



JSC68F016

8051-内核微控制器

用户数据手册

*无锡芯加集成电路有限公司已申请了专利，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。

*未经本公司许可，严禁以任何形式复制或转载本资料的部分或全部内容

*请在本资料所记载的极限范围内使用本产品，因使用不当造成的损失，我司不承担其责任。

*尽管本公司一向致力于提高产品质量与可靠性，但是半导体产品本身有一定的概率发生故障或错误工作，为防止因此类事故而造成的人身伤害或财产损失，请在使用过程中充分留心安全设计。

*本资料内容，随产品的改进，会进行相应更新，恕不另行通知。使用本资料前请咨询我司销售人员或者参考官方网站 www.champic.com.cn，以保证本资料内容为最新版本。

目录

1	产品概述	9
2	系统结构框图	11
3	引脚分配和引脚说明	12
3.1	封装引脚图	12
3.2	引脚说明	13
4	中央处理器	17
4.1	存储器结构	17
4.2	程序存储器	17
4.3	数据存储器	18
4.4	外部 XRAM	19
4.5	累加器 ACC	19
4.6	B 寄存器	19
4.7	堆栈指针寄存器 (SP)	19
4.8	特殊功能寄存器 (SFR)	19
4.9	特殊功能寄存器 (XSFR)	21
4.10	时效访问保护 (TA)	25
4.11	系统控制寄存器	26
5	系统时钟	30
5.1	系统时钟概述	30
5.2	系统时钟源	31
5.2.1	内部振荡器	31
5.3	系统时钟切换	31
5.4	系统时钟分频	32
5.5	系统时钟输出	32
5.6	系统时钟配置寄存器	33
6	复位	36
6.1	POR 上电复位	36
6.2	LVR 复位	37
6.3	BOD 欠压复位	37
6.4	HFR 硬件故障复位	37
6.5	RST 外部复位	37
6.6	看门狗复位	37
6.7	软件复位	38
6.8	复位寄存器	40
7	电源管理	42
7.1	空闲模式	42
7.2	掉电模式	42
7.3	唤醒等待状态	43
7.4	电源管理寄存器	45
8	中断	46
8.1	概述	46
8.2	中断使能	47

8.3	中断优先级	47
8.4	外部中断	49
8.4.1	INT0/INT1 中断	49
8.4.2	GPIO 中断	49
8.5	中断标志位的清除	50
8.5.1	硬件自动清除标志位	50
8.5.2	软件清除的标志位	50
8.5.3	读写操作清除的标志位	50
8.6	中断寄存器	51
9	I/O 端口	63
9.1	GPIO 普通功能	63
9.1.1	数据寄存器	64
9.1.2	输出控制寄存器	65
9.1.3	上拉使能寄存器	66
9.1.4	下拉使能寄存器	67
9.1.5	驱动电流控制寄存器	68
9.1.6	输出速度控制寄存器	69
9.1.7	数据输入选择寄存器	70
9.1.8	GPIO 方向寄存器	71
9.1.9	施密特输入使能寄存器	72
9.1.10	强上拉使能寄存器	73
9.2	复用功能	74
9.2.1	端口复用功能配置寄存器	74
9.2.2	端口输入功能分配寄存器	77
9.2.3	复用功能应用注意	78
9.3	端口外部中断控制	79
9.3.1	端口中断概述	79
9.3.2	端口中断控制寄存器	79
9.3.3	PxEXTF 管脚中断标志位寄存器	80
10	看门狗	83
10.1	超时复位定时器	83
10.2	通用定时器	85
10.3	看门狗定时器寄存器	86
11	WKT 自唤醒定时器	87
12	定时器 0/1	89
12.1	模式 0（13 位定时器）	90
12.2	模式 1（16 位定时器）	90
12.3	模式 2（8 位自动重载定时器）	91
12.4	模式 3（两组独立 8 位定时器）	92
12.5	定时器 0/1 寄存器	93
13	定时器 2	98
13.1	16 位自动重载功能模式。	99
13.2	比较功能模式：	100
13.3	输入捕获功能模式：	100

13.4 定时器 2 寄存器	101
14 定时器 3/4	109
14.1 概述	109
14.2 定时器 3	109
14.3 定时器 4	110
14.4 32 位定时器使用	110
14.5 定时器 3/4 寄存器	111
15 增强型 PWM 模块	115
15.1 概述	115
15.2 特性	115
15.3 功能描述	115
15.4 端口配置	117
15.5 PWM 配置	117
15.6 PWM 类型	118
15.6.1 边沿对齐模式	118
15.6.2 中心对齐模式	118
15.7 工作模式	119
15.7.1 独立输出模式	119
15.7.2 互补模式与死区延时	119
15.7.3 同步模式	120
15.8 群组模式应用	120
15.9 输出掩码控制	121
15.10 故障刹车	121
15.11 极性控制	122
15.12 PWM 中断	122
15.13 PWM 寄存器	123
16 UART0,1 模块	134
16.1 概述	134
16.2 UARTn 端口配置	134
16.3 UARTn 波特率	134
16.4 UARTn 模式 (n=0,1)	138
16.4.1 模式 0 同步 8 位收发器	138
16.4.2 UART 模式 1: 异步 8 位收发器	139
16.4.3 UART 模式 2、3: 异步 9 位收发器	140
16.5 UART 寄存器	141
17 I ² C 模块	146
17.1 I ² C 概述	146
17.2 端口配置	147
17.3 I ² C 总线传输方式	147
17.3.1 开始和停止条件	147
17.3.2 地址数据格式	148
17.3.3 应答 ACK	148
17.3.4 I ² C 时钟时序	149
17.4 I ² C 主控模式	150

17.4.1 I ² C 主控模式时钟速率.....	150
17.4.2 I ² C 主控模式控制与状态.....	150
17.5 I ² C 从动模式.....	152
17.6 I ² C 中断.....	154
17.7 仲裁与同步模式.....	154
17.8 I ² C 状态码及应用程序示例.....	155
17.8.1 状态码.....	155
17.8.2 程序示例.....	156
17.9 I ² C 寄存器.....	159
18 SPI 模块.....	161
18.1 SPI 简介.....	161
18.2 SPI 端口配置.....	162
18.3 SPI 硬件描述.....	162
18.4 SPI 时钟分频.....	163
18.5 SPI 主控模式.....	163
18.5.1 写冲突错误.....	164
18.6 SPI 从动模式.....	164
18.6.1 写冲突错误.....	165
18.7 SPI 时钟控制逻辑.....	165
18.7.1 SPI 时钟相位与极性控制.....	165
18.7.2 SPI 传输格式.....	165
18.7.3 CPHA=0 传输格式.....	165
18.7.4 CPHA=1 传输格式.....	166
18.8 SPI 数据传输.....	166
18.8.1 SPI 传输启动.....	166
18.8.2 SPI 传输结束.....	167
18.9 SPI 时序图.....	167
18.9.1 主控模式传输.....	167
18.9.2 从动模式传输.....	168
18.10 SPI 中断.....	168
18.11 SPI 寄存器.....	170
19 MDU 模块.....	172
19.1 乘法（16 位×16 位）.....	172
19.2 除法（32 位/16 位和 16 位/16 位）.....	172
19.3 移位和规范化.....	172
19.4 错误标志（MDEF）.....	173
19.5 溢出标志（MDOV）.....	173
19.6 MDU 寄存器.....	174
20 加减移位模块.....	175
20.1 概述.....	175
20.2 加减移位寄存器.....	175
21 模数转换(ADC).....	178
21.1 ADC 概述.....	178
21.2 ADC 配置.....	178

21.2.1	端口配置	179
21.2.2	通道选择	179
21.2.3	ADC 参考电压	179
21.2.4	转换时钟	179
21.2.5	ADC 中断	180
21.2.6	结果存储	180
21.3	ADC 硬件触发启动	180
21.3.1	外部端口边沿触发 ADC	180
21.3.2	PWM 触发 ADC	180
21.3.3	硬件触发启动前的延时	180
21.4	ADC 结果比较器	181
21.5	ADC 偏移量校准	181
21.6	ADC 工作原理	182
21.6.1	启动转换	182
21.6.2	完成转换	182
21.6.3	终止转换	182
21.6.4	A/D 转换步骤	182
21.6.5	转换过程中进入休眠	183
21.7	基于 OP 的 ADC 应用	183
21.8	ADC 寄存器	184
22	OPA 模块	191
22.1	运放特性	191
22.2	运放结构	191
22.3	运放失调电压修调	191
22.4	运放使用步骤	192
22.5	运放寄存器	193
23	ACMP 比较器模块	195
23.1	概述	195
23.2	比较器特性	195
23.3	比较器框图	196
23.4	通用比较器功能	197
23.5	欠压监测功能	198
23.6	比较器寄存器	200
24	FLASH 操作 (IAP)	204
24.1	IAP 命令	205
24.2	IAP 用户指南	205
24.3	使用 FLASH 存储器作为数据存储	206
24.4	唯一 ID 96 位序列号	206
24.5	在线系统编程 (ISP)	206
24.6	IAP 指令及寄存器	207
25	在线编程及 OCDS 调试	210
25.1	在线串行编程 (ICP)	210
25.2	OCDS 功能描述	210
25.2	OCDS 限制条件	211

25.3 OCD 寄存器	211
26 系统配置 CONFIG.....	212
26.1 Config 信息	212
27 电气参数	216
27.1 绝对最大额定值	216
27.2 直流电气特性	216
27.3 交流电气特性	218
27.3.1 上电与掉电操作	218
27.3.2 外部振荡器	218
27.3.3 内部振荡器	218
27.4 GPIO 电气特性	219
27.5 低压复位电气参数	219
27.6 BOD 电气参数	220
27.7 比较器电气参数	220
27.8 ADC 电压特性	221
27.9 BANDGAP 电气特性	221
27.10 FLASH 电气特性	221
27.11 OPA 电气特性	222
27.12 EFT 电气特性	223
27.13 ESD 电气特性	223
27.14 Latch-UP 电气特性	223
28 封装信息	224
28.1 SOP16	224
28.2 SOP20	225
29 版本说明	226
图 2-1 系统结构框图	11
图 3-1 SOP-20 封装管脚配置	12
图 3-2 SOP-16 封装管脚配置	12
图 5-1 系统时钟框图	30
图 6-1 系统复位框图	36
图 10-1 看门狗复位定时器功能框图	83
图 10-2 看门狗通用定时器功能框图	85
图 11-1 自唤醒定时器功能框图	87
图 12-1 模式 0 框图	90
图 12-2 模式 1 框图	90
图 12-3 模式 2 框图	91
图 12-4 模式 3 框图	92
图 13-1 定时器 2 框图	98
图 13-2 定时器 2 自动重载和捕获功能模块	99
图 13-3 定时器 2 比较功能	100
图 14-1 定时器 3 功能框图	109
图 15-1 PWM 功能框图	116
图 15-2 边沿对齐模式	118

图 15-3 中心对齐模式.....	119
图 15-4 死区.....	120
图 15-5 刹车功能.....	121
图 15-6 PWM 中断方式.....	122
图 16-1 串口模式 0 时序图.....	138
图 16-2 串口模式 1 时序图.....	139
图 16-3 串口模式 2、3 时序图.....	140
图 17-1 I2C 连接框图.....	146
图 17-2 I2C 传输数据框图.....	147
图 17-3 I ² C 起始和停止条件	148
图 17-4 I ² C 传输格式	148
图 17-5 I ² C 应答	149
图 17-6 I ² C 时钟时序	149
图 18-1 SPI 框图.....	162
图 18-2 自动从机选择线使用.....	163
图 18-3 SPI 主模式下的写冲突错误.....	164
图 18-4 SPI 从模式下的写冲突错误.....	165
图 18-5 CPHA=0 SPI 传输格式.....	166
图 18-6 CPHA=1 SPI 传输格式.....	166
图 18-7 主模式时序图.....	167
图 18-8 从模式时序图.....	168
图 18-9 SPIF 中断	168
图 18-10 MODF 中断	169
图 21-1 ADC 框图.....	178
图 21-2 硬件触发启动延时	181
图 21-3 ADC 结果比较功能.....	181
图 22-1 OP 功能框图.....	191
图 23-1 比较器结构图.....	196
图 23-2 比较器迟滞电压.....	196
图 25-1 在线电路编程 ICP	210
表 3-1 管脚说明.....	13
表 3-2 数字功能.....	16
表 4-1 SFR 特殊功能寄存器	20
表 4-2 特殊功能寄存器 (XSFR)	21
表 6-1 复位寄存器相关复位后的值.....	39
表 7-1 退出低功耗方式一 唤醒.....	44
表 7-2 退出低功耗方式二 复位.....	44
表 8-1 中断向量表.....	46
表 8-2 中断优先级设置表.....	48
表 8-3 中断源特性描述.....	49
表 9-1 数字配置功能表.....	75
表 9-2 端口对应的模拟功能.....	76
表 9-3 输入功能端口分配寄存器表.....	77

表 10-1 WDT 溢出时间	84
表 12-1 定时器 0/1 模式	89
表 16-1 串口模式描述	138
表 17-1 I ² C 状态码	155
表 23-1 电阻分压系数表	197
表 23-2 BOD 分压选择表	199
表 24-1 IAP 模式和命令代码	205
表 26-1 P20 引脚功能配置表	215

1 产品概述

◆ 兼容 MCS-51 的 1T 指令系统

- 最高主频 48MHz
- 机器周期最快支持 1TSYS @ $F_{SYS} = 24\text{MHz}$
- 机器周期最快支持 2TSYS @ $F_{SYS} = 48\text{MHz}$

◆ 内存

- 程序 FLASH: APROM+LDROM 16K x 8Bit
- 通用 RAM: 256 x 8Bit
- 通用 XRAM: 1K x 8Bit
- 可配置 4K/3K/2K/1K/0K 字节 LDROM 引导代码区
- 支持 FLASH 分区保护功能

◆ 4 种时钟模式

- HIRC-内部 RC 振荡: 48MHz
- LIRC-内部低速振荡: 16KHz
- XT32 外部低速振荡: 32.768KHz
- ECLK-外部时钟输入: 16MHz

◆ 低压复位(LVR)

- 1.8V/2.0V/2.5V/3.5V

◆ 欠压检测及欠压复位(BOD)

- 1.7V/1.85V/1.95V/2.05V/2.15V/2.3V/2.45V/2.6V/2.8V/3V/3.25V/3.55V/3.9V/4.3V

◆ 乘除法器 (MDU)

- 提供 16 位 x 16 位的乘法运算
- 提供 32 位/16 位, 16 位/16 位除法运算

◆ 加减移位模块

- 提供直接对 16 位数据移位和加 1 功能
- 提供 4 组 16 位数据移位寄存器

◆ 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
- 均支持强上拉、弱上拉、弱下拉
- 均支持 SMT 和普通反相器输入
- 均支持 输出高速斜率和普通斜率
- 均支持 强驱动输出 普通驱动输出
- 均支持边沿 (上升沿/下降沿/双沿) 中断
- 均支持唤醒功能, 输入有 180ns 滤波

◆ 中断源

- 支持所有的外部管脚中断
- 6 个定时器中断

- 4 个中断优先级设置

- 其他外设中断

◆ 定时器

- WDT 定时器 (看门狗定时器)
- 两个 16 位普通定时器 Timer0/1
- 一个 16 位定时 Timer2, 具备自动重载、比较、捕获功能
- 最多可达两个 16 位自动重载定时器 Timer3、Timer4
- 一个可由 Timer3, Timer4 级联组成的 32 位定时器
- WKT(自唤醒定时器)
- BRT(独立串口波特率时钟发生器)

◆ 通信模块

- 1×SPI (通信速率最高可达 6Mb/s)
- 1×I2C (通信速率最高可达 400Kb/s)
- 2×UART (波特率最高可达 1Mb/s)

◆ 独立 PWM

- 一路由 Timer3 控制的独立 PWM 输出

◆ 增强型 EPWM

- 8 通道增强型 PWM
- 8 个相互独立占空比寄存器
- 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 (18 I/Os) 均支持 AD 通道
- 参考电压可选 (2.0V/3.0V/4.0V/VDD/外部基准)
- 提供检测内部 1.2V 基准电压通道
- 提供检测温度传感器的内部通道
- 支持硬件触发启动转换功能
- 支持一组结果数字比较功能

◆ 模拟比较器 (ACMP)

- 正端和负端都有 16 个输入通道选择
- 正端和负端支持内部 1.2V 电压基准输入
- 负端支持对 VDD 或者 VREF 进行 32 等分分压后输入
- 支持比较器触发 PWM 刹车功能
- 支持比较器结果通过 GPIO 输出

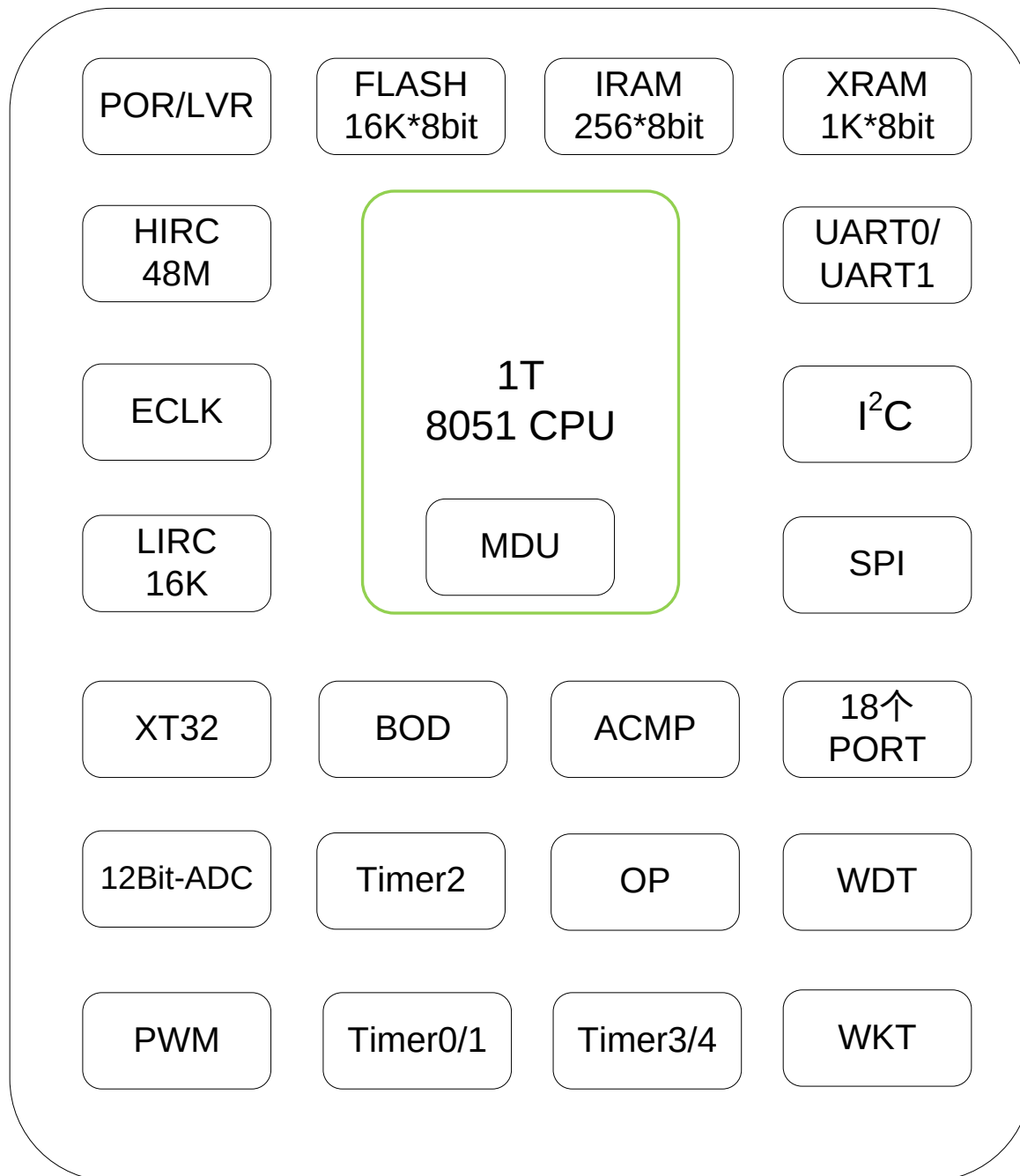
- 支持上升沿、上升沿下降沿触发中断
- 支持输出极性选择
- 支持 0、20、30、60mv 迟滞电压选择
- ◆ **运算放大器 (OPA)**
 - 正端支持三路外部输入、一路 0.25VDD 输入
 - 负端支持两路外部输入或配置成 PGA 模式，当配置成 PGA 功能时，可使用内部放大网络对输入信号进行放大
 - 可设置为 PGA 功能，支持 1-128 倍放大
 - 支持放大器/比较器两种模式
 - 运放的输出可通过配置由 P05 口输出
 - 运放的输出可通过配置由内部送至 ADC 或比较器
 - 支持失调电压软件修调
- ◆ **P20 引脚功能**
 - 可以通过配置字使 P20 具有以下功能：
 - 普通 GPIO 口
 - 外部复位引脚
 - 外部 ADC 基准电压输入引脚
 - 外部低速晶振引脚
- ◆ **外部低速晶振**
 - 通过配置字可配置 P20/P30 引脚为外部低速晶振输入引脚，为系统提供低速 32.768k 时钟
- ◆ **低功耗模式**
 - 空闲模式 (IDLE)
 - 掉电模式 (STOP, 功耗约 3uA)，VDD=3.0V
- ◆ **工作电压范围**
 - $2.5V < VDD \leq 5.5V$
(2T 最高主频 48MHz 运行，全功能)
 - $1.8V < VDD \leq 2.5V$
(2T 最高主频 24MHz 运行，IAP 操作禁止)
- ◆ **工作温度范围**
 - $-40^{\circ}\text{C} \sim 105^{\circ}\text{C}$
- ◆ **支持 96 位唯一 ID 号 (UID)**
 - 每颗芯片都有独立的 ID 号
- ◆ **IAP**
 - 内建 IAP 编程功能：
 - 可更新任意配置字
 - 可对任意地址进行字节编程
 - 可进行扇区擦除
 - 可进行页擦除
 - 可进行全部擦除
- ◆ **开发与调试**
 - 提供一线片上调试功能 (OCD)
 - 提供电在线路编程 (ICP 编程)
 - 提供系统编程 (ISP 编程)
 - 提供两线下载功能

型号说明：

型号	FLASH	RAM	XRAM	I/O	ADC	ACMP	OPA	封装
JSC68F016SO20(B)	16KB	256B	1KB	18	12BitX16	1 路	1 路	SOP20
JSC68F016SO16(B)	16KB	256B	1KB	14	12BitX13	1 路	1 路	SOP16

2 系统结构框图

图 2-1 系统结构框图



3 引脚分配和引脚说明

3.1 封装引脚图

图 3-1 SOP-20 封装管脚配置

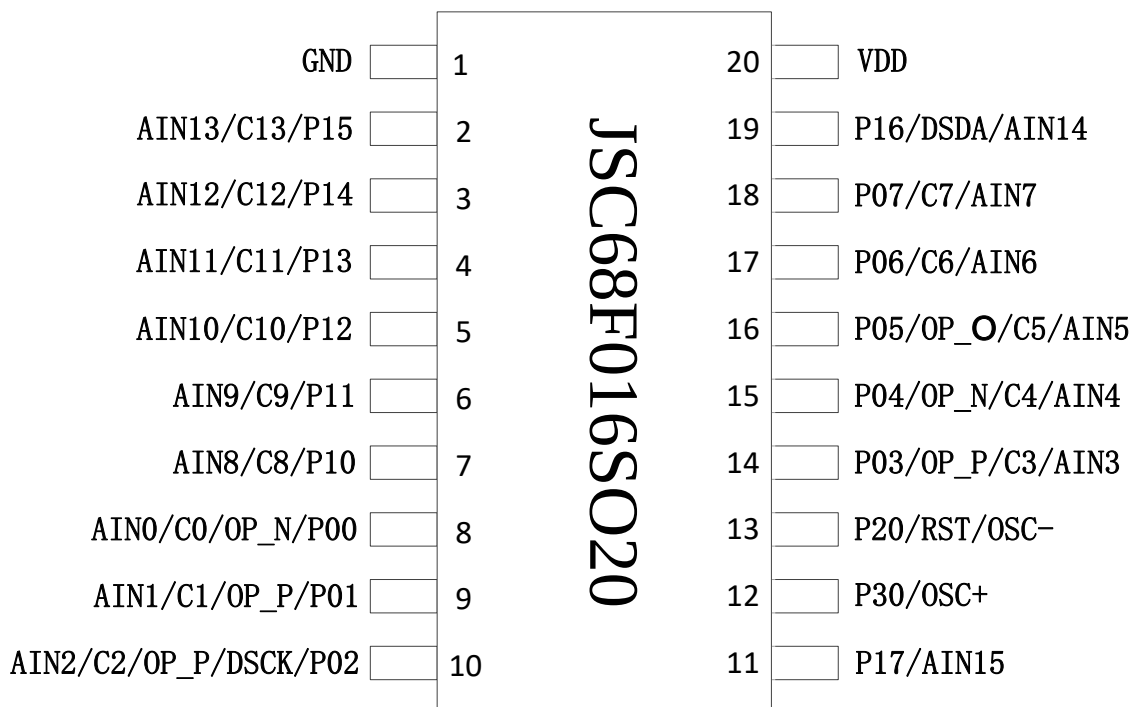
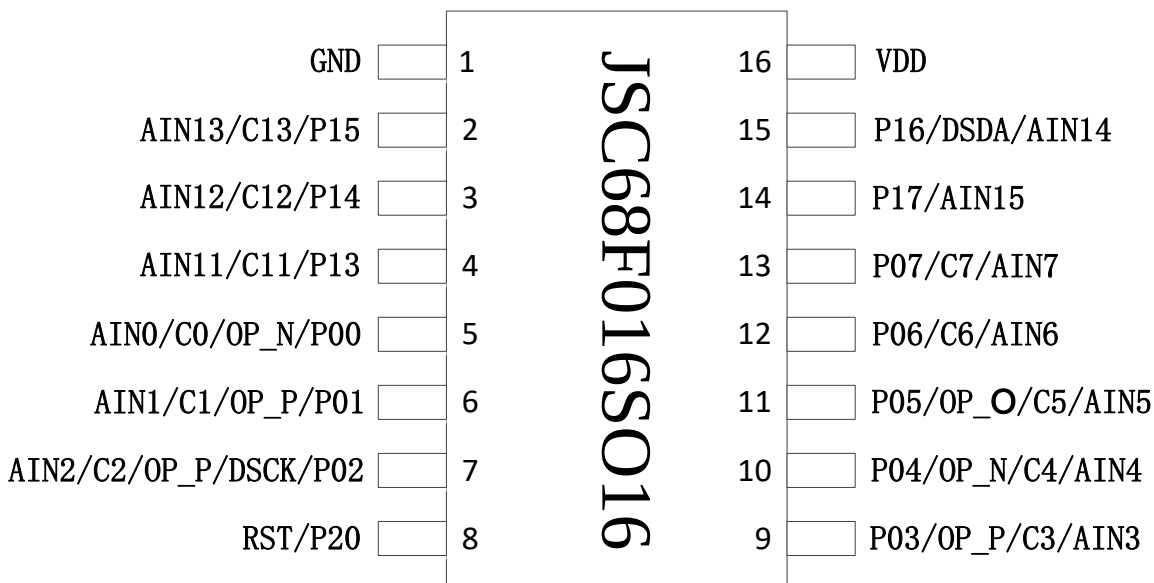


图 3-2 SOP-16 封装管脚配置



3.2 引脚说明

表 3-1 管脚说明

管脚		引脚名称	功能描述
JSC68F016SO20	JSC68F016SO16		
20	16	VDD	电源电压输入
1	1	GND	接地
8	5	P00	通用 GPIO: 可配置为输入输出, 上、下拉等功能
			OP_N: 放大器负端输入
			C0: 比较器通道 0 输入
			AIN0: ADC 通道 0 输入
			外设功能: 全映射, 见表 3-2 数字功能
			输入功能 (INT0/1、T0/1、T0G、T1G、CA0/1/2、ADET、FB)
9	6	P01	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			OP_P: 放大器正端输入
			C1: 比较器通道 1 输入
			AIN1: ADC 通道 1 输入
			外设功能: 全映射, 见表 3-2 数字功能
			输入功能 (INT0/1、T0/1、T0G、T1G、CA0/1/2、ADET、FB)
10	7	P02	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			DSCK: 编程、调试时钟输入输出
			OP_P: 放大器正端输入
			C2: 比较器通道 2 输入
			AIN2: ADC 通道 2 输入
			外设功能: 全映射, 见表 3-2 数字功能
14	9	P03	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			OP_P: 放大器正端输入
			C3: 比较器通道 3 输入
			AIN3: ADC 通道 3 输入
			外设功能: 全映射, 见表 3-2 数字功能
			输入功能 (INT0/1、T0/1、T0G、T1G、CA0/1/2、ADET、FB)
15	10	P04	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			OP_N: 放大器负端输入
			C4: 比较器通道 4 输入
			AIN4: ADC 通道 4 输入
			外设功能: 全映射, 见表 3-2 数字功能
			输入功能 (INT0/1、T0/1、T0G、T1G、CA0/1/2、ADET、FB)

16	11	P05	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			OP_O: 放大器输出端口
			C5: 比较器通道 5 输入
			AIN5: ADC 通道 5 输入
			外设功能: 全映射, 见表 3-2 数字功能
			输入功能 (INT0/1、T0/1、T0G、T1G、CA0/1/2、ADET、FB)
17	12	P06	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			C6: 比较器通道 6 输入
			AIN6: ADC 通道 6 输入
			外设功能: 全映射, 见表 3-2 数字功能
18	13	P07	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			C7: 比较器通道 7 输入
			AIN7: ADC 通道 7 输入
			外设功能: 全映射, 见表 3-2 数字功能
7	-	P10	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			C8: 比较器通道 8 输入
			AIN8: ADC 通道 8 输入
			外设功能: 全映射, 见表 3-2 数字功能
6	-	P11	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			C9: 比较器通道 9 输入
			AIN9: ADC 通道 9 输入
			外设功能: 全映射, 见表 3-2 数字功能
5	-	P12	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			C10: 比较器通道 10 输入
			AIN10: ADC 通道 10 输入
			外设功能: 全映射, 见表 3-2 数字功能
4	4	P13	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			C11: 比较器通道 11 输入
			AIN11: ADC 通道 11 输入
			外设功能: 全映射, 见表 3-2 数字功能
3	3	P14	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			C12: 比较器通道 12 输入
			AIN12: ADC 通道 12 输入
			外设功能: 全映射, 见表 3-2 数字功能
2	2	P15	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			C13: 比较器通道 13 输入

			AIN13: ADC 通道 13 输入
			外设功能: 全映射, 见表 3-2 数字功能
			输入功能 (INT0/1、T0/1、T0G、T1G、CA0/1/2、ADET、FB)
19	15	P16	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			DSDA: 编程、调试数据输入输出
			AIN14: ADC 通道 14 输入
			外设功能: 全映射, 见表 3-2 数字功能
			输入功能 (INT0/1、T0/1、T0G、T1G、CA0/1/2、ADET、FB)
11	14	P17	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			AIN15: ADC 通道 15 输入
			外设功能: 全映射, 见表 3-2 数字功能
			输入功能 (INT0/1、T0/1、T0G、T1G、CA0/1/2、ADET、FB)
13	8	P20	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			RST: 外部复位
			OSC-: 外部低速晶振输入(JSC68F006SO16 无此功能)
			ADC_VREF: 内部 ADC 基准电压输出
			外设功能: 全映射, 见表 3-2 数字功能
			输入功能 (INT0/1、T0/1、T0G、T1G、CA0/1/2、ADET、FB)
12	-	P30	通用 GPIO: GPIO 通过寄存器配置输入输出, 上、下拉等功能
			OSC+: 外部低速晶振输出
			ADC_VREF: 外部 ADC 基准电压输入
			外设功能: 全映射, 见表 3-2 数字功能
			输入功能 (INT0/1、T0/1、T0G、T1G、CA0/1/2、ADET、FB)

表 3-2 数字功能

配置值	功能	方向	功能描述
0x00	GPIO	I/O	通用 IO 口，
0x01	ANALOG		模拟功能
0x02	T0_OUT	O	Timer0 溢出翻转输出
0x03	T0_OUT2	O	Timer0 溢出翻转输出 2
0x08	TXD0	O	UART0 数据输出
0x09	RXD0	I/O	UART0 数据输入
0x0A	TXD1	O	UART1 数据输出
0x0B	RXD1	I/O	UART1 数据输入
0x0C	SCL	I/O	I2C 时钟输入输出
0x0D	SDA	I/O	I2C 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
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	T3_PWM_OUT	O	定时器 3 的 PWM 输出
0x19	T1_OUT	O	Timer1 溢出翻转输出
0x1A	VCOUT	O	比较器输出
0x1C	PG6	O	PWM 通道 6 输出
0x1D	PG7	O	PWM 通道 7 输出

4 中央处理器

JSC68F016 是一颗增强型的 8051 微控制器，且完全兼容 MCS-51 指令集，因此可以使用目前流行的编译环境仿真（例如 Keil C51）。

4.1 存储器结构

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

数据内存占用的地址空间独立于编程内存。在 JSC68F016 中，有 256 字节内部 RAM。对于需要更多内部 RAM 的许多应用，JSC68F016 提供另外片上 1K 字节 RAM，叫做 XRAM，通过 MOVX 指令访问。

整个嵌入的 FLASH，作为编程内存的功能，被分成两块。应用 ROM（APROM）通常存储用户代码，加载 ROM（LDROM）通常存储启动代码。FLASH 控制单元支持扇区、页、芯片擦除；支持字节编程和读模式。在应用编程（IAP）或在系统编程（ISP）都可以执行这些模式。

4.2 程序存储器

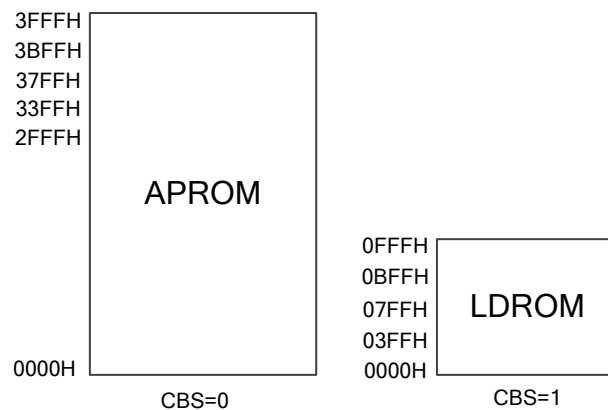
整个嵌入的 FLASH，作为编程内存的功能，被分成两块。应用 ROM（APROM）通常存储用户代码，加载 ROM（LDROM）通常存储启动代码。事实上，APROM 和 LDROM 功能相似。FLASH 可以通过扇区或者页进行划分，每一个扇区位 128 字节，每一页 1K 字节。FLASH 控制单元支持扇区、页、芯片擦除；支持字节编程和读模式。在应用编程（IAP）或在系统编程（ISP）都可以执行这些模式。

单独的编程块叫做 LDROM，它的功能通常是存储启动代码用于 ISP。它可以更新 APROM 空间和 CONFIG 字节。APROM 中的代码也可以重新编程 LDROM。APROM 和 LDROM 关于 IAP 的细节和配置位设置，请看章节 24.在应用编程 (IAP)。注意：APROM 和 LDROM 是硬件独立模块，因此如果 CPU 从 LDROM 启动，CPU 会自动重映射 PC 指针 0000H 到 LDROM 开始的地址。因此 CPU 认为 LDROM 是单独的编程内存且所有中断向量独立于 APROM

如下图所示，程序存储器的划分通过 CONFIG1[2:0]进行设置。通过 CONFIG0.7 配置，使 CPU 在执行程序时选择是从 APROM 还是 LDROM 启动。

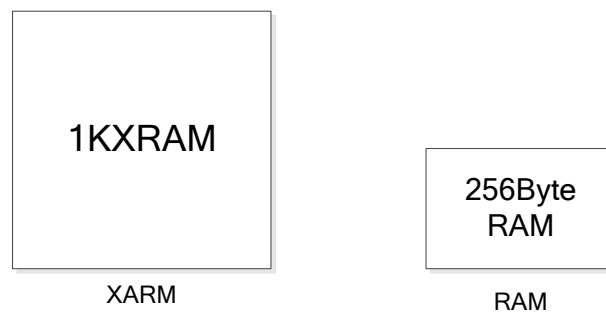
Config1:

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



APROM和LDROM的逻辑边界地址由
CONFIG1[2:0]定义

4.3 数据存储器



如上图所示内部数据内存占用一个独立于编程内存的地址空间。内部数据内存可以分割成三块。分别是 RAM 的低 128 字节, RAM 的高 128 字节, 和 SFR 空间的 128 字节。内部数据内存的地址是 8 位的宽度, 可用于 256 字节的地址空间。直接寻址高于 7FH 的地址会访问特殊功能寄存器(SFR)空间, 间接寻址高于 7FH 的地址会访问 RAM 的高 128 字节。虽然 SFR 地址空间和 RAM 高 128 字节共享相同的逻辑地址 80H 到 FFH, 事实上他们是物理独立的实体。区别于 RAM 的高 128 字节直接寻址仅可以访问 SFR。SFR 空间中的 16 个地址既可以字节寻址也可以位寻址。这些位寻址的 SFR 分布在地址以 0H 或 8H 结尾的地方

直接寻址指令(如 MOV A, direct)可以访问低 256 字节的内部 RAM(地址范围: 00-7FH)和

所有系统寄存器

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

4.4 外部 XRAM

外部 RAM 增加了变量的容量，但跟内部 RAM 相比，它的访问能力是最低的，这是因为频繁地使用变量和本地变量是为了存入内部 RAM，而外部 RAM 的绝大多数用法都是特定的。外部 RAM 可以作为优先级别较低的，或者在 ROM 中预装查表以加速访问周期的变量存储区域

4.5 累加器 ACC

ALU 是 8Bit 宽的算术逻辑单元，MCU 所有的数据、逻辑运算都是通过它来完成的，ALU 可以对数据进行加、减、移位及逻辑运算；ALU 也控制状态位，用来表示运算结果状态。

4.6 B 寄存器

B 寄存器在使用乘法和除法指令时使用，而且还能作为普通寄存器来保留临时数据。

4.7 堆栈指针寄存器 (SP)

该寄存器指向内部数据内存空间中的堆栈顶部。它用于在执行中断例程或子程序之前存储程序的返回地址。SP 在执行 PUSH 或 CALL 指令前递增，在执行 POP 或 RET(I)指令后递减(它总是指向栈顶)。

4.8 特殊功能寄存器 (SFR)

JSC68F016 用特殊功能寄存器 (SFRs) 来控制或监视外设和其它模块。SFRs 位于地址 80 到 FFH 地址空间，仅可以通过直接寻址访问。那些地址以 0H 或 8H 结尾的 SFRs 是可以位寻址的。当用户需要修改某一位而不改变其他位的情况下，位寻址是非常有用的。其它所有的 SFRs 仅可以字节寻址。JSC68F016 包含标准 8051 中出现的所有 SFRs，然而一些额外的 SFRs 也包含在内。因此在原始 8051 中一些没有使用的字节被给予了新的功能。SFRs 如下所列：

SFR 模块还具有写保护功能。JSC68F016 中有一些特殊功能，如看门狗定时器，欠压检测功能等。这些功能是系统正常运行的关键。如果没有对这些寄存器进行写保护，无关代码可能对其写

入不确定的值，结果导致不正确的操作和异常控制。为了防止这种风险，JSC68F016 的时效访问保护功能，可以限制对关键的 SFR 的写访问。此保护方案，在有效时间范围里才能访问。

表 4-1 SFR 特殊功能寄存器

BIN HEX	000	001	010	011	100	101	110	111
F8	S1CON	P0EXTF	P1EXTF	P2EXTF		AS3L	AS3H	AUXR1
F0	B	C0L	C0H	C1L	C1H	C2L	C2H	SRST
E8	ADCCON0	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
E0	ACC	SPSTA	SPCON	SPDAT		CLKSEL	CLKCMD	PWMF
D8	SCON2	ADCCON1	ADCCON2	ADCCON3	ADCDLY	P0EXTEN	P1EXTEN	P2EXTEN
D0	PSW	ADCRL	ADCRH	ADCMPH	ADOC	T3L	T3H	
C8	T2CON	T2MOD	RCMP2L	RCMP2H	TL2	TH2	T3DL	T3DH
C0	I2CCON	I2CADR	I2CDAT	I2CSTA	T3CON	RL3	RH3	TA
B8	EIE	IPH	S0RELH	S1BUF	S1RELL	S1RELH	AS2L	AS2H
B0	P3	EIPH	EIPH1	ASCON	AS0L	AS0H	AS1L	AS1H
A8	IE	IP	S0RELL	TCKCON	CKSWT	CKEN	BODCON1	P2S
A0	P2	EIP	EIP1	CAPCON0	CAPCON1	CAPCON2	BODCON0	CAPCON3
98	S0CON	S0BUF	EIE1		IAPFD	IAPUEN	CHPCON	WDCON
90	P1		DPS	DPC	IAPAL	IAPAH	IAPCN	IAPTRG
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	WKCON
80	P0	SP	DPL	DPH	DPL1	DPH1	RWK	PCON

4.9 特殊功能寄存器（XSFR）

表 4-2 特殊功能寄存器（XSFR）

地址	符号	描述
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	P06CFG	P0.6 口配置寄存器
F007H	P07CFG	P0.7 口配置寄存器
F008H	P0SUP	P0 强上拉使能寄存器
F009H		
F00AH		
F00BH		
F00CH		
F00DH		
F00EH		
--	--	--
F010H	P10CFG	P1.0 口配置寄存器
F011H	P11CFG	P1.1 口配置寄存器
F012H	P12CFG	P1.2 口配置寄存器
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
F018H	P1SUP	P1 强上拉使能寄存器
F019H		
F01AH		
F01BH		
F01CH		
F01DH		
F01EH		
--	--	--
F020H	P20CFG	P2.0 口配置寄存器
F021H	--	--
F022H	--	--
F023H	--	--
F024H	--	--
F025H	--	--
F026H	--	--

F027H	--	--
F028H	P2SUP	P2 强上拉使能寄存器
F029H		
F02AH		
F02BH		
F02CH		
F02DH		
F02EH		
--	--	--
F030H	P30CFG	P3.0 配置寄存器
F031H	--	--
F032H	--	--
F037H	--	--
F038H	P3SUP	P3 强上拉使能寄存器
F039H		
F03EH		
--	--	--
F040H	PS_INT0	外部中断 0 输入端口分配寄存器
F041H	PS_INT1	外部中断 1 输入端口分配寄存器
F042H	PS_T0	Timer0 外部时钟输入端口分配寄存器
F043H	PS_T0G	Timer0 门控输入端口分配寄存器
F044H	PS_T1	Timer1 外部时钟输入端口分配寄存器
F045H	PS_T1G	Timer1 门控输入端口分配寄存器
F046H	PS_CAP0	Timer2 输入捕获通道 0 端口分配寄存器
F047H	PS_CAP1	Timer2 输入捕获通道 1 端口分配寄存器
F048H	PS_CAP2	Timer2 输入捕获通道 2 端口分配寄存器
F049H	PS_ADET	ADC 的外部触发输入端口分配寄存器
F04AH	PS_FB	PWM 的外部刹车信号输入端口分配寄存器
F04BH		
F050H	P0OD	P0 开漏使能寄存器
F051H	P0UP	P0 上拉使能寄存器
F052H	P0RD	P0 下拉使能寄存器
F053H	P0DR	P0 驱动电流控制寄存器
F054H	P0HS	P0 高压摆率使能寄存器
F055H	P0DS	P0 数据输入选择寄存器
F056H	P0TRIS	P0 方向寄存器
F057H	P0SMT	P0 施密特输入使能寄存器
F058H	P1OD	P1 开漏使能寄存器
F059H	P1UP	P1 上拉使能寄存器
F05AH	P1RD	P1 下拉使能寄存器
F05BH	P1DR	P1 驱动电流控制寄存器
F05CH	P1HS	P1 高压摆率使能寄存器

F05DH	P1DS	P1 数据输入选择寄存器
F05EH	P1TRIS	P1 方向寄存器
F05FH	P1SMT	P1 施密特输入使能寄存器
F060H	P2OD	P2 开漏使能寄存器
F061H	P2UP	P2 上拉使能寄存器
F062H	P2RD	P2 下拉使能寄存器
F063H	P2DR	P2 驱动电流控制寄存器
F064H	P2HS	P2 高压摆率使能寄存器
F065H	P2DS	P2 数据输入选择寄存器
F066H	P2TRIS	P2 方向寄存器
F067H	P2SMT	P2 施密特输入使能寄存器
F068H	P3OD	P3 开漏使能寄存器
F069H	P3UP	P3 上拉使能寄存器
F06AH	P3RD	P3 下拉使能寄存器
F06BH	P3DR	P3 驱动电流控制寄存器
F06CH	P3HS	P3 高压摆率使能寄存器
F06DH	P3DS	P3 数据输入选择寄存器
F06EH	P3TRIS	P3 方向寄存器
F06FH	P3SMT	P3 施密特输入使能寄存器
F070H	P00EICFG	P00 口边沿触发选择寄存器
F071H	P01EICFG	P01 口边沿触发选择寄存器
F072H	P02EICFG	P02 口边沿触发选择寄存器
F073H	P03EICFG	P03 口边沿触发选择寄存器
F074H	P04EICFG	P04 口边沿触发选择寄存器
F075H	P05EICFG	P05 口边沿触发选择寄存器
F076H	P06EICFG	P06 口边沿触发选择寄存器
F077H	P07EICFG	P07 口边沿触发选择寄存器
F078H	P10EICFG	P10 口边沿触发选择寄存器
F079H	P11EICFG	P11 口边沿触发选择寄存器
F07AH	P12EICFG	P12 口边沿触发选择寄存器
F07BH	P13EICFG	P13 口边沿触发选择寄存器
F07CH	P14EICFG	P14 口边沿触发选择寄存器
F07DH	P15EICFG	P15 口边沿触发选择寄存器
F07EH	P16EICFG	P16 口边沿触发选择寄存器
F07FH	P17EICFG	P17 口边沿触发选择寄存器
F080H	P20EICFG	P20 口边沿触发选择寄存器
F081H	P30EICFG	P30 口边沿触发选择寄存器
--	--	--
F090H	PWMCON0	PWM 控制寄存器 0
F091H	PWMCON1	PWM 控制寄存器 1
F092H	PWMINTC	PWM 中断控制寄存器
F093H	PWMPL	PWM 周期低字节

F094H	PWMPH	PWM 周期高字节
F095H	PWM0L	PWM0 占空比低字节
F096H	PWM0H	PWM0 占空比高字节
F097H	PWM1L	PWM1 占空比低字节
F098H	PWM1H	PWM1 占空比高字节
F099H	PWM2L	PWM2 占空比低字节
F09AH	PWM2H	PWM2 占空比高字节
F09BH	PWM3L	PWM3 占空比低字节
F09CH	PWM3H	PWM3 占空比高字节
F09DH	PWM4L	PWM4 占空比低字节
F09EH	PWM4H	PWM4 占空比高字节
F09FH	PWM5L	PWM5 占空比低字节
F0A0H	PWM5H	PWM5 占空比高字节
F0A1H	PWM6L	PWM6 占空比低字节
F0A2H	PWM6H	PWM6 占空比高字节
F0A3H	PWM7L	PWM7 占空比低字节
F0A4H	PWM7H	PWM7 占空比高字节
F0A5H	PDTEN	PWM 死区时间使能位
F0A6H	PDTCNT	PWM 死区时间计数
F0A7H	PMEN	PWM 掩码使能
F0A8H	PMD	PWM 掩码数据
F0A9H	PNP	PWMn 负极性输出使能
F0AAH	FBD	PWM 故障刹车数据
--	--	--
F0B0H	OPCON0	OPA 控制寄存器 0
F0B1H	OPCOFG	OPA 修调寄存器
F0B2H	OPCON1	OPA 控制寄存器 1
F0C0H	T4CON	Timer4 控制寄存器
F0C1H	RL4	定时器 4 自动重载低字节
F0C2H	RH4	定时器 4 自动重载高字节
F0C3H	RCNTL4	定时器 4 计数器低字节
F0C4H	RCNTH4	定时器 4 计数器高字节
F0D0H	VCCON	比较器控制寄存器
F0D1H	VCCFG	比较器通道配置寄存器
F0D2H	VCOUTCON	比较器输出控制寄存器

4.10 时效访问保护 (TA)

JSC68F016 有几个特殊功能，如看门狗定时器，欠压检测功能等。这些功能是系统正常运行的关键。如果没有对这些寄存器进行写保护，无关代码可能对其写入不确定的值，结果导致不正确的操作和异常控制。为了防止这种风险，JSC68F016 的时效访问保护功能，可以限制对关键的 SFR 的写访问。以下寄存器是相关的时效访问保护(TA)。

对被保护的位的访问是受时间限制的。要对他进行写操作，那么时效访问窗口必须打开，否则写操作无效。当对 TA 写入 AAH 时，计数器计数 3 个时钟周期等待对 TA 写入 55H。如果在写完 AAH 后的 3 个时钟周期内再写入 55H 则时效访问窗口被打开。该窗口保持 4 个时钟周期，4 个时钟周期过后窗口自动关闭。一旦定时器窗口关闭，那么要重复上述过程来访问被保护的位。注意受 TA 保护的 SFRs 只是写需要时效访问保护，读不受保护。用户读这些受 TA 保护的寄存器，不需要对 TA 写入 AAH 和 55H 的。下面列出对时效寄存器进行访问的推荐代码。

