i560

安全主控芯片
技术手册

修改记录

版本号	描述	日期
v0.1	草稿版	2020/03/11

Table of Content

1 概述

1.1 产品简介

i560 是由方寸微电子自主研发的新一代 SoC 存储安全芯片，具有功能丰富、性能强劲、功耗低、安全性高等特点，可广泛适用于安全 U 盘、SD 卡、USB 接口芯片等众多存储安全领域产品。

该芯片集成 32 位国产高性能 RISC CPU，可支持 USB3.0、SD Host、SD Device、eMMC 等多种高速接口，并集成多种国密算法（如 SM2、SM3、SM4），可满足国家信息安全领域需求；同时该芯片也支持国际标准 AES 加密算法及 ECC 算法，可应用于全球通用安全市场。

该芯片提供完整的 SDK 供客户进行定制化开发，尤其针对典型应用场景提供了源码级方案支撑，可帮助客户缩短产品开发周期、降低整体开发成本，提升产品市场竞争力。

1.2 应用产品

图 1.1 产品应用方案

1.3 芯片架构

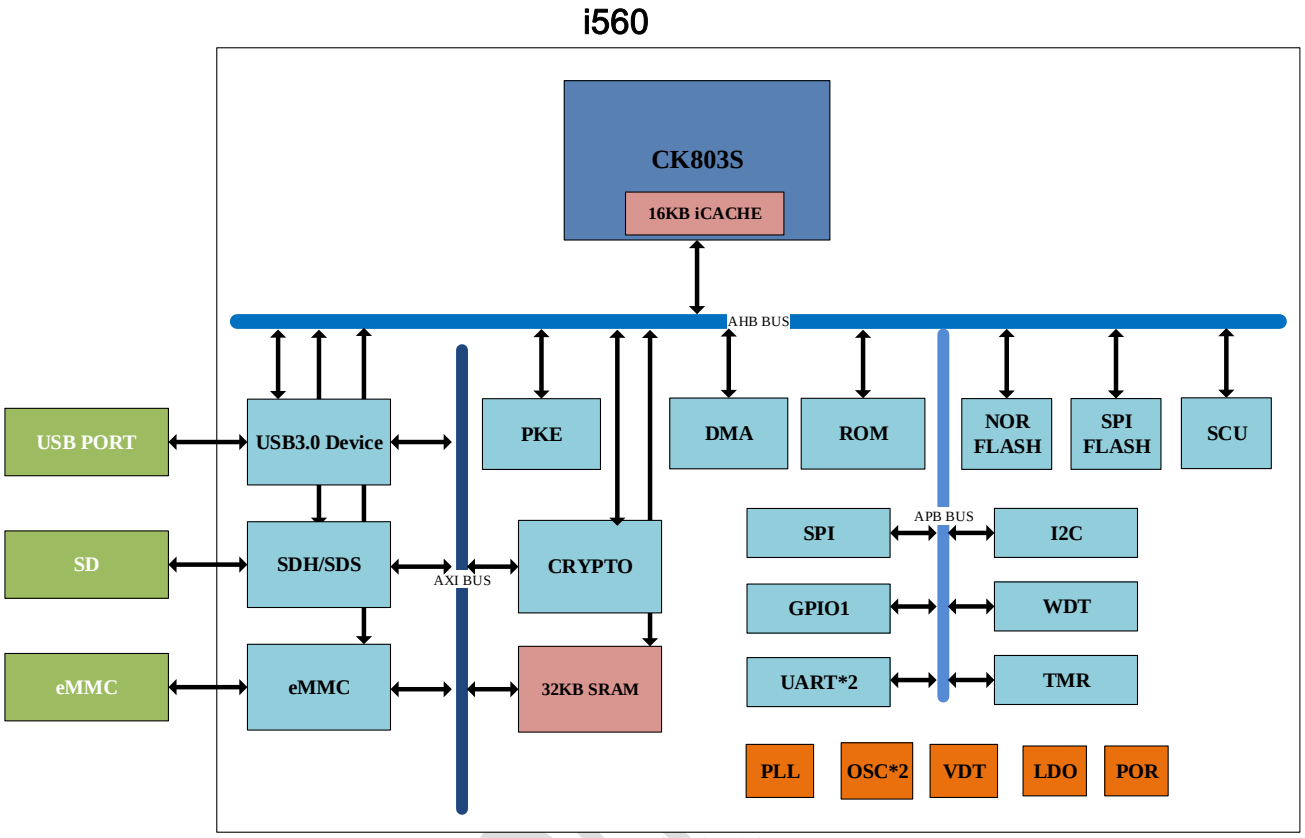


图 1.2 芯片系统架构框图

1.4 芯片特性

1.4.1 CPU 资源

- 集成 32 位国产 CPU CK803s
- 最高工作频率 150Mhz
- 内置 16KB I Cache

1.4.2 USB3.0 Device 接口

- 支持一路 USB3.0 Device 接口速率 5Gbps，向下兼容 USB2.0/USB1.1
- 支持控制/批量/中断/等时传输类型
- 符合 Universal Serial Bus（USB） revision 3.0 标准协议

1.4.3 eMMC 接口

- 支持 1 路 eMMC 接口
- 支持 eMMC5.1 协议标准
- 最高接口速率 HS400，向下兼容
- 支持 3.3V/1.8V IO 电压

1.4.4 安全引擎

- 支持 SM4、AES256 数据加密，加密性能 600MB/s@150Mhz
- 支持 ECB、CBC、OFB、CFB、CTR、XTS 6 种加密模式
- RSA（可选 CRT）：512~4096 比特
- ECC（素数域）：192、224、256、384 和 521 比特
- 支持大数模加、模减、模乘运算协处理
- SM2 密钥对生成速度 250 对/s
- 支持 SM2 签名验签，性能≥600/300 次/s@100Mhz
- RSA1024 密钥对生成时间<0.2s
- 支持 RSA1024 签名验签，性能≥600/6000 次/s@100Mhz
- RSA2048 密钥对生成时间<2s
- 支持 RSA2048 签名验签，性能≥100/2000 次/s@100Mhz
- 支持 SM3/SHA1/SHA224/SHA256 算法
- 支持一路 TRNG 发生器，生成速率≥30Mbps@50Mhz

*以上为硬件引擎性能，非最终产品性能

1.4.5 存储资源

- 32KB ROM
- 32KB SRAM
- 512KB 片内 flash

1.4.6 其他资源

- 内置硬件 DMA
- 内置 POR（Power on reset）电路
- 内置 8 个定时器
- 内置中断控制器
- 内置 1 个看门狗
- 支持 1 路 SPI Flash 控制器接口

- 支持 1 路 SPI 主/从接口
- 支持 1 路 I2C 主/从接口
- 支持 2 路 UART 接口
- 支持 16 位 GPIO 接口
- 支持在线调试

