

第 3 章 STM32W108 体系结构

3.1 STM32W108 架构

3.1.1 片内结构

STM32W108 是一个集成的片上系统,将一个 2.4GHz 的 IEEE 802.15.4 标准的收发器、32 位的 ARM Cortex-M3 微处理器、Flash 和 RAM 存储器融合,添加相应的外设,设计成基于 802.15.4 的系统。图 3.1 为 STM32W108 的框图。

STM32W108 的 RF 收发器采用高效的结构,超出 IEEE 802.15.4:2003 标准的动态变化范围 15dB。集成的接收信道过滤允许与其他 2.4GHz 频谱的通信标准共存,例如 IEEE 802.11、蓝牙等。为了降低外部元件的数量,选择使用集成的稳压器、VCO、环路滤波器、功率放大器,一个可选的高性能无线模式(增强模式)可通过软件选择,以提高动态范围。

集成的 32 位的 ARM Cortex-M3 微处理器在高性能、低功耗、高效的内存利用方面进行了高度优化,拥有一个集成的 MPU,它支持两种不同的模式操作:特权模式和非特权模式。这种体系结构能够将网络协议栈从应用程序代码中分离,以避免多余的对限制的内存区域和寄存器的修改,这样可以增加已部署解决方案的稳定性和可靠性。

3.1.2 功能单元描述

STM32W108 拥有 128KB 的 Flash(支持 128KB/192KB/256KB)和 8KB 的 RAM(支持 8KB/12KB/16KB)用于数据和程序存储,及可配置的 ARM 内存保护单元(MPU)。STM32W108 HAL 软件采用了有效的磨损均衡算法,用于优化延长嵌入式 Flash 的寿命。

为了满足 ZigBee 和 IEEE 802.15.4:2003 标准严格的时序要求,STM32W108 在硬件中集成了 MAC 功能。MAC 硬件处理能够进行自动 ACK 发送和接收,自动补偿延迟,为了更有效地传输自动清空信道以及自动过滤接收到的数据包。数据包跟踪接口也集成在 MAC 中,允许完整、非侵入式的捕获来自和发送到 STM32W108 的所有数据包。

STM32W108 提供一系列的先进的电源管理功能以延长电池寿命,高频的内部 RC 振荡器能够保证处理器在迅速唤醒之后开始执行代码。各种深睡眠模式下功耗小于 $1\mu\text{A}$,同时保留 RAM 内容。为了支持用户自定义的应用程序,片上外设包括 UART、SPI、I²C、ADC 和通用定时器,以及多达 24 个 GPIO 管脚。此外,集成稳压器、上电复位电路以及睡眠时钟都是可用的。

STM32W108 无线接收器是一个低 IF、超外差接收机,该体系结构已经被选择用于优化与其他 2.4GHz 频段(例如 Wi-Fi 和蓝牙)设备的共存和降低能量消耗。接收器使用差分信号路径,以降低对噪声的敏感度。信号经过 RF 放大后,通过一个镜频抑制混频器下变频,然后过滤,最后通过 ADC 数字化。

