

UM800x 用户手册

版本：V1.5



UNICMICRO
广芯微电子

广芯微电子（广州）股份有限公司

<http://www.unicmicro.com/>

条款协议

本文档的所有部分，其著作权归广芯微电子（广州）股份有限公司（以下简称广芯微电子）所有，未经广芯微电子授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。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，广芯微电子及所属员工恕不为其担保任何责任。除此以外，本文档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，广芯微电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，广芯微电子已尽量做到合理注意，但是，广芯微电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，广芯微电子不承担任何责任。
3. 对于因使用本文档中的广芯微电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，广芯微电子不承担任何责任。本文档所记载的内容不应视为对广芯微电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 使用本文档中记载的广芯微电子产品时，应在广芯微电子指定的范围内，特别是在最大额定值、电源工作电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用广芯微电子产品而产生的故障或损失，广芯微电子不承担任何责任。
5. 虽然广芯微电子一直致力于提高广芯微电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，广芯微电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当广芯微电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。

目录

1	系统概述	1
1.1	主要特性	1
1.2	功能框图	3
1.3	电源框图	4
2	处理器	5
2.1	主要特性	5
2.2	程序状态字寄存器 (PSW)	5
2.3	累加器 (ACC)	5
2.4	B 寄存器	5
2.5	数据指针 (DPTR)	5
3	存储器	6
3.1	主要特性	6
3.2	内部 RAM(ISRAM)	6
3.3	外部存储空间	6
3.4	程序存储器 EFLASH	7
4	系统配置 (SFR)	8
4.1	地址映射	8
4.1.1	P0	8
4.1.2	SP	9
4.1.3	DPTR	9
4.1.4	BEEPCTR	10
4.1.5	PCON	11
4.1.6	TCON	11
4.1.7	TMOD	11
4.1.8	TL0	12
4.1.9	TL1	12
4.1.10	TH0	12
4.1.11	TH1	12
4.1.12	PDSEL	13
4.1.13	POREN	13
4.1.14	P1	13
4.1.15	LDOTRIML	14
4.1.16	DPS	15
4.1.17	ADCIER	15
4.1.18	P0DR	15
4.1.19	S0CON	16
4.1.20	S0BUF	16
4.1.21	S1CON	17
4.1.22	S1BUF	17

4.1.23	S1REL.....	17
4.1.24	UARTEN.....	18
4.1.25	RCLTRIM.....	18
4.1.26	P2.....	19
4.1.27	OUS.....	19
4.1.28	OINTEN.....	20
4.1.29	OINTUS.....	20
4.1.30	OSTATUS.....	21
4.1.31	OPSET.....	21
4.1.32	IEN0.....	21
4.1.33	IP.....	22
4.1.34	S0REL.....	23
4.1.35	ADCGCR0.....	23
4.1.36	ADCGCR1.....	23
4.1.37	REMAP.....	24
4.1.38	CLKST.....	24
4.1.39	ESTCR.....	25
4.1.40	XTHCTR.....	25
4.1.41	ADCGCR2.....	26
4.1.42	ADCGCR3.....	26
4.1.43	ADCDR0.....	27
4.1.44	ADCDR1.....	27
4.1.45	IEN1.....	27
4.1.46	LDOTRIMH.....	28
4.1.47	RCHTRIMH.....	28
4.1.48	RCHTRIML.....	28
4.1.49	PWMxPL/H.....	29
4.1.50	PWMxDL/H.....	29
4.1.51	PWMxCON.....	29
4.1.52	PSW.....	30
4.1.53	OADRL/H.....	31
4.1.54	ODATA.....	31
4.1.55	OCTRL.....	31
4.1.56	P0PD.....	32
4.1.57	P0OD.....	32
4.1.58	P0CS.....	33
4.1.59	RSTSTAT.....	33
4.1.60	SYSDIV.....	34
4.1.61	P1PD.....	34
4.1.62	P1OD.....	35
4.1.63	P1CS.....	35
4.1.64	LVDCON.....	36
4.1.65	PCLK0.....	36
4.1.66	PCLK1.....	37

4.1.67	ACC.....	37
4.1.68	PxIRQ.....	38
4.1.69	P2PD.....	38
4.1.70	P1DR.....	38
4.1.71	PRESET0.....	39
4.1.72	PRESET1.....	40
4.1.73	PxIEN.....	40
4.1.74	P2OD.....	40
4.1.75	ADCCONT.....	41
4.1.76	RCHDIV.....	41
4.1.77	B.....	41
4.1.78	PxPU.....	42
4.1.79	P2CS.....	42
4.1.80	ADCHL.....	43
4.1.81	ADCCSTAT.....	43
4.1.82	ADCSPW.....	44
4.1.83	CLKCON.....	44
4.1.84	PxOEN.....	45
4.1.85	P2DR.....	45
4.1.86	ADCVREF.....	46
4.1.87	ADCCDR0.....	46
4.1.88	ADCCDR1.....	46
4.1.89	LVD_LV.....	47
4.2	系统时钟.....	47
4.2.1	主要特性.....	47
4.2.2	时钟定义.....	47
4.2.3	时钟结构图.....	48
4.3	复位源.....	48
4.3.1	主要特性.....	48
4.3.2	看门狗复位.....	49
4.3.3	LVD 与 LVR 复位.....	49
4.3.4	外部复位.....	49
4.3.5	寄存器.....	49
4.4	低功耗模式.....	49
4.4.1	主要特性.....	49
4.4.2	低功耗模式.....	50
4.4.3	低功耗模式表.....	50
5	EFC.....	51
5.1	概述.....	51
5.2	主要特性.....	51
5.3	EFLASH 读效率.....	51
5.4	软件流程.....	51
5.4.1	Read 操作.....	51
5.4.2	Write 操作.....	51

5.4.3	Erase 操作.....	52
6	PWM (脉宽调制模块).....	54
6.1	特性	54
6.2	功能描述	54
6.3	PWM 输出时序	55
6.4	使用流程	55
7	GPIO(I/O 端口)	56
7.1	主要特性	56
7.2	端口复用	56
7.3	端口中断	57
7.4	IO_IE	58
7.4.1	P0_IE.....	58
7.4.2	P1_IE.....	58
7.4.3	P2_IE.....	59
7.5	IO_SR.....	60
7.5.1	P0_SR.....	60
7.5.2	P1_SR.....	60
7.5.3	P2_SR.....	61
7.6	使用流程	62
7.6.1	IO 输出	62
7.6.2	IO 输入	62
7.6.3	中断触发模式.....	62
8	TIMER0/1(定时器).....	63
8.1	主要特性	63
8.2	定时器 0/1 工作方式	63
8.3	使用流程	63
9	BEEPER (蜂鸣器)	64
9.1	特性	64
9.2	使用流程	64
10	UART0/1 (增强型串口).....	65
10.1	特性	65
10.2	UART0 工作模式	65
10.3	UART1 工作模式	68
10.4	多机通讯	68
10.5	波特率	69
10.6	使用流程	69
10.6.1	UART0 mode1 使用流程	69
10.6.2	UART1 modeB 使用流程	70
11	SPI	72
11.1	主要特性	72

11.2	寄存器描述.....	72
11.2.1	SPCR1 控制寄存器.....	72
11.2.2	SPCR2 控制寄存器.....	73
11.2.3	SPCR3 控制寄存器.....	74
11.2.4	SPCR4 控制寄存器.....	74
11.2.5	SPIIE 中断使能寄存器.....	75
11.2.6	SPSR 状态寄存器.....	75
11.2.7	TXBUF 发送数据寄存器.....	76
11.2.8	RXBUF 接收数据缓冲寄存器.....	76
11.3	使用流程.....	76
11.3.1	SPI 初始化.....	76
11.3.2	SPI 发送流程.....	77
11.3.3	SPI 接收流程.....	77
12	LPTIM（低功耗定时器）.....	78
12.1	概述.....	78
12.2	主要特性.....	78
12.3	工作模式.....	78
12.3.1	普通定时器.....	78
12.3.2	Trigger 脉冲触发计数.....	78
12.3.3	外部异步脉冲计数.....	79
12.3.4	Timeout 模式.....	79
12.3.5	计数模式.....	79
12.3.6	外部触发的超时唤醒.....	79
12.3.7	16bit PWM.....	80
12.4	寄存器描述.....	80
12.4.1	LPTCFG0 寄存器.....	80
12.4.2	LPTCFG1 寄存器.....	81
12.4.3	LPTCNT 计数值寄存器.....	81
12.4.4	LPTCMP 比较值寄存器.....	82
12.4.5	LPTTARGET 目标值寄存器.....	82
12.4.6	LPTIE 中断使能寄存器.....	82
12.4.7	LPTIF 中断标志寄存器.....	83
12.4.8	LPTCTRL 控制寄存器.....	83
12.5	使用流程.....	83
12.5.1	普通定时器.....	83
12.5.2	PWM 输出.....	84
12.5.3	Trigger 脉冲触发计数模式.....	84
12.5.4	外部异步脉冲计数模式.....	85
12.5.5	Timeout 模式.....	85
13	I2C.....	87
13.1	概述.....	87
13.2	主要特性.....	87
13.3	寄存器描述.....	87

13.3.1	I2C_SLAVE_ADDR1 从设备地址寄存器 1	87
13.3.2	I2C_CLK_DIV 时钟分频寄存器	88
13.3.3	I2C_CR0 控制寄存器 0	88
13.3.4	I2C_CR1 控制寄存器 1	89
13.3.5	I2C_SR0 状态寄存器 0	90
13.3.6	I2C_SR1 状态寄存器 1	90
13.3.7	I2C_DR 数据寄存器	91
13.3.8	I2C_SLAVE_ADDR2 从设备地址寄存器 2	91
13.4	功能描述	91
13.4.1	模式选择	91
13.4.2	I2C 从模式	92
13.4.3	I2C 主模式	93
13.5	使用流程	94
13.5.1	I2C 初始化	94
13.5.2	作为主发送器	94
13.5.3	作为主接收器	94
13.5.4	作为从发送器	95
13.5.5	作为从接收器	95
14	ADC (模/数转换器)	96
14.1	主要特性	96
14.2	使用流程	96
15	WDT (看门狗定时器)	97
15.1	概述	97
15.2	使用流程	97
16	LVD	98
16.1	概述	98
16.2	主要特性	98
16.3	使用流程	98
17	中断	99
17.1	特性	99
17.2	中断汇总	99
18	指令集	100
18.1	指令操作数说明	100
18.2	算数操作指令	100
18.3	逻辑操作指令	101
18.4	数据传送指令	101
18.5	控制程序转移指令	102
18.6	位操作指令	103
19	版本维护	104

图目录

图 1-1: 芯片功能图	3
图 1-2: 电源框图	4
图 4-1: 时钟结构图	48
图 5-1: 写操作流程	52
图 5-2: 擦除操作流程	53
图 6-1: PWM 输出范例	54
图 6-2: PWM 输出周期或占空比改变范例	54
图 13-1: 7 位从发送器的传送图	92
图 13-2: 7 位从接收器的传送图	92
图 13-3: 7 位主发送器的传送图	93
图 13-4: 7 位主接收器的传送图	93

表目录

表 4-1: 系统 SFR 映射表	8
表 6-1: 输出时序 (PWMxSS=1)	55
表 7-1: P0 口复用	56
表 7-2: P1 口复用	56
表 7-3: P2 口复用	57
表 10-1: UART0 工作方式列表	65
表 10-2: UART1 工作方式列表	68
表 10-3: 波特率误差表	69
表 13-1: I2C 寄存器列表	87

1 系统概述

UM800x 单片机是广芯微电子（广州）股份有限公司研制的基于单周期 8051 内核的低功耗 8 位 IoT 处理器芯片。芯片系统采用了独特的低功耗设计技术, 宽工作电压 2.0~5.5V, 内部集成了 32KB 的 Flash、最大 2KB+256B 的 SRAM 以及 12 位 1Mps 的 SAR ADC 以及 UART、SPI、I2C、PWM 等通用外围通讯接口。

UM800x 单片机具有资源高整合度、高抗干扰性能、高可靠性、低功耗设计以及极简外围器件等技术特点。内置高速时钟 ROSC、LDO 和 POR 模块, 板级电路可免晶振、LDO、复位电路。支持 Keil MDK 通用集成开发环境, 支持 C 语言和汇编语言进行软件开发。

典型应用场景

- 工业物联网终端应用
- 智慧城市, 智能家居应用
- 智能传感器终端应用
- 玩具控制等通用控制器应用

1.1 主要特性

- **超低功耗电源管理系统**
 - 0.59 μ A @3.0V DeepSleep+定时唤醒, 低速时钟运行, IO、SRAM 以及寄存器数据保持
 - 0.32 μ A @3.0V Stop 模式, 所有时钟停止, IO、SRAM 以及寄存器数据保持
 - 95 μ A/MHz @3.0V Active 模式
 - 内置 ROSC/LDO/POR 模块, 板级系统可免去晶振/LDO/复位电路
- **处理器**
 - 8 位高性能 8051 单片机
 - 单周期指令集, 系统最高主频 24MHz
 - 0 等待周期取指 @0~16MHz
- **存储器**
 - RAM: Idata 256B, Xdata 最大 2KB
 - 32KB eFlash:
 - ✓ Sector 大小: 512B
 - ✓ Sector 擦除时间: 5ms
 - ✓ Byte 编程时间: 7.5 μ s

- ✓ 擦写次数：20,000 次
- ✓ 数据保存时间：100 年 @常温

● 定时/计数器

- 2 个 16 位通用定时器 T0/T1，可作 LPTimer 用
- 1 个 16 位低功耗定时器 LPTimer，支持 PWM 输出
- 1 个看门狗 WDT

● 时钟

- 内部高速时钟 RCH：48MHz（1 分频，不能跑），最高可跑 24MHz（2 分频，需设 flash 读等待时间为 1），默认下为 16MHz（3 分频）
- 内部低速时钟 RCL：32KHz
- 外部晶体振荡器：24MHz（max）
- 外部时钟输入：24MHz（max）

● 外围接口

- PWM：4 路 16 位 PWM 输出（含 LPTimer 中 1 路）
- UART：2 个增强型串口 UART0/UART1
- ADC：8 通道 12 位 SAR ADC，1Msps 采样速率
- I2C：主/从模式，速率 400Kbps（max）
- SPI：1 路，主/从模式，Mode0/1/2/3 协议，最高速率 12Mbps
- GPIO：最大 17 个，内置上/下拉可配置
- BEEPER：蜂鸣器，输出频率和极性可配置

● 安全功能

- 硬件级防抄板设计
- 低电压检测 LVD，可监控电源和 I/O 口电压
- 掉电复位 LVR，防死机设计
- 16 字节全球唯一芯片序列号 UID

● 主要电气参数

- 工作电压：2.0 ~ 5.5V
- 工作温度：-40 ~ 105°C
- ESD 保护：6KV（HBM）

● 开发支持

- 内置 Boot 引导程序，支持 UART 下载，支持 ISP 和 IAP 应用程序更新
- 完整 SDK 开发包、EVB 硬件开发套件

- 1 拖 14 并行下载器、代码烧写生产工具

1.2 功能框图

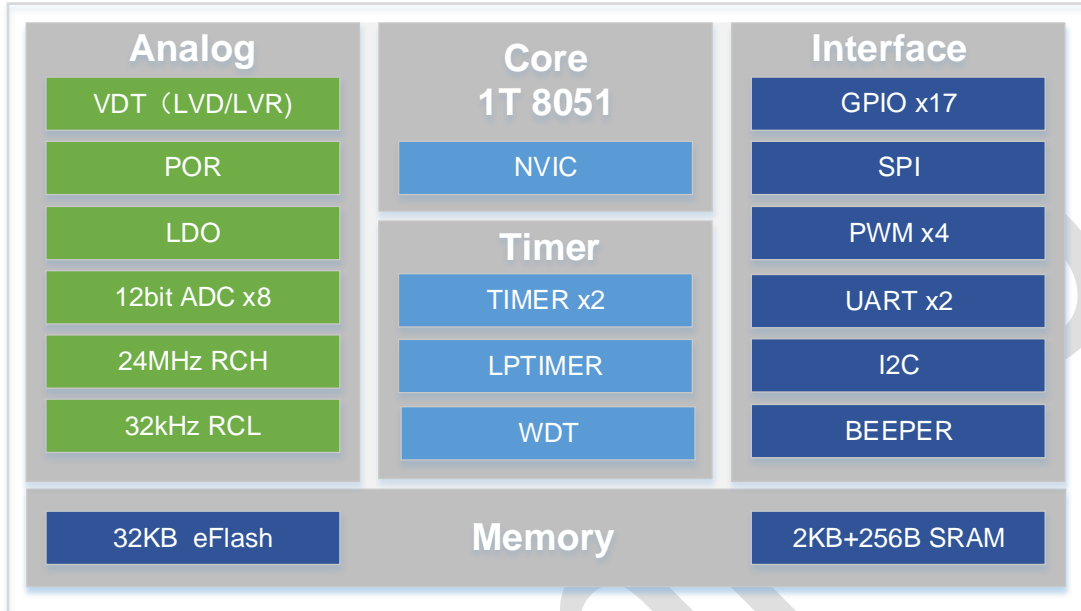


图 1-1：芯片功能图

1.3 电源框图

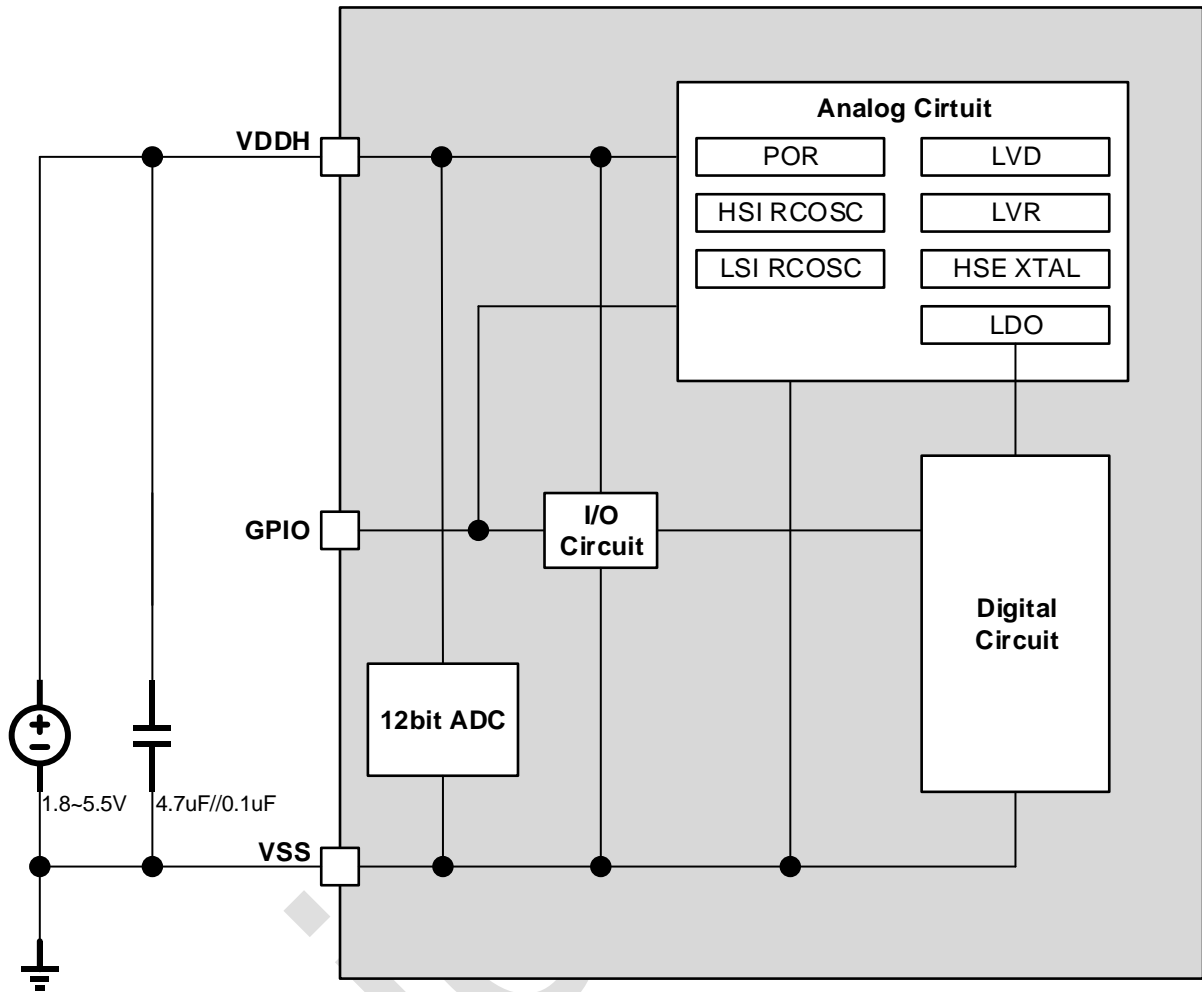


图 1-2：电源框图

注：每组电源都需要一个去耦电容，去耦电容尽量靠近相应电源管脚。

2 处理器

2.1 主要特性

CPU 内核寄存器：ACC, B, PSW, SP, DPS, DPL, DPH, DPL1, DPH1

2.2 程序状态字寄存器（PSW）

程序状态字（PSW）寄存器包含了程序状态信息。

2.3 累加器（ACC）

累加器 ACC 是一个常用的专用寄存器，常用于存放参加运算的操作数及运算结果。

2.4 B 寄存器

在乘除指令中，会用到 B 寄存器，在其他指令中，B 寄存器可作为通用暂存寄存器。

2.5 数据指针（DPTR）

数据指针 DPTR 是一个 16 位专用寄存器，其高位字节寄存器用 DPH 表示，低位字节寄存器用 DPL 表示。它们既可以作为一个 16 位寄存器 DPTR 来处理，也可以作为 2 个独立的 8 位寄存器 DPH 和 DPL 来处理。包含双数据指针 DPTR&DPTR1，通过 DPS(bit0)寄存器选择。

3 存储器

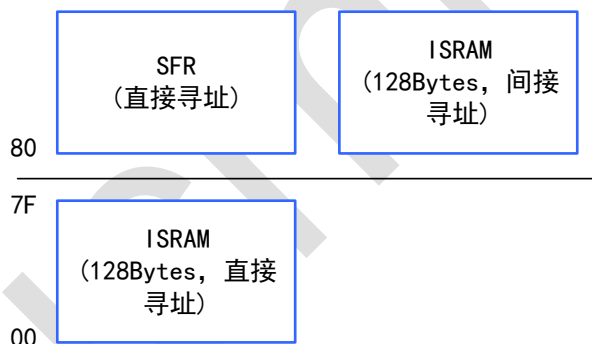
3.1 主要特性

包括内部 RAM(ISRAM)、外部存储空间。外部存储空间中包括用于存放用户程序的程序空间 EFLASH。

3.2 内部 RAM(ISRAM)

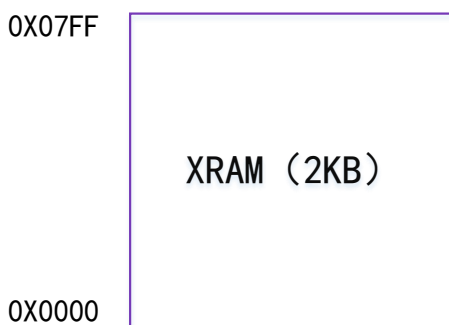
为数据存储提供了 256 个字节的内部 RAM(ISRAM)，通过 MOV 指令访问。ISRAM 分为低 128 字节和高 128 个字节。

- ◆ 低 128 字节的 ISRAM（地址从 00H 到 7FH）可直接或间接寻址
- ◆ 高 128 字节的 ISRAM（地址从 80H 到 FFH）只能间接寻址。高位 128 字节的 ISRAM 占用的地址空间和 SFR 相同，但在物理上与 SFR 的空间是分离的。当一个指令访问高于地址 7FH 的内部位置时，CPU 可以根据访问的指令类型来区分是访问高位 128 字节数据 ISRAM 还是访问 SFR。



3.3 外部存储空间

为数据存储提供了 2K /1K Bytes 字节的外部存储空间 XRAM，可通过 MOVX 指令直接访问。外部存储空间的低 2K Bytes 地址可以是外部 XRAM，或者是 EFLASH 空间，两种占用的地址空间相同，可通过 xdata 关键字访问 XRAM 空间。



