

厦门国科安芯科技有限公司

AS32S601 数据手册

32-bit RISC-V MCU

目 录

1 芯片介绍	2
1.1 特色	3
1.2 典型应用场景	4
2 芯片概述	5
2.1 内核	5
2.2 总线架构	5
2.3 时钟管理	6
2.4 安全概述	7
2.5 存储系统	8
2.5.1 内置 SRAM	8
2.5.2 片内 Flash 存储器	8
2.6 QSPI	9
2.7 DMA	10
2.8 MPU	11
2.9 PMU	11
2.10 SMU	12
2.11 CMU	12
2.12 WDG	13
2.13 DSE	14
2.14 PLIC	14
2.15 CLINT	15
2.16 RTC	15
2.17 CRC	15
2.18 ADC	16
2.19 DAC	17
2.20 TIMER	17
2.21 CAN	18
2.22 USART	19
2.23 SPI	20
2.24 IIC	21
2.25 MAC	21
2.26 GPIO	22
2.27 DEBUG	22
2.28 地址映射	23
3 封装	27

1 芯片介绍

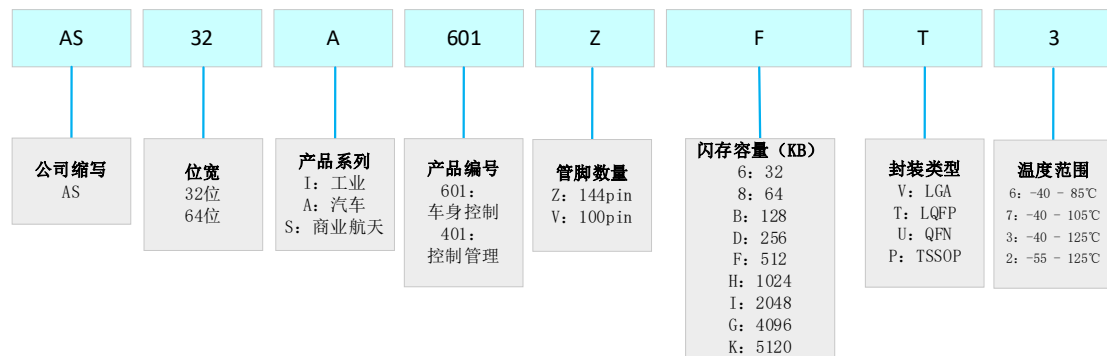
AS32S601 是国科安芯公司研制的一款基于 32 位 RISC-V 指令集 MCU 产品。产品具有丰富的 Flash 容量、支持 ASIL-B 等级的功能安全 ISO26262，同时具有高安全、低失效、多 IO、低成本等特点。整个产品系列有 AS32I601、AS32A601、AS32S601 三个产品等级。

- ◆ 工作频率高达 180MHz
- ◆ 工作输入电压支持：2.7V~5.5V
- ◆ 休眠电流：≤200uA(可唤醒)
- ◆ 典型工作电流：≤50mA
- ◆ 符合 AEC-Q100 grade1 认证标准（汽车级）
- ◆ SEU：≥75 MeV.cm²/mg 或 10⁻⁵次/器件.天（商业航天级）
- ◆ SEL：≥75 MeV.cm²/mg（商业航天级）
- ◆ 封装工艺：LQFP144

芯片型号如下：

芯片类型	芯片型号	等级	封装	温度范围
MCU	AS32I601ZIT6	工业级	LQFP144	-40 to 85 °C
MCU	AS32A601ZIT3	汽车级	LQFP144	-40 to 125 °C
MCU	AS32S601ZIT2	商业航天级	LQFP144	-55 to 125 °C

芯片命名规则



1.1 特色

模块	说明
内核	自研 E7 内核，带有 FPU 与 L1Cache：16KiB 数据缓存，16KiB 指令缓存，允许零等待访问嵌入式 Flash 与外部内存，最高频率 180MHz，482DIMPS/2.68DIMPS/MHz。
时钟	● 外部晶振（OSC）：范围在 8MHz~40MHz ● 内部高频振荡器（FIRC）：16MHz ● 内部低频振荡器（SIRC）：32KHz ● 系统锁相环（PLL）：最大支持 500MHz 输出
存储	● 512KiB 内部 SRAM（带 ECC） ● 16KiB ICache 和 16KiB DCache（带 ECC） ● 512KiB D-Flash（带 ECC） ● 2MiB P-Flash（带 ECC）
系统	● 2 个 16 通道的 DMA 模块 ● 5 个内存保护模块（MPU） ● 4 个时钟监测模块（CMU） ● 1 个错误控制模块（FCU） ● 1 个电源管理模块（PMU） ● 1 个系统控制模块（SMU） ● 1 个实时计数器模块（RTC） ● 1 个硬件加密模块（DSU），支持 AES、SM2/3/4 和 TRNG ● 1 个核心中断控制模块（CLINT） ● 1 个外部中断控制模块（PLIC） ● 1 个 CRC 校验模块
电源管理	● 4 种电源管理模式：RUN，SRUN，SLEEP，DEEPSLEEP ● 低电压检测和复位功能（LVD/LVR） ● 高电压检测功能（HVD）
模拟接口	● 3 个 12 位的模数转换器（ADC），最多支持 48 通道模拟

	通路 ● 2 个模拟比较器（ACMP） ● 2 个 8 位的数模转换器（DAC） ● 1 个温度传感器
定时器	● 4 个 32 位高级定时器 ● 4 个 16 位通用定时器
通信接口	● 6 路 SPI，支持主从模式标准 SPI 协议，速率最高可达 30MHz ● 4 路 CAN，支持 CANFD ● 4 路 USART 模块，支持 LIN 模式、同步串口模式 ● 1 个以太网（MAC）模块，支持 10/100M 模式、全/半双工模式 ● 2 路 IIC，支持主从模式标准 IIC 协议
调试接口	满足 RISC-V Debug Spec 0.13.2 标准的带有 JTAG 接口的调试器

