

概述

CH572 是集成 2.4G 无线通讯的 RISC-V MCU 微控制器。片上集成了 2Mbps 低功耗蓝牙 BLE 通讯模块、USB 全速控制器及收发器、电压比较器 CMP、按键检测模块、SPI、串口、I2C 等丰富的外设资源。CH572 内置了 LD05V 调压器产生 3.3V，支持单一 3.3V 或单一 5V 电源供电，支持单线或双线仿真调试，适用于 2.4G 无线通讯应用和低引脚数的简单蓝牙应用。

CH570 是低功耗的 2.4G 无线通讯 MCU，不支持蓝牙，其它模块参考 CH572。

功能

● 内核 Core:

- 青稞 32 位 RISC-V3C 内核
- 支持 RV32IMBC 指令集和自扩展指令
- 低功耗三级流水线
- 多档系统主频，最高 100MHz
- 特有高速的中断响应机制

● 256K 字节非易失存储器 FlashROM:

- 240KB 用户应用程序存储区 CodeFlash
- 8KB 系统引导程序存储区 BootLoader
- 8KB 系统非易失配置信息存储区 InfoFlash
- 支持 ICP、ISP 和 IAP，支持 OTA 无线升级

● 12K 字节易失数据存储 SRAM:

- 12KB 双电源供电的睡眠保持存储区

● 电源管理和低功耗:

- 内置 5V 转 3.3V 调压器 LD05V
- 单一 V5 供电额定电压: 5V
- 或单一 VDD33 供电额定电压: 3.3V
- 空闲模式 Idle: 1.7mA
- 暂停模式 Halt: 1.3mA (PLL/HSE 不停止)
420uA (PLL/HSE 停止)
- 睡眠模式 Sleep: 0.46uA~1.2uA 多档
- 下电模式 Shutdown: 0.3uA~0.9uA 多档
- 可选低功耗的电池电压低压监控

● 安全特性: AES-128 加解密，芯片唯一 ID

● 2.4G 和低功耗蓝牙 BLE:

- 集成 2.4GHzRF 收发器和基带及链路控制
- 支持 GFSK 数字调制与解调
- 接收灵敏度 -95dBm，可编程 +7.5dBm 发送功率
- BLE 符合 Bluetooth Low Energy 5.0 规范
- 支持 2Mbps 和 1Mbps
- 支持 2.4G 模式下最高 8KHz 上报率
- 提供优化的协议栈和应用层 API，支持组网

● 实时时钟 RTC: 支持定时和触发两种模式

● 时钟: 内置 PLL，内置低频 RC 振荡器

● 按键检测模块

- 支持 20 路按键检测，包括 10 路矩阵区按键和 10 路独立区按键

● 模拟电压比较器 CMP

- 内置 16 档参考电压，等效为 4 位 ADC

● 1 组全速 USB 2.0 控制器及 PHY:

- 15 个端点，支持 64 字节数据包，支持 DMA
- 支持全/低速的 Host 主机和 Device 设备模式
- 支持单线 USB 通讯

● 定时器 Timer 和脉宽调制 PWM:

- 1 组 26 位定时器，16MHz 主频定时可达 4.2S
- 1 路捕捉/采样，支持上升沿/下降沿/双边沿
- 支持编码器模式 (ENC MODE)，提供 2 路通道，用于捕捉旋转编码器信号进行编码
- 1 路 26 位 PWM 输出
- 5 路 16 位 PWM 输出
- 2 个看门狗定时器：独立和窗口型
- 系统时基定时器：32 位计数器

● 异步串口 UART:

- 1 组独立 UART，兼容 16C550，内置 8 级 FIFO
- 23 位计数器，通讯波特率可达 12.5Mbps

● 串行外设接口 SPI:

- 支持 Master 和 Slave 模式
- 内置 FIFO，支持 DMA

● 两线串行接口 I2C:

- 支持 Master 和 Slave 模式，兼容 SMBus
- 支持 7 位或 10 位地址和总线广播
- 支持仲裁、错误检测、PEC 校验、延长时钟

● 通用输入输出端口 GPIO:

- 12 个 GPIO，其中 1 个支持 5V 信号输入
- 可选上拉或下拉电阻，可选输出驱动能力
- 12 个 GPIO 支持电平或边沿中断输入

● 单/双线仿真调试接口

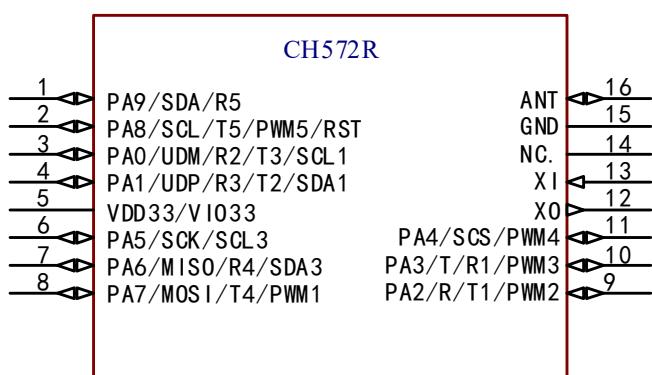
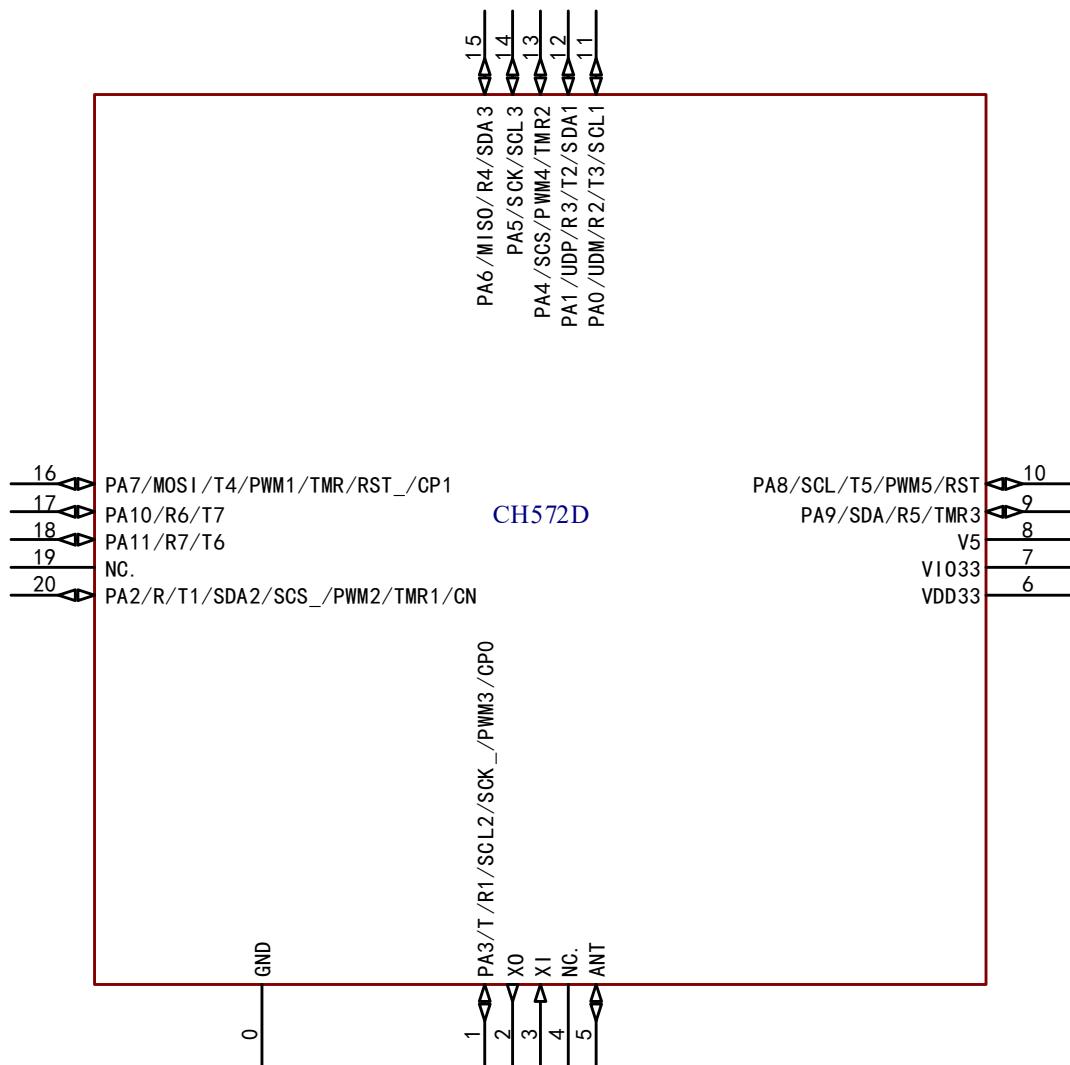
● 封装形式: QFN、DFN、SOP

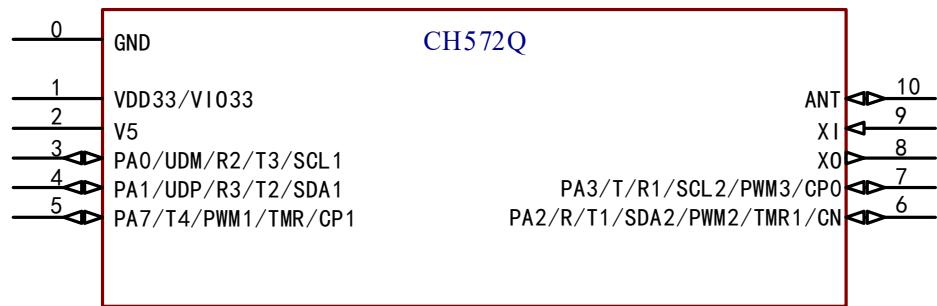
芯片型号	用户闪存+引导闪存	数据存储RAM	RTC时钟	定时器	捕捉	PWM	串口	SPI	I2C	CMP	按键监测	USB主机	USB设备	2.4G无线	BLE蓝牙	电源电压	通用I/O	封装形式
CH572D	240+8K	12K	√	1	1+2	1+5	1	1	1	1	20	√	√	√	√	2.0~3.6V 或 4.5~5.3V	12	QFN20
CH572Q	240+8K	12K		1	1+2	1+3	1	—	1	1	5	√	√		√	4.5~5.3V	5	DFN10X3
CH572R	240+8K	12K		1	1+2	1+5	1	1	1	1	9	√	√		√	2.0~3.6V	10	TSSOP16
CH570D	240+8K	12K		1	1+2	1+5	1	1	1	1	20	√	√		—	2.0~3.6V 或 4.5~5.3V	12	QFN20
CH570Q	240+8K	12K		1	1+2	1	1	—	1	—	5	—	√		—	4.5~5.3V	5	DFN10X3
CH570E	240+8K	12K		1	—	1	1	—	1	—	—	—	—		—	2.0~3.6V	3	SOP8

第1章 引脚信息

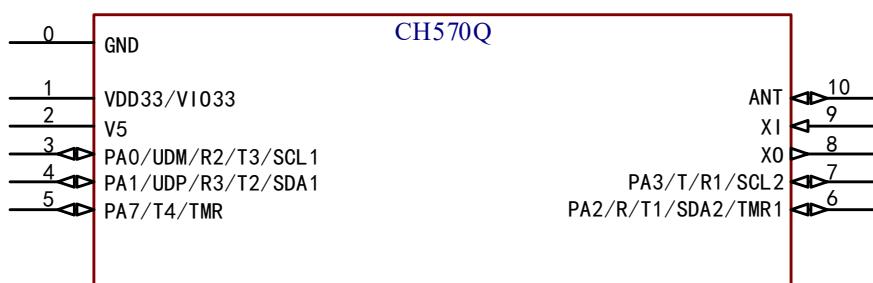
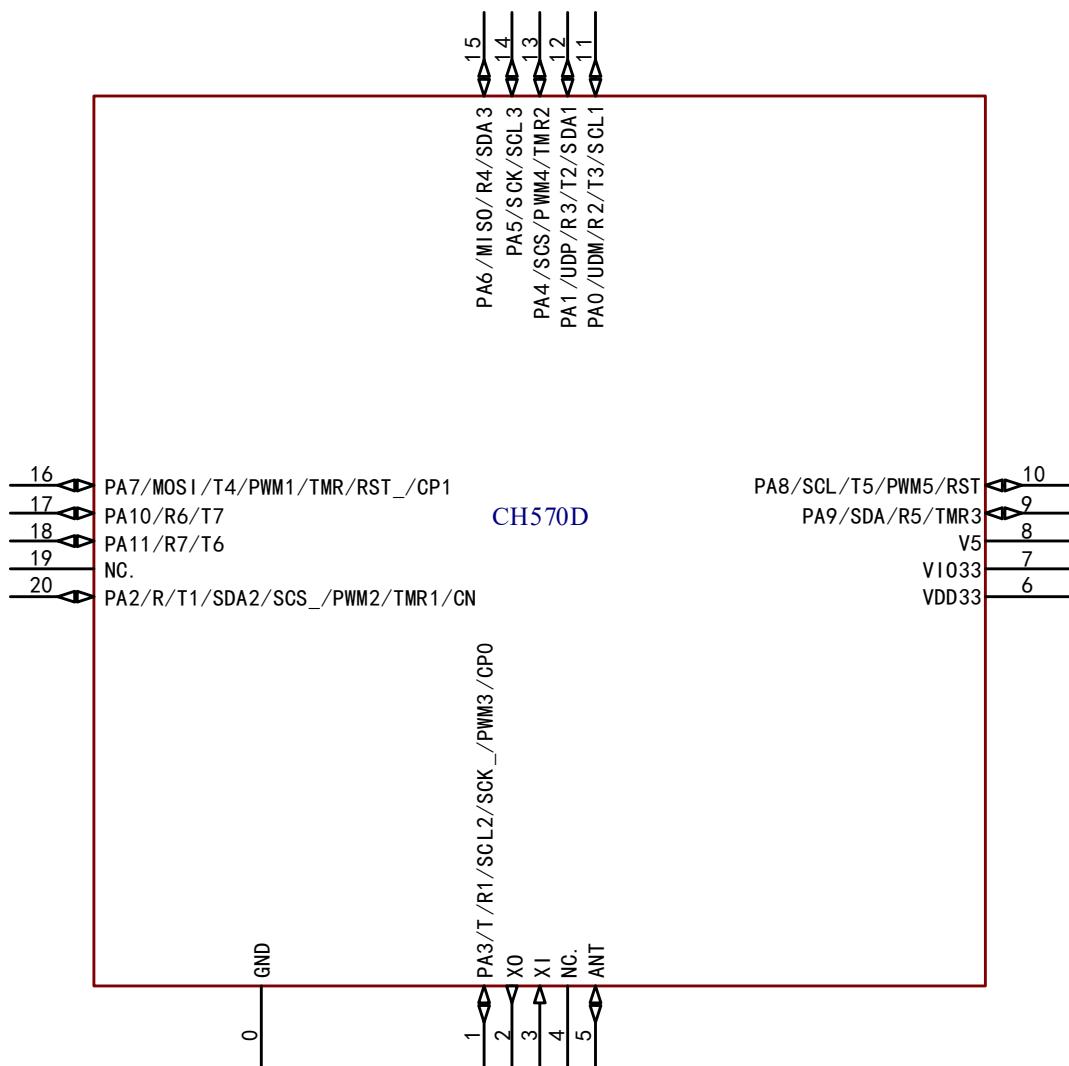
1.1 引脚排列

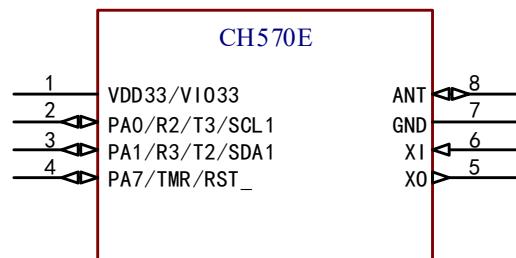
1.1.1 CH572 引脚排列





1.1.2 CH570 引脚排列





1.2 引脚定义

注意，下表中的引脚功能描述针对的是所有功能，不涉及具体型号产品。不同型号之间外设资源有差异，查看前请先根据产品型号资源表确认是否有此功能。

表 1-1 CH572 引脚定义

引脚编号			引脚名称	引脚类型 ⁽¹⁾	复用功能 在前优先 ⁽²⁾	功能描述
CH572Q	CH572R	CH572D				
0	-	0	GND	P	-	芯片底板，公共接地端，电压 0V 参考点。
-	15	-	GND	P	-	公共接地端，电压 0V 参考点。
7	10	1	PA3	I/O/A	/CMP_P0 /TXD /SCL_2 /SCK_ /PWM3 /RXD_1 /KEYSCAN1	PA3：通用双向数字 I/O 引脚。 CMP_P0：比较器的输入正端。 TXD：UART 串行数据输出。 SCL_2：I2C 串行时钟引脚映射。 SCK_：SPI 的 SCK 引脚映射。 PWM3：脉宽调制输出通道 3。 RXD_1：UART 的 RXD 引脚映射。 KEYSCAN1：按键扫描输入 1。
8	12	2	X0	0/A	-	高频振荡器 HSE 的反相输出端，外接 32MHz 晶体的一端。
9	13	3	X1	A	-	高频振荡器 HSE 的输入端，外接 32MHz 晶体的另一端。
-	14	4, 19	NC.	NC.	-	空脚。
10	16	5	ANT	A	-	RF 射频信号输入输出，建议直连天线。
1	5	6	VDD33	P	-	模拟电源输入，需贴近引脚外接 2.2uF 并联 0.1uF 电容。 如果是单一 3.3V 电源供电，则 VDD33 引脚输入额定 3.3V 电源，V5 引脚悬空或与 VDD33 引脚短接。
		7	VI033	P	-	I/O 电源输入，需要与 VDD33 引脚短接。
2	-	8	V5	P	-	当单一 5V 电源供电时，V5 引脚输入额定 5V 电源，为内部 LD05V 供电，由 LD05V 在 VDD33 引脚产生 3.3V，需贴近引脚外接 2.2uF 电容，并需在 V5 与 VDD33 引脚之间串接额定 1.5KΩ 电阻。 当单一 3.3V 电源供电时，V5 引脚悬空或与 VDD33 引脚短接。
-	1	9	PA9	I/O/5VT	SDA /TMR_3 /RXD_5 /CAP_IN1_3 /CAP_IN2_2	PA9：通用双向数字 I/O 引脚。 SDA：I2C 串行数据引脚，开漏输出和输入。 TMR_3：定时器的 TMR 引脚映射。 RXD_5：UART 的 RXD 引脚映射。 CAP_IN1_3：定时器的捕获输入通道 1 的映射。 CAP_IN2_2：定时器的捕获输入通道 2 的映射。
-	2	10	PA8	I/O	TXD_5 /SCL /PWM5 /RST /KEYSCAN2	PA8：通用双向数字 I/O 引脚。 TXD_5：UART 的 TXD 引脚映射。 SCL：I2C 串行时钟引脚，主机输出和输入/从机输入。 PWM5：脉宽调制输出通道 5。 RST：外部复位输入，低电平有效，内置上拉电阻。 KEYSCAN2：按键扫描输入 2。
3	3	11	PA0	I/O/A	UDM /SWDIO /TXD_3	PA0：通用双向数字 I/O 引脚。 UDM：全速 USB 2.0 的 D- 数据线。 SWDIO ⁽³⁾ ：仿真调试接口的串行数据输入输出。

引脚编号			引脚名称	引脚类型 ⁽¹⁾	复用功能 在前优先 ⁽²⁾	功能描述
CH572Q	CH572R	CH572D				
					/SCL_1 /RXD_2	TXD_3: UART 的 TXD 引脚映射。 SCL_1: I2C 的 SCL 引脚映射。 RXD_2: UART 的 RXD 引脚映射。
4	4	12	PA1	I/O/A	UDP /SWCLK /TXD_2 /SDA_1 /RXD_3	PA1: 通用双向数字 I/O 引脚。 UDP: 全速 USB 2.0 的 D+数据线。 SWCLK ⁽³⁾ : 仿真调试接口的串行时钟输入。 TXD_2: UART 的 TXD 引脚映射。 SDA_1: I2C 的 SDA 引脚映射。 RXD_3: UART 的 RXD 引脚映射。
-	11	13	PA4	I/O	X25M0 /PWM4 /TMR_2 /SCS /CAP_IN1_2 /CAP_IN2_3	PA4: 通用双向数字 I/O 引脚。 X25M0: 25MHz 时钟输出。 PWM4: 脉宽调制输出通道 4。 TMR_2: 定时器的 TMR 引脚映射。 SCS: SPI 从机模式下的片选输入, 低电平有效。 CAP_IN1_2: 定时器的捕获输入通道 1 的映射。 CAP_IN2_3: 定时器的捕获输入通道 2 的映射。
-	6	14	PA5	I/O	SCL_3 /SCK	PA5: 通用双向数字 I/O 引脚。 SCL_3: I2C 的 SCL 引脚映射。 SCK: SPI 串行时钟引脚, 主机输出/从机输入。
-	7	15	PA6	I/O	SDA_3 /MISO /RXD_4	PA6: 通用双向数字 I/O 引脚。 SDA_3: I2C 的 SDA 引脚映射。 MISO: SPI 串行数据引脚, 主机输入。 RXD_4: UART 的 RXD 引脚映射。
5	8	16	PA7	I/O/A	CMP_P1 /TXD_4 /MOSI /PWM1 /TMR /RST_ /CAP_IN1 /CAP_IN2_1	PA7: 通用双向数字 I/O 引脚。 CMP_P1: 比较器的输入正端。 TXD_4: UART 的 TXD 引脚映射。 MOSI: SPI 串行数据引脚, 主机输出。 PWM1: 脉宽调制输出通道 1。 TMR: 定时器 1 的捕捉输入 1 和 PWM 输出通道 1。 RST_: RST 引脚映射。 CAP_IN1: 定时器的捕获输入通道 1。 CAP_IN2_1: 定时器的捕获输入通道 2 的映射。
-	-	17	PA10	I/O	TXD_7 /RXD_6 /KEYSCAN3	PA10: 通用双向数字 I/O 引脚。 TXD_7: UART 的 TXD 引脚映射。 RXD_6: UART 的 RXD 引脚映射。 KEYSCAN3: 按键扫描输入 3。
-	-	18	PA11	I/O	TXD_6 /RXD_7 /KEYSCAN4	PA11: 通用双向数字 I/O 引脚。 TXD_6: UART 的 TXD 引脚映射。 RXD_7: UART 的 RXD 引脚映射。 KEYSCAN4: 按键扫描输入 4。
6	9	20	PA2	I/O/A	CMP_N /TXD_1 /SDA_2 /PWM2	PA2: 通用双向数字 I/O 引脚。 CMP_N: 比较器的输入负端。 TXD_1: UART 的 TXD 引脚映射。 SDA_2: I2C 的 SDA 引脚映射。

引脚编号			引脚名称	引脚类型 ⁽¹⁾	复用功能 在前优先 ⁽²⁾	功能描述
CH572Q	CH572R	CH572D				
					/TMR_1 /RXD /SCS_ /KEYSCAN0 /CAP_IN1_1 /CAP_IN2	PWM2: 脉宽调制输出通道 2。 TMR_1: 定时器的 TMR 引脚映射。 RXD: UART 的串行数据输入。 SCS_: SPI 的 SCS 引脚映射。 KEYSCAN0: 按键扫描输入 0。 CAP_IN1_1: 定时器的捕获输入通道 1 的映射。 CAP_IN2: 定时器的捕获输入通道 2。

表 1-2 CH570 引脚定义

引脚编号			引脚名称	引脚类型 ⁽¹⁾	复用功能 在前优先 ⁽²⁾	功能描述
CH570E	CH570R	CH570D				
7	0	0	GND	P	-	芯片底板, 公共接地端, 电压 0V 参考点。
-	7	1	PA3	I/O/A	/CMP_P0 /TXD /SCL_2 /SCK_ /PWM3 /RXD_1 /KEYSCAN1	PA3: 通用双向数字 I/O 引脚。 CMP_P0: 比较器的输入正端。 TXD: UART 串行数据输出。 SCL_2: I2C 串行时钟引脚映射。 SCK_: SPI 的 SCK 引脚映射。 PWM3: 脉宽调制输出通道 3。 RXD_1: UART 的 RXD 引脚映射。 KEYSCAN1: 按键扫描输入 1。
5	8	2	X0	0/A	-	高频振荡器 HSE 的反相输出端, 外接 32MHz 晶体的一端。
6	9	3	X1	A	-	高频振荡器 HSE 的输入端, 外接 32MHz 晶体的另一端。
-	-	4, 19	NC.	NC.	-	空脚。
8	10	5	ANT	A	-	RF 射频信号输入输出, 建议直连天线。
1	1	6	VDD33	P	-	模拟电源输入, 需贴近引脚外接 2.2uF 并联 0.1uF 电容。 如果是单一 3.3V 电源供电, 则 VDD33 引脚输入额定 3.3V 电源, V5 引脚悬空或与 VDD33 引脚短接。
		7	VI033	P	-	I/O 电源输入, 需要与 VDD33 引脚短接。
-	2	8	V5	P	-	当单一 5V 电源供电时, V5 引脚输入额定 5V 电源, 为内部 LD05V 供电, 由 LD05V 在 VDD33 引脚产生 3.3V, 需贴近引脚外接 2.2uF 电容, 并需在 V5 与 VDD33 引脚之间串接额定 1.5KΩ 电阻。 当单一 3.3V 电源供电时, V5 引脚悬空或与 VDD33 引脚短接。
-	-	9	PA9	I/O/5VT	SDA /TMR_3 /RXD_5 /CAP_IN1_3 /CAP_IN2_2	PA9: 通用双向数字 I/O 引脚。 SDA: I2C 串行数据引脚, 开漏输出和输入。 TMR_3: 定时器的 TMR 引脚映射。 RXD_5: UART 的 RXD 引脚映射。 CAP_IN1_3: 定时器的捕获输入通道 1 的映射。 CAP_IN2_2: 定时器的捕获输入通道 2 的映射。
-	-	10	PA8	I/O	TXD_5	PA8: 通用双向数字 I/O 引脚。

引脚编号			引脚名称	引脚类型 ⁽¹⁾	复用功能 在前优先 ⁽²⁾	功能描述
CH570E	CH570Q	CH570D				
					/SCL /PWM5 /RST /KEYSCAN2	TXD_5: UART 的 TXD 引脚映射。 SCL: I2C 串行时钟引脚, 主机输出和输入/从机输入。 PWM5: 脉宽调制输出通道 5。 RST: 外部复位输入, 低电平有效, 内置上拉电阻。 KEYSCAN2: 按键扫描输入 2。
2	3	11	PA0	I/O/A	UDM /SWDIO /TXD_3 /SCL_1 /RXD_2	PA0: 通用双向数字 I/O 引脚。 UDM: 全速 USB 2.0 的 D-数据线。 SWDIO ⁽³⁾ : 仿真调试接口的串行数据输入输出。 TXD_3: UART 的 TXD 引脚映射。 SCL_1: I2C 的 SCL 引脚映射。 RXD_2: UART 的 RXD 引脚映射。
3	4	12	PA1	I/O/A	UDP /SWCLK /TXD_2 /SDA_1 /RXD_3	PA1: 通用双向数字 I/O 引脚。 UDP: 全速 USB 2.0 的 D+数据线。 SWCLK ⁽³⁾ : 仿真调试接口的串行时钟输入。 TXD_2: UART 的 TXD 引脚映射。 SDA_1: I2C 的 SDA 引脚映射。 RXD_3: UART 的 RXD 引脚映射。
-	-	13	PA4	I/O	X25M0 /PWM4 /TMR_2 /SCS /CAP_IN1_2 /CAP_IN2_3	PA4: 通用双向数字 I/O 引脚。 X25M0: 25MHz 时钟输出。 PWM4: 脉宽调制输出通道 4。 TMR_2: 定时器的 TMR 引脚映射。 SCS: SPI 从机模式下的片选输入, 低电平有效。 CAP_IN1_2: 定时器的捕获输入通道 1 的映射。 CAP_IN2_3: 定时器的捕获输入通道 2 的映射。
-	-	14	PA5	I/O	SCL_3 /SCK	PA5: 通用双向数字 I/O 引脚。 SCL_3: I2C 的 SCL 引脚映射。 SCK: SPI 串行时钟引脚, 主机输出/从机输入。
-	-	15	PA6	I/O	SDA_3 /MISO /RXD_4	PA6: 通用双向数字 I/O 引脚。 SDA_3: I2C 的 SDA 引脚映射。 MISO: SPI 串行数据引脚, 主机输入。 RXD_4: UART 的 RXD 引脚映射。
4	5	16	PA7	I/O/A	CMP_P1 /TXD_4 /MOSI /PWM1 /TMR /RST_ /CAP_IN1 /CAP_IN2_1	PA7: 通用双向数字 I/O 引脚。 CMP_P1: 比较器的输入正端。 TXD_4: UART 的 TXD 引脚映射。 MOSI: SPI 串行数据引脚, 主机输出。 PWM1: 脉宽调制输出通道 1。 TMR: 定时器 1 的捕捉输入 1 和 PWM 输出通道 1。 RST_+: RST 引脚映射。 CAP_IN1: 定时器的捕获输入通道 1。 CAP_IN2_1: 定时器的捕获输入通道 2 的映射。
-	-	17	PA10	I/O	RXD_6 /TXD_7 /KEYSCAN3	PA10: 通用双向数字 I/O 引脚。 RXD_6: UART 的 RXD 引脚映射。 TXD_7: UART 的 TXD 引脚映射。 KEYSCAN3: 按键扫描输入 3。

引脚编号			引脚名称	引脚类型 ⁽¹⁾	复用功能 在前优先 ⁽²⁾	功能描述
CH570E	CH570Q	CH570D				
-	-	18	PA11	I/O	TXD_6 /RXD_7 /KEYSCAN4	PA11: 通用双向数字 I/O 引脚。 TXD_6: UART 的 TXD 引脚映射。 RXD_7: UART 的 RXD 引脚映射。 KEYSCAN4: 按键扫描输入 4。
-	6	20	PA2	I/O/A	CMP_N /TXD_1 /SDA_2 /PWM2 /TMR_1 /RXD /SCS /KEYSCAN0 /CAP_IN1_1 /CAP_IN2	PA2: 通用双向数字 I/O 引脚。 CMP_N: 比较器的输入负端。 TXD_1: UART 的 TXD 引脚映射。 SDA_2: I2C 的 SDA 引脚映射。 PWM2: 脉宽调制输出通道 2。 TMR_1: 定时器的 TMR 引脚映射。 RXD: UART 的串行数据输入。 SCS_: SPI 的 SCS 引脚映射。 KEYSCAN0: 按键扫描输入 0。 CAP_IN1_1: 定时器的捕获输入通道 1 的映射。 CAP_IN2: 定时器的捕获输入通道 2。

注: 1. 引脚类型: P = 电源; I = TTL/CMOS 电平斯密特输入; O = CMOS 电平三态输出;
 A = 模拟信号输入或输出; $5VT$ = 支持 5V 耐压信号输入。

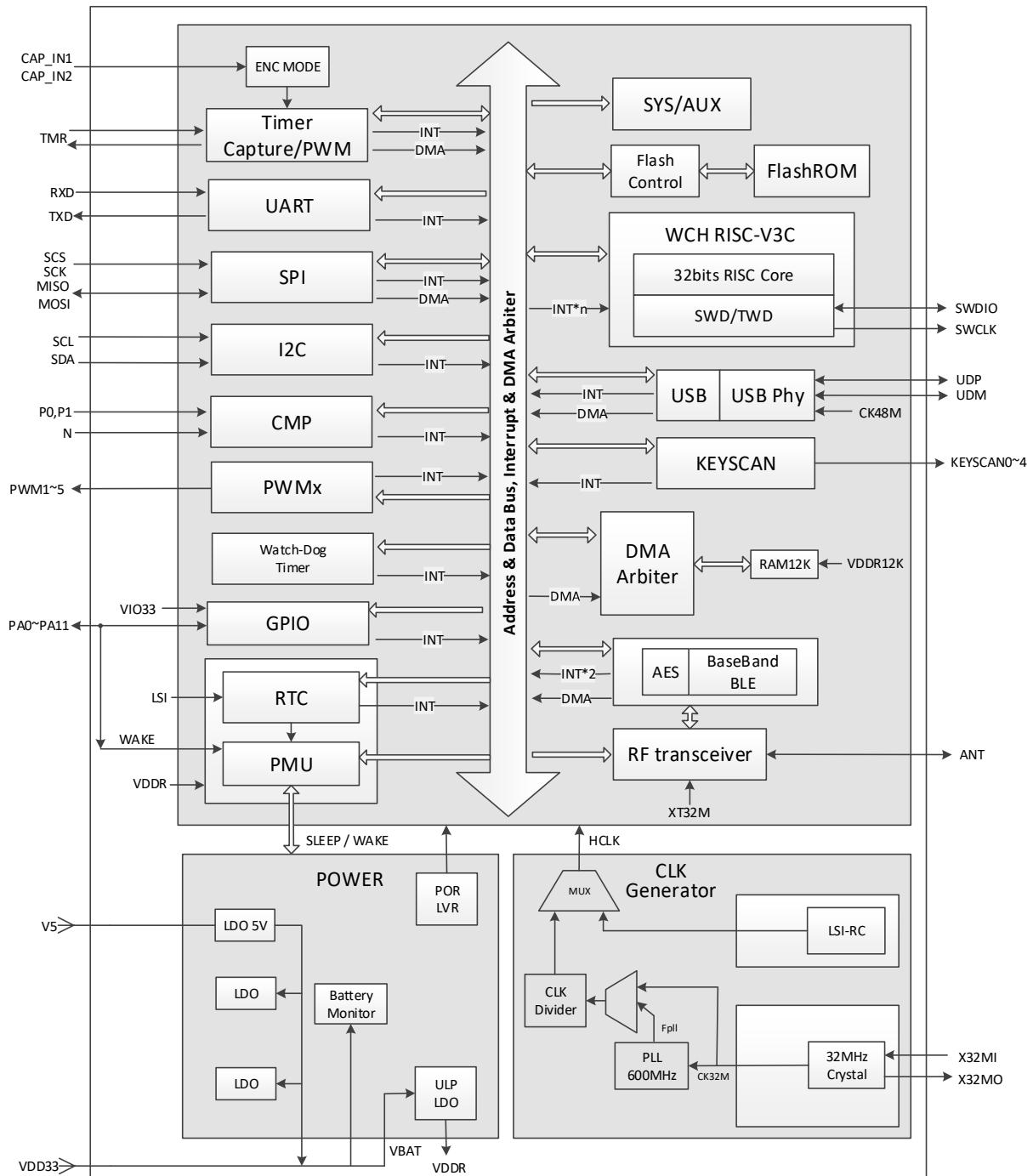
2. 引脚的复用功能及映射根据其优先级在表中按从高到底排列, 其中 GPIO 功能为最低优先级。
3. 系统上电或复位后默认调试接口引脚功能开启, 可根据需要配置寄存器关闭调试接口。仿真调试接口启用后, PA0 和 PA1 仅用作 SWDIO 和 SWCLK, 不再用于 GPIO 或外设复用功能引脚。关闭仿真调试接口后, PA0 和 PA1 才可用于 GPIO 和外设复用功能引脚。

第 2 章 系统结构及存储器

2.1 系统结构

下图为 CH572 系统结构框图。内核是青稞 RISC-V 微处理器，详细说明参考青稞内核手册。

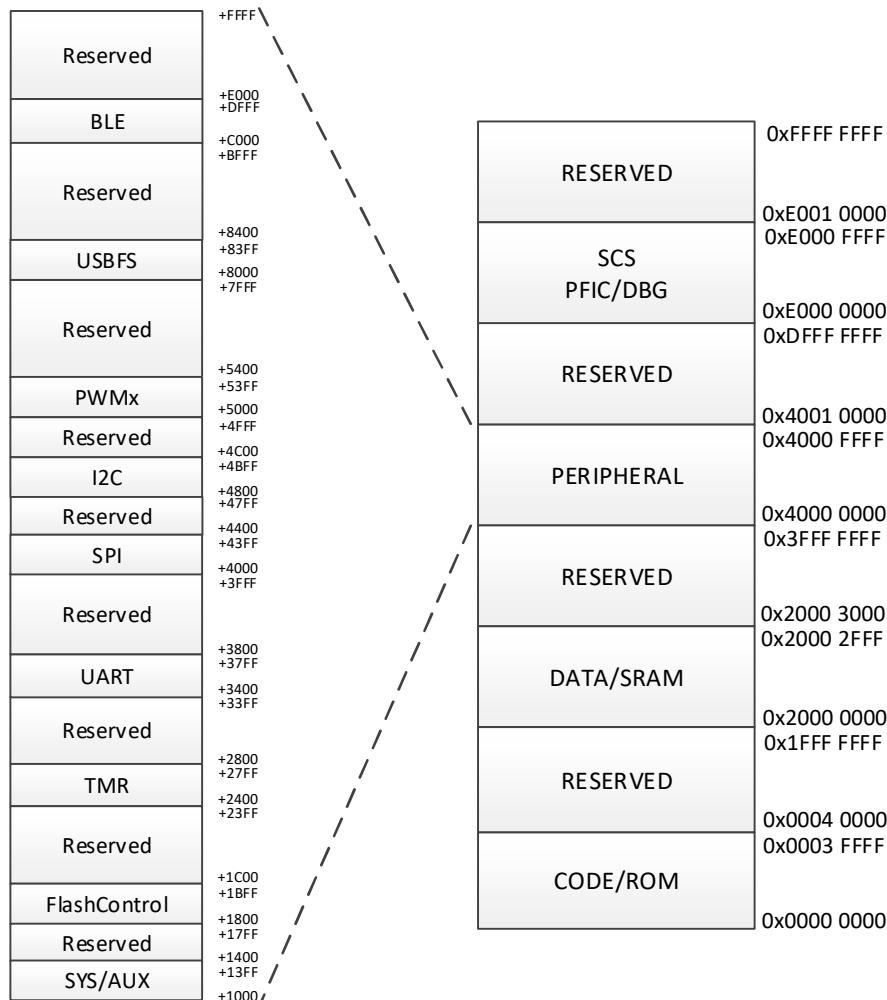
图 2-1 CH572 内部结构框图



2.2 存储器映像

CH572 的寻址空间主要包括 CODE 区/FlashROM、DATA 区/SRAM、外设等几个不同区域，详见下图所示。

图 2-2 存储器映射图



2.3 存储器映像表

各存储器映射区域地址范围如下表所示：

表 2-1 存储器映射区域地址

地址范围	用途	描述
0x00000000-0x0003FFFF	片上 CODE 区域, 非易失存储器	256KB, FlashROM
0x00040000-0x1FFFFFFF	保留	-
0x20000000-0x20002FFF	片上 DATA 区域, 易失存储器	12KB, SRAM
0x20003000-0x3FFFFFFF	保留	-
0x40000000-0x4000FFFF	各种外设	多个外设模块
0x40010000-0xEFFFFFFF	保留	-
0xE0000000-0xE000FFFF	系统内部各种外设	系统控制空间 SCS
0xE0010000-0xFFFFFFFF	保留	-

2.3.1 片上 CODE 区域映射表

表 2-2 CODE 区域地址

地址范围	用途	描述
0x00000000-0x0003BFFF	用户应用程序存储区 CodeFlash	240KB
0x0003C000-0x0003DFFF	系统引导程序存储区 BootLoader	8KB

0x0003E000~0x0003FFFF	系统非易失配置信息存储区 InfoFlash	8KB
地址 0x0003E000~0x0003EEFF 的配置信息可以由用户通过工具设置。		

表 2-3 用户级非易失配置信息说明

位地址	名称	用途	默认值
位 2~位 0	RESERVED	保留。	0
位 3	CFG_RESET_EN	RST 外部手工复位输入引脚使能。	0
位 4	CFG_RST_PIN_EN	外部复位引脚选择： 1: PA7; 0: PA8。	0
位 5	CFG_IWDG_EN	独立看门狗使能： 1: 开启独立看门狗; 0: 关闭独立看门狗。	0
位 6	CFG_BOOT_EN	系统引导程序 BootLoader 使能。	1b
位 15~位 7	RESERVED	保留。	0
位 23~位 16	CFG_ROM_READ	FlashROM 中的代码和数据保护模式： 3A: 允许读出; ! 3A: 禁止两线调试读出, 程序保密。	3Ah
位 27~位 24	RESERVED	保留。	0
位 31~位 28	VALID_SIG	配置信息有效标志, 固定值。	0100b

注: 关闭 FlashROM 代码和数据保护模式: 全擦除用户区, 并将配置信息的 CFG_ROM_READ (bit[23:16]) 写成 0x3A。

2.3.2 片上 DATA 区域映射表

表 2-4 DATA 区域地址

地址范围	用途	描述
0x20000000~0x20002FFF	主+辅双电源供电的可独立保持存储区 RAM12K	12KB

2.3.3 外设地址分配

CH572 主要包含以下外设, 每个外设占用一定的地址空间, 外设寄存器的实际访问地址为: 基地址+偏移地址。在后续章节中, 寄存器的地址有详细说明。下表为各个外设基地址的分配表。

表 2-5 外设基地址分配表

外设编号	外设名称	外设基址
1	SYS (PMU/RTC/GPIO/KEY SCAN 等) AUX (PLL/CMP 等)	0x4000 1000
2	FlashROM-Control	0x4000 1800
3	TMR	0x4000 2400
4	UART	0x4000 3400
5	SPI	0x4000 4000
6	I2C	0x4000 4800
7	PWMx (PWM1~PWM5)	0x4000 5000
8	USBFS	0x4000 8000
9	Radio:BLE	0x4000 C000 0x4000 D000

下表为后续章节寄存器描述中“访问”的解释说明：

表 2-6 访问属性说明

缩写词	描述
RF	只读且读取值为固定值，不受复位影响。
RO	只读。
WO	只写，读取值为 0 或无效。
RZ	只读，读取完之后自动清 0。
WZ	写则清 0。
RW	可读可写。
RW1	可读，写 1 则清 0。
WA	只写且仅安全模式下，读取值为 0 或无效。
RWA	可读，仅安全模式下可写。

