

概述

CH596 是集成 BLE 无线通讯的 RISC-V MCU 微控制器。片上集成 2Mbps 低功耗蓝牙 BLE 通讯模块、SPI、2 个串口、8 路 ADC、触摸按键检测模块等丰富的外设资源。CH596 支持单节 1.5V 电池供电。

功能

● 内核 Core:

- 青稞 32 位 RISC-V3C 内核
- 支持 RV32IMBC-X 指令集
- 快速可编程中断控制器+硬件中断堆栈
- 分支预测、冲突处理机制
- 单周期乘法、硬件除法
- 多档系统主频，最低 32KHz

● 非易失存储 FlashROM:

- 240KB 用户应用程序存储区 CodeFlash
- 16KB 系统引导程序存储区 BootLoader
- 256B 用户自定义信息存储区
- 256B 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
- 80MHz 系统主频下几乎零等待

● 32K 字节易失数据存储 SRAM:

- 24KB 双电源供电的睡眠保持存储区 RAM24K
- 8KB 双电源供电的睡眠保持存储区 RAM8K

● 电源管理和低功耗:

- 支持 1.2V~1.8V 电源供电
- 空闲模式 Idle: 1.5mA
- 暂停模式 Halt: 1.02mA (PLL/HSE 不停止)
180uA (PLL/HSE 停止)
- 睡眠模式 Sleep: 1.8uA~3.8uA 多档
- 下电模式 Shutdown: 0.85uA~2.6uA 多档
- 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

● 低功耗蓝牙 BLE:

- 集成 2.4GHz RF 收发器和基带及链路控制
- 接收灵敏度-95dBm, 可编程+4dBm 发送功率
- BLE 符合 Bluetooth Low Energy 5.4 规范
- 支持 2Mbps、1Mbps 速率
- 提供优化的协议栈和应用层 API, 支持组网

● 模数转换 ADC:

- 12 位模数转换器, 支持单端输入
- 8 路外部模拟信号通道和 1 路内部信号

● 实时时钟 RTC: 支持定时和触发两种模式

● 触摸按键检测模块 TouchKey: 8 路通道

● 定时器 Timer 和脉宽调制 PWM:

- 3 组 26 位定时器, 16MHz 主频定时可达 4.2S
- 3 路捕捉/采样, 支持上升沿/下降沿/双边沿
- 3 路 26 位 PWM 输出
- 4 路 8 位 PWM 输出, 前 3 路支持 16 位 PWM 输出

● 编码器 ENC: 2 路通道, 用于捕捉旋转编码器信号进行编码

● 异步串口 UART:

- 2 组独立 UART, 内置 8 级 FIFO
- 23 位计数器, 通讯波特率可达 6Mbps
- UART0 支持部分 Modem
- UART0 支持多机通讯时从机地址自动匹配

● 串行外设接口 SPI:

- 支持 Master 和 Slave 模式
- 内置 FIFO, 支持 DMA
- SCK 串行时钟频率可达系统主频的一半

● LED 点阵屏接口: 支持 1/2/4 路数据线

● 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

● 时钟: 内置 PLL、16MHz 和 32KHz 时钟

● 通用输入输出端口 GPIO:

- 15 个 GPIO, 其中 4 个可承受 5V 信号输入
- 可选上拉或下拉电阻, 可选输出驱动能力
- 15 个 GPIO 支持电平或边沿中断、唤醒输入

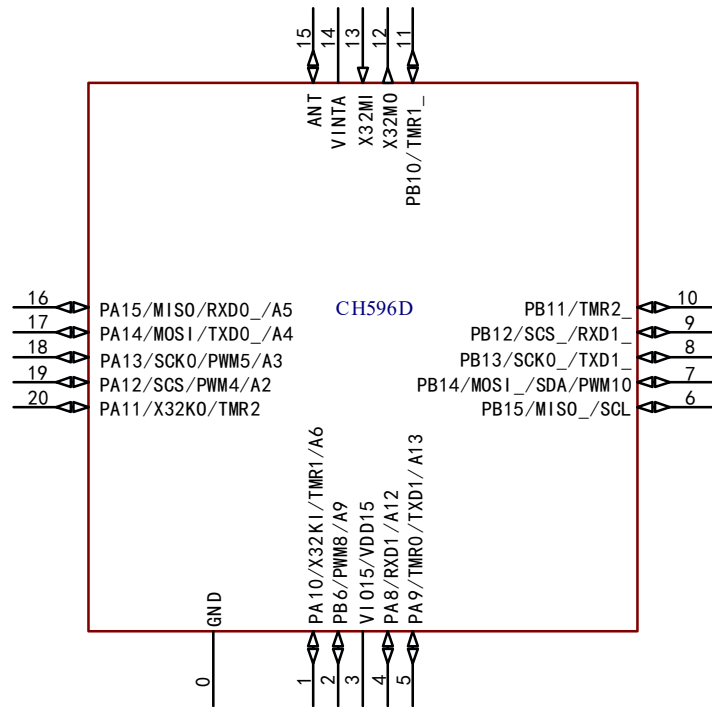
● 封装形式: QFN

支持最低 1.2V 电源电压，适用于单节小型电池供电的应用场景。

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	BLE蓝牙	ADC	电容触摸按键	最低电源电压	通用I/O	封装形式
CH596D	240+16K	24+8K	√	3	3	3+4	2	√	√	√	8	8路	1.2V	15	QFN20

第 1 章 引脚信息

1.1 引脚排列



1.2 引脚描述

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。

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
0	GND	P	-	芯片底板，公共接地端，电压 0V 参考点。
1	PA10	I/O/A	X32K1/TMR1 /ENCCAP_IN1_ /AIN6	PA10：通用双向数字 I/O 引脚。 X32K1：低频振荡器的输入端，外接 32KHz 晶体的另一端。 TMR1：定时器 1 的捕捉输入 1 和 PWM 输出通道 1。 ENCCAP_IN1_：编码器捕获通道 1 映射。 AIN6：ADC 模拟信号输入通道 6。
2	PB6	I/O/A	PWM8 /AIN9	PB6：通用双向数字 I/O 引脚。 PWM8：PWM 输出通道 8。 AIN9：ADC 模拟信号输入通道 9。
3	VDD15	P	VBAT	电池电源输入，需贴近引脚外接 2.2uF 并联 0.1uF 退耦电容。
	VI015	P	-	I/O 电源输入。
4	PA8	I/O/A	RXD1 /AIN12	PA8：通用双向数字 I/O 引脚。 RXD1：UART1 串行数据输入。 AIN12：ADC 模拟信号输入通道 12。
5	PA9	I/O/A	TMR0 /TXD1 /AIN13	PA9：通用双向数字 I/O 引脚。 TMR0：定时器 0 的捕捉输入 0 和 PWM 输出通道 0。 TXD1：UART1 串行数据输出通道。 AIN13：ADC 模拟信号输入通道 13。
6	PB15	I/O/5VT	MISO_ /SCL /DTR /LED_CLK_	PB15：通用双向数字 I/O 引脚。 MISO_：SPI 的 MISO 引脚映射。 SCL：I2C 串行时钟引脚，主机输出和输入/从机输入。 DTR：UART0 的 MODEM 输出信号，数据终端就绪。 LED_CLK_：LED 输出时钟引脚映射。
7	PB14	I/O/5VT	MOSI_ /SDA /PWM10 /LED[0]_ /DSR	PB14：通用双向数字 I/O 引脚。 MOSI_：SPI 的 MOSI 引脚映射。 SDA：I2C 的 SDA 引脚。 PWM10：PWM 输出通道 10。 LED[0]_：LED 输出 bit0 引脚映射。 DSR：UART0 的 DSR 引脚。
8	PB13	I/O/5VT	SCK_ /TXD1_ /LED[1]_ /DCD_	PB13：通用双向数字 I/O 引脚。 SCK_：SPI 的 SCK 引脚映射。 TXD1_：UART1 的 TXD1 引脚映射。 LED[1]_：LED 输出 bit1 引脚映射。 DCD_：UART0 的 DCD 引脚映射。
9	PB12	I/O/5VT	LED[2]_ /RXD1_ /SCS_ /RI_	PB12：通用双向数字 I/O 引脚。 LED[2]_：LED 输出 bit2 引脚映射。 RXD1_：UART1 的 RXD1 引脚映射。 SCS_：SPI 的 SCS 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
				RI_: UART0 的 RI 引脚映射。
10	PB11	I/O/A	LED[3]_ /ENCCAP_IN2 /TMR2_	PB11: 通用双向数字 I/O 引脚。 LED[3]_: LED 输出 bit3 引脚映射。 ENCCAP_IN2: 编码器捕获通道 2。 TMR2_: 定时器 2 的 TMR2、CAP2 引脚映射。
11	PB10	I/O/A	ENCCAP_IN1 /TMR1_	PB10: 通用双向数字 I/O 引脚。 ENCCAP_IN1: 编码器捕获通道 1。 TMR1_: 定时器 1 的 TMR1、CAP1 引脚映射。
12	X32M0	I/A	-	高频振荡器 HSE 的反相输出端, 外接 32MHz 晶体的一端。
13	X32MI	A	-	高频振荡器 HSE 的输入端, 外接 32MHz 晶体的另一端。
14	VINTA	P	-	内部模拟电路的电源节点, 需贴近引脚外接退耦电容。建议 1uF(支持 0.1uF~2.2uF, 容值大功耗略大)。
15	ANT	A	-	RF 射频信号输入输出, 建议直连天线。
16	PA15	I/O/A	MISO /LED_CLK /RXD0_ /AIN5	PA15: 通用双向数字 I/O 引脚。 MISO: SPI 的 MISO 引脚, 主机输入/从机输出。 LED_CLK: LED 输出时钟。 RXD0_: UART0 的 RXD0 引脚映射。 AIN5: ADC 模拟信号输入通道 5。
17	PA14	I/O/A	MOSI /TXD0_ /LED[0] /AIN4	PA14: 通用双向数字 I/O 引脚。 MOSI: SPI 的 MOSI 引脚, 主机输出/从机输入。 TXD0_: UART0 的 TXD0 引脚映射。 LED[0]: LED 输出 bit0。 AIN4: ADC 模拟信号输入通道 4。
18	PA13	I/O/A	SCK /PWM5 /LED[1] /SWCLK /AIN3	PA13: 通用双向数字 I/O 引脚。 SCK: SPI 的串行时钟引脚, 主机输出/从机输入。 PWM5: PWM 输出通道 5。 LED[1]: LED 输出 bit1。 SWCLK ⁽³⁾ : 仿真调试接口的串行时钟输入。 AIN3: ADC 模拟信号输入通道 3。
19	PA12	I/O/A	SWDIO /PWM4 /LED[2] /SCS/AIN2	PA12: 通用双向数字 I/O 引脚。 SWDIO ⁽³⁾ : 仿真调试接口的串行数据输入输出。 PWM4: PWM 输出通道 4。 LED[2]: LED 输出 bit2。 SCS: SPI 的 SCS 片选输入, 低有效。 AIN2: ADC 模拟信号输入通道 2。
20	PA11	I/O/A	TMR2 /LED[3] /ENCCAP_IN2_ /X32K0	PA11 ⁽⁴⁾ : 通用双向数字 I/O 引脚。 TMR2: 定时器 2 的捕捉输入 2 和 PWM 输出通道 2。 LED[3]: LED 输出 bit3。 ENCCAP_IN2_: 编码器捕获通道 2 映射。 X32K0: 低频振荡器的反相输出端, 外接 32KHz 晶体一端。

注: 1. 引脚类型: 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=可承受 5V 信号电压输入。

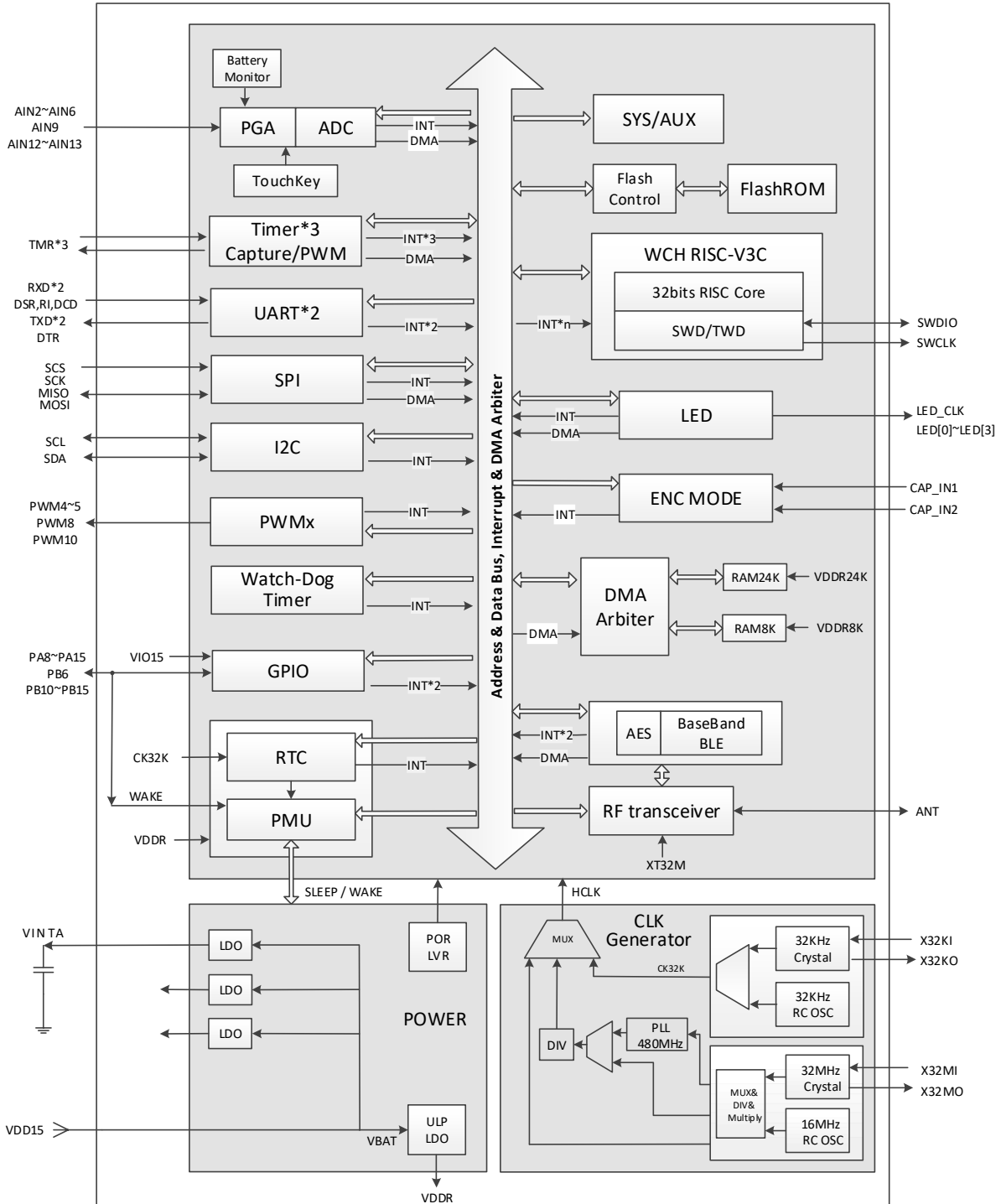
2. 引脚的复用功能及映射根据其优先级在表中按从高到底排列，其中 GPIO 功能为最低优先级。
3. 两线仿真调试接口通过 ISP 工具配置。仿真调试接口启用后，PA13 和 PA12 仅用作 SWCLK 和 SWDIO，不再用于 GPIO 或外设复用功能引脚。关闭仿真调试接口后，PA13 和 PA12 才可用于 GPIO 和外设复用功能引脚。
4. 不支持上拉，睡眠模式时不支持推挽输出高。

第 2 章 系统结构及存储器

2.1 系统结构

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

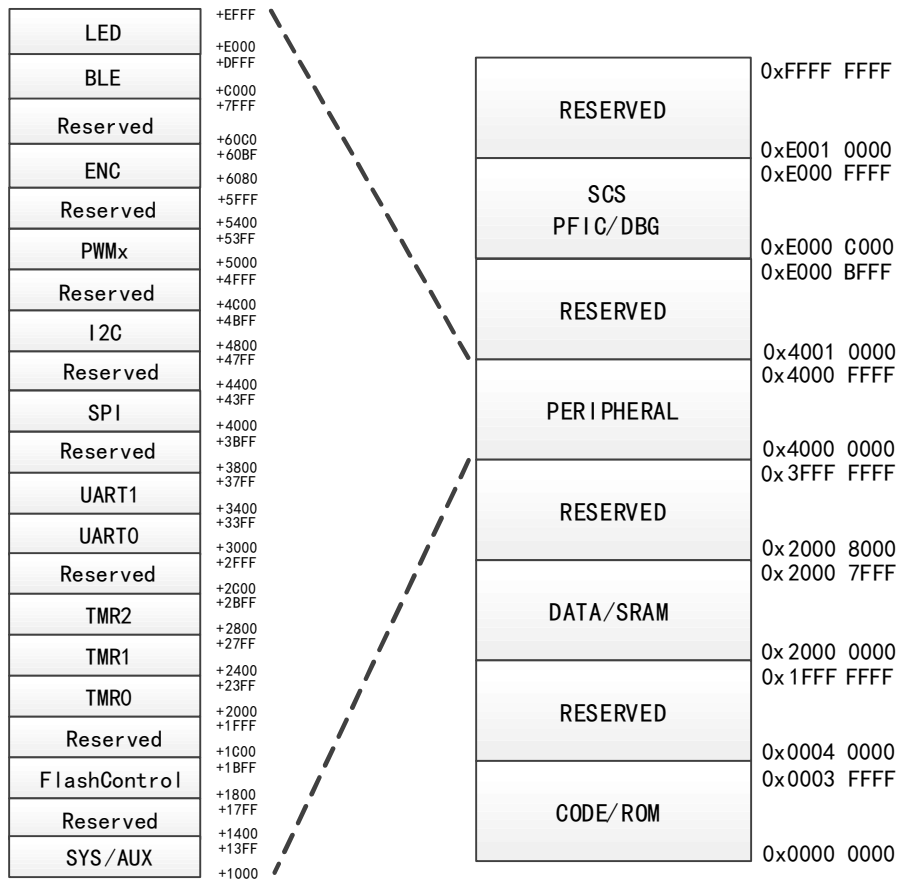
图 2-1 内部结构框图



2.2 存储器映像

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	-
0x20000000-0x20007FFF	片上 DATA 区域，易失存储器	32KB, SRAM
0x20008000-0x3FFFFFFF	保留	-
0x40000000-0x4000FFFF	各种外设	多个外设模块
0x40010000-0xE000BFFF	保留	-
0xE000C000-0xE000FFFF	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
0x0003C000-0x0003FFFF	系统引导程序存储区 BootLoader	16KB
0x00040100-0x000401FF	用户自定义信息存储区	256B
0x00040200-0x000402FF	系统非易失配置信息存储区 InfoFlash	256B

地址 0x00040100-0x000401FF 的配置信息可以由用户通过工具设置。

注：系统引导程序存储区也可用于存放用户应用程序，具体操作请参考 EVT 例程。

表 2-3 Boot 说明

芯片型号	Boot 引脚	下载接口	进入条件
CH596D	PB15 低电平	UART1_(PB12/PB13) 波特率：115200/1M/2M	1、串口支持免按键方式（上电 40ms 内检测串口通信数据包）。 2、上电检测 boot 引脚符合条件。 3、上电判断用户区 16byte[0-15]为擦除状态。

注：任何方式进入 Boot，10s 内无数据通信则退出 Boot；握手成功后，60s 无数据通信则退出 Boot。

表 2-4 用户级非易失配置信息说明

地址	位地址	名称	用途	默认值
0x40100	位 4~位 0	RESERVED	保留。	0
	位 5	CFG_IWDG_EN	独立看门狗使能： 1：开启独立开门狗； 0：关闭独立看门狗。	0
	位 6	CFG_BOOT_EN	系统引导程序 BootLoader 使能。	1
	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： 数值 0x3A：允许读出； 非数值 0x3A：禁止两线调试读出，程序保密。	0x3A
0x4010C	位 31~位 24	RESERVED	保留。	0
	位 31~位 0	VALID_SIG	配置信息有效标志，固定值。	0x235654AD

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 2-5 DATA 区域地址

地址范围	用途	描述
0x20000000-0x20005FFF	主+辅双电源供电的可独立保持存储区 RAM24K	24KB
0x20006000-0x20007FFF	主+辅双电源供电的可独立保持存储区 RAM8K	8KB

2.3.3 外设地址分配

芯片主要包含以下外设，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 2-6 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 等) AUX (ADC/TKEY/PLL 等)	0x4000 1000
2	FlashROM-Control	0x4000 1800
3	TMR0	0x4000 2000
4	TMR1	0x4000 2400
5	TMR2	0x4000 2800
6	UART0	0x4000 3000
7	UART1	0x4000 3400

8	SPI	0x4000 4000
9	I2C	0x4000 4800
10	PWMx (PWM8、PWM10、PWM4~PWM5)	0x4000 5000
11	ENC	0x4000 6080
12	Radio: BLE	0x4000 C000 0x4000 D000
13	LED	0x4000 E000

下表为后续章节寄存器描述中“访问”的解释说明：

表 2-7 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 2-8 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）
HSI	内部高频 RC 时钟振荡源（应用软件运行时校准后为 16MHz）
LSE	外部低频晶体振荡时钟源（建议 32KHz）
LSI	内部低频 RC 时钟振荡源（应用软件运行时校准后为 32KHz）
CK32M	高频时钟源（默认 32MHz）
T_{CK32M}	高频时钟周期（1/CK32M）
CK16M	高频时钟源（默认 16MHz）
T_{CK16M}	高频时钟周期（1/CK16M）
CK32K	低频时钟源（默认 32KHz）
Fp11	PLL 输出时钟（默认频率为 480MHz）
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）
RAM8K	高地址的 8KB SRAM
RAM24K	低地址的 24KB SRAM
0x	以其开头的的数据表示 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 18 个外设中断通道和 7 个内核中断通道，其他中断源保留。

3.1 中断控制器

18 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	系统嘀嗒定时器	0x0000_0030
13	-	-	-	保留	-
14	1	可设置	SWI	软件中断	0x0000_0038
15	-	-	-	保留	0x0000_003C
16	2	可设置	TMR0	TMR0 定时器 0 中断	0x0000_0040
17	3	可设置	GPIO_A	GPIO 端口 PA 通用 I/O 中断	0x0000_0044
18	4	可设置	GPIO_B	GPIO 端口 PB 通用 I/O 中断	0x0000_0048
19	5	可设置	SPI0	SPI0 中断	0x0000_004C
20	6	可设置	BLEB	无线模块的 BB 中断	0x0000_0050
21	7	可设置	BLEL	无线模块的 LLE 中断	0x0000_0054
22	-	-	-	保留	0x0000_0058
23	-	-	-	保留	0x0000_005C
24	8	可设置	TMR1	TMR1 定时器 1 中断	0x0000_0060
25	9	可设置	TMR2	TMR2 定时器 2 中断	0x0000_0064
26	10	可设置	UART0	UART0 异步串口 0 中断	0x0000_0068
27	11	可设置	UART1	UART1 异步串口 1 中断	0x0000_006C

28	12	可设置	RTC	RTC 和振荡器校准中断	0x0000_0070
29	13	可设置	ADC	ADC 和 TouchKey 中断	0x0000_0074
30	14	可设置	I2C	I2C 中断	0x0000_0078
31	15	可设置	PWMX	PWMX 中断	0x0000_007C
32	-	-	-	保留	0x0000_0080
33	-	-	-	保留	0x0000_0084
34	-	-	-	保留	0x0000_0088
35	16	可设置	WDOG_BAT	看门狗定时器中断/电池低电压中断	0x0000_008C
36	17	可设置	ENCODER	编码器中断	0x0000_0090
37	-	-	-	保留	0x0000_0094
38	18	可设置	LED	LED 中断	0x0000_0098