1.2 典型应用场景

AS32I601ZIT6

- 工业领域：机器人控制（关节控制、通信管理控制）、工业通用控制系统、自动化控制系统（平台升降机控制）

AS32A601ZIT3

- 汽车领域：BCM 车身控制系统（内外灯光控制、中控锁控制、车窗控制、喇叭控制、后视镜控制等）、电机驱动系统（引擎散热风扇、水泵、高低压压缩机等）

AS32S601ZIT2

- 商业航天：运动控制、信号系统

2 芯片概述

2.1 内核

带有硬件 FPU 的 E7 内核是自行研发的最新一代的嵌入式处理器。它的设计目标是低成本、低功耗、高实时性、高安全性，在这基础之上同时提供优秀的运算性能。

E7 处理器是一款高效的高性能处理器：

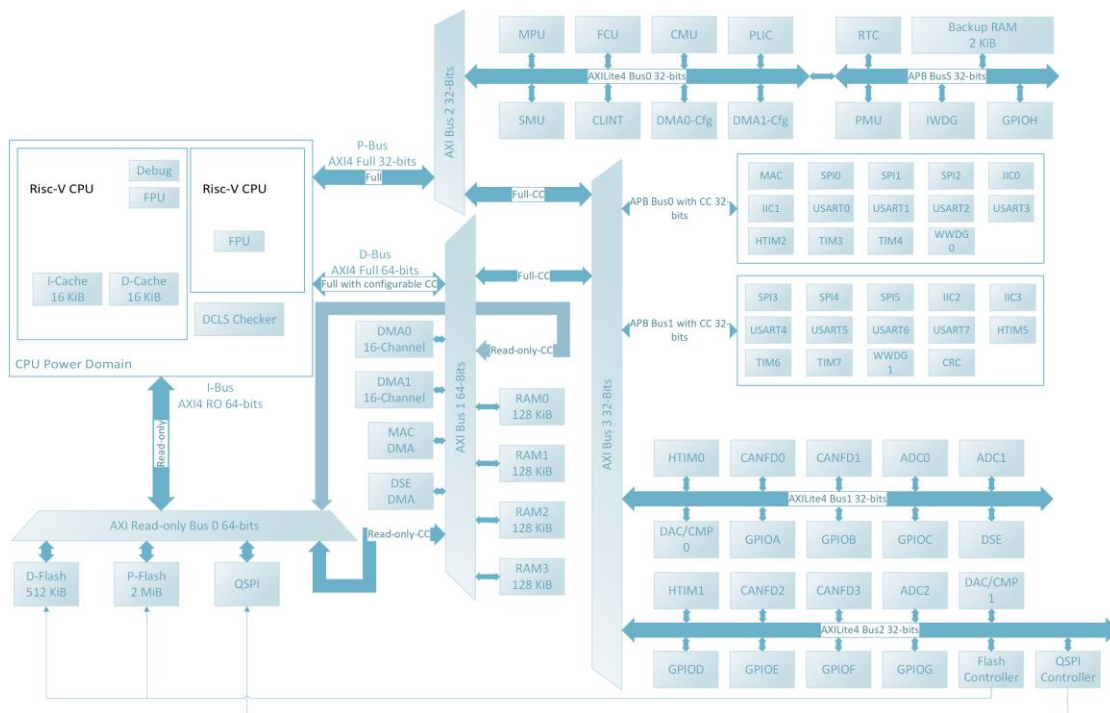
- 8 级双发射流水线
- 动态分支预测
- 哈弗架构的缓存（16 KiB 的 I-cache 和 16KiB 的 D-cache）
- 64 位 AXI4 总线接口

处理器支持以下的内存接口：

- 指令 AXI 接口（AXI2）
- 数据 AXI 接口（AXI4）
- 低延迟外设 AXI 接口（AXI4P）

它内建的双浮点 FPU（浮点单元）可以加速浮点相关的软件运行。

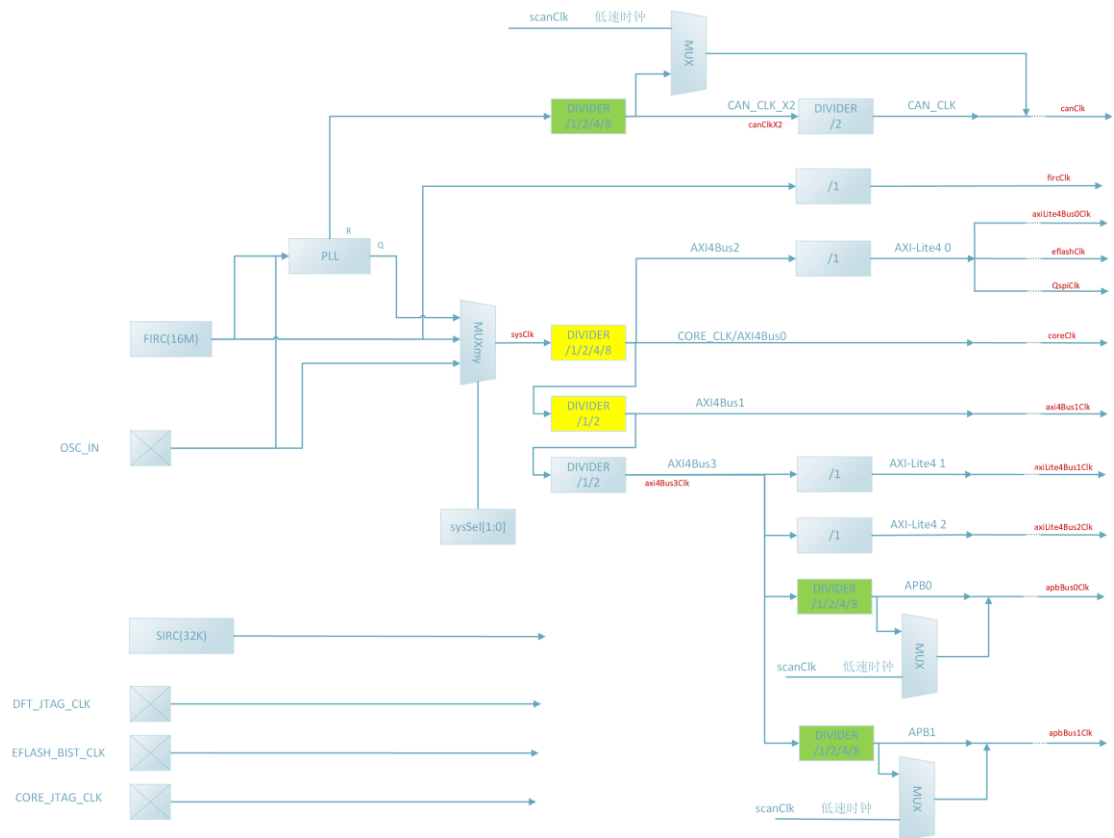
2.2 总线架构



MCU 的总线的 AXI Crossbar 是一个总线矩阵，用于互联 CPU 内核与系统存储器及外设模块的访问（读写）。

- Crossbar 的主机可以主动发起数据访问请求，而从机则只能被动接受访问
- Crossbar 的每个主机/从机与总线之间都有 ECC 编解码模块
- 每个端口都有独立的控制总线、地址总线和数据总线
- 不同的主机可以同时访问不同的从机，从而保证 MCU 系统工作带宽
- 主机对任意从机的保护均受 MPU 的保护

2.3 时钟管理



时钟管理模块为 MCU 提供时钟选择。包括快速内部时钟（FIRC）、内部慢速时钟（SIRC）、外部晶振（OSC）和 PLL 作为时钟源。其中 PLL 的参考时钟可以来源于 FIRC，也可以来源于 OSC。

