分频控制寄存器
F12EH	PWM4DIV	PWM4 分频控制寄存器
F12FH	PWM5DIV	PWM5 分频控制寄存器
F130H	PWMP0L	PWM0 周期数据寄存器低 8 位
F131H	PWMP0H	PWM0 周期数据寄存器高 8 位
F132H	PWMP1L	PWM1 周期数据寄存器低 8 位
F133H	PWMP1H	PWM1 周期数据寄存器高 8 位
F134H	PWMP2L	PWM2 周期数据寄存器低 8 位
F135H	PWMP2H	PWM2 周期数据寄存器高 8 位

地址	符号	说明
F136H	PWMP3L	PWM3 周期数据寄存器低 8 位
F137H	PWMP3H	PWM3 周期数据寄存器高 8 位
F138H	PWMP4L	PWM4 周期数据寄存器低 8 位
F139H	PWMP4H	PWM4 周期数据寄存器高 8 位
F13AH	PWMP5L	PWM5 周期数据寄存器低 8 位
F13BH	PWMP5H	PWM5 周期数据寄存器高 8 位
--	--	--
F140H	PWMD0L	PWM0 比较数据寄存器低 8 位
F141H	PWMD0H	PWM0 比较数据寄存器高 8 位
F142H	PWMD1L	PWM1 比较数据寄存器低 8 位
F143H	PWMD1H	PWM1 比较数据寄存器高 8 位
F144H	PWMD2L	PWM2 比较数据寄存器低 8 位
F145H	PWMD2H	PWM2 比较数据寄存器高 8 位
F146H	PWMD3L	PWM3 比较数据寄存器低 8 位
F147H	PWMD3H	PWM3 比较数据寄存器高 8 位
F148H	PWMD4L	PWM4 比较数据寄存器低 8 位
F149H	PWMD4H	PWM4 比较数据寄存器高 8 位
F14AH	PWMD5L	PWM5 比较数据寄存器低 8 位
F14BH	PWMD5H	PWM5 比较数据寄存器高 8 位
--	--	--
F150H	PWMDD0L	PWM0 非对称向下比较数据寄存器低 8 位
F151H	PWMDD0H	PWM0 非对称向下比较数据寄存器高 8 位
F152H	PWMDD1L	PWM1 非对称向下比较数据寄存器低 8 位
F153H	PWMDD1H	PWM1 非对称向下比较数据寄存器高 8 位
F154H	PWMDD2L	PWM2 非对称向下比较数据寄存器低 8 位
F155H	PWMDD2H	PWM2 非对称向下比较数据寄存器高 8 位
F156H	PWMDD3L	PWM3 非对称向下比较数据寄存器低 8 位
F157H	PWMDD3H	PWM3 非对称向下比较数据寄存器高 8 位
F158H	PWMDD4L	PWM4 非对称向下比较数据寄存器低 8 位
F159H	PWMDD4H	PWM4 非对称向下比较数据寄存器高 8 位
F15AH	PWMDD5L	PWM5 非对称向下比较数据寄存器低 8 位
F15BH	PWMDD5H	PWM5 非对称向下比较数据寄存器高 8 位
--	--	--
F160H	PWMDTE	PWM 可编程死区延时控制寄存器
F161H	PWM01DT	PWM0/PWM1 可编程死区延时时间寄存器
F162H	PWM23DT	PWM2/PWM3 可编程死区延时时间寄存器
F163H	PWM45DT	PWM4/PWM5 可编程死区延时时间寄存器
F164H	PWMMASKE	PWM 掩码使能控制寄存器
F165H	PWMMASKD	PWM 掩码数据寄存器
F166H	PWMFBKC	PWM 刹车控制寄存器
F167H	PWMFBKD	PWM 刹车数据寄存器
F168H	PWMPIE	PWM 周期中断使能寄存器
F169H	PWMZIE	PWM 零点中断使能寄存器
F16AH	PWMUIE	PWM 向上比较中断使能寄存器
F16BH	PWMDIE	PWM 向下比较中断使能寄存器
F16CH	PWMPIF	PWM 周期中断标志寄存器
F16DH	PWMZIF	PWM 零点中断标志寄存器

地址	符号	说明
F16EH	PWUIF	PWM 向上比较中断标志寄存器
F16FH	PWMDIF	PWM 向下比较中断标志寄存器
--	--	--
F5C0H	BRTCON	BRT 模块控制寄存器
F5C1H	BRTDL	BRT 定时器加载值低 8 位
F5C2H	BRTDH	BRT 定时器加载值高 8 位
--	--	--
F690H	LVDCON	电源监控寄存器
F691H	BOOTCON	BOOT 控制寄存器
F692H	ADCLDO	ADC 参考电压控制寄存器
--	--	-
F694H	LSECRL	LSE 定时器数据寄存器低 8 位
F695H	LSECRH	LSE 定时器数据寄存器高 8 位
F696H	LSECON	LSE 定时器控制寄存器
--	--	-
F5E0H	UID0	UID<7:0>
F5E1H	UID1	UID<16:8>
F5E2H	UID2	UID<23:17>
F5E3H	UID3	UID<31:24>
F5E4H	UID4	UID<39:32>
F5E5H	UID5	UID<47:40>
F5E6H	UID6	UID<55:48>
F5E7H	UID7	UID<63:56>
F5E8H	UID8	UID<71:64>
F5E9H	UID9	UID<79:72>
F5EAH	UID10	UID<87:80>
F5EBH	UID11	UID<95:88>

注：表格中标注为“--”的地址禁止访问。

2.2 累加器 (ACC)

ALU 是 8Bit 宽的算术逻辑单元, MCU 所有的数学、逻辑运算均通过它来完成。它可以对数据进行加、减、移位及逻辑运算; ALU 也控制状态位 (PSW 状态寄存器中), 用来表示运算结果的状态。

ACC 寄存器是一个 8Bit 的寄存器, ALU 的运算结果可以存放在此。

2.3 B 寄存器 (B)

B 寄存器在使用乘法和除法指令时使用。如不使用乘除法指令, 也可作为通用寄存器使用。

2.4 堆栈指针寄存器 (SP)

SP 寄存器指向堆栈的地址, 复位后默认值位 0x07, 意味着堆栈的区域从 IRAM 地址的 08H 开始。该 SP 的值可以修改, 如果将堆栈区域设置为 0C0 开始, 则系统复位后需要将 SP 的值设置为 0xBF。

影响 SP 的操作有: 指令 PUSH、LCALL、ACALL、POP、RET、RETI 以及进入中断。

PUSH 指令占用堆栈中一个字节, LCALL, ACALL 及中断占用堆栈中两个字节, POP 指令释放一个字节, RET/RETI 指令释放两个字节。

使用 PUSH 指令会将被操作的寄存器的当前值自动保存到 RAM 中。

2.5 数据指针寄存器 (DPTR0/DPTR1)

数据指针主要用在 MOVX, MOVC 指令中, 其作用是定位 XRAM 与 ROM 的地址。芯片内部有两个数据指针寄存器 DPTR0 与 DPTR1。

2.6 数据指针选择寄存器 (DPS)

数据指针选择寄存器 DPS

0x86	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPS	ID1	ID0	TSL	AU	--	--	--	SEL
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6 ID1-ID0: 自减/自加功能选择。

00= DPTR0加1或者 DPTR1加1;

01= DPTR0减1或者 DPTR1加1;

10= DPTR0加1或者 DPTR1减1;

11= DPTR0减1或者 DPTR1减1。

Bit5 TSL: 翻转选择使能;
1= 执行DPTR指令后, SEL位会自动翻转;
0= DPTR相关指令不影响SEL位。

Bit4 AU: 自加/减使能位;
1= 允许MOVX @DPTR或者MOVC @DPTR指令运行后, 执行自减/自加的操作 (由ID1-ID0决定)。
0= DPTR相关指令不影响SEL位。

Bit3~Bit1 --

Bit0 SEL: 数据指针选择位;
1= 选择DPTR1;
0= 选择DPTR0。

2.7 程序状态寄存器 (PSW)

程序状态寄存器 PSW

0xD0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	F0	RS1	RS0	OV	--	P
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	CY: 进位标志位; 1= 有进位; 0= 无进位。
Bit6	AC: 辅助进位标志位 (半进位标志位); 1= 有进位; 0= 无进位。
Bit5	F0: 通用标志位。
Bit4~Bit3	RS1-RS0: 工作寄存器BANK选择位; 00= 选择Bank0; 01= 选择Bank1; 10= 选择Bank2; 11= 选择Bank3。
Bit2	OV: 溢出标志位; 1= 算术或逻辑运算有溢出; 0= 算术或逻辑运算无溢出。
Bit0	P: 校验位; 1= 结果的最高位发生了进位。 0= 结果的最高位没有发生进位。

2.8 程序计数器 (PC)

程序计数器 (PC) 控制程序内存 FLASH 中的指令执行顺序, 它可以寻址整个 FLASH 的范围, 取得指令码后, 程序计数器 (PC) 会自动加一, 指向下一个指令码的地址。但如果执行跳转、条件跳转、向 PCL 赋值、子程序调用、初始化复位、中断、中断返回、子程序返回等操作时, PC 会加载与指令相关的地址而不是下一条指令的地址。

当遇到条件跳转指令且符合跳转条件时, 当前指令执行过程中读取的下一条指令将会被丢弃, 且会插入一个空指令操作周期, 随后才能取得正确的指令。反之, 就会顺序执行下一条指令。

2.9 时序存取寄存器 (TA)

时序存取寄存器 TA

0x96	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TA	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0

TA[7:0]: 时序存取控制位。

某些被保护的寄存器必须在之前对TA进行如下操作才能写入。

```
MOV TA, #0AAH
```

```
MOV TA, #055H
```

中间不能插入其他任何指令, 再次修改时需要重新执行此序列。

被保护的寄存器: WDCON, CLKDIV。

2.10 看门狗计数器（WDT）

看门狗定时器（Watch Dog TIMER）是一个片内定时器，WDT 计时溢出将产生复位。

2.10.1 WDT 溢出周期

详见第 4.4 章节《看门狗复位》。

2.10.2 看门狗控制寄存器 WDCON

看门狗定时器控制寄存器 WDCON

0x97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	SWRST	PORF	--	--	WDTIF	WDTRF	WDTRE	WDTCLR
R/W	R/W	R/W	--	--	R/W	R/W	R/W	R/W
复位值	0	1	--	--	0	0	0	0

Bit7	SWRST: 软件复位控制位； 1: 执行系统软件复位（复位后需写0清除）。 0: --
Bit6	PORF: 上电复位标志位； 1: 系统为上电复位（写0清除，不需要TA写时序）。 0: --
Bit5~Bit4	未用。
Bit3	WDTIF: WDT溢出中断标志位； 1= WDT溢出（写0清除）； 0= WDT没有溢出。
Bit2	WDTRF: WDT复位标志位； 1= 系统由WDT复位（写0清除）； 0= 系统不由WDT复位。
Bit1	WDTRE: WDT复位使能位； 1= 使能WDT复位CPU； 0= 禁止WDT复位CPU。
Bit0	WDTCLR: WDT计数器清除位； 1= 清除WDT计数器（硬件自动清零）； 0= 禁止WDT计数器（写0无效）。

注：

- 1) 如果 CONFIG 中 WDT 配置为：ENABLE，则 WDT 始终被使能，而与 WDTRE 控制位的状态无关。且 WDT 的溢出复位功能强制打开。
- 2) 如果 CONFIG 中 WDT 配置为：SOFTWARE CONTROL，则可以使用 WDTRE 控制位使能或禁止 WDT。

修改 WDCON 需要的指令序列（中间不能插入其他任何指令）：

```
MOV    TA,#0AAH
MOV    TA,#055H
ORL    WDCON,#01H
```

2.10.3 WDT 溢出控制寄存器 CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit5 WTS2-WTS0: WDT溢出时间选择位;

000= $2^{17} \cdot T_{sys}$;

001= $2^{18} \cdot T_{sys}$;

010= $2^{19} \cdot T_{sys}$;

011= $2^{20} \cdot T_{sys}$;

100= $2^{21} \cdot T_{sys}$;

101= $2^{22} \cdot T_{sys}$;

110= $2^{24} \cdot T_{sys}$;

111= $2^{26} \cdot T_{sys}$ 。

Bit4 T1M: Timer1 的时钟源选择位;

0= $F_{sys}/12$;

1= $F_{sys}/4$ 。

Bit3 T0M: Timer0 的时钟源选择位;

0= $F_{sys}/12$;

1= $F_{sys}/4$ 。

Bit2~Bit0 未用。

2.11 功能控制寄存器 (FUNCCR)

0x91	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR	--	--	--	--	UART1_CKS1	UART0_CKS1	UART1_CKS	UART0_CKS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit4 保留位: 必须为0。

Bit3 UART1_CKS1: UART1的定时器时钟源选择位高位, {UART1_CKS1, UART1_CKS}:

00= Timer1的溢出时钟;

01= Timer4的溢出时钟;

10= Timer2的溢出时钟;

11= BRT的溢出时钟;

Bit2 UART0_CKS1: UART0的定时器时钟源选择位高位, {UART0_CKS1, UART0_CKS}:

00= Timer1的溢出时钟;

01= Timer4的溢出时钟;

10= Timer2的溢出时钟;

11= BRT的溢出时钟;

Bit1 UART1_CKS: UART1的定时器时钟源选择位低位, 见UART1_CKS1说明;

Bit0 UART0_CKS: UART0的定时器时钟源选择位低位, 见UART1_CKS0说明;

3. 系统时钟

3.1 系统振荡器

芯片有 4 种振荡方式：

- ◆ 内部 RC 振荡（HSI 48MHz）；
- ◆ 外部高速晶体振荡（HSE 8/16MHz）；
- ◆ 外部低速晶体振荡（LSE32.768KHz）；
- ◆ 内部低功耗振荡（LSI 125KHz）。

芯片默认的振荡方式为内部 RC 振荡，可以在 CONFIG 修改为其他振荡器。

3.2 复位时间

复位时间（Reset Time）是指从芯片复位到芯片开始执行指令这段时间，其默认设计值约为 16ms。该时间包括振荡器起振时间，配置时间。

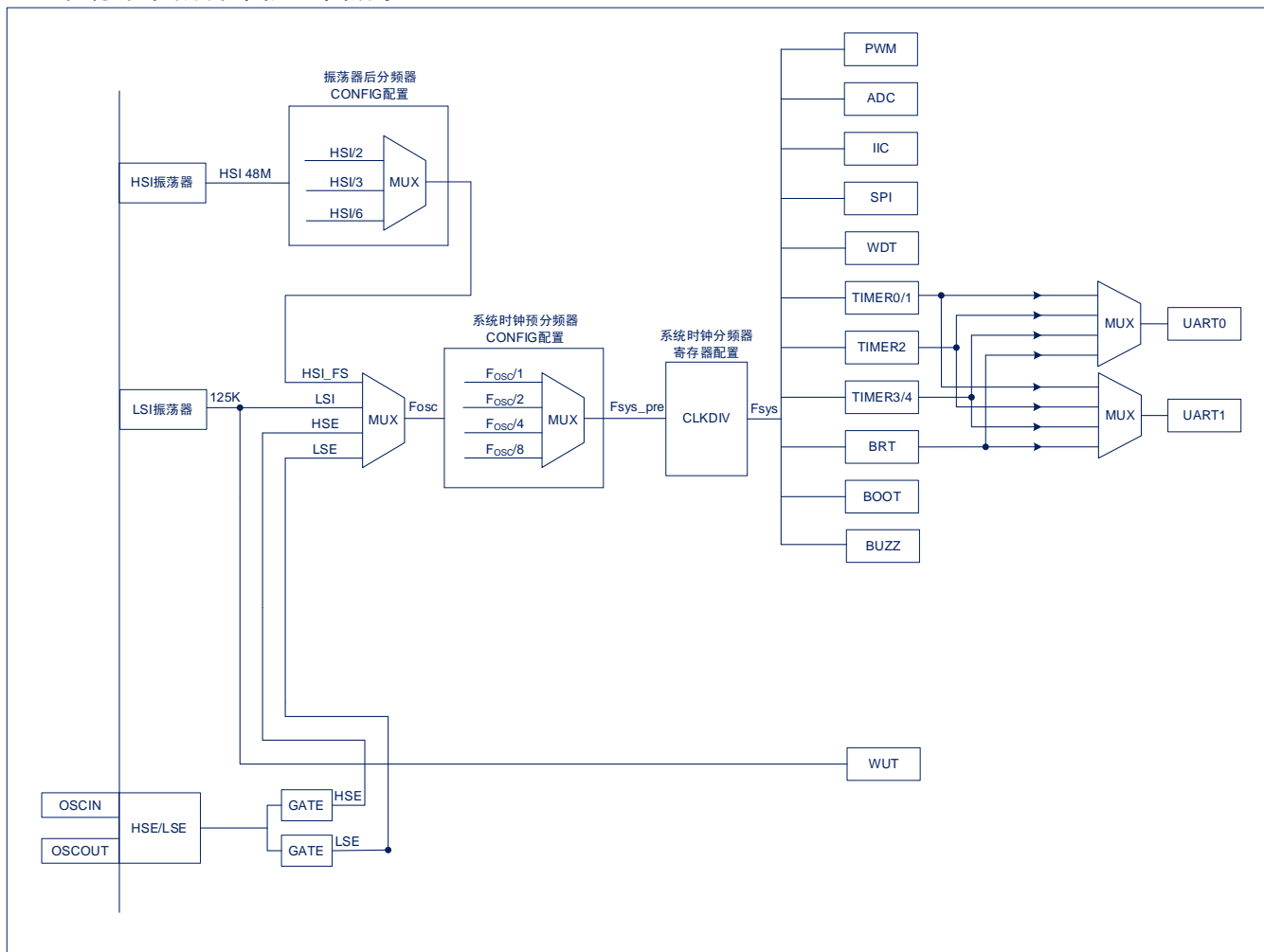
注：无论芯片是电源上电复位，还是其它原因引起的复位，都会存在这个复位时间。

另外振荡器选择为外部低速晶体振荡（32.768KHz）时，复位时间（包括起振时间）默认约为 1.5s（外接电容 10pF~22pF）。

3.3 时钟结构

3.3.1 系统时钟结构

系统时钟结构框图如下图所示：



注： CPU 机器周期对应运行频率为 F_{CPU} ，与存储器等待时钟相关，详见<系统配置寄存器>章节。

3.3.2 振荡器控制寄存器 CLKDIV

振荡器控制（CLKDIV）寄存器控制系统时钟（System Clock）和频率选择。

0x8F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKDIV	CLKDIV7	CLKDIV6	CLKDIV5	CLKDIV4	CLKDIV3	CLKDIV2	CLKDIV1	CLKDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 CLKDIV[7:0]，系统时钟 F_{sys} 分频位；

00H: F_{sys}=F_{sys_pre};

其他: F_{sys} =F_{sys_pre}/ (2*CLKDIV) (2,4...510 分频)。

修改 CLKDIV 需要的指令序列（中间不能插入其他任何指令）：

MOV	TA,#0AAH
MOV	TA,#055H
MOV	CLKDIV,#02H

3.3.3 功能时钟控制寄存器 CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit5 WTS2-WTS0: WDT溢出时间选择位；

000= 2^{17} *T_{sys};

001= 2^{18} *T_{sys};

010= 2^{19} *T_{sys};

011= 2^{20} *T_{sys};

100= 2^{21} *T_{sys}

101= 2^{22} *T_{sys};

110= 2^{24} *T_{sys};

111= 2^{26} *T_{sys}。

Bit4 T1M: Timer1 的时钟源选择位；

0= F_{sys}/12;

1= F_{sys}/4。

Bit3 T0M: Timer0 的时钟源选择位；

0= F_{sys}/12;

1= F_{sys}/4。

Bit2~Bit0 未用。

4. 复位

芯片可用如下几种复位方式：

- ◆ 上电复位；
- ◆ 外部复位；
- ◆ 低电压复位；
- ◆ 正常工作下的看门狗溢出复位；
- ◆ 内部 CONFIG 状态保护复位。

上述任意一种复位发生时，所有的系统寄存器将恢复默认状态，程序停止运行，同时程序计数器 PC 清零，复位结束后程序从复位向量 0000H 开始运行。

任何一种复位情况都需要一定的响应时间，系统提供完善的复位流程以保证复位动作的顺利进行。

4.1 上电复位

上电复位与 LVR 操作密切相关。系统上电的过程呈逐渐上升的曲线形式，需要一定时间才能达到正常电平值。下面给出上电复位的正常时序：

- 上电：系统检测到电源电压上升并等待其稳定；
- 系统初始化：所有的系统寄存器被置为初始值；
- 振荡器开始工作：振荡器开始提供系统时钟；
- 执行程序：上电结束，程序开始运行。

稳定时间（Stabilization Time）默认为 16ms，若配置选择 32.768KHz 晶振，则稳定时间约为 1.5s。

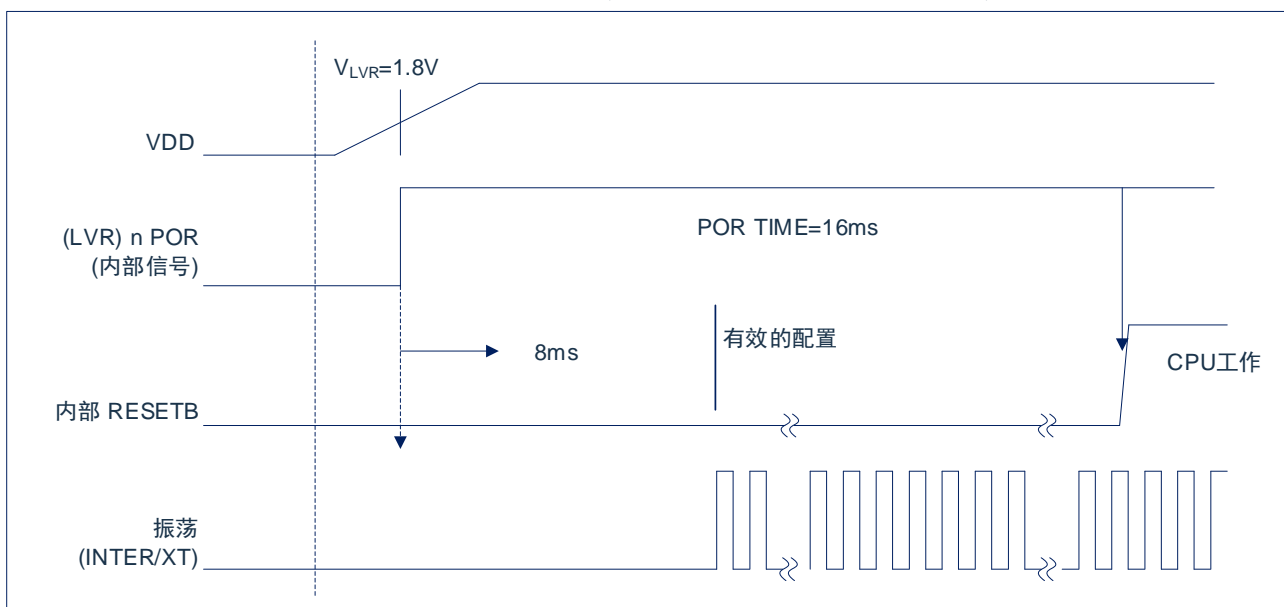


图 4-1：上电复位时序图

系统是否为上电复位可通过 PORF (WDCON.6) 标志位来判断。

WDCON 寄存器

0x97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	SWRST	PORF	--	--	WDTIF	WDTRF	WDTRE	WDTCLR
R/W	R/W	R/W	--	--	R/W	R/W	R/W	R/W
复位值	0	1	--	--	0	0	0	0

Bit7 SWRST: 软件复位控制位;
 1: 执行系统软件复位 (复位后需写 0 清掉)。
 0: --

Bit6 PORF: 复位标志位;
 1: 系统为上电复位/LVR 复位/外部复位/CONFIG 保护复位 (写 0 清掉, 不需要 TA 写时序);
 0: --

Bit5~ Bit4 未用。

Bit3~ Bit0 WDT:相关控制/标志位, 详见 WDT 章节。

可置 PORF 标志位为 1 的复位类型有: 上电复位、LVR 复位、外部复位、CONFIG 保护复位。

4.2 外部复位

外部复位指来自外部端口（NRST）的复位信号，经施密特触发器输入后复位芯片。如果在操作电压范围及稳定振荡情况下 NRST 脚保持低电平约 16us 以上（内部 LSI 时钟采样 3 个上升沿），就会请求复位。内部状态被初始化复位状态变为“1”后，需要 16ms 的稳定时间，内部 RESETB 信号才变为“1”，程序从向量地址 0000H 处开始执行。

稳定时间（Stabilization Time）内芯片重新进行配置过程，与上电复位配置过程相同。外部复位引脚 NRST 及其上拉电阻使能，通过 CONFIG 配置。

外部复位引脚 NRST 可配置到任意一外部端口，配置后该端口的 GPIO 功能及复用功能被禁止。

4.3 LVR 低压复位

芯片内部集成了低压复位（LVR）功能，当系统电压 VDD 跌至低于 LVR 电压时，LVR 被触发，系统复位。触发复位的电压点可在 CONFIG 中设置。

LVR 模块检测到 $VDD < V_{LVR}$ ，即 LVR 输出保持低电平约 16us 以上（内部 LSI 时钟采样 3 个上升沿），就会请求复位。

休眠模式（STOP）模式下，LVR 低压复位功能禁止。

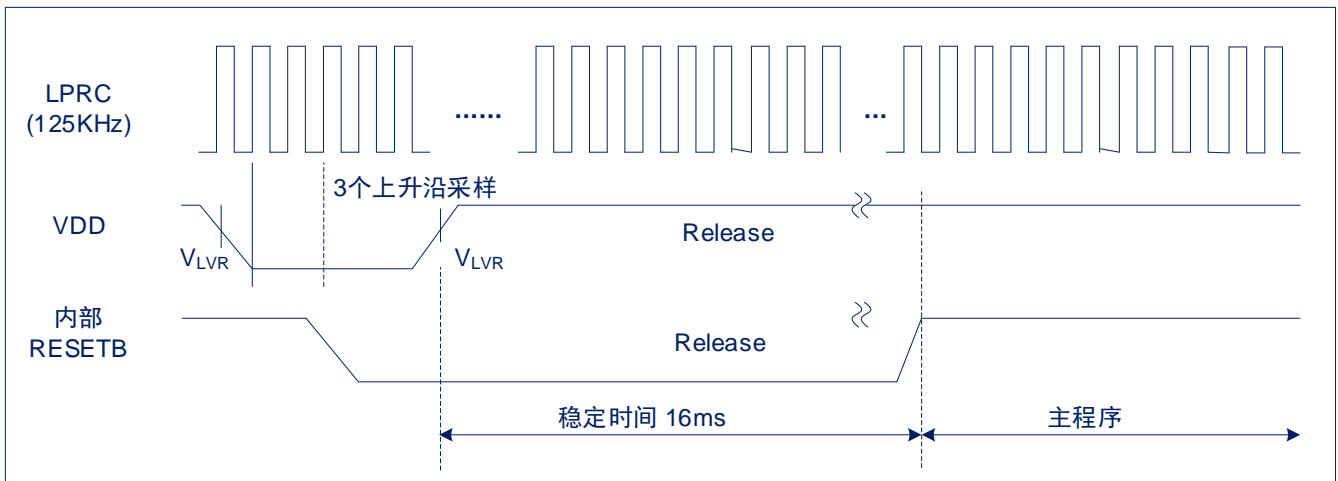


图 4-2: LVR 低压复位时序图

稳定时间（Stabilization Time）内芯片重新进行配置过程，与上电复位配置过程相同。

4.4 看门狗复位 (WDT)

看门狗复位是系统的一种保护设置。在正常状态下，由程序将看门狗定时器清零。若出错，系统处于未知状态，看门狗定时器溢出，此时系统复位。看门狗复位后，系统重启进入正常状态。

WDT 的计数器不可被寻址，在上电复位结束后程序运行时就开始计数，设置 WDT 寄存器时建议将 WDT 计数器清除，以便准确控制 WDT 的溢出时间。

看门狗复位的时序如下：

- 看门狗定时器状态：系统检测看门狗定时器是否溢出，若溢出，则系统复位；
- 初始化：所有的系统寄存器被置为默认状态；
- 程序：复位完成，程序开始从 0000H 运行。

WDT 溢出后复位 CPU 与所有的寄存器，1 个 T_{sys} 后程序立即从 0000H 开始执行。WDT 复位不会重新进行上电复位配置。

WDT 的时钟源由系统时钟 (SystemClock) 提供，WDT 计数器的计时基本周期为 T_{sys} 。

看门狗的溢出时间可由程序设置，在 CKCON 寄存器 WDS2-WTS0 两位可选择溢出时间。

WTS[2:0]	Watchdog 间隔	时钟数	OVT@Fsys=16MHz
000	2^{17}	131072	8.192ms
001	2^{18}	262144	16.384ms
010	2^{19}	524288	32.768ms
011	2^{20}	1048576	65.536ms
100	2^{21}	2097152	131.072ms
101	2^{22}	4194304	262.144ms
110	2^{24}	16777216	1.048s
111	2^{26}	67108864	4.194s

WDT 也可以设置为不会复位系统，可以产生中断。

4.5 软件复位

芯片内部可实现程序软件复位，软件复位可将程序流程重新定位到复位地址 0000H，然后再次运行程序。用户可实现自定义的软件复位，实现软件复位的控制位 SWRST (WDCON.7)。软件复位不会重新进行上电复位配置。

WDCON

0x97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	SWRST	PORF	--	--	WDTIF	WDTRF	WDTRE	WDTCLR
R/W	R/W	R/W	--	--	R/W	R/W	R/W	R/W
复位值	0	1	--	--	0	0	0	0

Bit7 SWRST: 软件复位控制位。

1: 执行系统软件复位（由 0->1 改变产生复位，软件复位后不会清零，需写 0 清掉）。
0: --

Bit6 PORF: 上电复位标志位。

1: 系统为上电复位（写 0 清掉，不需要 TA 写时序）。
0: --

Bit5~ Bit4 未用。

Bit3~ Bit0 WDT 相关控制/标志位，详见 WDT 章节。

4.6 CONFIG 状态保护复位

CONFIG 状态保护复位是系统的一种加强保护机制。在上电复位时，内部有一组 16 位 CONFIG 寄存器，加载 FLASH 中设置的固定代码 (A569H)，正常工作时不会操作该寄存器。若在特定非程序操作的情况下，该寄存器的值发生变化且不等于原先的固定代码，在经过若干时钟采样后，寄存器继续保持不为固定代码的状态，则系统会产生复位。

该复位机制避免在特定条件下，配置位发生变化，使得系统进入非预知的状态。

在正常工作时，采样寄存器值的时钟为内部 RC 固定时钟 Fixed_Clock (8MHz，时钟源来自 HSI) 与低功耗时钟 (LSI 125KHz)，一旦寄存器的值不为固定代码，强制使能 LSI 振荡器与 HSI 振荡器，且系统时钟切换到 LSI 时钟，若经过 12 个 Fixed_Clock 的采样或者 3 个 LSI 时钟的采样，寄存器还保持不为固定代码的状态，则系统产生复位。

在特定条件下，为防止振荡器停振的情况，所以采用两种时钟进行采样。

4.7 上电配置监控复位

在上电配置过程中，芯片内部有一个配置监控电路，如果在上电配置时间过长，或者上电配置进入某种状态不能重新配置，内部的监控电路从配置开始进行计时，若超过设置时间，则监控电路复位配置模块，让配置模块重新进行配置过程。以防系统在上电时进入非预知状态。

监控电路工作时钟为 LSI (125KHz)，默认的监控时间为 65ms，若选择 32.768KHz 晶振，则监控时间为 2.1s。

5. 电源管理

低功耗模式分为 2 类：

- IDLE：空闲模式
- STOP：休眠模式

用户利用 C 语言进行程序开发时，强烈建议使用 IDLE 和 STOP 宏指令来控制单片机的系统模式，不要直接设置 IDLE 和 STOP 位。

- 进入空闲模式：IDLE();
- 进入休眠模式：STOP();

5.1 电源管理寄存器 PCON

电源管理控制寄存器 PCON

0x87	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	SMOD1	--	--	--	SWE	STOP	IDLE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	SMOD0: UART0 波特率倍增位; 0= UART0 波特率正常; 1= UART0 波特率加倍。
Bit6	SMOD1: UART1 波特率倍增位; 0= UART1 波特率正常; 1= UART1 波特率加倍。
Bit5	--
Bit4~Bit3	保留位: 必须为 0。
Bit2	SWE: STOP 状态功能唤醒使能位; (不论 SWE 为何值, 系统均可由掉电复位或使能的外部复位重新启动) 0= 禁止功能唤醒; 1= 允许功能唤醒 (可由外部中断唤醒及定时唤醒)。
Bit1	STOP: 休眠状态控制位; 0= 未进入休眠状态; 1= 进入休眠状态 (退出 STOP 模式自动清零)。
Bit0	IDLE: 空闲状态控制位; 0= 未进入空闲状态; 1= 进入空闲状态 (退出 IDLE 模式自动清零)。

5.2 电源监控寄存器 LVDCON

该 MCU 自带电源检测功能。若设置 LVD 模块使能 (LV DEN=1)，同时设置好电压监测点 LVDSEL，当电源电压下降到低于 LVD 设定值时，将会产生中断，提醒用户。

如果休眠前 LVD 模块使能，进入休眠后硬件不会关闭该模块电路，需软件关闭 (LV DEN=0)。

电源监控寄存器 LVDCON

F690H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDCON	--	LVDSEL[2:0]			LV DEN	--	LVDINTE	LVDINTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	--	
Bit6~Bit4	LVDSEL[2:0]: LVD 电压监测点	
	000=	2.0V
	001=	2.2V
	010=	2.4V
	011=	2.7V
	100=	3.0V
	101=	3.7V
	110=	4.0V
	111=	4.3V
Bit3	LV DEN: LVD 模块使能	
	0=	禁止
	1=	使能
Bit2	--	保留，须为 0
Bit1	LVDINTE: LVD 中断使能位	
	0=	禁止
	1=	使能
Bit0	LVDINTF: LVD 中断标志位 (软件清零)	
	0=	电源电压高于监测电压
	1=	电源电压低于监测电压

5.3 IDLE 空闲模式

在此种模式下，只有 CPU 时钟源被关闭。因此，在这种状态下，外设功能 (如定时器、PWM 和 I²C) 和时钟发生器 (HSI/晶振驱动器) 仍然正常工作。

系统进入空闲模式后，可由任意中断唤醒，唤醒后进入中断处理程序，中断返回后，继续执行休眠操作后指令。

如果在中断服务程序中进入空闲模式，则只能由优先级较高的中断唤醒系统。

5.4 STOP 休眠模式

在此种模式下，除 LVD 模块、LSE 模块外其他所有的电路被关闭（LVD/LSE 模块须通过软件关闭），系统处于低功耗模式，数字电路均不工作。

5.4.1 休眠模式功能唤醒

进入休眠模式后，可开启休眠唤醒功能（需设置 SWE=1）唤醒休眠模式。有以下几种方式唤醒休眠模式：

1) INT0/1 中断

采用 INT0/1 中断唤醒休眠模式，必须在进入休眠前将总中断使能与 INT0/1 中断使能打开，才能唤醒系统。INT0，INT1 中断相关寄存器包括 IE，IP，TCON，IO 复用映射寄存器，INT0/1 中断唤醒只能下降沿中断唤醒休眠。

2) 外部（GPIO）中断

采用外部 GPIO 中断唤醒，必须在进入休眠前将总中断使能与端口中断使能打开，才能唤醒系统。外部 GPIO 中断唤醒可选择上升沿、下降沿、双沿中断唤醒休眠，中断唤醒边沿由外部中断控制寄存器 PxnEICFG 设置。

3) WUT 定时唤醒

由 WUT 定时唤醒，必须在进入休眠前将定时唤醒功能打开，同时设置好休眠状态到唤醒的时间。定时唤醒电路的时钟源由 LSI（低功耗振荡器）提供，开启定时唤醒功能则 LSI 在休眠状态下自动打开。

4) LSE 定时唤醒

由 LSE 定时唤醒，必须在进入休眠前将 LSE 模块使能、计数使能、定时唤醒功能打开，同时设置好休眠状态到唤醒的时间。

5.4.2 唤醒等待状态

无论是 INT0/1 中断、外部 GPIO 中断，还是 WUT 定时唤醒、LSE 定时唤醒休眠模式，在中断产生或定时时间到后，都需要等待一段时间才能唤醒系统，执行程序的下一条指令。中断产生或定时时间到后，系统振荡器启动，但振荡频率还未稳定，CPU 未工作，PC 仍停止在休眠状态，系统需要等待一段时间才将时钟提供给 CPU。唤醒 CPU 等待的时间在烧写 CONFIG 中设置，等待时间可设置为 50us~1s。唤醒等待时间过后，MCU 认为系统时钟已经稳定，才将时钟提供给 CPU，程序继续执行。

如果内部唤醒定时器和外部中断唤醒功能都为打开状态，系统进入休眠模式后，任何一种唤醒方式都能唤醒 CPU。如果内部定时器先唤醒振荡器，之后有外部中断输入，那么在唤醒等待时间过后，程序先执行中断处理程序然后再继续执行休眠操作后的指令。

5.4.3 休眠唤醒时间

利用外部中断唤醒系统总的唤醒时间为：

电源管理器稳定时间（200us）+唤醒等待时间

利用定时唤醒系统总的唤醒时间为：

电源管理器稳定时间（200us）+唤醒定时器的计时+唤醒等待时间

（上述给出时间的条件为 $F_{sys} > 1\text{MHz}$ ）

5.4.4 复位操作重新启动系统

在休眠模式下，也可通过掉电复位或外部复位重新启动系统，该重启方式与 SWE 的值无关，即使 SWE=0 也可通过上述复位操作重启系统。

掉电复位：不需要任何其他条件，VDD 降低至 0V 后重新上电至工作电压，进入上电复位状态。

外部复位：需要开启外部复位功能，相关端口配置成专用复位口，在休眠时复位口保持>1us 的低电平，系统产生复位，释放复位口，则系统重新启动。

5.5 休眠模式应用举例

系统在进入休眠模式之前，若用户需要获得较小的休眠电流，请先确认所有 I/O 的状态，若用户方案中存在悬空的 I/O 口，把所有悬空口都设置为输出口，确保每一个输入口都有一个固定的状态，以避免 I/O 为输入状态时，口线电平处于不定态而增大休眠电流；关断 ADC 模块、LSE 模块、LVD 模块及其他外设以减小休眠电流。

例：进入休眠的处理程序（汇编程序）

```
SLEEP_MODE:
    MOV            IE,#00h
    MOV            P0TRIS,#0FFh
    MOV            P0,#0FFh
    MOV            P1TRIS,#0FFh
    MOV            P1,#0FFh
    MOV            P2TRIS,#0FFh
    MOV            P2,#0FFh
    MOV            P3TRIS,#0FFh
    MOV            P3,#0FFh
    关闭其它功能的操作指令
    MOV            PCON,#06H        ; 执行可功能唤醒的休眠操作,
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP                ;执行休眠操作的指令后必须跟 6 条 NOP 指令
    唤醒后的其他操作指令
```

5.6 调试模式下的休眠功耗

调试模式下的休眠状态并不能反映实际芯片休眠状态。

在调试模式下，系统进入休眠状态后，相关的电源管理电路，振荡器并没有关闭，而是继续开启。调试模式下也可以进行唤醒操作，其唤醒方式与正常模式相同。

故在此状态下，测试得到的休眠电流并不是真正的休眠功耗。建议在调试模式下将休眠唤醒功能开发完成后关闭调试模式，然后重新启动系统，此时的测得电流即为实际休眠功耗。

6. I/O 端口

6.1 GPIO 功能

芯片有四组 I/O 端口：PORT0、PORT1、PORT2、PORT3，可读写端口数据寄存器直接存取这些端口。

PORTx 是双向端口。它所对应的数据方向寄存器是 PxTRIS。将 PxTRIS 的一个位置 1 (= 1) 可以将相应的引脚配置为输出。清零 PxTRIS 的一个位 (= 0) 可将相应的 PORTx 引脚配置为输入。

PORTx 作输出口时，写 Px 寄存器将会写入端口锁存器，所有写操作都是读-修改-写操作。因此，写一个端口就意味着先读该端口的引脚电平，然后修改读到的值，最后再将改好的值写入端口数据锁存器。

PORTx 作输出口时，读 Px 寄存器，与 PxDS 寄存器的设置有关。PxDS 的一个位置 1 (=1)，读取的 Px 相应位为引脚的状态，清零 PxDS 的一个位 (=0)，读取的 Px 相应位为端口数据锁存器的状态；PORTx 作输入口时，读 Px 寄存器读的是引脚的状态，与 PxDS 寄存器的设置无关。

当将 PORTx 引脚用作模拟输入时，用户必须确保 PxTRIS 寄存器中的位保持为置 0 状态。配置为模拟输入的 I/O 引脚总是读为 0。

与 PORTx 口相关寄存器有 Px、PxTRIS、PxOD、PxUP、PxRD、PxDS 等。

6.1.1 PORTx 数据寄存器 Px

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Px	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

寄存器 P0 地址：0x80；寄存器 P1 地址：0x90；寄存器 P2 地址：0xA0；寄存器 P3 地址：0xB0。

Bit7~Bit0 Px<7:0>: Px I/O 引脚位。
 1= 端口引脚电平>VIH;
 0= 端口引脚电平<VIL。

6.1.2 PORTx 方向寄存器 PxTRIS

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxTRIS	PxTRIS7	PxTRIS6	PxTRIS5	PxTRIS4	PxTRIS3	PxTRIS2	PxTRIS1	PxTRIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0TRIS 地址：0x9A；寄存器 P1TRIS 地址：0xA1；寄存器 P2TRIS 地址：0xA2；寄存器 P3TRIS 地址：0xA3。

Bit7~Bit0 PxTRIS<7:0>: 三态控制位。
 1= 引脚被配置为输出；
 0= 引脚被配置为输入（三态）。

注：

- 1) 端口设置为输出口后，读取该端口的数据为输出寄存器的值。
- 2) 端口设置为输入口后，对端口进行的<读取-修改-写入>类型的指令实际是对输出寄存器的操作。

6.1.3 PORTx 开漏控制寄存器 PxOD

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxOD	PxOD7	PxOD6	PxOD5	PxOD4	PxOD3	PxOD2	PxOD1	PxOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0OD 地址: F009H; 寄存器 P1OD 地址: F019H;

寄存器 P2OD 地址: F029H; 寄存器 P3OD 地址: F039H。

- Bit7~Bit0 PxOD<7:0>: 开漏控制位。
- 1= 引脚被配置为开漏状态 (输出为开漏输出);
 - 0= 引脚被配置为正常状态 (输出为推挽输出)。

6.1.4 PORTx 上拉电阻控制寄存器 PxUP

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxUP	PxUP7	PxUP6	PxUP5	PxUP4	PxUP3	PxUP2	PxUP1	PxUP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0UP 地址: F00AH; 寄存器 P1UP 地址: F01AH;