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

在执行上述指令过程中，所有中断必须关闭，避免随中断产生的延时影响三条指令的有效时间。

若没有打开任何中断，则 CLR EA 和 SETB EA 指令则可省略

Example 1,两个受 TA 保护的寄存器一起写，

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

4.11 系统控制寄存器

TA-时控保护寄存器

地址：C7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TA	TA[7:0]							
R/W	W	W	W	W	W	W	W	W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	TA[7:0]	时控保护 TA 寄存器控制着对被保护的 SFRs 的访问权限。当需要写特殊规定的寄存器时，必须先对 TA 寄存器写入 AAH，接着是 55H，当写完这两条后，才可以有 4 个时钟周期的时间对具有时控保护的寄存器写入数据。

ACC-累加器寄存器

地址：E0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ACC	ACC[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	ACC[7:0]	标准 80C51 累加器 MCU 所有的数字、逻辑运算均通过 8Bit 宽的算数逻辑单元 ALU 完成。它可以对数据进行加、减、移位及逻辑运算。ALU 的运算结果存放在 8Bit 寄存器 ACC 中。

SP-堆栈指针寄存器

地址：81H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SP	SP[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	1	1	1

位号	符号	描述
7:0	SP[7:0]	堆栈指针寄存器 堆栈指针存储的是 RAM 的地址，该地址是堆栈的起始地址。

PSW-系统标志寄存器

地址：D0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	F0	RS[1:0]		OV	F1	P
R/W	R/W	R/W	R/W	R/W		R/W	R/W	R
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	CY	进位标志。 0: 加法运算后没有进位、减法运算有借位发生或移位后移出逻辑“0”或比较运算的结果<0; 1: 加法运算后有进位、减法运算没有借位发生或移位后移出逻辑“1”或比较运算的结果≥0。
6	AC	辅助进位标志。 0: BCD 操作时没有从 ACC 的第三位开始执行; 1: BCD 操作时从 ACC 的第三位开始执行。
5	F0	通用标志位, 可任意设定。
4:3	RS[1:0]	寄存器 bank 选择控制位, 用于选择工作寄存器 bank。 00: 00H-07H (Bank0); 01: 08H-0FH (Bank1); 10: 10H-17H (Bank2); 11: 18H-1FH (Bank3)。
2	OV	溢出标志。 0: 算术操作时, ACC 没有溢出; 1: 算术操作时, ACC 溢出。
1	F1	通用标志位, 可任意设定。
0	P	奇偶标志位。 0: A 中 1 的个数为偶数; 1: A 中 1 的个数为奇数

DPS-数据指针选择寄存器

地址：92H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPS	-							DPSEL
R/W	-							R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	DPSEL	DPTR 选择控制位 0: 选择 DPH/DPL(DPTR0); 1: 选择 DPH1/DPL1(DPTR1)。

DPC-数据指针控制寄存器

地址：93H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPC	-				NDPS	ATMS	ATMD	ATME
R/W	-				R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
3	NDPS	下一个 DPTR 选择位。 执行 MOVX @DPTR 指令后，自动加载该位到 DPSEL。
2	ATMS	自动加减数据指针（使能 ATME 位时有效）。
1	ATMD	00: 执行 MOVX @DPTR 指令后+1; 01: 执行 MOVX @DPTR 指令后-1; 10: 执行 MOVX @DPTR 指令后+2; 11: 执行 MOVX @DPTR 指令后-2。
0	ATME	自动加减数据指针功能。 0: 关闭; 1: 使能。

DPL-数据指针低字节

地址：82H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPL	DPL[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	DPL[7:0]	数据指针寄存器低 8 位 这是 16 位数据指针的低字节,DPL 和 DPH 共同作为 16 位的数据指针 DPTR 访问间接地址 RAM 或编程内存地址。

DPH-数据指针高字节

地址：83H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPH	DPH[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	DPH[7:0]	数据指针寄存器高 8 位 这是 16 位数据指针的高字节,DPH 和 DPL 共同作为 16 位的数据指针 DPTR 访问间接地址 RAM 或编程内存地址。

DPL1-数据指针低字节

地址：84H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPL1	DPL1[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	DPL1[7:0]	数据指针寄存器 1 低 8 位 这是 16 位数据指针的低字节，DPL1 和 DPH1 共同作为 16 位的数据指针 DPTR1 访问间接地址 RAM 或编程内存地址。

DPH1-数据指针高字节

地址：85H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPH1	DPH1[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	DPH1[7:0]	数据指针寄存器 1 高 8 位 这是 16 位数据指针的高字节，DPH1 和 DPL1 共同作为 16 位的数据指针 DPTR1 访问间接地址 RAM 或编程内存地址。

CKCON-扩展周期寄存器

地址：8EH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	-	PWSC[2:0]			--	--	--	--
R/W	-	R/W			--	--	--	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
6:4	PWSC[2:0]	增加读取程序存储器的周期。 000: 建议设置为缺省值（1T） 001: 1 个周期；（2T 保留） 010: 2 个周期；（3T 保留） 011: 3 个周期；（4T 保留） 100: 4 个周期；（5T 保留） 101: 5 个周期；（6T 保留） 110: 6 个周期；（7T 保留） 111: 7 个周期。（8T 保留）

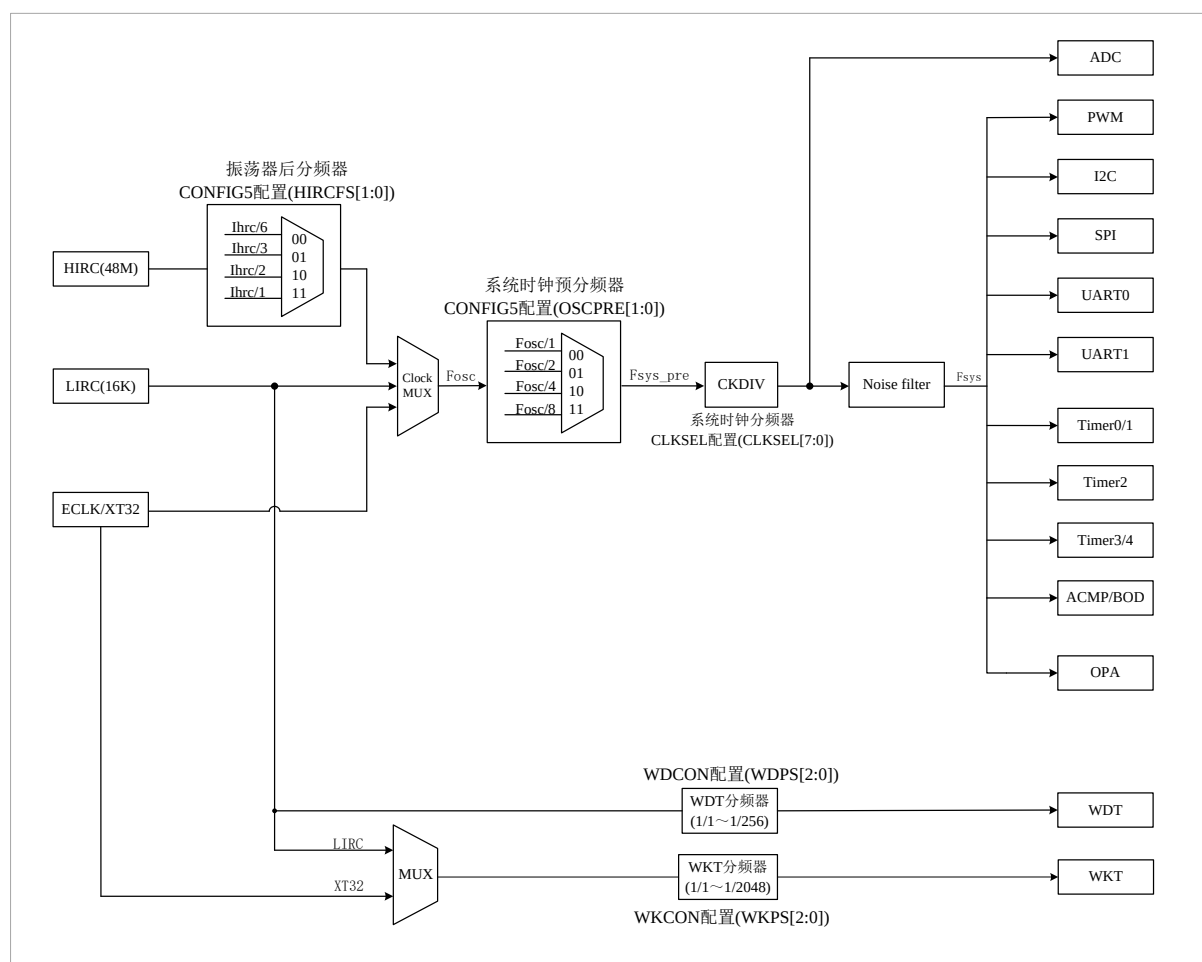
*注意：CKCON 寄存器操作需要小心，需要严格按照上面表格里面的设置，否则会出现不可预测的错乱

5 系统时钟

5.1 系统时钟概述

JSC68F016 拥有多种时钟源可供选择, 这样在应用中可以有多种选择, 使系统性能发挥到最佳, 并且功耗降到最低。JSC68F016 共有 4 种系统时钟源可供选择, 包括: 内部高速振荡器 HIRC、内部低速振荡器 LIRC、外部时钟源 ECLK 和外部低速振荡器 XT32。可以通过软件设置选择。JSC68F016 内置 2 个 RC 振荡器, 一个 16KHz 低速, 一个 32-64MHz 高速 RC 振荡器, 高速 RC 振荡器误差在出厂时校准到 $\pm 1\%$ (全温度、全电压范围内)。CLKSEL 分频寄存器选择可以灵活的调整 JSC68F016 功耗与性能。

图 5-1 系统时钟框图



5.2 系统时钟源

JSC68F016 共有 4 种系统时钟源，包括：内部高速/低速振荡器、外部输入时钟和外部低速振荡器。它们每一个都可以作为 JSC68F016 的系统时钟源。

- 内部高速振荡器（HIRC 32MHz-64MHz）
- 内部低速振荡器（LIRC 16KHz）
- 外部时钟源（ECLK 最大 16MHz）
- 外部低速振荡器（XT32 32.768KHz）

5.2.1 内部振荡器

JSC68F016 内部有两个 RC 振荡器，一个高速 32-48MHz（HIRC）和一个低速 16KHz（LIRC）。它们都可被选择用作系统时钟。通过设置 HIRCEN 位使能 HIRC，设备上电时 LIRC 被使能。用户可通过设置 CKSWT 寄存器中的 OSC[1:0]为[0,0]选择 HIRC 作为系统时钟，设置 OSC[1:0]为[1,0]选择 LIRC 作为系统时钟。注意：JSC68F016 上电后 HIRC 和 LIRC 都被使能，并且系统默认 HIRC 为系统时钟。当系统使用内部振荡器作为时钟源时，XIN 自动作为普通 I/O，P3.0 用来扩展普通 I/O 数量。

5.3 系统时钟切换

JSC68F016 可以通过软件设置 CKSWT 和 CKEN 寄存器切换时钟源。这给应用带来了很大方便。注意，这些特殊寄存器是写时效访问控制受 TA 保护寄存器。在时钟源控制下，时钟源可以在外部时钟、内部时钟，甚至内部高速与低速之间自由的切换。然而在切换时钟源时，必须确保待切换时钟源已稳定。因此，用户需要遵循一下设置步骤才能成功完成时钟源切换。用户首先要通过配置 CKEN 寄存器打开目标时钟源，再通过查询 CKSWT 寄存器中对应的标志位，确定时钟源是否稳定，并且通过写 OSC[1:0]切换到目标时钟源。这些步骤过后，将会成功的切换时钟源。如果用户关心功耗的话，可以将原先时钟源关闭了。如果不遵守以上步骤，硬件将会采取以下一些措施来应对这些违规的操作。

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

标时钟源稳定后，标志位 CKSWT 被置位，时钟将会成功切换，CKSWTF 会被硬件自动清 0。

5.4 系统时钟分频

振荡频率（Fosc）通过配置分频寄存器 CLKSEL，整数倍（最大到 1/510）分频后，再供给系统作为系统时钟（Fsys）。这一特征可以临时让 MCU 跑在很低的速度下来降低功耗。通过时钟分频，可以让 MCU 在正常工作模式下，很低的速度运行，确保其及时能够响应中断事件（比如空闲模式只能通过中断事件退出）。这有可能比空闲模式还要省电。这样，可以避免掉电模式情况下，需要等待振荡器重新起振的事件。CLKSEL 的值可以在任何时间被程序改变，除了不能在中断服务程序里改变。

*设置 CLKSEL[2:0]后，CLKCMD 寄存器必须写入 0x69 以应用 CLKSEL 的设置。

注意：当系统时钟振荡器设置为 HIRC 的 1 分频，且 Fsys_pre=Fosc/1 时，Fcpu 固定为 Fsys/2（24MHz）。

5.5 系统时钟输出

JSC68F016 提供一个 CLO 引脚可以输出系统时钟，该频率与 Fsys 频率相同。通过设置 CLOEN 打开这个功能。在掉电模式下 CLO 输出会停止，因为系统时钟已被关闭。注意当有干扰问题或是功耗问题时，用户最好关闭 CLO 输出。

5.6 系统时钟配置寄存器

CKSWT 时钟选择及开关寄存器（TA 保护）

地址：ACH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKSWT	--	--	HIRCST	LIRCST	ECLKST	OSC[1:0]		--
R/W	--	--	R	R	R	W	W	--
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
5	HIRCST	内部高速振荡器状态 0 = 高速内部振荡器不稳定或是没有开启. 1 = 高速内部振荡器开启并稳定
4	LIRCST	内部低速 16KHz 振荡器状态 0 = 内部低速振荡器不稳定或是没有开启. 1 = 内部低速振荡器开启并稳定
3	ECLKST	外部时钟或者外部晶振状态 0 = 外部去晶振或者外部时钟不稳定或没开启 1 = 外部去晶振或者外部时钟开启并稳定
2:1	OSC[1:0]	振荡器选择位 该位是用来选择系统时钟源 00 = 内部高速晶振 01 = XIN 引脚输入的外部时钟，通过 EXTEN[1:0] (CKEN[7:6]) 设置 10 = 内部 16 KHz 晶振 11 = 保留位

CKEN 时钟使能寄存器 (TA 保护)

地址: ADH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKEN	EXTEN[1:0]		HIRCEN	LIRCEN	XT32EN	--	--	CKSWTF
R/W	R/W	R/W	R/W	R/W	R/W	--	--	R
复位值	0	0	1	1	0	0	0	0

位号	符号	描述
7:6	EXTEN[1:0]	外部时钟源使能 11 = 外部时钟源使能 其他 = 外部时钟禁用. P3.0 当作通用 I/O 口
5	HIRCEN	高速内部高速振荡器使能 0 = 内部高速振荡器禁用 1 = 内部高速振荡器使能 注意, 一旦设置 IAPEN (CHPCON.0)位开启 IAP 功能, HIRC 将会自动使能, 硬件也会设置 HIRCEN 和 HIRCST 位。IAPEN 被清除后, HIRCEN 和 EHRCST 位会恢复为原始值
3	XT32EN	外部晶振使能 0 = 外部输入时钟使能 1 = 外部晶振 XT32 使能
0	CKSWTF	时钟切换错误标志位 0 = 系统时钟源切换成功 1 = 切换系统时钟的时钟源, 没有开启或不稳定。如果待切换的时钟源不稳定, 该位将一直保持为 1, 直到时钟源稳定并切换成功为止

CLKSEL --时钟分频寄存器

地址: E5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKSEL	CLKSEL[7:0]							
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

位号	符号	描述
7:0	CLKSEL[7:0]	时钟分频 系统频率 F_{sys} 计算公式 当 CLKSEL = 00 时, $F_{sys} = F_{osc}$ 当 CLKSEL = 01H-FFH 时, $F_{sys} = \frac{F_{osc}}{2 \times CLKSEL}$

*设置 CLKSEL[2:0]后, CLKCMD 寄存器必须写入 0x69 以应用 CLKSEL 的设置。

CLKCMD 时钟使能寄存器

地址：E6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCMD	CLKCMD[7:0]							
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

位号	符号	描述
7:0	CLKCMD[7:0]	写入 0x69 来使能 CLKSEL 设置

CKCON 扩展周期寄存器

地址：8EH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	CKCON[7:0]							
R/W	--	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	1	0	0	0	1

位号	符号	描述
6:4	CLKCMD[7:0]	增加读取程序存储器的周期。 000: 无; (若 $F_{cpu} \leq 24\text{MHz}$, 设置为 000) 001: 1 个周期; (若 $F_{cpu} > 24\text{MHz}$, 设置为 001) 010: 2 个周期; (保留) 011: 3 个周期; (保留) 100: 4 个周期; (保留) 101: 5 个周期; (保留) 110: 6 个周期; (保留) 111: 7 个周期; (保留)
3	ESYN	增加额外的写数据到 XRAM 的周期。(用户设置为 0)
2:0	EWSC[2:0]	增加读取 XRAM 的周期。(用户设置为 1) 000: 无; 001: 1 个周期; 010: 2 个周期; 011: 3 个周期; (保留) 100: 4 个周期; (保留) 101: 5 个周期; (保留) 110: 6 个周期; (保留) 111: 7 个周期; (保留)

注意：CKCON[2:0]位，不要修改，容易产生不可预料的结果

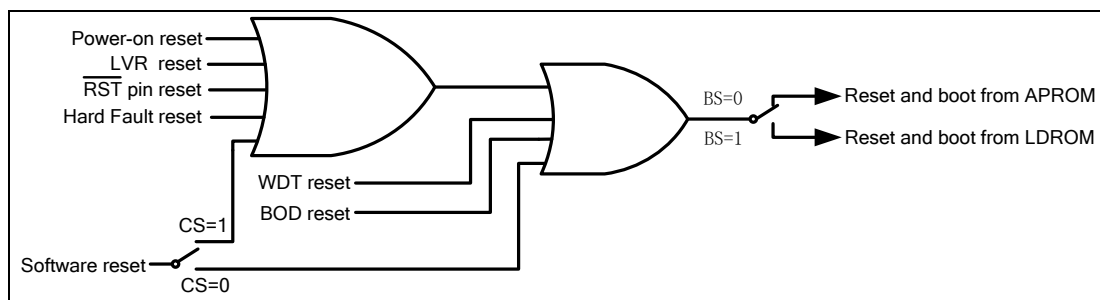
6 复位

JSC68F016 的复位条件有几种类型。通过寄存器标志位可以确定复位源。通常，大部分特殊功能寄存器复位后的值与复位条件无关，但是一些复位源的标志位的状态取决于复位源。有 7 种方法使芯片进入复位状态。

芯片有以下几种复位方式：

- 上电复位
- LVR 复位
- BOD 复位
- 外部复位
- 硬件故障复位
- 软件复位
- 看门狗复位

图 6-1 系统复位框图



上电复位、外部复位、硬件复位后芯片启动上电流程。上电流程包括 2 个阶段：电源稳定期和时钟稳定期。

在典型条件下，上电复位和 LVR 复位后，电源重新上电，上电后电源稳定期花费 5ms，然后单片机自动选择 CPU 时钟源。系统选择驱动的时钟源后，系统计数时钟周期的 4096 次以确保时钟稳定。

6.1 POR 上电复位

上电复位与 LVR 的操作相关，系统的上电过程呈逐渐上升的曲线形势，需要一定时间才能达到正常电平值。上电复位的正常时序：

- 上电：系统检测到电源电压上升等待期稳定；
- 系统初始化：所有的系统寄存器被初始化；
- 振荡器开始工作：经过一段时间的周期数开始为系统提供时钟

- 执行程序：上电结束，程序开始执行。

6.2 LVR 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 $VDD < VLVR$ ，就会请求复位

6.3 BOD 欠压复位

低电压检测监控 VDD 引脚的电压，当系统电压 VDD 低于 BOD 电压时，BOD 触发复位，触发复位的电压用户可以通过 BODCON0[6:4]和 VCCON[5:4]位进行配置欠压检测复位电压。

如果 BORST (BODCON0.2) 置 1，CPU 将欠压复位。发生欠压复位后，BORF (BODCON0.1) 通过硬件自动置 1，该位可通过软件设置或清除。

6.4 HFR 硬件故障复位

程序计数器 PC 溢出 flash 地址空间，硬件故障将发生。硬件故障复位后辅助寄存器 1 HardF(AUXR1.5)被硬件置位，这位能通过软件清零。当 MCU 运行在 OCD 调试模式并且 OCDEN=1，硬件故障复位被禁用，仅仅 HardF (AUXR1.5)标志位置位。

6.5 RST 外部复位

外部复位引脚 RST 保持最少 175us 的低电平，以确保能检测到有效的硬件复位信号，完成一次硬件复位动作。复位电路同步请求内部复位信号，因此，复位是同步运行，要求时钟在此期间运行来促使外部复位。

在复位条件下，只要引脚 RST 电平从低到高，CPU 将退出复位状态，并从地址 0000H 处开始执行代码。如果 CPU 在掉电模式下，系统通过外部引脚复位时，触发复位的方法略有不同。因为掉电模式下系统时钟是停止的，复位信号将等待系统时钟恢复。在系统时钟稳定后，CPU 将进入复位状态，然后退出，并从地址 0000H 处开始执行程序。

6.6 看门狗复位

看门狗是周期性的复位信号发生器，用于监控程序的执行流程。其内部定时器可在程序流程的检测点被清零，因此，只有在发生软件问题后才会产生实际的复位信号。

看门狗定时器是一个自由运行的定时器，带可编程溢出时间间隔和专用内部时钟源。用户可以

在任何时候清除看门狗定时器，使它重新开始计数。当选择的溢出时间间隔发生溢出后，看门狗定时器将直接复位系统。复位完成后，芯片从地址 0000H 开始运行。

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

6.7 软件复位

JSC68F016 提供软件复位功能，软件复位后，初使化 MCU 为复位状态。软件复位在 ISP 动作结束后非常有用。例如，如果通过 ISP 启动代码更新用户代码完成，软件复位能重启 CPU 立即执行用户代码。通过写两次 1 到 SRSTREQ (SRST.0) 触发软件复位。注意，执行设置 SRST 位是设备复位之前的最后指令，必须对 SRST.0 写两次，才可以启动软件复位。

如果 CHPCON.2 位 CS 位置位，则复位后加载 config 值，并根据 config 值进行系统配置，如果 CHPCON.2 位 CS 位不置位，则只复位，不进行 config 加载

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

表 6-1 复位寄存器相关复位后的值

符号	地址	地址	MSB								LSB	复位值
AUXR1	FFH	复位标志寄存器	SWRF	--	HardF	LVRF	--	--	--	--	--	POR、PINRST、LVD: 0001 0000b 软件: 100U 0000b 硬件故障: 001U 0000b 其他: U0UU 0000b
WDCON	9FH	看门狗定时器控制	WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS[2:0]				POR、PINRST、LVD: 0000_0111b WDT: 0000_1UUUb 硬件故障: 0000_0111b 其他: 0000_UUUUb
BODCON0	A6H	欠压检测控制0	BODEN	BOV[2:0]			BOF	BORST	BORF	BOS		POR、PINRST、LVD: CCCC XC0Xb BOD: UUUU XU1Xb 硬件故障: CCCC XC0Xb 其它, UUUU XUUXb
PCON	87H	电源控制	SMOD	SMOD1	POF	--	--	GF0	PD	IDL		POR、PINRST、LVD 0010 0000b 其它: 00U0 0000b
CHPCON	9EH	芯片控制	-	IAPFF	-	-	-	CS	BS	IAPEN		POR、PINRST、LVD、 硬件故障: 0000 0000b 其它: U000 0UUUb

*上述描述中软复位为不重载 config 的软复位，重载 config 的软复位复位值同硬件故障复位（HardF 位除外，此位在发生重载 config 的软复位时也清零）

复位值符号描述。

0: 逻辑 0;

1: 逻辑 1;

U: 不变;

C: 在规定复位后，SFRs 有些位初始化时设置;

X: ①BOF 复位值取决于寄存器不同的设置和 VDD 电压值

②当欠压检测使能，BOS 是只读标志，由 VDD 值来决定

BODEN	BORST	VDD 电位	BOF
1	1	总是> VBOD	0
1	0	< VBOD	1
1	0	> VBOD	0
0	X	X	0

6.8 复位寄存器

AUXR1

地址：FFH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AUXR1	SWRF	--	HardF	LVRF	LVRF18	--	--	--
R/W	R/W	--	R/W	R/W	R/W	--	--	--
复位值	0	--	0	1	1	0	0	0

位号	符号	描述
7	SWRF	软件复位标志位 当 MCU 发生软件复位后，该位硬件置 1。该位建议复位之后通过软件清零。
5	HardF	硬件故障复位标志 一旦程序计数器(PC)溢出 flash 地址空间，MCU 将复位并且 HardF 硬件置位。该位建议复位之后通过软件清零 注意：当 MCU 运行在 OCD 调试模式下并且 OCDEN=1，硬件故障复位将被禁用，仅仅 HardF 置位
4	LVRF	低压复位标志(LVR+RST 复位) 当系统电压 VDD 下跌至低于所设定的 LVR 电压时，LVR 被触发，系统复位。 该位建议复位之后通过软件清零
3	LVRF18	LVD18 低压复位标志 当系统电压下跌到低于 LVR18 电压时，LVR18 倍触发，系统复位。 建议复位后软件清零该位

PCON,电源管理寄存器

地址：87H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	SMOD1	POF	--	--	GF0	PD	IDL
R/W	R/W	R/W	R/W	--	--	R/W	R/W	R/W
复位值	0	0	1	--	--	0	0	0

位号	符号	描述
5	POF	当上电后该位置 1，用以表示当前冷复位，上电复位完成。其它任何复位不会影响，建议通过软件清零

BODCON0-欠压检测控制寄存器 0 (TA 保护)

地址: A6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BODCON0	BOD_VC_EN	REF_DIV[2:0]			BOF	BORST	BORF	BOS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
复位值	0	0	1	0	0	0	0	0

位号	符号	描述
1	BORF	欠压复位标志 当 MCU 发生欠压复位时, 该位被硬件值 1。该位建议复位之后通过软件清零。

WDCON -看门狗定时器控制寄存器 (TA 保护)

地址: 9FH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS[2:0]		
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

位号	符号	描述
3	WDTRF	WDT 复位标志 当 CPU 通过 WDT 超时事件复位后, 该位会由硬件置位。该位建议复位之后通过软件清零。

SRST -软件复位管理寄存器

地址: F7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SRST	--	--	--	--	--	--	--	SRSTREQ
R/W	--	--	--	--	--	--	--	R/W
复位值	--	--	--	--	--	--	--	0

位号	符号	描述
0	SRSTREQ	软件复位, 连续写两次启动复位 SRST = 0x01; SRST = 0x01;

7 电源管理

系统低功耗分为两种模式：

- IDLE:空闲模式
- STOP:掉电模式

复位信号和上电结束后，CPU 以 F_{cpu} 的速率开始执行程序。系统以 Normal 模式开始工作。

PCON 寄存器的最低 2 位（PD 和 IDL）控制单片机的电源管理部分。

7.1 空闲模式

若 IDL 位设置为 1，系统将会进入 IDLE 模式，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、SPI、UART 和 I2C）和时钟发生器（IHRC）仍然正常工作。任何中断都会导致单片机返回到 Normal 模式，IDL 位自动清零。

有两种方法可以中止空闲模式：

方法一：任何使能的中断发生都可以使系统退出空闲模式。中断发生自动清零 IDL 位，中止空闲模式，且将执行中断源对应的中断服务程序(ISR)。详细参见表 7-1

方法二：是除软件复位外的所有复位。详细请参见表 7-2

7.2 掉电模式

若 PD 位设置为 1，系统将会进入 STOP 模式。CPU、外设和时钟发生器都处于停止状态，在这个模式下，寄存器中存储的数据和 RAM 都保持不变。外部引脚中断可将单片机唤醒并使系统重新开始执行，PD 位自动清零。

有多种方法可以退出掉电模式：

方法一，可以通过外部中断使 JSC68F016 从掉电模式唤醒。从掉电模式唤醒的中断有：外部中断 INT0、INT1、引脚中断、WDT 中断、WTK 中断、欠压中断；详细参见表 7-1 退出低功耗方式一 唤醒

方法二，除软件复位外的所有复位。详细见表 7-2 退出低功耗方式二 复位

***注：**用户用 C 编译器开发时，强烈建议使用 IDLE 和 STOP 宏来控制单片机的系统模式，来代替直接设置 IDL 和 PD 位。

7.3 唤醒等待状态

无论是 INT0/1、外部 GPIO，还是 WDT 定时唤醒、WKT 定时唤醒 STOP 模式，在唤醒条件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。

当使用 IHRC 作为主时钟源，唤醒系统的总时间固定为 230us，当使用 ILRC 或外部时钟作为主时钟源，唤醒系统的总时间固定为 2048 个系统时钟周期，这个时候不能使用 WKT,WDT 唤醒系统。

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 STOP 模式后，任意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。

表 7-1 退出低功耗方式一 唤醒

模式	唤醒方式	唤醒条件	备注
IDLE 模式	任何中断的发生	使能全局中断 EA(IE.7), 使能其它中断源	
STOP 模式	INT0、INT1 唤醒	配置 TCON 寄存器中的 IT1 和 IT0 位, 使能唤醒触发条件	如果需要进行外部中断, 则配置全局中断 EA=1, 配置外部中断 EX1, EX0 使能, 则在唤醒后会进入外部中断服务程序
	引脚唤醒	设置端口为输入模式, 设置相应端口在 PxEXTEN 寄存器中相关使能位, 并对 PxxEICFG 相关位进行上升沿, 下降沿或者双沿唤醒条件配置。	如果同时需要端口中断, 则配置全局中断 EA=1, EPI=1, 使能外部中断
	WDT 定时唤醒	WDT 配置为普通定时器并使能, 配置相应的溢出时间, WIDPD (WDCON.4)需要设置为 1, 让 WDT 在低功耗模式下继续运行, 在定时溢出时唤醒 MCU,	如果需要 WDT 定时中断, 则需配置全局中断 EA=1, WDT 溢出中断 EWDT=1
	WKT 定时唤醒	使能 WKT, 并配置相应的溢出时间, 在溢出时唤醒 MCU	如果需要 WKT 定时中断, 则需要配置全局中断 EA=1, 配置 EIE1 寄存器中的 EWKT=1
	欠压	配置相关的 BOD 寄存器, 使能欠压检测, 当检测到 VDD 下降到 VBOD 以下时, 唤醒 MCU	如果需要欠压中断, 则配置全局中断 EA=1 和欠压中断 EBOD=1, 当 VDD 下降到 VBOD 以下或 VDD 上升到 VBOD 以上时, 产生中断

表 7-2 退出低功耗方式二 复位

模式	退出方式	退出条件	备注
IDLE 模式	LVR 低压复位	电源电压低于 LVR 电压	
	BOD 欠压复位	将 BORST 位置 1, 使能欠压复位功能, 设置 BOD 电压, 当 VDD 低于 BOD 电压时, 触发复位	
	RST 外部复位	检测到外部复位引脚 RST 信号保持一定时间 低电平	
	WDT 复位	使能 WDT, 若设置的看门狗溢出时间后没有及时清零, 将产生复位	通过 config4 中的 WDTEN[3:0]位为 0x05 或者 0x0A 使能 WDT 看门狗功能
STOP 模式	LVR 低压复位	电源电压低于 LVR 电压	
	BOD 欠压复位	将 BORST 位置 1, 使能欠压检测复位, 设置 BOD 电压, 当 VDD 低于 BOD 电压时, 触发复位	
	RST 外部复位	检测到外部复位引脚 RST 信号, 保持一定时间的低电平	
	WDT 复位	若设置的看门狗溢出时间后没有及时清零, 将产生复位	通过 config4 中的 WDTEN[3:0]位为 0x05 或者 0x0A 使能 WDT 看门狗功能

7.4 电源管理寄存器

PCON-电源管理寄存器

地址：87H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	SMOD1	POF	--	--	GF0	PD	IDL
R/W	R/W	R/W	R/W	--	--	R/W	R/W	R/W
复位值	0	0	1	--	--	0	0	0

位号	符号	描述
7	SMOD0	串口 0 波特率加倍使能。 串口 0 在模式 2 或用 Timer1 的溢出率做波特率发生器的模式 1 和模式 3，设置该位，波特率会加倍。
6	SMOD1	串口 1 波特率加倍使能。 串口 1 在模式 2 或用 Timer1 的溢出率做波特率发生器的模式 1 和模式 3，设置该位，波特率会加倍。
5	POF	当上电后该位置 1，用以标示当前冷复位，上电复位完成。其它任何复位不会影响，建议通过软件清零
2	GF0	通用标志
1	PD	掉电模式 设置该位使 MCU 进入掉电模式。在此模式下，CPU 和外设时钟停止，程序计数器（PC）挂起。此时为最小功耗。CPU 从掉电模式下唤醒后，该位自动由硬件清零，且在系统唤醒之前程序继续执行中断服务程序（ISR）。从 ISR 返回后，设备继续执行系统进入掉电模式时所处的指令。 注: 如果 IDL 位和 PD 位同时置位，MCU 进入掉电模式。从掉电模式退出后不会进入空闲模式。
0	IDL	空闲模式 设置该位使 MCU 进入空闲模式。在此模式下，CPU 时钟停止，且程序计数器（PC）挂起，但是所有外设继续工作。CPU 从空闲模式唤醒后，该位自动由硬件清零，且在系统唤醒之前程序继续执行中断服务程序（ISR）。从 ISR 返回后，设备继续执行系统进入空闲模式时所处的指令。

8 中断

8.1 概述

中断的目的是让软件处理非常规或异步的事件。JSC68F016 有 4 个中断优先级、18 个中断源。每个中断源都有独立的优先级、标志位、中断向量和使能位。另外，中断可被全局使能或关闭。当中断发生时，CPU 将执行对应的中断服务子程序(ISR)。ISR 被分配到预先指定的地址如中断向量表。如果中断使能，当中断发生时，CPU 将根据中断源跳转到相应的中断向量地址，执行此地址处的程序，保持中断服务状态直到执行中断服务程序 ISR。一旦 ISR 开始执行，仅能被更高优先级的中断抢占。ISR 通过指令 RETI 返回，该指令强迫 CPU 回到中断发生前所执行指令的下一条指令。中断源的简称，中断向量，优先级别和控制位如下表所示：

表 8-1 中断向量表

中断源	中断向量地址	使能位	优先级控制位	默认优先级
系统复位	0x0000			
INT0 外部中断 0	0x0003	IE[0]	IP[0]、IPH[0]	1
欠压检测中断	0x0043	IE[5]	IP[5]、IPH[5]	2
WDT 中断	0x0053	EIE[4]	EIP[4]、EIPH[4]	3
Timer0 中断	0x000b	IE[1]	IP[1]、IPH[1]	4
I2C 中断	0x0033	EIE[0]	EIP[0]、EIPH[0]	5
ADC 中断	0x005b	IE[6]	IP[6]、IPH[6]	6
INT1 外部中断 1	0x0013	IE[2]	IP[2]、IPH[2]	7
管脚中断	0x003b	EIE[1]	EIP[1]、EIPH[1]	8
Timer1 中断	0x001b	IE[3]	IP[3]、IPH[3]	9
串口 0 中断	0x0023	IE[4]	IP[4]、IPH[4]	10
故障刹车中断	0x0073	EIE[5]	EIP[5]、EIPH[5]	11
SPI 中断	0x004b	EIE[6]	EIP[6]、EIPH[6]	12
Timer2 中断	0x002b	EIE[7]	EIP[7]、EIPH[7]	13
定时器输入捕获	0x0063	EIE[2]	EIP[2]、EIPH[2]	14
PWM 中断	0x006b	EIE[3]	EIP[3]、EIPH[3]	15
串口 1 中断	0x007b	EIE1[0]	EIP1[0]、EIPH1[0]	16
Timer3/4 溢出中断	0x0083	EIE1[1]/[3]	EIP1[1]、EIPH1[1]	17
WKT 自唤醒定时器中断	0x008b	EIE1[2]	EIP1[2]、EIPH1[2]	18

8.2 中断使能

每一个中断源都可以通过各自的中断使能位开启或关闭，这些位在 **IE** 和 **EIE** 特殊功能寄存器 **SFRs** 中。系统有一个全局使能中断 **EA(IE.7)**位，清 **0** 将关闭所有中断，置位则启用已单独使能的非全局中断。当 **EA** 为 **0** 时有中断请求，所有中断会被挂起直到 **EA** 恢复为 **1**，才去执行该中断。

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

8.3 中断优先级

所有中断都可以设置系统提供的中断优先级设置，优先级从高到低分为 **4** 级，分别是 **3**（最高），**2**（高），**1**（低），**0**（最低）。每个中断源都可以单独设置自己的优先级，参考表 **8-2** 中断优先级设置表。中断优先级用来控制多个中断发生时的系统执行顺序。当中断源配置了优先级时，配置优先级高的可以打断配置优先级低的中断。在多个中断同时存在时，执行遵循以下规则：

1、当一个配置为低优先级的中断正在运行，这时一个配置为高优先级的中断产生，低优先级的中断会被打断去执行高优先级的中断。在高优先级的中断执行完成后，低优先级中断继续执行。低优先级中断完成后，回到主程序执行。

2、如果一个配置为高的优先级的中断正在执行，不能被其他任何中断打断：包括配置为同等高的优先级和默认优先级比现行中断的默认优先级高的中断。

3、配置为低的优先级中断只有在其他中断没有执行的情况下才能被调用。配置为低优先级的中断不能被另一个同样配置为低优先级的中断打断：即使这个中断在系统默认优先级比正在运行的中断默认优先级高。

4、如果两个中断同时发生时，配置为高优先级的中断先执行。如果两个中断配置为相同优先级，则系统默认优先级高的先执行。

系统的默认优先级可以参照表 8-1 中断向量表

表 8-2 中断优先级设置表

优先级别	IPH.x	IP.X	中断优先级
Level0	0	0	0
Level1	0	1	1
Level2	1	0	2
Level3	1	1	3

优先级别	EIPH.x	EIP.X	中断优先级
Level0	0	0	0
Level1	0	1	1
Level2	1	0	2
Level3	1	1	3

优先级别	EIPH1.x	EIP1.X	中断优先级
Level0	0	0	0
Level1	0	1	1
Level2	1	0	2
Level3	1	1	3

首先执行优先级别较高的 ISR，甚至可以打断执行中的优先级别较低的 ISR，直到优先级别较高的 ISR 执行完成之后再执行优先级别较低的

表 8-3 中断源特性描述

中断源	向量地址	中断标志	使能位	默认优先级	优先级控制位	中断服务号
复位	0000H	--	--	最高	-	--
外部中断 0	0003H	IE0	EX0	1	PX0, PX0H	0
欠压检测	0043H	BOF	EBOD	2	PBOD, PBODH	8
看门狗定时器	0053H	WDTF	EWDT	3	PWDT, PWDTH	10
定时器 0	000BH	TF0	ET0	4	PT0, PT0H	1
I2C 状态/超时	0033H	SI	EI2C	5	PI2C, PI2CH	6
ADC 中断	005BH	ADCF	EADC	6	PADC, PADCH	11
外部中断 1	0013H	IE1	EX1	7	PX1, PX1H	2
管脚中断	003BH	PxEXTF	EPI	8	PPI, PPIH	7
定时器 1	001BH	TF1	ET1	9	PT1, PT1H	3
串口 0	0023H	RI + TI	ES	10	PS, PSH	4
故障刹车中断	0073H	FBF	EFB	11	PFB, PFBH	14
SPI	004BH	SPIF+ MODF	ESPI	12	PSPI, PSPIH	9
定时器 2	002BH	TF2	ET2	13	PT2, PT2H	5
定时器输入捕获	0063H	CAPF[2:0]	ECAP	14	PCAP, PCAPH	12
PWM 中断	006BH	PWMF	EPWM	15	PPWM, PPWMH	13
串口 1	007BH	RI1 + TI1	ES_1	16	PS_1, PSH_1	15
定时器 3/4	0083H	TF3/TF4	ET3/ET4	17	PT34, PT34H	16
自唤醒定时器	008BH	WKTF	EWKT	18	PWKT, PWKTH	17

8.4 外部中断

8.4.1 INT0/INT1 中断

芯片支持 INT0,INT1 外部中断，INT0,INT1 可选择下降沿或者低电平触发中断，通过寄存器 TCON 的相关位进行设置。

8.4.2 GPIO 中断

芯片的每个 GPIO 引脚均支持外部中断。并且支持低电平、下降沿、上升沿、双沿中断。GPIO 中断设置可通过如下步骤：以 P1.0 为例

```
P10CFG = 0x00; //将 P1.0 设置为 GPIO
```

```
P1TRIS &= 0xFE; //P10 作 GPIO 输入功能
```

```
P10EICFG = 0x02; //P10 端口配置为上升沿触发中断
```

P1EXTEN |= 0x01; //P10 口使能中断

详细见 GPIO 中断控制章节

8.5 中断标志位的清除

在系统中中断标志位通过一下方式清除：

- 硬件自动清除
- 软件清除
- 读写操作清除

8.5.1 硬件自动清除标志位

支持硬件自动清除的中断有 INT0,INT1,T0,T1 产生的中断，该中断的标志位由硬件自动清除。清除的条件为：打开总中断使能 EA = 1,并且相应的中断使能位，在产生中断后系统进入相应的中断服务程序，标志位自动清除。如果中断使能关闭，则这些标志位可以使用软件进行清除。

8.5.2 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会再次进入中断服务程序。

8.5.3 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 中断标志位寄存器的传输完成标志位 SPIF，置 1 后需要先读 SPSTA，再读 SPDAT 后清零。

8.6 中断寄存器

IE - 中断使能寄存器 0

地址: A8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	EADC	EBOD	ES	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

位号	符号	描述
7	EA	使能所有中断 该位全局使能/禁止所有中断，单独使能 0 = 禁止所有中断源。 1 = 每个中断使能依靠使能单个中断，将会产生相应的单个中断。
6	EADC	使能 ADC 中断 0 = 禁用 ADC 中断 1 = 使能由 ADCF (ADCCON0.7) 产生中断
5	EBOD	使能 BOD 中断 0 = 禁用 BOD 中断 1 = 使能由 BOF (BODCON0.3)产生中断
4	ES	使能串口 0 中断 0 = 禁用串口 0 中断 1 = 使能由 TI (SCON0) 或 RI (SCON0)产生中断
3	ET1	使能定时器 1 中断 0 = 禁用定时器 1 中断 1 = 使能由 TF1 (TCON.7)产生中断
2	EX1	使能外部中断 1 0 = 禁用外部中断 1 1 = 使能由INT1产生中断
1	ET0	使能定时器 0 中断 0 = 禁用定时器 0 中断 1 = 使能由 TF0 (TCON.5)产生中断
0	EX0	使能外部中断 0 0 = 禁用外部中断 0 1 = 使能由INT0产生中断

EIE - 中断使能寄存器 1

地址: B8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE	ET2	ESPI	EFB	EWDT	EPWM	ECAP	EPI	EI2C
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

位号	符号	描述
7	ET2	定时器 2 中断使能位 0 = 禁用定时器 2 中断 1 = 使能由 TF2 (T2CON.7)产生的中断
6	ESPI	SPI 中断使能位 0 = 禁用 SPI 中断 1 = 使能由 SPIF (SPSTA.7), 或MODF (SPSTA.4)产生的中断
5	EFB	PWM 故障刹车中断使能位 0 = 禁用 PWM 故障刹车中断 1 = 使能由 FBF (PWMF.6) 产生的中断
4	EWDT	使能 WDT 中断 0 = 禁用 WDT 中断 1 =使能由 WDTF (WDCON.5)产生的中断
3	EPWM	PWM 中断使能位 0 = 禁用 PWM 中断 1 = 使能由 PWMF (PWMF.7)产生的中断
2	ECAP	定时器输入捕获中断 0 = 禁用输入捕获中断. 1 = 使能由 CAPF[2:0] (CAPCON0[2:0])产生的中断
1	EPI	外部引脚中断使能 0 = 禁用外部引脚使能 1 = 使能由管脚中断产生的中断
0	EI2C	I ² C 中断使能位 0 = 禁用 I ² C 中断 1 = 使能由 SI(I2CON.3)产生的中断

EIE1 - 中断使能寄存器 2

地址: 9AH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE1	-	-	-	-	ET4	EWKT	ET3	ES_1
R/W	-	-	-	-	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
3	ET4	定时器 4 中断使能位 0 = 禁用定时器 4 中断 1 = 当 TF4 为 1 时产生中断
2	EWKT	WKT 中断使能位 0 = 禁用 WKT 中断 1 = 当 WKTF (WKCON.4) 为 1 产生中断
1	ET3	定时器 3 中断使能位 0 = 禁用定时器 3 中断 1 = 当 TF3 (T3CON.4) 为 1 产生中断
0	ES_1	串口 1 中断使能位 0 = 禁用串口 1 中断 1 = 当 TI_1 (SCON1.1) 或 RI_1 (SCON1.0) 为 1 产生中断

*注意: EIE1 寄存器不可以进行位操作, 对 EIE1 的操作只能对寄存器进行操作

IP - 中断优先级寄存器

地址: A9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	-	PADC	PBOD	PS	PT1	PX1	PT0	PX0
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

位号	符号	描述
6	PADC	ADC 中断优先级低位
5	PBOD	BOD 检测中断优先级低位
4	PS	串口 0 中断优先级低位
3	PT1	定时器 1 中断优先级低位
2	PX1	外部中断 1 中断优先级低位
1	PT0	定时器 0 中断优先级低位
0	PX0	外部中断 0 中断优先级低位

IPH - 中断优先级高位寄存器

地址：B9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IPH	-	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H
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

位号	符号	描述
6	PADCH	ADC 中断优先级高位
5	PBODH	BOD 检测中断优先级高位
4	PSH	串口 0 中断优先级高位
3	PT1H	定时器 1 中断优先级高位
2	PX1H	外部中断 1 中断优先级高位
1	PT0H	定时器 0 中断优先级高位
0	PX0H	外部中断 0 中断优先级高位

EIP - 扩展中断优先级

地址：A1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP	PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI2C
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

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

EIPH - 扩展中断优先级高位

地址: B1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIPH	PT2H	PSPIH	PFBH	PWDTH	PPWMH	PCAPH	PPIH	PI2CH
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

位号	符号	描述
7	PT2H	定时器 2 中断优先级高位
6	PSPIH	SPI 中断优先级高位
5	PFBH	故障刹车中断优先级高位
4	PWDTH	WDT 中断优先级高位
3	PPWMH	PWM 中断优先级高位
2	PCAPH	定时器输入捕获中断优先级高位
1	PPIH	引脚中断优先级高位
0	PI2CH	I ² C 中断优先级高位

EIP1 - 扩展中断优先级 1 低位

地址: A2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	-	-	-	-	-	PWKT	PT34	PS_1
R/W	-	-	-	-	-	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
2	PWKT	WKT 中断优先级低位
1	PT34	定时器 3/4 中断优先级低位 定时器 3/4 具有相同的优先级, 当独立使用定时器 3/4 时, 中断会进入同一个中断入口, 进入中断后, 通过对定时器 3/4 中断标志位的判断来区分是哪个定时器产生的中断。 使用 32 位定时器时, 中断优先级配置也通过 PT34 和 PT34H 设置。
0	PS_1	串口 1 中断优先级低位

EIPH1 - 扩展中断优先级 1 高位

地址: B2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIPH1	-	-	-	-	-	PWKTH	PT34H	PSH_1
R/W	-	-	-	-	-	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
2	PWKTH	WKT 中断优先级高位
1	PT34H	定时器 3/4 中断优先级高位, 见 PT34 描述。
0	PSH_1	串口 1 中断优先级高位

ADCCON0 - ADC 控制寄存器 0 (可位寻址)

地址: E8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCON0	ADCF	ADCS	--	ADCHS[4:0]				
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

位号	符号	描述
7	ADCF	ADC 中断标志 当 AD 转换完成, 该位置 1。如果中断允许, 可启动中断服务。可读取到当前 AD 转换结果。该位为 1 时无法开始新一轮转换, 需要软件清零。

BODCON0-欠压检测控制寄存器 0 (TA 保护)

地址: A6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BODCON0	BOD_VC_EN	REF_DIV[2:0]			BOF	BORST	BORF	BOS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
复位值	0	0	1	0	0	0	0	0

位号	符号	描述
3	BOF	欠压中断标志 当 VDD 下降到 VBOD 以下或 VDD 上升到 VBOD 以上时, 该标志由硬件设置为逻辑 1。 如果 EBOD (EIE.5)和 EA (IE.7) 都置位, 将请求欠压检测中断。该位必须由软件清零。

S0CON - 串口控制寄存器

地址: 98H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
R/W	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

位号	符号	描述
1	TI0	UART 发送中断请求标志位, 显示 UART 串行传输的完成状态。在模式 0 位 8 结束时, 或者其他模式的停止位开始时由硬件设置为 1: 必须由软件清零。 0: 无 UART 发送中断请求 1: UART 发送中断请求
0	RI0	UART 接收中断请求标志位, 显示 UART 串行接收的完成状态。在模式 0 位 8 结束时, 或者其他模式的停止位中间时由硬件设置位 1: 必须由软件清零 0: 无 UART 接收中断请求 1: UART 接收中断请求

