

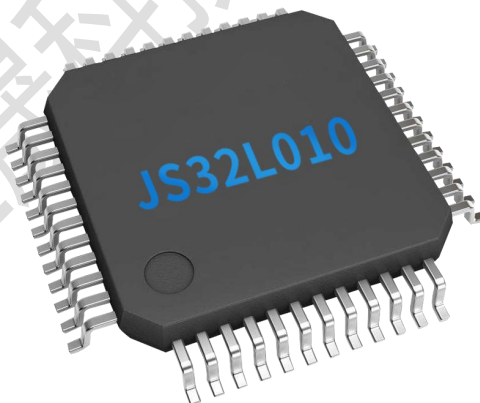


集成 LCD 驱动+升压的 M0 核微控制器

JS32L010 低功耗系列

用户手册

V1.9



珠海巨晟科技股份有限公司

地 址：广东省珠海市高新区金唐路 1 号港湾 1 号湾 8 栋 4 楼

电 话：0756-3335384 传 真：0756-3335384

网 站：www.honor-ic.com 邮 编：519080



版本历史

变更类型：A - 增加 M - 修订 D - 删除

| 变更版本号 | 日期 | 变更类型 | 修改人 | 审核 | 摘要 |
|-------|------------|------|-----|-----|---|
| V1.0 | 2020-3-23 | | | | 正式版本 |
| V1.1 | 2020-5-11 | M | RLD | MCW | 更新时序图 |
| V1.2 | 2020-7-28 | A | RLD | MCW | Add ssop24 |
| V1.3 | 2020-8-21 | M | RLD | MCW | 增加 IO 类型说明 |
| V1.4 | 2020-9-10 | M | RLD | MCW | 修改 ADKEY_CFG :ADCPRE |
| V1.5 | 2020-9-15 | M | RLD | YH | |
| V1.6 | 2020-10-15 | M | RLD | YH | 修改 GPIOx_OSPEEDL/ GPIOx_OSPEEDH |
| V1.7 | 2020-10-23 | M | RLD | | 1.更新 SPI 时序图 2.修改产品主要特性名称：集成 LCD 驱动+升压的 M0 核微控制器 3.修正文档错别字和语句 |
| V1.8 | 2021-9-9 | M | QZH | | 1. 修改 swd 引脚映射 2. 修正文档部分语句 3. 增加 UART_DMA_STA 寄存器 |
| V1.9 | 2021-10-9 | M | QZH | | 1.修正 PC9 定义 |

版权声明

本资料是为了让用户根据用途选择合适的产品而提供的参考资料，不转让属于珠海巨晟科技股份有限公司或者第三方所有的知识产权以及其他权利的许可。在使用本资料所记载的信息并对有关产品是否适用做出最终判断前，请您务必将所有信息作为一个整体系统来评价。对于本资料所记载的信息使用不当而引起的损害、责任问题或者其他损失，珠海巨晟科技股份有限公司将不承担责任。未经珠海巨晟科技股份有限公司的许可，不得翻印或者复制全部或部分本资料的内容。

今后日常产品的更新会在适当的时候发布，恕不另行通知。在购买本资料所记载的产品时，请预先向珠海巨晟科技股份有限公司确认最新信息，并请您通过各种方式关注珠海巨晟

科技股份有限公司公布的信息。

如果您需要了解有关本资料所记载的信息或产品的详情，请与珠海巨晟科技股份有限公司的技术服务部门联系，我们会为您提供全方位的技术支持。

珠海巨晟科技股份有限公司

目录

| | |
|----------------------------------|-----------|
| 1 产品特性 | 1 |
| 2 产品简介 | 2 |
| 2.1 典型应用 | 2 |
| 2.2 选型表 | 2 |
| 2.3 命名规则 | 2 |
| 3 终端配置与功能 | 3 |
| 3.1 管脚图 | 3 |
| 3.2 管脚定义 | 5 |
| 4 系统及存储器架构 | 10 |
| 4.1 系统架构 | 10 |
| 4.2 存储器映射 | 11 |
| 5 嵌入式闪存 (FLASH) | 14 |
| 5.1 闪存主要特性 | 14 |
| 5.2 闪存功能描述 | 14 |
| 6 中断和事件(INT/EVT) | 24 |
| 6.1 嵌套向量中断控制器 | 24 |
| 6.2 系统滴答(SysTick)校准值寄存器 | 24 |
| 6.3 中断功能描述 | 24 |
| 6.4 外部中断/事件控制器(EXTI) | 25 |
| 7 循环冗余校验计算单元 (CRC) | 26 |
| 7.1 简介 | 26 |
| 7.2 模块框图 | 26 |
| 7.3 主要特性 | 26 |
| 7.4 寄存器定义 | 26 |
| 7.5 操作流程 | 27 |
| 8 电源管理 (POWER MANAGEMENT) | 29 |
| 8.1 电源 | 29 |
| 8.2 电源管理器 | 29 |
| 8.3 电源控制寄存器 | 30 |
| 9 低功耗(LOW POWER) | 32 |
| 9.1 低功耗模式 | 32 |
| 9.2 进入低功耗 | 32 |
| 9.3 低功耗唤醒 | 32 |
| 10 复位和时钟控制 (RESET/CLOCK) | 34 |
| 10.1 引脚复位(MCLR)功能 | 34 |
| 10.2 复位 | 34 |
| 10.3 时钟 | 35 |
| 11 通用输入输出 (GPIO) | 37 |
| 11.1 简介 | 37 |
| 11.2 模块框图 | 37 |
| 11.3 GPIO 主要特征 | 37 |
| 11.4 GPIO 功能描述 | 37 |
| 11.5 GPIO 寄存器描述 | 40 |
| 12 通信接口外设 (CSI) | 47 |
| 12.1 串行外设接口和内部集成电路(SPI_IIC) | 47 |
| 12.2 SPI/IIC 寄存器描述 | 50 |
| 13 通用异步收发器 (UART) | 57 |
| 13.1 简介 | 57 |
| 13.2 模块框图 | 57 |
| 13.3 UART 特性 | 57 |
| 13.4 IO 映射 | 57 |
| 13.5 寄存器描述 | 58 |

| | |
|--------------------------------|------------|
| 14 比较器 (COMP) | 64 |
| 14.1 简介 | 64 |
| 14.2 主要特性 | 64 |
| 14.3 寄存器描述 | 65 |
| 14.4 操作流程 | 65 |
| 15 数模转换器 (DAC) | 66 |
| 15.1 简介 | 66 |
| 15.2 模块框图 | 66 |
| 15.3 主要特征 | 66 |
| 15.4 寄存器描述 | 66 |
| 16 模数转换器 (ADC) | 67 |
| 16.1 功能简介 | 67 |
| 16.2 模块框图 | 67 |
| 16.3 主要特征 | 67 |
| 16.4 功能描述 | 67 |
| 16.5 通道选择 | 68 |
| 16.6 寄存器描述 | 69 |
| 16.7 操作流程 | 73 |
| 17 定时器 (TMR0/1/2/3/4/5) | 75 |
| 17.1 16 位定时器 0/1/2/3/5 | 75 |
| 18 看门狗 (WDT) | 87 |
| 18.1 简介 | 87 |
| 18.2 模块框图 | 87 |
| 18.3 寄存器描述 | 87 |
| 19 系统寄存器 (SYSTEM REG) | 89 |
| 19.1 系统寄存器基地址表 | 89 |
| 19.2 寄存器描述 | 89 |
| 20 液晶控制器 (LCD) | 104 |
| 20.1 简介 | 104 |
| 20.2 功能框图 | 104 |
| 20.3 主要特性 | 104 |
| 20.4 原理介绍 | 104 |
| 20.5 寄存器描述 | 109 |
| 21 实时时钟 (RTC) | 118 |
| 21.1 简介 | 118 |
| 21.2 功能框图 | 118 |
| 21.3 寄存器描述 | 119 |
| 21.4 使用说明 | 123 |
| 22 频率锁相环 (PLL) | 124 |
| 22.1 功能简介 | 124 |
| 22.2 寄存器描述 | 124 |
| 23 器件电子签名 (E-SIGNATURE) | 126 |
| 23.1 存储器容量寄存器 | 126 |
| 24 调试支持 (DBG) | 127 |
| 24.1 引脚分布和调试端口脚 | 127 |
| 24.2 SWD 调试端口脚 | 127 |
| 24.3 SW 调试端口 | 128 |
| 24.4 MCU 调试 | 129 |
| 25 电气特性 | 130 |
| 25.1 绝对最大额定值 | 130 |
| 25.2 工作条件/特性参数 | 130 |
| 26 封装信息 | 141 |
| 26.1 LQFP48 | 141 |

| | |
|------------------|-----|
| 26.2 LQFP64..... | 141 |
| 26.3 SSOP24..... | 142 |

珠海巨晟科技股份有限公司

1 产品特性

- **内核与系统**
 - 32 位 ARM® Cortex™-M0 处理器内核
 - 最高工作频率可达 48MHz
 - 32 个指令周期 32 位硬件乘法器
 - 32 个中断源, 可配置 4 层中断优先级
 - 支持 SWD 调试接口
- **存储器**
 - 32K 字节的闪存程序存储器
 - 2K 字节的 SRAM
 - Boot loader 支持片内 Flash、支持单线 UART、在线用户编程(IAP)/系统编程(ISP)
- **时钟、复位和电源管理**
 - 1.8V ~ 3.6V 供电
 - 片上电容 LDO 为芯片内系统供电, LDO 带过流保护
 - 上电/断电复位(POR/PDR)、16 档可编程电压监测器(PVD)
 - 外部 32.768KHz 晶体振荡器
 - 内嵌 26MHz($\pm 1.5\%$) 高速振荡器
 - 内嵌低速振荡器 32.768KHz($\pm 1.5\%$)
 - PLL 支持 CPU 最高运行在 48MHz
 - 内嵌 128KHz 启动低速振荡器
 - 内置时钟安全系统(CSS)
- **DMA 支持**
 - 支持的外设: EFLASH, UART, SPI, CRC, ADC, TIMER1
- **多达 57(LQFP64) /41(LQFP48) 个快速 I/O 端口**
 - 所有 I/O 口可以触发边沿或电平响应中断, 唤醒低功耗模式
 - 支持一个输出低电压大电流 IO(漏电流 500mA)
- **通讯接口外设**
 - 1 路 SPI, master 下最高支持 26Mbit/s, 支持 1/2/4 线主从模式, 支持 I2C 模式
 - 1 路 UART, 支持程序升级和多机通讯
- **LCD 显示控制器**
 - 最大支持 COM+SEG 数量是 42 个, COM 个数不大于 8 个, SEG 不大于 38 个
 - 支持静态扫描和动态扫描
 - 动态扫描 duty: 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8
 - 动态扫描偏置电压: 1/2, 1/3
 - 允许软件随时更新显存数据, 不影响显示效果
 - 支持内部升压电源给 LCD 供电, 内部电阻分压和外部电源直接给 LCD 供电
 - 支持对比度调节和低功耗模式显示
- 支持闪烁功能
- 扫描频率为 30Hz~120Hz 可调
- **实时时钟 RTC**
 - 数字时钟补偿
 - 硬件万年历
 - 支持 12 小时、24 小时显示模式
 - 两个可编程闹钟
 - 低功耗工作
- **8 个定时器**
 - 5 个 16 位定时器, 1 个 32 位定时器, 每个定时器支持 1 个 IC/OC, 可组合用于 IR 控制编码, 支持 PWM 输出
 - 1 个看门狗定时器
 - 1 个系统时间定时器: 24 位自减型计数器
- **高安全性**
 - 支持硬件加解密程序, 防止程序被盗
 - 支持 5/7/8/16/32 位 CRC 校验, 保证数据准确性
- **低功耗模式**
 - 支持 IDLE, STOP, SLEEP 模式
 - 8.52mA@IDLE: 3.3V, PLL 48MHz, LCD 显示, 等待按键唤醒, 不关时钟, 外设运行, CPU 停止
 - 2.89mA@STOP: 3.3V, LCD 显示, 等待按键唤醒, 数字模块时钟关闭, CPU 停止
 - 7.8uA@SLEEP: 3.3V, LCD 显示, 等待按键唤醒, 除 32KHz 其它时钟关闭, CPU 停止
 - 2.2uA@SLEEP: 3.3V, 没有其它外设, 等待按键唤醒, 所有时钟关闭, CPU 停止
 - 唤醒时间最快 10us
 - GPIO 口可通过边沿或电平触发中断唤醒
- **1 个 12 位高速模数转换器**
 - 支持最高 250K 采样率
 - 26 个通道@LQFP64, 13 个通道@LQFP48
 - 转换范围: 0 ~ V_{VREFP}
- **1 个比较器**
- **1 个 8 位精度 DAC, 1 个 12 位精度 DAC**
- **内置温度传感器**
- **高可靠性**
 - ESD HBM 4KV
 - EFT $\pm 4KV$
 - Latch-up $\pm 200mA$ @105°C
- **96 位的芯片唯一 ID(UID)**
- **封装**
 - LQFP48/LQFP64/SSOP24
- **工业级温度范围**
 - -40°C ~ 105°C

2 产品简介

JS32L010 系列是超低功耗，最高 48MHz 主频的 32 位 ARM Cortex M0 微控制器。它集成了 12 位 250K 转换率高精度 ADC，DAC，比较器，CRC，支持 PWM 的定时器，LCD 显示，UART，SPI，IIC 等通讯外设，还支持硬件加解密程序，保护代码安全。具有在线仿真，高集成度，高抗干扰性，高可靠性和超低功耗的特点。

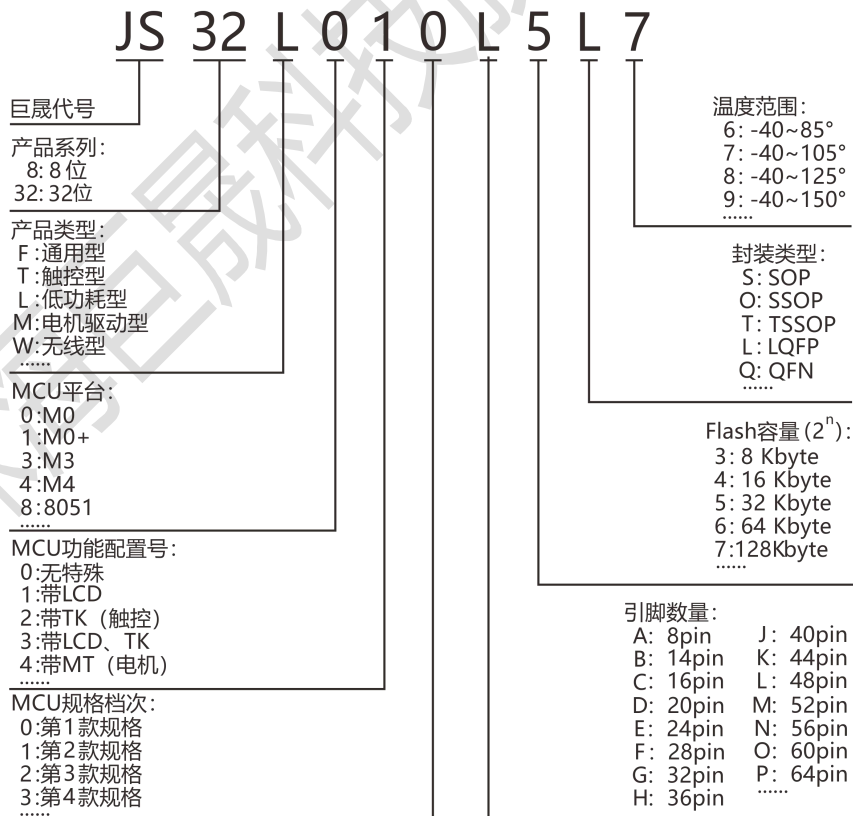
2.1 典型应用

- 额/耳温枪、血糖/血压/血氧/心电监测仪等医疗电子产品
- 儿童玩具、电子烟、酒精测试仪、按摩仪等消费电子产品
- 遥控器、火警探头，温控器、智能门锁等智能传感器产品
- 各种对于电池供电和对于功耗苛求的便携式设备等

2.2 选型表

| 产品型号 | 封装 | Flash | RAM | GPIO | LCD (COM*SEG) | Timer | PWM | RTC | UART | SPI/I ² C | ADC 12bit | DAC | 比较器 | 温度传感器 | 大电流 IO |
|--------------|--------|-------|-----|------|-------------------------|---------|-----|-----|------|----------------------|--------------|---------|-----|-------|--------|
| JS32L010L5L7 | LQFP48 | 32K | 2K | 41 | 8*34/6*36/ 5*37/4*38 | 5*16bit | | | | | 13ch | 1*8bit | | | |
| JS32L010P5L7 | LQFP64 | 32K | 2K | 57 | | 1*32bit | 6 | 1 | 1 | 1 | 26ch | 1*12bit | 1 | ±2°C | 1 |
| JS32L010E5O7 | SSOP24 | 32K | 2K | 22 | 8*12/6*14.... | | | | | | 6ch | | | | |

2.3 命名规则



3 终端配置与功能

3.1 管脚图

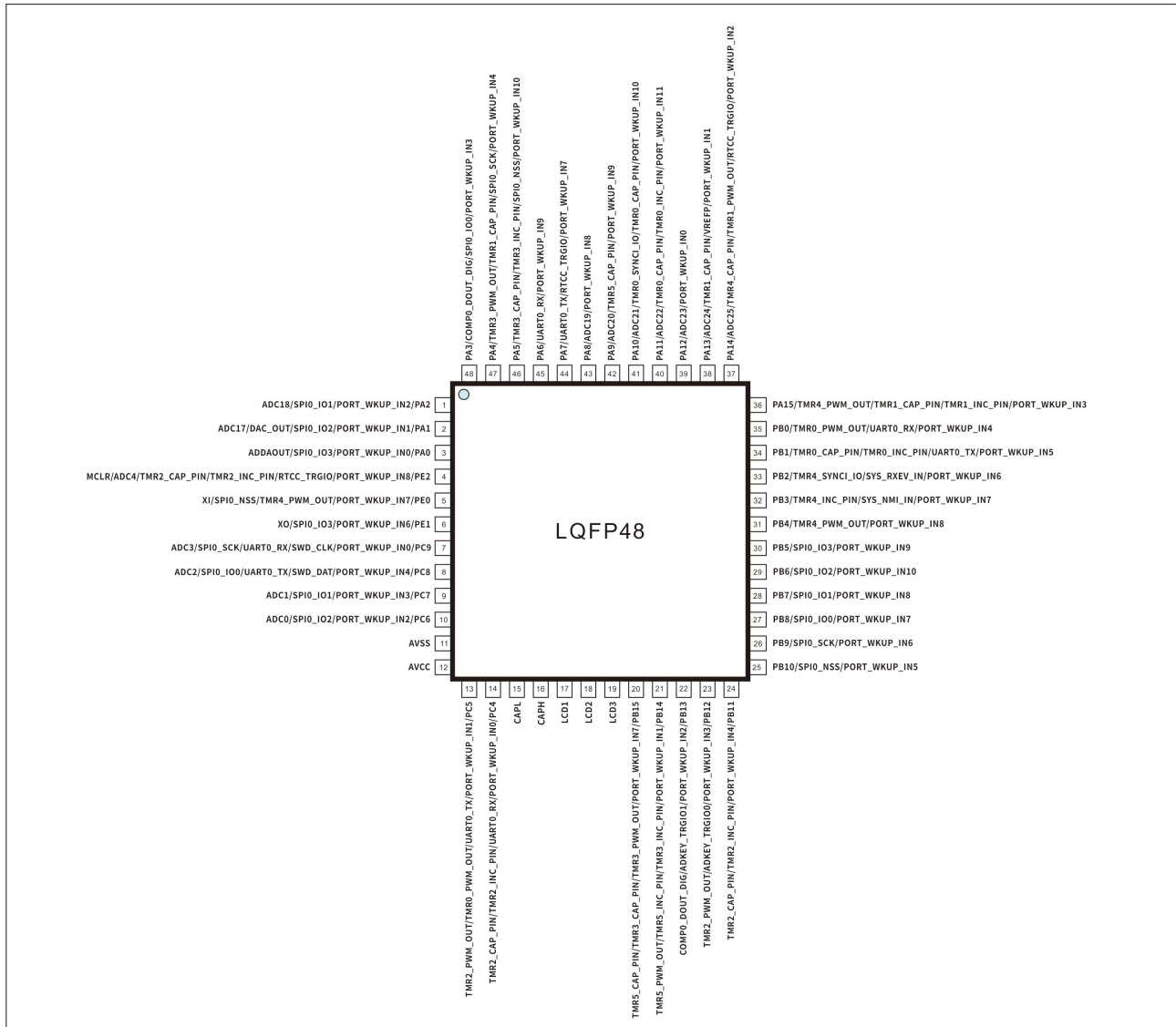


图 3-1 JS32L010 LQFP48 Package

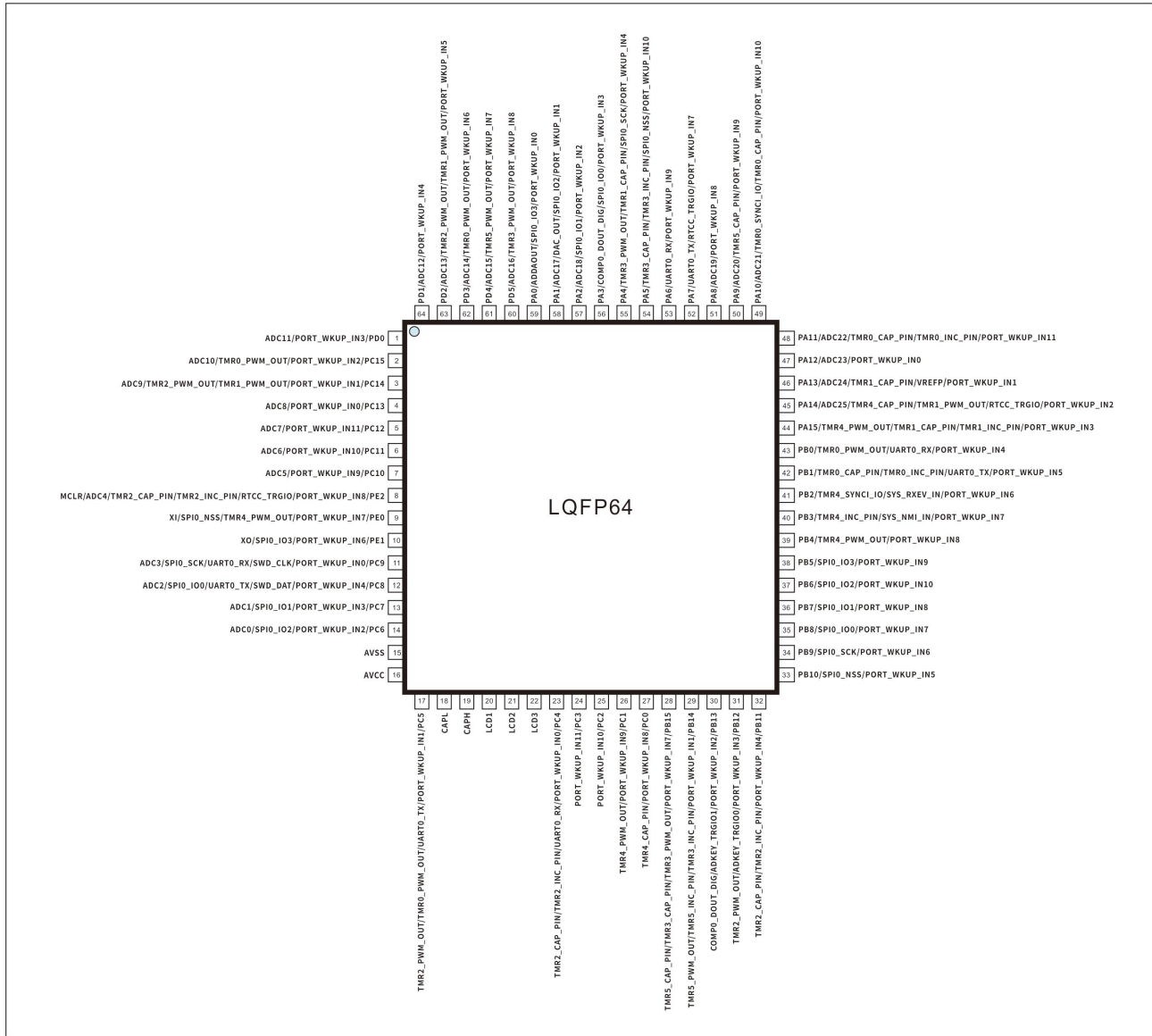


图 3-2 JS32L010 LQFP64 Package

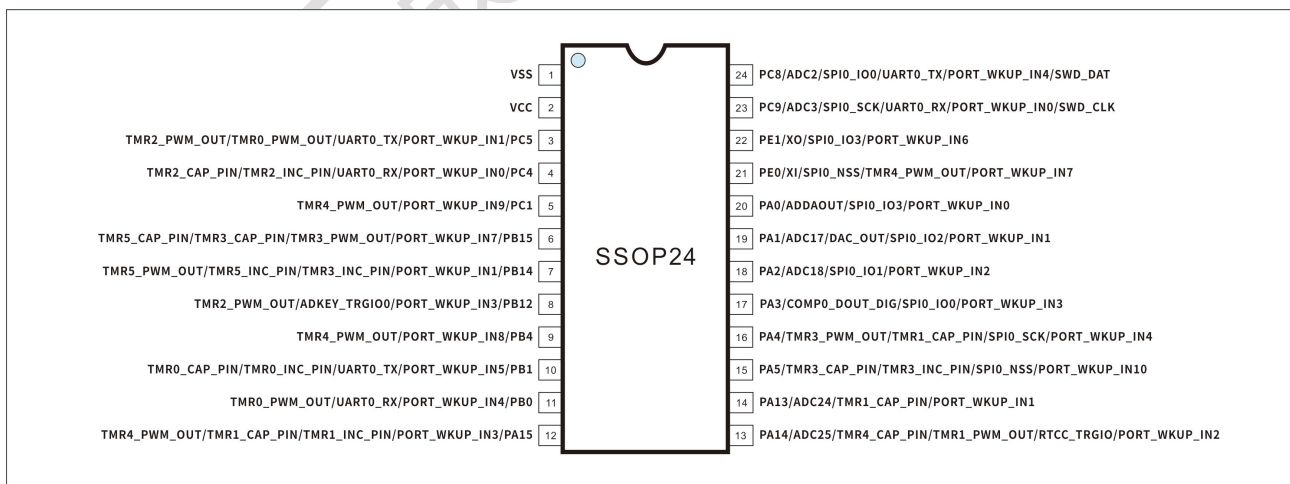


图 3-3 JS32L010 SSOP24 Package

3.2 管脚定义

表 3-1 GPIO 管脚定义

| LQFP48 | LQFP64 | SSOP24 | 引脚 | 引脚类型 | I/O 电路类型 | 复用功能 | 额外功能 |
|--------|--------|--------|------|------|-----------|---|------------------|
| 3 | 59 | 20 | PA0 | I/O | LCDIO[0] | SPI0_IO3 PORT_WKUP_IN0 | ADDAOUT |
| 2 | 58 | 19 | PA1 | I/O | LCDIO[1] | SPI0_IO2 PORT_WKUP_IN1 | ADC17 DAC_OUT |
| 1 | 57 | 18 | PA2 | I/O | LCDIO[2] | SPI0_IO1 PORT_WKUP_IN2 | ADC18 |
| 48 | 56 | 17 | PA3 | I/O | LCDIO[3] | COMP0_DOUT_DIG SPI0_IO0 PORT_WKUP_IN3 | |
| 47 | 55 | 16 | PA4 | I/O | LCDIO[4] | TMR3_PWM_OUT TMR1_CAP_PIN SPI0_SCK PORT_WKUP_IN4 | - |
| 46 | 54 | 15 | PA5 | I/O | LCDIO[5] | TMR3_CAP_PIN TMR3_INC_PIN SPI0_NSS PORT_WKUP_IN10 | - |
| 45 | 53 | - | PA6 | I/O | LCDIO[6] | UART0_RX PORT_WKUP_IN9 | - |
| 44 | 52 | - | PA7 | I/O | LCDIO[7] | UART0_TX RTCC_TRGIO PORT_WKUP_IN7 | - |
| 43 | 51 | - | PA8 | I/O | LCDIO[8] | PORT_WKUP_IN8 | ADC19 |
| 42 | 50 | - | PA9 | I/O | LCDIO[9] | TMR5_CAP_PIN PORT_WKUP_IN9 | ADC20 |
| 41 | 49 | - | PA10 | I/O | LCDIO[10] | TMR0_SYNCI_IO TMR0_CAP_PIN PORT_WKUP_IN10 | ADC21 |
| 40 | 48 | - | PA11 | I/O | LCDIO[11] | TMR0_CAP_PIN TMR0_INC_PIN PORT_WKUP_IN11 | ADC22 |
| 39 | 47 | - | PA12 | I/O | LCDIO[12] | PORT_WKUP_IN0 | ADC23 |
| 38 | 46 | 14 | PA13 | I/O | LCDIO[13] | TMR1_CAP_PIN PORT_WKUP_IN1 | ADC24 VREFP |
| 37 | 45 | 13 | PA14 | I/O | LCDIO[14] | TMR4_CAP_PIN TMR1_PWM_OUT RTCC_TRGIO PORT_WKUP_IN2 | ADC25 |
| 36 | 44 | 12 | PA15 | I/O | LCDIO[15] | TMR4_PWM_OUT TMR1_CAP_PIN TMR1_INC_PIN | - |

| | | | | | | | |
|----|----|----|------|-----|-----------|---|---|
| | | | | | | PORT_WKUP_IN3 | |
| 35 | 43 | 11 | PB0 | I/O | LCDIO[16] | TMR0_PWM_OUT UART0_RX PORT_WKUP_IN4 | - |
| 34 | 42 | 10 | PB1 | I/O | LCDIO[17] | TMR0_CAP_PIN TMR0_INC_PIN UART0_TX PORT_WKUP_IN5 | - |
| 33 | 41 | - | PB2 | I/O | LCDIO[18] | TMR4_SYNCIO SYS_RXEV_IN PORT_WKUP_IN6 | - |
| 32 | 40 | - | PB3 | I/O | LCDIO[19] | TMR4_INC_PIN SYS_NMI_IN PORT_WKUP_IN7 | - |
| 31 | 39 | 9 | PB4 | I/O | LCDIO[20] | TMR4_PWM_OUT PORT_WKUP_IN8 | - |
| 30 | 38 | - | PB5 | I/O | LCDIO[21] | SPI0_IO3 PORT_WKUP_IN9 | - |
| 29 | 37 | - | PB6 | I/O | LCDIO[22] | SPI0_IO2 PORT_WKUP_IN10 | - |
| 28 | 36 | - | PB7 | I/O | LCDIO[23] | SPI0_IO1 PORT_WKUP_IN8 | - |
| 27 | 35 | - | PB8 | I/O | LCDIO[24] | SPI0_IO0 PORT_WKUP_IN7 | - |
| 26 | 34 | - | PB9 | I/O | LCDIO[25] | SPI0_SCK PORT_WKUP_IN6 | - |
| 25 | 33 | - | PB10 | I/O | LCDIO[26] | SPI0_NSS PORT_WKUP_IN5 | - |
| 24 | 32 | - | PB11 | I/O | LCDIO[27] | TMR2_CAP_PIN TMR2_INC_PIN PORT_WKUP_IN4 | - |
| 23 | 31 | 8 | PB12 | I/O | LCDIO[28] | TMR2_PWM_OUT ADKEY_TRGIO0 PORT_WKUP_IN3 | - |
| 22 | 30 | - | PB13 | I/O | LCDIO[29] | COMP0_DOUT_DIG ADKEY_TRGIO1 PORT_WKUP_IN2 | - |
| 21 | 29 | 7 | PB14 | I/O | LCDIO[30] | TMR5_PWM_OUT TMR5_INC_PIN TMR3_INC_PIN PORT_WKUP_IN1 | - |
| 20 | 28 | 6 | PB15 | I/O | LCDIO[31] | TMR5_CAP_PIN TMR3_CAP_PIN TMR3_PWM_OUT PORT_WKUP_IN7 | - |
| - | 27 | - | PC0 | I/O | LCDIO[32] | TMR4_CAP_PIN PORT_WKUP_IN8 | - |
| - | 26 | 5 | PC1 | I/O | LCDIO[33] | TMR4_PWM_OUT | - |

| | | | | | | | |
|----|----|----|------|-----|-------------------------|---|-----------------|
| | | | | | | PORT_WKUP_IN9 | |
| - | 25 | - | PC2 | I/O | LCDIO[34] | PORT_WKUP_IN10 | - |
| - | 24 | - | PC3 | I/O | LCDIO[35] | PORT_WKUP_IN11 | - |
| 19 | 22 | - | LCD3 | S | - | - | - |
| 18 | 21 | - | LCD2 | S | - | - | - |
| 17 | 20 | - | LCD1 | S | - | - | - |
| 16 | 19 | - | CAPH | S | - | - | - |
| 15 | 18 | - | CAPL | S | - | - | - |
| 14 | 23 | 4 | PC4 | I/O | LCDIO[36] | TMR2_CAP_PIN TMR2_INC_PIN UART0_RX PORT_WKUP_IN0 | - |
| 13 | 17 | 3 | PC5 | I/O | CURRENTIO/ LCDIO[37] | TMR2_PWM_OUT TMR0_PWM_OUT UART0_TX PORT_WKUP_IN1 | - |
| 12 | 16 | 2 | AVCC | S | - | - | - |
| 11 | 15 | 1 | AVSS | S | - | - | - |
| 10 | 14 | - | PC6 | I/O | LCDIO[38] | SPI0_IO2 PORT_WKUP_IN2 | ADC0 |
| 9 | 13 | - | PC7 | I/O | LCDIO[39] | SPI0_IO1 PORT_WKUP_IN3 | ADC1 |
| 8 | 12 | 24 | PC8 | I/O | LCDIO[40] | SPI0_IO0 UART0_TX PORT_WKUP_IN4 | ADC2 SWD_DAT |
| 7 | 11 | 23 | PC9 | I/O | LCDIO[41] | SPI0_SCK UART0_RX PORT_WKUP_IN0 | ADC3 SWD_CLK |
| 6 | 10 | 22 | PE1 | I/O | IOG1 | SPI0_IO3 PORT_WKUP_IN6 | XO |
| 5 | 9 | 21 | PE0 | I/O | IOG1 | SPI0_NSS TMR4_PWM_OUT PORT_WKUP_IN7 | XI |
| 4 | 8 | - | PE2 | I/O | IOG1 | TMR2_CAP_PIN TMR2_INC_PIN RTCC_TRGIO PORT_WKUP_IN8 | MCLR ADC4 |
| - | 7 | - | PC10 | I/O | IOG1 | PORT_WKUP_IN9 | ADC5 |
| - | 6 | - | PC11 | I/O | IOG1 | PORT_WKUP_IN10 | ADC6 |
| - | 5 | - | PC12 | I/O | IOG1 | PORT_WKUP_IN11 | ADC7 |
| - | 4 | - | PC13 | I/O | IOG1 | PORT_WKUP_IN0 | ADC8 |
| - | 3 | - | PC14 | I/O | IOG1 | TMR2_PWM_OUT TMR1_PWM_OUT PORT_WKUP_IN1 | ADC9 |
| - | 2 | - | PC15 | I/O | IOG1 | TMR0_PWM_OUT PORT_WKUP_IN2 | ADC10 |
| - | 1 | - | PD0 | I/O | IOG1 | PORT_WKUP_IN3 | ADC11 |
| - | 64 | - | PD1 | I/O | IOG1 | PORT_WKUP_IN4 | ADC12 |

| | | | | | | | |
|---|----|---|-----|-----|------|---|-------|
| - | 63 | - | PD2 | I/O | IOG1 | TMR2_PWM_OUT TMR1_PWM_OUT PORT_WKUP_IN5 | ADC13 |
| - | 62 | - | PD3 | I/O | IOG1 | TMR0_PWM_OUT PORT_WKUP_IN6 | ADC14 |
| - | 61 | - | PD4 | I/O | IOG1 | TMR5_PWM_OUT PORT_WKUP_IN7 | ADC15 |
| - | 60 | - | PD5 | I/O | IOG1 | TMR3_PWM_OUT PORT_WKUP_IN8 | ADC16 |

注: IOG1 和 LCDIO 属于 A/B 类 IO, CURRENTIO 属于 C 类 IO.

表 3-2 GPIO 复用功能 AF0~AF3

| Port | | AF0 | AF1 | AF2 | AF3 |
|------|------|----------------|--------------|--------------|----------------|
| PA | PA0 | - | - | SPI0_IO3 | PORT_WKUP_IN0 |
| | PA1 | - | - | SPI0_IO2 | PORT_WKUP_IN1 |
| | PA2 | - | - | SPI0_IO1 | PORT_WKUP_IN2 |
| | PA3 | COMP0_DOUT_DIG | - | SPI0_IO0 | PORT_WKUP_IN3 |
| | PA4 | TMR3_PWM_OUT | TMR1_CAP_PIN | SPI0_SCK | PORT_WKUP_IN4 |
| | PA5 | TMR3_CAP_PIN | TMR3_INC_PIN | SPI0_NSS | PORT_WKUP_IN10 |
| | PA6 | - | UART0_RX | - | PORT_WKUP_IN9 |
| | PA7 | - | UART0_TX | RTCC_TRGIO | PORT_WKUP_IN7 |
| | PA8 | - | - | - | PORT_WKUP_IN8 |
| | PA9 | - | - | TMR5_CAP_PIN | PORT_WKUP_IN9 |
| | PA10 | TMR0_SYNCIO | - | TMR0_CAP_PIN | PORT_WKUP_IN10 |
| | PA11 | TMR0_CAP_PIN | TMR0_INC_PIN | - | PORT_WKUP_IN11 |
| | PA12 | - | - | - | PORT_WKUP_IN0 |
| | PA13 | - | TMR1_CAP_PIN | - | PORT_WKUP_IN1 |
| | PA14 | TMR4_CAP_PIN | TMR1_PWM_OUT | RTCC_TRGIO | PORT_WKUP_IN2 |
| | PA15 | TMR4_PWM_OUT | TMR1_CAP_PIN | TMR1_INC_PIN | PORT_WKUP_IN3 |
| PB | PB0 | TMR0_PWM_OUT | - | UART0_RX | PORT_WKUP_IN4 |
| | PB1 | TMR0_CAP_PIN | TMR0_INC_PIN | UART0_TX | PORT_WKUP_IN5 |
| | PB2 | TMR4_SYNCIO | SYS_RXEN_IN | - | PORT_WKUP_IN6 |
| | PB3 | TMR4_INC_PIN | SYS_NMI_IN | - | PORT_WKUP_IN7 |
| | PB4 | TMR4_PWM_OUT | - | - | PORT_WKUP_IN8 |
| | PB5 | - | SPI0_IO3 | - | PORT_WKUP_IN9 |
| | PB6 | - | SPI0_IO2 | - | PORT_WKUP_IN10 |
| | PB7 | - | SPI0_IO1 | - | PORT_WKUP_IN8 |
| | PB8 | - | SPI0_IO0 | - | PORT_WKUP_IN7 |
| | PB9 | - | SPI0_SCK | - | PORT_WKUP_IN6 |
| | PB10 | - | SPI0_NSS | - | PORT_WKUP_IN5 |
| | PB11 | TMR2_CAP_PIN | TMR2_INC_PIN | - | PORT_WKUP_IN4 |
| | PB12 | TMR2_PWM_OUT | ADKEY_TRGIO0 | - | PORT_WKUP_IN3 |
| | PB13 | COMP0_DOUT_DIG | ADKEY_TRGIO1 | - | PORT_WKUP_IN2 |
| | PB14 | TMR5_PWM_OUT | TMR5_INC_PIN | TMR3_INC_PIN | PORT_WKUP_IN1 |

| | | | | | |
|----|------|--------------|--------------|--------------|----------------|
| | PB15 | TMR5_CAP_PIN | TMR3_CAP_PIN | TMR3_PWM_OUT | PORT_WKUP_IN7 |
| PC | PC0 | TMR4_CAP_PIN | - | - | PORT_WKUP_IN8 |
| | PC1 | TMR4_PWM_OUT | - | - | PORT_WKUP_IN9 |
| | PC2 | - | - | - | PORT_WKUP_IN10 |
| | PC3 | - | - | - | PORT_WKUP_IN11 |
| | PC4 | TMR2_CAP_PIN | TMR2_INC_PIN | UART0_RX | PORT_WKUP_IN0 |
| | PC5 | TMR2_PWM_OUT | TMR0_PWM_OUT | UART0_TX | PORT_WKUP_IN1 |
| | PC6 | SPI0_IO2 | - | - | PORT_WKUP_IN2 |
| | PC7 | SPI0_IO1 | - | - | PORT_WKUP_IN3 |
| | PC8 | SPI0_IO0 | - | UART0_TX | PORT_WKUP_IN4 |
| | PC9 | SPI0_SCK | - | UART0_RX | PORT_WKUP_IN0 |
| | PC10 | - | - | - | PORT_WKUP_IN9 |
| | PC11 | - | - | - | PORT_WKUP_IN10 |
| | PC12 | - | - | - | PORT_WKUP_IN11 |
| | PC13 | - | - | - | PORT_WKUP_IN0 |
| | PC14 | TMR2_PWM_OUT | TMR1_PWM_OUT | - | PORT_WKUP_IN1 |
| | PC15 | - | TMR0_PWM_OUT | - | PORT_WKUP_IN2 |
| PD | PD0 | - | - | - | PORT_WKUP_IN3 |
| | PD1 | - | - | - | PORT_WKUP_IN4 |
| | PD2 | TMR2_PWM_OUT | TMR1_PWM_OUT | - | PORT_WKUP_IN5 |
| | PD3 | - | TMR0_PWM_OUT | - | PORT_WKUP_IN6 |
| | PD4 | - | TMR5_PWM_OUT | - | PORT_WKUP_IN7 |
| | PD5 | - | TMR3_PWM_OUT | - | PORT_WKUP_IN8 |
| PE | PE0 | SPI0_NSS | TMR4_PWM_OUT | - | PORT_WKUP_IN7 |
| | PE1 | SPI0_IO3 | - | - | PORT_WKUP_IN6 |
| | PE2 | TMR2_CAP_PIN | TMR2_INC_PIN | RTCC_TRGIO | PORT_WKUP_IN8 |

注：UART0_RX 单线通信时作收发引脚，双线工作时作接收引脚。

4 系统及存储器架构

JS32L010 芯片采用了哈佛结构，具有低中断延迟时间和高集成度特性，而且非常适合于那些需要高性能和低功耗微控制器的市场领域。预先定义的存储器映射和高达 4 GB 的存储空间，充分保证了系统的灵活性和可扩展性。

4.1 系统架构

JS32L010 芯片采用 32 位多层总线结构，该结构可使系统中的多个主机和从机之间的并行通信成为可能。多层总线结构包括一个 AHB 互联矩阵、两个 AHB 总线和两个 APB 总线。AHB 互联矩阵的互联关系将在下面进行说明。

JS32L010 主系统由以下两部分构成：

- 2 个驱动单元：
 - CPU 内核系统总线(S-bus)
 - DMA 总线
- 2 个存储单元
 - 内部闪存存储器
 - 内部 SRAM

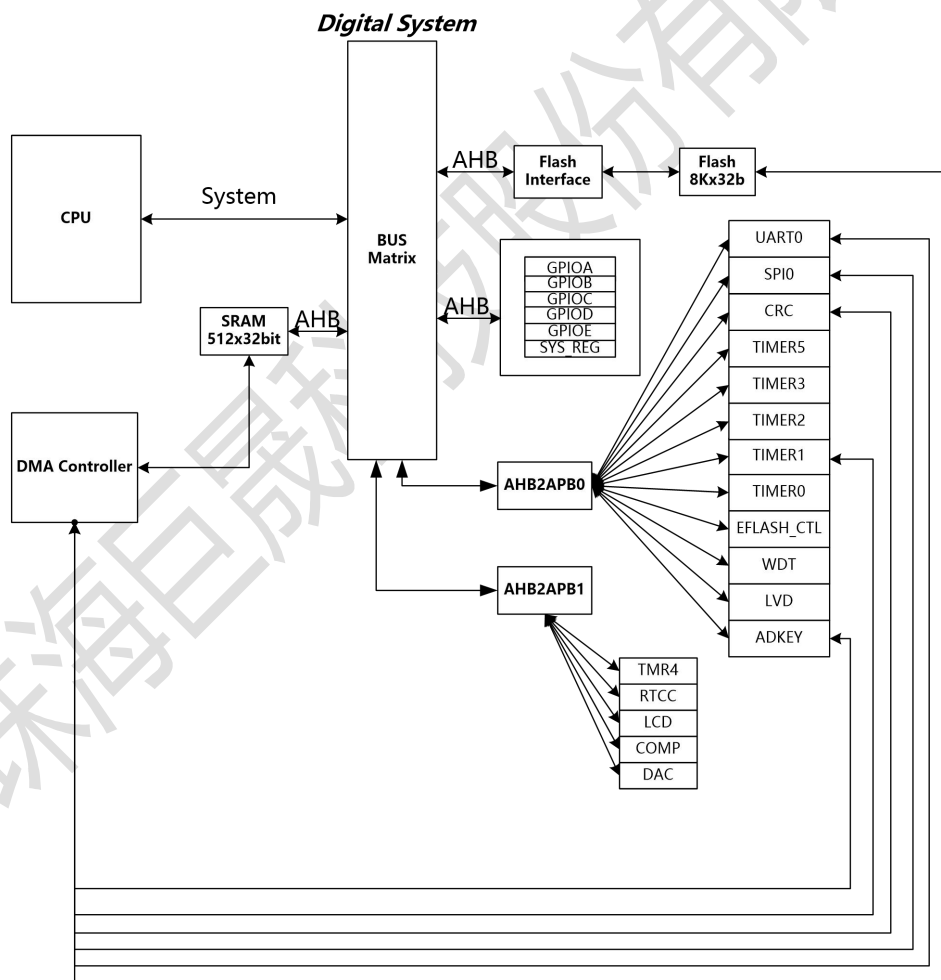


图 4-1 JS32L010 总线系统架构图

系统总线

此总线连接 CPU 内核的系统总线(外设总线)到总线矩阵，总线矩阵协调着内核和各个高速部件间的访问。

DMA 控制器

此总线将 CPU 与各外设模块访问相联竞争，协调访问优先级，仲裁等。

表 4-1 DMA 优先级

| 模块 | 优先级 | 备注 |
|--------|-----|------------|
| CPU | 0 | ↑ 最高 最低 |
| SPI0 | 1 | |
| UART0 | 2 | |
| ADKEY | 3 | |
| CRC | 4 | |
| TIMER1 | 5 | |
| FLASH | 6 | |

总线矩阵(Bus Matrix)

- 总线矩阵管理着内核系统总线与各外设模块的访问仲裁，总线矩阵由主模块总线及从模块总线组成。
- AHB 外设通过总线矩阵与系统总线相连。
- AHB 到 APB 桥(AHB2APB bridges-APB)。
- AHB 到 APB 桥在 AHB 与 APB 总线间提供同步连接。

注：当对 APB 寄存器进行 8 位或者 16 位访问时，该访问会被自动转换成 32 位的访问：桥会自动将 16 位或者 8 位的数据扩展以配合 32 位的宽度。

4.2 存储器映射

此 32 位处理器采用同一套总线来读取指令和加载/存储数据。指令代码和数据都位于相同的存储器地址空间，但在不同的地址范围。程序存储器，数据存储器，寄存器和 I/O 端口都在同一个线性的 4 GB 的地址空间之内。这是 32 位处理器的最大地址范围，因为它的地址总线宽度是 32 位。此外，为了降低不同客户在相同应用时的软件复杂度，存储映射是按 32 位处理器提供的规则预先定义的。在存储器映射表中，一部分地址空间由 32 位处理器的系统外设所占用，且不可更改。此外，其余部分地址空间可由芯片供应商定义使用。JS32L010 芯片的存储器映射表显示了 JS32L010 芯片的存储器映射，包括代码、SRAM、外设和其他预先定义的区域。简化了每个外设的地址译码。