寄存器 P2UP 地址: F02AH; 寄存器 P3UP 地址: F03AH。

- Bit7~Bit0 PxUP<7:0>: 上拉电阻控制位;
- 1= 引脚上拉电阻打开;
 - 0= 引脚上拉电阻关闭。

6.1.5 PORTx 下拉电阻控制寄存器 PxRD

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxRD	PxRD7	PxRD6	PxRD5	PxRD4	PxRD3	PxRD2	PxRD1	PxRD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0RD 地址: F00BH; 寄存器 P1RD 地址: F01BH;

寄存器 P2RD 地址: F02BH; 寄存器 P3RD 地址: F03BH。

- Bit7~Bit0 PxRD<7:0>: 下拉电阻控制位;
- 1= 引脚下拉电阻打开;
 - 0= 引脚下拉电阻关闭。

注: 下拉电阻的控制与 GPIO 的配置和复用的功能无关, 由 PxRD 寄存器单独控制。

6.1.6 PORTx 驱动电流控制寄存器 PxDR

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxDR	PxDR7	PxDR6	PxDR5	PxDR4	PxDR3	PxDR2	PxDR1	PxDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0DR 地址: F00CH; 寄存器 P1DR 地址: F01CH;

寄存器 P2DR 地址: F02CH; 寄存器 P3DR 地址: F03CH;

- Bit7~Bit0 PxDR<7:0>: 驱动电流控制位 (在端口配置成输出状态时生效)。
- 1= 驱动为弱驱动;
 - 0= 驱动为强驱动。

6.1.7 PORTx 斜率控制寄存器 PxSR

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxSR	PxSR7	PxSR6	PxSR5	PxSR4	PxSR3	PxSR2	PxSR1	PxSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0SR 地址: F00DH; 寄存器 P1SR 地址: F01DH;

寄存器 P2SR 地址: F02DH; 寄存器 P3SR 地址: F03DH;

Bit7~Bit0 PxSR<7:0>: Px 斜率控制寄存器 (在端口配置成输出状态时生效)。
 1= Px引脚为慢斜率;
 0= Px引脚为快斜率。

6.1.8 PORTx 数据输入选择寄存器 PxDS

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxDS	PxDS7	PxDS6	PxDS5	PxDS4	PxDS3	PxDS2	PxDS1	PxDS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 P0DS 地址: F00EH; 寄存器 P1DS 地址: F01EH;

寄存器 P2DS 地址: F02EH; 寄存器 P3DS 地址: F03EH。

Bit7~Bit0 PxDS<7:0>: 数据输入选择位, 配置为GPIO时影响读取Px寄存器的值。
 1= 输出/输入模式均读取引脚状态
 (端口设置为输出态时斯密特电路也保持打开);
 0= 输出模式: 读为数据锁存器状态,
 输入模式: 读为引脚状态。

注: 如需要读取端口为复用功能输入结构时的引脚状态, 则需将该端口方向控制置为输入模式。

6.2 复用功能

6.2.1 端口复用功能配置寄存器

PORTx 功能配置寄存器 PxnCFG

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxnCFG	--	--	--	PxnCFG4	PxnCFG3	PxnCFG2	PxnCFG1	PxnCFG0
R/W	--	--	--	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit5 -- 保留，须均为0。

Bit4~Bit0 PxnCFG<4:0>: 功能配置位，默认为GPIO功能。详见端口功能配置说明。

Px的功能配置寄存器有8个，包括Px0CFG~Px7CFG，分别控制Px0~Px7的功能配置。

每个端口都有一个功能配置寄存器，通过功能配置寄存器 PxnCFG 每个端口均能配置成任意一种数字功能。如端口 P00 的功能配置寄存器为 P00CFG，寄存器设置不同的值对应不同的数字功能。例如：要将 P2.4 设置为 BEEP 蜂鸣器功能，配置如下：

汇编：MOV #P24CFG, #18H

C: P24CFG = 0x18;

端口作复用功能时，无需配置端口方向寄存器 PxTRIS。并且除 SCL 和 SDA 功能外，其他复用功能均硬件强制关闭上拉电阻、关闭开漏输出，即通过软件设置上拉电阻 PxUP 或者开漏输出 PxOD 无效。

端口复用为 SCL 和 SDA 功能时，硬件强制端口为开漏输出，可通过软件设置上拉电阻 PxUP。

不同配置值对应的数字功能如下：

配置值	功能	方向	功能说明
0x00	GPIO	I/O	通用 IO 口，通过寄存器配置输入输出，上下拉等功能
0x01	ANALOG		模拟功能
0x02	--	--	--
0x03	--	--	--
0x04	CC0	O	Timer2 比较输出通道 0
0x05	CC1	O	Timer2 比较输出通道 1
0x06	CC2	O	Timer2 比较输出通道 2
0x07	CC3	O	Timer2 比较输出通道 3
0x08	TXD0	O	UART0 数据输出
0x09	RXD0	I/O	UART0 数据输入/同步模式数据输出
0x0A	TXD1	O	UART1 数据输出
0x0B	RXD1	I/O	UART1 数据输入/同步模式数据输出
0x0C	SCL	I/O	I ² C 时钟输入输出
0x0D	SDA	I/O	I ² C 数据输入输出
0x0E	NSS	I/O	SPI 从控模式机片选信号（输入/输出）
0x0F	SCLK	I/O	SPI 时钟输入输出
0x10	MOSI	I/O	SPI 主控发送从控接收
0x11	MISO	I/O	SPI 主控接收从控发送

配置值	功能	方向	功能说明
0x12	PG0	O	PWM 通道 0 输出
0x13	PG1	O	PWM 通道 1 输出
0x14	PG2	O	PWM 通道 2 输出
0x15	PG3	O	PWM 通道 3 输出
0x16	PG4	O	PWM 通道 4 输出
0x17	PG5	O	PWM 通道 5 输出
0x18	BEEP	O	蜂鸣器输出
0x19	--	--	--
0x1A	--	--	--
0x1B	--	--	--
0x1C	--	--	--
0x1D	--	--	--
0x1E	--	--	--
0x1F	--	--	--

注：

- 1) 表格中标注为“--”的配置值保留，禁止使用。
- 2) 功能配置寄存器默认为 0x00，端口作 GPIO 功能。可通过第 6.1 章的 GPIO 功能寄存器设置不同的功能。
- 3) 功能配置寄存器设置为 0x01 时，硬件关闭数字电路以降低功耗，GPIO 功能相关寄存器设置无效。端口支持多个模拟功能，见下表描述。
- 4) 复用功能中作输出功能的端口没有优先级顺序的限制。如果有多个端口配置成相同的输出功能，则该功能会在这些端口同时输出。
- 5) 复用功能中作输入功能的端口有优先级顺序的限制。如果有两个或以上的端口同时配置相同输入功能，则按照 P0.0, P0.1,, P3.2, P3.5 从高到低的优先级顺序来配置选择。
 如将 P00 与 P32 同时配置成 RXD0 口：P00CFG = 0x09;P32CFG = 0x09。
 由于 P0.0 有更高的优先级，实际上 RXD0 信号源连接到 P0.0 口的输入，即使 P3.2 端口的有数据波形，也将不会作为 RXD0 信号源。

端口对应的模拟功能如下:

PIN	CONFIG	ANALOG	其他数字功能优先级
P0.0		AN0	最高
P0.1		AN1	
P0.2		AN2	
P0.3		AN3	
P0.4		AN4	
P0.5		AN5	
P1.3		AN6	
P1.4		AN7	
P1.5		AN18	
P1.6		AN19	
P1.7		AN20	
P2.1	DSCK	AN21	
P2.2		AN8	
P2.3		AN9	
P2.4		AN10	
P2.5		AN11	
P2.6		AN12	
P3.0		AN22	
P3.1	OSCIN	AN13	
P3.2	OSCOUT	AN14	
P3.5	DSDA	AN16	
P3.6		AN17	最低

- P0TRIS = 0x00;//P00 作 GPIO 输入功能
 PS_INT0 = 0x00;//P00 端口配置为 INT0 功能
 PS_CAP0 = 0x00;//P00 端口配置为 CAP0 功能
- 2) 该输入功能分配结构相对独立, 可支持与其他复用功能端口同时使用, 此时无需配置对应端口的方向寄存器, 如可同时将 RXD0 与 INT0 分配到 P00 口, 配置如下:
 P00CFG = 0x09;//P00 端口配置为 UART0 的 RXD0 功能
 PS_INT0 = 0x00;//P00 端口配置为 INT0 功能
- 3) 该输入功能配置结构也可与端口外部中断功能同时使用。如可同时将 CAP0 和 GPIO 中断功能分配到 P00 口, 配置如下:
 P00CFG = 0x00;//P00 端口配置为 GPIO 功能
 P0TRIS = 0x00;//P00 作 GPIO 输入功能
 PS_CAP0 = 0x00;//P00 端口配置为 CAP0 功能
 P00EICFG = 0x01;//P00 端口配置为上升沿触发中断
 P0EXTIE = 0x01;//允许 P00 的端口外部中断

6.2.3 端口外部中断控制寄存器

使用外部中断时, 需要将端口配置成 GPIO 功能, 且将方向置为输入口。或者复用功能为输入端口 (如 RXD0, RXD1)。

PORTx 外部中断控制寄存器 P_{XN}EICFG

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P _{XN} EICFG	--	--	--	--	--	--	P _{X1} EICFG1	P _{X0} EICFG0
R/W	--	--	--	--	--	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7~Bit2 未用。
 Bit1~Bit0 P_{XN}EICFG<1:0>: P_{XN}外部中断控制位;
 00= 外部中断禁止;
 01= 上升沿触发中断;
 10= 下降沿触发中断;
 11= 上升沿或下降沿均触发中断。

P_x的外部中断控制寄存器有8个, 包括P_{x0}EICFG~P_{x7}EICFG, 分别控制P_{x0}~P_{x7}的外部中断。

6.2.4 复用功能应用注意

- 1) 复用功能的输入与端口的的外部中断 (GPIO 中断)、端口输入功能的结构相对独立。
 例如将 P01 口配置为 RXD0, 同时配置 P01 的 GPIO 中断触发方式为上升沿触发、中断使能, 当 P01 输入从低电平变到高电平时, 将触发 P01 的 GPIO 中断。
- 2) 数字信号的输入结构不受系统配置状态的影响。
 例如将 P01 口上电配置成外部复位口, 则该端口输入模块打开。若在程序中同时将 P01 配置成 INT0, 中断使能打开, 则复位信号采样时间有效之前, 先执行中断服务程序, 然后产生复位操作。
- 3) 需要注意的是, 在调试模式下, 如复用功能配置到 DSDA 口上, 其输入功能同样有效, 建议在调试模式下, 不要将相关的复用功能配置到 DSDA 口。

7. 中断

7.1 中断概述

芯片具有 23 个中断源及中断向量：

中断源	中断描述	中断向量	同级优先序列
INT0	外部中断 0	0-0x0003	1
Timer0	定时器 0 中断	1-0x000B	2
INT1	外部中断 1	2-0x0013	3
Timer1	定时器 1 中断	3-0x001B	4
UART0	Ti0or Ri0	4-0x0023	5
Timer2	定时器 2 中断	5-0x002B	6
UART1	Ti1or Ri1	6-0x0033	7
P0EXTIF<7:0>	P0 口外部中断	7-0x003B	8
P1EXTIF<7:0>	P1 口外部中断	8-0x0043	9
P2EXTIF<7:0>	P2 口外部中断	9-0x004B	10
P3EXTIF<7:0>	P3 口外部中断	10-0x0053	11
--	--	11-0x005B	12
LVD	LVD 掉电中断	12-0x0063	13
LSE_Timer	LSE 定时器中断	13-0x006B	14
--	--	14-0x0073	15
Timer3	定时器 3 中断	15-0x007B	16
Timer4	定时器 4 中断	16-0x0083	17
--	--	17-0x008B	18
PWM	PWM 中断	18-0x0093	19
ADC	ADC 中断	19-0x009B	20
WDT	WDT 中断	20-0x00A3	21
I ² C	I ² C 中断	21-0x00AB	22
SPI	SPI 中断	22-0x00B3	23

芯片规定两个中断优先级，可实现两级中断嵌套。

当一个中断已经响应，若有高级别中断发出请求，后者可以中断前者，实现中断嵌套。但同级别或低级别中断不能中断高级别中断响应。

7.2 外部中断

7.2.1 INTO/INT1 中断

芯片支持 8051 的原生的 INTO, INT1 外部中断, INTO/INT1 可选择下降沿或低电平触发中断, 相关控制寄存器为 TCON。INT0 与 INT1 占用两个中断向量。

7.2.2 GPIO 中断

芯片每个 GPIO 引脚均支持外部中断。且可支持下降沿/上升沿/双沿中断, 选择何种边沿触发在 P_{xN}EICFG 寄存器配置, 例: 如需要将 P1.3 口配置下降沿中断:

```
P13CFG=0x00; //将 P1.3 设置为 GPIO
P1TRIS&=0xF7; //将 P1.3 置为输入口
P13EICFG=0x02; //设置 P1.3 为下降沿触发中断
```

GPIO 的中断共占用 4 个中断向量:

- P0.0-P0.5 占用一个中断向量 0x003B;
- P1.3-P1.7 占用一个中断向量 0x0043;
- P2.1-P2.6 占用一个中断向量 0x004B;
- P3.0-P3.2, P3.5-P3.6 占用一个中断向量 0x0053。

如中断产生后, 进入中断服务程序可先判断是哪个端口触发中断, 然后再进行相应的处理。

7.2.3 中断与休眠唤醒

系统在进入休眠模式 (STOP 可唤醒模式) 后, 每个外部中断可设置唤醒系统。

INT0/INT1 中断唤醒系统需要打开相应的中断使能以及总中断使能, 唤醒方式为下降沿唤醒 (INT0/INT1 唤醒方式和中断触发方式选择位 IT0/IT1 无关)。

GPIO 中断唤醒系统, 建议在进入修改模式前设置好相应的端口中断触发边沿方式 (GPIO 的唤醒方式与中断触发边沿方式相同, 可选择上升沿/下降沿/双沿唤醒), 以及打开相应的中断使能及总中断使能。

系统由外部中断唤醒后, 首先进入中断服务程序处理中断唤醒任务, 退出中断服务程序后, 系统继续执行休眠操作后的指令。

7.3 中断寄存器

7.3.1 中断屏蔽寄存器

中断屏蔽寄存器 IE 是可读写的寄存器，可位操作。

当有中断条件产生时，无论对应的中断允许位或（IE 寄存器中的）全局允许位 EA 的状态如何，中断标志位都将置 1。用户软件应在允许一个中断之前，确保先将相应的中断标志位清零。

中断屏蔽寄存器 IE（0xA8）

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	EA: 全局中断允许位; 1= 允许所有未被屏蔽的中断; 0= 禁止所有中断。
Bit6	ES1: UART1中断允许位; 1= 允许UART1中断; 0= 禁止UART1中断。
Bit5	ET2: TIMER2总中断允许位; 1= 允许TIMER2所有中断; 0= 禁止TIMER2所有中断。
Bit4	ES0: UART0中断允许位; 1= 允许UART0中断; 0= 禁止UART0中断。
Bit3	ET1: TIMER1中断允许位; 1= 允许TIMER1中断; 0= 禁止TIMER1中断。
Bit2	EX1: 外部中断1中断允许位; 1= 允许外部中断1中断; 0= 禁止外部中断1中断。
Bit1	ET0: TIMER0中断允许位; 1= 允许TIMER0中断; 0= 禁止TIMER 0中断。
Bit0	EX0: 外部中断0中断允许位; 1= 允许外部中断0中断; 0= 禁止外部中断0中断。

Timer2 中断屏蔽寄存器 T2IE (0xCF)

0xCF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IE	T2OVIE	T2EXIE	--	--	T2C3IE	T2C2IE	T2C1IE	T2C0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 T2OVIE: Timer2溢出中断允许位;
1= 允许中断;
0= 禁止中断。
- Bit6 T2EXIE: Timer2外部加载中断允许位;
1= 允许中断;
0= 禁止中断。
- Bit5 --
- Bit4 --
- Bit3 T2C3IE: Timer2比较/捕获通道3中断允许位;
1= 允许中断;
0= 禁止中断。
- Bit2 T2C2IE: Timer2比较/捕获通道2中断允许位;
1= 允许中断;
0= 禁止中断。
- Bit1 T2C1IE: Timer2比较/捕获通道1中断允许位;
1= 允许中断;
0= 禁止中断。
- Bit0 T2C0IE: Timer2比较/捕获通道0中断允许位;
1= 允许中断;
0= 禁止中断。

若开启 Timer2 的中断，还需要开启 Timer2 的总中断使能位 ET2=1 (IE.5=1)

P0 口中断控制寄存器 P0EXTIE (0xAC)

0xAC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTIE	--	--	P05IE	P04IE	P03IE	P02IE	P01IE	P00IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7~Bit6 --
- Bit5~Bit0 P0iIE: P0i口中断允许位 (i=0-5);
1= 允许中断;
0= 禁止中断。

P1 口中断控制寄存器 P1EXTIE (0xAD)

0xAD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTIE	P17IE	P16IE	P15IE	P14IE	P13IE	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3 P1iIE: P1i口中断允许位(i=3-7);
 1= 允许中断;
 0= 禁止中断。

Bit2~Bit0 --

P2 口中断控制寄存器 P2EXTIE (0xAE)

0xAE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTIE	--	P26IE	P25IE	P24IE	P23IE	P22IE	P21IE	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 --

Bit6~Bit1 P2iIE: P2i口中断允许位(i=1-6);
 1= 允许中断;
 0= 禁止中断。

Bit0 --

P3 口中断控制寄存器 P3EXTIE (0xAF)

0xAF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3EXTIE	--	P36IE	P35IE	--	--	P32IE	P31IE	P30IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 --

Bit6~Bit5 P3iIE: P3i口中断允许位(i=5-6);
 1= 允许中断;
 0= 禁止中断。

Bit4~Bit3 --

Bit2~Bit0 P3iIE: P3i口中断允许位(i=0-2);
 1= 允许中断;
 0= 禁止中断。

中断屏蔽寄存器 EIE2 (0xAA)

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 SPIIE: SPI中断使能位;
 1= 允许SPI中断;
 0= 禁止SPI中断。
- Bit6 I2CIE: I²C中断使能位;
 1= 允许I²C中断;
 0= 禁止I²C中断。
- Bit5 WDTIE: WDT中断使能位;
 1= 允许WDT溢出中断;
 0= 禁止WDT溢出中断。
- Bit4 ADCIE: ADC中断使能位;
 1= 允许ADC中断;
 0= 禁止ADC中断。
- Bit3 PWMIE: PWM总中断使能位;
 1= 允许PWM所有中断;
 0= 禁止PWM所有中断。
- Bit2 --
- Bit1 ET4: Timer4中断使能位;
 1= 允许Timer4中断;
 0= 禁止Timer4中断。
- Bit0 ET3: Timer3中断使能位;
 1= 允许Timer3中断;
 0= 禁止Timer3中断。

7.3.2 中断优先级控制寄存器

中断优先级控制寄存器 IP 是可读写的寄存器，可位操作。

中断优先级控制寄存器 IP (0xB8)

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	--
Bit6	PS1: UART1中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit5	PT2: TIMER2中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit4	PS0: UART0中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit3	PT1: TIMER1中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit2	PX1: 外部中断1中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit1	PT0: TIMER0中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit0	PX0: 外部中断0中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。

端口中断优先级控制寄存器 EIP1 (0xB9)

0xB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	--	PLSE	PLVD	--	PP3	PP2	PP1	PP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	--	
Bit6	PLSE	低速晶振定时器中断优先级控制位 1= 设置为高级中断; 0= 设置为低级中断。
Bit5	PLVD	LVD电压监测中断优先级控制位 1= 设置为高级中断; 0= 设置为低级中断。
Bit4	--	
Bit3	PP3:	P3口中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit2	PP2:	P2口中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit1	PP1:	P1口中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit0	PP0:	P0口中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。

外设中断优先级控制寄存器 EIP2 (0xBA)

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 PSPI: SPI中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit6 PI2C: I²C中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit5 PWDT: WDT中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit4 PADC: ADC中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit3 PPWM: PWM中断优先级控制位
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit2 --
- Bit1 PT4: TIMER4中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit0 PT3: TIMER3中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。

7.3.3 中断标志位寄存器

Timer0/1、INT0/1 中断标志位寄存器 TCON，可位寻址

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 TF1: Timer1 计数器溢出中断标志位；
 1= Timer1计数器溢出，进入中断服务程序时硬件自动清零，也可软件清零；
 0= Timer1 计数器无溢出。
- Bit6 TR1: Timer1运行控制位；
 1= Timer1 启动；
 0= Timer1 关闭。
- Bit5 TF0: Timer0计数器溢出中断标志位；
 1= Timer0计数器溢出，进入中断服务程序时硬件自动清零，也可软件清零；
 0= Timer0计数器无溢出。
- Bit4 TR0: Timer0运行控制位；
 1= Timer0启动。
 0= Timer0关闭。
- Bit3 IE1: 外部中断1标志位；
 1= 外部中断1产生中断，进入中断服务程序时硬件自动清零，也可软件清零；
 0= 外部中断1没有产生中断。
- Bit2 IT1: 外部中断1触发方式控制位；
 1= 下降沿触发；
 0= 低电平触发。
- Bit1 IE0: 外部中断0标志位；
 1= 外部中断0产生中断，进入中断服务程序时硬件自动清零，也可软件清零；
 0= 外部中断0没有产生中断。
- Bit0 IT0: 外部中断0触发方式控制位；
 1= 下降沿触发；
 0= 低电平触发。

UART0 中断标志位寄存器 SCON0，可位寻址

0x98	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON0	U0SM0	U0SM1	U0SM2	U0REN	U0TB8	U0RB8	TI0	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7~Bit2 U0SM0、U0SM1、U0SM2、U0REN、U0TB8、U0RB8:UART相关控制位，详见UART0功能说明。
- Bit1 TI0: UART0发送中断标志位；
 1= 同步模式在第8位发送结束，或者异步发送停止位。需软件清零。
 0= --
- Bit0 RI0: UART0接收中断标志位；
 1= 同步模式接收到第8位，或者异步接收完毕。需软件清零。
 0= --

TI0 与 RI0 占用同一中断向量（0023H），需要查询来判断是接收中断还是发送中断。

UART1 中断标志位寄存器 SCON1，该寄存器不可位寻址

0xEA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON1	U1SM0	U1SM1	U1SM2	U1REN	U1TB8	U1RB8	TI1	RI1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit2 U1SM0、U1SM1、U1SM2、U1REN、U1TB8、U1RB8: UART1相关控制位，详见UART1功能说明

Bit1 TI1: UART1发送中断标志位。
 1= 同步模式在第8位发送结束，或者异步发送停止位。需软件清零。
 0= --

Bit0 RI1: UART1接收中断标志位。
 1= 同步模式接收到第8位，或者异步接收完毕。需软件清零。
 0= --

TI1 与 RI1 占用同一中断向量(0033H)，需要查询来判断是接收中断还是发送中断。

Timer2 中断标志位寄存器 T2IF (0xC9)

0xC9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IF	TF2	T2EXIF	--	--	T2C3IF	T2C2IF	T2C1IF	T2C0IF
R/W	R/W	R/W	--	--	R/W	R/W	R/W	R/W
复位值	0	0	--	--	0	0	0	0

Bit7 TF2: Timer2 计数器溢出中断标志位；
 1= Timer2 计数器溢出，需软件清零；
 0= Timer2 计数器无溢出。

Bit6 T2EXIF: Timer2 外部加载标志位；
 1= Timer2 的T2EX口产生下降沿，需软件清零；
 0= --

Bit5 --

Bit4 --

Bit3 T2C3IF: Timer2 比较/捕获通道3标志位；
 1= Timer2 比较通道3 {CCH3:CCL3}={TH2:TL2}或捕获通道3产生了捕获操作，需软件清零。
 0= --

Bit2 T2C2IF: Timer2 比较/捕获通道2标志位；
 1= Timer2 比较通道2 {CCH2:CCL2}={TH2:TL2}或捕获通道2产生了捕获操作，需软件清零。
 0= --

Bit1 T2C1IF: Timer2 比较/捕获通道1标志位；
 1= Timer2 比较通道1 {CCH1:CCL1}={TH2:TL2}或捕获通道1产生了捕获操作，需软件清零。
 0= --

Bit0 T2C0IF: Timer2 比较/捕获通道0标志位；
 1= Timer2 比较通道0{RLDH:RLDL}={TH2:TL2}或捕获通道0产生了捕获操作，需软件清零。
 0= --

P0 口中断标志位寄存器 P0EXTIF (0xB4)

0xB4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTIF	--	--	P05IF	P04IF	P03IF	P02IF	P01IF	P00IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6

--

Bit5~Bit0

P0iIF: P0i口中断标志位(i=0-5);

1= P0i口产生中断, 需软件清零;

0= P0i口未产生中断。

P1 口中断标志位寄存器 P1EXTIF (0xB5)

0xB5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTIF	P17IF	P16IF	P15IF	P14IF	P13IF	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3

P1iIF: P1i口中断标志位(i=3-7);

1= P1i口产生中断, 需软件清零;

0= P1i口未产生中断。

Bit2~Bit0

--

P2 口中断标志位寄存器 P2EXTIF (0xB6)

0xB6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTIF	--	P26IF	P25IF	P24IF	P23IF	P22IF	P21IF	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7

--

Bit6~Bit1

P2[i]IF: P2[i] 口中断标志位(i=1-6);

1= P2[i] 口产生中断, 需软件清零;

0= P2[i] 口未产生中断。

Bit0

--

P3 口中断标志位寄存器 P3EXTIF (0xB7)

0xB7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3EXTIF	--	P36IF	P35IF	--	--	P32IF	P31IF	P30IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 --
- Bit6~Bit5 P3[i]IF: P3[i] 口中断标志位(i=5-6);
 1= P3[i] 口产生中断, 需软件清零;
 0= P3[i] 口未产生中断。
- Bit4~Bit3 --
- Bit2~Bit0 P3[i]IF: P3[i] 口中断标志位(i=0-2);
 1= P3[i] 口产生中断, 需软件清零;
 0= P3[i] 口未产生中断。

外设中断标志位寄存器 EIF2 (0xB2)

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 SPIIF: SPI总中断指示位, 只读;
 1= SPI产生中断, (清除具体的中断标志位后, 此位自动清除);
 0= SPI未产生中断。
- Bit6 I2CIF: I²C总中断指示位, 只读;
 1= I²C产生中断, 需软件清零 (清除具体的中断标志位后, 此位自动清除);
 0= I²C未产生中断。
- Bit5 --
- Bit4 ADCIF: ADC中断标志位;
 1= ADC转换完成, 需软件清零;
 0= ADC转换未完成。
- Bit3 PWMIF: PWM总中断指示位, 只读;
 1= PWM产生中断, (清除具体的中断标志位后, 此位自动清除);
 0= PWM未产生中断。
- Bit2 --
- Bit1 TF4: Timer4定时器溢出中断标志位;
 1= Timer4定时器溢出, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= Timer4定时器无溢出。
- Bit0 TF3: Timer3定时器溢出中断标志位;
 1= Timer3定时器溢出, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= Timer3定时器无溢出。

SPI 中断标志位寄存器 SPSR (0xED)

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	SPISIF	WCOL	--	--	--	--	--	SSCEN
R/W	R	R	--	--	--	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 SPISIF: SPI传输完毕中断标志位, 只读;
 1= SPI传输完毕 (先读SPSR, 再读/写SPDR后清零);
 0= SPI未传输完毕。
- Bit6 WCOL: SPI写冲突中断标志位, 只读;
 1= SPI传输未完成时产生写SPDR操作冲突 (先读SPSR, 再读/写SPDR后清零);
 0= 无写冲突。
- Bit5 --
- Bit4 --
- Bit3~Bit2 --
- Bit1 保留: 必须为0
- Bit0 SSCEN: SPI主控模式NSS输出控制位。
 1= SPI处于空闲状态时, NSS输出高电平;
 0= NSS输出寄存器SSCR的内容。

I²C 主控模式中断标志位寄存器 I2CMCR/I2CMSR

0xF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMCR	RSTS	--	--	--	ACK	STOP	START	RUN
I2CMSR	I2CMIF	BUS_BUSY	IDLE	ARB_LOST	DATA_ACK	ADDR_ACK	ERROR	BUSY
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 RSTS: I²C主动模块复位控制位;
 1= 复位主控模块 (整个主控模块的I²C寄存器, 包括I2CMSR);
 0= I²C主控模式下中断标志位清0。
- I2CMIF: I²C主控模式下中断标志位;
 1= 主控模式下, 发送/接收完成, 或者出现传输错误。(软件清零, 写0清除);
 0= 未产生中断。
- Bit6~Bit0 I²C主控模式下控制与标志位, 详见I2CM说明。

I²C 从动模式状态寄存器 I2CSSR

0xF2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSSR	--	--	--	--	--	SENDFIN	TREQ	RREQ
R/W	--	--	--	--	--	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3 --

Bit2 SENDFIN: I²C从动模式下发送操作完毕标志位，只读；
 1= 主控制器不再需要数据，TREQ不再设置为1，已经完成本次数据传送。（读I2CSSR后自动清零）。
 0= --

Bit1 TREQ: I²C从动模式准备发送标志位，只读；
 1= 作为发送器件已经被寻址，主控制器已经做好接收数据的准备。（写I2CSBUF后自动清零）。
 0= --

Bit0 RREQ: I²C从动模式接收完成标志位，只读；
 1= 接收完毕。（读I2CSBUF后自动清零）；
 0= 未接收完毕。

I²C 从动模式的相关状态位也是中断标志位

注意：I²C 主控模式中断与从动中断共用同一中断向量（00ABH）

7.3.4 中断标志位的清除操作

在系统中中断标志位的清除操作分为以下几种：

- 硬件自动清除（需进入中断服务程序）
- 软件清除
- 读/写操作清除

1) 硬件自动清除的标志位

支持硬件自动清除的位有 INT0、INT1、T0、T1、T3、T4 产生的中断标志位。硬件自动清除标志的条件为：打开总中断使能位 EA=1，且打开相应的中断使能位，在产生中断后系统进入相应中断服务程序，标志位自动清除。如中断使能关闭，则这些标志位也可使用软件进行清除操作。

2) 软件清除的标志位

系统中存在只能用软件清除的标志位。这些标志位在进入中断服务程序后不会自动清除，需要软件写 0 清除。否则在退出中断服务程序后又再次进入中断服务程序。

3) 读写操作清除的标志位

系统中有标志位并不是写 0 到该标志位清零，而需要读/写其他寄存器来清除标志位。如 SPI 中断标志寄存器中的传输完成标志位 SPISIF，置 1 后需要先读 SPSR，再读/写 SPDR 后清零。

软件清除操作需要注意：当多个中断标志位在同一个寄存器中，这些标志位产生的时刻相互无关联时，不建议用读-修改-写操作来进行。

比如 PWMUIF 中断标志位寄存器，其中包含 PG0-PG5 通道的向上比较中断，这些中断标志位互不相关。当 PG0 产生向上比较中断时，PWMUIF 的值为 0x01，进入中断服务程序后进行读-修改-写操作清除该位

```
PWMUIF &= 0xFE;
```

该操作具体实现为先将 PWMUIF 的值读回到 CPU，然后再进行运算，最后送回到 PWMUIF。如果 PG1 的中断标志位 PWMUIF[1]在 CPU 读之后置 1，而 PWMUIF[1]读取时为 0，则进行运算后，送回到 PWMUIF[1]也为 0，此时将会清除 PG1 已经产生的向上中断标志位 PWMUIF[1]。

清除上述类型的中断标志位，建议直接写 0 操作，其他不相关的标志位写 1：

```
PWMUIF = 0xFE;
```

该操作对不相关的中断标志写 1 无实际影响。

7.3.5 调试模式下特殊中断标志位

系统中有标志位并不是写 0 到该标志位清零，而需要读/写其他寄存器来清除标志位。

在调试态下，断点执行，单步运行或停止操作后，仿真器会从系统中读出所有的寄存器的值到仿真软件，仿真器的读/写操作与正常模式下的读/写完全一样。

所以在调试过程中，会出现暂停后，应该出现置 1 的中断标志位，但在观察窗口中显示为 0。

例：调试模式下 SPI 中断标志寄存器中的传输完成标志位 SPISIF

```
... //设置端口及中断使能
SPDR = 0x56; //发送SPDR数据
delay();
...

void SPI_int (void) interrupt SPI_VECTOR // SPI中断服务程序
{
    O1 _nop(); //设置断点1
    _nop();
    O2 k = SPSR; //设置断点2
    _nop();
    ...
}
```

在断点运行时，停止在断点 1 后，SPI 完成发送操作，已经产生发送完成中断，因此 SPSR.7=1，此时仿真器已经完成一次读所有寄存的操作（包括读 SPSR），

再次执行断点运行，停止在断点 2 后，此时仿真器再次完成读所有寄存器的操作（包括 SPDR），所以此时 SPSR.7=0。单步运行两次也会出现上述的情况，在调试模式下需要注意。

8. LSE 定时器

8.1 概述

LSE 定时器是一个时钟源来自外部低速时钟 LSE，16 位向上计数定时器。使用 LSE 定时器功能时，应先设置 LSE 模块使能后，等待 LSE 时钟稳定（约 1.5s），再设置 LSE 计数使能。定时器的在 LSE 时钟的上升沿计数值加 1，当计数值等于定时值时，中断标志位 LSECON[0]置 1，同时计数器又重新从 0 开始计数。定时值由寄存器{LSECRH[7:0]，LSECRL[7:0]}设置。

若休眠之前配置 LSE 定时功能，芯片休眠时 LSE 振荡器和 LSE 定时器仍能继续工作，而不受影响。若休眠前设置 LSE 定时唤醒功能，当计数值等于定时值时，将唤醒系统。

8.2 寄存器

LSECRL 定时器数据寄存器

F694H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSECRL	LSED7	LSED6	LSED5	LSED4	LSED3	LSED2	LSED1	LSED0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit0 LSED<7:0>: LSE定时/唤醒时间数据低8位。

LSECRH 定时器数据寄存器

F695H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSECRH	LSED15	LSED14	LSED13	LSED12	LSED11	LSED10	LSED9	LSED8
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit0 LSED<15:8>: LSE定时/唤醒时间数据高8位。

LSECON 定时器控制寄存器

F696H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSECON	LSEEN	LSEWUEN	LSECNTEN	LSESTA	LSEIE	--	--	LSEIF
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	LSEEN	LSE模块使能控制
	1=	使能
	0=	禁止
Bit6	LSEWUEN	LSE定时器唤醒使能控制
	1=	使能
	0=	禁止
Bit5	LSECNTEN	LSE作定时器计数使能控制
	1=	使能
	0=	禁止
Bit4	LSESTA	LSE稳定状态位（只读）
	1=	LSE稳定
	0=	LSE未稳定
Bit3	LSEIE	LSE作定时器中断使能控制
	1=	使能
	0=	禁止
Bit2~Bit1	--	保留
Bit0	LSEIF	LSE作定时器中断标志位（软件清0）
	1=	产生中断
	0=	未产生中断或中断清零

8.3 功能说明

使用 LSE 定时器功能，需先设置 LSEEN=1，以使能 LSE 定时器功能模块，然后等待 LSE 时钟稳定状态位 LSESTA=1 后，再配置 LSE 定时值{LSECRH[7:0], LSECRL[7:0]}，最后设置 LSECNT=1，使能 LSE 计数，开始 LSE 计数功能。LSE 定时器从 0 开始计数，当计数值与定时值相等时，中断标志位置 1，同时将定时值更新为定时器数据寄存器中的值（即 LSE 定时值为计数值与定时值相等之前最后一次写入{LSECRH[7:0], LSECRL[7:0]}的值）。定时器最小定时值为 1，若定时值设置为 0，定时器默认将 1 作为定时值。LSE 定时器定时时间计算公式如下：

$$\text{LSE 定时时间} = \frac{1}{32.768} \times (\{ \text{LSECRH}[7:0], \text{LSECRL}[7:0] \} + 1) \text{ ms}$$

LSE 定时器的 LSEEN、LSECNTEN、LSESTA 任意一位为 0，都将清零 LSE 的计数值。

8.4 中断与休眠唤醒

当 LSE 定时器的计数值等于定时值时，定时器中断标志位 LSEIF 置 1。若全局中断使能（EA=1），并且 LSE 定时器中断使能（LSEIE=1），CPU 将执行中断服务程序。

采用 LSE 定时中断唤醒休眠模式，需在休眠之前打开 LSEEN、LSECNT、LSEWUEN，并且设置后休眠状态到唤醒的时间 {LSECRH[7:0], LSECRL[7:0]}。若休眠之前打开全局中断使能和 LSE 中断使能，休眠唤醒后，将首先执行中断服务程序，中断返回之后再执行休眠指令的下一条指令。

9. 定时计数器 0/1 (Timer0/1)

定时器 0 与定时器 1 的类型和结构相似，是两个 16 位定时器。定时器 1 有三种工作模式，定时器 0 有四种工作模式。它们提供基本的定时和事件计数操作。

- 在“定时器模式”时，定时寄存器在定时器时钟使能时，每 12 个或 4 个系统周期递增一次。
- 在“计数器模式”时，定时寄存器每当检测到相应的输入引脚上 (T0 或 T1) 的下降沿就会增加。

9.1 概述

定时器 0 和定时器 1 与标准的 8051 定时器完全兼容。

每个定时器由两个 8 位寄存器：{TH0 (0x8C) :TL0 (0x8A)}和{TH1 (0x8D) :TL1 (0x8B)}组成。定时器 0、1 在四种相同的模式下工作。Timer0 和 Timer1 模式模式如下所述。

模式	M1	M0	功能描述
0	0	0	THx[7:0], TLx[4:0]组成 13 位定时/计数器
1	0	1	THx[7:0], TLx[7:0]组成 16 位定时/计数器
2	1	0	TLx[7:0]组成 8 位自动重装定时/计数器，从 THx 中重装
3	1	1	TL0, TH0 为两个 8 位定时/计数器，Timer1 停止计数

寄存器 THx 和 TLx 为特殊功能寄存器，具有储存实际定时器值的功能。THx 和 TLx 可以通过模式选项级联成 13 位或 16 位寄存器。每收到一个内部时钟脉冲或外部定时器引脚发生状态跳变时，寄存器的值加 1。定时器将从预置寄存器所载入的值开始计数直到定时器计满溢出，此时会产生一个内部中断信号。如果选为定时器的自动重载模式，定时器的值将重置为预载寄存器的初始值并继续计数，否则定时器的值将重置为零。注意，为了得到定时/计数器的最大计算范围，预置寄存器必须先被清零。

9.2 定时器 0/1 寄存器

9.2.1 定时器 0/1 模式寄存器 TMOD

0x89	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMOD	GATE1	CT1	T1M1	T1M0	GATE0	CT0	T0M1	T0M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	GATE1: 定时器1门控制位; 1= 使能; 0= 禁止。
Bit6	CT1: 定时器1定时/计数选择位; 1= 计数; 0= 定时。
Bit5~ Bit4	T1M1, T1M0: 定时器1模式选择位; 00= 模式0, 13位定时/计数器; 01= 模式1, 16位定时/计数器; 10= 模式2, 8位自动重装定时/计数器; 11= 模式3, 停止计数。
Bit3	GATE0: 定时器0门控制位; 1= 使能; 0= 禁止。
Bit2	CT0: 定时器0定时/计数选择位; 1= 计数; 0= 定时。
Bit1~ Bit0	T0M1, T0M0: 定时器0模式选择位; 00= 模式0, 13位定时/计数器; 01= 模式1, 16位定时/计数器; 10= 模式2, 8位自动重装定时/计数器; 11= 模式3, 两个独立的8位定时/计数器。

9.2.2 Timer0/1 控制寄存器 TCON, 可位寻址

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 TF1: Timer1 计数器溢出中断标志位;
 1= Timer1 计数器溢出, 进入中断服务程序硬件自动清零;
 0= Timer1 计数器无溢出。
- Bit6 TR1: Timer1 运行控制位;
 1= Timer1 启动;
 0= Timer1 关闭。
- Bit5 TF0: Timer0计数器溢出中断标志位;
 1= Timer0, 进入中断服务程序硬件自动清零;
 0= Timer0计数器无溢出。
- Bit4 TR0: Timer0运行控制位;
 1= Timer0启动;
 0= Timer0关闭。
- Bit3 IE1: 外部中断1标志位;
 1= 外部中断1产生中断, 进入中断服务程序硬件自动清零;
 0= 外部中断1没有产生中断。
- Bit2 IT1: 外部中断1触发方式控制位;
 1= 下降沿触发;
 0= 低电平触发。
- Bit1 IE0: 外部中断0标志位;
 1= 外部中断0产生中断, 进入中断服务程序硬件自动清零;
 0= 外部中断0没有产生中断。
- Bit0 IT0: 外部中断0触发方式控制位;
 1= 下降沿触发;
 0= 低电平触发。

9.2.3 定时器 0 数据寄存器低位 TL0

0x8A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit0 TL07-TL00: 定时器0低位数据寄存器 (同时作为计数器低位)。

9.2.4 定时器 0 数据寄存器高位 TH0

0x8C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TH07-TH00: 定时器0高位数据寄存器（同时作为计数器高位）。

9.2.5 定时器 1 数据寄存器低位 TL1

0x8B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TL17-TL10: 定时器1低位数据寄存器（同时作为计数器低位）。

9.2.6 定时器 1 数据寄存器高位 TH1

0x8D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TH17-TH10: 定时器1高位数据寄存器（同时作为计数器高位）。

9.2.7 功能时钟控制寄存器 CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	1	1	1

Bit7~Bit5 WTS2-WTS0: WDT溢出时间选择位;

000= $2^{17} \cdot T_{sys}$;

001= $2^{18} \cdot T_{sys}$;

010= $2^{19} \cdot T_{sys}$;

011= $2^{20} \cdot T_{sys}$;

100= $2^{21} \cdot T_{sys}$;

101= $2^{22} \cdot T_{sys}$;

110= $2^{24} \cdot T_{sys}$;

111= $2^{26} \cdot T_{sys}$ 。

Bit4 T1M: Timer1 的时钟源选择位;

0= $F_{sys}/12$;

1= $F_{sys}/4$ 。

Bit3 T0M: Timer0 的时钟源选择位;

0= $F_{sys}/12$;

1= $F_{sys}/4$ 。

Bit2~Bit0 未用。

9.3 定时器 0/1 中断

定时器 0/1 可以通过 IE 寄存器使能或关闭中断，还可以通过 IP 寄存器设置高/低优先级，其中断相关位如下：

中断屏蔽寄存器 IE (0xA8)

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	EA: 全局中断允许位; 1= 允许所有未被屏蔽的中断; 0= 禁止所有中断。
Bit6	ES1: UART1中断允许位; 1= 允许UART1中断; 0= 禁止UART1中断。
Bit5	ET2: TIMER2总中断允许位; 1= 允许TIMER2 所有中断; 0= 禁止TIMER2 所有中断。
Bit4	ES0: UART0中断允许位; 1= 允许UART0中断; 0= 禁止UART0中断。
Bit3	ET1: TIMER1中断允许位; 1= 允许TIMER1中断; 0= 禁止TIMER1中断。
Bit2	EX1: 外部中断1中断允许位; 1= 允许外部中断1中断; 0= 禁止外部中断1中断。
Bit1	ET0: TIMERO中断允许位; 1= 允许TIMERO中断; 0= 禁止TIMERO中断。
Bit0	EX0: 外部中断0中断允许位; 1= 允许外部中断0中断; 0= 禁止外部中断0中断。