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	0x0000032C
R32_PFIC_ISR2	0x04	PFIC 中断使能状态寄存器 2	0x00000000
R32_PFIC_IPR1	0x20	PFIC 中断挂起状态寄存器 1	0x00000000
R32_PFIC_IPR2	0x24	PFIC 中断挂起状态寄存器 2	0x00000000
R32_PFIC_ITHRESDR	0x40	PFIC 中断优先级阈值配置寄存器	0x00000000
R32_PFIC_CFGR	0x48	PFIC 中断配置寄存器	0x00000000
R32_PFIC_GISR	0x4C	PFIC 中断全局状态寄存器	0x00000000
R32_PFIC_IDCFGFR	0x50	PFIC 快速中断 ID 配置寄存器	0x00000000
R32_PFIC_FIADDRR0	0x60	PFIC 快速中断 0 地址寄存器	0x00000000
R32_PFIC_FIADDRR1	0x64	PFIC 快速中断 1 地址寄存器	0x00000000
R32_PFIC_FIADDRR2	0x68	PFIC 快速中断 2 地址寄存器	0x00000000
R32_PFIC_FIADDRR3	0x6C	PFIC 快速中断 3 地址寄存器	0x00000000
R32_PFIC_IENR1	0x100	PFIC 中断使能设置寄存器 1	0x00000000
R32_PFIC_IENR2	0x104	PFIC 中断使能设置寄存器 2	0x00000000
R32_PFIC_IRER1	0x180	PFIC 中断使能清除寄存器 1	0x00000000
R32_PFIC_IRER2	0x184	PFIC 中断使能清除寄存器 2	0x00000000
R32_PFIC_IPSR1	0x200	PFIC 中断挂起设置寄存器 1	0x00000000
R32_PFIC_IPSR2	0x204	PFIC 中断挂起设置寄存器 2	0x00000000
R32_PFIC_IPRR1	0x280	PFIC 中断挂起清除寄存器 1	0x00000000
R32_PFIC_IPRR2	0x284	PFIC 中断挂起清除寄存器 2	0x00000000
R32_PFIC_IACTR1	0x300	PFIC 中断激活状态寄存器 1	0x00000000
R32_PFIC_IACTR2	0x304	PFIC 中断激活状态寄存器 2	0x00000000
R32_PFIC_IPRIORx	0x400	PFIC 中断优先级配置寄存器	0x00000000
R32_PFIC_EENR	0xC80	PFIC 事件使能寄存器	0xFFFFFFFF
R32_PFIC_EPR	0xC84	PFIC 事件挂起寄存器	0x00000000
R32_PFIC_EWUPR	0xC88	PFIC 事件唤醒寄存器	0x00000000
R32_PFIC_SCTLR	0xD10	PFIC 系统控制寄存器	0x00000000

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 (PFIC_ISR1)

位	名称	访问	描述	复位值
[31:12]	INTENSTA	RO	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。	0
[11:0]	Reserved	RO	保留。 Reset、NMI、EXC、ECALL 等中断位, 下同。	320h

PFIC 中断使能状态寄存器 2 (PFIC_ISR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	RO	保留。	0
[3:0]	INTENSTA	RO	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。	0

PFIC 中断挂起状态寄存器 1 (PFIC_IPR1)

位	名称	访问	描述	复位值
[31:12]	PENDSTA	RO	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。	0
[11:0]	Reserved	RO	保留。	0

PFIC 中断挂起状态寄存器 2 (PFIC_IPR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	RO	保留。	0
[3:0]	PENDSTA	RO	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。	0

PFIC 中断优先级阈值配置寄存器 (PFIC_I THRESDR)

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[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。 [7:4]: 优先级阈值; [3:0]: 保留, 固定为 0, 写无效。	0

PFIC 中断配置寄存器 (PFIC_CFGR)

位	名称	访问	描述	复位值
[31:16]	KEYCODE	WO	对于不同的目标控制位, 需要同步写入相应的安全访问标识数据才能修改, 读出数据固定为 0: KEY1 = 0xFA05; KEY2 = 0xBCAF; KEY3 = 0xBEEF。	0
[15:8]	Reserved	RO	保留。	0
7	RESETSYS	WO	系统复位 (同步写入 KEY3)。自动清 0。	0

			写 1 有效，写 0 无效。 与 PFIC_SCTLR 寄存器 SYSRESET 位作用相同。	
[6:0]	Reserved	R0	保留。	0

PFIC 中断全局状态寄存器 (PFIC_GISR)

位	名称	访问	描述	复位值
[31:10]	Reserved	R0	保留。	0
9	GPENDSTA	R0	当前是否有中断处于挂起： 1: 有； 0: 没有。	0
8	GACTSTA	R0	当前是否有中断被执行： 1: 有； 0: 没有。	0
[7:0]	NESTSTA	R0	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 (PFIC_IDCFGR)

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。	0
[23:16]	FIID2	RW	配置快速中断 2 的中断编号。	0
[15:8]	FIID1	RW	配置快速中断 1 的中断编号。	0
[7:0]	FIID0	RW	配置快速中断 0 的中断编号。	0

PFIC 快速中断 0 地址寄存器 (PFIC_FIADDRR0)

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。	0
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 (PFIC_FIADDRR1)

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。	0
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 (PFIC_FIADDRR2)

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。	0
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 (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。	0
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 (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。	0
[11:0]	Reserved	RO	保留。	0

PFIC 中断使能设置寄存器 2 (PFIC_IENR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	RO	保留。	0
[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。	0

PFIC 中断使能清除寄存器 1 (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。	0
[11:0]	Reserved	RO	保留。	0

PFIC 中断使能清除寄存器 2 (PFIC_IRER2)

位	名称	访问	描述	复位值
[31:4]	Reserved	RO	保留。	0
[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。	0

PFIC 中断挂起设置寄存器 1 (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。	0
[11:0]	Reserved	RO	保留。	0

PFIC 中断挂起设置寄存器 2 (PFIC_IPSR2)

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

			0: 无影响。	
--	--	--	---------	--

PFIC 中断挂起清除寄存器 1 (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。	0
[11:0]	Reserved	RO	保留。	0

PFIC 中断挂起清除寄存器 2 (PFIC_IPRR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	RO	保留。	0
[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。	0

PFIC 中断激活状态寄存器 1 (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。	0
[11:0]	Reserved	RO	保留。	0

PFIC 中断激活状态寄存器 2 (PFIC_IACTR2)

位	名称	访问	描述	复位值
[31:4]	Reserved	RO	保留。	0
[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。	0

PFIC 中断优先级配置寄存器 (PFIC_IPRIORx) (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
IPRIOR63	PRIO_255		PRIO_254		PRIO_253		PRIO_252	
...
IPRIORx	PRIO_(4x+3)		PRIO_(4x+2)		PRIO_(4x+1)		PRIO_(4x)	
...
IPRIOR0	PRIO_3		PRIO_2		PRIO_1		PRIO_0	

位	名称	访问	描述	复位值
[2047:2040]	IP_255	RW	同 IP_0 描述。	0
...
[31:24]	IP_3	RW	同 IP_0 描述。	0
[23:16]	IP_2	RW	同 IP_0 描述。	0

[15:8]	IP_1	RW	同 IP_0 描述。	0
[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。	0

PFIC 事件使能寄存器 (PFIC_EENR)

位	名称	访问	描述	复位值
[31:0]	EVENTEN	RW	31#-0#事件唤醒使能： 1：允许对应事件在睡眠中唤醒内核； 0：屏蔽对应事件，本事件发生时不允许唤醒内核。 最多支持 32 个事件源，其中 0-7 为电平触发（高有效），8-31 为脉冲触发： 0#：内核暂停请求； 1#：内核中断请求； 7#-2#：保留； 8#：内核进入中断； 9#：内核退出中断； 10#：新的中断挂起；（对于共享中断，不区分该中断是否路由至本地内核）； 11#：内核发送事件；（本地内核的发送事件）； 12#：内核接收事件；（其它内核的发送事件）； 13#：系统接收事件；（系统配置的事件，例如 GPIO 翻转事件…）； 31#-14#：保留。	0xFFFF FFFFFF

PFIC 事件挂起寄存器 (PFIC_EPR)

位	名称	访问	描述	复位值
[31:8]	EVENT_PEND	RW1Z	31#-8#事件挂起状态，写 1 可清零： 1：对应事件发生未被处理； 0：对应事件未发生。	0
[7:0]	EVENT_PEND	RO	7#-0#事件挂起状态，不可清除： 1：对应事件发生未被处理； 0：对应事件未发生。	0

PFIC 事件唤醒寄存器 (PFIC_EWUPR)

位	名称	访问	描述	复位值
[31:0]	EVENT_WUP	RO	31#-0#事件唤醒寄存器： 1：对应事件将内核从睡眠中唤醒； 0：对应事件不是本次唤醒内核的事件。	0

PFIC 系统控制寄存器 (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。 与 PFIC_CFGFR 寄存器相同效果。	0
[30:7]	Reserved	RO	保留。	0

6	LOCKUP	RW	死锁使能： 1：关闭； 0：开启。	0
5	SETEVENT	WO	设置事件，可以唤醒 WFE 的情况。	0
4	SEVONPEND	RW	当发生事件或者中断挂起状态时，可以从 WFE 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：启用的事件和所有中断都能唤醒系统（包括未开启中断）； 0：只有启用的事件和启用的中断可以唤醒系统。	0
3	WFIOWFE	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：deepsleep； 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、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。说明：此类寄存器标注为“MRW”属性的需要系统在机器模式下才能访问。

异常入口基地址寄存器（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。	0
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

用户模式全局中断使能寄存器（GINTENR）

该寄存器用于控制全局中断的使能与屏蔽，机器模式下全局中断的使能与屏蔽可以通过寄存器 `mstatus` 中的 MIE, MPIE 位控制，该寄存器用户模式下无法操作。而全局中断使能寄存器 `gintenr` 是 `mstatus` 中 MIE 和 MPIE 的映射，用户模式下可以通过操作 `gintenr`，用于 MIE 和 MPIE 的置位和清零。

CSR 地址：0x800

位	名称	访问	描述	复位值
[31:13]	Reserved	URO	保留。	0
[12:11]	MPP[1:0]	URO	进中断前特权模式。	0
[10:8]	Reserved	URO	保留。	0
7	MPIE	URW	当 0xBC0 (CSR) bit5 位使能时，该位在用户模	0

			式下可读可写。	
[6:4]	Reserved	URO	保留。	0
3	MIE	URW	当 0xBC0 (GSR) bit5 位使能时, 该位在用户模式下可读可写。	0
[2:0]	Reserved	URO	保留。	0

中断系统控制寄存器 (INTSYSCR)

GSR 地址: 0x804

位	名称	访问	描述	复位值
31	LOCK	URO	1: 该寄存器仅机器模式下可读写; 0: 该寄存器用户模式下可读写。	0
[30:6]	Reserved	URO	保留。	0x380
5	GIHWSTKNEN	URW1	全局中断和硬件压栈关闭使能。 注: 该位常使用于实时操作系统中, 中断切换上下文时, 置位该位, 可关闭全局中断和硬件压栈出栈, 当上下文切换完成, 执行完中断返回后, 硬件自动清除该位。	0
4	Reserved	URO	保留。	0
[3:2]	PMTCFG	RW	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 [7] 位为抢占优先级; 10: 抢占位宽为 2, 即中断优先级寄存器的 [7:6] 位为抢占优先级; 11: 抢占位宽为 3, 即中断优先级寄存器的 [7:5] 位为抢占优先级。	0x1
1	INESTEN	URO	中断嵌套功能使能, 固定值为 1: 1: 使能; 0: 关闭。 注: 实际嵌套级数由 GSR 0xBC1 中 NEST_LVL 控制。	1
0	HWSTKEN	URW	硬件压栈使能: 1: 硬件压栈功能使能; 0: 硬件压栈功能关闭。	0

中断嵌套控制寄存器 (INESTCR)

GSR 地址: 0xBC1

位	名称	访问	描述	复位值
31	Reserved	MRO	保留。	0
30	NEST_OV	MRW	中断/异常嵌套溢出标志位, 写 1 清零: 1: 中断溢出标志; 0: 中断未溢出。 注: 中断溢出仅会在执行二级中断服务函数产生指令异常或 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。	0
[29:12]	Reserved	MRO	保留。	0

[11:8]	NEST_STA[3:0]	MRO	嵌套状态标志位： 0000：无中断； 0001：一级中断； 0011：二级中断，（一级嵌套）； 0111：三级中断，（溢出）； 1111：四级中断，（溢出）。	0
[7:2]	Reserved	MRO	保留。	0
[1:0]	NEST_LVL[1:0]	MRW	嵌套等级： 00：禁止嵌套，关闭嵌套功能； 01：一级嵌套，打开嵌套功能； 其他：无效。 注：对该域写 10 或 11，该域被置为 01，对该与写 11 时，读该寄存器可获得芯片最高嵌套等级。	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。	0																					
[23:16]	pmp2cfg	MRW	见 pmp0cfg。	0																					
[15:8]	pmp1cfg	MRW	见 pmp0cfg。	0																					
[7:0]	pmp0cfg	MRW	<table border="1"> <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。	0
			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, $\text{pmpaddr2} \leq \text{region} < \text{pmpaddr3}$ 。 $\text{pmpaddr}_{i-1} = \text{A_ADDR} \gg 2$; $\text{pmpaddr}_i = \text{B_ADDR} \gg 2$ 。
2	NA4	固定 4 字节区域保护。 pmp0cfg~pmp3cfg 对应 pmpaddr0~pmpaddr3 作为起始地址。 $\text{pmpaddr}_i = \text{A_ADDR} \gg 2$ 。
3	NAP0T	保护 $2^{(G+2)}$ 区域, $G \geq 1$, 其中 A_ADDR 为 $2^{(G+2)}$ 对齐。 $\text{pmpaddr}_i = (\text{A_ADDR} \gg 2) (2^{(G-1)} - 1)$ 。

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。	0

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。	0

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。	0

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	0

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	0xE000F008	系统计数器低位寄存器	0xFFFFFFFF
R32_STK_CMPLR	0xE000F010	计数比较低位寄存器	0xFFFFFFFF

系统计数控制寄存器 (STK_CTRL)

位	名称	访问	描述	复位值
31	SWIE	RW	软件中断触发使能: 1: 触发软件中断; 0: 关闭触发。 进入软件中断后须将此位清 0, 否则一直触发。	0
[30:6]	Reserved	RO	保留。	0
5	INIT	W1	计数器初始值更新: 1: 向上计数从 0 开始, 向下计数从比较值开始;	0

			0: 无效。	
4	MODE	RW	计数模式: 1: 向下计数; 0: 向上计数。	0
3	STRE	RW	自动重载计数使能: 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

系统计数状态寄存器 (STK_SR)

位	名称	访问	描述	复位值
31	SWIE	RW	软件中断触发使能: 1: 触发软件中断; 0: 关闭触发。 进入软件中断后须将此位清 0, 否则一直触发。	0
[30:1]	Reserved	RO	保留。	0
0	CNTIF	RWO	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

系统计数器低位寄存器 (STK_CNTL)

位	名称	访问	描述	复位值
[31:0]	CNT[31:0]	RW	当前计数器计数值低 32 位。	X

计数比较低位寄存器 (STK_CMPLR)

位	名称	访问	描述	复位值
[31:0]	CMP[31:0]	RW	设置比较计数器值低 32 位。	X

第 4 章 系统控制

4.1 复位控制

芯片支持 5 种复位形式，分别为电源上电复位 RPOR(real power on 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)。

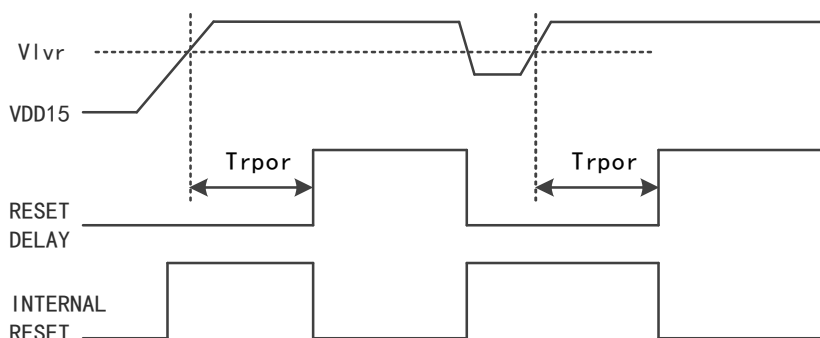
寄存器 R8_GLOB_RESET_KEEP 和 RB_ROM_CODE_OFS，只在 RPOR 或者 GRWSM 复位时被复位，而不受其它复位形式影响。

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 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。

4.1.3 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。

4.1.4 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。

4.1.5 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	0x96
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	0xX9
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
R8_PREPG_CTRL	0x40001800	FlashROM 预编程控制寄存器	0x00
R8_BUF_CTRL	0x40001801	FlashROM 页锁寄存器控制寄存器	0x00
R8_ERASE_CTRL	0x40001802	FlashROM 擦除控制寄存器	0x00
R8_PROGRAM_CTRL	0x40001803	FlashROM 编程控制寄存器	0x00
R8_READ_CTRL	0x40001807	FlashROM 读控制寄存器	0x00
R8_RDPRCTDIS_CTRL	0x40001808	FlashROM 读保护控制寄存器	0x00
R8_CHIP_STATUS	0x40001809	芯片状态寄存器	0x04
R32_FLASH_ADDR	0x4000180C	FlashROM 地址寄存器	0x00000000
R32_FLASH_DATA	0x40001810	FlashROM 数据寄存器	0x00000000
R32_FLASH_DATA1	0x40001814	FlashROM 数据寄存器 1	0x00000000
R32_FLASH_DATA2	0x40001818	FlashROM 数据寄存器 2	0x00000000
R32_FLASH_DATA3	0x4000181C	FlashROM 数据寄存器 3	0x00000000
R32_FLASH_STATUS	0x40001820	FlashROM 状态寄存器	0x000DXXXX

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,	0

			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。	
7	Reserved	R0	保留。	0
[6:4]	RB_SAFE_ACC_TIMER	R0	当前安全访问时间计数。	0
3	RB_SAFE_ACC_ACT	R0	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
2	Reserved	R0	保留。	0
[1:0]	RB_SAFE_ACC_MODE	R0	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。	0

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	固定值 96h，用于识别芯片。	96h

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。	0

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	RB_ROM_CODE_WE	RWA	FlashROM 程序存储区 CodeFlash 的擦除/编程使能位： 1：允许擦除/编程； 0：该区擦写保护。	0
6	RB_ROM_DATA_WE	RWA	FlashROM 数据存储区 DataFlash 的擦除/编程使能位： 1：允许擦除/编程； 0：全部擦写保护。	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 的起始偏移地址，该值不受 SR、WTR 或 GRWSM 影响，仅在 RPOR 有效时才能清零： 1：0x040000（跳过 ROM 中前 256KB）； 0：0x000000。	0
3	Reserved	R0	保留。	0
[2:0]	RB_RESET_FLAG	R0	最近一次复位状态：	001b

			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: 保留; 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。
--	--	--	--

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	Boot loader 状态: 1: 当前处于 Boot loader 状态; 0: 当前处于用户程序状态。	1/0
4	Reserved	R0	保留。	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
[2:0]	Reserved	R0	保留。	001b

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。	0
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 RB_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 (RB_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: 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
0	RB_BOOT_LOADER_MAN	R0	手动进入 BOOT LOADER 标志: 1: 手动进入 BOOT; 0: 空闲, 无动作。 (当配置 RB_WDOG_INT_EN 与 RB_ROM_CODE_WE 为 1、RB_ROM_DATA_WE 为 0、 $128 \leq RB_WDOG_COUNT < 192$ 时, 位 RB_BOOT_LOAD_MAN 为 1 并且从 BOOT 启动, 否则位 RB_BOOT_LOAD_MAN 为 0)。	0

	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。	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。	0

FlashROM 预编程控制寄存器 (R8_PREPG_CTRL)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
4	RB_PREPG_START	RWA	预编程启动，预编程过程中一直为高，预编程完成会自动清 0。 如果 RRB_PREG_AUTO_EN 置为 1，则等到相应的擦除过程完成才能清 0。 1：启动预编程。	0
3	Reserved	RO	保留。	0
[2:1]	RB_PREG_MODE	RWA	预编程模式： 00：页预编程（256BYTE）； 01：扇区预编程（2K）； 10：BANK 区预编程（64K），由地址 R32_FLASH_ADDR 的 BIT[8] 决定是奇数页还是偶数页。1：奇数页；0：偶数页； 11：BANK 区预编程（64K）。	0
0	RB_PREG_AUTO_EN	RWA	预编程自动模式： 1：预编程完成后会自动进行擦除操作（需要提前配置擦除命令 RB_ERASE_MODE）； 0：预编程完成后不会自动进行擦除操作。	0

FlashROM 页锁寄存器控制寄存器 (R8_BUF_CTRL)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	0
5	RB_BUF_LOAD_AUTO	RWA	1：页锁寄存器加载自动模式开启，开启后写数据 R32_FLASH_DATA 会触发页锁寄存器 RB_BUF_LOAD_START 拉高并开始加载，页锁寄存器加载完成 RB_BUF_LOAD_START 会自动清 0。 0：页锁寄存器加载自动模式关闭。	0
4	RB_BUF_LOAD_START	RWA	页锁寄存器加载启动，页锁寄存器加载过程中一直为高，页锁寄存器加载完成会自动清 0。 1：启动页锁寄存器加载。	0
[3:1]	Reserved	RO	保留。	0
0	RB_BUF_CLR_START	RWA	页锁寄存器清除启动，页锁寄存器清除过程中一直为高，页锁寄存器清除完成会自动清 0。 1：启动页锁寄存器清除。	0

FlashROM 擦除控制寄存器 (R8_ERASE_CTRL)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
4	RB_ERASE_START	RWA	擦除启动，擦除过程中一直为高，擦除完成会自动清0。 1：启动FlashROM擦除。	0
[3:2]	Reserved	RO	保留。	0
[1:0]	RB_ERASE_MODE	RWA	00：页擦除； 01：扇区擦除； 10/11：BANK擦除。	0

FlashROM 编程控制寄存器 (R8_PROGRAM_CTRL)

位	名称	访问	描述	复位值
[7:1]	Reserved	RO	保留。	0
0	RB_PROGRAM_START	RWA	页编程启动，编程过程中一直为高，编程完成会自动清0； 1：启动FlashROM页编程。	0

FlashROM 读控制寄存器 (R8_READ_CTRL)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	0
[5:4]	RB_FLASH_CLK	RWA	FLASH 预编程/擦除/页锁寄存器加载/页锁寄存器清除/编程/读等操作时钟调节： 00：高电平为系统时钟周期的一半； 01/10：高电平为一个系统时钟周期； 11：高电平为两个系统时钟周期。 注：高电平必须大于10ns。	0
3	Reserved	RO	保留。	0
2	RB_WRRD_ADDR_INC	RWA	1：读写一次完成后地址会自动加1； 0：读写1次完成后地址不变。 注：“加1”为4个字。	0
1	RB_RD_START	RO	读操作启动，读操作完成后自动清0： 1：启动读操作。	0
0	RB_CFG_RD_EN	RWA	1：读操作此时R32_FLASH_DATA会更新读出的值，但是该地址不能写数据； 0：写使能R32_FLASH_DATA可以写入编程数据。	0

FlashROM 读保护控制寄存器 (R8_RDPRCTDIS_CTRL)

位	名称	访问	描述	复位值
[7:0]	R8_RDPRCTDIS_CTRL	WO	读保护控制寄存器： 该地址依次配置0x3C/0x9A/0x6D/0x73使能开始读保护解除； 该地址依次配置0x3C/0x9A/0x6D/0x99清除读保护解除逻辑。	0
[7:6]	RB_RD_PROTECTDIS_START	RO	当前读保护控制状态。 11：可在R8_RDPRCTDIS_CTRL，配置0x73使能开始读保护解除，或配置0x99清除读保护解除逻辑。	

[5:4]	Reserved	R0	保留。	0
[3:1]	RB_RDPROTECT_DISSTATE	R0	000: 读保护解除未启动; 001~011: 正在进行读保护解除; 100: 读保护解除完成; 101: 读保护解除失败。	0
0	RB_RD_PROTECTDIS_FLAG	R0	1: 处于解除读保护过程中; 0: 未处于解除读保护过程。	0

芯片状态寄存器 (R8_CHIP_STATUS)

位	名称	访问	描述	复位值
[7:4]	Reserved	R0	保留。	0
3	RB_FLASH_HALTED	R0	DEBUG 模式下: 1: FLASH 禁止操作; 0: FLASH 允许操作。	0
2	Reserved	R0	保留。	1
1	RB_DBG_MODE	R0	1: 进入调试模式; 0: 未进入调试模式。	0
0	RB_RD_PROTECT	R0	1: 读保护开启; 0: 读保护解除。	0

FlashROM 地址寄存器 (R32_FLASH_ADDR)

位	名称	访问	描述	复位值
[31:19]	Reserved	R0	保留。	0
[18:2]	R32_FLASH_ADDR	RWA	RB_CFG_RD_EN = 0 时, R32_FLASH_ADDR [7:2] 表示页锁寄存器加载地址, R32_FLASH_ADDR[18:2] 编程地址 (页锁寄存器加载和编程操作是独立的因此不冲突, 是分时用的); RB_CFG_RD_EN = 1 时, R32_FLASH_ADDR [18:2] 表示读数据地址。	0
[1:0]	Reserved	R0	保留。	0

FlashROM 数据寄存器 (R32_FLASH_DATA)

