



RJM8L151 系列

用户手册

文档版本 V2.04

武汉瑞纳捷半导体有限公司

Wuhan Runjet Semiconductor Co.,Ltd

武汉瑞纳捷半导体有限公司

重要声明

本文档必须以用户光盘的形式或者加密 E-MAIL 的形式发布给某个个人或组织，并且以其官方 E-MAIL 地址作为水印标记。禁止其他的发布形式。

本文档没有任何形式的担保、立场表达或其他暗示，若有任何因本文档或其中提及的产品所有资讯所引起的直接或间接损失，瑞纳捷公司及所属员工恕不为其担保任何责任。武汉瑞纳捷半导体有限公司保留不发出通知就改变其产品或产品说明书的权利，保留不更新本文档以反映这些改变的权利。

目录

1. 系统概述.....	10
1.1. 功能框图.....	10
1.2. 电源框图.....	10
2. 处理器.....	11
2.1. 概述.....	11
2.2. 主要特性.....	11
2.3. 内部数据存储器.....	11
2.3.1. 介绍.....	11
2.3.2. 特殊功能寄存器.....	12
3. FLASH 程序存储器和数据.....	13
3.1. 概述.....	13
3.2. 术语.....	13
3.3. FLASH 主要特性.....	13
3.4. 存储器组织结构.....	13
3.4.1. 数据区 (XDATA)	14
3.4.2. 主程序区 (CODE)	14
3.5. 存储器保护.....	14
3.6. FLASH 寄存器.....	15
3.6.1. 控制寄存器 FLASH_CTRL (偏移: 21H).....	15
3.6.2. DMA 源地址高 8 位 FLASH_DMA_CTRL0 (偏移: 29H).....	16
3.6.3. DMA 源地址低 8 位 FLASH_DMA_CTRL1 (偏移: 2AH).....	16
3.6.4. DMA 目的地址高 8 位 FLASH_DMA_CTRL2 (偏移: 2BH).....	16
3.6.5. DMA 目的地址低 8 位 FLASH_DMA_CTRL3 (偏移: 2CH).....	16
3.6.6. DMA 传输长度 FLASH_DMA_CTRL4 (偏移: 2EH).....	16
3.6.7. SCREMAP 控制寄存器 SCRREMAP (偏移: 12H)	16
3.7. FLASH 操作说明.....	17
3.7.1. 擦除操作.....	17
3.7.2. 编程操作.....	17
3.8. MEMORY 保护配置.....	17
3.8.1. 保护操作.....	17
4. 电源检测控制.....	18
4.1. 电源.....	18
4.2. 上电复位 (POR) /掉电复位(PDR).....	18
4.3. 欠压复位 (BOR)	18
4.4. 可编程电压检测 (PVD)	18
4.5. 电压检测控制寄存器.....	18
4.5.1. 电压检测及复位使能寄存器 SCRSTEN (偏移: 16H).....	19
4.5.2. 电压检测档位选择寄存器 LVD_CFG (偏移: F6H).....	19
4.6. 电压检测操作说明.....	20
4.6.1. PVD 初始化流程.....	20
4.6.2. BOR 初始化流程.....	20
4.6.3. PDR 初始化流程.....	20
5. 低功耗模式.....	21
5.1. 概述.....	21
5.2. DEEPSLEEP.....	21
5.2.1. DeepSleep 模式唤醒源.....	21

5.2.2. 进入 DeepSleep 模式.....	22
5.2.3. DeepSleep 模式唤醒.....	22
5.2.4. DeepSleep 模式寄存器.....	22
5.3. HALT.....	24
5.3.1. Halt 模式唤醒源.....	24
5.3.2. 进入 Halt 模式.....	24
5.3.3. Halt 模式唤醒.....	24
5.3.4. Halt 模式寄存器.....	24
5.4. ACTIVE_HALT.....	25
5.4.1. Active_Halt 模式唤醒源.....	25
5.4.2. 进入 Active_Halt 模式唤醒源.....	25
5.4.3. Active_Halt 模式唤醒.....	26
5.5. WAIT.....	26
5.5.1. Wait 模式唤醒源.....	26
5.5.2. 进入 Wait 模式.....	26
5.5.3. Wait 模式唤醒.....	26
5.6. LOW_POWER_WAIT.....	26
5.6.1. Low_Power_Wait 模式唤醒源.....	26
5.6.2. 进入 Low_Power_Wait 模式.....	27
5.6.3. Low_Power_Wait 模式唤醒.....	27
5.7. LOW_POWER_RUN.....	27
5.7.1. 进入 Low_Power_Run 模式.....	27
6. 复位 (RST)	28
6.1. 复位电路.....	28
6.2. 内部复位源.....	28
6.2.1. 上/掉电复位 (POR/PDR)	28
6.2.2. 欠压复位 (BOR)	28
6.2.3. 看门狗复位.....	28
6.2.4. 模块软件复位.....	28
6.3. 复位寄存器.....	28
6.3.1. 软复位复位 CPU 和 7816 的 C2 脚 REST 请求寄存器 SCU(偏移: 11H).....	29
6.3.2. 安全模块复位使能寄存器 SCRSTEN(偏移: 16H).....	29
6.3.3. 系统软复位控制寄存器 SCSSFTRST(偏移: 17H).....	29
6.3.4. 复位使能寄存器 RSTCON0(偏移: 18H).....	30
7. 时钟控制.....	31
7.1. 主时钟源.....	31
7.1.1. HSE.....	31
7.1.2. HSI.....	32
7.1.3. LSE.....	32
7.1.4. LSI.....	32
7.2. 系统时钟源选择.....	32
7.2.1. 系统启动.....	32
7.2.2. 系统时钟切换.....	32
7.3. RTC 时钟选择.....	32
7.4. 外设时钟选择.....	32
7.5. 时钟控制寄存器.....	33
7.5.1. 系统时钟选择寄存器 SCSYS (偏移: 15H)	33
7.5.2. OSC 控制寄存器 LP_CTRL 1(偏移: 10H).....	33
7.5.3. 时钟控制寄存器 SCCM0 (偏移: 13H)	34
7.5.4. 时钟控制寄存器 SCCM1 (偏移: 14H)	35
7.5.5. 时钟控制寄存器 SCCM2 (偏移: 19H)	35
8. 通用输入输出 (GPIO) 及端口复用.....	37

8.1. 概述.....	37
8.2. GPIO 主要特性.....	37
8.3. I/O 的配置和使用.....	37
8.3.1. 普通输入/输出 I/O.....	37
8.3.2. 作为外部中断源.....	37
8.4. I/O 管脚复用寄存器.....	37
8.4.1. 管脚复用控制寄存器 IOCFG_CTRL0 (偏移: 00H)	38
8.4.2. 管脚复用控制寄存器 IOCFG_CTRL1 (偏移: 07H)	38
8.4.3. 管脚复用控制寄存器 IOCFG_CTRL2 (偏移: 08H)	39
8.4.4. 管脚复用控制寄存器 IOCFG_CTRL3 (偏移: 09H)	39
8.4.5. 管脚复用控制寄存器 IOCFG_CTRL4 (偏移: 0AH)	40
8.4.6. 管脚复用控制寄存器 IOCFG_CTRL5 (偏移: 0BH)	40
8.4.7. 管脚复用控制寄存器 IOCFG_CTRL6 (偏移: 0CH)	41
8.4.8. 管脚复用控制寄存器 IOCFG_CTRL7 (偏移: 0DH)	41
8.4.9. 管脚复用控制寄存器 IOCFG_CTRL8 (偏移: 0EH)	42
8.4.10. GPIO 功能复用表.....	42
8.5. GPIO 寄存器.....	44
8.5.1. GPIOx 寄存器 Px (偏移: 80H、90H、A0H、B0H、91H)	45
8.5.2. GPIOx 输出使能寄存器 Px_OEN (偏移: A4H、A5H、A6H、A7H、96H)	46
8.5.3. GPIOx 中断使能寄存器 Px_INT_EN (偏移: D0H、D5H、DAH、DFH、E4H).....	46
8.5.4. GPIOx 中断触发方式选择寄存器 Px_INT_SEL0, Px_INT_SEL1, Px_INT_SEL2, (偏移: 见寄存器列表).....	46
8.5.5. GPIOx 中断状态寄存器 Px_INT_REG(偏移: D4H、D9H、DEH、E3H、E8H).....	47
8.6. GPIO 操作说明.....	47
8.6.1. 输入输出 IO 配置.....	48
8.6.2. 中断触发模式配置.....	48
8.6.3. 清除中断标志.....	48
9. 中断服务程序单元 (ISR)	49
9.1. 概述.....	49
9.2. 中断服务单元特性.....	49
9.3. 中断源.....	49
9.4. 中断优先级.....	50
9.5. 中断服务程序单元寄存器.....	51
9.5.1. 中断使能寄存器 IEN0(偏移: A8H).....	51
9.5.2. 中断使能寄存器 IEN1(偏移: B8H).....	52
9.5.3. 中断使能寄存器 IEN2(偏移: 9AH).....	52
9.5.4. 中断使能寄存器 IEN3(偏移: C9H).....	53
9.5.5. 中断使能寄存器 IEN4(偏移: D1H).....	53
9.5.6. 中断优先级寄存器 IPO(偏移: A9H).....	54
9.5.7. 中断优先级寄存器 IP1(偏移: B9H).....	54
10. 模拟/数字转换器 (ADC)	55
10.1. 概述.....	55
10.2. ADC 主要特性.....	55
10.3. ADC 引脚描述.....	55
10.4. ADC 功能描述.....	55
10.4.1. ADC 开-关控制.....	55
10.4.2. 通道选择.....	56
10.4.3. 转换模式.....	56
10.4.4. ADC 校准及异常处理.....	56
10.5. ADC 寄存器.....	56
10.5.1. 配置寄存器 ADC_CSR(偏移: 03H).....	57
10.5.2. 配置寄存器 ADC_CR1(偏移: 04H).....	57

10.5.3. 配置寄存器 ADC_CR2(偏移: 05H).....	58
10.5.4. 配置寄存器 ADC_CR3(偏移: 06H).....	58
10.5.5. 数据高位寄存器 ADC_DRH(偏移: 07H).....	59
10.5.6. 数据低位寄存器 ADC_DRL(偏移: 08H).....	59
10.5.7. 状态寄存器 ADC_STS(偏移: 09H).....	59
10.5.8. 中断屏蔽寄存器 ADC_MSK(偏移: 0AH).....	60
10.6. ADC 操作说明.....	60
10.6.1. 单次模式.....	60
10.6.2. 连续模式.....	61
11. 比较器 (COMP)	62
11.1. 概述.....	62
11.2. COMP 主要特性.....	62
11.3. COMP 引脚描述.....	62
11.4. 比较器寄存器.....	62
11.4.1. 控制寄存器 COMP_CTRL0(偏移: 1EH).....	62
11.4.2. 控制寄存器 COMP_CTRL1(偏移: 1FH).....	63
11.4.3. 状态/控制寄存器 IOCFG_CTRL6(偏移: FCH).....	64
11.5. 比较器操作说明.....	64
11.5.1. 比较器 1 (COMP1)	64
11.5.2. 比较器 2 (COMP2)	64
12. 基本定时器 (TIMER0/1)	65
12.1. 概述.....	65
12.2. 基本定时器主要特性.....	65
12.3. 基本定时器工作模式.....	65
12.4. 基本定时器寄存器.....	66
12.4.1. 定时器控制寄存器 TCON(偏移: 88H).....	66
12.4.2. 定时器模式寄存器 TMOD(偏移: 89H).....	66
12.4.3. TIMER0 计数寄存器低字节 TLO(偏移: 8AH).....	67
12.4.4. TIMER1 计数寄存器低字节 TL1(偏移: 8BH).....	67
12.4.5. TIMER0 计数寄存器高字节 TH0(偏移: 8CH).....	67
12.4.6. TIMER1 计数寄存器高字节 TH1(偏移: 8DH).....	67
12.5. 基本定时器操作说明.....	67
13. 通用定时器 (TIMER2)	69
13.1. 概述.....	69
13.2. 基本定时器主要特性.....	69
13.3. 分频器.....	69
13.4. 定时器模式.....	69
13.5. 事件计数模式.....	69
13.6. 门控定时模式.....	70
13.7. 定时器重载功能.....	70
13.8. 定时器比较输出功能.....	70
13.8.1. 比较模式.....	70
13.8.2. PWM 模式.....	71
13.9. 定时器捕获功能.....	71
13.10. 定时器 2 中断.....	72
13.11. 定时器 2 寄存器.....	72
13.11.1. TIMER2 控制寄存器 T2CON(偏移: C8H).....	73
13.11.2. TIMER2 捕获/比较模式选择寄存器 CCEN(偏移: C1H).....	73
13.11.3. TIMER2 比较输出极性选择寄存器 CCL3(偏移: C6H).....	74
13.11.4. TIMER2 比较/捕获寄存器 CCL1(偏移: C2H).....	74
13.11.5. TIMER2 比较/捕获寄存器 CCH1(偏移: C3H).....	74

13.11.6. TIMER2 比较/捕获寄存器 CCL2(偏移: C4H).....	74
13.11.7. TIMER2 比较/捕获寄存器 CCH2(偏移: C5H).....	74
13.11.8. TIMER2 比较/捕获寄存器 CRCL(偏移: CAH).....	74
13.11.9. TIMER2 比较/捕获寄存器 CRCH(偏移: CBH).....	75
13.11.10. TIMER2 计数寄存器低字节 TL2(偏移: CCH).....	75
13.11.11. TIMER2 计数寄存器高字节 TH2(偏移: 0DH).....	75
13.11.12. TIMER2 中断标志状态寄存器 IRCON(偏移: C0H).....	75
13.12. 定时器 2 操作说明.....	75
13.12.1. 定时.....	75
13.12.2. 比较.....	76
13.12.3. 捕获.....	77
14. 高级定时器 (TIMER3/TIMER4)	79
14.1. 概述.....	79
14.2. 高级定时器主要特性.....	79
14.3. 分频器.....	79
14.4. 定时器模式.....	79
14.5. 事件计数模式.....	80
14.6. 门控定时模式.....	80
14.7. 定时器重载功能.....	80
14.8. 定时器比较输出功能.....	80
14.8.1. 比较模式.....	80
14.8.2. PWM 模式 (互补及死区插入)	81
14.8.3. 刹车功能.....	83
14.9. 定时器捕获功能.....	83
14.10. 定时器 3/4 中断.....	85
14.11. 定时器 3/4 寄存器.....	85
14.11.1. TIMER3/4 中断标志及控制寄存器 TIMERi_IRCON(偏移: 00H, 此处 x 代表 3、4, 下同).....	86
14.11.2. TIMER3/4 中断请求控制寄存器 TIMERi_IEN1(偏移: 01H).....	87
14.11.3. TIMER3/4 控制寄存器 TIMERi_T2CON(偏移: 02H).....	87
14.11.4. TIMER3/4 模式使能寄存器 TIMERi_CCEN(偏移: 03H).....	88
14.11.5. TIMER3/4 比较/捕获寄存器 TIMERi_CCL1(偏移: 04H).....	88
14.11.6. TIMER3/4 比较/捕获寄存器 TIMERi_CCH1(偏移: 05H).....	88
14.11.7. TIMER3/4 比较/捕获寄存器 TIMERi_CCL2(偏移: 06H).....	88
14.11.8. TIMER3/4 比较/捕获寄存器 TIMERi_CCH2(偏移: 07H).....	88
14.11.9. TIMER3/4 死区时间寄存器低字节 TIMERi_CCL3(偏移: 08H).....	89
14.11.10. TIMER3/4 死区时间寄存器高字节 TIMERi_CCH3(偏移: 09H).....	89
14.11.11. TIMER3/4 比较/捕获寄存器 TIMERi_CRCL(偏移: 0AH).....	89
14.11.12. TIMER3/4 比较/捕获寄存器 TIMERi_CRCH(偏移: 0BH).....	89
14.11.13. TIMER3/4 计数寄存器 TIMERi_TL2(偏移: 0CH).....	89
14.11.14. TIMER3/4 计数寄存器 TIMERi_TH2(偏移: 0DH).....	89
14.11.15. TIMER3/4 刹车控制寄存器 TIMERi_BOEN(偏移: 0EH).....	89
14.11.16. TIMER3/4 比较模式 1 数据寄存器 TIMERi_ID(偏移: 0FH).....	90
14.11.17. TIMER3/4 刹车输出状态控制寄存器 TIMERi_BREAK_OIS(偏移: 10H).....	90
14.11.18. TIMER3/4 比较输出极性选择寄存器 TIMERi_OUT_P(偏移: 11H).....	91
14.11.19. TIMER3/4 通道使能寄存器 TIMERi_OUT_EN(偏移: 12H).....	91
14.12. 定时器 3/4 操作说明.....	92
14.12.1. 定时.....	92
14.12.2. 比较.....	93
14.12.3. 捕获.....	94
15. 实时时钟 (RTC)	96
15.1. 概述.....	96

15.2. RTC 主要特性.....	96
15.3. RTC 功能描述.....	96
15.4. RTC 寄存器.....	96
15.4.1. 小时配置寄存器(偏移: 01H).....	97
15.4.2. 分钟配置寄存器(偏移: 02H).....	97
15.4.3. 秒配置寄存器(偏移: 03H).....	97
15.4.4. 年配置寄存器(偏移: 05H).....	97
15.4.5. 月配置寄存器(偏移: 06H).....	97
15.4.6. 日期配置寄存器(偏移: 07H).....	97
15.4.7. 世纪配置寄存器(偏移: 08H).....	98
15.4.8. 闹铃小时配置寄存器(偏移: 09h).....	98
15.4.9. 闹铃分钟配置寄存器(偏移: 0AH).....	98
15.4.10. 闹铃秒配置寄存器(偏移: 0BH).....	98
15.4.11. 闹铃年配置寄存器(偏移: 0DH).....	98
15.4.12. 闹铃月份配置寄存器(偏移: 0EH).....	98
15.4.13. 闹铃日期状态寄存器(偏移: 0FH).....	98
15.4.14. 基本计数器 SEC_CNT_3~ SEC_CNT_0 秒计数校准寄存器(偏移: 10H~13H).....	99
15.4.15. TICK 中断配置寄存器(偏移: 14H).....	99
15.4.16. ALARM 中断配置寄存器(偏移: 15H).....	99
15.4.17. 重新配置 RTC 寄存器使能寄存器(偏移: 16H).....	100
15.4.18. RTC 开始工作控制寄存器(偏移: 17H).....	100
15.4.19. RTC 中断标志清除寄存器(偏移: 18H).....	100
15.5. RTC 操作说明.....	100
16. 看门狗(WDT)	102
16.1. 概述.....	102
16.2. WDT 主要特性.....	102
16.3. 看门狗复位.....	102
16.4. 看门狗中断.....	102
16.5. 看门狗开始工作.....	102
16.6. 看门狗计数器.....	102
16.7. 看门狗寄存器.....	102
16.7.1. WDTREL 加载寄存器(偏移: 86H).....	103
16.8. 看门狗操作说明.....	103
16.8.1. 看门狗定时器溢出产生中断.....	103
16.8.2. 看门狗定时器溢出产生系统复位.....	103
17. AES/SM4/DES 加解密.....	104
17.1. 概述.....	104
17.2. 主要特性.....	104
17.3. 加解密算法寄存器.....	104
17.3.1. SM4_DES_CON 控制寄存器(偏移: 00H).....	105
17.3.2. SM4_DES_STS 状态寄存器(偏移: 01H).....	106
17.3.3. SM4_DES_MSK 中断控制寄存器(偏移: 02H).....	106
17.3.4. SM4_DES_DAT0_X 数据寄存器 32 位(偏移: 03H).....	106
17.3.5. SM4_DES_DAT1_X 数据寄存器 32 位(偏移: 07H).....	106
17.3.6. SM4_DES_DAT2_X 数据寄存器 32 位(偏移: 0BH).....	106
17.3.7. SM4_DES_DAT3_X 数据寄存器 32 位(偏移: 0FH).....	107
17.3.8. SM4_DES_KEY0_X 数据寄存器 32 位(偏移: 13H).....	107
17.3.9. SM4_DES_KEY1_X 数据寄存器 32 位(偏移: 17H).....	107
17.3.10. SM4_DES_KEY2_X 数据寄存器 32 位(偏移: 1BH).....	107
17.3.11. SM4_DES_KEY3_X 数据寄存器 32 位(偏移: 1FH).....	107
17.3.12. SM4_DES_KEY4_X 数据寄存器 32 位(偏移: 23H).....	108

17.3.13. SM4_DES_KEY5_X 数据寄存器 32 位(偏移: 27H).....	108
17.4. 加解密算法操作说明.....	108
18. 乘除法器 (MDU)	110
18.1. 概述.....	110
18.2. 主要特性.....	110
18.3. MDU 计算操作描述.....	110
18.3.1. 16bit*16bit 乘法运算.....	110
18.3.2. 16bit/16bit 除法运算.....	110
18.3.3. 32bit/16bit 除法运算.....	110
18.3.4. 移位操作运算.....	110
18.4. MDU 寄存器.....	110
18.4.1. 运算控制寄存器(ARCON).....	111
18.5. MDU 计算操作说明.....	111
18.5.1. 16bit*16bit 乘法运算.....	111
18.5.2. 16bit/16bit 除法运算.....	111
18.5.3. 32bit/16bit 除法运算.....	112
18.5.4. 32bit/16bit 除法运算.....	112
19. 随机数 (RNG)	113
19.1. 概述.....	113
19.2. 随机数寄存器.....	113
19.2.1. RNG 控制寄存器 (DTRNGCON)	113
19.2.2. RNG 数据寄存器 (DTRNGDAT)	113
19.2.3. RNG 状态寄存器 (DTRNGSTS)	113
19.3. 随机数操作流程.....	113
20. CRC8 计算单元.....	114
20.1. 概述.....	114
20.2. 主要特性.....	114
20.3. CRC8 功能描述.....	114
20.4. CRC8 计算单元寄存器.....	114
20.4.1. 数据寄存器 CRC8_DATA (偏移: 00H)	115
20.4.2. 控制寄存器 CRC8_CTRL (偏移: 01H)	115
20.4.3. 初始值寄存器 CRC8_INIT (偏移: 02H)	115
20.5. CRC8 计算单元操作说明.....	115
21. I2C 接口.....	117
21.1. 概述.....	117
21.2. 主要特性.....	117
21.3. I2C 功能描述.....	117
21.3.1. I2C 通信过程.....	117
21.4. I2C 寄存器.....	118
21.4.1. I2Cx 主机波特率寄存器低位 I2Cx_M_PERER1 (偏移: C0H、50H) (此处x 代表0,1, 下同)	119
21.4.2. I2Cx 主机波特率寄存器高位 I2Cx_M_PERER2 (偏移: C1H、51H).....	119
21.4.3. I2Cx 主机控制寄存器 I2Cx_M_CTRL (偏移: C2H、52H).....	119
21.4.4. I2Cx 主机发送数据寄存器 I2Cx_M_TXR (偏移: C3H、53H).....	119
21.4.5. I2Cx 主机发送控制寄存器 I2Cx_M_CR (偏移: C4H、54H)	119
21.4.6. I2Cx 主机接收数据寄存器 I2Cx_M_RXR (偏移: C5H、55H)	120
21.4.7. I2Cx 主机状态寄存器 I2Cx_M_STS (偏移: C6H、56H)	120
21.4.8. I2Cx 主机软复位寄存器 I2Cx_M_RST (偏移: C7H、57H)	120
21.4.9. I2Cx 从机数据寄存器 I2Cx_S_DATA (偏移: C8H、58H)	120
21.4.10. I2Cx 从机状态寄存器 I2Cx_S_STS (偏移: C9H、59H)	121
21.4.11. I2Cx 从机设备地址寄存器 I2Cx_S_ADDR (偏移: CBH、5BH)	121

21.4.12. I2Cx 从机中断屏蔽寄存器 I2Cx_S_CON (偏移: CCH、5CH)	121
21.5. I2C 操作说明	122
21.5.1. 主机初始化流程	122
21.5.2. 主机发送流程	122
21.5.3. 主机接收流程	122
21.5.4. 从机初始化流程	123
21.5.5. 从机发送流程	123
21.5.6. 从机接收流程	124
22. UART	125
22.1. 概述	125
22.2. 主要特性	125
22.3. UART 功能描述	125
22.4. UART 寄存器	126
22.4.1. 控制状态寄存器 SxCON (此处 x 代表 0,1,2,3, 下同)	127
22.4.2. 数据缓冲寄存器 SxBUF	127
22.4.3. 重装寄存器低字节 SxRELL	128
22.4.4. 重装寄存器高字节 SxRELH	128
22.4.5. 特殊功能寄存器 PxCON	128
22.5. UART 操作说明	128
22.5.1. 串口的发送	128
22.5.2. 串口的接收	128
22.5.3. 多机通信	128
23. 串行外设接口 (SPI)	130
23.1. 概述	130
23.2. 主要特性	130
23.3. SPI 功能描述	130
23.3.1. SPI 的 4 个管脚描述	130
23.3.2. SPI 的时钟信号的相位和极性	130
23.4. SPI 寄存器	131
23.4.1. SPIx 控制寄存器 SPIxCON(偏移: 00H、08H、10H、18H) (此处 x 代表 0,1,2,3, 下同)	132
23.4.2. SPIx 数据寄存器 SPIxDAT(偏移: 01H、09H、11H、19H)	133
23.4.3. SPIx 状态寄存器 SPIxSTS(偏移: 02H、0AH、12H、1AH)	133
23.4.4. SPIx 中断屏蔽寄存器 SPIxMSK(偏移: 03H、0BH、13H、1BH)	133
23.5. SPI 操作说明	133
23.5.1. SPI 主模式初始化流程	133
23.5.2. SPI 主模式发送	133
23.5.3. SPI 主模式接收	134
23.5.4. SPI 从模式初始化流程	134
23.5.5. SPI 从模式发送	134
23.5.6. SPI 从模式接收	134
24. 接触式 IC 卡接口 (ISO7816MS)	135
24.1. 概述	135
24.2. 主要特性	135
24.3. ISO7816MS 寄存器	135
24.3.1. 控制寄存器 ISOCON0 (偏移: A0H)	136
24.3.2. 控制寄存器 ISOCON1 (偏移: A1H)	136
24.3.3. 控制寄存器 ISOCON2 (偏移: A2H)	137
24.3.4. 状态寄存器 ISOSTS0 (偏移: A3H)	137
24.3.5. 状态寄存器 ISOSTS1(偏移: A4H)	138
24.3.6. FIFO 复位寄存器 ISOFIFOCL(偏移: A5H)	138
24.3.7. 字符等待时间寄存器 WTDATA(偏移: A6H、A7H)	139

24.3.8. 波特率控制寄存器 ISOBRC (偏移: A8H).....	139
24.3.9. 数据缓存寄存器 ISOBUF (偏移: A9H).....	139
24.3.10. 直接 IO 控制寄存器 ISODIO(偏移: AAH).....	140
24.3.11. 中断屏蔽寄存器 ISOMSK (偏移: ABH).....	140
24.3.12. ETU 计数器控制寄存器 ISOTCON(偏移: ACH).....	140
24.3.13. ETU 计数器初始值寄存器 ISOTDAT(偏移: ADH、AEH).....	141
24.3.14. ETU 计数器自动重载寄存器 ISOTRLD(偏移: AFH、B0H).....	141
24.3.15. ETU 计数器中断屏蔽寄存器 ISOTMSK (偏移: B1H).....	141
24.3.16. ETU 计数器过程字符设置寄存器 ISONULL(偏移: B2H).....	141
24.3.17. C3 计数/主从模式选择寄存器 ISOFUC (偏移: B3H).....	141
24.4. ISO7816MS 操作说明.....	142
24.4.1. 主机模式选择与使用.....	142
24.4.2. 查询方式发送.....	142
24.4.3. 中断方式发送.....	142
24.4.4. 中断方式接收.....	143
24.4.5. ETU timer 操作.....	143
24.4.6. 芯片休眠之后 7816 唤醒.....	144
25. 版本号.....	145
25.1. 版本号寄存器.....	145
25.1.1. 版本寄存器 VERSION0 (偏移: 1AH)	145
25.1.2. 版本寄存器 VERSION1 (偏移: 1BH)	145
25.1.3. 版本寄存器 VERSION2 (偏移: 1CH)	145
25.1.4. 版本寄存器 VERSION3 (偏移: 1DH)	145
26. 编程接口.....	146
附录 1: 寄存器比特位属性及说明.....	147
附录 2: 专业术语缩写及说明.....	148
附录 3: 版本修订.....	149

1. 系统概述

RJM8L151 系列单片机是一款资源丰富的 8 位高性能 8051 内核芯片。芯片内置 256 字节内部数据存储器，2~8K 字节 SRAM，16~64K 字节的可编程 FLASH，具备快速中断响应，高效低功耗设计，多种休眠模式。片内集成多种安全密码模块，包括 SM4 / DES / 3DES/AES 安全算法，支持真随机数发生器，集成 CRC8 校验模块。芯片提供了多种外围接口：SPI、UART、ISO7816、I2C、GPIO。

1.1. 功能框图

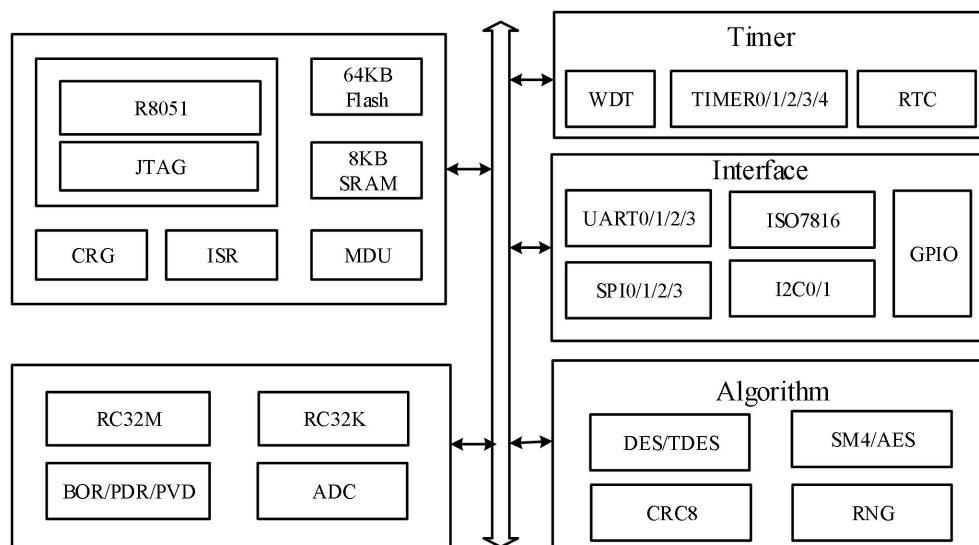


图 1-1 芯片功能框图

1.2. 电源框图

RJM8L151 芯片供电方案如下：供电电压在 1.8~5.0V 之间，外部电源给 VCC 供电。

RJM8L151 支持 Deep Sleep 工作模式，通过设置 ALWAYS_SLEEP 寄存器的值为 0xE2，可以使芯片进入 Deep Sleep 模式。在此模式，内部 LDO15 被关闭，整体功耗约 2nA。通过外部管脚唤醒（P04~07）。

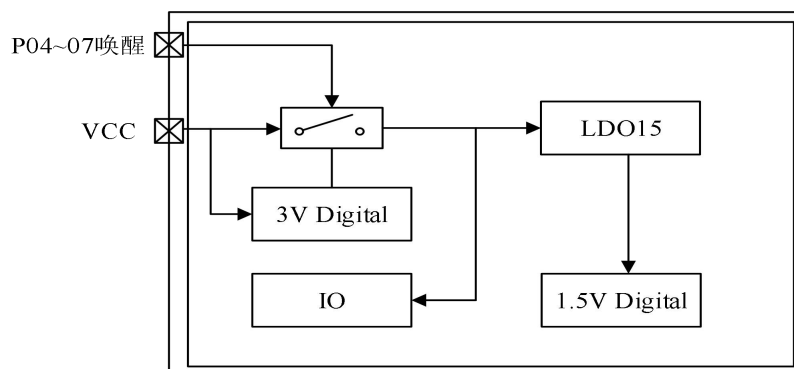


图 1-2 电源框图

2. 处理器

2.1. 概述

CPU 是一颗具有超高性能 8 位架构微控制器。支持紧凑指令译码器和指令执行机，支持多种低功耗模式，适合超低功耗 MCU 应用的场景。

2.2. 主要特性

- 8 位 ALU（加法、减法、逻辑运算和位操作）
- 紧凑指令译码器和指令执行机
- 主频 16M
- 支持 2 个 DPTR

2.3. 内部数据存储器

2.3.1. 介绍

CPU 有 256 字节的内部数据存储器，包含 128 字节的特殊功能寄存器。

地址低于 80H 的 128 字节数据可以直接寻址，也可以间接寻址。间接寻址高于 7FH 的地址访问高 128 字节内部数据内存，而直接寻址高于 7FH 的地址访问 SFR 空间。

低 128 字节包括工作寄存器组（00H...1FH）位寻址空间（20H...2FH）。最低 32 字节共 4 个组，每个组由 8 个寄存器组成（R0-R7），通过 PSW 的两个位来选择使用哪一组寄存器。往上 16 字节为位寻址空间，可通过 00H-7FH 地址访问。

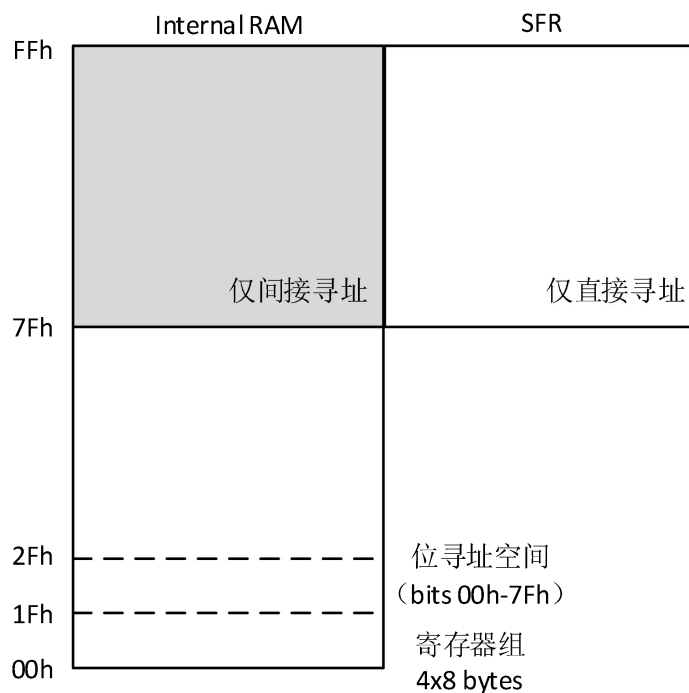


图 2-1 内存映射

2.3.2. 特殊功能寄存器

表格 2-1 特殊功能寄存器位置

0xF8	-	-	-	-	-	-	-	-
0xF0	B	-	-	-	-	-	-	SRST
0xE8	-	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
0xE0	ACC	-	-	-	-	-	-	-
0xD8	ADCON	-	-	-	-	-	-	-
0xD0	PSW	IEN4	-	-	-	-	-	-
0xC8	T2CON	IEN3	CRCL	CRCH	TL2	TH2	-	-
0xC0	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	-
0xB8	IP/IEN1	IP1	S0RELH	S1RELH	-	-	-	-
0xB0	P3	-	-	-	-	-	-	-
0xA8	IEN0	IP0	S0RELL	-	-	-	-	-
0xA0	P2	-	-	-	P0_OEN	P1_OEN	P2_OEN	P3_OEN
0x98	S0CON	S0BUF	IEN2	S1CON	S1BUF	S1RELL		
0x90	P1	P4	DPS	DPC	PAGESEL	D_PAGESEL	P4_OEN	-
0x88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	P1CON
0x80	P0	SP	DPL	DPH	DPL1	DPH1	WDTREL	PCON

3. FLASH 程序存储器和数据

3.1. 概述

RJM8L151 系列单片机内集成了 64KB 的 FLASH 存储器，用于保存芯片所有的关键脱机信息和数据。由一组寄存器来控制，用户可以使用这些寄存器来编程或擦除存储器的内容、读保护或者配置特定的低功耗模式。

3.2. 术语

- 扇区（sector）

一个扇区指可由一个简单编程操作编程或擦除的一组字节，其操作是标准的编程和擦除操作。

每个扇区大小为 128 字节。

3.3. FLASH 主要特性

- RJM8L151 FLASH 最多至 128K 字节的 FLASH 程序存储器，不同的器件容量有所不同。
- 编程模式
 - DMA 编程
 - 在每次编程之前进行擦除操作
- 保护特性
 - 使能存储器保护，芯片上电启动后进入保护模式，保护等级分为三级（高级别保护优先）。
- 在系统设置为低电量等待（Low Power Wait）和低电量运行（Low Power Run）模式下，存储器可配置为低功耗模式。