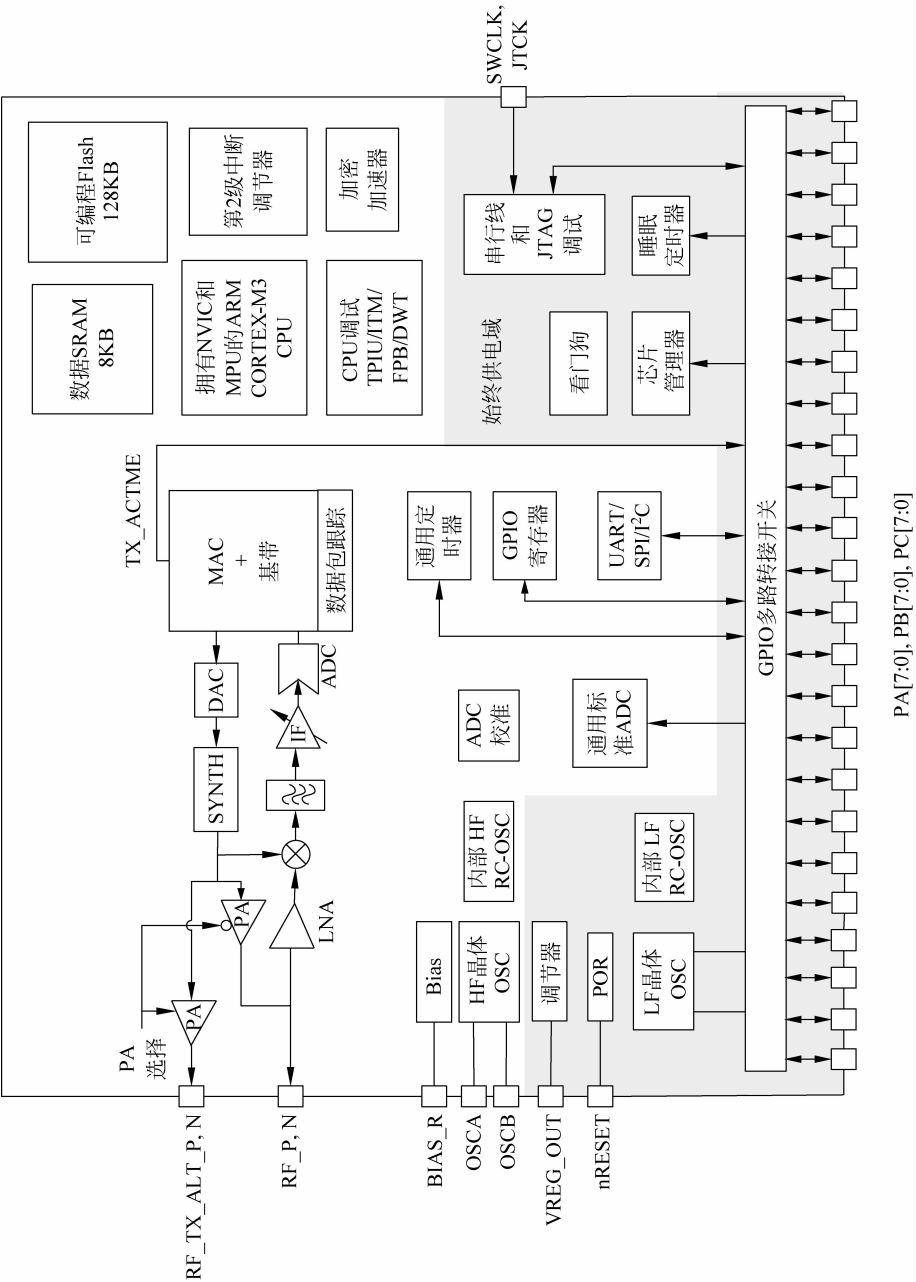


图 3.1 STM32W108 框图

无线射频使用一个高效的体系结构,该结构将数据流直接调制成 VCO 频率,集成功率放大器(PA)提供输出功率,数字逻辑控制 Tx 路径和输出功率校准。如果 STM32W108 使用一个外部 PA,那么利用 TX_ACTIVE 和 nTX_ACTIVE 信号控制外部逻辑开关的时序。

集成的 4.8GHz 的 VCO 和环路滤波器最大限度地减少了芯片的外电路,只再需要一个 24MHz 晶振及负载电容来构造 PLL 本地振荡器信号。

MAC 的接口将片上 RAM 与收发基带模块连接起来,支持基于硬件的 IEEE 802.15.4 数据包过滤,并且提供精确的时间信号基准,最大限度地减少软件栈在同步方面的工作,同时满足协议栈的时序要求。此外,为 IEEE 802.15.4 CSMA-CA 算法提供定时和同步协助。

STM32W108 采用业界领先的 32 位能量高效的 ARM Cortex-M3 内核,通过选择使用 ARM Thumb2 指令集,保证优越的源码密度,提高内存利用率。处理器能够被 12MHz 和 24MHz 水晶振荡器驱动,或者被 6MHz 和 12MHz 的集成高频 RC 振荡器驱动。

STM32W108 拥有 24 个 GPIO 引脚,可被其他外设共用或作为复用功能引脚,外设可以使用不同 GPIO 引脚的复用功能。内部集成的串口控制器 SC1 能够作为 SPI(主或从)、IIC(主)、UART 功能使用,串口控制器 SC2 能够被配置为 SPI(主或从)、IIC(主)功能使用。

STM32W108 拥有一个通用 ADC,可以对单端或差分模式的 6 个 GPIO 引脚进行模拟信号采样,还可以对稳压电源 VDD_PADSA、参考电压 VREF 和 GND 进行采样。ADC 有两个可选的电压范围:0~1.2V(正常)以及在高电源电压供给下的 0.1~(电源电压-0.1)V(高)。ADC 有一个 DMA 样本采集模式,该模式下采集的数据会自动传输到 RAM 中。ADC 集成的参考电压 VREF 可以供给电路使用,外部参考电压也可以供 ADC 使用。

STM32W108 包含 4 个振荡器:一个 24MHz 外部高频晶体振荡器、一个 12MHz 内部高频 RC 振荡器、一个可选的 32.768kHz 外部低频晶体振荡器和一个 10kHz 的内部 RC 振荡器。