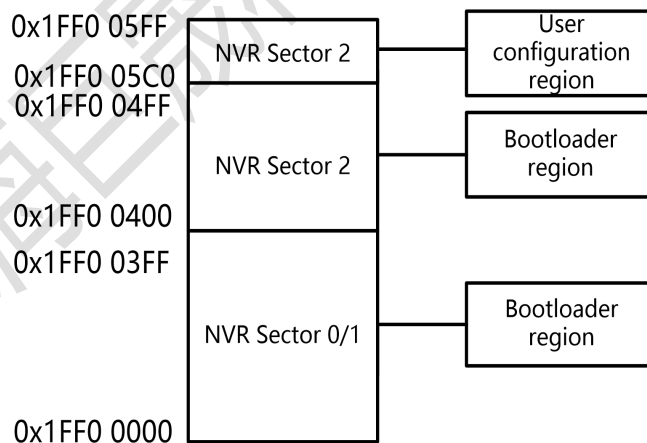


图 4-2 Flash NVR 区域定义

| | | | |
|----------------------------|---------------------|----------------------------|---------------|
| 0x5FFF FFFF | IO device | 0x5FFF FFFF | Reserved |
| | | | Reserved |
| | | 0x4002 1000 0x4002 0FFF | GPIOA/B/C/D/E |
| | | 0x4002 0B00 0x4002 0AFF | SYS_REG |
| 0x4000 0000 0x3FFF FFFF | Reserved | 0x4002 0000 0x4001 04FF | LCD |
| | | 0x4001 0400 0x4001 03FF | DAC |
| | | 0x4001 0220 0x4001 021F | COMP |
| | | 0x4001 0200 0x4001 01FF | TIMER4 |
| | | 0x4001 0100 0x4001 00FF | RTCC |
| | | 0x4001 0130 0x4001012F | TIMER4 |
| | | 0x4001 0100 0x4000 44FF | SPI0/IIC0 |
| | | 0x4000 4400 0x4000 40FF | UART0 |
| | | 0x4000 4000 0x4000 3FFF | EFLASH |
| | | 0x4000 3000 0x4000 2FFF | CRC |
| | | 0x4000 2000 0x4000 16FF | TIMER5 |
| | | 0x4000 1600 0x4000 14FF | TIMER3 |
| | | 0x4000 1400 0x4000 13FF | TIMER2 |
| 0x2000 0800 0x2000 07FF | SRAM (2KByte) | 0x4000 1300 0x4000 12FF | TIMER1_DMA |
| 0x2000 0000 0x1FFF FFFF | Reserved | 0x40001228 0x4000 1227 | TIMER1 |
| 0x1FF0 0600 0x1FF0 05FF | NVR | 0x4000 1200 0x4000 11FF | TIMER0 |
| 0x1FF0 0000 | Reserved | 0x4000 1100 0x4000 10FF | WDT |
| 0x0000 8000 0x0000 7FFF | eFlash (32KByte) | 0x4000 1000 0x4000 03FF | PLLFRAC |
| 0x0000 0000 | | 0x4000 0300 0x4000 02FF | ADKEY |
| | | 0x4000 0200 0x4000 01FF | LVD |
| | | 0x4000 0100 | |

System memory mapping

IO device space mapping

图 4-3 JS32L010 存储器映射表

4.2.1 片上 SRAM

芯片内置 2K 字节的 SRAM。它可以按字节(8 位)、半字(16 位)或字(32 位)进行访问。SRAM 起始地址为 0x2000 0000。可以被 CPU 或者 DMA 用最快的系统时钟且不插入任何等待进行访问。

注：DMA 支持访问有 EFLASH, UART, SPI, CRC, ADC, TIMER1。

4.2.2 片上 FLASH 概述

闪存存储器有两个不同存储区域：

- 主闪存存储块，它包括应用程序和用户数据区(若需要时)
 - 信息块，其包含两个部分：
 - 选项字节(Option bytes) - 内含硬件及存储保护用户配置选项。
 - 系统存储器(System memory) - 其包含 Boot loader 代码。详细参见内置闪存存储器章节。
- 闪存接口基于 AHB 协议执行指令和数据存取。其预取缓冲的功能可加速 CPU 执行代码的速度。

4.2.3 引导配置

芯片复位后，通过客户自己在副闪存的配置，选择启动模式引脚是 PC9 还是 PA3，默认工作是上拉还是下拉。正常启动后，CPU 从地址 0x0000 0000 获取堆栈顶的地址，并从存储器的 0x0000 0004 位置指示的地址开始执行代码。

Boot loader 程序存放在副闪存中，由用户根据实际情况自行决定是否写入使用。若已写入 Boot loader 程序，可以通过 UART 对 Boot loader 程序或者用户程序进行升级。

珠海巨晟科技股份有限公司

5 嵌入式闪存 (FLASH)

5.1 闪存主要特性

- 高达 32K 字节闪存存储器
- 存储器结构：
 - 主闪存空间：32K 字节
 - 副闪存空间：1.5K 字节
- 带预取缓冲器的读接口
- 闪存编程和擦除操作
- 访问和写保护
- 低功耗模式

5.2 闪存功能描述

5.2.1 闪存结构

闪存空间由 32 位宽的存储单元组成，既可以存代码又可以存数据。主闪存块按 32 页（每页 1K 字节）分块，以页为单位设置写保护（参见存储保护相关内容）。

表 5-1 闪存结构

| 模块 | 名称 | 地址 | 大小(字节) |
|---------|---------------|---------------------------|--------|
| 主闪存空间 | Page 0 | 0x0000 0000 – 0x0000 03FF | 1K |
| | Page 1 | 0x0000 0400 – 0x0000 07FF | 1K |
| | Page 2 | 0x0000 0800 – 0x0000 0BFF | 1K |
| | Page 3 | 0x0000 0C00 – 0x0000 0FFF | 1K |
| | ... | ... | 1K |
| | Page 30 | 0x0000 7800 – 0x0000 7BFF | 1K |
| | Page 31 | 0x0000 7C00 – 0x0000 7FFF | 1K |
| 副闪存空间 | Sector 0 | 0x1FF0 0000 – 0x1FF0 01FF | 512 |
| | Sector 1 | 0x1FF0 0200 – 0x1FF0 03FF | 512 |
| | Sector 2 | 0x1FF0 0400 – 0x1FF0 05FF | 512 |
| 闪存寄存器接口 | CTRLR0 | 0x4000 3000 – 0x4000 3003 | 4 |
| | KST | 0x4000 3004 – 0x4000 3007 | 4 |
| | DONE | 0x4000 3008 – 0x4000 300B | 4 |
| | PROG_ADDR | 0x4000 3010 – 0x4000 3013 | 4 |
| | PROG_DATA | 0x4000 3018 – 0x4000 301B | 4 |
| | ERASE_CTRL | 0x4000 3020 – 0x4000 3023 | 4 |
| | TIME_REG0 | 0x4000 3030 – 0x4000 3033 | 4 |
| | TIME_REG1 | 0x4000 3034 – 0x4000 3037 | 4 |
| | NVR_PASSWORD | 0x4000 3050 – 0x4000 3053 | 4 |
| | MAIN_PASSWORD | 0x4000 3054 – 0x4000 3057 | 4 |
| | CRC_ADDR | 0x4000 3058 – 0x4000 305B | 4 |
| | CRC_LEN | 0x4000 305C – 0x4000 305F | 4 |
| | CRC_OUT | 0x4000 3060 – 0x4000 3063 | 4 |
| | MODE | 0x4000 3070 – 0x4000 3073 | 4 |
| | STA1 | 0x4000 3074 – 0x4000 3077 | 4 |
| | STA2 | 0x4000 3078 – 0x4000 307B | 4 |
| | CFG_SECTOR | 0x4000 307C – 0x4000 307F | 4 |

5.2.2 闪存读保护

读操作在整个芯片工作电压范围内都可以完成，用于存放指令或者数据，若工作频率提升到 30MHz 以上，需要让 Flash 的读时序执行分频。

芯片带有缓冲区和预取缓冲区，提升 Flash 的访问效率。

当用户配置区经过自定义的保护配置后，若 UART 单线升级或 SWD 接口连接上时，会自动对 Flash 执行保护机制。

5.2.3 闪存烧写和擦除操作

烧写和擦除操作在整个芯片工作电压范围内都可以完成。

烧写和擦除操作由下列 6 个寄存器完成，先根据烧写的时钟配置好烧写时序(TIME_REG0)，再配置烧写密码，配置好编程地址，最后配置好编程数据，即可开始执行烧写，然后等待操作结束。

- 时序 0 寄存器 (TIME_REG0)
- 密码 0 寄存器 (NVR_PASSWORD)
- 密码 1 寄存器 (MAIN_PASSWORD)
- 编程地址寄存器 (PROG_ADDR)
- 编程数据寄存器 (PROG_DATA)
- 状态寄存器 (DONE)

5.2.4 闪存寄存器接口

5.2.4.1 配置寄存器 (CTRLR0)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|------------------|-------|----------|--|
| 31:17 | Reserved | - | - | - |
| 16 | PRO_CLK_SET | 1'b0 | RW | FLASH 烧写时钟源选择，推荐使用 RC 时钟 0: 高速 RC 时钟 2 分频 1: 晶振，若 RC 不准时才使用 |
| 15:12 | Reserved | - | - | - |
| 11 | LVD_PRO_EN | 1'b0 | RW | 在 LVD 断电时，是否允许打断 FLASH 编程和擦除，一般在断电时，立刻把重要数据保存在 FLASH 上，就打开此功能，用于快速让 FLASH 处于空闲状态 0: 不允许 1: 允许 |
| 10 | PRO_RAM_MODE_SET | 1'b0 | RW | 直接把 RAM 的数据写到 FLASH，并在 FLASH 最后自动添加 CRC 校验值，具体影响到几个寄存器 PROG_ADDR、CRC_ADDR、CRC_LEN 0: CRC 模式 1: RAM TO FLASH 模式 |
| 9 | READM1 | 1'b0 | RW | FLASH 测试使用，正常情况下配置成 0 |
| 8 | READM0 | 1'b0 | RW | FLASH 测试使用，正常情况下配置成 0 |
| 7 | SRAM_DIR_OUT_EN | 1'b0 | RW | FLASH 的数据直接输出，用于系统时钟低频时，且 CACHE_EN 和 PREFETCH_EN 都不使能时使用 0: 不使能 1: 使能 |
| 6 | Reserved | - | - | - |
| 5 | Reserved | - | - | - |

| | | | | |
|---|---------------|------|----|--|
| 4 | WRITE_BACK_EN | 1'b1 | RW | 编程/扇区擦除 自动回写到 CACHE 0: 不使能 1: 使能 |
| 3 | Reserved | - | - | - |
| 2 | PREFETCH_EN | 1'b0 | RW | 预取使能位 0: 不使能 1: 使能 |
| 1 | Reserved | - | - | - |
| 0 | CACHE_EN | 1'b0 | RW | CACHE 使能位 0: 不使能 1: 使能 |

5.2.4.2 触发寄存器 (KST)

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|-----------------|-------|----------|------------------------------------|
| 31:27 | Reserved | - | - | - |
| 26 | CRC_KICK_EN | 1'b0 | WO | FLASH CRC 校验使能 0: 不使能 1: 使能 |
| 25:21 | Reserved | - | - | - |
| 20 | CACHE_CLR_EN | 1'b0 | WO | CACHE 清除使能 0: 不使能 1: 使能 |
| 19:11 | Reserved | - | - | - |
| 10 | CRC_START | 1'b0 | WO | FLASH CRC Check, 和第 26 位同时写 '1' 触发 |
| 9:5 | Reserved | - | - | - |
| 4 | CACHE_CLR_START | 1'b0 | WO | CACHE 开始清除, 和第 20 位同时写 '1' 触发 |
| 3:0 | Reserved | - | - | - |

5.2.4.3 状态寄存器 (DONE)

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|----------------|-------|----------|-----------------------------|
| 31:13 | Reserved | - | - | - |
| 12 | ERASE_OK_FLAG | 1'b1 | RO | 0: 擦除失败 1: 擦除成功 |
| 11 | PRO_OK_FLAG | 1'b1 | RO | 0: 烧写失败 1: 烧写成功 |
| 10 | CRC_DONE | 1'b1 | RO | 0: 进行中 1: 空闲状态 |
| 9:7 | Reserved | - | - | - |
| 6 | PRO_DONE | 1'b1 | RO | 编程结束标志 0: 进行中 1: 空闲状态 |
| 5 | Reserved | - | - | - |
| 4 | CACHE_CLR_FLAG | 1'b1 | RO | CACHE 清除标志 0: 进行中 |

| | | | | |
|-----|-------------------|------|----|--|
| | | | | 1: 空闲状态 |
| 3:2 | Reserved | - | - | - |
| 1 | MAIN_ERASE_FLAG | 1'b1 | RO | MAIN 区域全擦除标志 0: 正在运行 1: 空闲状态 |
| 0 | SECTOR_ERASE_FLAG | 1'b1 | RO | 扇区(512 BYTE)擦除标志 0: 正在运行 1: 空闲状态 |

5.2.4.4 编程地址寄存器 (PROG_ADDR)

Address offset: 0x10

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|--|
| 31:30 | PRO_BYT_SET | 2'b10 | RW | 一次编程的 BYTE 位选择 b00: PROG_DATA[7:0]被编程 b01: PROG_DATA[15:0]被编程 b10: PROG_DATA[31:0]被编程 |
| 29 | PRO_NVR_SET | 1'b0 | RW | 编程的地址是否 NVR 区域 0: MAIN 区域 1: NVR 区域 |
| 28:2 | PRO_ADDR | 27'h0 | RW | FLASH 编程的地址, 以 4 字节为单位进行编程 NOTE: 在配置寄存器里的第 10 位有效时, 此寄存器作为 FLASH 编程的开始地址 |
| 1:0 | PRO_ADDR_L | 2'b0 | RO | 低 2 位地址固定为 0, 4 字节对齐 |

5.2.4.5 编程数据寄存器 (PROG_DATA)

Address offset: 0x18

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:0 | PRO_DATA | 32'h0 | RW | FLASH 编程的数据, 需要配置好地址才可进行 NOTE: 在配置寄存器里的第 10 位有效时, 此寄存器作为 FLASH 编程的数据 |

5.2.4.6 擦除控制寄存器 (ERASE_CTRL)

Address offset: 0x20

| Width | Name | Reset | Property | Description |
|-------|--------------------|-------|----------|---|
| 31 | CHIP_ERASE_START | 1'b0 | RO | CHIP 全擦除的触发, 写 “1” 触发, 需要先配置密码 |
| 30 | SECTOR_ERASE_START | 1'b0 | RO | SECTOR 擦除的触发, 写 “1” 触发, 需要先配置密码 |
| 29 | NVR_SECTOR_EN | 1'b0 | RW | NVR 的 SECTOR 使能位 0: MAIN 区域 1: NVR 区域 |
| 28:7 | Reserved | - | - | - |
| 6:0 | ERASE_SECTOR_ADDR | 7'h0 | RW | 擦除的 SECTOR 选择, 范围 0-127 |

5.2.4.7 时序 0 寄存器 (TIME_REG0)

Address offset: 0x30

| Width | Name | Reset | Property | Description |
|-------|----------|---------|----------|------------------------------|
| 31:20 | Reserved | - | - | - |
| 19:16 | PGH | 4'b0001 | RW | WEB 低 到 PROG2 高 保持时间最小为 15ns |
| 15:12 | ADS | 4'b0001 | RW | 字节/地址/数据设置的最小时间是 15ns |
| 11:8 | ADH | 4'b0001 | RW | 字节/地址/数据保持时间最小为 15ns |
| 7:4 | RW | 4'b1000 | RW | 编程/擦除后的下一个操作的延迟时间为 100ns |
| 3:0 | RC | 1'b0 | RW | 读取周期最小时间为 25/30ns |

5.2.4.8 时序 1 寄存器 (TIME_REG1)

Address offset: 0x34

| Width | Name | Reset | Property | Description |
|-------|-------------|---------|----------|--------------------------------|
| 31:20 | Reserved | - | - | - |
| 18:8 | 1MS_CNT_SET | 11'h3E8 | RO | 1ms 的时间配置值, 以 1us 为单位 |
| 7:0 | 1US_CNT_SET | 8'h0D | RW | 1us 的时间配置值, 系统默认是 26MHz 的 2 分频 |

5.2.4.9 密码 1 寄存器 (MAIN_PASSWORD)

Address offset: 0x54

| Width | Name | Reset | Property | Description |
|-------|----------------|-------|----------|---|
| 31:10 | MAIN_PASS_WORD | 32'h0 | RW | 密码为 0x20170230, 只有打开密码后, 才能对 MAIN 进行擦除和编程 |

5.2.4.10 密码 0 寄存器 (NVR_PASSWORD)

Address offset: 0x50

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|--|
| 31:10 | NVR_PASS_WORD | 32'h0 | RW | 密码为 0x20150931, 只有打开密码后, 才能对 NVR 进行擦除和编程 |

5.2.4.11 CRC 地址配置寄存器 (CRC_ADDR)

Address offset: 0x58

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:30 | Reserved | - | - | - |
| 29 | NVR_SET | 1'b0 | RW | 地址是否 NVR 区域 0: MAIN 区域 1: NVR 区域 |
| 28:2 | DMA_ADDR | 27'h0 | RW | CRC DMA 的开始地址, 只指向 EFLASH, 且是物理地址(4 字节对齐) NOTE: 在配置寄存器里的第 10 位有效时, 此寄存器配置成 RAM 的地址 |
| 1:0 | Reserved | - | - | - |

5.2.4.12 CRC 长度配置寄存器 (CRC_LEN)

Address offset: 0x5C

| Width | Name | Reset | Property | Description |
|-------|---------|-------|----------|--|
| 31:0 | DMA_LEN | 32'h0 | RW | CRC 长度 Note: 在配置寄存器里的第 10 位有效时, 此寄存器配置成 RAM 的长度 |

5.2.4.13 CRC 结果寄存器 (CRC_OUT)

Address offset: 0x60

| Width | Name | Reset | Property | Description |
|-------|---------|-------|----------|--|
| 31:0 | CRC_OUT | 32'h0 | RO | CRC 的结果, 多项式 CRC-32 如下: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ |

5.2.4.14 模式寄存器(MODE)

Address offset: 0x70

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|---|
| 31:19 | Reserved | - | - | - |
| 18 | DEBUG_MODE | 1'b0 | RO | FLASH 进入程序 DEBUG 模式标志位 0: 无效 1: 有效 |
| 17 | UPDATE_MODE | 1'b0 | RO | FLASH 进入程序单 PIN 升级模式标志位 0: 无效 1: 有效 |
| 16 | TEST_MODE | 1'b0 | RO | EFLASH 进入出厂 TEST 模式标志位 0: 无效 1: 有效 |
| 15:4 | Reserved | - | - | - |
| 3 | PROT0_MODE | 1'b0 | RO | FLASH 进入全被保护模式标志位 0: 无效 1: 有效 |
| 2 | ALL_MODE | 1'b0 | RO | FLASH 进入自由模式标志位 0: 无效 1: 有效 |
| 1 | SUPER_MODE | 1'b0 | RO | FLASH 进入超级模式标志位 0: 无效 1: 有效 |
| 0 | Reserved | - | - | - |

5.2.4.15 状态寄存器 1(STA1)

Address offset: 0x74

| Width | Name | Reset | Property | Description |
|-------|-----------------|-------|----------|--|
| 31 | MAIN_FALID_FLAG | 1'b0 | RO | FLASH MAIN 区域 CODE CRC 校验出错标志位 0: 正确 1: 错误 |
| 30 | BOOT_FALID_FLAG | 1'b0 | RO | FLASH 有 Bootloader 的区域 CRC 校验出错标志位 |

| | | | | |
|-------|-------------------------|------|----|--|
| | | | | 志位 0: 正确 1: 错误 |
| 29 | MAIN_EPMTY_FLAG | 1'b1 | RO | FLASH MAIN 区域 CODE 全空标志位 0: 非空 1: 全空 |
| 28 | EFLASH_ALL_OK_FLAG | 1'b1 | RO | FLASH 所有的 SECTOR 都是好片标志位 0: 有坏片 1: 全是好片 |
| 27:23 | Reserved | - | - | - |
| 22 | BOOT_USR_CFG_VALID_FLAG | 1'b0 | RO | FLASH Bootloader+用户配置拷贝有效标志位 0: 无效 1: 有效 |
| 21 | BOOT_USR_CFG_DIR_FLAG | 1'b0 | RO | FLASH Bootloader+用户配置拷贝方向标志位 0: NVR 区更新到 MAIN 区 1: MAIN 区更新到 NVR 区 |
| 20 | BOOT_USR_CFG_OK_FLAG | 1'b0 | RO | FLASH Bootloader+用户配置拷贝后 CRC 校验正确标志位 0: 错误 1: 正确 |
| 19 | MAIN_BOOT_EMPTY_FLAG | 1'b1 | RO | FLASH Main 区 Bootloader 全空标志位 0: 非空 1: 全空 |
| 18 | MAIN_BOOT_OK_FLAG | 1'b0 | RO | FLASH MAIN 区 BootloaderCRC 校验正确标志位 0: 错误 1: 正确 |
| 17 | MAIN_USER_EMPTY_FLAG | 1'b0 | RO | FLASH MAIN 区用户配置区全空标志位 0: 非空 1: 全空 |
| 16 | MAIN_USER_OK_FLAG | 1'b0 | RO | FLASH MAIN 区用户配置 CRC 校验正确标志位 0: 错误 1: 正确 |
| 15 | NVR_BOOT_EMPTY_FLAG | 1'b1 | RO | FLASH NVR 区 Bootloader 全空标志位 0: 非空 1: 全空 |
| 14 | NVR_BOOT_OK_FLAG | 1'b0 | RO | FLASH NVR 区 Bootloader CRC 校验正确标志位 0: 错误 1: 正确 |
| 13 | NVR_USER_EMPTY_FLAG | 1'b0 | RO | FLASH NVR 区用户配置区全空标志位 0: 非空 1: 全空 |
| 12 | NVR_USER_OK_FLAG | 1'b0 | RO | FLASH NVR 区用户配置 CRC 校验正确标志位 0: 错误 1: 正确 |
| 11:0 | Reserved | - | - | - |

5.2.4.16 状态寄存器 2(STA2)

Address offset: 0x78

| Width | Name | Reset | Property | Description |
|-------|--------------------|-------|----------|--|
| 31:20 | Reserved | - | - | - |
| 19:17 | Reserved | - | - | - |
| 16 | CHIP_ERASE_OK_FLAG | 1'b0 | RO | FLASH CHIP 擦写成功标志位 0: 擦写失败 1: 擦写成功 |
| 15:12 | Reserved | - | - | - |
| 11:8 | Reserved | - | - | - |
| 7:6 | Reserved | - | - | - |
| 5 | Reserved | - | - | - |
| 4 | RC_CLK_SEL | 1'b0 | RO | RC 启动时钟选择位 0: RC 启动时钟为 128KHz 频率 1: RC 启动时钟为 HROSC 的 2 分频(13MHz) |
| 3 | Reserved | - | - | - |
| 2 | MAIN_CRC_EN | 1'b1 | RO | MAIN 区 CRC 校验使能位 0: 不使能 1: 使能 |
| 1 | BOOT_BKP_EN | 1'b0 | RO | Bootloader 备份使能位 0: 不使能 1: 使能 |
| 0 | SCRAMBLE_EN | 1'b1 | RO | EFLASH 加扰有效位标志位 0: 无效 1: 有效 |

5.2.4.17 主扇区信息寄存器 (CFG_SECTOR)

Address offset: 0x7C

| Width | Name | Reset | Property | Description |
|-------|------------|--------|----------|-------------|
| 31:00 | SECTOR_NUM | 32'h40 | RO | 主程序区最大扇区个数 |

5.2.5 NVR2 闪存用户配置区

5.2.5.1 配置 0 (0x1FF0 05C0)

| Width | Name | Reset | Property | Description |
|-------|------------------|-------|----------|--|
| 31:11 | Reserved | - | - | - |
| 10 | CODE_PROTECT_DIS | 1'b1 | RO | 代码保护使能位 0: 保护 1: 不保护 |
| 9 | BOOT_EN | 1'b1 | RO | BOOTLOADER 使能位 0: 使能 1: 不使能 |
| 8 | NVR_BACKUP_EN | 1'b1 | RO | NVR0/1/2 区域是否需要备份到 MAIN 区域中 0: 备份 1: 不备份 |
| 7:2 | Reserved | - | - | - |

| | | | | |
|-----|---------------|-------|----|-------------------------|
| 1:0 | NVR_WRITE_PER | 2'b11 | RO | NVR 的写权限，每一位代表一个 SECTOR |
|-----|---------------|-------|----|-------------------------|

5.2.5.2 配置 1 (0x1FF0 05C4)

| Width | Name | Reset | Property | Description |
|-------|----------------|-------------|----------|--|
| 31:0 | MAIN_WRITE_PER | 32'hFFFFFFF | RO | MAIN 的写权限，根据 FLASH 的存储空间大小确定每一位控制多少字节 每一位代表 1K BYTE |

5.2.5.3 配置 2 (0x1FF0 05D4)

| Width | Name | Reset | Property | Description |
|-------|------------------|----------|----------|--|
| 31 | MAIN_CODE_CRC_EN | 1'b1 | RO | MAIN 区域程序 CRC 检验的使能位 0: 关闭 1: 打开 |
| 30:16 | Reserved | - | - | - |
| 15:0 | MAIN_CODE_LEN | 16'hFFFF | RO | MAIN 区域程序的长度，4 字节为单元。 MAIN 区域除去该程序区域后，剩下的为 DATA 区域。 |

5.2.5.4 配置 3 (0x1FF0 05E0)

| Width | Name | Reset | Property | Description |
|-------|-----------------|-------|----------|---|
| 31:15 | Reserved | - | - | - |
| 14 | SWD_EN | 1'b1 | RO | SWD 接口是否有效 0: 不使能 1: 使能 |
| 13 | Reserved | - | - | - |
| 12 | MCLR_EN | 1'b1 | RO | MCLR 使能位 0: 不使能 1: 使能 |
| 11 | Reserved | - | - | - |
| 10:9 | SWD_PULL_MODE | 1'b11 | RO | SWD 复位上拉或下拉，根据此位来确定默认为复位上拉或下拉 b10: 下拉 b11: 上拉 其它: 没有上下拉 (SWD 无效) |
| 8 | Reserved | - | - | - |
| 7:6 | Reserved | - | - | - |
| 5:0 | UART_UPDATE_MAP | 6'h3F | RO | UART Port 单线升级映射 h00~h01: 分别对应 PA0~PA1 h04~h0F: 分别对应 PA4~PA15 h10~h1F: 分别对应 PB0~PB15 h20~h27: 分别对应 PC0~PC7 h2A: 对应 PE1 h2B: 对应 PE0 h2C: 对应 PE2 h2D~h32: 分别对应 PC10~PC15 h33~h38: 分别对应 PD0~PD5 |

| | | | | |
|--|--|--|--|-------------------|
| | | | | hFF: 无效 其它: 禁止 |
|--|--|--|--|-------------------|

5.2.6 NVR3 闪存芯片信息区

5.2.6.1 芯片信息 0(0x1FF0 0600)

| Width | Name | Reset | Property | Description |
|-------|------|--------------|----------|-------------|
| 31:0 | UID0 | 32'hFFFFFFFF | RO | UID 信息 0 |

5.2.6.2 芯片信息 0(0x1FF0 0604)

| Width | Name | Reset | Property | Description |
|-------|------|--------------|----------|-------------|
| 31:0 | UID1 | 32'hFFFFFFFF | RO | UID 信息 1 |

5.2.6.3 芯片信息 0(0x1FF0 0608)

| Width | Name | Reset | Property | Description |
|-------|------|--------------|----------|-------------|
| 31:0 | UID2 | 32'hFFFFFFFF | RO | UID 信息 2 |

5.2.6.4 芯片信息 0(0x1FF0 0618)

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|-------------|
| 31:0 | CUSTOMER ID | 32'b0 | RO | 客户信息 |

6 中断和事件(INT/EVT)

6.1 嵌套向量中断控制器

- 特征
- 中断都可屏蔽(除了 NMI)
- 4 个可编程的优先等级
- 低延迟的异常和中断处理
- 电源管理控制

系统控制寄存器的实现

嵌套向量中断控制器(NVIC)和处理器核的接口紧密相连,可以实现低延迟的中断处理和高效地处理晚到的中断。

嵌套向量中断控制器管理着包括核异常等中断。关于更多的异常和 NVIC 编程的说明请参考 CPU 技术参考手册。

6.2 系统滴答(SysTick)校准值寄存器

本芯片支持系统滴答计时。

6.3 中断功能描述

处理器和嵌套式矢量型中断控制器(NVIC)在处理(Handler)模式下对所有异常进行优先级区分以及处理。当异常发生时,系统会自动将当前处理器工作状态压栈,在执行完中断服务子程序 (ISR)后自动将其出栈。

取向量是和当前工作状态压栈并行进行的,从而提高了中断入口效率。处理器支持咬尾中断,可实现背靠背中断,大大削减了反复切换工作状态所带来的负荷。

表 6-1 NVIC 异常类型

| 异常类型 | 向量编号 | 优先级 | 向量地址 | 描述 |
|-----------------|-------|---------|-----------------------------|-------------------|
| - | 0 | -- | 0x0000 0000 | Reserved |
| 复位 | 1 | -3 (最高) | 0x0000 0004 | 复位 |
| NMI | 2 | -2 | 0x0000 0008 | 不可屏蔽中断 |
| 硬件故障(HardFault) | 3 | -1 | 0x0000 000C | 各种硬件级别故障 |
| - | 4-10 | -- | 0x0000 0010- 0x0000 002B | Reserved |
| SVcall 服务调用 | 11 | -- | 0x0000 002C | 通过 SWI 指令实现系统服务调用 |
| - | 12-13 | -- | 0x0000 0030- 0x0000 0037 | Reserved |
| PendSV 挂起服务 | 14 | 可编程设置 | 0x0000 0038 | 可挂起的系统服务请求 |
| 系统节拍 | 15 | 可编程设置 | 0x0000 003C | 系统节拍定时器 |

表 6-2 中断向量表

| 中断编号 | 向量编号 | 外设中断描述 | 向量地址 |
|------|------|------------|-------------|
| IRQ0 | 16 | LVD_IRQn | 0x0000 0040 |
| IRQ1 | 17 | UART0_IRQn | 0x0000 0044 |
| IRQ2 | 18 | SPI0_IRQn | 0x0000 0048 |
| IRQ3 | 19 | GPIOA_IRQn | 0x0000 004C |
| IRQ4 | 20 | GPIOB_IRQn | 0x0000 0050 |
| IRQ5 | 21 | GPIOC_IRQn | 0x0000 0054 |
| IRQ6 | 22 | GPIOD_IRQn | 0x0000 0058 |

| | | | |
|-------|----|-------------------|-------------|
| IRQ7 | 23 | GPIOE_IRQn | 0x0000 005C |
| IRQ8 | 24 | WKPND_IRQn | 0x0000 0060 |
| IRQ9 | 25 | TIM0_IRQn | 0x0000 0064 |
| IRQ10 | 26 | TIM1_IRQn | 0x0000 0068 |
| IRQ11 | 27 | TIM2_IRQn | 0x0000 006C |
| IRQ12 | 28 | TIM3_IRQn | 0x0000 0070 |
| IRQ13 | 29 | TIM4_IRQn | 0x0000 0074 |
| IRQ14 | 30 | TIM5_IRQn | 0x0000 0078 |
| IRQ15 | 31 | - | 0x0000 007C |
| IRQ16 | 32 | - | 0x0000 0080 |
| IRQ17 | 33 | ADKEY_IRQn | 0x0000 0084 |
| IRQ18 | 34 | POWER_UPDATE_IRQn | 0x0000 0088 |
| IRQ19 | 35 | CRC_DMA_IRQn | 0x0000 008C |
| IRQ20 | 36 | WDT_IRQn | 0x0000 0090 |
| IRQ21 | 37 | RTCC_IRQn | 0x0000 0094 |
| IRQ22 | 38 | LCD_FRAME_IRQn | 0x0000 0098 |
| IRQ23 | 39 | COMP_IRQn | 0x0000 009C |
| IRQ24 | 40 | - | 0x0000 00A0 |
| IRQ25 | 41 | - | 0x0000 00A4 |
| IRQ26 | 42 | - | 0x0000 00A8 |
| IRQ27 | 43 | - | 0x0000 00AC |
| IRQ28 | 44 | - | 0x0000 00B0 |
| IRQ29 | 45 | - | 0x0000 00B4 |
| IRQ30 | 46 | - | 0x0000 00B8 |
| IRQ31 | 47 | - | 0x0000 00BC |

6.4 外部中断/事件控制器(EXTI)

外部中断和事件控制器(EXTI)管理外部和内部异步事件/中断，并生成相应的事件请求到 CPU/中断控制器和到电源管理的唤醒请求。每个输入线可以独立地配置输入类型（脉冲或挂起）和对应的触发事件（上升沿或下降沿或者双边沿都触发）。每个输入线都可以独立地被屏蔽。挂起寄存器保持着状态线的中断请求。

6.4.1 主要特征

EXTI 控制器的主要特性如下：

- 每个中断/事件都有独立的触发和屏蔽
- 每个中断线都有专用的状态位
- 支持多达 27 个软件的中断/事件请求
- 检测脉冲宽度低于 APB0 时钟宽度的外部信号。参见用户手册中电气特性部分的相关参数。

6.4.2 唤醒事件管理

JS32L010 可以处理外部或内部事件来唤醒内核(WFE)。唤醒事件可以通过下述配置产生：

- 外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时在 CPU 的系统控制寄存器中使能 SEVONPEND 位。当 CPU 从 WFE 恢复后，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）。
- 配置一个外部或内部 EXTI 线为事件模式，当 CPU 从 WFE 恢复后，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

7 循环冗余校验计算单元 (CRC)

7.1 简介

循环冗余校验(CRC)计算单元是根据自定义的生成多项式得到任意一个 32 位全字的 CRC 计算结果。在其他的应用中, CRC 技术主要应用于核实数据传输或者数据存储的正确性和完整性。CRC 计算单元可以在程序运行时计算出软件的标识, 之后与在连接时生成的参考标识比较, 然后存放在指定的存储器空间。

7.2 模块框图

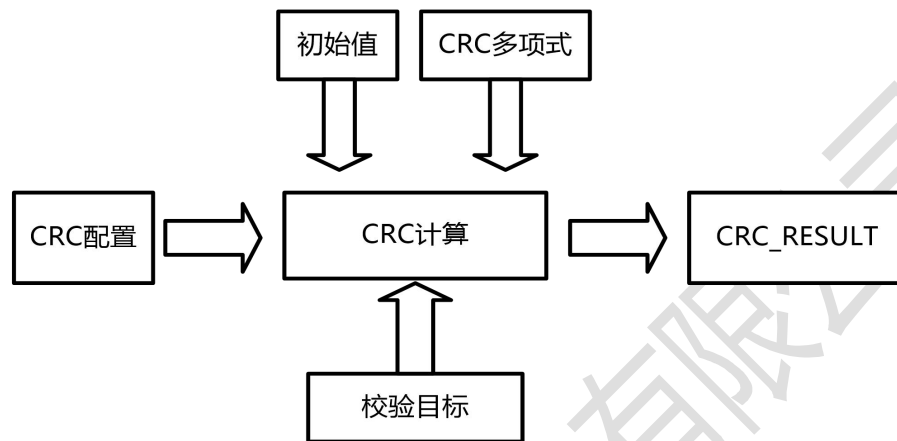


图 7-1 CRC 框图

7.3 主要特性

- 支持 5/7/8/16/32 位等不同长度的多项式
- 支持自定义的多项式
- 默认是 32 位多项式:
- $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
- 一个 32 位初始值配置寄存器用于输入, 一个 32 位结果寄存器用于输出结果

7.4 寄存器定义

7.4.1 控制配置寄存器 (CRC_CFG)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|--------------|-------|----------|-----------------------------------|
| 31:14 | Reserved | - | - | - |
| 13:8 | POLY_WIDTH | 6'h20 | RW | 支持 5/7/8/16/32 等不同长度的多项式 |
| 7:2 | Reserved | - | - | - |
| 1 | BIT_ORDER_EN | 1'b0 | RW | 输入数据的顺序 0: 表示低位在右 1: 表示低位在左 |
| 0 | INT_EN | 1'b0 | RW | CRC 中断 0: 不使能 1: 使能 |

7.4.2 初始值配置寄存器(CRC_INIT)

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|------------|-------------------|----------|-------------|
| 31:0 | INIT_VALUE | 32'hFFFFFF FFF | RW | CRC 初始值 |

7.4.3 取反寄存器(CRC_INV)

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|-----------|-------------------|----------|-------------|
| 31:0 | INV_VALUE | 32'hFFFFFF FFF | RW | CRC 输出结果取反 |

7.4.4 多项式配置寄存器(CRC_POLY)

Address offset: 0x0C

| Width | Name | Reset | Property | Description |
|-------|------------|------------------|----------|----------------------|
| 31:0 | POLY_VALUE | 32'hEDB8 8320 | RW | CRC 多项式, 默认为 32 位多项式 |

7.4.5 触发寄存器(CRC_KST)

Address offset: 0x10

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|-------------|
| 31:1 | Reserved | - | - | - |
| 0 | PENDING_CLR | 1'b0 | WO | 清除 CRC 状态位 |

7.4.6 状态寄存器(CRC_STA)

Address offset: 0x14

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:1 | Reserved | - | - | - |
| 0 | PENDING | 1'b0 | RO | CRC 状态位 |

7.4.7 地址寄存器(CRC_ADDR)

Address offset: 0x1C

| Width | Name | Reset | Property | Description |
|-------|----------------|-------|----------|---|
| 31:0 | DMA_ADDR_START | 32'h0 | RO | CRC DMA 开始的物理地址(WORD 对齐) 注意: 物理地址, 比如地址为 0x0 重定向 0x2000 0000 |

7.4.8 长度寄存器(CRC_LEN)

Address offset: 0x20

| Width | Name | Reset | Property | Description |
|-------|---------|-------|----------|-------------|
| 31:0 | DMA_LEN | 32'h0 | RW | CRC DMA 长度 |

7.4.9 结果寄存器(CRC_OUT)

Address offset: 0x24

| Width | Name | Reset | Property | Description |
|-------|---------|-------|----------|-------------|
| 31:0 | CRC_OUT | 32'h0 | RO | CRC 输出结果 |

7.5 操作流程

- 配置 CRC 初始值(CRC_INIT)和取反(CRC_INV)。

- 配置 CRC 多项式(CRC_POLY), CRC 多项式的位宽(POLY_WIDTH), 输入数据的顺序(BIT_ORDER_EN)。
- 配置 DMA 的开始地址(物理地址), 最后配置 DMA 的长度, 一旦长度不为 0, 即开始执行 CRC 校验。
- 等待 CRC 的 Pending(CRC_STA), 清除 Pending(CRC_KST), 最后得到 CRC 的结果(CRC_OUT)。

珠海巨晟科技股份有限公司

8 电源管理 (POWER MANAGEMENT)

8.1 电源

芯片的工作电压为 1.8-3.6V。本芯片采用 Cap-Less 设计，无需在内置 LDO 输出上外挂电容。内置 LDO 具有 2 挡位下拉电流使能 PMUCON0[18:17]，同时也有过流保护功能寄存器 PMUCON0[16]，内部逻辑系统电流超过 250mA 属于异常判断。

8.1.1 电压调节器

复位后调节器总是使能的。在需要低功耗的场合，可以使能低功耗工作模式。

8.2 电源管理器

8.2.1 上电复位(POR) 和掉电复位(PDR)

JS32L010 内部有一个完整的上电复位(POR)和掉电复位(PDR) 电路，当供电电压达到 1.8V 时系统即能正常工作。当 VDD 低于指定的限位电压 VPOR/VPDR 时，系统保持为复位状态，无需外部复位电路。

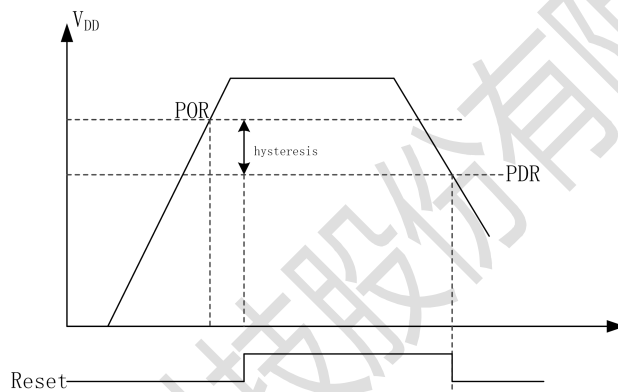


图 8-1 上下电复位波形图

8.2.2 可编程电压监测器(PVD)

JS32L010 内部集成两个电压检测器，一个检测外部供电 VCC，一个检测内部 LDO 输出 VDD，LDO 采用 Cap-Less 结构，封装上 VDD 不可见。两种检测电压均阈值可选。当系统监测到 VCC 或 VDD 电压低于配置电压值时，可以选择触发系统复位或通过使能 PVD 中断进入中断子函数。这一特性可用于执行紧急关闭任务。检测信号可以选择经过毛刺滤波电路或直接检测，由 LVDCON 的 LVD_VCC_BPS_EN 和 LVDCON 的 LVD_VDD_BPS_EN 来控制。

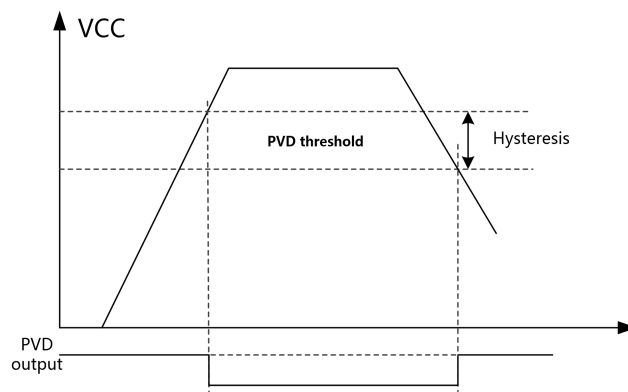


图 8-2 掉电检测波形图

8.3 电源控制寄存器

8.3.1 寄存器: LVDCON

Address: 0x4000A100

| Width | Name | Reset | Property | Description |
|-------|----------------------|---------|----------|---|
| 31 | LVD_VDD_PENDING | 1'b0 | RW | VDD 电压低于设定阈值触发 PENDING, 写 0 清除 |
| 30 | LVD_VCC_PENDING | 1'b0 | RW | VCC 电压低于设定阈值触发 PENDING, 写 0 清除 |
| 29 | LVD_OCP_PENDING | 1'b0 | RW | VDD LDO 电流超出保护范围触发 PENDING, 写 0 清除 |
| 28:22 | LVD_VDD_DBS_LO_LIMIT | 7'h0 | RW | 对 LVD_VDD 信号的低电平毛刺滤波宽度, 单位为系统时钟周期。 |
| 21:15 | LVD_VDD_DBS_HI_LIMIT | 7'h0 | RW | 对 LVD_VDD 信号的高电平毛刺滤波宽度, 单位为系统时钟周期 |
| 14 | LVD_VCC_SYNC_DIS | 1'b0 | RW | 检测到 VCC 低电压信号触发 GPIO 复位是否需要同步 0: 需要同步 1: 不同步 |
| 13 | LVD_VDD_BPS_EN | 1'b0 | RW | 屏蔽对 VDD 电压状态信号的 DEBOUNCE 滤波处理 0: 不使能 1: 使能 |
| 12 | LVD_VCC_BPS_EN | 1'b0 | RW | 屏蔽对 VCC 电压状态信号的 DEBOUNCE 滤波处理 0: 不使能 1: 使能 |
| 11 | LVD_OE | 1'b1 | RW | 使能阈值判断触发条件后中断, 以及复位功能 0: 不使能 1: 使能 |
| 10 | LVD_VDD_RST_EN | 1'b1 | RW | VDD 阈值判断触发后复位系统使能 0: 中断, 不复位 1: 复位, 不中断 |
| 9 | LVD_VCC_RST_EN | 1'b1 | RW | VCC 阈值判断触发后复位系统使能 0: 中断, 不复位 1: 复位, 不中断 |
| 8 | LVD_OCP_INT_EN | 1'b0 | RW | 过流保护信号产生中断使能 0: 不使能 1: 使能 |
| 7 | LVD_OCP_BPS_EN | 1'b0 | RW | 屏蔽对过流保护信号的消抖处理 0: 不使能 1: 使能 |
| 6:3 | H_LVD_S | 4'b0100 | RW | VCC LVD 设置(单位 V): b0000: 1.5 b0001: 1.55 b0010: 1.6 b0011: 1.65 |

| | | | | |
|---|----------|------|----|---|
| | | | | b0100: 1.7 b0101: 1.75 b0110: 1.8 b0111: 1.9 b1000: 2.0 b1001: 2.1 b1010: 2.2 b1011: 2.3 b1100: 2.4 b1101: 2.5 b1110: 2.6 b1111: 2.7 |
| 2 | L_LVD_S | 1'b0 | RW | VDD LVD 检测设置(单位 V) 0: 1.25 1: 1.35 |
| 1 | L_LVD_EN | 1'b1 | RW | VDD LVD 使能 0: 不使能 1: 使能 |
| 0 | HLVD_EN | 1'b1 | RW | VCC LVD 使能 0: 不使能 1: 使能 |

8.3.2 寄存器 LVDCON1

Address: 0x4000A104

| Width | Name | Reset | Property | Description |
|-------|---------------------|-------|----------|------------------------------------|
| 31 | Reserved | - | - | - |
| 30:24 | OCP_DBS_LO_LIMIT | 7'h0 | RW | 对过流保护信号的低电平毛刺滤波宽度, 单位为系统时钟周期。 |
| 23 | Reserved | - | - | - |
| 22:16 | OCP_DBS_HI_LIMIT | 7'h0 | RW | 对过流保护信号的高电平毛刺滤波宽度, 单位为系统时钟周期。 |
| 15 | Reserved | - | - | - |
| 14:8 | LVDVCC_DBS_LO_LIMIT | 7'h0 | RW | 对 LVD VCC 信号的低电平毛刺滤波宽度, 单位为系统时钟周期。 |
| 7 | Reserved | - | - | - |
| 6:0 | LVDVCC_DBS_HI_LIMIT | 7'h0 | RW | 对 LVD VCC 信号的高电平毛刺滤波宽度, 单位为系统时钟周期。 |

9 低功耗(LOW POWER)

9.1 低功耗模式

在系统或电源复位以后，微控制器处于正常模式运行状态，系统所用时钟为 128KHz 内部 RC 振荡器输出。当 CPU 不需继续运行时，可以利用进入多种低功耗模式来节省功耗。例如等待某个外部事件时，用户需要根据最低电源消耗、最快启动时间和可用的唤醒源等条件，选定一个最佳的低功耗模式。

三种低功耗模式：

- 待机模式(Idle Mode)
- 停止模式(Stop Mode)
- 睡眠模式(Sleep Mode)

表 9-1 低功耗模式说明

| 待机模式(Idle Mode) | 停止模式(Stop Mode) | 睡眠模式(Sleep Mode) |
|--------------------|------------------------------|---------------------------------|
| - | 数字模块时钟停止 | 数字模块时钟停止 |
| - | 可选：片内 26M 时钟源关闭(设置 LPCON[3]) | 可选：片内 26MHz 时钟源关闭(设置 LPCON[3]) |
| - | 可选：SRAM 关闭(设置 LPCON[2]) | 可选：SRAM 关闭(设置 LPCON[2]) |
| - | - | 可选：片内 128KHz 时钟源关闭(设置 LPCON[4]) |
| - | - | 可选：PMU 进入低功耗模式(设置 LPCON[7]) |
| - | FLASH 进入普通睡眠模式 | FLASH 进入深度睡眠模式(功耗更低) |
| - | - | 可选：唤醒后复位(设置 SYSCON0[30]) |
| CPU 停止 | CPU 停止 | CPU 停止 |
| _ASM("WFI")指令进入该模式 | 配置 LPCON[1]进入该模式 | 配置 LPCON[0]进入该模式 |

注：可以通过关闭未使用的外设、时钟源使功耗降到最低。

此外，在运行模式下，可以通过以下方式中的一种降低功耗：

- 降低系统时钟
- 关闭 APB 和 AHB 总线上未被使用的外设时钟。
- 合理配置 APB 与 AHB 的频率关系

9.2 进入低功耗

进入低功耗模式(Sleep)步骤：

- 关闭相应的模拟模块，以及 PMU 的部分功能(PMU_BGBUF, PMU_TSEN 等)。
- 切换系统时钟到 128KHz（依靠端口唤醒）或者 32KHz（只依靠其它源唤醒），并且关闭其它时钟源。
- 配置 PMU 全面进入低功耗。
- 关闭 LCD(若有 LCD)模拟模块的一些非必要功能 (VFBDET_EN,VTEST_EN,VDET_EN，这些功能默认是关闭的)。
- 无用的 IO 都设成模拟模式。
- 进入 Sleep 前的数字准备（配置唤醒源，使能 WKUP_EN，初始化 WKPND_IRQn，关闭看门狗，使能进入 Sleep 时自动关闭 128KHz 内部 RC 振荡器的选项，使能 SLEEP_GOON_EN 唤醒时不复位）。
- 配置 LP_CON0[0]进入 Sleep。