在复位后 CPU 工作的默认时钟是来自 FIRC。FIRC 是 16MHz 的内部 RC 振荡器，其误差精度在 1%左右。然后应用程序可以选择 FIRC 或者外部的 8-40MHz 晶振作为时钟源。如果时钟监测模块检测到 OSC 时钟不稳定，则自动切换到 FIRC，并通过中断的方式通知应用程序。PLL 的倍频时钟可以高达 500MHz。时钟监测模块检测到 PLL 输出的时钟出现错误时，也会通过中断告知应用程序。

2.4 安全概述

系统针对需要高安全完整性的相关系统，进行了安全设计，来保证 MCU 的功能安全。对于不同的模块，保护措施也有所不同：

- 对于内核类的作为主机对系统进行操作的设备，采用延迟锁步方法保证安全

- 存储器及存储器和外设数据路径的安全由端到端 ECC 保护。对于外设，端到端保护在 APB 转接桥处结束
- 时钟由多个分立的 CMU 进行监控
- 电源由 PMU 进行监控
- 外设的安全由应用级措施确保（多外设同时访问同一单元），硬件支持此应用级冗余，方法是提供连接至不同外设桥的 IO 模块，以最大化被监控和监控资源之间的独立性
- MBIST 用于避免功能逻辑和安全机制中的潜在故障累积。专用机制用于检查安全机制的可用性及每个错误反应路径的功能
- 故障收集单元负责收集故障并作出反应
- 对于错误事件（包括 ECC 校正与检测、软件配置的验证错误）FDU（Fault Detect Unit）负责收集并向 FCU（Fault Control Unit）报告系统的错误事件
- 共因故障（Common Cause Failure, CCF）通过一组措施处理，以控制和避免跨系统级方法（温度及非功能信号监控）和后端技术（物理隔离硅区域、路由限制）的 CCF
- 通过分层处理器保护方法确保运行干扰保护，以允许不同 ASIL 标准的软件并行运行

MCU 集成的安全设计能够在高诊断范围内检测单点故障和潜在故障。但不是所有的共模故障都可以检测到。为了实现安全要求，需要在系统层面上对硬件环境与软件环境进行一定的要求，此项要求请参考安全手册。

2.5 存储系统

2.5.1 内置 SRAM

- 多达 512KiB 内部 SRAM;
- 提供 16KiB ICache;
- 提供 16KiB DCache;
- 支持误码校正（ECC）。

2.5.2 片内 Flash 存储器

片内 Flash 共包含两个存储器，分别为程序存储器（PFlash）和数据存储器（DFlash）。

Flash 控制器作为 CPU 内核与片内 Flash 之间的桥梁，可用于管理任何主设备对片内 Flash 进行的访问。Flash 控制器可对 Flash 执行读取、编程和擦除操作，并实施写保护和读保护等安全机制。

- EFlash 存储器
 - PFlash 最大支持 2MB（包括 4 个 block，即 4×512KB）
 - DFlash 最大支持 512KB（包括 1 个 block）
 - 寿命：≥100,000 周期
 - 块(Block)容量:512KB/block
 - 扇区(Sector)容量：4KB/sector
 - 行(Row)容量：512B/row
- EFlash 控制器
 - 操作列表：
 - 擦除：扇区擦除、块擦除、写保护信息区擦除、全片擦除
 - 编程：行编程，编程最小单位 64-bit
 - 读：支持 8-bit/16-bit/32-bit/64-bit 宽度读数据
 - 安全措施：
 - 支持写保护功能
 - 支持读保护功能
 - 可通过安全密钥临时解锁读保护功能
- EFlash MBIST
 - 支持 EFlash MBIST 测试
 - 支持 IEEE1149.7 接口
 - 支持自动修复功能

2.6 QSPI

QSPI Flash 控制器主要用于完成系统对外部 QSPI Flash 的访问控制，按照 QSPI 接口协议与 Flash 进行通信。该控制器主要包括以下三种工作模式：

- 间接模式：AXI Lite 总线控制寄存器实现全部操作；
- 自动轮询模式：周期性读取外部 Flash 状态寄存器，查询外部 Flash 状态，如是否完成烧写或擦除等操作；
- 内存映射模式：外部 Flash 映射到 CPU 地址空间，直接根据 CPU 地址从外部 Flash 中读取数据。

2.7 DMA

- 直接存储器访问（DMA）控制器用于在外设和存储器之间，或存储器与存储器之间的高速数据传输，无需 CPU 干预，数据可以通过 DMA 快速移动，提升了微处理器的性能。单个 DMA 有 16 个通用通道用于不同类型的外设，支持 USART、SPI、ADC、TIM。DMA 控制器具有一个轮询仲裁器来协调通道间的优先级。此外，Ethernet MAC 与 DSE 使用自己独立的 DMA。16 个独立的可配置通道
- 每个 DMA 通道都支持 USART、SPI、ADC、TIM 的 DMA 请求，每个通道上都同样支持存储器到存储器搬移操作。该配置可通过软件完成
- 在同一个 DMA 模块上，多个请求间的优先级可以通过软件编程设置(4 个等级分别为：很高、高、中等和低)，优先级设置相等时由硬件决定（通道 0 优先级最高，通道 15 优先级最低，依次类推）
- 独立数据源和目标数据区的传输大小(字节，半字，字)，源/目标地址必须按数据传输宽度对齐
- 每个通道都支持 DMA 传输完成和 DMA 传输错误等状态标志位和中断，这些状态标志逻辑或成为一个单独的中断请求
- 存储器和存储器间的传输，外设和外设间的传输
- 外设和存储器、存储器和外设间的传输
- SRAM、外设、Flash(只读)均可作为访问的源和目标
- 可编程的数据传输字节长度：最大为 32767 bytes
- 可选的双核锁步的安全备份
- 内置超时看门狗

2.8 MPU

存储保护单元（MPU）作为 MCU 的安全机制之一，提供区域保护功能。MPU 可以限定总线主机对选定地址区域的内存/外设的访问权限。可以对每个总线上的主机独立配置保护存储器区域的不同访问权限。

