

# UM321x 用户手册

版本：V1.3



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    | 硬件功能 .....                                   | 1  |
| 1.2    | 功能框图 .....                                   | 3  |
| 1.3    | 电源框图 .....                                   | 4  |
| 2      | 处理器 .....                                    | 5  |
| 2.1    | 概述 .....                                     | 5  |
| 2.2    | 主要特性 .....                                   | 5  |
| 2.3    | 功能框图 .....                                   | 6  |
| 2.4    | 内核寄存器组 .....                                 | 6  |
| 3      | 系统配置(SCU) .....                              | 7  |
| 3.1    | 地址映射 .....                                   | 7  |
| 3.2    | 时钟框图 .....                                   | 8  |
| 3.3    | 时钟选择 .....                                   | 8  |
| 3.4    | 复位源 .....                                    | 9  |
| 3.4.1  | 内部 POR 上电复位 .....                            | 9  |
| 3.4.2  | RESETEN 复位 .....                             | 9  |
| 3.4.3  | WDT 复位 .....                                 | 9  |
| 3.4.4  | SOFT_RESETN 复位 .....                         | 9  |
| 3.4.5  | 模块软件复位 .....                                 | 10 |
| 3.4.6  | LVD 复位 .....                                 | 10 |
| 3.4.7  | LOCKUP 复位 .....                              | 10 |
| 3.5    | 低功耗模式 .....                                  | 10 |
| 3.5.1  | Sleep 模式 .....                               | 11 |
| 3.5.2  | Deepsleep 模式 .....                           | 11 |
| 3.5.3  | Stop 模式 .....                                | 12 |
| 3.6    | 系统寄存器 .....                                  | 13 |
| 3.6.1  | 系统控制寄存器 0 SYSCTRL0 (偏移: 000h) .....          | 13 |
| 3.6.2  | 系统控制寄存器 1 SYSCTRL1 (偏移: 004h) .....          | 15 |
| 3.6.3  | 系统控制保护寄存器 SYSCTRL_PROTECT (偏移: 008h) .....   | 16 |
| 3.6.4  | 时钟控制寄存器 OSC_CTRL (偏移: 0x00Ch) .....          | 16 |
| 3.6.5  | 外围模块时钟寄存器 PERI_CLKEN (偏移: 010h) .....        | 17 |
| 3.6.6  | 复位标识寄存器 RESET_FLAG (偏移: 020h) .....          | 18 |
| 3.6.7  | 外围模块复位控制寄存器 PERI_RESET (偏移: 024h) .....      | 19 |
| 3.6.8  | 外部复位滤波控制寄存器 EXT_RESET_CTRL (偏移: 028h) .....  | 20 |
| 3.6.9  | 端口 PA 功能配置寄存器 PA_SEL (偏移: 030h) .....        | 20 |
| 3.6.10 | 端口 PB 功能配置寄存器 PB_SEL (偏移: 034h) .....        | 21 |
| 3.6.11 | 端口 PC 功能配置寄存器 PC_SEL (偏移: 038h) .....        | 22 |
| 3.6.12 | 端口 PD 功能配置寄存器 PD_SEL (偏移: 03Ch) .....        | 23 |
| 3.6.13 | 端口数模配置寄存器 PAD_ADS (偏移: 040h) .....           | 24 |
| 3.6.14 | 端口驱动能力配置寄存器 PAD_DR (偏移: 044h) .....          | 26 |
| 3.6.15 | 端口上拉配置寄存器 PAD_PU (偏移: 048h) .....            | 27 |
| 3.6.16 | 端口下拉配置寄存器 PAD_PD (偏移: 04Ch) .....            | 29 |
| 3.6.17 | 端口开漏输出配置寄存器 PAD_OD (偏移: 050h) .....          | 30 |
| 3.6.18 | 端口输入类型配置寄存器 PAD_CS (偏移: 054h) .....          | 32 |
| 3.6.19 | 端口输入配置寄存器 PAD_IE (偏移: 058h) .....            | 33 |
| 3.6.20 | IO 控制保护寄存器 IOCTRL_PROTECT (偏移: 05Ch) .....   | 35 |
| 3.6.21 | 唤醒状态清除寄存器 IOWACK_INTR_CLR (偏移: 0x060h) ..... | 35 |

|        |  |    |
|--------|--|----|
| 3.6.22 | LVD 控制寄存器 LVD_CFG (偏移: 064h).....          | 36 |
| 3.6.23 | 端口辅助功能寄存器 GPIO_CFG (偏移: 070h).....         | 37 |
| 3.6.24 | 外部复位端口选择寄存器 EXTRST_SEL (偏移: 074h).....     | 37 |
| 3.6.25 | 停止模式选择寄存器 STOPMODE_SEL (偏移: 078h).....     | 37 |
| 3.6.26 | REMAP 寄存器 REMAP_ADDR (偏移: 07Ch).....       | 37 |
| 3.6.27 | 中断向量地址重映射寄存器 VECTOR_OFFSET (偏移: 080h)..... | 38 |
| 3.6.28 | 随机数控制寄存器 HRNG_CR (偏移: 084h).....           | 38 |
| 3.6.29 | 随机数种子寄存器 HRNG_SEED (偏移: 088h).....         | 38 |
| 3.6.30 | 随机数数据寄存器 HRNG_DATA (偏移: 08Ch).....         | 38 |
| 3.6.31 | 蜂鸣器控制寄存器 BUZZERCR (偏移: 090h).....          | 39 |
| 3.6.32 | 保留寄存器 0 Reserved0 (偏移: 0A0h) (保留).....     | 39 |
| 3.6.33 | 保留寄存器 1 Reserved1 (偏移: 0A4h) (保留).....     | 39 |
| 4      | NVIC.....                                  | 40 |
| 4.1    | 概述.....                                    | 40 |
| 4.2    | 主要特性.....                                  | 40 |
| 4.3    | 中断源.....                                   | 40 |
| 5      | UART0.....                                 | 42 |
| 5.1    | 概述.....                                    | 42 |
| 5.2    | 主要特性.....                                  | 42 |
| 5.3    | 寄存器描述.....                                 | 42 |
| 5.3.1  | 中断状态寄存器 UARTISR (偏移: 00h).....             | 43 |
| 5.3.2  | 中断使能寄存器 UARTIER (偏移: 04h).....             | 43 |
| 5.3.3  | 控制寄存器 UARTCR (偏移: 08h).....                | 44 |
| 5.3.4  | 发送数据寄存器 UARTTDR (偏移: 0Ch).....             | 44 |
| 5.3.5  | 接收数据寄存器 UARTRDR (偏移: 0Ch).....             | 44 |
| 5.3.6  | 波特率参数低位寄存器 UARTBRPL (偏移: 10h).....         | 44 |
| 5.3.7  | 波特率参数高位寄存器 UARTBRPH (偏移: 14h).....         | 45 |
| 5.4    | 使用流程.....                                  | 45 |
| 5.4.1  | 串口的发送和接收.....                              | 45 |
| 5.4.2  | 串口初始化.....                                 | 46 |
| 5.4.3  | 串口发送字节.....                                | 46 |
| 5.4.4  | 串口接收字节.....                                | 46 |
| 6      | UART1.....                                 | 47 |
| 6.1    | 概述.....                                    | 47 |
| 6.2    | 主要特性.....                                  | 47 |
| 6.3    | 寄存器描述.....                                 | 47 |
| 6.3.1  | 接收缓冲寄存器 UART1_RBR (偏移: 00h).....           | 48 |
| 6.3.2  | 发送缓冲寄存器 UART1_THR (偏移: 00h).....           | 48 |
| 6.3.3  | 波特率分频低位寄存器 UART1_DLL (偏移: 00h).....        | 48 |
| 6.3.4  | 波特率分频高位寄存器 UART1_DLH (偏移: 04h).....        | 48 |
| 6.3.5  | 中断使能寄存器 UART1_IER (偏移: 04h).....           | 48 |
| 6.3.6  | 中断状态寄存器 UART1_IIR (偏移: 08h).....           | 49 |
| 6.3.7  | FIFO 控制寄存器 UART1_FCR (偏移: 08h).....        | 50 |
| 6.3.8  | LINE 控制寄存器 UART1_LCR (偏移: 0Ch).....        | 50 |
| 6.3.9  | 流控制寄存器 UART1_MCR (偏移: 10h).....            | 51 |
| 6.3.10 | LINE 中断状态寄存器 UART1_LSR (偏移: 14h).....      | 51 |
| 6.3.11 | 流状态寄存器 UART1_MSR (偏移: 18h).....            | 52 |
| 6.3.12 | 状态寄存器 UART1_USR (偏移: 7Ch).....             | 53 |
| 6.3.13 | 发送 FIFO 数据个数寄存器 UART1_TFL (偏移: 80h).....   | 53 |
| 6.3.14 | 接收 FIFO 数据个数寄存器 UART1_RFL (偏移: 84h).....   | 53 |

|        |   |    |
|--------|---|----|
| 6.3.15 | 小数分频寄存器 UART1_DLF(偏移: C0h)                  | 53 |
| 6.3.16 | 接收地址匹配寄存器 UART1_RAR(偏移: C4h)                | 53 |
| 6.3.17 | 发送地址匹配寄存器 UART1_TAR (偏移: C8h)               | 54 |
| 6.4    | 使用流程  | 54 |
| 6.4.1  | UART1 发送流程                                  | 54 |
| 6.4.2  | UART1 接收流程                                  | 54 |
| 6.4.3  | CTS 和 RTS 控制流功能设置流程                         | 55 |
| 6.4.4  | UART1 DMA 传输配置流程                            | 55 |
| 7      | LPUART                                      | 57 |
| 7.1    | 概述  | 57 |
| 7.2    | 主要特性  | 57 |
| 7.3    | 寄存器描述                                       | 57 |
| 7.3.1  | 接收数据寄存器 LPURXD (偏移: 00h)                    | 58 |
| 7.3.2  | 发送数据寄存器 LPUTXD (偏移: 04h)                    | 58 |
| 7.3.3  | 状态寄存器 LPUSTA (偏移: 08h)                      | 58 |
| 7.3.4  | 控制寄存器 LPUCON (偏移: 0Ch)                      | 58 |
| 7.3.5  | 中断标志寄存器 LPUIF (偏移: 10h)                     | 59 |
| 7.3.6  | 波特率寄存器 LPUBAUD (偏移: 14h)                    | 60 |
| 7.3.7  | 接收使能寄存器 LPUEN (偏移: 18h)                     | 60 |
| 7.3.8  | 数据匹配寄存器 COMPARE (偏移: 1Ch)                   | 60 |
| 7.3.9  | 波特率调制控制寄存器 MODU (偏移: 20h)                   | 61 |
| 7.4    | 软件流程  | 61 |
| 7.4.1  | 数据接收  | 61 |
| 7.4.2  | 数据发送  | 61 |
| 7.4.3  | 调制控制寄存器配置建议                                 | 61 |
| 7.4.4  | 休眠模式下的数据接收唤醒                                | 62 |
| 8      | I2C   | 63 |
| 8.1    | 概述  | 63 |
| 8.2    | 主要特征  | 63 |
| 8.3    | 寄存器描述                                       | 63 |
| 8.3.1  | I2C 配置寄存器 I2C_CR (偏移: 00h)                  | 64 |
| 8.3.2  | I2C 配置清除寄存器 I2C_CLR (偏移: 04h)               | 65 |
| 8.3.3  | I2C 状态寄存器 I2C_STAT(偏移: 08h)                 | 65 |
| 8.3.4  | I2C 数据寄存器 I2C_DATA(偏移: 0Ch)                 | 66 |
| 8.3.5  | I2C 波特率配置寄存器 I2C_CCR(偏移: 10h)               | 66 |
| 8.3.6  | I2C SLAVE 地址寄存器 0 I2C_SAD0 (偏移: 14h)        | 67 |
| 8.3.7  | I2C SLAVE 地址屏蔽寄存器 0 I2C_SADM0 (偏移: 18h)     | 67 |
| 8.3.8  | 10 比特 I2C SLAVE 地址寄存器 I2C_XSAD (偏移: 1Ch)    | 67 |
| 8.3.9  | 10 比特 I2C SLAVE 地址屏蔽寄存器 I2C_XSADM (偏移: 20h) | 67 |
| 8.3.10 | I2C 复位寄存器 I2C_SRST (偏移: 24h)                | 68 |
| 8.3.11 | I2C SLAVE 地址寄存器 1 I2C_SAD1 (偏移: 28h)        | 68 |
| 8.3.12 | I2C SLAVE 地址屏蔽寄存器 1 I2C_SADM1 (偏移: 2Ch)     | 68 |
| 8.3.13 | I2C SLAVE 地址寄存器 2 I2C_SAD2 (偏移: 30h)        | 68 |
| 8.3.14 | I2C SLAVE 地址屏蔽寄存器 2 I2C_SADM2 (偏移: 34h)     | 68 |
| 8.3.15 | I2C SLAVE 地址寄存器 2 I2C_SAD3 (偏移: 38h)        | 69 |
| 8.3.16 | I2C SLAVE 地址屏蔽寄存器 3 I2C_SADM3 (偏移: 3Ch)     | 69 |
| 8.4    | 协议描述  | 69 |
| 8.4.1  | I2C 通信协议 (7 位寻址)                            | 70 |
| 8.4.2  | I2C 通信协议 (10 位寻址)                           | 71 |
| 8.5    | 使用流程  | 72 |

|        |   |    |
|--------|---|----|
| 8.5.1  | 初始化程序 .....                                     | 72 |
| 8.5.2  | 主机发送功能 .....                                    | 72 |
| 8.5.3  | 主机接收功能 .....                                    | 72 |
| 8.5.4  | 从机接收功能 .....                                    | 73 |
| 8.5.5  | 从机发送功能 .....                                    | 74 |
| 9      | SPI0 .....                                      | 76 |
| 9.1    | 概述 .....  | 76 |
| 9.2    | 主要特性 .....                                      | 76 |
| 9.3    | 寄存器描述 .....                                     | 76 |
| 9.3.1  | SPI0 配置寄存器 SPI0CR (偏移: 00h) .....               | 77 |
| 9.3.2  | SPI0 主模式控制寄存器 0 SPI0CS0 (偏移: 04h) .....         | 78 |
| 9.3.3  | SPI0 主模式控制寄存器 1 SPI0CS1 (偏移: 08h) .....         | 78 |
| 9.3.4  | SPI0 过程控制寄存器 SPI0OPCR (偏移: 14h) .....           | 79 |
| 9.3.5  | SPI0 中断控制寄存器 SPI0IE (偏移: 18h) .....             | 79 |
| 9.3.6  | SPI0 中断标志寄存器 SPI0IF (偏移: 1Ch) .....             | 80 |
| 9.3.7  | SPI0 发送缓存寄存器 SPI0TXBUF (偏移: 20h) .....          | 80 |
| 9.3.8  | SPI0 接收缓存寄存器 SPI0RXBUF (偏移: 24h) .....          | 81 |
| 9.3.9  | SPI0 DMA 接收设置寄存器 DMA_SPI0RX_LEV (偏移: 28h) ..... | 81 |
| 9.3.10 | SPI0 DMA 发送设置寄存器 DMA_SPI0TX_LEV (偏移: 2Ch) ..... | 81 |
| 9.4    | 接口时序 .....                                      | 81 |
| 9.4.1  | CPHA=0 .....                                    | 81 |
| 9.4.2  | CPHA=1 .....                                    | 82 |
| 9.4.3  | 从器件 SSN .....                                   | 82 |
| 9.5    | 使用流程 .....                                      | 83 |
| 9.5.1  | 初始化程序 .....                                     | 83 |
| 9.5.2  | 发送流程 .....                                      | 84 |
| 9.5.3  | 接收流程 .....                                      | 84 |
| 9.5.4  | SPI0 DMA 发送流程 .....                             | 84 |
| 9.5.5  | SPI0 DMA 接收流程 .....                             | 85 |
| 10     | SPI1 .....                                      | 86 |
| 10.1   | 概述 .....  | 86 |
| 10.2   | 主要特性 .....                                      | 86 |
| 10.3   | 寄存器描述 .....                                     | 86 |
| 10.3.1 | SPI1 配置寄存器 SPI1CR (偏移: 00h) .....               | 87 |
| 10.3.2 | SPI1 主模式控制寄存器 0 SPI1CS0 (偏移: 04h) .....         | 88 |
| 10.3.3 | SPI1 过程控制寄存器 SPI1OPCR (偏移: 14h) .....           | 88 |
| 10.3.4 | SPI1 中断控制寄存器 SPI1IE (偏移: 18h) .....             | 89 |
| 10.3.5 | SPI1 中断标志寄存器 SPI1IF (偏移: 1Ch) .....             | 89 |
| 10.3.6 | SPI1 发送缓存寄存器 SPI1TXBUF (偏移: 20h) .....          | 90 |
| 10.3.7 | SPI1 接收缓存寄存器 SPI1RXBUF (偏移: 24h) .....          | 90 |
| 10.3.8 | SPI1 DMA 接收设置寄存器 DMA_SPI1RX_LEV (偏移: 28h) ..... | 90 |
| 10.3.9 | SPI1 DMA 发送设置寄存器 DMA_SPI1TX_LEV (偏移: 2Ch) ..... | 90 |
| 10.4   | 接口时序 .....                                      | 90 |
| 10.4.1 | CPHA=0 .....                                    | 91 |
| 10.4.2 | CPHA=1 .....                                    | 91 |
| 10.4.3 | 从器件 SSN .....                                   | 92 |
| 10.5   | 使用流程 .....                                      | 92 |
| 10.5.1 | 初始化程序 .....                                     | 92 |
| 10.5.2 | 发送流程 .....                                      | 93 |
| 10.5.3 | 接收流程 .....                                      | 93 |
| 10.5.4 | SPI1 DMA 发送流程 .....                             | 93 |

|         |   |     |
|---------|---|-----|
| 10.5.5  | SPI1 DMA 接收流程 .....                               | 94  |
| 11      | GTIMER .....                                      | 95  |
| 11.1    | 概述 .....  | 95  |
| 11.2    | 主要特性 .....  | 95  |
| 11.3    | 寄存器描述 .....                                       | 95  |
| 11.3.1  | Timer0 加载寄存器 T0_ARR (偏移: 00h) .....               | 96  |
| 11.3.2  | Timer0 计数寄存器 T0_CNT (偏移: 04h) .....               | 96  |
| 11.3.3  | Timer0 控制寄存器 T0_CR (偏移: 08h) .....                | 97  |
| 11.3.4  | Timer0 中断状态寄存器 T0_IF (偏移: 0Ch) .....              | 97  |
| 11.3.5  | Timer0 中断清除寄存器 T0_CIF (偏移: 10h) .....             | 98  |
| 11.3.6  | Timer1 加载寄存器 T1_ARR (偏移: 14h) .....               | 98  |
| 11.3.7  | Timer1 计数寄存器 T1_CNT (偏移: 18h) .....               | 98  |
| 11.3.8  | Timer1 控制寄存器 T1_CR (偏移: 1Ch) .....                | 98  |
| 11.3.9  | Timer1 中断状态寄存器 T1_IF (偏移: 20h) .....              | 99  |
| 11.3.10 | Timer1 中断清除寄存器 T1_CIF (偏移: 24h) .....             | 99  |
| 11.3.11 | Timer2 加载寄存器 T2_ARR (偏移: 28h) .....               | 99  |
| 11.3.12 | Timer2 计数寄存器 T2_CNT (偏移: 2Ch) .....               | 100 |
| 11.3.13 | Timer2 控制寄存器 T2_CR (偏移: 30h) .....                | 100 |
| 11.3.14 | Timer2 中断状态寄存器 T2_IF (偏移: 34h) .....              | 100 |
| 11.3.15 | Timer2 中断清除寄存器 T2_CIF (偏移: 38h) .....             | 101 |
| 11.3.16 | Timer3 加载寄存器 T3_ARR (偏移: 3Ch) .....               | 101 |
| 11.3.17 | Timer3 计数寄存器 T3_CNT (偏移: 40h) .....               | 101 |
| 11.3.18 | Timer3 控制寄存器 T3_CR (偏移: 44h) .....                | 101 |
| 11.3.19 | Timer3 中断状态寄存器 T3_IF (偏移: 48h) .....              | 102 |
| 11.3.20 | Timer3 中断清除寄存器 T3_CIF (偏移: 4Ch) .....             | 102 |
| 11.3.21 | 定时器预分频寄存器 TIM_PSC (偏移: 50h) .....                 | 102 |
| 11.3.22 | 工作模式控制寄存器 ICMODE (偏移: 54h) .....                  | 104 |
| 11.3.23 | 输入捕获模式控制寄存器 CCR (偏移: 58h) .....                   | 104 |
| 11.3.24 | 输入捕获模式中断状态寄存器 CCIF (偏移: 5Ch) .....                | 104 |
| 11.3.25 | Channel0 输入捕获模式 counter 寄存器 C0_CR (偏移: 60h) ..... | 105 |
| 11.3.26 | Channel2 输入捕获模式 counter 寄存器 C2_CR (偏移: 64h) ..... | 105 |
| 11.3.27 | PWM 控制寄存器 PCR (偏移: 68h) .....                     | 105 |
| 11.3.28 | PWM 中断状态寄存器 CPIF (偏移: 6Ch) .....                  | 106 |
| 11.3.29 | PWM0 比较寄存器 C0_PR (偏移: 70h) .....                  | 106 |
| 11.3.30 | PWM1 比较寄存器 C1_PR (偏移: 74h) .....                  | 106 |
| 11.3.31 | PWM2 比较寄存器 C2_PR (偏移: 78h) .....                  | 106 |
| 11.3.32 | PWM3 比较寄存器 C3_PR (偏移: 7Ch) .....                  | 107 |
| 11.4    | 使用说明 .....  | 107 |
| 11.4.1  | 计数方向 .....  | 107 |
| 11.4.2  | 计数模式 .....  | 108 |
| 11.4.3  | 输入捕获模式 .....                                      | 108 |
| 11.4.4  | PWM 模式 .....                                      | 108 |
| 11.4.5  | PWM 边沿对齐模式 .....                                  | 108 |
| 11.4.6  | PWM 中心对齐模式 .....                                  | 109 |
| 11.4.7  | 输入捕获模式 .....                                      | 110 |
| 11.5    | 使用流程 .....  | 110 |
| 11.5.1  | 普通定时器 .....                                       | 111 |
| 11.5.2  | PWM 输出 .....                                      | 111 |
| 11.5.3  | 输入捕获 .....  | 111 |
| 12      | LPTIMER .....                                     | 113 |

|         |  |     |
|---------|--|-----|
| 12.1    | 概述.....                                  | 113 |
| 12.2    | 主要特性.....                                | 113 |
| 12.3    | 寄存器描述.....                               | 113 |
| 12.3.1  | 配置寄存器 LPTCFG (偏移: 00h) .....             | 114 |
| 12.3.2  | 计数值寄存器 LPTCNT (偏移: 04h) .....            | 115 |
| 12.3.3  | 比较寄存器 LPTCMP (偏移: 08h) .....             | 115 |
| 12.3.4  | 目标寄存器 LPTTARGET (偏移: 0Ch) .....          | 115 |
| 12.3.5  | 中断使能寄存器 LPTIE (偏移: 10h) .....            | 115 |
| 12.3.6  | 中断标志寄存器 LPTIF (偏移: 14h) .....            | 116 |
| 12.3.7  | 控制寄存器 LPTCTRL (偏移: 18h) .....            | 116 |
| 12.4    | 使用流程.....                                | 116 |
| 12.4.1  | 普通定时器.....                               | 116 |
| 12.4.2  | PWM 输出.....                              | 116 |
| 12.4.3  | Trigger 脉冲触发计数模式.....                    | 117 |
| 12.4.4  | 外部异步脉冲计数模式.....                          | 117 |
| 12.4.5  | Timeout 模式.....                          | 117 |
| 13      | ADC.....                                 | 119 |
| 13.1    | 概述.....                                  | 119 |
| 13.2    | 主要特性.....                                | 119 |
| 13.3    | 寄存器描述.....                               | 119 |
| 13.3.1  | ADC 通用控制寄存器 ADC_GCR (偏移: 000h) .....     | 120 |
| 13.3.2  | A/D 通道 0 数据寄存器 ADC0_DR (偏移: 004h) .....  | 121 |
| 13.3.3  | A/D 通道 1 数据寄存器 ADC1_DR (偏移: 008h) .....  | 121 |
| 13.3.4  | A/D 通道 2 数据寄存器 ADC2_DR (偏移: 00Ch) .....  | 122 |
| 13.3.5  | A/D 通道 3 数据寄存器 ADC3_DR (偏移: 010h) .....  | 122 |
| 13.3.6  | ADC 时钟分频寄存器 ADC_CDR (偏移: 024h) .....     | 122 |
| 13.3.7  | ADC 中断状态寄存器 ADC_ISR (偏移: 028h) .....     | 123 |
| 13.3.8  | ADC 中断使能寄存器 ADC_IER (偏移: 02Ch) .....     | 123 |
| 13.3.9  | ADC 中断清除寄存器 ADC_ICR (偏移: 030h) .....     | 124 |
| 13.3.10 | ADC 切换间隔计数寄存器(ADC_COUNT (偏移: 034h) ..... | 125 |
| 13.3.11 | ADC 接收数据寄存器 ADC_RXREG (偏移: 038h) .....   | 125 |
| 13.3.12 | ADC 状态寄存器 ADC_CSTAT (偏移: 03Ch) .....     | 125 |
| 13.3.13 | ADC 采样脉宽寄存器 ADC_SPW (偏移: 040h) .....     | 125 |
| 13.4    | 使用流程.....                                | 126 |
| 13.4.1  | 单次扫描模式单通道 A/D 转换.....                    | 126 |
| 13.4.2  | 单次扫描模式多通道 A/D 转换.....                    | 126 |
| 13.4.3  | 连续扫描模式多通道 A/D 转换.....                    | 127 |
| 13.4.4  | 注意事项.....                                | 127 |
| 14      | GPIO.....                                | 128 |
| 14.1    | 概述.....                                  | 128 |
| 14.2    | 主要特性.....                                | 128 |
| 14.3    | 寄存器描述.....                               | 128 |
| 14.3.1  | 数据方向寄存器 GPIO_DIR(偏移: 00h) .....          | 129 |
| 14.3.2  | 输出置位寄存器 GPIO_SET(偏移: 08h) .....          | 129 |
| 14.3.3  | 输出清零寄存器 GPIO_CLR(偏移: 0Ch).....           | 129 |
| 14.3.4  | GPIO 输出引脚映射寄存器 GPIO_ODATA(偏移: 10h) ..... | 129 |
| 14.3.5  | GPIO 输入引脚映射寄存器 GPIO_IDATA(偏移: 14h) ..... | 129 |
| 14.3.6  | GPIO 中断使能寄存器 GPIO_IEN(偏移: 18h).....      | 130 |
| 14.3.7  | GPIO 中断触发模式寄存器 GPIO_IS(偏移: 1Ch) .....    | 130 |
| 14.3.8  | GPIO 中断边沿触发设置寄存器 GPIO_IBE(偏移: 20h) ..... | 130 |



|         |                                       |     |
|---------|---------------------------------------|-----|
| 14.3.9  | GPIO 中断高低电平触发设置寄存器 GPIO_I EV(偏移: 24h) | 130 |
| 14.3.10 | GPIO 中断状态清除寄存器 GPIO_IC(偏移: 28h)       | 131 |
| 14.3.11 | GPIO 原始中断状态寄存器 GPIO_RIS(偏移: 2Ch)      | 131 |
| 14.3.12 | GPIO 屏蔽后中断状态寄存器 GPIO_MIS(偏移: 30h)     | 131 |
| 14.4    | 使用流程                                  | 131 |
| 14.4.1  | 输入输出 IO                               | 131 |
| 14.4.2  | 中断触发模式                                | 131 |
| 14.4.3  | 清除中断                                  | 132 |
| 15      | CRC16                                 | 133 |
| 15.1    | 概述                                    | 133 |
| 15.2    | 寄存器描述                                 | 133 |
| 15.2.1  | 数据寄存器 CRC16_DATA (偏移: 00H)            | 133 |
| 15.2.2  | 初始值寄存器 CRC16_INIT (偏移: 04H)           | 133 |
| 15.2.3  | 控制寄存器 CRC16_CTRL (偏移: 08H)            | 134 |
| 15.3    | 使用流程                                  | 134 |
| 16      | WDT                                   | 135 |
| 16.1    | 概述                                    | 135 |
| 16.2    | 主要特性                                  | 135 |
| 16.3    | 寄存器描述                                 | 135 |
| 16.3.1  | 装载寄存器 WDT_LOAD(偏移: 00h)               | 135 |
| 16.3.2  | 计数寄存器 WDT_CNT(偏移: 04h)                | 136 |
| 16.3.3  | 控制寄存器 WDT_CTRL(偏移: 08h)               | 136 |
| 16.3.4  | 清除寄存器 WDT_CLR(偏移: 0Ch)                | 136 |
| 16.3.5  | RAW 中断状态寄存器 WDT_INTRAW(偏移: 10h)       | 136 |
| 16.3.6  | MASK 中断状态寄存器 WDT_MINTS(偏移: 14h)       | 136 |
| 16.3.7  | STALL 控制寄存器 WDT_STALL(偏移: 18h)        | 137 |
| 16.3.8  | LOCK 寄存器 WDT_LOCK(偏移: 1Ch)            | 137 |
| 16.4    | 使用流程                                  | 137 |
| 17      | RTC                                   | 139 |
| 17.1    | 概述                                    | 139 |
| 17.2    | 主要特性                                  | 139 |
| 17.3    | 寄存器描述                                 | 139 |
| 17.3.1  | RTC 数据寄存器 RTCDR (偏移: 00h)             | 139 |
| 17.3.2  | RTC 匹配寄存器 RTCMR (偏移: 04h)             | 140 |
| 17.3.3  | RTC 加载寄存器 RTCLR (偏移: 08h)             | 140 |
| 17.3.4  | RTC 控制寄存器 RTCCR (偏移: 0Ch)             | 140 |
| 17.3.5  | RTC 中断使能寄存器 RTCIMSC (偏移: 10h)         | 140 |
| 17.3.6  | RTC 原始中断状态寄存器 RTCRIS (偏移: 14h)        | 140 |
| 17.3.7  | RTC 屏蔽中断状态寄存器 RTCMIS (偏移: 18h)        | 141 |
| 17.3.8  | RTC 中断清除寄存器 RTCICR (偏移: 1Ch)          | 141 |
| 17.4    | 使用流程                                  | 141 |
| 17.4.1  | RTC 配置流程                              | 141 |
| 17.4.2  | RTC 设置闹钟                              | 142 |
| 18      | DMA                                   | 143 |
| 18.1    | 概述                                    | 143 |
| 18.2    | 主要特性                                  | 143 |
| 18.3    | 硬件握手信号                                | 143 |
| 18.4    | 寄存器描述                                 | 144 |
| 18.4.1  | 源地址寄存器 SAR0 (偏移: 00h)                 | 145 |
| 18.4.2  | 目的地址寄存器 DAR0 (偏移: 08h)                | 145 |

|         |  |     |
|---------|--|-----|
| 18.4.3  | 控制寄存器 CTL0L (偏移: 18h)                      | 145 |
| 18.4.4  | 控制寄存器 CTL0H (偏移: 1Ch)                      | 146 |
| 18.4.5  | 设置寄存器 CFG0L (偏移: 40h)                      | 146 |
| 18.4.6  | 设置寄存器 CFG0H (偏移: 44h)                      | 147 |
| 18.4.7  | 源地址寄存器 SAR1 (偏移: 58h)                      | 148 |
| 18.4.8  | 目的地址寄存器 DAR1 (偏移: 60h)                     | 148 |
| 18.4.9  | 控制寄存器 CTL1L (偏移: 70h)                      | 148 |
| 18.4.10 | 控制寄存器 CTL1H (偏移: 74h)                      | 149 |
| 18.4.11 | 设置寄存器 CFG1L (偏移: 98h)                      | 149 |
| 18.4.12 | 设置寄存器 CFG1H (偏移: 9Ch)                      | 150 |
| 18.4.13 | 原始传输中断寄存器 RawTfr (偏移: 2C0h)                | 151 |
| 18.4.14 | 原始 Block 传输中断寄存器 RawBlock (偏移: 2C8h)       | 151 |
| 18.4.15 | 原始源传输中断寄存器 RawSrcTran (偏移: 2D0h)           | 151 |
| 18.4.16 | 原始目标传输中断寄存器 RawDstTran (偏移: 2D8h)          | 151 |
| 18.4.17 | 原始错误中断寄存器 RawErr (偏移: 2E0h)                | 151 |
| 18.4.18 | 传输中断状态寄存器 StatusTfr (偏移: 2E8h)             | 152 |
| 18.4.19 | Block 传输中断状态寄存器 StatusBlock (偏移: 2F0h)     | 152 |
| 18.4.20 | 源传输中断状态寄存器 StatusSrcTran (偏移: 2F8h)        | 152 |
| 18.4.21 | 目标传输中断状态寄存器 StatusDstTran (偏移: 300h)       | 152 |
| 18.4.22 | 错误中断状态寄存器 StatusErr (偏移: 308h)             | 153 |
| 18.4.23 | 传输中断屏蔽寄存器 MaskTfr (偏移: 310h)               | 153 |
| 18.4.24 | Block 传输中断屏蔽寄存器 MaskBlock (偏移: 318h)       | 153 |
| 18.4.25 | 源传输中断屏蔽寄存器 MaskSrcTran (偏移: 320h)          | 154 |
| 18.4.26 | 目标传输中断屏蔽寄存器 MaskDstTran (偏移: 328h)         | 154 |
| 18.4.27 | 错误中断屏蔽寄存器 MaskErr (偏移: 330h)               | 154 |
| 18.4.28 | 传输中断清除寄存器 ClearTfr (偏移: 338h)              | 154 |
| 18.4.29 | Block 传输中断清除寄存器 ClearBlock (偏移: 340h)      | 155 |
| 18.4.30 | 源传输中断清除寄存器 ClearSrcTran (偏移: 348h)         | 155 |
| 18.4.31 | 目标传输中断清除寄存器 ClearDstTran (偏移: 350h)        | 155 |
| 18.4.32 | 错误中断清除寄存器 ClearErr (偏移: 358h)              | 155 |
| 18.4.33 | 中断状态寄存器 StatusInt (偏移: 360h)               | 155 |
| 18.4.34 | 源传输 Req 信号软件握手寄存器 ReqSrc (偏移: 368h)        | 155 |
| 18.4.35 | 目标传输 Req 信号软件握手寄存器 ReqDst (偏移: 370h)       | 156 |
| 18.4.36 | 源传输 Single 信号软件握手寄存器 SglReqSrc (偏移: 378h)  | 156 |
| 18.4.37 | 目标传输 Single 信号软件握手寄存器 SglReqDst (偏移: 380h) | 156 |
| 18.4.38 | 源传输 Last 信号软件握手寄存器 LstSrc (偏移: 388h)       | 157 |
| 18.4.39 | 目标传输 Last 信号软件握手寄存器 LstDst (偏移: 390h)      | 157 |
| 18.4.40 | DMA 模块使能寄存器 DmaCfg (偏移: 398h)              | 157 |
| 18.4.41 | 通道使能寄存器 ChEn (偏移: 3A0h)                    | 157 |
| 18.4.42 | 测试模式寄存器 DmaTest (偏移: 3B0h)                 | 158 |
| 18.5    | 使用流程                                       | 158 |
| 18.5.1  | 基本硬件流控使用方法                                 | 158 |
| 18.5.2  | 软件流控使用方法                                   | 158 |
| 18.5.3  | GPIOA 和 GTimer0 中断启动 DMA 传输流程              | 159 |
| 19      | SYSTICK                                    | 161 |
| 19.1    | 概述   | 161 |
| 19.2    | 寄存器描述                                      | 161 |
| 19.2.1  | 控制和状态寄存器 SYS_CSR (偏移: 00h)                 | 161 |
| 19.2.2  | 重载值寄存器 SYS_RVR (偏移: 04h)                   | 162 |
| 19.2.3  | 当前值寄存器 SYS_CVR (SysTick->VAL) (偏移: 08h)    | 162 |
| 19.3    | 使用流程                                       | 162 |

---

|    |           |     |
|----|-----------|-----|
| 20 | 版本维护..... | 163 |
|----|-----------|-----|

Unichmicro

## 图目录

|                                     |     |
|-------------------------------------|-----|
| 图 1-1: 芯片功能框图.....                  | 3   |
| 图 1-2: 电源框图.....                    | 4   |
| 图 2-1: Cortex-M0+处理器功能框图.....       | 6   |
| 图 2-2: Cortex-M0+的寄存器组.....         | 6   |
| 图 3-1: 时钟模块框图.....                  | 8   |
| 图 6-1: CTS 时序图.....                 | 55  |
| 图 8-1: I2C 通信协议(7 位寻址)框图.....       | 70  |
| 图 8-2: I2C 通信协议(10 位寻址)框图.....      | 71  |
| 图 9-1: SPI0 数据/时钟时序图 (CPHA=0).....  | 82  |
| 图 9-2: SPI0 数据/时钟时序图 (CPHA=1).....  | 82  |
| 图 9-3: SPI0 SSN 时序图 (CPHA=0).....   | 83  |
| 图 9-4: SPI0 SSN 时序图 (CPHA=1).....   | 83  |
| 图 10-1: SPI1 数据/时钟时序图 (CPHA=0)..... | 91  |
| 图 10-2: SPI1 数据/时钟时序图 (CPHA=1)..... | 91  |
| 图 10-3: SPI1 SSN 时序图 (CPHA=0).....  | 92  |
| 图 10-4: SPI1 SSN 时序图 (CPHA=1).....  | 92  |
| 图 11-1: PWM 向上计数时序图.....            | 109 |
| 图 17-1: RTC 配置流程.....               | 142 |

## 表目录

|                            |     |
|----------------------------|-----|
| 表 3-1: 模块地址划分.....         | 7   |
| 表 3-2: 系统时钟选择.....         | 9   |
| 表 3-3: 系统复位源.....          | 9   |
| 表 3-4: 低功耗模式.....          | 10  |
| 表 3-5: 系统寄存器列表.....        | 13  |
| 表 4-1: 中断源.....            | 40  |
| 表 5-1: UART0 寄存器列表.....    | 42  |
| 表 6-1: UART1 寄存器列表.....    | 47  |
| 表 7-1: LPUART 寄存器列表.....   | 57  |
| 表 7-2: 调制控制寄存器配置建议.....    | 61  |
| 表 8-1: I2C 寄存器列表.....      | 63  |
| 表 9-1: SPI0 寄存器列表.....     | 76  |
| 表 10-1: SPI1 寄存器列表.....    | 86  |
| 表 11-1: GTIMER 寄存器列表.....  | 95  |
| 表 12-1: LPTIMER 寄存器列表..... | 113 |
| 表 13-1: ADC 寄存器列表.....     | 119 |
| 表 14-1: GPIO 寄存器列表.....    | 128 |
| 表 15-1: CRC 寄存器列表.....     | 133 |
| 表 16-1: WDT 寄存器列表.....     | 135 |
| 表 17-1: RTC 寄存器列表.....     | 139 |
| 表 18-1: 硬件握手信号.....        | 143 |
| 表 18-2: DMA 寄存器列表.....     | 144 |
| 表 19-1: SysTick 寄存器列表..... | 161 |

# 1 系统概述

UM321x 系列芯片是广芯微电子（广州）股份有限公司研制的基于 ARM Cortex-M0+内核的超低功耗、Low Pin Count、宽电压工作范围的 32 位 IoT 处理器 SoC 芯片系列，重点面向物联网行业便携式传感测量系统中的电池应用场景。依据行业应用场景的具体应用需求，芯片系统采用了独特的低功耗设计技术，内部集成了 12 位 1MSPS 的 SAR ADC 及多路 UART、SPI、I2C 等通用外围通讯接口以及 LPUART、LPTIMER 等多种超低功耗模块接口。具有高整合度、高抗干扰、高可靠性和超低功耗等技术特点。内置多种 ROOSC，可支持免晶振应用。支持 Keil MDK 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

- 工业物联网应用
- 智能交通，智慧城市，智能家居
- 智能门锁，资产追踪、无线监控等智能传感器终端应用
- 电池供电应用

## 1.1 硬件功能

### ● 超低功耗电源管理系统

- 0.7 $\mu$ A @3.0V DeepSleep+RTC 模式，低速时钟运行，IO、SRAM 以及寄存器数据保持
- 0.37 $\mu$ A @3.0V Stop 模式，所有时钟停止，IO、SRAM 以及寄存器数据保持
- 110 $\mu$ A/MHz @3.0V @32MHz Active 模式
- 3.7 $\mu$ s 快速睡眠唤醒系统
- 集成 LPTimer、LPUART、RTC、WDT
- 内置 ROOSC/LDO/POR 模块，板级系统可免去晶振/LDO/复位电路

### ● 处理器

- 32 位 ARM Cortex-M0+
- 两级流水线，系统最高主频 32MHz
- 单周期硬件乘法器
- 0 等待周期取指 @0~32MHz
- 指令效率：1.11 DMIPS/MHz in Dhrystone

### ● 存储器

- 64KB/32KB/16KB FLASH
- 8KB SRAM
- **时钟**
  - 外部高速晶振 4MHz 到 16MHz
  - 外部低速晶振 32.768kHz
  - 内部高速时钟 32MHz
  - 内部低速时钟 32kHz
- **模拟外设**
  - 1 个 12 位 1Msps ADC 转换器，具有 4 通道
- **通讯接口**
  - 2 个 UART 串口
  - 1 个低功耗 LPUART 串口
  - 2 个通用的 SPI 接口（最高速率支持 16Mbps）
  - 1 个 I2C 接口
- **定时器**
  - 4 个可编程的 32 位定时器/计数器（GTIMER），支持输入捕获和 PWM 输出
  - 1 个低功耗 16 位定时器/计数器（LPTIMER），支持 PWM 输出
  - 1 个 32 位 RTC 实时定时器/计数器
  - 1 个 32 位 WDT 看门狗定时/计数器，复位系统或产生中断
- **Buzzer 蜂鸣器**：输出频率和极性可配置
- **GPIO 通用输入/输出端口**
  - 21 个通用输入/输出管脚
  - 支持边沿/电平触发中断
  - 16/8mA 两档驱动能力可配置
- **调试模式**
  - JTAG->SWD 调试接口
- **安全功能**
  - CRC16-CCITT 数据校验算法硬件加速
  - 低电压检测(LVD)，可监控电源电压和 IO 端口电压
  - HRNG 硬件真随机数发生器
  - 128 位唯一芯片序列号
- **主要电气参数**
  - 工作电压：1.8 ~ 5.5V
  - 工作温度：-40 ~ 85°C

- ESD 防护: 2kV(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
  - JTAG->SWD 模式在线调试/下载功能
  - 完整 SDK 开发包、EVB 硬件开发套件
  - MDK 开发环境: Keil
  - 1 拖 14 并行下载器、代码烧写夹具生产工具
- **封装形式**
  - QFN20
  - QFN24

## 1.2 功能框图

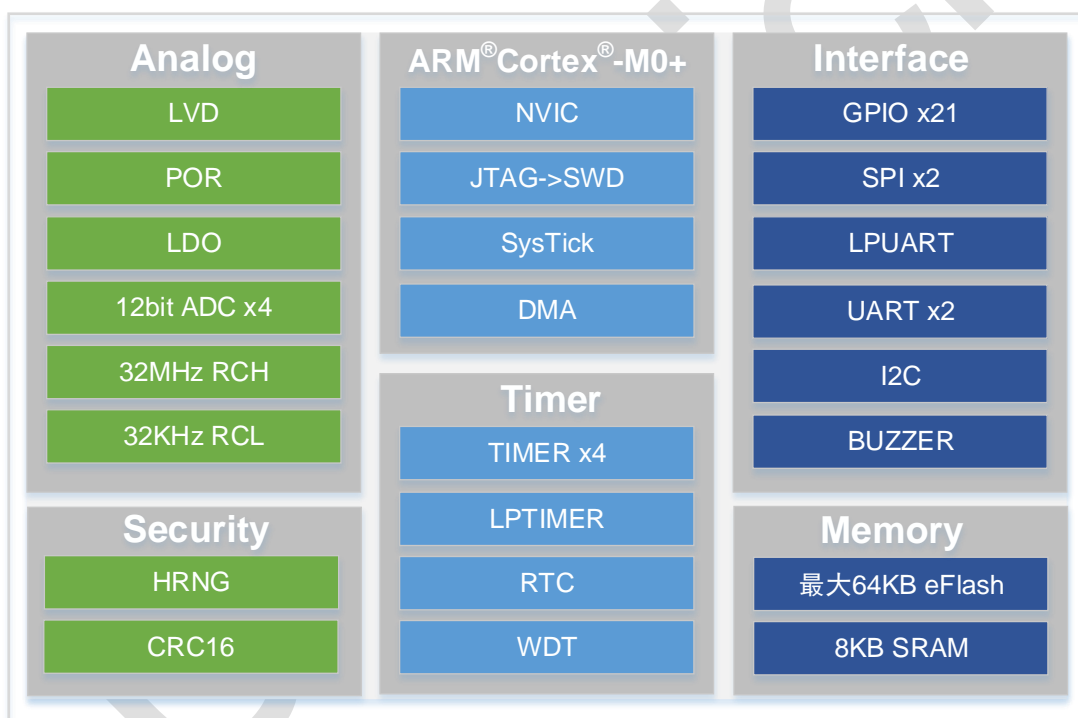


图 1-1: 芯片功能框图



## 1.3 电源框图

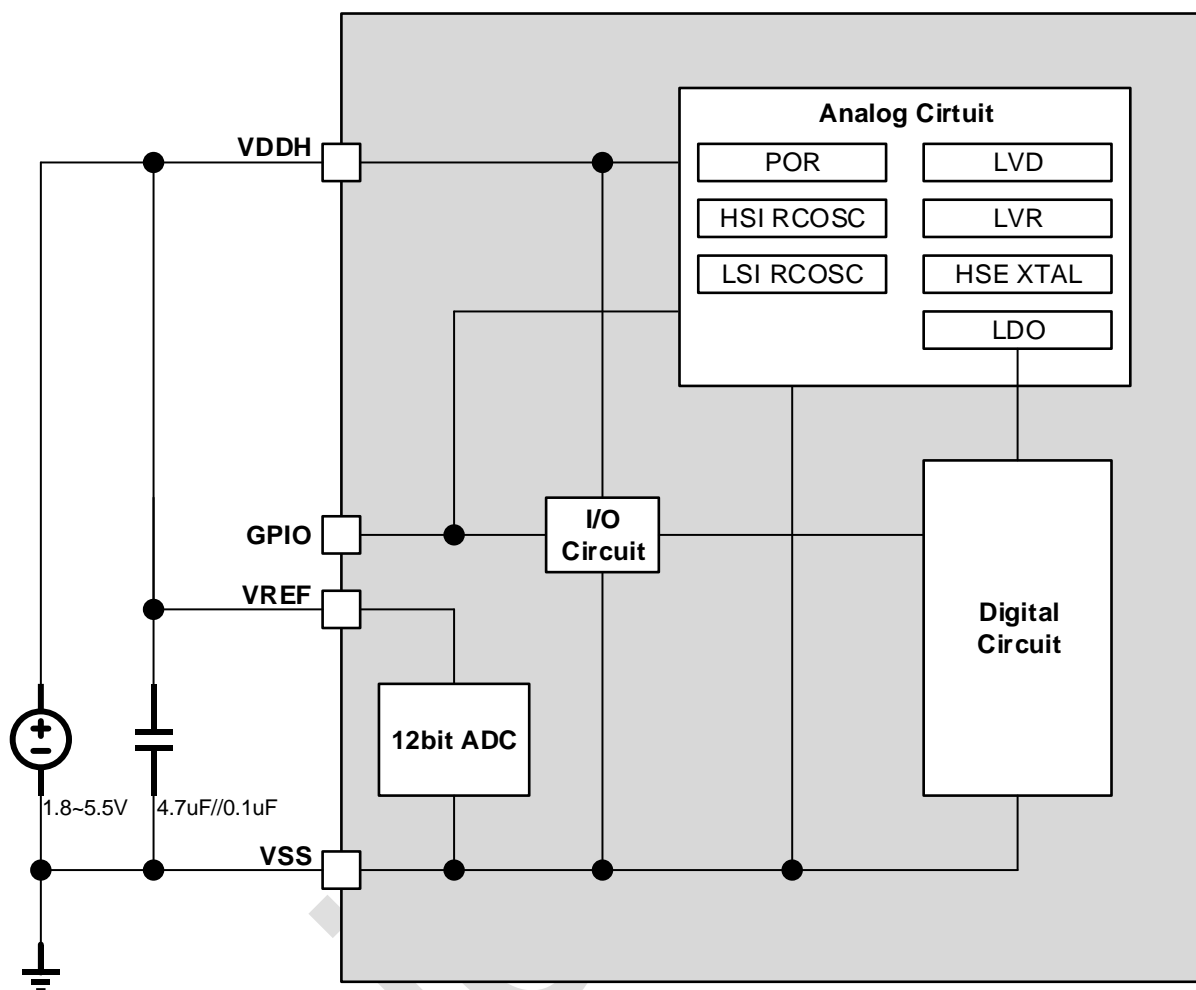


图 1-2: 电源框图

注:

- 上图所示, VDDH 和 VDDA 管脚短接在一起, 使用同一个电源供电。VDDH 和 VDDA 管脚也可以分开独立供电, 改善 ADC 的测量精度。
- 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 概述

Cortex™ M0+处理器是 32 位的两级流水线 RISC 处理器，内嵌 AMBA-Lite 接口和嵌套向量中断控制器（NVIC）。具有硬件调试功能，可以执行 Thumb 指令，并与其它 Cortex-M 系列兼容。处理器运算能力达到 1.11Drystone MIPS/MHz。同时加入多项全新设计，改进调试和追踪能力、减少每条指令循环（IPC）数量和改进 Flash 访问的两级流水线等，更纳入了节能降耗技术。Cortex M0+ 处理器全面支持已整合 Keil & IAR 调试器。

### 2.2 主要特性

- ARMv6 M Thumb
- Thumb/Thumb 2 技术
- ARMv6 M 兼容 24 位 SysTick 定时器
- 32 位硬件乘法器
- 系统接口支持小端（little-endian）数据访问
- 准确而及时的中断处理能力
- 加载、存储多个数据和多周期乘法指令可被终止然后重新开始，从而实现快速中断处理
- C 应用程序二进制接口的异常兼容模式（C-ABI）。ARMv6 M 的模式允许用户使用纯 C 函数实现中断处理
- 使用中断唤醒（WFI）与事件唤醒（WFE）指令进入低功耗的休眠模式，或者从中断退出休眠模式

## 2.3 功能框图

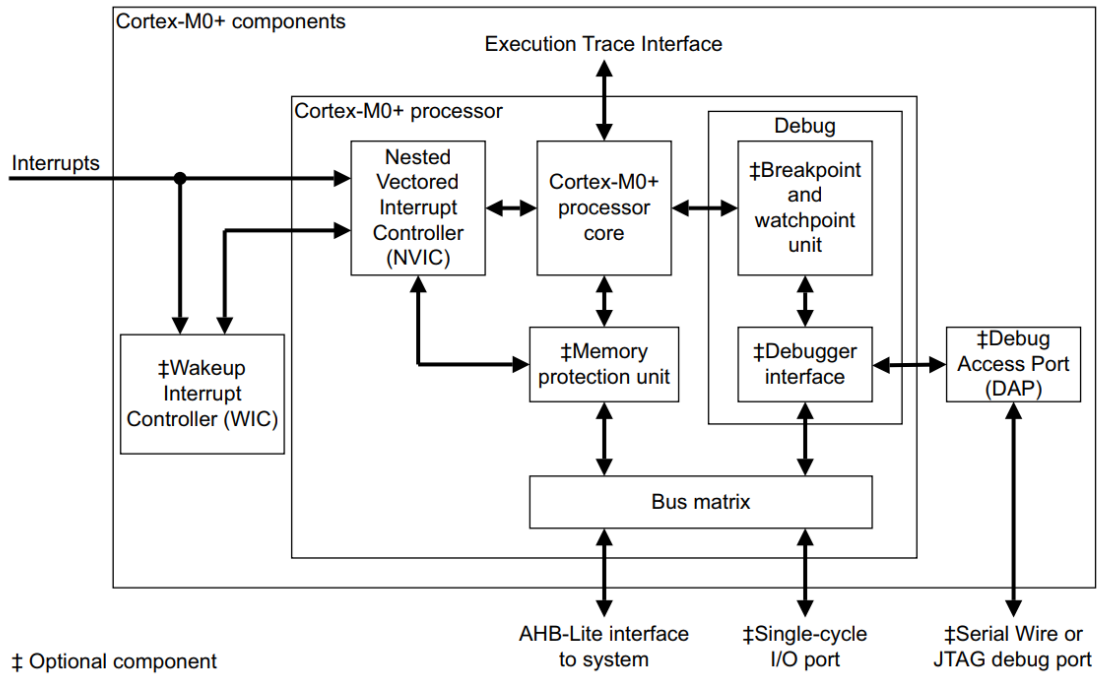


图 2-1: Cortex-M0+处理器功能框图

## 2.4 内核寄存器组

Cortex-M0+处理器寄存器组如下图:

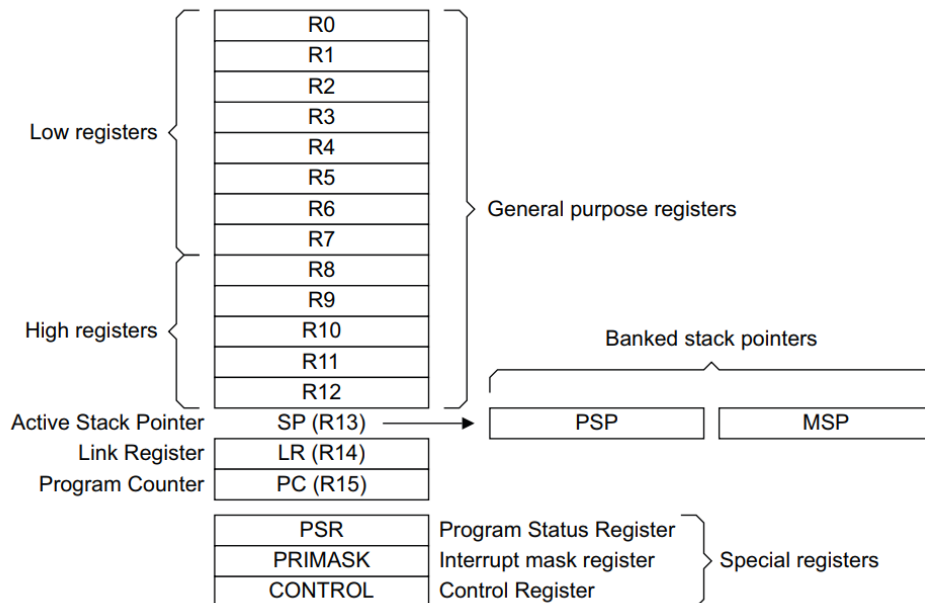


图 2-2: Cortex-M0+的寄存器组

## 3 系统配置(SCU)

### 3.1 地址映射

表 3-1: 模块地址划分

| 模块名          | 模块地址                      | 大小                        |          |
|--------------|---------------------------|---------------------------|----------|
| FLASH        | UM3211                    | 0x0000_0000 ~ 0x0000_4000 | 16KBytes |
|              | UM3212                    | 0x0000_0000 ~ 0x0000_8000 | 32KBytes |
|              | UM3213                    | 0x0000_0000 ~ 0x0001_0000 | 64KBytes |
| SRAM         | 0x2000_0000 ~ 0x2000_2000 | 8KBytes                   |          |
| Reserved     | 0x2000_2000 ~ 0x4000_0000 | -                         |          |
| UART0        | 0x4000_0000 ~ 0x4000_0400 | 1KBytes                   |          |
| LPUART       | 0x4000_0400 ~ 0x4000_0800 | 1KBytes                   |          |
| SPI0         | 0x4000_0800 ~ 0x4000_0C00 | 1KBytes                   |          |
| TIMER0/1/2/3 | 0x4000_0C00 ~ 0x4000_1000 | 1KBytes                   |          |
| LPTIMER      | 0x4000_1000 ~ 0x4000_1400 | 1KBytes                   |          |
| RTC          | 0x4000_1400 ~ 0x4000_1800 | 1KBytes                   |          |
| CRC          | 0x4000_1800 ~ 0x4000_1C00 | 1KBytes                   |          |
| ADC          | 0x4000_1C00 ~ 0x4000_2000 | 1KBytes                   |          |
| SYSREG(SCU)  | 0x4000_2000 ~ 0x4000_2400 | 1KBytes                   |          |
| WDT          | 0x4000_2400 ~ 0x4000_2800 | 1KBytes                   |          |
| Reserved     | 0x4000_2800 ~ 0x4000_2C00 | -                         |          |
| Reserved     | 0x4000_2C00 ~ 0x4000_3000 | -                         |          |
| UART1        | 0x4000_3000 ~ 0x4000_3400 | 1KBytes                   |          |
| Reserved     | 0x4000_3400 ~ 0x4000_3800 | -                         |          |
| Reserved     | 0x4000_3800 ~ 0x4000_3C00 | -                         |          |
| Reserved     | 0x4000_3C00 ~ 0x4000_4000 | -                         |          |
| GPIOA        | 0x4000_4000 ~ 0x4000_4400 | 1KBytes                   |          |
| GPIOB        | 0x4000_4400 ~ 0x4000_4800 | 1KBytes                   |          |
| GPIOC        | 0x4000_4800 ~ 0x4000_4C00 | 1KBytes                   |          |
| GIPOD        | 0x4000_4C00 ~ 0x4000_5000 | 1KBytes                   |          |
| I2C          | 0x4000_5000 ~ 0x4000_6000 | 4KBytes                   |          |
| SPI1         | 0x4000_6000 ~ 0x4000_6400 | 1KBytes                   |          |
| DMA          | 0x4002_0000 ~ 0x4002_1000 | 4KBytes                   |          |

## 3.2 时钟框图

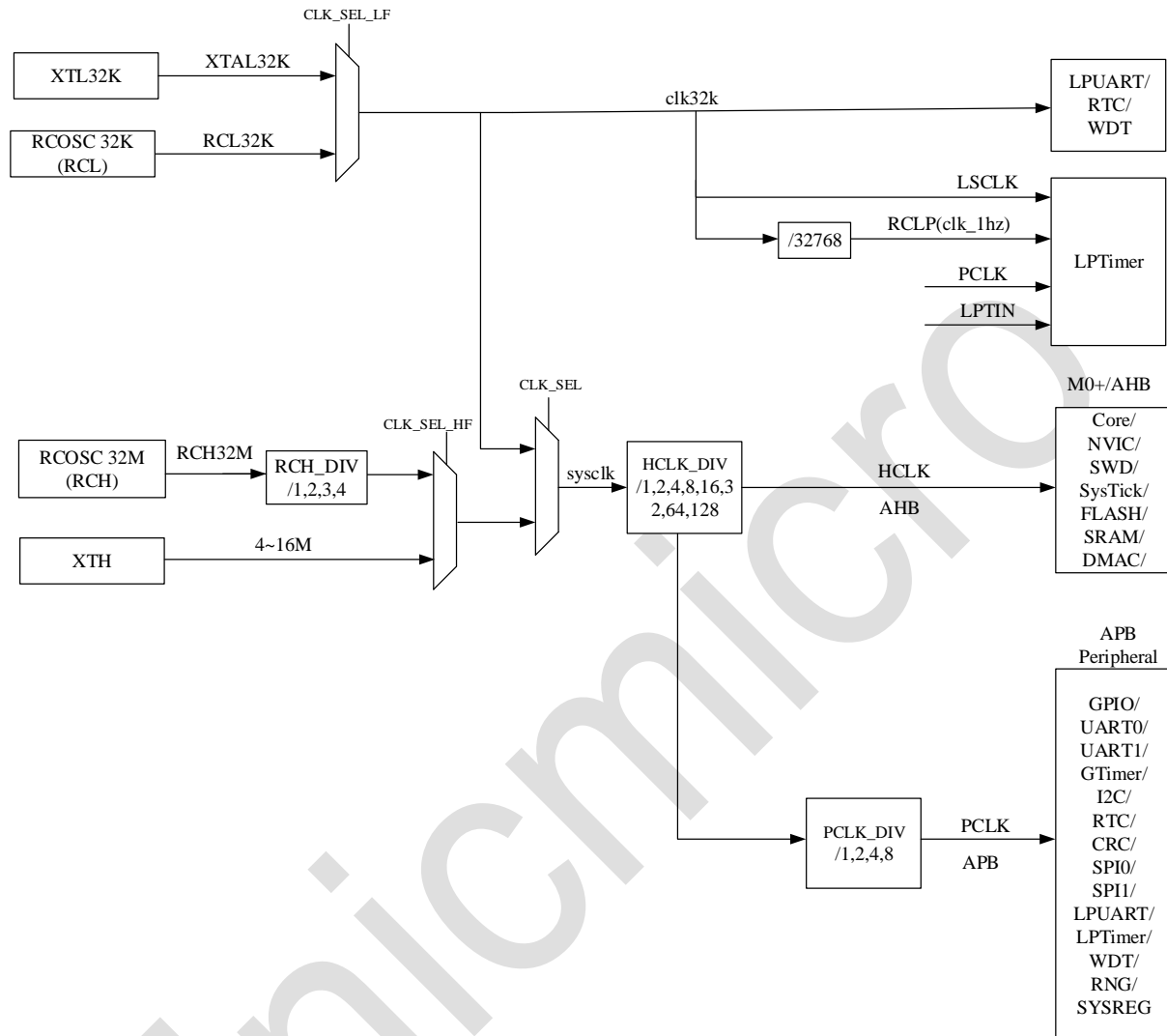


图 3-1: 时钟模块框图

## 3.3 时钟选择

芯片共有四个系统时钟源:

1. 32MHz 高精度内部时钟 RCH，作为系统时钟源。
2. 4 ~ 16MHz 的外部晶振 XTH，作为系统时钟源。
3. 32kHz 的内部时钟 RCL，作为低功耗时钟，可作为系统时钟源
4. 32.768kHz 的外部晶振 XTL，主要提供 RTC 实时时钟，也可作为系统时钟源。

根据工作模式不同，采用不同时钟方案，通过配置系统控制寄存器 0 (SYSCTRL0) [14:12]位 CLK\_SEL, CLK\_SEL\_HF 和 CLK\_SEL\_LF 来选择系统时钟的来源。关系如下表所示:

表 3-2: 系统时钟选择

| CLK_SEL | CLK_SEL_HF | 系统时钟来源 |
|---------|------------|--------|
| 0       | 0          | RCH    |
| 0       | 1          | XTH    |
| CLK_SEL | CLK_SEL_LF | 系统时钟来源 |
| 1       | 0          | RCL    |
| 1       | 1          | XTL    |

### 3.4 复位源

芯片有多个复位源, 包括 POR 复位, RESETEEN 复位, WDT 复位, SOFT\_RESETEEN 复位, 模块软件复位, LVD 复位, LOCKUP 复位。具体复位源如下表:

表 3-3: 系统复位源

| 复位源           | 描述                   |
|---------------|----------------------|
| 内部模拟 POR 上电复位 | 复位所有                 |
| RESETEEN 复位   | 复位除 CPU DEBUG 逻辑外的所有 |
| LOCKUP 复位     | 复位除EFC外的其它逻辑         |
| LVD 复位        |                      |
| WDT 复位        |                      |
| SOFT_RESETEEN |                      |
| 各模块复位         | 复位对应IP模块             |

#### 3.4.1 内部 POR 上电复位

内部上电复位 PORN: 无条件复位整个芯片。

#### 3.4.2 RESETEEN 复位

外部复位 RESETEEN 复位除 CPU DEBUG 逻辑外的整个芯片。在 RESETEEN 复位状态时, 芯片可以连接 SWD 接口。RESETEEN 管脚在上电完成后默认作为外部复位, 但通过软件关闭外部复位功能。

#### 3.4.3 WDT 复位

看门狗定时器复位除 EFC 外的整个用户电路, 缺省为释放状态;

当软件未能有效阻止超时事件时, 看门狗定时器复位。该复位仅发生在软件无法正常执行, 可能破坏数据时。在 CPU 处于 HALT 状态时, WDT 停止计数, 不会产生复位信号。

#### 3.4.4 SOFT\_RESETEEN 复位

此复位由系统产生。系统可以通过写软复位重启, 但不复位 EFC 控制器。

### 3.4.5 模块软件复位

模块软件复位，通过软件复位各数字模块。

### 3.4.6 LVD 复位

LVD 复位除 EFC 外的其它逻辑。

### 3.4.7 LOCKUP 复位

当系统连续发生两次 HardFault 时，CPU 会进入 LOCKUP 状态，系统会产生 LOCKUP 复位。LOCKUP 复位除 EFC 外的其它逻辑。

## 3.5 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，CM0+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 M0+特定指令 WFI / WFE 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，CM0+停止运行，高速时钟停止运行，低功耗功能模块（LPTIMER、LPUART、RTC、WDT）可以运行。深度休眠模式要先设置 CM0+内部的 DEEPSLEEP 寄存器，然后由 M0+特定指令 WFI / WFE 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 SYSREG 模块中 STOPMODE\_SEL 寄存器和 CM0+内部的 DEEPSLEEP 寄存器，然后由 M0+特定指令 WFI / WFE 进入，唤醒只能通过 GPIO 电平唤醒或者通过 LPTIMER 外部异步脉冲计数产生中断唤醒。

详细的描述如下表：

表 3-4：低功耗模式

| 模式    | 模式描述                                  | 进入条件   | 退出条件   |
|-------|---------------------------------------|--|--|
| Sleep | CPU 大部分休眠（包括 NVIC），WIC 不休眠；软件可关闭各模块时钟 | <ol style="list-style-type: none"> <li>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。</li> <li>2. 执行 WFI/WFE 指令。</li> </ol> | <ol style="list-style-type: none"> <li>1. CM0+检测到中断或事件发生。</li> <li>2. 进入中断服务程序清中断并返回。</li> <li>3. 继续执行后续指令。</li> </ol> |

| 模式        | 模式描述                                       | 进入条件   | 退出条件  |
|-----------|--|--|---|
| Deepsleep | CPU 大部分休眠（包括 NVIC），WIC 不休眠；高速时钟源关闭，低速时钟源运行 | <ol style="list-style-type: none"> <li>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。</li> <li>2. 设置 CM0+内部的 DEEP SLEEP 寄存器。</li> <li>3. 执行 WFI/WFE 指令。</li> </ol>  | <ol style="list-style-type: none"> <li>1. CM0+检测到中断或事件发生。</li> <li>2. 进入中断服务程序清中断并返回。</li> <li>3. 继续执行后续指令。</li> </ol>  |
| Stop      | 关闭系统所有时钟                                   | <ol style="list-style-type: none"> <li>1. 根据需要，设置 IO 唤醒或者 LPTIMER 唤醒的条件。</li> <li>2. 设置 CM0+内部的 DEEP SLEEP 寄存器。</li> <li>3. 设置 SYSREG 中的 STOPMODE_SEL 寄存器。</li> <li>4. 执行 WFI/WFE 指令。</li> </ol> | <ol style="list-style-type: none"> <li>1. 外部 IO 唤醒事件到来或者 LPTIMER 计数溢出。</li> <li>2. CM0+检测到 IO 唤醒事件中断或者 LPTIMER 溢出中断发生。</li> <li>3. 进入中断服务程序清中断并返回。</li> <li>4. 继续执行后续指令。</li> </ol> |

低功耗模式的进入和唤醒条件阐述如下：

Sleep, Deepsleep, Stop 模式进入条件都需要设置 SCB->SCR 寄存器，调用 WFI/WFE；三者模式的唤醒本质都是需要产生中断或者事件发生。

Sleep 模式下，内部高速时钟 RCH（32MHz）和内部低速时钟 RCL（32K）没有关闭，只要系统产生中断就可以唤醒退出。

Deepsleep 模式下，RCH 时钟关闭了，RCL 时钟在工作，所以只有工作在 RCL（32K）时钟源的模块如 WDT、RTC、LPTIMER、LPUART 可以产生中断唤醒退出，以及 GPIO 电平模式可以在无时钟情况下产生中断唤醒退出。

Stop 模式下，所有时钟源关闭，只能通过 GPIO 电平模式在无时钟情况下产生中断唤醒退出或者通过 LPTIMER 外部异步脉冲计数产生中断唤醒退出。

### 3.5.1 Sleep 模式

进入 sleep 模式设置：

1. SCB->SCR 的 bit2 配置为 0。
2. 调用 WFI/WFE 进入 sleep 模式。

唤醒条件：中断唤醒。

### 3.5.2 Deepsleep 模式

进入 Deepsleep 模式设置，下面以 PA3 管脚低电平唤醒为例阐述配置流程：

1. 配置端口辅助功能寄存器 GPIO\_CFG，配置为 DEEPSLEEP/STOP 模式。



2. 配置外围模块时钟控制寄存器 PERI\_CLKEN，打开 GPIOA 时钟。
3. 配置外围模块复位控制寄存器 PERI\_RESET，GPIOA 设置正常工作。
4. 配置 GPIO\_DIR 寄存器，PA3 为输入。
5. 配置 REG\_GPIO\_IEN 寄存器，禁止 PA3 中断。
6. 配置 GPIO\_IS 寄存器，触发模式为电平触发。
7. 配置 GPIO\_IEV 寄存器，低电平触发。
8. 配置 GPIO\_IC 寄存器，清除 PA3 中断状态。
9. 配置 GPIO\_IEN 寄存器，使能 PA3 中断。
10. 清除 GPIOA 中断，使能外部 GPIOA 中断。
11. SCB->SCR 的 bit2 配置为 1。
12. 调用 WFI/WFE 进入 DeepSleep 模式。

唤醒条件：WDT、RTC、LPTIMER、LPUART（XTL 时钟源情况下）中断唤醒，GPIO PA3 中断唤醒。

### 3.5.3 Stop 模式

进入 Stop 模式设置，下面以 PA3 管脚低电平唤醒为例阐述配置流程：

1. 配置端口辅助功能寄存器 GPIO\_CFG，配置为 DEEPSLEEP/STOP 模式。
2. 配置 STOPMODE\_SEL 寄存器，使能 STOP mode 停止模式有效。
3. 配置外围模块时钟控制寄存器 PERI\_CLKEN，打开 GPIOA 时钟。
4. 配置外围模块复位控制寄存器 PERI\_RESET，GPIOA 设置正常工作。
5. 配置 GPIO\_DIR 寄存器，PA3 为输入。
6. 配置 REG\_GPIO\_IEN 寄存器，禁止 PA3 中断。
7. 配置 GPIO\_IS 寄存器，触发模式为电平触发。
8. 配置 GPIO\_IEV 寄存器，低电平触发。
9. 配置 GPIO\_IC 寄存器，清除 PA3 中断状态。
10. 配置 GPIO\_IEN 寄存器，使能 PD3 中断。
11. 清除 GPIOA 中断，使能外部 GPIOA 中断。
12. SCB->SCR 的 bit2 配置为 1。
13. 调用 WFI/WFE 进入 Stop 模式。

唤醒条件：GPIO PA3 中断唤醒。

## 3.6 系统寄存器

SYSREG (SCU) 寄存器基地址: 0x40002000

表 3-5: 系统寄存器列表

| 偏置    | 名称              | 描述                |
|-------|-----------------|-------------------|
| 0x000 | SYSCTRL0        | 系统控制寄存器 0         |
| 0x004 | SYSCTRL1        | 系统控制寄存器 1         |
| 0x008 | SYSCTRL_PROTECT | 系统控制保护寄存器         |
| 0x00C | OSC_CTRL        | 时钟控制寄存器           |
| 0x010 | PERI_CLKEN      | 外围模块时钟寄存器         |
| 0x020 | RESET_FLAG      | 复位标识寄存器           |
| 0x024 | PERI_RESET      | 外围模块复位控制寄存器       |
| 0x028 | EXT_RESET_CTRL  | 外部复位滤波控制寄存器       |
| 0x030 | PA_SEL          | 端口 PA 功能配置寄存器     |
| 0x034 | PB_SEL          | 端口 PB 功能配置寄存器     |
| 0x038 | PC_SEL          | 端口 PC 功能配置寄存器     |
| 0x03C | PD_SEL          | 端口 PD 功能配置寄存器     |
| 0x040 | PAD_ADS         | 端口数模配置寄存器         |
| 0x044 | PAD_DR          | 端口驱动能力配置寄存器       |
| 0x048 | PAD_PU          | 端口上拉配置寄存器         |
| 0x04C | PAD_PD          | 端口下拉配置寄存器         |
| 0x050 | PAD_OD          | 端口开漏输出配置寄存器       |
| 0x054 | PAD_CS          | 端口输入类型配置寄存器       |
| 0x058 | PAD_IE          | 端口输入配置寄存器         |
| 0x05C | IOCTRL_PROTECT  | IO 控制保护寄存器        |
| 0x060 | IOWACK_INTR_CLR | 唤醒状态清除寄存器         |
| 0x064 | LVD_CFG         | LVD 控制寄存器         |
| 0x070 | GPIO_CFG        | 端口辅助功能寄存器         |
| 0x074 | EXTRST_SEL      | 外部复位端口选择寄存器       |
| 0x078 | STOPMODE_SEL    | 停止模式选择寄存器         |
| 0x07C | REMAP_ADDR      | REMAP 寄存器         |
| 0x080 | VECTOR_OFFSET   | 中断向量地址重映射寄存器      |
| 0x084 | HRNG_CR         | 随机数控制寄存器          |
| 0x088 | HRNG_SEED       | 随机数种子寄存器          |
| 0x08C | HRNG_DATA       | 随机数数据寄存器          |
| 0x090 | BUZZERCR        | 蜂鸣器控制寄存器/BUZZERCR |
| 0x0A0 | Reserved0       | 保留寄存器 0           |
| 0x0A4 | Reserved1       | 保留寄存器 1           |

### 3.6.1 系统控制寄存器 0 SYSCTRL0 (偏移: 000h)

| 比特    | 名称  | 属性 | 复位值 | 描述 |
|-------|-----|----|-----|----|
| 31:23 | RSV | -  | -   | 保留 |

| 比特    | 名称                | 属性  | 复位值 | 描述  |
|-------|-------------------|-----|-----|---|
| 22    | LPTIMER_EXTIG_SEL | R/W | 0   | LPTIMER 的 EXTIG（外部触发）信号来源选择信号。<br>1: EXTIG 来自于 RTC 的 1HZ 时钟输出;<br>0: EXTIG 来自于外部引脚;   |
| 21    | LPTIMER_LPTIN_SEL | R/W | 0   | LPTIMER 的 LPTIN（外部时钟输入）信号来源选择信号。<br>1: LPTIN 来自于 RTC 的 1HZ 时钟输出;<br>0: LPTIN 来自于外部引脚;   |
| 20    | DMA_C7HS_G0GT_SEL | R/W | 0   | DMA 第七个握手信号选择。<br>1: 选择 GPIO0 的中断信号作为启动 DMA 搬移的信号;<br>0: 选择 GTIMER 的 TIMER0 的中断信号作为启动 DMA 搬移的信号。  |
| 19    | DMA_C7HS_EN       | R/W | 0   | DMA 第七个握手信号使能。当使能此位后, GPIO0 或 GTIMER 的中断信号可以启动 DMA 搬移。<br>1: 使能<br>0: 不使能   |
| 18:17 | RCH_DIV           | R/W | 1   | RCH 时钟源分频设置位<br>00: 不分频<br>01: 二分频<br>10: 三分频<br>11: 四分频  |
| 16    | SWD_WACK_EN       | R/W | 1   | 连接 ULINK 或者 JLINK 后, 在 SLEEP、DEEPSLEEP 或者 STOP 模式下是否唤醒系统的使能位。<br>1: 连接 ULINK 或者 JLINK 后, 在 SLEEP、DEEPSLEEP 或者 STOP 模式下, 使用 NMI 中断唤醒系统;<br>0: 连接 ULINK 或者 JLINK 后, 在 SLEEP、DEEPSLEEP 或者 STOP 模式下, 不唤醒系统。 |
| 15    | Wakeup_byRCH      | R/W | 0   | 1: 从 Deep Sleep 唤醒后, system clock 来源为 RCH, 原时钟继续使能。<br>0: 从 Deep Sleep 唤醒后, 不改变 system clock 来源。  |
| 14    | CLK_SEL           | R/W | 0   | 系统时钟来源选择<br>0: 高速时钟 CLK_SEL_HF<br>1: 低速时钟 CLK_SEL_LF  |
| 13    | CLK_SEL_HF        | R/W | 0   | 0: 高频时钟选择内部高速时钟 RCH<br>1: 高频时钟选择外部高速时钟 XTH  |
| 12    | CLK_SEL_LF        | R/W | 0   | 0: 低频时钟选择内部低速时钟 RCL<br>1: 低频时钟选择外部低速时钟 XTL  |
| 11    | RSV               | -   | -   | 保留  |
| 10:9  | PCLK_DIV          | R/W | 0   | PCLK 分频选择<br>00: HCLK<br>01: HCLK/2<br>10: HCLK/4<br>11: HCLK/8   |

| 比特  | 名称       | 属性  | 复位值 | 描述   |
|-----|----------|-----|-----|--|
| 8:6 | HCLK_DIV | R/W | 0   | HCLK 分频选择<br>000: SystemClk<br>001: SystemClk/2<br>010: SystemClk/4<br>011: SystemClk/8<br>100: SystemClk/16<br>101: SystemClk/32<br>110: SystemClk/64<br>111: SystemClk/128 |
| 5:4 | RSV      | -   | -   | 保留   |
| 3   | XTL_EN   | R/W | 0   | 外部低速晶振 XTL 使能控制<br>0: 关闭 1: 使能<br>注: 需要将 PB4、PB5 设置成模拟端口。此位寄存器只能由 POR 和外部管脚复位。   |
| 2   | RCL_EN   | R/W | 1   | 内部低速时钟 RCL 使能控制<br>0: 关闭<br>1: 使能  |
| 1   | XTH_EN   | R/W | 0   | 外部高速晶振 XTH 使能控制<br>0: 关闭<br>1: 使能<br>注: 当系统进入 DeepSleep, 此高速时钟会自动关闭。此位寄存器只能由 POR 和外部管脚复位。  |
| 0   | RCH_EN   | R/W | 1   | 内部高速时钟 RCH 使能信号。<br>0: 关闭<br>1: 使能<br>注: 当系统进入 DeepSleep, 此高速时钟会自动关闭。  |

### 3.6.2 系统控制寄存器 1 SYSCTRL1 (偏移: 004h)

| 比特    | 名称       | 属性  | 复位值 | 描述   |
|-------|----------|-----|-----|--|
| 31:10 | RSV      | -   | -   | 保留   |
| 9:8   | XTL_RTRM | R/W | 1   | XTL RTRM 信号配置。默认值是 2'b01, 建议不要修改。<br>此位寄存器只能由 POR 和外部管脚复位                  |
| 7:5   | RSV      | -   | -   | 保留   |
| 4     | XTH_BS   | R/W | 0   | XTH 的频段选择:<br>0: 0.4MHz~2.5MHz<br>1: 2.5MHz~16MHz<br>此位寄存器只能由 POR 和外部管脚复位。 |
| 3:2   | RSV      | -   | -   | 保留   |

| 比特 | 名称      | 属性  | 复位值 | 描述   |
|----|---------|-----|-----|--|
| 1  | EXTL_EN | R/W | 0   | 外部 XTL 时钟输入控制<br>1: XTL 输出时钟从 PB4 输入。<br>0: XTL 输出时钟由晶振产生。<br>注: 使用 Pb4 输入时钟时, 需设置 SYSCTRL0.XTL_EN 为 1。<br>此位寄存器只能由 POR 和外部管脚复位。 |
| 0  | EXTH_EN | R/W | 0   | 外部 XTH 输入控制<br>1: XTH 输出时钟从 PA1 输入。<br>0: XTH 输出时钟由晶振产生。<br>注: 使用 PA1 输入时钟时, 需设置 SYSCTRL0.XTH_EN 为 1。<br>此位寄存器只能由 POR 和外部管脚复位。   |

### 3.6.3 系统控制保护寄存器 SYSCTRL\_PROTECT (偏移: 008h)

| 比特   | 名称              | 属性  | 复位值 | 描述   |
|------|-----------------|-----|-----|--|
| 31:0 | SYSCTRL_PROTECT | R/W | 0   | 寄存器 SYSCTRL0 和 SYSCTRL1 写保护的控制寄存器。给此寄存器写 0xA5A5_5A5A, 启动寄存器 SYSCTRL0 和 SYSCTRL1 的写使能。给此寄存器写其他值, 关闭它们的写使能。SYSCTRL0 或 SYSCTRL1 寄存器配置完后, 它们的写使能会自动关闭。<br>读取此寄存器返回 SYSCTRL0 和 SYSCTRL1 寄存器的写使能状态。<br>0: 写未使能;<br>1: 写已经使能。 |

### 3.6.4 时钟控制寄存器 OSC\_CTRL (偏移: 0x00Ch)

| 比特    | 名称          | 属性  | 复位值    | 描述   |
|-------|-------------|-----|--------|--|
| 31:24 | RSV         | -   | -      | 保留   |
| 27:16 | WAKP_delay  | R/W | 12'hC8 | 系统从 DEEPSLEEP/STOP 模式下唤醒, CLK 给出的时间延时。此寄存器给出的时间为系统时钟计数的时钟周期数。                        |
| 15    | RSV         | -   | -      | 保留   |
| 14    | XTH_stable  | R   | 0      | 外部高速时钟 XTH 稳定标志位<br>1: 代表 XTH 已经稳定, 可以被内部电路使用<br>0: 代表 XTH 未稳定, 不可以被内部电路使用           |
| 13:12 | XTH_startup | R/W | 2'b10  | 外部高速时钟 XTH 稳定时间选择<br>11: 65535 个周期<br>10: 32768 个周期<br>01: 16384 个周期<br>00: 4096 个周期 |

| 比特  | 名称          | 属性  | 复位值   | 描述  |
|-----|-------------|-----|-------|---|
| 11  | RSV         | -   | -     | 保留  |
| 10  | RCH_stable  | R   | 0     | 内部高速时钟 RCH 稳定标志位<br>1: 代表 RCH 已经稳定, 可以被内部电路使用<br>0: 代表 RCH 未稳定, 不可以被内部电路使用          |
| 9:8 | RCH_startup | R/W | 2'b00 | 内部高速时钟 RCH 稳定时间选择<br>11: 256 个周期<br>10: 64 个周期<br>01: 16 个周期<br>00: 4 个周期           |
| 7   | RSV         | -   | -     | 保留  |
| 6   | XTL_stable  | R   | 0     | 外部低速时钟 XTL 稳定标志位<br>1: 代表 XTL 已经稳定, 可以被内部电路使用<br>0: 代表 XTL 未稳定, 不可以被内部电路使用          |
| 5:4 | XTL_startup | R/W | 2'b10 | 外部低速时钟 XTL 稳定时间选择<br>11: 32768 个周期<br>10: 16384 个周期<br>01: 4096 个周期<br>00: 1024 个周期 |
| 3   | RSV         | -   | -     | 保留  |
| 2   | RCL_stable  | R/W | 0     | 内部低速时钟 RCL 稳定标志位<br>1: 代表 RCL 已经稳定, 可以被内部电路使用<br>0: 代表 RCL 未稳定, 不可以被内部电路使用          |
| 1:0 | RCL_startup | R/W | 2'b00 | 内部低速时钟 RCL 稳定时间选择<br>11: 256 个周期<br>10: 64 个周期<br>01: 16 个周期<br>00: 4 个周期           |

### 3.6.5 外围模块时钟寄存器 PERI\_CLKEN (偏移: 010h)

| 比特    | 名称          | 属性  | 复位值 | 描述                          |
|-------|-------------|-----|-----|-----------------------------|
| 31:23 | RSV         | -   | -   | 保留                          |
| 22    | DMA_CLKEN   | R/W | 0   | DMA 模块时钟使能<br>1: 使能 0: 关闭   |
| 21    | SPI1_CLKEN  | R/W | 0   | SPI1 模块时钟使能<br>1: 使能 0: 关闭  |
| 20    | EFC_CLKEN   | R/W | 1   | EFC 模块时钟使能<br>1: 使能 0: 关闭   |
| 19    | GPIOD_CLKEN | R/W | 0   | GPIOD 模块时钟使能<br>1: 使能 0: 关闭 |
| 18    | GPIOC_CLKEN | R/W | 0   | GPIOC 模块时钟使能<br>1: 使能 0: 关闭 |
| 17    | GPIOB_CLKEN | R/W | 0   | GPIOB 模块时钟使能<br>1: 使能 0: 关闭 |
| 16    | GPIOA_CLKEN | R/W | 0   | GPIOA 模块时钟使能<br>1: 使能 0: 关闭 |

| 比特  | 名称           | 属性  | 复位值 | 描述                            |
|-----|--------------|-----|-----|-------------------------------|
| 15  | I2C_CLKEN    | R/W | 0   | I2C 控制器模块时钟使能<br>1: 使能 0: 关闭  |
| 14  | ADC_CLKEN    | R/W | 0   | ADC 控制器模块时钟使能<br>1: 使能 0: 关闭  |
| 13  | RTC_CLKEN    | R/W | 1   | RTC 模块时钟使能<br>1: 使能 0: 关闭     |
| 12  | WDT_CLKEN    | R/W | 0   | WDT 模块时钟使能<br>1: 使能 0: 关闭     |
| 11  | CRC_CLKEN    | R/W | 0   | CRC 模块时钟使能<br>1: 使能 0: 关闭     |
| 10  | UART1_CLKEN  | R/W | 0   | UART1 模块时钟使能<br>1: 使能 0: 关闭   |
| 9   | GTIM_CLKEN   | R/W | 0   | GTimer 模块时钟使能<br>1: 使能 0: 关闭  |
| 8   | LPTIM_CLKEN  | R/W | 0   | LPTimer 模块时钟使能<br>1: 使能 0: 关闭 |
| 7:5 | RSV          | -   | -   | 保留                            |
| 4   | SPI0_CLKEN   | R/W | 0   | SPI0 模块时钟使能<br>1: 使能 0: 关闭    |
| 3:2 | RSV          | -   | -   | 保留                            |
| 1   | LPUART_CLKEN | R/W | 0   | LPUART 模块时钟使能<br>1: 使能 0: 关闭  |
| 0   | UART0_CLKEN  | R/W | 0   | UART0 模块时钟使能<br>1: 使能 0: 关闭   |

### 3.6.6 复位标识寄存器 RESET\_FLAG (偏移: 020h)

| 比特   | 名称               | 属性  | 复位值 | 描述  |
|------|------------------|-----|-----|---|
| 31:8 | RSV              | -   | -   | 保留  |
| 7    | SYSRESETREQ_FLAG | R/W | 0   | CPU 复位状态。需要软件初始化和清除。写 1 清零。<br>1: M0 SYSTEMRESETN 复位发生<br>0: 无复位发生              |
| 6    | LOCKUP_RSTN_FLAG | R/W | 0   | CPU 死锁复位状态。需要软件初始化和清除。写 1 清零。<br>1: Lockup 复位发生<br>0: 无复位发生                     |
| 5    | LVD_RSTN_FLAG    | R/W | 0   | 低电压复位状态。需要软件初始化和清除。写 1 清零。<br>1: LVD 复位发生<br>0: 无复位发生                           |
| 4    | SOFTTRSTN_FLAG   | R/W | 0   | 软件复位状态。需要软件初始化和清除。写 1 清零。<br>1: 写 REMAP_ADDR 寄存器的 SOFT_RESETN 位发生复位<br>0: 无复位发生 |

| 比特 | 名称          | 属性  | 复位值 | 描述   |
|----|-------------|-----|-----|--|
| 3  | WDT_FLAG    | R/W | 0   | 看门狗复位状态。需要软件初始化和清除。写 1 清零。<br>1: WDT 复位发生<br>0: 无复位发生      |
| 2  | RESETN_FLAG | R/W | 0   | 外部复位状态。需要软件初始化和清除。写 1 清零。<br>1: 外部 RESETN 复位发生<br>0: 无复位发生 |
| 1  | PORN15_FLAG | R/W | 0   | POR15 复位状态, 写 1 清零。<br>1: POR 复位发生<br>0: 无复位发生             |
| 0  | PORN33_FLAG | R/W | 0   | POR33 复位状态, 写 1 清零。<br>1: POR 复位发生<br>0: 无复位发生             |

Note: 这些寄存器只能被 PORN 复位

### 3.6.7 外围模块复位控制寄存器 PERI\_RESET (偏移: 024h)

| 比特    | 名称          | 属性  | 复位值 | 描述                                    |
|-------|-------------|-----|-----|---------------------------------------|
| 31:23 | RSV         | -   | -   | 保留                                    |
| 22    | DMA_RESET   | R/W | 0   | DMA 控制器模块复位使能<br>1: 正常工作 0: 模块处于复位状态  |
| 21    | SPI1_RESET  | R/W | 0   | SPI1 控制器模块复位使能<br>1: 正常工作 0: 模块处于复位状态 |
| 20    | RSV         | -   | -   | 保留                                    |
| 19    | GPIOD_RESET | R/W | 0   | GPIOD 模块复位使能<br>1: 正常工作 0: 模块处于复位状态   |
| 18    | GPIOC_RESET | R/W | 0   | GPIOC 模块复位使能<br>1: 正常工作 0: 模块处于复位状态   |
| 17    | GPIOB_RESET | R/W | 0   | GPIOB 模块复位使能<br>1: 正常工作 0: 模块处于复位状态   |
| 16    | GPIOA_RESET | R/W | 0   | GPIOA 模块复位使能<br>1: 正常工作 0: 模块处于复位状态   |
| 15    | I2C_RESET   | R/W | 0   | I2C 模块复位使能<br>1: 正常工作 0: 模块处于复位状态     |
| 14    | ADC_RESET   | R/W | 0   | ADC 控制器模块复位使能<br>1: 正常工作 0: 模块处于复位状态  |
| 13    | RTC_RESET   | R/W | 1   | RTC 模块复位使能<br>1: 正常工作 0: 模块处于复位状态     |
| 12    | WDT_RESET   | R/W | 0   | WDT 模块复位使能<br>1: 正常工作 0: 模块处于复位状态     |
| 11    | CRC_RESET   | R/W | 0   | CRC 模块复位使能<br>1: 正常工作 0: 模块处于复位状态     |
| 10    | UART1_RESET | R/W | 0   | UART1 模块复位使能<br>1: 正常工作 0: 模块处于复位状态   |



| 比特  | 名称           | 属性  | 复位值 | 描述                                    |
|-----|--------------|-----|-----|---------------------------------------|
| 9   | GTIM_RESET   | R/W | 0   | GTimer 模块复位使能<br>1: 正常工作 0: 模块处于复位状态  |
| 8   | LPTIM_RESET  | R/W | 0   | LPTimer 模块复位使能<br>1: 正常工作 0: 模块处于复位状态 |
| 7:5 | RSV          | -   | -   | 保留                                    |
| 4   | SPI0_RESET   | R/W | 0   | SPI0 控制器模块复位使能<br>1: 正常工作 0: 模块处于复位状态 |
| 3:2 | RSV          | -   | -   | 保留                                    |
| 1   | LPUART_RESET | R/W | 0   | LPUART 模块复位使能<br>1: 正常工作 0: 模块处于复位状态  |
| 0   | UART0_RESET  | R/W | 0   | UART0 模块复位使能<br>1: 正常工作 0: 模块处于复位状态   |

### 3.6.8 外部复位滤波控制寄存器 EXT\_RESET\_CTRL (偏移: 028h)

| 比特   | 名称            | 属性  | 复位值 | 描述   |
|------|---------------|-----|-----|--|
| 31:1 | RSV           | -   | -   | 保留   |
| 0    | EXT_FILTER_EN | R/W | 0   | 外部复位滤波使能位。<br>1: 外部复位滤波使能;<br>0: 外部复位滤波禁止; |

### 3.6.9 端口 PA 功能配置寄存器 PA\_SEL (偏移: 030h)

| 比特    | 名称      | 属性  | 复位值    | 描述   |
|-------|---------|-----|--------|--|
| 31:19 | RSV     | -   | -      | 保留   |
| 18:16 | PA4_SEL | R/W | 3'b001 | 端口 PA4 功能选择<br>3'b000: GPIO PA4<br>3'b001: RSTOut  |
| 15    | RSV     | -   | -      | 保留   |
| 14:12 | PA3_SEL | R/W | 0      | 端口 PA3 功能选择<br>3'b000: GPIO PA3<br>3'b001: SPI0_CS0(只能与 SPI0_MISO0 搭配使用)<br>3'b010: LPTIM_EXT<br>3'b011: CLK_1HZ<br>3'b100: BUZZ_OUT |
| 11    | RSV     | -   | -      | 保留   |

| 比特   | 名称      | 属性  | 复位值 | 描述  |
|------|---------|-----|-----|---|
| 10:8 | PA2_SEL | R/W | 0   | 端口 PA2 功能选择<br>3'b000: GPIO PA2<br>3'b001: UART0_TXD<br>3'b010: TIM0_TOGN<br>3'b011: SPI0_CS0 (只能与 SPI0_MISO0 搭配使用)<br>3'b100: I <sup>2</sup> C_SCL |
| 7    | RSV     | -   | -   | 保留  |
| 6:4  | PA1_SEL | R/W | 0   | 端口 PA1 功能选择<br>3'b000: GPIO PA1<br>3'b001: UART0_RXD<br>3'b010: TIM0_TOG<br>3'b011: SPI0_SCK<br>3'b100: TIM2_EXT                                    |
| 3:0  | RSV     | -   | -   | 保留<br>注意: PA0 的输入功能描述详见外部复位端口选择寄存器 (EXTRST_SEL)。  |

### 3.6.10 端口 PB 功能配置寄存器 PB\_SEL (偏移: 034h)

| 比特    | 名称      | 属性  | 复位值 | 描述   |
|-------|---------|-----|-----|--|
| 31:23 | RSV     | -   | -   | 保留   |
| 22:20 | PB5_SEL | R/W | 0   | 端口 PB5 功能选择<br>3'b000: GPIO PB5<br>3'b001: TIM2_TOG<br>3'b010: LPTIM_GATE<br>3'b011: SPI0_SCK<br>3'b100: UART0_RXD<br>3'b101: LVD_OUT<br>3'b110: UART1_TXD |
| 19    | RSV     | -   | -   | 保留   |
| 18:16 | PB4_SEL | R/W | 0   | 端口 PB4 功能选择<br>3'b000: GPIO PB4<br>3'b001: TIM2_TOGN<br>3'b010: SPI0_CS1 (只能与 SPI0_MISO1 搭配使用)<br>3'b011: UART1_RXD<br>3'b100: UART0_TXD                   |
| 15:0  | RSV     | -   | -   | 保留   |

### 3.6.11 端口 PC 功能配置寄存器 PC\_SEL (偏移: 038h)

| 比特    | 名称      | 属性  | 复位值    | 描述  |
|-------|---------|-----|--------|---|
| 31    | RSV     | -   | -      | 保留  |
| 30:28 | PC7_SEL | R/W | 3'b101 | 端口 PC7 功能选择<br>3'b000: GPIO PC7<br>3'b001: SPI0_MISO0 (只能与 SPI0_CS0 搭配使用)<br>3'b010: UART0_RXD<br>3'b011: RCH_OUT<br>3'b100: XTH_OUT<br>3'b101: SWDIO         |
| 27    | RSV     | -   | -      | 保留  |
| 26:24 | PC6_SEL | R/W | 0      | 端口 PC6 功能选择<br>3'b000: GPIO PC6<br>3'b001: SPI0_MOSI<br>3'b010: SPI0_CS0 (只能与 SPI0_MISO0 搭配使用)<br>3'b011: UART1_RTS<br>3'b100: LPUART_TXD<br>3'b101: SPI1_CLK |
| 23    | RSV     | -   | -      | 保留  |
| 22:20 | PC5_SEL | R/W | 0      | 端口 PC5 功能选择<br>3'b000: GPIO PC5<br>3'b001: SPI0_SCK<br>3'b010: LVD_OUT<br>3'b011: UART1_CTS<br>3'b100: LPUART_RXD<br>3'b101: SPI1_MOSI                        |
| 19    | RSV     | -   | -      | 保留  |
| 18:16 | PC4_SEL | R/W | 0      | 端口 PC4 功能选择<br>3'b000: GPIO PC4<br>3'b001: UART1_RXD<br>3'b010: UART0_TXD<br>3'b011: SPI0_MOSI<br>3'b100: HCLK_OUT<br>3'b101: SPI1_CS                         |
| 15    | RSV     | -   | -      | 保留  |
| 14:12 | PC3_SEL | R/W | 0      | 端口 PC3 功能选择<br>3'b000: GPIO PC3<br>3'b001: UART1_TXD<br>3'b010: UART0_RXD<br>3'b011: SPI0_MISO0 (只能与 SPI0_CS0 搭配使用)   |
| 11:7  | RSV     | -   | -      | 保留  |

| 比特  | 名称      | 属性  | 复位值    | 描述   |
|-----|---------|-----|--------|--|
| 6:4 | PC1_SEL | R/W | 3'b001 | 端口 PC1 功能选择<br>3'b000: GPIO PC1<br>3'b001: HCLK_OUT<br>3'b010: XTH_OUT<br>3'b011: XTL_OUT<br>3'b100: RCL_OUT |
| 3:0 | RSV     | -   | -      | 保留   |

