BL32P003xx – 高性价比 M0 单片机

数据手册
Datasheet

BL32P003 系列

32 位基于 ARM Cortex M0 核心的微控制器

目录

文档说明

由于版本升级或存在其他原因，本文档内容会不定期进行更新。除非另有约定，本文档内容仅作 为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

修改历史

版本	日期	内容	相关文档
1.0	2021.2.8	初版	《BL32P003 系列_Datasheet_V1.0》
1.1	2021.3.8	修改一些文字符号	《BL32P003 系列_Datasheet_V1.1》
1.2	2021.5.20	增加 QFN32 封装	《BL32P003 系列_Datasheet_V1.2》
1.3	2021.6.20	去掉 SOP 封装,修改一些文字	《BL32P003 系列_Datasheet_V1.3》

1. 简介

1.1 概述

本产品采用高性能的 ARM Cortex M0 为内核的 32 位微控制器，最高工作频率可达 48MHz，内置高速存储器，丰富的增强型 IO 端口和外设连接到总线。本产品包括 1 个（8 路）12 位的 ADC、2 个 16 位的定时器，可以级联为 32 位定时器、1 个 32 位的 WDT、5 路独立的 PWM、1 个简易的 RTC、1 个 SPI、1 个 IIC、3 个 UART、1 个 128bit 的 AES。

本产品系列工作电压为 2.0V~3.6V。工作温度为-40℃~85℃。多种省电工作模式保证低功耗应用的要求。

本产品提供 TSSOP20、QFN32 的封装形式，后续根据需要还会增加不同的封装形式。

- 本产品适用于以下应用场合：灯光控制、小家电、运动健康、玩具、无线充等消费电子及其周边产品；物联网智能家电、智能家居等领域。

1.2 产品特性

- 内核与系统
  - 32 位 ARM Cortex M0 处理器内核
  - 最高工作频率为 48MHz
  - 24 位的系统定时器
  - 集成嵌套向量中断控制器（NVIC）,提供最多 32 个中断
  - 通过 SWD 接口烧录程序
- 存储器
  - 内置 16K 字节的 OTP 存储器作为程序存储区
  - 内置两块共 4K RAM：其中一块 3K 作为数据存储区，另一块 1K 作为 cache 的缓存
  - 具有 CACHE 功能
- 时钟、复位及电源管理
  - 2.0V~3.6V 供电电压
  - 上电/断电复位(POR/PDR)、看门狗复位
  - 内置 4~32MHz 高频晶体振荡驱动器

- 内置经出厂调校的 24/48MHz 的高频 RC 振荡器
- 内置 32KHz 低频 RC 振荡器

- 低功耗

  standby mode、sleep mode、deepsleep mode 和 stop mode

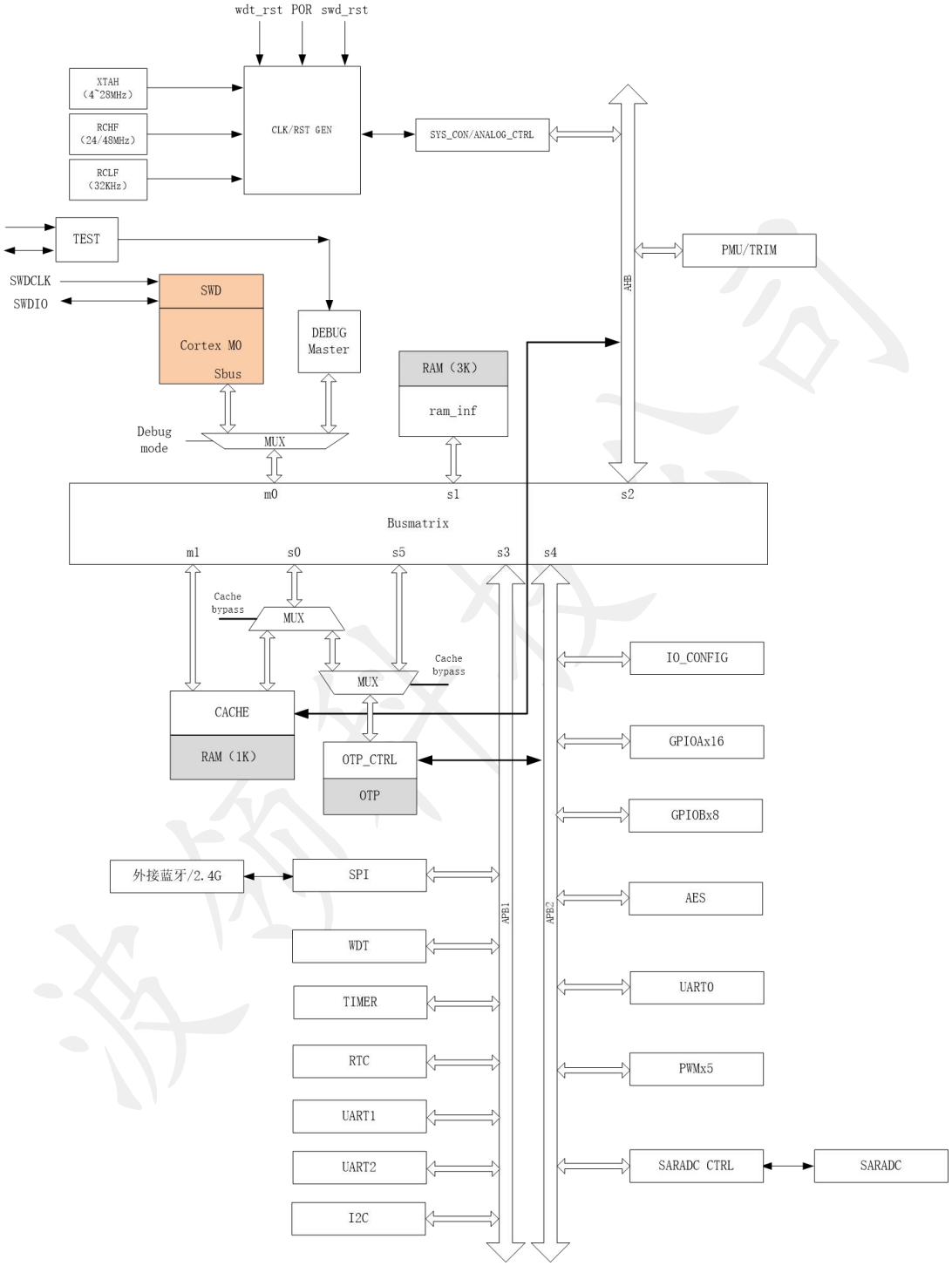
- SARADC
  - 8 通道 12bit SARADC
  - 采样率可以达到 2.4M
  - 用于电源电压和外部信号采样
- GPIO
  - 最多可达 24 个 IO 口
  - 可配置为以下模式：浮空输入、上拉输入、下拉输入、推挽输出、开漏输出、模拟 IO
  - 灵活的中断配置，可配置为电平触发和边沿触发，电平触发可设置为低电平和高电平，边沿触发可设置为上升沿、下降沿和双边沿
  - 每个 GPIO 都支持按键唤醒功能，可配置为上升沿唤醒和下降沿唤醒
- 通信接口
  - 1 个 SPI 接口，可配置主从模式，可编程时钟极性和相位，主模式速率可配置，最高频率为 Fcpu/4，数据传输顺序可配置，读写数据寄存器独立，支持乒乓传输
  - 3 个 UART 接口
  - 1 个 I2C 接口　支持主模式
- AES
  - 128bit AES，支持 ECB、CBC 和 CTR 模式
- RTC
  - 简易的 RTC 时钟模块，支持闹钟、秒、溢出中断
- 定时器
  - 2 个 16 位的定时器，可以级联为 32 位的定时器，计数时钟支持 1-256 分频
  - 1 个 32 位的独立看门狗定时器，计数时钟可选择系统时钟或 32K 时钟
  - 5 路独立的 PWM 波形发生器，计数时钟支持 4、8、16、32、64、128、256、512 分频，支持高电平结束中断和周期溢出中断
- 环境
  - 工作温度：-40℃~85℃
  - 保存温度：-40℃~125℃

- 湿度等级：MSL3

- 封装
  - QFN32
  - TSSOP20

- 应用范围
  - 灯光控制、小家电、玩具、消费电子及其周边产品；物联网智能家电、智能家居等领域

2. 芯片结构框图


wdt_rst POR swd_rst
XTAH (4~28MHz)
RCHF (24/48MHz)
RCLF (32KHz)
CLK/RST GEN
SYS_CON/ANALOG_CTRL
PMU/TRIM
AHB
TEST
SWDCLK
SWDIO
SWD
Cortex M0
Sbus
DEBUG Master
RAM (3K)
ram_inf
Debug mode
MUX
m0
s1
s2
Busmatrix
m1
s0
s5
s3
s4
Cache bypass
MUX
Cache bypass
MUX
CACHE
RAM (1K)
OTP_CTRL
OTP
IO_CONFIG
GPIOAx16
GPIOBx8
外接蓝牙/2.4G
SPI
AES
APB1
APB2
WDT
UART0
TIMER
RTC
PWMx5
UART1
UART2
SARADC CTRL
SARADC
I2C

3. 引脚定义

3.1 引脚示意图

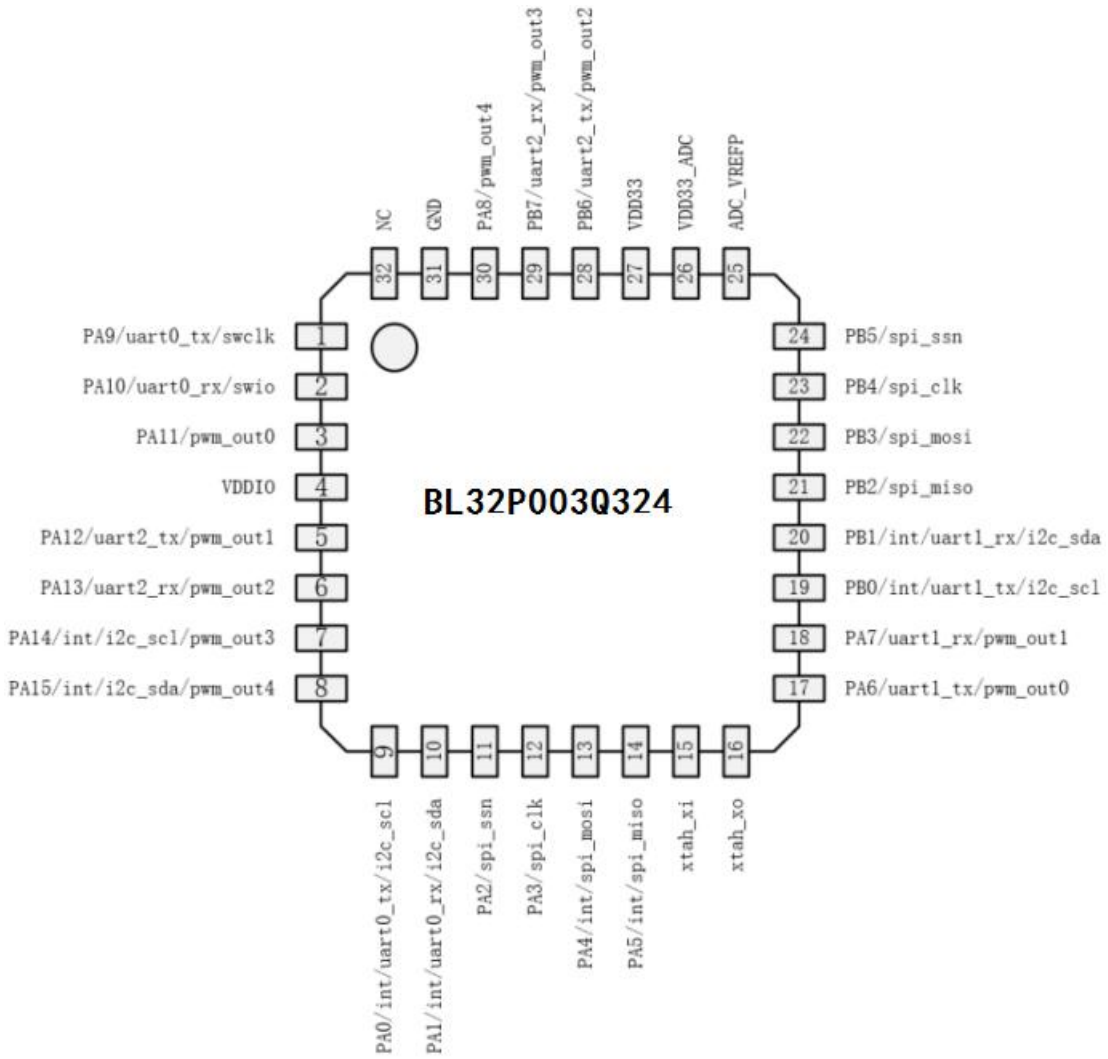


Figure 1：QFN32 PIN 脚定义图

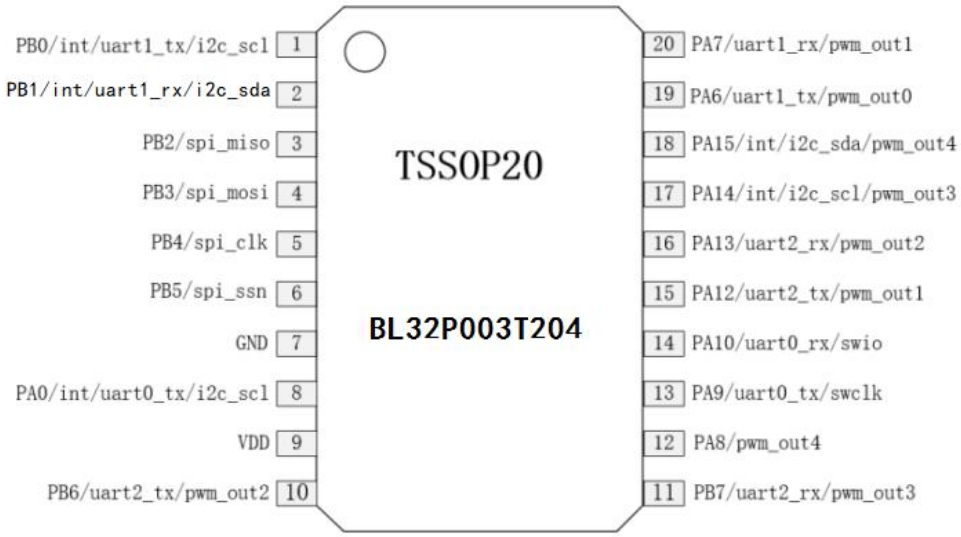


Figure 2：TSSOP20 PIN 脚定义图

3.2 引脚定义

引脚编号	引脚定义	类型	复用功能	描述
	PA0	I/O	UART0_TX IIC_SCL	PA0：数字 GPIO 功能引脚 UART0_TX：串口 0 的发送引脚 IIC_SCL：IIC 的时钟引脚
	PA1	I/O	UART0_RX IIC_SDA	PA1：数字 GPIO 功能引脚 UART0_RX：串口 0 的接收引脚 IIC_SDA：IIC 的数据引脚
	PA2	I/O	SPI_SSN	PA2：数字 GPIO 功能引脚 SPI_SSN：SPI 的片选引脚
	PA3	I/O	SPI_CLK	PA3：数字 GPIO 功能引脚 SPI_ CLK：SPI 的时钟引脚
	PA4	I/O	SPI_MOSI	PA4：数字 GPIO 功能引脚 SPI_ MOSI：SPI 的主机发送引脚
	PA5	I/O	SPI_MISO	PA5：数字 GPIO 功能引脚 SPI_ MISO：SPI 的主机接收引脚
	PA6	I/O	PWM_OUT0 UART1_TX ADC_CH7	PA6：数字 GPIO 功能引脚 PWM_OUT0：PWM 的通道 0 输出引脚 UART1_TX：串口 1 的发送引脚 ADC_CH7:ADC 的通道 7 引脚
	PA7	I/O	PWM_OUT1 UART1_RX ADC_CH6	PA7：数字 GPIO 功能引脚 PWM_OUT1：PWM 的通道 1 输出引脚 UART1_RX：串口 1 的接收引脚

				ADC_CH6:ADC 的通道 6 引脚
	PA8	I/O	PWM_OUT4	PA8：数字 GPIO 功能引脚 PWM_OUT4：PWM 的通道 4 输出引脚
	PA9	I/O	SWCLK UART0_TX	PA9：数字 GPIO 功能引脚 SWCLK:SW 下载口的时钟引脚 UART0_TX：串口 0 的发送引脚
	PA10	I/O	SWDIO UART0_RX	PA10：数字 GPIO 功能引脚 SWDIO:SW 下载口的数据引脚 UART0_RX：串口 0 的接收引脚
	PA11	I/O	PWM_OUT0	PA11：数字 GPIO 功能引脚 PWM_OUT0：PWM 的通道 0 输出引脚
	PA12	I/O	PWM_OUT1 UART2_TX	PA12：数字 GPIO 功能引脚 PWM_OUT1：PWM 的通道 1 输出引脚 UART2_TX：串口 2 的发送引脚
	PA13	I/O	PWM_OUT2 UART2_RX	PA13：数字 GPIO 功能引脚 PWM_OUT2：PWM 的通道 2 输出引脚 UART2_RX：串口 2 的接收引脚
	PA14	I/O	PWM_OUT3 IIC_SCL	PA14：数字 GPIO 功能引脚 PWM_OUT3：PWM 的通道 3 输出引脚 IIC_SCL：IIC 的时钟引脚
	PA15	I/O	PWM_OUT4 IIC_SDA	PA15：数字 GPIO 功能引脚 PWM_OUT4：PWM 的通道 4 输出引脚 IIC_SDA：IIC 的数据引脚
	PB0	I/O	UART1_TX IIC_SCL ADC_CH5	PB0：数字 GPIO 功能引脚 UART1_TX：串口 1 的发送引脚 IIC_SCL：IIC 的时钟引脚 ADC_CH5:ADC 的通道 5 引脚
	PB1	I/O	UART1_RX IIC_SDA ADC_CH4	PB1：数字 GPIO 功能引脚 UART1_RX：串口 1 的接收引脚 IIC_SDA：IIC 的数据引脚 ADC_CH4:ADC 的通道 4 引脚
	PB2	I/O	SPI_MISO ADC_CH3	PB2：数字 GPIO 功能引脚 SPI_MISO：SPI 的主机接收引脚 ADC_CH3:ADC 的通道 3 引脚
	PB3	I/O	SPI_MOSI ADC_CH2	PB3：数字 GPIO 功能引脚 SPI_ MOSI：SPI 的主机发送引脚 ADC_CH2:ADC 的通道 2 引脚
	PB4	I/O	SPI_ CLK ADC_CH1	PB4：数字 GPIO 功能引脚 SPI_ CLK：SPI 的时钟引脚 ADC_CH1:ADC 的通道 1 引脚
	PB5	I/O	SPI_ SSN ADC_CH0	PB5：数字 GPIO 功能引脚 SPI_SSN：SPI 的片选引脚 ADC_CH0:ADC 的通道 0 引脚
	PB6	I/O	PWM_OUT2	PB6：数字 GPIO 功能引脚

			UART2_TX	PWM_OUT2：PWM 的通道 2 输出引脚 UART2_TX：串口 2 的发送引脚
	PB7	I/O	PWM_OUT3 UART2_RX	PB7：数字 GPIO 功能引脚 PWM_OUT3：PWM 的通道 3 输出引脚 UART2_RX：串口 2 的接收引脚

3.3 引脚复用功能

引脚名称	SEL00	SEL01	SEL10	SEL11
PA0	GPIOA0	UART0_TX	IIC_SCL	---
PA1	GPIOA1	UART0_RX	IIC_SDA	---
PA2	GPIOA2	SPI_SSN	---	---
PA3	GPIOA3	SPI_CLK	---	---
PA4	GPIOA4	SPI_MOSI	---	---
PA5	GPIOA5	SPI_MISO	---	---
PA6	GPIOA6	PWM_OUT0	UART1_TX	ADC_CH7
PA7	GPIOA7	PWM_OUT1	UART1_RX	ADC_CH6
PA8	GPIOA8	PWM_OUT4	---	---
PA9	GPIOA9	SWCLK	UART0_TX	---
PA10	GPIOA10	SWDIO	UART0_RX	---
PA11	GPIOA11	PWM_OUT0	---	---
PA12	GPIOA12	PWM_OUT1	UART2_TX	---
PA13	GPIOA13	PWM_OUT2	UART2_RX	---
PA14	GPIOA14	PWM_OUT3	IIC_SCL	---
PA15	GPIOA15	PWM_OUT4	IIC_SDA	---
PB0	GPIOB0	UART1_TX	IIC_SCL	ADC_CH5
PB1	GPIOB1	UART1_RX	IIC_SDA	ADC_CH4
PB2	GPIOB2	SPI_MISO	---	ADC_CH3
PB3	GPIOB3	SPI_MOSI	---	ADC_CH2
PB4	GPIOB4	SPI_CLK	---	ADC_CH1
PB5	GPIOB5	SPI_SSN	---	ADC_CH0
PB6	GPIOB6	PWM_OUT2	UART2_TX	---
PB7	GPIOB7	PWM_OUT3	UART2_RX	---

4. 电气特性

本章说明芯片的电气参数，包括工作电压、工作温度、功耗、模拟特性参数及 IO 的特性参数等。

4.1 额定值

表 1 额定值

参数	符号	最小值	典型值	最大值	单位
工作电压	VDD	2.0	3.3	3.6	V
工作温度	Ti	-40	25	85	℃
存储温度	T_{STG}	-40		125	℃
ESD HBM	ESD			2000	V
ESD CDM	ESD			500	V

4.2 DC 电气特性

表 2 DC 电气特性

参数	符号	最小值	典型值	最大值	单位
标准工作电压	VDD	2	3.3	3.6	V
POR 上升阈值	VPOR_R	1.64	1.72	1.8	V
POR 下降阈值	VPOR_F	1.6	1.68	1.76	V
POR 磁滞	VPORhyst		40		mV
BandGap 精度	BG_acc	-5%		5%	%
BandGap 静态电流	BG_Iq		1	2	uA
LDO 输出电压	VLDO	1.16	1.2	1.284	V
LDO 静态电流 (sleep 模式)	Isleep_LDO		1		uA
LDO 最大负载电流(normal 模式)	Iload_max_LDO_norm			20	mA
LDO 最大负载电流(sleep 模式)	Iload max LDO_slp			100	uA
32KHz OSC 调整范围	Fosc32K_pretrim	-30		30	%
32KHz 精准度(校准后)	Acc_32K_aft_cal	-0.05		0.05	%
48MHz Osc 调整范围	Fosc48M_pretrim	-30%		30	%
48MHz 精准度(校准后)	Acc_48M_after_cal	-3		3	%
Sleep mode 电流	Isleep		30(5 analog)		uA

5. 功能描述

5.1 地址空间映射

BL32P003 系列控制器为 32 位通用控制器，提供了 4G 字节的寻址空间，如下表所示。数据格式仅支持小端模式，各模块具体寄存器排布及操作说明在后面章节有详细的描述。

起始	结束	模块
存储器		
0x00000000	0x00003FFF	CACHE CODE
0x20000000	0x20000BFF	RAM
0x41000000	0x41003FFF	OTP DATA
AHB 总线外设		
0x40000000	0x400007FF	SYSCON
0x40001000	0x400017FF	PMU
0x40002000	0x400027FF	CACHE
APB1 总线外设		
0x40041000	0x400417FF	WDT
0x40041800	0x40041FFF	TIMER
0x40042000	0x400427FF	RTC
0x40043000	0x400437FF	SPI
0x40044000	0x400447FF	UART1
0x40044800	0x40044FFF	UART2
0x40045000	0x400457FF	IIC
APB2 总线外设		
0x400A0000	0x400A07FF	IO_CONFIG
0x400A1000	0x400A17FF	GPIOA
0x400A1800	0x400A1FFF	GPIOB
0x400A3000	0x400A37FF	OTP REG
0x400A3800	0x400A3FFF	AES_128
0x400A7000	0x400A77FF	UART0
0x400AA000	0x400AA7FF	PWM
0x400AB000	0x400AB7FF	SARADC_CTRL