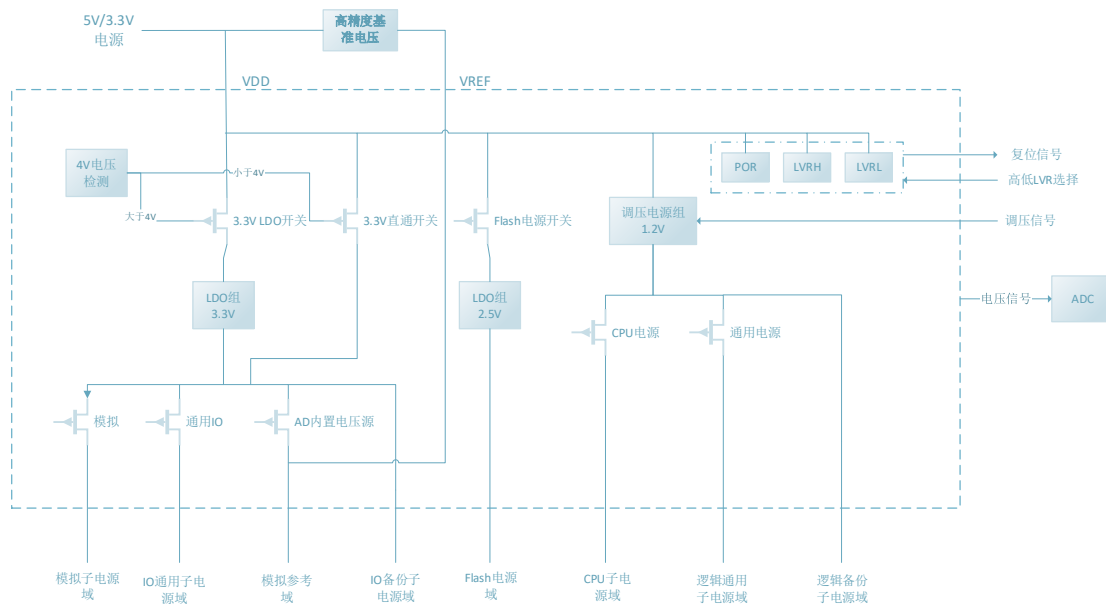
MPU 特别有助于那些必须保护的某些关键或认证代码免受其他任务不当行为影响的应用程序。它通常由实时操作系统（RTOS）管理。如果程序访问 MPU 禁止的存储位置，RTOS 可以检测到它并采取行动。在 RTOS 环境中，内核可以根据要执行的进程动态更新 MPU 区域设置。

MPU 具有以下特性

- 任意保护宽度，特定阈值间的任意保护范围；
- 16 个（CPU I/D/P MPU）/8 个（DMA MPU）互相独立的内存保护通道，各个通道之间的保护区域可叠加；
- 单周期快速异常处理；
- 在命中时偏移访问地址以防止出现非法访问情况；
- 支持三种保护模式：读保护，写保护，或锁定区域（读保护+写保护）。

2.9 PMU

PMU 为一整体的电源管理模块，可进行电源开关的控制与 1.2V 电源域的调压，并有上电复位（POR）与低压复位（LVR）功能，整体框图如下：



电源管理控制器（PMU）为软件提供了电源方面的系统管理功能，包含睡眠/深度睡眠/唤醒功能的控制，电源监测功能（LVR/LVD/HVD）。

- 支持低功耗模式（SLEEP/DEEP_SLEEP/WAKEUP）的切换
- 支持唤醒源的配置（外部 IO/RTC/IWDG）
- 支持 FIRC 的开启与关闭

2.10 SMU

系统管理单元（SMU）的主要功能是完成时钟和复位的管理。在默认状态下 SMU 工作在 IDLE 状态。只有接收到 PMU 的使能信号后才开始工作。SMU 模块会根据 PMU 的指令自动配置 COR、AXIBUS0/1/2 等总线的时钟和复位。

- 支持系统时钟的切换（可以工作在 OSC 时钟、FIRC 时钟和 PLL 时钟）
- 支持总线 and 外设复位的单独管理
- 支持总线 and 外设时钟的单独使能管理
- 支持对 PLL 的单独配置
- 支持独立的总线分频器
- 支持根据 PMU 的指令自动配置 CORE 等时钟和复位

2.11 CMU

时钟监控单元(Clock Monitor Unit, CMU)主要有两大功能：

- 1、检测时钟频率的偏差是否超出最大的允许范围；
- 2、检测时钟是否丢失。

其主要特征包括：

- 如果被监控的频率大于参考时钟的最大值（HFREF），产生一个频率大于高阈值（FHH）事件
- 如果被监控的频率小于参考时钟的最小值（LFREF），产生一个频率小于低阈值（FLL）事件
- 当被监控的时钟丢失时，会产生 FLL 事件
- 参考时钟计数的窗口时间支持软件配置

2.12 WDG

MCU 内置独立看门狗（IWDG）和窗口看门狗（WWDG）。

独立看门狗（IWDG）是一个独立定时器，一般用来检测系统软件程序是否按预期运行。如果看门狗模块没有被按时刷新，看门狗会产生错误上报给 FCU。比如程序中存在死循环，但因某些原因没有跳出，或者使用多任务操作系统，喂狗任务没有按时得到执行，都会由于看门狗模块没有被刷新而产生错误，一般用于高安全性场合。

- 内部晶振独立时钟源
- 可编程 8 位时钟预分频器
- 可编程 16 位时钟分频器
- 可编程 16 位超时时间计数
- 锁定安全机制

窗口看门狗（WWDG）是一个独立定时器，一般用来检测系统软件程序是否按预期运行。如果看门狗模块没有及时或过早刷新，看门狗会产生错误上报给 FCU。比如程序中存在死循环，但因某些原因没有跳出，或者使用多任务操作系统，喂狗任务没有按时得到执行，都会由于看门狗模块没有被刷新而产生错误，一般用于高安全性场合。

- 每个控制器 4 个独立通道
- 可编程 8 位时钟预分频器

- 可编程 16 位时钟分频器
- 可编程 16 位超时时间计数
- 可编程 16 位窗口时间计数
- 锁定安全机制

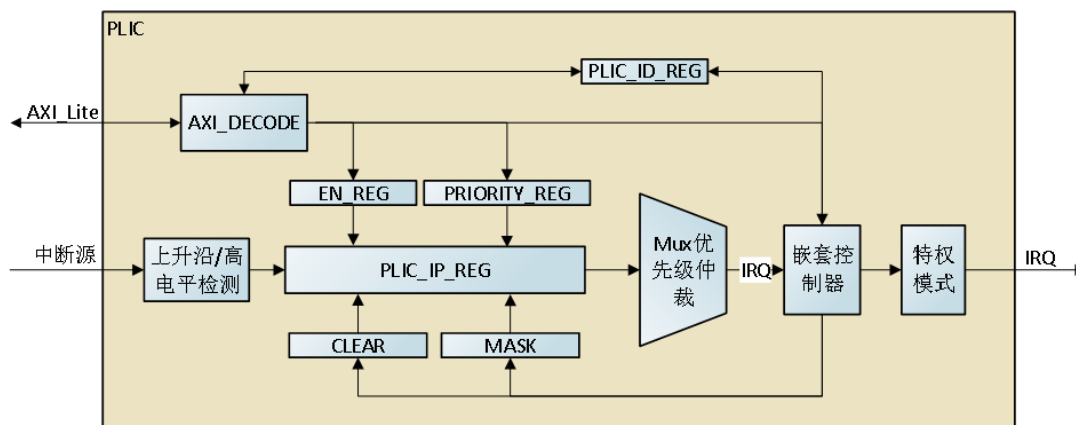
2.13 DSE

DSE（Device Security Engine）是硬件加解密模块，符合 HIS-SHE 安全规范标准。

- AES 的密钥长度支持 128bit、192bit 和 256bit
- AES 的加密方式支持 ECB、CBC 和 OFB
- 支持 AES CMAC 生成和验证
- 支持 SM2、SM3 和 SM4
- 支持真随机数（TRNG）的生成