9.3 低功耗唤醒

支持多种唤醒方式：

- 端口唤醒
总共有 12 组信号源（由 WKUP_CON[11:0]控制使能），每组信号源是 3~6 个 IO 的或逻辑。

这些唤醒源唤醒之后是会产生中断并且有对应的中断状态位，中断是不可屏蔽的（即有中断状态就一定有中断）。另外，由 SLEEP_GOON_EN (SYS_CON0[30]) 决定用端口唤醒时是产生系统复位还是继续运行。

- 其它源唤醒

支持看门狗唤醒、比较器唤醒，RTC 唤醒、串口升级中断唤醒，LVD 唤醒，电量检测唤醒、多机通信串口唤醒。

其中串口多机通讯唤醒、比较器唤醒、电量检测唤醒与端口唤醒组 10 复用 (WKUP_CON[10]使能)。LVD 唤醒、RTC 唤醒与端口唤醒组 11 复用 (WKUP_CON[11]使能)。

珠海巨晟科技股份有限公司

10 复位和时钟控制 (RESET/CLOCK)

10.1 引脚复位(MCLR)功能

默认状态下 JS32L010 支持 MCLR 复位功能，PE2 为 MCLR 引脚，该引脚默认使能内部上拉功能，在 PE2 输入一个持续 1.7ms 以上的低电平导致系统复位，如同重新上电复位一样。用户可以通过修改 FLASH 中用户自定义位来关闭 MCLR 功能，关闭后 PE2 会变成 GPIO。详细情况请参考闪存存储器中关于用户自定义区域的描述。

10.2 复位

JS32L010 支持系统复位、电源复位和主复位。

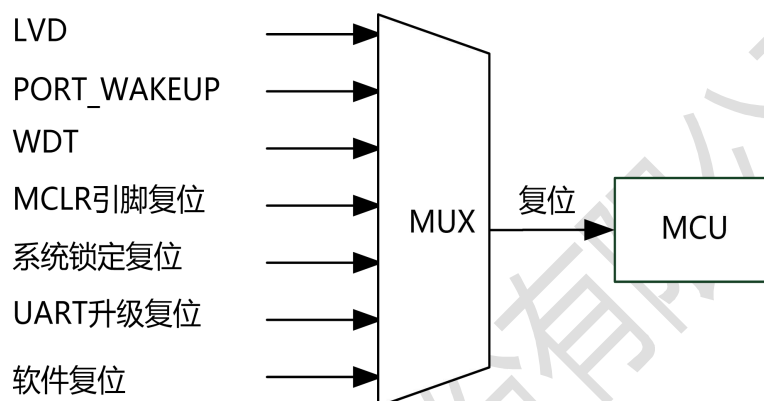


图 10-1 复位源框图

10.2.1 系统复位

系统复位将复位除某些复位状态寄存器和特殊功能寄存器之外的所有寄存器。

当以下事件中的一件发生时，产生一个系统复位：

- SLEEP 模式下外部 IO 口唤醒
- WDT 计数溢出复位
- UART 单线升级程序复位
- 系统锁定复位

10.2.2 主复位

主复位能将部分系统复位无法复位的寄存器复位。

以下事件可以触发一个主复位：

- 软件复位
- PVD 检测到电压低事件，且控制器处于复位功能模式
- 当芯片支持 MCLR(引脚)复位时，MCLR 指在某个特定 IO 引脚上输入一个持续 1.7ms 以上的低电平导致系统复位

10.2.3 电源复位

上电/掉电复位(POR/PDR 复位)都属于电源复位。电源复位将复位所有的逻辑和模拟模块。复位入口矢量被固定在地址 0x0000 0004。

10.3 时钟

10.3.1 模块框图

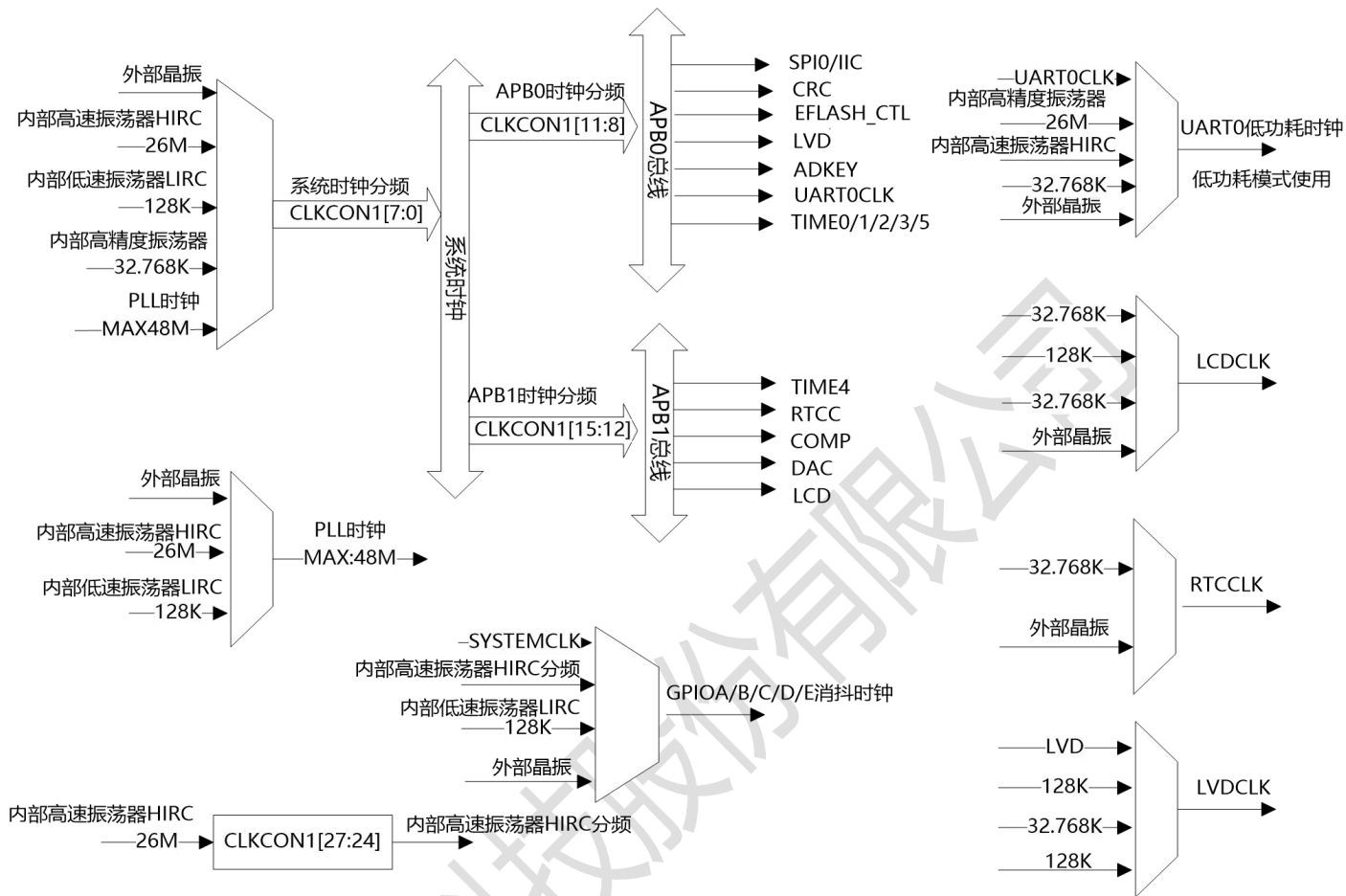


图 10-2 时钟模块

10.3.2 XOSC 时钟

外部时钟信号(32.768KHz)由以下两种时钟源产生：

- 外部晶体/陶瓷谐振器
- 用户外部时钟

10.3.3 HIRC 时钟

HIRC 时钟信号由内部 26MHz 的振荡器产生，可直接作为系统时钟或作为 PLL 输入。HIRC 振荡器能够在不需要任何外部器件的条件下提供系统时钟。HIRC 的启动时间比 XOSC 晶体振荡器短，需要频率校准，校准值写在 FLASH 系统存储区域。在程序使用这个时钟前，可以读取并配置出高精度的 HIRC 时钟。经过出厂效验后，正常温度范围下 HIRC 精度为 26MHz($\pm 1.5\%$)，具体请参考电气特性参数。

系统内置时钟安全系统，如果 XOSC 晶体振荡器失效，切换到高精度 32KHz。

10.3.4 PLL 时钟

内部 PLL 可以选择内部高精度 32KHz RC 振荡器、内部 26MHz、外部 32KHz 晶振作为参考时钟，支持小数分频。进一步信息请参考第二十章。

10.3.5 LIRC 时钟

LIRC 振荡器担当一个低功耗时钟源的角色，它作为系统启动时钟为其他单元提供时钟。LIRC 时钟频率大约 128KHz。

10.3.6 内部高精度 32K

高精度 32KHz 振荡器担当一个高精度时钟源的角色,它主要为 RTC,LCD 提供时钟。大约 32.768KHz,具体精度参考电气参数。

10.3.7 系统时钟 (SYSCLK)

五种不同的时钟源可被用来驱动系统时钟(SYSCLK):

- 内部低速 128KHz LIRC
- 内部高速 26MHz($\pm 1.5\%$) 高速振荡器
- 内部高精度 32.768KHz($\pm 1.5\%$) 低功耗低速振荡器
- 片内高速 PLL 时钟
- 支持外部 32.768KHz 晶体振荡器(内置电容)

10.3.8 毛刺滤波时钟源选择

三种不同的时钟源可被用来驱动 GPIO 的毛刺滤波时钟:

- 内部高速 26MHz HIRC 的分频时钟
- 系统时钟
- 内部低速 128KHz LIRC

当不被使用时,任一个时钟源都可被独立地启动或关闭,由此优化系统功耗。

10.3.9 时钟安全系统(CSS)

时钟安全系统可以通过软件被激活。在使用这个功能时,请确保 XOSC 已经被成功使能。如果 XOSC 时钟发生故障,产生时钟安全中断 CSSI,允许软件完成营救操作。此 CSSI 中断连接到 CPU 的 NMI 中断。

注:一旦 CSS 被激活,并且 XOSC 时钟出现故障,CSS 中断就产生,并且 NMI 也自动产生。NMI 将被不断执行,直到 CSS 中断挂起位被清除。因此,在 NMI 的处理程序中必须通过设置时钟中断寄存器, (HOSC_MNT) 里的 HOSC_LOSS_PENDING 位写 1 来清除 CSS 中断。通过寄存器使能,时钟故障将导致内部 XOSC 时钟自动切换到内部高精度 32KHz。

11 通用输入输出 (GPIO)

11.1 简介

每组 GPIO 端口有四个 32 位配置寄存器(GPIOx_MODER,GPIOx_OTYPER, GPIOx_OSPEEDR and GPIOx_PUPDR), 两个 32 位数据寄存器(GPIOx_IDAT and GPIOx_ODAT), 一个 32 位置位/复位寄存器 (GPIOx_BSR) 和一个 32 位翻转寄存器(GPIOx_TGL)。另外, 所有 GPIO 有一个 32 位锁定寄存器 (GPIOx_LCKR) 和两个复用功能选择寄存器 (GPIOx_AFRH and GPIOx_AFRL)。

注: GPIOx 中的 x 表示 GPIO 组数。LQFP48 封装, x 代表 A, B, C, E。LQFP64 封装, x 代表 A, B, C, D, E。

11.2 模块框图

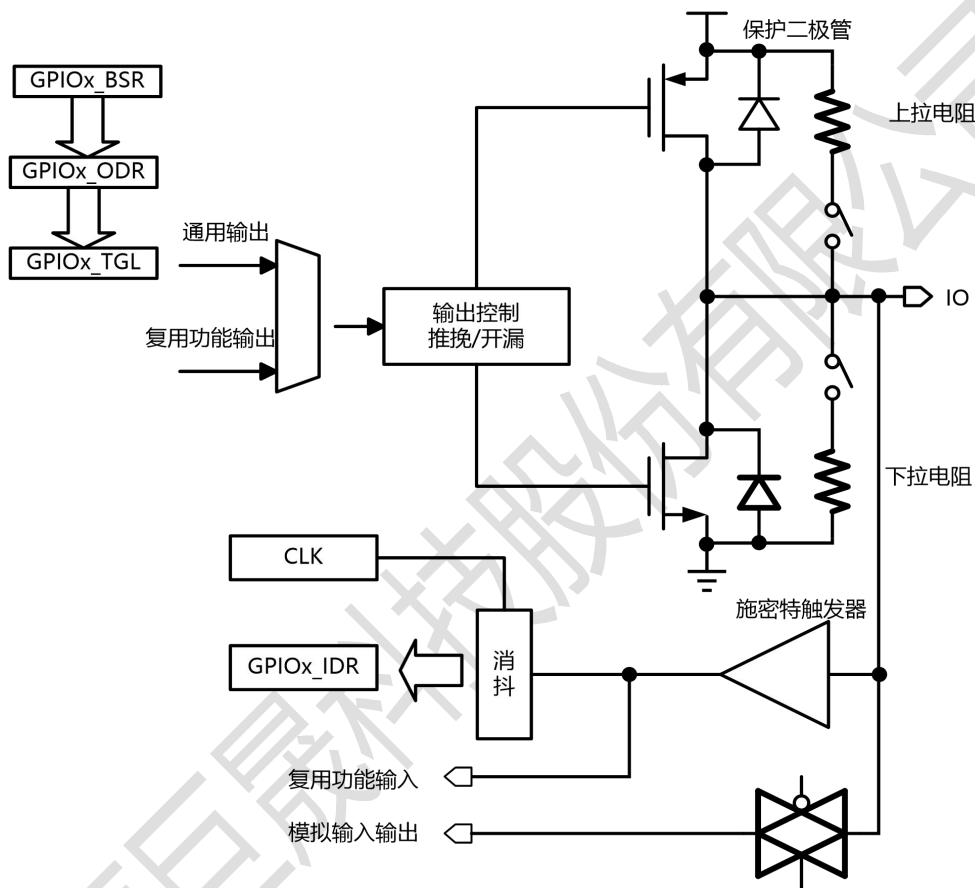


图 11-1 GPIO 模块框图

11.3 GPIO 主要特征

- 输出状态: 推挽或开漏+上下拉
- 输出寄存器状态值 (GPIOx_ODAT) 或者复用功能输出
- 输入状态: 浮空、上下拉、模拟
- 输入数据到数据寄存器(GPIOx_IDAT) 或复用功能输入
- 独立置位/复位/翻转 IO 状态 (GPIOx_BSR、GPIOx_TGL)
- 通过加锁配置 (GPIOx_LCKR) 锁定 IO 状态
- 模拟功能
- 复用功能

11.4 GPIO 功能描述

GPIO 的每一个端口可以通过软件独立配置成下面状态:

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟功能
- 开漏输出
- 推挽输出
- 复用功能

表 11-1 GPIO 端口位配置表

| MODE/模式 | OTYPE/输出类型 | OSPEED/驱动能力 | | PUPD/上下拉 | | IO configuration/端口配置 | |
|---------|------------|-------------|---|----------|---|-----------------------|--------|
| 01 | 0 | SPEED[3:0] | | 0 | 0 | GP output | PP |
| | 0 | | | 0 | 1 | Reserved | |
| | 0 | | | 1 | 0 | | |
| | 0 | | | 1 | 1 | | |
| | 1 | | | 0 | 0 | GP output | OD |
| | 1 | | | 0 | 1 | GP output | OD+PU |
| | 1 | | | 1 | 0 | Reserved | |
| | 1 | | | 1 | 1 | | |
| 10 | 0 | SPEED[3:0] | | 0 | 0 | AF | PP |
| | 0 | | | 0 | 1 | Reserved | |
| | 0 | | | 1 | 0 | | |
| | 0 | | | 1 | 1 | | |
| | 1 | | | 0 | 0 | AF | OD |
| | 1 | | | 0 | 1 | AF | OD+PU |
| | 1 | | | 1 | 0 | Reserved | |
| | 1 | | | 1 | 1 | | |
| 00 | x | x | x | 0 | 0 | Input | PP |
| | x | x | x | 0 | 1 | Input | PP+PU |
| | x | x | x | 1 | 0 | Input | PP+PD |
| | x | x | x | 1 | 1 | Reserved | |
| 11 | x | x | x | 0 | 0 | Input/Output | Analog |
| | x | x | x | 0 | 1 | Reserved | |
| | x | x | x | 1 | 0 | | |
| | x | x | x | 1 | 1 | | |

注：GP = Generate purpose (通用)，PP = Push pull(推挽)，PU = Pull up(上拉)，PD = Pull down(下拉)，OD = Open drain(开漏)，AF = Alternate function(复用)，SPEED = 驱动能力。

11.4.1 通用 IO (GPIO)

复位期间和刚复位后，复用功能未开启，I/O 端口被配置成浮空输入模式。

当作为输出配置时，写到输出数据寄存器上的值(GPIOx_ODAT)输出到相应的 I/O 引脚。可以以推挽或开漏模式使用输出驱动器。

输入数据寄存器(GPIOx_IDAT)在每个 APB 时钟周期捕捉 I/O 引脚上的数据。

所有 GPIO 引脚有一个内部弱上拉和弱下拉，当配置为输入时，它们可以被激活也可以被断开。

11.4.2 单独的位操作

当对 GPIOx_ODAT 的个别位编程时，软件不需要禁止中断：在单次 APB 写操作里，可以只更改一个或多个位。只需要通过对“置位/复位寄存器”(GPIOx_BSR)或“取反寄存器”(GPIOx_TGL)中想要更改的位写“1”来实现。没被选择的位将不被更改。

11.4.3 复用功能(AF)

芯片 IO 引脚通过多路选择器连接到片内外设，每个 IO 上同一时刻只能选通一个复用功能。每个 IO 引脚有一个 4 输入的多路选择器连接到复用功能(AF0~AF3)，通过配置 GPIOx_AFRH/L 选择输入功能。如果把端口配置成复用输出功能，则引脚和输出寄存器断开，并和片上外设的输出信号连接。如果软件把一个 GPIO 脚配置成复用输出功能，但是外设没有被激活，它的输出将不确定。

11.4.4 GPIO 锁定机制

锁定机制允许在 GPIO 控制寄存器 GPIOx_LCKR 上执行一串锁定程序，然后把 GPIO 的状态锁定，一旦 GPIO 状态被锁定，将不可改变，直到 CPU 复位。被锁定的寄存器有(GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR, GPIOx_AFRH and GPIOx_AFRH)。

锁定序列参考 GPIOx_LCKR 寄存器描述。

11.4.5 输入配置

当 I/O 端口配置为输入时：

- 输出缓冲器被禁止
- 施密特触发输入被激活
- 根据输入配置(上拉、下拉或浮空)的不同，弱上拉和下拉电阻被连接
- 出现在 I/O 脚上的数据在每个 APB 时钟被采样到输入数据寄存器
- 对输入数据寄存器的读访问可得到 I/O 状态

11.4.6 输出配置

当 I/O 端口被配置为输出时：

- 输出缓冲器被激活
 - 开漏模式：输出寄存器上的“0”激活 N-MOS，而输出寄存器上的“1”将端口置于高阻态(P-MOS 从不被激活)
 - 推挽模式：输出寄存器上的“0”激活 N-MOS，而输出寄存器上的“1”将激活 P-MOS。
- 施密特触发输入被激活
- 弱上拉和下拉电阻被禁止
- 出现在 I/O 脚上的数据在每个 APB 时钟被采样到输入数据寄存器
- 在开漏模式时，对输入数据寄存器的读访问可得到 I/O 状态
- 在推挽模式时，对输出数据寄存器的读访问得到最后一次写的值

11.4.7 模拟输入配置

当 I/O 端口被配置为模拟输入配置时：

- 输出缓冲器被禁止；
- 禁止施密特触发输入，实现了每个模拟 I/O 引脚上的零消耗。施密特触发输出值被强制为“0”；
- 弱上拉和下拉电阻被禁止；
- 读取输入数据寄存器时数值为“0”。

11.4.8 复用功能配置

对 I/O 端口进行编程作为复用功能时：

- 在开漏或推挽式配置中，输出缓冲器被打开
- 内置外设的信号驱动输出缓冲器（复用功能输出）
- 施密特触发输入被激活
- 弱上拉和下拉电阻被禁止
- 在每个 APB 时钟周期，出现在 I/O 脚上的数据被采样到输入数据寄存器
- 开漏模式时，读输入数据寄存器时可得到 I/O 口状态
- 在推挽模式时，读输出数据寄存器时可得到最后一次写的值

11.5 GPIO 寄存器描述

11.5.1 GPIO 端口模式寄存器(GPIOx_MODE)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|---------|-------|----------|--|
| 31:30 | MODER15 | 2'b0 | RW | Bits $2y+1:2y$ MODERy[1:0]: Port x configuration bits (y = 0..15) 这些位由软件编写以配置 I/O 模式 b00: 输入模式 (复位状态) b01: 通用输出模式 b10: 复用功能模式 b11: 模拟模式 |
| 29:28 | MODER14 | 2'b0 | RW | |
| 27:26 | MODER13 | 2'b0 | RW | |
| 25:24 | MODER12 | 2'b0 | RW | |
| 23:22 | MODER11 | 2'b0 | RW | |
| 21:20 | MODER10 | 2'b0 | RW | |
| 19:18 | MODER9 | 2'b0 | RW | |
| 17:16 | MODER8 | 2'b0 | RW | |
| 15:14 | MODER7 | 2'b0 | RW | |
| 13:12 | MODER6 | 2'b0 | RW | |
| 11:10 | MODER5 | 2'b0 | RW | |
| 9:8 | MODER4 | 2'b0 | RW | |
| 7:6 | MODER3 | 2'b0 | RW | |
| 5:4 | MODER2 | 2'b0 | RW | |
| 3:2 | MODER1 | 2'b0 | RW | |
| 1:0 | MODER0 | 2'b0 | RW | |

11.5.2 GPIO 端口输出类型寄存器(GPIOx_OTYPE)

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:16 | Reserved | - | - | - |
| 15 | OT15 | 1'b0 | RW | OTy: Port x configuration bits (y = 0..15) 这些位可以通过软件配置 I/O 输出类型 0: Output 推挽 (复位状态) 1: Output 开漏 |
| 14 | OT14 | 1'b0 | RW | |
| 13 | OT13 | 1'b0 | RW | |
| 12 | OT12 | 1'b0 | RW | |
| 11 | OT11 | 1'b0 | RW | |
| 10 | OT10 | 1'b0 | RW | |
| 9 | OT9 | 1'b0 | RW | |
| 8 | OT8 | 1'b0 | RW | |
| 7 | OT7 | 1'b0 | RW | |
| 6 | OT6 | 1'b0 | RW | |
| 5 | OT5 | 1'b0 | RW | |
| 4 | OT4 | 1'b0 | RW | |
| 3 | OT3 | 1'b0 | RW | |
| 2 | OT2 | 1'b0 | RW | |
| 1 | OT1 | 1'b0 | RW | |
| 0 | OT0 | 1'b0 | RW | |

11.5.3 GPIO 端口输出低位驱动寄存器(GPIOx_OSPEEDL)

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|---------|-------|----------|---|
| 31:28 | OSPEED7 | 4'b1 | RW | Bits 31:0 OSPEEDy[3: 0]: Port x configuration |

| | | | | |
|-------|---------|------|----|------------------------|
| 27:24 | OSPEED6 | 4'b1 | RW | bits (y = 0..7) |
| 23:20 | OSPEED5 | 4'b1 | RW | 这些位可以通过软件配置 I/O 驱动能力等级 |
| 19:16 | OSPEED4 | 4'b1 | RW | PC5 引脚: |
| 15:12 | OSPEED3 | 4'b1 | RW | bx000: low speed |
| 11:8 | OSPEED2 | 4'b1 | RW | bx001: speed1 |
| 7:4 | OSPEED1 | 4'b1 | RW | bx010: speed2 |
| 3:0 | OSPEED0 | 4'b1 | RW | bx011: speed3 |
| | | | | bx100: speed4 |
| | | | | bx101: speed5 |
| | | | | bx110: speed6 |
| | | | | bx111: speed7 |
| | | | | 其它: 保留 |
| | | | | 其他引脚: |
| | | | | bx00: low speed |
| | | | | bx01: speed1 |
| | | | | bx10: speed2 |
| | | | | bx11: speed3 |
| | | | | 其它: 保留 |

11.5.4 GPIO 端口输出高位驱动寄存器(GPIOx_OSPEEDH)

Address offset: 0x0C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:28 | OSPEED15 | 4'b1 | RW | Bits 31:0 OSPEEDy[3 : 0]: Port x configuration |
| 27:24 | OSPEED14 | 4'b1 | RW | bits (y = 8..15) |
| 23:20 | OSPEED13 | 4'b1 | RW | 这些位可以通过软件配置 I/O 驱动能力等级 |
| 19:16 | OSPEED12 | 4'b1 | RW | PC5 引脚: |
| 15:12 | OSPEED11 | 4'b1 | RW | bx000: low speed |
| 11:8 | OSPEED10 | 4'b1 | RW | bx001: speed1 |
| 7:4 | OSPEED9 | 4'b1 | RW | bx010: speed2 |
| 3:0 | OSPEED8 | 4'b1 | RW | bx011: speed3 |
| | | | | bx100: speed4 |
| | | | | bx101: speed5 |
| | | | | bx110: speed6 |
| | | | | bx111: speed7 |
| | | | | 其它: 保留 |
| | | | | 其他引脚: |
| | | | | bx00: low speed |
| | | | | bx01: speed1 |
| | | | | bx10: speed2 |
| | | | | bx11: speed3 |
| | | | | 其它: 保留 |

11.5.5 GPIO 端口上拉/下拉寄存器(GPIOx_PUPD)

Address offset: 0x10

| Width | Name | Reset | Property | Description |
|-------|------|-------|----------|---|
| 31 | PD15 | 1'b0 | RW | Bits PDy: Port x configuration bits (y = 16...31) |
| 30 | PD14 | 1'b0 | RW | 这些位可以通过软件配置 I/O 下拉 |
| 29 | PD13 | 1'b0 | RW | 0: 不下拉 |
| 28 | PD12 | 1'b0 | RW | 1: 下拉 |

| | | | | |
|----|------|------|----|---|
| 27 | PD11 | 1'b0 | RW | |
| 26 | PD10 | 1'b0 | RW | |
| 25 | PD9 | 1'b0 | RW | |
| 24 | PD8 | 1'b0 | RW | |
| 23 | PD7 | 1'b0 | RW | |
| 22 | PD6 | 1'b0 | RW | |
| 21 | PD5 | 1'b0 | RW | |
| 20 | PD4 | 1'b0 | RW | |
| 19 | PD3 | 1'b0 | RW | |
| 18 | PD2 | 1'b0 | RW | |
| 17 | PD1 | 1'b0 | RW | |
| 16 | PD0 | 1'b0 | RW | |
| 15 | PU15 | 1'b0 | RW | Bits PUy: Port x configuration bits (y = 0...15) 这些位可以通过软件配置 I/O 上拉 0: 不上拉 1: 上拉 Note: 此位在 GPIO 配置为输入模式、输出开漏模式且输出电平较高时才有效 |
| 14 | PU14 | 1'b0 | RW | |
| 13 | PU13 | 1'b0 | RW | |
| 12 | PU12 | 1'b0 | RW | |
| 11 | PU11 | 1'b0 | RW | |
| 10 | PU10 | 1'b0 | RW | |
| 9 | PU9 | 1'b0 | RW | |
| 8 | PU8 | 1'b0 | RW | |
| 7 | PU7 | 1'b0 | RW | |
| 6 | PU6 | 1'b0 | RW | |
| 5 | PU5 | 1'b0 | RW | |
| 4 | PU4 | 1'b0 | RW | |
| 3 | PU3 | 1'b0 | RW | |
| 2 | PU2 | 1'b0 | RW | |
| 1 | PU1 | 1'b0 | RW | |
| 0 | PU0 | 1'b0 | RW | |

11.5.6 GPIO 端口输入数据寄存器(GPIOx_IDAT)

Address offset: 0x14

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:16 | Reserved | - | - | - |
| 15 | IDAT15 | 1'b0 | RO | Bits 15:0 IDATy: 端口输入数据位(y = 0..15) 这些位只读. 它们包含相应 I/O 端口的输入值 |
| 14 | IDAT14 | 1'b0 | RO | |
| 13 | IDAT13 | 1'b0 | RO | |
| 12 | IDAT12 | 1'b0 | RO | |
| 11 | IDAT11 | 1'b0 | RO | |
| 10 | IDAT10 | 1'b0 | RO | |
| 9 | IDAT9 | 1'b0 | RO | |
| 8 | IDAT8 | 1'b0 | RO | |
| 7 | IDAT7 | 1'b0 | RO | |
| 6 | IDAT6 | 1'b0 | RO | |
| 5 | IDAT5 | 1'b0 | RO | |
| 4 | IDAT4 | 1'b0 | RO | |
| 3 | IDAT3 | 1'b0 | RO | |
| 2 | IDAT2 | 1'b0 | RO | |

| | | | | |
|---|-------|------|----|--|
| 1 | IDAT1 | 1'b0 | RO | |
| 0 | IDAT0 | 1'b0 | RO | |

11.5.7 GPIO 端口输出数据寄存器(GPIOx_ODAT)

Address offset: 0x18

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:16 | Reserved | - | - | - |
| 15 | ODAT15 | 1'b0 | RW | Bits 15:0 ODATy: 端口输出数据位 (y = 0..15) 这些位可读可写。 对于初始位置位和复位, ODAT 寄存器的这些位可以单独置位或者通过写入 GPIOx_BSRR 或 GPIOx_BRR 寄存器进行复位(x = A..F)。 |
| 14 | ODAT14 | 1'b0 | RW | |
| 13 | ODAT13 | 1'b0 | RW | |
| 12 | ODAT12 | 1'b0 | RW | |
| 11 | ODAT11 | 1'b0 | RW | |
| 10 | ODAT10 | 1'b0 | RW | |
| 9 | ODAT9 | 1'b0 | RW | |
| 8 | ODAT8 | 1'b0 | RW | |
| 7 | ODAT7 | 1'b0 | RW | |
| 6 | ODAT6 | 1'b0 | RW | |
| 5 | ODAT5 | 1'b0 | RW | |
| 4 | ODAT4 | 1'b0 | RW | |
| 3 | ODAT3 | 1'b0 | RW | |
| 2 | ODAT2 | 1'b0 | RW | |
| 1 | ODAT1 | 1'b0 | RW | |
| 0 | ODAT0 | 1'b0 | RW | |

11.5.8 GPIO 端口置位/复位寄存器(GPIOx_BSR)

Address offset: 0x1C

| Width | Name | Reset | Property | Description |
|-------|------|-------|----------|--|
| 31 | BR15 | 1'b0 | WO | Bits 31:16 BRy: Port x reset bit y (y = 0..15) 这些位只写。对这些位的读取将返回值 0x0000。 0: 对应的 ODATx 位没有动作 1: 复位相应的 ODATx 位 Note: 如果设置了 BSx 和 BRx, 则 BRx 具有优先级。 |
| 30 | BR14 | 1'b0 | WO | |
| 29 | BR13 | 1'b0 | WO | |
| 28 | BR12 | 1'b0 | WO | |
| 27 | BR11 | 1'b0 | WO | |
| 26 | BR10 | 1'b0 | WO | |
| 25 | BR9 | 1'b0 | WO | |
| 24 | BR8 | 1'b0 | WO | |
| 23 | BR7 | 1'b0 | WO | |
| 22 | BR6 | 1'b0 | WO | |
| 21 | BR5 | 1'b0 | WO | |
| 20 | BR4 | 1'b0 | WO | |
| 19 | BR3 | 1'b0 | WO | |
| 18 | BR2 | 1'b0 | WO | |
| 17 | BR1 | 1'b0 | WO | |
| 16 | BR0 | 1'b0 | WO | |
| 15 | BS15 | 1'b0 | WO | Bits 15:0 BSy: 端口 x 设置位 y (y = 0..15) 这些位只写。对这些位的读取将返回值 0x0000。 0: 对应的 ODATx 位没有动作 1: 置位相应的 ODATx 位 |
| 14 | BS14 | 1'b0 | WO | |
| 13 | BS13 | 1'b0 | WO | |
| 12 | BS12 | 1'b0 | WO | |
| 11 | BS11 | 1'b0 | WO | |

| | | | |
|----|------|------|----|
| 10 | BS10 | 1'b0 | WO |
| 9 | BS9 | 1'b0 | WO |
| 8 | BS8 | 1'b0 | WO |
| 7 | BS7 | 1'b0 | WO |
| 6 | BS6 | 1'b0 | WO |
| 5 | BS5 | 1'b0 | WO |
| 4 | BS4 | 1'b0 | WO |
| 3 | BS3 | 1'b0 | WO |
| 2 | BS2 | 1'b0 | WO |
| 1 | BS1 | 1'b0 | WO |
| 0 | BS0 | 1'b0 | WO |

11.5.9 GPIO 端口配置锁定寄存器(GPIOx_LCK)

当正确的写入序列应用于第 16 位(LCKK)时, 此寄存器用于锁定端口位的配置。寄存器位 [15:0]用于锁定 GPIO 的配置。在写入序列期间, LCKR[15:0]的值不能更改。当锁序列被应用在一个端口位上时, 该端口位的值在下次 MCU 复位或外设复位之前不能再更改。

Address offset: 0x20

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:17 | Reserved | - | - | - |
| 16 | LCKK | 1'b0 | RW | Bit 16 LCKK: 锁键 可在任何时候读取, 只能用锁定键写入序列修改它 0: 端口配置锁定键不活动 1: 端口配置锁定键激活。GPIOx_LCK 寄存器被锁定, 直到下一次 MCU 复位或外围设备复位。 LOCK key write sequence: WR LCK [16] = '1' + LCK [15:0] WR LCK [16] = '0' + LCK [15:0] WR LCK [16] = '1' + LCK [15:0] RD LCKR RD LCK [16] = '1' (读取可选, 但确认锁是激活) Note: 在锁键写入序列时, LCK[15:0]的值不改变。 锁定序列中的任何错误都会中止锁定 第一次锁后序列在任何端口,任何读访问 LCKK 的位将返回'1',直到下一个单片机复位或外围重置。 |
| 15 | LCK15 | 1'b0 | RW | Bits 15:0 LCKy: 端口 x 引脚 y lock (y= 0..15) 这些位是读/写的, 但只能在 LCKK 位为 0 写入。 0: 端口配置未锁定 1: 端口配置锁定 |
| 14 | LCK14 | 1'b0 | RW | |
| 13 | LCK13 | 1'b0 | RW | |
| 12 | LCK12 | 1'b0 | RW | |
| 11 | LCK11 | 1'b0 | RW | |
| 10 | LCK10 | 1'b0 | RW | |
| 9 | LCK9 | 1'b0 | RW | |
| 8 | LCK8 | 1'b0 | RW | |
| 7 | LCK7 | 1'b0 | RW | |
| 6 | LCK6 | 1'b0 | RW | |
| 5 | LCK5 | 1'b0 | RW | |
| 4 | LCK4 | 1'b0 | RW | |
| 3 | LCK3 | 1'b0 | RW | |
| 2 | LCK2 | 1'b0 | RW | |

| | | | | |
|---|------|------|----|--|
| 1 | LCK1 | 1'b0 | RW | |
| 0 | LCK0 | 1'b0 | RW | |

11.5.10 GPIO 复用功能低位寄存器(GPIOx_AFRL)

Address offset: 0x24

| Width | Name | Reset | Property | Description |
|-------|------|-------|----------|--|
| 31:28 | AFR7 | 4'b0 | RW | Bits 31:0 AFRy[3:0]: 端口 x 引脚 y 复用功能设置位 y (y = 0...7) 这些位通过软件写入用于配置 I/O 口复用功能 AFSELy selection: 4'b0000: AF0 4'b0001: AF1 4'b0010: AF2 4'b0011: AF3 其它: 保留 |
| 27:24 | AFR6 | 4'b0 | RW | |
| 23:20 | AFR5 | 4'b0 | RW | |
| 19:16 | AFR4 | 4'b0 | RW | |
| 15:12 | AFR3 | 4'b0 | RW | |
| 11:8 | AFR2 | 4'b0 | RW | |
| 7:4 | AFR1 | 4'b0 | RW | |
| 3:0 | AFR0 | 4'b0 | RW | |

11.5.11 GPIO 复用功能高位寄存器(GPIOx_AFRH)

Address offset: 0x28

| Width | Name | Reset | Property | Description |
|-------|-------|-------|----------|---|
| 31:28 | AFR15 | 4'b0 | RW | Bits 31:0 AFRy[3:0]: 端口 x 引脚 y 复用功能设置位 y (y = 8...15) 这些位通过软件写入用于配置 I/O 口复用功能 AFSELy selection: 4'b0000: AF0 4'b0001: AF1 4'b0010: AF2 4'b0011: AF3 其它: 保留 |
| 27:24 | AFR14 | 4'b0 | RW | |
| 23:20 | AFR13 | 4'b0 | RW | |
| 19:16 | AFR12 | 4'b0 | RW | |
| 15:12 | AFR11 | 4'b0 | RW | |
| 11:8 | AFR10 | 4'b0 | RW | |
| 7:4 | AFR9 | 4'b0 | RW | |
| 3:0 | AFR8 | 4'b0 | RW | |

11.5.12 GPIO 端口位切换寄存器(GPIOx_TGL)

Address offset: 0x2C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:16 | Reserved | - | - | - |
| 15 | TG15 | 1'b0 | WO | Bits 15:0 TGy: 端口 x 引脚 y 切换 (y = 0..15) 这些位是只写的。对这些位的读取将返回值 0x0000 0: 对应的 ODATx 位没有动作 1: 翻转相应的 ODATx 位的输出电平 Note: 如果设置了 BSx BRx 和 TGx, 则 BRx 具有第一优先级, BSx 具有第二优先级。 |
| 14 | TG14 | 1'b0 | WO | |
| 13 | TG13 | 1'b0 | WO | |
| 12 | TG12 | 1'b0 | WO | |
| 11 | TG11 | 1'b0 | WO | |
| 10 | TG10 | 1'b0 | WO | |
| 9 | TG9 | 1'b0 | WO | |
| 8 | TG8 | 1'b0 | WO | |
| 7 | TG7 | 1'b0 | WO | |
| 6 | TG6 | 1'b0 | WO | |
| 5 | TG5 | 1'b0 | WO | |
| 4 | TG4 | 1'b0 | WO | |
| 3 | TG3 | 1'b0 | WO | |
| 2 | TG2 | 1'b0 | WO | |
| 1 | TG1 | 1'b0 | WO | |

| | | | | |
|---|-----|------|----|--|
| 0 | TG0 | 1'b0 | WO | |
|---|-----|------|----|--|

11.5.13 GPIO 端口中断屏蔽寄存器(GPIOx_IMK)

Address offset: 0x30

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:16 | Reserved | - | - | - |
| 15 | IMK15 | 1'b0 | RW | Bits 15:0 IMKy: 端口 x 引脚 y 中断配置 (y = 0..15). 如果 IMKy 置 1, IO 输入电平翻转时会出现中断 0: 禁止中断 1: 使能中断 |
| 14 | IMK14 | 1'b0 | RW | |
| 13 | IMK13 | 1'b0 | RW | |
| 12 | IMK12 | 1'b0 | RW | |
| 11 | IMK11 | 1'b0 | RW | |
| 10 | IMK10 | 1'b0 | RW | |
| 9 | IMK9 | 1'b0 | RW | |
| 8 | IMK8 | 1'b0 | RW | |
| 7 | IMK7 | 1'b0 | RW | |
| 6 | IMK6 | 1'b0 | RW | |
| 5 | IMK5 | 1'b0 | RW | |
| 4 | IMK4 | 1'b0 | RW | |
| 3 | IMK3 | 1'b0 | RW | |
| 2 | IMK2 | 1'b0 | RW | |
| 1 | IMK1 | 1'b0 | RW | |
| 0 | IMK0 | 1'b0 | RW | |

11.5.14 GPIO 中断事件挂起寄存器(GPIOx_TGPEND)

Address offset: 0x34

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:16 | Reserved | - | - | - |
| 15 | TGPEND15 | 1'b0 | RW | |
| 14 | TGPEND14 | 1'b0 | RW | |
| 13 | TGPEND13 | 1'b0 | RW | |
| 12 | TGPEND12 | 1'b0 | RW | |
| 11 | TGPEND11 | 1'b0 | RW | |
| 10 | TGPEND10 | 1'b0 | RW | |
| 9 | TGPEND9 | 1'b0 | RW | |
| 8 | TGPEND8 | 1'b0 | RW | |
| 7 | TGPEND7 | 1'b0 | RW | |
| 6 | TGPEND6 | 1'b0 | RW | |
| 5 | TGPEND5 | 1'b0 | RW | |
| 4 | TGPEND4 | 1'b0 | RW | |
| 3 | TGPEND3 | 1'b0 | RW | |
| 2 | TGPEND2 | 1'b0 | RW | |
| 1 | TGPEND1 | 1'b0 | RW | |
| 0 | TGPEND0 | 1'b0 | RW | Bits 15:0 TGPENDy: 端口 x 位 y 中断挂起 (y = 0..15). 当 IO 输入切换时, PENDy 将被置 1 0: IO 口没有切换 1: IO 口电平切换 软件写 1 清除 TGPEND |

12 通信接口外设 (CSI)

12.1 串行外设接口和内部集成电路(SPI_IIC)

12.1.1 SPI 简介

SPI 接口广泛用于不同设备之间的板级通讯，如扩展串行 Flash，ADC 等。

12.1.2 SPI 模块框图

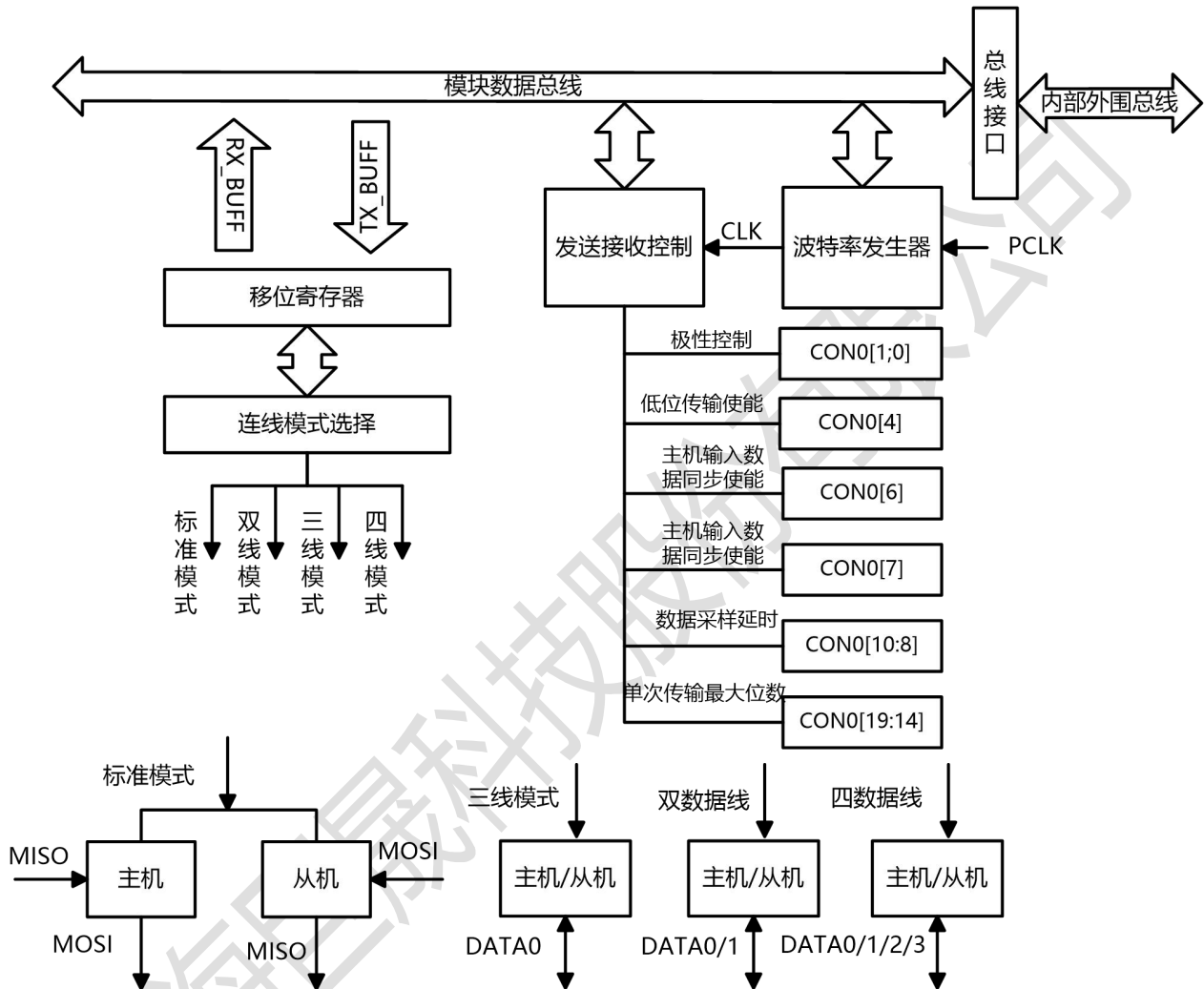


图 12-1 SPI 模块框图

12.1.3 SPI 特性

- 支持主模式和从模式工作
- 可编程时钟极性，支持 4 种模式
 - 模式 0：时钟空闲为 0，上升沿采样，下降沿出数据
 - 模式 1：时钟空闲为 0，下降沿采样，上升沿出数据
 - 模式 2：时钟空闲为 1，下降沿采样，上升沿出数据
 - 模式 3：时钟空闲为 1，上升沿采样，下降沿出数据
- 支持 8Bit、16Bit、24Bit and 32Bit 传输
- 传输数据顺序 MSB 和 LSB
- 支持标准模式，三线模式，两数据线模式和四数据线模式
 - 标准模式：通信线有 CLK, CS, IO0(MOSI), IO1(MISO), 一个 CLK 传输 1Bit 数据
 - 三线模式：通信线有 CLK, CS, IO0, 接收和发送都通过 IO0, 一个 CLK 传输 1Bit 数据

- 两数据模式：通信线 CLK,CS,IO0,IO1，接收和发送都通过 IO0 和 IO1,一个 CLK 传输 2Bit 数据
- 四数据模式：数据线 CLK,CS,IO0,IO1,IO2,IO3，接收和发送都通过 IO0,IO1,IO2,IO3, 一个 CLK 传输 4Bit 数据
- 可触发中断的专用发送和接收标志
- 支持 DMA 功能的发送接收缓冲器：产生发送和接收的请求
- 支持 8 位、16 位、24 位、32 位传输帧格式
- SPI 总线忙状态标志

12.1.4 IIC 简介

IIC(Inter - Integrated Circuit)总线是由 PHILIPS 公司开发的两线式串行总线，用于连接微控制器及其外围设备。它提供多主模式功能，可以控制所有 IIC 总线特定的序列、协议、仲裁和时序。是微电子通信控制领域广泛采用的一种总线标准。

