

CIU32F011/F031

32 位触控系列微控制器

参考手册

RM1002



声 明

本手册的版权属北京中电华大电子设计有限责任公司所有。任何未经授权对本手册进行复印、印刷、出版发行的行为，都将被视为是对北京中电华大电子设计有限责任公司版权的侵害。北京中电华大电子设计有限责任公司保留对此行为诉诸法律的权利。

北京中电华大电子设计有限责任公司保留未经通知用户对本手册内容进行修改的权利。虽然我们已经核对本手册的内容，但是差错有时候难以完全避免，所以，我们会对手册的内容进行定期的审查，并在下一版的文件中作必要的修改。建议您在最终设计前从华大电子获取本文档的最新版本。

目录

1. 简介	4
2. 引脚分配与功能描述	5
2.1. 引脚分配图	5
2.2. 引脚功能描述.....	8
3. 系统及存储器架构	12
3.1. 系统架构	12
3.2. 存储器映射	13
4. 嵌入式闪存 (FLASH)	16
4.1. 模块介绍	16
4.2. 功能特点	16
4.3. 功能说明	16
4.4. 模块框图	17
4.5. 寄存器描述	17
5. 中断和事件 (INT/EVT)	24
5.1. 嵌套向量中断控制器.....	24
5.2. 系统滴答(SysTick)校准值寄存器	24
5.3. 中断功能描述.....	24
5.4. 外部中断/事件控制器(EXTI).....	25
6. 循环冗余校验计算单元 (CRC)	26
6.1. 模块介绍	26
6.2. 功能特点	26
6.3. 功能说明	26
6.4. 模块框图	26
6.5. 时钟与复位	26
6.6. 寄存器描述	26
7. 电源管理 (POWER MANAGEMENT)	29
7.1. 电源.....	29
7.2. 电源管理器	29
7.3. 电源控制寄存器.....	30
8. 低功耗 (LOW POWER)	32
8.1. 低功耗模式	32
8.2. 进入低功耗	32
8.3. 低功耗唤醒	32
9. 复位和时钟系统 (RESET/CLOCK)	34
9.1. 引脚复位(MCLR)功能.....	34
9.2. 复位.....	34
9.3. 时钟.....	35
10. 通用输入输出 (GPIO)	37
10.1. 模块介绍	37
10.2. 功能特点	37
10.3. 功能说明	37
10.4. 模块框图	39
10.5. 寄存器描述	40
11. 同步串行接口 (SSP)	48
11.1. 模块介绍	48
11.2. 功能特点	48

11.3. 功能说明	48
11.4. IO 映射	49
11.5. 模块框图与接口时序	49
11.6. 时钟与复位	51
11.7. 寄存器描述	51
12. 通用异步收发器 (UART0/1)	58
12.1. 模块介绍	58
12.2. 功能特点	58
12.3. 功能说明	58
12.4. IO 映射	59
12.5. 模块框图与接口时序	59
12.6. 时钟与复位	59
12.7. 寄存器描述	60
13. 复合通信接口 (UST)	64
13.1. 模块介绍	64
13.2. 功能特点	64
13.3. 功能说明	64
13.4. IO 映射	66
13.5. 模块框图与接口时序	66
13.6. 时钟与复位	68
13.7. 寄存器描述	69
14. 定时器 (TIMER0/1/4)	75
14.1. 模块介绍	75
14.2. 功能特点	75
14.3. 功能说明	75
14.4. IO 映射	79
14.5. 模块框图与接口时序	79
14.6. 时钟与复位	79
14.7. 寄存器描述	79
15. 数模转换模块 (ADC)	85
15.1. 模块介绍	85
15.2. 功能特点	85
15.3. 功能说明	85
15.4. IO 映射	86
15.5. 模块框图与接口时序	86
15.6. 时钟与复位	86
15.7. 寄存器描述	87
16. 触摸按键 (TK)	92
16.1. 模块介绍	92
16.2. 功能特点	92
16.3. 功能说明	92
16.4. 模块框图	92
17. LED 显示控制	93
17.1. 模块介绍	93
17.2. 功能特点	93
17.3. 功能说明	93
18. LCD 驱动模块 (LCD)	95
18.1. 模块介绍	95
18.2. 功能特点	95
18.3. 功能说明	95

18.4. IO 映射	96
18.5. 模块框图与接口时序.....	96
18.6. 时钟与复位	99
18.7. 寄存器描述	99
19. 看门狗 (WDT).....	103
19.1. 模块介绍	103
19.2. 功能特点	103
19.3. 模块框图	103
19.4. 时钟与复位	103
19.5. 寄存器描述	103
20. 系统寄存器 (SYSTEM_REG)	106
20.1. 模块介绍	106
20.2. 寄存器描述	106
21. 器件电子签名 (E-SIGNATURE)	116
21.1. 产品唯一身份标识寄存器 (96 位).....	116
22. 调试支持 (DEBUG)	117
22.1. 引脚分布和调试端口脚	117
22.2. SWD 调试端口脚.....	117
22.3. SW 调试端口	118
22.4. MCU 调试	119
23. 版本历史.....	120
24. 联系方式.....	121

1. 简介

CIU32F011、CIU32F031 是基于 ARM Cortex-M0 内核的触控系列 MCU 产品,支持 SOP16、SOP20、SSOP20、QFN20、SOP28、SSOP28、LQFP32、QFN32 封装,最高频率可达 48MHz,内部集成 LCD、ADC、多种定时器、SPI/I²C/UART 等丰富的外设资源,也包含 8 个大电流 IO 可驱动 LED 显示和最大 26 个触摸按键。

CIU32F011、CIU32F031 触控系列 MCU 应用场景:

- 电磁炉、取暖桌、热水器等带触摸按键的家用电器产品
- 触控开关、智能门锁、美容仪等消费电子产品

2. 引脚分配与功能描述

2.1. 引脚分配图

图 2-1 SOP16(CIU32F031F5FB)

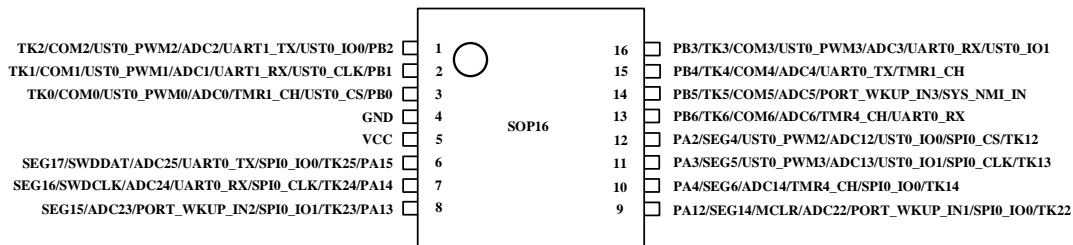


图 2-2 SOP20/SSOP20(CIU32F011G3FB、CIU32F011G3HB、CIU32F031G5FB、CIU32F031G5HB)

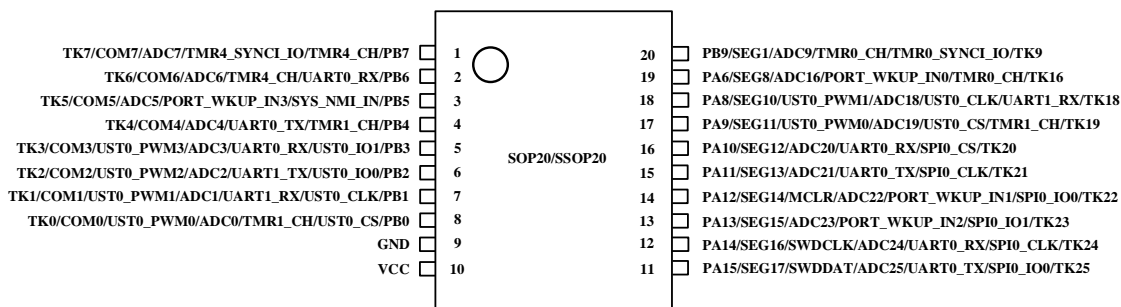


图 2-3 QFN20(CIU32F011G3NB)

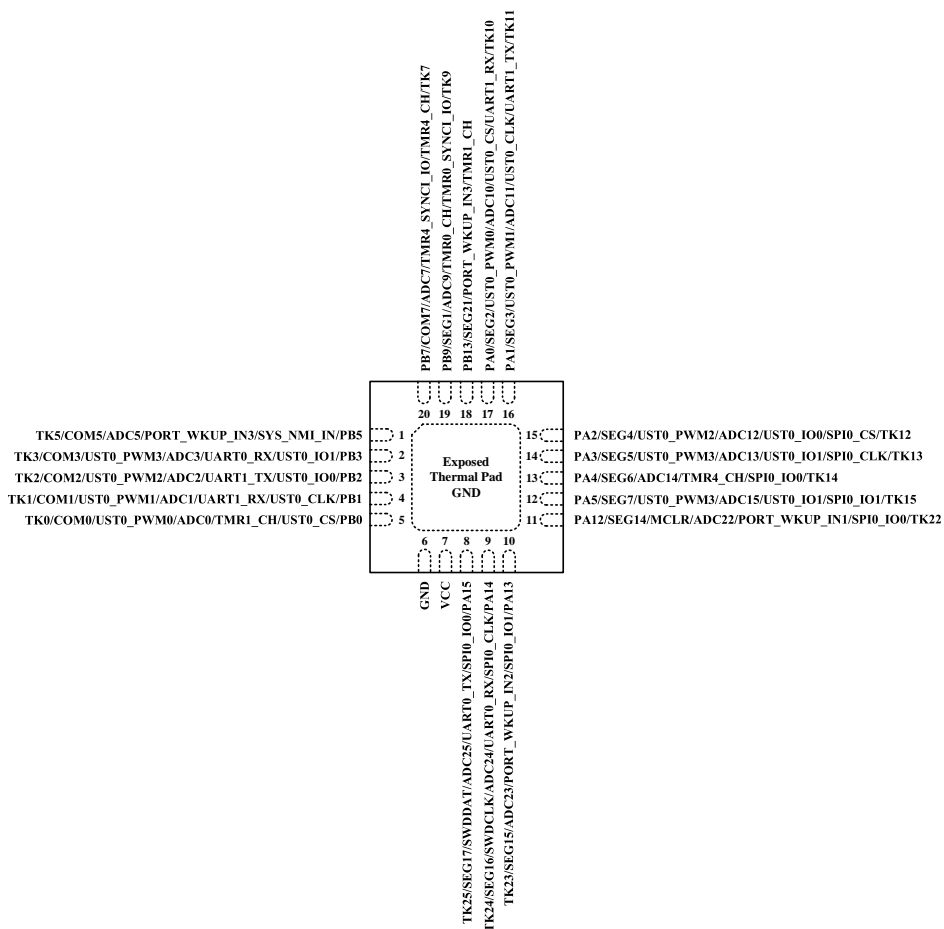


图 2-4 SOP28/SSOP28(CIU32F011I3FB、CIU32F011I3HB、CIU32F031I5FB、CIU32F031I5HB)

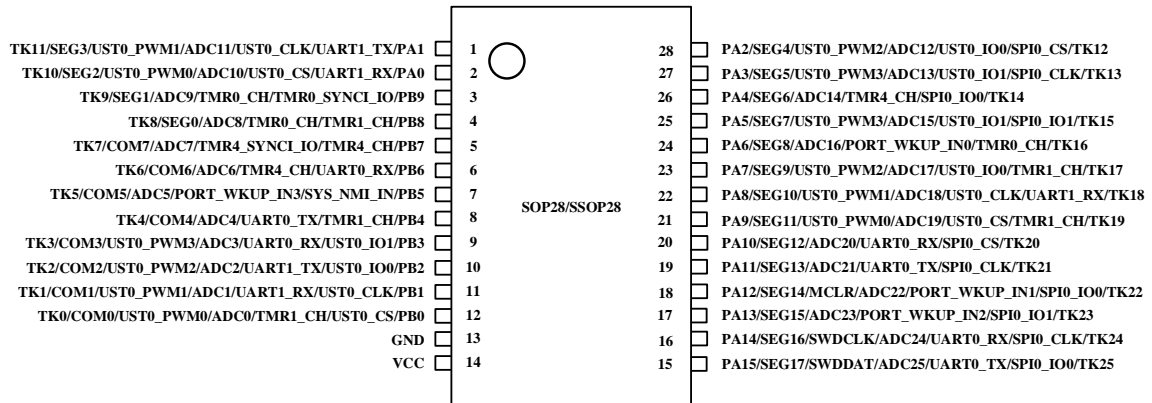


图 2-5 LQFP32(CIU32F031J5PB)

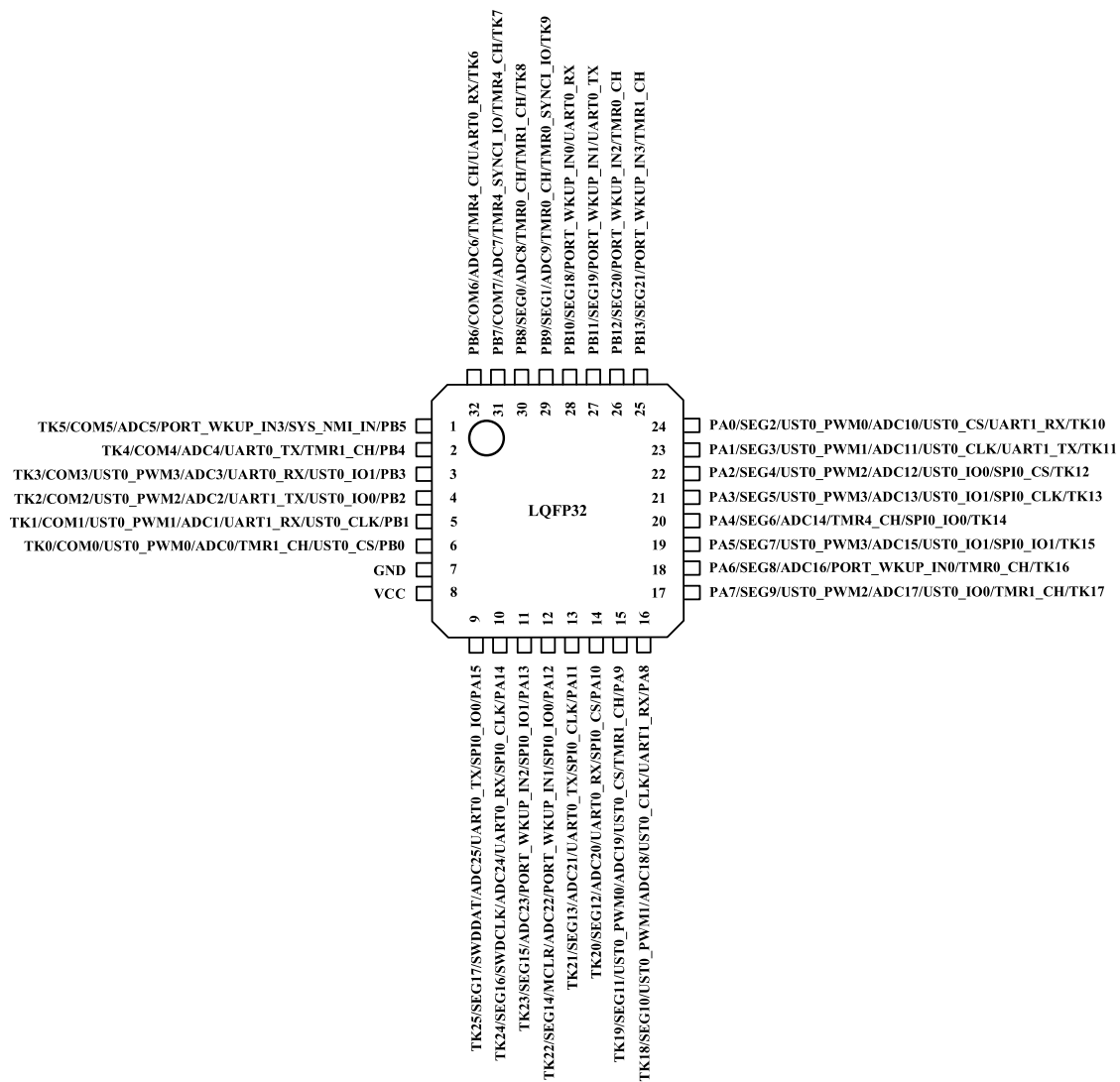
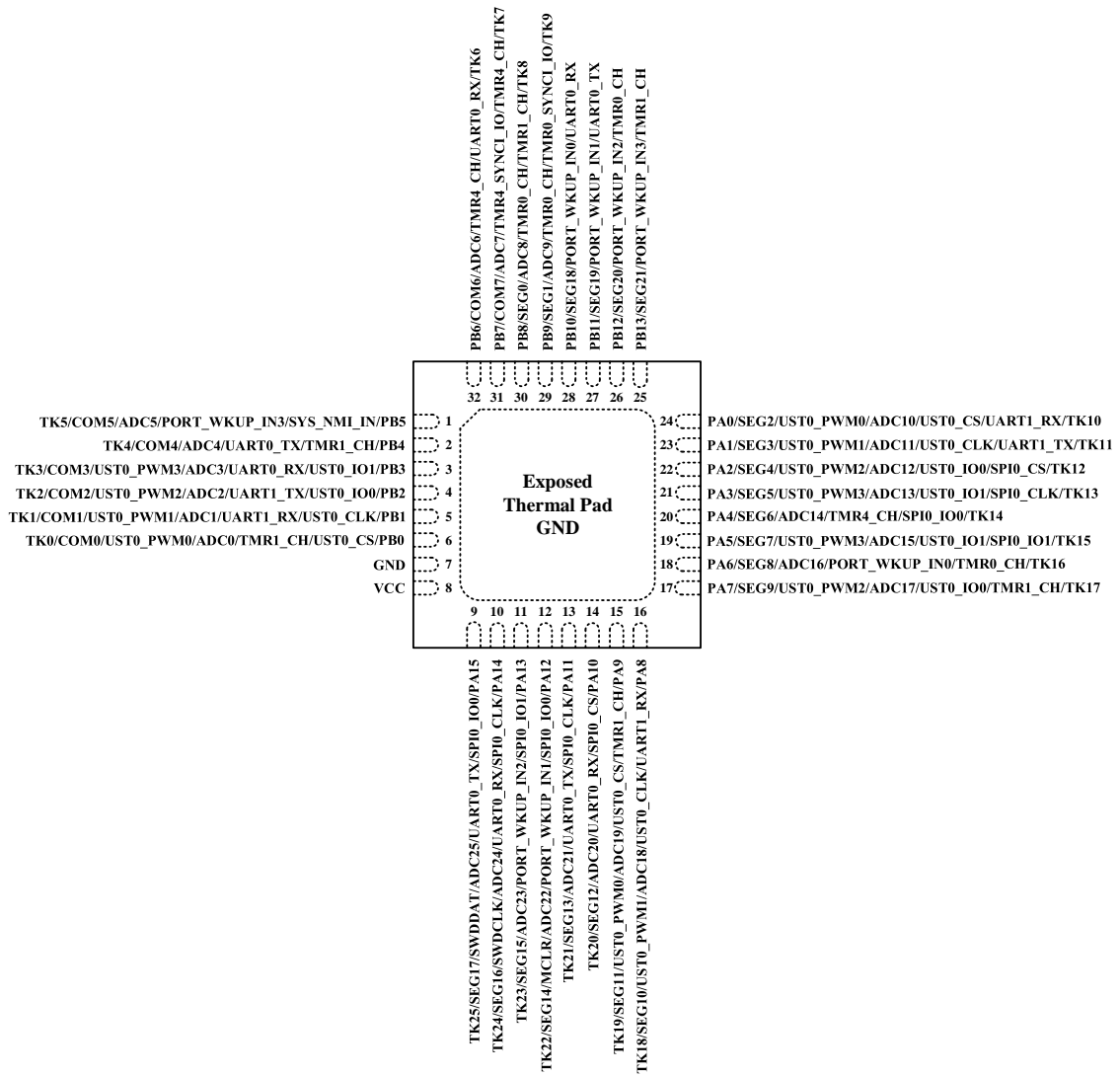


图 2-6 QFN32(CIU32F031J5NB)



2.2. 引脚功能描述

表 2-1 GPIO 引脚定义

LQFP32 QFN32	SOP28 SSOP28	SOP20 SSOP20	QFN20	SOP16	引脚	引脚 类型	I/O 电路 类型	附加功能	复用功能
1	7	3	1	14	PB5	I/O	COMIO	COM5 ADC5 TK5	PORT_WKUP3 SYS_NMI_IN
2	8	4	-	15	PB4	I/O	COMIO	COM4 ADC4 TK4	UART0_TX TMR1_CH
3	9	5	2	16	PB3	I/O	COMIO	COM3 ADC3 TK3	UART0_RX UST0_IO1 UST0_PWM3
4	10	6	3	1	PB2	I/O	COMIO	COM2 ADC2 TK2	UART1_TX UST0_IO0 UST0_PWM2
5	11	7	4	2	PB1	I/O	COMIO	COM1 ADC1 TK1	UART1_RX UST0_CLK UST0_PWM1
6	12	8	5	3	PB0	I/O	COMIO	COM0 ADC0 TK0	TMR1_CH UST0_CS UST0_PWM0
7	13	9	6	4	GND	-	-	-	-
8	14	10	7	5	VCC	-	-	-	-
9	15	11	8	6	PA15	I/O	I/OG1	SEG17 SWDDAT ADC25 UART1_TX TK25	UART0_TX SPI0_IO0
10	16	12	9	7	PA14	I/O	I/OG1	SEG16 SWDCLK ADC24 UART1_RX TK24	UART0_RX SPI0_CLK
11	17	13	10	8	PA13	I/O	I/OG1	SEG15 ADC23 TK23	PORT_WKUP_IN2 SPI0_IO1
12	18	14	11	9	PA12	I/O	I/OG1	SEG14 MCLR ADC22 TK22	PORT_WKUP_IN1 SPI0_IO0
13	19	15	-	-	PA11	I/O	I/OG1	SEG13 ADC21 TK21	UART0_TX SPI0_CLK
14	20	16	-	-	PA10	I/O	I/OG1	SEG12 ADC20	UART0_RX SPI0_CS

								TK20	
15	21	17	-	-	PA9	I/O	IOG1	SEG11 ADC19 TK19	UST0_CS UST0_PWM0 TMR1_CH
16	22	18	-	-	PA8	I/O	IOG1	SEG10 ADC18 TK18	UST0_CLK UST0_PWM1 UART1_RX
17	23	-	-	-	PA7	I/O	IOG1	SEG9 ADC17 TK17	UST0_IO0 UST0_PWM2 TMR1_CH
18	24	19	-	-	PA6	I/O	IOG1	SEG8 ADC16 TK16	PORT_WKUP_IN0 TMR0_CH
19	25	-	12	-	PA5	I/O	IOG1	SEG7 ADC15 TK15	UST0_IO1 UST0_PWM3 SPI0_IO1
20	26	-	13	10	PA4	I/O	IOG1	SEG6 ADC14 TK14	TMR4_CH SPI0_IO0
21	27	-	14	11	PA3	I/O	IOG1	SEG5 ADC13 TK13	UST0_IO1 UST0_PWM3 SPI0_CLK
22	28	-	15	12	PA2	I/O	IOG1	SEG4 ADC12 TK12	UST0_IO0 UST0_PWM2 SPI0_CS
23	1	-	16	-	PA1	I/O	IOG1	SEG3 ADC11 TK11	UST0_CLK UST0_PWM1 UART1_TX
24	2	-	17	-	PA0	I/O	IOG1	SEG2 ADC10 TK10	UST0_CS UST0_PWM0 UART1_RX
25	-	-	18	-	PB13	I/O	IOG1	SEG21	PORT_WKUP_IN3 TMR1_CH
26	-	-	-	-	PB12	I/O	IOG1	SEG20	PORT_WKUP_IN2 TMR0_CH
27	-	-	-	-	PB11	I/O	IOG1	SEG19	PORT_WKUP_IN1 UART0_TX
28	-	-	-	-	PB10	I/O	IOG1	SEG18	PORT_WKUP_IN0 UART0_RX
29	3	20	19	-	PB9	I/O	IOG1	SEG1 ADC9 TK9	TMR0_SY-I_IO TMR0_CH
30	4	-	-	-	PB8	I/O	IOG1	SEG0 ADC8 TK8	TMR0_CH TMR1_CH
31	5	1	20	-	PB7	I/O	COMIO	COM7 ADC7	TMR4_SYNCI_IO TMR4_CH

								TK7	
32	6	2	-	13	PB6	I/O	COMIO	COM6 ADC6 TK6	TMR4_CH UART0_RX

表 2-2 GPIO 复用功能 AF0~AF1

Port	AF0	AF1
PA0	UST0_CS	UART1_RX
PA1	UST0_CLK	UART1_TX
PA2	UST0_IO0	SPI0_CS
PA3	UST0_IO1	SPI0_CLK
PA4	TMR4_CH	SPI0_IO0
PA5	UST0_IO1	SPI0_IO1
PA6	PORT_WKUP_IN0	TMR0_CH
PA7	UST0_IO0	TMR1_CH
PA8	UST0_CLK	UART1_RX
PA9	UST0_CS	TMR1_CH
PA10	UART0_RX	SPI0_CS
PA11	UART0_TX	SPI0_CLK
PA12	PORT_WKUP_IN1	SPI0_IO0
PA13	PORT_WKUP_IN2	SPI0_IO1
PA14	UART0_RX	SPI0_CLK
PA15	UART0_TX	SPI0_IO0
PB0	TMR1_CH	UST0_CS
PB1	UART1_RX	UST0_CLK
PB2	UART1_TX	UST0_IO0
PB3	UART0_RX	UST0_IO1
PB4	UART0_TX	TMR1_CH
PB5	PORT_WKUP3	SYS_NMI_IN

PB6	TMR4_CH	UART0_RX
PB7	TMR4_SYNCI_IO	TMR4_CH
PB8	TMR0_CH	TMR1_CH
PB9	TMR0_CH	TMR0_SYNCI_IO
PB10	PORT_WKUP_IN0	UART0_RX
PB11	PORT_WKUP_IN1	UART0_TX
PB12	PORT_WKUP_IN2	TMR0_CH
PB13	PORT_WKUP_IN3	TMR1_CH

注:

1. UART0_RX/UART1_RX 单线工作时作收发引脚，双线工作时作接收引脚。
2. 当使用 SPI 的标准模式时，通信线 IO0 为 MOSI，IO1 为 MISO。

表 2-3 SSP 引脚选择

	SPI 功能	I2C 功能
SSP_CLK	SPI0_CLK	I2C_CLK
SSP_CS	SPI0_CS	-
SSP_IO0	SPI0_IO0	I2C_DATA
SSP_IO1	SPI0_IO1	-

表 2-4 UST 引脚选择

	UART 功能	SPI 功能	TIMER 功能 PWM 输出
UST0_CS	-	CS	PWM0
UST0_CLK	-	CLK	PWM1
UST0_IO0	TX	IO0	PWM2
UST0_IO1	RX	IO1	PWM3

3. 系统及存储器架构

CIU32F011、CIU32F031 器件是基于 ARM Cortex M0 处理器的 32 位通用微控制器存储器芯片。采用了哈佛结构，具有低中断延迟时间和低成本调试特性，而且高集成度和增强的特性使这颗处理器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 4GB 的存储空间，充分保证了系统的灵活性和可扩展性。

3.1. 系统架构

CIU32F011、CIU32F031 器件采用 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接下来将进行说明。

CIU32F011、CIU32F031 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

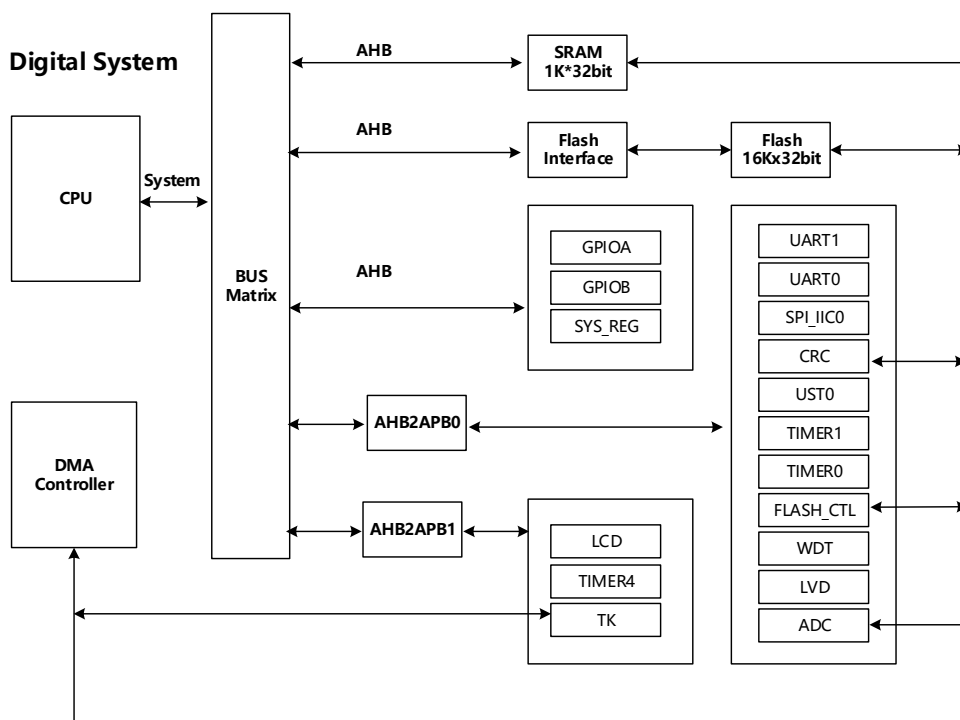


图 3-1 总线系统架构图

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 3-1 DMA 优先级

模块	优先级	备注
CPU	0	↑ 最高 ↓ 最低
TK	1	
ADC	2	
CRC	3	
FLASH	4	

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。

3.2. 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 IO 端口都在同一个线性的 4GB 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。CIU32F011、CIU32F031 器件的存储器映射表显示了 CIU32F011、CIU32F031 器件的存储器映射，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。

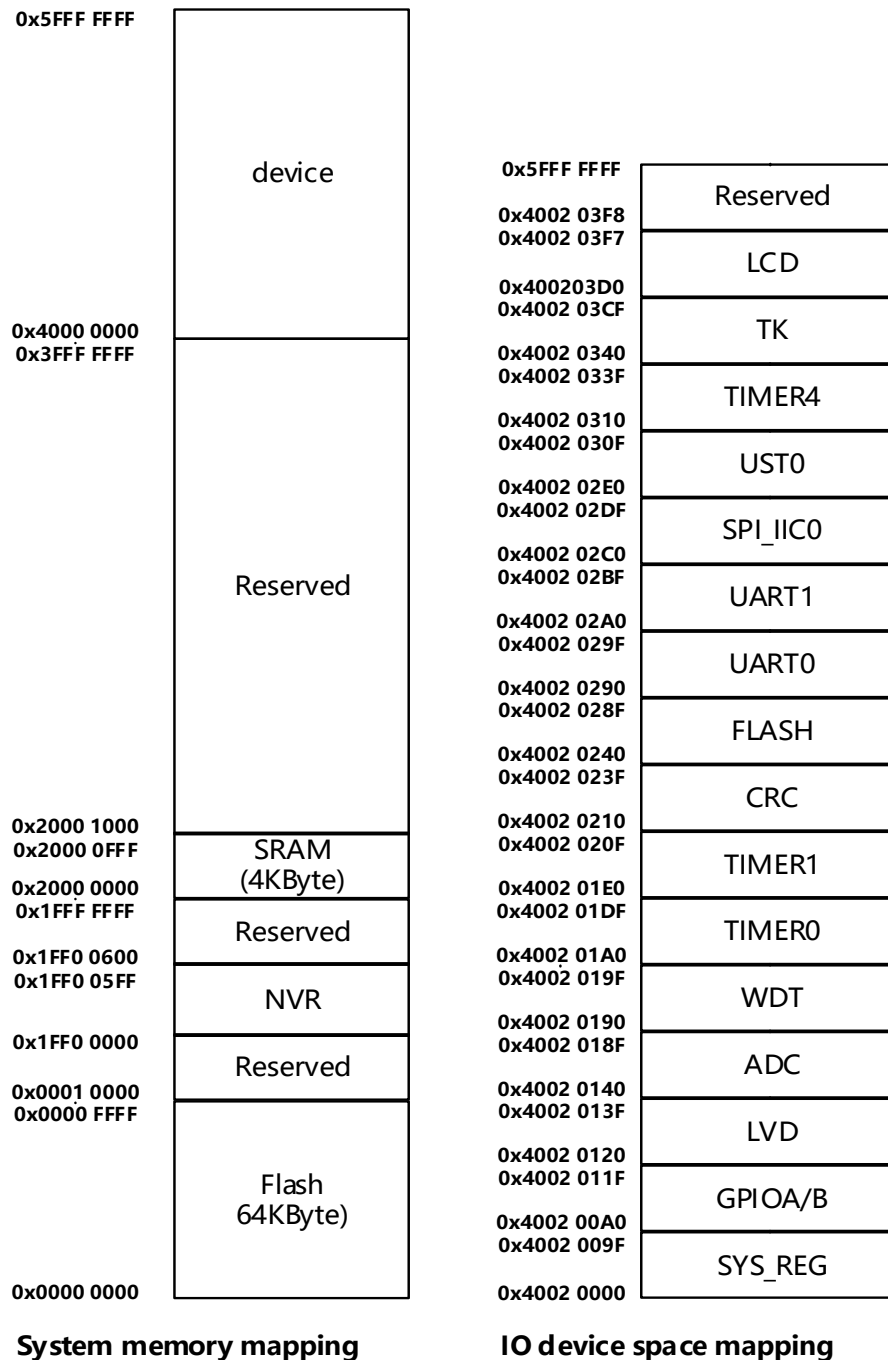


图 3-2 存储器映射表

3.2.1. 片上 SRAM

芯片内置 4K 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 FLASH, CRC, ADC, TK.