TCON - 定时器 0 和 1 控制寄存器

地址: 88H

Register	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 (电平) R/W(边沿)	R/W	R (电平) R/W(边沿)	R/W
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7	TF1	定时器 1 溢出标志 在定时器 1 溢出时该位置 1。当程序响应定时器 1 中断执行相应的中断服务程序时，该位自动清 0。
5	TF0	定时器 0 溢出标志 在定时器 0 溢出时该位置 1。当程序响应定时器 0 中断执行相应的中断服务程序时，该位自动清 0。
3	IE1	外部中断 1 边沿标志 当检测到边沿/电平类型时，该标志由硬件置位 如果 IT1 = 1(下降沿触发)，该位将保持置 1 直到软件清零或在外中断 1 服务程序中硬件清零 如果 IT1 = 0(低电平触发)，该标志是 $\overline{INT1}$ 输入信号逻辑电平的反转。软件不可控制。
2	IT1	外部中断 1 类型选择 该位选择 $\overline{INT1}$ 的中断触发类型是下降沿还是低电平 0 = $\overline{INT1}$ 为低电平触发 1 = $\overline{INT1}$ 为下降沿触发
1	IE0	外部中断 0 边沿标志 当检测到边沿/电平类型时，该标志由硬件置位 如果 IT0 = 1(下降沿触发)，该位将保持置 1 直到软件清零或在外中断 0 服务程序中硬件清零 如果 IT0 = 0(低电平触发)，该标志是 $\overline{INT0}$ 输入信号逻辑电平的反转。软件不可控制。
0	IT0	外部中断 0 类型选择 该位选择的中断触发类型是下降沿还是低电平 0 = $\overline{INT0}$ 为低电平触发 1 = $\overline{INT0}$ 为下降沿触发

T2CON - 定时器 2 控制寄存器

地址: C8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	TF2	-	-	-	-	TR2	-	CM_RL2
R/W	R/W	-	-	-	-	R/W	-	R/W
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7	TF2	定时器 2 溢出标志 当定时器 2 溢出或者比较数据符合, 该位置 1。如果已使能定时器 2 中断, 该位置 1 会让芯片进入定时器 2 中断运行。该位硬件置 1, 但需要软件清 0。

SPSTA - SPI 状态寄存器

地址: E1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSTA	SPIF	WCOL	SSERR	MODF	-	-	-	-
R/W	R	R	R	R	-	-	-	-
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7	SPIF	SPI 通讯完成标志位 数据传输完成后由硬件设置。 通过先后取 SPSTA 和 SPDAT 寄存器清零。
4	MODF	模式错误标志位。也可以通过软件来清除

PWMF - PWM 故障刹车控制

地址: E7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMF	PWMF	FBF	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7	PWMF	PWM 标志位 周期完成置位, 响应条件根据 INTSEL[2:0]和 INTTYP[1:0] (PWMINTC[2:0]和[5:4])设置。通过软件清除
6	FBF	故障刹车标志位 当 FBINEN 设置为 1, FB 管脚上检测到符合 FBINLS 设定的边沿信号后, 或者 ADFBEN 位置 1, ADC 比较结果为 1 后, 或者 ACMP 的 CFBEN 位置 1, 比较器边沿触发输出 PWM 刹车后, 则 FBF 位置 1。该位需要通过软件清 0。当 FBF 清 0 后, 故障刹车仍然不会释放 PWM 输出, 需要重新输出, 通过设置 PWMRUN (PWMCON0.0)为 1, 重新启动 PWM 输出。

WDCON - 看门狗定时器控制寄存器 (TA 保护)

地址：9FH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS[2:0]		
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

位号	符号	描述
5	WDTF	WDT 超时标志 该位表示 WDT 计数器的溢出。该标志必须通过软件清零。

CAPCON0 - 输入捕获控制寄存器 0

地址：A3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CAPCON0	-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0
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

位	名称	描述
2	CAPF2	输入捕获通道 2 标志位 如果输入捕获通道 2 检测到边沿触发事件发生，该位由硬件置位，由软件清零。
1	CAPF1	输入捕获通道 1 标志位 如果输入捕获通道 1 检测到边沿触发事件发生，该位由硬件置位，由软件清零。
0	CAPF0	输入捕获通道 0 标志位 如果输入捕获通道 0 检测到边沿触发事件发生，该位由硬件置位，由软件清零。

P0EXTF - P0 管脚中断标志位寄存器

地址：F9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTF	P0EXTF7	P0EXTF6	P0EXTF5	P0EXTF4	P0EXTF3	P0EXTF2	P0EXTF1	P0EXTF0
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

位号	符号	描述
7:0	P0EXTF[7:0]	P0 管脚中断标志位 0 = 管脚中断清除，对该寄存器写 0 清除标志位； 1 = 管脚中断发生；

P1EXTF - P1 管脚中断标志位寄存器

地址：FAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTF	P1EXTF7	P1EXTF6	P1EXTF5	P1EXTF4	P1EXTF3	P1EXTF2	P1EXTF1	P1EXTF0
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

位号	符号	描述
7:0	P1EXTF[7:0]	P1 管脚中断标志位 0 = 管脚中断清除, 对该寄存器写 0 清除标志位; 1 = 管脚中断发生;

P2EXTF - P20,P30 管脚中断标志位寄存器

地址：FBH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTF	--	--	--	P3EXTF0	--	--	--	P2EXTF0
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

位号	符号	描述
4	P3EXTF0	P30 管脚中断标志位 0 = 管脚中断清除, 对该寄存器写 0 清除标志位; 1 = 管脚中断发生;
0	P2EXTF0	P20 管脚中断标志位 0 = 管脚中断清除, 对该寄存器写 0 清除标志位; 1 = 管脚中断发生;

I2CCON - I2C 控制寄存器

地址：C0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCON	CR[2]	ENS1	STA	STO	SI	AA	CR[1:0]	
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

位号	符号	描述
3	SI	串行中断标志位 当进入了 I2C 的 26 个状态中的 25 个状态时, SI 标志位会被硬件置 1, 只有当 I2C 状态寄存器位 0xF8 时, SI 标志位才没有被置 1, 表示没有可用的相关状态信息。SI 标志位必须由软件清 0, 必须通过写 0 到 SI 标志才可以清 SI 标志位, 写入 1 到该位并不能更改 SI 的值

T4CON - 定时器 4 控制寄存器

XSFR 地址: F0C0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T4CON	-	-	TMOD4	TF4	TR4	T4PS[2:0]		
R/W	-	-	R/W	R/W	R/W	R/W		
复位值	0	0	0	0	0	0	0	0

位	名称	描述
4	TF4	定时器 4 溢出标志 当定时器 4 溢出, 该位置位。该位需要通过软件清零

WKCON - 自唤醒定时器控制寄存器

地址: 8FH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WKCON	--	--	WKCKS	WKTF	WKTR	WKPS[2:0]		
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

位号	符号	描述
4	WKTF	WKT 溢出标志 当 WKT 溢出, 该位置位。如果 WKT 中断和全局中断使能, 置位该位会使 CPU 执行 WKT 中断服务程序。该位不会被硬件自动清零, 必须通过软件清零。

T3CON - 定时器 3 控制寄存器

地址: C4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3CON	PWME	-	-	TF3	TR3	T3PS[2:0]		
R/W	R/W	-	-	R/W	R/W	R/W		
复位值	0	0	0	0	0	0	0	0

位	名称	描述
4	TF3	定时器 3 溢出标志 当定时器 3 溢出, 该位置位。当程序执行定时器 3 的中断服务程序, 该位通过软件置位或清零。

S1CON - 串口 1 控制寄存器

地址：F8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1CON	SM01	SM11	SM21	REN1	TB81	RB81	TI1	RI1
R/W	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

位号	符号	描述
1	TI1	串口 1 发送中断标志位 发送中断标志：模式 0 下该标志在发送完 8 位数据后由硬件置位，其它模式下在发送到数据最后一位后由硬件置位。该位必须由软件来清除。
0	RI1	串口 1 接收中断标志 该标志由硬件置位。在模式 0 中，接收到第 8 位；模式 1，2 或 3 中接收到停止位。该位必须由软件清除。

9 I/O 端口

JSC68F016 最多支持 18 个可位寻址的通用 I/O 引脚，分成 4 组 P0 到 P3。每一个端口有它的端口控制器，端口控制寄存器的写和读有不同的意思。

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 可以将相应的引脚配置为输出。清零 Px TRIS 的一个位可将相应的 PORT x 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，写相应的寄存器位 1，则端口输出高电平。写 0 则端口输出低电平

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，读取的 Px 相应位为引脚的状态。清零 Px DS 的相应位，读取的 Px 相应位为端口数据锁存器的状态；PORT x 作输入口时，读 Px 寄存器读的是引脚的状态，与 Px DS 寄存器的设置无关。

当将 PORT x 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

所有 I/O 引脚可以通过 PxOD 寄存器里对应的位选择为推挽输出或开漏输出。所有 I/O 引脚可以通过 PxSMT 寄存器里对应的位选择为 TTL 电平输入或施密特触发输入。施密特触发输入有更好的抗干扰能力。所有的 I/O 引脚可通过软件选择位控制，斜率输出能力。输出斜率控制寄存器是 PxHS。默认是慢斜率。如果用户想要增加 I/O 引脚输出速率。设置 PxHS 的相应位，将斜率设置成高速输出。

当配置 CONFIG5.5,CONFIG5.4 两个位都为 0,且 CONFIG0.2 置为 0。P2.0 被配置为普通 GPIO 引脚。详细配置见表 26-1 P20 引脚功能配置表。

9.1 GPIO 普通功能

PORTx 是双向端口，每一个端口有它的端口 9 个控制寄存器（PxDS、PxTRIS、PxUP、PxRD、PxSMT、PxOD、PxHS、PxDR、PxSUP），1 个数据寄存器 Px。通过 9 个控制寄存器可以配置输入、输出、输出斜率高低、驱动能力强弱，施密特输入，弱上拉，强上拉、弱下拉等功能。同时对于 GPIO 的即时状态通过数据寄存器 Px 进行读取。

- 输出状态：开漏输出、推挽输出
- 输入状态：SMT输入、普通反向器输入

9.1.1 数据寄存器

Px(x=0,1)-Px 数据寄存器

地址：80H--90H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Px	Px.7	Px.6	Px.5	Px.4	Px.3	Px.2	Px.1	Px.0
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

位号	符号	描述
7:0	Px [7:0]	端口 x 的 8-bit 数据 当端口为输出模式：0：对应端口输出低 1：对应端口输出高 当端口为输入模式（根据数据输入选择寄存器配置）：0：对应端口低 1：对应端口高

P2-P2 数据寄存器

地址：A0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2	--	--	--	--	--	--	--	P2.0
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P20	端口的 P20 数据 当端口为输出模式：0：对应端口输出低. 1：对应端口输出高 当端口为输入模式（根据数据输入选择寄存器配置）：0：对应端口低 1：对应端口高

P3-P3 数据寄存器

地址：B0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3	--	--	--	--	--	--	--	P3.0
R/W	R	R	R	R	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P30	端口的 P30 数据 当端口为输出模式：0：对应端口输出低. 1：对应端口输出高 当端口为输入模式（根据数据输入选择寄存器配置）：0：对应端口低 1：对应端口高

9.1.2 输出控制寄存器

PxOD(x=0,1)-Px 输出控制寄存器

XSFR 地址: F050H -- F058H

Register	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

位号	符号	描述
7:0	PxOD[7:0]	GPIO 的 Px 端输出控制 0 = 推挽输出; 1 = 开漏输出; 例: 使能 P1.2 端口开漏。P1OD = 0x04;

P2OD-P2 输出寄存器

XSFR 地址: F060H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2OD	--	--	--	--	--	--	--	P2OD0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P2OD0	GPIO 的 P20 端口输出控制 0 = 推挽输出; 1 = 开漏输出;

P3OD-P3 输出控制寄存器

XSFR 地址: F068H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3OD	--	--	--	--	--	--	--	P3OD0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P3OD0	GPIO 的 P30 端口输出控制 0 = 推挽输出; 1 = 开漏输出;

9.1.3 上拉使能寄存器

PxUP(x=0,1)-Px 上拉使能寄存器

XSFR 地址: F051H -- F059H

Register	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

位号	符号	描述
7:0	PxUP[7:0]	GPIO 的 Px 端口上拉控制 0 = 上拉关闭; 1 = 上拉使能;

P2UP-P2 上拉使能寄存器

XSFR 地址: F061H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2UP	--	--	--	--	--	--	--	P2UP0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P2UP0	GPIO 的 P20 端口上拉控制 0 = 上拉关闭; 1 = 上拉使能;

P3UP-P3 上拉使能寄存器

XSFR 地址: F069H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3UP	--	--	--	--	--	--	--	P3UP0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P3UP0	GPIO 的 P30 端口上拉控制 0 = 上拉关闭; 1 = 上拉使能;

9.1.4 下拉使能寄存器

PxRD(x=0,1)-Px 下拉使能寄存器

XSFR 地址: F052H -- F05AH

Register	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

位号	符号	描述
7:0	PxRD[7:0]	GPIO 的 Px 端口下拉控制 0 = 下拉关闭; 1 = 下拉使能;

P2RD-P2 下拉使能寄存器

XSFR 地址: F062H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2RD	--	--	--	--	--	--	--	P2RD0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P2RD0	GPIO 的 P20 端口下拉控制 0 = 下拉关闭; 1 = 下拉使能;

P3RD-P3 下拉使能寄存器

XSFR 地址: F06AH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3RD	--	--	--	--	--	--	--	P3RD0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P3RD0	GPIO 的 P30 端口下拉控制 0 = 下拉关闭; 1 = 下拉使能;

9.1.5 驱动电流控制寄存器

PxDR(x=0,1)-Px 驱动电流控制寄存器

XSFR 地址: F053H -- F05BH

Register	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	1	1	1	1	1	1	1	1

位号	符号	描述
7:0	PxDR[7:0]	GPIO 的 Px 端口驱动电流控制 0 = 弱驱动; 1 = 强驱动;

P2DR-P2 驱动电流控制寄存器

XSFR 地址: F063H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2DR	--	--	--	--	--	--	--	P2DR0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	1

位号	符号	描述
0	P2DR0	GPIO 的 P20 端口驱动电流控制 0 = 弱驱动; 1 = 强驱动;

P3DR-P3 驱动电流控制寄存器

XSFR 地址: F06BH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3DR	--	--	--	--	--	--	--	P3DR0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	1

位号	符号	描述
0	P3DR0	GPIO 的 P30 端口驱动电流控制 0 = 弱驱动; 1 = 强驱动;

9.1.6 输出速度控制寄存器

PxHS(x=0,1) -Px 斜率控制寄存器

XSFR 地址: F054H -- F05CH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxHS	PxHS7	PxHS6	PxHS5	PxHS4	PxHS3	PxHS2	PxHS1	PxHS0
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

位号	符号	描述
7:0	PxHS [7:0]	GPIO 的 Px 端口斜率控制 0 = 低速输出; 1 = 高速输出;

P2HS -P2 斜率控制寄存器

XSFR 地址: F064H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2HS	--	--	--	--	--	--	--	P2HS0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P2HS0	GPIO 的 P20 端口斜率控制 0 = 低速输出; 1 = 高速输出;

P3HS -P3 斜率控制寄存器

XSFR 地址: F06CH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3HS	--	--	--	--	--	--	--	P3HS0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P3HS0	GPIO 的 P30 端口斜率控制 0 = 低速输出; 1 = 高速输出;

9.1.7 数据输入选择寄存器

PxDS(x=0,1) -Px 数据输入选择寄存器

XSFR 地址: F055H -- F05DH

Register	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

位号	符号	描述
7:0	PxDS [7:0]	GPIO 输入选择, 配置为 GPIO 时影响读取 Px 寄存器的值 0 = 输出模式: 读为数据锁存器状态; 输入模式: 读为引脚状态; 1 = 输出/输入模式均读取引脚状态;

P2DS - P2 数据输入选择寄存器

XSFR 地址: F065H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2DS	--	--	--	--	--	--	--	P2DS0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P2DS0	GPIO 输入选择, 配置为 GPIO 时影响读取 P2 寄存器的值 0 = 输出模式: 读为数据锁存器状态; 输入模式: 读为引脚状态; 1 = 输出/输入模式均读取引脚状态;

P3DS -P3 数据输入选择寄存器

XSFR 地址: F06DH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3DS	--	--	--	--	--	--	--	P3DS0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P3DS0	GPIO 输入选择, 配置为 GPIO 时影响读取 P3 寄存器的值 0 = 输出模式: 读为数据锁存器状态; 输入模式: 读为引脚状态; 1 = 输出/输入模式均读取引脚状态;

9.1.8 GPIO 方向寄存器

PxTRIS(x=0,1) - Px 方向寄存器

XSFR 地址: F056H -- F05EH

Register	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

位号	符号	描述
7:0	PxTRIS [7:0]	GPIO 的 Px 端口方向控制 0 = 引脚配置为输入; 1 = 引脚配置为输出;

P2TRIS - P2 方向寄存器

XSFR 地址: F066H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2TRIS	--	--	--	--	--	--	--	P2TRIS0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P2TRIS0	GPIO 的 P20 端口方向控制 0 = 引脚配置为输入; 1 = 引脚配置为输出;

P3TRIS - P3 方向寄存器

XSFR 地址: F06EH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3TRIS	--	--	--	--	--	--	--	P3TRIS0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P3TRIS0	GPIO 的 P30 端口方向控制 0 = 引脚配置为输入; 1 = 引脚配置为输出;

9.1.9 施密特输入使能寄存器

PxSMT(x=0,1) - Px 施密特输入使能寄存器

地址：F057H/F05FH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxSMT	PxSMT7	PxSMT6	PxSMT5	PxSMT4	PxSMT3	PxSMT2	PxSMT1	PxSMT0
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

位号	符号	描述
7:0	PxSMT [7:0]	GPIO 的 Px 端口施密特输入使能控制 0 = 施密特输入关闭; 1 = 施密特输入使能;

P2SMT - P2 施密特输入使能寄存器

XSFR 地址：F067H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2SMT	--	--	--	--	--	--	--	P2SMT0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P2SMT0	GPIO 的 P20 端口施密特输入使能控制 0 = 施密特输入关闭; 1 = 施密特输入使能;

P3SMT - P3 施密特输入使能寄存器

XSFR 地址：F06FH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3SMT	--	--	--	--	--	--	--	P3SMT0
R/W	--	--	--	--	--	--	--	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
0	P3SMT0	GPIO 的 P30 端口施密特输入使能控制 0 = 施密特输入关闭; 1 = 施密特输入使能;

9.1.10 强上拉使能寄存器

P0SUP-P0 口强上拉使能寄存器

地址：F008H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0SUP	P0SUP7	P0SUP6	P0SUP5	P0SUP4	P0SUP3	P0SUP2	P0SUP1	P0SUP0
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

位号	符号	描述
7:0	P0SUP[7:0]	GPIO 的 P0 端口强上拉控制 0 = 强上拉关闭; 1 = 强上拉使能;

P1SUP-P1 口强上拉使能寄存器

地址：F018H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1SUP	P1SUP7	P1SUP6	P1SUP5	P1SUP4	P1SUP3	P1SUP2	P1SUP1	P1SUP0
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

位号	符号	描述
7:0	P1SUP[7:0]	GPIO 的 P1 端口强上拉控制 0 = 强上拉关闭; 1 = 强上拉使能;

P2SUP-P2 口强上拉使能寄存器

地址：F028H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2SUP	--	--	--	--	--	--	--	P2SUP0
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

位号	符号	描述
0	P2SUP0	GPIO 的 P2.0 端口强上拉控制 0 = 强上拉关闭; 1 = 强上拉使能;

P3SUP-P3 口强上拉使能寄存器

地址: F038H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3SUP	--	--	--	--	--	--	--	P3SUP0
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

位号	符号	描述
0	P3SUP0	GPIO 的 P3.0 端口强上拉控制 0 = 强上拉关闭; 1 = 强上拉使能;

9.2 复用功能

每个端口都有一个功能配置寄存器,通过功能配置寄存器 PxCFG,每个端口均能配置成任何一种数字功能。寄存器设置不同的值对应于不同的数字功能。

IO 口的功能配置寄存器有 18 个,包括 P00CFG~P07CFG,分别控制 P00~P07 的功能配置;包括 P10CFG~P17CFG,分别控制 P10~P17 的功能配置;包括 P20CFG,控制 P20 的功能配置;包括 P30CFG,控制 P30 的功能配置。

端口作复用功能时,无需配置端口方向寄存器。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或者开漏输出无效端口复用为 SCL 和 SDA 功能时,硬件强制端口为开漏输出,可以通过软件配置上拉电阻具体的数字功能与相应的配置参照表 9-1 数字配置功能表

9.2.1 端口复用功能配置寄存器

PxnCFG -PORTX 功能配置寄存器

地址: F0XNH

Register	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	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
4:0	PxnCFG [4:0]	GPIO 的 Pn.X 复用功能配置 复用功能配置参照表 10-1 最大配置功能 0x1F,默认为 00, GPIO 功能

表 9-1 数字配置功能表

配置值	功能	方向	功能描述
0x00	GPIO	I/O	通用 IO 口，
0x01	ANALOG		模拟功能
0x02	T0_OUT	O	Timer0 溢出翻转输出
0x03	T0_OUT1	O	Timer0 溢出翻转输出 1（在模式 3 对应的 TH0 溢出输出）
0x04	--	--	--
0x05	--	--	--
0x06	--	--	--
0x07	--	--	--
0x08	TXD0	O	UART0 数据输出
0x09	RXD0	I/O	UART0 数据输入
0x0A	TXD1	O	UART1 数据输出
0x0B	RXD1	I/O	UART1 数据输入
0x0C	SCL	I/O	I2C 时钟输入输出
0x0D	SDA	I/O	I2C 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
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	T3_PWM_OUT	O	定时器 3 的 PWM 输出
0x19	T1_OUT	O	Timer1 溢出翻转输出
0x1A	VCOUT	O	比较器输出
0x1B	--	--	--
0x1C	PG6	O	PWM 通道 6 输出
0x1D	PG7	O	PWM 通道 7 输出
0x1E	--	--	--
0x1F	--	--	--

- 默认为 0x00，GPIO 功能，可通过 GPIO 功能寄存器设置不同的功能。
- 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。

- 复用功能中端口为输出时没有优先级，所有信号同时输出
- 输入功能时有优先级，如果有两个或以上配置相同的输入功能时，优先级按照 P0.0.....P30 由高到低进行。

表 9-2 端口对应的模拟功能

管脚	配置	模拟功能			
P0.0		AIN0	OPIN-	ACMPIN0	
P0.1		AIN 1	OPIN+	ACMPIN1	
P0.2	DSCK	AIN 2	OPIN+	ACMPIN2	
P0.3		AIN 3	OPIN+	ACMPIN3	
P0.4		AIN 4	OPIN-	ACMPIN4	
P0.5		AIN 5	OPOUT	ACMPIN5	
P0.6		AIN 6		ACMPIN6	
P0.7		AIN 7		ACMPIN7	
P1.0		AIN 8		ACMPIN8	
P1.1		AIN 9		ACMPIN9	
P1.2		AIN 10		ACMPIN10	
P1.3		AIN 11		ACMPIN11	
P1.4		AIN 12		ACMPIN12	
P1.5		AIN 13		ACMPIN13	
P1.6	DSDA	AIN 14			
P1.7		AIN 15			
P2.0	XT32IN	内部 VREF 输出			
P3.0	XT32OUT	外部 VREF 输入			

9.2.2 端口输入功能分配寄存器

当端口选择为普通 GPIO 口，芯片内部存在的只有输入状态的数字功能，比如 INT0,INT1.....等，该类型的数字输入功能与端口复用状态无关。只要分配的端口支持数字输入，则该端口支持此功能。

表 9-3 输入功能端口分配寄存器表

序号	输入功能配置寄存器 (PS_XXX)	地址	功能	功能描述
1	PS_INT0	F040H	INT0	外部中断 0 输入端口分配寄存器
2	PS_INT1	F041H	INT1	外部中断 1 输入端口分配寄存器
3	PS_T0	F042H	T0	Timer0 外部时钟输入端口分配寄存器
4	PS_T0G	F043H	T0G	Timer0 门控输入端口分配寄存器
5	PS_T1	F044H	T1	Timer1 外部时钟输入端口分配寄存器
6	PS_T1G	F045H	T1G	Timer1 门控输入端口分配寄存器
7	PS_CAP0	F046H	CA0	Timer2 输入捕获 0 端口分配寄存器
8	PS_CAP1	F047H	CA1	Timer2 输入捕获 1 端口分配寄存器
9	PS_CAP2	F048H	CA2	Timer2 输入捕获 2 端口分配寄存器
10	PS_ADET	F049H	ADET	ADC 外部触发输入端口分配寄存器
11	PS_FB	F04AH	FB	PWM 外部刹车信号输入端口分配寄存器

注：PS_XXX 代表上表里面序号 1-11 寄存器每个寄存器都有 5 位去配置成 P0-P3 的不同端口

PS_XX -PORTX 输入功能分配寄存器

地址：F040H-F04AH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_XX	--	--	PS_XX5	PS_XX4	PS_XX3	PS_XX2	PS_XX1	PS_XX0
R/W	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

位号	符号	描述
5:0	PS_XX [5:0]	PORTX 的复用功能选择位 GPIO 功能并为输入时，通过 PS_XX 功能选择寄存器可以把表 9-3 的功能复用到端口上 Bit5-Bit0 输入功能分配控制位 0x00 = 把 PS_XX 功能分配到 P00 端口 0x01 = 把 PS_XX 功能分配到 P01 端口 0x02 = 把 PS_XX 功能分配到 P02 端口 ⋮ 0x12 = 把 PS_XX 功能分配到 P12 端口 ⋮ 0x20 = 把 PS_XX 功能分配到 P20 端口 0x30 = 把 PS_XX 功能分配到 P30 端口

- 1) 该输入功能分配结构支持多个输入功能分配到同一个端口。例如可同时将 INT0 与 CAP0 分配到 P00 口，配置如下：

P00CFG = 0x00; //P00 端口配置为 GPIO 功能

P0TRIS &= 0xFE; //P00 作 GPIO 输入功能

P0_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 &= 0xFE; //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

P0EXTEN |= 0x01; //允许 P00 的端口外部中断

9.2.3 复用功能应用注意

- 1) 复位功能的输入与端口的外部中断、端口的输入功能的结构相对独立

例如：将 P01 配置为 RXD0，同时配置 P01 为上升沿中断，并使能中断，当上升沿时触发 P01 中断

- 2) 数字信号的输入结构不受系统配置状态影响

例如：P20 口配置为外部复位口，则该端口输入模块打开。若程序中配置 P20 配置成 INT0，中断使能打开，则在复位信号采样有效之前，先执行中断服务，然后进行复位

- 3) 在调试模式下，如复用功能配置在 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复用功能配置到 DSDA 口

9.3 端口外部中断控制

9.3.1 端口中断概述

JSC68F016 每个管脚都提供管脚输入中断功能，用于检测管脚电平状态，18 个 I/O 管脚都可自由配置管脚中断。管脚中断能使 MCU 从空闲模式或掉电模式唤醒。管脚中断输入有 200 ns 的滤波。

每个管脚通过 PxEXTEN 寄存器设置使能中断功能。通过 PxEICFG 寄存器设置管脚中断的触发方式。每个管脚中断都有自己的中断标志，总共 18 个中断标志，存放在寄存器 PxEXTF 中，进入中断后可根据中断寄存器的值判断具体哪个管脚发生了中断。PxEXTF 寄存器中的所有标志位都是通过硬件置位，通过软件清 0。

端口中断的特点：

- 受控I/O：18个
- 中断开启/屏蔽功能
- 中断触发低电平/下降沿/上升沿/双沿模式

如果要使用端口的中断功能，需要以下几个步骤进行配置：

- 配置I/O口位普通GPIO功能，输入模式
- 配置对应I/O口的边沿触发选择寄存器PxEICFG
- 配置对应I/O口的中断使能寄存器PxEXTEN
- 配置IO的中断使能位EPI

9.3.2 端口中断控制寄存器

PxEXTEN - 管脚中断使能寄存器

P0EXTEN - P0 管脚中断使能寄存器

地址：DDH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTE N	P0EXTE N7	P0EXTE N6	P0EXTE N5	P0EXTE N4	P0EXTE N3	P0EXTE N2	P0EXTE N1	P0EXTE N0
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

位号	符号	描述
7:0	P0EXTEN[7:0]	P0 管脚中断和唤醒使能控制位 0 = 管脚中断和唤醒关闭； 1 = 管脚中断和唤醒使能；

P1EXTEN - P1 管脚中断使能寄存器

地址：DEH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTE N	P1EXTE N7	P1EXTE N6	P1EXTE N5	P1EXTE N4	P1EXTE N3	P1EXTE N2	P1EXTE N1	P1EXTE N0
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

位号	符号	描述
7:0	P1EXTEN[7:0]	P1 管脚中断和唤醒使能控制位 0 = 管脚中断和唤醒关闭; 1 = 管脚中断和唤醒使能;

P2EXTEN - P20, P30 管脚中断使能寄存器

地址：DFH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTE N				P3EXTE N0				P2EXTE N0
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

位号	符号	描述
4	P3EXTEN0	P30 管脚中断和唤醒使能控制位 0 = 管脚中断和唤醒关闭; 1 = 管脚中断和唤醒使能;
0	P2EXTEN0	P20 管脚中断和唤醒使能控制位 0 = 管脚中断和唤醒关闭; 1 = 管脚中断和唤醒使能;

9.3.3 PxEXTF 管脚中断标志位寄存器

P0EXTF - P0 管脚中断标志位寄存器

地址：F9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTF	P0EXTF7	P0EXTF6	P0EXTF5	P0EXTF4	P0EXTF3	P0EXTF2	P0EXTF1	P0EXTF0
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

位号	符号	描述
7:0	P0EXTF[7:0]	P0 管脚中断标志位 0 = 管脚中断清除, 对该寄存器写 0 清除标志位; 1 = 管脚中断发生;

P1EXTF - P1 管脚中断标志位寄存器

地址：FAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTF	P1EXTF7	P1EXTF6	P1EXTF5	P1EXTF4	P1EXTF3	P1EXTF2	P1EXTF1	P1EXTF0
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

位号	符号	描述
7:0	P1EXTF[7:0]	P1 管脚中断标志位 0 = 管脚中断清除，对该寄存器写 0 清除标志位； 1 = 管脚中断发生；

P2EXTF - P20,P30 管脚中断标志位寄存器

地址：FBH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTF	--	--	--	P3EXTF0	--	--	--	P2EXTF0
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

位号	符号	描述
4	P3EXTF0	P30 管脚中断标志位 0 = 管脚中断清除，对该寄存器写 0 清除标志位； 1 = 管脚中断发生；
0	P2EXTF0	P20 管脚中断标志位 0 = 管脚中断清除，对该寄存器写 0 清除标志位； 1 = 管脚中断发生；

PXxEICFG - PXx 端口边沿触发选择寄存器

地址：F070H-F081H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PXxEICFG	--	--	--	--	--	--	PXxEICFG1	PXxEICFG0
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

位号	符号	描述
1:0	PXxEICFG [1:0]	端口管脚中断边沿触发选择 00 = 低电平触发； 01 = 下降沿触发； 10 = 上升沿触发； 11 = 双沿触发；

每个 Px 端口的中断控制寄存器有 8 个, 包括 Px0EICFG~Px7EICFG, 这些对应着 Px0-Px7 的外部中断。例: P0 口管脚 2 的中断配置:

P02EICFG[1:0] = 00: 低电平触发;

01: 下降沿触发;

10: 上升沿触发;

11: 双沿触发;

10 看门狗

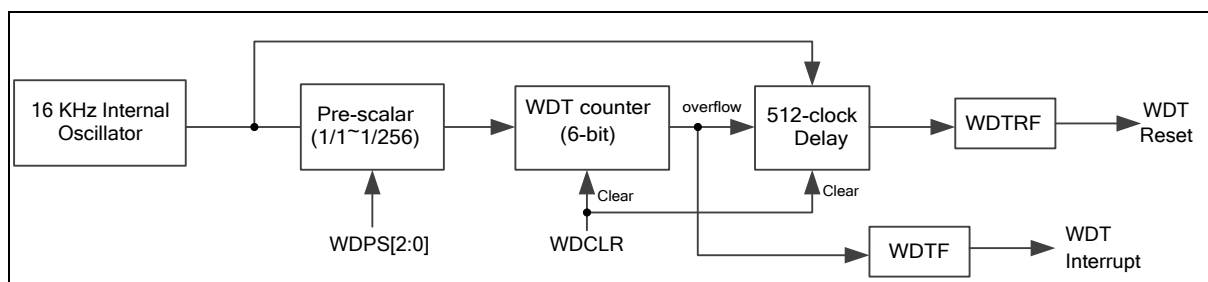
JSC68F016 提供一个看门狗定时器(WDT)，它是一个超时复位定时器用于复位整个设备。一旦由于外界干扰设备进入非正常状态或挂起，看门狗可以复位恢复系统。这有利于监测系统运行以提高系统可靠性。对于容易受到噪声，电源干扰或静电放电干扰的系统，是十分有用的。WDT 也可以配置成通用定时器，用于周期中断服务作为事件定时器。WDT 可以工作在空闲模式或掉电模式。WDT 通过 WDTEN[3:0](CONFIG4[7:4])初始化 WDT 工作在超时复位定时器或通用定时器模式。

WDT带一个独立的分频器用于分频16K LIRC时钟。分频器的时钟分频可选，来决定超时间隔。当达到超时间隔，系统会被从空闲或掉电模式唤醒，且如果WDT中断使能会产生一个中断事件。如果WDT通过CONFIG配置为一个超时复位定时器，在经过一个延时周期而软件没有任何动作后会产生系统复位。

10.1 超时复位定时器

在配置 CONFIG 位 WDTEN[3:0] (CONFIG4[7:4]) 不是 00H 时，WDT 将配置为一个超时复位定时器。如果 WDTEN[3:0]是 05H，WDT 在系统进入空闲或掉电模式后允许继续运行。如果当 WDTEN[3:0]是 0AH，WDT 系统进入空闲或掉电模式后停止运行。注意当 WDT 配置为超时复位定时器时，WDTR 和 WIDPD 没有作用。

图 10-1 看门狗复位定时器功能框图



在设备上电后，系统开始执行软件代码，同时 WDT 开始计数。超时间隔时间通过 WDP5[2:0] (WDCON[2:0])配置。当配置的超时事件发生，WDT 会置位中断标志 WDTF (WDCON.5)。如果 WDT 中断使能位 EWDT (EIE.4)和全局中断使能 EA 都置位，WDT 中断程序被执行。同时如果系统在正常运行中，在 512 个 LIRC 时钟延时期间，系统可通过置位 WDCLR 来清零计数器来避免系统被 WDT 复位。如果在这 512 个 LIRC 时钟内没有写 1 到 WDCLR，WDT 复位将会发生。置位 WDCLR 位用来清零 WDT 计数器。如系统正常运行，该位是自我清零。一旦 WDT 复位发生，WDT 复位标志 WDTRF (WDCON.3)将会被置位。用户可以通过软件清零 WDTRF。

注意：WDT 计数器需特别注意。下面事件发生时，需要对 WDT 计数器和中断进行清零：

- (1)进入空闲或掉电模式，或从空闲或掉电模式唤醒
- (2)重启。它能阻止不可预测的系统重启。
- (3)在溢出中断中要清除计数器和中断标志

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

看门狗定时器溢出时间间隔

$$T = \frac{1}{F_{LIRC} \times \text{clockdividerscalar}} \times 64$$

F_{LIRC} 是内部振荡 16KHz 的频率。下表所示的分频超时间隔表：

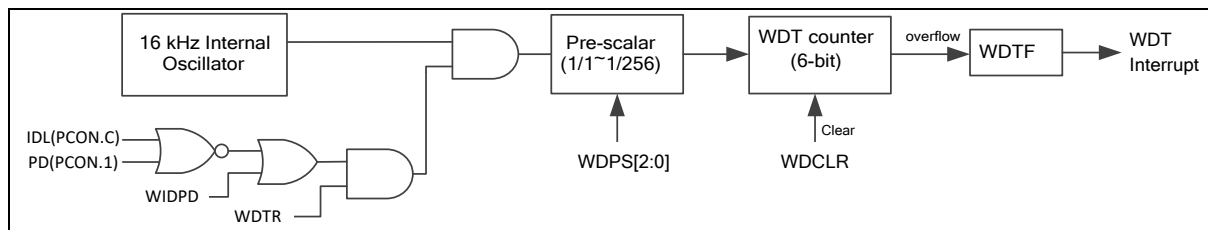
表 10-1 WDT 溢出时间

WDPS.2	WDPS.1	WDPS.0	时钟除频	看门狗溢出时间 ($F_{LIRC} = 16 \text{ KHz}$)
0	0	0	1/1	4 ms
0	0	1	1/4	16 ms
0	1	0	1/8	32 ms
0	1	1	1/16	64 ms
1	0	0	1/32	128 ms
1	0	1	1/64	256 ms
1	1	0	1/128	512 ms
1	1	1	1/256	1.024 s

10.2 通用定时器

WDT 的另一个应用是用作简单的,长周期定时器。当 CONFIG 位 WDTEN[3:0] (CONFIG4[7:4]) 是 00H, WDT 配置为通用定时器。在这种模式下, WDTR 和 WIDPD 是可以通过软件进行访问操作。

图 10-2 看门狗通用定时器功能框图



WDT 通过设置 WDTR 为 1 开始运行,通过清零 WDTR 停止。当 WDT 运行的时间到配置的时间间隔到后, WDTF 标志会置位。可通过软件查询 WDTF 标志来检测是否超时。如果 EWDT (EIE.4) 和 EA 置位, WDT 会产生中断,接着 WDT 会继续计数。用户可通过查询 WDTF 标志或等待中断发生, WDTF 位必须软件清零并等待下一次溢出。

在一些低功耗的应用中,为节省功耗, CPU 在没有处理事件时常处于空闲模式。通过运行定时器 0~3, 周期性的唤醒察看是否有需求响应,然后这种空闲模式下耗电量将达到毫安(mA)级,为了减少耗电量达到微安(μA)级的要求,当没有需求需要响应的时候, CPU 应该停留在掉电模式,并且可以通过编程的时间间隔来唤醒。JSC68F016 配备了很有用的 WDT 唤醒功能,由于基于内部 16KHz 的 RC 时钟源,看门狗定时器功耗非常低,它能够在掉电模式下计数并唤醒 CPU

注:

1. 如果 CONFIG4 中 WDTEN[3:0]配置为: 0x55 或者 0xAA, 则 WDT 始终被使能,而与 WDTR 控制位的状态无关。且 WDT 的溢出复位功能强制打开。
2. 如果 CONFIG4 中 WDTEN[3:0]配置为: 0x00, 则可以使用 WDTR 控制位使能或禁止 WDT。

10.3 看门狗定时器寄存器

WDCON - 看门狗定时器控制寄存器 (TA 保护)

地址：9FH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	WDTR	WDCLR	WDTF	WIDPD	WDTRF	WDPS[2:0]		
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

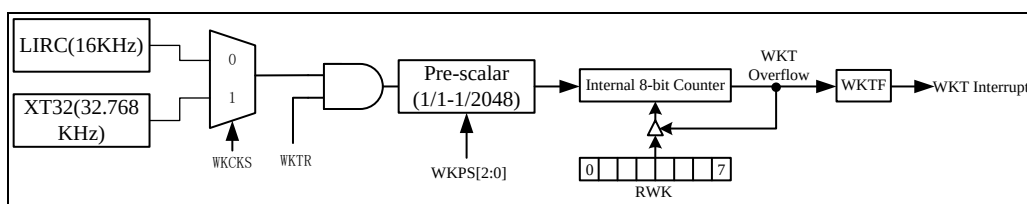
位号	符号	描述
7	WDTR	WDT 运行 该位仅当控制位 WDTEN[3:0] (CONFIG4[7:4])全为 0 时有效。这时 WDT 工作在通用定时器模式。 0 = WDT 禁止。 1 = WDT 使能。WDT 计数器开始运行。
6	WDCLR	WDT 清除 设置该位将复位 WDT 计数到 00H。它使计数器到一个已知的状态，防止系统出现不可预知的复位。写和读 WDCLR 位意思是不一样的。 <u>写:</u> 0 = 无影响 1 = 清 WDT 计数器。 <u>读:</u> 0 = WDT 计数器完全清零 1 = WDT 计数器还没有清零。
5	WDTF	WDT 超时标志 WDT 计数器的溢出时置位。该标志必须通过软件清零。
4	WIDPD	WDT 工作在空闲或掉电模式 该位仅当控制位 WDTEN[3:0] (CONFIG4[7:4])全为 0 时有效。它决定 WDT 作为通用定时器在空闲或掉电模式下是否保持工作。 0 = WDT 在空闲或掉电模式下停止工作 1 = WDT 在空闲或掉电模式下保持工作。
3	WDTRF	WDT 复位标志 当 CPU 通过 WDT 超时事件复位后，该位会由硬件置位。该位建议复位之后通过软件清零。
2:0	WDPS[2:0]	WDT 时钟预分频选择 这些位决定了 WDT 时钟的预分频，从 1/1 到 1/256。默认是最大分频值。详细见表 10-1 WDT 溢出时间

11 WKT 自唤醒定时器

WKT 模块用于低功耗下的周期唤醒芯片，也可以作通用定时器。WKT 可保持计数在空闲或掉电模式下。当 WKT 用作唤醒定时器时，WKT 要在进入省电模式之前开启。WKT 可以通过 WKCKS(WKCON.5)选择时钟源：内部 16KHz 时钟源 LIRC（系统时钟频率必须大于 WKT 时钟两倍以上）或者外部 XT32 时钟源。如果 WKT 开始计数，在设备进入空闲或掉电模式下，选择的时钟源也要保持工作（选择的 WKT 时钟不会连同 WKT 的配置自动使能，用户应该手动使能选择的时钟源并等待它稳定确保操作的成功）。

WKT 配备了一个简单的 8 位自动重装载向上计数定时器。它的预分频可选择从 1/1 到 1/2048，通过 WKPS[2:0](WKCON[2:0])来设置。用户填重装载值到 RWK 寄存器来决定它的溢出速率。WKTR(WKCON.3)置位开始计数。当计数器溢出，WKTF(WKCON.4)置为 1，并重载 RWK 寄存器的值到内部 8 位计数器。如果 EWKT(EIE1.2)置 1，WKT 中断服务程序被执行。

图 11-1 自唤醒定时器功能框图



WKT 寄存器

WKCON - 自唤醒定时器控制寄存器

地址：8FH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WKCON	--	--	WKCKS	WKTF	WKTR	WKPS[2:0]		
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

位号	符号	描述
5	WKCKS	WKT 时钟源选择 0 = LIRC 1 = XT32
4	WKTF	WKT 溢出标志 当 WKT 溢出, 该位置位。如果 WKT 中断和全局中断使能, 置位该位会使 CPU 执行 WKT 中断服务程序。该位不会被硬件自动清零, 必须通过软件清零。
3	WKTR	WKT 运行控制 0 = WKT 停止 1 = WKT 开始运行 注意重载寄存器 RWK 仅在 WKT 停止(WKTR 位为 0)的时候可以写入。否则结果是不可预知的。
2:0	WKPS[2:0]	WKT 预分频 000 = 1/1 001 = 1/4 010 = 1/16 011 = 1/64 100 = 1/256 101 = 1/512 110 = 1/1024 111 = 1/2048

RWK - 自唤醒定时器重载数据寄存器

地址：86H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RWK	RWK[7:0]							
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

位号	符号	描述
7:0	RWK[7:0]	WKT 重载数据 用来保存 WKT 的 8 位重载值。如果分频比是 1/1, RWK 限制不能是 FF

12 定时器 0/1

定时器/计数器 0 和 1 是 2 个 16 位定时器/计数器。定时器 0 与定时器 1 的类型和结构相似，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。通过 TMOD.2 和 TMOD.6 可以分别选择定时器 0/1 的定时器或计数器功能。

- 在“定时器模式”时，定时器将对定时器时钟周期进行计数。
- 在“计数器模式”时，每当检测到外部输入脚 T0 或 T1 上的下降沿，计数寄存器的内容就会加一。

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、TL0}和{TH1、TL1}组成。定时器 0、1 在四种相同的模式下工作。其模式描述如下：

表 12-1 定时器 0/1 模式

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 配置为 8 位定时器/计数器，定时器 1 停止

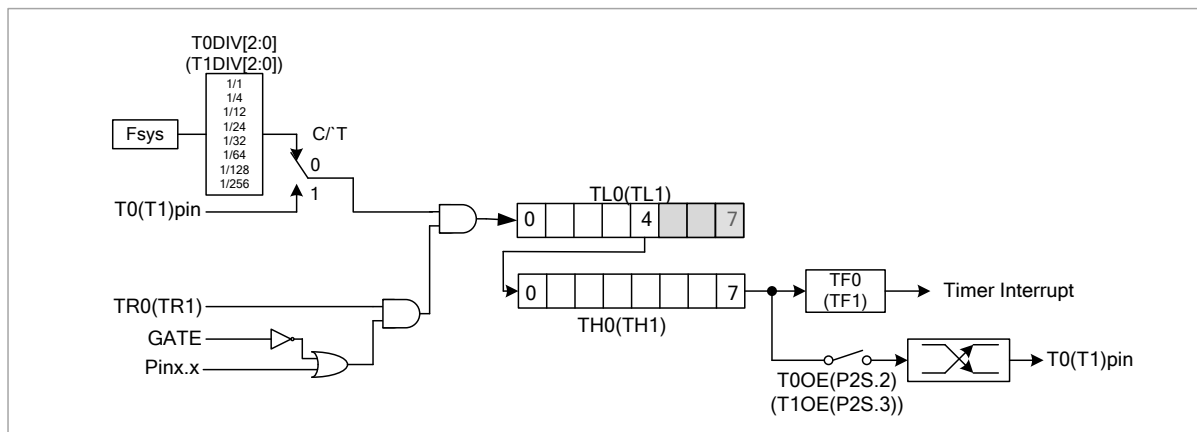
当有定时器溢出发生，定时器 0 和 1 能配置引脚 T0/T1 自动翻转输出。这个功能通过设 P2S 寄存器的 T0OE 和 T1OE 来设置，分别对应于定时器 0 和定时器 1。当打开这个功能，输出端口在第一个定时溢出之前输出逻辑 1。

定时的启动由 TRx 和 GATE 位共同控制，当门控位不使能时，定时器/计数器的启动由 TRx 控制。当门控位使能时，定时器/计数器的启动由 TRx 和 Pinx（由 I/O 功能配置使能）共同控制，这时当 TRx 置位后，Pinx 高电平时启动定时器/计数器。

12.1 模式 0（13 位定时器）

在模式 0，定时器/计数器是 13 位的计数器。13 位的计数器由 TH0 (TH1) 和 TL0 (TL1) 的低五位组成。TL0 (TL1) 的高三位被忽略。当 TR0 (TR1)=1 且 GATE=0 时或 TR0 (TR1)=1, GATE=1, pinx.x=1 时定时器/计数器使能。当 13 位的定时器计数值从 1FFFH 变为 0000H 后，定时器溢出标志 TF0 (TF1) 置位，如果中断打开，此时会产生一个定时器中断。

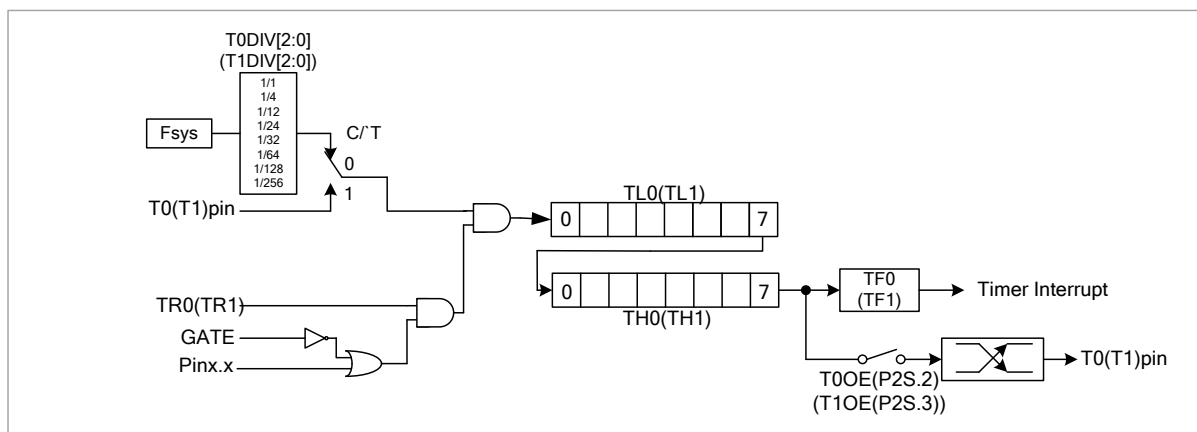
图 12-1 模式 0 框图



12.2 模式 1（16 位定时器）

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

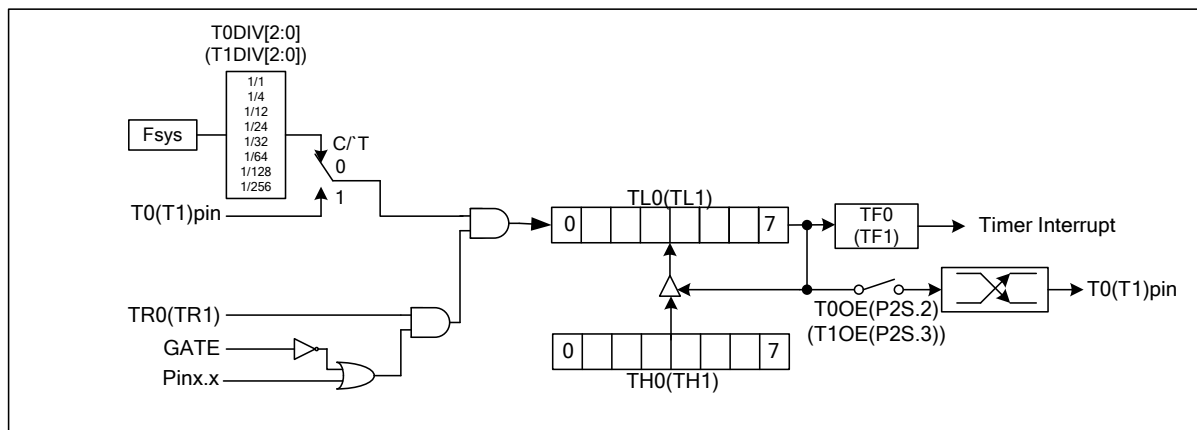
图 12-2 模式 1 框图



12.3 模式 2（8 位自动重装载定时器）

模式 2 下定时器/计数器为自动重装模式。此模式下 TL0(TL1)是一个 8 位的计数器，TH0(TH1)保存重装计数值。当 TL0(TL1)溢出后，TCON 中的 TF0(TF1)标志置位且 TH0(TH1)中内容重装至 TL0(TL1)，然后继续计数过程。重装过程中 TH0(TH1)内的值保持不变。该特征能很好的适用于 UART 波特率发生器，不需要连续软件介入。