位	名称	访问	描述	复位值
[31:0]	R32_FLASH_DATA	RWA	在 RB_CFG_RD_EN = 0 表示编程数据; RB_CFG_RD_EN = 1 表示读数据; 四字对齐情况下, R32_FLASH_DATA 表示第一个字; R32_FLASH_DATA1 表示第二个字; R32_FLASH_DATA2 表示第三个字; R32_FLASH_DATA3 表示第四个字; 例如: 读的地址是 R32_FLASH_ADDR [18:2] = 0x00000, 则读完成后 R32_FLASH_DATA 为 R32_FLASH_ADDR [18:2] = 0x00000 的值, R32_FLASH_DATA1 为 R32_FLASH_ADDR [18:2] = 0x00001 的值。 R32_FLASH_DATA2 为 R32_FLASH_ADDR	0

			[18:2] = 0x0002 的值。 R32_FLASH_DATA3 为 R32_FLASH_ADDR [18:2] = 0x0003 的值。 即一次能读 4 个字。	
--	--	--	---	--

FlashROM 数据寄存器 1 (R32_FLASH_DATA1)

位	名称	访问	描述	复位值
[31:0]	R32_FLASH_DATA1	R0	Flash 读数据 1。	0

FlashROM 数据寄存器 2 (R32_FLASH_DATA2)

位	名称	访问	描述	复位值
[31:0]	R32_FLASH_DATA2	R0	Flash 读数据 2。	0

FlashROM 数据寄存器 3 (R32_FLASH_DATA3)

位	名称	访问	描述	复位值
[31:0]	R32_FLASH_DATA3	R0	Flash 读数据 3。	0

FlashROM 状态寄存器 (R32_FLASH_STATUS)

位	名称	访问	描述	复位值
31	RB_CTRL_BUSY	R0	1: 指示 FLASH 正在进行预编程/擦除/页锁寄存器清除/页锁寄存器加载/编程/读。	0
[30:20]	Reserved	R0	保留。	0
19	RB_FLASH_READY	R0	1: 可以进行预编程/擦除/页锁寄存器清除/页锁寄存器加载/编程/读操作; 0: 不可以进行预编程/擦除/页锁寄存器清除/页锁寄存器加载/编程/读操作。	1
[18:0]	Reserved	R0	保留。	5XXXh

4.4 Flash-ROM 操作步骤

4.4.1. 编程

编程开始前先写 R8_RESET_STATUS (0x40001044) 为 0xe1 开启擦除编程通道，打开安全寄存器通道。

编程操作需要 5 步操作：

- (1)、预编程 (pre-program) 所要编程的区域；
- (2)、擦除 (Erase) 所要编程的区域；
- (3)、清除页锁寄存器；
- (4)、将页锁寄存器中写入需要编程的数据；
- (5)、将页锁寄存器的内容编程到指定地址的页中；
- (6)、如果预编程和擦除的区域大于页编程，则可以重复 (3)~(5)，继续编程。比如预编程和擦除均为 2K，则 (3)~(5) 可以进行 8 次。

注：擦除动作必须执行步骤 (1) 和步骤 (2)。

以下是具体各个步骤的操作 (结合 FLASH 相关寄存器)。

预编程：

- (1)、地址 R32_FLASH_ADDR (0x4000180C)，写入预编程所需要指的地址；
- (2)、地址 R8_PREPG_CTRL (0x40001800)，写入预编程的命令，如果 bit[0] 同时置 1，则需要提前写 R8_ERASE_CTRL (0x40001802) bit[1:0] 的擦除模式。同时将 RB_PREPG_START 置 1 (即命令和使能可以一次写入) 开始预编程；

(3)、依次查询到 RB_CTRL_BUSY 置 0、RB_FLASH_READY 置 1 后，表示预编程结束。

擦除：

- (1)、地址 R32_FLASH_ADDR (0x4000180C)，写入擦除所需要指的地址；
- (2)、地址 R8_ERASE_CTRL (0x40001802)，写入擦除命令 (RB_ERASE_MODE) 和擦除开始使能 (RB_ERASE_START)；
- (3)、依次查询到 RB_CTRL_BUSY 置 0、RB_FLASH_READY 置 1 后，表示擦除结束。

清除页锁寄存器：

- (1)、地址 R8_BUF_CTRL (0x40001801)，写 RB_BUF_CLR_START 为 1，开始清除页锁寄存器；
- (2)、依次查询到 RB_CTRL_BUSY 置 0、RB_FLASH_READY 置 1 后，表示页锁寄存器清除结束。

页锁寄存器中写入需要编程的数据：

- (1)、地址 R8_FLASH_CTRL2 (0x40001807)，写 RB_CFG_RD_EN 为 0，确保 R32_FLASH_DATA 有效。(注意整个编程过程只需要确保 RB_CFG_RD_EN 为 0 即可，不必要每次都写，即可以在开始编程时统一写一次)；
- (2)、地址 R8_FLASH_CTRL2 (0x40001807)，RB_WRRD_ADDR_INC 置 1 (在每次页锁寄存器写入数据完成后地址 R32_FLASH_ADDR[7:2] 都会自动加 1，初始地址 R32_FLASH_ADDR[7:2] 需要配置)。地址 R8_BUF_CTRL (0x40001801)，RB_BUF_LOAD_AUTO 置 1，(在每次 R32_FLASH_DATA 写入数据会自动触发 RB_BUF_LOAD_START)；
- (3)、地址 R32_FLASH_ADDR (0x4000180C)，配置 R32_FLASH_ADDR[7:2] 初始地址；
- (4)、地址 R32_FLASH_DATA (0x40001810)，写入 R32_FLASH_DATA；
- (5)、依次查询到 RB_CTRL_BUSY 置 0、RB_FLASH_READY 置 1 后，表示此数据写入完成结束；
- (6)、重复 (4) 和 (5) 步骤。

编程：

- (1)、地址 R32_FLASH_ADDR (0x4000180C)，写入编程所需要指的地址；
- (2)、地址 R8_PROGRAM_CTRL (0x40001803)，写 RB_PROGRAM_START 为 1，开始页编程；
- (3)、依次查询到 RB_CTRL_BUSY 置 0、RB_FLASH_READY 置 1 后，表示页编程结束。

4.4.2. 读操作

读操作：

- (1)、地址 R8_FLASH_CTRL2 (0x40001807)，写 RB_CFG_RD_EN 为 1，确保 R32_FLASH_DATA 为读数据。配置读地址 R32_FLASH_ADDR[18:2]，RB_WRRD_ADDR_INC 置 0；
- (2)、读地址 R32_FLASH_DATA (0x40001810)，并依次查询到 RB_CTRL_BUSY 置 0、RB_FLASH_READY 置 1 后，再进行下一次读；

例如读的地址是 R32_FLASH_ADDR[18:2] = 0x00000；则读完成后，R32_FLASH_DATA 为 R32_FLASH_ADDR[18:2] = 0x00000 的值，R32_FLASH_DATA1 为 R32_FLASH_ADDR[18:2] = 0x00001 的值。R32_FLASH_DATA2 为 R32_FLASH_ADDR[18:2] = 0x00002 的值。R32_FLASH_DATA3 为 R32_FLASH_ADDR[18:2] = 0x00003 的值。即一次能读 4 个字。

自动地址增加的读操作：

- (1)、配置需要读取的初始地址，即 R32_FLASH_ADDR (0x4000180C)；
- (2)、配置读使能和读写地址增加使能，即 R8_FLASH_CTRL2 (0x40001807) (按字节写) 为 0x05；
- (3)、读取数据寄存器，即 R32_FLASH_DATA (0x40001810)，便会触发一次读，此时 R32_FLASH_DATA，R32_FLASH_DATA1，R32_FLASH_DATA2，R32_FLASH_DATA3 分别更新，并且地址自动增加到下一个 128 位；
- (4)、按照 R32_FLASH_DATA3，R32_FLASH_DATA2，R32_FLASH_DATA1，R32_FLASH_DATA 读取顺序进行，当读取 R32_FLASH_DATA 时会再次触发读；
- (5)、重复步骤 4 即可。

举例：初始地址是 0x00000：

第 3 步操作会使地址更新到 0x00010，R32_FLASH_DATA，R32_FLASH_DATA1，R32_FLASH_DATA2，R32_FLASH_DATA3 分别更新数据为 0x00000，0x00004，0x00008，0x0000C。

第 4 步分别读出 R32_FLASH_DATA3，R32_FLASH_DATA2，R32_FLASH_DATA1，R32_FLASH_DATA，当读到 R32_FLASH_DATA 时此时触发了地址 0x00010 的读，并将 DATA 都更新到下一个 128 位。以此类推。

4.5 芯片唯一 ID 号

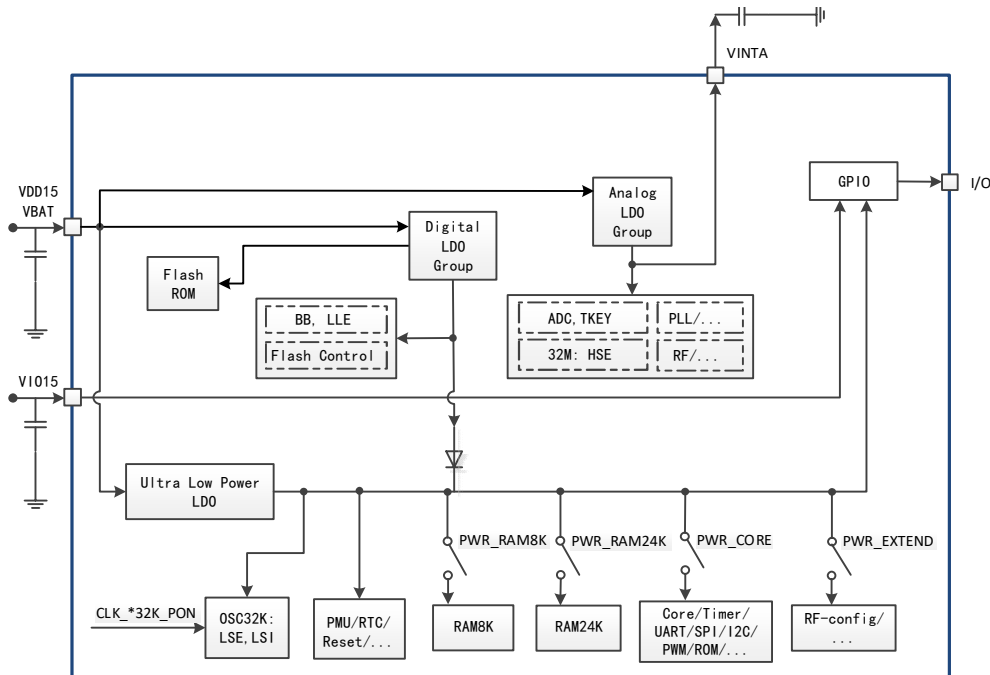
每个芯片出厂时都具有唯一的 ID 号，即芯片身份识别号。该 ID 数据及其校验和共 8 字节，存储于芯片内部只读区域中，具体操作请参考例子程序。

第 5 章 电源控制

5.1 电源管理

芯片内置有电源管理单元 PMU。系统电源从 VDD15 输入，通过内置的多个 LDO 电压调整器为系统的 FlashROM、系统的数字电路（包括内核等）和系统的模拟电路（包括高频振荡器、PLL、ADC 和 RF 收发器等）提供所需的电源。GPIO 的电源从 VDD15 输入。

图 5-1 电源系统



芯片提供了 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 LDO，切换成由系统内置的超低功耗 ULP-LDO 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，系统的高 8KB 和低 24KB 的 SRAM、内核及所有的外设皆可选择是否维持供电，LSE/LSI 可选择是否开启。

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
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	0x40001014	长复位配置寄存器	0x00
R8_SLP_CLK_OFF2	0x40001015	睡眠时钟控制寄存器 2	0x00
R16_SLP_WAKE_CFG	0x40001016	睡眠配置寄存器	0x0900
R16_POWER_PLAN	0x40001020	睡眠电源管理寄存器	0x11DF
R16_AUX_POWER_ADJ	0x40001022	辅助电源调整控制寄存器	0xXXXX
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	0x00
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
[7:6]	Reserved	RO	保留。	0
5	RB_SLP_CLK_UART1	RWA	串口 1 时钟源： 1: 关闭； 0: 开启。	0
4	RB_SLP_CLK_UART0	RWA	串口 0 时钟源： 1: 关闭； 0: 开启。	0
3	Reserved	RO	保留。	0
2	RB_SLP_CLK_TMR2	RWA	定时器 2 时钟源： 1: 关闭； 0: 开启。	0
1	RB_SLP_CLK_TMR1	RWA	定时器 1 时钟源： 1: 关闭； 0: 开启。	0
0	RB_SLP_CLK_TMR0	RWA	定时器 0 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	Reserved	RO	保留。	0
5	RB_SLP_CLK_ADC	RWA	ADC 时钟源： 1: 关闭； 0: 开启。	0
4	Reserved	RO	保留。	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_SLP_CLK_LED	RWA	LED 时钟源： 1: 关闭； 0: 开启。	0
0	RB_SLP_CLK_SPI0	RWA	SPI0 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： 1: RB_SLP_GPIO_EDGE_MODE=1, 边沿唤醒； RB_SLP_GPIO_EDGE_MODE=0, 遵照极性的	0

			边沿唤醒； 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_UART_WAKE	RWA	使能 UART 事件唤醒系统： 1：使能； 0：关闭。	0
0	Reserved	RO	保留。	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	RB_SLP_CLK_RAM8K	RWA	RAM8K 的 SRAM 时钟控制： 1：关闭； 0：开启。	0
4	RB_SLP_CLK_RAMX	RWA	主 SRAM (RAM24K) 的时钟控制： 1：关闭； 0：开启。	0
[3:2]	Reserved	RO	保留。	0
[1:0]	RB_WAKE_DLY_MOD	RWA	选择唤醒后的延时周期数： 11：无延时，8 个周期+ T_{SUCLK} ，禁用； 10：超短延时，70 个周期+ T_{SUCLK} ； 01：短延时，520 个周期+ T_{SUCLK} ，建议； 00：长延时，3590 个周期+ T_{SUCLK} 。 其中， T_{SUCLK} 取决于睡眠模式和时钟配置，可能包含 T_{SUHSE} 或 PLL 或两者合计的启动时间，具体组合参考评估板中示例。	0

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	RO	保留。	0

[2:1]	RB_LONG_TIM_SEL	RWA	长复位时长选择。 使用 HSE 时钟时： 11: 32.768ms; 10: 25.000ms; 01: 20.000ms; 00: 15.000ms。 使用 HSI 时钟时： 11: 65.536ms; 10: 50.000ms; 01: 40.000ms; 00: 30.000ms。	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	Reserved	RO	保留。	0
6	RB_SLP_ENC_WAKE	RWA	使能编码器事件唤醒系统： 1: 开启； 0: 关闭。	0
5	Reserved	RO	保留。	0
4	RB_CLK_OFF_ENC	RWA	1: 关闭编码器时钟； 0: 开启编码器时钟。	0
[3:2]	Reserved	RO	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	Reserved	RO	保留。	0

睡眠配置寄存器 (R16_SLP_WAKE_CFG)

位	名称	访问	描述	复位值
[15:13]	Reserved	RO	保留。	0
12	RB_DEBUG_EN	RWA	1: 调试使能开启； 0: 调试使能关闭。	0
11	RB_UART_DLY_IDLE	RO	开启 RB_UART_DLY_EN 后, UART 接收状态： 1: 无接收数据； 0: 有接收数据。	1
10	RB_UART_WAKE_INT	RW1	UART 唤醒中断标志： 1: 已产生中断； 0: 未产生中断。	0
9	RB_UART_DLY_EN	RWA	UART 接收延迟使能： 1: 开启延迟； 0: 关闭延迟。	0
8	RB_ACAUTO_ENABLE	RW	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	RO	保留。	0
[6:5]	RB_PRECLK_CNT_SEL	RWA	睡眠时间配置，睡眠时长=周期数*F _{sys} ： 11: 256；	0

			10: 512; 01: 1024; 00: 2048。	
4	RB_PRECLK_CNT_EN	RWA	1: 睡眠唤醒时需要等到一定时间才会释放 fclk; 0: 睡眠唤醒后立即释放 fclk。	0
[3:0]	Reserved	RO	保留。	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/ WZ	睡眠电源规划控制使能: 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。	0
14	RB_XT_PRE_EN	RWA	外部 32MHz 时钟 (HSE) 预先唤醒使能: 1: 使能, 此时必须采用 LSI 或 LSE 进行计数; 0: 禁止。	0
13	Reserved	RWA	保留, 必须写 1。	0
[12:11]	RB_XT_PRE_CFG	RWA	外部 32MHz 时钟 (HSE) 预先唤醒时间: 00: 0.531ms; 01: 0.968ms; 10: 1.406ms; 11: 1.781ms。	10b
[10:9]	Reserved	RO	保留。	0
8	RB_PWR_LDO_EN	RWA	内部 LDO 控制 (睡眠规划): 1: 开启 LDO; 0: 规划将关闭 LDO, 更省。	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	Reserved	RO	保留。	0
4	RB_PWR_RAM8K	RWA	RAM8K 的 SRAM 供电 (睡眠规划): 1: 双供电; 0: 不用辅助电源。	1
3	RB_PWR_EXTEND	RWA	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_RAM24K	RWA	RAM24K 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
---	-------------	-----	---	---

注：此寄存器为睡眠规划预置，其电源配置在进入低功耗的睡眠模式和下电模式后生效。

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。	00XXb
[11:8]	RB_CFG_IVREF	RWA	保留。写入时必须保持原值不变。	XXXXb
[7:5]	Reserved	RO	保留。	0
[4:3]	Reserved	RWA	保留。写入时必须保持原值不变。	XXb
[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）： 000：0.77V；001：0.80V； 010：0.83V；011：0.85V； 100：0.88V；101：0.91V； 110：0.95V；111：0.98V。	XXXb

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：关闭。	0
2	RB_BAT_LOW_NMI	RWA	电池低电压中断类型： 1：不可屏蔽中断 NMI； 0：外设中断。	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	Reserved	RO	保留。	0

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
[7:3]	Reserved	RO	保留。	0
[2:0]	RB_BAT_LOW_VTH	RWA	当 RB_BAT_MON_EN=1 时，以设置低功耗的 低压监控阈值： 000：1.06V；001：1.11V； 010：1.16V；011：1.21V； 100：1.26V；101：1.31V； 110：1.36V；111：1.41V。	0

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

			0: 否。	
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，LSE 或 LSI 可以选择是否开启，RAM8K、RAM24K 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 RF 配置，LSE 或 LSI 可以选择是否开启，RAM8K、RAM24K 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 中的保持数据区分于 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 I2C 或 SysTick 或 SPI 或 TMR 或 UART	主频 HSE=16MHz， FLASH 片选关闭， 外设时钟关闭	1.5mA
				主频 HSI=16MHz， FLASH 片选关闭， 外设时钟关闭	1.4mA
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	主频 HSE=16MHz， XT_FORCE_EN=1	1.02mA
				主频 HSI=16MHz， XT_FORCE_EN=1	0.83mA
				XT_FORCE_EN=0	180uA
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，LSE 或 LSI 可以选择是否开启，RAM8K、RAM24K 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。 芯片唤醒后会继续运行	详见表 5-3	1.8uA ~ 3.8uA
下电模式 Shut down	超低功耗 LDO 维持 PMU 供电，LSE 或 LSI 可以选择是否开启，RAM8K、RAM24K 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。 芯片唤醒后会 自动复位	详见表 5-3	0.85uA ~ 2.6uA

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	RAM8K	RAM24K	CK32K	CORE	EXTEND	功耗 ⁽¹⁾ (仅供参考)
------	--------	-------	--------	-------	------	--------	--------------------------

维持供电的功能	系统电源	数据区 8KB	数据区 24KB	LSE/LSI 二选一 RTC 唤醒	CPU 内核和 基本外设	RF 配置	PMU 和 RTC 寄存器 常供电
下电模式 常用配置	0	0	0	0	0	0	0.85uA
	0	1	0	0	0	0	1.6uA
	0	0	1	0	0	0	2.6uA
	0	0	0	1	0	0	1.1uA (LSI) 1.2uA (LSE)
	0	1	0	1	0	0	1.8uA (LSI) 1.9uA (LSE)
睡眠模式 常用配置	0	1	0	0	1	0	1.8uA
	0	1	0	1	1	0	2.1uA (LSI) 2.2uA (LSE)
	0	0	1	0	1	0	2.8uA
	0	0	1	1	1	0	3.1uA (LSI) 3.2uA (LSE)
	0	1	1	1	1	1	3.7uA (LSI) 3.8uA (LSE)

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

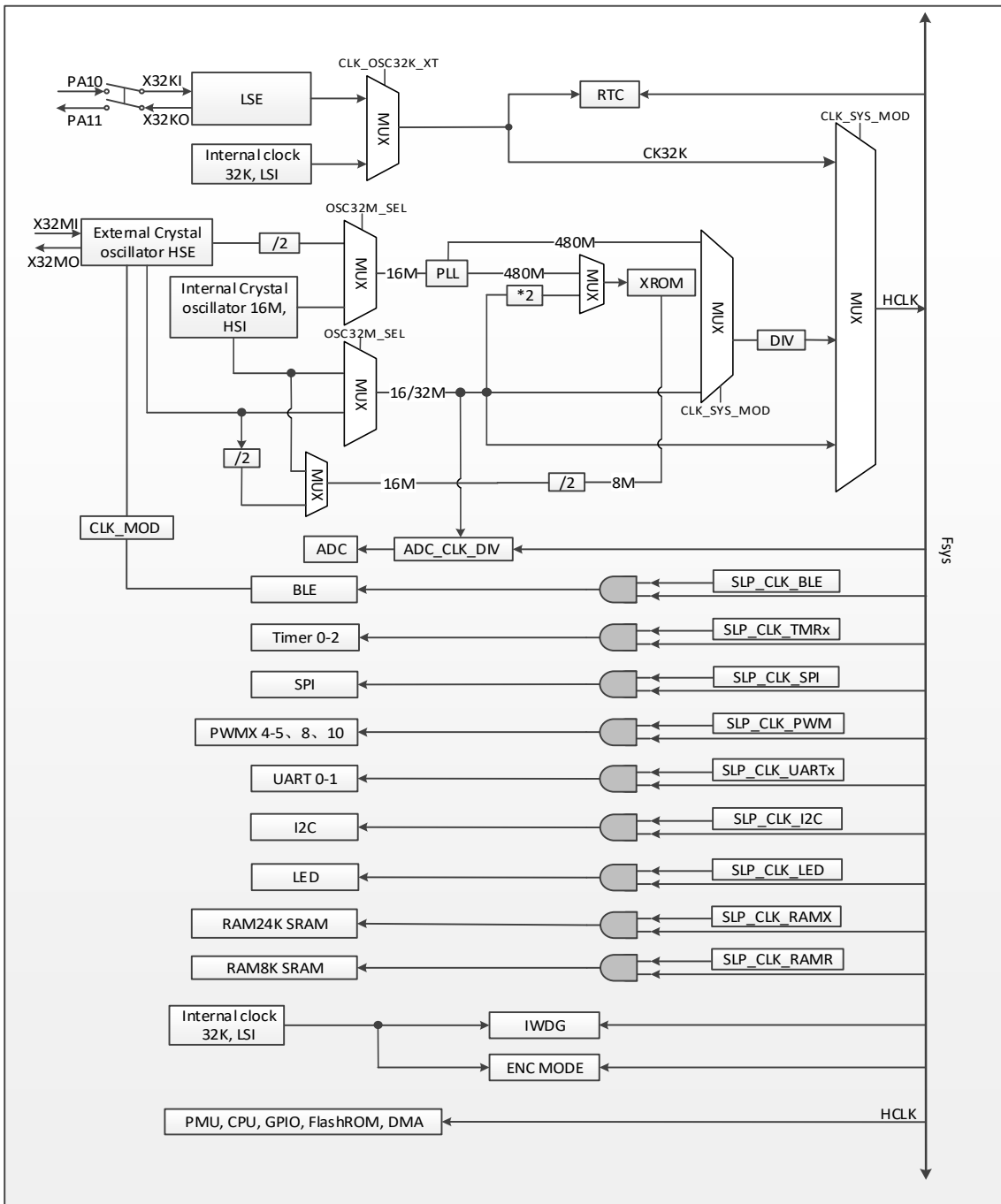
以下几种不同的时钟源可被选择来驱动系统时钟 HCLK (F_{sys}):

- HSE 或 HSI 之分频
- 内部 PLL (默认 480MHz) 分频
- HSE 原始时钟 CK32M 或 HSI 原始时钟 CK16M
- LSE 或 LSI 原始时钟 CK32K

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 32KHz 时钟源 CK32K，所以使用这些功能必须打开低频时钟。