该模块默认工作在主模式。

12.1.5 模块框图

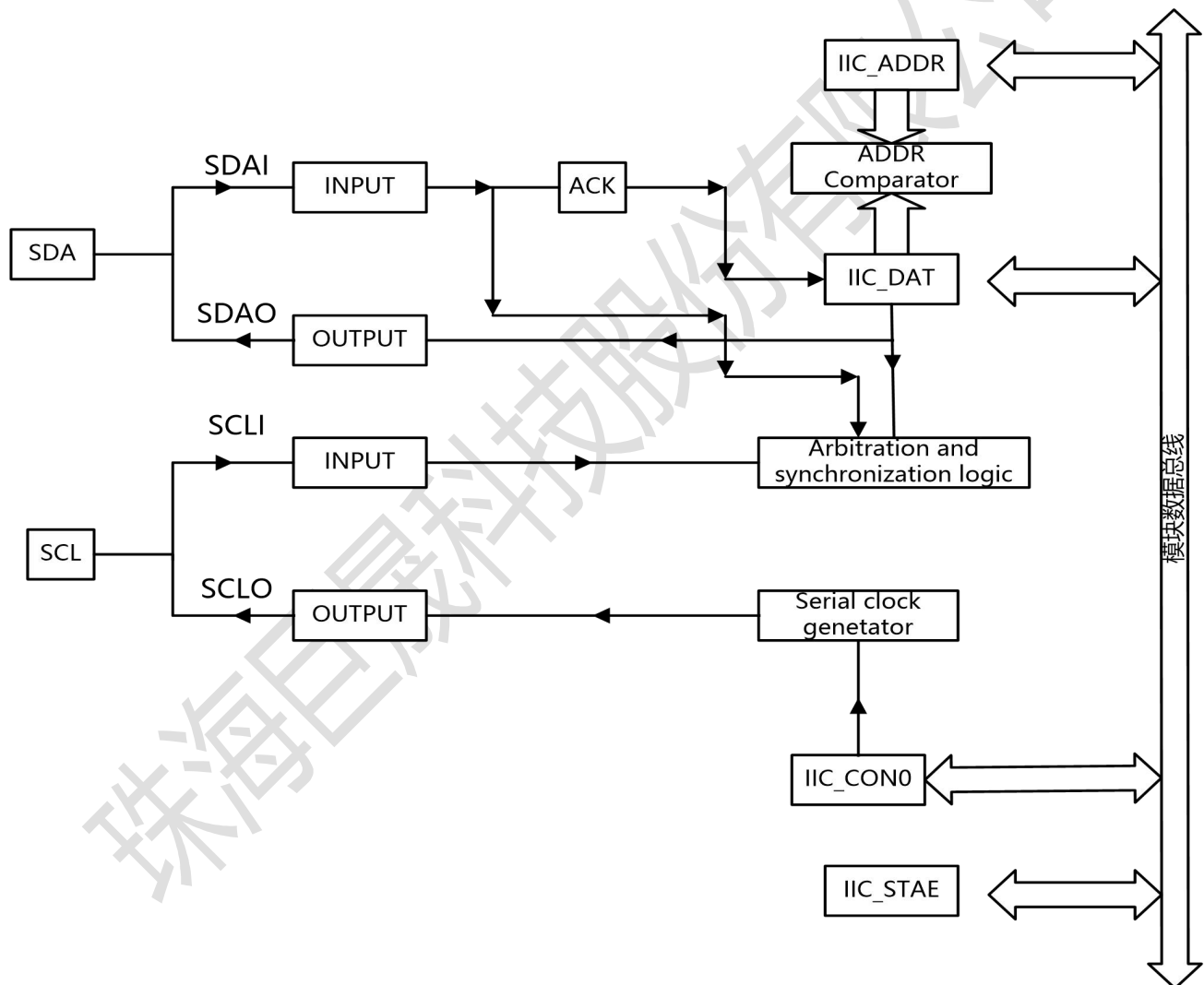


图 12-2 IIC 模块图

12.1.6 IIC 特性

- 支持主模式和从模式
- 主模式支持时钟同步和仲裁
- 从模式支持在发送数据没有准备好或者接收缓冲器满时候拉低 SCL
- 从模式支持 7bit 地址或者 10bit 地址
- 主模式自动生成起始信号和停止信号

- 从模式停止位检测
- 总线忙标志

12.1.7 SPI 时序图

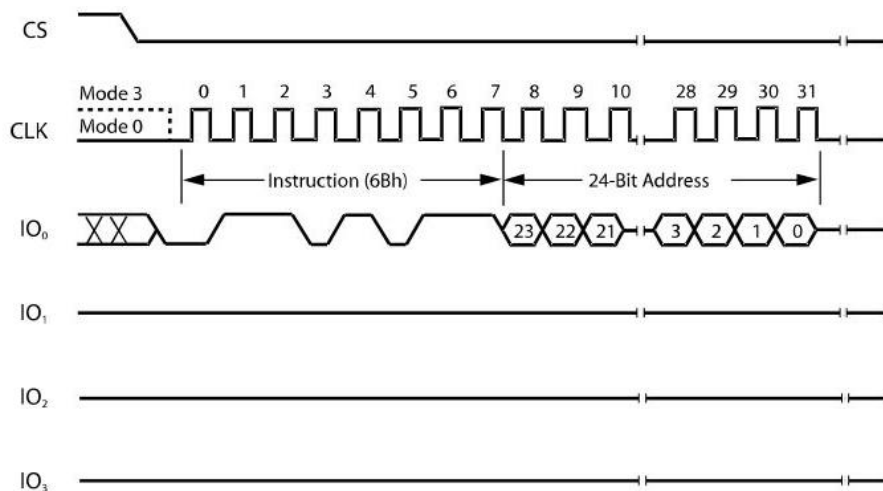
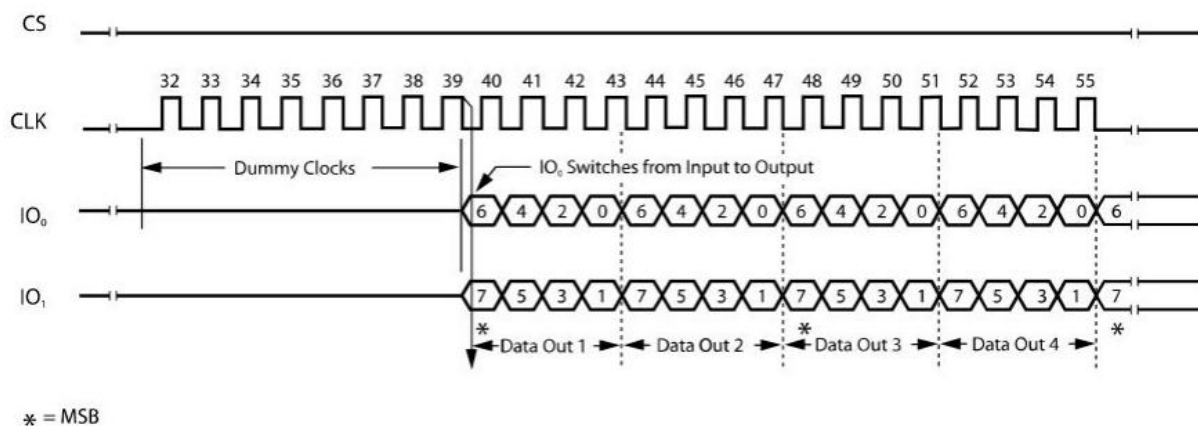


图 12-3 三线模式 SPI



* = MSB

图 12-4 两线模式 SPI

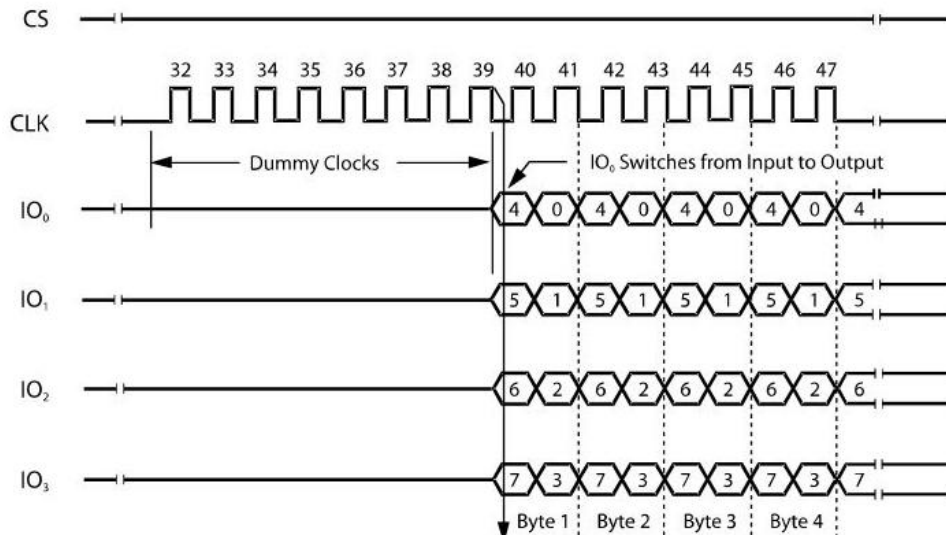


图 12-5 四线模式 SPI

12.1.8 IO 映射

IO 映射参考管脚定义。

12.2 SPI/IIC 寄存器描述

12.2.1 CON0(SPI 接口)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|----------------|-------|----------|--|
| 31:20 | Reserved | - | - | - |
| 19:14 | SPI_FRAME_SIZE | 6'h0 | RW | SPI frame size 0: 非法 h01: 一次传输 1 位数据 h02: 一次传输 2 位数据 h03: 一次传输 3 位数据 ... h1F: 一次传输 31 位数据 h20: 一次传输 32 位数据 h21~h3F: 非法 注意:当传输大小小于 h20 时,有效数据将保存在 32 位的低位。 例如, 如果 SPI_FRAME_SIZE 为 8, 那么 SSP_CMD_DATA 的低 8 位就是有效数据。 |
| 13 | NSS_POS_IE | 1'b0 | RW | SPI 从机模式 SPI_NSS 引脚上升沿中断使能 0: 不使能 1: 使能 |
| 12 | SPI_NSS_EN | 1'b0 | RW | SPI_NSS 引脚使能, 此位仅在 SPI 从模式下有效 0: 不使能, SPI 总线没有 NSS 引脚 1: 使能, SPI 总线有 NSS 引脚 |
| 11 | SPI_NSS | 1'b0 | RW | SPI NSS 引脚控制输出, 此位仅在 SPI 主模式下有效 0: 输出低电平 1: 输出高电平 |
| 10: 8 | RXSEL | 3'b0 | RW | SPI 主机模式, 捕捉数据延时 |

| | | | | |
|------|----------------|------|----|---|
| | | | | b000: 没有延时 b001: 延时一个周期 b010: 延时两个周期 ... b111: 延时七个周期 |
| 7 | SLAVE_SYNC_EN | 1'b0 | RW | SPI SLAVE 模式下, 输入数据是否需要同步的选择位 0: 不使能 1: 使能 |
| 6 | MASTER_SYNC_EN | 1'b0 | RW | SPI 主模式下, 输入数据是否需要同步的选择位 0: 不使能 1: 使能 |
| 5 | Reserved | - | - | - |
| 4 | LSBFE | 1'b0 | RW | SPI 从低位传输使能 0: 先传输高位 1: 先传输低位 |
| 3: 2 | WIREMODE | 2'b0 | RW | SPI 连线模式: b00: 标准模式 b01: 三线模式 b10: 双线模式 b11: 四线模式 |
| 1: 0 | SPI MODE | 2'b0 | RW | SPI 极性: b00: Mode 0: 时钟空闲为 0, 上升沿采样, 下降沿出数据 b01: Mode 1: 时钟空闲为 0, 下降沿采样, 上升沿出数据 b10: Mode 2: 时钟空闲为 1, 下降沿采样, 上升沿出数据 b11: Mode 3: 时钟空闲为 1, 上升沿采样, 下降沿出数据 |

12.2.2 CON0(IIC 接口)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|----------------|-------|----------|--|
| 31:23 | Reserved | - | - | - |
| 22 | RX_NACK_IE | 1'b0 | RW | 接收到 NACK 中断使能 0: 不使能 1: 使能 |
| 21 | IIC_AL_IE | 1'b0 | RW | 主机仲裁丢失中断使能 0: 不使能 1: 使能 |
| 20 | STOP_IE | 1'b0 | RW | 检测到线上有 STOP 信号中断使能, 不管是从机主机都会起来 0: 不使能 1: 使能 |
| 19 | ADR_MATCH_IE | 1'b0 | RW | SLAVE 地址匹配中断使能 0: 不使能 1: 使能 |
| 18:14 | IIC_FILTER_CNT | 5'h0 | RW | IIC 每(IIC_FILTER_CNT + 1)个 IIC 模块时钟, 采样一次 SCL 和 SDA, 用于滤除一定宽度的线路毛刺。IIC_FILTER_CNT 配置越大滤掉的毛刺宽度越大。 |
| 13 | BROADCAST_IE | 1'b0 | RW | 广播中断使能 0: 不使能 1: 使能 |

| | | | | |
|-------|-----------------|-------|----|---|
| 12 | BROADCAST_EN | 1'b0 | RW | IIC 作为 slave 时候, 是否使能接收广播地址 0: 忽略广播地址 1: 接收到广播地址时, 回应 ACK, 并且将 BROADCAST_PEND 置起来, 可以产生中断 |
| 11: 2 | SLAEV_ADR | 10'h0 | RW | IIC 作为 SLAVE 时候, SLAVE 地址 |
| 1 | TX_NACK | 0 | RW | IIC 在接收数据的时候, 回应 NACK 还是 ACK 选择位 0: ACK 1: NACK |
| 0 | SLAVE_ADR_WIDTH | 0 | RW | IIC 作为 SLAVE 时候, SLAVE 地址宽度 0: 7bit 1: 10bit |

12.2.3 CON1

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|----------------|-------|----------|--|
| 31:10 | Reserved | - | - | - |
| 9 | DMA_IE | 1'b0 | RW | DMA 完成中断使能 0: 不使能 1: 使能 |
| 8 | BUF_OV_IE | 1'b0 | RW | 接收 buffer 溢出中断使能 0: 不使能 1: 使能 |
| 7 | RBUF_NEMPTY_IE | 1'b0 | RW | 接收 buffer 不空中断使能 0: 不使能 1: 使能 |
| 6 | TBUF_NFULL_IE | 1'b0 | RW | 发送 buffer 不满中断使能 0: 不使能 1: 使能 |
| 5 | SSP_IE | 1'b0 | RW | IIC 接口: IIC master 接收或者发送完成一帧数据 (START + WRITE/READ(8bit+ack) + STOP)中断使能 IIC slave 接收或者发送完成一帧数据(8bit+ack)中断使能; 0: 不使能 1: 使能 |
| 4 | DMA_EN | 1'b0 | RW | 发送 DMA 使能 0: 不使能 1: 使能 注: 如果 SSP_TX_RX_EN==0, 表示 DMA 接收; 如果 SSP_TX_RX_EN==1, 表示 DMA 发送; |
| 3 | SSP_TX_RX_EN | 1'b0 | RW | 接口发送使能 0: RX_EN 1: TX_EN |
| 2 | SLAVE | 1'b0 | RW | SLAVE mode 0: 主模式 1: 从模式 |
| 1 | SPI_IIC_SEL | 1'b0 | RW | SPI 和 IIC 选择位 0: SPI 接口 1: IIC 接口 |

| | | | | |
|---|--------|------|----|-------------------------|
| 0 | SSP_EN | 1'b0 | RW | 模块使能 0: 不使能 1: 使能 |
|---|--------|------|----|-------------------------|

12.2.4 SSP_CMD_DATA

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|--------------|-------|----------|---|
| 31:0 | SSP_CMD_DATA | 32'h0 | RW | <p>SPI 接口: 写: 将想要发送的数据写入这个寄存器, 触发 SPI 发送 读: 读这个寄存器, 获取收到的数据</p> <p>IIC 接口: 写: [7:0] 将想要发送的数据写入这 8bit; [8] START 使能位, 在将这 byte 数据发出去前, 先插入一个 start bit(只在 IIC_MASTER 模式有效)。 [9] STOP 使能位, 在将这 byte 数据发出去后, 紧跟一个 stop bit(只在 IIC_MASTER 模式有效)。 读: 读这个寄存器, 获取收到的数据(bit7~bit0) [31:10] 没有使用</p> |

12.2.5 BAUD

Address offset: 0x0C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:16 | Reserved | - | - | - |
| 15:0 | BAUD | 16'h0 | RW | <p>MASTER 模式时候: SPI 波特率=apb0_clock/(2*(BAUD+1)) IIC 波特率= apb0_clock/(4*(BAUD+1))</p> <p>SLAVE 模式时候: SPI 没有用 IIC 用于定义从机准备好发送数据, 再 delay(BAUD+1)个 apb0_clock 周期后才释放 SCL。</p> |

12.2.6 DMA_LEN

Address offset: 0x10

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|--|
| 31:12 | Reserved | - | - | - |
| 11:0 | SPI_DMA_LEN | 12'h0 | RW | 接收和发送 DMA 模式: 计划 DMA 的数据 byte 长度。 在 master 模式, RX_DMA_EN 为 0 时候, 接收数据: 表示计划接收的数据 byte 长度。 |

12.2.7 DMA_CNT

Address offset: 0x14

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|--|
| 31:12 | Reserved | - | - | - |
| 11:0 | SPI_DMA_CNT | 12'h0 | RO | 接收和发送 DMA 模式: 实际接收并且保存到 SRAM 的数据 byte 长度。 在 master 模式, RX_DMA_EN 为 0 时候, 接收数据: 表示实际接收并且已经被 CPU 读走的数据 byte 长度。 |

| | | | | |
|--|--|--|--|--|
| | | | | 注：在 RX_DMA_EN 上升沿，TX_DMA_EN 上升沿，以及 MASTER 读上升沿的时候，被清零。 |
|--|--|--|--|--|

12.2.8 DMA_STADR

Address offset: 0x18

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|--|
| 31:13 | Reserved | - | - | - |
| 12:0 | SPI_DMA_STADR | 13'h0 | RW | DMA 地址，一帧数据是 32bit 时候，需要 32bit 对齐，其他时候可以任意配置 |

12.2.9 SPI_STA

Address offset: 0x1C

| Width | Name | Reset | Property | Description | | | |
|-------|----------------|-------|----------|---|-------------|-------------------------|--|
| 31:28 | Reserved | - | - | - | | | |
| 27 | SLV_ADDRED | 1'b0 | RO | 从机模式，从机被寻址标志位 0: 从机没有被寻址 1: 从机已经被寻址 | | | |
| 26:24 | STATE | 3'b0 | RO | IIC 状态 | 主 | 从 | |
| | | | | b000: IDLE | 空闲 | 空闲 | |
| | | | | b001: START | 发送 start | 已经接收到 start, 等待 SCL 变 0 | |
| | | | | b010: TX | 发送 1byte 数据 | 发送 1byte 数据 | |
| | | | | b011: RX | 接收 1byte 数据 | 接收 1byte 数据 | |
| | | | | b100: STOP | 发送 stop | 保留 | |
| | | | | b101: ADR0 | 保留 | 等待接收 1byte 地址 | |
| | | | | b110: ADR1 | 保留 | 等待接收 2byte 地址 | |
| | | | | b111: 保留 | 保留 | 保留 | |
| 23:20 | Reserved | - | - | - | | | |
| 19 | CLR_BUF_CNT | 1'b0 | WO | 写 0: 没有任何作用 读 1: 缓冲计数清零 | | | |
| 18:16 | BUF_CNT | 3'b0 | RO | Buffer 里面有多少 byte 有效数据，CPU 每读(RX_EN)/写(TX_EN)一次 CMD_DATA 寄存器，减/加一个帧数据宽度，8bit 数据减 1，16 位数据减 2，24 位和 32 位数据减 4。 | | | |
| 15 | MASTER_RX_BUSY | 1'b0 | RO | 在 MASTER 读从机模式，指示软件配置的连续读取 DMA_LEN byte 数据是否完成 0: 已经读完 1: 还没有读完 | | | |
| 14 | IIC_RX_NACK | 1'b0 | RO | IIC 主机或者从机，在发送数据的时候，在第 9bit 握手阶段，检测到 ACK 还是 NACK 0: ACK 1: NACK | | | |
| 13 | IIC_SLAVE_RW | 1'b0 | RO | IIC 从机，在地址阶段，接收到的主机发过来的读写标志 0: 主机将写从机 | | | |

| | | | | |
|----|--------------------|------|----|--|
| | | | | 1: 主机将读从机 |
| 12 | IIC_BUS_BUSY | 1'b0 | RO | IIC 线路是否 Busy 指示位 0: 线路上一直没有出现 START, 或者出现 START 后又出现了 STOP, 线路空闲。 1: 检测到线路上出现了 START, 并且一直没有出现 STOP, 线路忙。 注: 只能硬件清除 |
| 11 | SPI_SLAVE_CS | 1'b1 | RO | SPI SLAVE, CS 的状态 |
| 10 | SSP_BUSY | 1'b0 | RO | SPI 或者 IIC 正忙 0: MASTER 空闲 SLAVE 没有在发送数据, 接收的时候不使用 1: MASTER 正在接收或者发送一帧数据 SLAVE 正在发送一帧数据, 接收的时候不使用 注: 硬件清, 或者关掉 SPI 和 IIC(CON1[0]=0), 或者在 SLAVE 时候, CLR_BUF_CNT 会同时清掉 SSP_BUSY。 |
| 9 | AL_PEND | 1'b0 | RW | IIC 主机, 检测到仲裁丢失 0: 没有仲裁丢失 1: 仲裁丢失 注: 只能软件清, 必须清掉 IIC 才能正常工作 |
| 8 | STOP_PEND | 1'b0 | RW | IIC 接口, 检测到线路上产生了 STOP 位 0: 没有检测到 STOP 位。 1: 检测到 STOP 位。 注: 只能软件清 |
| 7 | ADR_MTCH_PEND | 1'b0 | RW | IIC 从机, 接收到主机发送过来的对的从机地址 0: 从机地址不匹配 1: 从机地址匹配 注: 只能软件清 |
| 6 | IIC_BROADCAST_PEND | 1'b0 | RW | IIC 作为 SLAVE 时候, 检测到广播地址标志位 0: 没有检测到广播地址 1: 检测到广播地址 注: 只能软件清 |
| 5 | SPI_NSS_POS | 1'b0 | RW | 检测到 SPI NSS 引脚上升沿 0: 没有检测到上升沿 1: 检测到上升沿 注: 只能软件清 |
| 4 | DMA_PEND | 1'b0 | RW | 发送 DMA 或者接收 DMA 完成标志 0: DMA 没有完成 1: DMA 已经完成 注: 只能软件清 |
| 3 | BUF_OV | 1'b0 | RW | Buffer 溢出, 有数据丢失了 0: buffer 没有溢出 1: 在 buffer 已经是满状态(Buffer 容量是 5byte, 满并不一定是 5byte, 而是空间容不下一帧数据了), 又有数据接收到。丢弃后来的数据。 注: 只能软件清 |
| 2 | BUF_EMPTY | 1'b1 | RO | Buffer 空标志 0: 不空 1: 空 |

| | | | | |
|---|----------|------|----|--|
| 1 | BUF_FULL | 1'b0 | RO | Buffer 满标志位 0: 不满 1: 满 |
| 0 | SSP_DONE | 1'b0 | RW | 0: 没有完成 1: SPI 接口 SPI 已经完成一帧(8bit/16bit/24bit/32bit)数据的接收或者发送 IIC 接口。 IIC MASTER 已经完成了一帧数据的接收或者发送(START(可选) + 写/读(8 位+ack) + STOP(可选))。 IIC SLAVE 已经完成了一帧数据的接收或者发送(8bit+ack)。 注: 只能软件清 |

珠海巨晟科技股份有限公司

13 通用异步收发器 (UART)

13.1 简介

通用异步收发器(UART)提供了一种灵活的方法与使用工业标准 NRZ 异步串行数据格式的外部设备之间进行全双工数据交换。UART 利用分数波特率发生器提供宽度范围的波特率选择。使用多缓冲器配置的 DMA 方式，可以实现高速数据通信。

13.2 模块框图

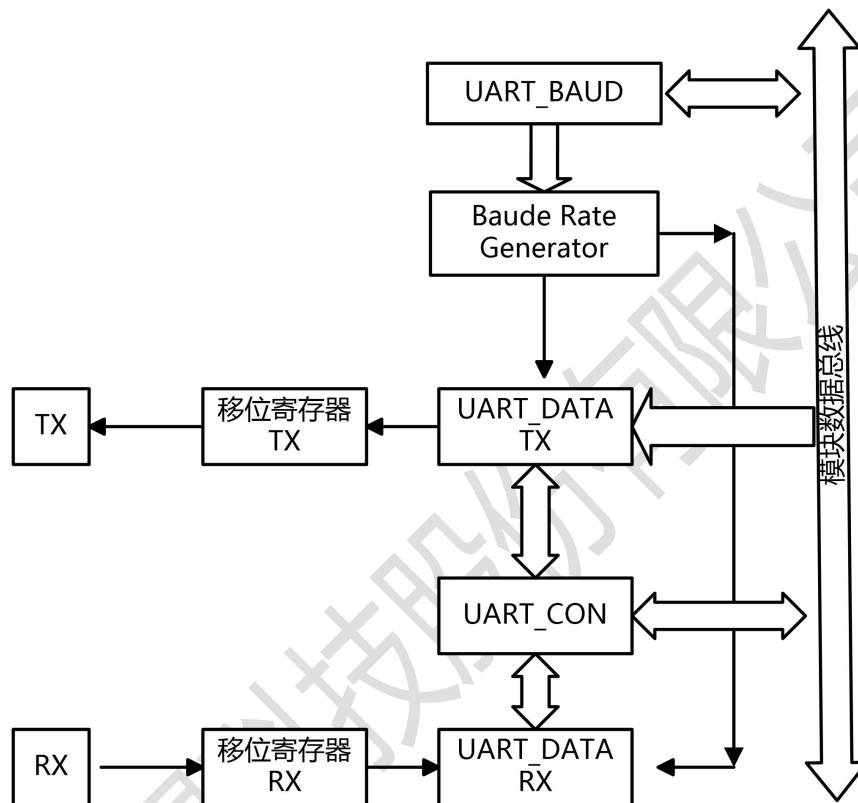


图 13-1 UART 框图

13.3 UART 特性

UART0:

- 支持 8bit 数据和 9bit 数据模式
- 支持全双工异步通信、单工发送、单工接收、单线收发模式
- 支持奇偶校验，奇校验/偶校验可选
- 支持检测系统更新数据串(0x70->0x61->0x73->0x53->0xF5->0x1E->0xF4->0xEC)
- 具有 4 帧数据的接收缓存
- 硬件检测接收 Time out，Time out 长度可以配置，配置范围：1~65536 比特率时间。
- 支持多机通信
- 支持超时检测
- 支持帧出错检测

13.4 IO 映射

详细参考管脚定义

13.5 寄存器描述

13.5.1 UART_CON

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|-----------------|--------|----------|--|
| 31:16 | TO_BIT_LEN | 16'h36 | RW | 超时配置.单位是比特率时间 $\text{Time_out_time} = (\text{TO_BIT_LEN} + 1) * \text{bit_rate_time}$ |
| 15 | TCIE | 1'b0 | RW | 传输完成中断使能 0: 不使能 1: 当 UART_STA 寄存器中的 TC=1 时, 就会生成一个 UART 中断 |
| 14 | TMR_PWM_EN | 1'b0 | RW | UART 的输出和 TMR 的 PWM 一起运算后才输出使能位(UART0 输出 timer0 的 PWM) 0: 不使能 1: 使能 |
| 13 | TO_IE | 1'b0 | RW | Time out 中断使能 0: 不使能 1: 使能 |
| 12 | TO_EN | 1'b0 | RW | 超时检测使能位。检测是否在经历了 $(\text{TO_BIT_LEN} + 1) * \text{比特率}$ 时间, 没有接收到数据。每次 TO_EN 后, 需要等到接收到一 byte 数据才会开始检测。接收到 1byte 数据这个标志, 会在每次清 TO_PEND 的时候被清掉, 或者在 TO_EN 等于 0 的时候被清掉。 0: 不使能 1: 使能 |
| 11 | FERR_IE | 1'b0 | RW | 帧出错中断使能位 0: 不使能 1: 使能 |
| 10 | TXBUF_EMPTY_IE | 1'b0 | RW | 发送 buffer 空中断使能位 0: 不使能 1: 使能 |
| 9 | RXBUF_NEMPTY_IE | 1'b0 | RW | 接收 buffer 非空中断使能位 0: 不使能 1: 使能 |
| 8 | TX_INV | 1'b0 | RW | 发送输出信号取反选择位 0: 不使能 1: 使能 |
| 7 | RX_INV | 1'b0 | RW | 接收输入信号取反选择位 0: 不取反 1: 取反 |
| 6 | ODD_EN | 1'b0 | RW | 奇偶校验选择 0: 偶校验 1: 奇校验 |
| 5 | PARITY_EN | 1'b0 | RW | 奇偶校验使能 0: 不使能 |

| | | | | |
|-----|-----------|------|----|--|
| | | | | 1: 使能 注: 奇偶检验和 BIT9_EN 只能二选一 |
| 4 | BIT9_EN | 1'b0 | RW | UART 传输 9bit 数据选择位 0: UART 传输 8bit 数据 1: UART 传输 9bit 数据 注: 奇偶检验和 BIT9_EN 只能二选一 |
| 3 | STOP_BIT | 1'b0 | RW | 结束位使能位 0: 一个结束位 1: 2 结束位 |
| 2:1 | WORK_MODE | 2'b0 | RW | b00: 双工, 可以同时收发 b01: 单工发送 b10: 单工接收 b11: 一根数据线(单 pin 收发), 硬件自动切换该 IO 方向 |
| 0 | UART_EN | 1'b0 | RW | UART 模块使能位 0: 不使能 1: 使能 |

13.5.2 UART_BAUD

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|----------|---------|----------|--|
| 31:18 | Reserved | - | - | - |
| 17:0 | UARTBAUD | 18'ha93 | RW | UART 波特率设置 波特率=SYS_CLK/(UARTBAUD+1) 注: UARTBAUD 一定要配置为大于等于 6, 否则输入信号会被内部滤波器滤掉。 |

13.5.3 UART_DATA

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:9 | Reserved | - | - | - |
| 8:0 | UARTDATA | 9h0 | RW | UART 传输数据 写 8/9 位数据到 UARTDATA: 触发 UART 模块开始发送数据。 读 UARTDATA: 获取接收到的低 8/9bit 数据, 读之前, 先读取 PERR[0]获取奇偶校验信息。 |

13.5.4 UART_STA

Address offset: 0x0C

| Width | Name | Reset | Property | Description |
|-------|-----------------|-------|----------|---|
| 31:17 | Reserved | - | - | - |
| 16 | ADDR_MATCH_PEND | 1'b0 | RW | 多机通信地址匹配 Pending, 写 1 清零 1: 地址匹配 0: 地址不匹配 |
| 15 | TC | 1'b0 | RW | 发送完成 当一帧发送完成时或 TX_BUF_EMPTY 位置位时由硬件置位, 如果 TCIE=1(UARTCON)将生成一个中断。 写 1 清除或者对 USART_DR 寄存器写数据 |

| | | | | |
|------|----------------------|------|----|---|
| | | | | 0: 发送未完成 1: 发送完成 |
| 14 | UPDATE_DETECT_EN | 1'b0 | RO | 系统升级检测使能位。检测是否接收到下面一串连续数据 0x70->0x61->0x73->0x53->0xF5->0x1E->0xF4->0xEC 0: 不使能 1: 使能 注：当 UPDATE_DETECT_EN 为 1 且 UPDATE_DETECT_PEND 为 0 时候， UARTCON：将固定为 0x00361005(单工接收，1stop, 8bit 模式，没有奇偶校验，没有取反，所有中断不使能)； UARTBAUD：将由 FLASH 模块配置，默认值会是 0xa93； UART_DMACON：将固定为 0x00(不使能 DMA) |
| 13 | UPDATE_DETECT_PEND_G | 1'b0 | RO | 这一位和 UPDATE_DETECT_PEND 一样，只是这位一旦为 1，除非 reset UART0，否则一直不会变为 0. |
| 12 | UPDATE_DETECT_PEND | 1'b0 | RW | 检测到系统升级标志位。表示接收到了下面的一串数据 0x70->0x61->0x73->0x53->0xF5->0x1E->0xF4->0xEC。 写 1 清零 0: 没有检测到系统升级 1: 检测到系统升级 注：只有 UART0 有该位。只能软件清。 |
| 11 | TO_PEND | 1'b0 | RW | Time out 标志位。表示经历了 Time_out_time 时间，没有接收到数据。Time_out_time 配置详见 UART_CON 的 TO_BIT_LEN 写 1 清零。 0: time out 没有出现 1: time out 出现 注：只有 UART0 有该位。 |
| 10:7 | PERR | 4'b0 | RO | 接收数据奇偶校验出错，4bit，分别对应 BUF 里面的 4 个数据。在读 UARTDATA 之前，需要先读这个寄存器，硬件会自动将 PERR[0]对应你读 UARTDATA 获取的 data。 |
| 6:4 | RX_CNT | 3'b0 | RO | RX BUF 有多少个数据 b000: 0 个数据 b001: 1 个数据 b010: 2 个数据 b011: 3 个数据 b100: 4 个数据 其它: 不允许 |
| 3 | FERR | 1'b0 | RW | 帧错误，表示在 STOPbit 期间检测到 RX_IN 出现了低电平。写 1 清零 0: 没有帧错误 |

| | | | | |
|---|------------------|------|----|--|
| | | | | 1: 出现了帧错误 |
| 2 | RX_BUF_OV | 1'b0 | RW | 接收 BUF 最多可以容纳 4 个 8/9bit 的数据, 接收到 4 个数据后, 如果软件还没有来得及读走, 又收到一个数据, 这个标志位会起来。写 1 清零。 0: 接收了(0~4byte)数据 1: 接收了(>4byte)的数据, BUF 里面只保存最开始的 4byte, 其他丢弃 |
| 1 | RX_BUF_NOT_EMPTY | 1'b0 | RW | 接收 BUF 不空标志位, 写 1 清零。 0: 接收 BUF 没有数据 1: 接收 BUF 有数据 |
| 0 | TX_BUF_EMPTY | 1'b0 | RW | 发送 buffer 空标志。UART 内部除了发送移位寄存器外, 还有 1 帧(9bit)发送缓存, 若发送缓存为空这位会被置 1。软件通过写寄存器 UARTDATA 清零该位。 0: 发送缓存满或者 UART 发送不使能 1: 发送缓存空 |

13.5.5 UART_TSTADR

Address offset: 0x10

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:13 | Reserved | - | - | - |
| 12:0 | TSTADRT | 13'h0 | RW | 发送 DMA 起始地址 如果是 9bit 数据, 需要 16bit 对齐 如果是 8bit 数据, 可以任意配置 |

13.5.6 UART_RSTADR

Address offset: 0x14

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:13 | Reserved | - | - | - |
| 12:0 | RSTADRT | 13'h0 | RW | 接收 DMA 起始地址 如果是 9bit 数据, 需要 16bit 对齐 如果是 8bit 数据, 可以任意配置 |

13.5.7 UART_TDMALEN

Address offset: 0x18

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:12 | Reserved | - | - | - |
| 11:0 | TDMALEN | 12'h0 | RW | 计划发送 DMA byte 长度 8bit 数据模式, 可以任意配置 9bit 数据, 需 16 位对齐, TDMALEN 需要配置成 2 的倍数 |

13.5.8 UART_RDMALEN

Address offset: 0x1C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:12 | Reserved | - | - | - |
| 11:0 | RDMALEN | 12'h0 | RW | 计划接收 DMA byte 长度 8bit 数据模式, 可以任意配置 9bit 数据, 需 16 位对齐, TDMALEN 需要配置成 2 的倍数 |

13.5.9 UART_TDMACNT

Address offset: 0x20

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:12 | Reserved | - | - | - |
| 11:0 | TDMACNT | 12'h0 | RO | 已经发送 DMA byte 长度 8bit 数据模式, 可以任意配置。 9bit 数据, 需 16 位对齐, TDMALEN 需要配置成 2 的倍数。 |

13.5.10 UART_RDMACNT

Address offset: 0x24

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:12 | Reserved | - | - | - |
| 11:0 | RDMACNT | 12'h0 | RO | 已经接收 DMA byte 长度 8bit 数据模式, 可以任意配置。 9bit 数据, 需 16 位对齐, TDMALEN 需要配置成 2 的倍数。 |

13.5.11 UART_DMACON

Address offset: 0x28

| Width | Name | Reset | Property | Description |
|-------|----------------|-------|----------|--|
| 31:5 | Reserved | - | - | - |
| 9 | RX_DMA_EN_PRT | 1'b0 | WO | 配置 RX_DMA_EN 时, 必须将该位同时写 1 |
| 8 | TX_DMA_EN_PRT | 1'b0 | WO | 配置 TX_DMA_EN 时, 必须将该位同时写 1 |
| 7:5 | Reserved | - | - | - |
| 4 | RX_DMA_PERR_IE | 1'b0 | RW | 接收 DMA 一包数据出现至少一个数据奇偶校验错误, 中断使能 0: 不使能 1: 使能 |
| 3 | RX_DMA_IE | 1'b0 | RW | 接收 DMA 中断使能 0: 不使能 1: 使能 |
| 2 | TX_DMA_IE | 1'b0 | RW | 发送 DMA 中断使能 0: 不使能 1: 使能 |
| 1 | RX_DMA_EN | 1'b0 | RW | 接收 DMA 使能 0: 不使能 1: 使能 |
| 0 | TX_DMA_EN | 1'b0 | RW | 发送 DMA 使能 0: 不使能 1: 使能 |

13.5.12 UART_DMA_STA

Address offset: 0x2C

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|------------------------------------|
| 31:3 | Reserved | - | - | - |
| 2 | RX_DMA_PERR | 1'b0 | RC | 接收 DMA 数据出现至少一包数据奇偶校验错误标志。写 1 清 0。 |

| | | | | |
|---|-------------|------|----|---|
| | | | | 0: 没有出现奇偶校验错误 1: 出现至少一包数据奇偶校验错误 |
| 1 | RX_DMA_PEND | 1'b0 | RC | 接收 DMA 完成标志位。指数数据接收完成并保存到 SRAM。写 1 清 0。 0: DMA 没有完成 1: DMA 已经完成 |
| 0 | TX_DMA_PEND | 1'b0 | RC | 发送 DMA 完成标志位。指数数据从 SRAM 取出并且通过 UART 完整发送出去。写 1 清 0。 0: DMA 没有完成 1: DMA 已经完成 |

13.5.13 UART_ML_CON

Address offset: 0x3C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--------------------------------|
| 31:24 | SADEN | 8'h0 | RW | 从机地址 mask |
| 23:16 | SADDR | 8'h0 | RW | 从机地址 |
| 15:3 | Reserved | - | - | - |
| 2 | AMPIE | 1'b0 | RW | 多机通信地址匹配中断使能 0: 禁止 1: 使能 |
| 1 | MISMATCH | 1'b1 | RO | 多机通信地址配置 0: 匹配 1: 不匹配 |
| 0 | MLEN | 1'b0 | RW | 多机通信使能 0: 禁止 1: 使能 |

14 比较器 (COMP)

14.1 简介

芯片内嵌一个通用比较器 COMP0，该比较器属于 SARADC，ADC 工作时比较器不能使用，特点是可独立使用，也可与定时器结合使用。比较器是将一个模拟电压信号与一个基准电压相比较的电路。比较器的两路输入（正极输入和负极输入）为模拟信号，输出则为二进制信号 0 或 1，当输入的电压的差值增大或减小且正负符号不变时，其输出保持恒定。

14.2 主要特性

- 轨对轨比较器
- 内部负极连接到 12Bit DAC1 输出
- 支持比较结果的滤波功能
- 输出端可以重定向到一个 I/O 端口或多个定时器输入端，可以触发定时器的捕获事件
- 每个比较器都可产生中断，并支持把 CPU 从睡眠模式和停机模式唤醒

14.2.1 模块框图

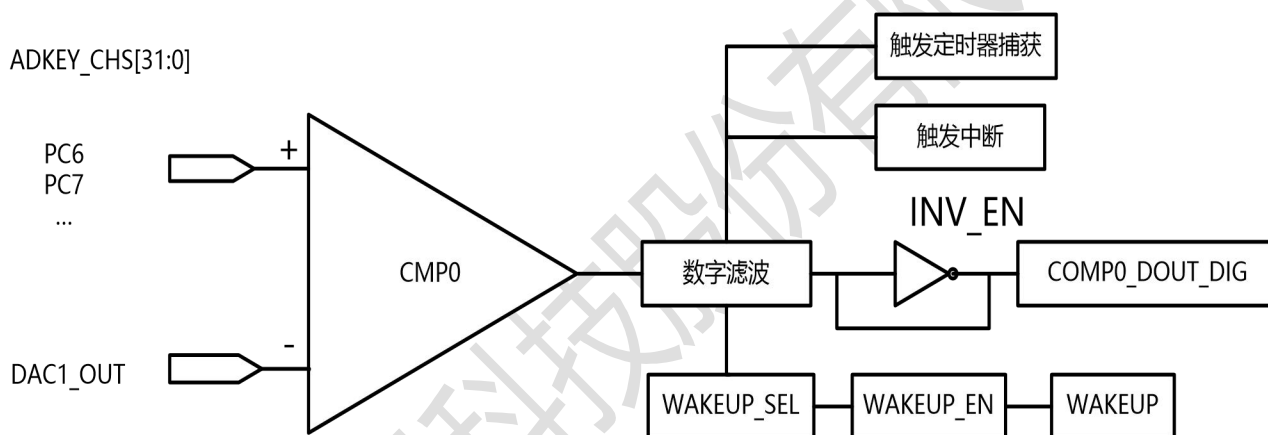


图 14-1 比较器模块框图

14.2.2 比较器 0 输入和输出

作为比较器正极输入的 I/O 引脚必须在 GPIO 寄存器中设置为模拟模式，通道选择在 ADKEY_CHS 寄存器。

比较器的输出可以内部重定向到各种定时器的输入：作为输入捕获，测量时序。

比较器的正极输入可以内部重定向到 DAC0 的输出，DAC0_OUT 为内部独立 8 位 DAC 输出。

14.2.3 中断和唤醒

比较器的输出可以内部连接到外部中断和时间控制器。每个比较器都有一个输出唤醒信号，能产生中断或事件，同样的机制能用来退出低功耗模式。

14.2.4 比较器锁定机制

比较器能用于安全的用途，比如过流保护或者过热保护。在某些特定的安全需求的应用中，有必要保证比较器设置不能被无效寄存器访问或者程序计数器破坏所改变。

为了这个目的，比较器控制和状态寄存器可以设为写保护。一旦设置完成，LOCK 位必须设为 1，这导致整个 CMP_CON0 寄存器变为只读，包括 LOCK 位在内。写保护只能被 MCU 复位清除。

14.3 寄存器描述

14.3.1 配置寄存器 0 (CMP_CON0)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31 | LOCK | 1'b0 | RW | 比较器锁 该位只能写一次，由软件置“1”，由系统复位清零。 它令比较器的所有控制位为只读。 0: CMP_CON0 可读可写 1: CMP_CON0 只读 |
| 30:19 | Reserved | - | - | - |
| 18:17 | PEND_SEL | 2'b0 | RW | COMP0_PEND 触发选择： b01: 比较器 0 上升沿触发 PEND b10: 比较器 0 下降沿触发 PEND b11: 比较器 0 上升沿和下降沿都触发 PEND 其它：保留 |
| 16:12 | FILT_NUM | 5'h0 | RW | 比较器输出滤波周期选择 |
| 11:4 | Reserved | - | - | - |
| 3 | INV_EN | 1'b0 | RW | 输出结果取反使能 |
| 2 | WKUP_EN | 1'b0 | RW | 唤醒使能 |
| 1 | INT_EN | 1'b0 | RW | 中断使能(输出高触发中断) |
| 0 | ENABLE | 1'b0 | RW | 比较器使能 |

14.3.2 状态寄存器 (CMP_STA)

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|-----------|-------|----------|---------------|
| 31:1 | Reserved | - | - | - |
| 0 | CMP0_PEND | 1'b0 | RO | 比较器 0 PEND 标志 |

14.3.3 状态寄存器 (CMP_CLR)

Address offset: 0x0C

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|--------------------|
| 31:1 | Reserved | - | - | - |
| 0 | CMP0_PEND_CLR | 1'b0 | WO | 写“1”清比较器 0 PEND 标志 |

14.4 操作流程

- 配置寄存器 CMP_CON0 选择比较器的正极和负极的输入引脚
- 根据比较器的工作方式配置对应模块功能（唤醒、输出到 IO、启动 ADC 采样等）
- 配置寄存器 CMP_CON0 使比较器工作
- 通过读取状态寄存器(CMP_STA)可以得到比较器的输出标志信号，并且配置状态寄存器(CMP_CLR)清除输出标志信号

15 数模转换器 (DAC)

15.1 简介

芯片内嵌两个 DAC，DAC0 是 8 位数字输入，电压输出的数模转换器。DAC1 是 12 位数字输入，该 DAC 属于 SARADC 的组成部分，SARADC 工作时不能使用。

15.2 模块框图

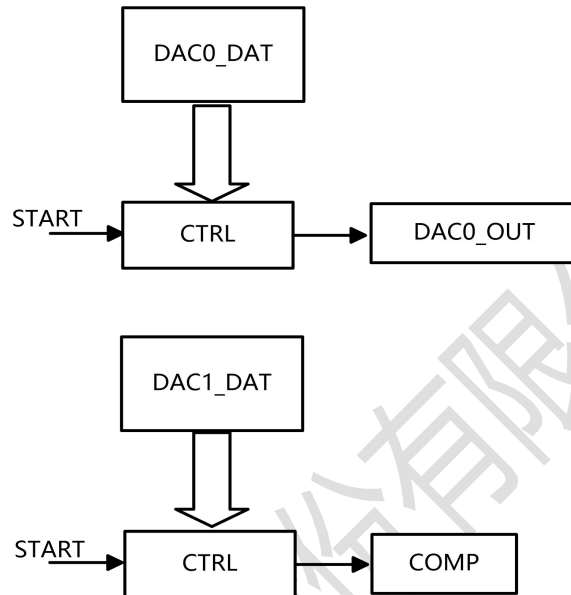


图 15-1 DAC 示意框图

15.3 主要特征

- DAC 输出可重定向到 IO
- DAC0 可输出到 COMP0 的正端，以及 PA1 输出

15.4 寄存器描述

15.4.1 配置寄存器 (DAC_CON)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31 | LOCK | 1'b0 | RW | DAC 寄存器锁 该位只能写一次，由软件置“1”，由系统复位清零。 它令比较器的所有控制位为只读。 0: DAC_CON 可读可写 1: DAC_CON 只读 |
| 30:10 | Reserved | - | - | - |
| 9:2 | DAT_SEL | 8'h0 | RW | DAC 输入 data |
| 1 | DIVEN | 1'b0 | RW | IO 输出使能位，置“1”有效 |
| 0 | ENABLE | 1'b0 | RW | DAC 使能位，置“1”有效 |

注：DAC1 配置寄存器，参考 ADC 章节 ADKEY_CFG 和 ADKEY_DADAT 寄存器。

16 模数转换器 (ADC)

16.1 功能简介

该模块是一个 12 位逐次逼近式的 ADC 控制器。ADC 支持多种工作模式：单次转换、单周期扫描和连续扫描模式，并且可选择通道自动扫描，转换的结果可以左对齐或右对齐的方式存储在 16 位寄存器中。ADC 的启动包括软件启动、外部引脚触发以及其他片内外设启动（TIMER 触发启动）。

16.2 模块框图

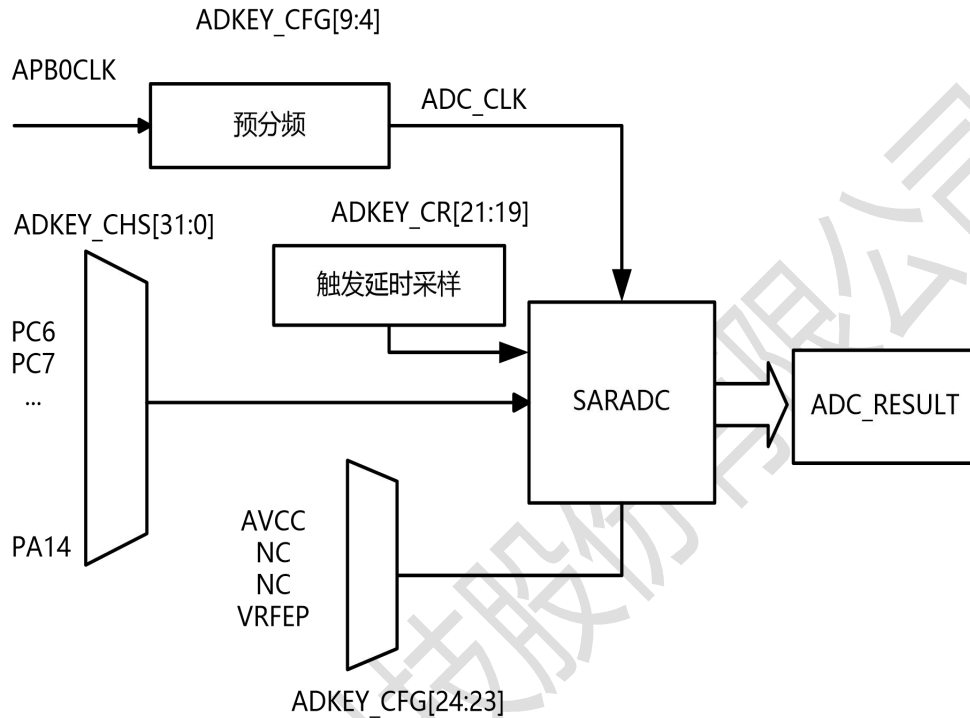


图 16-2 ADC 示意框图

16.3 主要特征

- 最高 12 位可编程分辨率的 ADC，多达 26 路（26 路@LQFP64，13 路@LQFP48）外部输入通道
- 高达 250K 采样率
- 支持多种工作模式：
 - 单次转换模式：A/D 转换在指定通道完成一次转换
 - 单周期扫描模式：A/D 转换在所有指定通道完成一个周期(从低序号通道到高序号通道或者从高序号通道到低序号通道)转换
 - 连续扫描模式：A/D 转换连续执行单周期扫描模式直到软件停止 A/D 转换
- 通道采样时间，分辨率可软件配置
- 支持 DMA 传输
- A/D 转换开始条件：
 - 软件启动
 - 外部 IO 触发启动
 - 内部 TIMER 触发启动