STM32W108 具有超低功耗特性,拥有可选的深度睡眠时钟,睡眠定时器可以使用外部 32.768kHz 的晶体振荡器或内部 10kHz RC 振荡器产生 1kHz 时钟作为计数时钟。另外,在最低功耗模式下,所有时钟都可以被禁止。在最低功耗模式下,只有 GPIO 引脚上的外部事件能够唤醒芯片。STM32W108 从深度睡眠到执行第一条 ARM Cortex-M3 指令的启动时间非常短(一般为 100 μ s)。

STM32W108 有三个电源域,始终开启的高压电源给 GPIO 焊盘和芯片关键功能部件供电,稳定低压电源给芯片的其他部件供电。低压电源在深睡眠模式下是被禁止的,这样可以降低功耗。集成稳压器从不稳定的电源电压产生 1.25V 和 1.8V 的稳定电压,1.8V 稳压输出为模拟电路、RAM 和 Flash 存储器供电,1.25V 稳压输出为核心逻辑电路供电。

接收机的数字部分使用一个相干解调器为 MAC 产生基于硬件的信号,数字接收器还包括模拟无线电校准,并控制接收路径的增益。

除了两个通用定时器,STM32W108 还包含一个看门狗定时器、一个 32 位睡眠定时器和一个 NVIC 中的 ARM 标准的系统事件定时器,看门狗定时器用于从软件崩溃、CPU 死锁中恢复,32 位睡眠定时器用于系统定时和在指定时间唤醒系统。

STM32W108 同时支持 ARM 串行线和 JTAG 两种调试接口,这两种接口提供实时、非侵入编程和调试功能。虽然串行线和 JTAG 提供相同的功能,但是不能同时执行。串行线接口使用两个针脚,JTAG 接口使用 5 个针脚,串行线是首选的,因为它需要的针脚更少。

STM32W108 也融合了标准的 ARM 系统调试组件:Flash 补丁和断点(FPB)、数据观察跟踪(DWT)、仪表跟踪宏单元(DWT)。

3.2 封装与引脚说明

3.2.1 封装

STM32W108 有两种封装定义,图 3.2 是 48 脚的 VFQFPN 封装,封装大小为 $7\text{mm} \times 7\text{mm}$,图 3.3 是 40 脚的 VFQFPN 封装,封装大小为 $6\text{mm} \times 6\text{mm}$,40 脚的封装少了一些 GPIO 引脚,有关引脚的功能说明见 3.2.2 节。