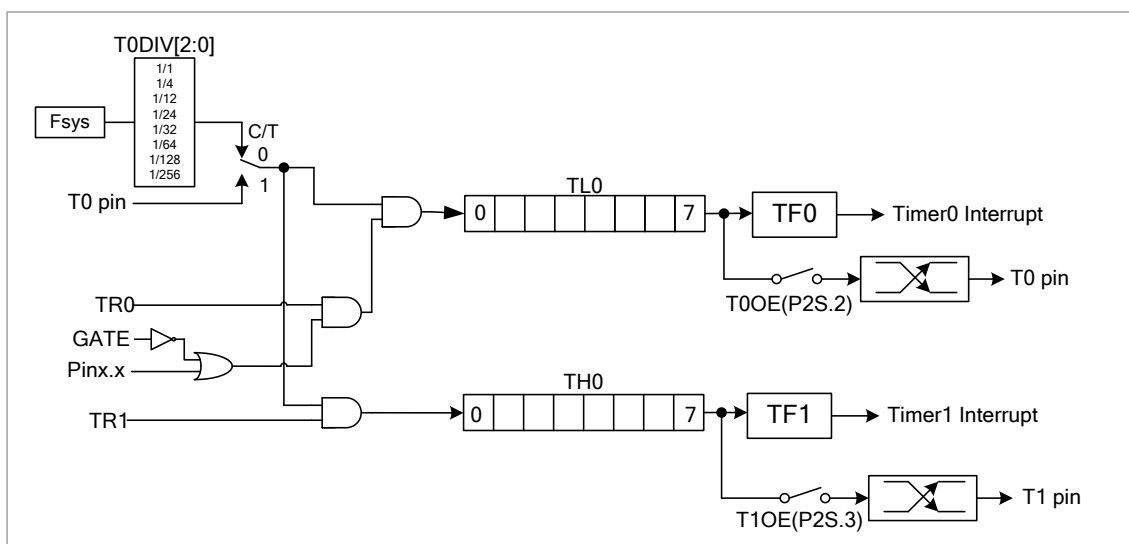
图 12-3 模式 2 框图



12.4 模式 3（两组独立 8 位定时器）

定时器 0 和定时器 1 的模式 3 有着不同的工作方式。对定时器/计数器 1 来说模式 3 会将其停用；对定时器/计数器 0 来说，模式 3 下 TL0 和 TH0 是 2 个独立的 8 位计数寄存器。模式 3 下 TL0 使用定时器 0 的控制位：如 C/T，GATE，TR0 和 TF0。TL0 也可以用来对 T0 脚上的 1 到 0 跳变计数，由 C/T(TM0D.2)来决定。TH0 使用定时器/计数器 1 的控制位（TR1 和 TF1）。当需要额外的 8 位定时器时可以使用模式 3。当定时器 0 配置为模式 3 时，定时器 1 可以通过配置其进入或离开模式 3 的方式来打开或关闭自己。定时器 0 和 1 在模式 3 下可以输出到外部。定时器 1 依然可以工作在模式 0、1、2 下，但它的灵活性受到限制。虽然基本功能得以维持，但不能对 TF1 和 TR1 进行控制。此时定时器 1 依然可以使用 GATE、T1M、外部引脚。

图 12-4 模式 3 框图



12.5 定时器 0/1 寄存器

TCON - 定时器 0 和 1 控制寄存器

地址: 88H

Register	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 (电平) R/W(边沿)	R/W	R (电平) R/W(边沿)	R/W
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7	TF1	定时器 1 溢出标志 在定时器 1 溢出时该位置 1。当程序响应定时器 1 中断执行相应的中断服务程序时，该位自动清 0。软件也可对其写 1 或写 0。
6	TR1	定时器 1 启动控制 0 = 定时器 1 中止。清该位将中止定时器 1 并且当前计数值将保存到 TH1 和 TL1 中 1 = 使能定时器 1
5	TF0	定时器 0 溢出标志 在定时器 0 溢出时该位置 1。当程序响应定时器 0 中断执行相应的中断服务程序时，该位自动清 0。软件也可对其写 1 或写 0。
4	TR0	定时器 0 启动控制 0 = 定时器 0 中止。清该位将中止定时器 0 并且当前计数值将保存到 TH0 和 TL0 中 1 = 使能定时器 0
3	IE1	外部中断 1 边沿标志 当检测到边沿/电平类型时，该标志由硬件置位 如果 IT1 = 1(下降沿触发)，该位将保持置 1 直到软件清零或在外中断 1 服务程序中硬件清零 如果 IT1 = 0(低电平触发)，该标志是 INT1 输入信号逻辑电平的反转。软件不可控制。
2	IT1	外部中断 1 类型选择 该位选择 INT1 的中断触发类型是下降沿还是低电平 0 =INT1 为低电平触发 1 =INT1 为下降沿触发
1	IE0	外部中断 0 边沿标志 当检测到边沿/电平类型时，该标志由硬件置位 如果 IT0 = 1(下降沿触发)，该位将保持置 1 直到软件清零或在外中断 0 服务程序中硬件清零。 如果 IT0 = 0(低电平触发)，该标志是 INT0 输入信号逻辑电平的反转。软件不可控制。
0	IT0	外部中断 0 类型选择 该位选择的中断触发类型是下降沿还是低电平 0 =INT0 为低电平触发。 1 =INT0 为下降沿触发。

TMOD - 定时器 0 及 1 模式寄存器

地址：89H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
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

位	名称	描述															
7	GATE	定时器 1 门选择 0 = 禁止, 定时器启动由 TRx 位控制 1 = 使能, 定时器启动由 TRx 位和外部使能引脚 Pinx 高电平时共同控制															
6	C/T	定时器 1 计数器/定时器选择 0 = 定时器 1 随内部时钟而递增 1 = 定时器 1 随外部引脚 T1 的下降沿递增															
5	M1	定时器 1 模式选择															
4	M0	<table border="1"> <thead> <tr> <th>M1</th><th>M0</th><th>定时器 1 模式</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>模式 0: 13 位定时器/计数器</td></tr> <tr> <td>0</td><td>1</td><td>模式 1: 16 位定时器/计数器</td></tr> <tr> <td>1</td><td>0</td><td>模式 2: 8 位定时器/计数器, 数值从 TH1 自动重装载入 TL1</td></tr> <tr> <td>1</td><td>1</td><td>模式 3: 定时器 1 停止</td></tr> </tbody> </table>	M1	M0	定时器 1 模式	0	0	模式 0: 13 位定时器/计数器	0	1	模式 1: 16 位定时器/计数器	1	0	模式 2: 8 位定时器/计数器, 数值从 TH1 自动重装载入 TL1	1	1	模式 3: 定时器 1 停止
M1	M0	定时器 1 模式															
0	0	模式 0: 13 位定时器/计数器															
0	1	模式 1: 16 位定时器/计数器															
1	0	模式 2: 8 位定时器/计数器, 数值从 TH1 自动重装载入 TL1															
1	1	模式 3: 定时器 1 停止															
3	GATE	定时器 0 门选择 0 = 禁止, 定时器启动由 TRx 位控制 1 = 使能, 定时器启动由 TRx 位和外部使能引脚 Pinx 高电平共同控制															
2	C/T	定时器 0 计数器/定时器选择 0 = 定时器 0 随内部时钟而递增 1 = 定时器 0 随外部引脚 T0 的下降沿递增															
1	M1	定时器 0 模式选择															
0	M0	<table border="1"> <thead> <tr> <th>M1</th><th>M0</th><th>定时器 0 模式</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>模式 0: 13 位定时器/计数器</td></tr> <tr> <td>0</td><td>1</td><td>模式 1: 16 位定时器/计数器</td></tr> <tr> <td>1</td><td>0</td><td>模式 2: 8 位定时器/计数器, 数值从 TH0 自动重装载入 TL0</td></tr> <tr> <td>1</td><td>1</td><td>模式 3: TL0 配置为 8 位定时器/计数器, TH0 配置为 8 位定时器</td></tr> </tbody> </table>	M1	M0	定时器 0 模式	0	0	模式 0: 13 位定时器/计数器	0	1	模式 1: 16 位定时器/计数器	1	0	模式 2: 8 位定时器/计数器, 数值从 TH0 自动重装载入 TL0	1	1	模式 3: TL0 配置为 8 位定时器/计数器, TH0 配置为 8 位定时器
M1	M0	定时器 0 模式															
0	0	模式 0: 13 位定时器/计数器															
0	1	模式 1: 16 位定时器/计数器															
1	0	模式 2: 8 位定时器/计数器, 数值从 TH0 自动重装载入 TL0															
1	1	模式 3: TL0 配置为 8 位定时器/计数器, TH0 配置为 8 位定时器															

TL0 - 定时器 0 低字节

地址：8AH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL0	TL0[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

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

TL1 - 定时器 1 低字节

地址：8BH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL1	TL1[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

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

TH0 - 定时器 0 高字节

地址：8CH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH0	TH0[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

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

TH1 - 定时器 1 高字节

地址：8DH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH1	TH1[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

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

TCKCON - 时钟控制寄存器

地址: ABH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCKCON	T1DIV[2:0]			T0DIV[2:0]			PWMCKS	CLOEN
R/W	R/W			R/W			R/W	R/W
复位值	0	0	0	0	0	0	0	0

位	名称	描述																		
7:5	T1DIV[2:0]	定时器 1 时钟选择 Timer1 分频比: <table><tr><th>T1DIV[2:0]</th><th>分频比</th></tr><tr><td>000</td><td>1/1</td></tr><tr><td>001</td><td>1/4</td></tr><tr><td>010</td><td>1/12</td></tr><tr><td>011</td><td>1/24</td></tr><tr><td>100</td><td>1/32</td></tr><tr><td>101</td><td>1/64</td></tr><tr><td>110</td><td>1/128</td></tr><tr><td>111</td><td>1/256</td></tr></table>	T1DIV[2:0]	分频比	000	1/1	001	1/4	010	1/12	011	1/24	100	1/32	101	1/64	110	1/128	111	1/256
T1DIV[2:0]	分频比																			
000	1/1																			
001	1/4																			
010	1/12																			
011	1/24																			
100	1/32																			
101	1/64																			
110	1/128																			
111	1/256																			
4:2	T0DIV[2:0]	定时器 0 时钟选择 Timer0 分频比: <table><tr><th>T0DIV[2:0]</th><th>分频比</th></tr><tr><td>000</td><td>1/1</td></tr><tr><td>001</td><td>1/4</td></tr><tr><td>010</td><td>1/12</td></tr><tr><td>011</td><td>1/24</td></tr><tr><td>100</td><td>1/32</td></tr><tr><td>101</td><td>1/64</td></tr><tr><td>110</td><td>1/128</td></tr><tr><td>111</td><td>1/256</td></tr></table>	T0DIV[2:0]	分频比	000	1/1	001	1/4	010	1/12	011	1/24	100	1/32	101	1/64	110	1/128	111	1/256
T0DIV[2:0]	分频比																			
000	1/1																			
001	1/4																			
010	1/12																			
011	1/24																			
100	1/32																			
101	1/64																			
110	1/128																			
111	1/256																			
1	PWMCKS	PWM 时钟源选择 0 = PWM 时钟源为系统时钟 F_{sys} 1 = PWM 时钟源为定时器 1 的溢出																		
0	CLOEN	系统时钟输出使能 0 = 禁用系统时钟输出 1 = 使能系统时钟输出, 从 CLO (P1.1)输出																		

P2S - 定时器 0/1 输出使能

地址: AFH

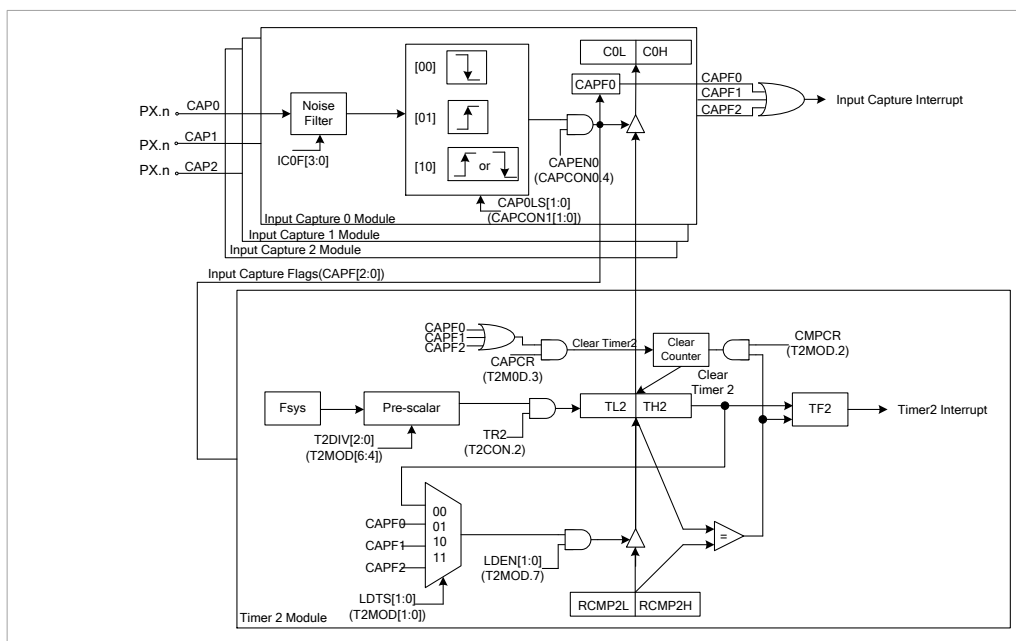
Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2S	-	-	-	-	T1OE	T0OE	-	-
R/W	-	-	-	-	R/W	R/W	-	-
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
3	T1OE	定时器 1 输出使能 0 = 定时器 1 从 T1 引脚输出关闭。 1 = 定时器 1 从 T1 引脚输出打开。
2	T0OE	定时器 0 输出使能 0 = 定时器 0 从 T0 引脚输出关闭 1 = 定时器 0 从 T0 引脚输出打开

13 定时器 2

定时器 2 是一个 16 位的向上计数器，由高 8 位寄存器(TH2)和低 8 位寄存器(TL2)组成。通过配置寄存器 RCMP2H 和 RCMP2L，设置 CM_RL2(T2CON.0)后，定时器 2 能工作在比较模式和自动重载模式下。定时器 2 具有 3 通道输入捕获模块，可用于测量输入脉冲宽度或周期。3 通道捕获结果分别存放在寄存器 C0H 及 C0L, C1H 及 C1L, C2H 及 C2L 中。定时器 2 的时钟来自系统时钟的分频，总共具有 8 级分频，可适用于更多应用需求。当 TR2(T2CON.2) 置 1，定时器使能；TR2 置 0 时，定时器关闭。

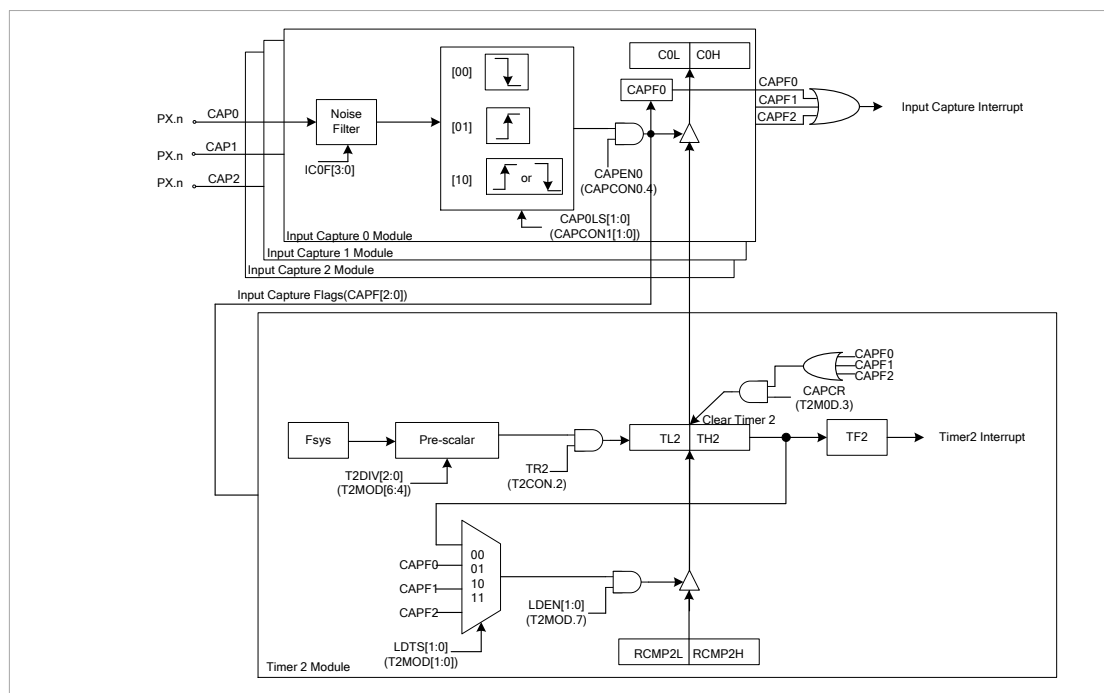
图 13-1 定时器 2 框图



13.1 16 位自动重载功能模式。

当 CM_RL2 清 0，定时器 2 配置为自动重载模式。在该模式下，RCMP2H 及 RCMP2L 保存重载的数值。当 LDEN 置位后，一旦有触发事件发生，硬件将自动把 RCMP2H 及 RCMP2L 寄存器内的值写入 TH2 及 TL2 中。触发事件可以是定时器 2 溢出或是一个所配置的捕获信道有触发事件发生（根据 LDTS[1:0] (T2MOD[1:0])配置）。注意，一旦 CAPCR (T2MOD.3) 置 1，如有一个捕获事件发生，仅清除 TH2 及 TL2 内的值，不会将 RCMP2H 及 RCMP2L 的值载入。

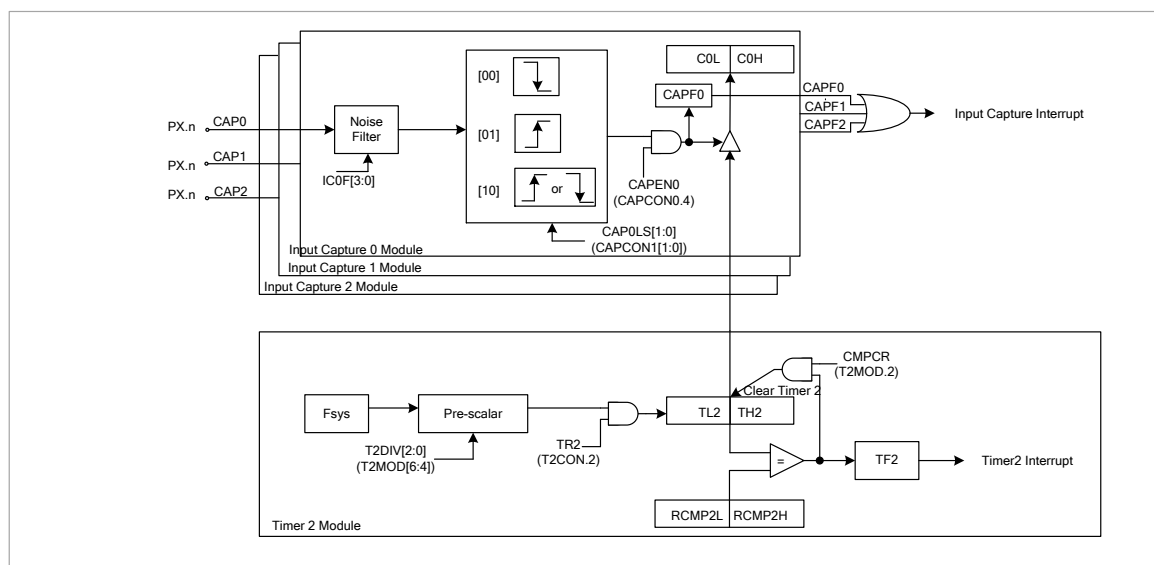
图 13-2 定时器 2 自动重载和捕获功能模块



13.2 比较功能模式:

当 CM_RL2 (T2CON.0) 置 1, 定时器 2 配置为比较器模式。TH2 和 TL2 匹配 RCMP2H 和 RCMP2L 的设定值时, TF2 置 1 进中断。在计数匹配后, 如果 CMPCR 位置位, 则清零 TH2 和 TL2, 计数器继续从 0000H 开始计数。然后再一次进行匹配。如果 CMPCR 位未置位, 则计数器会继续计数到 FFFFH 后清零, 然后继续计数。该功能作为周期性比较计数功能提供给用户一种比较方便直观的应用。

图 13-3 定时器 2 比较功能



13.3 输入捕获功能模式:

输入捕获模块依靠定时器 2 实现输入捕获功能。输入捕获模块通过寄存器 CAPCON0~2 配置来支持 3 组信道输入(CAP0, CAP1 和 CAP2), 可选择配置任何引脚。每个输入通道的噪声滤波器可通过 ICx F[3:0]配置, 可滤除小于通过设置的采样时钟周期的输入毛刺。每组输入捕获通道共享定时器 2 计数, 但有自己独立的边沿检测。每个触发边沿检测可由寄存器 CAPCON1 的相关位独立配置, 支持正边沿捕获, 负边沿捕获, 或双边沿捕获。

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

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

13.4 定时器 2 寄存器

T2CON - 定时器 2 控制寄存器

地址: C8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	TF2	-	-	-	-	TR2	-	CM_RL2
R/W	R/W	-	-	-	-	R/W	-	R/W
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7	TF2	定时器 2 溢出标志 当定时器 2 溢出或者比较数据符合, 该位置 1。如果已使能定时器 2 中断, 该位置 1 会让芯片进入定时器 2 中断运行。该位硬件置 1, 但需要软件清 0。
2	TR2	定时器 2 运行控制 0 = 定时器计数关闭。清除该位关闭定时器计数, 并清除计数器值 1 = 定时器 2 计数使能
0	CM_RL2	定时器 2 比较及自动重装载功能选择 0 = 自动重装载模式 1 = 比较器模式

T2MOD - 定时器 2 模式选择寄存器

地址：C9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2MOD	LDEN	T2DIV[2:0]			CAPCR	CMPCR	LDTS[1:0]	
R/W	R/W	R/W			R/W	R/W	R/W	
复位值	0	0	0	0	0	0	0	0

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

RCMP2L - 定时器 2 装载/比较数据低字节

地址：CAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCMP2L	RCMP2L [7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	RCMP2L[7:0]	定时器 2 重载/比较器低字节 当定时器 2 设定为比较模式，预存放低字节待比较数据。 当设定为自动重载模式，预存放低字节数据。

RCMP2H - 定时器 2 装载/比较数据高字节

地址：CBH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCMP2H	RCMP2H [7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	RCMP2H[7:0]	定时器 2 重载/比较器高字节 当定时器 2 设定为比较模式，预存放高字节待比较数据。 当设定为自动重载模式，预存放高字节数据。

TL2 - 定时器 2 计数值低字节

地址：CCH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL2	TL2 [7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	TL2[7:0]	定时器 2 低字节数据 该寄存器存放 16 位定时器 2 实际计数的低 8 位字节数据。

TH2 - 定时器 2 计数值高字节

地址：CDH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH2	TH2 [7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	TH2[7:0]	定时器 2 高字节数据 该寄存器存放 16 位定时器 2 实际计数的高 8 位字节数据。

C0L - 捕获通道 0 低字节

地址：F1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C0L	C0L[7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

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

C0H - 捕获通道 0 高字节

地址：F2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C0H	C0H[7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

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

C1L - 捕获通道 1 低字节

地址：F3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C1L	C1L[7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

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

C1H - 捕获通道 1 高字节

地址：F4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C1H	C1H[7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

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

C2L - 捕获通道 2 低字节

地址：F5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C2L	C2L[7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

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

C2H -捕获通道 2 高字节

地址：F6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C2H	C2H[7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

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

CAPCON0 - 输入捕获控制寄存器 0

地址：A3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CAPCON0	-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0
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

位	名称	描述
6	CAPEN2	输入捕获通道 2 使能位 0 = 关闭输入捕获通道 2 1 = 打开输入捕获通道 2
5	CAPEN1	输入捕获通道 1 使能位 0 = 关闭输入捕获通道 1 1 = 打开输入捕获通道 1
4	CAPEN0	输入捕获通道 0 使能位 0 = 关闭输入捕获通道 0 1 = 打开输入捕获通道 0
2	CAPF2	输入捕获通道 2 标志位 如果输入捕获通道 2 检测到边沿触发事件发生，该位由硬件置位，由软件清零。
1	CAPF1	输入捕获通道 1 标志位 如果输入捕获通道 1 检测到边沿触发事件发生，该位由硬件置位，由软件清零。
0	CAPF0	输入捕获通道 0 标志位 如果输入捕获通道 0 检测到边沿触发事件发生，该位由硬件置位，由软件清零。

CAPCON1 - 输入捕获控制寄存器 1

地址: A4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CAPCON1	-	-	CAP2LS[1:0]		CAP1LS[1:0]		CAP0LS[1:0]	
R/W	-	-	R/W		R/W		R/W	
复位值	0	0	0	0	0	0	0	0

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

CAPCON2 - 输入捕获控制寄存器 2

地址：A5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CAPCON2	IC1F[3:0]				IC0F[3:0]			
R/W	R/W				R/W			
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:4	IC1F[3:0]	输入捕获通道 1 滤波设置 0000: 关闭滤波功能, noise filter 输出波形与输入一致; 0001: Fsample=Fsys, 滤除小于 2 个采样时钟周期的噪声; 0010: Fsample=Fsys, 滤除小于 4 个采样时钟周期的噪声; 0011: Fsample=Fsys, 滤除小于 8 个采样时钟周期的噪声; 0100: Fsample=Fsys/2, 滤除小于 6 个采样时钟周期的噪声; 0101: Fsample=Fsys/2, 滤除小于 8 个采样时钟周期的噪声; 0110: Fsample=Fsys/4, 滤除小于 6 个采样时钟周期的噪声; 0111: Fsample=Fsys/4, 滤除小于 8 个采样时钟周期的噪声; 1000: Fsample=Fsys/8, 滤除小于 6 个采样时钟周期的噪声; 1001: Fsample=Fsys/8, 滤除小于 8 个采样时钟周期的噪声; 1010: Fsample=Fsys/16, 滤除小于 5 个采样时钟周期的噪声; 1011: Fsample=Fsys/16, 滤除小于 6 个采样时钟周期的噪声; 1100: Fsample=Fsys/16, 滤除小于 8 个采样时钟周期的噪声; 1101: Fsample=Fsys/32, 滤除小于 5 个采样时钟周期的噪声; 1110: Fsample=Fsys/32, 滤除小于 6 个采样时钟周期的噪声; 1111: Fsample=Fsys/32, 滤除小于 8 个采样时钟周期的噪声;
3:0	IC0F[3:0]	输入捕获通道 0 滤波设置 具体设置同通道 1

CAPCON3 - 输入捕获控制寄存器 3

地址: A7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CAPCON3	IC2F[3:0]				-	-	-	-
R/W	R/W				-	-	-	-
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:4	IC2F[3:0]	输入捕获通道 2 滤波设置 0000: 关闭滤波功能, noise filter 输出波形与输入一致; 0001: Fsample=Fsys, 滤除小于 2 个采样时钟周期的噪声; 0010: Fsample=Fsys, 滤除小于 4 个采样时钟周期的噪声; 0011: Fsample=Fsys, 滤除小于 8 个采样时钟周期的噪声; 0100: Fsample=Fsys/2, 滤除小于 6 个采样时钟周期的噪声; 0101: Fsample=Fsys/2, 滤除小于 8 个采样时钟周期的噪声; 0110: Fsample=Fsys/4, 滤除小于 6 个采样时钟周期的噪声; 0111: Fsample=Fsys/4, 滤除小于 8 个采样时钟周期的噪声; 1000: Fsample=Fsys/8, 滤除小于 6 个采样时钟周期的噪声; 1001: Fsample=Fsys/8, 滤除小于 8 个采样时钟周期的噪声; 1010: Fsample=Fsys/16, 滤除小于 5 个采样时钟周期的噪声; 1011: Fsample=Fsys/16, 滤除小于 6 个采样时钟周期的噪声; 1100: Fsample=Fsys/16, 滤除小于 8 个采样时钟周期的噪声; 1101: Fsample=Fsys/32, 滤除小于 5 个采样时钟周期的噪声; 1110: Fsample=Fsys/32, 滤除小于 6 个采样时钟周期的噪声; 1111: Fsample=Fsys/32, 滤除小于 8 个采样时钟周期的噪声;

14 定时器 3/4

14.1 概述

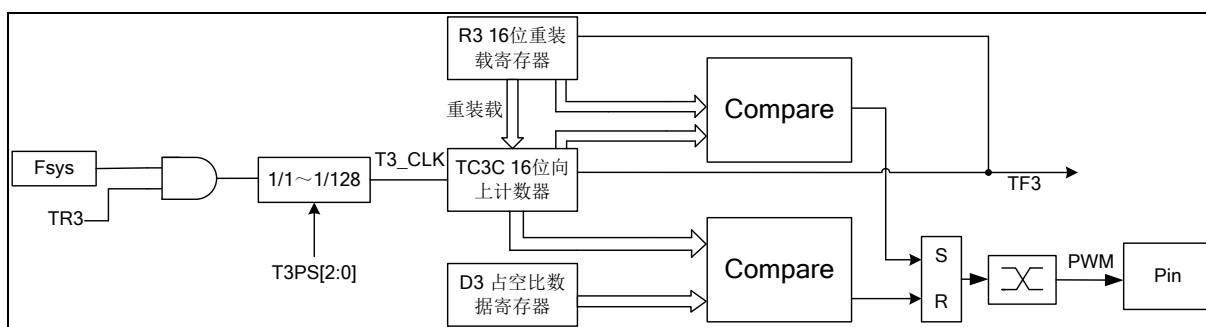
定时器 3 和 4 是两个运行模式相似的 16 位定时器，定时器 3 可以通过配置输出 PWM。定时器 4 是一个 16 位向上计数器。同时定时器 3/4 可以级联成为一个 32 位可自动重装载定时器。

14.2 定时器 3

定时器 3 是一个 16 位自动重装载，向上计数定时器。用户可以通过配置 T3PS[2:0] (T3CON[2:0]) 选择预分频，并通过写入重载值到 RH3 和 RL3 寄存器来决定它的溢出速率。用户可以设置 TR3 (T3CON.3) 来开始计数。当计数跨过 FFFFH，TF3 (T3CON.4) 置为 1，且 RH3 和 RL3 寄存器的内容重载到内部 16 位计数器。如果 ET3 (EIE1.1) 置为 1，定时器 3 中断服务程序被执行。当进入中断服务程序，TF3 需要软件进行清零

另外定时器 3 内含了一个 16 位的 PWM 可以配置周期和占空比，当用户需要使用 PWM 输出时，设置 PWME(T3CON.7) 为 1，使能定时器 3 的 PWM 功能，配置 T3DL 和 T3DH 寄存器设置 PWM 占空比，设置 T3PNP(T3CON.6) 设置 PWM 输出极性，同时配置相对应的 GPIO 端口，这时 PWM 波形通过芯片的管脚进行输出。

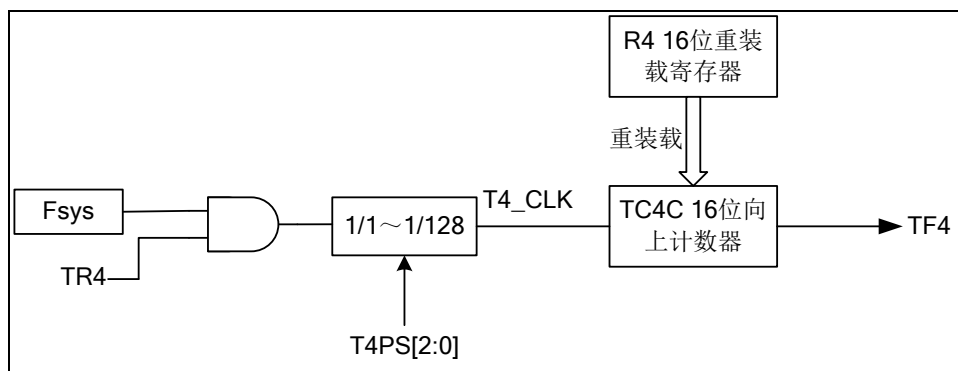
图 14-1 定时器 3 功能框图



14.3 定时器 4

定时器 4 是一个 16 位自动重装载, 向上计数定时器。用户可以通过配置 T4PS[2:0] (T4CON[2:0]) 选择预分频, 并写入重载值到 RH4 和 RL4 寄存器来决定它的溢出速率。用户可以设置 TR4 (T4CON.3) 来开始计数。当计数跨过 FFFFH, TF4 (T4CON.4) 置为 1, 且 RH4 和 RL4 寄存器的内容重载到内部 16 位计数器。如果 ET4 (EIE1.3) 置为 1, 定时器 4 中断服务程序被执行。TF4 必须软件清零。

图 14-2 定时器 4 功能框图



14.4 32 位定时器使用

JSC68F016 可以为用户提供一个 32 位的定时器, 通过使能 TMOD4 (T4CON[5] 位), TIM3, TIM4 级联成为一个定时器。这时 TIM3/TIM4 的单独定时功能失效。通过配置预分频器和设置自动重载寄存器实现定时时长和精细度的任意调整。定时器 3 和 4 用户可以通过如下步骤使用 32 位定时器:

- ◆ 设置 TMOD4 (T4CON.5) 为 1, 这时系统会将 TIM3 和 TIM4 级联变成一个 32 位的计数器
- ◆ 计数器时钟由 TIM4 的 T4PS[2:0] (T4CON[2:0]) 分频得到
- ◆ 计数溢出中断和标志位全部由 TIM4 的标志位进行表达和操作。

14.5 定时器 3/4 寄存器

T3CON - 定时器 3 控制寄存器

地址: C4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3CON	PWME	T3PNP	-	TF3	TR3	T3PS[2:0]		
R/W	R/W	R/W	-	R/W	R/W	R/W		
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7	PWME	定时器 3 PWM 功能使能 0 = 定时器 3 的 PWM 功能停止 1 = 定时器 3 的 PWM 功能使能
6	T3PNP	定时器 3 PWM 输出极性控制 0 = 输出正极性 1 = 输出负极性
4	TF3	定时器 3 溢出标志 当定时器 3 溢出, 该位置位。该位需要通过软件清零。
3	TR3	定时器 3 运行控制 0 = 定时器 3 停止 1 = 定时器 3 开始计数 注意重载寄存器 R3H 和 R3L 仅在定时器 3 停止(TR3 为 0)的时候才可以被写。如果 TR3 位为 1 写入 RH3 或 RL3, 结果是不可预知的。
2:0	T3PS[2:0]	定时器 3 预分频 这些位决定定时器 3 的时钟分频。 000 = 1/1 001 = 1/2 010 = 1/4 011 = 1/8 100 = 1/16 101 = 1/32 110 = 1/64 111 = 1/128

T3L - 定时器 3 计数器低字节寄存器

地址: D6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3L	T3L [7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	T3L [7:0]	定时器 3 计数器低字节寄存器

T3H - 定时器 3 计数器高字节寄存器

地址：D7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3H	T3H [7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	T3H [7:0]	定时器 3 计数器高字节寄存器

RL3 - 定时器 3 自动重载低字节

地址：C5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RL3	RL3 [7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	RL3 [7:0]	定时器 3 重载低字节 它保持着定时器 3 重载值的低字节

RH3 - 定时器 3 自动重载高字节

地址：C6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RH3	RH3 [7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	RH3 [7:0]	定时器 3 重载高字节 它保持着定时器 3 重载值的高字节

T3DL - 定时器 3 PWM 占空比低字节

地址：CEH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3DL	T3DL [7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	T3DL [7:0]	定时器 3 PWM 功能计数寄存器低位

T3DH - 定时器 3 PWM 占空比高字节

地址: CFH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3DH	T3DH [7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	T3DH [7:0]	定时器 3 PWM 功能计数寄存器高位

T4CON - 定时器 4 控制寄存器

XSFR 地址: F0C0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T4CON	-	-	TMOD4	TF4	TR4	T4PS[2:0]		
R/W	-	-	R/W	R/W	R/W	R/W		
复位值	0	0	0	0	0	0	0	0

位	名称	描述
5	TMOD4	定时器级联模式 0 = 定时器 4 单独计数 1 = 定时器 4 跟定时器 3 一起级联, 组成一个 32 位的计数器一起计数
4	TF4	定时器 4 溢出标志 当定时器 4 溢出, 该位置位。该位需要通过软件清零
3	TR4	定时器 4 运行控制 0 = 定时器 4 停止 1 = 定时器 4 开始计数 注意重载寄存器 R4H 和 R4L 仅在定时器 4 停止(TR4 为 0)的时候才可以被写。如果 TR4 位为 1 写入 RH4 或 RL4, 结果是不可预知的。
2:0	T4PS[2:0]	定时器 4 预分频 这些位决定定时器 4 的时钟分频。 000 = 1/1 001 = 1/2 010 = 1/4 011 = 1/8 100 = 1/16 101 = 1/32 110 = 1/64 111 = 1/128

RCNTL4 - 定时器 4 计数器低字节寄存器

地址: XSFR 地址: F0C3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCNTL4	RCNTL4 [7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	RCNTL4 [7:0]	定时器 4 计数器低字节寄存器

RCNTH4-定时器 4 计数器高字节寄存器

地址: XSFR 地址: F0C4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCNTH4	RCNTH4 [7:0]							
R/W	R							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	RCNTH4 [7:0]	定时器 4 计数器高字节寄存器

RL4 - 定时器 4 自动重载低字节

XSFR 地址: F0C1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RL4	RL4 [7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	RL4 [7:0]	定时器 4 重载低字节 它保持着定时器 4 重载值的低字节

RH4 - 定时器 4 自动重载高字节

XSFR 地址: F0C2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RH4	RH4 [7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	RH4 [7:0]	定时器 4 重载高字节 它保持着定时器 4 重载值的高字节

15 增强型 PWM 模块

15.1 概述

增强型 PWM 模块包含四组（8 个 PWM 通道） 16 位精度、可调周期和占空比的 PWM 输出，非常适合用于电机控制。该模块架构适用于驱动单相或三相无刷直流电机(BLDC)，或三相交流感应电机。每个信道 PWM 输出可配置为独立输出模式、互补模式或同步模式。当设定为互补模式时，通过插入可配置的死区时间，保护 MOS 管同时导通。PWM 波形可配置边沿对齐或中心对齐来选择中断响应位置。

15.2 特性

增强型 PWM 模块具有如下特性：

- 8 路独立的 16 位 PWM 控制模式。

8 路独立输出：PG0、PG1、PG2、PG3、PG4、PG5、PG6、PG7

4 组互补 PWM 对：PG1/3/5/7 输出与 PG0/2/4/6 输出的信号互补

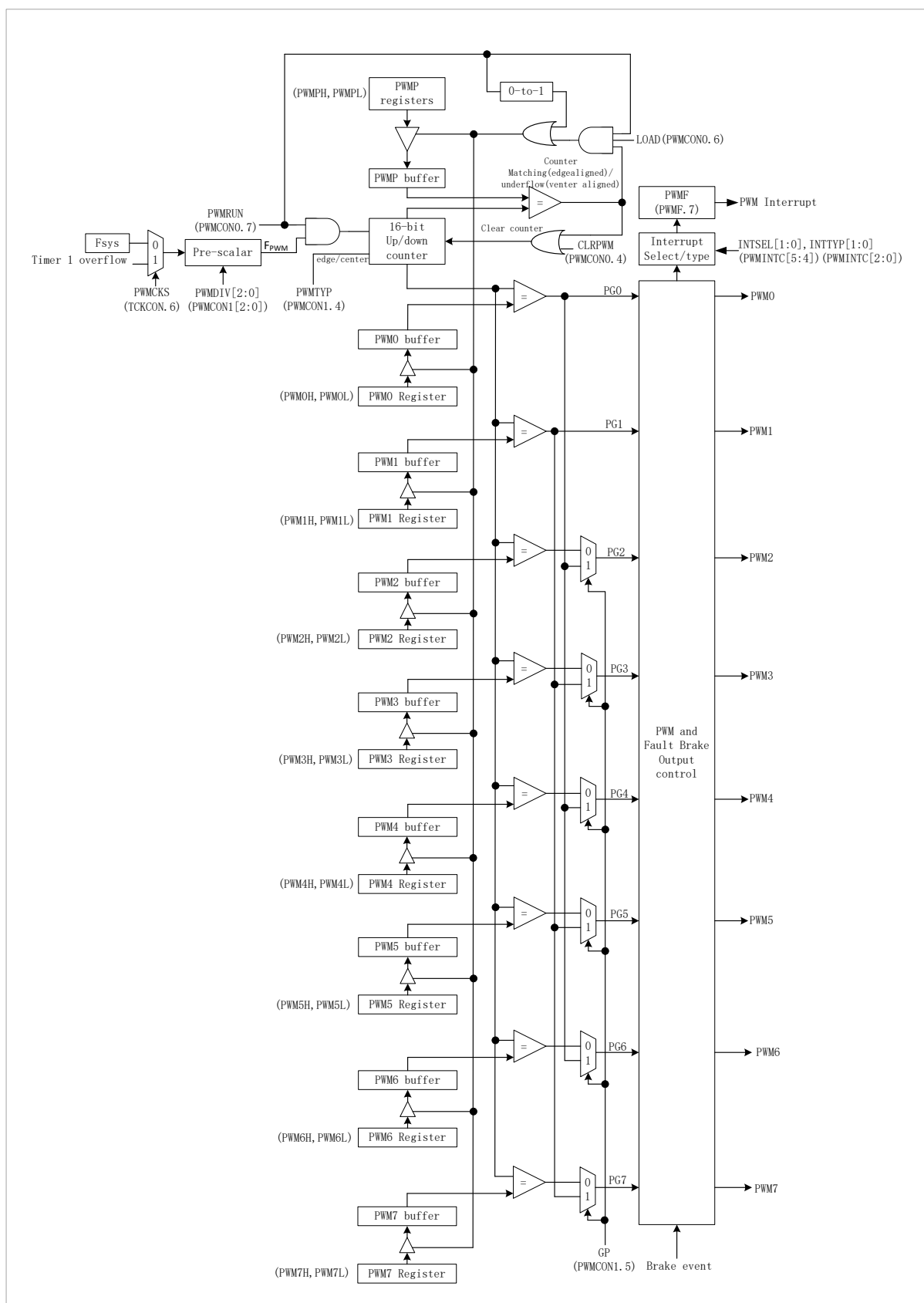
4 组同步 PWM 对：PG1/3/5/7 信号与 PG0/2/4/6 信号相同

- 支持组控制，PG1/3/5/7 信号可与 PG0/2/4/6 信号输出同步
- 支持边沿对齐和中心对齐模式
- 互补模式支持可编程死区发生器
- 支持 PWM 输出掩码控制
- 支持故障刹车功能（可由比较器，外部 FB 或者 ADC 比较事件触发）
- PWM 输出有独立的极性控制
- PWM 中断输出类型可配置

15.3 功能描述

增强型 PWM 模块的时钟有系统时钟或者定时器 1 计数溢出驱动，通过 PWM 时钟预分频器进行分频 1-1/128 后输入计数器。PWM 周期由 16 位周期寄存器 {PWMPH, PWMPL} 组合预先设置。所有 PWM 通道共享同一个 16 位周期计数器，周期是相同的。每个 PWM 通道都有独立占空比寄存器分别为 {PWM0H, PWM0L}、{PWM1H, PWM1L}、{PWM2H, PWM2L}、{PWM3H, PWM3L}、{PWM4H, PWM4L}、{PWM5H, PWM5L}、{PWM6H, PWM6L}、{PWM7H, PWM7L}有 8 个占空比寄存器，8 个通道都能产生独立占空比的 PWM 信号，PWM 信号的周期和占空比可通过 16 位计数器跟周期和占空比寄存器比较产生。

图 15-1 PWM 功能框图



脉冲宽度调制计数器产生 8 个 PWM 信号，称为 PG0、PG1、PG2、PG3、PG4、PG5、PG6 及 PG7。这些信号通过 PWM 及故障刹车输出控制电路，输出至 I/O 管脚。输出控制电路决定输出

PWM 信号的模式、死区时间、输出掩码、故障刹车以及 PWM 的极性。用户可以通过端口配置复用功能进行 PWM 输出配置。输出控制优先级 PNP>刹车>掩码>PWM 信号。

15.4 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-PG7 来标注，分别对应 PWM 通道 0-7。

PWM 通道的分配由相应的端口配置寄存器来控制，例如：

P10CFG = 0x12; //选择 P10 配置成 PG0 通道

P11CFG = 0x13; //选择 P11 配置成 PG1 通道

P12CFG = 0x14; //选择 P12 配置成 PG2 通道

P13CFG = 0x15; //选择 P13 配置成 PG3 通道

P14CFG = 0x16; //选择 P14 配置成 PG4 通道

P15CFG = 0x17; //选择 P15 配置成 PG5 通道

P16CFG = 0x1C; //选择 P16 配置成 PG6 通道

P17CFG = 0x1D; //选择 P17 配置成 PG7 通道

15.5 PWM 配置

- ◆ 配置相应 GPIO 为 PWM 输出端口
- ◆ 配置 PWM 驱动频率
- ◆ 配置周期寄存器产生需要的频率
- ◆ 配置 PWM 波的运行模式：独立、互补、同步
- ◆ 根据实际情况配置是否使用群组模式
- ◆ 配置各个 PWM 波的输出占空比寄存器，设定占空比值
- ◆ 清除 PWM 计数器
- ◆ 设置 PWMCON0.6 的 LOAD 位载入周期和占空比值
- ◆ 设置 PWMCON0.7 的 PWMRUN 位，启动 PWM 输出

所有的周期及占空比寄存器，具有硬件双缓存设计，因此{PWMPH, PWMPL}及{PWMnH, PWMnL}寄存器可随时被改写，但不会立即更新 PWM 周期和占空比，直到寄存器 LOAD (PWMCON0.6) 置 1 后和当前周期完成。

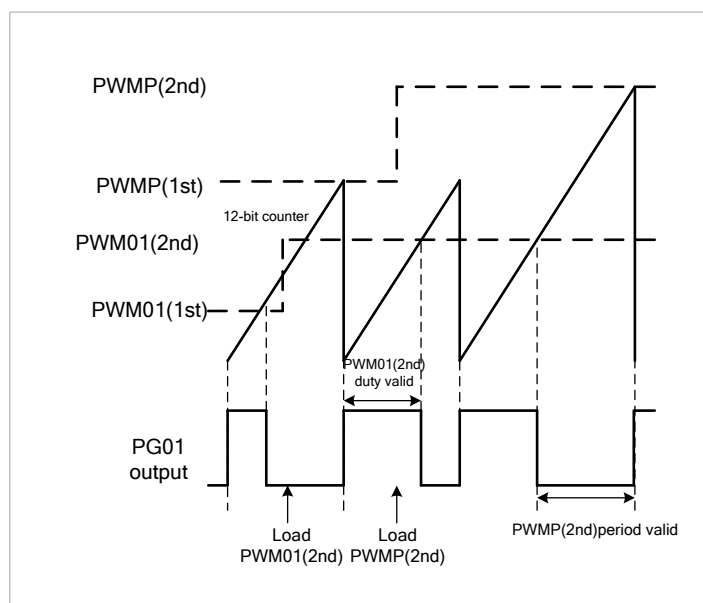
设置 LOAD 更新 PWM 周期及占空比寄存器值，通过监视 LOAD 位硬件自动清零确保完成，任何当 LOAD 还在保持 1 的时候，对周期或占空比寄存器内容的更改，可能引发无法预测的结果。

15.6 PWM 类型

15.6.1 边沿对齐模式

边沿对齐模式，16 位计数器设定为单周期模式，从 0000H 向上计数，直到与{PWMPH, PWMPH}匹配，然后重新从 0000H 开始开始向上计数。PWM 信号（PGn 信号在 PWM 和故障刹车输出控制模块之前）在 16 位计数器与周期寄存器和占空比寄存器{PWMnH, PWMnL}设置值匹配时停止 PGn 信号，并且设置 16 位计数器为 0000H。PWM 输出波形为左边沿对齐方式。