16.4 功能描述

16.4.1 ADC 开关控制

通过设置 ADCFG 寄存器的 ADEN 位可给 ADC 上电。当第一次设置 ADEN 位时，它将 ADC 从断电

状态下唤醒。ADC 上电延迟一段时间后(t_{STAB})，设置采样模式、通道之后，设置 ADST 位开始进行转换。通过清除 ADST 位可以停止转换，设置 ADEN 位可置于断电模式。

16.5 通道选择

包含 26 路外部输入通道、6 路内部通道。每个外部输入通道都有独立的使能位，可通过设置 ADCHS 寄存器的对应位来设置。

16.5.1 ADC 工作模式

16.5.1.1 单次转换模式

在单次转换模式下，A/D 转换相应通道上只执行一次，具体流程如下：

- 软件配置 ADC_EN，给 ADC 上电。
- 通过软件、外部触发输入及定时器溢出置位 ADST，开始 A/D 转换。
- 当 A/D 转换完成，A/D 转换的数据值将存储于 A/D 的数据寄存器 ADDATA 中。
- A/D 转换完成，状态寄存器 ADSTA 的 ADIF 位置 '1'。若此时控制寄存器 ADCRL 的 ADIE 位置 '1'，将产生 AD 转换结束中断请求。
- A/D 转换期间，ADST 位保持为 1。A/D 转换结束，ADST 位自动清 0，A/D 转换器进入空闲模式。

16.5.1.2 单周期扫描模式

在单周期扫描模式下，将进行一次从被使能的最小序号通道向最大序号通道的 A/D 转换，可通过配置寄存器位 SCAN_DIR 选择扫描通道方向，操作步骤如下：

- 配置为 DMA 模式。
- 软件或外部触发置位 ADST 开始，外部触发可软件配置触发延时，方向设置默认从最小序号通道到最大序号通道的 A/D 转换。
- 每路 A/D 转换完成后，A/D 转换数值将有序装载到相应的 SRAM 中，ADIF 转换结束标志被设置，如果设置了转换结束中断，则在所有通道转换都完成后产生中断请求。
- 转换结束后，ADST 位自动清 0，A/D 转换器进入空闲状态。

16.5.1.3 连续扫描模式

在连续扫描模式下，A/D 转换在 ADCHS 寄存器中的 CHENn 位被使能的通道上顺序进行(可通过配置寄存器位 SCAN_DIR 选择扫描通道方向)，操作步骤如下：

- 软件或外部触发置位 ADST 开始，外部触发可软件配置触发延时，方向设置默认从最小序号通道到最大序号通道的 A/D 转换。
- 每路 A/D 转换完成后，A/D 转换数值将有序装载到相应的 SRAM 中。
- 当所有通道的 A/D 转换完成一遍后，ADIF 转换结束标志被设置。如果设置了转换结束中断，则在所有通道转换都完成后产生中断请求。
- 只要 ADST 位保持为 1，就重复步骤 2 到 3。当 ADST 位被清 0，A/D 转换停止，A/D 转换器进入空闲状态。当 ADST 清 0，A/D 转换结束。

16.5.2 DMA 请求

单周期扫描和连续扫描时最近一次转换的结果会保存在 ADDATA 寄存器中。DMA 传输时传输所有扫描通道的结果顺序存放到对应 SRAM 中，并且由 DATCNT 寄存器指示存放的采样个数。

16.5.3 采样频率设置

ADC 的时钟 ADCLK 由 APB0 分频得到，分频系数可通过设置 ADKEY_CFG 寄存器的 ADCPRE 位来确定，即 $APB0/(N+1)$ 分频后作为 ADC 时钟。

$$ADC \text{ 采样率} = F_{PCLK} / (ADCPRE + 1) / 15$$

16.5.4 连续采样间隔时间可配置

对于连续扫描模式，连续两个通道采样之间可以配置采样延时(即在一个通道扫描完成之后，等待一段时间再进行下一通道扫描)。

16.5.5 外部触发转换

ADC 转换可以由外部事件触发(例如定时器、IO)。如果设置了 ADKEY_CR 寄存器的 TRGEN 位，就可以使用外部事件触发转换。通过设置 TRGSEL 位可以选择外部触发源。

具体的外部触发源选择情况，可以参考 AD 控制寄存器(ADKEY_CR)位[17]和[6: 4] TRGSEL 的描述。

外部触发可设置延时控制，具体参考 ADKEY_CR[20:18]的 TRGSHIFT 的描述。

在触发信号产生后，延时 N 个 PCLK 的时钟周期再开始采样。如果是触发扫描模式，则只有第一个通道采样被延时，其他通道是在上一个采样结束后立即开始。

16.6 寄存器描述

16.6.1 ADKEY_CFG

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|--------|----------|-------|----------|---|
| 31:25 | Reserved | - | - | - |
| 24: 23 | AD_VSEL | 2'b0 | RW | 基准电压选择 b00: AVCC b01: 保留 b10: 保留 b11: VREFP |
| 22:17 | Reserved | - | - | - |
| 16:13 | D2DCYC | 4'b0 | RW | 两次连续采样之间的间隔时间周期($n \times \text{ADCCLK}$) |
| 12:10 | Reserved | - | - | - |
| 9:4 | ADCPRE | 6'hf | RW | ADC 预分频 0: 不分频; 其他: $(n+1)$ 分频 |
| 3 | Reserved | - | - | - |
| 2 | DACEN | 1'b0 | RW | DAC 使能 0: 不使能 1: 使能 |
| 1 | Reserved | - | - | - |
| 0 | ADEN | 1'b0 | RW | A/D 转换使能 0: 禁用 1: 使能 |

16.6.2 ADKEY_CR

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:29 | Reserved | - | - | - |
| 28 | CALEN | 1'b1 | RW | 硬件数据校正使能，高有效 |
| 27 | SW_RST | 1'b0 | WO | 软件复位模块内部状态，高有效 |
| 26 | DATASIGN | 1'b0 | RW | 数据扩张位选择，影响 ADKEY_DATA 的最高 3 位或最低 3 位 |
| 25 | OVRIE | 1'b0 | RW | 溢出中断使能 |
| 24 | EOCS | 1'b0 | RW | 设置 ADIF 标志位，只有 scan mode 的时候有效 0: 每次 sequence 完成产生 ADIF |

| | | | | |
|-------|----------|------|----|--|
| | | | | 1: 在 scan mode 结束的时候产生 ADIF |
| 23 | DMAFIE | 1'b0 | RW | DMA 全满中断使能 |
| 22 | DMAHIE | 1'b0 | RW | DMA 半满中断使能 |
| 21:19 | TRGSHIFT | 3'b0 | RW | 外部触发延时采样(External trigger shift sample), 在触发信号产生后, 延时 N 个 PCLK 的时钟周期再开始采样。 b000: 不延时 b001: 4 个周期 b010: 16 个周期 b011: 32 个周期 b100: 64 个周期 b101: 128 个周期 b110: 256 个周期 b111: 512 个周期 |
| 18:17 | Reserved | - | - | - |
| 16 | SCANDIR | 1'b0 | RW | ADC 扫描通道顺序(ADC scan direction) 在单周期扫描或者连续扫描方式时, 设置扫描通道的顺序 0: ADC 通道选择寄存器按从低到高的顺序扫描 1: ADC 通道选择寄存器按从高到低的顺序扫描 |
| 15:12 | Reserved | - | - | - |
| 11 | ALIGN | 1'b0 | RW | 数据对齐(Data alignment) 0: 左对齐 1: 右对齐 |
| 10:9 | ADMD | 2'b0 | RW | A/D 转换模式(ADC mode) b00: 单次转换 b01: 单周期扫描 b10: 连续扫描 当改变转换模式时, 软件要先禁用 ADST 位。 |
| 8 | ADST | 1'b0 | RW | ADST: A/D 转换开始(ADC start) 0: 转换结束或进入空闲状态 1: 转换开始 ADST 置位有下列两种方式: 在单次模式或者单周期模式下, 转换完成后, ADST 将被硬件自动清除。 在连续扫描模式下, A/D 转换将一直进行, 直到软件写 0 到该位或系统复位。 |
| 7:4 | TRGSEL | 4'b0 | RW | 外部触发源选择 b0000: timer0 触发 b0001: timer1 触发 b0010: timer2 触发 b0011: timer3 触发 b0100: timer4 触发 b0101: timer5 触发 b1110: PB12 触发 b1111: PB13 触发 其他: 保留 |
| 3 | DMAEN | 1'b0 | RW | DMA 使能(Direct memory access enable) |

| | | | | |
|---|----------|------|----|--|
| | | | | 0: DMA 禁止 1: DMA 请求使能 |
| 2 | TRGEN | 1'b0 | RW | 外部硬件触发源(External trigger enable) 0: 不用外部触发信号启动 A/D 转换 1: 使用外部触发信号启动 A/D 转换 |
| 1 | Reserved | - | - | - |
| 0 | ADIE | 1'b0 | RW | A/D 中断使能(ADC interrupt enable) 0: 禁用 A/D 中断 1: 使能 A/D 中断 如果 ADINT 置位, A/D 转换结束后产生中断请求 |

16.6.3 ADKEY_CHS

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|-------|-------|----------|--|
| 31:0 | CHXEN | 32'h0 | RW | 连续扫描模式下 ADC 输入通道使能 0: 禁用 1: 使能 CHXEN<0>--PC6 CHXEN<1>--PC7 CHXEN<2>--PC8 CHXEN<3>--PC9 CHXEN<4>--PE2 CHXEN<5>--PC10 CHXEN<6>--PC11 CHXEN<7>--PC12 CHXEN<8>--PC13 CHXEN<9>--PC14 CHXEN<10>--PC15 CHXEN<11>--PD0 CHXEN<12>--PD1 CHXEN<13>--PD2 CHXEN<14>--PD3 CHXEN<15>--PD4 CHXEN<16>--PD5 CHXEN<17>--PA1 CHXEN<18>--PA2 CHXEN<19>--PA8 CHXEN<20>--PA9 CHXEN<21>--PA10 CHXEN<22>--PA11 CHXEN<23>--PA12 CHXEN<24>--PA13 CHXEN<25>--PA14 CHXEN<27>--VPTAT CHXEN<28>--VDIV2_OUT CHXEN<29>--DAC0_OUT,PA1 |

16.6.4 ADKEY_STA

Address offset: 0x10

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:24 | Reserved | - | - | - |
| 23:19 | CSPAD | 5'h1f | RW | 测试输出 PIN 选择位 输出 PIN 为 ChxEN[CSPAD](CSPAD 小于 26) |
| 18:16 | Reserved | - | - | - |
| 15:12 | Reserved | - | - | - |
| 11 | DMAFF | 1'b0 | RW | DMA 全满标志 写 1 清零 |
| 10 | DMAHF | 1'b0 | RW | DMA 半满标志 写 1 清零 |
| 9:4 | CHANNEL | 6'h21 | RW | 当前转换通道(Current conversion channel) 该 6 位在 BUSY = 1 时表示进行转换中的通道。BUSY = 0 时表示可进行下次转换的通道。 |
| 3 | Reserved | - | - | - |
| 2 | BUSY | 1'b0 | RW | 忙/空闲(Busy) 0: AD 转换器空闲 1: AD 转换器忙碌 |
| 1 | Reserved | - | - | - |
| 0 | ADIF | 1'b0 | RW | AD 转换结束标志位(ADC interrupt flag) 该位由硬件在通道组转换结束时设置, 由软件清除。 0: AD 转换未完成 1: AD 转换完成 写 1 清零。 |

16.6.5 ADKEY_DMAADR

Address offset: 0x14

| Width | Name | Reset | Property | Description |
|-------|--------|-------|----------|-------------|
| 31:0 | DMAADR | 32'h0 | RW | DMA 起始地址 |

16.6.6 ADKEY_DATA

Address offset: 0x18

| Width | Name | Reset | Property | Description |
|-------|------------|-------|----------|---|
| 31:24 | Reserved | - | - | - |
| 23 | VALID | 1'b0 | RO | 有效标志位 0: DATA[11: 0]位数据无效 1: DATA[11: 0]位数据有效 相应模拟通道转换完成后, 将该位置位, 读 ADDATA 寄存器后, 该位由硬件清除。 |
| 22 | OVERRUN | 1'b0 | RO | 数据覆盖标志位 0: DATA [11: 0]数据最近一次转换结果 1: DATA [11: 0]数据被覆盖 新的转换结果装载至寄存器之前, 若 DATA[15: 0]的数据没有被读取, OVERRUN 将置 1。读 ADDATA 寄存器后, 该位由硬件清除。 |
| 21:16 | CHANNELSEL | 6'h0 | RO | 该 6 位显示当前数据所对应的通道 n: 通道 n 的转换数据 |
| 15:0 | DATA | 16'h0 | RO | 12 位 AD 转换结果 根据设置左对齐或者右对齐。 |

16.6.7 ADKEY_DADAT

Address offset: 0x1C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:12 | Reserved | - | - | - |
| 11:0 | DACDAT | 12'h0 | RW | DAC 数据 |

16.6.8 ADKEY_DATCNT

Address offset: 0x20

| Width | Name | Reset | Property | Description |
|-------|-----------|-------|----------|----------------|
| 31:12 | Reserved | - | - | - |
| 11:0 | DMADATCNT | 12'h0 | RW | 已经 DMA 完成的数据长度 |

16.6.9 ADKEY_DMALEN

Address offset: 0x24

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:12 | Reserved | - | - | - |
| 11:0 | DMALEN | 12'h0 | RW | 配置 DMA 数据缓存长度：数据缓冲长度 = DMALEN+1 个 ADC 数据 当收满指定数据缓冲长度之后，循环存储到指定 SRAM 地址 |

16.7 操作流程

16.7.1 单次采样

- 配置寄存器 ADKEY_CFG 使能 ADC
- 配置寄存器 ADKEY_CFG 选择对应工作的频率
- 配置寄存器 ADKEY_STA 选择示例通道
- 配置寄存器 ADKEY_CR 选择 ADC 转换模式为单次采样
- 配置寄存器 ADKEY_CR 选择数据对齐格式
- 配置寄存器 ADKEY_CR 使 ADC 转换开始
- 等待转换完成：读取寄存器 ADKEY_STA 第 0 位的值进行判断
- 清除 ADC 转换开始标志位，停止工作：配置寄存器 ADKEY_CR 使 ADC 转换进入空闲状态

16.7.2 周期采样

- 配置寄存器 ADKEY_CFG 使能 ADC
- 配置寄存器 ADKEY_CFG 选择对应工作的频率
- 配置寄存器 ADKEY_CFG 选择相应的两次连续采样之间的间隔时间周期
- 配置寄存器 ADKEY_CHS 选择扫描第一条通道
- 配置寄存器 ADKEY_CR 选择扫描通道顺序
- 配置寄存器 ADKEY_CR 选择 ADC 转换模式为单周期扫描
- 配置寄存器 ADKEY_CR 选择数据对齐格式
- 通过配置寄存器 ADKEY_DMAADR 配置 DMA 地址
- 寄存器 ADKEY_CR 使能 DMA
- 配置寄存器 ADKEY_CR 使 ADC 转换开始
- 等待转换完成：读取寄存器 ADKEY_STA 第 0 位的值进行判断
- 清除 ADC 转换开始标志位，停止工作：配置寄存器 ADKEY_CR 使 ADC 转换进入空闲状态

16.7.3 扫描模式

- 配置寄存器 ADKEY_CFG 使能 ADC
- 配置寄存器 ADKEY_CFG 选择对应工作的频率

- 配置寄存器 ADKEY_CFG 选择相应的两次连续采样之间的间隔时间周期
- 配置寄存器 ADKEY_CHS 选择扫描通道
- 配置寄存器 ADKEY_CR 选择扫描通道顺序
- 配置寄存器 ADKEY_CR 选择 ADC 转换模式为连续扫描
- 配置寄存器 ADKEY_CR 选择数据对齐格式
- 通过配置寄存器 ADKEY_DMAADR 配置 DMA 地址
- 寄存器 ADKEY_CR 使能 DMA
- 配置寄存器 ADKEY_CR 使 ADC 转换开始

16.7.4 外部 trigger 启动采样

- 配置寄存器 ADKEY_CFG 使能 ADC
- 配置寄存器 ADKEY_CFG 选择对应工作的频率
- 配置寄存器 ADKEY_STA 选择示例通道
- 配置寄存器 ADKEY_CR 选择 ADC 转换模式为外部触发
- 配置寄存器 ADKEY_CR 选择数据对齐格式
- 配置寄存器 ADKEY_CR 使能外部触发
- 配置寄存器 ADKEY_CR 选择外部触发延时采样周期

17 定时器 (TMR0/1/2/3/4/5)

JS32L010 一共包含 6 个普通 Timer。其中 Timer0/1/2/3/5 为 16 位 Timer，Timer4 为 32 位 Timer。下面仅以 16 位计数器进行说明。Timer4 对应的把各种位宽定义为 32 位。

注：TIMx 中的 x 表示第几个 Timer，从 0 到 5。

17.1 16 位定时器 0/1/2/3/5

17.1.1 简介

定时器是由一个通过可编程预分频器驱动的自动装载寄存器构成，适用于测量输入信号脉冲长度（输入捕获）和产生输出波形（输出比较和 PWM），根据选择不同的计数源（系统时钟、内部低速 RC 时钟、外部时钟、GPIO 等）和预分频器的设置，脉冲长度和输出波形周期可以在微秒级和毫秒级别，甚至秒级别的范围。每个定时器都是完全独立的，不共享任何资源，可以一起同步操作。

17.1.2 主要特性

- 16 位递增递减计数器
- 可编程预分频寄存器
- 支持选择 GPIO 作为计数时钟源
- 支持不同计数时钟源
- 捕获功能
 - 最多支持同时保存 2 个捕获事件指针
 - 每个捕获事件极性独立可配置
 - 每次捕获事件发生可配置是否复位计数值
- PWM 输出
- 支持周期和占空比自动重装
- 使用外部信号控制定时器和定时器互联的同步电路

17.1.3 功能描述

17.1.3.1 时基单元

时基单元包括：

- 计数器寄存器(TIMx_CNT)
- 周期寄存器(TIMx_PR)
- 分频寄存器(TIMx_CON[10:8])

在计数模式下，计数器从 0 计数到周期值(TIMx_PR)，然后重新从 0 开始计数，并且产生一个计数器溢出中断。

分频寄存器可以将计数器的时钟频率按 2 的 n 次方分频，从而提高计数的最大周期。

17.1.3.2 计数源选择

计数器时钟可由下列时钟源提供

- 内部低速 RC
- 内部高速 HIRC
- 外部晶振
- 外部 GPIO
- 内部系统时钟

通过配置 TMRx_CTL 寄存器的 INC_SRC_SEL 选择不同的计数源，配置 TMRx_CTL 的 PSC 配置不同的分频系数。

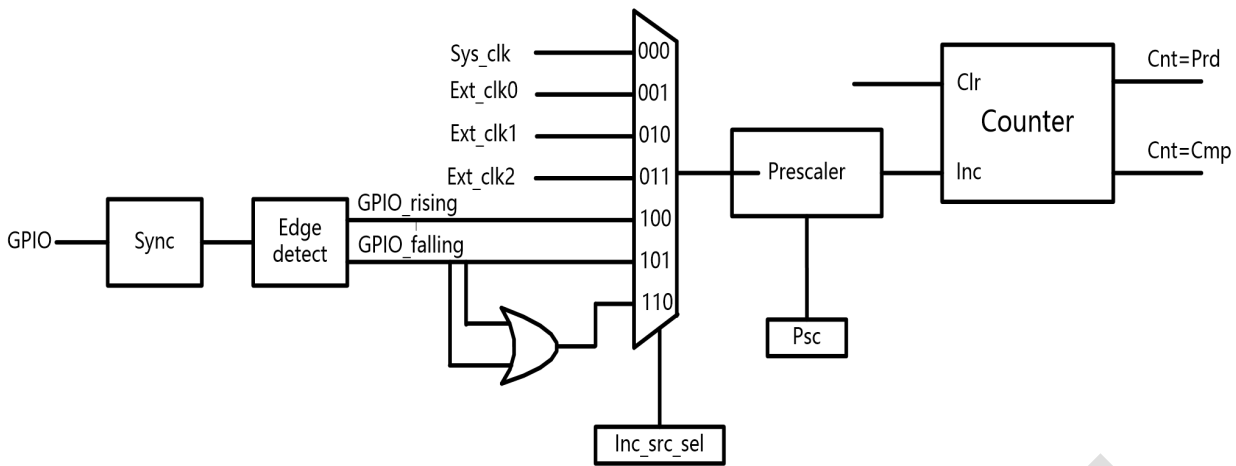


图 17-1 计数器示意框图

17.1.3.3 支持 capture 工作模式

输入捕获模式，支持外部引脚触发或者内部比较器触发，捕获当前计数值并保存到内部寄存器 (TIMx_PWM)。结构如下图：

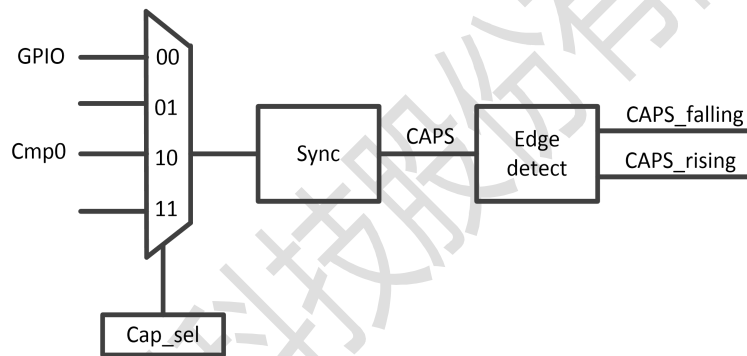


图 17-2 输入捕获通道

上图中 pin 对应 GPIO 输入详细参考芯片管脚定义。

CMP0 是比较器的比较输出信号。首先通过一个选择器选择输入触发通路后，经过同步器同步得到 CAPS，然后边沿检测器产生边沿触发信号，最后通过一个选择器选出最终捕获信号 CAPO。当捕获信号有效时，自动把当前计数器的值存入 TIMx_PWM 寄存器，并且产生捕获中断信号。

17.1.3.4 输入捕获模式

当捕获事件发生时，把当前计数值保存到对应捕获寄存器寄存器(TMRx_CAP1/2)。

通过配置寄存器 TMRx_CON 的 CAP_CNT 可以配置使用 1~2 个捕获寄存器。

通过配置寄存器 TMRx_CON 的 CAP1POL 或 CAP2POL 可以独立配置每个捕获事件极性，选择上升沿或者下降沿捕获。

通过配置寄存器 TMRx_CON 的 CTRRST1/2 可以独立配置对应的捕获事件发生时，是否复位计数器的值。

每次捕获事件发生，都会产生对应的捕获标志(TMRx_FLAG)。通过配置寄存器 TMRx_IE 可以独立配置每次捕获事件发生是否产生中断。

捕获模式下，支持计数值溢出中断。该模式下，计数周期固定为 0xFFFF。当计数值达到 0xFFFF，会产生溢出标志 ovf_flag，通过配置 TMRx_IE 寄存器可以产生溢出中断。

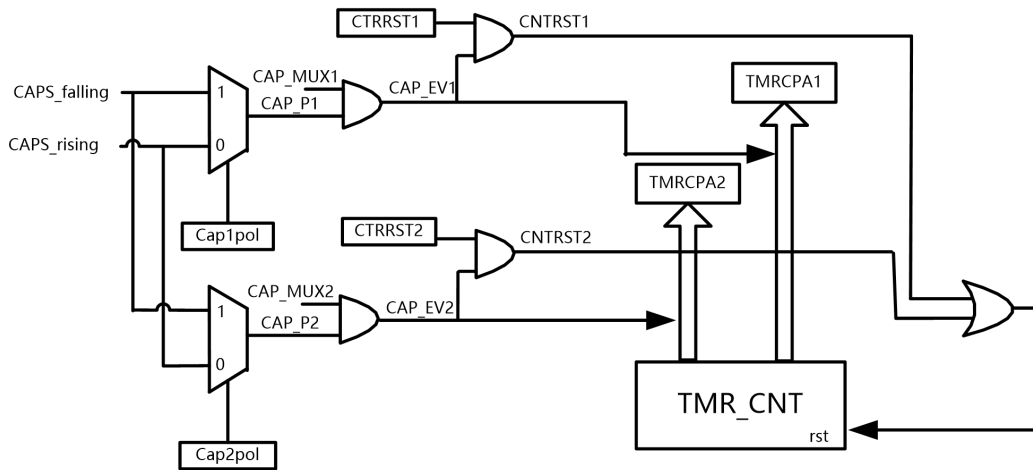


图 17-3 Timer 捕获功能

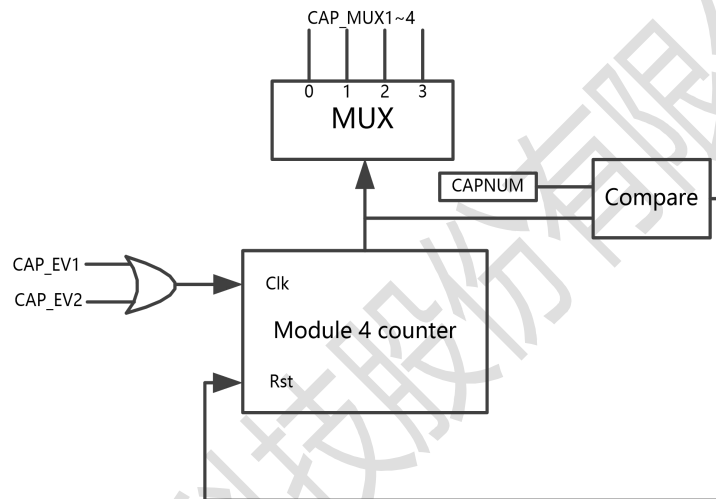


图 17-4 Timer 捕获功能示意

17.1.3.5 触发 ADC 采样

Timer 还有一个增强型的功能，可以在每次 TIMER 计数满的时候，触发一次 ADC 采样。具体参考 ADC 章节。

17.1.3.6 timer 同步输出

输出同步信号可选择：

- 同步输入 SYNCI
- 计数值等于 TMRx_PRD
- 计数值等于 TMRx_CMP

17.1.3.7 从模式

TMR 定时器能够在外部模式下和一个外部的触发 SYNCI 同步：复位模式、触发模式和门控模式。

复位模式：在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；例如，计数器在正常运转，SYNCI 出现一个上升沿，此时计数器被清零然后从 0 重新开始计数。同时触发标志(TMRx_FLAG 中的 SLAVE_FLAG 位)被设置，根据 TMRx_IE 寄存器中的 SLAVE_IE 的设置，产生中断。

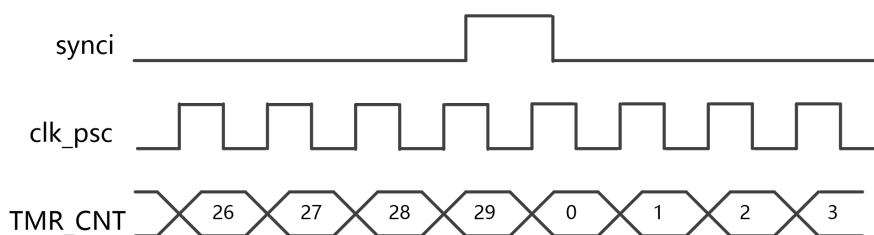


图 17-5 Timer 复位波形

触发模式：计数器的使能依赖于同步输入端 SYNCI 的事件。

如下例子，当 SYNCI 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 SLAVE_FLAG 位，并且根据 TMRx_IE 寄存器中的 SLAVE_IE 的设置，产生中断。

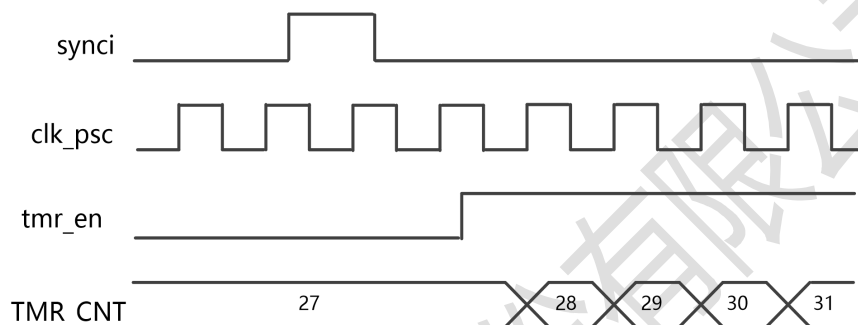


图 17-6 Timer 触发模式波形

门控模式：计数器的使能依赖于输入同步信号 SYNCI 的电平。在 SYNCI 有效电平时，TMRx_CNT 计数，无效电平，停止计数。

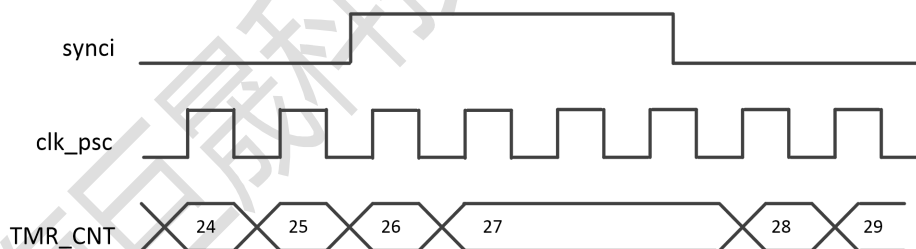


图 17-7 Timer 门控模式波形

17.1.3.8 递增/递减计数模式（计数模式有效）

Timer 工作在计数模式下(MODE_SEL=0)时，通过配置 TMR_CON[26](TMR_DIR_EN)使能，Timer 由外部捕获引脚电平决定递增还是递减计数的模式。

- 当 TMR_DIR_EN=1 时，捕获引脚为低电平则 Timer 递增计数，捕获引脚为高电平则 Timer 递减计数。
- 使用该功能时，应当设置 TMRx_CAP1 为上限值，TMRx_CAP2 设置为下限值，TMRx_CNT 设置为 TMRx_CAP1 和 TMRx_CAP2 之间的中间值。
- 如果外部捕获引脚电平持续为 0 时，Timer 计时会向上限值溢出，溢出会使 OVF_FLAG 标志位置 1 同时 PRD_FLAG 置 1。如果外部捕获引脚电平持续为 1 时，timer 计时会向下限值溢出，溢出会使 OVF_FLAG 置 1 同时 CMP_FLAG 置 1。

17.1.3.9 DMA 传输模式 (只有 Timer1 有 DMA 功能)

DMA 产生 PWM

TMR 支持 DMA 传输, 可以用来先把 PWM 的周期和占空比提前存放到 SRAM 指定位置, 然后每个周期硬件自动从 SRAM 中 load 数据, 修改 PWM 的周期和占空比, 产生期望的 PWM 信号。

具体流程如下:

- 提前把期望产生的 PWM 周期和占空比写到 SRAM, 对于 16bit 寄存器, 低 16 位是 PWM 周期, 高 16 位是占空比。
- 配置 TMRx_DADR 和 TMRx_DLEN 寄存器, 配置 DMA 的起始地址和长度。
- 配置 TMRx_DCTL 寄存器的 DMA_LPBK, 配置采用循环或者单次模式。如果 DMA_LPBK=0, 在执行完指定 dma 长度之后, PWM 自动停止输出。如果 DMA_LPBK=1, 在执行完指定 DMA 长度之后, 自动从起始地址从新开始取数据, 直到软件把 DMA_EN 关闭, 才停止。
- 通过配置 TMRx_DCTL 寄存器的 DMA_EN, 使能 DMA 传输模式, 使能 TIMER。

DMA 读取红外编码自动完成发送 (Timer1 产生门控信号)

流程和上述相同, 需要注意下面 3 点:

- 通过配置 TMRx_DCTL 寄存器中 DMA_IR_EN 使能 DMA 红外模式
- 该模式下写到 SRAM 里面的是需要发送的红外编码, 而不是周期和占空比, TMRx_DLEN 代表的是需要发送的红外编码位数。
- 该模式下还需要配置 TMR1_DMAIR0 和 TMR1_DMAIR1, 其中 TMR1_DMAIR0 设置红外编码为 0 时 timer1 的周期和占空比, TMR1_DMAIR1 设置红外编码为 1 时 Timer1 的周期和占空比。

注意:

如果选择 dma_lpbk=0 的模式, 发送完一次后, DMA_EN 和 TIMER_EN 都会变成 0, 如果想要再次发送, 需要软件重新使能。

捕获数据通过 DMA 写到 SRAM

支持把捕获数据自动保存到 SRAM 中。

- 配置 TMRx_DADR 和 TMRx_DLEN 寄存器, 配置 DMA 的起始地址和长度。
- 配置 TMRx_IE 寄存器的 DMA_FL_IE, 使能 DMA 完成中断。
- 配置并启动捕获功能。
- 中断之后, 在 SRAM 中获取之前的捕获数据。

DMA 中断标志

支持 DMA Buffer 半满中断和全满中断, 通过 TMRx_IE 寄存器的 DMA_HF_IE 和 DMA_FL_IE 使能中断。

The diagram illustrates the internal structure of two timer modules, TMR1 and TMR2, which are part of a larger timer chain. Each module contains a TMR_CNT, Compare, Slave mode control, and logic for Sincip, Syncosel, and Synco signals. The diagram shows the internal structure of two timer modules, TMR1 and TMR2, which are part of a larger timer chain.

TMR1:

- Inputs:** Sincip (from Sincipol and Sincip), Syncosel (from Compare), and Synco (from Synco).
- Outputs:** Pwm_out (from Synco) and Synco (from Synco).
- Internal Components:** TMR_CNT, Compare, Slave mode control, and logic for Sincip, Syncosel, and Synco.

TMR2:

- Inputs:** Sincip (from Sincipol and Sincip), Syncosel (from Compare), and Synco (from Synco).
- Outputs:** Pwm_out (from Synco) and Synco (from Synco).
- Internal Components:** TMR_CNT, Compare, Slave mode control, and logic for Sincip, Syncosel, and Synco.

The diagram shows the internal structure of two timer modules, TMR1 and TMR2, which are part of a larger timer chain. The diagram shows the internal structure of two timer modules, TMR1 and TMR2, which are part of a larger timer chain.

图 17-8 Timer 之间级联

支持把所有 Timer 的计数值同时清零。下面例子，把两个 TMR 的计数值同时清零，同步配置如下：

- 配置 TMR0_CON 寄存器的 SYNCIPOL = 0。
- 配置 TMR0_CON 寄存器的 SYNCOSSEL = 2, 直接把 TMR0 的 Synci 连接到 TMR0 的 Synco。
- 配置 TMR0_CON 寄存器的 SLAVE_MODE = 2, 复位模式。
- 配置 TMR1_CON 寄存器的 SYNCIPOL = 0。
- 配置 TMR1_CON 寄存器的 SLAVE_MODE = 2, 复位模式。
- 当在 synci 上产生一个有效边沿跳变, 就会同时复位 TMR0 和 TMR1 的计数值。

一个 Timer 作为后面 Timer 的载波；具体例子如下：

- 把 TMR1 配置为 PWM 输出模式(低频);
- 配置 TMR1_CON 寄存器的 SYNCSEL, 选择 PWM_OUT 输出到 Synco。
- 配置 TMR2_CON 寄存器的 SLAVE_MODE = 3, 门控模式。
- 配置 TMR2 为 PWM 输出模式(高频(38KHz))。
- 配置 TMR1 和 TMR2 使能。
- 这样, 就能产生带载波的 PWM 信号。

Timer1 和 Timer2 级联并且 Timer2 的 PWM 输出作为红外发送载波时,可以选择使能 TMRx_CON[27] 使 PWM 输出 IO 在输出高电平时切换到高阻态。

17.1.4.1 控制寄存器 (TMRx CON)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|---|
| 31:28 | Reserved | - | - | - |
| 27 | IRPWM_HZ_EN | 1'b0 | RW | Timer2 发送的 PWM 载波只有低电平和高阻态 (只有 Timer2 有效) 0: 不使能 |

| | | | | |
|-------|-------------|------|----|--|
| | | | | 1: 使能 |
| 26 | TMR_DIR_EN | 1'b0 | RW | Timer 计数模式由捕获引脚电平决定递增计数或者递减计数 0: 不使能 1: 使能(只能在 MODE_SEL=0 时使能) |
| 25 | PWMPOL | 1'b0 | RW | PWM 极性选择 |
| 24:23 | Reserved | - | - | - |
| 22 | CAP2POL | 1'b0 | RW | 捕获事件 2 的极性选择 0: 上升沿 1: 下降沿 |
| 21 | CAP1POL | 1'b0 | RW | 捕获事件 1 的极性选择 0: 上升沿 1: 下降沿 |
| 20:19 | Reserved | - | - | - |
| 18 | CTRRST2 | 1'b0 | RW | 捕获事件 2 发生时, 计数值自动清零 |
| 17 | CTRRST1 | 1'b0 | RW | 捕获事件 1 发生时, 计数值自动清零 |
| 16 | Reserved | - | - | - |
| 15 | CAP_CNT | 1'b0 | RW | 捕获寄存器的数量 0: 捕获数据存储在 CAP1 1: 捕获数据存储在 CAP1 CAP2 |
| 14:13 | CAP_SEL | 2'b0 | RW | 捕获设置: b00: GPIO b01: 保留 b10: COMP0 b11: 保留 |
| 12:11 | SYNCOSEL | 2'b0 | RW | 输出同步信号选择 b00: CNT 值=PRD 值 b01: CNT 值=CMP 值 b10: 把 SYNCI 值输出到 SYNCO b11: PWM 输出赋值到 SYNCO |
| 10 | SYNCIPOL | 1'b0 | RW | Synci 极性取反 0: 不取反 1: 取反 |
| 9:8 | SLAVE_MODE | 2'b0 | RW | Synci 功能选择 b00: 不使能 b01: kick start b10: reset b11: gating |
| 7:5 | PSC | 3'b0 | RW | Timer 预分频设置 b000: 0 分频 b001: 2 分频 b010: 4 分频 b011: 8 分频 b100: 16 分频 b101: 32 分频 b110: 64 分频 b111: 128 分频 |
| 4:2 | INC_SRC_SEL | 3'b0 | RW | 计数源设置 |

| | | | | |
|-----|----------|------|----|--|
| | | | | b001: 内部低速 RC 除 2 时钟 (系统时钟为 RC 时不可选) b010: 内部高速 HIRC 除 2 时钟 (系统时钟为 HIRC 时不可选) b011: 外部晶振除 2 时钟(系统时钟为晶振时不可选) b100: Timer inc 引脚上升沿 b101: Timer inc 引脚下降沿 b110: Timer inc 引脚上升下降沿 其它: 系统时钟 |
| 1:0 | MODE_SEL | 2'b0 | RW | 定时器模式设置 b00: 计数模式 b01: PWM 模式 b10: 捕获模式 其它: 保留 |

17.1.4.2 控制使能寄存器 (TMRx_EN)

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---------------|
| 31:1 | Reserved | - | - | - |
| 0 | TMREN | 1'b0 | RW | TMR 使能信号, 高有效 |

17.1.4.3 中断使能寄存器 (TMRx_IE)

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|-----------|-------|----------|--------------------------------------|
| 31:10 | Reserved | - | - | - |
| 9 | DMA_FL_IE | 1'b0 | RW | DMA 缓冲全满中断使能 |
| 8 | DMA_HF_IE | 1'b0 | RW | DMA 缓冲半满中断使能 |
| 7 | SLAVE_IE | 1'b0 | RW | 从模式的触发模式或者复位模式中断使能 |
| 6 | CMP_IE | 1'b0 | RW | CNT 值等于 CMP 值时, 中断使能, 仅在 PWM 模式时有效 |
| 5 | PRD_IE | 1'b0 | RW | 计数值等于 PRD 值时, 中断使能, 仅在计数模式/PWM 模式时有效 |
| 4 | OVF_IE | 1'b0 | RW | 计数值溢出(hffff)时, 中断使能 |
| 3:2 | Reserved | - | - | - |
| 1 | CAP2_IE | 1'b0 | RW | 捕获事件 2 发生时, 中断使能 |
| 0 | CAP1_IE | 1'b0 | RW | 捕获事件 1 发生时, 中断使能 |

17.1.4.4 计数寄存器 (TMRx_CNT)

Address offset: 0x0C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:16 | Reserved | - | - | - |
| 15:0 | CNT | 16'h0 | RW | 计数寄存器 |

17.1.4.5 标志寄存器 (TMRx_FLG)

Address offset: 0x10

| Width | Name | Reset | Property | Description |
|-------|------------|-------|----------|------------------------------------|
| 31:10 | Reserved | - | - | - |
| 9 | DMA_FL_FLG | 1'b0 | RO | DMA 缓冲全满标志 |
| 8 | DMA_HF_FLG | 1'b0 | RO | DMA 缓冲半满标志 |
| 7 | SLAVE_FLG | 1'b0 | RO | 从模式发生标志(仅复位或触发) |
| 6 | CMP_FLG | 1'b0 | RO | 计数值等于 CMP 值标志, 仅在 PWM 模式时有效 |
| 5 | PRD_FLG | 1'b0 | RO | 计数值等于 PRD 值标志, 仅在计数模式/PWM 模式时有效 |
| 4 | OVF_FLG | 1'b0 | RO | 计数值溢出标志 |
| 3:2 | Reserved | - | - | - |
| 1 | CAP2_FLG | 1'b0 | RO | 捕获事件 2 发生标志 |
| 0 | CAP1_FLG | 1'b0 | RO | 捕获事件 1 发生标志 |

17.1.4.6 清除寄存器 (TMRx_CLR)

Address offset: 0x14

| Width | Name | Reset | Property | Description |
|-------|------------|-------|----------|-------------------|
| 31:10 | Reserved | - | - | - |
| 9 | DMA_FL_CLR | 1'b0 | WO | DMA 缓冲全满标志清除 |
| 8 | DMA_HF_CLR | 1'b0 | WO | DMA 缓冲半满标志清除 |
| 7 | SLAVE_CLR | 1'b0 | WO | 从模式发生标志清除 |
| 6 | CMP_CLR | 1'b0 | WO | 计数值等于 CMP 值标志清除 |
| 5 | PRD_CLR | 1'b0 | WO | 计数值等于 PRD 值标志清除 |
| 4 | OVF_CLR | 1'b0 | WO | 计数值溢出(0xFFFF)标志清除 |
| 3:2 | Reserved | - | - | - |
| 1 | CAP2_CLR | 1'b0 | WO | 捕获事件 2 发生标志清除 |
| 0 | CAP1_CLR | 1'b0 | WO | 捕获事件 1 发生标志清除 |

17.1.4.7 捕获寄存器 1/周期寄存器 (TMRx_CAP1)

Address offset: 0x18

| Width | Name | Reset | Property | Description |
|-------|----------|----------|----------|---------------------------------------|
| 31:16 | Reserved | - | - | - |
| 15:0 | CAP1/PR | 16'hffff | RW | 捕获模式: 捕获寄存器 1 定时模式、PWM 模式: 计数周期寄存器 |

17.1.4.8 捕获寄存器 2/比较寄存器(TMRx_CAP2)

Address offset: 0x1C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------------------------------|
| 31:16 | Reserved | - | - | - |
| 15: 0 | CAP2/CMP | 16'h0 | RW | 捕获模式: 捕获寄存器 2 定时模式、PWM 模式: 比较寄存器 |

17.1.4.9 DMA 控制寄存器 (TMR1_DCTL)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|-----------|-------|----------|---|
| 31:3 | Reserved | - | - | - |
| 2 | DMA_IR_EN | 1'b0 | RW | DMA IR 模式使能 0: 不使能 1: 使能 |
| 1 | DMA_LPBK | 1'b0 | RW | DMA 模式选择 0: 单次模式, 指定 DMA 长度完成之后, TMR 失能 1: 循环模式, 指定 DMA 长度完成之后, 重新从起始地址开始 |
| 0 | DMA_EN | 1'b0 | RW | DMA 使能 0: 不使能 1: 使能 |

17.1.4.10 DMA 地址寄存器 (TMR1_DADR)

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|-----------|-------|----------|-------------------|
| 31:18 | Reserved | - | - | - |
| 17: 0 | DMA_STADR | 18'h0 | RW | DMA 起始地址需要 4 字节对齐 |

17.1.4.11 DMA 长度寄存器 (TMR1_DLEN)

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:16 | Reserved | - | - | - |
| 15: 0 | DMA_LEN | 16'h0 | RW | DMA_IR_EN=0 时: DMA_LEN 代表需要 DMA 的 WORD 个数, 且 DMA (WORD)数量=DMA_LEN+1 DMA_IR_EN =1 时: DMA_LEN 代表需要红外自动发送的位数, 且此时 DMA(WORD)数量=(DMA_LEN/32)+1, 需要 自动发送的红外位数=DMA_LEN+1 捕获模式时: DMA_LEN 代表捕获并写到 SRAM 的数据个数, 且捕获数据个数=DMA_LEN+1 |

17.1.4.12 DMA 计数寄存器 (TMR1_DMACNT)

Address offset: 0x0C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:16 | Reserved | - | - | - |
| 15: 0 | DMA_CNT | 16'h0 | RO | 已完成的 DMA 个数 显示已完成的 DMA (WORD)个数,循环模式下会自动清 0 |

17.1.4.13 DMA 发送周期寄存器 (TMR1_DMAIR0)

Address offset: 0x10

| Width | Name | Reset | Property | Description |
|-------|------------|-------|----------|-------------------------|
| 31:16 | DMAIR0_CMP | 16'h0 | WR | DMA IR 模式发送数据 0 时对应的占空比 |
| 15: 0 | DMAIR0_PRD | 16'h0 | WR | DMA IR 模式发送数据 0 时对应的周期 |

17.1.4.14 DMA 发送周期寄存器 (TMR1_DMAIR1)

Address offset: 0x14

| Width | Name | Reset | Property | Description |
|-------|------------|-------|----------|-------------------------|
| 31:16 | DMAIR1_CMP | 16'h0 | WR | DMA IR 模式发送数据 1 时对应的占空比 |
| 15: 0 | DMAIR1_PRD | 16'h0 | WR | DMA IR 模式发送数据 1 时对应的周期 |

17.1.4.15 同步寄存器 (TMR_ALLCON)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|-----------|-------|----------|--------------|
| 31:14 | Reserved | - | - | - |
| 13 | TMR5_SYNC | 1'b0 | WO | Timer5 计数值清零 |
| 12 | Reserved | - | - | - |
| 11 | TMR3_SYNC | 1'b0 | WO | Timer3 计数值清零 |
| 10 | TMR2_SYNC | 1'b0 | WO | Timer2 计数值清零 |
| 9 | TMR1_SYNC | 1'b0 | WO | Timer1 计数值清零 |
| 8 | TMR0_SYNC | 1'b0 | WO | Timer0 计数值清零 |
| 7:6 | Reserved | - | - | - |
| 5 | TMR5_KICK | 1'b0 | WO | Timer5 启动计数 |
| 4 | Reserved | - | - | - |
| 3 | TMR3_KICK | 1'b0 | WO | Timer3 启动计数 |
| 2 | TMR2_KICK | 1'b0 | WO | Timer2 启动计数 |
| 1 | TMR1_KICK | 1'b0 | WO | Timer1 启动计数 |
| 0 | TMR0_KICK | 1'b0 | WO | Timer0 启动计数 |