5.2 存储器划分及权限控制

本芯片有 16K 程序存储器 OTP 和 4K 数据存储器 RAM。

其中 16K OTP 的最高 16 个字节用于保存模拟模块的 trim 值，该 trim 值可在芯片上电后由应用程序自己读取后写入相对应的寄存器中，并上锁，以防止程序意外改写。

4K 的 RAM 在物理上共有两块：一块 1K 的 RAM 作为 CACHE 的缓存区，另一块 3K 的 RAM 作为芯片的数据区。

OTP 地址	存储内容	对应关系
3FFF~3FFC （OTP 最高 4 字节）	用于存储使用外部基准的 ADC 校准的 OFFSET 和 K 值	[6'b0, KD, 8'b0, OFFSET]
3FFB~3FF8 （OTP 次高 4 字节）	用于存储使用内部基准的 ADC 校准的 OFFSET 和 K 值	[6'b0, KD, 8'b0, OFFSET]
3FF7~3FF4 （OTP 再次高 4 字节）	用于存储 RC 振荡器相关的 trim 值	[TRIM_RC]
3FF3~3FF0 （OTP 第三次高 4 字节）	用于存储 POWER 相关的 tirm 值	[TRIM_POW]

以上 4 个字在量产测试完毕后，会写入测试后的 trim 值。该值需要应用程序读出并写入到对应的寄存器中。

5.3 中断向量表

中断向量表如下：

表 1 BL32P003 系列中断向量表

中断源	外设中断
0	UART0_IRQn
1	UART1_IRQn
2	UART2_IRQn
3	TIMER_IRQn
4	PWM_IRQn
5	ADC_IRQn
6	RTC_IRQn
7	IIC_IRQn

8	SPI_IRQn
9	AES_IRQn
10	WDT_IRQn
11	GPIOA0_IRQn
12	GPIOA1_IRQn
13	GPIOA4_IRQn
14	GPIOA5_IRQn
15	GPIOB0_IRQn
16	GPIOB1_IRQn
17	GPIOA14_IRQn
18	GPIOA15_IRQn
19	GPIOA_IRQn
20	GPIOB_IRQn

5.4 系统定时器（SYSTICK）

5.4.1 概述

CortexTM-M0 核内部提供了一个 24 位系统定时器。该定时器使能后装载当前值寄存器（SYST_VAL）内数值并向下递减至 0，并在下个时钟沿重新加载重载寄存器（SYST_LOAD）内数值。计数器再次递减至 0 时，计数器状态寄存器（SYST_CTRL）中标识位 COUNTFLAG 置位，读该位可清零。

复位后，SYST_VAL 寄存器与 SYST_LOAD 寄存器值均未知，因此使用前需初始化，向 SYST_VAL 写入任意值，清零同时复位状态寄存器，保证装载值为 SYST_LOAD 寄存器中数值。

当 SYST_LOAD 寄存器值为 0 时，重新装载后计时器保持为 0，并停止重新装载。

5.4.2 特性

- 24 位系统定时器
- 递减
- 写清零

5.4.3 模块结构框图

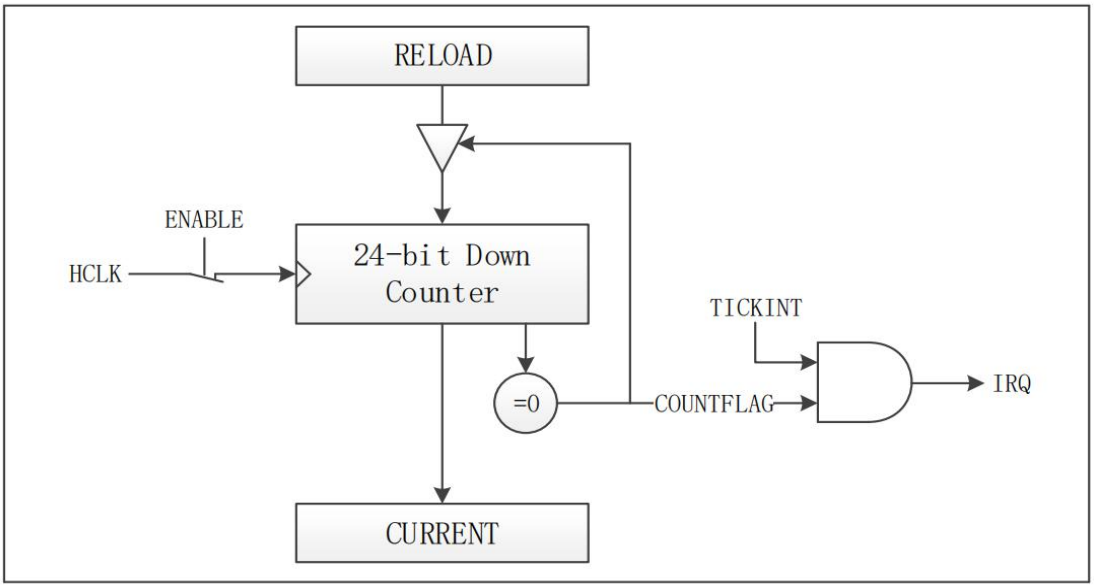


图 5-4-1 Systick 模块结构图

5.4.4 功能描述

该定时器使能后，装载当前值寄存器（SYST_VAL）内数值并向下递减至 0，并在下个时钟重新加载重载寄存器（SYST_LOAD）内数值。计数器再次递减至 0 时，计数器状态寄存器（SYST_CTRL）中的 标志位 COUNTFLAG 置位，读该位可清零。

复位后，SYST_VAL 寄存器与 SYST_LOAD 寄存器值均未知，因此使用前需初始化，向 SYST_VAL 写入任意值，清零同时复位状态寄存器，保证装载值为 SYST_LOAD 寄存器中数值。

当 SYST_LOAD 寄存器值为 0 时，重新装载后计时器保持为 0，并停止重新装载。该计数器可用作实时系统的滴答定时器或一个简单的计数器。

SysTick 计数时序图如图 5-4-2 所示。

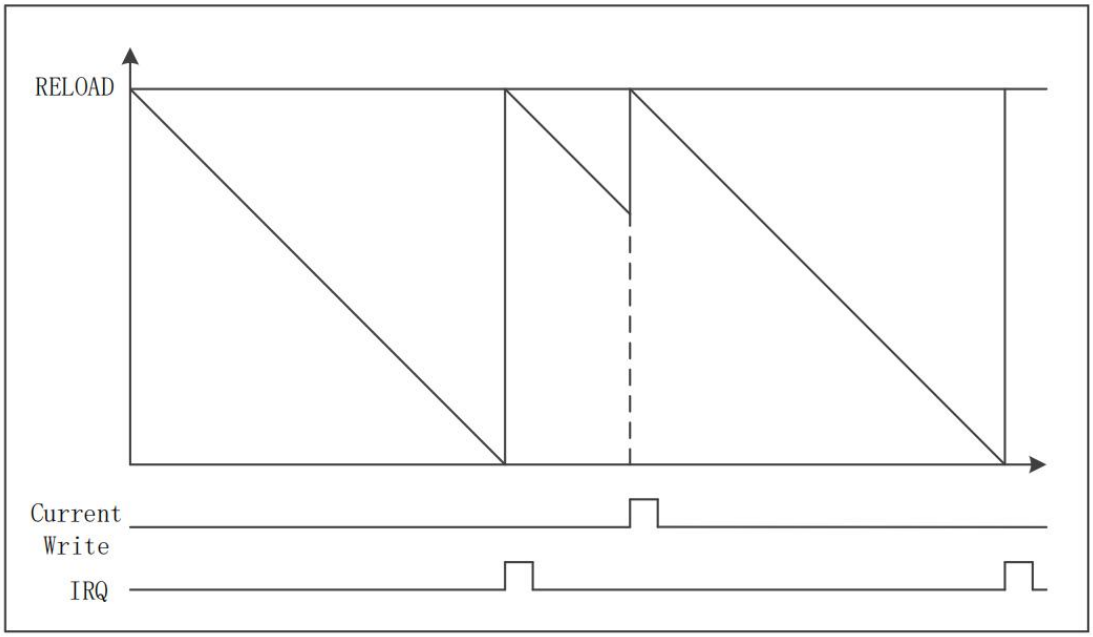


图 5-4-2 systick 计数时序图

5.4.5 寄存器映射

名称	偏移	类型	复位值	描述
SYSTICK	BASE:0xE000E010			
SYST_CTRL	0x00	R/W	0x00	状态寄存器
SYST_LOAD	0x04	R/W	---	重载寄存器
SYST_VAL	0x08	R/W	---	当前值寄存器

5.4.6 寄存器描述

SYST_CTRL 状态寄存器（0x00）

位域	名称	类型	复位值	描述
31:17	RESERVED	R	0	保留位
16	COUNTFLAG	R	0	计数器计数到 0，此位置 1
15:2	RESERVED	R	0	保留位
1	TINKINT	R/W	0	中断使能位 0：禁能

				1：使能
0	ENABLE	R/W	0	定时器使能位 0：禁能 1：使能

SYST_LOAD 重载寄存器（0x04）

位域	名称	类型	复位值	描述
31:24	RESERVED	R	0	保留位
23:0	RELOAD	R/W	0	计数器计数到 0 加载本寄存器的值，重新开始计数

SYST_VAL 当前值寄存器（0x08）

位域	名称	类型	复位值	描述
31:24	RESERVED	R	0	保留位
23:0	VAL	R/W	0	读操作返回当前计数值，写操作会清零该寄存器，同时清除 COUNTFLAG 标志位

5.5 工作模式

5.5.1 概述

工作模式有：正常工作模式（normal）、暂停模式（standby）、休眠模式（sleep）、深度休眠模式（deepsleep）和停止模式（stop）。

5.5.2 功能描述

正常工作模式：芯片完成上电并且 POR 释放后，RCHF 时钟正常产生输出（上电默认时钟频率是 24MHz），数字电路可以正常工作，CPU 开始进行正常取值，程序正常运行。

暂停模式（standby：数字不掉电，系统时钟停止，用 RCHF 作为唤醒时钟）：在正常工作模式下，当配置 STANDBY_MODE 寄存器为 1 后，芯片进入暂停模式。此时，模拟电路将不发生变化，供电和时钟不变。仅将 CPU 和外设时钟关掉，唤醒电路还是由 RCHF 作为唤醒时钟。STANDBY 模式可通过 RTC 定时和外部 IO 信号唤醒，PMU 检测到相应信号后，将 STANDBY_MODE 信号清为 0，表示退出暂停模式，回到正常工作模式。该模式下唤醒时间小于 5us。

休眠模式（sleep mode：数字不掉电，系统时钟停止）：正常工作模式下，当配置 SLEEP_MODE 寄存器为 1 后，芯片进入休眠模式。此时，模拟电路将功耗消耗比较大的模拟模块全部关闭，只保留 BG、LP_LDO 和 RCLF（32K 时钟），数字电源域由 LP_LDO 供电，系统时钟由于 RCHF 关闭而停止，此时整个数字电路只有 PMU、RTC 和 WDT 工作在 RCLF 下，其他所有数字电路由于没有时钟而停止。SLEEP 模式可通过 RTC 定时和外部 IO 信号唤醒，PMU 检测到相应信号后，将 SLEEP_MODE 信号清为 0，表示退出停止模式，模拟电路检测到 SLEEP_MODE 为 0 后，将主 LDO 打开，并且将 RCHF 打开，从而使得数字电路恢复时钟正常工作。

深度休眠模式（deepsleep：大部分数字电路掉电）：正常工作模式下，当配置 DEEPSLEEP_MODE 寄存器为 1 后，芯片进入深度休眠模式。此时，模拟电路将功耗消耗比较大的模拟模块全部关闭，只保留 BG、LP_LDO 和 RCLF（32K 时钟），数字电源域由 LP_LDO 供电，并且 pg_core 将被断开，使 CORE 电源域掉电进一步节省功耗。always on 电源域由 LDO_LP 供电，RAM 处于数据保持状态，只有 RTC、WDT 及 PMU 工作在 RCLF 时钟下。DEEPSLEEP 模式可通过 RTC 定时和外部 IO 信号唤醒，PMU 检测到相应信号后，将 DEEPSLEEP_MODE 信号清为 0，表示退出停止模式，模拟电路检测到 SLEEP_MODE 为 0 后，将主 LDO 打开，将 RCHF 打开，CORE 电源域重新上电，从而使得数字电路恢复到正常工作模式。

停止模式（stop mode：1.2V 电源关闭）：正常工作模式下，当配置 STOP_MODE 寄存器为 1 后，芯片进入停止模式。此时，所有的 LDO 全部关闭，所有 1.2V 下数字电路全部掉电，所有的 3.3V 电源域下的模拟模块全部关闭，只保留极少部分的模拟唤醒电路在工作。该模式下只有 IO 可以作为唤醒信号，当外部 IO 唤醒信号到来时，模拟唤醒电路检测到相

应唤醒信号后，开启 BG、LDO、RCHF 等所有电路，使芯片重新上电，进入正常工作模式。

5.5.3 寄存器映射

名称	偏移	类型	复位值	描述
PMU	BASE:0x40001000			
LPOW_MD	0x00	R/W	0x00	低功耗模式选择寄存器
LPMD_WKEN	0x04	R/W	0x00	低功耗唤醒源使能寄存器
LPMD_WKST	0x08	R/W	0x00	低功耗唤醒源状态寄存器
TRIM _POW	0x20	R/W	0x00	POW 相关模拟模块 TRIM 寄存器
TRIM_RC	0x24	R/W	0x00	RC 时钟模块 TRIM 寄存器
TRIM_LOCK	0x28	R/W	0x00	TRIM 锁定寄存器
RCHF_CON	0x30	R/W	0	RCHF_CON 寄存器

5.5.4 寄存器描述

LPOW_MD 寄存器（0x00）

位域	名称	类型	复位值	描述
31:4	RESERVED	R	0	保留位
3	STOP	R/W	0	向该寄存器写 1，芯片进入 STOP 模式 软件写 1，硬件自动清零
2	DEEPSLEEP	R/W	0	向该寄存器写 1，芯片进入 DEEPSLEEP 模式 软件写 1，硬件自动清零
1	SLEEP	R/W	0	向该寄存器写 1，芯片进入 SLEEP 模式 软件写 1，硬件自动清零
0	STANDBY	R/W	0	向该寄存器写 1，芯片进入 STANDBY 模式 软件写 1，硬件自动清零

注：芯片从正常工作模式进入低功耗模式，每次只能进入一种低功耗模式，唤醒后将从该低功耗模式退出回到正常工作模式，由软件再配置进入另一种低功耗模式。

LPMD_WKEN 寄存器（0x04）

位域	名称	类型	复位值	描述
31:3	RESERVED	R	0	保留位
2	IO_WKEN	R/W	0	低功耗模式下，IO 唤醒使能 0：禁能 1：使能 注 1：具体哪个 IO 具有唤醒功能可通过 PORTA_WKE 和 PORTB_WKE 寄存器进行配置
1	RTC_SEC_WKEN	R/W	0	低功耗模式下，RTC 秒信号唤醒使能 0：禁能 1：使能
0	RTC_ALA_WKEN	R/W	0	低功耗模式下，RTC 闹钟信号唤醒使能 0：禁能 1：使能

LPMD_WKST 寄存器（0x08）

位域	名称	类型	复位值	描述
31:3	RESERVED	R	0	保留位
2	IO_WKST	R/W	0	低功耗模式下，IO 唤醒标志 1：发生 IO 事件唤醒 0：未发生 IO 事件唤醒 硬件置 1，软件写 1 清除
1	RTC_SEC_WKST	R/W	0	低功耗模式下，RTC 秒唤醒标志 1：发生 RTC 秒事件唤醒 0：未发生 RTC 秒事件唤醒 硬件置 1，软件写 1 清除
0	RTC_ALA_WKST	R/W	0	低功耗模式下，RTC 闹钟唤醒标志 1：发生 RTC 闹钟事件唤醒 0：未发生 RTC 闹钟事件唤醒 硬件置 1，软件写 1 清除

TRIM_POW 寄存器（0x20）

位域	名称	类型	复位值	描述
31:23	RESERVED	R	0	保留位
22	TRIM_HPLDO_H	R/W	0	HPLDO 电压调整到 1.264v 0：不变 1：向上调整到 1.264v
21:20	TRIM_LPLDO	R/W	0	LPLDO 电压输出 trim 位
19	TRIM_PD_UVLO	R/W	0	UVLO33 trim 位
18:16	TRIM_TEMPCO_HPBG	R/W	0	HPBG 温度 trim 位
15:12	TRIM_I_HP	R/W	0	HPBG 电流 trim 位
11:8	TRIM_V_HP	R/W	0	HPBG 电压 trim 位
7:4	TRIM_V_LP	R/W	0	LPBG 电压 trim 位
3:0	TRIM_TEMPCO_LPBG	R/W	0	LPBG 温度 trim 位

该 trim 寄存器值可在芯片上电后应用程序直接读取 OTP 相应数据后直接写入相对应的寄存器中，写入后立刻生效。

TRIM_RC 寄存器（0x24）

位域	名称	类型	复位值	描述
31	RESERVED	R	0	保留位
30:28	TRIM_CS	R/W	3‘b100	RCLF CS trim 位
27:24	TRIM_FINE	R/W	0	RCLF FINE trim 位
23	RESERVED	R	0	保留位

22:20	TRIM_N	R/W	0	RCHF N trim 位
19:16	TRIM_P	R/W	0	RCHF P trim 位
15:0	RESERVED	R	0	保留位

该 trim 寄存器值可在芯片上电后应用程序直接读取 OTP 相应数据后直接写入相对应的寄存器中，写入后立刻生效。

TRIM_LOCK 寄存器（0x28）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	TRIM_LOCK	W	0	将该寄存器写入 0x55 后，TRIM_POW 和 TRIM_RC 不能被改写，用于保护对 TRIM 寄存器的误改写。

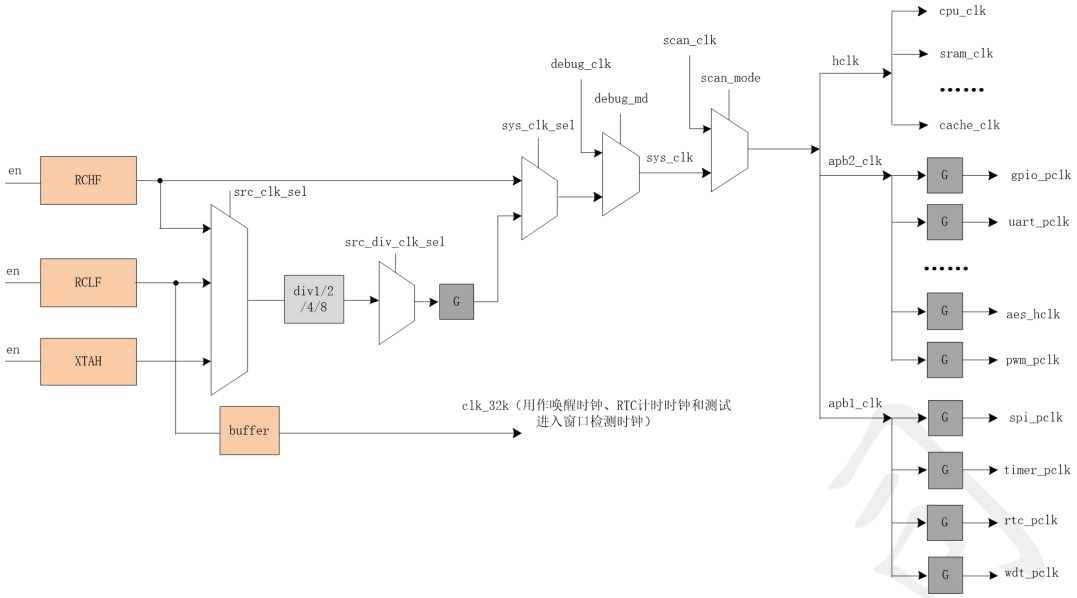
RCHF_CON 寄存器（0x30）

位域	名称	类型	复位值	描述
31:1	RESERVED	R	0	保留位
0	RCHF_EN	R/W	1	RCHF 使能 0：关闭 1：开启

5.6 时钟和复位

本芯片共有 3 个时钟源：XTAH（4~32MHz 晶体振荡器）、RCHF（24/48MHz RC 振荡器）、RCLF（32KHz RC 振荡器）。

芯片时钟结构图如下：



本芯片复位源主要有上/下电复位、看门狗复位。

5.7 系统管理（SYSCON）

5.7.1 概述

系统管理为整个芯片提供时钟源，包括系统时钟和所有外围设备时钟。该控制器还可以通过单独的外设时钟的开或关，时钟源的选择来进行功耗控制。

5.7.2 寄存器映射

名称	偏移量	类型	复位值	描述
SYSCON BASE: 0x40000000				
CLK_SEL	0x00	R/W	0	时钟选择寄存器
DIV_CLK_GATE	0x04	R/W	0	分频时钟门控寄存器
DEV_CLK_GATE	0x08	R/W	0	外设时钟门控寄存器
CHIP_RST_ST	0x10	R/W	0	芯片复位状态寄存器
XTAL _CON	0x88	R/W	0	XTAL 模拟模块控制寄存器

5.7.3 寄存器描述

CLK_SEL 寄存器（0x00）

位域	名称	类型	复位值	描述
31:9	RESERVED	R	0	保留位
8	WDT_CLK_SEL	R/W	0	WDT 计数时钟选择 0：RCLF 1：系统时钟
7:6	ADC_SMPL_CLK_SEL	R/W	0	ADC 采样时钟选择 00：系统时钟的 1 分频 01：系统时钟的 2 分频 10：系统时钟的 4 分频 11：系统时钟的 8 分频
5:4	SRC_CLK_SEL	R/W	0	源时钟（SRC_CLK）选择 00：RCHF 01：RCLF 10：XTAH 11：保留
3:1	DIV_CLK_SEL	R/W	01	分频时钟（DIV_CLK）选择 000：SRC_CLK 的 1 分频 001：SRC_CLK 的 2 分频 010：SRC_CLK 的 4 分频 011：SRC_CLK 的 8 分频 100：SRC_CLK 的 16 分频 101：SRC_CLK 的 32 分频 其他：保留

0	SYS_CLK_SEL	R/W	1	系统时钟选择 0：RCHF 时钟 1：DIV_CLK 时钟

DIV_CLK_GATE 寄存器（0x04）

位域	名称	类型	复位值	描述
31:1	RESERVED	R	0	保留位
0	DIV_CLK_GATE	R/W	1	分频时钟门控 1：分频时钟输出 0：分频时钟禁止 注：当需要改变 DIV_CLK_SEL 或 SRC_CLK_SEL 寄存器时，需要先将系统时钟切换至 RCHF，然后将该寄存器置为 0，使 DIV_CLK 关闭，最后再改变 DIV_CLK_SEL 或 SRC_CLK_SEL 的值，从而保证时钟的可靠性。

DEV_CLK_GATE 寄存器（0x08）

位域	名称	类型	复位值	描述
31:22	RESERVED	R	0	保留位
21	ADCCTRL_CLK_GATE	R/W	0	ADC_CTRL 模块时钟门控 0：禁能 1：使能
20	AES_CLK_GATE	R/W	0	AES 模块时钟门控 0：禁能 1：使能
19:17	RESERVED	R	0	保留位
16	UART2_CLK_GATE	R/W	0	UART2 模块时钟门控 0：禁能 1：使能
15	UART1_CLK_GATE	R/W	0	UART1 模块时钟门控 0：禁能 1：使能