图 15-2 边沿对齐模式



PWM 边沿对齐方式输出频率及占空比算式如下：

$$\text{PWM 频率} = \frac{F_{\text{pwm}}}{\{\text{PWMPH}, \text{PWMPH}\} + 1}$$

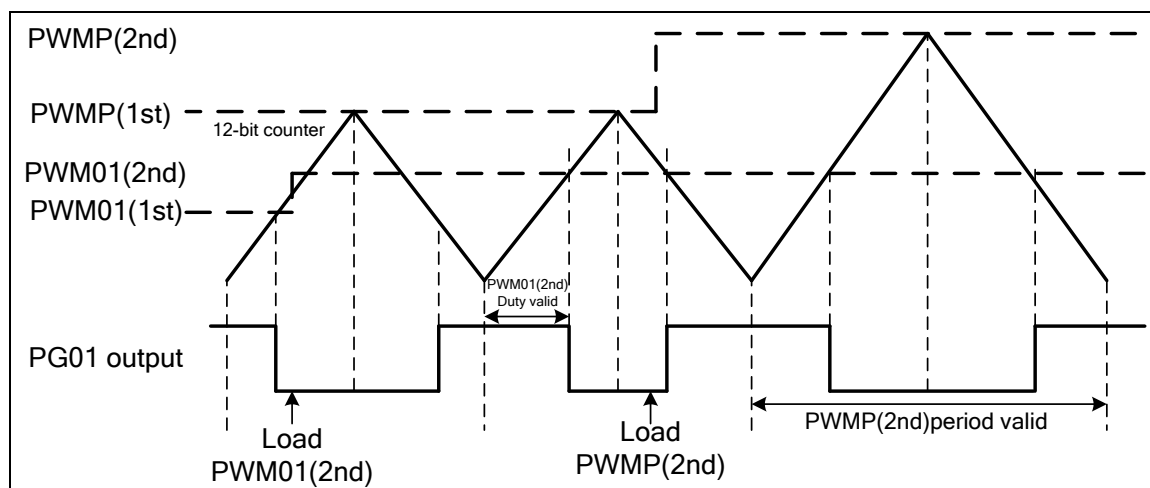
$$\text{PWM 占空比高电平} = \frac{\{\text{PWMnH}, \text{PWMnL}\}}{\{\text{PWMPH}, \text{PWMPH}\} + 1}$$

*注：当需要占空比为 100%的时候需要将{ PWMnH , PWMnHL}+1。可输出全部高电平

15.6.2 中心对齐模式

中心对齐模式，16 位计数器采用双周期模式，从 0000H 开始向上计数至{PWMPH, PWMPH}，然后由{PWMPH, PWMPH}向下计数至 0000H。PGn 信号在 16 位计数器向上计数，与周期寄存器和占空比寄存器{PWMnH, PWMnL}设置值匹配，然后 16 位计数器向下计数，与周期寄存器和占空比寄存器{PWMnH, PWMnL}设置值匹配，计数至 0000H 时停止 PGn 信号。中心对齐型 PWM 用于产生非重叠波形。

图 15-3 中心对齐模式



中心对齐模式输出频率及占空比算式如下：

$$\text{PWM 频率} = \frac{F_{\text{pwm}}}{2 * \{\text{PWMPH}, \text{PWML}\}} \quad (\text{FPWM 为时钟源除以 PWMDIV}).$$

$$\text{PWM 占空比高电平} = \frac{\{\text{PWMnH}, \text{PWMnL}\}}{\{\text{PWMPH}, \text{PWML}\}}$$

15.7 工作模式

PGn 信号通过 PWM 和故障刹车输出控制电路后，PWM 模式选择电路将产生不同类型的 PWM 输出模式，总共 8 个通道、四组，分别是 PG0~PG7。支持独立输出模式，互补模式及同步模式。

15.7.1 独立输出模式

当 PWMMOD[1:0] (PWMCON1[7:6]) 设定为 [0:0]，PWM 为独立输出模式。该模式为默认输出模式。PG0, PG1, PG2, PG3, PG4, PG5, PG6, PG7 独立输出 PWM 信号。

15.7.2 互补模式与死区延时

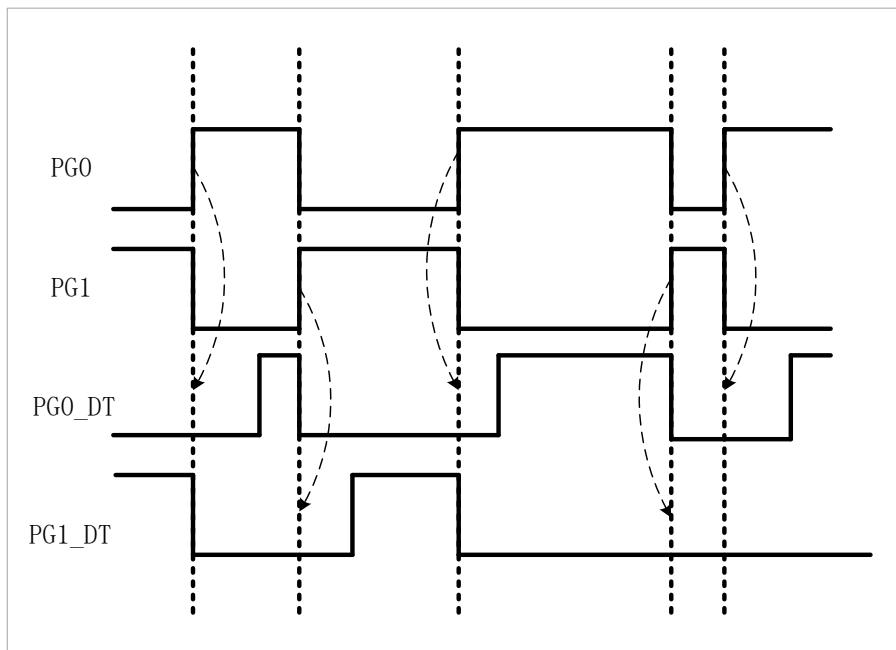
当 PWMMOD[1:0] = [0:1]，设定为互补模式。在该模式中 PG0/2/4/6 输出信号与独立模式下输出信号相同，但 PG1/3/5/7 输出与 PG0/2/4/6 输出的信号互补。同时忽略 PG1/3/5/7 占空比寄存器 {PWMnH, PWMnL} (n:1/3/5/7)。该模式可使 PG0/PG1 形成一对互补的 PWM 输出。同样 PG2/PG3, PG4/PG5, PG6/PG7 也可以用于互补输出。

在实际的电机应用中，互补模式 PWM 输出需要插入“死区时间”用来防止损坏电源开关器件，像用于控制半桥连续开关的 GPIBS，用来控制不能同时开关的电源器件。每组 PWM 共享一个 9 位“死区时间”计数器 PDT CNT，用于产生在同组中两通道之间关闭时间，对同组两个 PWM 信号中插入“关闭时间”，同样，在 PDT CNT 定时器溢出，电平从 0 到 1 转换的边沿会加入一段延迟。下图：互补

模式 PG0/PG1 组插入“死区时间”的时序图，PG2/PG3 组，PG4/PG5 组和 PG6/PG7 组也是同样的“死区时间”时序图。每组是否加入“死区时间”都可以通过 PDTCNT[3:0]寄存器来配置。

注：PDTCNT 及 PDTCNT 寄存器都有时控保护(TA)。仅当 PWM 配置为互补模式，“死区时间”控制才有效。

图 15-4 死区



15.7.3 同步模式

当 PWMMOD[1:0] = [1:0]，PWM 选择同步模式。在该模式下 PG0/2/4/6 信号输出与独立输出模式相同。PG1/3/5/7 信号与 PG0/2/4/6 信号也完全相同。

15.8 群组模式应用

通过设置 GP 位来使用组群应用，在群组应用跟独立、互补和同步模式相互独立，即在独立模式、互补模式和同步模式下同样可以使能群组模式。一旦使能则在互补模式，{PWM0H, PWM0L} 和 {PWM1H, PWM1L} 占空比寄存器决定所有 PWM 输出的占空比，{PWM2H, PWM2L}、{PWM3H, PWM3L}、{PWM4H, PWM4L}、{PWM5H, PWM5L}、{PWM6H, PWM6L}、{PWM7H, PWM7L} 寄存器失效。PWM2, PWM4, PWM6 的占空比跟 PWM0 占空比相同。PWM1, PWM3, PWM5, PWM7 的占空比跟 PWM1 的占空比相同。这意味着 PG0 输出与 PG2/4/6 相同，PG1 输出与 PG3/5/7 相同。如果使能互补模式，则 PG0 与 PG1 互补，意味着只需要设置 PG0 的输出占空比，所有其他输出占空比都全部确定。同步模式下，与互补输出相类似。

15.9 输出掩码控制

通过软件设置 PWM 的掩码寄存器可以屏蔽每个通道的 PWM 信号, PWM 掩码输出功能广泛应用于电子换向电机如直流无刷电机 BLDC。PMEN 寄存器包含 8 位掩码使能位, 每个掩码使能位掩码使能各自的 PWM 通道。PMEN 的默认值为 00H, 即所有 PWM 通道输出都不掩码。注: 掩码电位按照 PMD 设定值决定, 并受 PNP 寄存器影响。

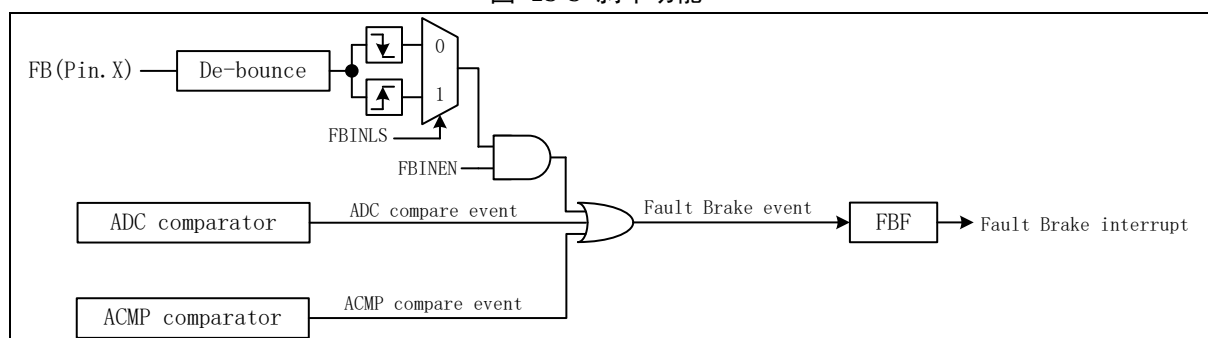
15.10 故障刹车

故障刹车功能应用在增强型 PWM 电路中, 配置为输入故障侦测, 保护电机系统防止损坏。有三种方式可以使 PWM 故障刹车。

- ◆ 当 FBINEN(PWMCON1.3)置 1, 故障刹车输入脚(FB)生效, 通过 FBINLS 设置触发方式, 通过 GPIO 选择相应的管脚进行外部刹车输入
- ◆ 当 ADC 比较器使能 ADCMPEN (ADCCON2.5), 每次 AD 转换结束都会进行比较。ADCMPO 输出的比较结果。当 ADFBEN (ADCCON2.7)设置后, ADC 比较结果可触发 PWM 故障刹车。
- ◆ 当比较器的 VCOUTCON 寄存器中使能 CFBEN 位, 并且通过 CFBL5 位选择触发方式, 输出结果符合设置后, 将进行 PWM 故障刹车

当故障发生, PWM 相对应的管脚的 FBD 值会被更改, PWMRUN (PWMCON0.7)位自动被硬件清除, PWM 输出停止, PWM 16 位计数器复位清 0, 标志位 FBF 通过硬件置 1, 如果中断使能了将会产生故障刹车中断。即便软件清除 FBF 值, FBD 数据内容仍然保持不变。用户需要重新设置 PWMRUN 的值来重新启动输出 PWM 信号。此时, 故障刹车状态才会被释放, PWM 按设定值正常输出。故障刹车具有极性选择位 FBINLS(PWMCON0.2)。注意, FB 管脚内部有固定的 24 个系统时钟周期响应过滤结构, FB 脚信号发生需要超过 24 个系统时钟故障刹车才会响应, 以避免管脚干扰信号引发误操作。

图 15-5 刹车功能



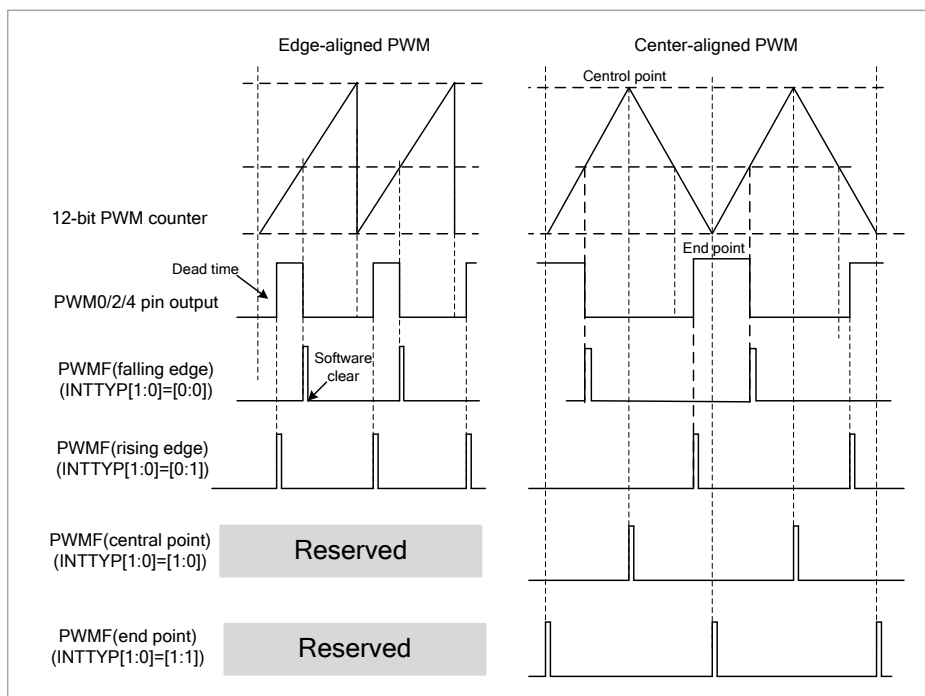
15.11 极性控制

每路 PWM 带有独立的极性控制位 PNP0~PNP7。默认正逻辑为高电平有效，即 PWM 输出高电平电源切换开，低电平电源切换关。用户可通过设置 PNP 位来改变 PWM 输出极性，产生相反的信号。

15.12 PWM 中断

PWM 模块带有标志位 PWMF (PWMF.7) 用来标志当前 PWM 周期完成状态。响应条件根据 INTSEL[2:0]和 INTTYP[1:0] (PWMINTC[2:0]和[5:4])设置。注：中心点触发或终点触发中断仅适用于中心对齐模式。PWMF 通过软件清零

图 15-6 PWM 中断方式



15.13 PWM 寄存器

PWMCON0 - PWM 控制寄存器 0

XSFR 地址: F090H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON0	PWMRUN	LOAD		CLRPWM		FBINLS -	-	-
R/W	R/W	R/W		R/W		R/W	-	-
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7	PWMRUN	PWM 运行使能位 0 = PWM 模块空闲 1 = PWM 开始运行
6	LOAD	PWM 载入新周期及占空比 如果一个新的周期和占空比需要更新，置 1 这位导入周期和和占空比寄存器所设新值。当前一个 PWM 周期输出结束，载入动作才会开始。更改的周期及占空比值将在下一个周期展现。当载入动作结束，硬件自动将 LOAD 位清 0。这一特性会使得 LOAD 位写入及读出值可能不一致，意义也不相同。 写： 0 = 不动作 1 = 当前一个 PWM 周期输出结束，载入动作才会开始 读： 0 = 导入新周期和占空比已经完成 1 = 导入新周期和占空比仍然没完成
4	CLRPWM	清除 PWM 计数器 该位置 1，会直接清 PWM 16 位计数器至 0000H。当清除计数器动作完成，硬件会自动将该位清 0。这特性会使得 CLRPWM 位写入和读出值不一致，意义也不相同。 写： 0 = 无动作 1 = 清 PWM 16 位计数器 读： 0 = PWM 16 位计数器已清零 1 = PWM 16 位计数器内还有数值，未清零
2	FBINLS	FB 管脚输入选择 0 = 下降沿 1 = 上升沿

PWMCON1 - PWM 控制寄存器 1

XSFR 地址: F091H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON1	PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		
R/W	R/W		R/W	R/W	R/W	R/W		
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:6	PWMMOD[1:0]	PWM 模式选择 00 = 独立输出模式 01 = 互补模式 10 = 同步模式 11 = 保留位
5	GP	群组模式使能位 该位使能 PWM 群组模式。一旦使能，四对 PWM 的占空比由 PWM01H 和 PWM01L 决定，原本配置的数据失效。 0 = 未组成群组模式。 1 = 群组模式使能
4	PWMTYP	PWM 类型选择 0 = 边沿对齐型 PWM 1 = 中心对齐型 PWM
3	FBINEN	FB 管脚输入使能位 0 = 禁止 1 = 通过 FB 管脚输入使能 PWM 故障刹车功能。一旦在 FB 管脚输入的边沿信号与 FBINLS 选择位相同。PWM0~7 输出 FBD 寄存器所设置的信号电平，PWMRUN (PWMCON0.7) 位硬件清除。当 PWMRUN 信号重置 1，PWM 信号重新输出。
2:0	PWMDIV[2:0]	PWM 时钟除频 该寄存器段用于配置 PWM 时钟频率预分频。 000 = 1/1 001 = 1/2 010 = 1/4 011 = 1/8 100 = 1/16 101 = 1/32 110 = 1/64 111 = 1/128

PWMINTC - PWM 中断控制寄存器

XSFR 地址: F092H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMINTC	-	-	INTTYP[1:0]		-	INTSEL[2:0]		
R/W	-	-	R/W		-	R/W		
复位值	0	0	0	0	0	0	0	0

位	名称	描述
5:4	INTTYP[1:0]	PWM 中断类型选择 00 = PWM0~7 管脚下降沿 01 = PWM0~7 管脚上升沿 10 = 每个 PWM 周期的中点 11 = 每个 PWM 周期的终点 注: 中心点中断方式或终点中断仅适用于 PWM 中心对齐模式。
2:0	INTSEL[2:0]	PWM 中断通道选择 该位段用以选择中断响应所对应的 PWM 通道的中断。 000 = PWM0 001 = PWM1 010 = PWM2 011 = PWM3 100 = PWM4 101 = PWM5 110 = PWM6 111 = PWM7 其他 = PWM0

TCKCON - 时钟控制寄存器

地址: ABH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCKCON	T1DIV[2:0]			T0DIV[2:0]			PWMCKS	CLOEN
R/W	R/W			R/W			R/W	R/W
复位值	0	0	0	0	0	0	0	0

位	名称	描述
1	PWMCKS	PWM 时钟源选择 0 = PWM 时钟源为系统时钟 F_{sys} 1 = PWM 时钟源为定时器 1 的溢出
0	CLOEN	系统时钟输出使能 0 = 禁用系统时钟输出 1 = 使能系统时钟输出, 从 CLO (P1.1)输出

PWMPL - PWM 周期寄存器低字节

XSFR 地址: F093H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPL	PWMPL[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWMP[7:0]	PWM 周期低字节 这字节和 PWMPH 确定 PWM 信号发生器的周期。

PWMPH - PWM 周期寄存器高字节

V XSFR 地址: F094H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPH	PWMPH[15:8]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWMPH[15:8]	PWM 周期低字节 这字节和 PWMPL 确定 PWM 信号发生器的周期。

PWM0L - PWM0 占空比低字节

XSFR 地址: F095H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0L	PWM0[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM0L[7:0]	PWM0 占空比低字节 这字节与 PWM0H 搭配确定 PG0 输出的占空比

PWM0H - PWM0 占空比高字节

XSFR 地址: F096H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0H	PWM0[15:8]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM0[15:8]	PWM0 占空比高字节 这字节与 PWM0L 搭配确定 PG0 输出的占空比

PWM1L - PWM1 占空比低字节

XSFR 地址: F097H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1L	PWM1[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM1[7:0]	PWM1 占空比低字节 这字节与 PWM1H 搭配确定 PG1 输出的占空比

PWM1H - PWM1 占空比高字节

XSFR 地址: F098H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1H	PWM1[15:8]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM1[15:8]	PWM1 占空比高字节 这字节与 PWM1L 搭配确定 PG1 输出的占空比

PWM2L - PWM2 占空比低字节

XSFR 地址: F099H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2L	PWM2[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM2[7:0]	PWM2 占空比低字节 这字节与 PWM2H 搭配确定 PG2 输出的占空比

PWM2H - PWM2 占空比高字节

XSFR 地址: F09AH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2H	PWM2[15:8]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM2[15:8]	PWM2 占空比高字节 这字节与 PWM2L 搭配确定 PG2 输出的占空比

PWM3L - PWM3 占空比低字节

XSFR 地址: F09BH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3L	PWM3[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM3[7:0]	PWM3 占空比低字节 这字节与 PWM3H 搭配确定 PG3 输出的占空比

PWM3H - PWM3 占空比高字节

XSFR 地址: F09CH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM3H	PWM3[15:8]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM3[15:8]	PWM3 占空比高字节 这字节与 PWM3L 搭配确定 PG3 输出的占空比

PWM4L - PWM4 占空比低字节

XSFR 地址: F09DH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4L	PWM4[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM4[7:0]	PWM4 占空比低字节 这字节与 PWM4H 搭配确定 PG4 输出的占空比

PWM4H - PWM4 占空比高字节

XSFR 地址: F09EH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM4H	PWM4[15:8]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM4[15:8]	PWM4 占空比高字节 这字节与 PWM4L 搭配确定 PG4 输出的占空比

PWM5L - PWM5 占空比低字节

XSFR 地址: F09FH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5L	PWM5[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM5[7:0]	PWM5 占空比低字节 这字节与 PWM5H 搭配确定 PG5 输出的占空比

PWM5H - PWM5 占空比高字节

XSFR 地址: F0A0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM5H	PWM5[15:8]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM5[15:8]	PWM5 占空比高字节 这字节与 PWM5L 搭配确定 PG5 输出的占空比

PWM6L - PWM6 占空比低字节

XSFR 地址: F0A1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM6L	PWM6[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM6[7:0]	PWM6 占空比低字节 这字节与 PWM6H 搭配确定 PG6 输出的占空比

PWM6H - PWM6 占空比高字节

XSFR 地址: F0A2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM6H	PWM6[15:8]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM6[15:8]	PWM6 占空比高字节 这字节与 PWM6L 搭配确定 PG6 输出的占空比

PWM7L - PWM7 占空比低字节

XSFR 地址: F0A3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM7L	PWM7[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM7[7:0]	PWM7 占空比低字节 这字节与 PWM7H 搭配确定 PG7 输出的占空比

PWM7H - PWM7 占空比高字节

XSFR 地址: F0A4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM7H	PWM7[15:8]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PWM7[15:8]	PWM7 占空比高字节 这字节与 PWM7L 搭配确定 PG7 输出的占空比

PDTEN - PWM 死区寄存器(TA 保护)

XSFR 地址: F0A5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PDTEN	-	-	-	PDTCNT.8	PDT67EN	PDT45EN	PDT23EN	PDT01EN
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

位	名称	描述
4	PDTCNT.8	PWM “死区时间”计数器位 8 详见 PDTCNT 寄存器.
3	PDT67EN	PWM6/7 组死区时间使能位 仅当 PWM6/7 配置为互补模式，死区功能才会生效。 0 = GP6/GP7 信号无延时 1 = 在 GP6/GP7 信号上升沿加入死区时间延时
2	PDT45EN	PWM4/5 组死区时间使能位 仅当 PWM4/5 配置为互补模式，死区功能才会生效。 0 = GP4/GP5 信号无延时 1 = 在 GP4/GP5 信号上升沿加入死区时间延时
1	PDT23EN	PWM2/3 组死区时间使能位 仅当 PWM2/3 配置为互补模式，死区功能才会生效。 0 = GP2/GP3 信号无延时 1 = 在 GP2/GP3 信号上升沿加入死区时间延时
0	PDT01EN	PWM0/1 组死区时间使能位 仅当 PWM0/1 配置为互补模式，死区功能才会生效。 0 = GP0/GP1 信号无延时 1 = 在 GP0/GP1 信号上升沿加入死区时间延时

PDTCNT - PWM 死区时间计数器(TA 保护)

XSFR 地址: F0A6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PDTCNT	PDTCNT[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	PDTCNT[7:0]	PWM 死区时间计数器低字节 该 8 位寄存器与 PD TEN.4 组成 9 位 PWM 死区时间计数器 PDT CNT。该计数器仅当 PWM 设定为互补模式，且有效死区使能位已设置时有效。 $\text{PWM 死区时间} = \frac{\text{PDT CNT} + 1}{\text{Fsys/Pre} - \text{scale}}$ 注：在 PWM 运行过程中，请勿更改 PDT CNT 的值

PMEN - PWM 掩码使能寄存器

XSFR 地址: F0A7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PMEN	PMEN7	PMEN6	PMEN5	PMEN4	PMEN3	PMEN2	PMEN1	PMEN0
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

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

PMD - PWM 掩码数据寄存器

XSFR 地址: F0A8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PMD	PMD7	PMD6	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0
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

位	名称	描述
n	PMDn	PWMn 掩码数据寄存器 PWMn 信号输出掩码寄存器设置的电平，当其相应的 PMENn 位设置时。 0 = PWMn 掩码，输出低电平 1 = PWMn 掩码，输出高电平

PNP - PWM 负极性寄存器

XSFR 地址: F0A9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PNP	PNP7	PNP6	PNP5	PNP4	PNP3	PNP2	PNP1	PNP0
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

位	名称	描述
n	PNPn	PWMn 负极性输出使能 0 = PWMn 输出按照设定值输出到 PWMn 管脚 1 = PWMn 输出按照设定值取反输出到 PWMn 管脚

FBD - PWM 故障刹车数据

XSFR 地址: F0AAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FBD	FBD7	FBD6	FBD5	FBD4	FBD3	FBD2	FBD1	FBD0
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

位	类型	描述
n	FBDn	PWMn 故障刹车数据 0 = 当故障发生时 PWMn 信号输出为 0 1 = 当故障发生时 PWMn 信号输出为 1

PWMF - PWM 故障刹车控制

地址: E7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMF	PWMF	FBF	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
复位值	0	0	0	0	0	0	0	0

位	类型	描述
7	PWMF	PWM 标志位 周期完成置位, 响应条件根据 INTSEL[2:0]和 INTTYP[1:0] (PWMINTC[2:0]和[5:4])设置。通过软件清除
6	FBF	故障刹车标志位 当 FBINEN 设置为 1, FB 管脚上检测到符合 FBINLS 设定的边沿信号后, 或者 ADFBEN 位置 1, ADC 比较结果为 1 后, 或者 ACMP 的 CFBN 位置 1, 比较器边沿触发输出 PWM 刹车后, 则 FBF 位置 1。该位需要通过软件清 0。当 FBF 清 0 后, 故障刹车仍然不会释放 PWM 输出, 需要重新输出, 通过设置 PWMRUN (PWMCON0.0)为 1, 重新启动 PWM 输出。

16 UART0,1 模块

16.1 概述

芯片内部包含两个 UART 模块：UART0 和 UART1, 他们具有完全相同的功能。

串行端口为全双工的，它可以同时发送和接收。接收模块有接收缓存，意味着在接收前一个数据被读取之前，串口可以同时接收第二个数据。接收和发送都是对 SBUF 进行操作访问。写入 SBUF 数据将直接传到发送寄存器。写入 SBUF 数据加载到发送寄存器，而读取 SBUF 是访问一个具有独立物理地址的接收寄存器。

UART 共有 4 种操作模式：一种同步，3 种异步的。同步模式发送 8 位数据，没有起始/停止位；其它模式则分别支持 8 位和 9 位数据的发送，包含起始/停止位。模式 2 和 3 具有多处理器通信的特殊功能，通过将 SnCON 寄存器中的 SM2n 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 SM2n=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 SM2n 位，并准备接收将要来的数据字节。未被寻址的从机使 SM2n 置 1 并忽略传入的数据。

16.2 UARTn 端口配置

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

P03CFG = 0x08 选择 P0.3 配置成 TXD0 通道；

P04CFG = 0x09 选择 P0.4 配置成 RXD0 通道；

P05CFG = 0x0A 选择 P0.5 配置成 TXD1 通道；

P06CFG = 0x0B 选择 P0.6 配置成 RXD1 通道；

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。

16.3 UARTn 波特率

UART 模式 0 的波特率是固定的，为 $F_{sys}/12$ ；模式 2 有 2 个波特率选项，由 PCON 寄存器的 SMODn 位选择控制，其选项分别为 $F_{sys}/32$ （SMODn=1）和 $F_{sys}/64$ （SMODn=0）。

UART 模式 1 和模式 3 的波特率由内部 BRT 产生或者定时器 1 的模式 2 产生或者定时器 3 产生。通过设置 SMODn 位可以将波特率增加 1 倍。

详细配置见下表:

BDn	BRCKn	波特率发生器
0	0	定时器 1 作为 UARTn 波特率发生器
	1	定时器 3 作为 UARTn 波特率发生器
1	X	内部 BRT 作为 UARTn 波特率发生器

在模式 1 和模式 3 中若选择 内部 BRT 作为波特率发生器, 则波特率由下面公式产生:

$$\text{Baud Rate} = 2^{\text{SMODn}} \times \frac{F_{\text{sys}}}{64 \times (1024 - \{SxRELH, SxRELL\})}$$

SMODn 为波特率是否加倍选择位由寄存器 PCON 设置, Fsys 为系统时钟, 内部 BRT 作为波特率发生器下 SxRELH/SxRELL 的值应设置为

$$\{SxRELH, SxRELL\} = 1024 - \frac{2^{\text{SMODn}} \times F_{\text{sys}}}{64 \times \text{BaudRate}}$$

在模式 1 和模式 3 中若选择定时器 1 的模式 2 或者选择定时器 3 作为波特率发生器, 则波特率由下面公式产生。

定时器 1 的模式 2 作为波特率发生器:

$$\text{Baud Rate} = \frac{2^{\text{SMODn}}}{32} \times \frac{F_{\text{sys}}}{\text{Prescale} \times (256 - TH1)}$$

Fsys 为系统时钟, Prescale 为定时器分频比, 定时器 1 在相应波特率下 TH1 的相应设置为:

$$TH1 = 256 - \frac{2^{\text{SMODn}} \times F_{\text{sys}}}{32 \times \text{Prescale} \times \text{BaudRate}}$$

定时器 3 作为波特率发生器:

$$\text{Baud Rate} = \frac{2^{\text{SMODn}}}{32} \times \frac{F_{\text{sys}}}{\text{Prescale} \times (65536 - \{RH3, RL3\})}$$

Fsys 为系统时钟, Prescale 为定时器分频比, 定时器 3 在相应波特率下{RH3,RL3}的相应设置为:

$$\{RH3, RL3\} = 65536 - \frac{2^{\text{SMODn}} \times F_{\text{sys}}}{32 \times \text{Prescale} \times \text{BaudRate}}$$

下表位使用定时器 3 和内部波特率发生器情况下部分波特率的相关信息：

SMOD = 0 , F_{sys}= 48MHz

BaudRate	TIMER3					内部 BRT			
	Prescale	RH3	RL3	实际波特率	误差	SxRELH	SxRELL	实际波特率	误差
4800	1	0xFE	0xC8	4808	-0.16%	0x03	0x64	4808	-0.16%
9600	1	0xFF	0x64	9615	-0.16%	0x03	0xB2	9615	-0.16%
19200	1	0xFF	0xB2	19231	-0.16%	0x03	0xD9	19231	-0.16%
38400	1	0xFF	0xD9	38462	-0.16%	0x03	0xEC	37500	2.34%
57600	1	0xFF	0xE6	57692	-0.16%	0x03	0xF3	57692	-0.16%
115200	1	0xFF	0xF3	115385	-0.16%	-	-	-	-

SMOD = 0 F_{sys}= 24MHz

BaudRate	TIMER3					内部 BRT			
	Prescale	RH3	RL3	实际波特率	误差	SxRELH	SxRELL	实际波特率	误差
4800	1	0xFF	0x64	4808	-0.16%	0x03	0xB2	4808	-0.16%
9600	1	0xFF	0xB2	9615	-0.16%	0x03	0xD9	9615	-0.16%
19200	1	0xFF	0xD9	19231	-0.16%	0x03	0xEC	18750	2.34%
38400	1	0xFF	0xEC	37500	2.34%	0x03	0xF6	37500	2.34%
57600	1	0xFF	0xF3	57692	-0.16%	-	-	-	-
115200	1	-	-	-	-	-	-	-	-

SMOD = 0 F_{sys}= 8MHz

BaudRate	TIMER3					内部 BRT			
	Prescale	RH3	RL3	实际波特率	误差	SxRELH	SxRELL	实际波特率	误差
4800	1	0xFF	0xCC	4808	-0.16%	0x03	0xE6	4808	-0.16%
9600	1	0xFF	0xE6	9615	-0.16%	0x03	0xF3	9615	-0.16%
19200	1	0xFF	0xF3	19231	-0.16%	-	-	-	-
38400	1	-	-	-	-	-	-	-	-
57600	1	-	-	-	-	-	-	-	-
115200	1	-	-	-	-	-	-	-	-

SMOD = 1 F_{sys} = 48MHz

BaudRate	TIMER3					内部 BRT			
	Prescale	RH3	RL3	实际波特率	误差	SxRELH	SxRELL	实际波特率	误差
4800	1	0xFD	0x8F	4800	0	0x02	0xC7	4792	0.16%
9600	1	0xFE	0xC8	9615	-0.16%	0x03	0x64	9615	-0.16%
19200	1	0xFF	0x64	19231	-0.16%	0x03	0xB2	19231	-0.16%
38400	1	0xFF	0xB2	38462	-0.16%	0x03	0xD9	38462	-0.16%
57600	1	0xFF	0xCC	57692	-0.16%	0x03	0xE6	57692	-0.16%
115200	1	0xFF	0xE6	115385	-0.16%	0x03	0xF3	115385	-0.16%

SMOD = 1 F_{sys} = 24MHz

BaudRate	TIMER3					内部 BRT			
	Prescale	RH3	RL3	实际波特率	误差	SxRELH	SxRELL	实际波特率	误差
4800	1	0xFE	0xC8	4808	-0.16%	0x03	0x64	4808	-0.16%
9600	1	0xFF	0x64	9615	-0.16%	0x03	0xB2	9615	-0.16%
19200	1	0xFF	0xB2	19231	-0.16%	0x03	0xD9	19231	-0.16%
38400	1	0xFF	0xD9	38462	-0.16%	0x03	0xEC	37500	2.34%
57600	1	0xFF	0xE6	57692	-0.16%	0x03	0xF3	57692	-0.16%
115200	1	0xFF	0xF3	115385	-0.16%	-	-	-	-

SMOD = 1 F_{sys} = 8MHz

BaudRate	TIMER3					内部 BRT			
	Prescale	RH3	RL3	实际波特率	误差	SxRELH	SxRELL	实际波特率	误差
4800	1	0xFF	0x98	4808	-0.16%	0x03	0xCC	4808	-0.16%
9600	1	0xFF	0xCC	9615	-0.16%	0x03	0xE6	9615	-0.16%
19200	1	0xFF	0xE6	19231	-0.16%	0x03	0xF3	19231	-0.16%
38400	1	0xFF	0xF3	38462	-0.16%	-	-	-	-
57600	1	0xFF	0xF7	55556	3.55%	-	-	-	-
115200	1	-	-	-	-	-	-	-	-

16.4 UARTn 模式 (n=0,1)

串口 0 和 1 都有四种工作模式,可通过寄存器 SnCON 的相关位进行配置,如下表:

表 16-1 串口模式描述

模式	SM0/SM01	SM1/SM11	描述	帧位数	波特率
0	0	0	同步	8	$F_{sys}/12$
1	0	1	异步	10	见章节 16.3 波特率
2	1	0	异步	11	当 SMODn=0 时: $F_{sys}/64$ 当 SMODn=1 时: $F_{sys}/32$
3	1	1	异步	11	见章节 16.3 波特率

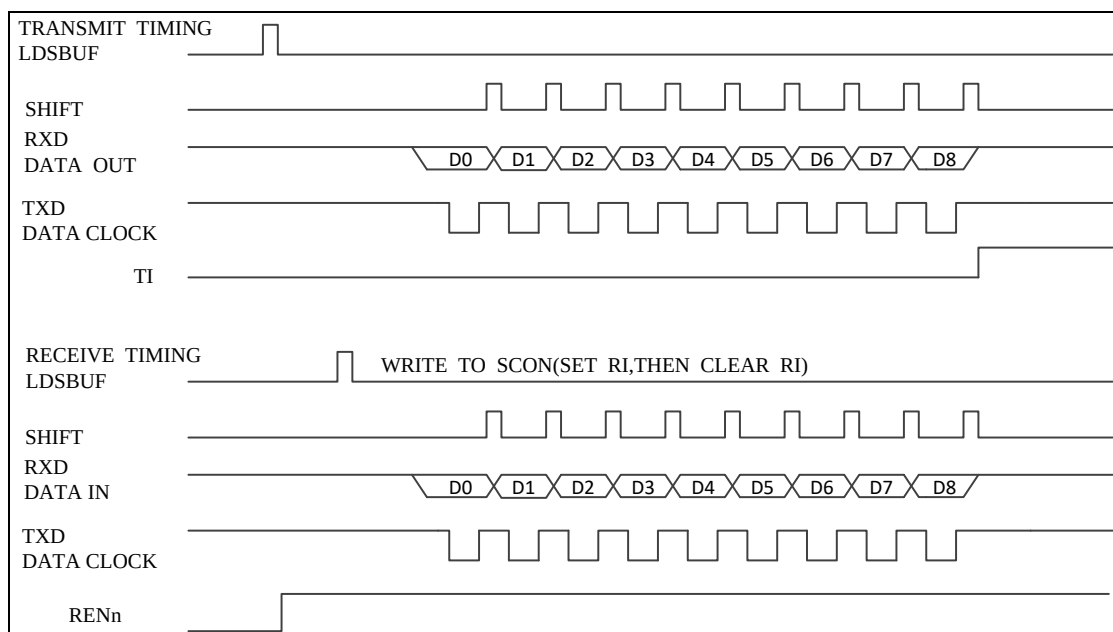
16.4.1 模式 0 同步 8 位收发器

模式 0 下, UART 发送/接收 8 位长度的数据, 没有起始/停止位。总线的首位为最低有效位 LSB, 波特率固定为 $F_{sys}/12$ 。接收数据时, 设置 REN0 位为 1, 先设置 RIn 位为 1, 然后清零 RIn 位开始接收, 接收 8 位数据后接收标志 RI0(或者 RI1)置 1; 发送时, 通过写入数据到 SxBUF 开始发送, 此时数据从 RXD 送出, 直到 8 位数据传输完成, 完成后 TI0(或者 TI1)标志位置 1。

模式 0 是与外部设备进行同步通信的方式。在该模式下, 串行数据由 RXD 脚进行收发, 而 TXD 脚用于产生移位时钟。这种方式下是以半双工的形式进行通信, 每帧接收或发送 8 位数据。数据的最低位被最先发送或接收。无论发送或接收数据, 串行时钟将一直由 MCU 产生, 因此串口模式 0 为主机模式。

下图是模式 0 时序图:

图 16-1 串口模式 0 时序图



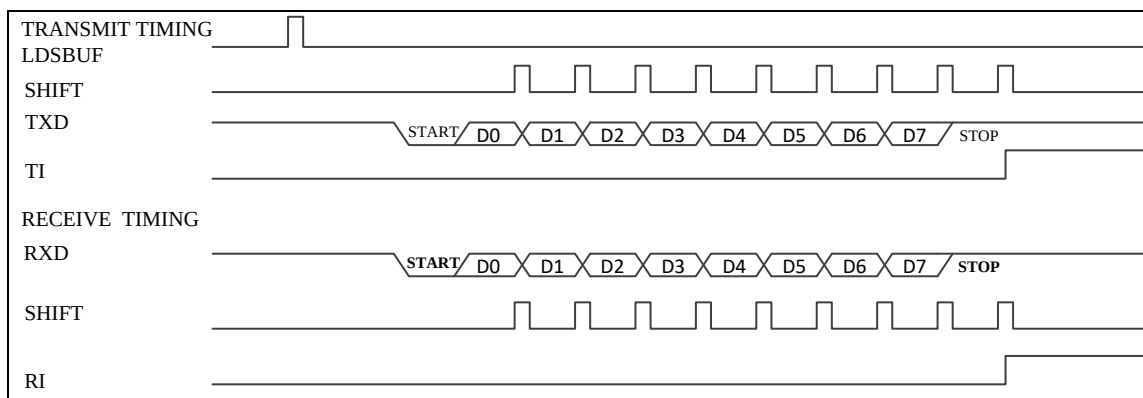
16.4.2 UART 模式 1: 异步 8 位收发器

模式 1 为异步全双工方式，UART 操作为典型的格式协议：包括起始位，8 位数据位和停止位。数据通过 TXD 送出，通过 RXD 接收。模式 1 的波特率可通过 SxRELH/SxRELL 寄存器溢出周期来控制，也可以通过定时器 1 的模式 2 或者定时器 3 的溢出周期来控制。

通过写入数据到 SxBUF 寄存器开始发送，发送的首位是起始位（始终为 0），然后是 SxBUF 的数据（首先 LSB），发送完停止位（始终为 1）后，将 TI0 位设置为 1。

若使能接收功能（REN0 或者 REN1=1），UART 从主机同步起始位。把数据位的第一位当成最低有效位（LSB），接收完成后，SxBUF 会进行更新。

图 16-2 串口模式 1 时序图



16.5 UART 寄存器

S0CON - 串口 0 控制寄存器

地址：98H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
R/W	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

地址：98H

位	名称	描述
7	SM0	串口 0 模式选择位（见 16.4 章节的表 16-1 串口模式描述）
6	SM1	
5	SM20	多机通信模式使能 当设置 SM20 位时，只有当第九个接受位（RB80）=1 时才会产生接收中断。否则，接收时不会产生中断。为了利用该特性进行多处理器通信，从处理器将其 SM20 位设置为 1。主处理器传输从处理器的地址时，将第 9 位设置为 1，可导致所有从处理器的接收中断。从处理器的软件将接收到的字节与它们的网络地址进行比较。如果有匹配，寻址从机清除它的 SM20 标志，其余的消息从主机发送，第 9 位设置为 0。 其他从服务器将它们的 SM20 设置为 1，此时忽略主服务器发送的其余消息。
4	REN0	串口 0 接收使能 0 = 关闭串口 0 接收功能 1 = 打开串口 0 在模式 1, 2 或 3 模式下的接收功能。
3	TB80	串口 0 第 9 位发送位 串口 0 在模式 2 和 3 中发送数据通过此位。它反映了第九位发送位的状态。 此位由软件控制。
2	RB80	串口 0 第 9 位接收位 串口 0 在模式 2 和 3 中接收数据通过此位。它反映了第九位接收位的状态。 模式 1 下，若 SM20=0，则此位是接收到的停止位。 模式 0 下该位无意义
1	TI0	串口 0 发送中断标志位 发送中断标志：模式 0 下该标志在发送完 8 位数据后由硬件置位，其它模式下在发送到数据最后一位后由硬件置位。该位必须由软件来清除。
0	RI0	串口 0 接收中断标志 该标志由硬件置位。在模式 0 中，接收到第 8 位；模式 1, 2 或 3 中接收到停止位。 该位必须由软件清除。

S0BUF - 串口 0 数据缓存寄存器

地址: 99H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0BUF	S0BUF[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	S0BUF[7:0]	串口 0 数据缓存寄存器 串口 0 接收或发送的数据都放在这个寄存器中。对它进行读操作将会接收串行数据, 对它进行写操作则发送串行数据。每次向 S0BUF 写入一字节数据, 将启动一次数据传输。

S0RELL - 串口 0 重装低字节寄存器

地址: AAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0RELL	S0REL[7:0]							
R/W	R/W							
复位值	1	1	0	1	1	0	0	1

位	名称	描述
7:0	S0REL[7:0]	串口 0 重装寄存器低字节 与 S0RELH 用于串口 0 波特率生成。 S0RELL 中的 8 位作为低位, S0RELH 中的 2 位 (S0RELH.1、S0RELH.0) 作为高位

S0RELH - 串口 0 重装高字节寄存器

地址: BAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0RELH	-	-	-	-	-	-	S0REL9	S0REL8
R/W	-	-	-	-	-	-	R/W	R/W
复位值	0	0	0	0	0	0	1	1

位	名称	描述
1:0	S0REL[9:8]	串口 0 重装寄存器高字节 与 S0RELL 用于串口 0 波特率生成。 S0RELL 中的 8 位作为低位, S0RELH 中的 2 位 (S0RELH.1、S0RELH.0) 作为高位。

SCON2 - 串口波特率发生器选择寄存器

地址：D8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON2	BD0	BRCK0	BD1	BRCK1	-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7	BD0	串口 0 波特率选择(模式 1 和 3) 0: 定时器 1 或者 3 作为波特率发生器 1: 使用内部波特率发生器
6	BRCK0	该位用于设置串口 0 在模式 1 或模式 3 下, 所使用的波特率发生器 0: 定时器 1 作为波特率发生器 1: 定时器 3 作为波特率发生器
5	BD1	串口 1 波特率选择(模式 1 和 3) 0: 定时器 1 或者 3 作为波特率发生器 1: 使用内部波特率发生器
4	BRCK1	该位用于设置串口 1 在模式 1 或模式 3 下, 所使用的波特率发生器 0: 定时器 1 作为波特率发生器 1: 定时器 3 作为波特率发生器

PCON - 串口波特率加倍选择寄存器

地址：87H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	SMOD1	POF	--	--	GF0	PD	IDL
R/W	R/W	R/W	R/W	--	--	R/W	R/W	R/W
复位值	0	0	1	--	--	0	0	0

位	名称	描述
7	SMOD0	串口 0 波特率加倍使能 串口 0 在模式 1、2、3 的波特率应用中, 如果设置该位, 波特率会加倍。 0: 波特率不加倍 1: 波特率加倍
6	SMOD1	串口 1 波特率加倍使能 串口 1 在模式 1、2、3 的波特率应用中, 如果设置该位, 波特率会加倍。 0: 波特率不加倍 1: 波特率加倍

S1CON - 串口 1 控制寄存器

地址：F8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1CON	SM01	SM11	SM21	REN1	TB81	RB81	TI1	RI1
R/W	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

位	名称	描述
7	SM01	串口 1 模式选择位（见 16.4 章节的表 16-1 串口模式描述）
6	SM11	
5	SM21	多机通信模式使能 当设置 SM21 位时，只有当第九个接受位（RB81）=1 时才会产生接收中断。否则，接收时不会产生中断。为了利用该特性进行多处理器通信，从处理器将其 SM21 位设置为 1。主处理器传输从处理器的地址时，将第 9 位设置为 1，可导致所有从处理器的接收中断。从处理器的软件将接收到的字节与它们的网络地址进行比较。如果有匹配，寻址从机清除它的 SM21 标志，其余的消息从主机发送，第 9 位设置为 0。 其他从服务器将它们 SM21 设置为 1，此时忽略主服务器发送的其余消息。
4	REN1	串口 1 接收使能 0 = 关闭串口 1 接收功能 1 = 打开串口 1 在模式 1，2 或 3 模式下的接收功能。
3	TB81	串口 1 第 9 位发送位 串口 1 在模式 2 和 3 中发送数据通过此位。它反映了第九位发送位的状态。此位由软件控制。
2	RB81	串口 1 第 9 位接收位 串口 1 在模式 2 和 3 中接收数据通过此位。它反映了第九位接收位的状态。 模式 1 下，若 SM21=0，则此位是接收到的停止位。 模式 0 下该位无意义
1	TI1	串口 1 发送中断标志位 发送中断标志：模式 0 下该标志在发送完 8 位数据后由硬件置位，其它模式下在发送到数据最后一位后由硬件置位。该位必须由软件来清除。
0	RI1	串口 1 接收中断标志 该标志由硬件置位。在模式 0 中，接收到第 8 位；模式 1，2 或 3 中接收到停止位。该位必须由软件清除。

S1BUF - 串口 1 数据缓存寄存器

地址：BBH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1BUF	S1BUF[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7: 0	S1BUF[7:0]	串口 1 数据缓存寄存器 串口 1 接收或发送的数据都放在这个寄存器中。对它进行读操作将会接收串行数据，对它进行写操作则发送串行数据。每次向 S1BUF 写入一字节数据，将启动一次数据传输。

S1RELL - 串口 1 重装低字节寄存器

地址：BCH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1RELL	S1REL[7:0]							
R/W	R/W							
复位值	1	1	0	1	1	0	0	1

位	名称	描述
7: 0	S1REL[7:0]	串口 1 重装寄存器低字节 与 S1RELH 用于串口 1 波特率生成。 S1RELL 中的 8 位作为低位，S1RELH 中的 2 位（S1RELH.1、S1RELH.0）作为高位

S1RELH - 串口 1 重装高字节寄存器

地址：BDH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1RELH	-	-	-	-	-	-	S1REL9	S1REL8
R/W	-	-	-	-	-	-	R/W	R/W
复位值	0	0	0	0	0	0	1	1

位	名称	描述
1:0	S1REL[9:8]	串口 1 重装寄存器高字节 与 S1RELL 用于串口 1 波特率生成。 S1RELL 中的 8 位作为低位，S1RELH 中的 2 位（S1RELH.1、S1RELH.0）作为高位。