下表为后续章节中使用的缩写解释说明：

表 2-7 名词缩写说明

缩写词	描述
HSE	外部高频晶体振荡时钟源（建议 32MHz）
LSI	内部低频 RC 时钟振荡源
CK32M	高频时钟源（默认 32MHz）
T_{CK32M}	高频时钟周期（1/CK32M）
FPLL	PLL 输出时钟（默认频率为 600MHz）
HCLK	系统主频时钟
Fsys	系统主频时钟频率
Tsys	系统主频时钟周期（1/Fsys）
RAM12K	12KB SRAM
0x	以其开头的数据表示 16 进制数
h	以其结束的数据表示 16 进制数
b	以其结束的数据表示 2 进制数

第 3 章 中断

系统内置快速可编程中断控制器 (PFIC:Programmable Fast Interrupt Controller)，最多支持 255 个中断向量。当前系统管理了 20 个外设中断通道和 8 个内核中断通道，其他中断源保留。

3.1 中断控制器

20 个外设中断，每个中断请求都有独立的触发和屏蔽控制位，有专用的状态位。

1 个不可屏蔽中断 NMI。

特有快速中断进出机制，硬件自动压栈和恢复，无需指令开销。

特有快速中断响应机制，4 路可编程直达中断向量地址。

3.2 系统 SysTick 定时器

内核自带了一个 32 位计数器 (SysTick)，支持 HCLK 或者 HCLK/8 作为时基，具有较高优先级。

3.3 中断和异常向量

下表列出了芯片系统的向量表。

表 3-1 中断向量表

编号	优先级	优先级类型	名称	说明	地址
0	-3	固定	Reset	复位	0x0000 0000
1	-	-	-	保留	0x0000 0004
2	-2	固定	NMI	不可屏蔽中断	0x0000 0008
3	-1	固定	EXC	所有类型的失效、异常中断	0x0000 000C
4	-	-	-	保留	-
5	-1	固定	ECALL-M	机器模式回调中断	0x0000 0014
6-7	-	-	-	保留	-
8	-1	固定	ECALL-U	用户模式回调中断	0x0000 0020
9	-1	固定	BREAKPOINT	断点回调中断	0x0000 0024
10-11	-	-	-	保留	-
12	0	可设置	SysTick	系统滴答定时器	0x0000 0030
13	-	-	-	保留	-
14	1	可设置	SWI	软件中断	0x0000 0038
15	-	-	-	保留	0x0000 003C
16	-	-	-	保留	-
17	2	可设置	GPIO_A	GPIO 端口 PA 通用 I/O 中断	0x0000_0044
18	-	-	-	保留	-
19	3	可设置	SPI	SPI 中断	0x0000_004C
20	4	可设置	BLEB	无线模块的 BB 中断	0x0000_0050
21	5	可设置	BLEL	无线模块的 LLE 中断	0x0000_0054
22	6	可设置	USB	USB 中断	0x0000_0058
23	-	-	-	保留	0x0000_005C
24	7	可设置	TMR	TMR 定时器中断	0x0000_0060
25	-	-	-	保留	-
26	-	-	-	保留	-
27	8	可设置	UART	UART 异步串口中断	0x0000_006C

28	9	可设置	RTC	RTC 和振荡器捕捉完成中断	0x0000_0070
29	10	可设置	CMP	比较器中断	0x0000_0074
30	11	可设置	I2C	I2C 中断	0x0000_0078
31	12	可设置	PWMx	PWMx 中断	0x0000_007C
32	-	-	-	保留	-
33	13	可设置	KEYSCAN	按键扫描中断	0x0000_0080
34	14	可设置	ENCODER	编码器终端	0x0000_0084
35	15	可设置	WDOG_BAT	看门狗定时器中断/电池低电压中断	0x0000_008C

3.4 寄存器

3.4.1 PFIC 寄存器描述

PFIC 相关寄存器基地址: 0xE000E000

表 3-2 PFIC 相关寄存器列表

名称	偏移地址	描述	复位值
R32_PFIC_ISR1	0x00	PFIC 中断使能状态寄存器 1	0x00000032C
R32_PFIC_ISR2	0x04	PFIC 中断使能状态寄存器 2	0x000000000
R32_PFIC_IPR1	0x20	PFIC 中断挂起状态寄存器 1	0x000000000
R32_PFIC_IPR2	0x24	PFIC 中断挂起状态寄存器 2	0x000000000
R32_PFIC_IHRESDR	0x40	PFIC 中断优先级阈值配置寄存器	0x000000000
R32_PFIC_RESTSYS	0x48	PFIC 软复位寄存器	0x000000000
R32_PFIC_GISR	0x4C	PFIC 中断全局状态寄存器	0x000000000
R32_PFIC_IDCFG	0x50	PFIC 快速中断 ID 配置寄存器	0x000000000
R32_PFIC_FIADDR0	0x60	PFIC 快速中断 0 地址寄存器	0x000000000
R32_PFIC_FIADDR1	0x64	PFIC 快速中断 1 地址寄存器	0x000000000
R32_PFIC_FIADDR2	0x68	PFIC 快速中断 2 地址寄存器	0x000000000
R32_PFIC_FIADDR3	0x6C	PFIC 快速中断 3 地址寄存器	0x000000000
R32_PFIC_IENR1	0x100	PFIC 中断使能设置寄存器 1	0x000000000
R32_PFIC_IENR2	0x104	PFIC 中断使能设置寄存器 2	0x000000000
R32_PFIC_IRER1	0x180	PFIC 中断使能清除寄存器 1	0x000000000
R32_PFIC_IRER2	0x184	PFIC 中断使能清除寄存器 2	0x000000000
R32_PFIC_IPSR1	0x200	PFIC 中断挂起设置寄存器 1	0x000000000
R32_PFIC_IPSR2	0x204	PFIC 中断挂起设置寄存器 2	0x000000000
R32_PFIC_IPRR1	0x280	PFIC 中断挂起清除寄存器 1	0x000000000
R32_PFIC_IPRR2	0x284	PFIC 中断挂起清除寄存器 2	0x000000000
R32_PFIC_IACTR1	0x300	PFIC 中断激活状态寄存器 1	0x000000000
R32_PFIC_IACTR2	0x304	PFIC 中断激活状态寄存器 2	0x000000000
R32_PFIC_IPR10Rx	0x400	PFIC 中断优先级配置寄存器	0x000000000
R32_PFIC_SCTLR	0xD10	PFIC 系统控制寄存器	0x000000000

用户模式下，可以支持全局中断控制，请参考 EVT 评估板资料中提供的示例。

内核中断控制位说明：

- 1、Reset、NMI、EXC、ECALL-M、ECALL-U、BREAKPOINT 中断默认总是开启。
- 2、NMI、EXC 支持中断挂起清除和置位控制 (PFIC_IPSR1 和 PFIC_IPRR1 寄存器控制)，不支持中断使能设置和清除控制。
- 3、Reset、ECALL-M、ECALL-U、BREAKPOINT 不支持中断挂起清除和置位控制、中断使能设置和清除控制。

PFIC 中断使能状态寄存器 1 (R32_PFIC_ISR1)

位	名称	访问	描述	复位值
[31:12]	INTENSTA	R0	31#及以下中断当前使能状态。 1: 当前编号中断已使能; 0: 当前编号中断未启用。	00000h
[11:0]	Reserved	R0	保留。 Reset、NMI、EXC、ECALL 等中断位, 下同。	32Ch

PFIC 中断使能状态寄存器 2 (R32_PFIC_ISR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0000000h
[3:0]	INTENSTA	R0	32#及以上中断当前使能状态。 1: 当前编号中断已使能; 0: 当前编号中断未启用。	0h

PFIC 中断挂起状态寄存器 1 (R32_PFIC_IPR1)

位	名称	访问	描述	复位值
[31:12]	PENDSTA	R0	31#及以下中断当前挂起状态。 1: 当前编号中断已挂起; 0: 当前编号中断未挂起。	00000h
[11:0]	Reserved	R0	保留。	000h

PFIC 中断挂起状态寄存器 2 (R32_PFIC_IPR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0000000h
[3:0]	PENDSTA	R0	32#及以上中断当前挂起状态。 1: 当前编号中断已挂起; 0: 当前编号中断未挂起。	0h

PFIC 中断优先级阈值配置寄存器 (R32_PFIC_IHRESDR)

位	名称	访问	描述	复位值
[31:8]	Reserved	R0	保留	000000h
[7:0]	THRESHOLD	RW	中断优先级阈值设置值。 低于当前设置值的中断优先级值, 当挂起时不执行中断服务; 此寄存器为 0 时表示阈值寄存器功能无效。 [7:5]: 优先级阈值; [4:0]: 保留, 固定为 0, 写无效。	00h

PFIC 软复位寄存器 (R32_PFIC_RESTSYS)

位	名称	访问	描述	复位值
[31:16]	KEYCODE	WO	复位关键位, 当关键字匹配时允许执行位 RESETSYS 配置的操作, KEY = 0xBEEF, 否则无响应。	0000h
[15:8]	Reserved	R0	保留。	00h
7	RESETSYS	WO	系统复位, 关关键字匹配时写 1 执行系统复位, 否则无效。	0

[6:0]	Reserved	RO	保留。	00h
-------	----------	----	-----	-----

PFIC 中断全局状态寄存器 (R32_PFIC_GISR)

位	名称	访问	描述	复位值
[31:14]	Reserved	RO	保留。	0
13	CPU_LOCK_STA	RO	当前内核处于锁定状态： 1: 锁定； 0: 非锁定。	0
12	CPU_DBG_MODE	RO	当前内核处于调试状态： 1: 调试； 0: 非调试。	0
11	CPU_GLOBL_IE	RO	全局中断使能： 1: 使能中断； 0: 禁止中断。	0
10	Reserved	RO	保留。	0
9	GPENDSTA	RO	当前是否有中断处于挂起： 1: 有； 0: 没有。	0
8	GACTSTA	RO	当前是否有中断被执行： 1: 有； 0: 没有。	0
[7:0]	NESTSTA	RO	当前中断嵌套状态, 目前支持 2 级嵌套, [1:0] 有效。 3: 第 2 级中断中； 1: 第 1 级中断中； 0: 没有中断发生； 其他: 不可能情况。	0

PFIC 快速中断 ID 配置寄存器 (R32_PFIC_IDCFG)

位	名称	访问	描述	复位值
[31:24]	FIID3	RW	配置快速中断 3 的中断编号。	00h
[23:16]	FIID2	RW	配置快速中断 2 的中断编号。	00h
[15:8]	FIID1	RW	配置快速中断 1 的中断编号。	00h
[7:0]	FIID0	RW	配置快速中断 0 的中断编号。	00h

PFIC 快速中断 0 地址寄存器 (R32_PFIC_FIADDR0)

位	名称	访问	描述	复位值
[31:1]	ADDR0	RW	快速中断 0 服务程序地址 bit[31:1], bit0 为 0。	00000000h
0	FI0EN	RW	快速中断 0 通道使能位： 1: 启用快速中断 0 通道； 0: 关闭。	0

PFIC 快速中断 1 地址寄存器 (R32_PFIC_FIADDR1)

位	名称	访问	描述	复位值
[31:1]	ADDR1	RW	快速中断 1 服务程序地址 bit[31:1], bit0 为 0。	00000000h
0	FI1EN	RW	快速中断 1 通道使能位： 1: 启用快速中断 1 通道；0: 关闭。	0

PFIC 快速中断 2 地址寄存器 (R32_PFIC_FIADDR2)

位	名称	访问	描述	复位值
[31:1]	ADDR2	RW	快速中断 2 服务程序地址 bit[31:1], bit0 为 0。	00000000h
0	FI2EN	RW	快速中断 2 通道使能位: 1: 启用快速中断 2 通道; 0: 关闭。	0

PFIC 快速中断 3 地址寄存器 (R32_PFIC_FIADDR3)

位	名称	访问	描述	复位值
[31:1]	ADDR3	RW	快速中断 3 服务程序地址 bit[31:1], bit0 为 0。	00000000h
0	FI3EN	RW	快速中断 3 通道使能位: 1: 启用快速中断 3 通道; 0: 关闭。	0

PFIC 中断使能设置寄存器 1 (R32_PFIC_IENR1)

位	名称	访问	描述	复位值
[31:12]	INTEN	WO	31#及以下中断使能控制: 1: 当前编号中断使能; 0: 无影响。	00000h
[11:0]	Reserved	RO	保留。	000h

PFIC 中断使能设置寄存器 2 (R32_PFIC_IENR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	RO	保留。	0000000h
[3:0]	INTEN	WO	32#及以上中断使能控制: 1: 当前编号中断使能; 0: 无影响。	0h

PFIC 中断使能清除寄存器 1 (R32_PFIC_IRER1)

位	名称	访问	描述	复位值
[31:12]	INTRESET	WO	31#及以下中断关闭控制: 1: 当前编号中断关闭; 0: 无影响。	00000h
[11:0]	Reserved	RO	保留。	000h

PFIC 中断使能清除寄存器 2 (R32_PFIC_IRER2)

位	名称	访问	描述	复位值
[31:4]	Reserved	RO	保留。	0000000h
[3:0]	INTRESET	WO	32#及以上中断关闭控制: 1: 当前编号中断关闭; 0: 无影响。	0h

PFIC 中断挂起设置寄存器 1 (R32_PFIC_IPSR1)

位	名称	访问	描述	复位值
[31:12]	PENDSET	WO	31#及以下中断挂起设置: 1: 当前编号中断挂起; 0: 无影响。	00000h

[11:0]	Reserved	R0	保留。	000h
--------	----------	----	-----	------

PFIC 中断挂起设置寄存器 2 (R32_PFIC_IPSR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0000000h
[3:0]	PENDSET	WO	32#及以上中断挂起设置。 1: 当前编号中断挂起; 0: 无影响。	0h

PFIC 中断挂起清除寄存器 1 (R32_PFIC_IPRR1)

位	名称	访问	描述	复位值
[31:12]	PENDRESET	WO	31#及以下中断挂起清除。 1: 当前编号中断清除挂起状态; 0: 无影响。	00000h
[11:0]	Reserved	R0	保留。	000h

PFIC 中断挂起清除寄存器 2 (R32_PFIC_IPRR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0000000h
[3:0]	PENDRESET	WO	32#及以上中断挂起清除。 1: 当前编号中断清除挂起状态; 0: 无影响。	0h

PFIC 中断激活状态寄存器 1 (R32_PFIC_IACTR1)

位	名称	访问	描述	复位值
[31:12]	IACTS	RW1	31#及以下中断执行状态。 1: 当前编号中断执行中; 0: 当前编号中断未执行。	00000h
[11:0]	Reserved	R0	保留。	000h

PFIC 中断激活状态寄存器 2 (R32_PFIC_IACTR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0000000h
[3:0]	IACTS	RW1	32#及以上中断执行状态。 1: 当前编号中断执行中; 0: 当前编号中断未执行。	0h

PFIC 中断优先级配置寄存器 (R32_PFIC_IPR10Rx) (x=0-63)

控制器支持 256 个中断 (0-255), 每个中断使用 8bit 来设置控制优先级。

31	24	23	16	15	8	7	0
IPR10R63	PR10_255	PR10_254	PR10_253	PR10_252			

...

IPR10Rx	PR10_(4x+3)	PR10_(4x+2)	PR10_(4x+1)	PR10_(4x)
---------	-------------	-------------	-------------	-----------

...

IPR10R0	PR10_3	PR10_2	PR10_1	PR10_0
---------	--------	--------	--------	--------

位	名称	访问	描述	复位值
[2047:2040]	IP_255	RW	同 IP_0 描述。	00h
...
[31:24]	IP_3	RW	同 IP_0 描述。	00h
[23:16]	IP_2	RW	同 IP_0 描述。	00h
[15:8]	IP_1	RW	同 IP_0 描述。	00h
[7:0]	IP_0	RW	编号 0 中断优先级配置： [7:5]：优先级控制位。 [4:0]：保留，固定为 0，写无效。 优先级数值越小则优先级越高。只有 2 级中断嵌套，即只能抢占 1 次。	00h

PFIC 系统控制寄存器 (R32_PFIC_SCTLR)

位	名称	访问	描述	复位值
31	SYSRESET	WO	系统复位。自动清 0。写 1 有效，写 0 无效。	0
[30:6]	Reserved	RO	保留。	0
5	SETEVENT	WO	设置事件，可以唤醒 WFE 的情况。	0
4	SEVONPEND	RW	设置新的中断进入挂起态为唤醒事件，可以从 WFI 指令后唤醒系统，如果未执行 WFE 指令，将在下次执行该指令后立即唤醒系统。 1：新的中断进入挂起态为唤醒事件； 0：新的中断进入挂起态不作为唤醒事件。	0
3	WFI2WFE	RW	将 WFI 指令当成是 WFE 执行。 1：将之后的 WFI 指令当做 WFE 指令； 0：无作用。	0
2	SLEEPDEEP	RW	控制系统的低功耗模式： 1：deep sleep； 0：sleep。	0
1	SLEEPONEXIT	RW	控制离开中断服务程序后，系统状态： 1：系统进入低功耗模式； 0：系统进入主程序。	0
0	Reserved	RO	保留。	0

3.4.2 WCH 定义 CSR 寄存器

RISC-V 架构中定义了一些控制和状态寄存器 (Control and Status Register, CSR)，用于配置或标识或记录运行状态。CSR 寄存器属于内核内部的寄存器，使用专用的 12 位地址空间。WCH 芯片除了 RISC-V 特权架构文档中定义的标准寄存器外，还增加了一些厂商定义寄存器，需要使用 csr 指令进行访问。

注：部分 CSR 寄存器需要系统在机器模式下访问，非机器模式下访问这些 CSR 寄存器将导致芯片进入异常。标注为“MRW”：需要系统在机器模式下才能访问；标注为“MRO”：在机器模式下只读。

中断系统控制寄存器 (INTSYSSCR)

CSR 地址：0x804

位	名称	访问	描述	复位值
31	LOCK	MRW	用户模式锁定标志，锁定后用户模式不可开启	0

			本组寄存器配置。 1: 锁定, 仅机器模式可配置; 0: 非锁定, 机器/用户模式可配置。	
[30:6]	Reserved	RO	保留。	0
[5]	HW_POP_OFF	MRW	一次性硬件出栈关闭, 退出中断后自动复位。 1: 下次退出中断时屏蔽硬件出栈; 0: 不屏蔽硬件出栈。	0
4	Reserved	RO	保留。	0
[3:2]	PREEMPT	MRW	抢占优先级位宽配置寄存器, 用于配置中断优先级中抢占优先级位宽。 00: 抢占位宽为 0, 任何优先级的中断无法嵌套; 01: 抢占位宽为 1, 即中断优先级寄存器的位[7]为抢占优先级; 10: 抢占位宽为 2, 即中断优先级寄存器的位[7:6]为抢占优先级; 11: 抢占位宽为 3, 即中断优先级寄存器的位[7:5]为抢占优先级。	01b
1	INESTEN	MRW	中断嵌套功能使能。 1: 使能; 0: 关闭。	0
0	HWSTKEN	MRW	硬件压栈保护使能。 1: 进入中断时使用硬件堆栈保护; 0: 进入中断时不使用硬件堆栈保护。	0

注: 中断嵌套功能的开启和关闭, 由寄存器 INESTCR 的位 NEST_LVL 控制。

用户访问机器状态寄存器 (USER_ACCESS_MSTATUS)

CSR 地址: 0x800

寄存器功能设置同 MSTATUS 寄存器。

位	名称	访问	描述	复位值
[31:13]	Reserved	RO	保留。	0
[12:11]	MPP	RO	退出异常时机器状态: 00: 退出异常时机器状态设置为 U 模式; 11: 退出异常时机器状态设置为 M 模式; 01: 保留; 10: 保留。	0
[10:8]	Reserved	RO	保留。	0
7	MPIE	RW	当寄存器 CORECFG 的位 IE_REMAP_EN 使能时, 该位在用户模式下可读可写。	0
[6:4]	Reserved	RO	保留。	0
3	MIE	RW	当寄存器 CORECFG 的位 IE_REMAP_EN 使能时, 该位在用户模式下可读可写。	0
[2:0]	Reserved	RO	保留。	0

机器状态寄存器 (MSTATUS)

CSR 地址: 0x300

位	名称	访问	描述	复位值
[31:13]	Reserved	MRO	保留。	0
[12:11]	MPP	MRW	退出异常时机器状态:	0

			00: 退出异常时机器状态设置为 U 模式; 11: 退出异常时机器状态设置为 M 模式; 01: 保留; 10: 保留。	
[10:8]	Reserved	MRO	保留。	0
7	MPIE	MRW	退出中断后, 全局中断使能 (进入中断时更新为进入 MIE 值): 1: 退出中断后, 使能全局中断; 0: 退出中断后, 关闭全局中断。	0
[6:4]	Reserved	MRO	保留。	0
3	MIE	MRW	进入中断时, 全局中断使能, (退出中断时更新为 MPIE 值): 1: 使能全局中断; 0: 关闭全局中断。	0
[2:0]	Reserved	MRO	保留。	0

异常入口地址寄存器 (MTVEC)

CSR 地址: 0x305

位	名称	访问	描述	复位值
[31:2]	BASEADD	MRW	中断向量表基地址。	00000000h
1	MODE1	MRW	中断向量表识别模式。 1: 按绝对地址识别, 支持全范围, 但必须跳转; 0: 按跳转指令识别, 有限范围, 支持非跳指令。	0
0	MODE0	MRW	中断或异常入口地址模式选择: 1: 根据其编号*4 进行地址偏移; 0: 使用统一入口地址。	0

微处理器配置寄存器 (CORECFGR)

CSR 地址: 0xBC0

位	名称	访问	描述	复位值
[31:8]	Reserved	MRO	保留。	0
7	HF_NMI	MRW	硬件错误产生 NMI: 1: 硬件错误时, 产生 NMI 中断; 0: 硬件错误时, 产生异常中断; 注: 当前硬件错误仅包括嵌套溢出。	0
6	Reserved	MRO	保留。	0
5	IE_REMAP_EN	MRW	MIE 寄存器映射使能: 1: CSR 地址 0x800 的位 3 和位 7 分别映射为 MSTATUS 寄存器的位 MIE 和 MSTATUS 寄存器的位 MPIE; 0: CSR 地址 0x800 为只读寄存器, 返回值为 MSTATUS 的值。	0
4	Reserved	MRO	保留。	0
3	ROM_CACHE_EN	MRW	ROM 区指令缓存使能: 1: ROM 区域指令缓存使能, 最多可缓存 128 字节指令; 0: ROM 区域指令缓存关闭。	0
2	ROM_JUMP_ACC	MRW	ROM 区指令跳转加速使能:	0

			1: 使能 ROM 区指令跳转加速; 0: 关闭 ROM 区指令跳转加速。	
[1:0]	FETCH_MODE	MRW	<p>取指模式:</p> <p>00: 预取关闭。指令预取功能关闭, 避免无效的取指操作, CPU 流水线上至多存在 1 条有效指令。此模式功耗最低, 性能下降约 2~3 倍;</p> <p>01: 预取使能。指令预取功能打开, CPU 将持续对指令存储器进行访问, 直到内部指令缓冲器待执行指令数量超过一定数量, 或者指令缓冲器已满时, 暂停取指令, 此模式功耗高, 性能强; (CPU 预测失败将引入冗余取指操作, 在部分情况下, 执行单元将额外引入 0~2 个周期的气泡, 大部分程序性能下降不明显);</p> <p>其他: 保留。</p>	01b

中断嵌套控制寄存器 (INESTCR)

CSR 地址: 0xBC1

位	名称	访问	描述	复位值
31	Reserved	MRO	保留。	0
30	NEST_OVR	MRW	<p>硬件错误中断嵌套溢出标志位, 写 1 清零:</p> <p>1: 中断溢出标志;</p> <p>0: 中断未溢出。</p> <p>注: 中断溢出仅会在执行二级中断服务函数产生指令异常和 NMI 中断时发生。此时异常和 NMI 中断正常进入, 但是 CPU 堆栈溢出, 不可从此异常和 NMI 中断退出。</p>	0
[29:12]	Reserved	MRO	保留。	0
[11:8]	NEST_STA	MRO	<p>嵌套状态标志位:</p> <p>0000: 无中断;</p> <p>0001: 一级中断;</p> <p>0011: 二级中断, (一级嵌套);</p> <p>0111: 三级中断, (二级嵌套, 发生溢出)。</p>	0
[7:2]	Reserved	MRO	保留。	0
[1:0]	NEST_LVL	MRW	<p>嵌套等级:</p> <p>00: 禁止嵌套, (关闭嵌套功能);</p> <p>01: 一级嵌套, (打开嵌套功能);</p> <p>10: 无法写入;</p> <p>11: 无法写入。</p> <p>注 1: 对该域写 10 或者 11, 寄存器被置为 01;</p> <p>注 2: 写入 11 时, 读该寄存器, 可获得芯片的最高嵌套等级。</p>	0

3.4.3 物理内存保护单元 PMP

为了提高系统安全, RISC-V 架构中定义了一套物理地址访问限制, 可以为区域内物理内存设置其读、写、执行属性, 区域长度最小 4 字节保护。PMP 单元在用户模式下一直生效, 在机器模式下可选生效, 如果违背了当前内存限制, 将会产生系统异常中断 (EXC)。

PMP 单元包含 4 组 8bit 的配置寄存器 (合计 32bit) 和 4 组地址寄存器, 需要使用 csr 指令进行访问, 并且在机器模式下进行。

PMP 配置寄存器 (PMPCFG0)

CSR 地址: 0x3A0

位	名称	访问	描述	复位值																					
[31:24]	pmp3cfg	MRW	见 pmp0cfg。	00h																					
[23:16]	pmp2cfg	MRW	见 pmp0cfg。	00h																					
[15:8]	pmp1cfg	MRW	见 pmp0cfg。	00h																					
[7:0]	pmp0cfg	MRW	<table border="1" style="display: inline-table; vertical-align: middle;"> <thead> <tr> <th>位</th> <th>名称</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>L</td> <td>锁定使能, 机器模式下可解锁: 1: 锁定相关寄存器; 0: 不锁定。</td> </tr> <tr> <td>[6:5]</td> <td>-</td> <td>保留。</td> </tr> <tr> <td>[4:3]</td> <td>A</td> <td>地址对齐及保护区域范围选择。</td> </tr> <tr> <td>2</td> <td>X</td> <td>可执行属性。</td> </tr> <tr> <td>1</td> <td>W</td> <td>可写入属性。</td> </tr> <tr> <td>0</td> <td>R</td> <td>可读出属性</td> </tr> </tbody> </table>	位	名称	描述	7	L	锁定使能, 机器模式下可解锁: 1: 锁定相关寄存器; 0: 不锁定。	[6:5]	-	保留。	[4:3]	A	地址对齐及保护区域范围选择。	2	X	可执行属性。	1	W	可写入属性。	0	R	可读出属性	00h
位	名称	描述																							
7	L	锁定使能, 机器模式下可解锁: 1: 锁定相关寄存器; 0: 不锁定。																							
[6:5]	-	保留。																							
[4:3]	A	地址对齐及保护区域范围选择。																							
2	X	可执行属性。																							
1	W	可写入属性。																							
0	R	可读出属性																							

地址对齐及保护区域范围选择, 对于 $A_ADDR \leq region < B_ADDR$ 区域进行内存保护 (要求 A_ADDR 和 B_ADDR 均 4 字节对齐):

- 1、如果 $B_ADDR - A_ADDR == 2^2$, 则采用 NA4 方式;
- 2、如果 $B_ADDR - A_ADDR == 2^{(G+2)}$, $G \geq 1$, 且 A_ADDR 为 $2^{(G+2)}$ 对齐则采用 NAPOT 方式;
- 3、否则使用 TOR 方式。

A 值	方式	描述
0	OFF	没有区域要保护
1	TOR	顶端对齐区域保护: pmp0cfg 下, $0 \leq region < pmpaddr0$; pmp1cfg 下, $pmpaddr0 \leq region < pmpaddr1$; pmp2cfg 下, $pmpaddr1 \leq region < pmpaddr2$; pmp3cfg 下, $pmpaddr2 \leq region < pmpaddr3$ 。 $pmpaddr_{i-1} = A_ADDR \gg 2$; $pmpaddr_i = B_ADDR \gg 2$ 。
2	NA4	固定 4 字节区域保护。 pmp0cfg~pmp3cfg 对应 $pmpaddr0 \sim pmpaddr3$ 作为起始地址。 $pmpaddr_i = A_ADDR \gg 2$ 。
3	NAPOT	保护 $2^{(G+2)}$ 区域, $G \geq 1$, 其中 A_ADDR 为 $2^{(G+2)}$ 对齐。 $pmpaddr_i = (A_ADDR \gg 2) (2^{(G-1)} - 1)$ 。

PMP 地址 0 寄存器 (PMPADDR0)

CSR 地址: 0x3B0

位	名称	访问	描述	复位值
[31:0]	ADDR0	MRW	PMP 设置地址 0 的 bit[33:2]。实际高 2 位未用	00000000h

PMP 地址 1 寄存器 (PMPADDR1)

CSR 地址: 0x3B1

位	名称	访问	描述	复位值
[31:0]	ADDR1	MRW	PMP 设置地址 1 的 bit[33:2]。实际高 2 位未用	00000000h

PMP 地址 2 寄存器 (PMPADDR2)

CSR 地址: 0x3B2

位	名称	访问	描述	复位值
[31:0]	ADDR2	MRW	PMP 设置地址 2 的 bit[33:2]。实际高 2 位未用	00000000h

PMP 地址 3 寄存器 (PMPADDR3)

CSR 地址: 0x3B3

位	名称	访问	描述	复位值
[31:0]	ADDR3	MRW	PMP 设置地址 3 的 bit[33:2]。实际高 2 位未用	00000000h

3.4.4 SysTick 寄存器描述

STK 相关寄存器基地址: 0xE000F000

表 3-3 STK 相关寄存器列表

名称	偏移地址	描述		复位值
R32_STK_CTRL	0x00	系统计数控制寄存器		0x00000000
R32_STK_SR	0x04	系统计数状态寄存器		0x00000000
R32_STK_CNTL	0x08	系统计数器低位寄存器		0x00000000
R32_STK_CMPLR	0x10	计数重加载低位寄存器		0x00000000

系统计数控制寄存器 (R32_STK_CTRL)

位	名称	访问	描述	复位值
31	SWIE	RW	软件中断触发使能 (SWI)，同 STK_SR: 1: 触发软件中断； 0: 关闭触发。 注: 进入软件中断后须将此位清 0, 否则将一直触发。	0
[30:5]	Reserved	RO	保留。	0000000h
4	MODE	RW	计数模式: 1: 向下计数； 0: 向上计数。	0
3	STRE	RW	自动重装载计数使能: 1: 向上计数到比较值之后从 0 开始计数, 向下计数到 0 之后从比较值开始向下计数； 0: 继续向上/向下计数。	0
2	STCLK	RW	计数器时钟源选择: 1: HCLK 做时基； 0: HCLK/8 做计数时基。	0
1	STIE	RW	计数器中断使能控制位: 1: 使能计数器中断； 0: 无计数器中断。	0
0	STE	RW	系统计数器使能控制位: 1: 启动系统计数器 STK； 0: 关闭系统计数器 STK, 计数器停止计数。	0

计数状态寄存器 (R32_STK_SR)

位	名称	访问	描述	复位值
31	SWIE	RW	软件中断触发使能 (SWI): 1: 触发软件中断； 0: 关闭触发。	0

			注：进入软件中断后须将此位清 0，否则将一直触发。	
[30:1]	Reserved	RO	保留。	0
0	CNTIF	RW	计数值比较标志，写 0 清除，写 1 无效： 1：向上计数达到比较值，向下计数达到 0； 0：未达到比较值。	0

系统统计数器低位寄存器 (R32_STK_CNTL)

位	名称	访问	描述	复位值
[31:0]	CNTL	RW	STK 计数器计数值低 32 位。	00000000h

计数重加载低位寄存器 (R32_STK_CMPLR)

位	名称	访问	描述	复位值
[31:0]	CMPL	RW	设置重加载计数器值低 32 位。	00000000h

3.4.5 硬件断点设置

处理器支持 4 路指令地址和数据地址断点，其中 TDATA1 寄存器的位 TYPE 固定值为 2，而 TDATA1 寄存器中其它位符合调试标准中 mcontrol 定义，最多支持四通道断点。TSELECT 触发器低 2 位有效，通过 TSELECT 寄存器的值来选择中断点通道，再配置断点地址和控制信息。

断点通道选择寄存器 (TSELECT)

CSR 地址：0x7A0

位	名称	访问	描述	复位值
[31:2]	Reserved	MRO	保留。	0
[1:0]	TSELECT	MRW	断点通道选择寄存器，配置后即选中对应通道，可操作 TDATA1 和 TDATA2 寄存器配置断点信息。	X

断点通道控制寄存器 (TDATA1) (上电复位)

CSR 地址：0x7A1

位	名称	访问	描述	复位值
[31:28]	TYPE	MRO	断点类型定义，mcontrol 类型。	0010b
[27]	DMODE	MRO	调试模式和机器模式可以修改触发器的相关寄存器。	0
[26:21]	MASKMAX	MRO	当 MATCH = 1 时，允许匹配的最大指数幂范围，即最大允许匹配 2^{31} 字节范围。	011111b
[20:13]	Reserved	MRO	保留。	0
12	ACTION	MRW	设置触发断点时采取的处理模式： 1：触发时进入到调试模式； 0：触发时进入到断点回调中断。	0
[11:8]	Reserved	MRO	保留。	0
7	MATCH	MRW	匹配策略配置： 1：触发值和 TDATA2 的高 M 位相等时匹配，其中 M = 31 - N，N 为 TDATA2 的第一个 0 的索引（从低位开始）； 0：触发值和 TDATA2 相等时匹配。	0
6	M	MRW	M 模式下触发器使能： 1：M 模式下使能触发器； 0：M 模式下关闭触发器。	0

3	U	MRW	U 模式下触发器使能： 1: U 模式下使能触发器； 0: U 模式下关闭触发器。	0
2	EXECUTE	MRW	指令读地址触发使能： 1: 使能； 0: 关闭。	0
1	STORE	MRW	数据写地址触发使能： 1: 使能； 0: 关闭。	0
0	LOAD	MRW	数据读地址触发使能： 1: 使能； 0: 关闭。	0

断点通道地址寄存器 (TDATA2)

CSR 地址: 0x7A2

位	名称	访问	描述	复位值
[31:0]	TDATA2	MRW	用于保存匹配值。	X

第 4 章 系统控制

4.1 复位控制

系统支持 6 种复位形式，分别为电源上电复位 RPOR (real power on reset)、外部手工复位 MR (manual reset)、内部软件复位 SR (software reset)、看门狗复位 (watch-dog reset)、下电模式下唤醒导致的全局复位 GRWSM (global reset by waking under shutdown mode)、常规唤醒导致的局部寄存器复位 LRW (local reset by waking)。

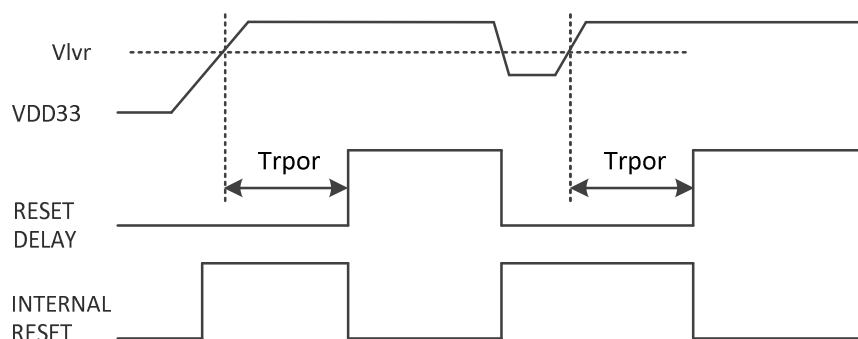
寄存器 R8_GLOB_RESET_KEEP 和 RB_ROM_CODE_OFS，只在 RPOR 或者 GRWSM 复位时被复位，而不受其它复位形式影响。

下图中时序参数和复位特性参数请参考第 20.5 节的时序参数表。

4.1.1 电源上电复位 RPOR

当电源上电时，芯片内部 POR 模块会产生上电复位，并延时以等待电源稳定。另外，在运行过程中，当电源电压低于 V_{lvr} 时，芯片内部 LVR 模块会产生低压复位直到电压回升，并延时以等待电源稳定。下图为上电复位过程和低压复位过程。

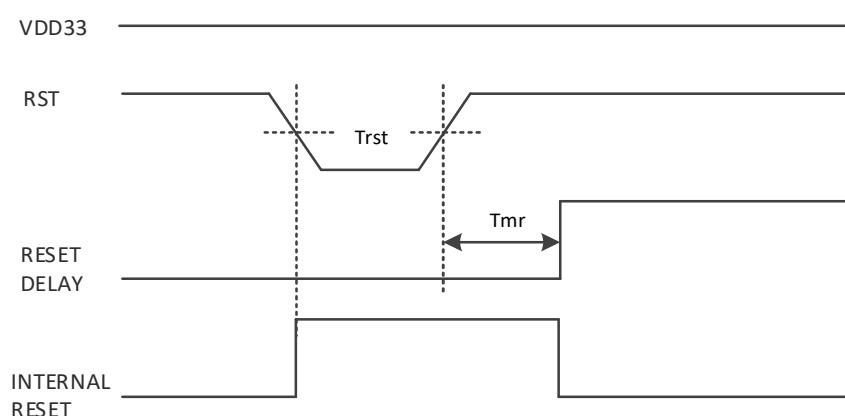
图 4-1 上电复位



4.1.2 外部手工复位 MR

外部手工复位由外部加到 RST 引脚的低电平触发，当复位低电平持续时间大于最小复位脉冲宽度 (T_{rst}) 时即触发系统进行复位。

图 4-2 外部复位



4.1.3 内部软件复位 SR

内部软件复位，用于不需要外部干预而自行复位。设置全局复位配置寄存器 (R8_RST_WDOG_CTRL) 的位 RB_SOFTWARE_RESET 为 1，即可实现软件复位。该位会自动清 0。

4.1.4 看门狗复位 (WTR、IWDG)

看门狗复位包括看门狗超时复位和独立看门狗复位两种方式。看门狗超时复位功能 (WTR) 是基于一个 8 位的递增计数器, 计数时钟周期为 $131072/F_{sys}$, 当开启了看门狗超时复位功能后, 一旦此计数器溢出会复位整个系统。

独立看门狗 (IWDG) 内部是一个递减运行的 12 位计数器, 当计数器的值减为 0 时, 将会产生系统复位。

4.1.5 下电模式唤醒后的复位 GRWSM

一旦系统进入下电模式 (详见电源管理章节) 后, 在唤醒信号的作用下, 系统将有序执行唤醒操作, 唤醒之后系统将执行全局复位, 此复位效果类似于上电复位。

4.1.6 常规唤醒操作引起的复位 LRW

如果系统是从睡眠模式中被唤醒, 则在相关电源准备就绪之后会产生复位, 此复位为局部复位, 根据需要对睡眠模式下掉电的寄存器进行有选择的复位。

在睡眠模式下, 各功能模块的寄存器分为三类:

第一种是属于需数据保持的功能模块的关键寄存器 (例如配置/模式等), 睡眠时由辅助电源继续供电, 数据不丢失, 睡眠和唤醒对其数据无影响;

第二种是属于需数据保持的功能模块的可再生寄存器 (例如计数器、FIFO 等), 睡眠时断电, 唤醒后数据是随机数 (例如 FIFO 存储单元) 或者被复位 (例如 FIFO 计数器);

第三种是属于无需数据保持的功能模块的寄存器, 睡眠时断电, 唤醒后数据是随机数 (例如 FIFO 存储单元) 或者被复位 (例如 FIFO 计数器、配置/模式寄存器)。

LRW 就是用于上述后两种被复位的寄存器。

4.2 安全访问

系统某些寄存器的属性是 “RWA” 或者 “WA”, 表示当前寄存器为安全访问寄存器, 可以直接读取但是写入需要进入安全访问模式。

先写入 R8_SAFE_ACCESS_SIG 寄存器 0x57;

再写入 R8_SAFE_ACCESS_SIG 寄存器 0xA8;

即可进入安全访问模式, 此时可以操作具有 “RWA/WA” 属性的寄存器, 此后约 112 个系统主频周期 (T_{sys}) 都处于安全模式下, 该有效期内可以改写一个或多个安全类寄存器, 超出上述有效期后将自动终止安全模式。或者可提前向 R8_SAFE_ACCESS_SIG 寄存器写入 0x00 提前终止安全模式。

4.3 寄存器描述

表 4-1 系统控制相关寄存器列表

名称	访问地址	描述	复位值
R8_SAFE_ACCESS_SIG	0x40001040	安全访问标记寄存器	0x00
R8_CHIP_ID	0x40001041	芯片 ID 寄存器	0x72/0x70
R8_SAFE_ACCESS_ID	0x40001042	安全访问 ID 寄存器	0x0C
R8_WDOG_COUNT	0x40001043	看门狗计数器寄存器	0x00
R8_RESET_STATUS	0x40001044	复位状态寄存器	0x01
R8_GLOB_ROM_CFG	0x40001044	FlashROM 应用配置寄存器	0x01
R8_GLOB_CFG_INFO	0x40001045	全局配置信息状态寄存器	0xX8
R8_RST_WDOG_CTRL	0x40001046	看门狗及复位配置寄存器	0x00
R8_GLOB_RESET_KEEP	0x40001047	复位保持寄存器	0x00
R32_SAFE_ACCESS_SIG2	0x40001058	安全访问标记 2 寄存器	0x07000000
R32_FLASH_DATA	0x40001800	FlashROM 字数据寄存器	0xFFFFFFFF
R32_FLASH_CONTROL	0x40001804	FlashROM 控制寄存器	0x074000XX

R8_FLASH_DATA	0x40001804	FlashROM 字节数据寄存器	0xXX
R8_FLASH_CTRL	0x40001806	FlashROM 存取控制寄存器	0x40
R8_FLASH_CFG	0x40001807	FlashROM 存取配置寄存器	0x07

安全访问标记寄存器 (R8_SAFE_ACCESS_SIG)