14	UART0_CLK_GATE	R/W	0	UART0 模块时钟门控 0：禁能 1：使能
13:11	RESERVED	R	0	保留位
10	SPI_CLK_GATE	R/W	0	SPI 模块时钟门控 0：禁能 1：使能
9	PWM_CLK_GATE	R/W	0	PWM 模块时钟门控 0：禁能 1：使能
8	TIMER_CLK_GATE	R/W	0	TIMER 模块时钟门控 0：禁能 1：使能
7	RTC_CLK_GATE	R/W	0	RTC 模块时钟门控 0：禁能 1：使能
6	WDT_CLK_GATE	R/W	0	WDT 模块时钟门控 0：禁能 1：使能
5	RESERVED	R	0	保留位
4	IIC_CLK_GATE	R/W	0	IIC 模块时钟门控 0：禁能 1：使能
3:2	RESERVED	R	0	保留位
1	GPIOB_CLK_GATE	R/W	0	GPIOB 模块时钟门控 0：禁能 1：使能
0	GPIOA_CLK_GATE	R/W	0	GPIOA 模块时钟门控 0：禁能 1：使能

CHIP_RST_ST 寄存器（0x10）

位域	名称	类型	复位值	描述
31:1	RESERVED	R	0	保留位
0	WDT_RST_ST	R/W	0	WDT 复位状态标志寄存器 0：表示没有出现 WDT 复位 1：表示出现 WDT 复位 写 1 清零

XTAL_CON 寄存器（0x88）

位域	名称	类型	复位值	描述
31:1	RESERVED	R	0	保留位
0	XTAL_EN	R/W	0	XTAL 使能 0：关闭 1：开启 注：XTAL 使能后，至少需要等待 2ms 再使用

5.8 PORTCON

5.8.1 概述

端口控制模块主要包括引脚输入使能，引脚功能配置，IO 口上拉、下拉、推挽、开漏，引脚唤醒功能，唤醒边沿，驱动电流，上拉电阻可配置。

5.8.2 特性

- 配置 IO 口为特定功能
- 支持上拉、下拉、推挽、开漏配置
- 可以配置引脚的输入使能

- 可以配置引脚唤醒使能
- 引脚唤醒边沿可以配置
- 驱动电流可以配置
- 上拉电阻 32K、40K 和 150K 可以配置

5.8.3 模块结构框图

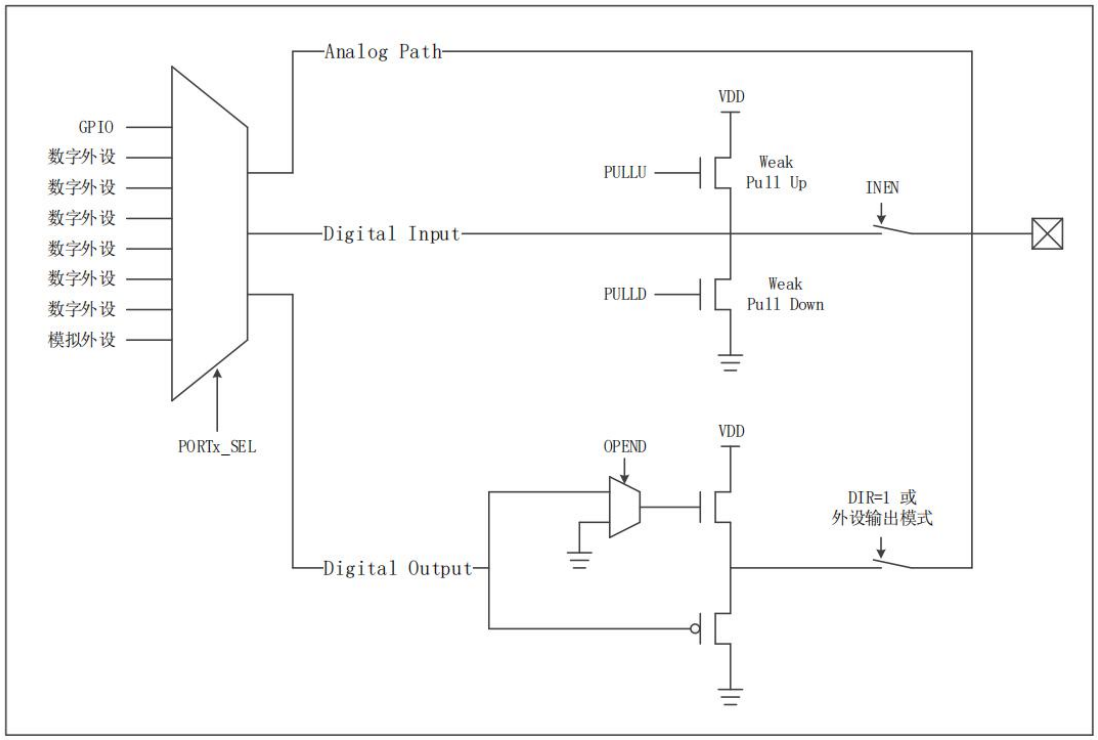


图 5.8.1 PORTCON 模块结构框图

5.8.4 功能描述

引脚输入使能

本芯片引脚作为输入或需要输入的外设时，需要打开引脚对应输入使能寄存器（PORTx_IE），当引脚所在寄存器对应位设置为 1 时，输入使能打开，引脚可获取外部状态。

复用功能选择配置

端口复用通过端口复用寄存器 PORTx_SEL 寄存器实现。当指定位配置为对应值时，

引脚功能实现切换。

每个端口可能具备以下功能：

- 通用输入输出接口：引脚作为通用输入输出功能
- 外设接口：将对应引脚切换至指定数字功能，如 TIMER/UART/PWM 等
- 模拟接口：将对应引脚切换至模拟功能，如模数转换器、时钟输入等
- 下载接口：使用仿真器连接下载程序及单步执行

配置示意图如图 5.8.2 所示。

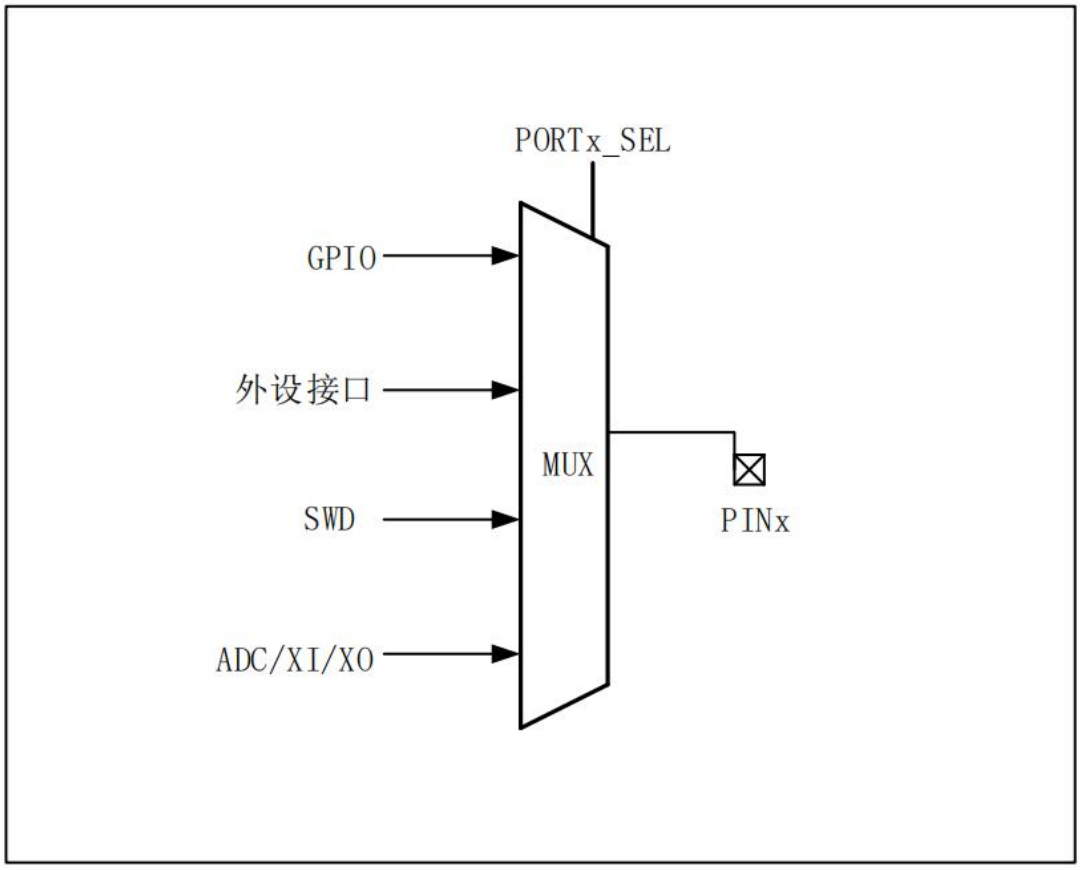


图 5.8.2 引脚配置示意图

上拉/下拉/推挽/开漏配置

本芯片每个 IO 引脚均可配置为以下模式：

- 上拉输入。
- 下拉输入。
- 推挽输出。
- 开漏输出。

注：在未配置 PORT 端口状态时，上电默认为浮空状态。

作为输入功能使用时，GPIO DIR 寄存器对应位为 0，该状态为上电默认状态。此时可以开启内部上拉和下拉功能，通过配置 PORTx_PU 及 PORTx_PD 寄存器实现，将引脚所对应寄存器指定位配置为 1，即可实现该功能。如图 5.8.3 所示。

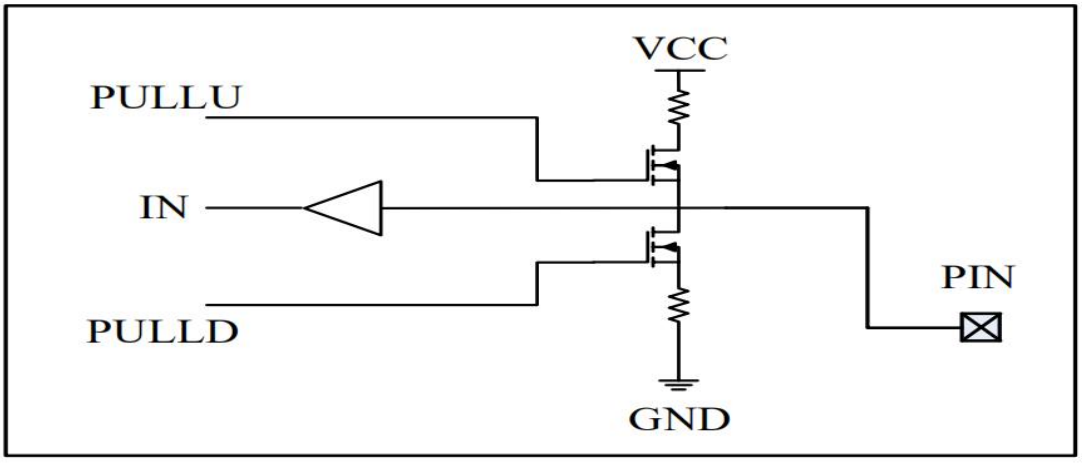


图 5.8.3 IO 输入上拉/下拉

作为输出功能使用时，GPIO DIR 寄存器对应位为 1，此时可配置引脚状态为推挽输出或开漏输出,，通过配置 PORTx_OD 寄存器实现。作为推挽输出时，PORTx_OD 寄存器对应位为 0，芯片具备拉/灌电流的能力，GPIO DATA 寄存器配置值将反映到对应引脚电平。如图 5.8.4 所示。

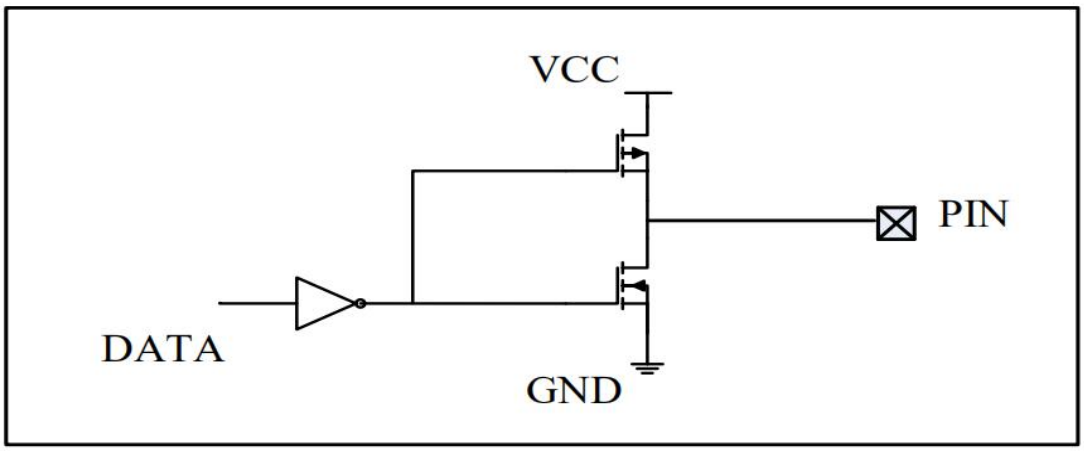


图 5.8.4 推挽输出

作为开漏输出时，PORTx_OD 寄存器对应位为 1，芯片只具备灌电流的能力，不具备拉电流能力。GPIO 输出配置为 0 时，对应引脚将输出 0，配置为 1 时，输出高阻。若需要输出 1 时，需要将外部引脚接上拉电阻，通过外部上拉实现高电平输出。示意图如图 5.8.5 所示。

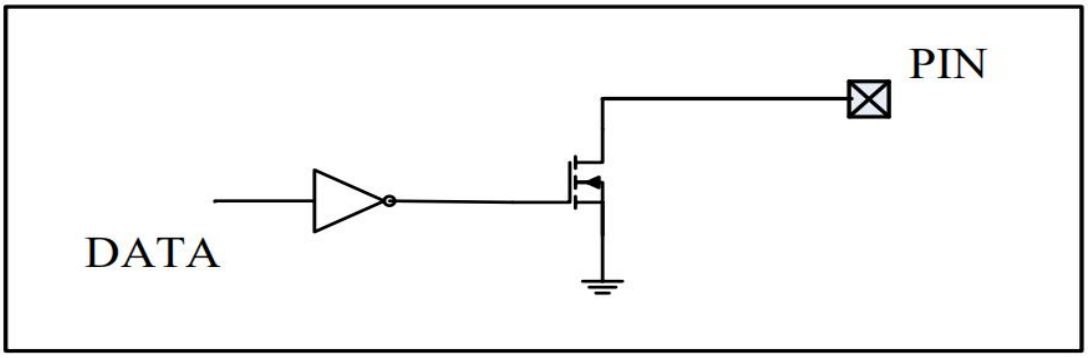


图 5.8.5 开漏输出

引脚唤醒功能

引脚可以配置为唤醒功能，可以在低功耗模式下通过引脚进行唤醒操作，唤醒的边沿可以配置为上升沿或者下降沿。

驱动电流配置

端口的驱动能力可以配置为以下四种模式：5ma 14ma 22ma 30ma。

上拉电阻配置

上拉电阻可以配置为 32K、40K、150K。

5.8.5 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
PORT BASE: 0x400A0000					
PORTA_SEL	0x00	32	R/W	0	PORTA 功能选择寄存器
PORTB_SEL	0x04	32	R/W	0	PORTB 功能选择寄存器
PORTA_IE	0x100	32	R/W	0	PORTA 输入使能寄存器
PORTB_IE	0x104	32	R/W	0	PORTB 输入使能寄存器
PORTA_PU	0x200	32	R/W	0	PORTA 上拉使能寄存器
PORTB_PU	0x204	32	R/W	0	PORTB 上拉使能寄存器
PORTA_PD	0x300	32	R/W	0	PORTA 下拉使能寄存器
PORTB_PD	0x304	32	R/W	0	PORTB 下拉使能寄存器

PORTA_OD	0x400	32	R/W	0	PORTA 开漏使能寄存器
PORTB_OD	0x404	32	R/W	0	PORTB 开漏使能寄存器
PORTA_WKE	0x500	32	R/W	0	PORTA 唤醒使能寄存器
PORTB_WKE	0x504	32	R/W	0	PORTB 唤醒使能寄存器
PORT_CFG	0x700	32	R/W	0	PORT 配置寄存器

5.8.6 寄存器描述

PORTA_SEL 寄存器（0x00）

位域	名称	类型	复位值	描述
31:30	Bit15	R/W	0	00：GPIOA15 01：PWM_OUT4 10：IIC_SDA 11：保留
29:28	Bit14	R/W	0	00：GPIOA14 01：PWM_OUT3 10：IIC_SCL 11：保留
27:26	Bit13	R/W	0	00：GPIOA13 01：PWM_OUT2 10：UART2_RX 11：保留
25:24	Bit12	R/W	0	00：GPIOA12 01：PWM_OUT1 10：UART2_TX 11：保留

23:22	Bit11	R/W	0	00：GPIOA11 01：PWM_OUT0 10：保留 11：保留
21:20	Bit10	R/W	01	00：GPIOA10 01：SWDIO 10：UART0_RX 11：保留
19:18	Bit9	R/W	01	00：GPIOA9 01：SWCLK 10：UART0_TX 11：保留
17:16	Bit8	R/W	0	00：GPIOA8 01：PWM_OUT4 10：保留 11：保留
15:14	Bit7	R/W	0	00：GPIOA7 01：PWM_OUT 1 10：UART1_RX 11：ADC_CH6
13:12	Bit6	R/W	0	00：GPIOA6 01：PWM_OUT0 10：UART1_TX 11：ADC_CH7
11:10	Bit5	R/W	0	00：GPIOA5 01：SPI_MISO 10：保留 11：保留

9:8	Bit4	R/W	0	00：GPIOA4 01：SPI_MOSI 10：保留 11：保留
7:6	Bit3	R/W	0	00：GPIOA3 01：SPI_CLK 10：保留 11：保留
5:4	Bit2	R/W	0	00：GPIOA2 01：SPI_SSN 10：保留 11：保留
3:2	Bit1	R/W	0	00：GPIOA1 01：UART0_RX 10：IIC_SDA 11：保留
1:0	Bit0	R/W	0	00：GPIOA0 01：UART0_TX 10：IIC_SCL 11：保留

PORTB_SEL 寄存器（0x04）

位域	名称	类型	复位值	描述
31:16	RESERVED	R	0	保留位
15:14	Bit7	R/W	01	00：GPIOB7 01：PWM_OUT3 10：UART2_RX 11：保留

13:12	Bit6	R/W	01	00：GPIOB6 01：PWM_OUT2 10：UART2_TX 11：保留
11:10	Bit5	R/W	0	00：GPIOB5 01：SPI_SSN 10：保留 11：ADC_CH0
9:8	Bit4	R/W	0	00：GPIOB4 01：SPI_CLK 10：保留 11：ADC_CH1
7:6	Bit3	R/W	0	00：GPIOB3 01：SPI_MOSI 10：保留 11：ADC_CH2
5:4	Bit2	R/W	0	00：GPIOB2 01：SPI_MISO 10：保留 11：ADC_CH3
3:2	Bit1	R/W	0	00：GPIOB1 01：UART1_RX 10：IIC_SDA 11：ADC_CH4
1:0	Bit0	R/W	0	00：GPIOB0 01：UART1_TX 10：IIC_SCL 11：ADC_CH5

PORTA_IE 寄存器（0x100）

位域	名称	类型	复位值	描述
31:16	RESERVED	R	0	保留位
15:0	PORTA_IE	R/W	0x600	PORTA 输入使能寄存器 0：禁能 1：使能 （每个 bit 对应 1 个 IO，bit0 对应 A0,bit1 对应 A1）

PORTB_IE 寄存器（0x104）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	PORTB_IE	R/W	0	PORTB 输入使能寄存器 0：禁能 1：使能 （每个 bit 对应 1 个 IO，bit0 对应 B0,bit1 对应 B1）

PORTA_PU 寄存器（0x200）

位域	名称	类型	复位值	描述
31:16	RESERVED	R	0	保留位
15:0	PORTA_PU	R/W	0	PORTA 上拉使能寄存器 0：禁能 1：使能 （每个 bit 对应 1 个 IO，bit0 对应 A0,bit1 对应 A1）

PORTB_PU 寄存器（0x204）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	PORTB_PU	R/W	0	PORTB 上拉使能寄存器 0：禁能 1：使能 （每个 bit 对应 1 个 IO，bit0 对应 B0,bit1 对应 B1）

PORTA_PD 寄存器（0x300）

位域	名称	类型	复位值	描述
31:16	RESERVED	R	0	保留位
15:0	PORTA_PD	R/W	0	PORTA 下拉使能寄存器 0：禁能 1：使能 （每个 bit 对应 1 个 IO，bit0 对应 A0,bit1 对应 A1）

PORTB_PD 寄存器（0x304）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	PORTB_PD	R/W	0	PORTB 下拉使能寄存器 0：禁能 1：使能 （每个 bit 对应 1 个 IO，bit0 对应 B0,bit1 对应 B1）

PORTA_OD 寄存器（0x400）

位域	名称	类型	复位值	描述
31:16	RESERVED	R	0	保留位
15:0	PORTA_OD	R/W	0	PORTA 开漏使能寄存器 0：禁能 1：使能 （每个 bit 对应 1 个 IO，bit0 对应 A0,bit1 对应 A1）

PORTB_OD 寄存器（0x404）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	PORTB_OD	R/W	0	PORTB 开漏使能寄存器 0：禁能 1：使能 （每个 bit 对应 1 个 IO，bit0 对应 B0,bit1 对应 B1）

PORTA_WKE 寄存器（0x500）

位域	名称	类型	复位值	描述
31:16	RESERVED	R	0	保留位
15:0	PORTA_WKE	R/W	0	PORTA 唤醒使能寄存器 0：禁能 1：使能 （每个 bit 对应 1 个 IO，bit0 对应 A0,bit1 对应 A1）

PORTB_WKE 寄存器（0x504）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	PORTB_WKE	R/W	0	PORTB 唤醒使能寄存器 0：禁能 1：使能 （每个 bit 对应 1 个 IO，bit0 对应 B0,bit1 对应 B1）

PORT_CFG 寄存器（0x700）

位域	名称	类型	复位值	描述
31:12	RESERVED	R	0	保留位
11	PORT_WK_RF	R/W	0	PORT 唤醒功能沿功能选择 0：下降沿 1：上升沿
10	PORT_HYS	R/W	0	PORT 输入迟滞等级选择 0：低输入迟滞（输入信号大于 0.7VDD 和小于 0.3VDD） 1：高输入迟滞（输入信号大于 0.85VDD 和小于 0.15VDD）

9:8	PORT_PUR	R/W	10	PORT 上拉电阻阻值选择 00：保留 01：150kΩ 10：40kΩ 11：32kΩ
7:6	RESERVED	R	0	保留位
5:4	PORTB0_7_DS	R/W	01	PORT B0_7 驱动能力选择寄存器 00：5mA 01：14mA 10：22mA 11：30mA
3:2	PORTA8_15_DS	R/W	01	PORT A8_15 驱动能力选择寄存器 00：5mA 01：14mA 10：22mA 11：30mA
1:0	PORTA0_7_DS	R/W	01	PORT A0_7 驱动能力选择寄存器 00：5mA 01：14mA 10：22mA 11：30mA

5.9 通用 IO(GPIO)

5.9.1 概述

通用输入输出模块主要功能包括数据控制、中断控制功能。使用前需使能对应 GPIO 模块时钟。

5.9.2 特性

- 最多 24 个独立 IO
- 特定的 IO 专用中断入口
- 中断触发条件可配置，支持电平触发和边沿触发
- 电平触发支持高电平和低电平
- 边沿触发支持上升沿、下降沿和双边沿触发
- 每个 IO 均支持上拉、下拉、推挽、开漏功能