3.2.2. 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)

副闪存存储块, 也叫信息块, 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) - 闪存接口基于 AHB 协议执行指令和数据存取。

3.2.3. 引导配置

芯片复位后，通过客户自己在副闪存的配置，选择 SWDCLK 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。

3.2.4. 位带操作

为了减少“读-改-写”操作的次数，32 位 RISC 处理器提供了一个可以执行单原子比特操作的位带功能。存储器映射包含了两个支持位带操作的区域。其中一个是 SRAM 区的最低 1MB 范围，第二个是片内外设区的最低 1MB 范围。这两个区域中的地址除了普通应用外，还有自己的“位带别名区”。位带别名区把每个比特扩展成一个 32 位的字。当用户访问位带别名区时，就可以达到访问原始比特的目的。

下面的公式表明了位带别名区中的每个字如何对应位带区的相应比特或目标比特。

$$\text{bit_word_addr} = \text{bit_band_base} + (\text{byte_offset} \times 32) + (\text{bit_number} \times 4)$$

其中

bit_word_addr 指的是位带区目标比特对应位在位带别名区的地址；

bit_band_base 指的是位带别名区的起始地址；

byte_offset 指的是位带区目标比特所在的字节的字节地址偏移量；

bit_number 指的是目标比特在对应字节中的位置(0-7)。

例如，要想访问 0x2000_0200 地址的第 7 位，可访问的位带别名区地址是：

$$\text{bit_word_addr} = 0x2200_0000 + (0x200 * 32) + (7 * 4) = 0x2200_401C$$

如果对 0x2200_401C 进行写操作，那么 0x2000_0200 的第 7 位将会相应变化；如果对 0x2200_401C 进行读操作，那么视 0x2000_0200 的第 7 位状态而返回 0x01 或 0x00。

4. 嵌入式闪存 (FLASH)

4.1. 模块介绍

CIU32F011、CIU32F031 集成了嵌入式 FLASH 控制模块，该模块控制 FLASH 的擦除、编程以及读取数据。上电时会从 FLASH 中读取相关数据进行校验以及初始化配置，保证芯片程序在正确且安全的情况下运行。

4.2. 功能特点

支持高达 64K 主闪存空间的 FLASH

存储器结构

- 主闪存空间 64K 字节
- 副闪存空间 4.5K 字节

指出对闪存空间的擦写、编程和读操作

支持对闪存空间访问限制和擦写保护

支持低功耗模式

4.3. 功能说明

4.3.1. 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 64 页（每页 1K 字节、2 个扇区）分块，以页为单位设置写保护（参见存储保护相关内容），擦写以扇区为单位。

表 4-1 闪存结构

闪存空间	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 62	0x0000_F800 – 0x0000_FBFF	1K
	Page 63	0x0000_FC00 – 0x0000_FFFF	1K
副闪存空间	Sector 0	0x1FF0_0000 – 0x1FF0_01FF	512
	Sector 1	0x1FF0_0200 – 0x1FF0_03FF	512
	512
	Sector 8	0x1FF0_1000 – 0x1FF0_11FF	512

注：当主闪存空间 64KB 不够存放用户程序时，可把副闪存空间的扇区 0 至扇区 7 扩展为程序存放空间，即支持最大 68KB 的程序存放空间。

4.3.2. 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。

当 NVR8 用户配置区经过自定义的保护配置后，SWD 连接时会对 FLASH 的代码数据执行保护机制。

注：FLASH 运行在 24MHz 工作频率，当系统时钟超过 30MHz 时，需要配置 TIMER_REG0 的 RC 参数，增加时钟周期数再把 FLASH 接口的数据写到寄存器。

4.3.3. 闪存擦除和烧写操作

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_REG1)，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。

烧写操作相关寄存器

时序寄存器 1 : TIME_REG1

密码寄存器 : NVR_PASSWORD/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

擦除操作相关寄存器:

擦除控制寄存器: ERASE_CTRL

注: 需要注意的是，FLASH 在擦除/烧写的同时不可以从 FLASH 取数据，所以 FLASH 在擦除/烧写过程中会让总线停顿，直到完成后才能继续运行。

4.4. 模块框图

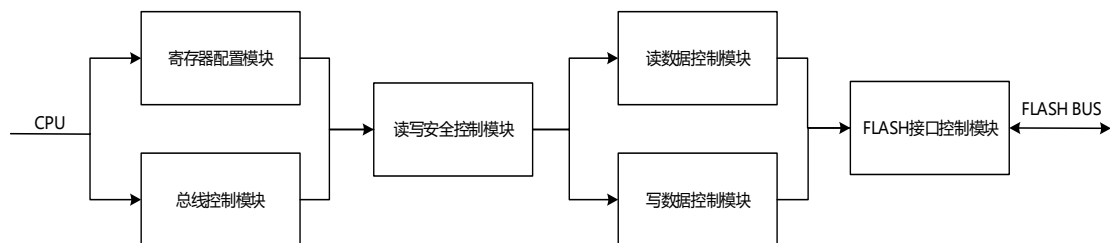


图 4-1 FLASH 模块架构图

4.5. 寄存器描述

4.5.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
CTRL0	0x0240	32'h1000	FLASH 控制寄存器 0
KST	0x0244	32'h0	触发寄存器
DONE	0x0248	32'h1c53	完成状态寄存器
PROG_ADDR	0x024C	32'h80000000	编程地址寄存器
PROG_DATA	0x0250	32'h0	编程数据寄存器
ERASE_CTRL	0x0254	32'h0	擦除控制寄存器
TIME_REG0	0x0258	32'hdddd0	时序控制寄存器 0
TIME_REG1	0x025C	32'h3e80c	时序控制寄存器 1
NVR_PASSWORD	0x0260	32'h0	NVR 密钥寄存器
MAIN_PASSWORD	0x0264	32'h0	MAIN 密钥寄存器
CRC_ADDR	0x0268	32'h0	CRC 校验地址
CRC_LEN	0x026C	32'h0	CRC 校验长度
CRC_OUT	0x0270	32'h5a5a55aa	CRC 校验结果
CFG_SECTOR	0x0280	32'h40	MAIN 区域扇区数量寄存器

4.5.2. 寄存器详细说明

4.5.2.1. 控制寄存器 (CTRL0)

Width	Name	Reset	Property	Description
31:17	Reserved	-	-	-
16	PROG_CLK_SEL	1'b0	RW	FLASH 烧写时钟源选择, 推荐使用 RC 时钟 0: HIRC 时钟 2 分频 1: 保留
15:13	Reserved	-	-	-
12	PROG_RAM_CRC_EN	1'b1	RW	把 RAM 的数据搬移到 FLASH 时在最后添加 CRC 值 0: 不使能 1: 使能
11	LVD_PROG_STOP	1'b0	RW	在 LVD 断电时, 是否允许打断 FLASH 编程和擦除, 如果断电时, 需要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: 不允许打断 1: 允许打断
10	PROG_RAM_MODE	1'b0	RW	直接把 RAM 的数据写到 FLASH, 根据 PROG_RAM_CRC_EN 为 1 时在 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: FLASH CRC 模式 1: RAM to FLASH 模式
9:0	Reserved	-	-	-

4.5.2.2. 触发寄存器 (KST)

Width	Name	Reset	Property	Description
31:27	Reserved	-	-	-
26	CRC_KST_EN	1'b0	WO	FLASH 的 CRC 触发使能 0: 不使能 1: 使能
25:11	Reserved	-	-	-
10	CRC_KST	1'b0	WO	FLASH 的 CRC 校验触发 与第 26 位同时写 '1' 触发
9:0	Reserved	-	-	-

4.5.2.3. 状态寄存器 (DONE)

Width	Name	Reset	Property	Description
31:13	Reserved	-	-	-
12	CHIP_ERASE_OK_FLAG	1'b1	RO	Chip 擦除结果 0: 擦除失败 1: 擦除成功
11	PROG_OK_FLAG	1'b1	RO	编程结果, 写入后读出数据检查是否与写入数据一致 0: 烧写失败 1: 烧写成功

10	CRC_DONE	1'b1	RO	FLASH 的 CRC 完成标志 0: 进行中 1: 空闲状态
9:7	Reserved	-	-	-
6	PROG_DONE	1'b1	RO	编程结束标志 0: 进行中 1: 空闲状态
5:2	Reserved	-	-	-
1	CHIP_ERASE_DONE	1'b1	RO	Main 区域全擦除标志 0: 正在运行 1: 空闲状态
0	SECTOR_ERASE_DONE	1'b1	RO	扇区擦除标志 0: 正在运行 1: 空闲状态

4.5.2.4. 编程地址寄存器 (PROG_ADDR)

Width	Name	Reset	Property	Description
31:30	Reserved	-	-	-
29	PROG_NVR_SEL	1'b0	RW	编程的地址选择 MAIN/NVR 0: MAIN 区域 1: NVR 区域
28:15	Reserved	-	-	-
14:0	ADDR	15'h0	RW	FLASH 编程的逻辑地址 (即单位是字节), 编程需要以 word 为单位 (即地址低两位需为 0)。 注: 在 CTRL0 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。

4.5.2.5. 编程数据寄存器 (PROG_DATA)

Width	Name	Reset	Property	Description
31:0	DATA	32'h0	RW	PROG_RAM_MODE 为 0 时: 配置好 PROG_ADDR 和 PASSWORD 后, 配置 PROG_DATA 启动编程; PROG_RAM_MODE 为 1 时: 该寄存器显示当前往 FLASH 编程的数据。

4.5.2.6. 擦除控制寄存器 (ERASE_CTRL)

Width	Name	Reset	Property	Description
31	CHIP_ERASE_KST	1'b0	WO	Chip 擦除的触发, 写 "1" 触发, 需要先配置密码
30	SECTOR_ERASE_KST	1'b0	WO	扇区擦除的触发, 写 "1" 触发, 需要先配置密码
29	NVR_SECTOR_EN	1'b0	RW	NVR 的扇区使能位 0: MAIN 区域 1: NVR 区域
28:7	Reserved	-	-	-
6:0	SECTOR_ERASE_ADDR	7'h0	RW	擦除的扇区选择

4.5.2.7. 时序 0 寄存器 (TIME_REG0)

Width	Name	Reset	Property	Description
31:20	Reserved	-	-	-
19:16	PGH	4'hd	RW	WEB 低到 PROG2 高 保持时间最小为 15ns
15:12	ADS	4'hd	RW	字节/地址/数据设置的最小时间是 15ns
11:8	ADH	4'hd	RW	字节/地址/数据保持时间最小为 15ns
7:4	RW	4'hd	RW	编程/擦除后的下一个操作的延迟时间为 100ns
3:0	RC	4'h0	RW	读取周期最小时间为 25/30ns

4.5.2.8. 时序 1 寄存器 (TIME_REG1)

Width	Name	Reset	Property	Description
31:19	Reserved	-	-	-
18:8	MS_CNT_SET	11'h3e8	RO	1ms 的时间配置值, 以 1us 为单位
7:0	US_CNT_SET	8'hc	RW	1us 的时间配置值, 时钟频率默认是 24MHz 的 2 分频

4.5.2.9. 密码 0 寄存器 (NVR_PASSWORD)

Width	Name	Reset	Property	Description
31:0	NVR_PASS_WORD	32'h0	RW	NVR 区域密码, 配置后才能对 NVR 区域编程/擦除。 NVR0~8 密码为: 0x20150931

4.5.2.10. 密码 1 寄存器 (MAIN_PASSWORD)

Width	Name	Reset	Property	Description
31:0	MAIN_PASS_WORD	32'h0	RW	MAIN 区域密码, 配置后才能对 MAIN 区域编程/擦除, 密码为: 0x20170230

4.5.2.11. CRC 地址配置寄存器 (CRC_ADDR)

Width	Name	Reset	Property	Description
31:30	Reserved	-	-	-
29	NVR_SEL	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	PROG_RAM_MODE 为 0 时: 配置成 FLASH 中 CRC DMA 的开始地址, 且是物理地址。 PROG_RAM_MODE 为 1 时: 器配置成 RAM 数据的起始地址(RAM 起始地址为 0x2000_0000) 注意: 地址需要按 word 地址配置
1:0	Reserved	-	-	-

4.5.2.12. CRC 长度配置寄存器 (CRC_LEN)

Width	Name	Reset	Property	Description
31:17	Reserved	-	-	-
16:2	DMA_LEN	15'h0	RW	CRC DMA 长度

				PROG_RAM_MODE 为 1 时表示 SRAM 中数据的长度。 注意: 长度需要按 word 长度配置
1:0	Reserved	-	-	-

4.5.2.13. CRC 结果配置寄存器 (CRC_OUT)

Width	Name	Reset	Property	Description
31:0	CRC_OUT	32'h5a5a 55aa	RO	CRC 的结果, 多项式 CRC-32 如下: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

4.5.2.14. 主扇区信息寄存器 (CFG_SECTOR)

Name	Width	Reset	Property	Description
MAIN_SECTOR_NUM	31:0	32'h40	RO	MAIN 区域的扇区数

4.5.3. NVR8 闪存用户配置区

副闪存 NVR8 最后 64byte 为用户配置区, 逻辑地址区间: 0x1FF0_11C0-0x1FF0_11FF, 该区域的配置数据需要在程序烧写阶段烧写, 芯片正常上电后这些信息为只读。

4.5.3.1. 系统配置 0 (0x1FF0_11C0)

Width	Name	Reset	Property	Description
28:11	Reserved	-	-	-
10	CODE_PROTECT_DIS	1'b1	RO	代码保护使能 0: 使能 1: 不使能 (使能后 SWD 连接时, FLASH 的读、写、擦权限都会关闭, 且部分模块寄存器的读写权限也关闭)
9:8	Reserved	-	-	-
7	NVR7_WRITE_EN	1'b1	RO	NVR7 擦写权限 0: 关闭 1: 打开
6	NVR6_WRITE_EN	1'b1	RO	NVR6 擦写权限 0: 关闭 1: 打开
5	NVR5_WRITE_EN	1'b1	RO	NVR5 擦写权限 0: 关闭 1: 打开
4	NVR4_WRITE_EN	1'b1	RO	NVR4 擦写权限 0: 关闭 1: 打开
3	NVR3_WRITE_EN	1'b1	RO	NVR3 擦写权限 0: 关闭 1: 打开
2	NVR2_WRITE_EN	1'b1	RO	NVR2 擦写权限 0: 关闭 1: 打开

1	NVR1_WRITE_EN	1'b1	RO	NVR1 擦写权限 0: 关闭 1: 打开
0	NVR0_WRITE_EN	1'b1	RO	NVR0 擦写权限 0: 关闭 1: 打开

4.5.3.2. 系统配置 1 (0x1FF0_11C4)

Width	Name	Reset	Property	Description
31:0	MAIN_WRITE_EN	32'hfffffff	RO	MAIN 前 32K 擦写权限, 每 1bit 代表 1Kbyte 0: 关闭 1: 打开

4.5.3.3. 系统配置 2 (0x1FF0_11C8)

Width	Name	Reset	Property	Description
31:0	MAIN_WRITE_EN1	32'hfffffff	RO	MAIN 后 32K 擦写权限, 每 1bit 代表 1Kbyte 0: 关闭 1: 打开

4.5.3.4. 系统配置 5 (0x1FF0_11D4)

Width	Name	Reset	Property	Description
31	CODE_CHECK_EN	1'b1	RO	CODE CRC 校验使能 0: 不使能 1: 使能
30:16	Reserved	-	-	-
15:0	CODE_LEN	16'hffff	RO	需要进行 CRC 校验的 CODE 长度, 以 word 为单位, 超过该长度的区域为 DATA 区。

4.5.3.5. 功能配置 0 (0x1FF0_11E0)

Width	Name	Reset	Property	Description
31:15	Reserved	-	-	-
14	SWD_EN	1'b1	RO	SWD 接口是否使能 0: 关闭 1: 使能 该位和 SYS_CON1[8]同时为 0 时才关闭
13	Reserved	-	-	-
12	MCLR_EN	1'b1	RO	MCLR (复位) 功能使能 0: MCLR 功能有效 1: MCLR 功能无效
11	Reserved	1'b1	RO	-
10:9	SWD_PULL_MODE	2'b11	RO	SWD 接口的 TCK 复位状态下上拉或下拉 10: 下拉 11: 上拉 Others: 上下拉关闭, SWD 也关闭
8:0	Reserved	-	-	-

4.5.4. NVR9 闪存芯片信息区

4.5.4.1. 芯片信息 0 (0x1FF0_1200)

Width	Name	Reset	Property	Description
31:0	UID0	32'hfffffff	RO	UID 信息 0

4.5.4.2. 芯片信息 1 (0x1FF0_1204)

Width	Name	Reset	Property	Description
31:0	UID1	32'hfffffff	RO	UID 信息 1

4.5.4.3. 芯片信息 2 (0x1FF0_1208)

Width	Name	Reset	Property	Description
31:0	UID2	32'hfffffff	RO	UID 信息 2

4.5.4.4. 芯片信息 6 (0x1FF0_1218)

Width	Name	Reset	Property	Description
31:0	CUSTOMER_ID	32'b0	RO	客户信息

5. 中断和事件 (INT/EVT)

5.1. 嵌套向量中断控制器

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。

5.2. 系统滴答(SysTick)校准值寄存器

本芯片支持系统滴答计时。

5.3. 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态所带来的开销。

表 5-1 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 5-2 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	-	0x0000_0044
IRQ2	18	TKSCDN_IRQn	0x0000_0048
IRQ3	19	-	0x0000_004C
IRQ4	20	UART0_IRQn	0x0000_0050
IRQ5	21	UART1_IRQn	0x0000_0054
IRQ6	22	UST0_IRQn	0x0000_0058
IRQ7	23	SPI0_IRQn	0x0000_005C

IRQ8	24	GPIOA_IRQn	0x0000_0060
IRQ9	25	GPIOB_IRQn	0x0000_0064
IRQ10	26	WKPND_IRQn	0x0000_0068
IRQ11	27	TIMER0_IRQn	0x0000_006C
IRQ12	28	TIMER1_IRQn	0x0000_0070
IRQ13	29	TIMER4_IRQn	0x0000_0074
IRQ14	30	ADC_IRQn	0x0000_0078
IRQ15	31	CRC_DMA_IRQn	0x0000_007C
IRQ16	32	WDT_IRQn	0x0000_0080
IRQ17	33	LCD_IRQn	0x0000_0084
IRQ18	34	-	0x0000_0088
IRQ19	35	-	0x0000_008C
IRQ20	36	-	0x0000_0090
IRQ21	37	-	0x0000_0094
IRQ22	38	-	0x0000_0098
IRQ23	39	-	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

5.4. 外部中断/事件控制器(EXTI)

外部中断/事件控制器包含 26 个产生中断/事件触发的边沿检测电路，每条输入线可以独立地配置触发事件类型（上升沿或下降沿或者双边沿都触发）。每条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，可通过对挂起的寄存器对应位写“1”清除中断请求。

5.4.1. 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 26 个软件中断/事件请求

支持上升沿、下降沿和双边沿 3 种触发事件类型

5.4.2. 唤醒事件管理

CIU32F011、CIU32F031 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。

6. 循环冗余校验计算单元 (CRC)

6.1. 模块介绍

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。

6.2. 功能特点

支持 16/32 位不同长度的多项式

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

6.3. 功能说明

该模块用于计算 SRAM 中指定数据段的 CRC 校验值, 软件配置计算初始值、校验多项式、起始地址、数据长度, 启动 CRC 计算后等待硬件完成标志有效时读取 CRC_OUT 寄存器可获得 CRC 校验值。

6.4. 模块框图

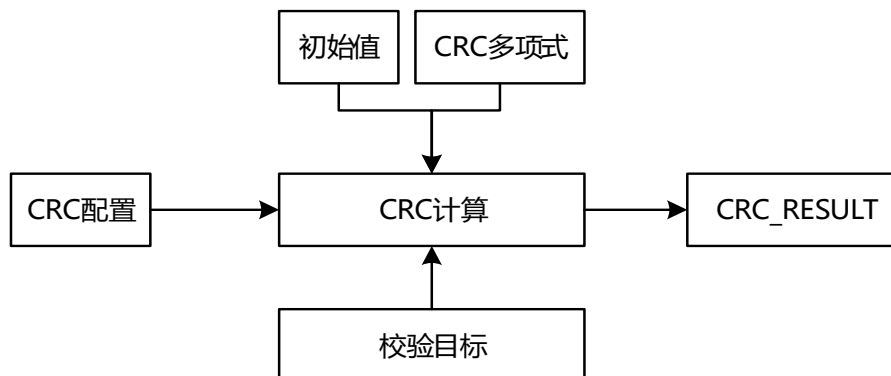


图 6-1 CRC 计算单元框图

6.5. 时钟与复位

6.5.1. 时钟介绍

该模块时钟源为系统时钟, 低功耗模式下到达模块的时钟会被自动停止。

6.5.2. 复位介绍

该模块的复位源有两个, 分别是系统复位和软件复位, 软件复位可通过配置系统寄存器触发。

6.6. 寄存器描述

6.6.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
CRC_CFG	0x0210	32'h100	CRC 控制寄存器

CRC_INIT	0x0214	32'hfffffff	CRC 初始值寄存器
CRC_INV	0x0218	32'hfffffff	CRC 输出结果取反控制寄存器
CRC_POLY	0x021C	32'hedb88320	CRC 多项式配置寄存器
CRC_KST	0x0220	32'h0	CRC 状态清除寄存器
CRC_STA	0x0224	32'h0	CRC 状态标志寄存器
CRC_ADDR	0x0228	32'h0	CRC DMA 起始地址寄存器
CRC_LEN	0x022C	32'h0	CRC 计算长度寄存器
CRC_OUT	0x0230	32'h0	CRC 输出结果寄存器

6.6.2. 寄存器详细说明

6.6.2.1. 控制配置寄存器 (CRC_CFG)

Width	Name	Reset	Property	Description
31:9	Reserved	-	-	-
8	POLY_WIDTH	1'b1	RW	多项式宽度 0: 16 位 1: 32 位 注: POLY_WIDTH=1 时, 只支持 BIT_ORDER_EN=0。
7:2	Reserved	-	-	-
1	BIT_ORDER	1'b0	RW	输入数据顺序 0: 高位到低位(7:0) 1: 低位到高位(0:7)
0	IE	1'b0	RW	CRC 中断使能 0: 关闭 1: 使能

6.6.2.2. 初始值配置寄存器 (CRC_INIT)

Width	Name	Reset	Property	Description
31:0	INIT_VALUE	32'hfffffff	RW	CRC 初始值

6.6.2.3. 取反寄存器 (CRC_INV)

Width	Name	Reset	Property	Description
31:0	INV_VALUE	32'hfffffff	RW	CRC 输出结果取反控制 0: 不取反 1: 取反

6.6.2.4. 多项式配置寄存器 (CRC_POLY)

Width	Name	Reset	Property	Description
31:0	POLY_VALUE	32'hedb88320	RW	CRC 多项式值, 默认为 32 位多项式 注: 写入的值是多项式二进制表示值的反向值, 比如多项式 $x^{16} + x^{15} + x^2 + 1$ 的多项式二进制表示是 0x8005, 则高低位反转为 0xa001 写入 CRC_POLY 寄存器。

6.6.2.5. 触发寄存器 (CRC_KST)

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

31:1	Reserved	-	-	-
0	PEND_CLR	1'b0	RC	清除 CRC 状态位, 写 1 清除

6.6.2.6. 状态寄存器 (CRC_STA)

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

6.6.2.7. 地址寄存器 (CRC_ADDR)

Width	Name	Reset	Property	Description
31:14	Reserved	-	-	-
13:0	DMA_ADDR	14'b0	RW	CRC DMA 起始地址(word 对齐) 注意: 地址 0 指向 0x2000_0000

6.6.2.8. 长度寄存器 (CRC_LEN)

Width	Name	Reset	Property	Description
31:14	Reserved	-	-	-
13:0	DMA_LEN	14'h0	RW	CRC DMA 字节长度

6.6.2.9. 输出寄存器 (CRC_OUT)

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

7. 电源管理 (POWER MANAGEMENT)

7.1. 电源

芯片的工作电压为 2.5V~5.5V。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]，同时也有低功耗参考电流寄存器 PMUCON0[16:14]。

7.1.1. 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。

7.2. 电源管理器

7.2.1. 上电复位(POR) 和掉电复位(PDR)

CIU32F011、CIU32F031 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 2.2V 时系统能正常工作。当 VDD 低于指定的限位电压 VPOR(1.2V)/VPDR(1.1V)时，系统保持为复位状态，而无需外部复位电路。

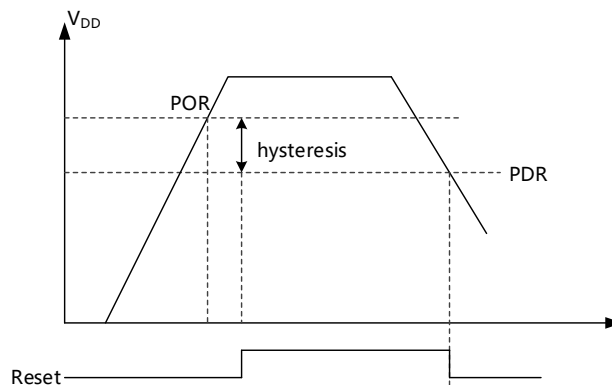


图 7-1 上下电复位波形图

7.2.2. 可编程电压监测器(PVD)

CIU32F011、CIU32F031 内部集成一个外部供电 VCC 电压检测器，检测电压均阈值可选。当系统监测到 VCC 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 来控制。

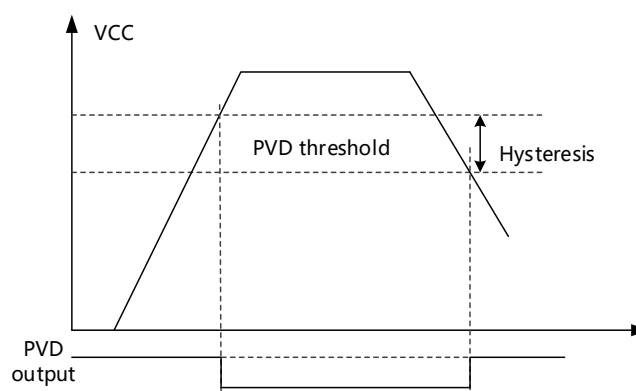


图 7-2 掉电检测波形图

7.3. 电源控制寄存器

7.3.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
LVDCON	0x0120	32'he51	LVD 控制寄存器。(需要写 SYS_KEY 寄存器才能操作 LVDCON 寄存器)
LVDCON1	0x0124	32'h0	LVD 控制寄存器 1。(需要写 SYS_KEY 寄存器才能操作 LVDCON1 寄存器)