2.14 PLIC

系统平台中断控制器（Platform Level Interrupt Controller, PLIC）是 CPU 的中断控制器，主要对中断源进行采样，优先级仲裁和分发。各外设中断统一连到 PLIC，PLIC 统一管理并输出中断请求到内核。



- 支持 64 个中断源
- 每个中断源分配一个 ID，从 1 开始，0 代表无中断
- 支持设置每个中断源优先级，共计 64 级，可以设置为相同优先级，63

级最高，0 级最低，优先级一样时，ID 号小的先执行

- 支持设置中断使能位，注意区别于接口私有的中断使能，这两种需要全部开启
- 输出单周期中断请求到内核，相同时间只有一个最高优先级中断输出，执行中断函数时不响应相同中断
- 中断源上升沿触发和高电平触发
- 支持中断嵌套（保存现场和恢复现场时不支持中断嵌套）
- 保存恢复现场固定为软件方式

2.15 CLINT

内核本地中断控制器（CLINT）模块产生 CSR 寄存器的定时器中断、软件中断和计时器。该模块采用 AXILite 接口，支持定时器中断、软件中断和计时器输出。

2.16 RTC

- 实时时钟 (RTC) 是一个独立的 BCD 定时器/计数器。RTC 提供具有可编程闹钟中断功能的日历时钟/日历。RTC 还包含具有中断功能的周期性可编程唤醒标志。两个 32 位寄存器包含 BCD 格式的十分之一秒、秒、分钟、小时、星期几、日期、月份和年份。此外,还可提供二进制格式的亚秒值。系统可以自动将月份的天数补偿为 28、29(闰年)、30 和 31 天。包含十分之一秒、秒、分钟、小时、星期几、日期、月份和年份的日历；
- 具有中断功能的可编程闹钟。可通过任意日历字段的组合触发闹钟；
- 自动唤醒单元,可周期性地生成标志以触发自动唤醒中断。

2.17 CRC

循环冗余校验（Cyclic Redundancy Check, CRC）是一种根据网络数据包或计算机文件等数据产生简短固定位数校验码的一种信道编码技术，主要用来检测或校验数据传输或者保存后可能出现的错误。它是利用除法及余数的原

理来作错误侦测的。

CRC（循环冗余校验）计算单元使用一个多项式发生器从 8 位/16 位/32 位的数据字中产生 **CRC** 码。

在众多的应用中，基于 **CRC** 的技术还常用来验证数据传输或存储的完整性。根据功能安全标准的规定，这些技术提供了验证 **Flash** 完整性的方法。**CRC** 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

- 使用位数可预设定的（7 位、8 位、16 位和 32 位）完全可编程多项式；
- 处理 8 位、16 位、32 位数据大小
- 可编程 **CRC** 初始值
- 单输入/输出 32 位数据寄存器
- 输入缓冲器可避免计算期间发生总线阻塞
- 对于 32 位数据大小，**CRC** 计算在 4 个 **APB** 时钟周期（**PCLK**）内完成
- 8 位通用寄存器（可用于临时存储）
- **I/O** 数据的可逆性选项
- 支持异或计算

2.18 ADC

MCU 中共有 3 个 **ADC** 模块，其 **ADC** 采用的是逐次逼近型模拟数字转换器，分辨率为 12bit，拥有最多 16 路外部通道。

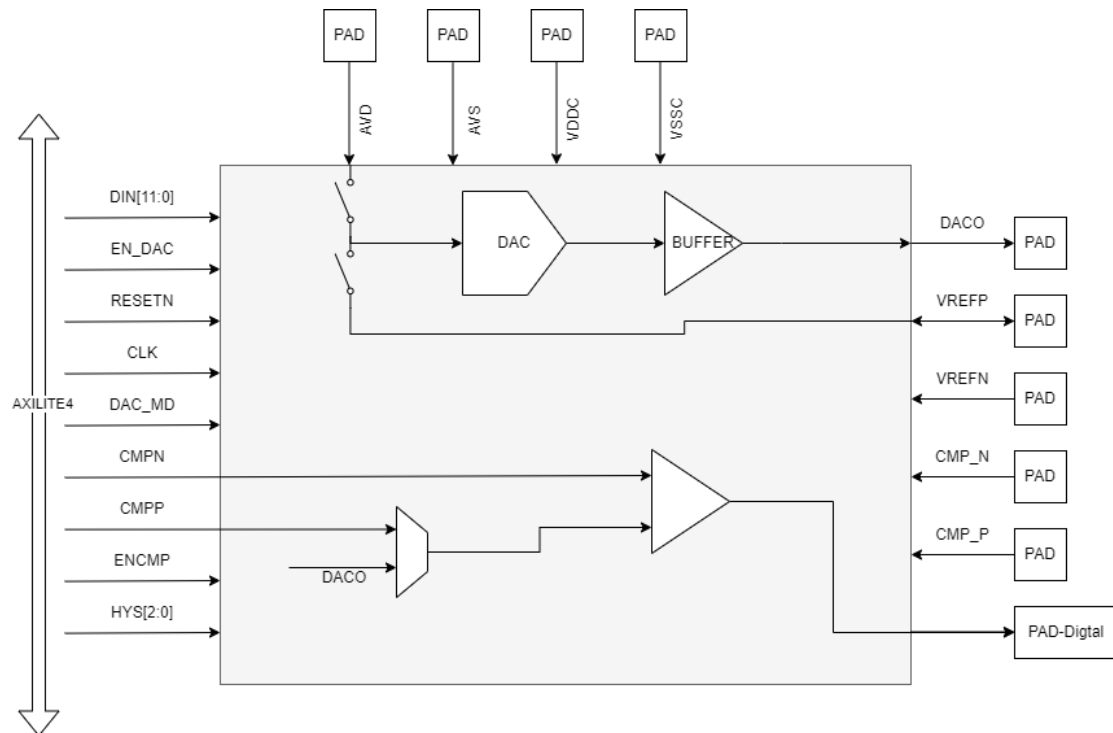
该 **ADC** 支持单次、连续、扫描或间接转换等多种工作模式；支持模拟监控器功能，可以监测输入电压是否超过用户设定的电压范围，并可在超出范围时发送中断。**ADC** 转换结果可按照左对齐或右对齐的方式存储在 16 位数据寄存器中。