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 CK32K 原始周期、2 秒周期、1 天周期的计数。

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
R16_CLK_SYS_CFG	0x40001008	系统时钟配置寄存器	0x0803
R8_HFCK_PWR_CTRL	0x4000100A	高频时钟模块电源控制寄存器	0x16
R16_INT32K_TUNE	0x4000102C	内部 32KHz 时钟校准寄存器	0x1011
R8_XT32K_TUNE	0x4000102E	32KHz 时钟谐振控制寄存器	0xC7
R8_CK32K_CONFIG	0x4000102F	32KHz 振荡器配置寄存器	0xX2
R8_XT32M_TUNE	0x4000104E	外部 32MHz 时钟谐振控制寄存器	0x03
R16_OSC_CAL_CNT	0x40001050	振荡器频率校准计数值寄存器	0xFFFF
R8_OSC_CAL_OV_CNT	0x40001052	振荡器频率校准溢出次数寄存器	0x00
R8_OSC_CAL_CTRL	0x40001053	振荡器频率校准控制寄存器	0x09
R8_PLL_CONFIG	0x4000104B	PLL 配置寄存器	0x0C
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_32K	0x40001038	RTC 基于 32768Hz 计数值寄存器	0xFFFF
R16_RTC_CNT_2S	0x4000103A	RTC 以 2S 为单位的计数值寄存器	0xFFFF
R32_RTC_CNT_DAY	0x4000103C	RTC 以天为单位的计数值寄存器	0x0000XXXX

系统时钟配置寄存器 (R16_CLK_SYS_CFG)

位	名称	访问	描述	复位值
[15:14]	Reserved	RO	保留。	0
13	RB_PLL_GATE_TIME	RWA	在切换 PLL 时钟来源时，关闭 PLL 时钟的时间选择： 1: 40us; 0: 30us。	0
12	RB_PLL_GATE_DISS	RWA	在切换 PLL 时钟来源时，是否关闭 PLL 时钟： 1: 不关闭; 0: 关闭。	0
11	RB_PLL_GATE_STATUS	RO	PLL 时钟的开启标志：	1

			1: PLL 时钟已开启; 0: PLL 时钟已关闭。	
10	Reserved	RO	保留。	0
9	RB_OSC32M_SEL	RWA	PLL 时钟源的选择: 1: 外部 32MHz 振荡器 HSE; 0: 内部 16MHz 振荡器 HSI。 注: 完成切换需要 1us。	0
8	Reserved	RO	保留。	0
[7:6]	RB_CLK_SYS_MOD	RWA	HCLK 系统时钟源模式选择: 00: CK32M 或 CK16M 进行分频; 01: PLL (默认 480MHz) 进行分频; 10: CK32M或CK16M作为HCLK(默认16MHz); 11: CK32K (默认 32KHz) 作为 HCLK。	0
5	Reserved	RO	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。	00011b

计算:

$F_{ck32m} = XT_32MHz$;

$F_{ck16m} = RC_16MHz$;

$F_{ck32k} = RB_CLK_OSC32K_XT ? XT_32KHz : RC_32KHz$;

$F_{pll} = F_{ck32m} * 15 = 480MHz$;

$F_{sys} = RB_CLK_SYS_MOD[1] ? (RB_CLK_SYS_MOD[0] ? F_{ck32k} : (RB_OSC32M_SEL ? F_{ck32m} : F_{ck16m})) :$

$((RB_CLK_SYS_MOD[0] ? F_{pll} / RB_CLK_PLL_DIV : (RB_OSC32M_SEL ? F_{ck32m} : F_{ck16m})) / RB_CLK_PLL_DIV)$;

上电默认值 $F_{sys} = F_{ck16m} / RB_CLK_PLL_DIV = 16MHz / 3 = 5.33MHz$;

F_{sys} 范围:

位 RB_CLK_SYS_MOD[1:0]	HCLK 系统时钟源模式	F_{sys} 范围
11	CK32K	32KHz (在 RAM 中运行)
00/10	CK32M或CK16M进行分频/ CK32M 或 CK16M	1MHz~32MHz
01	PLL 进行分频	15MHz~80MHz

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	RB_CLK_RC16M_PON	RWA	内部 16MHz 振荡器 HSI 的使能: 1: 开启; 0: 关闭。	1
0	Reserved	RO	保留。	0

内部 32KHz 时钟校准寄存器 (R16_INT32K_TUNE)

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

[15:13]	Reserved	RO	保留。	0
[12:0]	RB_INT32K_TUNE	RWA	内部 RC 32KHz 时钟频率校准值。	1011h

32KHz 时钟谐振控制寄存器 (R8_XT32K_TUNE)

位	名称	访问	描述	复位值
[7:4]	RB_XT32K_C_LOAD	RWA	选择与外部 32KHz 晶体匹配的内置负载电容（可能影响 RTC 时钟精度）： 电容量 = RB_XT32K_C_LOAD + 12pF， 0000b~1111b 分别对应约 12pF~27pF。 根据所用晶体参数选择。	1100b
[3:2]	RB_RC32K_I_TUNE	RWA	RC32K 调节位，内部 32KHz 振荡器偏置电流选择： 00：120%额定电流； 01：160%额定电流； 10：230%额定电流； 11：额定电流。 注：默认采用 11。	01b
[1:0]	RB_XT32K_I_TUNE	RWA	32KHz 振荡器偏置电流选择： 00：70%额定电流； 01：额定电流； 10：140%额定电流； 11：200%额定电流。 注：在晶体振荡器稳定后可改为额定电流。	11b

32KHz 振荡器配置寄存器 (R8_CK32K_CONFIG)

位	名称	访问	描述	复位值
7	RB_32K_CLK_PIN	RO	32KHz 时钟引脚状态（异步信号）。	X
[6:4]	Reserved	RO	保留。	0
3	RB_CLK_OSC32K_FILT	RWA	内部 32KHz 振荡器噪声过滤模式： 1：启用； 0：禁用。	0
2	RB_CLK_OSC32K_XT	RWA	CK32K（32KHz）时钟源选择位： 1：外部 32KHz 振荡器； 0：内部 32KHz 振荡器。	0
1	RB_CLK_INT32K_PON	RWA	内部 32KHz 振荡器电源控制位： 1：上电； 0：掉电。	1
0	RB_CLK_XT32K_PON	RWA	外部 32KHz 振荡器电源控制位： 1：上电； 0：掉电。	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，	0

			000b~111b 分别对应约 6pF~20pF。根据所用晶体参数选择，常用为 111b。	
[3:2]	Reserved	R0	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

振荡器频率校准计数值寄存器 (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	R8_OSC_CAL_OV_CNT 寄存器数值非零指示，写 1 将 R8_OSC_CAL_OV_CNT 清零。	0
[13:0]	RB_OSC_CAL_CNT	R0	对多个 CK32K 周期基于系统主频的计数值，用于校准内部 32KHz 振荡器频率。	XXXXh

振荡器频率校准溢出次数寄存器 (R8_OSC_CAL_OV_CNT)

位	名称	访问	描述	复位值
[7:0]	RB_OSC_CAL_OV_CNT	R0	振荡器频率校准计数溢出次数，向 RB_OSC_CAL_OV_CLR 写 1 清零此寄存器。	0

振荡器频率校准控制寄存器 (R8_OSC_CAL_CTRL)

位	名称	访问	描述	复位值
7	Reserved	R0	保留。	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	振荡器频率校准计数器使能位： 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	振荡器频率校准计数器计数状态位： 1: 正在暂停计数； 0: 正在计数中。	1
[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。	0Ch
-------	----------------	-----	-----------	-----

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	RO	RTC 触发模式激活标志。	0
6	RB_RTC_TMR_FLAG	RO	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	RO	保留。	0

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_DAY。	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_2S；将 R32_RTC_TRIG 低 16 位加载到 R16_RTC_CNT_32K。	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_BO	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： 000：0.125S；001：0.25S； 010：0.5S；011：1S； 100：2S；101：4S； 110：8S；111：16S。	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 位和低 16 位分别与 R16_RTC_CNT_2S 和 R16_RTC_CNT_32K 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。	0

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 32768Hz 计数值寄存器 (R16_RTC_CNT_32K)

位	名称	访问	描述	复位值
[15:0]	R16_RTC_CNT_32K	RO	RTC 基于 32768Hz 计数值寄存器。	XXXXh

RTC 以 2S 为单位的计数值寄存器 (R16_RTC_CNT_2S)

位	名称	访问	描述	复位值
[15:0]	R16_RTC_CNT_2S	RO	RTC 以 2S 为单位的当前计数值。	XXXXh

RTC 以天为单位的计数值寄存器 (R32_RTC_CNT_DAY)

位	名称	访问	描述	复位值
[31:14]	Reserved	RO	保留。	0
[13:0]	R32_RTC_CNT_DAY	RO	RTC 以天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_DAY 寄存器中；
- (2)、设置 R32_RTC_TRIG 寄存器数值，将 RB_RTC_LOAD_LO 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_2S 寄存器和 R16_RTC_CNT_32K 寄存器。

6.4.2 RTC 时钟源切换为 LSE 晶振

- (1)、确认 X32K1 和 X32K0 所在的 GPIO 引脚没有设置为输出，没有设置上拉和下拉电阻，只有晶体；
- (2)、配置 R8_CK32K_CONFIG 寄存器，置 RB_CLK_XT32K_PON 为 1，开启外部 32KHz 晶体振荡器；
- (3)、建议先置 RB_XT32K_I_TUNE 为最大，等待晶体振荡器稳定（约数百 ms）后再改为额定电流；
- (4)、配置 R8_CK32K_CONFIG 寄存器，置 RB_CLK_OSC32K_XT 为 1，要求时钟源切换到晶体振荡器；
- (5)、等待至少半个 32KHz 时钟周期，通常 16us，真正完成时钟源的切换。

6.4.3 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。

6.4.4 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_32K（高 16 位 R16_RTC_CNT_2S 和低 16 位 R16_RTC_CNT_32K）加上间隔时间 DelayTime（单位 S），计算出目标时间数值， $T32 = R32_RTC_CNT_32K + DelayTime * 32768$ ，将 T32 写入 R32_RTC_TRIG 寄存器中，完成匹配数值设定；
- (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_2S 和 R16_RTC_CNT_32K 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。

6.4.5 用 HSE 校准内部 32K 时钟 LSI

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 2 组 GPIO 端口 PA 和 PB，共 15 个通用输入输出引脚，其中 15 个引脚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_Px_DIR，一个 32 位引脚输入寄存器 R32_Px_PIN，一个 32 位数据输出寄存器 R32_Px_OUT，一个 32 位数据复位寄存器 R32_Px_CLR，一个 32 位上拉电阻配置寄存器 R32_Px_PU，一个 32 位下拉电阻/驱动能力配置寄存器 R32_Px_PD_DRV。

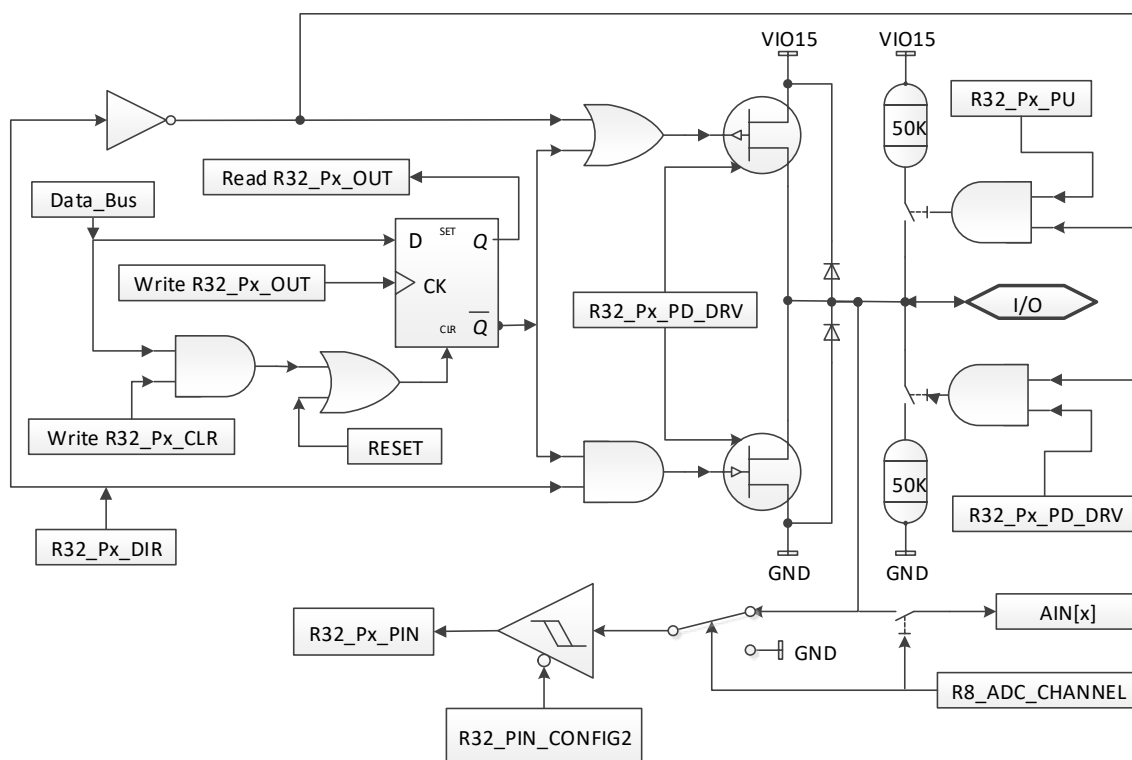
PA 端口中 PA[8]~PA[15]位有效，对应芯片上 8 个 GPIO 引脚。

PB 端口中 PB[6]、PB[10]~PB[15]位有效，对应芯片上 7 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

芯片的部分 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_Px_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。

例如：PA9 脚复用为 TMR0/TXD1/AIN13，则 TMR0 功能优先，AIN13 功能的优先级最低。这样可以在多个复用功能中，将功能优先级最低的部分不需使用的引脚启用相对更高优先级的复用功能。

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

TMR0/1/2 引脚	功能配置	GPIO 配置
TMRx	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 UARTx

UART0/1 引脚	功能配置	GPIO 配置
TXDx	串口发送 x	推挽输出
RxDx	串口接收 x	上拉输入（推荐）或浮空输入
DTR	MODEM 信号输出或 RS485 控制	推挽输出
DSR, RI, DCD	MODEM 信号输入	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
SCKx	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）
MOSIx	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
MISOx	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 ADC

ADC 采样通道引脚	功能配置	GPIO 配置
AINx	模拟数字转换输入通道	仅模拟输入

表 7-7 LED

LED 信号引脚	功能配置	GPIO 配置
----------	------	---------

LED[3:0]	LED 输出 bit3~bit0	推挽输出
LED_CLK	LED 输出时钟	推挽输出

表 7-8 ENCODER

ENCODER 信号引脚	功能配置	GPIO 配置
ENCCAP_IN1	编码器捕获通道 1	输入（浮空输入/上拉输入/下拉输入）
ENCCAP_IN2	编码器捕获通道 2	输入（浮空输入/上拉输入/下拉输入）

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和 R16_PIN_ANALOG_IE 外设模拟管脚配置寄存器将一些功能引脚重新映射到其他引脚上。

表 7-9 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/MOS1/MISO	PA12/PA13/PA14/PA15	PB12/PB13/PB14/PB15
RXD1/TXD1	PA8/PA9	PB12/PB13
RXD0/TXD0/DSR/DTR/R1/DCD	-/-/PB14/PB15/-/-	PA15/PA14/-/-/PB12/PB13
TMR2/PWM2/CAP2	PA11	PB11
TMR1/PWM1/CAP1	PA10	PB10
TMRO/PWMO/CAPO	PA9	-
PWM4/PWM5	PA12/PA13	-
PWM8/PWM10	PB6/PB14	-
SCL/SDA (I2C)	PB15/PB14	-
LED[3]/LED[2]/LED[1]/LED[0]/LED_CLK	PA11/PA12/PA13/PA14/PA15	PB11/PB12/PB13/PB14/PB15
ENCCAP_IN1/ENCCAP_IN2	PB10/PB11	PA10/PA11

7.4 寄存器描述

表 7-10 GPIO 相关寄存器列表

名称	访问地址	描述	复位值
R32_PIN_CONFIG1	0x40001018	管脚配置寄存器 1	0x00000000
R16_PIN_ALTERNATE	0x40001018	功能引脚重映射寄存器	0x0000
R16_PIN_ANALOG_IE	0x4000101A	外设模拟管脚配置寄存器	0x0000
R32_PIN_CONFIG2	0x4000101C	管脚配置寄存器 2	0x00000000
R16_PA_INT_EN	0x40001090	PA 端口中断使能寄存器	0x0000
R16_PB_INT_EN	0x40001092	PB 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_PB_INT_MODE	0x40001096	PB 端口中断模式配置寄存器	0x0000
R16_PA_INT_EDGE_TYPE	0x40001098	PA 端口中断边沿类型配置寄存器	0x0000
R16_PB_INT_EDGE_TYPE	0x4000109A	PB 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
R16_PB_INT_IF	0x4000109E	PB 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	0x0000XXXX
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
R32_PB_DIR	0x400010C0	PB 端口方向配置寄存器	0x00000000
R32_PB_PIN	0x400010C4	PB 端口引脚输入寄存器	0x00XXXXXX
R32_PB_OUT	0x400010C8	PB 端口数据输出寄存器	0x00000000
R32_PB_CLR	0x400010CC	PB 端口数据复位寄存器	0x00000000
R32_PB_PU	0x400010D0	PB 端口上拉电阻配置寄存器	0x00000000
R32_PB_PD_DRV	0x400010D4	PB 端口下拉/驱动配置寄存器	0x00000000
R32_PB_SET	0x400010D8	PB 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
[15:13]	Reserved	R0	保留。	0
12	RB_PIN_MODEM	RW	UART0 MODEM 功能引脚映射选择位： 1: RI/DCD 映射到 PB[12]/PB[13]； 0: 保留。	0
[11:9]	Reserved	R0	保留。	0
8	RB_PIN_SPI0	RW	SPI0 功能引脚映射选择位： 1: SCK0_/SCS_/MOSI_/MISO_映射到 PB[13]/PB[12]/PB[14]/PB[15]； 0: SCK0/SCS/MOSI/MISO 映射到 PA[13]/PA[12]/PA[14]/PA[15]。	0
[7:6]	Reserved	R0	保留。	0
5	RB_PIN_UART1	RW	UART1 功能引脚映射选择位： 1: RXD1_/TXD1_映射到 PB[12]/PB[13]； 0: RXD1/TXD1 映射到 PA[8]/PA[9]。	0
4	RB_PIN_UART0	RW	UART0 功能引脚映射选择位： 1: RXD0_/TXD0_映射到 PA[15]/PA[14]； 0: 保留。	0
3	Reserved	R0	保留。	0
2	RB_PIN_TMR2	RW	TMR2 功能引脚映射选择位： 1: TMR2_/PWM2_/CAP2_映射到 PB[11]； 0: TMR2/PWM2/CAP2 映射到 PA[11]。	0
1	RB_PIN_TMR1	RW	TMR1 功能引脚映射选择位： 1: TMR1_/PWM1_/CAP1_映射到 PB[10]； 0: TMR1/PWM1/CAP1 映射到 PA[10]。	0
0	Reserved	R0	保留。	0

外设模拟管脚配置寄存器 (R16_PIN_ANALOG_IE)

位	名称	访问	描述	复位值
[15:6]	Reserved	R0	保留。	0
5	RB_PIN_LED	RW	LED 引脚映射： 1： LED[3]_/LED[2]_/LED[1]_/LED[0]_/LED_/_ CLK_映射到 PB[11]/PB[12]/PB[13]/PB[14]/PB[15]； 0： LED[3]/LED[2]/LED[1]/LED[0]/LED_CLK 映射到	0

			PA[11]/PA[12]/PA[13]/PA[14]/PA[15]。	
4	RB_PIN_ENC	RW	编码器功能引脚映射选择位： 1： ENCCAP_IN1_/ENCCAP_IN2_ 映射到 PA[10]/PA[11]； 0： CAP_IN1/CAP_IN2 映射到 PB[10]/PB[11]。	0
[3:0]	Reserved	R0	保留。	0

注：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。

管脚配置寄存器 2 (R32_PIN_CONFIG2)

位	名称	访问	描述	复位值
[31:26]	RB_PIN_PB_DIS[10:15]	RW	PB10–PB15 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
[25:23]	Reserved	R0	保留。	0
22	RB_PIN_PB_DIS[6]	RW	PB6 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
[21:16]	Reserved	R0	保留。	0
[15:8]	RB_PIN_PA_DIS[8:15]	RW	PA8–PA15 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
[7:0]	Reserved	R0	保留。	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。	0

PB 端口中断使能寄存器 (R16_PB_INT_EN)

位	名称	访问	描述	复位值
[15:0]	R16_PB_INT_EN	RW	PB 引脚中断使能位： 1： 使能相应中断； 0： 禁止相应中断。	0

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。	0

PB 端口中断模式配置寄存器 (R16_PB_INT_MODE)

位	名称	访问	描述	复位值
[15:0]	R16_PB_INT_MODE	RW	PB 引脚中断模式选择位： 1： 边沿触发； 0： 电平触发。	0

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。	0

PB 端口中断边沿类型配置寄存器 (R16_PB_INT_EDGE_TYPE)

位	名称	访问	描述	复位值
[15:0]	R16_PB_INT_EDGE_TYPE	RW	PB 引脚中断边沿类型配置： 1：不遵照极性； 0：遵照极性。	0

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。	0

PB 端口中断标志寄存器 (R16_PB_INT_IF)

位	名称	访问	描述	复位值
[15:0]	R16_PB_INT_IF	RW1	PB 引脚中断标志位，写 1 清零： 1：有中断； 0：无中断。	0

PA 端口方向配置寄存器 (R32_PA_DIR)

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:8]	R8_PA_DIR_1	RW	PA 引脚当前输入输出方向配置： 1：引脚为输出模式； 0：引脚为输入模式。	0
[7:0]	R8_PA_DIR_0	RW		0

PA 端口引脚输入寄存器 (R32_PA_PIN)

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:8]	R8_PA_PIN_1	RO	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）： 1：引脚输入高电平； 0：引脚输入低电平。	XXh
[7:0]	R8_PA_PIN_0	RO		XXh

PA 端口数据输出寄存器 (R32_PA_OUT)

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:8]	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		0

			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。	
--	--	--	---	--

PA 端口数据复位寄存器 (R32_PA_CLR)

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:8]	R8_PA_CLR_1	WZ	PA 数据寄存器复位控制：	0
[7:0]	R8_PA_CLR_0	WZ	1：R32_PA_OUT 对应位数据清 0； 0：无影响。	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
[31:16]	Reserved	RO	保留。	0
[15:8]	R8_PA_PU_1	RW	PA 引脚上拉电阻使能控制：	0
[7:0]	R8_PA_PU_0	RW	1：启用上拉电阻； 0：关闭上拉电阻。	0

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
[31:16]	Reserved	RO	保留。	0
[15:8]	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
[31:16]	Reserved	RO	保留。	0
[15:8]	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 高电平。	0
[7:0]	R8_PA_SET_0	WZ		0

PB 端口方向配置寄存器 (R32_PB_DIR)

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:16]	R8_PB_DIR_2	RW	PB 引脚当前输入输出方向配置：	0
[15:8]	R8_PB_DIR_1	RW	1：引脚为输出模式；	0
[7:0]	R8_PB_DIR_0	RW	0：引脚为输入模式。	0

PB 端口引脚输入寄存器 (R32_PB_PIN)

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0

[23:16]	R8_PB_PIN_2	RO	PB 引脚当前电平状态（仅在 R32_PB_DIR 对应位为 0 时，该位值有效）： 1：引脚输入高电平； 0：引脚输入低电平。	XXh
[15:8]	R8_PB_PIN_1	RO		XXh
[7:0]	R8_PB_PIN_0	RO		XXh

PB 端口数据输出寄存器 (R32_PB_OUT)

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:16]	R8_PB_OUT_2	RW	方向寄存器 R32_PB_DIR 对应位为 1 时： 控制 PB 引脚输出电平状态： 1：输出高电平； 0：输出低电平。	0
[15:8]	R8_PB_OUT_1	RW	方向寄存器 R32_PB_DIR 对应位为 1 时： 控制 PB 引脚输出电平状态： 1：输出高电平； 0：输出低电平。	0
[7:0]	R8_PB_OUT_0	RW	方向寄存器 R32_PB_DIR 对应位为 0 时： 控制 PB 引脚中断极性选择： 1：高电平/上升沿； 0：低电平/下降沿。	0

PB 端口数据复位寄存器 (R32_PB_CLR)