### 3.6.12 端口 PD 功能配置寄存器 PD\_SEL (偏移: 03Ch)

| 比特    | 名称      | 属性  | 复位值 | 描述   |
|-------|---------|-----|-----|--|
| 31:27 | RSV     | -   | -   | 保留   |
| 26:24 | PD6_SEL | R/W | 0   | 端口 PD6 功能选择<br>3'b000: GPIO PD6<br>3'b001: UART0_RXD<br>3'b010: SPI0_MOSI<br>3'b011: I2C_SDA<br>3'b100: SPI1_CS  |
| 23    | RSV     | -   | -   | 保留   |
| 22:20 | PD5_SEL | R/W | 0   | 端口 PD5 功能选择<br>3'b000: GPIO PD5<br>3'b001: UART0_TXD<br>3'b010: I2C_SCL<br>3'b011: LPTIM_OUT<br>3'b100: SPI0_MISO0 (只能与 SPI0_CS0 搭配使用)<br>3'b101: SPI1_MISO  |
| 19    | RSV     | -   | -   | 保留   |
| 18:16 | PD4_SEL | R/W | 0   | 端口 PD4 功能选择<br>3'b000: GPIO PD4<br>3'b001: LPUART_TXD<br>3'b010: TIM0_EXT<br>3'b011: I2C_SDA<br>3'b100: CLK_1HZ<br>3'b101: TIM1_TOG<br>3'b110: SPI0_MISO1 (只能与 SPI0_CS1 搭配使用)<br>3'b111: SPI1_MISO |
| 15    | RSV     | -   | -   | 保留   |

| 比特    | 名称      | 属性  | 复位值    | 描述   |
|-------|---------|-----|--------|--|
| 14:12 | PD3_SEL | R/W | 0      | 端口 PD3 功能选择<br>3'b000: GPIO PD3<br>3'b001: LPUART_RXD<br>3'b010: UART0_RXD<br>3'b011: XTL_OUT<br>3'b100: TIM1_TOGN<br>3'b101: BUZZ_OUT<br>3'b110: SPI0_MISO1 (只能与 SPI0_CS1 搭配使用)<br>3'b111: SPI1_CLK |
| 11    | RSV     | -   | -      | 保留   |
| 10:8  | PD2_SEL | R/W | 0      | 端口 PD2 功能选择<br>3'b000: GPIO PD2<br>3'b001: TIM3_TOGN<br>3'b010: UART0_TXD<br>3'b011: CLK_1HZ<br>3'b100: SPI0_CS1 (只能与 SPI0_MISO1 搭配使用)<br>3'b101: BUZZ_OUT<br>3'b110: SPI1_MOSI                      |
| 7     | RSV     | -   | -      | 保留   |
| 6:4   | PD1_SEL | R/W | 3'b111 | 端口 PD1 功能选择<br>3'b000: GPIO PD1<br>3'b001: TIM3_TOG<br>3'b010: UART0_TXD<br>3'b011: PCLK_OUT<br>3'b100: RCL_OUT<br>3'b101: RCH_OUT<br>3'b110: HCLK_OUT<br>3'b111: SWCLK                              |
| 3:0   | RSV     | -   | -      | 保留   |

### 3.6.13 端口数模配置寄存器 PAD\_ADS (偏移: 040h)

| 比特 | 名称      | 属性  | 复位值 | 描述   |
|----|---------|-----|-----|--|
| 31 | RSV     | -   | -   | 保留   |
| 30 | PD6_ADS | R/W | 0   | 端口 PD6 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 29 | PD5_ADS | R/W | 0   | 端口 PD5 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 28 | PD4_ADS | R/W | 0   | 端口 PD4 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |

| 比特    | 名称      | 属性  | 复位值 | 描述   |
|-------|---------|-----|-----|--|
| 27    | PD3_ADS | R/W | 0   | 端口 PD3 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 26    | PD2_ADS | R/W | 0   | 端口 PD2 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 25    | PD1_ADS | R/W | 0   | 端口 PD1 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 24    | RSV     | -   | -   | 保留   |
| 23    | PC7_ADS | R/W | 0   | 端口 PC7 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 22    | PC6_ADS | R/W | 0   | 端口 PC6 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 21    | PC5_ADS | R/W | 0   | 端口 PC5 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 20    | PC4_ADS | R/W | 0   | 端口 PC4 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 19    | PC3_ADS | R/W | 0   | 端口 PC3 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 18:14 | RSV     | -   | -   | 保留   |
| 13    | PB5_ADS | R/W | 0   | 端口 PB5 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 12    | PB4_ADS | R/W | 0   | 端口 PB4 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 11:4  | RSV     | -   | -   | 保留   |
| 3     | PA3_ADS | R/W | 0   | 端口 PA3 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 2     | PA2_ADS | R/W | 0   | 端口 PA2 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 1     | PA1_ADS | R/W | 0   | 端口 PA1 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |
| 0     | PA0_ADS | R/W | 0   | 端口 PA0 数模配置寄存器<br>0: 配置为数字接口<br>1: 配置为模拟接口 |

### 3.6.14 端口驱动能力配置寄存器 PAD\_DR (偏移: 044h)

| 比特    | 名称     | 属性  | 复位值 | 描述                                       |
|-------|--------|-----|-----|--|
| 31    | PD7_DR | R/W | 0   | 端口 PD7 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 30    | PD6_DR | R/W | 0   | 端口 PD6 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 29    | PD5_DR | R/W | 0   | 端口 PD5 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 28    | PD4_DR | R/W | 0   | 端口 PD4 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 27    | PD3_DR | R/W | 0   | 端口 PD3 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 26    | PD2_DR | R/W | 0   | 端口 PD2 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 25    | PD1_DR | R/W | 0   | 端口 PD1 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 24    | RSV    | -   | -   | 保留                                       |
| 23    | PC7_DR | R/W | 0   | 端口 PC7 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 22    | PC6_DR | R/W | 0   | 端口 PC6 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 21    | PC5_DR | R/W | 0   | 端口 PC5 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 20    | PC4_DR | R/W | 0   | 端口 PC4 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 19    | PC3_DR | R/W | 0   | 端口 PC3 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 18    | PC2_DR | R/W | 0   | 端口 PC2 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 17    | PC1_DR | R/W | 0   | 端口 PC1 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 16:14 | RSV    | -   | -   | 保留                                       |

| 比特   | 名称     | 属性  | 复位值 | 描述                                       |
|------|--------|-----|-----|--|
| 13   | PB5_DR | R/W | 0   | 端口 PB5 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 12   | PB4_DR | R/W | 0   | 端口 PB4 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 11:5 | RSV    | -   | -   | 保留                                       |
| 4    | PA4_DR | R/W | 0   | 端口 PA4 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 3    | PA3_DR | R/W | 0   | 端口 PA3 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 2    | PA2_DR | R/W | 0   | 端口 PA2 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 1    | PA1_DR | R/W | 0   | 端口 PA1 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |
| 0    | PA0_DR | R/W | 0   | 端口 PA0 驱动能力配置寄存器<br>0: 高驱动能力<br>1: 低驱动能力 |

### 3.6.15 端口上拉配置寄存器 PAD\_PU (偏移: 048h)

| 比特 | 名称     | 属性  | 复位值 | 描述                               |
|----|--------|-----|-----|----------------------------------|
| 31 | PD7_PU | R/W | 0   | 端口 PD7 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 30 | PD6_PU | R/W | 0   | 端口 PD6 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 29 | PD5_PU | R/W | 0   | 端口 PD5 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 28 | PD4_PU | R/W | 0   | 端口 PD4 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 27 | PD3_PU | R/W | 0   | 端口 PD3 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 26 | PD2_PU | R/W | 0   | 端口 PD2 上拉配置寄存器<br>0: 禁止<br>1: 使能 |



| 比特    | 名称     | 属性  | 复位值 | 描述                               |
|-------|--------|-----|-----|----------------------------------|
| 25    | PD1_PU | R/W | 1   | 端口 PD1 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 24    | RSV    | -   | -   | 保留                               |
| 23    | PC7_PU | R/W | 1   | 端口 PC7 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 22    | PC6_PU | R/W | 0   | 端口 PC6 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 21    | PC5_PU | R/W | 0   | 端口 PC5 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 20    | PC4_PU | R/W | 0   | 端口 PC4 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 19    | PC3_PU | R/W | 0   | 端口 PC3 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 18    | PC2_PU | R/W | 0   | 端口 PC2 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 17    | PC1_PU | R/W | 0   | 端口 PC1 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 16:14 | RSV    | -   | -   | 保留                               |
| 13    | PB5_PU | R/W | 0   | 端口 PB5 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 12    | PB4_PU | R/W | 0   | 端口 PB4 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 11:5  | RSV    | -   | -   | 保留                               |
| 4     | PA4_PU | R/W | 0   | 端口 PA4 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 3     | PA3_PU | R/W | 0   | 端口 PA3 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 2     | PA2_PU | R/W | 0   | 端口 PA2 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 1     | PA1_PU | R/W | 0   | 端口 PA1 上拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 0     | PA0_PU | R/W | 1   | 端口 PA0 上拉配置寄存器<br>0: 禁止<br>1: 使能 |

### 3.6.16 端口下拉配置寄存器 PAD\_PD (偏移: 04Ch)

| 比特    | 名称     | 属性  | 复位值 | 描述                               |
|-------|--------|-----|-----|----------------------------------|
| 31    | PD7_PD | R/W | 0   | 端口 PD7 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 30    | PD6_PD | R/W | 0   | 端口 PD6 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 29    | PD5_PD | R/W | 0   | 端口 PD5 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 28    | PD4_PD | R/W | 0   | 端口 PD4 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 27    | PD3_PD | R/W | 0   | 端口 PD3 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 26    | PD2_PD | R/W | 0   | 端口 PD2 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 25    | PD1_PD | R/W | 0   | 端口 PD1 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 24    | RSV    | -   | -   | 保留                               |
| 23    | PC7_PD | R/W | 0   | 端口 PC7 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 22    | PC6_PD | R/W | 0   | 端口 PC6 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 21    | PC5_PD | R/W | 0   | 端口 PC5 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 20    | PC4_PD | R/W | 0   | 端口 PC4 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 19    | PC3_PD | R/W | 0   | 端口 PC3 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 18    | PC2_PD | R/W | 0   | 端口 PC2 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 17    | PC1_PD | R/W | 0   | 端口 PC1 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 16:14 | RSV    | -   | -   | 保留                               |

| 比特   | 名称     | 属性  | 复位值 | 描述                               |
|------|--------|-----|-----|----------------------------------|
| 13   | PB5_PD | R/W | 0   | 端口 PB5 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 12   | PB4_PD | R/W | 0   | 端口 PB4 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 11:5 | RSV    | -   | -   | 保留                               |
| 4    | PA4_PD | R/W | 0   | 端口 PA4 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 3    | PA3_PD | R/W | 0   | 端口 PA3 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 2    | PA2_PD | R/W | 0   | 端口 PA2 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 1    | PA1_PD | R/W | 0   | 端口 PA1 下拉配置寄存器<br>0: 禁止<br>1: 使能 |
| 0    | PA0_PD | R/W | 0   | 端口 PA0 下拉配置寄存器<br>0: 禁止<br>1: 使能 |

### 3.6.17 端口开漏输出配置寄存器 PAD\_OD (偏移: 050h)

| 比特 | 名称     | 属性  | 复位值 | 描述                                 |
|----|--------|-----|-----|------------------------------------|
| 31 | PD7_OD | R/W | 0   | 端口 PD7 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 30 | PD6_OD | R/W | 0   | 端口 PD6 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 29 | PD5_OD | R/W | 0   | 端口 PD5 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 28 | PD4_OD | R/W | 0   | 端口 PD4 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 27 | PD3_OD | R/W | 0   | 端口 PD3 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 26 | PD2_OD | R/W | 0   | 端口 PD2 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |

| 比特    | 名称     | 属性  | 复位值 | 描述                                 |
|-------|--------|-----|-----|------------------------------------|
| 25    | PD1_OD | R/W | 0   | 端口 PD1 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 24    | RSV    | -   | -   | 保留                                 |
| 23    | PC7_OD | R/W | 0   | 端口 PC7 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 22    | PC6_OD | R/W | 0   | 端口 PC6 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 21    | PC5_OD | R/W | 0   | 端口 PC5 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 20    | PC4_OD | R/W | 0   | 端口 PC4 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 19    | PC3_OD | R/W | 0   | 端口 PC3 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 18    | PC2_OD | R/W | 0   | 端口 PC2 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 17    | PC1_OD | R/W | 0   | 端口 PC1 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 16:14 | RSV    | -   | -   | 保留                                 |
| 13    | PB5_OD | R/W | 0   | 端口 PB5 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 12    | PB4_OD | R/W | 0   | 端口 PB4 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 11:5  | RSV    | -   | -   | 保留                                 |
| 4     | PA4_OD | R/W | 0   | 端口 PA4 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 3     | PA3_OD | R/W | 0   | 端口 PA3 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 2     | PA2_OD | R/W | 0   | 端口 PA2 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 1     | PA1_OD | R/W | 0   | 端口 PA1 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |
| 0     | PA0_OD | R/W | 0   | 端口 PA0 开漏输出配置寄存器<br>0: 禁止<br>1: 使能 |

### 3.6.18 端口输入类型配置寄存器 PAD\_CS (偏移: 054h)

| 比特    | 名称     | 属性  | 复位值 | 描述  |
|-------|--------|-----|-----|---|
| 31    | PD7_CS | R/W | 1   | 端口 PD7 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 30    | PD6_CS | R/W | 1   | 端口 PD6 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 29    | PD5_CS | R/W | 1   | 端口 PD5 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 28    | PD4_CS | R/W | 1   | 端口 PD4 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 27    | PD3_CS | R/W | 1   | 端口 PD3 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 26    | PD2_CS | R/W | 1   | 端口 PD2 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 25    | PD1_CS | R/W | 1   | 端口 PD1 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 24    | RSV    | -   | -   | 保留  |
| 23    | PC7_CS | R/W | 1   | 端口 PC7 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 22    | PC6_CS | R/W | 1   | 端口 PC6 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 21    | PC5_CS | R/W | 1   | 端口 PC5 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 20    | PC4_CS | R/W | 1   | 端口 PC4 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 19    | PC3_CS | R/W | 1   | 端口 PC3 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 18    | PC2_CS | R/W | 1   | 端口 PC2 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 17    | PC1_CS | R/W | 1   | 端口 PC1 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 16:14 | RSV    | -   | -   | 保留  |

| 比特   | 名称     | 属性  | 复位值 | 描述  |
|------|--------|-----|-----|---|
| 13   | PB5_CS | R/W | 1   | 端口 PB5 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 12   | PB4_CS | R/W | 1   | 端口 PB4 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 11:5 | RSV    | -   | -   | 保留  |
| 4    | PA4_CS | R/W | 1   | 端口 PA4 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 3    | PA3_CS | R/W | 1   | 端口 PA3 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 2    | PA2_CS | R/W | 1   | 端口 PA2 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 1    | PA1_CS | R/W | 1   | 端口 PA1 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |
| 0    | PA0_CS | R/W | 1   | 端口 PA0 输入类型配置寄存器<br>0: Schmitt input buffer<br>1: CMOS input buffer |

### 3.6.19 端口输入配置寄存器 PAD\_IE (偏移: 058h)

| 比特 | 名称     | 属性  | 复位值 | 描述                                   |
|----|--------|-----|-----|--------------------------------------|
| 31 | PD7_IE | R/W | 1   | 端口 PD7 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 30 | PD6_IE | R/W | 1   | 端口 PD6 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 29 | PD5_IE | R/W | 1   | 端口 PD5 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 28 | PD4_IE | R/W | 1   | 端口 PD4 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 27 | PD3_IE | R/W | 1   | 端口 PD3 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 26 | PD2_IE | R/W | 1   | 端口 PD2 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |

| 比特    | 名称     | 属性  | 复位值 | 描述                                   |
|-------|--------|-----|-----|--------------------------------------|
| 25    | PD1_IE | R/W | 1   | 端口 PD1 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 24    | RSV    | -   | -   | 保留                                   |
| 23    | PC7_IE | R/W | 1   | 端口 PC7 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 22    | PC6_IE | R/W | 1   | 端口 PC6 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 21    | PC5_IE | R/W | 1   | 端口 PC5 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 20    | PC4_IE | R/W | 1   | 端口 PC4 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 19    | PC3_IE | R/W | 1   | 端口 PC3 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 18    | PC2_IE | R/W | 1   | 端口 PC2 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 17    | PC1_IE | R/W | 0   | 端口 PC1 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 16:14 | RSV    | -   | -   | 保留                                   |
| 13    | PB5_IE | R/W | 1   | 端口 PB5 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 12    | PB4_IE | R/W | 1   | 端口 PB4 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 11:5  | RSV    | -   | -   | 保留                                   |
| 4     | PA4_IE | R/W | 0   | 端口 PA4 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 3     | PA3_IE | R/W | 1   | 端口 PA3 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 2     | PA2_IE | R/W | 1   | 端口 PA2 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 1     | PA1_IE | R/W | 1   | 端口 PA1 输入配置寄存器<br>0: 输入禁止<br>1: 输入使能 |
| 0     | RSV    | -   | -   | 保留                                   |

### 3.6.20 IO 控制保护寄存器 IOCTRL\_PROTECT (偏移: 05Ch)

| 比特   | 名称             | 属性 | 复位值 | 描述  |
|------|----------------|----|-----|---|
| 31:0 | IOCTRL_PROTECT | W  | 0   | IO 寄存器 PA_SEL/ PB_SEL/ PC_SEL/ PD_SEL/ PAD_ADS/ PAD_DR/ PAD_PU/ PAD_PD/ PAD_OD/ PAD_CS/ PAD_IE 保护的寄存器。给此寄存器写 0xA5A5_5A5A, 启动这些 IO 寄存器的写使能。配置完 IO 寄存器后, 它们的写使能不会自动关闭。可以给此寄存器写其它值, 来关闭 IO 寄存器的写使能。<br>0: 写保护未使能<br>1: 写保护已经使能 |

### 3.6.21 唤醒状态清除寄存器 IOWACK\_INTR\_CLR (偏移: 0x060h)

| 比特   | 名称             | 属性 | 复位值 | 描述   |
|------|----------------|----|-----|--|
| 31:4 | RSV            | -  | -   | 保留   |
| 3    | GPIOD_WACK_CLR | W  | 0   | 当 GPIO_CFG 寄存器中的 IESEL 位为 1, 且唤醒源为 GPIOD 时, 在系统唤醒后, 此位写 1, 清除 GPIO 唤醒中断。<br>1: 系统唤醒后, 清除 GPIOD 的唤醒中断;<br>0: 无操作。 |
| 2    | GPIOC_WACK_CLR | W  | 0   | 当 GPIO_CFG 寄存器中的 IESEL 位为 1, 且唤醒源为 GPIOC 时, 在系统唤醒后, 此位写 1, 清除 GPIO 唤醒中断。<br>1: 系统唤醒后, 清除 GPIOC 的唤醒中断;<br>0: 无操作。 |
| 1    | GPIOB_WACK_CLR | W  | 0   | 当 GPIO_CFG 寄存器中的 IESEL 位为 1, 且唤醒源为 GPIOB 时, 在系统唤醒后, 此位写 1, 清除 GPIO 唤醒中断。<br>1: 系统唤醒后, 清除 GPIOB 的唤醒中断;<br>0: 无操作。 |
| 0    | GPIOA_WACK_CLR | W  | 0   | 当 GPIO_CFG 寄存器中的 IESEL 位为 1, 且唤醒源为 GPIOA 时, 在系统唤醒后, 此位写 1, 清除 GPIO 唤醒中断。<br>1: 系统唤醒后, 清除 GPIOA 的唤醒中断;<br>0: 无操作。 |



### 3.6.22 LVD 控制寄存器 LVD\_CFG (偏移: 064h)

| 比特    | 名称           | 属性  | 复位值  | 描述   |
|-------|--------------|-----|------|--|
| 31:16 | LVD_FILTER   | R/W | 0xC8 | LVD 滤波配置位。<br>0: 对 LVD 滤除 1 个系统时钟的毛刺<br>1: 对 LVD 滤除 2 个系统时钟的毛刺<br>....<br>65535: 对 LVD 滤除 65536 个系统时钟的毛刺。  |
| 15:10 | RSV          | -   | -    | 保留   |
| 9     | LVD_INTR_EN  | R/W | 0    | LVD 中断使能控制位。<br>0: 不使能 LVD 中断<br>1: 使能 LVD 中断  |
| 8     | LVD_RESET_EN | R/W | 0    | LVD 复位使能控制位。<br>0: 不使能 LVD 复位<br>1: 使能 LVD 复位  |
| 7:4   | LVD_VADJ     | R/W | 0    | LVD 检测阈值的配置信号, 默认值为 0000, 阈值 1.8V。<br>0000: 1.8V<br>0001: 1.9V<br>0010: 2.0V<br>0011: 2.1V<br>0100: 2.2V<br>0101: 2.3V<br>0110: 2.4V<br>0111: 2.5V<br>1000: 2.6V<br>1001: 2.7V<br>1010: 2.8V<br>1011: 2.9V<br>1100: 3.0V<br>1101: 3.1V<br>1110: 3.2V<br>1111: 3.3V |
| 3     | RSV          | -   | -    | 保留   |
| 2:1   | LVD_CHSEL    | R/W | 0    | LVD 通道选择寄存器<br>00: VDDH<br>01: LVD_VIN1<br>10: LVD_VIN2<br>11: LVD_VIN3  |
| 0     | LVD_EN       | R/W | 0    | LVD 模块使能寄存器<br>0: 禁止<br>1: 使能  |

### 3.6.23 端口辅助功能寄存器 GPIO\_CFG (偏移: 070h)

| 比特   | 名称    | 属性  | 复位值 | 描述  |
|------|-------|-----|-----|---|
| 31:1 | RSV   | -   | -   | 保留  |
| 0    | IESEL | R/W | 0   | 端口中断模式选择寄存器<br>1: DEEPSLEEP/STOP 模式<br>0: ACTIVE/SLEEP 模式 |

注: 当系统处于 ACTIVE/SLEEP 模式下时, 系统时钟不会被关闭, 可以将 IESEL 设置为 0, 此时触发端口中断的外部信号源将经过系统时钟同步之后产生中断信号, 可以滤除外部信号源的毛刺。当系统即将进入 DEEPSLEEP/STOP 模式下时, 系统时钟将被关闭, 可以将 IESEL 设置为 1, 此时触发端口中断的外部信号源直接产生中断信号, 不能滤除外部信号源的毛刺。

### 3.6.24 外部复位端口选择寄存器 EXTRST\_SEL (偏移: 074h)

| 比特   | 名称         | 属性  | 复位值 | 描述   |
|------|------------|-----|-----|--|
| 31:1 | RSV        | -   | -   | 保留   |
| 0    | RESETN_SEL | R/W | 0   | 外部复位端口选择寄存器。只有该寄存器的【31:16】的高 16 位写 0xA5A5 时才能写这个 bit。<br>1: 外部复位信号无效。该管脚可以作为 GPIO 使用<br>0: 外部复位信号有效。 |

### 3.6.25 停止模式选择寄存器 STOPMODE\_SEL (偏移: 078h)

| 比特   | 名称           | 属性  | 复位值 | 描述  |
|------|--------------|-----|-----|---|
| 31:1 | RSV          | -   | -   | 保留  |
| 0    | STOPMODE_SEL | R/W | 0   | 停止模式选择寄存器。只有该寄存器的【31:16】的高 16 位写 0xA5A5 时才能写这个 bit。<br>1: STOP mode 停止模式有效<br>0: STOP mode 停止模式无效 |

### 3.6.26 REMAP 寄存器 REMAP\_ADDR (偏移: 07Ch)

| 比特   | 名称    | 属性 | 复位值 | 描述  |
|------|-------|----|-----|---|
| 31:3 | RSV   | -  | -   | 保留  |
| 2    | REMAP | R  | 0   | Eflash 地址 remap 标志位:<br>0: Eflash 地址没有重映射, Bootloader 启动;<br>1: Eflash 地址进行重映射, Main 区启动。 |

| 比特 | 名称          | 属性 | 复位值 | 描述  |
|----|-------------|----|-----|---|
| 1  | REMAP_IM    | W  | 1   | 立即进行 REMAP 操作，但是系统不发生复位。<br>0: 立即进行 eflash 地址重映射；<br>1: eflash 地址不进行重映射                                     |
| 0  | SOFT_RESETN | W  | 1   | 软复位，当此位写 0 时，会产生一次软件复位，复位 CPU 及 AHB/APB 总线上的所有 IP。并且，eflash 地址重新映射（remap 为 1）<br>0: 系统进行软复位；<br>1: 系统不进行软复位 |

注：寄存器 REMAP 位复位信号为 SYSTEM\_RESETN，不会随 PRESETN 复位而复位掉。

### 3.6.27 中断向量地址重映射寄存器 VECTOR\_OFFSET (偏移：080h)

| 比特    | 名称              | 属性  | 复位值 | 描述  |
|-------|-----------------|-----|-----|---|
| 31:10 | VECTOROFFSET    | R/W | 0   | 中断向量重映射功能使能后，中断向量的基地址是本寄存器中的值                       |
| 9:1   | RSV             | -   | -   | 保留  |
| 0     | VECTOROFFSET_EN | R/W | 0   | 中断向量重映射功能使能：<br>0: 不使能中断向量重映射功能；<br>1: 使能中断向量重映射功能。 |

### 3.6.28 随机数控制寄存器 HRNG\_CR (偏移：084h)

| 比特   | 名称      | 属性  | 复位值 | 描述                     |
|------|---------|-----|-----|------------------------|
| 31:1 | RSV     | -   | -   | 保留                     |
| 0    | HRNG_EN | R/W | 0   | 1: 随机数使能；<br>0: 随机数禁止。 |

### 3.6.29 随机数种子寄存器 HRNG\_SEED (偏移：088h)

| 比特   | 名称        | 属性  | 复位值 | 描述       |
|------|-----------|-----|-----|----------|
| 31:0 | HRNG_SEED | R/W | 0   | 随机数种子寄存器 |

### 3.6.30 随机数数据寄存器 HRNG\_DATA (偏移：08Ch)

| 比特   | 名称        | 属性 | 复位值        | 描述                    |
|------|-----------|----|------------|-----------------------|
| 31:0 | HRNG_DATA | R  | 32'hFF00FF | 随机数寄存器。读取此寄存器，读出随机数值。 |

### 3.6.31 蜂鸣器控制寄存器 BUZZERCR (偏移: 090h)

| 比特    | 名称         | 属性  | 复位值  | 描述  |
|-------|------------|-----|------|---|
| 31:18 | RSV        | -   | -    | 保留  |
| 17    | BUZZER_EN  | R/W | 1    | 蜂鸣器时钟输出使能<br>0: 不使能, 信号为 BUZZER_POL<br>1: 使能        |
| 16    | BUZZER_POL | R/W | 0    | 蜂鸣器时钟极性选择<br>0: 原极性 (停止的时候为 0)<br>1: 反极性 (停止的时候为 1) |
| 15:0  | BUZZER_DIV | R/W | 0x3B | 蜂鸣器时钟分频值: 分频数为寄存器值+1                                |

### 3.6.32 保留寄存器 0 Reserved0 (偏移: 0A0h) (保留)

### 3.6.33 保留寄存器 1 Reserved1 (偏移: 0A4h) (保留)

## 4 NVIC

### 4.1 概述

内嵌向量中断控制器(NVIC) 是 Cortex-M0+的一个重要组成部分。它与 CPU 处理器内核紧密耦合，实现低中断延迟以及对新到中断的有效处理，外部中断信号连接到 NVIC，NVIC 将对这些中断进行优先级排序。

Cortex-M0+处理器内置了嵌套向量中断控制器（NVIC），可支持最多 32 个中断请求（IRQ）输入：有 4 个中断优先级，可处理复杂逻辑，能进行实时控制和中断处理。

所有的 NVIC 寄存器只能采用字传输。任何试图读/写半字或字节的结果都是不可预知的。

NVIC 寄存器都是小端格式。访问处理器要正确处理处理器的大小端配置。

(关于 NVIC 更详细的内容可查看 Cortex-M0+系列内核的相关官方文档)

### 4.2 主要特性

- 32 个外部中断，每个中断具有 4 级优先级
- 专用的不可屏蔽中断（NMI）
- 同时支持电平和脉冲中断触发
- 中断唤醒控制器，支持极低功耗休眠模式

### 4.3 中断源

表 4-1：中断源

| 中断号  | 中断源     | 备注 |
|------|---------|----|
| [0]  | GPIO_PA |    |
| [1]  | GPIO_PB |    |
| [2]  | GPIO_PC |    |
| [3]  | GPIO_PD |    |
| [4]  | DMA     |    |
| [5]  | -       |    |
| [6]  | UART0   |    |
| [7]  | LPUART  |    |
| [8]  | UART1   |    |
| [9]  | I2C     |    |
| [10] | SPI0    |    |
| [11] | SPI1    |    |
| [12] | -       |    |
| [13] | -       |    |
| [14] | GTimer0 |    |
| [15] | GTimer1 |    |
| [16] | GTimer2 |    |

| 中断号  | 中断源             | 备注 |
|------|-----------------|----|
| [17] | GTimer3         |    |
| [18] | LPTimer         |    |
| [19] | -               |    |
| [20] | -               |    |
| [21] | -               |    |
| [22] | WDT             |    |
| [23] | RTC             |    |
| [24] | ADC             |    |
| [25] | -               |    |
| [26] | -               |    |
| [27] | -               |    |
| [28] | LVD             |    |
| [29] | -               |    |
| [30] | FLASH interrupt |    |
| [31] | -               |    |
| NMI  | -               | 保留 |

## 5 UART0

### 5.1 概述

UART0 串口模块，带有 8 比特 4 级的接收 FIFO。

### 5.2 主要特性

- 提供标准的异步通讯位（起始位、奇偶位和停止位）
  - 生成 1 位起始位
  - 支持 8bit 的数据位宽
  - 生成 1 位校验位(可设置奇校验或偶校验)，或无校验位
  - 生成 1 位停止位
  - 字节从低位到高位依次传输
- 8 比特 4 级的接收 FIFO，无发送 FIFO
- 可编程波特率(波特率可以根据参数 F/D 调整)，2\*8bits 波特率参数寄存器
- 支持数据通讯及错误处理中断
  - 状态位的访问可采用查询或者中断两种方式
  - FIFO 非空、半满、全满、溢出标志
  - 奇偶校验错误标志
- 具有起始位有效性检测功能
- 可支持 9600bps、19200bps、115200bps 等常见波特率的传输

### 5.3 寄存器描述

UART0 寄存器基地址：0x40000000

表 5-1：UART0 寄存器列表

| 偏置   | 名称       | 描述         |
|------|----------|------------|
| 0x00 | UARTISR  | 中断状态寄存器    |
| 0x04 | UARTIER  | 中断使能寄存器    |
| 0x08 | UARTCR   | 控制寄存器      |
| 0x0C | UARTTDR  | 发送数据寄存器    |
| 0x0C | UARTRDR  | 接收数据寄存器    |
| 0x10 | UARTBRPL | 波特率参数低位寄存器 |
| 0x14 | UARTBRPH | 波特率参数高位寄存器 |

### 5.3.1 中断状态寄存器 UARTISR (偏移: 00h)

| 比特   | 名称      | 属性  | 复位值 | 描述  |
|------|---------|-----|-----|---|
| 31:6 | RSV     | -   | -   | 保留  |
| 5    | FIFO_NE | R/W | 0   | FIFO 非空标志:<br>FIFO_NE =0 则 FIFO 空;<br>FIFO_NE =1 则 FIFO 非空。<br>当 FIFO 读空时, 此位自动清 0。软件也可以清除此位, 写 0 清除。             |
| 4    | FIFO_HF | R/W | 0   | FIFO 半满标志:<br>FIFO_HF =0 则 FIFO 非半满;<br>FIFO_HF =1 则 FIFO 半满;<br>当 FIFO 中数据读空时, 此位自动清 0。软件也可以清除此位, 写 0 清除。        |
| 3    | FIFO_FU | R/W | 0   | FIFO 全满标志:<br>FIFO_FU =0 则 FIFO 非全满;<br>FIFO_FU =1 则 FIFO 全满;<br>当读取 FIFO 中数据, 此位自动清 0。软件也可以清除此位, 写 0 清除。         |
| 2    | FIFO_OV | R   | 0   | Rx-FIFO 接收溢出错误:<br>FIFO_OV =0 没有接收溢出错误发生;<br>FIFO_OV =1 发生了接收溢出错误;<br>软件清除此位, 写 0 清除。                             |
| 1    | TXEND   | R/W | 0   | UART 发送完成标志:<br>TXEND =0 表示发送没有完成;<br>TXEND =1 发送完成;<br>此位硬件置 1, 软件清除, 写 0 清除。                                    |
| 0    | TRE     | R/W | 0   | UART 发送/接收奇偶校验错误标示:<br>TRE =0 则 UART 发送/接收完成时无奇偶校验错误;<br>TRE =1 则 UART 发送/接收完成时有奇偶校验错误。<br>此位硬件置 1, 软件清除, 写 0 清除。 |

### 5.3.2 中断使能寄存器 UARTIER (偏移: 04h)

| 比特   | 名称        | 属性  | 复位值 | 描述  |
|------|-----------|-----|-----|---|
| 31:6 | RSV       | -   | -   | 保留  |
| 5    | FIFO_EN   | R/W | 0   | FIFO 非空中断使能: 当 FIFO_EN =0 时禁止;<br>当 FIFO_EN =1 使能。          |
| 4    | FIFO_HFEn | R/W | 0   | FIFO 半满中断使能: 当 FIFO_HFEn =0 时禁止;<br>当 FIFO_HFEn =1 使能。      |
| 3    | FIFO_FUEn | R/W | 0   | FIFO 全满中断使能: 当 FIFO_FUEn =0 时禁止;<br>当 FIFO_FUEn =1 使能。      |
| 2    | FIFO_OVEn | R/W | 0   | Rx-FIFO 接收溢出中断使能: 当 FIFO_OVEn =0 时禁止;<br>当 FIFO_OVEn =1 使能。 |
| 1    | TXENDEn   | R/W | 0   | Uart 发送完成中断使能: 当 TXENDEn =0 时禁止;<br>当 TXENDEn =1 使能。        |
| 0    | TREEN     | R/W | 0   | Uart 发送/接收奇偶校验错误中断使能: 当 TREEN =0 时禁止;<br>当 TREEN =1 使能。     |



### 5.3.3 控制寄存器 UARTCR (偏移: 08h)

| 比特   | 名称      | 属性  | 复位值 | 描述   |
|------|---------|-----|-----|--|
| 31:5 | RSV     | -   | -   | 保留   |
| 4    | UART_LB | R/W | 0   | Uart 自测模式使能控制:<br>UART_LB =0, 不使能<br>UART_LB =1, 使能                  |
| 3    | UART_PD | R/W | 0   | 奇偶校验使能控制:<br>UART_PD =0, 有奇偶校验<br>UART_PD =1, 没有奇偶校验                 |
| 2    | FLUSH   | R/W | 0   | 清除 uart 接收 FIFO 中的数据 and 指针<br>FLUSH=0, 不清除<br>FLUSH=1, 清除           |
| 1    | TRS     | R/W | 0   | UART 发送数据标志:<br>TRS =0 发送数据不使能<br>TRS =1 发送数据使能                      |
| 0    | ODD_EN  | R/W | 0   | 奇偶校验方式选择:<br>ODD_EN =0, 偶校验 Even Parity<br>ODD_EN =1, 奇校验 Odd Parity |

### 5.3.4 发送数据寄存器 UARTTDR (偏移: 0Ch)

| 比特   | 名称       | 属性 | 复位值 | 描述       |
|------|----------|----|-----|----------|
| 31:8 | RSV      | -  | -   | 保留       |
| 7:0  | UARTDATA | W  | 00  | 存放待发送的数据 |

### 5.3.5 接收数据寄存器 UARTRDR (偏移: 0Ch)

| 比特   | 名称       | 属性 | 复位值 | 描述        |
|------|----------|----|-----|-----------|
| 31:8 | RSV      | -  | -   | 保留        |
| 7:0  | UARTDATA | R  | 00  | 存放待接收到的数据 |

### 5.3.6 波特率参数低位寄存器 UARTBRPL (偏移: 10h)

| 比特   | 名称  | 属性 | 复位值 | 描述 |
|------|-----|----|-----|----|
| 31:8 | RSV | -  | -   | 保留 |

| 比特  | 名称       | 属性  | 复位值 | 描述  |
|-----|----------|-----|-----|---|
| 7:0 | UARTBRPL | R/W | 74  | 波特率参数寄存器 UARTBPRH、UARTBPRL 构成 16 位分频器。<br>例如：系统时钟为 40MHz，为获得 9600 波特率，则 $UARTBPR=40 \times 1000000 \div 9600 = 1046H$ ，<br>即 $UARTBPRH=10H$ ， $UARTBPRL=46H$ 。<br>例如：系统时钟为 40MHz，为获得 19200 波特率，则 $UARTBPR=0823H$ ，即 $UARTBPRH=08H$ ， $UARTBPRL=23H$ 。 |

### 5.3.7 波特率参数高位寄存器 UARTBRPH (偏移：14h)

| 比特   | 名称       | 属性  | 复位值 | 描述  |
|------|----------|-----|-----|---|
| 31:8 | RSV      | -   | -   | 保留  |
| 7:0  | UARTBRPH | R/W | 01  | 波特率参数寄存器 UARTBPRH、UARTBPRL 构成 16 位分频器。<br>例如：系统时钟为 40MHz，为获得 9600 波特率，则 $UARTBPR=40 \times 1000000 \div 9600 = 1046H$ ，<br>即 $UARTBPRH=10H$ ， $UARTBPRL=46H$ 。<br>例如：系统时钟为 40MHz，为获得 19200 波特率，则 $UARTBPR=0823H$ ，即 $UARTBPRH=08H$ ， $UARTBPRL=23H$ 。 |

## 5.4 使用流程

### 5.4.1 串口的发送和接收

1. 配置系统配置寄存器的串口模块时钟。
2. 配置系统配置寄存器的串口模块复位使能。
3. 配置系统配置寄存器的串口引脚复用功能。
4. 配置串口中断。
5. 配置串口中断使能寄存器（是否使用中断）。
6. 配置串口控制寄存器（清除接收 FIFO 中的数据和指针）。
7. 配置串口控制寄存器（奇偶校验位等）。
8. 配置波特率。
9. 使能串口。

## 5.4.2 串口初始化

1. 清除 UARTISR 寄存器，写 0 清除。
2. 配置 UARTIER，中断使能寄存器，是否产生相应的中断脉冲。
3. 设置 UARTTCR.FLUSH，清除 FIFO 中数据及 FIFO 指针。
4. 清除 UARTTCR 寄存器，写 0 清除。
5. 配置 UARTBPRL[7:0]和 UARTBPRH[7:0]。

## 5.4.3 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，奇偶校验类型、中断使能。
2. 设置 UARTTCR.TRS=1。
3. 写入第一个字节数据到 UARTTDR。
4. 查询发送完成标志 UARTISR.TXEND，如果 TXEND=1 表示当前数据发送完成；软件清除此位（写 0 清除）。
5. 如果发送出错：UART 产生中断或者查询 SCCISR 寄存器标志，判断错误类型，执行相应的错误处理，处理完之后软件清除标志位。
6. 可以继续写入下一个字节到 UARTTDR。

## 5.4.4 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、奇偶校验类型、中断使能。
2. 接收数据，查询 UARTISR 标志位或者等待中断，FIFO\_NE（即接收数据 FIFO 非空），或者 FIFO\_HF（即接收数据 FIFO 半满），或者 FIFO\_FU（即接收数据 FIFO 全满）；查询到相应标志位则读取 UARTRDR 中的数据，FIFO 相应的标志位自动清除。
3. 接收错误处理：等待中断或者查询 UARTISR 寄存器标志位，判断错误类型，执行相应的错误处理，处理完之后软件清除标志位。
4. 继续接收数据。

## 6 UART1

### 6.1 概述

UART1 串口模块，带有 16 字节的 FIFO，可小数分频。

### 6.2 主要特性

- 16 字节的硬件 FIFO
- 波特率支持整数和小数分频
- 支持 CTS, RTS 流控制
- 错误起始位侦测
- 帧中断检测
- 可编程位宽，奇偶校验，停止位个数
- 支持 DMA 传输方式

### 6.3 寄存器描述

UART1 寄存器基地址：0x40003000

表 6-1: UART1 寄存器列表

| 偏置   | 名称        | 描述              |
|------|-----------|-----------------|
| 0x00 | UART1_RBR | 接收缓冲寄存器         |
| 0x00 | UART1_THR | 发送缓冲寄存器         |
| 0x00 | UART1_DLL | 波特率分频低位寄存器      |
| 0x04 | UART1_DLH | 波特率分频高位寄存器      |
| 0x04 | UART1_IER | 中断使能寄存器         |
| 0x08 | UART1_IIR | 中断状态寄存器         |
| 0x08 | UART1_FCR | FIFO 控制寄存器      |
| 0x0C | UART1_LCR | LINE 控制寄存器      |
| 0x10 | UART1_MCR | 流控制寄存器          |
| 0x14 | UART1_LSR | LINE 中断状态寄存器    |
| 0x18 | UART1_MSR | 流状态寄存器          |
| 0x7C | UART1_USR | 状态寄存器           |
| 0x80 | UART1_TFL | 发送 FIFO 数据个数寄存器 |
| 0x84 | UART1_RFL | 接收 FIFO 数据个数寄存器 |
| 0xC0 | UART1_DLF | 小数分频寄存器         |
| 0xC4 | UART1_RAR | 接收地址匹配寄存器       |
| 0xC8 | UART1_TAR | 发送地址匹配寄存器       |

### 6.3.1 接收缓冲寄存器 UART1\_RBR (偏移: 00h)

| 比特   | 名称  | 属性 | 复位值 | 描述  |
|------|-----|----|-----|---|
| 31:9 | RSV | -  | -   | 保留  |
| 8:0  | RBR | R  | 0   | 接收数据寄存器。此字段为接收 FIFO 入口, 仅当 UART1_LCR 的 DLAB 位为 0 时, 此字段才可以访问。 |

### 6.3.2 发送缓冲寄存器 UART1\_THR (偏移: 00h)

| 比特   | 名称  | 属性 | 复位值 | 描述  |
|------|-----|----|-----|---|
| 31:9 | RSV | -  | -   | 保留  |
| 8:0  | THR | W  | 0   | 发送数据寄存器。此字段为发送 FIFO 入口, 仅当 UART1_LCR 的 DLAB 位为 0 时, 此字段才可以访问。 |

### 6.3.3 波特率分频低位寄存器 UART1\_DLL (偏移: 00h)

| 比特   | 名称  | 属性  | 复位值 | 描述   |
|------|-----|-----|-----|--|
| 31:8 | RSV | -   | -   | 保留   |
| 7:0  | DLL | R/W | 0   | 波特率配置寄存器低位。仅当 UART1_LCR 的 DLAB 位为 1 时, 此字段才可以访问。<br>波特率整数部分计算公式:<br>$\text{baud rate} = \text{fclk} / (16 * \{\text{DLH}, \text{DLL}\})$ |

### 6.3.4 波特率分频高位寄存器 UART1\_DLH (偏移: 04h)

| 比特   | 名称  | 属性  | 复位值 | 描述   |
|------|-----|-----|-----|--|
| 31:8 | RSV | -   | -   | 保留   |
| 7:0  | DLH | R/W | 0   | 波特率配置寄存器高位。仅当 UART1_LCR 的 DLAB 位为 1 时, 此字段才可以访问。<br>波特率整数部分计算公式:<br>$\text{baud rate} = \text{fclk} / (16 * \{\text{DLH}, \text{DLL}\})$ |

### 6.3.5 中断使能寄存器 UART1\_IER (偏移: 04h)

| 比特   | 名称  | 属性 | 复位值 | 描述 |
|------|-----|----|-----|----|
| 31:8 | RSV | -  | -   | 保留 |

| 比特  | 名称    | 属性  | 复位值 | 描述   |
|-----|-------|-----|-----|--|
| 7   | PTIME | R/W | 0   | THRE 中断使能，仅当 UART1_LCR 的 DLAB 位为 0 时，此字段才可以访问。<br>1: 使能 THRE 中断<br>0: 禁止 THRE 中断           |
| 6:3 | RSV   | -   | -   | 保留   |
| 2   | ELSI  | R/W | 0   | LINE 中断使能，仅当 UART1_LCR 的 DLAB 位为 0 时，此字段才可以访问。<br>1: 使能 LINE 中断<br>0: 禁止 LINE 中断           |
| 1   | ETBEI | R/W | 0   | 发送 FIFO 空中断使能，仅当 UART1_LCR 的 DLAB 位为 0 时，此字段才可以访问。<br>1: 使能发送 FIFO 空中断<br>0: 禁止发送 FIFO 空中断 |
| 0   | ERBFI | R/W | 0   | 接收数据中断使能，仅当 UART1_LCR 的 DLAB 位为 0 时，此字段才可以访问。<br>1: 使能接收 FIFO 非空中断<br>0: 禁止接收 FIFO 非空中断    |

### 6.3.6 中断状态寄存器 UART1\_IIR (偏移: 08h)

| 比特   | 名称     | 属性 | 复位值  | 描述   |
|------|--------|----|------|--|
| 31:4 | RSV    | -  | -    | 保留   |
| 7:6  | FIFOSE | R  | 0    | FIFO 使能标志。<br>11: FIFO 使能<br>00: FIFO 禁止   |
| 5:4  | RSV    | -  | -    | 保留   |
| 3:0  | IID    | R  | 0001 | 状态 ID:<br>0000: CTS/RTS 中断状态<br>0001: 无中断<br>0010: 发送 FIFO 空<br>0100: 接收 FIFO 非空<br>0110: LINE 中断状态<br>0111: Busy 状态<br>1100: TimeOut 状态, 当使能 FIFO 和接收 FIFO 非空中断后, 如果在接收 FIFO 中存在至少 1 个数据, 在 4 个 UART 帧内, CPU 如果未读 FIFO, 则此字段会进入 TimeOut 中断状态。<br>其它: 保留 |

### 6.3.7 FIFO 控制寄存器 UART1\_FCR (偏移: 08h)

| 比特   | 名称     | 属性 | 复位值 | 描述   |
|------|--------|----|-----|--|
| 31:8 | RSV    | -  | -   | 保留   |
| 7:6  | RT     | W  | 0   | 接收 FIFO 非空中断设置, 当 FIFO 中数据大于或等于此设置对应的 FIFO 状态时, 接收 FIFO 非空中断置位:<br>00: 1 帧数据<br>01: 4 帧数据<br>10: 8 帧数据<br>11: 14 帧数据 |
| 5:4  | TET    | W  | 0   | 发送 FIFO 空中断设置, 当 FIFO 中数据少于或等于此设置对应的 FIFO 状态时, 发送 FIFO 空中断置位:<br>00: FIFO 空<br>01: 2 帧数据<br>10: 4 帧数据<br>11: 8 帧数据   |
| 3    | RSV    | -  | -   | 保留   |
| 2    | XFIFOR | W  | 0   | 发送 FIFO 复位位:<br>1: 复位发送 FIFO<br>0: 不复位发送 FIFO  |
| 1    | RFIFOR | W  | 0   | 接收 FIFO 复位位:<br>1: 复位接收 FIFO<br>0: 不复位接收 FIFO  |
| 0    | FIFOE  | W  | 0   | FIFO 使能位:<br>1: 使能 FIFO<br>0: 禁止 FIFO<br>改变此位的值将会同时复位接收和发送 FIFO。   |

### 6.3.8 LINE 控制寄存器 UART1\_LCR (偏移: 0Ch)

| 比特   | 名称   | 属性  | 复位值 | 描述  |
|------|------|-----|-----|---|
| 31:8 | RSV  | -   | -   | 保留  |
| 7    | DLAB | R/W | 0   | UART1_DLL 和 UART1_DLH 寄存器访问设置位。<br>1: UART1_DLL 可以通过偏移地址 0x0 访问, UART1_DLH 可以通过偏移地址 0x4 访问。<br>0: UART1_RBR/UART1_THR 可以通过偏移地址 0x0 访问, UART1_IER 可以通过偏移地址 0x4 访问。 |
| 6    | RSV  | -   | -   | 保留  |

| 比特  | 名称   | 属性  | 复位值 | 描述  |
|-----|------|-----|-----|---|
| 5   | SEPS | R/W | 0   | 奇偶校验位强制设置位，仅当 UART 处于空闲状态时可写：<br>1：当 PEN 为 1，EPS 为 1，奇偶校验位被传输并检查为逻辑 0；PEN 为 1，当 EPS 为 0 时，奇偶校验位被传输并检查为逻辑 1；当 PEN 为 0 时，发送和接收均无奇偶校验。<br>0：奇偶校验位强制设置功能禁止。 |
| 4   | EPS  | R/W | 0   | 奇偶校验选择位，仅当 UART 处于空闲状态时可写：<br>1：偶校验<br>0：奇校验  |
| 3   | PEN  | R/W | 0   | 奇偶校验位使能设置，仅当 UART 处于空闲状态时可写：<br>1：奇偶校验位使能<br>0：奇偶校验位禁止  |
| 2   | STOP | R/W | 0   | STOP 比特长度设置，仅当 UART 处于空闲状态时可写：<br>1：1.5 比特 STOP 位<br>0：1 比特 STOP 位  |
| 1:0 | DLS  | R/W | 0   | UART 帧数据长度设置位，仅当 UART 处于空闲状态时可写：<br>00：5 比特<br>01：6 比特<br>10：7 比特<br>11：8 比特  |