其主要特征包括：

- 3 个独立 **ADC** 模块
- 转换序列分为规则组（**regular group**）和注入组（**injection group**）
- 8 种工作模式
- 通过内部软件触发或外部硬件触发启动 **ADC**

- 模拟监控器功能
- DMA 访问，仅用于规则组通道

2.19 DAC



- 8 位分辨率
- 集成非线性: ± 2 LSB
- 差分非线性: ± 1 LSB
- 模拟电源操作: 2.97V ~ 3.63V
- 设置时间: $\leq 1\mu s$
- 支持数字断电保护

2.20 TIMER

芯片内共包含 8 个定时器，其中 4 个高级定时器、4 个通用定时器。

高级定时器特性：

- 32 位递增、递减、递增/递减自动重载计数器
- 32 位可编程预分频器，用于对计数器时钟频率进行分频
- 多达 6 个独立通道，可用于：

- 输入捕获
- 输出比较
- PWM 生成
- 单脉冲模式输出
- 可编程死区的互补输出
- 使用外部信号控制定时器，可实现多个定时器互联
- 16 位重复计数器
- 断路输入，用于将定时器的输出信号置于用户可选的安全配置中
- 发生如下事件时生成中断/DMA 请求：
 - 更新：计数器上溢/下溢、计数器初始化
 - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
 - 输入捕获
 - 输出比较

通用定时器特性：

- 16 位递增、递减、递增/递减自动重载计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频
- 多达 4 个独立通道，可用于：
 - 输入捕获
 - 输出比较
 - PWM 生成
 - 单脉冲模式输出
- 使用外部信号控制定时器，可实现多个定时器互联
- 发生如下事件时生成中断/DMA 请求：
 - 更新：计数器上溢/下溢、计数器初始化
 - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
 - 输入捕获
 - 输出比较

2.21 CAN

控制器局域网（CAN）根据 ISO11898-1: 2015 和 Bosch CAN FD 规范进行通信。连接到物理层需要额外的收发器硬件。

所有有关处理消息的函数都由接收处理程序和发送处理程序实现。接收处理程序管理消息接受过滤，将接收到的消息从 CAN 核心发送到消息 RAM，以及提供接收消息状态信息。发送处理程序负责将发送消息从消息 RAM 发送到 CAN 核心，并提供发送状态信息。

接受过滤通过最多 32 个过滤器元素的组合来实现，其中每个元素可以配置为范围、位掩码或专用 ID 过滤器。

- 设计符合 ISO11898-1/2015 规范；
- 支持 CAN 和 CAN FD 帧；
- 支持 ISO11899: 2015 规范中指定的 CAN FD 帧格式；
- 支持 64 字节 CAN FD 帧；
- 支持可变数据速率高达 8Mb/s；
- 支持正常数据速率 1Mb/s；
- 支持高达三个数据位的发射器延迟补偿；
- 支持可配置的发送和缓冲区；
- 支持两个深度为 64 帧消息的缓冲与 32 个 ID 过滤掩码；
- 支持低 ID 的消息优先发送；
- 支持待发送消息消除；
- 支持快速数据速率时的单独错误记录。

2.22 USART

USART 能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 可实现多种波特率。

USART 不仅支持同步单向通信和半双工单线通信，以及 LIN（局域互连网络）和调制解调器操作（CTS/RTS）。通过配置多个缓冲区使用 DMA（直接存储器访问）可实现高速数据通信。

- 支持全双工/半双工异步通信
- 支持 NRZ 标准格式

- 可配置 16 倍过采样和 8 倍过采样
- 自动波特率检测
- 两个可用于发送和接收数据的内部 FIFO，可通过软件使能/禁止
- 数据字长度可编程，可以实现 6 位、7 位、8 位和 9 位数据字
- 可配置停止位，支持 1 个和 2 个停止位
- 用于同步通信的时钟输入/输出
- 支持 SPI 模式
- 支持 DMA 连续数据传输
- 支持发送器和接收器单独使能
- 支持发送和接收的单独信号极性控制
- 支持调制解调 RS-485 发送器的硬件流控制
- 支持奇偶校验控制
- 支持 LIN 主从模式，可配置中断符

2.23 SPI

SPI 全称是 **serial peripheral interface**，串行外围设备接口。MCU 提供了 4 个 SPI 接口，支持全双工的同步串行通信。该接口可配置为主机或从机模式，配置为主机模式时，它可为外部从器件提供通信时钟（SCK），SPI2 和 SPI3 接口支持 4 个从机，拥有 4 个从机片选，SPI0 和 SPI1 支持 1 个从机。

- 支持 Motorola 模式串行外设接口（SPI）
- 4 线全双工同步传输
- 4~32 位传输帧格式选择
- SPI 时钟速率可配置
- 主模式或从模式操作
- 主模式时最高频率 PCLK/2
- 从模式时最高频率 PCLK/8
- 数据发送顺序 MSB 在前
- 可编程的时钟极性和相位
- 可触发中断的专用发送和接收标志

- 支持 DMA 功能

2.24 IIC

IIC（Inter-Integrated Circuit）总线是一种简单、双向二线制同步串行总线，通过时钟线（SCL）与数据线（SDA）进行数据传输。SCL 是由主机驱动产生的时钟信号，SDA 是双向数据信号，既可由主设备（主机）产生，也可由从设备（从机）产生。

- 支持主从一体模式
- 支持 7bit 范围地址和 10bit 范围地址
- 支持从机监测功能
- 支持多主机仲裁
- 支持可编程毛刺过滤
- 支持 SDA/SCL 低电平超时中断

2.25 MAC

MAC 接口实现了由 IEEE 802.3 定义的媒体访问控制（MAC）通过以太网连接的碰撞检测（CSMA/CD）算法。与外部主机的通信是通过一组控制和状态寄存器和针对外部共享 RAM 的直接内存访问（DMA）控制器来实现的。MAC 接口通过 DMA 方式与主机实现数据发送。它自动获取发送缓冲区数据和存储接收缓冲区数据到外部 RAM。它通过对接收和发送描述符链表的管理，实现多种内存分配方案。MAC 接口内部 RAM 被用作可配置的 FIFO 存储器块，并且有单独的用于发送和接收进程的存储器块。