外部存储空间其地址映射（低 2K Bytes 为 EFLASH 空间时）如下表所示，用来存储用户程序。

模块名	地址空间	大小
EFLASH	0x0000—0x8000	32K Bytes
BootLoaderShadow	0x8000—0x8600	1.5K Bytes
Reserved	0x8600—0x9000	2.5K Bytes
NVR0	0x9000—0x9200	512Bytes
NVR1	0x9200—0x9400	512Bytes
Reserved	0x9400—0xC000	11K Bytes
IO_IE/LVR_LV/IO_SR	0xC000—0xC400	1K Bytes
SPI	0xC400—0xC800	1K Bytes
LPTIM	0xC800—0xCC00	1K Bytes
I2C	0xCC00—0xD000	1K Bytes
Reserved	0xCC00—0xFFFF	13K Bytes

3.4 程序存储器 EFLASH

EFLASH 的主要特性如下：

- ◆ 32K Bytes 程序空间。
- ◆ Sector 大小：512B
- ◆ Sector 擦除时间：5ms
- ◆ Byte 编程时间：7.5us
- ◆ 擦写次数：20,000 次
- ◆ 数据保存年限：100 年@常温
- ◆ EFLASH 支持 IAP 功能（In Applying Programing）

EFLASH 编程（可通过以下两种方式对 EFLASH 进行读写）：

- 1) IAP（In Applying Programing）模式，用户程序代码可对未使用的 EFLASH 区进行编程，具体参照 EFC 章节。
- 2) 通过系统 Boot 程序，通过串口对 EFLASH 编程。

4 系统配置 (SFR)

4.1 地址映射

系统 SFR 映射表如下：

表 4-1：系统 SFR 映射表

	可位寻址		不可位寻址						
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8H	CLKCON	P0OEN	P1OEN	P2OEN	P2DR	ADCVREF	ADCCDR0	ADCCDR1	FFH
F0H	B	P0PU	P1PU	P2PU	P2CS	ADCHL	ADCCSTAT	ADCSPW	F7H
E8H	-	P0IEN	P1IEN	P2IEN	P2OD	ADCCONT	RCHDIV	-	EFH
E0H	ACC	P0IRQ	P1IRQ	P2IRQ	P2PD	P1DR	PRESET0	PRESET1	E7H
D8H	RSTSTAT	SYSDIV	P1PD	P1OD	P1CS	LVDCON	PCLK0	PCLK1	DFH
D0H	PSW	OADRL	OADRH	ODATA	OCTRL	P0PD	P0OD	P0CS	D7H
C8H	PWM0CON	PWM1CON	PWM2CON	PWM0PL	PWM0PH	PWM1PL	PWM1PH	PWM2DL	CFH
C0H	-	PWM2PL	PWM2PH	PWM0DL	PWM0DH	PWM1DL	PWM1DH	PWM2DH	C7H
B8H	IEN1	IP1	S0RELH	S1RELH	-	LDOTRIMH	RCHTRIMH	RCHTRIML	BFH
B0H	-	CLKST	ESTCR	XTHCTR	ADCGCR2	ADCGCR3	ADCDR0	ADCDR1	B7H
A8H	IEN0	IP0	S0RELL	-	ADCGCR0	ADCGCR1	-	REMAP	AFH
A0H	P2	OUS	OINTEN	OINTUS	-	-	OSTATUS	OPSET	A7H
98H	S0CON	S0BUF	-	S1CON	S1BUF	S1RELL	UARTEN	RCLTRIM	9FH
90H	P1	LDOTRIML	DPS	ADCIER	-	-	-	P0DR	97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1	PDSEL	POREN	8FH
80H	P0	SP	DPL	DPH	DPL1	DPH1	BEEPCTR	PCON	87H
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

4.1.1 P0

80H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0	-	-	-	P0.4	P0.3	P0.2	P0.1	P0.0
读/写	读	读	读	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	1	1	1	1	1

位编号	位符号	说明
7:5	-	-
4	P0.4	当端口配置为输出状态时： 0：输出低电平 1：输出高电平 当端口配置为输入状态时： 读到的是端口状态

3	P0.3	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
2	P0.2	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
1	P0.1	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
0	P0.0	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态

4.1.2 SP

栈指针 SP 是一个 8 位专用寄存器，在执行 PUSH、各种子程序调用、中断响应等指令时，SP 先加 1，再将数据压栈；执行 POP、RET、RETI 等指令时，数据退出堆栈后 SP 再减 1。堆栈栈顶可以是片上内部 RAM (00H-FFH) 的任意地址，系统复位后，SP 初始化为 07H，使得堆栈事实上由 08H 地址开始。

81H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SP	SP							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	1	1	1

位编号	位符号	说明
7:0	SP	栈指针 SP 寄存器。可通过写操作更新 SP 栈指针的值。在执行 PUSH、各种子程序调用、中断响应等指令时，SP 先加 1，再将数据压栈；执行 POP、RET、RETI 等指令时，数据退出堆栈后 SP 再减 1。

4.1.3 DPTR

数据指针 DPTR 是一个 16 位专用寄存器，其高位字节寄存器用 DPH 表示，低位字节寄存器用 DPL 表示。它们既可以作为一个 16 位寄存器 DPTR 来处理，也可以作为 2 个独立的 8 位寄存器 DPH 和 DPL 来处理。包含双数据指针 DPTR&DPTR1，通过 DPS(bit0)寄存器选择。

82H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPL	DPL							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明						
7:0	DPL	数据指针 DPTR0 的低 8 位。						
83H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPH	DPH							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明						
7:0	DPH	数据指针 DPTR0 的高 8 位。						

84H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPL1	DPL1							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明						
7:0	DPL1	数据指针 DPTR1 的低 8 位。						

85H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPH1	DPH1							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明						
7:0	DPH1	数据指针 DPTR1 的高 8 位。						

4.1.4 BEEPCTR

86H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BEEPCTR	-	-	-	BEEPEN	-	BEEPCOLSET	BEEPSEL[1: 0]	
读/写	-	-	-	读/写	-	读/写	读/写	
复位值	-	-	-	0	-	0	0	0

位编号	位符号	说明
7:5	-	-
4	BEEPEN	BEEP 使能控制位： 0: 关闭 BEEP 模块，P2.5 作为 GPIO 功能 1: 打开 BEEP 模块，P2.5 作为 BEEP 输出
3	-	-
2	BEEPCOLSET	BEEP 极性控制： 0: BEEP 输出默认状态下为低电平； 1: BEEP 输出默认状态下为高电平；
1:0	BEEPSEL	BEEP 输出频率控制： 00/11: 1KHz 01: 2KHz 10: 4KHz

4.1.5 PCON

87H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	-	-	-	-	-	-	STOP	IDLE
读/写	读	读	读	读	读	读	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:2	-	-
1	STOP	写 1, 进入 Stop Mode 或 DeepSleep Mode, 读一直返回 0。
0	IDLE	写 1, 进入 Sleep Mode, 读一直返回 0。

4.1.6 TCON

88H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	-	TR1	T1CLKS	TF0	-	TR0	T0CLKS
读/写	读/写	读	读/写	读/写	读/写	读	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	TF1	定时器 1 溢出标志: 0: 无溢出 (必须软件清 0) 1: 溢出 (硬件置位, 软件写 1 无效, 写 0 清 0)
6	-	-
5	TR1	定时器 1 使能位: 0: 停止定时器 1 1: 开始定时器 1
4	T1CLKS	定时器 1 计数时钟选择位: 0: 选择系统时钟作为定时器 1 的计数时钟 1: 选择 RC32K 作为定时器 1 的计数时钟
3	TF0	定时器 0 溢出标志: 0: 无溢出 (必须软件清 0) 1: 溢出 (硬件置位, 软件写 1 无效, 写 0 清 0)
2	-	-
1	TR0	定时器 0 使能位: 0: 停止定时器 0 1: 开始定时器 0
0	T0CLKS	定时器 0 计数时钟选择位: 0: 选择系统时钟作为定时器 0 的计数时钟 1: 选择 RC32K 作为定时器 0 的计数时钟

4.1.7 TMOD

89H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMOD	-	-	T1PS		-	-	T0PS	
读/写	-	-	读/写	读/写	-	-	读/写	读/写
复位值	-	-	0	0	-	-	0	0

位编号	位符号	说明
-----	-----	----

7:6	-	-
5:4	T1PS[1:0]	定时器 1 预分频比选择位: 00: 1/1 01:1/8 10: 1/64 11:1/256
3:2	-	-
1:0	T0PS[1:0]	定时器 0 预分频比选择位: 00: 1/1 01:1/8 10: 1/64 11:1/256

4.1.8 TL0

8AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL0	TL0							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	TL0	Timer0 低 8 位 [7:0]

4.1.9 TL1

8BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL1	TL1							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	TL1	Timer1 低 8 位 [7:0]

4.1.10 TH0

8CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH0	TH0							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	TH0	Timer0 高 8 位 [15:8]

4.1.11 TH1

8DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH1	TH1							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	TH1	Timer1 高 8 位 [15:8]

4.1.12 PDSEL

8EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PDSEL	-	-	-	-	-	-	-	PDSEL
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:1	-	-
0	PDSEL	STOP 模式选择位： 1: STOP 模式使能。当此位为 1，向 PCON 的 STOP 位写 1，将进入 STOP 模式，系统所有时钟源全部关掉； 0: STOP 模式禁止。当此位为 0，向 PCON 的 STOP 位写 1，将进入 Deepsleep 模式，系统 RCL 时钟源在运行，XTH 和 RCH 时钟源关闭；

4.1.13 POREN

8FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
POREN	-	-	-	LVRS				LVR_ENB
读/写	读	读	读	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	1	0

位编号	位符号	说明																																				
7:5	-	-																																				
4:1	LVRS	LVR 检测点电压设置： <table border="1"> <thead> <tr> <th>LVRS</th> <th>LVR point</th> <th>LVRS</th> <th>LVR point</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>1.65V</td> <td>1000</td> <td>2.45V</td> </tr> <tr> <td>0001</td> <td>1.75V</td> <td>1001</td> <td>2.55V</td> </tr> <tr> <td>0010</td> <td>1.85V</td> <td>1010</td> <td>2.65V</td> </tr> <tr> <td>0011</td> <td>1.95V</td> <td>1011</td> <td>2.75V</td> </tr> <tr> <td>0100</td> <td>2.05V</td> <td>1100</td> <td>2.85V</td> </tr> <tr> <td>0101</td> <td>2.15V</td> <td>1101</td> <td>2.95V</td> </tr> <tr> <td>0110</td> <td>2.25V</td> <td>1110</td> <td>3.05V</td> </tr> <tr> <td>0111</td> <td>2.35V</td> <td>1111</td> <td>3.15V</td> </tr> </tbody> </table>	LVRS	LVR point	LVRS	LVR point	0000	1.65V	1000	2.45V	0001	1.75V	1001	2.55V	0010	1.85V	1010	2.65V	0011	1.95V	1011	2.75V	0100	2.05V	1100	2.85V	0101	2.15V	1101	2.95V	0110	2.25V	1110	3.05V	0111	2.35V	1111	3.15V
LVRS	LVR point	LVRS	LVR point																																			
0000	1.65V	1000	2.45V																																			
0001	1.75V	1001	2.55V																																			
0010	1.85V	1010	2.65V																																			
0011	1.95V	1011	2.75V																																			
0100	2.05V	1100	2.85V																																			
0101	2.15V	1101	2.95V																																			
0110	2.25V	1110	3.05V																																			
0111	2.35V	1111	3.15V																																			
0	LVR_ENB	掉电复位使能信号： 0: 掉电复位使能； 1: 关闭掉电复位功能；																																				

4.1.14 P1

90H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1	-	-	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

读/写	-	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值	-	-	1	1	1	1	1	1

位编号	位符号	说明
7:6	-	-
5	P1.5	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
4	P1.4	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
3	P1.3	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
2	P1.2	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
1	P1.1	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
0	P1.0	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态

4.1.15 LDOTRIML

91H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LDOTRIML	-	-	VADJ_STANDBY		VTRM_STANDBY			
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	1	1	0	1	1	1

位编号	位符号	说明
7:6	-	-
5:4	VADJ_STANDBY	STANDBY 模式下电压中心值设置
3:0	VTRM_STANDBY	STANDBY 模式下 TRIM 值

4.1.16 DPS

92H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPS	-	-	-	-	-	-	-	DPS
读/写	读	读	读	读	读	读	读	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:1	-	-
0	DPS	1: 数据指针选择 DPTR1; 0: 数据指针选择 DPTR0。

4.1.17 ADCIER

93H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCIER	-	-	-	-	-	-	-	RXINTEN
读/写	读	读	读	读	读	读	读	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:1	-	-
0	RXINTEN	接收器 BUF 存有效数据中断使能位 1: 启用中断 0: 未启用中断

4.1.18 P0DR

97H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0DR	-	-	-	P0_4DR	P0_3DR	P0_2DR	P0_1DR	P0_0DR
读/写	读	读	读	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:5	-	-
4	P0_4DR	端口 P0_4 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
3	P0_3DR	端口 P0_3 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
2	P0_2DR	端口 P0_2 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
1	P0_1DR	端口 P0_1 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力

0	P0_ODR	端口 P0_0 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
---	--------	---

4.1.19 S0CON

98H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0CON	SM0	SM1	SM20	RENO	TB80	RB80	TI0	RI0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明				
7	SM0	UART0 模式选择控制位:				
6	SM1	SM0	SM1	MODE	描述	Baud Rate
		0	0	0	Shift register	SYSCCLK/12
		0	1	1	8-bit UART	可配置
		1	0	2	9-bit UART	SYSCCLK/16
1	1	3	9-bit UART	可配置		
5	SM20	UART0 多机通讯使能控制位: 0: 关闭多机通讯功能, 1: 使能多机通讯功能只接收 RB80 = 1 的 9 位数据				
4	RENO	UART0 接收允许使能控制位: 0: 不允许串口 0 接收 1: 允许串口 0 数据接收				
3	TB80	UART0 Mode2, 3 第 9 位发送数据位				
2	RB80	UART0 Mode2, 3 第 9 位接收数据位				
1	TI0	UART0 发送中断标志位: 当发送数据完成后, 硬件置位, 必须由软件清 0				
0	RI0	UART0 接收中断标志位: 当完成一次数据接收, 硬件置位, 必须由软件清 0				

4.1.20 S0BUF

99H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0BUF	S0BUF.7	S0BUF.6	S0BUF.5	S0BUF.4	S0BUF.3	S0BUF.2	S0BUF.1	S0BUF.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	S0BUF[7:0]	UART0 数据寄存器: 读 S0BUF 返回 UART0 接收到的数据 写 S0BUF 启动 UART0 数据发送

4.1.21 S1CON

9BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1CON	SM	-	SM21	REN1	TB81	RB81	TI1	RI1
读/写	读/写	-	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明			
7	SM	UART1 模式选择控制寄存器:			
		SM	Mode	描述	Baud Rate
		0	A	9-bit UART	可配置
		1	B	8-bit UART	可配置
6	-	-			
5	SM21	UART1 多机通讯使能控制位: 0: 关闭多级通讯功能 1: 使能多级通讯功能, 只接收 RB81= 1 的 9 位数据			
4	REN1	UART1 接收允许使能控制位: 0: 不允许串口 1 接收 1: 允许串口 1 数据接收			
3	TB81	UART1 第 9 位发送数据位			
2	RB81	UART1 第 9 位接收数据位			
1	TI1	UART1 发送中断标志位: 当发送数据完成后, 硬件置位, 必须由软件清 0			
0	RI1	UART1 接收中断标志位: 当完成一次数据接收, 硬件置位, 必须由软件清 0			

4.1.22 S1BUF

9CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1BUF	S1BUF.7	S1BUF.6	S1BUF.5	S1BUF.4	S1BUF.3	S1BUF.2	S1BUF.1	S1BUF.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	S1BUF[7:0]	UART1 数据寄存器: 读 S1BUF 返回 UART1 接收到的数据 写 S1BUF 启动 UART1 数据发送

4.1.23 S1REL

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S1RELL(9DH)	S1RELL.7	S1RELL.6	S1RELL.5	S1RELL.4	S1RELL.3	S1RELL.2	S1RELL.1	S1RELL.0
复位值	1	1	1	0	0	1	1	0

S1RELH(BBH)	-	-	-	-	-	-	S1RELH.1	S1RELH.0
复位值	-	-	-	-	-	-	1	1
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位编号	位符号	说明
7:0 1:0	S1RELL[7:0] S1REH[1:0]	UART0 Mode 波特率配置寄存器: S1RELH +S1RELL 构成波特率设置 S1REL[9:0] $\text{Baud Rate} = \frac{\text{SYSCK}}{16 \times (1024 - \text{S1REL})}$

4.1.24 UARTEN

9EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UARTEN	OUTSEL		OUTEN	-	-	-	UART1EN	UART0EN
读/写	读/写	读/写	读/写	读	读	读	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:6	OUTSEL	11: XTH 输出; 10: 系统时钟输出; 01: RCL32K 时钟输出; 00: RCHDIV 时钟输出
5	OUTEN	CLK 输出使能; 1: P0_3 作为 CLKOUT 输出功能; 0: P0_3 不作为 CLKOUT 输出功能
4:2	-	-
1	UART1EN	UART1 使能控制位: 0: 关闭 UART1 功能, P1.4, P1.5 作为 GPIO 功能 1: 打开 UART1 功能, P1.4, P1.5 作为 UART1 RX1, TX1 功能
0	UART0EN	UART0 使能控制位: 0: 关闭 UART0 功能, P2.6, P2.7 作为 GPIO 功能 1: 打开 UART0 功能, P2.6, P2.7 作为 UART0 TX0, RX0 功能

4.1.25 RCLTRIM

9FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCLTRIM	-	RTRIM		S				
读/写	读	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	1	0	1	1	1	1

位编号	位符号	说明
7	-	-
6:5	RTRIM	片内低频 RCL 修调值
4:0	S	片内低频 RCL 修调值

4.1.26 P2

A0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2	P2.7	P2.6	P2.5	-	P2.3	P2.2	-	P2.0
读/写	读/写	读/写	读/写	读	读/写	读/写	读	读/写
复位值	1	1	1	0	1	1	0	1

位编号	位符号	说明
7	P2.7	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
6	P2.6	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
5	P2.5	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
4	-	-
3	P2.3	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
2	P2.2	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态
1	-	-
0	P2.0	当端口配置为输出状态时: 0: 输出低电平 1: 输出高电平 当端口配置为输入状态时: 读到的是端口状态

4.1.27 OUS

A1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OUS	-	-	-	OUS				
读/写	-	-	-	读/写	读/写	读/写	读/写	读/写
复位值	-	-	-	0	0	1	1	1

位编号	位符号	说明
7:5	-	-
4:0	OUS	Flash 擦写时间标尺。 此寄存器的值应为: (系统时钟频率/2)-1; 如, 系统运行在 16MHz, 此寄存器的值应为 7。

4.1.28 OINTEN

A2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OINTEN	-	-	VDDLLOWEN	WPOGERREN	-	-	-	PRODONEEN
读/写	读	读	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:6	-	-
5	VDDLLOWEN	1: LVD 低电压中断使能 0: LVD 低电压中断禁止
4	WPOGERREN	1: 操作错误中断使能 0: 操作错误中断禁止
3	-	-
2	-	-
1	-	-
0	PRODONEEN	1: 擦写完成中断使能 0: 擦写完成中断禁止

4.1.29 OINTUS

A3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OINTUS	-	-	VDD_LOW	WPOGERR	-	-	-	PRODONE
读/写	读	读	读	读	读	读	读	读
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:6	-	-
5	VDD_LOW	1: LVD 低电压中断状态位 0: LVD 低电压中断状态位 写 1 清 0
4	WPOGERR	1: 操作错误中断状态位 0: 操作错误中断状态位 写 1 清 0
3	-	-
2	-	-
1	-	-

0	PRODONE	1: 擦写完成中断状态位 0: 擦写完成中断状态位 写 1 清 0
---	---------	---

4.1.30 OSTATUS

A6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSTATUS	-				LVDLOW	-	-	EFCREADY
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	1

位编号	位符号	说明
7:4	-	-
3	LVDLOW	0: LVD检测电压正常; 1: LVD检测电压过低。 此位反应当前LVD检测的实时状态。
2	-	-
1	-	-
0	EFCREADY	EFlash状态指示位。该反映EFlash工作的状态。 1: EFlash状态空闲; 0: EFlash状态忙。

4.1.31 OPSET

A7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPSET	READM1	READM0	RDWAIT		CHIPSERSET	PAGESERSET	PAGEWRSET	
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	1

位编号	位符号	说明
7	READM1	此位不可写 1, 保留为 0。
6	READM0	此位不可写 1, 保留为 0。
5:3	RDWAIT	读等待时间设置 (系统时钟跑 24MHz, 读等待时间需设置为 1)
2	CHIPSERSET	1: CHIP 擦除模式使能 0: CHIP 擦除模式关闭
1	PAGESERSET	1: PAGE 擦除模式使能 0: PAGE 擦除模式关闭
0	PAGEWRSET	1: PAGE 写模式使能; 0: PAGE 写模式关闭;

4.1.32 IEN0

A8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EA	EADC	EPWM	ES0	ET1	ES1	ET0	EX0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

复位值	0	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

位编号	位符号	说明
7	EA	总中断开关： 0: 关闭中断 1: 打开中断
6	EADC	ADC 中断使能开关： 0: 关闭中断 1: 打开中断
5	EPWM	PWM 中断使能开关： 0: 关闭中断 1: 打开中断
4	ES0	UART0 中断使能开关： 0: 关闭中断 1: 打开中断
3	ET1	Timer1 中断使能开关： 0: 关闭中断 1: 打开中断
2	ES1	UART1 中断使能开关： 0: 关闭中断 1: 打开中断
1	ET0	Timer0 中断使能开关： 0: 关闭中断 1: 打开中断
0	EX0	外部中断初级使能开关： 0: 关闭中断 1: 打开中断

4.1.33 IP

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP0(A9H)	-	-	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0
IP1(B9H)	-	-	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0
读/写	读	读	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

每个中断源都可被单独设置为 4 个中断优先级之一，分别通过 IP0 和 IP1 的相应位来设置实现，具体设置见下表：

IP1.x	IP0.x	PriorityLevel
0	0	Level0(lowest)
0	1	Level1
1	0	Level2
1	1	Level3(highest)

Bit	对应中断
IP1.0, IP0.0	外部中断 0EX0 和 ADC 中断
IP1.1, IP0.1	定时器 0 中断 ET0 和 SPI 中断
IP1.2, IP0.2	串口 1 中断 ES1
IP1.3, IP0.3	定时器 1 中断 ET1 和 EFC 中断

IP1.4, 1P0.4	串口 0 中断 ES0 和 LPTIMER 中断
IP1.5, 1P0.5	PWM 中断 EPWM 和 I2C 中断

4.1.34 S0REL

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
S0RELL(AAH)	S0RELL.7	S0RELL.6	S0RELL.5	S0RELL.4	S0RELL.3	S0RELL.2	S0RELL.1	S0RELL.0
复位值	1	1	1	0	0	1	1	0
S0RELH(BAH)	-	-	-	-	-	-	S0RELH.1	S0RELH.0
复位值	-	-	-	-	-	-	1	1
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写

位编号	位符号	说明
7-0 1-0	S0RELL[7:0] S0RELH[1:0]	UART0 Mode1, 3 波特率配置寄存器: S0RELH + S0RELL 构成波特率设置 S0REL[9:0] $\text{Baud Rate} = \frac{\text{SYSCK}}{16 \times (1024 - \text{S0REL})}$

4.1.35 ADCGCR0

ACH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCGCR0	-	ADCCLKSEL	ADRCEN	-	-	-	ADCCUNSET	ADCEN
读/写	读	读/写	读/写	读	读	读	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	-	-
6	ADCCLKSEL	ADC 时钟源选择信号。 0: ADC 内部时钟分频器产生的时钟 1: 系统时钟发生器产生的时钟 (与之反相)
5	ADRCEN	ADC 数据寄存器 (ADCDRx) 读清除使能。 0: 禁止读取 ADC 数据寄存器的清除 1: 使能读取清除 ADC 数据寄存器
4:2	-	-
1	ADCCUNSET	连续模式设置位 1: ADC 工作在连续模式 0: ADC 工作在单次模式
0	ADCEN	ADC 控制器使能信号 0: 模块禁用 1: 模块启用

4.1.36 ADCGCR1

ADH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCGCR1	-	-	-	-	-	ADCSTEN	ADCRST	ADCPDEN
读/写	读	读	读	读	读	读/写	读/写	读/写

复位值	0	0	0	0	0	0	1	1
-----	---	---	---	---	---	---	---	---

位编号	位符号	说明
7:3	-	-
2	ADCSTEN	ADC 转换启动使能信号。当信号具有从低到高的转换时, ADC 转换开始。当信号具有从高到低的转换时, ADC 转换操作完成。当 ADCEN=0 时, 它将被清除。默认值: 0
1	ADCRST	ADC 内部数字逻辑复位信号。 1: SAR ADC 复位 0: SAR ADC 释放
0	ADCPDEN	SAR ADC 掉电使能信号。 0: SAR ADC 上电 (power on) 1: SAR ADC 掉电 (power down)