1.4.7 安全特性

- 支持电压检测
- 支持物理探测防护
- 每颗芯片具备全球唯一 ID

1.4.8 物理规格

- Core 电压为 1.0V
- IO 电压为 3.3V/1.8V
- 支持 QFN64/TF 卡封装
- 工作温度 0~70℃，-40~85℃

1.5 地址映射

表 1.1 地址映射表

基地址	大小	模块名称	说明
0x0000_0000	1MB	ROM	
0x0100_0000	16MB	Norflash XIP 端口	
0x1000_0000	1MB	SDH 寄存器端口	
0x1010_0000	1MB	CRYPTO 寄存器端口	
0x1020_0000	1MB	DMA 寄存器端口	
0x1030_0000	1MB	Norflash Controller 寄存器端口	
0x1040_0000	1MB	SPI Flash Controller 寄存器端口	
0x1050_0000	1MB	SCU 寄存器端口	
0x1060_0000	1MB	eMMC 寄存器端口	
0x1070_0000	1MB	USB Device 寄存器端口	
0x1080_0000	1MB	USBD_H2X 寄存器端口	
0x1090_0000	1MB	TRNG0 寄存器端口	
0x10A0_0000	1MB	PKE 寄存器端口	
0x10B0_0000	1MB	SD_H2X 寄存器端口	
0x10C0_0000	1MB	AHBC 寄存器端口	
0x10D0_0000	1MB	PKE SRAM	
0x10E0_0000	1MB	32KB SRAM	32KB SRAM 在 AHB 总线地址
0x10F0_0000	1MB	SD Device 寄存器端口	
0x1200_0000	32MB	AHB2APB 端口	
0x1200_0000	1MB	AXIC 寄存器端口	
0x1210_0000	1MB	SPI 寄存器端口	
0x1220_0000	1MB	UART0 寄存器端口	
0x1230_0000	1MB	UART1 寄存器端口	
0x1240_0000	1MB	TIMER 寄存器端口	
0x1250_0000	1MB	WDT 寄存器端口	
0x1260_0000	1MB	TRNG1 寄存器端口	
0x1270_0000	1MB	GPIO 寄存器端口	
0x1280_0000	1MB	H2X 寄存器端口	
0x1290_0000	1MB	IIC 寄存器端口	
0x12A0_0000	1MB	IP_TRMING_EFUSE	
0x12B0_0000	1MB	CHIP_ID_EFUSE	
0x2000_0000	64MB	AXI_BUS_SLAVE	
0x2000_0000	1MB	32K SRAM 数据端口	
0x2010_0000	1MB	eMMC 数据端口	
0x2180_0000	8MB	CRYPTO_S 数据端口	

1.6 中断源

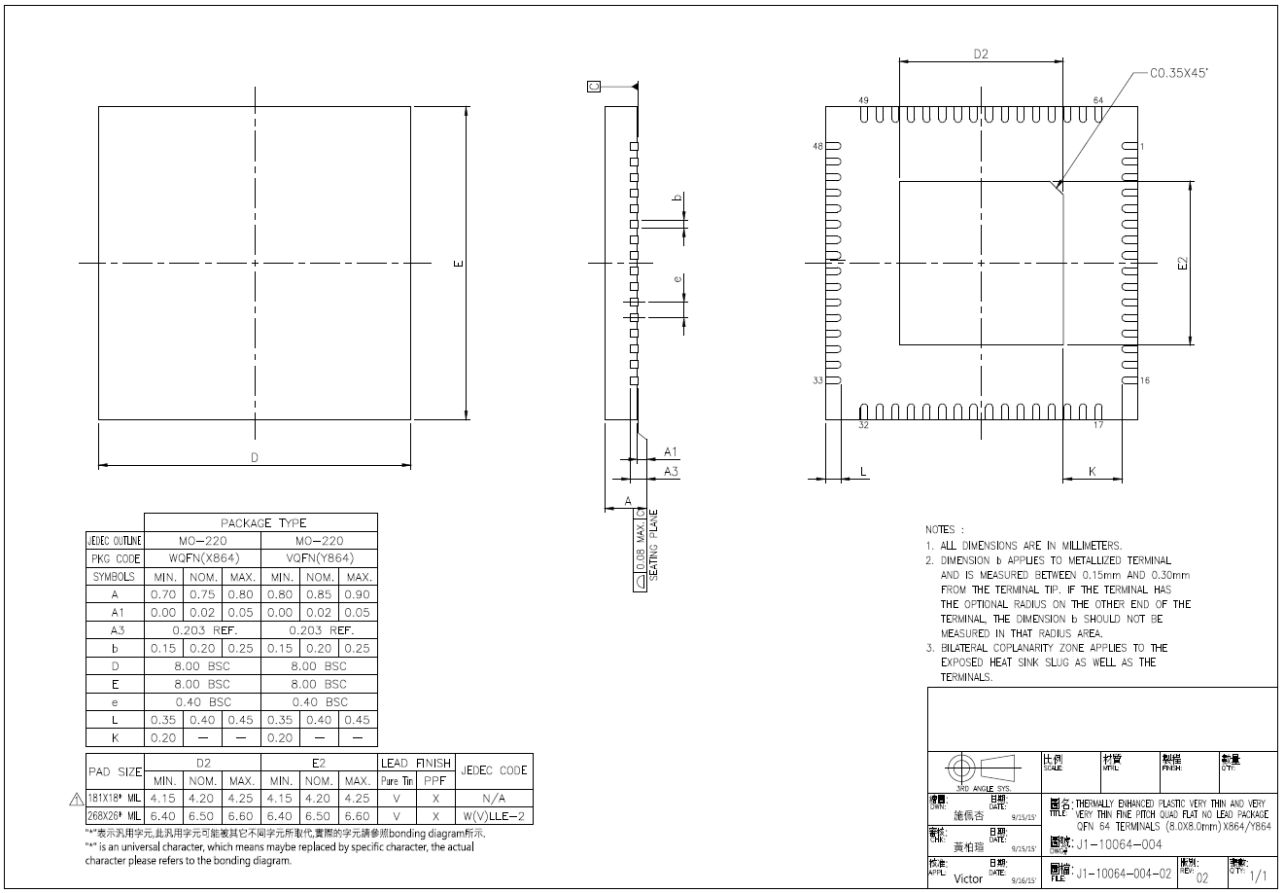
CK803S 的中断源映射如下：

表 1.2 CK803S 中断源

No.	中断源	说明
20	Core_Timer	
19	VDT	
18	TRNG0	
17	PKE	
16	AXIC	
15	AHBC	
14	GPIO	
13	WDT	
12	IIC	
11	UART_1	
10	UART_0	
9	SPI Flash Ctrl	
8	Norflash Ctrl	
7	SPI	
6	TIMER	
5	DMA	
4	eMMC	
3	CRYPTO	
2	SD	
1	USBD1	
0	USBD0	

2 硬件特性

2.1 芯片封装



	PACKAGE TYPE					
JEDEC OUTLINE	MO-220			MO-220		
PKG CODE	WQFN(X864)			VQFN(Y864)		
SYMBOLS	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.75	0.80	0.80	0.85	0.90
A1	0.00	0.02	0.05	0.00	0.02	0.05
A3	0.203 REF.			0.203 REF.		
b	0.15	0.20	0.25	0.15	0.20	0.25
D	8.00 BSC			8.00 BSC		
E	8.00 BSC			8.00 BSC		
e	0.40 BSC			0.40 BSC		
L	0.35	0.40	0.45	0.35	0.40	0.45
K	0.20	—	—	0.20	—	—

PAD SIZE	D2			E2			LEAD FINISH		JEDEC CODE
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	Pure Tin	PPF	
181X18* MIL	4.15	4.20	4.25	4.15	4.20	4.25	V	X	N/A
268X26* MIL	6.40	6.50	6.60	6.40	6.50	6.60	V	X	W(V)LLE-2

"*"表示泛用字元,此泛用字元可能被其它不同字元所取代,實際的字元請參照bonding diagram所示.
"*" is an universal character, which means maybe replaced by specific character, the actual character please refers to the bonding diagram.