3.4. 存储器组织结构

RJM8L151 的 FLASH 存储器组织机构映射

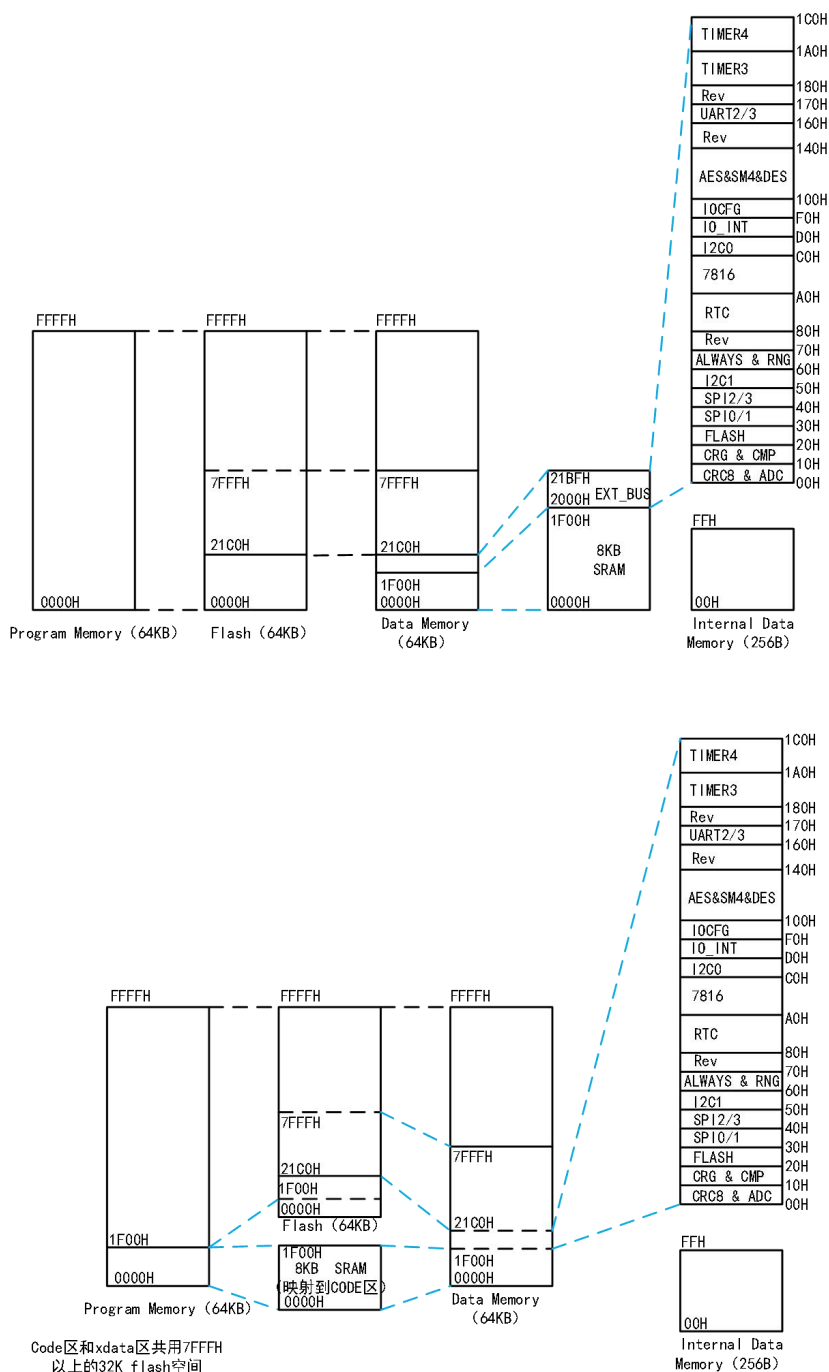


图 3-1 存储器地址映射

3.4.1. 数据区 (XDATA)

数据 (XDATA) 区域可用于存储用户具体项目所需的数据。

3.4.2. 主程序区 (CODE)

主程序区是指在 FLASH 程序存储器中用于存储应用代码的区域。

3.5. 存储器保护

使能存储器保护功能后，芯片上电启动后会进入保护模式，根据保护数据来使能保

护级别，分三级（高级别保护优先）：

一级：无保护；

二级：可以连接仿真器和在 SRAM 中运行，但是，禁止仿真器和代码在 SRAM 中运行时读写 Flash；

三级：禁止仿真器连接和在 SRAM 中运行；

Memory 保护机制如下：在 memory 保护模式下，Flash 数据不可读；若想通过改变保护权限来读取 Flash 数据，则在改写 Memory 保护权限位时，硬件会擦除 Flash，这样确保在 Memory 保护模式下，Flash 数据不被读出。

3.6. FLASH 寄存器

寄存器基地址：0x2000

偏置	名称	描述
0x21	FLASH_CTRL	Flash 控制寄存器
0x29	FLASH_DMA_CTRL0	FlashDMA 控制寄存器 0
0x2A	FLASH_DMA_CTRL1	FlashDMA 控制寄存器 1
0x2B	FLASH_DMA_CTRL2	FlashDMA 控制寄存器 2
0x2C	FLASH_DMA_CTRL3	FlashDMA 控制寄存器 3
0x2E	FLASH_DMA_CTRL4	FlashDMA 控制寄存器 4
0x12	SCREMAP	SRAM 映射位

3.6.1. 控制寄存器 FLASH_CTRL (偏移：21H)

比特	名称	属性	默认值	功能描述
7	RSV	-	-	保留
6	DMA_CODE_TO_DATA	RW	0x0	DMA 搬运 Flash 数据到 DATA 区 SRAM 使能位 0：禁止 1：使能 硬件清 0；
5	DMA_PRO_DATA	RW	0x0	DMA 编程 XDATA 区 Flash 使能位 0：禁止 1：使能 软件置 1，硬件清 0；
4	PROG_SECT_EN	RW	0x0	Sector 编程使能位 0：禁止 1：使能 软件置 1，硬件清 0；
3	DMA_AREA_SEL	RW	0x0	DMA 操作选择位 0：选择 XDATA 区 1：选择 CODE 区

2	DMA_PROG_EN	RW	0x0	DMA 编程使能位 0: 禁止 1: 使能 软件置 1 硬件清 0
1	DMA_ERASE_EN	RW	0x0	DMA 擦使能位 0: 禁止 1: 使能 软件置 1 硬件清 0
0	SECT_ERASE_EN	RW	0x0	擦 sector 使能位 0: 禁止 1: 使能 软件置 1 硬件清 0

3.6.2. DMA 源地址高 8 位 FLASH_DMA_CTRL0(偏移: 29H)

比特	名称	属性	默认值	功能描述
7:0	FLASH_DMA_CTRL0	RW	0x00	源地址高 8 位

3.6.3. DMA 源地址低 8 位 FLASH_DMA_CTRL1(偏移: 2AH)

比特	名称	属性	默认值	功能描述
7:0	FLASH_DMA_CTRL1	RW	0x00	源地址低 8 位

3.6.4. DMA 目的地址高 8 位 FLASH_DMA_CTRL2(偏移: 2BH)

比特	名称	属性	默认值	功能描述
7:0	FLASH_DMA_CTRL2	RW	0x00	目的地址高 8 位

3.6.5. DMA 目的地址低 8 位 FLASH_DMA_CTRL3(偏移: 2CH)

比特	名称	属性	默认值	功能描述
7:0	FLASH_DMA_CTRL3	RW	0x00	目的地址低 8 位

3.6.6. DMA 传输长度 FLASH_DMA_CTRL4(偏移: 2EH)

比特	名称	属性	默认值	功能描述
7	RSV	-	-	保留
6:0	FLASH_DMA_CTRL4	RW	0x00	DMA 传输长度

3.6.7. SCREMAP 控制寄存器 SCRREMAP (偏移: 12H)

比特	名称	属性	复位值	描述
7:1	RSV	-	-	保留
0	SCREMAP	RW	0x0	SRAM 映射位 写 1: 将 8K SRAM 映射到 Code 区 (0~0x2000) ;

3.7. FLASH 操作说明

FLASH 操作采用 DMA 方式编程操作，在编程之前需先将要编程的区域擦除，然后进行编程操作。

3.7.1. 擦除操作

擦除操作会擦除目标地址所在的一个 Sector，共 128Byte。

- 1) 配置 DMA 目的地址寄存器 FLASH_DMA_CTRL2 和 FLASH_DMA_CTRL3;
- 2) 配置 FLASH_CTRL 寄存器为 **0x23**（擦除 XDATA 区）。

3.7.2. 编程操作

- 1) 配置 DMA 目的地址寄存器 FLASH_DMA_CTRL2 和 FLASH_DMA_CTRL3。
- 2) 配置 DMA 源地址寄存器 FLASH_DMA_CTRL0 和 FLASH_DMA_CTRL1。
- 3) 配置 FLASH_DMA_CTRL4，确定编程长度（最长 128Byte）。
- 4) 使能 DMA 编程；配置 FLASH_CTRL 为 **0x34**（编程 XDATA 区）。

3.8. Memory 保护配置

Memory 保护操作是基于 flash 操作流程，相应地址和值的配置。配置对应保护位为固定值，则 Memory 保护生效；将对应保护位擦除，则 Memory 保护失效。

二级保护位对应地址为 **0xFF01H**，设置值位 **0x5A**；

三级保护位对应地址为 **0xFF02H**，设置值为 **0xA5**；

注：在使用保护功能时，不能重复在 OPT 区写入特征值。

3.8.1. 保护操作

- 1) 配置 DMA 目的地址寄存器 FLASH_DMA_CTRL2 和 FLASH_DMA_CTRL3;
- 2) 配置 DMA 源地址寄存器 FLASH_DMA_CTRL0 和 FLASH_DMA_CTRL1。
- 3) 配置 FLASH_DMA_CTRL4，确定编程长度 1 字节。
- 4) 使能 DMA 编程，配置 FLASH_CTRL 为 **0xB4**（编程 OTP 区）。

4. 电源检测控制

本节适用所有型号的 RJM8L151 设备。

4.1. 电源

设备需要一个 1.8V 到 5.0V 电压供应 (VDD)。

4.2. 上电复位 (POR) /掉电复位(PDR)

该设备有一个集成的 POR 电路，操作允许在 1.67V。在上电过程中，当 VDD 低于规定的阈值 VPOR 时，设备保持复位模式，不需要外部复位。POR 的特性始终启用，POR 的阈值为 1.67V。在掉电过程中，当 VDD 低于规定的阈值 VPDR 时，设备保持复位模式，不需要外部复位。PDR 的特性可一直开启，PDR 的阈值为 1.56V。

在 1.67V 到 3.3V 中，BOR 开始工作在 PDR 阈值之上。为了确保最低工作电压 (1.67V)，BOR 应设置为最低档位 0。

4.3. 欠压复位 (BOR)

在 1.7V 到 2.8V 工作时，启用 BOR 时，需要配置 LVD_CFG 寄存器的第 0-2 比特位配置 BOR 的阈值。分别有 5 个不同等级可以选择，从 1.7V 到 2.8V。当 VDD 大于设置的阈值时，MCU 正常工作。

如果 BOR 被禁用，则在电源关闭时的阈值为最低档位，保证 VDD 最小工作电压为 1.65V。

启用 BOR 时确保电压大于设置阈值，否则 MCU 无法运行。

4.4. 可编程电压检测 (PVD)

该设备具有一个嵌入式可编程电压检测器 (PVD)，监控 VDD 电源，并将与其 Vpvd 阈值进行比较。

当 VDD 低于 Vpvd 阈值时或者当 VDD 高于 Vpvd 阈值时，会产生一个指示位。通过指示位来确定电压是否超过设定阈值 (SCRSTEN 寄存器的第 7 比特位)。软件通过在 SCRSTEN 寄存器的第 4 比特位设置使能启用电压检测 PVD。

PVD 在 1.8V 和 3.0V 之间有 7 个不同档位电平，由 LVD_CFG 寄存器的第 3-5 比特位配置 PVD 的选择阈值。

4.5. 电压检测控制寄存器

寄存器基地址：0x2000

偏置	名称	描述
----	----	----

0x16	SCRSTEN	电压检测及复位使能寄存器
0xF6	LVD_CFG	电压检测档位选择寄存器

4.5.1. 电压检测及复位使能寄存器 SCRSTEN(偏移: 16H)

比特	名称	属性	复位值	描述
7	PVD_IN	RW	0x0	可编程电压监测指示位 0: VCC<Vth 1: VCC>Vth
6	BOR_EN	RW	0x0	欠压电压检测使能位 0: 禁止 1: 使能
5	PDR_EN	RW	0x1	掉电电压检测使能位 0: 禁止 1: 使能
4	PVD_EN_REG	RW	0x0	可编程电压检测使能 0: 禁止 1: 使能
3	BOR_RST_EN	RW	0x0	欠压是否复位整个芯片位 0: 禁止 1: 复位
2	PDR_RST_EN	RW	0x1	掉电是否复位整个芯片位 0: 禁止 1: 复位
1	WDT_RST_EN	RW	0x0	WDT RST 是否复位整个芯片 0: 禁止 1: 复位
0	ISO7816S_RST_EN	RW	0x0	7816S RST 是否复位整个芯片 0: 禁止 1: 复位

4.5.2. 电压检测档位选择寄存器 LVD_CFG(偏移: F6H)

比特	名称	属性	复位值	描述
7: 6	RSV	-	-	保留
5: 3	LVD_CFG_H	RW	0x0	PVD 档位表 Vth: 000: 1.8V 001: 2.0V 010: 2.2V 011: 2.4V 100: 2.6V 101: 2.8V

				11x: 3.0V 当 VCC>Vth,SCRSTEN 的 PVD_IN 位读出值 1
2:0	LVD_CFG_L	RW	0x0	BOR 档位表 Vth: 000: 1.7V 001: 1.9V 010: 2.3V 011: 2.6V 1xx: 2.8V 当 VCC<Vth,若使能 SCRSTEN 的 BOR_EN 位, 且使能 SCRSTEN 的 BOR_RST_EN 位, 系统复位

注：在配置欠压检测（BOR）功能时，应先配置 BOR 检测使能（即 SCRSTEN 的 bit6），然后再配置欠压复位（即 SCRSTEN 的 bit3）。

4.6. 电压检测操作说明

4.6.1. PVD 初始化流程

- 1) 配置 SCRSTEN 寄存器的 bit4 为 1，使能可编程电压检测功能。
- 2) 配置 CRG_LVD_CFG 的 bit3-bit5，选择电压检测档位。

4.6.2. BOR 初始化流程

- 1) 配置 SCRSTEN 寄存器的 bit6 为 1，使能欠压电压检测功能。
- 2) 配置 CRG_LVD_CFG 的 bit3-bit5，选择欠压电压档位。
- 3) 配置 SCRSTEN 寄存器的 bit3 为 1，使能欠压复位芯片功能。

4.6.3. PDR 初始化流程

- 1) 配置 SCRSTEN 寄存器的 bit5 为 1，使能掉电电压检测功能。
- 2) 配置 SCRSTEN 寄存器的 bit2 为 1，使能掉电复位芯片功能。

5. 低功耗模式

5.1. 概述

在默认情况下，微处理器在系统或电源复位后处于正常的工作模式，但是该设备支持 6 种低功耗模式，以实现节省功耗。

- **Deep Sleep 模式^①**，CPU 及外设均处于掉电状态，系统此时只有 4 个 IO 处于正常工作状态。4 个 IO 带输出上下拉功能，可配置为唤醒源，即 Deep Sleep 模式下唤醒源最多有 4 个，且唤醒极性可配置。系统从 Deep Sleep 模式唤醒后，CPU、SRAM、FLASH 及外设重新上电，系统从 0 地址执行程序。
- **Halt 模式**，系统所有时钟停止，系统不掉电，RAM 处于保持状态。Halt 模式下，可以由外设进入工作状态唤醒（ISO7816MS、I2C Slave、Spi Slave、Uart、GPIO 及 COMP）。唤醒过程中不涉及重新上下电，CPU 从上次进入 Halt 低功耗处接着运行，唤醒时间<5us。
- **Active Halt 模式**，CPU 和外设时钟停止，但 RTC 时钟开启，除 Halt 模式下列出的唤醒源外，RTC 中断也可将系统从 Active-Halt 模式唤醒到 Active 正常工作模式。
- **Wait 模式**，此模式下，除 CPU 时钟停止外，其他模块均正常工作，系统工作在内部高速 OSC（32M 的 2 分频）或外部高速晶振下。外设中断可将系统从 Wait 模式唤醒到正常工作模式。和 Halt 模式相比，Wait 模式唤醒时间非常短，唤醒源有效后，只需 3 个周期系统时钟可切换到 Active 模式。
- **Low Power Wait 模式**，与 Wait 模式相比，Low Power Wait 模式系统工作在内部低速时钟（32M 时钟分频到 1M 或 32K 内部低速时钟）或外部低速时钟下。
- **Low Power Run 模式**，CPU 和选定的外设运行，程序运行在 SRAM 中，此时时钟为低速时钟。Flash、32MHZ RCOSC、VR 等处于低功耗模式。

注：1、除 Deep Sleep 模式，在其他低功耗模式下，系统初始化时需要配置 COMP_CTRL1=0x90；
2、进入 DeepSleep 模式前，需将所有引脚功能配置为 GPIO 输入模式。避免因干扰造成误触发唤醒芯片。

5.2. DeepSleep

5.2.1. DeepSleep 模式唤醒源

唤醒源	唤醒方式
P04	高电平或低电平唤醒
P05	高电平或低电平唤醒
P06	高电平或低电平唤醒
P07	高电平或低电平唤醒

配置 SCCM1 寄存器的第 7 比特位，使能低功耗控制模块；配置 ALWAYS_IO_EN 寄存器的第 4 至 7 比特位，选择唤醒源 PAD 的唤醒极性并通过 ALWAYS_IO_EN 寄存器的第 0 至 3 比特位，设置输出不使能；根据唤醒源 PAD 的极性，配置 ALWAYS_IO_PU 寄存器设置唤醒源 PAD 上拉或者下拉（例：PAD 设置高电平唤醒，则配置下拉）；配置 ALWAYS_SLEEP_SRC 寄存器选择相应的唤醒源 PAD。

5.2.2. 进入 DeepSleep 模式

在寄存器 ALWAYS_SLEEP 中写入 0xE2，则系统进入此模式。

注：1、进入 DeepSleep 模式前，需将所有引脚功能配置为 GPIO 输入模式。

2、进入 DeepSleep 模式后，SRAM 掉电数据丢失。

5.2.3. DeepSleep 模式唤醒

在设置完唤醒源及系统进入 DeepSleep 后，系统等待有效唤醒源信号，当系统等待到有效唤醒信号后，系统退出 DeepSleep 模式，此时 CPU 从 0 地址重新开始运行。

5.2.4. DeepSleep 模式寄存器

偏置	名称	描述
0x61	ALWAYS_IO_EN	P04、P05、P06、P07 输出控制及唤醒极性控制寄存器
0x62	ALWAYS_IO_PU	P04、P05、P06、P07 PU/PD 控制寄存器
0x63	ALWAYS_SLEEP_SRC	SLEEP 唤醒源选择寄存器
0x65	ALWAYS_SLEEP_ID	DeepSleep 配置寄存器

5.2.4.1. P04、P05、P06、P07 输出控制及唤醒极性控制寄存器 ALWAYS_IO_EN(偏置：61H)

比特	名称	属性	复位值	描述
7	Wakeup_io_polar_3	WO	0x0	P07 唤醒极性选择位 0：低电平时唤醒 1：高电平时唤醒
6	Wakeup_io_polar_2	WO	0x0	P06 唤醒极性选择位 0：低电平时唤醒 1：高电平时唤醒
5	Wakeup_io_polar_1	WO	0x0	P05 唤醒极性选择位 0：低电平时唤醒 1：高电平时唤醒
4	Wakeup_io_polar_0	WO	0x0	P04 唤醒极性选择位 0：低电平时唤醒 1：高电平时唤醒
3	P07_oen	WO	0x0	P07 输出使能 0：不使能

				1: 使能
2	P06_oen	WO	0x0	P06 输出使能 0: 不使能 1: 使能
1	P05_oen	WO	0x0	P05 输出使能 0: 不使能 1: 使能
0	P04_oen	WO	0x0	P04 输出使能 0: 不使能 1: 使能

5.2.4.2. P04、P05、P06、P07 PU/PD 控制寄存器 ALWAYS_IO_PU(偏置: 62H)

比特	名称	属性	复位值	描述
7	P07_pd	WO	0x0	P07 下拉电阻使能 0: 不使能 1: 使能
6	P07_pu	WO	0x0	P07 上拉电阻使能 0: 不使能 1: 使能
5	P06_pd	WO	0x0	P06 下拉电阻使能 0: 不使能 1: 使能
4	P06_pu	WO	0x0	P06 上拉电阻使能 0: 不使能 1: 使能
3	P05_pd	WO	0x0	P05 下拉电阻使能 0: 不使能 1: 使能
2	P05_pu	WO	0x0	P05 上拉电阻使能 0: 不使能 1: 使能
1	P04_pd	WO	0x0	P04 下拉电阻使能 0: 不使能 1: 使能
0	P04_pu	WO	0x0	P04 上拉电阻使能 0: 不使能 1: 使能

5.2.4.3. SLEEP 唤醒源选择寄存器 ALWAYS_SLEEP_SRC (偏置: 63H)

比特	名称	属性	复位值	描述
7:4	RSV	-	-	保留

3:0	Sleep_src	WO	0x01	芯片唤醒源选择：为 1 表示选中该唤醒源 [3]:P07 [2]:P06 [1]:P05 [0]:P04
-----	-----------	----	------	--

5.2.4.4. DeepSleep 配置寄存器 ALWAYS_SLEEP_ID(偏置：65H)

比特	名称	属性	复位值	描述
7:0	Sleep	WO	0x00	写入 0xE2 进入 Deep Sleep 状态,其他值无效

5.3. Halt

5.3.1. Halt 模式唤醒源

唤醒源	唤醒方式
ISO7816MS	有效的 C2_RST 信号 (>5us)
I2C0/1	SDA 或 SCL 为低 (>5us)
SPI0/1/2/3	SPI 做从机时 CS 被选中 (>5us)
UART0/1/2/3	RX 为低 (>5us)
GPIO0/1/2/3/4	GPIO 输入为高 (>5us)
COMP1/2	比较器 COMP1/COMP2 为高 (>5us)

在 Halt 模式下，唤醒源为 GPIO，配置 CRG_PORT_SEL0 或 CRG_PORT_SEL1 或 CRG_PORT_SEL2 或 CRG_PORT_SEL3 或 CRG_PORT_SEL4 选择唤醒源的端口（例：选择 gpio10 为 Halt 唤醒源，则配置 CRG_PORT_SEL1 的第 0 比特位为 1）。其他唤醒源是在使用该外设资源时自动唤醒系统，无需再配置其他寄存器。

5.3.2. 进入 Halt 模式

系统进入 Halt 模式，配置 SCSYS 寄存器的第 2 比特位，则系统进入此模式，此时 CPU 停止运行。

5.3.3. Halt 模式唤醒

在设置完唤醒源及系统进入 Halt 后，系统等待有效唤醒源信号，当系统等待到有效唤醒信号后，系统退出 Halt 模式，此时 CPU 从上次停止运行处开始运行。

5.3.4. Halt 模式寄存器

低功耗模式基地址：0x2000

地址	名称	描述
0xF1	CRG_PORT_SEL0	Halt 模式 GPIO 唤醒源选择寄存器 0
0xF2	CRG_PORT_SEL1	Halt 模式 GPIO 唤醒源选择寄存器 1
0xF3	CRG_PORT_SEL2	Halt 模式 GPIO 唤醒源选择寄存器 2
0xF4	CRG_PORT_SEL3	Halt 模式 GPIO 唤醒源选择寄存器 3
0xF5	CRG_PORT_SEL4	Halt 模式 GPIO 唤醒源选择寄存器 4

5.3.4.1. Halt 模式 GPIO 唤醒源选择寄存器 CRG_PORT_SEL0（偏移：F1H）

比特	名称	属性	复位值	描述
7:0	CRG_PORT_SEL0	RW	0x00	相应 bit 位为 1 即选择 GPIO0 中相对于 bit 作为 Halt 模式唤醒源，高电平唤醒；

5.3.4.2. Halt 模式 GPIO 唤醒源选择寄存器 CRG_PORT_SEL1（偏移：F2H）

比特	名称	属性	复位值	描述
7:0	CRG_PORT_SEL1	RW	0x00	相应 bit 位为 1 即选择 GPIO1 中相对于 bit 作为 Halt 模式唤醒源，高电平唤醒；

5.3.4.3. Halt 模式 GPIO 唤醒源选择寄存器 CRG_PORT_SEL2（偏移：F3H）

比特	名称	属性	复位值	描述
7:0	CRG_PORT_SEL2	RW	0x00	相应 bit 位为 1 即选择 GPIO2 中相对于 bit 作为 Halt 模式唤醒源，高电平唤醒；

5.3.4.4. Halt 模式 GPIO 唤醒源选择寄存器 CRG_PORT_SEL3（偏移：F4H）

比特	名称	属性	复位值	描述
7:0	CRG_PORT_SEL3	RW	0x00	相应 bit 位为 1 即选择 GPIO3 中相对于 bit 作为 Halt 模式唤醒源，高电平唤醒；

5.3.4.5. Halt 模式 GPIO 唤醒源选择寄存器 CRG_PORT_SEL4（偏移：F5H）

比特	名称	属性	复位值	描述
7:0	CRG_PORT_SEL4	RW	0x00	相应 bit 位为 1 即选择 GPIO4 中相对于 bit 作为 Halt 模式唤醒源，高电平唤醒；

5.4. Active_Halt

5.4.1. Active_Halt 模式唤醒源

唤醒源	唤醒方式
ISO7816MS	有效的 C2_RST 信号（>5us）
I2C0/1	SDA 或 SCL 为低（>5us）
SPI0/1/2/3	SPI 做从机时 CS 被选中（>5us）
UART0/1/2/3	RX 为低（>5us）
GPIO0/1/2/3/4	GPIO 输入为高（>5us）
COMP1/2	比较器 COMP1/COMP2 为高（>5us）
RTC	RTC 中断

在 Active_Halt 模式下，唤醒源为 GPIO，配置的流程与 Halt 模式下相同，不过 Active_Halt 模式下多了一种 RTC 唤醒（RTC 正常工作具体配置参考相关章节）。

5.4.2. 进入 Active_Halt 模式唤醒源

系统进入 Halt 模式，配置 SCSYS 寄存器的第 2 比特位，则系统进入此模式，此时 CPU 停止运行。

5.4.3. Active_Halt 模式唤醒

在设置完唤醒源及系统进入 Halt 后，系统等待有效唤醒源信号，当系统等待到有效唤醒信号后，系统退出 Halt 模式，此时 CPU 从上次停止运行处开始运行。

5.5. Wait

5.5.1. Wait 模式唤醒源

唤醒源	唤醒方式
GPIO0/1/2/3/4	GPIO 模块中断
TIMER0/1/2/3/4	TIMER 模块中断
UART0/1/2/3/	UART 模块中断
SPI0/1/2/3	SPI 模块中断
I2C0/1	I2C 模块中断
ISO7816MS	ISO7816MS 数据中断和 C2 RST 中断
SM4/DES/AES	加密模块中断
COMP1/2	比较器 1/2 中断
WDT	WDT 模块中断
ADC	ADC 模块中断
RTC	RTC 模块中断

在 Wait 模式下，唤醒源为外设模块的中断信号，则在配置唤醒源时注意使能该模块中断。

5.5.2. 进入 Wait 模式

系统进入 Wait 模式之前，解除配置外设软复位，并使能外设时钟，使能外设中断，使外设正常工作。配置寄存器 LP_CTRL1 的第 6 比特位，关闭 CPU 时钟，则系统进入此模式，此时 CPU 停止运行，外设正常工作。

5.5.3. Wait 模式唤醒

在设置完唤醒源及系统进入 Wait 后，系统等待外设中断信号，当系统等待到外设中断后，系统退出 Wait 模式，此时 CPU 从上次停止运行处开始运行。

5.6. Low_Power_Wait

5.6.1. Low_Power_Wait 模式唤醒源

唤醒源	唤醒方式
GPIO0/1/2/3/4	GPIO 模块中断
TIMER0/1/2/3/4	TIMER 模块中断
UART0/1/2/3/	UART 模块中断
SPI0/1/2/3	SPI 模块中断
I2C0/1	I2C 模块中断

ISO7816MS	ISO7816MS 数据中断和 C2 RST 中断
SM4/DES/AES	加密模块中断
COMP1/2	比较器 1/2 中断
WDT	WDT 模块中断
ADC	ADC 模块中断
RTC	RTC 模块中断

在 Low_Power_Wait 模式下，唤醒源为外设模块的中断信号，则在配置唤醒源时注意使能该模块中断。

5.6.2. 进入 Low_Power_Wait 模式

系统进入 Low_Powr_Wait 模式之前，解除配置外设软复位，并使能外设时钟，使能外设中断，使外设正常工作；配置 LP_CTRL1 寄存器的第 1,2,3 比特位使能 32K 时钟；配置 SCSYS 寄存器的第 0,1 比特位，设置系统时钟为低频时钟，关闭 32M 时钟；设置配置寄存器 LP_CTRL1 的第 6 比特位，关闭 CPU 时钟，则系统进入此模式，此时 CPU 停止运行，外设正常工作。

5.6.3. Low_Power_Wait 模式唤醒

在设置完唤醒源及系统进入 Low_Powr_Wait 后，系统等待外设中断信号，当系统等待到外设中断后，系统退出 Low_Powr_Wait 模式，此时 CPU 从上次停止运行处开始运行。

5.7. Low_Power_Run

5.7.1. 进入 Low_Power_Run 模式

系统进入 Low_Powr_Run 模式之前，配置 SCREMAP 寄存器的第 0 比特位，将 8K DRAM 映射到 CODE 区（将代码搬到 SRAM 中）；配置 LP_CTRL1 的第 4,5 比特位使 FLASH 处于低功耗状态；

LP_CTRL1 的 1,2,3 比特位使能 32K 时钟；配置 SCSYS 寄存器的第 0,1 比特位，设置系统时钟为低频时钟，关闭 32M 时钟。此时进入该模式，系统以 32K 时钟在 SRAM 中运行。

6. 复位（RST）

RJM8L151 系统复位将所有寄存器至它们的复位状态，共有 5 个复位源：

- NRST 引脚产生的外部复位
- 上/掉电复位
- 欠压复位
- 看门狗复位
- 模块软件复位

6.1. 复位电路

复位引脚 NRST 上给最小为 500ns 的低电平脉冲即可产生一个外部复位。

6.2. 内部复位源

每个内部复位源在复位控制器中都有与之对应的控制位，这些控制位通过软件写‘1’，来使能该内部复位源的复位功能。

6.2.1. 上/掉电复位（POR/PDR）

在上电期间，POR 保持设备处于复位状态，直到供电电压达到上电阈值电压，此时 PDR 取代 POR，POR 关闭。PDR 检测一直开启，直到供电电压小于掉电阈值电压，此时 POR 又开启工作。

6.2.2. 欠压复位（BOR）

当供电电压小于 BOR 设定的电压时，系统会产生复位。

6.2.3. 看门狗复位

详情参见看门狗章节。

6.2.4. 模块软件复位

RJM8L151 外设模块可以通过配置复位寄存器（RSTCON0）开启相关外设的软复位功能。

6.3. 复位寄存器

系统寄存器基地址：0x2000

地址	名称	描述
0x11	SCU	软复位复位 CPU 和 7816 的 C2 脚 REST 请求寄存器
0x16	SCRSTEN	安全模块复位使能寄存器
0x17	SCSSFTRST	系统软复位控制寄存器

0x18	RSTCON0	复位使能寄存器
------	---------	---------

6.3.1. 软复位复位 CPU 和 7816 的 C2 脚 REST 请求寄存器 SCU(偏移: 11H)

比特	名称	属性	复位值	描述
7:2	RSV	-	-	保留
1	SOFT_RST_CPU	WO	0x0	软复位使能位 0: 不使能软复位 1: 使能软复位复位 CPU
0	SCU_INT	RW	0x0	7816 C2 管脚复位请求 0: 7816 C2 无复位请求 1: 7816 C2 有复位请求 硬件置 1, 软件清 0

6.3.2. 安全模块复位使能寄存器 SCRSTEN(偏移: 16H)

比特	名称	属性	复位值	描述
7	PVD_IN	RW	0x0	可编程电压监测指示位 0: $VCC < V_{th}$ 1: $VCC > V_{th}$
6	BOR_EN	RW	0x0	欠压电压检测使能位 0: 禁止 1: 使能
5	PDR_EN	RW	0x1	掉电电压检测使能位 0: 禁止 1: 使能
4	PVD_EN_REG	RW	0x0	可编程电压检测使能 0: 禁止 1: 使能
3	BOR_RST_EN	RW	0x0	欠压是否复位整个芯片位 0: 禁止 1: 复位
2	PDR_RST_EN	RW	0x1	掉电是否复位整个芯片位 0: 禁止 1: 复位
1	WDT_RST_EN	RW	0x0	WDT RST 是否复位整个芯片 0: 禁止 1: 复位
0	ISO7816S_RST_EN	RW	0x0	7816S RST 是否复位整个芯片 0: 禁止 1: 复位

6.3.3. 系统软复位控制寄存器 SCSSFTRST(偏移: 17H)

比特	名称	属性	复位值	描述
7:1	RSV	-	-	保留
0	SOFT_RST_EN	RW	0x0	CPU 外的其他外软复位选择位 0: 无影响 1: 复位除 CPU 外的其他外设

6.3.4. 复位使能寄存器 RSTCON0(偏移: 18H)

比特	名称	属性	复位值	描述
7	SPI3_SRST_REQ	RW	0x1	SPI3 模块软复位使能位 0: 禁止 1: 使能
6	SPI2_SRST_REQ	RW	0x1	SPI2 模块软复位使能位 0: 禁止 1: 使能
5	SPI1_SRST_REQ	RW	0x1	SPI1 模块软复位使能位 0: 禁止 1: 使能
4	SPI0_SRST_REQ	RW	0x1	SPI0 模块软复位使能位 0: 禁止 1: 使能
3	ISO1786_SRST_REQ	RW	0x1	ISO7816 模块软复位使能位 0: 禁止 1: 使能
2	DES_SRST_REQ	RW	0x1	DES/SM4 模块软复位使能位 0: 禁止 1: 使能
1	ADC_SRST_REQ	RW	0x1	ADC 模块软复位使能位 0: 禁止 1: 使能
0	RTC_SRST_REQ	RW	0x1	RTC 模块软复位使能位 0: 禁止 1: 使能

时钟控制器功能强大，目的在于使用户在获取最好性能的同时，能保证消耗功率最低。

注：1.RTC_iclk为RTC秒计数时钟；
2.当配置GPIO和TIMER模块时钟使能时，x代表0,1,2,3,4；
当配置UART和SPI模块时钟使能时，x代表0,1,2,3；
当配置I2C模块时钟使能时，x代表0,1；

图 7-1 时钟模块框图

7.1. 主时钟源

- 16MHz 高速外部晶体振荡器 (HSE)
- 32MHz 高速内部 RC 振荡器 (HSI)
- 32.768KHz 低速外部晶体振荡器 (LSE)
- 32KHz 低速内部 RC 振荡器 (LSI)

7.1.1. HSE

高速外部时钟信号由外部晶体谐振器产生，外部 16M 振荡器其优点在于能够产生精确的占空比为 50% 的时钟信号。

HSE 高速外部时钟晶体可通过设置系统时钟选择寄存器 SCSYS 的第 3 比特位，时钟控制寄存器 SCCM2 的第 7 比特位来打开或关闭。

7.1.2. HSI

高速内部时钟信号由内部 RC 振荡器与一个可编程分频器（分频因子 2,4,32）产生。分频因子由寄存器 SCSYS 的第 6,7 比特位来决定（默认 2 分频）。

HSI RC 可通过设置系统时钟选择寄存器 SCSYS 的第 1,3 比特位，开启或关闭内部 32M 时钟。

7.1.3. LSE

32.768KHz 的外部低速外部时钟信号为外设 RTC 提供准确的时钟信号。

LSE 低速外部时钟可通过 OSC 控制寄存器 LP_CTRL1 的第 1 比特位，开启或关闭外部低速时钟。

7.1.4. LSI

32K 内部低速时钟振荡器是一个低功耗，低成本的可选主时钟源，可以为低功耗功能和 RTC 外设提供时钟。

LSI 低速内部时钟可通过 OSC 控制寄存器 LP_CTRL1 的第 2 比特位，开启或关闭内部低速时钟。

7.2. 系统时钟源选择

7.2.1. 系统启动

为了系统快速启动，复位后时钟自控使用 HSI 的 2 分频（HSI/2）作为主时钟。如有需要用户可将系统时钟切换到另外的时钟源。

7.2.2. 系统时钟切换

用户根据需要可以手动切换系统时钟，可通过配置系统时钟选择寄存器 SCSYS 的第 0,1,3,6,7 比特位，OSC 控制寄存器 LP_CTRL1 的第 1,2,3 比特位，选择系统时钟源，系统工作的时钟为 HSE 或 HSI 或 LSE 或 LSI。

注：时钟切换时，必须先开启并使能要切换的时钟，再关闭切换前的时钟。

7.3. RTC 时钟选择

RTC 时钟可通过配置寄存器 LP_CTRL1 的第 1,2 比特位选择 RTC 的时钟为内部低速或者外部低速时钟。

7.4. 外设时钟选择

除 RTC 设备外，其他的外部设备在工作时也需要使能该外设的时钟，可通过寄存器 SCCM0，SCCM1，SCCM2 开启或关闭外设时钟。

7.5. 时钟控制寄存器

时钟控制寄存器基地址：0x2000

地址	名称	描述
0x15	SCSYS	系统时钟选择寄存器
0x10	LP_CTRL1	OSC 控制寄存器
0x13	SCCM0	时钟控制寄存器 0
0x14	SCCM1	时钟控制寄存器 1
0x19	SCCM2	时钟控制寄存器 2

7.5.1. 系统时钟选择寄存器 SCSYS（偏移：15H）

比特	名称	属性	复位值	描述
7: 6	FREQ_SEL	RW	0x2	高频时钟 CLK_OUT 分频选择 00: 32 分频 01: 4 分频 10: 2 分频 11: 1 分频
5:4	RSV	-	-	保留
3	SRC_CLK_SEL	RW	0x0	CLK_32M 源头选择位 1: 选择 XT_16M（选中前通过 SCCM2 寄存器的 bit7 配使能）； 0: 选择 RC_32M（选中前通过 RC_32M_EN 配使能）；
2	Halt	RW	0x0	Halt 模式选择位 0: 无影响 1: 进入 halt 模式
1	RC_32M_EN	RW	0x0	RC_32M 时钟使能位 0: 使能 RC_32M 时钟位 1: 不使能 RC_32M 时钟位
0	CK_SEL	RW	0x0	系统时钟 clk_sys 选择位 0: RC_32MHz 1: LF_32K（选中后需通过配置 LP_CTRL1 寄存器选择 RC_32K 或 XT_32K）