4.1.37 REMAP

AFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
REMAP	-	-	-	-	-	REMAP_FLAG	REMAP_IM	REMAP
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	1	1

位编号	位符号	说明
7:3	-	-
2	REMAP_FLAG	系统是否发生 REMAP 的标志。 1: 系统发生 REMAP; 0: 系统未发生 REMAP
1	REMAP_IM	写 0, 地址直接发生 REMAP。
0	REMAP	写 0, 地址发生 REMAP, 并产生系统复位, 复位后从 eFlash 的 main 区启动。

4.1.38 CLKST

B1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKST	WACKDELAY		XTHSTAB		RCHSTAB		RCLSTAB	
读写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	1	0	0	0	0	0	0	0

位编号	位符号	说明
7:6	WACKDELAY	唤醒时间延时设置: 00: 64 个系统时钟周期; 01: 128 个系统时钟周期; 10: 245 个系统时钟周期; 11: 384 个系统时钟周期;

5:4	XTHSTAB	XTH 稳定时间设定 00: 4096 个 XTH 时钟周期; 01: 16384 个 XTH 时钟周期; 10: 32768 个 XTH 时钟周期; 11: 65535 个 XTH 时钟周期。
3:2	RCHSTAB	RCH 稳定时间设定 00: 1 个 RCH32K 时钟周期; 01: 4 个 RCH32K 时钟周期; 10: 32 个 RCH32K 时钟周期; 11: 256 个 RCH32K 时钟周期。
1:0	RCLSTAB	RCL 稳定时间设定 00: 1 个 RCL32K 时钟周期; 01: 4 个 RCL32K 时钟周期; 10: 32 个 RCL32K 时钟周期; 11: 256 个 RCL32K 时钟周期。

4.1.39 ESTCR

B2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ESTCR	-	-	-	-	ERSTEN	ERSTLVT		ERSTLVEN
读/写	读	读	读	读	读/写	读/写	读/写	读/写
复位值	0	0	0	0	1	0	0	0

位编号	位符号	说明
7:4	-	-
3	ERSTEN	外部复位引脚使能位。 1: 外部复位功能使能; 0: 外部复位功能禁止;
2:1	ERSTLVT	外部复位滤波时间设置。 11: 1 个 RCL32K 时钟周期; 10: 8 个 RCL32K 时钟周期; 01: 16 个 RCL32K 时钟周期; 00: 29 个 RCL32K 时钟周期;
0	ERSTLVEN	1: 外部复位滤波使能; 0: 外部复位滤波禁止;

4.1.40 XTHCTR

B3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XTHCTR	-	-	EXTH_RSEL		EXTH_GSEL			EXTH_EN
读/写	读	读	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	1	0	0	1	1	0

位编号	位符号	说明
7:6	-	-
5:4	EXTH_RSEL	XTH 修调位

3:1	EXTH_GSEL	XTH 修调位
0	EXTH_EN	外部 XTH 时钟输入控制 1: XTH 输出时钟从 P0_0 输入 0: XTH 输出时钟由晶振产生 注: 使用 P0_0 输入时钟时, 需设置 XCLKEN 为 1。
	Crystal Frequency	GSEL[2:0>
	F≤1MHz	000
	1MHz< F ≤6MHz	001
	6MHz< F ≤12MHz	010
	12MHz< F ≤16MHz	011
	16MHz< F ≤20MHz	101
	20MHz< F ≤24MHz	110

4.1.41 ADCGCR2

B4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCGCR2	-	-	-	-	CHEN			
读/写	读	读	读	读	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:4	-	-
3:0	CHEN	启用相关 ADC 通道进行模数转换。默认值: 0。 0000: 关闭通道 0001: 通道 0 0010: 通道 1 0011: 通道 2 0100: 通道 3 0101: 通道 4 0110: 通道 5 0111: 通道 6 1000: 通道 7 (输入来源固定为内部 LDO)

4.1.42 ADCGCR3

B5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCGCR3	-	-	-	-	-	-	-	SAMPNEG
读/写	读	读	读	读	读	读	读	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:1	-	-

0	SAMPNEG	ADC 数据在 EOC 信号的边沿采样选择： 0: ADC 数据在 EOC 的上升沿被采样 1: ADC 数据在 EOC 的下降沿被采样 注：在本芯片设计中此位只能设置为 0。
---	---------	---

4.1.43 ADCDR0

B6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCDR0	CHDATAL							
读/写	读	读	读	读	读	读	读	读
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	CHDATAL	A/D 通道接收数据低位寄存器。

4.1.44 ADCDR1

B7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCDR1	CHDATAV	-	-	-	CHDATAH			
读/写	读	读	读	读	读	读	读	读
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	CHDATAV	数据有效信号。信号在获取有效数据后激活，在 ADCEN = 0 时清除，或在 ADCRCEN = 1 时由软件读取 ADCDR0/1 寄存器后清除。 0: 数据无效 1: 数据有效
6:4	-	-
3:0	CHDATAH	A/D 通道接收数据高位寄存器。

4.1.45 IEN1

B8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN1	-	-	I2CINTEN	LPTIMINTEN	EFCINTEN	-	SPIINTEN	-
读/写	读	读	读/写	读/写	读/写	读	读/写	读
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:6	-	-
5	I2CINTEN	I2C 中断使能。 1: I2C 中断使能 0: I2C 中断关闭

4	LPTIMINTEN	LPTIMER 中断使能。 1: LPTIMER 中断使能 0: LPTIMER 中断关闭
3	EFCINTEN	EFC 中断使能。 1: EFC 中断使能 0: EFC 中断关闭
2	-	-
1	SPIINTEN	SPI 中断使能。 1: SPI 中断使能 0: SPI 中断关闭
0	-	-

4.1.46 LDOTRIMH

BDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LDOTRIMH	-	VADJ_ACT		VTRM_ACT				
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	1	1	1	0	0	0	0

位编号	位符号	说明
7	-	-
6:5	VADJ_ACT	ACTIVE 模式下电压中心值设置
4:0	VTRM_ACT	ACTIVE 模式下 TRIM 值

4.1.47 RCHTRIMH

BEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCHTRIMH	RCHTRIMH							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	1	1	0	1	1

位编号	位符号	说明
7:0	RCHTRIMH	片内高频 RCH 频率修调值高位

4.1.48 RCHTRIML

BFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCHTRIML	RCHTRIML						RCHTRIML	
读/写	读	读	读	读	读	读	读/写	读/写
复位值	0	0	0	0	0	0	1	1

位编号	位符号	说明
7:2	-	-
1:0	RCHTRIML	片内高频 RCH 频率修调值低位

4.1.49 PWMxPL/H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0PL(CBH)	PWM0P[7:0]							
PWM0PH(CCH)	PWM0P[15:8]							
PWM1PL(CDH)	PWM1P[7:0]							
PWM1PH(CEH)	PWM1P[15:8]							
PWM2PL(C1H)	PWM2P[7:0]							
PWM2PH(C2H)	PWM2P[15:8]							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
15:0	PWMxP[15:0] (x=0, 1, 2)	PWMx 数据寄存器

注意：修改寄存器 PWMxPH 将使得 PWMx 的输出在下一个周期生效。如果用户需要修改 PWM 周期，先要修改 PWMxPL，再修改 PWMxPH。

4.1.50 PWMxDL/H

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0DL(C3H)	PWM0D[7:0]							
PWM0DH(C4H)	PWM0D[15:8]							
PWM1DL(C5H)	PWM1D[7:0]							
PWM1DH(C6H)	PWM1D[15:8]							
PWM2DL(CFH)	PWM2D[7:0]							
PWM2DH(C7H)	PWM2D[15:8]							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
15:0	PWMxD[15:0] (x=0, 1, 2)	PWMx占空比控制，控制PWM0波形占空比的输出时间 详细 PWM 输出时序，见下一节 12.3 PWM 输出时序

注意：修改寄存器 PWMxDH 将使得 PWMx 的输出在下一个周期生效。如果用户需要修改 PWM 占空比，要先修改 PWMxDL，再修改 PWMxDH。

4.1.51 PWMxCON

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0CON(C8H)	PWM0EN	PWM0S	-	-	-	PWM0IE	PWM0IF	PWM0SS
PWM1CON(C9H)	PWM1EN	PWM1S	-	-	-	PWM1IE	PWM1IF	PWM1SS
PWM2CON(CAH)	PWM2EN	PWM2S	-	-	-	PWM2IE	PWM2IF	PWM2SS
读/写	读/写	读/写	-	-	-	读/写	读/写	读/写
复位值	0	0	-	-	-	0	0	0

位编号	位符号	说明
7	PWMxEN	PWMx 使能： 0：禁止 PWMx 模块 1：使能 PWMx 模块

6	PWMxS	PWMx 输出模式: 0: PWMx 占空比期间输出高电平, 占空比溢出后输出低电平 1: PWMx 占空比期间输出低电平, 占空比溢出后输出高电平
5:3	-	-
2	PWMxIE	PWMx 中断使能控制位: 0: 禁止 PWMx 中断 1: 允许 PWMx 中断 3 路 PWM 共用中断入口地址
1	PWMxIF	PWMx 中断标志位: 0: PWM 周期计数器没有溢出 1: PWMx 周期计数器溢出, 硬件置位, 软件写 1 无效, 写 0 清 0
0	PWMxSS	PWMx 引脚输出控制位: 0: PWMx 输出禁止, 用作 I/O 功能 1: PWMx 输出允许

4.1.52 PSW

DOH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	F0	RS1	RS0	OV	F1	P
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	CY	进位标志: 当最后一次算术操作产生进位 (加法) 或借位 (减法) 时, 该位置 1, 其它算术操作将其清 0。
6	AC	辅助进位标志: 当最后一次算术操作向高半字节有进位 (加法) 或借位 (减法) 时, 该位置 1, 其它算术操作将其清 0。
5	F0	用户标志 0: 这是一个可位寻址、用于软件控制的通用标志位。
4:3	RS[1:0]	RS1-RS0: 寄存器区选择: 00: 页 0 (映射到 00H-07H) 01: 页 1 (映射到 08H-0FH) 10: 页 2 (映射到 10H-17H) 11: 页 3 (映射到 18H-1FH)
1	F1	用户标志 1: 这是一个可位寻址、用于软件控制的通用标志位
0	P	奇偶校验位: 0: 累加器中 8 个位的和为偶数 1: 累加器中 8 个位的和为奇数

4.1.53 OADRL/H

D1H~D2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OADRL	EFLASH 烧录地址低位[7:0]							
OADRH	EFLASH 烧录地址低位[15:8]							
读/写	读/写							
复位值	1	1	1	1	1	1	1	1

4.1.54 ODATA

D3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ODATA	EFLASH 烧录数据[7:0]							
读/写	读/写							
复位值	1	1	1	1	1	1	1	1

4.1.55 OCTRL

D4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OCTRL	VPPO_EN	PUMP_EN	PUMP_SEL<2:0>		-	PUMP_OK	PUMP_6O5V	
读/写	读/写	读/写	读/写		-	读	读	
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明																				
7	VPPO_EN	EFLASH IAP 编程启动控制信号： 0: 关闭（硬件自动清 0） 1: 启动 EFLASH 编程（软件写 1 启动 EFLASH IAP 编程，硬件完成 IAP 后清 0） 此 bit 置位后，CPU 将处在 Idle 状态，待 IAP 完成后恢复																				
6	PUMP_EN	PUMP 模块使能控制位： 0: 关闭内部 PUMP 功能； 1: 打开内部 PUMP 功能；																				
5:3	PUMP_SEL<2:0>	Pump 输出电压选择控制位： <table border="1"> <thead> <tr> <th>PUMP_SEL</th> <th>VPP(V)</th> <th>PUMP_SEL</th> <th>VPP(V)</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>6.5</td> <td>100</td> <td>7.5</td> </tr> <tr> <td>001</td> <td>6.75</td> <td>101</td> <td>7.75</td> </tr> <tr> <td>010</td> <td>7.0</td> <td>110</td> <td>8</td> </tr> <tr> <td>011</td> <td>7.25</td> <td>111</td> <td>8.25</td> </tr> </tbody> </table> 建议实际 IAP 编程设置 PUMP_SEL<2:0> = 001, 6.75V	PUMP_SEL	VPP(V)	PUMP_SEL	VPP(V)	000	6.5	100	7.5	001	6.75	101	7.75	010	7.0	110	8	011	7.25	111	8.25
PUMP_SEL	VPP(V)	PUMP_SEL	VPP(V)																			
000	6.5	100	7.5																			
001	6.75	101	7.75																			
010	7.0	110	8																			
011	7.25	111	8.25																			
2	-	-																				
1	PUMP_OK	VPP 输出电压标志 1： 0: VPP OUT 电压低于 PUMP_SEL 设定电压 1: VPP OUT 电压高于 PUMP_SEL 设定电压																				

0	PUMP_6O5V	VPP 输出电压标志 2: 1: VPP OUT 电压高于 6.5V 0: VPP OUT 电压低于 6.5V
---	-----------	---

4.1.56 P0PD

D5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0PD	-	-	-	P0_4PD	P0_3PD	P0_2PD	P0_1PD	P0_0PD
读/写	读	读	读	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:5	-	-
4	P0_4PD	端口 P0_4 下拉配置寄存器 0: 禁止 1: 使能
3	P0_3PD	端口 P0_3 下拉配置寄存器 0: 禁止 1: 使能
2	P0_2PD	端口 P0_2 下拉配置寄存器 0: 禁止 1: 使能
1	P0_1PD	端口 P0_1 下拉配置寄存器 0: 禁止 1: 使能
0	P0_0PD	端口 P0_0 下拉配置寄存器 0: 禁止 1: 使能

4.1.57 P0OD

D6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0OD	-	-	-	P0_4OD	P0_3OD	P0_2OD	P0_1OD	P0_0OD
读/写	读	读	读	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:5	-	-
4	P0_4OD	端口 P0_4 开漏输出配置寄存器 0: 禁止 1: 使能
3	P0_3OD	端口 P0_3 开漏输出配置寄存器 0: 禁止 1: 使能
2	P0_2OD	端口 P0_2 开漏输出配置寄存器 0: 禁止 1: 使能
1	P0_1OD	端口 P0_1 开漏输出配置寄存器 0: 禁止 1: 使能
0	P0_0OD	端口 P0_0 开漏输出配置寄存器 0: 禁止 1: 使能

4.1.58 P0CS

D7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0CS	-	-	-	P0_4CS	P0_3CS	P0_2CS	P0_1CS	P0_0CS
读/写	读	读	读	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	1	1	1	1	1

位编号	位符号	说明
7:5	-	-
4	P0_4CS	端口 P0_4 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
3	P0_3CS	端口 P0_3 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
2	P0_2CS	端口 P0_2 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
1	P0_1CS	端口 P0_1 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
0	P0_0CS	端口 P0_0 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer

4.1.59 RSTSTAT

D8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RSTSTAT	WDOF	WDEN	LVDRF	PORF	ERSTF	WDT[2:0]		
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值(POR)	0	0	x	1	x	0	0	0
复位值(WDT)	1	0	x	x	x	0	0	0
复位值(PIN)	x	0	x	x	1	0	0	0
复位值(LVD)	x	0	1	x	x	0	0	0

位编号	位符号	说明
7	WDOF	看门狗溢出标志位： 看门狗溢出时由硬件置 1，可由软件或上电复位清 0 0: 未发生 WDT 溢出 1: 发生 WDT 溢出
6	WDEN	看门狗使能控制位： 0: 关闭看门狗功能 1: 使能看门狗功能
5	LVDRF	LVD 复位标志位： LVD 复位后由硬件置 1，由软件清 0 0: 没有发生低电压复位 1: 发生过低电压复位
4	PORF	上电复位标志位： 上电复位后硬件置 1，由软件清 0 0: 没有发生上电复位 1: 发生过上电复位

3	ERSTF	Reset 引脚复位标志位： 引脚复位后置 1，由软件清 0 0：没有发生引脚复位 1：发生过引脚复位
2:0	WDT[2:0]	WDT 溢出周期控制位： 000：溢出周期最小值= 4096ms 001：溢出周期最小值= 1024ms 010：溢出周期最小值= 256ms 011：溢出周期最小值= 128ms 100：溢出周期最小值= 64ms 101：溢出周期最小值= 16ms 110：溢出周期最小值= 4ms 111：溢出周期最小值= 1ms

4.1.60 SYSDIV

D9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SYSDIV	-	-	-	-	-	SYSDIV[2]	SYSDIV[1]	SYSDIV[0]
读/写	读	读	读	读	读	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:3	-	-
2:0	SYSDIV[1:0]	高频时钟(经过 RCHDIV 分频后 RCH48M 或者 XCLK)分频控制，分频后输出作为系统时钟 000：HSCLK 不分频 001：HSCLK/2 分频输出 010：HSCLK /4 分频输出 011：HSCLK /8 分频输出 100：HSCLK /16 分频输出 101：HSCLK /32 分频输出 110：HSCLK /64 分频输出 111：HSCLK /128 分频输出

4.1.61 P1PD

DAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1PD	-	-	P1_5PD	P1_4PD	P1_3PD	P1_2PD	P1_1PD	P1_0PD
读/写	读	读	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:6	-	-
5	P1_5PD	端口 P1_5 下拉配置寄存器 0：禁止 1：使能
4	P1_4PD	端口 P1_4 下拉配置寄存器 0：禁止 1：使能

3	P1_3PD	端口 P1_3 下拉配置寄存器 0: 禁止 1: 使能
2	P1_2PD	端口 P1_2 下拉配置寄存器 0: 禁止 1: 使能
1	P1_1PD	端口 P1_1 下拉配置寄存器 0: 禁止 1: 使能
0	P1_0PD	端口 P1_0 下拉配置寄存器 0: 禁止 1: 使能

4.1.62 P1OD

DBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1OD	-	-	P1_5OD	P1_4OD	P1_3OD	P1_2OD	P1_1OD	P1_0OD
读/写	读	读	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:6	-	-
5	P1_5OD	端口 P1_5 开漏输出配置寄存器 0: 禁止 1: 使能
4	P1_4OD	端口 P1_4 开漏输出配置寄存器 0: 禁止 1: 使能
3	P1_3OD	端口 P1_3 开漏输出配置寄存器 0: 禁止 1: 使能
2	P1_2OD	端口 P1_2 开漏输出配置寄存器 0: 禁止 1: 使能
1	P1_1OD	端口 P1_1 开漏输出配置寄存器 0: 禁止 1: 使能
0	P1_0OD	端口 P1_0 开漏输出配置寄存器 0: 禁止 1: 使能

4.1.63 P1CS

DCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1CS	-	-	P1_5CS	P1_4CS	P1_3CS	P1_2CS	P1_1CS	P1_0CS
读/写	读	读	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	1	1	1	1	1	1

位编号	位符号	说明
7:6	-	-
5	P1_5CS	端口 P1_5 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
4	P1_4CS	端口 P1_4 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer

3	P1_3CS	端口 P1_3 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
2	P1_2CS	端口 P1_2 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
1	P1_1CS	端口 P1_1 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
0	P1_0CS	端口 P1_0 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer

4.1.64 LVDFCON

DDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDFCON	LVDF	-	LVDRN	LVDFEN	LVDS[3: 0]			
读写	读	读	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	1	0	0	0

位编号	位符号	说明																																				
7	LVDF	LVD 检测标志位: 0: 未检测到低电压 1: 检测到低电压																																				
6	-	-																																				
5	LVDRN	LVD 复位使能控制: 0: 关闭检测到低电压复位功能 1: 使能检测到低电压复位功能, 使能复位前必须使能 LVDFEN																																				
4	LVDFEN	LVD 模块使能控制: 0: 使能 LVD 模块 1: 关闭 LVD 模块																																				
3:0	LVDS[3:0]	LVD 检测点电压设置: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>LVDS</th> <th>LVD point</th> <th>LVDS</th> <th>LVD point</th> </tr> </thead> <tbody> <tr><td>0000</td><td>1.65V</td><td>1000</td><td>2.45V</td></tr> <tr><td>0001</td><td>1.75V</td><td>1001</td><td>2.55V</td></tr> <tr><td>0010</td><td>1.85V</td><td>1010</td><td>2.65V</td></tr> <tr><td>0011</td><td>1.95V</td><td>1011</td><td>2.75V</td></tr> <tr><td>0100</td><td>2.05V</td><td>1100</td><td>2.85V</td></tr> <tr><td>0101</td><td>2.15V</td><td>1101</td><td>2.95V</td></tr> <tr><td>0110</td><td>2.25V</td><td>1110</td><td>3.05V</td></tr> <tr><td>0111</td><td>2.35V</td><td>1111</td><td>3.15V</td></tr> </tbody> </table>	LVDS	LVD point	LVDS	LVD point	0000	1.65V	1000	2.45V	0001	1.75V	1001	2.55V	0010	1.85V	1010	2.65V	0011	1.95V	1011	2.75V	0100	2.05V	1100	2.85V	0101	2.15V	1101	2.95V	0110	2.25V	1110	3.05V	0111	2.35V	1111	3.15V
LVDS	LVD point	LVDS	LVD point																																			
0000	1.65V	1000	2.45V																																			
0001	1.75V	1001	2.55V																																			
0010	1.85V	1010	2.65V																																			
0011	1.95V	1011	2.75V																																			
0100	2.05V	1100	2.85V																																			
0101	2.15V	1101	2.95V																																			
0110	2.25V	1110	3.05V																																			
0111	2.35V	1111	3.15V																																			

4.1.65 PCLK0

DEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCLK0	I2CCEN	LPTIMCEN	PWMCEN	ADCCEN	SPICEN	WDTCCEN	UART1CEN	UART0CEN
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7	I2CCEN	1: I2C 时钟使能 0: I2C 时钟禁止
6	LPTIMCEN	1: LPTIMER 时钟使能 0: LPTIMER 时钟禁止
5	PWMCEN	1: PWM 时钟使能 0: PWM 时钟禁止
4	ADCCEN	1: ADC 时钟使能 0: ADC 时钟禁止
3	SPICEN	1: SPI 时钟使能 0: SPI 时钟禁止
2	WDTCCEN	1: WDT 时钟使能 0: WDT 时钟禁止
1	UART1CEN	1: UART1 时钟使能 0: UART1 时钟禁止
0	UART0CEN	1: UART0 时钟使能 0: UART0 时钟禁止

4.1.66 PCLK1

DFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCLK1	-	-	-	TIMCEN	-	GPIO2CEN	GPIO1CEN	GPIO0CEN
读/写	读	读	读	读/写	读	读/写	读/写	读/写
复位值	0	0	0	1	0	1	1	1

位编号	位符号	说明
7:5	-	-
4	TIMCEN	1: TIMER 时钟使能 0: TIMER 时钟禁止
3	-	-
2	GPIO2CEN	1: GPIO2 时钟使能 0: GPIO2 时钟禁止
1	GPIO1CEN	1: GPIO1 时钟使能 0: GPIO1 时钟禁止
0	GPIO0CEN	1: GPIO0 时钟使能 0: GPIO0 时钟禁止

注：PCLK1 的比特位写操作必须用位与等 (&=) 和位或等 (|=) 进行操作，例如操作 GPIO0CEN 时用“PCLK1 &= ~(1<<0)”，“PCLK1 |= (1<<0)”。不能用直接赋值的方式进行寄存器写操作 (PCLK1=0x1)，否则会带来 TIMER 不能正常工作。

4.1.67 ACC

累加器 ACC 是一个常用的专用寄存器，常用于存放参加运算的操作数及运算结果。

E0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ACC	ACC							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	ACC	存放参加运算的操作数及运算结果

4.1.68 PxIRQ

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0IRQ(E1H)	-	-	-	P0IRQ.4	P0IRQ.3	P0IRQ.2	P0IRQ.1	P0IRQ.0
P1IRQ(E2H)	-	-	P1IRQ.5	P1IRQ.4	P1IRQ.3	P1IRQ.2	P1IRQ.1	P1IRQ.0
P2IRQ(E3H)	P2IRQ.7	P2IRQ.6	P2IRQ.5	-	P2IRQ.3	P2IRQ.2	-	P2IRQ.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	PxIRQ.y x=0-4, y=0-7	端口中断标志位: 0: 端口未产生中断 1: 端口产生了中断 写0清0