5.9.3 功能描述

方向控制

除 SWD 引脚外，所有引脚上电后默认状态均为 GPIO 浮空输入（DIR = 0）。

GPIO 方向寄存器（DIRx）用来将每个独立的管脚配置为输入模式或者输出模式：

- 当数据方向设为 0 时，GPIO 对应引脚配置为输入

通过读取相应数据寄存器（DATA）对应位获取指定 GPIO 端口当前状态值

- 当数据方向设为 1 时，GPIO 对应引脚配置为输出

通过向对应端口数据寄存器（DATA）对应位写入值改变指定引脚输出，0 输出低电平，1 输出高电平。

中断配置和清除

特定的 IO 专用中断，本芯片的 GPIOA0、GPIOA1、GPIOA4、GPIOA5、GPIOA14、GPIOA15、GPIOB0、GPIOB1 有相应的专用中断向量，GPIOA 和 GPIOB 中断向量则所有的 IO 都可以触发。

具体详情可以查看中断向量表。

可根据需求将 GPIO 端口对应引脚配置为中断模式，并通过相关寄存器配置中断极性及触发方式。

触发方式分为边沿触发和电平触发两种模式。

- 对于边沿触发中断，可以设置为上升沿触发，下降沿触发或双边沿触发。中断发生

后，标志位具备保持特性，必须通过软件对中断标志位进行清除

● 对于电平触发中断，当外部引脚输入为指定电平时，中断发生。当电平翻转后，中断信号消失，无需软件进行清除。使用电平触发中断，需保证外部信号源保持电平稳定，以便有效中断电平能被端口识别

使用以下寄存器来对产生中断触发方式和极性进行定义：

● GPIO 中断触发方式寄存器（INTLVLTRG），用于配置电平触发或边沿触发

● GPIO 中断触发极性寄存器（INTRISEEN），用于配置电平或边沿触发极性

● GPIO 中断边沿触发配置寄存器（INTBE），选择为边沿触发后，用于配置单边沿触发或双边沿触发

通过 GPIO 中断使能寄存器（INTEN）可以使能或者禁止相应端口对应位中断，GPIO 原始中断状态（INTRAWSTAUS）不受使能位影响。当产生中断时，可以在 GPIO 原始中断状态（RAWINTSTAUS）获取中断信号的状态。当中断使能寄存器（INTEN）对应位为 1 时，中断状态（INTSTAUS）寄存器可读取到对应中断信号，且中断信号会进入中断配置模块及 NVIC 模块，执行中断程序。

通过写 1 到 GPIO 中断清除寄存器（INTCLR）指定位可以清除相应位中断。

5.9.4 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
GPIOA BASE: 0x400A1000 GPIOB BASE: 0x400A1800					
GPIODATA	0x00	GPIO_WIDTH	R/W	0	数据寄存器
GPIODIR	0x04	GPIO_WIDTH	R/W	0	方向设置寄存器
INTLVLTRG	0x08	GPIO_WIDTH	R/W	0	中断检测方式寄存器
INTBE	0x0c	GPIO_WIDTH	R/W	0	沿触发方式寄存器
INTRISEEN	0x10	GPIO_WIDTH	R/W	0	中断事件方式寄存器
INTEN	0x14	GPIO_WIDTH	R/W	0	中断使能寄存器
INTRAWSTAUS	0x18	GPIO_WIDTH	R	0	中断原始状态寄存器
INTSTAUS	0x1c	GPIO_WIDTH	R	0	中断状态寄存器

INTCLR	0x20	GPIO_WIDTH	W	0	沿触发中断清除寄存器

GPIOA 的 GPIO_WIDTH 为 16，GPIOB 的 GPIO_WIDTH 为 8。

5.9.5 寄存器描述

GPIODATA 寄存器（0x00）

位域	名称	类型	复位值	描述
31:GPIO_WIDTH	RESERVED	RO	0	保留位
GPIO_WIDTH	GPIODATA	R/W	0	数据寄存器

GPIODIR 寄存器（0x04）

位域	名称	类型	复位值	描述
31:GPIO_WIDTH	RESERVED	RO	0	保留位
GPIO_WIDTH	GPIODIR	R/W	0	设置 GPIO 管脚方向： 1：设置相应位的 GPIO 管脚为输出管脚 0：设置相应位的 GPIO 管脚为输入管脚

INTLVLTRG 寄存器（0x08）

位域	名称	类型	复位值	描述
31:GPIO_WIDTH	RESERVED	RO	0	保留位
GPIO_WIDTH	INTLVLTRG	R/W	0	设置 GPIO 管脚中断敏感条件： 1：设置相应位的 GPIO 管脚为电平检测 0：设置相应位的 GPIO 管脚为沿检测

INTBE 寄存器（0x0C）

位域	名称	类型	复位值	描述
31:GPIO_WIDTH	RESERVED	RO	0	保留位

GPIO_WIDTH	INTBE	R/W	0	设置 GPIO 管脚沿触发方式： 1：设置相应位的 GPIO 管脚为双沿触发中断，即上升沿和下降沿都会触发中断 0：设置相应位的 GPIO 管脚为单沿触发中断，由 INTRISEEN 寄存器相应位确定是上升沿/下降沿触发

INTRISEEN 寄存器（0x10）

位域	名称	类型	复位值	描述
31:GPIO_WIDTH	RESERVED	RO	0	保留位
GPIO_WIDTH	INTRISEEN	R/W	0	设置 GPIO 管脚中断事件方式： 1：设置相应位的 GPIO 管脚为上升沿/高电平触发中断 0：设置相应位的 GPIO 管脚为下降沿/低电平触发中断

INTEN 寄存器（0x14）

位域	名称	类型	复位值	描述
31:GPIO_WIDTH	RESERVED	RO	0	保留位
GPIO_WIDTH	INTEN	R/W	0	设置 GPIO 管脚中断使能： 1：设置相应位的 GPIO 管脚中断使能 0：设置相应位的 GPIO 管脚中断禁止

INTRAWSTAUS 寄存器（0x18）

位域	名称	类型	复位值	描述
31:GPIO_WIDTH	RESERVED	RO	0	保留位

GPIO_WIDTH	INTRAWSTAUS	R	0	当 GPIO 被配置为输入模式时，根据设置的触发条件产生中断标志，不受中断使能寄存器的影响。由硬件置位，软件向 GPIOIC 写 1 清除。 1：表示检测到相应位的 GPIO 中断触发条件(原始，掩码之前) 0：表示没有检测到相应位的 GPIO 中断触发条件

INTSTAUS 寄存器（0x1C）

位域	名称	类型	复位值	描述
31:GPIO_WIDTH	RESERVED	RO	0	保留位
GPIO_WIDTH	INTSTAUS	R	0	当 GPIO 被配置为输入模式，且相应位的中断被使能时，根据设置的触发条件产生中断标志。由硬件置位，软件向 GPIOIC 写 1 清除。 1：表示检测到相应位的 GPIO 管脚产生的中断(掩码之后) 0：表示没有检测到相应位的 GPIO 管脚产生的中断

INTCLR 寄存器（0x20）

位域	名称	类型	复位值	描述
31:GPIO_WIDTH	RESERVED	RO	0	保留位

GPIO_WIDTH	INTCLR	W	0	写 1：清除相应位 GPIO 管脚沿触发中断标志 INTRAWSTAUS 和 INTSTAUS。清除中断标志后，INTCLR 相应位硬件自动恢复为 0。 写 0：没有影响。 对该寄存器进行读操作，返回值为 0。

5.10 基本定时器（TIMER）

5.10.1 概述

基本定时器模块具备定时功能，具有一个 8 位分频器，支持中断，2 个 16 位的定时器，可以级联为 1 个 32 位的定时器使用，使用前需要使能定时器模块时钟。

5.10.2 特性

- 2 路 16 位定时器
- 可以级联为 1 路 32 位定时器
- 8 位预分频
- 支持中断

5.10.3 功能描述

定时时长计算公式

基本定时器作为独立的两个 16 位定时器时，递增计数，计数源为系统时钟 Sys。

定时时长 Tout 计算公式如下：

Tout = (Tpre + 1)*(Tload + 1)/Sys。

注：Tpre 为分频系数，Tload 为装载值的高 16 位或者低 16 位，Sys 为系统时钟。

基本定时器级联作为 1 个 32 位定时器时，递增计数，计数源为系统时钟 Sys。

定时时长 Tout 计算公式如下：

Tout = (Tpre + 1)*(Tloadh * 0x1000 +(Tloadl + 1)) / Sys。

注：Tpre 为分频系数，Tloadh 为装载值的高 16 位，Tloadl 为装载值的低 16 位，Sys 为系统时钟。

使用流程

- 通过预分频寄存器（TIMER_DIV）设置预分频目标值（8bit,1-256）,对系统时钟进行分频。
- 通过装载值寄存器（TIMER_LOAD）设置计数目标值（16bit）。这里分为高 16 位和低 16 位两个定时器，可以根据相应的需求进行配置。
- 通过级联控制寄存器（TIMER_CTR）来配置是否开启级联功能。
- 通过中断使能寄存器（TIMER_IE）配置中断使能。
- 通过使能寄存器（TIMER_EN）进行相应定时器的使能。
- 对应的定时器开始递加计数，当计数到装载值时，产生中断，同时重新从 0 开始计数，进入下一个周期的计数。
- 中断可以通过中断状态寄存器（TIMER_ST）进行查询（中断使能的情况下），同时可以对寄存器进行写 1 操作清除中断
- 在计数过程中，可以通过对当前计数值寄存器（TIMER_CNT）进行读取，获取当前计数值。

5.10.4 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
TIMER	BASE: 0x40041800				
TIMER_EN	0x00	32	R/W	0	TIMER 使能寄存器
TIMER_DIV	0x04	32	R/W	0	TIMER 计数时钟分频寄存器
TIMER_CTR	0x08	32	R/W	0	TIMER 配置寄存器
TIMER_IE	0x10	32	R/W	0	TIMER 中断使能寄存器
TIMER_ST	0x14	32	R/W	0	TIMER 中断状态寄存器

TIMER_LOAD	0x80	32	R/W	0	TIMER 目标配置寄存器
TIMER_CNT	0x84	32	R/W	0	TIMER 当前计数值寄存器

5.10.5 寄存器描述

TIMER_EN 寄存器（0x00）

位域	名称	类型	复位值	描述
31:2	RESERVED	R	0	保留位
1	TIMERH_EN	R/W	0	TIMER 高 16bit 定时器使能寄存器 0：禁能 1：使能 注：当配置为级联模式时，该位无效
0	TIMERL_EN	R/W	0	TIMER 低 16bit 定时器使能寄存器 0：禁能 1：使能 注：当配置为级联模式时，整个 32bit 定时器使能由该位来控制

TIMER_DIV 寄存器（0x04）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7：0	TIMER_DIV	R/W	0	TIMER 计数时钟分频寄存器 0x00：表示 1 分频 0x01：表示 2 分频 …… 0xff：表示 256 分频

TIMER_CTR 寄存器（0x08）

位域	名称	类型	复位值	描述

31:1	RESERVED	R	0	保留位
0	TIMER_CC	R/W	0	TIMER 级联控制寄存器 0：表示低 16bit 定时器和高 16bit 定时器独立 1：表示低 16bit 定时器和高 16bit 定时器级联

TIMER_IE 寄存器（0x10）

位域	名称	类型	复位值	描述
31:3	RESERVED	R	0	保留位
2	TIMER_IE	R/W	0	TIMER32bit 定时器中断使能寄存器 0：禁能 1：使能 注：只有配置为级联模式时，该位才有效
1	TIMERH_IE	R/W	0	TIMER 高 16bit 定时器中断使能寄存器 0：禁能 1：使能 注：当配置为级联模式时，该位无效
0	TIMERL_IE	R/W	0	TIMER 低 16bit 定时器中断使能寄存器 0：禁能 1：使能 注：当配置为级联模式时，该位无效

TIMER_ST 寄存器（0x14）

位域	名称	类型	复位值	描述
31:3	RESERVED	R	0	保留位
2	TIMER_ST	R/W	0	TIMER32bit 定时器定时状态寄存器 写 1 清零 注：只有配置为级联模式时，该位才有效
1	TIMERH_ST	R/W	0	TIMER 高 16bit 定时器定时状态寄存器 写 1 清零 注：当配置为级联模式时，该位无效

0	TIMERL_ST	R/W	0	TIMER 低 16bit 定时器定时状态寄存器 写 1 清零 注：当配置为级联模式时，该位无效

TIMER_LOAD 寄存器（0x80）

位域	名称	类型	复位值	描述
31:16	TIMERH_LOAD	R/W	0	TIMER 高 16bit 定时器目标配置寄存器 当高 16bit 计数器向上计数达到该寄存器设定值后，会产生状态信号
15:0	TIMERL_LOAD	R/W	0	TIMER 低 16bit 定时器目标配置寄存器 当低 16bit 计数器向上计数达到该寄存器设定值后，会产生状态信号

注：当配置为级联模式时，TIMERL_LOAD 和 TIMERH_LOAD 会合并作为一个整体 32bit 定时器的目标配置寄存器。

TIMER_CNT 寄存器（0x84）

位域	名称	类型	复位值	描述
31:16	TIMERH_CNT	R	0	TIMER 高 16bit 定时器当前计数值寄存器
15:0	TIMERL_CNT	R	0	TIMER 低 16bit 定时器当前计数值寄存器

5.11 看门狗时钟（WDT）

5.11.1 概述

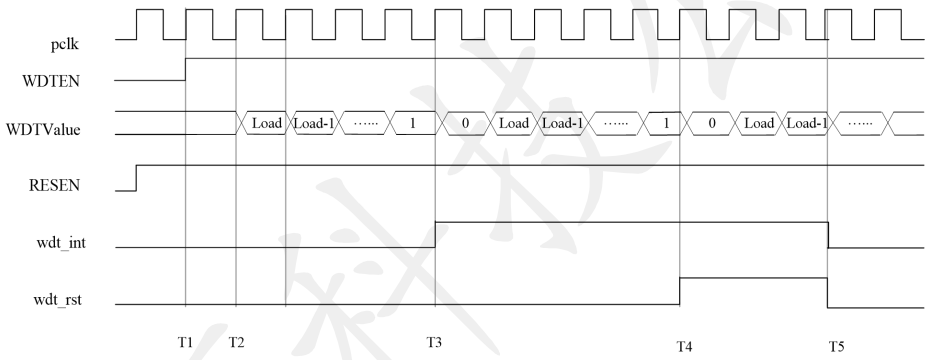
看门狗定时器（WDT）主要用于控制程序流程正确，在程序流长时间未按既定流程执行指定程序的情况下复位芯片。使用前需使能对应 WDT 模块时钟。

5.11.2 特性

- 产生计数器溢出复位信号，复位信号使能可配
- 具有 32 位计数位宽，可配置灵活、宽范围的溢出周期
- 具有中断功能
- 具有喂狗功能

5.11.3 功能描述

中断模式



1）软件首先设置计数初值和参数，之后启动 WDT 计数（T1 时刻， WDTEN 有效）。

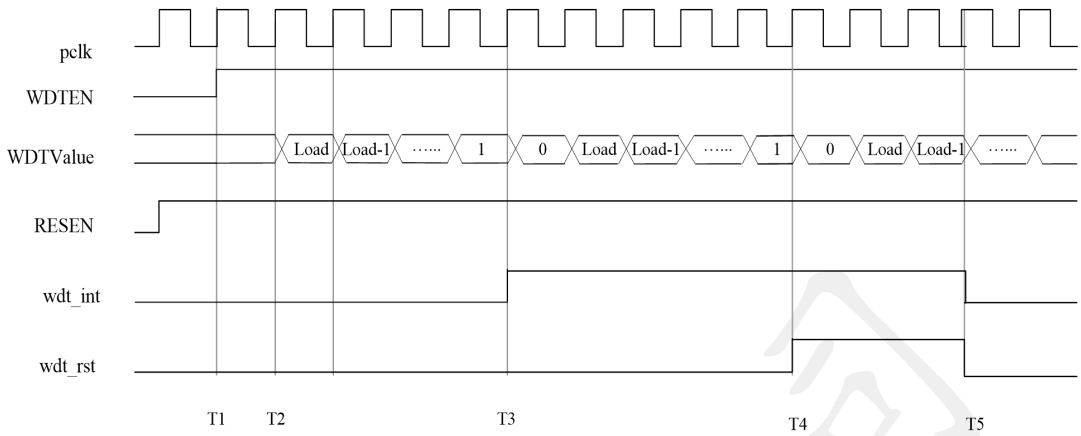
2）WDTEN 有效后，在 pclk 上升沿（T2 时刻），装载初值。开始递减计数，每一个 pclk 时钟沿减 1，计数值为 0(T4 时刻）后继续重置为 Load 后计数。

3）T4 时刻，当 WDTValue 递减为 0 时，WDTIS 寄存器的 Watchdog Interrupt 位被硬件置 1，向系统输出 wdt_int 中断信号。

4）CPU 接收到 wdt_int 中断后，进行中断处理，软件清除中断（T5 时刻）。

5）软件设置 WDTEN 为 0（T6 时刻），则停止计数。

复位模式



1）软件首先设置计数初值和参数，之后启动 WDT 计数（T1 时刻， WDTEN 有效）。

2）WDTEN 有效后，在 pclk 上升沿（T2 时刻），装载初值。开始递减计数，每一个 pclk 时钟沿减 1，计数值为 0(T3 时刻）后继续重置为 Load 后计数。

3）T3 时刻，当 WDTValue 递减为 0 时，WDTIS 寄存器的 Watchdog Interrupt 位被硬件置 1，向系统输出 wdt_int 中断信号。

4）wdt_int 中断信号产生（T3 时刻）后，如果一直没有得到响应，那么当 WDTValue 再次计数到 0（T4 时刻），如果 WDTControl 寄存器中 RSTEN 位为 1 时（图中所示），wdt_rst 信号有效，否则当 RSTEN 为 0 时，wdt_rst 信号无效。wdt_rst 有效时可用于产生系统复位。

5） 如果 wdt_rst 信号产生后，未发生系统复位，则通过清除 WDT 中断标志，可以清除 wdt_rst 信号，如图所示 T5 时刻 CPU 清中断，wdt_int 和 wdt_rst 信号同时无效。

5.11.4 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
WDT	BASE: 0x40041000				
WDTLOAD	0x00	32	R/W	0	WDT 初值寄存器
WDTVALUE	0x04	32	R	0	WDT 当前计数值寄存器
WDTCTRL	0x08	32	R/W	0	WDT 控制寄存器
WDTIF	0x0C	32	R/W	0	WDT 中断状态寄存器
WDTFEED	0x10	32	R/W	0	WDT 喂狗寄存器

5.11.5 寄存器描述

WDTLOAD 寄存器（0x00）

位域	名称	类型	复位值	描述
31:0	WDTLOAD	R/W	0	包含 WDT 计数器的初始值。 WDT 启动时，计数器自动装载 WDTLOAD 值，开始递减计数。当计数器值计到 0 时，将 WDTLOAD 寄存器中的值再次装载到计数器中，继续计数。再次计数到 0 时，产生中断。 WDT 启动后设置 WDTLOAD 寄存器无效。

WDTVALUE 寄存器（0x04）

位域	名称	类型	复位值	描述
31:0	WDTVALUE	R	0	该寄存器为只读寄存器，复位值为 0。读该寄存器时，返回计数器的当前计数值。

WDTCTRL 寄存器（0x08）

位域	名称	类型	复位值	描述
31:2	RESERVED	RO	0	保留位
1	RSTEN	R/W	0	WDT 复位输出使能位 1：使能复位 0：禁止复位
0	WDTEN	R/W	0	WDT 启动位 1：启动 WDT 计数 0：停止计数

WDTIF 寄存器（0x0C）

位域	名称	类型	复位值	描述
31:1	RESERVED	RO	0	保留位
0	WDT_IF	R/W	0	WDT 中断状态位，高有效 硬件置位，软件写 0 清除，写 1 无效

WDTFEED 寄存器（0x10）

位域	名称	类型	复位值	描述
31:8	RESERVED	RO	0	保留位
7:0	FEED	R/W	0	WDT 重启计数器寄存器 向该寄存器写入 0x55 后会重启 WDT 计数器（喂狗操作）

5.12 实时时钟（RTC）

5.12.1 概述

该 RTC 为极简实时时钟。使用 32bit 可配置计数器，计数时钟为 32K，由软件自己计算日历。具有秒中断、闹钟中断和溢出中断。使用前需使能 RTC 模块时钟。

5.12.2 特性

- 秒中断
- 闹钟中断
- 溢出中断

5.12.3 功能描述

使用流程

使用 RTC 前，需进行如下操作：

- 通过寄存器 EN 禁能 RTC。
- 配置 SEC 秒寄存器。
- 如果设置闹钟，配置 ALA 闹钟寄存器。
- 根据需要配置秒中断、闹钟中断、溢出中断。
- 通过寄存器 EN 使能 RTC。
- 通过 TIM 寄存器可以查看当前计数值。
- 如果配置了中断，可以通过 IF 寄存器查看相关中断状态。

5.12.4 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
RTC	BASE: 0x40042000				
RTC_EN	0x00	32	R/W	0	RTC 使能寄存器
RTC_ALA	0x04	32	R/W	0	RTC 闹钟设置寄存器
RTC_SEC	0x08	32	R/W	0	RTC 秒设置寄存器
RTC_TIM	0x0c	32	R	0	RTC 时间计数器寄存器
RTC_IE	0x10	32	R/W	0	RTC 中断使能寄存器
RTC_IF	0x14	32	R/W	0	RTC 中断状态寄存器

5.12.5 寄存器描述

RTC_EN 寄存器（0x00）

位域	名称	类型	复位值	描述
31:1	RESERVED	R	0	保留位

0	RTC_EN	R/W	0	RTC 使能寄存器 0：禁能 1：使能 将该位置为 1 后，内部时间计数器开始以 32K 时钟进行计数。在 RTC 使能之前，必须将各个配置寄存器都配置完毕。

RTC_ALA 寄存器（0x04）

位域	名称	类型	复位值	描述
31:0	RTC_ALARM	RW	0	RTC 闹钟配置寄存器 当时间计数器的计数值等于闹钟配置的值后，将产生闹钟信号，若闹钟中断打开，则产生闹钟中断。 配置的数值应该大于等于 2。

RTC_SEC 寄存器（0x08）

位域	名称	类型	复位值	描述
31:0	RTC_SEC	RW	0	RTC 秒配置寄存器 当时间计数器的计数值每间隔一个秒配置值，则将产生秒信号，若秒中断打开，则产生秒中断。 配置该寄存器可以产生一个周期性的秒信号。配置的数值应该大于等于 2。

RTC_TIM 寄存器（0x0C）

位域	名称	类型	复位值	描述
31:0	RTC_TIM	R	0	RTC 时间计数器计数值寄存器 通过该寄存器可以读取时间计数器的当前计数值。

RTC_IE 寄存器（0x10）

位域	名称	类型	复位值	描述

31:3	RESERVED	R	0	保留位
2	OVF_IE	RW	0	时间计数器溢出中断使能 0：禁能 1：使能
1	ALA_IE	RW	0	闹钟中断使能 0：禁能 1：使能
0	SEC_IE	RW	0	秒中断使能 0：禁能 1：使能

RTC_IF 寄存器（0x14）

位域	名称	类型	复位值	描述
31:3	RESERVED	R	0	保留位
2	OVF_IF	RW	0	时间计数器溢出状态 1：表示产生时间计数器溢出状态 写 1 清零
1	ALA_IF	RW	0	闹钟状态 1：表示产生闹钟状态 写 1 清零
0	SEC_IF	RW	0	秒状态 1：表示产生秒状态 写 1 清零