7.5.2. OSC 控制寄存器 LP_CTRL 1(偏移：10H)

比特	名称	属性	复位值	描述
7	SYS_BUS_EN	RW	0x0	总线时钟使能位 0: 禁止 1: 使能

6	CPU_CK_EN	RW	0x1	CPU 时钟使能位 0: 禁止 1: 使能
5	FLASH_DPSDB	RW	0x0	Flash Standby 使能选择位 0: Flash 处于正常状态 1: Flash 处于低功耗状态
4	FLASH_CE	RW	0x0	Flash 使能选择位 0: Flash 使能 1: Flash 不使能
3	LF_32K_EN	RW	0x0	LF_32K 时钟使能位 0: 禁止 1: 使能
2	RC_32K_EN	RW	0x1	32K RC 时钟启动使能位 0: 使能 1: 禁止
1	XT_32K_EN	RW	0x0	32K XT 晶振时钟启动使能位 0: 禁止 1: 使能;
0	RSV	-	-	保留

7.5.3. 时钟控制寄存器 SCCM0 (偏移: 13H)

比特	名称	属性	复位值	描述
7	WDG_EN	RW	0x0	WDG 时钟使能控制位 0: 禁止 1: 使能
6	I2C1_EN	RW	0x0	I2C1 时钟使能控制位 0: 禁止 1: 使能
5	I2C0_EN	RW	0x0	I2C0 时钟使能控制位 0: 禁止 1: 使能
4	SPI3_EN	RW	0x0	SPI3 时钟使能控制 0: 禁止 1: 使能
3	SPI2_EN	RW	0x0	SPI2 时钟使能控制 0: 禁止 1: 使能
2	SPI1_EN	RW	0x0	SPI1 时钟使能控制 0: 禁止 1: 使能

1	SPI0_EN	RW	0x0	SPI0 时钟使能控制 0: 禁止 1: 使能
0	ISO7816_EN	RW	0x0	ISO7816 时钟使能控制位 0: 禁止 1: 使能

7.5.4. 时钟控制寄存器 SCCM1（偏移：14H）

比特	名称	属性	复位值	描述
7	ALWAYS_EN	RW	0x0	低功耗控制模块（常电区时钟）时钟使能控制位 0: 禁止 1: 使能
6	RNG_EN	RW	0x0	RNG 时钟使能控制 0: 禁止 1: 使能
5	GPIO_CK_EN	RW	0x0	GPIO 时钟使能位 0: 禁止 1: 使能
4	TIMER4_EN	RW	0x0	Timer4 时钟使能控制 0: 禁止 1: 使能
3	TIMER3_EN	RW	0x0	Timer3 时钟使能控制 0: 禁止 1: 使能
2	TIMER2_EN	RW	0x0	Timer2 时钟使能控制 0: 禁止 1: 使能
1	TIMER1_EN	RW	0x0	Timer1 时钟使能控制 0: 禁止 1: 使能
0	TIMER0_EN	RW	0x0	Timer0 时钟使能控制 0: 禁止 1: 使能

7.5.5. 时钟控制寄存器 SCCM2（偏移：19H）

比特	名称	属性	复位值	描述
7	XT_16M_EN	RW	0x0	XT_16M 时钟使能位 0: 使能; 1: 禁止;
6	ADC_EN	RW	0x0	ADC 时钟使能控制位

				0: 禁止 1: 使能
5	DES_SM4_AES_EN	RW	0x0	DES/SM4/AES 时钟使能控制位 0: 禁止 1: 使能
4	CRC_EN	RW	0x0	CRC 时钟使能控制位 0: 禁止 1: 使能
3	UART 3_EN	RW	0x0	UART 3 时钟使能控制位 0: 禁止 1: 使能
2	UART 2_EN	RW	0x0	UART 2 时钟使能控制位 0: 禁止 1: 使能
1	UART 1_EN	RW	0x0	UART 1 时钟使能控制位 0: 禁止 1: 使能
0	UART0_EN	RW	0x0	UART 0 时钟使能控制位 0: 禁止 1: 使能

8. 通用输入输出口（GPIO）及端口复用

8.1. 概述

GPIO 包含通用输入输出接口用于芯片和外部进行数据传输，一组 IO 端口包括 8 个引脚，每个引脚都可以被独立编程作为数字输入或数字输出口。另外部分端口还可以作为模拟输入，外部中断，片上外设的输入/输出等复用功能。在同一时刻仅有一个复用功能可以映射到引脚上。复用功能可通过相关寄存器进行控制（具体参考引脚描述部分）。此外，每个端口都可作为外部中断信号输入。

每个端口都分配有一个输出/输入数据寄存器，一个数据方向寄存器。

8.2. GPIO 主要特性

- 端口的各个位可以被单独配置
- 片上外设的 I/O 功能复用
- 外部中断可以单独使能和关闭（支持边沿触发和电平触发）

8.3. I/O 的配置和使用

8.3.1. 普通输入/输出 I/O

每个端口都有一个数据方向控制寄存器 Px_OEN(x=0,1,2,3,4)，控制选择数据方向。一个数据输入/输出寄存器 Px，用于输入/输出数字信号。

注：P04,P05,P06,P07 等端口，因是低功耗 DeepSleep 模式的唤醒源，所以在使用这几个管脚的输出功能时，还需要开始常电区时钟 SCCM1 的第 7 比特位，及输出控制及唤醒极性控制寄存器 ALWAYS_IO_EN 的第 0,1,2,3 比特位，控制相关引脚的数据方向。

8.3.2. 作为外部中断源

每个端口都可以作为外部中断源，但是每组 IO 端口只能用 1 个引脚。

在配置中断时，需要将端口复用配置为 GPIO 口，数据方向配置为输入，触发方式配置寄存器 Px_INT_SEL（x=0,1,2,3,4），选择边沿触发/电平触发，配置中断使能寄存器 Px_INT_EN 选择相应位的中断，此外还需配置中断服务程序单元（ISR），具体参照相关章节。

中断的清除通过寄存器 Px_INT_REG 来清除中断标志。

注：GPIO0 与 GPIO1 可支持边沿或电平触发，GPIO2，GPIO3，GPIO4 仅支持边沿触发。GPIO0 和 GPIO1 在配置电平触发是还需设置寄存器 TCON 的第 0，1 比特位。

8.4. I/O 管脚复用寄存器

芯片每个端口都有相应的复用功能，其管脚复用关系见如下管脚复用寄存器。

管脚复用寄存器基地址：0x20F0

偏置	名称	描述
0x00	IOCFG_CTRL0	管脚复用控制寄存器 0
0x07	IOCFG_CTRL1	管脚复用控制寄存器 1
0x08	IOCFG_CTRL2	管脚复用控制寄存器 2
0x09	IOCFG_CTRL3	管脚复用控制寄存器 3
0x0A	IOCFG_CTRL4	管脚复用控制寄存器 4
0x0B	IOCFG_CTRL5	管脚复用控制寄存器 5
0x0C	IOCFG_CTRL6	管脚复用控制寄存器 6
0x0D	IOCFG_CTRL7	管脚复用控制寄存器 7
0x0E	IOCFG_CTRL8	管脚复用控制寄存器 8

8.4.1. 管脚复用控制寄存器 IOCFG_CTRL0（偏移：00H）

比特	名称	属性	复位值	描述
7:6	IOCFG_CTRL0[7:6]	RW	0x0	00: jtag_tdi 01: uart0_tx 10: gpio03 11: comp02（timer2 的比较模式正相输出 2）
5:4	IOCFG_CTRL0[5:4]	RW	0x0	00: jtag_tms 01: sda0_s 10: gpio02 11: comp01（timer2 的比较模式正相输出 1）
3:2	IOCFG_CTRL0[3:2]	RW	0x0	00: jtag_tck 01: scl0_s 10: gpio01 11: comp00（timer2 的比较模式正相输出 0）
1:0	IOCFG_CTRL0[1:0]	RW	0x0	00: jtag_tdo 01: rev（保留） 10: gpio00 11: t2ex0（timer2 的外部重载/捕获输入）

注：***_s 中的 s 代表 slave（从机），***_m 中的 m 代表 master（主机），下同。

8.4.2. 管脚复用控制寄存器 IOCFG_CTRL1（偏移：07H）

比特	名称	属性	复位值	描述
7	IOCFG_CTRL1[7]	RW	0x0	0: gpio10 1: uart2_rx

6	IOCFG_CTRL1[6]	RW	0x0	0: gpio06 1: cap02 (timer2 的捕获通道 2 输入)
5:4	IOCFG_CTRL1[5:4]	RW	0x0	00: 7816_rst_s 01: gpio07 10: rev 11: cap20 (timer4 的捕获通道 0 输入)
3:2	IOCFG_CTRL1[3:2]	RW	0x0	00: 7816_clk_s 01: 7816_clk_m 10: gpio05 11: cap01 (timer2 的捕获通道 1 输入)
1:0	IOCFG_CTRL1[1:0]	RW	0x0	00: gpio04 01: uart0_rx 10: rev (保留) 11: cap00 (timer2 的捕获通道 0 输入)

8.4.3. 管脚复用控制寄存器 IOCFG_CTRL2 (偏移: 08H)

比特	名称	属性	复位值	描述
7:6	IOCFG_CTRL2[7:6]	RW	0x0	00: spi2_dout_m 01: spi2_din_s 10: gpio14 11: comp11 (timer3 的比较模式正相输出 1)
5:4	IOCFG_CTRL2[5:4]	RW	0x0	00: spi2_din_m 01: spi2_dout_s 10: gpio13 11: compn10 (timer3 的比较模式反相输出 0)
3:2	IOCFG_CTRL2[3:2]	RW	0x0	00: spi2_clk_m 01: spi2_clk_s 10: gpio12 11: comp10 (timer3 的比较模式正相输出 0)
1:0	IOCFG_CTRL2[1:0]	RW	0x0	00: gpio11 01: uart2_tx 10: rev (保留) 11: cap10 (timer3 的捕获通道 0 输入)

8.4.4. 管脚复用控制寄存器 IOCFG_CTRL3 (偏移: 09H)

比特	名称	属性	复位值	描述
7:6	IOCFG_CTRL3[7:6]	RW	0x0	00: sda0_m 01: rev (保留) 10: gpio40

				11: t2ex1 (timer3 的外部重载/捕获输入)
5:4	IOCFG_CTRL3[5:4]	RW	0x0	00: scl0_m 01: rev (保留) 10: gpio17 11: compn12 (timer3 的比较模式反相输出 2)
3	IOCFG_CTRL3[3]	RW	0x0	0: gpio41 1: t2ffl (timer3 的事件计数/门控定时输入)
2	IOCFG_CTRL3[2]	RW	0x0	0: gpio16 1: comp12 (timer3 的比较模式正相输出 2)
1:0	IOCFG_CTRL3[1:0]	RW	0x0	00: gpio15 01: spi2_cs 10: rev (保留) 11: compn11 (timer3 的比较模式反相输出 1)

8.4.5. 管脚复用控制寄存器 IOCFG_CTRL4 (偏移: 0AH)

比特	名称	属性	复位值	描述
7	IOCFG_CTRL4[7]	RW	0x0	0: gpio46 1: uart1_tx
6	IOCFG_CTRL4[6]	RW	0x0	0: gpio23 (ADC_IN7) 1: spi0_cs
5:4	IOCFG_CTRL4[5:4]	RW	0x0	00: spi2_dout_m 01: spi2_din_s 10: gpio22 11: rev (保留)
3:2	IOCFG_CTRL4[3:2]	RW	0x0	00: spi0_din_m 01: spi0_dout_s 10: gpio21 (ADC_REFP) 11: cap22 (timer4 的捕获通道 2 输入)
1:0	IOCFG_CTRL4[1:0]	RW	0x0	00: spi0_clk_m 01: spi0_clk_s 10: gpio20 (ADC_REFN) 11: cap21 (timer4 的捕获通道 1 输入)

8.4.6. 管脚复用控制寄存器 IOCFG_CTRL5 (偏移: 0BH)

比特	名称	属性	复位值	描述
7:6	IOCFG_CTRL5[7:6]	RW	0x0	00: gpio27 (ADC_IN3) 01: spi1_cs 10: rev (保留)

				11: compn21 (timer4 的比较模式反相输出 1)
5:4	IOCFG_CTRL5[5:4]	RW	0x0	00: spi1_dout_m 01: spi1_din_s 10: gpio26 (ADC_IN4) 11: comp21 (timer4 的比较模式正相输出 1)
3:2	IOCFG_CTRL5[3:2]	RW	0x0	00: spi1_din_m 01: spi1_dout_s 10: gpio25 (ADC_IN5) 11: compn20 (timer4 的比较模式反相输出 0)
1:0	IOCFG_CTRL5[1:0]	RW	0x0	00: spi1_clk_m 01: spi1_clk_s 10: gpio24 (ADC_IN6) 11: comp20 (timer4 的比较模式正相输出 0)

8.4.7. 管脚复用控制寄存器 IOCFG_CTRL6 (偏移: 0CH)

比特	名称	属性	复位值	描述
3:2	IOCFG_CTRL6[3:2]	RW	0x0	00: gpio45 (ADC_IN1) 01: uart1_rx 10: rev (保留) 11: compn22 (timer4 的比较模式反相输出 2)
1:0	IOCFG_CTRL6[1:0]	RW	0x0	00: 7816_sio 01: rev (保留) 10: gpio44 (ADC_IN2) 11: comp22 (timer4 的比较模式正相输出 2)

8.4.8. 管脚复用控制寄存器 IOCFG_CTRL7 (偏移: 0DH)

比特	名称	属性	复位值	描述
7:6	IOCFG_CTRL7[7:6]	RW	0x0	00: gpio33 01: uart3_tx 10: rev (保留) 11: timer3_bkin (timer3 的刹车输入)
5:4	IOCFG_CTRL7[5:4]	RW	0x0	00: gpio32 01: uart3_rx 10: rev (保留) 11: t2ff2 (timer4 的事件计数/门控定时输入)
3:2	IOCFG_CTRL7[3:2]	RW	0x0	00: sda1_s 01: sda1_m 10: gpio31 11: t2ex2 (timer4 的外部重载/捕获输入)
1:0	IOCFG_CTRL7[1:0]	RW	0x0	00: scl1_s 01: scl1_m

				10: gpio30 11: t2ff0 (timer2 的事件计数/门控定时输入)
--	--	--	--	---

8.4.9. 管脚复用控制寄存器 IOCFG_CTRL8 (偏移: 0EH)

比特	名称	属性	复位值	描述
7	IOCFG_CTRL8[7]	RW	0x0	Rev (保留)
6	IOCFG_CTRL8[6]	RW	0x0	0: gpio37 1: spi3_cs
5:4	IOCFG_CTRL8[5:4]	RW	0x0	00: spi3_dout_m 01: spi3_din_s 10: gpio36 11: rev (保留)
3:2	IOCFG_CTRL8[3:2]	RW	0x0	00: spi3_din_m 01: spi3_dout_s 10: gpio35 11: rev (保留)
1:0	IOCFG_CTRL8[1:0]	RW	0x0	00: spi3_clk_m 01: spi3_clk_s 10: gpio34 11: timer4_bkin (timer4 的刹车输入) timer4 的刹车输入也可来自 comp2 的输出, comp2 的优先级高

8.4.10. GPIO 功能复用表

库函数中定义的 GPIO 功能对应于下表的 AF0/

引脚序号	引脚名称	复用功能寄存器配置			
		AF0	AF1	AF2	AF3
1	P10	GPI010	uart2_rx		
2	P11	GPI011	uart2_tx	rev (保留)	cap10(timer3 的捕获通道 0 输入)
3	P12	SPI2_CLK_M	spi2_clk_s	gpio12	comp10 (timer3 的比较模式正相输出 0)
4	P13	SPI2_Master_IN	SPI2_Slave_OUT	gpio13	compn10(timer3 的比较模式反相输出 0)
5	P14	SPI2_Master_OUT	SPI2_Slave_IN	gpio14	comp11 (timer3 的比较模式正相输出 1)
6	P15	GPI015	spi2_cs	rev (保留)	compn11(timer3 的比较模式反相输出 1)
7	P16	GPI016	comp12 (timer3 的比较模式正相输出 2)		

8	P17	IIC0_SCL_M	rev (保留)	gpio17	compn12(timer3 的比较模式反相输出 2)
9	P40	IIC0_SDA_M	rev (保留)	gpio40	t2ex1(timer3 的外部重载/捕获输入)
10	P41	GPI041	t2ff1(timer3 的事件计数/门控定时输入)		
11	NC	-			
12	P42	GPI042(timer3 捕获通道 1)			
13	P43	GPI043(timer3 捕获通道 2)			
14	P20	SPI0_CLK_M	SPI0_CLK_S	gpio20 (ADC_REFN)	cap21(timer4 的捕获通道 1 输入)
15	P21	SPI0_Master_IN	SPI0_Slave_OUT	gpio21 (ADC_REFP)	cap22(timer4 的捕获通道 2 输入)
16	P22	SPI0_Master_OUT	SPI0_Slave_IN	gpio22	
17	P23	GPI023(ADC_IN7)	spi0_cs		
18	P24	SPI1_CLK_M	SPI1_CLK_S	gpio24 (ADC_IN6)	comp20(timer4 的比较模式正相输出 0)
19	P25	SPI1_Master_IN	SPI1_Slave_OUT	gpio25 (ADC_IN5)	compn20(timer4 的比较模式反相输出 0)
20	P26	SPI1_Master_OUT	SPI1_Slave_IN	gpio26 (ADC_IN4)	comp21(timer4 的比较模式正相输出 1)
21	P27	GPI027(ADC_IN3)	spi1_cs	rev (保留)	compn21(timer4 的比较模式反相输出 1)
22	P44	7816_SIO	rev (保留)	gpio44 (ADC_IN2)	comp22(timer4 的比较模式正相输出 2)
23	P45	GPI045(ADC_IN1)	uart1_rx	rev (保留)	compn22(timer4 的比较模式反相输出 2)
24	P46	GPI046	uart1_tx		
25	P47	GPI047			
26	P30	IIC1_SCL_S	IIC1_SCL_M	gpio30	t2ff0(timer2 的事件计数/门控定时输入)
27	P31	IIC1_SDA_S	IIC1_SDA_M	gpio31	t2ex2(timer4 的外部重载/捕获输入)
28	P32	GPI032	uart3_rx	rev (保留)	t2ff2(timer4 的事件计数/门控定时输入)
29	P33	GPI033	uart3_tx	rev (保留)	timer3_bkin(timer3 的刹车输入)
30	P34	SPI3_CLK_M	SPI3_CLK_S	gpio34	timer4_bkin(timer4 的刹车输入)
31	P35	SPI3_Master_IN	SPI3_Slave_OUT	gpio35	

32	P36	SPI3_Master_OUT	SPI3_Slave_IN	gpio36	
33	P37	GPI037	spi3_cs		
34	X32M IN	X32MIN			
35	X32M OUT	X32MOUT			
36	X32K OUT	X32KOUT			
37	X32K IN	X32KIN			
38	P00	JTAG_TDO	rev (保留)	gpio00	t2ex0(timer2 的外部 重载/捕获输入)
39	P01	JTAG_TCK	sc10_s	gpio01	comp00 (timer2 的比 较模式正相输出 0)
40	P02	JTAG_TMS	sda0_s	gpio02	comp01 (timer2 的比 较模式正相输出 1)
41	P03	JTAG_TDI	uart0_tx	gpio03	comp02 (timer2 的比 较模式正相输出 2)
42	P04	GPI004	uart0_rx	rev (保留)	cap00(timer2 的捕获 通道 0 输入)
43	P05	7816_CLK_S	7816_clk_m	gpio05	cap01(timer2 的捕获 通道 1 输入)
44	P06	GPI006	cap02 (timer2 的捕获通 道 2 输入)		
45	P07	7816_RST_S	gpio07	rev	cap20(timer4 的捕获 通道 0 输入)
46	RST	RST			
47	VSS	VSS			
48	VDD	VDD			

8.5. GPIO 寄存器

GPIO 特殊功能寄存器

偏置	名称	描述
0x80	P0	GPIO0 数据寄存器
0x90	P1	GPIO1 数据寄存器
0xA0	P2	GPIO2 数据寄存器
0xB0	P3	GPIO3 数据寄存器
0x91	P4	GPIO4 数据寄存器
0xA4	P0_OEN	GPIO0 输出使能寄存器

0xA5	P1_OEN	GPIO1 输出使能寄存器
0xA6	P2_OEN	GPIO2 输出使能寄存器
0xA7	P3_OEN	GPIO3 输出使能寄存器
0x96	P4_OEN	GPIO4 输出使能寄存器

GPIO 外部中断寄存器基地址：0x2000

偏置	名称	描述
0xD0	P0_INT_EN	GPIO0 中断使能寄存器
0xD1	P0_INT_SEL0	GPIO0 口中断触发方式选择寄存器低字节
0xD2	P0_INT_SEL1	GPIO0 口中断触发方式选择寄存器中间字节
0xD3	P0_INT_SEL2	GPIO0 口中断触发方式选择寄存器高字节
0xD4	P0_INT_REG	GPIO0 中断状态寄存器
0xD5	P1_INT_EN	GPIO1 中断使能寄存器
0xD6	P1_INT_SEL0	GPIO1 口中断触发方式选择寄存器低字节
0xD7	P1_INT_SEL1	GPIO1 口中断触发方式选择寄存器中间字节
0xD8	P1_INT_SEL2	GPIO1 口中断触发方式选择寄存器高字节
0xD9	P1_INT_REG	GPIO1 中断状态寄存器
0xDA	P2_INT_EN	GPIO2 中断使能寄存器
0xDB	P2_INT_SEL0	GPIO2 口中断触发方式选择寄存器低字节
0xDC	P2_INT_SEL1	GPIO2 口中断触发方式选择寄存器中间字节
0xDD	P2_INT_SEL2	GPIO2 口中断触发方式选择寄存器高字节
0xDE	P2_INT_REG	GPIO2 中断状态寄存器
0xDF	P3_INT_EN	GPIO3 中断使能寄存器
0xE0	P3_INT_SEL0	GPIO3 口中断触发方式选择寄存器低字节
0xE1	P3_INT_SEL1	GPIO3 口中断触发方式选择寄存器中间字节
0xE2	P3_INT_SEL2	GPIO3 口中断触发方式选择寄存器高字节
0xE3	P3_INT_REG	GPIO3 中断状态寄存器
0xE4	P4_INT_EN	GPIO4 中断使能寄存器
0xE5	P4_INT_SEL0	GPIO4 口中断触发方式选择寄存器低字节
0xE6	P4_INT_SEL1	GPIO4 口中断触发方式选择寄存器中间字节
0xE7	P4_INT_SEL2	GPIO4 口中断触发方式选择寄存器高字节
0xE8	P4_INT_REG	GPIO4 中断状态寄存器

8.5.1. GPIOx 寄存器 Px（偏移：80H、90H、A0H、B0H、91H）

比特	名称	属性	复位值	描述
7:0	Px	RW	0xFF	数据寄存器 例如：GPIO1.5 输出高电平，则 P1=0x20； GPIO1.5 输入，则直接读取 P1 寄存器，

注：此处 x 为 0,1,2,3,4，下同。

8.5.2. GPIOx 输出使能寄存器 Px_OEN (偏移: A4H、A5H、A6H、A7H、96H)

比特	名称	属性	复位值	描述
7:0	Px_OEN	RW	0x00	配置 GPIOx 口的输出使能 例如: 将 GPIO2.3 配置成输出, 则 P2_OEN=0x08;

8.5.3. GPIOx 中断使能寄存器 Px_INT_EN (偏移: D0H、D5H、DAH、DFH、E4H)

比特	名称	属性	复位值	描述
7:0	Px_INT_EN	RW	0x00	配置 GPIOx 口的中断使能 例如, 将 GPIO3.4 配置成中断使能, 则 P3_INT_EN=0x10;

8.5.4. GPIOx 中断触发方式选择寄存器 Px_INT_SEL0, Px_INT_SEL1, Px_INT_SEL2, (偏移: 见寄存器列表)

比特	名称	属性	复位值	描述
7:5	Px_INT_SEL2	RW	0x00	GPIOx.7 的中断触发方式选择: 3'b000: 低电平中断 3'b001: 高电平中断 3'b010: 上升沿中断 3'b011: 下降沿中断 3'b100: 双沿中断
4:2	Px_INT_SEL2	RW	0x00	GPIOx.6 的中断触发方式选择: 3'b000: 低电平中断 3'b001: 高电平中断 3'b010: 上升沿中断 3'b011: 下降沿中断 3'b100: 双沿中断
1:0 7	Px_INT_SEL2 Px_INT_SEL1	RW	0x00	GPIOx.5 的中断触发方式选择: 3'b000: 低电平中断 3'b001: 高电平中断 3'b010: 上升沿中断 3'b011: 下降沿中断 3'b100: 双沿中断
6:4	Px_INT_SEL1	RW	0x00	GPIOx.4 的中断触发方式选择: 3'b000: 低电平中断 3'b001: 高电平中断 3'b010: 上升沿中断 3'b011: 下降沿中断

				3'b100: 双沿中断
3:1	Px_INT_SEL1	RW	0x00	GPIOx.3 的中断触发方式选择: 3'b000: 低电平中断 3'b001: 高电平中断 3'b010: 上升沿中断 3'b011: 下降沿中断 3'b100: 双沿中断
0 7:6	Px_INT_SEL1 Px_INT_SEL0	RW	0x00	GPIOx.2 的中断触发方式选择: 3'b000: 低电平中断 3'b001: 高电平中断 3'b010: 上升沿中断 3'b011: 下降沿中断 3'b100: 双沿中断
5:3	Px_INT_SEL0	RW	0x00	GPIOx.1 的中断触发方式选择: 3'b000: 低电平中断 3'b001: 高电平中断 3'b010: 上升沿中断 3'b011: 下降沿中断 3'b100: 双沿中断
2:0	Px_INT_SEL0	RW	0x00	GPIOx.0 的中断触发方式选择: 3'b000: 低电平中断 3'b001: 高电平中断 3'b010: 上升沿中断 3'b011: 下降沿中断 3'b100: 双沿中断

注: 1.{ Px_INT_SEL2, Px_INT_SEL1, Px_INT_SEL0 }, 3 个 8bit 的寄存器按高低字节顺序拼接用于配置 GPIO 中断触发方式。

2.GPIO0、GPIO1 口支持 5 种触发方式, GPIO2、GPIO3、GPIO4 仅支持上升沿、下降沿和双沿触发, 配置高电平、低电平触发时实际触发方式对应为上升沿和下降沿触发。

8.5.5. GPIOx 中断状态寄存器 Px_INT_REG(偏移: D4H、D9H、DEH、E3H、E8H)

比特	名称	属性	复位值	描述
7:0	Px_INT_REG	RW	0x00	GPIOx 中断状态寄存器 0: 无中断 1: 产生中断 硬件置 1, 软件写 1 清 0 例如: 清除 GPIO4.5 口的中断状态 P4_INT_REG=0x20;

8.6.GPIO 操作说明

8.6.1. 输入输出 IO 配置

- 1) 配置 IOCFG_CTRLx 寄存器，将引脚配置为 GPIO 功能；
- 2) 配置 Px_OEN 寄存器，选择 GPIO 方向（输入/输出）；
- 3) 设置 Px 的任意位来控制对应输出引脚的电平；
- 4) 读取 Px 来获取输入引脚电平。

8.6.2. 中断触发模式配置

中断初始化过程：

- 1) 配置 IOCFG_CTRLx 寄存器，将引脚配置为 GPIO 功能；
- 2) 设置 Px_OEN，将 GPIO 口设置为输入；
- 3) 配置寄存器 Px_INT_SEL，选择是边沿/电平触发类型；
- 4) 配置寄存器 Px_INT_EN 使能相应位中断
- 5) 配置使能 GPIOx 中断和开总中断（EA）。

8.6.3. 清除中断标志

中断处理函数通过写 Px_INT_REG 来清除中断状态。如果在清除寄存器的同时有新的边沿触发中断产生，这个新的中断将会保持有效直到下一次清除。

注：读取中断状态操作应该在对 Px_INT_EN 相应位写 0 之前进行，因为对 Px_INT_EN 相应位写 0 将清除相应中断状态。

9. 中断服务程序单元（ISR）

9.1. 概述

中断服务程序单元(ISR) 是 8051 的一个重要组成部分。它与 CPU 处理器内核紧密耦合，实现低中断延迟以及对新到中断的有效处理，外部中断信号连接到 ISR，ISR 将对这些中断进行优先级排序。

每个中断源都有自己的请求标志，它位于中断源的设备中。没有中断请求标志是直接位于 ISR。所有中断请求是通过相应高电平输入到 ISR。通过相应的使能标志位（在“ien0”、“ien1”和“ien2 ”“ien3”“ien4 ”SFR 寄存器中），每个中断源都可以单独启用或禁用。此外，所有的中断可以在全局范围内启用或禁用的“EAL”的标志位（在“ien0”SFR 寄存器中）。

所有中断源被分成 6 个中断组。每个中断组可以有 1 到 4 个中断优先级分配。中断优先级的定义是由位于“IP0”和“IP1”SFR 寄存器的标志决定。

9.2. 中断服务单元特性

- 支持 30 路向量中断
- 4 级可编程中断优先级
- 可嵌套中断支持
- 中断可屏蔽

9.3. 中断源

表格 9-1 中断源

中断号	中断入口地址	中断源	备注	使能位
Int0	03H	GPIO0	GPIO0 模块中断	IEN0.0
Int1	0bH	TIMER0	定时器 0 中断	IEN0.1
Int2	13H	GPIO1	GPIO1 模块中断	IEN0.2
Int3	1bH	TIMER1	定时器 1 中断	IEN0.3
Int4	23H	UART0	串口 0 模块中断	IEN0.4
Int5	2bH	TIMER2	定时器 2 中断	IEN0.5
Int6	33H	GPIO3	GPIO3 模块中断	IEN1.0
Int7	3bH	GPIO2	GPIO2 模块中断	IEN1.1
Int8	43H	SPI0	SPI0 模块中断	IEN1.2
Int9	4bH	ISO7816S	ISO7816S 模块数据中断	IEN1.3
Int10	53H	GPIO4	GPIO4 模块中断	IEN1.4
Int11	5bH	SM4/DES/AES	SM4/DES/AES 加解密模块中断	IEN1.5
Int12	63H	UART1	串口 1 模块中断	IEN2.0

Int13	6bH	ISO7816S_RST	ISO7816S 模块复位中断	IEN2.1
Int14	73H	SPI1	SPI1 模块中断	IEN2.2
Int15	7bH	SPI2	SPI2 模块中断	IEN2.3
Int16	83H	SPI3	SPI3 模块中断	IEN2.4
Int17	8bH	I2C0_M	I2C0 主机中断	IEN2.5
Int18	93H	RTC	RTC 模块中断	IEN3.0
Int19	9bH	WDT	看门狗模块中断	IEN3.1
Int20	a3H	I2C0_S	I2C0 从机中断	IEN3.2
Int21	abH	ADC	ADC 模块中断	IEN3.3
Int22	b3H	I2C1_M	I2C1 主机中断	IEN3.4
Int23	bbH	I2C1_S	I2C1 从机中断	IEN3.5
Int24	c3H	TIMER3	定时器 3 中断	IEN4.0
Int25	cbH	TIMER4	定时器 4 中断	IEN4.1
Int26	d3H	UART2	串口 2 模块中断	IEN4.2
Int27	dbH	UART3	串口 3 模块中断	IEN4.3
Int28	e3H	COMP1	比较器 1 模块中断	IEN4.4
Int29	ebH	COMP2	比较器 2 模块中断	IEN4.5

9.4. 中断优先级

表格 9-2 中断优先级组

中断组	组间自然优先级	最高优先级	较高优先级	中间优先级	次低优先级	最低优先级
Group0	最高优先级 ↑ 最低优先级	Int0	Int6	Int12	Int18	Int24
Group1		Int1	Int7	Int13	Int19	Int25
Group2		Int2	Int8	Int14	Int20	Int26
Group3		Int3	Int9	Int15	Int21	Int27
Group4		Int4	Int10	Int16	Int22	Int28
Group5		Int5	Int11	Int17	Int23	Int29

表格 9-3 可配置组间中断优先级

可配置优先级组	组间优先级	IP1.x bit	IP0.x bit
Level3	最高优先级 ↑ 最低优先级	1	1
Level2		1	0
Level1		0	1
Level0		0	0

注：1.上表中 x 的值为 0,1,2,3,4,5，分别代表 Group0, Group1, Group2, Group3, Group4, Group5；
例如，将 Group2 的中断优先级设为最高，则 IP1=0x04,IP0=0x04；

2. 当请求多个中断时，所有优先级类型都考虑在内。最重要的是由“IP0”和“IP1”寄存器决定的优先级，然后组间自然优先，最后在每个组组内的优先级；

3. 优先级高的中断可以打断优先级低的中断，反之则不行，同一时间最多有 4 个中断服务。

9.5. 中断服务程序单元寄存器

ISR 特殊功能寄存器

偏置	名称	描述
0xA8	IEN0	中断使能寄存器 0
0xB8	IEN1	中断使能寄存器 1
0x9A	IEN2	中断使能寄存器 2
0xC9	IEN3	中断使能寄存器 3
0xD1	IEN4	中断使能寄存器 4
0xA9	IP0	中断优先级寄存器 0
0xB9	IP1	中断优先级寄存器 1

9.5.1. 中断使能寄存器 IEN0(偏移: A8H)

比特	名称	属性	复位值	描述
7	EAL	RW	0x0	总中断使能位 0: 总中断不使能 1: 开总中断（使用模块的中断都必须将这一位置 1）
6	WDTREL_EN	RW	0x0	看门狗计数值装载使能位 0: 禁止 1: 使能
5	ET2	RW	0x0	定时器 2 中断使能位 0: 禁止 1: 使能
4	ES0	RW	0x0	串行端口 0 中断使能位 0: 禁止 1: 使能
3	ET1	RW	0x0	定时器 1 溢出中断使能位 0: 禁止 1: 使能
2	EX1	RW	0x0	GPIO1 中断使能位 0: 禁止 1: 使能
1	ET0	RW	0x0	定时器 0 溢出使能位 0: 禁止 1: 使能
0	EX0	RW	0x0	GPIO0 中断使能位

				0: 禁止 1: 使能
--	--	--	--	----------------

9.5.2. 中断使能寄存器 IEN1(偏移: B8H)

比特	名称	属性	复位值	描述
7	EXEN2	RW	0x0	定时器 2 外部重载中断使能位 (捕获模式 2 中断) 0: 禁止 1: 使能
6	SWDT	RW	0x0	看门狗定时器开始/刷新标志位
5	EX6	RW	0x0	SM4/DES/TDES/AES 中断使能位 0: 禁止 1: 使能
4	EX5	RW	0x0	GPIO4 中断使能位 0: 禁止 1: 使能
3	EX4	RW	0x0	ISO7816MS 中断使能位 0: 禁止 1: 使能
2	EX3	RW	0x0	SPI0 中断使能位 0: 禁止 1: 使能
1	EX2	RW	0x0	GPIO2 中断使能位 0: 禁止 1: 使能
0	EX7	RW	0x0	GPIO3 中断使能位 0: 禁止 1: 使能

9.5.3. 中断使能寄存器 IEN2(偏移: 9AH)

比特	名称	属性	复位值	描述
7: 6	RSV	-	-	保留
5	I2C0_M_EN	RW	0x0	I2C0 主机中断使能位 0: 禁止 1: 使能
4	SPI3_EN	RW	0x0	SPI3 中断使能位 0: 禁止 1: 使能
3	SPI2_EN	RW	0x0	SPI2 中断使能位 0: 禁止 1: 使能
2	SPI1_EN	RW	0x0	SPI1 中断使能位

				0: 禁止 1: 使能
1	7816C2_EN	RW	0x0	7816 C2 复位中断使能位 0: 禁止 1: 使能
0	ES1	RW	0x0	UART1 中断使能位 0: 禁止 1: 使能

9.5.4. 中断使能寄存器 IEN3(偏移: C9H)

比特	名称	属性	复位值	描述
7: 6	RSV	-	-	保留
5	I2C1_S_EN	RW	0x0	I2C1 从机中断使能位 0: 禁止 1: 使能
4	I2C1_M_EN	RW	0x0	I2C1 主机中断使能位 0: 禁止 1: 使能
3	ADC_EN	RW	0x0	ADC 中断使能位 0: 禁止 1: 使能
2	I2C0_S_EN	RW	0x0	I2C0 从机中断使能位 0: 禁止 1: 使能
1	WDT_EN	RW	0x0	WDT 中断使能位 0: 禁止 1: 使能
0	RTC_EN	RW	0x0	RTC 中断使能位 0: 禁止 1: 使能

9.5.5. 中断使能寄存器 IEN4(偏移: D1H)

比特	名称	属性	复位值	描述
7: 6	RSV	-	-	保留
5	COMP2_EN	RW	0x0	比较器 2 中断使能位 0: 禁止 1: 使能
4	COMP1_EN	RW	0x0	比较器 1 中断使能位 0: 禁止 1: 使能
3	UART3_EN	RW	0x0	串口 3 中断使能位

				0: 禁止 1: 使能
2	UART2_EN	RW	0x0	串口 2 中断使能位 0: 禁止 1: 使能
1	TIMER4_EN	RW	0x0	定时器 4 中断使能位 0: 禁止 1: 使能
0	TIMER3_EN	RW	0x0	定时器 3 中断使能位 0: 禁止 1: 使能

9.5.6. 中断优先级寄存器 IP0(偏移: A9H)

比特	名称	属性	复位值	描述
7	RSV	-	-	保留
6	WDTS	RW	0x0	看门狗定时器状态标志位 当看门狗定时器复位发生, 由硬件将此位置 1
5	IP0.5	RW	0x0	中断优先级配置 与 IP1 相应的位配合指定中断优先级组各自的优先级。 详见表 10-3 及其注释。
4	IP0.4	RW	0x0	
3	IP0.3	RW	0x0	
2	IP0.2	RW	0x0	
1	IP0.1	RW	0x0	
0	IP0.0	RW	0x0	

9.5.7. 中断优先级寄存器 IP1(偏移: B9H)