4.1.69 P2PD

E4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2PD	P2_7PD	P2_6PD	P2_5PD	-	P2_3PD	P2_2PD	-	P2_0PD
读/写	读/写	读/写	读/写	读	读/写	读/写	读	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	P2_7PD	端口 P2_7 下拉配置寄存器 0: 禁止 1: 使能
6	P2_6PD	端口 P2_6 下拉配置寄存器 0: 禁止 1: 使能
5	P2_5PD	端口 P2_5 下拉配置寄存器 0: 禁止 1: 使能
4	-	-
3	P2_3PD	端口 P2_3 下拉配置寄存器 0: 禁止 1: 使能
2	P2_2PD	端口 P2_2 下拉配置寄存器 0: 禁止 1: 使能
1	-	-
0	P2_0PD	端口 P1_0 下拉配置寄存器 0: 禁止 1: 使能

4.1.70 P1DR

E5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1DR	-	-	P1_5DR	P1_4DR	P1_3DR	P1_2DR	P1_1DR	P1_0DR

读/写	读	读	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:6	-	-
5	P1_5DR	端口 P1_5 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
4	P1_4DR	端口 P1_4 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
3	P1_3DR	端口 P1_3 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
2	P1_2DR	端口 P1_2 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
1	P1_1DR	端口 P1_1 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
0	P1_0DR	端口 P1_0 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力

4.1.71 PRESET0

E6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PRESET 0	I2CRE N	LPTIMRE N	PWMRE N	ADCRE N	SPIRE N	WDTRE N	UART1RE N	UART0RE N
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7	I2CREN	1: I2C 复位释放 0: I2C 复位
6	LPTIMREN	1: LPTIM 复位释放 0: LPTIM 复位
5	PWMREN	1: PWM 复位释放 0: PWM 复位
4	ADCREN	1: ADC 复位释放 0: ADC 复位
3	SPIREN	1: SPI 复位释放 0: SPI 复位
2	WDTREN	1: WDT 复位释放 0: WDT 复位
1	UART1REN	1: UART1 复位释放 0: UART1 复位
0	UART0REN	1: UART0 复位释放 0: UART0 复位

4.1.72 PRESET1

E7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PRESET1	-	-	-	TIMREN	-	GPIO2REN	GPIO1REN	GPIO0REN
读/写	读	读	读	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	1	0	1	1	1

位编号	位符号	说明
7:5	-	-
4	TIMREN	1: TIMER 复位释放 0: TIMER 复位
3	-	-
2	GPIO2REN	1: GPIO2 复位释放 0: GPIO2 复位
1	GPIO1REN	1: GPIO1 复位释放 0: GPIO1 复位
0	GPIO0REN	1: GPIO0 复位释放 0: GPIO0 复位

4.1.73 PxIEN

-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0IEN(E9H)	-	-	-	P0IEN.4	P0IEN.3	P0IEN.2	P0IEN.1	P0IEN.0
P1IEN(EAH)	-	-	P1IEN.5	P1IEN.4	P1IEN.3	P1IEN.2	P1IEN.1	P1IEN.0
P2IEN(EBH)	P2IEN.7	P2IEN.6	P2IEN.5	-	P2IEN.3	P2IEN.2	-	P2IEN.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	PxIEN.y x=0-4, y=0-7	端口中断使能控制位: 0: 关闭端口中断功能 1: 使能端口中断功能

4.1.74 P2OD

ECH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2OD	P2_7OD	P2_6OD	P2_5OD	-	P2_3OD	P2_2OD	-	P2_0OD
读/写	读/写	读/写	读/写	读	读/写	读/写	读	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	P2_7OD	端口 P2_7 开漏输出配置寄存器 0: 禁止 1: 使能
6	P2_6OD	端口 P2_6 开漏输出配置寄存器 0: 禁止 1: 使能

5	P2_5OD	端口 P2_5 开漏输出配置寄存器 0: 禁止 1: 使能
4	-	-
3	P2_3OD	端口 P2_3 开漏输出配置寄存器 0: 禁止 1: 使能
2	P2_2OD	端口 P2_2 开漏输出配置寄存器 0: 禁止 1: 使能
1	-	-
0	P2_0OD	端口 P2_0 开漏输出配置寄存器 0: 禁止 1: 使能

4.1.75 ADCCONT

EDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCONT	ADCCOUNT							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	1

位编号	位符号	说明
7:0	ADCCOUNT	通道切换间隔时间，该值的单位是 ADC 时钟周期。 实际的通道切换时间 = (adc_count + 16) * ADC 时钟周期时间 注意：该寄存器只能在启用 ADC 控制器之前配置

4.1.76 RCHDIV

EEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCHDIV	-	-	-	-	-	-	RCHDIV	
读/写	读	读	读	读	读	读	读/写	读/写
复位值	0	0	0	0	0	0	1	0

位编号	位符号	说明
7:2	-	-
1:0	RCHDIV	RCH 分频设置： 00: 不分频（系统跑不到，不配置） 01: 2 分频 10: 3 分频 11: 4 分频

4.1.77 B

在乘除指令中，会用到 B 寄存器，在其他指令中，B 寄存器可作为通用暂存寄存器。

F0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
B	B							
读/写	读	读/写	读/写	读/写	读/写	读/写	读/写	读/写

复位值	0	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

位编号	位符号	说明
7:0	B	

4.1.78 PxPU

-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0PU(F1H)	-	-	-	P0PU.4	P0PU.3	P0PU.2	P0PU.1	P0PU.0
P1PU(F2H)	-	-	P1PU.5	P1PU.4	P1PU.3	P1PU.2	P1PU.1	P1PU.0
P2PU(F3H)	P2PU.7	P2PU.6	P2PU.5	-	P2PU.3	P2PU.2	-	P2PU.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7:0	PxPU.y x=0-4, y=0-7	端口上拉使能控制位（上拉电阻参考值 60KΩ）： 0: 使能内部上拉电阻 1: 关闭内部上拉电阻

4.1.79 P2CS

F4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2CS	P2_7CS	P2_6CS	P2_5CS	-	P2_3CS	P2_2CS	-	P2_0CS
读/写	读/写	读/写	读/写	读	读/写	读/写	读	读/写
复位值	1	1	1	0	1	1	0	1

位编号	位符号	说明
7	P2_7CS	端口 P2_7 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
6	P2_6CS	端口 P2_6 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
5	P2_5CS	端口 P2_5 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
4	-	-
3	P2_3CS	端口 P2_3 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
2	P2_2CS	端口 P2_2 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer
1	-	-
0	P2_0CS	端口 P2_0 输入类型配置寄存器 0: Schmitt input buffer 1: CMOS input buffer

4.1.80 ADCHL

F5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCHL	-	ADCHL[6:0]						
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	-	-
6:0	ADCHL	ADCH[6:0]通道设置配置位 ADCH[0]: 0: P1_4 配置为 GPIO 功能 1: P1_4 配置为 ADC 输入 ADCH[1]: 0: P1_5 配置为 GPIO 功能 1: P1_5 配置为 ADC 输入 ADCH[2]: 0: P2_0 配置为 GPIO 功能 1: P2_0 配置为 ADC 输入 ADCH[3]: 0: P2_2 配置为 GPIO 功能 1: P2_2 配置为 ADC 输入 ADCH[4]: 0: P2_3 配置为 GPIO 功能 1: P2_3 配置为 ADC 输入 ADCH[5]: 0: P2_6 配置为 GPIO 功能 1: P2_6 配置为 ADC 输入 ADCH[6]: 0: P2_7 配置为 GPIO 功能 1: P2_7 配置为 ADC 输入

4.1.81 ADCCSTAT

F6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCSTAT	-	-	-	-	-	-	-	RXAVL
读/写	读	读	读	读	读	读	读	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:1	-	-
0	RXAVL	此信号表示 BUF 接收到数据。此信号便于 CPU 轮询操作。 当接收器 BUF 非空时，此位为 1。 1: 接收器 BUF 存有数据 0: 接收器 BUF 为空 写 1 清 0

4.1.82 ADCSPW

F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCSPW	-	-	-	-	-	SAMPW		
读/写	读	读	读	读	读	读/写	读/写	读/写
复位值	0	0	0	0	0	1	0	0

位编号	位符号	说明
7:3	-	-
2:0	SAMPW	<p>采样时钟脉冲宽度配置。注意：在本芯片设计中，此寄存器应该设置大于或等于 3 的值。</p> <p>3: SAMPCLK 宽为 4 个 ADC_CLK 脉冲信号 4: SAMPCLK 宽为 5 个 ADC_CLK 脉冲信号 5: SAMPCLK 宽为 6 个 ADC_CLK 脉冲信号</p> <p>此寄存器的合法值范围为 3~5，超过此范围可能会引起 ADC 工作不正常。</p>

4.1.83 CLKCON

F8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	-	RC32KF	RC48MF	XCLKF	RC32KEN	RC48MEN	XCLKEN	HSCLKSEL
读/写	-	读	读	读	读/写	读/写	读/写	读/写
复位值	-	1	1	0	1	1	0	0

位编号	位符号	说明
7	-	-
6	RC32KF	RC32K 时钟源标志位： 0: RC32K 未正常启动 1: RC32K 已正常启动
5	RC48MF	RC48M 时钟源标志位： 0: RC48M 未正常启动 1: RC48M 已正常启动
4	XCLKF	XCLK 时钟源开关标志位： 0: 未检测到有效的外部输入时钟 1: 已检测到有效的外部输入时钟
3	RC32KEN	RC32K 时钟源开关控制位： 0: 关闭 RC32K 1: 打开 RC32K
2	RC48MEN	RC48M 时钟源开关控制位： 0: 关闭内部 RC48M 时钟源，若当前系统时钟为 RC48M，写 0 无效 1: 打开内部 RC48M 时钟源

1	XCLKEN	XCLK 时钟源开关控制位： 0: 关闭 XCLK, P0.0, P0.1 为 GPIO 功能 如果当前系统时钟为 XCLK, XCLKEN 写 0 不能关闭 XCLK 1: 打开 XCLK, P0.0, P0.1 作为晶振或外部 Clock 输入功能
0	HSCLKSEL	系统时钟源头选择位, 即 HSCLK 来自经过 RCHDIV 分频后的内部高频时钟 RCH48M 和 XCLK 的选择位： 0: 选择经过 RCHDIV 分频后的内部高频时钟 RCH48M 作为系统时钟源 1: 选择 XCLK 作为系统时钟源

4.1.84 PxOEN

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0OEN(F9H)	-	-	-	P0OEN.4	P0OEN.3	P0OEN.2	P0OEN.1	P0OEN.0
P1OEN(FAH)	-	-	P1OEN.5	P1OEN.4	P1OEN.3	P1OEN.2	P1OEN.1	P1OEN.0
P2OEN(FBH)	P2OEN.7	P2OEN.6	P2OEN.5	-	P2OEN.3	P2OEN.2	-	P2OEN.0
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7:0	PxOEN.y x=0-4, y=0-7	端口方向控制位： 0: 输出模式 1: 输入模式

4.1.85 P2DR

FCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2DR	P2_7DR	P2_6DR	P2_5DR	-	P2_3DR	P2_2DR	-	P2_0DR
读/写	读/写	读/写	读/写	读	读/写	读/写	读	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	P2_7DR	端口 P2_7 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
6	P2_6DR	端口 P2_6 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
5	P2_5DR	端口 P2_5 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
4	-	-
3	P2_3DR	端口 P2_3 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力

2	P2_2DR	端口 P2_2 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力
1	-	-
0	P2_0DR	端口 P2_0 驱动能力配置寄存器 0: 高驱动能力 1: 低驱动能力

4.1.86 ADCVREF

FDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCVREF	-	-	ADCSPEED		-	-	-	VREFSEL
读/写	-	-	读/写	读/写	-	-	-	读/写
复位值	-	-	0	0	-	-	-	0

位编号	位符号	说明
7:6	-	-
5:4	ADCSPEED	ADCSPEED 选择位, 保留为默认值 0
3:1	-	-
0	VREFSEL	ADC 电压基准选择控制位: 0: VDDH 作为 ADC 电压基准源 1: 外接 VREF 作为 ADC 电源基准源(当配置为 1 时, P2_5 端口将作为外部基准输入)

4.1.87 ADCCDR0

FEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCDR0	CLKDIV0							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	1	1	1	1	1	1	1	1

位编号	位符号	说明
7:0	CLKDIV0	ADC 内部时钟分频倍数。 ADC 时钟频率公式为 $f_{\text{adc_clk}} = f_{\text{SYSCLK}} / \text{clkdiv}$ 其中, $f_{\text{adc_clk}}$ 是 ADC 内部时钟的频率, f_{SYSCLK} 是系统时钟频率, clkdiv 是分频倍数 = {CLKDIV1, CLKDIV0}。 注意: 请勿把 clkdiv 设为 0 或 1, 若把 clkdiv 设为 0 或 1, 也当作 2 分频。如需使用 1 分频, 建议使用外部时钟。

4.1.88 ADCCDR1

FFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCDR1	CLKDIV1							
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:0	CLKDIV1	ADC 内部时钟分频倍数。 ADC 时钟频率公式为 $f_{\text{adc_clk}} = f_{\text{SYSCLK}} / \text{clkdiv}$ 其中, $f_{\text{adc_clk}}$ 是 ADC 内部时钟的频率, f_{SYSCLK} 是系统时钟频率, clkdiv 是分频倍数 = {CLKDIV1, CLKDIV0}。 注意: 请勿把 clkdiv 设为 0 或 1, 若把 clkdiv 设为 0 或 1, 也当作 2 分频。如需使用 1 分频, 建议使用外部时钟。

4.1.89 LVD_LV

C004H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVD_LV	-	-	-	-	-	LVD_TSET		LVDLVEN
读/写	读	读	读	读	读	读/写	读/写	读/写
复位值	0	0	0	0	0	1	1	1

位编号	位符号	说明
7:3	-	-
2:1	LVD_TSET	滤波时间设置位: 11: 滤波时间为 1 个 RC32K 时钟 10: 滤波时间为 8 个 RC32K 时钟 01: 滤波时间为 16 个 RC32K 时钟 00: 滤波时间为 29 个 RC32K 时钟
0	LVDLVEN	LVD 滤波使能位 1: 开启 RC32K 时钟滤波功能 0: 关闭 RC32K 时钟滤波功能

4.2 系统时钟

4.2.1 主要特性

- 内建 48M RC 高频振荡器 (系统最高可以跑 24MHz, 也即内部高频 RC 时钟的 2 分频, 此时需将 flash 读等待时间设置为 1。默认情况下内部高频 RC 时钟 3 分频之后作为系统时钟, 也即 16MHz)
- 内建 32K RC 低频振荡器
- 支持外部晶振输入 (2M~24M) 作为系统时钟
- 内建系统时钟分频器

4.2.2 时钟定义

- RC48M:
表示内部 48M RC 高频振荡器。
- RCH48M:
表示内部 48M RC 高频振荡器经 RCHDIV 分频后的内部高速时钟。

- RC32K:
表示内部 32K RC 低频振荡器。
- XCLK:
晶体谐振器时钟 (2-24MHZ 晶体谐振器) 或外部时钟输入。

4.2.3 时钟结构图

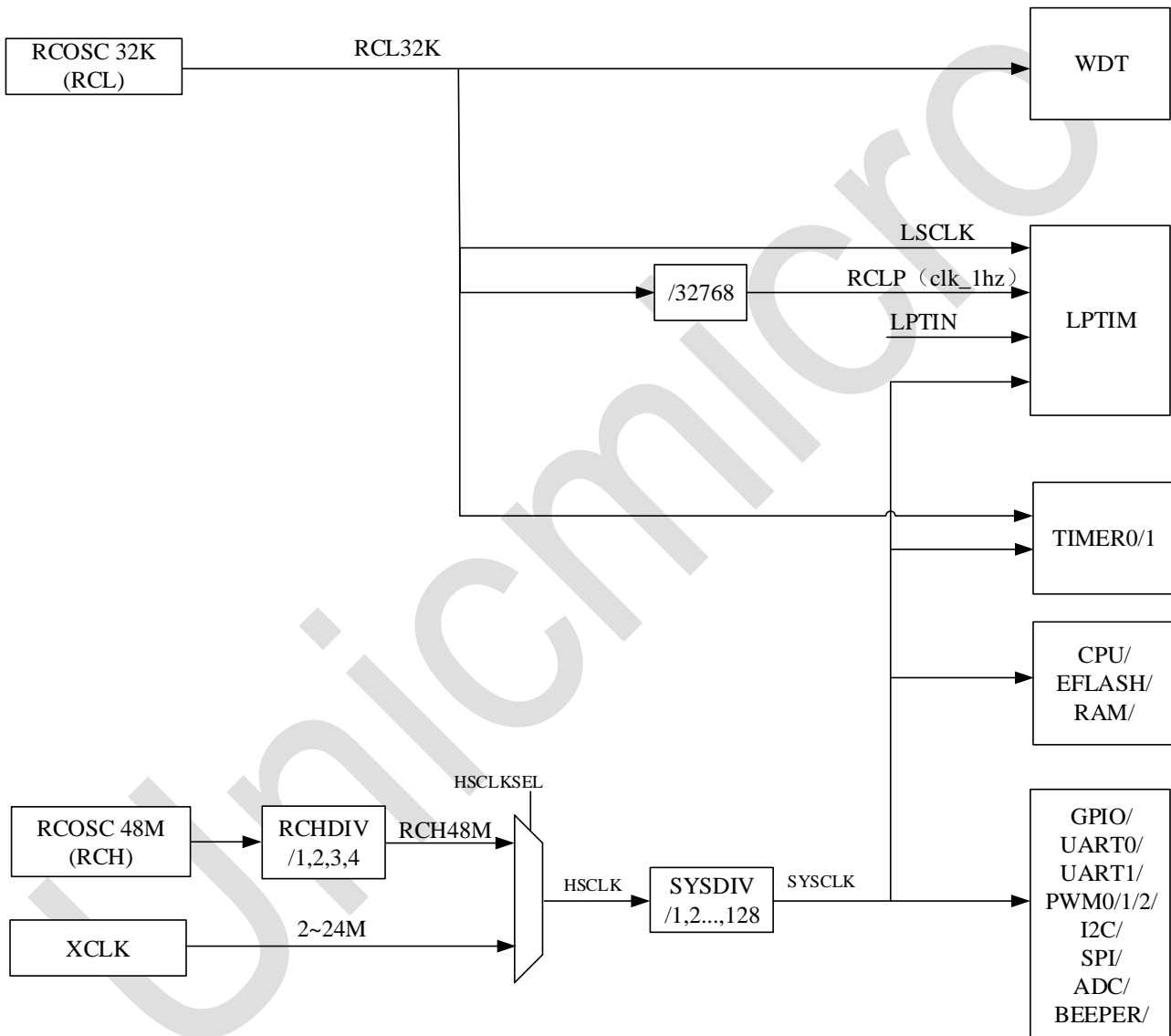


图 4-1: 时钟结构图

4.3 复位源

4.3.1 主要特性

包含以下复位源:

- PIN RESET

- LVD 复位
- LVR 复位
- 看门狗复位 (WDT)
- 上电复位 (POR)

注意：LVD 复位不复位 LVDCON 寄存器；LVR 复位与 POR 复位同为全局复位，复位所有的数字逻辑；

4.3.2 看门狗复位

看门狗定时器是一个递减计数器，看门狗定时器使用内部 RC32K 为时钟源，如果要使能看门狗，必须先确保 RC32K 已打开。在掉电模式下，看门狗会在定时器溢出时，复位芯片。

读或者写 RSTSTAT 寄存器均自动清除看门狗计数。

4.3.3 LVD 与 LVR 复位

LVD 为低电压检测单元，可以由软件配置，在外部供电电压低到检测阈值以下时，会产生一个复位。LVD 复位可以复位 CPU 和除 EFC 外的其它外设，不复位 EFC。

LVR 为下电复位单元，其作用域和 POR 相同，为全局复位。当外部供电电压小于 LVR 设置的阈值电压时，LVR 产生复位，复位整个芯片。

4.3.4 外部复位

外部复位作用域和 POR 相同，为全局复位。当 RESETN IO 为 0，且 RESETN 作为外部复位功能使用时，外部复位有效，复位整个芯片。

4.3.5 寄存器

复位相关寄存器参见 SFR 部分的 RSTSTAT、LVDCON、POREN 等部分

4.4 低功耗模式

4.4.1 主要特性

- 支持睡眠模式 (Sleep)、深度睡眠模式(DeepSleep)、停止模式(Stop)三种省电模式
- 中断和复位可以退出三种省电模式

4.4.2 低功耗模式

芯片除正常工作模式外，为了降低芯片的电流消耗，提供三种低功耗模式：休眠（Sleep）模式、深度休眠（Deepsleep）模式和停止（Stop）模式。

在休眠模式下，CPU 停止工作，保留中断处理功能。其它外设等模块时钟和复位可由软件设置。休眠模式由软件向特定的 SFR（PCON -> IDLE）写 1 进入，唤醒由中断触发。

深度休眠模式是休眠模式的升级，在此模式下，CPU 停止运行，高速时钟停止运行，低功耗功能模块（LPTIMER、TIMER0/1、WDT）可以运行。深度休眠模式由软件向特定的 SFR（PCON -> STOP）写 1 进入，唤醒由中断触发。

停止模式下，高速时钟和低速时钟均停止运行，系统无任何运行的时钟，一切外围模块均停止运行。上电复位信号有效，IO 状态保持，IO 中断有效，所有寄存器，RAM 和 CPU 数据保存状态时的功耗；停止模式要先设置 SFR 中 PDSEL 寄存器为 1，然后向 PCON -> STOP 写 1 进入，唤醒只能由外部引脚的电平来唤醒。

4.4.3 低功耗模式表

模式	模式描述	进入条件	退出条件
Sleep	CPU 大部分休眠；软件可关闭各模块时钟。	<ol style="list-style-type: none"> 1. 根据需要，关闭各外设模块时钟，仅留下需要监测中断事件的模块。 2. PCON -> IDLE 写 1。 	<ol style="list-style-type: none"> 1. CPU 检测到中断或事件发生。 2. 进入中断服务程序清中断并返回。 3. 继续执行后续指令。
Deepsleep	CPU 大部分休眠；高速时钟源关闭，低速时钟源运行	<ol style="list-style-type: none"> 1. 根据需要，关闭各外设模块时钟，仅留下需要监测中断事件的模块。 2. PCON -> STOP 写 1。 	<ol style="list-style-type: none"> 1. CPU 检测到中断或事件发生。 2. 进入中断服务程序清中断并返回。 3. 继续执行后续指令。
Stop	关闭系统所有时钟	<ol style="list-style-type: none"> 1. 根据需要，设置 IO 唤醒的条件。 2. 设置 SFR 中 PDSEL 寄存器为 1。 3. PCON -> STOP 写 1。 	<ol style="list-style-type: none"> 1. 外部 IO 唤醒事件到来。 2. CPU 检测到 IO 唤醒事件中发生。 3. 进入中断服务程序清中断并返回。 4. 继续执行后续指令。

5 EFC

5.1 概述

芯片上集成了 32K Bytes 的 EFLASH 存储器，用于保存芯片所有的关键脱机信息和数据。EFC 为 EFLASH 控制器，在 CPU 的配置下，完成 EFLASH 读、写、擦除等操作。

5.2 主要特性

- 支持 EFLASH 的读写（8bit）、sector 擦除等操作流程
- 读等待时间可以配置
- 主区有 64 个 sector，每个 512 字节
- NVR 区有 2 个 sector，每个 512 字节
- 支持对 NVR 区域擦/写保护功能
- 支持擦写保护功能
- 支持自动锁总线功能

5.3 EFLASH 读效率

当 RD_WAIT 值设置为 0 时，CPU 取指时无效率损失，读 EFLASH 与读取 ROM 在控制器端时序相同。RD_WAIT 设置为 1 时，EFC 总线在每个读操作时会被拉低 1 个周期(系统时钟运行在 24MHz 时，需将 RD_WAIT 设置为 1)

5.4 软件流程

5.4.1 Read 操作

EFLASH 上电稳定后可以执行读操作。读操作注意配置读等待时间 RD_WAIT。

5.4.2 Write 操作

- 使能 OPSET.PAGEWSET 写模式
- 往 OADR 写入 EFlash 地址
- 往 ODATA 写入数据
- 配置 OCTRL.PUMP_SEL 电压

- 使能 OCTRL.PUMP_EN
- 等待 PUMP_OK 置 1
- 使能 VPPO_EN 启动编程
- 等待 VPPO_EN 清 0 编程完成
- 关闭 PUMP_EN
- 关闭 OPSET.PAGEWSET 写模式