7.3.2. 寄存器详细说明

7.3.2.1. LVD 控制寄存器 (LVDCON)

Width	Name	Reset	Property	Description
31	VDD_PEND	1'b0	RC	VDD 电压低于设定阈值触发 PEND, 写 0 清除
30	VCC_PEND	1'b0	RC	VCC 电压低于设定阈值触发 PEND, 写 0 清除
29	VDD_LVD_ANA_OUTPUT_DIS	1'b0	RW	VDD_LVD 模拟信号输出使能 0: 使能 1: 不使能 先使能 VDD_LVD_EN, 再使能 VDD_LVD_ANA_OUTPUT_DIS
28:22	VDD_DBS_LO_LIMIT	7'h0	RW	对 PMU_LVD_VDD 信号的低电平毛刺滤波宽度, 单位为 lvd_dbs_clk 时钟周期。
21:15	VDD_DBS_HI_LIMIT	7'h0	RW	对 PMU_LVD_VDD 信号的高电平毛刺滤波宽度, 单位为 lvd_dbs_clk 时钟周期
14	VCC_SYNC_DIS	1'b0	RW	VCC_LVD 在中断模式下唤醒 CPU 的信号是否需要先经过系统时钟同步 0: 同步 1: 不同步
13	VDD_SHIELD_DEBOUNCE_EN	1'b0	RW	屏蔽对 PMU_VDD_LVD 信号的滤波处理 0: 不屏蔽 1: 屏蔽
12	VCC_SHIELD_DEBOUNCE_EN	1'b0	RW	屏蔽对 PMU_VCC_LVD 信号的滤波处理 0: 不屏蔽 1: 屏蔽
11	LVD_DIG_EN	1'b1	RW	LVD 数字开关, 使能后才会对 LVD 事件产生复位或者中断 0: 不使能 1: 使能
10	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	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	VCC_LVD_ANA_OUTPUT	1'b0	RW	VCC_LVD 模拟信号输出使能

	PUT_DIS			0: 使能 1: 不使能 先使能 VCC_LVD_EN, 再使能 VCC_LVD_ANA_OUTPUT_DIS
7:5	VDD_LVD_SET	3'b010	RW	VDD_LVD 电压检测设置(单位 V) b000: 1.15 b001: 1.20 b010: 1.25 b011: 1.30 b100: 1.35 b101: 1.40 b110: 1.45 b111: 1.50
4	VDD_LVD_EN	1'b1	RW	VDD_LVD 模拟开关, 使能后才会检测 VDD_LVD 电压 0: 不使能 1: 使能
3:1	VCC_LVD_SET	3'b000	RW	VCC_LVD 电压检测设置(单位 V): b000: 2.10 b001: 2.20 b010: 2.40 b011: 2.50 b100: 2.70 b101: 3.00 b110: 3.60 b111: 4.20
0	VCC_LVD_EN	1'b1	RW	VCC_LVD 模拟开关, 使能后才会检测 VCC_LVD 电压 0: 不使能 1: 使能

7.3.2.2. LVD 控制寄存器 (LVDCON1)

Width	Name	Reset	Property	Description
31:15	Reserved	-	-	-
14:8	VCC_DBS_LO_LIMIT	7'h0	RW	对 PMU_LVD_VCC 信号的低电平毛刺滤波宽度, 单位为 lvd_dbs_clk 时钟周期。
7	Reserved	-	-	-
6:0	VCC_DBS_HI_LIMIT	7'h0	RW	对 PMU_LVD_VCC 信号的高电平毛刺滤波宽度, 单位为 lvd_dbs_clk 时钟周期

8. 低功耗 (LOW POWER)

8.1. 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 256KHz 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 8-1 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 HIRC 时钟源关闭(设置 LPCON[3])	可选: 片内 HIRC 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 256K 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

8.2. 进入低功耗

进入低功耗模式(Sleep)步骤:

Step1: 关闭低功耗下不需要工作的模拟模块

Step2: 系统时钟切换到 LIRC_256K, 并且关闭除 LIRC_256K 以外的时钟源

Step3: 配置 PMU_LPDOS 使其电压档位比 PMU_HPLDOS 电压档位高

Step4: 把 PMU_HPXCPC, PMU_HPPDI, PMU_HPPDLI 配置为 0

Step5 (可选): 配置 LPCON[7]为 1, 并且配置 PMUBK 寄存器设置低功耗下 LPLDO 电压

Step6: 配置 PMU_HP2I 为 0 (如果执行了 step5, 则无需执行 step6)

Step7: 配置空闲的 IO 为模拟模式

Step8: 进入 Sleep 前的准备工作: 包括配置唤醒源, 初始化唤醒中断服务函数, 关闭看门狗(可选), 使能进入 Sleep 时自动关闭 256K 内部 RC 振荡器 (可选), 使能 SLEEP_GOON_EN 唤醒时不复位 (可选)

Step9: 配置 LP_CON0[0]进入 Sleep

8.3. 低功耗唤醒

支持多种唤醒方式

端口唤醒

总共有 4 个 IO 唤醒源 (由 WKUP_CON[3:0]控制使能)

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。

IO 唤醒初始化步骤：配置 IO 模式->配置唤醒边沿->清除唤醒标志位->中断初始化使能（根据需要）->使能唤醒位->清除唤醒标志位->配置 LP_CON 进入低功耗模式。后续只需要在进入低功耗模式之前，切换 IO 模式->清除唤醒标志位->进入低功耗模式。

注意：

如果选择了上升沿/下降沿唤醒，而此时 IO 为高电平/低电平，则唤醒标志位马上会置 1。使能后 IO 保持为高电平/低电平不会重复触发，唤醒标志位保持为 1，上升沿/下降沿才会触发。

只要唤醒标志位为 1，则无法进入低功耗模式。

对于唤醒 IO 的电平翻转时间无法确定的应用场景，如果在主程序中使能 wkup_en/int_en，在唤醒中断函数中关闭 wkup_en/int_en，会存在以下风险：使能 wkup_en/int_en 时马上触发 wkup_pend 导致进入中断，然后在中断中关闭 wkup_en/int_en，退出中断后进入低功耗模式，导致无法唤醒/唤醒后不进入唤醒中断。因此唤醒中断要慎重使用。

内部源唤醒

支持看门狗唤醒，TIMER4 中断唤醒，LVD 唤醒，触摸按键唤醒，flash_wkup(STOP 模式下 prog_ram_done)，LVDVCC_WKUP。

9. 复位和时钟系统 (RESET/CLOCK)

9.1. 引脚复位(MCLR)功能

默认状态下 CIU32F011、CIU32F031 的 MCLR 复位功能关闭，PA12 为 MCLR 引脚，使能复位功能时该引脚默认使能内部上拉功能，在 PA12 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 中用户自定义位来打开/关闭 MCLR 功能，关闭后 PA12 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。

9.2. 复位

CIU32F011、CIU32F031 支持系统复位、电源复位和主复位。

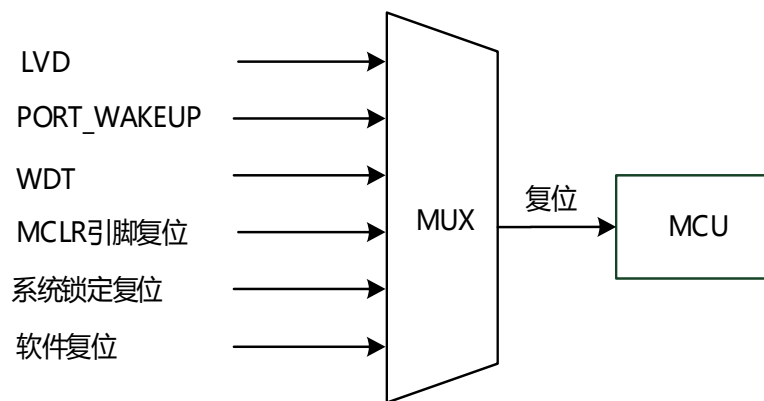


图 9-1 复位源框图

9.2.1. 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
- 系统锁定复位

9.2.2. 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

9.2.3. 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。

9.3. 时钟

9.3.1. 模块框图

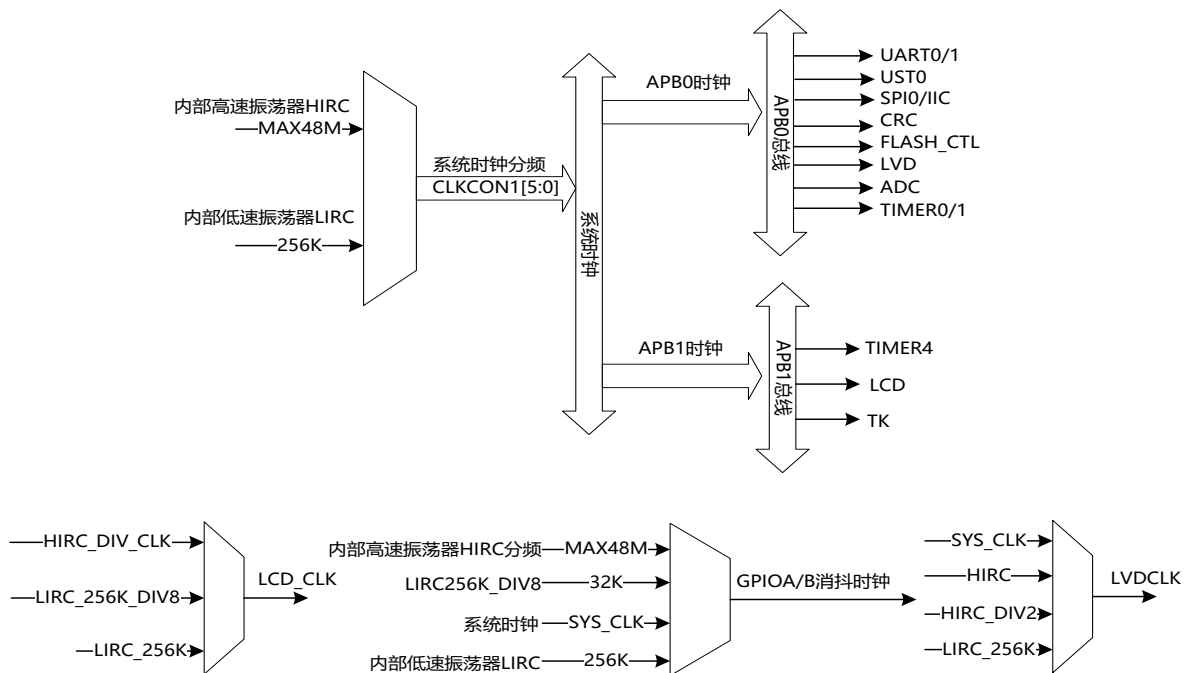


图 9-2 时钟模块

9.3.2. HIRC 时钟

HIRC 时钟信号由内部 RC 振荡器产生，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 需要频率校准，校准值写在 FLASH 系统存储区域。芯片上电时会自动读取 FLASH 中的校准值，把 HIRC 配置到高精度状态。经过出厂校验后，-40°C 至 125°C 范围内 HIRC 精度为 24/32/36/48MHz ($\pm 1\%$)，具体请参考电气特性参数。

9.3.3. 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 256KHz。

9.3.4. 系统时钟(SYSCLK)

两种不同的时钟源可被用来驱动系统时钟(SYSCLK)

内部低速振荡器 256KHz LIRC

内部高速振荡器 HIRC

9.3.5. 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

内部高速振荡器 HIRC

内部低速振荡器 LIRC_256K 8 分频时钟 (32KHz)

系统时钟

内部低速 LIRC_256KHz

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. 通用输入输出 (GPIO)

10.1. 模块介绍

每组 GPIO 端口有四个 32 位配置寄存器(GPIOx_MODE,GPIOx_OTYPE, GPIOx_OSPEED and GPIOx_PUPD), 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。

10.2. 功能特点

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)

模拟功能

复用功能(开漏或推挽、上拉或下拉)

10.3. 功能说明

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(上拉或下拉)

推挽输出

复用功能(开漏或推挽、上拉或下拉)

表 10-1 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	0	GP output	OD
	1			0	0	1	GP output	OD+PU
	1			1	0	0	Reserved	
	1			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	0	AF	OD
	1			0	0	1	AF	OD+PU
	1			1	0	0	Reserved	
	1			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。

10.3.1. 通用 IO (GPIO)

复位期间和刚复位后, 复用功能未开启, IO 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 IO 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 IO 引脚上的数据。

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

10.3.2. 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。

10.3.3. 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 2 输入的多路选择器连接到复用功能 (AF0~AF1), 通过配置 GPIOx_AFRH/L 选择功能。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。

10.3.4. 输入配置

当 IO 端口配置为输入时

输出寄存器被禁止

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 IO 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 IO 状态

10.3.5. 输出配置

当 IO 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 IO 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 IO 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

10.3.6. 模拟输入配置

当 IO 端口被配置为模拟输入配置时
 输出缓存器被禁止
 禁止施密特触发输入，实现了每个模拟 IO 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”

10.3.7. 复用功能配置

对 IO 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 IO 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 IO 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

10.4. 模块框图

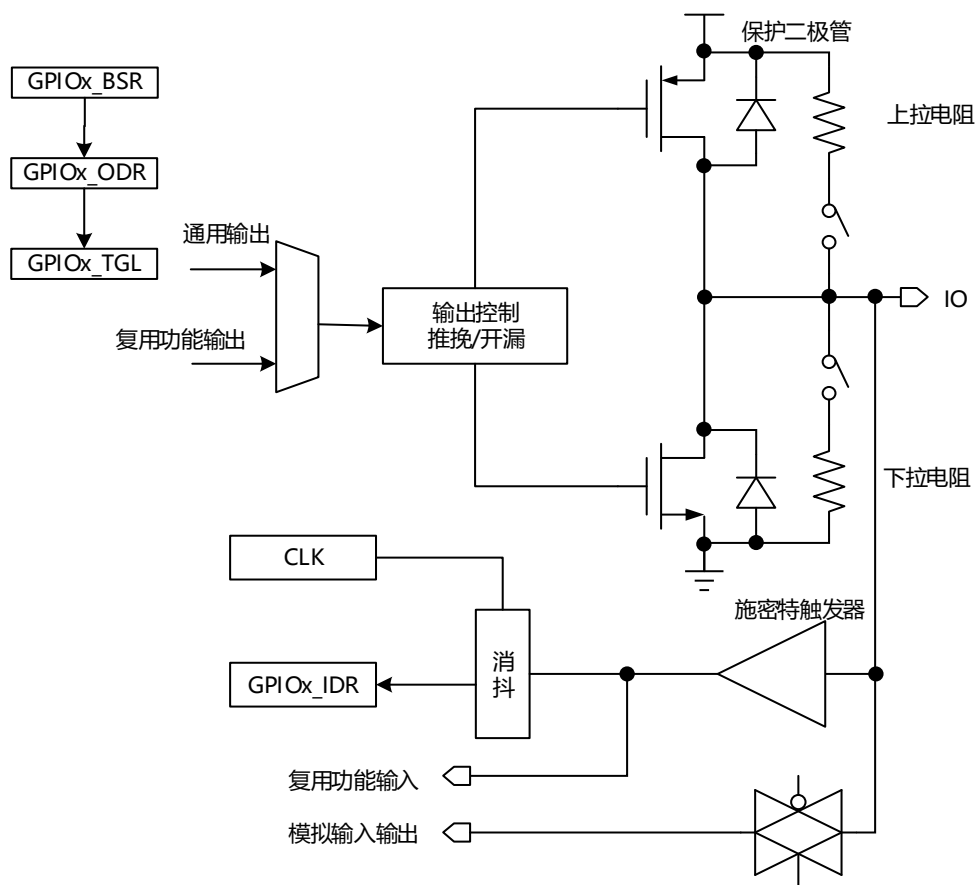


图 10-1 GPIO 模块框图

10.5. 寄存器描述

10.5.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset (A/B)	Reset	Description
GPIOx_MODE	0x00A0/0x00E0	32'h0	GPIO 端口模式寄存器
GPIOx_OTYPE	0x00A4/0x00E4	32'h0	GPIO 端口输出类型寄存器
GPIOx_OSPEEDL	0x00A8/0x00E8	32'h0	GPIO 端口输出速度档位低位寄存器
GPIOx_OSPEEDH	0x00AC/0x00EC	32'h0	GPIO 端口输出速度档位高位寄存器
GPIOx_PUPD	0x00B0/0x00F0	32'h0	GPIO 端口上拉/下拉寄存器
GPIOx_IDAT	0x00B4/0x00F4	32'h0	GPIO 端口输入数据寄存器
GPIOx_ODAT	0x00B8/0x00F8	32'h0	GPIO 端口输出数据寄存器
GPIOx_BSR	0x00BC/0x00FC	32'h0	GPIO 端口置位/复位寄存器
Reserved	0x00C0/0x0100	-	-
GPIOx_AFR_L	0x00C4/0x0104	32'h0	GPIO 端口复用功能低位寄存器
GPIOx_AFR_H	0x00C8/0x0108	32'h0	GPIO 端口复用功能高位寄存器
GPIOx_TGL	0x00CC/0x010C	32'h0	GPIO 端口取反寄存器
GPIOx_IMK	0x00D0/0x0110	32'h0	GPIO 端口中断寄存器
GPIOx_TG_PEND	0x00D4/0x0114	32'h0	GPIO 端口标记寄存器
GPIOx_IE_EN	0x00D8/0x0118	32'h0	GPIO 端口输入控制寄存器
GPIOx_TG_EDGE	0x00DC/0x011C	32'h0	GPIO 端口翻转沿控制寄存器

10.5.2. 寄存器详细说明

10.5.2.1. GPIO 端口模式寄存器 (GPIOx_MODE) (X=A..B)

Width	Name	Reset	Property	Description
31:30	MODER15	2'b0	RW	MODERy[1:0]: GPIOx 的模式位 (y=0..15) 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	

10.5.2.2. GPIO 端口输出类型寄存器 (GPIOx_OTYPE) (X=A..B)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	OT15	1'b0	RW	OTy: GPIOx 的输出类型选择位 (y = 0..15) 0: 推挽输出 (复位状态) 1: 开漏输出
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	

10.5.2.3. GPIO 端口输出速度档位低位寄存器 (GPIOx_OSPEEDL) (X=A..B)

Width	Name	Reset	Property	Description
31:28	OSPEED7	4'b0	RW	OSPEEDy[3:0]: GPIOx 驱动电流档位选择位 (y=0..7) 对于 GPIOB0~7: bx000: 低档 bx001: 1 档 bx010: 2 档 bx011: 3 档 bx100: 4 档 bx101: 5 档 bx110: 6 档 bx111: 7 档 其它: 保留
27:24	OSPEED6	4'b0	RW	
23:20	OSPEED5	4'b0	RW	
19:16	OSPEED4	4'b0	RW	
15:12	OSPEED3	4'b0	RW	
11:8	OSPEED2	4'b0	RW	
7:4	OSPEED1	4'b0	RW	
3:0	OSPEED0	4'b0	RW	对于 GPIOA&GPIOB8~13: bxx00: 低档 bxx01: 1 档 bxx10: 2 档 bxx11: 3 档 其它: 保留

10.5.2.4. GPIO 端口输出速度档位高位寄存器 (GPIOx_OSPEEDH) (X=A..B)

Width	Name	Reset	Property	Description
31:28	OSPEED15	4'b0	RW	OSPEEDy[3:0]: GPIOx 驱动电流档位选择位 (y=0..7) 对于 GPIOB0~7: bx000: 低档
27:24	OSPEED14	4'b0	RW	
23:20	OSPEED13	4'b0	RW	

19:16	OSPEED12	4'b0	RW	bx001: 1 档 bx010: 2 档 bx011: 3 档 bx100: 4 档 bx101: 5 档 bx110: 6 档 bx111: 7 档 其它: 保留
15:12	OSPEED11	4'b0	RW	
11:8	OSPEED10	4'b0	RW	
7:4	OSPEED9	4'b0	RW	
3:0	OSPEED8	4'b0	RW	
				对于 GPIOA&GPIOB8~9: bxx00: 低档 bxx01: 1 档 bxx10: 2 档 bxx11: 3 档 其它: 保留

10.5.2.5. GPIO 端口上拉/下拉寄存器 (GPIOx_PUPD) (X=A..B)

Width	Name	Reset	Property	Description
31	PD15	1'b0	RW	PDy: GPIOx 的下拉使能位 (y = 0..15) 0: 不使能 1: 使能 (输入模式下使用)
30	PD14	1'b0	RW	
29	PD13	1'b0	RW	
28	PD12	1'b0	RW	
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	PUy: GPIOx 的上拉使能位 (y = 0..15) 0: 不使能 1: 使能 (输入模式或开漏输出模式下使用)
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	

10.5.2.6. GPIO 端口输入数据寄存器 (GPIOx_IDAT) (X=A..B)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	IDAT15	1'b0	RO	IDATy: GPIOx 的输入数据位(y = 0..15) 这些位为只读, 它们代表相应 IO 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	

10.5.2.7. GPIO 端口输出数据寄存器 (GPIOx_ODAT) (X=A..B)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	ODAT15	1'b0	RW	ODATy: GPIOx 的输出数据位(y = 0..15) ODAT 寄存器的这些位可以单独设置或者通过写入 GPIOx_BSR 中的 BS/BR 对这组寄存器进行置位/复位。
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	

10.5.2.8. GPIO 端口置位/复位寄存器(GPIOx_BSR) (X=A..B)

Width	Name	Reset	Property	Description
31	BR15	1'b0	WO	BRy: GPIOx 的清除位(y = 0..15) 这些位只写, 对这些位进行读操作将返回值 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	BSy: GPIOx 的设置位(y = 0..15) 这些位只写, 对这些位进行读操作将返回值 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	

10.5.2.9. GPIO 端口复用功能低位寄存器(GPIOx_AFR1) (X=A..B)

Width	Name	Reset	Property	Description
31:28	AFR7	4'b0	RW	AFRy: GPIOx 的复用通道选择位(y = 0..7) b0000: AF0 b0001: AF1 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	

10.5.2.10. GPIO 端口复用功能高位寄存器(GPIOx_AFRH) (X=A..B)

Width	Name	Reset	Property	Description
31:28	AFR15	4'b0	RW	AFRy: GPIOx 的复用通道选择位(y = 0..7) b0000: AF0 b0001: AF1 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	

10.5.2.11. GPIO 端口取反寄存器(GPIOx_TGL) (X=A..B)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	TG15	1'b0	WO	TGy: GPIOx 的翻转位(y = 0..15) 这些位是只写的。对这些位进行读操作将返回值 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	

10.5.2.12. GPIO 端口中断寄存器(GPIOx_IMK) (X=A..B)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	IMK15	1'b0	RW	IMKy: GPIOx 的中断使能位(y = 0..15) 如果 IMKx 使能, 那么当对应的 TGPEND 位为 1 时, 会产生中断 0: 中断不使能 1: 中断使能
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	

10.5.2.13. GPIO 端口标志寄存器(GPIOx_TGPEND) (X=A..B)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	TGPEND15	1'b0	RC	TGPENDy: GPIOx 的输入边沿检测标志位(y = 0..15) 根据 GPIOx_TG_EDGE 的配置决定检测 IO 输入的哪个边沿。 0: 检测 IO 输入边沿无效 1: 检测 IO 输入边沿有效 软件可写 1 清 0
14	TGPEND14	1'b0	RC	
13	TGPEND13	1'b0	RC	
12	TGPEND12	1'b0	RC	
11	TGPEND11	1'b0	RC	
10	TGPEND10	1'b0	RC	
9	TGPEND9	1'b0	RC	
8	TGPEND8	1'b0	RC	
7	TGPEND7	1'b0	RC	
6	TGPEND6	1'b0	RC	
5	TGPEND5	1'b0	RC	
4	TGPEND4	1'b0	RC	
3	TGPEND3	1'b0	RC	
2	TGPEND2	1'b0	RC	
1	TGPEND1	1'b0	RC	
0	TGPEND0	1'b0	RC	

10.5.2.14. GPIO 端口输入使能寄存器(GPIOx_IE_EN) (X=A..B)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	GPIOx15_IE_EN	1'b0	RW	GPIOx_IE_ENy: GPIOx 的输入 (IE) 强制使能控制位 (y = 0..15) 0: 不强制使能 IE 1: 强制使能 IE
14	GPIOx14_IE_EN	1'b0	RW	
13	GPIOx13_IE_EN	1'b0	RW	
12	GPIOx12_IE_EN	1'b0	RW	
11	GPIOx11_IE_EN	1'b0	RW	
10	GPIOx10_IE_EN	1'b0	RW	
9	GPIOx9_IE_EN	1'b0	RW	
8	GPIOx8_IE_EN	1'b0	RW	
7	GPIOx7_IE_EN	1'b0	RW	
6	GPIOx6_IE_EN	1'b0	RW	
5	GPIOx5_IE_EN	1'b0	RW	
4	GPIOx4_IE_EN	1'b0	RW	
3	GPIOx3_IE_EN	1'b0	RW	
2	GPIOx2_IE_EN	1'b0	RW	
1	GPIOx1_IE_EN	1'b0	RW	

0	GPIOx0_IE_EN	1'b0	RW	
---	--------------	------	----	--

10.5.2.15. GPIO 端口边沿检测寄存器(GPIOx_TG_EDGE) (X=A..B)

Width	Name	Reset	Property	Description
31:30	TG_DEGE15	2'b0	RW	TG_EDGEy: GPIOx 的输入边沿的检测控制位(y = 0..15) b00: 不检测边沿 b01: 检测上升沿 b10: 检测下降沿 b11: 检测上升沿与下降沿
29:28	TG_DEGE14	2'b0	RW	
27:26	TG_DEGE13	2'b0	RW	
25:24	TG_DEGE12	2'b0	RW	
23:22	TG_DEGE11	2'b0	RW	
21:20	TG_DEGE10	2'b0	RW	
19:18	TG_DEGE9	2'b0	RW	
17:16	TG_DEGE8	2'b0	RW	
15:14	TG_DEGE7	2'b0	RW	
13:12	TG_DEGE6	2'b0	RW	
11:10	TG_DEGE5	2'b0	RW	
9:8	TG_DEGE4	2'b0	RW	
7:6	TG_DEGE3	2'b0	RW	
5:4	TG_DEGE2	2'b0	RW	
3:2	TG_DEGE1	2'b0	RW	
1:0	TG_DEGE0	2'b0	RW	

11. 同步串行接口 (SSP)

11.1. 模块介绍

SPI_IIC 模块可用作 SPI 接口通信和 IIC 接口通信，两种功能同一时间只能选择其中一种使用。该模块集成两种接口协议，节省资源的同时又能满足不同的应用需求。

11.2. 功能特点

11.2.1. SPI 功能

- 支持主模式和从模式工作
- 可编程时钟极性，采样相位，支持 4 种模式
- 支持 1~32bit 传输
- 支持 5byte 发送/接收数据缓冲
- 传输数据顺序 MSB 和 LSB
- 支持标准模式，三线模式
- 可触发中断的专用发送和接收标志

11.2.2. IIC 功能

- 支持主模式和从模式
- 主模式支持时钟同步和总线仲裁
- 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
- 从模式支持接收广播地址
- 支持 5byte 发送/接收数据缓冲

11.3. 功能说明

11.3.1. SPI 工作模式

- 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

11.3.2. SPI 接口模式

- 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

11.3.3. SPI 数据帧与内部缓存

SPI 可支持 1~32bit 帧数据传输，内部集成了一个 40bit 的缓冲区，根据配置的数据帧长度不同，缓冲区能缓存的帧数也不一样。数据帧长度 $\leq 8\text{bit}$ 时，缓冲区可容纳 5 帧数据， $8\text{bit} < \text{数据帧长度} \leq 16\text{bit}$ 时，缓冲区可容纳两帧数据，数据帧长度 $> 16\text{bit}$ 时，缓冲区可容纳 1 帧数据，当缓冲区无法再容下一帧数据时，缓冲区满标志会置 1。

11.3.4. IIC 主机时钟同步和总线仲裁

IIC 主机模式时，在多主机的应用场景下，支持时钟同步和总线仲裁。当总线上连接了不止一个主机时，就会存在同时发起通信的情况，这时候需要时钟同步以及总线仲裁机制决定由哪个主机占用总线完成数据传输。

时钟同步的原理：IIC 总线上的不同主机可能发起传输时的时钟频率不一样，通过时钟同步机制，可

以让所有主机的时钟同步，才能进行逐位仲裁。所有主机的 SCL 在总线上是线与的关系，当总线上的 SCL 由高切换到低电平时，所有主机从 0 开始计算低电平周期时间。当电平时间达到时，如果总线上的其它主机的 SCL 低电平仍然保持，那么其它主机进入高电平等待状态，等低电平时间最长的主机的 SCL 拉高时再统一拉高。因此，总线上同步后的 SCL 的低电平时间由低电平周期最长的主机决定，而高电平时间由高电平周期最短的主机决定。

总线仲裁原理：IIC 总线上不同主机的 SDA 线也是线与的关系，各主机在 SCL 线为高电平时，检查 SDA 线的电平是否和自己发送的 SDA 信号一致，如果检测到 SDA 线为低电平时，自己要发送的 SDA 信号为高电平，那么该主机仲裁失败，停止总线上的传输动作。

11.3.5. IIC 从机拉低 SCL

IIC 从机在发送状态下如果缓冲区中没有可发送的数据，或者在接收状态下缓冲区已满时，将会在 SCL 端口输出低电平，拉低总线上的 SCL 信号使主机暂停发送时钟。当从机准备好发送或者接收后，将在 SCL 输出高电平，主机又可以重新控制总线上的 SCL 线，恢复数据传输。

11.3.6. IIC 从机支持 7bit/10bit 寻址

IIC 从机支持 7bit 或者 10bit 寻址模式，由寄存器 CON0 的 IIC_SLAVE_ADDR_WIDTH 位决定。

7bit 模式下，主机需要在 TX 模式下发送带 START 位的 7bit 地址，最后 1bit 为 R/W 标志。当主机写从机时，寻址完成即可进行数据发送。当主机读从机时，寻址完成后，主机需要改成 RX 模式，然后配置准备接收的数据长度（寄存器 DMA_LEN），然后对寄存器 CMD_DATA 写任意值启动接收。

10bit 模式下，主机需要在 TX 模式下发送带 START 位的第 1byte 地址（此时 R/W 位为 1），接着发送第 2byte 地址，此时如果收到从机的 ACK 信号，则为寻址成功。接下来如果是主机写从机，就可以直接进行数据发送。如果是主机读从机，那么需要主机再次发送带 START 为的第 1byte 地址（此时 R/W 为 1），然后切换成 RX 模式，配置寄存器 DMA_LEN，并且对寄存器 CMD_DATA 写任意值启动读数据。

主机读从机时，从机被寻址成功且收到读标志时，需要切换成 TX 模式，并且往缓冲区中写入要发送的数据。

11.4. IO 映射

详细参考 [3.2. 管脚定义](#)