### 6.3.9 流控制寄存器 UART1\_MCR (偏移：10h)

| 比特   | 名称   | 属性  | 复位值 | 描述   |
|------|------|-----|-----|--|
| 31:6 | RSV  | -   | -   | 保留   |
| 5    | AFCE | R/W | 0   | 1：CTS/RTS 自动流控制使能<br>0：CTS/RTS 自动流控制禁止       |
| 4:2  | RSV  | -   | -   | 保留   |
| 1    | RTS  | R/W | 0   | RTS 接口软件控制位：<br>1：RTS 请求输出有效<br>0：RTS 请求输出无效 |
| 0    | RSV  | -   | -   | 保留   |

### 6.3.10 LINE 中断状态寄存器 UART1\_LSR (偏移：14h)

| 比特   | 名称  | 属性 | 复位值 | 描述 |
|------|-----|----|-----|----|
| 31:9 | RSV | -  | -   | 保留 |



| 比特 | 名称        | 属性 | 复位值 | 描述   |
|----|-----------|----|-----|--|
| 8  | ADDR_RCVD | R  | 0   | 9 比特数据模式下，接收到的数据是地址还是数据的标志：<br>1：接收的数据为地址信息<br>0：接收的数据为数据信息<br>读取此寄存器，清 0。   |
| 7  | RFE       | R  | 0   | 接收 FIFO 错误标志：<br>1：接收 FIFO 中数据至少有一个有奇偶校验错误或者 UART 帧格式错误<br>0：接收 FIFO 中数据没有错误<br>当接收 FIFO 中出错的数据是下一个将要读取的数据，且接收 FIFO 中其它的数据没有错误时，读取此寄存器清 0。 |
| 6  | TEMT      | R  | 1   | 发送完成标志：<br>1：发送完成，发送 FIFO 为空，且移位寄存器为空<br>0：发送未完成   |
| 5  | THRE      | R  | 1   | 发送 FIFO 空标志：<br>1：发送 FIFO 空<br>0：发送 FIFO 满   |
| 4  | RSV       | -  | -   | 保留   |
| 3  | FE        | R  | 0   | 帧格式出错标志：<br>1：帧格式错误<br>0：帧格式未出错<br>读此寄存器清 0。   |
| 2  | PE        | R  | 0   | 奇偶校验出错标志：<br>1：奇偶校验错误<br>0：奇偶校验未出错<br>读此寄存器清 0。  |
| 1  | OE        | R  | 0   | 接收 FIFO 溢出标志：<br>1：接收 FIFO 溢出<br>0：接收 FIFO 非溢出<br>读此寄存器清 0。  |
| 0  | DR        | R  | 0   | 接收 FIFO 非空标志：<br>1：接收 FIFO 非空<br>0：接收 FIFO 空   |

### 6.3.11 流状态寄存器 UART1\_MSR (偏移：18h)

| 比特   | 名称  | 属性 | 复位值 | 描述                                  |
|------|-----|----|-----|-------------------------------------|
| 31:5 | RSV | -  | -   | 保留                                  |
| 4    | CTS | R  | 0   | CTS 标志位<br>1：有 CTS 请求<br>0：无 CTS 请求 |
| 3:0  | RSV | -  | -   | 保留                                  |

### 6.3.12 状态寄存器 UART1\_USR (偏移: 7Ch)

| 比特   | 名称   | 属性 | 复位值 | 描述   |
|------|------|----|-----|--|
| 31:5 | RSV  | -  | -   | 保留   |
| 4    | RFF  | R  | 0   | 接收 FIFO 满标志。<br>1: 接收 FIFO 满<br>0: 接收 FIFO 非满  |
| 3    | RFNE | R  | 0   | 接收 FIFO 非空标志:<br>1: 接收 FIFO 非空<br>0: 接收 FIFO 空 |
| 2    | TFE  | R  | 1   | 发送 FIFO 空标志:<br>1: 发送 FIFO 空<br>0: 发送 FIFO 非空  |
| 1    | TFNF | R  | 1   | 发送 FIFO 非满标志:<br>1: 发送 FIFO 非满<br>0: 发送 FIFO 满 |
| 0    | BUSY | R  | 0   | 1: UART1 正在传输<br>0: UART1 处于空闲状态               |

### 6.3.13 发送 FIFO 数据个数寄存器 UART1\_TFL (偏移: 80h)

| 比特   | 名称  | 属性 | 复位值 | 描述              |
|------|-----|----|-----|-----------------|
| 31:5 | RSV | -  | -   | 保留              |
| 4:0  | TFL | R  | 0   | 发送 FIFO 中数据个数位。 |

### 6.3.14 接收 FIFO 数据个数寄存器 UART1\_RFL (偏移: 84h)

| 比特   | 名称  | 属性 | 复位值 | 描述              |
|------|-----|----|-----|-----------------|
| 31:5 | RSV | -  | -   | 保留              |
| 4:0  | RFL | R  | 0   | 接收 FIFO 中数据个数位。 |

### 6.3.15 小数分频寄存器 UART1\_DLF(偏移: C0h)

| 比特   | 名称  | 属性  | 复位值 | 描述  |
|------|-----|-----|-----|---|
| 31:5 | RSV | -   | -   | 保留  |
| 3:0  | DLF | R/W | 0   | 小数分频寄存器。<br>小数部分波特率为 DLF/16。<br>计算公式为: $(PCLK\%(BAUDRATE*16)) / BAUDRATE$ 。 |

### 6.3.16 接收地址匹配寄存器 UART1\_RAR(偏移: C4h)

| 比特   | 名称  | 属性 | 复位值 | 描述 |
|------|-----|----|-----|----|
| 31:5 | RSV | -  | -   | 保留 |

| 比特  | 名称  | 属性  | 复位值 | 描述                                |
|-----|-----|-----|-----|-----------------------------------|
| 7:0 | RAR | R/W | 0   | 接收地址匹配寄存器。此寄存器只有在 UART 处于空闲状态时可写。 |

### 6.3.17 发送地址匹配寄存器 UART1\_TAR (偏移: C8h)

| 比特   | 名称  | 属性  | 复位值 | 描述                                |
|------|-----|-----|-----|-----------------------------------|
| 31:5 | RSV | -   | -   | 保留                                |
| 4:0  | TAR | R/W | 0   | 发送地址匹配寄存器。此寄存器只有在 UART 处于空闲状态时可写。 |

## 6.4 使用流程

### 6.4.1 UART1 发送流程

1. 设置 UART1\_MCR 寄存器。
2. 设置 UART1\_LCR 寄存器的第 7 比特 DLAB 为 1。
3. 设置 UART1\_DLL/UART1\_DLH/UART1\_DLF 寄存器。
4. 设置 UART1\_LCR 寄存器的第 7 比特 DLAB 为 0，设置 UART1\_LCR 寄存器的其它位。
5. 设置 UART1\_FCR 寄存器。
6. 设置 UART1\_IER 寄存器。
7. 写 UART1\_THR 寄存器，向发送 FIFO 中填写数据。
8. 查询 UART1\_IIR 中断状态。
9. 完成传输。

### 6.4.2 UART1 接收流程

1. 设置 UART1\_MCR 寄存器。
2. 设置 UART1\_LCR 寄存器的第 7 比特 DLAB 为 1。
3. 设置 UART1\_DLL/UART1\_DLH/UART1\_DLF 寄存器。
4. 设置 UART1\_LCR 寄存器的第 7 比特 DLAB 为 0，设置 UART1\_LCR 寄存器的其它位。
5. 设置 UART1\_FCR 寄存器。
6. 设置 UART1\_IER 寄存器。
7. 查询 UART1\_IIR 中断状态。
8. 读取 UART1\_RBR，取走收到的数据。
9. 完成传输。

### 6.4.3 CTS 和 RTS 控制流功能设置流程

#### ● CTS

CTS为UART 输入端口，低电平有效，表示UART可以发送数据。如果CTS输入状态为 1，写 UART1\_THR 寄存器时，数据只会保存在发送FIFO中不会被发出，为0时开始发送。

CTS配置流程如下：

1. 配置 UART1\_CTS 管脚。
2. 配置 REG\_UART1\_MCR，使能 CTS/RTS 自动流控制。
3. 配置 REG\_UART1\_FCR，使能 FIFO。
4. UART1\_CTS 管脚输入为低电平时，UART 正常发送数据；输入为高电平时，数据保存在发送 FIFO 中。



图 6-1: CTS 时序图

#### ● RTS

RTS为UART 输出端口，低电平有效，输出为低电平时，表示UART已经准备好可以接收数据了；当接收FIFO中数据个数大于FIFO控制寄存器UART1\_FCR中接收FIFO非空中断设置的帧数据个数触发中断条件时，RTS输出状态为高电平，表示UART不能接收更多数据。

RTS 配置流程如下：

1. 配置 UART1\_RTS 管脚。
2. 配置 REG\_UART1\_MCR，使能 CTS/RTS 自动流控制，RTS 接口软件控制位为请求输出有效。
3. 配置 REG\_UART1\_FCR，使能 FIFO，接收 FIFO 非空中断设置。

### 6.4.4 UART1 DMA 传输配置流程

UART1 模块可支持 DMA 传输功能，支持三种传输模式：Memory to Peripheral 模式、Peripheral to Memory 模式、Peripheral to Peripheral 模式。配置流程如下：

1. 配置 SARx 寄存器，指定源地址。
2. 配置 DARx 寄存器，指定目标地址。
3. 配置 CTLx 寄存器。有需要时使能中断，设置传输数据宽度和大小，选择流控类型，注意 SRAM 和 FLASH 没有握手信号，在 TT\_FC 中将 SRAM 和 FLASH 认定为存储器选择地址是否递增/递减，选择 burst 传输长度。
4. 配置 CFGx 寄存器。选择通道优先级，设置是否自动重启传输，选定握手信号，将 HS\_SEL

域设为 0。

5. 有需要时配置 MaskBlock 寄存器，开启对应通道的中断。
6. 配置 DmaCfgReg 寄存器使能 DMAC。
7. 配置 ChEnReg 寄存器使能通道。
8. 等待中断或查询 ChEnReg。
9. 清除中断。
10. 配置 SARx 寄存器，指定源地址。
11. 配置 DARx 寄存器，指定目标地址。
12. 配置 CTLx 寄存器。有需要时使能中断，设置传输数据宽度和大小，选择流控类型，选择地址是否递增/递减，选择 burst 传输长度。
13. 配置 CFGx 寄存器。选择通道优先级，将 HS\_SEL 域设为 1。
14. 有需要时配置 MaskBlock 寄存器，开启对应通道的中断。
15. 配置 DmaCfgReg 寄存器使能 DMAC。
16. 配置 ChEnReg 寄存器使能通道。
17. 等待中断或查询 ChEnReg。
18. 清除中断。

另外，DMA 的第七握手信号功能可以通过 GPIO 中断或者 GTIMER 的 TIMER0 的中断信号作为启动 DMA 搬移的信号。通过以下流程配置 DMA 的第七握手信号功能：

1. 配置系统控制寄存器 DMA 第七握手信号使能。
2. 选择 TIMER0 或 GPIO 作为第七握手信号的选择。
3. 配置 TIMER0 或 GPIO 的中断模块。
4. 加入 DMA 传输方式配置流程（若使用了第七握手信号配置，不添加此模块功能前配置的是 Memory to Peripheral 模式功能，在这里要在配置改变为 Peripheral to Memory 模式）。

## 7 LPUART

### 7.1 概述

芯片有一个低功耗串口模块 LPUART，其工作仅需 32kHz 时钟，可以支持到最高 9600 波特率的数据接收。LPUART 功耗极低，可以在 Sleep/DeepSleep 模式下工作。

### 7.2 主要特性

- 异步数据收发
- 标准 UART 帧格式
  - 1bit 起始位
  - 7 或 8bit 数据
  - 奇校验、偶校验或无校验位
  - 1 或 2bit 停止位
- 使用 32768Hz XTL 时钟或者 32kHz RCL 时钟工作，支持波特率 300 bps~9600bps
- 可编程数据极性
- 支持 Sleep/DeepSleep 模式下的数据收发
- 休眠模式下唤醒芯片
  - RXD 下降沿唤醒
  - 起始位检测唤醒
  - 1 字节接收完成唤醒
  - 1 字节数据匹配唤醒

### 7.3 寄存器描述

LPUART 寄存器基地址：0x40000400

表 7-1: LPUART 寄存器列表

| 偏置   | 名称      | 描述      |
|------|---------|---------|
| 0x00 | LPURXD  | 接收数据寄存器 |
| 0x04 | LPUTXD  | 发送数据寄存器 |
| 0x08 | LPUSTA  | 状态寄存器   |
| 0x0C | LPUCON  | 控制寄存器   |
| 0x10 | LPUIF   | 中断标志寄存器 |
| 0x14 | LPUBAUD | 波特率寄存器  |
| 0x18 | LPUEN   | 接收使能寄存器 |
| 0x1C | COMPARE | 数据匹配寄存器 |

| 偏置   | 名称   | 描述         |
|------|------|------------|
| 0x20 | MODU | 波特率调制控制寄存器 |

### 7.3.1 接收数据寄存器 LPURXD (偏移: 00h)

| 比特   | 名称     | 属性 | 复位值 | 描述     |
|------|--------|----|-----|--------|
| 31:8 | RSV    | -  | -   | 保留     |
| 7:0  | LPURXD | R  | 0   | 接收数据缓冲 |

### 7.3.2 发送数据寄存器 LPUTXD (偏移: 04h)

| 比特   | 名称     | 属性 | 复位值 | 描述     |
|------|--------|----|-----|--------|
| 31:8 | RSV    | -  | -   | 保留     |
| 7:0  | LPUTXD | R  | 0   | 发送数据缓冲 |

### 7.3.3 状态寄存器 LPUSTA (偏移: 08h)

| 比特   | 名称    | 属性  | 复位值 | 描述  |
|------|-------|-----|-----|---|
| 31:8 | RSV   | -   | -   | 保留  |
| 7    | TC    | R   | 0   | 发送完成标志, 当一帧数据发送完成且发送 buffer 为空时置位。数据发送时清零。  |
| 6    | TXE   | R   | 0   | 发送 buffer 空标志, 硬件置位, 软件向发送 buffer 写数据时自动清零。 |
| 5    | START | R/W | 0   | 起始位检测标志, 写 1 清零。                            |
| 4    | PERR  | R/W | 0   | 校验位错误, 写 1 清零。                              |
| 3    | FERR  | R/W | 0   | 帧格式错误, 写 1 清零。                              |
| 2    | RXOV  | R/W | 0   | 接收缓冲溢出, 写 1 清零。                             |
| 1    | RXF   | R   | 0   | 接收缓冲满, 读 LPUdata 寄存器清零。                     |
| 0    | MATCH | R/W | 0   | 数据匹配标志, 表示接收缓冲区内的数据与比较寄存器相同, 写 1 清零。        |

### 7.3.4 控制寄存器 LPUCON (偏移: 0Ch)

| 比特    | 名称    | 属性  | 复位值 | 描述                                     |
|-------|-------|-----|-----|--|
| 31:13 | RSV   | -   | -   | 保留                                     |
| 12    | TXPOL | R/W | 0   | 数据发送极性<br>0: 非取反<br>1: 取反              |
| 11    | TCIE  | R/W | 0   | 发送完成中断使能<br>0: 禁止发送完成中断<br>1: 允许发送完成中断 |

| 比特  | 名称    | 属性  | 复位值 | 描述   |
|-----|-------|-----|-----|--|
| 10  | TXIE  | R/W | 0   | 发送 buffer 空中断使能<br>0: 禁止发送 buffer 空中断<br>1: 允许发送 buffer 空中断  |
| 9   | NEDET | R/W | 0   | 下降沿采样使能位<br>0: 使用 32k 时钟上升沿检测 start bit<br>1: 使用 32k 时钟下降沿检测 start bit                               |
| 8   | PAREN | R/W | 0   | 校验位使能<br>0: 数据帧无奇偶校验位<br>1: 数据帧有奇偶校验位  |
| 7   | PTYP  | R/W | 0   | 校验位类型<br>0: 偶校验<br>1: 奇校验  |
| 6   | SL    | R/W | 0   | 停止位长度<br>0: 1bit<br>1: 2bits   |
| 5   | DL    | R/W | 0   | 数据长度<br>0: 8bits<br>1: 7bits   |
| 4   | RXPOL | R/W | 0   | 接收极性<br>0: 非取反<br>1: 取反  |
| 3   | ERRIE | R/W | 0   | 错误中断使能<br>0: 禁止接收错误中断<br>1: 允许接收错误中断   |
| 2   | RXIE  | R/W | 0   | 接收中断使能<br>0: 禁止接收中断<br>1: 允许接收中断   |
| 1:0 | RXEVS | R/W | 00  | 接收中断事件配置，用于控制何种事件下向 CPU 提供接收中断<br>00: START 位检测唤醒<br>01: 1byte 数据接收完成<br>10: 接收数据匹配成功<br>11: 下降沿检测唤醒 |

### 7.3.5 中断标志寄存器 LPUIF (偏移: 10h)

| 比特   | 名称    | 属性  | 复位值 | 描述  |
|------|-------|-----|-----|---|
| 31:4 | RSV   | -   | -   | 保留  |
| 3    | TC_IF | R/W | 0   | 发送完成中断标志<br>1: 发送完一帧数据后中断产生<br>0: 无中断产生<br>写 1 清 0            |
| 2    | TXIF  | R/W | 1   | 发送 buffer 空中断标志<br>1: 发送 buffer 空后中断产生<br>0: 无中断产生<br>写 1 清 0 |



| 比特 | 名称      | 属性  | 复位值 | 描述   |
|----|---------|-----|-----|--|
| 1  | RXNEGIF | R/W | 0   | RXD 下降沿中断标志<br>1: 中断产生<br>0: 无中断产生<br>写 1 清 0      |
| 0  | RXIF    | R/W | 0   | 接收完成中断标志<br>1: 接收完一帧数据后中断产生<br>0: 无中断产生<br>写 1 清 0 |

### 7.3.6 波特率寄存器 LPUBAUD (偏移: 14h)

| 比特   | 名称   | 属性  | 复位值 | 描述  |
|------|------|-----|-----|---|
| 31:3 | RSV  | -   | -   | 保留  |
| 2:0  | BAUD | R/W | 000 | 波特率控制 (bps)<br>000: 9600<br>001: 4800<br>010: 2400<br>011: 1200<br>100: 600<br>101/110/111: 300 |

### 7.3.7 接收使能寄存器 LPUEN (偏移: 18h)

| 比特   | 名称   | 属性  | 复位值 | 描述  |
|------|------|-----|-----|---|
| 31:2 | RSV  | -   | -   | 保留  |
| 1    | TXEN | R/W | 0   | 发送使能<br>0: 关闭 LPUART 发送<br>1: 打开 LPUART 发送<br>Cpu 写 1 使能后, 要反复读取此寄存器, 直到读到 1 为止才能进行后面的操作。   |
| 0    | RXEN | R/W | 0   | 接收使能<br>0: 关闭 LPUART 接收;<br>1: 打开 LPUART 接收;<br>Cpu 写 1 使能后, 要反复读取此寄存器, 直到读到 1 为止才能进行后面的操作。 |

### 7.3.8 数据匹配寄存器 COMPARE (偏移: 1Ch)

| 比特   | 名称      | 属性  | 复位值      | 描述   |
|------|---------|-----|----------|--|
| 31:8 | RSV     | -   | -        | 保留   |
| 7:0  | COMPARE | R/W | 00000000 | 比较数据, 如果 RXEV=10/11, 当接收缓冲区内的数据与 COMPARE 相同时, 触发接收完成中断 |

### 7.3.9 波特率调制控制寄存器 MODU (偏移: 20h)

| 比特    | 名称   | 属性  | 复位值          | 描述                    |
|-------|------|-----|--------------|-----------------------|
| 31:12 | RSV  | -   | -            | 保留                    |
| 11:0  | MCTL | R/W | 000000000000 | LPUART 每个 bit 的调制控制信号 |

## 7.4 软件流程

### 7.4.1 数据接收

1. 配置 LPUBAUD 寄存器决定波特率。
2. 根据波特率选择合适的调制参数，配置调制控制寄存器 MODU 的 MCTL 值。
3. 配置 LPUCON 寄存器，选择帧格式、极性、中断参数等。
4. 配置 LPUEN 寄存器打开接收使能。
5. 等待中断事件。

### 7.4.2 数据发送

1. 配置 LPUBAUD 寄存器决定波特率。
2. 根据波特率选择合适的调制参数，配置调制控制寄存器 MODU 的 MCTL 值。
3. 配置 LPUCON 寄存器，选择帧格式、极性、中断参数等。
4. 配置 LPUEN 寄存器打开发送使能。
5. 等待中断事件。

### 7.4.3 调制控制寄存器配置建议

软件需要根据通信波特率的不同合理配置调制控制寄存器 MODU 的 MCTL，建议的配置参数表如下：

表 7-2：调制控制寄存器配置建议

| Baud | MCTL            |      |      |      |      |      |      |      |      |      |       |       |
|------|-----------------|------|------|------|------|------|------|------|------|------|-------|-------|
|      | Bit0<br>(start) | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 |
| 9600 | 0               | 1    | 0    | 0    | 1    | 0    | 1    | 0    | 1    | 0    | 0     | 1     |
| 4800 | 1               | 1    | 0    | 1    | 1    | 1    | 1    | 1    | 0    | 1    | 1     | 1     |
| 2400 | 1               | 1    | 0    | 1    | 1    | 0    | 1    | 1    | 0    | 1    | 1     | 0     |

| Baud | MCTL            |      |      |      |      |      |      |      |      |      |       |       |
|------|-----------------|------|------|------|------|------|------|------|------|------|-------|-------|
|      | Bit0<br>(start) | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7 | Bit8 | Bit9 | Bit10 | Bit11 |
| 1200 | 0               | 1    | 0    | 0    | 1    | 0    | 0    | 1    | 0    | 0    | 1     | 0     |
| 600  | 0               | 1    | 1    | 0    | 1    | 0    | 1    | 1    | 0    | 1    | 1     | 0     |
| 300  | 0               | 1    | 0    | 0    | 0    | 0    | 1    | 0    | 0    | 0    | 0     | 1     |

以上参数表假设 LPUART 工作时钟为准确的 32768Hz，如果使用 RCL 工作，则会引入额外的误差，可能需要微调波特率调制方案来获得更好的通信效果。

#### 7.4.4 休眠模式下的数据接收唤醒

LPUART 支持在 Sleep、DeepSleep 模式下进行数据接收并唤醒芯片。此时芯片功耗极低，并保持对 RXD 引脚的监听，直到特定事件到来后唤醒芯片退出休眠模式。

1. 配置 LPUBAUD 寄存器决定波特率。
2. 根据波特率选择合适的调制参数，配置 MCTL 寄存器。
3. 配置 LPUCON 寄存器，选择帧格式、极性，通过 LPUCON.RXEV 选择唤醒事件为 START 位、一帧接收完成、一帧数据匹配或 RXD 下降沿检测。
4. 配置 LPUEN 寄存器打开接收使能。
5. 软件进入 Sleep/DeepSleep。

## 8 I2C

### 8.1 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。

### 8.2 主要特征

- 支持主机接收、发送，从机接收、发送四种工作模式
- 支持标准（100Kbps）/快速（400Kbps）/高速（1Mbps）三种工作速率
- 支持 7 位寻址功能和 10 位寻址功能
- 支持广播地址
- 支持中断查询功能

### 8.3 寄存器描述

I2C 寄存器基地址：0x40005000

表 8-1：I2C 寄存器列表

| 偏置   | 名称         | 描述                  |
|------|------------|---------------------|
| 0x00 | I2C_CR     | I2C 配置寄存器           |
| 0x04 | I2C_CLR    | I2C 配置清除寄存器         |
| 0x08 | I2C_STAT   | I2C 状态寄存器           |
| 0x0C | I2C_DATA   | I2C 数据寄存器           |
| 0x10 | I2C_CCR    | I2C 波特率配置寄存器        |
| 0x14 | I2C_SAD0   | I2C SLAVE 地址寄存器 0   |
| 0x18 | I2C_SADM0  | I2C SLAVE 地址屏蔽寄存器 0 |
| 0x1C | I2C_XSAD0  | I2C SLAVE 扩展地址寄存器   |
| 0x20 | I2C_XSADM0 | I2C SLAVE 扩展地址屏蔽寄存器 |
| 0x24 | I2C_SRST   | I2C 复位寄存器           |
| 0x28 | I2C_SAD1   | I2C SLAVE 地址寄存器 1   |
| 0x2C | I2C_SADM1  | I2C SLAVE 地址屏蔽寄存器 1 |
| 0x30 | I2C_SAD2   | I2C SLAVE 地址寄存器 2   |
| 0x34 | I2C_SADM2  | I2C SLAVE 地址屏蔽寄存器 2 |
| 0x38 | I2C_SAD3   | I2C SLAVE 地址寄存器 3   |
| 0x3c | I2C_SADM3  | I2C SLAVE 地址屏蔽寄存器 3 |

### 8.3.1 I2C 配置寄存器 I2C\_CR (偏移: 00h)

| 比特   | 名称      | 属性  | 复位值 | 描述  |
|------|---------|-----|-----|---|
| 31:9 | RSV     | -   | -   | 保留  |
| 8    | GCAVAL  | R   | 0   | General Call 地址标志位。<br>1: 收到 General Call 地址<br>0: 未收到 General Call 地址  |
| 7    | IEN     | R/W | 0   | I2C 模块中断使能。<br>1: 中断使能<br>0: 中断禁止   |
| 6    | ENAB    | R/W | 0   | 从模式下, I2C 模块使能位。<br>1: I2C 从模式下, 模块使能。<br>0: I2C 从模式下, 模块禁止。I2C 不进行地址匹配, 忽略掉 SCL/SDA 线上的信息。                         |
| 5    | STA     | R/W | 0   | 开始标志使能。<br>1: 发送 START 标志; 发送 START 表之后, 自动清 0<br>0: 不发送 START 标志   |
| 4    | STP     | R/W | 0   | 停止标志使能。<br>1: 发送 STOP 标志; 发送 STOP 表之后, 自动清 0<br>0: 不发送 STOP 标志  |
| 3    | IFLG    | R/W | 0   | 中断标志位;<br>I2C_STAT 寄存器处于 0xf8 以外的任何状态, 此位都会置位。<br>写 I2C_CLR 寄存器的 CLR_IFLG 位, 清 0。STP 位写 1 时, 即发送 STOP 标志后, 此位也将清 0。 |
| 2    | AAK     | R/W | 0   | 应答标志使能。<br>1: 应答 ACK<br>0: 应答 NACK<br>置 1 后, 写 I2C_CLR 寄存器的 CLR_AAK 位清 0。   |
| 1    | SLAV10M | R   | 0   | 作为 SLAVE 时, 收到的数据与扩展地址寄存器中数据相匹配的标志位。<br>1: 接收到的数据与 SLAVE 扩展地址寄存器中的数值相匹配<br>0: 接收到的数据与 SLAVE 扩展地址寄存器中的数据不相同          |
| 0    | SLAV7M  | R   | 0   | 作为 SLAVE 时, 收到的数据与地址寄存器中数据相匹配的标志位。<br>1: 接收到的数据与 SLAVE 地址寄存器中的数值相匹配<br>0: 接收到的数据与 SLAVE 地址寄存器中的数据不相同                |

### 8.3.2 I2C 配置清除寄存器 I2C\_CLR (偏移: 04h)

| 比特   | 名称       | 属性  | 复位值 | 描述  |
|------|----------|-----|-----|---|
| 31:9 | RSV      | -   | -   | 保留  |
| 8    | GCAVAL   | R/W | 0   | General Call 功能使能位。<br>1: General Call 功能使能<br>0: General Call 功能禁止 |
| 7    | CLR_IEN  | W   | 0   | I2C 模块中断使能清除寄存器。<br>1: 清除中断使能<br>0: 中断使能状态保持                        |
| 6    | CLR_ENAB | W   | 0   | I2C 模块使能清除寄存器。<br>1: 关闭 I2C 模块<br>0: 保持 I2C 当前状态                    |
| 5    | CLR_STA  | W   | 0   | 开始标志清除寄存器。<br>1: 清除发送 START 标志<br>0: 保持 START 当权设置不变                |
| 4    | RSV      | -   | -   | 保留  |
| 3    | CLR_IFLG | W   | 0   | 中断标志清除寄存器；<br>1: 清除中断标志<br>0: 保持中断标志不变                              |
| 2    | CLR_AAK  | W   | 0   | 应答标志清除寄存器：<br>1: 清除应答标志<br>0: 保持应答标志不变                              |
| 1:0  | RSV      | -   | -   | 保留  |

### 8.3.3 I2C 状态寄存器 I2C\_STAT(偏移: 08h)

| 比特   | 名称  | 属性 | 复位值  | 描述         |
|------|-----|----|------|------------|
| 31:8 | RSV | -  | -    | 保留         |
| 7:0  | STA | R  | 0xF8 | I2C 状态寄存器。 |

I2C\_STAT 寄存器 STA 字段不同代码代表的意义如下表所示：

| 状态代码 | I2C 总线和硬件状态                       |
|------|-----------------------------------|
| 0x08 | 已发送 START 标志                      |
| 0x10 | 已发送 RESTART 标志                    |
| 0x18 | 已发送 SLAVE 地址加 W 标志，并接收 ACK 位      |
| 0x20 | 已发送 SLAVE 地址加 W 标志，并接收 NAK 位      |
| 0x28 | 已发送 I2C_DATA 中的数据，已接收 ACK         |
| 0x30 | 已发送 I2C_DATA 中的数据，接收到 NAK         |
| 0x38 | 丢失仲裁                              |
| 0x40 | 已发送 SLAVE 地址加 R 标志，并加收到 ACK       |
| 0x48 | 已发送 SLAVE 地址加 R 标志，并加收到 NAK       |
| 0x50 | 已接收数据字节，ACK 已发出                   |
| 0x58 | 已接收数据字节，NAK 已发出                   |
| 0x60 | 已接收自身的 SLAVE 寄存器地址加 W 标志，ACK 已发出。 |

| 状态代码 | I2C 总线和硬件状态   |
|------|---|
| 0x68 | 丢失掉仲裁，并且以接收到自身的 SLAVE 寄存器地址加 W 标志，ACK 已发出。                        |
| 0x70 | 已接收通用调用地址 (0x00)；已发出 ACK  |
| 0x78 | 丢失掉仲裁，并且已接收到通用调用地址加 W 标志，ACK 已发出                                  |
| 0x80 | 前一次寻址使用自身从地址；已接收数据字节；已返回 ACK                                      |
| 0x88 | 前一次寻址使用自身从地址；已接收数据字节；已返回非 ACK                                     |
| 0x90 | 前一次寻址使用通用调用地址；已接收数据；已返回 ACK                                       |
| 0x98 | 前一次寻址使用通用调用地址；已接收数据；已返回非 ACK                                      |
| 0xA0 | 当使用从接收/从发送模式中静态寻址时，接收到停止条件或重复起始条件                                 |
| 0xA8 | 已接收自身的从地址加 R 标志；已返回 ACK   |
| 0xB0 | 丢失掉仲裁，并且已接收到通用调用地址加 R 标志，ACK 已发出。                                 |
| 0xB8 | 已发送数据；已接收 ACK   |
| 0xC0 | 已发送数据字节；已接收非 ACK  |
| 0xC8 | 装入的数据字节已被发送；已接收 ACK   |
| 0xF8 | 无可用的相关状态信息  |
| 0x00 | 由于非法的起始或停止条件的出现，在主机或被选中的从机将出现总线错误；当外部干扰使 I2C 进入未定义的状态时也会出 0x00 状态 |
| 0xE0 | 已发送第二次设备地址，已接收 ACK  |
| 0xE8 | 已发送第二次设备地址，已接收非 ACK   |

### 8.3.4 I2C 数据寄存器 I2C\_DATA(偏移：0Ch)

| 比特   | 名称   | 属性  | 复位值 | 描述   |
|------|------|-----|-----|--|
| 31:8 | RSV  | -   | -   | 保留   |
| 7:0  | DATA | R/W | 0x0 | I2C 数据寄存器。<br>在 I2C 发送模式下，写发送数据到这个寄存器。<br>在 I2C 接收模式下，读接收数据从这个寄存器。 |

### 8.3.5 I2C 波特率配置寄存器 I2C\_CCR(偏移：10h)

| 比特   | 名称   | 属性  | 复位值 | 描述       |
|------|------|-----|-----|----------|
| 31:7 | RSV  | -   | -   | 保留       |
| 6:4  | CCRM | R/W | 0x0 | 波特率配置位 M |
| 3:0  | CCRN | R/W | 0x0 | 波特率配置位 N |

$F_{OSCL} = F_{SCL} = P_{CLK} / (2^M \times (N+1) \times 10)$ ；其中， $F_{OSCL}$  是 I2C 接口输出的 SCL 得频率。

### 8.3.6 I2C SLAVE 地址寄存器 0 I2C\_SAD0 (偏移: 14h)

| 比特   | 名称   | 属性  | 复位值 | 描述                           |
|------|------|-----|-----|------------------------------|
| 31:8 | RSV  | -   | -   | 保留                           |
| 7:1  | ADR0 | R/W | 0x0 | I2C 从机模式地址 0                 |
| 0    | GC0  | R/W | 0x0 | 广播地址应答使能。<br>1: 使能<br>0: 不使能 |

### 8.3.7 I2C SLAVE 地址屏蔽寄存器 0 I2C\_SADM0 (偏移: 18h)

| 比特   | 名称   | 属性  | 复位值  | 描述   |
|------|------|-----|------|--|
| 31:8 | RSV  | -   | -    | 保留   |
| 7:1  | AMR0 | R/W | 0x7F | I2C 从机模式地址屏蔽寄存器 0。每一位与 ADR0 中的位相对应。<br>对应位为 1 表示, 在此 I2C 模块作为从机时, 比较 ADR0 中对应位的值。<br>对应位为 0 表示, 在此 I2C 模块作为从机时, 不比较 ADR0 中对应位的值。 |
| 0    | RSV  | -   | -    | 保留   |

### 8.3.8 10 比特 I2C SLAVE 地址寄存器 I2C\_XSAD (偏移: 1Ch)

| 比特    | 名称   | 属性  | 复位值 | 描述                                       |
|-------|------|-----|-----|--|
| 31:11 | RSV  | -   | -   | 保留                                       |
| 10:1  | XADR | R/W | 0x0 | I2C 从机模式 10 比特地址位。                       |
| 0     | XGC  | R/W | 0x0 | 10 比特地址模式下, 广播地址应答使能。<br>1: 使能<br>0: 不使能 |

### 8.3.9 10 比特 I2C SLAVE 地址屏蔽寄存器 I2C\_XSADM (偏移: 20h)

| 比特    | 名称   | 属性  | 复位值  | 描述   |
|-------|------|-----|------|--|
| 31:11 | RSV  | -   | -    | 保留   |
| 10:1  | XAMR | R/W | 0xff | 10 比特 I2C 从机模式地址屏蔽寄存器 0。每一位与 XADR 中的位相对应。<br>对应位为 1 表示, 在此 I2C 模块作为从机时, 比较 XADR 中对应位的值。<br>对应位为 0 表示, 在此 I2C 模块作为从机时, 不比较 XADR 中对应位的值。 |
| 0     | RSV  | -   | -    | 保留   |



### 8.3.10 I2C 复位寄存器 I2C\_SRST (偏移: 24h)

| 比特   | 名称   | 属性 | 复位值 | 描述                |
|------|------|----|-----|-------------------|
| 31:0 | SRST | W  | 0x0 | 写此寄存器, 复位 I2C 模块。 |

### 8.3.11 I2C SLAVE 地址寄存器 1 I2C\_SAD1 (偏移: 28h)

| 比特   | 名称   | 属性  | 复位值 | 描述                           |
|------|------|-----|-----|------------------------------|
| 31:8 | RSV  | -   | -   | 保留                           |
| 7:1  | ADR1 | R/W | 0x0 | I2C 从机模式地址 1                 |
| 0    | GC1  | R/W | 0x0 | 广播地址应答使能。<br>1: 使能<br>0: 不使能 |

### 8.3.12 I2C SLAVE 地址屏蔽寄存器 1 I2C\_SADM1 (偏移: 2Ch)

| 比特   | 名称   | 属性  | 复位值  | 描述   |
|------|------|-----|------|--|
| 31:8 | RSV  | -   | -    | 保留   |
| 7:1  | AMR1 | R/W | 0x7f | I2C 从机模式地址屏蔽寄存器 1。每一位与 ADR1 中的位相对应。<br>对应位为 1 表示, 在此 I2C 模块作为从机时, 比较 ADR1 中对应位的值。<br>对应位为 0 表示, 在此 I2C 模块作为从机时, 不比较 ADR1 中对应位的值。 |
| 0    | RSV  | -   | -    | 保留   |

### 8.3.13 I2C SLAVE 地址寄存器 2 I2C\_SAD2 (偏移: 30h)

| 比特   | 名称   | 属性  | 复位值 | 描述                           |
|------|------|-----|-----|------------------------------|
| 31:8 | RSV  | -   | -   | 保留                           |
| 7:1  | ADR2 | R/W | 0x0 | I2C 从机模式地址 2                 |
| 0    | GC2  | R/W | 0x0 | 广播地址应答使能。<br>1: 使能<br>0: 不使能 |

### 8.3.14 I2C SLAVE 地址屏蔽寄存器 2 I2C\_SADM2 (偏移: 34h)

| 比特   | 名称  | 属性 | 复位值 | 描述 |
|------|-----|----|-----|----|
| 31:8 | RSV | -  | -   | 保留 |

| 比特  | 名称   | 属性  | 复位值  | 描述   |
|-----|------|-----|------|--|
| 7:1 | AMR2 | R/W | 0x7f | I2C 从机模式地址屏蔽寄存器 2。每一位与 ADR2 中的位相对应。<br>对应位为 1 表示，在此 I2C 模块作为从机时，比较 ADR2 中对应位的值。<br>对应位为 0 表示，在此 I2C 模块作为从机时，不比较 ADR2 中对应位的值。 |
| 0   | RSV  | -   | -    | 保留   |

### 8.3.15 I2C SLAVE 地址寄存器 2 I2C\_SAD3 (偏移：38h)

| 比特   | 名称   | 属性  | 复位值 | 描述                           |
|------|------|-----|-----|------------------------------|
| 31:8 | RSV  | -   | -   | 保留                           |
| 7:1  | ADR3 | R/W | 0x0 | I2C 从机模式地址 3                 |
| 0    | GC3  | R/W | 0x0 | 广播地址应答使能。<br>1: 使能<br>0: 不使能 |

### 8.3.16 I2C SLAVE 地址屏蔽寄存器 3 I2C\_SADM3 (偏移：3Ch)

| 比特   | 名称   | 属性  | 复位值  | 描述   |
|------|------|-----|------|--|
| 31:8 | RSV  | -   | -    | 保留   |
| 7:1  | AMR3 | R/W | 0x7f | I2C 从机模式地址屏蔽寄存器 2。每一位与 ADR3 中的位相对应。<br>对应位为 1 表示，在此 I2C 模块作为从机时，比较 ADR3 中对应位的值。<br>对应位为 0 表示，在此 I2C 模块作为从机时，不比较 ADR3 中对应位的值。 |
| 0    | RSV  | -   | -    | 保留   |

## 8.4 协议描述

标准 I2C 协议包含了五个部分：起始信号或重复其实信号、从机地址传输和 R/W 位传输、数据信号、确认信号和结束信号。

### 8.4.1 I2C 通信协议（7 位寻址）

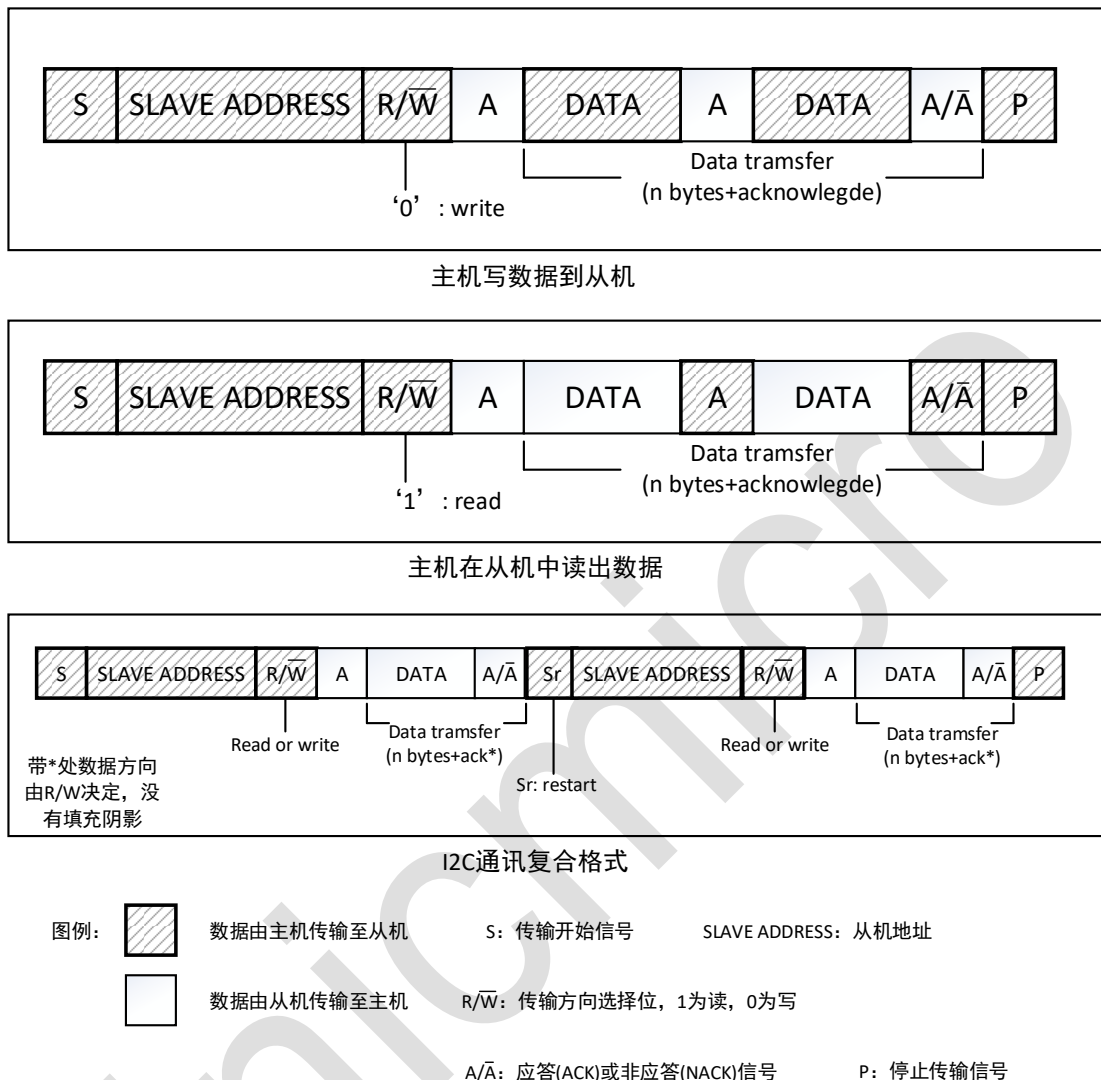


图 8-1: I2C 通信协议(7 位寻址)框图

I2C 的基本读写过程如下:

在第一幅图中, 配置的方向为“写数据 (W)”。主机在广播完从机地址, 接收到应答信号后, 开始正式向从机发送数据包 (DATA), 数据包的大小为 8 位, 主机每发送完一个字节数据后都要等待从机发来的应答信号 (ACK), 然后再发送下一个字节数据。发送数据包的数量没有限制。最后当数据传输结束时, 主机向从机发送一个停止传输信号 (P), 传输停止。

在第二幅图中, 配置的方向为“读数据 (R)”。主机在广播完从机地址, 接收到应答信号后, 开始接收从机发送的数据包 (DATA), 数据包的大小为 8 位, 主机每接收完一个字节数据后都要发送一个应答信号 (ACK), 从机在收到此应答信号以后再发送下一个字节数据。接收数据包的数量没有限制。最后当主机希望停止数据传输时, 要向从机发送一个非应答信号 (NACK), 则从机停止传输。

除了基本的读写, I2C 通讯更常用的是复合格式, 即第三幅图的情况, 在该传输中, 主机会在第一次传输的数据段 (DATA 部分) 中发送从设备内部的寄存器或存储器地址 (注意不是 SLAVE

ADDRESS); 在第二次传输中, 对该地址的内容进行读写。

### 8.4.2 I2C 通信协议 (10 位寻址)

I2C 总线的 10bit 寻址和 7bit 寻址是兼容的, 这样就可以在同一个总线上同时使用 7bit 地址和 10bit 地址模式的设备。10bit 的从机地址由开始条件(S)或重复开始条件(Sr)后的两个字节数据组成。第一个字节的前 7 位是 1111 0XX, XX 是 10bit 地址的最高有效位的前两位(A9, A8), 第 8bit 是读写位, 决定传输方向。第二个字节为 10bit 地址剩下的 8 位(A7-A0)。

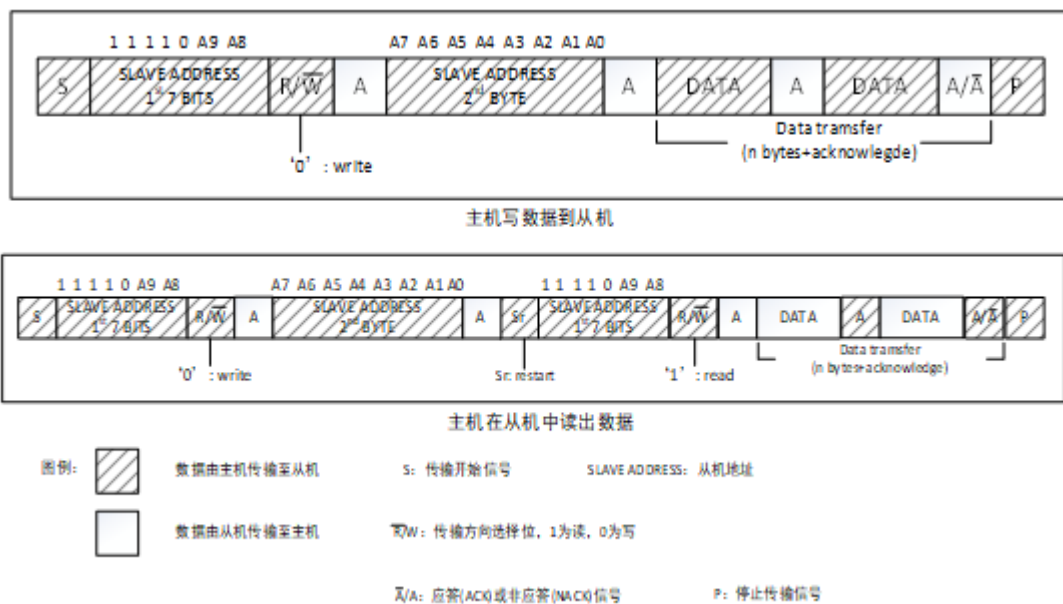


图 8-2: I2C 通信协议(10 位寻址)框图

I2C 的基本读写过程如下:

在第一幅图中, 主机作为发送器向从机发送数据。当接收到 START 条件后的 7bit 地址, 从机会用存在 XSAD 寄存器里的地址与接收到的第一个字节(11110XX)进行比较, 并检查第八个 bit (读写位) 是否为 0。有可能多个设备都匹配并产生应答 ACK。接下来从机开始匹配自己的地址与第二个字节的 8 个 bit (XXXXXXXX), 这时就只有一个从机匹配并产生应答 ACK。匹配完成后主机可开始向从机传输数据。被主机寻址匹配的从机会保持被寻址的状态直到接收到终止条件或者是重复开始条件后跟着一个不同的从机地址。

在第二幅图中, 主机作为接收器从从机接收数据。在第二个应答 ACK 之前, 处理过程与上面图一一致。在重复开始条件(Sr)之后, 匹配的从机会保持被寻址的状态, 此时从机会检查 Sr 之后的第一个字节的前 7bit 是否正确, 并测试第 8 个 bit 是否为 1 (读)。如果是, 从机就认定它被作为一个发送器被寻址并产生应答 ACK。匹配完成后从机可开始向主机传输数据。

## 8.5 使用流程

### 8.5.1 初始化程序

将 I2C 接口初始化用作从机和/或主机的例子：

1. 将自身的从机地址装入 I2C\_SAD0/ I2C\_SAD1/ I2C\_SAD2/ I2C\_SAD3/ I2C\_XSAD，设置好地址匹配寄存器 I2C\_SADM0/ I2C\_SADM1/ I2C\_SADM2/ I2C\_SADM3/ I2C\_XSADM, I2C\_SADx 寄存器 GC 位置位，使能广播地址应答（如果需要）。
2. I2C\_CR 寄存器 IEN 位置位，使能 I2C 中断。
3. 写 I2C\_CCR 寄存器设置 I2C 通信速率（标准/快速/高速）。
4. 对于从机模式，设置 I2C\_CR 寄存器的 AAK、ENAB 位。