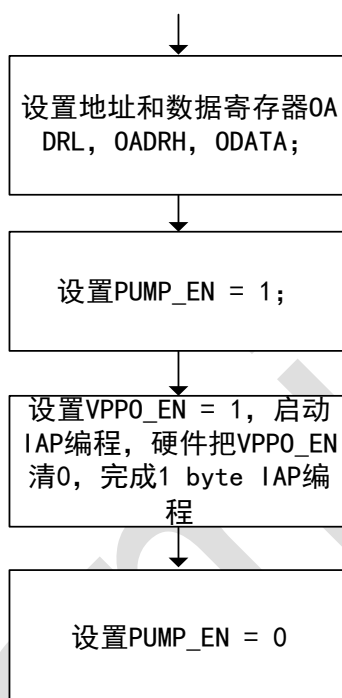


图 5-1: 写操作流程

5.4.3 Erase 操作

- 关闭 OPSET.PAGEWSET 写模式
- 使能 OPSET.PAGESERSET page 擦除模式
- 往 OADR 写入 page 地址
- 配置 OCTRL.PUMP_SEL 电压
- 使能 OCTRL.PUMP_EN
- 等待 PUMP_OK 置 1
- 使能 VPPO_EN 启动编程
- 等待 VPPO_EN 清 0 编程完成
- 关闭 PUMP_EN
- 关闭 OPSET.PAGESERSET page 擦除模式

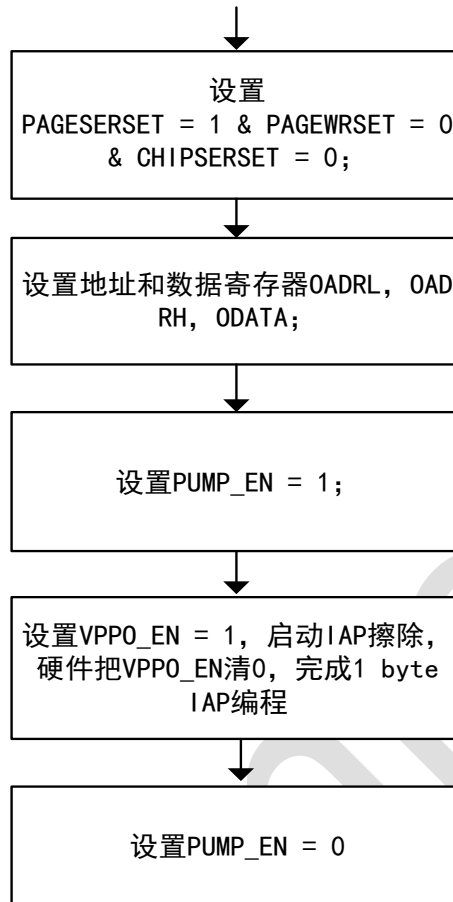


图 5-2: 擦除操作流程

6 PWM (脉宽调制模块)

6.1 特性

- 三路 16 位精度 PWM 模块
- 提供每个 PWM 周期溢出中断
- 输出极性选择

6.2 功能描述

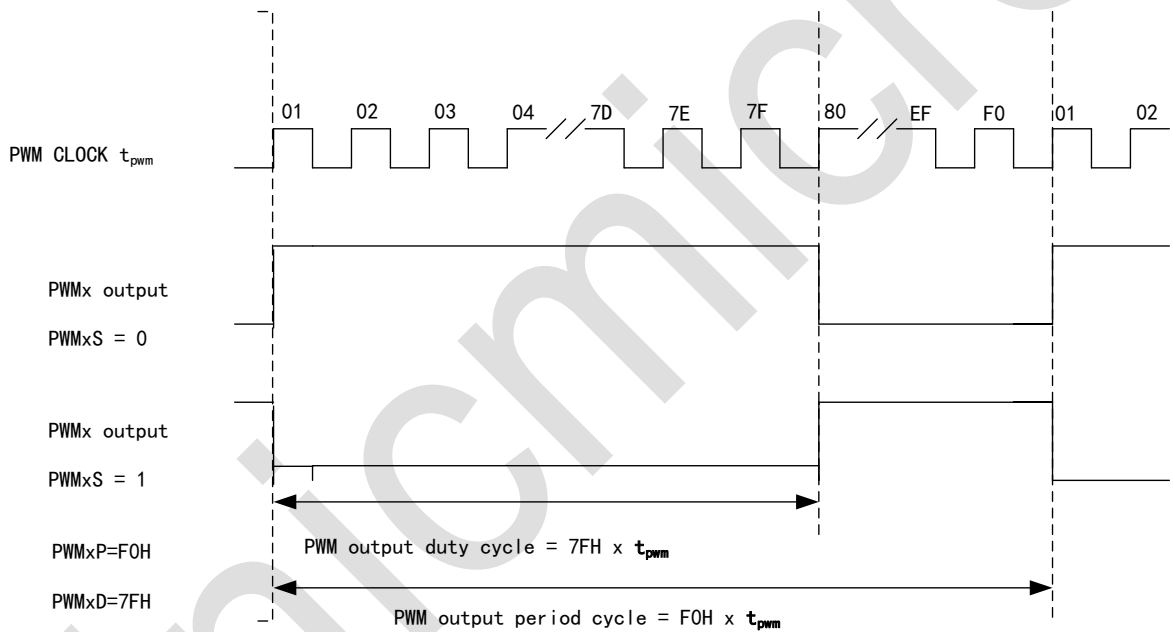


图 6-1: PWM 输出范例

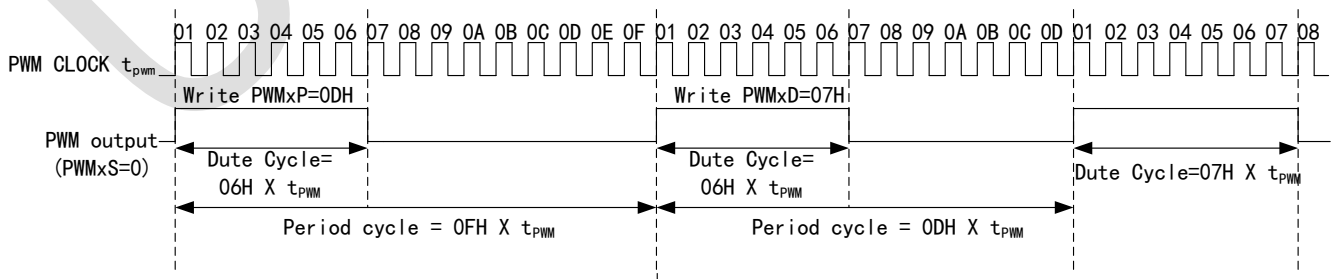


图 6-2: PWM 输出周期或占空比改变范例

6.3 PWM 输出时序

表 6-1: 输出时序 (PWMxSS=1)

PWMxS	条件	PWMx 端口输出状态
0	$PWMxP > PWMxD$	输出方波, 先输出 PWMxD 个时钟周期高电平, 然后输出 PWMxP-PWMxD 个时钟周期低电平
	$PWMxP \leq PWMxD$	高电平
	$PWMxD = 0, PWMxP = 0;$	高电平
	$PWMxD = 0, PWMxP \neq 0;$	低电平
1	$PWMxP > PWMxD$	输出方波, 先输出 PWMxD 个时钟周期低电平, 然后输出 PWMxP-PWMxD 个时钟周期高电平
	$PWMxP \leq PWMxD$	低电平
	$PWMxD = 0, PWMxP = 0;$	低电平
	$PWMxD = 0, PWMxP \neq 0;$	高电平

注意:

1. PWMxEN 位控制 PWMx 模块开关。
2. PWMxSS (x = 0-2) 位能选择端口是作为 I/O 端口还是 PWM 输出端口, PWMxSS = 1, 但 PWMxEN=0, 则相应端口处于输入状态。
3. IEN0 寄存器中的 EPWM 位和 PWMxCON 寄存器中的 PWMxIE 位会共同控制 PWMx 中断。
4. 3 路 PWM 模块共用中断向量。
5. 当 PWMENx=1, PWMxSS=0 时, PWMx 模块输出关闭, 此时 PWM 模块可用作 16 位定时器, 如果 PWM 中断被使能且 PWMxIE=1, 每个 PWM 周期会同样触发中断。

6.4 使用流程

PWM 使用流程:

1. 设置寄存器 PWMxCON.PWMxSS, PWMx 引脚输出允许。
2. 设置寄存器 PWMxCON.PWMxS, 选择 PWMx 输出模式。
3. 设置寄存器 PWMxDL/H, 配置占空比。
4. 设置寄存器 PWMxPL/H, 配置重装载值。
5. 设置寄存器 PWMxCON.PWMxEN, 使能 PWMx 模块。
6. 若需产生中断, 则需设置寄存器 PWMxCON.PWMxIE, 并设置寄存器 IEN0.EA 和 IEN0.EPWM。

7 GPIO(I/O 端口)

7.1 主要特性

- 最多支持 17 个可编程双向 I/O 端口
- 全部 IO 支持中断功能，双沿触发
- 内建上拉电阻
- I/O 口可与其他功能共用

最多支持 17 个可编程双向 I/O 端口。

7.2 端口复用

部分端口可复用做其他功能，当第二功能开始时，GPIO 配置为输出状态时依然可读写寄存器，配置为输入状态时，读无效。

表 7-1: P0 口复用

端口	优先级	功能	使能方式
P0.0	1	XIN	XCLKEN = 1 (CLKCON bit1)
	2	P0.0	不存在上述情况
P0.1	1	XOUT	XCLKEN = 1 (CLKCON bit1)
	2	P0.1	不存在上述情况
P0.2	1	NRST	ESTCR[3]=1, P0.2 作为外部复位 RESETn
	2	P0.2	不存在上述情况
P0.3	1	SPI_CSN	SPI_EN=1 且 CS_EN=1, P0.3 作为 SPI_CSN
	2	LPOUT	LPOUT_IO_IEN = 1
	3	P0.3	不存在上述情况
P0.4	1	I2C_SDA	I2C_EN = 1
	2	P0.4	不存在上述情况

表 7-2: P1 口复用

端口	优先级	功能	使能方式
P1.0	1	PWM0	PWM0SS = 1
	2	I2C_SCL	I2C_EN = 1
	3	LPTIM	LPTIN_IO_IEN = 1
	4	P1.0	不存在上述情况
P1.1	1	PWM1	PWM1SS = 1
	2	LPTIM_EXT	EXTRIGGER_IO_IEN = 1
	3	P1.1	不存在上述情况
P1.2	1	PWM2	PWM2SS = 1
	2	P1.2	不存在上述情况

端口	优先级	功能	使能方式
P1.3	1	SPI_SCK	SPI_EN=1 且 SCK_EN=1, P1.3 作为 SPI_SCK
	2	P1.3	不存在上述情况
P1.4	1	AIN0	ADCHL[0]=1, P1.4 作为 AIN0
	2	RX1	UART1_EN = 1, P1.4 作为 RX1
	3	SPI_MOSI	SPI_EN=1 且 MOSI_EN=1, P1.4 作为 SPI_MOSI
	4	P1.4	不存在上述情况
P1.5	1	TX1	UART1_EN = 1, P1.5 作为 TX1
	2	AIN1	ADCHL[1]=1, P1.5 作为 AIN1
	3	SPI_MISO	SPI_EN=1 且 MISO_EN=1, P1.4 作为 SPI_MISO
	4	P1.5	不存在上述情况

表 7-3: P2 口复用

端口	优先级	功能	使能方式
P2.0	1	AIN2	ADCHL[2]=1, P2_0 作为 AIN2
	2	P2.0	不存在上述情况
P2.2	1	AIN3	ADCHL[3]=1, P2_2 作为 AIN3
	2	P2.2	不存在上述情况
P2.3	1	AIN4	ADCHL[4]=1, P2_3 作为 AIN4
	2	P2.3	不存在上述情况
P2.5	1	VREF	ADCVREF[0]=1, P2_5 作为 VREF
	2	BEEP	BEEPEN =1, P2.5 作为 BEEP 输出
	3	P2.5	不存在上述情况
P2.6	1	AIN5	ADCHL[5]=1, P2.6 作为 AIN5
	2	TX0	UART0_EN = 1, P2.6 作为 TX0
	3	P2.6	不存在上述情况
P2.7	1	AIN6	ACHL[6]=1, P2.7 作为 AIN6
	2	RX0	UART0_EN = 1, P2.7 作为 RX0
	3	P2.7	不存在上述情况

7.3 端口中断

全部端口都支持中断功能, 如果使能中断, 上升沿或者下降沿均会触发中断, 全部端口中断共用 INTO 中断入口。

PxIEN 寄存器控制所有 IO 口中断使能的打开与关闭, 如果某一端口的中断功能被打开, 那么当 MCU 检测到引脚上出现上升沿或者下降沿则产生外部中断, 并且置相应的 PxIRQ 标志位为 1。用户程序可以在外部中断 0 的服务程序内查询是哪个引脚产生了中断标志位。

如果端口中断被使能, 那么当 MCU 进入掉电模式时, 端口电平的变化, 可以中断唤醒 MCU。

注意事项: 端口中断模式仅支持低功耗模式下的 IO 唤醒, 正常模式下不支持端口中断, 仅支持查询 IO 口电平状态 (P0/P1/P2)。

7.4 IO_IE

IO_IE 寄存器用 xdata 关键字访问

P0_IE 地址: 0xC000

P1_IE 地址: 0xC001

P2_IE 地址: 0xC002

IO_IE 控制 IO 的输入使能。禁止输入使能后, P0/P1/P2 寄存器无法读取 IO 状态, 无法使用输入功能, 输出功能正常。

7.4.1 P0_IE

C000H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0_IE	-	-	-	P0_4_IE	P0_3_IE	P0_2_IE	P0_1_IE	P0_0_IE
读/写	读	读	读	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	1	1	1	1	1

位编号	位符号	说明
7-5	-	-
4	P0_4_IE	P0_4 IO 输入控制位 0: P0_4 禁止输入 1: P0_4 开启输入
3	P0_3_IE	P0_3 IO 输入控制位 0: P0_3 禁止输入 1: P0_3 开启输入
2	P0_2_IE	P0_2 IO 输入控制位 0: P0_2 禁止输入 1: P0_2 开启输入
1	P0_1_IE	P0_1 IO 输入控制位 0: P0_1 禁止输入 1: P0_1 开启输入
0	P0_0_IE	P0_0 IO 输入控制位 0: P0_0 禁止输入 1: P0_0 开启输入

7.4.2 P1_IE

C001H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1_IE	-	-	P1_5_IE	P1_4_IE	P1_3_IE	P1_2_IE	P1_1_IE	P1_0_IE
读/写	读	读	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	1	1	1	1	1	1

位编号	位符号	说明
7:6	-	-

5	P1_5_IE	P1_5 IO 输入控制位 0: P1_5 禁止输入 1: P1_5 开启输入
4	P1_4_IE	P1_4 IO 输入控制位 0: P1_4 禁止输入 1: P1_4 开启输入
3	P1_3_IE	P1_3 IO 输入控制位 0: P1_3 禁止输入 1: P1_3 开启输入
2	P1_2_IE	P1_2 IO 输入控制位 0: P1_2 禁止输入 1: P1_2 开启输入
1	P1_1_IE	P1_1 IO 输入控制位 0: P1_1 禁止输入 1: P1_1 开启输入
0	P1_0_IE	P1_0 IO 输入控制位 0: P1_0 禁止输入 1: P1_0 开启输入

7.4.3 P2_IE

C002H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2_IE	P2_7_IE	P2_6_IE	P2_5_IE	-	P2_3_IE	P2_2_IE	-	P2_0_IE
读/写	读/写	读/写	读/写	读	读/写	读/写	读	读/写
复位值	1	1	1	0	1	1	0	1

位编号	位符号	说明
7	P2_7_IE	P2_7 IO 输入控制位 0: P2_7 禁止输入 1: P2_7 开启输入
6	P2_6_IE	P2_6 IO 输入控制位 0: P2_6 禁止输入 1: P2_6 开启输入
5	P2_5_IE	P2_5 IO 输入控制位 0: P2_5 禁止输入 1: P2_5 开启输入
4	-	-
3	P2_3_IE	P2_3 IO 输入控制位 0: P2_3 禁止输入 1: P2_3 开启输入
2	P2_2_IE	P2_2 IO 输入控制位 0: P2_2 禁止输入 1: P2_2 开启输入
1	-	-
0	P2_0_IE	P2_0 IO 输入控制位 0: P2_0 禁止输入 1: P2_0 开启输入

7.5 IO_SR

IO_SR 寄存器用 xdata 关键字访问

P0_SR 地址: 0xC005

P1_SR 地址: 0xC006

P2_SR 地址: 0xC007

IO_SR 控制 IO 的 Slew rate。Slew rate 指的是压摆率，又称转换速率，表示 IO 高低电平切换时，边沿上升和下降速度的快慢（边沿斜率的陡峭和缓慢）

7.5.1 P0_SR

C005H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0_SR	-	-	-	P0_4_SR	P0_3_SR	P0_2_SR	P0_1_SR	P0_0_SR
读/写	读	读	读	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	1	1	1	1	1

位编号	位符号	说明
7:5	-	-
4	P0_4_SR	P0_4 IO 速度控制位 0: P0_4 快速 1: P0_4 慢速
3	P0_3_SR	P0_3 IO 速度控制位 0: P0_3 快速 1: P0_3 慢速
2	P0_2_SR	P0_2 IO 速度控制位 0: P0_2 快速 1: P0_2 慢速
1	P0_1_SR	P0_1 IO 速度控制位 0: P0_1 快速 1: P0_1 慢速
0	P0_0_SR	P0_0 IO 速度控制位 0: P0_0 快速 1: P0_0 慢速

7.5.2 P1_SR

C006H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1_SR	-	-	P1_5_SR	P1_4_SR	P1_3_SR	P1_2_SR	P1_1_SR	P1_0_SR
读/写	读	读	读/写	读/写	读/写	读/写	读/写	读/写
复位值	0	0	1	1	1	1	1	1

位编号	位符号	说明
-----	-----	----

7:6	-	-
5	P1_5_SR	P1_5 IO 速度控制位 0: P1_5 快速 1: P1_5 慢速
4	P1_4_SR	P1_4 IO 速度控制位 0: P1_4 快速 1: P1_4 慢速
3	P1_3_SR	P1_3 IO 速度控制位 0: P1_3 快速 1: P1_3 慢速
2	P1_2_SR	P1_2 IO 速度控制位 0: P1_2 快速 1: P1_2 慢速
1	P1_1_SR	P1_1 IO 速度控制位 0: P1_1 快速 1: P1_1 慢速
0	P1_0_SR	P1_0 IO 速度控制位 0: P1_0 快速 1: P1_0 慢速

7.5.3 P2_SR

C007H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2_SR	P2_7_SR	P2_6_SR	P2_5_SR	-	P2_3_SR	P2_2_SR	-	P2_0_SR
读/写	读/写	读/写	读/写	读	读/写	读/写	读	读/写
复位值	1	1	1	0	1	1	0	1

位编号	位符号	说明
7	P2_7_SR	P2_7 IO 速度控制位 0: P2_7 快速 1: P2_7 慢速
6	P2_6_SR	P2_6 IO 速度控制位 0: P2_6 快速 1: P2_6 慢速
5	P2_5_SR	P2_5 IO 速度控制位 0: P2_5 快速 1: P2_5 慢速
4	-	-
3	P2_3_SR	P2_3 IO 速度控制位 0: P2_3 快速 1: P2_3 慢速
2	P2_2_SR	P2_2 IO 速度控制位 0: P2_2 快速 1: P2_2 慢速
1	-	-
0	P2_0_SR	P2_0 IO 速度控制位 0: P2_0 快速 1: P2_0 慢速

7.6 使用流程

7.6.1 IO 输出

1. 配置系统配置寄存器（PCLK1）的 GPIO 模块时钟
2. 配置系统配置寄存器(PRESET1)的 GPIO 模块复位使能
3. 配置 Px0EN 寄存器，对相应的 Bit 位写入 1，配置对应的 IO 为输出模式。
4. 配置 Px 寄存器，配置相应的 Bit 位，设置对应的 IO 为输出高/低电平。

7.6.2 IO 输入

1. 配置系统配置寄存器（PCLK1）的 GPIO 模块时钟
2. 配置系统配置寄存器(PRESET1)的 GPIO 模块复位使能
3. 配置 Px0EN 寄存器，对相应的 Bit 位写入 0，配置对应的 IO 为输入模式。
4. 配置 PxPU 寄存器，对相应的 Bit 位写入 0，使能对应的 IO 口内部上拉电阻。
5. 读取 Px 寄存器，获取相应的 Bit 位，即对应的 IO 的电平状态。

7.6.3 中断触发模式

1. 配置系统配置寄存器（PCLK1）的 GPIO 模块时钟
2. 配置系统配置寄存器(PRESET1)的 GPIO 模块复位使能
3. 配置 Px0EN 寄存器，对相应的 Bit 位写入 0，配置对应的 IO 为输入模式。
4. 配置 PxPU 寄存器，对相应的 Bit 位写入 0，使能对应的 IO 内部上拉电阻。
5. 配置 PxIEN 寄存器，对相应的 Bit 位写入 1，使能对应的 IO 端口中断功能。

8 TIMER0/1(定时器)

8.1 主要特性

- 2 个 16 位自动重载定时器 Timer0/1
- 可选系统时钟和内部低频时钟作为计数源
- 可工作在低功耗模式 (Sleep、DeepSleep)

8.2 定时器 0/1 工作方式

定时器 0, 1 的方式完全相同。下面以定时器 0 为例来说明定时器 0, 1 的工作方式, 定时器 0 是 16 位自动重载定时器, 可以设置预分频比。定时器 0 有内置 16 位计数/定时寄存器 (TH0, TL0)。当 TH0 和 TL0 被写入时, 用作定时器重载寄存器, 当被读时, 用作计数寄存器, 对 TR0 置位可启动定时器。当定时器计至 0xFFFF 时, 产生溢出并置位 TF0。同时会触发定时器 0 中断 (仅当定时器 0 中断打开时) 并重新装载初始值。

当定时器 0, 1 的时钟源选择 RC32K 时, 定时器 0, 1 可以工作在低功耗模式, 如果使能中断, 可以唤醒 MCU。

TH0 和 TL0 读写操作遵循以下顺序:

- 写操作: 先低位后高位
- 读操作: 先高位后低位

8.3 使用流程

1. 配置 TCON 寄存器 T0CLKS 位, 选择定时器 0 的计数时钟
2. 配置 TOMD 寄存器 T0PS 位, 设置定时器 0 预分频比
3. 配置 TL0 和 TH0 寄存器, 设置定时时间
4. 使能 EA 总中断
5. 使能 TIMER0 中断 ET0
6. 配置 TCON 寄存器 TR0 位, 使能定时器 0

9 BEEPER (蜂鸣器)

9.1 特性

内建蜂鸣器信号发生器，可硬件自动输出 1、2、4kHz 方波。

9.2 使用流程

BEEPER 使用流程：

1. 设置寄存器 BEEPCTR.BEEPEN，打开 BEEP 模块，P2.5 作为 BEEP 输出
2. 设置寄存器 BEEPCTR.BEEPCOLSET，选择 BEEP 输出极性。
3. 设置寄存器 BEEPCTR.BEEPSEL，选择 BEEP 输出频率。

10 UART0/1 (增强型串口)

10.1 特性

- UART0/1 均自带波特率发生器
- UART0 有四种工作模式
- UART1 有两种工作模式

两个串行口分别由一个移位寄存器，一个串行控制寄存器，一个波特率发生器以及两个独立的数据缓冲器(分别用于发送和接收数据)组成。两个数据缓冲器统称为 S0BUF(S1BUF)，其共用地址 99H(9CH)。向 S0BUF 或 S1BUF 写数据启动串口数据发送，读 S0BUF 或 S1BUF 返回缓冲器已经接收到的数据。

串行口在接收数据时，数据先进入移位寄存器，完成一帧的接收后将数据移入 S0BUF (S1BUF)，并立即接收下一帧数据，主机应保证该帧数据接收完成之前将 S0BUF (S1BUF) 缓冲器中数据取走，否则将导致前一帧数据被该帧数据覆盖而导致数据丢失。