中断优先级控制寄存器 IP

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	--
Bit6	PS1: UART1中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit5	PT2: TIMER2中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit4	PS0: UART0中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit3	PT1: TIMER1中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit2	PX1: 外部中断1中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit1	PT0: TIMER0中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit0	PX0: 外部中断0中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。

Timer0/1、INT0/1 中断标志位寄存器 TCON，可位寻址

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 TF1: Timer1 计数器溢出中断标志位;
 1= Timer1 计数器溢出, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= Timer1 计数器无溢出。
- Bit6 TR1: Timer1运行控制位;
 1= Timer1 启动;
 0= Timer1 关闭。
- Bit5 TF0: Timer0计数器溢出中断标志位;
 1= Timer0计数器溢出, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= Timer0计数器无溢出。
- Bit4 TR0: Timer0运行控制位;
 1= Timer0启动;
 0= Timer0关闭。
- Bit3 IE1: 外部中断1标志位;
 1= 外部中断1产生中断, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= 外部中断1没有产生中断。
- Bit2 IT1: 外部中断1触发方式控制位;
 1= 下降沿触发;
 0= 低电平触发。
- Bit1 IE0: 外部中断0标志位;
 1= 外部中断0产生中断, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= 外部中断0没有产生中断。
- Bit0 IT0: 外部中断0触发方式控制位;
 1= 下降沿触发;
 0= 低电平触发。

产生中断的标志位可以通过软件清零, 与通过硬件清零的结果相同。也就是说, 可以通过软件来产生中断 (不建议通过写标志位来产生中断) 或者取消待处理的中断。

TF0, TF1 标志位在没有开启中断的情况下, 可以通过写 0 清除。

9.4 定时器 0 工作模式

9.4.1 T0 -模式 0（13 位定时/计数模式）

在这种模式下，定时器 0 是 13 位的寄存器。当计数器所有的位由 1 向 0 翻转后，定时器 0 中断标志 TF0 置 1。当 TCON.4=1 且 TMOD.3=0 或 TCON.4=1，TMOD.3=1，T0G=1 时，计数输入使能到定时器 0。（设置 TMOD.3=1 允许定时器 0 由外部引脚 T0G 控制，以便脉宽测量）。13 位的寄存器由 TH0 和 TL0 低 5 位组成。TL0 高 3 位应忽略。

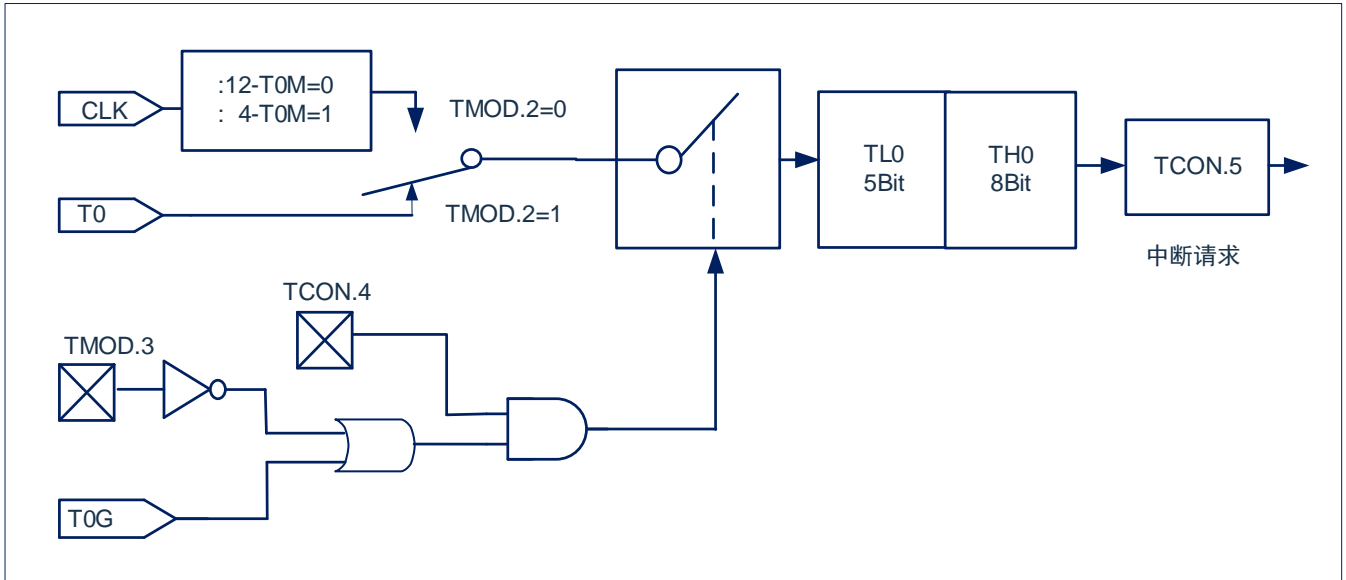


图 9-1: Timer0, 模式 0: 13-位定时/计数器

9.4.2 T0 -模式 1（16 位定时/计数模式）

模式 1 与模式 0 相同，只是模式 1 下定时器 0 数据寄存器 16 位全部运行。模式 1 如下图所示。

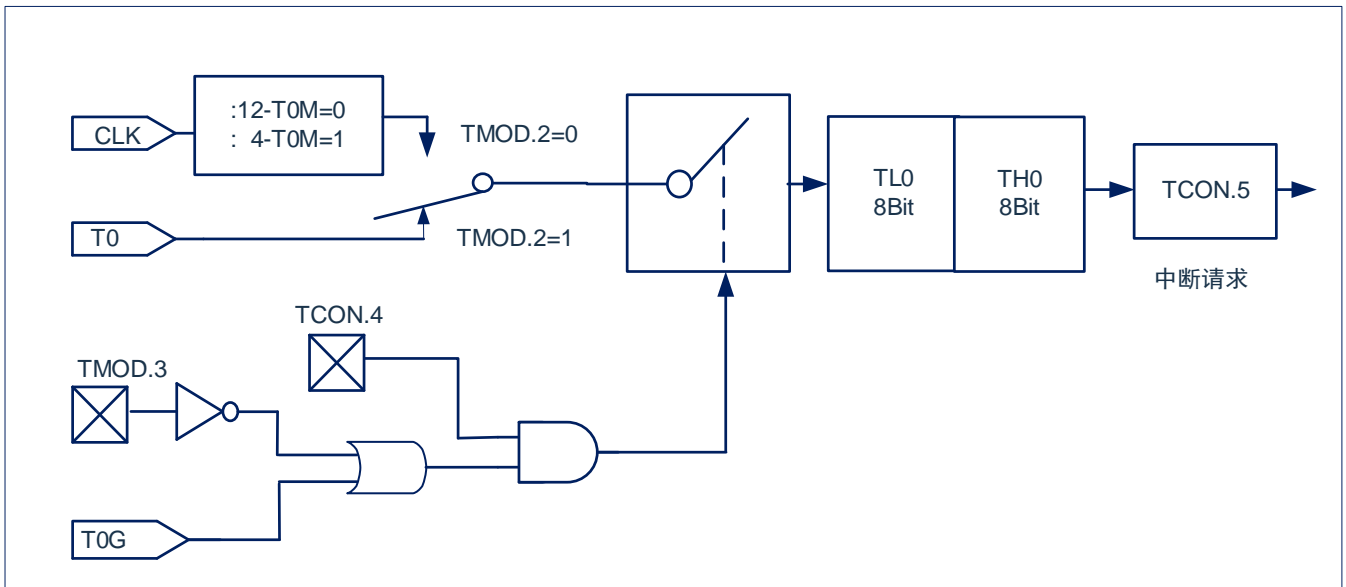


图 9-2: Timer0, 模式 1: 16-位定时/计数器

9.4.3 T0 -模式 2（8 位自动重装定时/计数模式）

模式 2 下定时器寄存器是一个配有自动重装模式的 8 位计数器（TL0），如下图所示。来自 TL0 的溢出不仅使 TF0 置 1，而且还将 TH0 的内容由软件重装到 TL0。重装过程中 TH0 的值保持不变。

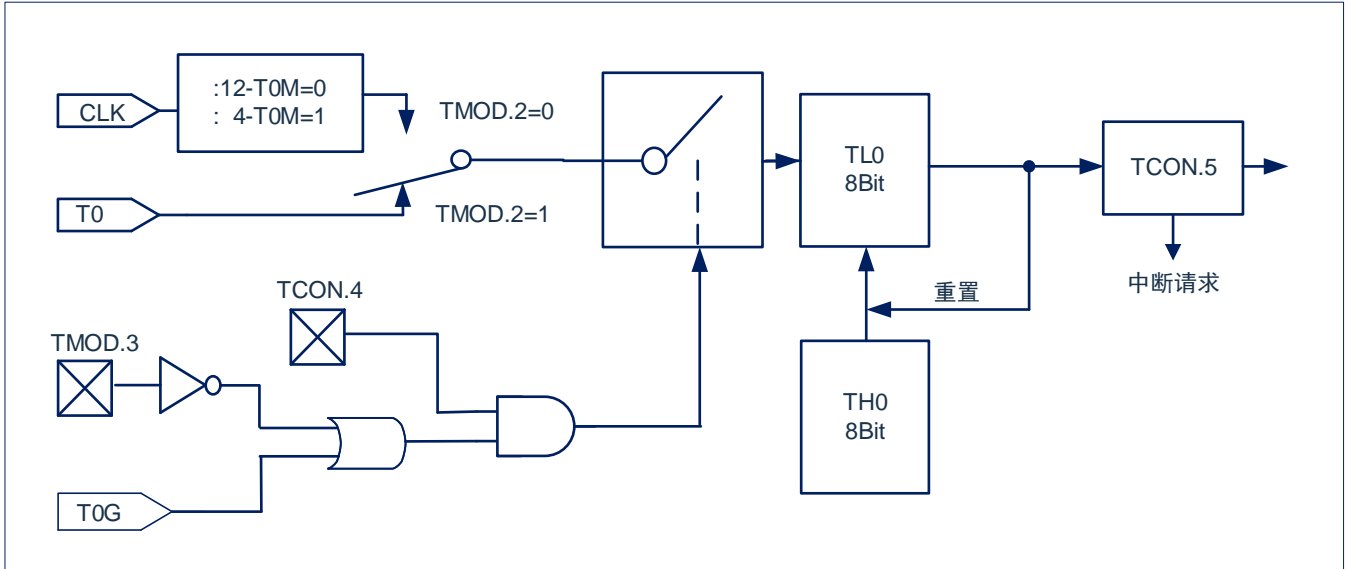


图 9-3: Timer0, 模式 2: 8-位定时/计数器（自动重装）

9.4.4 T0 -模式 3（两个单独的 8 位定时/计数器）

模式 3 下的定时器 0 将 TL0 和 TH0 设置为两个独立的计数器。定时器 0 模式 3 的逻辑如下图所示。

TL0 可作为定时器或计数器，并使用定时器 0 的控制位：如 CT0、TR0、GATE0 和 TF0。

TH0 只能作为定时器，且使用定时器 1 的 TR1 和 TF1 标志并控制定时器 1 的中断。

需要使用两个 8 位定时器/计数器时可以使用模式 3。当定时器 0 处于模式 3 时，定时器 1 可以通过切换到自己的模式 3 而将其关闭，或者仍然可以被串行通道用作波特率发生器，或在不需要定时器 1 中断的任何应用中。

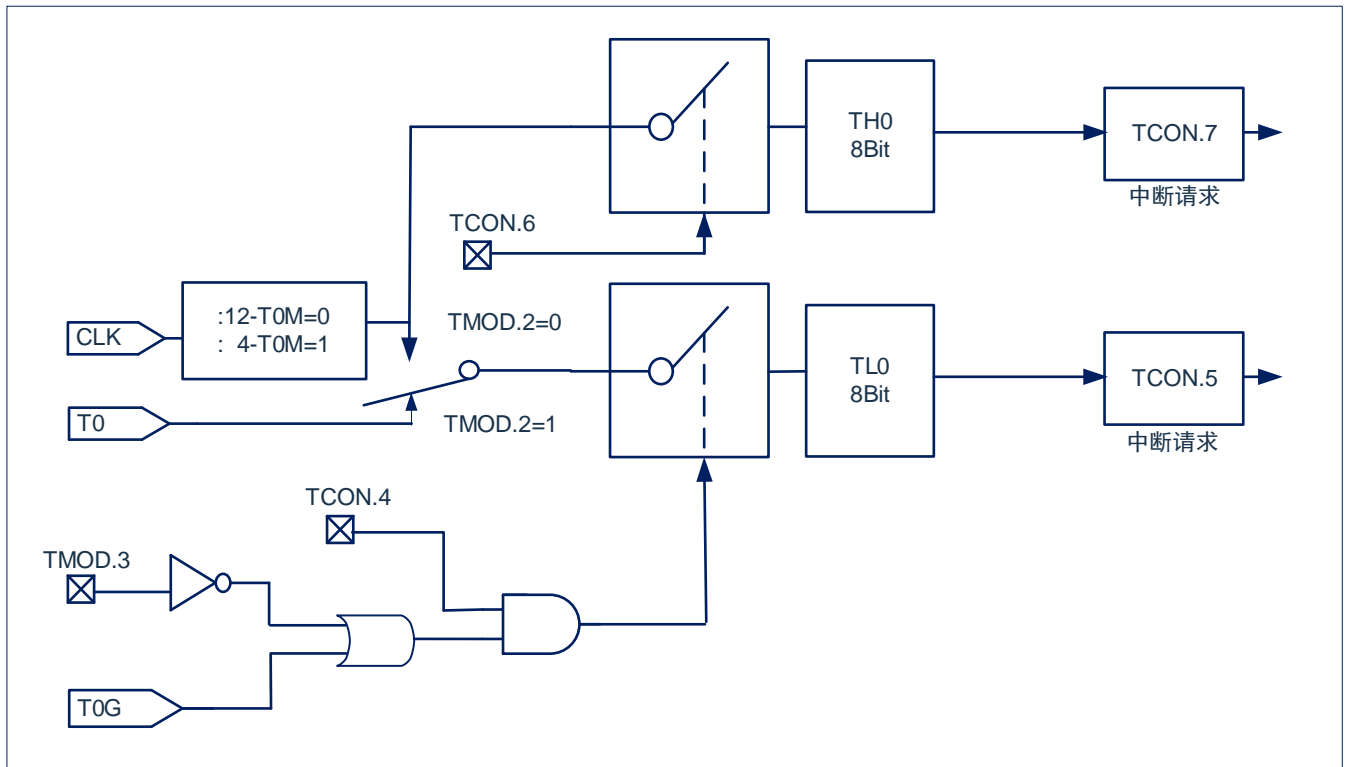


图 9-4：Timer0，模式 3：两个 8-位定时/计数器

9.5 定时器 1 工作模式

9.5.1 T1 -模式 0（13 位定时/计数模式）

在这种模式下，定时器 1 是 13 位的寄存器。当计数器所有的位由 1 向 0 翻转后，定时器 1 中断标志 TF1 置 1。当 TCON.6=1 且 TMOD.7=0 或当 TCON.6=1，TMOD.7=1 且 T1G=1 时，计数输入使能到定时器 1。（设置 TMOD.7=1 允许定时器 1 由外部引脚 T1G 控制，以便脉宽测量）。13 位的寄存器由 TH1 8 位和 TL1 低 5 位组成。TL1 高三位应忽略。

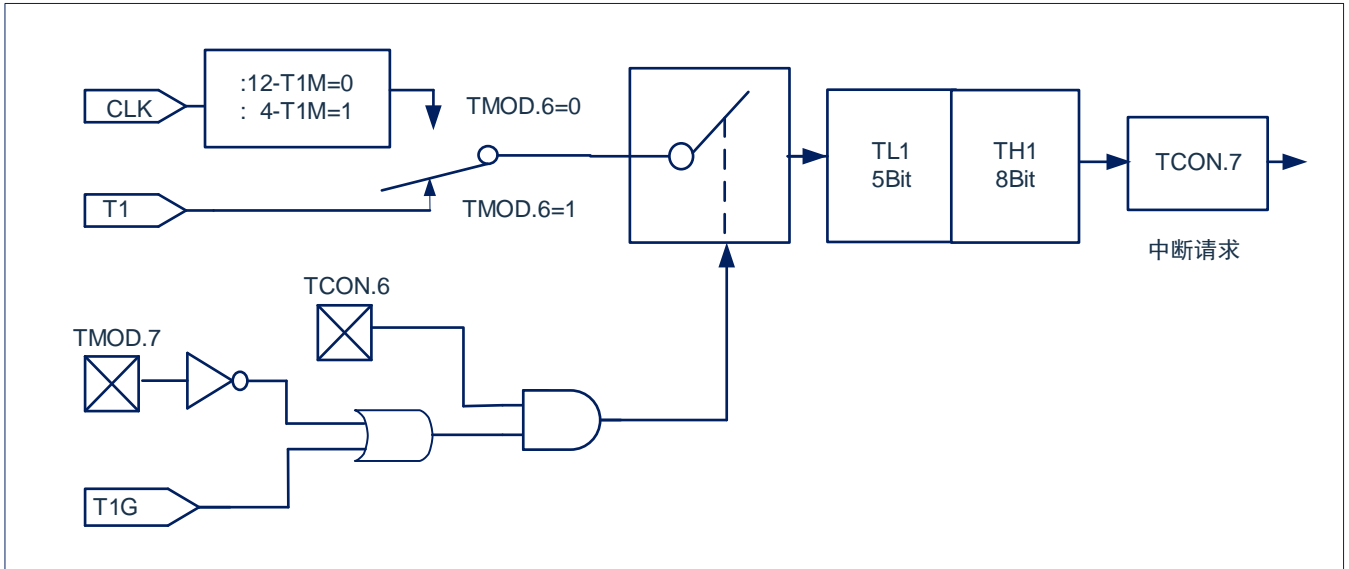


图 9-5: Timer1, 模式 0: 13-位定时/计数器

9.5.2 T1 -模式 1（16 位定时/计数模式）

模式 1 与模式 0 相同，只是模式 1 下定时器 1 寄存器 16 位全部运行。模式 1 如下图所示。

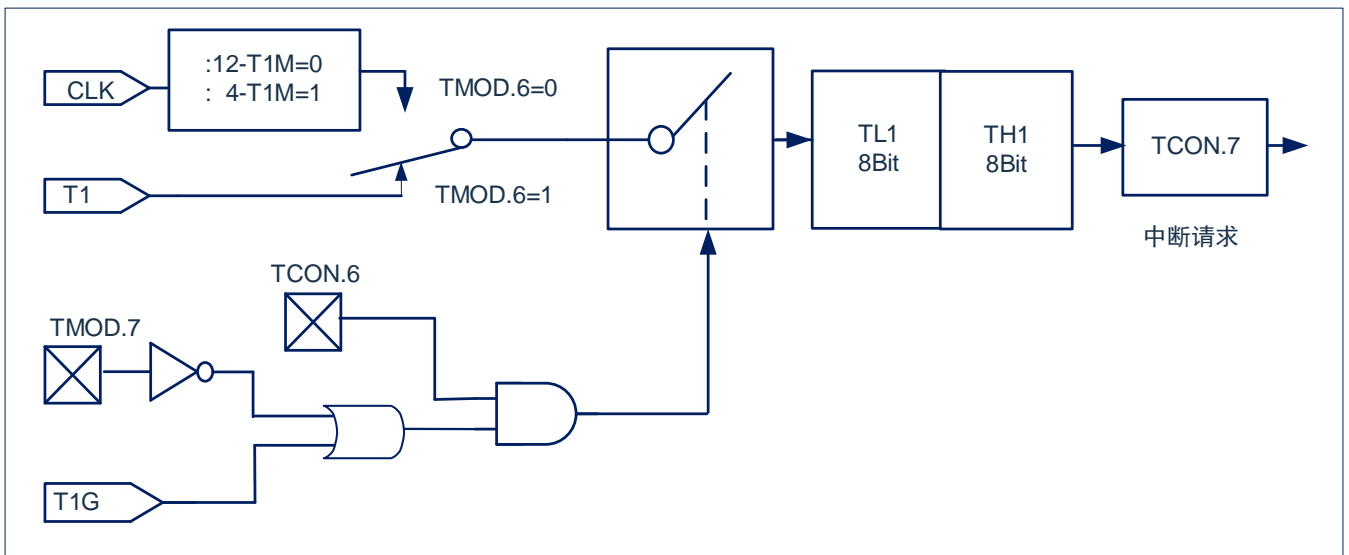


图 9-6: Timer1, 模式 1: 16-位定时/计数器

9.5.3 T1 -模式 2（8 位自动重装定时/计数模式）

模式 2 下定时器 1 寄存器是一个配有自动重装模式的 8 位计数器（TL1），如下图所示。来自 TL1 的溢出不仅使 TF1 置 1，而且还将 TH1 的内容由软件重装到 TL1。重装过程中 TH1 的值保持不变。

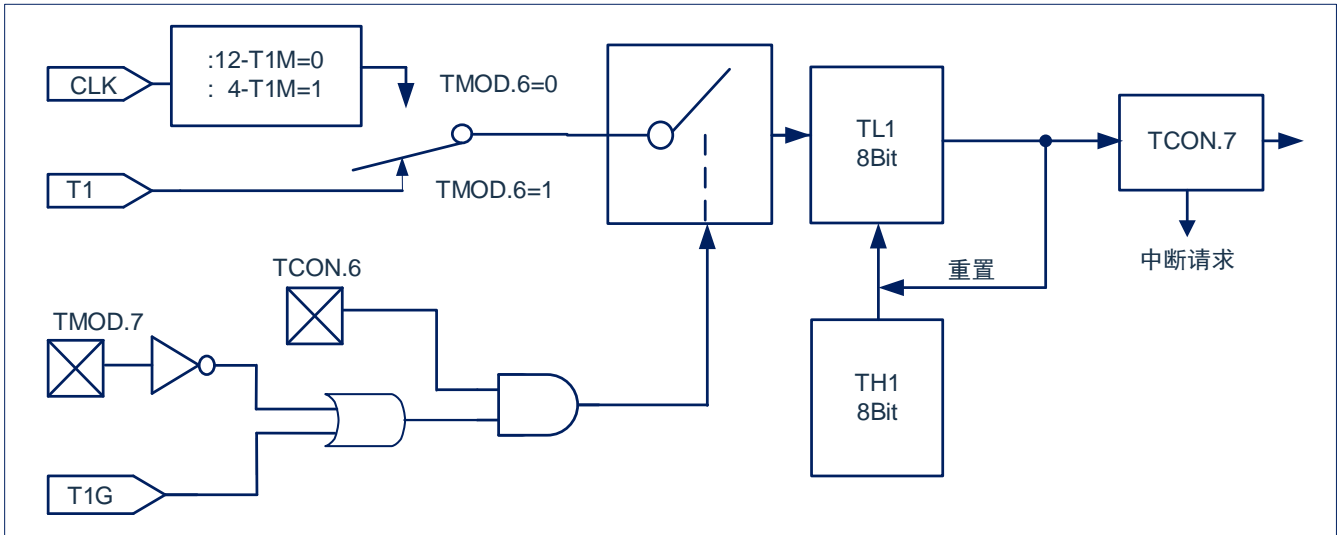


图 9-7: Timer1, 模式 2: 8-位定时/计数器（自动重装）

9.5.4 T1 -模式 3（停止计数）

模式 3 中的定时器 1 停止计数，其效果与设置 TR1=0 相同。

10. 定时计数器 2 (Timer2)

具有额外的比较/捕获/重装功能的定时器 2 是最核心的外设单元之一。它可以用于各种数字信号的生成和事件捕获，比如脉冲发生、脉冲宽度调制、脉宽测量等。

10.1 概述

下面框图显示了具有附加比较/捕获/重装寄存器功能的定时器 2 的一般配置。

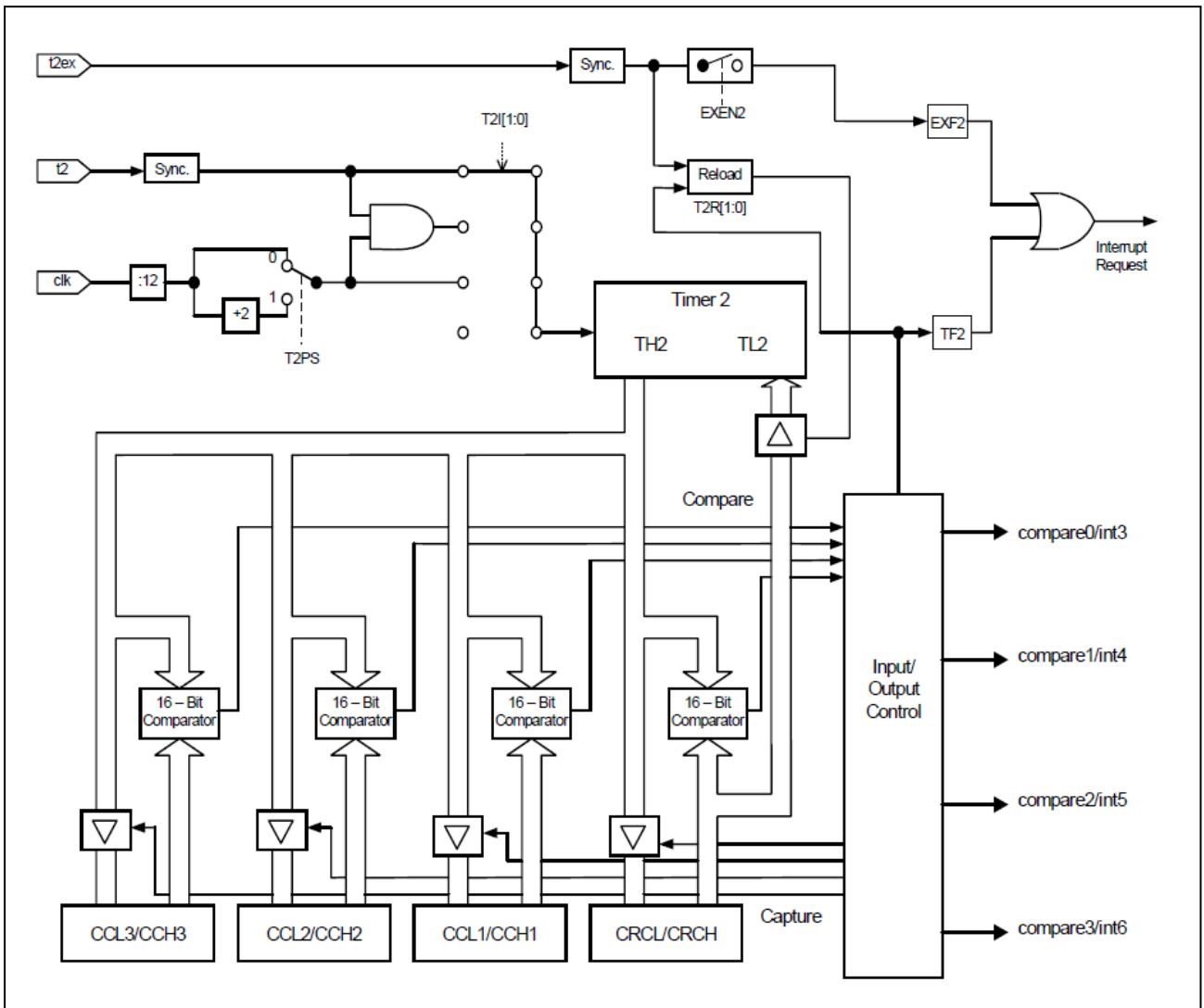


图 10-1: 比较/捕获模块

10.2 Timer2 寄存器

具有比较/捕获寄存器功能的定时器 2 的寄存器如下：

10.2.1 T2 控制寄存器 T2CON

0xC8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	T2PS	I3FR	CAPES	T2R1	T2R0	T2CM	T2I1	T2I0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	T2PS: Timer2 时钟预分频选择位； 1= Fsys/24； 0= Fsys/12。
Bit6	I3FR: 捕获通道0输入单边沿选择与比较中断时刻选择位； 捕获通道0模式： 1= 上升沿捕获到RLDL/RLDH寄存器； 0= 下降沿捕获到RLDL/RLDH寄存器。 比较通道0模式： 1= TL2/TH2与RLDL/RLDH从不相等到相等时刻产生中断； 0= TL2/TH2与RLDL/RLDH从相等到不相等时刻产生中断；
Bit5	CAPES: 捕获通道1-3输入单边沿选择（对捕获通道1-3一起生效）。 0= 上升沿捕获到CCL1/CCH1-CCL3/CCH3寄存器； 1= 下降沿捕获到CCL1/CCH1-CCL3/CCH3寄存器。
Bit4~Bit3	T2R1-T2R0: Timer2 加载模式选择位； 0x= 重装禁止； 10= 加载模式1：Timer2溢出时自动重装； 11= 加载模式2：在T2EX引脚下降沿重装。
Bit2	T2CM: 比较模式选择； 1= 比较模式1； 0= 比较模式0。
Bit1~ Bit0	T2I1-T2I0: Timer2时钟输入选择位； 00= Timer2停止； 01= 系统时钟的分频（由T2PS控制分频选择）； 10= 外部引脚T2作事件输入（事件计数模式） 11= 外部引脚T2作门控输入（门控定时模式）

10.2.2 定时器 2 数据寄存器低位 TL2

0xCC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL2	TL27	TL26	TL25	TL24	TL23	TL22	TL21	TL20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TL27-TL20: 定时器2低位数据寄存器（同时作为计数器低位）。

10.2.3 定时器 2 数据寄存器高位 TH2

0xCD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH2	TH27	TH26	TH25	TH24	TH23	TH22	TH21	TH20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TH27-TH20: 定时器2高位数据寄存器（同时作为计数器低位）。

10.2.4 定时器 2 比较/捕获/自动重装寄存器低位 RLDL

0xCA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RLDL	RLDL7	RLDL6	RLDL5	RLDL4	RLDL3	RLDL2	RLDL1	RLDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 RLDL7- RLDL0: 定时器2比较/捕获/自动重装寄存器低位。

10.2.5 定时器 2 比较/捕获/自动重装寄存器高位 RLDH

0xCB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RLDH	RLDH7	RLDH6	RLDH5	RLDH4	RLDH3	RLDH2	RLDH1	RLDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 RLDH7- RLDH0: 定时器2比较/捕获/自动重装寄存器高位。

10.2.6 定时器 2 比较/捕获通道 1 寄存器低位 CCL1

0xC2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCL1	CCL17	CCL16	CCL15	CCL14	CCL13	CCL12	CCL11	CCL10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 CCL17-CCL10: 定时器2比较/捕获通道1寄存器低位。

10.2.7 定时器 2 比较/捕获通道 1 寄存器高位 CCH1

0xC3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCH1	CCH17	CCH16	CCH15	CCH14	CCH13	CCH12	CCH11	CCH10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 CCH17-CCH10: 定时器2比较/捕获通道1寄存器高位。

10.2.8 定时器 2 比较/捕获通道 2 寄存器低位 CCL2

0xC4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCL2	CCL27	CCL26	CCL25	CCL24	CCL23	CCL22	CCL21	CCL20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 CCL27-CCL20: 定时器2比较/捕获通道2寄存器低位。

10.2.9 定时器 2 比较/捕获通道 2 寄存器高位 CCH2

0xC5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCH2	CCH27	CCH26	CCH25	CCH24	CCH23	CCH22	CCH21	CCH20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 CCH27-CCH20: 定时器2比较/捕获通道2寄存器高位。

10.2.10 定时器 2 比较/捕获通道 3 寄存器低位 CCL3

0xC6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCL3	CCL37	CCL36	CCL35	CCL34	CCL33	CCL32	CCL31	CCL30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 CCL37-CCL30: 定时器2比较/捕获通道3寄存器低位。

10.2.11 定时器 2 比较/捕获通道 3 寄存器高位 CCH3

0xC7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCH3	CCH37	CCH36	CCH35	CCH34	CCH33	CCH32	CCH31	CCH30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 CCH37-CCH30: 定时器2比较/捕获通道3寄存器高位。

10.2.12 T2 比较捕获控制寄存器 CCEN

0xCE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCEN	CMH3	CML3	CMH2	CML2	CMH1	CML1	CMH0	CML0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7~Bit6 CMH3-CML3: 捕获/比较模式控制位;
 00= 捕获/比较禁止;
 01= 捕获操作在通道3的上升沿或下降沿触发(CAPES选择);
 10= 比较模式使能;
 11= 捕获操作在写CCL3时触发或通道3的双沿触发。
- Bit5~Bit4 CMH2-CML2: 捕获/比较模式控制位;
 00= 捕获/比较禁止;
 01= 捕获操作在通道2的上升沿或下降沿触发(CAPES选择);
 10= 比较模式使能;
 11= 捕获操作在写CCL2时触发或通道2的双沿触发。
- Bit3~Bit2 CMH1-CML1: 捕获/比较模式控制位;
 00= 捕获/比较禁止;
 01= 捕获操作在通道1的上升沿或下降沿触发(CAPES选择);
 10= 比较模式使能;
 11= 捕获操作在写CCL1时触发或通道1的双沿触发。
- Bit1~Bit0 CMH0-CML0: 捕获/比较模式控制位;
 00= 捕获/比较禁止;
 01= 捕获操作在通道0的上升沿或下降沿触发(I3FR选择);
 10= 比较模式使能;
 11= 捕获操作在写RLDL时触发或通道0的双沿触发。

10.3 Timer2 中断

定时器 2 可以通过 IE 寄存器使能或关闭中断, 还可以通过 IP 寄存器设置高/低优先级, 其中断相关位如下:
中断屏蔽寄存器 IE (0xA8)

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 EA: 全局中断允许位;
 1= 允许所有未被屏蔽的中断;
 0= 禁止所有中断。
- Bit6 ES1: UART1中断允许位;
 1= 允许UART1中断;
 0= 禁止UART1中断。
- Bit5 ET2: TIMER2 总中断允许位;
 1= 允许TIMER2所有中断;
 0= 禁止TIMER2所有中断。
- Bit4 ES0: UART0中断允许位;
 1= 允许UART0中断;
 0= 禁止UART0中断。
- Bit3 ET1: TIMER1中断允许位;
 1= 允许TIMER1中断;
 0= 禁止TIMER1中断。
- Bit2 EX1: 外部中断1中断允许位;
 1= 允许外部中断1中断;
 0= 禁止外部中断1中断。
- Bit1 ET0: TIMER0中断允许位;
 1= 允许TIMER0中断;
 0= 禁止TIMER0中断。
- Bit0 EX0: 外部中断0中断允许位;
 1= 允许外部中断0中断;
 0= 禁止外部中断0中断。

中断优先级控制寄存器 IP

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 --
- Bit6 PS1: UART1中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit5 PT2: TIMER2中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit4 PS0: UART0 中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit3 PT1: TIMER1中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit2 PX1: 外部中断1中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit1 PT0: TIMER0中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit0 PX0: 外部中断0中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。

Timer2 中断屏蔽寄存器 T2IE

0xCF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IE	T2OVIE	T2EXIE	--	--	T2C3IE	T2C2IE	T2C1IE	T2C0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 T2OVIE: Timer2溢出中断允许位;
 1= 允许中断;
 0= 禁止中断。
- Bit6 T2EXIE: Timer2外部加载中断允许位;
 1= 允许中断;
 0= 禁止中断。
- Bit5~Bit4 --
- Bit3 T2C3IE: Timer2比较通道3中断允许位;
 1= 允许中断;
 0= 禁止中断。
- Bit2 T2C2IE: Timer2比较通道2中断允许位;
 1= 允许中断;
 0= 禁止中断。
- Bit1 T2C1IE: Timer2比较通道1中断允许位;
 1= 允许中断;
 0= 禁止中断。
- Bit0 T2C0IE: Timer2比较通道0中断允许位;
 1= 允许中断;
 0= 禁止中断。

若开启 Timer2 的中断，还需要开启 Timer2 的总中断使能位 ET2=1 (IE.5=1)。

Timer2 中断标志位寄存器 T2IF (0xC9)

0xC9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IF	TF2	T2EXIF	--	--	T2C3IF	T2C2IF	T2C1IF	T2C0IF
R/W	R/W	R/W	--	--	R/W	R/W	R/W	R/W
复位值	0	0	--	--	0	0	0	0

- Bit7 TF2: Timer2 计数器溢出中断标志位;
 1= Timer2 计数器溢出, 需软件清零;
 0= Timer2 计数器无溢出。
- Bit6 T2EXIF: Timer2 外部加载标志位;
 1= Timer2 的T2EX口产生下降沿, 需软件清零;
 0= --
- Bit5 --
- Bit4 --
- Bit3 T2C3IF: Timer2 比较/捕获通道3标志位;
 1= Timer2 比较通道3 (CCH3:CCL3)={TH2:TL2}或捕获通道3产生了捕获操作, 需软件清零。
 0= --
- Bit2 T2C2IF: Timer2 比较/捕获通道2标志位;
 1= Timer2 比较通道2 (CCH2:CCL2)={TH2:TL2}或捕获通道2产生了捕获操作, 需软件清零。
 0= --
- Bit1 T2C1IF: Timer2 比较/捕获通道1标志位;
 1= Timer2 比较通道1 (CCH1:CCL1)={TH2:TL2}或捕获通道1产生了捕获操作, 需软件清零。
 0= --
- Bit0 T2C0IF: Timer2 比较/捕获通道0标志位;
 1= Timer2 比较通道0 (RLDH:RLDL)={TH2:TL2}或捕获通道0产生了捕获操作, 需软件清零。
 0= --

10.3.1 定时中断

当 Timer2 定时器溢出时，会产生中断。TF2 将会置 1。

10.3.2 外部触发中断

当 T2EXIE 位置 1 时，T2EX 引脚的下降沿也会产生中断。T2EXIF 将会置 1。

10.3.3 比较中断

4 个比较通道均支持比较中断。

比较通道 0 可选择比较中断产生的时刻，产生中断则会将比较通道 0 的中断标志 T2C0IF 置 1。

I3FR = 0 时，TL2/TH2 与 RLDL/RLDH 从不相等到相等时刻产生中断；

I3FR = 1 时，TL2/TH2 与 RLDL/RLDH 从相等到不相等时刻产生中断；

比较通道 1-3 不能选择中断产生时刻，固定为 TL2/TH2 与 CCxL/CCxH 从不相等到相等时刻产生中断。如果产生中断则会将相应的比较通道 x 的中断标志 T2CxIF 置 1。

Timer2 所有相关中断均共用一个中断向量，进入中断服务程序后，需要判断相关标志位才能确定是哪一种情况产生了中断。

10.3.4 捕获中断

4 个捕获通道均支持外部捕获中断。产生捕获操作时会将相应捕获通道的中断标志为 T2CxIF 置 1。

注意写操作捕获方式不会产生中断。

10.4 Timer2 功能说明

定时器 2 是一个 16 位宽的寄存器，可以作为定时器、事件计数器或门控定时器运行。

10.4.1 T2 定时模式

用作定时器功能时，时钟源来自系统频率。预分频器提供为 1/12 或 1/24 的系统频率选择。因此，16 位定时器寄存器(由 TH2 和 TL2 组成)每 12 个时钟周期或每 24 个时钟周期递增。预分频器的值由 T2CON 的 T2PS 位选择。

10.4.2 T2 重装模式

定时器 2 的重装模式由 T2CON 的 T2R0 和 T2R1 位选择。

在加载模式 1 中，当计数器 2 由所有的 1 向 0 翻转后（计数器溢出），不仅 TF2 置 1，而且定时器 2 寄存器自动加载来自 RLDL/RLDH 寄存器的 16 位值。所需的 RLDL/RLDH 值可由软件预设。重载发生在 TF2 置 1 的相同的时钟周期中，从而覆盖计数值 0x0000。

在加载模式 2 中，来自 RLDL/RLDH 寄存器的 16 位重装操作由相应 T2EX 输入引脚的下降沿触发。此外，若 T2EXIE=1，则 T2EXIF 标志将会置 1。如果定时器 2 总中断使能（ET2=1），则产生中断。

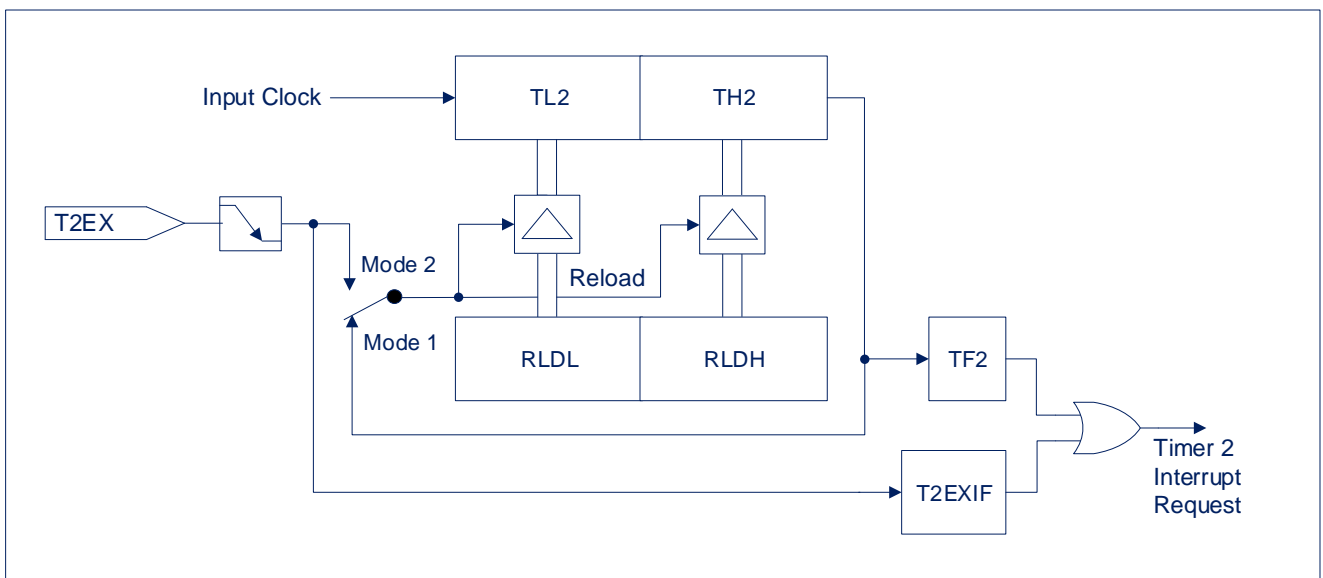


图 10-2: Timer2 重装结构

10.4.3 Timer2 门控定时模式

用作门控定时器功能，则外部输入引脚 T2 作为定时器 2 的输入门控。如果 T2 引脚为高电平，则内部时钟输入被选通到定时器。T2 引脚为低电平则会终止计数。该功能常被用来测量脉冲宽度。