比特	名称	属性	复位值	描述
7:6	RSV	-	-	保留
5	IP1.5	RW	0x0	中断优先级配置 与 IP0 相应的位配合指定中断优先级组各自的优先级。 详见表 10-3 及其注释。
4	IP1.4	RW	0x0	
3	IP1.3	RW	0x0	
2	IP1.2	RW	0x0	
1	IP1.1	RW	0x0	
0	IP1.0	RW	0x0	

10.模拟/数字转换器（ADC）

10.1. 概述

该设备内置了一个高速的 12 位逐次比较模拟数字转换器。提供 7 个多功能的输入通道（具体请参考 ADC 引脚描述说明）。A/D 转换的各个通道可以执行单次和有缓存功能的连续转换模式。

10.2. ADC 主要特性

- 12 位的分辨率
- 单次和有缓存功能的连续转换模式
- 转换结束时可以产生中断
- 内置校准单元
- 转换速率高达 1MSPS（16M 时钟下）

10.3. ADC 引脚描述

表 10-1 ADC 引脚

引脚名称	ADC引脚名称	信号类型	注释
VDD	VDDA50	输入，模拟电源	模拟电源供电端，默认 ADC 参考电压高端
VSS	GNDA50	输入，模拟电源地	模拟电源地端，默认 ADC 参考电压低端
P20	ADC_REFN	输入，模拟参考负极	ADC 使用的低端/负极参考电压
P21	ADC_REFP	输入，模拟参考正极	ADC 使用的高端/正极参考电压
P23	ADC_IN7	模拟输入信号 7	采集信号端 7
P24	ADC_IN6	模拟输入信号 6	采集信号端 6
P25	ADC_IN5	模拟输入信号 5	采集信号端 5
P26	ADC_IN4	模拟输入信号 4	采集信号端 4
P27	ADC_IN3	模拟输入信号 3	采集信号端 3
P44	ADC_IN2	模拟输入信号 2	采集信号端 2
P45	ADC_IN1	模拟输入信号 1	采集信号端 1

单端模式 ADC 采集时，每次只使用 1 个通道。
差分模式 ADC 采集时，通道 2 和 3、4 和 5、6 和 7 组成差分对。（具体参考 ADC 寄存器描述）

10.4. ADC 功能描述

10.4.1. ADC 开-关控制

通过寄存器 ADC_CR1 的第 0 比特位来开启 ADC。如果长时间不使用 ADC，可以通过清零这个位来关闭 ADC。

当 ADC 模块上电后，所选通道的 I/O 口输出模块是被禁止的。因此，在使用 ADC

模块时要选择合适的 ADC 转换通道。

10.4.2. 通道选择

ADC 有 7 个外部输入通道，实际外部通道的数量取决于 MCU 封装大小。单端时可通过配置 ADC_CSR 的第 0 至 3 比特位选择模拟转换通道。差分时可通过配置 ADC_CR2 的第 3 至 6 比特位选择差分对。

注：配置 ADC 通道前，需要将对应的通道引脚功能配置为 GPIO 输入模式。

10.4.3. 转换模式

ADC 支持单次模式，带缓存的连续模式。模式的选择可通过 ADC_CR1 寄存器的第 1 比特位选择转换模式。

单次模式：在单次转换模式中，每次通过 ADC_CR1 的第 4 比特位开始单次转换（转换完成后自动清零）。当 ADC_STS 寄存器的第 0 比特位置 1（如果使能 ADC_MSK 寄存器的第 0 比特位，则会产生 1 个中断），转换后的数据存储在 ADC_DRH 和 ADC_DRL 寄存器中。

带缓存的连续模式：在连续转换模式中，每次通过 ADC_CR1 的第 2 比特位开始连续转换（转换完成后自动清零）。每完成 1 次转换数据存在 FIFO 中，转化 8 次，当 ADC_STS 寄存器的第 7 比特位置 1，FIFO 被填满。ADC_STS 寄存器的第 0 比特位置 1（如果使能 ADC_MSK 寄存器的第 0 比特位，则会产生 1 个中断），然后读取 FIFO 数据 8 次（即读 ADC_DRH 和 ADC_DRL 寄存器 8 次）。

注：当发生转换错误及数据溢出时，可通过 ADC_STS 寄存器了解相关信息，及通过 ADC_CR3 寄存器清除该错误标志（具体参考寄存器描述）。

10.4.4. ADC 校准及异常处理

ADC 自带校准功能，可通过 ADC_CR1 寄存器的第 3 比特位开启校准功能。通过 ADC_STS 寄存器的第 3,5 比特位，了解校准是否有错误及是否完成。当发生校准错误时及完成时，通过 ADC_CR3 寄存器的第 0,1 比特位置位清除状态。

10.5. ADC 寄存器

ADC 寄存器基地址：0x2000

偏置	名称	描述
0x03	ADC_CSR	ADC 配置寄存器
0x04	ADC_CR1	ADC 配置寄存器 1
0x05	ADC_CR2	ADC 配置寄存器 2
0x06	ADC_CR3	ADC 配置寄存器 3
0x07	ADC_DRH	ADC 数据高位寄存器（低 4bit 有效）
0x08	ADC_DRL	ADC 数据低位寄存器
0x09	ADC_STS	ADC 状态寄存器

0x0A	ADC_MSK	ADC 中断屏蔽寄存器
------	---------	-------------

10.5.1. 配置寄存器 ADC_CSR(偏移: 03H)

比特	名称	属性	复位值	描述
7	RSV	-	-	保留
6:4	ADCSREF[2:0]	RW	0x0	ADC 参考电压选择 000: VREF+ = VDDA50 VREF- = GNDA50 001: 保留 010: VREF+ = VREFPE VREF- = GNDA50 011: VREF+ = VREFPE VREF- = GNDA50 100: VREF+ = VDDA50 VREF- = VREFNE 101: 保留 110: VREF+ = VREFPE VREF- = VREFNE 111: VREF+ = VREFPE VREF- = VREFNE
3:0	AMUX[3:0]	RW	0x0	模拟转换通道选择 0000: 保留 0001: ADC_IN1 0010: ADC_IN2 0011: ADC_IN3 0100: ADC_IN4 0101: ADC_IN5 0110: ADC_IN6 0111: ADC_IN7 100x: 保留 101x: 1/2 VDDA50 110x: VREFPE 111x: VREFNE

10.5.2. 配置寄存器 ADC_CR1(偏移: 04H)

比特	名称	属性	复位值	描述
7:5	RSV	-	-	保留
4	CONV_EN	RW	0x0	单次转换使能位 0: 禁止 1: 使能 硬件清 0
3	CAL_EN	RW	0x0	校准使能位 0: 禁止 1: 使能 硬件清 0

2	CONT_CONV_EN	RW	0x0	连续转换使能位 0: 禁止 1: 使能 硬件清 0
1	CONT	RW	0x0	转换模式选择 0: 单次模式 1: 连续模式
0	ADC_EN	RW	0x0	ADC 使能位 0: 禁止 1: 使能

10.5.3. 配置寄存器 ADC_CR2(偏移: 05H)

比特	名称	属性	复位值	描述
7	ADCCR	RW	0x0	ADC 转换速率 0: 最大时钟频率为 16MHz 1: 最大时钟频率为 4MHz 时钟频率 $\leq$ 4MHz 时, 将该位置 1 可降低功耗
6	AINS[3]	RW	0x0	AIN[6],AIN[7]输入对配置 0: AIN[6],AIN[7]是独立的单端输入 1: AIN[6],AIN[7]是+和-的差分对
5	AINS[2]	RW	0x0	AIN[4],AIN[5] 输入对配置 0: AIN[4],AIN[5]是独立的单端输入 1: AIN[4],AIN[5]是+和-的差分对
4	AINS[1]	RW	0x0	AIN[2],AIN[3] 输入对配置 0: AIN[2],AIN[3]是独立的单端输入 1: AIN[2],AIN[3]是+和-的差分对
3	AINS[0]	RW	0x0	AIN[0],AIN[1] 输入对配置 0: AIN[1]是独立的单端输入 1: 保留
2:1	ADCSHT[1:0]	RW	0x0	ADC 采样保持时间选择 00: 3.5 个 CLKADC 时钟周期 01: 4.5 个 CLKADC 时钟周期 10: 6.5 个 CLKADC 时钟周期 11: 10.5 个 CLKADC 时钟周期
0	CONV_CAL	RW	0x0	0: 不带校准值转换 1: 带校准值转换

10.5.4. 配置寄存器 ADC_CR3(偏移: 06H)

比特	名称	属性	复位值	描述
7:4	RSV	-	-	保留

3	CLR_FIFO	WO	0x0	FIFO 数据清除位 写 1 清空 FIFO 数据
2	CLR_CONV_ERR	WO	0x0	转换错误标志清除位 写 1 清 0
1	CLR_CAL_ERR	WO	0x0	校准错误标志清除位 写 1 清 0
0	CLR_EOCAL	WO	0x0	校准完成标志清除位 写 1 清 0

10.5.5. 数据高位寄存器 ADC_DRH(偏移: 07H)

比特	名称	属性	复位值	描述
7:4	RSV	-	-	保留
3:0	DH[3:0]	RO	0x0	转换结果的高 4 位

10.5.6. 数据低位寄存器 ADC_DRL(偏移: 08H)

比特	名称	属性	复位值	描述
7:0	DL[7:0]	RO	0x00	转换结果的低 8 位 先读高 4 位, 再读低 8 位

10.5.7. 状态寄存器 ADC_STS(偏移: 09H)

比特	名称	属性	复位值	描述
7	FIFO_FULL	RO	0x0	FIFO 满标志位 0: 非满 1: 满
6	BUSY	RO	0x0	ADC 状态位 0: ADC 空闲 1: ADC 忙
5	EOCAL	RO	0x0	校准完成标志位 0: 未完成 1: 完成
4	CONV_ERR	RO	0x0	转换错误标志位 0: 无错误 1: 有错误
3	CAL_ERR	RO	0x0	校准错误标志位 0: 无错误 1: 有错误
2	EMPTY	RO	0x0	FIFO 空标志位 0: 非空 1: 空
1	OVR	RW	0x0	数据溢出标志位

				0: 无溢出 1: 有溢出
0	EOC	RW	0x0	转换完成标志位 0: 未完成 1: 完成

10.5.8. 中断屏蔽寄存器 ADC_MSK(偏移: 0AH)

比特	名称	属性	复位值	描述
7:4	RSV	-	-	保留
3	FIFO_HALF_FULL	RO	0x0	FIFO 半满标志位 0: 未半满 1: 半满
2	ADC_MSK2	RW	0x1	FIFO 半满中断屏蔽位 0: 不屏蔽 1: 屏蔽
1	ADC_MSK1	RW	0x1	FIFO 空中断屏蔽位 0: 不屏蔽 1: 屏蔽
0	ADC_MSK0	RW	0x1	转换完成中断屏蔽位 0: 不屏蔽 1: 屏蔽

10.6. ADC 操作说明

10.6.1. 单次模式

- 1) 配置 ADC_CSR, 选择参考电压以及转换通道;
- 2) 配置 ADC_CR1 的 CONT 位为 0, 选择单次模式;
- 3) 配置 ADC_CR2, 选择带校准值转换、设置采样保持时间、设置选定的转换通道模式、选择转换速率;
- 4) 配置 ADC_MSK, 选择是否使能中断;
- 5) 配置 ADC_CR3 的 CLR_FIFO 位, 清空 FIFO;
- 6) 置位 ADC_CR1 的 ADON 位, 启动 ADC;
- 7) 置位 ADC_CR1 的 CONV_EN 位, 启动单次采样和转换;
- 8) 查询方式: 软件查询 ADC_STS 的 EOC 位, 判断转换完成, 清除状态, 读取转换结果;
中断方式: 转换完成后产生中断, 中断服务函数内判断转换完成, 清除 ADC_STS 的 EOC 位, 读取转换结果;
- 9) 异常处理: 查询状态, 判断异常后, 根据异常结果做如下处理:
清除校准异常; 清除转换完成异常; 清除溢出标志; 清除 fifo。

10.6.2. 连续模式

- 1) 配置 ADC_CSR, 选择参考电压, 选择转换通道;
- 2) 配置 ADC_CR1 的 CONT 位为 1, 选择连续模式;
- 3) 配置 ADC_CR2, 选择带校准值转换、设置采样保持时间、设置选定的转换通道模式、选择转换速率;
- 4) 写 ADC_MSK, 选择是否使能转换结束中断;
- 5) 配置 ADC_CR3 的 CLR_FIFO 位, 清空 FIFO;
- 6) 置位 ADC_CR1 的 ADON 位, 启动 ADC;
- 7) 置位 ADC_CR1 的 CONT_CONV_EN 位, 启动连续采样和转换;
- 8) 每完成一次转换, 将转换数据存在 FIFO 中, 连续转换 8 次 FIFO 被填满, ADC_STS 的 EOC 为硬件置位
- 9) 查询模式: 等待转换完成, 查询状态, 判断成功, 读取 fifo 数据 8 次; 关闭连续转换使能, 清除状态;
中断模式: 进入中断后, 查询状态, 读取 fifo 数据 8 次, 清除状态;
- 10) 异常处理, 查询状态判断异常后, 根据异常结果做如下处理:
清除校准异常; 清除转换完成异常; 清除溢出标志; 清除 fifo。

11. 比较器（COMP）

11.1. 概述

该设备内置了 2 路比较器 COMP1 和 COMP2，它们共享相同的偏置电流。

11.2. COMP 主要特性

- COMP1:反相输入端是固定阈值，为内部参考电压；正相输入端为外部 IO 输入
- COMP2:反相输入端阈值可选，可为内部参考电压或外部 IO 输入参考电压；正相输入端为外部 IO 输入
- 两路比较器都可产生中断
- 两路比较器输出极性可控
- COMP2 的反相输入端阈值可选，内部参考电压有 4 个档位
- COMP2 的输出可重定向为 TIMER4 的刹车输入或者 TIMER3 和 TIMER2 的捕获 2 输入
- COMP2 的工作模式可选：快速模式下，比较器翻转速度快，功耗高
低速模式下，比较器翻转速度慢，功耗低

11.3. COMP 引脚描述

表 11-1 COMP 引脚

引脚名称	ADC引脚名称	信号类型	注释
P30	COMP_VIO	输入，模拟反相输入端	比较器 2 外部反相输入通道
P31	COMP_IN1	输入，模拟正相输入端	比较器 1 外部正相输入通道
P32	COMP_IN2	输入，模拟正相输入端	比较器 2 外部正相输入通道

11.4. 比较器寄存器

COMP 寄存器基地址：0x2000

偏置	名称	描述
0x1E	COMP_CTRL0	控制寄存器 0
0x1F	COMP_CTRL1	控制寄存器 1
0xFC	IOCFG_CTRL6	状态/控制寄存器

11.4.1. 控制寄存器 COMP_CTRL0(偏移：1EH)

比特	名称	属性	复位值	描述
7	MC	RW	0x0	比较器 2 的工作模式控制位 0：低速模式 1：快速模式

6:4	CM[2:0]	RW	0x0	比较器 2 的反相参考电压选择位： 000: 1.0V 001: 0.9V 010: 0.6V 011: 0.3V 100: IO 输入电压
3	ENBUFFER	RW	0x0	缓冲器模块使能位 0: 禁止 1: 使能
2	ENBIAS	RW	0x0	偏置电流模块使能位 0: 禁止 1: 使能
1	ENCOMP2	RW	0x0	比较器 2 的使能位 0: 禁止 1: 使能
0	ENCOMP1	RW	0x0	比较器 1 的使能位 0: 禁止 1: 使能

11.4.2. 控制寄存器 COMP_CTRL1(偏移: 1FH)

比特	名称	属性	复位值	描述
7	COMP_POL1	RW	0x0	比较器 1 的输出极性选择 0: 输出取反 1: 输出不变
6:5	OUTSEL[1:0]	RW	0x0	比较器 2 输出选择 00: 保留 01: 比较器 2 的输出接 TIMER4 的刹车输入 10: 比较器 2 的输出接 TIMER3 的捕获 2 输入 11: 比较器 2 的输出接 TIMER2 的捕获 2 输入
4	COMP_POL2	RW	0x0	比较器 2 的输出极性选择 0: 输出取反 1: 输出不变
3:2	EDGE_SEL2	RW	0x0	比较器 2 配置 00: 比较器 2 禁止 01: 比较器 2 输出端检测到上升沿事件 10: 比较器 2 输出端检测到下降沿事件 11: 比较器 2 输出端检测到双沿事件
1:0	EDGE_SEL1	RW	0x0	比较器 1 配置 00: 比较器 1 禁止 01: 比较器 1 输出端检测到上升沿事件 10: 比较器 1 输出端检测到下降沿事件 11: 比较器 1 输出端检测到双沿事件

11.4.3. 状态/控制寄存器 IOCFG_CTRL6(偏移: FCH)

比特	名称	属性	复位值	描述
7	COMP_MSK[1]	RW	0x1	比较器 2 中断屏蔽使能位 0: 不屏蔽 1: 屏蔽
6	COMP_MSK[0]	RW	0x1	比较器 1 中断屏蔽使能位 0: 不屏蔽 1: 屏蔽
5	COMP_STS[1]	RW	0x0	比较器 2 事件标志 输出端发生 EDGE_SEL2 选择的事件, 该位置 1, 通过软件写 0 清。可根据屏蔽位选择是否产生中断。 0: 未检测到事件 1: 检测到事件
4	COMP_STS[0]	RW	0x0	比较器 1 事件标志 输出端发生 EDGE_SEL1 选择的事件, 该位置 1, 通过软件写 0 清。可根据屏蔽位选择是否产生中断。 0: 未检测到事件 1: 检测到事件
3:0	-	-	-	低 4 位不是比较器的功能位, 不在此描述

11.5. 比较器操作说明

11.5.1. 比较器 1 (COMP1)

使用比较 1, 应用程序必须遵循以下步骤:

1. 配置寄存器 COMP_CTRL0 的 ENBISA 位, 使能偏置电流
2. 配置寄存器 COMP_CTRL1 的 COMP_POL1 位设置输出极性; EDGE_SEL1 位选择沿事件
3. 配置寄存器 IOCFG_CTRL6 的 COMP_MSK[0]位, 设置是否产生中断
4. 配置 COMP_CTRL0 的 ENCOMP1 位, 使能比较器 1

11.5.2. 比较器 2 (COMP2)

使用比较 2, 应用程序必须遵循以下步骤:

1. 配置寄存器 COMP_CTRL0 的 MC、CM、ENBISA 和 ENBUFFER 位, 设置工作模式, 选择反相参考电压, 缓冲器和偏置电流使能
2. 配置 COMP_CTRL1 的 OUTSEL、COMP_POL2、EDGE_SEL2 位, 设置输出选择、输出极性、选择沿事件
3. 配置寄存器 IOCFG_CTRL6 的 COMP_MSK[1]位, 设置是否产生中断
4. 配置 COMP_CTRL0 的 ENCOMP2 位, 使能比较器 2。

12.基本定时器（TIMER0/1）

12.1. 概述

该设备内置了 2 个向上计数的基本定时器/计数器：TIMER0 和 TIMER1。

每个定时器都是可以完全独立的，可以用来作为时基发生器，具有溢出中断功能，其还可以配置为 16 位/13 位/8 位定时/计数器。

12.2. 基本定时器主要特性

- 最大 16 位递增定时器
- TIMER0 支持 4 种模式：13 位/16 位/8 位重载/双 8 位定时/计数器
- TIMER1 支持 3 种模式：13 位/16 位/8 位重载定时/计数器
- 中断产生--在计数器更新时：计数器溢出

12.3. 基本定时器工作模式

基本定时器都是可以配置为 16 位/13 位/8 位定时/计数器。其可以通过 TMOD 寄存器第 0,1 比特位设置定时器 0 的工作模式，通过 TMOD 寄存器第 4,5 比特位设置定时器 1 的工作模式。

表 12-1 基本定时器工作模式（表中 x=0 或 1）

模式	M1	M0	功能
模式 0	0	0	13 位定时/计数器，TLx 低 5 位，和 THx 构成 13 位定时/计数器
模式 1	0	1	16 位定时/计数器，TLx 和 THx 构成 16 位定时/计数器
模式 2	1	0	8 位重载，溢出后，将 THx 的值重载到 TLx 重新计数
模式 3	1	1	TIMER0 拆成两个 8 位计数器(会借用 TIMER1 的控制位、标志位和中断线)

定时是指当计数脉冲接入周期变化的脉冲信号，计数器从某个初值 x（可通过 TL0 和 TH0 寄存器进行配置）开始加 1 计数，到计满回零的瞬间产生溢出中断，表示定时时间到。定时时间计算如下：

$$t = (M - x) * T$$

式中，t 为定时时间；

M 为计数器的模；

x 为计数初值；

T 为计数脉冲的周期。

模 M 是指计数器从 0 开始计数到溢出时的最大计数值，即 $M = 2^n$ 。

例如，8 位计数器 $M = 2^8 = 256$

13 位计数器 $M = 2^{13} = 8192$

16 位计数器 $M = 2^{16} = 65536$

计数脉冲周期: $T = \frac{1}{f_{osc}} * 12$ (f_{osc} 为系统时钟频率)。

12.4. 基本定时器寄存器

TIMER0/1 特殊功能寄存器

偏置	名称	描述
0x88	TCON	定时器控制寄存器
0x89	TMOD	定时器模式寄存器
0x8A	TL0	TIMER0 计数寄存器低字节
0x8B	TL1	TIMER1 计数寄存器低字节
0x8C	TH0	TIMER0 计数寄存器高字节
0x8D	TH1	TIMER1 计数寄存器高字节

12.4.1. 定时器控制寄存器 TCON(偏移: 88H)

比特	名称	属性	复位值	描述
7	TF1	RW	0x0	定时器 1 溢出标志位 计数溢出时硬件置 1; 中断处理后自动清 0 或软件清 0
6	TR1	RW	0x0	定时器 1 运行控制位 0: 禁止 1: 运行
5	TF0	RW	0x0	定时器 0 溢出控制位 计数溢出时硬件置 1; 中断处理后自动清 0 或软件清 0
4	TR0	RW	0x0	定时器 0 运行控制位 0: 禁止 1: 运行
3	IE1	RW	0x0	外部中断 1 标志位 硬件置 1; 中断处理后自动清 0 或软件清 0
2	IT1	RW	0x1	外部中断 1 触发方式选择位(GPIO1) 0: 电平触发 1: 沿触发
1	IE0	RW	0x0	外部中断 0 标志位 硬件置 1, 中断处理后自动清 0 或软件清 0
0	IT0	RW	0x1	外部中断 0 触发方式选择位(GPIO0) 0: 电平触发 1: 沿触发

12.4.2. 定时器模式寄存器 TMOD(偏移: 89H)

比特	名称	属性	复位值	描述
7	TIMER1_GATE	RW	0x0	定时器 1 门控位 0: 关闭 1: 打开
6	TIMER1_C/T	RW	0x0	定时器 1 定时/计数选择位 0: 定时 1: 计数
5	TIMER1_M1	RW	0x0	模式选择位
4	TIMER1_M0	RW	0x0	参考表 12-1 中描述
3	TIMER0_GATE	RW	0x0	定时器 0 门控位 0: 关闭 1: 打开
2	TIMER0_C/T	RW	0x0	定时器 0 定时/计数选择位 0: 定时 1: 计数
1	TIMER0_M1	RW	0x0	参考表 12-1 中描述
0	TIMER0_M0	RW	0x0	

12.4.3. TIMER0 计数寄存器低字节 TL0(偏移: 8AH)

比特	名称	属性	复位值	描述
7:0	TL0	RW	0x00	TIMER0 计数寄存器低字节

12.4.4. TIMER1 计数寄存器低字节 TL1(偏移: 8BH)

比特	名称	属性	复位值	描述
7:0	TL1	RW	0x00	TIMER1 计数寄存器低字节

12.4.5. TIMER0 计数寄存器高字节 TH0(偏移: 8CH)

比特	名称	属性	复位值	描述
7:0	TH0	RW	0x00	TIMER0 计数寄存器高字节

12.4.6. TIMER1 计数寄存器高字节 TH1(偏移: 8DH)

比特	名称	属性	复位值	描述
7:0	TH1	RW	0x00	TIMER1 计数寄存器高字节

12.5. 基本定时器操作说明

1、TIMER0 的操作:

- 1) 配置 TMOD, 设置定时器工作模式;
- 2) 设置 TH0, TL0 初值;
- 3) 设置 TCON 的 TR0 位, 启动 TIMER0
- 4) 溢出后, TCON 的 TF0 为被置位, 需要软件清除 TF0 标志位

可采用轮询或中断方式处理定时器中断。

2、TIMER1 的操作：

参考 TIMER0，但不支持模式 3。

13. 通用定时器（TIMER2）

13.1. 概述

该设备内置了 1 个带有可编程分频器，向上计数的 16 位基本定时器：TIMER2。适用于多个场合，包括基本的定时，测量输入信号脉冲长度，产生输出波形（输出比较，PWM）。

13.2. 基本定时器主要特性

- 16 位向上计数和自动装载计数器
- 工作模式有 3 种
 - 基本定时模式
 - 事件计数模式
 - 门控定时模式
- 可更改的计数分频（12 分频/24 分频）
- 3 个独立通道
 - 输入捕获
 - 输出比较
 - PWM 生成（无互补输出）
- 如下事件发生时可产生中断
 - 更新中断：计数向上溢出
 - 外部重载
 - 捕获输入（捕获模式 2）
 - 输出比较

13.3. 分频器

该通用定时器，计数时钟分频可以通过寄存器 T2CON 的第 7 比特位选择，即 T2CON[7]=0 为 12 分频，T2CON[7]=1 为 24 分频。

13.4. 定时器模式

TIMER2 基本定时器模式，可通过配置寄存器 T2CON 的第 0,1 比特位来选择该模式，即 T2CON[1]=0，T2CON[0]=1。

13.5. 事件计数模式

TIMER2 事件计数模式，可通过配置寄存器 T2CON 的第 0,1 比特位来选择该模式，即 T2CON[1]=1，T2CON[0]=0。该模式下当外部信号由高变低，则计数增加。在每个

时钟的上升沿都会对外部输入信号进行采样。

注：外部输入引脚端口复用

13.6. 门控定时模式

TIMER2 门控定时模式，可通过配置寄存器 T2CON 的第 0,1 比特位来选择该模式，即 T2CON[1]=1，T2CON[0]=1。该模式下当外部信号为高时，则计数增加。外部信号为低时，则计数增加停止。

注：外部输入引脚端口复用

13.7. 定时器重载功能

TIMER2 重载功能有 2 种模式，可通过配置寄存器 T2CON 的第 3,4 比特位来选择重载模式，即 T2CON[4]=1，T2CON[3]=0 选择模式 0；T2CON[4]=1，T2CON[3]=1 选择模式 1。

重载模式 0：计数溢出后产生重载信号（自动重载）

重载模式 1：输入引脚信号有下降沿，则产生重载信号。

重载功能下，重载值为寄存器 CRCH,CRCL 的值。

注：外部输入引脚端口复用

13.8. 定时器比较输出功能

13.8.1. 比较模式

TIMER2 比较输出功能有 2 种模式，可通过配置寄存器 T2CON 的第 2 比特位来选择比较模式，即 T2CON[2]=0 选择模式 0；T2CON[2]=1 选择模式 1。

在 2 种比较模式中有 3 个比较输出通道即 CRC,CC1,CC2，3 个通道下对应 3 个端口和 3 个比较器，可通过寄存器 CCEN 的 0,1 比特位选择 CRC；CCEN 的 2,3 比特位选择 CC1；CCEN 的 4,5 比特位选择 CC2（具体配置参考寄存器描述）。

设定比较值时：CRC 通道对应寄存器 CRCH,CRCL；CC1 通道对应寄存器 CCH1,CCL1；CC2 通道对应寄存器 CCH2,CCL2

表 13-1 TIMER2 比较通道对应输出引脚

引脚名称	输出引脚名称	信号类型	注释(在使用时注意端口复用)
P01	COMP00	输出	与通道 CRC 相应的比较输出引脚
P02	COMP01	输出	与通道 CC1 相应的比较输出引脚
P03	COMP02	输出	与通道 CC2 相应的比较输出引脚

比较模式 0：当 TIMER2 的计数值等于比较设定的值时，相应比较器通道的比较输出信号变化。具体可由寄存器 CCL3[i](i=0/1/2)控制，即

当 CCL3[i](i=0/1/2)=0 时，相应通道比较输出由低变高，

当 CCL3[i](i=0/1/2)=1 时，相应通道比较输出由高变低；

TIMER2 计数值溢出时，相应比较器通道的输出信号变为与比较器的值相等时输出信号相反的极性。

注：i=0 对应 CRC 通道；i=1 对应 CC1 通道；i=2 对应 CC2 通道

比较模式 1：当 TIMER2 的计数值等于比较器的值时，相应比较器通道的比较输出信号变化（具体由寄存器 P1 的第 0,1,2 位控制，即当计数值等于设定的比较值时，输出信号被激活，输出引脚输出 P1[i](i=0/1/2)的数值(需要使能 P1 口的时钟)；TIMER2 计数值溢出时，相应比较器通道的输出信号不会变化。

13.8.2. PWM 模式

脉冲宽度调制基于定时器的比较输出功能，可以通过设定比较值来改变占空比。

在 T2CON[2]=0 选择比较模式 0，来选择 PWM 模式，能够独立地设定每个输出通道产生 PWM,其通过 CCEN[i][j]=10(i=1,3,5 j=0,2,4)来选择 1 路 PWM 也可选择多路。

PWM 的极性可以通过软件在寄存器中 CCL3[i](i=0/1/2)设置。其计数值和比较值始终进行比较用以确定是否触发比较输出信号的产生。

通过设置通道的比较值来产生不同占空比的 PWM。（CRC 通道对应寄存器 CRCH,CRCL；CC1 通道对应寄存器 CCH1,CCL1；CC2 通道对应寄存器 CCH2,CCL2）。

13.9. 定时器捕获功能

TIMER2 捕获功能有 3 种模式，该功能下实际的时间计数内容保存到相应的寄存器中。

捕获模式 0：检测到外部输入引脚的沿变化，则发生捕获。该模式下有 3 个通道对应 3 个外部输入引脚和 3 个数据缓存寄存器。

表 13-2 TIMER2 捕获模式 0 通道对应输出引脚

引脚名称	输出引脚名称	信号类型	注释(在使用时注意端口复用)
P04	CAP00	输入	与通道 CRC 相应的通道输入引脚
P05	CAP01	输入	与通道 CC1 相应的通道输入引脚
P06	CAP02	输入	与通道 CC2 相应的通道输入引脚

可通过设置寄存器 CCEN 选择不同的通道即 CCEN[i][j]=01(i=1,3,5 j=0,2,4)选择相应的捕获通道。

发生捕获时可读取相应的数据缓存寄存器的值，获取当前的实际计数值。（CRC 通道对应寄存器 CRCH,CRCL；CC1 通道对应寄存器 CCH1,CCL1；CC2 通道对应寄存器 CCH2,CCL2）。

捕获的极性：CRC 通道的极性可通过 T2CON 的第 6 比特位选择上升沿或下降沿。CC1 和 CC2 只支持上升沿。

捕获模式 1：检测到专用的捕获寄存器的低字节写入数据，则发生捕获。定时器的值会被锁存到到相应的寄存器中。

该模式下有 3 个通道对应 3 个专用的捕获写入寄存器和 3 个数据缓存寄存器。

通过设置寄存器 CCEN 选择不同的通道即 CCEN[i][j]=11(i=1,3,5 j=0,2,4)选择相应的捕获通道。

当专用捕获寄存器的低字节写入数据时，则发生捕获。（CRC 通道对应寄存器 CRCL；CC1 通道对应寄存器 CCL1；CC2 通道对应寄存器 CCL2）

发生捕获时读取相应通道的数据缓存寄存器的计数值。（CRC 通道对应寄存器 CRCH,CRCL；CC1 通道对应寄存器 CCH1,CCL1；CC2 通道对应寄存器 CCH2,CCL2）。

捕获模式 2：检测到外部信号发生相应的变化时则可以产生捕获。

表 13-3 定时器 2 捕获模式 2 通道对应输出引脚

引脚名称	输出引脚名称	信号类型	注释(在使用时注意端口复用)
P00	T2EX0	输入	通道相应输入引脚

此模式下捕获的极性只支持外部输入信号的下降沿。此模式下可产生中断。

13.10. 定时器 2 中断

中断的产生需使能相应的中断源。

可产生中断如下事件发生时可产生中断：

- 更新中断：计数向上溢出
- 外部重载
- 捕获输入（捕获模式 2）
- 输出比较中断

13.11. 定时器 2 寄存器

TIMER2 特殊功能寄存器

偏置	名称	描述
0xC8	T2CON	TIMER2 控制寄存器
0xC1	CCEN	TIMER2 捕获/比较模式选择寄存器
0xC6	CCL3	TIMER2 比较输出极性选择寄存器
0xC2	CCL1	TIMER2 比较/捕获寄存器低字节
0xC3	CCH1	TIMER2 比较/捕获寄存器高字节
0xC4	CCL2	TIMER2 比较/捕获寄存器低字节
0xC5	CCH2	TIMER2 比较/捕获寄存器高字节
0xCA	CRCL	TIMER2 比较/捕获寄存器低字节
0xCB	CRCH	TIMER2 比较/捕获寄存器高字节

0xCC	TL2	TIMER2 计数寄存器低字节
0xCD	TH2	TIMER2 计数寄存器高字节
0xC0	IRCON	TIMER2 中断标志状态寄存器

13.11.1. TIMER2 控制寄存器 T2CON(偏移: C8H)

比特	名称	属性	复位值	描述
7	T2PS	RW	0x0	预分频选择 0: TIMER2 以晶振频率的 1/12 计数 1: TIMER2 以晶振频率的 1/24 计数
6	I3FR	RW	0x0	外部中断 3 的沿选择(同样用于捕获模式 0 信号) 0: 下降沿 1: 上升沿
5	I2FR	RW	0x0	外部中断 2 的沿选择 0: 下降沿 1: 上升沿
4	T2R1	RW	0x0	重载模式选择 0x: 重载禁止
3	T2R0	RW	0x0	10: 模式 0 11: 模式 1
2	T2CM	RW	0x0	比较模式选择 0: 模式 0 1: 模式 1
1	T2I1	RW	0x0	输入选择 00: TIMER2 停止 01: 定时器模式
0	T2I0	RW	0x0	10: 事件计数模式 11: 门控定时模式

13.11.2. TIMER2 捕获/比较模式选择寄存器 CCEN(偏移: C1H)

比特	名称	属性	复位值	描述
7:6	-	-	-	保留
5:4	COCAH2 COCAL2	RW	0x0	比较/捕获模式下, CC2 通道
				COCAH2 COCAL2 描述
				0 0 比较/捕获禁止
				0 1 通道 CC2 的上升沿捕获
				1 0 通道比较输出使能
3:2	COCAH1 COCAL1	RW	0x0	1 1 写 CCL2 寄存器捕获
				比较/捕获模式下, CC1 通道
				COCAH1 COCAL1 描述
				0 0 比较/捕获禁止

				0	1	通道 CC1 的上升沿捕获
				1	0	通道比较输出使能
				1	1	写 CCL1 寄存器捕获
1:0	COCAH0 COCAL0	RW	0x0	比较/捕获模式下，CRC 通道		
				COCAH0	COCAL0	描述
				0	0	比较/捕获禁止
				0	1	通道 CRC 的捕获
				1	0	通道比较输出使能
				1	1	写 CRCL 寄存器捕获

13.11.3. TIMER2 比较输出极性选择寄存器 CCL3(偏移: C6H)

比特	名称	属性	复位值	描述
7	-	-	-	保留
2	CCL3[2]	RW	0x0	通道 CC2 的输出极性选择 0: 计数值等于比较值时, 信号由低变高 1: 计数值等于比较值时, 信号由高变低
1	CCL3[1]	RW	0x0	通道 CC1 的输出极性选择 0: 计数值等于比较值时, 信号由低变高 1: 计数值等于比较值时, 信号由高变低
0	CCL3[0]	RW	0x0	通道 CRC 的输出极性选择 0: 计数值等于比较值时, 信号由低变高 1: 计数值等于比较值时, 信号由高变低

13.11.4. TIMER2 比较/捕获寄存器 CCL1(偏移: C2H)

比特	名称	属性	复位值	描述
7:0	CCL1	RW	0x00	TIMER2 的通道 CC1 比较/捕获寄存器低字节

13.11.5. TIMER2 比较/捕获寄存器 CCH1(偏移: C3H)

比特	名称	属性	复位值	描述
7:0	CCH1	RW	0x00	TIMER2 的通道 CC1 比较/捕获寄存器高字节

13.11.6. TIMER2 比较/捕获寄存器 CCL2(偏移: C4H)

比特	名称	属性	复位值	描述
7:0	CCL2	RW	0x00	TIMER2 的通道 CC2 比较/捕获寄存器低字节

13.11.7. TIMER2 比较/捕获寄存器 CCH2(偏移: C5H)

比特	名称	属性	复位值	描述
7:0	CCH2	RW	0x00	TIMER2 的通道 CC2 比较/捕获寄存器高字节

13.11.8. TIMER2 比较/捕获寄存器 CRCL(偏移: CAH)

比特	名称	属性	复位值	描述
7:0	CRCL	RW	0x00	TIMER2 的通道 CRC 比较/捕获寄存器高字节

13.11.9. TIMER2 比较/捕获寄存器 CRCH(偏移: CBH)

比特	名称	属性	复位值	描述
7:0	CRCH	RW	0x00	TIMER2 的通道 CRC 比较/捕获寄存器高字节

13.11.10. TIMER2 计数寄存器低字节 TL2(偏移: CCH)

比特	名称	属性	复位值	描述
7:0	TL2	RW	0x00	TIMER2 计数初值设置低字节

13.11.11. TIMER2 计数寄存器高字节 TH2(偏移: 0DH)

比特	名称	属性	复位值	描述
7:0	TH2	RW	0x00	TIMER2 计数初值设置高字节

13.11.12. TIMER2 中断标志状态寄存器 IRCON(偏移: C0H)

比特	名称	属性	复位值	描述
7	EXF2	RW	0x0	TIMER2 外部重载/捕获模式 2 标志位 外部重载/捕获发生时硬件置 1; 软件清 0
6	TF2	RW	0x0	TIMER2 溢出标志位 0: 未溢出 1: 溢出 硬件置 1, 软件清 0
5	IEX6	RW	0x0	外部中断 6 边沿标志 硬件置 1; 中断处理后自动清 0 或软件清 0
4	IEX5	RW	0x0	外部中断 5 边沿标志 硬件置 1; 中断处理后自动清 0 或软件清 0
3	IEX4	RW	0x0	外部中断 4 边沿标志 硬件置 1; 中断处理后自动清 0 或软件清 0
2	IEX3	RW	0x0	外部中断 3 边沿标志 硬件置 1; 中断处理后自动清 0 或软件清 0
1	IEX2	RW	0x0	外部中断 2 边沿标志 硬件置 1, 中断处理后自动清 0 或软件清 0
0	IEX7	RW	0x0	外部中断 7 边沿标志 硬件置 1, 中断处理后自动清 0 或软件清 0