- 支持外部 PHY 接口实现 10/100Mbit/s 数据传输速率；
- 通过符合 IEEE 802.3 的 MII 接口与外部快速以太网 PHY 进行通信；
- 支持全双工和半双工操作；
- 报头和帧起始数据（SFD）在发送路径中插入、在接收路径中删除；
- 可逐帧控制 CRC 和 PAD 自动生成；
- 接收帧时可自动去除 PAD/CRC；
- 支持多种灵活的地址过滤模式；

- 两组缓存 FIFO（接收 FIFO 和发送 FIFO）；
- DMA 在内存和 MAC 的缓存区进行数据搬移。

2.26 GPIO

片上设备用 GPIO 来实现逻辑输入/输出功能。每个 GPIO 端口有相关的控制和配置寄存器以满足特定应用的需求。外设 GPIO 引脚上都有其独立的中断控制。GPIO 端口和其他的备用功能（Afs）共用引脚，在特定的封装下获得最大的灵活性。GPIO 引脚通过配置相关的寄存器可以用作备用功能引脚，备用功能输入/输出都可。每个 GPIO 引脚可以由软件配置为输出（推挽或开漏）、输入、外设的备用功能或者模拟模式。每个 GPIO 引脚都可以配置为上拉、下拉或无上拉/下拉。

- 输入/输出方向控制
- 触发器输入功能使能控制
- 每个引脚都具有弱上拉/下拉功能
- 推挽/开漏输出使能控制
- 置位/复位输出使能
- 可编程沿触发/电平触发的中断
- 模拟输入/输出配置
- 备用功能输入/输出配置

2.27 DEBUG

调试器基于 RISC-V Foundation 的 Debug Spec 0.13.2 制作，外部接口可以访问内部的寄存器与存储器，控制程序的运行/停止/复位等操作。

- JTAG 调试接口
- 16 个硬件断点
- 调试安全性

附录

2.28 地址映射

地址映射表

空间	起始地址	结束地址	空间大小	说明
AXI Bus 0	0x0000_0000	0x1FFF_FFFF	512 MiB	指令总线
	0x0000_0000	0x0000_0FFF	4 KiB	BOOT-ROM
	0x0100_0000	0x011F_FFFF	2 MiB	P-FLASH Main
	0x0120_0000	0x0120_3FFF	16 KiB	P-FLASH Info
	0x0200_0000	0x0207_FFFF	512 KiB	D-FLASH Main
	0x0208_0000	0x0208_0FFF	4 KiB	D-FLASH Info
	0x1000_0000	0x1FFF_FFFF	256 MiB	QSPI-FLASH
AXI Bus 1	0x2000_0000	0x2FFF_2FFF	256 MiB	数据总线
	0x2000_0000	0x2001_FFFF	128KiB	SRAM0
	0x2002_0000	0x2003_FFFF	128KiB	SRAM1
	0x2004_0000	0x2005_FFFF	128KiB	SRAM2
	0x2006_0000	0x2007_FFFF	128KiB	SRAM3
AXi Bus 2	0x3000_0000	0x7FFF_FFFF	1.25 GiB	配置/外设总线
AxiLite4 Bus0	0x3000_0000	0x3FFF_FFFF	256MiB	设置总线
	0x3000_0000	0x3000_03FF	1 KiB	MPU0-IBUS
	0x3000_0400	0x3000_07FF	1 KiB	MPU1-DBUS
	0x3000_0800	0x3000_0BFF	1 KiB	MPU2-PBUS
	0x3000_0C00	0x3000_0FFF	1 KiB	MPU3-DMA0
	0x3000_1000	0x3000_13FF	1 KiB	MPU4-DMA1
	0x3000_2000	0x3000_23FF	1 KiB	FCU
	0x3000_3000	0x3000_33FF	1 KiB	CMU0
	0x3000_3400	0x3000_37FF	1 KiB	CMU1

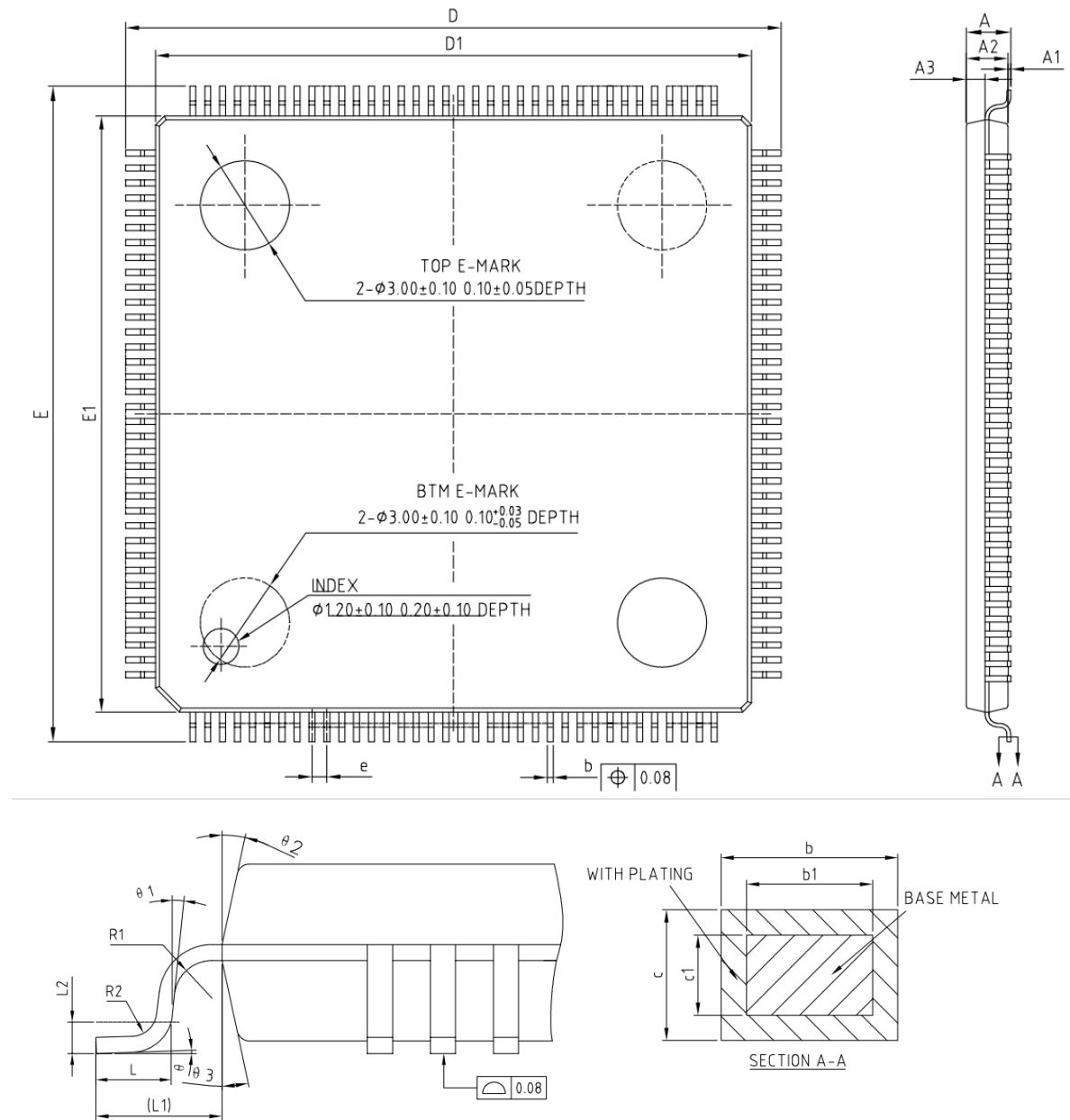
空间	起始地址	结束地址	空间大小	说明
	0x3000_3800	0x3000_3BFF	1 KiB	CMU2
	0x3000_3C00	0x3000_3FFF	1 KiB	CMU3
	0x3000_4000	0x3000_43FF	1 KiB	DMA0-CFG
	0x3000_4400	0x3000_47FF	1 KiB	DMA1-CFG
	0x3000_6000	0x3000_63FF	1 KiB	SMU
	0x3001_0000	0x3001_FFFF	64 KiB	CLINT
	0x3002_0000	0x3002_FFFF	64 KiB	PLIC
APB Bus S	0x3010_0000	0x301F_FFFF	1 MiB	备份总线
	0x3010_0000	0x3010_07FF	2 KiB	BACKUP-RAM
	0x3010_1000	0x3010_13FF	1 KiB	RTC
	0x3010_2000	0x3010_23FF	1 KiB	GPIOH
	0x3010_3000	0x3010_33FF	1 KiB	PMU
	0x3010_4000	0x3010_43FF	1 KiB	IWDG
AXI Bus 3	0x4000_0000	0x7FFF_FFFF	1 GiB	外设总线
AxiLite4 Bus1	0x4100_0000	0x41FF_FFFF	16 MiB	外设 Lite 总线 1
	0x4100_0000	0x4100_03FF	1 KiB	TIM0
	0x4101_0000	0x4101_FFFF	16 KiB	CANFD0
	0x4102_0000	0x4102_FFFF	16 KiB	CANFD1
	0x4100_1000	0x4100_13FF	1 KiB	ADC0
	0x4100_2000	0x4100_23FF	1 KiB	ADC1
	0x4100_3000	0x4100_33FF	1 KiB	DAC/CMP0
	0x4100_4000	0x4100_43FF	1 KiB	GPIOA
	0x4100_5000	0x4100_53FF	1 KiB	GPIOB
	0x4100_6000	0x4100_63FF	1 KiB	GPIOC
	0x4110_0000	0x414F_FFFF	1 MiB	DSE
AxiLite4 Bus2	0x4200_0000	0x42FF_FFFF	16 MiB	外设 Lite 总线 2