10.4.4 Timer2 事件计数模式

在计数器功能中时，定时器 2 随着其对应的外部输入引脚 T2 上的 1 到 0 的转换而递增。在此功能中，对外部输入在每个系统时钟周期进行采样。当采样输入在一个周期内显示高电平，在下一个周期内显示为低电平时，计数增加。在接下来的周期再检测到 T2 引脚高到低的变化，则新的计数值出现在定时器寄存器中。

10.5 比较功能/PWM 功能

定时器/寄存器组合的比较功能可以描述如下：

存储在比较/捕获寄存器中的 16 位值与定时寄存器的值进行比较。如果定时器寄存器中的计数值与存储值相匹配，则在相应的端口引脚上会产生输出信号的跳变，并产生中断请求标志。

这种可产生周期性信号且可改变占空比控制方式常被用于脉冲宽度调制（PWM）以及需要产生连续方波的控制场合。

比较功能（PWM）输出的通道为 CC0、CC1、CC2、CC3。分别对应 16 位比较寄存器{RLDH, RLDL}、{CCH1, CCL1}、{CCH2, CCL2}、{CCH3, CCL3}与 Timer2 的{TH2, TL2}寄存器比较结果的输出信号。

比较功能包含两种模式：比较模式 0 和 1 由特殊功能寄存器 T2CON 中的 T2CM 位选择。这两种比较模式涵盖了较为广泛的应用。

10.5.1 比较模式 0

在模式 0 中，在匹配定时器和比较寄存器内容后，输出信号从低电平变为高电平。定时器溢出时，输出信号变为低电平。下图显示了比较模式 0 的功能图。比较输出通道由两个事件直接控制：定时器溢出和比较操作。

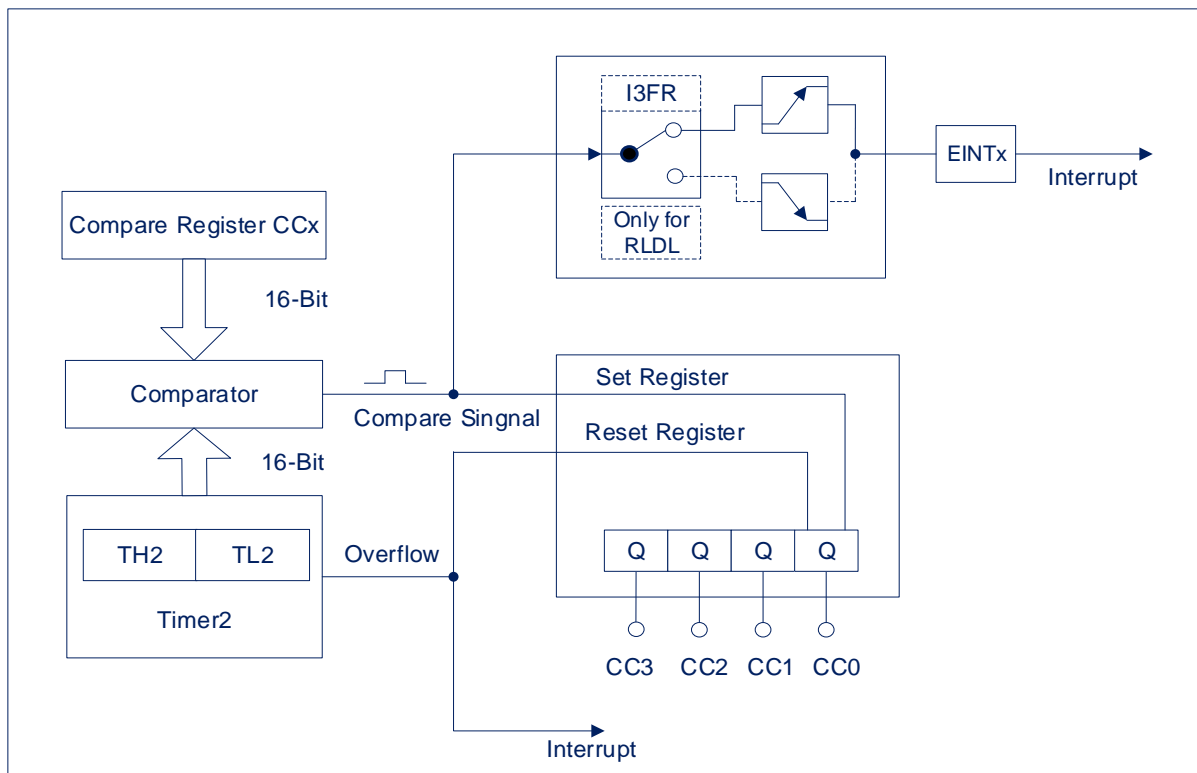


图 10-3: Timer2 比较模式 0 结构

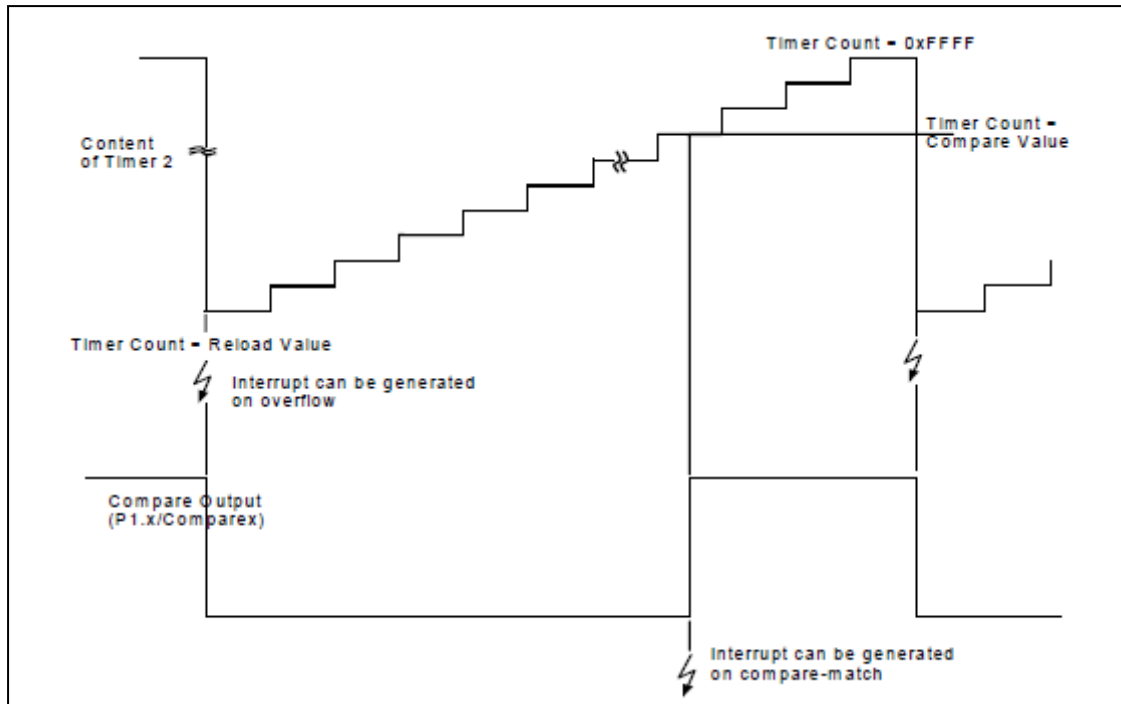


图 10-4: Timer2 比较模式 0 的寄存器及输出

10.5.2 比较模式 1

在比较模式 1 中，通常使用在输出信号与恒定信号周期无关时的，软件自适应地确定输出信号的跳变。在比较模式 1 中，可以控制信号的两个转移。在此模式下比较输出可认为是不依赖于 CPU 运行的高速输出。

如果模式 1 使能，软件将写入到 CCx 端口的相应的输出寄存器，在下一个比较匹配发生之前，新的值不会出现在输出引脚上。当定时器 2 计数器与存储的比较值相匹配时，用户可以选择输出信号是进行新的改变还是保持其旧值这两种方式其一。下图显示了定时器 2 比较模式 1 中的功能框图。

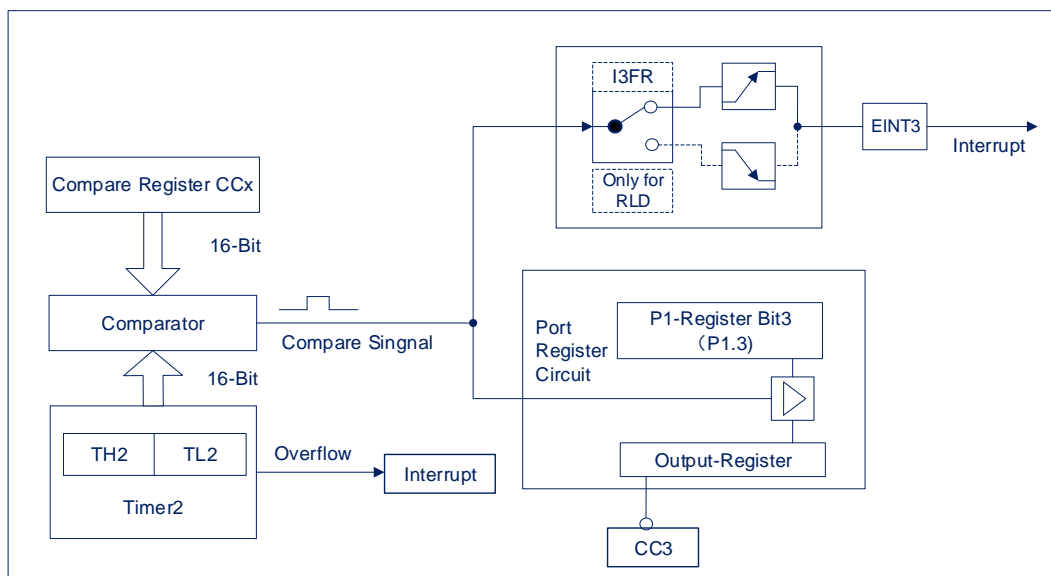


图 10-5: Timer2 比较模式 1 的结构

注：比较模式 1 仅仅对比较通道 3 有效。

10.6 捕获功能

4 个作为捕获功能的 16 位寄存器{RLDH, RLDL}、{CCH1, CCL1}、{CCH2, CCL2}、{CCH3, CCL3} 中的每一组寄存器都可用于锁存{TH2, TL2}的当前 16 位值。该功能提供了两种不同的捕获模式。

在模式 0 中，一个外部事件可将定时器 2 的内容锁存到捕获寄存器中。

在模式 1 中，捕获操作发生在写入 16 位捕获寄存器的低位字节（RLDL/CCL1/CCL2/CCL3）时。这种模式允许软件在运行时读取{TH2, TL2}的内容。

捕获通道 0-3 分别选择捕获输入引脚 CAP0-CAP3 的作为输入源信号。以下为相应捕获通道的控制方式：

捕获通道	输入引脚
0	CAP0
1	CAP1
2	CAP2
3	CAP3

10.6.1 捕获模式 0

该模式下导致捕获的外部事件为：

- 对于 CC 寄存器 1 到 3：捕获通道 1-3 上的正跳变或负跳变，也可支持正负跳变。
- 对于 RLD 寄存器：捕获通道 0 上的正跳变或负跳变，也可支持正负跳变。

捕获通道 0 上的是正跳变触发捕获操作还是负跳变触发捕获操作取决于 T2CON 的 I3FR 位。I3FR=0，负跳变触发捕获；I3FR=1，正跳变触发捕获。

捕获通道 1-3 上的是正跳变触发捕获操作还是负跳变触发捕获操作取决于 T2CON 的 CAPES 位。CAPES=0，正跳变触发捕获；CAPES =1，负跳变触发捕获。捕获通道 1-3 的选择的跳变方式是相同的

另外捕获通道 0-3 同时支持双跳变的捕获操作，将 CCEN 寄存器相应的工作模式控制位选择为 11，则该通道支持双跳变的捕获操作。需要注意的是，该工作模式下也支持捕获模式 1，即写操作可产生捕获动作。

通道 0-3 的外部捕获事件均可产生中断。

10.6.2 捕获模式 1

该模式下捕获操作事件为写捕获寄存器的低位字节指令的执行。写寄存器信号（例如写 RLDL）启动捕获操作，且写入的值与此功能无关。在写指令执行后，定时器 2 的内容将被锁存到相应的捕获寄存器中。在此模式下，不会产生中断请求标志。

下图显示了定时器 2 捕获功能。

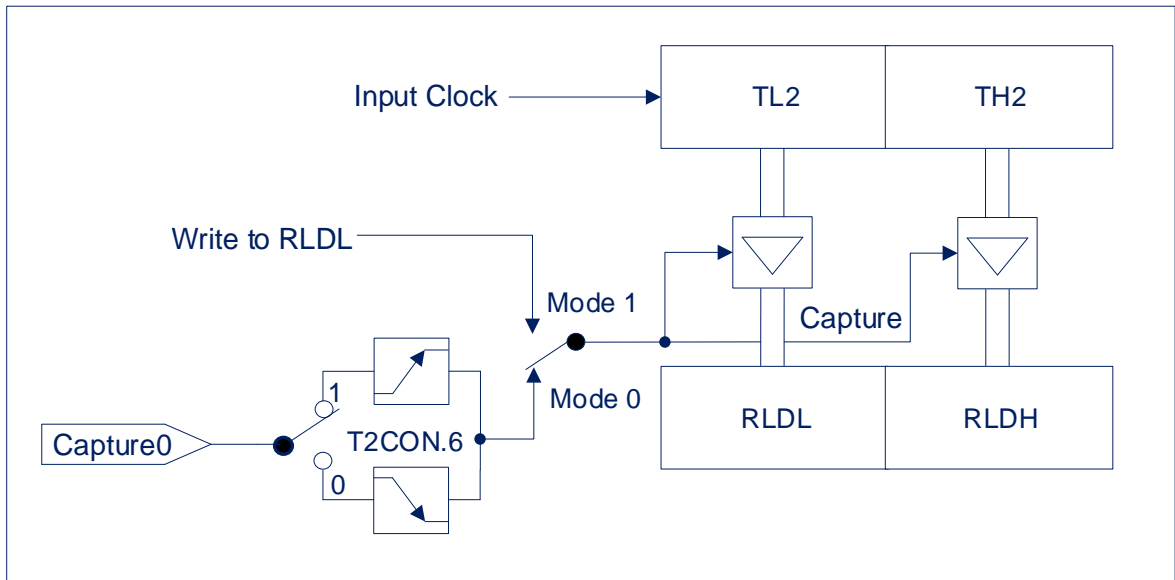


图 10-6: Timer2 捕获模式对应 RLDL 与 RLDH 寄存器

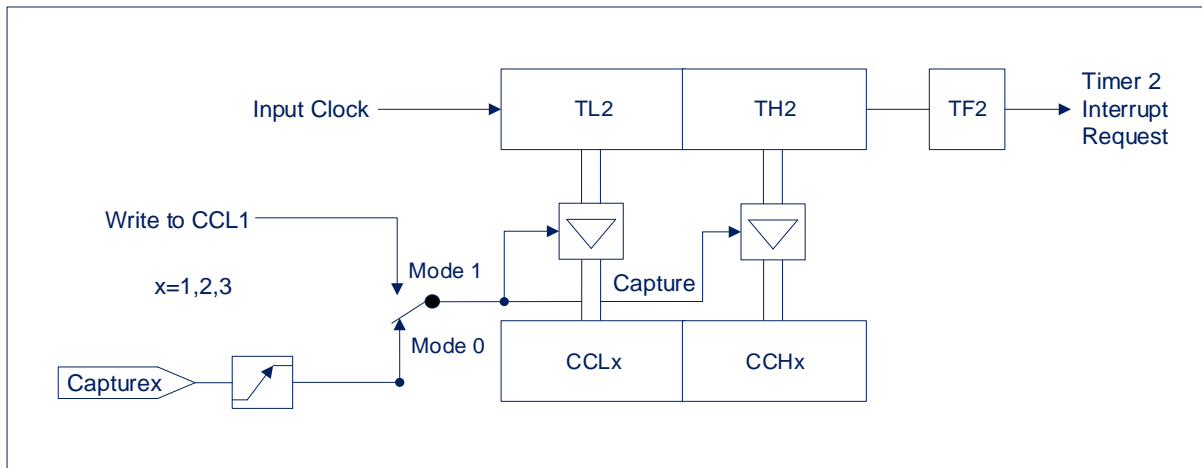


图 10-7: Timer2 捕获模式对应 CCLx 与 CCHx 寄存器

11. 定时器 3/4 (Timer3/4)

定时器 3/4 与定时器 0/1 的相似，是两个 16 位定时器。定时器 3 有四种工作模式，定时器 4 有三种工作模式。与 Timer0/1 相比，Timer3/4 仅提供定时操作。

在定时器启动的情况下，寄存器的值每 12 个或 4 个系统周期递增一次。

11.1 概述

定时器 3 和定时器 4 由两个 8 位寄存器{TH3, TL3}和{TH4, TL4}组成。定时器 3, 4 在四种相同的模式下工作。Timer3 和 Timer4 模式如下所述：

模式	M1	M0	功能描述
0	0	0	THx[7:0], TLx[4:0]组成 13 位定时器
1	0	1	THx[7:0], TLx[7:0]组成 16 位定时器
2	1	0	TLx[7:0]组成 8 位自动重装定时器，从 THx 中重装
3	1	1	TL3, TH3 为两个 8 位定时器，Timer4 停止计数

11.2 定时器 3/4 寄存器

11.2.1 定时器 3/4 控制寄存器 T34MOD

0xD2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T34MOD	TR4	T4M	T4M1	T4M0	TR3	T3M	T3M1	T3M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	TR4: Timer4运行控制位; 1= Timer4启动; 0= Timer4关闭。
Bit6	T4M: 定时器4时钟选择位; 1= Fsys/4; 0= Fsys/12。
Bit5~Bit4	T4M1, T4M0: 定时器4模式选择位; 00= 模式0, 13位定时器; 01= 模式1, 16位定时器; 10= 模式2, 8位自动重装定时器; 11= 模式3, 停止计数。
Bit3	TR3: Timer3运行控制位; 1= Timer3启动; 0= Timer3关闭。
Bit2	T3M: 定时器3时钟选择位; 1= Fsys/4; 0= Fsys/12。
Bit1~Bit0	T3M1, T3M0: 定时器3模式选择位; 00= 模式0, 13位定时器; 01= 模式1, 16位定时器; 10= 模式2, 8位自动重装定时器; 11= 模式3, 两个独立的8位定时器。

11.2.2 定时器 3 数据寄存器低位 TL3

0xDA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL3	TL37	TL36	TL35	TL34	TL33	TL32	TL31	TL30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TL37-TL30: 定时器3低位数据寄存器（同时作为定时器低位）。

11.2.3 定时器 3 数据寄存器高位 TH3

0xDB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH3	TH37	TH36	TH35	TH34	TH33	TH32	TH31	TH30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TH37-TH30: 定时器3高位数据寄存器（同时作为定时器高位）。

11.2.4 定时器 4 数据寄存器低位 TL4

0xE2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL4	TL47	TL46	TL45	TL44	TL43	TL42	TL41	TL40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TL47-TL40: 定时器4低位数据寄存器（同时作为定时器低位）。

11.2.5 定时器 4 数据寄存器高位 TH4

0xE3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH4	TH47	TH46	TH45	TH44	TH43	TH42	TH41	TH40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 TH47-TH40: 定时器4高位数据寄存器（同时作为定时器高位）。

11.3 定时器 3/4 中断

定时器 3/4 可以通过 IE 寄存器使能或关闭中断，还可以通过 IP 寄存器设置高/低优先级，其中断相关位如下：

中断屏蔽寄存器 EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 SPIIE: SPI中断使能位；
 1= 允许SPI中断；
 0= 禁止SPI中断。
- Bit6 I2CIE I²C中断使能位；
 1= 允许I²C中断；
 0= 禁止I²C中断。
- Bit5 WDTIE: WDT中断使能位；
 1= 允许WDT溢出中断；
 0= 禁止WDT溢出中断。
- Bit4 ADCIE: ADC中断使能位；
 1= 允许ADC中断；
 0= 禁止ADC中断。
- Bit3 PWMIE: PWM总中断使能位；
 1= 允许PWM所有中断；
 0= 禁止PWM所有中断。
- Bit2 --
- Bit1 ET4: Timer4中断使能位；
 1= 允许Timer4中断；
 0= 禁止Timer4中断。
- Bit0 ET3: Timer3中断使能位；
 1= 允许Timer3中断；
 0= 禁止Timer3中断。

外设中断标志位寄存器 EIF2 (0xB2)

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 SPIIF: SPI总中断指示位, 只读;
 1= SPI产生中断, (清除具体的中断标志位后, 此位自动清除);
 0= SPI未产生中断。
- Bit6 I2CIF: I²C总中断指示位, 只读;
 1= I²C产生中断, 需软件清零 (清除具体的中断标志位后, 此位自动清除);
 0= I²C未产生中断。
- Bit5 --
- Bit4 ADCIF: ADC中断标志位;
 1= ADC转换完成, 需软件清零;
 0= ADC转换未完成。
- Bit3 PWMIF: PWM总中断指示位, 只读;
 1= PWM产生中断, (清除具体的中断标志位后, 此位自动清除);
 0= PWM未产生中断。
- Bit2 --
- Bit1 TF4: Timer4定时器溢出中断标志位;
 1= Timer4定时器溢出, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= Timer4定时器无溢出。
- Bit0 TF3: Timer3定时器溢出中断标志位;
 1= Timer3定时器溢出, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= Timer3定时器无溢出。

11.4 定时器 3 工作模式

11.4.1 T3 -模式 0（13 位定时模式）

在这种模式下，定时器 3 是 13 位的寄存器。当定时器所有的位由 1 向 0 翻转后，定时器 3 中断标志 TF3 置 1。13 位的寄存器由 TH3 和 TL3 低 5 位组成。TL3 高 3 位应忽略。

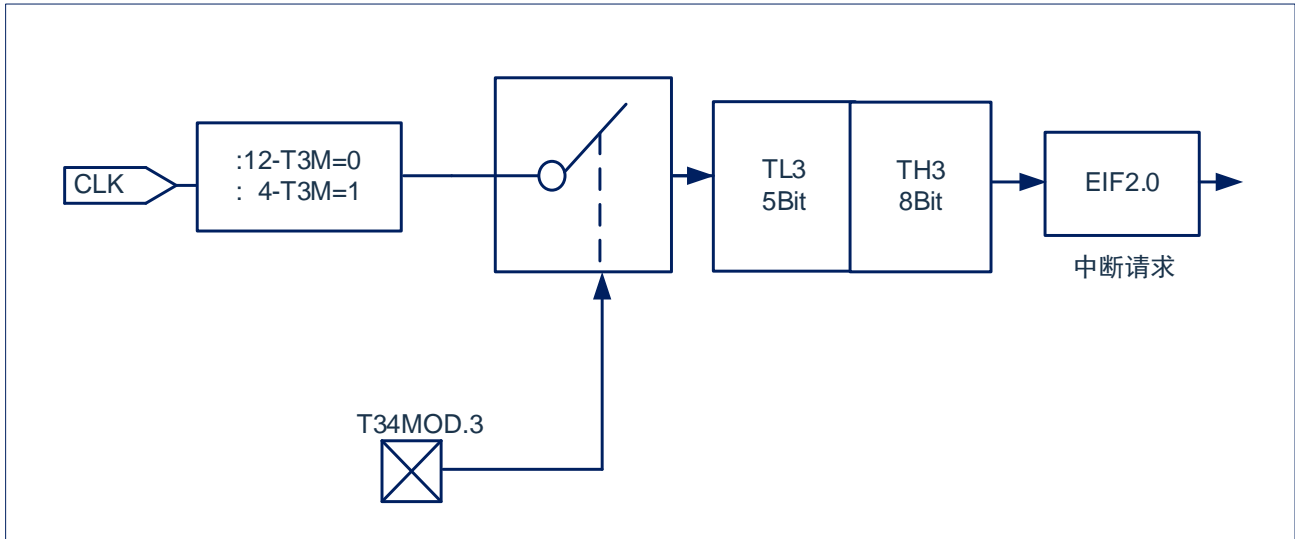


图 11-1: Timer3, 模式 0: 13-位定时器

11.4.2 T3 -模式 1（16 位定时模式）

模式 1 与模式 0 相同，只是模式 1 下定时器 3 寄存器 16 位全部运行。模式 1 如下图所示。

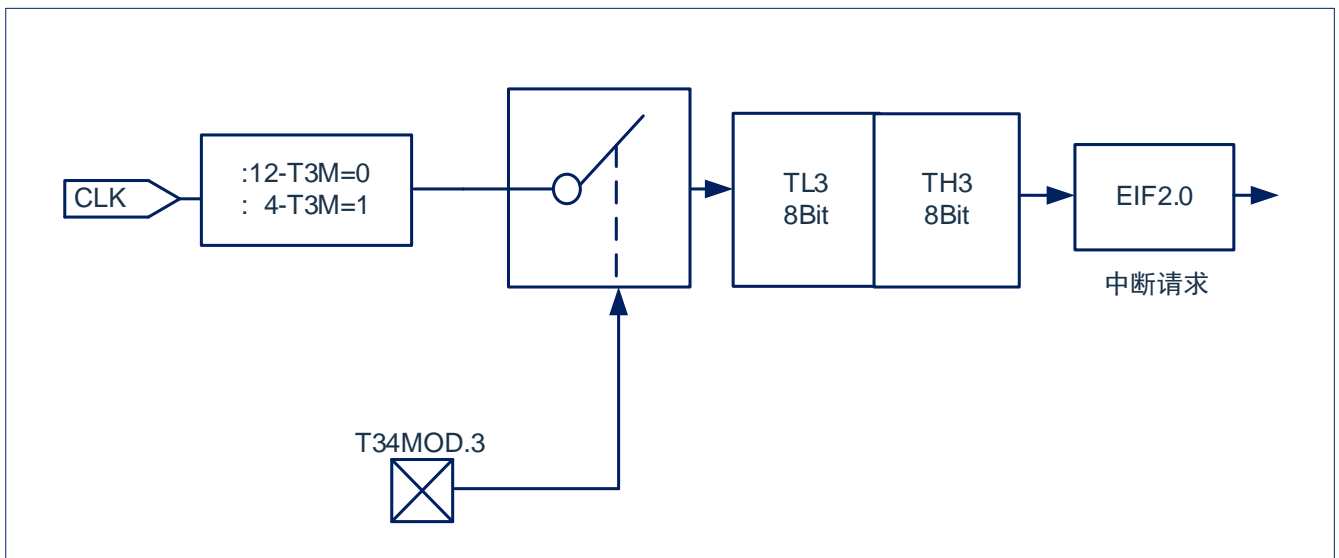


图 11-2: Timer3, 模式 1: 16-位定时器

11.4.3 T3 -模式 2（8 位自动重装定时模式）

模式 2 下定时器 3 寄存器是一个配有自动重装模式的 8 位定时器（TL3），如下图所示。来自 TL3 的溢出不仅使 TF3 置 1，而且还将 TH3 的内容由软件重装到 TL3。重装过程中 TH3 的值保持不变。

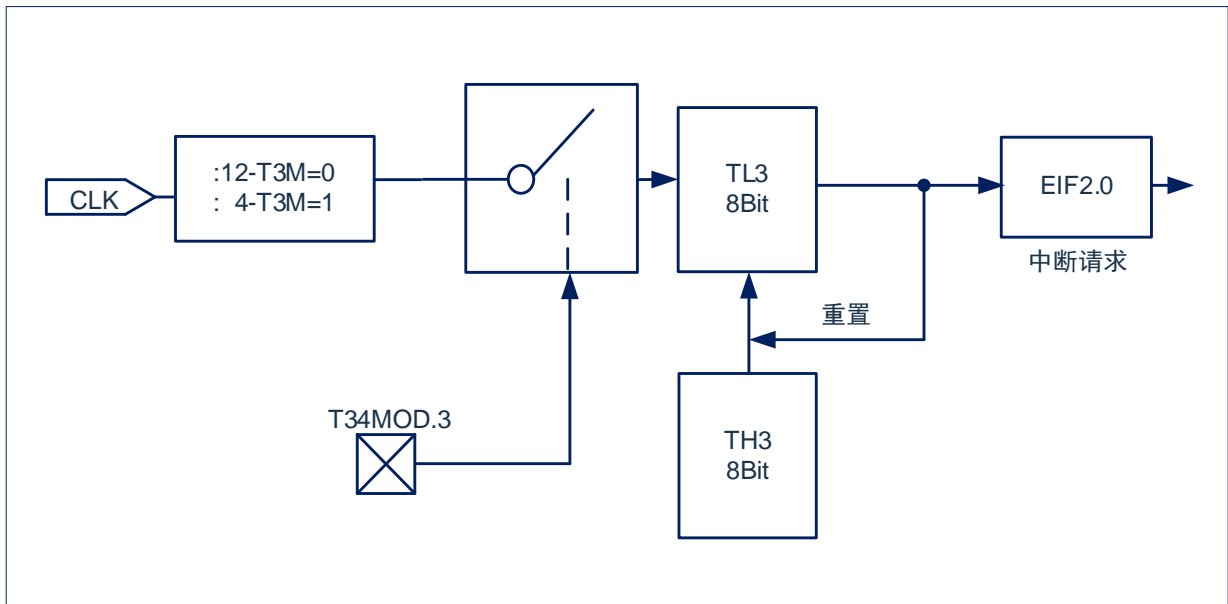


图 11-3: Timer3, 模式 2: 8-位定时器（自动重装）

11.4.4 T3 -模式 3（两个单独的 8 位定时器）

模式 3 下的定时器 3 将 TL3 和 TH3 设置为两个独立的定时器。定时器 3 模式 3 的逻辑如下图所示。

TL3 工作为 8 位定时器，并使用定时器 3 的控制位：如 TR3，和 TF3。

TH3 工作为 8 位定时器，且使用定时器 4 的 TR4 和 TF4 标志并控制定时器 4 中断。

需要使用两个 8 位定时器时可以使用模式 3。当定时器 3 处于模式 3 时，定时器 4 可以通过切换到自己的模式 3 而将其关闭，或者仍然可以被串行通道用作波特率发生器，或在不需要定时器 4 中断的任何应用中。

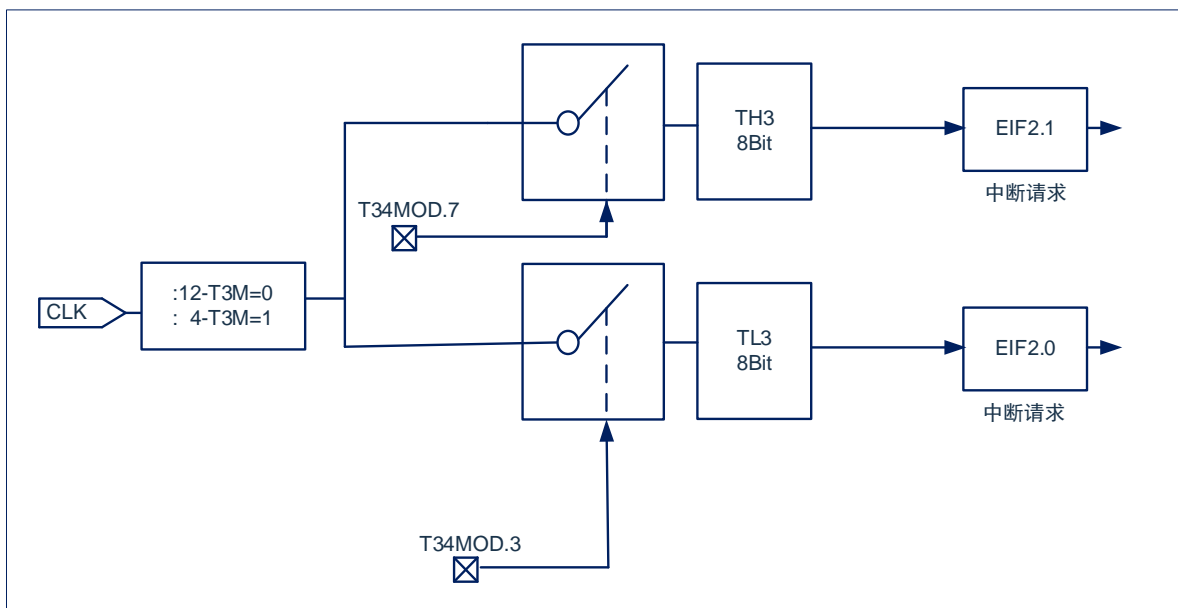


图 11-4: Timer3, 模式 3: 两个 8-位定时器

11.5 定时器 4 工作模式

11.5.1 T4 -模式 0（13 位定时模式）

在这种模式下，定时器 4 是 13 位的寄存器。当定时器所有的位由 1 向 0 翻转后，定时器 4 中断标志 TF4 置 1。13 位的寄存器由 TH4 8 位和 TL4 低 5 位组成。TL4 高三位应忽略。

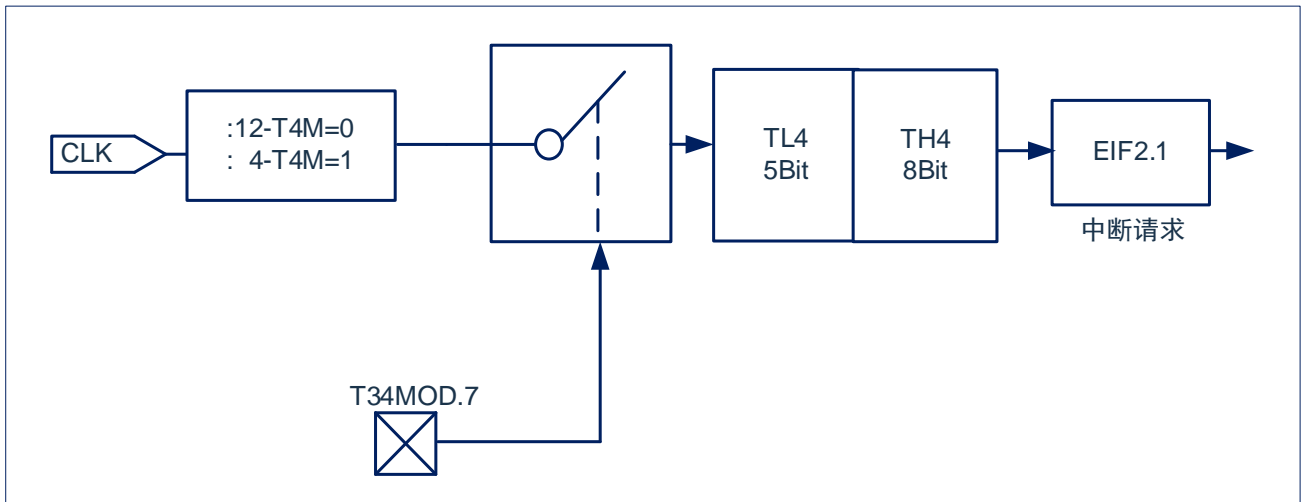


图 11-5: Timer4, 模式 0: 13-位定时器

11.5.2 T4 -模式 1（16 位定时模式）

模式 1 与模式 0 相同，只是模式 1 下定时器 4 寄存器 16 位全部运行。模式 1 如下图所示。

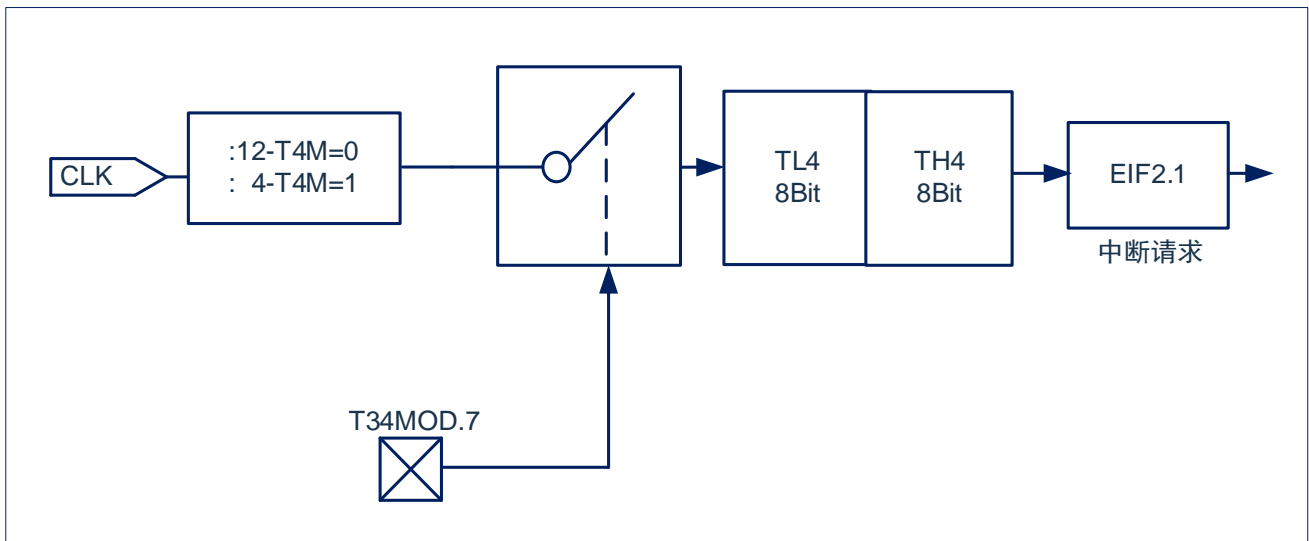


图 11-6: Timer4, 模式 1: 16-位定时器

11.5.3 T4- 模式 2（8 位自动重装定时模式）

模式 2 下定时器 4 寄存器是一个配有自动重装模式的 8 位定时器（TL4），如下图所示。来自 TL4 的溢出不仅使 TF4 置 1，而且还将 TH4 的内容由软件重装到 TL4。重装过程中 TH4 的值保持不变。

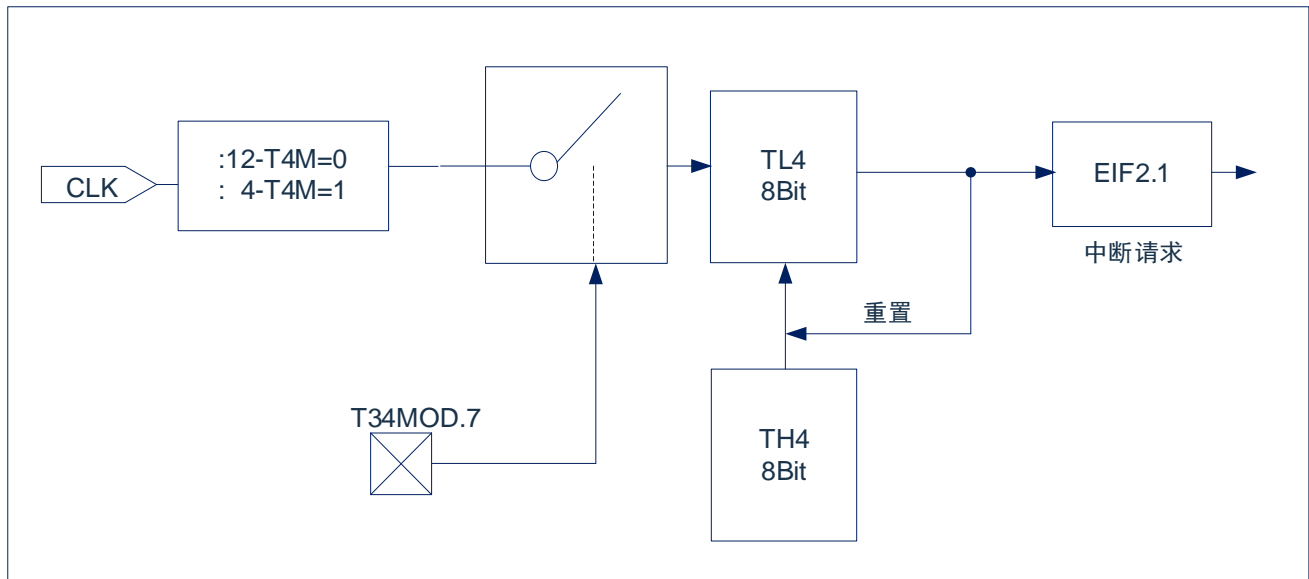


图 11-7: Timer4，模式 2: 8-位定时器（自动重装）

11.5.4 T4- 模式 3（停止计数）

模式 3 中的定时器 4 停止计数，其效果与设置 TR4=0 相同。

12. 蜂鸣器 (BUZZER)

蜂鸣驱动器由 8-Bit 计数器，时钟驱动器，控制寄存器组成。它产生 50%的占空方波，其频率覆盖一个较宽的范围。BUZZER 的输出频率由 BUZCON 寄存器与 BUZDIV 寄存器控制。

使用蜂鸣器时需要先将相应端口（支持任意 GPIO）配置成蜂鸣器口：

```
P24CFG = 0x18; // P2.4 配置成蜂鸣器口
```

BUZZER 控制寄存器 BUZCON

0xBF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BUZCON	BUZEN	--	--	--	--	--	BUZCKS1	BUZCKS0
R/W	R/W	R	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 BUZEN: BUZZER使能位;

1= 使能;

0= 禁止。

Bit6~ Bit2 --

Bit1~ Bit0 BUZCKS<1:0>: BUZZER分频比选择位;

00= Fsys/8;

01= Fsys/16;

10= Fsys/32;

11= Fsys/64。

BUZZER 频率控制寄存器 BUZDIV

0xBE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BUZDIV	BUZDIV7	BUZDIV6	BUZDIV5	BUZDIV4	BUZDIV3	BUZDIV2	BUZDIV1	BUZDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 BUZDIV[7:0]: BUZZER频率选择位;

00H: 无方波输出（输出低电平）;

其他值: $F_{buz} = F_{sys} / (2 * CLKDIV * BUZCKS)$ 。

例 1: $F_{sys} = 8\text{MHz}$, $BUZCKS<1:0> = 01$, $BUZDIV = 125$

$$F_{buz} = 8\text{MHz} / (2 * 125) / 16 = 2\text{KHz}$$

例 2: $F_{sys} = 16\text{MHz}$, $BUZCKS<1:0> = 11$, $BUZDIV = 125$

$$F_{buz} = 16\text{MHz} / (2 * 125) / 64 = 1\text{KHz}$$

例 3: $F_{sys} = 24\text{MHz}$, $BUZCKS<1:0> = 11$, $BUZDIV = 94$

$$F_{buz} = 24\text{MHz} / (2 * 94) / 64 = 2\text{KHz}$$

蜂鸣器输出频率表

BUZCKS<1:0>	Fbuz@Fsys=8MHz	Fbuz@Fsys=16MHz	Fbuz@Fsys=24MHz
00	2KHz-500KHz	4KHz-1MHz	6KHz-1.5MHz
01	1KHz-250KHz	2KHz-500KHz	3KHz-750KHz
10	0.5KHz-125KHz	1KHz-250KHz	1.5KHz-375KHz
11	0.25KHz-62.5KHz	0.5KHz-125KHz	0.75KHz-187.5KHz

13. 模数转换 (ADC)