13.12. 定时器 2 操作说明

13.12.1. 定时

a. 定时器配置例程

- 1) 配置 SCCM1 寄存器, 使能 TIMER2 的时钟
- 2) 配置 T2CON 寄存器, 设置 TIMER2 工作模式为定时器模式, 禁止重载功能, 预分频数 12

- 3) 配置 TH2, TL2 寄存器, 设置定时器初值
 - 4) 使能 TIEMR2 中断和总中断 (若采用查询方式可跳过此步)
 - 5) 溢出后, IRCON 的 TF2 位被置位, 并触发中断, 需要软件清除 TF2 标志位, 可采用轮询或中断方式处理。
- b. 重载定时器配置例程
- 1) 配置 SCCM1 寄存器, 使能 TIMER2 的时钟
 - 2) 配置 T2CON 寄存器, 设置 TIMER2 工作模式为定时器模式, 使能重载模式 0, 分频数 12
 - 3) 配置 TH2, TL2 寄存器, 设置定时器初值
 - 4) 配置 CRCL, CRCH 寄存器, 设置重载值
 - 5) 溢出后, IRCON 的 TF2 位被置位, 并触发中断, 需要软件清除 TF2 标志位, 可采用轮询或中断方式处理。
- c. 门控定时器配置流程
- 1) 配置 SCCM1 寄存器, 使能 TIMER2 的时钟
 - 2) 配置 T2CON 寄存器, 设置 TIMER2 工作模式为门控定时器模式, 禁止重载, 预分频数 12
 - 3) 配置 TH2, TL2 寄存器, 设置定时器初值
 - 4) 设置 P30 复用为 T2FF0, T2FF0 给高电平, TIMER2 工作; T2FF0 给低电平, TIMER2 停止, TH2, TL2 的值保持。

13.12.2. 比较

- a. 比较模式 0
- 1) 配置 SCCM1 寄存器, 使能 TIMER2 的时钟
 - 2) 配置 T2CON 寄存器, 设置 TIMER2 工作模式为定时器模式, 禁止重载功能, 预分频数 12, 设置为比较模式 0
 - 3) 配置 IOCFG_CTRLx 寄存器, 将相应管脚配置为比较/互补输出功能
 - 4) 设置计数周期 TH2, TL2
 - 5) 配置 CRC 通道比较值 CRCL, CRCH 寄存器
 - 6) 配置 CC1 通道比较值 CCL1, CCH1 寄存器
 - 7) 配置 CC2 通道比较值 CCL2, CCH2 寄存器
 - 8) 写 CCL3 寄存器, 设置输出极性
 - 9) 写 CCEN 寄存器, 使能比较模式
 - 10) 使能 TIMER2 中断和总中断
 - 11) 当 TIMER2 的计数值与设定的比较值相等时, 对应通道输出设定的电平
 - 12) 溢出后, IRCON 的 TF2 位被置位, 并触发溢出中断, 需要软件清除 TF2 标志位并重装计数周期 TH2, TL2。

b. 比较模式 1

- 1) 配置 SCCM1 寄存器，使能 TIMER2 的时钟
- 2) 配置 T2CON 寄存器，设置 TIMER2 工作模式为定时器模式，禁止重载功能，预分频数 12，设置为比较模式 1
- 3) 配置 IOC_CFG_CTRLx 寄存器，将相应管脚配置为比较/互补输出功能
- 4) 设置计数周期 TH2，TL2
- 5) 配置 CRC 通道比较值 CRCL，CRCH 寄存器
- 6) 配置 CC1 通道比较值 CCL1，CCH1 寄存器
- 7) 配置 CC2 通道比较值 CCL2，CCH2 寄存器
- 8) 配置 CCEN 寄存器，使能比较模式
- 9) 使能 TIMER2 中断和总中断
- 10) 当 TIMER2 的计数值与设定的比较值相等时，对应的比较器输出为 P1 口 P10、P11、P12 的电平值
- 11) 溢出后，IRCON 的 TF2 位被置位，并触发溢出中断，需要软件清除 TF2 标志位并重装计数周期 TH2，TL2，但比较的输出不受影响。

13.12.3. 捕获

a. 捕获模式 0

- 1) 配置 SCCM1 寄存器，使能 TIMER2 的时钟
- 2) 配置 T2CON 寄存器，设置 TIMER2 工作模式为定时器模式，禁止重载功能，预分频数 12
- 3) 配置 IOC_CFG_CTRLx 寄存器，将相应管脚配置为捕获输入功能
- 4) 设置计数初值 TH2，TL2
- 5) 配置 CCEN 寄存器，设置捕获的条件
- 6) 使能 TIEMR2 中断和总中断
- 7) 当检测到 CAP00..CAP02 满足条件时，TIMER2 的值保存到合适的寄存器 ((CRCH,CRCL)(CCH1,CCL1)(CCH2,CCL2))中

b. 捕获模式 1

- 1) 配置 SCCM1 寄存器，使能 TIMER2 的时钟
- 2) 配置 T2CON 寄存器，设置 TIMER2 工作模式为定时器模式，禁止重载功能，预分频数 12
- 3) 配置 IOC_CFG_CTRLx 寄存器，将相应管脚配置为捕获输入功能
- 4) 设置计数初值 TH2，TL2
- 5) 配置 CCEN 寄存器，设置捕获的条件
- 6) 使能 TIEMR2 中断和总中断
- 7) 当对捕获寄存器的低字节进行写操作，产生捕获，TIMER2 的值保存到合适的

寄存器((CRCH,CRCL)(CCH1,CCL1)(CCH2,CCL2))中

c. 捕获模式 2

- 1) 配置 SCCM1 寄存器，使能 TIMER2 的时钟
- 2) 配置 T2CON 寄存器，设置 TIMER2 工作模式为定时器模式，禁止重载功能，预分频数 12
- 3) 配置 IOC_CFG_CTRLx 寄存器，将相应管脚配置为捕获输入功能
- 4) 设置计数初值 TH2，TL2
- 5) 使能外部重载/捕获模式 2 中断
- 6) 使能 TIMER2 中断和总中断

注：TIMER2 捕获模式 2 可用外部 IO 输入或比较器 2 的输出进行触发。当选择比较器 2 的输出触发时，外部 IO 输入无效。

14. 高级定时器（TIMER3/TIMER4）

14.1. 概述

该设备内置了 2 个带有可编程分频器，向上计数的 16 位基本定时器：TIMER3 和 TIMER4。适用于多个场合，包括基本的定时，测量输入信号脉冲长度，产生输出波形（输出比较，PWM）。

14.2. 高级定时器主要特性

- 16 位向上计数和自动装载计数器
- 工作模式有 3 种
 - 基本定时模式
 - 事件计数模式
 - 门控定时模式
- 可更改的计数分频（12 分频/24 分频）
- 3 个独立通道
 - 输入捕获
 - 输出比较
 - PWM 生成
 - 3 个支持带互补输出的通道
- 支持死区时间的插入
- 刹车输入信号可以将定时器输出信号置于一个已知状态
- 如下事件发生时可产生中断
 - 更新中断：计数向上溢出
 - 外部重载
 - 捕获输入（捕获模式 2）
 - 输出比较
 - 刹车信号输入

14.3. 分频器

该高级定时器，计数时钟分频可以通过 $TIMERx_T2CON$ （ $x=3$ 时代表定时器 3 或 $x=4$ 时代表定时器 4，下同）的第 7 比特位选择，即 $TIMERx_T2CON[7]=0$ 为 12 分频， $TIMERx_T2CON[7]=1$ 为 24 分频。

14.4. 定时器模式

基本定时器模式，可通过配置寄存器 $TIMERx_T2CON$ 的第 0,1 比特位来选择该模

式，即 $TIMERx_T2CON[1]=0$ ， $TIMERx_T2CON[0]=1$ 。

14.5. 事件计数模式

事件计数模式，可通过配置寄存器 $TIMERx_T2CON$ 的第 0,1 比特位来选择该模式，即 $TIMERx_T2CON[1]=1$ ， $TIMERx_T2CON[0]=0$ 。该模式下当外部信号由高变低，则计数增加。在每个时钟的上升沿都会对外部输入信号进行采样。

注：外部输入引脚端口复用

14.6. 门控定时模式

门控定时模式，可通过配置寄存器 $TIMERx_T2CON$ 的第 0,1 比特位来选择该模式，即 $TIMERx_T2CON[1]=1$ ， $TIMERx_T2CON[0]=1$ 。该模式下当外部信号为高时，则计数增加。外部信号为低时，则计数增加停止。

注：外部输入引脚端口复用

14.7. 定时器重载功能

重载功能有 2 种模式，可通过配置寄存器 $TIMERx_T2CON$ 的第 3,4 比特位来选择重载模式，即 $TIMERx_T2CON[4]=1$ ， $TIMERx_T2CON[3]=0$ 选择模式 0； $TIMERx_T2CON[4]=1$ ， $TIMERx_T2CON[3]=1$ 选择模式 1。

重载模式 0：计数溢出后产生重载信号（自动重载）

重载模式 1：输入引脚信号有下降沿，则产生重载信号。

重载功能下，重载值为寄存器 $TIMERx_TH2$, $TIMERx_TL2$ 的值。

注：外部输入引脚端口复用

14.8. 定时器比较输出功能

14.8.1. 比较模式

$TIMER3$ 和 $TIMER4$ 比较输出功能有 2 种模式，可通过配置寄存器 $TIMERx_T2CON$ 的第 2 比特位来选择比较模式，即 $TIMERx_T2CON[2]=0$ 选择模式 0； $TIMERx_T2CON[2]=1$ 选择模式 1。

在 2 种比较模式中有 3 个比较输出通道即 CRC , $CC1$, $CC2$ ，3 个通道下对应 3 个端口和 3 个比较器。可通过寄存器 $TIMERx_CCEN$ 的 0,1 比特位选择 CRC 通道； $TIMERx_CCEN$ 的 2,3 比特位选择 $CC1$ 通道； $TIMERx_CCEN$ 的 4,5 比特位选择 $CC2$ 通道（具体配置参考寄存器描述）。

设定比较值时： CRC 通道对应寄存器 $TIMERx_CRCH$, $TIMERx_CRCL$ ； $CC1$ 通道对应寄存器 $TIMERx_CCH1$, $TIMERx_CCL1$ ； $CC2$ 通道对应寄存器 $TIMERx_CCH2$, $TIMERx_CCL2$ 。

比较输出和比较互补输出时：需要配置 $TIMERx_OUT_EN$ 的第 4,5,6 比特位来使能

比较输出；

配置 `TIMERx_OUT_EN` 的第 0,1,2 比特位来使能比较互补输出（具体配置参考寄存器描述）。

表 14-1 TIMER3 和 TIMER4 比较通道对应输出引脚

引脚名称	输出引脚名称	信号类型	注释(在使用时注意端口复用)
P12	COMP10	输出	与定时器 3 的比较通道 CRC 相应的比较输出引脚
P13	COMP10	输出	与定时器 3 的比较通道 CRC 相应的比较互补输出引脚
P14	COMP11	输出	与定时器 3 的比较通道 CC1 相应的比较输出引脚
P15	COMP11	输出	与定时器 3 的比较通道 CC1 相应的比较互补输出引脚
P16	COMP12	输出	与定时器 3 的比较通道 CC2 相应的比较输出引脚
P17	COMP12	输出	与定时器 3 的比较通道 CC2 相应的比较互补输出引脚
P24	COMP20	输出	与定时器 4 的比较通道 CRC 相应的比较输出引脚
P25	COMP20	输出	与定时器 4 的比较通道 CRC 相应的比较互补输出引脚
P26	COMP21	输出	与定时器 4 的比较通道 CC1 相应的比较输出引脚
P27	COMP21	输出	与定时器 4 的比较通道 CC1 相应的比较互补输出引脚
P44	COMP22	输出	与定时器 4 的比较通道 CC2 相应的比较输出引脚
P45	COMP22	输出	与定时器 4 的比较通道 CC2 相应的比较互补输出引脚

比较模式 0：当 `TIMER2` 的计数值等于比较设定的值时，相应比较器通道的比较输出信号变化。具体可由寄存器 `TIMERx_OUT_P[i]`($i=0/1/2$ 比特位和 4/5/6 比特位)控制，即

当 `TIMERx_OUT_P[i]`($i=4/5/6$)=0 时，相应通道的比较输出由高变低，

当 `TIMERx_OUT_P[i]`($i=0/1/2$)=0 时，相应通道的比较互补输出由低变高；

当 `TIMERx_OUT_P[i]`($i=4/5/6$)=1 时，相应通道的比较输出由低变高，

当 `TIMERx_OUT_P[i]`($i=0/1/2$)=1 时，相应通道的比较互补输出由高变低；

`TIMER2` 计数值溢出时，相应比较器通道的输出信号变为与比较器的值相等时输出信号相反的极性。

注： $i=4$ 对应 CRC 比较通道； $i=5$ 对应 CC1 比较通道； $i=6$ 对应 CC2 比较通道

$i=0$ 对应 CRC 比较互补通道； $i=1$ 对应 CC1 比较互补通道； $i=2$ 对应 CC2 比较互补通道

比较模式 1：当 `TIMER2` 的计数值等于比较器的值时，相应比较器通道的比较输出信号变化（具体由寄存器 `TIMERx_ID` 的第 0,1,2 位控制，即当计数值等于设定的比较值时，输出信号被激活，输出引脚输出 `TIMERx_ID[i]`($i=0/1/2$)的数值；`TIMER2` 计数值溢出时，相应比较器通道的输出信号不会变化。此模式下比较互补输出模式与比较输出模式一致。

14.8.2. PWM 模式（互补及死区插入）

脉冲宽度调制基于定时器的比较输出功能，可以通过设定比较值来改变占空比。

在 `TIMERx_T2CON[2]=0` 选择比较模式 0，来选择 PWM 模式，能够独立地设定每

个输出通道产生 PWM,其通过 $TIMERx_CCEN[i][j]=10(i=1,3,5 \quad j=0,2,4)$ 来选择 1 路 PWM 也可选择多路。

PWM 的极性及其互补的极性可以通过软件在寄存器中 $TIMERx_OUT_P[i](i=0/1/2 \text{ 和 } 4/5/6)$ 设置。其计数值和比较值始终进行比较用以确定是否触发比较输出信号的产生。

通过设置通道的比较值来产生不同占空比的 PWM。（CRC 通道对应寄存器 $TIMERx_CRCH$, $TIMERx_CRCL$ ；CC1 通道对应寄存器 $TIMERx_CCH1$, $TIMERx_CCL1$ ；CC2 通道对应寄存器 $TIMERx_CCH2$, $TIMERx_CCL2$ ）

PWM 信号的输出与互补输出可配置 $TIMERx_OUT_EN$ 的第 4,5,6 比特位来使能 PWM 输出；配置 $TIMERx_OUT_EN$ 的第 0,1,2 比特位来使能 PWM 互补输出。

此模式下还支持死区时间的插入，用户可根据具体需要来调整死区时间。配置寄存器 $TIMERx_IRCON$ 的第 0,1,2 比特位，每个通道将插入死区时间。需要插入的死区时间大小配置通过寄存器 $TIMERx_CCH3$ 和 $TIMERx_CCL3$ 。

注：死区时间功能需要 PWM 的输出及互补输出同时使能。

以互补输出都是高为例：OCiREF 为参考电压，下降沿为比较条件满足，上升沿为计数溢出。

- COMP0i 输出信号在计数值小于比较值时为高；计数值在大于等于比较值时为低。在计数溢出时，延迟死区时间后拉高。
- COMPN0i 输出信号在计数值小于比较值时为低；计数值在大于等于比较值时，延时死区时间后拉高。在计数溢出时输出拉低。

如果延迟大于当前有效的输出宽度(COMP0i 或者 COMPN0i)，则相应的通道不会产生脉。

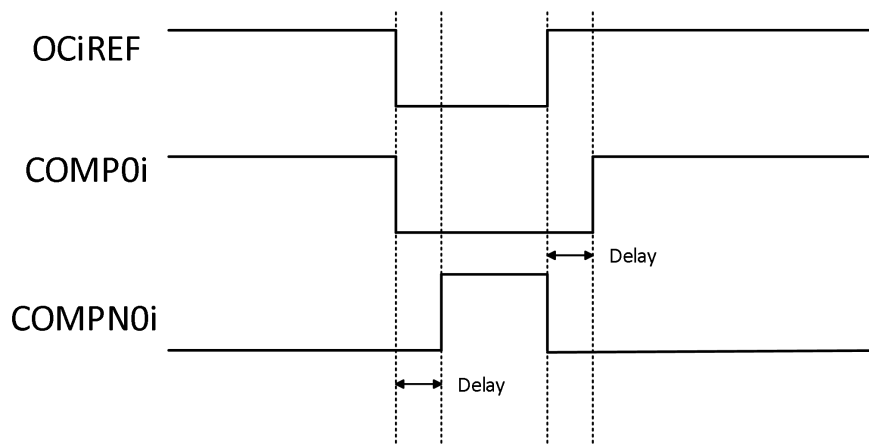


图 14-1 带死区的互补输出

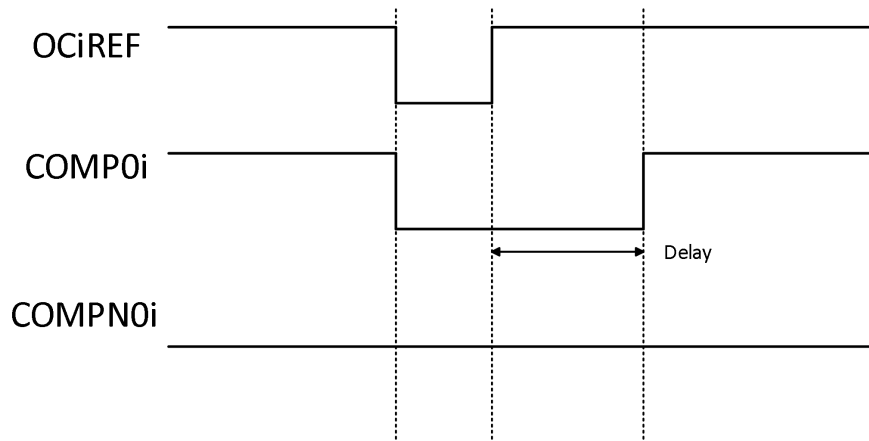


图 14-2 死区时间大于负脉冲

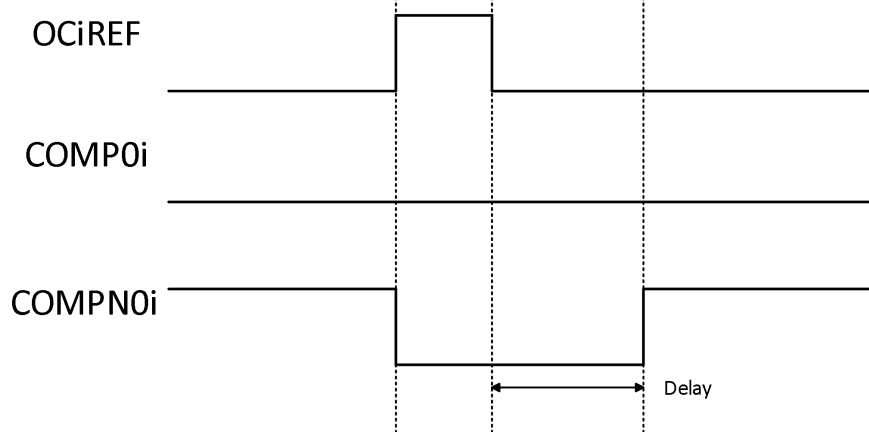


图 14-3 死区时间大于正脉冲

14.8.3. 刹车功能

刹车功能常用于马达控制中。当使用刹车功能时，根据相应的控制寄存器 `TIMERx_BREAK_OIS` 的第 0,1,2 和 4,5,6 位，比较输出和互补输出状态会被修改。

设置寄存器 `TIMERx_BOCEN` 第 4 比特位可以使能刹车功能。刹车信号的极性可通过寄存器 `TIMERx_BOCEN` 的第 5 比特位选择。刹车信号滤波可以寄存器 `TIMERx_BOCEN` 第 0,1,2,3 比特位选择不同的滤波。

当发生刹车时（外部刹车输入端出现设定的电平）：

- 比较输出及比较互补输出置于设定已知的电平状态
- 如果设置了 `TIMERx_IEN1` 的第 4 比特位，则会产生一个中断
- 如果设置了 `TIMERx_BOCEN` 的第 6 比特位，则会自动清除该事件标志

14.9. 定时器捕获功能

`TIMER3` 和 `TIMER4` 捕获功能有 3 种模式，该功能下实际的时间计数内容保存到相应的寄存器中。

捕获模式 0：检测到外部输入引脚的沿变化，则发生捕获。该模式下有 3 个通道对

应 3 个外部输入引脚和 3 个数据缓存寄存器。

表 14-2 TIMER3 和 TIMER4 捕获模式 0 通道对应输出引脚

引脚名称	输出引脚名称	信号类型	注释(在使用时注意端口复用)
P11	CAP10	输入	与 TIMER3 的 CRC 通道相应的通道输入引脚
P42	CAP11	输入	与 TIMER3 的 CC1 通道相应的通道输入引脚
P43	CAP12	输入	与 TIMER3 的 CC2 通道相应的通道输入引脚
P07	CAP20	输入	与 TIMER4 的 CRC 通道相应的通道输入引脚
P20	CAP21	输入	与 TIMER4 的 CC1 通道相应的通道输入引脚
P21	CAP22	输入	与 TIMER4 的 CC2 通道相应的通道输入引脚

可通过设置寄存器 `TIMERx_CCEN` 选择不同的通道即

`TIMERx_CCEN[i][j]=01`($i=1,3,5$ $j=0,2,4$)选择相应的捕获通道。

发生捕获时可读取相应的数据缓存寄存器的值，获取当前的实际计数值。

(CRC 通道对应寄存器 `TIMERx_CRCH`,`TIMERx_CRCL` ; CC1 通道对应寄存器 `TIMERx_CCH1` ,`TIMERx_CCL1` ;CC2 通道对应 寄存器 `TIMERx_CCH2`,`TIMERx_CCL2`)。

捕获的极性: CRC 通道的极性可通过 `TIMERx_T2CON` 的第 6 比特位选择上升沿或下降沿。 CC1 和 CC2 只支持上升沿。

捕获模式 1: 检测到专用的捕获寄存器的低字节写入数据，则发生捕获。定时器的值会被锁存到到相应的寄存器中。

该模式下有 3 个通道对应 3 个专用的捕获写入寄存器和 3 个数据缓存寄存器。

通过设置寄存器 `TIMERx_CCEN` 选择不同的通道即 `TIMERx_CCEN[i][j]=11`($i=1,3,5$ $j=0,2,4$)选择相应的捕获通道。

当专用捕获寄存器的低字节写入数据时，则发生捕获。(CRC 通道对应寄存器 `TIMERx_CRCL` ; CC1 通道对应寄存器 `TIMERx_CCL1` ; CC2 通道对应寄存器 `TIMERx_CCL2`)

发生捕获时读取相应通道的数据缓存寄存器的计数值。(CRC 通道对应寄存器 `TIMERx_CRCH`,`TIMERx_CRCL` ; CC1 通道对应寄存器 `TIMERx_CCH1` ,`TIMERx_CCL1` ; CC2 通道对应 寄存器 `TIMERx_CCH2`,`TIMERx_CCL2`)。

捕获模式 2: 检测到外部信号发生相应的变化时则可以产生捕获。

表 14-3 TIMER3 和 TIMER4 捕获模式 2 通道对应输出引脚

引脚名称	输出引脚名称	信号类型	注释(在使用时注意端口复用)
P40	T2EX1	输入	TIMER3 捕获通道相应输入引脚
P31	T2EX2	输入	TIMER4 捕获通道相应输入引脚

此模式下捕获的极性可通过寄存器 `TIMERx_IRCON` 的第 3,4 比特位选择下降沿, 上升沿, 双沿。此模式下可产生中断。

14.10. 定时器 3/4 中断

中断的产生需使能相应的中断源。

可产生中断如下事件发生时可产生中断:

- 更新中断: 计数向上溢出
- 外部重载
- 捕获输入 (捕获模式 2)
- 输出比较中断
- 刹车中断

14.11. 定时器 3/4 寄存器

TIMER3 寄存器基地址: 0x2180

偏置	名称	描述
0x00	TIMER3_IRCON	TIMER3 中断标志及控制寄存器
0x01	TIMER3_IEN1	TIMER3 比较中断控制寄存器
0x02	TIMER3_T2CON	TIMER3 控制寄存器
0x03	TIMER3_CCEN	TIMER3 模式使能寄存器
0x04	TIMER3_CCL1	TIMER3 比较/捕获寄存器低字节
0x05	TIMER3_CCH1	TIMER3 比较/捕获寄存器高字节
0x06	TIMER3_CCL2	TIMER3 比较/捕获寄存器低字节
0x07	TIMER3_CCH2	TIMER3 比较/捕获寄存器高字节
0x08	TIMER3_CCL3	TIMER3 死区寄存器低字节
0x09	TIMER3_CCH3	TIMER3 死区寄存器高字节
0x0A	TIMER3_CRCL	TIMER3 比较/捕获寄存器低字节
0x0B	TIMER3_CRCH	TIMER3 比较/捕获寄存器高字节
0x0C	TIMER3_TL2	TIMER3 计数寄存器低字节
0x0D	TIMER3_TH2	TIMER3 计数寄存器高字节
0x0E	TIMER3_BOEN	TIMER3 刹车控制寄存器
0x0F	TIMER3_ID	TIMER3 比较模式 1 数据寄存器
0x10	TIMER3_BREAK_OIS	TIMER3 刹车输出状态控制寄存器
0x11	TIMER3_OUT_P	TIMER3 比较输出极性选择寄存器
0x12	TIMER3_OUT_EN	TIMER3 通道使能寄存器

TIMER4 寄存器基地址: 0x21A0

偏置	名称	描述
0x00	TIMER4_IRCON	TIMER4 中断标志及控制寄存器

0x01	TIMER4_IEN1	TIMER4 比较中断控制寄存器
0x02	TIMER4_T2CON	TIMER4 控制寄存器
0x03	TIMER4_CCEN	TIMER4 模式使能寄存器
0x04	TIMER4_CCL1	TIMER4 比较/捕获寄存器低字节
0x05	TIMER4_CCH1	TIMER4 比较/捕获寄存器高字节
0x06	TIMER4_CCL2	TIMER4 比较/捕获寄存器低字节
0x07	TIMER4_CCH2	TIMER4 比较/捕获寄存器高字节
0x08	TIMER4_CCL3	TIMER4 死区寄存器低字节
0x09	TIMER4_CCH3	TIMER4 死区寄存器高字节
0x0A	TIMER4_CRCL	TIMER4 比较/捕获寄存器低字节
0x0B	TIMER4_CRCH	TIMER4 比较/捕获寄存器高字节
0x0C	TIMER4_TL2	TIMER4 计数寄存器低字节
0x0D	TIMER4_TH2	TIMER4 计数寄存器高字节
0x0E	TIMER4_BOEN	TIMER4 刹车控制寄存器
0x0F	TIMER4_ID	TIMER4 比较模式 1 数据寄存器
0x10	TIMER4_BREAK_OIS	TIMER4 刹车输出状态控制寄存器
0x11	TIMER4_OUT_P	TIMER4 比较输出极性选择寄存器
0x12	TIMER4_OUT_EN	TIMER4 通道使能寄存器

14.11.1. TIMER3/4 中断标志及控制寄存器 TIMERi_IRCON(偏移:00H, 此处 x 代表 3、4, 下同)

比特	名称	属性	复位值	描述
7	EXF2	RW	0x0	外部重载/捕获 2 标志位 外部重载/捕获 2 发生时硬件置 1; 软件清 0
6	TF2	RW	0x0	溢出标志 硬件置 1; 软件清 0
5	BK_FLAG	RW	0x0	刹车标志位 硬件置 1; 溢出自动清 0 或软件清 0
4:3	EXF2_SEL	RW	0x0	捕获触发方式选择 00: 下降沿触发 01: 上升沿触发 1x: 双沿触发
2	COMPL_EN2	RW	0x0	通道 CC2 的死区时间功能 0: 关闭 1: 打开
1	COMPL_EN1	RW	0x0	通道 CC1 的死区时间功能 0: 关闭 1: 打开
0	COMPL_EN0	RW	0x0	通道 CRC 的死区时间功能

				0: 关闭 1: 打开
--	--	--	--	----------------

14.11.2. TIMER3/4 中断请求控制寄存器 TIMERi_IEN1(偏移: 01H)

比特	名称	属性	复位值	描述
7	EXEN2	RW	0x0	外部重载/捕获 2 中断使能位 0: 禁止 1: 使能
6:5	-	-	-	保留
4	BK_MSK	RW	0x1	刹车中断屏蔽位 0: 不屏蔽 1: 屏蔽
3	-	-	-	保留
2	INT_ACK2	W	0x0	比较中断响应位 写 1 清 CC1 比较中断
1	INT_ACK1	W	0x0	比较中断响应位 写 1 清 CC0 比较中断
0	INT_ACK0	W	0x0	比较中断响应位 写 1 清 CRC 比较中断

14.11.3. TIMER3/4 控制寄存器 TIMERi_T2CON(偏移: 02H)

比特	名称	属性	复位值	描述
7	T2PS	RW	0x0	预分频选择 0: TIMER 以晶振频率的 1/12 计数 1: TIMER 以晶振频率的 1/24 计数
6	I3FR	RW	0x0	外部中断 3 的沿选择(捕获模式 0) 0: 下降沿 1: 上升沿
5	I2FR	RW	0x0	外部中断 2 的沿选择 0: 下降沿 1: 上升沿
4	T2R1	RW	0x0	重载模式选择 0x: 重载禁止 10: 模式 0 11: 模式 1
3	T2R0	RW	0x0	
2	T2CM	RW	0x0	比较模式选择 0: 模式 0 1: 模式 1
1	T2I1	RW	0x0	输入选择 00: TIMER 停止 01: 定时器模式
0	T2I0	RW	0x0	

				10: 事件计数模式 11: 门控定时模式
--	--	--	--	--------------------------

14.11.4. TIMER3/4 模式是能寄存器 TIMERi_CCEN(偏移: 03H)

比特	名称	属性	复位值	描述
7:6	-	-	-	保留
5:4	Cocah2 Cocal2	RW	0x0	比较/捕获模式下, CC2 寄存器
				COCAH2 COCAL2 描述
				0 0 比较/捕获禁止
				0 1 通道 CC2 的上升沿捕获
				1 0 通道 CC2 比较输出选择
3:2	Cocah1 Cocal1	RW	0x0	比较/捕获模式下, CC1 寄存器
				COCAH1 COCAL1 描述
				0 0 比较/捕获禁止
				0 1 通道 CC1 的上升沿捕获
				1 0 通道 CC1 比较输出选择
1:0	Cocah0 Cocal0	RW	0x0	比较/捕获模式下, CRC 寄存器
				COCAH0 COCAL0 描述
				0 0 比较/捕获禁止
				0 1 通道 CRC 的上升沿捕获
				1 0 通道 CRC 比较输出选择
				1 1 写 TIMERi_CRCL 寄存器捕获

14.11.5. TIMER3/4 比较/捕获寄存器 TIMERi_CCL1(偏移: 04H)

比特	名称	属性	复位值	描述
7:0	TIMERi_CCL1	RW	0x00	TIMER 的通道 CC1 比较/捕获寄存器低字节

14.11.6. TIMER3/4 比较/捕获寄存器 TIMERi_CCH1(偏移: 05H)

比特	名称	属性	复位值	描述
7:0	TIMERi_CCH1	RW	0x00	TIMER 的通道 CC1 比较/捕获寄存器高字节

14.11.7. TIMER3/4 比较/捕获寄存器 TIMERi_CCL2(偏移: 06H)

比特	名称	属性	复位值	描述
7:0	TIMERi_CCL2	RW	0x00	TIMER 的通道 CC2 比较/捕获寄存器低字节

14.11.8. TIMER3/4 比较/捕获寄存器 TIMERi_CCH2(偏移: 07H)

比特	名称	属性	复位值	描述
7:0	TIMERi_CCH2	RW	0x00	TIMER 的通道 CC2 比较/捕获寄存器高字节

14.11.9. TIMER3/4 死区时间寄存器低字节 TIMERi_CCL3(偏移: 08H)

比特	名称	属性	复位值	描述
7:0	TIMERi_CCL3	RW	0x00	TIMER 的死区时间寄存器低字节

14.11.10. TIMER3/4 死区时间寄存器高字节 TIMERi_CCH3(偏移: 09H)

比特	名称	属性	复位值	描述
7:0	TIMERi_CCH3	RW	0x00	TIMER 的死区时间寄存器高字节

14.11.11. TIMER3/4 比较/捕获寄存器 TIMERi_CRCL(偏移: 0AH)

比特	名称	属性	复位值	描述
7:0	TIMERi_CRCL	RW	0x00	TIMER 的通道 CRC 比较/捕获寄存器低字节

14.11.12. TIMER3/4 比较/捕获寄存器 TIMERi_CRCH(偏移: 0BH)

比特	名称	属性	复位值	描述
7:0	TIMERi_CRCH	RW	0x00	TIMER 的通道 CRC 比较/捕获寄存器高字节

14.11.13. TIMER3/4 计数寄存器 TIMERi_TL2(偏移: 0CH)

比特	名称	属性	复位值	描述
7:0	TIMERi_TL2	RW	0x00	TIMER 的计数寄存器低字节

14.11.14. TIMER3/4 计数寄存器 TIMERi_TH2(偏移: 0DH)

比特	名称	属性	复位值	描述
7:0	TIMERi_TH2	RW	0x00	TIMER 的计数寄存器高字节

14.11.15. TIMER3/4 刹车控制寄存器 TIMERi_BOCEN(偏移: 0EH)

比特	名称	属性	复位值	描述
7	-	-	-	保留
6	AUTO_EN	RW	0x0	自动清刹车标志使能位 0: 软件写 0 清 1: 软件写 0 清或溢出自动清
5	BK_POLA	RW	0x0	刹车输入极性选择位 0: 低电平 1: 高电平
4	BK_EN	RW	0x0	刹车使能位 0: 禁止 1: 使能
3:0	SAMPLE_FILTER	RW	0x0	刹车信号滤波 0000: 无滤波 0001: 信号宽度大于 2 个时钟周期 0010: 信号宽度大于 4 个时钟周期 0011: 信号宽度大于 8 个时钟周期 0100: 信号宽度大于 12 个时钟周期

				0101: 信号宽度大于 16 个时钟周期 0110: 信号宽度大于 24 个时钟周期 0111: 信号宽度大于 32 个时钟周期 1000: 信号宽度大于 48 个时钟周期 1001: 信号宽度大于 64 个时钟周期 1010: 信号宽度大于 80 个时钟周期 1011: 信号宽度大于 96 个时钟周期 1100: 信号宽度大于 128 个时钟周期 1101: 信号宽度大于 160 个时钟周期 1110: 信号宽度大于 192 个时钟周期 1111: 信号宽度大于 256 个时钟周期
--	--	--	--	---

14.11.16. TIMER3/4 比较模式 1 数据寄存器 TIMERi_ID(偏移: 0FH)

比特	名称	属性	复位值	描述
7:3	-	-	-	保留
2	TIMERi_ID2	WO	0x0	比较模式 1 的 CC2 通道写数据寄存器
1	TIMERi_ID1	WO	0x0	比较模式 1 的 CC1 通道写数据寄存器
0	TIMERi_ID0	WO	0x0	比较模式 1 的 CRC 通道写数据寄存器

14.11.17. TIMER3/4 刹车输出状态控制寄存器 TIMERi_BREAK_OIS(偏移: 10H)

比特	名称	属性	复位值	描述
7	-	-	-	保留
6	BREAK_OIS_2	RW	0x0	通道 CC2 比较输出空闲状态 0: 发生刹车, 比较输出为 0 1: 发生刹车, 比较输出为 1
5	BREAK_OIS_1	RW	0x0	通道 CC1 比较输出空闲状态 0: 发生刹车, 比较输出为 0 1: 发生刹车, 比较输出为 1
4	BREAK_OIS_0	RW	0x0	通道 CRC 比较输出空闲状态 0: 发生刹车, 比较输出为 0 1: 发生刹车, 比较输出为 1
3	-	-	-	保留
2	BREAK_OIS_N2	RW	0x0	通道 CC2 比较互补输出空闲状态 0: 发生刹车, 比较互补输出为 0 1: 发生刹车, 比较互补输出为 1
1	BREAK_OIS_N1	RW	0x0	通道 CC1 比较互补输出空闲状态 0: 发生刹车, 比较互补输出为 0 1: 发生刹车, 比较互补输出为 1
0	BREAK_OIS_N0	RW	0x0	通道 CRC 比较互补输出空闲状态 0: 发生刹车, 比较互补输出为 0