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:16]	R8_PB_CLR_2	WZ	PB 数据寄存器复位控制： 1：R32_PB_OUT 对应位数据清 0； 0：无影响。	0
[15:8]	R8_PB_CLR_1	WZ		0
[7:0]	R8_PB_CLR_0	WZ		0

PB 端口上拉电阻配置寄存器 (R32_PB_PU)

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:16]	R8_PB_PU_2	RW	PB 引脚上拉电阻使能控制： 1：启用上拉电阻； 0：关闭上拉电阻。	0
[15:8]	R8_PB_PU_1	RW		0
[7:0]	R8_PB_PU_0	RW		0

PB 端口下拉/驱动配置寄存器 (R32_PB_PD_DRV)

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:16]	R8_PB_PD_DRV_2	RW	方向寄存器 R32_PB_DIR 对应位为 0 时： PB 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
[15:8]	R8_PB_PD_DRV_1	RW		0
[7:0]	R8_PB_PD_DRV_0	RW	方向寄存器 R32_PB_DIR 对应位为 1 时： PB 引脚电流驱动能力选择： 1：20mA 级别； 0：5mA 级别。	0

PB 端口输出置位寄存器 (R32_PB_SET)

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:16]	R8_PB_SET_2	WZ	置位寄存器 R32_PB_SET 对应位为 0 时，PB 引脚输出保持；为 1 时，PB 引脚输出高电平。	0
[15:8]	R8_PB_SET_1	WZ		0
[7:0]	R8_PB_SET_0	WZ		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 7-11 端口配置表

模式	R32_Px_DIR	R32_Px_PU	R32_PIN_CONFIG2	R32_Px_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 TMRx

8.1 TMRx 简介

芯片提供了 3 个 26 位定时器，TMR0、TMR1 和 TMR2，最长定时时间为 2^{26} 个时钟周期。且适用于多种场合，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），另外 TMR1 和 TMR2 支持 DMA 功能。每个定时器都是完全独立的，可以一起同步操作。

8.1.1 主要特性

- 3 个 26 位定时器，每个定时器定时时间最大为 2^{26} 个时钟周期。
- 支持定时器中断，其中 TMR1 和 TMR2 支持 DMA 及中断。
- 支持捕捉功能，测量输入脉冲长度或周期。
- 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 TMR0 相关寄存器列表

名称	访问地址	描述	复位值
R8_TMR0_CTRL_MOD	0x40002000	模式设置寄存器	0x02
R8_TMR0_INTER_EN	0x40002002	中断使能寄存器	0x00
R8_TMR0_INT_FLAG	0x40002006	中断标志寄存器	0x00
R8_TMR0_FIFO_COUNT	0x40002007	FIFO 计数寄存器	0x0X
R32_TMR0_COUNT	0x40002008	当前计数值寄存器	0x0XXXXXXX
R32_TMR0_CNT_END	0x4000200C	计数终值设置寄存器	0x0XXXXXXX
R32_TMR0_FIFO	0x40002010	FIFO 寄存器	0x0XXXXXXX

表 8-2 TMR1 相关寄存器列表

名称	访问地址	描述	复位值
R8_TMR1_CTRL_MOD	0x40002400	模式设置寄存器	0x02
R8_TMR1_CTRL_DMA	0x40002401	DMA 控制寄存器	0x00
R8_TMR1_INTER_EN	0x40002402	中断使能寄存器	0x00
R8_TMR1_INT_FLAG	0x40002406	中断标志寄存器	0x00
R8_TMR1_FIFO_COUNT	0x40002407	FIFO 计数寄存器	0x0X
R32_TMR1_COUNT	0x40002408	当前计数值寄存器	0x0XXXXXXX
R32_TMR1_CNT_END	0x4000240C	计数终值寄存器	0x0XXXXXXX
R32_TMR1_FIFO	0x40002410	FIFO 寄存器	0x0XXXXXXX
R16_TMR1_DMA_NOW	0x40002414	DMA 当前缓冲区地址	0xXXXX
R16_TMR1_DMA_BEG	0x40002418	DMA 起始缓冲区地址	0xXXXX
R16_TMR1_DMA_END	0x4000241C	DMA 结束缓冲区地址	0xXXXX

表 8-3 TMR2 相关寄存器列表

名称	访问地址	描述	复位值
R8_TMR2_CTRL_MOD	0x40002800	模式设置寄存器	0x02
R8_TMR2_CTRL_DMA	0x40002801	DMA 控制寄存器	0x00
R8_TMR2_INTER_EN	0x40002802	中断使能寄存器	0x00
R8_TMR2_INT_FLAG	0x40002806	中断标志寄存器	0x00
R8_TMR2_FIFO_COUNT	0x40002807	FIFO 计数寄存器	0x0X

R32_TMR2_COUNT	0x40002808	当前计数值寄存器	0x0XXXXXXX
R32_TMR2_CNT_END	0x4000280C	计数终值寄存器	0x0XXXXXXX
R32_TMR2_FIFO	0x40002810	FIFO 寄存器	0x0XXXXXXX
R16_TMR2_DMA_NOW	0x40002814	DMA 当前缓冲区地址	0xXXXX
R16_TMR2_DMA_BEG	0x40002818	DMA 起始缓冲区地址	0xXXXX
R16_TMR2_DMA_END	0x4000281C	DMA 结束缓冲区地址	0xXXXX

模式设置寄存器 (R8_TMRx_CTRL_MOD) (x=0/1/2)

位	名称	访问	描述	复位值
[7:6]	RB_TMR_CAP_EDGE	RW	捕捉模式下，选择捕捉触发方式： 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。 计数模式下，选择计数的边沿： 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； 11：采样到上升沿计数。	0
[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。	0
5	RB_TMR_PWM_REPEAT_EN	RW	PWM 模式重复发送时，FIFO 为空且写入新数据时，切换数据的方式： 1：完成当前重复次数后切换到新数据； 0：立即切换到新数据。	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_TMRx_INTER_EN) (x=0/1/2)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
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（仅 TMR1/2 支持）： 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>=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 (RB_TMRx_INT_FLAG) (x=0/1/2)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
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（仅 TMR1/2 支持）： 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_TMRx_FIFO_COUNT) (x=0/1/2)

位	名称	访问	描述	复位值
[7:4]	Reserved	R0	保留。	0
[3:0]	R8_TMRx_FIFO_COUNT	R0	FIFO 内数据计数, 最大值 8。	0Xh

当前计数值寄存器 (R32_TMRx_COUNT) (x=0/1/2)

位	名称	访问	描述	复位值
[31:26]	Reserved	R0	保留。	0Xh
[25:0]	R32_TMRx_COUNT	R0	计数器当前计数值。	XXXXXXh

计数终值设置寄存器 (R32_TMRx_CNT_END) (x=0/1/2)

位	名称	访问	描述	复位值
[31:0]	R32_TMRx_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)。 <i>注: 对此寄存器的写操作, 将自动清零 R32_TMRx_COUNT 寄存器中的值。</i>	0XXXXXXh

FIFO 寄存器 (R32_TMRx_FIFO) (x=0/1/2)

位	名称	访问	描述	复位值
[31:0]	R32_TMRx_FIFO	RO/WO	FIFO 数据寄存器, 仅低 26 位有效。	0XXXXXXh

DMA 控制寄存器 (R8_TMRx_CTRL_DMA) (x=1/2) (仅 TMR1/2 支持)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
4	RB_HI_PWM_EN	RW	160M 高频 PWM 模式使能: 1: 开启高频 PWM 模式; 0: 禁止高频 PWM 模式。	0
3	Reserved	R0	保留。	0
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	R0	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_TMRx_DMA_NOW) (x=1/2)

位	名称	访问	描述	复位值
15	Reserved	R0	保留。	0
[14:2]	R16_TMRx_DMA_NOW	R0	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	R0	保留。	0
-------	----------	----	-----	---

DMA 起始缓冲区地址(R16_TMRx_DMA_BEG) (x=1/2)

位	名称	访问	描述	复位值
15	Reserved	R0	保留。	0
[14:2]	R16_TMRx_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	R0	保留。	0

DMA 结束缓冲区地址(R16_TMRx_DMA_END) (x=1/2)

位	名称	访问	描述	复位值
15	Reserved	R0	保留。	0
[14:2]	R16_TMRx_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	R0	保留。	0

8.3 功能描述及配置

8.3.1 定时、计数功能

芯片的每个定时器都支持最长定时时间 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： $31.25\text{nS} \times 2^{26} \approx 2\text{S}$ 。每个定时器都有独立的中断。

定时功能操作步骤如下：

- (1)、将 RB_TMR_ALL_CLEAR 置位，清零 R32_TMRx_COUNT 和中断标志等；
- (2)、设置寄存器 R32_TMRx_CNT_END 为需要定时的时间值；
具体计算方法为： $\text{Time} = T_{\text{sys}} * R32_TMRx_CNT_END$ ；
- (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_TMRx_INTER_EN 寄存器，置位 RB_TMR_IE_CYC_END 可打开定时周期中断；
- (5)、将 R8_TMRx_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_TMRx_COUNT 等于 R32_TMRx_CNT_END 时，定时完成。此时 R8_TMRx_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_TMRx_CNT_END 中设定计数溢出终值；
- (3)、配置 R8_TMRx_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_TMRx_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_TMRx_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_TMRx_COUNT 清 0，如果开启中断则会触发硬件中断。

8.3.2 PWM 功能

芯片的每个定时器均具有 PWM 功能。其中 TMR1 和 TMR2 的 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_TMRx_FIFO 和中断标志等；
- (2)、设置 PWM 总周期寄存器 R32_TMRx_CNT_END，该值应该不小于 R32_TMRx_FIFO 寄存器中的值；
- (3)、配置 R8_TMRx_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；若要开启 160M 高频 PWM，需将 RB_HI_PWM_EN 置 1；

(4)、设置数据寄存器 R32_TMRx_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： $R32_TMRx_FIFO/R32_TMRx_CNT_END$ 。TMR1 和 TMR2 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_TMRx_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；

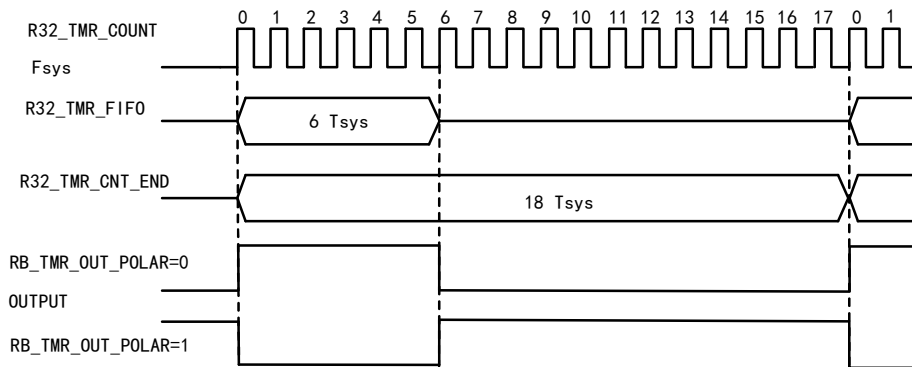
(7)、可选步骤，如果需要启用中断则设置相应的中断使能寄存器位；

(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_TMRx_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_TMRx_FIFO 为 6，R32_TMRx_CNT_END 为 18，则产生 PWM 的基本时序图如下所示，其占空比为： $R32_TMRx_FIFO/R32_TMRx_CNT_END = 1/3$ 。

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

芯片的每个定时器都具备捕捉功能，其中 TMR1 和 TMR2 的捕捉功能支持 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

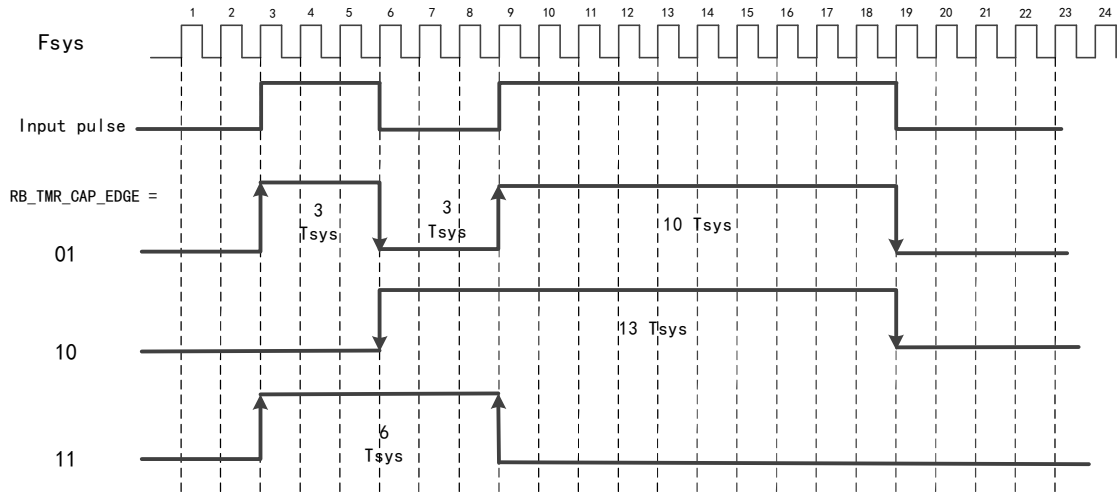
捕捉模式选择位 RB_TMR_CATCH_EDGE	触发方式	图示
00	不捕捉	无
01	边沿触发 边沿至边沿	
10	下降沿至下降沿	
11	上升沿至上升沿	

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_TMRx_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_TMRx_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_TMRx_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_TMRx_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_TMRx_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_TMRx_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_TMRx_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_TMRx_DMA_BEG 为存储捕捉数据缓冲区的首地址，设置寄存器 R16_TMRx_DMA_END 为存储捕捉数据缓冲区的结束地址（不含），并设置 R8_TMRx_CTRL_DMA 的 RB_TMR_DMA_ENABLE 为 1，使能 DMA 功能；
- (7)、将 R8_TMRx_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_TMRx_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 DMA 功能

芯片的定时器 TMR1 和 TMR2 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_TMR1_DMA_NOW，将 R16_TMR1_DMA_END 赋值为任一不和 R16_TMR1_DMA_NOW 相等的值（例：可以取值为 R16_TMR1_DMA_NOW + 0x100）；
- (2)、将 R8_TMR1_INT_FLAG 的位 RB_TMR_IF_DMA_END 写 1 清零；
- (3)、将 R8_TMR1_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_TMR1_DMA_NOW，将 R16_TMR1_DMA_END 赋值为任一不和 R16_TMR1_DMA_NOW 相等的值（例：可以取值为 R16_TMR1_DMA_NOW + 0x100）；
- (2)、将 R8_TMR1_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_TMR1_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

芯片提供了 2 组全双工的异步串口 UART0 和 UART1。支持全双工和半双工串口通讯，其中 UART0 提供发送状态引脚用于切换 RS485，并且支持 MODEM 调制解调器信号 DSR、RI、DCD、DTR。

9.1.1 主要特性

- 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 6Mbps 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。
- UART0 支持 MODEM 调制解调器信号 DSR、RI、DCD、DTR。
- 支持串口帧错误检测、支持 Break 线路间隔检测。
- 支持全双工和半双工串口通讯，UART0 提供发送状态引脚用于切换 RS485。

9.2 寄存器描述

表 9-1 UART0 相关寄存器列表

名称	访问地址	描述	复位值
R8_UART0_MCR	0x40003000	调制解调器 MODEM 控制寄存器	0x00
R8_UART0_IER	0x40003001	中断使能寄存器	0x00
R8_UART0_FCR	0x40003002	FIFO 控制寄存器	0x00
R8_UART0_LCR	0x40003003	线路控制寄存器	0x00
R8_UART0_IIR	0x40003004	中断识别寄存器	0x01
R8_UART0_LSR	0x40003005	线路状态寄存器	0x60
R8_UART0_MSR	0x40003006	调制解调器 MODEM 状态寄存器	0xX0
R8_UART0_TXE	0x40003007	发送完成寄存器	0x00
R8_UART0_RBR	0x40003008	接收缓冲寄存器	0xXX
R8_UART0_THR	0x40003008	发送保持寄存器	0xXX
R8_UART0_RFC	0x4000300A	接收 FIFO 计数寄存器	0x00
R8_UART0_TFC	0x4000300B	发送 FIFO 计数寄存器	0x00
R16_UART0_DL	0x4000300C	波特率除数锁存器	0xFFFF
R8_UART0_DIV	0x4000300E	预分频除数寄存器	0xXX
R8_UART0_ADR	0x4000300F	从机地址寄存器	0xFF

表 9-2 UART1 相关寄存器列表

名称	访问地址	描述	复位值
R8_UART1_MCR	0x40003400	调制解调器 MODEM 控制寄存器	0x00
R8_UART1_IER	0x40003401	中断使能寄存器	0x00
R8_UART1_FCR	0x40003402	FIFO 控制寄存器	0x00
R8_UART1_LCR	0x40003403	线路控制寄存器	0x00
R8_UART1_IIR	0x40003404	中断识别寄存器	0x01
R8_UART1_LSR	0x40003405	线路状态寄存器	0x60
R8_UART1_TXE	0x40003407	发送完成寄存器	0x00
R8_UART1_RBR	0x40003408	接收缓冲寄存器	0xXX
R8_UART1_THR	0x40003408	发送保持寄存器	0xXX
R8_UART1_RFC	0x4000340A	接收 FIFO 计数寄存器	0x00

R8_UART1_TFC	0x4000340B	发送 FIFO 计数寄存器	0x00
R16_UART1_DL	0x4000340C	波特率除数锁存器	0xXXXX
R8_UART1_DIV	0x4000340E	预分频除数寄存器	0xXX

调制解调器 MODEM 控制寄存器 (R8_UARTx_MCR) (x=0/1)

位	名称	访问	描述	复位值
7	RB_MCR_HALF	RW	半双工收发模式控制 (仅 UART0 支持): 1: 进入半双工收发模式, 发送优先, 不发送时为接收; 0: 禁止半双工模式。	0
6	RB_MCR_TNOW	RW	DTR 引脚输出正在发送状态使能 (仅 UART0 支持): 1: 将正在发送指示状态输出到 DTR 引脚, 可以用于控制 RS485 收发切换; 0: DTR 引脚为正常功能。	0
5	Reserved	RO	保留。	0
4	RB_MCR_LOOP	RW	内部回路的测试模式使能 (仅 UART0 支持): 1: 使能内部回路的测试模式; 0: 禁止内部回路的测试模式。 在内部回路的测试模式下, 串口所有对外输出引脚均为无效状态, TXD 内部返回到 RXD, DTR 内部返回到 DSR, OUT1 内部返回到 RI, OUT2 内部返回到 DCD。	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	RB_MCR_OUT1	RW	用户自定义 MODEM 控制位 (仅 UART0 支持), 没有连接实际输出引脚: 1: 置高; 0: 置低。	0
1	Reserved	RO	保留。	0
0	RB_MCR_DTR	RW	DTR 信号输出电平控制 (仅 UART0 支持): 1: DTR 信号输出有效 (低电平); 0: DTR 信号输出高电平 (默认)。	0

中断使能寄存器 (R8_UARTx_IER) (x=0/1)

位	名称	访问	描述	复位值
7	RB_IER_RESET	WZ	串口软件复位控制位, 自动清零: 1: 软件复位该串口; 0: 正常工作。	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	Reserved	RO	保留。	0
4	RB_IER_DTR_EN	RW	DTR 引脚输出使能位 (仅 UART0 支持): 1: 使能引脚输出; 0: 禁止引脚输出。	0

3	RB_IER_MODEM_CHG	RW	调制解调器输入状态变化中断使能位 (仅 UART0 支持): 1: 使能中断; 0: 禁止中断。	0
2	RB_IER_LINE_STAT	RW	接收线路状态中断使能位: 1: 使能中断; 0: 禁止中断。	0
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_UARTx_FCR) (x=0/1)

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。	0
[5:3]	Reserved	RO	保留。	0
2	RB_FCR_TX_FIFO_CLR	WZ	发送 FIFO 数据清空使能位, 自动清零: 1: 清空发送 FIFO 的数据 (不含 TSR); 0: 不清空发送 FIFO 的数据。	0
1	RB_FCR_RX_FIFO_CLR	WZ	接收 FIFO 数据清空使能位, 自动清零: 1: 清空接收 FIFO 的数据 (不含 RSR); 0: 不清空接收 FIFO 的数据。	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_UARTx_LCR) (x=0/1)

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;	0

			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。	
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: 1: 两个停止位; 0: 一个停止位。	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。	0

中断识别寄存器 (R8_UARTx_IIR) (x=0/1)

位	名称	访问	描述	复位值
[7:6]	RB_IIR_FIFO_ID	RO	串口 FIFO 启用状态位: 11: FIFO 已启用; 00: FIFO 未启用。	0
[5:4]	Reserved	RO	保留。	0
[3:1]	RB_IIR_INT_MASK	RO	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。	0
0	RB_IIR_NO_INT	RO	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_UARTx_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 9-3 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	总线地址匹配	接收到 1 个数据是串口总线地址, 且该地址与预置从机值相匹配或是广播地址。(仅 UART0 支持)	读 IIR 或禁用多机模式
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	MODEM 输入变化	△DSR、△RI、△DCD 置 1 触发。	读 MSR

线路状态寄存器 (R8_UARTx_LSR) (x=0/1)

位	名称	访问	描述	复位值
7	RB_LSR_ERR_RX_FIFO	RO	接收 FIFO 错误标志位: 1: 接收 FIFO 中存在至少一个 PAR_ERR、	0

			FRAM_ERR 或 BREAK_ERR 错误; 0: 接收 FIFO 中不存在错误。	
6	RB_LSR_TX_ALL_EMP	RO	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	RO	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

调制解调器 MODEM 状态寄存器 (R8_UART0_MSR) (仅 UART0 支持)

位	名称	访问	描述	复位值
7	RB_MSR_DCD	RO	DCD 引脚状态位: 1: DCD 有效 (低电平); 0: DCD 无效 (高电平)。	0
6	RB_MSR_RI	RO	RI 引脚状态位: 1: RI 有效 (低电平); 0: RI 无效 (高电平)。	0
5	RB_MSR_DSR	RO	DSR 引脚状态位: 1: DSR 引脚有效 (低电平); 0: DSR 引脚无效 (高电平)。	X
4	Reserved	RO	保留。	0
3	RB_MSR_DCD_CHG	RZ	DCD 引脚输入状态变化标志位: 1: 发生过变化; 0: 无变化。	0
2	RB_MSR_RI_CHG	RZ	RI 引脚输入状态变化标志位: 1: 发生过变化; 0: 无变化。	0
1	RB_MSR_DSR_CHG	RZ	DSR 引脚输入状态变化标志位: 1: 发生过变化; 0: 无变化。	0

0	Reserved	R0	保留。	0
---	----------	----	-----	---

发送完成寄存器 (R8_UARTx_TXE) (x=0/1)

位	名称	访问	描述	复位值
7	RB_UART_TRAN_IF	RW1	FIFO 中数据传输完成标志位： 1：已传输完成； 0：未传输完成。	0
[6:1]	Reserved	R0	保留。	0
0	RB_UART_TRAN_IE	RW	FIFO 中数据传输完成的中断使能： 1：使能中断； 0：禁止中断。	0

接收缓冲寄存器 (R8_UARTx_RBR) (x=0/1)

位	名称	访问	描述	复位值
[7:0]	R8_UARTx_RBR	R0	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_UARTx_THR) (x=0/1)

位	名称	访问	描述	复位值
[7:0]	R8_UARTx_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_UARTx_RFC) (x=0/1)

位	名称	访问	描述	复位值
[7:0]	R8_UARTx_RFC	R0	当前接收 FIFO 中数据计数。	0

发送 FIFO 计数寄存器 (R8_UARTx_TFC) (x=0/1)

位	名称	访问	描述	复位值
[7:0]	R8_UARTx_TFC	R0	当前发送 FIFO 中数据计数。	0

波特率除数锁存器 (R16_UARTx_DL) (x=0/1)

位	名称	访问	描述	复位值
[15:0]	R16_UARTx_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, 则除数 = 1843200/16/9600 = 12。	XXXXh

预分频除数寄存器 (R8_UARTx_DIV) (x=0/1)

位	名称	访问	描述	复位值
[7:0]	R8_UARTx_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

从机地址寄存器(R8_UART0_ADR)（仅 UART0 支持）

位	名称	访问	描述	复位值
[7:0]	R8_UART0_ADR	RW	串口 0 多机通讯时的从机地址： FFh：不使用； 其他：从机地址。	FFh

R8_UART0_ADR 预置本机作为从机时的地址，用于在多机通讯时自动比较接收到的地址，并在地址匹配或者在接收到广播地址 0FFh 时产生中断，同时允许接收后续数据包。在地址没有匹配之前不接收任何数据，开始发送数据后或者重写 R8_UART0_ADR 寄存器后停止接收任何数据，直到下次地址再次匹配或者接收到广播地址时再允许接收。