5.13 UART 控制器（UART）

5.13.1 概述

本芯片具有 3 路串口，支持波特率配置，支持多种中断，支持 break 功能，支持 LOOPBACK 功能，使用前需要使能对应的 UART 时钟。

5.13.2 特性

- 支持标准的 UART 协议
- 支持全双工模式
- 支持波特率配置
- 支持 5/6/7/8 位数据格式选择
- 可配置奇偶校验位 奇校验 偶校验 常 0 常 1
- 支持 1/2 位停止位选择
- 支持 break 功能
- 支持 LOOPBACK 功能
- 支持多种中断

5.13.3 功能描述

数据位

可以通过向 CTRL 寄存器中的 DLS 位写入相应的值，选择不同的数据位。

00：5bits 01：6bits 10：7bits 11：8bits。

奇偶校验位

可以通过向 CTRL 寄存器中的 PEN、EPS、Stick Parity 位写入相应的值，选择不同的奇偶校验方式。

xx0 无校验 001 奇校验 011 偶校验 101 常 1 111 常 0

停止位

可以通过向 CTRL 寄存器中的 STOP 位写入相应的值，选择不同的停止位。

写入 0 表示 1 位停止位，写入 1 表示 2 位停止位。

波特率

设置波特率时需先将 CTRL 寄存器中的 DLAB 位置为 1，再设置 DLH 与 DLL 寄存器。BAUD = PCLK/(16*(DLH*0x100 + DLL))。

数据发送接收

发送数据时向发送数据寄存器 THR 写入数据，数据将发送到 TX 线上。通过读取数据状态寄存器 LSR 的 THRE 位状态，获取当前发送状态。

接收数据时通过读取数据状态寄存器 LSR 的 DR 位状态，获取当前接收状态，判断是否接收到有效数据，读取接收数据寄存器 RBR，可以获得 RX 线上的数据。可以设置接收中断，这样可以在中断中进行数据的处理。

5.13.4 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
UART0： UART1： UART2：	BASE: 0x400A7000 BASE: 0x40044000 BASE: 0x40044800				
RBR	0x00	32	R	0	接收数据寄存器
THR	0x00	32	W	0	发送数据寄存器
DLH	0x04	32	R/W	0	波特率高 8 位寄存器
DLL	0x00	32	R/W	0	波特率低 8 位寄存器
IER	0x04	32	R/W	0	中断使能寄存器
CTRL	0x0c	32	R/W	0	数据控制寄存器
MCR	0x10	32	R/W	0	LOOPBACK 使能寄存器
LSR	0x14	32	R	0	数据状态寄存器

5.13.5 寄存器描述

RBR 寄存器（0x00）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	RBR	R	0	接收数据寄存器 读操作返回缓存中接收的数据

THR 寄存器（0x00）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	THR	W	0	发送数据寄存器 写操作写入待发送的数据

DLH 寄存器（0x04）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	DLH	R/W	0	波特率高 8 位配置寄存器 UART 波特率= pclk/(16*{DLH,DLL})。

DLL 寄存器（0x00）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留
7:0	DLL	R/W	0	波特率低 8 位配置寄存器 UART 波特率= pclk/(16*{DLH,DLL})。

IER 寄存器（0x04）

位域	名称	类型	复位值	描述
31:2	RESERVED	R	0	保留位
1	ETBEI	R/W	0	发送数据寄存器空中断使能
0	ERBFI	R/W	0	接收数据有效中断使能

CTRL 寄存器（0x0C）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7	DLAB	RW	0	波特率设置开启位 设置波特率时需先置此位为 1，再设置 DLH 与 DLL 寄存器
6	BREAK	RW	0	BREAK 控制使能 此位设为 1 时产生 break，强制把输出信号置低，直到此位被清除
5	Stick Parity	R/W	0	强制改变发送奇偶校验位数值使能 0：不强制改变 1：若 PEN、EPS 为 1，则强制奇偶校验位为 0 若 PEN 为 1、EPS 为 0，则强制奇偶校验位为 1
4	EPS	R/W	0	奇偶校验选择 0：奇校验 1：偶校验
3	PEN	R/W	0	奇偶校验使能 0：关闭校验 1：打开校验

2	STOP	R/W	0	停止位位宽选择 0：1 stop bit 1：2 stop bit
1:0	DLS	R/W	0	数据位宽选择 00：5bits 01：6bits 10：7bits 11：8bits

MCR 寄存器（0x10）

位域	名称	类型	复位值	描述
31:5	RESERVED	R	0	保留位
4	LOOPBACK	R/W	0	LOOPBACK 使能 0：LOOPBACK 关闭 1：LOOPBACK 打开，输入输出信号短接
3:0	RESERVED	R	0	保留位

LSR 寄存器（0x14）

位域	名称	类型	复位值	描述
31:7	RESERVED	R	0	保留位
6	TEMT	R	0	无发送数据状态 当 THR 以及发送移位寄存器都无数据时，此位置 1
5	THRE	R	0	发送数据寄存器空状态 当 THR 无待发送数据时，此位置 1

4	BI	R	0	检测输入 BREAK 状态 若接收数据端口上低电平持续了一整个数据周期时（起始位+数据位+奇偶校验位+停止位），此位置 1 0：未检测到 BREAK 1：检测到 BREAK
3	FE	R	0	STOP 校验状态 0：STOP 位校验正确 1：STOP 位校验错误
2	PE	R	0	奇偶校验状态 0：奇偶校验正确 1：奇偶校验错误
1	OE	R	0	数据溢出状态 若接收数据时，上一次的接收数据还未读走，但是又接收到了新的数据 0：没有溢出 1：发生溢出
0	DR	R	0	数据接收状态 RBR 中的数据是否已经传输完毕，可以被读走 0：未接收到数据 1：已接收到数据

5.14 SPI 总线控制器（SPI）

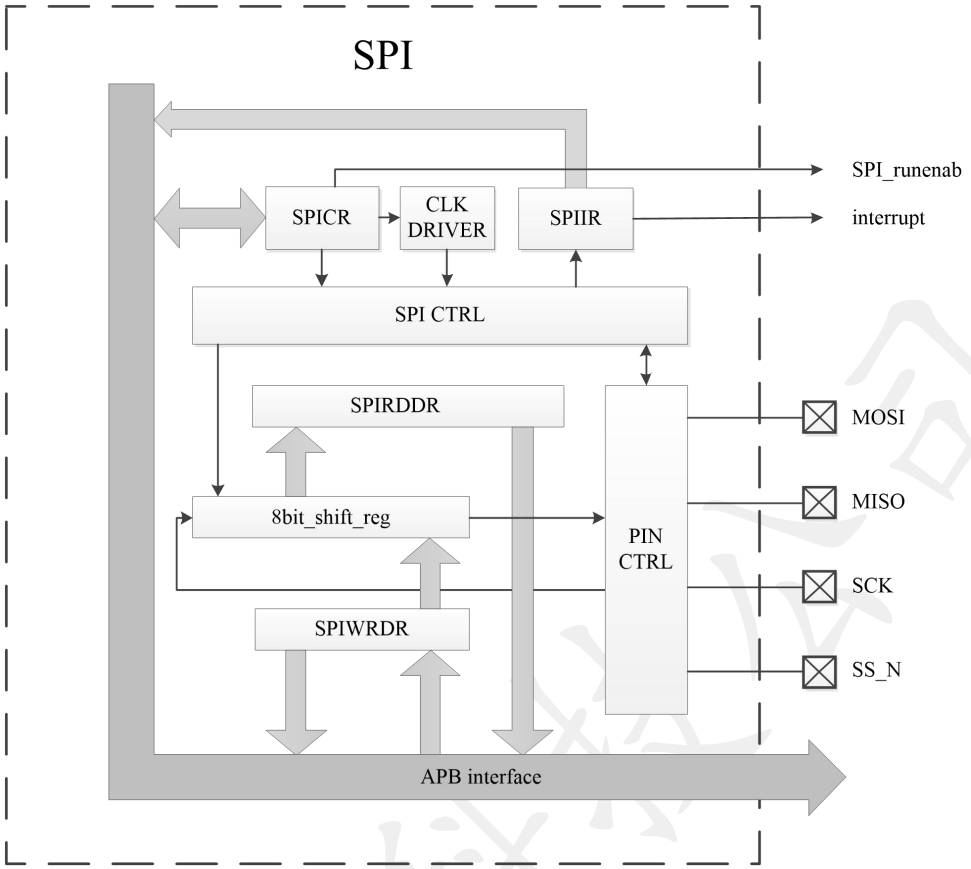
5.14.1 概述

SPI 是一种用于全双工模式的串行同步数据通讯协议。该模块为支持 SPI 通讯协议的接口控制模块，使用前需使能对应 SPI 模块时钟。

5.14.2 特性

- 支持主机模式和从机模式
- 可编程时钟极性和相位
- 主模式速率可配，最大频率为系统时钟 4 分频
- 数据传输顺序可配置
- 传输结束中断标志
- 读数据寄存器和写数据寄存器分开

5.14.3 模块结构框图



5.14.4 功能描述

IO 配置

- 主输出、从输入（MOSI）

主出从入（MOSI）引脚是主器件的输出和从器件的输入，用于主器件到从器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输出，当 SPI 配置为从器件时，该引脚为输入。

- 主输入、从输出（MISO）

主入从出（MISO）引脚是从器件的输出和主器件的输入，用于从器件到主器件的串行数据传输。当 SPI 配置为主器件时，该引脚为输入，当 SPI 配置为从器件时，该引脚为输出。

- 串行时钟（SCK）

串行时钟（SCK）引脚是主器件的输出和从器件的输入，用于同步主器件和从器件之间

在 MOSI 和 MISO 线上的串行数据传输。当 SPI 配置为主器件时，该引脚输出时钟，当 SPI 配置为从器件时，该引脚为输入。

- 从选择（SSN）

从选择（SSN）引脚用来控制从器件选中，当 SPI 配置为主器件时，SSN 引脚可以通过寄存器的方式控制从器件的选中与否，当 SPI 配置为从器件时，SSN 引脚取决于主器件的控制信号。

主从器件的 MOSI、MISO 和 SCK 分别连在一起。主从器件通过 MOSI、MISO 连成一个环路，主器件输出时钟，数据传输时，主器件通过 MOSI 输出数据，从器件通过 MISO 输出数据。一个字节数据传输完毕，主从器件将交换 8 位移位寄存器数值。

数据传输配置

- 配置 SPICR.CPHA 位和 SPICR.CPOL 位，以设置串行时钟相位和极性（主从器件需一致）。
- 配置 SPICR.SPR[2:0]位，以设置串行时钟波特率（若为从器件模式则不用设置，串行时钟速率由主器件决定）。
- 配置 SPICR.LSB 位，设置传输顺序，配置 SPICR.CPHA_DATAHOLD_S,，来设置从模式下传输数据保持周期数目。
- 需要时，配置中断，配置 SPIIE 和 SPIIF 位。
- 进行数据传输前需先配置 SPICR.SPE 位和 SPICR.MSTR 位，以使能 SPI 和设置主从模式。
- 主器件模式下数据传输前需先将从器件的 SSN 引脚拉低，。主器件模式下 MCU 写 SPIWDR 寄存器的动作启动数据传输，中断标志 SPIIF.SPIF 置起完成数据传输。

5.14.5 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
SPI: BASE: 0x40043000					
SPICR	0x00	32	R/W	0	SPI 控制寄存器
SPIWDR	0x04	32	R/W	0	SPI 写数据寄存器

SPIRDR	0x08	32	R	0	SPI 读数据寄存器
SPIIE	0x10	32	RW	0	SPI 中断使能寄存器
SPIIF	0x14	32	R/W	0	SPI 中断状态寄存器

5.14.6 寄存器描述

SPICR 寄存器（0x00）

位域	名称	类型	复位值	描述
31:13	RESERVED	R	0	保留位
12	MSR_SSN	R/W	1	主模式下 SSN 输出，默认情况下输出 1 该寄存器仅在主模式下有效
11:8	CPHA_DATAHOLD_S	R/W	0	从模式下 CPHA 为 1 时，数据保持时间配置寄存器 0000：1 个 pclk 0001：2 个 pclk 。。。 1111：16 个 pclk
7	LSB	R/W	0	数据传输顺序选择 0：MSB 1：LSB
6	MSTR	R/W	0	主从模式选择 0 = SPI 系统配置为从器件模式 1 = SPI 系统配置为主器件模式
5	CPOL	R/W	0	时钟极性选择 0 = 串行时钟空闲状态为低电平，有效电平为高电平 1 = 串行时钟空闲状态为高电平，有效电平为低电平

4	CPHA	R/W	0	时钟相位选择 0 = 在串行时钟的第一个跳变沿采样数据 1 = 在串行时钟的第二个跳变沿采样数据
3	SPE	R/W	0	SPI 系统使能 0 = SPI 系统关闭 1 = SPI 系统使能
2	SPR2	R/W	0	SPI 波特率选择位 2
1	SPR1	R/W	0	SPI 波特率选择位 1
0	SPR0	R/W	0	SPI 波特率选择位 0

SPR0，SPR1，SPR2 表示波特率的选择：

SPR2	SPR1	SPR0	Fsck	Fsck(Fcpu=48Mhz)
0	0	0	Fpclk/4	12MHz
0	0	1	Fpclk/8	6MHz
0	1	0	Fpclk/16	3MHz
0	1	1	Fpclk/32	1.5MHz
1	0	0	Fpclk/64	750KHz
1	0	1	Fpclk/128	375KHz
1	1	0	Fpclk/256	187.5KHz
1	1	1	Fpclk/512	93.75KHz

SPIWDR 寄存器（0x04）

位域	名称	类型	复位值	描述
7:0	SPIWDR	R/W	0	数据的写通过 SPIWDR 的操作完成。准备发送数据时，移位寄存器读取 SPIWDR 的值，进行数据的发送。 注：SPIWDR 能够在传输过程中多次写入新数据，下一次传输开始前最后写入 SPIWDR 的数据为下一次传输的数据。

SPIRDR 寄存器（0x08）

位域	名称	类型	复位值	描述
7:0	SPIRDR	R	0	数据的读通过 SPIRDR 的操作完成。接收数据时，移位寄存器在每次传输完成后，将数据写入 SPIRDR 中。

SPIIE 寄存器（0x10）

位域	名称	类型	复位值	描述
31:2	RESERVED	R	0	保留位
1	SPIF_IE	R/W	0	SPI 传输结束标志中断使能
0	RESERVED	R	0	保留位

SPIIF 寄存器（0x14）

位域	名称	类型	复位值	描述
31:2	RESERVED	R	0	保留位
1	SPIF	R/W	0	SPI 传输结束标志 该标志在一次 SPI 数据传输结束时硬件置位。 写 1 清零。 注：传输结束标志到来后，没有及时读走数据，则下次传输的数据会覆盖前一次数据。
0	RESERVED	R	0	保留位

5.15 脉冲宽度调制发生器（PWM）

5.15.1 概述

PWM 模块提供 5 路（PWM0，PWM1，PWM2，PWM3，PWM4）独立通道，支持预分频功能，支持输出电平翻转，支持高电平结束中断和周期结束中断。使用 PWM 模块之前需要使能 PWM 时钟。

5.15.2 特性

- 5 路 12 位宽 PWM 控制
- 预分频功能，可以选择 4、8、16、32、64、128、256、512 分频
- 输出电平翻转
- 支持高电平结束中断和周期结束中断

5.15.3 功能描述

使用流程

- PWM 时钟使能
- PORT 端口配置为 PWM 功能
- 配置预分频（PWM_DIV）寄存器
- 配置输出翻转（PWM_CON）寄存器
- 配置中断使能（INT_EN）寄存器
- 配置周期（PWMx_PERIOD）和高电平（PWMx_HIGH）寄存器
- 如果配置了中断，使能 PWM 中断
- 配置 PWM 使能位（PWM_EN），开启 PWM

5.15.4 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
PWM	BASE: 0x400AA000				
PWM_EN	0x00	32	R/W	0	PWM 使能寄存器
PWM_DIV	0x04	32	R/W	0	PWM 计数时钟分频寄存器
PWM_CON	0x08	32	R/W	0	PWM 配置寄存器
INT_EN	0x10	32	R/W	0	PWM 中断使能寄存器
INT_ST	0x14	32	R/W	0	PWM 中断状态寄存器
PWM0_PERIOD	0x100	32	R/W	0	PWM0 周期配置寄存器
PWM0_HIGH	0x104	32	R/W	0	PWM0 高电平配置寄存器
PWM1_PERIOD	0x200	32	R/W	0	PWM1 周期配置寄存器
PWM1_HIGH	0x204	32	R/W	0	PWM1 高电平配置寄存器
PWM2_PERIOD	0x300	32	R/W	0	PWM2 周期配置寄存器
PWM2_HIGH	0x304	32	R/W	0	PWM2 高电平配置寄存器
PWM3_PERIOD	0x400	32	R/W	0	PWM3 周期配置寄存器
PWM3_HIGH	0x404	32	R/W	0	PWM3 高电平配置寄存器
PWM4_PERIOD	0x500	32	R/W	0	PWM4 周期配置寄存器
PWM4_HIGH	0x504	32	R/W	0	PWM4 高电平配置寄存器

5.15.5 寄存器描述

PWM_EN 寄存器（0x00）

位域	名称	类型	复位值	描述
31:5	RESERVED	R	0	保留位

4	PWM4_EN	R/W	0	PWM4 使能寄存器 0：禁能 1：使能 将该位置为 1 后，PWM4 计数器开始以分频时钟进行计数。在使能之前，必须将各个配置寄存器都配置完毕。
3	PWM3_EN	R/W	0	PWM3 使能寄存器 0：禁能 1：使能 将该位置为 1 后，PWM3 计数器开始以分频时钟进行计数。在使能之前，必须将各个配置寄存器都配置完毕。
2	PWM2_EN	R/W	0	PWM2 使能寄存器 0：禁能 1：使能 将该位置为 1 后，PWM2 计数器开始以分频时钟进行计数。在使能之前，必须将各个配置寄存器都配置完毕。
1	PWM1_EN	R/W	0	PWM1 使能寄存器 0：禁能 1：使能 将该位置为 1 后，PWM1 计数器开始以分频时钟进行计数。在使能之前，必须将各个配置寄存器都配置完毕。
0	PWM0_EN	R/W	0	PWM0 使能寄存器 0：禁能 1：使能 将该位置为 1 后，PWM0 计数器开始以分频时钟进行计数。在使能之前，必须将各个配置寄存器都配置完毕。

PWM_DIV 寄存器（0x04）

位域	名称	类型	复位值	描述
31:3	RESERVED	R	0	保留位

2：0	PWM_DIV	R/W	0	PWM 计数时钟分频寄存器 0x0：系统时钟 4 分频 0x1：系统时钟 8 分频 0x2：系统时钟 16 分频 0x3：系统时钟 32 分频 0x4：系统时钟 64 分频 0x5：系统时钟 128 分频 0x6：系统时钟 256 分频 0x7：系统时钟 512 分频

PWM_CON 寄存器（0x08）

位域	名称	类型	复位值	描述
31:5	RESERVED	R	0	保留位
4	PWM4_OUT_INV	R/W	0	PWM4 输出是否翻转寄存器 0：禁能 1：使能 将该位置为 1 后，PWM4 输出波形为原波形翻转
3	PWM3_OUT_INV	R/W	0	PWM3 输出是否翻转寄存器 0：禁能 1：使能 将该位置为 1 后，PWM3 输出波形为原波形翻转
2	PWM2_OUT_INV	R/W	0	PWM2 输出是否翻转寄存器 0：禁能 1：使能 将该位置为 1 后，PWM2 输出波形为原波形翻转
1	PWM1_OUT_INV	R/W	0	PWM1 输出是否翻转寄存器 0：禁能 1：使能 将该位置为 1 后，PWM1 输出波形为原波形翻转
0	PWM0_OUT_INV	R/W	0	PWM0 输出是否翻转寄存器 0：禁能 1：使能 将该位置为 1 后，PWM0 输出波形为原波形翻转

PWM_INTEN 寄存器（0x10）

位域	名称	类型	复位值	描述
31:18	RESERVED	R	0	保留位
17	PWM4_P_INTEN	R/W	0	PWM4 周期溢出中断使能 0：禁能 1：使能
16	PWM4_HP_INTEN	R/W	0	PWM4 高电平溢出中断使能 0：禁能 1：使能
15:14	RESERVED	R	0	保留位
13	PWM3_P_INTEN	R/W	0	PWM3 周期溢出中断使能 0：禁能 1：使能
12	PWM3_HP_INTEN	R/W	0	PWM3 高电平溢出中断使能 0：禁能 1：使能
11:10	RESERVED	R	0	保留位
9	PWM2_P_INTEN	R/W	0	PWM2 周期溢出中断使能 0：禁能 1：使能
8	PWM2_HP_INTEN	R/W	0	PWM2 高电平溢出中断使能 0：禁能 1：使能
7:6	RESERVED	R	0	保留位
5	PWM1_P_INTEN	R/W	0	PWM1 周期溢出中断使能 0：禁能 1：使能
4	PWM1_HP_INTEN	R/W	0	PWM1 高电平溢出中断使能 0：禁能 1：使能
3:2	RESERVED	R	0	保留位
1	PWM0_P_INTEN	R/W	0	PWM0 周期溢出中断使能 0：禁能 1：使能
0	PWM0_HP_INTEN	R/W	0	PWM0 高电平溢出中断使能 0：禁能 1：使能

PWM_INTST 寄存器（0x14）

位域	名称	类型	复位值	描述
31:18	RESERVED	R	0	保留位
17	PWM4_P_ST	R/W	0	PWM4 周期溢出状态 写 1 清零
16	PWM4_HP_ST	R/W	0	PWM4 高电平溢出状态 写 1 清零
15:14	RESERVED	R	0	保留位
13	PWM3_P_ST	R/W	0	PWM3 周期溢出状态 写 1 清零
12	PWM3_HP_ST	R/W	0	PWM3 高电平溢出状态 写 1 清零
11:10	RESERVED	R	0	保留位
9	PWM2_P_ST	R/W	0	PWM2 周期溢出状态 写 1 清零
8	PWM2_HP_ST	R/W	0	PWM2 高电平溢出状态 写 1 清零
7:6	RESERVED	R	0	保留位
5	PWM1_P_ST	R/W	0	PWM1 周期溢出状态 写 1 清零
4	PWM1_HP_ST	R/W	0	PWM1 高电平溢出状态 写 1 清零
3:2	RESERVED	R	0	保留位
1	PWM0_P_ST	R/W	0	PWM0 周期溢出状态 写 1 清零
0	PWM0_HP_ST	R/W	0	PWM0 高电平溢出状态 写 1 清零

PWMx_PERIOD 寄存器（0x100*（x+1））

位域	名称	类型	复位值	描述
31:12	RESERVED	R	0	保留位
11：0	PWMx_PERIOD	R/W	0xFFF	PWMx 输出周期配置寄存器 注 1：该寄存器不能配置为 0

PWMx_HIGH 寄存器（0x100*（x+1） + 0x04）

位域	名称	类型	复位值	描述
31:12	RESERVED	R	0	保留位
11：0	PWMx_HIGH	R/W	0	PWMx 输出高电平配置寄存器 注 1：若该寄存器配置为 0，则输出 0； 注 2：该寄存器配置值必须小于周期值。

5.16 模数转换器（ADC）

5.16.1 概述

该模块为 SARADC 模拟电路的数字控制单元，通过该模块可实现 CPU 对 ADC 进行采样控制并完成转换数据获取。

5.16.2 特性

- 9 通道 12bit SARADC
- 采样率最高可达 2.4M
- 支持单次模式和连续模式
- 具备深度为 9 的 FIFO
- 支持硬件求平均功能，可以配置为 1、2、4、8 次求平均
- 采样建立时间可配置，可以配置为 1、2、4、8、16、32、64、128 个系统时钟周期以及