				1: 发生刹车, 比较互补输出为 1
--	--	--	--	--------------------

14.11.18. TIMER3/4 比较输出极性选择寄存器 TIMERi_OUT_P(偏移: 11H)

比特	名称	属性	复位值	描述
7	-	-	-	保留
6	OUT_P_2	RW	0x0	通道 CC2 比较输出极性选择位 0: 激活为低电平(高电平变为低电平) 1: 激活为高电平(低电平变为高电平)
5	OUT_P_1	RW	0x0	通道 CC1 比较输出极性选择位 0: 激活为低电平 1: 激活为高电平
4	OUT_P_0	RW	0x0	通道 CRC 比较输出极性选择位 0: 激活为低电平 1: 激活为高电平
3	-	-	-	保留
2	OUT_P_N2	RW	0x0	通道 CC2 比较互补输出极性选择位 0: 激活为高电平 1: 激活为低电平
1	OUT_P_N1	RW	0x0	通道 CC1 比较互补输出极性选择位 0: 激活为高电平 1: 激活为低电平
0	OUT_P_N0	RW	0x0	通道 CRC 比较互补输出极性选择位 0: 激活为高电平 1: 激活为低电平

14.11.19. TIMER3/4 通道使能寄存器 TIMERi_OUT_EN(偏移: 12H)

比特	名称	属性	复位值	描述
7	-	-	-	保留
6	OUT_EN_2	RW	0x0	通道 CC2 比较输出使能位 0: 禁止 1: 使能
5	OUT_EN_1	RW	0x0	通道 CC1 比较输出使能位 0: 禁止 1: 使能
4	OUT_EN_0	RW	0x0	通道 CRC 比较输出使能位 0: 禁止 1: 使能
3	-	-	-	保留
2	OUT_EN_N2	RW	0x0	通道 CC2 比较互补输出使能位 0: 禁止 1: 使能

1	OUT_EN_N1	RW	0x0	通道 CC1 比较互补输出使能位 0: 禁止 1: 使能
0	OUT_EN_N0	RW	0x0	通道 CRC 比较互补输出使能位 0: 禁止 1: 使能

14.12. 定时器 3/4 操作说明

TIMER3 与 TIMER4 的配置相同，以下将使用 TIMERx 代表

14.12.1. 定时

a. 定时器配置例程

- 1) 配置 SCCM1 寄存器，使能 TIMERx 的时钟
- 2) 配置 TIMERx_T2CON 寄存器,设置 TIMERx 工作模式为定时器模式,禁止重载,预分频数 12
- 3) 配置 TIMERx_TH2, TIMERx_TL2 寄存器，设置定时器初值
- 4) 使能 TIMERx 中断和总中断（若采用查询方式可跳过此步）
- 5) 溢出后，TIMERx_IRCON 的 TF2 位被置位，并触发中断，需要软件清除 TF2 标志位并重装定时器初值（若不重装，则定时器溢出后从 0 开始计数）

b. 重载定时器配置例程

- 1) 配置 SCCM1 寄存器，使能 TIMERx 的时钟
- 2) 配置 TIMERx_T2CON 寄存器,设置 TIMERx 工作模式为定时器模式,使能重载模式 0,预分频数 12
- 3) 配置 TIMERx_TH2, TIMERx_TL2 寄存器，设置定时器初值（也是重载值）
- 4) 使能 TIMERx 中断和总中断（若采用查询方式可跳过此步）
- 5) 溢出后，TIMERx_IRCON 的 TF2 位被置位，并触发中断，需要软件清除 TF2 标志位，不需要再重装定时器初值（会自动重载）

c. 门控定时器配置例程

- 1) 配置 SCCM1 寄存器，使能 TIMERx 的时钟
- 2) 配置 TIMERx_T2CON 寄存器的,设置 TIMERx 工作模式为门控定时器模式,使能重载模式 0,预分频数 12
- 3) 配置 TIMERx_TH2, TIMERx_TL2 寄存器，设置定时器初值（也是重载值）
- 4) 设置 P41 复用为 T2FF1(或 P32 复用为 T2FF2), T2FF1/T2FF2 给高电平, TIMERx 工作; T2FF1/T2FF2 给低电平, TIMERx 停止, TIMERx_TH2, TIMERx_TL 的值保持

14.12.2. 比较

a. 比较模式 0 例程

- 1) 配置 SCCM1 寄存器, 使能 TIMERx 的时钟
- 2) 配置 TIMERx_T2CON 寄存器, 设置 TIMERx 工作模式为定时器模式, 使能重载模式 0, 预分频数 12, 设置为比较模式 0
- 3) 配置 TIMERx_TH2, TIMERx_TL2 寄存器, 设置定时器初值 (也是重载值)
- 4) 配置 IOC_CFG_CTRLx 寄存器, 将相应管脚配置为比较/互补输出功能
- 5) 配置 CRC 通道比较值 TIMERx_CRCL, TIMERx_CRCH 寄存器
- 6) 配置 CC1 通道比较值 TIMERx_CCL1, TIMERx_CCH1 寄存器
- 7) 配置 CC2 通道比较值 TIMERx_CCL2, TIMERx_CCH2 寄存器
- 8) 配置 TIMERx_OUT_EN、TIMERx_OUT_P 寄存器, 设置输出使能和输出极性
- 9) 配置 TIMERx_CCEN 寄存器, 使能比较输出通道 CRC/CC1/CC2
- 10) 配置 TIMERx_CCH3, TIMERx_CCL3 寄存器, 配置死区时间
- 11) 配置 TIMERx_IRCON 寄存器, 设置是否打开 CRC/CC1/CC2 通道的死区功能
- 12) 配置 TIMERx_BOEN 寄存器, 设置刹车输入的极性, 滤波时间, 是否使能刹车功能。(若不使能刹车功能, 则 12~15 步可跳过)
- 13) 配置 TIMER4_BREAK_OIS 寄存器, 设置比较输出和互补输出在发生刹车后的输出电平
- 14) 配置 IOC_CFG_CTRLx 寄存器, 将相应管脚设置为刹车输入功能
- 15) 配置 TIMERx_IEN, 设置刹车中断不屏蔽
- 16) 使能 TIMERx 中断和总中断

中断响应:

溢出中断——TIMERx_IRCON 的 TF2 位被置位, 并触发溢出中断, 需要软件清除 TF2 标志位, 并清除 CC1、CC2、CRC 通道比较响应标志 (TIMERx_IEN1 低 3 位)。

刹车中断——TIMERx_IRCON 的 BK_FLAG 位被置位, 并触发刹车中断, 需要软件清除 BK_FLAG 标志位。

b. 比较模式 1 例程

- 1) 配置 SCCM1 寄存器, 使能 TIMERx 的时钟
- 2) 配置 TIMERx_T2CON 寄存器, 设置 TIMERx 工作模式为定时器模式, 使能重载模式 0, 预分频数 12, 设置为比较模式 1
- 3) 配置 TIMERx_TH2, TIMERx_TL2 寄存器, 设置定时器初值 (也是重载值)
- 4) 配置 IOC_CFG_CTRLx 寄存器, 将相应管脚配置为比较/互补输出功能
- 5) 配置 CRC 通道比较值 TIMERx_CRCL, TIMERx_CRCH 寄存器
- 6) 配置 CC1 通道比较值 TIMERx_CCL1, TIMERx_CCH1 寄存器
- 7) 配置 CC2 通道比较值 TIMERx_CCL2, TIMERx_CCH2 寄存器

- 8) 配置 `TIMERx_OUT_EN` 寄存器, 使能通道 `CRC/CC1/CC2` 输出
- 9) 配置 `TIMERx_CCEN` 寄存器, 使能比较输出通道 `CRC/CC1/CC2`
- 10) 配置 `TIMERx_ID` 寄存器, 设置比较通道输出的电平
- 11) 使能 `TIMERx` 中断和总中断
- 12) 当 `TIMERx` 的计数值与设定的比较值相等时, 对应的比较通道输出 `TIMERx_ID` 的写入值
- 13) 溢出后, `TIMERx_IRCON` 的 `TF2` 位被置位, 并触发溢出中断, 需要软件清除 `TF2` 标志位, 并清除 `CC1/CC2/CRC` 通道比较响应标志(`TIMERx_IEN1` 低 3 位)

14.12.3. 捕获

a. 捕获模式 0 例程

- 1) 配置 `SCCM1` 寄存器, 使能 `TIMERx` 的时钟
- 2) 配置 `IOCFG_CTRLx` 寄存器, 将相应管脚配置为捕获输入功能
- 3) 配置 `TIMERx_T2CON` 寄存器, 设置 `TIMERx` 工作模式为定时器模式, 使能重载模式 0, 预分频数 12
- 4) 配置 `TIMERx_TH2`, `TIMERx_TL2` 寄存器, 设置定时器初值 (也是重载值)
- 5) 配置 `TIMERx_CCEN` 寄存器, 使能通道 `CRC/CC1/CC2` 上升沿捕获
- 6) 当检测到 `CAP10..CAP12` (或 `CAP20..CAP22`) 满足条件时, `TIMERx` 的值保存到合适的寄存器 ((`TIMERx_CRCH`, `TIMERx_CRCL`) (`TIMERx_CCH1`, `TIMERx_CCL1`) (`TIMERx_CCH2`, `TIMERx_CCL2`)) 中

b. 捕获模式 1 例程

- 1) 配置 `SCCM1` 寄存器, 使能 `TIMERx` 的时钟
- 2) 配置 `TIMERx_T2CON` 寄存器, 设置 `TIMERx` 工作模式为定时器模式, 使能重载模式 0, 预分频数 12
- 3) 配置 `TIMERx_TH2`, `TIMERx_TL2` 寄存器, 设置定时器初值 (也是重载值)
- 4) 配置 `TIMERx_CCEN` 寄存器, 使能通道 `CRC/CC1/CC2` 写寄存器捕获
- 5) 当对捕获寄存器的低字节进行写操作时, 产生捕获, `TIMERx` 的值保存到合适的寄存器 ((`TIMERx_CRCH`, `TIMERx_CRCL`) (`TIMERx_CCH1`, `TIMERx_CCL1`) (`TIMERx_CCH2`, `TIMERx_CCL2`)) 中

c. 捕获模式 2 例程

- 1) 配置 `SCCM1` 寄存器, 使能 `TIMERx` 的时钟
- 2) 配置 `IOCFG_CTRLx` 寄存器, 将相应管脚配置为捕获输入功能 `T1EX2` (或 `T2EX2`)
- 3) 配置 `TIMERx_T2CON` 寄存器, 设置 `TIMERx` 工作模式为定时器模式, 使能重载模式 0, 预分频数 12
- 4) 配置 `TIMERx_TH2`, `TIMERx_TL2` 寄存器, 设置定时器初值 (也是重载值)

- 5) 配置 `TIMERx_IRCON` 寄存器，设置 `T1EX2`（或 `T2EX2`）捕获极性
- 6) 配置 `TIMERx_IEN1` 寄存器，使能重载/捕获 2 中断
- 7) 使能 `TIMERx` 中断和总中断。
- 8) 当检测到 `T1EX2`（或 `T2EX2`）满足条件时，`TIMERx` 的值保存到合适的寄存器（（`TIMERx_CRCH`，`TIMERx_CRCL`）（`TIMERx_CCH1`，`TIMERx_CCL1`）（`TIMERx_CCH2`，`TIMERx_CCL2`））中。

中断响应：

溢出中断——`TIMERx_IRCON` 的 `TF2` 位被置位，并触发溢出中断，需要软件清除 `TF2` 标志位。

捕获中断——`TIMERx_IRCON` 的 `EXF2` 位被置位，并触发捕获中断，需要软件清除 `EXF2` 标志位。

注：`TIMER3` 捕获模式 2 可用外部 IO 输入或比较器 2 的输出进行触发。当选择比较器 2 的输出触发时，外部 IO 输入无效。

15. 实时时钟（RTC）

15.1. 概述

该设备带有一个独立的实时计时器。它提供了一个时钟，日期和可编程的闹钟。RTC 还包括一个用于唤醒低功耗的单元。

RTC 包含了对世纪/年/月/日/小时（24 小时制）/分/秒的二进制编码，对 28 天、29 天（闰年）、30 天和 31 天的月份自动执行调整。

15.2. RTC 主要特性

- 时钟源可为片外 32.768K 低速晶振、内部 32K 低速 RC 振荡器
- 日历功能：世纪/年/月/日/小时（24 小时制）/分/秒
- 月份天数自动调整
- 周期定时功能：支持日/时/分/秒产生周期的 TICK 中断
- 闹钟功能：当日历时间与设置的闹钟时间匹配则会产生 ALARM 中断

15.3. RTC 功能描述

该 RTC 单元功能的实现由一组可读写的 8 位寄存器控制具体（具体参照 RTC 寄存器描述）。时钟选择具体参照寄存器 LP_CTRL1（具体参照第 8 章 时钟控制）。

15.4. RTC 寄存器

RTC 寄存器基地址：0x2080

偏置	名称	描述
0x01	RTC_HOUR_SEL	小时配置寄存器
0x02	RTC_MIN_SEL	分钟配置寄存器
0x03	RTC_SEC_SEL	秒配置寄存器
0x05	RTC_YEAR_SEL	年配置寄存器
0x06	RTC_MON_SEL	月份配置寄存器
0x07	RTC_DATE_SEL	日期配置寄存器
0x08	RTC_CENTURY_SEL	世纪配置寄存器
0x09	RTC_ALARM_HOUR_SEL	闹铃小时配置寄存器
0x0A	RTC_ALARM_MIN_SEL	闹铃分钟配置寄存器
0x0B	RTC_ALARM_SEC_SEL	闹铃秒配置寄存器
0x0D	RTC_ALARM_YEAR_SEL	闹铃年配置寄存器
0x0E	RTC_ALARM_MON_SEL	闹铃月份配置寄存器
0x0F	RTC_ALARM_DATE_SEL	闹铃日期配置寄存器
0x10	RTC_SEC_INIT_CNT_SEL_3	基本计数器[31:24]寄存器

0x11	RTC_SEC_INIT_CNT_SEL_2	基本计数器[23:16]寄存器
0x12	RTC_SEC_INIT_CNT_SEL_1	基本计数器[15:8]寄存器
0x13	RTC_SEC_INIT_CNT_SEL_0	基本计数器[7:0]寄存器
0x14	RTC_TICK_CTRL_SEL	Tick 中断配置寄存器
0x15	RTC_ALARM_CTRL_SEL	Alarm 中断配置寄存器
0x16	RTC_RELOAD_SEL	重新配置 RTC 寄存器使能寄存器
0x17	RTC_START_SEL	RTC 开始工作控制寄存器
0x18	RTC_CLR_INIT_SEL	RTC 中断标志清除寄存器

15.4.1. 小时配置寄存器(偏移: 01H)

比特	名称	属性	复位值	描述
7:0	RTC_HOUR_SEL	RW	0x00	Hour 配置寄存器（范围为 0~59） W: 配置初始 hour; R: RTC 计数 hour 值;

15.4.2. 分钟配置寄存器(偏移: 02H)

比特	名称	属性	复位值	描述
7:0	RTC_MIN_SEL	RW	0x00	Minute 配置寄存器（范围为 0~59） W: 配置初始 Minute; R: RTC 计数 Minute 值;

15.4.3. 秒配置寄存器(偏移: 03H)

比特	名称	属性	复位值	描述
7:0	RTC_SEC_SEL	RW	0x00	Second 配置寄存器（范围为 0~59） W: 配置初始 Second; R: RTC 计数 Second 值;

15.4.4. 年配置寄存器(偏移: 05H)

比特	名称	属性	复位值	描述
7:0	RTC_YEAR_SEL	RW	0x00	Year 配置寄存器（范围为 0~99） W: 配置初始 Year; R: RTC 计数 Year 值;

15.4.5. 月配置寄存器(偏移: 06H)

比特	名称	属性	复位值	描述
7:0	RTC_MON_SEL	RW	0x00	Month 配置寄存器（范围为 0~59） W: 配置初始 Month; R: RTC 计数 Month 值;

15.4.6. 日期配置寄存器(偏移: 07H)

比特	名称	属性	复位值	描述
----	----	----	-----	----

7:0	RTC_DATE_SEL	RW	0x00	Month 配置寄存器（范围为 0~59） W：配置初始 Month； R：RTC 计数 Month 值；
-----	--------------	----	------	---

15.4.7. 世纪配置寄存器(偏移：08H)

比特	名称	属性	复位值	描述
7:0	RTC_CENTURY_SEL	RW	0x00	Century 配置寄存器（范围为 0~59） W：配置初始 Century； R：RTC 计数 Century 值；

15.4.8. 闹铃小时配置寄存器(偏移：09h)

比特	名称	属性	复位值	描述
7:0	ALARM_HOUR	RW	0x00	闹铃 Hour 配置寄存器 W：配置初始闹铃 Hour； R：读取闹铃 Hour 值；

15.4.9. 闹铃分钟配置寄存器(偏移：0AH)

比特	名称	属性	复位值	描述
7:0	ALARM_MINUTE	RW	0x00	闹铃 Minute 配置寄存器 W：配置初始闹铃 Minute； R：读取闹铃 Minute 值；

15.4.10. 闹铃秒配置寄存器(偏移：0BH)

比特	名称	属性	复位值	描述
7:0	ALARM_SECOND	RW	0x00	闹铃 Second 配置寄存器 W：配置初始闹铃 Second； R：读取闹铃 Second 值；

15.4.11. 闹铃年配置寄存器(偏移：0DH)

比特	名称	属性	复位值	描述
7:0	ALARM_YEAR	RW	0x00	闹铃 Year 配置寄存器 W：配置初始闹铃 Year； R：读取闹铃 Year 值；

15.4.12. 闹铃月份配置寄存器(偏移：0EH)

比特	名称	属性	复位值	描述
7:0	ALARM_MONTH	RW	0x00	闹铃 Month 配置寄存器 W：配置初始闹铃 Month； R：读取闹铃 Month 值；

15.4.13. 闹铃日期状态寄存器(偏移：0FH)

比特	名称	属性	复位值	描述
----	----	----	-----	----

7:0	ALARM_DATE	RW	0x00	闹铃 Date 配置寄存器 W: 配置初始闹铃 Date; R: 读取闹铃 Date 值;
-----	------------	----	------	---

15.4.14. 基本计数器 SEC_CNT_3~SEC_CNT_0 秒计数校准寄存器(偏移: 10H~13H)

比特	名称	属性	复位值	描述
31:0	SEC_CNT	RW	0x8000	秒计数校准寄存器 当采用内部 RC_32K 作为 RTC 计数时钟时, 需要配置此寄存器。配置值为使用内部 RC_32K 时钟计数 1 秒需要的 cycle 个数 W: 配置秒计数初始值; R: 读取秒计数初始值;

注: 该寄存器由 Sec_cnt_3、Sec_cnt_2、Sec_cnt_1、Sec_cnt_0 依次拼接而成, 其中 Sec_cnt_3 为最高 byte, Sec_cnt_0 为最低 byte。

15.4.15. TICK 中断配置寄存器(偏移: 14H)

比特	名称	属性	复位值	描述
7:3	RSV	-	-	保留
2:1	TICK_SEL	RW	0x0	Tick 中断源头选择 00: Second 01: Minute 10: Hour 11: Date
0	TICK_INT_EN	RW	0x0	Tick 中断使能位 0: 不使能 1: 使能

15.4.16. ALARM 中断配置寄存器(偏移: 15H)

比特	名称	属性	复位值	描述
7	RSV	-	-	保留
6	ALARM_YEAR_SEL	RW	0x0	闹铃中断选择位: 0: 闹铃中断不选中 Year 1: 闹铃中断选中 Year
5	ALARM_MON_SEL	RW	0x0	闹铃中断选择位: 0: 闹铃中断不选中 Month 1: 闹铃中断选中 Month
4	ALARM_DATE_SEL	RW	0x0	闹铃中断选择位: 0: 闹铃中断不选中 Date 1: 闹铃中断选中 Date
3	ALARM_HOUR_SEL	RW	0x0	闹铃中断选择位:

				0: 闹铃中断不选中 Hour 1: 闹铃中断选中 Hour
2	ALARM_MIN_SEL	RW	0x0	闹铃中断选择位: 0: 闹铃中断不选中 Minute 1: 闹铃中断选中 Minute
1	ALARM_SEC_SEL	RW	0x0	闹铃中断选择位: 0: 闹铃中断不选中 Second 1: 闹铃中断选中 Second
0	ALARM_INT_EN	RW	0x0	闹铃中断使能位 0: 不使能 1: 使能

注：使能闹铃中断后，若 RTC 当前各寄存器计数值与 Alarm_Ctrl[6:1]选中中断的寄存器的值全部相等时，产生中断。

15.4.17. 重新配置 RTC 寄存器使能寄存器(偏移：16H)

比特	名称	属性	复位值	描述
7:1	RSV	-	-	保留
0	RTC_RELOAD	RW	0x0	RTC 寄存器重载使能位 0: 不重载 RTC 寄存器值; 1: 重载 RTC 寄存器值;

注：此处重新加载的寄存器为：Hour、Minute、Second、Year、Month、Date、Century。

15.4.18. RTC 开始工作控制寄存器(偏移：17H)

比特	名称	属性	复位值	描述
7:1	RSV	-	-	保留
0	RTC_START	RW	0x0	RTC 工作使能位 0: 不使能; 1: 使能;

15.4.19. RTC 中断标志清除寄存器(偏移：18H)

比特	名称	属性	复位值	描述
7:1	RSV	-	-	保留
1	CLR_ALARM_INT	RW	0x0	清 ALARM 中断位: W: 1: 清除闹钟中断; R: 0: 此时无闹钟中断; 1: 此时有闹钟中断;
0	CLR_TICK_INT	RW	0x0	清 TICK 中断位: W: 1: 清除 Tick 中断; R: 0: 此时有 Tick 中断; 1: 此时有 Tick 中断;

15.5. RTC 操作说明

- 1) 配置 RTC 系统时钟，工作时钟（配置 LP_CTRL 1 为外部 32.768K 或者内部 32K）；
- 2) 初始化 HOUR、MINUTE、SECOND、YEAR、MONTH、DATE、CENTURY 等寄存器；
- 3) 如果 RTC 工作时钟为内部 OSC_32K，则根据工作时钟频率，配置 SEC_CNT 寄存器；SEC_CNT 为工作时钟计数 1 S 时对应的计数值；
- 4) 根据场景配置 TICK 中断对应的寄存器 TICK_CTRL；
- 5) 根据场景配置 ALARM 中断对应的寄存器 ALARM_HOUR、ALARM_MINUTE、ALARM_SECOND、ALARM_YEAR、ALARM_MONTH、ALARM_DATE、ALARM_CTRL；
- 6) 配置 RTC_START 寄存器，启动 RTC 正常工作；

注：改变 HOUR、MINUTE、SECOND、YEAR、MONTH、DATE、CENTURY 等寄存器值，必须再次配置 RTC_RELOAD 和 RTC_START 寄存器；

16. 看门狗（WDT）

16.1. 概述

看门狗模块（WDT）采用 15 位的递增计数器，从加载值开始递增计数。当计数器从加载值计数到计数器溢出时，如果看门狗动作设为复位，则看门狗模块输出复位信号，复位系统；如果看门狗动作设为中断，则触发看门狗中断，如果在中断中未及时进行喂狗操作（重新设置加载值），则产生复位信号复位系统。用户可以通过设置看门狗使能位来停止/启动计数器。

16.2. WDT 主要特性

- 15 位的递增计数器
- 可选择的计数时钟周期
- 可编程装载值

16.3. 看门狗复位

看门狗复位功能时可配置寄存器 SCRSTEN 的第 1 比特位，当看门狗计数器计数到加载值溢出时，则会产生一个系统复位。

16.4. 看门狗中断

看门狗中断时可配置寄存器 IEN3 的第 1 比特位，使能看门狗中断，同时使能总中断。

16.5. 看门狗开始工作

看门狗工作使能可配置寄存器 IEN1 的第 6 比特位，使能看门狗定时器开始工作标志。

16.6. 看门狗计数器

看门狗计数器计数时钟有 24 或 3072 个时钟周期计数递增一次，可通过配置寄存器 WDTREL 的第 7 比特位选择。

看门狗计数器为 15 位计数器，可配置寄存器 WDTREL 的第 0 至第 6 比特位设置计数加载值的高 7 位，低 8 位默认为 0x00。

16.7. 看门狗寄存器

WDT 特殊功能寄存器

偏置	名称	描述
----	----	----

0x86	WDTREL	加载寄存器
------	--------	-------

16.7.1. WDTREL 加载寄存器(偏移: 86H)

比特	名称	属性	复位值	描述
7	DIV	WO	0x0	计数时钟周期选择位: 0: 看门狗计数器每 24 个时钟周期递增一次 1: 看门狗计数器每 3072 (128*24) 个时钟周期递增一次
6:0	LOAD	WO	0x00	7 位寄存器, 用来存入加载值的高 7 位。

注: LOAD 的最低有效值为 1, 配置完 LOAD 值后, 实际 15 位计数器计数值的高 7 位为 LOAD 值, 低 8 位值为 0x00。

假设 LOAD 的 7 位寄存器值对应的十进制数据为 n, 看门狗复位时间 (单位: 秒) 的计算公式如下:

- 当 DIV 配置为 0 时, 则看门狗复位时间为:

$$\frac{(128 - n) * 256 * 24}{16000000}$$

- 当 DIV 配置为 1 时, 则看门狗复位时间为:

$$\frac{(128 - n) * 256 * 3072}{16000000}$$

16.8. 看门狗操作说明

16.8.1. 看门狗定时器溢出产生中断

- 1) 配置加载寄存器;
- 2) 配置计数时钟周期选择位;
- 3) 配置使能看门狗时钟;
- 4) 配置中断使能;
- 5) 使能看门狗复位;
- 6) 使能看门狗定时器。

注: 如果没有使能中断, 则在计数到加载值溢出, 经过中断未在 2 个计数值内喂狗, 将产生一个系统复位。

16.8.2. 看门狗定时器溢出产生系统复位

- 1) 配置 SCCM0, 使能看门狗时钟;
- 2) 配置 SCRSTEN 使能看门狗复位芯片;
- 3) 配置使能看门狗中断 IEN3.WDT_EN;
- 4) 开启总中断
- 5) 配置加载寄存器 WDTREL, 设置计数时钟周期以及加载值;
- 6) 开启看门狗定时器 (IEN1.SWDT)。

注: 计数器计数到加载值溢出, 不喂狗将产生一个系统复位。

17.AES/SM4/DES 加解密

17.1. 概述

该设备内置 DES 协处理器，支持单 DES 和三重 DES 算法，实现数据的加解密。

该设备内置 SM4 协处理器，支持 SM4 算法，实现数据的加解密。

该设备内置 AES128 协处理器，支持 AES128 算法，实现数据的加解密。

17.2. 主要特性

- DES 符合 FIPS 46-3 的加解密标准
- 单 DES 支持 64bit 密钥，三重 DES 支持 192bit 密钥
- DES 支持 ECB(Electronic Code Book)模式
- AES 支持 128bit 密钥
- SM4 支持 128bit 密钥

17.3. 加解密算法寄存器

加解密算法寄存器基地址：0x2100

偏置	名称	描述
0x00	SM4_DES_CON	控制寄存器
0x01	SM4_DES_STS	状态寄存器
0x02	SM4_DES_MSK	中断控制寄存器
0x03	SM4_DES_DAT0_0	数据寄存器
0x04	SM4_DES_DAT0_1	数据寄存器
0x05	SM4_DES_DAT0_2	数据寄存器
0x06	SM4_DES_DAT0_3	数据寄存器
0x07	SM4_DES_DAT1_0	数据寄存器
0x08	SM4_DES_DAT1_1	数据寄存器
0x09	SM4_DES_DAT1_2	数据寄存器
0x0a	SM4_DES_DAT1_3	数据寄存器
0x0b	SM4_DES_DAT2_0	数据寄存器
0x0c	SM4_DES_DAT2_1	数据寄存器
0x0d	SM4_DES_DAT2_2	数据寄存器
0x0e	SM4_DES_DAT2_3	数据寄存器
0x0f	SM4_DES_DAT3_0	数据寄存器
0x10	SM4_DES_DAT3_1	数据寄存器
0x11	SM4_DES_DAT3_2	数据寄存器
0x12	SM4_DES_DAT3_3	数据寄存器

0x13	SM4_DES_KEY0_0	密钥寄存器
0x14	SM4_DES_KEY0_1	密钥寄存器
0x15	SM4_DES_KEY0_2	密钥寄存器
0x16	SM4_DES_KEY0_3	密钥寄存器
0x17	SM4_DES_KEY1_0	密钥寄存器
0x18	SM4_DES_KEY1_1	密钥寄存器
0x19	SM4_DES_KEY1_2	密钥寄存器
0x1a	SM4_DES_KEY1_3	密钥寄存器
0x1b	SM4_DES_KEY2_0	密钥寄存器
0x1c	SM4_DES_KEY2_1	密钥寄存器
0x1d	SM4_DES_KEY2_2	密钥寄存器
0x1e	SM4_DES_KEY2_3	密钥寄存器
0x1f	SM4_DES_KEY3_0	密钥寄存器
0x20	SM4_DES_KEY3_1	密钥寄存器
0x21	SM4_DES_KEY3_2	密钥寄存器
0x22	SM4_DES_KEY3_3	密钥寄存器
0x23	SM4_DES_KEY4_0	密钥寄存器
0x24	SM4_DES_KEY4_1	密钥寄存器
0x25	SM4_DES_KEY4_2	密钥寄存器
0x26	SM4_DES_KEY4_3	密钥寄存器
0x27	SM4_DES_KEY5_0	密钥寄存器
0x28	SM4_DES_KEY5_1	密钥寄存器
0x29	SM4_DES_KEY5_2	密钥寄存器
0x2a	SM4_DES_KEY5_3	密钥寄存器

17.3.1. SM4_DES_CON 控制寄存器(偏移: 00H)

比特	名称	属性	复位值	描述
7:4	保留			
3:2	mode	R/W	0x0	算法选择: 00: AES128 01: DES 10: 3DES 11: SM4
1	ende_reg	R/W	0x0	运算方式控制位 1: 加密 0: 解密
0	start	R/W	0x0	启动控制位 0: 停止 1: 启动。软件置 1, 加解密结束时硬件自动清零

17.3.2. SM4_DES_STS 状态寄存器(偏移: 01H)

比特	名称	属性	复位值	描述
7:1	保留			
0	sts	R/W	0x0	加解密完成标志 0: 未完成 1: 运算完成, 硬件置 1, 软件写清零。

17.3.3. SM4_DES_MSK 中断控制寄存器(偏移: 02H)

比特	名称	属性	复位值	描述
7:1	保留			
0	msk	R/W	0x0	屏蔽加解密中断控制位 1: 屏蔽 0: 不屏蔽

17.3.4. SM4_DES_DAT0_X 数据寄存器 32 位(偏移: 03H)

比特	名称	属性	复位值	描述
31:24	SM4_DES_DAT0_3	R/W	0x00	1、SM4/AES128/DES/3DES 明文输入数据寄存器 2、加密/解密输出的数据
23:16	SM4_DES_DAT0_2	R/W	0x00	
15:8	SM4_DES_DAT0_1	R/W	0x00	
7:0	SM4_DES_DAT0_0	R/W	0x00	

17.3.5. SM4_DES_DAT1_X 数据寄存器 32 位(偏移: 07H)

比特	名称	属性	复位值	描述
31:24	SM4_DES_DAT1_3	R/W	0x00	1、SM4/AES128/DES/3DES 明文输入数据寄存器 2、加密/解密输出的数据
23:16	SM4_DES_DAT1_2	R/W	0x00	
15:8	SM4_DES_DAT1_1	R/W	0x00	
7:0	SM4_DES_DAT1_0	R/W	0x00	

17.3.6. SM4_DES_DAT2_X 数据寄存器 32 位(偏移: 0BH)

比特	名称	属性	复位值	描述
31:24	SM4_DES_DAT0_3	R/W	0x00	1、SM4/AES128 明文输入数据寄存器 2、加密/解密输出的数据
23:16	SM4_DES_DAT0_2	R/W	0x00	
15:8	SM4_DES_DAT0_1	R/W	0x00	
7:0	SM4_DES_DAT0_0	R/W	0x00	

17.3.7. SM4_DES_DAT3_X 数据寄存器 32 位(偏移: 0FH)

比特	名称	属性	复位值	描述
31:24	SM4_DES_DAT0_3	R/W	0x00	1、SM4/AES128 明文输入数据寄存器 2、加密/解密输出的数据
23:16	SM4_DES_DAT0_2	R/W	0x00	
15:8	SM4_DES_DAT0_1	R/W	0x00	
7:0	SM4_DES_DAT0_0	R/W	0x00	

17.3.8. SM4_DES_KEY0_X 数据寄存器 32 位(偏移: 13H)

比特	名称	属性	复位值	描述
31:24	SM4_DES_KEY0_3	R/W	0x00	1、SM4/AES128/DES/3DES 密钥输入数据寄存器
23:16	SM4_DES_KEY0_2	R/W	0x00	
15:8	SM4_DES_KEY0_1	R/W	0x00	
7:0	SM4_DES_KEY0_0	R/W	0x00	

17.3.9. SM4_DES_KEY1_X 数据寄存器 32 位(偏移: 17H)

比特	名称	属性	复位值	描述
31:24	SM4_DES_KEY1_3	R/W	0x00	1、SM4/AES128/DES/3DES 密钥输入数据寄存器
23:16	SM4_DES_KEY1_2	R/W	0x00	
15:8	SM4_DES_KEY1_1	R/W	0x00	
7:0	SM4_DES_KEY1_0	R/W	0x00	

17.3.10. SM4_DES_KEY2_X 数据寄存器 32 位(偏移: 1BH)

比特	名称	属性	复位值	描述
31:24	SM4_DES_KEY2_3	R/W	0x00	1、SM4/AES128/3DES 密钥输入数据寄存器
23:16	SM4_DES_KEY2_2	R/W	0x00	
15:8	SM4_DES_KEY2_1	R/W	0x00	
7:0	SM4_DES_KEY2_0	R/W	0x00	

17.3.11. SM4_DES_KEY3_X 数据寄存器 32 位(偏移: 1FH)

比特	名称	属性	复位值	描述
31:24	SM4_DES_KEY3_3	R/W	0x00	1、SM4/AES128/3DES 密钥输入数据寄存器

23:16	SM4_DES_KEY3_2	R/W	0x00	
15:8	SM4_DES_KEY3_1	R/W	0x00	
7:0	SM4_DES_KEY3_0	R/W	0x00	

17.3.12. SM4_DES_KEY4_X 数据寄存器 32 位(偏移: 23H)

比特	名称	属性	复位值	描述
31:24	SM4_DES_KEY4_3	R/W	0x00	1、3DES 密钥输入数据寄存器
23:16	SM4_DES_KEY4_2	R/W	0x00	
15:8	SM4_DES_KEY4_1	R/W	0x00	
7:0	SM4_DES_KEY4_0	R/W	0x00	

17.3.13. SM4_DES_KEY5_X 数据寄存器 32 位(偏移: 27H)

比特	名称	属性	复位值	描述
31:24	SM4_DES_KEY5_3	R/W	0x00	1、3DES 密钥输入数据寄存器
23:16	SM4_DES_KEY5_2	R/W	0x00	
15:8	SM4_DES_KEY5_1	R/W	0x00	
7:0	SM4_DES_KEY5_0	R/W	0x00	

17.4. 加解密算法操作说明

1. 初始化流程

- 1) 配置 SCCM2 寄存器，使能加解密算法时钟
- 2) 使能加解密算法中断和总中断
- 3) 配置 SM4_DES_MSK 寄存器，选择是否屏蔽 DES 和 SM4 中断，

若采用中断方式，则不屏蔽中断，若采用查询方式则屏蔽中断且第 2 步可以跳过。

2. 算法加、解密流程

- 1) 写入待加密或解密的数据到寄存器 SM4_DES_DAT0_X 至 SM4_DES_DAT3_X (X=0,1,2,3。下同)
 - DES/TDES 对应 SM4_DES_DAT0_X ~ SM4_DES_DAT1_X;
 - SM4/AES 对应 SM4_DES_DAT0_X ~ SM4_DES_DAT3_X;
- 2) 写入相应的密钥数据到寄存器 SM4_DES_KEY0_X 至 SM4_DES_KEY5_X
 - DES 对应 SM4_DES_KEY0_X ~ SM4_DES_KEY1_X;

- TDES 对应 SM4_DES_KEY0_X ~ SM4_DES_KEY5_X;
 - SM4/AES 对应 SM4_DES_KEY0_X ~ SM4_DES_KEY3_X;
- 3) 写寄存器 SM4_DES_CON 的第 0,1,2,3 比特位配置加解密模式
 - 4) 查询 SM4_DES_STS 寄存器，等待加解密完成（若采用中断方式，则在加解密完成后会产生中断）。
 - 5) 查到 SM4_DES_STS 寄存器标志位（或进入中断）后，清除标志位。然后读数据寄存器 SM4_DES_DAT0_X 至 SM4_DES_DAT3_X
 - DES/TDES 对应 SM4_DES_DAT0_X~SM4_DES_DAT1_X;
 - SM4/AES 对应 SM4_DES_DAT0_X ~ SM4_DES_DAT3_X;

注：若要改变 SM4_DES_MSK 中断的值，建议在写入密钥数据之前执行

需加解数据和密钥的写入与读取，都是高字节数据在后，低字节数据在前，依次填入。例如：
0x11,0x12,0x13,0x14。

SM4_DES_DAT0_0=0x14, SM4_DES_DAT0_1=0x13,

SM4_DES_DAT0_2=0x12, SM4_DES_DAT0_3=0x11。

18. 乘除法器 (MDU)

18.1. 概述

该设备集成一个算术协处理器使高性能的 8051 执行额外的扩展算术运算。

18.2. 主要特性

- 支持 32 位除法
- 支持 16 位乘法
- 移位和标准化操作

18.3. MDU 计算操作描述

18.3.1. 16bit*16bit 乘法运算

将乘数写入寄存器 MD0, MD1 (高位在后, 低位在前, 下同), 将被乘数写入寄存器 MD4, MD5。等待计算出结果, 读取寄存器 MD0/MD1/MD2/MD3 (MD0 为低字节, MD3 为高字节)。

18.3.2. 16bit/16bit 除法运算

将被除数写入寄存器 MD0, MD1; 将除数写入寄存器 MD4, MD5。等待计算出结果, 商存储在寄存器 MD0 (低 8 位), MD1 (高 8 位); 余数存储在寄存器 MD4 (低 8 位)、MD5 (高 8 位)。