11.5. 模块框图与接口时序

11.5.1. 模块架构图

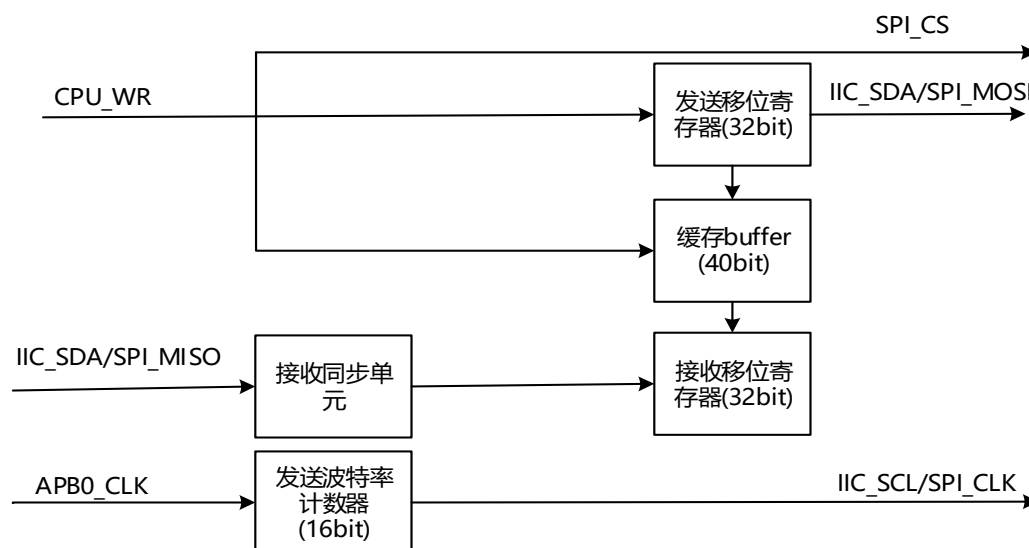


图 11-1 SPI/IIC 模块架构图

11.5.2. SPI 时序图

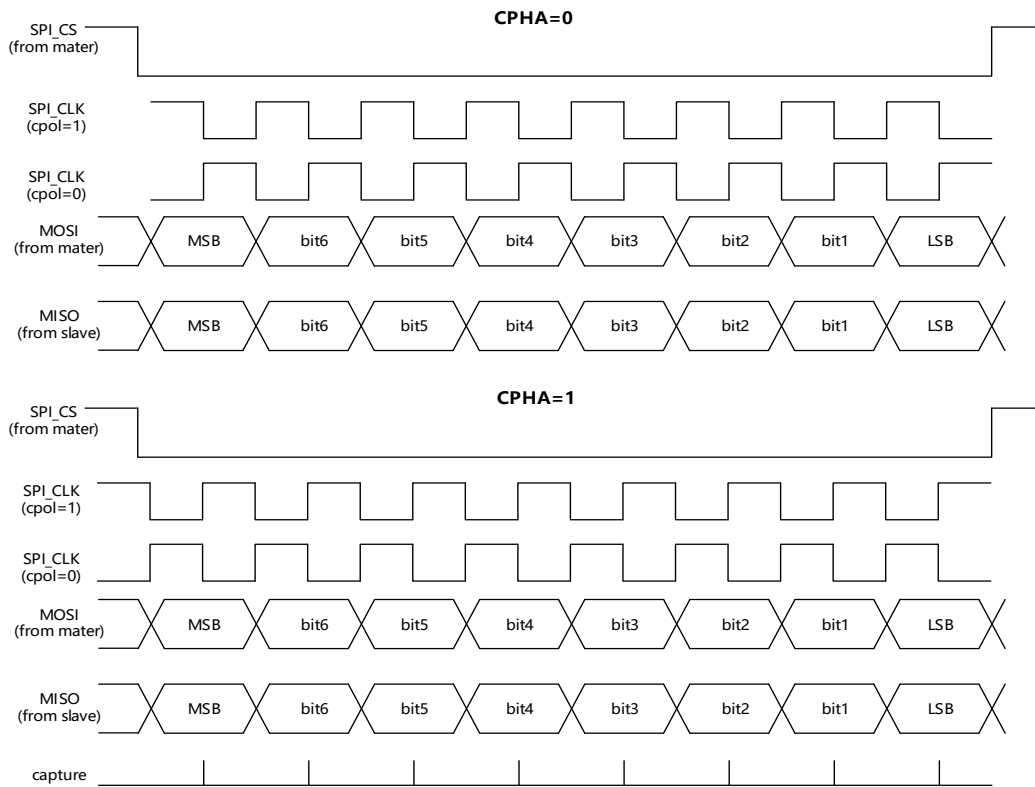


图 11-2 SPI 时序图

11.5.3. IIC 时序图

IIC 数据传输中，每帧数据为 8bit，传送数据时先传高位(MSB)，每帧数据传输完之后必须跟随一位应答位（低电平为应答，高电平为非应答）。

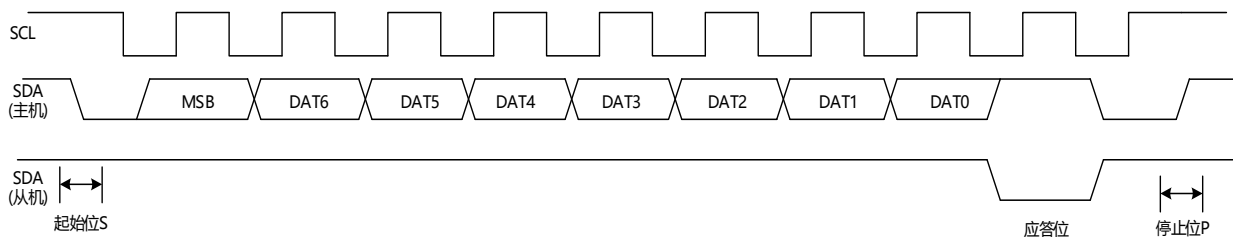


图 11-3 IIC 时序图

11.5.4. IIC 数据传输

7bit 寻址，主机向从机写数据：

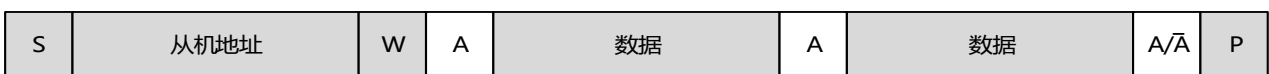


图 11-4 IIC 主机向从机写数据 (7bit 寻址)

7bit 寻址，主机向从机读数据：

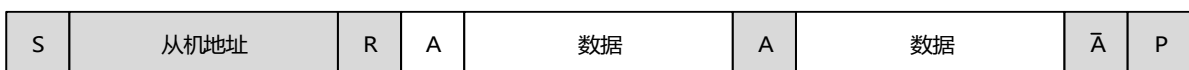


图 11-5 IIC 主机向从机读数据 (7bit 寻址)

7bit 寻址，主机向从机先写数据后读数据：

S	从机地址	W	A	数据	A	S	从机地址	R	数据	\bar{A}	P
---	------	---	---	----	---	---	------	---	----	-----------	---

图 11-6 IIC 主机向从机先写数据后读数据（7bit 寻址）

10bit 寻址主机向从机写数据：

S	从机地址1 (7bit)	W	A	从机地址2 (8bit)	A	数据	\bar{A}	数据	A/A	P
---	--------------	---	---	--------------	---	----	-----------	----	-----	---

图 11-7 IIC 主机向从机写数据（10bit 寻址）

10bit 寻址主机向从机读数据：

S	从机地址1 (7bit)	W	A	从机地址2 (8bit)	A	Sr	从机地址1 (7bit)	R	\bar{A}	数据	A	数据	A/A	P
---	--------------	---	---	--------------	---	----	--------------	---	-----------	----	---	----	-----	---

图 11-8 IIC 主机向从机读数据（10bit 寻址）

11.6. 时钟与复位

11.6.1. 时钟介绍

该模块时钟来源于系统时钟，可通过配置系统寄存中 CLKCON2 来使能时钟。

11.6.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，软件复位可通过配置系统寄存器触发。

11.7. 寄存器描述

11.7.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
CON0	0x02C0	32'h0	SPI_IIC 控制寄存器 0
CON1	0x02C4	32'h0	SPI_IIC 控制寄存器 1
CMD_DATA	0x02C8	32'h0	SPI_IIC 数据寄存器
BAUD	0x02CC	32'h0	SPI_IIC 波特率寄存器
DMA_LEN	0x02D0	32'h0	SPI_IIC 数据长度寄存器
DMA_CNT	0x02D4	32'h0	SPI_IIC 接收数据计数寄存器
Reserved	0x02D8	32'h0	-
STA	0x02DC	32'h804	SPI_IIC 状态寄存器

11.7.2. 寄存器详细说明

11.7.2.1. SPI 控制寄存器 (CON0)

Width	Name	Reset	Property	Description
31:20	Reserved	-	-	-
19:14	SPI_FRAME_SIZE	6'h0	RW	SPI 数据帧长度 0: 禁止 1: 每帧传输 1bit 2: 每帧传输 2bit 3: 每帧传输 3bit

				... 31: 每帧传输 31bit 32: 每帧传输 32bit 33~63: 禁止 注意: 当数据帧小于 32bit 时, 数据寄存器的低位有效, 比如 SPI_FRAME_SIZE=8 那么 CMD_DATA 的低 8 位为有效数据。
13	CS_POS_IE	1'b0	RW	SPI 从机模式 CS 上升沿中断使能 0: 不使能 1: 使能 (检测到 CS 上升沿产生中断)
12	SPI_CS_EN	1'b0	RW	SPI_CS 引脚使能, 此位仅在 SPI 从模式下有效 0: 不使能, SPI 总线没有 CS 引脚 1: 使能, SPI 总线有 CS 引脚
11	SPI_CS	1'b0	RW	SPI CS 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	SPI_MASTER_CAP_DELAY	3'b0	RW	SPI 主机模式延时采样数据 b000: 不延时 b001: 延时 1 个周期 b010: 延时 2 个周期 ... b111: 延时 7 个周期
7	SPI_SLAVE_SYNC_EN	1'b0	RW	SPI 从机模式同步输入数据使能 0: 不使能 1: 使能
6	SPI_MASTER_SYNC_EN	1'b0	RW	SPI 主机模式同步输入数据使能 0: 不使能 1: 使能
5	Reserved	-	-	-
4	LSBFE	1'b0	RW	SPI 先传输低位使能 0: 先传输高位 1: 先传输低位
3	Reserved	-	-	-
2	WIRE_MODE	1'b0	RW	SPI 通信模式: 0: 普通模式 (CS, MOSI, MISO) 1: 3 线模式 (CS, CLK, IO0)
1:0	SPI_MODE	2'b0	RW	SPI 接口时钟极性与采样相位 CPOL, CPHA: b00: 时钟初始为 0, 上升沿采样, 下降沿出数据 b01: 时钟初始为 0, 下降沿采样, 上升沿出数据 b10: 时钟初始为 1, 下降沿采样, 上升沿出数据 b11: 时钟初始为 1, 上升沿采样, 下降沿出数据

11.7.2.2. IIC 控制寄存器 (CON0)

Width	Name	Reset	Property	Description
31:23	Reserved	-	-	-
22	IIC_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	IIC_STOP_IE	1'b0	RW	检测到线上有 STOP 信号中断使能(主机和从机) 0: 不使能 1: 使能
19	IIC_ADDR_MATCH_IE	1'b0	RW	从机地址匹配中断使能 0: 不使能 1: 使能
18:14	IIC_FILTER_MAX	5'h0	RW	IIC 每 IIC_FILTER_CNT 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	IIC_BROADCAST_IE	1'b0	RW	IIC 从机接收到广播地址中断使能 0: 不使能 1: 使能
12	IIC_BROADCAST_EN	1'b0	RW	IIC 从机接收到广播地址使能 0: 忽略广播地址 1: 接收到广播地址时, 回应 ACK, 并产生 BROADCAST_PEND 标志
11:2	IIC_ADDR	10'h0	RW	IIC 从机地址
1	IIC_TX_NACK	1'b0	RW	IIC 在接收数据的时候, 回应 NACK 还是 ACK 选择位 0: ACK 1: NACK
0	IIC_ADDR_WIDTH	1'b0	RW	IIC 从机地址宽度 0: 7bit 1: 10bit

11.7.2.3. 控制寄存器 (CON1)

Width	Name	Reset	Property	Description
31:9	Reserved	-	-	-
8	BUF_OV_IE	1'b0	RW	接收缓冲区溢出中断使能 0: 不使能 1: 使能
7	RBUF_NEMPTY_IE	1'b0	RW	接收缓冲区不空中断使能 0: 不使能 1: 使能
6	TBUF_NFULL_IE	1'b0	RW	发送缓冲区不满中断使能 0: 不使能 1: 使能
5	SSP_DONE_IE	1'b0	RW	发送/接收完一帧数据中断使能 0: 不使能 1: 使能
4	Reserved	-	-	-
3	SSP_TX_RX	1'b0	RW	接口发送/接收使能 0: RX_EN 1: TX_EN

2	MASTER_SLAVE_SEL	1'b0	RW	接口主从模式设置 0: 主模式 1: 从模式
1	SPI_IIC_SEL	1'b0	RW	SPI/IIC 模式选择 0: SPI 模式 1: IIC 模式
0	SSP_EN	1'b0	RW	SPI 和 IIC 模块使能 0: 不使能 1: 使能

11.7.2.4. 数据寄存器 (CMD_DATA)

Width	Name	Reset	Property	Description
31:0	CMD_DATA	32'h0	RW	SPI 接口 写: 将要发送的数据写入该寄存器, 触发 SPI 发送 读: 读该寄存器, 获取收到的数据 IIC 接口 写: [31:10] 无效位 [9] STOP 使能位, 在发送完 1byte 数据后, 发送一个 STOP 位。(仅在主机模式有效) [8] START 使能位, 使能后, 在发送 1byte 数据前会先发 START 位。(仅在主机模式有效) [7:0] 将要发送的 8bit 数据 读: [31:8] 无效位 [7:0] 获取收到的数据

11.7.2.5. 波特率寄存器 (BAUD)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	BAUD	16'h0	RW	主机模式 SPI 波特率= $\text{SYSCLK}/(2*(\text{BAUD}+1))$ IIC 波特率= $\text{SYSCLK}/(4*(\text{BAUD}+1))$ 从机模式 SPI: 无效 IIC: 从机准备好发送数据再延时 (BAUD+1) 个 SYSCLK 时钟周期再释放 SCL

11.7.2.6. 控制寄存器 (DMA_LEN)

Width	Name	Reset	Property	Description
31:12	Reserved	-	-	-
11:0	SPI_DMA_LEN	12'h0	RW	主机模式接收时, 表示要接收的数据长度。

11.7.2.7. 控制寄存器 (DMA_CNT)

Width	Name	Reset	Property	Description
31:12	Reserved	-	-	-
11:0	SPI_DMA_CNT	12'h0	RO	SPI 主机接收模式时，表示已经接收到并且被 CPU 读走的数据 byte 长度。 该寄存器在启动接收时被清零。

11.7.2.8. 状态寄存器 (STA)

Width	Name	Reset	Property	Description		
31:28	Reserved	-	-	-		
27	SLAVE_ADDRED	1'b0	RO	IIC 从机模式，被寻址标志 0: 没有被寻址 1: 已经被寻址		
26:24	STATE	3'h0	RO	IIC 状态	主	从
				b000: IDLE	空闲	空闲
				b001:STAR T	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:ADD R0	保留	等待接收 1byte 地址
				b110:ADD R1	保留	等待接收 2byte 地址
23:20	Reserved	-	-	-		
19	CLR_BUF_CNT	1'b0	WO	软件对该 bit 写 1 使 BUF_CNT 清零		
18:16	BUF_CNT	3'h0	RO	表示缓冲区里面有多少 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/加 1，16bit 数据减/加 2，24bit 和 32bit 数据减/加 4		
15	MASTER_RX_BUSY	1'b0	RO	主机模式接收数据时，指示是否读完需要读取的数据长度 0: 已读完 1: 未读完		
14	IIC_RX_NACK	1'b0	RC	IIC 主从模式发送数据的时，在第 9bit 收到的应答信号 0: ACK 1: NACK		
13	IIC_SLAVE_RW	1'b0	RO	IIC 从机在接收地址阶段，接收到的读写标志 0: 主机写从机 1: 主机读从机		
12	IIC_BUS_BUSY	1'b0	RO	IIC 检测到线路 busy 0: 线路空闲		

				1: 线路繁忙 该位在检测到 start 位时置 1, 检测到 stop 位时清零.
11	SPI_SLAVE_CS	1'b1	RO	SPI 从机模式接收到的 CS 电平状态
10	SSP_BUSY	1'b0	RO	SPI/IIC 作为主机或从机发送模式时的状态 0: 空闲 1: 主机正在发送/接收一帧数据 从机正在发送一帧数据 注: 从机接收模式下该 bit 无效
9	AL_PEND	1'b0	RC	IIC 主机模式检测到仲裁丢失 0: 没有仲裁丢失 1: 仲裁丢失 注: 只能软件清零, 且必须清掉 IIC 才能正常工作
8	STOP_PEND	1'b0	RC	IIC 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: 只能软件清零
7	ADDR_MATCH_PEN D	1'b0	RC	IIC 从机模式下, 接收到主机发送过来的正确从机地址 0: 从机地址不匹配 1: 从机地址匹配 注: 只能软件清零
6	IIC_BROADCAST_PEN D	1'b0	RC	IIC 从机模式下, 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: 只能软件清零
5	SPI_CS_POS_PEND	1'b0	RC	SPI 从机模式检测到 CS 引脚上升沿 0: 没有检测到上升沿 1: 检测到上升沿 注: 只能软件清零
4	Reserved	-	-	-
3	BUF_OV	1'b0	RC	缓冲区溢出, 有数据丢失标志 0: 缓冲区没有溢出 1: 缓冲区已经满, 又有新数据要写入, 丢失最新收到的数据 注: 缓冲区容量是 5byte, 但不一定是装满 5byte 数据才是满, 而是装不下一帧新的数据时就认为是满。
2	BUF_EMPTY	1'b1	RO	缓冲区空标志 0: 非空 1: 空
1	BUF_FULL	1'b0	RO	缓冲区满标志 0: 非满 1: 满
0	SSP_DONE	1'b0	RC	SPI 模式 0: 未完成数据收发 1: 已经完成一帧数据的发送/接收 IIC 模式 0: 未完成数据收发 1: 已经完成一帧数据发送/接收

				<p>注: IIC 主机模式的 数据帧 (START(可 选)+WRITE/READ(8bit+ACK)+STOP(可选)) IIC 从机模式的数据帧(8bit+ACK)/ADDR0/ADDR1</p>
--	--	--	--	---

12. 通用异步收发器 (UART0/1)

12.1. 模块介绍

CIU32F011、CIU32F031 集成的通用异步收发器(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。

12.2. 功能特点

- 支持 8bit 数据和 9bit 数据模式
- 支持 18bit 波特率寄存器灵活配置
- 支持全双工异步通信、单工发送、单工接收、单线通信模式
- 支持硬件奇偶校验可选
- 具有 4 帧数据的接收缓存, 1 帧数据的发送缓冲
- 硬件检测接收超时, 超时长度可以配置, 配置范围: 1~16384 比特率时间
- 支持帧出错检测
- 可选 STOP 位为 1 位或 2 位

12.3. 功能说明

12.3.1. 波特率生成功能

波特率计算公式 = $\text{SYSCLK}/(\text{UART_BAUD}+1)$ ($\text{UART_BAUD} \geq 6$)

UART 模块的工作时钟为系统时钟, 因此, 应当根据当前系统时钟频率以及需要的波特率计算 UART_BAUD 的配置值。一旦系统时钟频率改变, 应当相应修改波特率配置。

12.3.2. UART 发送器

发送器可发送 8/9bit 数据, 由寄存器 UART_CON 中的 BIT9_EN 位的配置值决定。软件向寄存器 UART_DATA 写入数据时, 发送器将通过 IO 管脚 UART_TX 发送数据帧, UART_TX 的极性可以通过寄存器 UART_CON 中的 TX_INV 位来配置。

发送数据流程: 发射器输出管脚 (UART_TX) 闲置状态时, 默认为高电平。模块使能后, 软件向寄存器 UART_DATA 写入要发送的数据启动发送。发送器有一帧的发送缓冲, 寄存器 UART_STA 中 TX_BUF_EMPTY 为高电平时, 软件可向寄存器 UART_DATA 再写入一帧数据, 它将被存入发送缓冲区, TX_BUF_EMPTY 会变零, 在当前帧发送完成时, 会接着发送缓冲区中的数据。

数据发送完成且缓冲区为空时, 寄存器 UART_STA 中 TC_PEND 位置 1, 此时如果寄存器 UART_CON 中 TCIE 位使能, 则产生中断。

发送器发送序列: 起始位->数据位(LSB)->停止位

12.3.3. UART 接收器

接收器可接收 8/9bit 数据, 由寄存器 UART_CON 中的 BIT9_EN 位的配置值决定。

数据字符由逻辑 0 的起始位、8/9bit 数据位, 奇偶校验位和逻辑 1 的停止位组成。

接收器有 4 个 8/9bit 的数据缓冲。在缓冲区接收到 4 个数据之后, 且又有一帧数据接收完成时, 寄存器 UART_STA 中 RX_BUF_OV 会置 1, 新的数据将不会存储在数据缓冲区中, 即新数据丢失。

在接收一个数据帧的过程中可使能奇偶错误检测、帧错误检测和超时检测, 通过配置寄存器 UART_CON 打开相应的错误检测使能位和相应的错误中断使能位。

帧错误检测机制是指在 stop 位检测到 IO 引脚 UART_RX 是低电平时, 为帧错误。

超时检测机制是指在接收到 1byte 之后, 检测是否超过了设置的时间, 如果没有则继续接收数据, 超时时间可以通过寄存器 UART_CON 中 TO_BIT_LEN 进行配置。

接收缓冲区有数据, 即寄存器 UART_STA 中的 RX_BUF_NOT_EMPTY=1 时, 软件可以通过读 UART_DATA 寄存器的方式来获取收到的数据。

12.4. IO 映射

详细参考 [3.2. 管脚定义](#)

12.5. 模块框图与接口时序

12.5.1. UART 模块架构图

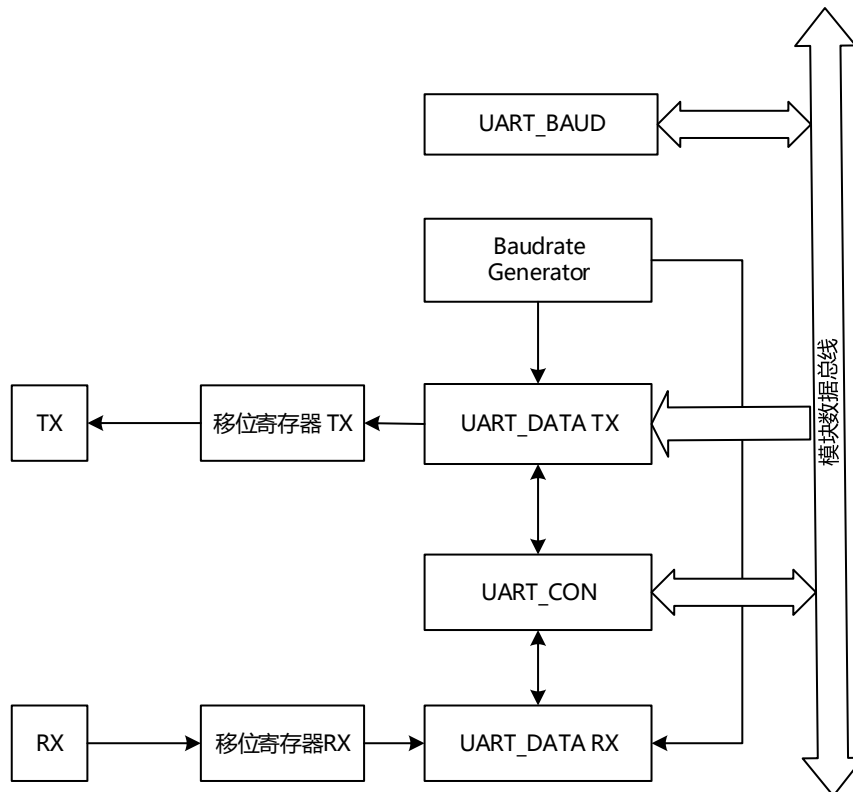


图 12-1 UART 模块架构图

12.5.2. UART 协议图

UART 数据发送/接收的数据格式如下所示, 其中数据位可选择 8bit 或 9bit, 需要注意的是, 选择 9bit 数据时, 就不可以使能奇偶校验位。相反, 使能奇偶校验位时, 不能选择 9bit 数据。

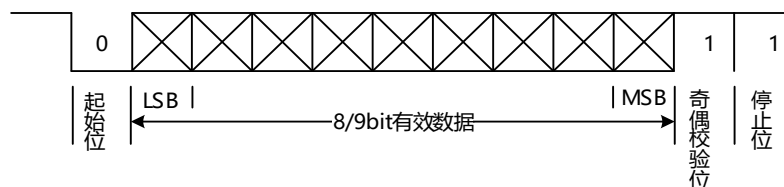


图 12-2 UART 时序图

12.6. 时钟与复位

12.6.1. 时钟介绍

该模块时钟来源于系统时钟, 可通过配置系统寄存中 CLKCON2 来使能时钟。

12.6.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，软件复位可通过配置系统寄存器触发。

12.7. 寄存器描述

12.7.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
UART_CON	0x0290/0x02A0	32'h360000	UART 模式控制寄存器
UART_BAUD	0x0294/0x02A4	18'h9c3	UART 波特率控制寄存器
UART_DATA	0x0298/0x02A8	xxxx	UART 数据寄存器
UART_STA	0x029C/0x02AC	17'h0	UART 状态寄存器

12.7.2. 寄存器详细说明

12.7.2.1. 控制寄存器 (UART_CON)

Width	Name	Reset	Property	Description
31	RX_CLOSE_EN	1'b0	RW	关闭 UART 接收功能 0: 不关闭 1: 关闭
30	Reserved	-	-	-
29:16	TO_BIT_LEN	14'h36	RW	超时配置.单位是比特率时间 $TIME\ OUT\ TIME = (TO_BIT_LEN + 1) * BIT\ RATE\ TIME$ 默认值设置的 TIME OUT TIME = $5 * (start_bit + data_bit(8\9) + parity_bit + stop_bit(1\2)) = 55$
15	TC_IE	1'b0	RW	发送完成中断使能 0: 不使能 1: 使能 (UART_STA 寄存器中 TC=1 时产生中断)
14	TMR_PWM_EN	1'b0	RW	UART 的 TX 带上 TIMR PWM 载波 (UART0 对应 TIMER0 的 PWM, UART1 对应 TIMER1 的 PWM) 0: 不使能 1: 使能
13	TO_IE	1'b0	RW	超时中断使能 0: 不使能 1: 使能
12	TO_EN	1'b0	RW	超时检测使能 检测是否经历了 $(TO_BIT_LEN + 1) * BIT_RATE_TIME$ 时间, 没有接收到数据。 0: 不使能 1: 使能 注:每次 TO_EN 后, 收到 1byte 数据后会产生 rx_one_byte 标志, 该标志为 1 才会开始检测。该 标志会在每次清 TO_PEND 的时候清 0, 在 TO_EN 等 于 0 时也会清 0。

11	FERR_IE	1'b0	RW	帧出错中断使能 0: 不使能 1: 使能
10	TX_BUF_EMPTY_IE	1'b0	RW	发送缓冲空中断使能 0: 不使能 1: 使能
9	RX_BUF_NEMPTY_IE	1'b0	RW	接收缓冲非空中断使能 0: 不使能 1: 使能
8	TX_INV	1'b0	RW	发送输出信号电平极性 0: 不使能 1: 使能
7	RX_INV	1'b0	RW	接收输入信号电平极性 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 注: PARITY_EN 和 BIT9_EN 只能二选一 0: 不使能 1: 使能
4	BIT9_EN	1'b0	RW	UART 数据位长度选择 注: PARITY_EN 和 BIT9_EN 只能二选一 0: 8 个数据位 1: 9 个数据位
3	STOP_BIT	1'b0	RW	UART 停止位选择 0: 一个停止位 1: 两个停止位
2	Reserved	-	-	-
1	WORK_MODE	1'b0	RW	UART 通信模式选择 0: 双线全双工模式 1: 单线半双工模式 (软件启动发送时 IO 为输出状态, 不发送时 IO 为输入状态)
0	UART_EN	1'b0	RW	UART 使能 0: 不使能 1: 使能

12.7.2.2. 波特率寄存器 (UART_BAUD)

Width	Name	Reset	Property	Description
31:18	Reserved	-	-	-
17:0	UART_BAUD	18'h9c 3	RW	UART 波特率控制寄存器 波特率=SYSCLK/(UART_BAUD+1) 注: 需配置 UART_BAUD >=6, 否则输入信号会被内部滤波器滤掉。

12.7.2.3. 数据寄存器 (UART_DATA)

Width	Name	Reset	Property	Description
31:9	Reserved	-	-	-
8:0	UART_DATA	xxx	RW	UART 数据寄存器 写: 写入发送数据 读: 获取接收数据 注: 读之前, 先读取 PERR[0]获取奇偶校验信息

12.7.2.4. 状态寄存器 (UART_STA)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15	TC_PEND	1'b0	RC	发送完成标志 当 UART 发送完一帧数据且缓冲区为空时, TC 由硬件置 1, 此时如果 TCIE=1, 则产生中断; 软件向寄存器 UART_DATA 写数据时该位会被清 0, 软件也可以对该位写 1 清 0。 软件读: 0: UART 发送未完成 1: UART 发送完成
14:12	Reserved	-	-	-
11	TO_PEND	1'b0	RC	超时标志 按照 TO_BIT_LEN 默认值配置的时间, 没有收到数据 0: 无效 1: 有效 (软件对该位写 1 清零)
10:7	PERR	4'b0	RO	奇偶检验错误标志 4bit 分别对应 BUF 里面的 4 个数据。硬件会自动将 PERR[0]对应的数据存放在 UARTDATA, 在读 UARTDATA 之前, 需要先读这个寄存器。每读走一个 UARTDATA 的数据, 硬件会自动将这 4bit 右移 1 位。
6:4	RX_CNT	3'b0	RO	接收缓存区数据个数 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: 无效
3	FERR	1'b0	RC	帧错误, 表示在 STOP bit 期间检测到 IO 引脚 UART_RX 出现了低电平。 0: 没有帧错误 1: 出现了帧错误 (软件对该位写 1 清零)
2	RX_BUF_OV	1'b0	RC	接收缓存区满标志 接收 BUF 最多可以容纳 4 个 8/9bit 数据, 如果软件还没有来得及读走, 又收到一个数据, 标志会起来。