17.1.4.16 寄存器地址 mapping

| Module base | Register name | Offset |
|--|---------------|--------|
| TMR0: 0x4000 1100 TMR1: 0x4000 1200 TMR2: 0x4000 1300 TMR3: 0x4000 1400 TMR4: 0x4001 0100 TMR5: 0x4000 1600 | TMRx_CON | 0x00 |
| | TMRx_EN | 0x04 |
| | TMRx_IE | 0x08 |
| | TMRx_CNT | 0x0C |
| | TMRx_FLG | 0x10 |
| | TMRx_CLR | 0x14 |
| | TMRx_CAP1 | 0x18 |
| | TMRx_CAP2 | 0x1C |
| TMR1_DMA: 0x4000 1128 | TMRx_DCTL | 0x00 |
| | TMR1_DADR | 0x04 |
| | TMR1_DLEN | 0x08 |
| | TMR1_DMACNT | 0x0C |
| | TMR1_DMAIR0 | 0x10 |

| | | |
|-------------------------|-------------|------|
| | TMR1_DMAIR1 | 0x14 |
| TMR_ALLCON: 0x4000 1128 | TMR_ALLCON | 0x00 |

珠海巨晟科技股份有限公司

18 看门狗 (WDT)

18.1 简介

JS32L010 内置看门狗，为系统提供了更高的安全性、时间的精确性和使用的灵活性。看门狗可用于检测和解决软件错误引起的故障，当计数值超出阈值时，产生系统复位或者中断，保证系统的安全性。看门狗由专门的低速时钟驱动，即使主时钟发生故障它也不受影响。支持修改分频系数改变超时时间，发生超时事件的时候可以选择中断或者复位。

18.2 模块框图

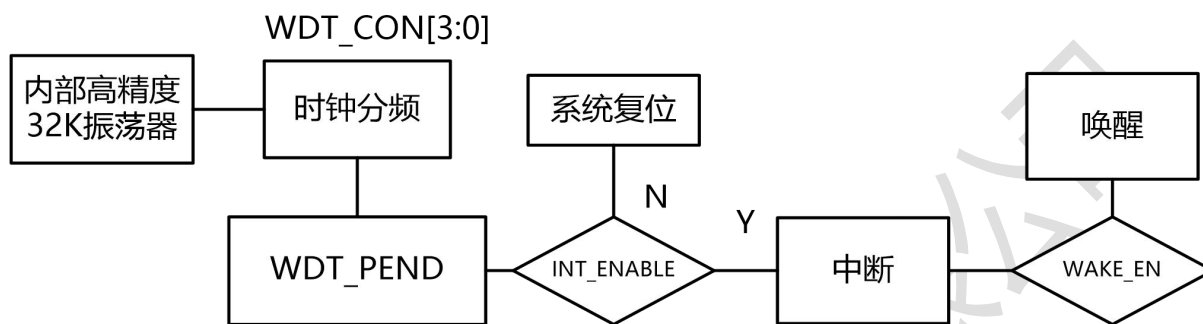


图 18-1 看门狗模块框图

18.3 寄存器描述

18.3.1 WDT_CON

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|------------|---------|----------|--|
| 31:10 | Reserved | - | - | - |
| 9 | WAKE_EN | 1'b0 | RO | 唤醒使能 |
| 8:7 | Reserved | - | - | - |
| 6 | WDT_PEND | 1'b0 | RO | 看门狗计数满标志 WDG_KEY 写 0xAAAA，清除 WDT_PEND |
| 5 | INT_ENABLE | 1'b0 | RO | 中断使能 0: 计数满时复位系统 1: 计数满时产生中断 |
| 4 | WDT_EN_STA | 1'b1 | RO | 写 WDT_KEY=0xC000，置位。 写 WDT_KEY=0xD000，复位。 |
| 3:0 | WDT_PSR | 4'b1000 | RW | 分频系数，每次配置该位域之前必须先写 WDG_KEY=0x5555 b0000: 不分频 b0001: 2 b0010: 4 b0011: 8 b0100: 16 b0101: 32 b0110: 64 b0111: 128 b1000: 256 b1001: 512 |

| | | | | |
|--|--|--|--|--|
| | | | | b1010: 1024 b1011: 2048 b1100: 4096 b1101: 8192 b1110: 16384 b1111: 32768 看门狗复位时间=1/32K*256*分频系数 |
|--|--|--|--|--|

18.3.2 WDT_KEY

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:16 | Reserved | - | - | - |
| 15:0 | KEY | 16'h0 | WO | KEY[15: 0]: 键值(只写寄存器, 读出值为 0x0000)(KEY 值) 软件必须以一定的间隔写入 0xAAAA, 否则, 当计数器为 0 时, 看门狗会产生复位。当 WDT_PEND 为 1 的时候, 写 0xAAAA 会清除 WDT_PEND。 写入 h5555 表示允许访问 WDT_PSR 写入 hCCCC, 启动看门狗工作 写入 hDDDD, 关闭看门狗 写入 hAAAA, 清除 WDT_PEND 写入 h55AA, 开启中断使能 写入 hAA55, 关闭中断使能 写入 h5A5A, 开启 WAKE_EN 写入 hA5A5, 关闭 WAKE_EN |

19 系统寄存器 (SYSTEM REG)

19.1 系统寄存器基地址表

| | |
|-----------------|-------------|
| System Register | |
| Base address | 0x4002 0000 |

表 19-1 系统寄存器列表

| Offset | Register Name | Description |
|--------|---------------|---------------|
| 0x0000 | SYS_KEY | 密匙控制寄存器 |
| 0x0004 | SYS_CON0 | 系统控制寄存器 0 |
| 0x0008 | SYS_CON1 | 系统控制寄存器 1 |
| 0x000C | SYS_CON2 | 系统控制寄存器 2 |
| 0x0010 | SYS_CON3 | 系统控制寄存器 3 |
| 0x0014 | SYS_CON4 | 系统控制寄存器 4 |
| 0x0018 | SYS_CON5 | 系统控制寄存器 5 |
| 0x001C | SYS_CON6 | 系统控制寄存器 6 |
| 0x0020 | SYS_CON7 | 系统控制寄存器 7 |
| 0x0024 | CLKCON0 | 时钟控制寄存器 0 |
| 0x0028 | CLKCON1 | 时钟控制寄存器 1 |
| 0x002C | CLKCON2 | 时钟控制寄存器 2 |
| 0x0030 | CLKCON3 | 时钟控制寄存器 3 |
| 0x0034 | CLKCON4 | 时钟控制寄存器 4 |
| 0x0038 | CLKCON5 | 时钟控制寄存器 5 |
| 0x003C | CLKCON6 | 时钟控制寄存器 6 |
| 0x0040 | CLKCON7 | 时钟控制寄存器 7 |
| 0x0044 | HOSC_MNT | 晶振起振失败监控寄存器 |
| 0x0048 | SYS_ERR | 系统出错记录寄存器 |
| 0x004C | WKUP_CON | 唤醒控制寄存器 |
| 0x0050 | LP_CON | 低功耗控制寄存器 |
| 0x005C | CHIPID_DCN | CHIP ID 和 DCN |
| 0x0060 | MODE | 特殊模式记录寄存器 |
| 0x0064 | PMUCON0 | PMU 控制寄存器 |
| 0x0068 | RPCON | 系统暂挂记录寄存器 |
| 0x0078 | WKUP_CON1 | 唤醒控制寄存器 1 |

19.2 寄存器描述

19.2.1 SYS_KEY

Address offset = 0x00

| Width | Name | Reset | Property | Description |
|-------|---------|------------------|----------|--|
| 31:0 | SYS_KEY | 32'h0000 0001 | RW | 写入 0X3FAC87E4 以使能所有系统寄存器写操作,写入除了这个数外都会将这个 BIT 置 0。读取返回 SYS_KEY 状态 0: 锁定所有系统寄存器写操作 |

1: 解锁所有系统寄存器写操作

19.2.2 SYS_CON0

Address offset = 0x04

| Width | Name | Reset | Property | Description |
|-------|----------------|-------|----------|---|
| 31 | FAST_RST_EN | 1'b0 | RW | 快速解除复位使能 0: 不使能 1: 使能 |
| 30 | SLEEP_GOON_EN | 1'b0 | RW | 当端口唤醒时不复位 0: 不使能 1: 使能 |
| 29:27 | SLEEP_DLY_CNT | 3'b0 | RW | 端口延时 n 个系统时钟后从睡眠模式唤醒 |
| 26 | DBS_SOFT_RST | 1'b1 | RW | 0: 复位端口消抖配置寄存器 1: 完成复位 |
| 25 | CRC_SOFT_RST | 1'b1 | RW | 0: 复位 CRC 校验相关配置寄存器 1: 完成复位 |
| 24:21 | Reserved | - | - | - |
| 20 | GPIO_SOFT_RST | 1'b1 | RW | 0: 复位 GPIOA/GPIOB/GPIOC/GPIOD/GPIOE 配置寄存器 1:完成复位 |
| 19 | ADKEY_SOFT_RST | 1'b1 | RW | 0: 复位 ADC 配置寄存器. 1: 完成复位. |
| 18 | Reserved | - | - | - |
| 17 | UART0_SOFT_RST | 1'b1 | RW | 0: 复位 UART0 相关配置寄存器. 1: 完成复位. |
| 16 | Reserved | - | - | - |
| 15 | SPI0_SOFT_RST | 1'b1 | RW | 0: 复位 SPI0 相关配置寄存器. 1: 完成复位. |
| 14 | Reserved | - | - | - |
| 13 | RTCC_SOFT_RST | 1'b1 | RW | 0: 复位 RTC 相关寄存器 1: 完成复位 |
| 12 | LCD_SOFT_RST | 1'b1 | RW | 0: 复位 LCD 控制器相关控制器 1: 完成复位 |
| 11:7 | Reserved | - | - | - |
| 6 | TIMER_SOFT_RST | 1'b1 | RW | 0: 复位 TIMERS 相关配置寄存器. 1: 完成复位. |
| 5:0 | Reserved | - | - | - |

19.2.3 SYS_CON1

Address offset = 0x08

| Width | Name | Reset | Property | Description |
|-------|------------------|-------|----------|-------------------------------|
| 31:18 | Reserved | - | - | - |
| 17 | Reserved | - | - | - |
| 16 | Reserved | - | - | - |
| 15 | UART0_ML_WKUP_EN | 1'b0 | RW | UART0 唤醒使能 0: 不使能 1: 使能 |

| | | | | |
|----|------------------|------|----|--|
| 14 | Reserved | - | - | - |
| 13 | UART0_UD_ENABLE | 1'b1 | RW | UART0 更新模式 0: 不使能 1: 使能 |
| 12 | DEBUG_EN | 1'b0 | RW | 在调试期间停用一些功能(睡眠模式,时钟停止模式,看门狗复位) 0: 停用 1: 允许使用 |
| 11 | INT_REMAP_EN | 1'b0 | RW | 从 EFLASH 映射中断入口地址到 SRAM 0-192Bytes 0: 中断入口地址在 EFALSH 1: 中断入口地址在 SRAM |
| 10 | NMI_INV_SEL | 1'b0 | RW | NMI 引脚极性反向选择 0: 高电平引脚触发 NMI 1: 低电平引脚触发 NMI |
| 9 | Reserved | - | - | - |
| 8 | SWD_EN | 1'b1 | RW | SWD 使能 0: 不使能 1: 使能 |
| 7 | LVDVCC_WKUP_EN | 1'b0 | RW | LVD_VCC 低电唤醒使能 0: 不使能 1: 使能 |
| 6 | Reserved | - | - | - |
| 5 | SYS_ERR_RESP_EN | 1'b0 | RW | 系统总线访问超出内存空间的区域返回错误响应 0: 不使能 1: 使能 |
| 4 | SYS_ERR_INT_EN | 1'b0 | RW | 系统总线访问超出内存空间的区域触发 NMI 中断 0: 不使能 1: 使能 |
| 3 | HOSC_LOSS_NMI_EN | 1'b0 | RW | 监控 XOSC 丢失触发 NMI 中断 0: 不使能 1: 使能 |
| 2 | RXEV_ENABLE | 1'b0 | RW | PB2 作为接收方事件发送到 CPU 0: 不使能 1: 使能 |
| 1 | NMI_INT_ENABLE | 1'b0 | RW | PB3 作为 CPU 的外部 NMI 输入 0: 不使能 1: 使能 |
| 0 | LOCKUP_ENABLE | 1'b0 | RW | 系统锁定触发系统复位 0: 不使能 1: 使能 |

19.2.4 **SYS_CON2**

Address offset = 0x0C

| Width | Name | Reset | Property | Description |
|-------|-----------|-------|----------|----------------------------------|
| 31:16 | PB_DEB_EN | 16'h0 | RW | Port B 消抖寄存器 [15:0] for PB15-PB0 |
| 15:0 | PA_DEB_EN | 16'h0 | RW | Port A 消抖寄存器 [15:0] for PA15-PA0 |

19.2.5 SYS_CON3

Address offset = 0x10

| Width | Name | Reset | Property | Description |
|-------|-----------|-------|----------|----------------------------------|
| 31:25 | Reserved | - | - | - |
| 24:22 | PE_DEB_EN | 3'b0 | RW | Port E 消抖寄存器 [2:0] for PE2-PE0 |
| 21:16 | PD_DEB_EN | 6'h0 | RW | Port D 消抖寄存器 [5:0] for PD5-PD0 |
| 15:0 | PC_DEB_EN | 16'h0 | RW | Port C 消抖寄存器 [15:0] for PC15-PC0 |

19.2.6 SYS_CON4

Address offset = 0x14

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:0 | Reserved | - | - | - |

19.2.7 SYS_CON5

Address offset = 0x18

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:0 | Reserved | - | - | - |

19.2.8 SYS_CON6

Address offset = 0x1C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:0 | Reserved | - | - | - |

19.2.9 SYS_CON7

Address offset = 0x20

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:0 | Reserved | - | - | - |

19.2.10 CLKCON0

Address offset = 0x24

| Width | Name | Reset | Property | Description |
|-------|------------------|-------|----------|---|
| 31:25 | Reserved | - | - | - |
| 24:23 | UART0_LP_CLK_SEL | 2'b0 | RW | UART0 低功耗时钟选择 b00: apb0 时钟 b01: 内部高速 26MHz b10: 内部高精度 32KHz b11: 外部晶振 |
| 22 | SYSCLK_HJDSEL | 1'b0 | RW | 系统选择内部高精度 32KHz 0: 不选择内部高精度 32KHz 1: 选择内部高精度 32KHz |
| 21:20 | LCD_CLK_SEL | 2'b0 | RW | LCD 时钟选择 b00: 内部高精度 32KHz b01: 外部晶振 b10: 内部低速 128KHz b11: 内部高精度 32KHz |
| 19:18 | RTCC_CLK_SEL | 2'b0 | RW | RTC 时钟选择 b00: 内部高精度 32KHz b01: 外部晶振 b10: 保留 |

| | | | | |
|-------|-----------------|------|----|---|
| | | | | b11: 保留 |
| 17:16 | LVD_DEBCLK_SEL | 2'b0 | RW | 电源控制器时钟选择 b00: apb0 时钟 b01: 内部高精度 32KHz b10: 内部 128KHz b11: 内部 128KHz |
| 15:14 | GPIOE_DBS_SEL | 2'b0 | RW | Port E 消抖工作时钟 b00: 内部高速 26MHz 分频 b01: 外部晶振 b10: 系统时钟 b11: 内部 128K |
| 13:12 | GPIOD_DBS_SEL | 2'b0 | RW | Port D 消抖工作时钟 b00: 内部高速 26MHz 分频 b01: 外部晶振 b10: 系统时钟 b11: 内部 128KHz |
| 11:10 | GPIOC_DBS_SEL | 2'b0 | RW | Port C 消抖工作时钟 b00: 内部高速 26MHz 分频 b01: 外部晶振 b10: 系统时钟 b11: 内部 128K |
| 9:8 | GPIOB_DBS_SEL | 2'b0 | RW | Port B 消抖工作时钟 b00: 内部高速 26MHz 分频 b01: 外部晶振 b10: 系统时钟 b11: 内部 128K |
| 7:6 | GPIOA_DBS_SEL | 2'b0 | RW | Port A 消抖工作时钟 b00: 内部高速 26MHz 分频 b01: 外部晶振 b10: 系统时钟 b11: 内部 128KHz |
| 5:4 | PLL_REF_CLK_SEL | 2'b0 | RW | PLL 参考时钟设置 b00: 内部低速高精度 32KHz 时钟 b01: 内部高速 26MHz 时钟 b10: 晶振 b11: 保留 |
| 3:2 | Reserved | - | - | - |
| 1:0 | SYS_CLK_SEL | 2'b0 | RW | 系统时钟选择 b00: 内部 128KHz b01: 外部晶振或者高精度 32KHz 时钟(由选择 CLKCON0[22]) b10: 内部高速 26MHz b11: 片内高速 PLL 时钟 |

19.2.11 CLKCON1

Address offset = 0x28

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31 | Reserved | - | - | - |

| | | | | |
|-------|---------------|------|----|---|
| 30:28 | CLK_TO_IO_DIV | 3'b0 | RW | 时钟到 IO 分频 b000: 1 分频 b001: 2 分频 b010: 3 分频 b011: 4 分频 b100: 5 分频 b101: 6 分频 b110: 7 分频 b111: 关闭时钟选择 |
| 27:24 | HIRC_CLK_DIV | 4'b0 | RW | 内部高速 26M 分频 b0000: 1 分频 b0001: 2 分频 ... b1110: 15 分频 b1111: 关闭时钟选择 |
| 23:19 | Reserved | - | - | - |
| 18:16 | PLL_CLK_DIV | 3'b0 | RW | PLL 时钟分频 b000: 1 分频 b001: 2 分频 ... b110: 7 分频 b111: 关闭 |
| 15:12 | APB1CLK_DIV | 4'b0 | RW | APB1 时钟分频 b0000: 1 分频 b0001: 2 分频 ... b1110: 15 分频 b1111: 关闭 apb1_clk |
| 11:8 | APB0CLK_DIV | 4'b0 | RW | APB0 时钟分频 b0000: 1 分频 b0001: 2 分频 ... b1110: 15 分频 b1111: 关闭 apb0_clk |
| 7:0 | SYSClk_DIV | 8'h0 | RW | 系统时钟分频 h00: 1 分频 h01: 2 分频 ... hFE: 255 分频 hFF: 关闭系统时钟 |

19.2.12 CLKCON2

Address offset = 0x2C

| Width | Name | Reset | Property | Description |
|-------|-------------------|-------|----------|---|
| 31 | Reserved | - | - | - |
| 30 | CLK_SOURCE_EN_BPS | 1'b1 | RW | 对 HXOSC 和 PLL 使能信号滤波 0: 不滤波 1: 滤波 |

| | | | | |
|-------|-------------------|------|----|-------------------------------------|
| 29 | Reserved | - | - | - |
| 28 | Reserved | - | - | - |
| 27 | COMP_CLK_EN | 1'b1 | RW | 使能比较器时钟 0: 不使能 1: 使能 |
| 26 | LCD_LP_CLK_EN | 1'b1 | RW | 使能 LCD_LP_CLK 0: 不使能 1: 使能 |
| 25 | CRC_CLK_EN | 1'b1 | RW | 使能 CRC 时钟 0: 不使能 1: 使能 |
| 24 | EFLASH_MEM_CLK_EN | 1'b1 | RW | 使能 FLASH 擦除/编程时钟 0: 不使能 1: 使能 |
| 23:18 | Reserved | - | - | - |
| 17 | UART0_CLK_EN | 1'b1 | RW | 使能 UART0 时钟 0: 不使能 1: 使能 |
| 16 | Reserved | - | - | - |
| 15 | SPI0_CLK_EN | 1'b1 | RW | 使能 SPI0 时钟 0: 不使能 1: 使能 |
| 14 | Reserved | - | - | - |
| 13 | TIMER5_CLK_EN | 1'b1 | RW | 使能 TIMER5 时钟 0: 不使能 1: 使能 |
| 12 | TIMER3_CLK_EN | 1'b1 | RW | 使能 TIMER3 时钟 0: 不使能 1: 使能 |
| 11 | TIMER2_CLK_EN | 1'b1 | RW | 使能 TIMER2 时钟 0: 不使能 1: 使能 |
| 10 | TIMER1_CLK_EN | 1'b1 | RW | 使能 TIMER1 时钟 0: 不使能 1: 使能 |
| 9 | TIMER0_CLK_EN | 1'b1 | RW | 使能 TIMER0 时钟 0: 不使能 1: 使能 |
| 8 | WDT_CLK_EN | 1'b1 | RW | 使能 watchdog 时钟 0: 不使能 1: 使能 |
| 7 | WDT_SYS_CLK_EN | 1'b1 | RW | 使能 watchdog 系统时钟 0: 不使能 1: 使能 |
| 6 | TIMER4_CLK_EN | 1'b1 | RW | 使能 TIMER4 时钟 0: 不使能 1: 使能 |
| 5 | RTCC_CLK_EN | 1'b1 | RW | 使能 RTC 时钟 |

| | | | | |
|-----|--------------|------|----|-------------------------------|
| | | | | 0: 不使能 1: 使能 |
| 4:3 | Reserved | - | - | - |
| 2 | SRAM0_CLK_EN | 1'b1 | RW | 使能 SRAM 时钟 0: 不使能 1: 使能 |
| 1 | AHB1_CLK_EN | 1'b1 | RW | 使能 AHB1 时钟 0: 不使能 1: 使能 |
| 0 | AHB0_CLK_EN | 1'b1 | RW | 使能 AHB0 时钟 0: 不使能 1: 使能 |

19.2.13 CLKCON3

Address offset = 0x30

| Width | Name | Reset | Property | Description |
|-------|------------------|--------|----------|--|
| 31 | HIRC_OSC_EN_FLAG | 1'b1 | RO | HIRC_OSC 使能标志 0: 不使能 1: 使能 |
| 30:29 | Reserved | - | - | - |
| 28 | Reserved | - | - | - |
| 27 | Reserved | - | - | - |
| 26:19 | HIRC_OSC_SC | 8'h0 | RW | HIRC_OSC 频率控制: h00: 最低 hFF: 最高 |
| 18 | HIRC_OSC_LDOS | 1'b0 | RW | HIRC_OSC LDO 电压设置: 0: 1.5V 1: 1.6V |
| 17 | HIRC_OSC_EN | 1'b1 | RW | HIRC_OSC 使能信号: 0: 不使能 1: 使能 |
| 16 | LXOSC_TSCKEN | 1'b0 | RW | LXOSC 支持从 XOSCI 输入时钟测试 0: 不使能 1: 使能 |
| 15 | LXOSC_TRIGEN | 1'b0 | RW | LXOSC 启动触发时钟脉冲 0: 不使能 1: 使能 |
| 14 | LXOSC_RESFB | 1'b0 | RW | LXOSC 反馈电阻选择 0: 8.5M 1: 5.3M |
| 13 | Reserved | - | - | - |
| 12:10 | LXOSC_ICCS | 3'b111 | RW | LXOSC 电流驱动能力信号 b000: 470nA b001: 630nA b010: 780nA b011: 940nA b100: 1100nA b101: 1250nA |

| | | | | |
|-----|-------------|---------|----|--|
| | | | | b110: 1410nA b111: 1570nA |
| 9:7 | LXOSC_DRVS | 3'b111 | RW | LXOSC 驱动能力控制信号: b000: gm b001: 2*gm b010: 3*gm b011: 4*gm b100: 5*gm b101: 6*gm b110: 7*gm b111: 8*gm |
| 6 | LXOSC_CLAMS | 1'b1 | RW | LXOSC OSCO/OSCI 限幅控制 0: 限制到 1.2v 1: 限制到 1.5v |
| 5 | LXOSC_CLAMP | 1'b0 | RW | LXOSC OSCO/OSCI 限幅控制信号 0: 不使能 1: 使能 |
| 4:1 | LXOSC_CAPS | 4'b1000 | RW | LXOSC 内置电容选择: b0000: 5.9pF b0001: 6.9pF b0010: 7.9pF b0011: 8.9pF b0100: 9.9pF b0101: 10.9pF b0110: 11.9pF b0111: 12.9pF b1000: 13.9pF b1001: 14.9pF b1010: 15.9pF b1011: 16.9pF b1100: 17.9pF b1101: 18.9pF b1110: 19.9pF b1111: 20.9pF |
| 0 | LXOSC_EN | 1'b0 | RW | LXOSC 使能信号; 0: 不使能 1: 使能 |

19.2.14 CLKCON4

Address offset = 0x34

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|--|
| 31:28 | LXOSC_XOCFT | 4'b0 | RW | LXOSC XOSCO 终端负载电容微调信号 b0000: 0pF b0001: 0.2pF b0010: 0.4pF b0011: 0.6pF b0100: 0.8pF b0101: 1.0pF |

| | | | | |
|-------|--------------|-------|----|--|
| | | | | b0110: 1.2pF b0111: 1.4pF b1000: 1.6pF b1001: 1.8pF b1010: 2.0pF b1011: 2.2pF b1100: 2.4pF b1101: 2.6pF b1110: 2.8pF b1111: 3.0pF |
| 27:24 | LXOSC_XICFT | 4'b0 | RW | LXOSC XOSCI 终端负载电容微调信号 b0000: 0pF b0001: 0.2pF b0010: 0.4pF b0011: 0.6pF b0100: 0.8pF b0101: 1.0pF b0110: 1.2pF b0111: 1.4pF b1000: 1.6pF b1001: 1.8pF b1010: 2.0pF b1011: 2.2pF b1100: 2.4pF b1101: 2.6pF b1110: 2.8pF b1111: 3.0pF |
| 23:11 | Reserved | - | - | - |
| 10:9 | HJDLIRC_TCPS | 2'b01 | RW | 温度补偿控制, 包装后用温度测定, 补偿范围 1~2% |
| 8:1 | HJDLIRC_SC | 8'h9F | RW | LIRCOSC 频率控制位, 用于在量产测试中校准, 校准后常温下精度高于 1%. |
| 0 | HJDLIRC_EN | 1'b1 | RW | 高精度 32k 时钟使能 : 0: 不使能 1: 使能 |

19.2.15 CLKCON5

Address offset = 0x38

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:0 | Reserved | - | - | - |

19.2.16 CLKCON6

Address offset = 0x3C

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:0 | Reserved | - | - | - |

19.2.17 CLKCON7

Address offset = 0x40

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:0 | Reserved | - | - | - |

19.2.18 HOSC_MNT

Address offset = 0x44

| Width | Name | Reset | Property | Description |
|-------|-------------------|-------|----------|--|
| 31:16 | Reserved | - | - | - |
| 15 | HOSC_MNT_EN | 1'b0 | RW | 使能 XOSC 工作监控 0: 不使能 1: 使能 |
| 14 | HOSC_LOSS_PENDING | 1'b0 | RW | READ :外部 XOSC 出错状态标志 WRITE: 0: 清除标志 1: 不采取操作 |
| 13 | HOSC_LOSS_SW_EN | 1'b0 | RW | 使能当检测到外部 XOSC 故障时, 硬件自动将 XOSC 时钟从 XOSC 切换到 LIRC 0: 不自动切换 1: 自动切换 |
| 12:5 | Reserved | - | - | - |
| 4:0 | XOSC_LOSS_LIMIT | 5'h1C | RW | 外部晶振失效时间限制: 1/128k * XOSC_LOSS_LIMIT 时钟后将触发 XOSC_LOSS_PENDING |

19.2.19 SYS_ERR

Address offset = 0x48

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:2 | Reserved | - | - | - |
| 1 | CLK_ERR | 1'b0 | RW | 时钟使用出错状态. 1: 时钟使用发生错误时, 此位可由硬件设置。清除由软件 0: 时钟使用没有错误 |
| 0 | SYS_ERR0 | 1'b0 | RW | 指示出侧存储器的总线操作的出错标志。如果设置 SYS_ERR_INT_EN, 这个标志将使 NMI 中断发生 |

表 19-1 Clk_err 触发条件

| Clk_err 触发条件 | | | |
|--|---------------|----------------|-----------------------|
| | 片内 128K 时钟关使能 | 片内 26M 高速时钟关使能 | 外部晶振关使能或者外部晶振 Ido 关使能 |
| SYS_CLK_SEL 使用片内 128K 时钟 | √ | - | - |
| SYS_CLK_SEL 使用片内 26M 高速时钟 | - | √ | - |
| SYS_CLK_SEL 使用外部晶振 | - | - | √ |
| SYS_CLK_SEL 使用片内高速 PLL 时钟同时 PLL_REF_SEL 使用片内 128K 时钟 | √ | - | - |
| SYS_CLK_SEL 使用片内高速 PLL 时钟同时 PLL_REF_SEL 使用片内 26M 高速时钟 | - | √ | - |
| SYS_CLK_SEL 使用片内高速 PLL 时钟 | - | - | √ |

| | | | |
|-----------------------|--|--|--|
| 同时 PLL_REF_SEL 使用外部晶振 | | | |
|-----------------------|--|--|--|

19.2.20 WKUP_CON

Address offset = 0x4C

| Width | Name | Reset | Property | Description |
|-------|-----------|-------|----------|-------------------------------------|
| 31:28 | Reserved | - | - | - |
| 27:16 | WKUP_EDGE | 12'h0 | RW | 选择要监视所选外部端口的边缘 0: 上升沿 1: 下降沿 |
| 15:12 | Reserved | - | - | - |
| 11:0 | WKUP_EN | 12'h0 | RW | 使能选择要监视所选外部端口的边缘 0: 不使能 1: 使能 |

19.2.21 LP_CON

Address offset = 0x50

| Width | Name | Reset | Property | Description |
|-------|--------------------------|-------|----------|---|
| 31:11 | Reserved | - | - | - |
| 10 | XOSC_AUTO_DIS | 1'b0 | RW | 使能睡眠模式时，外部晶振自动关闭 0: 不使能 1: 使能 |
| 9 | HJD_RC32_AUTO_DIS | 1'b0 | RW | 使能睡眠模式时，32.768KHz 时钟自动关闭 0: 不使能 1: 使能 |
| 8 | PMU_LP_SW_EN | 1'b0 | RW | 使能后，PMU_HPLDO、PMU_VI2EN 、 PMU_PUULDI_15V 关闭 0: 不使能 1: 使能 |
| 7 | PMU_LP_HW_EN | 1'b0 | RW | 使能 PMU_HPLDO、PMU_VI2EN 、 PMU_PUULDI_15V 睡眠时自动关闭 0: 不使能 1: 使能 |
| 6 | HJD_RC32_AUTO_ENAB LE | 1'b0 | RW | 当外部晶振监控丢失事件发生时，自动开启内部高精度 32.768KHz 时钟 0: 不使能 1: 使能自动打开内部高精度 32.768KHz 时钟 |
| 5 | RC32K_SOFT_EN | 1'b1 | RW | 使能 LIRC_128K 0: 不使能 1: 使能 |
| 4 | RC32K_AUTO_DIS | 1'b0 | RW | 进入睡眠模式时自动关闭 LIRC_128KHz 0: 不使能 1: 使能 |
| 3 | HIRC_AUTO_DIS | 1'b0 | RW | 进入睡眠模式或时钟停止模式时自动关闭 HIRC 26MHz 0: 不使能 1: 使能 |
| 2 | SRAM0_AUTO_DIS | 1'b0 | RW | 自动关闭 SRAM CE 当进入时钟停止或睡眠模式 0: 不使能 |

| | | | | |
|---|---------|------|----|---|
| | | | | 1: 使能 |
| 1 | STOPCLK | 1'b0 | RW | 时钟停止模式 0: 未进入时钟停止模式 1: 进入时钟停止模式模式 |
| 0 | SLEEP | 1'b0 | RW | 睡眠模式 0: 未进入睡眠模式 1: 进入睡眠模式 |

19.2.22 CHIPID_DCN

Address offset = 0x5C

| Width | Name | Reset | Property | Description |
|-------|----------|----------|----------|-------------|
| 31:24 | Reserved | - | - | - |
| 23:16 | CHIP_DCN | 8'h0 | RO | 设计变更版本说明 |
| 15:0 | CHIP_ID | 16'h2000 | RO | 芯片 ID 编号 |

19.2.23 MODE

Address offset = 0x60

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:0 | Reserved | - | - | - |

19.2.24 PMUCON0

Address offset = 0x64

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|--|
| 31:21 | Reserved | - | - | - |
| 20 | PMU_V2IEN | 1'b1 | RW | 模拟模块偏置电流使能 0: 不使能 1: 使能 |
| 19 | PMU_TSEN | 1'b0 | RW | 温度传感器使能 0: 不使能 1: 使能 |
| 18 | PMU_PULLDLI | 1'b0 | RW | VDD LDO 0.12mA 下拉电流设置; 仅仅在正常功耗 LDO 使能时起作用 0: 无效 1: 0.12mA |
| 17 | PMU_PULLDI | 1'b1 | RW | VDD LDO 2.2mA 下拉电流; 仅仅在正常功耗 LDO 使能时起作用 0: 无效 1: 2.2mA |
| 16 | PMU_OCPEN | 1'b1 | RW | VDD LDO 过流保护使能 0: 不使能 1: 使能 |
| 15:14 | PMU_LPBGI | 2'b0 | RW | 电流量产功耗校准调整 b00: 160nA b01: 100nA b10: 160nA b11: 80nA |
| 13 | PMU_HPLDO | 1'b1 | RW | 正常功耗 LDO 使能 0: 不使能 1: 使能 |

| | | | | |
|------|-----------|---------|----|--|
| 12 | PMU_BGBUF | 1'b1 | RW | 带隙电压缓冲器使能 0: 不使能 1: 使能 |
| 11:8 | PMU_LPBGS | 4'b0111 | RW | 量产校准控制带隙电压 b0000: 1.154v b1000: 1.211v b0001: 1.161v b1001: 1.218v b0010: 1.168v b1010: 1.226v b0011: 1.175v b1011: 1.232v b0100: 1.183v b1100: 1.239v b0101: 1.189v b1101: 1.246v b0110: 1.196v b1110: 1.254v b0111: 1.204v b1111: 1.261v |
| 7:4 | PMU_LLDOS | 4'b0 | RW | 低功耗 LDO 电压设置: b0000: 1.50v b0001: 1.30v b0010: 1.35v b0011: 1.40v b0100: 1.45v b0101: 1.25v b0110: 1.55v b0111: 1.60v b1000: 1.65v b1001: 1.70v b1010: 1.75v b1011: 1.80v 其它: 保留 |
| 3:0 | PMU_LDOS | 4'b0 | RW | 正常 LDO 电压设置: b0000: 1.50v b0001: 1.30v b0010: 1.35v b0011: 1.40v b0100: 1.45v b0101: 1.25v b0110: 1.55v b0111: 1.60v b1000: 1.65v b1001: 1.70v b1010: 1.75v b1011: 1.80v |

注：使用低速时钟作系统时钟时，想改变 LDO 电压需要同时设置 PMU_LDOS 和 PMU_LLDOS

19.2.25 RPCON

Address offset = 0x68

| Width | Name | Reset | Property | Description |
|-------|------------------|-------|----------|---------------------|
| 31:20 | Reserved | - | - | - |
| 19 | UART0_RESET_CLR | 1'b0 | WO | 写 1 清除 UART0 更新复位标志 |
| 18 | LOCKUP_RESET_CLR | 1'b0 | WO | 写 1 清除锁定复位标志 |
| 17 | SOFT_RESET_CLR | 1'b0 | WO | 写 1 清除软件复位标志 |

| | | | | |
|------|----------------------|------|----|--|
| 16 | SLEEP_STA_CLR | 1'b0 | WO | 写 1 清除睡眠标志 |
| 15:4 | Reserved | - | - | - |
| 3 | UART0_UPDATE_PENDING | 1'b0 | RO | UART0 更新事件发生标志 |
| 2 | LOCK_RESET_PENDING | 1'b0 | RO | 当发生锁定复位时, 这个位将记录复位标志直到 CPU 清除它 0: 没发生锁定复位 1: 锁定复位已发生 |
| 1 | SOFT_RESET_PENDING | 1'b0 | RW | 写入 1 到此位将使芯片复位。这个复位可以使 FLASH 重新获取 NVR 和主数据 0: 不使能软件复位 1: 使能软件复位和记录这个复位标志 |
| 0 | SLEEP_PENDING | 1'b0 | RO | 写 LP_CON[0]将记录休眠模式标志, 即使端口唤醒系统复位 |

19.2.26 WKUP_CON1

Address offset = 0x78

| Width | Name | Reset | Property | Description |
|-------|--------------|-------|----------|--|
| 31:28 | Reserved | - | - | - |
| 27:16 | WKUP_PENDING | 12'h0 | RO | 检测 IO 唤醒 pending 0: 没有检测到唤醒 pending 1: 检测到唤醒 pending |
| 15:12 | Reserved | - | - | - |
| 11:0 | CLR_PENDING | 12'h0 | WO | 写 1 清除标志 |

20 液晶控制器 (LCD)

20.1 简介

LCD Controller 是段码式液晶屏的数字控制模块，COM+SEG 总数是 42 个，其中 COM 最大支持 8 个，SEG 最大支持 38 个。

20.2 功能框图

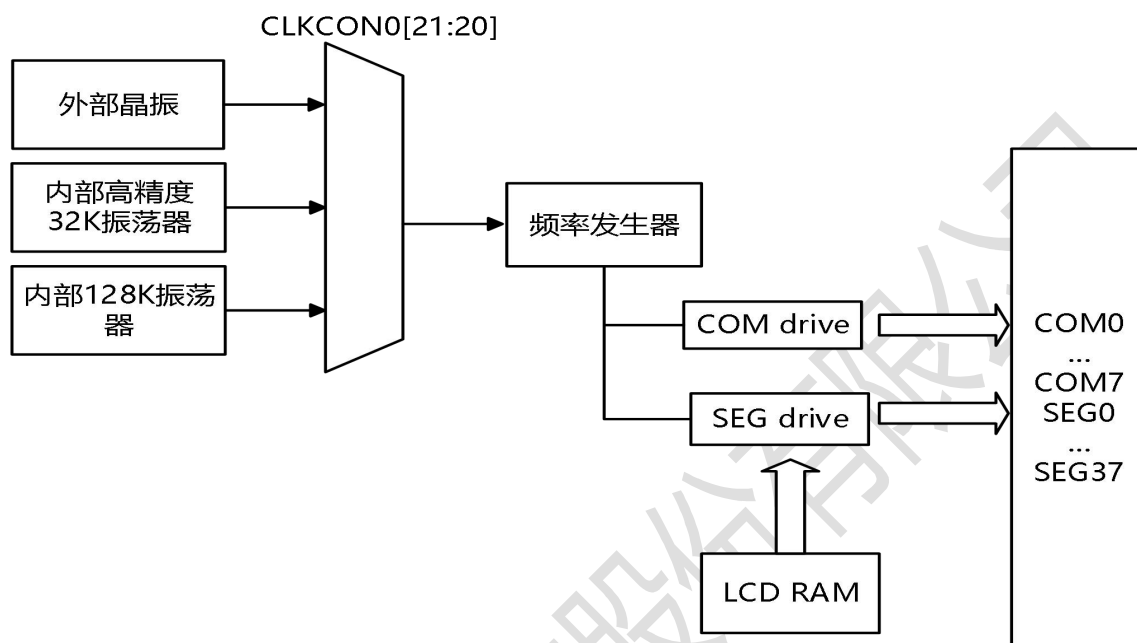


图 20-1 LCD 控制模块框图

20.3 主要特性

- 最大支持 COM+SEG 数量是 42 个，COM 最大可以选 8 个，SEG 最大可以选 38 个，COM 和 SEG 都支持在 42 个引脚中任意选择
- 支持调节 LCD 扫描刷新率
- 支持 duty: 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7 和 1/8
- 支持 bias: 1/2, 1/3
- Double buffer 寄存器，允许软件随时更新显存数据，不影响显示效果
- 三种供电方式：支持内部升压电源给 LCD 供电、内部电阻分压和外部电源直接给 LCD 供电
- 支持两种方法调节对比度：1. 采用内部升压电源供电时，调节电压最大值；2. 调节 frame 之间的死区时间
- 支持低功耗模式 Sleep Mode, Stop Mode 时 LCD 显示
- 闪烁功能：支持配置选中的 COM 和 SEG 闪烁，闪烁频率可调
- 支持软件清屏功能，对寄存器 1bit 写 1 关闭显示，写 0 恢复显示
- LCD 的 COM 和 SEG 对应的引脚不用作驱动 LCD 时，可以用于其它外设功能或者 GPIO

20.4 原理介绍

20.4.1 时钟源选择

LCD 控制器时钟源有四种，可根据 CLKCON0[21:20] 自由配置。具体参考系统寄存器章节。

20.4.2 LCD 复位源

LCD 可有两种复位操作：1. 系统复位 2. 软件复位(SYSCON0[12])。

20.4.3 LCD 扫描控制

通过配置模块寄存器 LCD_COML_ENA、LCD_SEGL_ENA 和 LCD_COMH_SEGH_ENA 使能 COM 口, SEG 口, 最大支持 COM+SEG 数量是 42 个, COM 最大可以选 8 个, SEG 最大可以选 38 个, COM 和 SEG 都支持在 42 个引脚中任意选择, 可在 3.2 管脚定义, 表 3-1 查看引脚和 LCDIO 对应, 在 COM 口使能对应的 bit, 相对应的引脚作为 COM 口, 在 SEG 口使能对应的 bit, 相对应的引脚作为 SEG 口。

例:

LCD_COML_ENA = 0x71;

PA0 为 COM0, PA4 为 COM1, PA5 为 COM2, PA6 为 COM3;

LCD_SEGL_ENA = 0xF0000;

PB0 为 SEG0, PB1 为 SEG1, PB2 为 SEG2, PB3 为 SEG3;

通过配置模块寄存器 LCD_DIV_CON 的 LCD_DIV[15:0]值可以调节 LCD 的扫描帧率, 帧率计算公式如下:

帧模式(Frame Mode):

$$\text{Freq_frame} = \text{Freq_lcd_clk} / \text{LCD_CLK_DIV} * \text{duty. (Duty=1/COM 数量)}$$

行模式(Row Mode):

$$\text{Freq_frame} = \text{Freq_lcd_clk} / (\text{LCD_CLK_DIV} * 2) * \text{duty. (Duty=1/COM 数量)}$$

通常帧率会调节到 30Hz~100Hz 之间, 根据显示效果和功耗之间作平衡。

20.4.4 LCD COM SEG 电压控制

BIAS:电压等级, 每一个 COM 信号 SEG 信号在不同时刻会有不同的电压值。

当 COM[n]有效时, 该信号为最高电压值 VLCD 或者 VSS, 其它时刻处于下面的电压等级。

当 SEG[n]有效时, 该信号电压值与 COM 电压相反, 非使能的 SEG 口则输出与当前有效的 COM 电压最接近的电压等级。

COM[n]:

有效:

电压值=VLCD / VSS

无效:

- 1/3BIAS 模式下: 1/3VLCD 或者 2/3VLCD
- 1/2BIAS 模式下: 1/2VLCD

SEG[n]:

有效:

电压值=VSS / VLCD

无效:

- 1/3BIAS 模式下: 2/3VLCD 或者 1/3VLCD
- 1/2BIAS 模式下: VLCD/VSS

DUTY: 1/COM, 支持 1/1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8 八种模式。

所有使能的 COM 信号轮流使能, 分时扫描, 扫描方式有 Row 和 frame 两种。

20.4.4.1 1/2 DUTY 1/2BIAS Row Mode

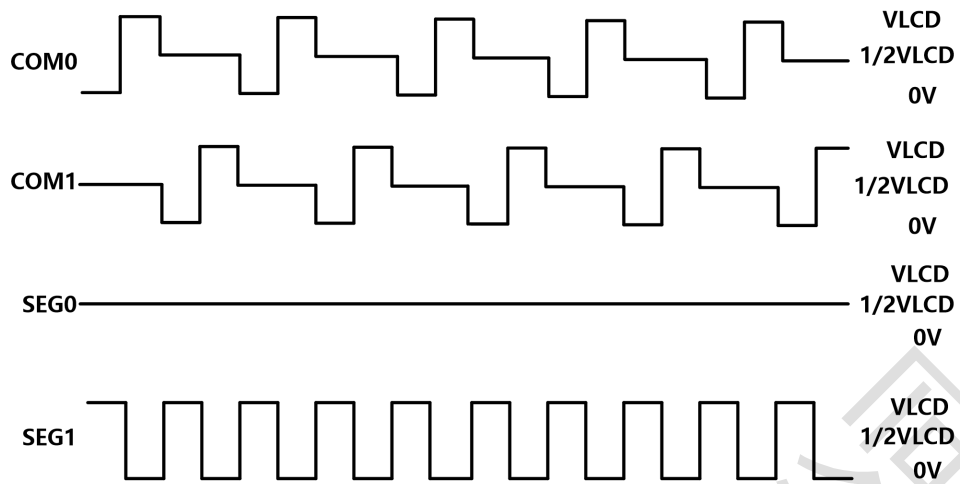


图 20-2 LCD 1/2 占空比 1/2 电压偏置波形

20.4.4.2 1/3 DUTY 1/3BIAS Row Mode

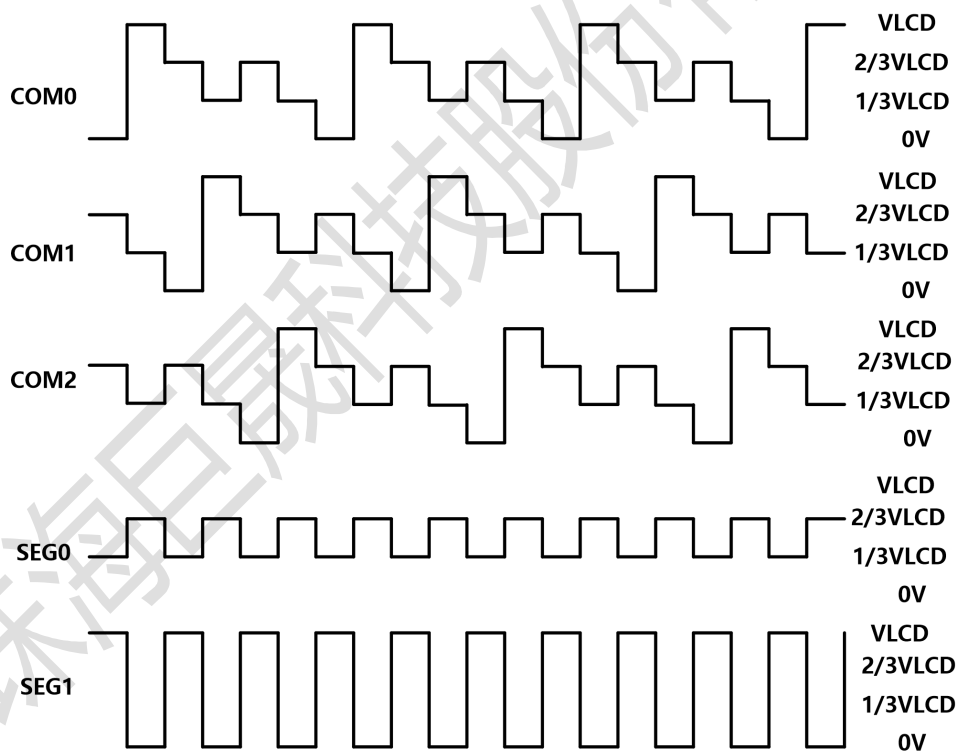


图 20-3 LCD 1/3 占空比 1/3 电压偏置波形

20.4.4.3 1/3 DUTY 1/2BIAS Frame Mode

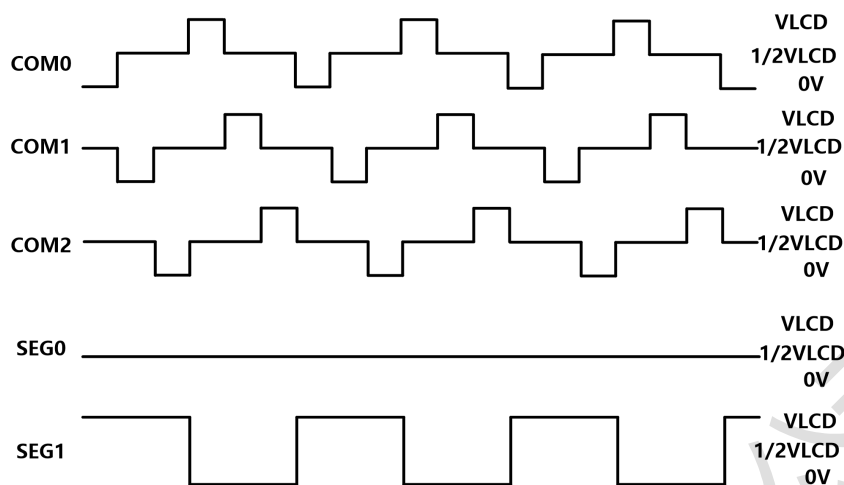


图 20-4 1/3 占空比 1/2 电压偏置波形

20.4.4.4 1/3 DUTY 1/3BIAS Frame Mode

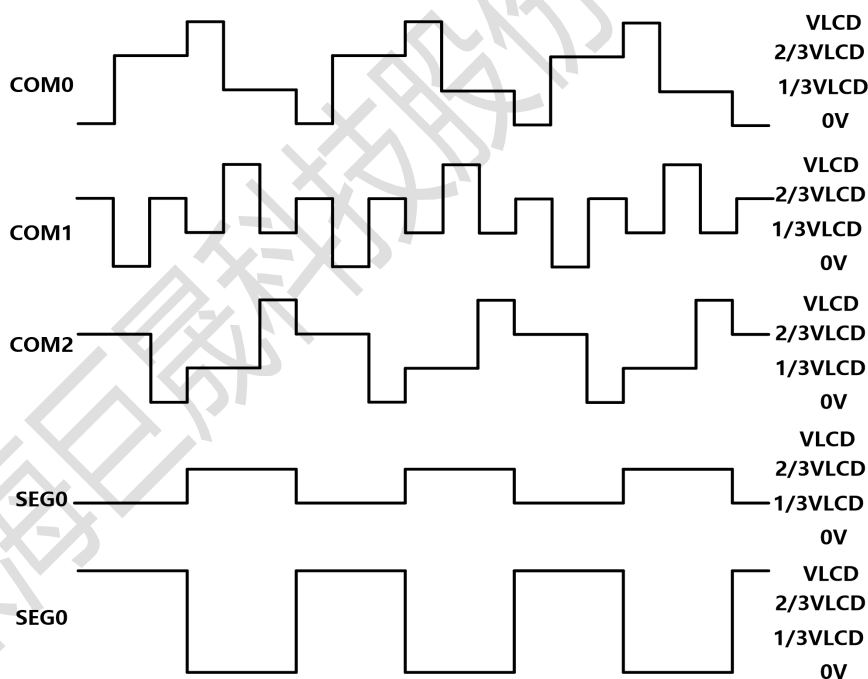


图 20-5 1/3 占空比 1/3 电压偏置帧模式波形

20.4.5 死区控制

支持用户设置在帧与帧之间设置死区时间，在死区时间内，所有 COM 和 SEG 都拉到 VSS。配置寄存器 LCD_DIV_CON 的 LCD_DEAD_ZONE[7:0] 控制死区时间为 LCD_DEAD_ZONE+1 个 LCD_CLK 周期数。