10.2 UART0 工作模式

UART 有 4 种工作方式。进行通信之前用户须初始化相关寄存器，选择合适的工作方式和波特率。用户可以通过设置 SM0/SM1 来选择不同的工作方式。

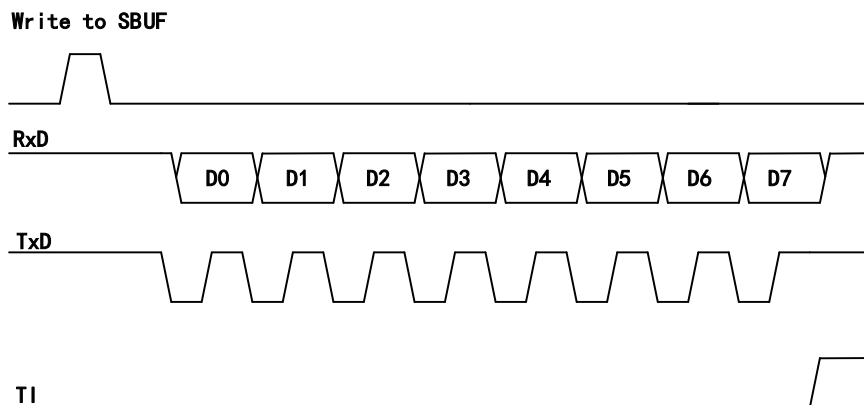
表 10-1: UART0 工作方式列表

SM0	SM1	Mode	描述	Baud Rate
0	0	0	Shift register	SYSCCLK /12
0	1	1	8-bit UART	可配置
1	0	2	9-bit UART	SYSCCLK/16
1	1	3	9-bit UART	可配置

方式 0: 同步，半双工通讯

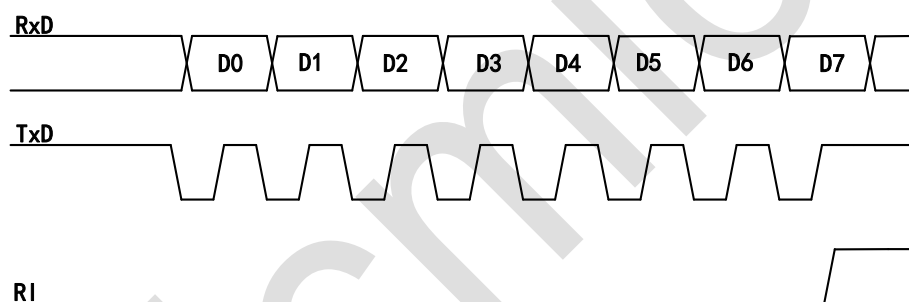
方式 0 支持与外部设备的同步通信。在 RX 引脚上收发串行数据，TX 引脚发送移位时钟。在这个方式中，每帧收发 8 位，低位先接收或发送。

任何将 S0BUF 作为目标寄存器的写操作都会启动发送，下一个系统时钟 TX 控制块开始发送。数据转换发生在移位时钟下降沿，移位寄存器内容逐次从左往右移位，空位置 0。发送完成后，TX 控制模块停止发送操作，然后在下一个系统时钟的上升沿将 TI 置位。



Transmit Timing of Mode 0

REN 置 1 和 RI 清 0 初始化接收。在移位时钟的上升沿锁存数据，接收转换寄存器的内容逐次向左移位。当所有的 8 位数据都移到移位寄存器后，RX 控制块停止接收，下一个系统时钟上升沿 RI 置位，直到软件清零才允许下一次接收。

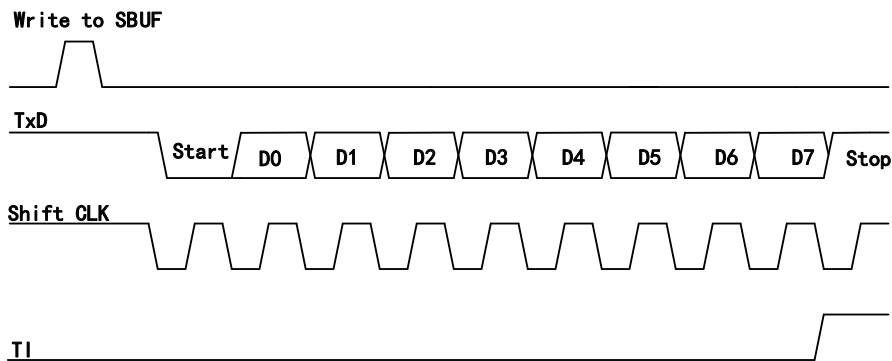


Receive Timing of Mode 0

方式 1: 8 位 UART, 可变波特率, 异步全双工

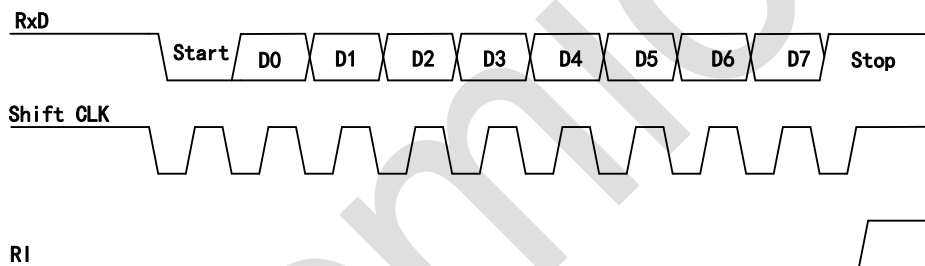
方式 1 提供 10 位全双工异步通信, 10 位由一个起始位 (逻辑 0), 8 个数据位 (低位为第一位), 一个停止位 (逻辑 1) 组成。在接收时, 8 个数据位保存在 S0BUF 中, 停止位保存在 RB8 中。

任何将 S0BUF 作为目标寄存器的写操作都会启动发送, 起始位首先在 TX 引脚上移出, 然后是 8 位数据位。在发送移位寄存器中的所有 8 位数据位都发送完后, 停止位在 TX 引脚上移出, 在停止位发出的同时 TI 标志置位发出中断请求。



Transmit Timing of Mode 1

只有 REN 置位时才允许接收。当 RX 引脚检测到下降沿时串口开始接收串行数据。若起始位有效，则移入移位寄存器，并接着移入其他位到移位寄存器。8 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 S0BUF 和 RB8 中，随后 RI 被置位。这时，接收器继续探测 RX 的下一个下降沿。用户需要软件清零 RI，然后才能再次接收。

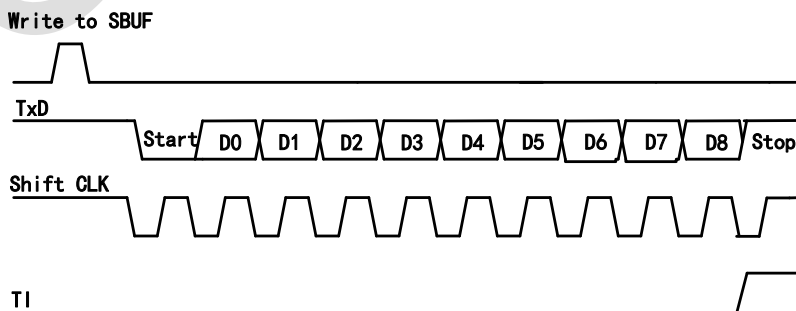


Receive Timing of Mode 1

方式 2: 9 位 UART, 固定波特率, 异步全双工

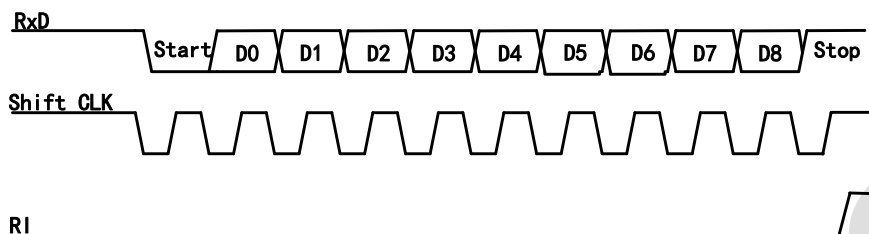
方式 2 提供 11 位全双工异步通信，波特率固定为系统时钟的 1/16。一帧由一个起始位（逻辑 0），8 个数据位（低位为第一位），一个可编程第 9 位和一个停止位（逻辑 1）组成。方式 2 和方式 3 支持多机通信功能。

任何将 S0BUF 作为目标寄存器的写操作都会启动发送，同时将 TB8 载入发送移位寄存器中的第九位。起始位首先在 TX 引脚移出，然后是 9 位数据位。在所有数据发送完成后，停止位在 TX 引脚上移出，在停止位开始发送时 TI 标志置位发出中断请求。



Transmit Timing of Mode 2

只有 REN 置位时才允许接收。当 RX 引脚检测到下降沿时串行口开始接收串行数据。若起始位有效，则移入移位寄存器，并接着移入其他位到移位寄存器。9 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 S0BUF 和 RB8 中，随后 RI 被置位。这时，接收器继续探测 RX 的下一个下降沿。用户需要软件清零 RI，然后才能再次接收。



Receive Timing of Mode 2

方式 3：9 位 UART，可变波特率，异步全双工

方式 3 使用方式 2 的传输协议以及方式 1 的波特率产生方式

10.3 UART1 工作模式

表 10-2: UART1 工作方式列表

SM	Mode	描述	Baud Rate
0	A	9-bit UART	可配置
1	B	8-bit UART	可配置

UART1 的这两种方式可以参考 UART0 的方式 3 和方式 1

10.4 多机通讯

UART0 的方式 2 和方式 3 以及 UART1 的方式 A 具有多机通讯功能。在采用多机通讯系统时，当主机要发送一数据块给数个从机中的一个时，先发送一个地址字节，以寻址目标从机。地址字节与数据字节可用第 9 位数据位来区别，地址字节的第 9 位为 1，数据字节的第 9 位为 0。接收方根据第 9 位信息判断是否接收，多机通讯过程如下：

设置为主机通信发送流程如下：

1. 设置为 9 bit Mode，发送接收方地址，且置 TB8 = 1；
2. 根据自定义的协议发送数据，且置 TB8 = 0。

设置为主机通信接收流程如下：

1. 设置 SM2=0，（无条件接收全部数据）；
2. 主机根据自定义协议解析数据。

设置为从机通信接收流程如下：

1. 置从机的 SM2=1，处于只接收地址帧状态，此时只会接收第 9bit 为 1 的地址数据。
2. 当接收到数据时，软件判断是否与本站设定的串口地址匹配。
3. 所有从机接收到地址帧后，各自将接收的地址与本站地址比较：若匹配即为目标从机，清除 SM2=0，准备接收主机即将发送的数据帧，接收完毕后再置 SM2=1；若不匹配，保持 SM2=1，忽略接下来的所有数据帧，不产生中断请求，直到接收到地址帧再次进行比较确认。

10.5 波特率

◆ UART0 Mode1and3

$$\text{Baud Rate} = \frac{\text{SYSCK}}{16 \times (1024 - \text{S0REL})}$$

◆ UART1

$$\text{Baud Rate} = \frac{\text{SYSCK}}{16 \times (1024 - \text{S1REL})}$$

SYSCLK 为 16M，常见波特率 SxREL 配置值以及实际误差如下表

表 10-3: 波特率误差表

目标波特率	SxREL	实际波特率	误差
115200	1015	111111	3.5%
57600	1007	58824	-2.1%
38400	998	38462	-0.2%
19200	972	19231	-0.2%
9600	920	9615	-0.2%
4800	816	4808	-0.16%
2400	607	2398	0.08%

10.6 使用流程

10.6.1 UART0 mode1 使用流程

10.6.1.1 串口初始化

1. 配置系统配置寄存器（PCLK0）的串口模块时钟。
2. 配置系统配置寄存器(PRESET0)的串口模块复位使能。
3. 配置 UARTEN 寄存器，打开串口引脚复用功能。
4. 选择串口的工作模式（S0CON.SM0 和 S0CON.SM1 位）。
5. 写 0 清除中断状态（S0CON.TI0 和 S0CON.RI0 位）。
6. 配置 S0RELL 和 S0RELH 寄存器，设置波特率。
7. 使能总中断（IEN0.EA）。

8. 使能串口 0 中断 (IEN0.ES0)。
9. 使能串口接收 (S0CON.REN0)。

10.6.1.2 串口发送字节

1. 发送、接收数据前软件可以配置波特率参数，中断使能。
2. 写入第一个字节数据到 S0BUF。
3. 查询发送完成标志 S0CON.TI0，如果 TI0=1 表示当前数据发送完成；软件清除此位（写 0 清除）。
4. 可以继续写入下一个字节到 S0BUF。

10.6.1.3 串口接收字节

1. 发送、接收数据前软件可以配置波特率参数，中断使能。
2. 接收数据，查询 S0CON.RI0 标志位，如果 RI0=1 则读取 S0BUF 中的数据，软件清除此位（写 0 清除）。
3. 继续接收数据。

10.6.2 UART1 modeB 使用流程

10.6.2.1 串口初始化

1. 配置系统配置寄存器 (PCLK0) 的串口模块时钟。
2. 配置系统配置寄存器(PRESET0)的串口模块复位使能。
3. 配置 UARTEN 寄存器，打开串口引脚复用功能。
4. 选择串口的工作模式 (S1CON.SM0 和 S1CON.SM1 位)。
5. 写 0 清除中断状态 (S1CON.TI0 和 S1CON.RI0 位)。
6. 配置 S1RELL 和 S1RELH 寄存器，设置波特率。
7. 使能总中断 (IEN0.EA)。
8. 使能串口 1 中断 (IEN0.ES1)。
9. 使能串口接收 (S1CON.REN1)。

10.6.2.2 串口发送字节

1. 发送、接收数据前软件可以配置波特率参数，中断使能。
2. 写入第一个字节数据到 S1BUF。
3. 查询发送完成标志 S1CON.TI1，如果 TI1=1 表示当前数据发送完成；软件清除此位（写 0 清除）。
4. 可以继续写入下一个字节到 S1BUF。

10.6.2.3 串口接收字节

1. 发送、接收数据前软件可以配置波特率参数，中断使能。
2. 接收数据，查询 S1CON.RI1 标志位，如果 RI1=1 则读取 S1BUF 中的数据，软件清除此位（写 0 清除）。
3. 继续接收数据。

11 SPI

11.1 主要特性

- 支持 SPI 标准协议，主从模式可配
- MISO、MOSI，单线传输，支持半双工、全双工传输，支持数据大小端配置。
- 可编程时钟极性和相位
- 支持 TX_ONLY 模式传输
- 支持 PCB 板延时补偿功能，slave 模式下，支持 SSN/SCK/MOSI 输入信号的组合逻辑滤波功能

11.2 寄存器描述

SPI 寄存器用 xdata 关键字访问

SPI 寄存器基地址：0xC400

11.2.1 SPCR1 控制寄存器

C400	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR1	BR[2:0]			SSN_MODE	LSBFIRST	MSTR	CPOL	CPHA
读/写	读/写			读/写	读/写	读/写	读/写	读/写
复位值	1			0	0	1	0	0

位编号	位符号	说明
7-5	BR[2:0]	master 模式波特率配置位： 000: $f_{SYSCLK} / 2$ 001: $f_{SYSCLK} / 4$ 010: $f_{SYSCLK} / 8$ 011: $f_{SYSCLK} / 16$ 100: $f_{SYSCLK} / 32$ 101: $f_{SYSCLK} / 64$ 110: $f_{SYSCLK} / 128$ 111: $f_{SYSCLK} / 256$ 当通信正在进行的时候，不能修改这些位
4	SSN_MODE	在 Master 模式下，SSN_MODE 表示在每传完 8Bit 后，SSN 是否会被拉高。 0: 在 Txbuf 为非空，且已经发送完毕 8Bit，如果 WAIT_CNT 不为 0，在等待 1+WAIT_CNT 个 SCK Cycle 后，SSN 被拉高 1: 在 Txbuf 为非空，且已经发送完毕 8Bit，SSN 被拉高
3	LSBFIRST	帧格式 (Frame format) 0: 先发送MSB (Bit7) 1: 先发送LSB (Bit0) 注：当通信在进行时不能改变该位的值

2	MSTR	Master/Slave 模式选择。 0: slave 1: master
1	CPOL	时钟极性选择。 0: 串行时钟 stop 在低电平 1: 串行时钟 stop 在高电平 注: 当通信在进行时不能改变该位的值
0	CPHA	时钟相位选择。 0: 第一个时钟边沿是第一个捕捉边沿 1: 第二个时钟边沿是第一个捕捉边沿 注: 当通信在进行时不能改变该位的值

PS:

1. 当出现 ERROR 后, 该寄存器保持不变, 若需要重新启动 spi, 软件先写 spi_en 为 0, 再写为 1。
2. 改变 CPOL, CPHA 后需要重新启动 spi, 软件先写 spi_en 为 0, 再写为 1。

11.2.2 SPCR2 控制寄存器

C401	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR2	SAMPLE_P	TXONLY_AUTO_CLR	SPI_EN	SSN_MCU_EN	WAIT_CNT	RSV	RSV	RSV
读/写	读/写	读/写	读/写	读/写	读/写	读	读	读
复位值	0	1	0	0	0	0	0	0

位编号	位符号	说明
7	SAMPLE_P Sdin_sample_mode	Master 模式下, 对 Slave 输出的 MISO 信号采样点位置的选择 SamplePosition。 0: 采样点与协议一致 1: 波特率速度低于 $f_{SYSCLK} / 2$ 时, 采样点延后半 spi sck 周期。 注: 当波特率为 $f_{SYSCLK} / 2$ 的时候, 该位不起作用。
6	TXONLY_AUTO_CLR (原 TXONLY_EN)	TXONLY 硬件自动清零的使能 0: 关闭 TXONLY 硬件自动清零 1: TXONLY 硬件自动清零有效, 软件打开 SPCR2 的 TXONLY 后, 等待发送完毕后, 硬件清零
5	SPI_EN	SPI 使能。采用关闭时钟的方式来关闭使能。 0: 关闭 SPI。复位状态、清空 TxBuf 清空 RxBuf 1: 使能 SPI
4	SSN_MCU_EN	在 Master 模式下, 由软件控制 SSN 端口的使能: 1: 由软件控制 SSN 输出, 使能有效 0: 由内部硬件控制 SSN 输出
3-2	WAIT_CNT	在 Master 模式下, 每发完 8Bit 后加入 WAIT 再传输下一个 8Bit 的数据 00: 无等待 01: 中间加入 2 个 SCK Cycle 等待 10: 中间加入 3 个 SCK Cycle 等待 11: 中间加入 4 个 SCK Cycle 等待
1-0	RSV	保留

PS:

1. 当出现 ERROR 后, 该寄存器保持不变, 若需要重新启动 spi, 软件先写 spi_en 为 0, 再写为 1。

11.2.3 SPCR3 控制寄存器

C402	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR3	SCK_EN	MOSI_EN	MISO_EN	CS_EN	TX_ONLY	SSN_MCU	Signal_filter	send_p
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写	读/写
复位值	1	1	1	1	0	1	1	0

位编号	位符号	说明
7	SCK_EN	P1_3 作为 SPI_SCK 信号使能。当此位为 1, 且 SPI_EN 为 1 时, P1_3 作为 SPI_SCK 信号;
6	MOSI_EN	P1_4 作为 SPI_MOSI 信号使能。当此位为 1, 且 SPI_EN 为 1 时, P1_4 作为 SPI_MOSI 信号;
5	MISO_EN	P1_5 作为 SPI_MISO 信号使能。当此位为 1, 且 SPI_EN 为 1 时, P1_5 作为 SPI_MISO 信号;
4	CS_EN	P0_3 作为 SPI_CSN 信号使能。当此位为 1, 且 SPI_EN 为 1 时, P0_3 作为 SPI_CSN 信号;
3	TX_ONLY	限制 SPI 仅启动发送 0: 禁止 TX_ONLY 模式 1: 使能 TX_ONLY 模式
2	SSN_MCU	在 Master 模式下, 如果 SSN_MCU_EN=1, MCU 可以通过写此位控制 SSN 输出端口: 0: SSN 被软件写成 0 1: SSN 被软件写成 1
1	Signal_filter	是否对 SSN/SCK/MOSI 上可能产生的毛刺数字滤波 0: 不滤波 1: 滤波
0	send_p Sdout_send_mode	SLAVE 模式下, 对 slave 输出的 MOSI 信号使用 transmit 起始点时钟 0: 按照协议的时钟点进行发送 1: 提前半个周期进行发送

PS:

1. 当出现 ERROR 后, 该寄存器保持不变, 若需要重新启动 spi, 软件先写 spi_en 为 0, 再写为 1。

11.2.4 SPCR4 控制寄存器

C403	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR4	RSV	RSV	RSV	RSV	CLR_TXBUF	CLR_RXBUF	RSV	RSV
读/写	读	读	读	读	写	写	读	读
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:4	RSV	保留
3	CLR_TXBUF	写 1 清除 tx_buf 所有内容, 并且清除 TXBUF_EMPTY 标志位, 硬件自动回 0
2	CLR_RXBUF	写 1 清除 rx_buf 所有内容, 并且清除 RXBUF_FULL 标志位, 硬件自动回 0

1:0	RSV	保留
-----	-----	----

注：中，均指对应的收发 buffer 及标志；

PS：当出现 ERROR 后，TXONLY 位会被硬件自动重置为 0。所以，当重新启动 spi 时，该寄存器需要重新配置。

11.2.5 SPIE 中断使能寄存器

中断使能寄存，中根据此寄存器的使能产生中断

C404	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPIE	RSV	RSV	RSV	RSV	RSV	ERROR_I E	TX_E_IE	RX_NE_ IE
读/写	读	读	读	读	读	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7:3	RSV	保留
2	ERROR_IE	错误中断使能，包括 TXBUF/RXBUF 溢出，Master/Slave Err 0：关掉中断 1：使能中断
1	TX_E_IE	发送 TXBUF 空中断使能 0：关掉中断 1：使能中断
0	RX_NE_IE	接收 RXBUF 非空中断使能 0：关掉中断 1：使能中断

11.2.6 SPSR 状态寄存器

C405	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	RSV	RSV	RSV	RXBUF_ WCOL	TXBUF_ WCOL	BUSY	TXBUF_E MPTY	RXBUF_ FULL
读/写	读	读	读	读/写	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	1	0

位编号	位符号	说明
7:5	RSV	保留
4	RXBUF_WCOL	RX 溢出，软件写 0 清 0
3	TXBUF_WCOL	当 TXBUF 为满时，MCU 进行写 TXBUF 操作。 1：冲突 0：无冲突 软件写 0 清 0
2	BUSY	指示 0：表示 TXBUF 为空，且 SPI 没有在传输数据 1：表示 TXBUF 不为空，或者 SPI 在传输数据 slave 模式下根据 SSN 和 TXBUF，SSN 为高，且 TXBUF 为空，BUSY 信号为 0

1	TXBUF_EMPTY	写 txbuf 动作可清除该标志位 0: TXBUF 中有数据等待发送 1: TXBUF 中无数据, 可以写入
0	RXBUF_FULL	读 rxbuf 动作可清除该标志位 0: RXBUF 中无数据 1: RXBUF 中有数据

11.2.7 TXBUF 发送数据寄存器

MCU 把需要发送的数据写入该寄存器, 达到把发送数据缓存到 TxBuf 的目的。TxBuf 无实际寄存器, 只支持写操作。

C406	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TXBUF	WR_DATA							
读/写	写							
复位值	0							

位编号	位符号	说明
7-0	WR_DATA	该寄存器在写的时候, 根据 TxBuf 的空满状态, 如果 TxBuf 已经为满, 将产生溢出中断

11.2.8 RXBUF 接收数据缓冲寄存器

通过 Spi 接口接收的数据, 会先缓存到 RxBUF 中, 当 Spi 接口完成一个字节的接收后, 会往 RXBUF 写入一个字节的数据。MCU 通过读该寄存器, 可以得到从 Spi 接收到的数据。

C407	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RXBUF	D7-D0							
读/写	读							
复位值	0							

位编号	位符号	说明
7:0	D7-D0	MCU 根据 RXBUF 的非空中断, 对该地址进行读操作, 读到内部 RxBuf 中的数据

11.3 使用流程

11.3.1 SPI 初始化

1. 配置 PCLK0.SPICEN 为 1, SPI 时钟使能。
2. 配置 PRESET0.SPIREN 为 1, SPI 复位释放。
3. SPI 主模式初始化:
 - A. 配置 SPCR1.MSTR 为主模式。
 - B. 配置 SPCR1.LSBFIRST 选择帧格式。

- C. 配置 SPCR1.BR[2:0]选择波特率分频。
 - D. 配置 SPCR1.CPOL 和 SPCR1.CPHA 选择 SPI 工作模式 0/1/2/3。
 - E. 配置 SPCR2.SSN_MCU_EN 选择软件控制 SSN(片选)。
 - F. 配置 SPCR2.SPIEN 使能 SPI。
4. SPI 从模式初始化:
- A. 配置 SPCR1.MSTR 为从模式。
 - B. 配置 SPCR1.LSBFIRST 选择帧格式。
 - C. 配置 SPCR1.CPOL 和 SPCR1.CPHA 选择 SPI 工作模式 0/1/2/3。
 - D. 配置 SPCR2.SPIEN 使能 SPI。

11.3.2 SPI 发送流程

1. 等待 SPSR.TXBUF_EMPTY 置 0。
2. 往 TXBUF 发送数据寄存器写入数据。
3. 等待 SPSR.BUSY 置 0(从模式不需要等这一步)。