位	名称	访问	描述	复位值
[7:0]	R8_SAFE_ACCESS_SIG	WO	安全访问标记寄存器。 部分寄存器(访问属性为 RWA)为保护寄存器, 必须进入安全访问模式才能进行写操作。对该寄存器先写入 0x57, 再写入 0xA8, 即可进入安全访问模式, 并且限时约 112(7*16)个主时钟周期(Tsys), 超过则自动保护。可以写入其它任意值强制直接退出安全访问模式, 回到保护状态。	00h
[7:3]	RB_SAFE_ACC_TIMER	RO	当前安全访问时间计数。	000b
2	RB_SAFE_ACC_ACT	RO	当前安全访问模式状态: 1: 未锁定/安全访问模式下, 可写; 0: 锁定, RWA 属性寄存器不可改写。	0
[1:0]	RB_SAFE_ACC_MODE	RO	当前安全访问模式状态: 11: 安全模式, 可写入属性 RWA 寄存器; 其他: 非安全模式。	00b

芯片 ID 寄存器 (R8_CHIP_ID)

位	名称	访问	描述	复位值
[7:0]	R8_CHIP_ID	RF	CH572: 固定值 72h, 用于识别芯片。 CH570: 固定值 70h, 用于识别芯片。	72h 70h

安全访问 ID 寄存器 (R8_SAFE_ACCESS_ID)

位	名称	访问	描述	复位值
[7:0]	R8_SAFE_ACCESS_ID	RF	固定值 0Ch。	0Ch

看门狗计数器寄存器 (R8_WDOG_COUNT)

位	名称	访问	描述	复位值
[7:0]	R8_WDOG_COUNT	RW	可预置初值的看门狗计数器, 一直自动递增, 可从 0xFF 循环到 0x00 再继续。 计数周期=131072/Fsys。	00h

复位状态寄存器 (R8_RESET_STATUS)、FlashROM 应用配置寄存器 (R8_GLOB_ROM_CFG)

位	名称	访问	描述	复位值
[7:6]	RB_ROM_CODE_WE	RWA	FlashROM 程序存储区 CodeFlash 的擦除、编程使能位: X0: 全部擦写保护; 01: 129~240K 允许擦写; 11: 0~240K 允许擦写。	0
5	RB_ROM_CTRL_EN	RWA	FlashROM 存取控制接口使能: 1: 允许控制; 0: 禁止存取。	0
4	RB_ROM_CODE_OFS	RWA	选择用户程序代码在 FlashROM 的起始	0

			位置偏移地址： 1: 0x008000; 0: 0x000000。	
3	Reserved	R0	保留。	0
[2:0]	RB_RESET_FLAG	R0	最近一次复位状态： 000: 软件复位 SR (RB_WDOG_RST_EN=0 时软件复位可产生此状态, 否则可复位但不产生此状态); 001: 上电复位 RPOR; 010: 看门狗超时复位 WTR; 011: 外部手动复位 MR; 101: 从下电模式唤醒时的复位 GRWSM; 100/110/111: 唤醒复位 LRW, 且此前的上一次复位分别是 SR/WTR/MR。	001b

全局配置信息状态寄存器 (R8_GLOB_CFG_INFO)

位	名称	访问	描述	复位值
[7:6]	Reserved	R0	保留。	11b
5	RB_BOOT_LOADER	R0	Bootloader 状态： 1: 当前处于 Bootloader 状态; 0: 当前处于用户程序状态。	1/0
4	RB_CFG_RST_PIN	R0	复位引脚选择： 1: PA7; 0: PA8。	0
3	RB_CFG_BOOT_EN	R0	系统引导程序 BootLoader 使能状态： 1: 已启用; 0: 未启用。	1
2	RB_CFG_RESET_EN	R0	RST 外部手动复位输入使能状态： 1: 已启用; 0: 未启用。	0
1	Reserved	R0	保留。	0
0	Reserved	R0	保留。	0

看门狗及复位配置寄存器 (R8_RST_WDOG_CTRL)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	000b
4	RB_WDOG_INT_FLAG	RW1	看门狗定时器中断标志： 1: 发生了看门狗计数溢出, 即检测到 R8_WDOG_COUNT 递增 0xFF 到 0x00; 0: 看门狗计数未溢出。 标志写 1 清 0, 或者重新加载看门狗计数器值 (R8_WDOG_COUNT) 清 0, 或者执行 __SEV() 清 0。	0
3	Reserved	R0	保留。	0
2	RB_WDOG_INT_EN	RWA	看门狗定时器中断使能位： 1: 使能, 看门狗计数溢出后产生中断; 0: 关闭看门狗定时器中断。	0
1	RB_WDOG_RST_EN	RWA	看门狗超时复位使能位：	0

			1: 使能, 看门狗计数溢出后系统复位; 0: 仅作为看门狗定时器。 注: 此位置 1 后软件复位操作将不影响 RB_RESET_FLAG 状态。	
0	RB_BOOT_LOAD_MAN	RO	手动进入 BOOT LOADER 标志: 1: 手动进入 BOOT; 0: 空闲, 无动作。	0
0	RB_SOFTWARE_RESET	WA/ WZ	系统软件复位控制, 复位后将自动清零: 1: 执行系统软件复位。 (当配置 RB_WDOG_INT_EN 与 RB_ROM_CODE_WE 为 1、RB_ROM_DATA_WE 为 0、 $128 \leq R8_WDOG_COUNT < 192$ 时, 位 RB_BOOT_LOAD_MAN 为 1 并且从 BOOT 启动, 否则位 RB_BOOT_LOAD_MAN 为 0)	0

复位保持寄存器 (R8_GLOB_RESET_KEEP)

位	名称	访问	描述	复位值
[7:0]	R8_GLOB_RESET_KEEP	RW	复位保持寄存器, 该寄存器值不受手动复位、软件复位、看门狗复位或者普通唤醒复位的影响。	00h

FlashROM 安全访问标志 2 寄存器 (R32_SAFE_ACCESS_SIG2)

位	名称	访问	描述	复位值
[31:27]	Reserved	RO	保留。	0
[26:24]	RB_FUN_MODE	RO	功能使能寄存器。	111b
23	RB_FLASH_HALTED	RO	两线 Flash 禁止操作标志: 1: 禁止操作; 0: 允许操作。	0
22	RB_MANU_CFG_LOCK	RO	厂商配置字锁定标志: 1: 已锁定; 0: 未锁定。	0
21	RB_RD_PROTECT	RO	读保护标志: 1: 开启; 0: 关闭。	0
20	RB_SAFE_AC_DIS	RO	安全寄存器自动关闭标志: 1: 自动关闭模式已关闭; 0: 自动关闭模式仍在生效。	0
[19:0]	Reserved	RO	保留。	0

有关 FlashROM 的操作或者设置可以参考相关子程序, 本手册不提供有关 FlashROM 字数据寄存器和 FlashROM 控制寄存器的说明。

4.4 Flash-ROM 操作步骤

1. 擦除 Flash-ROM, 请参考和调用相关子程序。
2. 写 Flash-ROM, 请参考和调用相关子程序。
3. 读 DataFlash, 请参考和调用相关子程序。
4. 读 CodeFlash, 通过指向程序存储空间的指针。

4.5 芯片唯一 ID 号

每个芯片出厂时都具有唯一的 ID 号，即芯片身份识别号。该 ID 数据及其校验和共 8 字节，存储于芯片内部只读区域中，具体操作请参考例子程序。

第 5 章 电源控制

5.1 电源管理

CH572 内置有电源管理单元 PMU。

单一 5V 电源供电时, 需在 V5 与 VDD33 引脚之间串接额定 $1.5\text{ k}\Omega$ 电阻。V5 引脚输入额定 5V 电源, 为内部 LD05V 供电, 由 LD05V 输出 3.3V, 再由内置的多个 LD0 电压调整器为系统的 FlashROM、系统的数字电路 (包括内核和 USB 等) 和系统的模拟电路 (包括高频振荡器、PLL、CMP 和 RF 收发器等) 提供所需的电源。5V 电源供电时不支持暂停、睡眠或下电这 3 种低功耗模式。

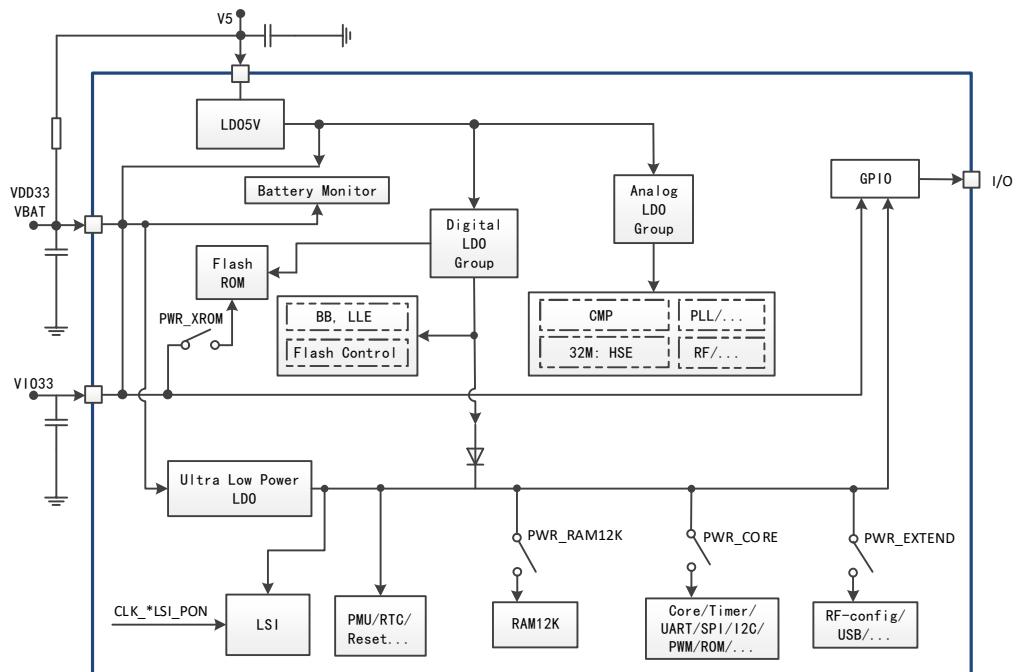
单一 3.3V 电源供电时, V5 引脚悬空或者短接 VDD33。VDD33 引脚输入额定 3.3V 电源, 由内置的多个 LD0 电压调整器为系统的各个模块提供所需的电源。

GPIO 和 FlashROM 的电源从 VI033 输入。

正常工作时的电源供电为: 直接电源。在正常工作之外, CH572 提供了 4 种低功耗模式: 空闲模式、暂停模式、睡眠模式、下电模式。

为了降低睡眠时的系统功耗, 可以选择关闭系统主 LD0, 切换成由系统内置的超低功耗 ULP-LD0 提供辅助电源。当系统进入睡眠或下电模式时, 除了电源管理和 RTC 寄存器等常供电单元外, 系统 12KB 的 SRAM、内核及所有的外设皆可选择是否维持供电, LSI 可选择是否开启。

图 5-1 电源系统



5.2 寄存器描述

表 5-1 功耗管理相关寄存器列表

名称	访问地址	描述	复位值
R16_SLP_CLK_OFF	0x4000100C	睡眠时钟控制寄存器	0x0000
R8_SLP_CLK_OFF0	0x4000100C	睡眠时钟控制寄存器 0	0x00
R8_SLP_CLK_OFF1	0x4000100D	睡眠时钟控制寄存器 1	0x00
R8_SLP_WAKE_CTRL	0x4000100E	唤醒事件配置寄存器	0x20
R8_SLP_POWER_CTRL	0x4000100F	外设睡眠电源控制寄存器	0x00
R8_LONG_RST_CFG	0x4000101C	长复位配置寄存器	0x00
R8_SLP_CLK_OFF2	0x4000101D	睡眠时钟控制寄存器 2	0x00
R16_SLP_WAKE_CFG	0x4000101E	睡眠配置寄存器	0x0100

R16_POWER_PLAN	0x40001020	睡眠电源管理寄存器	0x11CF
R16_AUX_POWER_ADJ	0x40001022	辅助电源调整控制寄存器	0x0XXX
R8_BAT_DET_CTRL	0x40001024	电池电压检测控制寄存器	0x00
R8_BAT_DET_CFG	0x40001025	电池电压检测配置寄存器	0x02
R8_BAT_STATUS	0x40001026	电池状态寄存器	0x00

睡眠时钟控制寄存器 0 (R8_SLP_CLK_OFF0)

位	名称	访问	描述	复位值
7	RB_SLP_KEYSCAN_WAKE	RWA	使能按键扫描事件唤醒系统: 1: 开启; 0: 关闭。	0
[6:5]	Reserved	RO	保留	0
4	RB_SLP_CLK_UART	RWA	串口时钟源: 1: 关闭; 0: 开启。	0
[3:2]	Reserved	RO	保留	0
1	RB_SLP_CLK_CMP	RWA	比较器时钟源: 1: 关闭; 0: 开启。	0
0	RB_SLP_CLK_TMR	RWA	定时器时钟源: 1: 关闭; 0: 开启。	0

睡眠时钟控制寄存器 1 (R8_SLP_CLK_OFF1)

位	名称	访问	描述	复位值
7	RB_SLP_CLK_BLE	RWA	BLE 控制器时钟源: 1: 关闭; 0: 开启。	0
[6:5]	Reserved	RO	保留	0
4	RB_SLP_CLK_USB	RWA	USBFS 控制器时钟源: 1: 关闭; 0: 开启。	0
3	RB_SLP_CLK_I2C	RWA	I2C 时钟源: 1: 关闭; 0: 开启。	0
2	RB_SLP_CLK_PWMx	RWA	PWMx 时钟源: 1: 关闭; 0: 开启。	0
1	RB_CLK_OFF_AESCCM	RWA	AES_CCM 时钟源: 1: 关闭; 0: 开启。	0
0	RB_SLP_CLK_SPI	RWA	SPI 时钟源: 1: 关闭; 0: 开启。	0

唤醒事件配置寄存器 (R8_SLP_WAKE_CTRL)

位	名称	访问	描述	复位值
7	RB_GPIO_WAKE_MODE	RWA	GPIO 边沿唤醒模式使能:	0

			1: RB_SLP_GPIO_EDGE_MODE=1, 边沿唤醒; RB_SLP_GPIO_EDGE_MODE=0, 上升沿唤醒; 0: 电平唤醒。	
6	RB_WAKE_EV_MODE	RWA	唤醒事件内部记忆模式使能: 1: 使能记忆, 支持短脉冲事件唤醒; 0: 不记忆, 事件需保持有效直到唤醒。	0
5	RB_SLP_BAT_WAKE	RWA	使能电池低压事件唤醒系统: 1: 使能; 0: 关闭。	1
4	RB_SLP_GPIO_WAKE	RWA	使能 GPIO 事件唤醒系统: 1: 使能; 0: 关闭。	0
3	RB_SLP_RTC_WAKE	RWA	使能 RTC 事件唤醒系统: 1: 使能; 0: 关闭。	0
2	RB_SLP_GPIO_EDGE_MODE	RWA	RB_GPIO_WAKE_MODE = 1 时, GPIO 事件唤醒事件选择: 1: 不分极性, 任意边沿都可唤醒; 0: 边沿唤醒。	0
1	RB_SLP_ENC_WAKE	RWA	使能编码器事件唤醒系统: 1: 使能; 0: 关闭。	0
0	RB_SLP_USB_WAKE	RWA	使能 USB 事件唤醒系统: 1: 使能; 0: 关闭。	0

外设睡眠电源控制寄存器 (R8_SLP_POWER_CTRL)

位	名称	访问	描述	复位值
[7:6]	RB_RAM_RET_LV	RWA	SRAM 睡眠时辅助电源低压使能: 00: 禁止; 01: 低功耗模式一; 10: 低功耗模式二; 11: 低功耗模式三。	0
5	Reserved	R0	保留。	0
4	RB_SLP_CLK_RAMX	RWA	主 SRAM (RAM12K) 的时钟控制: 1: 关闭; 0: 开启。	0
3	Reserved	R0	保留。	0
[2:0]	RB_WAKE_DLY_MOD	RWA	选择唤醒后的延时周期数 (Fsys) : 000: 3584; 001: 512; 010: 64; 011: 1; 100: 8191; 101: 7168; 110: 6144; 111: 4096。 注: bit[2] = 1 时, 为长延时; bit[2]	0

			= 0 时, 为短延时。	
--	--	--	--------------	--

长复位配置寄存器 (R8_LONG_RST_CFG)

位	名称	访问	描述	复位值
[7:3]	Reserved	R0	保留。	0
[2:1]	RB_LONG_TIM_SEL	RWA	长复位时长选择: 11: 32.768ms; 10: 25.000ms; 01: 20.000ms; 00: 15.000ms。	0
0	RB_LONG_RST_EN	RWA	长复位使能: 1: 开启长复位; 0: 关闭长复位。	0

睡眠时钟控制寄存器 2 (R8_SLP_CLK_OFF2)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
4	RB_CLK_OFF_HCLK	RWA	1: 关闭 FLASH 控制器的 HCLK 时钟; 0: 使能 FLASH 控制器的 HCLK 时钟。	0
[3:2]	Reserved	R0	保留。	0
1	RB_CLK_OFF_DEBUG	RWA	1: 关闭两线调试 32M 时钟; 0: 开启两线调试 32M 时钟。	0
0	RB_CLK_OFF_XROM	RWA	1: 关闭 FLASH 控制器的 64M 或 600M 时钟; 0: 使能 FLASH 控制器的 64M 或 600M 时钟。	0

睡眠配置寄存器 (R16_SLP_WAKE_CFG)

位	名称	访问	描述	复位值
[15:9]	Reserved	R0	保留。	0
8	RB_AC AUTO_ENABLE	RWA	1: 安全寄存器通道自动关闭; 0: 安全寄存器通道开启后不自动关闭。	1
7	Reserved	R0	保留。	0
[6:5]	RB_PRECLK_CNT_SEL	RWA	睡眠时间配置, 睡眠时长=周期数*Fsys: 11: 2048; 10: 1024; 01: 512; 00: 256。	0
4	RB_PRECLK_CNT_EN	RWA	1: 睡眠唤醒时需要等到一定时间才会释放 HCLK; 0: 睡眠唤醒后立即释放 HCLK。	0
[3:2]	Reserved	R0	保留。	0
0	RB_OSCCLK_RDY_KEEP	RWA	1: Halt 睡眠期间外设时钟保持开启; 0: Halt 睡眠期间外设时钟保持关闭。	0

睡眠电源管理寄存器 (R16_POWER_PLAN)

位	名称	访问	描述	复位值
15	RB_PWR_PLAN_EN	RWA/	睡眠电源规划控制使能:	0

		WZ	1: 开启规划; 0: 关闭或结束规划。 开启电源规划, 用于稍后进入睡眠或下电模式时执行, 执行后该位自动清 0。	
14	Reserved	RWA	保留。	0
13	Reserved	RWA	保留。	0
[12:9]	Reserved	RWA	保留。	1000b
8	RB_PWR_LD05V_EN	RWA	内部 LD05V 控制 (睡眠规划), 睡眠前配置: 1: 单一 V5 供电; 0: 单一 VDD33 供电。	1
7	RB_PWR_SYS_EN	RWA	系统电源控制 (睡眠规划): 1: 提供系统电源; 0: 关闭系统电源, 规划将进入睡眠模式或者下电模式。	1
6	RB_MAIN_ACT	RWA	主电源选择 1: ULL_LDO; 0: LDO。	1
[5:4]	Reserved	RO	保留。	0
3	RB_PWR_EXTEND	RWA	USBFS 和 RF 配置供电 (睡眠规划): 1: 双供电; 0: 不用辅助电源。	1
2	RB_PWR_CORE	RWA	内核和基本外设供电 (睡眠规划): 1: 双供电; 0: 不用辅助电源。	1
1	RB_PWR_RAM12K	RWA	RAM12K 的 SRAM 供电 (睡眠规划): 1: 双供电; 0: 不用辅助电源。	1
0	RB_PWR_XROM	RWA	FlashROM 供电 (睡眠规划): 1: 持续供电; 0: 睡眠时关闭电源。	1

辅助电源调整控制寄存器 (R16_AUX_POWER_ADJ)

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0
[11:8]	RB_CFG_IVREF	RO	保留。写入时必须保持原值不变。	XXXXb
[7:3]	Reserved	RO	保留。	00h
[2:0]	RB_ULPLDO_ADJ	RWA	超低功耗 LDO 的辅助电源输出电压调节值 (数值仅供参考, 不建议修改): 000: 0.756V; 001: 0.789V; 010: 0.823V; 011: 0.855V; 100: 0.889V; 101: 0.928V; 110: 0.966V; 111: 1.0V。	100b

电池电压检测控制寄存器 (R8_BAT_DET_CTRL)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
3	RB_BAT_LOW_IE	RWA	电池低电压中断使能: 1: 使能;	0

			0: 关闭。	
2	Reserved	RO	保留。	0
1	RB_BAT_MON_EN	RWA	低功耗的电池电压监控功能使能： 1: 使能低压监控, 增加电流约 1uA; 0: 关闭低功耗的电池低压监控。	0
0	RB_PWR_LDO_EN	RWA	保留。	0

注：如果电池电压达到电压检测阈值，且 RB_BAT_MON_EN 和 RB_BAT_LOW_IE 都使能，那么将产生 NMI 不可屏蔽中断。

电池电压检测配置寄存器 (R8_BAT_DET_CFG)

位	名称	访问	描述	复位值
[7:2]	Reserved	RO	保留。	0
[1:0]	RB_BAT_LOW_VTH	RWA	低电压检测阈值： 00: 1.8V; 01: 2.0V; 10: 2.2V; 11: 2.4V。	10b

电池状态寄存器 (R8_BAT_STATUS)

位	名称	访问	描述	复位值
[7:2]	Reserved	RO	保留。	0
1	RB_BAT_STAT_LOW	RO	电池低压检测或低压监控的结果，指示电池电压处于低电压状态： 1: 低于低电压阈值； 0: 不低于低电压阈值。	0
0	Reserved	RO	保留。	0

5.3 低功耗模式

在系统复位后，微控制器处于正常运行状态。当 MCU 不需要运行时，可以选择适当的低功耗模式来节省功耗。用户需要根据最低电源消耗、最快启动时间和可用唤醒事件等条件，选定一个合适的低功耗模式。

芯片提供以下 4 种主要的低功耗模式：

- 空闲模式 (Idle)：

所有外设保持正常供电，内核停止运行，时钟系统运转。检测到唤醒事件后，可以立即唤醒。

- 暂停模式 (Halt)：

在空闲模式的基础上，时钟系统停止。检测到唤醒事件后，首先时钟运转，然后唤醒内核运行。

- 睡眠模式 (Sleep)：

主 LDO 关闭，由超低功耗 ULP-LDO 维持 PMU、内核和基本外设供电，LSI 可以选择是否开启，RAM12K、USB 和 RF 配置可以选择是否维持供电。检测到唤醒事件后，首先主 LDO 开启，然后时钟运转，最后唤醒内核，程序继续运行，需要时可以重新设置到更高主频。

- 下电模式 (Shutdown)：

在睡眠模式的基础上，关闭了内核和基本外设以及 USB 和 RF 配置，LSI 可以选择是否开启，RAM12K 可以选择是否维持供电。检测到唤醒事件后，PMU 将执行 GRWSM 复位，软件可根据复位标志 RB_RESET_FLAG 和可选的 RAM 中的保持数据区分子 RPOR。

下表详细描述了几种低功耗模式的特征及唤醒途径：

表 5-2 低功耗模式

模式	特征	进入条件	唤醒事件	测试条件	功耗 ⁽¹⁾
空闲模式 Idle	外设均正常供电, 内核停止运行, 时钟系统运转, 但可以通过外设时钟控制位选择关闭各外设的时钟。	设置 SLEEPDEEP=0, 设置唤醒条件后执行 __WFI() 或 __WFE()	I/O 或 RTC 或 BAT 或 USB 或 I2C 或 SysTick 或 SPI 或 TMR 或 UART 或 KEYS defense	主频 16MHz, FLASH 片选关闭, 外设时钟关闭	1.7mA
暂停模式 Halt	外设均正常供电, 内核停止运行, 时钟系统 (PLL/HSE) 可配置开或关。	设置 SLEEPDEEP=1, 设置唤醒条件后执行 __WFI() 或 __WFE()	I/O 或 RTC 或 BAT 或 USB 或 KEYS defense	XT_FORCE_EN=1	1.3mA
				XT_FORCE_EN=0	420uA
睡眠模式 Sleep	主 LDO 关闭, 超低功耗 ULP-LDO 维持 PMU、内核和基本外设供电, LSI 可以选择是否开启, RAM12K、USB 和 RF 配置可以选择是否维持供电。	设置 SLEEPDEEP=1, 设置 POWER_PLAN, 设置唤醒条件后执行 __WFI() 或 __WFE()	I/O 或 RTC 或 BAT 或 KEYS defense。芯片唤醒后会继续运行	详见表 5-3	0.46uA ~ 1.2uA
下电模式 Shutdown	超低功耗 LDO 维持 PMU 供电, LSI 可以选择是否开启, RAM12K 可以选择是否维持供电, 用于保持数据。	设置 SLEEPDEEP=1, 设置 POWER_PLAN, 设置唤醒条件后执行 __WFI() 或 __WFE()	I/O 或 RTC 或 BAT 或 KEYS defense。芯片唤醒后会自动复位	详见表 5-3	0.3uA ~ 0.9uA

下表描述了几种低功耗模式的详细配置：

表 5-3 低功耗模式详细配置示例

规划配置	SYS_EN	RAM12K	LSI	CORE	EXTEND	功耗 ⁽¹⁾ (仅供参考)
维持供电的功能	系统电源	数据区 12KB	LSI RTC 唤醒	CPU 内核和 基本外设	USB 和 RF 配置	PMU 和 RTC 寄存器 常供电, 约 0.3uA
下电模式 常用配置	0	0	0	0	0	0.3uA
	0	1	0	0	0	0.6uA
	0	0	1	0	0	0.56uA
	0	1	1	0	0	0.9uA
睡眠模式 常用配置	0	0	0	1	0	0.46uA
	0	0	0	1	1	0.5uA
	0	1	0	1	0	0.8uA
	0	1	1	1	0	1.1uA
	0	0	1	1	0	0.8uA
	0	1	1	1	1	1.2uA

注：1. 电流参数均在室温下抽测得出, 注意温度变化导致电流变化;

第 6 章 系统时钟及 RTC

6.1 系统时钟简介

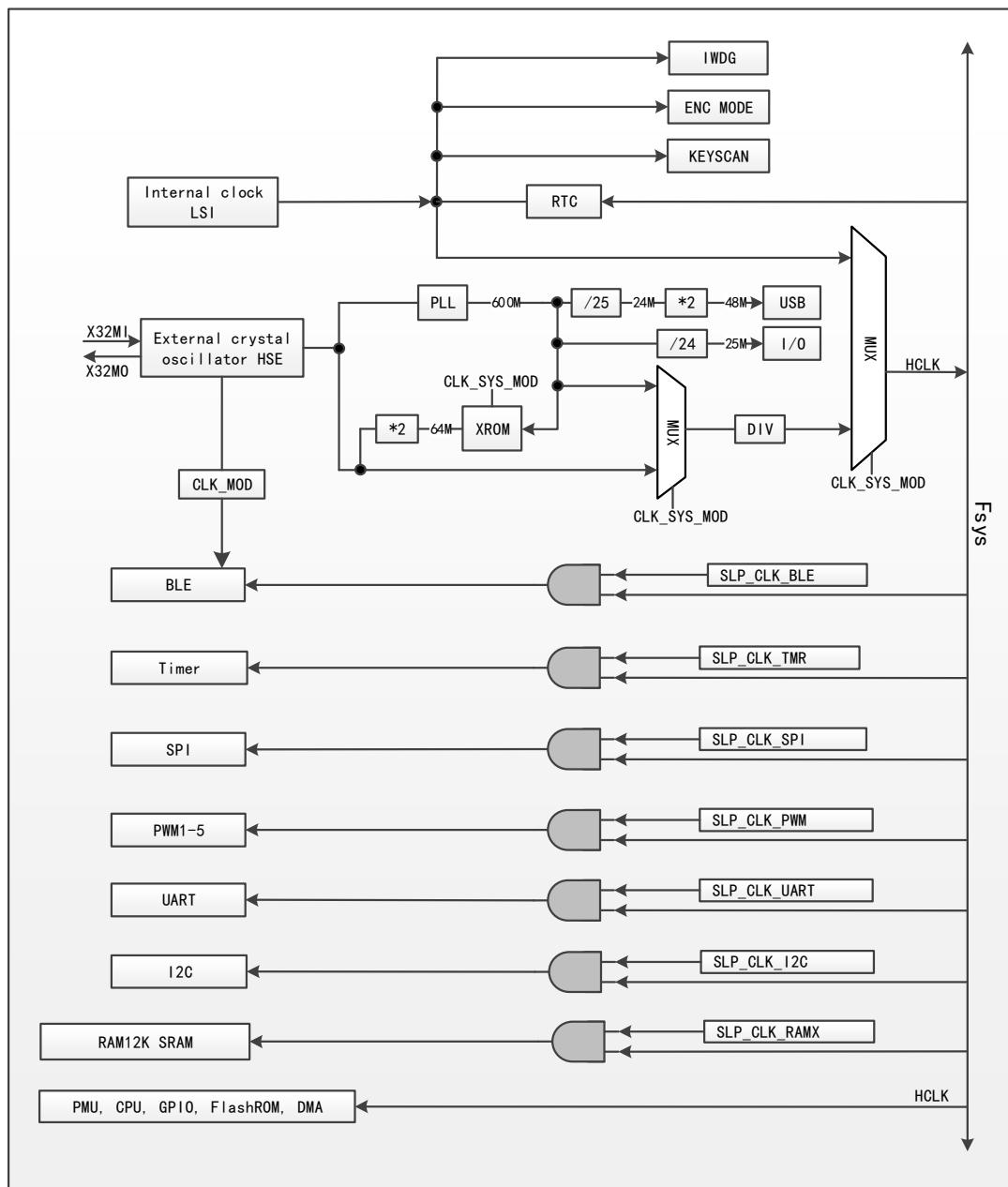
以下几种不同的时钟源可被选择来驱动系统时钟 HCLK (F_{sys})

- HSE 之分频。
- 内部 PLL (默认 600MHz) 分频。
- LSI 原始时钟。

任意一个时钟源都可以被独立地启动或关闭，由此可以优化系统功耗。

6.1.1 时钟结构

图 6-1 时钟树框图



上图是系统内部的时钟树结构，其中的 RTC 功能使用的是 LSI，所以使用这些功能必须打开低频时钟；USB 的数据传输依靠 PLL 分频后产生的时钟源；其他的外设驱动时钟以及数字控制逻辑由系统时钟或者再分频驱动。

6.2 RTC 简介

实时时钟 (RTC) 是一个独立的定时器，包含一组连续计数的计数器。在相应软件配置下，可提供简单日历功能。修改计数器的值可以重新设置当前的时间和日期。

RTC 寄存器与 PMU 一样常供电，在系统复位或从低功耗模式唤醒后，RTC 的设置和时间维持不变。

6.2.1 主要特性

- 可配置 2 种模式：
 - 定时模式：软件可选择固定周期时间（定时）产生中断通知。
 - 触发模式：匹配一个软件预设的目标闹钟时间，产生中断通知。
- 3 组 16 位计数器，提供了 LSI 原始周期、65536 分频周期、2764800000 分频周期的计数。

6.3 寄存器描述

表 6-1 时钟及振荡器控制相关寄存器列表

名称	访问地址	描述	复位值
R8_CLK_SYS_CFG	0x40001008	系统时钟配置寄存器	0x0005
R8_HFCK_PWR_CTRL	0x4000100A	高频时钟模块电源控制寄存器	0x14
R8_LSI_CONFIG	0x4000102F	内部低频振荡器 LSI 配置寄存器	0xX2
R8_XT32M_TUNE	0x4000104E	外部 32MHz 时钟谐振控制寄存器	0x53
R16_OSC_CAL_CNT	0x40001050	振荡器捕捉计数器计数值寄存器	0xXXXX
R8_OSC_CAL_OV_CNT	0x40001052	振荡器捕捉计数器溢出次数寄存器	0x00
R8_OSC_CAL_CTRL	0x40001053	振荡器捕捉控制寄存器	0x01
R8_PLL_CONFIG	0x4000104B	PLL 配置寄存器	0x0A
R8_RTC_FLAG_CTRL	0x40001030	RTC 标志和控制寄存器	0x30
R8_RTC_MODE_CTRL	0x40001031	RTC 模式配置寄存器	0xC2
R32_RTC_TRIG	0x40001034	RTC 触发数值寄存器	0x00000000
R16_RTC_CNT_LSI	0x40001038	RTC 基于 LSI 时钟周期的计数值寄存器	0xXXXX
R16_RTC_CNT_DIV1	0x4000103A	RTC 以 65536 个 LSI 时钟周期为单位的计数值寄存器	0xXXXX
R32_RTC_CNT_DIV2	0x4000103C	RTC 以 2831155200 个 LSI 时钟周期的计数值寄存器	0x0000XXXX

系统时钟配置寄存器 (R8_CLK_SYS_CFG)

位	名称	访问	描述	复位值
[7:6]	RB_CLK_SYS_MOD	RWA	HCLK 系统时钟源模式选择： 00/10: CK32M (默认 32MHz) 进行分频； 01: PLL (默认 600MHz) 进行分频； 11: LSI 作为 HCLK。	00b
5	Reserved	R0	保留。	0
[4:0]	RB_CLK_PLL_DIV	RWA	HCLK 输出时钟分频系数，最小值为 2， 0 代表最大值 32，写 1 将关闭 HCLK。	00101b

计算：

$F_{CK32m} = XT_32\text{ MHz}$ ；

$F_{CKLSI} = 24\sim42\text{ KHz}$ ；

$F_{PLL} = F_{CK32m} * 18.75 = 600\text{ MHz}$ ；

$F_{SYS} = RB_CLK_SYS_MOD == 3 ? F_{CKLSI} : (RB_CLK_SYS_MOD[0] ? F_{PLL} : F_{CK32m}) / RB_CLK_PLL_DIV$ ；

上电默认值 $F_{SYS} = F_{CK32m} / RB_CLK_PLL_DIV = 32\text{ MHz} / 5 = 6.4\text{ MHz}$ ；

F_{SYS} 范围：

位 RB_CLK_SYS_MOD[1:0]	HCLK 系统时钟源模式	Fsys 范围
11	LSI	24~42KHz (在 RAM 中运行)
00/10	CK32M 进行分频	1MHz~16MHz
01	PLL 进行分频	18.75MHz~100MHz

高频时钟模块电源控制寄存器 (R8_HFCK_PWR_CTRL)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
4	RB_CLK_PLL_PON	RWA	PLL 电源控制位： 1: 上电； 0: 掉电。	1
3	RB_CLK_XT32M_KEEP	RWA	用于暂停模式下控制时钟系统的停止： 1: Halt 模式下不自动停止 HSE 和 PLL； 0: Halt 模式下自动停止 HSE 和 PLL。	0
2	RB_CLK_XT32M_PON	RWA	外部 32MHz 振荡器 HSE 电源控制位： 1: 上电； 0: 掉电。	1
[1:0]	Reserved	RO	保留。	0

内部低频振荡器 LSI 配置寄存器 (R8_LSI_CONFIG)

位	名称	访问	描述	复位值
7	RB_LSI_CLK_PIN	RO	LSI 时钟引脚状态 (异步信号)。	X
[6:2]	Reserved	RO	保留。	0
1	RB_CLK_LSI_PON	RWA	内部低频 RC 振荡器 LSI 电源控制位： 1: 上电； 0: 掉电。	1
0	Reserved	RO	保留。	0

外部 32MHz 时钟谐振控制寄存器 (R8_XT32M_TUNE)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
[6:4]	RB_XT32M_C_LOAD	RWA	选择与外部 32MHz 晶体匹配的内置负载电容 (可能影响无线通讯)： 电容量 = RB_XT32M_C_LOAD * 2 + 6pF， 000b~111b 分别对应约 6pF~20pF。 根据所用晶体参数选择，常用为 111b。	101b
[3:2]	Reserved	RO	保留。	0
[1:0]	RB_XT32M_I_BIAS	RWA	外部 32MHz 振荡器偏置电流选择： 00: 75%额定电流； 01: 额定电流； 10: 125%额定电流； 11: 150%额定电流。	11b

振荡器捕捉计数器计数值寄存器 (R16_OSC_CAL_CNT)

位	名称	访问	描述	复位值
15	RB_OSC_CAL_IF	RW1	振荡器捕捉完成中断标志位，写 1 清零： 1: 有中断； 0: 无中断。	0
14	RB_OSC_CAL_OV_CLR	RW1	指示 RB_OSC_CAL_CNT 计满有溢出，写 1 将 R8_OSC_CAL_OV_CNT 清零。	0
[13:0]	RB_OSC_CAL_CNT	RO	对多个 LSI 周期基于系统主频的捕捉计数值。	XXXXh

振荡器捕捉计数器溢出次数寄存器 (R8_OSC_CAL_OV_CNT)

位	名称	访问	描述	复位值
[7:0]	RB_OSC_CAL_OV_CNT	R0	RB_OSC_CAL_CNT 计数溢出次数，向 RB_OSC_CAL_OV_CLR 写 1 清零此寄存器。	00h

振荡器捕捉控制寄存器 (R8_OSC_CAL_CTRL)

位	名称	访问	描述	复位值
7	RB_CNT_CLR	RWA	软件复位 RB_OSC_CNT_TOTAL: 1: 复位； 0: 不做操作。	0
6	RB_OSC_CNT_END	RWA	振荡器捕捉终点选择： 1: 追加 2 个周期； 0: 不追加。	0
5	RB_OSC_CNT_EN	RWA	振荡器捕捉计数器使能位： 1: 使能计数； 0: 禁止计数。	0
4	RB_OSC_CAL_IE	RWA	振荡器捕捉完成中断使能位： 1: 使能中断； 0: 禁止中断。	0
3	RB_OSC_CNT_HALT	R0	振荡器捕捉计数器计数状态位： 1: 正在暂停计数； 0: 正在计数中。	0
[2:0]	RB_OSC_CNT_TOTAL	RWA	振荡器捕捉总周期数选择： 000: 1 个； 001: 2 个； 010: 4 个； 011: 32 个； 100: 64 个； 101: 128 个； 110: 1024 个； 111: 2047 个。	001b

PLL 配置寄存器 (R8_PLL_CONFIG)

位	名称	访问	描述	复位值
[7:6]	Reserved	R0	保留。	0
[5:0]	RB_PLL_CFG_DAT	RWA	PLL 配置参数。	0Ah

RTC 标志和控制寄存器 (R8_RTC_FLAG_CTRL)

位	名称	访问	描述	复位值
7	RB_RTC_TRIG_FLAG	R0	RTC 触发模式激活标志。	0
6	RB_RTC_TMR_FLAG	R0	RTC 定时模式激活标志。	0
5	RB_RTC_TRIG_CLR	RW	禁用触发模式时，此位固定为 1。 使能触发模式时，写 1 清零触发模式激活标志 RB_RTC_TRIG_FLAG 并自动清 0。	1
4	RB_RTC_TMR_CLR	RW	禁用定时模式时，此位固定为 1。 使能定时模式时，写 1 清零定时模式激活标志 RB_RTC_TMR_FLAG 并自动清 0。	1
[3:0]	Reserved	R0	保留。	0h

RTC 模式配置寄存器 (R8_RTC_MODE_CTRL)

位	名称	访问	描述	复位值

7	RB_RTC_LOAD_HI	RWA	写1 将加载 RTC 计数器高字，加载后自动清0。将 R32_RTC_TRIG (实际仅低14位) 加载到 R32_RTC_CNT_DIV2。	1
6	RB_RTC_LOAD_LO	RWA	写1 将加载 RTC 计数器低字，加载后自动清0。将 R32_RTC_TRIG 高16位加载到 R16_RTC_CNT_DIV1；将 R32_RTC_TRIG 低16位加载到 R16_RTC_CNT_LSI。	1
5	RB_RTC_TRIG_EN	RWA	RTC 触发模式使能位： 1: 使能；0: 禁用。	0
4	RB_RTC_TMR_EN	RWA	RTC 定时模式使能位： 1: 使能；0: 禁用。	0
3	RB_RTC_IGNORE_B0	RWA	触发模式下忽略比较匹配值的最低位： 1: 忽略最低位；0: 比较最低位。	0
[2:0]	RB_RTC_TMR_MODE	RWA	RTC 定时模式固定周期(定时)选择， 单位为 LSI 的一个周期： 000: 4096; 001: 8192; 010: 16384; 011: 32768; 100: 65536; 101: 131072; 110: 262144; 111: 524288。	010b

RTC 触发数值寄存器 (R32_RTC_TRIG)

位	名称	访问	描述	复位值
[31:0]	R32_RTC_TRIG	RWA	RTC 触发模式下的预设匹配数值，其高16位与 R16_RTC_CNT_DIV1 匹配，低16位和 R16_RTC_CNT_LSI 进行匹配。 与 RB_RTC_LOAD_LO 及 RB_RTC_LOAD_HI 配合，用于更新 RTC 计数器当前值。	00000000h

注：预设匹配数值不是直接写入目标时间，涉及简单计算，请参考后面的说明。

RTC 基于 LSI 时钟周期的计数值寄存器 (R16_RTC_CNT_LSI)

位	名称	访问	描述	复位值
[15:0]	R16_RTC_CNT_LSI	R0	RTC 基于 LSI 时钟周期的计数值寄存器。	XXXXh

RTC 以 65536 个 LSI 时钟周期为单位的计数值寄存器 (R16_RTC_CNT_DIV1)

位	名称	访问	描述	复位值
[15:0]	R16_RTC_CNT_DIV1	R0	RTC 以 65536 个 LSI 时钟周期为单位的当前计数值。	XXXXh

RTC 以 2831155200 个 LSI 时钟周期的计数值寄存器 (R32_RTC_CNT_DIV2)

位	名称	访问	描述	复位值
[31:14]	Reserved	R0	保留。	0
[13:0]	R32_RTC_CNT_DIV2	R0	RTC 以 2831155200 个 LSI 时钟周期为单位的当前计数值。	XXXXh