R8_UART0_ADR 为 0FFh 时或者 RB_LCR_PAR_EN=0 时，禁用总线地址自动比较功能。

R8_UART0_ADR 不为 0FFh 并且 RB_LCR_PAR_EN=1 时，启用总线地址自动比较功能，同时应该配置下述参数：RB_LCR_WORD_SZ 为 11b 以选择 8 个数据位方式，对于地址字节为 MARK 的情况（即数据字节的位 9 为 0），应设置 RB_LCR_PAR_MOD 为 10b，对于地址字节为 SPACE 的情况（即数据字节的位 9 为 1），应设置 RB_LCR_PAR_MOD 为 11b。

9.3 功能描述及配置

UART0/1 输出引脚都是 1.5V LVCMOS 电平。异步串口方式下引脚包括：数据传输引脚和 MODEM 联络信号引脚（仅 UART0 支持）。数据传输引脚包括：TXD 引脚和 RXD 引脚，默认都是高电平；MODEM 联络信号引脚包括：DSR 引脚、RI 引脚、DCD 引脚、DTR 引脚，默认都是高电平。所有这些 MODEM 联络信号都可以作为通用 I/O 引脚，由应用程序控制并定义其用途。

2 组 UART 各自拥有独立的收发缓冲区及 8 字节 FIFO，支持单工、半双工或者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_UARTx_DIV 寄存器，最大值 127，通常写入 1。

2) 计算波特率，设置 R16_UARTx_DL 寄存器。

波特率公式 = Fsys * 2 / R8_UARTx_DIV / 16 / R16_UARTx_DL。

9.3.2 串口发送

串口发送的“THR 寄存器空”中断 UART_II_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_UARTx_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_UARTx_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_UARTx_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_UARTx_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

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_SPI0_CTRL_MOD	0x40004000	SPI0 模式控制寄存器	0x02
R8_SPI0_CTRL_CFG	0x40004001	SPI0 配置寄存器	0x00
R8_SPI0_INTER_EN	0x40004002	SPI0 中断使能寄存器	0x00
R8_SPI0_CLOCK_DIV R8_SPI0_SLAVE_PRE	0x40004003	SPI0 主机模式时钟分频寄存器 SPI0 从机模式预置数据寄存器	0x10
R8_SPI0_BUFFER	0x40004004	SPI0 数据缓冲区	0xFF
R8_SPI0_RUN_FLAG	0x40004005	SPI0 工作状态寄存器	0x00
R8_SPI0_INT_FLAG	0x40004006	SPI0 中断标志寄存器	0x40
R8_SPI0_FIFO_COUNT	0x40004007	SPI0 收发 FIFO 计数寄存器	0x00
R8_SPI0_INT_TYPE	0x40004008	SPI0 中断触发方式选择寄存器	0x00
R8_SPI0_INTER1_EN	0x40004009	SPI0 中断 1 使能寄存器	0x00
R8_SPI0_INT1_FLAG	0x4000400A	SPI0 中断 1 标志寄存器	0x00
R16_SPI0_TOTAL_CNT	0x4000400C	SPI0 收发数据总长度寄存器	0x0000
R8_SPI0_FIFO	0x40004010	SPI0 数据 FIFO 寄存器	0xFF
R8_SPI0_FIFO_COUNT1	0x40004013	SPI0 收发 FIFO 计数寄存器 1	0x00
R16_SPI0_DMA_NOW	0x40004014	SPI0 DMA 缓冲区当前地址	0xFFFF
R16_SPI0_DMA_BEG	0x40004018	SPI0 DMA 缓冲区起始地址	0xFFFF
R16_SPI0_DMA_END	0x4000401C	SPI0 DMA 缓冲区结束地址	0xFFFF

SPI 模式控制寄存器(R8_SPI0_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	SPI0 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	从机模式 2 线或 3 线 SPI 模式选择: 1: 2 线模式/半双工 (SCK/MISO); 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	SPI0 主从模式选择: 1: 从机模式; 0: 主机模式。	0

SPI 配置寄存器 (R8_SPI0_CTRL_CFG)

位	名称	访问	描述	复位值
7	Reserved	RO	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	RO	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	0

			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	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_SPI0_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	RO	保留。	0
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_SPI0_CLOCK_DIV)

位	名称	访问	描述	复位值
[7:0]	R8_SPI0_CLOCK_DIV	RW	主机模式分频系数，最小值为 2，最大值为 254。 F _{sck} = 系统主频 F _{sys} /分频系数。	10h

SPI 从机模式预置数据寄存器 (R8_SPI0_SLAVE_PRE)

位	名称	访问	描述	复位值
[7:0]	R8_SPI0_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_SPI0_BUFFER)

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

[7:0]	R8_SPI0_BUFFER	RW	SPI 数据发送和接收缓冲区。	XXh
-------	----------------	----	-----------------	-----

SPI 工作状态寄存器(R8_SPI0_RUN_FLAG)

位	名称	访问	描述	复位值
7	RB_SPI_SLV_SELECT	RO	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	RO	从机模式被片选后首次加载状态位： 1：正在加载 R8_SPI0_SLAVE_PRE； 0：尚未加载或者已完成。	0
5	RB_SPI_FIFO_READY	RO	FIFO 准备就绪状态位： 1：FIFO 就绪 (R16_SPI0_TOTAL_CNT 非 0，且接收时 FIFO 未满载或发送时 FIFO 不空)； 0：FIFO 未准备好。	0
4	RB_SPI_SLV_CMD_ACT	RO	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	RO	保留。	0

SPI 中断标志寄存器(R8_SPI0_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	RO	当前 SPI 空闲状态位： 1：当前 SPI 空闲； 0：当前 SPI 非空闲。	1
5	Reserved	RO	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>=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_SPI0_FIFO_COUNT)

位	名称	访问	描述	复位值
[7:4]	Reserved	R0	保留。	0
[3:0]	R8_SPI0_FIFO_COUNT	RW	当前 FIFO 中字节计数。	0

SPI 中断触发方式选择寄存器 (R8_SPI0_INT_TYPE)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
[4:0]	RB_SPI_INT_TYPE	RW	中断触发方式选择, 1 = 上升沿触发, 0 = 电平触发。 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_SPI0_INTER1_EN)

位	名称	访问	描述	复位值
[7:2]	Reserved	R0	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_SPI0_INT1_FLAG)

位	名称	访问	描述	复位值
[7:2]	Reserved	R0	保留。	0
1	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 未滿。	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 收发数据总长度寄存器 (R16_SPI0_TOTAL_CNT)

位	名称	访问	描述	复位值
[15:0]	R16_SPI0_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_SPI0_FIFO)

位	名称	访问	描述	复位值
[7:0]	R8_SPI0_FIFO	RO/ WO	数据 FIFO 寄存器。	XXh

寄存器 R8_SPI0_BUFFER 和 R8_SPI0_FIFO 均为 SPI 数据相关寄存器, 主要区别在于:

读 R8_SPI0_BUFFER 是取自 SPI 最近一次交换到的数据，不影响 FIFO 和 R8_SPIx_FIFO_COUNT，主机模式下写 R8_SPI0_BUFFER 是直接发送该字节，从机模式下写操作未定义；
 读 R8_SPI0_FIFO 是取自 FIFO 中最早交换到的数据，将减少 FIFO 和 R8_SPI0_FIFO_COUNT，
 写 R8_SPI0_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_SPI0_TOTAL_CNT 非 0 时自动启动发送。

SPI 收发 FIFO 计数寄存器 1 (R8_SPI0_FIFO_COUNT1)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
[3:0]	R8_SPI0_FIFO_COUNT1	RW	当前 FIFO 中字节计数。 同 R8_SPI0_FIFO_COUNT。	0

SPI0 DMA 缓冲区当前地址 (R16_SPI0_DMA_NOW)

位	名称	访问	描述	复位值
15	Reserved	RO	保留。	0
[14:0]	R16_SPI0_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 = SPI0_DMA_NOW - SPI0_DMA_BEG。	XXXXh

SPI0 DMA 缓冲区起始地址 (R16_SPI0_DMA_BEG)

位	名称	访问	描述	复位值
15	Reserved	RO	保留。	0
[14:0]	R16_SPI0_DMA_BEG	RW	DMA 数据缓冲区起始地址，仅低 15 位有效。	XXXXh

SPI0 DMA 缓冲区结束地址 (R16_SPI0_DMA_END)

位	名称	访问	描述	复位值
15	Reserved	RO	保留。	0
[14:0]	R16_SPI0_DMA_END	RW	DMA 数据缓冲区结束地址（不含），仅低 15 位有效。	XXXXh

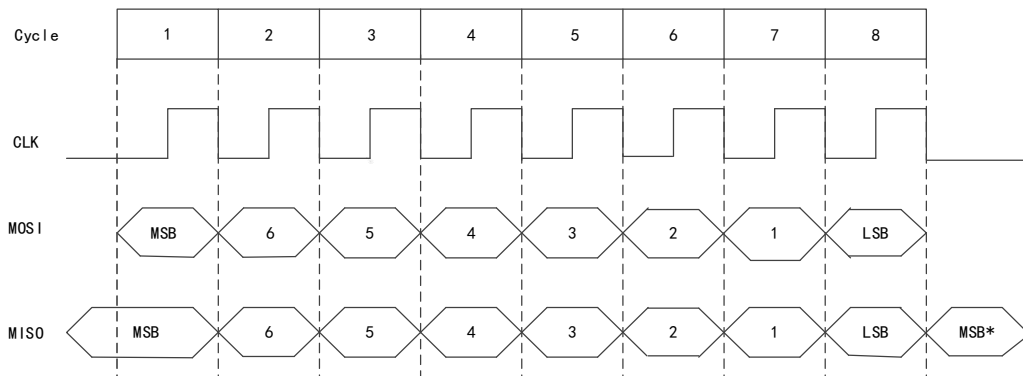
10.3 SPI 传输格式

SPI 支持模式 0 和模式 3 两种传输格式，通过设置 R8_SPI0_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：

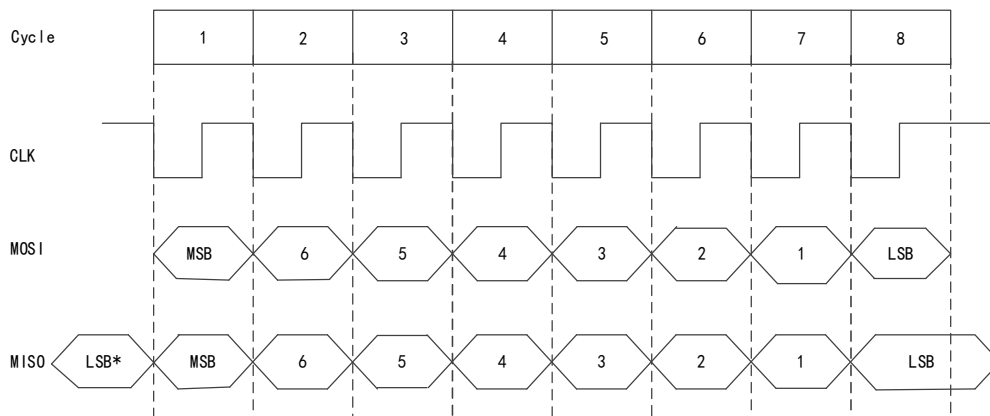
模式 0: RB_SPI_MST_SCK_MOD = 0

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_SPI0_CLOCK_DIV，配置 SPI 时钟频率；
- (2)、设置 R8_SPI0_CTRL_MOD 的 RB_SPI_MODE_SLAVE 为 0，配置 SPI 为主机模式；
- (3)、设置 R8_SPI0_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_SPI0_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_SPI0_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 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；
- (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_SPI0_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_SPI0_TOTAL_CNT 寄存器，设置要发送的数据长度；
- (3)、写 R8_SPI0_FIFO 寄存器，向 FIFO 中写入要发送的数据，如果 R8_SPI0_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_SPI0_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_SPI0_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_SPI0_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_SPI0_TOTAL_CNT 寄存器，设置要接收的数据长度；
- (3)、只要 R16_SPI0_TOTAL_CNT 非 0 并且 FIFO 未滿，SPI 主机就会自动接收数据，否则暂停；
- (4)、等待 R8_SPI0_FIFO_COUNT 寄存器不为 0，则说明接收到返回数据，读取 R8_SPI0_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

SPI0 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_SPI0_CTRL_MOD 的 RB_SPI_MODE_SLAVE 为 1，配置 SPI0 为从机模式；
- (2)、根据需要设置 R8_SPI0_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_SPI0_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_SPI0_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)、可选的，设置 SPI0 从机模式预置数据寄存器 R8_SPI0_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_SPI0_SLAVE_PRE 中的预置数据（状态值）。R8_SPI0_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_SPI0_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_SPI0_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_SPI0_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）；
- (6)、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。

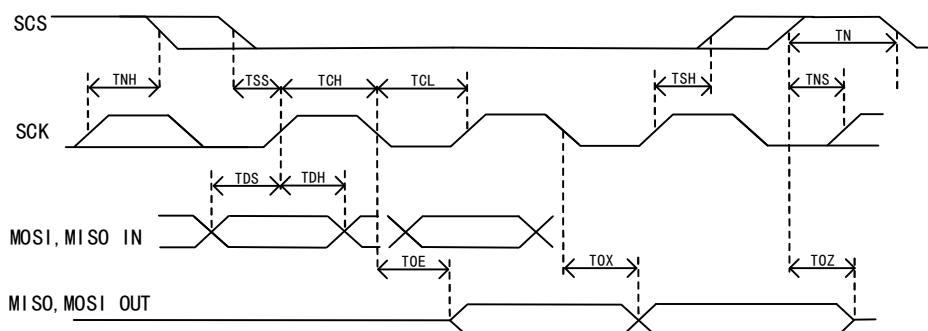
数据发送过程：

- (1)、设置 R8_SPI0_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_SPI0_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_SPI0_FIFO_COUNT，如果未满载则继续向 FIFO 写入待发送的数据。

数据接收过程：

- (1)、设置 R8_SPI0_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_SPI0_FIFO_COUNT，如果不空则说明已接收到数据，通过读取 R8_SPI0_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_SPI0_BUFFER。

10.5 SPI 时序



名称	参数说明 (TA=25°C, VI015=1.5V)	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	$T_{sys} * 1.05$			nS
TSH	SCK 上升沿之后 SCS 有效的保持时间	$T_{sys} * 1.05$			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)	$T_{sys} * 2$			nS
TCH	SCK 时钟的高电平时间	$T_{sys} * 0.55$			nS
TCL	SCK 时钟的低电平时间	$T_{sys} * 0.55$			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 MISO/MOSI 输出有效	0		18	nS
TOX	SCK 下降沿到 MISO/MOSI 输出改变	0	5	16	nS
TOZ	SCS 无效到 MISO/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 PWM

11.1 PWM 控制器简介

除了定时器提供的 3 路 26 位 PWM 输出之外，芯片还提供了 4 路 8 位 PWM 输出 (PWM4~PWM5、PWM8、PWM10) 或 6 路 16 位 PWM 输出 (PWM4~PWM5、PWM8)，占空比可调；其中，8 位的 PWM 固定可选 8 种周期，而 16 位的 PWM 周期任意可调，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	0x0X
R8_PWM_CLOCK_DIV	0x40005003	PWMx 时钟分频寄存器	0x00
R32_PWM4_7_DATA	0x40005004	PWM4/5 数据保持寄存器	0xFFFFFFFF
R8_PWM4_DATA	0x40005004	PWM4 数据保持寄存器	0xXX
R8_PWM5_DATA	0x40005005	PWM5 数据保持寄存器	0xXX
R32_PWM8_11_DATA	0x40005008	PWM8/10 数据保持寄存器	0xFFFFFFFF
R8_PWM8_DATA	0x40005008	PWM8 数据保持寄存器	0xXX
R8_PWM10_DATA	0x4000500A	PWM10 数据保持寄存器	0xXX
R8_PWM_INT_CTRL	0x4000500C	PWMx 中断控制和状态寄存器	0x00
R32_PWM_REG_DATA8	0x40005010	PWM8 数据寄存器	0xFFFFFFFF
R32_PWM_REG_CYCLE	0x40005014	PWMx 周期数结束寄存器	0x0000XXXX

PWMx 输出使能寄存器 (R8_PWM_OUT_EN)

位	名称	访问	描述	复位值
7	Reserved	R0	保留。	0
6	RB_PWM10_OUT_EN	RW	PWM10 输出使能位： 1：使能； 0：禁止。	0
5	Reserved	R0	保留。	0
4	RB_PWM8_OUT_EN	RW	PWM8 输出使能位： 1：使能； 0：禁止。	0
[3:2]	Reserved	R0	保留。	0
1	RB_PWM5_OUT_EN	RW	PWM5 输出使能位： 1：使能； 0：禁止。	0
0	RB_PWM4_OUT_EN	RW	PWM4 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	Reserved	R0	保留。	0
6	RB_PWM10_POLAR	RW	PWM10 输出极性控制位：	0

			1: 默认高电平, 低有效; 0: 默认低电平, 高有效。	
5	Reserved	R0	保留。	0
4	RB_PWM8_POLAR	RW	PWM8 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
[3:2]	Reserved	R0	保留。	0
1	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
0	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

PWMx 配置控制寄存器 (R8_PWM_CONFIG)

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
4	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
[3:2]	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
1	RB_PWM_STAG_ST	R0	PWM 交错标志位: 1: 指示 PWM5 被允许输出; 0: 指示 PWM4 被允许输出。	X
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 个时钟周期; 16 位数据宽度对应时钟周期受 R32_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 R32_PWM_REG_CYCLE 控制。	0

PWMx 时钟分频寄存器 (R8_PWM_CLOCK_DIV)

位	名称	访问	描述	复位值
[7:0]	R8_PWM_CLOCK_DIV	RW	PWM 基准时钟分频系数。 $F_{pwm} = F_{sys}/R8_PWM_CLOCK_DIV$ 。	0

PWM 数据保持寄存器组 1 (R32_PWM4_7_DATA)

位	名称	访问	描述	复位值
[15:8]	R8_PWM5_DATA	RW	PWM5 数据保持寄存器。	XXh
[7:0]	R8_PWM4_DATA	RW	PWM4 数据保持寄存器。	XXh
[31:16]	R16_PWM5_DATA	RW	PWM5 数据保持寄存器 (16 位宽)。	XXh

[15:0]	R16_PWM4_DATA	RW	PWM4 数据保持寄存器（16 位宽）。	XXh
--------	---------------	----	----------------------	-----

PWM 数据保持寄存器组 2 (R32_PWM8_11_DATA)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:16]	R8_PWM10_DATA	RW	PWM10 数据保持寄存器。	XXh
[15:8]	Reserved	R0	保留。	0
[7:0]	R8_PWM8_DATA	RW	PWM8 数据保持寄存器。	XXh

PWMx 中断控制和状态寄存器 (R8_PWM_INT_CTRL)

位	名称	访问	描述	复位值
7	RB_PWM_IF_CYC	RW1	PWM 周期结束标志位，写 1 清零： 1：周期结束； 0：未结束。	0
[6:2]	Reserved	R0	保留。	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	PWM 周期结束中断使能位： 1：使能中断； 0：禁止中断。	0

PWM8 数据寄存器 (R32_PWM_REG_DATA8)

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:0]	R16_PWM8_REG_DATA	RW	PWM8 通道的 16 位 bit 数据。	XXXXh

注：只有 16 数据宽度时有效。

PWM 周期数结束寄存器 (R32_PWM_REG_CYCLE)

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:0]	RB_PWM_CYC_VALUE	RW	16 数据宽度时的 PWM 时钟周期： $N_{cyc} = RB_PWM_CYC_VALUE + 1$ 。	XXXXh

11.3 PWM 配置

- 设置寄存器 R8_PWM_CLOCK_DIV，配置 PWM 的基准时钟频率；
- 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；
- 设置 PWM 配置控制寄存器 R8_PWM_CONFIG，设置 PWM 的模式、数据位宽、周期；
- 设置 PWM 输出使能寄存器 R8_PWM_OUT_EN，开启对应的 PWMx 输出使能；
- 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；
- 设置 PWM4~PWM5、PWM8、PWM10 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；
- 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：

数据宽度 $n=8/7/6$: $N_{cyc}=2^n-RB_PWM_CYCLE_SEL$;

16 位数据宽度: $N_{cyc}=RB_PWM_CYC_VALUE+1$;

PWMx 占空比计算公式:

占空比= $R8_PWMx_DATA/N_{cyc}$;

PWMx 输出频率 F_{pwmout} 计算公式:

PWMx 输出频率 $F_{pwmout}=F_{pwm}/N_{cyc}=F_{sys}/R8_PWM_CLOCK_DIV/N_{cyc}$ 。

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。

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

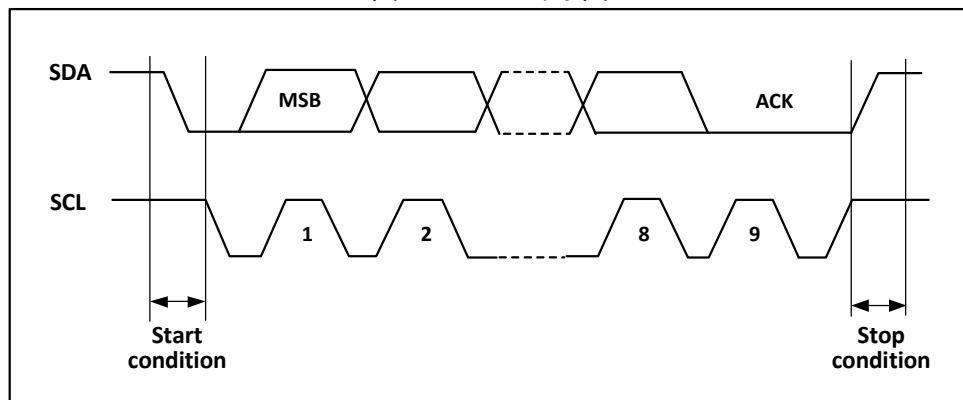
芯片提供 I2C 接口，特性如下：

- 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，其同时只能运行在下列四种模式之一：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。

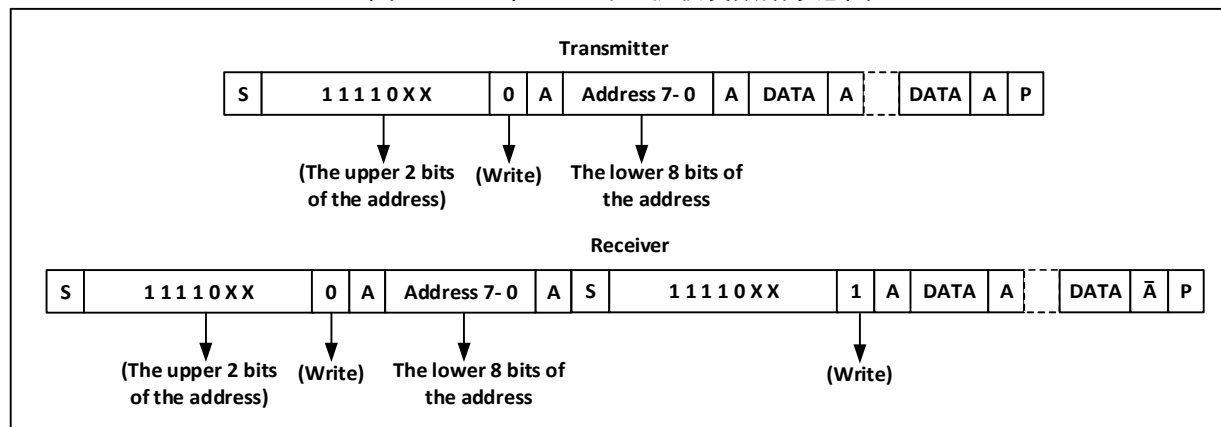
然后写数据寄存器发送第二个地址字节，在发送完第二个地址字节后，状态寄存器的 ADDR 位会被置位，如果 ITEVTEN 位已经置位，则会产生中断，此时应读取 R16_I2C_STAR1 寄存器后再读一次 R16_I2C_STAR2 寄存器以清除 ADDR 位；

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 12-2 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 12-2 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 模块能识别其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；如果第二个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，并不再响应针对其从地址发起的传输，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：