				0: 接收了 (<=4byte) 数据 1: 接收了 (>4byte) 的数据, 只保存最开始的 4byte, 其他丢弃。 (软件对该位写 1 清零)
1	RX_BUF_NO_EMPTY	1'b0	RO	接收缓冲区非空标志 0: 接收缓冲区空 1: 接收缓冲区非空
0	TX_BUF_EMPTY	1'b0	RO	发送缓存区空标志 UART 内部处理发送移位寄存器外, 还有一帧 (9bit) 发送缓存, 如果发送缓存为空标志会起来。 0: 发送缓存非空 1: 发送缓存为空

13. 复合通信接口 (UST)

13.1. 模块介绍

CIU32F011、CIU32F031 内部集成了一个复用型的通信接口模块 (UST)，该模块可以用作 UART 收发通信、SPI 收发通信(仅用做主机)，以及 TIMER 定时/PWM 输出，功能三选一。

13.2. 功能特点

UART 功能

- 仅支持半双工异步通信
- 8 位字符通信，无奇偶校验位
- 1 位 STOP 位
- 可编程波特率 (16 位)
- 支持 1byte 接收/发送缓存

SPI 功能

- 仅支持主机发送/接收模式
- 仅支持按 byte 发送/接收
- 传输数据顺序 MSB 和 LSB
- 支持标准模式 (CS,CLK,MOSI,MISO) 和三线模式 (CS,CLK,MOSI)
- 支持时钟极性控制(CPOL)和时钟相位控制(CPHA)
- 可编程 SPI 时钟分频 (16 位)
- 支持 1byte 发送/接收缓存

TIMER 功能

- 支持 16 位 TIMER 计数
- 支持 TIMER 计数预分频
- 计时满产生标志位
- 支持计时满产生中断
- 支持 PWM 输出，占空比和周期有缓冲寄存器，更新不影响 PWM 输出
- 支持同时输出 4 路同周期不同占空比的 PWM
- 支持输出两对带死区的互补 PWM
- 支持 PWM 码流输出，控制外部集成 IC 智能 LED

13.3. 功能说明

13.3.1. UST 工作模式

UST 模块支持三种工作模式: UART 模式、SPI 模式和 TIMER 模式，同一时间只能选择其中一种模式。软件通过配置寄存器 UST_MODE 选择哪种模式，选择完模式后再配置控制寄存器。

13.3.2. UART 波特率生成

波特率生成公式 = $UST_CLK/(UART_BAUD+1)$ ($UART_BAUD \geq 6$)，通信要求发送端和接收端都使用相同的波特率。

UST_CLK 使用系统时钟的分频时钟，所以在使能 UART 之前，必须确保系统时钟切换到正常工作的频率下。如果系统时钟频率在数据接收过程中发生了变化，可能会导致接收错误或导致数据丢失，为了避免此问题，应该确保改变系统时钟频率之前，UART 处于空闲状态。

13.3.3. UART 发送功能

该模块工作在 UART 模式时，只支持半双工模式通信，模块默认处于接收模式，当软件向寄存器 UST_DATA 写入数据时自动切换到发送模式。

UART 发送模式仅支持发送每帧 8bit 数据，发送器在发送当前帧数据的同时，还支持缓存一帧数据，软件判断寄存器 UST_STA 的 UST_BUF_VLD 为 0 时，可以写入一帧数据。

发送数据流程：发送器输出端 (UART_TX) 闲置状态时输出高电平，软件选择 UART 模式并且配置好波特率后，软件应当检测到寄存器 UST_STA 的 UST_BUF_VLD 为 0 且 UST_START 为 0 时，对数据寄存器 UST_DATA 写入要发送的数据启动发送，在发送当前帧数据过程中软件可以把下一帧数据写到发送缓冲区中。发送器在发送完当前帧数据后，如果发送缓冲区中有数据，则继续发送缓冲区中的数据，同时对 UST_BUF_VLD 清 0，允许软件向缓冲区写下一帧数据。

当发送完当前帧数据且发送缓冲区中无数据时，则发送完成标志，即寄存器 UST_STA 的 UART_DONE_PEND 置 1，根据寄存器 UST_CON0 的 UART_DONE_IE 的状态决定是否产生 DONE 中断。发送器发送序列：起始位->数据位->停止位。（发送接收均为 LSB 先发）。

13.3.4. UART 接收功能

该模块工作在 UART 模式时，只支持半双工模式通信，模块默认处于接收模式，当软件向寄存器 UST_DATA 写入数据时自动切换到发送模式，仅支持接收每帧 8bit 数据。

接收器支持 1 帧数据缓存，软件通过读取寄存器 UST_STA 的 UST_BUF_VLD 判断接收缓冲区是否有数据可读。

接收数据流程：接收器输入端 (UART_RX) 检测到下降沿时启动接收，在波特率一半的时间点采样输入信号到移位寄存器中，接收完一帧数据后再保存到接收缓冲区中，并对 UST_BUF_VLD 标志位置 1。

13.3.5. SPI 时钟生成

SPI 无论时发送还是接收模式，SPI_CLK 都由主机输出。

发送模式下，软件向寄存器 UST_DATA 写入数据触发 SPI_CLK 开始输出，直到发送完当前帧数据且缓冲区为空时停止输出。

接收模式下，软件向寄存器 UST_DATA 写入任意值触发 SPI_CLK 开始输出，收完一帧数据则停止输出，需要软件再次触发接收下一帧数据。

$SPI_CLK \text{ 频率} = SYSCLK / (2 * (SPI_BAUD + 1))$ 。

13.3.6. SPI 发送功能

SPI 模式支持主机发送功能，软件通过配置模式寄存器选择了 SPI 模式后，配成控制寄存器选择发送模式并使能模块，写入要发送的数据启动发送。内部支持缓存 1byte 发送数据，寄存器 UST_STA 的 UST_BUF_VLD 标志为 0 时可写入下一帧发送数据。

发送数据流程：软件对寄存器 UST_DATA 写入数据启动发送，并且按照配置的时钟波特率产生 SPI 时钟，并且按照配置在时钟的上升沿/下降沿输出发送信号，支持先发高位/先发低位。当发送完一 byte 数据且缓冲区中无数据时停止发送 SPI 时钟。

标准模式在 MOSI 端口发送数据，三线模式在 MISO 端口完成发送和接收。

13.3.7. SPI 接收功能

SPI 模式支持主机接收功能，软件通过配置模式寄存器选择了 SPI 模式后，配成控制寄存器选择接收模式并使能模块。内部支持缓存 1byte 接收数据，软件通过读取寄存器 UST_STA 的 UST_BUF_VLD 标志判断缓冲区非空时即可读取数据寄存器 SPI_DATA 获得缓冲区的数据。

接收数据流程：软件配置好 SPI_CLK 分频比后，对寄存器 UST_DATA 写任意值启动数据接收。启动后输出 SPI 时钟，支持先收高位/先收低位，每接收完 1byte 数据则停止输出 SPI 时钟。接收到的数据会保存到缓冲区中，如果需要继续接收数据，软件需要及时读取缓冲区中的数据并再次对寄存器 UST_DATA 写入任意值启动接收。

标准模式在 MISO 端口接收数据，三线模式在 MISO 端口完成发送和接收。

13.3.8. TIMER 计时功能

TIMER 模式支持计数功能，通过软件配置模式寄存器选择 TIMER 模式，设置预分频时钟的分频比，以及计数周期寄存器 UST_BAUD，对寄存器 UST_CNT 写任意值触发开始计数。计数器数到寄存器 UST_PRD_BUF 的 TIMER_PRD 值时，寄存器 UST_STA 的 TIMER_DONE_PEND 置 1，计数器自动清零进行新一轮计数，根据寄存器 UST_CON0 的 TIMER_DONE_IE 状态决定是否产生中断。

另外，UST TIMER 模式支持与 TIMER0/1 同步启动计时，由寄存器 TMR_ALLCON 控制。

13.3.9. TIMER PWM 功能

TIMER 同时支持 4 路 PWM 输出, 4 路 PWM 共用一个 16 位计数器, 周期相同, 占空比可单独配置。配置占空比为 0xffff 时, PWM 恒输出 1, 配置占空比为 0x0 时, PWM 恒输出 0。

通过配置 PWM 的输出极性即寄存器 UST_CON0 的 PWMxPOL 和死区时间 PWM_DEAD_TIME, 可以使 4 路 PWM 形成两对带死区的互补 PWM。

13.3.10. TIMER PWM LED 功能

TIMER 能够发送两种不同占空比的 PWM 波, 从而控制外部集成 IC 智能 LED, 当 UST_MODE 配置成 TIMER 模式且 UST_CON0 的 PWM_LED_MODE 配置为 1 时该功能有效。PWM_LED_MODE 有效时, 寄存器 UST_CMP01、UST_CMP01_BUF、UST_CMP23、UST_CMP23_BUF 的功能有所改变, 详细请参考下面寄存器列表中的描述。

13.4. IO 映射

详细参考 [3.2. 管脚定义](#)

13.5. 模块框图与接口时序

13.5.1. 模块架构图

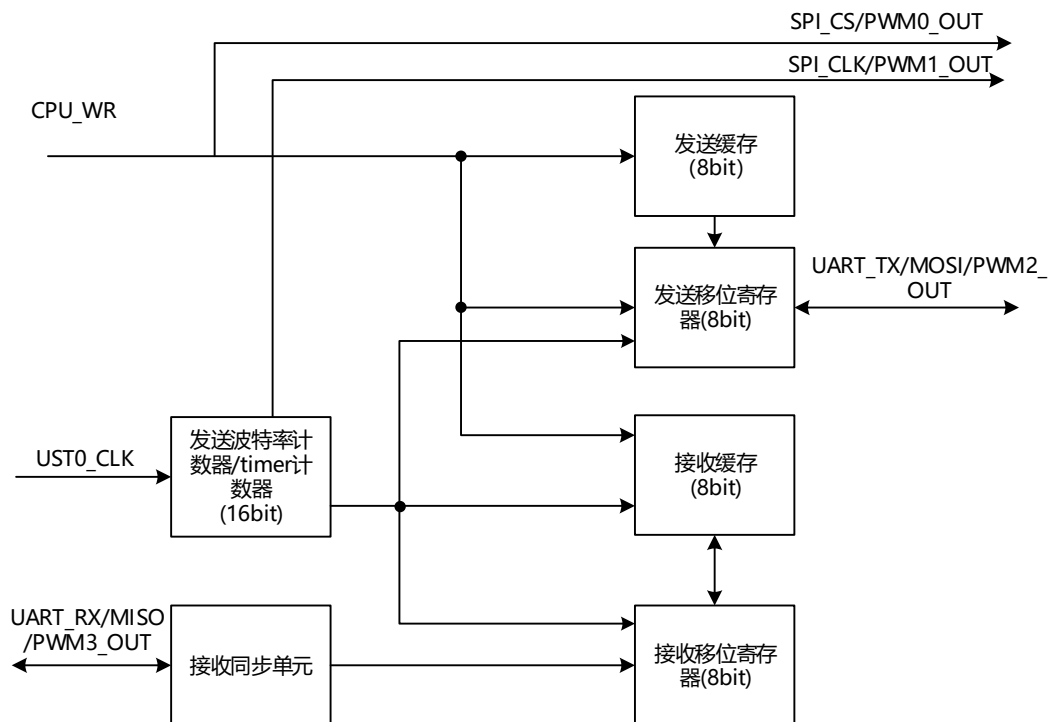


图 13-1 UST 模块架构图

13.5.2. UART 数据传输时序图

UART 支持 8bit 数据传输，无奇偶校验位。

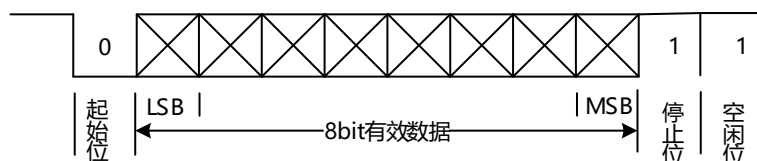


图 13-2 UART 数据传输时序图

13.5.3. SPI 数据传输时序图

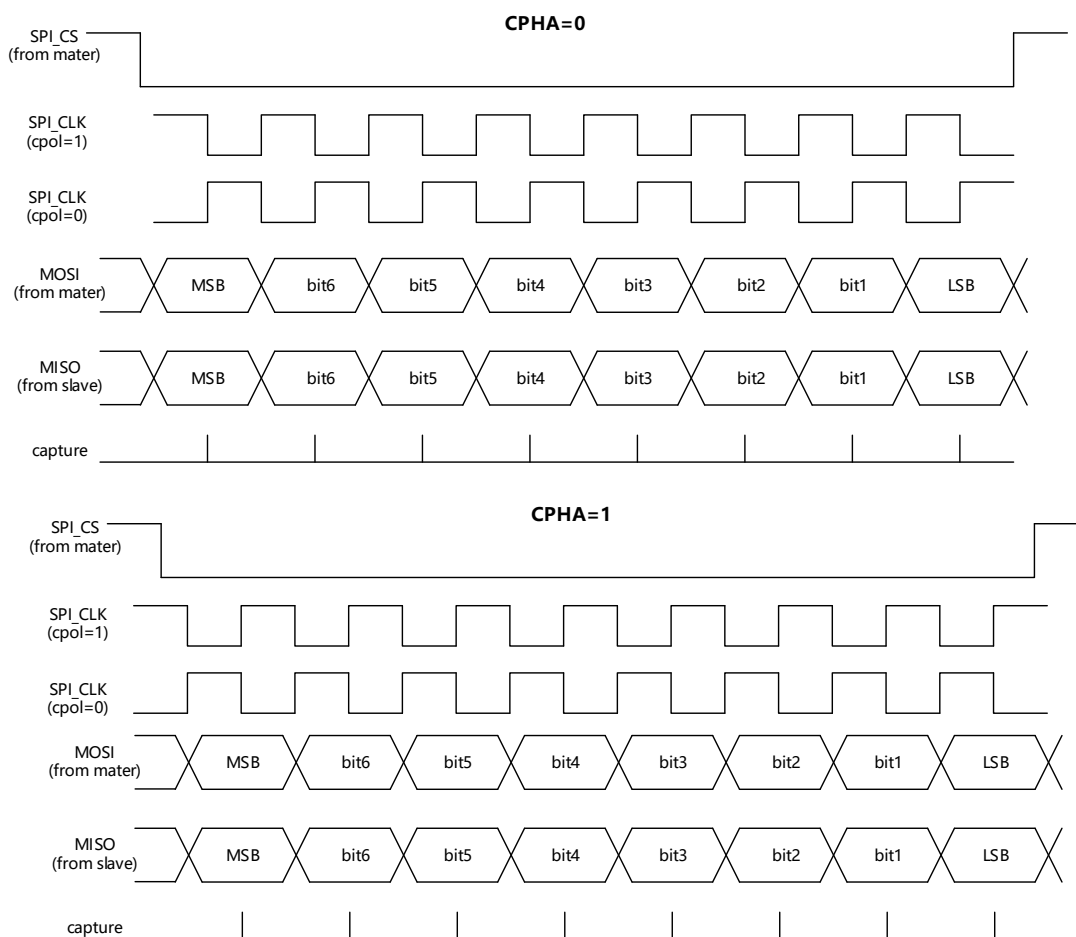


图 13-3 SPI 数据传输时序图

13.5.4. PWM 输出时序图

死区时间非零时，PWM1 和 PWM3 输出高电平的时间会比 PWM0 和 PWM2 的时间延迟 PWM_DEAD_TIME 个计数周期，所以，只要把 PWM0 和 PWM2 的占空比配置比 PWM1 和 PWM3 大 PWM_DEAD_TIME，同时配置 PWM0 与 PWM2 极性取反，即可使 PWM0 和 PWM1、PWM2 和 PWM3 形成两对互补 PWM。

互补 PWM 配置如下

- $PWM0_CMP = PWM1_CMP + PWM_DEAD_ZONE$
- $PWM2_CMP = PWM3_CMP + PWM_DEAD_ZONE$
- $PWM0POL = 1; PWM1POL = 0; PWM2POL = 1; PWM3POL = 0$

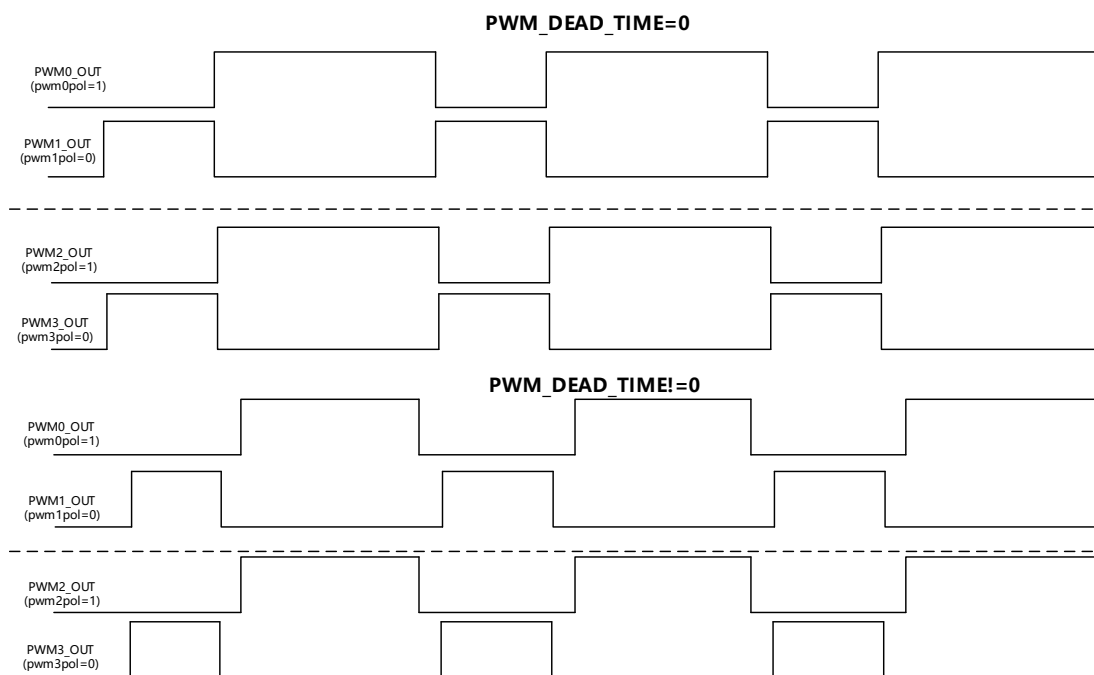


图 13-4 PWM 输出时序图

13.5.5. PWM_LED 模式下的 PWM0 输出时序图

寄存器 `PWM_LED_FRAME_SET` 设置每次发送多少帧数据，发完设定的帧数后定时器会进入复位状态，`PWM0` 输出低电平。寄存器 `PWM_LED_BIT_SET` 设置每一帧的位数，即每一帧的 PWM 周期个数，发完设定的位数代表发完一帧，此时 `PWM_BIT_STREAM_BUF` 中的数据会自动加载 `PWM_BIT_STREAM` 中。

每发完一帧，寄存器 `UST_STA` 中的 `TIMER_DONE_PEND` 会置 1，软件检测到该位为 1 时即可把下一帧数据写入到 `PWM_BIT_STREAM_BUF` 中。

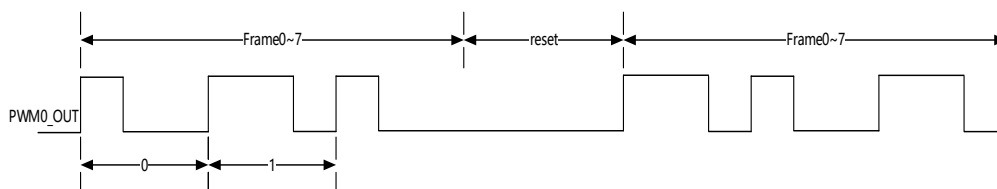


图 13-5 PWM_LED 模式下的 PWM0 输出时序图

13.6. 时钟与复位

13.6.1. 时钟介绍

UST 该模块时钟来源于系统时钟，可通过配置系统寄存中 `CLKCON2` 来使能时钟。

13.6.2. 复位介绍

UST 模块的复位源有两个，分别是系统复位和软件复位，软件复位可通过配置系统寄存器触发。

13.7. 寄存器描述

13.7.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
UST_MODE	0x02E0	32'h0	UST 模式控制寄存器
UST_CON0	0x02E4	32'h0	UST 控制寄存器 0
UST_DATA	0x02E8	32'h0	UST 数据寄存器
UST_BAUD	0x02EC	32'h0	UST 波特率寄存器
UST_PRD_BUF	0x02F0	32'h0	UST_TIMER 周期缓冲寄存器
UST_CMP01	0x02F4	32'h0	UST_PWM0/UST_PWM1 占空比
UST_CMP01_BUF	0x02F8	32'h0	UST_PWM0/UST_PWM1 占空比缓冲
UST_CMP23	0x02FC	32'h0	UST_PWM2/UST_PWM3 占空比
UST_CMP23_BUF	0x0300	32'h0	UST_PWM2/UST_PWM3 占空比缓冲
UST_CNT	0x0304	32'h0	UST 计数寄存器
UST_STA	0x0308	32'h0	UST 状态寄存器

13.7.2. 寄存器详细说明

13.7.2.1. 模式寄存器 (UST_MODE)

Width	Name	Reset	Property	Description
31:2	Reserved	-	-	-
1:0	UST_MODE	2'b0	RW	UST 模式选择 b00: Reserved b01: UART 模式 b10: SPI 模式 b11: TIMER 模式

13.7.2.2. UART 控制寄存器 (UST_CON0)

Width	Name	Reset	Property	Description
31:17	Reserved	-	-	-
16	UART_DONE_IE	1'b0	RW	UART 传输完成中断使能 0: 不使能 1: 使能
15:6	Reserved	-	-	-
5	UART_RX_TX	1'b0	RO	UART 接收/发送状态，默认为接收状态，软件向 UST_DATA 寄存器写入数据自动切换到发送状态 0: 接收状态 1: 发送状态
4:3	Reserved	-	-	-
2	UART_WIRE_MODE	1'b0	RW	UART 单线模式 0: 不使能 1: 使能
1:0	Reserved	-	-	-

13.7.2.3. SPI 控制寄存器 (UST_CON0)

Width	Name	Reset	Property	Description
31:17	Reserved	-	-	-
16	SPI_DONE_IE	1'b0	RW	SPI 传输完成中断使能 0: 不使能 1: 使能(SPI_DONE_PEND 为 1 时产生中断)
15:7	Reserved	-	-	-
6	SPI_LSBF	1'b0	RW	SPI 帧数据格式 0: 先传输高位 1: 先传输低位
5	SPI_RX_TX	1'b0	RW	SPI 接收/发送 0: 接收 1: 发送
4	SPI_CS	1'b0	RW	SPI CS 引脚输出电平极性 0: 低电平 1: 高电平
3	SPI_SYNC_EN	1'b0	RW	SPI 输入信号同步使能 0: 不使能 1: 使能
2	SPI_WIRE_MODE	1'b0	RW	SPI 通信模式 0: 普通模式(CS,CLK,IO0(MOSI),IO1(MISO)) 1: 3 线模式 (CS,CLK,IO0)
1:0	SPI_CPOL_CPHA	2'b0	RW	CPOL CPHA b00: 时钟 idle 为 0, 上升沿采样, 下降沿出数据 b01: 时钟 idle 为 0, 下降沿采样, 上升沿出数据 b10: 时钟 idle 为 1, 下降沿采样, 上升沿出数据 b11: 时钟 idle 为 1, 上升沿采样, 下降沿出数据

13.7.2.4. TIMER 控制寄存器 (UST_CON0)

Width	Name	Reset	Property	Description
31:17	Reserved	-	-	-
16	TIMER_DONE_IE	1'b0	RW	TIMER 完成中断使能 0: 不使能 1: 使能(TIMER_DONE_PEND 为 1 时产生中断)
15:14	Reserved	-	-	-
13:9	PWM_DEAD_TIME	5'b0	RW	互补 PWM 死区长度 0: 无死区 1: 死区为 1 个计数步长 2: 死区为 2 个计数步长 31: 死区为 31 个计数步长
8	PWM_LED_MODE	1'b0	RW	PWM_LED 控制模式 0: 不使能 1: 使能
7	PWM_UD_HOLD	1'b0	RW	PWM 占空比保持 0: 更新

				1: 保持 为了确保 PWM0/1/2/3 的占空比在同一个时刻切换, 可在更新占空比缓冲寄存器前保持原值。
6	PWM3_POL	1'b0	RW	PWM3 输出电平极性 0: 不取反 1: 取反
5	PWM2_POL	1'b0	RW	PWM2 输出电平极性 0: 不取反 1: 取反
4	PWM1_POL	1'b0	RW	PWM1 输出电平极性 0: 不取反 1: 取反
3	PWM0_POL	1'b0	RW	PWM0 输出电平极性 0: 不取反 1: 取反
2:0	TIMER_PSC	3'b0	RW	TIMER 时钟预分频 b000: 1 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

13.7.2.5. 数据寄存器 (UST_DATA)

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7:0	UART_DATA SPI_DATA	8'h0	RW	数据寄存器 写: 写入发送数据 读: 获取接收数据

13.7.2.6. 配置寄存器 (UST_BAUD)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	UART_BAUD SPI_BAUD TIMER_PRD	16'h0	RW	UART 波特率控制: 波特率=SYSCLK/(UART_BAUD+1) 注: 需配置 UART_BAUD >=6, 否则输入信号会被内部 滤波器滤掉。 SPI 波特率控制: SPI_CLK=SYSCLK/(2*(SPI_BAUD+1)) TIMER 计数周期控制 计数周期=TIMER_PRD+1

13.7.2.7. 配置寄存器 (UST_PRD_BUF)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	TIMER_PRD_BUF	16'h0	RW	TIMER_PRD 的缓冲寄存器 软件配置 TIMER_PRD 时, TIMER_PRD_BUF 会同时被配置; 定时器工作期间修改 TIMER_PRD_BUF 的值, 将在计数周期满时自动重载到 TIMER_PRD。

13.7.2.8. 配置寄存器 (UST_CMP01)

Width	Name	Reset	Property	Description
31:16	TIMER_CMP1	16'h0	RW	PWM1 占空比 当 PWM_LED_MODE 为 1 时, TIMER_CMP1 代表 PWM_BIT_STREAM 为 1 时的占空比
15:0	TIMER_CMP0	16'h0	RW	PWM0 占空比 当 PWM_LED_MODE 为 1 时, TIMER_CMP0 代表 PWM_BIT_STREAM 为 0 时的占空比

13.7.2.9. 配置寄存器 (UST_CMP01_BUF)

Width	Name	Reset	Property	Description
31:16	TIMER_CMP1_BUF	16'h0	RW	TIMER_CMP1 的缓冲寄存器 软件配置 TIMER_CMP1 时, TIMER_CMP1_BUF 会同时被配置; 定时器工作期间修改 TIMER_CMP1_BUF 的值, 将在计数周期满时自动重载到 TIMER_CMP1。
15:0	TIMER_CMP0_BUF	16'h0	RW	TIMER_CMP0 的缓冲寄存器 软件配置 TIMER_CMP0 时, TIMER_CMP0_BUF 会同时被配置; 定时器工作期间修改 TIMER_CMP0_BUF 的值, 将在计数周期满时自动重载到 TIMER_CMP0。

当 PWM_LED_MODE 为 1 时, 寄存器 UST_CMP01_BUF 的功能如下:

Width	Name	Reset	Property	Description
31:0	PWM_BIT_STREAM	32'h0	RW	PWM_LED 模式下的帧比特流寄存器, 每一位表示输出 PWM 一个周期内占空比的值: 0: 占空比的值等于 TIMER_CMP0 1: 占空比的值等于 TIMER_CMP1

13.7.2.10. 配置寄存器 (UST_CMP23)

Width	Name	Reset	Property	Description
31:16	TIMER_CMP3	16'h0	RW	PWM3 占空比
15:0	TIMER_CMP2	16'h0	RW	PWM2 占空比

当 PWM_LED_MODE 为 1 时, 寄存器 UST_CMP23 的功能如下:

Width	Name	Reset	Property	Description
31:0	PWM_BIT_STREAM_B	32'h0	RW	PWM_BIT_STREAM 的缓冲寄存器

	UF			当 PWM_BIT_STREAM 中的有效数据发送完毕时, PWM_BIT_STREAM_BUF 中的数据自动装载到 PWM_BIT_STREAM。
--	----	--	--	--

13.7.2.11. 配置寄存器 (UST_CMP23_BUF)

Width	Name	Reset	Property	Description
31:16	TIMER_CMP3_BUF	16'h0	RW	TIMER_CMP3 的缓冲寄存器 软件配置 TIMER_CMP3 时, TIMER_CMP3_BUF 会同时被配置; 定时器工作期间修改 TIMER_CMP3_BUF 的值, 将在计数周期满时自动重载到 TIMER_CMP3。
15:0	TIMER_CMP2_BUF	16'h0	RW	TIMER_CMP2 的缓冲寄存器 软件配置 TIMER_CMP2 时, TIMER_CMP2_BUF 会同时被配置; 定时器工作期间修改 TIMER_CMP2_BUF 的值, 将在计数周期满时自动重载到 TIMER_CMP2。

当 PWM_LED_MODE 为 1 时, 寄存器 UST_CMP23_BUF 的功能如下:

Width	Name	Reset	Property	Description
31:27	Reserved	-	-	-
26:24	PWM_LED_FRAME_CNT	3'h0	RO	PWM_LED 帧计数器, 显示当前发送到第几帧 PWM。
23:21	Reserved	-	-	-
20:16	PWM_LED_BIT_CNT	5'h0	RO	PWM_LED 位计数器, 显示当前 PWM 帧发送到第几位数据。
15:11	Reserved	-	-	-
10:8	PWM_LED_FRAME_SET	3'h0	RW	PWM 连续发送的帧数 0: 1 帧 1: 2 帧 ... 7: 8 帧
7:5	Reserved	-	-	-
4:0	PWM_LED_BIT_SET	5'h0	RW	每帧 PWM 发送的位数 0: 1 位 1: 2 位 ... 31: 32 位

13.7.2.12. 配置寄存器 (UST_CNT)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	TIMER_CNT	16'b0	RW	计数寄存器 读: TIMER 当前计数值 写: 空闲时写入任意值启动从 0 开始计数 工作时写入任意值停止计数并计数值清 0

13.7.2.13. 配置寄存器 (UST_STA)