6.4 功能描述及配置

6.4.1 RTC 计数器初始化

(1)、设置 R32_RTC_TRIG 寄存器数值，将 RB_RTC_LOAD_HI 置位，即可将 R32_RTC_TRIG 寄存器的数值

加载到 R32_RTC_CNT_DIV2 寄存器中；

(2)、设置 R32_RTC_TRIG 寄存器数值，将 RB_RTC_LOAD_L0 置位，即可将 R32_RTC_TRIG 寄存器的高低各 16 位数值分别加载到 R16_RTC_CNT_DIV1 寄存器和 R16_RTC_CNT_LSI 寄存器。

6.4.2 RTC 定时功能

(1)、配置 R8_RTC_MODE_CTRL 寄存器，设置 RB_RTC_TMR_MODE 选择合适的定时周期，置 RB_RTC_TMR_EN 为 1，打开 RTC 定时功能；

(2)、达到定时周期后，会产生 RTC 定时激活标志 RB_RTC_TMR_FLAG 及中断，查询 R8_RTC_FLAG_CTRL 寄存器，置位 RB_RTC_TMR_CLR 可清零标志。

6.4.3 RTC 触发功能

(1)、在 R32_RTC_TRIG 寄存器中设置好目标匹配数值，计算和操作步骤：

以当前时间 R32_RTC_CNT_LSI (高 16 位 R16_RTC_CNT_DIV1 和低 16 位 R16_RTC_CNT_LSI) 加上间隔时间 DelayTime (单位 S)，计算出目标时间数值， $T32 = R32_RTC_CNT_LSI + DelayTime * FckLSI$ ，将 T32 写入 R32_RTC_TRIG 寄存器中，完成匹配数值设定。例：假设 FckLSI 为 42KHz，触发时间设置为 1S 后，计算目标时间数值， $T32 = R32_RTC_CNT_LSI + 1 * 42000$ ；

(2)、配置 R8_RTC_MODE_CTRL 寄存器，置 RB_RTC_TRIG_EN 为 1，打开 RTC 触发功能；

(3)、当 RTC 当前计数值 R16_RTC_CNT_DIV1 和 R16_RTC_CNT_LSI 分别与 R32_RTC_TRIG 预设的高和低 16 位匹配时，产生 RTC 触发激活标志 RB_RTC_TRIG_FLAG 及中断，置位 RB_RTC_TRIG_CLR 可清零标志。具体可参考评估板例子程序。

第 7 章 通用 I/O 和复用功能

7.1 GPIO 简介

芯片提供了 1 组 GPIO 端口 PA，共 12 个通用输入输出引脚，均具有中断和唤醒功能，部分引脚具有复用及映射功能。

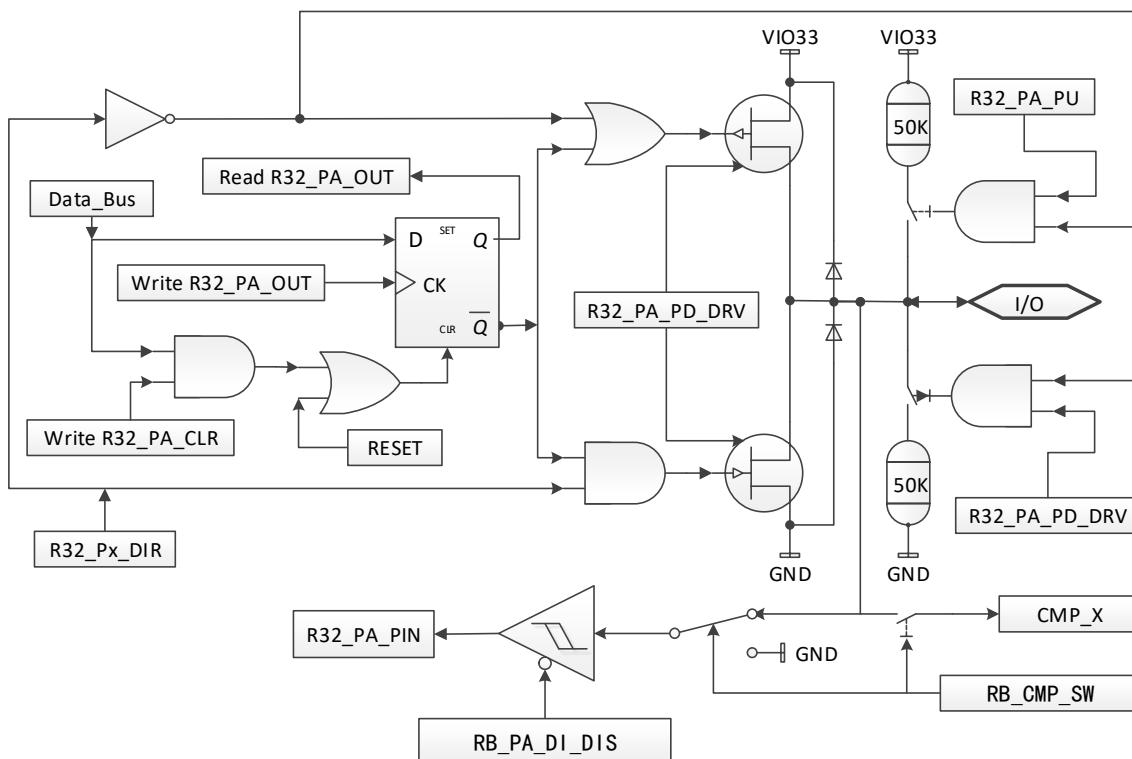
每个 GPIO 端口有一个 32 位方向配置寄存器 R32_PA_DIR，一个 32 位引脚输入寄存器 R32_PA_PIN，一个 32 位数据输出寄存器 R32_PA_OUT，一个 32 位数据复位寄存器 R32_PA_CLR，一个 32 位上拉电阻配置寄存器 R32_PA_PU，一个 32 位下拉电阻/驱动能力配置寄存器 R32_PA_PD_DRV。

PA 端口中 PA[0]~PA[11]位有效，对应芯片上 12 个 GPIO 引脚。

每个 I/O 端口位可以独立编程，但是 I/O 端口寄存器必须按 8 位、16 位或者 32 位字访问。如果引脚的复用功能没有开启，则默认作为通用 I/O 口使用。

下图是 GPIO 内部结构框图：

图 7-1 I/O 内部结构框图



7.2 外部中断/唤醒

芯片的所有 I/O 管脚具有中断功能，并可实现睡眠唤醒。

为了使用外部中断，端口位必须配置成输入模式。并提供 4 种触发模式：高电平、低电平、上升沿、下降沿。

唤醒功能需要打开端口位的中断使能 R16_PA_INT_EN，并开启寄存器 R8_SLP_WAKE_CTRL 中的 GPIO 唤醒控制位 RB_SLP_GPIO_WAKE。

7.3 GPIO 的复用与重映射

7.3.1 复用功能

部分 I/O 引脚具有复用功能，上电后默认所有 I/O 引脚均为通用 I/O 功能，启用各功能模块后，相应的原 GPIO 引脚被配置成各自功能模块对应的功能引脚。

如果一个管脚复用多个功能，并且多个功能都已开启，那么复用功能的优先级顺序请参考 1.2 节

引脚描述的“复用功能”列表中功能顺序。

下表列出了部分用于外设模块的功能引脚的 GPIO 配置。

表 7-1 定时器 x

TMR 引脚	功能配置	GPIO 配置
TMR	输入捕捉通道 x	输入 (浮空输入/上拉输入/下拉输入)
	输出 PWM 通道 x	推挽输出

表 7-2 UART

UART 引脚	功能配置	GPIO 配置
TXD	串口发送 x	推挽输出
RXD	串口接收 x	上拉输入 (推荐) 或浮空输入

表 7-3 SPI

SPI 引脚	功能配置	GPIO 配置
SCK	主模式时钟输出	推挽输出
	从模式时钟输入	输入 (浮空输入/上拉输入/下拉输入)
MOSI	全双工模式-主模式	推挽输出
	全双工模式-从模式	输入 (浮空输入/上拉输入/下拉输入)
	半双工模式-主模式	未用到, 可做通用 I/O
	半双工模式-从模式	未用到, 可做通用 I/O
MISO	全双工模式-主模式	输入 (浮空输入/上拉输入/下拉输入)
	全双工模式-从模式	输入 (推荐上拉, 片选后自动切为推挽输出) 或推挽输出 (禁止用于总线连接方式)
	半双工模式-主模式	输入或推挽输出, 手工切换
	半双工模式-从模式	输入 (推荐上拉, 片选后自动切为推挽输出)
SCS	主模式片选输出	推挽输出 (可换用其它引脚)
	从模式片选输入	上拉输入 (推荐) 或浮空输入

表 7-4 I2C

I2C 引脚	功能配置	GPIO 配置
SCL	串行时钟输出-主模式	推挽输出 (此模式不支持多主机)
	串行时钟输出/输入-多主模式	输入 (推荐上拉, 需要时自动开漏输出)
	串行时钟输入-从模式	上拉输入 (推荐) 或浮空输入
SDA	串行数据输入输出	输入 (推荐上拉, 需要时自动开漏输出)

表 7-5 USB

USB 信号引脚	功能配置	GPIO 配置
UDP	连接到内部全速 USB 收发器	浮空输入
UDM	连接到内部全速 USB 收发器	浮空输入

表 7-6 CMP

CMP 信号引脚	功能配置	GPIO 配置
CMP_P0, CMP_P1	CMP 模拟输入+端	仅模拟输入
CMP_N	CMP 模拟输入-端	仅模拟输入

7.3.2 功能引脚重映射

为了使外设功能的同时利用率达到最优,可以通过设置 R16_PIN_ALTERNATE 功能引脚重映射寄存器把一些功能引脚重新映射到其他引脚上。

表 7-7 复用功能重映射引脚

外设功能引脚	默认所在的 GPIO 引脚	重映射到的 GPIO 引脚
SCS/SCK/MOSI/MISO	PA4/PA5/PA7/PA6	PA2/PA3/-/-
RXD	PA2	PA3/PA0/PA1/PA4/PA9/PA10/PA11
TXD	PA3	PA2/PA1/PA0/PA7/PA8/PA11/PA10
TMR/PWM0	PA7	PA2/PA4/PA9
CAP_IN1	PA7	PA2/PA4/PA9
CAP_IN2	PA2	PA7/PA9/PA4
PWM1	PA7	-
PWM2	PA2	-
PWM3	PA3	-
PWM4	PA4	-
PWM5	PA8	-
SCL	PA8	PA0/PA3/PA5
SDA	PA9	PA1/PA2/PA6

7.4 寄存器描述

表 7-8 GPIO 相关寄存器列表

名称	访问地址	描述	复位值
R16_PIN_ALTERNATE	0x40001018	功能引脚重映射寄存器	0x4000
R16_PIN_ALTERNATE_H	0x4000101A	功能引脚高位重映射寄存器	0x0000
R16_PA_INT_EN	0x40001090	PA 端口中断使能寄存器	0x0000
R16_PA_INT_MODE	0x40001094	PA 端口中断模式配置寄存器	0x0000
R16_PA_INT_EDGE_TYPE	0x40001096	PA 端口中断边沿类型配置寄存器	0x0000
R16_PA_INT_IF	0x4000109C	PA 端口中断标志寄存器	0x0000
R32_PA_DIR	0x400010A0	PA 端口方向配置寄存器	0x00000000
R32_PA_PIN	0x400010A4	PA 端口引脚输入寄存器	0x00000XXX
R32_PA_OUT	0x400010A8	PA 端口数据输出寄存器	0x00000000
R32_PA_CLR	0x400010AC	PA 端口数据复位寄存器	0x00000000
R32_PA_PU	0x400010B0	PA 端口上拉电阻配置寄存器	0x00000000
R32_PA_PD_DRV	0x400010B4	PA 端口下拉/驱动配置寄存器	0x00000000
R32_PA_SET	0x400010B8	PA 端口输出置位寄存器	0x00000000

功能引脚重映射寄存器 (R16_PIN_ALTERNATE)

位	名称	访问	描述	复位值
15	Reserved	R0	保留。	0
14	RB_PIN_DEBUG_EN	RW	调试接口控制位: 1: 开启调试接口; 0: 禁用调试接口。	1
13	RB_PIN_USB_EN	RW	全速 USB 引脚使能: 1: 使能全速 USB 通讯引脚; 0: 禁用全速 USB 通讯引脚。	0
12	RB_UDP_PU_EN	RW	全速 USB 的 UDP 引脚内部上拉电阻使能: 1: 强制使能上拉 (RB_UC_DEV_PU_EN 在	0

			睡眠或下电模式下不起作用，由此代替）； 0：由 RB_UC_DEV_PU_EN 控制是否上拉。	
[11:0]	RB_PA_DI_DIS	RW	PA0-PA11 通道引脚数字输入使能： 1：关闭数字输入，可节约功耗； 0：打开数字输入。	0

功能引脚高位重映射寄存器 (R16_PIN_ALTERNATE_H)

位	名称	访问	描述	复位值
[15:13]	Reserved	R0	保留。	0
12	RB_25M_EN	RW	25MHz 时钟输出使能： 1：PA4 输出 25MHz 时钟； 0：不输出。	0
11	RB_SPI_CLK	R0	SPI 的 CLK 功能引脚映射选择位： 1：重映射 (SCK/PA3)； 0：默认映射 (SCK/PA5)。	0
[10:9]	RB_I2C_PIN	RW	I2C 功能引脚映射选择位： 00：默认映射 (SCL/PA8, SDA/PA9)； 01：重映射 (SCL/PA0, SDA/PA1)； 10：重映射 (SCL/PA3, SDA/PA2)； 11：重映射 (SCL/PA5, SDA/PA6)。	0
8	RB_SPI_CS	RW	SPI 的 CS 功能引脚映射选择位： 1：重映射 (SCS/PA2)； 0：默认映射 (SCS/PA4)。	0
[7:6]	RB_TMR_PIN	RW	定时器功能引脚映射选择位： 00：默认映射 (PWMO/PA7, CAP_IN1/PA7, CAP_IN2/PA2)； 01：重映射 (PWMO/PA2, CAP_IN1/PA2, CAP_IN2/PA7)； 10：重映射 (PWMO/PA4, CAP_IN1/PA4, CAP_IN2/PA9)； 11：重映射 (PWMO/PA9, CAP_IN1/PA9, CAP_IN2/PA4)。	0
[5:3]	RB_UART_TXD	RW	UART 的 TXD 功能引脚映射选择位： 000：默认映射 (TXD/PA3)； 001：重映射 (TXD/PA2)； 010：重映射 (TXD/PA1)； 011：重映射 (TXD/PA0)； 100：重映射 (TXD/PA7)； 101：重映射 (TXD/PA8)； 110：重映射 (TXD/PA11)； 111：重映射 (TXD/PA10)。	0
[2:0]	RB_UART_RXD	RW	UART 的 RXD 功能引脚映射选择位： 000：默认映射 (RXD/PA2)； 001：重映射 (RXD/PA3)； 010：重映射 (RXD/PA0)； 011：重映射 (RXD/PA1)； 100：重映射 (RXD/PA4)；	0

		101: 重映射 (RXD/PA9); 110: 重映射 (RXD/PA10); 111: 重映射 (RXD/PA11)。	
--	--	---	--

注: 1. 如果引脚用于模拟功能, 建议将该引脚的数字输入功能关闭, 即设置数字输入禁用, 从而可以降低功耗, 并有利于减少干扰。

2. 该寄存器仅在上电复位和 *Shutdown* 睡眠时清 0。

PA 端口中断使能寄存器 (R16_PA_INT_EN)

位	名称	访问	描述	复位值
[15:0]	R16_PA_INT_EN	RW	PA 引脚中断使能位: 1: 使能相应中断; 0: 禁止相应中断。	0000h

PA 端口中断模式配置寄存器 (R16_PA_INT_MODE)

位	名称	访问	描述	复位值
[15:0]	R16_PA_INT_MODE	RW	PA 引脚中断模式选择位: 1: 边沿触发; 0: 电平触发。	0000h

PA 端口中断边沿类型配置寄存器 (R16_PA_INT_EDGE_TYPE)

位	名称	访问	描述	复位值
[15:0]	R16_PA_INT_EDGE_TYPE	RW	PA 引脚中断边沿类型配置: 1: 不遵照极性; 0: 遵照极性。	0000h

PA 端口中断标志寄存器 (R16_PA_INT_IF)

位	名称	访问	描述	复位值
[15:0]	R16_PA_INT_IF	RW1	PA 引脚中断标志位, 写 1 清零: 1: 有中断; 0: 无中断。	0000h

PA 端口方向配置寄存器 (R32_PA_DIR)

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
[11:8]	R8_PA_DIR_1	RW	PA 引脚当前输入输出方向配置:	0
[7:0]	R8_PA_DIR_0	RW	1: 引脚为输出模式; 0: 引脚为输入模式。	0

PA 端口引脚输入寄存器 (R32_PA_PIN)

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
[11:8]	R8_PA_PIN_1	R0	PA 引脚当前电平状态 (仅在 R32_PA_DIR 对应位为 0 时, 该位值有效):	Xh
[7:0]	R8_PA_PIN_0	R0	1: 引脚输入高电平; 0: 引脚输入低电平。	XXh

PA 端口数据输出寄存器 (R32_PA_OUT)

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0000h
[11:8]	R8_PA_OUT_1	RW	方向寄存器 R32_PA_DIR 对应位为 1 时： 控制 PA 引脚输出电平状态： 1: 输出高电平； 0: 输出低电平。	0
[7:0]	R8_PA_OUT_0	RW	方向寄存器 R32_PA_DIR 对应位为 0 时： 控制 PA 引脚中断极性选择： 1: 高电平/上升沿； 0: 低电平/下降沿。	0

PA 端口数据复位寄存器 (R32_PA_CLR)

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
[11:8]	R8_PA_CLR_1	WZ	PA 数据寄存器复位控制： 1: R32_PA_OUT 对应位数据清 0； 0: 无影响。	0
[7:0]	R8_PA_CLR_0	WZ		0

PA 端口上拉电阻配置寄存器 (R32_PA_PU)

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0000h
[11:8]	R8_PA_PU_1	RW	PA 引脚上拉电阻使能控制： 1: 启用上拉电阻； 0: 关闭上拉电阻。	00h
[7:0]	R8_PA_PU_0	RW		00h

PA 端口下拉/驱动配置寄存器 (R32_PA_PD_DRV)

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
[11:8]	R8_PA_PD_DRV_1	RW	方向寄存器 R32_PA_DIR 对应位为 0 时： PA 引脚下拉电阻使能控制： 1: 启用下拉电阻； 0: 关闭下拉电阻。	0
[7:0]	R8_PA_PD_DRV_0	RW	方向寄存器 R32_PA_DIR 对应位为 1 时： PA 引脚电流驱动能力选择： 1: 20mA 级别； 0: 5mA 级别。	0

PA 端口输出置位寄存器 (R32_PA_SET)

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
[11:8]	R8_PA_SET_1	WZ	置位寄存器 R32_PA_SET 对应位为 0 时， PA 引脚输出保持；为 1 时，PA 引脚输出	0
[7:0]	R8_PA_SET_0	WZ	高电平。	0

7.5 GPIO 引脚模式配置

每个 GPIO 都可以配置成 5 种模式，具体见下表：

表 7-9 端口配置表

模式	R32_PA_DIR	R32_PA_PU	RB_PA_DI_DIS	R32_PA_PD_DRV
浮空输入/高阻输入/模拟输入	0	0	0	0
仅模拟输入 (减少 I/O 漏电)	0	0	1	0
带上拉电阻的输入	0	1	0	0
带下拉电阻的输入	0	0	0	1
推挽输出, 驱动能力 5mA 级别	1	X	X	0
推挽输出, 驱动能力 20mA 级别	1	X	X	1

第 8 章 通用定时器 TMR

8.1 TMR 简介

芯片提供了 1 个 26 位定时器，最长定时时间为 2^{26} 个时钟周期。它适用于多种场合，包括测量输入信号脉冲长度（输入捕捉）或者产生输出波形（PWM）；还支持编码器接口模式，提供了 2 路通道，以捕捉旋转编码器信号进行编码；支持 DMA 功能。

8.1.1 主要特性

- 1 个 26 位定时器，定时时间最长为 2^{26} 个时钟周期。
- 支持定时器中断，以及支持 DMA 及中断。
- 支持捕捉功能，提供 1 路捕捉/采样，测量输入脉冲长度或周期。
- 支持编码器模式（ENC MODE），2 路通道从 I/O 口捕捉 2 路旋转编码器信号进行编码，可在一路信号边沿处比较另一路信号的高低电平，判断编码方向，其时钟来源为 LSI。
- 捕捉功能可设置为电平变化捕捉和高或低电平保持时间捕捉功能。
- 支持 26 位 PWM 功能，可动态调整 PWM 占空比设置。

8.2 寄存器描述

表 8-1 TMR 相关寄存器列表

名称	访问地址	描述	复位值
R8_TMR_CTRL_MOD	0x40002400	模式设置寄存器	0x02
R8_TMR_CTRL_DMA	0x40002401	DMA 控制寄存器	0x00
R8_TMR_INTER_EN	0x40002402	中断使能寄存器	0x00
R8_TMR_INT_FLAG	0x40002406	中断标志寄存器	0x00
R8_TMR_FIFO_COUNT	0x40002407	FIFO 计数寄存器	0x0X
R32_TMR_COUNT	0x40002408	当前计数值寄存器	0x0XXXXXXX
R32_TMR_CNT_END	0x4000240C	计数终值寄存器	0x0XXXXXXX
R32_TMR_FIFO	0x40002410	FIFO 寄存器	0x0XXXXXXX
R16_TMR_DMA_NOW	0x40002414	DMA 当前缓冲区地址	0x0000XXXX
R16_TMR_DMA_BEG	0x40002418	DMA 起始缓冲区地址	0x0000XXXX
R16_TMR_DMA_END	0x4000241C	DMA 结束缓冲区地址	0x0000XXXX
R8_ENC_REG_CTRL	0x40002420	编码器模式控制寄存器	0x00
R8_ENC_INTER_EN	0x40002421	编码器模式中断使能寄存器	0x00
R8_ENC_INT_FLAG	0x40002422	编码器模式中断标志寄存器	0x00
R32_ENC_REG_CEND	0x40002424	编码器模式终值配置寄存器	0x00000000
R32_ENC_REG_CCNT	0x40002428	编码器模式当前值配置寄存器	0x00000000

模式设置寄存器 (R8_TMR_CTRL_MOD)

位	名称	访问	描述	复位值
[7:6]	RB_TMR_CAP_EDGE	RW	<p>捕捉模式下，选择捕捉触发方式：</p> <p>00：不触发； 01：捕捉任何边沿变化之间的时间； 10：捕捉下降沿到下降沿之间时间； 11：捕捉上升沿到上升沿之间时间。</p> <p>计数模式下，选择计数的边沿：</p> <p>00：不采样计数； 01：采样到任意边沿计数； 10：采样到下降沿计数；</p>	00b

			11：采样到上升沿计数。	
[7:6]	RB_TMR_PWM_REPEAT	RW	PWM 模式下, 选择数据重复方式: 00: 重复 1 次; 01: 重复 4 次; 10: 重复 8 次; 11: 重复 16 次。	00b
5	Reserved	RO	保留。	0
4	RB_TMR_CAP_COUNT	RW	RB_TMR_MODE_IN=1 输入模式的子模式: 1: 计数模式; 0: 捕捉模式。	0
4	RB_TMR_OUT_POLAR	RW	PWM 模式下, 输出极性设置位: 1: 默认高电平, 低电平有效; 0: 默认低电平, 高电平有效。	0
3	RB_TMR_OUT_EN	RW	定时器输出使能位: 1: 输出使能; 0: 输出禁止。	0
2	RB_TMR_COUNT_EN	RW	定时器计数使能位: 1: 使能计数; 0: 停止计数。	0
1	RB_TMR_ALL_CLEAR	RW	定时器的 FIFO/计数器/中断标志清零: 1: 强制清空和清零; 0: 不清。	1
0	RB_TMR_MODE_IN	RW	定时器模式设置位: 1: 输入模式 (捕捉模式或计数模式); 0: 定时模式或 PWM 模式。	0

中断使能寄存器 (R8_TMR_INTER_EN)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	000b
4	RB_TMR_IE_FIFO_OV	RW	FIFO 溢出 (捕捉模式 FIFO 满或 PWM 模式 FIFO 空) 中断使能位: 1: 使能中断; 0: 禁止中断。	0
3	RB_TMR_IE_DMA_END	RW	DMA 结束中断使能位: 1: 使能中断; 0: 禁止中断。	0
2	RB_TMR_IE_FIFO_HF	RW	FIFO 使用过半 (捕捉模式 $FIFO \geq 4$ 或 PWM 模式 $FIFO < 4$) 中断使能位: 1: 使能中断; 0: 禁止中断。	0
1	RB_TMR_IE_DATA_ACT	RW	数据激活 (捕捉模式指每次捕捉到新数据, PWM 模式指数值触发导致有效电平结束) 中断使能位: 1: 使能中断; 0: 禁止中断。	0
0	RB_TMR_IE_CYC_END	RW	周期结束 (捕捉模式指超时, PWM 模式和定时模式指周期结束) 中断使能位: 1: 使能中断; 0: 禁止中断。	0

中断标志寄存器 (R8_TMR_INT_FLAG)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	000b
4	RB_TMR_IF_FIFO_OV	RW1	FIFO 溢出 (捕捉模式 FIFO 满或 PWM 模式 FIFO 空) 标志位, 写 1 清零: 1: 已溢出; 0: 未溢出。	0
3	RB_TMR_IF_DMA_END	RW1	DMA 完成标志位, 写 1 清零: 1: 已完成; 0: 未完成。	0
2	RB_TMR_IF_FIFO_HF	RW1	FIFO 使用过半 (捕捉模式 FIFO>=4 或 PWM 模式 FIFO<4) 标志位, 写 1 清零: 1: FIFO 使用已过半; 0: FIFO 使用未过半。	0
1	RB_TMR_IF_DATA_ACT	RW1	数据激活 (捕捉模式指每次捕捉到新数据, PWM 模式指数值触发导致有效电平结束) 标志位, 写 1 清零: 1: 产生/用了数据; 0: 未产生/未用。	0
0	RB_TMR_IF_CYC_END	RW1	周期结束 (捕捉模式指超时, PWM 模式和定时模式指周期结束, 计数模式指计数溢出) 标志位, 写 1 清零: 1: 超时/周期结束; 0: 未超时/未结束。	0

FIFO 计数寄存器 (R8_TMR_FIFO_COUNT)

位	名称	访问	描述	复位值
[7:4]	Reserved	R0	保留。	0h
[3:0]	R8_TMR_FIFO_COUNT	R0	FIFO 内数据计数, 最大值 8。	0Xh

当前计数值寄存器 (R32_TMR_COUNT)

位	名称	访问	描述	复位值
[31:26]	Reserved	R0	保留。	0Xh
[25:0]	R32_TMR_COUNT	R0	计数器当前计数值。	XXXXXXXXh

计数终值设置寄存器 (R32_TMR_CNT_END)

位	名称	访问	描述	复位值
[31:0]	R32_TMR_CNT_END	RW	定时器模式下, 一个定时周期时钟数; PWM 模式下, PWM 单周期总时钟数; 捕捉模式下, 捕捉超时时钟数。 仅低 26 位有效, 最大值 67108863。 计数模式下, 计数值终值-2 (溢出)。 注: 对此寄存器的写操作, 将自动清零 R32_TMR_COUNT 寄存器中的值。	XXXXXXXXh

FIFO 寄存器 (R32_TMR_FIFO)

位	名称	访问	描述	复位值

[31:0]	R32_TMR_FIFO	RO/ WO	FIFO 数据寄存器, 仅低 26 位有效。	0XXXXXXh
--------	--------------	-----------	------------------------	----------

DMA 控制寄存器 (R8_TMR_CTRL_DMA)

位	名称	访问	描述	复位值
[7:3]	Reserved	RO	保留。	00000b
2	RB_TMR_DMA_LOOP	RW	DMA 地址循环功能使能位: 1: 使能地址循环; 0: 禁止地址循环。 如果使能 DMA 地址循环, 当 DMA 地址增加到设置的末尾地址时, 自动循环指向设置的首地址。	0
1	Reserved	RO	保留。	0
0	RB_TMR_DMA_ENABLE	RW	DMA 功能使能位: 1: 使能 DMA; 0: 禁止 DMA。	0

DMA 当前缓冲区地址 (R16_TMR_DMA_NOW)

位	名称	访问	描述	复位值
15	Reserved	RO	保留。	0
[14:2]	R16_TMR_DMA_NOW	RO	DMA 数据缓冲区当前地址。 可用于计算已转换次数, 计算方法: COUNT=(TMR_DMA_NOW-TMR_DMA_BEG)/4。	XXXXh
[1:0]	Reserved	RO	保留。	00b

DMA 起始缓冲区地址 (R16_TMR_DMA_BEG)

位	名称	访问	描述	复位值
15	Reserved	RO	保留。	0
[14:2]	R16_TMR_DMA_BEG	RW	DMA 数据缓冲区起始地址, 地址必须 4 字节对齐。	XXXXh
[1:0]	Reserved	RO	保留。	00b

DMA 结束缓冲区地址 (R16_TMR_DMA_END)

位	名称	访问	描述	复位值
15	Reserved	RO	保留。	0
[14:2]	R16_TMR_DMA_END	RW	DMA 数据缓冲区结束地址 (不含), 地址必须 4 字节对齐。	XXXXh
[1:0]	Reserved	RO	保留。	00b

编码器模式控制寄存器 (R8_ENC_REG_CTRL)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	0
5	RB_ENC_DIR	RO	编码器当前方向: 0: 前进; 1: 后退。	0
4	Reserved	RO	保留。	0
3	RB_RD_CLR_EN	RW	编码器模式读计数并清零。	0

[2:1]	RB_SMS_MODE	RW	编码器模式边沿工作模式： 00 = IDLE； 10 = 在 T1 边沿计数； 01 = 在 T2 边沿计数； 11 = 在 T1 和 T2 边沿计数。	0
0	RB_START_ENC_EN	RW	编码器模式工作使能。 1: 使能（必须打开 LSI）； 0: 禁止。	0

编码器模式中断使能寄存器 (R8_ENC_INTER_EN)

位	名称	访问	描述	复位值
[7:2]	Reserved	RO	保留。	0
1	RB_IE_DIR_DEC	RW	编码器模式后退中断使能。 1: 使能中断； 0: 禁止中断。	0
0	RB_IE_DIR_INC	RW	编码器模式前进中断使能。 1: 使能中断； 0: 禁止中断。	0

编码器模式中断标志寄存器 (R8_ENC_INT_FLAG)

位	名称	访问	描述	复位值
[7:2]	Reserved	RO	保留。	0
1	RB_IF_DIR_DEC	RW	编码器模式后退中断标志, 写 1 清 0。 1: 编码器模式后退； 0: 编码器模式保持不变。	0
0	RB_IF_DIR_INC	RW	编码器模式前进中断标志, 写 1 清 0。 1: 编码器模式前进； 0: 编码器模式保持不变。	0

编码器模式终值配置寄存器 (R32_ENC_REG_CEND)

位	名称	访问	描述	复位值
[31:0]	R32_ENC_REG_CEND	RW	预设编码器模式终值。	0

编码器模式当前值配置寄存器 (R32_ENC_REG_CCNT)

位	名称	访问	描述	复位值
[31:0]	R32_ENC_REG_CCNT	RO	当前编码器模式值。	0

8.3 功能描述及配置

8.3.1 定时、计数功能

定时器 TMR 支持最长定时时间 2^{26} 个时钟周期, 执行增量计数模式。如果系统时钟周期为 32MHz, 则最长定时时间为: $31.25\text{ns} \times 2^{26} \approx 2\text{s}$ 。定时器还支持独立的中断。

定时功能操作步骤如下:

- (1)、将 RB_TMR_ALL_CLEAR 置位, 清零 R32_TMR_COUNT 和中断标志等;
- (2)、设置寄存器 R32_TMR_CNT_END 为需要定时的时间值;
具体计算方法为: $\text{Time} = \text{Ts}_{\text{sys}} \times \text{R32_TMR_CNT_END}$;
- (3)、清零 RB_TMR_ALL_CLEAR, 清零 RB_TMR_MODE_IN 对应定时模式;
- (4)、可选步骤, 设置 R8_TMR_INTER_EN 寄存器, 置位 RB_TMR_IE_CYC_END 可打开定时周期中断;

- (5)、将 R8_TMR_CTRL_MOD 寄存器的 RB_TMR_COUNT_EN 置位，启动定时器计数；
 (6)、当计数到 R32_TMR_COUNT 等于 R32_TMR_CNT_END 时，定时完成。此时 R8_TMR_INT_FLAG 的 RB_TMR_IF_CYC_END 将置 1，可写 1 清零。

计数功能操作步骤如下：

- (1)、将计数对应的 I/O 引脚方向设置为输入；
- (2)、在 R32_TMR_CNT_END 中设定计数溢出终值；
- (3)、配置 R8_TMR_CTRL_MOD，置位 RB_TMR_MODE_IN 和 RB_TMR_CAP_COUNT 对应计数模式，清零 RB_TMR_ALL_CLEAR 位，通过 RB_TMR_CAP_EDGE 选择采样边沿方式，将 R8_TMR_CTRL_MOD 的 RB_TMR_COUNT_EN 置 1，使能计数功能；
- (4)、可选步骤，如果需要启用中断则设置相应的中断使能寄存器位；
- (5)、R32_TMR_COUNT 存放当前计数值，每次计数达到计数终值，RB_TMR_IE_CYC_END 都会置 1 并且 R32_TMR_COUNT 清 0，如果开启中断则会触发硬件中断。

8.3.2 PWM 功能

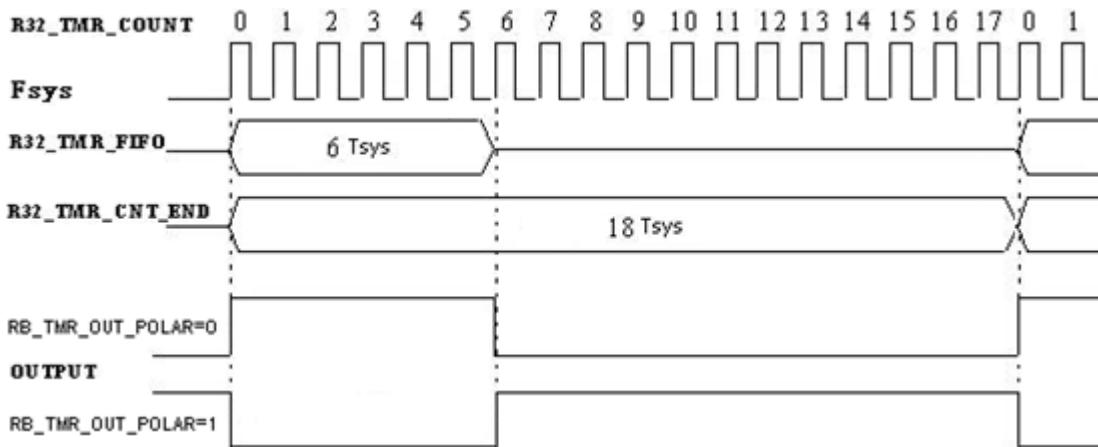
定时器 TMR 具有 PWM 功能，该 PWM 功能支持 DMA 数据加载。PWM 可设置默认输出极性为高电平或低电平，同一数据重复输出次数可选为 1, 4, 8 或 16 次，该重复功能结合 DMA 可以用于模仿 DAC 的效果。PWM 输出有效电平的最短时间单位为 1 个系统时钟，可动态修改 PWM 的占空比，模仿出特殊波形。

PWM 操作步骤如下：

- (1)、将 RB_TMR_ALL_CLEAR 置位，清空和清零 R32_TMR_FIFO 和中断标志等；
- (2)、设置 PWM 总周期寄存器 R32_TMR_CNT_END，该值应该不小于 R32_TMR_FIFO 寄存器中的值；
- (3)、配置 R8_TMR_CTRL_MOD，清零 RB_TMR_ALL_CLEAR，清零 RB_TMR_MODE_IN 对应 PWM 模式，通过 RB_TMR_OUT_POLAR 选择输出极性，根据需要通过 RB_TMR_PWM_REPEAT 选择同一数据重复次数；
- (4)、设置数据寄存器 R32_TMR_FIFO，最小值为 0，对应占空比 0%，最大值同 R32_TMR_CNT_END，对应占空比 100%，占空比计算：R32_TMR_FIFO/R32_TMR_CNT_END。TMR 可通过 DMA 加载连续动态数据，结合同一数据重复输出次数，可以模仿出特殊波形；
- (5)、配置 R8_TMR_CTRL_MOD，置位 RB_TMR_COUNT_EN 启动计数和 RB_TMR_OUT_EN 允许 PWM 输出；
- (6)、将 PWM 对应的 I/O 引脚设置为输出；
- (7)、可选步骤，如果需要启用中断则设置相应的中断使能寄存器位；
- (8)、PWM 一个周期完成后，如果开启中断则当 RB_TMR_IF_DATA_ACT 或 RB_TMR_IF_CYC_END 置位后会触发硬件中断；
- (9)、更新 R32_TMR_FIFO 中数据可以动态改变 PWM 的占空比，建议通过 DMA 加载。

例如：设置 RB_TMR_OUT_POLAR 位为 0, R32_TMR_FIFO 为 6, R32_TMR_CNT_END 为 18，则产生 PWM 的基本时序图如下所示，其占空比为：R32_TMR_FIFO/R32_TMR_CNT_END = 1/3

图 8-1 PWM 输出时序图



如果 RB_TMR_PWM_REPEAT 设置为 00 则表示上述过程重复 1 次，01 表示重复 4 次，10 表示重复 8 次，11 表示重复 16 次。重复之后再加载 FIFO 中的下一个数据继续。

8.3.3 捕捉功能

定时器 TMR 具备捕捉功能，该捕捉功能支持 DMA 数据存储。捕捉模式可以选择任何边沿触发开始至任何边沿触发结束、上升沿触发开始至上升沿触发结束、下降沿触发开始至下降沿触发结束三种模式。以下为捕捉触发模式说明表：

表 8-5 捕捉触发模式说明表

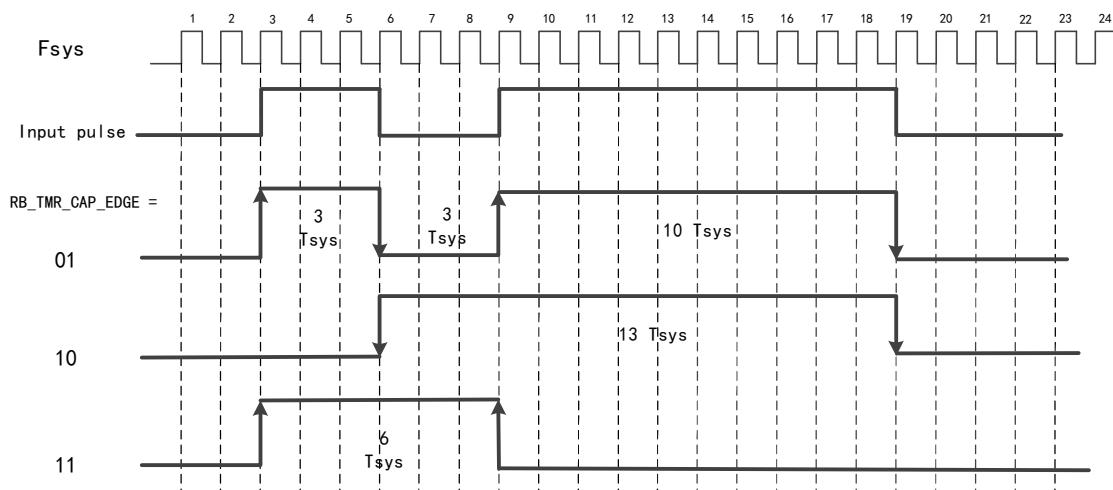
捕捉模式选择位 RB_TMR_CATCH_EDGE	触发方式	图示
00	不捕捉	无
01	边沿触发 边沿至边沿	
10	下降沿至下降沿	
11	上升沿至上升沿	

边沿触发模式下有 2 种触发状态，可以捕捉高电平宽度或低电平宽度。数据寄存器 R32_TMR_FIFO 的有效数据的最高位（位 25）为 1 表示捕捉到的是高电平，为 0 表示捕捉到的是低电平。如果连续多组数据的位 25 都是 1（或 0），说明该高（或低）电平的宽度超过超时值，需多组合并累计。

下降沿至下降沿、上升沿至上升沿触发模式下，可以捕捉一个输入变化周期。数据寄存器 R32_TMR_FIFO 的有效数据的最高位（位 25）为 0 表示正常采样到一个周期，为 1 则表示输入变化周期超过超时值 R32_TMR_CNT_END，需加上后一组数据累计为单个输入变化周期。

具体说明如下图所示：

图 8-2 以系统时钟计数捕捉周期



如上图所示，每个时钟周期内采样一次：

当 RB_TMR_CATCH_EDGE=01b 时，设置为边沿触发采样，采样到的时间宽度为 3、3、10；

当 RB_TMR_CATCH_EDGE=10b 时，设置为下降沿至下降沿采样，采样到的时间宽度为 13；

当 RB_TMR_CATCH_EDGE=11b 时，设置为上升沿至上升沿采样，采样到的时间宽度为 6。

捕捉模式操作步骤：

- (1)、将 RB_TMR_ALL_CLEAR 置位，清空和清零 R32_TMR_FIFO 和中断标志等；
- (2)、将捕捉对应的 I/O 引脚方向设置为输入；
- (3)、在 R32_TMR_CNT_END 中设定合理的捕捉超时时间，可用于在输入信号长时间无变化时产生超时中断，也可以在输入信号无变化超时后，产生超时数据（数据的位 25 为 1，低 25 位可向后累计）；
- (4)、配置 R8_TMR_CTRL_MOD，置位 RB_TMR_MODE_IN 对应捕捉模式，通过 RB_TMR_CAP_EDGE 选择捕捉