1、3、5、7、9、11、13、15 个系统时钟周期

- 具有数据转换完成中断、FIFO 半满中断、FIFO 满中断、FIFO 溢出中断
- ADC 参考电压可以选择内部参考或者外部参考　内部参考电压 1.4V
- ADC 可以采集 VDD 电压，选择内部参考，1/3 分压后进入通道 8 采集，采集的数据为 data,电压计算公式为：VDD/(3*1.4) = data/4096。
- 支持输出数据校准

5.16.3 功能描述

操作说明

使用 ADC 之前需要针对对应的引脚和模块进行如下操作：

- 配置引脚为 ADC 功能
- 配置 ADC 的转换时钟分频值
- 配置 ADC 的转换通道
- 配置采样取平均
- 配置转换模式为单次模式还是连续模式
- 配置数据存储为 FIFO 还是通道
- 配置采样时钟为内部采样还是外部采样
- 配置内部或者外部采样窗口
- 配置参考源选择为内部参考还是外部参考
- 配置 VDD 检测使能是否常开
- 配置 KD 数据是否有效
- 配置 offset 数据是否有效
- 配置通道转换完成中断是否使能
- 配置 FIFO 满中断是否使能
- 配置 FIFO 半满中断是否使能
- 配置 FIFO 溢出中断是否使能
- ADC 开启

- ADC 软复位
- ADC 启动转换，等待转换完成，读取转换数据

5.16.4 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
ADC_CFG	0x00	32	R/W	1	ADC 配置寄存器
ADC_START	0x04	32	R/W	0	ADC 启动寄存器
ADC_IE	0x08	32	R/W	0	ADC 中断使能寄存器
ADC_IF	0x0c	32	RW	0	ADC 中断状态寄存器
ADC_CH0_STAT	0x10	32	R/W	0	ADC 通道 0 状态寄存器
ADC_CH0_DATA	0x14	32	R/W	0	ADC 通道 0 数据寄存器
ADC_CH1_STAT	0x20	32	R/W	0	ADC 通道 1 状态寄存器
ADC_CH1_DATA	0x24	32	R/W	0	ADC 通道 1 数据寄存器
ADC_CH2_STAT	0x30	32	R/W	0	ADC 通道 2 状态寄存器
ADC_CH2_DATA	0x34	32	R/W	0	ADC 通道 2 数据寄存器
ADC_CH3_STAT	0x40	32	R/W	0	ADC 通道 3 状态寄存器
ADC_CH3_DATA	0x44	32	R/W	0	ADC 通道 3 数据寄存器
ADC_CH4_STAT	0x50	32	R/W	0	ADC 通道 4 状态寄存器
ADC_CH4_DATA	0x54	32	R/W	0	ADC 通道 4 数据寄存器
ADC_CH5_STAT	0x60	32	R/W	0	ADC 通道 5 状态寄存器
ADC_CH5_DATA	0x64	32	R/W	0	ADC 通道 5 数据寄存器

ADC_CH6_STAT	0x70	32	R/W	0	ADC 通道 6 状态寄存器
ADC_CH6_DATA	0x74	32	R/W	0	ADC 通道 6 数据寄存器
ADC_CH7_STAT	0x80	32	R/W	0	ADC 通道 7 状态寄存器
ADC_CH7_DATA	0x84	32	R/W	0	ADC 通道 7 数据寄存器
ADC_CH8_STAT	0x90	32	R/W	0	ADC 通道 8 状态寄存器
ADC_CH8_DATA	0x94	32	R/W	0	ADC 通道 8 数据寄存器
ADC_FIFO_STAT	0xa0	32	R/W	0	ADC FIFO 状态寄存器
ADC_FIFO_DATA	0xa4	32	R/W	0	ADC FIFO 数据寄存器
ADC_CTRL	0xe0	32	R/W	0	ADC 控制寄存器
ADC_CALIB_OFFSET	0xf0	32	R/W	0	ADC 校准 OFFSET 寄存器
ADC_CALIB_KD	0xf4	32	R/W	0	ADC 校准 KD 寄存器

5.16.5 寄存器描述

ADC_CFG 寄存器（0x00）

位域	名称	类型	复位值	描述
30：22	RESERVED	R	0	保留位
21	EN_AVDDSNS	R/W	0	ADC VDD 检测使能常开控制位 1：ADC VDD 检测使能常开 0：只有通道 8 有效时，ADC VDD 检测使能才打开

20	ADC_EN	R/W	0	ADC 使能位 0：禁能 1：使能
19:17	IN_SMPL_WIN	R/W	010	ADC 内部采样时钟方式采样窗口设置 000: 采样建立时间保持 1 个 ADC 时钟周期 001: 采样建立时间保持 3 个 ADC 时钟周期 010: 采样建立时间保持 5 个 ADC 时钟周期 011: 采样建立时间保持 7 个 ADC 时钟周期 100: 采样建立时间保持 9 个 ADC 时钟周期 101: 采样建立时间保持 11 个 ADC 时钟周期 110: 采样建立时间保持 13 个 ADC 时钟周期 111: 采样建立时间保持 15 个 ADC 时钟周期 **注：选择内部时钟采样时，ADC 时钟只能选择系统时钟 1 或 2 分频**
16	ADC_SMPL_CLK	R/W	0	ADC 采样模式选择 1：表示 ADC 采用内部采样时钟方式 0：表示 ADC 采用外部采样时钟方式
15	ADC_MEM_MODE	R/W	0	ADC 数据存储方式选择： 0：ADC 数据存储为 FIFO 模式； 1：ADC 数据存储为通道模式；
14:12	SMPL_SETUP	R/W	010	ADC 外部采样时钟方式采样窗口选择 000：采样建立时间保持 1 个系统时钟周期 001：采样建立时间保持 2 个系统时钟周期 010：采样建立时间保持 4 个系统时钟周期 011：采样建立时间保持 8 个系统时钟周期 100：采样建立时间保持 16 个系统时钟周期 101：采样建立时间保持 32 个系统时钟周期 110：采样建立时间保持 64 个系统时钟周期 111：采样建立时间保持 128 个系统时钟周期

11	CONT	R/W	0	ADC 采样工作模式 0：单次采样 1：连续采样
10:9	AVG	R/W	0	一次启动 ADC 采样取平均次数配置寄存器 00：1 次采样取平均 01：2 次采样取平均 10：4 次采样取平均 11：8 次采样取平均
8：0	ADC_CH _SEL	R/W	0	ADC 通道选择寄存器 Bit8~bit0 分别对应通道 8~通道 0。 对应位配置为 1 则表示该通道有效。

ADC_START 寄存器（0x04）

位域	名称	类型	复位值	描述
30：4	RESERVED	R	0	保留位
3	FIFO_CLR	R/W	0	FIFO 清除使能 0：禁能 1：使能 写 1 清 FIFO
2	ADC_SOFT_RESET	R/W	0	ADC 软复位 0 复位有效 在 ADC 使能之后，启动之前必须要复位一次 ADC
1	ADC_BUSY	R	0	ADC 工作状态 1 表示忙

0	ADC_START	R/W	0	ADC 启动信号 0：禁能 1：使能 该位写 1，则启动一次转换。可以 ADC_ CONT 配合使用 若 ADC_ CONT 处于单次采样模式，则该位置 1 后，将对有效通道依次轮询进行采样转换，并将转换的数据保存在相应通道的 FIFO 或寄存器中。转换完成后硬件会自动清零。 若 ADC_ CONT 处于连续采样模式，则该位置 1 表示启动 ADC 转换，清零后表示停止 ADC 转换。启动 ADC 转换后，将对有效通道依次轮询进行采样转换，并将转换的数据保存在相应通道的 FIFO 或寄存器中。每次转换完成后判断该位是否为 1，若为 1 则继续转换，若为 0 则停止转换。

ADC_IE 寄存器（0x08）

位域	名称	类型	复位值	描述
30：12	RESERVED	R	0	保留位
11	ADC_FIFO_OVF_IE	R/W	0	ADC FIFO 溢出中断使能 0：禁能 1：使能
10	ADC_FIFO_HFULL_IE	R/W	0	ADC FIFO 半满中断使能 0：禁能 1：使能
9	ADC_FIFO_FULL_IE	R/W	0	ADC FIFO 满中断使能 0：禁能 1：使能
8	ADC_CH8_EOC_IE	R/W	0	ADC 通道 8 数据转换完成中断使能 0：禁能 1：使能
7	ADC_CH7_EOC_IE	R/W	0	ADC 通道 7 数据转换完成中断使能 0：禁能 1：使能
6	ADC_CH6_EOC_IE	R/W	0	ADC 通道 6 数据转换完成中断使能 0：禁能 1：使能

5	ADC_CH5_EOC_IE	R/W	0	ADC 通道 5 数据转换完成中断使能 0：禁能 1：使能
4	ADC_CH4_EOC_IE	R/W	0	ADC 通道 4 数据转换完成中断使能 0：禁能 1：使能
3	ADC_CH3_EOC_IE	R/W	0	ADC 通道 3 数据转换完成中断使能 0：禁能 1：使能
2	ADC_CH2_EOC_IE	R/W	0	ADC 通道 2 数据转换完成中断使能 0：禁能 1：使能
1	ADC_CH1_EOC_IE	R/W	0	ADC 通道 1 数据转换完成中断使能 0：禁能 1：使能
0	ADC_CH0_EOC_IE	R/W	0	ADC 通道 0 数据转换完成中断使能 0：禁能 1：使能

ADC_IF 寄存器（0x0C）

位域	名称	类型	复位值	描述
30：12	RESERVED	R	0	保留位
11	ADC_FIFO_OF_IF	R/W	0	ADC FIFO 溢出中断状态 往 bit10 位写 1 清零
10	ADC_FIFO_HFULL_IF	R/W	0	ADC FIFO 半满中断状态 往 bit9 位写 1 清零
9	ADC_FIFO_FULL_IF	R/W	0	ADC FIFO 满中断状态 往 bit8 位写 1 清零
8	ADC_CH8_EOC_IF	R/W	0	ADC 通道 8 数据转换完成中断状态 写 1 清零
7	ADC_CH7_EOC_IF	R/W	0	ADC 通道 7 数据转换完成中断状态 写 1 清零
6	ADC_CH6_EOC_IF	R/W	0	ADC 通道 6 数据转换完成中断状态 写 1 清零

5	ADC_CH5_EOC_IF	R/W	0	ADC 通道 5 数据转换完成中断状态 写 1 清零
4	ADC_CH4_EOC_IF	R/W	0	ADC 通道 4 数据转换完成中断状态 写 1 清零
3	ADC_CH3_EOC_IF	R/W	0	ADC 通道 3 数据转换完成中断状态 写 1 清零
2	ADC_CH2_EOC_IF	R/W	0	ADC 通道 2 数据转换完成中断状态 写 1 清零
1	ADC_CH1_EOC_IF	R/W	0	ADC 通道 1 数据转换完成中断状态 写 1 清零
0	ADC_CH0_EOC_IF	R/W	0	ADC 通道 0 数据转换完成中断状态 写 1 清零

ADC_CHx_STAT 寄存器（0x10*（x+1））

位域	名称	类型	复位值	描述
30：2	RESERVED	R	0	保留位
1	ADC_CH_OV	R	0	ADC 通道 x 数据寄存器溢出标志 读数据寄存器清除
0	ADC_CH_EOC	R	0	ADC 通道 x 数据转换完成标志 1：表示 ADC 对通道 x 一次采样转换完成 通过向 ADC_IF 对应位写 1 可清零

ADC_CHx_DATA 寄存器（0x10*（x+1） + 4）

位域	名称	类型	复位值	描述

30：16	RESERVED	R	0	保留位
15:12	ADC_CH_NUM	R	0	ADC 数据对应的通道编号
11:0	ADC_CH_DATA	R	0	ADC 通道 x 数据寄存器 注：溢出后，再次转换的数据会覆盖旧数据

ADC_FIFO_STAT 寄存器（0xA0）

位域	名称	类型	复位值	描述
30：8	RESERVED	R	0	保留位
7:4	ADC_FIFO_LEVEL	R	0	ADC 数据 FIFO 水位
3	ADC_CH_OVF	R	0	ADC 数据 FIFO 溢出标志 1：表示 FIFO 溢出
2	ADC_FIFO_EMPTY	R	1	ADC 数据 FIFO 空标志 1：表示 FIFO 空 0：表示 FIFO 非空
1	ADC_FIFO_HFULL	R	0	ADC 数据 FIFO 半满标志 1：表示 FIFO 半满 0：表示 FIFO 非半满
0	ADC_FIFO_FULL	R	0	ADC 数据 FIFO 满标志 1：表示 FIFO 满 0：表示 FIFO 非满

ADC_FIFO_DATA 寄存器（0xA4）

位域	名称	类型	复位值	描述
30：16	RESERVED	R	0	保留位

15:12	ADC_FIFO_NUM	R	0	ADC 数据对应的通道编号
11:0	ADC_FIFO_DATA	R	0	ADC 数据 FIFO 寄存器 注：溢出后，再次转换的数据会被丢掉

ADC_CTRL 寄存器（0xE0）

位域	名称	类型	复位值	描述
30：1	RESERVED	R	0	保留位
0	IN_VREFP	R/W	1	ADC 内部 vrefp 选择使能 1：选择内部 vrefp 1.4V 0：选择外部 verfp

ADC_CALIB_OFFSET 寄存器（0xF0）

位域	名称	类型	复位值	描述
30：17	RESERVED	R	0	保留位
16	OFFSET_VALID	R/W	0	OFFSET 数据是否有效
15：8	RESERVED	R	0	保留位
7:0	OFFSET	R/W		ADC 数据校准的 OFFSET 值。计算出的 OFFSET 需要存入该寄存器。用于在使用 ADC 时进行校准。

ADC_CALIB_KD 寄存器（0xF4）

位域	名称	类型	复位值	描述
30：17	RESERVED	R	0	保留位

16	KD_VALID	R/W	0	KD 数据是否有效
15：10	RESERVED	R	0	保留位
9:0	KD	R/W	0	ADC 数据校准 K 值的小数部分。计算出的 K 值的小数部分需要存入该寄存器。用于在使用 ADC 时进行校准。若 K 大于 1，则符号位为 1，若 K 小于 1，则符号位为 0。

5.17 AES128 加密模块

5.17.1 概述

AES 模块可用于使用 AES 算法对数据进行加密和解密。

使用 128 位密钥长度对 128 位块进行加密和解密。 它还可以执行密钥派生、 加密或解密密钥存储在内部寄存器中，以便在使用同一密钥处理多个数据块时最大程度地减少 CPU 的写操作。

默认情况下，选择电子密码本模式（ECB）。 硬件还支持密码块链接（CBC）或计数器（CTR 模式）链接算法。

5.17.2 特性

- 使用 AES Rijndael 块密码算法进行加密/解密
- 内部 128 位寄存器，用于存储加密或派生密钥（4 个 32 位寄存器）
- 支持电子密码本（ECB），密码块链接（CBC）和计数器模式（CTR）
- 解密的派生密钥推导
- 128 位数据块处理
- 128 位密钥长度
- 213 个时钟周期来加密或解密一个 128 位块（包括输入和输出阶段）
- 1 个 32 位 INPUT 数据 buffer 和 1 个 32 位 OUTPUT 数据 buffer
- 仅支持 32 位数据宽度的寄存器访问
- 一个 128 位寄存器，用于在 CBC 模式下配置 AES 时用于初始化向量，或在选择 CTR

模式时用于 32 位计数器初始化

5.17.3 功能描述

加密

加密是指 AES 利用密钥寄存器（KEY）和初始向量寄存器（IV）在不同模式下对原数据进行加密处理，从而得到一系列密文的过程。加密流程如下：

1. 设置 AES_CR・EN=0 禁用 AES；
2. 设置 AES_CR・MODE[1:0]=00 配置为加密模式，设置 AES_CR・CHMOD[1:0]配置链接模式；
3. ECB 模式时配置 AES_KEYRx 寄存器，CTR 或 CBC 模式时还要配置 AES_IVRx 寄存器；
4. 设置 AES_CR・EN=1 开启 AES；
5. 把待加密的明文数据分四次连续写入 AES_DINR 寄存器（先写入 MSB）；
6. 等待 AES_SR 寄存器中 CCF 标志产生；
7. 连续读取 AES_DOUTR 寄存器四次来获取加密后的密文数据（先读出 MSB）；
8. 重复 5、6、7 完成所有数据的加密。

注：加密时使用的 KEY 为数据加密的初始密钥；

加密时 KEY 寄存器不改变；

CTR 模式加密时 IV 会改变；

密钥派生

密钥派生是指 AES 通过利用密钥寄存器（KEY）在不同模式下进行密钥派生处理，从而得到一系列派生密钥的过程。派生密钥流程如下：

1. 设置 AES_CR・EN=0 禁用 AES；
2. 设置 AES_CR・MODE[1:0]=01 配置为密钥派生模式，派生密钥时无需设置 AES_CR・CHMOD[1:0]，因为密钥派生模式时与所选的链接算法无关；
3. 配置 AES_KEYRx 寄存器，配置 AES_IVRx 寄存器无效；
4. 设置 AES_CR・EN=1 开启 AES；

5. 等待 AES_SR 寄存器中 CCF 标志产生；
6. 此时派生的密钥会储存在 AES_KEYRx 寄存器中，如果需要请读取 AES_KEYRx 寄存器来获得派生密钥；
7. 如果需要重新开始密钥派生操作，请重复步骤 3、4、5、6。

注：密钥派生与链接算法无关，只和原始密钥 KEY 有关。

解密

解密是指 AES 通过利用密钥寄存器（KEY）和初始向量寄存器（IV）在不同模式下对输入的密文数据进行解密处理，从而得到一系列明文数据的过程。解密流程如下：

1. 设置 AES_CR·EN=0 禁用 AES；
2. 设置 AES_CR·MODE[1:0]=10 配置为解密模式，设置 AES_CR·CHMOD[1:0]配置链接模式；
3. ECB 模式时配置 AES_KEYRx 寄存器，CTR 或 CBC 模式时还要配置 AES_IVRx 寄存器；
4. 设置 AES_CR·EN=1 开启 AES；
5. 把待解密的密文数据分四次连续写入 AES_DINR 寄存器（先写入 MSB）；
6. 等待 AES_SR 寄存器中 CCF 标志产生；
7. 连续读取 AES_DOUTR 寄存器四次来获取解密后的明文数据（先读出 MSB）；
8. 重复 5、6、7 完成所有数据的解密。

注：解密模式下使用的 KEY 为派生密钥，派生密钥可以通过密钥派生模式生成，然后再用于解密模式；

CTR 模式下的 KEY 为初始 KEY，IV 为初始 IV。

密钥派生+解密

密钥派生+解密是指 AES 先后进行密钥派生和解密过程，因此这里使用的密钥寄存器（KEY）为初始密钥。流程如下：

1. 设置 AES_CR·EN=0 禁用 AES；
2. 设置 AES_CR·MODE[1:0]=11 配置为密钥派生+解密模式，设置 AES_CR·CHMOD[1:0]配置链接模式（CTR 模式不支持密钥派生+解密模式，若

配置此模式，则会硬件强制返回解密模式）；

3. ECB 模式时配置 AES_KEYRx 寄存器，CTR 或 CBC 模式时还要配置 AES_IVRx 寄存器；
4. 设置 AES_CR·EN=1 开启 AES；
5. 把待解密的密文数据分四次连续写入 AES_DINR 寄存器（先写入 MSB）；
6. 等待 AES_SR 寄存器中 CCF 标志产生；
7. 连续读取 AES_DOUTR 寄存器四次来获取解密后的明文数据（先读出 MSB）；
8. 重复 5、6、7 完成所有数据的解密。

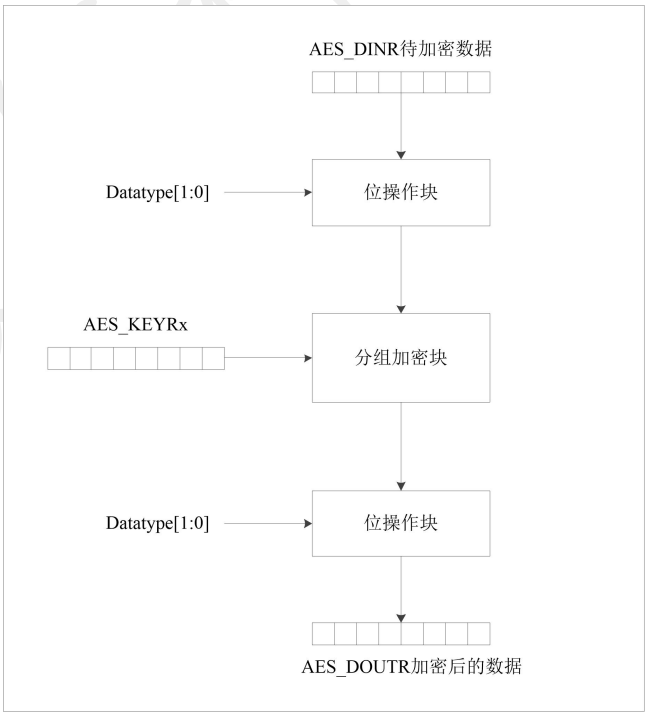
注：此模式下使用的 KEY 为原始密钥，AES 工作过程中会进行密钥派生，并且把派生密钥存放在 AES_KEY 寄存器中；