18.3.3. 32bit/16bit 除法运算

将被除数写入寄存器 MD0, MD1, MD2, MD3; 将除数写入寄存器 MD4, MD5。等待计算出结果, 商存储在寄存器 MD0 (低字节), MD1, MD2, MD3 (最高字节); 余数存储在寄存器 MD4 (低 8 位)、MD5 (高 8 位)。

18.3.4. 移位操作运算

将需要移位的数据写入寄存器 MD0, MD1, MD2, MD3 (高位在后, 低位在前); 配置 ARCON 寄存器的第 5 比特位选择左移或者右移; 配置 ARCON 的 0,1,2,3,4 比特位确定移动位数。等待计算结果, 读取寄存器 MD0 (最低字节), MD1, MD2, MD3 (最高字节)。

例: 16bit*16bit 以 $0x1234*0x5678=0x6260060$ 为例

乘数写入寄存器 MD0=0x34, MD1=0x12; 被乘数写入寄存器 MD4=0x78, MD5=0x56
等待结果, 读取寄存器 MD0=0x60, MD1=0x00, MD2=0x26, MD3=0x06。

18.4. MDU 寄存器

寄存器	地址	描述	复位值
MD0	E9H	乘法/除法数据寄存器	00H
MD1	EAH		00H
MD2	EBH		00H
MD3	ECH		00H
MD4	EDH		00H
MD5	EEH		00H
ARCON	EFH	运算控制寄存器	00H

18.4.1. 运算控制寄存器(ARCON)

比特	名称	属性	功能
7	MDEF	R/W	MDU 错误标志
6	MDOV	R/W	MDU 溢出标志
5	SLR	R/W	位移方向 0: 左移操作。 1: 右移操作。
4	SC.4	R/W	移位计数器 When set to all '0's, normalize operation is selected. After normalization, the "sc.0" ... "sc.4" contain the number of normalizing shifts performed. When at least one of these bit is set high shift operation is selected. The number of shifts performed is determined by the number written to "sc.4" ..., "sc.0", where "sc.4" is the MSB.
3	SC.3	R/W	
2	SC.2	R/W	
1	SC.1	R/W	
0	SC.0	R/W	

18.5. MDU 计算操作说明

18.5.1. 16bit*16bit 乘法运算

- 1) 写入乘数 MD0（低 8 位）和 MD1（高 8 位）；
- 2) 写入被乘数 MD4（低 8 位）和 MD5（高 8 位）；
- 3) 软件延时约 20us（系统 16M 时钟下）；
- 4) 读出计算结果，即读取：MD0（最低字节）、MD1、MD2 和 MD3（最高字节）；

18.5.2. 16bit/16bit 除法运算

- 1) 写入除数 MD0（低 8 位）和 MD1（高 8 位）；
- 2) 写入被除数 MD4（低 8 位）和 MD5（高 8 位）；
- 3) 软件延时约 20us（系统 16M 时钟下）；
- 4) 读出计算结果，即读取商：MD0（低 8 位），MD1（高 8 位）
余数：MD4（低 8 位），MD5（高 8 位）

18.5.3. 32bit/16bit 除法运算

- 1) 写入除数 MD0（最低字节）、MD1、MD2 和 MD3（最高字节）；
- 2) 写入被除数 MD4（低 8 位）和 MD5（高 8 位）；
- 3) 软件延时约 100us（系统 16M 时钟下）；
- 4) 读出计算结果，即读取商：MD0（最低字节）、MD1、MD2 和 MD3（最高字节）
余数：MD4（低 8 位），MD5（高 8 位）

18.5.4. 32bit/16bit 除法运算

- 1) 写入移位操作数 MD0（最低字节）、MD1、MD2 和 MD3（最高字节）；
- 2) 配置 ARCON 寄存器，第五位 SLR，选择左移或右移操作；
- 3) 配置 ARCON 寄存器，SC0~SC4，选择移动位数；
- 4) 软件延时约 100us（系统 16M 时钟下）；
- 5) 读出计算结果，即读取：MD0（最低字节）、MD1、MD2 和 MD3（最高字节）

19. 随机数（RNG）

19.1. 概述

芯片内嵌 1 个 8 位随机数发生器，用以满足某些应用中的安全交易流程。

19.2. 随机数寄存器

RNG 基址：0x2068

寄存器	偏移地址	描述	复位值
DTRNGCON	00H	RNG 控制寄存器	0x00
DTRNGSTS	01H	RNG 状态寄存器	0x00
DTRNGDAT	02H	RNG 数据寄存器	0x00

19.2.1. RNG 控制寄存器（DTRNGCON）

比特	名称	属性	功能
7:2	-	-	保留
1	model_sel	R/W	0: 选择 RNG 工作于 model0; 1: 选择 RNG 工作于 model1;
0	trng_start	R/W	1: 关闭 RNG; 0: 启动 RNG;

19.2.2. RNG 数据寄存器（DTRNGDAT）

比特	名称	属性	功能
7:0	trng_data	R/W	数据寄存器

19.2.3. RNG 状态寄存器（DTRNGSTS）

比特	名称	属性	功能
7:1	-	-	保留
0	trng_sts	R/W	0: 随机数生产未开始或未完成 1: 随机数生成完成

19.3. 随机数操作流程

- 1) 配置 RNG 时钟（SCCM1.RNG_EN）使能；
- 2) 配置 RNG 模式（DTRNGCON.model_sel）选择；
- 3) 开启 RNG 模块（DTRNGCON.trng_start）；
- 4) 判断完成标志（DTRNGSTS.trng_sts），并读取数据（DTRNGDAT）

20.CRC8 计算单元

20.1. 概述

CRC 即循环冗余校验码 (Cyclic Redundancy Check)：是根据固定的生成多项式得到任一 8 位的 CRC 计算结果。

在其他的应用中，CRC 技术主要应用于核实数据传输或者数据存储的正确性和完整性。标准 EN/IEC 60335-1 即提供了一种核实闪存存储器完整性的方法。CRC 计算单元可以在程序运行时计算出软件的标识，之后与在连接时生成的参考标识比较，然后存放在指定的存储器空间。

20.2. 主要特性

- 使用 CRC-8 多项式：0x07
 - x^8+x^2+x+1
- 一个 8 位数据寄存器用于输入/输出
- CRC 计算时间：X 个时钟周期

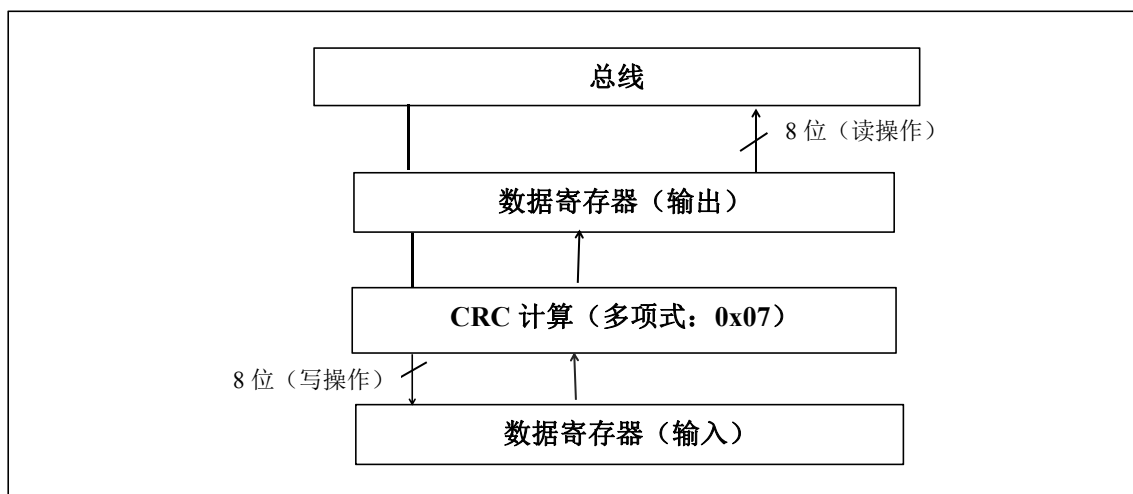


图 20-1 CRC 计算单元框图

20.3. CRC8 功能描述

CRC 计算单元含有 1 个 8 位数据寄存器：

- 对该寄存器进行写操作时，作为输入寄存器，可以输入要进行 CRC 计算的新数据。
- 对该寄存器进行读操作时，返回上一次 CRC 计算的结果。

20.4. CRC8 计算单元寄存器

CRC 计算单元包括 2 个数据寄存器和 1 个控制寄存器。（基址：0x2000）

偏置	名称	描述
0x00	CRC8_DATA	CRC8 数据寄存器
0x01	CRC8_CTRL	CRC8 控制寄存器
0x02	CRC8_INIT	CRC8 初始值配置寄存器

20.4.1. 数据寄存器 CRC8_DATA (偏移: 00H)

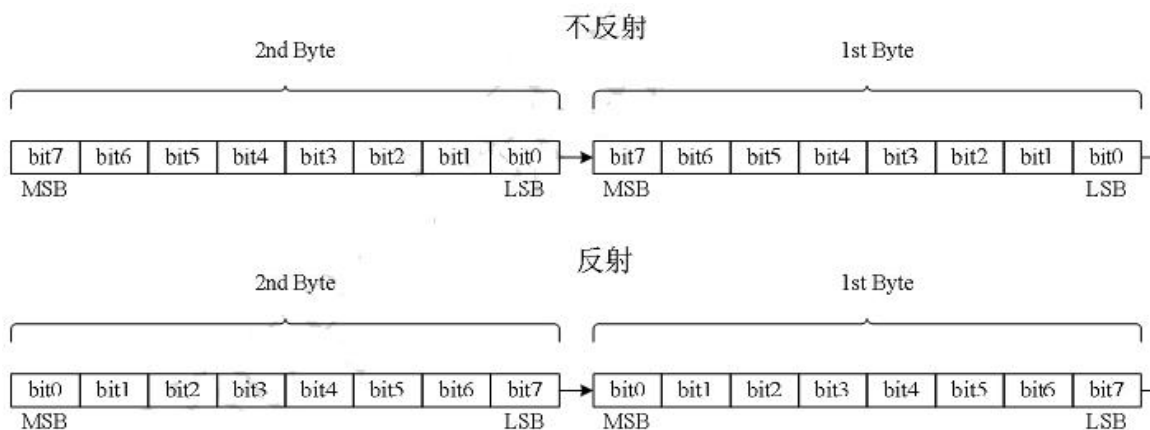
比特	名称	属性	复位值	描述
7:0	DATA	RW	0xff	写: 写入需要进行 CRC 校验计算的数据, 如需要校验的数据不止一个 byte 需按顺序逐次写入 读: 读出 8 位 CRC 计算结果

注: 读取完计算结果后, 如需再次计算 CRC, 必须重新初始化 CRC8_INIT

20.4.2. 控制寄存器 CRC8_CTRL (偏移: 01H)

比特	名称	属性	复位值	描述
7:2	RSV	-	-	保留
1	RSLT_REV	RW	0x0	CRC 计算结果是否进行高低位倒序 0: 不倒序 1: 倒序
0	DATA_REV	RW	0x0	CRC 计算数据是否进行高低位倒序 0: 不倒序 1: 倒序

注: 输入数据与输出结果的反转说明如下



20.4.3. 初始值寄存器 CRC8_INIT (偏移: 02H)

比特	名称	属性	复位值	描述
7:0	CRC_INIT	RW	0xff	8 位 CRC 初始值

20.5. CRC8 计算单元操作说明

- 1) 设置 8 位初始值 CRC8_INIT。
- 2) 设置 CRC8_CTRL，选择计算数据和结算结果是否倒序。
- 3) 向 CRC8_DATA 中写入 8 位 CRC 计算数据，如需要校验的数据不止一个 byte 需按顺序逐次写入，直至输入完成。
- 4) 读 CRC8_DATA，将一次返回 CRC 计算结果。
- 5) 若需重新进行一组数据的 CRC8 计算，需从步骤 1) 开始，而不是直接向 CRC8_DATA 中填数据。

21.I2C 接口

21.1. 概述

I2C 总线接口连接微控制器和串行 I2C 总线。I2C 模块接收和发送数据，并将数据从串行转换成并行，或并行转换成串行。I2C 模块通过数据引脚 SDA 和时钟引脚 SCL 连接到 I2C 总线，控制所有 I2C 总线规定的时序。芯片有两路 I2C 模块，每路 I2C 模块都支持主机模式和从机模式。I2C0 的主/从机接口分布在不同的两组 GPIO 上，I2C1 的主/从机接口分布在同一组 GPIO 上。

21.2. 主要特性

- I2C 主设备功能
 - 产生时钟
 - 产生起始和停止信号
- I2C 从设备功能
 - 可编程的 I2C 从设备地址
 - 停止位检测
- 支持 100Kbps、400Kbps 两种速率模式
- 产生和检测 7bit 设备地址
- 状态标志
 - 总线忙标志
 - 字节发送结束完成标志

21.3. I2C 功能描述

21.3.1. I2C 通信过程

主模式时，I2C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。起始条件和停止条件都是在主模式下由软件控制产生。

从模式时，I2C 接口能识别它自己的地址(7 位)。

数据和地址按 8 位/字节进行传输，高位在前。跟在起始条件后的 1 字节是地址。地址只在主模式发送。

在一个字节传输的 8 个时钟后的第 9 个时钟期间，接收器必须回送一个应答位(ACK)给发送器。参考下图。

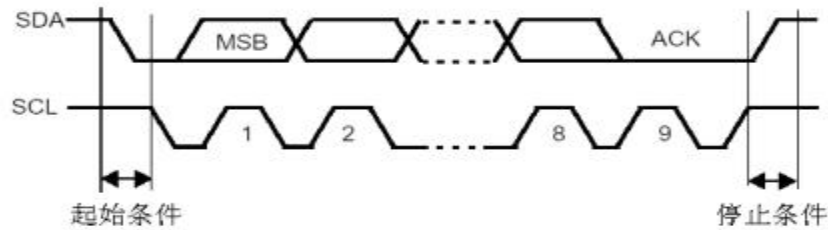


图 21-1 IIC 总线协议时序

注：I2C 的 SDA/SCL 通过 IO 复用来配置为 I2C 功能。

21.4. I2C 寄存器

I2C0 寄存器基地址：0x2000

偏置	名称	描述
0xC0	I2C0_M_PERER_L	I2C0 主机波特率寄存器低位
0xC1	I2C0_M_PERER_H	I2C0 主机波特率寄存器高位
0xC2	I2C0_M_CTR	I2C0 主机控制寄存器
0xC3	I2C0_M_TXR	I2C0 主机发送数据寄存器
0xC4	I2C0_M_CR	I2C0 主机发送控制寄存器
0xC5	I2C0_M_RXR	I2C0 主机接收数据寄存器
0xC6	I2C0_M_STS	I2C0 主机状态寄存器
0xC7	I2C0_M_RST	I2C0 软复位寄存器
0xC8	I2C0_S_DATA	I2C0 从机数据寄存器
0xC9	I2C0_S_STS	I2C0 从机状态寄存器
0xCB	I2C0_S_ADDR	I2C0 从机设备地址寄存器
0xCC	I2C0_S_CON	I2C0 中断屏蔽寄存器

注：I2C0_M_***中的 M 代表 Master（主机），I2C0_S_***中的 S 代表 Slave（从机），下同。

I2C1 寄存器基地址：0x2000

偏置	名称	描述
0x50	I2C1_M_PERER_L	I2C1 主机波特率寄存器低位
0x51	I2C1_M_PERER_H	I2C1 主机波特率寄存器高位
0x52	I2C1_M_CTR	I2C1 主机控制寄存器
0x53	I2C1_M_TXR	I2C1 主机发送数据寄存器
0x54	I2C1_M_CR	I2C1 主机发送控制寄存器
0x55	I2C1_M_RXR	I2C1 主机接收数据寄存器
0x56	I2C1_M_STS	I2C1 主机状态寄存器
0x57	I2C1_M_RST	I2C1 软复位寄存器
0x58	I2C1_S_DATA	I2C1 从机数据寄存器
0x59	I2C1_S_STS	I2C1 从机状态寄存器
0x5B	I2C1_S_ADDR	I2C1 从机设备地址寄存器
0x5C	I2C1_S_CON	I2C1 中断屏蔽寄存器

21.4.1. I2Cx 主机波特率寄存器低位 I2Cx_M_PERER1 (偏移: C0H、50H) (此处 x 代表 0,1, 下同)

比特	名称	属性	复位值	描述
7:0	I2Cx_M_PERER_L	R/W	0xFF	SCL 时钟计数器 M_PERER 低 8 位

21.4.2. I2Cx 主机波特率寄存器高位 I2Cx_M_PERER2 (偏移: C1H、51H)

比特	名称	属性	复位值	描述
7:0	I2Cx_M_PERER_H	R/W	0xFF	SCL 时钟计数器 M_PERER 高 8 位

注: $SCL \text{ 时钟} = (\text{系统时钟}/5) / M_PERER$ 。

例: 16M 时钟下: I2Cx_M_PERER_H=0x00; I2Cx_M_PERER_L=0x1C 则速率为 100k。

21.4.3. I2Cx 主机控制寄存器 I2Cx_M_CTR(偏移: C2H、52H)

比特	名称	属性	复位值	描述
7	CORE_EN	R/W	0x0	使能主机模块 0: 不使能 1: 使能
6	INT_MASK	R/W	0x0	中断屏蔽 0: 不屏蔽 1: 屏蔽
5:0	RSV	-	-	保留

21.4.4. I2Cx 主机发送数据寄存器 I2Cx_M_TXR(偏移: C3H、53H)

比特	名称	属性	复位值	描述
7:0	TXR_BUF	RW	0x00	发送数据寄存器

21.4.5. I2Cx 主机发送控制寄存器 I2Cx_M_CR (偏移: C4H、54H)

比特	名称	属性	复位值	描述
7	START	RW	0x0	Start 起始条件产生位 0: 无影响 1: 发送 start 条件, 开始发送
6	STOP	RW	0x0	Stop 停止条件产生位 0: 无影响 1: 发送 stop 条件, 结束
5	RD	RW	0x0	读数据选择位 0: 无影响 1: 主机为接收数据状态
4	WR	RW	0x0	写数据选择位 0: 无影响 1: 主机为发送数据状态

3	ACK	RO	0x0	主机发送 ACK/NACK 选择 0: 发送完 1 byte 后回 ACK 1: 发送完 1 byte 后回 NACK
2: 1	RSV	-	-	保留
0	IACK	RW	0x0	主机 1byte 数据传输完状态控制位 0: 无影响 1: 清除完成状态位 写 1 清除 I2Cx_M_STS 的 Irq_flag 状态位

21.4.6. I2Cx 主机接收数据寄存器 I2Cx_M_RXR (偏移: C5H、55H)

比特	名称	属性	复位值	描述
7:0	RXR_BUF	RW	0	接收数据寄存器

21.4.7. I2Cx 主机状态寄存器 I2Cx_M_STS (偏移: C6H、56H)

比特	名称	属性	复位值	描述
7	Rxack	RO	0x0	从机应答状态位 0: 从机返回 ACK 1: 从机返回 NACK
6	I2c_busy	RO	0x0	总线忙状态位 0: 总线空闲 1: 总线忙
5	Al	RO	0x0	总线丢失判断位 0: 总线正常 1: 总线丢失
4:2	RSV	-	-	保留
1	Tip	RO	0x0	传输状态位 0: 空闲 1: 位正在传输
0	Irq_flag	RO	0x0	主机 1byte 数据传输完成状态位 0: 主机 1byte 传输未完成 1: 主机 1byte 传输完成 写 I2Cx_M_CR 寄存器 Iack 位为 1 清除该状态

21.4.8. I2Cx 主机软复位寄存器 I2Cx_M_RST (偏移: C7H、57H)

比特	名称	属性	复位值	描述
7:1	RSV	-	-	保留
0	Wb_rst_i	WO	0x0	软复位控制位 0: 无影响 1: 软复位

21.4.9. I2Cx 从机数据寄存器 I2Cx_S_DATA (偏移: C8H、58H)

比特	名称	属性	复位值	描述
7:0	Tx_fifo_data	RW	0x00	I2C数据寄存器

21.4.10. I2Cx 从机状态寄存器 I2Cx_S_STS (偏移: C9H、59H)

比特	名称	属性	复位值	描述
7: 4	RSV	-	-	保留
3	STOP_FLAG	RW	0x0	停止信号状态位 0: 未检测到总线停止状态 1: 检测到总线停止状态 硬件置 1, 软件清 0
2	WR_FLAG	RW	0x0	发送数据状态 0: 发送数据寄存器有数据 1: 发送数据寄存器为空 硬件置 1, 软件清 0
1	RD_FLAG	RW	0x0	接收数据状态 0: 接收数据寄存器为空 1: 接收到数据 硬件置 1, 软件清 0
0	START_FLAG	RW	0x0	开始信号检测位 0: 未检测到 start 条件 1: 检测到 start 条件 硬件置 1, 软件清 0

21.4.11. I2Cx 从机设备地址寄存器 I2Cx_S_ADDR (偏移: CBH、5BH)

比特	名称	属性	复位值	描述
7:0	I2C_S_ADDR	RW	0x00	从机设备地址

21.4.12. I2Cx 从机中断屏蔽寄存器 I2Cx_S_CON (偏移: CCH、5CH)

比特	名称	属性	复位值	描述
7:4	RSV	-	-	保留
3	STOP_MASK	RW	0x0	停止条件中断屏蔽位 0:不屏蔽中断 1:屏蔽中断
2	RD_MASK	RW	0x0	接收数据中断屏蔽位 0:不屏蔽中断 1:屏蔽中断
1	WR_MASK	RW	0x0	发送数据中断屏蔽位 0:不屏蔽中断 1:屏蔽中断

0	START_MASK	RW	0x0	开始条件中断屏蔽位 0:不屏蔽中断 1:屏蔽中断
---	------------	----	-----	--------------------------------

21.5. I2C 操作说明

21.5.1. 主机初始化流程

- 1) 配置相关引脚为 I2Cx 主机功能，使能 I2Cx 时钟。
- 2) 写 I2Cx_M_PERER1, I2Cx_M_PERER2 寄存器，设置主机速率；

21.5.2. 主机发送流程

- 1) 读 I2Cx_M_STS 寄存器，忙检测，等待总线空闲；
- 2) 配置 I2Cx_M_CTR 寄存器，使能主机模块，配置中断屏蔽；
- 3) 配置 I2Cx_M_CR 寄存器为 0x90，发送 start 信号；
- 4) 往 I2Cx_M_TXR 寄存器写入要发送的数据；
- 5) 配置 I2Cx_M_CR 寄存器 WR 位，设置为写操作，主机将写入的数据发送给从机；
- 6) 等待读到 I2Cx_M_STS 寄存器的 IRQ_FLAG 位为 1，数据传输结束；
- 7) 写 I2Cx_M_CR 的 IACK 位，清完成标志；
- 8) 重复步骤 4~7，分别写从机地址（7bit 从机地址和 1bit 写信号）、操作地址、数据长度、对应长度的数据；
- 9) 完成后写 I2Cx_M_CR 的 STOP 位发送停止条件结束当前传输；
- 10) 写 I2Cx_M_RST，清状态。

21.5.3. 主机接收流程

- 1) 读 I2Cx_M_STS 寄存器，忙检测，等待总线空闲；
- 2) 配置 I2Cx_M_CTR 寄存器，使能主机模块，配置中断屏蔽；
/*给从机发送从机地址、需要操作的地址、需要读取的数据长度*/
- 3) 配置 I2Cx_M_CR 寄存器为 0x90，发送 start 信号；
- 4) 往 I2Cx_M_TXR 寄存器写入发送数据；
- 5) 配置 I2Cx_M_CR 寄存器 WR 位，设置为写操作，主机将写入的数据发送给从机；
- 6) 等待读到 I2Cx_M_STS 寄存器的 IRQ_FLAG 位为 1，数据传输结束；
- 7) 写 I2Cx_M_CR 的 IACK 位，清完成标志；
- 8) 重复步骤 4~7，分别写从机地址（7bit 从机地址+1bit 写信号）、操作地址、数据长度；

/*给从机发送从机地址和读信号*/

- 9) 配置 I2Cx_M_CR 寄存器为 0x90，再次发送 start 信号；
 - 10) 配置 I2Cx_M_TXR 寄存器，从机地址（7bit 从机地址+1bit 读信号）；
 - 11) 配置 I2Cx_M_CR 寄存器 WR 位，设置为写操作，主机将写入的地址发送给从机；
 - 12) 等待读到 I2Cx_M_STS 寄存器的 IRQ_FLAG 位为 1，数据传输结束；
 - 13) 写 I2Cx_M_CR 的 IACK 位，清完成标志；
- /*读数据*/
- 14) 配置 I2Cx_M_CR 寄存器的 RD 位为 1，设置为读操作；
 - 15) 等待读到 I2Cx_M_STS 寄存器的 IRQ_FLAG 位为 1，1byte 数据传输结束；
 - 16) 读 I2Cx_M_RXR 寄存器，接收数据；
 - 17) 写 I2Cx_M_CR，清完成标志；
 - 18) 重复 14~17，完成整帧数据读取；
 - 19) 在读取最后一个数据之前，配置 I2Cx_M_CR 寄存器 ACK 为 1，不返回从机 ACK；之后重复 14~17。
 - 20) 完成后写 I2Cx_M_CR 的 STOP 位发送停止条件结束当前传输；
 - 21) 写 I2Cx_M_RST，清状态。

注意：若不需要给从机发送操作地址和操作长度，可跳过 3~8 步。

21.5.4. 从机初始化流程

- 1) 配置相关引脚为 I2Cx 从机功能，使能 I2Cx 时钟；
- 2) 写寄存器 I2Cx_S_ADDR，配置从机地址；
- 3) 写 I2C1_S_CON 寄存器为 0x01，配置停止条件中断、接收数据中断、发送数据中断不屏蔽，开始条件中断屏蔽。
- 4) 使能 I2Cx 中断和总中断。

若使用查询方式可跳过 3~4 步。

21.5.5. 从机发送流程

- 1) 等待 I2Cx_S_STS 的 START_FLAG 位为 1，并写 I2Cx_S_STS 的 START 位为 0 清除状态；
- 2) 等待 I2Cx_S_STS 的 RD_FLAG 位为 1,接收数据，写 I2Cx_S_STS 的 RD_FLAG 位为 0，清除状态
- 3) 重复 2 收取目标地址和长度字节；
- 4) 将待发送的数据写入 I2Cx_S_DATA 寄存器中；
- 5) 等待 I2Cx_S_STS 的 WR_FLAG 位为 1，写 I2Cx_S_STS 的 WR_FLAG 位为 0，清除状态
- 6) 重复 4-5 直到数据全部发送结束；

21.5.6. 从机接收流程

- 1) 等待 I2Cx_S_STS 的 START_FLAG 位为 1，并写 I2Cx_S_STS 的 START 位为 0 清除状态；
- 2) 等待 I2Cx_S_STS 的 RD_FLAG 位为 1,接收数据，写 I2Cx_S_STS 的 RD_FLAG 位为 0，清除状态
- 3) 重复 2 收取目标地址和长度字节；
- 4) 等待 I2Cx_S_STS 的 START_FLAG 位为 1，并写 I2Cx_S_STS 的 START 位为 0 清除状态；
- 5) 等待 I2Cx_S_STS 的 RD_FLAG 位为 1,写 I2Cx_S_STS 的 RD_FLAG 位为 0，清除状态
- 6) 读取 I2Cx_S_DATA 寄存器数据
- 7) 重复 5-6 直到数据全部发送结束；

22.UART

22.1. 概述

UART 是一种通用串行数据总线，用于异步通信。该总线双向通信。可以实现全双工传输和接收；UART 接口 IP 可以工作在查询或中断方式下。

该设备包含 4 路 UART 接口，即 UART0、UART1、UART2 和 UART3。

22.2. 主要特性

- 16bit 的波特率分频因子寄存器
- 两种工作模式：模式 1、模式 3
- 可编程数据位宽(8/9bit)
- 模式 3 下多机通信
- 模式 3 下校验控制
- 2 个中断向量，接收中断和发送中断

22.3. UART 功能描述

接口通过两个或三个引脚与其他设备连接在一起,任何 UART 双向通信至少需要两个脚：接收数据输入(UART_RX)和发送数据输出(UART_TX)。

UART 模式下串行数据的发送接收帧结构组成如下：

- 总线在发送或接收前应处于空闲状态
- 一个起始位
- 一个数据字(8 或 9 位)，最低有效位在前
- 一个的停止位，由此表明数据帧的结束

UART 有两种工作模式：模式 1 和模式 3。波特率可由内部波特率发生器产生。

工作模式如下表所示：

SM0	SM1	工作方式	功能描述	波特率
0	1	模式 1	8 位 UART	可变
1	1	模式 3	9 位 UART	可变

波特率计算公式如下：

$$\text{baud rate} = \frac{F_{clk}}{(SxRELH * 256 + SxRELL)}$$

注：SMOD(PxCON.7) – 串口 0 波特率倍增位。

模式 1：

在模式 1 下，10 位的发送或接收。

发送时：一个起始位(总是 0)，8 个数据位(首先是 LSB)和 1 个停止位(总是 1)。

接收时：起始位用来同步数据，8 个数据位可通过读 SxBUF 获得，停止位表明通信结束。若多机通信使能位 SxCON 的 SM20=0，停止位会设置 SxCON 的 RB80 标志位。

模式 3:

在模式 3 下，11 位的发送或接收。一个起始位(总是 0)，8 个数据位(首先是 LSB)，一个可编程的第 9 位和 1 个停止位(总是 1)，第 9 位用来控制奇偶性。

发送时：SxCON 的 TB80 位作为第 9 位输出。

接收时：第 9 位影响 SxCON 的 RB80 位

多机通信:

在模式 3 下可以实现多机通信的功能。通过 SxCON 的 SM20 位来使能。接收中断只有在接收到的第 9 位数据为 1 时才能产生，否则，不会产生接收中断。

为了实现多机通信，所有的从机需要把 SxCON 的 SM20 置 1，主处理器发送一个地址字节来识别目标从处理器。地址字节不同于数据字节，地址字节的第 9 位是 1，而数据字节的第 9 位是 0。当 SxCON 的 SM20 = 1 时，数据字节不会导致从处理器中断，而地址字节则会中断所有从处理器。从机将接收到的地址和自身地址进行软件匹配，被寻址的从处理器会清 SxCON 的 SM20 位，并准备接收后续的数据。没有被寻址的从处理器保留其 SxCON 的 SM20 的设置，并忽略即将传入的数据。

22.4. UART 寄存器

UART0/1 特殊功能寄存器

偏置	名称	描述
0x98	S0CON	UART0 控制状态寄存器
0x99	S0BUF	UART0 数据缓冲寄存器
0xAA	S0RELL	UART0 重装寄存器低字节
0xBA	S0RELH	UART0 重装寄存器高字节
0x87	PCON	UART0 特殊功能寄存器
0x9B	S1CON	UART1 控制状态寄存器
0x9C	S1BUF	UART1 数据缓冲寄存器
0x9D	S1RELL	UART1 重装寄存器低字节
0xBB	S1RELH	UART1 重装寄存器高字节
0x8F	P1CON	UART1 特殊功能寄存器

UART2/3 寄存器基地址：0x2160

偏置	名称	描述
0x00	S2CON	UART2 控制状态寄存器
0x01	S2RELL	UART2 重装寄存器低字节
0x02	S2RELH	UART2 重装寄存器高字节

0x04	P2CON	UART2 特殊功能寄存器
0x05	S2BUF	UART2 数据缓冲寄存器
0x08	S3CON	UART3 控制状态寄存器
0x09	S3RELL	UART3 重装寄存器低字节
0x0A	S3RELH	UART3 重装寄存器高字节
0x0C	P3CON	UART3 特殊功能寄存器
0x0D	S3BUF	UART3 数据缓冲寄存器

22.4.1. 控制状态寄存器 SxCON (此处 x 代表 0,1,2,3, 下同)

比特	名称	属性	复位值	描述
7	SM0	RW	0x0	模式选择 00: 保留 01: 模式 1 10: 保留 11: 模式 3
6	SM1	RW	0x0	
5	SM20	RW	0x0	多机通信使能位 0: 禁止 1: 使能
4	REN0	RW	0x0	串行接收使能位 0: 禁止 1: 使能
3	TB80	RW	0x0	发送的第 9 位数据 该位只在模式 3 中发送数据时使用(奇偶校验或多机通信), 由软件控制
2	RB80	RW	0x0	接收的第 9 位数据 在模式 1 中, 如果 SM20 是 0, 该位为停止位 在模式 3 中, 该位为接收的第 9 位数据
1	TI0	RW	0x0	发送完成标志位 0: 未完成 1: 完成 硬件置 1, 软件清 0
0	RI0	RW	0x0	接收完成标志位 0: 未完成 1: 完成 硬件置 1, 软件清 0

22.4.2. 数据缓冲寄存器 SxBUF

比特	名称	属性	复位值	描述
7:0	SxBUF	RW	0x00	数据缓冲

22.4.3. 重装寄存器低字节 SxRELL

比特	名称	属性	复位值	描述
7:0	SxRELL	RW	0x00	重载值低 8 位

22.4.4. 重装寄存器高字节 SxRELH

比特	名称	属性	复位值	描述
7:4	RSV	-	-	保留
3:0	SxRELH	RW	0x00	重载值高 4 位

22.4.5. 特殊功能寄存器 PxCON

比特	名称	属性	复位值	描述
7	SMOD	RW	0x0	串口波特率选择 0: 不倍增 1: 倍增
6	WDT_TM	RW	0x0	看门狗定时器测试模式标志位(只存在于 PCON 寄存器)
5	ISR_TM	RW	0x0	中断服务例程测试模式标志位(只存在于 PCON 寄存器)
4	RSV	-	-	保留
3	RSV	-	-	保留
2	GF0	RW	0x0	通用标志位(只存在于 PCON 寄存器)
1	RSV	-	-	保留
0	RSV	-	-	保留

22.5. UART 操作说明

22.5.1. 串口的发送

- 1) 设置模式、波特率;
- 2) 写数据到缓冲寄存器 SxBUF, 写入 1Byte 数据待发送;
- 3) 查询 SxCON 的 TI0 位, 为“1”时发送完成;
- 4) 写 SxCON 的 TI0 位为“0”, 软件清零。

22.5.2. 串口的接收

- 1) 设置模式、波特率、接收使能位;
- 2) 查询 SxCON 的 RI0 位为“1”时, 接收完成;
- 3) 读 SxBUF;
- 4) 写 SxCON 的 RI0 位为“0”, 软件清零。

22.5.3. 多机通信

做为主处理器时:

- 1) 设置模式 3、波特率、将 SxCON 的 SM20 位置 1;

- 2) 写从机地址到缓冲器区寄存器 SxBUF;
- 3) 查询 SxCON 的 TI0 位, 为“1”时发送完成;
- 4) 写 SxCON 的 TI0 位为“0”, 软件清零
- 5) 将 SxCON 的 SM20 位清 0;
- 6) 写发送数据到缓冲器区寄存器 SXBUF, 保持 SxCON 的 TB80 位为 0;
- 7) 重复 3~6 到所有数据发送完成。

做为从处理器时:

- 1) 设置模式 3、波特率、将 SxCON 的 SM20 置 1, 接收使能位;
- 2) 查询 SxCON 的 RI0 位, 为“1”时接收完成;
- 3) 读 SxBUF, 软件匹配从机地址;
- 4) 写 SxCON 的 RI0 位为“0”, 软件清零;
- 5) 若从机地址匹配, 将 SxCON 的 SM20 位清 0, 主、从机正常通信; 若从机地址不匹配, 则 SxCON 的 SM20 位仍为 1, 忽略主机发送的数据。

23. 串行外设接口（SPI）

23.1. 概述

SPI 接口模块用于微控制器（MCU）与满足 SPI 外设之间进行全双工、全同步、串行通讯；SPI 接口 IP 可以工作在查询或中断方式下。

该设备包含 4 个 SPI 接口模块，即 SPI0、SPI1、SPI2 和 SPI3。

23.2. 主要特性

- 支持主/从 SPI 模式
- 可配置的传输速率，支持主时钟的 2、4、8、16、32、64、128 分频
- 支持模式 0/1/2 /3 四种传输协议
- 支持 SPI 四线传输
- 支持 8 位的串行传输，高位先出
- 从机仅支持模式 0

23.3. SPI 功能描述

23.3.1. SPI 的 4 个管脚描述

MISO：主设备输入/从设备输出管脚。

MOSI：主设备输出/从设备输入管脚。

SCK：串口时钟，作为主设备的输出，从设备的输入。

NSS：从设备选择。这是一个可选的管脚。它的功能是用来作为“片选管脚”，让主设备可以单独的与特定的从设备通信，避免数据线上的冲突。

注：SPI 的 MISO/MOSI/SCK 通过 IO 复用来配置为 SPI 功能。

23.3.2. SPI 的时钟信号的相位和极性

使用 CPOL 和 CPHA 位，能够组合成四种可能的时序关系。

CPOL(时钟极性)位控制在没有数据传输时时钟的空闲状态电平，此位对主模式和从模式下的设备都有效。如果 CPOL 被清 ‘0’，SCK 引脚在空闲状态保持低电平；如果 CPOL 被置 ‘1’，SCK 引脚在空闲状态保持高电平。

注：确保 SPI 引脚置成 SPI 空闲状态时的电平，以免在使能或者禁止 SPI 模块的时候在 SPI 时钟引脚上产生一个边沿。

如果 CPHA(时钟相位)被置 ‘1’，SCK 时钟的第二个边沿(CPOL 位为 0 时就是下降沿，CPOL 位为 1 时就是上升沿)进行数据位采样，数据在第一个时钟传输周期被锁存。如果 CPHA(时钟相位)被清 ‘0’，SCK 时钟的第一个边沿(CPOL 位为 0 时就是下降沿，

CPOL 位为 1 时就是上升沿)进行数据位采样，数据在第二个时钟传输周期被锁存。

图 24-1 显示了 SPI 的 4 种 CPHA 和 CPOL 位选择不同的组合主设备与从设备的 SCK、MISO、MOSI 引脚直接连接时，这些管脚上的时序。