13.1 ADC 概述

模数转换器 (ADC) 可以将模拟输入信号转换为表示该信号的一个 12 位二进制数。端口模拟输入信号经过多路选择器之后与模数转换器的输入相连。模数转换器采用逐次逼近法产生一个 12 位二进制结果，并将该结果保存在 ADC 结果寄存器 (ADRESL 和 ADRESH) 中。

ADC 参考电压始终为内部产生。ADC 在转换完成之后可以产生一个中断。

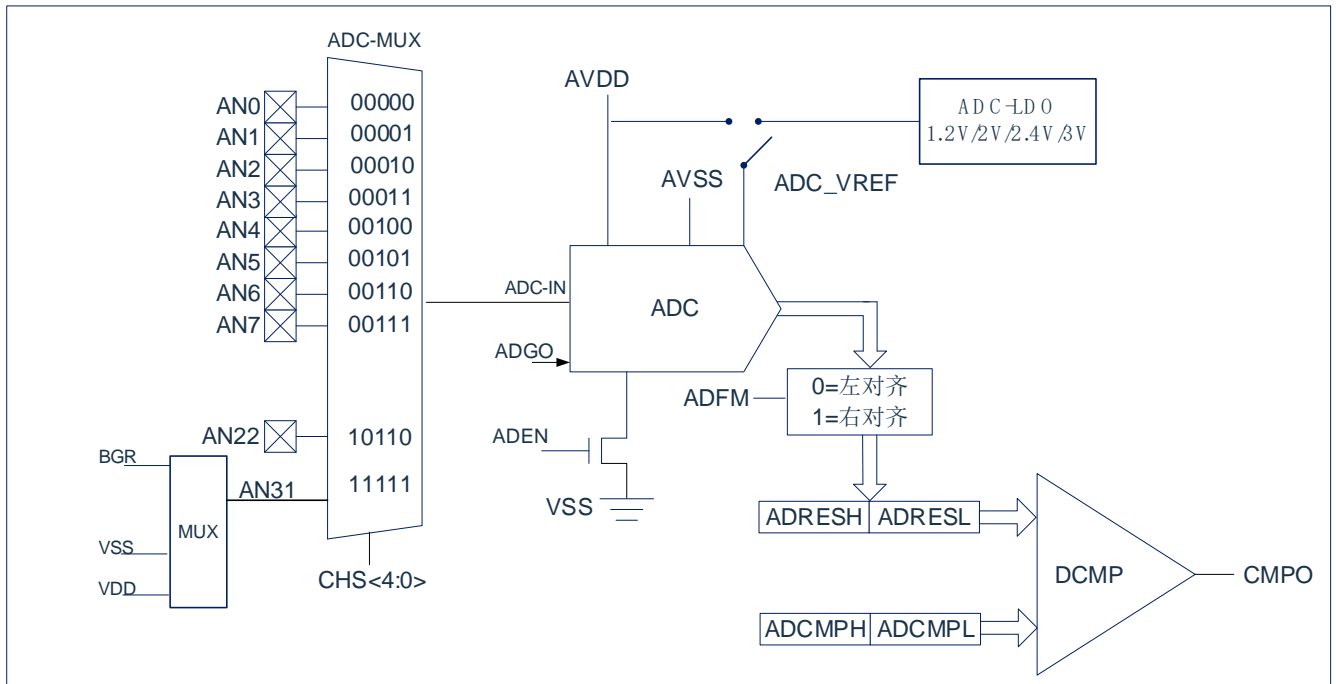


图 13-1: ADC 框图

13.2 ADC 配置

配置和使用 ADC 时，必须考虑如下因素：

- ◆ 端口配置；
- ◆ 通道选择；
- ◆ ADC 转换时钟源；
- ◆ 中断控制；
- ◆ 结果的存储格式。

13.2.1 端口配置

ADC 既可以转换模拟信号，又可以转换数字信号。当转换模拟信号时，应该通过将相应的端口配置为模拟端口。更多信息请参见相应的端口章节。

注：对定义为数字输入的引脚施加模拟电压可能导致输入缓冲器出现过电流。

13.2.2 通道选择

由 ADCON1 寄存器的 ADCHS 位决定将哪个通道连接到模数转换器。

如果更改了通道，在下一次转换开始前需要一定的延迟。ADC 延时时间如下表所示：

延时时间	工作电压
500ns	2.5~4.5V
200ns	4.5~5.5V

更多信息请参见“
ADC 工作”章节。

13.2.3 ADC 参考电压

ADC 的参考电压默认由芯片的 VDD 提供，也可由内部 ADC-LDO 提供，ADC-LDO 可选择 4 种电压输出：1.2V/2.0V/2.4V/3.0V。

13.2.4 转换时钟

可以通过软件设置 ADCON1 寄存器的 ADCKS 位来选择转换的时钟源。有以下 8 种可能的时钟频率可供选择：

- ◆ $F_{svs}/2$
- ◆ $F_{svs}/4$
- ◆ $F_{svs}/8$
- ◆ $F_{svs}/16$
- ◆ $F_{svs}/32$
- ◆ $F_{svs}/64$
- ◆ $F_{svs}/128$
- ◆ $F_{svs}/256$

完成一位转换的时间定义为 T_{AD} 。一个完整的 12 位转换需要 18.5 个 T_{AD} 周期（完成一次转换 ADGO 持续为高的时间）。

必须符合相应的 T_{AD} 规范，才能获得正确的转换结果，下表为选择 ADC 时钟的参考示例。

F_{sys}	T_{AD}
8MHz	$F_{sys}/8$
16MHz	$F_{sys}/16$
24MHz	$F_{sys}/32$

注：系统时钟频率的任何改变都会改变 ADC 时钟的频率，从而对 ADC 转换结果产生负面影响。

13.2.5 ADC 中断

ADC 模块允许在完成模数转换后产生一个中断。ADC 中断标志位是 EIF2 寄存器中的 ADCIF 位。ADC 中断允许位是 EIE2 寄存器中的 ADCIE 位。ADCIF 位必须用软件清零。每次转换结束后 ADCIF 位都会被置 1，与是否允许 ADC 中断无关。

13.2.6 结果格式化

12 位 A/D 转换的结果可采用两种格式：左对齐或右对齐。由 ADCON0 寄存器的 ADFM 位控制输出格式。

- 当 ADFM=0 时，AD 转换结果左对齐；
- 当 ADFM=1 时，AD 转换结果右对齐。

13.3 ADC 硬件触发启动

除了软件触发 AD 转换，该 ADC 模块还提供了硬件触发启动的方式，一种为外部端口边沿触发方式，一种为 PWM 的边沿或周期触发方式。使用硬件触发 ADC 需要将 ADCEX 置 1，即使能外部触发 ADC 功能。硬件触发信号经过一定延时后将 ADGO 位置 1，转换完毕后将自动清零。开启硬件触发功能后，不会关闭软件触发功能，ADC 空闲状态下，写 1 到 ADGO 位也能启动 AD 转换。

13.3.1 外部端口边沿触发 ADC

ADET 引脚边沿自动触发 AD 转换。此时 ADTGS[1:0]需要为 11（选择外部端口边沿触发），ADEGS[1:0]可选择何种边沿触发。

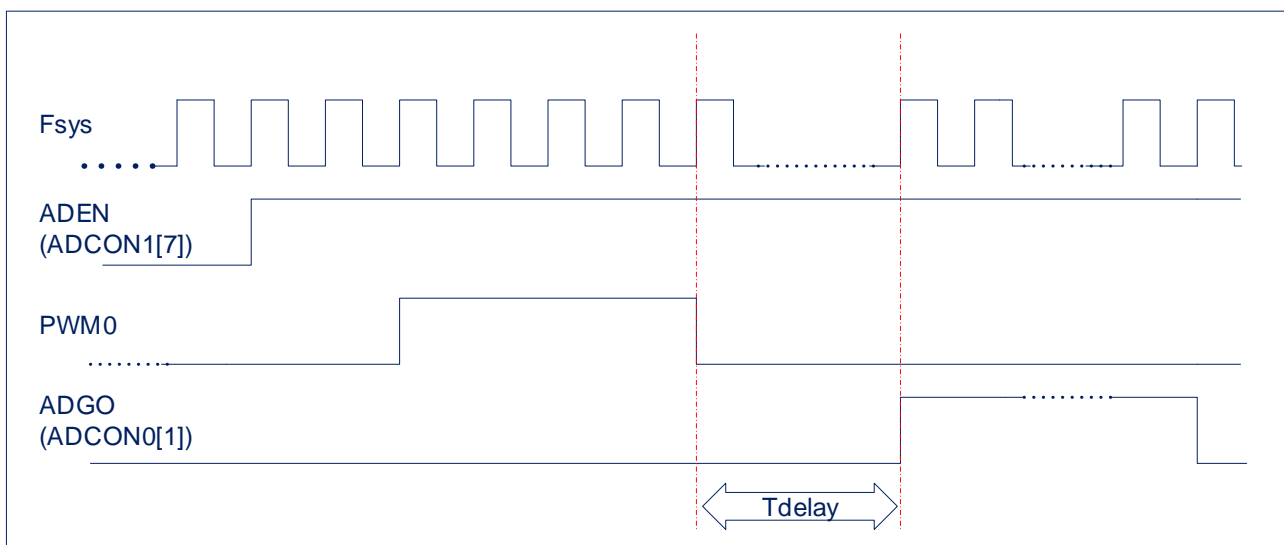
13.3.2 PWM 触发 ADC

PWM 可选择是由边沿或者周期零点/中点触发 ADC 转换。ADTGS[1:0]选择 PWM 通道 (PG0、PG2、PG4)，ADEGS[1:0]可选择边沿类型或者周期类型的触发方式。

13.3.3 硬件触发启动前的延时

硬件触发信号产生后，并不是立即启动 AD 转换，需要经过一定的延时后才将 ADGO 置 1。延时由 ADDLY[9:0]决定，结构框图如下图所示。

硬件触发信号的延时时间： $T_{delay} = (ADDLY + 3) * T_{sys}$



13.4 ADC 工作原理

13.4.1 启动转换

要使能 ADC 模块，必须先将 ADCON1 寄存器的 ADEN 位置 1，然后将 ADCON0 寄存器的 ADGO 位置 1 开始模数转换（ADEN 为 0 时无法将 ADGO 置 1）。

13.4.2 完成转换

当转换完成时，ADC 模块将：

- ◆ 清零 ADGO 位；
- ◆ 将 ADCIF 标志位置 1；
- ◆ 用转换的新结果更新 ADRESH:ADRESL 寄存器。

13.4.3 终止转换

ADC 启动后必须等待 ADC 转换完成才能终止 ADC 转换，禁止在 ADC 转换过程中终止 ADC 转换。

注：器件复位将强制所有寄存器进入复位状态。因此，复位会关闭 ADC 模块并且终止任何待处理的转换。

13.4.4 A/D 转换步骤

如下步骤给出了使用 ADC 进行模数转换的示例：

1. 端口配置：
 - 禁止引脚输出驱动器（见 PxTRIS 寄存器）；
 - 将引脚配置为模拟输入引脚。
2. 配置 ADC 模块：
 - 选择 ADC 转换时钟；
 - 选择 ADC 输入通道；
 - 选择结果的格式；
 - 启动 ADC 模块。
3. 配置 ADC 中断（可选）：
 - 清零 ADC 中断标志位；
 - 允许 ADC 中断；
 - 允许外设中断；
 - 允许全局中断。
4. 等待所需的采集时间。
5. 将 ADGO 置 1 启动转换。
6. 由如下方法之一等待 ADC 转换结束：
 - 查询 ADGO 位；
 - 等待 ADC 中断（允许中断）。
7. 读 ADC 结果。
8. 将 ADC 中断标志位清零（如果允许中断的话，需要进行此操作）。

注：如果用户尝试在使器件从休眠模式唤醒后恢复顺序代码执行，则必须禁止全局中断。

13.4.5 转换过程中休眠进入休眠

系统进入休眠时，必须等待 ADC 正在进行的转换完成后，再进入休眠状态。禁止 ADC 转换过程中让芯片进入休眠。

13.5 ADC 相关寄存器

主要有 10 个寄存器与 AD 转换相关，分别是：

- AD 控制寄存器 ADCON0、ADCON1、ADCON2、ADCLDO；
- 比较器控制寄存器 ADCMPC；
- 延时数据寄存器 ADDLYL；
- AD 结果数据寄存器 ADRESH/L；
- 比较器数据寄存器 ADCMPH/L。

AD 控制寄存器 ADCON0

0xDF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADCHS4	ADFM	--	AN31SEL2	AN31SEL1	AN31SEL0	ADGO	--
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	ADCHS4	ADC模拟通道选择位第4位； 1= 选择通道分配参考下文寄存器ADCON1中描述； 0= --
Bit6	ADFM	ADC 转换结果格式选择位； 1= 右对齐； 0= 左对齐。
Bit5	--	保留
Bit4~Bit2	AN31SEL<2:0>	ADC通道31输入源选择位； 000= BGR(1.2V)； 101= VSS (ADC参考地)； 110= 保留，禁止使用； 111= VDD (ADC默认参考电压)。
Bit1	ADGO	ADC转换启动位 (对该位置1时ADEN必须为1，否则操作无效)； 1= 写入：开始ADC转换，(硬件触发ADC时也会将该位置1)； 读取：ADC正在进行转换。 0= 写入：无效。 读取：ADC空闲/转换完毕； 在ADC的转换期间 (ADGO=1)，任何软件和硬件触发信号将被忽略。
Bit0	--	

AD 控制寄存器 ADCON1

0xDE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	ADEN	ADCKS2	ADCKS1	ADCKS0	ADCHS3	ADCHS2	ADCHS1	ADCHS0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	1	0	0	0	0	0	0

Bit7	ADEN: ADC使能位; 1= 使能ADC; 0= 禁止ADC, 不消耗工作电流。
Bit6~Bit4	ADCKS<2:0>: ADC转换时钟选择位; 000= Fsys/2 100= Fsys/32 001= Fsys/4 101= Fsys/64 010= Fsys/8 110= Fsys/128 011= Fsys/16 111= Fsys/256
Bit3~Bit0	ADCHS<3:0>: 模拟通道选择位低4位, 与ADCHS<4>组成5位通道选择位, ADCHS<4:0>; 00000= AN0 10000= AN16 00001= AN1 10001= AN17 00010= AN2 10010= AN18 00011= AN3 10011= AN19 00100= AN4 10100= AN20 00101= AN5 10101= AN21 00110= AN6 10110= AN22 00111= AN7 其他= -- 01000= AN8 11111= 见ADCON0.AN31SEL说明 01001= AN9 01010= AN10 01011= AN11 01100= AN12 01101= AN13 01110= AN14

AD 控制寄存器 ADCON2

0xE9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON2	ADCEX	--	ADTGS1	ADTGS0	ADEGS1	ADEGS0	--	--
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 ADCEX: ADC硬件触发使能位;
 1= 使能;
 0= 禁止。
- Bit6 保留: 必须为0。
- Bit5~Bit4 ADTGS: ADC硬件触发源选择位;
 00= PG0(PWM0);
 01= PG2 (PWM2);
 10= PG4 (PWM4);
 11= 端口引脚(ADET)。
- Bit3~ Bit2 ADEGS: ADC 硬件触发边沿选择位;
 00= 下降沿;
 01= 上升沿;
 10= PWM周期的周期点;
 11= PWM周期的零点。
- Bit1~Bit0 --

AD 比较器控制寄存器 ADCMPC

0xD1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPC	ADFBEN	ADCMPPS	--	ADCMPO	--	--	ADDLY9	ADDLY8
读写	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 ADFBEN: ADC比较器结果控制PWM刹车使能位;
 1= 使能;
 0= 禁止。
- Bit6 ADCMPPS: ADC比较器输出极性选择位;
 1= 若ADRES<ADCMP, 则ADCMPO=1;
 0= 若ADRES>=ADCMP, 则ADCMPO=1。
- Bit5 --
- Bit4 ADCMPO: ADC比较器结果输出位 (只读)。
 该位输出ADC比较器输出的结果, 每次ADC转换结束都会更新该位。
- Bit3~Bit2 --
- Bit1~ Bit0 ADDLY[9:8]: ADC硬件触发延时数据[9:8]位。

AD 硬件触发延时数据寄存器 ADDLYL

0xD3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDLYL	ADDLY7	ADDLY6	ADDLY5	ADDLY4	ADDLY3	ADDLY2	ADDLY1	ADDLY0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 ADDLY[7:0]: ADC硬件触发延时数据低8位。

AD 数据寄存器高位 ADRESH, ADFM=0 (左对齐)

0xDD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESH	ADRES11	ADRES10	ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4
读写	R	R	R	R	R	R	R	R
复位值	X	X	X	X	X	X	X	X

Bit7~Bit0 ADRES<11:4>: ADC结果寄存器位。
12位转换结果的第11-4位。

AD 数据寄存器低位 ADRESL, ADFM=0 (左对齐)

0xDC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESL	ADRES3	ADRES2	ADRES1	ADRES0	--	--	--	--
读写	R	R	R	R	--	--	--	--
复位值	X	X	X	X	--	--	--	--

Bit7~Bit4 ADRES<3:0>: ADC结果寄存器位。
12位转换结果的第3-0位。

Bit3~Bit0 未用。

AD 数据寄存器高位 ADRESH, ADFM=1 (右对齐)

0xDD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESH	--	--	--	--	ADRES11	ADRES10	ADRES9	ADRES8
读写	--	--	--	--	R	R	R	R
复位值	--	--	--	--	X	X	X	X

Bit7~Bit4 未用。

Bit3~Bit0 ADRES<11:8>: ADC结果寄存器位。
12位转换结果的第11-8位。

AD 数据寄存器低位 ADRESL, ADFM = 1 (右对齐)

0xDC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESL	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
读写	R	R	R	R	R	R	R	R
复位值	X	X	X	X	X	X	X	X

Bit7~Bit0 ADRES<7:0>: ADC结果寄存器位。
12位转换结果的第7-0位。

AD 比较器数据寄存器 ADCMPH

0xD5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPH	D11	D10	D9	D8	D7	D6	D5	D4
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit0 ADCMP<11:4>: ADC比较器数据高8位。

AD 比较器数据寄存器 ADCMPL

0xD4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPL	--	--	--	--	D3	D2	D1	D0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit4 未用。

Bit 3~Bit0 ADCMP[3:0]: ADC比较器数据低4位。

ADC 参考电压控制寄存器 ADCLDO

F692H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCLDO	LDOEN	VSEL1	VSEL0	--	--	--	--	--
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 LDOEN ADC_LDO使能;

1= LDO使能, 参考电压只能选择VSEL[1:0]对应的电压;

0= LDO禁止, 参考电压为芯片电源电压。

Bit 6~Bit5 VSEL[1:0]: ADC参考电压选择位;

00= 1.2V ;

01= 2.0V ;

10= 2.4V ;

11= 3.0V 。

Bit4 -- 保留, 须为0。

Bit 3~Bit0 --

13.6 ADC 结果比较

ADC 模块提供了一组数字比较器，用于比较 ADC 的结果与预先装入{ADCMPL, ADCMPH}的值大小。每次 ADC 转换的结果将会与预设值 ADCMP 进行比较，比较的结果存放在 ADCMPO 标志位中，转换完毕后，该标志位会自动更新。ADCMPPS 位可以改变输出结果的极性。

ADC 比较结果可触发 PWM 故障刹车，开启这一功能需要将 ADFBEN 置 1。

当 PWM 功能开启，ADFBEN=1 时，AD 转换的结果同预设值{ADCMPL, ADCMPH}进行比较，若比较的结果 ADCMPO 为 1，则 PWM 立即产生故障刹车操作，将所有 PWM 通道的启动位清零，终止所有 PWM 通道输出。

14. 增强型 PWM 模块

14.1 概述

增强性 PWM 模块支持 6 路 PWM 发生器，可以配置成相互独立的 6 路 PWM 输出（PG0-PG5），也可以配置成 3 对分别带有编程死区发生器的互补 PWM（PG0-PG1，PG2-PG3，PG4-PG5）。

每一对 PWM 共用 8 位预分频器，有 6 组时钟分频器，提供 5 种分频系数（1、1/2、1/4、1/8、1/16）。每一路 PWM 输出有独立的 16 位计数器进行控制，另外 16 位的比较器用以调节占空比。6 路 PWM 发生器提供 24 个中断标志，相关 PWM 通道的周期或占空比与计数器相符，将产生中断标志，每一路 PWM 有单独的使能位。

每路 PWM 可配置成单次模式（产生一个 PWM 信号周期）或者循环模式（连续输出 PWM 波形）。

14.2 特性

增强性 PWM 模块有如下特性：

- 6 路独立的 16 位 PWM 控制模式。
 - 6 路独立输出：PG0、PG1、PG2、PG3、PG4、PG5；
 - 3 组互补 PWM 对：（PG0-PG1）、（PG2-PG3）、（PG4-PG5），可插入可编程死区时间；
 - 3 组同步 PWM 对：（PG0-PG1）、（PG2-PG3）、（PG4-PG5），每组 PWM 对引脚同步。
- 支持组控制，PG0，PG2，PG4 输出同步，PG1，PG3，PG5 输出同步。
- 单次模式或者自动装载模式。
- 支持边沿对齐，中心对齐 2 种模式。
- 中心对齐模式支持对称计数和非对称计数。
- 互补的 PWM 中，支持可编程死区发生器。
- 每路 PWM 有独立的极性控制。
- 硬件刹车保护（外部 FB 触发，支持软件触发）。
- ADC 比较事件可触发硬件刹车保护。
- PWM 边沿或周期可触发启动 AD 转换。

14.3 端口配置

使用增强型 PWM 模块前需要先将相应端口配置成 PWM 通道（任意 GPIO 均可配置为 PWM 通道），PWM 通道在复用功能分配图上用 PG0-PG5 来标注，分别对应 PWM 通道 0-5。

PWM 通道的分配由相应的端口配置寄存器来控制，例如：

```
P13CFG=0x12; //选择 P13 配置成 PG0 通道
```

```
P14CFG=0x13; //选择 P14 配置成 PG1 通道
```

```
P15CFG=0x14; //选择 P15 配置成 PG2 通道
```

```
P16CFG=0x15; //选择 P16 配置成 PG3 通道
```

```
P17CFG=0x16; //选择 P17 配置成 PG4 通道
```

```
P22CFG=0x17; //选择 P22 配置成 PG5 通道
```

14.4 功能描述

14.4.1 加载更新模式

计数器加载模式有两种：单次模式与自动加载模式。单次模式下，周期和占空比相关数据在计数器开始加载一次。自动加载模式下，周期和占空比相关数据在 PWM 周期内零点自动加载。

由于 PWM 存在双缓存结构，在 PWM 运行的过程中，改变相关运行寄存器：PWMPnL/PWMPnH/PWMDnL/PWMDnH/PWMDDnL/PWMDDnH/的值，PWM 输出波形不会立即改变，只有在零点时这些寄存器的值才会加载到相应的缓存中。这样的结构在改变周期占空比数据后，不会立即改变当前 PWM 周期的输出波形，在下一个周期 PWM 波形才会做出相应的变化。即任何 PWM 相关数据的改变不会影响当前一个完整 PWM 周期。

在高速的应用中，有可能会出现加载点已经到来，但写入运行寄存器的操作还未完成。此时不期望出现部分运行数据已经加载，另外一部分运行数据没有加载的情况。针对该高速应用情况，PWM 模块提供了加载使能位。

当改变相关运行寄存器后，需要将加载寄存器 PWMLOADEN 的使能位 PWMnLE 置 1，周期和占空比加载完毕后 PWMnLE 位自动清零。即可以通过读取该位来判断是否将相关寄存器的值加载到实际电路中。如果 PWMnLE=0，则表示已经加载，将影响正在输出的 PWM 波形；如果 PWMnLE=1，则表示还未加载，当前的 PWM 波形还未发生变化，将在下一个加载点加载之前改变的寄存器的值。如果再次改变相关运行寄存器的值，也需重新将 PWMnLE=1 置 1。

注：当 PWMnLE=1 时，对周期和占空比寄存器内容的更改，可能引发无法预测的结果。

建议先更改周期和占空比寄存器内容，再将加载使能位 PWMnLE 置 1，最后等待加载完成 (PWMnLE=0)。

14.4.2 边沿对齐

边沿对齐模式下，16 位 PWM 计数器 CNTn 在每个周期开始向下计数，与锁存 CMPn 值进行比较，当 CNTn=CMPn 时 PGn 输出高电平，CMPnDIF 置 1。CNTn 继续向下计数至 0，此时 PGn 将输出低电平，当前 CMPn 和 PERIODn 在 PWMnCNTM=1 的情况会重新加载，PIF 周期中断标志置位。

边沿对齐相关参数如下：

$$\text{高电平时间} = (\text{CMPn} + 1) \times T_{\text{pwm}} (\text{CMPn} \geq 1)$$

$$\text{周期} = (\text{PERIODn} + 1) \times T_{\text{pwm}}$$

$$\text{占空比} = \frac{\text{CMPn} + 1}{\text{PERIODn} + 1} (\text{CMPn} \geq 1)$$

CMPn=0 时，占空比为 0%。

边沿对齐时序如下图所示：

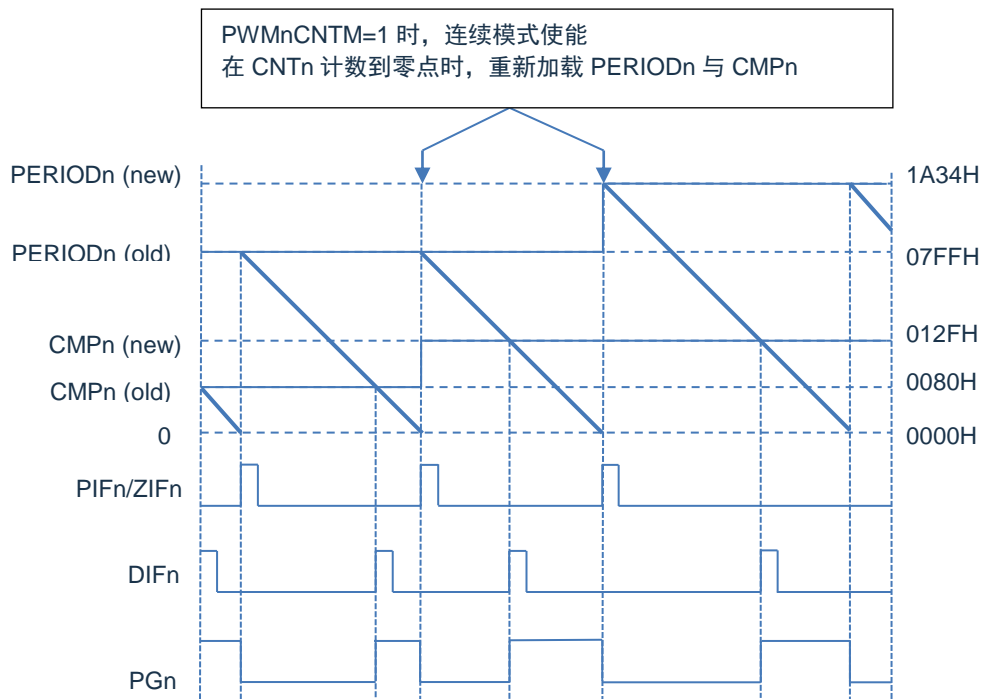


图 14-1：边沿对齐时序图

14.4.3 中心对齐

中心对齐对称计数方式下，16 位 PWM 计数器 CNTn 从 0 开始向上计数，当 CNTn=CMPn 时，PGn 输出高电平，之后 CNTn 继续向上计数至与 PERIODn 相等，然后 CNTn 开始向下计数，在向下计数的过程中 CNTn=CMPn 时，PGn 输出低电平，之后继续向下计数至 0。

中心对齐对称计数方式相关参数如下：

$$\text{高电平时间} = (\text{PERIODn} \times 2 - \text{CMPn} \times 2 - 1) \times T_{\text{pwm}}; (\text{CMPn} \geq 1)$$

$$\text{周期} = (\text{PERIODn}) \times 2 \times T_{\text{pwm}};$$

$$\text{占空比} = \frac{\text{PERIODn} \times 2 - \text{CMPn} \times 2 - 1}{\text{PERIODn} \times 2}; (\text{CMPn} \geq 1)$$

CMPn=0 时，占空比为 100%；

中心对齐时序（对称计数）如下图所示：

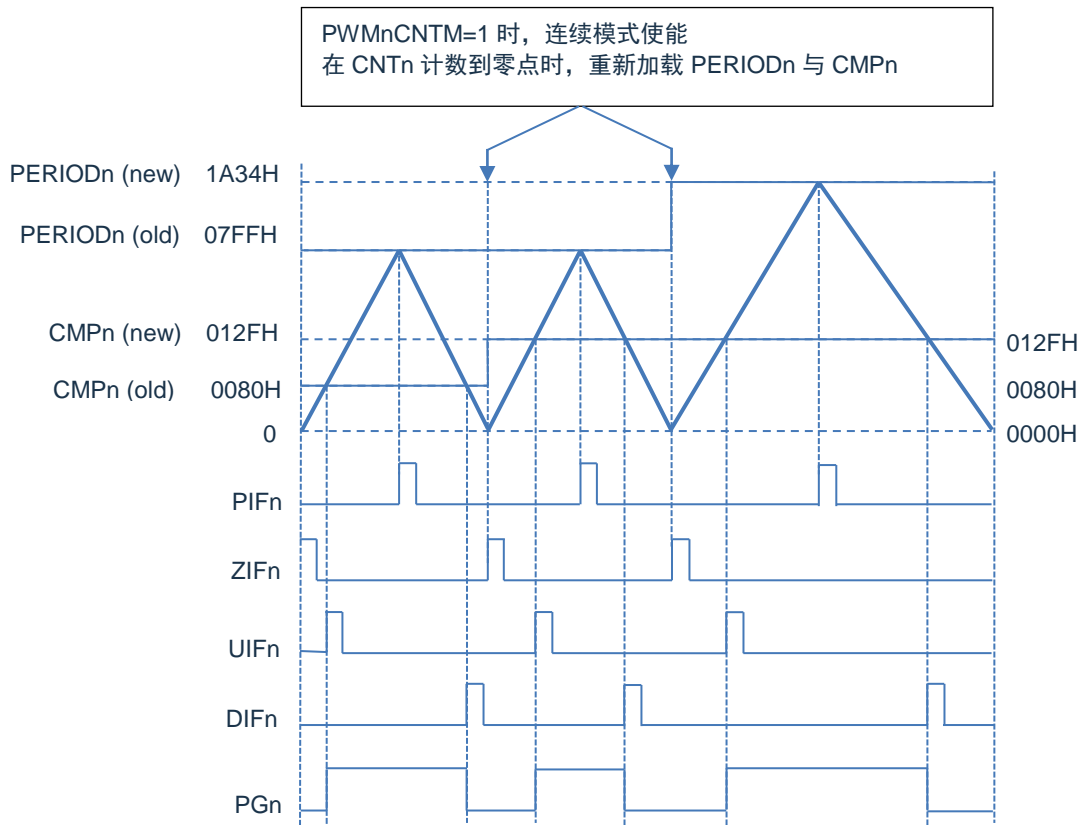


图 14-2：中心对齐时序图（对称计数）

中心对其计数器波形（对称计数）如下图所示：

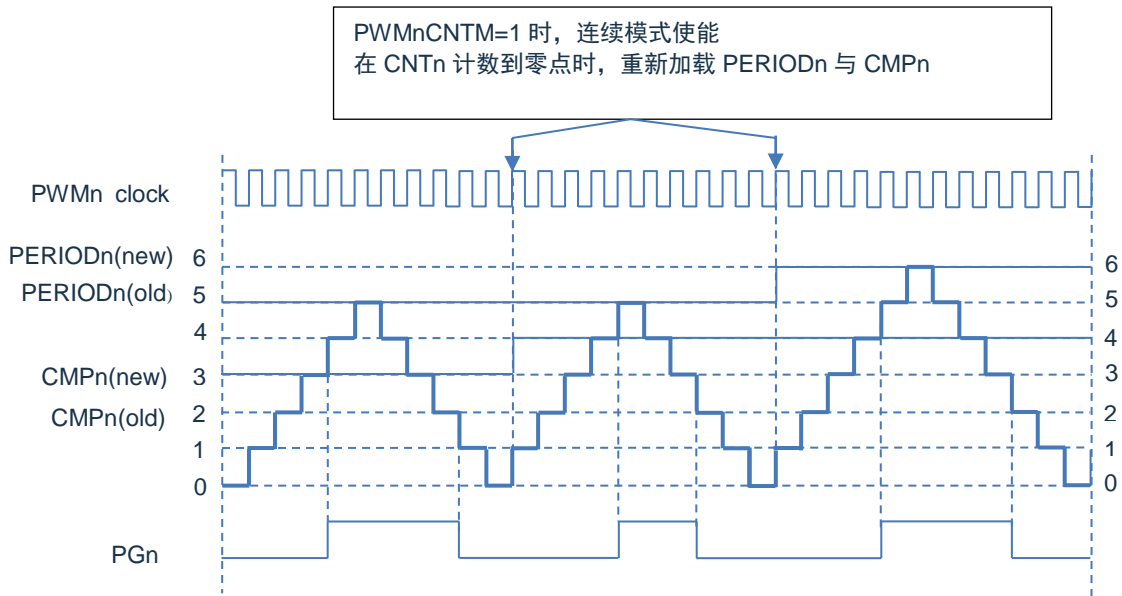


图 14-3: 中心对齐计数器波形 (对称计数)

中心对齐非对称计数方式下，16 位 PWM 计数器 CNTn 从 0 开始向上计数，当 CNTn=CMPn 时，PGn 输出高电平，之后 CNTn 继续向上计数至与 PERIODn 相等，然后 CNTn 开始向下计数，在向下计数的过程中 CNTn=CMPDn 时，PGn 输出低电平，之后继续向下计数至 0。开启非对称计数方式需要将 ASYMEN 置 1，非对称计数方式下可实现精确的中心对齐波形。

中心对齐非对称计数方式相关参数如下：

$$\text{高电平时间} = (\text{PERIODn} \times 2 - \text{CMPDn} - \text{CMPn} - 1) \times T_{\text{pwm}}$$

$$\text{周期} = (\text{PERIODn}) \times 2 \times T_{\text{pwm}}$$

$$\text{占空比} = \frac{\text{PERIODn} \times 2 - \text{CMPDn} - \text{CMPn} - 1}{\text{PERIODn} \times 2}$$

CMPn=0 与 CMPDn=0 时，占空比为 100%；

中心对齐时序（非对称计数）如下图所示：

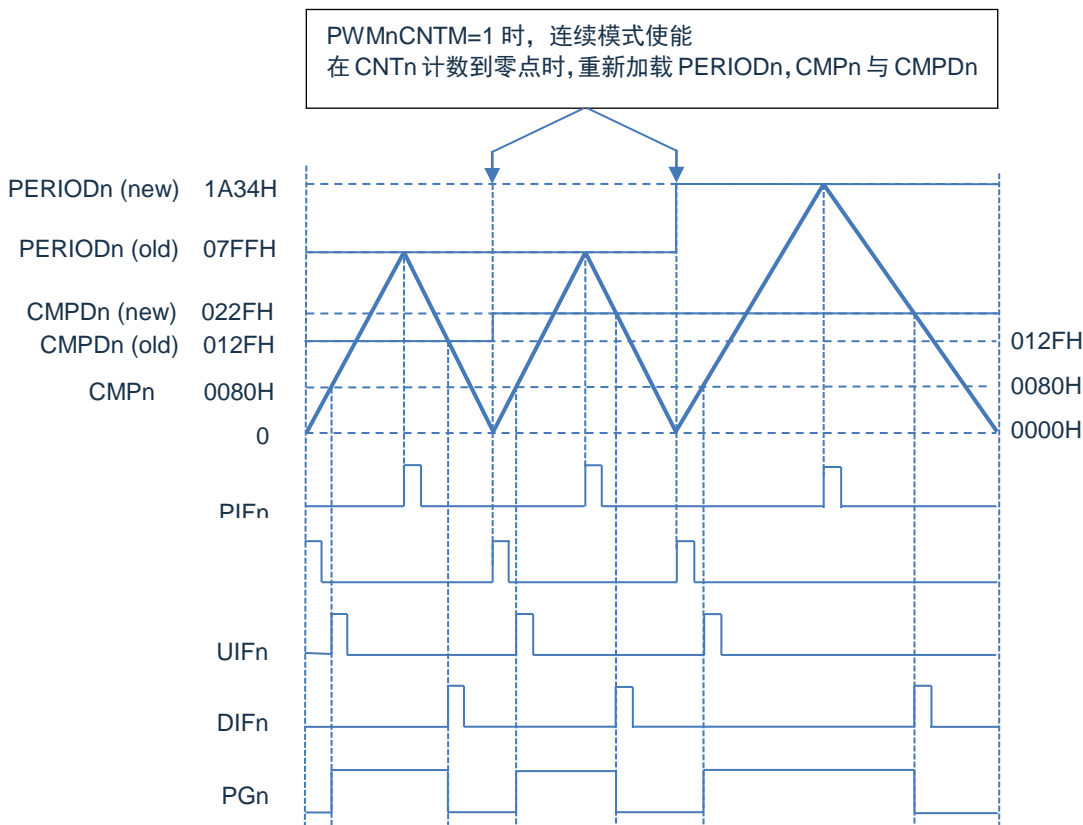


图 14-4：中心对齐时序图（非对称计数）

中心对齐时序（非对称计数）如下图所示：

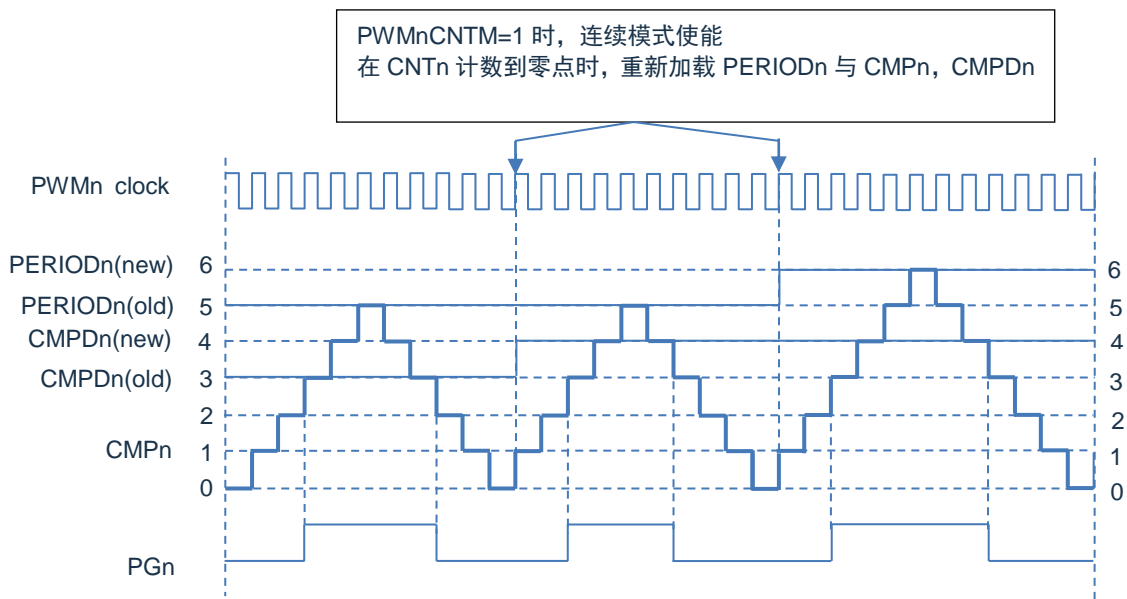


图 14-5: 中心对齐时序图 (非对称计数)

14.4.4 互补模式与死区延时

6 通道 PWM 可设置为 3 组互补对。在互补模式下, PWM1, PWM3, PWM5 的周期与占空比分别由 PWM0, PWM2, PWM4 相关寄存器决定, 同时死区延时寄存器也可影响 PWM 互补对的占空比。此时除了对应的输出使能控制位 (PWMnOE), PWM1/PWM3/PWM5 输出波形不再受自己的寄存器控制。

在互补模式下, 每组互补 PWM 对均支持插入死区延时, 插入的死区时间如下:

$$\text{PWM0/1 死区时间: } (\text{PWM01DT}+1) * T_{\text{PWM0}}$$

$$\text{PWM2/3 死区时间: } (\text{PWM23DT}+1) * T_{\text{PWM2}}$$

$$\text{PWM4/5 死区时间: } (\text{PWM45DT}+1) * T_{\text{PWM4}}$$

$T_{\text{PWM0}}/T_{\text{PWM2}}/T_{\text{PWM4}}$ 分别为 PWM0/PWM2/PWM4 的时钟源周期。

中心对齐与边沿对齐均支持互补模式。

14.4.5 刹车功能

可触发 PWM 刹车的信号源有如下几种:

1. 外部触发端口 FB;
2. ADC 结果比较输出;

触发刹车后, 刹车标志位置 1, 所有通道的计数器使能位清零, 且 PWM 输出预设的刹车数据。如需恢复正常输出, 则需要将刹车标志清零, 并且将 PWM 通道计数器重新使能。

14.5 PWM 相关寄存器

PWM 控制寄存器 PWMCON

F120H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON	--	--	PWMMODE1	PWMMODE0	GROUPEN	ASYMEN	CNTTYPE	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit6	--
Bit5~Bit4	PWMMODE: PWM的模式控制位; 00= 独立模式; 01= 互补模式; 10= 同步模式; 11= 保留。
Bit3	GROUPEN: PWM成组功能使能位; 1= PGO控制PG2, PG4; PG1控制PG3, PG5; 0= 所有PWM通道信号相互独立。
Bit2	ASYMEN: PWM中心对齐方式下非对称计数使能位; 1= 非对称计数使能; 0= 对称计数使能。
Bit1	CNTTYPE: PWM计数对齐方式选择位; 1= 中心对齐方式; 0= 边沿对齐方式。
Bit0	--

PWM 输出使能控制寄存器 PWMOE

F121H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWNOE	--	--	PWM5OE	PWM4OE	PWM3OE	PWM2OE	PWM1OE	PWM0OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6	--
Bit5	PWM5OE: PWM通道5的输出使能位; 1= 使能; 0= 禁止。
Bit4	PWM4OE: PWM通道4的输出使能位; 1= 使能; 0= 禁止。
Bit3	PWM3OE: PWM通道3的输出使能位; 1= 使能; 0= 禁止。
Bit2	PWM2OE: PWM通道2的输出使能位; 1= 使能; 0= 禁止。
Bit1	PWM1OE: PWM通道1的输出使能位; 1= 使能; 0= 禁止。
Bit0	PWM0OE: PWM通道0的输出使能位; 1= 使能; 0= 禁止。

PWM0/1 时钟预分频控制寄存器 PWM01PSC

F123H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM01PSC	PWM01PSC7	PWM01PSC6	PWM01PSC5	PWM01PSC4	PWM01PSC3	PWM01PSC2	PWM01PSC1	PWM01PSC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM01PSC<7:0>: PWM通道0/1预分频控制位;
 00= 预分频时钟停止, PWM0/1的计数器停止;
 其他= 系统时钟的 (PWM01PSC+1) 分频。