Width	Name	Reset	Property	Description
31:17	Reserved	-	-	-
16	UART_DONE_PEND SPI_DONE_PEND TIMER_DONE_PEND	1'b0	RC	UST 完成标志 1.当 UART/SPI 发送完 1byte 且缓冲区为空, 或 UART/SPI 接收到 1byte 数据时该位为 1, 否则为 0。 2.TIMER 计数值等于 TIMER_PRD0 时该位为 1, 否则为 0。 3.PWM_LED_MODE 为 1 时, 发完一帧 PWM 时该位为 1。 读: 0: 未完成 1: 完成 写: 软件对该 bit 写 1 清 0, 写 0 无效。
15:4	Reserved	-	-	-
3	UST_BUF_OVR	1'b0	RC	UST 接收模式下, 当缓冲区中的数据未被读走又接收到新的数据时, 该标志置 1, 软件对该位写 1 清零。
2	UART_FERR	1'b0	RC	UART 帧错误标志 0: 没有帧错误 1: 出现帧错误
1	UST_BUF_STATUS	1'b0	RO	UST 缓冲区标志 0: 缓冲区空 1: 缓冲区满
0	UST_BUSY	1'b0	RO	UST 状态标志 0: 空闲 1: 工作

14. 定时器 (TIMER0/1/4)

14.1. 模块介绍

CIU32F011、CIU32F031 一共包含 3 个通用 TIMER 模块，该模块可用于计数，可输出 PWM 波，可捕获外部信号，获取外部信号的高/低电平宽度。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。

注：TIMERx 中的 x 表示第几个 TIMER。

14.2. 功能特点

16 位递增计数器

可编程预分频计数

支持外部 GPIO 或内部时钟作为计数时钟源

支持输入捕获功能

- 最多支持同时保存 2 个捕获事件
- 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

支持 PWM 输出

允许软件在任意时刻修改占空比和周期值，硬件自动在周期结束时重载

支持使用外部信号控制定时器和定时器互联的同步电路

14.3. 功能说明

14.3.1. 计数源选择

计数器时钟可由下列时钟源提供

系统时钟

内部高速 RC 的 2 分频 (HIRC_DIV2)

内部低速 RC 的 2 分频 (LIRC256K_DIV8_DIV2)

外部 GPIO

注：HIRC_DIV2 设置为计数时钟源配置仅适用于 TIMER0/1。

通过配置寄存器 TMRx_CON 的 TMR_INC_SEL 选择不同的计数源，配置 TMRx_CON 的 TMR_PSC 配置不同的分频系数。

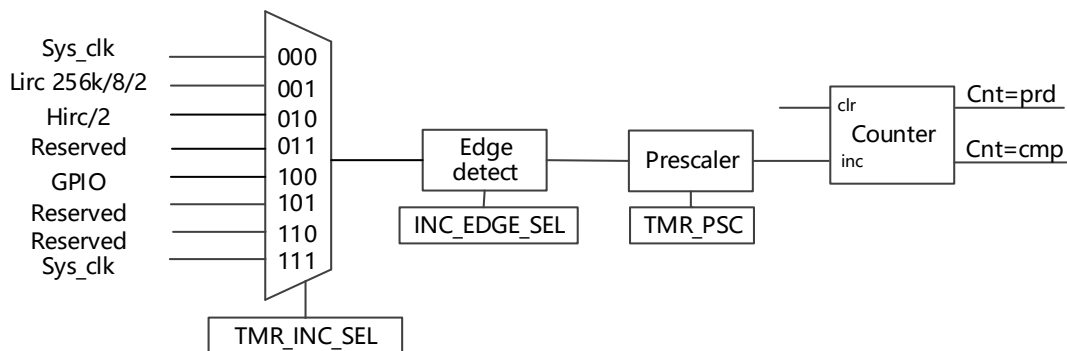


图 14-1 计数源选择框图

14.3.2. 输入捕获源

输入捕获，支持外部 GPIO 触发捕获事件。

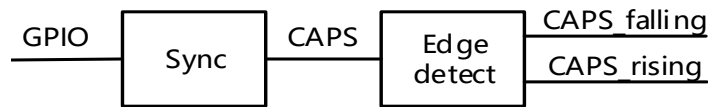


图 14-2 输入捕获源框图

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

通过配置寄存器 TMRx_CON 的 TMR_CAP_CNT 可以配置使用 1~2 个捕获寄存器，例如使用 2 个寄存器，则第 1 次捕获事件产生，捕获值保存在 TMRx_CAP1，第 2 次捕获事件产生，捕获值保存在 TMRx_CAP2，第 3 次捕获事件产生，捕获值保存在 TMRx_CAP1，第 4 次捕获事件产生，捕获值保存在 TMRx_CAP2，依次循环。

通过配置寄存器 TMRx_CON 的 TMR_CAP1_POL/TMR_CAP2_POL 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 TMR_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，寄存器 TMRx_FLAG 都会产生对应的捕获标志。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 16'hFFFF。当计数值达到 16'hFFFF，会产生溢出标志 TMR_OVF_FLAG，通过配置寄存器 TMRx_IE 可以产生溢出中断。

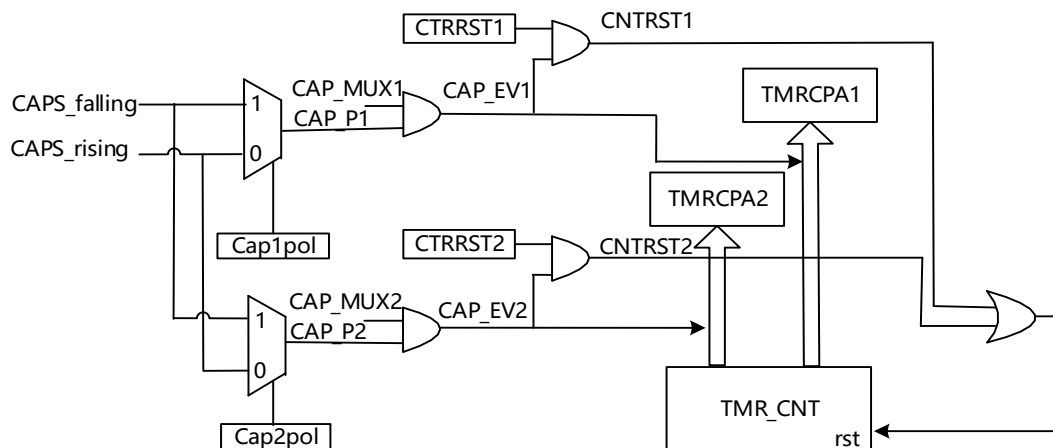


图 14-3 输入捕获模式框图

14.3.3. PWM 模式

PWM 工作模式可以产生一个由寄存器 TMRx_PRD0 确定周期，寄存器 TMRx_CMP0 确定占空比的信号。

当计数值达到寄存器 TMRx_CMP0 值或计数值达到 TMRx_PRD0 值时，会产生对应标志位，如果设置了对应的中断使能位，会产生中断。

选择 PWM 输出 IO 对应的复用功能，在寄存器 TMRx_PRD0 和 TMRx_CMP0 中配置所需的 PWM 周期和占空比，配置寄存器 TMRx_CON 的 TMR_MODE 为定时器 PWM 模式，然后启动 PWM 输出。这样，对应 IO 上就会产生期望的 PWM 波形信号，直到软件配置关闭 PWM 输出。

软件可以在 PWM 输出过程中的任意时刻修改占空比/周期影子寄存器的值，计数器在数满一个周期时才会把影子寄存器的值自动更新到实际使用的占空比/周期寄存器中，从而避免输出错误的 PWM 波形。其中，TMRx_SHADOW[31:16]为 CMP0 的影子寄存器，TMRx_SHADOW[15:0]为 PRD0 的影子寄存器。

14.3.4. 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。

14.3.5. 从模式

定时器能够在从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。

14.3.5.1. 复位模式

当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，即寄存器 TMRx_FLAG 的 TMR_SLAVE_FLAG 位被设置，根据寄存器 TMRx_IE 中的 SLAVE_IE 的设置，产生中断。

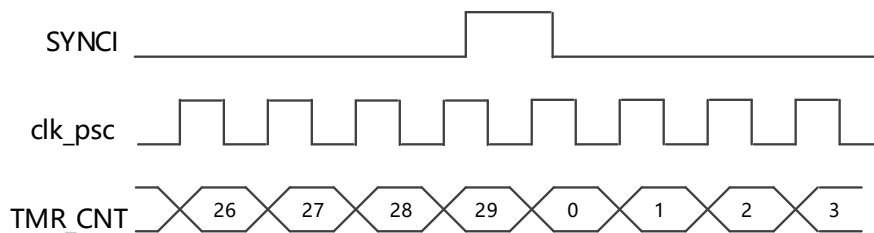


图 14-4 TIMER 复位模式波形图

14.3.5.2. 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，同时寄存器 TMRx_FLAG 的 TMR_SLAVE_FLAG 位被设置，并且根据寄存器 TMRx_IE 中的 TMR_SLAVE_IE 的设置，产生中断。

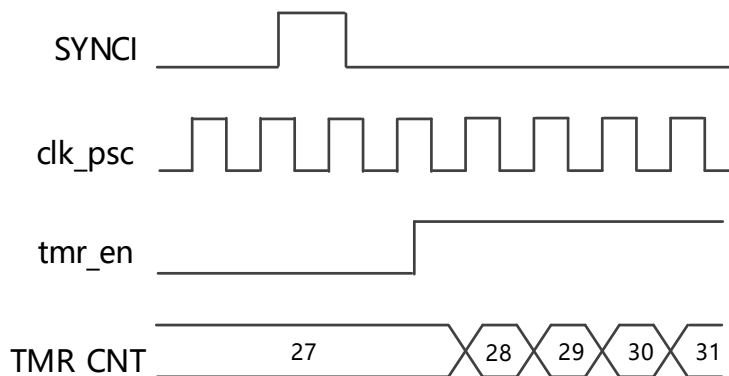


图 14-5 TIMER 触发模式波形图

14.3.5.3. 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 为低电平时，寄存器 TMRx_CNT 计数，高电平时，停止计数。

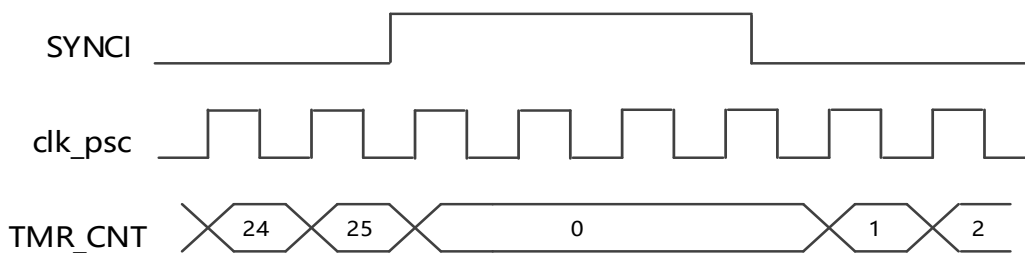


图 14-6 TIMER 门控模式波形图

14.3.6. 定时器同步

TIMER0 和 TIMER1 在内部相连，用于定时器同步或链接。

同步输出信号选择可选择

SYNCI

计数值等于 TMRx_PRD 时输出脉冲

计数值等于 TMRx_CMP 时输出脉冲

PWM

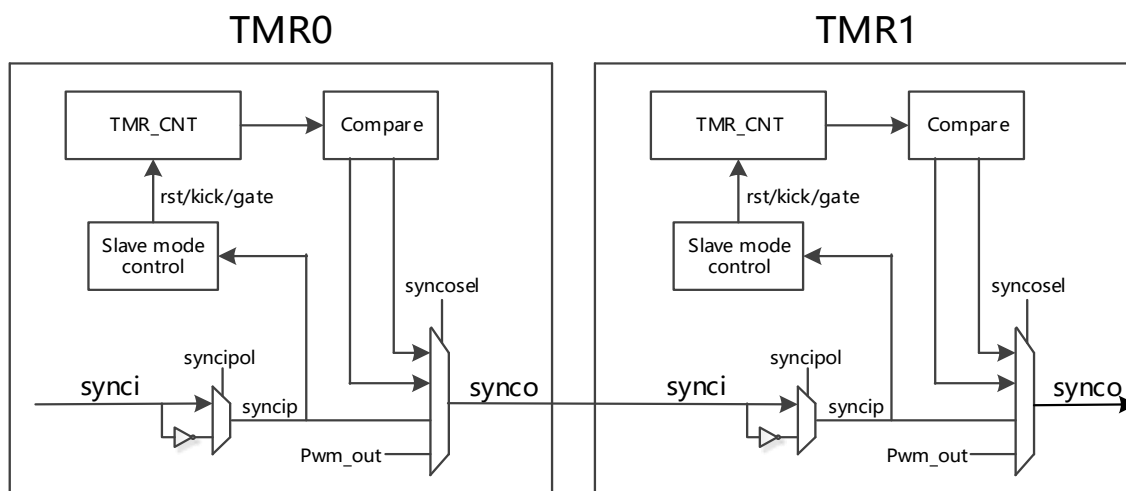


图 14-7 TIMER 级联 (GPIO->TIMER0->TIMER1) 框图

14.3.6.1. 多个定时器计数值同时清零

下面例子，把两个定时器的计数值同时清零，配置如下

配置寄存器 TMR0_CON 的 TMR_SYNCIPOL = 0。

配置寄存器 TMR0_CON 的 TMR_SYNCOSSEL = 2，直接把 TIMR0 的 SYNCI 连接到 TIMR1 的 SYNCI。

配置寄存器 TMR0_CON 的 TMR_SLAVE_MODE = 2，复位模式。

配置寄存器 TMR1_CON 的 TMR_SYNCIPOL = 0。

配置寄存器 TMR1_CON 的 TMR_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 TIMR0 和 TIMR1 的计数值。

14.3.6.2. 产生带载波的 PWM 信号

上一级 TIMER 输出低频 PWM，下一级 TIMER 配置成门控模式输出高频 PWM，具体步骤如下
配置 TIMR0 为 PWM 输出模式（低频）。

配置寄存器 TMR0_CON 的 TMR_SYNCOSSEL，选择 TIMR0 的 PWM 值输出到 TIMR1 的 SYNCI。

配置寄存器 TMR1_CON 的 TMR_SLAVE_MODE = 3，门控模式。

配置 TIMR1 为 PWM 输出模式。
使能 TIMR0 和 TIMR1。

14.4. IO 映射

详细参考 [3.2. 管脚定义](#)

14.5. 模块框图与接口时序

14.5.1. TIMER 模块架构图

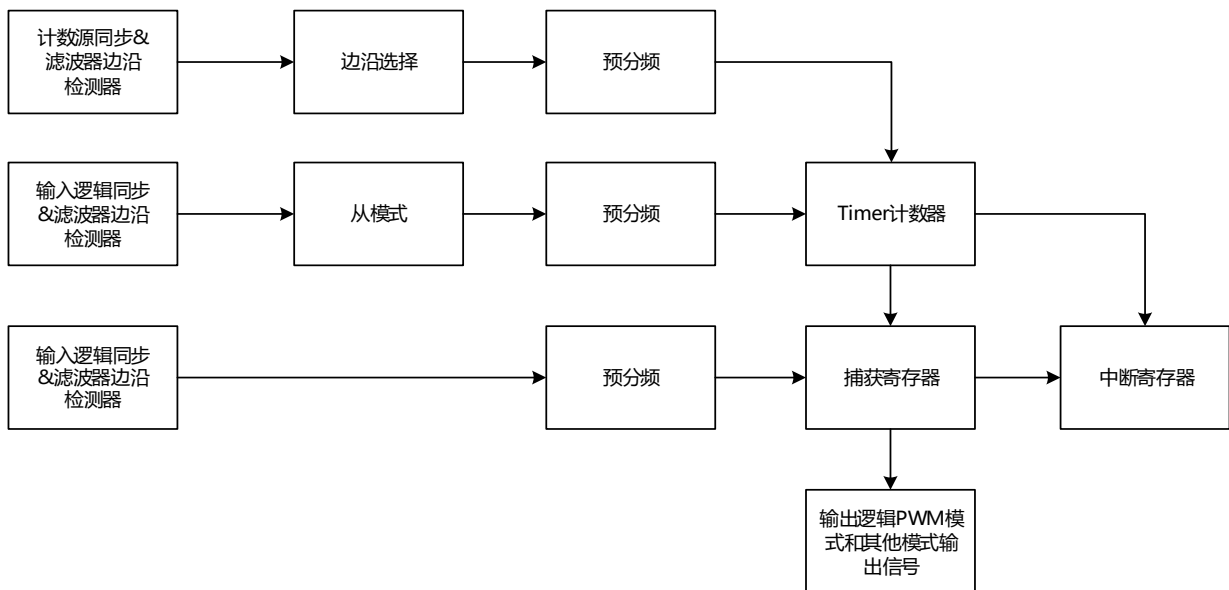


图 14-8 通用定时器结构框图

14.6. 时钟与复位

14.6.1. 时钟介绍

该模块时钟来源于系统时钟，可通过配置系统寄存中 CLKCON2 来使能时钟。

14.6.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，软件复位可通过配置系统寄存器触发。

14.7. 寄存器描述

14.7.1. TIMER0 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
TMR0_CON	0x01A0	32'h0	TIMER 控制寄存器
TMR0_EN	0x01A4	32'h0	TIMER 使能寄存器
TMR0_IE	0x01A8	32'h0	TIMER 中断寄存器
TMR0_CNT	0x01AC	32'h0	TIMER 计数寄存器
TMR0_FLAG	0x01B0	32'h0	TIMER 标志寄存器
TMR0_CLR	0x01B4	32'h0	TIMER 标志清除寄存器
TMR0_CAP1	0x01B8	32'hffff	TIMER 捕获 1 寄存器

TMR0_CAP2	0x01BC	32'h0	TIMER 捕获 2 寄存器
TMR0_SHADOW	0x01C0	32'hffff	TIMER 影子寄存器
TMR_ALLCON	0x01C8	32'h0	TIMER 同步寄存器

14.7.2. TIMER1 寄存器列表

Base address: 0x4002 0000

Name	Offset	Reset	Description
TMR1_CON	0x01E0	32'h0	TIMER 控制寄存器
TMR1_EN	0x01E4	32'h0	TIMER 使能寄存器
TMR1_IE	0x01E8	32'h0	TIMER 中断寄存器
TMR1_CNT	0x01EC	32'h0	TIMER 计数寄存器
TMR1_FLAG	0x01F0	32'h0	TIMER 标志寄存器
TMR1_CLR	0x01F4	32'h0	TIMER 标志清除寄存器
TMR1_CAP1	0x01F8	32'hffff	TIMER 捕获 1 寄存器
TMR1_CAP2	0x01FC	32'h0	TIMER 捕获 2 寄存器
TMR1_SHADOW	0x0200	32'hffff	TIMER 影子寄存器

14.7.3. TIMER4 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
TMR4_CON	0x0310	32'h0	TIMER 控制寄存器
TMR4_EN	0x0314	32'h0	TIMER 使能寄存器
TMR4_IE	0x0318	32'h0	TIMER 中断寄存器
TMR4_CNT	0x031C	32'h0	TIMER 计数寄存器
TMR4_FLAG	0x0320	32'h0	TIMER 标志寄存器
TMR4_CLR	0x0324	32'h0	TIMER 标志清除寄存器
TMR4_CAP1	0x0328	32'hffff	TIMER 捕获 1 寄存器
TMR4_CAP2	0x032C	32'h0	TIMER 捕获 2 寄存器
TMR4_SHADOW	0x0330	32'hffff	TIMER 影子寄存器

14.7.4. 寄存器详细说明

14.7.4.1. 控制寄存器 (TMRx_CON)

Width	Name	Reset	Property	Description
31:28	Reserved	-	-	-
27:26	INC_EDGE_SEL	2'b0	RW	计数源边沿选择,当计数源[TMR_INC_SEL]选择系统时钟时, 该配置无效。 b00: 上升沿计数 b01: 下降沿计数 b1x: 边沿计数
25	GATE_KEEP_CNT	1'b0	RW	门控模式下门控有效时, 寄存器 TMRx_CNT 值是否清零 0: TMR_CNT 值清零 1: TMR_CNT 值保持
24:23	Reserved	-	-	-
22	TMR_CAP2_POL	1'b0	RW	CAPTURE 事件 2 的触发边沿选择 0: 上升沿 1: 下降沿

21	TMR_CAP1_POL TMR_PWM_POL	1'b0	RW	CAPTURE 事件 1 的触发边沿选择 0: 上升沿 1: 下降沿 PWM 极性取反 0: 不取反 1: 取反
20:19	Reserved	-	-	-
18	TMR_CTRRST2	1'b0	RW	CAPTURE 事件 2 发送时, 寄存器 TMRx_CNT 值自动清零 0: 不使能 1: 使能
17	TMR_CTRRST1	1'b0	RW	CAPTURE 事件 1 发送时, 寄存器 TMRx_CNT 值自动清零 0: 不使能 1: 使能
16	Reserved	-	-	-
15	TMR_CAP_CNT	1'b0	RW	捕获寄存器使用选择 0: 使用寄存器 TMRx_CAP1 1: 使用寄存器 TMRx_CAP1、TMRx_CAP2
14:13	Reserved	-	-	-
12:11	TMR_SYNCOSSEL	2'b0	RW	定时器输出同步信号选择 b00: 计数值等于 TMRx_PRD0 值 b01: 计数值等于 TMRx_CMP0 值 b10: SYNCI b11: PWM
10	TMR_SYNCIPOL	1'b0	RW	定时器同步输入信号极性选择 0: 不取反 1: 取反
9:8	TMR_SLAVE_MODE	2'b0	RW	定时器从模式同步模式选择 b00: 保留 b01: 触发模式 b10: 复位模式 b11: 门控模式
7:5	TMR_PSC	3'b0	RW	定时器预分频设置 b000: 1 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	TMR_INC_SEL	3'b0	RW	定时器计数源选择位 b000: 系统时钟上升沿 b001: 内部 LIRC256K_DIV8 的 2 分频 b010: 内部 HIRC 的 2 分频时钟 注意: 仅 TIMER0/1 支持内部 HIRC 的 2 分频作为计数

				时钟源, TIMER4 此位域配置 b010 时为 Reserved。 b011: Reserved b100: 外部 GPIO b101: Reserved b110: Reserved b111: 系统时钟上升沿
1:0	TMR_MODE	2'b0	RW	定时器工作模式选择位: b00: 定时器计数模式 b01: 定时器 PWM 模式 b10: 定时器捕获模式 b11: 保留

14.7.4.2. 控制使能寄存器 (TMRx_EN)

Width	Name	Reset	Property	Description
31:1	Reserved	-	-	-
0	TMR_EN	1'b0	RW	定时器使能信号 0: 不使能 1: 使能

14.7.4.3. 中断使能寄存器 (TMRx_IE)

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	TMR_SLAVE_IE	1'b0	RW	从模式的触发模式/复位模式中断使能 0: 不使能 1: 使能
6	TMR_CMP_IE	1'b0	RW	计数值等于寄存器 TMRx_CMP0 值时, 中断使能 0: 不使能 1: 使能
5	TMR_PRD_IE	1'b0	RW	计数值等于寄存器 TMRx_PRD0 值时, 中断使能 0: 不使能 1: 使能
4	TMR_OVF_IE	1'b0	RW	计数值溢出 (16'hFFFF) 时, 中断使能 0: 不使能 1: 使能
3:2	Reserved	-	-	-
1	TMR_CAP2_IE	1'b0	RW	CAPTURE 事件 2 发生时, 中断使能 0: 不使能 1: 使能
0	TMR_CAP1_IE	1'b0	RW	CAPTURE 事件 1 发生时, 中断使能 0: 不使能 1: 使能

14.7.4.4. 计数寄存器 (TMRx_CNT)

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

14.7.4.5. 标志寄存器 (TMRx_FLAG)

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	TMR_SLAVE_FLAG	1'b0	RO	从模式的触发模式/复位模式发生标志
6	TMR_CMP_FLAG	1'b0	RO	计数值等于寄存器 TMRx_CMP0 值时标志, 仅在计数模式、PWM 模式时有效
5	TMR_PRD_FLAG	1'b0	RO	计数值等于寄存器 TMRx_PRD0 值时标志, 仅在计数模式、PWM 模式时有效
4	TMR_OVF_FLAG	1'b0	RO	计数值溢出 (16'hFFFF) 发生标志
3:2	Reserved	-	-	-
1	TMR_CAP2_FLAG	1'b0	RO	CAPTURE 事件 2 发生时标志
0	TMR_CAP1_FLAG	1'b0	RO	CAPTURE 事件 1 发生时标志

14.7.4.6. 清除寄存器 (TMRx_CLR)

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	TMR_SLAVE_CLR	1'b0	WO	从模式的触发模式/复位模式标志清除 0: 无效 1: 清除
6	TMR_CMP_CLR	1'b0	WO	计数值等于寄存器 TMRx_CMP0 值标志清除 0: 无效 1: 清除
5	TMR_PRD_CLR	1'b0	WO	计数值等于寄存器 TMRx_PRD0 值标志清除 0: 无效 1: 清除
4	TMR_OVF_CLR	1'b0	WO	计数值溢出标志清除 0: 无效 1: 清除
3:2	Reserved	-	-	-
1	TMR_CAP2_CLR	1'b0	WO	CAPTURE 事件 2 发生标志清除 0: 无效 1: 清除
0	TMR_CAP1_CLR	1'b0	WO	CAPTURE 事件 1 发生标志清除 0: 无效 1: 清除

14.7.4.7. 捕获寄存器 1/周期寄存器 (TMRx_CAP1/ PRD0)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:0	TMR_CAP1/PRD0	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

14.7.4.8. 捕获寄存器 2/占空比寄存器 (TMRx_CAP2/CMP0)

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

-	Reserved	-	-	-
15:0	TMR_CAP2/CMP0	16'b0	RW	捕获模式: 捕获寄存器 2 定时模式、PWM 模式: 占空比寄存器

14.7.4.9. 周期占空比影子寄存器 (TMRx_SHADOW)

Width	Name	Reset	Property	Description
31:16	TMR_CMP_SHADOW	16'h0	RW	定时模式、PWM 模式: 占空比影子寄存器 在 PWM 模式下会在周期结束时自动把影子寄存器加载到占空比寄存器
15:0	TMR_PRD_SHADOW	16'hffff	RW	定时模式、PWM 模式: 计数周期影子寄存器 在 PWM 模式下会在周期结束时自动把影子寄存器加载到周期寄存器

14.7.4.10. 同步寄存器 (TMR_ALLCON)

Width	Name	Reset	Property	Description
31:11	Reserved	-	-	-
10	UST0_TMR_SYCN	1'b0	WO	UST0_TIMER 计数值清零
9	TMR1_SYCN	1'b0	WO	TIMER1 计数值清零
8	TMR0_SYCN	1'b0	WO	TIMER0 计数值清零
7:3	Reserved	-	-	-
2	UST0_TMR_KST	1'b0	WO	UST0_TIMRE 启动计数
1	TMR1_KST	1'b0	WO	TIMER1 启动计数
0	TMR0_KST	1'b0	WO	TIMER0 启动计数

15. 数模转换模块 (ADC)

15.1. 模块介绍

该模块是一个 12 位的逐次逼近式的 ADC 控制器，ADC 支持单通道扫描和多通道扫描。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以及其他片内外设启动 (TIMER 触发启动)。

15.2. 功能特点

- 支持 DMA 保存采样数据
- 最高 12 位可编程分辨率的 SARADC，多达 26 路外部输入通道
- 通道采样时间可软件配置
- 支持最高采样率为 266KSPS
- 可选多个 ADC 转换开始条件
 - 软件启动
 - 外部触发启动 (TIMER)

15.3. 功能说明

15.3.1. ADC 开关控制

通过设置寄存器 ADC_CFG 的 EN 和 LDO_EN 位可给 ADC 上电。当第一次设置 EN 和 LDO_EN 位时，它将 ADC 从断电状态下唤醒。ADC 上电延迟一段时间后 ($>=10\mu s$)，设置采样通道之后，设置寄存器 ADC_CR 的 KST 位开始进行转换。软件可通过清除 KST 位停止转换，或等所有通道转换结束自动清除。设置 EN 和 LDO_EN 为 0 可置于断电模式。

15.3.2. 通道选择

包含 26 路外部输入通道。每个外部输入通道都有独立的使能位，可通过寄存器 ADC_CHS 的对应位来配置使能。

15.3.3. 单通道采样模式

在单通道采样模式下，ADC 转换器只对使能的通道采样一次，并把结果保存在寄存器 ADC_DATA 中，操作步骤如下

根据需要采样的通道配置寄存器 ADC_CHS

转换完成后，寄存器 ADC_STA 中的 DONE 标志位置 1，如果使能了中断，该标志置 1 时会进入中断函数。

软件可轮询 DONE 标志位为 1 或者在中断函数中读取寄存器 ADC_DATA 获得采样值

15.3.4. 多通道采样模式

在多通道采样模式下，ADC 转换器对寄存器 ADC_CHS 中使能的通道依次扫描，可通过配置寄存器 ADC_CR 的 SCAN_DIR 控制扫描的顺序方向，操作步骤如下

根据需要采样的通道配置寄存器 ADC_CHS。

配置存储采样数据的 SRAM 起始地址。

使能寄存器 ADC_CR 的 DMA_EN 使 ADC 工作在 DMA 模式下。

软件或外部触发寄存器 ADC_CR 的 KST 开始转换，外部触发可以通过软件配置触发延时，ADC 转换顺序按照通道序号默认从小到大。

每路 ADC 转换完成后，ADC 转换数值将有序装载到相应的 SRAM 地址中。

当所有通道转换都完成后产生 DMA 完成标志位，如果使能了中断，则产生中断。

只要 KST 位保持为 1，则转换未结束，当 KST 位被清 0，ADC 转换停止，ADC 转换器进入空闲状态。

15.3.5. DMA 请求

ADC 转换器最近一次转换的结果会保存在寄存器 ADC_DATA 中，所以在单通道采样模式下不需要使能 DMA 功能。在多通道采样模式下，使能 DMA 功能并配置初始存放地址，所有扫描通道的结果通过 DMA 传输顺序存放对应 SRAM 地址中，并且由寄存器 ADC_DMACNT 指示已存放的数据个数。

15.3.6. 采样频率设置

ADC 的时钟 ADC_CLK 由系统时钟分频得到，分频系数可通过设置寄存器 ADC_CFG 的 PSC 位来确定。

由于每次转换固定为 15 个 ADC_CLK 周期，则 ADC 采样率计算如下：

ADC 采样率 = $F_{pclk} / \text{ADC 分频系数} / 15$ ， ($F_{pclk} / \text{ADC 分频系数} \leq 4\text{Mhz}$)，理论上 ADC 支持最快采样率为 266KSPS。