17 I²C 模块

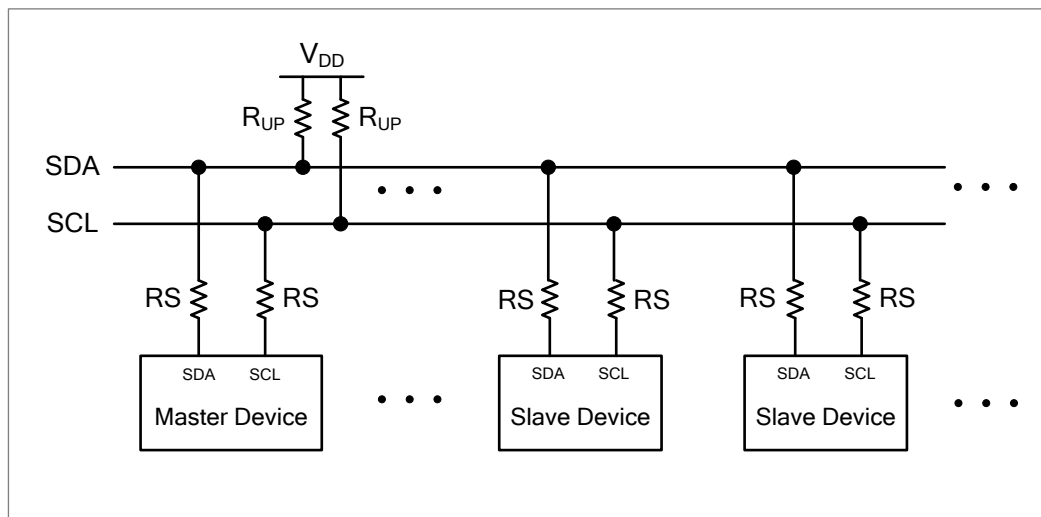
17.1 I²C 概述

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：主机发送、主机接收、从机发送、从机接收。
- ◆ 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。
- ◆ 从机方式支持 I²C 总线上的 7 位寻址模式。
- ◆ 中断产生。

图 17-1 I2C 连接框图



17.2 端口配置

若使用 I^2C 功能，则首先将相应端口配置成 SCL,SDA 通道（任意 GPIO 均可配置为 I^2C 通道）。

例如将 P0.1，P0.2 配置为 I^2C 功能：

P01CFG = 0x0C; //选择 P0.1 配置为 SCL 通道

P02CFG = 0x0D; //选择 P0.2 配置为 SDA 通道

GPIO 配置为 I^2C 通道后，该组端口默认为开漏状态，即输出数据为开漏输出。可通过 PXUP 配置是否使能 SCL，SDA 口内部的上拉电阻，或者在芯片外部增加上拉电阻。

在主机模式下， I^2C 输出 SCL 到从机，当发送完地址或数据后，从机需要拉低 SCL 并回送相应的应答信号给主机。主机需要回读 SCL 口线状态监测从机是否释放 SCL 以此判断是否需要进行下一帧数据发送。若 SCL 接的上拉电阻或寄生电容越大，会导致回读时间变长。从而影响 I^2C 的通信速度。

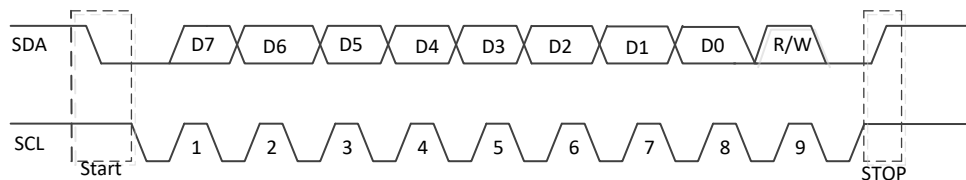
17.3 I^2C 总线传输方式

I^2C 发送结构包括 START(S)起始位，8 位地址字节，一个或多个数据字节，以及 STOP (P)停止位。起始位由主机产生，以便对传输进行初始化。

发送的数据为最高有效位（MSB）优先。在地址字节中，高 7 位为地址位，低位为数据方向位（R/W）R/W=0 时，表示该传输为写操作；R/W=1 时，表示该传输为读操作。

接收到每个字节后，接收器（主机或从机）必须发送一个 ACK 信号。若发送器没有接收到 ACK 信号，则会识别为 NACK 信号。在写操作中，主机发送数据给从机，然后等待从机返回 ACK 信号。在读操作中，从机发送数据给主机，然后等待主机返回 ACK 信号。最后，主机产生一个停止信号来结束数据传输

图 17-2 I^2C 传输数据框图

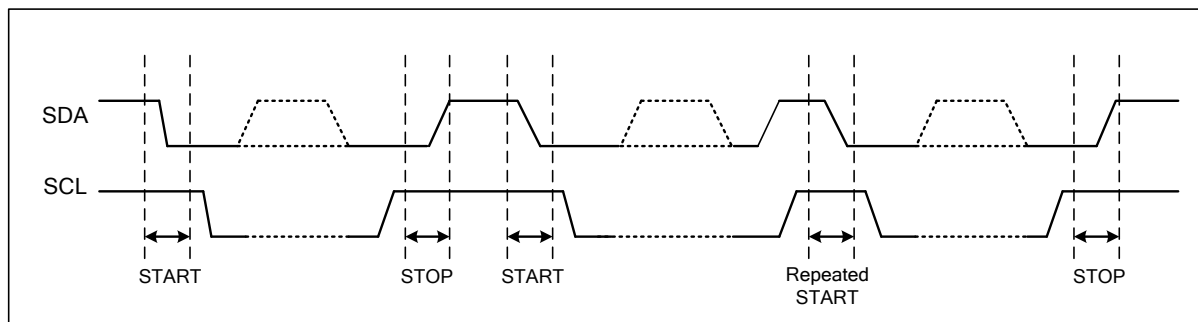


17.3.1 开始和停止条件

I^2C 总线时序定义了起始 START 和结束 STOP 的条件。时钟 SCL 位高时，数据线 SDA 由高电平变低电平的跳变被认为是 START 标志。时钟 SCL 位高时，数据线 SDA 由低电平变为高电平的跳变被认为是 STOP 标志。起始和结束由主机产生。

若主机发出停 STOP 止信号，传输停止。主机也可以不发停止位，直接再次发出起始 START 信号，则继续之前通信，或者换另一个地址通信。

图 17-3 I²C 起始和停止条件

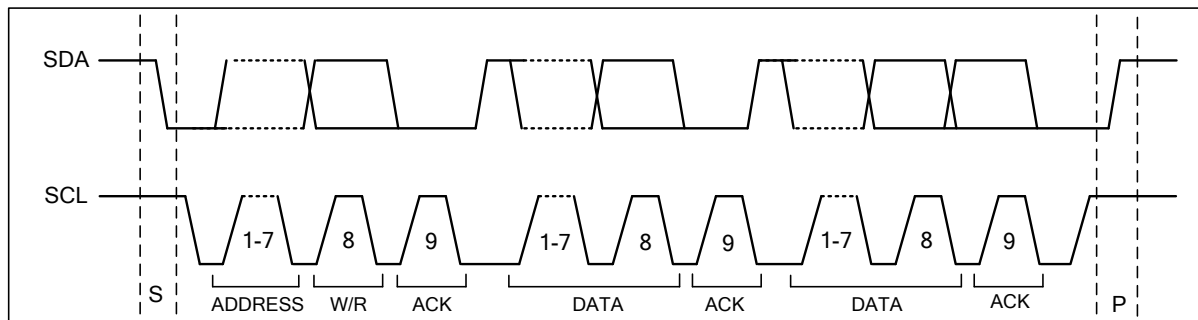


起始条件，重复起始条件和停止条件

17.3.2 地址数据格式

起始位 START 之后，第一个字节必须是 7 位地址 SLA+第 8 位读写方向位(R/W)，用以定义目标从机地址以及对从机写入或读出数据。若第 8 位是 0，即 SLA+W，表示下个字节开始主机向从机写数据；若是 1，即 SLA+R，就表示下字节开始，主机向从机读数据。所以，一个典型数据包含起始位 START，SLA+W/R，一个或多个字节数据，最后是停止位 STOP。当一个从机已被寻址及读写方向通过字节 SLA+W/R，随后的 8 位数据就跟随之前的设定进行传输。

图 17-4 I²C 传输格式



一个传输中的I2C数据格式

17.3.3 应答 ACK

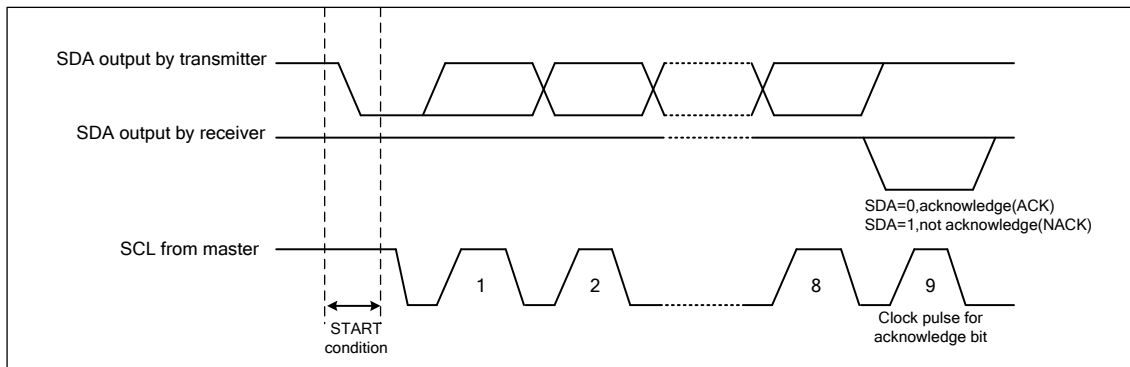
每字节开始传输后，SCL 的第 9 个脉冲用于传输应答位 (ACK)。主机或者从机通过将 SDA 拉低，来产生应答位 ACK。应答位时钟由主机产生。发送端设备在应答位时钟高电平周期内，需放弃对 SDA 的控制。在应答位时钟周期的高电平时，SDA 保持低电平用以表示接收端已成功接收到发送端的数据。通常被寻址的从机在整个传输过程中每字节都需要回复应答位。当该从机接收无法应答 (NACK) 从机地址时，从机将 SDA 线保持高以便主机产生停止(STOP)或发送重复开始(START)

信号。

若从机没有应答地址后，将自身切换到未寻址从机模式，从而无法接收更多数据字节。这时从机将 SDA 线拉高，此时主机应发送停止 STOP 信号或重复起始(repeated START)信号。

若是主机接收，主机控制着收发字节个数，主机在最后一个字节收发结束后不发应答位信号，从机端将切换为未寻址从机模式，并释放 SDA 线，以便主机直接发停止位 STOP 或重开始位 START。

图 17-5 I²C 应答



17.3.4 I²C 时钟时序

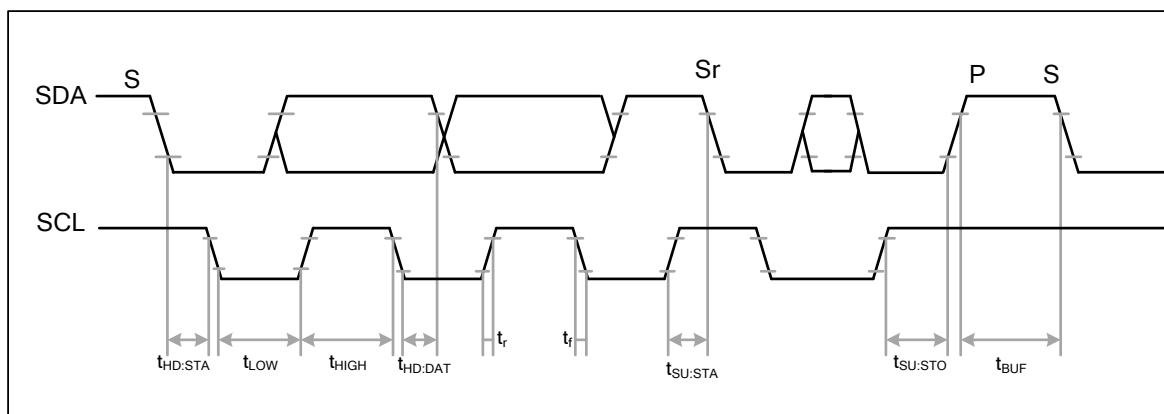
下图为 I²C 传输过程的最小应遵循的时序。

$t_{SU:STA}$ 和 t_{HIGH} 时序的计算假设 $t_r < \text{clk}$ 周期。如果 t_r 大于 clk 周期，则 I²C 内核检测到时钟延长。

clk 为系统时钟, SCL 为 I²C 的时钟周期

Time	最小值	Time	最小值
$t_{HD:STA}$	15clk	$t_{HD:DAT}$	7clk
t_{LOW}	30clk	$t_{SU:DAT}$	5clk
t_{HIGH}	30clk	$t_{SU:STO}$	23clk
$t_{SU:STA}$	15clk	t_{BUF}	55clk

图 17-6 I²C 时钟时序



I²C Clock Timings

17.4 I²C 主控模式

有四个寄存器用于与主机连接：操作控制、状态、数据接收和数据发送寄存器。

寄存器		地址
写入	读取	
主控模式控制寄存器 I2CCON	主控模式状态寄存器 I2CSTA	C0H、C3H
主控发送数据寄存器 I2CDAT	主控模式接收数据寄存器 I2CDAT	C2H

17.4.1 I²C 主控模式时钟速率

当 I²C 处于主模式时，这个可编程时钟脉冲发生器提供时钟脉冲。当 I²C 处于从模式时，时钟发生器被抑制。时钟生成器的功能由“I2CCON”寄存器的“CR0”、“CR1”和“CR2”位控制。下表列出了不同设置下的时钟速率：

CR2	CR1	CR0	I ² C Prescaler	比特率(Kbit/s)			
				8MHz	16MHz	24MHz	48MHz
0	0	0	256	31	63	94	187
0	0	1	224	36	71	107	214
0	1	0	192	42	83	125	250
0	1	1	160	50	100	150	300
1	0	0	960	6	17	25	50
1	0	1	120	67	133	200	400
1	1	0	60	133	267	400	--
1	1	1	(TC1 定时器溢出速率)/8				

17.4.2 I²C 主控模式控制与状态

I²C 主控模式主要有 4 个控制位：ENSI, STA, STO, AA 位，STA 位将产生 START 或 RESTART 条件。STO 位决定数据传输是否再周期结束时停止，或者继续。通过控制寄存器写入 STO=1, STA=1, ENSI=1，以执行相应的开始和停止。通过软件写入数据寄存器进行对从机的读写操作。通过设置 AA 位，主机将在收到数据时进行返回 ACK 或者 NACK。当操作完成，产生中断。可以从状态寄存器中读取相应的状态，并可在数据寄存器读取数据。

通过以下的组合列表可实现主控模式下各种操作：

STA：发送起始信号

STOP：发送结束信号

AA：使能应答

主控模式操作：（SLA：从机地址。W:写。R：读）

软件操作	操作解释	主控模式操作控位			软件操作后 I ² CSTA 寄存器值	I ² C 硬件下一步操作
		STA	STO	AA		
发送 START	发送开始或者重复开始	1	0	x	08H: 已发送 START 信号	发送 SLA+R;接收 ACK 发送 SLA+W
					10H: 已发送 Repeated START 信号	发送 SLA+R;接收 ACK;
						发送 SLA+W; I ² C 切换到主机接收模式
发送 STOP	发送停止	0	1	0		
发送 ACK	主机接收数据后返回 ACK	0	0	1	50H: 已接收数据字节; 并已回复 ACK	读取已经接收的数据
发送 NACK	主机接收数据后返回 NACK	0	0	0	58H: 已接收数据字节; 并已返回 NACK	读取已经接收的数据
发送 SLA+W	发送从机地址并进行写操作	x	0	x	18H: 已发送 SLA+W, 并已经接收到 ACK 20H: 已经发送 SLA+W 并已接收 NACK	发送数据; 接收 ACK;
						发送重复 START 开始信号;
						发送 STOP 结束信号, 复位 STO 标志位;
						发送 STOP 结束信号后接着再发送 START 开始信号, 复位 STO 标志位
发送 I ² CDAT 中的数据	发送数据	0	0	0	28H: 已经发送 I ² CDAT 寄存器中的自己, 并已接收 ACK 30H: 已经发送 I ² CDAT 寄存器中的自己, 并已接收 NACK	发送数据; 接收 ACK;
						发送重复 START 开始信号;
						发送 STOP 结束信号, 复位 STO 标志位;
						发送 STOP 结束信号后接着再发送 START 开始信号, 复位 STO 标志位
发送 SLA+R	发送从机地址并读操作	x	0	x	40H: 发送 SLA+R 并已经接收 ACK	接收数据字节; 返回 NACK 接收数据字节; 返回 ACK

					48H: 发送 SLA+R 并已经接收 NACK	发送重复 STATRXINH 发送 STOP 信号; 发送 STOP 信号, 接着发送 START 信号
--	--	--	--	--	--------------------------	--

17.5 I²C 从动模式

有五个寄存器用于与主机连接：操作控制、状态、从机地址、数据接收和数据发送寄存器。

寄存器		地址
写入	读取	
从动地址寄存器 I2CADR	从动地址寄存器 I2CADR	C1H
主控模式控制寄存器 I2CCON	主控模式状态寄存器 I2CSTA	C0H、C3H
主控发送数据寄存器 I2CDAT	主控模式接收数据寄存器 I2CDAT	C2H

在从动模式下，必须往地址寄存器中写入本机地址。地址寄存器由七个地址位组成。该地址寄存器可以读取或写入地址。

从动模式下通过下面组合列表可以各种操作：

（SLA：从机地址。W：写。R：读）

I ² C 操作	操作解释	从动模式操作控位			接收到信号或发送后 I ² CSTA 寄存器值	I ² C 硬件下一步操作
		STA	STO	AA		
接收到 SLA+R	接收到本机地址并且进行读数据操作	x	0	0	A8H: 接收到 SLA+R 并已经返回 ACK	发送最后一个字节数据, 并将接收 ACK
				1		发送一个自己数据, 并接收 ACK
接收到 SLA+R/W 时仲裁丢失或者接收到 SLA+R	接收到从机地址读写操作时丢失或者接收到本机地址并进行读操作	x	1	0	B0H: 接收到 SLA+R/W 仲裁丢失。或者接收到 SLA+R 并返回 ACK	发送最后一个字节数据, 并将接收 ACK
				1		发送一个自己数据, 并接收 ACK
发送 I ² CDAT 中的数据	本机发送数据	x	0	0	B8H: 发送数据字节	发送最后一个字节数据, 并将接收 ACK
				1		发送一个自己数据, 并接收 ACK
发送 I ² CDAT 中的数据	数据字节发送	0	0	0	C0H: 数据字节发送并接收到 NACK	切换到未寻址从机模式; 从机地址未识别
		0	0	1		切换到未寻址从机模式; 从

						机地址将被识别
		1	0	0		切换到未寻址从机模式：总线空闲后将发送开始信号
		1	0	1		切换到未寻址从机模式：从机地址将被识别；总线空闲后将发送开始信号
发送 I ² C DAT 中的数据	数据字节发送	0	0	0	C8H: 最后一个字节数据已经发送, 并接收到 ACK	切换到未寻址从机模式；从机地址未识别
		0	0	1		切换到未寻址从机模式；从机地址将被识别
		1	0	0		切换到未寻址从机模式：总线空闲后将发送开始信号
		1	0	1		切换到未寻址从机模式；从机地址将被识别；总线空闲后将发送开始信号
接收到 SLA+W	接收到本机地址并进行写入操作	x	0	0	60H: 已经接收到 SLA+W, 并返回 ACK	接收数据字节；返回 NACK；
				1		接收数据字节；返回 ACK；
接收到 SLA+R/W 时仲裁丢失；并接收到对应的 SLAR+W	接收到从机地址读写操作时丢失或者接收到本机地址并进行读操作	x	0	0	68H: 接收到 SLA+R/W 仲裁丢失。或者接收到 SLA+R 并返回 ACK	接收数据字节；返回 NACK；
				1		接收数据字节；返回 ACK；
接收到一个字节 DATA	从机寻址成功并已经接收到数据	x	0	0	80H: 寻址成功，接到数据，并已经回复 ACK	接收数据字节；返回 NACK；
				1		接收数据字节；返回 ACK
接收到一个字节 DATA	从机寻址成功并接收到数据	0	0	0	88H: 从机寻址成功，接收到 DATA，并回复 NACK	切换到未寻址从机模式；从机地址未识别
		0	0	1		切换到未寻址从机模式；从机地址将被识别
		1	0	0		切换到未寻址从机模式：总线空闲后将发送开始信号
		1	0	1		切换到未寻址从机模式；从机地址将被识别；总线空闲后将发送开始信号

接收到 STOP 信号或者重复 START 信号	当从机接收器或者从机发生器仍然被寻址时，接收到了 STOP 结束信号或者重复 START 开始信号	0	0	0	A0H: 接收到 STOP, 或者接收到重复的 START	切换到未寻址从机模式;从机地址未识别
		0		1		切换到未寻址从机模式; 从机地址将被识别
		1		0		切换到未寻址从机模式: 总线空闲后将发送开始信号
		1		1		切换到未寻址从机模式; 从机地址将被识别; 总线空闲后将发送开始信号

17.6 I²C 中断

I²C 中断标志位是 SI，如果 EI2C (EIE.0) =1，且 EA=1，当 I²C 中断发生时，CPU 会进入中断服务程序去执行中断代码。SI 需要软件清零。

当 SI 置位时，用户可以去读取 I2CSTA 寄存器的状态码，用户可根据读取的状态码，来确认哪部已经执行和决定下一步动作。

在 SI 被清零之前，SCL 低电平周期延迟，传输暂停，该状态对于从机处理接收到的数据非常有用，可以确保准确处理前一数据再接收下一个数据。SI 位被软件清零后，I²C 才会继续下一步：总线继续产生 START 或者 Repeat START 条件，STOP 条件，8 位数据，或者其他控制字节或位，因此在 SI 软件清除前，软件应该准备好合适的寄存器设置。

17.7 仲裁与同步模式

在多主机条件下，同一时间只有一个主机可在总线上传输数据。需要决定由哪个主机可控制总线以及实现传输。时钟同步与仲裁应用于配置多主机传输。时钟同步会在和其他设备同步 SCL 信号时运行。

当同一时间有两个主机要传输时，时钟同步将会在 SCL 由高电平转变成低电平的时候开始。如果主机 1 先将 SCL 切换为低电平，那么主机 1 会将 SCL 锁定在低电平状态直到将其切换成高电平状态。不过，如果有其他主机的 SCL 时序依然保持在低电平状态的话，那么主机 1 将 SCL 时序由低电平切换成高电平状态的过程将不会改变 SCL 的状态，SCL 时序将仍保持在低电平状态。也就是说，SCL 线被有最长低电平周期的主机保持在低电平状态。当所有设备的时钟周期都转换到高电平时，SCL 线将由低转成到高电平状态。在这期间，主 1 将保持在由低电平到高电平的等待状态，之后再继续它的传输。经过时钟同步之后，所有设备的时钟和 SCL 时钟是一样的了。仲裁是用来决定

哪个主机能够通过 SDA 信号来完成它的传输。两个主机可能会在同一时间发出开始信号以及传输数据，导致两者会互相影响。仲裁会强制使得其中一个主机失去总线的控制权。数据传输依然会继续，直到两个主机输出了不同的数据信号。如果其中一个主机传输了高电平状态，而另外一个主机传输了低电平状态，SDA 线会被拉低。输出高电平状态的主机将会检测到 SDA 线上的异常，并失去总线的控制权。输出低电平状态的主机成功获得总线的控制权，并继续它的传输，仲裁的过程中不会丢失数据

17.8 I²C 状态码及应用程序示例

17.8.1 状态码

I²STAT 状态寄存器中,有两种状态码不归属 20 种之前所述状态: F8H 及 00H。

F8H 用以标示在之前的传输中没有有效信息，意味着，SI 标志位为 0 且无 I²C 中断产生。

00H 标示在传输过程中，有总线错误发生。总线错误是指 START 或 STOP 在传输的过程中出现在错误的位置。所有状态码解释如下表：

表 17-1 I²C 状态码

主机模式		从机模式	
状态码	描述	状态码	描述
0x08	开始信号	0xA0	从机接收到 STOP 信号或者接收到重复 START 信号
0x10	主机重复开始信号	0xA8	接收到地址读并应答
0x18	主机发送地址写并接收应答	0xB0	仲裁丢失
0x20	主机发送地址写并无应答	0xB8	从机发送数据并应答
0x28	主机发送数据并接收应答	0xC0	从机发送数据无应答
0x30	主机发送数据并无应答	0xC8	从机发送最后一个字节
0x40	主机发送地址读并接收应答	0x60	从机接收到地址写并应答
0x48	主机发送地址读并无应答	0x68	仲裁丢失
0x50	主机接收数据并应答	0x80	从机接收数据并应答
0x58	主机接收数据并无应答	0x88	从机接收数据无应答
0xF8	释放总线		

17.8.2 程序示例

```
Void I2C_ISR (void) interrupt I2C_VECTOR // I2C_VECTOR=6
{
    switch (I2CSTA)
    {
        //=====
        //Bus Error, always put in ISR for noise handling
        //=====
        case 0x00: //00H, 发生总线错误*/
            STO = 1; //从总线错误中恢复
            break;
        //=====
        //Master Mode
        //=====
        case 0x08: //08H, 发送 START 信号*/
            STA = 0; //STA 位需要通过软件清除
            I2CDAT = SLA_ADDR; //加载 SLA+W/R
            break;
        case 0x10: //10H, 重复发送 START 信号*/
            STA = 0;
            I2CDAT = SLA_ADDR;
            break;
        //=====
        //Master Transmitter Mode
        //=====
        case 0x18: //18H, 发送 SLA+W, 接收 ACK*/
            I2CDAT = NEXT_SEND_DATA1; //加载数据
            break;
        case 0x20: //20H, 发送 SLA+W, 接收 NACK*/
            STO = 1; //传输停止
            AA = 1;
            break;
        case 0x28: //28H, 发送数据, 接收 ACK*/
            if (Conti_TX_Data) //如果继续发送数据
                I2CDAT = NEXT_SEND_DATA2;
            else //如果没有数据要发送
            {
                STO = 1;
                AA = 1;
            }
            break;
        case 0x30: //30H, 发送数据, 接收 NACK*/
            STO = 1;
```

```

        AA = 1;
        break;
//=====
//Master Receiver Mode
//=====
case 0x40:                                /*40H, 发送 SLA+R, 接收 ACK*/
        AA = 1;                          //接收数据, 返回 ACK。
        break;
case 0x48:                                /*48H, 发送 SLA+R, 接收 NACK*/
        STO = 1;
        AA = 1;
        break;
case 0x50:                                /*50H, 接收数据, 发送 ACK*/
        DATA_RECEIVED1 = I2CDAT;         //存储接收数据
        if (To_RX_Last_Data1)             //如果最后的数据被接收
            AA = 0;                       //接收数据, 返回 NACK。
        else                               //如果继续接收数据
            AA = 1;
        break;
case 0x58:                                /*58H, 接收数据, 发送 NACK*/
        DATA_RECEIVED_LAST1 = I2CDAT;
        STO = 1;
        AA = 1;
        break;
//=====
//Slave Receiver and General Call Mode
//=====
case 0x60:                                /*60H, 主机收到 SLA+W, 返回 ACK*/
        AA = 1;
        break;
case 0x68:                                /*68H, 主机在发送 SLA+R/W 时, 仲裁丢失;
        并接收到了对应的 SLA+W; 已返回 ACK。*/
        AA = 0;                          //仲裁丢失后, 接收数据, 返回 NACK
        STA = 1;                          //如果总线空闲, 重新尝试发送 START
        break;
case 0x80:                                /*80H, 接收 DATA, 并返回 ACK*/
        DATA_RECEIVED2 = I2CDAT;
        if (To_RX_Last_Data2)
            AA = 0;
        else
            AA = 1;
        break;
case 0x88:                                /*88H, 接收 DATA, 并返回 NACK,
        切换至未寻址的从机模式*/

```

```

        DATA_RECEIVED_LAST2 = I2CDAT;
        AA = 1;
        break;

//=====
//    Slave Mode
//=====
case 0xA0:                /*A0H, 当从机接收器或者从机发生器仍然被寻址时,
                           接收到了 STOP 结束信号或者重复 START 开始信号*/

        AA = 1;
        break;

//=====
//    Slave Transmitter Mode
//=====
case 0xA8:                /*A8H, 接收 SLA+R, 返回 ACK*/
        I2CDAT = NEXT_SEND_DATA3;
        AA = 1;           //当 AA 置 1, 发送一个字节的数
        break;
case 0xB0:                /*B0H, 主机在发送 SLA+R/W 时仲裁丢失了;
                           已接收到了对应的 SLA+R; 且已返回 ACK 信号。*/
        I2CDAT = DUMMY_DATA;
        AA = 0;           //当 AA 为 0, 发送最后一个字节的数据
        STA = 1;          //如果总线空闲, 重新尝试发送 START
        break;
case 0xB8:                /*B8H, 数据字节已经发送, 将接收到 ACK 信号。*/
        I2CDAT = NEXT_SEND_DATA4;
        if (To_TX_Last_Data)                //如果最后一个数据被传输
            AA = 0;
        else
            AA = 1;
        break;
case 0xC0:                /*C0H, 数据字节已经发送, 并已接收到 NACK 信号
                           切换至未寻址的从机模式*/

        AA = 1;
        break;
case 0xC8:                /*C8H, 最后一个字节的数据已经发送,
                           并已接收到 ACK 信号, 切换至未寻址的从机模式*/

        break;
    }
    SI = 0;                //SI 应该是 I2C ISR 的最后一个命令
    while(STO);            //等待传输停止或总线无错误, STO 由硬件清除
}

```

17.9 I²C 寄存器

I2CCON- I2C 接口操作控制寄存器

地址：C0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CCON	CR[2]	ENSI	STA	STO	SI	AA	CR[1]	CR[0]
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

位	名称	描述
7、1、0	CR[2:0]	I ² C 时钟速率 000: Fsys/256; 001: Fsys/224; 010: Fsys/192; 011: Fsys/160; 100: Fsys/960; 101: Fsys/120; 110: Fsys/60; 111: (TC1 计数器溢出率)/8
6	ENSI	I ² C 使能位 0: 关闭; 1: 使能。
5	STA	START 标志位 0: 没有发送 START 开始信号; 1: 若总线空闲则发送 START 开始信号。 用户需软件清除。
4	STO	STOP 标志位 0: 没有发送 STOP 结束信号; 1: 若 I ² C 总线为主机模式, 则发送 STOP 结束信号。 STO 由硬件自动清 0。
3	SI	串行中断标志位 I ² C 所有 20 种状态中出现一种, 硬件就会置 1 此位 (F8H 除外), 此时软件根据读取 I2STAT 值, 来确认哪步已经执行和决定下一步动作。 SI 由软件清 0。在 SI 被清 0 之前, SCL 低电平周期延长, 传输暂停, 该状态对于从机处理接收到的数据非常有用, 可以确保准确处理前一数据再接收下一个数据。 SI 位被软件清 0 后, I ² C 外设才会继续下一步: SI 清除后, 总线将继续产生 START 和 repeat START 条件, STOP 条件, 8 位数据, 或者其他控制字节或位。因此, 在 SI 软件清除前, 软件应该准备好合适的寄存器设置
2	AA	应答响应 ACK 标志位 当 AA = 1 时, 将在以下情况下返回 “ACK”: - 当 I ² C 处于主接收器模式时已接收到一个数据字节 - 当 I ² C 处于从接收器模式时已接收到一个数据字节 当 AA = 0 时, 在以下情况下将返回 “NOACK”: - 当 I ² C 处于主接收器模式时已接收到一个数据字节 - 当 I ² C 处于从接收器模式时已接收到一个数据字节

I2CADR - I²C 从动地址

地址: C1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CADR	I2CADR[6:0]							-
R/W	R/W							-
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:1	I2CADR [6:0]	I2C 从机地址
0		

I2CDAT - I²C 数据缓存器

地址: C2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CDAT	I2CDAT[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	I2CDAT[7:0]	I2CDAT 寄存器存储的是要通过 I2C 总线发送出去的一字节数据, 或者是刚从 I2C 总线上接收到的一字节数据。当它不在字节移位的过程中时, 控制器可以读写这个 8 位的直接寻址特殊功能寄存器。由于 I2CDAT 寄存器没有缓存和双缓冲, 因此当发生 I2C 中断时, 用户只能读 I2CDAT 寄存器。

I2CSTA - I²C 状态代码

地址: C3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSTA	I2CSTA[7:3]					-	-	-
R/W	R					-	-	-
复位值	1	1	1	1	1	0	0	0

位	名称	描述
7:3	I2CSTA[7:3]	I2C 状态代码

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 与串行的外围设备进行通信。串行时钟线 (SCLK) 与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 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, 串行时钟可在七种不同的比特率中选择其一。

SPI 的特性如下:

- ◆ 全双工同步串行数据传输
- ◆ 支持主/从机模式
- ◆ 七个 SPI 主控波特率: 系统时钟的 1/2、1/4、1/8、1/16、1/32、1/64、1/128
- ◆ 主机时钟频率最高达 16Mbps, 从机时钟频率高达 $F_{sys}/8$ bps
- ◆ 具有可编程极性和相位的串行时钟
- ◆ 产生中断
- ◆ 写入碰撞标志保护
- ◆ 8 位数据, 先传输最高有效位 (MSB), 后传输最低有效位 (LSB)
- ◆ 特殊功能寄存器连接主机 CPU
- ◆ 无双向端口, 标准 SPI 引脚外部连接到三态缓冲器
- ◆ 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：

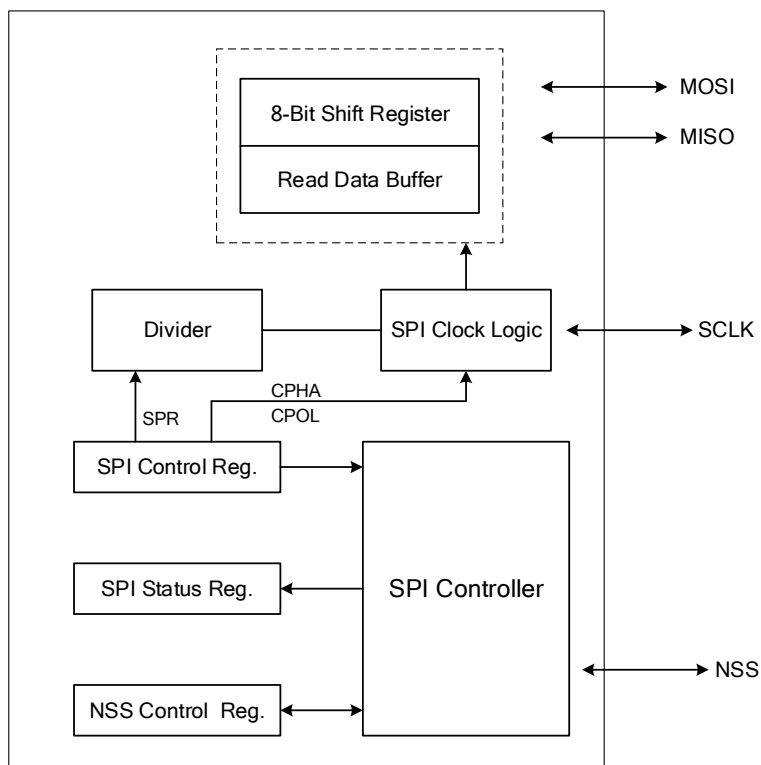
```
P14CFG = 0x0E;      //选择 P1.4 配置成 NSS 通道
P15CFG = 0x0F;      //选择 P1.5 配置成 SCLK 通道
P16CFG = 0x10;      //选择 P1.6 配置成 MOSI 通道
P17CFG = 0x11;      //选择 P1.7 配置成 MISO 通道
```

配置为 SCLK、MOSI、MISO 和 NSS 口。

18.3 SPI 硬件描述

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 18-1 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，MOSI 引脚是从机设备输入数据线，MISO 是从机设备输出数据线。
- 当 SPI 被配置为主机设备时，MISO 引脚是主机设备输入数据线，MOSI 是主机设备输出数据线。

18.4 SPI 时钟分频

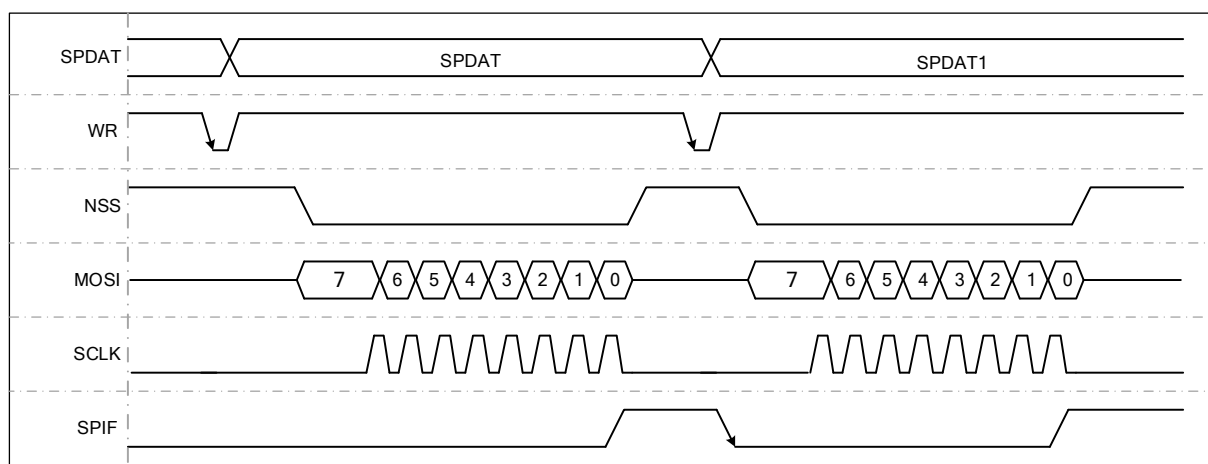
SM2-SM0 控制 SPI 时钟分频

SM2	SM1	SM0	System clock divided by
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128

18.5 SPI 主控模式

SPI 控制模式共有 7 种类型的时钟发生器，从 $F_{sys}/2 \sim F_{sys}/128$ ，产生的时钟由 SCK 引脚输出。当 SPI 配置为主机模式时，SPI 通过 NSS 线选择从机设备，通过写入 SPDAT 启动传输。当新字节写入 SPDAT 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。

图 18-2 自动从机选择线使用



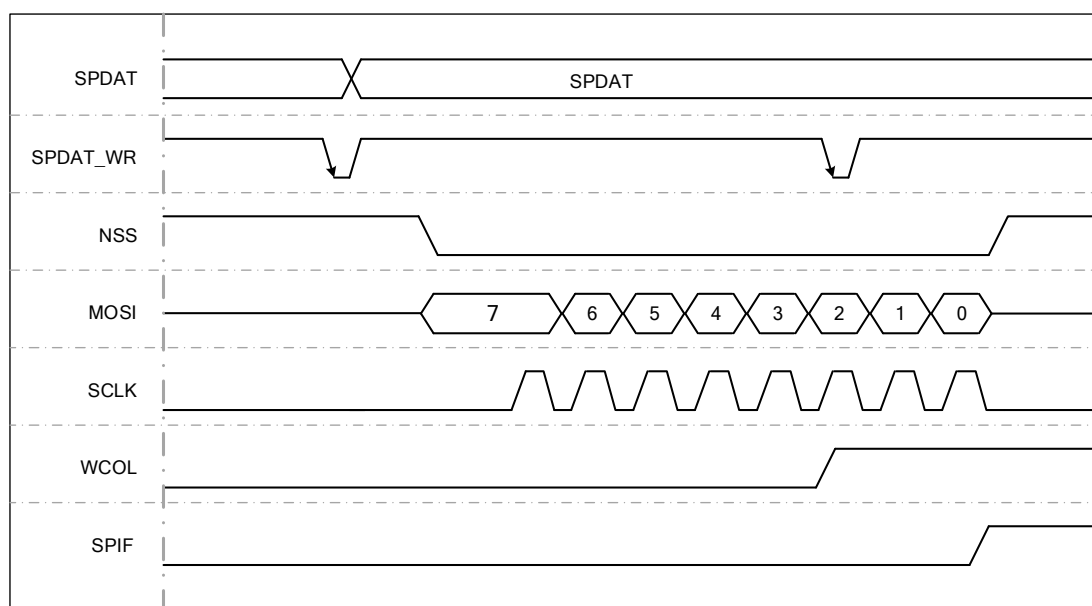
18.5.1 写冲突错误

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 SPSTA 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。如果要清除 WCOL 位，用户应执行以下步骤：

- 读取 SPSTA 寄存器
- 访问 SPDAT 寄存器

图 18-3 SPI 主模式下的写冲突错误



写冲突的具体条件为：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八个 SCLK 下降沿期间。如果在此期间写 SPDAT，则会出现写冲突，WCOL 置 1。

18.6 SPI 从动模式

设定 MSTR 为 0，SPI 将工作在从机模式。当作为从机模式时，SPCLK 管脚变为输入脚，SS 管脚也变为输入脚。当从机设备的 SS 管脚不为低时，主机设备不能与从机交换数据

在从机模式下，数据通过 MOSI 管脚从主机向从机传输，通过 MISO 管脚从从机向主机传输。通过主机 SPCLK 的时钟控制，数据进入位移寄存器。在移位寄存器接收到一个字节后，数据将移到读数据缓存，同时 SPIF 置 1

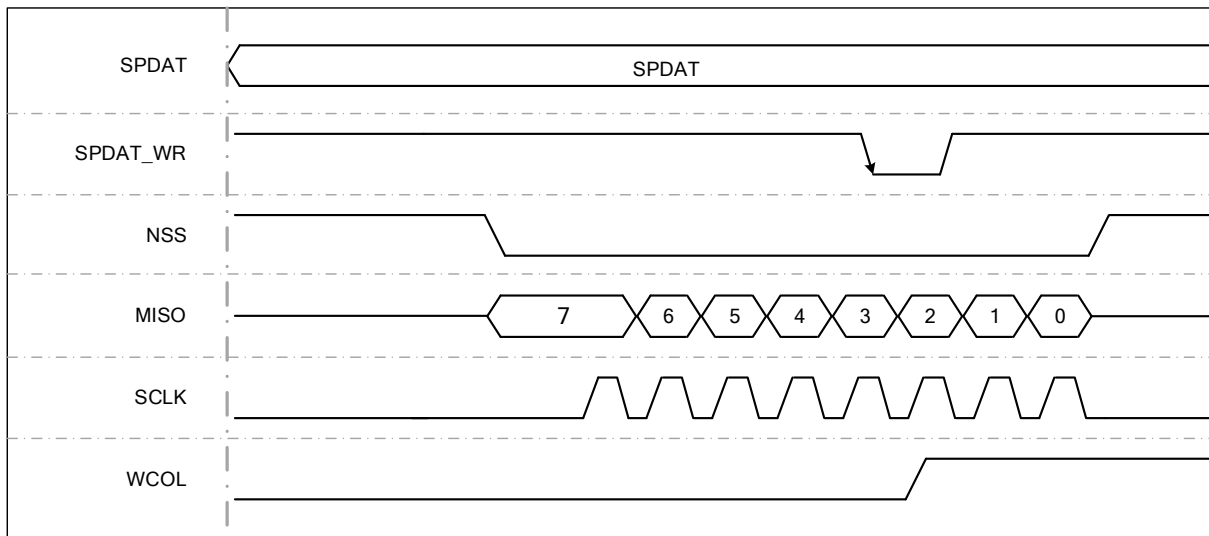
18.6.1 写冲突错误

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 SPSTA 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，用户应执行以下顺序：

- 读取 SPSTA 寄存器的内容
- 访问 SPDAT 寄存器（读或写）

图 18-4 SPI 从模式下的写冲突错误



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(SPCON) 中选择使用两个控制位的(串行时 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 控制位选择高电平或低电平有效时钟。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。

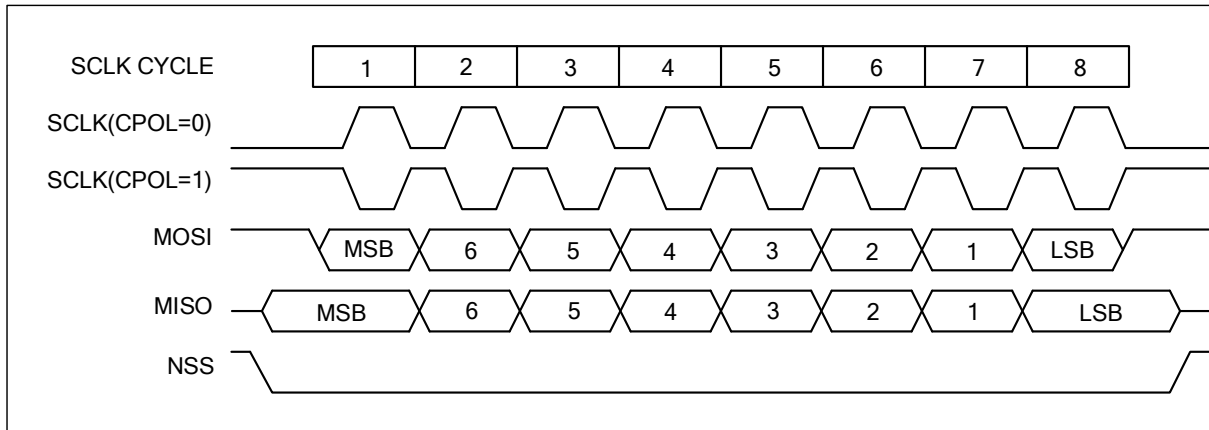
18.7.2 SPI 传输格式

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。

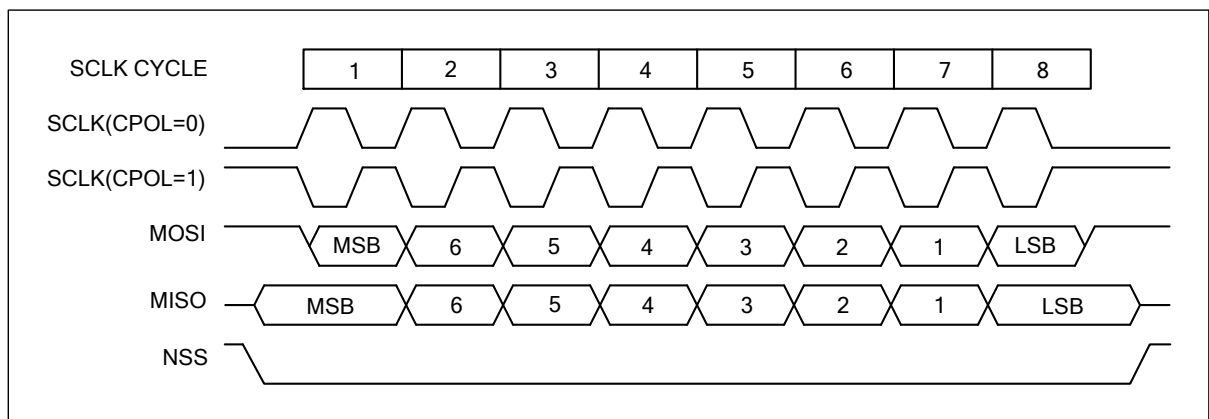
图 18-5 CPHA=0 SPI 传输格式



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 18-6 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 SPDAT 的软件启动传输。

18.8.2 SPI 传输结束

当 SPIF 标志置 1 时，SPI 传输完成。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。

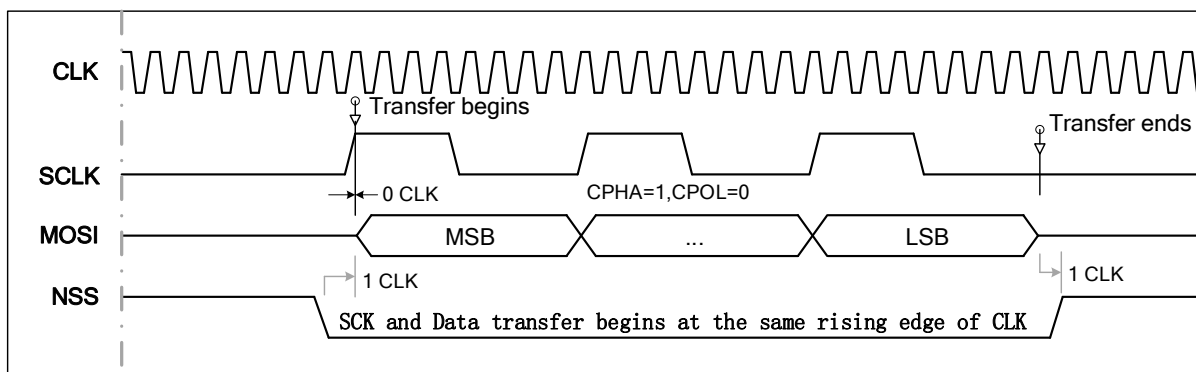
因为 SCLK 线可以与从机的 MCU 时钟异步，并且从机不能像访问 SCLK 周期那样访问主机尽可能多的信息，所以当 SPI 作为从机运行时，结束周期是不同的。例如，当 CPHA=1 时，其中最后一个 SCLK 边沿在第八个 SCLK 周期的中间发生，从机无法知道上一个 SCLK 周期是何时结束。由于这些原因，从机认为在串行数据的最后一位被采样之后，传输完成，这对应于第八个 SCLK 周期的中间。