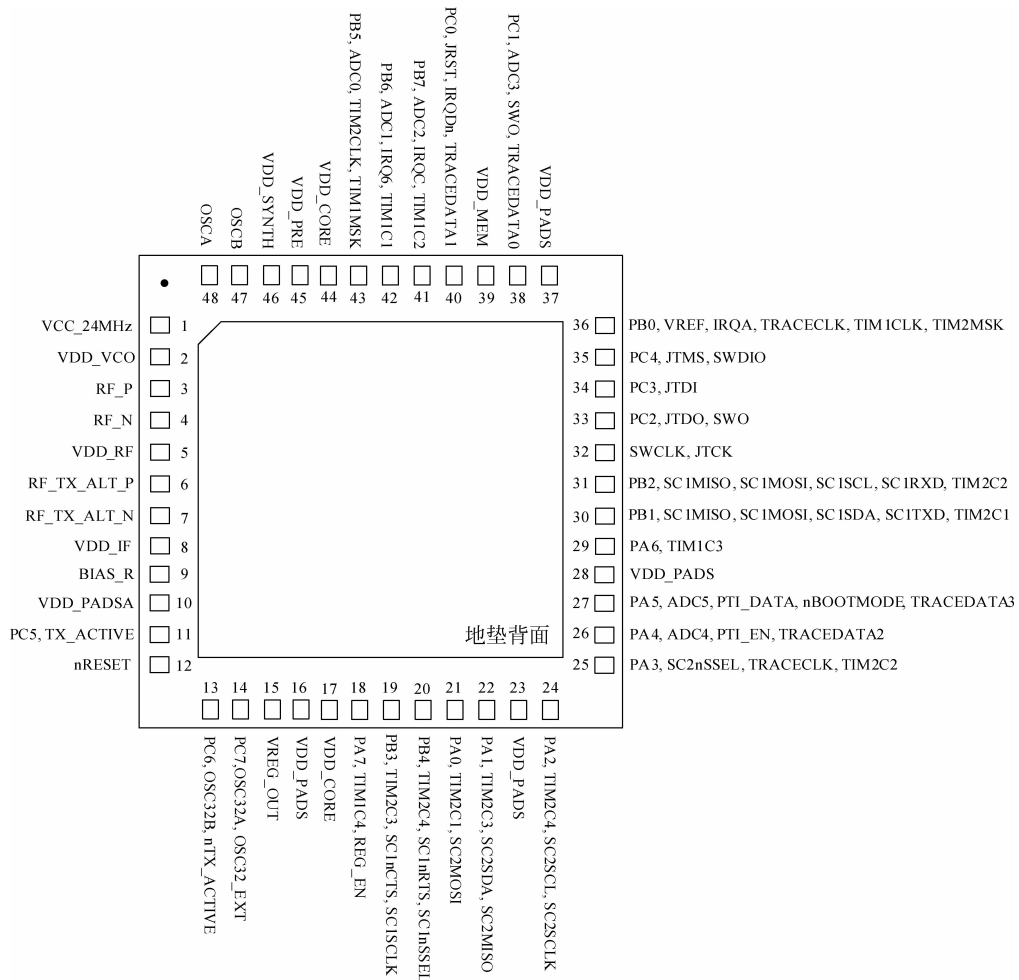


图 3.2 48 脚 VRQFPN 引脚分配

3.2.2 引脚说明

STM32W108 的引脚一般具有多个功能,在应用中需要首先配置选择寄存器,连接到相应的功能引脚上。STM32W108 的引脚功能说明如表 3.1 所示,在进行系统设计时需要详细阅读表 3.1 中的内容。

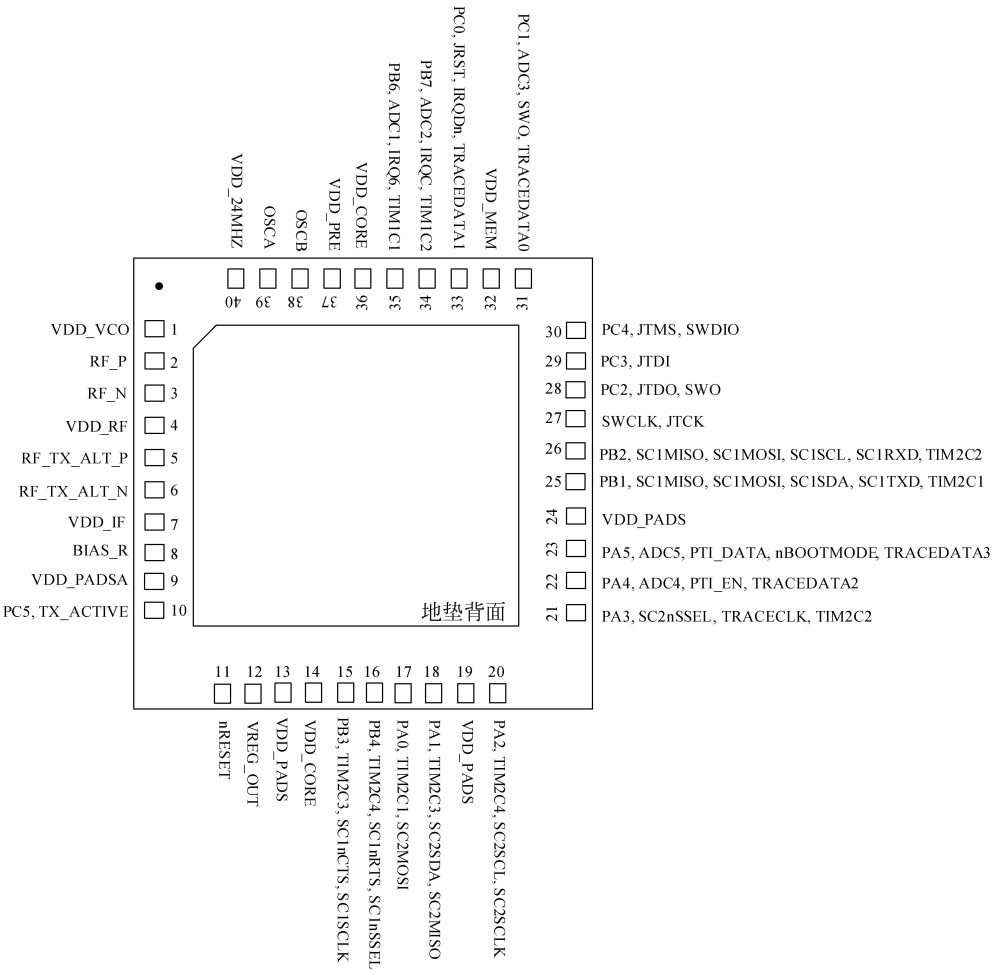


图 3.3 40 脚 VFQFPN 引脚分配

表 3.1 STM32W108 封装引脚描述

48 脚封装 引脚号	40 脚封装 引脚号	信 号	功 能	功 能 描 述
1	40	VDD_24MHz	电源	1.8V 高频晶振供电
2	1	VDD_VCO	电源	1.8V VCO 供电
3	2	RF_P	输入/输出	差分(RF_N)接收器输入/发送器输出
4	3	RF_N	输入/输出	差分(RF_P)接收器输入/发送器输出
5	4	VDD_RF	电源	1.8V RF 供电(LNA 和 PA)
6	5	RF_TX_ALT_P	输出	差分(RF_TX_ALT_N)发送器输出(可选)
7	6	RF_TX_ALT_N	输出	差分(RF_TX_ALT_P)发送器输出(可选)
8	7	VDD_IF	电源	1.8V IF 供电(混频器和滤波器)
9	8	BIAS_R	输入	偏置设置电阻
10	9	VDD_PADSA	电源	模拟电路供电(1.8V)
11	10	PC5	输入/输出	数字输入/输出