图 2.1 芯片封装尺寸图

*注：D2=E2=6.5mm

2.2 管脚分布

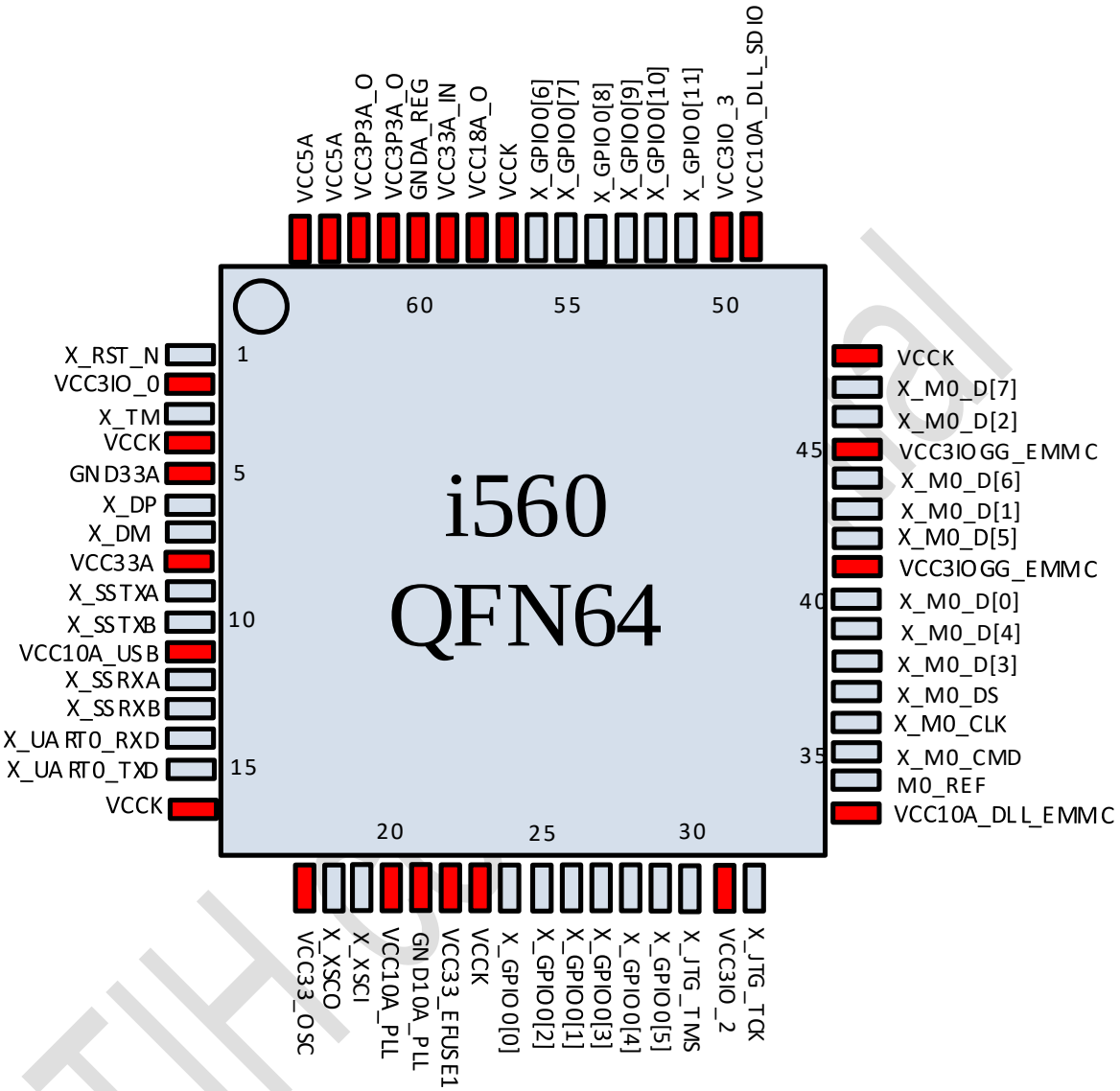


图 2.2 QFN64 封装图

2.3 管脚描述

表 2.1 管脚描述

位置	名称	I/O	功能描述
left			
1	X_RST_N	I	系统复位引脚，低电平有效
2	VCC3IO_0		3.3V 数字电源
3	X_TM	I	测试模式使能信号： 0：正常功能模式 1：测试模式 默认下拉
4	VCCK		1.0V 数字电源
5	GND33A		3.3v 模拟地
6	DP	IO	USB2.0 高速差分输入输出
7	DM	IO	USB2.0 高速差分输入输出
8	VCC33A		USB 3.3V 模拟电源
9	X_SSTXA	O	USB 3.0 接口差分输出 A
10	X_SSTXB	O	USB 3.0 接口差分输出 B
11	VCC10A_USB		USB 3.0 1.0V 模拟电源
12	X_SSRXA	I	USB 3.0 接口差分输入 A
13	X_SSRXB	I	USB 3.0 接口差分输入 B
14	UART0_TXD	O	UART0 TXD 信号
15	UART0_RXD	I	UART0 RXD 信号
16	VCCK		1.0V 数字电源
down			
17	VCC33_OSC		OSC 3.3V 数字电源
18	X_XSCO	O	系统晶振输出，30Mhz
19	X_XSCI	I	系统输入时钟，30MHz
20	VCC10A_PLL		PLL 1.0V 模拟电源
21	GND10A_PLL		PLL 模拟地
22	VCC33_EFUSE1	I	EFUSE1 3.3v 数字电源
23	VCCK		1.0V 数字电源
24	X_GPIO[0]	IO	GPIO 通用输入输出端口 0
25	X_GPIO[2]	IO	GPIO 通用输入输出端口 2
26	X_GPIO[1]	IO	GPIO 通用输入输出端口 1
27	X_GPIO[3]	IO	GPIO 通用输入输出端口 3
28	X_GPIO[4]	IO	GPIO 通用输入输出端口 4
29	X_GPIO[5]	IO	GPIO 通用输入输出端口 5
30	X_JTG_TMS	IO	JTAG 测试信号输入输出
31	VCC3IO_2		3.3V 数字电源
32	X_JTG_TCK	I	JTAG 测试时钟输入
31	VCCK		1.0V 数字电源
32	M0_D3	IO	eMMC DATA3 信号
right			
33	VCC10A_DLL_EMMC		eMMC DLL 1.0V 数字电源
34	M0_REF	I	eMMC 测试信号
35	M0_CMD	IO	eMMC CMD 信号
36	M0_CLK	O	eMMC CLK 信号
37	M0_DS	I	eMMC DS 信号
38	M0_D3	IO	eMMC DATA3 信号