### 8.5.2 主机发送功能

1. 向 I2C\_CR 寄存器的 STA 位写 1，发出 START 标志。
2. 等待 I2C\_STAT 寄存器数值变为 0x08（已发送 START 标志）。
3. 向 I2C\_CLR 中的 CLR\_STA 位写 1，清除 STA 发送标志。
4. 向 I2C\_DATA 寄存器写入 SLA（7 位/10 位）+W（0）。
5. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1，发送 SLA+W。
6. 等待 I2C\_STAT 寄存器数值变为 0x18（已发送 SLAVE 地址加 W 标志，并接收 ACK 位）。
7. 向 I2C\_DATA 寄存器写入待发送的数据/要写入的从设备内存地址（10bit 寻址为发送第二次设备地址+w）。
8. 接收到从设备 I2C\_STAT\_SECOND\_ADD\_ACK=0xe0 状态为第二次地址已经发送并收到 ACK，接着循环发送数据（10bit 寻址才有此处通讯动作）。
9. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1，发送数据。
10. 等待 I2C\_STAT 寄存器数值变为 0x28（已发送 I2C\_DATA 中的数据，已接收 ACK）。
11. 重复上述的步骤，直到待发的数据全部发送完毕。
12. 向 I2C\_CR 寄存器的 STP 位写 1，发送 STOP 标志，传输完成。

### 8.5.3 主机接收功能

1. 向 I2C\_CR 寄存器的 STA 位写 1，发出 START 标志。
2. 等待 I2C\_STAT 寄存器数值变为 0x08（已发送 START 标志）。
3. 向 I2C\_CLR 中的 CLR\_STA 位写 1，清除 STA 发送标志。
4. 向 I2C\_DATA 寄存器写入 SLA（7 位/10 位）+W(0)。

5. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1，发送 SLA+W。
6. 等待 I2C\_STAT 寄存器数值变为 0x18（已发送 SLAVE 地址加 W 标志，并接收 ACK 位）；
7. 向 I2C\_DATA 寄存器写入待发送的数据/要写入的从设备内存地址（10 寻址方式则为发送第二次从设备地址+W）。
8. 等待第二次地址发送成功 I2C\_STAT=e0，继续发出 START 标志（这里在 10 寻址方式出现）。
9. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1，发送数据。
10. 向 I2C\_CR 寄存器的 STA 位写 1，发出 RESTART 标志。
11. 等待 I2C\_STAT 寄存器数值变为 0x10（已发送 RESTART 标志）。
12. 向 I2C\_DATA 寄存器写入 SLA（7 位或 10 位）+R(1)。
13. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1，发送 SLA+R。
14. 等待 I2C\_STAT 寄存器数值变为 0x40（已发送 SLAVE 地址加 R 标志，并加收到 ACK）。
15. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
16. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1，开始接收数据。
17. 等待 I2C\_STAT 寄存器数值变为 0x50（已接收数据字节，ACK 已发出），读取 I2C\_DATA 中收到的数据；
18. 重复上述的步骤，直到接收完所有数据。
19. 向 I2C\_CR 寄存器的 STP 位写 1，发送 STOP 标志，传输完成。

## 8.5.4 从机接收功能

- I2C\_STAT: 0x60（已接收自身的 SLAVE 寄存器地址加 W 标志，ACK 已发出）
  1. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
  2. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1，开始接收数据。
- I2C\_STAT: 0x68（用作主机时丢失掉仲裁，并且已接收到自身的 SLAVE 寄存器地址加 W 标志，ACK 已发出）
  1. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
  2. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1，开始接收数据。
- I2C\_STAT: 0x78（用作主机时丢失掉仲裁，并且已接收到通用调用地址加 W 标志，ACK 已发出）
  1. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
  2. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1，开始接收数据。
- I2C\_STAT: 0x70（已接收通用调用地址（0x00）；已发出 ACK）
  1. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
  2. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1，开始接收数据。
- I2C\_STAT: 0x80（前一次寻址使用自身从地址；已接收数据字节；已返回 ACK）

1. 读取 I2C\_DATA 中收到的数据。
  2. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
  3. 接收数据长度加 1。
  4. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
- I2C\_STAT: 0x88（前一次寻址使用自身从地址；已接收数据字节；已返回非 ACK）
    1. 读取 I2C\_DATA 中收到的数据。
    2. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
    3. 接收数据长度加 1。
    4. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
  - I2C\_STAT: 0x90（前一次寻址使用通用调用地址；已接收数据；已返回 ACK）
    1. 读取 I2C\_DATA 中收到的数据。
    2. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
    3. 接收数据长度加 1。
    4. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
  - I2C\_STAT: 0x98（前一次寻址使用通用调用地址；已接收数据；已返回非 ACK）
    1. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
    2. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
  - I2C\_STAT: 0xA0（当使用从接收/从发送模式中静态寻址时，接收到停止条件或重复起始条件）
    1. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
    2. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。

### 8.5.5 从机发送功能

- I2C\_STAT: 0x60，已接收自身的 SLAVE 寄存器地址加 W 标志，ACK 已发出（10 位寻址出现）
  1. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
  2. 向 I2C\_DATA 寄存器写入待发送的数据。
  3. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
  4. 发送数据长度加 1。
- I2C\_STAT: 0x68，丢失掉仲裁，并且以接收到自身的 SLAVE 寄存器地址加 W 标志，ACK 已发出（10 位寻址出现）。
  1. 向 I2C\_CR 寄存器的 AAK 位写 1，设置 I2C\_CR 寄存器的 AAK 位。
  2. 向 I2C\_DATA 寄存器写入待发送的数据。
  3. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
  4. 发送数据长度加 1。
- I2C\_STAT: 0x70，已接收通用调用地址（0x00）；已发出 ACK（10 位寻址出现）；

1. 向 I2C\_CR 寄存器的 AAK 位写 1, 设置 I2C\_CR 寄存器的 AAK 位。
  2. 向 I2C\_DATA 寄存器写入待发送的数据。
  3. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
  4. 发送数据长度加 1。
- I2C\_STAT: 0xA0, 当使用从接收/从发送模式中静态寻址时, 接收到停止条件或重复起始条件(10 位寻址出现)
    1. 向 I2C\_CR 寄存器的 AAK 位写 1, 设置 I2C\_CR 寄存器的 AAK 位。
    2. 向 I2C\_DATA 寄存器写入待发送的数据。
    3. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
    4. 发送数据长度加 1。
  - I2C\_STAT: 0xA8 (已接收自身的从地址加 R 标志; 已返回 ACK)
    1. 向 I2C\_CR 寄存器的 AAK 位写 1, 设置 I2C\_CR 寄存器的 AAK 位。
    2. 向 I2C\_DATA 寄存器写入待发送的数据。
    3. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
    4. 发送数据长度加 1。
  - I2C\_STAT: 0xB0 (用作从机时, 丢失掉仲裁, 并且已接收到通用调用地址加 R 标志, ACK 已发出)
    1. 向 I2C\_CR 寄存器的 AAK 位写 1, 设置 I2C\_CR 寄存器的 AAK 位。
    2. 向 I2C\_DATA 寄存器写入待发送的数据。
    3. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
    4. 发送数据长度加 1。
  - I2C\_STAT: 0xB8 (已发送数据; 已接收 ACK)
    1. 向 I2C\_CR 寄存器的 AAK 位写 1, 设置 I2C\_CR 寄存器的 AAK 位。
    2. 向 I2C\_DATA 寄存器写入待发送的数据。
    3. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
    4. 发送数据长度加 1。
  - I2C\_STAT: 0xC8 (装入的数据字节已被发送; 已接收 ACK)
    1. 向 I2C\_CR 寄存器的 AAK 位写 1, 设置 I2C\_CR 寄存器的 AAK 位。
    2. 向 I2C\_DATA 寄存器写入待发送的数据。
    3. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。
    4. 发送数据长度加 1。
  - I2C\_STAT: 0xC0 (已发送数据字节; 已接收非 ACK)
    1. 向 I2C\_CR 寄存器的 AAK 位写 1, 设置 I2C\_CR 寄存器的 AAK 位。
    2. 向 I2C\_CLR 中的 CLR\_IFLG 位写 1。



## 9 SPI0

### 9.1 概述

串行外设接口（Serial Peripheral Interface, SPI）是外部设备通过单线交换数据的串行同步通讯手段。芯片提供了一个 SPI0 接口模块，可配置为主设备或从设备，实现与外部的 SPI 通信。

### 9.2 主要特性

- 全双工或半双工单数据线串行同步收发
- 主从模式
- 可编程时钟极性和相位（支持模式 0、1、2、3）
- 可编程比特速率
- 从模式最大频率为  $F_{sys}/2$
- 传输结束中断标志
- 写冲突错标志
- 主模式错误检测、保护和中断标志
- 支持 DMA

### 9.3 寄存器描述

SPI0 寄存器基地址：0x40000800

表 9-1: SPI0 寄存器列表

| 偏置   | 名称             | 描述               |
|------|----------------|------------------|
| 0x0  | SPI0CR         | SPI0 配置寄存器       |
| 0x4  | SPI0CS0        | SPI0 主模式控制寄存器 0  |
| 0x8  | SPI0CS1        | SPI0 主模式控制寄存器 1  |
| 0x14 | SPI0OPCR       | SPI0 过程控制寄存器     |
| 0x18 | SPI0IE         | SPI0 中断控制寄存器     |
| 0x1C | SPI0IF         | SPI0 中断标志寄存器     |
| 0x20 | SPI0TXBUF      | SPI0 发送缓存寄存器     |
| 0x24 | SPI0RXBUF      | SPI0 接收缓存寄存器     |
| 0x28 | DMA_SPI0RX_LEV | SPI0 DMA 接收设置寄存器 |
| 0x2C | DMA_SPI0TX_LEV | SPI0 DMA 发送设置寄存器 |

### 9.3.1 SPI0 配置寄存器 SPI0CR (偏移: 00h)

| 比特    | 名称        | 属性  | 复位值 | 描述   |
|-------|-----------|-----|-----|--|
| 31:14 | RSV       | -   | -   | 保留   |
| 13    | DMA_TX_EN | R/W | 0   | Dma Tx 使能<br>1: 使能 DMA TX 请求;<br>0: 关闭 DMA TX 请求。  |
| 12    | DMA_RX_EN | R/W | 0   | Dma Rx 使能<br>1: 使能 DMA RX 请求;<br>0: 关闭 DMA RX 请求。  |
| 11    | FLTEN     | R/W | 1   | Slave 输入管脚滤波使能 (SSN/SCK/MOSI)<br>1: 使能 4ns 滤波<br>0: 不滤波  |
| 10    | SSNM      | R/W | 0   | Master 模式下 SSN 控制模式选择<br>1: 每发送完 8bit 后 Master 拉高 SSN, 维持高电平时间由 WAIT 寄存器控制<br>0: 每发送完 8bit 后 Master 保持 SSN 为低, 维持低电平时间由 WAIT 寄存器控制 |
| 9     | TXO_AC    | R/W | 1   | TXONLY 硬件自动清空的使能<br>1: TXONLY 硬件自动清零有效, 软件使能 TXO 后, 等待发送完毕后, 硬件清零<br>0: 关闭 TXONLY 硬件自动清零   |
| 8     | TXO       | R/W | 0   | TXONLY 控制位<br>1: 启动 Master 的单发送模式<br>0: 关闭单发送模式  |
| 7     | MSPA      | R/W | 0   | Master Sampling Position Adjustment, Master 对 MISO 信号的采样位置调整, 用于高速通信时补偿 PCB 走线延迟<br>1: 采样点延迟半个 SCK 周期<br>0: 不调整                    |
| 6     | SSPA      | R/W | 0   | Slave Sending Position Adjustment, Slave MISO 发送位置调整<br>1: 提前半个 SCK 周期发送<br>0: 不调整   |
| 5     | MM        | R/W | 1   | Master/Slave 模式选择。<br>1: Master 模式<br>0: Slave 模式  |
| 4:3   | WAIT      | R/W | 0   | Master 模式下, 每发完 8Bit 后加入至少 (1+WAIT)个 SCK cycle 等待时间再传输下一个 8Bit 的数据   |
| 2     | RSV       | -   | -   | RFU: 未实现, 读为 0   |
| 1     | SSNSEN    | R/W | 0   | Master 模式下, 软件控制 SSN 使能<br>1: Master 模式下 SSN 输出由软件控制<br>0: Master 模式下 SSN 输出由硬件自动控制  |
| 0     | SPI0EN    | R/W | 0   | SPI0 使能。采用关闭时钟的方式来关闭使能。<br>1: 使能 SPI0<br>0: 关闭 SPI0, 清空发送接收缓存  |

### 9.3.2 SPI0 主模式控制寄存器 0 SPI0CS0 (偏移: 04h)

| 比特   | 名称     | 属性  | 复位值 | 描述  |
|------|--------|-----|-----|---|
| 31:7 | RSV    | -   | -   | 保留  |
| 6    | SSN0   | R/W | 0   | SPI0 主模式下, CS0 对应 Master 模式下, 如果 SSNSEN 为 1, 软件可以通过此位控制 SSN 输出电平<br>1: SSN 输出低电平<br>0: SSN 输出高电平  |
| 5:3  | BAUD0  | R/W | 001 | SPI0 主模式下, CS0 对应 Master 模式波特率配置位:<br>000: fPCLK/2<br>001: fPCLK/4<br>010: fPCLK/8<br>011: fPCLK/16<br>100: fPCLK/32<br>101: fPCLK/64<br>110: fPCLK/128<br>111: fPCLK/256<br>当通信正在进行的时候, 不能修改这些位。 |
| 2    | LSBF0  | R/W | 0   | SPI0 主模式下, CS0 对应帧格式 (Frame format)<br>0: 先发送 MSB<br>1: 先发送 LSB<br>注: 当通信在进行时不能改变该位的值。  |
| 1    | CPHOL0 | R/W | 0   | SPI0 主模式下, CS0 对应时钟极性选择。<br>1: 串行时钟停止在高电平<br>0: 串行时钟停止在低电平<br>注: 当通信在进行时不能改变该位的值。<br>注: 当 SSN 为低时不能改变该位的值   |
| 0    | CPHA0  | R/W | 0   | SPI0 主模式下, CS0 对应时钟相位选择:<br>1: 第二个时钟边沿是第一个捕捉边沿<br>0: 第一个时钟边沿是第一个捕捉边沿<br>注: 当通信在进行时不能改变该位的值。   |

### 9.3.3 SPI0 主模式控制寄存器 1 SPI0CS1 (偏移: 08h)

| 比特   | 名称   | 属性  | 复位值 | 描述   |
|------|------|-----|-----|--|
| 31:7 | RSV  | -   | -   | 保留   |
| 6    | SSN1 | R/W | 0   | SPI0 主模式下, CS1 对应 Master 模式下, 如果 SSNSEN 为 1, 软件可以通过此位控制 SSN 输出电平<br>1: SSN 输出低电平<br>0: SSN 输出高电平 |

| 比特  | 名称     | 属性  | 复位值 | 描述  |
|-----|--------|-----|-----|---|
| 5:3 | BAUD1  | R/W | 001 | SPI0 主模式下, CS1 对应 Master 模式波特率配置位:<br>000: fPCLK/2<br>001: fPCLK/4<br>010: fPCLK/8<br>011: fPCLK/16<br>100: fPCLK/32<br>101: fPCLK/64<br>110: fPCLK/128<br>111: fPCLK/256<br>当通信正在进行的时候, 不能修改这些位。 |
| 2   | LSBF1  | R/W | 0   | SPI0 主模式下, CS1 对应帧格式 (Frame format)<br>0: 先发送 MSB<br>1: 先发送 LSB<br>注: 当通信在进行时不能改变该位的值。  |
| 1   | CPHOL1 | R/W | 0   | SPI0 主模式下, CS1 对应时钟极性选择。<br>1: 串行时钟停止在高电平<br>0: 串行时钟停止在低电平<br>注: 当通信在进行时不能改变该位的值, 当 SSN 为低时不能改变该位的值   |
| 0   | CPHA1  | R/W | 0   | SPI0 主模式下, CS1 对应时钟相位选择:<br>1: 第二个时钟边沿是第一个捕捉边沿<br>0: 第一个时钟边沿是第一个捕捉边沿<br>注: 当通信在进行时不能改变该位的值。   |

### 9.3.4 SPI0 过程控制寄存器 SPI0OPCR (偏移: 14h)

| 比特   | 名称    | 属性  | 复位值 | 描述  |
|------|-------|-----|-----|---|
| 31:4 | RSV   | -   | -   | 保留 读为 0                                     |
| 3    | TXBFC | W1C | 0   | Transmit Buffer Clear, 软件写 1 清除发送缓存, 写 0 无效 |
| 2    | RXBFC | W1C | 0   | Receive Buffer Clear, 软件写 1 清除接收缓存, 写 0 无效  |
| 1    | MERRC | W1C | 0   | Master Error Clear, 软件写 1 清除 SPIIF.MERR 寄存器 |
| 0    | SERRC | W1C | 0   | Slave Error Clear, 软件写 1 清除 SPIIF.SERR 寄存器  |

### 9.3.5 SPI0 中断控制寄存器 SPI0IE (偏移: 18h)

| 比特   | 名称    | 属性  | 复位值 | 描述                    |
|------|-------|-----|-----|-----------------------|
| 31:9 | RSV   | -   | -   | 保留 读为 0               |
| 8    | RNFIE | R/W | 0   | Rx Fifo Full 中断使能     |
| 7    | TNFIE | R/W | 0   | Tx Fifo Not Full 中断使能 |

| 比特 | 名称      | 属性  | 复位值 | 描述                   |
|----|---------|-----|-----|----------------------|
| 6  | MERRIE  | R/W | 0   | Master Error 中断使能    |
| 5  | SERRIE  | R/W | 0   | Slave Error 中断使能     |
| 4  | RXCOLIE | R/W | 0   | 接收缓存溢出中断使能, 软件写 1 清零 |
| 3  | TXCOLIE | R/W | 0   | 发送缓存溢出中断使能, 软件写 1 清零 |
| 2  | IDLEIE  | R/W | 0   | SPI0 空闲标志中断使能        |
| 1  | TXBEIE  | R/W | 0   | TX Buffer Empty 中断使能 |
| 0  | RXBFIE  | R/W | 0   | RX Buffer 中断使能       |

### 9.3.6 SPI0 中断标志寄存器 SPI0IF (偏移: 1Ch)

| 比特   | 名称    | 属性  | 复位值 | 描述  |
|------|-------|-----|-----|---|
| 31:9 | RSV   | -   | -   | 保留 读为 0   |
| 8    | RNF   | R   | 0   | Spi Rx Fifo Full<br>1: SPI0 Rx Fifo 满<br>0: SPI0 Rx Fifo 未滿     |
| 7    | TNF   | R   | 1   | Spi Tx Fifo Not Full<br>1: SPI0 Tx Fifo 未滿<br>0: SPI0 Tx Fifo 滿 |
| 6    | MERR  | R   | 0   | Master Error 标志<br>当 Master 下传输未滿 8 位 SSN 就被拉高时,<br>MERR 置位     |
| 5    | SERR  | R   | 0   | Slave Error 标志<br>当 Slave 下传输未滿 8 位 SSN 就被拉高时,<br>SERR 置位       |
| 4    | RXCOL | R/W | 0   | 接收缓存溢出, 软件写 1 清零  |
| 3    | TXCOL | R/W | 0   | 发送缓存溢出, 软件写 1 清零  |
| 2    | IDLE  | R   | 1   | SPI0 空闲标志, 只读<br>1: SPI0 传输空闲<br>0: SPI0 传输进行中                  |
| 1    | TXBE  | R   | 1   | TX Buffer Empty 标志位<br>1: 发送缓存空, 软件写 TXBUF 清零<br>0: 发送缓存非空      |
| 0    | RXBF  | R   | 0   | RX Buffer 非空标志位<br>1: 接收缓存非空<br>0: 接收缓存空                        |

### 9.3.7 SPI0 发送缓存寄存器 SPI0TXBUF (偏移: 20h)

| 比特   | 名称    | 属性 | 复位值 | 描述   |
|------|-------|----|-----|--|
| 31:8 | RSV   | -  | -   | 保留 读为 0  |
| 7:0  | TXBUF | W  | 0   | SPI0 发送缓存, 发送 FIFO 入口地址。此 IP 一共含有 8 个 Byte 的发送 FIFO, 写此地址, 将要发送的数据写入 FIFO 中。 |

### 9.3.8 SPI0 接收缓存寄存器 SPI0RXBUF (偏移: 24h)

| 比特   | 名称    | 属性 | 复位值 | 描述   |
|------|-------|----|-----|--|
| 31:8 | RSV   | -  | -   | 保留 读为 0  |
| 7:0  | RXBUF | R  | 0   | SPI0 接收缓存, 接收 FIFO 入口地址。此 IP 一共含有 8 个 Byte 的接收 FIFO, 读此地址, 将接收的数据从 FIFO 中读取出来。 |

### 9.3.9 SPI0 DMA 接收设置寄存器 DMA\_SPI0RX\_LEV (偏移: 28h)

| 比特   | 名称         | 属性  | 复位值 | 描述  |
|------|------------|-----|-----|---|
| 31:3 | RSV        | -   | -   | 保留 读为 0   |
| 2:0  | DMA_RX_LEV | R/W | 0   | SPI0 接收 FIFO DMA 请求设置。<br>当 RX FIFO 中的数据个数大于此寄存器设置值时, 产生 DMA RX 请求。 |

### 9.3.10 SPI0 DMA 发送设置寄存器 DMA\_SPI0TX\_LEV (偏移: 2Ch)

| 比特   | 名称         | 属性  | 复位值 | 描述  |
|------|------------|-----|-----|---|
| 31:3 | RSV        | -   | -   | 保留 读为 0   |
| 2:0  | DMA_TX_LEV | R/W | 0   | SPI0 发送 FIFO DMA 请求设置。<br>当 TX FIFO 中的数据个数小于此寄存器设置值时, 产生 DMA TX 请求。 |

## 9.4 接口时序

为了兼容不同的 SPI 外设, SPI0 串行时钟的时序可以通过时钟相位选择位 (SPI0CSx.CPHA) 和时钟极性选择位 (SPI0CSx.CPOL) 设置产生 4 种不同组合。为保证数据正确传输, 主从器件的时序配置必需一致。

当处于从器件模式或 SPI0 系统使能位 (SPI0CR.SPIEN) 位为 0 时, SPI0 的 SCK 引脚无串行时钟输出。

### 9.4.1 CPHA=0

CPHA=0 时, SPI0 模块在串行时钟的第一个跳变沿采样数据, 即:

若 CPOL=1, 在串行时钟的下降沿采样数据;

若 CPOL=0, 在串行时钟的上升沿采样数据。如下图所示:

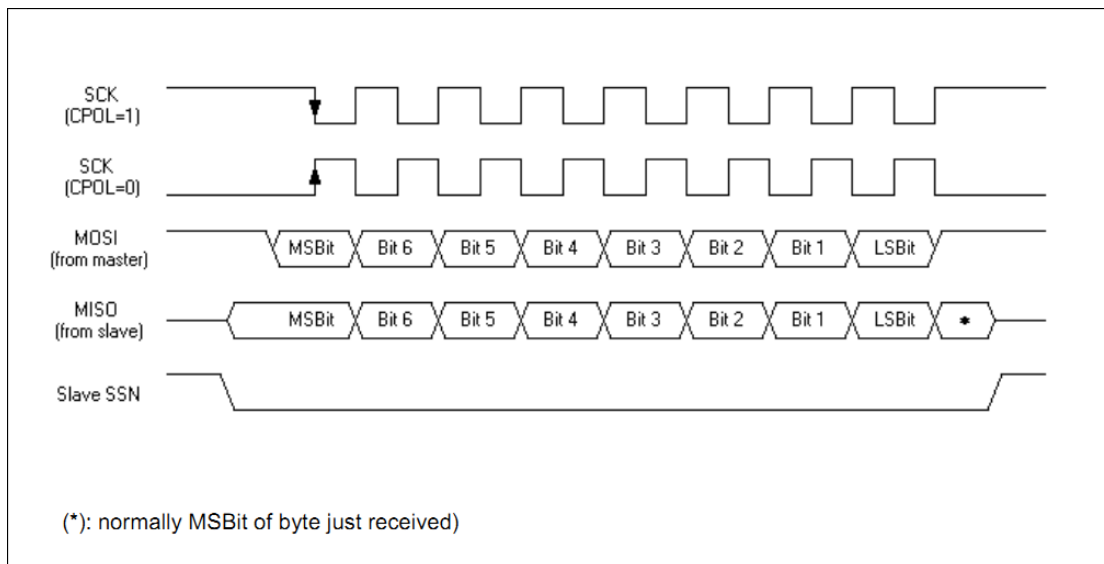


图 9-1：SPI0 数据/时钟时序图 (CPHA=0)

### 9.4.2 CPHA=1

CPHA=1 时，SPI0 模块在串行时钟的第二个跳变沿采样数据，即：

若 CPOL=1，在串行时钟的上升沿采样数据；

若 CPOL=0，在串行时钟的下降沿采样数据。如下图所示：

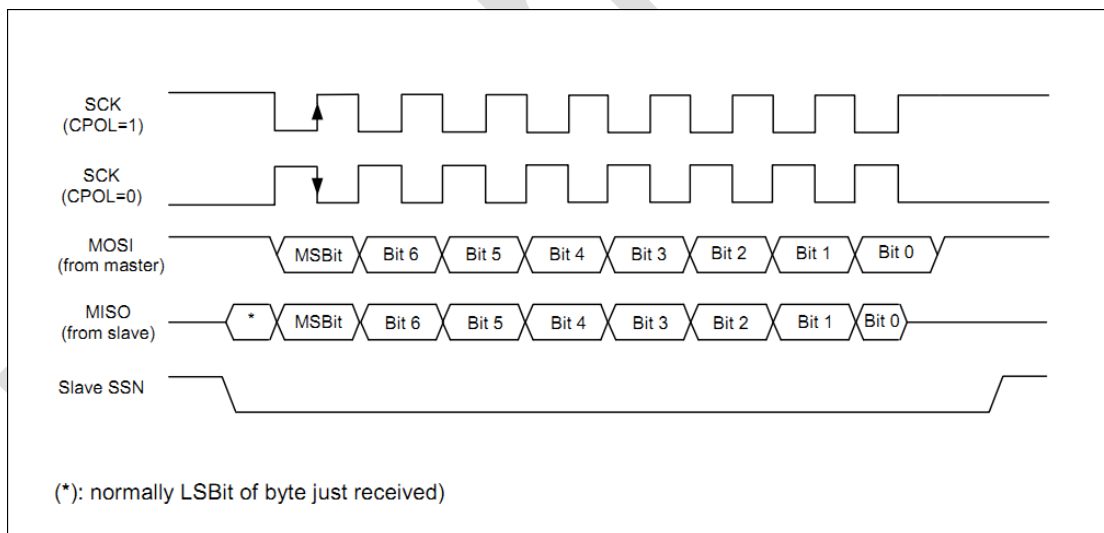


图 9-2：SPI0 数据/时钟时序图 (CPHA=1)

### 9.4.3 从器件 SSN

若 SPI0 为从器件，则 CPHA=0 时，SSN 引脚必须在每字节数据传输后拉高，以便可以拉低启动下一字节传输，并避免产生写冲突错误。如下图所示：

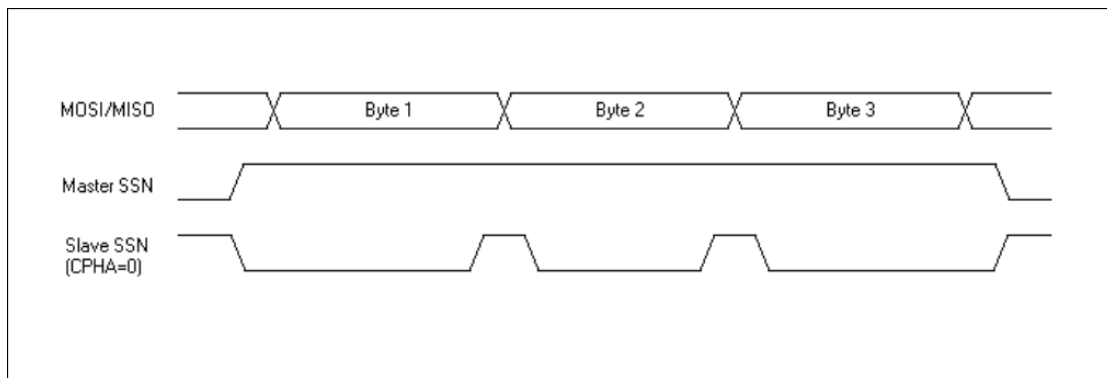


图 9-3: SPI0 SSN 时序图 (CPHA=0)

CPHA=1 时，从器件的 SSN 引脚可以在连续数据传输时一直为低，如下图所示：

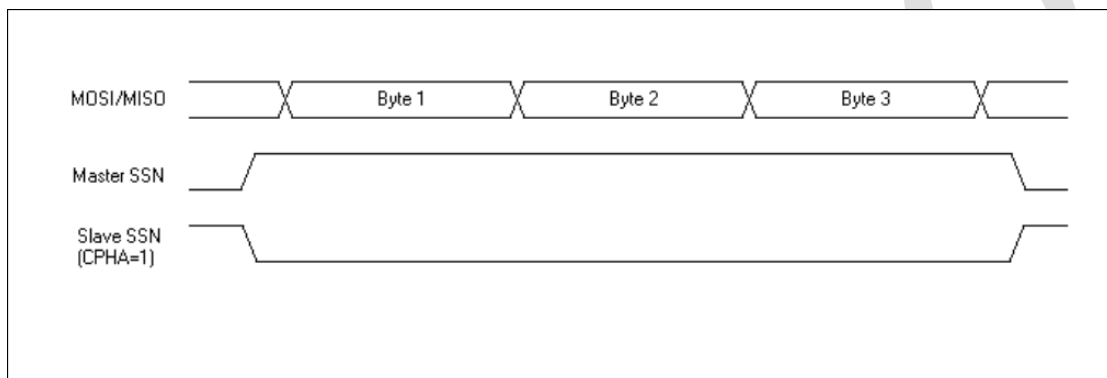


图 9-4: SPI0 SSN 时序图 (CPHA=1)

## 9.5 使用流程

SPI0 两种引脚搭配方式：

| 信号描述功能位 | SPI0 配置 1  | SPI0 配置 2  |
|---------|------------|------------|
| CS      | SPI0_CS0   | SPI0_CS1   |
| MISO    | SPI0_MISO0 | SPI0_MISO1 |
| MOSI    | SPI0_MOSI  | SPI0_MOSI  |
| CLK     | SPI0_SCK   | SPI0_SCK   |

注：引脚 SPI0\_CS1 和 SPI0\_MISO1 搭配使用，SPI0\_CS0 与 SPI0\_MISO0 搭配使用。

### 9.5.1 初始化程序

1. 配置 SPI0CR.MM 位，设置主从模式。
2. 配置 SPI0CR.SSNM 位，设置 SSN 控制模式。
3. 配置 SPI0CR.SSNSSEN 位，设置 SSN 输出由软件还是硬件控制。
4. 配置 SPI0CSx.SSNx 位和 SPI0CSx.LSBFx 位，设置 SSN 输出电平和帧格式。
5. 配置 SPI0CSx.CPHAx 位和 SPI0CSx.CPOLx 位，以设置串行时钟相位和极性。
6. 配置 SPI0CSx.BAUDx[2:0]位，以设置串行时钟波特率（若为从器件模式则不用设置，串行时钟速率由主器件决定）。需要时，配置中断，SPI0IE 和 SPI0IF 位。



## 7. 配置 SPI0CR.SPIEN, 使能 SPI0

### 9.5.2 发送流程

- 主器件发送流程:  
配置 SPI0CSx.SSNx 拉低 SSN 引脚启动传输, 等待 SPI0IF.TXBE 置位, 将数据写入 SPI0TXBUF 寄存器, 等待 SPI0IF.IDLE 置位发送完成, 传输完后将 SSN 拉高
- 从器件发送流程:  
等待 SPI0IF.TXBE 置位, 将数据写入 SPI0TXBUF 寄存器

### 9.5.3 接收流程

- 主器件接收流程:  
等待 SPI0IF.RXBF 置位, 读取 SPI0RXBUF 寄存器数据完成数据接收
- 从器件接收流程:  
等待 SPI0IF.RXBF 置位, 读取 SPI0RXBUF 寄存器数据完成数据接收

### 9.5.4 SPI0 DMA 发送流程

1. 配置 SPI0 DMA 发送设置寄存器 DMA\_SPI0TX\_LEV, 设置产生 DMA TX 请求的 FIFO 数据个数。
2. 使能 SPI0CR.DMA\_TX\_EN 位, 使能 DMA TX 请求。
3. 配置 SARx 寄存器, 指定源地址。
4. 配置 DARx 寄存器, 指定目标地址。
5. 配置 CTLx 寄存器。有需要时使能中断, 设置传输数据宽度和大小, 选择流控类型和传输模式, 注意 SRAM 和 FLASH 没有握手信号, 在 TT\_FC 中将 SRAM 和 FLASH 认定为存储器选择地址是否递增/递减, 选择 burst 传输长度。
6. 配置 CFGx 寄存器。选择通道优先级, 设置是否自动重启传输, 选定硬件握手信号, 将 HS\_SEL 域设为 0。
7. 配置 MaskBlock 寄存器, 开启对应通道的中断。
8. 配置 DmaCfgReg 寄存器使能 DMAC。
9. 配置 ChEnReg 寄存器使能通道启动 DMA 搬移。
10. 等待中断或查询 ChEnReg。
11. 清除中断完成传输。

### 9.5.5 SPI0 DMA 接收流程

1. 配置 SPI0 DMA 发送设置寄存器 DMA\_SPI0RX\_LEV，设置产生 DMA RX 请求的 FIFO 数据个数。
2. 使能 SPI0CR.DMA\_RX\_EN 位，使能 DMA RX 请求。
3. 配置 SARx 寄存器，指定源地址。
4. 配置 DARx 寄存器，指定目标地址。
5. 配置 CTLx 寄存器。有需要时使能中断，设置传输数据宽度和大小，选择流控类型和传输模式，注意 SRAM 和 FLASH 没有握手信号，在 TT\_FC 中将 SRAM 和 FLASH 认定为储存器选择地址是否递增/递减，选择 burst 传输长度。
6. 配置 CFGx 寄存器。选择通道优先级，设置是否自动重启传输，选定硬件握手信号，将 HS\_SEL 域设为 0。
7. 配置 MaskBlock 寄存器，开启对应通道的中断。
8. 配置 DmaCfgReg 寄存器使能 DMAC。
9. 配置 ChEnReg 寄存器使能通道启动 DMA 搬移。
10. 等待中断或查询 ChEnReg。
11. 清除中断完成传输。

# 10 SPI1

## 10.1 概述

串行外设接口（Serial Peripheral Interface, SPI）是外部设备通过单线交换数据的串行同步通讯手段。芯片提供了一个 SPI1 接口模块，可配置为主设备或从设备，实现与外部的 SPI 通信。

## 10.2 主要特性

- 全双工或半双工单数据线串行同步收发
- 主从模式
- 可编程时钟极性和相位（支持模式 0、1、2、3）
- 可编程比特速率
- 从模式最大频率为  $F_{sys}/2$
- 传输结束中断标志
- 写冲突错标志
- 主模式错误检测、保护和中断标志
- 支持 DMA

## 10.3 寄存器描述

SPI1 寄存器基地址：0x40006000

表 10-1: SPI1 寄存器列表

| 偏置   | 名称           | 描述               |
|------|--------------|------------------|
| 0x0  | SPI1CR       | SPI1 配置寄存器       |
| 0x4  | SPI1CS0      | SPI1 主模式控制寄存器 0  |
| 0x14 | SPI1OPCR     | SPI1 过程控制寄存器     |
| 0x18 | SPI1IE       | SPI1 中断控制寄存器     |
| 0x1C | SPI1IF       | SPI1 中断标志寄存器     |
| 0x20 | SPI1TXBUF    | SPI1 接收缓存寄存器     |
| 0x24 | SPI1RXBUF    | SPI1 接收缓存寄存器     |
| 0x28 | SPI1DMARXLEV | SPI1 DMA 接收设置寄存器 |
| 0x2c | SPI1DMATXLEV | SPI1 DMA 发送设置寄存器 |

### 10.3.1 SPI1 配置寄存器 SPI1CR (偏移: 00h)

| 比特    | 名称        | 属性  | 复位值 | 描述   |
|-------|-----------|-----|-----|--|
| 31:14 | RSV       | -   | -   | 保留   |
| 13    | DMA_TX_EN | R/W | 0   | Dma Tx 使能<br>1: 使能 DMA TX 请求<br>0: 关闭 DMA TX 请求  |
| 12    | DMA_RX_EN | R/W | 0   | Dma Rx 使能<br>1: 使能 DMA RX 请求<br>0: 关闭 DMA RX 请求  |
| 11    | FLTEN     | R/W | 1   | Slave 输入管脚滤波使能 (SSN/SCK/MOSI)<br>1: 使能 4ns 滤波<br>0: 不滤波  |
| 10    | SSNM      | R/W | 0   | Master 模式下 SSN 控制模式选择<br>1: 每发送完 8bit 后 Master 拉高 SSN, 维持高电平时间由 WAIT 寄存器控制<br>0: 每发送完 8bit 后 Master 保持 SSN 为低, 维持低电平时间由 WAIT 寄存器控制 |
| 9     | TXO_AC    | R/W | 1   | TXONLY 硬件自动清空的使能<br>1: TXONLY 硬件自动清零有效, 软件使能 TXO 后, 等待发送完毕后, 硬件清零<br>0: 关闭 TXONLY 硬件自动清零   |
| 8     | TXO       | R/W | 0   | TXONLY 控制位<br>1: 启动 Master 的单发送模式<br>0: 关闭单发送模式  |
| 7     | MSPA      | R/W | 0   | Master Sampling Position Adjustment, Master 对 MISO 信号的采样位置调整, 用于高速通信时补偿 PCB 走线延迟<br>1: 采样点延迟半个 SCK 周期<br>0: 不调整                    |
| 6     | SSPA      | R/W | 0   | Slave Sending Position Adjustment, Slave MISO 发送位置调整<br>1: 提前半个 SCK 周期发送<br>0: 不调整   |
| 5     | MM        | R/W | 1   | Master/Slave 模式选择。<br>1: Master 模式<br>0: Slave 模式  |
| 4:3   | WAIT      | R/W | 0   | Master 模式下, 每发完 8Bit 后加入至少 (1+WAIT)个 SCK cycle 等待时间再传输下一个 8Bit 的数据   |
| 2     | RSV       | -   | -   | RFU: 未实现, 读为 0   |
| 1     | SSNSEN    | R/W | 0   | Master 模式下, 软件控制 SSN 使能<br>1: Master 模式下 SSN 输出由软件控制<br>0: Master 模式下 SSN 输出由硬件自动控制  |
| 0     | SPI1EN    | R/W | 0   | SPI1 使能。采用关闭时钟的方式来关闭使能。<br>1: 使能 SPI1<br>0: 关闭 SPI1, 清空发送接收缓存  |

### 10.3.2 SPI1 主模式控制寄存器 0 SPI1CS0 (偏移: 04h)

| 比特   | 名称     | 属性  | 复位值 | 描述  |
|------|--------|-----|-----|---|
| 31:7 | RSV    | -   | -   | 保留  |
| 6    | SSN0   | R/W | 0   | SPI1 主模式下, CS0 对应 Master 模式下, 如果 SSNSEN 为 1, 软件可以通过此位控制 SSN 输出电平<br>1: SSN 输出低电平<br>0: SSN 输出高电平  |
| 5:3  | BAUD0  | R/W | 001 | SPI1 主模式下, CS0 对应 Master 模式波特率配置位:<br>000: fPCLK/2<br>001: fPCLK/4<br>010: fPCLK/8<br>011: fPCLK/16<br>100: fPCLK/32<br>101: fPCLK/64<br>110: fPCLK/128<br>111: fPCLK/256<br>当通信正在进行的时候, 不能修改这些位。 |
| 2    | LSBF0  | R/W | 0   | SPI1 主模式下, CS0 对应帧格式 (Frame format)<br>0: 先发送 MSB<br>1: 先发送 LSB<br>注: 当通信在进行时不能改变该位的值。  |
| 1    | CPHOL0 | R/W | 0   | SPI1 主模式下, CS0 对应时钟极性选择。<br>1: 串行时钟停止在高电平<br>0: 串行时钟停止在低电平<br>注: 当通信在进行时不能改变该位的值。<br>注: 当 SSN 为低时不能改变该位的值   |
| 0    | CPHA0  | R/W | 0   | SPI1 主模式下, CS0 对应时钟相位选择:<br>1: 第二个时钟边沿是第一个捕捉边沿<br>0: 第一个时钟边沿是第一个捕捉边沿<br>注: 当通信在进行时不能改变该位的值。   |

### 10.3.3 SPI1 过程控制寄存器 SPI1OPCR (偏移: 14h)

| 比特   | 名称    | 属性  | 复位值 | 描述  |
|------|-------|-----|-----|---|
| 31:4 | RSV   | -   | -   | 保留 读为 0                                     |
| 3    | TXBFC | W1C | 0   | Transmit Buffer Clear, 软件写 1 清除发送缓存, 写 0 无效 |
| 2    | RXBFC | W1C | 0   | Receive Buffer Clear, 软件写 1 清除接收缓存, 写 0 无效  |
| 1    | MERRC | W1C | 0   | Master Error Clear, 软件写 1 清除 SPIIF.MERR 寄存器 |
| 0    | SERRC | W1C | 0   | Slave Error Clear, 软件写 1 清除 SPIIF.SERR 寄存器  |

### 10.3.4 SPI1 中断控制寄存器 SPI1IE (偏移: 18h)

| 比特   | 名称      | 属性  | 复位值 | 描述                    |
|------|---------|-----|-----|-----------------------|
| 31:9 | RSV     | -   | -   | 保留 读为 0               |
| 8    | RNFIE   | R/W | 0   | Rx Fifo Full 中断使能     |
| 7    | TNFIE   | R/W | 0   | Tx Fifo Not Full 中断使能 |
| 6    | MERRIE  | R/W | 0   | Master Error 中断使能     |
| 5    | SERRIE  | R/W | 0   | Slave Error 中断使能      |
| 4    | RXCOLIE | R/W | 0   | 接收缓存溢出中断使能, 软件写 1 清零  |
| 3    | TXCOLIE | R/W | 0   | 发送缓存溢出中断使能, 软件写 1 清零  |
| 2    | IDLEIE  | R/W | 0   | SPI1 空闲标志中断使能         |
| 1    | TXBEIE  | R/W | 0   | TX Buffer Empty 中断使能  |
| 0    | RXBFIE  | R/W | 0   | RX Buffer 中断使能        |

### 10.3.5 SPI1 中断标志寄存器 SPI1IF (偏移: 1Ch)

| 比特   | 名称    | 属性  | 复位值 | 描述  |
|------|-------|-----|-----|---|
| 31:9 | RSV   | -   | -   | 保留 读为 0   |
| 8    | RNF   | R   | 0   | Spi RX Fifo Full<br>1: SPI1 RX Fifo 满<br>0: SPI1 RX Fifo 未滿     |
| 7    | TNF   | R   | 1   | Spi TX Fifo Not Full<br>1: SPI1 TX Fifo 未滿<br>0: SPI1 TX Fifo 滿 |
| 6    | MERR  | R   | 0   | Master Error 标志<br>当 Master 下传输未滿 8 位 SSN 就被拉高时,<br>MERR 置位     |
| 5    | SERR  | R   | 0   | Slave Error 标志<br>当 Slave 下传输未滿 8 位 SSN 就被拉高时,<br>SERR 置位       |
| 4    | RXCOL | R/W | 0   | 接收缓存溢出, 软件写 1 清零  |
| 3    | TXCOL | R/W | 0   | 发送缓存溢出, 软件写 1 清零  |
| 2    | IDLE  | R   | 1   | SPI1 空闲标志, 只读<br>1: SPI1 传输空闲<br>0: SPI1 传输进行中                  |
| 1    | TXBE  | R   | 1   | TX Buffer Empty 标志位<br>1: 发送缓存空, 软件写 TXBUF 清零<br>0: 发送缓存非空      |
| 0    | RXBF  | R   | 0   | RX Buffer 非空标志位<br>1: 接收缓存非空<br>0: 接收缓存空                        |

### 10.3.6 SPI1 发送缓存寄存器 SPI1TXBUF (偏移: 20h)

| 比特   | 名称    | 属性 | 复位值 | 描述   |
|------|-------|----|-----|--|
| 31:8 | RSV   | -  | -   | 保留 读为 0  |
| 7:0  | TXBUF | W  | 0   | SPI1 发送缓存, 发送 FIFO 入口地址。此 IP 一共含有 8 个 Byte 的发送 FIFO, 写此地址, 将要发送的数据写入 FIFO 中。 |

### 10.3.7 SPI1 接收缓存寄存器 SPI1RXBUF (偏移: 24h)

| 比特   | 名称    | 属性 | 复位值 | 描述   |
|------|-------|----|-----|--|
| 31:8 | RSV   | -  | -   | 保留 读为 0  |
| 7:0  | RXBUF | R  | 0   | SPI1 接收缓存, 接收 FIFO 入口地址。此 IP 一共含有 8 个 Byte 的接收 FIFO, 读此地址, 将接收的数据从 FIFO 中读取出来。 |

### 10.3.8 SPI1 DMA 接收设置寄存器 DMA\_SPI1RX\_LEV (偏移: 28h)

| 比特   | 名称         | 属性  | 复位值 | 描述  |
|------|------------|-----|-----|---|
| 31:3 | RSV        | -   | -   | 保留 读为 0   |
| 2:0  | DMA_RX_LEV | R/W | 0   | SPI1 接收 FIFO DMA 请求设置。<br>当 RX FIFO 中的数据个数大于此寄存器设置值时, 产生 DMA RX 请求。 |

### 10.3.9 SPI1 DMA 发送设置寄存器 DMA\_SPI1TX\_LEV (偏移: 2Ch)

| 比特   | 名称         | 属性  | 复位值 | 描述  |
|------|------------|-----|-----|---|
| 31:3 | RSV        | -   | -   | 保留 读为 0   |
| 2:0  | DMA_TX_LEV | R/W | 0   | SPI1 发送 FIFO DMA 请求设置。<br>当 TX FIFO 中的数据个数小于此寄存器设置值时, 产生 DMA TX 请求。 |

## 10.4 接口时序

为了兼容不同的 SPI 外设, SPI1 串行时钟的时序可以通过时钟相位选择位 (SPI1CSx.CPHA) 和时钟极性选择位 (SPI1CSx.CPOL) 设置产生 4 种不同组合。为保证数据正确传输, 主从器件的时序配置必需一致。