续表

48 脚封装 引脚号	40 脚封装 引脚号	信 号	功 能	功 能 描 述
		TX_ACTIVE	输出	逻辑层控制外部 Rx/Tx 开关,在 Tx 模式下 STM32W108 基带控制 TX_ACTIVE 并驱动它为高电平(VDD_PADS),用 GPIO_PCCFGH[7:4] 选择复用输出功能
12	11	nRESET	输入	低有效芯片复位(内部上拉)
13		PC6	输入/输出	数字输入/输出
		OSC32B	输入/输出	32.768kHz 晶振 用 GPIO_PCCFGH[11:8]选择模拟功能
		nTX_ACTIVE	输出	反向 TX_ACTIVE 信号(见 PC5) 用 GPIO_PCCFGH[11:8]选择复用输出功能
14		PC7	输入/输出	数字输入/输出
		OSC32A	输入/输出	32.768kHz 晶振 用 GPIO_PCCFGH[15:12]选择模拟功能
		OSC32_EXT	输入	数字 32kHz 时钟输入源
15	12	VREG_OUT	电源	稳压输出(唤醒时 1.8V,深睡眠时 0V)
16	13	VDD_PADS	电源	芯片供电(2.1~3.6V)
17	14	VDD_CORE	电源	1.25V 数字部分电源去耦
18		PA7	输入/输出 高电流	数字输入/输出 用 GPIO_DBGCFG[4]禁止 REG_EN
		TIM1_CH4	输出	定时器 1 通道 4 输出 用 TIM1_CCER 使用定时器输出 用 GPIO_PACFGH[15:12]选择复用输出功能 用 GPIO_DBGCFG[4]禁止 REG_EN
			输入	定时器 1 通道 4 输入(不能被重映射)
		REG_EN	输出	外部稳压器的开漏输出(复位后使能)
19	15	PB3	输入/输出	数字输入/输出
		TIM2_CH3 (见 22 引脚)	输出	定时器 2 通道 3 输出 用 TIM2_OR[6]使能重映射 用 TIM2_CCER 使用定时器输出 用 GPIO_PBCFGL[15:12]选择复用输出功能
			输入	定时器 2 通道 3 输入 用 TIM2_OR[6]使能重映射
		UART_CTS	输入	SC1 的 UART CTS 握手 用 SC1_UARTCFG[5]使能 用 SC1_MODE 选择 UART
		SC1SCLK	输出	SC1 的 SPI 主时钟 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[6]禁止重映射 用 SC1_SPICFG[4]使能主 SPI 用 SC1_MODE 选择 SPI 用 GPIO_PBCFGL[15:12]选择复用输出功能
			输入	SC1 的 SPI 从时钟 用 SC1_SPICFG[4]使能从 SPI 用 SC1_MODE 选择 SPI

续表

48 脚封装 引脚号	40 脚封装 引脚号	信 号	功 能	功 能 描 述
20	16	PB4	输入/输出	数字输入/输出
		TIM2_CH4 (见引脚 24)	输出	定时器 2 通道 4 输出 用 TIM2_OR[7]使能重映射 用 TIM2_CCER 使能定时器输出 用 GPIO_PBCFGH[3:0]选择复用输出功能
			输入	定时器 2 通道 4 输入 用 TIM2_OR[7]使能重映射
		UART_RTS	输出	SC1 的 UART RTS 握手 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[7]禁止重映射 用 SC1_UARTCFG[5]使能 用 SC1_MODE 选择 UART 用 GPIO_PBCFGH[3:0]选择复用输出功能
		SC1nSSEL	输入	SC1 的 SPI 从设置选择 用 SC1_SPICFG[4]使能从 SPI 用 SC1_MODE 选择 SPI
21	17	PA0	输入/输出	数字输入/输出
		TIM2_CH1 (见管脚 30)	输出	定时器 2 通道 1 输出 用 TIM2_OR[4]禁止重映射 用 TIM2_CCER 使能定时器输出 用 GPIO_PACFGL[3:0]选择复用输出功能
			输入	定时器 2 通道 1 输入 用 TIM2_OR[4]禁止重映射
		SC2MOSI	输出	SC2 的 SPI 主数据输出 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[4]使能重映射 用 SC2_SPICFG[4]使能主 SPI 用 SC2_MODE 选择 SPI 用 GPIO_PACFGL[3:0]选择复用输出功能
			输入	SC2 的 SPI 从数据输入 用 SC2_SPICFG[4]使能从 SPI 用 SC2_MODE 选择 SPI