39	M0_D4	IO	eMMC DATA4 信号
40	M0_D0	IO	eMMC DATA0 信号
41	VCC3IOGG_EMMC		eMMC 3.3V/1.8V 数字电源
42	M0_D5	IO	eMMC DATA5 信号
43	M0_D1	IO	eMMC DATA1 信号
44	M0_D6	IO	eMMC DATA6 信号
45	VCC3IOGG_EMMC		eMMC 3.3V/1.8V 数字电源
46	M0_D2	IO	eMMC DATA2 信号
47	M0_D7	IO	eMMC DATA7 信号
48	VCCK		1.0V 数字电源
up			
49	VCC10A_DLL_SDIO		SD DLL 1.0V 数字电源
50	VCC3IO_3		3.3V 数字电源
51	X_GPIO[11]	IO	GPIO 通用输入输出端口 11
52	X_GPIO[10]	IO	GPIO 通用输入输出端口 10
53	X_GPIO[9]	IO	GPIO 通用输入输出端口 9
54	X_GPIO[8]	IO	GPIO 通用输入输出端口 8
55	X_GPIO[7]	IO	GPIO 通用输入输出端口 7
56	X_GPIO[6]	IO	GPIO 通用输入输出端口 6
57	VCCK		1.0V 数字电源
58	VCC18A_O		SD 1.8V 模拟电源输出
59	VCC33A_IN		SD 3.3V 模拟电源输入
60	GNDA_REG		SD 模拟地
61	VCC3P3A_O		3.3V 模拟电源输出
62	VCC3P3A_O		3.3V 模拟电源输出
63	VCC5A		5.0V 模拟电源输入
64	VCC5A		5.0V 模拟电源输入

2.4 管脚复用

在芯片内部，GPIO、UART1、SPI，SD 等模块复用 12 根 IO 线，复用模式如下表 2.2 所示：

GPIO 可通过 SCU 寄存器配置为 0、1、2、3 四种功能模式。其中 GPIO[3:0]在任何功能模式下可通过 SCU 寄存器切换成不同的 IO 接口，GPIO[11:6]仅在功能模式 3 下可以切换成不同的 IO 接口。

表 2.2 管脚复用表

FUNC_MODE[1:0]	0	1	2	3
接口信号	功能模式 0	功能模式 1	功能模式 2	功能模式 3

GPIO[11]	SDS_CLK	SPI_CK	SDM_CLK	GPIO[11]/SPIFlash_SCK
GPIO[10]	SDS_CMD	SPI_CS	SDM_CMD	GPIO[10]/SPIFlash_CS
GPIO[9]	SDS_DAT3	SPI_TXD	SDM_DAT3	GPIO[9]/SPIFlash_TX
GPIO[8]	SDS_DAT2	SPI_RXD	SDM_DAT2	GPIO[8]/SPIFlash_RX
GPIO[7]	SDS_DAT1	PWM_OUT	SDM_DAT1	GPIO[7]/SPIFlash_WP
GPIO[6]	SDS_DAT0	GPIO[6]	SDM_DAT0	GPIO[6]/SPIFlash_HOLD
GPIO[5]	GPIO[5]	GPIO[5]	SDM_CD	GPIO[5]
GPIO[4]	GPIO[4]	GPIO[4]	SDM_WP	GPIO[4]
GPIO[3]	GPIO[3]/I2C_SDA	GPIO[3]/I2C_SDA	GPIO[3]/I2C_SDA	GPIO[3]/I2C_SDA
GPIO[2]	GPIO[2]/I2C_SCL	GPIO[2]/I2C_SCL	GPIO[2]/I2C_SCL	GPIO[2]/I2C_SCL
GPIO[1]	GPIO[1]/UART1_TXD	GPIO[1]/UART1_TXD	GPIO[1]/UART1_TXD	GPIO[1]/UART1_TXD
GPIO[0]	GPIO[0]/UART1_RXD	GPIO[0]/UART1_RXD	GPIO[0]/UART1_RXD	GPIO[0]/UART1_RXD

注意：

（1）GPIO[0]默认上拉

（2）GPIO[0]-GPIO[3]受 FUNC_IO_MODE 控制

（3）SPIFlash 信号受 FUNC_MODE 和 FUNC_IO_MODE 共同控制

2.5 上电时序

为确保芯片内部逻辑与外部器件通讯正常，上电时应先供应 VCCK 类（1.0V）引脚电压，再对 IO 类（3.3/1.8V）进行供电，最差情况也要保证 VCCK 类电压和 IO 类电压同时上电。

如下图所示：

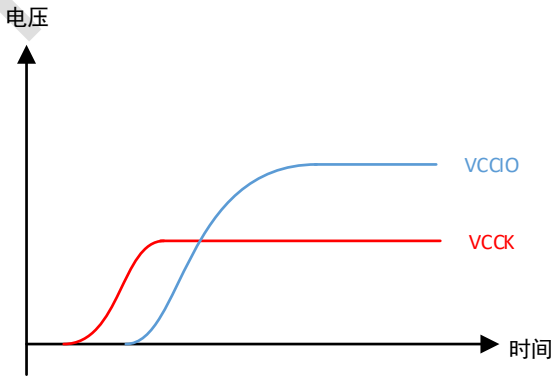


图 2.3 芯片上电时序要求图

2.6 电性能参数

表 2.3 电气特性参数

符号	描述	Min	Typ	Max	单位
VCC33IO_0	普通 IO 口电源	3.0/1.62	3.3/1.8	3.6/1.98	V
VCC33IO_2	普通 IO 口电源	3.0/1.62	3.3/1.8	3.6/1.98	V
VCC33IO_3	普通 IO 口电源	3.0/1.62	3.3/1.8	3.6/1.98	V
VCCK	Core 电源	0.9	1.0	1.1	V
VCC10A_PLL	PLL 模拟电源	0.9	1.0	1.1	V
VCC33A	USB 接口 IO 模拟电源	3.0	3.3	3.6	V
VCC10A_USB	USB 接口 core 模拟电源	0.9	1.0	1.1	V
VCC10A_DLL	eMMC DLL 内核数字电源	0.9	1.0	1.1	V
VCC33_OSC	系统晶振数字电源	3.0	3.3	3.6	V
VCC33_EFUSE1	EFUSE1 接口 IO 数字电源	3.0	3.3	3.6	V
VCC10A_DLL_EMMC	eMMC DLL 模拟电源	0.9	1.0	1.1	V
VCC33IOGG_EMMC	eMMC 接口 IO 数字电源	3.0/1.62	3.3/1.8	3.6/1.98	V
VCC10_DLL_SDIO	SD Host DLL DLL_SDIO 电源	0.9	1.0	1.1	V
VCC18A_O	LDO 输出模拟电源	1.62	1.8	1.98	V
VCC33A_IN	SD 接口模拟电源	3.0	3.3	3.6	V
VCC3P3A_O	LDO 输出模拟电源	3.0	3.3	3.6	V
VCC5A	LDO 输入模拟模拟电源	4.5	5.0	5.5	V

2.7 功耗

- 静态功耗＜0.1W
- 动态功耗＜1.0W

2.8 PCB 设计建议

请参考《i560 PCB 设计指南》

3 CPU 子系统

3.1 CK803S 处理器

3.1.1 简介

CK803S 是面向控制领域的 32 位高能效嵌入式 CPU 核，具有低成本、低功耗、高代码密度等多种特点。CK803S 采用 16/32 位混合编码指令系统，设计了精简高效的 3 级流水线。

CK803S 提供多总线接口，支持系统总线、指令总线、数据总线的灵活配置。CK803S 针对内存拷贝应用做了特殊优化，可以获得极致的内存拷贝性能。此外，CK803S 对中断响应做了特殊加速，中断响应延时仅需 13 个周期。