注意： 1. 在改变 CPOL/CPHA 位之前，必须清除 SPI 使能位将 SPI 禁止。
2. 主设备和从设备必须配置成相同的时序模式。

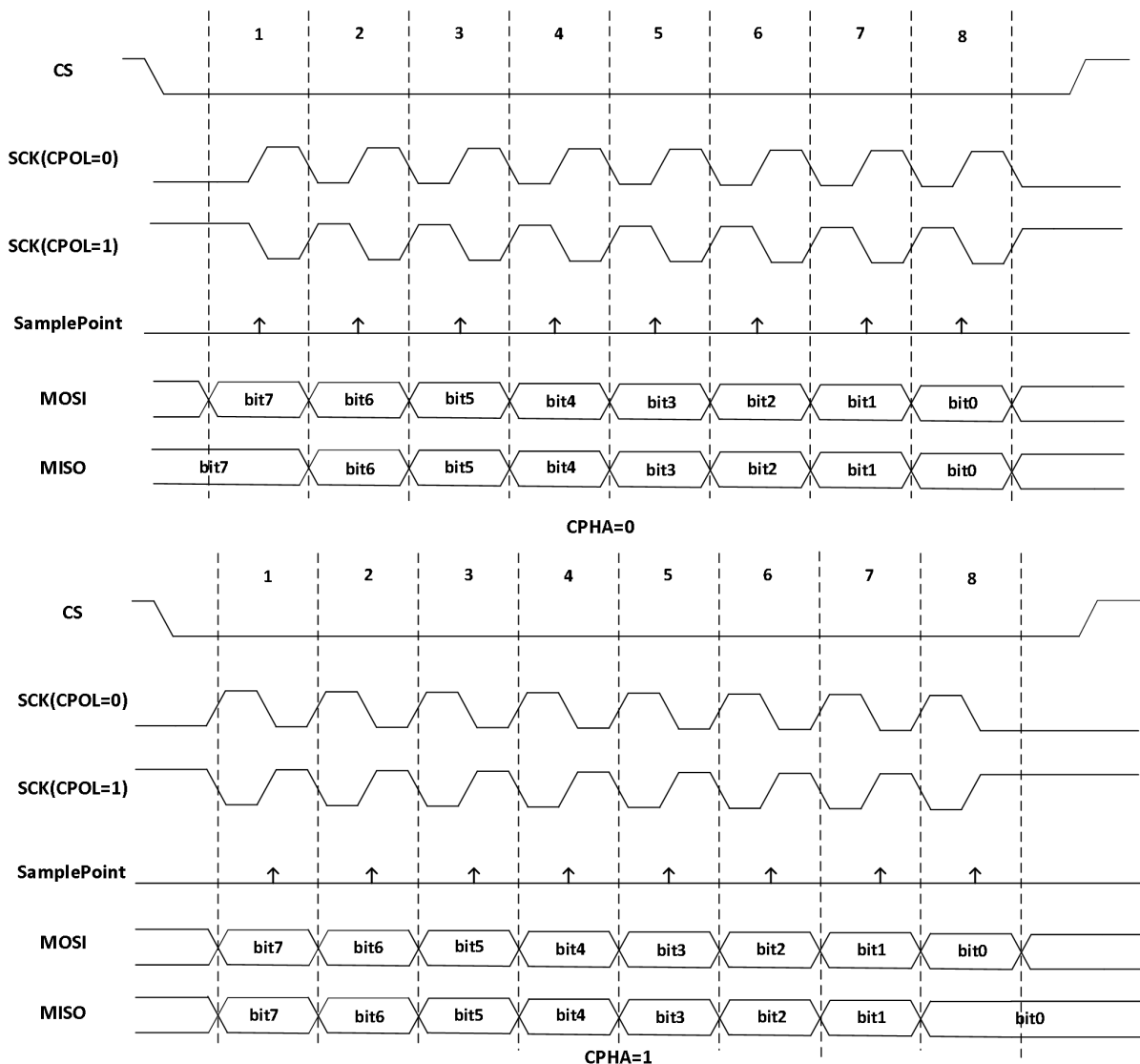


图 23-1 SPI 数据时钟时序图

23.4. SPI 寄存器

SPI 寄存器基地址：0x2030

偏置	名称	描述
0x00	SPI0CON	SPI0 控制寄存器
0x01	SPI0DAT	SPI0 数据寄存器
0x02	SPI0STS	SPI0 状态寄存器
0x03	SPI0MSK	SPI0 中断屏蔽寄存器

0x08	SPI1CON	SPI1 控制寄存器
0x09	SPI1DAT	SPI1 数据寄存器
0x0A	SPI1STS	SPI1 状态寄存器
0x0B	SPI1MSK	SPI1 中断屏蔽寄存器
0x10	SPI2CON	SPI2 控制寄存器
0x11	SPI2DAT	SPI2 数据寄存器
0x12	SPI2STS	SPI2 状态寄存器
0x13	SPI2MSK	SPI2 中断屏蔽寄存器
0x18	SPI3CON	SPI3 控制寄存器
0x19	SPI3DAT	SPI3 数据寄存器
0x1A	SPI3STS	SPI3 状态寄存器
0x1B	SPI3MSK	SPI3 中断屏蔽寄存器

23.4.1. SPIx 控制寄存器 SPIxCON(偏移: 00H、08H、10H、18H) (此处 x 代表 0,1,2,3, 下同)

比特	名称	属性	复位值	描述
7	RUNEN	RW	0x0	SPI 工作使能位 0: 禁止 1: 使能
6	SPI_SEL	RW	0x0	主从机选择位 0: 主机模式 1: 从机模式
5	RSV	-	-	保留
4:2	SPR	RW	0x0	SPI 波特率设置 000: SPI 系统时钟 2 分频 001: SPI 系统时钟 4 分频 010: SPI 系统时钟 8 分频 011: SPI 系统时钟 16 分频 100: SPI 系统时钟 32 分频 101: SPI 系统时钟 64 分频 110: SPI 系统时钟 128 分频 111: 保留
1	CPHA	RW	0x0	时钟相位控制 0: 在时钟 SCK 的奇边沿采样数据 1: 在时钟 SCK 的偶边沿采样数据
0	CPOL	RW	0x0	时钟极性控制 0: 时钟 SCK 初始电平为低电平 1: 时钟 SCK 初始电平为高电平

23.4.2. SPIx 数据寄存器 SPIxDAT(偏移：01H、09H、11H、19H)

比特	名称	属性	复位值	描述
7:0	SPIDAT	RW	0x00	主机时，作为发送数据寄存器； 从机时，作为接收数据寄存器。

23.4.3. SPIx 状态寄存器 SPIxSTS(偏移：02H、0AH、12H、1AH)

比特	名称	属性	复位值	描述
7:3	RSV	-	-	保留
2	SPI_CS_FALL	RW	0x0	SPI 从机通信开始标志位 0: SPI 通信未开始 1: SPI 通信开始，CS 下降沿置 1 该标志硬件置位，软件清 0
1	RX_TX_FLAG	RW	0x0	从机发送、接收选择标志位及控制 0: 接收 1: 发送 该标志软件置位，硬件 CS 下降沿清 0
0	BOVER	RO	0x0	当前字节数据交换结束标志位 0: 未结束 1: 结束 硬件置位，软件读 SPISTS 自动清除

23.4.4. SPIx 中断屏蔽寄存器 SPIxMSK(偏移：03H、0BH、13H、1BH)

比特	名称	属性	复位值	描述
7:2	RSV	-	-	保留
1	Filter_en	RW	0x0	滤波使能位，使能后，spi 的通信速率不高于 4M 0: 关闭 1: 开启
0	MBOVER	RW	0x0	屏蔽当前字节数据交换结束中断控制位 0: 不屏蔽 1: 屏蔽

23.5. SPI 操作说明

23.5.1. SPI 主模式初始化流程

- 1) 使能 SPIx 的时钟，关闭 SPIx 软复位。
- 2) 配置相应管脚为 SPIx 主机功能。
- 3) 配置 SPIxCON 寄存器。RUNEN、SPI_SEL、SPR、CPHA、CPOL 比特位。
- 4) 若使用中断方式，SPI0 需配置 T2CON 的 bit6 为 1（SPI1/2/3 无需配置此位）。

23.5.2. SPI 主模式发送

- 1) 拉低 CS 信号，选中从机器件。
- 2) 将数据写入 SPIxDAT。此动作将自动启动 SPI 主机进行交换。
- 3) 交换完成后可选择是否产生中断。如果中断被屏蔽，请轮询 SPIxSTS 的 BOVER 位。
- 4) 软件读 SPIxSTS 自动清除 SPIxSTS 的 BOVER 位。
- 5) 再次发送数据重复 2-4，直至 SPI 数据全部发送完成。
- 6) 拉高 CS 信号，结束发送。

23.5.3. SPI 主模式接收

- 1) 拉低 CS 信号，选中从机器件。
- 2) 将数据写入 SPIxDAT。此动作将自动启动 SPI 主机进行交换。
- 3) 交换完成后可选择是否产生中断。如果中断被屏蔽，请轮询 SPIxSTS 的 BOVER 位。接收到的数据将被保存在 SPIxDAT 中。
- 4) 软件读 SPIxSTS 自动清除 SPIxSTS 的 BOVER 位。
- 5) 再次接收数据重复 2-4，直至 SPI 数据全部接收完成。
- 6) 拉高 CS 信号，结束接收。

23.5.4. SPI 从模式初始化流程

- 1) 使能 SPIx 的时钟，关闭 SPIx 软复位。
- 2) 配置相应管脚为 SPIx 从机功能。
- 3) 配置 SPIxMSK 寄存器，不屏蔽 SPI 一字节交换完成中断
- 4) 使用中断方式，SPI0 需配置 T2CON 的 bit6 为 1（SPI1/2/3 无需配置此位）。
- 5) 使能 SPIx 中断，开总中断
- 6) 配置 SPIxCON 寄存器，选择 SPI 从机模式并运行。

23.5.5. SPI 从模式发送

- 1) 若产生中断，则先清除中断标志，判断 SPI 状态寄存器的 RX_TX_FLAG。
- 2) 若此时为发送，则将发送数据填入 SPIxDAT 中。

23.5.6. SPI 从模式接收

- 1) 若产生中断，则先清除中断标志，判断 SPI 状态寄存器的 RX_TX_FLAG。
- 2) 若此时为接收，则将 SPIxDAT 中数据读到数组中。

24. 接触式 IC 卡接口（ISO7816MS）

24.1. 概述

ISO7816MS 模块可作为满足 7816-3 标准的卡或读卡器使用，并同时兼容 T = 0 和 T = 1 传输协议。

24.2. 主要特性

- 支持 ISO7816 主从接口
- 符合 ISO/IEC7816-3 标准，支持 PBOC3.0 卡规范
- 支持 T=0/T=1 协议，即异步半双工字符传输协议
- 数据宽度 8 比特，奇偶校验可配置
- 奇偶校验位自动生成及奇偶校验错误检测
- 自动检测错误响应，检测到传输错误响应信号由硬件自动重发字符（自动重发不超过 7 次），是否重发可配
- 支持发送时的额外 ETU 功能
- 支持接收时最大等待时间配置，并提供超时中断
- FIFO 深度 16 字节
- 8 位的波特率选择器，波特率根据参数 F/D 调整
- 专用的 ETU 计数器用于自动发送过程字符（0x60）

24.3. ISO7816MS 寄存器

ISO7816MS 寄存器基地址：0x2000

偏置	名称	描述
0xA0	ISOCON0	控制寄存器 0
0xA1	ISOCON1	控制寄存器 1
0xA2	ISOCON2	控制寄存器 2
0xA3	ISOSTS0	状态寄存器 0
0xA4	ISOSTS1	状态寄存器 1
0xA5	ISOFIFOCL	FIFO 复位寄存器
0xA6	WTDATA_L	字符等待时间寄存器低字节
0xA7	WTDATA_H	字符等待时间寄存器高字节
0xA8	ISOBRC	波特率控制寄存器
0xA9	ISOBUFF	数据缓存寄存器
0xAA	ISODIO	直接 I/O 控制寄存器
0xAB	ISOMSK	中断屏蔽寄存器
0xAC	ISOTCON	ETU Counter 控制寄存器

0xAD	ISOTDAT_L	ETU Counter 计数初始值低字节
0xAE	ISOTDAT_H	ETU Counter 计数初始值高字节
0xAF	ISOTRLD_L	ETU Counter 计数自动重装低字节
0xB0	ISOTRLD_H	ETU Counter 计数自动重装高字节
0xB1	ISOTMSK	ETU Counter 中断屏蔽
0xB2	ISONULL	ETU Counter 过程字符设置寄存器
0xB3	ISOFUC	C3 计数/主从模式选择寄存器

24.3.1. 控制寄存器 ISOCON0 (偏移: A0H)

比特	名称	属性	复位值	描述
7	MOD	RW	0x0	ISO7816 接口工作方式控制位 0: 硬件自动方式 1: 软件直接 IO 方式
6	RX_FIFO_CLK_SEL	RW	0x0	接收模式时钟选择位 0: rx_fifo 时钟与系统时钟相同 1: 将 rx_fifo 时钟切换到 Rx_Clock(用于 7816 休眠唤醒, 休眠时需要将 RX fifo 时钟切换到由 C3 时钟分频而来的 RX clock)
5	TR	RW	0x0	收发状态控制位 0: 令硬件进入接收状态 1: 令硬件进入发送状态
4	TACT	RO	0x0	发送状态 (Transmit Active) 标志位 0: 发送已彻底完成 1: 数据正在 IO 线上发送, 或有数据在 SFR UBUF 中
3:0	RSV	-	-	保留

注: 在休眠之前需要把 RX_FIFO 时钟切换到 RX_CLOCK, 目的是为了让 RX_FIFO 在休眠唤醒的过程中保持稳定, 不随系统时钟变化(休眠之前系统时钟会切换到 32Khz)。休眠唤醒之后, 5 字节 APDU 指令接收完成之后再把 RX_FIFO 时钟切回到系统时钟。

24.3.2. 控制寄存器 ISOCON1 (偏移: A1H)

比特	名称	属性	复位值	描述
7	RSV	-	-	保留
6	FFPEN	RW	0x0	FIFO 满保护功能使能, 默认非使能 0: 关闭 FIFO 满保护功能 1: 打开 FIFO 满保护功能
5:3	EGT	RW	0x0	发送额外保护时间 (Extra Guard Time) 控制 000: 没有额外保护时间 (默认) 001~111: Extra Guard Time 为 1~7 etu

2:0	TRY	RW	0x7	自动重发次数控制 000: 重发次数为 0 001~110: 重发次数为 1~6 次 111: 重发次数为无限次 (默认)
-----	-----	----	-----	--

24.3.3. 控制寄存器 ISOCON2 (偏移: A2H)

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
3	TRSC	RW	0x0	硬件发送转接收模式选择 0: 软件切换发送接收模式 1: 发送结束硬件自动切换到接收模式
2	TPS	RW	0x0	协议选择控制 0: T = 0 (最少 12etu) 1: T = 1 (最少 11etu)
1	CWTEN	RW	0x0	CWT (字符等待时间) 功能使能 0: 关闭 CWT 计数器功能 1: 打开 CWT 计数器功能
0	WES0	RW	0x0	卡片发出的错误信号 (Error Signal) 的宽度控制 0: 1 个 etu 1: 2 个 etu

24.3.4. 状态寄存器 ISOSTS0 (偏移: A3H)

比特	名称	属性	复位值	描述
7:6	RSV	-	-	保留
5	CWT_FLAG	RW	0x0	字符等待标识, 用于接收模式 0: 等待时间未超过 WTDATA 设置值 1: 等待时间超过 WTDATA 设置值, (可标志 T1 模式一个 block 传输完成; 硬件置位, 软件写 1 清除)
4	RX_CHECK_DONE	RW	0x0	接收一个字节完成标志, 只有 check 完成后才能从 RX 切换到 TX, 否则可能出错。 0: 无中断 1: RX 接收完全完成(硬件置位, 软件写 1 清除)
3	SBIT	RW	0x0	IO 输入下降沿标志 (用于软件直接 IO 方式) 0: 尚未接收到下降沿, 或 ISOMSK.MSBIT=1 1: 接收到 1 个下降沿 硬件置 1, 软件写 1 清除
2	PE	WO	0x0	奇偶校验错误 (Parity Error) 标志, 软件实现 T=0 协议时, 应在发送后关注本标志: 0: 发送正确 1: 当前发送字符奇偶校验错误超过设定次数 软件实现 T=1 协议时, 应在接收命令帧后关注本标

				志： 0：接收正确 1：当前命令帧中曾出现字符奇偶校验错误
1	RX_INT	WO	0x0	接收一个字节成功中断标志 0：无中断 1：有中断(硬件置位，软件写 1 清除)
0	TX_INT	WO	0x0	发送 FIFO 为空且数据全部成功发送中断标志 0：无中断 1：有中断(硬件置位，软件写 1 清除)硬件自动清零，当 MS_SEL 为 0 时，写无效。

24.3.5. 状态寄存器 ISOSTS1(偏移：A4H)

比特	名称	属性	复位值	描述
7:6	RSV	-	-	保留位
5	TX_ERROR	RO	0x0	发送 Retry 后校验错标志 0：没有出错 1：Retry 后校验错
4	TX_FIFO_FULL	RO	0x0	发送 FIFO 满标志位 0：不满 1：满
3	TX_FIFO_EMPTY	RO	0x1	发送 FIFO 空标志 0：非空 1：空
2	FIFO_ALMOST_FULL	RO	0x0	接收 FIFO 内 5 字节数据标志 0：FIFO 内数据个数小于 5 1：FIFO 内数据个数大于等于 5(用于休眠的时候数据到来唤醒)
1	RX_FIFO_FULL	RO	0x0	接收 FIFO 满标志位 0：不满 1：满
0	RX_FIFO_EMPTY	RO	0x1	接收 FIFO 空标志 0：非空 1：空

注：FIFO_ALMOST_FULL 标志用于休眠的时候 7816 数据到来唤醒，等待 FIFO 里数据个数为 5 个的时候再开始从 FIFO 里读出数据，保持 RX_FIFO 在休眠唤醒之后接收 APDU 指令期间时钟的稳定。

24.3.6. FIFO 复位寄存器 ISOFIFOCL(偏移：A5H)

比特	名称	属性	复位值	描述
7:2	RSV	-	-	保留位

1	RX_FIFO_CLR	RW	0x0	接收 FIFO 清除/复位信号，在下一个时钟周期清除 FIFO 0: 不使能 1: 使能
0	TX_FIFO_CLR	RW	0x0	发送 FIFO 清除/复位信号，在下一个时钟周期清除 FIFO 0: 不使能 1: 使能

24.3.7. 字符等待时间寄存器 WTDATA(偏移: A6H、A7H)

比特	名称	属性	复位值	描述
15:8	WTDATA_H	RW	0x0B	7816 IO 的 WT 参数，即两个连续同向字符间最大延迟(用 ETU 计算，用与 T1 接收模式检测字组接收完成)
7:0	WTDATA_L	RW	0x20	

注: CWT (Character Waiting Time): 两个同向字符第一个沿之间的最大延迟。如下图所示。在 $T = 0$ 模式下，协议规定 $WT = WI * 960 * (F/f)$, ($WI=0\sim 255$, 缺省为 10), 缺省值为 9600 个 ETU; 在 $T = 1$ 模式下，协议规定 $CWT = 11etu + 2CWI$ ($CWI = 0\sim 15$, 缺省为 13), 缺省值为 8203 个 ETU。CWTDATA 寄存器的缺省值为 8203 ($T = 1$)。对卡来说，当其处于接收状态，两个连续的接收字符间延迟大于 CWTDATA 值时，SCISTAT 寄存器中的 CWT 标志置 1，表示传输超时。此寄存器的值的设置需要与 ATR 中设置的保护时间相符

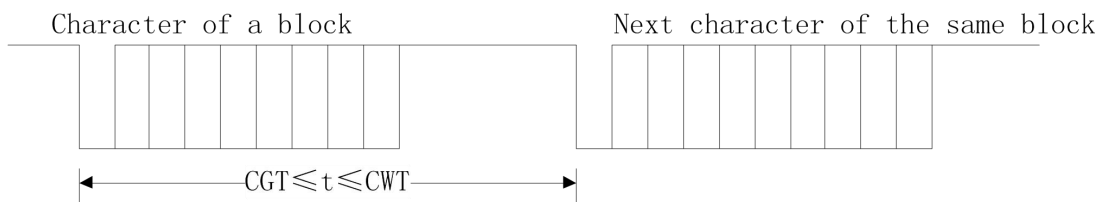


图 25-1 CWT 示意图

24.3.8. 波特率控制寄存器 ISOBRC (偏移: A8H)

比特	名称	属性	复位值	描述
7:4	FI	RW	0x0	分频值与波特率调整因子参数配置，详见下表
3:0	DI	RW	0x1	

波特率配置参数描述:

Clocks/etu	DI	0001	0010	0011	0100	0101	0110	1000
FI	F/D	1	2	4	8	16	32	64
0000/0001	372	372	186	93	X	X		31
1001	512	512	256	128	64	32	16	X

24.3.9. 数据缓存寄存器 ISOBUF (偏移: A9H)

比特	名称	属性	复位值	描述
7:0	DATA	RW	0x0	数据缓存

24.3.10. 直接 IO 控制寄存器 ISODIO(偏移: AAH)

比特	名称	属性	复位值	描述
7:2	RSV	-	-	保留
1	EN0	RW	0x0	数据方向控制 0: 输入使能 1: 输出使能
0	DIO	RW	0x0	数据输入输出 0: 低电平 1: 高电平

24.3.11. 中断屏蔽寄存器 ISOMSK (偏移: ABH)

比特	名称	属性	复位值	描述
7:6	RSV	-	-	保留
5	MCWT_FLAG	RW	0x0	字符等待标识中断屏蔽控制 0: 允许 CWT_FLAG 1: 屏蔽 CWT_FLAG
4	Mrx_check_done	RW	0x0	rx_check_done 中断屏蔽控制 0: 允许 rx_check_done 1: 屏蔽 rx_check_done
3	MSBIT	RW	0x0	SBF 中断屏蔽控制 (用于软件直接 IO 方式) 0: 允许 SBIT 中断 1: 屏蔽 SBIT 中断
2	MPS	RW	0x0	PE 断屏中蔽控制 0: 允许 PE 中断 1: 屏蔽 PE 中断
1	MRX_INT	RW	0x0	RX 中断屏蔽 0: 允许 RX 中断 1: 屏蔽 RX 中断
0	MTX_INT	RW	0x0	TX 中断屏蔽 0: 允许 TX 中断 1: 屏蔽 TX 中断

注: MCWT_FLAG 字符等待标识中断主要用于 T1 模式检测数据接收完成。(未知长度的字组)

24.3.12. ETU 计数器控制寄存器 ISOTCON(偏移: ACH)

比特	名称	属性	复位值	描述
7:5	RSV	-	-	保留
4	LAST	RW	0x0	ETU Counter 被停止以后继续将 60 发送完成的标志 0: 60 发送完成 1: 60 发送未完成 软件需要在启动自动发送 60 的时候置位 LAST 信号, 自动发送 60 完成, 硬件会拉低 LAST 信号

3	TF	RW	0x0	ETU 计数器溢出标志 0: ETU 计数器未溢出（或工作在模式 0） 1: ETU 计数器溢出 硬件置位，软件写 1 清除 只在模式 1 和模式 2 下，TF 有效
2:1	TMOD	RW	0x0	ETU 计数器 模式选择 00: 模式 0，自动重载计数模式。以 1ETU 为计数单位，ETU 计数器溢出后，自动发送 60，不设置 TF 标志位 01: 模式 1，自动重载计数模式。以 1ETU 为计数单位，ETU 计数器溢出后，不发送 60，设置 TF 标志位，可触发中断 1X: 模式 2，自动重载定时模式，定时时钟是系统时钟，ETU 计数器溢出后，设置 TF 标志位，可触发中断
0	TS	RW	0x0	ETU 计数器启动控制 0: 停止 1: 启动

注：写入 TS 停止 bit 可以任意时候进行,但是一旦已经进入 ETU 计数状态后,需要完成一次 ETU 发送,ETU 才会真正停止。

24.3.13. ETU 计数器初始值寄存器 ISOTDAT(偏移：ADH、AEH)

比特	名称	属性	复位值	描述
15:8	ISOTDAT_H	RW	0x00	ETU 计数器计数初始值
7:0	ISOTDAT_L	RW	0x00	

24.3.14. ETU 计数器自动重载寄存器 ISOTRLD(偏移：AFH、B0H)

比特	名称	属性	复位值	描述
15:8	ISOTRLD_L	RW	0x00	ETU 计数器计数重载值
7:0	ISOTRLD_H	RW	0x00	

24.3.15. ETU 计数器中断屏蔽寄存器 ISOTMSK (偏移：B1H)

比特	名称	属性	复位值	描述
7:1	RSV	-	-	保留
0	MTF	RW	0x0	ETU 计数器溢出中断（ISOTCON 的 TF 位）屏蔽控制 0: 不屏蔽 1: 屏蔽

24.3.16. ETU 计数器过程字符设置寄存器 ISONULL(偏移：B2H)

比特	名称	属性	复位值	描述
7:0	DATA	RW	0x60	ETU 计数器送的空字符数值

24.3.17. C3 计数/主从模式选择寄存器 ISOFC (偏移：B3H)

比特	名称	属性	复位值	描述
7:0	RSV	-	-	保留
1	M_BUAD	RW	0x0	C3 时钟分频选择 0: 系统时钟 2 分频 1: 系统时钟 4 分频
0	MS_SEL	RW	0x0	主从功能选择位 0: 从机 1: 主机

24.4. ISO7816MS 操作说明

24.4.1. 主机模式选择与使用

- 1) 软件选择一个 GPIO 为 C2 模拟 7816 主机复位信号;
- 2) 配置 ISOCON 的 M_BUAD 位选择 7816 主模式时钟分频;
- 3) 配置 ISOCON 的 MS_SEL 位 1, 选择主机模式;
- 4) 控制选择的 GPIO 电平和主机输出时钟, 模拟主机功能。

24.4.2. 查询方式发送

- 1) 7816 模块的时钟 (缺省为使能), 清除 7816 模块软复位;
- 2) 设置 ISOCON(硬件自动、发送模式), ISOCON1(保护时间、自动重发次数), ISOCON2(T=0 协议、错误信号宽度), ISOBRC(通讯波特率);
- 3) 设置 ISOMSK(屏蔽所有中断), ISOSTS(清空所有中断), ISOFIFOCL(清 FIFO);
- 4) 将待发送数据写入 ISOBUF 寄存器中(字节写);
- 5) 查询发送 FIFO 是否满, 如果发送 FIFO 没有满, 可以继续写下一个数据;
- 6) 当所有的数据全部写入 ISOBUF 寄存器后, 软件查询 tx_int 标志, 若全部数据传输成功完成, 则硬件会置 tx_int 标志, 程序查询到 tx_int 置起后需把 tx_int 位写 1 清 0;
- 7) 发送结束后需要清空发送 FIFO(ISOCON1)。

24.4.3. 中断方式发送

- 1) 使能 7816 模块的时钟 (缺省为使能);
- 2) 设置 ISOCON(硬件自动、发送模式), ISOCON1(保护时间、自动重发次数), ISOCON2(T=0 协议、错误信号宽度), ISOBRC(通讯波特率);
- 3) 设置 ISOMSK(打开发送中断), ISOSTS(清空所有中断标志), ISOFIFOCL(清 FIFO);
- 4) 设置 VIC(见 VIC 配置说明);
- 5) 将待发送数据写入 ISOBUF 寄存器中, 然后查询发送 FIFO 是否满, 如果发送 FIFO 没有满, 可以继续写下一个数据;
- 6) 当所有的数据发送成功, 会产生中断, 则软件应将下一组待发送数据写入

ISOBUF 寄存器中，然后清除中断标志。

24.4.4. 中断方式接收

- 1) 使能 7816 模块的时钟（缺省为使能）；
- 2) 设置 ISOCON(硬件自动、接收模式)，ISOCON1(保护时间、自动重发次数)，ISOCON2(T=0 协议、错误信号宽度) ISOBRC(通讯波特率)；
- 3) 设置 ISOMSK(打开接收中断)，ISOSTS(清空所有中断标志)；
- 4) 设置 VIC(见 VIC 配置说明)；
- 5) 当成功接收了一个字节之后，会产生 rx_int 中断请求，而后再判断接收 FIFO 是否为非空，非空则中断响应服务程序应从 ISOBUF 中读取收到的数据，然后清除中断标志。

24.4.5. ETU timer 操作

ISO7816 接口内置了一个具有自动重加载功能的 16 位计数器/定时器 ETU Counter。向下计数，溢出时计数值为 0。可以完成自动发送过程字符（0x60）功能。

ETU Counter 支持如下 3 种工作模式：

1. 模式 0：自动重载计数模式。SFR ISOTDAT 每个 ETU 减 1，溢出后自动将 SFR ISOTRLD 的内容重加载到 SFR ISOTDAT，并自动发送 0x60，但不设置 TF 标志位。
2. 模式 1：自动重载计数模式。SFR ISOTDAT 每个 ETU 加 1，溢出后自动将 SFR ISOTRLD 的内容重加载到 SFR ISOTDAT，但不会自动发送 0x60，设置 TF 标志位。
3. 模式 2：自动重载定时模式，使用系统时钟进行定时。SFR ISOTDAT 每个系统时钟周期加 1，溢出后自动将 SFR ISOTRLD 的内容重加载到 SFR ISOTDAT，但不发送 0x60，设置 TF 标志位。

注：如果预期计数值为 n etu，那么需要向 ISOTDAT & ISOTRLD 载入(n-1)值，才会得到比较准确的计数间隔。

使用 ETU Counter 完成自动发送过程字符 0x60 功能的操作流程如下：

- 1) 清自动发送 0x60 结束标志；
- 2) 设置 7816 硬件发送模式；
- 3) 填充 ISONULL 空字符(模式 0)；
- 4) ETUcounter 中断屏蔽位配置；
- 5) ISOTCON 模式配置(0、1、2)；
- 6) ISOTCON 启动定时器；
- 7) 拉高 LAST 信号(自动发送 60 结束标志)。

注：软件在启动 ISO7816 接口发送有效数据之前，必须先关闭 ETU Counter 的自动发送 60

功能(查询 ISOTCON bit4, 等待一个 60 发送结束才能进行数据收发操作)。

24.4.6. 芯片休眠之后 7816 唤醒

- 1) 休眠之前切换到 7816 硬件接收模式, 打开 7816 接收数据中断, 切换 RX_FIFO 时钟到 C3_CLK, 关闭全部中断源;
- 2) 使能 32Khz 时钟, 然后延时 30us 以上等待 32Khz 时钟稳定;
- 3) 选择 32Khz 时钟, 此时 CPU、RAM、7816 等模块都在 32 千赫兹时钟下运行, 然后关闭 32Mhz 时钟, 写 SLEEP 寄存器进入休眠;
- 4) 休眠唤醒之后首先打开 32Mhz 时钟源, 然后延时 1.5us 以上等待 32Mhz 时钟稳定;
- 5) 选择系统时钟为 32Mhz, 关闭 32Khz 时钟;
- 6) 使能 7816 Slave 中断源, 之后 CPU 会进入 7816 Slave 中断服务程序;
- 7) 清中断, 等待 FIFO_ALMOST_FULL 信号, FIFO_ALMOST_FULL 信号标志着 rxfifo 内数据量为 5 字节;
- 8) 读取 fifo 里面的 5 字节数据;
- 9) 再次清中断;
- 10) 切换 7816 rx_fifo 为系统时钟;
- 11) 处理接收到的 5 字节 APDU 指令;
- 12) 休眠唤醒软件代码之中, 开、关 32Mhz 时钟之间的代码需要放在 RAM 之中运行。

25. 版本号

版本号用于标识芯片型号，是一个固定的 32bit 数，本芯片型号为 RJM8L151，其版本号固定为 0x4d313531。

25.1. 版本号寄存器

版本号寄存器基地址：0x2000

地址	名称	描述
0x1A	VERSION0	版本号寄存器 0
0x1B	VERSION1	版本号寄存器 1
0x1C	VERSION2	版本号寄存器 2
0x1D	VERSION3	版本号寄存器 3

25.1.1. 版本寄存器 VERSION0（偏移：1AH）

比特	名称	属性	复位值	描述
7:0	VERSION0	RO	0x31	版本号 0

25.1.2. 版本寄存器 VERSION1（偏移：1BH）

比特	名称	属性	复位值	描述
7:0	VERSION1	RO	0x35	版本号 1

25.1.3. 版本寄存器 VERSION2（偏移：1CH）

比特	名称	属性	复位值	描述
7:0	VERSION2	RO	0x31	版本号 2

25.1.4. 版本寄存器 VERSION3（偏移：1DH）

比特	名称	属性	复位值	描述
7:0	VERSION3	RO	0x4d	版本号 3

26. 编程接口

Flash 型程序存储器提供用户便利地对同一芯片进行程序的更新和修改。另外 RJM8L151 系列芯片提供了 5 线的在线编程方式。用户可将进行过编程或未经过编程的单片机芯片连同电路板一起制成，最后阶段进行程序的更新和程序的烧录，在无需去除或者重新插入芯片的情况下方便地保持程序为最新版本。

RJM8L151 系列芯片与烧录器对应引脚的关系如下：

引脚名称		功能
烧录器	RJM8L151 系列芯片	
NRST	RST	芯片复位
JTAG_TDI	P03	数据输入
JTAG_TMS	P02	模式选择
JTAG_TCK	P01	时钟输入
JTAG_TDO	P00	数据输出

芯片内部的程序存储都可以通过 5 线接口在线进行编程。其中 P03 用于数据的输入，P02 用于模式的选择，P01 用于提供时钟，P00 用于数据的输出，RST 用于芯片的复位。芯片在线烧录的详细使用将由专门的参考文献提供。

附录 1：寄存器比特位属性及说明

比特位属性	说明
RW	Read/Write，软件可以读写这些位
RO	Read Only，软件只可以读这些位
WO	Write Only，软件仅可以写这些位，读这些位时将会返回该位的复位值
W1C	Write 1 Clear，软件可以通过向该位写 1 来清零，向其写 0 将不影响该位
W0C	Write 0 Clear，软件可以通过向该位写 0 来清零，向其写 1 将不影响该位

附录 2：专业术语缩写及说明

缩写	说明
B	Byte, 字节
COS	Chip Operating System, (智能卡)片上操作系统
CPU	Central Processing Unit, 中央处理器
CRC	Cyclic Redundancy Check, 循环冗余校验
DES	Data Encryption Standard, 数据加密标准
TDES	Triple Data Encryption Standard, 三重 DES
ESD	Electro-Static discharge, 静电放电
FD	Frequency Detector, 频率检测器
FPS	Flash Page Size, Flash 页面大小
ISO7816	International Standard ISO/IEC7816 module, ISO/IEC7816 国际标准接口
MAC	Memory Access Control, 内存访问控制模块
NMROM	Normal Mode Read Only Memory, 正常模式只读存储器
NVM	Non-Volatile Memory, 非易失性存储器
OTP	One Time Programmable, 一次可编程存储器
POR	Power On Reset, 上电复位
PDR	Power down Reset, 掉电复位
BOR	Brown-out Reset, 欠压复位
RNG	Random Number Generator, 随机数发生器
SFR	Special Function Register, 特殊功能寄存器
SIM	Subscriber Identity Module, 用户识别模块
SN	Chip Serial Number, 芯片序列号 (唯一标识符)
UI	User Information, 用户信息
VD	Voltage Detector, 电压检测器
VR	Voltage Regulator, 电压调整器
WDT	Watch Dog Timer, 看门狗计数器
WUT	Wake-Up Timer, 唤醒定时器

附录 3：版本修订

版本	日期	作者	描述
V1.0	2019.10.31	吴水源、王向辉	初始化版本
V1.1	2019.11.05	吴水源	1.增加中断使能寄存器IEN0第6位描述 2.修改部分文字描述错误
V1.2	2019.11.10	吴水源	1.修改目录格式错误 2.增加编程接口章节
V1.3	2020.03.09	吴水源	1.优化GPIO复用寄存器ADC通道说明； 2.增加ADC通道配置使用说明； 3.增加deepsleep使用配置注意；
V1.4	2020.05.11	吴水源	1.修改IOCFG_CTRL8最高位描述； 2.增加IO复用功能表； 3.MDU模块描述； 4.修改外部晶振使能SCCM2寄存器关于XT_16M_EN使能配置；
V1.5	2020.06.29	贾贝、吴水源	1.增加模块寄存器操作说明；
V1.6	2020.12.14	sunzuan	1.纠正21.4.10中关于发送状态位和接收状态位的描述错误（之前的版本把二者换位了，属于笔误）； 2.纠正21.5.5和21.5.6中关于I2Cx_S_STS书写上的笔误；
V1.7	2021.03.17	sunzuan	1.纠正16.1中关于WDT的描述错误；
V1.8	2021.04.27	sunzuan	1.修改6.2.1节上掉电复位描述；
V1.9	2021.07.15	sunzuan	1.增加21.4.12节I2Cx从机中断屏蔽寄存器的描述；
V1.10	2021.07.30	sunzuan	1.完善文档中描述不清之处； 2.增加附录1、附录2、附录3；
V2.00	2021.07.31	sunzuan	1.增加21.1节关于I2C主从机接口管脚分布的描述；
V2.01	2021.08.10	sunzuan	1.修改17.3节各寄存器偏移地址描述的笔误； 2.修改22.4.4节SxRELH寄存器bit位描述的笔误；
V2.02	2021.08.19	sunzuan	1.增加9.3节中断入口地址的描述； 2.修改 5.1 节注 1，删除“SCRSTEN=0x00”
V2.03	2021.09.24	sunzuan	1.增加16.7节看门狗复位时间计算公式； 2.修改 18.2 节文字笔误；
V2.04	2021.10.18	sunzuan	1.补充 16.7 节看门狗复位时间计算公式中时间单位秒； 2.修改 26 章中的文字笔误，“惊喜”改为“进行”；