15.3.7. 外部触发转换

ADC 转换可以由定时器触发，如果设置了寄存器 ADC_CR 的 TRG_EN 位，就可以使用定时器触发转换。通过设置 TRG_SEL 位可以选择触发源。

具体的外部触发源选择情况，可以参考 ADC 控制寄存器 ADC_CR 位[2]TRG_EN 和[5:4]TRG_SEL 的描述。

外部触发可设置延时控制，具体参考 ADC_CR[21:19]的 TRG_DLY 的描述。

15.4. IO 映射

详细参考 [3.2. 管脚定义](#)

15.5. 模块框图与接口时序

15.5.1. ADC 模块接口结构图

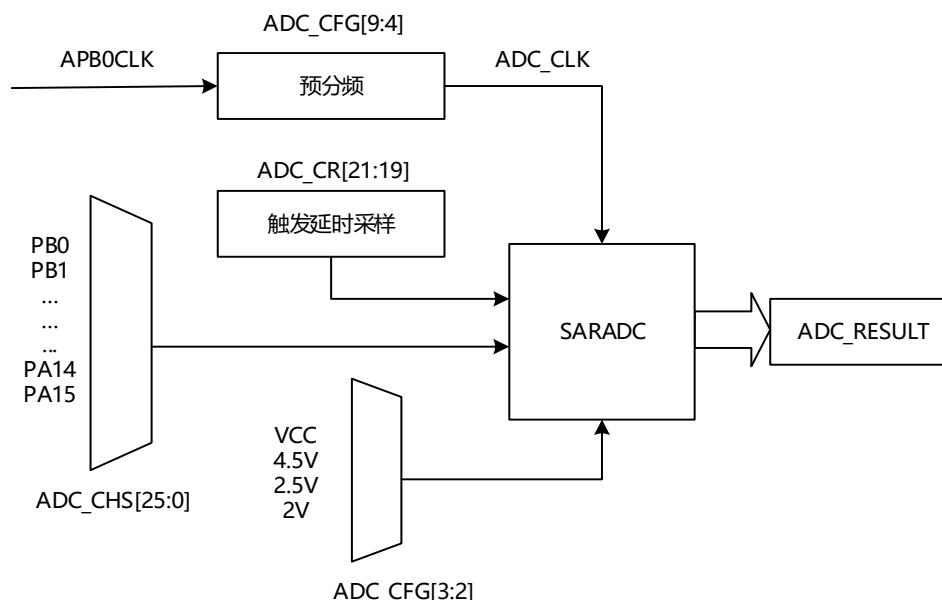


图 15-1 ADC 模块接口架构图

15.6. 时钟与复位

15.6.1. 时钟介绍

本模块的时钟来源只有系统时钟，在 sleep 或 stop 模式下，本模块时钟停止。

15.6.2. 复位介绍

该模块的复位源有两个，分别是系统复位和软件复位，其中软件复位由系统寄存器 SYSCON0[19]控制。

15.7. 寄存器描述

15.7.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
ADC_CFG	0x0140	32'hfc	ADC 配置寄存器
ADC_CR	0x0144	32'h10000000	ADC 控制寄存器
ADC_CHS	0x0148	32'h0	ADC 通道寄存器
ADC_STA	0x014C	32'h1f0	ADC 状态寄存器
ADC_DATA	0x0150	32'h0	ADC 数据寄存器
ADC_DMAADDR	0x0154	32'h0	DMA 地址寄存器
ADC_DMACNT	0x0158	32'h0	DMA 已完成 DMA 数据个数
ADC_DMALEN	0x015C	32'h0	DMA 长度寄存器

15.7.2. 寄存器详细说明

15.7.2.1. 配置寄存器 (ADC_CFG)

Width	Name	Reset	Property	Description
31:24	Reserved	-	-	-
22:21	Reserved	-	-	-
19	LDO_EN	1'b0	RW	ADC 模块内部 LDO 使能 0: 不使能 1: 使能
18:17	Reserved	-	-	-
16:13	D2DCYC	4'b0	RW	两次连续采样之间的间隔时间 (n+1) * ADCCLK
12:10	Reserved	-	-	-
9:4	PSC	6'hf	RW	ADC 时钟预分频, ADC 时钟不得超过 4MHz 0: 2 分频 其它: (n+1)分频
3:2	VREF_SEL	2'b11	RW	ADC 参考电压选择 00: 2V 01: 2.5V 10: 4.5V 11: VCC
1	VOL_SEL	1'b0	RW	ADC 供电电压选择 0: 5V 供电 1: 3.3V 供电
0	EN	1'b0	RW	ADC 使能 0: 不使能 1: 使能

15.7.2.2. 控制寄存器 (ADC_CR)

Width	Name	Reset	Property	Description
31:29	Reserved	-	-	-
28	CAL_EN	1'b1	RW	硬件数据校准使能 0: 不使能 1: 使能
27	SW_RST	1'b0	WO	软件复位模块内部状态机 0: 无效 1: 有效
26	DATASIGN	1'b0	RW	数据扩张位选择,影响寄存器 ADC_DATA 的最高 4 位或最低 4 位
25	OVR_IE	1'b0	RW	数据过载覆盖中断使能 0: 不使能 1: 使能
24	Reserved	-	-	-
23	DMAF_IE	1'b0	RW	DMA 全满中断使能 0: 不使能 1: 使能
22	DMAH_IE	1'b0	RW	DMA 半满中断使能 0: 不使能 1: 使能
21:19	TRG_DLY	3'b0	RW	外部触发延时采样 在触发信号产生后, 延时 N 个 PCLK 的时钟周期再开始采样 0: 不延时 1: 4 个周期 2: 16 个周期 3: 32 个周期 4: 64 个周期 5: 128 个周期 6: 256 个周期 7: 512 个周期
18:17	Reserved	-	-	-
16	SCAN_DIR	1'b0	RW	扫描通道方向 0: ADC 通道选择寄存器按从低到高扫描 1: ADC 通道选择寄存器按从高到低扫描
15:12	Reserved	-	-	-
11	ALIGN	1'b0	RW	数据对齐 0: 左对齐 1: 右对齐
10:9	Reserved	-	-	-
8	KST	1'b0	RW	软件对该位写 1, ADC 转换开始; 转换完成后, KST 将被硬件自动清除。 读: 0: ADC 空闲中 1: ADC 转换中

7:6	Reserved	-	-	-
5:4	TRG_SEL	2'b0	RW	外部触发源选择: b00: TIMER0 触发 b01: TIMER1 触发 b1x: TIMER4 触发
3	DMA_EN	1'b0	RW	DMA 使能 0: 不使能 1: 使能
2	TRG_EN	1'b0	RW	外部触发 ADC 转换使能 0: 不使能 1: 使能
1	Reserved	-	-	-
0	IE	1'b0	RW	ADC 中断使能位。如果 IE 置位, 标志位 DONE 为 1 时产生中断。 0: 不使能 1: 使能

15.7.2.3. 通道寄存器 (ADC_CHS)

Width	Name	Reset	Property	Description
31:26	Reserved	-	-	-
25:0	CHx_EN	26'h0	RW	输入通道使能 0: 不使能 1: 使能 CHxEN<0>--PB0 CHxEN<1>--PB1 CHxEN<2>--PB2 CHxEN<3>--PB3 CHxEN<4>--PB4 CHxEN<5>--PB5 CHxEN<6>--PB6 CHxEN<7>--PB7 CHxEN<8>--PB8 CHxEN<9>--PB9 CHxEN<10>--PA0 CHxEN<11>--PA1 CHxEN<12>--PA2 CHxEN<13>--PA3 CHxEN<14>--PA4 CHxEN<15>--PA5 CHxEN<16>--PA6 CHxEN<17>--PA7 CHxEN<18>--PA8 CHxEN<19>--PA9 CHxEN<20>--PA10 CHxEN<21>--PA11 CHxEN<22>--PA12 CHxEN<23>--PA13

				CHxEN<24>-PA14 CHxEN<25>-PA15
--	--	--	--	----------------------------------

15.7.2.4. 状态寄存器 (ADC_STA)

Width	Name	Reset	Property	Description
31:12	Reserved	-	-	-
11	DMAF_FLAG	1'b0	RC	DMA 全满标志, 软件可写 1 清 0 0: 非全满 1: 全满
10	DMAH_FLAG	1'b0	RC	DMA 半满标志, 软件可写 1 清 0 0: 非半满 1: 半满
9	Reserved	-	-	-
8:4	CHANNEL	5'h1f	RO	当前转换通道 读: BUSY = 1 时, 表示进行转换中的通道
3	Reserved	-	-	-
2	BUSY	1'b0	RO	ADC 转换器状态标志 0: 空闲 1: 忙碌
1	Reserved	-	-	-
0	DONE	1'b0	RC	ADC 转换结束标志位, 该位由硬件在所有通道采样结束时设置 1, 软件读取 ADC_DATA 寄存器时该位会被清 0, 软件也可以对该位写 1 清 0 0: ADC 转换未完成 1: ADC 转换完成

15.7.2.5. 数据寄存器 (ADC_DATA)

Width	Name	Reset	Property	Description
31:24	Reserved	-	-	-
23	VALID	1'b0	RO	转换结果有效标志位, 启动采样或读取寄存器 ADC_DATA 时该位清零 0: 有效 1: 无效
22	OVR_FLAG	1'b0	RO	数据过载覆盖标志位, 软件读寄存器 ADC_DATA 后, 该位自动清 0; 新的转换结果转载至寄存器之前, 若 ADC_DATA[15:0]数据没有被读取, 该位将被置 1。 0: DATA 为最近一次转换结果 1: DATA 上一次数据被覆盖
21	Reserved	-	-	-
20:16	DATA_CHANNEL	5'h0	RO	显示当前数据所对应的通道
15:0	DATA	16'h0	RO	12 位 ADC 转换结果 根据设置左对齐或者右对齐

15.7.2.6. DMA 地址寄存器 (ADC_DMAADDR)

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

15.7.2.7. DMA 个数寄存器 (ADC_DMACNT)

Width	Name	Reset	Property	Description
31:10	Reserved	-	-	-
9:0	DMA_CNT	10'h0	RO	当前已完成的 DMA 数据个数

15.7.2.8. DMA 长度寄存器 (ADC_DMALEN)

Width	Name	Reset	Property	Description
31:10	Reserved	-	-	-
9:0	DMA_LEN	10'h0	RW	DMA 缓冲区长度: $N=n+1$ 个 ADC 数据, 当接受数据的个数达到缓冲区长度后, 数据会循环存储到缓冲区的初始地址。

16. 触摸按键 (TK)

16.1. 模块介绍

CIU32F011、CIU32F031 集成了触摸感应控制模块，可应用于智能家电等需要用到触摸按键功能的产品方案。

16.2. 功能特点

- 支持 26 路按键扫描
- 支持单按键连续扫描 1~4 次，每次扫描频率可单独设置。
- 支持并联扫描模式
- 支持低功耗硬件自动扫描唤醒

16.3. 功能说明

16.3.1. 扫描方式

对触摸按键采用半自动扫描方式，通过软件初始化扫描的按键 IO，然后启动扫描，硬件会自动对所有使能的按键按顺序逐一扫描，并且把所有采样值存到预先定义好的缓冲区中。扫描完成时，软件可通过查询该标志或者在中断中读取按键采样值，并进行相应的按键判断处理，然后启动下一轮扫描。

16.3.2. 并联模式

区别于正常扫描方式，并联扫描模式下软件每启动一次扫描，硬件会把所有使能的按键当作一个按键，扫描过程中，会把所有使能的按键通道同时使能，形成并联状态。

16.4. 模块框图

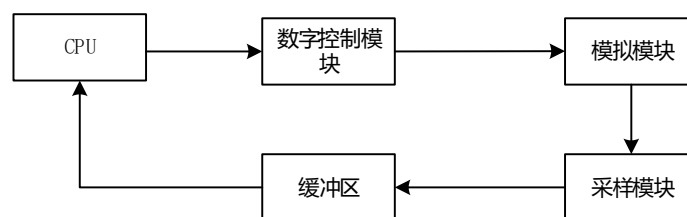


图 16-1 TK 模块接口架构图

17. LED 显示控制

17.1. 模块介绍

CIU32F011、CIU32F031 支持 8 个大电流 IO，可以用于软件配置共阴 LED、矩阵 LED。可以配置恒流驱动，抑制 LED 闪烁，增加电流稳定性。

17.2. 功能特点

- 支持 8 个 150mA 漏电流 IO
- 支持软件推共阴 LED，大电流 IO 用作 COM
- 支持软件配置正反推增加驱动 LED 数量
- 支持软件配置恒流驱动 LED 显示

17.3. 功能说明

17.3.1. 推共阴 LED

大电流 IO 用做 COM 口，普通 IO 用做 SEG 口，最大支持 8COM*18SEG 组合。

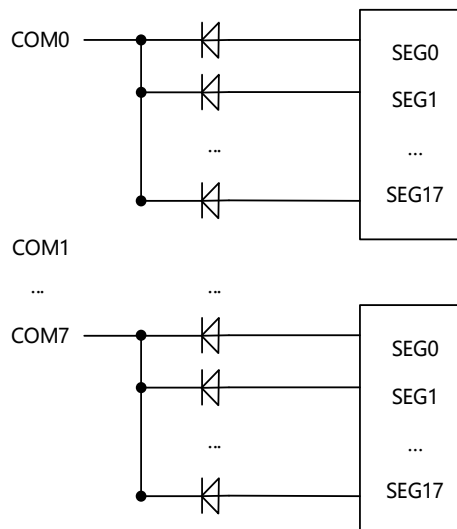


图 17-1 共阴 LED 接口图

17.3.2. 推矩阵 LED

大电流 IO 有 3 个状态，可以驱动 LED 灯的两个 IO 脚，控制多种矩阵 LED，例如 4*3 组合

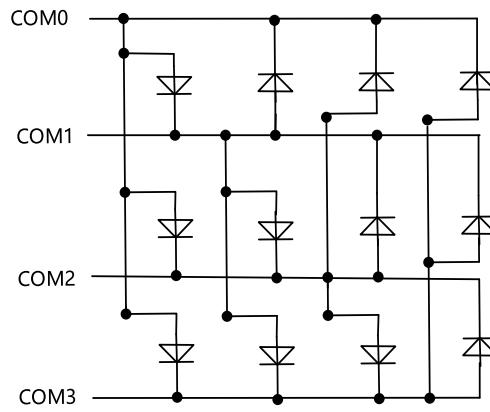


图 17-2 4*3LED 矩阵接口图

18. LCD 驱动模块 (LCD)

18.1. 模块介绍

CIU32F011、CIU32F031 内部集成了一个 LCD 段码屏驱动模块，可用于驱动家电及其它电子产品上的 LCD 段码屏，配置操作简单，可加快软件方案的研发速度。

18.2. 功能特点

- 支持 4COMx18SEG
- 支持 row 模式和 frame 模式
- 支持刷新率可配置
- 支持 duty: 1/1,1/2,1/3 和 1/4.
- 支持 bias: 1/2 和 1/3
- 支持 double buffer，软件可随时修改显示内容
- 支持死区控制

18.3. 功能说明

18.3.1. LCD 扫描控制

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：

Frame mode 与 Row mode 模式扫描频率相同的算法

$$\text{Freq_frame} = \text{Freq_lcd_clk} / (\text{LCD_CLK_DIV}[7:0] * 1/\text{duty} + \text{LCD_CLK_DIV}[15:8])$$
。（duty=1/COM 数量）

应注意的是，无论是 Frame 模式还是 Row 模式，分频控制寄存器 LCD_DIV_CON 所配置的分频数均是每个 com 口高低电平之和。

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。

18.3.2. 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 四种模式。

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。

18.4. IO 映射

详细参考 [3.2. 管脚定义](#)

18.5. 模块框图与接口时序

18.5.1. 模块架构图

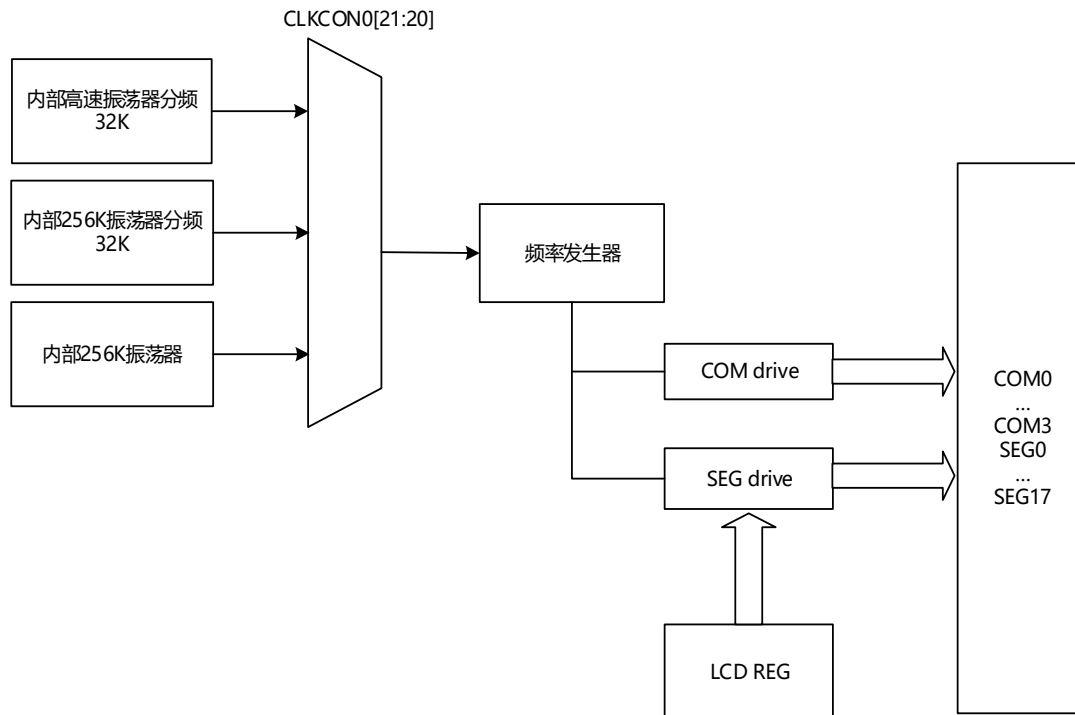


图 18-1 LCD 模块架构图

18.5.2. 1/2 DUTY 1/2BIAS Row Mode 时序图

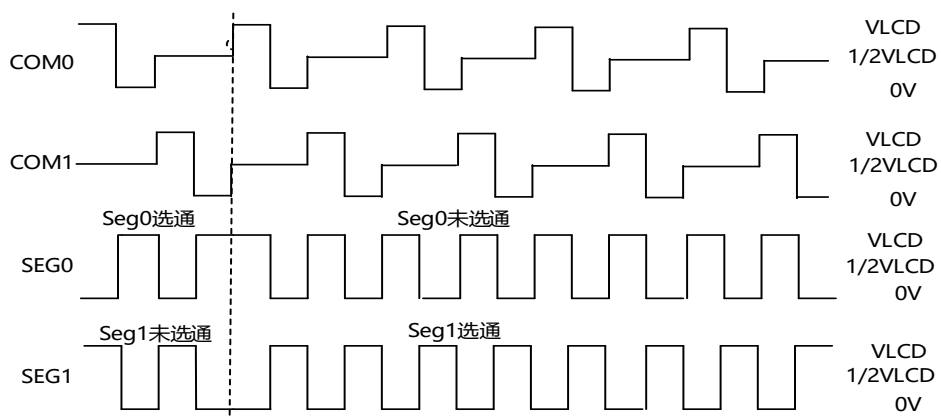


图 18-2 LCD 1/2 占空比 1/2 电压偏置 Row Mode 时序图

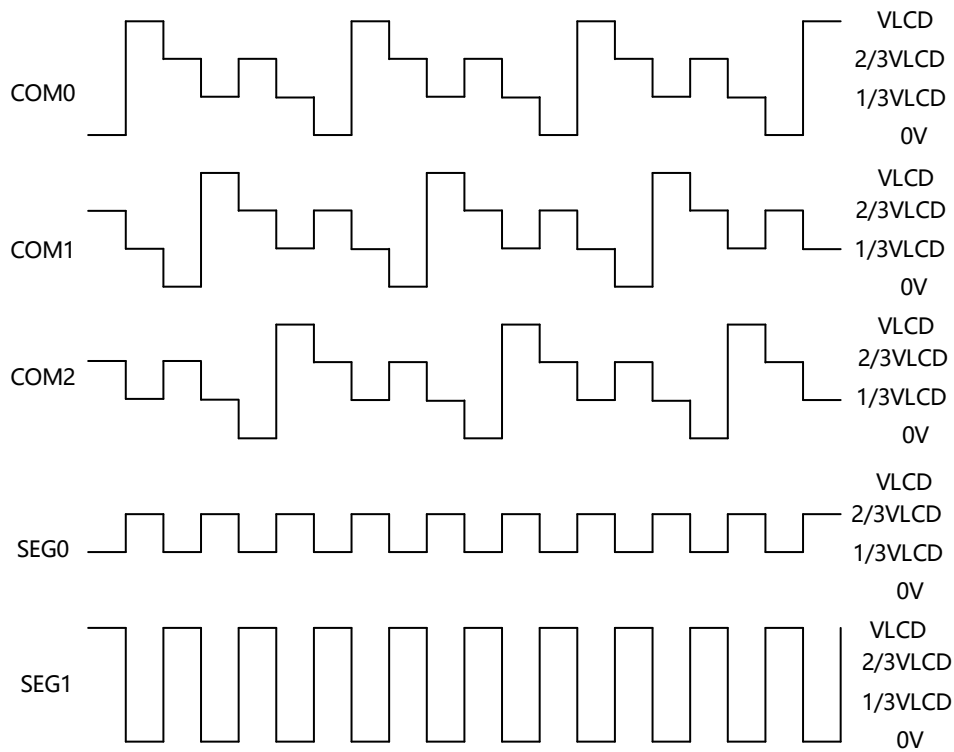
18.5.3. 1/3 DUTY 1/3BIAS Row Mode 时序图


图 18-3 LCD 1/3 占空比 1/3 电压偏置 Row Mode 时序图

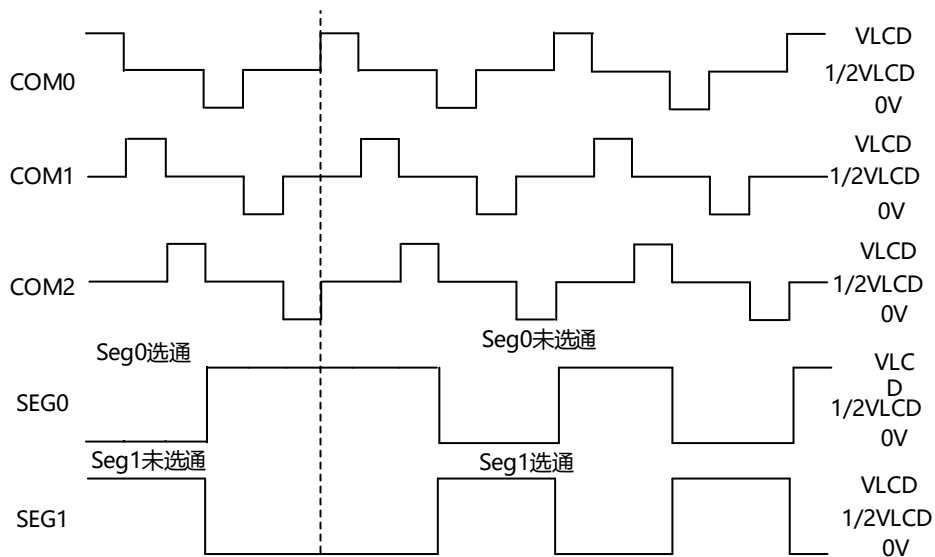
18.5.4. 1/3 DUTY 1/2BIAS Frame Mode 时序图


图 18-4 LCD 1/3 占空比 1/2 电压偏置 Frame Mode 时序图

18.5.5. 1/3 DUTY 1/3BIAS Frame Mode 时序图

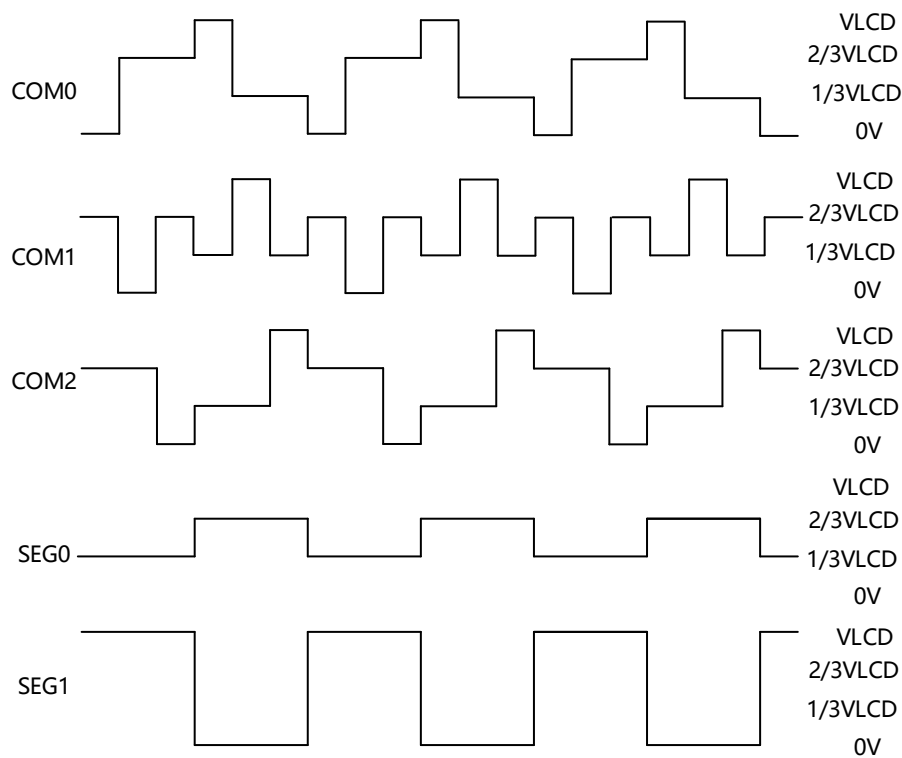


图 18-5 LCD 1/3 占空比 1/3 电压偏置 Frame Mode 时序图

18.5.6. 死区控制时序图

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[16:8] 控制死区时间为 LCD_DEAD_ZONE+1 个 LCD_CLK 时钟周期数。

18.5.6.1. Row Mode 死区控制时序图

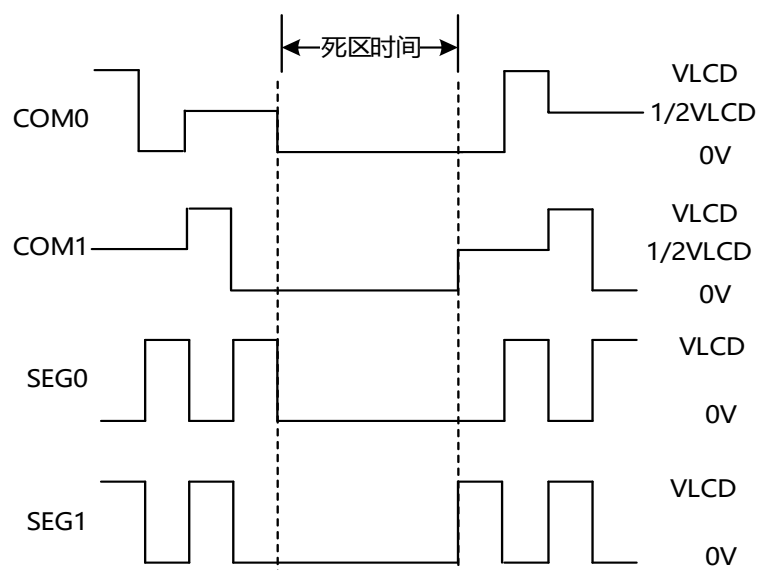


图 18-6 LCD Row Mode 死区控制时序图

18.5.6.2. Frame Mode 死区控制时序图

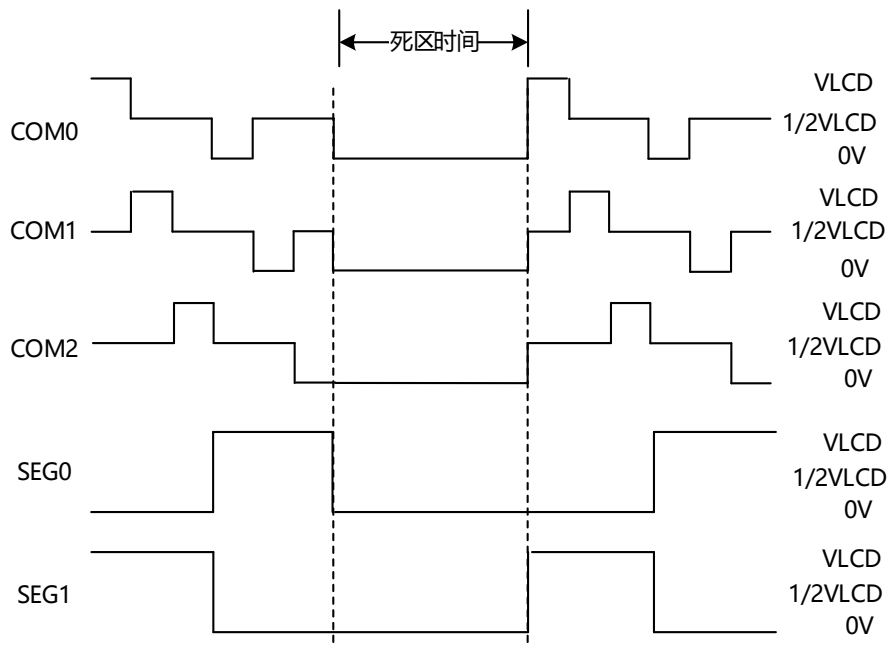


图 18-7 LCD Frame Mode 死区控制时序图

18.6. 时钟与复位

18.6.1. 时钟介绍

LCD 驱动模块有两个时钟: 系统时钟 (LCD_PCLK) 用于寄存器配置和低频工作时钟 LCD_CLK (32K/256K)

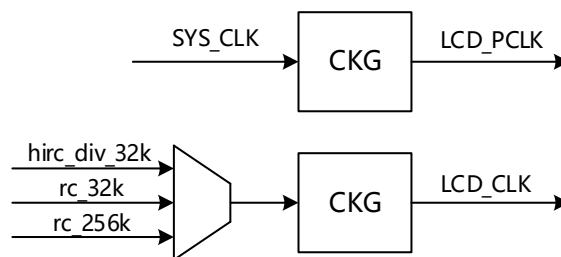


图 18-8 LCD 时钟信号结构图

18.6.2. 复位介绍

该模块的复位源有两个, 分别是系统复位和软件复位, 软件复位可通过配置系统寄存器触发。

18.7. 寄存器描述

18.7.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
LCD_CON	0x03D0	32'h0	LCD 控制寄存器
LCD_COM_NUM	0x03D4	32'h0	LCD COM 数量配置寄存器
LCD_REFRESH	0x03D8	32'h0	LCD 刷新控制寄存器
LCD_DIV_CON	0x03DC	32'h0	LCD 分频控制寄存器