3.1.2 特性

- 精简指令集（RISC）处理器架构
- 32 位数据，16 位/32 位混合编码指令
- 16 个 32 位通用寄存器
- 3 级流水线
- 最高工作频率 150Mhz
- 单位性能 1.5DMIPS/MHz
- 按序发射、按序执行、按序退出
- 支持 AHB 系统总线和 AHB Databus 总线接口
- 内置 16KB 高速缓存
- 内置 8 个内存保护单元
- 内置紧耦合矢量中断控制器与计时器
- 支持 1:1 处理器与系统时钟比
- 中断响应延时仅为 13 个处理器周期
- 静态分支预测
- 支持硬件乘除法
- 支持连续内存访问
- 仅支持 little endian

3.1.3 架构

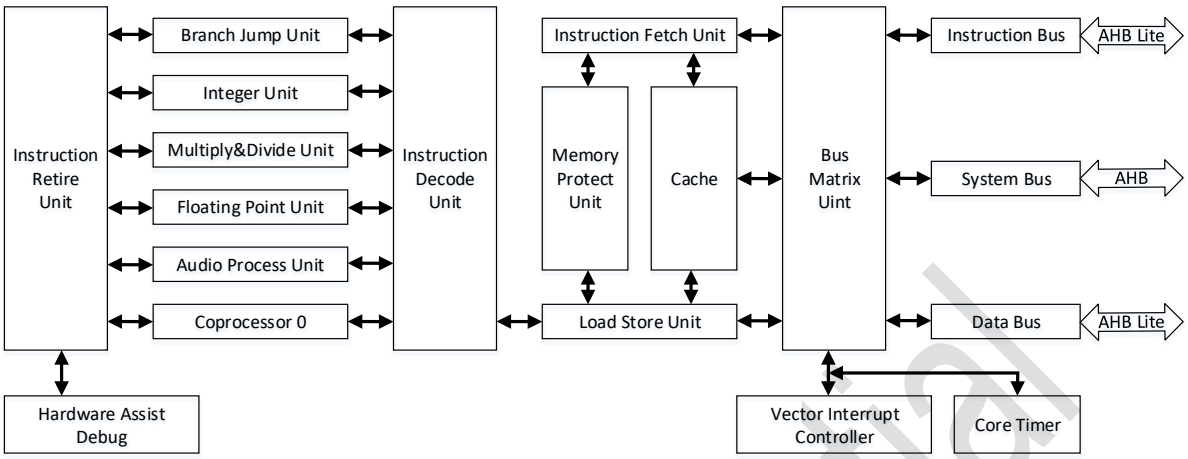


图 3.1 CK803S 系统架构图

*上图中，浮点处理单元、音频加速单元和指令总线模块本芯片中不支持。

3.1.4 矢量中断控制器

矢量中断控制器（VIC）是一个与 CK803S 紧耦合的 IP 单元，用于中断的高效处理。矢量中断控制器最大可支持 32 个中断源（IRQ[31:0]），每个中断源拥有软件可编程的中断优先级。矢量中断控制器收集来自不同中断源的中断请求，依据中断优先级对中断请求进行仲裁。最高优先级的中断将获得中断控制权并向处理器发出中断请求，当处理器响应中断请求，返回中断请求响应信号给 VIC；当处理器退出中断服务程序（ISR），返回中断退出信号给 VIC。

矢量中断控制器支持中断嵌套。当处理器正在处理一个中断请求时来了一个更高优先级的中断请求，处理器将暂停当前中断服务程序，响应更高优先级的中断请求。在更高优先级的中断请求处理结束时，CPU 返回被暂停的中断服务程序继续执行。矢量中断控制器允许高优先级的中断请求抢占低优先级的中断请求，但不允许同级别或者低优先级的中断抢占，保证了中断响应的实时性。

矢量中断控制器的系统结构图如图所示。

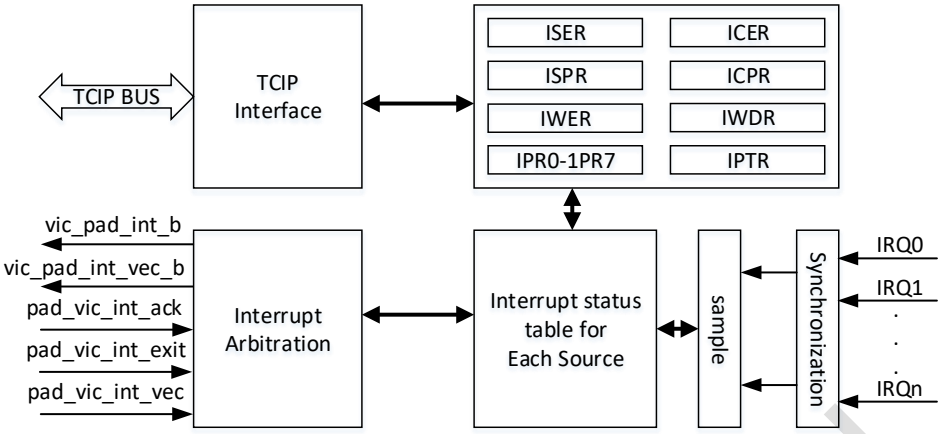


图 3.2 中断控制器结构框图

3.1.5 系统计时器

系统计时器 Core Timer 是 CK803S 内部集成的一个紧耦合模块，主要用于计时。Core Timer 提供了一个简单易用的 24 位循环递减的计数器，当 Core Timer 使能时，计数器开始工作，当计数器递减到 0 时，会向矢量中断控制器发起中断请求，申请获得处理器响应并处理 Core Timer 的事务。

Core Timer 的结构框图如图所示：

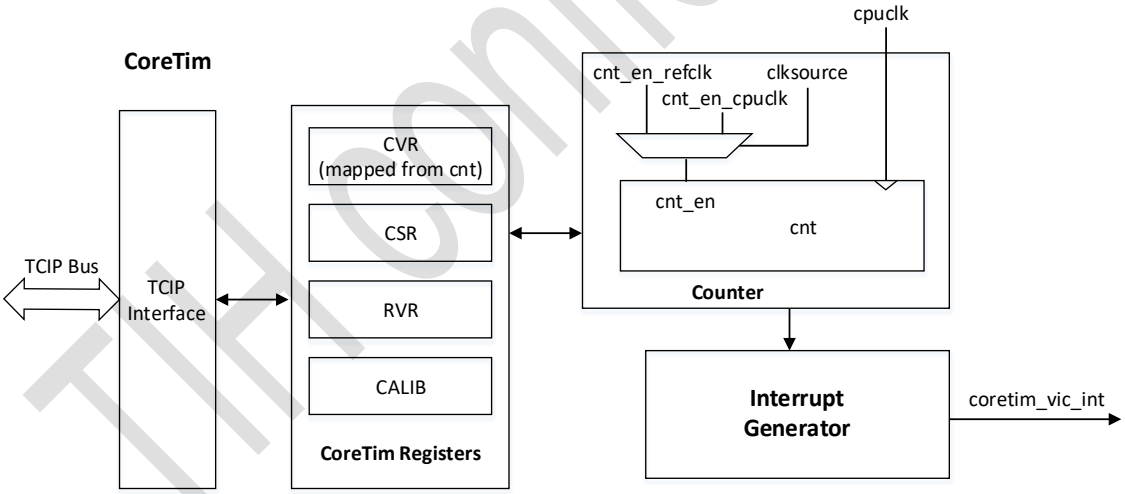


图 3.3 CoreTimer 结构框图

3.2 存储

芯片内部包含 3 块存储单元：ROM、Nor flash 和 SRAM。

内置 32KB ROM 固化了 Bootrom 程序，用于上电固件引导及固件下载，用户无法修改；

内置 512KB Nor flash，可用于存储固件代码及用户敏感信息；

Nor flash 主要参数如下：

- 页大小：512B
- 8/16/32bit 读、32bit 写
- 擦写次数：10 万次

内置 1 片 32KB SRAM 可供用户使用。可用于高速固件代码执行、临时数据存储和算法运算等，拥有独立 AHB 和 AXI 接口访问通道，可大大提升 AHB 和 AXI 接口之间数据搬运效率。