PWM2/3 时钟预分频控制寄存器 PWM23PSC

F124H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM23PSC	PWM23PSC7	PWM23PSC6	PWM23PSC5	PWM23PSC4	PWM23PSC3	PWM23PSC2	PWM23PSC1	PWM23PSC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM23PSC<7:0>: PWM通道2/3预分频控制位;
 00= 预分频时钟停止, PWM2/3的计数器停止;
 其他= 系统时钟的 (PWM23PSC+1) 分频。

PWM4/5 时钟预分频控制寄存器 PWM45PSC

F125H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM45PSC	PWM45PSC7	PWM45PSC6	PWM45PSC5	PWM45PSC4	PWM45PSC3	PWM45PSC2	PWM45PSC1	PWM45PSC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM45PSC<7:0>: PWM通道4/5预分频控制位;
 00= 预分频时钟停止, PWM4/5的计数器停止;
 其他= 系统时钟的 (PWM45PSC+1) 分频。

PWM 时钟分频控制寄存器 PWMnDIV(n=0-5)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMnDIV	--	--	--	--	--	PWMnDIV2	PWMnDIV1	PWMnDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMnDIV(n=0-5)地址: F12AH, F12BH, F12CH, F12DH, F12EH, F12FH。

Bit7~Bit3 --
 Bit2~Bit0 PWMnDIV<2:0>: PWM通道n时钟分频控制位;
 000= Fpwmn-PSC/2;
 001= Fpwmn-PSC/4;
 010= Fpwmn-PSC/8;
 011= Fpwmn-PSC/16;
 100= Fpwmn-PSC;
 其他= Fsys (系统时钟);
 (PSC为预分频后的时钟)。

PWM 数据加载使能控制寄存器 PWMLOADEN

F129H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMLOADEN	--	--	PWM5LE	PWM4LE	PWM3LE	PWM2LE	PWM1LE	PWM0LE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit6

--

Bit5~ Bit0

PWMnLE: PWM通道n的数据加载使能位(n=0-5) (加载完成后硬件清零), 当PWMnLE=1时, 对周期和占空比寄存器内容的更改, 可能引发无法预测的结果。

1= 使能加载周期, 占空比数据(PERIODn、CMPn、CMPDn)。

0= 写0无效。

PWM 输出极性控制寄存器 PWMPINV

F122H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPINV	--	--	PWM5PINV	PWM4PINV	PWM3PINV	PWM2PINV	PWM1PINV	PWM0PINV
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit6

--

Bit5~ Bit0

PWMnPINV: PWM通道n输出极性控制位 (n=0-5);

1= 反向输出;

0= 正常输出。

PWM 计数器模式控制寄存器 PWMCNTM

F127H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTM	--	--	PWM5CNTM	PWM4CNTM	PWM3CNTM	PWM2CNTM	PWM1CNTM	PWM0CNTM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit6

--

Bit5~ Bit0

PWMnCNTM: PWM通道n计数器模式控制位 (n=0-5);

1= 自动加载模式;

0= One-shot模式。

PWM 计数器使能控制寄存器 PWMCNTE

F126H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTE	--	--	PWM5CNTE	PWM4CNTE	PWM3CNTE	PWM2CNTE	PWM1CNTE	PWM0CNTE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit6

--

Bit5~ Bit0

PWMnCNTE: PWM通道n计数器使能控制位(n=0-5);

1= PWMn计数器开启 (PWMn开始输出);

0= PWMn计数器停止 (软件写0则计数器停止并清掉计数器值)。

PWM 计数器模式控制寄存器 PWMCNTCLR

F128H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTCLR	--	--	PWM5CNTCLR	PWM4CNTCLR	PWM3CNTCLR	PWM2CNTCLR	PWM1CNTCLR	PWM0CNTCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit6 --

Bit5~ Bit0 PWMnCNTCLR: PWM通道n计数器清零控制位(n=0-5) (硬件自动清零);

1= PWMn计数器清零;

0= 写0无效。

PWM 周期数据寄存器低 8 位 PWMPnL (n=0-5)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPnL	PWMPnL7	PWMPnL6	PWMPnL5	PWMPnL4	PWMPnL3	PWMPnL2	PWMPnL1	PWMPnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMPnL (n=0-5)地址: F130H, F132H, F134H, F136H, F138H, F13AH。

Bit7~ Bit0 PWMPnL<7:0>: PWM通道n周期数据寄存器低8位。

PWM 周期数据寄存器高 8 位 PWMPnH (n=0-5)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPnH	PWMPnH7	PWMPnH6	PWMPnH5	PWMPnH4	PWMPnH3	PWMPnH2	PWMPnH1	PWMPnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMPnH (n=0-5)地址: F131H, F133H, F135H, F137H, F139H, F13BH。

Bit7~ Bit0 PWMPnH<7:0>: PWM通道n周期数据寄存器高8位。

PWM 比较数据寄存器低 8 位 PWMDnL (n=0-5)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDnL	PWMDnL7	PWMDnL6	PWMDnL5	PWMDnL4	PWMDnL3	PWMDnL2	PWMDnL1	PWMDnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMDnL (n=0-5)地址: F140H, F142H, F144H, F146H, F148H, F14AH。

Bit7~ Bit0 PWMDnL<7:0>: PWM通道n比较数据 (占空比数据) 寄存器低8位。

PWM 比较数据寄存器高 8 位 PWMDnH (n=0-5)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDnH	PWMDnH7	PWMDnH6	PWMDnH5	PWMDnH4	PWMDnH3	PWMDnH2	PWMDnH1	PWMDnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMDnH (n=0-5)地址: F141H, F143H, F145H, F147H, F149H, F14BH。

Bit7~ Bit0 PWMDnH<7:0>: PWM通道n比较数据 (占空比数据) 寄存器高8位。

PWM 向下比较数据寄存器低 8 位 PWMDDnL (n=0-5)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDDnL	PWMDDnL7	PWMDDnL6	PWMDDnL5	PWMDDnL4	PWMDDnL3	PWMDDnL2	PWMDDnL1	PWMDDnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMDDnL (n=0-5)地址: F150H, F152H, F154H, F156H, F158H, F15AH。

Bit7~ Bit0 PWMDDnL<7:0>: PWM通道n向下比较数据 (非对称计数下占空比数据) 寄存器低8位。

PWM 向下比较数据寄存器高 8 位 PWMDDnH (n=0-5)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDDnH	PWMDDnH7	PWMDDnH6	PWMDDnH5	PWMDDnH4	PWMDDnH3	PWMDDnH2	PWMDDnH1	PWMDDnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 PWMDDnH (n=0-5)地址: F151H, F153H, F155H, F157H, F159H, F15BH。

Bit7~ Bit0 PWMDDnH<7:0>: PWM通道n向下比较数据 (非对称计数下占空比数据) 寄存器高8位。

PWM 死区使能控制寄存器 PWM DTE

F160H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM DTE	--	--	--	--	--	PWM45DTE	PWM23DTE	PWM01DTE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7~Bit3 --
- Bit2 PWM45DTE: PWM4/5通道死区延时使能位;
1= 使能;
0= 禁止。
- Bit1 PWM23DTE: PWM2/3通道死区延时使能位;
1= 使能;
0= 禁止。
- Bit0 PWM01DTE: PWM0/1通道死区延时使能位;
1= 使能;
0= 禁止。

PWM 0/1 死区延时数据寄存器 PWM01DT

F161H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM01DT	PWM01DT7	PWM01DT6	PWM01DT5	PWM01DT4	PWM01DT3	PWM01DT2	PWM01DT1	PWM01DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM01DT<7:0>: PWM通道0/1死区延时数据寄存器。

PWM 2/3 死区延时数据寄存器 PWM23DT

F162H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM23DT	PWM23DT7	PWM23DT6	PWM23DT5	PWM23DT4	PWM23DT3	PWM23DT2	PWM23DT1	PWM23DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM23DT<7:0>: PWM通道2/3死区延时数据寄存器。

PWM4/5 死区延时数据寄存器 PWM45DT

F163H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM45DT	PWM45DT7	PWM45DT6	PWM45DT5	PWM45DT4	PWM45DT3	PWM45DT2	PWM45DT1	PWM45DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM45DT<7:0>: PWM通道4/5死区延时数据寄存器。

PWM 掩码控制寄存器 PWMMASKE

F164H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMMASKE	--	--	PWM5MASKE	PWM4MASKE	PWM3MASKE	PWM2MASKE	PWM1MASKE	PWM0MASKE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit6 --

Bit5~Bit0 PWMnMASKE: PWM通道n掩码控制使能位 (n=0-5);
 1= PWMn通道使能掩码数据输出;
 0= PWMn通道禁止掩码数据输出 (正常输出PWM波形)。

PWM 掩码数据寄存器 PWMMASKD

F165H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMMASKD	--	--	PWM5MASKD	PWM4MASKD	PWM3MASKD	PWM2MASKD	PWM1MASKD	PWM0MASKD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit6 --

Bit5~Bit0 PWMnMASKD: PWM通道n掩码数据位 (n=0-5);
 1= PWMn通道输出高;
 0= PWMn通道输出低。

PWM 刹车控制寄存器 PWMFBKC

F166H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBKC	PWMFBIE	PWMFBF	PWM5FBCCE	PWMFBKSW	PWMFBES	--	PWMFBEN	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 PWMFBIE: PWM 刹车中断屏蔽位;
 1= 使能中断;
 0= 禁止中断。
- Bit6 PWMFBF: PWM 刹车标志位 (写0清除);
 1= 产生了刹车操作 (PWM输出刹车数据寄存器的值);
 0= 没有产生刹车操作。
- Bit5 PWMFBCCE: PWM 刹车时是否清掉所有通道计数器选择位;
 1= 刹车时计数器的值不会清掉;
 0= 刹车时清掉通道计数器值。
- Bit4 PWMFBKSW: PWM 软件刹车信号启动位;
 1= PWM产生软件刹车信号;
 0= 禁止。
- Bit3 PWMFBES: PWM 外部硬件刹车通道 (FB) 触发电平选择位;
 1= 高电平;
 0= 低电平。
- Bit2 -- 保留, 须为0。
- Bit1 PWMFBEN: PWM 外部硬件刹车通道 (FB) 使能位;
 1= 使能;
 0= 禁止。
- Bit0 -- 保留, 须为0。

PWM 刹车数据寄存器 PWMFBKD

F167H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBKD	--	--	PWM5FBKD	PWM4FBKD	PWM3FBKD	PWM2FBKD	PWM1FBKD	PWM0FBKD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7~Bit6 未用。
- Bit5~Bit0 PWMnFBKD: PWM通道n刹车数据位 (n=0-5);
 1= PWMn通道产生刹车操作后输出高。
 0= PWMn通道产生刹车操作后输出低。

PWM 周期中断屏蔽寄存器 PWMPPIE

F168H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPPIE	--	--	PWM5PIE	PWM4PIE	PWM3PIE	PWM2PIE	PWM1PIE	PWM0PIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6

--

Bit5~Bit0 PWMnPIE: PWM通道n周期中断屏蔽位 (n=0-5);

1= 使能中断;

0= 禁止中断。

PWM 零点中断屏蔽寄存器 PWMZIE

F169H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMZIE	--	--	PWM5ZIE	PWM4ZIE	PWM3ZIE	PWM2ZIE	PWM1ZIE	PWM0ZIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6

--

Bit5~Bit0 PWMnZIE: PWM通道n零点中断屏蔽位 (n=0-5);

1= 使能中断;

0= 禁止中断。

PWM 向上比较中断屏蔽寄存器 PWMUIE

F16BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMUIE	--	--	PWM5UIE	PWM4UIE	PWM3UIE	PWM2UIE	PWM1UIE	PWM0UIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6

--

Bit5~Bit0 PWMnUIE: PWM通道n向上比较中断屏蔽位 (n=0-5);

1= 使能中断;

0= 禁止中断。

PWM 向下比较中断屏蔽寄存器 PWMDIE

F16BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDIE	--	--	PWM5DIE	PWM4DIE	PWM3DIE	PWM2DIE	PWM1DIE	PWM0DIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6

--

Bit5~Bit0 PWMnDIE: PWM通道n向下比较中断屏蔽位 (n=0-5);

1= 使能中断;

0= 禁止中断。

PWM 周期中断标志寄存器 PWMPIF

F16CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPIF	--	--	PWM5PIF	PWM4PIF	PWM3PIF	PWM2PIF	PWM1PIF	PWM0PIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6

--

Bit5~Bit0

PWMnPIF: PWM通道n周期中断标志位 (n=0-5);

1= 产生中断 (软件清零);

0= 未产生中断。

PWM 零点中断标志寄存器 PWMZIF

F16DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMZIF	--	--	PWM5ZIF	PWM4ZIF	PWM3ZIF	PWM2ZIF	PWM1ZIF	PWM0ZIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6

--

Bit5~Bit0

PWMnZIF: PWM通道n零点中断标志位 (n=0-5);

1= 产生中断 (软件清零);

0= 未产生中断。

PWM 向上比较中断标志寄存器 PWMUIF

F16FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMUIF	--	--	PWM5UIF	PWM4UIF	PWM3UIF	PWM2UIF	PWM1UIF	PWM0UIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6

--

Bit5~Bit0

PWMnUIF: PWM通道n向上比较中断标志位 (n=0-5);

1= 产生中断 (软件清零);

0= 未产生中断。

PWM 向下比较中断标志寄存器 PWMDIF

F16FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDIF	--	--	PWM5DIF	PWM4DIF	PWM3DIF	PWM2DIF	PWM1DIF	PWM0DIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit6

--

Bit5~Bit0

PWMnDIF: PWM通道n向下比较中断标志位 (n=0-5);

1= 产生中断 (软件清零);

0= 未产生中断。

15. I²C 模块

15.1 概述

该模块提供微处理器和 I²C 总线之间的接口。且支持仲裁和时钟同步，以便允许在多主机系统中运行。I²C 支持正常、快速模式。

其特性如下：

- ◆ 支持 4 种工作方式：主控发送、主控接收、从动发送、从动接收。
- ◆ 支持 2 种传输速度模式：
 - 标准（高达 100Kb/s）；
 - 快速（高达 400Kb/s）；
- ◆ 执行仲裁和时钟同步。
- ◆ 支持多主机系统。
- ◆ 主机方式支持 I²C 总线上的 7 位寻址模式与 10 位寻址模式（软件支持）。
- ◆ 从机方式支持 I²C 总线上的 7 位寻址模式。
- ◆ 中断产生。
- ◆ 允许在较宽时钟频率范围进行操作（内置 8 位定时器）。

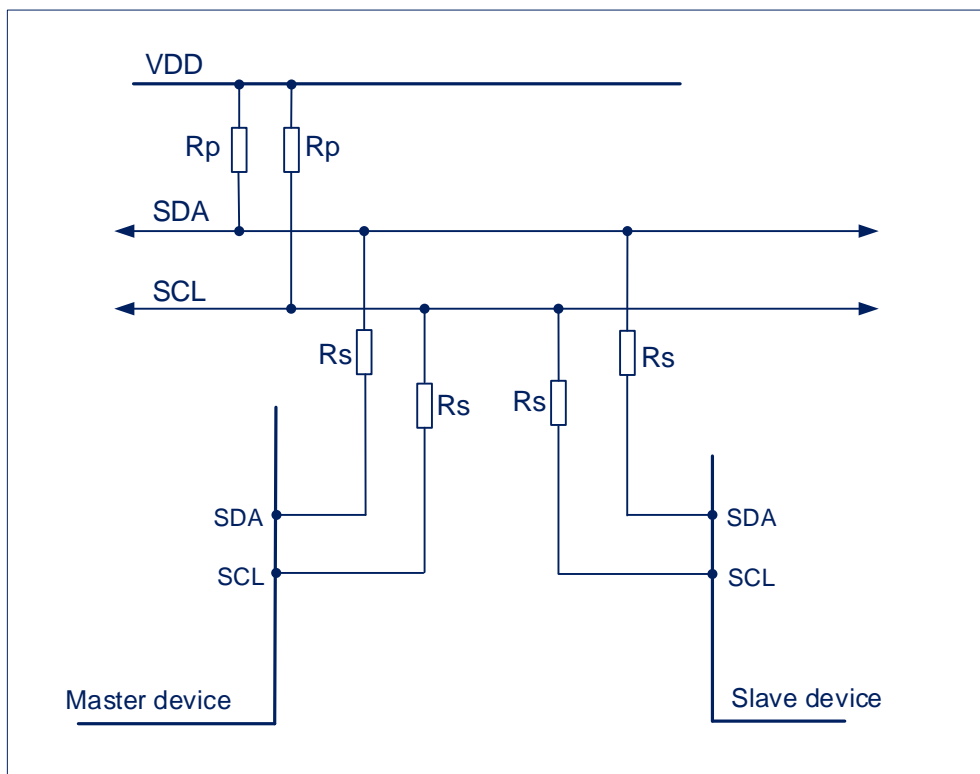


图 15-1: I²C 连接图

15.2 I²C 端口配置

若使用 I²C 功能，则首先应将相应端口配置成 SCL，SDA 通道（任意 GPIO 均可配置为 I²C 通道）。例如将 P0.4，P0.5 端口配置成 I²C 功能：

```
P04CFG=0x0C; //选择 P0.4 配置为 SCL 通道
```

```
P05CFG=0x0D; //选择 P0.5 配置为 SDA 通道
```

配置 I²C 通道后，该组端口默认为开漏状态，即输出数据 1 时为开漏输出。可通过 P0UP 配置是否使能 SCL，SDA 口的内部上拉电阻，或者在芯片外部增加上拉电阻。

15.3 I²C 主控模式

有六个寄存器用于与主机连接：控制、状态、从机地址、发送数据、接收数据和定时器周期寄存器。

寄存器		地址
写入	读取	
从动地址寄存器 I2CMSA	从动地址寄存器 I2CMSA	0xF4
主控模式控制寄存器 I2CMCR	主控模式状态寄存器 I2CMSR	0xF5
主控发送数据寄存器 I2CMBUF	主控接收数据寄存器 I2CMBUF	0xF6
定时周期寄存器 I2CMTP	定时周期寄存器 I2CMTP	0xF7

主控模式控制寄存器 I2CMCR 与主控模式状态寄存器 I2CMSR 共用一个寄存器地址，但物理上为两个不同的寄存器。

主控发送数据寄存器与主控接收数据寄存器共用一个寄存器地址，写操作访问发送寄存器 I2CMBUF，读操作访问接收寄存器 I2CMBUF。

写操作时是作为控制寄存器写入，读操作是作为状态寄存器读取。

15.3.1 I²C 主控模式定时周期寄存器

为了产生宽范围的 SCL 频率，该模块内置了 8 位定时器。用于标准和快速传输。

TIMER_PRD \neq 0 时，SCL 的时钟周期： $2 * (1 + \text{TIMER_PRD}) * 10^* \text{Tsys}$

TIMER_PRD = 0 时，SCL 的时钟周期： $3 * 10^* \text{Tsys}$

0xF7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMTP	--	MTP6	MTP5	MTP4	MTP3	MTP2	MTP1	MTP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	1

Bit7 保留：必须为0。

Bit6~ Bit0 MTP6-MTP0: 标准与快速模式的周期定时寄存器第6-0位：TIMER_PRD[6:0]。

15.3.2 I²C 主控模式控制与状态寄存器

控制寄存器包括 4 位：RUN、START、STOP、ACK 位。START 位将产生 START 或 REPEATED START 条件。STOP 位决定数据传输是否在周期结束时停止，或者继续。为了生成单个发送周期，从机地址寄存器写入所需的地址，R/S 位设置为 0，控制寄存器写入 ACK=x、STOP=1、START=1、RUN=1 (I2CMCR=xxx0_x111x) 以执行操作和停止。当操作完成（或发生错误）时，产生中断。可以从接收数据寄存器读取数据。

当 I²C 在主控模式下工作时，ACK 位必须设置为 1。这将使 I²C 总线控制器在每个字节后自动发送应答。当 I²C 总线控制器不再需要从机发送数据时，该位必须清 0。

主控模式控制寄存器

0xF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMCR	RSTS	--	--	--	ACK	STOP	START	RUN
W	W	W	W	W	W	W	W	W
复位值	0	0	0	0	0	0	0	0

Bit7	RSTS: I ² C主动模块复位控制位； 1= 复位主控模块（整个主控模块的I ² C寄存器，包括I2CMSR）； 0= I ² C主控模式下中断标志位清0（清I2CMSR.I2CMIF）。
Bit6~ Bit5	-- 禁止访问
Bit4	保留: 必须为0。
Bit3	ACK: 应答使能位； 1= 使能； 0= 禁止。
Bit2	STOP: 停止使能位； 1= 使能； 0= 禁止。
Bit1	START: 启动使能位； 1= 使能； 0= 禁止。
Bit0	RUN: 运行使能位； 1= 使能； 0= 禁止。

通过以下的控制位组合列表可实现主控模式下的各种操作：

START：发送起始信号。

SEND：发送数据或地址。

RECEIVE：接收数据。

STOP：发送结束信号。

控制位的组合 (IDLE 状态)

R/S	ACK	STOP	START	RUN	OPERATION
0	-	0	1	1	START 后跟 SEND (主机保持在发送模式)
0	-	1	1	1	START 后跟 SEND 和 STOP
1	0	0	1	1	START 后接收采用应答 (主机保持在接收器模式)
1	0	1	1	1	START 后跟 RECEIVE 和 STOP
1	1	0	1	1	START 后跟 RECEIVE (主机保持在接收器模式)
1	1	1	1	1	禁止组合
0	0	0	0	1	禁止组合

控制位的组合 (主控发送状态)

R/S	ACK	STOP	START	RUN	OPERATION
-	-	0	0	1	SEND 操作
-	-	1	0	0	停止
-	-	1	0	1	SEND 后跟 STOP
0	-	0	1	1	重复 START 后跟 SEND
0	-	1	1	1	重复 START, 后跟 SEND 和 STOP
1	0	0	1	1	重复 START 条件后跟应答 RECAIVE 操作 (主机保持在接收器模式)
1	0	1	1	1	重复 START, 后跟 SEND 和 STOP 条件
1	1	0	1	1	重复 START 条件后跟 RECEIVE (主机保持在接收器模式)
1	1	1	1	1	禁止组合

控制位的组合 (主控接收状态)

R/S	ACK	STOP	START	RUN	OPERATION
-	0	0	0	1	采用应答的 RECEIVE 操作 (主机保持在接收器模式)
-	-	1	0	0	STOP
-	0	1	0	1	RECEIVE 后跟 STOP
-	1	0	0	1	RECEIVE 操作 (主机保持在接收器模式)
-	1	1	0	1	禁止组合
1	0	0	1	1	重复 START, 接着是采用应答 RECEIVE 操作 (主机保持在接收器模式)
1	0	1	1	1	重复 START, 接着是 RECEIVE 和 STOP
1	1	0	1	1	重复 START 后跟 RECEIVE (主机保持在接收器模式)
0	-	0	1	1	重复 START 后跟 SEND (主机保持在发送器模式)
0	-	1	1	1	重复 START, 后跟 SEND 和 STOP

主控模式状态寄存器 I2CMSR

0xF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMSR	I2CMIF	BUS_BUSY	IDLE	ARB_LOST	DATA_ACK	ADD_ACK	ERROR	BUSY
R	R	R	R	R	R	R	R	R
复位值	0	0	1	0	0	0	0	0

- Bit7 I2CMIF: I²C主控模式下中断标志位;
 1= 主控模式下, 发送/接收完成, 或者出现传输错误。(清I2CMIF, 需对I2CMCR.RSTS写0);
 0= 未产生中断。
- Bit6 BUS_BUSY: 主控模式/从控模式下I²C总线忙标志位;
 1= I²C总线正在忙, 不能进行传输(由总线上的起始位置1, 停止条件清零)。
 0= --
- Bit5 IDLE: I²C主控模式空闲标志位;
 1= 为空闲状态;
 0= 为工作状态。
- Bit4 ARB_LOST: I²C主控模式仲裁标志位;
 1= 丢失了总线控制权。
 0= --
- Bit3 DATA_ACK: I²C主控模式发送数据应答标志位;
 1= 最后一次发送数据没有应答。
 0= --
- Bit2 ADD_ACK: I²C主控模式寻址应答标志位;
 1= 最后一次寻址没有应答。
 0= --
- Bit1 ERROR: I²C主控模式错误标志位;
 1= 寻址从机没有应答/发送数据没有应答/I²C总线仲裁冲突。
 0= --
- Bit0 BUSY: I²C主控模块忙标志位;
 1= I²C 模块正在传输数据。
 0= --

15.3.3 I²C 从机地址寄存器

从机地址寄存器由 8 位组成：7 位地址位（A6-A0）和接收/不发送位 R/S。R/S 位决定下一个操作是接收（1）还是发送（0）。

主控模式从机地址寄存器 I2CMSA

0xF4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMSA	SA6	SA5	SA4	SA3	SA2	SA1	SA0	R/S
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit1 SA6-SA0: I²C主控模式下从机地址。
 Bit0 R/S: I²C主控模式下发送从机地址后接收/发送状态选择位；
 1= 寻址正确后接收数据；
 0= 寻址正确后发送数据。

15.3.4 I²C 主控模式发送与接收数据寄存器

发送数据寄存器由八个数据位组成，这些位将在下一次发送或突发发送操作时在总线上发送，第一个发送位是 MD7（MSB）。

主控模式数据缓存寄存器 I2CMBUF

0xF6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMBUF	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 MD7-MD0: I²C主控模式下发送/接收数据。

15.4 I²C 从动模式

有五个寄存器用于连接到目标设备：自身地址、控制、状态、发送数据和接收数据寄存器。

寄存器		地址
写入	读取	
自身地址寄存器 I2CSADR	自身地址寄存器 I2CSADR	0xF1
控制寄存器 I2CSCR	状态寄存器 I2CSSR	0xF2
发送数据 I2CSBUF	接收数据 I2CSBUF	0xF3

15.4.1 I²C 自有地址寄存器 I2CSADR

自有地址寄存器由在 I²C 总线上标识 I²C 内核的七个地址位组成。该寄存器可以读取和写入地址。

自有地址寄存器 I2CSADR

0xF1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSADR	--	SA6	SA5	SA4	SA3	SA2	SA1	SA0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 --
Bit6~Bit0 SA6-SA0: I²C从动模式的自有地址。

15.4.2 I²C 从动模式控制与状态寄存器 I2CSCR/I2CSSR

从动模式控制寄存器与从动模式状态寄存器占用一个寄存器地址，使用不同的操作来区分访问这两个寄存器：

写操作：写入 I2CSCR（只写）

读操作：读取 I2CSSR（只读）

控制寄存器由两个位组成：RSTS 和 DA 位。RSTS 位控制整个 I²C 从动模块的复位，当 I²C 总线遇到某些问题时，使用该位软件应用程序可以重新初始化 I2CS。DA 位启用和禁用 I2CS 设备操作。读取该地址将状态寄存器置于数据总线上。

从动模式控制寄存器 I2CSCR

0xF2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSCR	RSTS	--	--	--	--	--	--	DA
R/W	W	R	R	R	R	R	R	W
复位值	0	0	0	0	0	0	0	0

Bit7 RSTS: I²C从动模块复位控制位；
1= 复位从动模块；
0= 无影响。

Bit6~ Bit1 --

Bit0 DA: I²C从动模式使能位；
1= 使能；
0= 禁止。

状态寄存器由三个位组成：SENFIN 位，RREQ 位，TREQ 位。发送完成的 SENFIN 位表示主机 I²C 控制器在 I2CS 单次或连续发送操作期间已完成数据的接收。接收请求 RREQ 位表示 I2CS 器件已从 I²C 主机接收数据字节，I2CS 设备应从接收数据寄存器 I2CSBUF 读取一个数据字节。发送请求 TREQ 位指示 I2CS 设备作为从机发送器寻址，I2CS 设备应将一个数据字节写入发送数据寄存器 I2CSBUF。若 I²C 中断使能打开，3 个标志位任意一个置 1 会产生中断。

从控模式下总线忙标志位由主控模式状态寄存器 I2CMSR 的 Bit6（BUS_BUSY）判断，当总线空闲时 I2CMSR 为 0x20，当起始条件产生到停止条件产生前 I2CMSR 寄存器为 0x60，当停止条件产生时，I2CMSR 为 0x20。

从动模式状态寄存器 I2CSSR

0xF2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSSR	--	--	--	--	--	SENFIN	TREQ	RREQ
R/W	--	--	--	--	--	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit3

--

Bit2 SENFIN: I²C从动模式下发送操作完毕标志位，只读。

1= 主控制器件不再需要数据，TREQ不再设置为1，已经完成本次数据传送。（读I2CSCR后自动清零）。

0= --

Bit1 TREQ: I²C从动模式准备发送标志位，只读。

1= 作为发送器件已经被寻址，主控制器件已经做好接收数据的准备。（写I2CSBUF后自动清零）。

0= --

Bit0 RREQ: I²C从动模式接收完成标志位，只读。

1= 接收完毕。（读I2CSBUF后自动清零）。

0= 未接收完毕。

15.4.3 I²C 从动模式发送与接收缓冲寄存器 I2CSBUF

0xF3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSBUF	I2CSBUF7	I2CSBUF6	I2CSBUF5	I2CSBUF4	I2CSBUF3	I2CSBUF2	I2CSBUF1	I2CSBUF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~ Bit0 I2CSBUF: I²C发送或接收的数据；

写操作: 写将要发送的数据（发送顺序由高位依次到低位）；

读操作: 已经接收到的数据。

15.5 I²C 中断

I²C 的中断号为 21，其中断向量为 0x00AB。使能 I²C 中断必须将其使能位 I2CIE 置 1，且将总中断使能位 EA 置 1。

若 I²C 相关的中断使能均打开，I²C 总中断指示位 I2CIF=1 时，CPU 将进入中断服务程序。I2CIF 操作属性为只读，且与 I2CIE 的状态无关。

I²C 主控模式中断标志位 I2CMIF、从动模式下发送操作完毕标志位 SENDFIN、从动模式准备发送标志位 TREQ、从动模式接收完成标志位 RREQ 任意一个为 1 时，I²C 总中断指示位 I2CIF 将会置 1。仅当这 4 个标志位均为 0 时，I2CIF 自动清 1。

中断屏蔽寄存器 EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 SPIIE: SPI中断使能位；
 1= 允许SPI中断；
 0= 禁止SPI中断。
- Bit6 I2CIE: I²C中断使能位；
 1= 允许I²C中断；
 0= 禁止I²C中断。
- Bit5 WDTIE: WDT中断使能位；
 1= 允许WDT溢出中断；
 0= 禁止WDT溢出中断。
- Bit4 ADCIE: ADC中断使能位；
 1= 允许ADC中断；
 0= 禁止ADC中断。
- Bit3 PWMIE: PWM总中断使能位；
 1= 允许PWM所有中断；
 0= 禁止PWM所有中断。
- Bit2 --
- Bit1 ET4: Timer4中断使能位；
 1= 允许Timer4中断；
 0= 禁止Timer4中断。
- Bit0 ET3: Timer3中断使能位；
 1= 允许Timer3中断；
 0= 禁止Timer3中断。

外设中断优先级控制寄存器 EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 PSPI: SPI中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit6 PI2C: I2C中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit5 PWDT: WDT中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit4 PADC: ADC中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit3 PPWM: PWM中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit2 --
- Bit1 PT4: TIMER4中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit0 PT3: TIMER3中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。

外设中断标志位寄存器 EIF2 (0xB2)

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 SPIIF: SPI总中断指示位, 只读;
 1= SPI产生中断, (清除具体的中断标志位后, 此位自动清除);
 0= SPI未产生中断。
- Bit6 I2CIF: I2C总中断指示位, 只读;
 1= I2C产生中断, 需软件清零 (清除具体的中断标志位后, 此位自动清除);
 0= I2C未产生中断。
- Bit5 --
- Bit4 ADCIF: ADC中断标志位;
 1= ADC转换完成, 需软件清零;
 0= ADC转换未完成。
- Bit3 PWMIF: PWM总中断指示位, 只读;
 1= PWM产生中断, (清除具体的中断标志位后, 此位自动清除);
 0= PWM未产生中断。
- Bit2 --
- Bit1 TF4: Timer4定时器溢出中断标志位;
 1= Timer4定时器溢出, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= Timer4定时器无溢出。
- Bit0 TF3: Timer3定时器溢出中断标志位;
 1= Timer3定时器溢出, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= Timer3定时器无溢出。

15.6 I²C 从动模式传输方式

本节介绍 I²C 内核中的所有可用传输模式。所有呈现的波形默认 I²C 自己的地址为 0x39 (“00111001”)。

15.6.1 单次接收

下图显示了在单数据期间由 I²C 接收的信号序列。单次接收序列：

- 开始条件；
- I²C 由 I²C 主机作为接收器寻址；
- 地址由 I²C 确认；
- 数据由 I²C 接收；
- 数据由 I²C 确认；
- 停止条件。

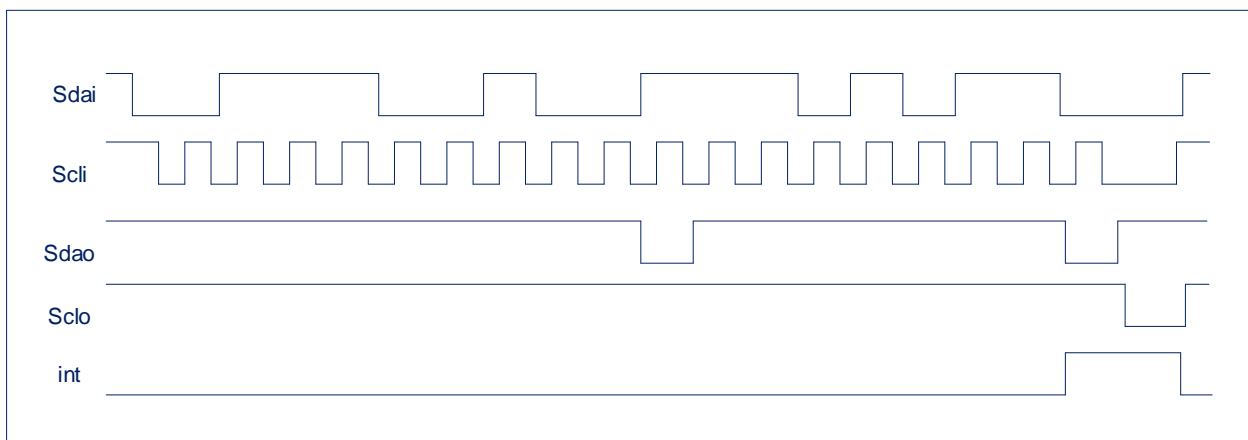


图 15-2：单次接收操作时序图

15.6.2 单次发送

下图显示了在单数据期间由 I²C 发送的信号序列。单次发送序列：

- 开始条件；
- I²C 由 I²C 主机作为发送器寻址；
- 地址由 I²C 确认；
- 数据由 I²C 传输；
- 数据不被 I²C 主机确认；
- 停止条件。

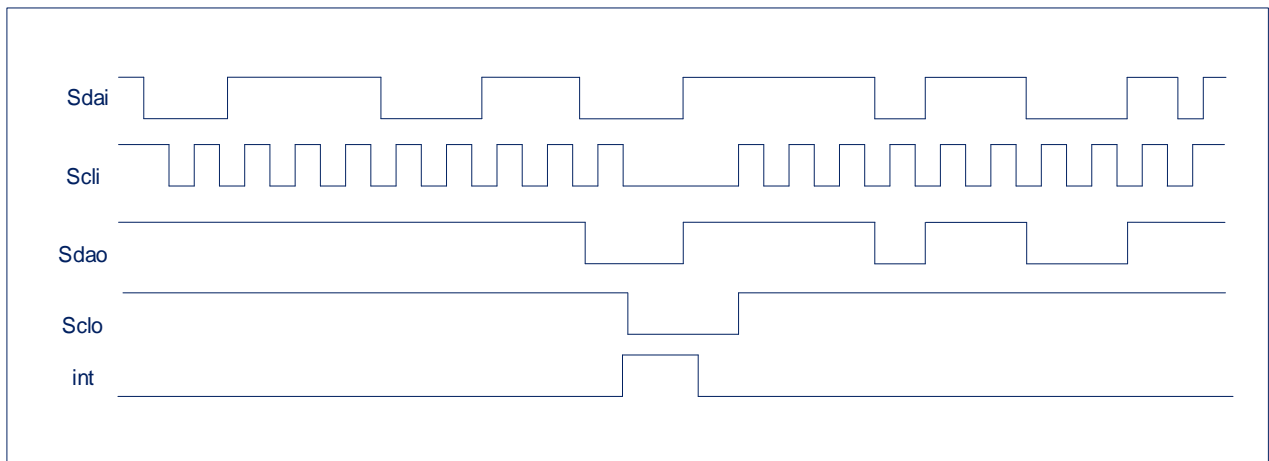


图 15-3: 单次发送操作时序图

15.6.3 连续接收

下图显示在连续接收数据期间由 I²C 接收的信号序列。连续接收序列：

- 开始条件。
- I²C 由 I²C 主机作为接收器寻址。
- 地址由 I²C 确认。
 - (1) 数据由 I²C 接收。
 - (2) 数据由 I²C 确认。
- 停止条件。

序列（1）和（2）重复直到停止条件发生。

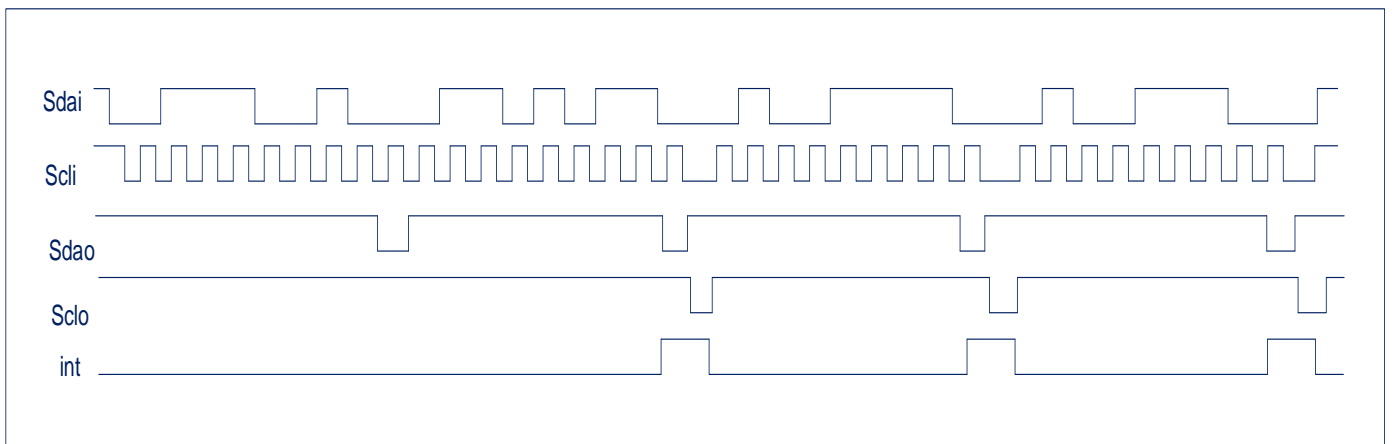


图 15-4：连续接收操作时序图

15.6.4 连续发送

下图显示了在连续发送数据期间由 I²C 发送的信号序列。连续发送序列：

- 发送条件。
- I²C 由 I²C 主机作为发送器寻址。
- 地址由 I²C 确认。
 - (1) 数据由 I²C 发送。
 - (2) 数据 I²C 主机确认数据。
 - (3) 最后的数据不被 I²C 主机确认。
- 停止条件。

重复序列 (1) 和 (2)，直到最后发送的数据未被 I²C 主机确认 (3)。

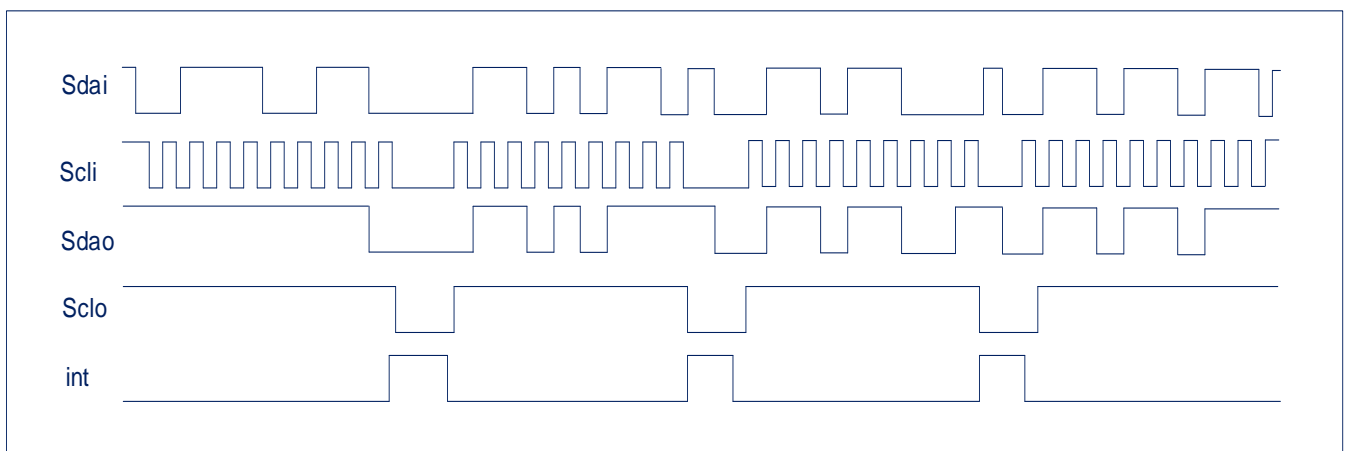


图 15-5：连续发送操作时序图

16. BRT 模块

16.1 简介

芯片内部有一个的 16 位波特率定时器 (baud-rate timer)，主要为 UART 模块提供时钟。

16.2 功能描述

BRT 内部有 1 个 16 位计数器，计数器的初值由{BRTDH, BRTDL}来加载，当 BRTEN=1 时，计数器开始工作，其时钟来源于预分频电路，预分频的时钟由 BRTCKDIV 决定。

计数器工作方式为递增计数，当 16 位计数器的值等于 FFFFH 时，BRT 计数器发生溢出。溢出后{BRTDH, BRTDL}的值自动加载到计数器中，然后重新进行计数。

BRT 计数器的溢出信号专门提供给 UART 模块作为波特率的时钟源。溢出时不会产生中断。也没有相应中断结构。

BRT 在调试模式下，其时钟不会停止，如果 UART 模块已经开始发送或接收数据，即使芯片进入暂停状态，UART 也会完成整个发送或接收的过程。

BRT 定时器溢出率：

$$BRTov = \frac{F_{sys}}{(65536 - \{BRTDH, BRTDL\}) \times 2^{BRTCKDIV}}$$

16.3 寄存器描述

BRTCON 寄存器

F5C0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTCON	BRTEN	--	--	--	--	CnNS0	CnPS1	CnPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 BRTEN: BRT定时器使能位;
 1= 使能;
 0= 禁止。
- Bit6~ Bit3 --
- Bit2~Bit0 BRTCKDIV<2:0> BRT定时器预分频选择位;
 000= Fsys/1;
 001= Fsys/2;
 010= Fsys/4;
 011= Fsys/8;
 100= Fsys/16;
 101= Fsys/32;
 110= Fsys/64;
 111= Fsys/128。