(注意: LCD_DEAD_ZONE=0~5 时为关闭死区功能，要使能死区，LCD_DEAD_ZONE 必须大于等于 6)

20.4.5.1 Row Mode 死区控制

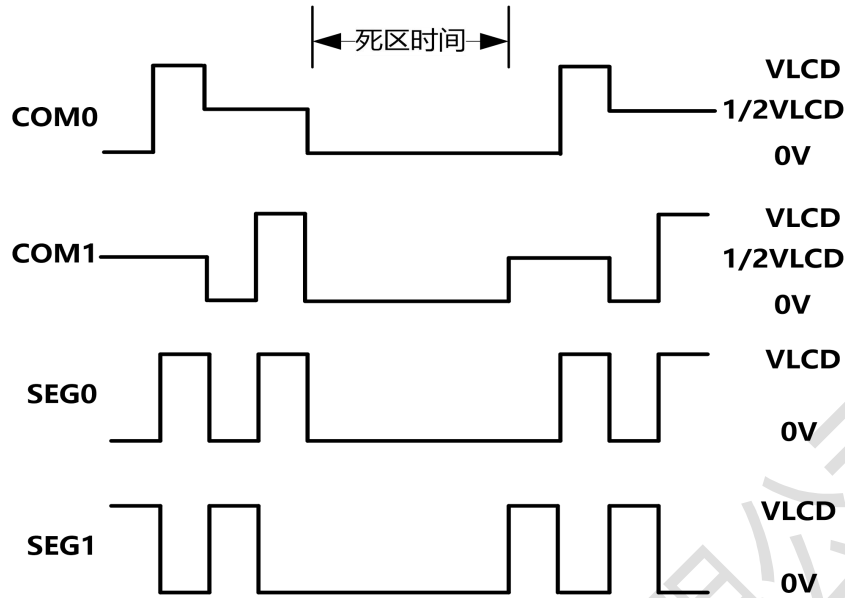


图 20-6 Row 模式死区控制波形

20.4.5.2 Frame Mode 死区控制

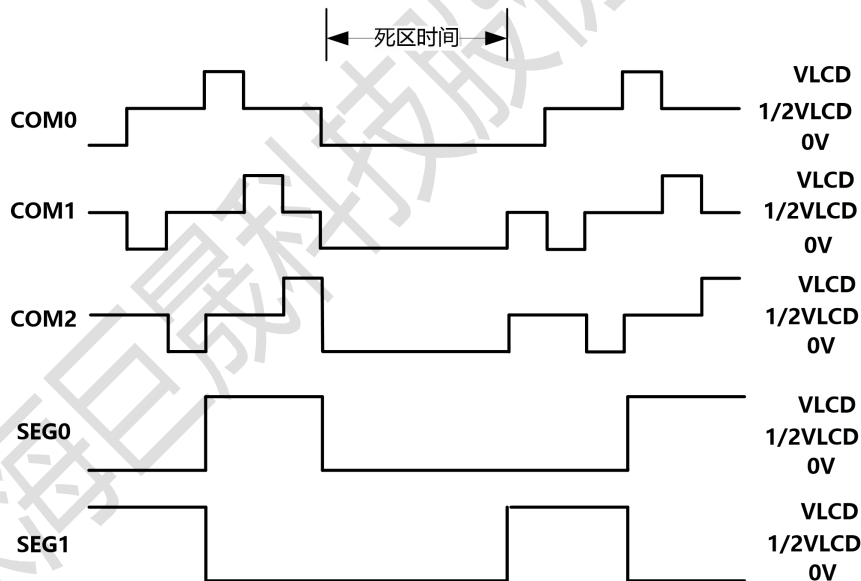


图 20-7 帧模式死区控制波形

20.4.5.3 COM SEG 翻转错位控制

通过配置寄存器 LCD_DIV_CON 的 LCD_COM_DZ 控制使 COM 和 SEG 翻转的时间点错开,即 COM 先翻转,延时 LCD_COM_DZ 个 LCD_CLK 后再到 SEG 翻转,或者反过来。目的是避免某些 LCD 屏在 COM 和 SEG 同时翻转时出现异常现象。

注: LCD_COM_DZ 必须小于等于 LCD_DIV。

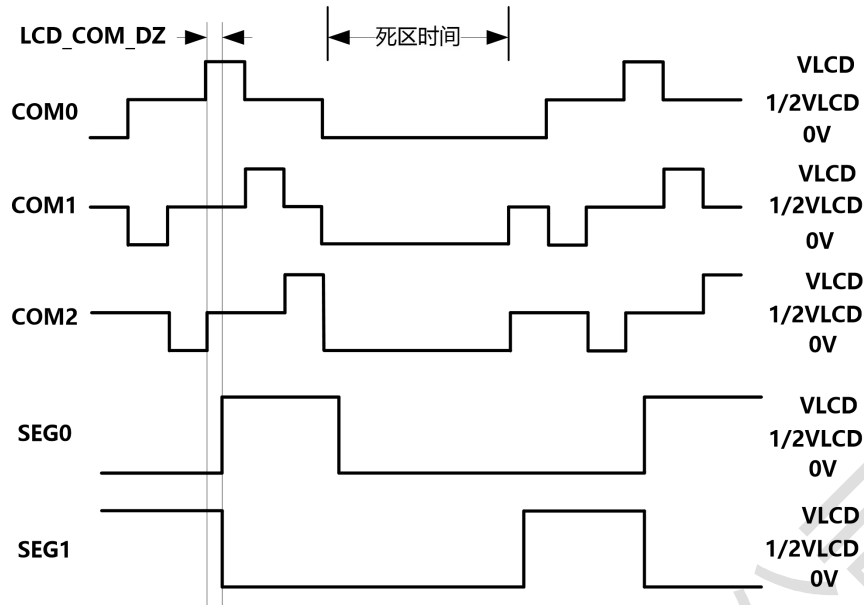


图 20-8 COM/SEG 翻转错位控制波形

20.4.6 闪烁功能

硬件支持 LCD 闪烁效果,通过配置寄存器 LCD_BLINK_CONH 的 LCD_BLINK_DIV 可以控制使 LCD 扫描过程中持续点亮 LCD_BLINK_DIV+1 帧,然后持续熄灭 LCD_BLINK_DIV+1 帧,如此交替点亮与熄灭,达到闪烁效果。闪烁频率为 $F_{blink} = Freq_frame / (LCD_BLINK_DIV + 1) / 2$ 。

20.4.7 清屏功能

对寄存器 LCD_CTRL 的 CLR_SCREEN 写 1, LCD 显示关闭,写 0, LCD 恢复显示。

20.4.8 电池电量检测

选定电压档位并使能 LCD_BOOST 内部的电压比较功能,如果选定的电压档位比 0.6V 电压高,则寄存器 LCD_PND->VDETOUT_FLAG 标志为 1,否则为 0。

比较器的其中一端为固定的 0.6V VBG 固定电压,另一端电压档位可以选择 LCD_BOOST 内部的分压或者 DAC 输出的电压,通过 LCD_VOTADT_CON->VOTADT_SRC_SEL 选择。

其中 LCD_BOOST 内部的分压值通过配置寄存器 LCD_ANA_CON0->LCD_VDET_SEL 选定, DAC 的输出电压则通过 DAC 自身的寄存器选定。

检测电池电量的方式:

- 软件控制:
 - 软件配置选择电压档位来源(来自 LCD_BOOST 内部分压或者 DAC),并配置电压档位,使能电压比较器(LCD_ANA_CON0->LCD_VDET_EN),查看比较结果标志位(LCD_PND->VDETOUT_FLAG)。
 - 根据需要更换电压档位,再使能电压比较器。或电量检查完毕,软件关闭电压比较器使能位。
- 硬件自动检测:

设定上限电压值(LCD_VOTADT_CON->VOTADT_HIGH_LIMIT)和下限电压值(LCD_VOTADT_LOW_LIMIT),当硬件检测到超过设定的上限/下限值时,会产生标志位以及中断。软件根据标志位以及当前设定的上、下限电压值确定当前电压范围,以及根据需要更新下一次检测的上、下限电压值。具体的配置请参考寄存器 LCD_VOTADT_CON 和 LCD_VOTADT_FREQ 的描述。

20.5 寄存器描述

20.5.1 LCD_CTRL

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|------|-------|----------|-------------|
|-------|------|-------|----------|-------------|

| | | | | |
|------|-----------------|------|----|--|
| 31:8 | Reserved | - | - | - |
| 7 | POWER_UPDATE_IE | 1'b0 | RW | 电量更新中断使能控制 0: 电量更新中断关闭 1: 电量更新中断使能 |
| 6 | LCD_FRAME_IE | 1'b0 | RW | LCD 扫描帧中断使能控制 0: 帧中断关闭 1: 帧中断使能 |
| 5 | LCD_COM_DZ_SIGN | 1'b0 | RW | COM 与 SEG 错位更新的先后顺序: 0: COM 先更新 1: SEG 先更新 |
| 4 | CLR_SCREEN | 1'b0 | RW | 关闭显示控制: 0: 关闭显示失能 1: 关闭显示使能 |
| 3 | FRAME_MODE | 1'b0 | RW | COM 动态扫描模式: 0: ROW 模式, 每次扫描都会出现 VCC 和 VSS 1: FRAME 模式, 每次扫描只会出现 VCC 或者 VSS |
| 2 | Reserved | - | - | - |
| 1 | BIAS_TYPE | 1'b0 | RW | COM, SEG 的电压偏移模式: 0: 1/3BIAS 模式 1: 1/2BIAS 模式 |
| 0 | LCD_EN | 1'b0 | RW | LCD Digital Controller 使能控制: 0: 不使能 1: 使能 |

20.5.2 LCD_COM_NUM

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|---|
| 31:3 | Reserved | - | - | - |
| 2:0 | LCD_COM_NUM | 3'b0 | RW | LCD COM 数量配置: b000: COM=1 b001: COM=2 b010: COM=3 b011: COM=4 b100: COM=5 b101: COM=6 b110: COM=7 b111: COM=8 |

20.5.3 LCD_REFRESH

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|-----------------|-------|----------|---|
| 31:1 | Reserved | - | - | - |
| 0 | LCD_REFRESH_PND | 1'b0 | RW | LCD 显示更新控制位 (软件置 1, 硬件清 0) 写: 0: 无效 1: 请求更新显示数据 读: 0: 更新已完成, 可以发起下一次请求 |

| | | | | |
|--|--|--|--|-----------------------|
| | | | | 1: 更新未完成, 等待完成再做下一次请求 |
|--|--|--|--|-----------------------|

20.5.4 LCD_DIV_CON

Address offset: 0x0C

| Width | Name | Reset | Property | Description |
|-------|---------------|----------|----------|--|
| 31:24 | LCD_DEAD_ZONE | 8'h0 | RW | LCD 死区时间控制 h00~h05: 死区功能关闭 h06~hFF: 死区时间为 LCD_DEAD_ZONE+1 个 LCD_CLK 时钟周期 注意: 需要同时使能错位功能和死区功能时, 需要满足条件: LCD_DEAD_ZONE<=LCD_CLK_DIV 或 LCD_DEAD_ZONE>=LCD_CLK_DIV*2 |
| 23:16 | LCD_COM_DZ | 8'h0 | RW | COM、SEG 错位更新延时控制 h00: COM 和 SEG 同时更新, 不错位 h01~hff: 错位时间为 LCD_COM_DZ 个 LCD_CLK 时钟周期 注: 需要同时使能错位功能和死区功能时, 需要满足条件: 1. LCD_COM_DZ>=6 2. LCD_COM_DZ<=(LCD_CLK_DIV-6) 3. LCD_COM_DZ<=LCD_DEAD_ZONE |
| 15:0 | LCD_CLK_DIV | 16'h0006 | RW | 每个 COM 占用的时间控制 h0000~h0005: 保留 h0006~hFFFF: 每个 COM 使能 LCD_CLK_DIV+1 个 LCD_CLK 注: 需要同时使能错位功能和死区功能时, LCD_CLK_DIV 的范围为 12~65535。 |

20.5.5 LCD_COML_ENA

| Width | Name | Reset | Property | Description |
|-------|--------------|-------|----------|------------------------------------|
| 31:0 | LCD_COML_ENA | 32'h0 | RW | COM 口低 32 位使能控制 0: 不使能 1: 使能 |

Address offset: 0x10

20.5.6 LCD_SEGL_ENA

Address offset: 0x14

| Width | Name | Reset | Property | Description |
|-------|--------------|-------|----------|------------------------------------|
| 31:0 | LCD_SEGL_ENA | 32'h0 | RW | SEG 口低 32 位使能控制 0: 不使能 1: 使能 |

20.5.7 LCD_COMH_SEGH_ENA

Address offset: 0x18

| Width | Name | Reset | Property | Description |
|-------|--------------|-------|----------|-----------------|
| 31:26 | Reserved | - | - | - |
| 25:16 | LCD_COMH_ENA | 10'h0 | RW | COM 口高 10 位使能控制 |

| | | | | |
|-------|--------------|-------|----|------------------------------------|
| | | | | 0: 不使能 1: 使能 |
| 15:10 | Reserved | - | - | - |
| 9:0 | LCD_SEGH_ENA | 10'h0 | RW | SEG 口高 10 位使能控制 0: 不使能 1: 使能 |

20.5.8 LCD_TEST_CFG

Address offset: 0x1C

| Width | Name | Reset | Property | Description |
|-------|-----------------|-------|----------|--|
| 31:5 | Reserved | - | - | - |
| 4:3 | SEG_TEST_VAL | 2'b0 | RW | 测试模式下, SEG 口恒定输出值 SEG_TEST_VAL=00、01、10、11 |
| 2:1 | COM_TEST_VAL | 2'b0 | RW | 测试模式下, COM 口恒定输出值 COM_TEST_VAL=00、01、10、11 |
| 0 | COM_SEG_TEST_EN | 1'b0 | RW | 测试模式使能控制 0: 不使能 1: 使能 使能测试模式后, COM 口和 SEG 口输出恒定值。 |

20.5.9 LCD_COM0_SEGL

Address offset: 0x20

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|-----------------------|
| 31:0 | LCD_COM0_SEGL | 32'h0 | RW | COM0 对应的低 32 位 SEG 数据 |

20.5.10 LCD_COM0_SEGH

Address offset: 0x24

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|----------------------|
| 31:6 | Reserved | - | - | - |
| 5:0 | LCD_COM0_SEGH | 6'h0 | RW | COM0 对应的高 6 位 SEG 数据 |

20.5.11 LCD_COM1_SEGL

Address offset: 0x28

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|-----------------------|
| 31:0 | LCD_COM1_SEGL | 32'h0 | RW | COM1 对应的低 32 位 SEG 数据 |

20.5.12 LCD_COM1_SEGH

Address offset: 0x2C

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|----------------------|
| 31:6 | Reserved | - | - | - |
| 5:0 | LCD_COM1_SEGH | 6'h0 | RW | COM1 对应的高 6 位 SEG 数据 |

20.5.13 LCD_COM2_SEGL

Address offset: 0x30

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|-----------------------|
| 31:0 | LCD_COM2_SEGL | 32'h0 | RW | COM2 对应的低 32 位 SEG 数据 |

20.5.14 LCD_COM2_SEGH

Address offset: 0x34

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|----------------------|
| 31:6 | Reserved | - | - | - |
| 5:0 | LCD_COM2_SEGH | 6'h0 | RW | COM2 对应的高 6 位 SEG 数据 |

20.5.15 LCD_COM3_SEGL

Address offset: 0x38

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|-----------------------|
| 31:0 | LCD_COM3_SEGL | 32'h0 | RW | COM3 对应的低 32 位 SEG 数据 |

20.5.16 LCD_COM3_SEGH

Address offset: 0x3C

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|----------------------|
| 31:6 | Reserved | - | - | - |
| 5:0 | LCD_COM3_SEGH | 6'h0 | RW | COM3 对应的高 6 位 SEG 数据 |

20.5.17 LCD_COM4_SEGL

Address offset: 0x40

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|-----------------------|
| 31:0 | LCD_COM4_SEGL | 32'h0 | RW | COM4 对应的低 32 位 SEG 数据 |

20.5.18 LCD_COM4_SEGH

Address offset: 0x44

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|----------------------|
| 31:6 | Reserved | - | - | - |
| 5:0 | LCD_COM4_SEGH | 6'h0 | RW | COM4 对应的高 6 位 SEG 数据 |

20.5.19 LCD_COM5_SEGL

Address offset: 0x48

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|-----------------------|
| 31:0 | LCD_COM5_SEGL | 32'h0 | RW | COM5 对应的低 32 位 SEG 数据 |

20.5.20 LCD_COM5_SEGH

Address offset: 0x4C

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|----------------------|
| 31:6 | Reserved | - | - | - |
| 5:0 | LCD_COM5_SEGH | 6'h0 | RW | COM5 对应的高 6 位 SEG 数据 |

20.5.21 LCD_COM6_SEGL

Address offset: 0x50

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|-----------------------|
| 31:0 | LCD_COM6_SEGL | 32'h0 | RW | COM6 对应的低 32 位 SEG 数据 |

20.5.22 LCD_COM6_SEGH

Address offset: 0x54

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|----------------------|
| 31:6 | Reserved | - | - | - |
| 5:0 | LCD_COM6_SEGH | 6'h0 | RW | COM6 对应的高 6 位 SEG 数据 |

20.5.23 LCD_COM7_SEGL

Address offset: 0x58

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|-----------------------|
| 31:0 | LCD_COM7_SEGL | 32'h0 | RW | COM7 对应的低 32 位 SEG 数据 |

20.5.24 LCD_COM7_SEGH

Address offset: 0x5C

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|----------------------|
| 31:6 | Reserved | - | - | - |
| 5:0 | LCD_COM7_SEGH | 6'h0 | RW | COM7 对应的高 6 位 SEG 数据 |

20.5.25 LCD_BLINK_ENAL

Address offset: 0x60

| Width | Name | Reset | Property | Description |
|-------|------------|-------|----------|-----------------------------------|
| 31:0 | BLINK_ENAL | 32'h0 | RW | COM、SEG 闪烁使能控制 0: 不使能 1: 使能 |

20.5.26 LCD_BLINK_ENAH

Address offset: 0x64

| Width | Name | Reset | Property | Description |
|-------|---------------|-------|----------|--|
| 31:24 | Reserved | - | - | - |
| 23:16 | LCD_BLINK_DIV | 8'h0 | RW | LCD 闪烁帧数控制 h00: 不使能闪烁 h01~hFF: 持续点亮/熄灭帧数 |
| 15:10 | Reserved | - | - | - |
| 9:0 | BLINK_ENAH | 10'h0 | RW | COM、SEG 闪烁使能控制 0: 不使能 1: 使能 |

20.5.27 LCD_PND

Address offset: 0x68

| Width | Name | Reset | Property | Description |
|-------|-----------------|-------|----------|--|
| 31:5 | Reserved | - | - | - |
| 4 | VFBOUT_FLAG | 1'b0 | RO | VLCD 电压检测结果反馈 0: VLCD 升压未完成 1: VLCD 升压已完成 |
| 3 | VDETOUT_FLAG | 1'b0 | RO | 电量比较结果标志: 0: 实际电压小于设定电压等级 1: 实际电压大于设定电压等级 |
| 2 | HIGH_POWER_FLAG | 1'b0 | RW | 电量上升标志 读: 0: 电量等级没有更新 1: 电量等级上升 写: 0: 无效 1: 清除 HIGH_POWER_FLAG |
| 1 | LOW_POWER_FLAG | 1'b0 | RW | 电量下降标志 读: 0: 电量等级没有更新 1: 电量等级下降 |

| | | | | |
|---|--------------------|------|----|--|
| | | | | 写: 0: 无效 1: 清除 LOW_POWER_FLAG |
| 0 | LCD_FRAME_INT PEND | 1'b0 | RW | LCD 帧中断 pending 标志 读: 0: 未扫描完一帧 1: 扫描完一帧 写: 0: 无效 1: 清除 LCD_FRAME_INT PEND |

20.5.28 LCD_ANA_CON0

Address offset: 0x6C

| Width | Name | Reset | Property | Description |
|-------|---------------|---------|----------|---|
| 31:19 | Reserved | - | - | - |
| 18 | LCD_VFBDET_EN | 1'b0 | RW | 电荷泵升压完成标记输出使能 0: 不输出 1: 输出到 VFBOU_FLAG |
| 17:16 | Reserved | - | - | - |
| 15 | LCD_VTEST_EN | 1'b0 | RW | 测试电压输出使能 0: 不使能 1: 使能 |
| 14:12 | LCD_VDET_SEL | 3'b0 | RW | VCC 电压检测; 当 VOTADT_SRC_SEL=0, VCC 大于此设置值时, VTSOUT 输出高电平 b000: 1.756v b001: 1.926v b010: 2.146v b011: 2.266v b100: 2.416v b101: 2.576v b110: 2.756v b111: 2.966v |
| 11 | LCD_VDET_EN | 1'b0 | RW | LCD 模拟升压模块 VCC 电压检测使能 0: 不使能 1: 使能, 比较结果输出到 VDETOUT_FLAG |
| 10:9 | LCD_NOLAP_SEL | 2'b0 | RW | 驱动电源 MOS 非交叠时间控制 b00: 1.2ns b01: 2.6ns b10: 5.2ns b11: 7.5ns |
| 8 | LCD_VCP_MODE | 1'b0 | RW | 1/3 VLCD 电压档位模式: 0: 等于 LCD_VCP_SEL 设定的档位值 1: 等于 LCD_VCP_SEL 档位值的两倍 |
| 7:4 | LCD_VCP_SEL | 4'b0011 | RW | 1/3 VLCD 电压档位选择 b0000: 0.95v b0001: 1.00v b0010: 1.05v |

| | | | | |
|---|----------------|------|----|--|
| | | | | b0011: 1.10v b0100: 1.15v b0100: 1.20v b0110: 1.25v b0111: 1.30v b1000: 1.35v b1001: 1.40v b1010: 1.45v b1011: 1.50v b1100: 1.55v b1101: 1.60v b1110: 1.65v b1111: 1.70v |
| 3 | LCD_BIAS_MODE | 1'b0 | RW | LCD 电压偏移模式 0: 1/3 VLCD 模式 1: 1/2 VLCD 模式 |
| 2 | LCD_BOOST_MODE | 1'b0 | RW | LCD 升压模式 0: 电荷泵升压模式 1: 带有驱动器的电阻分压模式 |
| 1 | LCD_ISEL_MODE | 1'b0 | RW | LCD_BOOST 电流模式 0: 小电流模式 1: 大电流模式 |
| 0 | LCD_POWER | 1'b0 | RW | LCD 模拟上电使能 0: 不使能 1: 使能 |

20.5.29 LCD_ANA_CLK_CON

Address offset: 0x70

| Width | Name | Reset | Property | Description |
|-------|-----------------|-------|----------|--|
| 31:9 | Reserved | - | - | - |
| 8:1 | LCD_ANA_CLK_DIV | 8'h0 | RW | LCD 模拟时钟分频比, 用于对 LCD_CLK 分频 h00: 不分频 h01~hFF: 2*LCD_ANA_CLK_DIV 分频 |
| 0 | LCD_ANA_CLK_EN | 1'b0 | RW | LCD 模拟时钟使能 0: 不使能 1: 使能 |

20.5.30 LCD_VOTADT_CON

Address offset: 0x74

| Width | Name | Reset | Property | Description |
|-------|-------------------|-------|----------|---|
| 31:24 | VOTADT_HIGH_LIMIT | 8'h0 | RW | 高电档位选择, 用于和 0.6V VBG 比较。 当 VOTADT_SRC_SEL=0 时, 只有低 3bit 有效, 电压档位可选 h00~h07, 具体电压值参考 LCD_ANA_CON0 寄存器中 LCD_VDET_SEL。 当 VOTADT_SRC_SEL=1 时, 8bit 有效, 档位可选 h2F~h5B, 具体电压值为: 0.6*256/VOTADT_HIGH_LIMIT。 |
| 23:16 | VOTADT_LOW_LIMIT | 8'h0 | RW | 低电档位选择, 用于和 0.6V VBG 比较。 当 VOTADT_SRC_SEL=0 时, 只有低 3bit 有效, |

| | | | | |
|------|--------------------|------|----|---|
| | | | | 电压档位可选 000~111，具体电压值参考 LCD_ANA_CON0-> LCD_VDET_SEL。 当 VOTADT_SRC_SEL=1 时，8bit 有效，档位可选 h2F~h5B，具体电压值为： 0.6*256/VOTADT_LOW_LIMIT。 |
| 15:8 | VOTADT_CONFIRM_NUM | 8'h0 | RW | 电量变化确认次数 h00: 不确认 h01: 确认 1 次 h02: 确认 2 次 hFF: 确认 255 次 |
| 7:6 | Reserved | - | - | - |
| 5 | VOTADT_TRIG_SEL | 1'b0 | RW | 电量检测触发源选择 0: 由 LCD 刷屏 N 帧后触发 1: 由 RTCC 每秒触发一次 |
| 4 | VOTADT_WKUP_EN | 1'b0 | RW | 电量检测出现 LOW_POWER 或 HIGH_POWER 时唤醒 CPU 0: 不使能 1: 使能 |
| 3:2 | VOTADT_WAIT_TIME | 2'b0 | RW | 使能 LCD_VDET_EN 后硬件等待时间 b00: 等待 1 个 cycle b01: 等待 2 个 cycle b10: 等待 3 个 cycle b11: 等待 4 个 cycle |
| 1 | VOTADT_SRC_SEL | 1'b0 | RW | 电量检测电压源选择 0: LCD 升压电路内部分压值 1: DAC 输出电压值 |
| 0 | VOTADT_EN | 1'b0 | RW | 电池电量自动检测使能 0: 不使能 1: 使能 |

20.5.31 LCD_VOTADT_FREQ

Address offset: 0x78

| Width | Name | Reset | Property | Description |
|-------|-------------|-------|----------|--|
| 31:8 | Reserved | - | - | - |
| 7:0 | VOTADT_FREQ | 8'h0 | RW | 电池电量自动检测频率 h00: LCD 每扫描完 1 帧检测一次 h01: LCD 每扫描完 2 帧检测一次 ... hFF: LCD 每扫描完 256 帧检测一次 |

21 实时时钟 (RTC)

21.1 简介

实时时钟控制器(Real Time Clock Control)是一个万年日历管理模块，拥有独立的定时器，修改计数器的值可以重新设置系统当前的时间和日期，系统可自动将月份的天数补偿为 28、29(闰年)、30、31 天。该模块工作时钟 RTCCLK 为 32.768KHz，在系统处于低功耗模式下仍正常工作。

21.2 功能框图

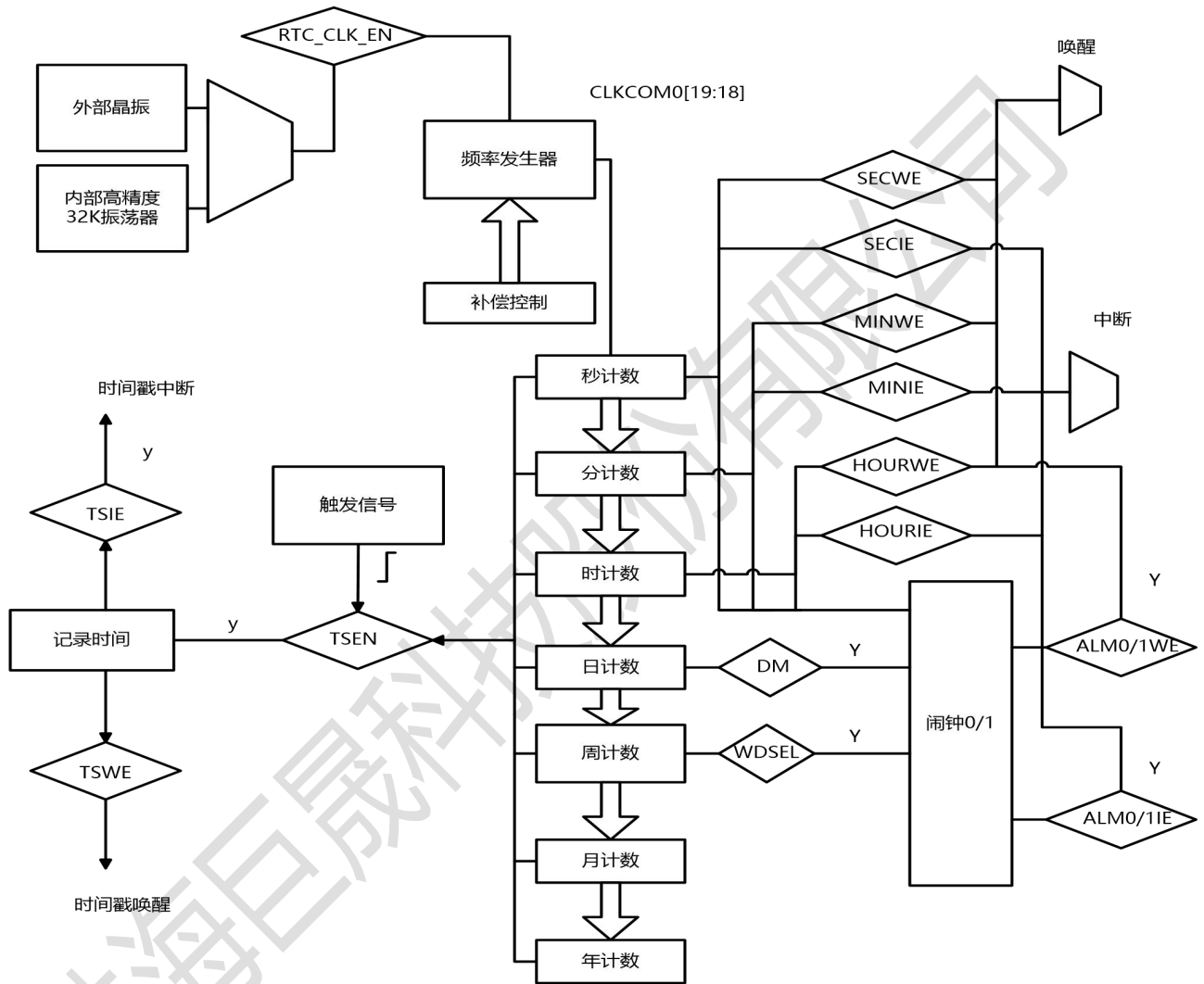


图 21-1 RTC 模块框图

主要特性：

- 包含秒、分、小时(12 或 24 小时制)、星期几、日期、月份和年份的日历
- 数字时钟补偿功能，补偿精度+30.5ppm(1/32768)，补偿范围为 0.119ppm(INTERVAL=255,VALUE=1)到 3906ppm(INTERVAL=1, VALUE=-128)
- 两个具有中断功能的可编程闹钟
- 一个用于事件保存的时间戳功能

备注：部分寄存器和 RTCC_EN 在配置之后需要 KST 触发才能生效。这些寄存器包括：时间寄存器 RTCC_TIME、日期寄存器 RTCC_DATE、补偿寄存器 RTCC_COMPE。

21.3 寄存器描述

21.3.1 控制寄存器 (RTCC_CR)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:22 | Reserved | - | - | - |
| 21:20 | TSSEL | 2'b0 | RW | 时间戳触发边沿选择 b00: 上升沿触发 b01: 下降沿触发 b1x: 双边沿触发 |
| 19 | TSWE | 1'b0 | RW | 时间戳唤醒使能位 0: 禁止唤醒 1: 使能唤醒 |
| 18 | HOURWE | 1'b0 | RW | 每小时唤醒使能位 0: 禁止唤醒 1: 使能唤醒 |
| 17 | MINWE | 1'b0 | RW | 每分钟唤醒使能位 0: 禁止唤醒 1: 使能唤醒 |
| 16 | SECWE | 1'b0 | RW | 每秒钟唤醒使能位 0: 禁止唤醒 1: 使能唤醒 |
| 15 | ALM1WE | 1'b0 | RW | 闹钟 1 唤醒使能位 0: 禁止唤醒 1: 使能唤醒 |
| 14 | ALM0WE | 1'b0 | RW | 闹钟 0 唤醒使能位 0: 禁止唤醒 1: 使能唤醒 |
| 13 | TSIE | 1'b0 | RW | 时间戳中断使能位 0: 禁止中断 1: 使能中断 |
| 12 | HOURIE | 1'b0 | RW | 每小时中断使能位 0: 禁止中断 1: 使能中断 |
| 11 | MINIE | 1'b0 | RW | 每分钟中断使能位 0: 禁止中断 1: 使能中断 |
| 10 | SECIE | 1'b0 | RW | 每秒钟中断使能位 0: 禁止中断 1: 使能中断 |
| 9 | ALM1IE | 1'b0 | RW | 闹钟 1 的中断使能位 0: 禁止中断 1: 使能中断 |
| 8 | ALM0IE | 1'b0 | RW | 闹钟 0 的中断使能位 0: 禁止中断 1: 使能中断 |
| 7 | Reserved | - | - | - |

| | | | | |
|---|--------|------|----|--|
| 6 | TSEN | 1'b0 | RW | 时间戳使能位 0: 禁止时间戳 1: 使能时间戳 |
| 5 | KST | 1'b0 | WO | 时间、日期更新配置 kick start 位 0: 无效操作 1: 配置有效 |
| 4 | REGSEL | 1'b1 | RW | 获取时间、日期寄存器值选择位 0: 软件配置值 1: 实际工作值 |
| 3 | PMEN | 1'b0 | RW | 时间 PM 使能位 0: 24 小时制 1: 12 小时制 |
| 2 | ALM1EN | 1'b0 | RW | 闹钟 1 使能控制位 0: 禁止闹钟 1 1: 使能闹钟 1 |
| 1 | ALM0EN | 1'b0 | RW | 闹钟 0 使能控制位 0: 禁止闹钟 0 1: 使能闹钟 0 |
| 0 | RTCCEN | 1'b0 | RW | RTCC 使能控制位 0: 禁止 RTCC 1: 使能 RTCC |

21.3.2 时间寄存器 (RTCC_TIME)

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:21 | Reserved | - | - | - |
| 20 | AMPM | 1'b0 | RO | 时间 AM/PM 选择位 0: 当前 TIME 为 AM 1: 当前 TIME 为 PM |
| 19:18 | HT | 2'b0 | RW | 时间小时的十位(Hour tens) |
| 17:14 | HU | 4'b0 | RW | 时间小时的个位(Hour units) |
| 13:11 | MT | 3'b0 | RW | 时间分钟的十位(Minute tens) |
| 10:7 | MU | 4'b0 | RW | 时间分钟的个位(Minute units) |
| 6:4 | ST | 3'b0 | RW | 时间秒的十位(Second tens) |
| 3:0 | SU | 4'b0 | RW | 时间秒的个位(Second units) |

21.3.3 日期寄存器 (RTCC_DATE)

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------------------------------|
| 31:22 | Reserved | - | - | - |
| 21:18 | YT | 4'b0 | RW | 年份的十位(Year tens) |
| 17:14 | YU | 4'b0 | RW | 年份的个位(Year units) |
| 13 | MT | 1'b0 | RW | 月份的十位(Month tens) |
| 12:9 | MU | 4'b0 | RW | 月份的个位(Month units) |
| 8:7 | DT | 2'b0 | RW | 日期的十位(Date tens) |
| 6:3 | DU | 4'b0 | RW | 日期的个位(Date units) |
| 2:0 | WDU | 3'b0 | RW | 星期几的个位(Week day units) b001: 星期一 |

| | | | | |
|--|--|--|--|-----------|
| | | | | ... |
| | | | | b111: 星期日 |

21.3.4 状态寄存器 (RTCC_STA)

Address offset: 0x0C

| Width | Name | Reset | Property | Description |
|-------|-----------|-------|----------|-----------------------------------|
| 31:6 | Reserved | - | - | - |
| 5 | TSPEND | 1'b0 | RW | 时间戳的 PENDING 位 写 1 清 PENDING |
| 4 | HOURLPEND | 1'b0 | RW | 时钟的 PENDING 位 写 1 清 PENDING |
| 3 | MINPEND | 1'b0 | RW | 分钟的 PENDING 位 写 1 清 PENDING |
| 2 | SECPEND | 1'b0 | RW | 秒钟的 PENDING 位 写 1 清 PENDING |
| 1 | ALM1PEND | 1'b0 | RW | 闹钟 1 的 PENDING 位 写 1 清 PENDING |
| 0 | ALM0PEND | 1'b0 | RW | 闹钟 0 的 PENDING 位 写 1 清 PENDING |

21.3.5 闹钟寄存器 (RTCC_ALARM0/1)

Address offset: 0x10

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:30 | Reserved | - | - | - |
| 29 | DM | 1'b0 | RW | 日期匹配位(Date match): 0: 禁止日期匹配 1: 使能日期匹配 |
| 28:27 | DT | 2'b0 | RW | 日期的十位(Date tens) |
| 26:23 | DU | 4'b0 | RW | 日期的个位(Date units) |
| 22:20 | WDSEL | 3'b0 | RW | 星期几选择(Week day select): b000: 禁止星期匹配 b001: 匹配星期一 ... b111: 匹配星期日 |
| 19:18 | HT | 2'b0 | RW | 时间小时的十位(Hour tens) |
| 17:14 | HU | 4'b0 | RW | 时间小时的个位(Hour units) |
| 13:11 | MT | 3'b0 | RW | 时间分钟的十位(Minute tens) |
| 10:7 | MU | 4'b0 | RW | 时间分钟的个位(Minute units) |
| 6:4 | ST | 3'b0 | RW | 时间秒的十位(Second tens) |
| 3:0 | SU | 4'b0 | RW | 时间秒的个位(Second units) |

21.3.6 时钟补偿寄存器 (RTCC_COMPE)

Address offset: 0x18

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:24 | Reserved | - | - | - |
| 23:16 | VALUE | 8'h0 | RW | RTCC 时钟补偿值(Compensation value) 每个补偿间隔增加或减少的时钟周期数 |

| | | | | |
|------|----------|------|----|--|
| | | | | h00: 禁止补偿 h01: 增加 1 个时钟周期 ... h7F: 增加 127 个时钟周期 h80: 减少 128 个时钟周期 ... hFF: 减少 1 个时钟周期 |
| 15:8 | Reserved | - | - | - |
| 7:0 | INTERVAL | 8'h0 | RW | RTCC 时钟补偿间隔(Compensation interval) h00: 从第 1 秒开始每隔 1 秒进行时钟补偿 h01: 从第 1 秒开始每隔 2 秒进行时钟补偿 ... hFF: 从第 1 秒开始每隔 256 秒进行时钟补偿 |

21.3.7 时间戳时间寄存器 (RTCC_TSTIME)

Address offset: 0x1c

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|--|
| 31:22 | Reserved | - | - | - |
| 21 | TSVAL | 1'b0 | RO | 时间戳事件的影子寄存器值 0: 无时间戳事件发生 1: 有时间戳事件发生 |
| 20 | AMPM | 1'b0 | RO | 时间 AM/PM 选择位 0: 当前 time 为 AM 1: 当前 time 为 PM |
| 19:18 | HT | 2'b0 | RO | 时间小时的十位(Hour tens) |
| 17:14 | HU | 4'b0 | RO | 时间小时的个位(Hour units) |
| 13:11 | MT | 3'b0 | RO | 时间分钟的十位(Minute tens) |
| 10:7 | MU | 4'b0 | RO | 时间分钟的个位(Minute units) |
| 6:4 | ST | 3'b0 | RO | 时间秒的十位(Second tens) |
| 3:0 | SU | 4'b0 | RO | 时间秒的个位(Second units) |

21.3.8 时间戳日期寄存器 (RTCC_TSDATE)

Address offset: 0x20

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|---|
| 31:22 | Reserved | - | - | - |
| 21:18 | YT | 4'b0 | RO | 年份的十位(Year tens) |
| 17:14 | YU | 4'b0 | RO | 年份的个位(Year units) |
| 13 | MT | 1'b0 | RO | 月份的十位(Month tens) |
| 12:9 | MU | 4'b0 | RO | 月份的个位(Month units) |
| 8:7 | DT | 2'b0 | RO | 日期的十位(Date tens) |
| 6:3 | DU | 4'b0 | RO | 日期的个位(Date units) |
| 2:0 | WDU | 3'b0 | RO | 星期几的个位(Week day units) b000: 禁止星期功能 b001: 星期一 ... b111: 星期日 |

21.3.9 寄存器偏移地址

| Register name | offset | Reset value |
|---------------|--------|-------------|
| RTCC_CR | 0x00 | 32'h0 |
| RTCC_TIME | 0x04 | 32'h0 |
| RTCC_DATE | 0x08 | 32'h0 |
| RTCC_STA | 0x0C | 32'h0 |
| RTCC_ALRM0 | 0x10 | 32'h0 |
| RTCC_ALRM1 | 0x14 | 32'h0 |
| RTCC_COMPE | 0x18 | 32'h0 |
| RTCC_TSTIME | 0x1C | 32'h0 |
| RTCC_TSDATE | 0x20 | 32'h0 |

21.4 使用说明

- TIME 配置按 24 小时制配置，当使能 PMEN 时，需读取 AMPM 位区分时间是上午或下午。
- 闹钟时间配置按 24 小时制配置。
- 时间、日期、补偿寄存器配置完成时，先使能 RTCCEN，再 KST，配置才生效并开始工作。若要在 RTCC 工作中停止工作，先关闭 RTCCEN，再 KST。
- 该模块上电后默认时间工作值为全 0，系统软件复位不影响工作值。
- 时间戳边沿触发有效，当 TSPEND 清零后，下一个事件发生时，时间戳寄存器值会记录新的时间；当 TSPEND 置 1 时，下一个事件发生时，时间戳寄存器值不会发生改变。

22 频率锁相环 (PLL)

22.1 功能简介

内部频率锁相环 PLL 是一个具有小数分频时钟倍频器，其参考时钟可以选择内部高精度 32KHz RC 振荡器、内部振荡器 26MHz、外部 32KHz 晶振时钟，选择参考时钟的寄存器是 CLKCON0[5:4]；支持小数分频，整数部分的分频比配置寄存器是 FPLL_INT[10:0]=0x5B9，默认配置是 32.786KHz 倍频到 48MHz，小数分频寄存器是 FPLL_FRAC[15:0]。PLL 支持 2 种频率输入 32KHz/13MHz，选择内部 RC 振荡器 26MHz 时钟作为输入参考时，需要配置输入到 PLL 模块内部的 2 分频使能，寄存器 FPLL_CON[8]，同时相应的环路带宽要配置为 13MHz 的选择，寄存器 FPLL_CON[6]。PLL 内部内置 LDO 降压，具体配置寄存器 FPLL_CON[5:4]。

22.2 寄存器描述

22.2.1 配置寄存器 (FPLL_CON)

Address offset: 0x00

| Width | Name | Reset | Property | Description |
|-------|-------------------|-------|----------|--|
| 31:9 | Reserved | - | - | - |
| 8 | FPLL_PREDIV2 | 1'b0 | RW | PLL 参考时钟除 2 使能 0: 不使能 1: 使能 |
| 7 | FPLL_POSDIV2 | 1'b1 | RW | PLL VCO 输出时钟除 2 使能 0: 不使能 1: 使能 |
| 6 | FPLL_LPFS0 | 1'b0 | RW | PLL 低通滤波选择 0: 32KHz 时钟输入 1: 13MHz 时钟输入 |
| 5:4 | FPLL_LDOS | 2'b10 | RW | PLL 无电容 LDO 输出电压选择 b00: 1.3 b01: 1.4 b10: 1.5 b11: 1.6 |
| 3 | FPLL_DTSEN | 1'b0 | RW | FPLL 测试 feedback 和 reference clock 0: 不使能 1: 使能 |
| 2 | FPLL_ATSEN | 1'b0 | RW | FPLL 模拟模块测试使能 0: 不使能 1: 使能 |
| 1 | FPLL_SOFT_RST_EN_ | 1'b0 | RW | FPLL 软件复位 0: 复位 1: 非复位 |
| 0 | FPLL_AEN | 1'b0 | RW | FPLL 模拟使能 0: 不使能 1: 使能 |

22.2.2 配置寄存器 (FPLL_INT)

Address offset: 0x04

| Width | Name | Reset | Property | Description |
|-------|----------|-------|----------|-------------|
| 31:11 | Reserved | - | - | - |

| | | | | |
|------|----------|---------|----|---|
| 10:0 | FPLL_INT | 11'h5B9 | RW | 分频系数的整数部分 h0~h1: 保留 h2: 2 分频 h3: 3 分频 ... |
|------|----------|---------|----|---|

22.2.3 配置寄存器 (FPLL_FRAC)

Address offset: 0x08

| Width | Name | Reset | Property | Description |
|-------|-----------|----------|----------|--|
| 31:10 | Reserved | - | - | - |
| 15:0 | FPLL_FRAC | 16'h3393 | RW | 分频系数的小数部分 小数分频比 = $FPLL_FRAC / 2^{16}$ |

23 器件电子签名 (E-SIGNATURE)

电子签名存放在闪存存储器模块的系统存储区域，可以通过 JTAG、DBG 或者 CPU 读取。它所包含的芯片识别信息在出厂时编写，用户固件或者外部设备可以读取电子签名，用以自动匹配不同配置的 JS32L010 系列微控制器。

23.1 存储器容量寄存器

23.1.1 产品唯一身份标识寄存器 (96 位)

产品唯一的身份标识非常合适：

- 用来作为序列号(例如 USB 字符序列号或者其他的终端应用)
- 用来作为密码，在编写闪存时，将此唯一标识与软件加解密算法结合使用，提高代码在闪存存储器的安全性
- 用来激活带安全机制的自举过程