注意 CTR 模式下不支持此类操作，强行配置会被硬件返回至解密模式；

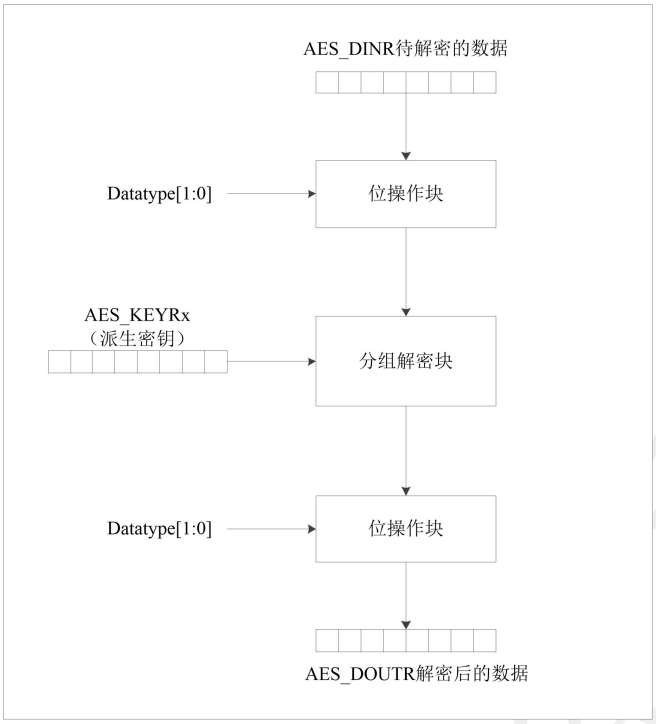
电子密码本（ECB）

这是默认模式。 此模式下不使用 AES_IVR 寄存器。 没有链接操作。 数据分为多个块，每个块分别进行加密。

电子密码本算法的加密和解密的原理图如下：



ECB 模式加密原理图

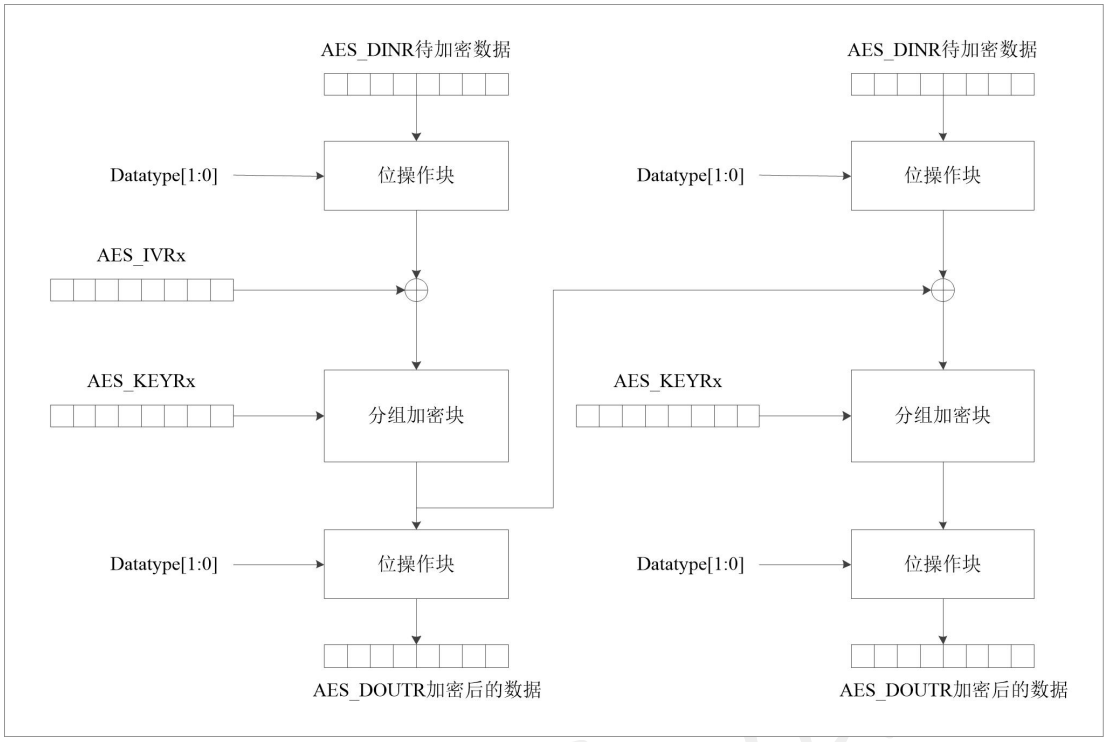


ECB 模式解密原理图

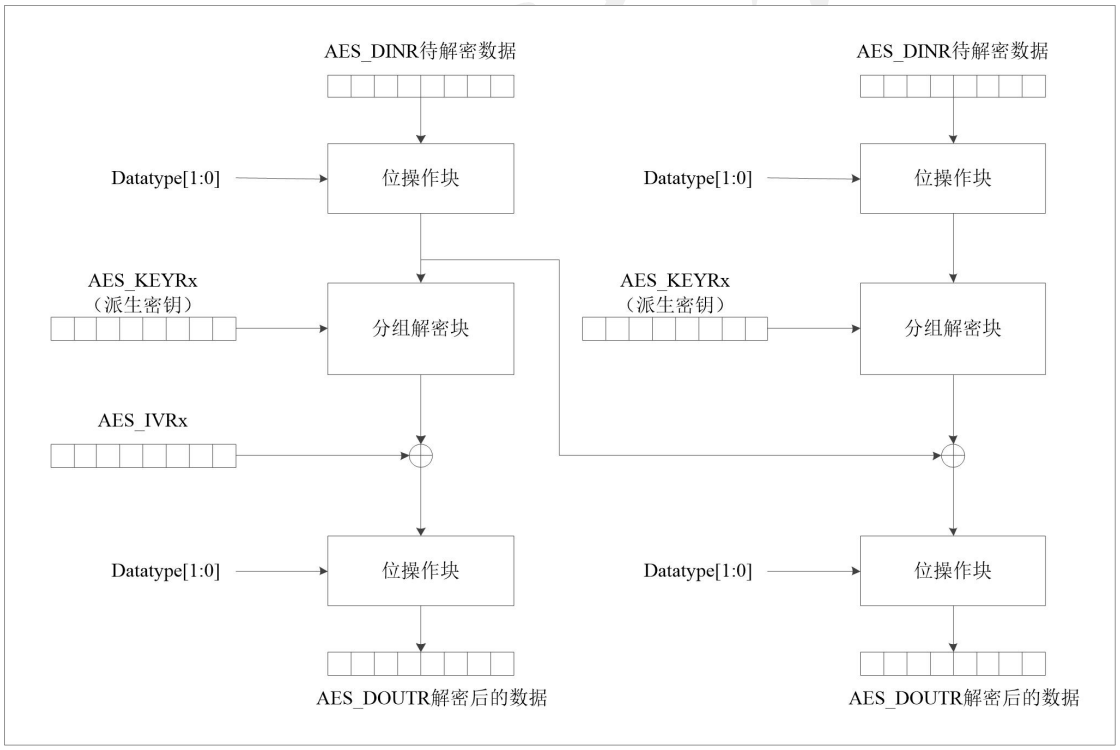
密码块链接（CBC）

在密码块链接（CBC）模式下，每个待加密数据在加密之前都与前一个加密密文进行异或。为了使每个消息都是唯一的，在处理第一个加密块时将使用初始化向量（AES_IVRx）。加密模式下 IV 在加密块之前参与异或，解密模式下 IV 在解密块之后参与异或。

密码块链接算法的加密和解密的原理图如下：



CBC 模式加密原理图

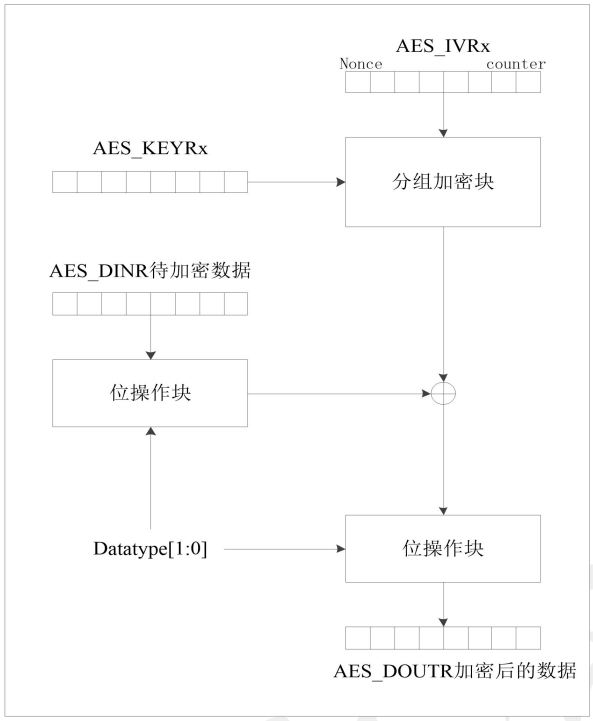


CBC 模式解密原理图

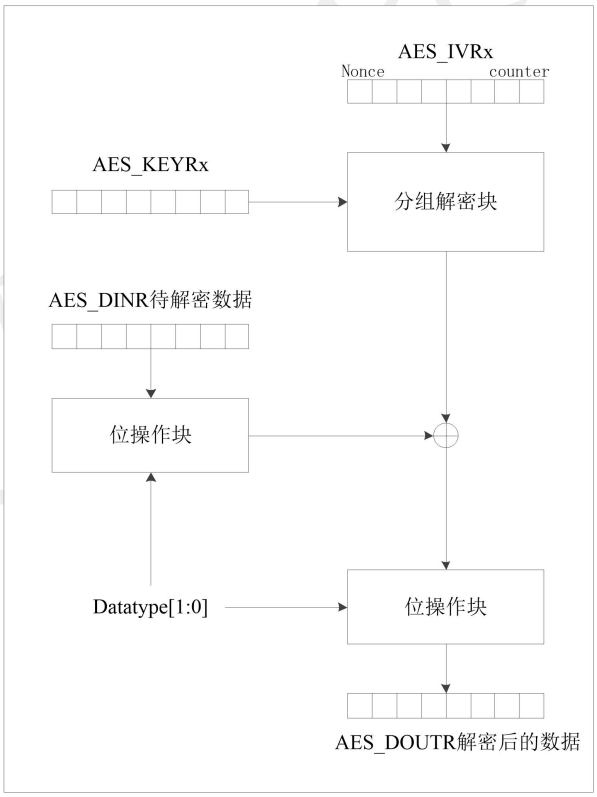
计数器模式（CTR）

在计数器模式下，还使用一个 32 位计数器用于与密文或待加密数据进行 XOR 操作。

计数器模式链接算法的加密和解密的原理图如下：



CTR 模式加密原理图

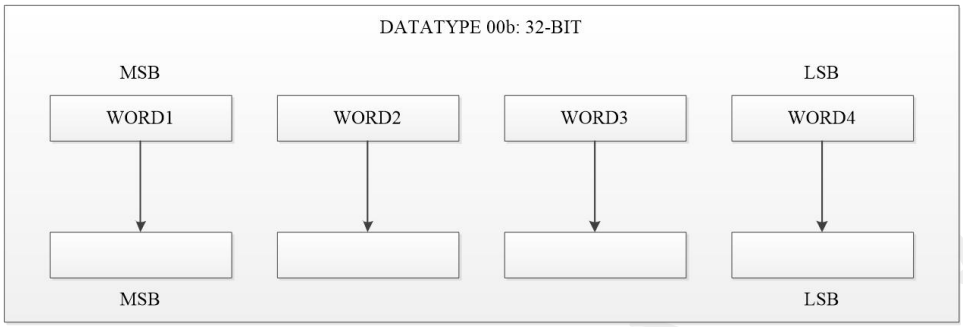


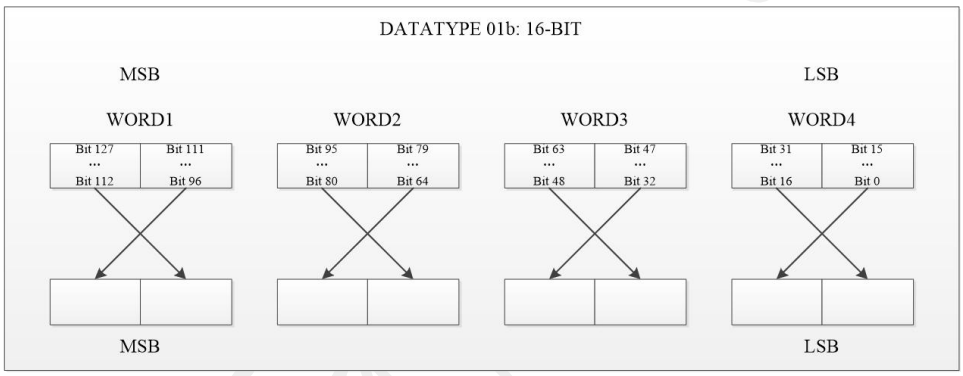
CTR 模式解密原理图

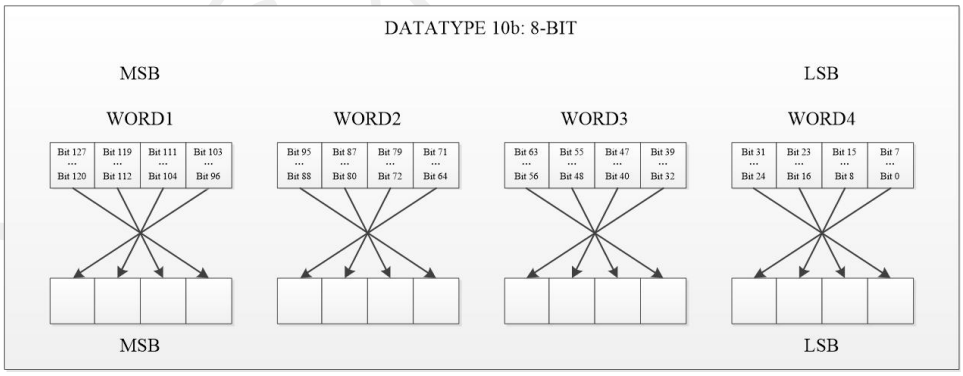
数据模式

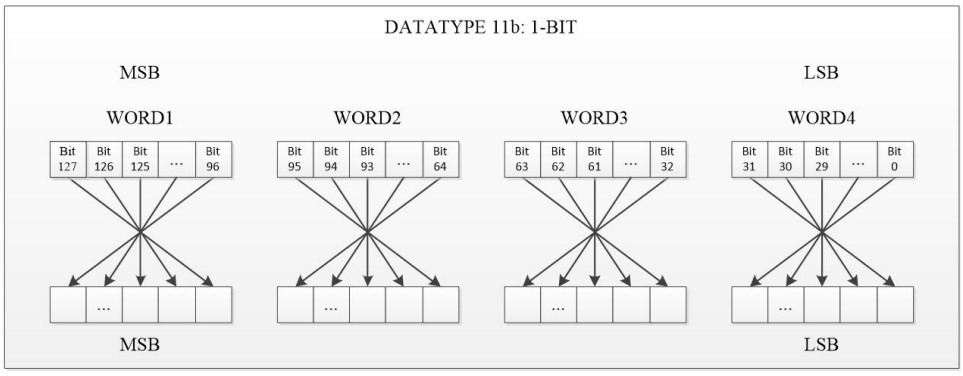
AES 提供了四种数据格式来对输入输出数据进行移位操作，具体移位操作请参照算法说明，可以通过修改 AES_CR.DATATYPE 位进行设置。

具体位移原理如下图：









5.17.4 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
AES	BASE: 0x400A3800				
AES_CR	0x00	32	R/W	0	AES 控制寄存器
AES_SR	0x04	32	R	0	AES 状态寄存器
AES_DINR	0x08	32	R/W	0	AES 输入数据寄存器
AES_DOUTR	0x0c	32	R	0	AES 输出数据寄存器
AES_KEYR0	0x10	32	R/W	0	AES 密钥寄存器 0
AES_KEYR1	0x14	32	R/W	0	AES 密钥寄存器 1
AES_KEYR2	0x18	32	R/W	0	AES 密钥寄存器 2
AES_KEYR3	0x1C	32	R/W	0	AES 密钥寄存器 3
AES_IVR0	0x20	32	R/W	0	AES 加密起始点寄存器 0
AES_IVR1	0x24	32	R/W	0	AES 加密起始点寄存器 1
AES_IVR2	0x28	32	R/W	0	AES 加密起始点寄存器 2
AES_IVR3	0x2C	32	R/W	0	AES 加密起始点寄存器 3

5.17.5 寄存器描述

AES_CR 寄存器（0x00）

位域	名称	类型	复位值	描述
31:11	RESERVED	R	0	保留位
10	ERRIE	R/W	0	错误中断使能。 RDERR 或者 WRERR 都能生成中断。 0：错误中断禁用 1：错误中断启用

9	CCFIE	R/W	0	计算完成中断使能。 0：CCF 中断禁用 1：CCF 中断启用
8	ERRC	R/W	0	错误标志清除 将 1 写入此位将清除 RDERR 和 WRERR 标志。 此位读出为低。
7	CCFC	R/W	0	计算完成标志清除。 此位读出为低。
6:5	CHMOD	R/W	0	AES 链接模式选择。 00：电子码本（ECB） 01：密码块链接（CBC） 10：计数器模式（CTR） 11：保留。 只有在禁用 AES 时才能更改 AES 链接模式。 禁止在启用 AES 时写入这些位，以避免不可预知的 AES 行为。
4:3	MODE	R/W	0	AES 模式选择。 00：模式一：加密 01：模式二：密钥派生 10：模式三：解密 11：模式四：密钥派生+解密 只有禁用 AES 时，才能更改操作模式。 禁止在启用 AES 时写入这些位，以避免不可预知的 AES 行为。 如果选择 CTR 模式，则禁止模式 4。如果软件试图为 CTR 模式配置模式 4，则将强制进入模式 3。

2:1	DATATYPE	R/W	0	数据类型选择。 00：32 位数据，不进行位交换。 01：16 位数据或半字，进行半字交换。例如，原数据 0x764356AB，给加密块的值是 0x56AB7643 10：8 位数据或字节，所有字节进行交换。例如，原数据是 0x764356AB，给加密块的值是 0xAB564376。 11：1 位数据。所有位都进行交换。例如，原数据为 0x00112233，给加密块的值是 0xCC448800 只有禁用 AES 时，才能更改操作模式。 禁止在启用 AES 时写入这些位，以避免一些不可预知的 AES 行为。
0	EN	R/W	0	AES 使能信号。 AES 随时可以通过重置此位来初始化： 当 EN 设置之后 AES 准备开始处理新块。 当 AES 在模式 2（密钥派生）完成操作时，该位将由硬件自动清除

AES_SR 寄存器（0x04）

位域	名称	类型	复位值	描述
31:3	RESERVED	R	0	保留位。
2	WRERR	R	0	写入错误标志。 当检测到来自 AES_DINR 寄存器的意外读取操作（在计算或数据输入阶段）时，该位由硬件设置。 软件通过在 AES_CR 寄存器中设置 ERRC 位来清除。 0：未检测到写入错误 1：检测到写入错误 这个标志对 AES 没有影响，即使产生了 WRERR 中断，AES 也会继续运行。

1	RDERR	R	0	读取错误标志。 当检测到来自 AES_DOUTR 寄存器的意外读取操作（在计算或数据输入阶段）时，该位由硬件设置。 软件通过在 AES_CR 寄存器中设置 ERRC 位来清除。 0：未检测到读取错误 1：检测到读取错误 这个标志对 AES 没有影响，即使产生了 RDERR 中断，AES 也会继续运行。
0	CCF	R	0	计算完成标志。 如果 CCFIE 位先前已在 AES_CR 寄存器中设置。 软件通过在 AES_CR 寄存器中设置 CCFC 位来清除。 1：计算完成 0：计算未完成

AES_DINR 寄存器（0x08）

位域	名称	类型	复位值	描述
31:0	DINR	R/W	0	输入数据寄存器。 在输入阶段，必须将该寄存器写入 4 次： - 在模式 1（加密）中，必须写入 4 个字节，表示从 MSB 到 LSB 的数据。 - 在模式 2（密钥派生）中，不使用此寄存器，因为此模式仅涉及从 AES_KEYRx 寄存器开始的密钥派生计算。 - 在模式 3（解密）和模式 4（密钥派生+解密）中，必须写入 4 个字节，表示密码文本 MSB 到 LSB。 注：此寄存器必须以 32 位数据宽度访问。

AES_DOUTR 寄存器（0x0C）

位域	名称	类型	复位值	描述
31:0	DOUTR	R	0	数据输出寄存器 这个寄存器是只读的。 一旦产生了 CCF 标志（计算完成标志），读取该数据寄存器 4 次，即可访问 128 位输出结果： -在模式 1（加密）中，读取的 4 个字节表示从 MSB 到 LSB 的加密数据。 -在模式 2（密钥派生）中，不需要读取该寄存器，因为派生密钥位于 AES_KEYRx 寄存器中。 -在模式 3（解密）和模式 4（密钥派生+解密）中，读取的 4 个字表示从 MSB 到 LSB 的纯文本。 注：此寄存器必须以 32 位数据宽度访问。

AES_KEYR0 寄存器（0x10）

位域	名称	类型	复位值	描述
31:0	KEYR	R/W	0	密钥寄存器[31:0] 必须在打开 AES 使能之前写入此寄存器： 在模式 1（加密）、模式 2（密钥派生）和模式 4（密钥派生+解密）中，要写入的值表示来自 LSB 的加密密钥，这意味着密钥[31:0]。 在模式 3（解密）中，要写入的值表示来自 LSB 的解密密钥 [31:0]。当在这种解密模式下用加密密钥写入寄存器时，在启用 AES 之前读取它将返回加密值。设置 CCF 标志后读取它将返回派生密钥。 启用 AES 时读取此寄存器将返回一个不可预测的值。 注意：此寄存器不包含模式 4 中的派生密钥（派生密钥+解密）。它始终包含加密密钥值。

AES_KEYR1 寄存器（0x14）

位域	名称	类型	复位值	描述
31:0	KEYR	R/W	0	密钥寄存器[63:32]

AES_KEYR2 寄存器（0x18）

位域	名称	类型	复位值	描述
31:0	KEYR	R/W	0	密钥寄存器[95:64]

AES_KEYR3 寄存器（0x1C）

位域	名称	类型	复位值	描述
31:0	KEYR	R/W	0	密钥寄存器[127:96]

AES_IVR0 寄存器（0x20）

位域	名称	类型	复位值	描述
31:0	IVR	R/W	0	初始化向量寄存器[31：0] 必须在 AES_CR 寄存器的 EN 位置 1 之前写入此寄存器： 在以下情况下，寄存器值没有意义： - ECB 模式（电子密码本）。 - CTR 或 CBC 模式下的密钥派生。 在 CTR 模式（计数器模式）下，该寄存器包含 32 位计数器值。 在启用 AES 的同时读取该寄存器将返回值 0x00000000。

AES_IVR1 寄存器（0x24）

位域	名称	类型	复位值	描述

31:0	IVR	R/W	0	初始化向量寄存器[63:32] 在 CTR 模式（计数器模式）下，该寄存器包含随机数值。 在启用 AES 的同时读取该寄存器将返回值 0x00000000。

AES_IVR2 寄存器（0x28）

位域	名称	类型	复位值	描述
31:0	IVR	R/W	0	初始化向量寄存器[95:64] 在 CTR 模式（计数器模式）下，该寄存器包含随机数值。 在启用 AES 的同时读取该寄存器将返回值 0x00000000。

AES_IVR3 寄存器（0x2C）

位域	名称	类型	复位值	描述
31:0	IVR	R/W	0	初始化向量寄存器（MSB IVR [127：96]） 在 CTR 模式（计数器模式）下，该寄存器包含随机数值。 在启用 AES 的同时读取该寄存器将返回值 0x00000000。

5.18 OTP 控制器

5.18.1 概述

OTP 控制器位于 CPU 和 OTP 之间，通过 CPU 的读和写操作实现对 OTP 的读操作和编程操作。

5.18.2 特性

- 根据系统时钟频率需要选择合适的读速率模式，以便满足正常的 OTP 读时间的要求
- 可以对 OTP 进行加锁操作，防止用户误操作修改 OTP 内的数据
- 可以通过两种方式对 OTP 进行读操作

5.18.3 功能描述

使用流程

OTP 编程数据：

（1） 等待 OTP 初始化完成，也就是等待 OTPPROGEN 寄存器的 OTP_INIT 位是 0

（2） 将要写入的数据放到编程数据寄存器 OTPPROGDATA 中

（3） 将要写入的地址写入地址寄存器 OTPADDR 中

（4） OTP 编程使能，也就是将 OTPPROGEN 寄存器的 PROG_EN 位置为 1

（5） 等待编程结束，也就是等待 OTPPROGEN 寄存器的 PROG_ST 位是 0

（6） 判断编程是否成功，也就是判断 OTPPROGEN 寄存器的 PROG_FAIL 位是否为 0

（7） 编程失败的话，清除编程失败标志

OTP 读取数据：

（1） 等待 OTP 初始化完成，也就是等待 OTPPROGEN 寄存器的 OTP_INIT 位是 0

（2） 将要读取的地址写入地址寄存器 OTPADDR 中

（3） OTP 读使能，也就是将 OTPRDEN 寄存器的 READ_EN 位置为 1

（4） 等待读操作完成，也就是等待 OTPRDEN 寄存器的 READ_ST 位为 0

（5） 从 OTPRDATA 寄存器读取该地址的数据

OTP 还可以直接通过 0x41000000 区域读取数据，具体操作如下：

addr = (addr>>2)<<2; 地址字对齐

return *((volatile unsigned int *)(0x41000000 + addr)); 读取该地址的数据

5.18.4 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
OTPREG	BASE: 0x400A3000				
OTPCFG	0x00	32	R/W	1	OTP 控制寄存器
OTPPROGDATA	0x04	32	R/W	0	OTP 编程数据寄存器
OTPADDR	0x08	32	R/W	0	OTP 地址寄存器
OTPPROGEN	0x0c	32	RW	0	OTP 编程使能寄存器
OTPRDATA	0x10	32	R/W	0	OTP 读数据寄存器
OTPRDEN	0x14	32	R/W	0	OTP 读使能寄存器
OTPPROG_UNLOCK	0x18	32	W	0	OTP 编程解锁寄存器