续表

48 脚封装 引脚号	40 脚封装 引脚号	信 号	功 能	功 能 描 述
22	18	PA1	输入/输出	数字输入/输出
		TIM2_CH3 (见 19 引脚)	输出	定时器 2 通道 3 输出 用 TIM2_OR[6]禁止重映射 用 TIM2_CCER 使能定时器输出 用 GPIO_PACFGL[7:4]选择复用输出功能
			输入	定时器 2 通道 3 输入 用 TIM2_OR[6]禁止重映射
		SC2SDA	输入/输出	SC2 的 I ² C 数据输出 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[6]使能重映射 用 SC2_MODE 选择 I ² C 用 GPIO_PACFGL[7:4]选择复用开漏输出功能
		SC2MISO	输出	SC2 的 SPI 从数据输出 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[6]使能重映射 用 SC2_SPICFG[4]使能从 SPI 用 SC2_MODE 选择 SPI 用 GPIO_PACFGL[7:4]选择复用输出功能
			输入	SC2 的 SPI 主数据输入 用 SC2_SPICFG[4]使能从 SPI 用 SC2_MODE 选择 SPI
23	19	VDD_PADSa	电源	芯片供电(2.1~3.6V)
24	20	PA2	输入/输出	数字输入/输出
		TIM2_CH4 (见 20 管脚)	输出	定时器 2 通道 4 输出 用 TIM2_OR[7]禁止重映射 用 TIM2_CCER 使能定时器输出 用 GPIO_PACFGL[11:8]选择复用输出功能
			输入	定时器 2 通道 4 输入 用 TIM2_OR[7]禁止重映射
		SC2SCL	输入/输出	SC2 的 I ² C 时钟 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[7]使能重映射 用 SC2_MODE 选择 I ² C 用 GPIO_PACFGL[11:8]选择复用开漏输出功能
		SC2SCLK	输出	SC2 的 SPI 主时钟 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[7]使能重映射 用 SC2_SPICFG[4]使能主 SPI 用 SC2_MODE 选择 SPI 用 GPIO_PACFGL[11:8]选择复用输出功能
			输入	SC2 的 SPI 从时钟 用 SC2_SPICFG[4]使能从 SPI 用 SC2_MODE 选择 SPI

续表

48 脚封装 引脚号	40 脚封装 引脚号	信 号	功 能	功 能 描 述
25	21	PA3	输入/输出	数字输入/输出
		SC2nSSEL	输入	SC2 的 SPI 从选择 用 SC2_SPICFG[4]使能从 SPI 用 SC2_MODE 选择 SPI
		TRACECLK (见 36 引脚)	输出	同步 CPU 跟踪时钟 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[5]使能重映射 使能 ARM 核跟踪接口 用 GPIO_PACFGL[15:12]选择复用输出功能
		TIM2_CH2 (见 31 引脚)	输出	定时器 2 通道 2 输出 用 TIM2_CCER 使能定时器输出 用 TIM2_OR[5]禁止重映射 用 GPIO_PACFGL[15:12]选择复用输出功能
			输入	定时器 2 通道 2 输入 用 TIM2_OR[5]禁止重映射
26	22	PA4	输入/输出	数字输入/输出
		ADC4	模拟	ADC 输入 4 用 GPIO_PACFGH[3:0]选择模拟功能
		PTI_EN	输出	包跟踪接口(PTI)的帧信号 禁止 ARM 内核中的跟踪接口 用 GPIO_PACFGH[3:0]选择复用输出功能
		TRACEDATA2	输出	同步 CPU 跟踪数据位 2 在 ARM 内核中选择 4 线同步跟踪接口 使能 ARM 内核中的跟踪接口 用 GPIO_PACFGH[3:0]选择复用输出功能
27	23	PA5	输入/输出	数字输入/输出
		ADC5	模拟	ADC 输入 5 用 GPIO_PACFGH[7:4]选择模拟功能
		PTI_DATA	输出	包跟踪接口(PTI)的数据信号 禁止 ARM 内核中的跟踪接口 用 GPIO_PACFGH[7:4]选择复用输出功能
		nBOOTMODE	输入	复位后嵌入式串行引导激活,在 NRST 复位期间和复位之后信号立即有效,详细信息见复位模块的有关章节
		TRACEDATA3	输出	同步 CPU 跟踪数据位 3 在 ARM 内核中选择 4 线同步跟踪接口 使能 ARM 内核中的跟踪接口 用 GPIO_PACFGH[7:4]选择复用输出功能
28	24	VDD_PADS	电源	芯片供电(2.1~3.6V)