的边沿方式，将 R8_TMR_CTRL_MOD 的 RB_TMR_COUNT_EN 置 1，使能计数；
 (5)、可选步骤，如果需要启用中断则设置相应的中断使能寄存器位；
 (6)、如果需要采用 DMA 方式保存捕捉的数据，需要设置寄存器 R16_TMR_DMA_BEG 为存储捕捉数据缓冲区的首地址，设置寄存器 R16_TMR_DMA_END 为存储捕捉数据缓冲区的结束地址（不含），并设置 R8_TMR_CTRL_DMA 的 RB_TMR_DMA_ENABLE 为 1，使能 DMA 功能；
 (7)、将 R8_TMR_CTRL_MOD 的 RB_TMR_ALL_CLEAR 清零，启动捕捉功能；
 (8)、每次捕捉到数据，RB_TMR_IF_DATA_ACT 都会置 1，如果开启中断则会触发硬件中断，捕捉到的数据默认存放在 R32_TMR_FIFO 中，如果开启 DMA，则捕捉到的数据会自动存放在 DMA 设置的数据缓冲区中。

8.3.4 编码器模式

定时器 TMR 模块支持编码器模式。提供了 2 路输入通道 CAP_IN1 和 CAP_IN2，来捕捉 2 路旋转编码器信号进行编码。可在一路捕捉信号边沿处比较另一路捕捉信号的高低电平，判断编码方向。

选择编码器模式配置：两个输入通道 CAP_IN1 和 CAP_IN2 被用来作为旋转编码器的接口，如果计数器只在 CAP_IN2 的边沿计数，则置 R8_ENC_REG_CTRL 寄存器中的 RB_SMS_MODE = 01；如果只在 CAP_IN1 边沿计数，则置 RB_SMS_MODE = 10；如果计数器同时在 CAP_IN1 和 CAP_IN2 边沿计数，则置 RB_SMS_MODE = 11。

依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 R8_ENC_REG_CTRL 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 CAP_IN1 计数、依靠 CAP_IN2 计数或者同时依靠 CAP_IN1 和 CAP_IN2 计数。在任一输入端(CAP_IN1 或者 CAP_IN2)的跳变都会重新计算 RB_ENC_DIR 位。

在开始计数之前必须配置 R32_ENC_REG_CEND。

表 8-6 技术方向与编码信号关系

有效边沿	相对信号的电平	通道 CAP_IN1 对应的信号		通道 CAP_IN2 对应的信号	
		上升	下降	上升	下降
仅在 CAP_IN1 计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在 CAP_IN2 计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在 CAP_IN1 和 CAP_IN2 计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

捕捉旋转编码器信号进行编码的操作步骤：

- (1)、设置 RB_SMS_MODE，配置编码边沿工作模式，若需要休眠唤醒，需开启唤醒使能；
- (2)、配置 R32_ENC_REG_CEND，值不为 0 时，方可工作；
- (3)、RB_START_ENC_EN 置 1，开始工作；
- (4)、RB_RD_CLR_EN 置 1，则在每次读取 R32_ENC_REG_CCNT 后自动清零当前编码值；在每次系统唤醒后自动清零唤醒信号。

8.3.5 DMA 功能

定时器 TMR 具有 DMA 功能，在使用 DMA 完成中断时需要注意相关寄存器的配置顺序。

使能 DMA 完成中断的步骤如下：

- (1)、读取 R16_TMR_DMA_NOW，将 R16_TMR_DMA_END 赋值为任一不和 R16_TMR_DMA_NOW 相等的值（例：可以取值为 R16_TMR_DMA_NOW + 0x100）；
- (2)、将 R8_TMR_INT_FLAG 的位 RB_TMR_IF_DMA_END 写 1 清零；
- (3)、将 R8_TMR_INTER_EN 中的位 RB_TMR_IE_DMA_END 置 1。

非循环模式清除 DMA 完成中断的步骤如下：

- (1)、读取 R16_TMR_DMA_NOW，将 R16_TMR_DMA_END 赋值为任一不和 R16_TMR_DMA_NOW 相等的值

(例：可以取值为 R16_TMR_DMA_NOW + 0x100)；

(2)、将 R8_TMR_INT_FLAG 的位 RB_TMR_IF_DMA_END 写 1 清零。

循环模式清除 DMA 完成中断的步骤如下：

(1)、将 R8_TMR_INT_FLAG 的位 RB_TMR_IF_DMA_END 写 1 清零。

第 9 章 通用异步收发器 UART

9.1 UART 简介

CH572 和 CH570 提供了 1 组全双工的异步串口 UART。支持全双工串口通讯。

9.1.1 主要特性

- 兼容 16C550 异步串口并且有所增强。
- 支持 5、6、7 或者 8 个数据位以及 1 或者 2 个停止位。
- 支持奇、偶、无校验、空白 0、标志 1 等校验方式。
- 可编程通讯波特率，最高达 12.5Mbps 波特率。
- 内置 8 个字节的 FIFO 先进先出缓冲器，支持 4 个 FIFO 触发级。
- 支持串口帧错误检测、支持 Break 线路间隔检测。
- 支持全双工串口通讯。

9.2 寄存器描述

表 9-1 UART 相关寄存器列表

名称	访问地址	描述	复位值
R8_UART_MCR	0x40003400	调制解调器 MODEM 控制寄存器	0x00
R8_UART_IER	0x40003401	中断使能寄存器	0x00
R8_UART_FCR	0x40003402	FIFO 控制寄存器	0x00
R8_UART_LCR	0x40003403	线路控制寄存器	0x00
R8_UART_IIR	0x40003404	中断识别寄存器	0x01
R8_UART_LSR	0x40003405	线路状态寄存器	0x60
R8_UART_RBR	0x40003408	接收缓冲寄存器	0xXX
R8_UART_THR	0x40003408	发送保持寄存器	0xXX
R8_UART_RFC	0x4000340A	接收 FIFO 计数寄存器	0x00
R8_UART_TFC	0x4000340B	发送 FIFO 计数寄存器	0x00
R16_UART_DL	0x4000340C	波特率除数锁存器	0xFFFF
R8_UART_DIV	0x4000340E	预分频除数寄存器	0xXX

调制解调器 MODEM 控制寄存器 (R8_UART_MCR)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
3	RB_MCR_OUT2 RB_MCR_INT_OE	RW	串口的中断请求输出控制位： 1：允许发出请求；0：禁止。	0
[2:0]	Reserved	RO	保留。	0

中断使能寄存器 (R8_UART_IER)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
6	RB_IER_TXD_EN	RW	串口 TXD 引脚输出使能位： 1：使能引脚输出； 0：禁止引脚输出。	0
[5:3]	Reserved	RO	保留。	0
2	RB_IER_LINE_STAT	RW	接收线路状态中断使能位： 1：使能中断；	0

			0: 禁止中断。 发送保持寄存器空中断使能位: 1: 使能中断; 0: 禁止中断。	
1	RB_IER_THR_EMPTY	RW	发送保持寄存器空中断使能位: 1: 使能中断; 0: 禁止中断。	0
0	RB_IER_RECV_RDY	RW	接收数据中断使能位: 1: 使能中断; 0: 禁止中断。	0

FIFO 控制寄存器 (R8_UART_FCR)

位	名称	访问	描述	复位值
[7:6]	RB_FCR_FIFO_TRIG	RW	接收 FIFO 的中断和硬件流控制的触发点选择: 00: 1 字节; 01: 2 字节; 10: 4 字节; 11: 7 字节。 用来设置接收 FIFO 的中断和硬件流控制的触发点, 例如: 10 对应 4 个字节, 即接收满 4 个字节产生接收数据可用的中断, 并在使能硬件流控制时自动无效 RTS 引脚。	0
[5:1]	Reserved	RO	保留。	0
0	RB_FCR_FIFO_EN	RW	FIFO 使能位: 1: 启用 8 字节 FIFO; 0: 禁用 FIFO。 禁用 FIFO 后为 16C450 兼容模式, 相当于 FIFO 只有一个字节 (RECV_TG1=0、RECV_TG0=0、FIFO_EN=1), 建议启用。	0

线路控制寄存器 (R8_UART_LCR)

位	名称	访问	描述	复位值
7	RB_LCR_DLAB RB_LCR_GP_BIT	RW	串口通用位, 用户自定义。	0
6	RB_LCR_BREAK_EN	RW	强制产生 BREAK 线路间隔使能位: 1: 强制产生; 0: 不产生。	0
[5:4]	RB_LCR_PAR_MOD	RW	奇偶校验位格式选择: 00: 奇校验; 01: 偶校验; 10: 标志位 (MARK, 置 1); 11: 空白位 (SPACE, 清 0)。 仅当 RB_LCR_PAR_EN 位为 1 时有效。	00b
3	RB_LCR_PAR_EN	RW	奇偶校验位使能位: 1: 允许发送时产生和接收时校验奇偶校验位; 0: 无奇偶校验位。	0
2	RB_LCR_STOP_BIT	RW	停止位格式设置位: 0: 一个停止位; 1: 两个停止位。	0

[1:0]	RB_LCR_WORD_SZ	RW	串口数据长度选择： 00: 5 个数据位； 01: 6 个数据位； 10: 7 个数据位； 11: 8 个数据位。	00b
-------	----------------	----	---	-----

中断识别寄存器(R8_UART_IIR)

位	名称	访问	描述	复位值
[7:6]	RB_IIR_FIFO_ID	R0	串口 FIFO 启用状态位： 11: FIFO 已启用； 00: FIFO 未启用。	00b
[5:4]	Reserved	R0	保留。	00b
[3:1]	RB_IIR_INT_MASK	R0	中断标志：如果 RB_IIR_NO_INT 位为 0，则表示有中断产生，需要读取后判断中断源。具体请参看表 9-5。	000b
0	RB_IIR_NO_INT	R0	串口无中断标志位： 1: 无中断； 0: 有中断。	1

中断识别寄存器 R8_UART_IIR 的 RB_IIR_NO_INT 位以及 RB_IIR_INT_MASK 的每一个位所表示的含义如下表所示：

表 9-5 IIR 寄存器中 RB_IIR_INT_MASK 含义

IIR 寄存器位				优先级	中断类型	中断源	清中断方法
IID3	IID2	IID1	NOINT				
0	0	0	1	无	没有中断产生	没有中断	-
1	1	1	0	0	Reserved	Reserved	Reserved
0	1	1	0	1	接收线路状态	OVER_ERR、PAR_ERR、FRAM_ERR、BREAK_ERR	读 LSR
0	1	0	0	2	接收数据可用	接收到的字节数达到 FIFO 的触发点。	读 RBR
1	1	0	0	2	接收数据超时	超过 4 个数据时间未收到下一数据。	读 RBR
0	0	1	0	3	THR 寄存器空	发送保持寄存器空，或者 RB_IER_THR_EMPTY 位从 0 变 1 触发。	读 IIR 或写 THR
0	0	0	0	4	Reserved	Reserved	Reserved

线路状态寄存器(R8_UART_LSR)

位	名称	访问	描述	复位值
7	RB_LSR_ERR_RX_FIFO	R0	接收 FIFO 错误标志位： 1: 接收 FIFO 中存在至少一个 PAR_ERR、FRAM_ERR 或 BREAK_ERR 错误； 0: 接收 FIFO 中不存在错误。	0
6	RB_LSR_TX_ALL_EMP	R0	发送保持寄存器 THR 和发送移位寄存器 TSR 全空标志位： 1: 两者全空； 0: 两者非全空。	1
5	RB_LSR_TX_FIFO_EMP	R0	发送 FIFO 空标志位： 1: 发送 FIFO 空； 0: 发送 FIFO 非空。	1
4	RB_LSR_BREAK_ERR	RZ	BREAK 线路间隔检测标志位： 1: 检测到 BREAK； 0: 未检测到 BREAK。	0

3	RB_LSR_FRAME_ERR	RZ	数据帧错误标志位： 1: 表示正在从接收 FIFO 中读取的数据存在帧错误，缺少有效的停止位； 0: 当前读取的数据帧没有错误。	0
2	RB_LSR_PAR_ERR	RZ	接收数据奇偶校验错误标志位： 1: 表示正在从接收 FIFO 中读取的数据存在奇偶校验错； 0: 当前读取的数据奇偶校验正确。	0
1	RB_LSR_OVER_ERR	RZ	接收 FIFO 缓冲区溢出标志位： 1: 已溢出； 0: 未溢出。	0
0	RB_LSR_DATA_RDY	RO	接收 FIFO 中有接收到的数据标志位： 1: FIFO 中有数据； 0: 无数据。 读取 FIFO 中所有数据后，该位自动清 0。	0

接收缓冲寄存器 (R8_UART_RBR)

位	名称	访问	描述	复位值
[7:0]	R8_UART_RBR	RO	数据接收缓冲寄存器。 如果 LSR 的 DATA_RDY 位为 1，则可以从该寄存器读取接收到的数据； 如果 FIFO_EN 为 1，则从串口移位寄存器 RSR 接收到的数据首先被存放于接收 FIFO 中，然后通过该寄存器读出。	XXh

发送保持寄存器 (R8_UART_THR)

位	名称	访问	描述	复位值
[7:0]	R8_UART_THR	WO	发送保持寄存器。 包括发送 FIFO，用于写入准备发送的数据；如果 FIFO_EN 为 1，则写入的数据首先被存放于发送 FIFO 中，然后通过发送移位寄存器 TSR 逐个输出。	XXh

接收 FIFO 计数寄存器 (R8_UART_RFC)

位	名称	访问	描述	复位值
[7:0]	R8_UART_RFC	RO	当前接收 FIFO 中数据计数。	00h

发送 FIFO 计数寄存器 (R8_UART_TFC)

位	名称	访问	描述	复位值
[7:0]	R8_UART_TFC	RO	当前发送 FIFO 中数据计数。	00h

波特率除数锁存器 (R16_UART_DL)

位	名称	访问	描述	复位值
[15:0]	R16_UART_DL	RW	16 位除数用于计算波特率。 公式：除数 = 串口内部基准时钟 Fuart / 16 / 所需通讯波特率。 例：如果串口内部基准时钟 Fuart 为 1.8432MHz，所需波特率为 9600bps，则	XXXXh

		除数=1843200/16/9600=12。	
--	--	------------------------	--

预分频除数寄存器 (R8_UART_DIV)

位	名称	访问	描述	复位值
[7:0]	R8_UART_DIV	RW	用于计算串口的内部基准时钟，低 7 位有效。 公式：除数 = Fsys * 2 / 串口内部基准时钟，最大值 127。	XXh

9.3 功能描述及配置

UART 的输出引脚都是 3.3V LVC MOS 电平。异步串口方式下引脚包括：数据传输引脚。数据传输引脚包括：TXD 引脚和 RXD 引脚，默认都是高电平。

UART 拥有独立的收发缓冲区及 8 字节 FIFO，支持全双工异步串行通讯。串行数据包括 1 个低电平起始位，5、6、7 或 8 个数据位，0 个或者 1 个附加校验位或者标志位，1 个或者 2 个高电平停止位，支持奇校验/偶校验/标志校验/空白校验。串口发送信号的波特率误差小于 0.5%，串口接收信号的允许波特率误差不大于 2%。

9.3.1 波特率计算

1) 计算串口内部基准时钟 Fuart，设置 R8_UART_DIV 寄存器，最大值 127，通常写入 1。

2) 计算波特率，设置 R16_UART_DL 寄存器。

波特率公式 $= F_{sys} * 2 / R8_UART_DIV / 16 / R16_UART_DL$ 。

9.3.2 串口发送

串口发送的“THR 寄存器空”中断 UART_I1_THR_EMPTY 是指当前发送 FIFO 空。当读取 IIR 寄存器后，该中断被清除，或者当向 THR 写入下一个数据后，该中断也能被清除。如果仅仅是向 THR 写入一个字节，那么由于该字节很快被转移到发送移位寄存器 TSR 中开始发送，所以很快会再次产生发送 THR 寄存器空中断的请求，此时可以写入下一个准备发送的数据。当 TSR 寄存器中的数据被全部移出后，串口发送才真正完成，此时 LSR 寄存器的 RB_LSR_TX_ALL_EMP 位变为 1 有效。

在中断触发方式下，当收到串口发送保持寄存器 THR 空的中断后，如果已使能 FIFO，那么可以向 THR 寄存器及 FIFO 一次写入最多 8 字节，然后控制器会按顺序自动发送；如果禁止 FIFO，那么一次只能写入一个字节；如果没有数据需要发送，那么可以直接退出（之前读取 IIR 时已经自动清除中断）。

在查询方式下，可以根据 LSR 寄存器的 RB_LSR_TX_FIFO_EMP 位判断发送 FIFO 是否为空，当此位为 1 则可以向 THR 寄存器及 FIFO 写入数据，如果使能 FIFO，那么一次可以写入最多 8 个字节。

也可读取 R8_UART_TFC 寄存器判断当前 FIFO 中待发送的剩余数据个数，如果不等于 8，则可继续向 FIFO 中写入待发送数据，这种方式可以节约填充时间。

9.3.3 串口接收

串口接收数据可用中断 UART_I1_RECV_RDY 是指接收 FIFO 中的已有数据字节数已经到或超过由 FCR 寄存器的 RB_FCR_FIFO_TRIG 设置选择的 FIFO 触发点。当从 RBR 读取数据使 FIFO 字数低于 FIFO 触发点时，该中断被清除。

串口接收数据超时中断 UART_I1_RECV_TOUT 是指接收 FIFO 中至少有一个字节的数据，并且从上一次串口接收到数据和从上一次被系统取走数据开始，已经等待了相当于接收 4 个数据的时间。当再次接收到一个新的数据后，该中断被清除，或者当单片读取一次 RBR 寄存器后，该中断也能被清除。当接收 FIFO 全空时，LSR 寄存器的 RB_LSR_DATA_RDY 位为 0，当接收 FIFO 中有数据时，RB_LSR_DATA_RDY 位为 1 有效。

在中断触发方式下，当收到串口接收数据超时的中断后，可以读取 R8_UART_RFC 寄存器查询当前 FIFO 中剩余数据计数，直接读取全部数据，或者不断查询 LSR 寄存器的 RB_LSR_DATA_RDY，如果此位

有效则读数据，直到此位无效。当收到串口接收数据可用的中断后，可以先从 RBR 寄存器一次性读取 RB_FCR_FIFO_TRIG 设定字节个数的数据，或者也可以根据 RB_LSR_DATA_RDY 位和 R8_UART_RFC 寄存器读取当前 FIFO 中所有数据。

在查询方式下，可以根据 LSR 寄存器的 RB_LSR_DATA_RDY 位判断接收 FIFO 是否为空，或读取 R8_UART_RFC 寄存器获取当前 FIFO 中数据计数，来获取串口接收的所有数据。

第 10 章 串行外设接口 SPI

10.1 SPI 简介

SPI 是一种全双工串行接口，总线上连接有一个主机和若干从机，同一时刻，仅有一对主从在通讯。通常 SPI 接口由 4 个引脚组成：SPI 片选引脚 SCS、SPI 时钟引脚 SCK、SPI 串行数据引脚 MISO（主机输入/从机输出引脚）和 SPI 串行数据引脚 MOSI（主机输出/从机输入引脚）。

10.1.1 主要特性

CH572 和 CH570 芯片提供 1 个 SPI 接口，特性如下：

- 支持主机模式（Master）和从机模式（Slave）。
- 兼容串行外设接口（SPI）规范。
- 支持模式 0 和模式 3 数据传输方式。
- 8 位数据传输方式，数据位序可选：字节低位在前或者高位在前。
- 时钟频率最高可达系统主频 F_{sys} 的一半。
- 8 字节 FIFO。
- 从机模式支持首字节为命令模式或数据流模式。
- 支持 DMA，数据传输效率更高。

10.2 寄存器描述

表 10-1 SPI 相关寄存器列表

名称	访问地址	描述	复位值
R8_SPI_CTRL_MOD	0x40004000	SPI 模式控制寄存器	0x02
R8_SPI_CTRL_CFG	0x40004001	SPI 配置寄存器	0x00
R8_SPI_INTER_EN	0x40004002	SPI 中断使能寄存器	0x00
R8_SPI_CLOCK_DIV	0x40004003	SPI 主机模式时钟分频寄存器 SPI 从机模式预置数据寄存器	0x10
R8_SPI_SLAVE_PRE			
R8_SPI_BUFFER	0x40004004	SPI 数据缓冲区	0XX
R8_SPI_RUN_FLAG	0x40004005	SPI 工作状态寄存器	0x00
R8_SPI_INT_FLAG	0x40004006	SPI 中断标志寄存器	0x40
R8_SPI_FIFO_COUNT	0x40004007	SPI 收发 FIFO 计数寄存器	0x00
R8_SPI_INT_TYPE	0x40004008	SPI 中断触发方式选择寄存器	0x00
R8_SPI_INTER1_EN	0x40004009	SPI 中断 1 使能寄存器	0x00
R8_SPI_INT1_FLAG	0x4000400A	SPI 中断 1 标志寄存器	0x00
R16_SPI_TOTAL_CNT	0x4000400C	SPI 收发数据总长度寄存器	0x0000
R8_SPI_FIFO	0x40004010	SPI 数据 FIFO 寄存器	0XX
R8_SPI_FIFO_COUNT1	0x40004013	SPI 收发 FIFO 计数寄存器	0x00
R16_SPI_DMA_NOW	0x40004014	SPI DMA 缓冲区当前地址	0XXXX
R16_SPI_DMA_BEG	0x40004018	SPI DMA 缓冲区起始地址	0XXXX
R16_SPI_DMA_END	0x4000401C	SPI DMA 缓冲区结束地址	0XXXX

SPI 模式控制寄存器 (R8_SPI_CTRL_MOD)

位	名称	访问	描述	复位值
7	RB_SPI_MISO_OE	RW	MISO 引脚输出使能位（可在 2 线模式数据线切换方向使用）： 1：MISO 输出使能； 0：MISO 输出禁止。	0
6	RB_SPI_MOSI_OE	RW	MOSI 引脚输出使能位：	0

			1: MOSI 输出使能； 0: MOSI 输出禁止。	
5	RB_SPI_SCK_OE	RW	SCK 引脚输出使能位： 1: SCK 输出使能； 0: SCK 输出禁止。	0
4	RB_SPI_FIFO_DIR	RW	FIFO 方向设置位： 1: 输入模式（指示接收数据）； 0: 输出模式（指示发送数据）。	0
3	RB_SPI_SLV_CMD_MOD	RW	SPI 从机模式下首字节方式选择： 1: 首字节命令模式； 0: 数据流模式。 在首字节命令模式下，当接收到 SPI 片选有效后的首字节数据后，将视为命令码，且 RB_SPI_IF_FST_BYTE 将置 1。	0
3	RB_SPI_MST_SCK_MOD	RW	主机模式时钟空闲方式选择： 1: 模式 3（空闲时 SCK 为高电平）； 0: 模式 0（空闲时 SCK 为低电平）。	0
2	RB_SPI_2WIRE_MOD	RW	2 线或 3 线 SPI 模式选择： 1: 2 线模式/半双工（从机为 MISO/主机为 MOSI）； 0: 3 线模式/全双工（SCK/MOSI/MISO）。	0
1	RB_SPI_ALL_CLEAR	RW	SPI 的 FIFO/计数器/中断标志清零： 1: 强制清空和清零； 0: 不清。	1
0	RB_SPI_MODE_SLAVE	RW	SPI 主从模式选择： 1: 从机模式； 0: 主机模式。	0

SPI 配置寄存器 (R8_SPI_CTRL_CFG)

位	名称	访问	描述	复位值
7	Reserved	R0	保留。	0
6	RB_SPI_MST_DLY_EN	RW	主机模式下输入延迟使能： 1: 使能，用于 SPI 时钟接近 Fsys 一半等高速应用； 0: 禁止，常规应用。	0
5	RB_SPI_BIT_ORDER	RW	SPI 数据位序选择： 1: 低位在前； 0: 高位在前。	0
4	RB_SPI_AUTO_IF	RW	使能访问 BUFFER/FIFO 时自动清除标志位 RB_SPI_IF_BYTE_END 的功能： 1: 使能； 0: 禁止。	0
3	Reserved	R0	保留。	0
2	RB_SPI_DMA_LOOP	RW	DMA 地址循环功能使能位： 1: 使能地址循环； 0: 禁止地址循环。 如果使能 DMA 地址循环，当 DMA 地址增加到设置的末尾地址时，自动循环指向设置的首地址。	0

1	RB_MST_CLK_SEL	RW	RB_SPI_MODE_SLAVE = 0 时, 为主时钟极性选择: 1: 极性反转; 0: 极性不变。 RB_SPI_MODE_SLAVE = 1 时, 为两线模式从机输入输出方向选择: 1: 从机输入; 0: 从机输出。	0
0	RB_SPI_DMA_ENABLE	RW	DMA 功能使能位: 1: 使能 DMA; 0: 禁止 DMA。	0

SPI 中断使能寄存器 (R8_SPI_INTER_EN)

位	名称	访问	描述	复位值
7	RB_SPI_IE_FST_BYTE	RW	从机模式的首字节命令模式下, 首字节接收中断使能位: 1: 使能接收到第一个字节中断; 0: 禁止接收到第一个字节中断。	0
[6:5]	Reserved	R0	保留。	00b
4	RB_SPI_IE_FIFO_OV	RW	从机模式下, FIFO 溢出 (接收时 FIFO 满或发送时 FIFO 空) 中断使能位: 1: 使能中断; 0: 禁止中断。	0
3	RB_SPI_IE_DMA_END	RW	DMA 结束中断使能位: 1: 使能中断; 0: 禁止中断。	0
2	RB_SPI_IE_FIFO_HF	RW	FIFO 使用过半中断使能位: 1: 使能中断; 0: 禁止中断。	0
1	RB_SPI_IE_BYTE_END	RW	SPI 单字节传输完成中断使能位: 1: 使能中断; 0: 禁止中断。	0
0	RB_SPI_IE_CNT_END	RW	SPI 全部字节传输完成中断使能位: 1: 使能中断; 0: 禁止中断。	0

SPI 主机模式时钟分频寄存器 (R8_SPI_CLOCK_DIV)

位	名称	访问	描述	复位值
[7:0]	R8_SPI_CLOCK_DIV	RW	主机模式分频系数, 最小值为 2, 最大值为 254。 Fsck = 系统主频 Fsys/分频系数。	10h

SPI 从机模式预置数据寄存器 (R8_SPI_SLAVE_PRE)

位	名称	访问	描述	复位值
[7:0]	R8_SPI_SLAVE_PRE	RW	从机模式下, 预置的首个返回数据。 用于接收首字节数据后的返回数据。	10h

SPI 数据缓冲区 (R8_SPI_BUFFER)

位	名称	访问	描述	复位值
[7:0]	R8_SPI_BUFFER	RW	SPI 数据发送和接收缓冲区。	XXh

SPI 工作状态寄存器 (R8_SPI_RUN_FLAG)

位	名称	访问	描述	复位值
7	RB_SPI_SLV_SELECT	R0	从机模式被片选状态位： 1: 正被选中； 0: 没有被片选。	0
6	RB_SPI_SLV_CS_LOAD	R0	从机模式被片选后首次加载状态位： 1: 正在加载 R8_SPI_SLAVE_PRE； 0: 尚未加载或者已完成。	0
5	RB_SPI_FIFO_READY	R0	FIFO 准备就绪状态位： 1: FIFO 就绪 (R16_SPI_TOTAL_CNT 非 0, 且接收时 FIFO 未满或发送时 FIFO 不空)； 0: FIFO 未准备好。	0
4	RB_SPI_SLV_CMD_ACT	R0	从机模式下命令接收完成状态位, 即交换完首字节数据： 1: 指示刚刚交换完成的是首字节； 0: 首字节尚未交换或不是首字节。	0
[3:0]	Reserved	R0	保留。	0000b

SPI 中断标志寄存器 (R8_SPI_INT_FLAG)

位	名称	访问	描述	复位值
7	RB_SPI_IF_FST_BYTE	RW1	从机模式下, 接收到首字节标志位： 1: 已接收到首字节； 0: 未接收到。	0
6	RB_SPI_FREE	R0	当前 SPI 空闲状态位： 1: 当前 SPI 空闲； 0: 当前 SPI 非空闲。	1
5	Reserved	R0	保留。	0
4	RB_SPI_IF_FIFO_OV	RW1	从机模式下, FIFO 溢出 (接收时 FIFO 满或发送时 FIFO 空) 标志位, 写 1 清零： 1: FIFO 溢出； 0: FIFO 未溢出。	0
3	RB_SPI_IF_DMA_END	RW1	DMA 完成标志位, 写 1 清零： 1: 已完成； 0: 未完成。	0
2	RB_SPI_IF_FIFO_HF	RW1	FIFO 使用过半 (接收时 FIFO \geq 4 或发送时 FIFO <4) 标志位, 写 1 清零： 1: FIFO 使用已过半； 0: FIFO 使用未过半。	0
1	RB_SPI_IF_BYTE_END	RW1	SPI 单字节传输完成标志位, 写 1 清零： 1: 单字节传输完成； 0: 传输未完成。	0
0	RB_SPI_IF_CNT_END	RW1	SPI 全部字节传输完成标志位, 写 1 清零： 1: 全部字节传输完成； 0: 传输未完成。	0

SPI 收发 FIFO 计数寄存器 (R8_SPI_FIFO_COUNT)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0h
[3:0]	R8_SPI_FIFO_COUNT	RW	当前 FIFO 中字节计数。	0h

SPI 收发数据总长度寄存器 (R16_SPI_TOTAL_CNT)

位	名称	访问	描述	复位值
[15:0]	R16_SPI_TOTAL_CNT	RW	主机模式下 SPI 数据收发总字节数, 低 12 位有效。在使用 DMA 时一次最多可以收发 4095 个字节。不支持从机模式。	0

SPI 数据 FIFO 寄存器 (R8_SPI_FIFO)

位	名称	访问	描述	复位值
[7:0]	R8_SPI_FIFO	RO/ WO	数据 FIFO 寄存器。	XXh

寄存器 R8_SPI_BUFFER 和 R8_SPI_FIFO 均为 SPI 数据相关寄存器, 主要区别在于:

读 R8_SPI_BUFFER 是取自 SPI 最近一次交换到的数据, 不影响 FIFO 和 R8_SPI_FIFO_COUNT,

主机模式下写 R8_SPI_BUFFER 是直接发送该字节, 从机模式下写操作未定义;

读 R8_SPI_FIFO 是取自 FIFO 中最早交换到的数据, 将减少 FIFO 和 R8_SPI_FIFO_COUNT,

写 R8_SPI_FIFO 是将数据暂存到 FIFO 中, 在从机模式下由外部 SPI 主机决定何时取走, 在主机模式下当 R16_SPI_TOTAL_CNT 非 0 时自动启动发送。

SPI 收发 FIFO 计数寄存器 (R8_SPI_FIFO_COUNT1)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0h
[3:0]	R8_SPI_FIFO_COUNT1	RW	当前 FIFO 中字节计数。 同 R8_SPI_FIFO_COUNT。	0h

SPI 中断触发方式选择寄存器 (R8_SPI_INT_TYPE)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
[4:0]	RB_SPI_INT_TYPE	RW	中断触发方式选择, 1=脉冲触发, 0=边沿触发 bit[4]: RB_SPI_IF_FIFO_FULL; bit[3]: RB_SPI_IF_FIFO_EMPTY; bit[2]: RB_SPI_IF_DMA_END; bit[1]: RB_SPI_IF_FIFO_HF; bit[0]: RB_SPI_IF_CNT_END。	0

SPI 中断 1 使能寄存器 (R8_SPI_INTER1_EN)

位	名称	访问	描述	复位值
[7:2]	Reserved	RO	保留。	0
1	RB_SPI_IE_FIFO_FULL	RW	当前 FIFO 数据满中断使能: 1: 开启使能; 0: 关闭使能。	0

0	RB_SPI_IE_FIFO_EMPTY	RW	当前 FIFO 数据空中断使能： 1: 开启使能； 0: 关闭使能。	0
---	----------------------	----	--	---

SPI 中断 1 标志寄存器 (R8_SPI_INT1_FLAG)

位	名称	访问	描述	复位值
[7:2]	Reserved	RO	保留。	0
0	RB_SPI_IF_FIFO_FULL	RW1	当前 FIFO 数据满标志位, 写 1 清零： 1: FIFO 已满； 0: FIFO 未满。	0
0	RB_SPI_IF_FIFO_EMPTY	RW1	当前 FIFO 数据空标志位, 写 1 清零： 1: FIFO 已空； 0: FIFO 未空。	0

SPI DMA 缓冲区当前地址 (R16_SPI_DMA_NOW)

位	名称	访问	描述	复位值
[15:14]	Reserved	RO	保留。	0
[13:0]	R16_SPI_DMA_NOW	RW	DMA 数据缓冲区当前地址。 可用于计算已转换次数, 计算方法: COUNT = SPI_DMA_NOW - SPI_DMA_BEG。	XXXXh

SPI DMA 缓冲区起始地址 (R16_SPI_DMA_BEG)

位	名称	访问	描述	复位值
[15:14]	Reserved	RO	保留。	0
[13:0]	R16_SPI_DMA_BEG	RW	DMA 数据缓冲区起始地址, 仅低 14 位有效。	XXXXh

SPI DMA 缓冲区结束地址 (R16_SPI_DMA_END)

位	名称	访问	描述	复位值
[15:14]	Reserved	RO	保留。	0
[13:0]	R16_SPI_DMA_END	RW	DMA 数据缓冲区结束地址 (不含), 仅低 14 位有效。	XXXXh

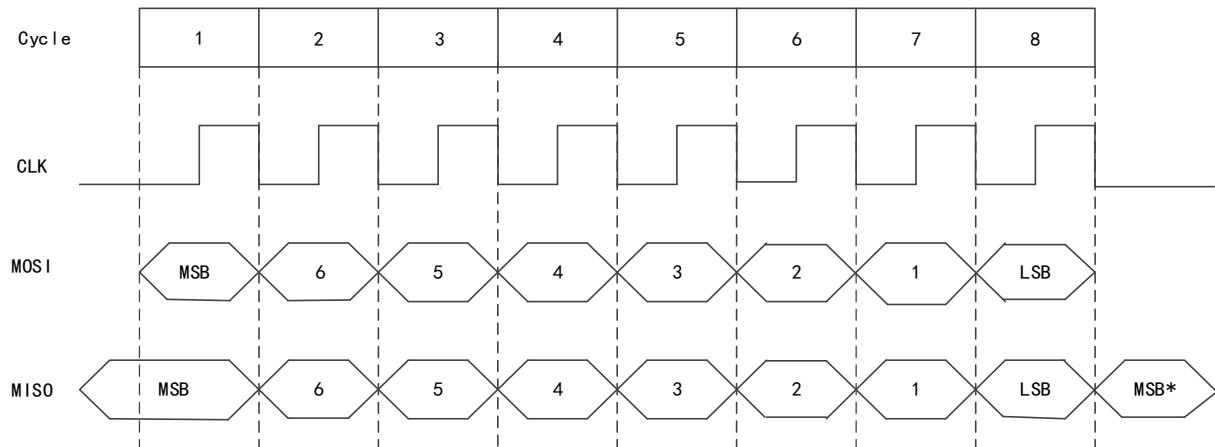
10.3 SPI 传输格式

SPI 支持模式 0 和模式 3 两种传输格式, 通过设置 R8_SPI_CTRL_MOD 的 RB_SPI_MST_SCK_MOD 进行选择。总是在 SCK 上升沿采样串行数据输入, 在下降沿输出串行数据。

数据传输格式如下图所示:

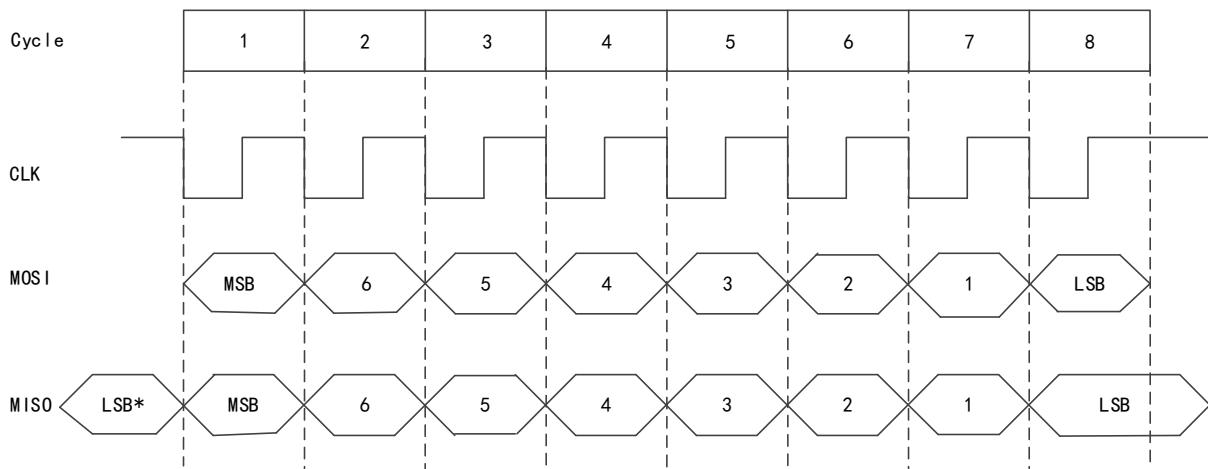
模式 0: RB_SPI_MST_SCK_MOD = 0

图 10-1 SPI 模式 0 传输格式



模式 3: RB_SPI_MST_SCK_MOD = 1

图 10-2 SPI 模式 3 传输格式



10.4 SPI 配置

10.4.1 SPI 主机模式

SPI 主机模式下，在 SCK 引脚产生串行时钟，片选引脚可以指定为任意 I/O 引脚。

配置步骤：

- (1)、设置 R8_SPI_CLOCK_DIV，配置 SPI 时钟频率；
- (2)、设置 R8_SPI_CTRL_MOD 的 RB_SPI_MODE_SLAVE 为 0，配置 SPI 为主机模式；
- (3)、设置 R8_SPI_CTRL_MOD 的 RB_SPI_MST_SCK_MOD，选择时钟空闲模式 0 或模式 3；
- (4)、设置 R8_SPI_CTRL_MOD 的 RB_SPI_FIFO_DIR 配置 FIFO 方向，为 1 则 FIFO 用于接收，为 0 则 FIFO 用于发送。
- (5)、设置 R8_SPI_CTRL_MOD 的 RB_SPI_MOSI_OE 和 RB_SPI_SCK_OE 为 1，RB_SPI_MISO_OE 为 0，并设置 GPIO 方向配置寄存器 (R32_PA/PB_DIR) 使 MOSI 引脚和 SCK 引脚为输出，MISO 引脚为输入；
- (6)、2 线模式下 SCK 不变，RB_SPI_MOSI_OE = 0，不用 MISO，由 MOSI 半双工实现输入（同 3 线模式，RB_SPI_MOSI_OE = 0 且引脚置为输入）和输出（RB_SPI_MOSI_OE = 1 且引脚置为输出），手工切换方向；
- (7)、可选步骤，如果启用 DMA，那么需将收发缓冲区起始地址写入 R16_SPI_DMA_BEG，结束地址（不含）写入 R16_SPI_DMA_END，建议在设置完 RB_SPI_FIFO_DIR 后再置位 RB_SPI_DMA_ENABLE，如果确认 R16_SPI_TOTAL_CNT 为 0，那么也可在此先设置 RB_SPI_DMA_ENABLE 为 1，使能 DMA 功能。

数据发送过程：

- (1)、设置 RB_SPI_FIFO_DIR 为 0，当前 FIFO 方向为输出；
- (2)、写 R16_SPI_TOTAL_CNT 寄存器，设置要发送的数据长度；
- (3)、写 R8_SPI_FIFO 寄存器，向 FIFO 中写入要发送的数据，如果 R8_SPI_FIFO_COUNT 小于 FIFO 容量则可以继续写 FIFO，如果是启用 DMA，则由 DMA 自动加载 FIFO 完成此步骤；
- (4)、只要 R16_SPI_TOTAL_CNT 非 0 并且 FIFO 中有数据，SPI 主机就会自动发送数据，否则暂停；
- (5)、等待 R16_SPI_TOTAL_CNT 寄存器为 0，说明数据发送完成，如果只发送一字节，也可以查询等待 RB_SPI_FREE 为空闲、或等待 R8_SPI_FIFO_COUNT 为 0。

数据接收过程：

- (1)、设置 RB_SPI_FIFO_DIR 为 1，当前 FIFO 方向为输入；
- (2)、写 R16_SPI_TOTAL_CNT 寄存器，设置要接收的数据长度；
- (3)、只要 R16_SPI_TOTAL_CNT 非 0 并且 FIFO 未满，SPI 主机就会自动接收数据，否则暂停；
- (4)、等待 R8_SPI_FIFO_COUNT 寄存器不为 0，则说明接收到返回数据，读取 R8_SPI_FIFO 中的值即为接收到的数据，如果是启用 DMA，则由 DMA 自动读取 FIFO 完成此步骤。