当处于从器件模式或 SPI1 系统使能位 (SPI1CR.SPIEN) 位为 0 时, SPI1 的 SCK 引脚无串行时钟输出。

### 10.4.1 CPHA=0

CPHA=0 时，SPI1 模块在串行时钟的第一个跳变沿采样数据，即：

若 CPOL=1，在串行时钟的下降沿采样数据；

若 CPOL=0，在串行时钟的上升沿采样数据。如下图所示：

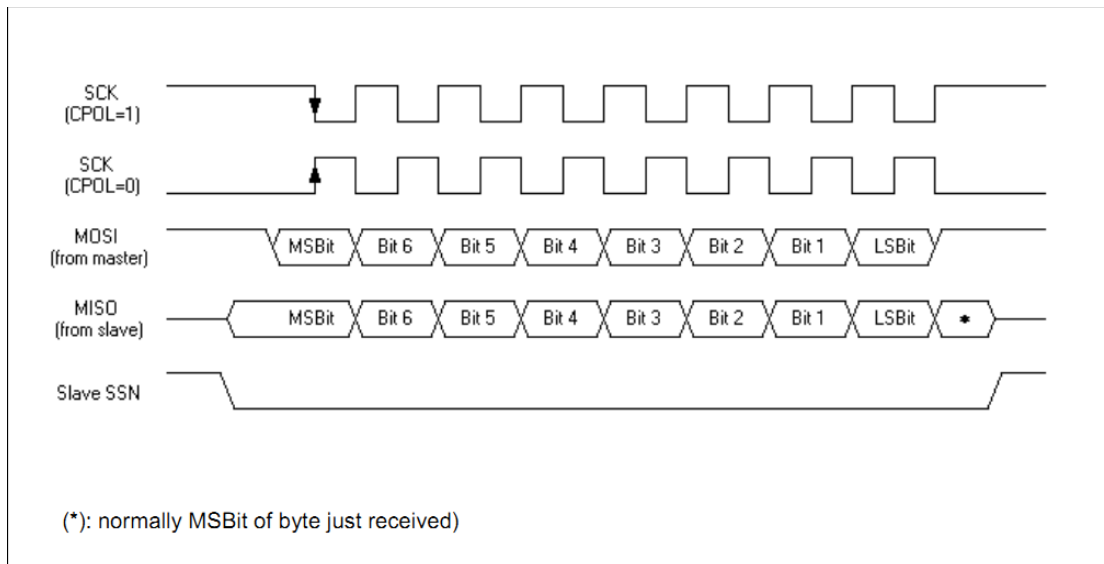


图 10-1：SPI1 数据/时钟时序图（CPHA=0）

### 10.4.2 CPHA=1

CPHA=1 时，SPI1 模块在串行时钟的第二个跳变沿采样数据，即：

若 CPOL=1，在串行时钟的上升沿采样数据；

若 CPOL=0，在串行时钟的下降沿采样数据。如下图所示：

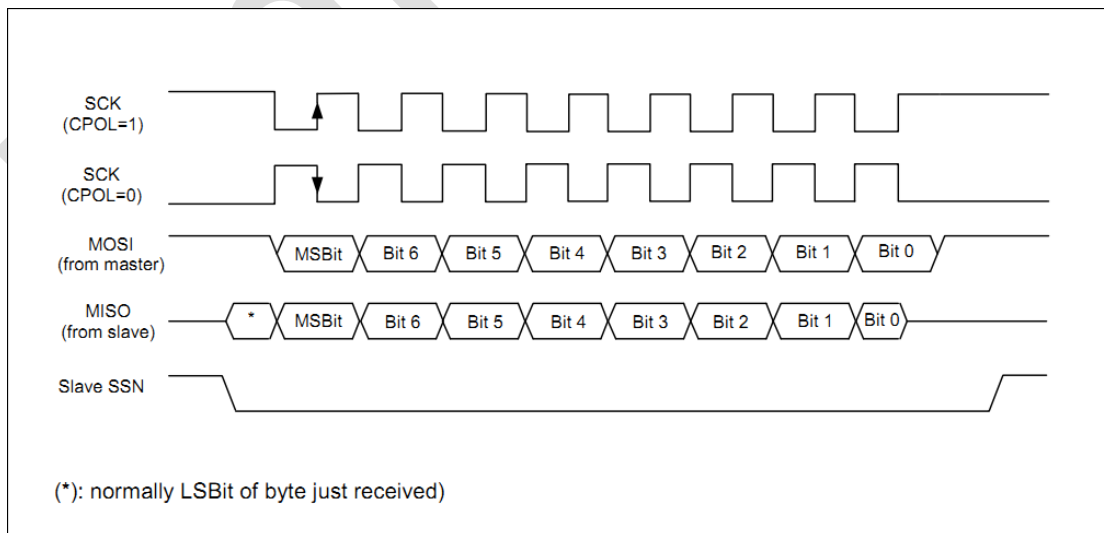


图 10-2：SPI1 数据/时钟时序图（CPHA=1）



### 10.4.3 从器件 SSN

若 SPI1 为从器件，则 CPHA=0 时，SSN 引脚必须在每字节数据传输后拉高，以便可以拉低启动下一字节传输，并避免产生写冲突错误。如下图所示：

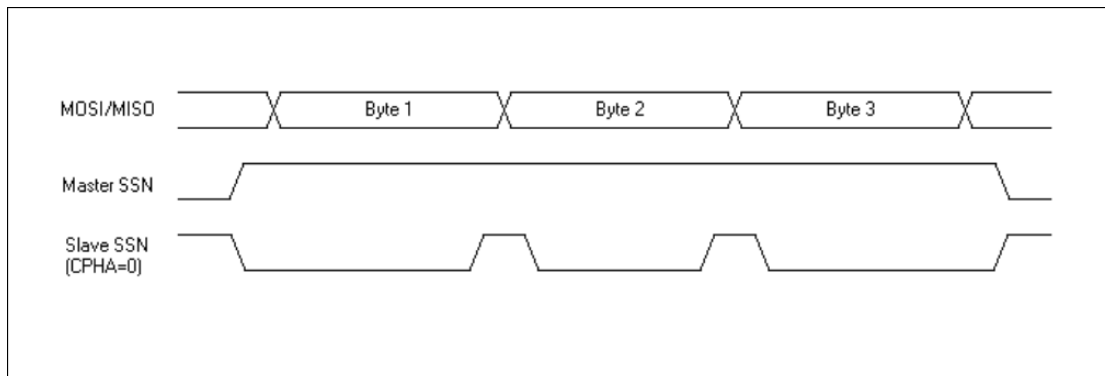


图 10-3: SPI1 SSN 时序图 (CPHA=0)

CPHA=1 时，从器件的 SSN 引脚可以在连续数据传输时一直为低，如下图所示：

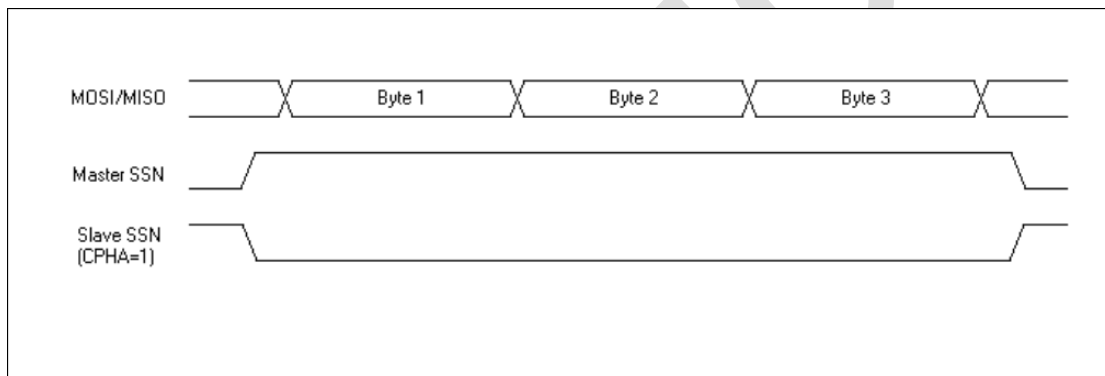


图 10-4: SPI1 SSN 时序图 (CPHA=1)

## 10.5 使用流程

### 10.5.1 初始化程序

1. 配置 SPI1CR.MM 位，设置主从模式。
2. 配置 SPI1CR.SSNM 位，设置 SSN 控制模式。
3. 配置 SPI1CR.SSENSE 位，设置 SSN 输出由软件还是硬件控制。
4. 配置 SPI1CSx.SSNx 位和 SPI1CSx.LSBFx 位，设置 SSN 输出电平和帧格式。
5. 配置 SPI1CSx.CPHAx 位和 SPI1CSx.CPOLx 位，以设置串行时钟相位和极性（主从器件需一致）。
6. 配置 SPI1CSx.BAUDx[2:0] 位，以设置串行时钟波特率（若为从器件模式则不用设置，串行时钟速率由主器件决定）。需要时，配置中断，SPI1IE 和 SPI1IF 位。

7. 配置 SPI1CR.SPIEN，使能 SPI1。

## 10.5.2 发送流程

- 主器件发送流程：  
配置 SPI1CSx.SSNx 拉低 SSN 引脚启动传输，等待 SPI1IF.TXBE 置位，将数据写入 SPI1TXBUF 寄存器，等待 SPI1IF.IDLE 置位发送完成，传输后将 SSN 拉高
- 从器件发送流程：  
等待 SPI1IF.TXBE 置位，将数据写入 SPI1TXBUF 寄存器

## 10.5.3 接收流程

- 主器件接收流程：  
等待 SPI1IF.RXBF 置位，读取 SPI1RXBUF 寄存器数据完成数据接
- 从器件接收流程：  
等待 SPI1IF.RXBF 置位，读取 SPI1RXBUF 寄存器数据完成数据接

## 10.5.4 SPI1 DMA 发送流程

1. 配置 SPI1 DMA 发送设置寄存器 DMA\_SPI1TX\_LEV，设置产生 DMA TX 请求的 FIFO 数据个数。
2. 使能 SPI1CR.DMA\_TX\_EN 位，使能 DMA TX 请求。
3. 配置 SARx 寄存器，指定源地址。
4. 配置 DARx 寄存器，指定目标地址。
5. 配置 CTLx 寄存器。有需要时使能中断，设置传输数据宽度和大小，选择流控类型和传输模式，注意 SRAM 和 FLASH 没有握手信号，在 TT\_FC 中将 SRAM 和 FLASH 认定为存储器选择地址是否递增/递减，选择 burst 传输长度。
6. 配置 CFGx 寄存器。选择通道优先级，设置是否自动重启传输，选定硬件握手信号，将 HS\_SEL 域设为 0。
7. 配置 MaskBlock 寄存器，开启对应通道的中断。
8. 配置 DmaCfgReg 寄存器使能 DMAC。
9. 配置 ChEnReg 寄存器使能通道启动 DMA 搬移。
10. 等待中断或查询 ChEnReg。
11. 清除中断完成传输。

### 10.5.5 SPI1 DMA 接收流程

1. 配置 SPI1 DMA 发送设置寄存器 DMA\_SPI1RX\_LEV，设置产生 DMA RX 请求的 FIFO 数据个数。
2. 使能 SPI1CR.DMA\_RX\_EN 位，使能 DMA RX 请求。
3. 配置 SARx 寄存器，指定源地址。
4. 配置 DARx 寄存器，指定目标地址。
5. 配置 CTLx 寄存器。有需要时使能中断，设置传输数据宽度和大小，选择流控类型和传输模式，注意 SRAM 和 FLASH 没有握手信号，在 TT\_FC 中将 SRAM 和 FLASH 认定为储存器选择地址是否递增/递减，选择 burst 传输长度。
6. 配置 CFGx 寄存器。选择通道优先级，设置是否自动重启传输，选定硬件握手信号，将 HS\_SEL 域设为 0。
7. 配置 MaskBlock 寄存器，开启对应通道的中断。
8. 配置 DmaCfgReg 寄存器使能 DMAC。
9. 配置 ChEnReg 寄存器使能通道启动 DMA 搬移。
10. 等待中断或查询 ChEnReg。
11. 清除中断完成传输。

# 11 GTIMER

## 11.1 概述

有 4 个 32 位的通用定时/计数器 Timer，每个定时器都有自己独立的中断。这些 timer 可以有多种用途，包括测量输入信号的脉冲宽度（输入捕获），产生输出波形（PWM），计数器可以往上，往下，往上下三种计数方向，且计数值可以随时由软件读取。每个 TIMER 都有 3 种工作模式：free-running, cyclic, and single。芯片有 4 路 PWM 输出，有 2 路输入捕获，提供每次脉冲中断。

## 11.2 主要特性

- 4 个独立的 32 位递增，递减，递增/递减自动重载定时器，Timer0、Timer1、Timer2、Timer3。
- 4 个独立的 channel，其中 channel0 和 channel2 可配置为输入捕获或 PWM 输出，channel1 和 channel3 可配置为 PWM 输出。4 个 PWM 分别对应 Timer0、Timer1、Timer2、Timer3。
- 三种计数模式：free-running, cyclic, single
- 中断在以下几种情况产生：
  - UEV（Update Event）：counter 往上计数到溢出值，或往下计数到 0
  - 输入捕获
  - PWM 输出
  - 可编程预分频因子

## 11.3 寄存器描述

GTIMER 寄存器基地址：0x40000C00

表 11-1: GTIMER 寄存器列表

| 偏置   | 名称     | 描述             |
|------|--------|----------------|
| 0x00 | T0_ARR | Timer0 加载寄存器   |
| 0x04 | T0_CNT | Timer0 计数寄存器   |
| 0x08 | T0_CR  | Timer0 控制寄存器   |
| 0x0C | T0_IF  | Timer0 中断状态寄存器 |
| 0x10 | T0_CIF | Timer0 中断清除寄存器 |
| 0x14 | T1_ARR | Timer1 加载寄存器   |
| 0x18 | T1_CNT | Timer1 计数寄存器   |
| 0x1C | T1_CR  | Timer1 控制寄存器   |
| 0x20 | T1_IF  | Timer1 中断状态寄存器 |

| 偏置   | 名称      | 描述                          |
|------|---------|-----------------------------|
| 0x24 | T1_CIF  | Timer1 中断清除寄存器              |
| 0x28 | T2_ARR  | Timer2 加载寄存器                |
| 0x2C | T2_CNT  | Timer2 计数寄存器                |
| 0x30 | T2_CR   | Timer2 控制寄存器                |
| 0x34 | T2_IF   | Timer2 中断状态寄存器              |
| 0x38 | T2_CIF  | Timer2 中断清除寄存器              |
| 0x3C | T3_ARR  | Timer3 加载寄存器                |
| 0x40 | T3_CNT  | Timer3 计数寄存器                |
| 0x44 | T3_CR   | Timer3 控制寄存器                |
| 0x48 | T3_IF   | Timer3 中断状态寄存器              |
| 0x4C | T3_CIF  | Timer3 中断清除寄存器              |
| 0x50 | TIM_PSC | 定时器预分频寄存器                   |
| 0x54 | ICMODE  | 工作模式控制寄存器                   |
| 0x58 | CCR     | 输入捕获模式控制寄存器                 |
| 0x5C | CCIF    | 输入捕获模式中断状态寄存器               |
| 0x60 | C0_CR   | Channel0 输入捕获模式 counter 寄存器 |
| 0x64 | C2_CR   | Channel2 输入捕获模式 counter 寄存器 |
| 0x68 | PCR     | PWM 控制寄存器                   |
| 0x6C | CPIF    | PWM 中断状态寄存器                 |
| 0x70 | C0_PR   | PWM0 比较寄存器                  |
| 0x74 | C1_PR   | PWM1 比较寄存器                  |
| 0x78 | C2_PR   | PWM2 比较寄存器                  |
| 0x7C | C3_PR   | PWM3 比较寄存器                  |

### 11.3.1 Timer0 加载寄存器 T0\_ARR (偏移: 00h)

| 比特   | 名称   | 属性  | 复位值 | 描述  |
|------|------|-----|-----|---|
| 31:0 | LOAD | R/W | 0x0 | 定时器0加载值位[31:0]<br>定时器0的计数器加载值。在循环模式和单次模式，只有在UEV事件发生时才会把加载值加载到定时器。 |

### 11.3.2 Timer0 计数寄存器 T0\_CNT (偏移: 04h)

| 比特   | 名称     | 属性 | 复位值 | 描述                                      |
|------|--------|----|-----|---|
| 31:0 | T0_CNT | R  | 0x0 | 31到0位: 定时器0当前计数位[31:0]<br>定时器0的当前只读计数值。 |

### 11.3.3 Timer0 控制寄存器 T0\_CR (偏移: 08h)

| 比特   | 名称   | 属性  | 复位值   | 描述  |
|------|------|-----|-------|---|
| 31:6 | RSV  | -   | -     | 保留  |
| 5    | CMS  | R/W | 0     | 中心对齐模式选择位[1:0]<br>0: 边沿对齐模式。counter计数方向由DIR决定。<br>1: 中心对齐模式1。Counter往上计数和往下计数交替进行。  |
| 4    | DIR  | R/W | 0     | 方向控制位<br>0: counter向上计数<br>1: counter向下计数   |
| 3    | INTM | R/W | 0     | 定时器0 中断屏蔽位<br>0: 定时器0中断使能<br>1: 定时器0中断屏蔽  |
| 2:1  | MODE | R/W | 2'b00 | 模式选择位[1:0]<br>0x: 自由运行模式, 计数从4294967295 到0 (向下计数) 或从0到4294967295 (向上计数) 循环或先从0到4294967295再从4294967295到0 (中心对齐模式)。<br>10: 循环模式, 计数从初始计数值开始循环。<br>11: 单次模式, 计数从初始计数值到溢出值 (向上计数) 或到0 (向下计数), 然后停止。 |
| 0    | EN   | R/W | 0     | 定时器0 使能位<br>0: 计数器0关闭<br>1: 计数器0使能  |

### 11.3.4 Timer0 中断状态寄存器 T0\_IF (偏移: 0Ch)

| 比特   | 名称  | 属性 | 复位值 | 描述  |
|------|-----|----|-----|---|
| 31:1 | RSV | -  | -   | 保留  |
| 0    | INT | R  | 0   | 反映定时器0的中断状态<br>0: 中断无效。<br>1: 中断有效。<br>请写中断清除寄存器清除中断。 |

### 11.3.5 Timer0 中断清除寄存器 T0\_CIF (偏移: 10h)

| 比特   | 名称     | 属性 | 复位值 | 描述                                     |
|------|--------|----|-----|--|
| 31:0 | T0_CIF | W  | 0x0 | 对 Timer0 中断清除寄存器写任意值, 都会清除计数器 0 的中断输出。 |

### 11.3.6 Timer1 加载寄存器 T1\_ARR (偏移: 14h)

| 比特   | 名称   | 属性  | 复位值 | 描述  |
|------|------|-----|-----|---|
| 31:0 | LOAD | R/W | 0x0 | 定时器1加载值位[31:0]<br>定时器1的计数器加载值。在循环模式和单次模式, 写操作立即会把加载值加载到定时器。如果Timer 工作中更改计数模式成循环或者单次, 需要重写加载值, 否则会按照定时器会按照当前计数值继续计数。 |

### 11.3.7 Timer1 计数寄存器 T1\_CNT (偏移: 18h)

| 比特   | 名称     | 属性 | 复位值 | 描述   |
|------|--------|----|-----|--|
| 31:0 | T1_CNT | R  | 0x0 | 31 到0 位: 定时器1当前计数位[31:0]<br>定时器1 的当前只读计数值。 |

### 11.3.8 Timer1 控制寄存器 T1\_CR (偏移: 1Ch)

| 比特   | 名称   | 属性  | 复位值 | 描述   |
|------|------|-----|-----|--|
| 31:6 | RSV  | -   | -   | 保留   |
| 5    | CMS  | R/W | 0   | 中心对齐模式选择位[1:0]<br>0: 边沿对齐模式。counter计数方向由DIR决定。<br>1: 中心对齐模式1。Counter往上计数和往下计数交替进行。 |
| 4    | DIR  | R/W | 0   | 方向控制位<br>0: counter向上计数<br>1: counter向下计数  |
| 3    | INTM | R/W | 0   | 定时器1 中断屏蔽位<br>0: 定时器1 中断使能<br>1: 定时器1 中断屏蔽。  |

| 比特  | 名称   | 属性  | 复位值   | 描述  |
|-----|------|-----|-------|---|
| 2:1 | MODE | R/W | 2'b00 | 模式选择位[1:0]<br>0x: 自由运行模式, 计数从4294967295 到0 (向下计数) 或从0到4294967295 (向上计数) 循环或先从0到4294967295再从4294967295到0 (中心对齐模式)。<br>10: 循环模式, 计数从初始计数值开始循环。<br>11: 单次模式, 计数从初始计数值到溢出值 (向上计数) 或到0 (向下计数), 然后停止。 |
| 0   | EN   | R/W | 0     | 定时器1 使能位<br>0: 计数器1关闭<br>1: 计数器1使能  |

### 11.3.9 Timer1 中断状态寄存器 T1\_IF (偏移: 20h)

| 比特   | 名称  | 属性 | 复位值 | 描述  |
|------|-----|----|-----|---|
| 31:1 | RSV | -  | -   | 保留  |
| 0    | INT | R  | 0   | 反映定时器1的原始中断状态<br>0: 中断无效。<br>1: 中断有效。<br>请写中断清除寄存器清除中断。 |

### 11.3.10 Timer1 中断清除寄存器 T1\_CIF (偏移: 24h)

| 比特   | 名称     | 属性 | 复位值 | 描述                                     |
|------|--------|----|-----|--|
| 31:0 | T1_CIF | W  | 0x0 | 对 Timer1 中断清除寄存器写任意值, 都会清除计数器 1 的中断输出。 |

### 11.3.11 Timer2 加载寄存器 T2\_ARR (偏移: 28h)

| 比特   | 名称   | 属性  | 复位值 | 描述   |
|------|------|-----|-----|--|
| 31:0 | LOAD | R/W | 0x0 | 定时器2加载值位[31:0]<br>定时器2的计数器加载值。在循环模式和单次模式, 只有在UEV事件发生时才会把加载值加载到定时器。 |



### 11.3.12 Timer2 计数寄存器 T2\_CNT (偏移: 2Ch)

| 比特   | 名称     | 属性 | 复位值 | 描述                                       |
|------|--------|----|-----|--|
| 31:0 | T2_CNT | R  | 0x0 | 31到0位: 定时器2当前计数位[31:0]<br>定时器2 的当前只读计数值。 |

### 11.3.13 Timer2 控制寄存器 T2\_CR (偏移: 30h)

| 比特   | 名称   | 属性  | 复位值   | 描述  |
|------|------|-----|-------|---|
| 31:6 | RSV  | -   | -     | 保留  |
| 5    | CMS  | R/W | 0     | 中心对齐模式选择位[1:0]<br>0: 边沿对齐模式。counter计数方向由DIR决定。<br>1: 中心对齐模式1。Counter往上计数和往下计数交替进行。  |
| 4    | DIR  | R/W | 0     | 方向控制位<br>0: counter向上计数<br>1: counter向下计数   |
| 3    | INTM | R/W | 0     | 定时器2 中断屏蔽位<br>0 定时器2 中断使能<br>1 定时器2 中断屏蔽。   |
| 2:1  | MODE | R/W | 2'b00 | 模式选择位[1:0]<br>0x: 自由运行模式, 计数从4294967295 到0 (向下计数) 或从0到4294967295 (向上计数) 循环或先从0到4294967295再从4294967295到0 (中心对齐模式)。<br>10: 循环模式, 计数从初始计数值开始循环。<br>11: 单次模式, 计数从初始计数值到溢出值 (向上计数) 或到0 (向下计数), 然后停止。 |
| 0    | EN   | R/W | 0     | 定时器2 使能位<br>0: 计数器2关闭<br>1: 计数器2使能  |

### 11.3.14 Timer2 中断状态寄存器 T2\_IF (偏移: 34h)

| 比特   | 名称  | 属性 | 复位值 | 描述 |
|------|-----|----|-----|----|
| 31:1 | RSV | -  | -   | 保留 |

| 比特 | 名称  | 属性 | 复位值 | 描述  |
|----|-----|----|-----|---|
| 0  | INT | R  | 0   | 反映定时器2的中断状态<br>0: 中断无效。<br>1: 中断有效。<br>请写中断清除寄存器清除中断。 |

### 11.3.15 Timer2 中断清除寄存器 T2\_CIF (偏移: 38h)

| 比特   | 名称     | 属性 | 复位值 | 描述                                     |
|------|--------|----|-----|--|
| 31:0 | T1_CIF | W  | 0x0 | 对 Timer2 中断清除寄存器写任意值, 都会清除计数器 1 的中断输出。 |

### 11.3.16 Timer3 加载寄存器 T3\_ARR (偏移: 3Ch)

| 比特   | 名称   | 属性  | 复位值 | 描述   |
|------|------|-----|-----|--|
| 31:0 | LOAD | R/W | 0x0 | 定时器3加载值位[31:0]<br>定时器3的计数器加载值。在循环模式和单次模式, 只有在UEV事件发生时才会把加载值加载到定时器。 |

### 11.3.17 Timer3 计数寄存器 T3\_CNT (偏移: 40h)

| 比特   | 名称     | 属性 | 复位值 | 描述  |
|------|--------|----|-----|---|
| 31:0 | T3_CNT | R  | 0x0 | 31 到0 位: 定时器3 当前计数位[31:0]<br>定时器3 的当前只读计数值。 |

### 11.3.18 Timer3 控制寄存器 T3\_CR (偏移: 44h)

| 比特   | 名称  | 属性  | 复位值 | 描述   |
|------|-----|-----|-----|--|
| 31:6 | RSV | -   | -   | 保留   |
| 5    | CMS | R/W | 0   | 中心对齐模式选择位[1:0]<br>0: 边沿对齐模式。counter计数方向由DIR决定。<br>1: 中心对齐模式1。Counter往上计数和往下计数交替进行。 |
| 4    | DIR | R/W | 0   | 方向控制位<br>0: counter向上计数<br>1: counter向下计数  |

| 比特  | 名称   | 属性  | 复位值   | 描述  |
|-----|------|-----|-------|---|
| 3   | INTM | R/W | 0     | 定时器3 中断屏蔽位<br>0 定时器3 中断使能<br>1 定时器3 中断屏蔽。   |
| 2:1 | MODE | R/W | 2'b00 | 模式选择位[1:0]<br>0x：自由运行模式，计数从4294967295到0（向下计数）或从0到4294967295（向上计数）循环或先从0到4294967295再从4294967295到0（中心对齐模式）。<br>10：循环模式，计数从初始计数值开始循环。<br>11：单次模式，计数从初始计数值到溢出值（向上计数）或到0（向下计数），然后停止。 |
| 0   | EN   | R/W | 0     | 定时器3 使能位<br>0：计数器3 关闭<br>1：计数器3 使能  |

### 11.3.19 Timer3 中断状态寄存器 T3\_IF (偏移：48h)

| 比特   | 名称  | 属性 | 复位值 | 描述  |
|------|-----|----|-----|---|
| 31:1 | RSV | R  | -   | 保留  |
| 0    | INT | R  | 0   | 反映定时器3的中断状态<br>0：中断无效。<br>1：中断有效。<br>请写中断清除寄存器清除中断。 |

### 11.3.20 Timer3 中断清除寄存器 T3\_CIF (偏移：4Ch)

| 比特   | 名称     | 属性 | 复位值 | 描述                                    |
|------|--------|----|-----|---------------------------------------|
| 31:0 | T1_CIF | W  | 0x0 | 对 Timer3 中断清除寄存器写任意值，都会清除计数器 1 的中断输出。 |

### 11.3.21 定时器预分频寄存器 TIM\_PSC (偏移：50h)

请注意，改变预分配寄存器只能在 Timer enable 关闭时才能进行，Timer enable 打开后不允许改变预分配寄存器的值。而且，在改变预分配寄存器后至少一个 timer 时钟（分频后时钟）才能打开 Timer enable。

| 比特    | 名称  | 属性 | 复位值 | 描述 |
|-------|-----|----|-----|----|
| 31:12 | RSV | -  | -   | 保留 |

| 比特   | 名称   | 属性  | 复位值    | 描述  |
|------|------|-----|--------|---|
| 11:9 | DIV3 | R/W | 0      | 定时器3的预分频位[2:0]<br>用各自的定时器使能位来选择相应的计时器。<br>000 除1<br>001 除2<br>010 除4<br>011 除8<br>100 除16<br>101 除32<br>110 除64<br>111 除128 |
| 8:6  | DIV2 | R/W | 3'b000 | 定时器2的预分频位[2:0]<br>用各自的定时器使能位来选择相应的计时器。<br>000 除1<br>001 除2<br>010 除4<br>011 除8<br>100 除16<br>101 除32<br>110 除64<br>111 除128 |
| 5:3  | DIV1 | R/W | 3'b000 | 定时器1的预分频位[2:0]<br>用各自的定时器使能位来选择相应的计时器。<br>000 除1<br>001 除2<br>010 除4<br>011 除8<br>100 除16<br>101 除32<br>110 除64<br>111 除128 |
| 2:0  | DIV0 | R/W | 3'b000 | 定时器0预分频位[2:0]<br>用各自的定时器使能位来选择相应的计时器。<br>000 除1<br>001 除2<br>010 除4<br>011 除8<br>100 除16<br>101 除32<br>110 除64<br>111 除128  |

### 11.3.22 工作模式控制寄存器 ICMODE (偏移: 54h)

| 比特   | 名称     | 属性  | 复位值 | 描述  |
|------|--------|-----|-----|---|
| 31:3 | RSV    | -   | -   | 保留  |
| 2    | C2MODE | R/W | 0   | Channel2工作模式控制位:<br>0: 输入捕获模式<br>1: PWM模式 |
| 1    | RSV    | -   | -   | 保留  |
| 0    | C0MODE | R/W | 0   | Channel0工作模式控制位:<br>0: 输入捕获模式<br>1: PWM模式 |

### 11.3.23 输入捕获模式控制寄存器 CCR (偏移: 58h)

| 比特   | 名称    | 属性  | 复位值 | 描述   |
|------|-------|-----|-----|--|
| 31:6 | RSV   | -   | -   | 保留   |
| 5    | TEC2  | R/W | 0   | Channel2沿触发控制位<br>0 上升沿触发<br>1 下降沿触发                 |
| 4    | INTM2 | R/W | 0   | Channel2输入捕获模式中断屏蔽位<br>0 输入捕获模式中断使能<br>1 输入捕获模式中断屏蔽。 |
| 3    | EN2   | R/W | 0   | Channel2输入捕获模式使能位<br>0: 输入捕获模式关闭<br>1: 输入捕获模式使能      |
| 2    | TEC0  | R/W | 0   | Channel0沿触发控制位<br>0 上升沿触发<br>1 下降沿触发                 |
| 1    | INTM0 | R/W | 0   | Channel0输入捕获模式中断屏蔽位<br>0 输入捕获模式中断使能<br>1 输入捕获模式中断屏蔽。 |
| 0    | EN0   | R/W | 0   | Channel0输入捕获模式使能位<br>0: 输入捕获模式关闭<br>1: 输入捕获模式使能      |

### 11.3.24 输入捕获模式中断状态寄存器 CCIF (偏移: 5Ch)

| 比特   | 名称   | 属性 | 复位值 | 描述  |
|------|------|----|-----|---|
| 31:2 | RSV  | -  | -   | 保留  |
| 1    | INT2 | R  | 0   | Channel2反映输入捕获模式的中断状态<br>1: 中断有效。<br>0: 中断无效。 |

| 比特 | 名称   | 属性 | 复位值 | 描述  |
|----|------|----|-----|---|
| 0  | INT0 | R  | 0   | Channel0反映输入捕获模式的中断状态<br>1: 中断有效。<br>0: 中断无效。 |

各个中断中断状态寄存器写 1 清 0

### 11.3.25 Channel0 输入捕获模式 counter 寄存器 C0\_CR (偏移: 60h)

| 比特   | 名称    | 属性 | 复位值 | 描述                                      |
|------|-------|----|-----|---|
| 31:0 | C0_CR | R  | 0x0 | 输入捕获模式下, 当在IC0上出现触发沿的时候, 记录当时的counter值。 |

### 11.3.26 Channel2 输入捕获模式 counter 寄存器 C2\_CR (偏移: 64h)

| 比特   | 名称    | 属性 | 复位值 | 描述                                      |
|------|-------|----|-----|---|
| 31:0 | C2_CR | R  | 0x0 | 输入捕获模式下, 当在IC2上出现触发沿的时候, 记录当时的counter值。 |

### 11.3.27 PWM 控制寄存器 PCR (偏移: 68h)

| 比特   | 名称    | 属性  | 复位值 | 描述                                       |
|------|-------|-----|-----|--|
| 31:8 | RSV   | -   | -   | 保留                                       |
| 7    | INTM3 | R/W | 0   | PWM3中断屏蔽位<br>0: PWM3中断使能<br>1: PWM3中断屏蔽。 |
| 6    | EN3   | R/W | 0   | PWM3使能位<br>0: PWM3关闭<br>1: PWM3使能        |
| 5    | INTM2 | R/W | 0   | PWM2中断屏蔽位<br>0: PWM2中断使能<br>1: PWM2中断屏蔽。 |
| 4    | EN2   | R/W | 0   | PWM2使能位<br>0: PWM2关闭<br>1: PWM2使能        |
| 3    | INTM1 | R/W | 0   | PWM1中断屏蔽位<br>0: PWM1中断使能<br>1: PWM1中断屏蔽。 |
| 2    | EN1   | R/W | 0   | PWM1使能位<br>0: PWM1关闭<br>1: PWM1使能        |
| 1    | INTM0 | R/W | 0   | PWM0中断屏蔽位<br>0: PWM0中断使能<br>1: PWM0中断屏蔽。 |

| 比特 | 名称  | 属性  | 复位值 | 描述                                |
|----|-----|-----|-----|-----------------------------------|
| 0  | EN0 | R/W | 0   | PWM0使能位<br>0: PWM0关闭<br>1: PWM0使能 |

### 11.3.28 PWM 中断状态寄存器 CPIF (偏移: 6Ch)

| 比特   | 名称   | 属性   | 复位值 | 描述                                  |
|------|------|------|-----|-------------------------------------|
| 31:4 | RSV  | -    | -   | 保留                                  |
| 3    | INT3 | RW1C | 0   | 反映PWM3的中断状态<br>1: 中断有效。<br>0: 中断无效。 |
| 2    | INT2 | RW1C | 0   | 反映PWM2的中断状态<br>1: 中断有效。<br>0: 中断无效。 |
| 1    | INT1 | RW1C | 0   | 反映PWM1的中断状态<br>1: 中断有效。<br>0: 中断无效。 |
| 0    | INT0 | RW1C | 0   | 反映PWM0的中断状态<br>1: 中断有效。<br>0: 中断无效。 |

各个中断状态寄存器写1清0

### 11.3.29 PWM0 比较寄存器 C0\_PR (偏移: 70h)

| 比特   | 名称   | 属性  | 复位值 | 描述        |
|------|------|-----|-----|-----------|
| 31:0 | P0CR | R/W | 0x0 | PWM0比较寄存器 |

### 11.3.30 PWM1 比较寄存器 C1\_PR (偏移: 74h)

| 比特   | 名称   | 属性  | 复位值 | 描述        |
|------|------|-----|-----|-----------|
| 31:0 | P1CR | R/W | 0x0 | PWM1比较寄存器 |

### 11.3.31 PWM2 比较寄存器 C2\_PR (偏移: 78h)

| 比特   | 名称   | 属性  | 复位值 | 描述        |
|------|------|-----|-----|-----------|
| 31:0 | P2CR | R/W | 0x0 | PWM2比较寄存器 |

### 11.3.32 PWM3 比较寄存器 C3\_PR (偏移: 7Ch)

| 比特   | 名称   | 属性  | 复位值 | 描述        |
|------|------|-----|-----|-----------|
| 31:0 | P3CR | R/W | 0x0 | PWM3比较寄存器 |

## 11.4 使用说明

Timer 的主体部分是四个 32 位的，拥有加载寄存器的计数器。该计数器可以往上、往下、往上/往下计数，也可以处于 free-running, cyclic, single 三种计数模式。计数时钟可以由可编程预分频因子控制。

控制寄存器、加载寄存器、可编程预分频因子都可以用软件读写。

控制这些的寄存器有：

- Timer 控制寄存器 (Tx\_CR)
- 可编程预分频因子寄存器 (TIM\_PSC)
- 加载寄存器 (Tx\_ARR)

加载寄存器是预先装载的。每次读写加载寄存器就会刷新加载寄存器。但是加载寄存器的内容只有在 timer enable 打开时或者更新事件 (UEV) 发生时才会更新到芯片中。

Counter 的 clock 由可编程预分频因子决定。

另外要注意：每次配置 Timer 的控制寄存器，enable 总要最后打开；在 enable 打开的时候，不要去修改控制寄存器。

### 11.4.1 计数方向

- 往上计数
  - 在往上计数模式中，counter 从 0 计数到自动重载值，然后重新到 0 开始计数，并产生中断。而且在此时，UEV 事件发生。
  - 当 UEV 事件发生时，芯片内部加载寄存器才会被更新。
- 往下计数
  - 在往下计数模式中，counter 从自动重载值计数到 0，然后重新到自动重载值开始计数，并产生中断。而且在此时，UEV 事件发生。
  - 当 UEV 事件发生时，芯片内部加载寄存器才会被更新。
- 中心对齐模式（上下计数）
  - 在中心对齐模式中，counter 从 0 计数到自动重载值-1，产生中断；然后又从自动重载值计数到 1，产生中断；然后又从 0 开始计数。当 counter 处于中心对齐模式时，DIR 寄存器无效。



- 每次向上溢出和向下溢出时，UEV 事件发生。
- 当 UEV 事件发生时，芯片内部加载寄存器才会被更新。

## 11.4.2 计数模式

- **free-running**

在 free-running 模式中，counter 的溢出值是 4294967295，counter 循环计数。

- **cyclic**

在 cyclic 模式中，counter 的溢出值是可配置的，配置为自动重载值，而且只在 UEV 事件发生时才会把加载寄存器中的值加载过来，counter 循环计数。

- **single**

在 single 模式中，counter 的溢出值是可配置的，配置为自动重载值，而且只在 UEV 事件发生时才会把加载寄存器中的值加载过来，counter 只计数一次。

## 11.4.3 输入捕获模式

在输入捕获模式中，当在相应的 ICx 信号出现触发沿的时候，捕捉寄存器 (Cx\_CR) 会把当时的 counter 值保存下来。当一次捕获发生后，相应的中断标志被置位，同时产生一次捕获中断。CCxIF 由软件清 0。触发变化沿可以由寄存器控制是上升沿或下降沿。注意，触发信号高电平与低电平持续时间必须大于 5 个内部时钟周期。

## 11.4.4 PWM 模式

PWM 模式可以产生波形，其频率取决于 Tx\_ARR 寄存器和 TIM\_PSC，而占空比取决于 Cx\_PR 寄存器。

使用 PWM 功能时，用户必须使能对应的 counter 和 PWM；同时，如果是使用 IC0 和 IC2，还需要在 ICMODE 中将其配置为 PWM 输出。

Timer 是产生边沿对齐的 PWM 还是中心对齐的 PWM 取决于 Tx\_CR 中的 CMS。

每个 PWM 周期都会产生中断，产生中断的时间为  $Tx\_CNT = Cx\_PR$ 。

## 11.4.5 PWM 边沿对齐模式

- **往上计数**

在下面这个例子中，我们考虑 counter 往上计数模式。当  $Tx\_CNT < Cx\_PR$  时，ICx 输出高；其余情况，ICx 输出低。中断只在  $Tx\_CNT = Cx\_PR$  时产生。

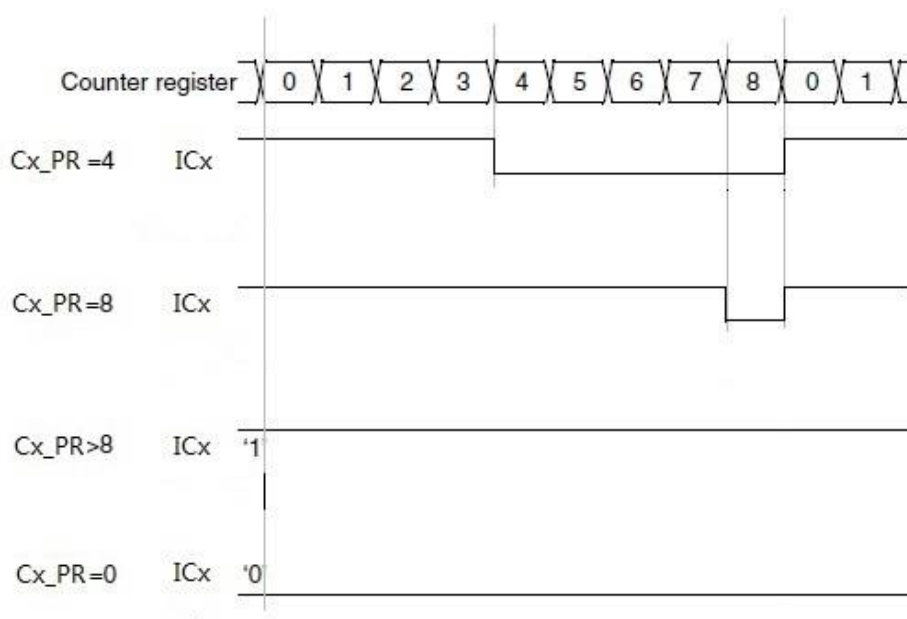


图 11-1: PWM 向上计数时序图

- 往下计数

Counter 处于往下计数模式，当  $Tx\_CNT > Cx\_PR$  时， $ICx$  输出低；其余情况， $ICx$  输出高。中断只在  $Tx\_CNT = Cx\_PR$  时产生。

#### 11.4.6 PWM 中心对齐模式

- 往上计数时，当  $Tx\_CNT < Cx\_PR$  时， $ICx$  输出高；其余情况， $ICx$  输出低。中断只在  $Tx\_CNT = Cx\_PR$  时产生。
- 往下计数时，当  $Tx\_CNT > Cx\_PR$  时， $ICx$  输出低；其余情况， $ICx$  输出高。中断只在  $Tx\_CNT = Cx\_PR$  时产生。

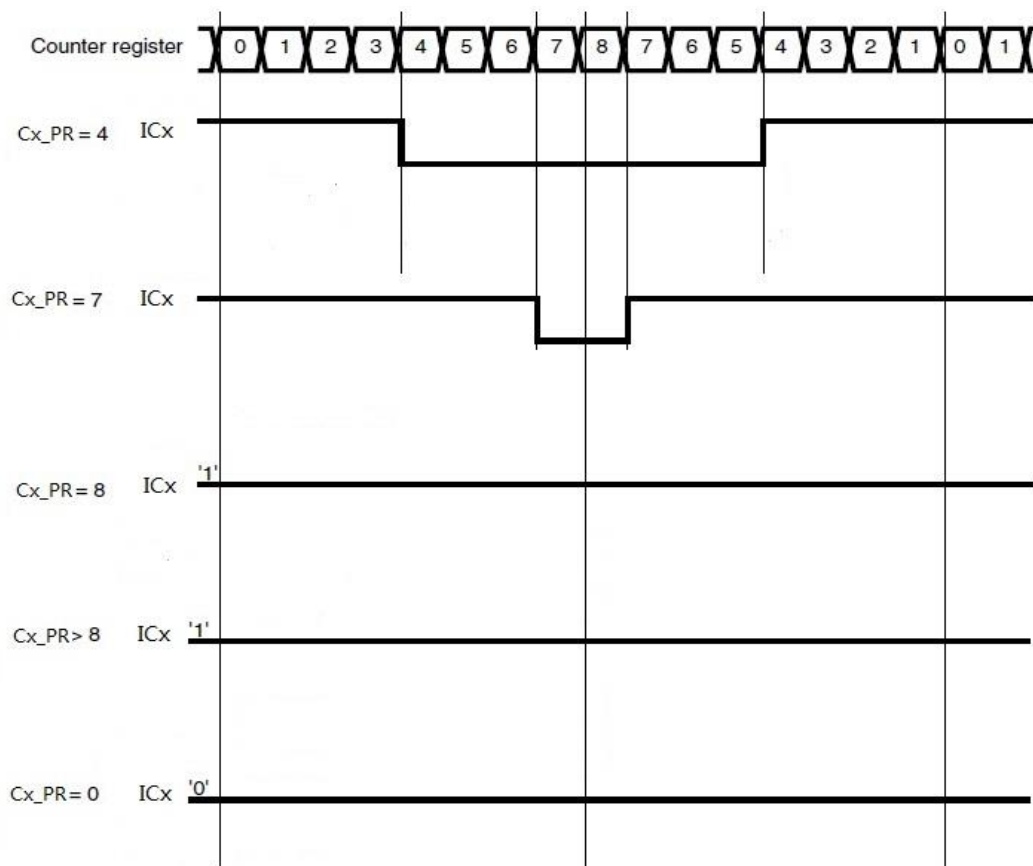


图 11-2: PWM 中心对齐时序图

### 11.4.7 输入捕获模式

在输入捕获模式中，当在相应的 ICx 信号出现触发沿的时候，捕捉寄存器（Cx\_CR）会把当时的 counter 值保存下来。当一次捕获发生后，相应的中断标志被置位，同时产生一次捕获中断。CCxIF 由软件清 0。触发变化沿可以由寄存器控制是上升沿或下降沿。注意，触发信号高电平与低电平持续时间必须大于 5 个内部时钟周期。

## 11.5 使用流程

Timer 的主体部分是四个 32 位的，拥有加载寄存器的计数器。该计数器可以往上、往下、往上/往下计数，也可以处于 free-running, cyclic, single 三种计数模式。计数时钟可以由可编程预分频因子控制。

控制寄存器、加载寄存器、可编程预分频因子都可以用软件读写。

控制这些的寄存器有：

- Timer 控制寄存器（Tx\_CR）
- 可编程预分频因子寄存器（TIM\_PSC）

- 加载寄存器 (Tx\_ARR)

加载寄存器是预先装载的。每次读写加载寄存器就会刷新加载寄存器。但是加载寄存器的内容只有在 timer enable 打开时或者更新事件 (UEV) 发生时才会更新到芯片中。

Counter 的 clock 由可编程预分频因子决定。

另外要注意：

每次配置 Timer 的控制寄存器，enable 总要最后打开；在 enable 打开的时候，不要去修改控制寄存器。

### 11.5.1 普通定时器

1. 配置 GTimer 控制寄存器 Tx\_CR，配置计数方向、计数对齐等。
2. 配置自动重载寄存器 Tx\_ARR。
3. 配置预分频寄存器 Tx\_PSC。
4. 配置 GTimer 控制寄存器 Tx\_CR，使能定时器更新中断。
5. 配置 GTimer 控制寄存器 Tx\_CR，选择定时器计数模式。
6. 配置 GTimer 控制寄存器 Tx\_CR，使能定时器 Tx 计数。

### 11.5.2 PWM 输出

1. 配置 GPIO 的 PWM 复用功能。
2. 配置 GTimer 控制寄存器 Tx\_CR，配置计数方向、计数对齐等。
3. 配置自动重载寄存器 Tx\_ARR。
4. 配置预分频寄存器 Tx\_PSC。
5. 配置 GTimer 控制寄存器 Tx\_CR，选择计数模式。
6. 配置工作模式寄存器 Tx\_ICMODE，选择使用的 PWM 通道和 PWM 模式。
7. 配置 PWM 控制寄存器 Tx\_PCR，使能相应通道的 PWM 中断。
8. 配置比较寄存器 Tx\_PR，配置占空比。
9. 配置 GTimer 控制寄存器 Tx\_CR，使能定时器 Tx 计数。
10. 配置 PWM 控制寄存器 Tx\_PCR，使能 PWM 输出。

### 11.5.3 输入捕获

1. 配置 GPIO 的输入捕获复用功能。
2. 配置预分频寄存器 Tx\_PSC。
3. 配置 GTimer 控制寄存器 Tx\_CR（仅 TIMER0,2 支持输入捕获功能），选择计数模式。
4. 配置工作模式寄存器 Tx\_ICMODE，选择使用的通道和 Capture 模式。

5. 配置输入捕获控制寄存器 Tx\_CCR，使能相应通道的输入捕获中断及触发沿。
6. 配置 GTimer 控制寄存器 Tx\_CR，使能定时器 Tx 计数。
7. 配置输入捕获控制寄存器 Tx\_CCR，使能输入捕获功能。

## 12 LPTIMER

### 12.1 概述

LPTIMER 是低功耗定时/计数器模块。通过选择合适的工作时钟，在各种低功耗模式下保持运行，并且只消耗很低的功耗。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 向上计数器
- 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（CLK32K）、RCLP（CLK1HZ）、PCLK
  - 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 寄存器描述

LPTIMER 寄存器基地址：0x40001000

表 12-1: LPTIMER 寄存器列表

| 偏置   | 名称        | 描述            |
|------|-----------|---------------|
| 0x00 | LPTCFG    | LPTIM 配置寄存器   |
| 0x04 | LPTCNT    | LPTIM 计数寄存器   |
| 0x08 | LPTCMP    | LPTIM 比较值寄存器  |
| 0x0C | LPTTARGET | LPTIM 目标值寄存器  |
| 0x10 | LPTIE     | LPTIM 中断使能寄存器 |
| 0x14 | LPTIF     | LPTIM 中断标志寄存器 |
| 0x18 | LPTCTRL   | LPTIM 控制寄存器   |

### 12.3.1 配置寄存器 LPTCFG (偏移: 00h)

| 比特    | 名称       | 属性  | 默认值 | 功能描述  |
|-------|----------|-----|-----|---|
| 31:13 | RSV      | -   | -   | 保留  |
| 12:10 | DIVSEL   | R/W | 0   | 计数时钟分频选择<br>000: 1 分频<br>001: 2 分频<br>010: 4 分频<br>011: 8 分频<br>100: 16 分频<br>101: 32 分频<br>110: 64 分频<br>111: 128 分频   |
| 9:8   | CLKSEL   | R/W | 0   | 时钟源选择<br>00: LSCLK (由 SYSREG->SYSCTRL0[12]选择的系统低速时钟 CLK32K, 即 RCL 或 XTL) 作为计数时钟<br>01: RCLP (CLK_1HZ) 作为计数时钟<br>10: PCLK 的门控时钟 (PCLK) 作为计数时钟<br>11: LPTIN (由 SYSREG->SYSCTRL0[21]选择的外部引脚或 CLK_1HZ) 作为计数时钟 |
| 7     | EDGESEL  | R/W | 0   | LPTIN 输入边沿选择<br>0: LPTIN 的上升沿计数<br>1: LPTIN 的下降沿计数  |
| 6:5   | TRIGCFG  | R/W | 0   | 外部触发边沿选择<br>00: 外部输入信号上升沿 trigger<br>01: 外部输入信号下降沿 trigger<br>10/11: 外部输入信号上升下降沿 trigger  |
| 4     | POLARITY | R/W | 0   | 计数时钟分频选择<br>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<br>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  |
| 3     | PWM      | R/W | 0   | 脉宽调制模式<br>0: 周期方波输出模式<br>1: PWM 输出模式  |
| 2     | MODE     | R/W | 0   | 计数模式<br>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。<br>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。   |

| 比特  | 名称    | 属性  | 默认值 | 功能描述  |
|-----|-------|-----|-----|---|
| 1:0 | TMODE | R/W | 0   | 工作模式选择<br>00: 带波形输出的普通定时器模式<br>01: Trigger 脉冲触发计数模式<br>10: 外部异步脉冲计数模式<br>11: Timeout 模式 |