续表

48 脚封装 引脚号	40 脚封装 引脚号	信 号	功 能	功 能 描 述
29		PA6	输入/输出 高电流	数字输入/输出
		TIM1_CH3	输出	定时器 1 通道 3 输出 用 TIM1_CCER 使能定时器输出 用 GPIO_PACFGH[11:8]选择复用输出功能
			输入	定时器 1 通道 3 输入(不能被重映射)
30	25	PB1	输入/输出	数字输入/输出
		SC1MISO	输出	SC1 的 SPI 从数据输出 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[4]禁止重映射 用 SC1_MODE 选择 SPI 用 SC1_SPICR 选择从 SPI 用 GPIO_PBCFGL[7:4]选择复用输出功能
		SC1MOSI	输出	SC1 的 SPI 主数据输出 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[4]禁止重映射 用 SC1_MODE 选择 SPI 用 SC1_SPICR 选择主 SPI 用 GPIO_PBCFGL[7:4]选择复用输出功能
		SC1SDA	输入/输出	SC1 的 I ² C 数据 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[4]禁止重映射 用 SC1_MODE 选择 I ² C 用 GPIO_PBCFGL[7:4]选择复用开漏输出功能
		SC1TXD	输出	SC1 的 UART 数据传输 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[4]禁止重映射 用 SC1_MODE 选择 UART 用 GPIO_PBCFGL[7:4]选择复用输出功能
		TIM2_CH1 (见管脚 21)	输出	定时器 2 通道 1 输出 用 TIM2_CCER 使能定时器输出 用 TIM2_OR[4]使能重映射 用 GPIO_PACFGL[7:4]选择复用输出功能
			输入	定时器 2 通道 1 输入 用 TIM2_OR[4]禁止重映射

续表

48 脚封装 引脚号	40 脚封装 引脚号	信 号	功 能	功 能 描 述
31	26	PB2	输入/输出	数字输入/输出
		SC1MISO	输入	SC1 的 SPI 主数据输入 用 SC1_MODE 选择 SPI 用 SC1_SPICR 选择主 SPI
		SC1MOSI	输入	SC1 的 SPI 从数据输入 用 SC1_MODE 选择 SPI 用 SC1_SPICR 选择从 SPI
		SC1SCL	输入/输出	SC1 的 I ² C 时钟 用 TIM2_CCER 禁止定时器输出或用 TIM2_OR[5]禁止重映射 用 SC1_MODE 选择 I ² C 用 GPIO_PBCFGL[11;8]选择复用开漏输出功能
		SC1RXD	输入	SC1 的 UART 数据接收 用 SC1_MODE 选择 UART
		TIM2_CH2 (见管脚 25)	输出	定时器 2 通道 2 输出 用 TIM2_CCER 使能定时器输出 用 TIM2_OR[5]使能重映射 用 GPIO_PACFGL[11;8]选择复用输出功能
			输入	定时器 2 通道 2 输入 用 TIM2_OR[5]使能重映射
32	27	SWCLK	输入/输出	串行线调试器时钟输入/输出 在串行线模式下选择(见 JTMS 描述,35 引脚)
		JTCK	输入	来自调试器的 JTAG 时钟输入 在 JTAG 模式下选择(JTAG 模式为默认的模式,见 JTMS 描述,引脚 35) 使用内部下拉
33	28	PC2	输入/输出	数字输入/输出 用 GPIO_DBGCFG[5]使能
		JTDO	输出	JTAG 数据输出到调试器 在 JTAG 模式下选择(JTAG 模式为默认的模式,见 JTMS 描述,引脚 35)
		SWO	输出	串行输出异步跟踪数据到调试器 在 ARM 内核中选择异步跟踪接口 ARM 内核使能跟踪接口 用 GPIO_PCCFGL[11;8]选择复用输出功能 使能串行线模式(见 JTMS 描述,35 引脚) 内部上拉使能
34	29	PC3	输入/输出	数字输入/输出 用 GPIO_DBGCFG[5]使能数字 IO 或串行线模式(见 JTMS 描述,35 引脚)
		JTDI	输入	调试器的 JTAG 数据输入 在 JTAG 模式选择(JTAG 模式为默认的模式,见 JTMS 描述,引脚 35) 内部上拉使能