10.4.2 SPI 从机模式

SPI 支持从机模式，在从机模式下，SCK 引脚用于接收外部连接的 SPI 主机的串行时钟。

配置步骤：

- (1)、设置 R8_SPI_CTRL_MOD 的 RB_SPI_MODE_SLAVE 为 1，配置 SPI 为从机模式；
- (2)、根据需要设置 R8_SPI_CTRL_MOD 的 RB_SPI_SLV_CMD_MOD，选择从机首字节模式或数据流模式；
- (3)、设置 R8_SPI_CTRL_MOD 的 RB_SPI_FIFO_DIR，配置 FIFO 方向，为 1 则 FIFO 用于接收，为 0 则 FIFO 用于发送；
- (4)、设置 R8_SPI_CTRL_MOD 的 RB_SPI_MOSI_OE 和 RB_SPI_SCK_OE 为 0，RB_SPI_MISO_OE 为 1，并设置 GPIO 方向配置寄存器(R32_PA/PB_DIR)使 MOSI 引脚、SCK 引脚和 SCS 引脚为输入，MISO 引脚为输入（支持总线下多个从机连接，被片选后 MISO 会自动切换为输出，也支持一主一从）或输出（仅用于一主一从对连）。在 SPI 从机模式下 MISO 的 I/O 引脚方向，除了能够由 GPIO 方向配置寄存器设为输出之外，还支持 SPI 片选有效期间自动切换为输出，但其输出数据由 RB_SPI_MISO_OE 选择，为 1 时输出 SPI 数据，为 0 时输出 GPIO 数据输出寄存器的数据。建议，设置 MISO 引脚为输入，使 MISO 在片选无效时不输出，便于多机操作时共享 SPI 总线；
- (5)、2 线模式下 SCK 不变，RB_SPI_MISO_OE=0，不用 MOSI，由 MISO 半双工实现输入（同 3 线模式，RB_SPI_MISO_OE=0 且引脚置为输入）和输出（RB_SPI_MISO_OE=1 且引脚置为输出），手工切换方向；
- (6)、可选的，设置 SPI 从机模式预置数据寄存器 R8_SPI_SLAVE_PRE，用于被片选后首次自动加载到缓冲区中用于对外输出。当 8 个时钟过去之后（即首个数据字节在主从双方之间交换完毕），控制器得到外部 SPI 主机发来的首字节数据（命令码），外部 SPI 主机交换得到 R8_SPI_SLAVE_PRE 中的预置数据（状态值）。R8_SPI_SLAVE_PRE 的位 7 将在 SPI 片选有效后的 SCK 低电平期间自动加载到 MISO 引脚上，对于 SPI 模式 0 (CLK 默认为低电平)，如果预置了 R8_SPI_SLAVE_PRE 的位 7，那么外部 SPI 主机将在 SPI 片选有效但尚未传输数据时，就能够通过查询 MISO 引脚得到 R8_SPI_SLAVE_PRE 的位 7 的预置值，从而通过仅仅有效一下 SPI 片选就能快速获得 R8_SPI_SLAVE_PRE 的位 7 的值（通常是向主机提供一个忙状态，便于主机快速查询）；
- (7)、可选步骤，如果启用 DMA，那么需将收发缓冲区起始地址写入 R16_SPI_DMA_BEG，结束地址（不含）写入 R16_SPI_DMA_END，必须在设置完 RB_SPI_FIFO_DIR 后才能置位 RB_SPI_DMA_ENABLE。

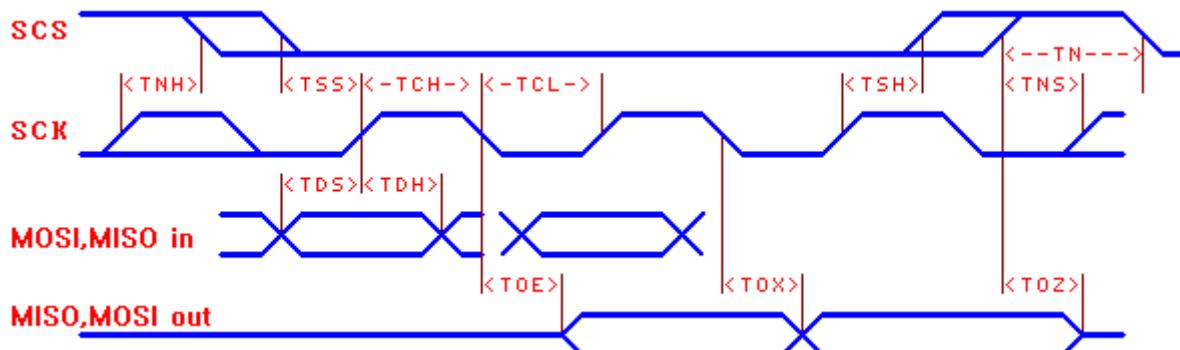
数据发送过程：

- (1)、设置 R8_SPI_CTRL_MOD 的 RB_SPI_FIFO_DIR 为 0，当前 FIFO 方向为输出；
- (2)、可选步骤，如果启用 DMA，那么需设置 RB_SPI_DMA_ENABLE 为 1，使能 DMA 功能；
- (3)、将多个发送数据写入到 FIFO 寄存器 R8_SPI_FIFO 中，由外部 SPI 主机决定何时取走，如果是启用 DMA，则由 DMA 自动加载 FIFO 完成此步骤；
- (4)、查询 R8_SPI_FIFO_COUNT，如果未满则继续向 FIFO 写入待发送的数据。

数据接收过程：

- (1)、设置 R8_SPI_CTRL_MOD 的 RB_SPI_FIFO_DIR 为 1, 当前 FIFO 方向为输入;
- (2)、可选步骤, 如果启用 DMA, 那么需设置 RB_SPI_DMA_ENABLE 为 1, 使能 DMA 功能;
- (3)、查询 R8_SPI_FIFO_COUNT, 如果不空则说明已接收到数据, 通过读取 R8_SPI_FIFO 取走数据, 如果是启用 DMA, 则由 DMA 自动读取 FIFO 完成此步骤;
- (4)、单个字节的数据接收, 也可以不使用 FIFO, 可以直接读取 R8_SPI_BUFFER。

10.5 SPI 时序



名称	参数说明 (TA = 25°C, VI033 = 3.3V)	最小值	典型值	最大值	单位
TSS	SCK 上升沿之前 SCS 有效的建立时间	Tsys*1.05			nS
TSH	SCK 上升沿之后 SCS 有效的保持时间	Tsys*1.05			nS
TNS	SCK 上升沿之前 SCS 无效的建立时间	15			nS
TNH	SCK 上升沿之后 SCS 无效的保持时间	15			nS
TN	SCS 无效时间 (SPI 操作间隔时间)	Tsys*2			nS
TCH	SCK 时钟的高电平时间	Tsys*0.55			nS
TCL	SCK 时钟的低电平时间	Tsys*0.55			nS
TDS	SCK 上升沿之前 MOSI/MISO 输入的建立时间	8			nS
TDH	SCK 上升沿之后 MOSI/MISO 输入的保持时间	5			nS
TOE	SCK 下降沿到 MOSI/MOSI 输出有效	0		18	nS
TOX	SCK 下降沿到 MOSI/MOSI 输出改变	0	5	16	nS
TOZ	SCS 无效到 MOSI/MOSI 输出无效	2		24	nS

注: T_{sys} 是系统主频时钟周期 ($1/F_{sys}$)。

第 11 章 脉冲宽度调制器 PWM

11.1 PWM 控制器简介

除了定时器提供的 1 路 26 位 PWM 输出之外, 系统还提供了 5 路 16 位或 8 位 PWM 输出(PWM1~PWM5), 占空比可调, PWM 周期固定可选 8 种周期, 操作简单。

11.2 寄存器描述

表 11-1 PWMx 相关寄存器列表

名称	访问地址	描述	复位值
R8_PWM_OUT_EN	0x40005000	PWMx 输出使能寄存器	0x00
R8_PWM_POLAR	0x40005001	PWMx 输出极性配置寄存器	0x00
R8_PWM_CONFIG	0x40005002	PWMx 配置控制寄存器	0x00
R8_PWM_DMA_CTRL	0x40005003	PWMx DMA 控制寄存器	0x00
R32_PWM1_3_DATA	0x40005004	PWM 数据保持寄存器 1	0xFFFFFFFF
R16_PWM1_DATA	0x40005004	PWM1 数据保持寄存器 (16 位宽)	0xFFFF
R8_PWM1_DATA	0x40005004	PWM1 数据保持寄存器 (8 位宽)	0xFF
R8_PWM2_DATA	0x40005005	PWM2 数据保持寄存器 (8 位宽)	0xFF
R16_PWM2_DATA	0x40005006	PWM2 数据保持寄存器 (16 位宽)	0xFFFF
R8_PWM3_DATA	0x40005006	PWM3 数据保持寄存器 (8 位宽)	0xFF
R16_PWM3_DATA	0x40005008	PWM3 数据保持寄存器 2	0xFFFF
R32_PWM4_5_DATA	0x40005010	PWM 数据保持寄存器 3	0xFFFFFFFF
R16_PWM4_DATA	0x40005010	PWM4 数据保持寄存器 (16 位宽)	0xFFFF
R8_PWM4_DATA	0x40005010	PWM4 数据保持寄存器 (8 位宽)	0xFF
R8_PWM5_DATA	0x40005011	PWM5 数据保持寄存器 (8 位宽)	0xFF
R16_PWM5_DATA	0x40005012	PWM5 数据保持寄存器 (16 位宽)	0xFFFF
R8_PWM_INT_EN	0x4000500C	PWMx 中断使能寄存器	0x00
R8_PWM_INT_FLAG	0x4000500D	PWMx 中断标志寄存器	0x00
R16_PWM_CYC_VALUE	0x40005014	PWM1、2、3 周期数结束寄存器	0xFFFF
R16_PWM_CYC1_VALUE	0x40005015	PWM4、5 周期数结束寄存器	0xFFFF
R16_PWM_CLOCK_DIV	0x40005018	PWMx 时钟分频寄存器	0x0000
R32_PWM_DMA_NOW	0x4000501C	PWMx DMA 缓冲区当前地址寄存器	0xFFFFFFFF
R32_PWM_DMA_BEG	0x40005020	PWMx DMA 缓冲区起始地址寄存器	0xFFFFFFFF
R32_PWM_DMA_END	0x40005024	PWMx DMA 缓冲区结束地址寄存器	0xFFFFFFFF

PWMx 输出使能寄存器 (R8_PWM_OUT_EN)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
4	RB_PWM5_OUT_EN	RW	PWM5 输出使能位: 1: 使能; 0: 禁止。	0
3	RB_PWM4_OUT_EN	RW	PWM4 输出使能位: 1: 使能; 0: 禁止。	0
2	RB_PWM3_OUT_EN	RW	PWM3 输出使能位: 1: 使能; 0: 禁止。	0

1	RB_PWM2_OUT_EN	RW	PWM2 输出使能位： 1: 使能； 0: 禁止。	0
0	RB_PWM1_OUT_EN	RW	PWM1 输出使能位： 1: 使能； 0: 禁止。	0

PWMx 输出极性配置寄存器 (R8_PWM_POLAR)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
4	RB_PWM5_POLAR	RW	PWM5 输出极性控制位： 1: 默认高电平, 低有效； 0: 默认低电平, 高有效。	0
3	RB_PWM4_POLAR	RW	PWM4 输出极性控制位： 1: 默认高电平, 低有效； 0: 默认低电平, 高有效。	0
2	RB_PWM3_POLAR	RW	PWM3 输出极性控制位： 1: 默认高电平, 低有效； 0: 默认低电平, 高有效。	0
1	RB_PWM2_POLAR	RW	PWM2 输出极性控制位： 1: 默认高电平, 低有效； 0: 默认低电平, 高有效。	0
0	RB_PWM1_POLAR	RW	PWM1 输出极性控制位： 1: 默认高电平, 低有效； 0: 默认低电平, 高有效。	0

PWMx 配置控制寄存器 (R8_PWM_CONFIG)

位	名称	访问	描述	复位值
7	RB_PWM_SYNC_EN	RW	同步使能允许： 1: 允许同步； 0: 禁止同步。	0
6	RB_PWM_SYNC_START	RW	同步开始： 1: 开始同步； 0: 不做操作。	0
5	Reserved	R0	保留。	0
4	RB_PWM4_5_CH	R0	PWM4/5 通道输出状态： 1: 4 通道输出； 0: 5 通道输出。	0
3	RB_PWM4_5_STAG_EN	RW	PWM4/5 交错输出使能位： 1: 交错输出； 0: 独立输出。	0
[2:1]	RB_PWM_CYC_MOD	RW	PWM 数据宽度选择： 00: 8 位数据宽度； 01: 7 位数据宽度； 10: 6 位数据宽度； 11: 16 位数据宽度。	0
0	RB_PWM_CYCLE_SEL	RW	PWM 周期选择： 1: 8/7/6 位数据宽度对应 255/127/63	0

			个时钟周期； 16 位数据宽度对应时钟周期受 R16_PWM_REG_CYCLE 控制。 0: 8/7/6 位数据宽度对应 256/128/64 个时钟周期； 16 位数据宽度对应时钟周期受 R16_PWM_REG_CYCLE 控制。	
--	--	--	--	--

PWMx DMA 控制寄存器 (R8_PWM_DMA_CTRL)

位	名称	访问	描述	复位值
[7:3]	Reserved	R0	保留。	0
2	RB_DMA_SEL	RW	当 RB_PWM_SYNC_EN = 0 时, DMA 输出通道选择： 1: DMA 选择 1、2、3 通道输出 0: DMA 选择 4、5 通道输出 注: 当 RB_PWM_SYNC_EN = 1 时, RB_DMA_SEL =X, DMA 选择 1、2、3、4、 5 通道输出, RB_PWM_SYNC_START 置 1 开 始输出。	0
1	RB_DMA_ADDR_LOOP	RW	DMA 循环输出使能： 1: DMA 循环输出； 0: DMA 单程输出。	0
0	RB_DMA_ENABLE	RW	DMA 使能 (只有 16 数据宽度时有效)： 1: 使能 DMA； 0: 禁止 DMA。	0

PWM 数据保持寄存器组 1 (R32_PWM1_3_DATA)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	XXh
[23:16]	R8_PWM3_DATA	RW	PWM3 数据保持寄存器 (8 位宽通道 3)。	XXh
[15:8]	R8_PWM2_DATA	RW	PWM2 数据保持寄存器 (8 位宽通道 2)。	XXh
[7:0]	R8_PWM1_DATA	RW	PWM1 数据保持寄存器 (8 位宽通道 1)。	XXh
[31:16]	R16_PWM2_DATA	RW	PWM2 数据保持寄存器 (16 位宽通道 2)。	XXXXh
[15:0]	R16_PWM1_DATA	RW	PWM1 数据保持寄存器 (16 位宽通道 1)。	XXXXh

PWM 数据保持寄存器组 2 (R16_PWM3_DATA)

位	名称	访问	描述	复位值
[15:0]	R16_PWM3_DATA	RW	PWM3 数据保持寄存器 (16 位宽通道 3)。	XXXXh

PWM 数据保持寄存器组 3 (R32_PWM4_5_DATA)

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	XXXXh
[15:8]	R8_PWM5_DATA	RW	PWM5 数据保持寄存器 (8 位宽通道 5)。	XXh
[7:0]	R8_PWM4_DATA	RW	PWM4 数据保持寄存器 (8 位宽通道 4)。	XXh
[31:16]	R16_PWM5_DATA	RW	PWM5 数据保持寄存器 (16 位宽通道 5)。	XXXXh
[15:0]	R16_PWM4_DATA	RW	PWM4 数据保持寄存器 (16 位宽通道 4)。	XXXXh

PWMx 中断使能寄存器 (R8_PWM_INT_EN)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
6	RB_PWM_IE_OVER	RW	FIFO 读空中断使能位： 1: 使能中断； 0: 禁止中断。	0
5	RB_PWM_IE_DMA	RW	DMA 传输结束中断使能位： 1: 使能中断； 0: 禁止中断。	0
4	RB_PWM_IE_FIFO	RW	FIFO 数据小于 4 中断使能位： 1: 使能中断； 0: 禁止中断。	0
3	Reserved	RO	保留。	0
2	RB_PWM1_IE_CYC	RW	PWM4、5 周期结束中断使能位： 1: 使能中断； 0: 禁止中断。	0
1	RB_PWM_CYC_PRE	RW	选择 PWM 周期结束中断时间点： 1: 提前 16 个计数产生中断 (以 8 位数据宽度为例, 计数到 240 时产生中断)； 0: 提前 2 个计数产生中断 (以 8 位数据宽度为例, 计数到 254 时产生中断)。	0
0	RB_PWM_IE_CYC	RW	PWM1、2、3 周期结束中断使能位： 1: 使能中断； 0: 禁止中断。	0

PWMx 中断标志寄存器 (R8_PWM_INT_FLAG)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	000b
4	RB_PWM_IF_OVER	RW1	FIFO 溢出中断标志位, 写 1 清零： 1: FIFO 溢出； 0: FIFO 未溢出。	0
3	RB_PWM_IF_DMA	RW1	DMA 传输结束中断标志位, 写 1 清零： 1: DMA 传输结束； 0: DMA 传输未结束。	0
2	RB_PWM_IF_FIFO	RW1	FIFO 数据小于 4 中断标志位, 写 1 清零： 1: FIFO 数据小于 4； 0: 禁止中断。	0
1	RB_PWM1_IF_CYC	RW1	PWM4、5 周期结束标志位, 写 1 清零： 1: 周期结束； 0: 未结束。	0
0	RB_PWM_IF_CYC	RW1	PWM1、2、3 周期结束标志位, 写 1 清零： 1: 周期结束； 0: 未结束。	0

PWM1、2、3 周期数结束寄存器 (R16_PWM_CYC_VALUE)

位	名称	访问	描述	复位值
[15:0]	R16_PWM_CYC_VALUE	RW	16 数据宽度时的 PWM1、2、3 时钟周期。	XXXXh

PWM4、5 周期数结束寄存器 (R16_PWM_CYC1_VALUE)

位	名称	访问	描述	复位值
[15:0]	R16_PWM_CYC1_VALUE	RW	16 数据宽度时的 PWM4、5 时钟周期。	XXXXh

PWMx 时钟分频寄存器 (R16_PWM_CLOCK_DIV)

位	名称	访问	描述	复位值
[15:0]	R16_PWM_CLOCK_DIV	RW	PWM 基准时钟分频系数。 $F_{pwm} = F_{sys}/R16_PWM_CLOCK_DIV$ 。	0000h

PWMx DMA 缓冲区当前地址 (R32_PWM_DMA_NOW)

位	名称	访问	描述	复位值
[31:0]	R32_PWM_DMA_NOW	RW	DMA 数据缓冲区当前地址。 可用于计算已转换次数, 计算方法: $COUNT = PWM_DMA_NOW - PWM_DMA_BEG$ 。	XXXXXXXXh

PWMx DMA 缓冲区起始地址 (R32_PWM_DMA_BEG)

位	名称	访问	描述	复位值
[31:0]	R32_PWM_DMA_BEG	RW	DMA 数据缓冲区起始地址	XXXXXXXXh

PWMx DMA 缓冲区结束地址 (R32_PWM_DMA_END)

位	名称	访问	描述	复位值
[31:0]	R32_PWM_DMA_END	RW	DMA 数据缓冲区结束地址 (不含)	XXXXXXXXh

11.3 PWM 配置

- (1)、设置寄存器 R16_PWM_CLOCK_DIV, 配置 PWM 的基准时钟频率;
- (2)、设置 PWM 输出极性配置寄存器 R8_PWM_POLAR, 配置对应 PWMx 的输出极性;
- (3)、设置 PWM 配置控制寄存器 R8_PWM_CONFIG, 设置 PWMx 的模式、数据位宽、周期;
- (4)、设置 PWM 输出使能寄存器 R8_PWM_OUT_EN, 开启对应的 PWMx 输出使能;
- (5)、根据需要的占空比计算出数据, 写入对应的数据保持寄存器 R8_PWMx_DATA;
- (6)、设置 PWM1~PWM5 中所需的 PWM 引脚方向为输出, 可选地, 设置相应 I/O 的驱动能力;
- (7)、根据需要更新 R8_PWMx_DATA 中的数据, 更新输出占空比。

时钟周期数 (Ncyc) 计算公式:

$$Ncyc = 2^{n-RB_PWM_CYCLE_SEL} \quad (数据宽度 n = 8/7/6);$$

$$Ncyc = RB_PWM_CYC_VALUE \quad (n = 6)$$

R8_PWMx_DATA 计算公式:

$$R8_PWMx_DATA = PWMx \text{ 占空比} * Ncyc$$

PWMx 输出频率 Fpwmout 计算公式:

$$PWMx \text{ 输出频率 } F_{pwmout} = F_{pwm}/Ncyc = F_{sys}/R16_PWM_CLOCK_DIV/Ncyc$$

注: 如果需要通过 PWM 产生直流信号, 那么可以 PWMx 输出端使用 R/C 等电路滤波, 建议用时间常数远大于 $4/F_{pwmout}$ 的两级 RC, 或时间常数远大于 $100/F_{pwmout}$ 的一级 RC。

11.4 DMA 功能

PWM 模块具有 DMA 功能(仅 16 位数据宽度时可以使用), 在使用 DMA 完成中断时需要注意相关寄存器的配置顺序。

使能 DMA 完成中断的步骤如下：

- (1)、读取 R32_PWM_DMA_NOW，将 R32_PWM_DMA_END 赋值为任一不和 R32_PWM_DMA_NOW 相等的值（例：可以取值为 R32_PWM_DMA_NOW + 0x100）；
- (2)、将 R8_PWM_INT_FLAG 的位 RB_PWM_IF_DMA 写 1 清零；
- (3)、将 R8_PWM_INT_EN 中的位 RB_PWM_IE_DMA 置 1。

非循环模式清除 DMA 完成中断的步骤如下：

- (1)、读取 R32_PWM_DMA_NOW，将 R32_PWM_DMA_END 赋值为任一不和 R32_PWM_DMA_NOW 相等的值（例：可以取值为 R32_PWM_DMA_NOW + 0x100）；
- (2)、将 R8_PWM_INT_FLAG 的位 RB_PWM_IF_DMA_END 写 1 清零。

循环模式清除 DMA 完成中断的步骤如下：

- (1)、在 DMA 完成中断内，先关闭 DMA 使能，R8_PWM_DMA_CTRL 的位 RB_DMA_ENABLE 写 0 关闭；
- (2)、将 R8_PWM_INT_FLAG 的位 RB_PWM_IF_DMA_END 写 1 清零。

第 12 章 两线串行接口 I2C

12.1 I2C 简介

I2C 是一种中低速串行总线，总线上可以连接有多个主机和若干从机。通常 I2C 接口由 2 个引脚组成：串行时钟引脚 SCL、串行数据引脚 SDA。

12.1.1 主要特性

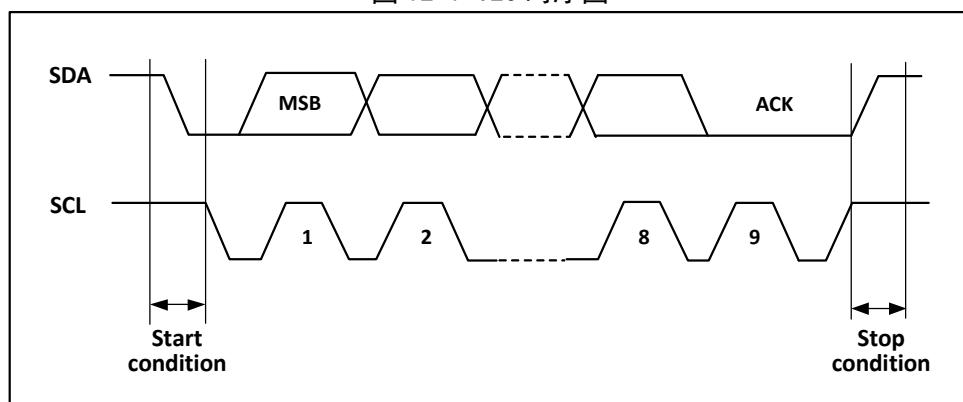
CH572 和 CH570 芯片提供 I2C 接口，特性如下：

- 支持主机模式（Master）和从机模式（Slave），支持多主多从。
- 支持两种速度模式：100KHz 和 400KHz，兼容 I2C 两线串行总线规范。
- 支持 7 位或 10 位地址。
- 从设备支持双 7 位地址。
- 支持总线广播。
- 支持总线仲裁、错误检测、PEC 校验、延长时钟。
- 兼容 SMBus。

12.2 I2C 概述

I2C 是个半双工的总线，它同时只能运行在下列四种模式中之一：主设备发送模式、主设备接收模式、从设备发送模式和从设备接收模式。I2C 模块默认工作在从模式，在产生起始条件后，会自动地切换到主模式，当仲裁丢失或者产生停止信号后，会切换到从模式。I2C 模块支持多主机功能。工作在主模式时，I2C 模块会主动发出数据和地址。数据和地址都以 8 位为单位进行传输，高位在前，低位在后，在起始事件后的是一个字节（7 位地址模式下）或两个字节（10 位地址模式下）地址，主机每发送 8 位数据或地址，从机需要回复一个应答 ACK，即把 SDA 总线拉低，如图 12-1 所示。

图 12-1 I2C 时序图



12.3 主模式

主模式时，I2C 模块主导数据传输并输出时钟信号，数据传输以开始事件开始，以结束事件结束。使用主模式通讯的步骤为：

- 1) 在控制寄存器 2 (R16_I2C_CTRL2) 和时钟控制寄存器 (R16_I2C_CKCFGR) 中设置正确的时钟；
- 2) 在上升沿寄存器 (R16_I2C_RTR) 设置合适的上升沿；
- 3) 在控制寄存器 (R16_I2C_CTRL1) 中置 PE 位启动外设；
- 4) 在控制寄存器 (R16_I2C_CTRL1) 中置 START 位，产生起始事件。

在置 START 位后，I2C 模块会自动切换到主模式，MSL 位会置位，产生起始事件，在产生起始事件后，SB 位会置位，如果 ITEVTEN 位（在 R16_I2C_CTRL2）被置位，则会产生中断。此时应该读取状态寄存器 1 (R16_I2C_STAR1)，写从地址到数据寄存器后，SB 位会自动清除；

- 5) 如果是使用 10 位地址模式，那么写数据寄存器发送头序列（头序列为 11110xx0b，其中的 xx 位

是 10 位地址的最高两位)。

在发送完头序列之后, 状态寄存器的 ADD10 位会被置位, 如果 ITEVTEN 位已经置位, 则会产生中断, 此时应读取 R16_I2C_STAR1 寄存器后, 写第二个地址字节到数据寄存器后, 清除 ADD10 位。

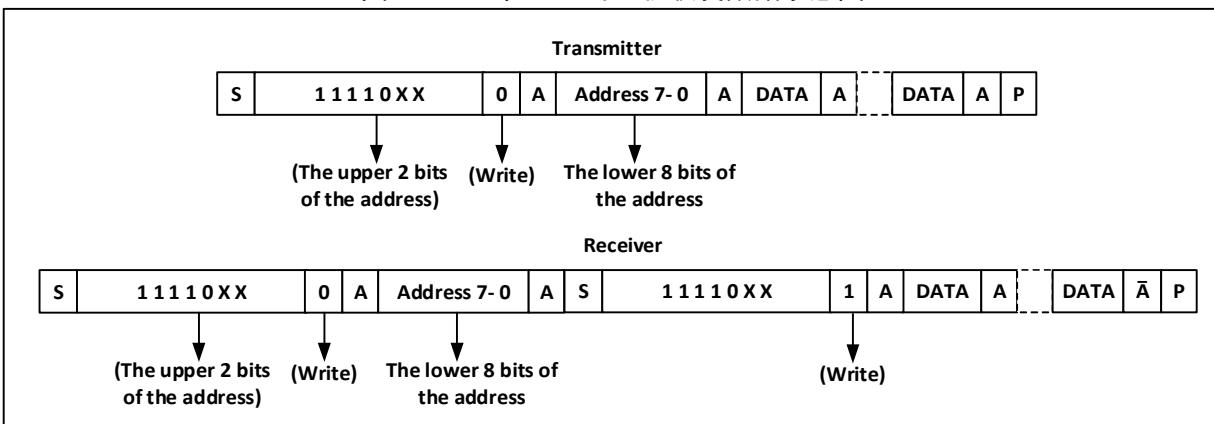
然后写数据寄存器发送第二个地址字节, 在发送完第二个地址字节后, 状态寄存器的 ADDR 位会被置位, 如果 ITEVTEN 位已经置位, 则会产生中断, 此时应读取 R16_I2C_STAR1 寄存器后再读一次 R16_I2C_STAR2 寄存器以清除 ADDR 位;

如果使用的是 7 位地址模式, 那么写数据寄存器发送地址字节, 在发送完地址字节后, 状态寄存器的 ADDR 位会被置位, 如果 ITEVTEN 位已经置位, 则会产生中断, 此时应读取 R16_I2C_STAR1 寄存器后再读一次 R16_I2C_STAR2 寄存器以清除 ADDR 位;

在 7 位地址模式下, 发送的第一个字节为地址字节, 头 7 位代表的是目标从设备地址, 第 8 位决定了后续报文的方向, 0 代表是主设备写入数据到从设备, 1 代表是主设备向从设备读取信息。

在 10 位地址模式下, 如图 12-3 所示, 在发送地址阶段, 第一个字节为 11110xx0, xx 为 10 位地址的最高 2 位, 第二个字节为 10 位地址的低 8 位。若后续进入主设备发送模式, 则继续发送数据; 若后续准备进入主设备接收模式, 则需要重新发送一个起始条件, 跟随发送一个字节为 11110xx1, 然后进入主设备接收模式。

图 12-3 10 位地址时主机收发数据示意图



6) 发送模式时, 主设备内部的移位寄存器将数据从数据寄存器发送到 SDA 线上, 当主设备接收到 ACK 时, 状态寄存器 1 (R16_I2C_STAR1) 的 TxE 被置位, 如果 ITEVTEN 和 ITBUFEN 被置位, 还会产生中断。向数据寄存器写入数据将会清除 TxE 位。

如果 TxE 位被置位且上次发送数据之前没有新的数据被写入数据寄存器, 那么 BTF 位会被置位, 在其被清除之前, SCL 将保持低电平, 读 R16_I2C_STAR1 后, 向数据寄存器写入数据将会清除 BTF 位。

而在接收模式时, I2C 模块会从 SDA 线接收数据, 通过移位寄存器写进数据寄存器。在每个字节之后, 如果 ACK 位被置位, 那么 I2C 模块将会发出一个应答低电平, 同时 RxNE 位会被置位, 如果 ITEVTEN 和 ITBUFEN 被置位, 还会产生中断。如果 RxNE 被置位且在新的数据被接收前, 原有的数据没有被读出, 则 BTF 位将被置位, 在清除 BTF 之前, SCL 将保持低电平, 读取 R16_I2C_STAR1 后, 再读取数据寄存器将会清除 BTF 位。

7) 主设备在结束发送数据时, 会主动发一个结束事件, 即置 STOP 位。在接收模式时, 主设备需要在最后一个数据位的应答位置 NAK。注意, 产生 NAK 后, I2C 模块将会切换至从模式。

12.4 从模式

从模式时, I2C 模块能识别它自己的地址和广播呼叫地址。软件能控制开启或禁止广播呼叫地址的识别。一旦检测到起始事件, I2C 模块将 SDA 的数据通过移位寄存器与自己的地址 (位数取决于 ENDUAL 和 ADDMODE) 或广播地址 (ENGC 置位时) 相比较, 如果不匹配将会忽略, 直到产生新的起始事件。如果与头序列相匹配, 则会产生一个 ACK 信号并等待第二个字节的地址; 如果第二字节的地址也匹配或者 7 位地址情况下全段地址匹配, 那么: 首先产生一个 ACK 应答; ADDR 位被置位, 如果 ITEVTEN 位已经置位, 那么还会产生相应的中断; 如果使用的是双地址模式 (ENDUAL 位被置位), 还需要读取 DUALF 位来判断主机唤起的是哪一个地址。

从模式默认是接收模式，在接收的头序列的最后一位为 1，或者 7 位地址最后一位为 1 时（取决于第一次接收到头序列还是普通的 7 位地址），I2C 模块将进入到发送器模式，TRA 位将指示当前是接收器还是发送器模式。

发送模式时，在清除 ADDR 位后，I2C 模块将字节从数据寄存器通过移位寄存器发送到 SDA 线上。在收到一个应答 ACK 后，TxE 位将被置位，如果设置了 ITEVTEN 和 ITBUFEN，还会产生一个中断。如果 TxE 被置位但在下一个数据发送结束前没有新的数据被写入数据寄存器时，BTF 位将被置位。在清除 BTF 前，SCL 将保持低电平，读取状态寄存器 1 (R16_I2C_STAR1) 后，再向数据寄存器写入数据将会清除 BTF 位。

接收模式时，在 ADDR 被清除后，I2C 模块将 SDA 上的数据通过移位寄存器存进数据寄存器，在每接收到一个字节后，I2C 模块都会置一个 ACK 位，并置 RxNE 位，如果设置了 ITEVTEN 和 ITBUFEN，还会产生一个中断。如果 RxNE 被置位，且在接收到新的数据前旧的数据没有被读出，那么 BTF 会被置位。在清除 BTF 位之前 SCL 会保持低电平。读取状态寄存器 1 (R16_I2C_STAR1) 并读取数据寄存器里的数据会清除 BTF 位。

当 I2C 模块检测到停止事件时，将置 STOPF 位，如果设置了 ITEVFEN 位，还会产生一个中断。用户需要读取状态寄存器 (R16_I2C_STAR1) 再写控制寄存器（比如复位控制字 SWRST）来清除。

12.5 错误

12.5.1 总线错误 BERR

在传输地址或者数据期间，I2C 模块检测到外部的起始或者停止事件时，将产生一个总线错误。产生总线错误时，BERR 位被置位，如果设置了 ITERREN 还会产生一个中断。在从模式下，数据被丢弃，硬件释放总线。如果是起始信号，硬件会认为是重启信号，开始等待地址或停止信号；如果是停止信号，则提前按正常的停止条件操作。在主模式下，硬件不会释放总线，同时不影响当前传输，由用户代码决定是否中止传输。

12.5.2 应答错误 AF

当 I2C 模块检测到一个字节后没有应答时，会产生应答错误。产生应答错误时：AF 会被置位，如果设置了 ITERREN 还会产生一个中断；遇到 AF 错误，如果 I2C 模块工作在从模式，硬件必须释放总线，如果处于主模式，软件必须生成一个停止事件。

12.5.3 仲裁丢失 ARLO

当 I2C 模块检测到仲裁丢失时，产生仲裁丢失错误。产生仲裁丢失错误时：ARLO 位被置位，如果设置了 ITERREN 还会产生一个中断；I2C 模块切换到从模式，并不再响应针对它的从地址发起的传输，除非有主机发起新的起始事件；硬件会释放总线。

12.5.4 过载/欠载错误 OVR

1) 过载错误：

在从机模式下，如果禁止时钟延长，I2C 模块正在接收数据，如果已经接受到一个字节的数据，但是上一次接收到数据还没有被读出，则会产生过载错误。发生过载错误时，最后收到的字节将被丢弃，发送方应当重发最后一次发送的字节。

2) 欠载错误：

在从模式下，如果禁止时钟延长，I2C 模块正在发送数据，如果在下一个字节的时钟到来之前新的数据还没有被写入到数据寄存器，那么将产生欠载错误。在发生欠载错误时，前一次数据寄存器里的数据将被发送两次，如果发生欠载错误，那么接收方应该丢弃重复收到的数据。为了不产生欠载错误，I2C 模块应当在下一个字节的第一个上升沿之前将数据写入数据寄存器。

12.6 时钟延长

如果禁止时钟延长，那么就存在发生过载/欠载错误的可能。但如果使能了时钟延长：

- 1) 在发送模式下, 如果 TxE 置位且 BTF 置位, SCL 将一直为低, 一直等待用户读取状态寄存器, 并向数据寄存器写入待发送的数据;
 - 2) 在接收模式下, 如果 RxNE 置位且 BTF 置位, 那么 SCL 在接收到数据后将保持低, 直到用户读取状态寄存器, 并读取数据寄存器;
- 由此可见, 使能时钟延长可以避免出现过载/欠载错误。

12.7 SMBus

SMBus 也是一种两线接口, 一般应用于在系统和电源管理之间。SMBus 和 I2C 有很多相似的地方, 例如 SMBus 使用和 I2C 一样的 7 位地址模式, 以下是他们的共同点:

- 1) 主从通信模式, 主机提供时钟, 支持多主多从;
- 2) 两线通讯结构;
- 3) 支持 7 位地址格式。

区别:

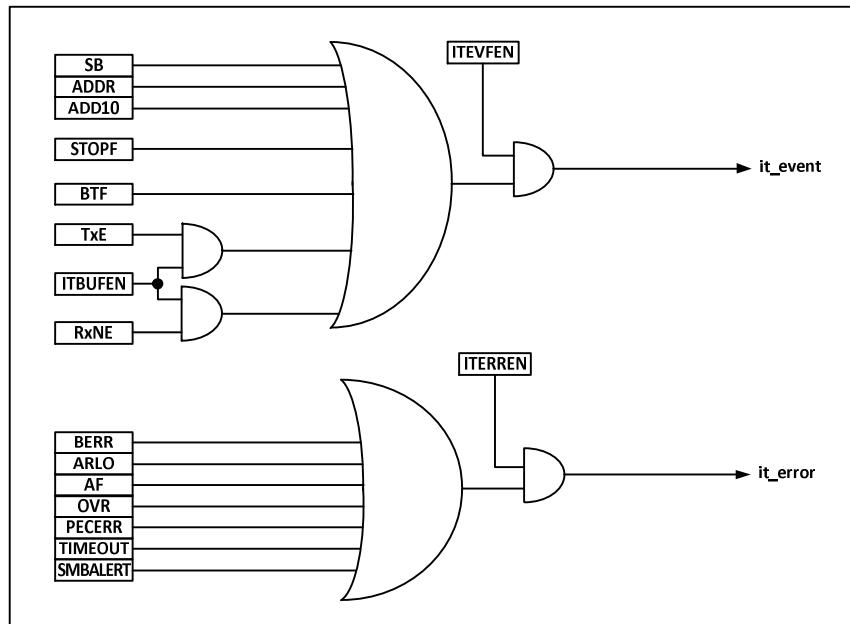
- 1) I2C 支持的速度可达 400KHz, SMBus 速度仅达 100KHz, 且 SMBus 有最小 10KHz 的速度限制;
- 2) SMBus 的时钟为低超过 35mS 时, 会报超时, 但 I2C 无此限制;
- 3) SMBus 有固定的逻辑电平, 而 I2C 没有, 取决于芯片电源电压 VDD;
- 4) SMBus 有总线协议, 而 I2C 没有。

SMBus 还包括设备识别、地址解析协议、唯一的设备标识符、SMBus 提醒和各种总线协议, 具体请参考 SMBus 规范 2.0 版本。当使用 SMBus 时, 只需要置控制寄存器的 SMBus 位, 按需配置 SMBTYPE 位和 ENAARP 位。

12.8 中断

I2C 模块提供事件中断和错误中断, 进入同一个中断服务程序, 经查询后区分处理。

图 12-4 I2C 中断请求



12.9 包校验错误 PEC

包错误校验 (PEC) 是为了提供传输的可靠性而增加一项 CRC8 校验的步骤, 使用以下多项式对每一位串行数据进行计算: $C = X^8 + X^2 + X + 1$ 。

PEC 计算是由控制寄存器的 ENPEC 位激活, 对所有信息字节进行计算, 包括地址和读写位在内。在发送时, 启用 PEC 会在最后一字节数据之后加上一个字节的 CRC8 计算结果; 而在接收模式, 在最

后一字节被认为是 CRC8 校验结果，如果和内部的计算结果不符合，就会回复一个 NAK，如果是主接收器，无论校验结果正确与否，都会回复一个 NAK。

12.10 寄存器描述

表 12-1 I2C 相关寄存器列表

名称	访问地址	描述	复位值
R16_I2C_CTRL1	0x40004800	I2C 控制寄存器 1	0x0000
R16_I2C_CTRL2	0x40004804	I2C 控制寄存器 2	0x0000
R16_I2C_OADDR1	0x40004808	I2C 地址寄存器 1	0x0000
R16_I2C_OADDR2	0x4000480C	I2C 地址寄存器 2	0x0000
R16_I2C_DATAR	0x40004810	I2C 数据寄存器	0x0000
R16_I2C_STAR1	0x40004814	I2C 状态寄存器 1	0x0000
R16_I2C_STAR2	0x40004818	I2C 状态寄存器 2	0x0000
R16_I2C_CKCFGR	0x4000481C	I2C 时钟寄存器	0x0000
R16_I2C_RTR	0x40004820	I2C 上升时间寄存器	0x0002

I2C 控制寄存器 (R16_I2C_CTRL1)

位	名称	访问	描述	复位值
15	RB_I2C_SWRST	RW	软件重置，用户代码置此位会使 I2C 外设重置。在复位前确定 I2C 总线的引脚被释放，总线处于空闲状态。 注：该位可以在总线上没有检测到停止条件但是 busy 位为 1 时，重置 I2C 模块。	0
14	Reserved	R0	保留。	0
13	RB_I2C_ALERT	RW	SMBus 提醒位，用户代码可以设置此位或者清除此位；当 PE 置位后，此位可以被硬件清除。 1：驱动 SMBusALERT 引脚使其变低，响应地址头应紧跟在 ACK 信号后面； 0：释放 SMBusALERT 引脚使其变高，响应地址头应紧跟在 NACK 信号后面。	0
12	RB_I2C_PEC	RW	数据包出错检测使能位，置此位启用数据包出错检测。用户代码可以对此位置位或清零；当 PEC 被传输后，或产生开始或结束信号，或者 PE 位清 0 时，硬件清零该位； 1：带 PEC； 0：不带 PEC。 注：仲裁丢失时，PEC 失效。	0
11	RB_I2C_POS	RW	ACK 和 PEC 位置设置位，该位可以被用户代码置位或者清零，在 PE 被清零后，可以被硬件清除； 1：ACK 位控制在移位寄存器里接收的下一个字节的 ACK 或者 NAK。PEC 移位寄存器里接收的下一字节是 PEC； 0：ACK 位控制当前移位寄存器内正在接受的字节的 ACK 或者 NAK。PEC 位表明当位前移位寄存器的字节是 PEC。 注：POS 位在 2 字节数据接收中的用法如下：必须在接收之前配置好。为了 NACK 第 2 个字	0