空间	起始地址	结束地址	空间大小	说明
	0x4200_0000	0x4200_03FF	1 KiB	TIM1
	0x4201_0000	0x4201_FFFF	16 KiB	CANFD2
	0x4202_0000	0x4202_FFFF	16 KiB	CANFD3
	0x4200_1000	0x4200_13FF	1 KiB	ADC2
	0x4200_2000	0x4200_23FF	1 KiB	DAC/CMP1
	0x4200_3000	0x4200_33FF	1 KiB	GPIOD
	0x4200_4000	0x4200_43FF	1 KiB	GPIOE
	0x4200_5000	0x4200_53FF	1 KiB	GPIOF
	0x4200_6000	0x4200_63FF	1 KiB	GPIOG
	0x4210_0000	0x4210_03FF	1 KiB	FLASH CTRL
	0x4210_0800	0x4210_0BFF	1 KiB	QSPI CTRL
APBBus0	0x5000_0000	0x50FF_FFFF	16 MiB	外设 APB 总线 0
	0x5000_0000	0x5000_03FF	1 KiB	MAC
	0x5000_1000	0x5000_13FF	1 KiB	SPI0
	0x5000_2000	0x5000_23FF	1 KiB	SPI1
	0x5000_3000	0x5000_33FF	1 KiB	SPI2
	0x5000_4000	0x5000_43FF	1 KiB	IIC/IIS0
	0x5000_5000	0x5000_53FF	1 KiB	IIC/IIS1
	0x5000_6000	0x5000_63FF	1 KiB	USART0
	0x5000_7000	0x5000_73FF	1 KiB	USART1
	0x5000_8000	0x5000_83FF	1 KiB	USART2
	0x5000_9000	0x5000_93FF	1 KiB	USART3
	0x5000_A000	0x5000_A3FF	1 KiB	TIM2
	0x5000_B000	0x5000_B3FF	1 KiB	TIM3
	0x5000_C000	0x5000_C3FF	1 KiB	TIM4
	0x5000_D000	0x5000_D3FF	1 KiB	WWDG0

空间	起始地址	结束地址	空间大小	说明
APBBus1	0x5100_0000	0x51FF_FFFF	16 MiB	外设 APB 总线 1
	0x5100_0000	0x5100_03FF	1 KiB	SPI3
	0x5100_1000	0x5100_13FF	1 KiB	SPI4
	0x5100_2000	0x5100_23FF	1 KiB	SPI5
	0x5100_3000	0x5100_33FF	1 KiB	IIC/IIS2
	0x5100_4000	0x5100_43FF	1 KiB	IIC/IIS3
	0x5100_5000	0x5100_53FF	1 KiB	USART4
	0x5100_6000	0x5100_63FF	1 KiB	USART5
	0x5100_7000	0x5100_73FF	1 KiB	USART6
	0x5100_8000	0x5100_83FF	1 KiB	USART7
	0x5100_9000	0x5100_93FF	1 KiB	TIM5
	0x5100_A000	0x5100_A3FF	1 KiB	TIM6
	0x5100_B000	0x5100_B3FF	1 KiB	TIM7
	0x5100_C000	0x5100_C3FF	1 KiB	WWDG1
	0x5100_D000	0x5100_D3FF	1 KiB	CRC

3 封装

封装类型是 LQFP144，封装尺寸如下：



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.17	—	0.27
b1	0.17	0.20	0.23
c	0.127	—	0.18
c1	0.119	0.127	0.135
D	21.80	22.00	22.20
D1	19.90	20.00	20.10
E	21.80	22.00	22.20
E1	19.90	20.00	20.10
e	0.40	0.50	0.60
L	0.45	0.60	0.75
L1	1.00REF		
L2	0.25BSC		
R1	0.08	—	—
R2	0.08	—	—
θ	0°	—	7°
θ_1	0°	—	—
θ_2	11°	12°	13°
θ_3	11°	12°	13°