BRTDL 寄存器

F5C1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTDL	BRTDL7	BRTDL6	BRTDL5	BRTDL4	BRTDL3	BRTDL2	BRTDL1	BRTDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 BRTDL<7:0>: BRT定时器加载值低8位;

BRTDH 寄存器

F5C2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTDH	BRTDH7	BRTDH6	BRTDH5	BRTDH4	BRTDH3	BRTDH2	BRTDH1	BRTDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 BRTDH<7:0>: BRT定时器加载值高8位;

下表给出了在可变波特率模式下，波特率定时器溢出率作为 UART 时钟源的部分波特率相关信息：

SMODn=0, BRTCKDIV=0

	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz		
	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error
4800	65484	4808	-0.16	65432	4808	-0.16	65380	4808	-0.16
9600	65510	9615	-0.16	65484	9615	-0.16	65458	9615	-0.16
19200	65523	19231	-0.16	65510	19231	-0.16	65497	19231	-0.16
38400	--	--	--	65523	38462	-0.16	65516	37500	2.34
115200	--	--	--	--	--	--	--	--	--
250000	--	--	--	--	--	--	--	--	--
500000	--	--	--	--	--	--	--	--	--

SMODn=1, BRTCKDIV=0

	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz		
	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error
4800	65432	4808	-0.16	65328	4808	-0.16	65224	4792	0.16
9600	65484	9615	-0.16	65432	9615	-0.16	65380	9615	-0.16
19200	65510	19231	-0.16	65484	19231	-0.16	65458	19231	-0.16
38400	65523	38462	-0.16	65510	38462	-0.16	65497	38462	-0.16
115200	--	--	--	--	--	--	65523	115385	-0.16
250000	--	--	--	--	--	--	--	--	--
500000	--	--	--	--	--	--	--	--	--
1000000	--	--	--	--	--	--	--	--	--

17. UARTn 模块 (n=0,1)

17.1 简介

芯片内部包含两个 UART 模块：UART0 与 UART1，这两个具有完全相同的功能。

串行端口是全双工的，意味着它可以同时发送和接收。接收模块有双缓冲，这意味着它可以在从接收寄存器读取上一次接收到的字节之前，开始接收下一个字节。写入 SBUF_n 数据加载到发送寄存器，而读取 SBUF_n 是访问一个具有独立物理地址的接收寄存器。

串口共有四种操作模式：一个同步和三个异步模式。模式 2 和 3 具有多处理器通信的特殊功能，通过将 SCON_n 寄存器中的 SMn2 位置 1 来使能该功能。主机处理器首先发送识别目标从机的地址字节。地址字节与数据字节不同，因为地址字节中第 9 位为 1，数据字节为 0。在 SMn2=1 时，从机不会被数据字节中断。地址字节将中断所有从机。寻址的从机将清除其 SMn2 位，并准备接收将要来的数据字节。未被寻址的从机使 SMn2 置 1 并忽略传入的数据。

17.2 UARTn 端口配置

使用 UARTn 模块前需要先将相应端口配置成 UARTn 的 TXD_n 与 RXD_n 通道（任意 GPIO 均可配置为 UARTn 通道），例如：

```
P25CFG = 0x08; //选择 P2.5 配置成 TXD0 通道
```

```
P26CFG = 0x09; //选择 P2.6 配置成 RXD0 通道，主控同步模式时端口自动配置为开漏并且有上拉电阻
```

```
P35CFG = 0x0A; //选择 P3.5 配置成 TXD1 通道
```

```
P21CFG = 0x0B; //选择 P2.1 配置成 RXD1 通道，主控同步模式时端口自动配置为开漏并且有上拉电阻  
使用时，建议先设置好工作模式，然后再将相应的口配置成串口。
```

17.3 UARTn 波特率

UARTn 在模式 0 时，波特率固定为系统时钟的十二分频 ($F_{sys}/12$)；在模式 2 时，波特率固定为系统时钟的三十二分频或者六十四分频 ($F_{sys}/32$ 、 $F_{sys}/64$)；在模式 1 和模式 3，波特率由定时器 Timer1 或者 Timer4 或者 Timer2 或者 BRT 模块产生，芯片选择何种定时器作为波特率时钟源由寄存器 FUNCCR 决定。

{FUNCCR[2],FUNCCR[0]}=00 时，选择 Timer1 作为 UART0 的波特率发生器；
 {FUNCCR[2],FUNCCR[0]}=01 时，选择 Timer4 作为 UART0 的波特率发生器；
 {FUNCCR[2],FUNCCR[0]}=10 时，选择 Timer2 作为 UART0 的波特率发生器；
 {FUNCCR[2],FUNCCR[0]}=11 时，选择 BRT 作为 UART0 的波特率发生器。
 {FUNCCR[3],FUNCCR[1]}=00 时，选择 Timer1 作为 UART1 的波特率发生器；
 {FUNCCR[3],FUNCCR[1]}=01 时，选择 Timer4 作为 UART1 的波特率发生器；
 {FUNCCR[3],FUNCCR[1]}=10 时，选择 Timer2 作为 UART1 的波特率发生器；
 {FUNCCR[3],FUNCCR[1]}=11 时，选择 BRT 作为 UART1 的波特率发生器。

Timer1 或 Timer4 工作在 8 位自动重装模式下波特率的公式：

$$BaudRate = \frac{F_{sys} \times 2^{SMODn}}{32 \times (4 \times 3^{1-TxM}) \times (256-THx)} \quad (x=1,4)$$

SMODn 为波特率选择位，由寄存器 PCON 设置。T1M 为定时器 1 时钟选择位，由寄存器 CKCON[4] 设置，T4M 为定时器 4 时钟选择位，由寄存器 T34MOD[6] 设置。即 Timer1 或 Timer4 在相应波特率下的 TH1/TH4 的值应设置为：

$$THx = 256 - \frac{F_{sys} \times 2^{SMODn}}{32 \times (4 \times 3^{1-TxM}) \times BaudRate} \quad (x=1,4)$$

Timer2 工作在溢出自动重装模式下波特率的公式：

$$BaudRate = \frac{F_{sys} \times 2^{SMODn}}{32 \times (12 \times 2^{T2PS}) \times (65536 - \{RLDH, RLDL\})}$$

T2PS 为定时器 2 时钟预分频选择位，由寄存器 T2CON[7] 设置。即 Timer2 在相应波特率下 {RLDH, RLDL} 的值应设置为：

$$\{RLDH, RLDL\} = 65536 - \frac{F_{sys} \times 2^{SMODn}}{32 \times (12 \times 2^{T2PS}) \times BaudRate}$$

BRT 作为波特率发生器时，波特率公式：

$$BaudRate = \frac{F_{sys} \times 2^{SMODn}}{32 \times (65536 - \{BRTDH, BRTDL\}) \times 2^{BRTCKDIV}}$$

BRTCKDIV 为 BRT 定时器预分频选择位，由寄存器 BRTCON 设置。即 BRT 在相应波特率下 {BRTDH, BRTDL} 的值应设置为：

$$\{BRTDH, BRTDL\} = 65536 - \frac{F_{sys} \times 2^{SMODn}}{32 \times 2^{BRTCKDIV} \times BaudRate}$$

下表为在可变波特率模式下，定时器 1/定时器 4 的 8 位自动重装模式下的部分波特率相关信息：

SMODn=0

	Fsys=8MHz				Fsys=16MHz				Fsys=24MHz			
	TH1 or TH4	T1M or T4M	Actual Rate	% Error	TH1 or TH4	T1M or T4M	Actual Rate	% Error	TH1 or TH4	T1M or T4M	Actual Rate	% Error
4800	243	1	4808	-0.16	230	1	4808	-0.16	217	1	4808	-0.16
9600	--	--	--	--	247	1	9615	-0.16	236	1	9375	2.34
19200	--	--	--	--	--	--	--	--	246	1	18750	2.34
38400	--	--	--	--	--	--	--	--	251	1	37500	2.34

SMODn=1

	Fsys=8MHz				Fsys=16MHz				Fsys=24MHz			
	TH1 or TH4	T1M or T4M	Actual Rate	% Error	TH1 or TH4	T1M or T4M	Actual Rate	% Error	TH1 or TH4	T1M or T4M	Actual Rate	% Error
4800	230	1	4808	-0.16	204	1	4808	-0.16	178	1	4808	-0.16
9600	243	1	9615	-0.16	230	1	9615	-0.16	217	1	9615	-0.16
19200	--	--	--	--	243	1	19230	-0.16	236	1	18750	2.34
38400	--	--	--	--	--	--	--	--	246	1	37500	2.34

17.4 UARTn 寄存器

UARTn 具有与标准 8051 UART 相同的功能。UARTn 相关寄存器是：SBUFn、SCONn、PCON (0x87)、IE (0xA8) 和 IP (0xB8)。UARTn 数据缓冲器 (SBUFn) 由两个独立的寄存器组成：发送和接收寄存器。写入 SBUFn 的数据将在 UARTn 输出寄存器中设置此数据并开始传输；读取 SBUFn 的数据将从 UARTn 接收寄存器中读取数据。

SCON0 寄存器支持位寻址操作，SCON1 寄存器不支持位寻址操作。使用汇编语言时需注意。

UART 缓冲寄存器 SBUFn

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBUFn	BUFFERn7	BUFFERn6	BUFFERn5	BUFFERn4	BUFFERn3	BUFFERn2	BUFFERn1	BUFFERn0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

寄存器 SBUF 地址：0x99；寄存器 SBUF1 地址：0xEB；

Bit7~Bit0 BUFFERn<7:0>: 缓冲数据寄存器。
 写: UARTn开始发送数据。
 读: 读取接收到的数据。

UART 控制寄存器 SCONn

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCONn	UnSM0	UnSM1	UnSM2	UnREN	UnTB8	UnRB8	TIn	RIn
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 SCON0 地址：0x98；寄存器 SCON1 地址：0xEA。

Bit7~Bit6 UnSM0- UnSM1: 多机通信控制位；
 00= 主控同步模式；
 01= 8位异步模式，波特率可变；
 10= 9位异步模式，波特率为Fsys/32或者Fsys/64；
 11= 9位异步模式，波特率可变。

Bit5 UnSM2: 多机通信控制位；
 1= 使能；
 0= 禁止。

Bit4 UnREN: 接收使能位；
 1= 使能；
 0= 禁止。

Bit3 UnTB8: 发送数据的第9位，主要用于9位异步模式的发送；
 1= 第9位数据为1；
 0= 第9位数据为0。

Bit2 UnRB8: 接收数据的第9位，主要用于9位异步模式的发送；
 1= 接收到的第9位数据为1；
 0= 接收到的第9位数据为0。

Bit1 TIn: 发送中断标志位（需要软件清零）；
 1= 说明发送缓冲器已空，可以发送一下帧数据。
 0= --

Bit0 RIn: 接收中断标志位（需要软件清零）；
 1= 说明接收缓冲器已满，读取后可以接收下一帧数据。
 0= --

UARTn 模式如下表:

SMn0	SMn1	模式	描述	波特率
0	0	0	Shift register	Fsys/12
0	1	1	8-Bit UART	variable
1	0	2	9-Bit UART	Fsys/32 or /64
1	1	3	9-Bit UART	variable

UARTn 波特率

模式	波特率
Mode0	Fsys/12
Mode1, 3	由 Timer4/Timer1/Timer2/BRT 控制, 参见 16.3 节
Mode2	SMODn=0: Fsys/64 SMODn=1: Fsys/32

SMODn 位在电源管理控制寄存器 PCON 寄存器中:

0x87	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	SMOD1	--	--	--	SWE	STOP	IDLE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 SMOD0: UART0波特率倍增位;
 1= UART0波特率加倍;
 0= UART0波特率正常。
- Bit6 SMOD1: UART1波特率倍增位;
 1= UART1波特率加倍;
 0= UART1波特率正常。
- Bit5 --
- Bit4~Bit3 保留位: 必须为0。
- Bit2 SWE: STOP 状态功能唤醒使能位;
 (不论 SWE 为何值, 系统均可由掉电复位或使能的外部复位重新启动)
 0= 禁止功能唤醒;
 1= 允许功能唤醒 (可由外部中断唤醒及定时唤醒)。
- Bit1 STOP: 休眠状态控制位;
 1= 进入休眠状态 (退出STOP模式自动清零);
 0= 未进入休眠状态。
- Bit0 IDLE: 空闲状态控制位;
 1= 进入空闲状态 (退出IDLE模式自动清零);
 0= 未进入空闲状态。

17.5 UARTn 中断

UART0 的中断号为 4, 其中断向量为 0x0023。UART1 的中断号为 6, 其中断向量为 0x0033。使能 UARTn 中断必须将其使能位 ESn 置 1, 且将总中断使能位 EA 置 1。

若 UARTn 相关的中断使能均打开, TIn=1 或者 RIn=1 时, CPU 将进入相应的中断服务程序。TIn/RIn 与 ESn 的状态无关。且需要软件清零。

中断屏蔽寄存器 IE

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	EA: 全局中断允许位; 1= 允许所有未被屏蔽的中断; 0= 禁止所有中断。
Bit6	ES1: UART1中断允许位; 1= 允许UART1中断; 0= 禁止UART1中断。
Bit5	ET2: TIMER2总中断允许位; 1= 允许TIMER2所有中断; 0= 禁止TIMER2所有中断。
Bit4	ES0: UART0中断允许位; 1= 允许UART0中断; 0= 禁止UART0中断。
Bit3	ET1: TIMER1中断允许位; 1= 允许TIMER1中断; 0= 禁止TIMER1中断。
Bit2	EX1: 外部中断1中断允许位; 1= 允许外部中断1中断; 0= 禁止外部中断1中断。
Bit1	ET0: TIMER0中断允许位; 1= 允许TIMER0中断; 0= 禁止TIMER0中断。
Bit0	EX0: 外部中断0中断允许位; 1= 允许外部中断0中断; 0= 禁止外部中断0中断。

中断优先级控制寄存器 IP

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7	--
Bit6	PS1: UART1中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit5	PT2: TIMER2中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit4	PS0: UART0中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit3	PT1: TIMER1中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit2	PX1: 外部中断1中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit1	PT0: TIMER0中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。
Bit0	PX0: 外部中断0中断优先级控制位; 1= 设置为高级中断; 0= 设置为低级中断。

UART 控制寄存器 SCOnn

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCOnn	SMn0	SMn1	SMn2	RENn	TBn8	RBn8	TIn	RIn
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

寄存器 SCOn0 地址：0x98；寄存器 SCOn1 地址：0xEA。

Bit7~Bit6	SMn0-SMn1:	多机通信控制位； 00= 主控同步模式； 01= 8位异步模式，波特率可变； 10= 9位异步模式，波特率为Fsys/32或者 Fsys/64； 11= 9位异步模式，波特率可变。
Bit5	SMn2:	多机通信控制位； 1= 使能； 0= 禁止。
Bit4	RENn:	接收使能位； 1= 使能； 0= 禁止。
Bit3	TBn8:	发送数据的第9位，主要用于9位异步模式的发送； 1= 第9位数据为1； 0= 第9位数据为0。
Bit2	RBn8:	接收数据的第9位，主要用于9位异步模式的发送； 1= 接收到的第9位数据为1； 0= 接收到的第9位数据为0。
Bit1	TIn:	发送中断标志位（需要软件清零）； 1= 说明发送缓冲器已空，可以发送一下帧数据。 0= --
Bit0	RIn:	接收中断标志位（需要软件清零）； 1= 说明接收缓冲器已满，读取后可以接收下一帧数据。 0= --

17.6 UARTn 模式 (n=0/1)

17.6.1 模式 0-同步模式

引脚 RXDn 作为输入或者输出, TXDn 作为时钟输出。TXDn 输出是一个移位时钟。波特率固定为系统时钟频率的 1/12。8 位以 LSB 优先传输。通过设置 SCONn 中的标志来初始化接收, 设置如下: R1n = 0 和 RENn = 1。

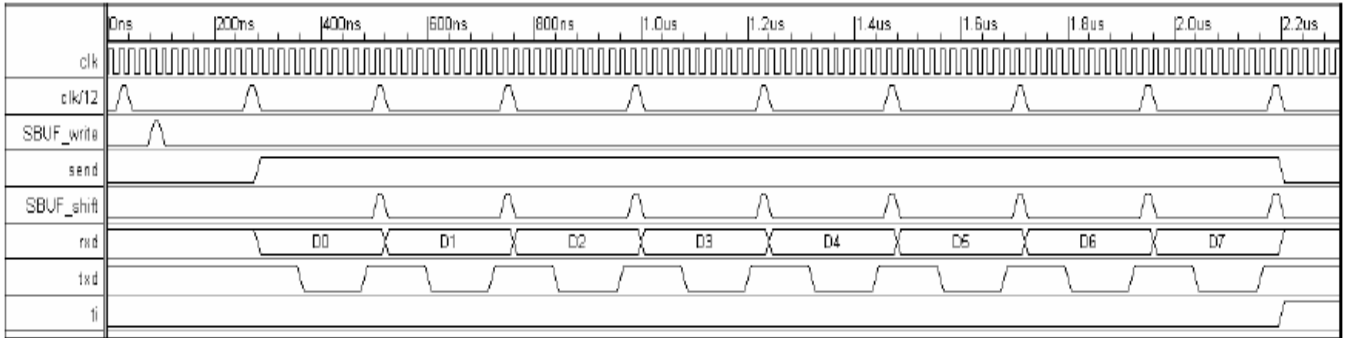


图 17-1: 通信模式 0 时序图

17.6.2 模式 1-8 位异步模式 (可变波特率)

引脚 RXDn 用作输入, TXDn 用作串行输出。发送 10 位: 起始位 (始终为 0), 8 位数据 (LSB 优先) 和停止位 (始终为 1)。接收时, 起始位同步传输, 通过读取 SBUFn 可以获得 8 个数据位, 停止位在 SCONn 中设置标志 RBn8。波特率是可变的, 取决于 TIMER1、TIMER2、TIMER4 和 BRT 模式。

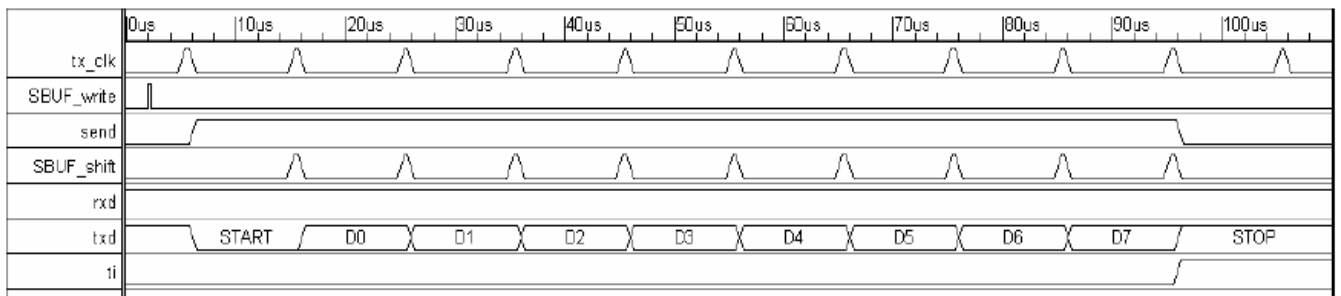


图 17-2: UARTn 通信模式 1 时序图

17.6.3 模式 2-9 位异步模式（固定波特率）

此模式与模式 1 类似，但有两个不同之处。波特率固定为 CLK 时钟频率的 1/32 或 1/64，11 位收发：起始位（0），8 位数据（LSB 优先），可编程第 9 位和停止位（1）。第 9 位可用于控制 UARTn 接口的奇偶校验：在发送时，SCONn 中的位 TBn8 作为第 9 位输出，在接收时，第 9 位影响 SCONn 中的 RBn8。

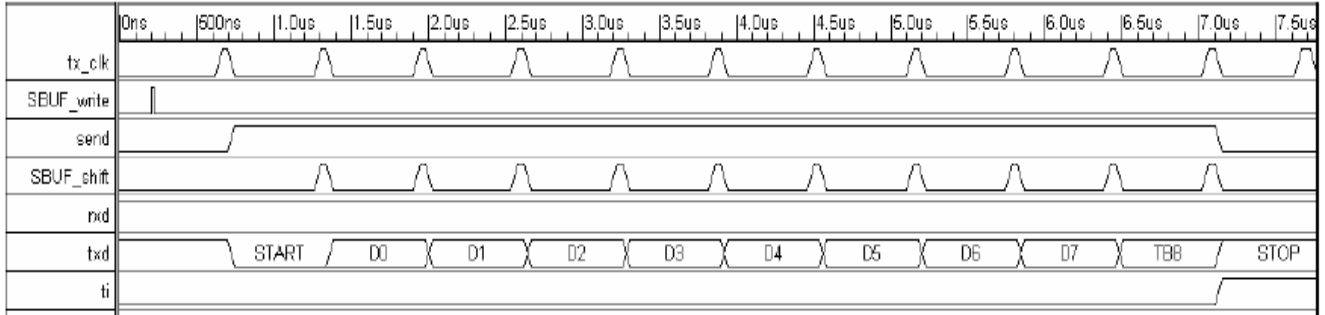


图 17-3: UARTn 通信模式 2 时序图

17.6.4 模式 3-9 位异步模式（可变波特率）

模式 2 和模式 3 之间的唯一区别是模式 3 中的波特率是可变的。当 REN0=1 时，数据接收使能。波特率是可变的并且取决于定时器 1 或定时器 4 模式。

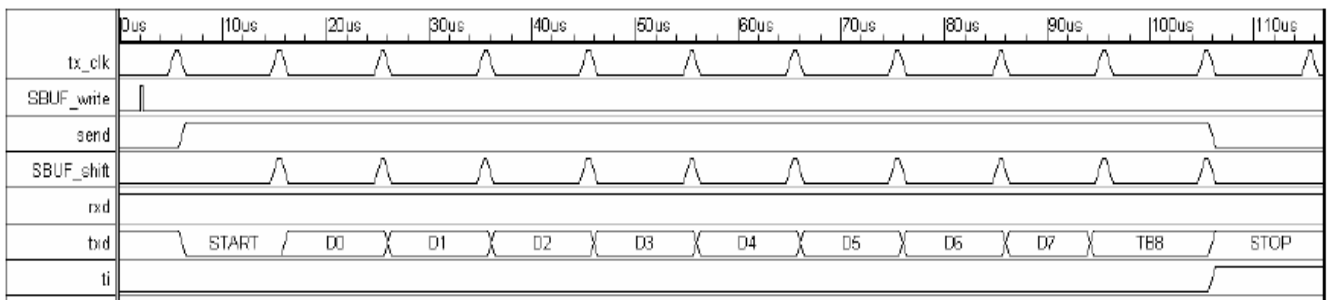


图 17-4: UART0 通信模式 3 时序图

18. SPI 模块

18.1 SPI 简介

此 SPI 是一个完全可配置的 SPI 主机/从机设备，允许用户配置串行时钟信号 SCLK 的极性和相位。SPI 允许 MCU 与串行外围设备进行通信。它还能够多主机系统中进行处理器间通信。串行时钟线（SCLK）与在两个独立串行数据线上信息的移位和采样同步。SPI 数据同时发送和接收。SPI 是一种可以在各种工艺技术中实现的技术独立的设计。

SPI 系统具有足够的灵活性，可以与多家制造商的许多标准产品外设直接连接。系统可以配置为主机设备或从机设备。为了适应大多数可用的同步串行外设，时钟控制逻辑允许选择时钟极性，并选择两种不同的时钟协议。当 SPI 被配置为主机设备时，软件为串行时钟在八种不同的比特率中选择其一。

SPI 自动驱动由 SSCR（从机选择控制寄存器）从机选择输出 NSS 选择，以及寻址 SPI 从机设备来交换串行数据。包含逻辑错误检测以支持处理器间通信。写入冲突检测器指示何时在传送过程中将数据写入串行移位寄存器。如果多个 SPI 设备同时尝试成为总线主机，则多主机设备模式故障检测器将自动禁用 SPI 输出驱动器。

SPI 的特性如下：

- ◆ 全双工同步串行数据传输；
- ◆ 支持主机/从机模式；
- ◆ 支持多主机系统；
- ◆ 系统错误检测；
- ◆ 产生中断；
- ◆ 支持速度高达系统时钟的 1/4 ($F_{SYS} \leq 24\text{MHz}$)；
- ◆ 比特率产生系统时钟的 1/4、1/8、1/16、1/32、1/64、1/128、1/256、1/512；
- ◆ 支持四种传输格式；
- ◆ 简单接口允许轻松连接到微控制器。

18.2 SPI 端口配置

使用 SPI 功能必须先配置 SPI 相关寄存器, 然后再将相应端口配置成 SPI 通道 (任意 GPIO 均可配置为 SPI 通道), 如将 P1.4 配置成 NSS 通道、P1.5 配置成 SCLK 通道、P1.6 配置成 MOSI 通道、P1.7 配置成 MISO 通道, 配置代码如下:

```
P14CFG = 0x0E; //选择 P1.4 配置成 NSS 通道
P15CFG = 0x0F; //选择 P1.5 配置成 SCLK 通道
P16CFG = 0x10; //选择 P1.6 配置成 MOSI 通道
P17CFG = 0x11; //选择 P1.7 配置成 MISO 通道
```

配置为 SCLK、MOSI、MISO 和 NSS 口, 其上拉电阻与开漏输出的开关强制关闭。多机 SPI 通讯结构示意图如下图所示:

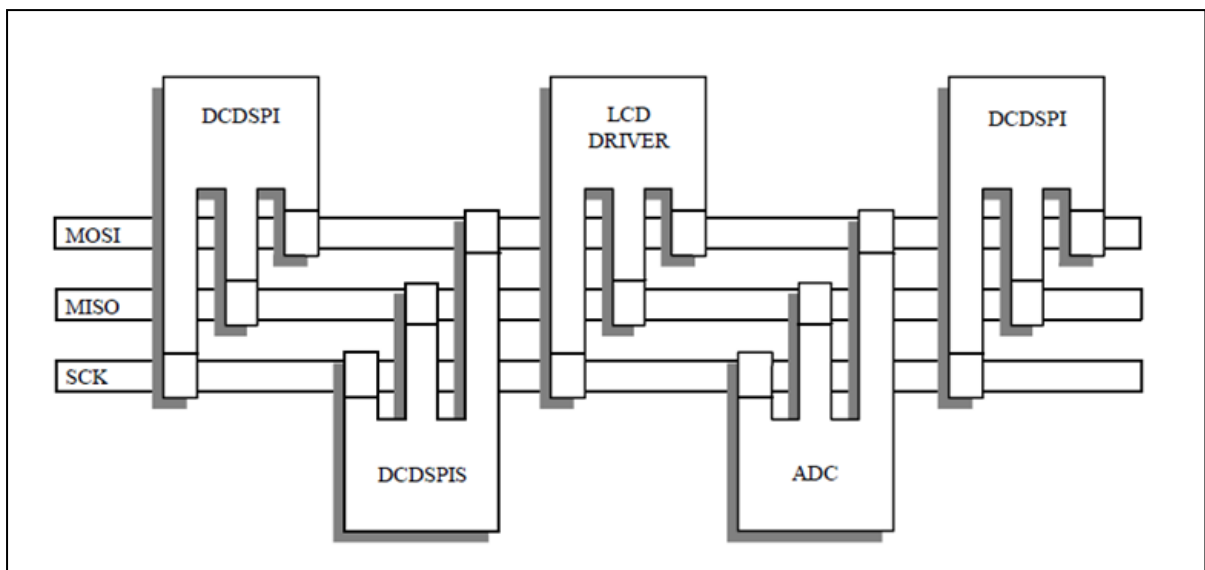


图 18-1: 多机 SPI 通讯

18.3 SPI 硬件描述

发生 SPI 传输时，当一个数据引脚移出一个 8 位字符时，同时另一个数据引脚上移入其他的 8 位字符。查看此传输的另一种方法是，主机设备中的 8 位移位寄存器和从机设备中的另一个 8 位移位寄存器作为循环 16 位移位寄存器连接。当发生转移时，该分布式移位寄存器被移位 8 位；从而有效的交换了主机从机的字符。

SPI 系统中的中心元件是包含移位寄存器和读取数据缓冲区的模块。系统在发送方向上为单缓冲器，在接收方向上为双缓冲器。这意味着直到前一个数据传输完，新的数据才能写入移位器；然而，接收的数据被传送到并行读取数据缓冲器中，所以移位器可以自由地接收第二串行字符。只要在下一个串行字符准备好传输之前，第一个字符从读取数据缓存器中读出，就不会发生覆盖状态。

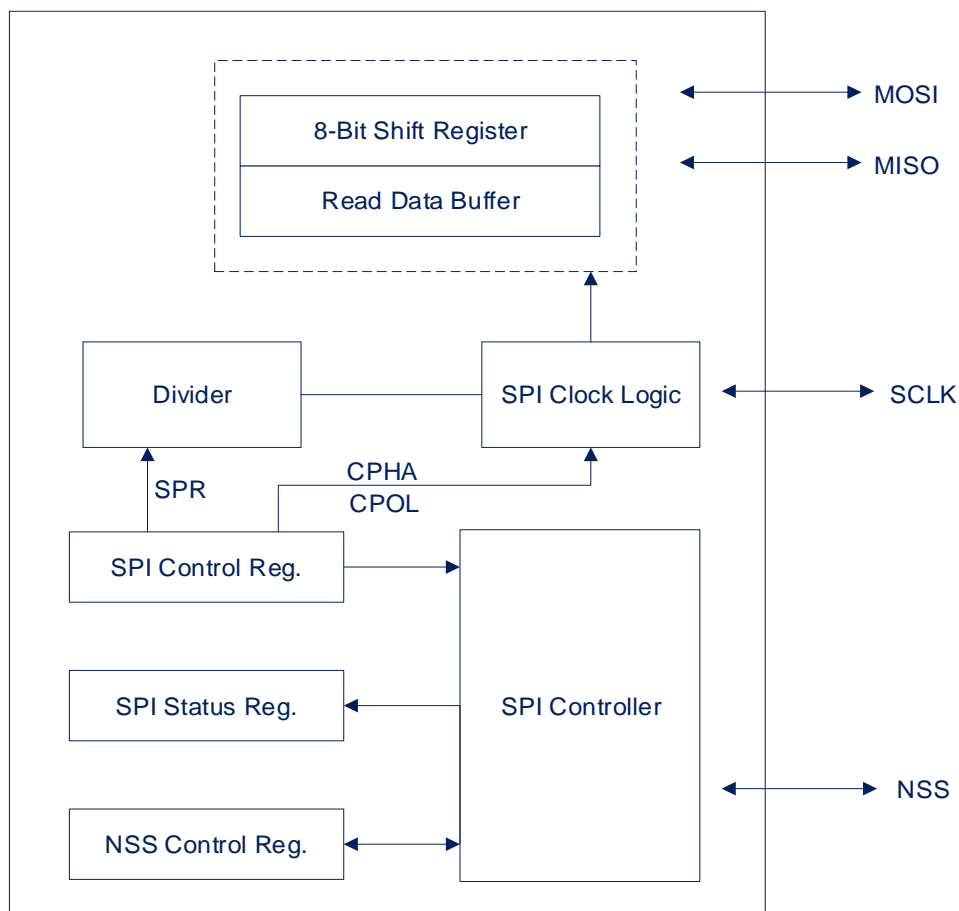


图 18-2: SPI 框图

与 SPI 相关联的引脚有：NSS，SCLK，MOSI，MISO。

主机模式下的 NSS 输入引脚用于检测模式故障错误。此引脚上的低电平表示多主机系统中的其他设备已成为主机设备，并尝试将 SPI 选择为从机设备。从机模式下的 NSS 输入引脚用于使能传输。

在主机模式下，SCLK 引脚用作 SPI 时钟信号参考。当主机设备启动传输时，SCLK 引脚上会自动生成八个时钟周期。

- 当 SPI 被配置为从机设备时，SI 引脚是从机设备输入数据线，SO 是从机设备输出数据线。
- 当 SPI 被配置为主机设备时，MI 引脚是主机设备输入数据线，MO 是主机设备输出数据线。

18.4 SPI 相关寄存器

18.4.1 SPI 控制寄存器 SPCR

0xEC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR	--	SPEN	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	1	0	0

Bit7	--
Bit6	SPEN: SPI模块使能位; 1= 使能; 0= 禁止。
Bit5	SPR2: SPI时钟频率选择位第[2]位。
Bit4	MSTR: SPI模式选择位; 1= 主动模式; 0= 从动模式。
Bit3	CPOL: SPI时钟极性选择位; 1= SCLK空闲时为高; 0= SCLK空闲时为低。
Bit2	CPHA: SPI时钟相位选择位。
Bit1~Bit0	SPR1-SPR0: SPI时钟频率选择位[1:0] (频率控制详见下面表格)

SPR2-SPR0 控制 SPI 时钟分频

SPR2	SPR1	SPR0	System clock divided by
0	0	0	4
0	0	1	8
0	1	0	16
0	1	1	32
1	0	0	64
1	0	1	128
1	1	0	256
1	1	1	512

18.4.2 SPI 数据寄存器 SPDR

0xEE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPDR	SPIDATA7	SPIDATA6	SPIDATA5	SPIDATA4	SPIDATA3	SPIDATA2	SPIDATA1	SPIDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0	SPIDATA: SPI发送或接收的数据。 写操作: 写将要发送的数据(发送顺序由高位依次到低位)。 读操作: 已经接收到的数据。
-----------	---

18.4.3 SPI 从器件选择控制寄存器 SSCR

控制寄存器可以随时读取或写入。它用于配置在确认 SPI 主机传输时应驱动哪个从机选择输出。当 SPI 主机传输启动时，SSCR 寄存器的内容将自动分配给 NSS 引脚。

0xEF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSCR	--	--	--	--	--	--	--	SSO0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit1 -- 保留，须全为1。
 Bit0 SSO0: SPI从器件选择控制位。
 0= SPI主机传输启动时，NSS输出0。
 1= SPI主机传输启动时，NSS输出1。

18.4.4 SPI 状态寄存器 SPSR

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	SPISIF	WCOL	--	--	--	--	--	SSCEN
R/W	R	R	--	R	--	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 SPISIF: SPI传输完毕中断标志位，只读；
 1= SPI传输完毕（先读SPSR，再读/写SPDR后清零）；
 0= SPI未传输完毕。
 Bit6 WCOL: SPI写冲突中断标志位，只读；
 1= SPI传输未完成时产生写SPDR操作冲突（先读SPSR，再读/写SPDR后清零）；
 0= 无写冲突。
 Bit5 --
 Bit4 --
 Bit3~Bit2 --
 Bit1 保留: 必须为0
 Bit0 SSCEN: SPI主控模式NSS输出控制位。
 1= SPI处于空闲状态时，NSS输出高电平；
 0= NSS输出寄存器SSCR的内容。

SPI 状态寄存器（SPSR）包含指示传输完成或系统错误发生的标志。当相应的事件发生并通过软件按顺序清除时，所有标志都将自动设置。通过读取 SPSR，然后访问 SPDR，SPISIF 和 WCOL 将会自动清除。

SSCEN 位是自动从机选择输出的使能位。当 SSCEN 置 1，传输正在进行时 NSS 线输出 SSCR 寄存器的内容，传输空闲时 NSS 为高电平。当 SSCEN 位清零时，NSS 线总是显示 SSCR 寄存器的内容。

18.5 SPI 主控模式

当 SPI 配置为主机模式时，通过写入 SPDR 寄存器启动传输。当新字节写入 SPDR 寄存器时，SPI 开始传输。串行时钟 SCLK 由 SPI 生成，在主机模式下 SPI 使能，SCLK 输出。

主机模式下的 SPI 可以通过 NSS 线选择 SPI 从机设备。NSS 线-从机选择输出线加载了寄存器 SSCR[0] 的内容。SPSR 寄存器的 SSCEN 位在自动 NSS 线路控制和软件控制之间进行选择。将 SSCEN 位置于主机模式下，当 SSCEN 置 1，传输正在进行时 NSS 线输出 SSCR 寄存器的内容，传输空闲时 NSS 为高电平。当 SSCEN 位清零时，NSS 线由软件控制，并且始终显示 SSCR 寄存器的内容，无论传输正在进行还是 SPI 处于空闲状态。

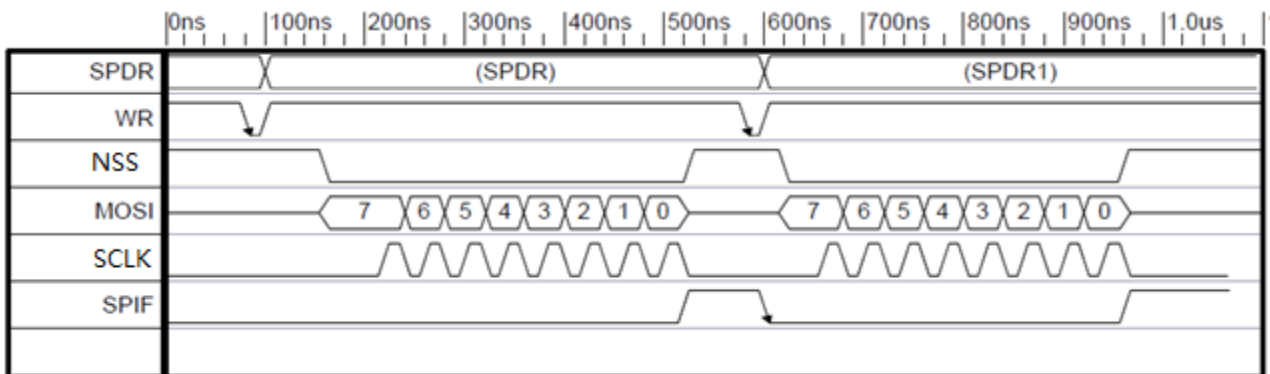


图 18-3: 自动从机选择线使用

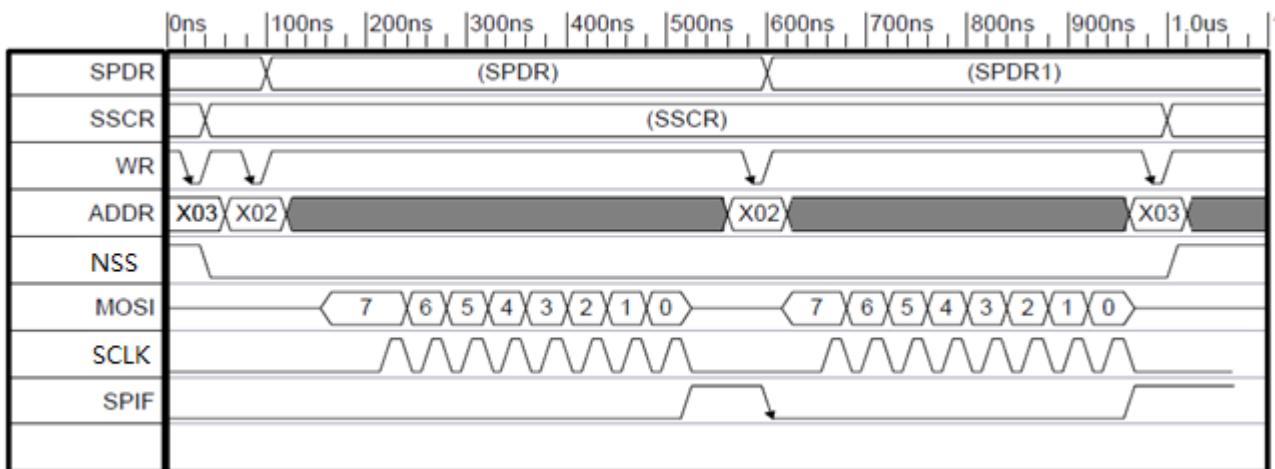


图 18-4: 软件控制 NSS 线

18.5.1 写冲突错误

如果在传输过程中写入了 SPI 数据寄存器，则会发生写冲突。传输继续不受干扰，导致错误的写入数据不会写入移位器。写冲突由 SPSR (3) 寄存器中的 WCOL 标志指示。

当 WCOL 错误发生时，WCOL 标志由硬件自动置 1。要清除 WCOL 位，用户应执行以下步骤：

- 读取 SPSR 寄存器的内容
- 访问 SPDR 寄存器（读或写）

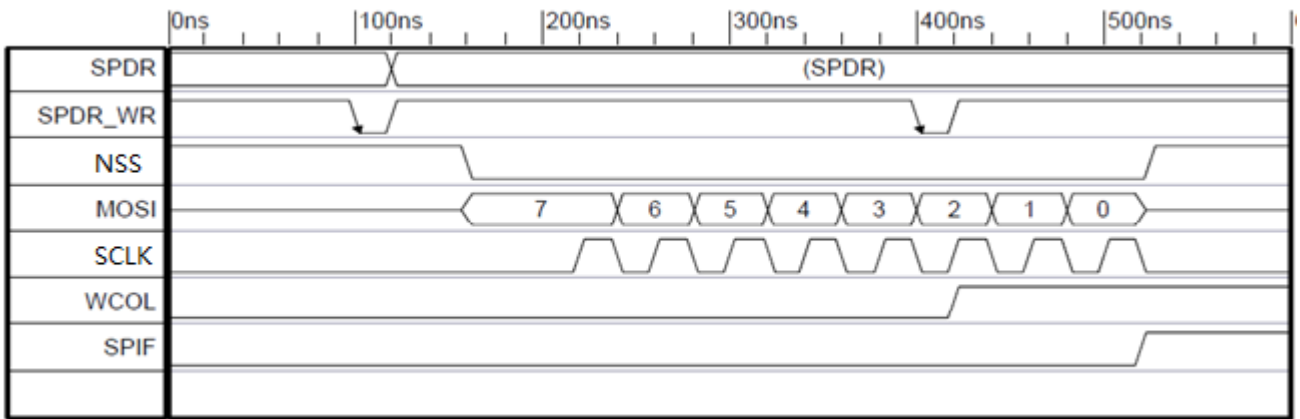


图 18-6: SPI 主模式下的写冲突错误

写冲突产生的具体条件为：在数据传输过程中，NSS 为低时，第一个数据开始发送时刻到第 8 个 SCLK 下降沿期间。如果在此期间写 SPDR，则会出现写冲突，WCOL 将置 1。

注意：开始发送数据时，在写 SPDR 之后，NSS 并不是立即变为低电平，需要等待最多一个 SPI 时钟后才开始为低。NSS 为低后，需要等待一个系统时钟才开始发送第一个数据，此时才进入真正的数据传输状态。在写 SPDR 到进入真正的数据传输状态期间，再次写入 SPDR 并不会产生写冲突。但该操作会更新准备发送的数据。如果有多次写 SPDR 的操作，发送的数据将会是最后一次写入 SPDR 的值。