			节, 必须在清除 ADDR 位后立刻清除 ACK 位; 为了检测第二个字节的 PEC, 必须在 ADDR 事件发生后, 配置 POS 位后设置 PEC 位。	
10	RB_I2C_ACK	RW	应答使能位, 该位可以被用户代码置位或者清零, 当 PE 位被置位时, 该位可以被硬件清除; 1: 在接收到一个字节后返回一个应答; 0: 不设应答。	0
9	RB_I2C_STOP	RW	停止事件产生位, 该位可以被用户代码置位或清零, 或当检测到停止事件时, 由硬件清除, 或检测到超时错误时, 由硬件将其置位。 主模式下: 1: 在当前字节传输或者当前起始条件发出后产生停止事件; 0: 无停止事件产生。 从模式下: 1: 在当前字节传输后释放 SCL 和 SDA 线; 0: 无停止事件产生。	0
8	RB_I2C_START	RW	起始事件产生位, 该位可以被用户代码置位或者清零, 当起始条件发出后或者 PE 被清零时, 由硬件清零。 主模式下: 1: 重复产生起始事件; 0: 无起始事件产生。 从模式下: 1: 当总线空闲时, 产生起始事件; 0: 无起始事件产生。	0
7	RB_I2C_NOSTRETCH	RW	禁止时钟延长位, 此位用于在 ADDB 或 BTF 标志被置位的情况下, 禁止从模式下的时钟延长, 直至被软件清零。 1: 禁止时钟延长; 0: 允许时钟延长。	0
6	RB_I2C_ENGC	RW	广播呼叫使能位, 置此位使能广播呼叫, 应答广播地址 00h。	0
5	RB_I2C_ENPEC	RW	PEC 使能位, 置此位开启 PEC 计算。	0
4	RB_I2C_ENARP	RW	ARP 使能位, 置此位使能 ARP。 如果 SMBTYPE=0, 则使用 SMBus 设备的默认地址; 如果 SMBTYPE=1, 则使用 SMBus 的主地址。	0
3	RB_I2C_SMBTYPE	RW	SMBus 设备类型, 置 1 为 SMBus 主设备, 置 0 为 SMBus 从设备。	0
2	Reserved	RO	保留。	0
1	RB_I2C_SMBUS	RW	SMBus 模式选择位, 置 1 为使用 SMBus 模式, 置 0 为使用 I2C 模式。	0
0	RB_I2C_PE	RW	I2C 外设使能位。 1: 启用 I2C 模块; 0: 禁用 I2C 模块。	0

I2C 控制寄存器 2 (R16_I2C_CTRL2)

位	名称	访问	描述	复位值
[15:11]	Reserved	RO	保留。	00000b

10	RB_I2C_ITBUFEN	RW	缓冲器中断使能位。 1: 当 TxE 或者 RxEN 被置位时, 产生事件中断; 0: 当 TxE 或者 RxEN 被置位时, 不产生中断。	0
9	RB_I2C_IYTEVTEN	RW	事件中断使能位, 置此位使能事件中断。 在下列条件下, 将产生此中断: SB=1 (主模式); ADDR=1 (主从模式); ADDR10=1 (主模式); STOPF=1 (从模式); BTF=1, 但是没有 TxE 或者 RxEN 事件; 如果 ITBUFEN=1, TxE 事件为 1; 如果 ITBUFEN=1, RxNE 事件为 1。	0
8	RB_I2C_ITERREN	RW	出错中断使能位, 置位表示允许出错中断。 在下列条件下, 将产生该中断; BERR=1; ARL0=1; AF=1; OVR=1; PECERR=1; TIMEOUT=1; SMBAlert=1。	0
[7:6]	Reserved	RO	保留。	00b
[5:0]	RB_I2C_FREQ	RW	I2C 模块时钟频率域, 必须输入正确的时钟频率以产生正确的时序, 允许的范围在 2~36MHz 之间。必须设置在 000010b 到 100100b 之间, 单位为 MHz。 建议: 标准模式下, 输入时钟最低为 2MHz; 快速模式下, 输入时钟最低为 4MHz。	000000b

I2C 地址寄存器 1 (R16_I2C_OADDR1)

位	名称	访问	描述	复位值
15	RB_I2C_ADDMODE	RW	地址模式。 1: 10 位从机地址 (不响应 7 位地址); 0: 7 位从机地址 (不响应 10 位地址)。	0
[14:10]	Reserved	RO	保留。	00000b
[9:8]	RB_I2C_ADD9_8	RW	总线地址, 在使用 10 位地址时为第 9-8 位, 在使用 7 位地址时忽略。	00b
[7:1]	RB_I2C_ADD7_1	RW	总线地址, 第 7-1 位。	0000000b
0	RB_I2C_ADD0	RW	总线地址, 使用 10 位地址时为第 0 位, 在使用 7 位地址时忽略。	0

I2C 地址寄存器 2 (R16_I2C_OADDR2)

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	00h
[7:1]	RB_I2C_ADD2	RW	总线地址, 双地址模式下地址的 7-1 位。	00h
0	RB_I2C_ENDUAL	RW	双地址模式使能位, 置此位可以让 ADD2 也能被识别。	0

I2C 数据寄存器 (R16_I2C_DATAR)

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	00h
[7:0]	RB_I2C_DATAR	RW	数据寄存器, 用来存放接收到的数据或者存放用于发送到总线的数据。	00h

I2C 状态寄存器 1 (R16_I2C_STAR1)

位	名称	访问	描述	复位值
15	RB_I2C_SMBALERT	RW0	SMBus 警示位, 该位可以由用户写 0 复位, 或在 PE 变低时由硬件复位。 在 SMBus 主机模式下: 1: 产生了 SMBus 警示; 0: 无 SMBus 警示。 在 SMBus 从机模式下: 1: 收到 SMBAlert 响应地址头序列直到 SMBAlert 变低; 0: 没有收到 SMBAlert 响应地址头序列。	0
14	RB_I2C_TIMEOUT	RW0	超时或者 Tlow 错误标志位, 该位可以由用户写 0 复位, 或在 PE 变低时由硬件复位。 1: SCL 处于低电平达到 25mS, 或者主机低电平累计时钟扩招时间超过 10mS, 或者从设备低电平累计时间超过 25mS; 0: 无超时错误。 注: 在从模式下此位被置位, 从设备会复位通讯, 硬件会释放总线; 在主模式下此位被置位, 硬件会发出停止条件。	0
13	Reserved	R0	保留。	0
12	RB_I2C_PECERR	RW0	在接收时发生 PEC 错误标志位, 该位可以由用户写 0 复位, 或在 PE 变低时由硬件复位。 1: 有 PEC 错误, 接收到 PEC 后, 返回 NAK; 0: 无 PEC 错误。	0
11	RB_I2C_OVR	RW0	过载、欠载标志位。 1: 有过载、欠载事件发生: 当 NOSTRETCH=1 时, 在接收模式中收到一个新的字节时, 数据寄存器里的内容还未被读出, 则新接收的字节将丢失; 在发送模式时, 没有新的数据写入数据寄存器, 同样的字节将被发送两次; 0: 无过载、欠载事件。	0
10	RB_I2C_AF	RW0	应答失败标志位, 该位可以由用户写 0 复位, 或在 PE 变低时由硬件复位。 1: 应答错误; 0: 应答正常。	0
9	RB_I2C_ARLO	RW0	仲裁丢失标志位, 该位可以由用户写 0 复位, 或在 PE 变低时由硬件复位。 1: 检测到仲裁丢失, 模块失去对总线的控制; 0: 仲裁正常。	0
8	RB_I2C_BERR	RW0	总线出错标志位, 该位可以由用户写 0 复位, 或在 PE 变低时由硬件复位。 1: 起始或者停止条件出错; 0: 正常。	0
7	RB_I2C_TxE	R0	数据寄存器为空标志位, 向数据寄存器写数据可以清除, 或者产生一个起始或者停止位后, 或当 PE 为 0 后, 由硬件自动清除。 1: 发送数据时, 发送数据寄存器为空; 0: 数据寄存器非空。	0

6	RB_I2C_RxNE	RO	数据寄存器非空标志位, 对数据寄存器的读写操作将清除此位, 或者当 PE 为 0 后, 由硬件清除此位。 1: 接收数据时, 数据寄存器不为空; 0: 正常。	0
5	Reserved	RO	保留。	0
4	RB_I2C_STOPF	RO	停止事件标志位, 用户读取状态寄存器 1 之后, 对控制寄存器 1 的写操作将会清除该位, 或者当 PE 为 0 后, 由硬件清除此位。 1: 在应答之后, 从设备在总线上检测到停止事件; 0: 没有检测到停止事件。	0
3	RB_I2C_ADD10	RO	10 位地址头序列发送标志位, 用户读取状态寄存器 1 之后, 对控制寄存器 1 的写操作将会清除该位, 或者当 PE 为 0 后, 由硬件清除此位。 1: 在 10 位地址模式下, 主设备已经将第一个地址字节发送出去; 0: 无。	0
2	RB_I2C_BTF	RO	字节发送结束标志位, 用户读取状态寄存器 1 后, 对数据寄存器的读写将清除此位; 在传输中, 发起一个起始或者停止事件后, 或当 PE 为 0 后, 由硬件清除此位。 1: 字节发送结束。当 NOSTRETCH=0 时: 发送时, 当一个新数据被发送且数据寄存器还未被写入新数据; 接收时, 当接收一个新的字节但是数据寄存器还未被读取; 0: 无。	0
1	RB_I2C_ADDR	RW0	地址被发送/地址匹配标志位, 用户读取状态寄存器 1 后, 对状态寄存器 2 的读操作将会清除此位, 或当 PE 为 0 时, 由硬件清除此位。 主模式: 1: 地址发送结束: 在 10 位地址模式下, 当收到地址的第二个字节的 ACK 后改为被置位; 在 7 位地址模式下, 当收到地址的 ACK 后被置位; 0: 地址发送没有结束。 从模式: 1: 收到的地址匹配; 0: 地址不匹配或者没有收到地址。	0
0	RB_I2C_SB	RO	起始位发送标志位, 读取状态寄存器 1 后写数据寄存器的操作将清除该位, 或当 PE 为 0 时, 硬件将会清除此位。 1: 已发送起始位; 0: 未发送起始位。	0

I2C 状态寄存器 2 (R16_I2C_STAR2)

位	名称	访问	描述	复位值
[15:8]	RB_I2C_PEC	RO	包错误检查域, 当 PEC 使能时 (ENPEC 置位), 此域存放 PEC 的值。	00h
7	RB_I2C_DUALF	RO	匹配检测标志位, 在产生停止位或起始位时, 或者在 PE = 0 时, 硬件会将该位清零。	0

			1: 接收到的地址与 OADDR2 中的内容相符; 0: 接收到的地址与 OADDR1 中的内容相符。	
6	RB_I2C_SMBHOST	RO	SMBus 主机头标志位, 在产生停止位或起始位时, 或者在 PE = 0 时, 硬件会将该位清零。 1: 当 SMBTYPE = 1 且 ENARP = 1 时, 收到了 SMBus 主机地址; 0: 未接收到 SMBus 主机地址。	0
5	RB_I2C_SMBDEFAULT	RO	SMBus 设备默认地址标志位, 在产生停止位或起始位时, 或者在 PE = 0 时, 硬件会将该位清零。 1: 当 ENARP = 1, 收到 SMBus 设备的默认地址; 0: 未收到地址。	0
4	RB_I2C_GENCALL	RO	广播呼叫地址标志位, 在产生停止位或起始位时, 或者在 PE = 0 时, 硬件会将该位清零。 1: 当 ENGC = 1 时, 收到广播呼叫的地址; 0: 未收到广播呼叫地址。	0
3	Reserved	RO	保留。	0
2	RB_I2C_TRA	RO	发送/接收标志位, 在检测到停止事件 (STOPF = 1), 重复的起始条件或者总线仲裁丢失 (ARLO = 1) 或者 PE = 0 时, 硬件会将其清零。 1: 数据已发送; 0: 接收了数据。 该位根据地址字节的 R/W 位来决定。	0
1	RB_I2C_BUSY	RO	总线忙标志位, 该位在检测到一个停止位时会被清零。在接口被禁用时 (PE = 0), 该信息仍被更新。 1: 总线忙: SDA 或 SCL 存在低电平; 0: 总线空闲无通讯。	0
0	RB_I2C_MSL	RO	主从模式指示位, 当接口处于主模式时 (SB = 1), 硬件将该位置位; 当总线检测到一个停止位, 仲裁丢失时, 或者 PE = 0 时, 硬件会清除该位。	0

I2C 时钟寄存器 (R16_I2C_CKCFGR)

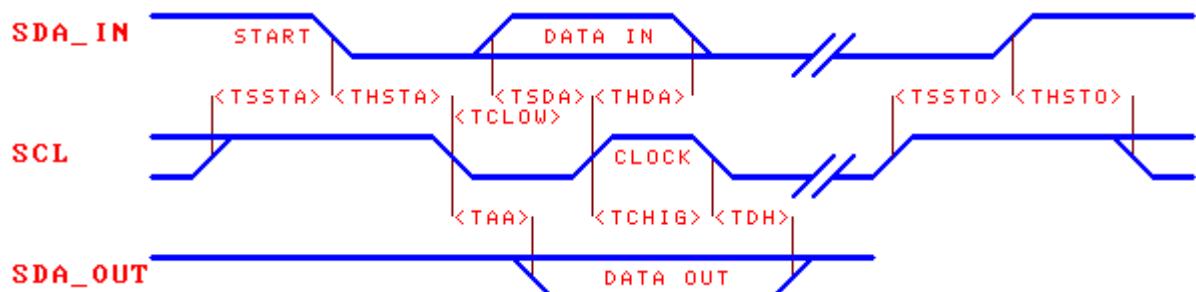
位	名称	访问	描述	复位值
15	RB_I2C_F_S	RW	主模式选择位。 1: 快速模式; 0: 标准模式。	0
14	RB_I2C_DUTY	RW	快速模式时的时钟高电平时间的占空比。 1: 36%; 0: 33.3%。	0
[13:12]	Reserved	RO	保留。	00b
[11:0]	RB_I2C_CCR	RW	时钟分频系数域, 决定 SCL 时钟的频率波形。 在 I2C 标准模式下或 SMBus 模式下: $T_{high} = CCR * T_{HCLK}$ $T_{low} = CCR * T_{HCLK}$ 在 I2C 快速模式下: 如果 DUTY = 0: $T_{high} = CCR * T_{HCLK}$	000h

		$T_{low} = 2 * CCR * T_{HCLK}$ 如果 DUTY = 1: (速度达到 400kHz) $T_{high} = 9 * CCR * T_{HCLK}$ $T_{low} = 16 * CCR * T_{HCLK}$ 例: 在标准模式下, 产生 100kHz 的 SCL 频率: $CCR = FREQR / (2 * 100kHz)$, FREQR = 60MHz 时, $T_{HCLK} = 1/60MHz$, 则 $CCR = 60MHz / (2 * 100kHz) = 300$, CCR 写入 0x12C。	
--	--	--	--

I2C 上升时间寄存器 (R16_I2C_RTR)

位	名称	访问	描述	复位值
[15:6]	Reserved	RO	保留。	000h
[5:0]	RB_I2C_TRISE	RW	最大上升时间域。这个位设置主模式的 SCL 的上升时间。最大的上升沿时间等于 TRISE-1 个时钟周期。此位只能在 PE 清零下设置。	000010b

12.11 I2C 时序



名称	参数说明 (TA = 25°C, VI033 = 3.3V)	系统时钟个数	60M 系统时钟时间 (nS)
TSSTA	SDA 下降沿之前 SCL 高电平的建立时间	5	84
THSTA	SDA 下降沿之后 SCL 高电平的保持时间	5	84
TSDA	SCL 上升沿之前 SDA 数据的建立时间	6	100
THDA	SCL 上升沿之后 SDA 数据的保持时间	>TCHIG	>84
TSSTO	SDA 上升沿之前 SCL 高电平的建立时间	5	84
THSTO	SDA 上升沿之后 SCL 高电平的保持时间	5	84
TCLOW	SCL 时钟的低电平时间	6	100
TCHIG	SCL 时钟的高电平时间	5	84
TAA	SCL 下降沿到 SDA 输出有效	2	34
TDH	SCL 下降沿之后 SDA 输出保持时间	2	34

第 13 章 独立看门狗 IWDG

13.1 IWDG 简介

独立看门狗 (IWDG) 由专用的内部低速时钟 (LSI) 驱动。

13.2 寄存器描述

表 13-1 独立看门狗相关寄存器列表

名称	访问地址	描述	复位值
R32_IWDG_KR	0x40001000	看门狗键寄存器	0xFFFFFFFF
R32_IWDG_CFG	0x40001004	看门狗配置寄存器	0x4FFFFFFF

看门狗键寄存器 (R32_IWDG_KR)

位	名称	访问	描述	复位值
[31:16]	Reserved	WO	保留	XXXXh
[15:0]	IWDG_KR	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, 重装载计数值, 否则, 当计数器为 0 时, 看门狗会产生复位, 写入 0x5555 表示解除保护。 写入 0xCCCC, 启动看门狗工作 (若选择了硬件看门狗则不受此命令字限制)。	XXXXh

看门狗配置寄存器 (R32_IWDG_CFG)

位	名称	访问	描述	复位值
31	IWDG_EN	RO	看门狗启动开关: 1: 开启; 0: 关闭。	0
30	WR_PROTECT	RO	写保护: 1: 禁止操作相应的字段; 0: 解除保护。	1
29	STOP_EN	RW	看门狗停止使能 (存在写保护): 0: 关闭停止开关; 1: 开启停止开关。	0
28	Reserved	RO	保留。	0
[27:16]	COUNT	RO	看门狗递减计数器。	FFFh
15	PVU	RO	配置寄存器更新标志位 (存在写保护): 1: 寄存器更新; 0: 寄存器不更新。	X
[14:12]	PR	RW	预分频因子 (存在写保护) 000: 4 分频; 001: 8 分频; 010: 16 分频; 011: 32 分频; 100: 64 分频; 101: 128 分频; 110: 256 分频; 111: 512 分频。	000b
[11:0]	RLR	RW	RL[11:0]: 看门狗计数器重装载 (具有写保护功能) 用于定义看门狗计数器的重装载值, 每当向 IWDG_KR 寄存器写入 0xAAAA 时, 重装载值会被传送到计数器中。随后计数器从这个值开始递减计数。看门狗超时周期可通过此重装载值和时钟预分频值来计算。	FFFh

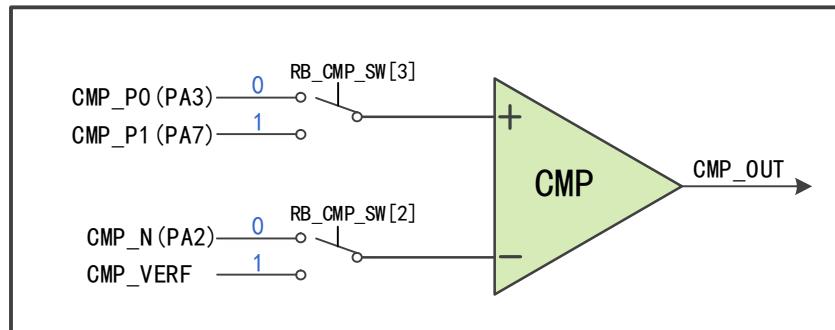
第 14 章 电压比较器 CMP

14.1 CMP 简介

CH572 提供了一个电压比较器 CMP，有两个输入端 P 端和 N 端。可选择其中一个输入端作为参考电压用于比较，当输入 P 端电压小于输入 N 端电压时，比较器输出低电平，反之则输出高电平。其结构框图如下图所示。

电压比较器 CMP 的 N 端可以用二分查找的形式从内置的 16 档参考电压中选择，通过 4 次比较等效为 4 位 ADC 使用。

图 14-1 CMP 结构框图



14.2 寄存器描述

表 14-1 CMP 相关寄存器列表

名称	访问地址	描述	复位值
R32_CMP_CTRL	0x40001054	比较器控制寄存器	0000h

比较器控制寄存器 (R32_CMP_CTRL)

位	名称	访问	描述	复位值
[31:26]	Reserved	R0	保留。	0
25	RB_AP_R_OUT_CMP	RW	比较器实时输出信号。	0
[24:17]	Reserved	R0	保留。	0
16	RB_CMP_IF	RW1	比较器中断标志，写 1 清零。 1: 已产生比较事件； 0: 未产生比较事件。	0
[15:12]	Reserved	R0	保留。	0
[11:10]	RB_CMP_OUT_SEL		比较器中断产生事件选择： 11: 上升沿； 10: 下降沿； 01: 低电平； 00: 高电平。	00
9	Reserved	R0	保留。	0
8	RB_CMP_INT	RW	比较器中断使能： 1: 开启中断； 0: 关闭中断。	0
[7:4]	RB_CMP_NREF_LEVEL	RW	比较器负端参考电平选择： 0000: 50mV； 每档位 50mV；	0

			1110: 750mV; 1111: 800mV。	
[3:2]	RB_CMP_SW	RW	<p>比较器通道选择：</p> <p>Bit[3]: 选择比较器正端输入信号选择 (PA3 或 PA7) :</p> <p>1: PA7; 0: PA3。</p> <p>Bit[2]: 选择比较器负端输入信号选择 (CMP_VERF 或 PA2) :</p> <p>1: CMP_VERF; 0: PA2。</p>	00
1	RB_CMP_CAP	RW	<p>定时器捕获通道信号来源选择：</p> <p>1: 输入来自比较器; 0: 输入来自 GPIO 口。</p>	0
0	RB_CMP_EN	RW	<p>比较器开启使能:</p> <p>1: 使能比较器; 0: 关闭比较器。</p>	0

14.3 CMP 配置

- (1)、配置 RB_CMP_SW 进行通道选择；
- (2)、配置 RB_CMP_NREF_LEVEL 选择参考电压大小；
- (3)、根据需要配置中断，并选择中断产生事件；
- (4)、置位 RB_CMP_EN，比较器开始工作。

第 15 章 按键检测 KEYS defense

15.1 按键检测简介

CH572 芯片提供了按键检测模块 (KEYSCAN)，支持系统睡眠和非睡眠期间的自动扫描，其时钟来源为 LSI。

15.1.1 主要特性

- 支持 5 个 I/O 口，总计 20 个按键，包括 10 个独立按键和 10 个矩阵按键。
- 支持扫描到同一按键编号值达到设定次数中断和单次扫描结束中断。
- 独立按键区每一列支持同时多个按键按下。此时，如果独立按键区有按键按下，则矩阵键盘区的操作将被视为无效。如果独立按键区无按键按下，则按键检测模块支持矩阵键盘区一个或者多个无电气直连属性的按键按下。

15.2 寄存器描述

表 15-1 KEYS defense 相关寄存器列表

名称	访问地址	描述	复位值
R16_KEY_SCAN_CTRL	0x40001064	按键扫描控制寄存器	0x0000
R8_KEY_SCAN_INT_EN	0x40001066	按键扫描中断使能寄存器	0x00
R8_KEY_SCAN_INT_FLAG	0x40001067	按键扫描中断标志寄存器	0x00
R32_KEY_SCAN_NUMB	0x40001068	按键扫描按键编号寄存器	0x00000000

按键扫描控制寄存器 (R16_KEY_SCAN_CTRL)

位	名称	访问	描述	复位值
15	Reserved	R0	保留。	0
14	RB_CLR_WAKEUP_EN	RW	唤醒后自动清除唤醒信号使能： 1: 使能； 0: 禁止。	0
13	RB_SCAN_1END_WAKE_EN	RW	单次扫描后触发唤醒信号使能： 1: 使能； 0: 禁止。	0
[12:8]	RB_PIN_SCAN_EN	RW	参与按键扫描的 I/O 管脚使能， bit[12]~[8]依次为 PA11、PA10、PA8、 PA3、PA2： 1: 参与按键扫描； 0: 不参与按键扫描。	0
[7:4]	RB_SCAN_CLK_DIV	RW	扫描时钟分频，可设 0000 (最低 1 分频) ~1111 (最高 16 分频)。	0
[3:1]	RB_SCAN_CNT_END	RW	设置扫描到相同按键值次数，重复次数为 0 表示不工作，最大可设 7。	0
0	RB_SCAN_START_EN	RW	开始扫描使能： 1: 使能； 0: 禁止。	0

按键扫描中断使能寄存器 (R8_KEY_SCAN_INT_EN)

位	名称	访问	描述	复位值
[7:2]	Reserved	R0	保留。	0

1	RB_SCAN_1END_IE	RW	单次扫描结束后中断使能。 1: 开启中断; 0: 关闭中断。	0
0	RB_KEY_PRESSED_IE	RW	检测到按键按下, 且相同按键值次数达到设定值 (RB_SCAN_CNT_END) 中断使能。 1: 开启中断; 0: 关闭中断。	0

按键扫描中断标志寄存器 (R8_KEY_SCAN_INT_FLAG)

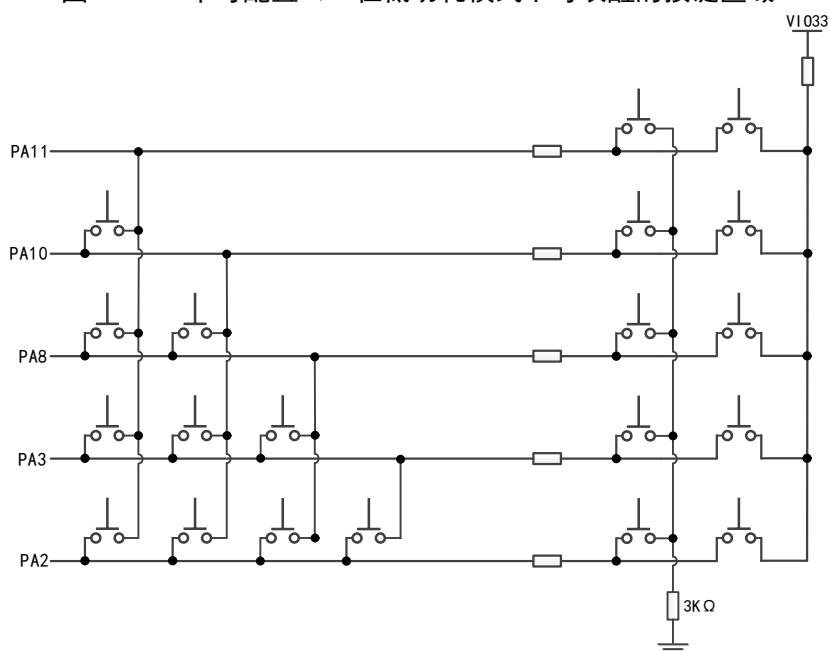
位	名称	访问	描述	复位值
[7:2]	Reserved	R0	保留。	0
1	RB_SCAN_1END_IF	RW1	单词扫描结束后中断标志, 写 1 清 0 (需开启 RB_SCAN_1END_WAKE_EN)。 1: 单次扫描结束; 0: IDLE。	0
0	RB_KEY_PRESSED_IF	RW1	检测到按键按下, 且相同按键值次数达到设定值 (RB_SCAN_CNT_END) 中断标志, 写 1 清 0(RB_KEY_SCAN_NUMB 为 0 不触发)。 1: 相同按键值次数达到设定值; 0: IDLE。	0

按键扫描按键编号寄存器 (R32_KEY_SCAN_NUMB)

位	名称	访问	描述	复位值
[31:23]	Reserved	R0	保留。	0
[22:20]	RB_KEY_SCAN_CNT	R0	当前按键扫描次数。	0
[19:0]	RB_KEY_SCAN_NUMB	R0	当前按键编号值, 编号对应 bit 位为 1 时, 表示检测到当前编号的按键被按下。	0

15.3 KEYS defense 功能描述

图 15-1 5 个可配置 I/O 在低功耗模式下可唤醒的按键区域



使能 KEYS defense 功能后, PA2/PA3/PA8/PA10/PA11 共 5 个 I/O 可被用于按键扫描, 这些 I/O 可分别配置是否参与按键扫描。最多支持 20 个按键, 包括 10 个矩阵键盘区按键和 10 个独立键盘区按键。

表 15-2 KEYS defense 按键信息表

寄存器	位	按键编号							
R32_KEY_SCAN_NUMB	[19:16]					PA2 PA3 (PD)	PA3 PA8 (PD)	PA2 PA8 (PD)	PA8 PA10 (PD)
	[15:8]	PA3 PA10 (PD)	PA2 PA10 (PD)	PA10 PA11 (PD)	PA8 PA11 (PD)	PA3 PA11 (PD)	PA2 PA11 (PD)	PA11 (PD)	PA10 (PD)
	[7:0]	PA8 (PD)	PA3 (PD)	PA2 (PD)	PA11 (PU)	PA10 (PU)	PA8 (PU)	PA3 (PU)	PA2 (PU)

注: 1、PD = 内接下拉电阻; PU = 内接上拉电阻。

2、表中两行 GPIO 表示两个 GPIO 口相连。

按键扫描模块的工作步骤如下:

- (1)、先检测独立按键的上拉按键输入;
- (2)、再检测独立按键的下拉按键输入;
- (3)、最后检测矩阵按键输入;
- (4)、若设置单次扫描后触发唤醒信号使能, 此时触发唤醒信号, 否则等待检测到相同值次数达到设定值, 触发唤醒信号。

15.4 KEYS defense 配置

- (1)、设置 RB_SCAN_CLK_DIV, 配置采样时钟分频系数, 最高 16 分频;
- (2)、设置 RB_PIN_SCAN_EN, 配置参与按键扫描的 I/O 口;
- (3)、设置 RB_SCAN_CNT_END, 配置采样相同值次数终值, 若开启 RB_KEY_PRESSED_IE, 则在达到采样相同值次数终值时, 触发中断;
- (4)、若使能 RB_SCAN_1END_IE, 则在单次扫描结束后, 无论是否检测到按键按下, 都会触发中断;
- (5)、使能 RB_SCAN_START_EN, 并开启按键扫描唤醒信号, 若检测到按键按下, 并连续多次相同值, 达到设定值会触发唤醒信号, 若开启 RB_SCAN_1END_WAKE_EN, 则会在单次扫描结束后即触发唤醒信号; 若开启 RB_CLR_WAKEUP_EN, 则在系统唤醒后清零唤醒信号。

第 16 章 USB 控制器

16.1 USB 控制器简介

芯片内嵌 USBFS 主从控制器及收发器，特性如下：

- 支持 USB Host 主机功能和 USB Device 设备功能。
- 支持 USB2.0 全速 12Mbps 或者低速 1.5Mbps。
- 支持 USB 控制传输、批量传输、中断传输、同步/实时传输。
- 支持最大 64 字节的数据包，内置 FIFO，支持中断和 DMA。
- 可选单线 USB 模式，支持单线 USB 通讯。

16.2 寄存器描述

全速 USB 控制器的基地址是 0x40008000，USB 相关寄存器分为 3 个部分，部分寄存器是在主机和设备模式下进行复用的。

- (1)、USB 全局寄存器
- (2)、USB 设备控制寄存器
- (3)、USB 主机控制寄存器

16.2.1 全局寄存器描述

表 16-1 USB 相关寄存器列表（标灰受 RB_UC_RESET_SIE 复位控制）

名称	访问地址	描述	复位值
R8_USB_CTRL	0x40008000	USB 控制寄存器	0x06
R8_USB_INT_EN	0x40008002	USB 中断使能寄存器	0x00
R8_USB_DEV_AD	0x40008003	USB 设备地址寄存器	0x00
R32_USB_STATUS	0x40008004	USB 状态寄存器	0xXX20XXXX
R8_USB_MIS_ST	0x40008005	USB 杂项状态寄存器	0xXX
R8_USB_INT_FG	0x40008006	USB 中断标志寄存器	0x20
R8_USB_INT_ST	0x40008007	USB 中断状态寄存器	0x3X
R8_USB_RX_LEN	0x40008008	USB 接收长度寄存器	0xXX

USB 控制寄存器 (R8_USB_CTRL)

位	名称	访问	描述	复位值
7	RB_UC_HOST_MODE	RW	USB 工作模式选择位： 1：主机模式 (HOST)； 0：设备模式 (DEVICE)。	0
6	RB_UC_LOW_SPEED	RW	USB 总线信号传输速率选择位： 1：1.5Mbps； 0：12Mbps。	0
5	RB_UC_DEV_PU_EN	RW	USB 设备模式下，USB 设备使能和内部上拉电阻控制位，为 1 则使能 USB 设备传输并且启用内部上拉电阻。 RB_UDP_PU_EN 可代替此位。	0
[5:4]	MASK_UC_SYS_CTRL	RW	见下表配置 USB 系统。	0
3	RB_UC_INT_BUSY	RW	USB 传输完成中断标志未清零前自动暂停使能位： 1：在中断标志 UIF_TRANSFER 未清零前自动暂停，设备模式下自动应答忙 NAK， 主机模式下自动暂停后续传输；	0

			0: 不暂停。	
2	RB_UC_RESET_SIE	RW	USB 协议处理器软件复位控制位: 1: 强制复位 USB 协议处理器 (SIE)，需要软件清零； 0: 不复位。	1
1	RB_UC_CLR_ALL	RW	USB 的 FIFO 和中断标志清零： 1: 强制清空和清零； 0: 不清。	1
0	RB_UC_DMA_EN	RW	USB 的 DMA 和 DMA 中断控制位： 1: 使能 DMA 功能和 DMA 中断； 0: 关闭 DMA。	0

由 RB_UC_HOST_MODE 和 MASK_UC_SYS_CTRL 组成 USB 系统控制组合：

RB_UC_HOST_MODE	MASK_UC_SYS_CTRL	USB 系统控制描述
0	00	禁止 USB 设备功能，关闭内部上拉电阻。 注：如果 RB_UDP_PU_EN=1 则强制启用 DP 上拉电阻。
0	01	使能 USB 设备功能，关闭内部上拉电阻，需加外部上拉。 注：如果 RB_UDP_PU_EN=1 则强制启用 DP 上拉电阻。
0	1x	使能 USB 设备功能，启用内部 1.5K 上拉电阻。该上拉电阻优先于下拉电阻，也可用于 GPIO 模式。
1	00	USB 主机模式，正常工作状态。
1	01	USB 主机模式，强制 DP/DM 输出 SE0 状态。
1	10	USB 主机模式，强制 DP/DM 输出 J 状态。
1	11	USB 主机模式，强制 DP/DM 输出 K 状态/唤醒。

USB 中断使能寄存器 (R8_USB_INT_EN)

位	名称	访问	描述	复位值
7	Reserved	RW	保留。	0
6	RB_UIE_DEV_NAK	RW	USB 设备模式，接收到 NAK 中断： 1: 使能中断； 0: 禁止中断。	0
5	RB_MOD_1_WIRE	RW	USB 单线模式使能： 1: 开启； 0: 关闭。	0
4	RB_UIE_FIFO_OV	RW	FIFO 溢出中断： 1: 使能中断； 0: 禁止中断。	0
3	RB_UIE_HST_SOF	RW	USB 主机模式，SOF 定时中断： 1: 使能中断； 0: 禁止中断。	0
2	RB_UIE_SUSPEND	RW	USB 总线挂起或唤醒事件中断： 1: 使能中断； 0: 禁止中断。	0
1	RB_UIE_TRANSFER	RW	USB 传输完成中断： 1: 使能中断； 0: 禁止中断。	0
0	RB_UIE_DETECT	RW	USB 主机模式，USB 设备连接或断开事件中断：	0

			1: 使能中断; 0: 禁止中断。	
	RB_UIE_BUS_RST	RW	USB 设备模式, USB 总线复位事件中断: 1: 使能中断; 0: 禁止中断。	0

USB 设备地址寄存器 (R8_USB_DEV_AD)

位	名称	访问	描述	复位值
7	RB_UA_GP_BIT	RW	USB 通用标志位, 用户自定义。	0
[6:0]	MASK_USB_ADDR	RW	主机模式: 当前操作的 USB 设备地址; 设备模式: 该 USB 自身地址。	0000000b

USB 杂项状态寄存器 (R8_USB_MIS_ST)

位	名称	访问	描述	复位值
7	RB_UMS_SOF_PRES	RO	USB 主机模式下 SOF 包预示状态位: 1: 将要发送 SOF 包, 此时如有其它 USB 数据包将被自动延后; 0: 无 SOF 包发送。	X
6	RB_UMS_SOF_ACT	RO	USB 主机模式下 SOF 包传输状态位: 1: 正在发出 SOF 包; 0: 发送完成或者空闲。	X
5	RB_UMS_SIE_FREE	RO	USB 协议处理器的空闲状态位: 1: 协议器空闲; 0: 忙, 正在进行 USB 传输。	1
4	RB_UMS_R_FIFO_RDY	RO	USB 接收 FIFO 数据就绪状态位: 1: 接收 FIFO 非空; 0: 接收 FIFO 为空。	0
3	RB_UMS_BUS_RESET	RO	USB 总线复位状态位: 1: 当前 USB 总线处于复位态; 0: 当前 USB 总线处于非复位态。	X
2	RB_UMS_SUSPEND	RO	USB 挂起状态位: 1: USB 总线处于挂起态, 有一段时间没有 USB 活动; 0: USB 总线处于非挂起态。	0
1	RB_UMS_DM_LEVEL	RO	USB 主机模式下, 设备刚连入 USB 端口是 DM 引脚的电平状态, 用于判断速度: 1: 高电平/低速; 0: 低电平/全速。	0
0	RB_UMS_DEV_ATTACH	RO	USB 主机模式下端口的 USB 设备连接状态位: 1: 端口已经连接 USB 设备; 0: 端口没有 USB 设备连接。	0

USB 中断标志寄存器 (R8_USB_INT_FG)

位	名称	访问	描述	复位值
7	RB_U_IS_NAK	RO	USB 设备模式下, NAK 响应状态位: 1: 当前 USB 传输过程中回应 NAK; 0: 无 NAK 响应。	0

6	RB_U_TOG_OK	RO	当前 USB 传输 DATA0/1 同步标志匹配状态位: 1: 同步; 0: 不同步。	0
5	RB_U_SIE_FREE	RO	USB 协议处理器空闲状态位: 1: USB 空闲; 0: 忙, 正在进行 USB 传输。	1
4	RB_UIF_FIFO_OV	RW	USB FIFO 溢出中断标志位, 写 1 清零: 1: FIFO 溢出触发; 0: 无事件。	0
3	RB_UIF_HST_SOF	RW	USB 主机模式下 SOF 定时中断标志位, 写 1 清零: 1: SOF 包传输完成触发; 0: 无事件。	0
2	RB_UIF_SUSPEND	RW	USB 总线挂起或唤醒事件中断标志位, 写 1 清零: 1: USB 挂起事件或唤醒事件触发; 0: 无事件。	0
1	RB_UIF_TRANSFER	RW	USB 传输完成中断标志位, 写 1 清零: 1: 一个 USB 传输完成触发; 0: 无事件。	0
0	RB_UIF_DETECT	RW	USB 主机模式下 USB 设备连接或断开事件中断标志位, 写 1 清零: 1: 检测到 USB 设备连接或断开触发; 0: 无事件。	0
	RB_UIF_BUS_RST	RW	USB 设备模式下 USB 总线复位事件中断标志位, 写 1 清零: 1: USB 总线复位事件触发; 0: 无事件。	0

USB 中断状态寄存器 (R8_USB_INT_ST)

位	名称	访问	描述	复位值
7	RB_UIS_SETUP_ACT	RO	USB 设备模式下, 该位为 1 表示当前已成功接收到 8 字节的 SETUP 请求包, SETUP 令牌不影响 RB_UIS_TOG_OK 和 MASK_UIS_TOKEN、MASK_UIS_ENDP 及 R8_USB_RX_LEN。	0
6	RB_UIS_TOG_OK	RO	当前 USB 传输 DATA0/1 同步标志匹配状态位, 同 RB_U_TOG_OK: 1: 同步; 0: 不同步。	0
[5:4]	MASK_UIS_TOKEN	RO	设备模式下, 当前 USB 传输事务的令牌 PID 标识。	11b
[3:0]	MASK_UIS_ENDP	RO	设备模式下, 当前 USB 传输事务的端点号。	XXXXb
	MASK_UIS_H_RES	RO	主机模式下, 当前 USB 传输事务的应答 PID 标识, 0000 表示设备无应答或超时; 其它值表示应答 PID。	XXXXb

MASK_UIS_TOKEN 用于 USB 设备模式下标识当前 USB 传输事务的令牌 PID: 00 表示 OUT 包; 10 表

示 IN 包；11 表示空闲；01 未定义。

当 MASK_UIS_TOKEN 非空闲、并且 RB_UIS_SETUP_ACT 也为 1 时，必须先处理前者，处理完前者后清零一次 RB_UIF_TRANSFER 使前者进入空闲状态，再处理后者，最后再清零一次 RB_UIF_TRANSFER。

MASK_UIS_H_RES 仅在主机模式下有效。在主机模式下，若主机发送 OUT/SETUP 令牌包时，则该 PID 是握手包 ACK/NAK/STALL，或者是设备无应答/超时。若主机发送 IN 令牌包，则该 PID 是数据包的 PID (DATA0/DATA1) 或者握手包 PID。

USB 接收长度寄存器 (R8_USB_RX_LEN)

位	名称	访问	描述	复位值
[7:0]	R8_USB_RX_LEN	R0	当前 USB 端点接收的数据字节数。	XXh

16.2.2 设备寄存器描述

在 USB 设备模式下，芯片提供了端点 0、1、2、3、4、5、6、7 共 8 组双向端点，所有端点的最大数据包长度都是 64 字节。

端点 0 是默认端点，支持控制传输，发送和接收共用一个 64 字节数据缓冲区。

端点 1、端点 2、端点 3 各自包括一个发送端点 IN 和一个接收端点 OUT，发送和接收各有一个独立的 64 字节或者双 64 字节数据缓冲区，支持批量传输、中断传输和实时/同步传输。

端点 4、端点 5、端点 6、端点 7 包括一个发送端点 IN 和一个接收端点 OUT，发送和接收各有一个独立的 64 字节数据缓冲区，支持批量传输、中断传输和实时/同步传输。

每组端点都具有一个控制寄存器 R8_UEPn_CTRL 和发送长度寄存器 R8_UEPn_T_LEN

(n=0/1/2/3/4/5/6/7)，用于设定该端点的同步触发位、对 OUT 事务和 IN 事务的响应以及发送数据的长度等。

作为 USB 设备所必要的 USB 总线上拉电阻可以由软件随时设置是否启用，当 USB 控制寄存器 R8_USB_CTRL 中的 RB_UC_DEV_PU_EN 置 1 时，控制器根据 RB_UD_LOW_SPEED 的速度设置，在内部为 USB 总线的 DP/DM 引脚连接上拉电阻，并启用 USB 设备功能。上述控制不能用于睡眠或下电模式，但 R16_PIN_ALTERNATE 中的 RB_UDP_PU_EN 可以不受睡眠模式的影响而强制启用 DP 引脚的上拉电阻。