3.3 DMA

3.3.1 模块概述

DMA (Direct Memory Access)是为了降低 CPU 负担专门用来进行数据搬运的模块。在 T680 中，单纯的 DMA 模块只在 AHB 总线上集成了一个，如果 AXI 总线上需要进行数据搬运，可以通过 CRYPTO 模块中的 DMA 实现。

DMA 模块架构如下：

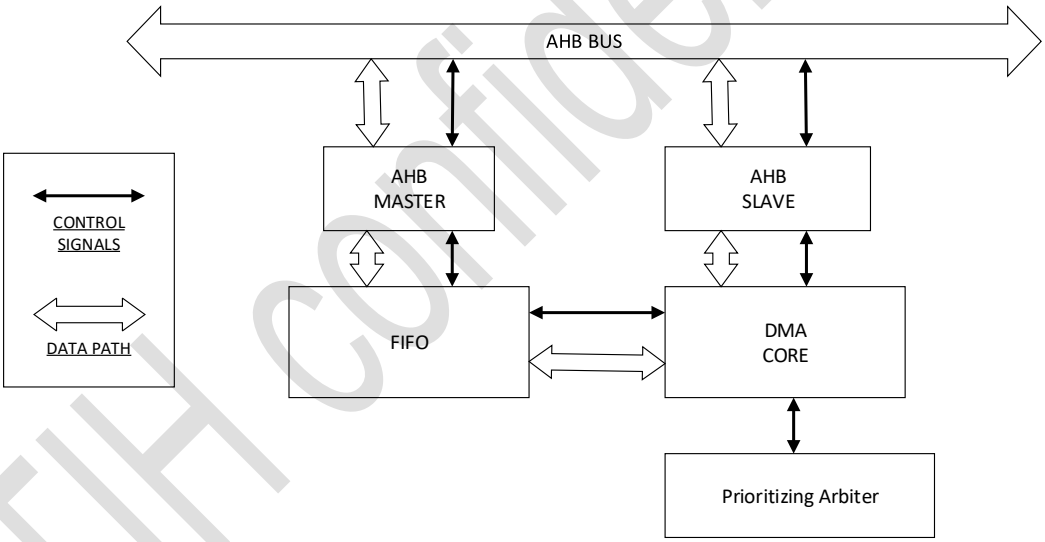


图 3.4 DMA 结构框图

3.3.2 模块特性

- 支持 8 路可配 DMA 通道
- 通道共享 16 个字节 buffer
- 支持链表模式传输
- 可在 AHB、AXI、APB bus 间进行数据搬运
- 支持 8/16/32 位数据传输
- 仅支持 little-endian 传输
- 支持 INCR 和 FIXED 地址传输模式

3.4 定时器

3.4.1 模块概述

定时器模块挂载于 APB 总线上，可提供 8 个独立的计数器，用于生成定时中断给 CPU 进行定时任务处理。同时定时器模块可生成并输出一路 PWM 信号，用于芯片外设时钟或者电机类设备的控制。