### 12.3.2 计数值寄存器 LPTCNT (偏移: 04h)

| 比特    | 名称    | 属性 | 默认值 | 功能描述  |
|-------|-------|----|-----|-------|
| 31:16 | RSV   | -  | -   | 保留    |
| 15:0  | CNT16 | R  | 0   | 计数器数值 |

### 12.3.3 比较寄存器 LPTCMP (偏移: 08h)

| 比特    | 名称          | 属性  | 默认值 | 功能描述   |
|-------|-------------|-----|-----|--------|
| 31:16 | RSV         | -   | -   | 保留     |
| 15:0  | COMPARE_REG | R/W | 0   | 比较值寄存器 |

### 12.3.4 目标寄存器 LPTTARGET (偏移: 0Ch)

| 比特    | 名称         | 属性  | 默认值 | 功能描述   |
|-------|------------|-----|-----|--------|
| 31:16 | RSV        | -   | -   | 保留     |
| 15:0  | TARGET_REG | R/W | 0   | 目标值寄存器 |

### 12.3.5 中断使能寄存器 LPTIE (偏移: 10h)

| 比特   | 名称     | 属性  | 默认值 | 功能描述  |
|------|--------|-----|-----|---|
| 31:3 | RSV    | -   | -   | 保留  |
| 2    | TRIGIE | R/W | 0   | 外部触发到来中断使能位<br>1: 外部触发到来中断使能<br>0: 外部触发到来中断禁止       |
| 1    | OVIE   | R/W | 0   | 计数器溢出中断使能位<br>1: 计数器溢出中断使能<br>0: 计数器溢出中断禁止          |
| 0    | COMPIE | R/W | 0   | 比较匹配中断使能位<br>1: 计数器值和比较值匹配中断使能<br>0: 计数器值和比较值匹配中断禁止 |



### 12.3.6 中断标志寄存器 LPTIF（偏移：14h）

| 比特   | 名称     | 属性  | 默认值 | 功能描述  |
|------|--------|-----|-----|---|
| 31:3 | RSV    | -   | -   | 保留  |
| 2    | TRIGIF | R/W | 0   | 外部触发到来中断标志位，写 1 清零<br>1：外部触发到来中断产生<br>0：无中断产生   |
| 1    | OVIF   | R/W | 0   | 计数器溢出中断使能位，写 1 清零<br>1：计数器溢出中断产生<br>0：无中断产生     |
| 0    | COMPIF | R/W | 0   | 比较匹配中断使能位，写 1 清零<br>1：计数器值和比较值匹配中断产生<br>0：无中断产生 |

### 12.3.7 控制寄存器 LPTCTRL（偏移：18h）

| 比特   | 名称    | 属性  | 默认值 | 功能描述                                |
|------|-------|-----|-----|-------------------------------------|
| 31:1 | RSV   | -   | -   | 保留                                  |
| 0    | LPTEN | R/W | 0   | LPTIM 使能位<br>1：使能计数器计数<br>0：禁止计数器计数 |

## 12.4 使用流程

### 12.4.1 普通定时器

1. 配置 LPTCFG.CLKSEL，选择时钟源。
2. 配置 LPTCFG.DIV，设置分频值。
3. 配置 LPTCFG.MODE，设置计数模式。
4. 配置 LPTCFG.TMODE，选择普通定时器模式。
5. 配置 LPTTARGET 目标寄存器值。
6. 使能 LPTIE 中断寄存器，选择溢出中断。
7. 使能 LPTCTRL.LPTEN 位，启动计数器。

### 12.4.2 PWM 输出

1. 配置 LPTCFG.CLKSEL，选择时钟源。
2. 配置 LPTCFG.DIV，设置分频值。
3. 配置 LPTCFG.MODE，设置计数模式。
4. 配置 LPTCFG.PWM，选择 PWM 输出模式。

5. 配置 LPTCFG.POLARITY，选择波形极性。
6. 配置 LPTCFG.TMODE，选择普通定时器模式。
7. 配置 LPTCMP 比较寄存器值。
8. 配置 LPTTARGET 目标寄存器值。
9. 使能 LPTIE 中断寄存器，打开中断。
10. 使能 LPTCTRL.LPTEN 位，启动计数器。

### 12.4.3 Trigger 脉冲触发计数模式

1. 配置 LPTCFG.CLKSEL，选择时钟源。
2. 配置 LPTCFG.DIV，设置分频值。
3. 配置 LPTCFG.MODE，设置计数模式。
4. 配置 LPTCFG.TRIGCFG，设置外部触发边沿。
5. 配置 LPTCFG.TMODE，选择 Trigger 脉冲触发计数模式。
6. 使能 LPTIE.TRIGIE 中断寄存器，打开外部触发中断。
7. 使能 LPTCTRL.LPTEN 位，启动计数器。

### 12.4.4 外部异步脉冲计数模式

1. 配置 LPTCFG.CLKSEL，选择时钟源为 LPTIN。
2. 配置 LPTCFG.DIV，设置分频值。
3. 配置 LPTCFG.MODE，设置计数模式。
4. 配置 LPTCFG.EDGESEL，设置 LPTIN 输入边沿。
5. 配置 LPTCFG.TMODE，选择外部异步脉冲计数模式。
6. 配置 LPTTARGET 目标寄存器值。
7. 使能 LPTIE 中断寄存器，打开中断。
8. 使能 LPTCTRL.LPTEN 位，启动计数器。

### 12.4.5 Timeout 模式

1. 配置 LPTCFG.CLKSEL，选择时钟源 LSCLK。
2. 配置 LPTCFG.DIV，设置分频值。
3. 配置 LPTCFG.MODE，设置计数模式。
4. 配置 LPTCFG.TRIGCFG，设置外部触发边沿。
5. 配置 LPTCFG.TMODE，选择 Timeout 模式。
6. 配置 LPTTARGET 目标寄存器值。

7. 使能 LPTIE 中断寄存器，打开溢出中断。
8. 使能 LPTCTRL.LPTEN 位，启动计数器。

# 13 ADC

## 13.1 概述

12 位 ADC 是逐次接近型模数转换器。它具有多达 4 个输入通道，可测量来自 4 个外部源的信号。这些通道的 A/D 转换可在单次或连续扫描模式下进行。ADC 控制器实现 CPU 和 SAR ADC 之间的通信。ADC 转换的结果存储在数据寄存器的低 12 位。

## 13.2 主要特性

- 支持 DMA 操作
- 16 位的可编程分频器，用于产生 A/D 时钟
- 支持 12 位分辨率 A/D 输入数据，最大采样率取决于 ADC 模块
- 支持 4 通道 ADC 输入
- 支持关闭模拟 ADC
- 支持轮询（poll）和中断（interrupt）传输模式
- 支持单次扫描或连续扫描模式
- 中断源：通道数据有效（4 个通道各有一个中断源）、FIFO 满（32 个数据）、FIFO 数据量达到设定值（16 个数据）

## 13.3 寄存器描述

ADC 寄存器基地址：0x40001C00

表 13-1: ADC 寄存器列表

| 偏置    | 名称        | 描述             |
|-------|-----------|----------------|
| 0x000 | ADC_GCR   | ADC 通用控制寄存器    |
| 0x004 | ADC0_DR   | A/D 通道 0 数据寄存器 |
| 0x008 | ADC1_DR   | A/D 通道 1 数据寄存器 |
| 0x00C | ADC2_DR   | A/D 通道 2 数据寄存器 |
| 0x010 | ADC3_DR   | A/D 通道 3 数据寄存器 |
| 0x024 | ADC_CDR   | A/D 时钟分频寄存器    |
| 0x028 | ADC_ISR   | A/D 中断状态寄存器    |
| 0x02C | ADC_IER   | A/D 中断使能寄存器    |
| 0x030 | ADC_ICR   | A/D 中断清除寄存器    |
| 0x034 | ADC_COUNT | A/D 切换间隔计数寄存器  |
| 0x038 | ADC_RXREG | A/D 接收数据寄存器    |
| 0x03C | ADC_CSTAT | ADC 状态寄存器      |
| 0x040 | ADC_SPW   | ADC 采样脉宽寄存器    |

### 13.3.1 ADC 通用控制寄存器 ADC\_GCR (偏移: 000h)

| 比特    | 名称            | 属性  | 复位值 | 描述  |
|-------|---------------|-----|-----|---|
| 31:25 | RSV           | -   | -   | 保留  |
| 24    | DATA_SAMP_NEG | R/W | 0x0 | ADC 数据在 EOC 信号的边沿采样选择:<br>0: ADC 数据在 EOC 的上升沿被采样;<br>1: ADC 数据在 EOC 的下降沿被采样;<br>注意: 在本芯片设计中此位只能设置为 0。   |
| 23:16 | CH_EN[7:0]    | R/W | 0x0 | 使能对应的 ADC 通道进行 AD 转换。默认值为 0。<br>0: 禁用通道转换<br>1: 使能通道转换<br>Bit[16]: CH_EN[0] 使能通道 0 信号<br>Bit[17]: CH_EN[1] 使能通道 1 信号<br>Bit[18]: CH_EN[2] 使能通道 2 信号<br>Bit[19]: CH_EN[3] 使能通道 3 信号<br>注:<br>1) 在单次扫描模式中, 被选择通道只能进行一次 AD 转换。例如, 选择通道 3 进行 AD 转换, 设置 CH_EN[7:0]=8'b0000 1000; 如果设置 CH_EN[7:0]= 8'b00001101, 即为选择 3 个通道进行转换(通道 0, 通道 2, 通道 3)。AD 转换会在通道 3 转换完成后结束。<br>2) 在连续扫描模式中, 被选择通道可按照从低到高的顺序进行循环转换。例如, 设置 CH_EN[7:0]= 8'b00001101, 即为选择 3 个通道进行转换(通道 0, 通道 2, 通道 3)。转换时通道 0 先进行 AD 转换, 然后到通道 2, 再到通道 3, 然后再从通道 0 开始转换。 |
| 15:11 | RSV           | -   | -   | 保留  |
| 10    | ADC_START_EN  | R/W | 0x0 | ADC 转换开始使能信号。当信号从低到高转换时, ADC 转换开始。当信号从高到低转换时, ADC 转换完成。此位在 ADC_EN=0 时清零。默认值为 0。   |
| 9     | ADC_RST       | R/W | 1   | ADC 内部模拟缓冲器复位信号<br>0: SAR ADC 正常工作<br>1: SAR ADC 复位   |
| 8     | ADC_PD_EN     | R/W | 1   | SAR ADC 掉电使能信号<br>0: SAR ADC 上电<br>1: SAR ADC 掉电  |
| 7     | RSV           | -   | -   | 保留  |
| 6     | ADC_CLK_SEL   | R/W | 0   | A/D 时钟源选择信号<br>0: ADC 时钟由内部时钟分频器产生<br>1: ADC 时钟由系统时钟产生  |
| 5     | ADC_RCLR_EN   | R/W | 0   | ADC 数据寄存器读取后清除使能<br>0: 禁止 ADC 数据寄存器读取后清除<br>1: 使能 ADC 数据寄存器读取后清除  |

| 比特 | 名称         | 属性  | 复位值 | 描述  |
|----|------------|-----|-----|---|
| 4  | RXTLF      | R/W | 0   | RX FIFO 中断触发条件位<br>0: RX FIFO 里有一个或以上有效数据<br>1: RX FIFO 里有 8 个或以上有效数据           |
| 3  | RXFIFO_EN  | R/W | 0   | RX FIFO 使能位<br>0: RX FIFO 禁用, 刷新 RX FIFO 中的数据<br>1: RX FIFO 使能                  |
| 2  | DMAMODE    | R/W | 0   | DMA 访问模式位<br>1: DMA 访问模式(只有 DMA 能访问 RX FIFO)<br>0: CPU 访问模式(只有 CPU 能访问 RX FIFO) |
| 1  | CONTINUOUS | R/W | 0   | ADC 工作模式选择位。默认值为 0。<br>0: 单次扫描模式<br>1: 连续扫描模式                                   |
| 0  | ADC_EN     | R/W | 0   | ADC 控制器使能信号。默认值为 0<br>0: 禁用 ADC 模块<br>1: 使能 ADC 模块                              |

### 13.3.2 A/D 通道 0 数据寄存器 ADC0\_DR (偏移: 004h)

| 比特    | 名称          | 属性 | 复位值 | 描述   |
|-------|-------------|----|-----|--|
| 31:16 | RSV         | -  | -   | 保留   |
| 15    | DATA_VALID0 | R  | 0   | 数据有效位 (只用于单次扫描轮询模式)。此位在 ADC_EN=0 时清除, 或者当 ADC_RCLR_EN=1 时的读取操作后清除。<br>0: 数据无效<br>1: 数据有效 |
| 14:12 | RSV         | -  | -   | 保留   |
| 11:0  | CH0_DATA    | R  | 0x0 | A/D 通道 0 接收数据寄存器<br>当 ADC_RCLR_EN=1 时进行读取操作后, 此位将被清除                                     |

### 13.3.3 A/D 通道 1 数据寄存器 ADC1\_DR (偏移: 008h)

| 比特    | 名称          | 属性 | 复位值 | 描述   |
|-------|-------------|----|-----|--|
| 31:16 | RSV         | -  | -   | 保留   |
| 15    | DATA_VALID1 | R  | 0   | 数据有效位 (只用于单次扫描轮询模式)。此位在 ADC_EN=0 时清除, 或者当 ADC_RCLR_EN=1 时的读取操作后清除。<br>0: 数据无效<br>1: 数据有效 |
| 14:12 | RSV         | -  | -   | 保留   |

| 比特   | 名称       | 属性 | 复位值 | 描述   |
|------|----------|----|-----|--|
| 11:0 | CH1_DATA | R  | 0x0 | A/D 通道 1 接收数据寄存器<br>当 ADC_RCLR_EN=1 时进行读取操作后, 此位将被清除 |

### 13.3.4 A/D 通道 2 数据寄存器 ADC2\_DR (偏移: 00Ch)

| 比特    | 名称           | 属性 | 复位值 | 描述   |
|-------|--------------|----|-----|--|
| 31:16 | RSV          | -  | -   | 保留   |
| 15    | DATA_VALID 2 | R  | 0   | 数据有效位 (只用于单次扫描轮询模式)。此位在 ADC_EN=0 时清除, 或者当 ADC_RCLR_EN=1 时的读取操作后清除。<br>0: 数据无效<br>1: 数据有效 |
| 14:12 | RSV          | -  | -   | 保留   |
| 11:0  | CH2_DATA     | R  | 0x0 | A/D 通道 2 接收数据寄存器<br>当 ADC_RCLR_EN=1 时进行读取操作后, 此位将被清除                                     |

### 13.3.5 A/D 通道 3 数据寄存器 ADC3\_DR (偏移: 010h)

| 比特    | 名称          | 属性 | 复位值 | 描述   |
|-------|-------------|----|-----|--|
| 31:16 | RSV         | -  | -   | 保留   |
| 15    | DATA_VALID3 | R  | 0   | 数据有效位 (只用于单次扫描轮询模式)。此位在 ADC_EN=0 时清除, 或者当 ADC_RCLR_EN=1 时的读取操作后清除。<br>0: 数据无效<br>1: 数据有效 |
| 14:12 | RSV         | -  | -   | 保留   |
| 11:0  | CH3_DATA    | R  | 0x0 | A/D 通道 3 接收数据寄存器<br>当 ADC_RCLR_EN=1 时进行读取操作后, 此位将被清除                                     |

### 13.3.6 ADC 时钟分频寄存器 ADC\_CDR (偏移: 024h)

| 比特    | 名称  | 属性 | 复位值 | 描述 |
|-------|-----|----|-----|----|
| 31:16 | RSV | -  | -   | 保留 |

| 比特   | 名称     | 属性  | 复位值    | 描述   |
|------|--------|-----|--------|--|
| 15:0 | CLKDIV | R/W | 0x00FF | ADC 内部时钟分频寄存器。<br>分频公式：<br>$adc\_clk = fpclk / CLKDIV$ (fpclk : APB 总线时钟)<br>注：建议 CLKDIV $\geq 1$ 。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。 |

### 13.3.7 ADC 中断状态寄存器 ADC\_ISR (偏移: 028h)

| 比特   | 名称               | 属性 | 复位值 | 描述   |
|------|------------------|----|-----|--|
| 31:8 | RSV              | -  | -   | 保留   |
| 9    | RXFIFO_FULL_INTF | R  | 0   | RX FIFO 满中断标志位 (只用于 FIFO 使能模式)<br>0 = 接收 FIFO 未满<br>1 = 接收 FIFO 满  |
| 8    | RX_INTF          | R  | 0   | 接收器数据可用中断标志位 (只用于 FIFO 使能模式)<br>当接收器 FIFO 接收到足够数据时，此标志位置位 (根据 RXTLF 位设置)<br>0 = 接收器 FIFO 无可用数据<br>1 = 接收器 FIFO 有可用数据 |
| 7:4  | RSV              | -  | -   | 保留   |
| 3    | CH3_INTF         | R  | 0   | 通道 3 数据中断，高电平有效<br>0 = 没有中断<br>1 = 中断激活  |
| 2    | CH2_INTF         | R  | 0   | 通道 2 数据中断，高电平有效<br>0 = 没有中断<br>1 = 中断激活  |
| 1    | CH1_INTF         | R  | 0   | 通道 1 数据中断，高电平有效<br>0 = 没有中断<br>1 = 中断激活  |
| 0    | CH0_INTF         | R  | 0   | 通道 0 数据中断，高电平有效<br>0 = 没有中断<br>1 = 中断激活  |

### 13.3.8 ADC 中断使能寄存器 ADC\_IER (偏移: 02Ch)

| 比特    | 名称  | 属性 | 复位值 | 描述 |
|-------|-----|----|-----|----|
| 31:10 | RSV | -  | -   | 保留 |



| 比特  | 名称              | 属性  | 复位值 | 描述   |
|-----|-----------------|-----|-----|--|
| 9   | RXFIFO_FULL_IEN | R/W | 0   | RX FIFO 满中断使能（只用于 FIFO 使能模式）<br>0 = 禁止中断<br>1 = 中断使能 |
| 8   | RXIEN           | R/W | 0   | 接收器 FIFO 中断使能（只用于 FIFO 使能模式）<br>0 = 禁止中断<br>1 = 中断使能 |
| 7:4 | RSV             | -   | -   | 保留   |
| 3   | CH3_INT_EN      | R/W | 0   | 通道 3 数据中断使能，高电平有效<br>0 = 禁止中断<br>1 = 中断使能            |
| 2   | CH2_INT_EN      | R/W | 0   | 通道 2 数据中断使能，高电平有效<br>0 = 禁止中断<br>1 = 中断使能            |
| 1   | CH1_INT_EN      | R/W | 0   | 通道 1 数据中断使能，高电平有效<br>0 = 禁止中断<br>1 = 中断使能            |
| 0   | CH0_INT_EN      | R/W | 0   | 通道 0 数据中断使能，高电平有效<br>0 = 禁止中断<br>1 = 中断使能            |

### 13.3.9 ADC 中断清除寄存器 ADC\_ICR（偏移：030h）

| 比特    | 名称               | 属性  | 复位值 | 描述  |
|-------|------------------|-----|-----|---|
| 31:10 | RSV              | -   | -   | 保留  |
| 9     | RXFIFO_FULL_ICLR | R/W | 0   | RX FIFO 满中断清除（只用于 FIFO 使能模式）<br>0 = 不清除中断<br>1 = 清除中断 |
| 8     | RXICLR           | R/W | 0   | RX FIFO 中断清除（只用于 FIFO 使能模式）<br>0 = 不清除中断<br>1 = 清除中断  |
| 7:4   | RSV              | -   | -   | 保留  |
| 3     | CH3_INT_CLR      | W   | 0   | 通道 3 数据中断清除，高电平有效<br>0 = 不清除中断<br>1 = 清除中断            |
| 2     | CH2_INT_CLR      | W   | 0   | 通道 2 数据中断清除，高电平有效<br>0 = 不清除中断<br>1 = 清除中断            |
| 1     | CH1_INT_CLR      | W   | 0   | 通道 1 数据中断清除，高电平有效<br>0 = 不清除中断<br>1 = 清除中断            |

| 比特 | 名称          | 属性 | 复位值 | 描述  |
|----|-------------|----|-----|---|
| 0  | CH0_INT_CLR | W  | 0   | 通道 0 数据中断清除，高电平有效。<br>0 = 不清除中断<br>1 = 清除中断 |

### 13.3.10 ADC 切换间隔计数寄存器(ADC\_COUNT (偏移: 034h))

| 比特   | 名称        | 属性  | 复位值  | 描述  |
|------|-----------|-----|------|---|
| 31:8 | RSV       | -   | -    | 保留  |
| 7:0  | ADC_COUNT | R/W | 0x01 | 通道切换间隔时间，这个值的单位为 ADC 时钟周期。<br>实际通道切换时间=(adc_count+16) * ADC 时钟周期<br>注：此寄存器只能在 ADC 控制器使能前配置，否则无效 |

### 13.3.11 ADC 接收数据寄存器 ADC\_RXREG (偏移: 038h)

| 比特    | 名称         | 属性 | 复位值 | 描述                                       |
|-------|------------|----|-----|--|
| 31:12 | RSV        | -  | -   | 保留                                       |
| 11:0  | RXFIFO_OUT | R  | 0x0 | ADC 接收数据寄存器的低 12 位为 SARADC 的转换数值。此寄存器只读。 |

### 13.3.12 ADC 状态寄存器 ADC\_CSTAT (偏移: 03Ch)

| 比特   | 名称    | 属性 | 复位值 | 描述  |
|------|-------|----|-----|---|
| 31:1 | RSV   | -  | -   | 保留  |
| 0    | RXAVL | R  | 0   | CPU 轮询模式下的接收数据可用标志位。<br>当接收器接收到有效数据时此位置位<br>0 = 接收器 FIFO 为空<br>1 = 接收器 FIFO 有有效数据 |

### 13.3.13 ADC 采样脉宽寄存器 ADC\_SPW (偏移: 040h)

| 比特   | 名称  | 属性 | 复位值 | 描述 |
|------|-----|----|-----|----|
| 31:3 | RSV | -  | -   | 保留 |

| 比特  | 名称            | 属性  | 复位值 | 描述  |
|-----|---------------|-----|-----|---|
| 2:0 | SAMPCLK_WIDTH | R/W | 0x3 | 采样脉宽配置<br>3: SAMPCLK 为 4 个 ADC_CLK 脉冲信号<br>4: SAMPCLK 为 5 个 ADC_CLK 脉冲信号<br>5: SAMPCLK 为 6 个 ADC_CLK 脉冲信号<br>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。 |

## 13.4 使用流程

### 13.4.1 单次扫描模式单通道 A/D 转换

1. 配置寄存器 ADC\_GCR, 选择 ADC 上电, ADC 复位, 使能单次扫描模式, 使用内部时钟分频器产生 ADC 时钟, 使能 ADC 数据寄存器读取后清除。
2. 配置寄存器 ADC\_CDR, 配置 ADC 时钟分频。
3. 配置寄存器 ADC\_COUNT, 配置通道切换时间。
4. 配置寄存器 ADC\_ICR, 清除所有 ADC 中断。
5. 配置寄存器 ADC\_IER, 使能通道 2 中断。
6. 配置寄存器 ADC\_GCR 的 ADC\_EN 位, 启动 ADC。
7. 选择待转换通道, 通道 2, 配置寄存器 ADC\_GCR 对应位。
8. 配置寄存器 ADC\_GCR 的 ADC\_START\_EN 位, 开始 ADC 转换。
9. 中断触发后, 读取 ADC2\_DR 寄存器的 CH2\_DATA 位数据, 然后 ADC\_ICR 寄存器置位, 清除所有 ADC 中断。
10. 配置寄存器 ADC\_GCR, 停止 ADC 转换, 禁用 SAR ADC。

### 13.4.2 单次扫描模式多通道 A/D 转换

1. 配置寄存器 ADC\_GCR, 选择 ADC 上电, ADC 复位, 使能单次扫描模式, 使用内部时钟分频器产生 ADC 时钟, 使能 ADC 数据寄存器读取后清除。
2. 配置寄存器 ADC\_CDR, 配置 ADC 时钟分频。
3. 配置寄存器 ADC\_COUNT, 配置通道切换时间。
4. 配置寄存器 ADC\_ICR, 清除所有 ADC 中断。
5. 配置寄存器 ADC\_IER, 使能最后一个转换的通道中断。
6. 配置寄存器 ADC\_GCR 的 ADC\_EN 位, 启动 ADC。
7. 选择待转换通道, 通道 2 和通道 3, 配置寄存器 ADC\_GCR 对应位。
8. 配置寄存器 ADC\_GCR 的 ADC\_START\_EN 位, 开始 ADC 转换。

9. 中断触发后，读取 ADC2\_DR 寄存器的 CH2\_DATA 位数据和 ADC3\_DR 寄存器的 CH3\_DATA 位数据，然后 ADC\_ICR 寄存器置位，清除所有 ADC 中断。
10. 配置寄存器 ADC\_GCR，停止 ADC 转换，禁用 SAR ADC。

### 13.4.3 连续扫描模式多通道 A/D 转换

1. 配置寄存器 ADC\_GCR，选择 ADC 上电，ADC 复位，使能连续扫描模式，使用内部时钟分频器产生 ADC 时钟，使能 ADC 数据寄存器读取后清除，使能 RX FIFO 模式，选择 RX FIFO 中断触发条件。
2. 配置寄存器 ADC\_CDR，配置 ADC 时钟分频。
3. 配置寄存器 ADC\_COUNT，配置通道切换时间。
4. 配置寄存器 ADC\_ICR，清除所有 ADC 中断。
5. 配置寄存器 ADC\_IER，使能 RX FIFO 中断。
6. 配置寄存器 ADC\_GCR 的 ADC\_EN 位，启动 ADC。
7. 选择待转换通道，通道 2 和通道 3，配置寄存器 ADC\_GCR 对应位。
8. 配置寄存器 ADC\_GCR 的 ADC\_START\_EN 位，开始 ADC 转换。
9. 中断触发后，依次读出 ADC\_RXREG 中的数据，然后写 ADC\_ICR 寄存器，清除所有 ADC 中断。
10. 配置寄存器 ADC\_GCR，停止 ADC 转换，禁用 SAR ADC。

### 13.4.4 注意事项

- 若 RX FIFO 已满，则新的数据无法保存进来。
- 若在单次扫描模式中使用 RX\_INTF 中断，须将 ADC\_GCR 寄存器的 RXTLF 位设为 0，否则无法触发中断。
- 在单次扫描模式中，若某通道的数据中断 CHx\_INTF 被使能而且触发，则 ADC 控制器将结束模数转换，停止读取后面未采样通道的数据。已采样通道的 CHx\_DATA 均保留在相应的 ADCx\_DR 寄存器。
- 如果禁用所有中断而采用轮询的查询方式，最好启用 RX FIFO，用 RXAVL 信号通知 CPU 接收数据。
- 在本芯片设计中，为了与 ADC 模块配合，ADC\_GCR 寄存器的 DATA\_SAMP\_NEG 位只能设为 0，ADC\_SPW 寄存器的值 SAMPCLK\_WIDTH 应该大于或等于 3。

# 14 GPIO

## 14.1 概述

GPIO 包含通用数据输入输出接口，这些管脚可以与其他功能管脚共享，这取决于芯片的配置。通过这些数据接口，可以配置任意数目的管脚作为中断信号。UM321x 有四组 GPIO，分别是 GPIOA、GPIOB、GPIOC、GPIOD，分别简称为 PA、PB、PC、PD。GPIO 的相关寄存器的功能需要设置对应的比特位，例如设置 PA1 方向为输出，GPIO\_DIR 的 bit[1]控制位需要设置为 1，其他位的设置遵循此原则，也即是 PAX 对应寄存器 GPIO\_DIR 的 bit[x]控制位。最大支持 21 个 GPIO。

## 14.2 主要特性

- 所有输入/输出引脚方向都可以通过软件进行配置
- 每个 GPIO\_IN 引脚可配置成边沿或电平方式触发中断

## 14.3 寄存器描述

GPIOA 寄存器基地址：0x40004000

GPIOB 寄存器基地址：0x40004400

GPIOC 寄存器基地址：0x40004800

GPIOD 寄存器基地址：0x40004C00

表 14-1: GPIO 寄存器列表

| 偏置   | 名称         | 描述                 |
|------|------------|--------------------|
| 0x00 | GPIO_DIR   | GPIO 数据方向寄存器       |
| 0x08 | GPIO_SET   | GPIO 输出置位寄存器       |
| 0x0C | GPIO_CLR   | GPIO 输出清零寄存器       |
| 0x10 | GPIO_ODATA | GPIO 输出引脚映射寄存器     |
| 0x14 | GPIO_IDATA | GPIO 输入引脚映射寄存器     |
| 0x18 | GPIO_IEN   | GPIO 中断使能寄存器       |
| 0x1C | GPIO_IS    | GPIO 中断触发模式寄存器     |
| 0x20 | GPIO_IBE   | GPIO 中断边沿触发设置寄存器   |
| 0x24 | GPIO_IEV   | GPIO 中断高低电平触发设置寄存器 |
| 0x28 | GPIO_IC    | GPIO 中断状态清除寄存器     |
| 0x2C | GPIO_RIS   | GPIO 原始中断状态寄存器     |
| 0x30 | GPIO_MIS   | GPIO 屏蔽后中断状态寄存器    |

### 14.3.1 数据方向寄存器 GPIO\_DIR(偏移: 00h)

| 比特  | 名称       | 属性  | 复位值  | 描述  |
|-----|----------|-----|------|---|
| 7:0 | GPIO_DIR | R/W | 0x00 | 8 位寄存器, GPIO 输入输出控制寄存器:<br>0: 输入;<br>1: 输出。 |

注: GPIOx\_DIR[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应, 例如 GPIOA\_DIR[1]与 PA1 对应。

### 14.3.2 输出置位寄存器 GPIO\_SET(偏移: 08h)

| 比特  | 名称       | 属性 | 复位值  | 描述  |
|-----|----------|----|------|---|
| 7:0 | GPIO_SET | W  | 0x00 | 8 位寄存器, GPIO 输出置位寄存器:<br>0: 无效操作;<br>1: 当 IO 为输出时, IO 置位。 |

注: GPIOx\_SET [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应, 例如 GPIOA\_SET [1]与 PA1 对应。

### 14.3.3 输出清零寄存器 GPIO\_CLR(偏移: 0Ch)

| 比特  | 名称       | 属性 | 复位值  | 描述  |
|-----|----------|----|------|---|
| 7:0 | GPIO_CLR | W  | 0x00 | 8 位寄存器, GPIO 输出清零寄存器:<br>0: 无效操作;<br>1: 当 IO 为输出时, IO 清零。 |

注: GPIOx\_CLR [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应, 例如 GPIOA\_CLR [1]与 PA1 对应。

### 14.3.4 GPIO 输出引脚映射寄存器 GPIO\_ODATA(偏移: 10h)

| 比特  | 名称         | 属性  | 复位值  | 描述  |
|-----|------------|-----|------|---|
| 7:0 | GPIO_ODATA | R/W | 0x00 | 8 位寄存器, GPIO 输出引脚映射寄存器:<br>当 GPIO 方向为输出有效, 写操作直接写至外部引脚, 读获得外部引脚值。 |

注: GPIOx\_ODATA[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应, 例如 GPIOA\_ODATA [1]与 PA1 对应。

### 14.3.5 GPIO 输入引脚映射寄存器 GPIO\_IDATA(偏移: 14h)

| 比特  | 名称         | 属性 | 复位值  | 描述   |
|-----|------------|----|------|--|
| 7:0 | GPIO_IDATA | R  | 0x00 | 8 位寄存器, GPIO 输入引脚映射寄存器:<br>当 GPIO 方向为输入有效, 读获得外部引脚值; 此寄存器为只读寄存器。 |

注：GPIOx\_IDATA[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_IDATA[1]与 PA1 对应。

### 14.3.6 GPIO 中断使能寄存器 GPIO\_IEN(偏移：18h)

| 比特  | 名称       | 属性  | 复位值  | 描述   |
|-----|----------|-----|------|--|
| 7:0 | GPIO_IEN | R/W | 0x00 | 8 位寄存器，GPIO 中断使能寄存器：<br>0= 禁止相应引脚中断；<br>1= 使能相应引脚中断。 |

注：GPIOx\_IEN[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_IEN[1]与 PA1 对应。

### 14.3.7 GPIO 中断触发模式寄存器 GPIO\_IS(偏移：1Ch)

| 比特  | 名称      | 属性  | 复位值  | 描述  |
|-----|---------|-----|------|---|
| 7:0 | GPIO_IS | R/W | 0x00 | 7 位寄存器，GPIO 中断触发模式：<br>0= 边沿检测；<br>1= 电平检测。 |

注：GPIOx\_IS[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_IS[1]与 PA1 对应。

### 14.3.8 GPIO 中断边沿触发设置寄存器 GPIO\_IBE(偏移：20h)

| 比特  | 名称       | 属性  | 复位值  | 描述  |
|-----|----------|-----|------|---|
| 7:0 | GPIO_IBE | R/W | 0x00 | 8 位寄存器，GPIO 中断边沿触发设置：<br>0= 单边沿触发；<br>1= 双边沿触发。 |

注：GPIOx\_SET[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_SET[1]与 PA1 对应。

### 14.3.9 GPIO 中断高低电平触发设置寄存器 GPIO\_IEV(偏移：24h)

| 比特  | 名称       | 属性  | 复位值  | 描述  |
|-----|----------|-----|------|---|
| 7:0 | GPIO_IEV | R/W | 0x00 | 8 位寄存器，GPIO 中断高低电平触发设置：<br>0= 下降沿/低电平触发；<br>1= 上升沿/高电平触发。 |

注：GPIOx\_IEV[y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_IEV[1]与 PA1 对应。

### 14.3.10 GPIO 中断状态清除寄存器 GPIO\_IC(偏移：28h)

| 比特  | 名称      | 属性 | 复位值  | 描述   |
|-----|---------|----|------|--|
| 7:0 | GPIO_IC | W  | 0x00 | 8 位寄存器，GPIO 中断清除寄存器：<br>0= 无效操作；<br>1= 清除对应引脚中断。 |

注：GPIOx\_IC [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_IC [1]与 PA1 对应。

### 14.3.11 GPIO 原始中断状态寄存器 GPIO\_RIS(偏移：2Ch)

| 比特  | 名称       | 属性 | 复位值  | 描述   |
|-----|----------|----|------|--|
| 7:0 | GPIO_RIS | R  | 0x00 | 8 位寄存器，GPIO 原始中断寄存器：<br>0= 对应引脚无中断挂起；<br>1= 对应引脚有中断挂起。 |

注：GPIOx\_RIS [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_RIS [1]与 PA1 对应。

### 14.3.12 GPIO 屏蔽后中断状态寄存器 GPIO\_MIS(偏移：30h)

| 比特   | 名称       | 属性 | 复位值  | 描述                                      |
|------|----------|----|------|---|
| 31:0 | GPIO_MIS | R  | 0x00 | 32 位寄存器，GPIO 屏蔽后中断状态寄存器：反映对应引脚屏蔽后的中断状态。 |

注：GPIOx\_MIS [y] (x=A...D) 寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA\_MIS [1]与 PA1 对应。

## 14.4 使用流程

### 14.4.1 输入输出 IO

1. 配置 GPIO\_DIR 寄存器，选择 GPIO 方向。
2. 可使用 GPIO\_SET/GPIO\_CLR 或 GPIO\_ODATA 来设置输出电平。
3. 使用 GPIO\_IDATA 来获取输入引脚电平。

### 14.4.2 中断触发模式

中断初始化过程：

1. 设置 GPIO\_DIR 为输入。
2. 清除 GPIO\_IE 以避免异常。
3. 配置寄存器 GPIO\_IS，选择是边沿/电平触发类型。
4. 在单边沿触发方式下，配置寄存器 GPIO\_IBE，确定是单边触发还是双边触发。



5. 在单边沿触发方式下，配置寄存器 GPIO\_I\_EV，确定是哪种边沿触发类型。
6. 在电平触发方式下，配置寄存器 GPIO\_I\_EV，确定是哪种电平触发类型。
7. 配置寄存器 GPIO\_IC 来清除中断。
8. 配置寄存器 GPIO\_I\_E 使能相应位中断。

### 14.4.3 清除中断

ISR 写 GPIO\_IC 来清除中断状态。如果在清除寄存器的同时有新的边沿触发中断产生，这个新的中断将会保持有效直到下一次清除。读取中断状态操作应该在清 GPIO\_I\_E 之前进行，清 GPIO\_I\_E 操作将清除相应中断状态。

# 15 CRC16

## 15.1 概述

CRC16 是一个以多项式  $G(x) = x^{16} + x^{12} + x^5 + 1$  为计算式的硬件 16 位 CRC 循环冗余校验计算电路。可以根据用户预设的 CRC 初值，通讯数据计算出合适的 CRC 结果，并且支持设置输入数据与结果的正反向。

## 15.2 寄存器描述

CRC 寄存器基地址：0x40001800

表 15-1：CRC 寄存器列表

| 偏置   | 名称         | 描述                 |
|------|------------|--------------------|
| 0x00 | CRC16_DATA | 写入需校验的数据与读出 CRC 结果 |
| 0x04 | CRC16_INIT | 写入 16 位 CRC 初值     |
| 0x08 | CRC16_CTRL | CRC 控制寄存器          |

### 15.2.1 数据寄存器 CRC16\_DATA（偏移：00H）

| 比特    | 名称         | 属性  | 复位值 | 描述  |
|-------|------------|-----|-----|---|
| 31:16 | RSV        | -   | -   | 保留  |
| 15:8  | RSLT2      | R   | 0x0 | 读出 16 位 CRC 计算结果的高 8 位  |
| 7:0   | DATA_RSLT1 | R/W | 0x0 | 写：写入需要进行 CRC 校验计算的数据，如需要校验的数据不止 8 位需按顺序逐次写入<br>读：读出 16 位 CRC 计算结果的低 8 位 |

注：

- 低 8 位对于需校验数据来说为只写，写入后无法再次读出。
- 读操作返回 16 位 CRC 计算结果，其中低 8 位为与数据寄存器复用。
- 读操作将会对 CRC 计算清零，即读操作后将会重新载入初始值，次输入数据时会进行与读之前结果无关的新一轮计算。

### 15.2.2 初始值寄存器 CRC16\_INIT（偏移：04H）

| 比特    | 名称   | 属性  | 复位值 | 描述              |
|-------|------|-----|-----|-----------------|
| 31:16 | RSV  | -   | -   | 保留              |
| 15:0  | INIT | R/W | 0x0 | 写入 16 位 CRC 初始值 |

### 15.2.3 控制寄存器 CRC16\_CTRL (偏移: 08H)

| 比特   | 名称          | 属性  | 复位值 | 描述                                   |
|------|-------------|-----|-----|--------------------------------------|
| 31:3 | RSV         | -   | -   | 保留                                   |
| 2    | RSLT_REV    | R/W | 0   | CRC 计算结果是否进行高低位倒序<br>1: 倒序<br>0: 不倒序 |
| 1    | DATA_REV    | R/W | 0   | CRC 计算数据是否进行高低位倒序<br>1: 倒序<br>0: 不倒序 |
| 0    | INITIAL_REV | R/W | 0   | CRC 初始值是否进行高低位倒序<br>1: 倒序<br>0: 不倒序  |

### 15.3 使用流程

1. 设置 16 位初始值 CRC16\_INIT。
2. 设置 CRC16\_CTRL, 选择数据是否倒序。
3. 向 CRC16\_DATA 中写入 8 位 CRC 计算数据, 如没有完成 CRC 数据输入顺次输入之后 8 位数据直至输入完成。
4. 读 CRC16\_DATA, 将一次返回 CRC 计算结果

注: 读取结果后, CRC 计算模块将结束当前计算并重新载入初始值以备下次计算使用。

## 16 WDT

### 16.1 概述

看门狗定时器在到达超时的值的时候可以产生不可屏蔽中断或者是复位。当系统由于软件错误或是由于外部设备故障而无法按照预期的方式响应的时候,使用看门狗定时器可以重新获得控制权。

### 16.2 主要特性

- 32 位递减并且可编程装载的寄存器
- 独立的看门狗时钟使能
- 带中断屏蔽的中断生成逻辑
- 软件跑飞保护锁定寄存器
- 复位使能/禁止产生逻辑
- 调试期间,微处理器的 CPU 暂停时,用户可使能的停滞

### 16.3 寄存器描述

WDT 寄存器基地址: 0x40002400

表 16-1: WDT 寄存器列表

| 偏置   | 名称         | 描述           |
|------|------------|--------------|
| 0x00 | WDT_LOAD   | 装载寄存器        |
| 0x04 | WDT_CNT    | 计数寄存器        |
| 0x08 | WDT_CTRL   | 控制寄存器        |
| 0x0C | WDT_CLR    | 清除寄存器        |
| 0x10 | WDT_INTRAW | RAW 中断状态寄存器  |
| 0x14 | WDT_MINTS  | MASK 中断状态寄存器 |
| 0x18 | WDT_STALL  | STALL 寄存器    |
| 0x1C | WDT_LOCK   | LOCK 寄存器     |

#### 16.3.1 装载寄存器 WDT\_LOAD(偏移: 00h)

| 比特   | 名称   | 属性  | 默认值        | 功能描述      |
|------|------|-----|------------|-----------|
| 31:0 | LOAD | R/W | 0xFFFFFFFF | WDOG初始装载值 |

### 16.3.2 计数寄存器 WDT\_CNT(偏移：04h)

| 比特   | 名称  | 属性 | 默认值        | 功能描述         |
|------|-----|----|------------|--------------|
| 31:0 | CNT | R  | 0xFFFFFFFF | WDOG内部CNT计数值 |

### 16.3.3 控制寄存器 WDT\_CTRL(偏移：08h)

| 比特   | 名称    | 属性  | 默认值 | 功能描述   |
|------|-------|-----|-----|--|
| 31   | WRC   | R   | 1   | WDT加载值设置或写WDT_CTRL寄存器生效。向WDT_LOAD或者WDT_CTRL寄存器进行写操作时，设置位生效会有一些时间的延时。<br>0：设置为仍未生效<br>1：设置位生效 |
| 30:2 | RSV   | -   | -   | 保留   |
| 1    | RSTEN | R/W | 0   | WDT溢出复位使能<br>0：不使能溢出复位功能<br>1：使能溢出复位功能   |
| 0    | INTEN | R/W | 0   | WDT中断使能<br>0：不使能中断<br>1：使能中断   |

### 16.3.4 清除寄存器 WDT\_CLR(偏移：0Ch)

| 比特   | 名称        | 属性 | 默认值 | 功能描述                              |
|------|-----------|----|-----|-----------------------------------|
| 31:0 | CLR_CARRY | W  | 0x0 | 向此寄存器写入任何值，将清除WDT溢出状态，从而清除掉中断和复位。 |

### 16.3.5 RAW 中断状态寄存器 WDT\_INTRAW(偏移：10h)

| 比特   | 名称     | 属性 | 默认值 | 功能描述  |
|------|--------|----|-----|---|
| 31:0 | INTRAW | R  | 0x0 | 原始中断寄存器，未经中断使能屏蔽<br>0：WDT内部未发生溢出<br>1：WDT内部发生溢出 |

### 16.3.6 MASK 中断状态寄存器 WDT\_MINTS(偏移：14h)

| 比特   | 名称    | 属性 | 默认值 | 功能描述                    |
|------|-------|----|-----|-------------------------|
| 31:0 | INTMS | R  | 0x0 | 0：WDT未产生中断<br>1：WDT产生中断 |

### 16.3.7 STALL 控制寄存器 WDT\_STALL(偏移：18h)

| 比特    | 名称     | 属性  | 默认值 | 功能描述   |
|-------|--------|-----|-----|--|
| 31:16 | CLKDIV | R/W | 0   | WDT计数时钟分频值。<br>0x0：不分频<br>0x1：2分频<br>0x2：3分频<br>....<br>0xFFFE：0xFFFF分频<br>0xFFFF：保留 |
| 15:9  | RSV    | -   | -   | 保留   |
| 8     | STALL  | R/W | 0   | WDT在芯片处于HALT状态时不计数功能的使能位：<br>0：不使能HALT状态计数器停止工作功能<br>1：使能HALT状态计数器停止工作功能             |
| 7:0   | RSV    | -   | -   | 保留   |

### 16.3.8 LOCK 寄存器 WDT\_LOCK(偏移：1Ch)

| 比特   | 名称   | 属性 | 默认值 | 功能描述   |
|------|------|----|-----|--|
| 31:0 | LOCK | W  | 0x0 | WDT LOCK功能使能，当使能LOCK功能时，除此寄存器外的所有WDT寄存器均不可写。向此寄存器写任意值，使能WDT LOCK功能，向此寄存器写0x1ACCE551清除LOCK功能。 |

## 16.4 使用流程

#### WDT 定时器配置：

1. 向 WDT\_LOCK 寄存器写入 0x1ACCE551 解锁寄存器。
2. 在 WDT\_STALL 寄存器设置分频值。
3. 在 WDT\_LOAD 寄存器里装载所需要的加载值。
4. 在 WDT\_CTRL 寄存器自行选择 RSTEN 复位功能和 INTEN 中断功能。
5. 等待 WDT\_CTRL 寄存器的 WRC 位被置位。
6. 向 WDT\_LOCK 寄存器写入任意值锁定寄存器。

#### WDT 喂狗流程配置：

1. 向 WDT\_LOCK 寄存器写入 0x1ACCE551 解锁寄存器。

2. 在 WDT\_LOAD 寄存器里装载所需要的重载值。
3. 等待 WDT\_CTRL 寄存器的 WRC 位被置位。
4. 向 WDT\_LOCK 寄存器写入任意值锁定寄存器。

# 17 RTC

## 17.1 概述

实时时钟 (RTC) 是一个独立的定时器/计数器, 可提供基本的闹钟中断或者长时间的计数服务。闹钟中断通过可配置的实时时钟计数周期实现。实时时钟通过 1 Hz 的时钟输入可实现以秒为间隔的计数。本模块符合高级微控制器总线架构 (AMBA) 的总线标准。

## 17.2 主要特性

- 32位的秒级计数器
- 内部或外部32KHz时钟源
- 可编程的闹钟中断
- 主电源关闭时仍可运行
- 可设置或重置当前时间
- 可编程的匹配寄存器
- 当计数器和匹配寄存器相同时的软件可触发中断

## 17.3 寄存器描述

RTC 寄存器基地址: 0x40001400

表 17-1: RTC 寄存器列表

| 偏置   | 名称      | 描述            |
|------|---------|---------------|
| 0x00 | RTCDR   | RTC 数据寄存器     |
| 0x04 | RTCMR   | RTC 匹配寄存器     |
| 0x08 | RTCLR   | RTC 加载寄存器     |
| 0x0C | RTCCR   | RTC 控制寄存器     |
| 0x10 | RTCIMSC | RTC 中断使能寄存器   |
| 0x14 | RTCRIIS | RTC 原始中断状态寄存器 |
| 0x18 | RTCMIS  | RTC 屏蔽中断状态寄存器 |
| 0x1C | RTICR   | RTC 中断清除寄存器   |

本小结对 RTC 寄存器进行了详细介绍。

### 17.3.1 RTC 数据寄存器 RTCDR (偏移: 00h)

| 比特   | 名称    | 属性 | 复位值 | 描述               |
|------|-------|----|-----|------------------|
| 31:0 | RTCDR | R  | 0x0 | 用于返回当前的 RTC 计数值。 |



### 17.3.2 RTC 匹配寄存器 RTCMR (偏移: 04h)

| 比特   | 名称    | 属性  | 复位值 | 描述  |
|------|-------|-----|-----|---|
| 31:0 | RTCMR | R/W | 0x0 | 用于配置 RTC 匹配值。写入此寄存器会加载 RTC 匹配值, 读取寄存器会返回最后写入的匹配值。当中断使能后, 系统会将此加载值与 Clk1hz 域中的计数器值比较以产生中断。 |

### 17.3.3 RTC 加载寄存器 RTCLR (偏移: 08h)

| 比特   | 名称    | 属性  | 复位值 | 描述  |
|------|-------|-----|-----|---|
| 31:0 | RTCLR | R/W | 0x0 | 用于配置 RTC 加载值。写入该寄存器会把更新值加载到 RTC 中。读取寄存器会返回最后写入的值。 |