当检测到 USB 总线复位、USB 总线挂起或唤醒事件，或者当 USB 成功处理完数据发送或者数据接收后，USB 协议处理器都将设置相应的中断标志，如果中断使能打开，还会产生相应的中断请求。应用程序可以直接查询或在 USB 中断服务程序中查询并分析中断标志寄存器 R8_USB_INT_FG，根据 RB_UIF_BUS_RST 和 RB_UIF_SUSPEND 进行相应的处理；并且，如果 RB_UIF_TRANSFER 有效，那么还需要继续分析 USB 中断状态寄存器 R8_USB_INT_ST，根据当前端点号 MASK_UIS_ENDP 和当前事务令牌 PID 标识 MASK_UIS_TOKEN 进行相应的处理。如果事先设定了各个端点的 OUT 事务的同步触发位 RB_UEP_R_TOG，那么可以通过 RB_U_TOG_OK 或者 RB_UIS_TOG_OK 判断当前所接收到的数据包的同步触发位是否与该端点的同步触发位匹配，如果数据同步，则数据有效；如果数据不同步，则数据应该被丢弃。每次处理完 USB 发送或者接收中断后，都应该正确修改相应端点的同步触发位，用于下次所发送的数据包或者下次所接收的数据包是否同步检测；另外，设置 RB_UEP_AUTO_TOG 可以实现在发送成功或者接收成功后自动翻转相应的同步触发位。

各个端点准备发送的数据在各自的缓冲区中，准备发送的数据长度是独立设定在 R8_UEPn_T_LEN 中；各个端点接收到的数据在各自的缓冲区中，但是接收到的数据长度都在 USB 接收长度寄存器 R8_USB_RX_LEN 中，可以在 USB 接收中断时根据当前端点号区分。

表 16-2 USB 设备相关寄存器列表（标灰受 RB_UC_RESET_SIE 复位控制）

名称	访问地址	描述	复位值
R8_UDEV_CTRL	0x40008001	USB 设备物理端口控制寄存器	0x0
R8_UEP4_1_MOD	0x4000800c	端点 1/4 模式控制寄存器	0x0
R8_UEP2_3_MOD	0x4000800d	端点 2/3 模式控制寄存器	0x0
R8_UEP567_MOD	0x4000800e	端点 5/6/7 模式控制寄存器	0x0
R16_UEP0_DMA	0x40008010	端点 0 缓冲区起始地址	0xFFFF
R16_UEP1_DMA	0x40008014	端点 1 缓冲区起始地址	0xFFFF

R16_UEP2_DMA	0x40008018	端点 2 缓冲区起始地址	0xXXXX
R16_UEP3_DMA	0x4000801c	端点 3 缓冲区起始地址	0xXXXX
R8_UEP0_T_LEN	0x40008020	端点 0 发送长度寄存器	0xXX
R8_UEP0_CTRL	0x40008022	端点 0 控制寄存器	0x00
R8_UEP1_T_LEN	0x40008024	端点 1 发送长度寄存器	0xXX
R8_UEP1_CTRL	0x40008026	端点 1 控制寄存器	0x00
R8_UEP2_T_LEN	0x40008028	端点 2 发送长度寄存器	0xXX
R8_UEP2_CTRL	0x4000802a	端点 2 控制寄存器	0x00
R8_UEP3_T_LEN	0x4000802c	端点 3 发送长度寄存器	0xXX
R8_UEP3_CTRL	0x4000802e	端点 3 控制寄存器	0x00
R8_UEP4_T_LEN	0x40008030	端点 4 发送长度寄存器	0xXX
R8_UEP4_CTRL	0x40008032	端点 4 控制寄存器	0x00
R16_UEP5_DMA	0x40008054	端点 5 缓冲区起始地址	0xXXXX
R16_UEP6_DMA	0x40008058	端点 6 缓冲区起始地址	0xXXXX
R16_UEP7_DMA	0x4000805c	端点 7 缓冲区起始地址	0xXXXX
R8_UEP5_T_LEN	0x40008064	端点 5 发送长度寄存器	0xXX
R8_UEP5_CTRL	0x40008066	端点 5 控制寄存器	0x00
R8_UEP6_T_LEN	0x40008068	端点 6 发送长度寄存器	0xXX
R8_UEP6_CTRL	0x4000806a	端点 6 控制寄存器	0x00
R8_UEP7_T_LEN	0x4000806c	端点 7 发送长度寄存器	0xXX
R8_UEP7_CTRL	0x4000806e	端点 7 控制寄存器	0x00
R32_EPX_MODE	0x40008070	端点 8-15 控制寄存器	0x00000000

USB 设备物理端口控制寄存器 (R8_UDEV_CTRL)

位	名称	访问	描述	复位值
7	RB_UD_PD_DIS	RW	USB 设备端口 UD+/UD-引脚内部下拉电阻控制位： 1: 禁用内部下拉； 0: 使能内部下拉。 可用于 GPIO 模式提供下拉电阻。	1
6	Reserved	R0	保留。	0
5	RB_UD_DP_PIN	R0	当前 UD+引脚状态： 1: 高电平； 0: 低电平。	X
4	RB_UD_DM_PIN	R0	当前 UD-引脚状态： 1: 高电平； 0: 低电平。	X
3	Reserved	R0	保留。	0
2	RB_UD_LOW_SPEED	RW	USB 设备物理端口低速模式使能位： 1: 选择 1.5Mbps 低速模式； 0: 选择 12Mbps 全速模式。	0
1	RB_UD_HUB0_RESET	RW	USB HUB0 复位控制	0
0	RB_UD_PORT_EN	RW	USB 设备物理端口使能位： 1: 使能物理端口； 0: 禁用物理端口。	0

端点 1/4 模式控制寄存器 (R8_UEP4_1_MOD)

位	名称	访问	描述	复位值
---	----	----	----	-----

7	RB_UEP1_RX_EN	RW	1: 使能端点 1 接收 (OUT); 0: 禁止端点 1 接收。	0
6	RB_UEP1_TX_EN	RW	1: 使能端点 1 发送 (IN); 0: 禁止端点 1 发送。	0
5	Reserved	RO	保留。	0
4	RB_UEP1_BUF_MOD	RW	端点 1 数据缓冲区模式控制位。	0
3	RB_UEP4_RX_EN	RW	1: 使能端点 4 接收 (OUT); 0: 禁止端点 4 接收。	0
2	RB_UEP4_TX_EN	RW	1: 使能端点 4 发送 (IN); 0: 禁止端点 4 发送。	0
[1:0]	Reserved	RO	保留。	00b

bUEP4_RX_EN 和 bUEP4_TX_EN 组合配置 USB 端点 0 和 4 的数据缓冲区模式, 具体参考下表:

表 16-3 端点 0 和 4 缓冲区模式

bUEP4_RX_EN	bUEP4_TX_EN	描述: 以 UEPO_DMA 为起始地址由低向高排列
0	0	端点 0 单 64 字节收发共用缓冲区 (IN 和 OUT)。
1	0	端点 0 单 64 字节收发共用缓冲区; 端点 4 单 64 字节接收缓冲区 (OUT)。
0	1	端点 0 单 64 字节收发共用缓冲区; 端点 4 单 64 字节发送缓冲区 (IN)。
1	1	端点 0 单 64 字节收发共用缓冲区; 端点 4 单 64 字节接收缓冲区 (OUT); 端点 4 单 64 字节接收缓冲区 (IN)。总共 192 字节排列如下: UEPO_DMA+0 地址: 端点 0 收发共用缓冲区 64 字节起始地址; UEPO_DMA+64 地址: 端点 4 接收缓冲区 64 字节起始地址; UEPO_DMA+128 地址: 端点 4 发送缓冲区 64 字节起始地址。

端点 2/3 模式控制寄存器 (R8_UEP2_3_MOD)

位	名称	访问	描述	复位值
7	RB_UEP3_RX_EN	RW	1: 使能端点 3 接收 (OUT); 0: 禁止端点 3 接收。	0
6	RB_UEP3_TX_EN	RW	1: 使能端点 3 发送 (IN); 0: 禁止端点 3 发送。	0
5	Reserved	RO	保留。	0
4	RB_UEP3_BUF_MOD	RW	端点 3 数据缓冲区模式控制位。	0
3	RB_UEP2_RX_EN	RW	1: 使能端点 2 接收 (OUT); 0: 禁止端点 2 接收。	0
2	RB_UEP2_TX_EN	RW	1: 使能端点 2 发送 (IN); 0: 禁止端点 2 发送。	0
1	Reserved	RO	保留。	0
0	RB_UEP2_BUF_MOD	RW	端点 2 数据缓冲区模式控制位。	0

由 RB_UEPn_RX_EN 和 RB_UEPn_TX_EN 以及 RB_UEPn_BUF_MOD (n=1/2/3) 组合分别配置 USB 端点 1、2、3 的数据缓冲区模式, 具体参考下表。其中, 在双 64 字节缓冲区模式下, USB 数据传输时将根据 RB_UEP_*_TOG=0 选择前 64 字节缓冲区, 根据 RB_UEP_*_TOG=1 选择后 64 字节缓冲区, 设置 RB_UEP_AUTO_TOG=1 可实现自动切换。

表 16-4 端点 n 缓冲区模式 (n=1/2/3)

RB_UEPn_RX_EN	RB_UEPn_TX_EN	RB_UEPn_BUF_MOD	描述: 以 R16_UEPn_DMA 为起始地址由低向高排列
0	0	X	端点被禁用, 未用到 R16_UEPn_DMA 缓冲区。
1	0	0	单 64 字节接收缓冲区 (OUT)。
1	0	1	双 64 字节接收缓冲区 (OUT), 由 RB_UEP_R_TOG 选择。
0	1	0	单 64 字节发送缓冲区 (IN)。

0	1	1	双 64 字节发送缓冲区 (IN), 由 RB_UEP_T_TOG 选择。
1	1	0	单 64 字节接收缓冲区 (OUT), 单 64 字节发送缓冲区 (IN)。
1	1	1	双 64 字节接收缓冲区 (OUT), 通过 RB_UEP_R_TOG 选择, 双 64 字节发送缓冲区 (IN), 通过 RB_UEP_T_TOG 选择。 全部 256 字节排列如下: UEPn_DMA+0 地址: RB_UEP_R_TOG=0 时端点接收地址; UEPn_DMA+64 地址: RB_UEP_R_TOG=1 时端点接收地址; UEPn_DMA+128 地址: RB_UEP_T_TOG=0 时端点发送地址; UEPn_DMA+192 地址: RB_UEP_T_TOG=1 时端点发送地址。

端点 5/6/7 模式控制寄存器 (R8_UEP567_MOD)

位	名称	访问	描述	复位值
[7:6]	Reserved	R0	保留。	00b
5	RB_UEP7_RX_EN	RW	1: 使能端点 7 接收 (OUT); 0: 禁止端点 7 接收。	0
4	RB_UEP7_TX_EN	RW	1: 使能端点 7 发送 (IN); 0: 禁止端点 7 发送。	0
3	RB_UEP6_RX_EN	RW	1: 使能端点 6 接收 (OUT); 0: 禁止端点 6 接收。	0
2	RB_UEP6_TX_EN	RW	1: 使能端点 6 发送 (IN); 0: 禁止端点 6 发送。	0
1	RB_UEP5_RX_EN	RW	1: 使能端点 5 接收 (OUT); 0: 禁止端点 5 接收。	0
0	RB_UEP5_TX_EN	RW	1: 使能端点 5 发送 (IN); 0: 禁止端点 5 发送。	0

由 RB_UEPn_RX_EN 和 RB_UEPn_TX_EN (n=5/6/7) 组合分别配置 USB 端点 5、6、7 的数据缓冲区模式, 具体参考下表。

表 16-5 端点 n 缓冲区模式 (n=5/6/7)

RB_UEPn_RX_EN	RB_UEPn_TX_EN	描述: 以 R16_UEPn_DMA 为起始地址由低向高排列
0	0	端点被禁用, 未用到 R16_UEPn_DMA 缓冲区。
1	0	单 64 字节接收缓冲区 (OUT)。
0	1	单 64 字节发送缓冲区 (IN)。
1	1	单 64 字节接收缓冲区 (OUT), 单 64 字节发送缓冲区 (IN)。

端点 n 缓冲区起始地址 (R16_UEPn_DMA) (n=0/1/2/3/5/6/7)

位	名称	访问	描述	复位值
[15:0]	R16_UEPn_DMA	RW	端点 n 缓冲区起始地址。 地址必须 4 字节对齐。	XXXXh

注: 接收数据的缓冲区的长度 $\geq \min$ (可能收到的最大数据包长度 + 2 字节, 64 字节)。

端点 n 发送长度寄存器 (R8_UEPn_T_LEN) (n=0/1/3/4/5/6/7)

位	名称	访问	描述	复位值
7	Reserved	R0	保留。	0
[6:0]	R8_UEPn_T_LEN	RW	设置 USB 端点 n 准备发送的数据字节数。	XXh

端点 2 发送长度寄存器 (R8_UEP2_T_LEN)

位	名称	访问	描述	复位值

7	R8_UEP2_HOST_PID3	RW	主机模式下的令牌标识 PID 的 bit3。	0
[6:0]	R8_UEP2_T_LEN	RW	设置 USB 端点 n 准备发送的数据字节数。	XXh

端点 n 控制寄存器 (R8_UEPn_CTRL) (n=0/1/2/3/4/5/6/7)

位	名称	访问	描述	复位值
7	RB_UEP_R_TOG	RW	USB 端点 n 的接收器 (处理 OUT 事务) 期望的同步触发位: 1: 期望 DATA1; 0: 期望 DATA0。	0
6	RB_UEP_T_TOG	RW	USB 端点 n 的发送器 (处理 IN 事务) 准备的同步触发位: 1: 发送 DATA1; 0: 发送 DATA0。	0
5	Reserved	RO	保留。	0
4	RB_UEP_AUTO_TOG	RW	同步触发位自动翻转使能控制位: 1: 数据发送或接收成功后自动翻转相应的同步触发位; 0: 不自动翻转, 可以手动切换。 只支持端点 1/2/3/5/6/7。	0
[3:2]	MASK_UEP_R_RES	RW	端点 n 的接收器对 OUT 事务的响应控制: 00: 应答 ACK; 01: 超时/无响应, 用于非端点 0 的实时/同步传输; 10: 应答 NAK 或忙; 11: 应答 STALL 或错误。	00b
[1:0]	MASK_UEP_T_RES	RW	端点 n 的发送器对 IN 事务的响应控制: 00: DATA0/DATA1 数据就绪并期望 ACK; 01: 应答 DATA0/DATA1 并期望无响应, 用于非端点 0 的实时/同步传输; 10: 应答 NAK 或忙; 11: 应答 STALL 或错误。	00b

端点 n 控制寄存器 (R32_EPX_MODE) (n=8/9/10/11/12/13/14/15)

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	00h
[23:17]	RB_EP_T_AF	RW	1: 使能端点 8-15 发送复用; 0: 禁止端点 8-15 发送复用。	00h
16	Reserved	RO	保留。	0
[15:8]	RB_EP_R_EN	RW	1: 使能端点 8-15 接收 (OUT); 0: 禁止端点 8-15 接收。	00h
[7:0]	RB_EP_T_EN	RW	1: 使能端点 8-15 发送 (IN); 0: 禁止端点 8-15 发送。	00h

16.2.3 主机寄存器描述

在 USB 主机模式下, 芯片提供了一组双向主机端点, 包括一个发送端点 OUT 和一个接收端点 IN, 数据包的最大长度是 64 字节, 支持控制传输、中断传输、批量传输和实时/同步传输。

主机端点发起的每一个 USB 事务, 在处理结束后总是自动设置 RB_UIF_TRANSFER 中断标志。应用程序可以直接查询或在 USB 中断服务程序中查询并分析中断标志寄存器 R8_USB_INT_FG, 根据各中断

标志分别进行相应的处理；并且，如果 RB_UIF_TRANSFER 有效，那么还需要继续分析 USB 中断状态寄存器 R8_USB_INT_ST，根据当前 USB 传输事务的应答 PID 标识 MASK_UIS_H_RES 进行相应的处理。

如果事先设定了主机接收端点的 IN 事务的同步触发位 RB_UH_R_TOG，那么可以通过 RB_U_TOG_OK 或者 RB_UIS_TOG_OK 判断当前所接收到的数据包的同步触发位是否与主机接收端点的同步触发位匹配，如果数据同步，则数据有效；如果数据不同步，则数据应该被丢弃。每次处理完 USB 发送或者接收中断后，都应该正确修改相应主机端点的同步触发位，用于同步下次所发送的数据包和检测下次所接收的数据包是否同步；另外，通过设置 RB_UH_T_AUTO_TOG 和 RB_UH_R_AUTO_TOG 可以实现在发送成功或接收成功后自动翻转相应的同步触发位。

USB 主机令牌设置寄存器 R8_UH_EP_PID 用于设置被操作的目标设备的端点号和本次 USB 传输事务的令牌 PID 包标识。SETUP 令牌和 OUT 令牌所对应的数据由主机发送端点提供，准备发送的数据在 R16_UH_TX_DMA 缓冲区中，准备发送的数据长度设置在 R16_UH_TX_LEN 中；IN 令牌所对应数据由目标设备返回给主机接收端点，接收到数据存放 R16_UH_RX_DMA 缓冲区中，接收到的数据长度存放在 R8_USB_RX_LEN 中。

表 16-6 USB 主机相关寄存器列表（标灰受 RB_UC_RESET_SIE 复位控制）

名称	访问地址	描述	复位值
R8_UHOST_CTRL	0x40008001	USB 主机物理端口控制寄存器	0xX0
R8_UH_EP_MOD	0x4000800d	USB 主机端点模式控制寄存器	0x00
R16_UH_RX_DMA	0x40008018	USB 主机接收缓冲区起始地址	0xFFFF
R16_UH_TX_DMA	0x4000801c	USB 主机发送缓冲区起始地址	0xFFFF
R8_UH_SETUP	0x40008026	USB 主机辅助设置寄存器	0x00
R8_UH_EP_PID	0x40008028	USB 主机令牌设置寄存器	0x00
R8_UH_RX_CTRL	0x4000802a	USB 主机接收端点控制寄存器	0x00
R8_UH_TX_LEN	0x4000802c	USB 主机发送长度寄存器	0xXX
R8_UH_TX_CTRL	0x4000802e	USB 主机发送端点控制寄存器	0x00

USB 主机物理端口控制寄存器 (R8_UHOST_CTRL)

位	名称	访问	描述	复位值
7	RB_UH_PD_DIS	RW	USB 主机端口 UD+/UD-引脚内部下拉电阻控制位： 1：禁用内部下拉； 0：使能内部下拉。 可用于 GPIO 模式提供下拉电阻。	1
6	Reserved	R0	保留。	0
5	RB_UH_DP_PIN	R0	当前 UD+引脚状态： 1：高电平； 0：低电平。	X
4	RB_UH_DM_PIN	R0	当前 UD-引脚状态： 1：高电平； 0：低电平。	X
3	Reserved	R0	保留。	0
2	RB_UH_LOW_SPEED	RW	USB 主机端口低速模式使能位： 1：选择 1.5Mbps 低速模式； 0：选择 12Mbps 全速模式。	0
1	RB_UH_BUS_RESET	RW	USB 主机模式总线复位控制位： 1：强制输出 USB 总线复位； 0：结束输出。	0
0	RB_UH_PORT_EN	RW	USB 主机端口使能位： 1：使能主机端口；	0

			0: 禁用主机端口。 当 USB 设备断开连接时, 该为自动清 0。	
--	--	--	---------------------------------------	--

USB 主机端点模式控制寄存器 (R8_UH_EP_MOD)

位	名称	访问	描述	复位值
7	Reserved	R0	保留。	0
6	RB_UH_EP_TX_EN	RW	主机发送端点发送 (SETUP/OUT) 使能位: 1: 使能端点发送; 0: 禁止端点发送。	0
5	Reserved	R0	保留。	0
4	RB_UH_EP_TBUF_MOD	RW	主机发送端点发送数据缓冲区模式控制位。	0
3	RB_UH_EP_RX_EN	RW	主机接收端点接收 (IN) 使能位: 1: 使能端点接收; 0: 禁止端点接收。	0
[2:1]	Reserved	R0	保留。	00b
0	RB_UH_EP_RBUF_MOD	RW	USB 主机接收端点接收数据缓冲区模式控制位。	0

由 RB_UH_EP_TX_EN 和 RB_UH_EP_TBUF_MOD 组合控制主机发送端点数据缓冲区模式, 参考下表。

表 16-7 主机发送缓冲区模式

RB_UH_EP_TX_EN	RB_UH_EP_TBUF_MOD	描述: 以 R16_UH_TX_DMA 为起始地址
0	X	端点被禁用, 未用到 R16_UH_TX_DMA 缓冲区。
1	0	单 64 字节发送缓冲区 (SETUP/OUT)。
1	1	双 64 字节发送缓冲区, 通过 RB_UH_T_TOG 选择: 当 RB_UH_T_TOG=0 时选择前 64 字节缓冲区; 当 RB_UH_T_TOG=1 时选择后 64 字节缓冲区。

由 RB_UH_EP_RX_EN 和 RB_UH_EP_RBUF_MOD 组合控制主机接收端点数据缓冲区模式, 参考下表。

表 16-8 主机接收缓冲区模式

RB_UH_EP_RX_EN	RB_UH_EP_RBUF_MOD	结构描述: 以 R16_UH_RX_DMA 为起始地址
0	X	端点被禁用, 未用到 R16_UH_RX_DMA 缓冲区。
1	0	单 64 字节接收缓冲区 (IN)。
1	1	双 64 字节接收缓冲区, 通过 RB_UH_R_TOG 选择: 当 RB_UH_R_TOG=0 时选择前 64 字节缓冲区; 当 RB_UH_R_TOG=1 时选择后 64 字节缓冲区。

USB 主机接收缓冲区起始地址 (R16_UH_RX_DMA)

位	名称	访问	描述	复位值
[15:0]	R16_UH_RX_DMA	RW	主机端点数据接收缓冲区起始地址。 低 15 位有效, 地址必须 4 字节对齐。	XXXXh

USB 主机发送缓冲区起始地址 (R16_UH_TX_DMA)

位	名称	访问	描述	复位值
[15:0]	R16_UH_TX_DMA	RW	主机端点数据发送缓冲区起始地址。 低 15 位有效, 地址必须 4 字节对齐。	XXXXh

USB 主机辅助设置寄存器 (R8_UH_SETUP)

位	名称	访问	描述	复位值

7	RB_UH_PRE_PID_EN	RW	低速前导包 PRE PID 使能位： 1: 使能, 用于通过外部 HUB 与低速 USB 设备通讯。 0: 禁用低速前导包。	0
6	RB_UH_SOF_EN	RW	自动产生 SOF 包使能位： 1: 主机自动产生 SOF 包； 0: 不自动产生, 但可手工产生。	0
[5:0]	Reserved	RO	保留。	000000b

USB 主机令牌设置寄存器 (R8_UH_EP_PID)

位	名称	访问	描述	复位值
[7:4]	MASK_UH_TOKEN	RW	设置本次 USB 传输事务的令牌 PID 包标识。	0000b
[3:0]	MASK_UH_ENDP	RW	设置本次被操作的目标设备的端点号。	0000b

USB 主机接收端点控制寄存器 (R8_UH_RX_CTRL)

位	名称	访问	描述	复位值
7	RB_UH_R_TOG	RW	USB 主机接收器 (处理 IN 事务) 期望的同步触发位： 1: 期望 DATA1； 0: 期望 DATA0。	0
[6:5]	Reserved	RO	保留。	00b
4	RB_UH_R_AUTO_TOG	RW	同步触发位自动翻转使能控制位： 1: 数据接收成功后自动翻转相应的期待同步触发位 (RB_UH_R_TOG)； 0: 不自动翻转, 可以手动切换。	0
3	Reserved	RO	保留。	0
2	RB_UH_R_RES	RW	主机接收器对 IN 事务的响应控制位： 1: 无响应, 用于非 0 端点的实时/同步传输； 0: 应答 ACK。	0
[1:0]	Reserved	RO	保留。	00b

USB 主机发送长度寄存器 (R8_UH_TX_LEN)

位	名称	访问	描述	复位值
[7:0]	R8_UH_TX_LEN	RW	设置 USB 主机发送端点准备发送的数据字节数。	XXh

USB 主机发送端点控制寄存器 (R8_UH_TX_CTRL)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
6	RB_UH_T_TOG	RW	USB 主机发送器 (处理 SETUP/OUT 事务) 准备的同步触发位： 1: 表示发送 DATA1； 0: 表示发送 DATA0。	0
5	Reserved	RO	保留。	0
4	RB_UH_T_AUTO_TOG	RW	同步触发位自动翻转使能控制位： 1: 数据发送成功后自动翻转相应的同步	0

			触发位 (RB_UH_T_TOG)； 0: 不自动翻转，可以手动切换。	
[3:1]	Reserved	RO	保留。	000b
0	RB_UH_T_RES	RW	USB 主机发送器对 SETUP/OUT 事务的响应控制位： 1: 期望无响应，用于非 0 端点的实时/同步传输； 0: 期望应答 ACK。	0

第 17 章 预留

第 18 章 无线通讯

18.1 简介

芯片集成低功耗 2.4-GHz 无线通讯模块，包括 RF 收发器、基带和链路控制以及天线匹配网络，支持低功耗蓝牙 BLE。支持 GFSK (Gaussian Frequency Shift Keying) 数字调制与解调，数据传输速率可以调节。内部提供数百个寄存器用于调节参数和控制过程及状态，本手册不对寄存器作详细介绍，无线通讯底层操作主要以子程序库提供应用支持。

主要特性：

- 集成 2.4GHz 射频收发器、BaseBand 基带和 LLE 链路控制。
- 支持低功耗蓝牙 BLE，符合 Bluetooth Low Energy 5.0 规范。
- 支持 GFSK 数字调制与解调。
- 支持 2Mbps、1Mbps。
- 支持 2.4G 模式下最高 8KHz 上报率。
- 接收灵敏度 -95dBm。
- 可编程 -25dBm 到 +7.5dBm 发送功率，支持动态调整。
- 单端 RF 接口，简化板级设计。
- 支持 AES-128 硬件加解密。
- 支持 DMA。
- 提供优化的协议栈和应用层 API，支持组网。

18.2 2.4GHz 模块

18.2.1 2.4GHz 模块功能特性

- 支持普通模式和增强模式。
- 地址长度可配置。
- CRC 可配：长度 0~2 字节、多项式和异或操作可配。
- 字节序可配：数据的位序 MSB/LSB 可配。
- 无线速率支持 1Mbps, 2Mbps。
- 接收的调制指数范围：0.3~0.9。

18.2.2 2.4GHz 模块数据帧格式

2.4GHz 模块的数据帧格式支持普通模式和增强模式。当设置为普通模式时，数据长度固定；当设置为增强模式时，由于增加了控制字，所以数据长度可变，且包含 PID 和有无应答标志。

具体的应用请参考提供的 RF 应用示例。

18.3 LLE 模块

LLE 模块支持自动收发模式和手动收发模式，4 组独立的硬件定时器可控制收发数据任意一过程的时间点。

18.4 DMA 模块

控制器有 1 组 DMA，有两个通道。DMA 的两个通道分别用于发送数据和接收数据，在自动发送模式中，可以同时配置发送 DMA 的地址和接收 DMA 的地址，这样在帧间隔期间不需要再进行配置。

18.5 BB 模块

18.6 AES 模块

具体的应用请基于 BLE 协议栈库使用，并参考提供的 BLE 应用示例。

第 19 章 参数

19.1 绝对最大值

临界或者超过绝对最大值将可能导致芯片工作不正常甚至损坏。

表 19-1 绝对最大值参数表

名称	参数说明	最小值	最大值	单位
TA	工作时的环境温度	-40	85	°C
TS	储存时的环境温度	-55	125	°C
V5	内部调压器 LD05V 输入电源电压	-0.4	5.5	V
VDD33	工作电源电压 (VDD33 接电源, GND 接地)	-0.4	4.0	V
VI033	I/O 电源电压 (VI033 接电源, GND 接地)	-0.4	4.0	V
V10	输入或者输出引脚上的电压	-0.4	VI033+0.4	V
VI05	支持 5V 耐压的输入或者输出引脚上的电压	-0.4	5.5	V
VXCK	X1/X0 引脚上的电压	-0.3	1.4	V

19.2 电气参数

测试条件: TA = 25°C, V5 = 5V 或 3.3V, VDD33 = VI033 = 3.3V, Fsys = 16MHz。

表 19-2 电气参数表

名称	参数说明		最小值	典型值	最大值	单位	
V5	内部调压器 LD05V 输入电源电压@V5		4.5	5	5.3	V	
VDD33	工作电源电压@VDD33		2.0	3.3	3.6	V	
	内部调压器 LD05V 输出电压@VDD33		3.2	3.3	3.4	V	
VI033	I/O 电源电压@VI033		2.0	3.3	3.6	V	
ICC ₈	静态电源电流 条件: 代码运行于 RAM 中	Fsys=8M	RAM	3.0		mA	
		Flash		3.3			
ICC ₁₆		Fsys=16M	RAM	3.3		mA	
		Flash		3.4			
ICC ₄₈		Fsys=60M	RAM	5.6		mA	
		Flash		8.6			
VIL	GPIO 低电平输入电压		0		0.9	V	
VIH	GPIO 高电平输入电压		2.0		VI033	V	
VIL5	支持 5V 耐压的 GPIO 低电平输入电压		0		0.9	V	
VIH5	支持 5V 耐压的 GPIO 高电平输入电压		2.0		5.0	V	
VOL	低电平输出电压 (20mA 吸入电流)		0	0.3	0.5	V	
VOH	高电平输出电压 (20mA 输出电流)		VI033-0.5	VI033-0.3	VI033	V	
IIN	GPIO 浮空输入端的输入电流		-3	0	3	uA	
IUP	GPIO 内置上拉电阻的输入端的输入电流		25	60	90	uA	
IDN	GPIO 内置下拉电阻的输入端的输入电流		-90	-60	25	uA	
VIvr	LVR 低压复位的门限电压		1.5	1.7	1.9	V	

19.3 低功耗模式功耗

测试条件: TA = 25°C, V5 = 3.3V 或浮空, VDD33 = VI033 = 3.3V, Fsys = 16MHz。

表 19-3 低功耗参数表 (仅供参考, 与温度相关)

低功耗模式	最小值	典型值	最大值	单位
空闲模式, 开启各模块时钟组合		1.7		mA

暂停模式, 关闭所有时钟	420		uA
睡眠模式, 多种组合, 参考表 5-3	0.46~1.2		uA
睡眠模式, PMU+内核+RAM12K, GPIO 唤醒, 无 RTC	1.2		uA
下电模式, 多种组合, 参考表 5-3	0.3~0.9		uA
下电模式, 仅 PMU, GPIO 唤醒后复位, 无 RTC	0.3		uA
单一 5V 电源供电下的最低电流	1.3		mA

表 19-4 各模块电流 (仅供参考, 与温度相关)

名称	参数说明		最小值	典型值	最大值	单位
$I_{DD(\text{RAM12K})}$	RAM12K: 12KB SRAM			0.3		uA
$I_{DD(\text{LSI})}$	内部 LSI 振荡器			0.3		uA
$I_{DD(\text{HSE})}$	外部 HSE 振荡器		130	170	260	uA
$I_{DD(\text{BM})}$	低功耗的电池低压监控 BM 模块			0.9		uA
$I_{DD(\text{PLL})}$	内部 PLL 振荡器			130		uA
$I_{DD(\text{CMP})}$	CMP 模块			30		uA
$I_{DD(\text{KEYSCAN})}$	KEYSCAN 模块			1.3		uA
$I_{DD(\text{USB})}$	USB 模块	非发送状态		0.6		mA
		发送状态		2.1		mA
$I_{DD(\text{BLE})}$	BLE 蓝牙	接收		7.1		mA
		-25dBm 发送功率		3.7		mA
		0dBm 发送功率		10.6		mA
		+7.5dBm 发送功率		21.5		mA

19.4 时钟源

表 19-5 高速振荡器 HSE

名称	参数说明	最小值	典型值	最大值	单位
F_{HSE}	外部 HSE 振荡器频率		32		MHz
T_{SUHSE}	外部 HSE 振荡器启动到可用时间	80	200	500	uS
T_{STHSE}	外部 HSE 振荡器启动到稳定时间	200	500	3000	uS

表 19-6 低速振荡器 LSI

名称	参数说明	最小值	典型值	最大值	单位
F_{LSI}	内部 LSI 振荡器频率	24		42	KHz
T_{SULSI}	内部 LSI 振荡器启动到稳定时间		40	100	uS

表 19-7 PLL 特性

名称	参数说明	最小值	典型值	最大值	单位
F_{PLL}	PLL 倍频输出时钟 (CK32M * 18.75 倍)		600		MHz
T_{PLLLK}	PLL 锁相时间		20	40	uS

19.5 时间参数

测试条件: $TA = 25^\circ\text{C}$, $V5 = 5\text{V}$ 或 3.3V , $VDD33 = VI033 = 3.3\text{V}$, $F_{sys} = 6.4\text{MHz}$ 。

表 19-8 时间参数

名称	参数说明		最小值	典型值	最大值	单位
T_{rpor}	电源上电复位 RPOR 后的复位延时		12	13	14.5	μs
T_{V5R}	V5 电源电压上升速率		1		10000	us/V
T_{rst}	RST 有效信号宽度			100		nS
T_{mr}	外部复位 MR 后的复位延时		2	8	18	μs
T_{sr}	软件复位 SR 后的复位延时		2	8	18	μs
T_{wtr}	看门狗复位 WTR 后的复位延时		10	12	18	μs
T_{WAK}	从低功耗状态退出的唤醒时间	空闲模式	0.6	1	3	μs
		暂停模式	$T_{SUHSE}+1$	$T_{SUHSE}+80$	$T_{SUHSE}+150$	μs
		睡眠模式	$T_{SUHSE}+1$	$T_{SUHSE}+300$	$T_{SUHSE}+400$	μs
		下电模式	$T_{SUHSE}+0.4$	$T_{SUHSE}+1$	$T_{SUHSE}+5$	mS

注: 上表中延时参数均是基于 T_{sys} 的倍数, 降低主频将增加延时。

上表中延时参数是基于使用外部 HSE 时钟源, 如果睡眠期间使用外部 HSE 时钟源, 那么表中暂停模式/睡眠模式/下电模式的延时参数 T_{WAK} 均额外增加约 $0.2\sim1\text{mS}$ (启动到可用 T_{SUHSE})。

19.6 其它参数

测试条件: $TA = 25^\circ\text{C}$, $V5 = 5\text{V}$ 或 3.3V , $VDD33 = VI033 = 3.3\text{V}$, $F_{sys} = 16\text{MHz}$ 。

表 19-9 其它参数

名称	参数说明		最小值	典型值	最大值	单位
T_{FRER}	Flash-ROM 的单次扇区擦除操作时间		1	3	8	mS
T_{FRPG}	Flash-ROM 的单次字编程操作时间		0.1	1	2	mS
N_{EPCE}	Flash-ROM 的擦写次数 erase/program cycle endurance	5~45°C	100K	2000K (抽测)		times
		-40~85°C	50K	500K (抽测)		
T_{DR}	Flash-ROM 的数据保持能力		20			years
V_{ESD}	I/O 输入或者输出	天线 ANT	2K	4K (抽测)		V
	引脚上的 ESD 耐压	I/O 引脚: PA	4K	6K (抽测)		V

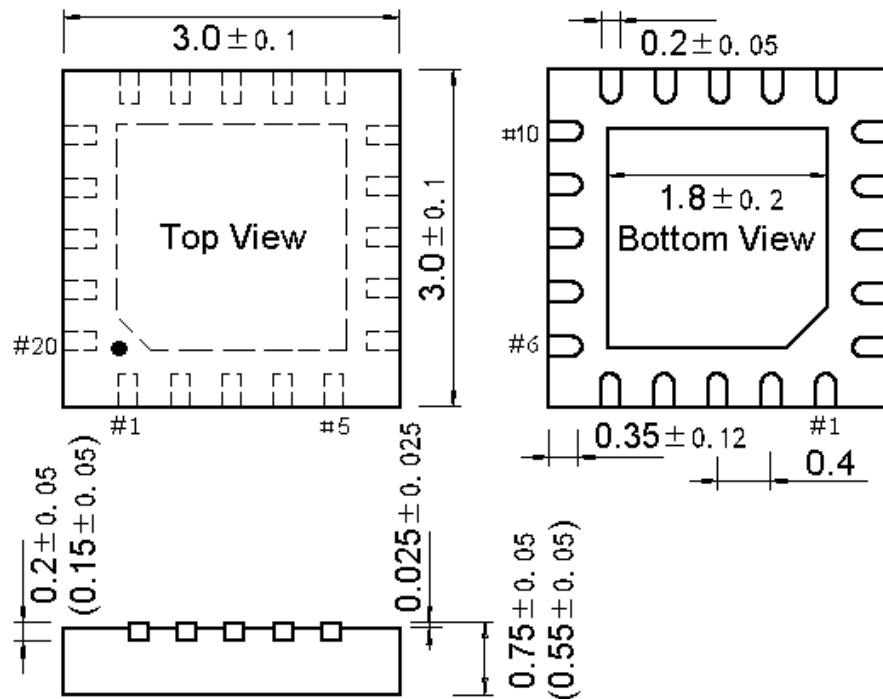
第 20 章 封装

芯片封装

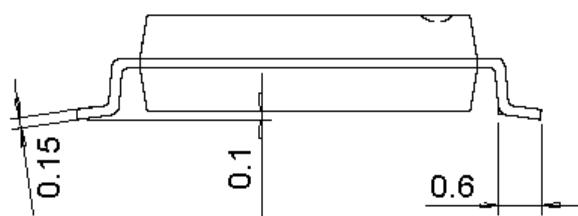
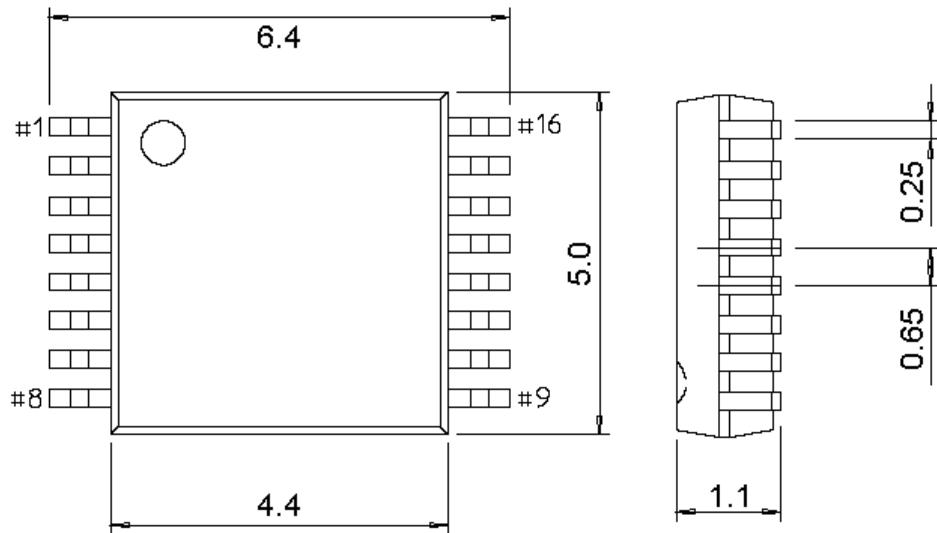
封装形式	塑体尺寸	引脚节距		封装说明	订货型号
QFN20	3*3mm	0. 4mm	15. 7mil	四边无引线 20 脚	CH572D
TSSOP16	4. 4*5. 0mm	0. 65mm	25. 6mil	薄小型的 16 脚贴片	CH572R
DFN10X3	3*3mm	0. 5mm	19. 7mil	双边无引线 10 脚	CH572Q
QFN20	3*3mm	0. 4mm	15. 7mil	四边无引线 20 脚	CH570D
DFN10X3	3*3mm	0. 5mm	19. 7mil	双边无引线 10 脚	CH570Q
SOP8	3. 9*5. 0mm	1. 27mm	50mil	标准 8 脚贴片	CH570E

说明：尺寸标注的单位是 mm (毫米)，引脚中心间距是标称值，除此之外的尺寸误差不大于 $\pm 0.2\text{mm}$ 。

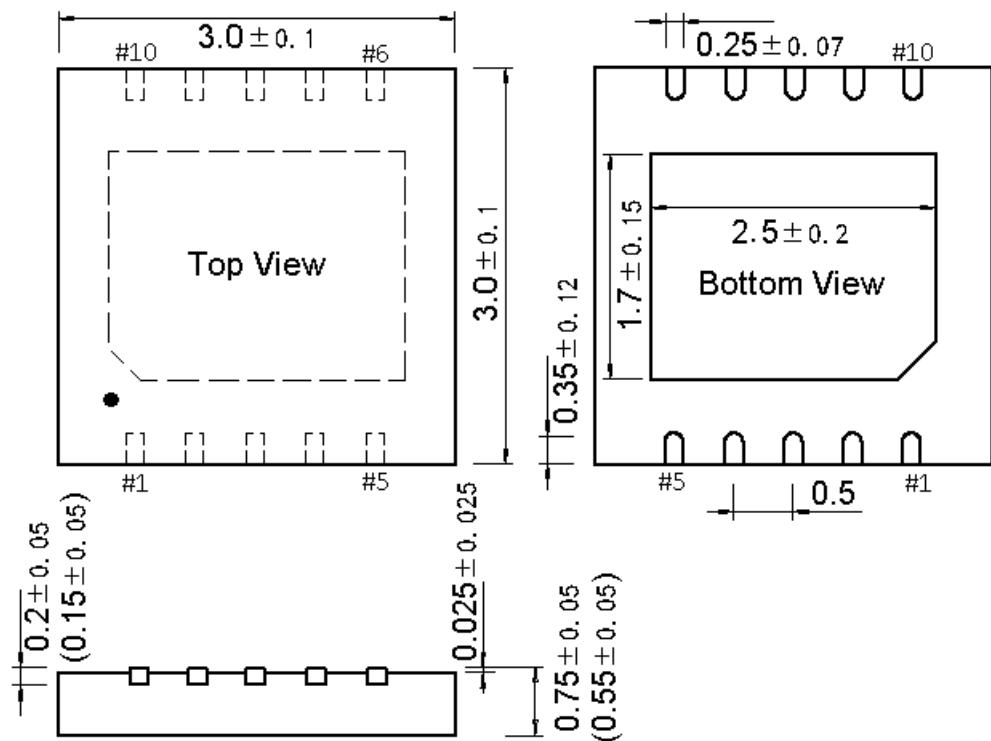
QFN20



TSSOP16



DFN10X3



SOP8