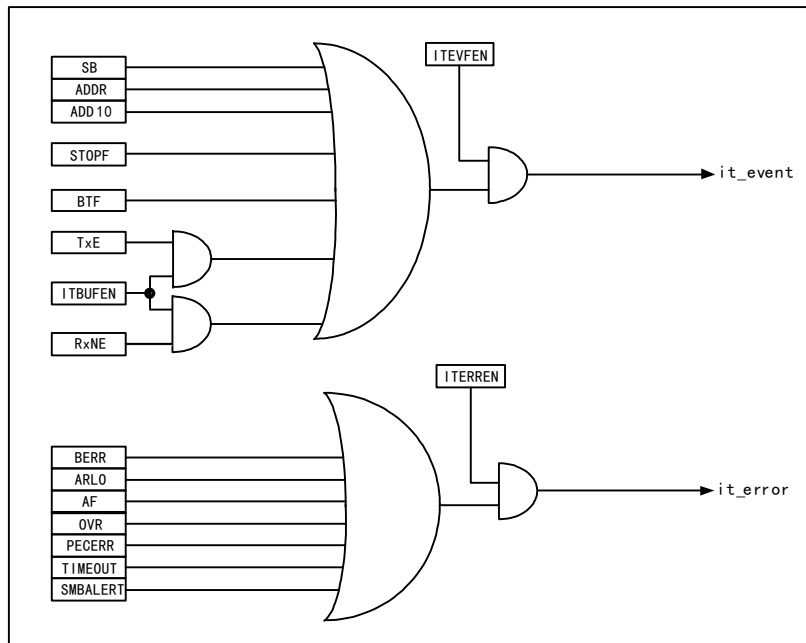
- 1) I2C 支持的速度可达 400KHz，SMBus 速度仅达 100KHz，且 SMBus 有最小 10KHz 的速度限制；
- 2) SMBus 的时钟为低超过 35mS 时，会报超时，但 I2C 无此限制；
- 3) SMBus 有固定的逻辑电平，而 I2C 没有，取决于芯片电源电压 VDD；
- 4) SMBus 有总线协议，而 I2C 没有。

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 12-3 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 1 (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	RO	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 个字节，	0

			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。	0

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_ITEVTEN	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。	0
[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。	0

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。	0
[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。	0
[7:1]	RB_I2C_ADD7_1	RW	总线地址, 第 7-1 位。	0
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。	0
[7:1]	RB_I2C_ADD2	RW	总线地址, 双地址模式下地址的 7-1 位。	0
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。	0
[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	0

			于发送到总线的的数据。	
--	--	--	-------------	--

I2C 状态寄存器 1 (R16_I2C_STAR1)

位	名称	访问	描述	复位值
15	RB_I2C_SMBALERT	RWO	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	RWO	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 处于低已达到 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。 <i>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。</i>	0
13	Reserved	RO	保留。	0
12	RB_I2C_PECERR	RWO	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	RWO	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	RWO	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	RWO	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	RWO	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	RO	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。	0

			1: 发送数据时, 发送数据寄存器为空; 0: 数据寄存器非空。	
6	RB_I2C_RxNE	R0	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	R0	保留。	0
4	RB_I2C_STOPF	R0	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	R0	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	R0	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	RWO	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	R0	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	R0	包错误检查域, 当 PEC 使能时 (ENPEC 置位),	0

			此域存放 PEC 的值。	
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。 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。	0
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)

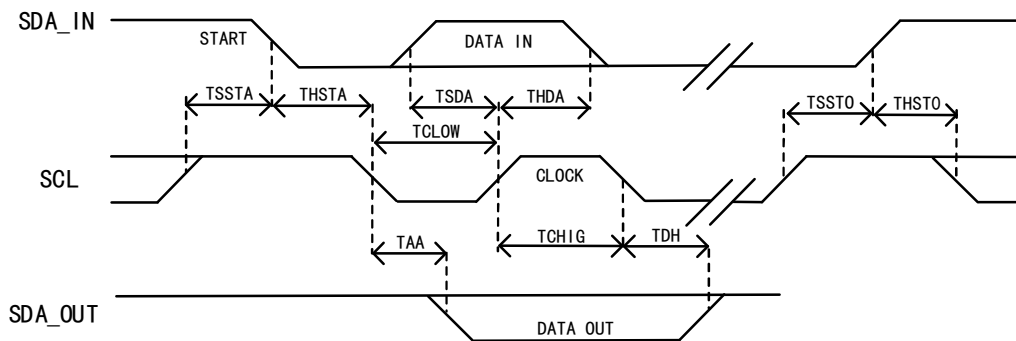
位	名称	访问	描述	复位值
15	RB_I2C_F_S	RW	主模式选择位。 1: 快速模式; 0: 标准模式。	0
14	RB_I2C_DUTY	RW	快速模式时的时钟高电平时间的占空比。 1: 36%; 0: 33.3%。	0
[13:12]	Reserved	RO	保留。	0
[11:0]	RB_I2C_CCR	RW	时钟分频系数域, 决定 SCL 时钟的频率波形。 在 I2C 标准模式下或 SMBus 模式下: $T_{high} = CCR * T_{HCLK}$ $T_{low} = CCR * T_{HCLK}$ 在 I2C 快速模式下:	0

		<p>如果 DUTY = 0:</p> $T_{high} = CCR * T_{HCLK}$ $T_{low} = 2 * CCR * T_{HCLK}$ <p>如果 DUTY = 1: (速度达到 400kHz)</p> $T_{high} = 9 * CCR * T_{HCLK}$ $T_{low} = 16 * CCR * T_{HCLK}$ <p>例: 在标准模式下, 产生 100kHz 的 SCL 频率: $CCR = FREQR / (2 * 100KHz)$, $FREQR = 48MHz$ 时, $T_{HCLK} = 1 / 48MHz$, 则 $CCR = 48MHz / (2 * 100KHz) = 240$, CCR 写入 0xF0。</p>	
--	--	--	--

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。	0
[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, VI015=1.5V)	最小值	典型值	最大值	单位
TSSTA	SDA 下降沿之前 SCL 高电平的建立时间	0.4			uS
THSTA	SDA 下降沿之后 SCL 高电平的保持时间	0.4			uS
TSDA	SCL 上升沿之前 SDA 数据的建立时间	0.05			uS
THDA	SCL 上升沿之后 SDA 数据的保持时间	>TCHIG			uS
TSSTO	SDA 上升沿之前 SCL 高电平的建立时间	0.4			uS
THSTO	SDA 上升沿之后 SCL 高电平的保持时间	0.8			uS
TCLOW	SCL 时钟的低电平时间	0.6			uS
TCHIG	SCL 时钟的高电平时间	0.5			uS
TAA	SCL 下降沿到 SDA 输出有效	0.006		0.4	uS
TDH	SCL 下降沿之后 SDA 输出保持时间	0.006			uS
TR	SCL 或 SDA 输入上升沿时间			0.2	uS

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 0xC000, 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): 1: 开启停止开关; 0: 关闭停止开关。	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。	0
[11:0]	RLR	RW	RL[11:0]: 看门狗计数器重载 (具有写保护功能)。用于定义看门狗计数器的重载值, 每当向	FFFh

			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。	
--	--	--	---	--

第 14 章 模数转换器 ADC

14.1 ADC 简介

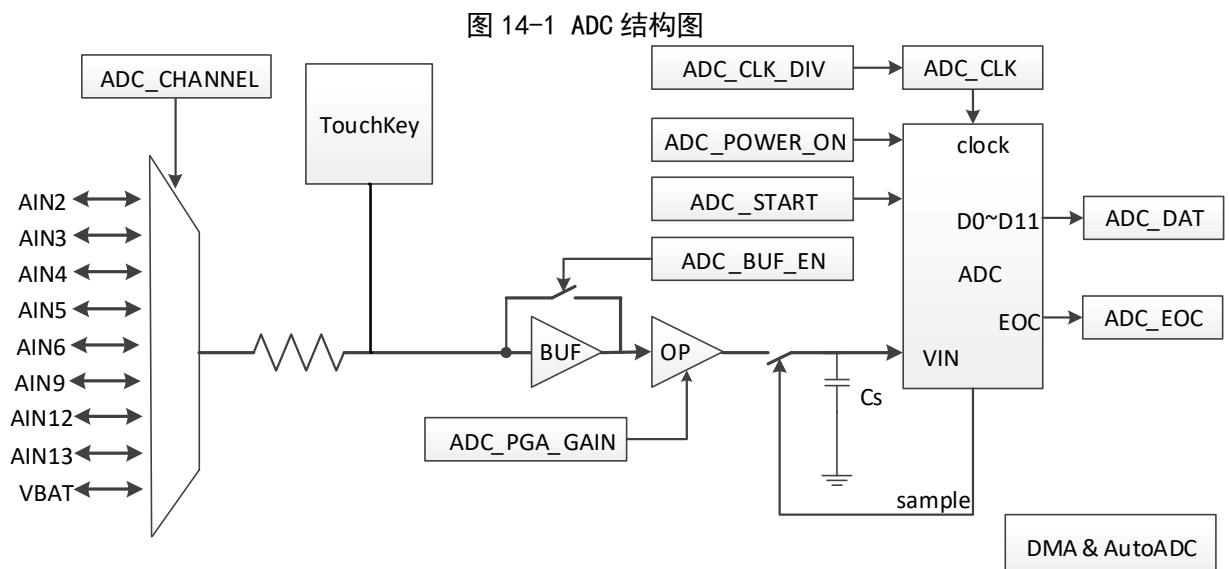
芯片提供一个 12 位逐次逼近型模拟数字转换器 ADC，提供多达 9 个通道，支持通道扫描（SCAN）功能，支持 8 个外部信号源和 1 个内部信号源。

14.1.1 主要特性

- 12 位分辨率
- 8 个外部电压采样通道、内部的电池电压检测通道
- 单端输入模式
- 采样时钟频率可选
- ADC 输入电压范围 $0V \sim V_{I015}$
- 可选 PGA，提供增益调节选择
- 可选输入缓冲器 BUF，支持高阻信号源
- 支持 DMA 和定时间隔自动连续 ADC 采样，间隔可调
- 支持通道扫描功能，可配置扫描的顺序、扫描通道的数量，并提供扫描结束中断

14.1.2 功能描述

下图为一个 ADC 模块的框图。



14.2 寄存器描述

表 14-1 ADC 相关寄存器列表

名称	访问地址	描述	复位值
R8_ADC_CHANNEL	0x40001058	ADC 输入通道选择寄存器	0x0F
R8_ADC_CFG	0x40001059	ADC 配置寄存器	0x20
R8_ADC_CONVERT	0x4000105A	ADC 转换控制寄存器	0x08
R16_ADC_DATA	0x4000105C	ADC 数据寄存器	0x0XXX
R8_ADC_INT_FLAG	0x4000105E	ADC 中断标志寄存器	0x00
R8_ADC_CTRL_DMA	0x40001061	DMA 和中断控制寄存器	0x00
R8_ADC_DMA_IF	0x40001062	ADC 和 DMA 中断标志寄存器	0x00
R8_ADC_AUTO_CYCLE	0x40001063	连续 ADC 定时周期寄存器	0xXX
R16_ADC_DMA_NOW	0x40001064	DMA 当前缓冲区地址	0xXXXX

R16_ADC_DMA_BEG	0x40001068	DMA 起始缓冲区地址	0xXXXX
R16_ADC_DMA_END	0x4000106C	DMA 结束缓冲区地址	0xXXXX
R32_ADC_SCAN_CFG1	0x40001070	扫描配置寄存器 1	0xFFFFFFFF
R32_ADC_SCAN_CFG2	0x40001074	扫描配置寄存器 2	0x00FFFFFF

ADC 输入通道选择寄存器 (R8_ADC_CHANNEL)

位	名称	访问	描述	复位值
[7:4]	Reserved	R0	保留。	0
[3:0]	RB_ADC_CH_INX	RW	ADC 通道索引号，共 9 个通道： 02h-06h：外部信号通道 AIN2~AIN6； 09h：外部信号通道 AIN9； 0Ch-0Dh：外部信号通道 AIN12~AIN13； 0Eh：电池电压 VBAT。	1111b

ADC 配置寄存器 (R8_ADC_CFG)

位	名称	访问	描述	复位值
7	Reserved	R0	保留。	0
6	RB_ADC_CLK_DIV	RW	ADC 时钟频率选择： 当 R16_CLK_SYS_CFG[9] = 1 时： 1：基于 CK32M 的 4 分频，8MHz； 0：基于 CK32M 的 2 分频，16MHz。 当 R16_CLK_SYS_CFG[9] = 0 时： 1：基于 CK16M 的 2 分频，8MHz； 0：基于 CK16M 直接送入，16MHz。	0
[5:4]	RB_ADC_PGA_GAIN[1:0]	RW	ADC 的输入 PGA 增益选择： 00：保留； 01：-6dB，1/2 倍； 10：0dB，1 倍，无增益； 11：6dB，2 倍。	10b
3	RB_ADC_OFS_TEST	RW	ADC 偏移误差测试模式： 1：测试/校准模式，测试模式下数据寄存器 R16_ADC_DATA 低 12 位数据将按位取反（0x0573 反为 0x0A8C）； 0：正常模式。	0
2	Reserved	R0	保留。	0
1	RB_ADC_BUF_EN	RW	ADC 输入缓冲器 BUF 使能： 1：开启； 0：关闭。	0
0	RB_ADC_POWER_ON	RW	ADC 模块电源使能控制： 1：使能； 0：关闭。	0

PGA 增益计算公式说明： $\Delta V * A + V_{ref} = 2 * V_{ref} * ADC / 4096$ 。

ΔV ：ADC 模块输入的电压。

单端模式下 $\Delta V = (VIN - 0)$ ，其中 VIN 就是被测电压。

ADC 模块输入电压参数为 $V * A$ ，需要满足： $-V_{ref} \leq V * A \leq +V_{ref}$ 。

A：增益系数，参考表 14-2 中“PGA 增益选择”。

ADC：ADC 转换后的数字量，即 R16_ADC_DATA。

Vref: 内部模拟电路的电源节点 VINTA 的实际电压值, 通常为 1.05V±0.015。

表 14-2 PGA 增益选择与输入电压范围表 (单端输入模式)

PGA 增益选择	VINn	由 ADC 转换后的数据 计算被采样的电压 VINp	理论可测电压 上限	测量电压范围 (假定 Vref=1.05V)	高精度 测量电压范围
-6dB (1/2)	0	(ADC/1024-2)*Vref	2*Vref	0V~VI015+0.2V	0.05V~ VI015-0.1
0dB (1)	0	(ADC/2048-1)*Vref	1*Vref	0V~1.05V	0.05V~1V
6dB (2)	0	(ADC/4096-0.5)*Vref	0.5*Vref	0V~0.525V	0.05V~0.5V

ADC 转换控制寄存器 (R8_ADC_CONVERT)

位	名称	访问	描述	复位值
7	RB_ADC_EOC_X	RO	ADC 转换结束标志 (异步信号): 1: 完成; 0: 进行中。	0
6	Reserved	RO	保留。	0
[5:4]	RB_ADC_SAMPLE_TIME	RW	ADC 采样周期选择: 00: 4*Tadc; 01: 6*Tadc; 10: 8*Tadc; 11: 10*Tadc。	0
3	RB_ADC_CHAN_EN	RW	ADC 通道开启使能: 1: 开启通道使能, 可以选择通道; 0: 关闭通道使能, 不选择任何通道。	1
[2:1]	Reserved	RO	保留。	0
0	RB_ADC_START	RW	ADC 转换启动控制及状态, 在非连续 ADC 结束时或在 DMA 结束时自动清零: 1: 开始转换/正在转换; 0: 停止转换。	0

ADC 数据寄存器 (R16_ADC_DATA)

位	名称	访问	描述	复位值
[15:12]	Reserved	RW	保留。	0
[11:0]	RB_ADC_DATA	RO	ADC 转换后的数据。	XXXh

ADC 中断标志寄存器 (R8_ADC_INT_FLAG)

位	名称	访问	描述	复位值
7	RB_ADC_IF_EOC	RO	ADC 转换完成标志。 对寄存器 R8_ADC_CONVERT 或 RB_TKEY_CTRANS_ACT 进行写操作可清零 此标志。	0
[6:0]	Reserved	RW	保留。	0

DMA 和中断控制寄存器 (R8_ADC_CTRL_DMA)

位	名称	访问	描述	复位值
7	RB_ADC_AUTO_EN	RW	定时间隔自动连续 ADC 采样使能: 1: 使能自动 ADC; 0: 禁止自动 ADC。	0
6	RB_ADC_CONT_EN	RW	ADC 连续转换模式使能:	0

			1: 使能连续 ADC; 0: 禁止连续 ADC。	
5	Reserved	RO	保留。	0
4	RB_ADC_IE_EOC	RW	ADC 转换完成中断使能位: 1: 使能中断; 0: 禁止中断。	0
3	RB_ADC_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_ADC_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_ADC_DMA_ENABLE	RW	DMA 功能使能位: 1: 使能 DMA; 0: 禁止 DMA。	0

ADC 和 DMA 中断标志寄存器 (R8_ADC_DMA_IF)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
4	RB_ADC_IF_END_ADC	RW1	ADC 转换完成中断标志, 写 1 清零或 DMA 取走数据或对寄存器 R8_ADC_CONVERT 进行写操作可清零此标志: 1: 已完成一次 ADC; 0: 未完成。	0
3	RB_ADC_IF_DMA_END	RW1	DMA 完成标志位, 写 1 清零: 1: 已完成; 0: 未完成。	0
[2:0]	Reserved	RO	保留。	0

连续 ADC 定时周期寄存器 (R8_ADC_AUTO_CYCLE)

位	名称	访问	描述	复位值
[7:0]	R8_ADC_AUTO_CYCLE	RW	连续 ADC 定时周期起始值, 以 16 个系统时钟为单位计数, 满 256 后重新加载。 计算方法: 定时 = (256-R8_ADC_AUTO_CYCLE)*16*T _{sys} 。 连续 TKEY 定时周期起始值时, 以 32MHz 或 16MHz 时钟为单位计数。 计算方法: 定时 = R8_ADC_AUTO_CYCLE*T _{CK32M} (或 T _{CK16M})。	XXh

DMA 当前缓冲区地址 (R16_ADC_DMA_NOW)

位	名称	访问	描述	复位值
15	Reserved	RO	保留。	0
[14:1]	R16_ADC_DMA_NOW	RO	DMA 数据缓冲区当前地址。	XXXXh

			可用于计算已转换次数，计算方法： COUNT=(ADC_DMA_NOW-ADC_DMA_BEG)/2。	
0	Reserved	R0	保留。	0

DMA 起始缓冲区地址(R16_ADC_DMA_BEG)

位	名称	访问	描述	复位值
15	Reserved	R0	保留。	0
[14:1]	R16_ADC_DMA_BEG	RW	DMA 数据缓冲区起始地址，地址必须 2 字节对齐。	XXXXh
0	Reserved	R0	保留。	0

DMA 结束缓冲区地址(R16_ADC_DMA_END)

位	名称	访问	描述	复位值
15	Reserved	R0	保留。	0
[14:1]	R16_ADC_DMA_END	RW	DMA 数据缓冲区结束地址（不含），地址必须 2 字节对齐。	XXXXh
0	Reserved	R0	保留。	0

扫描配置寄存器 1(R32_ADC_SCAN_CFG1)

位	名称	访问	描述	复位值
[31:28]	RB_ADC_SCAN_CH8	RW	通道 8 模拟输入通道选择位： 0010: AIN2; 0011: AIN3; 0100: AIN4; 0101: AIN5; 0110: AIN6; 1001: AIN9; 1100: AIN12; 1101: AIN13; 1110: 电池电压 VBAT; 其他: 保留。	1111b
[27:24]	RB_ADC_SCAN_CH7	RW	通道 7 模拟输入通道选择位： 同上。	1111b
[23:20]	RB_ADC_SCAN_CH6	RW	通道 6 模拟输入通道选择位： 同上。	1111b
[19:16]	RB_ADC_SCAN_CH5	RW	通道 5 模拟输入通道选择位： 同上。	1111b
[15:12]	RB_ADC_SCAN_CH4	RW	通道 4 模拟输入通道选择位： 同上。	1111b
[11:8]	RB_ADC_SCAN_CH3	RW	通道 3 模拟输入通道选择位： 同上。	1111b
[7:4]	RB_ADC_SCAN_CH2	RW	通道 2 模拟输入通道选择位： 同上。	1111b
[3:0]	RB_ADC_SCAN_CH1	RW	通道 1 模拟输入通道选择位： 同上。	1111b

扫描配置寄存器 2(R32_ADC_SCAN_CFG2)

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

31	RB_ADC_IF_SCAN_END	RW	扫描结束标志位，为 1 表示扫描结束，该位写 1 清零，写 0 无效。	0
30	RB_ADC_SCAN_MASKDIS	RW	ADC 采样通道结束后，关闭通道使能： 1：禁止关闭通道； 0：使能关闭通道。	0
29	RB_ADC_IE_SCAN_END	RW	扫描结束中断使能： 1：使能中断； 0：禁止中断。	0
28	RB_ADC_SCAN_SEL	RW	扫描选择： 1：ADC； 0：TKEY。	0
[27:24]	RB_ADC_SCAN_NUM	RW	扫描通道数量。 RB_ADC_SCAN_SEL 为 1 时：表示通道扫描数量减 1； RB_ADC_SCAN_SEL 为 0 时：表示通道扫描数量。	0
23	RB_ADC_SCAN_EN	RW	通道扫描模式使能： 1：使能模式； 0：不使能模式。	0
22	Reserved	RO	保留。	0
[21:20]	RB_EHCH_TIM	RW	通道扫描时，每个通道的 ADC 采样次数。 00：1 次； 01：2 次； 10：3 次； 11：4 次。	0
[19:16]	RB_ADC_SCAN_CH13	RW	通道 13 模拟输入通道选择位： 0010：AIN2； 0011：AIN3； 0100：AIN4； 0101：AIN5； 0110：AIN6； 1001：AIN9； 1100：AIN12； 1101：AIN13； 1110：电池电压 VBAT； 其他：保留。	1111b
[15:12]	RB_ADC_SCAN_CH12	RW	通道 12 模拟输入通道选择位： 同上。	1111b
[11:8]	RB_ADC_SCAN_CH11	RW	通道 11 模拟输入通道选择位： 同上。	1111b
[7:4]	RB_ADC_SCAN_CH10	RW	通道 10 模拟输入通道选择位： 同上。	1111b
[3:0]	RB_ADC_SCAN_CH9	RW	通道 9 模拟输入通道选择位： 同上。	1111b

14.3 ADC 配置

14.3.1 外部通道采样

(1)、设置 R8_ADC_CFG 的 RB_ADC_POWER_ON 为 1 开启 ADC，通过 RB_ADC_CLK_DIV 选择采样频率，通

过 RB_ADC_BUF_EN 和 RB_ADC_PGA_GAIN 启用输入缓冲器及选择信号增益，通过 RB_ADC_SAMPLE_TIME 选择采样周期等；

- (2)、设置 R8_ADC_CHANNEL 寄存器，选择外部或内部信号通道；
- (3)、设置 R8_ADC_CONVERT 寄存器，置位 RB_ADC_START，启动 ADC 转换；
- (4)、查询等待 RB_ADC_START 自动清零或者等待 RB_ADC_IF_EOC 置 1，表示转换完成，可以读取 R16_ADC_DATA 获取 12 位 ADC 转换数据，时间足够时建议再次转换并丢弃首次 ADC 数据；
- (5)、重复 2、3、4 步骤，可以继续采样另一通道或者下一组数据。
- (6)、单次 ADC 转换周期：

ADC 采样 (4/6/8/10 个时钟) + 转换时间 (12 个时钟) \approx 16/18/20/22 个 T_{adc} ，连续 ADC 时还要加上 1 个时间间隔，其中， $T_{adc} = T_{CK32M}$ (或 T_{CK16M}) / RB_ADC_CLK_DIV。

14.3.2 启用 DMA 定时间隔自动连续 ADC

- (1)、参考非 DMA 方式设置 ADC 参数和选择通道等；
- (2)、设置 R8_ADC_AUTO_CYCLE 选择连续 ADC 的周期；
- (3)、设置寄存器 R16_ADC_DMA_BEG 为存储 ADC 数据缓冲区的首地址，设置寄存器 R16_ADC_DMA_END 为存储 ADC 数据缓冲区的结束地址 (不含)，并设置 R8_ADC_CTRL_DMA 的 RB_ADC_DMA_ENABLE 为 1，使能 DMA 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，当 RB_ADC_IE_EOC = 1 时将由 RB_ADC_IF_END_ADC 触发 ADC 完成中断，当 RB_ADC_IE_EOC = 0 且 RB_ADC_IE_DMA_END = 0 时将由 RB_ADC_IF_EOC 触发 ADC 完成中断；
- (5)、将 RB_ADC_AUTO_EN 置 1 以开启定时间隔自动连续 ADC；
- (6)、每次 ADC 完成后，RB_ADC_IF_EOC 和 RB_ADC_IF_END_ADC 都会置 1，但 RB_ADC_IF_END_ADC 会在 DMA 取走数据时自动清零，所以，如果需要查询 ADC 完成状态，那么可以查询 RB_ADC_IF_EOC。