LCD_COM_SEG_E NA	0x03E0	32'h0	LCD 端口使能控制寄存器
LCD_COM0_SEG	0x03E4	32'h0	LCD COM0 数据配置寄存器
LCD_COM1_SEG	0x03E8	32'h0	LCD COM1 数据配置寄存器
LCD_COM2_SEG	0x03EC	32'h0	LCD COM2 数据配置寄存器
LCD_COM3_SEG	0x03F0	32'h0	LCD COM3 数据配置寄存器
LCD_STA	0x03F4	32'h0	LCD 状态寄存器

18.7.2. 寄存器详细说明

18.7.2.1. 控制寄存器 (LCD_CON)

Width	Name	Reset	Property	Description
31:27	Reserved	-	-	-
26:23	LCD_VSEL	4'h0	RW	Vlcd 电压选择 $V_{lcd} = (17 + lcd_vsel) / 32 * vcc$
22	LCD_AEN	1'b0	RW	LCD 模拟驱动使能 1: 使能 0: 不使能
21:19	LCD_IO_EDGE_DRV	3'h0	RW	LCD IO 电平跳变阶段的驱动能力
18:16	LCD_IO_DRV	3'h0	RW	LCD IO 电平保持阶段的驱动能力
15:7	Reserved	-	-	-
3	LCD_SCAN_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
2	FRAME_MODE	1'b0	RW	COM 动态扫描模式: 0: ROM 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
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 使能控制 0: 不使能 1: 使能

18.7.2.2. 控制寄存器 (LCD_COM_NUM)

Width	Name	Reset	Property	Description
31:2	Reserved	-	-	-
1:0	LCD_COM_NUM	2'b0	RW	LCD COM 数量配置: 0: COM=1 1: COM=2 2: COM=3 3: COM=4

18.7.2.3. 控制寄存器 (LCD_REFRESH)

Width	Name	Reset	Property	Description
31:1	Reserved	-	-	-

0	LCD_REFRESH_PEND	1'b0	RW	LCD 显示更新控制位 (软件置 1, 硬件清 0) : Write: 1: 请求更新显示数据 0: 无效 Read: 1: 更新未完成, 等待完成再做下一次请求。 0: 更新已完成, 可以发起下一次请求。
---	------------------	------	----	---

18.7.2.4. 控制寄存器 (LCD_DIV_CON)

Width	Name	Reset	Property	Description
31:16	Reserved	-	-	-
15:8	LCD_DEAD_ZONE	8'h0	RW	LCD 死区时间控制: 0:255: 死区时间为 LCD_DEAD_ZONE+1 个 LCD_CLK 时钟周期
7:0	LCD_CLK_DIV	8'h0	RW	每个 COM 占用的时间控制: 0:255: 每个 COM 使能 LCD_CLK_DIV+1 个 LCD_CLK 时钟周期

18.7.2.5. 控制寄存器 (LCD_COM_SEG_ENA)

Width	Name	Reset	Property	Description
31:30	Reserved	-	-	-
29:22	LCD_COM_EN	8'h0	RW	8 个 COM 口使能控制 LCD_COM_EN[7:0]对应 com7~0 1: 使能 0: 不使能 注意: 8 个 com 口最多只能选择 4 个。
21:0	LCD_SEG_EN	22'h0	RW	22 个 SEG 口使能控制 LCDSEG_EN[21:0]对应 seg21~0 1: 使能 0: 不使能

18.7.2.6. 控制寄存器 (LCD_COM0_SEG)

Width	Name	Reset	Property	Description
31:22	Reserved	-	-	-
21:0	LCD_COM0_SEG	22'h0	RW	COM0 对应的 22 位 SEG 数据

18.7.2.7. 控制寄存器 (LCD_COM1_SEG)

Width	Name	Reset	Property	Description
31:22	Reserved	-	-	-
21:0	LCD_COM1_SEG	22'h0	RW	COM1 对应的 22 位 SEG 数据

18.7.2.8. 控制寄存器 (LCD_COM2_SEG)

Width	Name	Reset	Property	Description
31:22	Reserved	-	-	-

21:0	LCD_COM2_SEG	22'h0	RW	COM2 对应的 22 位 SEG 数据
------	--------------	-------	----	----------------------

18.7.2.9. 控制寄存器 (LCD_COM3_SEG)

Width	Name	Reset	Property	Description
31:22	Reserved	-	-	-
21:0	LCD_COM3_SEG	22'h0	RW	COM3 对应的 22 位 SEG 数据

18.7.2.10. 控制寄存器 (LCD_STA)

Width	Name	Reset	Property	Description
31:1	Reserved	-	-	-
0	SCAN_DONE_PEND	1'b0	RC	帧扫描完成中断标志 0: 未完成 1: 完成 软件可写 1 清 0

19. 看门狗 (WDT)

19.1. 模块介绍

CIU32F011、CIU32F031 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 and 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。

19.2. 功能特点

- 分频系数可配置
- 可工作在中断模式或复位模式
- 可定时唤醒系统

19.3. 模块框图

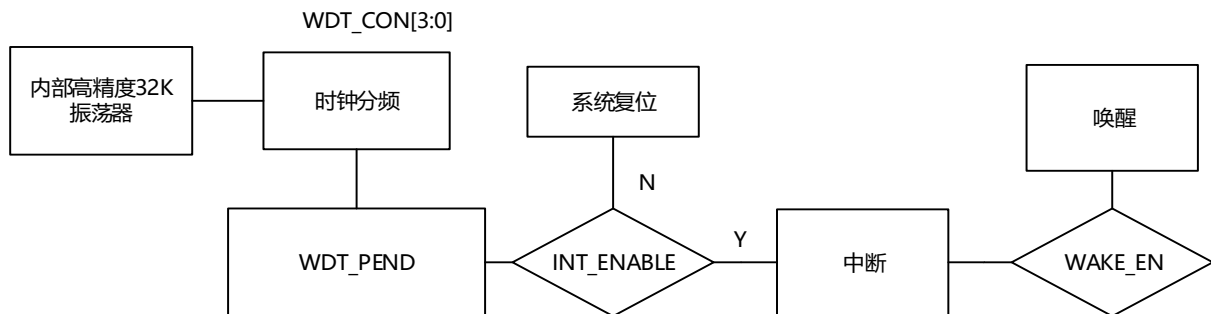


图 19-1 看门狗模块框图

19.4. 时钟与复位

19.4.1. 时钟介绍

该模块有两个时钟源，分别是系统时钟和看门狗时钟 `wdt_clk`，其中系统时钟用于配置模块寄存器，`wdt_clk` 为内部 LIRC_256K 分频得到的 32KHz 时钟，用于看门狗计数。

19.4.2. 复位介绍

该模块有两个复位源，分别是系统复位和看门狗模块复位，其中看门狗模块复位由外部 IO 或者掉电复位触发。

19.5. 寄存器描述

19.5.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
WDT_CON	0x0190	32'h18	WDT 控制寄存器
WDT_KEY	0x0194	32'h0	WDT 密钥寄存器

19.5.2. 寄存器详细说明

19.5.2.1. WDT_CON

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	IE	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	看门狗使能状态 0: 看门狗关闭 1: 看门狗使能
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

19.5.2.2. WDT_KEY

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) 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
--	--	--	--	--

20. 系统寄存器 (SYSTEM_REG)

20.1. 模块介绍

该模块用于对系统寄存器配置，部分系统寄存器会在上电过程中根据 FLASH 的系统配置区数据是否通过 CRC 校验，自动把配置参数装载到系统寄存器中。系统寄存器控制整个芯片系统的时钟使能、时钟选择、复位控制、唤醒控制和低功耗控制等。

20.2. 寄存器描述

20.2.1. 寄存器列表

Base address: 0x4002_0000

Name	Offset	Reset	Description
SYS_KEY	0x0000	32'h1	系统控制寄存器秘钥
SYS_CON0	0x0004	32'h065e9041	系统控制寄存器 0
SYS_CON1	0x0008	32'h300	系统控制寄存器 1
SYS_CON2	0x000C	32'h0	系统控制寄存器 2
SYS_CON3	0x0010	32'h0	系统控制寄存器 3
CLKCON0	0x0024	32'h0	时钟控制寄存器 0
CLKCON1	0x0028	32'h0	时钟控制寄存器 1
CLKCON2	0x002C	32'h0726e644	时钟控制寄存器 2
CLKCON3	0x0030	32'h80020240	时钟控制寄存器 3
SYS_ERR	0x0048	32'h0	系统错误记录寄存器
WKUP_CON	0x004C	32'h0	系统唤醒控制寄存器
LP_CON	0x0050	32'h20	低功耗控制寄存器
CHIPID_DCN	0x005C	32'h7	CHIPID 和 DCN 寄存器
MODE	0x0060	32'h0	系统启动模式寄存器
PMUCON0	0x0064	32'h11177800	PMU 控制寄存器
RPCON	0x0068	32'h0	复位控制寄存器
PMU_BK	0x006C	32'h10	PMU 备份寄存器

20.2.2. 寄存器详细说明

20.2.2.1. 控制寄存器 (SYS_KEY)

Width	Name	Reset	Property	Description
31:0	SYS_KEY	32'h1	RW	系统寄存器写使能 写: 0x3fac87e4: 使能系统寄存器写权限 others : 关闭系统寄存器写权限 读: 0: 系统寄存器写权限关闭 1: 系统寄存器写权限开启

20.2.2.2. 控制寄存器 (SYS_CON0)

Width	Name	Reset	Property	Description
31	FAST_RST_EN	1'b0	RW	快速释放复位 0: 不使能

				1: 使能
30	SLEEP_GOON_EN	1'b0	RW	系统唤醒时复位 0: 复位 1: 不复位
29:27	SLEEP_DLY_CNT	3'b0	RW	IO 唤醒时延时 SLEEP_DLY_CNT 个系统时钟周期, 再唤醒系统。 (如果睡眠时系统跑 256K, 该配置应 >=2)
26	DBS_SOFT_RST	1'b1	RW	IO 消抖电路软件复位控制 0: 复位 1: 完成复位
25	CRC_SOFT_RST	1'b1	RW	CRC 计算单元软件复位控制 0: 复位 1: 完成复位
24:23	Reserved	-	-	-
22	UST0_SOFT_RST	1'b1	RW	UST0 软件复位控制 0: 复位 1: 完成复位
21	Reserved	-	-	-
20	GPIO_SOFT_RST	1'b1	RW	GPIO 相关寄存器软件复位控制 0: 复位 1: 完成复位
19	ADC_SOFT_RST	1'b1	RW	ADC 控制模块软件复位控制 0: 复位 1: 完成复位
18	UART1_SOFT_RST	1'b1	RW	UART1 软件复位控制 0: 复位 1: 完成复位
17	UART0_SOFT_RST	1'b1	RW	UART0 软件复位控制 0: 复位 1: 完成复位
16	Reserved	-	-	-
15	SPI_IIC0_SOFT_RST	1'b1	RW	SPI_IIC0 软件复位控制 0: 复位 1: 完成复位
14:13	Reserved	-	-	-
12	LCD_SOFT_RST	1'b1	RW	LCD 软件复位控制 0: 复位 1: 完成复位
11:7	Reserved	-	-	-
6	TIMER_SOFT_RST	1'b1	RW	TIMERx 软件复位控制 0: 复位 1: 完成复位
5:1	Reserved	-	-	-
0	TK_SOFT_RST	1'h1	R/W	TK 软件复位控制 0: 复位 1: 完成复位

20.2.2.3. 控制寄存器 (SYS_CON1)

Width	Name	Reset	Property	Description
31	LED_AEN	1'b0	RW	LED 功能控制 0: 不使能 1: 使能
30	LED_ISELX	1'b0	RW	LED 电流乘 1.5 倍 0: 不使能 1: 使能
29	LED_ISELD	1'b0	RW	LED 电流除 2 0: 不使能 1: 使能
28:20	Reserved	-	-	-
19	UART1_SYS_EN1	1'b0	RW	PA15(SWD IO)选为 UART1_TX 功能 0: 不使能 1: 使能 (使用时需要把 SWD 功能关闭)
18	UART1_SYS_EN0	1'b0	RW	PA14(SWD IO)选为 UART1_RX 功能 0: 不使能 1: 使能 (使用时需要把 SWD 功能关闭)
17	CP_MCLR_IO_SEL	1'b0	RW	CP 模式下使能 MCLR 功能 0: 不使能 1: 使能
15:14	Reserved	-	-	-
13	WDT_LP_GATE_EN	1'b0	RW	看门狗进入 sleep/stop 模式时自动关闭时钟 0: 不使能 1: 使能
12	DEBUG_EN	1'b0	RW	如果配置成 0, 在 DEBUG(连接 SWD)模式下停止 sleep 和 stopclk 功能, 在超级模式中还会停止 wdt_reset 功能 0: 停止 1: 允许
11	INT_REMAP_EN	1'b0	RW	把中断入口从 FLASH 映射到 SRAM 0-192Bytes 0: 不映射 1: 映射
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	CP_MODE_SYSCLOCK_EN	1'b1	RW	在 CP 模式中系统时钟选择 0: 由 CLKCON0[1:0]选择 1: 系统时钟选择 IO 时钟
8	SWD_EN	1'b1	RW	SWD 使能控制 0: 不使能 1: 使能
7	LVDVCC_WKUP_EN	1'b0	RW	LVDVCC 中断模式下, 唤醒系统 0: 不使能

				1: 使能
5	SYS_ERR_RESP_EN	1'b0	RW	系统访问超出 SRAM 地址范围时, 返回出错信号给 CPU 0: 不使能 1: 使能
4	SYS_ERR_INT_EN	1'b0	RW	系统总线访问越界地址或时钟出错时触发 NMI 中断 0: 不使能 1: 使能
3:2	Reserved	-	-	-
1	NMI_INT_EN	1'b0	RW	PB5 触发 NMI 中断使能 0: 不使能 1: 使能
0	LOCKUP_EN	1'b0	RW	两次访问越界地址时触发系统复位 0: 不使能 1: 使能 (需要 SYS_ERR_RESP_EN 为 1)

20.2.2.4. 控制寄存器 (SYS_CON2)

Width	Name	Reset	Property	Description
31:30	Reserved	-	-	-
29:16	PB_DEB_EN	14'h0	RW	PB 消抖控制 0: 不使能 1: 使能
15:0	PA_DEB_EN	16'h0	RW	PA 消抖控制 0: 不使能 1: 使能

20.2.2.5. 控制寄存器 (SYS_CON3)

Width	Name	Reset	Property	Description
30	PMU_HPXCPCFB	1'b0	RW	内部 LDO 配置选项 0: 带反馈 1: 不带反馈
29:16	PB_CC	14'h0	RW	PB 电流模式, 驱动 LED 时使用 0: 不使能 1: 使能
15:0	PA_CC	16'h0	RW	PA 电流模式, 驱动 LED 时使用 0: 不使能 1: 使能

20.2.2.6. 控制寄存器 (CLKCON0)

Width	Name	Reset	Property	Description
31:22	Reserved	-	-	-
21:20	LCD_CLK_SEL	2'h0	RW	LCD 时钟选择 b00: HIRC_DIV_32K b01: LIRC_256K_DIV8 (32KHz) b10: Reserved b11: LIRC_256K
19:18	Reserved	-	-	-

17:16	LVD_DEB_CLK_SEL	2'h0	RW	LVD 消抖时钟选择 b00: SYSCLK b01: HIRC_CLK b10: HIRC_DIV2_CLK b11: LIRC_256K
15:8	Reserved	-	-	-
7:6	GPIO_DEB_CLK_SEL	2'h0	RW	GPIO 消抖时钟选择 b00: HIRC_CLK b01: LIRC_256K_DIV8 (32KHz) b10: SYSCLK b11: LIRC 256K
5:4	Reserved	-	-	-
1:0	SYSCLK_SEL	2'h0	RW	系统时钟选择 b00: LIRC_256K b1x: HIRC_CLK

20.2.2.7. 控制寄存器 (CLKCON1)

Width	Name	Reset	Property	Description
26:24	Reserved	-	-	-
23:19	TK_HCLK_DIV	5'h0	RW	TK_HCLK 高频采样时钟分频 0: 1 分频 1: 2 分频 ... 30: 31 分频 31: 时钟停止
18:6	Reserved	-	-	-
5:0	SYSCLK_DIV	6'h0	RW	系统时钟分频 0: 1 分频 1: 2 分频 ... 62: 63 分频 63: 时钟停止

20.2.2.8. 控制寄存器 (CLKCON2)

Width	Name	Reset	Property	Description
31:30	Reserved	-	-	-
27	Reserved	-	-	-
26	LCD_CLK_EN	1'b1	RW	LCD 时钟使能 0: 不使能 1: 使能
25	CRC_CLK_EN	1'b1	RW	CRC 时钟使能 0: 不使能 1: 使能
24	FLASH_MEM_CLK_EN	1'b1	RW	FLASH 擦除/烧写时钟使能 0: 不使能 1: 使能

23:22	Reserved	-	-	-
21	UST0_CLK_EN	1'b1	RW	UST0 时钟使能 0: 不使能 1: 使能
20:19	Reserved	-	-	-
18	UART1_CLK_EN	1'b1	RW	UART1 时钟使能 0: 不使能 1: 使能
17	UART0_CLK_EN	1'b1	RW	UART0 时钟使能 0: 不使能 1: 使能
16	Reserved	-	-	-
15	SPI_IIC0_CLK_EN	1'b1	RW	SPI_IIC0 时钟使能 0: 不使能 1: 使能
14	TK_CLK_EN	1'b1	RW	TK 时钟使能 0: 不使能 1: 使能
13	LCD_PCLK_EN	1'b1	RW	LCD PCLK 时钟使能 0: 不使能 1: 使能
12:11	Reserved	-	-	-
10	TIMER1_CLK_EN	1'b1	RW	TIMER1 时钟使能 0: 不使能 1: 使能
9	TIMER0_CLK_EN	1'b1	RW	TIMER0 时钟使能 0: 不使能 1: 使能
8:7	Reserved	-	-	-
6	TIMER4_CLK_EN	1'b1	RW	TIMER4 时钟使能 0: 不使能 1: 使能
5:3	Reserved	-	-	-
2	SRAM0_CLK_EN	1'b1	RW	SRAM0 时钟使能 0: 不使能 1: 使能
1:0	Reserved	-	-	-

20.2.2.9. 控制寄存器 (CLKCON3)

Width	Name	Reset	Property	Description
31	HIRC_EN_FLAG	1'b1	RO	HIRC_CLK 使能标志位 0: 未使能 1: 已使能
30	Reserved	-	-	-
29:18	HIRC_FSC	12'h0	RW	HIRC_CLK 频率控制(NVR TRIM) h0: 最低频

				... hfff: 最高频
17	HIRC_EN	1'b1	RW	HIRC_CLK 使能控制 0: 不使能 1: 使能
16:12	Reserved	-	-	-
11:10	HIRC_FSEL	2'b0	RW	HIRC 频率选择 b00: 24MHz b01: 32MHz b10: 36MHz b11: 48MHz
9:7	HIRC_FTS	3'b100	RW	HIRC 温度补偿系数
6:2	HIRC_FFS	5'b10000	RW	HIRC fine tune control (NVR TRIM) b00000: lowest freq ... b11111: highest freq

20.2.2.10. 控制寄存器 (SYS_ERR)

Width	Name	Reset	Property	Description
31:2	Reserved	-	-	-
1	CLK_ERR	1'b0	RW	时钟使用错误标志 0: 正确 1: 错误 软件写 0 清除
0	SYS_ERR0	1'b0	RW	系统总线访问越界地址错误标志 0: 正确 1: 错误 软件写 0 清除

20.2.2.11. 控制寄存器 (WKUP_CON)

Width	Name	Reset	Property	Description
31:28	Reserved	-	-	-
27:24	CLR_WKUP_PEND	4'h0	WO	软件写 1 清除 WKUP_PEND
23:20	Reserved	-	-	-
19:16	WKUP_PEND	4'h0	RO	IO 边沿唤醒标志 0: 未检测到边沿 1: 检测到边沿
15:12	Reserved	-	-	-
11:8	WKUP_EDGE	4'h0	RW	唤醒边沿选择 0: 上升沿 1: 下降沿
7:4	Reserved	-	-	-
3:0	WKUP_EN	4'h0	RW	IO 边沿检测唤醒 0: 不使能 1: 使能 WKUP_EN[2]: 复用为 TK 唤醒使能

				WKUP_EN[3]: 复用为 LVDVCC 唤醒使能
--	--	--	--	-----------------------------

20.2.2.12. 控制寄存器 (LPCON)

Width	Name	Reset	Property	Description
31:8	Reserved	-	-	-
7	PMU_LP_HW_EN	1'b0	RW	设置为 1 时, 当 PMU 进入 sleep 模式, 硬件自动关闭 PMU_V2IEN, 且 LPLDOS 切换到 PMUBK 寄存器的值 0: 不使能 1: 使能
6	Reserved	-	-	-
5	RC256K_SOFT_EN	1'b1	RW	LIRC_256K 使能 0: 不使能 1: 使能
4	RC256K_AUTO_DIS	1'b0	RW	进入 sleep 模式时自动关闭 LIRC_256K 0: 不使能 1: 使能
3	HIRC_AUTO_DIS	1'b0	RW	进入 sleep/stop 模式时自动关闭 HIRC 0: 不使能 1: 使能
2	SRAM0_AUTO_DIS	1'b0	RW	进入 sleep/stop 模式自动关闭 SRAM CE 0: 不使能 1: 使能
1	STOP_CLK_MODE	1'b0	RW	STOP 模式 0: 不使能 1: 使能
0	SLEEP	1'b0	RW	SLEEP 模式 0: 不使能 1: 使能

20.2.2.13. 控制寄存器 (CHIPID_DCN)

Width	Name	Reset	Property	Description
31:24	Reserved	-	-	-
23:16	CHIP_DCN	8'h0	RO	记录版本修改
15:0	CHIP_ID	16'h7	RO	芯片 ID

20.2.2.14. 控制寄存器 (PMUCON0)

Width	Name	Reset	Property	Description
28:25	HP_VREF_SEL	4'h8	RW	普通模式 VREF 电压控制(EFALSH TRIM) b0000: 最小值 ... b1000: 1.218 ... b1111: 最大值 Step=6mv
24:21	HP_V2I_SEL	4'h8	RW	模拟偏置电流选择(FLASH TRIM): h0: 最小值

				... hf: 最大值
20	HP_V2I_EN	1'b1	RW	模拟偏置电流使能 0: 不使能 1: 使能
18	HP_PDLI_EN	1'b1	RW	VDD LDO 弱下拉电阻开关 0: 不使能 1: 使能
17	HP_PDI_EN	1'b1	RW	VDD LDO 下拉电阻开关 0: 不使能 1: 使能
16:14	LP_IREF_SEL	3'b101	RW	低功耗参考电流控制(EFALSH TRIM) b000: 最小电流 ... b111: 最大电流
13	HP_LDO_EN	1'b1	RW	普通模式 LDO 使能 0: 不使能 1: 使能 进入 SLEEP 模式时自动清 0.
12	INNER_LDO_EN	1'b1	RW	内部 LDO 使能 0: 不使能 1: 使能
11:8	LP_VREF_SEL	4'h8	RW	低功耗模式 VREF 电压控制(EFALSH TRIM) b0000: 最小值 ... b1000: 1.218 ... b1111: 最大值 Step=10mv
7	LP_IDEEP	1'b0	RW	深度睡眠电流开关 0: 使能 1: 不使能
6:4	LP_VDD_SEL	3'b0	RW	低功耗模式 VDD 电压选择(FLASH TRIM): b000=1.50V b001=1.45V b010=1.55V b011=1.60V b100=1.65V b101=1.75V b110=1.85V b111=1.25V
3	LP_SSI	1'b0	RW	PMU 低功耗加速 0: 不使能 1: 使能
2:0	HP_VDD_SEL	3'b0	RW	普通模式 VDD 电压选择(FLASH TRIM): b000=1.50V b001=1.45V

				b010=1.55V b011=1.60V b100=1.65V b101=1.75V b110=1.85V b111=1.25V
--	--	--	--	--

20.2.2.15. 控制寄存器 (RPCON)

Width	Name	Reset	Property	Description
31:19	Reserved	-	-	-
18	LOCKUP_RESET_CLR	1'b0	WO	软件写 1 清除 LOCK_RESET_PEND
17	SOFT_RESET_CLR	1'b0	WO	软件写 1 清除 SOFT_RESET_PEND
16	SLEEP_STA_CLR	1'b0	WO	软件写 1 清除 SLEEP_PEND
15:3	Reserved	-	-	-
2	LOCK_RESET_PEND	1'b0	RO	CPU LOCK RESET 发生时, 该标志位置 1 0: 没有发生 LOCK_RESET 1: 发生过 LOCK_RESET
1	SOFT_RESET_PEND	1'b0	RW	软件对该位写 1 触发系统复位, 同时会重新获取 FLASH 的 NVR/MAIN 数据 0: 不使能软件系统复位 1: 软件触发系统复位且该位置 1
0	SLEEP_PEND	1'b0	RO	软件写 LPCON[0]进入 sleep 模式时, 该 bit 置 1, CPU 写 SLEEP_STA_CLR 清除。

20.2.2.16. 控制寄存器 (PMUBK)

Width	Name	Reset	Property	Description
31:7	Reserved	-	-	-
6:4	PMU_LPDOS	3'h1	RW	低功耗模式 LDO 电压档位选择 b000: 1.50V b001: 1.45V b010: 1.55V b011: 1.60V b100: 1.65V b101: 1.75V b110: 1.85V b111: 1.25V Note: use with LPCON[8:7]
3:0	Reserved	-	-	-

21. 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 CIU32F011、CIU32F031 系列微控制器。

21.1. 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 CIU32F011、CIU32F031 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。

22. 调试支持 (DEBUG)

CIU32F011、CIU32F031 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 CIU32F011、CIU32F031 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

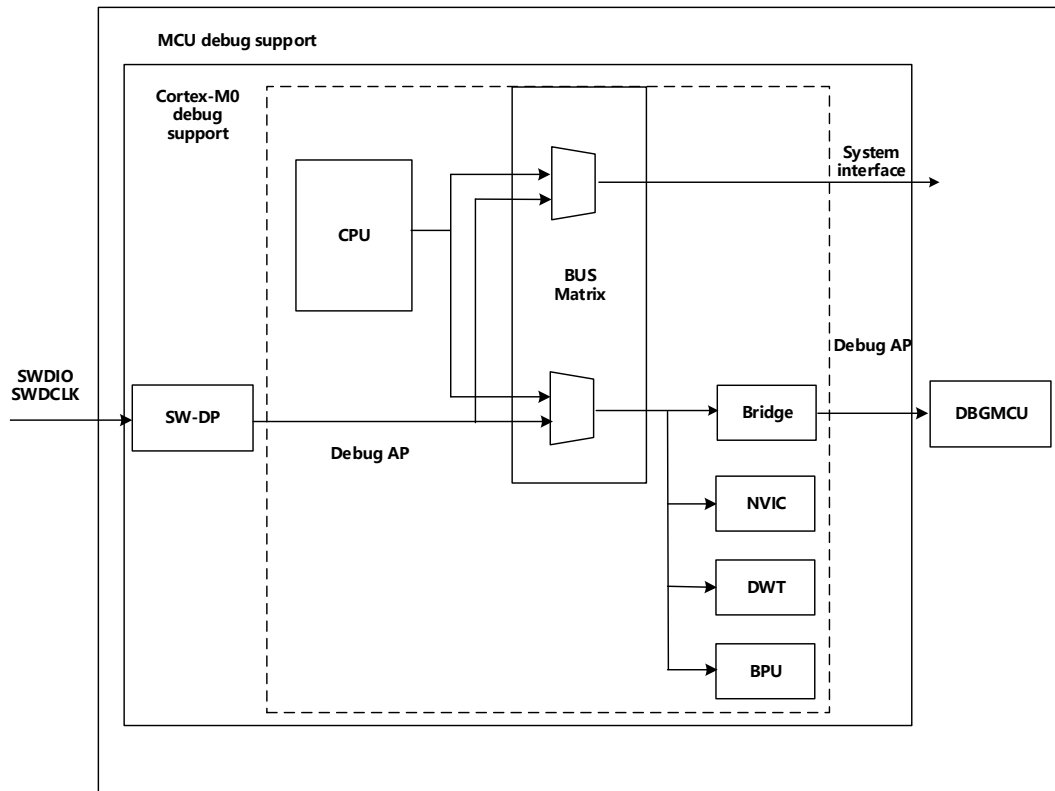


图 22-1 串行接口调试框图

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

22.1. 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。

22.2. SWD 调试端口脚

CIU32F011、CIU32F031 的 SW-DP 接口引脚如下所示：

SWD-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	PA15
SWCLK	输入	串行时钟	PA14

SW-DP 接口引脚选择在 flash 用户自定义功能区域定义。

22.2.1. 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，因为他们直接连接到 D 触发器控制着调试模式。必须特别注意 SWCLK 引脚，因为他们直接连接到一些 D 触发器的时钟端。

为了避免任何未受控制的 IO 电平，CIU32F011、CIU32F031 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 IO 被用户代码释放，GPIO 控制器再次取得控制。这些 IO 口的状态将恢复到复位时的状态。

软件可以把这些 IO 口作为普通的 IO 口使用。

22.3. SW 调试端口

22.3.1. 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。

22.3.2. 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。

22.4. 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。

23. 版本历史

表 23-1 版本更改履历

日期	版本号	修改范围
2022-03-15	V1.0	正式版本
2022-12-06	V1.1	增加 LQFP32 封装尺寸信息
2023-02-17	V1.2	订正 NVR 区相关描述
2023-07-12	V1.3	增加 SOP16 封装尺寸信息
2023-08-25	V1.4	1、引脚分配图章节：增加 QFN20、QFN32 封装，更新引脚分配图 2、引脚功能描述章节：增加 QFN20、QFN32 封装引脚功能描述

24. 联系方式

网址: www.hed.com.cn

地址: 北京市昌平区北七家未来科技城南区中国电子网络安全和信息化产业基地 C 栋

邮编: 102209

如果您在购买与使用过程中有任何意见或建议, 请随时与我们联系。