96 位的产品唯一身份标识所提供的参考号码对任意一个 JS32L010 系列微控制器，在任何情况下都是唯一的。用户在何种情况下，都不能修改这个身份标识。

关于这个寄存器的详细描述，请参考嵌入式闪存一章。

24 调试支持 (DBG)

JS32L010 系列内核内含硬件调试模块，支持复杂的调试操作。硬件调试模块允许内核在取指(指令断点)或访问数据(数据断点)时停止。内核停止时，内核的内部状态和系统的外部状态都是可以查询的。完成查询后，内核和外设可以被复原，程序将继续执行。当 JS32L010 系列微控制器连接到调试器并开始调试时，调试器将使用内核的硬件调试模块进行调试操作。

支持：串行调试接口

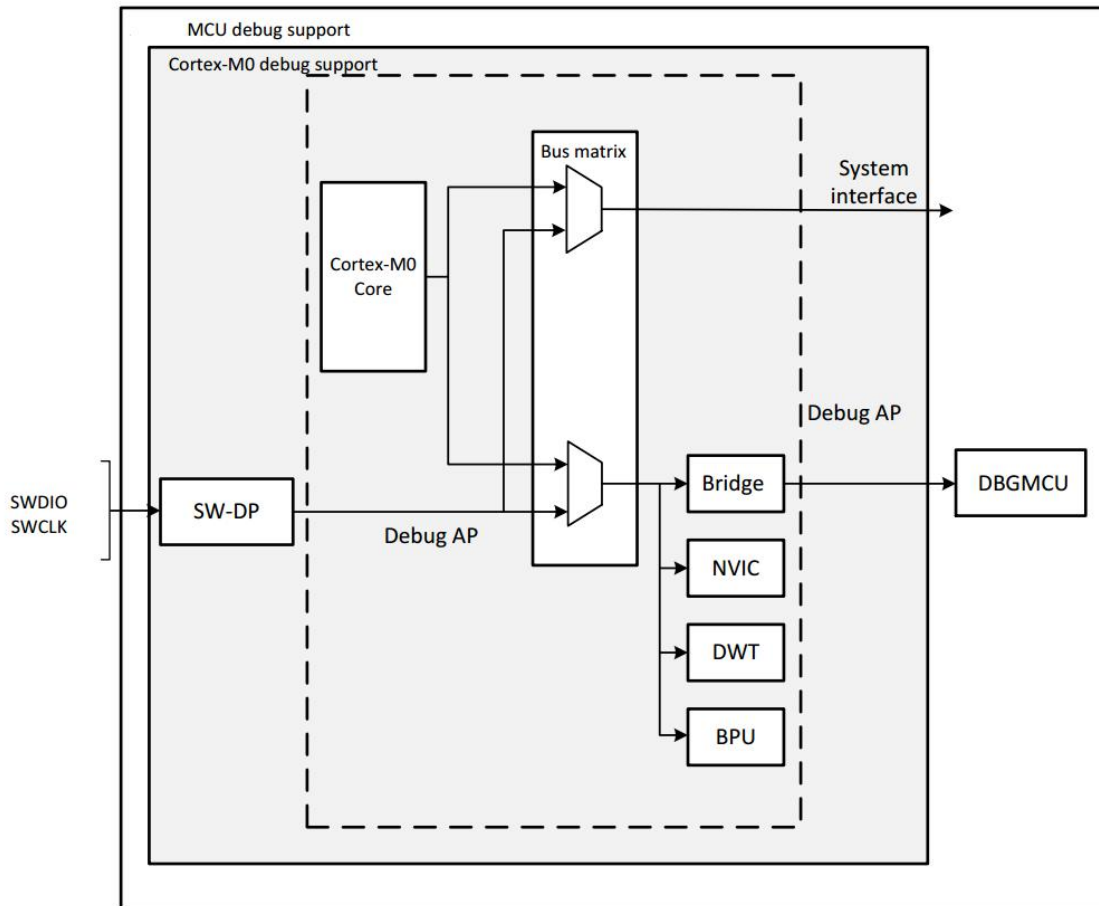


图 24-1 串行接口调试框图

CPU 内核提供集成的片上调试功能。它由以下部分组成：

- SW-DP: 串行调试端口
- AHP-AP: AHB 访问端
- ITM: 执行跟踪单元
- FPB: 闪存指令断点
- DWT: 数据触发
- TPUI: 跟踪单元接口

24.1 引脚分布和调试端口脚

不同封装有不同的有效引脚数。因此，某些与引脚相关的功能可能随封装而不同。

24.2 SWD 调试端口脚

JS32L010 的 1 组普通 I/O 口可用作 SW-DP 接口引脚。这些引脚在所有的封装里都存在。

| SWJ-DP 端口引脚名称 | SW 调试接口 | | 引脚分配 |
|---------------|---------|-----------|------|
| | 类型 | 调试功能 | |
| SWDIO | 输入/输出 | 串行数据输入/输出 | PC8 |
| SWCLK | 输入 | 串行时钟 | PC9 |

SW-DP 接口引脚选择在 flash 用户自定义功能区域定义。

24.2.1 SWD 脚上的内部上拉和下拉

保证 SWD 的输入引脚不是悬空的是非常必要的，因为它们直接连接到 D 触发器控制着调试模式。必须特别注意 SWCLK 引脚，因为它们直接连接到一些 D 触发器的时钟端。

为了避免任何未受控制的 I/O 电平，JS32L010 在 SWD 输入脚上嵌入了内部上拉和下拉。

SWDIO：默认状态是输入悬空

SWCLK：内部上/下拉可选

一旦 SWD I/O 被用户代码释放，GPIO 控制器再次取得控制。这些 I/O 口的状态将恢复到复位时的状态。

软件可以把这些 I/O 口作为普通的 I/O 口使用。

24.3 SW 调试端口

24.3.1 SW 协议介绍

此同步串行协议使用 2 个引脚：

- SWCLK：从主机到目标的时钟信号
- SWDIO：双向数据信号

协议允许读写 2 个寄存器组(DPACC 和 APACC 寄存器组)。

数据位按 LSB 传输。

由于 SWDIO 为双向口，该引脚需有上拉（默认芯片内部无上下拉，建议使用 100K 电阻）。

按协议每次 SWDIO 方向改变时，需插入一个转换时间。在该期间内主机和目标都不驱动此信号线。转换时间的默认值是 1 个比特，但可以通过配置 SWCLK 频率来调节。

24.3.2 SW 协议序列

每个序列由 3 个阶段组成：

- 主机发送包请求（8 位）
- 目标发送确认响应（3 位）
- 主机或目标发送数据（33 位）

| 比特位 | 名称 | 描述 |
|------|----------|------------------------|
| 0 | 起始 | 必须为 1 |
| 1 | APnDP | 0: 访问 DP 1: 访问 AP |
| 2 | RnW | 0: 写请求 1: 读请求 |
| 4: 3 | A (3: 2) | DP 或 AP 寄存器的地址 |
| 5 | Parity | 前面比特位的校验位 |
| 6 | Stop | 0 |
| 7 | Park | 不能由主机驱动，由于有上拉，目标永远读为 1 |

有关 DPACC 和 APACC 寄存器描述的详细资料，请参考 CPU 技术参考手册。

包请求后总是跟一个(缺省为 1 位)转换时间，此时主机和目标都不驱动线路。

| 比特位 | 名称 | 描述 |
|------|-----|-------------------------------|
| 0..2 | ACK | 001: 失败 010: 等待 100: 成功 |

当 ACK 为失败或等待，或者是一个回复读操作的 ACK，此 ACK 后有一个转换时间。

| 比特位 | 名称 | 描述 |
|-------|-------------|-------------|
| 0..31 | WDATA/RDATA | 写或读的数据 |
| 32 | Parity | 32位数据的奇偶校验位 |

读操作的数据传输操作后有一个转换时间。

24.4 MCU 调试

为了支持低功耗模式下(STOP/SLEEP)的调试，当 SWD 连接成功后，如果芯片处于 STOP 模式，则会被唤醒，如果芯片处于 SLEEP 模式，则会被复位。此复位不会让 SWD 失连。

25 电气特性

25.1 绝对最大额定值

加在芯片上的载荷如果超过“绝对组最大额定值”列表中给出的值，可能会导致芯片永久性地损坏。这里只是给出能承受的最大载荷，并不意味着在此条件下芯片的功能性操作无误。芯片长期工作在最大值条件下会影响芯片的可靠性。

表 25-1 输入电压范围

| 符号 | 描述 | 最小值 | 典型值 | 最大值 | 单位 |
|---------------------|----------------|--------------|-----|-----|----|
| $V_{VCC} - V_{VSS}$ | 外部主供电电压 | -0.3 | 3.3 | 3.6 | V |
| V_{IN} | 在 VCC 引脚上的输入电压 | $V_{SS}-0.3$ | 3.3 | 3.6 | V |
| | 在其它引脚上的输入电压 | $V_{SS}-0.3$ | 3.3 | 3.6 | |

注：

1. 所有的电源 (VCC) 和地 (VSS) 引脚必须始终连接到外部允许范围内的供电系统上。
2. 必须始终遵循 V_{IN} 的最大值。有关允许的最大注入电流值的信息。

表 25-2 电流范围

| 符号 | 描述 | 最大值 | 单位 |
|-----------|-----------------------|-----|----|
| I_{VCC} | 经过 VCC 电源线的总电流 (供应电流) | 100 | mA |
| I_{VSS} | 经过 VSS 地线的总电流 (流出电流) | 550 | mA |
| I_{IO} | 任意 I/O 和控制引脚上的输出灌电流 | 30 | mA |
| | 任意 I/O 和控制引脚上的输出电流 | | |

注：

1. 在允许的范围内，所有主电源 (VCC) 和接地 (VSS) 引脚必须始终连接到外部电源。
2. 此电流消耗必须正确分布至所有 I/O 和控制引脚。

表 25-3 温度范围

| 符号 | 描述 | 最小值 | 最大值 | 单位 |
|-----------|------|-----|-----|----|
| T_{STG} | 存储温度 | -45 | 150 | °C |
| T_J | 结温度 | -40 | 125 | °C |

25.2 工作条件/特性参数

25.2.1 通用工作条件

表 25-4 通用工作范围

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|-------------------|-------------|----|--------|---------|-----------|----|
| f_{SYSTEM} | 系统时钟 | - | 32K | - | 48M | Hz |
| f_{PLL} | 内部 PLL 输出时钟 | - | - | 48M | - | |
| f_{LIRC} | 内部低速时钟频率 | - | 95.5K | 128.3K | 183.7K | |
| f_{HIRC} | 内部高速时钟频率 | - | 25.64M | 26M | 26.36M | |
| | | 常温 | 25.9M | 26M | 26.1M | |
| f_{XOSC} | 外部时钟频率 | - | - | 32.768K | - | |
| f_{HJDRC} | 内部低速高精度时钟频率 | - | 32.27K | 32.77K | 33.26K | |
| | | 常温 | 32.64K | 32.77K | 32.89K | |
| V_{VCC} | 工作电压 | - | 1.8 | 3.3 | 3.6 | V |
| $V_{VREFP(PA13)}$ | ADC 外部参考电压 | - | 2.4 | - | V_{VCC} | |
| T_A | 环境温度 | - | -40 | - | 105 | °C |

注:

1. 非标识常温的测试数据, 表示在额定范围。
2. 对于 ADC 转换精度要求比较高的建议使用外部参考电压。
3. 对于工作电压范围, 除了 ADC 以外, 其他所有功能均可以工作在 1.8~3.6V。
4. 对于芯片内部功率较大的应用下, 建议环境温度不超过 85°C。

25.2.2 上电工作条件

表 25-5 电源上电速率

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|-------------------|----------|----------------------|-----|-----|-----|------|
| SR _{VCC} | VCC 上升速率 | T _A =27°C | 1 | - | - | us/V |

25.2.3 内嵌 PVD/POR 特性

下表中所给出的所有参数是依据室温和 V_{VCC}=3.3V 下测试得出。

表 25-6 LVD 电压档位

| 符号 | 参数 | 条件 | 最小值 | 典型值 下降/上升 | 最大值 | 单位 |
|----------------------|----------------|------------------|-----|--------------|-----|----|
| V _{PVD} | 可编程的电压检测器的电平选择 | LVDCON[6:3]=0000 | - | 1.50/1.58 | - | V |
| | | LVDCON[6:3]=0001 | - | 1.55/1.62 | - | V |
| | | LVDCON[6:3]=0010 | - | 1.60/1.68 | - | V |
| | | LVDCON[6:3]=0011 | - | 1.65/1.73 | - | V |
| | | LVDCON[6:3]=0100 | - | 1.70/1.78 | - | V |
| | | LVDCON[6:3]=0101 | - | 1.75/1.83 | - | V |
| | | LVDCON[6:3]=0110 | - | 1.80/1.88 | - | V |
| | | LVDCON[6:3]=0111 | - | 1.90/1.99 | - | V |
| | | LVDCON[6:3]=1000 | - | 2.00/2.11 | - | V |
| | | LVDCON[6:3]=1001 | - | 2.10/2.20 | - | V |
| | | LVDCON[6:3]=1010 | - | 2.20/2.31 | - | V |
| | | LVDCON[6:3]=1011 | - | 2.30/2.41 | - | V |
| | | LVDCON[6:3]=1100 | - | 2.40/2.51 | - | V |
| | | LVDCON[6:3]=1101 | - | 2.50/2.62 | - | V |
| | | LVDCON[6:3]=1110 | - | 2.60/2.73 | - | V |
| | | LVDCON[6:3]=1111 | - | 2.70/2.82 | - | V |
| V _{PVDhyst} | PVD 迟滞 | - | 80 | 100 | 140 | mV |
| V _{POR/PDR} | 上电/掉电复位 阈值 | 下降沿 | - | 0.65 | - | V |
| | | 上升沿 | - | 0.92 | - | V |
| T _{RSTHD} | 复位持续时间 | - | 1 | - | - | ms |

注:

1. 产品的特性由设计保证至最小的 V_{PVD} 数值。
2. 复位持续时间的测量方法为充上电 (POR 复位) 到用户应用代码读取第一条指令的时刻。

25.2.4 供电电流特性

电流消耗是多种参数和因素的综合指标, 这些参数和因素包括工作电压、工作环境、I/O 引脚的负载、产品的软件配置、工作频率、I/O 脚的翻转速度、程序在存储器中的位置以及执行的代码等。

本节中给出的所有运行模式下的电流消耗测量值, 都是在额定的典型供电条件下测试。

表 25-7 工作电流

| 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|---------------------------|--|-----|------|-----|----|
| $I_{\text{sleep-deep}}$ | Sleep 模式, 时钟停止, 等待 IO 唤醒 | - | 2.2 | - | uA |
| $I_{\text{sleep-wiLCD}}$ | Sleep 模式, 内部高精度 32KHz 时钟, 内置 LCD 升压, 推 4COM-17SEG 不接屏, 等待 IO 唤醒, | - | 5.5 | - | uA |
| $I_{\text{sleep-wiRTC}}$ | Sleep 模式, 外部 32KHz 晶振时钟, RTC 计时, 内置 LCD 升压, 推 4COM-17SEG 不接屏, 等待 IO 唤醒, | - | 6.8 | - | uA |
| $I_{\text{Normal-wiRTC}}$ | Normal 模式, 外部 32KHz 晶振时钟, RTC 计时, 内置 LCD 升压, 推 4COM-17SEG 不接屏, ADC 使能, 48MHz 系统时钟, | - | 12.1 | - | mA |
| | Normal 模式, 外部 32KHz 晶振时钟, RTC 计时, 内置 LCD 升压, 推 4COM-17SEG 不接屏, ADC 使能, 24MHz 系统时钟, | - | 10 | - | mA |
| | Normal 模式, 外部 32KHz 晶振时钟, RTC 计时, 内置 LCD 升压, 推 4COM-17SEG 不接屏, ADC 使能, 12MHz 系统时钟 | - | 8.6 | - | mA |
| | Normal 模式, 外部 32KHz 晶振时钟, RTC 计时, 内置 LCD 升压, 推 4COM-17SEG 不接屏, ADC 使能, 8MHz 系统时钟 | - | 8 | - | mA |
| | Normal 模式, 外部 32KHz 晶振时钟, RTC 计时, 内置 LCD 升压, 推 4COM-17SEG 不接屏, ADC 使能, 4MHz 系统时钟 | - | 7.3 | - | mA |

25.2.5 晶振驱动电路特性

表 25-8 晶振驱动电路参数

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|-------------------|---------------|----|-----|--------|------|------|
| f_{XOSC} | 外部晶振范围 | - | - | 32.768 | - | KHz |
| V_{OSCI} | 输入端偏置电压 | - | - | 571 | - | mV |
| V_{OSCO} | 输出端偏置电压 | - | - | 569 | - | mV |
| R_F | 内置偏置电阻 (默认挡位) | - | 6.8 | 8.5 | - | Mohm |
| C_L | 内置负载电容 | - | 0 | - | 20.9 | pF |
| C_F | 内置微调电容 | - | - | 0.2 | - | pF |
| I_{XOSC} | 驱动模块功耗 | - | 0.6 | - | 1.2 | uA |

注:

1. 晶振驱动电路建议使用 32.768KHz 晶振, 采用高可靠性封装晶体增强晶振起振能力。
2. 外部晶振时钟模块芯片内置电容, 包括单独可以微调 XOSCI/XOSCO 端的电容, 进行震荡频率微调。
3. 相对较低的 RF 电阻值, 可以为避免在潮湿环境下使用时所产生的问题提供保护, 这种环境下产生的泄漏和偏置条件都发生了变化。如果芯片是应用在恶劣的潮湿条件时, 设计时需要把这个参数考虑进去。
4. 芯片内置电容, 建议外部预留外接电容位置, 用于提高晶振震荡可靠性。
5. 芯片内置电容可以通过配置全部关掉, 使用外部电容。

25.2.6 内部时钟源特性

高速内部 (HIRC) 振荡器

下表中所给出的所有参数是在室温和 $V_{\text{VCC}}=3.3\text{V}$ 下测试得出。

表 25-9 内部高速振荡器参数

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|--------------|--------|--------------|---------|-----|------------|-----|
| V_{VCC} | 供电电压 | - | 1.8 | 3 | 3.6 | V |
| f_{HIRC} | 频率 | 27°C 校准后 | 25.9585 | 26 | 26.1167 | MHz |
| ACC_{HIRC} | 振荡器的精度 | -40°C至 125°C | - | - | ± 1.5 | % |
| | | -20°C至 80°C | - | - | ± 0.77 | |
| T_{STRT} | 启动时间 | - | - | - | 60 | us |
| I_{HIRC} | 振荡器功耗 | 平均功耗 | - | - | 1.5 | mA |

注

1. $V_{VCC} = 3.3V$, $T_A = -40^{\circ}C \sim 105^{\circ}C$, 除非特别说明。
2. 量产会经过校准, 校准后的频率会相应记录在 FLASH 中, 用于更精准的计时补偿。

低速高精度内部 (HJDRC) 振荡器

表 25-10 内部低速高精度振荡器参数

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|---------------|--------|-------------------------|-------|-------|------------|-----|
| V_{VCC} | 供电电压 | - | 1.8 | 3 | 3.6 | V |
| f_{HJDRC} | 频率 | $T_A = 27^{\circ}C$ 校准后 | 32.64 | 32.77 | 32.89 | kHz |
| ACC_{HJDRC} | 振荡器的精度 | -40°C至 125°C | - | - | ± 1.5 | % |
| | | -20°C至 80°C | - | - | ± 0.78 | % |
| T_{STRT} | 启动时间 | - | - | - | 10 | us |
| I_{HJDRC} | 振荡器功耗 | 平均功耗 | - | 1.8 | 2 | uA |

注:

1. $V_{VCC} = 3.3V$, $T_A = -40^{\circ}C \sim 105^{\circ}C$, 除非特别说明。
2. 低速内部振荡器测试方法和条件同高速内部振荡器, 注意关掉其他时钟源。

低速内部 (LIRC) 振荡器

表 25-11 内部低速振荡器参数

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|------------|-------|---------------------|-----|------|------|-----|
| f_{LIRC} | 频率 | $T_A = 27^{\circ}C$ | - | 128 | 145 | kHz |
| T_{STRT} | 启动时间 | - | - | - | 10 | us |
| I_{LIRC} | 振荡器功耗 | $T_A = 27^{\circ}C$ | - | 1.22 | 1.93 | uA |

注:

1. $V_{VCC} = 3.3V$, $T_A = -40^{\circ}C \sim 105^{\circ}C$, 除非特别说明
2. 低速内部振荡器测试方法和条件同高速内部振荡器, 注意关掉其他时钟源。

表 25-12 系统唤醒时间

| 符号 | 参数 | 条件 | 最大值 | 单位 |
|-----------------|----------------|--------------|------|----|
| $t_{WUSLEEP}$ | 从 sleep 模式唤醒 | I/O 唤醒 | 1.7 | ms |
| t_{WUIDLE} | 从 idle 模式唤醒 | $f = 48MHz$ | 41.6 | ns |
| | | $f = 128kHz$ | 28 | us |
| $t_{WUSTOPCLK}$ | 从 stopclk 模式唤醒 | $f = 48MHz$ | 41.6 | ns |
| | | $f = 128kHz$ | 28 | us |

注:

1. $V_{VCC} = 3.3V$, $T_A = -40^{\circ}C \sim 105^{\circ}C$, 除非特别说明
2. 唤醒时间的测量是从唤醒事件开始至用户程序读取第一条指令。

25.2.7 存储器特性

表 25-13 FLASH 操作相关参数

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|--------------------|---------|------|------|-----|------|----|
| t_{prog} | 八位的编程时间 | - | 6 | - | 7.5 | us |
| t_{ERASE} | 页擦除时间 | - | 4 | - | 5 | ms |
| t_{RC} | 读操作时间 | - | 30 | - | - | ns |
| t_{ME} | 整片擦除时间 | - | 20 | - | 40 | ms |
| I_{VCC} | 供电电流 | 读模式 | - | 3 | 4.5 | mA |
| | | 写模式 | - | - | 3.5 | mA |
| | | 擦除模式 | - | - | 2 | mA |
| V_{prog} | 编程电压 | - | 1.35 | 1.5 | 1.65 | V |

表 25-14 FLASH 操作寿命参数

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|------------------|-----------|---------------------------|-----|-----|-----|----|
| NEND | 寿命 (擦写次数) | - | - | 20 | - | 千次 |
| t_{RET} | 数据保存期限 | $T_A = 105^\circ\text{C}$ | - | 20 | - | 年 |
| | | $T_A = 25^\circ\text{C}$ | - | 100 | - | |

注:

1. 除非特别说明, 所有特性参数是在 $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$ 得到。
2. 循环测试均是在整个温度范围内下进行。
3. 擦写寿命是针对某个扇区, 单个扇区擦写次数超过 2 万次不影响其它扇区寿命。

25.2.8 电气敏感性

表 25-15 48PIN 封装 ESD 相关参数

| 符号 | 参数 | 条件 | 最小值 | 单位 |
|----------------------|------------|--------|------------|----|
| $V_{\text{ESD-HBM}}$ | 静电放电人体模型 | - | ± 4000 | V |
| $V_{\text{ESD-CDM}}$ | 静电放电充电设备模型 | - | ± 2000 | V |
| $V_{\text{ESD-MM}}$ | 静电放电机器模型 | - | ± 400 | V |
| I_{LU} | 静态门锁 | @105°C | ± 200 | mA |

表 25-16 64PIN 封装 ESD 相关参数

| 符号 | 参数 | 条件 | 最小值 | 单位 |
|----------------------|------------|--------|------------|----|
| $V_{\text{ESD-HBM}}$ | 静电放电人体模型 | - | ± 6000 | V |
| $V_{\text{ESD-CDM}}$ | 静电放电充电设备模型 | - | ± 1000 | V |
| $V_{\text{ESD-MM}}$ | 静电放电机器模型 | - | ± 400 | V |
| I_{LU} | 静态门锁 | @105°C | ± 200 | mA |

25.2.9 IO 端口特性

JS32L010 系列包含三种类型 I/O: A 类包含 PC10~PC15、PD0~PD5、PE0~PE2。B 类包含 PA0~PA15、PB0~PB15、PC0~PC9(不包含 PC5)。C 类大电流 I/O 仅包含 PC5。三种类型 I/O 的不同在于输出电流驱动能力大小和交流特性。本小节表中所给出的所有参数是依据额定的环境温度范围内的室温和 VCC 供电电压为 3.3V 下测试得出。所有的 I/O 端口都是兼容 CMOS。支持开漏输出模式。

GPIO (通用输入/输出端口) 可以吸收或输出 (灌/抽) 多达 $\pm 60\text{mA}$ 电流 (除 C 类 IO 外)。在用户应用中, I/O 脚驱动电流不能超过绝对最大额定值。所有 I/O 端口从上获取的电流总和, 加上芯片在 VCC

上获取的最大运行电流，不能超过绝对最大额定值 I_{VCC} 。所有 I/O 端口吸收并从 VSS 上流出的电流总和，加上芯片在 VSS 上流出的最大运行电流，不能超过绝对最大额定值 I_{VSS} 。

IO 通用输入特性

表 25-17 IO 静态参数

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|----------------|------------|----|-----|------|-----|------------|
| V_{IL} | 输入低电平电压 | - | - | 1.34 | - | V |
| V_{IH} | 输入高电平电压 | - | - | 1.99 | - | V |
| V_{hys} | I/O 输入电压迟滞 | - | - | 0.65 | - | V |
| I_{lkg} | 输入漏电流 | - | - | - | 0.5 | uA |
| R_{PU} | 上拉等效电阻 | - | - | 9.98 | - | k Ω |
| R_{PD} | 下拉等效电阻 | - | - | 9.98 | - | k Ω |
| C_{IO} | I/O 引脚的电容 | - | 1 | 1.5 | 2 | pF |
| $C_{IO} (PC5)$ | I/O 引脚的电容 | - | 3 | 4.2 | 6 | pF |

注：

1. 三种 I/O 的通用输入输出静态特性相同，测量典型参数时供电 3.3V。
2. 输入迟滞特性由施密特触发器产生。
3. V_{IH} 和 V_{IL} 是输入模式中，使得信号翻转的上升沿高电压和下降沿低电压。
4. PC5 属于大电流 IO 口，寄生电容比较大。

IO 输出驱动电流

表 25-18 IO 电流驱动能力

| IO 类型 | 符号 | 配置 | 最小值 | 典型值 | 最大值 | 单位 |
|-------|----------|-----|------|------|-----|----|
| A/B 类 | I_{OH} | 00 | 0.58 | 0.69 | - | mA |
| | | 01 | 8.6 | 9.8 | - | mA |
| | | 10 | 13.9 | 15.8 | - | mA |
| | | 11 | 21.8 | 24.7 | - | mA |
| | I_{OL} | 00 | 0.59 | 0.7 | - | mA |
| | | 01 | 15.6 | 17.6 | - | mA |
| | | 10 | 37.5 | 42 | - | mA |
| | | 11 | 51.7 | 57.8 | - | mA |
| C 类 | I_{OH} | 000 | 12.7 | 14.4 | - | mA |
| | | 001 | 22.6 | 25.5 | - | mA |
| | I_{OL} | 000 | 19.3 | 21.6 | - | mA |
| | | 001 | 47 | 52.8 | - | mA |
| | | 010 | 148 | 164 | - | mA |
| | | 011 | 235 | 258 | - | mA |
| | | 100 | 309 | 337 | - | mA |
| | | 101 | 374 | 405 | - | mA |
| | | 110 | 430 | 463 | - | mA |
| | | 111 | 476 | 510 | - | mA |

注：

1. 表中所给出的所有参数是依据室温和 VCC 供电电压为 3.3V 下测试得出。
2. 表中 I_{OH} 为上拉驱动电流， I_{OL} 为下拉驱动电流。
3. 工艺和温度导致的偏差：根据仿真结果可知， I_{OH} (0.9VCC) 的偏差范围为：-24.2%~39.5%。 I_{OL} (0.1VCC) 的偏差

范围为：-30%~35%。

4. C 类 IO 中的最高挡位 I_{OL} 需要考虑板级测量阻抗，测量时测量到 PIN 脚位置为准。
5. C 类 IO 中 I_{OL} 抽电流到芯片内部，需要考虑不要电流过大，限定流入芯片内部电流小于 550mA。
6. C 类 IO 实际使用推大电流，建议加一点小电阻限流或者考虑承受电流标定范围。
7. IOG1 和 LCDIO 属于 A/B 类 IO，CURRENTIO 属于 C 类 IO。

25.2.10 IO 输出交流特性

下表中所给出的所有参数是依据额定的环境温度范围内的室温和 V_{VCC} 供电电压为 3.3V 下测试得出。

表 25-19 IO 输出交流特性参数

| IO 类型 | 配置 | 最大频率 | 符号 | 参数 | 最小值 | 典型值 | 最大值 | 单位 |
|-------|-----|-------|-------------------|------|-----|------|-----|----|
| A/B 类 | 00 | 1MHz | t _{fout} | 下降时间 | - | 75 | - | ns |
| | | | t _{rout} | 上升时间 | - | 80 | - | ns |
| | 01 | 26MHz | t _{fout} | 下降时间 | - | 1.77 | - | ns |
| | | | t _{rout} | 上升时间 | - | 2.37 | - | ns |
| | 10 | | t _{fout} | 下降时间 | | 1.46 | | ns |
| | | | t _{rout} | 上升时间 | | 1.77 | | ns |
| | 11 | | t _{fout} | 下降时间 | | 1.43 | | ns |
| | | | t _{rout} | 上升时间 | | 1.44 | | ns |
| C 类 | 000 | 26MHz | t _{fout} | 下降时间 | - | 2.4 | - | ns |
| | | | t _{rout} | 上升时间 | - | 2.7 | - | ns |
| | 001 | | t _{fout} | 下降时间 | - | 2.3 | - | ns |
| | | | t _{rout} | 上升时间 | - | 2.3 | - | ns |

注：

1. I/O 端口的速度（驱动能力）可以通过 GPIOx_OSPEEDL 配置。参见本芯片参考手册中有关 GPIO 端口配置寄存器的说明。
2. 上升下降沿时间 t_{fout} 和 t_{rout} 为测量 0.1VCC 到 0.9VCC 范围定义。
3. 上升/下降沿时间测试方法：打开 I/O 推挽模式，输出方波，测试 A 类 IO 时在 PIN 脚到地间加入 16pF 电容，利用示波器查看 I/O 输出的波形。利用示波器分别测量上升沿和下降沿时间。
4. 不建议使用 C 类（PC5）高 6 挡（010、011、100、101、110、111）推较快方波，表中也没有给出高 6 挡的交流特性。
5. 如上表最大频率建议，A/B 类最低挡位建议输出方波频率低于 1MHz。其他挡位建议输出频率低于 26MHz。

25.2.11 TMR 定时器特性

下表列出的参数由设计保证

有关输入输出复用功能引脚(输出比较、输入捕获、外部时钟、PWM 输出) 的特性详情，参见 17 章。

表 25-20 定时器参数特性

| 符号 | 参数 | 条件 | 最小值 | 最大值 | 单位 |
|----------------|------------|---------------------|--------|-------------|---------------|
| $t_{res-TMR}$ | 定时器分辨时间 | - | 1 | - | $t_{TMRxCLK}$ |
| | 定时器分辨时间 | $f_{TMRxCLK}=48MHz$ | 20.8 | - | ns |
| $B_{res-TMR}$ | 定时器分辨率 | - | - | 16 | 位 |
| $t_{COUNTER}$ | 当选择 16 位计数 | - | 1 | 65536 | $t_{TMRxCLK}$ |
| | 器时钟周期 | $f_{TMRxCLK}=48MHz$ | 0.0208 | 1260 | us |
| $T_{max-CONT}$ | 最大可能的计数 | - | - | 65536 x 128 | $t_{TMRxCLK}$ |
| | | $f_{TMRxCLK}=48MHz$ | - | 161.28 | ms |

注：

1. TMRx 是一个通用的名称，代表 TIMER0/1/2/3/4/5。

25.2.12 通信接口特性

IIC 总线

除非特别说明，下表列出的参数是使用环境温度，VCC 供电电压符合额定条件测量得到。IIC 接口符合标准 IIC 通信协议，但有如下限制：SDA 和 SCL 不是“真”的引脚，当配置为开漏输出时，在引出脚和 VCC 之间的 PMOS 管被关闭，但仍然存在。IIC 接口特性列于下表，有关输入输出复用功能引脚(SDA 和 SCL) 的特性详情，参见小节 12.1。

表 25-21 IIC 接口参数特性

| 符号 | 参数 | 标准 IIC | | 快速 IIC | | 单位 |
|------------------|--------------------|--------|------|--------|-----|----|
| | | 最小值 | 最大值 | 最小值 | 最大值 | |
| $t_{w(SCLL)}$ | SCL 时钟低时间 | 4.7 | - | 1.3 | - | us |
| $t_{w(SCLH)}$ | SCL 时钟高时间 | 4.0 | - | 0.6 | - | us |
| $t_{su(SDA)}$ | SDA 建立时间 | 250 | - | 100 | - | us |
| $t_{h(SDA)}$ | SDA 数据保持时间 | 0 | - | 0 | 900 | ns |
| $t_{r(SDA)}$ | SDA 和 SCL 上升时间 | - | 1000 | 2.0 | 300 | |
| $t_{f(SDA)}$ | SDA 和 SCL 下降时间 | - | 300 | - | 300 | |
| $t_{r(STA)}$ | 开始条件保持时间 | 4.0 | - | 0.6 | - | us |
| $t_{su(STA)}$ | 重复的开始条件建立时间 | 4.7 | - | 0.6 | - | |
| $t_{su(STO)}$ | 停止条件建立时间 | 4.0 | - | 0.6 | - | |
| $t_{w(STO:STA)}$ | 停止条件至开始条件的时间（总线空闲） | 4.7 | - | 1.3 | - | |
| C_b | 每条总线的容性负载 | - | 400 | - | 400 | pF |

注：

1. 测量点设置在 CMOS 电平：0.3VCC 和 0.7VCC。

SPI 接口特性

除非特别说明，下表列出的参数是使用环境温度，VCC 供电电压符合额定条件测量得到。
有关输入输出复用功能引脚 (NSS、SCK、MOSI、MISO) 的特性详情，参见小节 12.2。

表 25-22 SPI 接口参数特性

| 符号 | 参数 | 条件 | 最小值 | 最大值 | 单位 |
|--------------------------------|---------------|-------------------------|-------------|-------------|-----|
| f_{SCK} | SPI 时钟频率 | 主模式 | 0 | 26 | MHz |
| | | 从模式 | 0 | 13 | |
| $T_{r/f-SCK}$ | SPI 时钟上升和下降时间 | 负载电容： $C_{I/O}=30pF$ | - | 8 | ns |
| $t_{su(NSS)}$ | NSS 建立时间 | 从模式 | $4t_{PCLK}$ | - | ns |
| $t_h(NSS)$ | NSS 保持时间 | 从模式 | 73 | - | ns |
| $t_{W(SCKH)}$ $t_{W(SCKL)}$ | SCK 高和低的时间 | 主模式， $4f_{PCLK}=26MHz$ | 50 | 60 | ns |
| $t_{su(SI)}$ | 数据输入建立时间，从模式 | - | 1 | - | ns |
| $t_h(SI)$ | 数据输入保持时间，从模式 | - | 3 | - | ns |
| $t_a(SO)$ | 数据输出访问时间 | 从模式， $f_{PCLK}=26MHz$ | 0 | 77 | ns |
| | | 从模式， $f_{PCLK}=17.3MHz$ | - | $4t_{PCLK}$ | ns |
| $t_{dis(SO)}$ | 数据输出禁止时间 | 从模式 | 10 | - | ns |
| $t_v(SO)$ | 数据输出有效时间 | 从模式（使能边沿之后） | - | 25 | ns |
| $t_v(MO)$ | | 主模式（使能边沿之后） | - | 3 | ns |
| $t_h(SO)$ | 数据输出保持时间 | 从模式（使能边沿之后） | 25 | - | ns |
| $t_h(MO)$ | | 主模式（使能边沿之后） | 4 | - | ns |

25.2.13 12 位 ADC 特性

下表中所给出的所有参数是在室温和 $V_{CC}=3.3V$ 下测试综合评估得出。

表 25-23 ADC 参数特性

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|------------|-----------|----|--------------|--------------|-------------------------|------|
| V_{CC} | 供电电压 | - | 2.4 | 3.3 | 3.6 | V |
| I_{VCC} | 电流消耗 | - | - | - | 0.48 | mA |
| f_{ADC} | ADC 时钟频率 | - | 0.8125 | 2 | 4 | MHz |
| V_{AIN} | 转换电压范围 | - | 0 | - | V_{CC} V_{VREFP} | V |
| R_{AIN} | 外部输入阻抗 | - | 3.2 | 4 | 4.8 | Kohm |
| C_{ADC} | 内部采样和保持电容 | - | 2.7 | 3.2 | 3.7 | pF |
| t_{STAB} | 上电时间 | - | - | $20/f_{ADC}$ | - | us |
| t_{CONV} | 总的转换时间 | - | $15/f_{ADC}$ | - | - | us |

注：

1. 在该系列产品中，ADC 参考电压在内部连接到 VCC 和 VREFP(PA13)。
2. PCB 设计建议电源的去耦必须按照下图连接。
3. 为了更稳定的采样转换建议使用 VREFP 供电，来实现更高的精准度。
4. 电容最好是瓷介电容（好的质量），应该尽可能地靠近芯片。

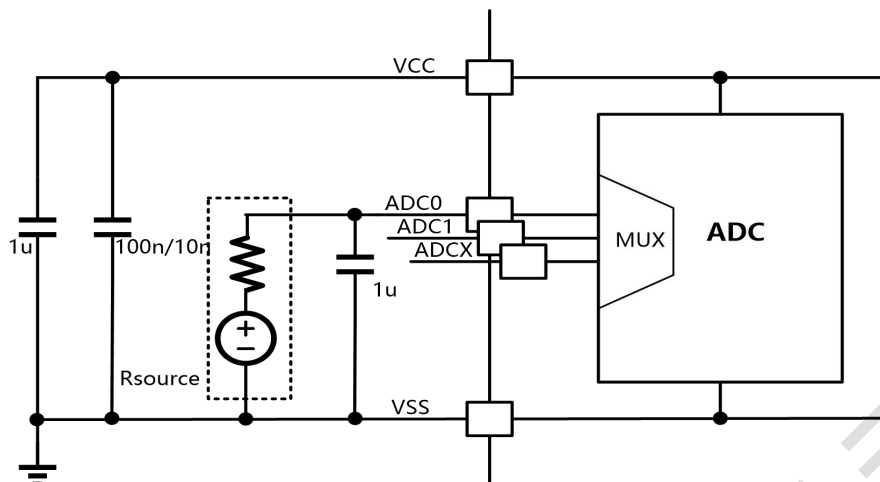


图 25-1 ADC 电源去耦

表 25-24 ADC 精度

| 符号 | 参数 | 测试条件 | 典型值 | 最大值 | 单位 |
|----|--------|------|-----|-----|-----|
| EO | 偏移误差 | - | - | ±2 | LSB |
| EG | 增益误差 | - | - | ±4 | LSB |
| ED | 微分线性误差 | - | - | ±1 | LSB |
| EL | 积分线性误差 | - | - | ±3 | LSB |

注:

EO = 偏移误差: 第一次实际转换和第一次理想转换间的偏离。

EG = 增益误差: 最后一次理想转换和最后一次实际转换间的偏离。

ED = 微分线性误差: 实际步进和理想值间的最大偏离。

EL = 积分线性误差: 任何实际转换和端点相关线间的最大偏离。

25.2.14 比较器特性

表 25-25 比较器参数

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|-----------|--------|----|-----|------|-----------|----|
| V_{VCC} | 供电电压 | - | 2.4 | 3 | 3.6 | V |
| V_{IN} | 输入电压 | - | 0 | - | V_{VCC} | |
| V_{OS} | 失调电压 | - | - | - | 9 | mV |
| T_{dly} | 传播延时 | - | - | 23.4 | 96 | ns |
| I_{CMP} | 工作电流均值 | - | - | 260 | - | uA |

注:

1. 比较器是属于 ADC 中的一部分, ADC 工作时不能单独使用比较器。
2. 比较器的负向输入端接的是 12 位 DAC, 正向输入端是 IO 输入通道。

25.2.15 温度传感器特性

所有参数是在高低温箱(-40°~125°)中, VCC 供电为 3.3V 下测试得出。

表 25-26 内置温度传感器特性

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|--------------|---------------------|--------------------------------------|------|------|------|-------|
| S_{temp} | 平均斜率 | $T_A = -40 \sim 125^{\circ}\text{C}$ | 2.56 | 2.78 | 3.02 | mV/°C |
| V_{T25} | 在 25°C 时的电压 | - | 622 | 819 | 945 | mV |
| t_{SU} | 建立时间 | - | - | - | 5 | us |
| t_{S-temp} | 当读取温度时, ADC 采样时间 | - | 5 | - | - | us |

注:

1. 温度传感器检测芯片内部温度, 量产时会把常温 27°C 时的记录到内部 FLASH 中。
2. 使用温度传感器需要使用内部 ADC 进行采样测量电压值, 再根据斜率进行温度计算。

25.2.16 8 位 DAC 特性

下表中所给出的所有参数是在室温和 $V_{VCC}=3.3\text{V}$ 下测试得出。

表 25-27 8 位 DAC 特性参数

| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|--------------|--------|----|------|------|-----------|------|
| V_{VCC} | 模拟供电电压 | - | 1.8 | 3.3 | 3.6 | V |
| I_{VCC} | 电流消耗 | - | - | 0.68 | - | mA |
| C_L | 电容负载 | - | - | 1.5 | - | pF |
| R_O | 输出阻抗 | - | 5.79 | 7.24 | 8.68 | Kohm |
| V_{DACO} | 电压输出 | - | 0 | - | V_{VCC} | V |
| DNL | 非线性误差 | - | - | - | 0.11 | LSB |
| INL | 线性误差 | - | - | - | 0.25 | LSB |
| V_{OFFSET} | 编码偏移误差 | - | - | - | 8.1 | mV |

25.2.17 12 位 DAC 特性

下表中所给出的所有参数是在室温和 $V_{VCC}=3.3\text{V}$ 下测试得出。

表 25-28 12 位 DAC 特性参数

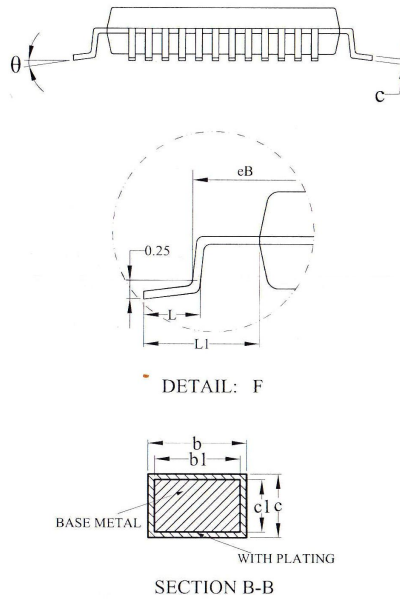
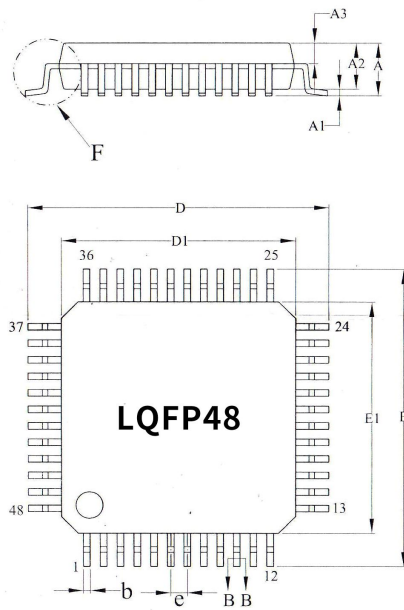
| 符号 | 参数 | 条件 | 最小值 | 典型值 | 最大值 | 单位 |
|--------------|--------|----|------|------|-----------|------|
| V_{VCC} | 模拟供电电压 | - | 1.8 | 3.3 | 3.6 | V |
| I_{VCC} | 电流消耗 | - | - | 0.3 | - | mA |
| C_L | 电容负载 | - | - | 3 | - | pF |
| R_O | 输出阻抗 | - | 8.64 | 10.8 | 12.9 | Kohm |
| V_{DACO} | 电压输出 | - | 0 | - | V_{VCC} | V |
| DNL | 非线性误差 | - | - | - | 2.11 | LSB |
| INL | 线性误差 | - | - | - | 3.12 | LSB |
| V_{OFFSET} | 编码偏移误差 | - | - | - | 1.9 | mV |

注:

1. 12 位 DAC 是 12 位 ADC 的组成部分, ADC 工作时不能使用。
2. 12 位 DAC 只有一个输出脚可以使用, 不能提供输出电流。

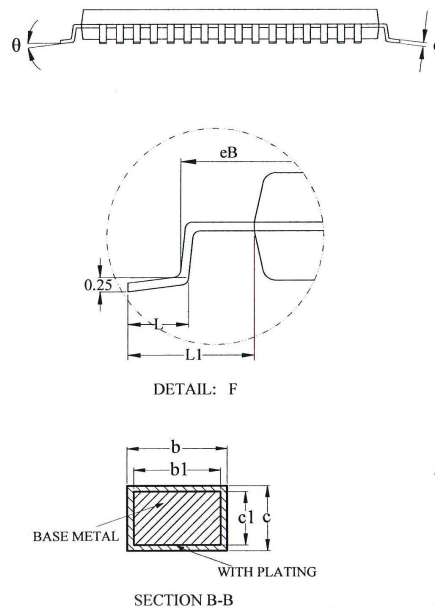
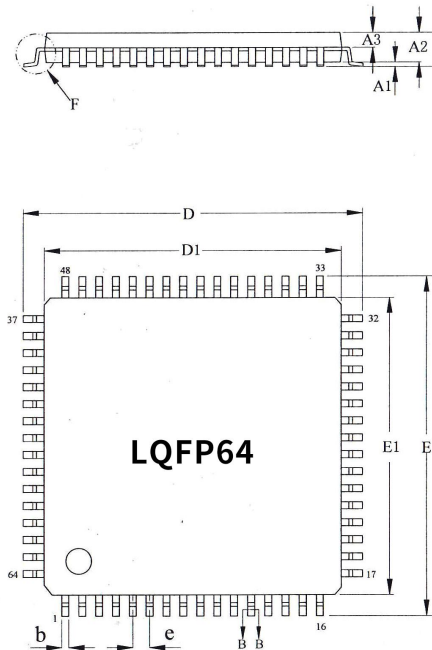
26 封装信息

26.1 LQFP48



| SYMBOL | MILLIMETER | | |
|--------|------------|------|------|
| | MIN | NOM | MAX |
| A | — | — | 1.60 |
| A1 | 0.05 | — | 0.15 |
| A2 | 1.35 | 1.40 | 1.45 |
| A3 | 0.59 | 0.64 | 0.69 |
| b | 0.18 | — | 0.26 |
| b1 | 0.17 | 0.20 | 0.23 |
| c | 0.13 | — | 0.17 |
| c1 | 0.12 | 0.13 | 0.14 |
| D | 8.80 | 9.00 | 9.20 |
| D1 | 6.90 | 7.00 | 7.10 |
| E | 8.80 | 9.00 | 9.20 |
| E1 | 6.90 | 7.00 | 7.10 |
| eB | 8.10 | — | 8.25 |
| e | 0.50BSC | | |
| L | 0.45 | — | 0.75 |
| L1 | 1.00REF | | |
| θ | 0 | — | 7° |

26.2 LQFP64



| SYMBOL | MILLIMETER | | |
|--------|------------|-------|-------|
| | MIN | NOM | MAX |
| A | — | — | 1.60 |
| A1 | 0.05 | — | 0.15 |
| A2 | 1.35 | 1.40 | 1.45 |
| A3 | 0.59 | 0.64 | 0.69 |
| b | 0.18 | — | 0.26 |
| b1 | 0.17 | 0.20 | 0.23 |
| c | 0.13 | — | 0.17 |
| c1 | 0.12 | 0.13 | 0.14 |
| D | 11.80 | 12.00 | 12.20 |
| D1 | 9.90 | 10.00 | 10.10 |
| E | 11.80 | 12.00 | 12.20 |
| E1 | 9.90 | 10.00 | 10.10 |
| eB | 11.05 | — | 11.25 |
| e | 0.50BSC | | |
| L | 0.45 | — | 0.75 |
| L1 | 1.00REF | | |
| θ | 0 | — | 7° |

26.3 SSOP24