11.3.3 SPI 接收流程

1. 等待 SPSR.RXBUF_FULL 置 1。
2. 读取 RXBUF 接收数据寄存器。

12 LPTIM（低功耗定时器）

12.1 概述

LPTIM 是运行在 Always-On 电源域下的 16bits 低功耗定时/计数器模块。通过选择合适的工作时钟，LPTIM 在各种低功耗模式下保持运行，并且只消耗很低的功耗。LPTIM 甚至可以在没有内部时钟的条件下工作，因此可实现休眠模式下的外部脉冲计数功能。此外，与外部输入的触发信号结合，可以实现低功耗超时唤醒功能。

12.2 主要特性

- 16bit upcounter
- 3bit 异步时钟预分频器，8 种分频系数（1、2、4、8、16、32、64、128）
- 可选工作时钟：
- 内部时钟源：LSCLK(RCL32K)、RCLP(CLK_1HZ)、系统时钟
- 外部时钟源：LPTIN（带有模拟滤波）
- 16bit 比较寄存器
- 16bit 目标值寄存器
- 软件/硬件触发
- 输入极性选择
- 无时钟外部脉冲计数
- 外部触发的休眠超时唤醒
- 支持 16bit PWM

12.3 工作模式

12.3.1 普通定时器

- 使用内部时钟或外部时钟输入工作
- 使能后有两个计数时钟的同步过程
- 使能后即开始工作，不需要 trigger 触发

12.3.2 Trigger 脉冲触发计数

- 使用内部时钟工作

- 内部时钟采样外部输入的异步 trigger 信号
- 可以对 trigger 的上升、下降、双边沿计数
- 使能后有两个计数时钟的同步过程

12.3.3 外部异步脉冲计数

- 直接使用外部输入脉冲作为计数工作时钟
- 输入极性可配置，实现上升沿计数或下降沿计数
- 不需要 trigger 触发
- 使能后无同步过程

12.3.4 Timeout 模式

- 使用内部时钟或外部输入时钟工作
- 采样外部输入的异步 trigger 信号
- 首次 trigger 启动计数器，启动后采样到 trigger 则清零并重启计数器
- 计数器溢出前没有出现新的 trigger，则产生溢出中断并停止计数，清除使能
- 使能后有两个计数时钟的同步过程

12.3.5 计数模式

LPTIM 有两种计数模式。

-连续计数模式：计数器被触发后保持运行，直到被关闭为止。计数器达到目标值后回到 0 重新开始计数，并产生溢出中断。

-单次计数模式：计数器被触发后计数到目标值后回到 0，并自动停止，产生溢出中断。

由于溢出信号和 lpten 使能信号位于不同的时钟域，关闭使能信号采用异步复位同步释放的方式实现。

12.3.6 外部触发的超时唤醒

LPTIM 可以由外部输入的 trigger 信号触发使能，也可以由软件触发使能。在 Timeout 模式下，第一个外部触发输入的有效沿将启动计数器，而后续触发信号将清零计数器。如果在计数器达到比较值之前没有有效触发信号到来，则产生超时中断，唤醒 MCU。

外部输入 trigger 信号的有效沿可以由寄存器配置，外部 trigger 信号被认为是一个异步输入，因此有效沿的采样和判决有至少 2 个计数时钟的 latency。

12.3.7 16bit PWM

使能 PWM 模式后 LPTIM 从 0x0000 开始计数，计数值等于比较值时输出置高，计数值等于终值寄存器时输出变低；PWM 周期由终值寄存器决定，占空比由比较值寄存器决定。

12.4 寄存器描述

LPTIMER 寄存器用 xdata 关键字访问

LPTIMER 寄存器基地址：0xC800

地址	名称	符号
0Xc800	LPTIM 配置寄存器	LPTCFG0
0Xc801	LPTIM 配置寄存器	LPTCFG1
0xc802	LPTIM 计数低位寄存器	LPTCNTL
0xc803	LPTIM 计数高位寄存器	LPTCNTH
0xc804	LPTIM 比较值低位寄存器	LPTCML
0xc805	LPTIM 比较值高位寄存器	LPTCMPH
0xc806	LPTIM 目标值低位寄存器	LPTTARGETL
0xc807	LPTIM 目标值高位寄存器	LPTTARGETH
0xc808	LPTIM 中断使能寄存器	LPTIE
0xc809	LPTIM 中断标志寄存器	LPTIF
0xc80a	LPTIM 控制寄存器	LPTCTRL

12.4.1 LPTCFG0 寄存器

C800H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPTCFG0	TRIGCFG		EDGESEL	CLKSEL		DIVSEL		
读/写	读/写		读/写	读/写		读/写		
复位值	0		0	1	0	0		

位编号	位符号	说明
7-6	TRIGCFG	外部触发边沿选择 00: 外部输入信号上升沿 trigger 01: 外部输入信号下降沿 trigger 10/11: 外部输入信号上升下降沿 trigger 当通信正在进行的时候, 不能修改这些位
5	EDGESEL	LPTIN 输入边沿选择 0: LPTIN 的上升沿计数 1: LPTIN 的下降沿计数
4:3	CLKSEL	时钟源选择 00: LSCLK (RCL32K)作为计数时钟 01: RCLP (CLK_1HZ)作为计数时钟 10: 系统时钟作为计数时钟 11: LPTIN (外部输入)作为计数时钟

2:0	DIVSEL	计数时钟分频选择 000: 1 分频 001: 2 分频 010: 4 分频 011: 8 分频 100: 16 分频 101: 32 分频 110: 64 分频 111: 128 分频
-----	--------	---

12.4.2 LPTCFG1 寄存器

C801H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPTCFG1	EXTRIGGER_I O_IEN	LPOUT_IO _IEN	LPTIN_I O_IEN	TMODE		MODE	PWM	POLARIT Y
读/写	读/写	读/写	读/写	读/写		读/写	读/写	读/写
复位值	0	0	0	0		0	0	0

位编号	位符号	说明
7	EXTRIGGER_IO_IEN	1: P1_1 作为 EXTRIGGER_IO_IEN
6	LPOUT_IO_IEN	1: P0_3 作为 LPOUT
5	LPTIN_IO_IEN	1: P1_0 作为 LPTIN
4:3	TMODE	工作模式选择 00: 带波形输出的普通定时器模式 01: Trigger 脉冲触发计数模式 10: 外部异步脉冲计数模式 11: Timeout 模式
2	MODE	计数模式 0: 连续计数模式: 计数器被触发后保持运行, 直到被关闭为止。计数器达到目标值后回到 0 重新开始计数, 并产生溢出中断。 1: 单次计数模式: 计数器被触发后计数到目标值后回到 0, 并自动停止, 产生溢出中断。
1	PWM	脉宽调制模式 0: 周期方波输出模式 1: PWM 输出模式
0	POLARITY	计数时钟分频选择 0: 正极性波形, 即第一次计数值=比较值时产生输出波形上升沿 1: 负极性波形, 即第一次计数值=比较值时产生输出波形下降沿

12.4.3 LPTCNT 计数值寄存器

-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPTCNTL(C802H)	CNT16[7:0]							
LPTCNTH(C803H)	CNT16[15:8]							
读/写	读							
复位值	0							

位编号	位符号	说明
15:0	CNT16	计数器数值

12.4.4 LPTCMP 比较值寄存器

-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPTC MPL(C804H)	COMPARE_REG[7:0]							
LPTC MPH(C805H)	COMPARE_REG[15:8]							
读/写	读/写							
复位值	0							

位编号	位符号	说明
15:0	COMPARE_REG	比较值寄存器

12.4.5 LPTTARGET 目标值寄存器

-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPTTARGETL(C806H)	TARGET_REG[7:0]							
LPTTARGETH(C807H)	TARGET_REG[15:8]							
读/写	读/写							
复位值	0							

位编号	位符号	说明
15:0	TARGET_REG	目标值寄存器

12.4.6 LPTIE 中断使能寄存器

中断使能寄存，中根据此寄存器的使能产生中断

C808H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPTIE	RSV					TRIGIE	OVIE	COMPIE
读/写	读					读/写	读/写	读/写
复位值	0					0	0	0

位编号	位符号	说明
7:3	RSV	保留
2	TRIGIE	外部触发到来中断使能位 1: 外部触发到来中断使能 0: 外部触发到来中断禁止
1	OVIE	计数器溢出中断使能位 1: 计数器溢出中断使能 0: 计数器溢出中断禁止

0	COMPIE	比较匹配中断使能位 1: 计数器值和比较值匹配中断使能 0: 计数器值和比较值匹配中断禁止
---	--------	---

12.4.7 LPTIF 中断标志寄存器

C809H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPTIF	RSV					TRIGIF	OVIF	COMPIF
读/写	读					读/写	读/写	读/写
复位值	0					0	0	0

位编号	位符号	说明
7:3	RSV	保留
2	TRIGIF	外部触发到来中断标志位，写 1 清零 1: 外部触发到来中断产生 0: 无中断产生
1	OVIF	计数器溢出中断标志位，写 1 清零 1: 计数器溢出中断产生 0: 无中断产生
0	COMPIF	比较匹配中断标志位，写 1 清零 1: 计数器值和比较值匹配中断产生 0: 无中断产生

12.4.8 LPTCTRL 控制寄存器

C80AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPTCTRL	RSV							LPTEN
读/写	读							读/写
复位值	0							0

位编号	位符号	说明
7:1	RSV	保留
0	LPTEN	LPTIM 使能位 1: 使能计数器计数 0: 禁止计数器计数

12.5 使用流程

12.5.1 普通定时器

1. 配置系统配置寄存器（PCLK0）的 LPTIMER 模块时钟使能。
2. 配置系统配置寄存器(PRESET0)的 LPTIMER 模块复位释放。
3. 配置 LPTCFG0 寄存器 CLKSEL 位，选择时钟源。

4. 配置 LPTCFG0 寄存器 DIVSEL 位, 设置时钟分频值。
5. 配置 LPTCFG1 寄存器 TMODE 位, 选择普通定时器模式。
6. 配置 LPTCFG1 寄存器 MODE 位, 选择计数模式。
7. 配置目标值寄存器 LPTARGETL 和 LPTARGETH, 设置定时时间。
8. 使能 LPTIMER 中断 IEN1。
9. 配置 LPTIE 寄存器 OVIE 位, 使能定时器溢出中断。
10. 配置 LPTCTRL 寄存器 LPTEN 位, 启动计数器计数。

12.5.2 PWM 输出

1. 配置系统配置寄存器 (PCLK0) 的 LPTIMER 模块时钟使能。
2. 配置系统配置寄存器(PRESET0)的 LPTIMER 模块复位释放。
3. 配置 LPTCFG0 寄存器 CLKSEL 位, 选择时钟源。
4. 配置 LPTCFG0 寄存器 DIVSEL 位, 设置时钟分频值。
5. 配置 LPTCFG1 寄存器 TMODE 位, 选择普通定时器模式。
6. 配置 LPTCFG1 寄存器 MODE 位, 选择计数模式。
7. 配置 LPTCFG1 寄存器 PWM 位, 选择 PWM 输出模式。
8. 配置 LPTCFG1 寄存器 POLARITY 位, 选择 PWM 极性。
9. 配置目标值寄存器 LPTARGETL 和 LPTARGETH, 设置 PWM 的周期时间。
10. 配置目标值寄存器 LPTCmpl 和 LPTCmpH, 设置 PWM 占空比时间。
11. 配置 LPTCFG1 寄存器 LPOUT_IO_IEN 位, 将 P03 复用为 LPOUT 输出。
12. 配置 LPTCTRL 寄存器 LPTEN 位, 启动计数器计数。

12.5.3 Trigger 脉冲触发计数模式

1. 配置系统配置寄存器 (PCLK0) 的 LPTIMER 模块时钟使能。
2. 配置系统配置寄存器(PRESET0)的 LPTIMER 模块复位释放。
3. 配置 LPTCFG0 寄存器 CLKSEL 位, 选择时钟源。
4. 配置 LPTCFG0 寄存器 DIVSEL 位, 设置时钟分频值。
5. 配置 LPTCFG1 寄存器 TMODE 位, 选择 Trigger 脉冲触发计数模式。
6. 配置 LPTCFG1 寄存器 MODE 位, 选择计数模式。
7. 配置 LPTCFG0 寄存器 TRIGCFG 位, 选择外部触发边沿方式。
8. 配置目标值寄存器 LPTARGETL 和 LPTARGETH, 设置定时时间。

9. 配置 LPTCFG1 寄存器 EXTRIGGER_IO_IEN 位, 将 P11 复用为 EXTRIGGER 输入。
10. 使能 LPTIMER 中断 IEN1。
11. 配置 LPTIE 寄存器 TRIGIE 位, 使能外部触发中断。
12. 配置 LPTCTRL 寄存器 LPTEN 位, 启动计数器计数。

12.5.4 外部异步脉冲计数模式

1. 配置系统配置寄存器 (PCLK0) 的 LPTIMER 模块时钟使能。
2. 配置系统配置寄存器(PRESET0)的 LPTIMER 模块复位释放。
3. 配置 LPTCFG0 寄存器 CLKSEL 位, 选择时钟源为 LPTIN。
4. 配置 LPTCFG0 寄存器 DIVSEL 位, 设置时钟分频值。
5. 配置 LPTCFG1 寄存器 TMODE 位, 选择外部异步脉冲计数模式。
6. 配置 LPTCFG1 寄存器 MODE 位, 选择计数模式。
7. 配置 LPTCFG0 寄存器 EDGESEL 位, 选择 LPTIN 输入边沿选择方式。
8. 配置目标值寄存器 LPTTARGETL 和 LPTTARGETH, 设置定时时间。
9. 配置 LPTCFG1 寄存器 LPTIN_IO_IEN 位, 将 P10 复用为 LPTIN 输入。
10. 使能 LPTIMER 中断 IEN1。
11. 配置 LPTIE 寄存器 OVIE 位, 使能定时器溢出中断。
12. 配置 LPTCTRL 寄存器 LPTEN 位, 启动计数器计数。

注:此模式可唤醒 Stop 模式。

12.5.5 Timeout 模式

1. 配置系统配置寄存器 (PCLK0) 的 LPTIMER 模块时钟使能。
2. 配置系统配置寄存器(PRESET0)的 LPTIMER 模块复位释放。
3. 配置 LPTCFG0 寄存器 CLKSEL 位, 选择时钟源。
4. 配置 LPTCFG0 寄存器 DIVSEL 位, 设置时钟分频值。
5. 配置 LPTCFG1 寄存器 TMODE 位, 选择 Timeout 模式。
6. 配置 LPTCFG1 寄存器 MODE 位, 选择计数模式。
7. 配置 LPTCFG0 寄存器 TRIGCFG 位, 选择外部触发边沿方式。
8. 配置目标值寄存器 LPTTARGETL 和 LPTTARGETH, 设置定时时间。
9. 配置 LPTCFG1 寄存器 EXTRIGGER_IO_IEN 位, 将 P11 复用为 EXTRIGGER 输入。
10. 使能 LPTIMER 中断 IEN1。

11. 配置 LPTIE 寄存器 OVIE 位, 使能定时器溢出中断。
12. 配置 LPTCTRL 寄存器 LPTEN 位, 启动计数器计数。

Unichmicro

13 I2C

13.1 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。本模块支持主模式和从模式。

13.2 主要特性

- I2C 主设备功能
- I2C 从设备功能：
 - 可编程的 I2C 从设备地址
 - 可编程的 NACK/ACK 回复
- 输入 SCL 总线滤波功能
- 支持 Standard/Fast 模式
- 支持 7bit 设备地址
- 支持多主模式

13.3 寄存器描述

I2C 寄存器用 xdata 关键字访问

I2C 寄存器基地址：0xCC00

表 13-1: I2C 寄存器列表

偏置	名称	描述
0x00	I2C_SLAVE_ADDR1	I2C设备地址寄存器1
0x01	I2C_CLK_DIV	I2C的SCL速率分频
0x02	I2C_CR0	控制寄存器0
0x03	I2C_CR1	控制寄存器1
0x04	I2C_SR0	状态寄存器0
0x05	I2C_SR1	状态寄存器1
0x06	I2C_DR	数据寄存器
0x07	I2C_SLAVE_ADDR2	I2C设备地址寄存器2

13.3.1 I2C_SLAVE_ADDR1 从设备地址寄存器 1

CC00H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C_SLAVE_ADDR1	ADD1[7:1]							RSV
读/写	读/写							读

复位值	0	0
位编号	位符号	说明
7-1	ADD1[7:1]	地址的 7~1 位
0	RSV	保留

13.3.2 I2C_CLK_DIV 时钟分频寄存器

CC01H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C_CLK_DIV	I2C_CLK_DIV							
读/写	读/写							
复位值	0							

位编号	位符号	说明
7-0	I2C_CLK_DIV	SCL 分频值，通过配置该寄存器设置 I2C 的传输速率。F _{scl} = (F 系统时钟) / (4 * (DIV 寄存器值+1))

13.3.3 I2C_CR0 控制寄存器 0

CC02H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C_CR0	MAAS1_INT_En	MIEN	RSTA	TACK	MTX	MSTA	RSV	MEN
读/写	读/写	读/写	读/写	读/写	读/写	读/写	读	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	MAAS1_INT_En	MAAS1（接收到的设备地址与从设备地址寄存器1相符）中断使能： 0: MAAS1中断不使能 1: MAAS1 中断使能
6	MIEN	MTF（字节数据（包括地址）传输完成）中断使能： 0: MTF中断不使能 1: MTF 中断使能
5	RSTA	RSTA产生位： 0: 不产生Repeat Start条件 1: 写该位后，在发送或接收完一个字节后，产生一个 Start。
4	TACK	传输应答位/STOP条件位； 对于主模式： 0: 接收一字节后，在应答周期产生ACK； 1: 主设备在发送完当前字节后，将产生一个STOP； 对于从模式： 0: 接收一字节后，在应答周期产生ACK； 1: 接收完一字节后，在应答周期产生 NACK。
3	MTX	0: 设备作为接收器 1: 设备作为发送器 当作为从设备时，处理器应该查询 I2C_SR 的 SRW 位，判断是作为发送器还是接收器，然后设置与之匹配的 MTX 位

2	MSTA	主从设备选择位，START位 0: 从模式 1: 主模式。 如果这位从 0 变成 1 时，模块产生一个 START 条件。当 STOP 条件产生时，MSTA 将被清零。当从地址匹配（MAAS1 或 MAAS2 为 1）时，MSTA 也被清零。
1	RSV	保留
0	MEN	0: 设备不使能 1: 设备使能

13.3.4 I2C_CR1 控制寄存器 1

CC03H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C_CR1	RSV	MTX_ANTO_EN	OD_MODE	RSV	MAAS2_INT_EN	WBT_INT_EN	RXNE_INT_EN	TXE_INT_EN
读/写	读	读/写	读/写	读	读/写	读/写	读/写	读/写
复位值	0	0	0	0	0	0	0	0

位编号	位符号	说明
7	RSV	保留
6	MTX_ANTO_EN	当此模块 SLAVE 模式下，传输数据和地址 SDA 线自动切换。当此模块 SLAVE 模式下，此位设置为 1，主设备传输完地址信息后，根据总线上的 RW 位，自动切换 SDA 传输方向。 0: 不使能自动切换功能； 1: 使能自动切换功能；
5	OD_MODE	SCL 与 SDA 输出模式选择： 0:push-pull 模式输出 1:open-drain 模式输出
4	RSV	保留
3	MAAS2_INT_EN	MAAS2（接收到的设备地址与从设备地址寄存器 2 相符）中断使能： 0: MAAS2 中断不使能 1: MAAS2 中断使能
2	WBT_INT_EN	WBT（字节传输完成且 TXE 或 RXNE 为 1）中断使能： 0: WBT 中断不使能 1: WBT 中断使能
1	RXNE_INT_EN	RXNE（接收时数据寄存器非空）中断使能： 0: RXNE 中断不使能 1: RXNE 中断使能
0	TXE_INT_EN	TXE（发送时数据寄存器空）中断使能： 0: TXE 中断不使能 1: TXE 中断使能

13.3.5 I2C_SR0 状态寄存器 0

CC04H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C_SR0	MAAS2	MTF	MAAS1	MBB	RSV	SRW	MTF_H	RACK
读/写	读/写	读/写	读/写	读	读	读	读/写	读
复位值	0	0	0	0	0	0	0	1

位编号	位符号	说明
7	MAAS2	0: 设备地址2和接收到的地址不相等 1: 设备地址2和接收到的地址相等 写1清0
6	MTF	0: 字节传输未完成 1: 字节传输完成 当一个字节数据（包括地址）正在传输时，该位为 0；在一个字节传输完后，在第 9 个 SCL 时钟下降沿（ACK 周期）MTF 被置为 1。比 MTF_H 晚半个周期 SCL 周期，写 1 清除
5	MAAS1	0: 设备地址1和接收到的地址不相等 1: 设备地址1和接收到的地址相等 写1清0
4	MBB	0: 总线上无数据通信（检测到总线上STOP标志，此位清0） 1: 总线上正在进行数据通信（检测到总线上 START 标志，此位清 1）
3	RSV	保留
2	SRW	0: 不作为从设备发送器 1: 作为从设备发送器 当地址匹配后，SRW 指示地址呼叫命令中的 R/W 位，该位仅在如下条件有效：一个完整的传输已经发生，没有其他传输被初始化；并且 I2C 被配置为从模式，且从地址匹配。当接收到 STOP 条件或一个新的 START 条件，该位自动清除
1	MTF_H	快速字节传输完全完成标志 0: 快速字节传输未完成（提早半个SCL时钟） 1: 快速字节传输完成。MTF_H在第9个SCL时钟上降沿（ACK周期）产生，比MTF早半个周期 通过对此位写 1 或写 MTF 为 1 清除
0	RACK	应答接收状态位 0: 最近的发送应答周期接收到应答 1: 最近的发送应答周期没有接收到应答 只有 START 条件将清除 RACK 位

13.3.6 I2C_SR1 状态寄存器 1

CC05H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C_SR1	RSV					WBT	RXNE	TXE
读/写	读					读/写	读/写	读/写
复位值	0					0	0	0

位编号	位符号	说明
7-3	RSV	保留

2	WBT	0: 字节传输未完成或字节传输完成但TXE, RXNE不为1 1: 字节传输完成且TXE或RXNE为1 通过读写数据寄存器I2C_DR可以清除该位, 也可以写1清0
1	RXNE	0: 接收时数据寄存器空 1: 接收时数据寄存器非空 硬件置位, 通过读数据寄存器I2C_DR可以清除该位, 写1清0
0	TXE	0: 发送时数据寄存器非空 1: 发送时数据寄存器空 硬件置位, 通过写数据寄存器 I2C_DR 可以清除该位, 写 1 清 0

13.3.7 I2C_DR 数据寄存器

CC06H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C_DR	I2CDR							
读/写	读/写							
复位值	0							

位编号	位符号	说明
7-0	I2CDR	I2C 数据寄存器值

13.3.8 I2C_SLAVE_ADDR2 从设备地址寄存器 2

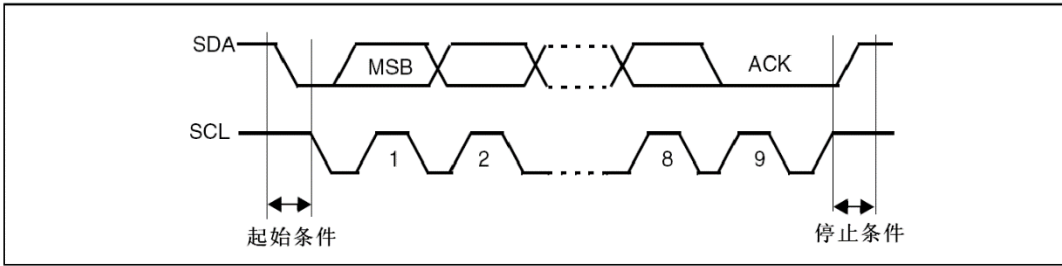
CC07H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2C_SLAVE_ADDR2	ADD2[7:1]							ADDR2_EN
读/写	读/写							读/写
复位值	0							0

位编号	位符号	说明
7:1	ADD2[7:1]	地址的 7~1 位
0	ADDR2_EN	0: SLAVE_ADDR2地址匹配不使能 1: SLAVE_ADDR2 地址匹配使能

13.4 功能描述

13.4.1 模式选择

该模块默认为从模式, 当软件发起START操作后设备变成主模式。主模式时, I2C接口启动数据传输并产生时钟信号, 并可以发出STOP信号停止传输。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:



13.4.2 I2C 从模式

一旦检测到起始条件，在SDA线上接收到的地址被送到移位寄存器。然后与芯片自己的设备地址相比较，如果地址不匹配I2C将其忽略并等待另一个起始条件。如果地址匹配，则此控制器会检测当前操作是发送还是接收（SRW寄存器），I2C接口进行如下操作：

从发送器：

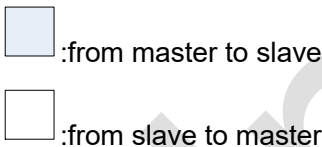
发送器将字节从数据寄存器load到内部移位寄存器发送到SDA线上，并产生数据寄存器空标志TXE，软件需要更新数据寄存器来清除TXE标志。

当收到应答脉冲后，如果在下一个数据发送结束之前新数据仍然没有被写进数据寄存器，即TXE仍然为1，则字节等待标志位（WBT）被置1，这时I2C接口保持SCL为低以等待新的数据被写进数据寄存器。



图 13-1：7 位从发送器的传送图

说明：S=Start(起始条件)， P=Stop(停止条件)， A=响应， NA=非响应



从接收器：

在接收到数据后，从接收器将通过内部移位寄存器从SDA线接收到的字节latch到数据寄存器，并产生数据寄存器非空标志RXNE，软件需要读出数据寄存器的值来清除RXNE标志。

I2C接口在接收到每个字节后都产生一个应答脉冲。

如果在接收新数据前数据寄存器的值未被读出，即RXNE仍然为1，则字节等待标志位（WBT）被置1，这时I2C接口保持SCL为低以等待数据寄存器的值被读出。



图 13-2：7 位从接收器的传送图

说明：S=Start(起始条件)， P=Stop(停止条件)， A=响应， NA=非响应

关闭从通信

在传输完最后一个数据字节后，主设备发出一个停止操作，I2C接口检测到这一条件时释放SCL和

SDA线。

13.4.3 I2C 主模式

在主模式时，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。当通过START位在总线上发起起始操作，设备就进入了主模式。

以下是主模式的操作顺序：

- 配置时钟控制寄存器。
- 配置数据寄存器（内部为从设备的地址和读写控制位）。
- 配置控制寄存器的MSTA位为1，产生起始条件。

主发送器

发送了地址后，主设备通过内部移位寄存器将字节从数据寄存器load到SDA线上，并产生数据寄存器空标志TXE，软件需要更新数据寄存器来清除TEX标志。

收到应答脉冲后确认新的数据已经发送到数据寄存器。如果在下一个数据发送结束之前新数据仍然没有被写进数据寄存器，即TXE仍然为1，则字节等待标志位（WBT）被置1，这时I2C接口保持SCL为低以等待新的数据被写进数据寄存器。

主设备发出STOP信号产生停止条件。



图 13-3：7 位主发送器的传送图

说明：S=Start(起始条件)， P=Stop(停止条件)， A=响应， NA=非响应

主接收器

发送了地址后，I2C接口从SDA线接收数据字节，并通过内部移位寄存器latch到数据寄存器，产生数据寄存器非空标志RXNE，软件需要读出数据寄存器的值来清除RXNE标志。

如果在接收新数据前数据寄存器的值未被读出，即RXNE仍然为1，则字节等待标志位（WBT）被置1，这时I2C接口保持SCL为低以等待数据寄存器的值被读出。

在每接收一个字节后发出一个应答脉冲并确认数据寄存器里的值已经被取走。

主设备在从从设备接收到最后一个字节后发送一个NACK。收到NACK后，从设备释放SCL线和SDA线的控制。主设备就可以发送一个停止/重新开始条件。

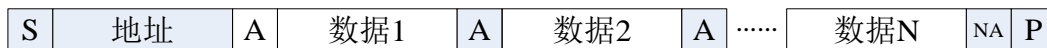


图 13-4：7 位主接收器的传送图

时钟延长

- 发送模式：当数据寄存器里的数据没有被更新时，把 SCL 拉低以等待新的数据写入。
- 接收模式：当数据寄存器里的数据没有被读走时，把 SCL 拉低以等待旧的数据被读走。

13.5 使用流程

13.5.1 I2C 初始化

1. 配置 PCLK0.I2CCEN 为 1，使能 I2C 时钟。
2. 配置 PRESET0.I2CREN 为 1，I2C 复位释放。
3. 将 P10，P04 配置为内部上拉。
4. 配置 I2C_CLK_DIV，设置 I2C 传输速率(从机不需要设置速率)。
5. 配置 I2C_CR0.MEN，使能 I2C。

13.5.2 作为主发送器

1. 写 I2C_CR0 寄存器的 MTX，MEN，MSTA 为 1，发起 START 条件。
2. 把 I2C 要访问的 SLAVE 的 7 位地址写入 I2C_DR 寄存器中(slave 地址+W 标志)。
3. 发送完 SLAVE 地址后等待 ACK 回应，判断有 ACK 回应后开始发送数据。
4. 发送完 SLAVE 地址有 ACK 回应后，开始发送数据，读到 TXE 为 1 时，向 I2C_DR 寄存器中写入要发送的字节，同时硬件会清除 TXE 位，等待 MTF 标志并软件清除 MTF 标志。
5. 重复 4 操作。
6. 最后一个字节发送前，如果 RSTA 为 1，在最后一个字节发送完成后硬件会自动产生 Restart 条件，并可继续发送数据；如果 TACK 为 1，在最后一个字节发送完成后会硬件会自动产生 STOP 条件，软件等待 MBB 为 0 后退出。

注：以上是主发送器标准发送流程，即没有出现拉时钟的情况，若出现拉时钟（WBT==1）软件要立刻处理，否则 SCL 时钟会停止。

13.5.3 作为主接收器

1. 写 I2C_CR0 寄存器的 MTX，MEN，MSTA 为 1，发起 START 条件。
2. 把 I2C 要访问的 SLAVE 的 7 位地址写入 I2C_DR 寄存器中(slave 地址+R 标志)。
3. 发送完 SLAVE 地址后等待 ACK 回应，判断有 ACK 回应后才开始接收数据，若无 ACK 回应，则重新发送 START 条件和 slave 地址，直到有 ACK 回应。
4. 发送完 SLAVE 地址有 ACK 回应后，开始接收数据，读到 RXNE 为 1 时，处理器读取 I2C_DR 寄存器中接收到的字节，同时硬件会清除 RXNE 位，等待 MTF 标志并软件清除 MTF 标志。
5. 重复 4 操作。
6. 最后一个字节发送前，如果 TACK 为 1，在最后一个字节接收完成后会硬件会自动产生 NACK 条件。

注：以上是主接收器标准接收流程，即没有出现拉时钟的情况，若出现拉时钟（WBT==1）软件要立刻处理，否则 SCL 时钟会停止。

13.5.4 作为从发送器

1. 向 I2C_SLAVE_ADDR1 寄存器或 I2C_SLAVE_ADDR2 寄存器写入 7 位地址作为自己在从机状态下被寻址的地址。使能 I2C_CR1.MTX_ANTO_EN，SLAVE 模式下会自动切换 SDA 的传输方向。
2. 等待 MAAS1 或 MASS2（ADDR2_EN=1）标志是否有效。地址匹配无效则重复此步。
3. 等待地址匹配有效，判断 SRW 位是否为 1。为 0 表示从接收，为 1 表示从发送。
4. 写第一个要发送的数据给 I2C_DR。
5. 等待 MBB 标志为 1，则可开始发送数据，读到 TXE 为 1 时，可以向 I2C_DR 寄存器中写入下一个要发送的字节，同时硬件会清除 TXE 位，等待 MTF 标志并软件清除 MTF 标志。
6. 重复 5 操作，当收到主机发来的 NACK 后，I2C 模块释放总线。软件等待 MBB 为 0 后退出。
注意：当 I2C 作为从机发送器时，地址将和数据一起发送。

13.5.5 作为从接收器

1. 向 I2C_SLAVE_ADDR1 寄存器或 I2C_SLAVE_ADDR2 寄存器写入 7 位地址作为自己在从机状态下被寻址的地址。使能 I2C_CR1.MTX_ANTO_EN，SLAVE 模式下会自动切换 SDA 的传输方向。
2. 写 I2C_CR0.TACK 为 0，从机接收完一个字节后自动发送 ACK。
3. 等待 MAAS1 或 MASS2（ADDR2_EN=1）标志是否有效。地址匹配无效则重复此步。
4. 等待地址匹配有效，判断 SRW 位是否为 0。为 0 表示从接收，为 1 表示从发送。
5. 等待 MBB 标志为 1，则可开始接收数据，等到读取 RXNE 为 1 时，处理器读取 I2C_DR 寄存器中接收到的字节，同时硬件会清除 RXNE 位，等待 MTF 标志并软件清除 MTF 标志。
6. 重复 5 操作，当收到主机发来的 STOP 后，I2C 模块释放总线。软件等待 MBB 为 0 后退出。

14 ADC（模/数转换器）

14.1 主要特性

- 12 位分辨率
- 参考电压可选择 V_{DD} 或外接 V_{REF}
- 最多可支持 8 路模拟通道输入，通道 0~6 为外部输入，通道 7 输入来源固定为内部 LDO
- 1Msps 采样速率
- ADC 电压基准可以选择 V_{DDH} 或者外接 V_{REF} 作为 ADC 电源基准源(当配置为 1 时，P2_5 端口将作为外部基准输入)

14.2 使用流程

1. 配置 PCLK0.ADCCEN 为 1,使能 ADC 时钟。
2. 配置 PRESET0.ADCREN 为 1, ADC 复位释放。
3. 配置 ADCGCR1.ADCRST 和 ADCGCR1.ADCPDEN, SAR ADC 释放和 SAR ADC 上电。
4. 配置 ADCVREF.VREFSEL, 选择 ADC 电压基准源。
5. 配置 ADCGCR0.ADCCLKSEL, 选择 ADC 时钟源, 若选择内部时钟分频器产生的时钟, 则配置 ADCCDR0 和 ADCCDR1 设置分频器的分频值。
6. 配置 ADCCONT 设置通道切换时间间隔。
7. 配置 ADCGCR3,配置 EOC 上升沿采样。
8. 配置 ADCGCR0.ADCCUNSET, 选择 ADC 工作模式。
9. 若需要使能中断, 则配置 ADCIER 使能中断。
10. 配置 ADCHL 选择 ADC 输入通道。
11. 配置 ADCGCR0.ADCEN, 使能 ADC。
12. 配置 ADCGCR1.ADCSTEN, 启动 ADC 转换。
13. 等待 ADCDR1.CHDATAV 置 1, 读取 ADCDR1.CHDATAH 和 ADCDR0.CHDATAH 通道数据寄存器。
14. 读取完后将 ADCCSTAT 写 1, 清除接收器 BUF。

使用注意事项:

- 单次采样模式下, ADC 使能后采到的第一个数据为无效数据, 需软件舍弃处理; 具体可参考 SDK 包中 ADC Demo 例程。
- ADC 通道 7 输入来源固定为内部 LDO, 若使用要求精度高时请使用 V_{REF} 外接外部参考电压源。

15 WDT (看门狗定时器)

15.1 概述

看门狗定时器是一个递减计数器，看门狗定时器使用内部 RC32K 为时钟源，如果要使能看门狗，必须先确保 RC32K 已打开。在掉电模式下，看门狗会在定时器溢出时，复位芯片。

读或者写 RSTSTAT 寄存器均自动清除看门狗计数。

15.2 使用流程

WDT 使用流程：

1. 设置寄存器 PCLK0.WDTCEN，开启 WDT 时钟；设置寄存器 PRESET0.WDTREN，WDT 复位释放
2. 设置寄存器 RSTSTAT.WDEN，使能看门狗功能。
3. 设置寄存器 RSTSTAT.WDT[2:0]的值，配置溢出周期最小值。
4. 在溢出周期内，重新写入寄存器 RSTSTAT.WDT[2:0]的值，喂狗，防止溢出。
5. 通过查看寄存器 RSTSTAT.WDOF 的值，判断看门狗是否发生过溢出。

使用注意事项：

- 1、由于芯片个体差异，RCL 时钟有一定的误差，尤其在高温环境下偏差较大，在应用上请尽量将看门狗定时时间比预设目标值大一些，具体需要测试得到；

16 LVD

16.1 概述

LVD 低电压检测功能。LVD 检测到低电压时可以产生复位或中断。LVD 可以对低电压检测的结果进行滤波，增强了系统的稳定性。

16.2 主要特性

- LVD 检测的电压源为 VDDH
- 可设低电压阈值
- 可软件使能低电压复位功能
- 可软件使能低电压中断功能
- 可对低电压检测的结果进行滤波

16.3 使用流程

1. 配置 LVDCON.LVDS[3:0],配置阈值电压。
2. 若需要对低电压检测结果进行滤波，则配置 LVD_LV 寄存器。
3. 配置 LVDCON.LVDEN，使能 LVD 模块。
4. 若需要低电压复位功能，则配置 LVDCON.LVDREN，使能低电压复位功能。
5. 若需要低电压中断功能，则使能 OINTEN.VDDLLOWEN 和 IEN1.EFCINTEN，LVD 中断号选用 EFC 的中断号。

17 中断

17.1 特性

- 11 个中断源 EX0, ET0, ES1, ET1, ES0, EPWM, EADC, EFCINTEN, SPIINTEN, I2CINTEN, LPTIMINTEN
- 4 层中断优先级可配

17.2 中断汇总

中断源	入口地址	允许位	标志位	轮询优先级	中断号(C 语言)
Reset	0000H	-	-	0(最高级)	-
INT0	0003H	EX0+PxIENy	PxIRQy	1	0
Timer0	000BH	ET0	TF0	3	1
UART1	0013H	ES1	RI1+TI1	5	2
Timer1	001BH	ET1	TF1	6	3
UART0	0023H	ES0	RI0+TI0	8	4
PWM	002BH	EPWM+PWMxIE	PWMxIF	10	5
ADC	0033H	EADC+ADCIER	RXAVL	2	6
SPI	003BH	ESPI+SPIIE	SPSR	4	7
EFC	005BH	EFCINTEN+OINTEN	OINTUS	7	11
LPTIMER	0063H	LPTIMINTEN+LPTIE	LPTIF	9	12
I2C	006BH	I2CINTEN +I2CCR	I2CSR	11	13

18 指令集

机器周期为 1 个时钟周期，且多数指令只需一个机器周期执行完成。

18.1 指令操作数说明

Rn	Working register R0-R7
direct	256 internal RAM locations,any Special Function Registers
@Ri	Indirect internal or external RAM location addressed by register R0 or R1
#data	8-bit constant included in instruction
#data 16	16-bit constant included as bytes 2 and 3 of instruction
bit	256 software flags,any bit-addressable I/O pin,control or status bit
A	Accumulator
addr16	Destination address for LCALL and LJMP may be anywhere within the 64K bytes of program memory address space
addr11	Destination address for ACALL and AJMP will be within the same 2K bytes page of program memory as the first byte of the following instruction
rel	SJMP and all conditional jumps include an 8bit offset byte .Range is +127/-128 bytes relative to the first byte of the following instruction

18.2 算术操作指令

Mnemonic	Description	Code	Bytes	Cycles
ADD A,Rn	Add register to accumulator	28-2F	1	1
ADD A,direct	Add direct byte to accumulator	25	2	2
ADD A,@Ri	Add indirect RAM to accumulator	26-27	1	2
ADD A,#data	Add immediate data to accumulator	24	2	2
ADDC A,Rn	Add register to accumulator with carry flag	38-3F	1	1
ADDC A,direct	Add direct byte to A with carry flag	35	2	2
ADDC A,@Ri	Add indirect RAM to A with carry flag	36-37	1	2
ADDC A,#data	Add immediate data to A with carry flag	34	2	2
SUBB A,Rn	Subtract register from A with borrow	98-9F	1	1
SUBB A,direct	Subtract direct byte from A with borrow	95	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	96-97	1	2
SUBB A,#data	Subtract immediate data from A with borrow	94	2	2
INC A	Increment accumulator	04	1	1
INC Rn	Increment register	08-0F	1	2
INC direct	Increment direct byte	05	2	3
INC @Ri	Increment indirect RAM	06-07	1	3
INC DPTR	Increment data pointer	A3	1	1
DEC A	Decrement accumulator	14	1	1
DEC Rn	Decrement register	18-1F	1	2
DEC direct	Decrement direct byte	15	2	3
DEC @Ri	Decrement indirect RAM	16-17	1	3

MUL AB	Multiply A and B	A4	1	5
DIV	Divide A by B	84	1	5
DA A	Decimal adjust accumulator	D4	1	1

18.3 逻辑操作指令

Mnemonic	Description	Code	Bytes	Cycles
ANL A,Rn	AND register to accumulator	58-5F	1	1
ANL A,direct	AND direct byte to accumulator	55	2	2
ANL A,@Ri	AND indirect RAM to accumulator	56-57	1	2
ANL A,#data	AND immediate data to accumulator	54	2	2
ANL direct,A	AND accumulator to direct byte	52	2	3
ANL direct,#data	AND immediate data to direct byte	53	3	4
ORL A,Rn	OR register to accumulator	48-4F	1	1
ORL A,direct	OR direct byte to accumulator	45	2	2
ORL A,@Ri	OR indirect RAM to accumulator	46-47	1	2
ORL A,#data	OR immediate data to accumulator	44	2	2
ORL direct,A	OR accumulator to direct byte	42	2	3
ORL direct,#data	OR immediate data to direct byte	43	3	4
XRL A,Rn	Exclusive OR register to accumulator	68-6F	1	1
XRL A,direct	Exclusive OR direct byte to accumulator	65	2	2
XRL A,@Ri	Exclusive OR indirect RAM to accumulator	66-67	1	2
XRL A,#data	Exclusive OR immediate data to accumulator	64	2	2
XRL direct,A	Exclusive OR accumulator to direct byte	62	2	3
XRL direct,#data	Exclusive OR immediate data to direct byte	63	3	4
CLR A	Clear accumulator	E4	1	1
CPL A	Complement accumulator	F4	1	1
RL A	Rotate accumulator left	23	1	1
RLC A	Rotate accumulator left through carry	33	1	1
RR A	Rotate accumulator right	03	1	1
RRC A	Rotate accumulator right through carry	13	1	1
SWAP A	Swap nibbles within the accumulator	C4	1	1

18.4 数据传送指令

Mnemonic	Description	Code	Bytes	Cycles
MOV A,Rn	Move register to accumulator	E8-EF	1	1
MOV A,direct	Move direct byte to accumulator	E5	2	2
MOV A,@Ri	Move indirect RAM to accumulator	E6-E7	1	2
MOV A,#data	Move immediate data to accumulator	74	2	2
MOV Rn,A	Move accumulator to register	F8-FF	1	2
MOV Rn,direct	Move direct byte to register	A8-AF	2	4

Mnemonic	Description	Code	Bytes	Cycles
MOV Rn,#data	Move immediate data to register	78-7F	2	2
MOV direct,A	Move accumulator to direct byte	F5	2	3
MOV direct,Rn	Move register to direct byte	88-8F	2	3
MOV direct1,direct2	Move direct byte to direct byte	85	3	4
MOV direct,@Ri	Move indirect RAM to direct byte	86-87	2	4
MOV direct,#data	Move immediate data to direct byte	75	3	3
MOV @Ri,A	Move accumulator to indirect RAM	F6-F7	1	3
MOV @Ri,direct	Move direct byte to indirect RAM	A6-A7	2	5
MOV @Ri,#data	Move immediate data to indirect RAM	76-77	2	3
MOV DPTR,#data16	Load data pointer with a 16-bit constant	90	3	3
MOVC A,@A+DPTR	Move code byte relative to DPTR to accumulator	93	1	3
MOVC A,@A+PC	Move code byte relative to PC to accumulator	83	1	3
MOVX A,@Ri	Move external RAM(8-bit addr.)to A	E2-E3	1	3-10
MOVX A,@DPTR	Move external RAM(16-bit addr.)to A	E0	1	3-10
MOVX @Ri,A	Move A to extern RAM(8-bit addr.)	F2-F3	1	4-11
MOVX @DPTR,A	Move A to extern RAM(16-bit addr.)	F0	1	4-11
PUSH direct	Push direct byte onto stack	C0	2	4
POP direct	Pop direct byte from stack	D0	2	3
XCH A,Rn	Exchange register with accumulator	C8-CF	1	2
XCH A,direct	Exchange direct byte with accumulator	C5	2	3
XCH A,@Ri	Exchange indirect RAM with accumulator	C6-C7	1	3
XCHD A,@Ri	Exchange low-order nibble indir.RAM with A	D6-D7	1	3

18.5 控制程序转移指令

Mnemonic	Description	Code	Bytes	Cycles
ACALL addr11	Absolute subroutine call	xxx11	2	6
LCALL addr16	Long subroutine call	12	3	6
RET	from subroutine	22	1	4
RETI	from interrupt	32	1	4
AJMP addr11	Absolute jump	xxx01	2	3
LJMP addr16	Long jump	02	3	4
SJMP rel	Short jump (relative addr.)	80	2	3
JMP @A+DPTR	Jump indirect relative to the DPTR	73	1	2
JZ rel	Jump if accumulator is zero	60	2	3
JNZ rel	Jump if accumulator is not zero	70	2	3
JC rel	Jump if carry flag is set	40	2	3
JNC	Jump if carry flag is not set	50	2	3
JB bit,rel	Jump if direct bit is set	20	3	4

Mnemonic	Description	Code	Bytes	Cycles
JNB bit,rel	Jump if direct bit is not set	30	3	4
JBC bit,direct rel	Jump if direct bit is set and clear bit	10	3	4
CJNE A,direct rel	Compare direct byte to A and jump if not equal	B5	3	4
CJNE A,#data rel	Compare immediate to A and jump if not equal	B4	3	4
CJNE Rn,#data rel	Compare immed. to reg. and jump if not equal	B8-BF	3	4
CJNE @Ri,#data rel	Compare immed. to ind. and jump if not equal	B6-B7	3	4
DJNE Rn,rel	Decrement register and jump if not zero	D8-DF	2	3
DJNZ direct,rel	Decrement direct byte and jump if not zero	D5	3	4
NOP	No operation	00	1	1

18.6 位操作指令

Mnemonic	Description	Code	Bytes	Cycles
CLR C	Clear carry flag	C3	1	1
CLR bit	Clear direct bit	C2	2	3
SETB C	Set carry flag	D3	1	1
SETB bit	Set direct bit	D2	2	3
CPL C	Complement carry flag	B3	1	1
CPL bit	Complement direct bit	B2	2	3
ANL C,bit	AND direct bit to carry flag	82	2	2
ANL C,/bit	AND complement of direct bit to carry	B0	2	2
ORL C,bit	OR direct bit to carry flag	72	2	2
ORL C,/bit	OR complement of direct bit to carry	A0	2	2
MOV C,bit	Move direct bit to carry flag	A2	2	2
MOV bit,C	Move carry flag to direct bit	92	2	3

19 版本维护

版本	日期	描述
V1.0	2020.12.1	初始版
V1.1	2021.2.23	增加 QFN20 封装, 更新 TSSOP20 封装信息; 更新电气参数; 更新芯片工作电压参数、ADC 通道数、LVD/LVR 消耗电流参数; 更新时钟结构图, LPTIM 时钟源描述; 修改 WDT 使用流程;
V1.2	2021.6.8	修改 ADC 寄存器 ADCIER、ADCGCR0、ADCDR1 描述; 修改 I2C 使用流程; 修改 LPTIMER 使用流程; 修改 ADC 描述及使用流程; 修改 GPIO 端口中断描述; 修改 IO_IE、IO_SR 功能描述; 增加 IO_IE、IO_SR、SPI、LPTIMER、I2C 寄存器访问方式描述; 增加 LVD 中断使用流程描述; 增加 PCLK1 备注描述; 将 RCH 时钟源描述(包括相关寄存器描述)统一改成 48MHz; 更新 QFN20 封装管脚分布图
V1.3	2021.6.15	修改 S0CON、S1CON 寄存器描述; 修改 ADCGCR1 寄存器描述; 更新时钟结构图; 更新 UART 方式 2 发送和接收时序图; 删除电气参数中外部 32.768K 晶振的相关描述;
V1.4	2023.7.13	增加 UART 波特率 4800 和 2400 误差范围; 增加 WDT 备注描述; 删除 flash 中关于 nvr 和 boot 描述; 修改 I2C 章节图编号排序; 修改 gpio 引脚复用中关于 SPI 和 ADC 复用描述; 修改 PCLK1 寄存器描述; 修改 PRESET1 寄存器描述; 修改 ADCCDR0 寄存器的 CLKDIV0 及 ADCCDR1 寄存器的 CLKDIV1 说明
V1.5	2024.05.08	删除“引脚描述”、“电气参数”及“封装信息”章节。 删除 SOP8 相关描述。