14.4 通道扫描

14.4.1 扫描模式

扫描模式用来扫描一组模拟通道，扫描队列中的 0 通道所对应的模拟输入通道由 R8_ADC_CHANNEL 寄存器的 [4:0] 来选择，扫描队列中的 1~13 通道所对应的模拟输入通道则由 R32_ADC_SCAN_CFG1 和 R32_ADC_SCAN_CFG2 寄存器来配置。

扫描通道模式可以通过设置 R32_ADC_SCAN_CFG2 寄存器的 RB_ADC_SCAN_NUM 位、RB_ADC_SCAN_SEL 位和 RB_ADC_SCAN_MASK_DIS 位来设置，如果 RB_ADC_SCAN_NUM 不等于 0，则会对输入的模拟通道按照配置的顺序依次扫描；通过 RB_ADC_SCAN_SEL 位来选择扫描的是 TKEY 还是 ADC；此外还可通过 RB_ADC_SCAN_MASK_DIS 位控制：在每一次通道扫描中，ADC 采样结束后是否立刻关闭该通道。当 RB_SCAN_MASK_DIS 位置 1 表示在每次通道扫描中，ADC 采样结束后不关闭该通道，直到进行下一个通道扫描时，再关闭上一个通道开启下一个通道，而当 RB_SCAN_MASK_DIS 置 0 表示在每次通道扫描中，ADC 采样结束后立刻关闭该通道，到进行下一通道扫描时，开启下一个通道，以此防止关闭上一通道的延时导致两个通道同时导通的问题。

14.4.2 扫描中断

在扫描模式下，如果 R32_ADC_SCAN_CFG2 寄存器的 RB_ADC_IE_SCAN_END 位被置位，将使能扫描结束中断，即在扫描队列的最后一个通道转换结束时将产生中断，此时 R32_ADC_SCAN_CFG2 寄存器的 RB_ADC_IF_SCAN_END 位会被硬件置 1 (该位需要软件清 0)，表明扫描队列中的所有通道转换结束。

第 15 章 触摸按键 Touch-Key

15.1 Touch-Key 简介

芯片提供了电容检测模式，配合 ADC 模块使用，可以实现电容类触摸按键检测功能。共 8 个输入通道，支持触摸按键电容值范围 10pF~100pF，提供驱动屏蔽输出以提高灵敏度。

15.2 寄存器描述

表 15-1 TouchKey 相关寄存器列表

名称	访问地址	描述	复位值
R8_TKEY_CFG	0x40001057	TouchKey 配置寄存器	0x00
R32_TKEY_CFG2	0x40001078	TouchKey 配置寄存器 2	0x00000000
R32_TKEY_SEL	0x4000107C	TouchKey 选择寄存器	0x00000000

TouchKey 配置寄存器 (R8_TKEY_CFG)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
6	RB_TKEY_DMA_EN	RW	TouchKey DMA 使能： 1：使能； 0：关闭。	0
[5:4]	Reserved	RO	保留。	0
3	RB_ADC_PGA_CFG	RW	选择 ADC 中 PGA 的运行速度： 1：高速但功耗略大； 0：正常速度。	0
2	RB_TKEY_DRV_EN	RW	TouchKey 驱动屏蔽使能： 1：使能； 0：禁用。	0
[1:0]	Reserved	RO	保留。	0

TouchKey 配置寄存器 2 (R32_TKEY_CFG2)

位	名称	访问	描述	复位值
[31:26]	Reserved	RO	保留。	0
[25:16]	RB_TRANSFER_N	RW	搬移周期次数。	0
[15:9]	Reserved	RO	保留。	0
8	RB_TKEY_CTRANS_ACT	RW	TKEY 模块启动信号及工作状态，自动清零： 1：启动检测/正在工作； 0：停止工作。	0
7	RB_TKEY_EN	RW	TKEY 模块使能： 1：TKEY 模块开启； 0：TKEY 模块关闭。	0
[6:5]	RB_TKADC_START_SEL	RW	触摸按键提前启动 ADC 时间选择，单位为搬移周期： 00：2； 01：4； 10：8； 11：16。	0

[4:3]	RB_TKEY_CC	RW	触摸按键电容充电时间： 00: 1Tsys; 01: 2Tsys; 10: 3Tsys; 11: 4Tsys。	0
[2:0]	RB_TKEY_TRANS	RW	触摸按键电容电荷搬移时间： 000: 1Tsys; 001: 4Tsys; 010: 8Tsys; 011: 12Tsys; 100: 16Tsys; 101: 24Tsys; 110: 32Tsys; 111: 48Tsys。	0

TouchKey 选择寄存器 (R32_TKEY_SEL)

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
[29:16]	RB_DRIVEROUT_EN	RW	TouchKey 驱动屏蔽通道选择开关： 1: 通道打开； 0: 通道关闭。	0
[15:14]	Reserved	RO	保留。	0
[13:0]	RB_SEL_CS	RW	电荷搬移通道选择： 10000000000000: ADCIN13; 01000000000000: ADCIN12; 00001000000000: ADCIN9; 00000001000000: ADCIN6; 00000000100000: ADCIN5; 00000000010000: ADCIN4; 00000000001000: ADCIN3; 00000000000100: ADCIN2;	0

注：RB_SEL_CS[13: 0]=*****中，14 位控制字哪些位为 1 则打开哪个通道，存在多通道同时打开的情况。

单次通道扫描模式操作流程：

(1)、设置 R8_ADC_CFG 的 RB_ADC_CLK_DIV 选择采样频率，设置 R8_ADC_CFG 的 RB_ADC_BUF_EN 为 1，启用输入缓冲器，设置 R8_ADC_CFG 的 RB_ADC_PGA_GAIN 选择信号增益，设置 R8_ADC_CONVERT 的 RB_ADC_SAMPLE_TIME 选择采样周期等（如果使用 DMA，还需配置 DMA）；设置 RB_TKEY_DRV_EN 开启 TouchKey 驱动屏蔽使能，设置 RB_DRIVEROUT_EN 选择驱动屏蔽通道，设置 RB_ADC_CH_INX 通道选择（接大电容通道选择）；

(2)、设置 RB_SEL_CS 通道选择（搬移通道选择），配置 RB_TKEY_CC 充电时间选择，RB_TKEY_TRANS 搬移时间选择，设置 RB_TRANSFER_N 配置单通道搬移次数，RB_TKADC_START_SEL 配置 ADC 使能信号提前打开的时间；

注：ADC_PGA 使能信号不需要配置为高电平，TKEY 模块中会根据 RB_TKADC_START_SEL 的配置在适当的时间使能 ADC 和 PGA。如果先配置 ADC_PGA 使能，则在刚搬移时 PGA 和 ADC 就开始工作，从而导致功耗的浪费；

(3)、配置 RB_TKEY_EN 信号，使能 TKEY，通过配置 RB_ADC_START 信号和 RB_TKEY_CTRANS_ACT 信号来启动电荷迁移；

(4)、当前通道搬移过程结束，检测到 RB_ADC_START 位置 0 后，读取 R16_ADC_DATA 获取 12 位触摸按

键转换数据；

(5)、重复(2)、(3)、(4)步骤，可以继续采样另一通道或者下一组数据。

连续通道扫描模式操作流程：

(1)、设置 R8_ADC_CFG 的 RB_ADC_CLK_DIV 选择采样频率，设置 R8_ADC_CFG 的 RB_ADC_BUF_EN 为 1，启用输入缓冲器，设置 R8_ADC_CFG 的 RB_ADC_PGA_GAIN 选择信号增益，设置 R8_ADC_CONVERT 的 RB_ADC_SAMPLE_TIME 选择采样周期等；设置 RB_TKEY_DRV_EN 开启 TouchKey 驱动屏蔽使能。

(2)、配置 DMA 相关设置，如设置 ADC DMA 首地址和结束地址（不含），数据缓冲区长度不小于 RB_ADC_SCAN_NUM 所配置的扫描个数；配置 R8_ADC_CTRL_DMA，使能 DMA 功能位 RB_ADC_DMA_ENABLE，使能 DMA 结束中断位 RB_ADC_IE_DMA_END；

(3)、配置 RB_ADC_CH_INX 位，这一通道配置为接大电容 C_x 的通道，不参与通道扫描轮循；

(4)、扫描队列中的 1~13 通道所对应的模拟输入通道则由 R32_ADC_SCAN_CFG1 和 R32_ADC_SCAN_CFG2 寄存器来配置（将待扫描的通道依次填入扫描队列中），则会按照填入通道名称的顺序依次进行扫描，由于是 TKEY 通道扫描，因此 RB_ADC_SCAN_SEL 置 0，由 RB_ADC_SCAN_NUM 来配置待扫描的通道个数，再配置扫描结束中断使能位（RB_ADC_IE_SCAN_END）；

(5)、配置一套连续通道扫描电荷迁移 TKEY 的使用配置，在(3)和(4)中设置的 RB_ADC_CH_INX 位所选择的通道即为接大电容 C_x 的通道，R32_ADC_SCAN_CFG1 和 R32_ADC_SCAN_CFG2 寄存器中所填入的通道即为依次进行电荷迁移扫描的通道；配置 RB_TRANSFER_N，确认单个通道进行电荷搬移周期次数 N；配置 RB_TKEY_CC 充电时间选择，RB_TKEY_TRANS 搬移时间选择；RB_TKADC_START_SEL 确认 PGA 和 ADC 的上电时间；将 RB_TKEY_EN 置高电平，配置 RB_ADC_START 信号和 RB_TKEY_CTRANS_ACT 信号后，当前单个通道的电荷迁移过程开始进行，ADC 采样完成之后，上面对于 TKEY 的功能配置不变，即可按照 R32_ADC_SCAN_CFG1 和 R32_ADC_SCAN_CFG2 寄存器中所设置的通道顺序依次对所有通道进行以上设置的功能扫描。

(6)、查询扫描结束标志位 RB_ADC_IF_SCAN_END，以确定扫描结束；从 DMA 数据缓冲区获取转换数据；

(7)、在连续通道扫描后启用 DMA 功能保存采样值时，可配置 RB_EHCH_TIM 位来选择每个通道采样的数值个数，最多一个通道连续采样四个值存储进 DMA。

15.3 Touch-Key 配置

请参考和调用相关子程序。

第 16 章 预留

第 17 章 无线通讯

17.1 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-24dBm 到+4dBm 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。

17.2 2.4GHz 模块

17.2.1 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。

17.2.2 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。

17.3 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。

17.4 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。

17.5 BB 模块

17.6 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 18 章 LED 屏控制器

18.1 LED 控制器简介

芯片提供了 LED 屏控制卡接口，内置 2 字节 FIFO，支持 DMA 和中断，节约 CPU 处理时间，支持 1/2/4 路数据线接口。

18.2 寄存器描述

表 18-1 LED 相关寄存器列表

名称	访问地址	描述	复位值
R8_LED_CTRL_MOD	0x4000E000	LED 模式配置寄存器	0x02
R8_LED_CLOCK_DIV	0x4000E001	LED 串行时钟分频寄存器	0x10
R8_LED_STATUS	0x4000E004	LED 状态寄存器	0xX0
R16_LED_FIFO	0x4000E008	LED 数据 FIFO 寄存器	0xXXXX
R16_LED_DMA_CNT	0x4000E010	LED DMA 剩余计数寄存器	0x0000
R16_LED_DMA_MAIN	0x4000E014	LED 主缓冲区 DMA 地址	0xXXXX
R16_LED_DMA_AUX	0x4000E018	LED 辅助缓冲区 DMA 地址	0xXXXX

LED 模式配置寄存器 (R8_LED_CTRL_MOD)

位	名称	访问	描述	复位值
[7:6]	RB_LED_CHAN_MOD	RW	LED 通道模式选择： 00: LED0，单通道输出； 01: LED0/1，双通道输出； 10: LED0~3，4 通道输出； 11: LED0~3，4 通道输出，其中 LED2/3 通道数据来源辅助缓冲区。	0
5	RB_LED_IE_FIFO	RW	FIFO 计数过半中断使能： 1: FIFO 计数≤2 中断触发； 0: 禁止中断。	0
4	RB_LED_DMA_EN	RW	LED DMA 功能和 DMA 中断使能： 1: 使能； 0: 禁止。	0
3	RB_LED_OUT_EN	RW	LED 信号输出使能： 1: 使能； 0: 禁止。	0
2	RB_LED_OUT_POLAR	RW	LED 数据输出极性控制位： 1: 反相，数据 0 输出 1，数据 1 输出 0； 0: 直通，数据 0 输出 0，数据 1 输出 1。	0
1	RB_LED_ALL_CLEAR	RW	LED 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_LED_BIT_ORDER	RW	LED 串行数据位序选择： 1: 高位在前； 0: 低位在前。	0

LED 串行时钟分频寄存器 (R8_LED_CLOCK_DIV)

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

[7:0]	R8_LED_CLOCK_DIV	RW	LED 输出时钟分频系数。 LEDC 频率 = Fsys/R8_LED_CLOCK_DIV。	10h
-------	------------------	----	---	-----

LED 状态寄存器 (R8_LED_STATUS)

位	名称	访问	描述	复位值
7	RB_LED_IF_DMA_END	RW1	DMA 完成标志位, 写 1 清零或者写 R16_LED_DMA_CNT 清零: 1: 已完成; 0: 未完成。	0
6	RB_LED_FIFO_EMPTY	RO	FIFO 空状态位: 1: FIFO 空; 0: FIFO 中有数据。	1
5	RB_LED_IF_FIFO	RW1	FIFO 计数过半中断标志位, 写 1 清零或者写 R16_LED_FIFO 清零: 1: FIFO 计数≤2; 0: FIFO 计数>2。	0
4	RB_LED_CLOCK	RO	当前 LED 时钟信号电平状态: 1: 高电平; 0: 低电平。	X
3	Reserved	RO	保留。	0
[2:0]	RB_LED_FIFO_COUNT	RO	当前 FIFO 中字节计数值, 一定偶数。	0

LED 数据 FIFO 寄存器 (R16_LED_FIFO)

位	名称	访问	描述	复位值
[15:0]	R16_LED_FIFO	WO	LED 数据 FIFO 入口, 16 位写入。	XXXXb

LED DMA 剩余计数寄存器 (R16_LED_DMA_CNT)

位	名称	访问	描述	复位值
[15:0]	R16_LED_DMA_CNT	RW	LED_DMA_MAIN 主缓冲区当前 DMA 剩余字 (16 位) 计数, DMA 启动后自动递减, 仅低 12 位有效。不包括辅助缓冲区。	0

LED 主缓冲区 DMA 地址 (R16_LED_DMA_MAIN)

位	名称	访问	描述	复位值
[15:0]	R16_LED_DMA_MAIN	RW	主缓冲区 DMA 起始地址/当前地址, 预设初值后, 自动递增, 必须 2 字节对齐。	XXXXb

LED 辅助缓冲区 DMA 地址 (R16_LED_DMA_AUX)

位	名称	访问	描述	复位值
[15:0]	R16_LED_DMA_AUX	RW	辅助缓冲区 DMA 起始地址/当前地址, 预设初值后, 自动递增, 必须 2 字节对齐。	XXXXb

18.3 LED 配置

- 设置 R8_LED_CLOCK_DIV 选择 LED 输出时钟频率, 设置 RB_LED_ALL_CLEAR 为 0;
- 设置 R16_LED_DMA_MAIN 指向准备输出数据的缓冲区, 即主缓冲区;
- 如果选择 LED 通道模式 3, 那么还要设置 R16_LED_DMA_AUX 指向辅助缓冲区;
- 设置 R8_LED_CTRL_MOD, 选择通道模式、输出极性、位顺序、启用中断和 DMA 功能等;

- (5)、设置 LEDC 和必要的 LED0~LED3 引脚方向为输出，可选地，设置相应 I/O 的驱动能力；
- (6)、设置 DMA 计数寄存器 R16_LED_DMA_CNT，启动 DMA 发送，或者用写 FIFO 的方式发送数据。

第 19 章 编码器

19.1 编码器简介

芯片支持编码器（ENC），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。

19.2 寄存器描述

芯片支持编码器，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 19-1 计数方向与编码信号关系

有效边沿	相对信号的电平 (CAP_IN1 信号相对 CAP_IN2, CAP_IN2 信号相对 CAP_IN1)	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。

表 19-2 编码器相关寄存器列表

名称	访问地址	描述	复位值
R8_ENC_REG_CTRL	0x40006080	编码器控制寄存器	0x00
R8_ENC_INTER_EN	0x40006081	编码器中断使能寄存器	0x00
R8_ENC_INT_FLAG	0x40006082	编码器中断标志寄存器	0x00
R32_ENC_REG_CEND	0x40006084	编码器终值配置寄存器	0x00000000
R32_ENC_REG_CCNT	0x40006088	编码器当前值配置寄存器	0x00000000

编码器控制寄存器 (R8_ENC_REG_CTRL)

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

7	RB_ENC_IN_CH2	RO	编码器捕获通道 2 的值。	0
6	RB_ENC_IN_CH1	RO	编码器捕获通道 1 的值。	0
5	RB_ENC_DIR	RO	编码器当前方向： 1: 后退； 0: 前进。	0
4	Reserved	RO	保留。	0
3	RB_RD_CLR_EN	RW	编码器读计数并清零。	0
[2:1]	RB_SMS_MODE	RW	编码器边沿工作模式： 00 = IDLE； 10 = 在 CAP_IN1 边沿计数； 01 = 在 CAP_IN2 边沿计数； 11 = 在 CAP_IN1 和 CAP_IN2 边沿计数。	0
0	RB_START_ENC_EN	RW	编码器工作使能： 1: 使能（必须打开 LSI）； 0: 禁止。	0

编码器中断使能寄存器 (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	编码器前进中断使能： 1: 使能中断； 0: 禁止中断。	0

编码器中断标志寄存器 (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	编码器后退中断标志，写 1 清 0。 1: 编码器后退； 0: 编码器保持不变。	0
0	RB_IF_DIR_INC	RW	编码器前进中断标志，写 1 清 0。 1: 编码器前进； 0: 编码器保持不变。	0

编码器终值配置寄存器 (R32_ENC_REG_CEND)

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

编码器当前值配置寄存器 (R32_ENC_REG_CCNT)

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

第 20 章 参数

20.1 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 20-1 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
VDD15	系统电源电压 (VDD15 接电源, GND 接地)	-0.4	2.2	V
VI015	I/O 电源电压 (VI015 接电源, GND 接地)	-0.4	2.2	V
VI0	输入或者输出引脚上的电压	-0.4	VI015+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	X32MI/X32MO/启用 LSE 后的 PA10/PA11 的电压	-0.3	1.4	V

20.2 电气参数

测试条件: TA = 25°C, VDD15 = VI015 = 1.5V, Fsys = 16MHz。

表 20-2 电气参数表

名称	参数说明	最小值	典型值	最大值	单位
VDD15	系统电源电压@VDD15	1.2	1.5	1.8	V
VI015	I/O 电源电压@VI015	1.2	1.5	1.8	V
ICC ₈	直通方式静态电源电流 @条件: 代码运行于 FLASH 中	Fsys=8M	2.9		mA
ICC ₁₆		Fsys=16M	3.4		mA
ICC ₄₈		Fsys=48M	6.4		mA
VIL	GPIO 低电平输入电压	0		0.5	V
VIH	GPIO 高电平输入电压	1.0		VI015	V
VIL5	支持 5V 耐压的 GPIO 低电平输入电压	0		0.5	V
VIH5	支持 5V 耐压的 GPIO 高电平输入电压	1.0		5	V
VOL	低电平输出电压 (2mA/8mA 吸入电流)	0	0.3	0.4	V
VOH	高电平输出电压 (2mA/8mA 输出电流)	VI015-0.4	VI015-0.3	VI015	V
IIN	GPIO 浮空输入端的输入电流	-2	0	2	uA
IUP	GPIO 内置上拉电阻的输入端的输入电流	3	9	18	uA
IDN	GPIO 内置下拉电阻的输入端的输入电流	-18	-9	-3	uA
Vref	VINTA 引脚的电压 (ADC 参考电压)	1.035	1.05	1.065	V
VIvr	LVR 低压复位的门限电压	1.03	1.1	1.16	V

20.3 低功耗模式功耗

测试条件: TA = 25°C, VDD15 = VI015 = 1.5V, Fsys = 16MHz。

表 20-3 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		1.4		mA
暂停模式, 关闭所有时钟		180		uA
睡眠模式, 多种组合, 参考表 5-3		1.8~3.8		uA
睡眠模式, PMU+内核+RAM8K, GPIO 唤醒, 无 RTC		1.8		uA
下电模式, 多种组合, 参考表 5-3		0.85~2.6		uA
下电模式, 仅 PMU, GPIO 唤醒后复位, 无 RTC		0.85		uA

表 20-4 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
$I_{DD}(RAM8K)$	RAM8K: 8KB SRAM			0.7		μA
$I_{DD}(RAM24K)$	RAM24K: 24KB SRAM			1.75		μA
$I_{DD}(LSI)$	内部 LSI 振荡器	高精度		0.4		μA
		低精度		0.26		μA
$I_{DD}(HSI)$	内部 HSI 振荡器			90		μA
$I_{DD}(LSE)$	外部 LSE 振荡器			0.4		μA
$I_{DD}(HSE)$	外部 HSE 振荡器		110	190	300	μA
$I_{DD}(BM)$	低功耗的电池低压监控 BM 模块			1		μA
$I_{DD}(PLL)$	内部 PLL 振荡器			170		μA
$I_{DD}(ADC)$	ADC 模块			0.45		mA
$I_{DD}(TKEY)$	TouchKey 模块			0.2		mA
$I_{DD}(BLE)$	BLE 蓝牙	接收	直通电源	5		mA
		-24dBm 发送功率	直通电源	3.3		mA
		0dBm 发送功率	直通电源	8.6		mA
		+4dBm 发送功率	直通电源	13		mA

20.4 时钟源

表 20-5 高速振荡器 HSI 和 HSE

名称	参数说明		最小值	典型值	最大值	单位
F_{HSI}	内部 HSI 振荡器频率			16		MHz
A_{HSI}	HSI 振荡器精度	$TA = -40^{\circ}C \sim 85^{\circ}C$		± 1	± 2.5	%
T_{SUHSI}	内部 HSI 振荡器启动到可用时间				1	μs
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	μs
T_{STHSE}	外部 HSE 振荡器启动到稳定时间		200	500	3000	μs

表 20-6 低速振荡器 LSI 和 LSE

名称	参数说明		最小值	典型值	最大值	单位
F_{LSIR}	内部 LSI 振荡器频率（校准前）		20K	32K	48K	Hz
F_{LSI}	内部 LSI 频率（应用软件运行时校准后）		32726	32768	32810	Hz
A_{LSI}	LSI 振荡器精度 （软件校准后）	$TA = -40^{\circ}C \sim 85^{\circ}C$		0.1	0.5	%
		$TA = -20^{\circ}C \sim 60^{\circ}C$		0.04	0.2	%
T_{SULSI}	内部 LSI 振荡器启动到稳定时间			40	100	μs
T_{SULSE}	外部 LSE 振荡器启动到可用时间		100	300	1500	mS
T_{STLSE}	外部 LSE 振荡器启动到稳定时间		500	1500	5000	mS

表 20-7 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 * 15 倍）			480		MHz
T_{PLLLK}	PLL 锁相时间			20	40	μs

20.5 时间参数

测试条件： $TA = 25^{\circ}C$ ， $VDD15 = VI015 = 1.5V$ ， $F_{sys} = 6.4MHz$ 。

表 20-8 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	3	5	7	mS	
T_{sr}	软件复位 SR 后的复位延时	2	8	18	uS	
T_{wtr}	看门狗复位 WTR 后的复位延时	10	12	18	uS	
T_{wak}	从低功耗状态退出的唤醒时间	空闲模式	0.6	1	3	uS
		暂停模式	$T_{SUHSE}+1$	$T_{SUHSE}+80$	$T_{SUHSE}+150$	uS
		睡眠模式	$T_{SUHSE}+1$	$T_{SUHSE}+300$	$T_{SUHSE}+400$	uS
		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 0.2~1ms（启动到可用 T_{SUHSE} ）。

20.6 其它参数

测试条件：TA = 25°C，VDD15 = VI015 = 1.5V，Fsys = 16MHz。

表 20-9 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 的单个扇区擦除操作时间	6.0	6.4	7.0	mS
T_{FRPG}	Flash-ROM 的单个字或单次页编程操作时间	2.0	2.2	2.6	mS
N_{EPCE}	Flash-ROM 的擦写次数 erase/program cycle endurance	5~45°C	100K	800K（抽测）	times
		-40~85°C	30K	100K（抽测）	
T_{DR}	Flash-ROM 的数据保持能力	10			years
V_{ESD}	I/O 输入或者输出 引脚上的 ESD 耐压	天线 ANT	2K	4K（抽测）	V
		I/O 引脚：PA 和 PB	4K	6K（抽测）	V

第 21 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	CH596D

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