### 17.3.4 RTC 控制寄存器 RTCCR (偏移: 0Ch)

| 比特   | 名称    | 属性  | 复位值 | 描述   |
|------|-------|-----|-----|--|
| 31:1 | RSV   | -   | -   | 保留   |
| 0    | RTCCR | R/W | 0x0 | 用于控制 RTC 加载值或者匹配值的更新。<br>0: RTC 忽略对 RTC 加载寄存器和匹配寄存器的任何访问, 并且使用最后设置的加载值和匹配值<br>1: 允许访问 RTC 加载寄存器和匹配寄存器。 |

### 17.3.5 RTC 中断使能寄存器 RTCIMSC (偏移: 10h)

| 比特   | 名称      | 属性  | 复位值 | 描述                               |
|------|---------|-----|-----|----------------------------------|
| 31:1 | RSV     | -   | -   | 保留                               |
| 0    | RTCIMSC | R/W | 0x0 | RTC 中断使能位。<br>0: 禁止中断<br>1: 使能中断 |

### 17.3.6 RTC 原始中断状态寄存器 RTCRIS (偏移: 14h)

| 比特   | 名称  | 属性 | 复位值 | 描述 |
|------|-----|----|-----|----|
| 31:1 | RSV | -  | -   | 保留 |

| 比特 | 名称     | 属性 | 复位值 | 描述                                     |
|----|--------|----|-----|--|
| 0  | RTCRIS | R  | 0x0 | 屏蔽中断前的原始中断状态位。<br>0: 无中断挂起<br>1: 有中断挂起 |

### 17.3.7 RTC 屏蔽中断状态寄存器 RTCMIS (偏移: 18h)

| 比特   | 名称     | 属性 | 复位值 | 描述   |
|------|--------|----|-----|--|
| 31:1 | RSV    | -  | -   | 保留   |
| 0    | RTCMIS | R  | 0x0 | 反映 RTC 屏蔽中断后的中断状态。<br>0: 无中断挂起<br>1: 有中断挂起 |

### 17.3.8 RTC 中断清除寄存器 RTCICR (偏移: 1Ch)

| 比特   | 名称     | 属性 | 复位值 | 描述                                      |
|------|--------|----|-----|---|
| 31:1 | RSV    | -  | -   | 保留                                      |
| 0    | RTCICR | W  | 0x0 | 用于清除 RTC 中断。<br>0: 无效操作<br>1: 清除 RTC 中断 |

## 17.4 使用流程

### 17.4.1 RTC 配置流程

下图为 RTC 配置流程:

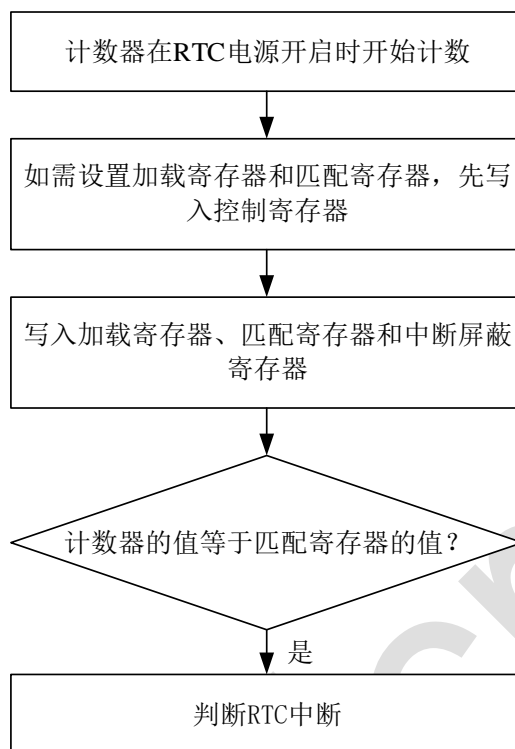


图 17-1： RTC 配置流程

注：

- 当 CPU 同时写入 RTCLR 和 RTCMR 时，RTCDR 的值将在两个 Clk1hz 周期后与 RTCLR 的值相同。RTCMR 的值不会设置。请勿如此操作。
- 当 CPU 仅写入 RTCLR 时，RTCDR 的值将在一个 Clk1hz 周期后与 RTCLR 值相同。
- 建议：首先写入 RTCLR，然后在两个 Clk1hz 周期后再写入 RTCMR。
- 若使用 RTC 匹配中断唤醒低功耗模式（deep sleep mode），在配置完 RTC 以后 RTC 模块需要一点时间准备，可适当延时（延时 2 秒钟），然后再进入低功耗模式。

## 17.4.2 RTC 设置闹钟

1. 配置 SCU\_CTRL0 寄存器，32kHz RCL 时钟，RTC 模块内部自动分频成 1Hz 时钟。
2. 配置 RTCLR 寄存器，设定 RTC 计数初始值。
3. 配置 RTCCR 寄存器，更新 RTC 计数器计数值。
4. 配置 RTCMR 寄存器，设定 RTC 闹钟匹配值。
5. 写 RTCICR 寄存器清除 RTC 中断。
6. 配置 RTCIMSC 寄存器，使能 RTC 中断。
7. RTC 中断触发后，配置 RTCICR 寄存器清除中断。

# 18 DMA

## 18.1 概述

直接存储器访问(DMA)，支持 2 通道数据传输。

## 18.2 主要特性

- 支持单 MASTER 口。
- 可以控制 FLASH、SRAM、SPI0、SPI1、UART1、ADC 模块之间的数据传输，其中 FLASH 仅可以作为源地址。
- 支持 Memory to Memory 模式、Memory to Peripheral 模式、Peripheral to Memory 模式、Peripheral to Peripheral 模式。
- 内部含有 2 个 DMA 通道。
- 数据传输的位宽可设、传输的 Block 长度可设。
- 内部含有深度为 16 的 FIFO。
- Block 最大长度可设为 4096。
- 支持源地址不变传输、递增传输。支持目的地址的不变传输、递增传输。

## 18.3 硬件握手信号

8 个硬件握手信号如下表所示：

表 18-1：硬件握手信号

| No. | Handshaking signal 握手信号 |
|-----|-------------------------|
| 0   | SPI0 TX                 |
| 1   | SPI0 RX                 |
| 2   | SPI1 TX                 |
| 3   | SPI1 RX                 |
| 4   | UART1 TX                |
| 5   | UART1 RX                |
| 6   | ADC RX                  |
| 7   | GPIOA GTimer0 中断触发      |

## 18.4 寄存器描述

DMA 寄存器基地址：0x40020000

表 18-2: DMA 寄存器列表

| 偏置    | 名称            | 描述                    |
|-------|---------------|-----------------------|
| 0x00  | SAR0          | 通道 0 源地址寄存器           |
| 0x08  | DAR0          | 通道 0 目的地址寄存器          |
| 0x18  | CTL0L         | 通道 0 低位控制寄存器          |
| 0x1C  | CTL0H         | 通道 0 高位控制寄存器          |
| 0x40  | CFG0L         | 通道 0 低位设置寄存器          |
| 0x44  | CFG0H         | 通道 0 高位设置寄存器          |
| 0x58  | SAR1          | 通道 1 源地址寄存器           |
| 0x60  | DAR1          | 通道 1 目的地址寄存器          |
| 0x70  | CTL1L         | 通道 1 低位控制寄存器          |
| 0x74  | CTL1H         | 通道 1 高位控制寄存器          |
| 0x98  | CFG1L         | 通道 1 低位设置寄存器          |
| 0x9C  | CFG1H         | 通道 1 高位设置寄存器          |
| 0x2c0 | RawTfr        | 原始传输中断寄存器             |
| 0x2c8 | RawBlock      | 原始 Block 传输中断寄存器      |
| 0x2d0 | RawSrcTran    | 原始源传输中断寄存器            |
| 0x2D8 | RawDstTran    | 原始目标传输中断寄存器           |
| 0x2E0 | RawErr        | 原始错误中断寄存器             |
| 0x2E8 | StatusTfr     | 传输中断状态寄存器             |
| 0x2F0 | StatusBlock   | Block 传输中断状态寄存器       |
| 0x2F8 | StatusSrcTran | 源传输中断状态寄存器            |
| 0x300 | StatusDstTran | 目标传输中断状态寄存器           |
| 0x308 | StatusErr     | 错误中断状态寄存器             |
| 0x310 | MaskTfr       | 传输中断屏蔽寄存器             |
| 0x318 | MaskBlock     | Block 传输中断屏蔽寄存器       |
| 0x320 | MaskSrcTran   | 源传输中断屏蔽寄存器            |
| 0x328 | MaskDstTran   | 目标传输中断屏蔽寄存器           |
| 0x330 | MaskErr       | 错误中断屏蔽寄存器             |
| 0x338 | ClearTfr      | 传输中断清除寄存器             |
| 0x340 | ClearBlock    | Block 传输中断清除寄存器       |
| 0x348 | ClearSrcTran  | 源传输中断清除寄存器            |
| 0x350 | ClearDstTran  | 目标传输中断清除寄存器           |
| 0x358 | ClearErr      | 错误中断清除寄存器             |
| 0x360 | StatusInt     | 中断状态寄存器               |
| 0x368 | ReqSrc        | 源传输 Req 信号软件握手寄存器     |
| 0x370 | ReqDst        | 目标传输 Req 信号软件握手寄存器    |
| 0x378 | SglReqSrc     | 源传输 Single 信号软件握手寄存器  |
| 0x380 | SglReqDst     | 目标传输 Single 信号软件握手寄存器 |
| 0x388 | LstSrc        | 源传输 Last 信号软件握手寄存器    |
| 0x390 | LstDst        | 目标传输 Last 信号软件握手寄存器   |
| 0x398 | DmaCfg        | DMA 模块使能寄存器           |
| 0x3a0 | ChEn          | 通道使能寄存器               |
| 0x3B0 | DmaTest       | 测试模式寄存器               |

### 18.4.1 源地址寄存器 SAR0 (偏移: 00h)

| 比特   | 名称   | 属性  | 默认值 | 功能描述            |
|------|------|-----|-----|-----------------|
| 31:0 | SAR0 | R/W | 0x0 | DMA传输源地址, 会自动更新 |

### 18.4.2 目的地址寄存器 DAR0 (偏移: 08h)

| 比特   | 名称   | 属性  | 默认值 | 功能描述             |
|------|------|-----|-----|------------------|
| 31:0 | DAR0 | R/W | 0x0 | DMA传输目的地址, 会自动更新 |

### 18.4.3 控制寄存器 CTL0L (偏移: 18h)

| 比特    | 名称           | 属性  | 默认值 | 功能描述   |
|-------|--------------|-----|-----|--|
| 31:22 | RSV          | -   | -   | 保留   |
| 21:20 | TT_FC        | R/W | 0x3 | 传输类型与流控制:<br>0x0: 储存器到储存器<br>0x1: 储存器到外设<br>0x2: 外设到储存器<br>0x3: 外设到外设  |
| 19:16 | RSV          | -   | -   | 保留   |
| 15:14 | SRC_MSIZ     | R/W | 0x1 | 源传输 Burst 长度, 每次源外设 Req 握手信号可用时读取数据个数:<br>0x0: 1<br>0x1: 4<br>0x2: 8   |
| 13    | RSV          | -   | -   | 保留   |
| 12:11 | DST_MSIZ     | R/W | 0x1 | 目标传输 Burst 长度, 每次目标外设 Req 握手信号可用时写入数据个数:<br>0x0: 1<br>0x1: 4<br>0x2: 8 |
| 10:9  | SINC         | R/W | 0x0 | 源地址递增:<br>0x0: 递增<br>0x1: 递减<br>0x2/0x3: 不变                            |
| 8:7   | DINC         | R/W | 0x0 | 目标地址递增:<br>0x0: 递增<br>0x1: 递减<br>0x2/0x3: 不变                           |
| 6     | RSV          | -   | -   | 保留   |
| 5:4   | SRC_TR_WIDTH | R/W | 0x2 | 源传输数据宽度:<br>0x0: 8bits<br>0x1: 16bits<br>0x2: 32bits                   |
| 3     | RSV          | -   | -   | 保留   |

| 比特  | 名称           | 属性  | 默认值 | 功能描述  |
|-----|--------------|-----|-----|---|
| 2:1 | DST_TR_WIDTH | R/W | 0x2 | 目标传输数据宽度：<br>0x0: 8bits<br>0x1: 16bits<br>0x2: 32bits |
| 0   | INT_EN       | R/W | 0x1 | 中断使能：<br>0x0: 不使能<br>0x1: 使能                          |

#### 18.4.4 控制寄存器 CTL0H (偏移: 1Ch)

| 比特    | 名称       | 属性  | 默认值 | 功能描述          |
|-------|----------|-----|-----|---------------|
| 31:12 | RSV      | -   | -   | 保留            |
| 11:0  | BLOCK_TS | R/W | 0x2 | Block 传输源数据个数 |

#### 18.4.5 设置寄存器 CFG0L (偏移: 40h)

| 比特    | 名称         | 属性  | 默认值 | 功能描述  |
|-------|------------|-----|-----|---|
| 31    | RELOAD_DST | R/W | 0   | 自动重启目标传输：<br>0x0: 关闭<br>0x1: 开启: 当一次 Block 传输结束时，<br>将 DARx 重设为初始值，并开始一次新的 Block 传输 |
| 30    | RELOAD_SRC | R/W | 0   | 自动重启源传输：<br>0x0: 关闭<br>0x1: 开启: 当一次 Block 传输结束时，<br>将 SARx 重设为初始值，并开始一次新的 Block 传输  |
| 29:20 | RSV        | -   | -   | 保留  |
| 19    | SRC_HS_POL | R/W | 0   | 请使用 0x0: 握手信号高有效  |
| 18    | DST_HS_POL | R/W | 0   | 请使用 0x0: 握手信号高有效  |
| 17:12 | RSV        | -   | -   | 保留  |
| 11    | HS_SEL_SRC | R/W | 1   | 源传输握手信号选择：<br>0x0: 硬件握手<br>0x1: 软件握手  |
| 10    | HS_SEL_DST | R/W | 1   | 目标传输握手信号选择：<br>0x0: 硬件握手<br>0x1: 软件握手   |
| 9     | FIFO_EMPTY | R   | 1   | 指示该通道的 FIFO 是否为空：<br>0x0: FIFO 非空<br>0x1: FIFO 空                                    |
| 8     | CH_SUSP    | R/W | 0   | 暂停该通道的传输：<br>0x0: 正常传输<br>0x1: 暂停传输   |
| 7:6   | RSV        | -   | -   | 保留  |

| 比特  | 名称       | 属性  | 默认值                    | 功能描述          |
|-----|----------|-----|------------------------|---------------|
| 5   | CH_PRIOR | R/W | CFG0: 0x0<br>CFG1: 0x1 | 指定通道优先级, 0为最低 |
| 4:0 | RSV      | -   | -                      | 保留            |

#### 18.4.6 设置寄存器 CFG0H (偏移: 44h)

| 比特    | 名称        | 属性  | 默认值 | 功能描述  |
|-------|-----------|-----|-----|---|
| 31:14 | RSV       | -   | -   | 保留  |
| 13:11 | DEST_PER  | R/W | 0x0 | 目标外设握手信号编号:<br>UM0068 中的编号:<br>0x0: SPI0 TX<br>0x1: SPI0 RX<br>0x2: SPI1 TX<br>0x3: SPI1 RX<br>0x4: UART1 TX<br>0x5: UART1 RX<br>0x6: ADC RX<br>0x7: GPIO0 或 GTIMER 中断信号<br>在 TT_FC 中指定目标为储存器时此项被忽略 |
| 10    | RSV       | -   | -   | 保留  |
| 9:7   | SRC_PER   | R/W | 0x0 | 源外设握手信号编号:<br>UM0068 中的编号:<br>0x0: SPI0 TX<br>0x1: SPI0 RX<br>0x2: SPI1 TX<br>0x3: SPI1 RX<br>0x4: UART1 TX<br>0x5: UART1 RX<br>0x6: ADC RX<br>0x7: GPIO0 或 GTIMER 中断信号<br>在 TT_FC 中指定源为储存器时此项被忽略   |
| 6:5   | RSV       | -   | -   | 保留  |
| 4:2   | PROTCTL   | R/W | 0x1 | 驱动 HPROT[3:1]   |
| 1     | FIFO_MODE | R/W | 0   | 指定需要多少可用数据/空间起一次 Burst 传输:<br>0x0: 只要有一个可用数据/空间就会发起传输<br>0x1: 可用数据大于或等于一半 FIFO 深度时发起对目标的传输, 空间大于或等于一半 FIFO 深度时发起源传输; 在一次 Burst 或 Block 传输的末尾时例外   |
| 0     | FCMODE    | R/W | 0   | 流控模式/数据预读:<br>0x0: 开启预读, 源数据传输可用时读取<br>0x1: 关闭预读, 目标传输完成前不进行源数据传输   |



### 18.4.7 源地址寄存器 SAR1 (偏移: 58h)

| 比特   | 名称   | 属性  | 默认值 | 功能描述            |
|------|------|-----|-----|-----------------|
| 31:0 | SAR1 | R/W | 0x0 | DMA传输源地址, 会自动更新 |

### 18.4.8 目的地址寄存器 DAR1 (偏移: 60h)

| 比特   | 名称   | 属性  | 默认值 | 功能描述             |
|------|------|-----|-----|------------------|
| 31:0 | DAR1 | R/W | 0x0 | DMA传输目的地址, 会自动更新 |

### 18.4.9 控制寄存器 CTL1L (偏移: 70h)

| 比特    | 名称           | 属性  | 默认值 | 功能描述   |
|-------|--------------|-----|-----|--|
| 31:22 | RSV          | -   | -   | 保留   |
| 21:20 | TT_FC        | R/W | 0x3 | 传输类型与流控制:<br>0x0: 储存器到储存器<br>0x1: 储存器到外设<br>0x2: 外设到储存器<br>0x3: 外设到外设  |
| 19:16 | RSV          | -   | -   | 保留   |
| 15:14 | SRC_MSIZE    | R/W | 0x1 | 源传输 Burst 长度, 每次源外设 Req 握手信号可用时读取数据个数:<br>0x0: 1<br>0x1: 4<br>0x2: 8   |
| 13    | RSV          | -   | -   | 保留   |
| 12:11 | DST_MSIZE    | R/W | 0x1 | 目标传输 Burst 长度, 每次目标外设 Req 握手信号可用时写入数据个数:<br>0x0: 1<br>0x1: 4<br>0x2: 8 |
| 10:9  | SINC         | R/W | 0x0 | 源地址递增:<br>0x0: 递增<br>0x1: 递减<br>0x2/0x3: 不变                            |
| 8:7   | DINC         | R/W | 0x0 | 目标地址递增:<br>0x0: 递增<br>0x1: 递减<br>0x2/0x3: 不变                           |
| 6     | RSV          | -   | -   | 保留   |
| 5:4   | SRC_TR_WIDTH | R/W | 0x2 | 源传输数据宽度:<br>0x0: 8bits<br>0x1: 16bits<br>0x2: 32bits                   |
| 3     | RSV          | -   | -   | 保留   |

| 比特  | 名称           | 属性  | 默认值 | 功能描述  |
|-----|--------------|-----|-----|---|
| 2:1 | DST_TR_WIDTH | R/W | 0x2 | 目标传输数据宽度：<br>0x0: 8bits<br>0x1: 16bits<br>0x2: 32bits |
| 0   | INT_EN       | R/W | 0x1 | 中断使能：<br>0x0: 不使能<br>0x1: 使能                          |

#### 18.4.10 控制寄存器 CTL1H (偏移: 74h)

| 比特    | 名称       | 属性  | 默认值 | 功能描述         |
|-------|----------|-----|-----|--------------|
| 31:12 | RSV      | -   | -   | 保留           |
| 11:0  | BLOCK_TS | R/W | 0x2 | Block传输源数据个数 |

#### 18.4.11 设置寄存器 CFG1L (偏移: 98h)

| 比特    | 名称         | 属性  | 默认值 | 功能描述  |
|-------|------------|-----|-----|---|
| 31    | RELOAD_DST | R/W | 0   | 自动重启目标传输：<br>0x0: 关闭<br>0x1: 开启：当一次 Block 传输结束时，<br>将 DARx 重设为初始值，并开始一次新的<br>Block 传输 |
| 30    | RELOAD_SRC | R/W | 0   | 自动重启源传输：<br>0x0: 关闭<br>0x1: 开启：当一次 Block 传输结束时，<br>将 SARx 重设为初始值，并开始一次新的<br>Block 传输  |
| 29:20 | RSV        | -   | -   | 保留  |
| 19    | SRC_HS_POL | R/W | 0   | 请使用 0x0: 握手信号高有效  |
| 18    | DST_HS_POL | R/W | 0   | 请使用 0x0: 握手信号高有效  |
| 17:12 | RSV        | -   | -   | 保留  |
| 11    | HS_SEL_SRC | R/W | 1   | 源传输握手信号选择：<br>0x0: 硬件握手<br>0x1: 软件握手  |
| 10    | HS_SEL_DST | R/W | 1   | 目标传输握手信号选择：<br>0x0: 硬件握手<br>0x1: 软件握手   |
| 9     | FIFO_EMPTY | R   | 1   | 指示该通道的 FIFO 是否为空：<br>0x0: FIFO 非空<br>0x1: FIFO 空                                      |
| 8     | CH_SUSP    | R/W | 0   | 暂停该通道的传输：<br>0x0: 正常传输<br>0x1: 暂停传输   |
| 7:6   | RSV        | -   | -   | 保留  |

| 比特  | 名称       | 属性  | 默认值                | 功能描述          |
|-----|----------|-----|--------------------|---------------|
| 5   | CH_PRIOR | R/W | CFG0: 0<br>CFG1: 1 | 指定通道优先级, 0为最低 |
| 4:0 | RSV      | -   | -                  | 保留            |

### 18.4.12 设置寄存器 CFG1H (偏移: 9Ch)

| 比特    | 名称        | 属性  | 默认值 | 功能描述  |
|-------|-----------|-----|-----|---|
| 31:13 | RSV       | -   | -   | 保留  |
| 12:10 | DEST_PER  | R/W | 0x0 | 目标外设握手信号编号:<br>UM0068 中的编号:<br>0x0: SPI0 TX<br>0x1: SPI0 RX<br>0x2: SPI1 TX<br>0x3: SPI1 RX<br>0x4: UART1 TX<br>0x5: UART1 RX<br>0x6: ADC RX<br>0x7: GPIO0 或 GTIMER 中断信号<br>在 TT_FC 中指定目标为储存器时此项被忽略 |
| 9     | RSV       | -   | -   | 保留  |
| 8:6   | SRC_PER   | R/W | 0x0 | 源外设握手信号编号:<br>UM0068 中的编号:<br>0x0: SPI0 TX<br>0x1: SPI0 RX<br>0x2: SPI1 TX<br>0x3: SPI1 RX<br>0x4: UART1 TX<br>0x5: UART1 RX<br>0x6: ADC RX<br>0x7: GPIO0 或 GTIMER 中断信号<br>在 TT_FC 中指定源为储存器时此项被忽略   |
| 5:4   | RSV       | -   | -   | 保留  |
| 3:1   | PROTCTL   | R/W | 0x1 | 驱动 HPROT[3:1]   |
| 0     | FIFO_MODE | R/W | 0   | 指定需要多少可用数据/空间起一次 Burst 传输:<br>0x0: 只要有一个可用数据/空间就会发起传输<br>0x1: 可用数据大于或等于一半 FIFO 深度时发起对目标的传输, 空间大于或等于一半 FIFO 深度时发起源传输; 在一次 Burst 或 Block 传输的末尾时例外   |
| 32    | FCMODE    | R/W | 0x0 | 流控模式/数据预读:<br>0x0: 开启预读, 源数据传输可用时读取<br>0x1: 关闭预读, 目标传输完成前不进行源数据传输   |

### 18.4.13 原始传输中断寄存器 RawTfr (偏移: 2C0h)

| 比特   | 名称     | 属性  | 默认值 | 功能描述   |
|------|--------|-----|-----|--|
| 31:2 | RSV    | -   | -   | 保留   |
| 1:0  | RawTfr | R/W | 0x0 | bit x 为通道 x 的原始 Tfr 中断状态<br>该中断会在该通道的完成所有传输时触发。<br>正常使用时不建议直接写入这个寄存器 |

### 18.4.14 原始 Block 传输中断寄存器 RawBlock (偏移: 2C8h)

| 比特   | 名称       | 属性  | 默认值 | 功能描述   |
|------|----------|-----|-----|--|
| 31:2 | RSV      | -   | -   | 保留   |
| 1:0  | RawBlock | R/W | 0x0 | bit x 为通道 x 的原始 Block 中断状态<br>该中断会在该通道完成一个 Block 传输时触发。<br>正常使用时不建议直接写入这个寄存器 |

### 18.4.15 原始源传输中断寄存器 RawSrcTran (偏移: 2D0h)

| 比特   | 名称         | 属性  | 默认值 | 功能描述  |
|------|------------|-----|-----|---|
| 31:2 | RSV        | -   | -   | 保留  |
| 1:0  | RawSrcTran | R/W | 0x0 | bit x 为通道 x 的原始 SrcTran 中断状态<br>该中断会在该通道完成一个响应源外设握手信号的 Burst/Single 传输时触发。<br>正常使用时不建议直接写入这个寄存器 |

### 18.4.16 原始目标传输中断寄存器 RawDstTran (偏移: 2D8h)

| 比特   | 名称         | 属性  | 默认值 | 功能描述   |
|------|------------|-----|-----|--|
| 31:2 | RSV        | -   | -   | 保留   |
| 1:0  | RawDstTran | R/W | 0x0 | bit x 为通道 x 的原始 DstTran 中断状态<br>该中断会在该通道完成一个响应目标外设握手信号的 Burst/Single 传输时触发。<br>正常使用时不建议直接写入这个寄存器 |

### 18.4.17 原始错误中断寄存器 RawErr (偏移: 2E0h)

| 比特   | 名称  | 属性 | 默认值 | 功能描述 |
|------|-----|----|-----|------|
| 31:2 | RSV | -  | -   | 保留   |

| 比特  | 名称     | 属性  | 默认值 | 功能描述   |
|-----|--------|-----|-----|--|
| 1:0 | RawErr | R/W | 0x0 | bit x 为通道 x 的原始 Err 中断状态<br>该中断会在该通道在传输时收到 HRSP 上的 ERROR 响应时触发, 并会导致传输被取消和该通道被关闭。<br>正常使用时不建议直接写入这个寄存器 |

#### 18.4.18 传输中断状态寄存器 StatusTfr (偏移: 2E8h)

| 比特   | 名称        | 属性 | 默认值 | 功能描述   |
|------|-----------|----|-----|--|
| 31:2 | RSV       | -  | -   | 保留   |
| 1:0  | StatusTfr | R  | 0x0 | bit x 为通道 x 的 Tfr 中断输出状态<br>该中断会在该通道的完成所有传输时触发<br>如果该中断被屏蔽则该寄存器不会被置1 |

#### 18.4.19 Block 传输中断状态寄存器 StatusBlock (偏移: 2F0h)

| 比特   | 名称          | 属性 | 默认值 | 功能描述   |
|------|-------------|----|-----|--|
| 31:2 | RSV         | -  | -   | 保留   |
| 1:0  | StatusBlock | R  | 0x0 | bit x 为通道 x 的 Block 中断输出状态<br>该中断会在该通道完成一个 Block 传输时触发<br>如果该中断被屏蔽则该寄存器不会被置1 |

#### 18.4.20 源传输中断状态寄存器 StatusSrcTran (偏移: 2F8h)

| 比特   | 名称            | 属性 | 默认值 | 功能描述  |
|------|---------------|----|-----|---|
| 31:2 | RSV           | -  | -   | 保留  |
| 1:0  | StatusSrcTran | R  | 0x0 | bit x 为通道 x 的 SrcTran 中断输出状态<br>该中断会在该通道完成一个响应源外设握手信号的 Burst/Single 传输时触发<br>如果该中断被屏蔽则该寄存器不会被置1 |

#### 18.4.21 目标传输中断状态寄存器 StatusDstTran (偏移: 300h)

| 比特   | 名称            | 属性 | 默认值 | 功能描述   |
|------|---------------|----|-----|--|
| 31:2 | RSV           | -  | -   | 保留   |
| 1:0  | StatusDstTran | R  | 0x0 | bit x 为通道 x 的 DstTran 中断输出状态<br>该中断会在该通道完成一个响应目标外设握手信号的 Burst/Single 传输时触发<br>如果该中断被屏蔽则该寄存器不会被置1 |

### 18.4.22 错误中断状态寄存器 StatusErr (偏移: 308h)

| 比特   | 名称        | 属性 | 默认值 | 功能描述   |
|------|-----------|----|-----|--|
| 31:2 | RSV       | -  | -   | 保留   |
| 1:0  | StatusErr | R  | 0x0 | bit x 为通道 x 的 Err 中断输出状态<br>该中断会在该通道在传输时收到 HRSP 上的 ERROR 响应时触发, 并会导致传输被取消和该通道被关闭<br>如果该中断被屏蔽则该寄存器不会被置1 |

### 18.4.23 传输中断屏蔽寄存器 MaskTfr (偏移: 310h)

| 比特    | 名称         | 属性  | 默认值 | 功能描述   |
|-------|------------|-----|-----|--|
| 31:10 | RSV        | -   | -   | 保留   |
| 9:8   | MaskTfr_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV        | -   | -   | 保留   |
| 1:0   | MaskTfr    | R/W | 0x0 | bit x 为通道 x 的 Tfr 中断屏蔽状态<br>0x0: 屏蔽<br>0x1: 不屏蔽            |

使用方法举例:

- 要将 bit 0 置 1, 请对该寄存器写入 0x101, bit 1 不受影响,
- 要将 bit 1 置 0, 请对该寄存器写入 0x200, bit 0 不受影响,
- 要将 bit 0 和 bit 1 置 1, 请对该寄存器写入 0x303。

### 18.4.24 Block 传输中断屏蔽寄存器 MaskBlock (偏移: 318h)

| 比特    | 名称           | 属性  | 默认值 | 功能描述   |
|-------|--------------|-----|-----|--|
| 31:10 | RSV          | -   | -   | 保留   |
| 9:8   | MaskBlock_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV          | -   | -   | 保留   |
| 1:0   | MaskBlock    | R/W | 0x0 | bit x 为通道 x 的 Block 中断屏蔽状态<br>0x0: 屏蔽<br>0x1: 不屏蔽          |

### 18.4.25 源传输中断屏蔽寄存器 MaskSrcTran (偏移: 320h)

| 比特    | 名称             | 属性  | 默认值 | 功能描述   |
|-------|----------------|-----|-----|--|
| 31:10 | RSV            | -   | -   | 保留   |
| 9:8   | MaskSrcTran_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV            | -   | -   | 保留   |
| 1:0   | MaskSrcTran    | R/W | 0x0 | bit x 为通道 x 的 SrcTran 中断屏蔽状态<br>0x0: 屏蔽<br>0x1: 不屏蔽        |

### 18.4.26 目标传输中断屏蔽寄存器 MaskDstTran (偏移: 328h)

| 比特    | 名称             | 属性  | 默认值 | 功能描述   |
|-------|----------------|-----|-----|--|
| 31:10 | RSV            | -   | -   | 保留   |
| 9:8   | MaskDstTran_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV            | -   | -   | 保留   |
| 1:0   | MaskDstTran    | R/W | 0x0 | bit x 为通道 x 的 DstTran 中断屏蔽状态<br>0x0: 屏蔽<br>0x1: 不屏蔽        |

### 18.4.27 错误中断屏蔽寄存器 MaskErr (偏移: 330h)

| 比特    | 名称         | 属性  | 默认值 | 功能描述   |
|-------|------------|-----|-----|--|
| 31:10 | RSV        | -   | -   | 保留   |
| 9:8   | MaskErr_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV        | -   | -   | 保留   |
| 1:0   | MaskErr    | R/W | 0x0 | bit x 为通道 x 的 Err 中断屏蔽状态<br>0x0: 屏蔽<br>0x1: 不屏蔽            |

### 18.4.28 传输中断清除寄存器 ClearTfr (偏移: 338h)

| 比特   | 名称       | 属性 | 默认值 | 功能描述                   |
|------|----------|----|-----|------------------------|
| 31:2 | RSV      | -  | -   | 保留                     |
| 1:0  | ClearTfr | W  | 0x0 | 对bit x写入1可以清除通道x的Tfr中断 |

### 18.4.29 Block 传输中断清除寄存器 ClearBlock (偏移：340h)

| 比特   | 名称         | 属性 | 默认值 | 功能描述                     |
|------|------------|----|-----|--------------------------|
| 31:2 | RSV        | -  | -   | 保留                       |
| 1:0  | ClearBlock | W  | 0x0 | 对bit x写入1可以清除通道x的Block中断 |

### 18.4.30 源传输中断清除寄存器 ClearSrcTran (偏移：348h)

| 比特   | 名称           | 属性 | 默认值 | 功能描述                       |
|------|--------------|----|-----|----------------------------|
| 31:2 | RSV          | -  | -   | 保留                         |
| 1:0  | ClearSrcTran | W  | 0x0 | 对bit x写入1可以清除通道x的SrcTran中断 |

### 18.4.31 目标传输中断清除寄存器 ClearDstTran (偏移：350h)

| 比特   | 名称           | 属性 | 默认值 | 功能描述                       |
|------|--------------|----|-----|----------------------------|
| 31:2 | RSV          | -  | -   | 保留                         |
| 1:0  | ClearDstTran | W  | 0x0 | 对bit x写入1可以清除通道x的DstTran中断 |

### 18.4.32 错误中断清除寄存器 ClearErr (偏移：358h)

| 比特   | 名称       | 属性 | 默认值 | 功能描述                   |
|------|----------|----|-----|------------------------|
| 31:2 | RSV      | -  | -   | 保留                     |
| 1:0  | ClearErr | W  | 0x0 | 对bit x写入1可以清除通道x的Err中断 |

### 18.4.33 中断状态寄存器 StatusInt (偏移：360h)

| 比特   | 名称      | 属性 | 默认值 | 功能描述                     |
|------|---------|----|-----|--------------------------|
| 31:5 | RSV     | -  | -   | 保留                       |
| 4    | Err     | R  | 0x0 | StatusErr各个bit或运算的结果     |
| 3    | DstTran | R  | 0x0 | StatusDstTran各个bit或运算的结果 |
| 2    | SrcTran | R  | 0x0 | StatusSrcTran各个bit或运算的结果 |
| 1    | Block   | R  | 0x0 | StatusBlock各个bit或运算的结果   |
| 0    | Tfr     | R  | 0x0 | StatusTfr各个bit或运算的结果     |

### 18.4.34 源传输 Req 信号软件握手寄存器 ReqSrc (偏移：368h)

| 比特    | 名称  | 属性 | 默认值 | 功能描述 |
|-------|-----|----|-----|------|
| 31:10 | RSV | -  | -   | 保留   |



| 比特  | 名称         | 属性  | 默认值 | 功能描述   |
|-----|------------|-----|-----|--|
| 9:8 | SRC_REQ_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2 | RSV        | -   | -   | 保留   |
| 1:0 | SRC_REQ    | R/W | 0x0 | bit x为通道x的源传输Req握手信号                                       |

#### 18.4.35 目标传输 Req 信号软件握手寄存器 ReqDst (偏移: 370h)

| 比特    | 名称         | 属性  | 默认值 | 功能描述   |
|-------|------------|-----|-----|--|
| 31:10 | RSV        | -   | -   | 保留   |
| 9:8   | DST_REQ_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV        | -   | -   | 保留   |
| 1:0   | DST_REQ    | R/W | 0x0 | bit x为通道x的目标传输Req握手信号                                      |

#### 18.4.36 源传输 Single 信号软件握手寄存器 SglReqSrc (偏移: 378h)

| 比特    | 名称            | 属性  | 默认值 | 功能描述   |
|-------|---------------|-----|-----|--|
| 31:10 | RSV           | -   | -   | 保留   |
| 9:8   | SRC_SGLREQ_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV           | -   | -   | 保留   |
| 1:0   | SRC_SGLREQ    | R/W | 0x0 | bit x为通道x的源传输Single握手信号                                    |

#### 18.4.37 目标传输 Single 信号软件握手寄存器 SglReqDst (偏移: 380h)

| 比特    | 名称            | 属性  | 默认值 | 功能描述   |
|-------|---------------|-----|-----|--|
| 31:10 | RSV           | -   | -   | 保留   |
| 9:8   | DST_SGLREQ_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV           | -   | -   | 保留   |
| 1:0   | DST_SGLREQ    | R/W | 0x0 | bit x为通道x的目标传输Single握手信号                                   |

### 18.4.38 源传输 Last 信号软件握手寄存器 LstSrc (偏移: 388h)

| 比特    | 名称        | 属性  | 默认值 | 功能描述   |
|-------|-----------|-----|-----|--|
| 31:10 | RSV       | -   | -   | 保留   |
| 9:8   | LSTSRC_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV       | -   | -   | 保留   |
| 1:0   | LSTSRC    | R/W | 0x0 | bit x为通道x的源传输Last握手信号                                      |

### 18.4.39 目标传输 Last 信号软件握手寄存器 LstDst (偏移: 390h)

| 比特    | 名称        | 属性  | 默认值 | 功能描述   |
|-------|-----------|-----|-----|--|
| 31:10 | RSV       | -   | -   | 保留   |
| 9:8   | LSTDST_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV       | -   | -   | 保留   |
| 1:0   | LSTDST    | R/W | 0x0 | bit x为通道x的目标传输Last握手信号                                     |

### 18.4.40 DMA 模块使能寄存器 DmaCfg (偏移: 398h)

| 比特   | 名称     | 属性  | 默认值 | 功能描述   |
|------|--------|-----|-----|--|
| 31:1 | RSV    | -   | -   | 保留   |
| 0    | DMA_EN | R/W | 0x0 | 0x0: DMA 模块功能不使能<br>0x1: DMA模块功能使能, 要在使能通道之前使能此位 |

### 18.4.41 通道使能寄存器 ChEn (偏移: 3A0h)

| 比特    | 名称       | 属性  | 默认值 | 功能描述   |
|-------|----------|-----|-----|--|
| 31:10 | RSV      | -   | -   | 保留   |
| 9:8   | CH_EN_WE | W   | 0x0 | bit (x+8)为通道 x 的写入激活位<br>0x0: 不写入<br>0x1: 写入<br>当次有效, 读取为0 |
| 7:2   | RSV      | -   | -   | 保留   |
| 1:0   | CH_EN    | R/W | 0x0 | bit x 为通道 x 的使能位<br>传输完成后此位会自动置0                           |

使用方法举例:

- 要将 bit 0 置 1, 请对该寄存器写入 0x101, bit 1 不受影响,

- 要将 bit 1 置 0，请对该寄存器写入 0x200，bit 0 不受影响，
- 要将 bit 0 和 bit 1 置 1，请对该寄存器写入 0x303。

#### 18.4.42 测试模式寄存器 DmaTest (偏移：3B0h)

| 比特   | 名称          | 属性  | 默认值 | 功能描述  |
|------|-------------|-----|-----|---|
| 31:1 | RSV         | -   | -   | 保留  |
| 0    | TEST_SLV_IF | R/W | 0   | 0x0：普通模式<br>0x1：测试模式，开启测试模式会使 DMA 模块所有可写的寄存器在被读取时总是返回写入的值 |

### 18.5 使用流程

#### 18.5.1 基本硬件流控使用方法

1. 配置 SARx 寄存器，指定源地址。
2. 配置 DARx 寄存器，指定目标地址。
3. 配置 CTLx 寄存器。有需要时使能中断，设置传输数据宽度和大小，选择流控类型和传输模式，注意 SRAM 和 FLASH 没有握手信号，在 TT\_FC 中将 SRAM 和 FLASH 认定为存储器选择地址是否递增/递减，选择 burst 传输长度。
4. 配置 CFGx 寄存器。选择通道优先级，设置是否自动重启传输，选定握手信号，将 HS\_SEL 域设为 0。
5. 有需要时配置 MaskBlock 寄存器，开启对应通道的中断。
6. 配置 DmaCfgReg 寄存器使能 DMAC。
7. 配置 ChEnReg 寄存器使能通道。
8. 等待中断或查询 ChEnReg。
9. 清除中断。

#### 18.5.2 软件流控使用方法

1. 配置 SARx 寄存器，指定源地址。
2. 配置 DARx 寄存器，指定目标地址。
3. 配置 CTLx 寄存器。有需要时使能中断，设置传输数据宽度和大小，选择流控类型和传输模式，选择地址是否递增/递减，选择 burst 传输长度。
4. 配置 CFGx 寄存器。选择通道优先级，将 HS\_SEL 域设为 1。
5. 有需要时配置 MaskBlock 寄存器，开启对应通道的中断。

6. 配置 DmaCfgReg 寄存器使能 DMAC。
7. 配置 ChEnReg 寄存器使能通道。
8. 等待中断或查询 ChEnReg。
9. 清除中断。

### 18.5.3 GPIOA 和 GTimer0 中断启动 DMA 传输流程

#### GPIOA 中断信号启动 DMA 搬移：

1. 使能系统寄存器 SYSCTRL0.DMA\_C7HS\_EN 位，使能 DMA 第七握手信号。
2. 配置系统寄存器 SYSCTRL0.DMA\_C7HS\_G0GT\_SEL 位，选择 GPIOA 中断信号启动 DMA 搬移。
3. 配置 GPIOA 低电平触发中断(使能中断，但不注册中断)。
4. 配置 SARx 寄存器，指定源地址。
5. 配置 DARx 寄存器，指定目标地址。
6. 配置 CTLx 寄存器。有需要时使能中断，设置传输数据宽度和大小，选择流控类型，传输模式(注意 GPIOA 中断信号作为握手信号时，当作 Peripheral)，在 TT\_FC 中将 SRAM 和 FLASH 认定为存储器选择地址是否递增/递减，选择 burst 传输长度。
7. 配置 CFGx 寄存器。选择通道优先级，设置是否自动重启传输，用第七握手信号控制 SRAM 和 FLASH 的地址端，将 HS\_SEL 域设为 0。
8. 配置 MaskBlock 寄存器，开启对应通道的中断。
9. 配置 DmaCfgReg 寄存器使能 DMAC。
10. 配置 ChEnReg 寄存器使能通道。
11. 等待中断或查询 ChEnReg。
12. 清除中断。

#### GTIMER0 中断信号启动 DMA 搬移：

1. 使能系统寄存器 SYSCTRL0.DMA\_C7HS\_EN 位，使能 DMA 第七握手信号。
2. 配置系统寄存器 SYSCTRL0.DMA\_C7HS\_G0GT\_SEL 位，选择 GTIMER0 中断信号启动 DMA 搬移。
3. 配置 GTIMER0 定时中断(使能中断，但不注册中断)。
4. 配置 SARx 寄存器，指定源地址。
5. 配置 DARx 寄存器，指定目标地址。
6. 配置 CTLx 寄存器。有需要时使能中断，设置传输数据宽度和大小，选择流控类型，传输模式(注意 GTIMER0 中断信号作为握手信号时，当作 Peripheral)，在 TT\_FC 中将 SRAM 和

FLASH 认定为存储器选择地址是否递增/递减，选择 burst 传输长度。

7. 配置 CFGx 寄存器。选择通道优先级，设置是否自动重启传输，用第七握手信号控制 SRAM 和 FLASH 的地址端，将 HS\_SEL 域设为 0。
8. 配置 MaskBlock 寄存器，开启对应通道的中断。
9. 配置 DmaCfgReg 寄存器使能 DMAC。
10. 配置 ChEnReg 寄存器使能通道。
11. 等待中断或查询 ChEnReg。
12. 清除中断。

# 19 SysTick

## 19.1 概述

OS 要想支持多任务，就需要周期执行上下文切换，这样就需要有定时器之类的硬件资源打断程序执行。当定时器中断产生时，处理器就会在异常处理中进行 OS 任务调度，同时还会进行 OS 维护的工作。Cortex-M0+处理器中有一个称为 SysTick 的简单定时器，用于产生周期性的中断请求。

SysTick 为 24 位的定时器，并且向下计数。定时器的计数减到 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 异常（异常编号为 15），该异常事件会引起 SysTick 异常处理的执行，这个过程是 OS 的一部分。

对于不需要 OS 的系统，SysTick 定时器也可以用作其他用途，比如定时、计时或者为需要周期执行的任务提供中断源。SysTick 异常的产生是可控的，如果异常被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询溢出标志。

## 19.2 寄存器描述

SysTick 寄存器基地址：0xE000E010

表 19-1: SysTick 寄存器列表

| 偏置   | 名称                        | 描述               |
|------|---------------------------|------------------|
| 0x00 | SYS_CSR<br>(SysTick_CTRL) | SysTick 控制和状态寄存器 |
| 0x04 | SYS_RVR<br>(SysTick_LOAD) | SysTick 重载值寄存器   |
| 0x08 | SYS_CVR<br>(SysTick_VAL)  | SysTick 当前值寄存器   |

### 19.2.1 控制和状态寄存器 SYS\_CSR (偏移: 00h)

| 比特    | 名称        | 属性  | 复位值 | 描述   |
|-------|-----------|-----|-----|--|
| 31:17 | RSV       | -   | -   | 保留   |
| 16    | COUNTFLAG | R   | 0   | Systick定时器溢出标志<br>1: Systick定时器发生下溢出。<br>0: Systick定时器未发生溢出。<br>读该寄存器，可清除 COUNTFLAG 标志 |
| 15:3  | RSV       | -   | -   | 保留   |
| 2     | CLKSOURCE | R/W | 1   | SysTick时钟源选择<br>1: HCLK<br>0: 外部参考时钟   |

| 比特 | 名称      | 属性  | 复位值 | 描述   |
|----|---------|-----|-----|--|
| 1  | TICKINT | R/W | 0   | SysTick中断使能<br>1: 使能中断<br>0: 禁止中断            |
| 0  | ENABLE  | R/W | 0   | SysTick定时器使能<br>1: 使能SysTick<br>0: 禁止SysTick |

### 19.2.2 重载值寄存器 SYS\_RVR (偏移: 04h)

| 比特    | 名称     | 属性  | 复位值        | 描述             |
|-------|--------|-----|------------|----------------|
| 31:24 | RSV    | -   | -          | 保留             |
| 23:0  | RELOAD | R/W | 0xFFFFFFFF | SysTick 定时器重载值 |

### 19.2.3 当前值寄存器 SYS\_CVR (SysTick->VAL) (偏移: 08h)

| 比特    | 名称      | 属性  | 复位值        | 描述   |
|-------|---------|-----|------------|--|
| 31:24 | RSV     | -   | -          | 保留   |
| 23:0  | CURRENT | R/W | 0xFFFFFFFF | 读该寄存器, 获取 SysTick 定时器的当前计数值; 写任意值到该寄存器, 清零该寄存器及 COUNTFLAG。 |

## 19.3 使用流程

由于 SysTick 定时器的重载值和当前值在复位时都是未定义的, 为了防止产生异常结果, 对 SysTick 的配置需要遵循一定的流程:

1. 配置 SysTick->CTRL. ENABLE 为 0, 禁止 SysTick。
2. 配置 SysTick->CTRL. CLKSOURCE, 选择 SysTick 的时钟源。
3. 配置 SysTick->LOAD, 选择 SysTick 的溢出周期。
4. 向 SysTick->VAL 写入任意值, 清零 SysTick->VAL 及 SysTick->CTRL.
5. COUNTFLAG。
6. 配置 SysTick->CTRL. TICKINT 为 1, 使能 SysTick 中断。
7. 配置 SysTick->CTRL. ENABLE 为 1, 使能 SysTick。
8. 查询等待定时器溢出标志到来之后关闭和清空计数器, 或者在中断服务程序中读取 SysTick->CTRL 以清除溢出标志。

## 20 版本维护

| 版本   | 日期         | 描述  |
|------|------------|---|
| V1.0 | 2019.10.16 | 初始版   |
| V1.1 | 2020.7.13  | 更新章节 4.1 地址映射描述, FLASH 分 16K/32K/64;<br>修改章节 4.4.2 外部复位管脚名称;<br>修改章节 4.6 增加 REAMP、VECTOR_OFFSET 寄存器描述;<br>修改章节 4.2 时钟框图;<br>统一时钟描述 (RTC_CLK 改为 clk32k, RTC_1HZ 改为 CLK_1HZ);<br>更新章节 4.6.9 PA_SEL 寄存器的 PA0 描述;<br>修改章节 13.2 LPTimer 工作时钟描述;                  |
| V1.2 | 2020.9.29  | 修改章节 10.3 SPI0 寄存器读写属性和默认值<br>修改章节 11.3 SPI1 寄存器读写属性和默认值<br>修改章节 20.2 SysTick 寄存器默认值  |
| V1.3 | 2023.12.28 | 更新外围模块时钟寄存器的 bit 15 位为 I2C 模块时钟使能;<br>更新管脚描述 PD1 的 TIM3_TOG 是 "GTimer3 输出 (PWM 输出)";<br>更新 PWM 中断状态寄存器 CPIF 的属性为 "RW1C";<br>更新 I2C_XSADM 比特描述 8:1 改为 10:1;<br>更新 PAD_PU 寄存器的默认值, bit23 和 bit25 默认值改为 1;<br>更新外部复位滤波控制寄存器为 EXT_RESET_CTRL;<br>删除引脚描述及电气参数章节。 |
|      |            |   |
|      |            |   |
|      |            |   |
|      |            |   |