5.18.5 寄存器描述

OTPCFG 寄存器（0x00）

位域	名称	类型	复位值	描述
31	OTP_SDY	R/W	0	配置 OTP 进入 standy 状态寄存器 1：OTP 进入 standy 状态 0：OTP 正常工作状态 该操作只能在 RAM 中执行。
30：24	RESERVED	R	0	保留位

23：16	TCPS	R/W	0x3E	配置 OTP 编程时 TCPS 时间寄存器 根据系统时钟的频率，需要对 TCPS 进行适当配置，使得 TCPS 时间至少达到 2.5us 的要求。 该寄存器同时也控制 TPEH（该参数要求最小值为 2us，配置值与 TCPS 相同），以及 TPES/TCPH/TRDEP（这三个参数要求最小为 1us，配置值为 TCPS/2）。 注：默认值处于系统为 25MHz 情况下
15	RESERVED	R	0	保留位
14：4	TPW	R/W	0x271	配置 OTP 编程时 TPW 时间寄存器 根据系统时钟的频率，需要对 TPW 进行适当配置，使得 TPW 时间在 23us~27us 范围内 注：默认值处于系统为 25MHz 情况下
3：0	READ_MD	R/W	4’h2	读速率模式选择 4’h0：4 个 sys_clk 4’h1：8 个 sys_clk 4’h2：12 个 sys_clk …… 4’hf：64 个 sys_clk 注 1：该寄存器表示从 OTP 的 READEN 有效开始，间隔多少个 sys_clk 对 OTP 输出的数据进行采样。 注 2：根据系统时钟频率需要选择合适的读速率模式，以便满足正常的 OTP 读时间的要求。 注 3：该寄存器主要为了满足 OTP 在低电压（例如 2v 工作电压）下能够正常进行读操作。

OTPPROGDATA 寄存器（0x04）

位域	名称	类型	复位值	描述

31:0	PROGDATA	R/W	0	编程数据寄存器 只有数据 1 才能被编程进 OTP 对应的地址中，数据 0 被忽略。

OTPADDR 寄存器（0x08）

位域	名称	类型	复位值	描述
31：12	RESERVED	R	0	保留位
11：0	OTPADDR	R/W	0	OTP 地址寄存器。编程操作时表示编程地址，寄存器读操作时表示读地址。 其中： 编程操作时，每次编程数据量为 32bit。 因此， 该地址为 0，表示往 OTP 第一个字的起始地址开始连续编程 32bit 数据； 该地址为 1，表示往 OTP 第二个字的起始地址开始连续编程 32bit 数据； 该地址为 2，表示往 OTP 第三个字的起始地址开始连续编程 32bit 数据； …… 以此类推。 寄存器读操作时，每次读操作都是读出 32bit 数据。 因此： 该地址为 0，表示读出 OTP 中第一个字的数据； 该地址为 1，表示读出 OTP 中第二个字的数据； 该地址为 2，表示读出 OTP 中第三个字的数据； …… 以此类推。

OTPPROGEN 寄存器（0x0C）

位域	名称	类型	复位值	描述
31:12	RESERVED	R	0	保留位
11:4	PROG_LOCK	W	0	将该寄存器写入 0xAA 后，OTP 的编程操作不能被执行，用于保护对 OTP 的误改写。
3	OTP_INIT	R	0	OTP 初始化状态 1：表示 OTP 正在进行初始化，还未进入 READY 状态 0：表示 OTP 已进入 READY 状态，可以接收通过寄存器配置发出的读操作或编程操作
2	PROG_FAIL	R/W	0	编程失败标志 1：表示编程失败 0：表示编程成功 启动编程后，当检测到 PROG_ST 由 1 变为 0 后，判断该寄存器是否为 1。如果为 0 表示此次编程全部成功，如果为 1 表示此次编程存在失败的情况。该寄存器写 1 清零。
1	PROG_ST	R	0	编程状态 1：表示正在进行编程 0：表示没有进行编程 将 PROG_EN 配置为 1 后，PROG_ST 会输出 1，表示正在进行编程，当编程结束后，PROG_ST 为 0

0	PROG_EN	R/W	0	编程使能 将该位置 1 后，启动一次 program 操作。 将 PROGDATA 中存入的数据编程进以 PROGADDR 为起始的地址中。并且仅将数据中为“1”的数值编程进对应的地址中。 该寄存器写 1 后仅维持一个系统时钟周期。 编程操作只能在 RAM 中执行。

OTPRDATA 寄存器（0x10）

位域	名称	类型	复位值	描述
31：0	RDATA	R	0	表示读出从配置读操作地址开始的 32bit 数据

OTPRDEN 寄存器（0x14）

位域	名称	类型	复位值	描述
31：2	RESERVED	R	0	保留位
1	READ_ST	R	0	读 OTP 操作状态位（该状态既包括寄存器读的方式，也包括 AHB 读的方式） 1：表示正在进行读操作 0：表示没有进行读操作
0	READ_EN	R/W	0	寄存器读操作使能 将该位置 1 后，启动一次读 OTP 操作。当该读操作完成后自动清零。

OTPPROG_UNLOCK 寄存器（0x18）

位域	名称	类型	复位值	描述
31：0	PROG_UNLOCK	W	0	将该寄存器写入 0x12345678 后，OTP 的编程锁定状态解锁，可以对 OTP 进行编程

5.19 IIC 控制器

5.19.1 概述

IIC(Inter-Integrated Circuit) 是一种串行通信总线，使用多主从架构， IIC 总线在物理连接上非常简单，分别由 SDA(串行数据线)和 SCL(串行时钟线)及上拉电阻组成。通信原理是通过对 SCL 和 SDA 线高低电平时序的控制，来产生 IIC 总线协议所需要的信号进行数据的传递。在总线空闲状态时，这两根线一般被上面所接的上拉电阻拉高，保持着高电平。IIC 通信方式为半双工，只有一根 SDA 线，同一时间只可以单向通信。IIC 总线数据传输速率在标准模式下可达 100kbit/s，快速模式下可达 400kbit/s。一般通过 IIC 总线接口可编程时钟来实现传输速率的调整，同时也跟所接的上拉电阻的阻值有关。芯片提供了一个 IIC 接口模块，实现与外部的 IIC 设备通信。

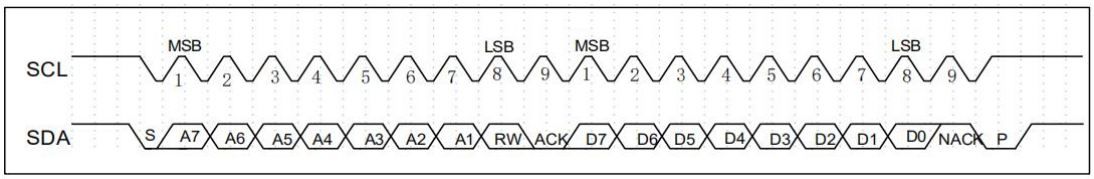
5.19.2 功能描述

协议介绍

IIC 总线采用串行数据线(SDA)和串行时钟线(SCL)传输数据。IIC 总线的设备端口为开漏输出，必须在接口外接上拉电阻。

数据在主从设备之间通过 SCL 时钟信号在 SDA 数据线上逐字节同步传输。每一个 SCL 时钟脉冲发送一位数据，高位在前。每发送一个字节的数据产生一个应答信号。在时钟线 SCL 高电平期间对数据的每一位进行采样。数据线 SDA 在时钟线 SCL 为低电平时改变，在时钟线 SCL 为高电平时保持稳定。

通常情况下，一个标准的通信包含四个部分：开始信号、从机地址、数据传输、停止信号。如下图所示：



开始信号发送

当总线空闲时，表示没有主机设备占用总线（SCL 和 SDA 都保持高电平），主机可以通过发送一个开始信号启动传输。启动信号，通常被称为 S 位。SCL 为高电平时，SDA 由高电平向低电平跳变。启动信号表示开始新的数据传输。

当命令寄存器 CR 的 STA 位被置位，同时 WR 位被置位时，系统核心产生一个开始信号。根据 SCL 当前的不同状态，生成开始信号。

从机地址发送

在开始信号后，由主机传输的第一个字节数据是从机地址。包含 7 位的从设备地址和 1 位的 R/W 指示位。R/W 指示位信号表示与从机的数据传输方向，0 表示写操作，1 表示读操作。在系统中的从机不可以具有相同的地址。 只有从机地址和主机发送的地址匹配时才能产生一个应答位（在第九个时钟周期拉低 SDA）进行响应。

发送从机地址为一次写操作，在传输寄存器中保存从机地址并对 WR 位置位，从机地址将被发送到总线上。

数据传输

一旦成功取得了从机地址，主机就可以通过 R/W 位控制逐字节的发送数据。每传输一个字节都需要在第九个时钟周期产生一个应答位。

如果从机信号无效或者从机返回一个 NACK 信号，主机可以生成一个停止信号中止数据传输。

如果主机作为接收设备，没有应答从机，从机就会释放 SDA，主机产生停止信号。

向从机写入数据，需把将要发送的数据存入传输寄存器中并设置 WR 位。从从机中读取数据，需设置 RD 位。在数据传输过程中系统核心设置 TIP 提示标志，指示传输正在进行。当传输完成后 TIP 提示标志会自动清除。当中断使能时，中断标志位 IF 被置位，并产生中断。当中断标志位 IF 被置位后，接收寄存器收到有效数据。当 TIP 提示标志复位后，用户可以发出新的写入或读取命令。

停止信号发送

主机可以通过生成一个停止信号终止通信。停止信号通常被称为 P 位，被定义为 SCL 为高电平时，SDA 由低电平向高电平跳变。

当命令寄存器 CR 的 STO 位被置位，同时 WR 位被置位时，系统核心产生一个停止信号。根据 SCL 当前的不同状态，生成停止信号。

操作流程

初始化配置

- 打开 IIC 时钟
- 端口配置为 IIC 功能，打开输入使能
- 配置 CTRL 寄存器的 EN 位，关闭 IIC 模块，确保配置寄存器过程中模块未工作
- 配置 CLKDIV 寄存器的 CLKDIV 位，设置 IIC 传输速度，计算公式见寄存器描述
- 配置 CTRL 寄存器的 EN 位，打开 IIC 模块

主机发送数据

- 主机发送从机器件地址：将从机的 7 位器件地址写入 TXR 寄存器，高 7 位为器件地址，最后一位为 0。
- 配置 CR 寄存器 STA 位和 WR 位为 1，发送起始信号和写命令。
- 发送数据：将需要往从机发送的数据写入 TXR 寄存器，同时置 CR 寄存器 WR 位为 1。数据发送完成后，SR 寄存器的 TIP 位变为 0，可通过查询该位确认发送完成。从机成功接收到数据后，向主机返回 ACK，主机接收到 ACK 后，SR 寄存器的 RACK 位变为 0。
- 主机按上步骤可重复发送数据，数据发送完成后置 CR 寄存器 STO 位为 1，则总线发送 停止信号，停止写入数据。

主机接收数据

IIC 作为主机从从机读取数据操作流程如下（以 EEPROM 流程为例）：

- 主机发送从机器件地址：将从机的 7 位器件地址写入 TXR 寄存器，高 7 位为器件地址，最后一位为 0。
- 配置 CR 寄存器 STA 位和 WR 位为 1，发送起始信号和写命令。
- 主机发送读取数据的地址：把读取数据的地址写入 TXR 寄存器，同时配置 CR 寄存器 WR 位为 1。
- 主机再次发送从机器件地址：将从机的 7 位器件地址写入 TXR 寄存器，高 7 位为器件地址，最后一位为 1。
- 配置 CR 寄存器 WR 位为 1，启动写命令。
- 读取数据：向从机发送读取命令，配置 CR 寄存器 RD 位为 1。数据传输完成后， CR 寄存器的 TIP 位变为 0，主机可通过读取 RXR 寄存器来读取从机数据。
- 主机按上述步骤可重复读取数据，当最后一个数据读取完成时，主机要向从机返 NACK 和停止信号。

5.19.3 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
IIC BASE: 0x40045000					
CLKDIV	0x00	32	R/W	FFFFH	IIC 分频控制器
CTRL	0x04	32	R/W	0	控制寄存器
TXR	0x08	32	R/W	0	发送寄存器
RXR	0x0c	32	RW	0	接收寄存器
CR	0x10	32	R/W	0	命令寄存器
SR	0x14	32	R/W	0	状态寄存器

5.19.4 寄存器描述

CLKDIV 寄存器描述（0x00）

位域	名称	类型	复位值	描述

31:16	RESERVED	R	0	保留位
15:0	CLKDIV	R/W	0xFFFF	分频控制寄存器，将内部工作频率分到 SCL 的 5 倍。 例如：SYSCLK=48MHZ,SCL 频率为 100KHZ，则需要设置 CLKDIV=48*1000/（5*100）-1 = 95 = 0x5F

CTRL 寄存器描述（0x04）

位域	名称	类型	复位值	描述
31:2	RESERVED	R	0	保留位
1	IE	R/W	0	中断使能 1:使能中断 0:禁能中断
0	EN	R/W	0	模块使能 1:使能模块 0:禁能模块

TXR 寄存器描述（0x08）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	TXR	W/R	0	希望发送到 I2C 总线上的下一个字节 BIT[0]:在数据传输过程中，这一位是数据的 LSB，在 slave 地址传输过程中，这一位 表示 RW。1：从 slave 读数据，0：向 slave 写数据。

RXR 寄存器描述（0x0C）

位域	名称	类型	复位值	描述
31:8	RESERVED	R	0	保留位
7:0	RXR	R	0	从 I2C 总线上接收的最后一个字节，RO

CR 寄存器描述（0x10）

位域	名称	类型	复位值	描述
31:6	RESERVED	R	0	保留位
5	STA	W	0	产生 START 自动清零
4	STO	W	0	产生 STOP 自动清零
3	RD	W	0	1：需从 Slave 读数据 0：不需从 Slave 读数据 自动清零
2	WR	W	0	1：向 Slave 写数据 0：不向 Slave 写数据 自动清零
1	ACK	W	0	接收模式下： 0：向总线反馈 ACK 1：向总线反馈 NACK
0	IF	W	0	向这一位写 1，清掉等待中的中断 1：清中断 0：不清中断

SR 寄存器描述（0x14）

位域	名称	类型	复位值	描述
31:5	RESERVED	R	0	保留位
4	RACK	R	0	接收到从设备发送的 ACK 位： 0：收到 ACK 1：收到 NACK

3	BUSY	R	0	总线忙 1：当检测到 START 0：当检测到 STOP
2	ARB	R	0	仲裁丢失 1：I2C 模块失去总线的访问权 0：I2C 模块得到总线的访问权
1	TIP	R	0	传输状态 1：传输正在进行中 0：传输已经结束
0	IF	R	0	1：产生中断 0：未产生中断 向 CR 的 IF 位写 1 清零，R/W1C 注：一个字节传输完成或总线访问权丢失，该位为 1

5.20 CACHE 控制器

5.20.1 概述

CACHE 控制器位于 CPU 和程序 RAM 之间。将慢速程序存储器中的数据，通过 CACHE 算法映射到程序 RAM，CPU 可以直接读取程序 RAM 中的数据，从而实现 CPU 的取指过程。

通过该模块可以使 CPU 以较低的代价，获得较大的可执行代码空间，提高代码执行效率。

5.20.2 特性

- 预取功能：预先将指定 OTP 的数据填充到 CACHE RAM LINE 中，并将该 LINE 设置为锁定状态，不允许硬件在动态替换时对其换出。
- 解锁功能：将处于锁定状态的 LINE 进行解锁，允许硬件在动态替换时对其换出。
- CACHE 缓存 RAM 大小为 1KBytes：共有 32 个 LINE，每个 LINE 的深度为 32 字节。
- CACHE 的寻址空间大小固定为 16KBytes。
- 预取和解锁之前需要把中断关掉。预取和解锁完成后再打开中断。
- 预取和解锁的程序必须放在 RAM 中执行。

5.20.3 寄存器映射

名称	偏移量	位宽	类型	复位值	描述
CACHE	BASE: 0x40002000				
CACHE_CFG	0x00	32	R/W	1	CACHE 配置寄存器
PF_CTRL	0x04	32	R/W	0	预取控制寄存器

5.20.4 寄存器描述

CACHE_CFG 寄存器描述（0x00）

位域	名称	类型	复位值	描述
30：2	RESERVED	R	0	保留位
1	IDLE	R	0	为 1 表示当前 CACHE 处于空闲状态。既没有发生 MISS，也没有处于填充或预取状态。

0	RESET	R/W	4'h8	复位信号，用于复位 CACHE 模块内的所有寄存器。高有效 在复位 CACHE 模块前，必须查询 IDLE，直到其为 1，表明目前没有在从 FLASH 取数。否则，复位 CACHE，但相应的 FLASH 控制器和 FLASH 的状态却没有复位，会造成取指错误。 写 RESET 寄存器的程序必须在 RAM 或 ROM 里面执行，不能在 CACHE 里面。否则如果 CACHE 正在从 FLASH 取指，复位 CACHE 可能导致 FLASH 控制器状态错误。

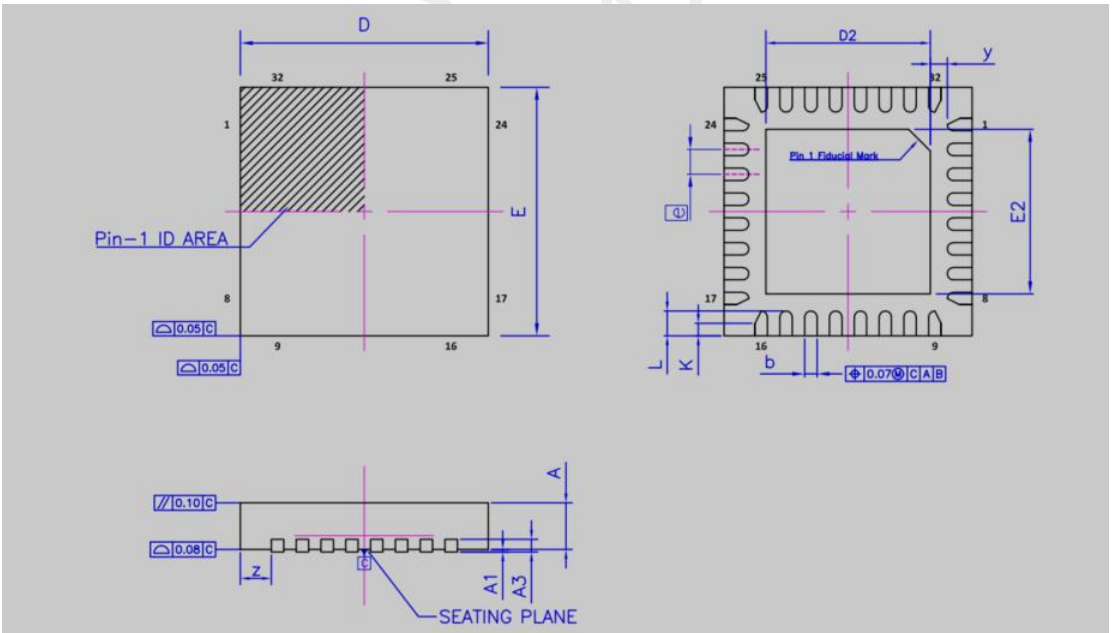
PF_CTRL 寄存器描述（0x04）

位域	名称	类型	复位值	描述
31:13	RESERVED	R	0	保留位
12:4	PR_ADDR	RW	0	需要预取时，表示预取空间的基地址。 基地址要 Line 对齐。
3:2	RESERVED	R	0	保留位
1	UNLOCK	RW	0	预取的空间 UNLOCK 配置寄存器。 预取启动后，若该位为 1，则预取后，对应的 LINE 处于 UNLOCK 状态，可以在动态替换时对其换出。

0	PF_START	RW	0	预取启动位。 向该位写 1 后，启动预取操作，预取结束后，硬件自动清零。 预取的 LINE 完成预取后为锁定状态，则不允许硬件在动态替换时对其换出。 需要对已锁定的 line 进行解锁，则只能重新对该预取空间的基地址重新执行预取操作并且将 UNLOCK 配置为 1。 注 1：预取锁定和解锁的程序必须放在 RAM 中执行； 注 2：预取锁定和解锁时芯片所有中断必须关闭，完成后才可以打开中断。

6. 封装尺寸

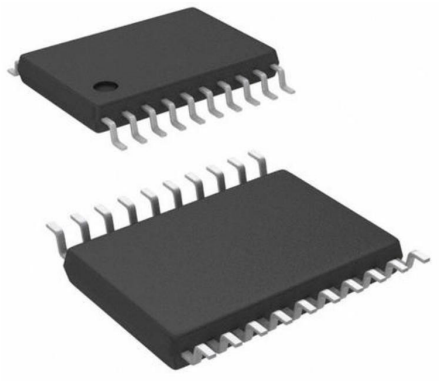
6.1 QFN32 封装尺寸



Dimensions

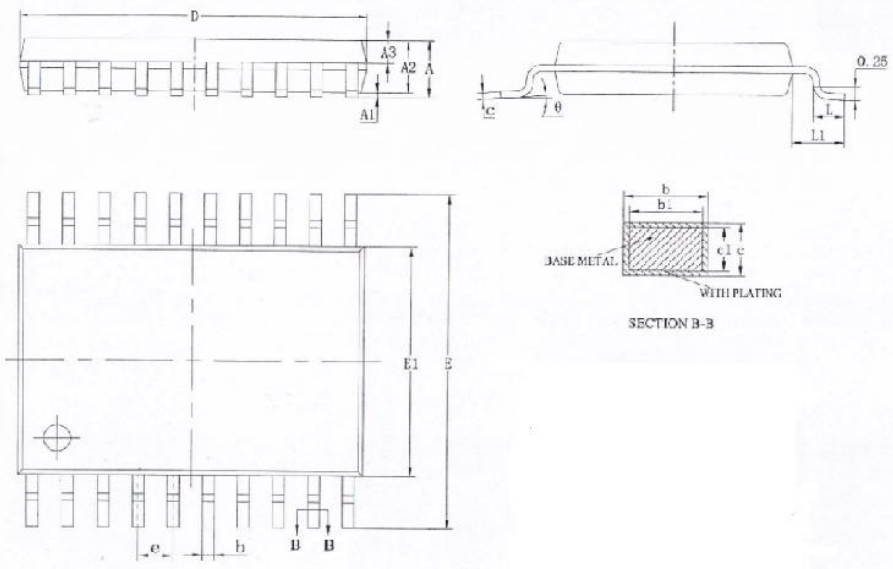
Unit	D	E	D2	E2	A	A1	A3	b	e	K	L	y	z
mm	4.05 (4.00) 3.95	4.05 (4.00) 3.95	2.75 (2.65) 2.55	2.75 (2.65) 2.55	0.80 (0.75) 0.70	0.05 (0.04) 0.00	0.203 REF	0.25 (0.20) 0.15	0.40 BSC	0.20 REF	0.45 (0.40) 0.35	0.275 REF	0.500 REF

6.2 TSSOP20 封装尺寸



TSSOP20L产品外形图

胶体尺寸：6.50mmx4.40mmx1.00mm
引脚间距：0.65mm



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	—	0.28
b1	0.19	0.22	0.25
c	0.13	—	0.17
c1	0.12	0.13	0.14
D	6.40	6.50	6.60
E1	4.30	4.40	4.50
E	6.20	6.40	6.60
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00REF		
θ	0	—	8°