模块框图如下：

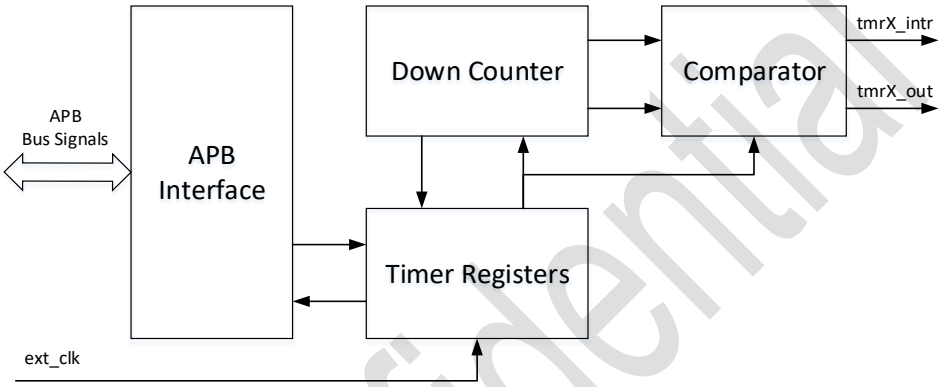


图 3.5 定时器结构框图

3.4.2 模块特性

- 支持 8 个独立的 32 位计数器
- 支持一路 PWM 输出，最高频率 20Mhz
- PWM 极性和占空比可配
- 支持自动加载模式

3.5 看门狗

3.5.1 模块概述

看门狗模块用于防止芯片固件跑飞或部分硬件造成的系统卡死情况，一旦发生上述情况，看门狗可以产生硬件复位，让整个芯片重新复位启动。

看门狗模块结构如下：

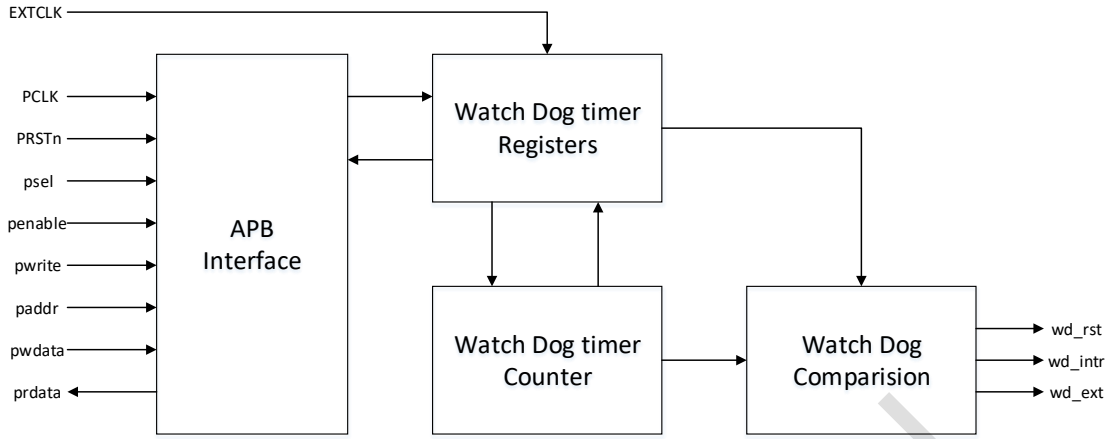


图 3.6 看门狗结构框图

3.5.2 模块特性

- 支持一路系统复位输出
- 复位输出时间可配置
- 支持一路 CPU 中断输出
- 内置 32 位递减计数器

3.6 SCU

3.6.1 模块概述

SCU 模块是系统控制单元，主要对芯片时钟、复位、功耗等芯片级配置进行控制。SCU 模块架构如下：

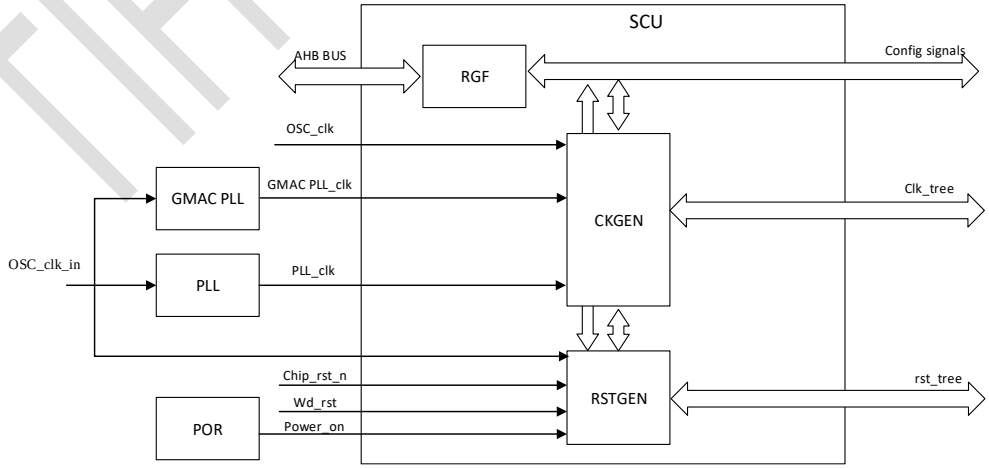


图 3.7 SCU 结构框图

3.6.2 模块特性

- 支持各模块时钟分频及门控
- 支持各模块复位控制
- 支持 PLL 输出频率可配
- 支持 PLL、OSC 时钟切换
- 支持管脚复用配置
- 内置看门狗复位状态寄存器