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 SPDAT。

18.9 SPI 时序图

18.9.1 主控模式传输

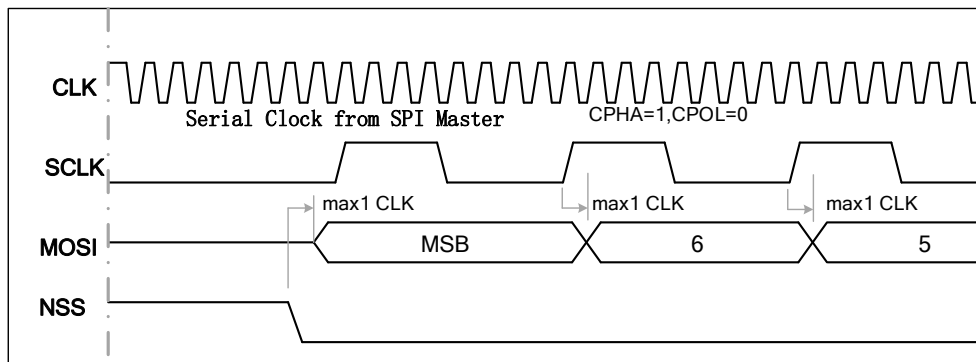
图 18-7 主模式时序图



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 18-8 从模式时序图



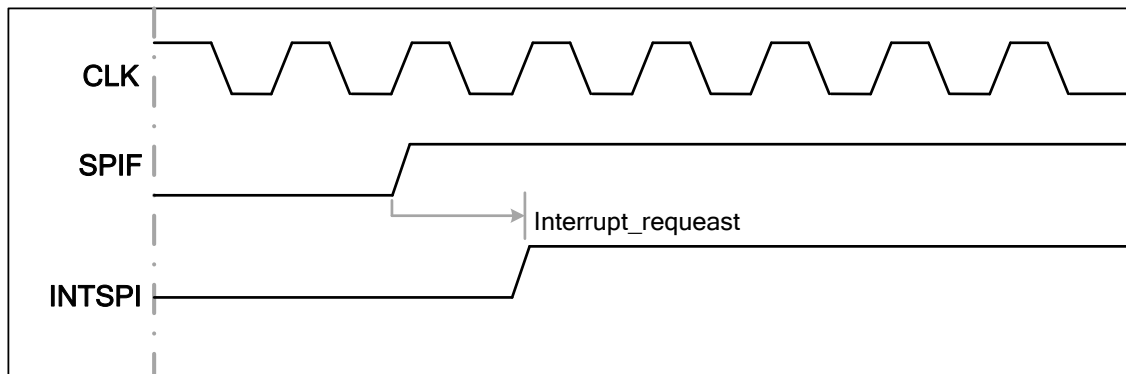
18.10 SPI 中断

SPI 的两个状态标志可以产生中断请求，分别是 SPIF 和 MODF。

使能 SPI 中断必须将其使能位 ESPI 置 1，且将总中断使能位 EA 置 1。

- 如果 SPI 相关的中断使能打开，SPI 完成自身数据传输后标志位 SPIF=1 时，CPU 将进入中断服务程序。SPIF 位通过先读取 SPSTA 寄存器，然后读取 SPDAT 寄存器来清除。

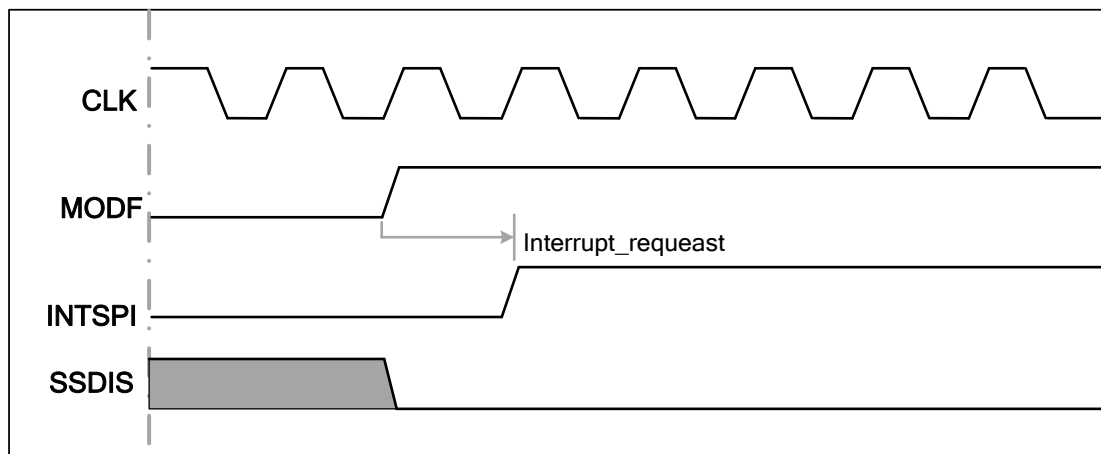
图 18-9 SPIF 中断



Interrupt Request Generated by "spif" Flag

- 如果当 SSN 输入的电平与所选操作模式不一致时，MODF 标志由硬件自动置位（SPI_MS 模块配置为主模式，并且在 SSN 输入处检测到低电平）。该位通过读取 SPSTA 寄存器来由软件清零。

图 18-10 MODF 中断



Interrupt Request Generated by “modf” Flag

18.11 SPI 寄存器

SPSTA - SPI 状态寄存器

地址: E1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSTA	SPIF	WCOL	SSERR	MODF	-	-	-	-
R/W	R	R	R	R	-	-	-	-
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7	SPIF	SPI 通讯完成标志位 数据传输完成后由硬件设置。 通过先读取 SPSTA 寄存器，然后读取 SPDAT 寄存器来清除。
6	WCOL	写操作冲突标志位 在通讯时对 SPDAT 执行写操作自动设置为 1 通过先读取 SPSTA 寄存器，然后读取 SPDAT 寄存器来清除。
5	SSERR	同步从模式错误标志位 若 SSN 错误控制时自动设置为 1 通过清 SPEN 自动清零。
4	MODF	模式错误标志位。 该位通过硬件置位。通过读取 SPSTA 寄存器来由软件清零。

SPCON - SPI 控制寄存器

地址：E2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCON	SM2	SPEN	SSDIS	MSTR	CPOL	CPHA	SM1	SM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	1	0	1	0	0

位	名称	描述
7、1、0	SM[2:0]	SPI 波特率发生器（仅在主模式下有效） 000: $F_{sys}/2$ 001: $F_{sys}/4$ 010: $F_{sys}/8$ 011: $F_{sys}/16$ 100: $F_{sys}/32$ 101: $F_{sys}/64$ 110: $F_{sys}/128$ 111: reserved
6	SPEN	SPI 通讯使能 0: 禁止； 1: 使能。
5	SSDIS	SS 使能 0: 使能 SSN 输入在主从模式 1: 禁止 SSN 输入在主从模式 在从模式，如果 CPHA=0，则 SSDIS 位不起作用，这时 MODF 中断不会请求
4	MSTR	SPI 模式选择位 0: 从模式； 1: 主模式。
3	CPOL	SCK 引脚空闲状态位 0: SCK 低电平空闲； 1: SCK 高电平空闲。
2	CPHA	数据的时钟相位锁存控制位 0: 当 SCKI/SCKO 非空闲时，锁存数据； 1: 当 SCKI/SCKO 空闲时，锁存数据。

SPDAT-SPI 数据缓存器

地址：E3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPDAT	SPDAT[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	SPDAT[7:0]	SPDAT 是“接收数据”寄存器的读/写缓冲区 当写入 SPDAT 时，数据被直接放入移位寄存器(没有传输缓冲区)。 读取 SPDAT 时，返回位于接收缓冲区中的值，而不是移位寄存器中的值。

19 MDU 模块

乘除法单元是一个内置算法协处理器，可以使单片机去执行额外扩展的算法操作。该单元提供 32 位无符号的除法，16 位无符号的乘法，移位和规范化操作，可以通过写入 MD0-MD5 寄存器不同的顺序来识别这些操作。

19.1 乘法（16 位×16 位）

乘法的基础原理包括 3 部分：被乘数、乘数和得数。进行一个乘法要求以下面的写入顺序：MD0（被乘数的低字节），MD4（乘数的低字节），MD1（被乘数的高字节），然后 MD5（乘数的高字节）。写入 MD5 寄存器结束时，自动开始乘法操作，乘法操作共需要 11 个 CPU 周期，其有效的得数可通过特定的顺序读取：MD0（LSB）、MD1、MD2，然后 MD3（MSB）寄存器。

19.2 除法（32 位/16 位和 16 位/16 位）

MDU 支持 2 种除法操作：32 位/16 位和 16 位/16 位。第一种需要 17 个 CPU 周期来进行计算；第二种只需要 9 个周期。

32 位除法以下面特定的写入顺序开始：MD0、MD1、MD2、MD3、MD4 和 MD5。32 位被除数存入 MD3（最高有效位）到 MD0 寄存器，16 位除数存入 MD5 到 MD4 寄存器（MSB 位于 MD5 寄存器）。

16 位除法只需要 4 个寄存器。16 位被除数存入 MD1 到 MD0 寄存器，16 位除数存入 MD5 到 MD4 寄存器（最高有效位位于 MD1 和 MD5 寄存器）。其合适顺序为：MD0、MD1、MD4 和 MD5。MDU 从写入数据到 MD5 寄存器之后开始计算，需要 9 个或者 17 个 CPU 周期，取决于被除数的长度。计算得出的商存入 MD3 到 MD0 寄存器（32 位除法），或者 MD1 到 MD0 寄存器（16 位除法），LSB 位于 MD0 寄存器。不管是否执行了除法操作，余数都放在 MD5（MSB）和 MD4 寄存器中。但是必须最后读取 MD5 以指示是否完成全部除法操作。

19.3 移位和规范化

移位和规范化操作移动 32 位寄存器（MD3 到 MD0，MSB 位于 MD3）一段特定或者不确定的次数。

移位操作中，通过指定的位数左移或者右移 32 位无符号的整数，移动的方向和位数都是指定的，存在 ARCON 寄存器中。移位操作需要 3-18 个 CPU 周期，取决于移位的次数。

规范化操作中，32 位无符号的整数会重复地向左移位，直到最高有效位（MD3 寄存器的第 7 位）为 1。规范化操作需要 4-19 个 CPU 周期，取决于实际移位的次数。

移位操作和规范化操作都以下面的写入顺序开始：MD0、MD1、MD2、MD3，最后为 ARCON 寄存器。最终结果存入 MD0-MD3 寄存器中，以下面顺序来读取：MD0、MD1、MD2、MD3。

19.4 错误标志（MDEF）

错误标志 MDEF 指示未正确运行的操作（算法操作被重新开始或者被新操作打断），检测错误机制会在第一阶段写数据到 MD0 寄存器时自动使能，并在第三阶段的最后一条对 MD3（乘法或者移位/正常化）或者 MD5 寄存器（除法）进行读取指令时关闭。

出现下列情形时，错误标志置 1：

在 MDU 操作的第二阶段中对“MDx”寄存器（MD0~MD5 以及 ARCON 寄存器当中的任意一个）进行写操作（重新启动或计算中断）。

在错误标志位机制已经使能的情况下，在 MDU 操作的第二阶段中对“MDx”寄存器进行读取操作。此时错误标志位会被置 1，但是计算不会被打断。

错误标志位只有在读“ARCON”寄存器之后才会复位，且该标志位为只读。

19.5 溢出标志（MDOV）

出现下列情形时，溢出标志 MDOV 置 1：

- 除数为 0；
- 乘法的结果大于 0xFFFF；
- MD3 的最高有效位置 1（MD3.7=1）时开始规范化操作；
- MDU 操作与上述情形不符合时，溢出标志清零。注意：溢出标志只受硬件控制，不能执行写操作。

19.6 MDU 寄存器

MDx - MDU 运算寄存器(MD0-MD5)

XSFR 地址: E9H--EEH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MDx	MDx7	MDx6	MDx5	MDx4	MDx3	MDx2	MDx1	MDx0
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

位	名称	描述
7:0	MDx[7:0]	乘法/除法寄存器

ARCON - MDU 状态寄存器

地址: EFH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ARCON	MDEF	MDOV	SLR	SC[4:0]				
R/W	R	R	R/W	R/W				
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7	MDEF	MDU 错误标志 MDEF 显示错误的操作（算法操作重新开始或者被新操作打断）
6	MDOV	MDU 溢出标志 MDU 操作时发生溢出
5	SLR	移位方向 0: 左移 1: 右移
4:0	SC[4:0]	移位计数器 写入 0x00: 执行规范化操作，完成后可读取实际的移位次数。 写入其它值: 指定移位操作的次数，MSB 位于 SC4。

20 加减移位模块

20.1 概述

JSC68F016 具有直接进行对数据移位计算功能，可以快速获取移位后的结果。移位功能模块由加减控制寄存器进行直接操作。

20.2 加减移位寄存器

ASCON - 加减控制寄存器

地址：B3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ASCON	ASCON [7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	ASCON [7:0]	加减控制位操作： 00000000: 移位控制寄存器无操作 00000001: {AS0H,AS0L}16 位寄存器数据加 1 00000010: {AS0H,AS0L}16 位寄存器数据减 1 00000011: {AS0H,AS0L}16 位寄存器数据左移 1 位 00000100: {AS0H,AS0L}16 位寄存器数据右移 1 位 00000101: {AS1H,AS1L}16 位寄存器数据加 1 00000110: {AS1H,AS1L}16 位寄存器数据减 1 00000111: {AS1H,AS1L}16 位寄存器数据左移 1 位 00001000: {AS1H,AS1L}16 位寄存器数据右移 1 位 00001001: {AS2H,AS2L}16 位寄存器数据加 1 00001010: {AS2H,AS2L}16 位寄存器数据减 1 00001011: {AS2H,AS2L}16 位寄存器数据左移 1 位 00001100: {AS2H,AS2L}16 位寄存器数据右移 1 位 00001101: {AS3H,AS3L}16 位寄存器数据加 1 00001110: {AS3H,AS3L}16 位寄存器数据减 1 00001111: {AS3H,AS3L}16 位寄存器数据左移 1 位 00010000: {AS3H,AS3L}16 位寄存器数据右移 1 位 其他：保留

AS0H - 第一个 16 位数据寄存器高 8 位

地址：B5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AS0H	AS0H[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	AS0H [7:0]	第一个 16 位数据寄存器高 8 位

AS0L - 第一个 16 位数据寄存器低 8 位

地址：B4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AS0L	AS0L[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	AS0L[7:0]	第一个 16 位数据寄存器低 8 位

AS1H - 第二个 16 位数据寄存器高 8 位

地址：B7H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AS1H	AS1H[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	AS1H [7:0]	第二个 16 位数据寄存器高 8 位

AS1L - 第二个 16 位数据寄存器低 8 位

地址：B6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AS1L	AS1L[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	AS1L[7:0]	第二个 16 位数据寄存器低 8 位

AS2H - 第三个 16 位数据寄存器高 8 位

地址：BFH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AS2H	AS2H[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	AS2H [7:0]	第三个 16 位数据寄存器高 8 位

AS2L - 第三个 16 位数据寄存器低 8 位

地址：BEH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AS2L	AS2L[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	AS02L[7:0]	第三个 16 位数据寄存器低 8 位

AS3H - 第三个 16 位数据寄存器高 8 位

地址：FEH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AS3H	AS3H[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	AS3H [7:0]	第三个 16 位数据寄存器高 8 位

AS3L - 第三个 16 位数据寄存器低 8 位

地址：FDH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AS3L	AS3L[7:0]							
R/W	R/W							
复位值	0	0	0	0	0	0	0	0

位	名称	描述
7:0	AS3L[7:0]	第三个 16 位数据寄存器低 8 位

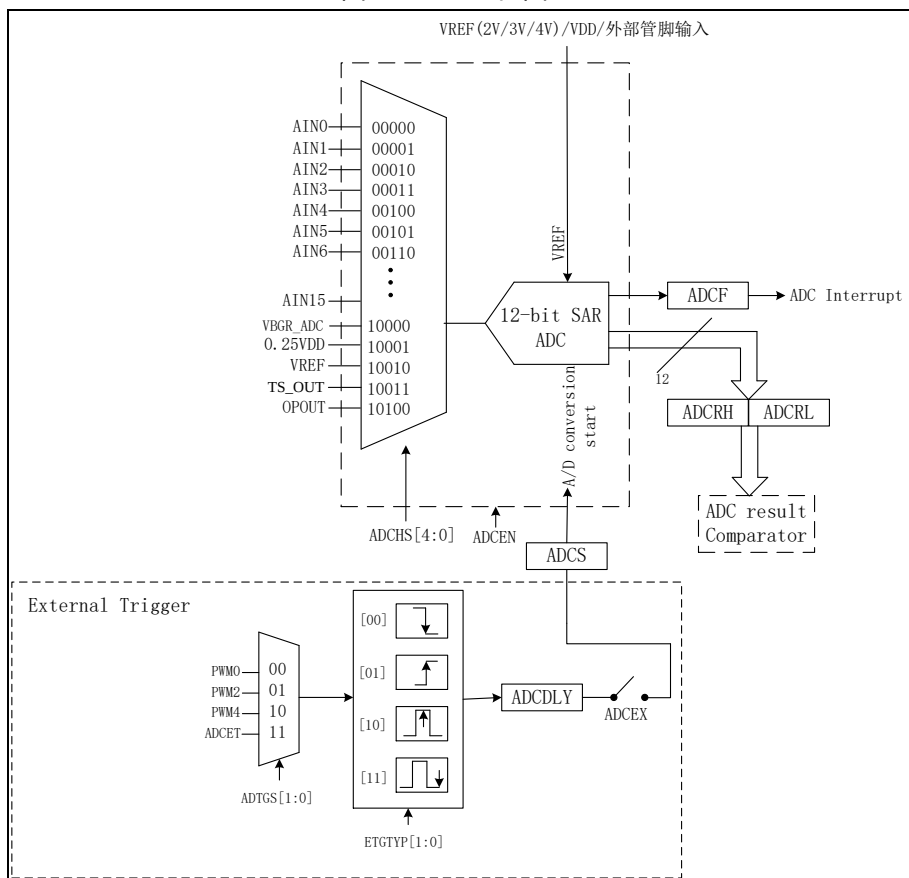
21 模数转换(ADC)

21.1 ADC 概述

JSC68F016内嵌12位逐次逼近寄存器型(SAR)的模拟数字转换器(ADC)。模数转换模块负责将管脚上的模拟信号转换为12位二进制数据。JSC68F016支持21通道单端输入模式。该组采样保持电容为转换电路的输入端。然后转换器通过逐次逼近的方式得到有效结果并存放在ADC结果寄存器中。

ADC参考电压可以用内部产生的基准，也可以用外部基准电压，并由外部管脚输入。

图 21-1 ADC 框图



21.2 ADC 配置

配置和使用 ADC 时应考虑如下因素：

- ◆ 端口配置
- ◆ 通道选择
- ◆ ADC转换时钟
- ◆ 中断控制
- ◆ 结果存储

21.2.1 端口配置

ADC 内部普通模拟转换通道对应于 16 个 GPIO 口，当模拟信号从端口输入并进行转换时，需要对相应的端口进行配置，通过将 GPIO 口配置为模拟输入。详细配置信息参见相应的端口章节。

21.2.2 通道选择

ADC 内部转换通道有 21 个，其中有 16 个是普通模拟转换通道，对应于外部 IO 端口输入，5 个为内部通道分别对内部电压 0.25VDD，VREF，VBGR_ADC(1.2V)，OPOUT 和温度传感器。通过 ADC 控制寄存器 ADCCON0 的 ADCHS [4:0]位进行相应的配置，如果选择 VBGR_ADC，则需要使能 BGREN（ADCCON3.2）位。如果是普通模拟转换通道则应把相应端口配置为模拟输入。详细见 GPIO 复用功能配置。

21.2.3 ADC 参考电压

ADC 参考电压可以由内部 VDD 经过内部的基准电压源产生，也可以通过外部引脚直接引入。内部基准电压源可产生如下几种电压：2V、3V、4V、VDD，使用内部基准电压源的时候，必须使能 ADCCON3 寄存器中的 VREFEN 位或者 VDDEN 位。同时在 ADOC 寄存器中通过 VREFSEL 位选择 VREF 源；通过 VREFS 选择使用内部还是外部基准电压。使用外部基准源时必须配置相应的 GPIO 输入管脚。

21.2.4 转换时钟

ADC 模块时钟是系统时钟经过分频后驱动，时钟可以通过寄存器 ADCON2 的寄存器的 ADCKS 位来选择转换的时钟源。有以下 8 种可能的时钟频率进行选择：

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

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 12.5 个 T_{AD} 周期+采样保持周期。必须符合相应的 T_{AD} 规范。才能获的正确的转换结果。

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	$F_{sys}/32$
48MHz	$F_{sys}/64$

21.2.5 ADC 中断

ADC 模块允许在完成数模转换后产生一个中断，其标志位为 ADCF。通过 IE 寄存器的 EDAC 位置位使能 ADC 中断。ADCF 置位后必须用软件清零。每次转换结束后 ADCF 位都会被置 1，与是否允许 ADC 中断无关

21.2.6 结果存储

输入信号经过 ADC 转换后得到 12 位转换结果。转换结果存储在两个寄存器 ADCRL，ADCRH 中，其中 12 位结果的高八位存储在高寄存器 ADCRH 中，12 位结果的低四位存储在低寄存器 ADCRL 中的低四位。

21.3 ADC 硬件触发启动

除了通过软件启动 AD 转换外，JSC68F016 的 ADC 模块还提供硬件触发方式启动 AD 转换。一种为外部边沿触发方式，一种为 PWM 通道的边沿或周期触发方式。硬件触发方式需要将 ADCEX 置 1，即外部触发 ADC 功能。通过 ETGSEL 位选择外部触发源，通过 ETGTYP 位选择触发类型。

21.3.1 外部端口边沿触发 ADC

ADET 引脚边沿自动触发 AD 转换，此时位 ETGSEL[1:0]为 11 选择外部引脚触发，位 ETGTYP[1:0]可选择任何边沿触发。

21.3.2 PWM 触发 ADC

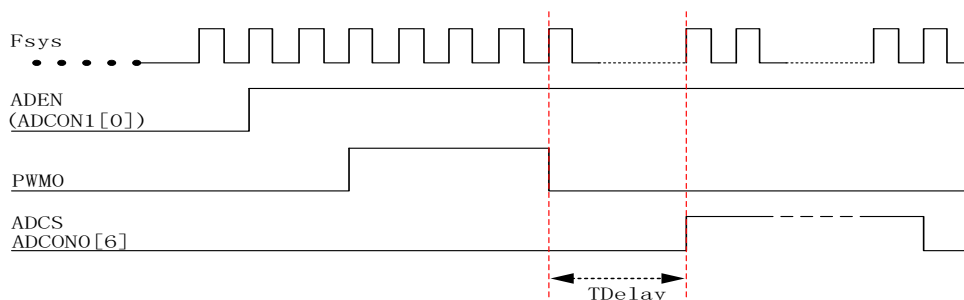
PWM 可选择是由边沿或者周期零点/中点触发 ADC 转换。此时位 ETGSEL[1:0]选择 PWM 通道（PWM0,PWM 2,PWM 4），位 ETGTYP[1:0]可选择边沿类型或者周期类型触发方式。

21.3.3 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，需要经过一定延时后才将触发启动 ADC 转换。延时由 ADDLY[7:0]和 ADCLY.8Y 一起决定，结构框图如下图所示，

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

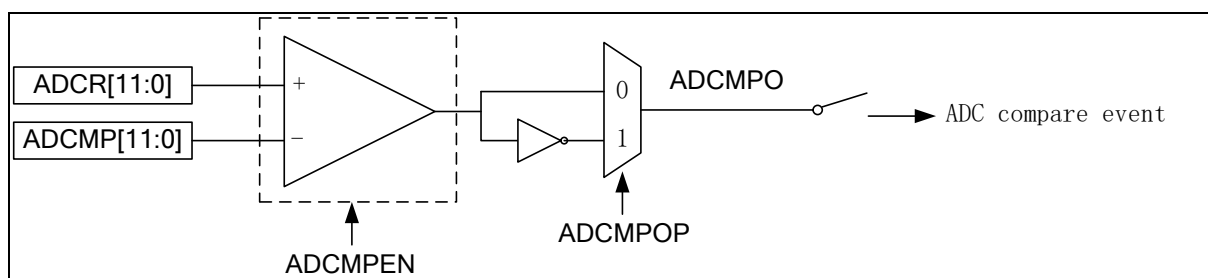
图 21-2 硬件触发启动延时



21.4 ADC 结果比较器

JSC68F016 ADC 提供一组数字比较器，用于比较 ADC 的 12 位转换结果与预先填入寄存器 ADCMPH 及 ADCMPL 的内容是否一致。ADC 比较器使能位为 ADCMPEN (ADCCON2.5) 一旦设定，每次 AD 转换结束都会进行比较。ADCMPO (ADCCON2.4) 的值显示根据 ADCMPOP (ADCCON2.6) 设定的比较结果。当 ADFBEN (ADCCON2.7) 设置后，ADC 比较结果可触发 PWM 故障刹车。

图 21-3 ADC 结果比较功能



21.5 ADC 偏移量校准

JSC68F016 可以对 ADC 进行偏移量校准，具体校准步骤如下：

- ◆ BGREN 使能；选用内部 VREF，VREFEN 使能；VREFSEL[1:0]=11 选 VREF=2V；ADC 选择时钟 3M；
- ◆ 对 ADOC 寄存器的 CALI 位置 1，SIGN 位置 1，同时对 VOF 置 111，这时转换结果正偏。启动修调。
- ◆ 启动 ADC 进行转换，并读取 ADC 结果。
- ◆ 如果 ADC 结果为 0 退出修调；如果 ADC 结果不为 0 则对 VOF[2:0] 的值进行减小。
- ◆ 如此循环测试，直到读取的 ADC 结果为 0，完成修调

- ◆ 如果 VOF[2:0]的值减小至 000，结果仍然是正极性（ADC 结果不为 0），也将不再修调，退出修调
- ◆ 清 ADOC 寄存器的 CALI 位。退出修调

21.6 ADC 工作原理

21.6.1 启动转换

将 ADCON1 寄存器的 ADCEN 位置 1，使能 ADC 模块，然后将 ADCON0 寄存器的 ADSC 位置 1，开始模数转换。如果是外部触发启动，则需在 ADSC 位置 1 后要等待触发信号的触发，然后才会开始启动转换

21.6.2 完成转换

当转换完成时,ADC 模块将:

- ◆ 将转换的结果更新ADRH:ADRL寄存器;
- ◆ 清零ADCS位;
- ◆ 将ADCF标志位置1;

21.6.3 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。

21.6.4 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

- 将引脚配置为模拟输入引脚
- 禁止引脚其他输入功能复用

2.配置ADC模块

- 选择ADC转换时钟
- 选择ADC输入通道（如果启用OP模块则应对OP模块寄存器进行相应配置）
- 选择触发源
- 选择基准电压
- 启动ADC模块

3.配置ADC中断（如果需要使用中断）

- 清零ADC中断标志位
- 允许ADC中断
- 允许全局中断

4.等待所需采集时间

5.将ADCS置位，启动转换（如果是外部触发需等待外部触发信号）

6.由如下方法之一等待ADC转换结束

- 查询ADCS位
- 等待ADC中断

7.读ADC结果

8.将ADC中断标志位清零

如果需要再一次转换，则需要再一次置位ADCS或者等待下一次外部触发

21.6.5 转换过程中进入休眠

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。禁止 ADC 转换过程中让芯片进入休眠

21.7 基于 OP 的 ADC 应用

在应用中经常遇到小信号模拟量，采集这些模拟信号需要放大后采集，芯片提供了一种经 OP 放大后的 ADC 应用，使用步骤如下：

- 配置外部端口为模拟输入
- 配置 OP 模块的相关寄存器，使能 OP 模块（在配置 OP 之前，如果对精度有比较严格要求，则先进行 OP 失调校准。详细操作参考第 22 章节 OP 模块）
- 配置输入通道选择 OPOUT，选择 ADC 时钟，选择触发源，
- 配置 ADC 中断，启动 ADC
- 启动 ADC 进行转换（软件启动或者外部触发）
- ADC 转换结束后（查询 ADCF 或者等待中断），进行 ADC 结果读取。

21.8 ADC 寄存器

ADCCON0 - ADC 控制寄存器 0

地址: E8H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCON0	ADCF	ADCS	--	ADCHS[4:0]				
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

位号	符号	描述
7	ADCF	ADC 中断标志 当 AD 转换完成, 该位置 1。如果中断允许, 可启动中断服务。可读取到当前 AD 转换结果。该位为 1 时无法开始新一轮转换, 需要软件清零。
6	ADCS	A/D 转换软件启动位 该位置 1 启动 AD 转换。在 AD 转换过程中该位保持为 1, 当转换结束硬件自动清零。这意味着写入 ADCS 的值和读出的不一定相符 写: 0 = 无动作 1 = 开始 AD 转换 读: 0 = ADC 模块空闲状态 1 = ADC 模块工作中
4:0	ADCHS [4:0]	A/D 转换通道选择 该位用于选择 ADC 转换通道。 00000 = AIN0 01010 = AIN10 00001 = AIN1 01011 = AIN11 00010 = AIN2 01100 = AIN12 00011 = AIN3 01101 = AIN13 00100 = AIN4 01110 = AIN14 00101 = AIN5 01111 = AIN15 00110 = AIN6 10000 = VBGR_ADC 00111 = AIN7 10001 = 0.25VDD 01000 = AIN8 10010 = VREF 01001 = AIN9 10011 = TS_OUT (温度传感器) 10100 = OPOUT (放大器输出)

ADCCON1 - ADC 控制寄存器 1

地址：D9H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCON1	--	--	SHS[1:0]		ETGTYP[1:0]		ADCEX	ADCEN
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

位号	符号	描述
5:4	SHS[1:0]	AD 信号采样维持时间 00: $2 \cdot T_{ad}$ 01: $4 \cdot T_{ad}$ 10: $8 \cdot T_{ad}$ 11: $12 \cdot T_{ad}$ 。
3:2	ETGTYP[1:0]	外部触发信号类型选择 当 ADCEX (ADCCON1.1) 置 1, 该位决定响应外部触发的类型。 00 = PWM0/2/4 或 ADCEX 脚的下降沿 01 = PWM0/2/4 或 ADCEX 脚的上升沿 10 = 一个 PWM 周期的中点 11 = 一个 PWM 周期的终点 注: PWM 周期中点或终点触发仅适用于中心对齐模式的 PWM 输出。
1	ADCEX	ADC 触发启动信号选择位 该位决定启动 ADC 的触发条件 0 = 当软件设定 ADCS 位, 启动 AD 转换 1 = 当软件设定 ADCS 位后, 还需要外部触发信号触发才启动。
0	ADCEN	ADC 使能位 0 = ADC 转换电路关闭 1 = ADC 转换电路打开

ADCCON2 - ADC 控制寄存器 2

地址：DAH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCON2	ADFBEN	ADCMPOP	ADCMPEM	ADCMPO	ADCKS [2:0]			ADCDLY8
R/W	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

位号	符号	描述
7	ADFBEN	ADC 比较结果响应故障刹车使能 0 = 禁止 1 = ADC 触发故障刹车功能使能。一旦比较结果 ADCMPO 为 1，触发故障刹车模块。即符合 PWM 故障刹车输出值后，硬件将清除 PWMRUN(PWMCON0.7)，并终止 PWM 输出。
6	ADCMPOP	ADC 比较器输出极性选择位 0 = 若 ADCR[11:0]大于或等于 ADCMP[11:0]， ADCMPO 为 1 1 = 若 ADCR[11:0]小于 ADCMP[11:0]， ADCMPO 为 1
5	ADCMPEM	ADC 结果比较使能位 0 = ADC 结果比较功能关闭 1 = ADC 结果比较功能打开
4	ADCMPO	ADC 比较结果输出位 该位输出 ACMPOP 设定比较输出的结果。每次 AD 转换结束都会更新输出。
3:1	ADCKS [2:0]	ADC 时钟分频 000 = Fsys/2 100 = Fsys/32 001 = Fsys/4 101 = Fsys/64 010 = Fsys/8 110 = Fsys/128 011 = Fsys/16 111 = Fsys/256
0	ADCDLY8	ADC 外部触发延时计数器数值第 8 位 详见 ADCDLY 寄存器描述

ADCCON3 - ADC 控制寄存器 3

地址：DBH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCON3	--	--	ETGSEL[1:0]		VREFEN	BGREN	TSEN	VDDEN
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	1	0	0	1

位号	符号	描述
5:4	ETGSEL[1:0]	外部触发源选择 当 ADCEX 为 1，该位选择外部触发 ADC 的来源 00 = PWM0 01 = PWM2 10 = PWM4 11 = ADCET（外部引脚）
3	VREFEN	ADC 基准电压源启动 0 = VREF 使能 1 = VREF 禁止
2	BGREN	ADC BGR 使能位 0 = ADC BGR 关闭 1 = ADC BGR 使能
1	TSEN	温度传感器使能 0 = 禁止 1 = 使能
0	VDDEN	VDD 1/4，3/4 电压同时使能 0 = 使能 1 = 禁止

ADOC - ADC 修调寄存器

地址: D5H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADOC	CALI	SIGN	VOF[2:0]			VREFSEL[1:0]		VREFS
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

位号	符号	描述
7	CALI	AD 精度校正使能位 0 = 禁止校正 1 = 使能校正
6	SIGN	校正电压的极性 0 = 负极性 1 = 正极性
5:3	VOF[2:0]	补偿电压位 000: 0LSB 001: 1LSB 010: 3LSB 011: 5LSB 100: 7LSB 101: 9LSB 110: 11LSB 111: 13LSB
2:1	VREFSEL[1:0]	ADC 内部基准电压选择 (ACMP 的 VREF 选择位) 00 = VDD; 01 = 4V 10 = 3V 11 = 2V
0	VREFS	ADC 的基准电压源选择 0 = ADC 基准电压源为内部 VREF(2V,3V,4V,VDD) 1 = ADC 基准电压源由外部管脚 (P3.0) 输入

ADCDLY - ADC 延时寄存器

地址：DCH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCDLY	ADCDLY[7:0]							
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

位号	符号	描述
7:0	ADCDLY[7:0]	ADC 外部触发启动延迟计数器低位 该8位寄存器与ADCCON2.0组成 9位计数器,用于在外部触发启动ADC之前加入一段延迟。延迟计数结束在开始ADC转换 外部延迟时间 = $ADCDLY \times T_{sys}$ 注：该延迟仅当 ADCEX 置 1 时有效。如果启用 PWM 输出触发ADC功能，在 PWM 运行过程中不得更改 ADCDLY 计数值。

ADCRH - ADC 转换结果高位寄存器

地址：D2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCRH	ADCRH[7:0]							
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	ADCRH[7:0]	ADC 转换结果高位 ADC 转换结果高 8 位

ADCRL - ADC 转换结果高位寄存器

地址：D1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCRL	--	--	--	--	ADCRL[3:0]			
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
3:0	ADCRL[3:0]	ADC 转换结果低 ADC 转换结果低 4 位

ADCMPH - ADC 比较值高位寄存器

地址：D4H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPH	ADMPH[7:0]							
R/W	R	R	R	R	R	R	R	R
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
7:0	ADCMPH[7:0]	ADC 比较值高位 8 位

ADCMPL - ADC 比较值低位寄存器

地址：D3H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPL	--	--	--	--	ADCMPH[3:0]			
R/W	--	--	--	--	R	R	R	R
复位值	0	0	0	0	0	0	0	0

位号	符号	描述
3:0	ADCMPH[3:0]	ADC 比较值低 4 位

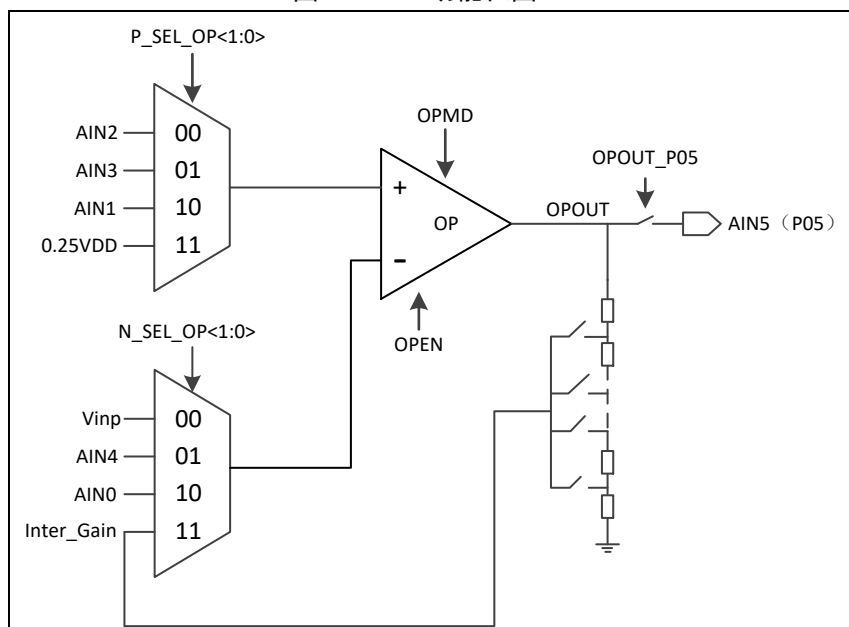
22 OPA 模块

22.1 运放特性

- 运放正输入可以选择 AIN3 (P03)、AIN2 (P02)、AIN1 (P01) 以及 0.25VDD
- 运放负输入可以选择端口 AIN4 (P04)、端口 AIN0 (P00)、修调通道，内部放大网络
- 运放输出直接送至 ADC 或者比较器正负两级以及端口 AIN5 (P05)
- OP 可以通过软件配置内部增益放大倍数
- 软件支持失调电压修调

22.2 运放结构

图 22-1 OP 功能框图



22.3 运放失调电压修调

运放每次使用需要进行失调电压修调，用户可以通过以下步骤进行失调电压修调：

- ◆ 设置 $P_SEL_OP[1:0]=0x11$ ，选择 0.25VDD，设置 $N_SEL_OP[1:0]=0x00$ ，使 OP 正负输入端相连，配置 $OPGS[3:0]$ 位为 1111 关闭内部电阻网络。
- ◆ 通过置位 ADCCON3 寄存器的 BGREN 位和 VDDEN 位，使能 BGR 输出和 VDD 相关输出电压。设置 $OPEN=1$ ，设置 $OPOD=1$ ，进入修调。
- ◆ 对寄存器 OPCOFG 的 $OPDTS[7:0]$ 里面写入初始数据 0x00，读 OPOD，若 OPOD 为 1，则退出修调模式；

- ◆ 若 OPOD 为 0，则将 OPDTS[7:0]的数据读出并加 1 后再次写入寄存器。读取 OPOD 值，若 OPOD 为 1，则退出修调。若 OPOD 为 0，则继续执行本步骤进行修调。
 - ◆ 若一直到 OPDTS[7:0]=0xFF，OPOD 位依然是 0，则保持 OPDTS 的值，清 OPMD 位退出修调模式。
- **注意：**往 OPCOFG 寄存器写入数据后，需要延迟至少 10us 时间等待修调完成后再去读 OPOD 的值。

22.4 运放使用步骤

- ◆ 配置相应端口为模拟输入
- ◆ 使能 ADCCON3 寄存器的 BGREN 位。
- ◆ 使用 22.3 的步骤进行运放修调（也可以补修调）
- ◆ 如果使用内部放大网络，则通过 OPCON1 的为 N_SEL_OP[1:0]=11，同时配置 OPGS[3:0] 选择放大系数。
- ◆ 选择运放模式，并使能 OP 模块

22.5 运放寄存器

OPCON0 - 运放控制寄存器 0

XSFR 地址: F0B0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPCON0	OPEN	OPMD	--	OPOD	OPGS<3:0>			
R/W	R/W	R/W	--	R	R/W			
复位值	0	0	0	0	1	1	1	1

位号	符号	描述
7	OPEN	OP 信号使能, 0: 不使能 1: OP 使能
6	OPMD	OP 模式选择: 0: 运放模式 1: 进入比较器模式或者失调修调模式
4	OPOD	OP 数字输出, 用于 OP 修调时判断是否修调完成
3:0	OPGS[3:0]	OP 放大系数 000x: 有内部增益系数确定 x1 0010: 有内部增益系数确定 x2 0011: 有内部增益系数确定 x4 0100: 有内部增益系数确定 x6 0101: 有内部增益系数确定 x8 0110: 有内部增益系数确定 x12 0111: 有内部增益系数确定 x16 1000: 有内部增益系数确定 x24 1001: 有内部增益系数确定 x32 1010: 有内部增益系数确定 x48 1011: 有内部增益系数确定 x64 1100: 有内部增益系数确定 x96 1101: 有内部增益系数确定 x128 111x: 关闭内部电阻网络, 用于运放负端使用外部输入, 或作为比较器模式。

OPCOFG -- 修调寄存器

XSFR 地址: F0B1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPCOFG	OPDTS[7:0]							
R/W	R/W							
复位值	0	1	1	1	1	1	1	1

位号	符号	描述
7:0	OPDTS[7:0]	OP 失调电压调节位

OPCON1 - 运放控制寄存器 1

XSFR 地址: F0B2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPCON1	P_SEL_OP[1:0]		N_SEL_OP[1:0]		OP_OUT	--	--	--
R/W	R/W		R/W		R/W	--	--	--
复位值	0	0	0	0	0	--	--	--

位号	符号	描述
7:6	P_SEL_OP[1:0]	放大器正端通道选择: 00: 放大器正端输入电压选择 AIN2 (P02) 01: 放大器正端输入电压选择 AIN3 (P03) 10: 放大器正端输入电压选择 AIN1 (P01) 11: 放大器正端输入电压选择 0.25VDD
5:4	N_SEL_OP[1:0]	放大器负端通道选择: 00: 放大器负端输入电压选择比较器正端输入电压 01: 放大器负端输入电压选择 AIN4 (P04) 10: 放大器负端输入电压选择 AIN0 (P00) 11: 选择内部电阻放大网络
3	OPOUT_P05	放大器端口输出使能 0: OPOUT 不输出到芯片端口 1: OPOUT 通过 P05 口输出

23 ACMP 比较器模块

23.1 概述

JSC68F016 为用户提供了—个可灵活配置的 ACMP 比较器模块，可按照比较器的配置适用于不同的应用场合。当正端电压大于负端电压时，比较器输出逻辑 1，反之输出 0，也可以通过输出极性选择位进行改变。当比较器输出值发生改变时，比较器都可通过配置产生中断。比较器同时可以实现低压监测和复位功能（BOD），通过配置相应的寄存器可以进行实时电压监测。

23.2 比较器特性

- ◆ 模拟输入电压范围：0 V~（VDD-1）V
- ◆ 比较器正极支持 12 个外部输入端口、片内 BGR、内部 VREF、VDD 输入和 OPOUT，同时支持 VREF 与 VDD 进行 32 阶级分压后输入。
- ◆ 比较器负极支持 14 个外部端，片内 BGR（1.2V）输入和 OPOUT
- ◆ 支持电压比较功能，可进行系统电源 VDD 的实时监测，监测结果可以使系统产生复位
- ◆ 比较器具有三种通过软件可配置的中断触发方式：上升沿触发/下降沿触发/双沿触发
- ◆ 比较器的输出可以作为 PWM 的刹车输入信号
- ◆ 支持在超低功耗模式下工作，比较器的中断输出可以将芯片从超低功耗模式下唤醒
- ◆ 比较器可以通过软件配置滤波时间以增强芯片的抗干扰能力

23.3 比较器框图

图 23-1 比较器结构图

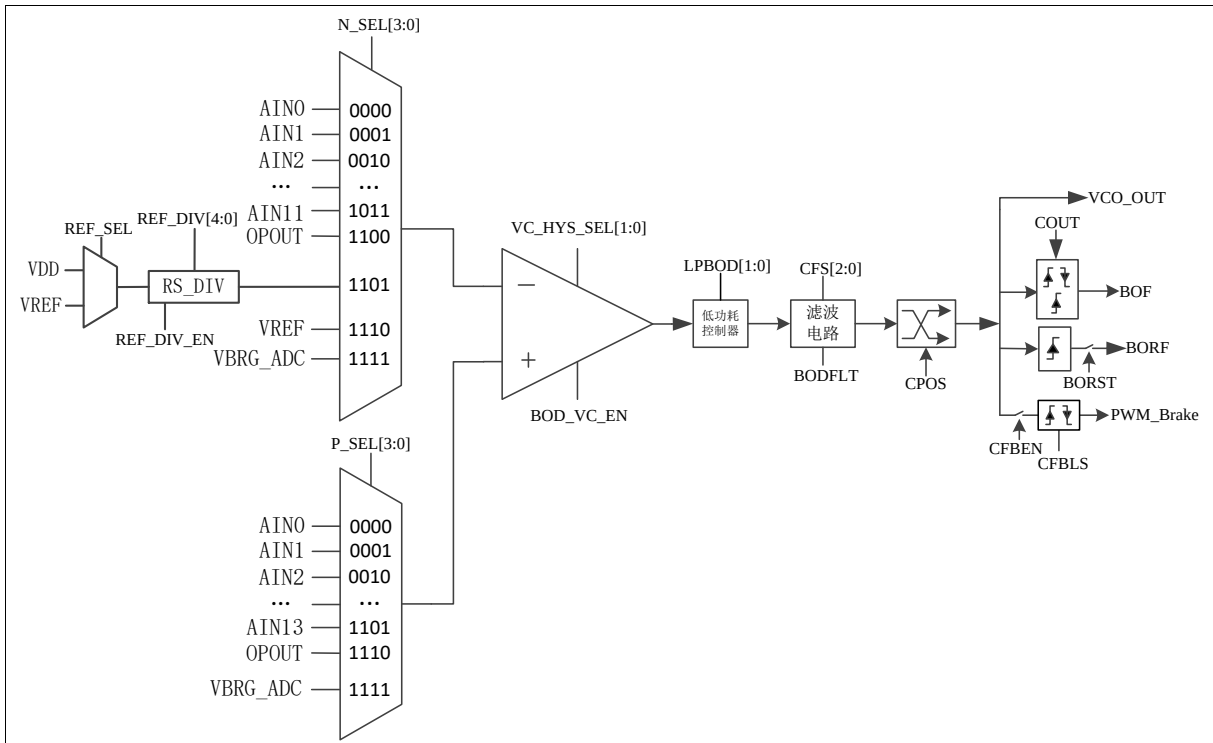
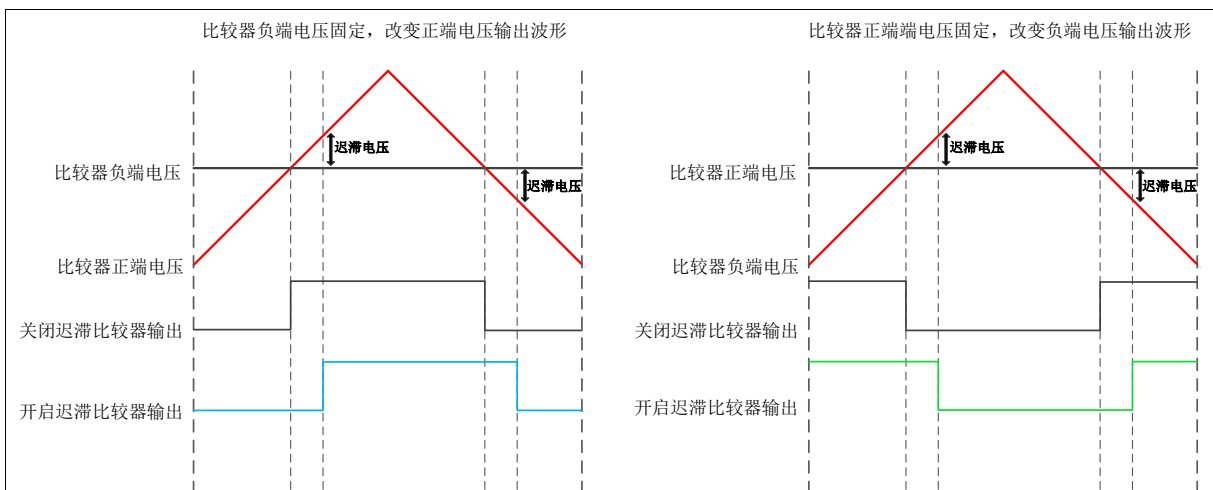


图 23-2 比较器迟滞电压



23.4 通用比较器功能

可以通过如下步骤进行比较器配置

- 配置相应的外部 GPIO 端口为模拟输入
- 通过寄存器 VCCFG 的 N_SEL[3:0] 进行比较器负端通道配置。负端通道总共有 12 个外部模拟输入通道，1 个电阻分压通道、1 个 VREF 通道、1 个 VBRG 电压比较通道，和 OP 放大器输入通道
- 通过寄存器 VCCFG 的 P_SEL[3:0] 进行比较器正端通道配置。正端通道总共有 14 个外部模拟输入通道，1 个 VBRG 电压比较通道和 OP 放大器通道
- 如果选择正端外部输入与负端固定电压进行比较，则需要在 VCCON 寄存器中使能 REF_DIVE_EN 位，并通过 REF_DIV 位进行配置选择电阻的分压源：VDD 或者 VREF。通过 BODCON0 寄存器的 REF_DIV[4:2] 位和 VCCON 寄存器的 REF_DIV[1:0] 位，进行电阻分压配置。
- 如果选择 VBRG 通道，则需要使能 ADCCON3 寄存器的 BGREN 位；如果选择 VREF/VDD 电阻分压，则需要使能 ADCCON3 寄存器的 VDDEN 或者 VREFEN 位；如果选择 VREF 通道则需要使能 ADCCON3 寄存器的 VREFEN 位。
- 配置 VC_HYS_SEL[1:0] 位，选择比较器迟滞电压大小
- 通过配置 CFS[2:0] 进行滤波电路配置，配置 CPOS 选择不同的输出极性
- 如果需要对 PWM 故障刹车，则需要配置 CFBLIS 位进行触发沿选择和 CFBEN 位进行使能选择
- 如果需要使用中断，则需要配置中断使能位 EBOD 并配置 EA 位进行使能中断
- 配置比较器使能位 BOD_VC_EN 使能比较器