续表

48 脚封装 引脚号	40 脚封装 引脚号	信 号	功 能	功 能 描 述
35	30	PC4	输入/输出	用 GPIO_DBGCFG[5]使能数字输入/输出
		JTMS	输入	来自调试器的 JTAG 模式选择 在 JTAG 模式选择(JTAG 模式为默认的模式) JTAG 模式在上电之后或强制将 NRST 位置低时使能 通过调试器,使用 ARM 定义的协议可以选择串行线模式 内部上拉使能
		SWDIO	输入/输出	串行线双向数据,来自或发送到调试器的数据 使能串行线模式(见 JTMS 描述) 通过调试器,使用 ARM 定义的协议可以选择串行线模式 内部上拉使能
36		PB0	输入/输出	数字输入/输出
		VREF	模拟输出	ADC 参考输出 用 GPIO_PBCFGL[3:0]使能模拟功能
		VREF	模拟输入	ADC 参考输入 用 GPIO_PBCFGL[3:0]使能模拟功能 用一个 ST 系统功能使能参考输出
		IRQA	输入	外部中断源 A
		TRACECLK (见 25 引脚)	输出	同步 CPU 跟踪时钟 在 ARM 内核上使能跟踪接口 用 GPIO_PBCFGL[3:0]选择复用输出功能
		TIM1CLK	输入	定时器 1 外部时钟输入
		TIM2MSK	输入	定时器 2 外部时钟屏蔽输入
37		VDD_PADSa	电源	芯片供电(2.1~3.6V)
38	31	PC1	输入/输出	数字输入/输出
		ADC3	模拟	ADC 输入 3 用 GPIO_PCCFGL[7:4]使能模拟功能
		SWO (见 33 引脚)	输出	串行输出异步跟踪数据到调试器 在 ARM 内核中选择异步跟踪接口 ARM 内核使能跟踪接口 用 GPIO_PCCFGL[7:4]选择复用输出功能
		TRACEDATA0	输出	同步 CPU 跟踪数据位 0 在 ARM 内核中选择 1 线、2 线或 4 线同步跟踪接口 ARM 内核使能跟踪接口 用 GPIO_PCCFGL[7:4]选择复用输出功能
39	32	VDD_MEM	电源	1.8V 供电(Flash, RAM)

续表

48 脚封装 引脚号	40 脚封装 引脚号	信 号	功 能	功 能 描 述
40	33	PC0	输入/输出 高电流	数字输入/输出 用 GPIO_DBFCFG[5]使能数字 IO 或使能串行 线模式(见 JTMS 描述,引脚 35)同时禁止 TRACEDATA1
		JRST	输入	调试器的复位输入 在 JTAG 模式(见 JTMS 描述,引脚 35)选择 JTAG 调试,并且禁止 TRACEDATA1 使能内部上拉
		IRQD	输入	默认的外部中断源 D
		TRACEDATA1	输出	同步 CPU 跟踪数据位 1 在 ARM 内核中选择 2 线或 4 线同步跟踪接口 ARM 内核使能跟踪接口 用 GPIO_PCCFGL[3:0]选择复用输出功能
41	34	PB7	输入/输出 高电流	数字输入/输出
		ADC2	模拟	ADC 输入 2 用 GPIO_PBCFGH[15:12]使能模拟功能
		IRQC	输入	默认的外部中断源 C
		TIM1_CH2	输出	定时器 1 通道 2 输出 用 TIM1_CCER 使能定时器输出 用 GPIO_PBCFGH[15:12]选择复用输出功能
			输入	定时器 1 通道 2 输入(不能被重映射)
42	35	PB6	输入/输出 高电流	数字输入/输出
		ADC1	模拟	ADC 输入 1 用 GPIO_PBCFGH[11:8]使能模拟功能
		IRQB	输入	默认的外部中断源 B
		TIM1_CH1	输出	定时器 1 通道 1 输出 用 TIM1_CCER 使能定时器输出 用 GPIO_PBCFGH[11:8]选择复用输出功能
			输入	定时器 1 通道 1 输入(不能被重映射)
43		PB5	输入/输出	数字输入/输出
		ADC0	模拟	ADC 输入 0 用 GPIO_PBCFGH[7:4]使能模拟功能
		TIM2CLK	输入	定时器 2 外部时钟输入
		TIM1MSK	输入	定时器 2 外部时钟屏蔽输入
44	36	VDD_CORE	电源	1.25V 数字内核电源去耦
45	37	VDD_PRE	电源	1.8V 预分频器供电
46		VDD_SYNTH	电源	1.8V 合成器供电
47	38	OSCB	输入/输出	24MHz 晶振,当使用外部时钟输入到 OSCA 时浮空
48	39	OSCA	输入/输出	24MHz 晶振或外部时钟输入
49	41	GND	地	在封装底部中央的接地焊盘

注意：使用 GPIO_IRQCSEL 和 GPIO_IRQDSEL 寄存器,IRQC 和 IRQD 外部中断能够被映射到任何的数字输入/输出引脚。

3.3 思考与练习题

1. STM32W108 是一个集成有多种功能于一体的微处理器,请指出其片内的功能单元,并简单描述其 IEEE 802.15.4 标准的收发器的构成。
2. STM32W108 支持哪几种大小的数据 SRAM 和可编程 Flash?
3. STM32W108 有哪几种封装定义,我们在硬件设计过程中应该如何做出选择?
4. STM32W108 有多少个 GPIO 口? 有哪些 GPIO 口是复用的?