由于 SPI 只有一个发送缓冲器，建议在写 SPDR 之前判断上一次的数据是否发送完毕，确定发送完成后再写 SPDR 寄存器，以防止产生写冲突。

18.6 SPI 从动模式

当配置为 SPI 从机设备时，SPI 传输由外部 SPI 主机模块通过使用 SPI 从机选择输入启动，并生成 SCLK 串行时钟。

在传输开始之前，需确定哪个 SPI 从机将用于交换数据。NSS 被使用，连接到 SCLK 线的时钟信号将使 SPI 从机设备转移到 MOSI 线的接收移位寄存器内容，并用发送器移位寄存器的内容驱动 MISO 线。当所有 8 位被移入/移出时，SPI 通过设置 IRQ 输出来产生中断请求。移位寄存器的内容驱动 MISO 线。

在 SPI 从机模式下，只能有一个传输错误（写冲突错误）。

18.6.1 被寻址错误

在从机模式下，只有写冲突错误能被 SPI 检测到。

当 SPI 传输进行时执行 SPDR 寄存器写操作，会发生写冲突错误。

在从机模式下，当 CPHA 清零时，只要 NSS 从机选择线被驱动为低电平，即使已经传输了所有位，也可能发生写冲突错误。这是因为没有明确指定传输开始，并且在全字节传输后 NSS 被驱动为低电平可能指示下一个字节传输的开始。

18.6.2 写冲突错误

如果在传输过程中写入 SPI 数据寄存器，则会发生写冲突。传输继续不受干扰，并且导致错误的写入数据不会写入移位器。写冲突由 SPSR (3) 寄存器中的 WCOL 标志指示。

当 WCOL 错误发生时，WCOL 标志由硬件自动置 1。要清除 WCOL 位，用户应该执行以下顺序：

- 读取 SPSR 寄存器的内容；
- 访问 SPDR 寄存器（读或写）。

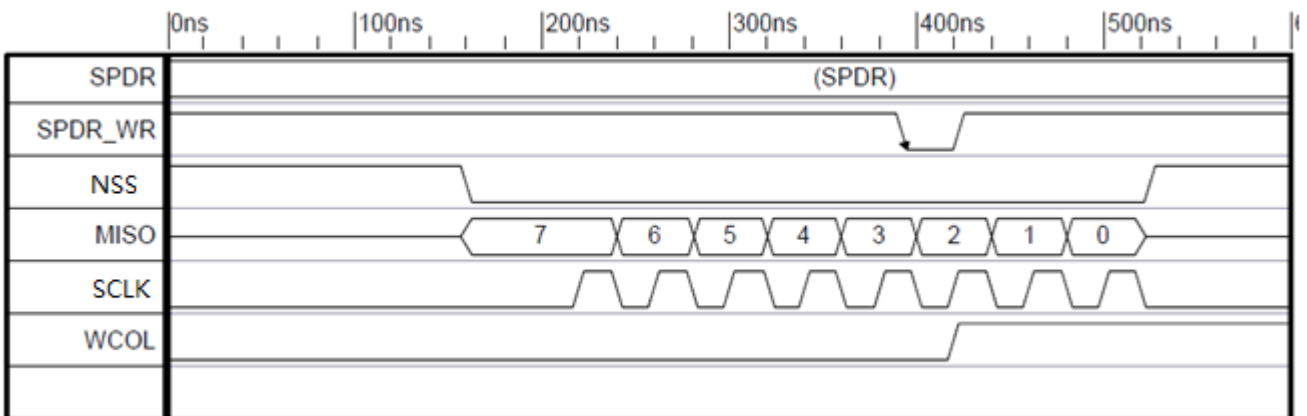


图 18-7：写冲突错误--SPI 从模式- SPDR write during transfer

下图显示了 WCOL 的生成，以防 CPHA 被清除。如图所示，WCOL 生成是由任一 NSS 线清零的 SPDR 寄存器写入引起的。即使 SPI 主机没有生成串行时钟 SCLK 也可以完成。这是因为没有明确指定传输开始，并且在全字节传输后 NSS 被驱动为低电平可能指示下一个字节传输的开始。

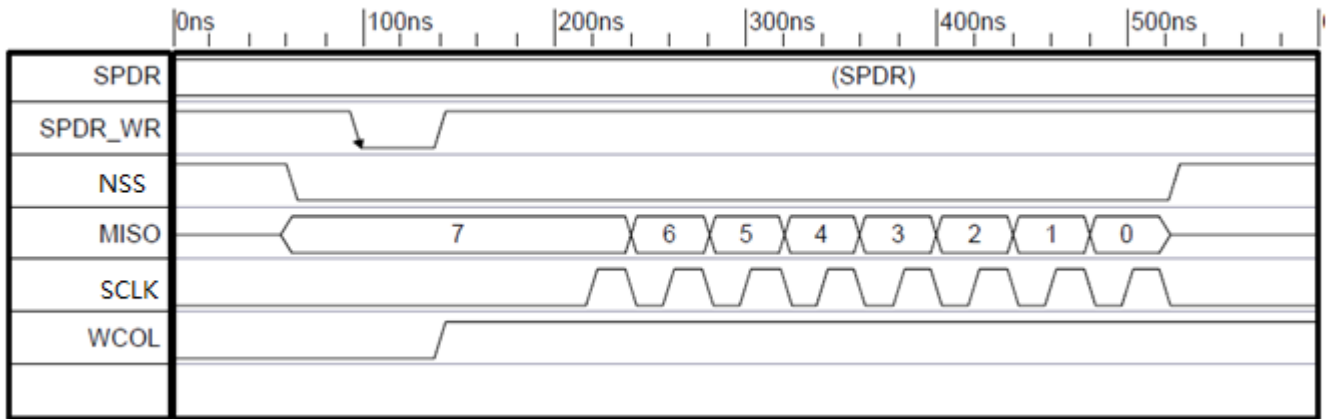


图 18-8: WCOL 错误-SPI 从模式-SPDR write when CPHA = 0 and NSS = 0

另外，从动模式下在写 SPDR 之后，主机控制的 NSS 并不是立即变为低电平。当 NSS 为低后，需要等待 SCLK 第二个边沿才开始才进入真正的数据传输状态。

在写 SPDR 到开始发送第一个数据期间，再次写入 SPDR 并不会产生写冲突。但该操作会更新准备发送的数据。如果有多次写 SPDR 的操作，发送的数据将会是最后一次写入 SPDR 的值。

在开始发送第一个数据到 SCLK 第二个边沿期间，再次写入 SPDR 并不会产生写冲突，也不会更新正在发送的数据。即忽略该次写 SPDR 的操作。

由于 SPI 只有一个发送缓冲器，建议在写 SPDR 之前判断上一次的数据是否发送完毕，确定发送完成后再写 SPDR 寄存器，以防止产生写冲突。

18.7 SPI 时钟控制逻辑

18.7.1 SPI 时钟相位与极性控制

软件可以在 SPI 控制寄存器 (SPCR) 中选择使用两个控制位的 (串行时钟 SCLK 的相位和极性) 四种组合中的任何一种。时钟极性由 CPOL 控制位指定, CPOL 控制位选择有效的高电平或低电平有效时钟, 并且对传输格式没有显著影响。时钟相位 (CPHA) 控制位选择两种基本不同的传输格式中的一种。主机 SPI 设备和通信从机设备的时钟相位和极性应相同。在某些情况下, 在传输间改变相位和极性, 以允许主机设备与具有不同要求的外设从机进行通信。SPI 系统的灵活性允许与几乎所有现有的同步串行外设直接连接。

18.7.2 SPI 传输格式

在 SPI 传输期间, 数据同时发送 (串行移出) 并接收 (串行移入)。串行时钟线与两条串行数据线移位和采样同步。从机选择线允许单独选择从机 SPI 设备; 未选择的从机设备不会干扰 SPI 总线活动。在 SPI 主机设备上, 从机选择线可以有选择性的用于指示多主机总线竞争。

18.7.3 CPHA=0 传输格式

下图显示了 CPHA 为 0 的 SPI 传输的时序图。SCLK 显示两个波形: 一个用于 CPOL 等于 0, 另一个用于 CPOL 等于 1。通过 SCLK 该图可以描述为主机设备或从机设备时序图, 主进/从出 (MISO) 和主出/从进 (MOSI) 引脚直接连接在主机和从机之间。MISO 信号是从机输出, MOSI 信号是主机输出。NSS 线是从机的从机选择输入; 主机的 NSS 引脚未显示, 但假定为无效。主机的 NSS 引脚必须为高电平。该时序图在功能上描述了如何进行传输; 不应将其用作数据表参数信息的替代品。

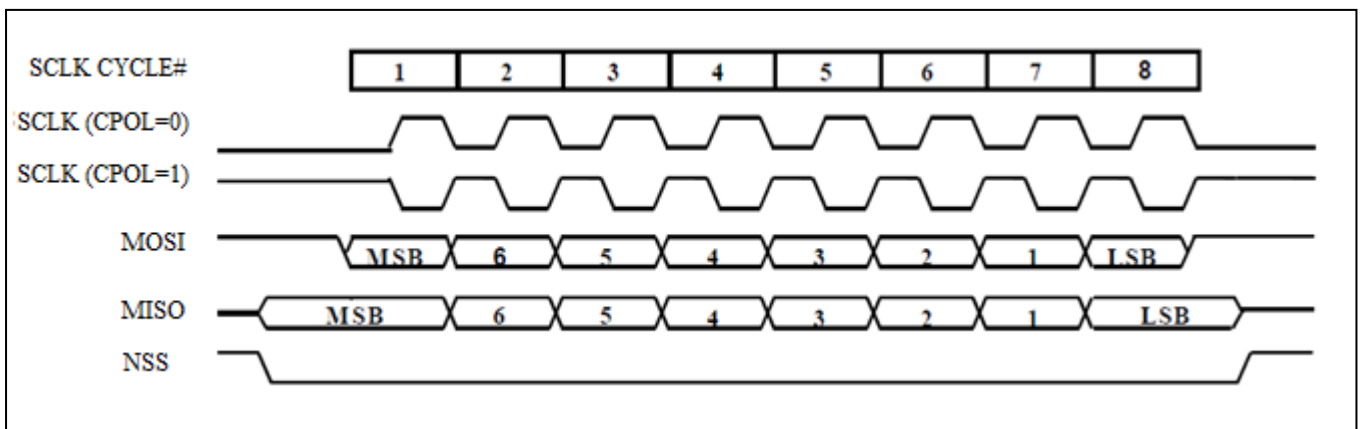


图 18-9: CPHA=0 SPI 传输格式

当 CPHA=0 时, NSS 线必须在每个连续的串行字节之间解除置 1 并重新置 1。此外, 如果当 NSS 处于低电平, 从机将数据写入 SPI 数据寄存器 (SPDR), 则会产生写冲突错误。当 CPHA = 1 时, NSS 线可能在连续传输之间保持低电平 (可以始终保持低电平)。在具有单个固定主机和驱动 MISO 数据线的单个从机的系统中, 这种格式有时是优先选择的。

18.7.4 CPHA=1 传输格式

下图是 CPHA = 1 的 SPI 传输的时序图。SCLK 显示两个波形：一个用于 CPOL=0, 另一个用于 CPOL=1。由于 SCLK、MISO 和 MOSI 引脚直接连接在主机和从机之间, 所以该图可以解释为主机或从机时序图。MISO 信号是从机输出, MOSI 信号是主机输出。NSS 线是从机的从机选择输入; 主机的 NSS 引脚未显示, 但假定为无效。主机的 NSS 引脚必须为高电平, 或必须重新配置为不影响 SPI 的通用输出。

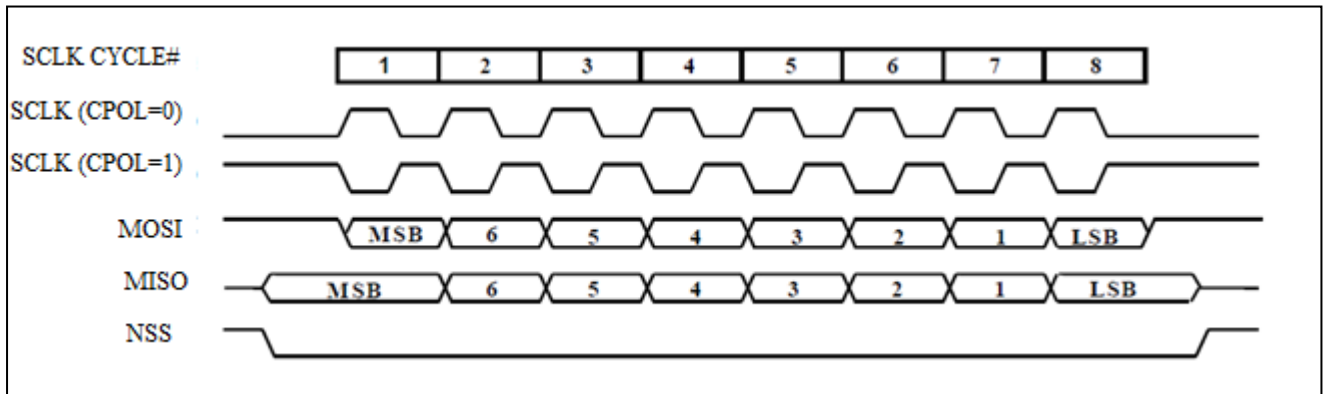


图 18-10: CPHA=1 SPI 传输格式

18.8 SPI 数据传输

18.8.1 SPI 传输启动

所有 SPI 传输都由主 SPI 设备启动和控制。作为从机设备，SPI 根据选择的 CPHA 格式，将考虑传输开始于第一个 SCLK 边沿或 NSS 的下降沿。当 CPHA = 0 时，NSS 的下降沿表示传输的开始。当 CPHA = 1 时，SCLK 上的第一个边沿表示传输的开始。无论哪种 CPHA 模式，通过使 NSS 线为高电平可以中止传输，但会使 SPI 从机逻辑和计数器复位。选择的 SCLK 速率对从机操作没有影响，因为主机的时钟正在控制传输。

当 SPI 配置为主机时，通过写入 SPDR 的软件启动传输。

18.8.2 SPI 传输结束

当 SPIF 标志置 1 时，SPI 传输在技术上完成，但是根据 SPI 系统的配置，可能还有其他任务。由于 SPI 比特率不影响结束期的时间，因此在结束期间的讨论中只考虑最快的速率。当 SPI 被配置为主机时，SPIF 在第八个 SCLK 周期循环结束时置位。当 CPHA 等于 1 时，SCLK 在第八个 SCLK 周期的最后一半处于不活动状态。

因为 SCLK 线可以与从机的 MCU 时钟异步，并且从机不能像访问 SCLK 周期那样访问主机尽可能多的信息，所以当 SPI 作为从机运行时，结束周期是不同的。例如，当 CPHA = 1 时，其中最后一个 SCLK 边沿在第八个 SCLK 周期的中间发生，从机无法知道上一个 SCLK 周期是何时结束。由于这些原因，从机认为在串行数据的最后一位被采样之后，传输完成，这对应于第八个 SCLK 周期的中间。

SPIF 标志设置在传输结束时，但是 NSS 线仍然为低电平时，从机不允许将新数据写入 SPDR。

18.9 SPI 时序图

18.9.1 主控模式传输

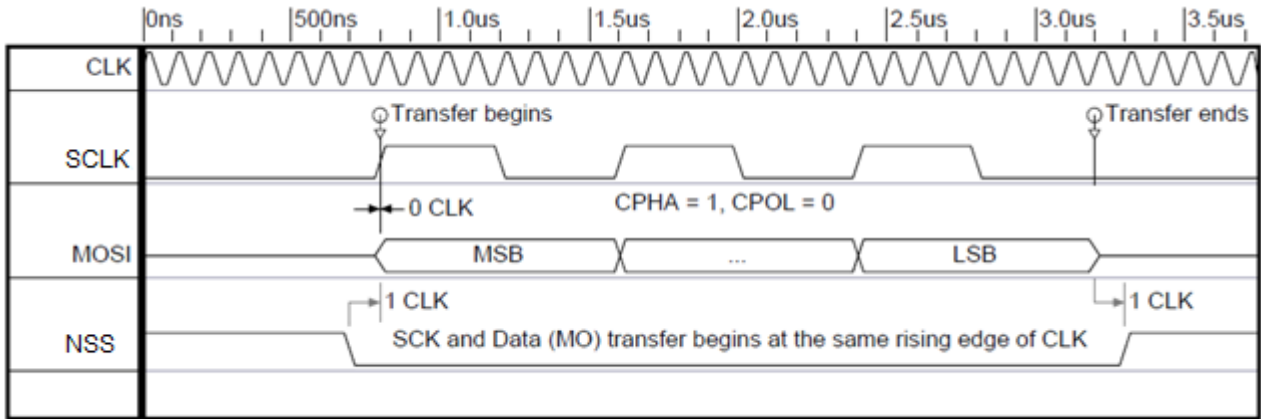


图 18-11: 主模式时序图

18.9.2 从动模式传输

在从机模式下的传输开始时，串行输出（MISO）上的数据出现在从机选择（NSS）线上的下降沿之后。

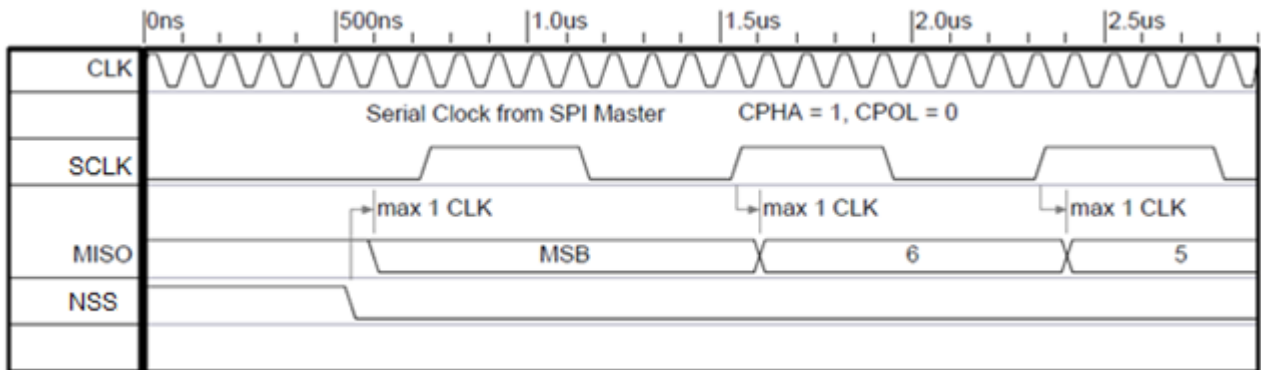


图 18-12: 从模式时序图

18.10 SPI 中断

SPI 的中断号为 22，其中断向量为 0x00B3。使能 SPI 中断必须将其使能位 SPIIE 置 1，且将总中断使能位 EA 置 1。

若 SPI 相关的中断使能均打开，SPI 总中断指示位 SPIIF=1 时，CPU 将进入中断服务程序。SPIIF 操作属性为只读，且与 SPIIE 的状态无关。

SPI 状态寄存器 SPSR 中传输完成标志 SPISIF、写冲突 WCOL 任意一个标志为 1 后，SPI 总中断指示位 SPIIF 将会置 1。仅当这 3 个标志位均为 0 时，SPIIF 自动清 1。

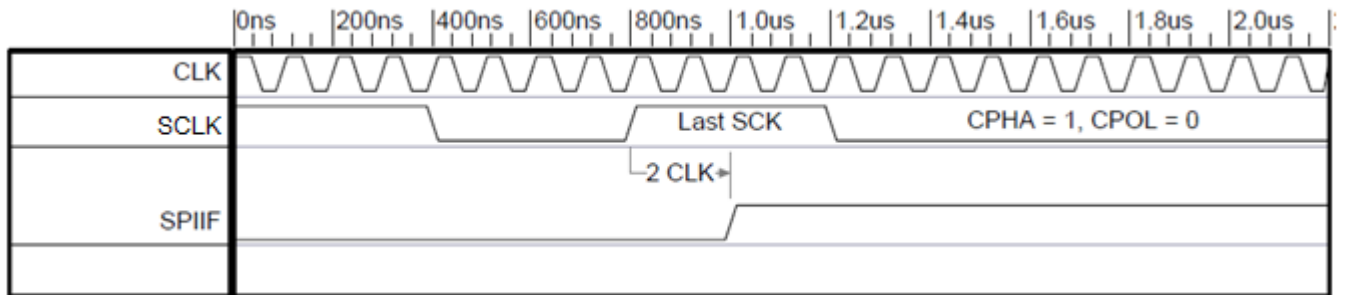


图 18-13: 中断产生

中断屏蔽寄存器 EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 SPIIE: SPI中断使能位;
 1= 允许SPI中断;
 0= 禁止SPI中断。
- Bit6 I2CIE: I²C中断使能位;
 1= 允许I²C中断;
 0= 禁止I²C中断。
- Bit5 WDTIE: WDT中断使能位;
 1= 允许WDT溢出中断;
 0= 禁止WDT溢出中断。
- Bit4 ADCIE: ADC中断使能位;
 1= 允许ADC中断;
 0= 禁止ADC中断。
- Bit3 PWMIE: PWM总中断使能位;
 1= 允许PWM所有中断;
 0= 禁止PWM所有中断。
- Bit2 --
- Bit1 ET4: Timer4中断使能位;
 1= 允许Timer4中断;
 0= 禁止Timer4中断。
- Bit0 ET3: Timer3中断使能位;
 1= 允许Timer3中断;
 0= 禁止Timer3中断。

设中断优先级控制寄存器 EIP2

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 PSPI: SPI中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit6 PI2C: I²C中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit5 PWDT: WDT中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit4 PADC: ADC中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit3 PPWM: PWM中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit2 --
- Bit1 PT4: TIMER4中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。
- Bit0 PT3: TIMER3中断优先级控制位;
 1= 设置为高级中断;
 0= 设置为低级中断。

外设中断标志位寄存器 EIF2 (0xB2)

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
复位值	0	0	0	0	0	0	0	0

- Bit7 SPIIF: SPI总中断指示位, 只读;
 1= SPI产生中断, (清除具体的中断标志位后, 此位自动清除);
 0= SPI未产生中断。
- Bit6 I2CIF: I²C总中断指示位, 只读;
 1= I²C产生中断, 需软件清零 (清除具体的中断标志位后, 此位自动清除);
 0= I²C未产生中断。
- Bit5 --
- Bit4 ADCIF: ADC中断标志位;
 1= ADC转换完成, 需软件清零;
 0= ADC转换未完成。
- Bit3 PWMIF: PWM总中断指示位, 只读;
 1= PWM产生中断, (清除具体的中断标志位后, 此位自动清除);
 0= PWM未产生中断。
- Bit2 --
- Bit1 TF4: Timer4定时器溢出中断标志位;
 1= Timer4定时器溢出, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= Timer4定时器无溢出。
- Bit0 TF3: Timer3定时器溢出中断标志位;
 1= Timer3定时器溢出, 进入中断服务程序时硬件自动清零, 也可软件清零;
 0= Timer3定时器无溢出。

19. Flash 存储器的操作

Flash 存储器对用户来说分为两块：程序区与数据区。

- 程序区空间大小为 16K*8Bit；数据区（Data Flash）为 1K*8Bit；
- 程序空间分为 32 个扇区，一个扇区包含 512 个字节；数据区则共为 2 个扇区。

存储器在正常工作状态下是可读的。另外可以通过特殊功能寄存器（SFR）对其进行间接寻址。共有 5 个

SFR 寄存器用于访问程序存储器：

- MCTRL
- MDATA
- MADRL
- MADRH
- MLOCK

操作存储器模块接口，MDATA 寄存器形成一个字节用于保存要读/写的 8 位数据，而 MADR 寄存器存放被访问的 MDATA 单元的地址。存储器允许字节读写。字节写操作写入新数据（在写入前擦除）。写入时间由片上定时器控制。写入和擦除电压是由片上电荷泵产生的，此电荷泵额定工作在器件的电压范围内，用于进行字节操作。

由于存储器类型为 Flash，擦除操作仅支持扇区擦除，不支持字节擦除。在修改某个地址的数据之前，建议将其他数据保存后，擦除当前扇区，然后进行数据写入操作。

通过存储器模块接口，可对存储器进行 R/W/E 操作，即读取/写入/擦除操作。

Flash 保护锁定寄存器 MLOCK

0xFB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MLOCK	MLOCK7	MLOCK6	MLOCK5	MLOCK4	MLOCK3	MLOCK2	MLOCK1	MLOCK0
读写	W	W	W	W	W	W	W	W
复位值	0	1	0	1	0	1	0	1

Bit7~Bit0 MLOCK<7:0>: 存储器操作使能位。

AAH: 允许存储器相关R/W/E操作

00H/FFH/55H: 不允许操作

禁止写入其他值

（该寄存器仅支持写操作，读取为00H）

Flash 存储器数据寄存器 MDATA

0xFE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MDATA	MDATA7	MDATA6	MDATA5	MDATA4	MDATA3	MDATA2	MDATA1	MDATA0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	X	X	X	X	X	X	X	X

Bit7~Bit0 MDATA<7:0>: 对程序存储器进行读取或写入的数据。

Flash 存储器地址寄存器 MADRL

0xFC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADRL	MADRL7	MADRL6	MADRL5	MADRL4	MADRL3	MADRL2	MADRL1	MADRL0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	1	1	1	1	1	1

Bit7~Bit0 MADRL<7:0>: 指定存储器读取/写入操作的地址低8位。

Flash 存储器地址寄存器 MADRH

0xFD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADRH	MADRH7	MADRH6	MADRH5	MADRH4	MADRH3	MADRH2	MADRH1	MADRH0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 MADRH<7:0>: 指定存储器读取/写入操作的地址高8位。

Flash 存储器控制寄存器 MCTRL

0xFF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MCTRL	--	--	MERR	MREG	MMODE1	MMODE0	--	MSTART
读写	R	R	R	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	1	0	0	0	0

Bit7~Bit6 --

Bit5 MERR: 操作出错标志位（写0清除）；
1= 在编程操作开始前，检测编程地址中的数据不为“FFH”（未擦除），写入操作立即终止。

0= --

Bit4 MREG: Flash区域选择位；
1= 选择数据区（低10位地址有效）；
0= 选择程序区（低14位地址有效）。

Bit3~ Bit2 MMODE<1:0>: 操作模式选择位：
11= 擦除操作模式（擦除操作的范围为：当前地址所在的整个扇区）；
10= 写操作模式；
01= 保留；
00= 读操作模式。

Bit1 未用，读为0。

Bit0 MSTART: 操作开始控制位；
1= 启动程序存储器R/W/E操作（操作完成后，可由硬件自动清零）；
0= 写：终止或不启动程序存储器R/W/E操作；
读：操作完成或操作未启动。

操作 Flash 存储器时，CPU 处于暂停状态，操作完成时，CPU 继续运行指令。

- 写入时间约为：30us（包含写入前的数据检测时间，编程时间，结束处理时间）
- 读取时间为：10* T_{sys}
- 扇区擦除时间约为：1.5ms

操作存储器指令后必须跟 6 条 NOP 指令：

```
MOV MCTRL,#09H           ;写操作开始
NOP
NOP
NOP
NOP
NOP
NOP
MOV MCTRL,#01H           ;读操作开始
NOP
NOP
NOP
NOP
NOP
NOP
```

20. 唯一 ID (UID)

20.1 概述

每颗芯片拥有不同 96 位唯一身份识别号，即唯一 ID (Unique identification)。出厂时已经设置，用户不能修改。

20.2 UID 寄存器说明

UID0

F5E0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID0	UID7	UID6	UID5	UID4	UID3	UID2	UID1	UID0
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<7:0>

UID1

F5E1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID1	UID15	UID14	UID13	UID12	UID11	UID10	UID9	UID8
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<15:8>

UID2

F5E2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID2	UID23	UID22	UID21	UID20	UID19	UID18	UID17	UID16
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<23:16>

UID3

F5E3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID3	UID31	UID30	UID29	UID28	UID27	UID26	UID25	UID24
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<31:24>

UID4

F5E4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID4	UID39	UID38	UID37	UID36	UID35	UID34	UID33	UID32
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<39:32>

UID5

F5E5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID5	UID47	UID46	UID45	UID44	UID43	UID42	UID41	UID40
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<47:40>

UID6

F5E6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID6	UID55	UID54	UID53	UID52	UID51	UID50	UID49	UID48
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<55:48>

UID7

F5E7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID7	UID63	UID62	UID61	UID60	UID59	UID58	UID57	UID56
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<63:56>

UID8

F5E8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID8	UID71	UID70	UID69	UID68	UID67	UID66	UID65	UID64
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<71:64>

UID9

F5E9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID9	UID79	UID78	UID77	UID76	UID75	UID74	UID73	UID72
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<79:72>

UID10 (0xF5EA)

F5EAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID10	UID87	UID86	UID85	UID84	UID83	UID82	UID81	UID80
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<87:80>

UID11

F5EBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID11	UID95	UID94	UID93	UID92	UID91	UID90	UID89	UID88
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 UID<95:88>

21. 休眠状态下定时唤醒

21.1 定时唤醒控制寄存器

WUTCRH 寄存器

0xBD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WUTCRH	WUTEN	--	WUTPS1	WUTPS0	WUTD11	WUTD10	WUTD9	WUTD8
读写	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7 WUTEN: 定时唤醒功能使能位;
 1= 定时唤醒功能打开;
 0= 定时唤醒功能关闭。

Bit6 未用

Bit5~Bit4 WUTPS<1:0>: 定时唤醒计数器时钟分频位。
 00= F/1
 01= F/8
 10= F/32
 11= F/256

Bit3~Bit0 WUTD<11:8>: 定时唤醒时间数据高4位。

WUTCRL 寄存器

0xBC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WUTCRL	WUTD7	WUTD6	WUTD5	WUTD4	WUTD3	WUTD2	WUTD1	WUTD0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	1	1	1	1	1	1	1	1

Bit7~Bit0 WUTD<7:0>: 定时唤醒时间数据低8位。

21.2 定时唤醒原理

内部唤醒定时器原理为：在系统进入休眠模式后，CPU 与外围所有电路停止工作，而内部的低功耗振荡器 LSI 开始工作，其振荡时钟为 125KHz($T_{LSI} \approx 8\mu s$)。为 WUT (Wake Up Timer) 计数器提供时钟。

内部唤醒定时寄存器有两个：WUTCRH 和 WUTRCL。

WUTCRH 寄存器的 Bit7 为内部定时唤醒使能位：

- WUTEN=1：打开定时唤醒功能；
- WUTEN=0：关闭定时唤醒。

{WUTCRH[3:0]与 WUTRCL[7:0]}组成 12 位定时唤醒数据寄存器，复位值为 0FFH，进入休眠模式后 WUT 计数器开始计时，当 WUT 计数器的值等于定时唤醒数据寄存器的值时，就启动系统振荡器，进入唤醒等待状态。

定时唤醒时间： $T=(WUTD[11:0]+1) \times WUTPS \times T_{LSI}$

22. 指令

汇编指令总共包括 5 类：算术运算、逻辑运算、数据传送运算、布尔操作和程序分支指令，这些指令全部都与标准 8051 兼容。

22.1 符号说明

符合	说明
Rn	工作寄存器 R0-R7
Direct	内部数据存储器 RAM 的单元地址 (00H-FFH) 或特殊功能寄存器 SFR 中的地址
@Ri	间接寻址寄存器 (@R0 或 @R1)
#data	8 位二进制常数
#data16	在指令中的 16 位二进制常数
Bit	内部数据存储器 RAM 或特殊功能寄存器 SFR 中的位地址
Addr16	16 位地址，地址范围 0-64KB 地址空间
Addr11	11 位地址，地址范围 0-2KB 地址空间
Rel	相对地址
A	累加器

22.2 指令一览表

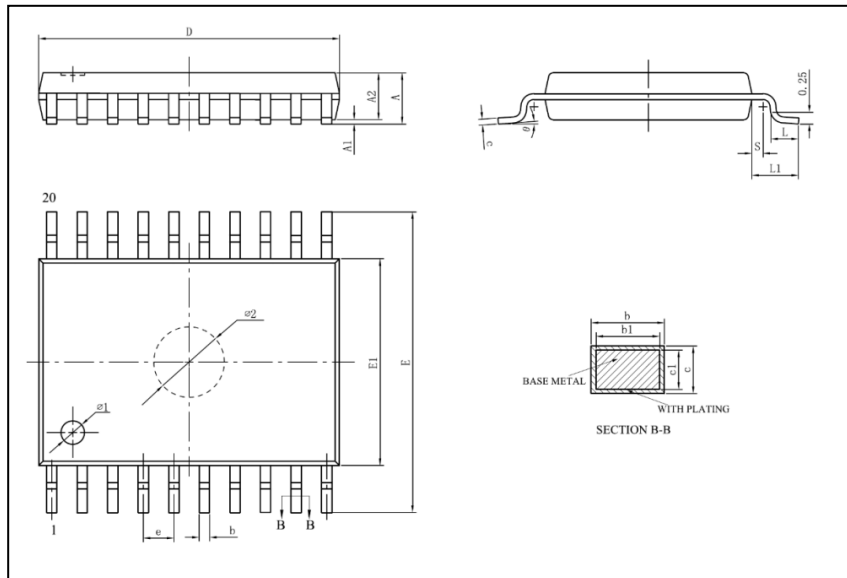
助记符	描述
运算类	
ADD A,Rn	累加器加寄存器
ADD A,direct	累加器加直接寻址单元
ADD A,@Ri	累加器加间接寻址RAM
ADD A,#data	累加器加立即数
ADDC A,Rn	累加器加寄存器和进位标志
ADDC A,direct	累加器加直接寻址单元和进位标志
ADDC A,@Ri	累加器加间接寻址RAM和进位标志
ADDC A,#data	累加器加立即数和进位标志
SUBB A,Rn	累加器减寄存器和进位标志
SUBB A,direct	累加器减直接寻址单元和进位标志
SUBB A,@Ri	累加器减间接寻址RAM和进位标志
SUBB A,#data	累加器减立即数和进位标志
INC A	累加器加1
INC Rn	寄存器加1
INC direct	直接寻址单元加1
INC @Ri	间接寻址RAM加1
INC DPTR	数据指针加1
DEC A	累加器减1
DEC Rn	寄存器减1
DEC direct	直接寻址单元减1
DEC @Ri	间接寻址RAM减1
MUL AB	累加器乘寄存器B
DIV AB	累加器除以寄存器B
DA A	十进制调整
逻辑运算类	
ANL A,Rn	累加器与寄存器
ANL A,direct	累加器与直接寻址单元
ANL A,@Ri	累加器与间接寻址RAM
ANL A,#data	累加器与立即数
ANL direct,A	直接寻址单元与累加器
ANL direct,#data	直接寻址单元与立即数
ORL A,Rn	累加器或寄存器
ORL A,direct	累加器或直接寻址单元
ORL A,@Ri	累加器或间接寻址RAM
ORL A,#data	累加器或立即数
ORL direct,A	直接寻址单元或累加器
ORL direct,#data	直接寻址单元或立即数
XRL A,Rn	累加器异或寄存器
XRL A,direct	累加器异或直接寻址单元
XRL A,@Ri	累加器异或间接寻址RAM
XRL A,#data	累加器异或立即数
XRL direct,A	直接寻址单元异或累加器
XRL direct,#data	直接寻址单元异或立即数
CLR A	累加器清0
CPL A	累加器取反

助记符	描述
RL A	累加器左循环移位
RLC A	累加器连进位标志左循环移位
RR A	累加器右循环移位
RRC A	累加器连进位标志右循环移位
SWAP A	累加器高4位与低4位交换
数据传输类	
MOV A,Rn	寄存器传送到累加器
MOV A,direct	直接寻址单元传送到累加器
MOV A,@Ri	间接寻址RAM送累加器
MOV A,#data	立即数送累加器
MOV Rn,A	累加器送寄存器
MOV Rn,direct	直接寻址单元送寄存器
MOV Rn,#data	立即数送寄存器
MOV direct,A	累加器送直接寻址单元
MOV direct,Rn	寄存器送直接寻址单元
MOV direct1,direct2	直接地址单元传送到直接寻址单元
MOV direct,@Ri	间接寻址RAM送直接寻址单元
MOV direct,#data	立即数送直接寻址单元
MOV @Ri,A	累加器送间接寻址RAM
MOV @Ri,direct	直接寻址单元送间接寻址RAM
MOV @Ri,#data	立即数送间接寻址RAM
MOV DPTR,#data16	16位立即数送数据指针
MOVC A,@A+DPTR	查表数据送累加器 (DPTR为基址)
MOVC A,@A+PC	查表数据送累加器 (PC为基址)
MOVX A,@Ri	外部RAM单元送累加器 (8位地址)
MOVX A,@DPTR	外部RAM单元送累加器 (16位地址)
MOVX @Ri,A	累加器送外部RAM单元 (8位地址)
MOVX @DPTR,A	累加器送外部RAM单元 (16位地址)
PUSH direct	直接寻址单元压入栈顶
POP direct	栈顶弹出直接寻址单元
XCH A,Rn	累加器与寄存器交换
XCH A,direct	累加器与直接寻址单元RAM交换
XCH A,@Ri	累加器与间接寻址单元RAM交换
XCHD A,@Ri	累加器与间接寻址单元RAM交换低4位
布尔运算类	
CLR C	C清零
CLR bit	直接寻址位清零
SETB C	C置位
SETB bit	直接寻址位置位
CPL C	C取反
CPL bit	直接寻址位取反
ANL C,bit	C逻辑与直接寻址位
ANL C,/bit	C逻辑与直接寻址位的反
ORL C,bit	C逻辑或直接寻址位
ORL C,/bit	C逻辑或直接寻址位的反
MOV C,bit	直接寻址位送C
MOV bit,C	C送直接寻址位
程序跳转类	

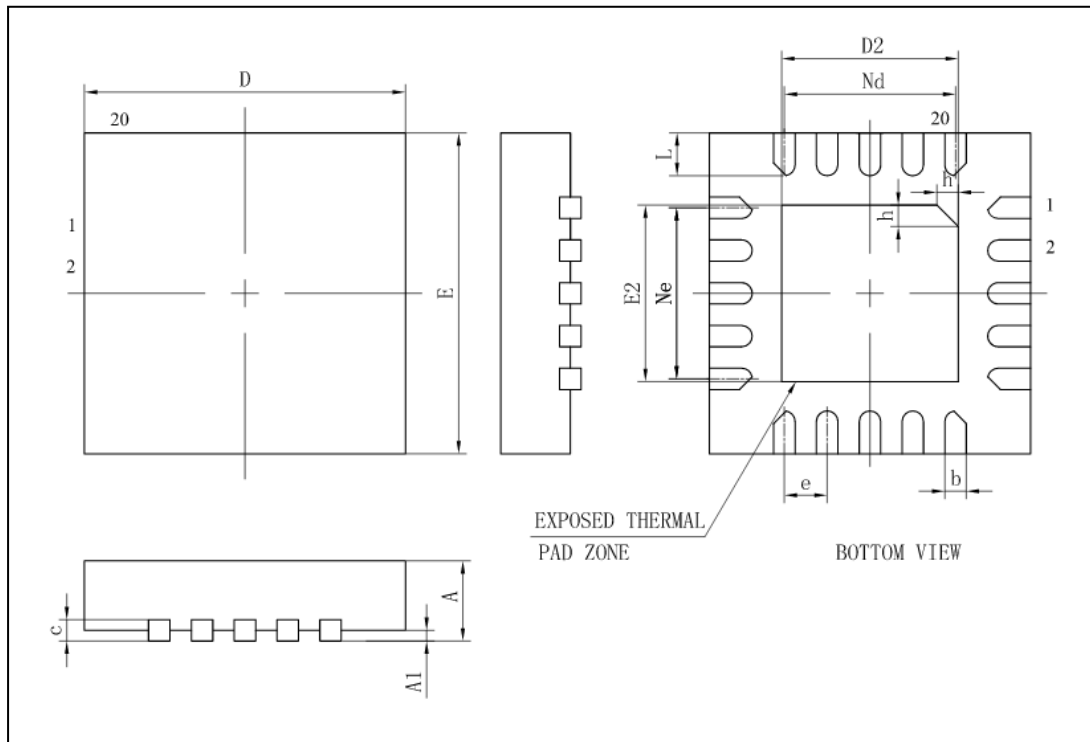
助记符	描述
ACALL addr11	2K地址范围内绝对调用
LCALL addr16	64K地址范围内长调用
RET	子程序返回
RETI	中断返回
AJMP addr11	2K地址范围内绝对转移
LJMP addr16	64K地址范围内长转移
SJMP rel	相对短转移
JMP @A+DPTR	相对长转移
JZ rel	累加器为0转移
JNZ rel	累加器不为0转移
JC rel	C为1转移
JNC rel	C为0转移
JB bit,rel	直接寻址位为1转移
JNB bit,rel	直接寻址位为0转移
JBC bit,rel	直接寻址位为1转移, 并清该位
CJNE A,direct,rel	累加器与直接寻址单元不等转移
CJNE A,#data,rel	累加器与立即数不等转移
CJNE Rn,#data,rel	寄存器与立即数不等转移
CJNE @Ri,#data,rel	间接寻址单元RAM与立即数不等转移
DJNZ Rn,rel	寄存器减1不为0转移
DJNZ direct,rel	直接寻址单元减1不为0转移
NOP	空指令
读取—修改—写入指令 (Read-Modify-Write)	
ANL	逻辑 (ANL direct, A 与 ANL direct, #data)
ORL	逻辑或(ORL direct, A 与 ORL direct, #data)
XRL	逻辑异或 (XRL direct, A 与 XRL direct, #data)
JBC	直接寻址位为1转移, 并清该位 (JBC bit, rel)
CPL	取反 (CPL bit)
INC	加1 (INC direct)
DEC	减1. (DEC direct)
DJNZ	减1不为0转移(DJNZ direct, rel)
MOV bit,C	C送直接寻址位
CLR bit	直接寻址位清零
SETB bit	直接寻址位置位

23. 封装

23.1 TSSOP20



Symbol	Millimeter		
	Min	Nom	Max
A	-	-	1.20
A1	0.05	-	0.15
A2	0.80	1.00	1.05
b	0.19	-	0.30
b1	0.19	0.22	0.25
c	0.09	-	0.20
c1	0.09	-	0.16
D	6.40	6.50	6.60
E1	4.30	4.40	4.50
E	6.20	6.40	6.60
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00BSC		
S	0.20	-	-
$\phi 1$	$\phi 0.8 \times 0.05 \sim 0.10DP$		
$\phi 2$	$\phi 1.50 \times 0.05 \sim 0.15DP$		
θ	0	-	8°

23.2 QFN20


Symbol	Millimeter		
	Min	Nom	Max
A	0.70	0.75	0.80
A1	-	0.02	0.05
b	0.15	0.20	0.25
c	0.18	0.20	0.25
D	2.90	3.00	3.10
D2	1.55	1.65	1.75
e	0.4BSC		
Ne	1.6BSC		
Nd	1.6BSC		
E	2.9	3.00	3.10
E2	1.55	1.65	1.75
L	0.35	0.40	0.45
h	0.20	0.25	0.30

24. 版本修订说明

版本号	时间	修改内容
V1.00	2020年3月15日	初始版本
V1.01	2021年5月17日	修改 IIC MTP 描述
V1.02	2021年8月25日	修改 MLOCK 复位值
V1.03	2021年10月12日	修改功能说明