表 23-1 电阻分压系数表

BODCON0[6:4] VCCON[5:4]	000	001	010	011	100	101	110	111
00	1/32	5/32	9/32	13/32	17/32	21/32	25/32	29/32
01	2/32	6/32	10/32	14/32	18/32	22/32	26/32	30/32
10	3/32	7/32	11/32	15/32	19/32	23/32	27/32	31/32
11	4/32	8/32	12/32	16/32	20/32	24/32	28/32	32/32

23.5 欠压监测功能

欠压检测是另一种电源监控功能，欠压检测电路是用于监测运行期间 VDD 电平。欠压检测电路跟比较器电路共用一个比较电路，当用作欠压检测功能时，有 32 个可编程的欠压检测触发电平，以适用于宽电压应用，通过 BODCON0[6:4]和 VCCON[5:4]来改变 BOD 电平。当 VDD 下降到所选择的欠压检测触发电平(VBOD)，欠压检测逻辑将复位 MCU 或请求欠压检测中断。用户可结合不同应用决定设备是欠压复位还是产生欠压中断。注意，上电后在软件打开 BOD 后需要等待 2 到 3 个 LIRC 时钟才能正常工作。

可以通过如下步骤进行比较器配置使比较器作为欠压监测使用

- 配置相应的 GPIO 端口为模拟输入
- 配置寄存器 VCCFG 的 N_SEL[3:0] = 1101 选择负端输入通道位电阻分压通道，配置 VCCON 寄存器的 REF_SEL 位为 0 选择 VDD 为参考电压。
- 配置 VCCFG 寄存器的 P_SEL[3:0] = 1111 选择正端比较电压为内部基准电压 1.2V (VBRG)，同时设置 ADCCON3 寄存器的 BGREN 位，使能 VBRG 电压。
- 通过 BODCON0 寄存器的 REF_DIV[4:2]位和 VCCON 寄存器的 REF_DIV[1:0]位，对欠压监测的电压进行配置，具体配置参数参考表 23-1
- 通过 VCCON 进行迟滞电压配置，建议配置 VC_HYS_SEL[1:0]为 11。
- 如果需要对输入 VDD 进行滤波，则可以使能 BODCON1 寄存器中的 BODFLT 位。并在 VCOUTCON 寄存器中设置相应的滤波时间。如果需要在低功耗模式下 BOD 继续运行，则需要设置 BODCON1 寄存器中的 LPBOD[1:0]相应功能位
- 设置 VCOUTCON 寄存器中的 CPOS 和 COUT 位，选择 BOD 的输出极性和中断方式。
- 设置 EBOD 位，使能 BOD 中断
- 通过 BODCON0 中的 BOD_VC_EN 位，使能或关闭欠压检测电路。

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

*注意：在 BOD 使能后，如果要修改其配置，必须先清 BODCON0.7 位，关闭 BOD 模块，配置完成后再开启 BOD 模块，

欠压检测功能支持低功耗 BOD 模式，通过设置 LPBOD[1:0]位，欠压检测电路可以周期性的检测电源电压，，同时对于噪声敏感系统，欠压检测电路还可以开启滤波器进行电源噪声滤波，滤波电路上电默认开启。

表 23-2 BOD 分压选择表

迟滞电压设置 VC_HYS_SEL[1:0] = 11, VDD 为参考电压分压

REF_DIV[4:0]	10101	10100	10011	10010	10001	10000	01111
电源由高到低	1.72	1.85	1.947	2.05	2.16	2.29	2.45
电源由低到高	1.87	1.95	2.05	2.15	2.27	2.41	2.56
REF_DIV[4:0]	01110	01101	01100	01011	01010	01001	01000
电源由高到低	2.59	2.78	2.99	3.25	3.54	3.89	4.33
电源由低到高	2.73	2.92	3.15	3.41	3.72	4.09	4.54

23.6 比较器寄存器

BODCON0-欠压检测及比较器控制寄存器 0 (TA 保护)

地址: A6H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BODCON0	BOD_VC_EN	REF_DIV[4:2]			BOF	BORST	BORF	BOS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
复位值	0	0	1	0	0	0	0	0

位号	符号	描述
7	BOD_VC_EN	欠压检测或者比较器使能 0 = 禁用欠压检测与比较器电路 1 = 使能欠压检测与比较器电路 注意: 在开启该功能后需要 2 到 3 个 LIRC 时钟 BOD 才能正常工作
6:4	REF_DIV[4:2]	配置欠压侦测电压或者比较器比较电压选择位 010: VBOD 生效电压 4.3V 011: VBOD 生效电压 3V 100: VBOD 生效电压 2.30V 101: VBOD 生效电压 1.85V 若需其他点, 需联合配置 REF_DIV[1:0], 具体配置参考表 23-2 BOD 分压选择表
3	BOF	比较器结果或者欠压中断标志 当 VDD 下降到 VBOD 以下或 VDD 上升到 VBOD 以上时, 该标志由硬件设置为逻辑 1。如果 EBOD (EIE.5) 和 EA (IE.7) 都置位, 将请求欠压检测中断。该位必须由软件清零。
2	BORST	欠压检测复位使能 该位决定在电源电压跌到以 VBOD 下时是否产生欠压检测复位 0 = 禁用欠压检测复位 1 = 使能欠压检测复位
1	BORF	欠压检测复位标志 当 MCU 发生欠压复位时, 该位被硬件值 1。记得通过软件清除该位。
0	BOS	欠压状态或者比较器结果标志(只读) 在用作欠压检测或者比较器中 当 CPOS=1 0 = 比较器正端输入大于负端输入 1 = 比较器正端输入小于负端输入 当 CPOS=0 0 = 比较器正端输入小于负端输入 1 = 比较器正端输入大于负端输入

BODCON1-欠压检测及比较器控制寄存器 1（TA 保护）

地址：AEH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BODCON1	--	--	--	--	--	LPBOD[1:0]		BODFLT
R/W	--	--	--	--	--	R/W	R/W	R/W
复位值	--	--	--	--	--	0	0	1

位号	符号	描述
2:1	LPBOD[1:0]	低功率 BOD 或者比较器使能 00 = BOD 正常模式，BOD 电路总是开启 01 = BOD 低功耗模式 1，每隔 1ms 周期性开启 BOD 电路 10 = BOD 低功耗模式 2，每隔 4ms 周期性开启 BOD 电路 11 = BOD 低功耗模式 3，每隔 16ms 周期性开启 BOD 电路
0	BODFLT	BOD 或者比较器滤波器控制 0 = 禁用 BOD 或者比较器滤波器 1 = 使能 BOD 或者比较器滤波器(上电默认开启) 滤波数值具体由 VCOUTCON 的 CFS[2:0]进行设置，可以通过 CFS 进行 8 种滤波时间选择

VCCON-比较器配置寄存器

XSFR 地址：F0D0H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
VCCON	--	--	REF_DIV[1:0]		REF_SEL	REF_DIV_EN	VC_HYS_SEL[1:0]	
R/W	--	--	R/W		R/W	R/W	R/W	
复位值	--	--	0	0	0	0	1	1

位号	符号	描述
5:4	REF_DIV[1:0]	细修调，每格修调 1/32 REF
3	REF_SEL	比较器内部参考电压选择 0 = VDD 1 = 内部生成的 VREF(2V、3V、4V 可选)
2	REF_DIV_EN	电阻分压使能 0 = 不使能 1 = 使能
1:0	VC_HYS_SEL[1:0]	VC 迟滞电压选择: 00:没有迟滞 01:迟滞电压大约 20mV 10:迟滞电压大约 30mV 11:迟滞电压大约 60mV

VCCFG-比较器通道选择寄存器

XSFR 地址: F0D1H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
VCCFG	N_SEL[3:0]				P_SEL[3:0]			
R/W	R/W				R/W			
复位值	1	1	0	1	1	1	1	1

位号	符号	描述
7:4	N_SEL[3:0]	电压比较器“-”端输入选择 0000 = select channel 0 input P00 (ACMP0) 0001 = select channel 1 input P01 (ACMP1) 0010 = select channel 2 input P02 (ACMP2) 0011 = select channel 3 input P03 (ACMP3) 0100 = select channel 4 input P04 (ACMP4) 0101 = select channel 5 input PA5 (ACMP5) 0110 = select channel 6 input P06 (ACMP6) 0111 = select channel 7 input P07 (ACMP7) 1000 = select channel 8 input P10 (ACMP8) 1001 = select channel 9 input P11 (ACMP9) 1010 = select channel 10 input P12 (ACMP10) 1011 = select channel 10 input P13 (ACMP11) 1100 = OPOUT 1101 = 电阻分压输出电压 (上电默认 BOD 功能) 1110 = VREF 电压 (需要使能 VREF) 1111= 内部基准 1.2V 输出电压
3:0	P_SEL[3:0]	电压比较器“+”端输入选择 0000 = select channel 0 input P00 (ACMP0) 0001 = select channel 1 input P01 (ACMP1) 0010 = select channel 2 input P02 (ACMP2) 0011 = select channel 3 input P03 (ACMP3) 0100 = select channel 4 input P04 (ACMP4) 0101 = select channel 5 input PA5 (ACMP5) 0110 = select channel 6 input P06 (ACMP6) 0111 = select channel 7 input P07 (ACMP7) 1000 = select channel 8 input P10 (ACMP8) 1001 = select channel 9 input P11 (ACMP9) 1010 = select channel 10 input P12 (ACMP10) 1011 = select channel 11 input P13 (ACMP11) 1100 = select channel 12 input P14 (ACMP12) 1101 = select channel 13 input P15 (ACMP13) 1110 = OPOUT 1111 = 内部基准 1.2V 输出电压 (上电默认 BOD 功能)

VCOUTCON-比较器输出控制寄存器

XSFR 地址: F0D2H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
VCOUTCON	CFBEN	CFBLS	CPOS	COUT	--	CFS[2:0]		
R/W	R/W	R/W	R/W	R/W	--	R/W		
复位值	0	0	0	1	--	0	1	1

位号	符号	描述
7	CFBEN	比较器输出控制 PWM 刹车使能位 0 = 禁止 1 = 使能;
6	CFBLS ⁽¹⁾	比较器输出控制 PWM 刹车边沿选择位 0 = 上升沿。 1 = 下降沿;
5	CPOS ⁽¹⁾	比较器输出极性选择位 0 = 正向输出 1 = 反向输出;
4	COUT	比较器输出中断沿选择位 0 = 双沿中断。 1 = 上升沿中断;
2:0	CFS[2:0]	比较器输出滤波时间选择位（系统时钟选择 HIRC、ECLK 并且 BOD 没有工作在低功率模式） 000= 4*Tsys 001= 8*Tsys 010= 16*Tsys 011= 32*Tsys（默认） 100= 64*Tsys 101= 128*Tsys 110= 256*Tsys 111= 512*Tsys

*注意：（1）在用比较器的输出进行 EPWM 模块的故障刹车使能时，需要先进行 CPOS、CFBLS 的设置，然后使能故障刹车中断位 CFBEN。在使能了比较器后，再进行 EPWM 模块，保证刹车信号的正常输入。

24 FLASH 操作 (IAP)

修改 FLASH 数据通常需要很长时间，不像 RAM 那样可以实时操作。而且擦除、编程或读取 FLASH 数据需要遵循相当复杂的时序步骤。JSC68F016 提供方便 FLASH 编程方式，可以帮助用户通过 IAP 方式，重新编程 FLASH 内容。IAP 就是通过软件实现在线电擦除和编程的方法。

通过设置 IAPEN (CHPCON.0 受 TA 保护) 使能 IAP，并且设置 IAPUEN 寄存器的相应位，使能需要升级的 FLASH 区域 (CONFIG、LDROM、APROM)，用户将 16 位操作地址写入 IAPAH 和 IAPAL，数据写入 IAPFD，命令写入 IAPCN。然后通过设置触发位 IAPGO(IAPTRG.0)，去执行 IAP。注意：IAPTRG 也受 TA 保护。擦除和编程时间是内部控制的，与工作电压和频率无关。通常页擦除时间为 110ms，字节编程时间是 70 μ s。IAP 动作完成后，程序计数器继续运行之后的指令，IAPGO 位将自动清零。IAPFF (CHPCON.6) 是 IAP 错误标志，可以用来检查之前 IAP 操作成功与否。通过这些纯软件的设置，用户可以很方便对 FLASH 存储器进行擦除、编程和校验。

24.1 IAP 命令

JSC68F016 通过 IAP 可操作 APROM、LDROM 或 CONFIG。IAP 操作模式和编程区域是由 IAP 控制寄存器 IAPCN 决定的。

表 24-1 IAP 模式和命令代码

IAP 模式		IAPCN				IAPA[13:0]	IAPFD[7:0]
		IAPB[1:0]	FOEN	FCEN	FCTRL[3:0]	{IAPAH,IAPAL}	
读公司 ID(CID)		XX	0	0	1011	X	B0H
读 器 件 ID	DID_L	XX	0	0	1100	0000H	43H
	DID_H					0001H	1FH
读 96 位 UID		XX	0	0	0100	0000H ~ 000BH	数据读出
字节读	APROM	00	0	0	0000	地址写入	数据读出
	LDROM	01					
	CONFIG	11					
字节编程	APROM	00	1	0	0001	地址写入	数据写入
	LDROM	01					
	CONFIG	11					
扇区擦除	APROM	00	1	0	0010	扇区地址	XX
	LDROM	01					
	CONFIG	11					
页擦除	APROM	00	1	0	0110	页地址	XX
	LDROM	01					
芯片擦除	APROM	00	1	0	1010	XX	XX
	APROM+ CONFIG	11					

注：芯片擦除功能仅在 LDROM 空间运行程序时有效，这时选 APROM 可以擦除 APROM 的数据，LDROM 和 CONFIG 的数据不变；选 CONFIG 可以擦除 APROM 和 CONFIG 的数据，LDROM 的数据不变。

注：芯片读保护(加密)置位时，扇区擦除功能擦 CONFIG 无效；字节读功能读 APROM 和 LDROM 时，读出的值为 0，读 CONFIG 正常。通过 MOVC 指令读本空间的数据则不受加密位影响。

24.2 IAP 用户指南

IAP 可以方便用户更新 FLASH 内容，不过用户必须遵循一定规则，以确保 IAP 正确执行，否则可能引起不确定的结果，甚至损坏器件。此外，下文对于正确执行 IAP 有很好建议。

(1) 没有 IAP 操作时，用户必须清 IAPEN 位。可以防止系统意外触发 IAP。此外，IAP 需要使用内部 HIRC 振荡器。如果选择外部时钟源，禁止 IAP 将停止内部 HIRC，可以达到省电的目的。

注：写 IAPEN 受 TA 保护。

(2) 当 LOCK 位被激活，IAP 读、写或擦除仍然有效。

*在进行 IAP 时，如果中断打开应该临时清除 EA 位

*擦除或编程的页不能是当前代码执行的页。否则会出现不可预计程序动作，甚至破坏存储的数据

*擦除或编程的时候必须确保 VDD 电压不低于 2.5V，否则将出现不可预测错误

24.3 使用 FLASH 存储器作为数据存储

在一般应用中，有时需要一些数据在断电情况下不能丢失，以使用户读回或更新，作为系统控制的参数。JSC68F016 支持 IAP 功能并且存储在 FLASH 中的字节都可以用 MOVC 指令读取，所以很适合作为非易失数据存储器。

24.4 唯一 ID 96 位序列号

出厂前，每颗 JSC68F016 都会预烧一个 96 位的序列号，用以确保该芯片的唯一性，这个唯一代码被称为序列号 UID (Unique Code)。用户获得序列号的唯一方式是通过 IAP 指令读取，详见 IAP 命令。

24.5 在线系统编程 (ISP)

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

用户可以开发自己的引导代码放在 LDR0M 中。LDR0M 最大为 4KB。用户开发的引导代码可以通过并行烧录器或是在线电路编程器 (ICP) 下载到 LDR0M 中去。

一般来说，ISP 是 PC 与 MCU 之间进行通讯。PC 通过串口传输新的用户代码给 MCU。然后引导代码接收这些数据，将这些数据通过 IAP 命令编程到用户代码区域。

24.6 IAP 指令及寄存器

CHPCON-芯片控制寄存器（受 TA 保护）

地址：9EH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CHPCON	--	IAPFF	--	--	--	CS	BS	IAPEN
R/W	--	R/W	--	--	--	R/W	R/W	R/W
复位值	--	xx	--	--	--	x	x	0

位号	符号	描述
6	IAPFF	IAP 错误标志:在 IAPGO 置位后置位该位 1: 访问地址超过其大小的区域 2: IAPCN 命令无效 3: IAP 擦除或编程更新没有使能的区域 4: 当 BOIAP(CONFIG2.5)为 1、 BODEN (BODCON0.7)为 1 以及 BORST(BODCON0.2) 为 0 时, 擦除或编程工作在 VBOD 下。 该位应该由软件清零
2	CS	复位后是否读取重新装载 CONFIG 0 = 复位后不重新装载 CONFIG 1 = 复位后重新装载 CONFIG
1	BS	启动选择, 复位后选择启动区 (启动时从 config 加载) 0 = 由 APROM 启动 1 = 由 LDROM 启动
0	IAPEN	IAP 使能 0 =禁用 IAP 功能 1 =使能 IAP 功能

IAPCN-IAP 控制寄存器

地址：96H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAPCN	IAPB[1:0]		FOEN	FCEN	FCTRL[3:0]			
R/W	R/W		R/W	R/W	R/W			
复位值	0	0	1	1	0	0	0	0

位号	符号	描述
7:6	IAPB[1:0]	IAP 控制 IAP 控制命令。详见表 24-1 IAP 模式和命令代码
5	FOEN	
4	FCEN	
3:0	FCTRL[3:0]	

IAPUEN-IAP 更新使能寄存器（受 TA 保护）

地址：9DH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAPUEN	--	--	--	--	--	CFUEN	LDUEN	APUEN
R/W	--	--	--	--	--	R/W	R/W	R/W
复位值	--	--	--	--	--	0	0	0

位号	符号	描述
2	CFUEN	CONFIG 更新使能 0 = 禁用 IAP 擦除或编程 CONFIG 1 = 使能 IAP 擦除或编程 CONFIG
1	LDUEN	LDR0M 更新使能 0 = 禁用 IAP 擦除或编程 LDR0M 1 = 使能 IAP 擦除或编程 LDR0M
0	APUEN	APROM 更新使能 0 = 禁用 IAP 擦除或编程 APROM 1 = 使能 IAP 擦除或编程 APROM

IAPTRG-IAP 触发（受 TA 保护）

地址：97H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAPTRG	--	--	--	--	--	--	--	IAPGO0
R/W	--	--	--	--	--	--	--	W
复位值	--	--	--	--	--	--	--	0

位号	符号	描述
0	IAPGO[0]	IAP 执行指令控制位 设置 IAPGO 位为 1 开始执行 IAP，执行完成后自动清零。 在执行之前请关闭中断。

IAPAH-IAP 地址高字节寄存器

地址：95H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAPAH	IAPA[15:8]							
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

位号	符号	描述
7:0	IAPA[15:8]	IAP 地址高字节 IAPAH 包含地址 IAPA[15:8]

IAPAL-IAP 地址低字节寄存器

地址：94H

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAPAL	IAPA[7:0]							
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

位号	符号	描述
7:0	IAPA[7:0]	IAP 地址高字节 IAPAH 包含地址 IAPA[7:0]

IAPFD-IAP 内存数据

地址：9CH

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IAPFD	IAPFD[7:0]							
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

位号	符号	描述
7:0	IAPFD[7:0]	AP 内存数据 该字节包含将要读或写进内存空间的数据，在编程模式下，用户需要在触发 IAP 之前写数据到 IAPFD，读/校验模式下，在 IAP 完成后从 IAPFD 读取数据

25 在线编程及 OCDS 调试

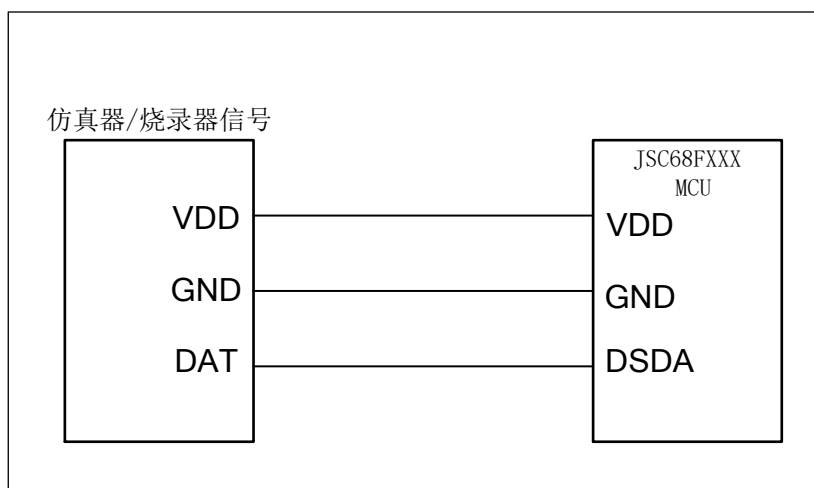
25.1 在线串行编程（ICP）

通过在线电路编程（ICP）编程 Flash，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 3 根线完成

- ◆ 电源线
- ◆ 接地线
- ◆ 数据线

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的固件或者定制固件烧写到单片机中

图 25-1 在线电路编程 ICP



25.2 OCDS 功能描述

JSC68F016 内嵌片上调试功能（OCDS），这为软件开发者提供了低成本调试方法，并且 JSC68F016 的每一种封装都适用。OCDS 具有完整的调试过程控制流程，包括 2 个硬件断点、单步运行、全速运行和非侵入命令的内存访问。OCDS 系统并不占用任何本地内存，也不和片上外设共享资源。

当 LOCK (CONFIG0.0) 为 0 时，OCD 才能有效。如果芯片已加密，OCDS 就不能工作。OCDS 系统使用单线串行接口：OCDDA，让目标设备和调试控制器建立通讯。OCDDA 是输入/输出引脚，调试时用作数据传输。JSC68F016 的 OCDS 和 ICP 功能是共享这个引脚。

JSC68F016 使用 OCDDA 与 OCDS 系统交互。在设计系统用到 OCDS 时，必须考虑下面一些限制条件：

1. 所有外部复位源必须断开

2. 所有与 OCDDA 连接的外围器件必须断开

25.2 OCDS 限制条件

由于 JSC68F016 功能比较丰富，而引脚比较有限，所以一个引脚上可能多个功能。使用 OCD 系统时肯定会牺牲一些功能，主要有以下一些限制条件：

1. OCDDA 与 P1.6 共享一个引脚，因此该引脚 I/O 功能或其他功能都不能使用
2. 当系统处在空闲或掉电模式时，因为部分外设时钟已经停止，所以任何访问可能无效的。读访问可能返回一个 无用的数据，写访问可能不成功。
3. 不能关闭 HIRC，因为 OCD 需要这个时钟监视内部工作状态。在调试模式下，关闭 HIRC 的指令将不起作用，CPU 进入掉电模式时 HIRC 会继续运行

JSC68F016 的 OCD 系统还有另一个限制，就是正在运行用户程序时不能执行非侵入命令。非侵入命令可以用调试器访问 MCU 的存储单元、状态或控制寄存器，一个读或写控制寄存器必须在 MCU 停止的条件下进行，产生停止条件是在与硬件断点匹配后或单步运行后。

25.3 OCD 寄存器

CONFIG0-系统配置寄存器 0

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CONFIG0	CBS	--	OCDPWM	OCDEN	--	RSTP20	--	LOCK
R/W	R/W	--	R/W	R/W	--	R/W	--	R/W
复位值	0	--	0	0	--	0	--	0

位号	符号	描述
4	OCDEN	OCDS 硬件故障复位使能 0 = 使能 OCDS 故障复位 1 = 禁用 OCDS 故障复位
0	LOCK	芯片加密使能位 0 = 芯片不加密。Flash 存储器不加密，用户可通过硬件编程器/ICP 编程器读取 FLASH 的值。 1 = 芯片加密。全芯片 FLASH 区域加密，通过硬件编程器/ICP 编程器读取 FLASH 的值，读回全部为 (FFH)，对 FLASH 进行编程无效。

26 系统配置 CONFIG

26.1 Config 信息

JSC68F016 具有几个硬件配置字节，设置这些配置位配置硬件选项，如安全位，系统时钟位等等。这些硬件配置位可通过编程器/烧录器或 ICP、IAP 来配置。有些特定的配置位定义的功能，也可以通过特殊寄存器位重新配置。因此，需要加载这些配置位到相应的寄存器位。这些加载发生在复位之后。

CONFIG0-系统配置寄存器 0

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CONFIG0	CBS	--	OCDPWM	OCDEN	--	RSTP20	--	LOCK
R/W	R/W	--	R/W	R/W	--	R/W	--	R/W
复位值	0	--	0	1	--	0	--	0

位号	符号	描述
7	CBS	配置启动选择位 该位定义，除软件复位外的所有复位后，MCU 选择 FLASH 区启动。 0 = 除软件复位外的所有复位后，MCU 从 APROM 启动 1 = 除软件复位外的所有复位后，MCU 从 LDROM 启动
5	OCDPWM	片上调试 OCDS 占用模式下，PWM 输出状态 0 = PWM 正常输出 1 = PWM 输出脚为三态模式
4	OCDEN	OCDS 硬件故障复位使能 0 = 允许 OCDS 故障复位 1 = 禁用 OCDS 故障复位
2	RSTP20	复位脚禁用位 0 = P2.0 复位功能关闭，管脚用作普通 GPIO P2.0 1 = P2.0 复位功能使能，管脚用作外部复位脚
0	LOCK	芯片加密使能位 0 = 芯片不加密。Flash 存储器不加密，用户可通过硬件编程器/ICP 编程器读取 FLASH 的值。 1 = 芯片加密。全芯片 FLASH 区域加密，通过硬件编程器/ICP 编程器读取 FLASH 的值，读回全部为 (FFH)，对 FLASH 进行编程无效。

CONFIG1-系统配置寄存器 1

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CONFIG1	--	--	--	--	--	LDSIZE[2:0]		
R/W	--	--	--	--	--	R/W	R/W	R/W
复位值	--	--	--	--	--	0	0	0

位号	符号	描述
2:0	LDSIZE[2:0]	LDROM 和 APROM 区域选择 0xx = 无 LDROM, APROM 为 16K 100 = LDROM 为 1K, APROM 为 15K 101 = LDROM 为 2K, APROM 为 14K 110 = LDROM 为 3K, APROM 为 13K 111 = LDROM 为 4K, APROM 为 12K

CONFIG3-系统配置寄存器 3

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CONFIG3	--	--	--	--	LIRC	LVR[2:0]		
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

位号	符号	描述
3	LIRC	低速时钟在睡眠时设置: 0: 可关闭 1: 常开
2:0	LVR[2:0]	低压复位点 000: 1.6V (默认) 100: 1.8V 101: 2.0V 110: 2.5V 111: 3.5V

CONFIG4-系统配置寄存器 4

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CONFIG4	WDTEN[3:0]				--	NF_Pow_Sel[1:0]		NF_cycle
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

位号	符号	描述
7:4	WDTEN[3:0]	看门狗定时器 WDT 使能 0000 = 看门狗定时器关闭, WDT 可用作普通定时器用 1010 = 使能 WDT 超时复位功能, 且 WDT 在空闲或掉电模式下停止运行 0101 = 使能 WDT 超时复位功能, 且 WDT 在空闲或掉电模式下保持运行
2:1	NF_Pow_Sel [1:0]	NoiseFilter 噪声滤波对电源选择 00 = disable 01 = 对 VDD 进行滤波 10 = 对 Core 电源进行滤波 11 = 对 Core 电源和 VDD 同时进行滤波
0	NF_cycle	NoiseFilter 噪声滤波周期选择 0: 滤波周期 4ms 1: 滤波周期 2ms

CONFIG5-系统配置寄存器 5

Register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CONFIG5	-	XT32MODE	OSC XT32	P20CFG	OSCPRE[1:0]		HIRCFS[1:0]	
R/W	--	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	--	1	0	0	0	0	1	1

位号	符号	描述
6	XT32MODE	XT32 模式 0: XT32 弱模式 1: XT32 强模式
5	OSC_XT32	XT32 振荡器管脚使能 0: P20/P30 用于 GPIO 或其他 1: P20/P30 用于 XT32 模式 (使用外部时钟通过 P30 驱动时, 请设置 OSC_XT32 为 0, 并设置 P30 为输入)
4	P20CFG	P20 引脚外部功能配置 0: P20 用于 GPIO 或其他 1: P20 用于 VREF 外部输出
3:2	OSCPRE[1:0]	主时钟分频 00: Fosc/1 01: Fosc/2 10: Fosc/4 11: Fosc/8
1:0	HIRCFS[1:0]	Hirc 分频 00: Hirc/6 01: Hirc/3 10: Hirc/2 11: Hirc/1

表 26-1 P20 引脚功能配置表

P20 引脚功能	OSC_XT32 (CONFIG5.5)	P20CFG (CONFIG5.4)	RSTP20 (CONFIG0.2)	描述
P20_GPIO	0	0	0	P20 设置为普通 GPIO 引脚
P20_RST	0	0	1	P20 设置为外部 RST 引脚
P20_ADC_VREF	0	1	x	P20 设置为 VREF 基准电压输出引脚
P20_OSC_XT32	1	x	x	P20、P30 作为外部低速晶振引脚

27 电气参数

27.1 绝对最大额定值

符号	参数	最小值	最大值	单位
T_{ST}	存储温度	-55	150	°C
T_A	工作温度	-40	105	°C
VDD-VSS	电源电压	-0.3	5.8	V
V_{IN}	输入电压	VSS-0.3	VDD+0.3	V
I_{DD}	VDD 最大输入电流	-	120	mA
I_{SS}	VSS 最大输出电流	-	120	mA
I_{IO}	单个 IO 最大灌电流	-	82	mA
	单个 IO 最大拉电流	-	71	mA

27.2 直流电气特性

VDD-VSS=2.1~5.5V, $T_A=25^{\circ}\text{C}$

符号	参数	测试条件	最小值	典型值	最大值	单位
VDD	工作电压	$F_{SYS}=48\text{MHz}$, $F_{CPU}=F_{SYS}/2$ $F_{SYS}=8\text{MHz}\sim 24\text{MHz}$, $F_{CPU}=F_{SYS}$	2.1	5	5.5	V
I_{DD}	正常模式	VDD=5V, $F_{SYS}=48\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}/2$	-	4.7	-	mA
		VDD=3V, $F_{SYS}=48\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}/2$	-	4.4	-	mA
		VDD=5V, $F_{SYS}=24\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	4.5	-	mA
		VDD=3V, $F_{SYS}=24\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	4.3	-	mA
		VDD=5V, $F_{SYS}=16\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	3.6	-	mA
		VDD=3V, $F_{SYS}=16\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	3.4	-	mA
		VDD=5V, $F_{SYS}=8\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	2.7	-	mA
		VDD=3V, $F_{SYS}=8\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	2.6	-	mA
	IDLE 模式	VDD=5V, $F_{SYS}=48\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}/2$	-	2.9	-	mA
		VDD=3V, $F_{SYS}=48\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}/2$	-	2.9	-	mA
		VDD=5V, $F_{SYS}=24\text{MHz}$, 所有外设关闭	-	2.8	-	mA

		$F_{CPU}=F_{SYS}$				
		VDD=3V, $F_{SYS}=24\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	2.8	-	mA
		VDD=5V, $F_{SYS}=16\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	2.4	-	mA
		VDD=3V, $F_{SYS}=16\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	2.4	-	mA
		VDD=5V, $F_{SYS}=8\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	2.0	-	mA
		VDD=3V, $F_{SYS}=8\text{MHz}$, 所有外设关闭 $F_{CPU}=F_{SYS}$	-	2.0	-	mA
I _{SLEEP1}	休眠电流	所有外设关闭, WDT 定时器使能, VDD=3V	-	5.8	-	uA
I _{SLEEP2}	休眠电流	所有外设关闭, WKT 定时器使能, VDD=3V	-	5.8	-	uA
I _{SLEEP3}	休眠电流	所有外设关闭, VDD=5V	-	4.3	-	uA
		所有外设关闭, VDD=3V		3.3		uA
I _{LI}	输入漏电	-	-	-	0.1	uA
V _{IL}	输入低电平	-	VSS	-	0.3VDD	V
V _{IH}	输入高电平	-	0.7VDD	-	VDD	V
V _{OL}	输出低电压	VDD=5V, I _{OL1} =12mA	-	0.3	0.5	V
		VDD=5V, I _{OL2} =7mA	-	0.3	0.5	V
		VDD=3V, I _{OL1} =9mA	-	0.3	0.5	V
		VDD=3V, I _{OL2} =5mA	-	0.3	0.5	V
V _{OH}	输出高电压	VDD=5V, I _{OH1} =40mA	3.0	3.4	-	V
		VDD=5V, I _{OH1} =20mA	3.0	4.3	-	
		VDD=5V, I _{OH2} =20mA	3.0	3.6	-	
		VDD=3V, I _{OH1} =15mA	1.9	2.2	-	V
		VDD=3V, I _{OH1} =20mA	1.9	1.7	-	V
		VDD=3V, I _{OH2} =8mA	1.9	2.2	-	V
R _{PH}	上拉电阻	VDD=5V	-	38	-	KΩ
		VDD=3V		60		KΩ
	强上拉电阻	VDD=5V		10		KΩ
		VDD=3V	-	17	-	KΩ
R _{PL}	下拉电阻	VDD=5V	-	31	-	KΩ
		VDD=3V		60		KΩ

注: VDD=5V, 温度为 27℃, I_{OL1} 为强模式, I_{OL2} 为普通模式, I_{OH1} 为强模式, I_{OH2} 为普通模式

27.3 交流电气特性

27.3.1 上电与掉电操作

$T_A=25^{\circ}\text{C}$ ，不包括 32.768K 晶振起振时间

符号	参数	测试条件	最小值	典型值	最大值	单位
T_{RESET}	时钟结构	VDD=5V	-	8	-	ms
T_{VDDR}	VDD rise rate	VDD=5V	2	-	∞	us/V
T_{VDDF}	VDD fall rate	VDD=5V	2	-	∞	us/V

27.3.2 外部振荡器

符号	参数	测试条件	最小值	典型值	最大值	单位
V_{LSE}	工作电压	$F_{\text{SYS}}=16\text{KHz}$	2.1	-	5.5	V

27.3.3 内部振荡器

VDD=2.1V-5.5V

符号	参数	测试条件	频率误差	最小值	典型值	最大值	单位
F_{HSI}	内部高速 48MHz	$T_A=0^{\circ}\text{C}$ 至 80°C	$\pm 1\%$	-	48	-	MHz
		$T_A=-40$ 至 105°C	$\pm 2\%$	-	48	-	MHz
F_{LSI}	内部低速 16KHz	$T_A=25^{\circ}\text{C}$	$\pm 10\%$	-	16	-	KHz
		$T_A=-40$ 至 105°C	$\pm 20\%$	-	16	-	KHz

27.4 GPIO 电气特性

PxSR.n 位数 值	符号	参数	条件	普通模式	强驱动模式	单位
0 (普通斜率)	F _{OUT}	最大输出频率	V _{DD} =5.0V, C _L =30pF	37.69	60.32	MHz
			V _{DD} =3.3V, C _L =30pF	26.65	43.75	
			V _{DD} =2.4V, C _L =30pF	18.59	29.69	
	T _R	输出由 0 至 1 上升时间	V _{DD} =5.0V, C _L =30pF	7.11	4.55	ns
			V _{DD} =3.3V, C _L =30pF	10.04	6.92	
			V _{DD} =2.4V, C _L =30pF	14.49	9.17	
	T _F	输出由 1 至 0 下降时间	V _{DD} =5.0V, C _L =30pF	6.04	3.71	ns
			V _{DD} =3.3V, C _L =30pF	8.57	4.8	
			V _{DD} =2.4V, C _L =30pF	12.46	6.07	
1 (高速斜率)	F _{OUT}	最大输出频率	V _{DD} =5.0V, C _L =30pF	52.34	73.36	MHz
			V _{DD} =3.3V, C _L =30pF	41.13	56.05	
			V _{DD} =2.4V, C _L =30pF	29.64	43.78	
	T _R	输出由 0 至 1 上升时间	V _{DD} =5.0V, C _L =30pF	5	2.83	ns
			V _{DD} =3.3V, C _L =30pF	6.88	3.84	
			V _{DD} =2.4V, C _L =30pF	9.7	5.36	
	T _F	输出由 1 至 0 下降时间	V _{DD} =5.0V, C _L =30pF	4.55	3.93	ns
			V _{DD} =3.3V, C _L =30pF	5.6	4.91	
			V _{DD} =2.4V, C _L =30pF	7.12	5.98	

27.5 低压复位电气参数

符号	参数	最小值	典型值	最大值	单位
V _{LVR0}	低压侦阈值 1.9V	1.75	1.9	2.05	V
V _{LVR1}	低压侦阈值 2.2V	2.05	2.2	2.35	V
V _{LVR2}	低压侦阈值 2.6V	2.45	2.6	2.75	V
V _{LVR3}	低压侦阈值 3.7V	3.50	3.7	3.90	V

27.6 BOD 电气参数

符号	参数	最小值	典型值	最大值	单位
V_{LVD1}	低压侦阈值 1.95V	1.85	1.95	2.05	V
V_{LVD2}	低压侦阈值 2.05V	1.95	2.05	2.15	V
V_{LVD3}	低压侦阈值 2.16V	2.05	2.15	2.25	V
V_{LVD4}	低压侦阈值 2.29V	2.20	2.30	2.40	V
V_{LVD5}	低压侦阈值 2.45V	2.35	2.45	2.55	V
V_{LVD6}	低压侦阈值 2.59V	2.50	2.60	2.70	V
V_{LVD7}	低压侦阈值 2.78V	2.70	2.80	2.90	V
V_{LVD8}	低压侦阈值 2.99V	2.90	3.00	3.10	V
V_{LVD9}	低压侦阈值 3.25V	3.15	3.25	3.35	V
V_{LVD10}	低压侦阈值 3.54V	3.45	3.55	3.65	V
V_{LVD11}	低压侦阈值 3.89V	3.80	3.90	4.00	V
V_{LVD12}	低压侦阈值 4.33V	4.20	4.30	4.40	V
V_{HY}	迟滞窗口	30	100	300	mV

27.7 比较器电气参数

符号	参数	最小值	典型值	最大值	单位
A_{vol}	开环增益	-	70	-	dB
V_{HY}	迟滞窗口 1	-	0	-	mV
V_{HY}	迟滞窗口 2	-	20	-	mV
V_{HY}	迟滞窗口 3	-	30	-	mV
V_{HY}	迟滞窗口 4	-	60	-	mV

27.8 ADC 电压特性

 $T_A=25^{\circ}\text{C}$

符号	参数	最小值	典型值	最大值	单位
V_{AVDD}	ADC 工作电压	2.1	-	5.5	V
V_{REF1}	参考电压 1	-	V_{AVDD}	-	V
V_{REF2}	参考电压 2	1.998	2	2.002	V
V_{REF3}	参考电压 3	2.997	3	3.003	V
V_{REF4}	参考电压 4	3.996	4	4.004	V
V_{ADI}	输入分压	0	-	V_{REF}	V
N_R	分辨率	12			Bit
DNL	微分非线性误差 ($V_{REF}=V_{AVDD}=5V, T_{ADCK}=0.5\mu s$)	± 2			LSB
INL	积分非线性误差 ($V_{REF}=V_{AVDD}=5V, T_{ADCK}=0.5\mu s$)	± 4			LSB
T_{ADCK}	ADC 时钟周期	0.08	-	32	μs
T_{ADC}	ADC 转换时间	14.5	18.5	22.5	T_{ADCK}
F_S	采样率 ($V_{REF}=V_{AVDD}=5V$)	-	100	740	Ksps

27.9 BANDGAP 电气特性

符号	参数	测试条件	最小值	典型值	最大值	单位
V_{BG}	内部基准 1.2V	$V_{DD}=2.1-5.5V, T_A=-40^{\circ}\text{C}$ 至 105°C	1.2	1.24	1.28	V

27.10 FLASH 电气特性

符号	参数	测试条件	最小值	典型值	最大值	单位
V_F	FLASH 工作电压	-	2.1	-	5.5	V
T_F	FLASH 工作温度	-	-40	27	105	$^{\circ}\text{C}$
V_{RF}	FLASH 读电压	16M/1T 或者 32M/2T	2.5		5.5	V
		8M/1T 或者 16M/2T	1.8		2.5	V
V_{WF}	FLASH 写电压		2.4		5.5	V
$N_{ENDURANCE}$	擦写次数	程序 FLASH	10,000	-	-	Cycle
T_{RET}	数据保存时间	105°C	10	-	-	year
T_{ERASE}	扇区擦除时间	-	100	-	200	ms
T_{PROG}	编程时间	-	60	-	100	μs
I_{DD1}	读取电流	-	-	-	3	mA
I_{DD2}	编程电流	-	-	-	8	mA
I_{DD3}	擦除电流	-	-	-	9	mA

27.11 OPA 电气特性

$T_A=25^{\circ}\text{C}$, $V_{\text{SENSE}}=V_{\text{IN}+}-V_{\text{IN}-}$, $V_{\text{DD}}=5\text{V}$, $V_{\text{IN}+}=1\text{V}$, 除非另有说明。

符号	参数	条件	最小值	典型值	最大值	单位
VDD	电源电压	-	2.5	-	5.5	V
I _Q	静态电流	$V_{\text{SENSE}}=0\text{mV}$	-	0.2	0.25	mA
I _{SD}	关断电流	-	-	5	-	nA
T _A	工作温度	-	-40	25	105	°C
输入特性						
V _{OS}	输入失调电压	未调零	-	±2	-	mV
		调零后	-	±0.2	±0.5	
V _{CM}	共模输入电压范围	-40°C~105°C	0	-	VDD	V
I _B	输入偏置电流	$V_{\text{SENSE}}=0\text{mV}$	-	10	-	pA
I _{OS}	输入失调电流	$V_{\text{SENSE}}=0\text{mV}$	-	10	-	pA
输出特性						
V _{OH}	最大输出电压	-40°C~105°C	-	-	VDD-0.1	V
V _{OL}	最小输出电压	-40°C~105°C	0.1	-	-	V
频率特性						
A _{OL}	开环增益	-	-	85	-	dB
GB	增益带宽	Buffer	-	1	-	MHz
PSRR	电源抑制比	VDD=2.5~5.5V, $V_{\text{IN}+}=1\text{V}$, $V_{\text{SENSE}}=0\text{mV}$	-	81	-	dB
瞬态特性						
SR	摆率	$R_{\text{LOAD}}=2\text{K}$, $C_{\text{LOAD}}=100\text{pF}$	-	±2	-	V/μs
T _{STB}	稳定时间	-	-	-	3	μs
SNR	信噪比	VDD=5V 输入 0-2V 正弦波		81.2		dB
THD	总谐波失真	VDD=5V 输入 0-2V 正弦波		0.008		%

27.12 EFT 电气特性

符号	参数	测试条件	最大值	单位	等级
V_{EFTB}	Fast transient voltage burst limits to be applied through 0.1uF(capacitance) on VDD and VSS pins to induce a functional disturbance	$T_A = +25^{\circ}\text{C}$ HSI=8MHz, conforms to IEC 61000-4-4	4000	V	4B

27.13 ESD 电气特性

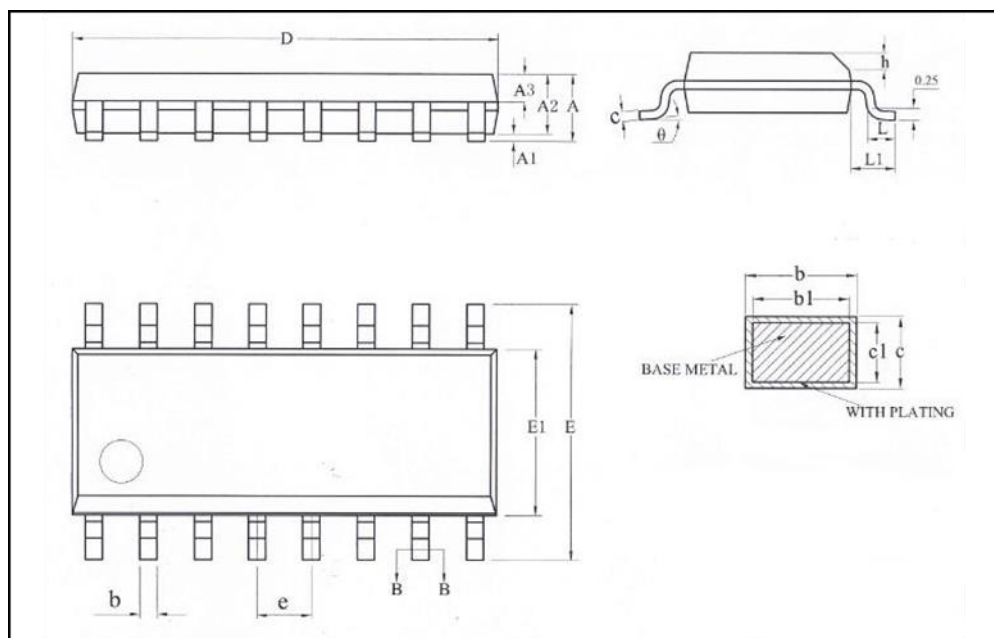
符号	参数	测试条件	最大值	单位	等级
V_{ESD}	静电放电 (人体放电模式 HBM)	$T_A = +25^{\circ}\text{C}$ JEDEC EIA/JESD22A-114	4000	V	3B
	静电放电 (机器放电模式 MM)	$T_A = +25^{\circ}\text{C}$ JEDEC EIA/JESD22A-115	400	V	C

27.14 Latch-UP 电气特性

符号	参数	测试条件	测试类型	最大值	单位
LU	Static latch up class	JEDEC STANDARD NO.78D NOVEMBER 2011	Class I ($T_A = +25^{\circ}\text{C}$)	± 200	mA

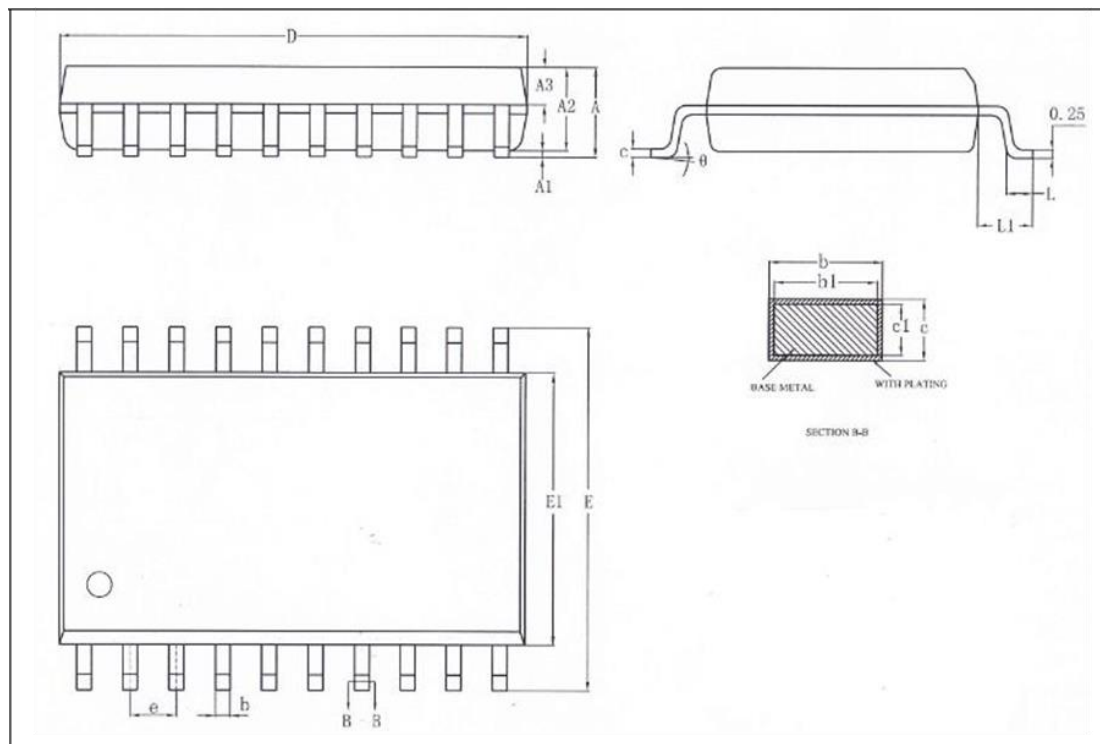
28 封装信息

28.1 SOP16



Symbol	Millimeter		
	Min	Nom	Max
A	-	-	1.75
A1	0.10	-	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	-	0.47
b1	0.38	0.41	0.44
c	0.20	-	0.24
c1	0.19	0.20	0.21
D	9.80	9.90	10.00
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e		1.27BSC	
h	0.25	-	0.50
L	0.50	-	0.80
L1		1.05	
θ	0	-	8°

28.2 SOP20



Symbol	Millimeter		
	Min	Nom	Max
A	-	-	2.65
A1	0.10	-	0.30
A2	2.25	2.30	2.35
A3	0.97	1.02	1.07
b	0.35	-	0.43
b1	0.34	0.37	0.40
c	0.25	-	0.29
c1	0.24	0.25	0.26
D	12.70	12.80	12.90
E	10.10	10.30	10.50
E1	7.40	7.50	7.60
e		1.27BSC	
L	0.70	-	1.00
L1		1.40	
θ	0	-	8°

29 版本说明

版本号	时间	修改内容
V1.0	2024.03.01	初始版本
V1.1	2024.03.10	修改部分文字描述
V1.2	2024.03.27	增加对文档的适用范围和权利义务的描述
V1.3	2024.03.28	修改电气参数
V1.4	2024.05.15	修改 EB0D 在中断位的描述
V1.5	2024.07.09	增加 CKCON 寄存器描述及其用法 修正定时器 2 捕获功能的部分描述