3.6.3 时钟树

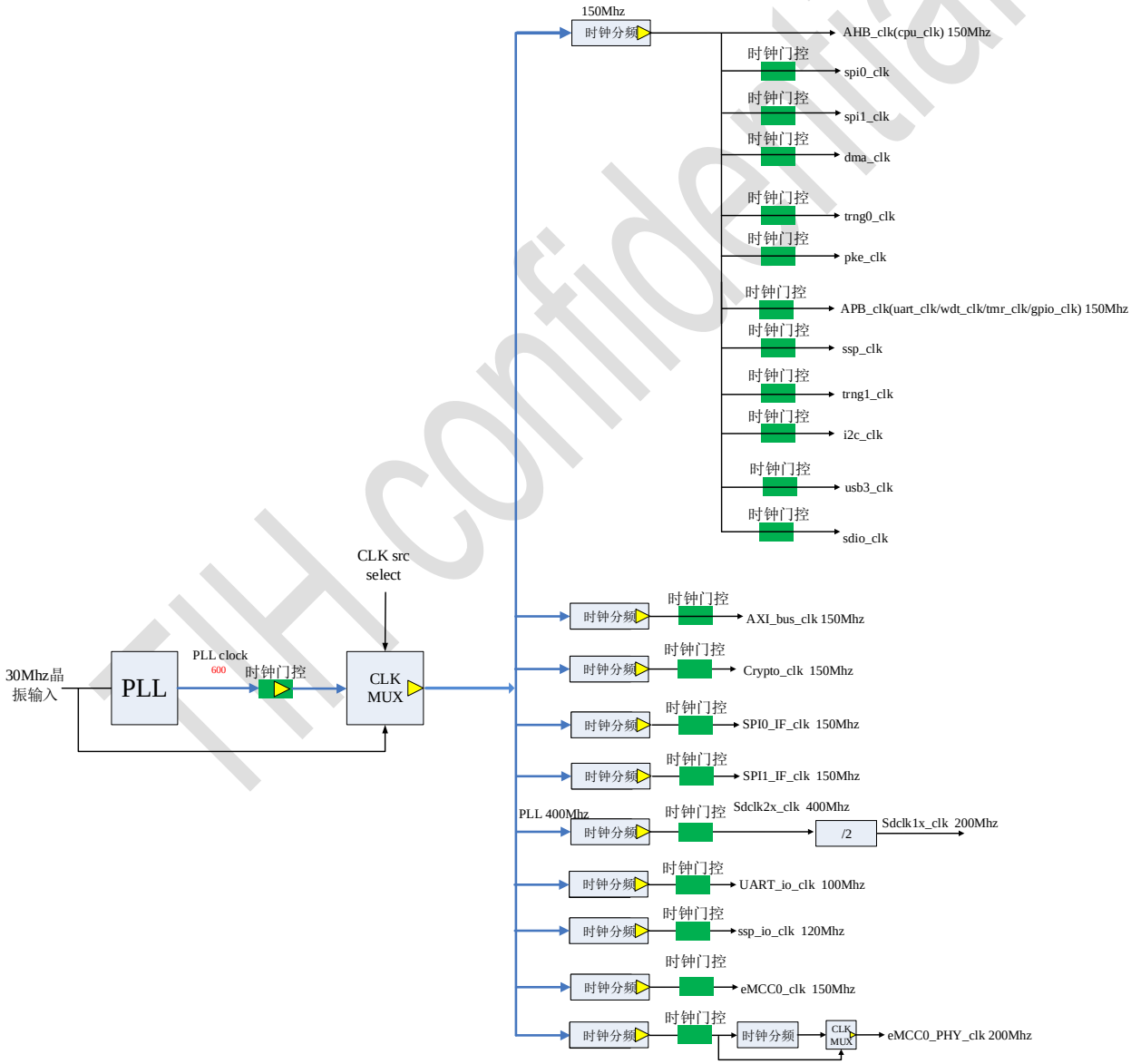


图 3.8 系统时钟树

3.6.4 复位树

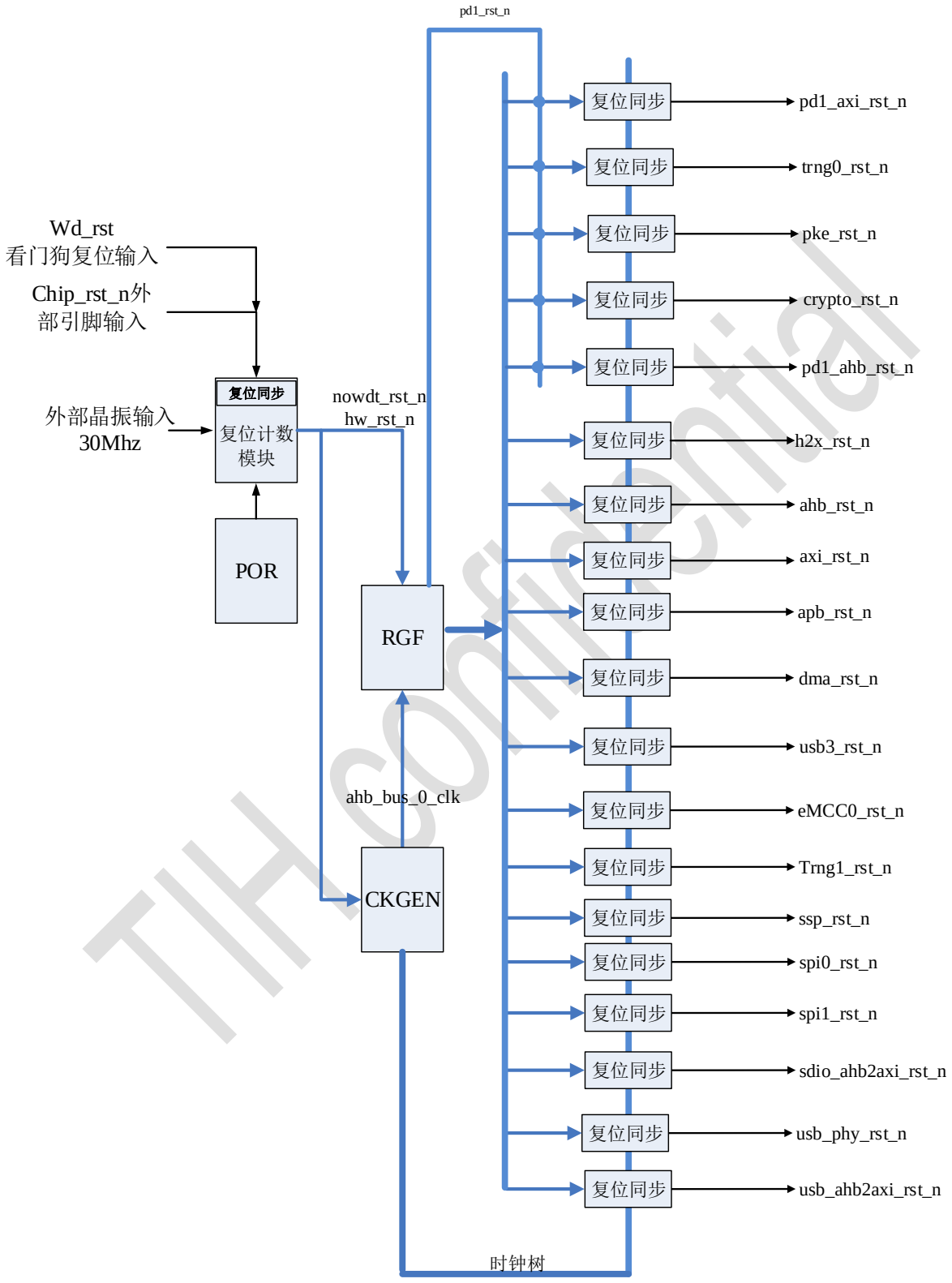


图 3.9 系统复位树

4 安全引擎

4.1 CRYPTO 引擎

4.1.1 模块概述

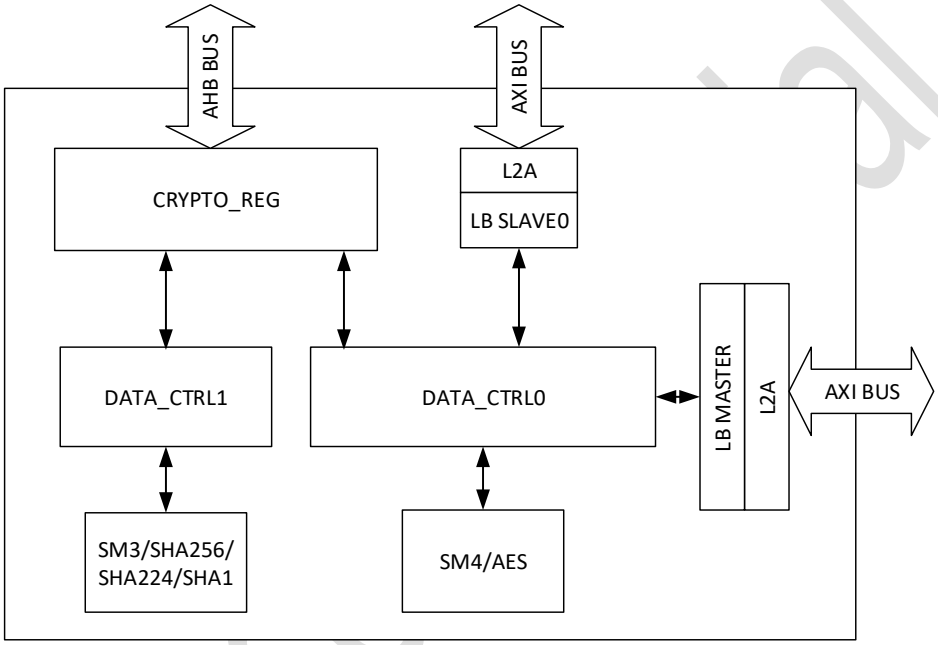


图 4.1 CRYPTO 引擎结构框图

CRYPTO 模块的作用主要是对进入模块的数据进行加解密操作。目前支持 SM4、AES 两种对称加密算法，同时支持 SM3、SHA1、SHA224、SHA256 四种 HASH 算法。两者之间相互独立，SM4、AES 数据走 AXI 总线，SM3、SHA1、SHA224、SHA256 数据走 AHB 总线。当数据从 AXI SLAVE 端口、AXI MASTER 端口或者 AHB SLAVE 端口进入 CRYPTO 模块之后，CPU 通过 AHB 总线配置 CRYPTO 模块寄存器，选择对应功能，直至加解密结束。

4.1.2 模块特性

- 支持一路 AHB SLAVE 配置端口
- 支持一路 AXI MASTER 端口
- 支持一路 AXI SLAVE 数据端口
- 支持 SM4 KEY 128 bit
- 支持 AES KEY 128/256 bit
- 支持 BYPASS 模式

- 支持数据填充和舍弃功能
- 对称加密模式支持数据输入输出端口选择
- 支持 ECB/CBC/CFB/OFB/CTR/XTS 六种操作模式（SM4/AES）
- 支持 SM4 and AES ≥600MB/s@150MHz（ECB/CTR/XTS）
- 支持 SM3/SHA1/SHA224/SHA256 ≥80MB/s@150MHz

4.1.3 工作方式

1) BYPASS 模式

数据从 CRYPTO 模块流过，不做任何处理，输入输出相同。

a) 配置控制寄存器选择 bypass 模式
b) 配置中断使能寄存器（根据需求）
c) 配置数据流向寄存器选择输入输出端口
d) 配置数据长度寄存器
e) 配置开始寄存器开启数据传输
f) 等待数据传输完成

2) FIFO 模式

将 CRYPTO 模块看成一个带有加解密功能的 FIFO，数据从 SLAVE 写端口写入，从 SLAVE 读端口将数据取走（对应 FIFO 的读写端口）。上层 MASTER 将数据写入 CRYPTO 模块之后，对数据进行加解密操作，然后再由上层 MASTER 将已加解密完数据取走。

a) 配置控制寄存器选择密码算法、加/解密、算法模式、数据大小端、密钥类型，如果选择 CTR 模式还需配置步长寄存器
b) 配置中断使能寄存器（根据需求）
c) 配置密钥以及初始值寄存器
d) 配置数据流向寄存器，选择 AXI SLAVE 端口（可读可写）
e) 配置数据长度寄存器
f) 配置开始寄存器，开始进行密钥扩展
g) 等待密钥扩展完成，配置开始寄存器开启数据传输
h) 等待数据传输完成

3) BRIDGE 模式

由一个 MASTER 端口和一个 SLAVE 端口组成。可分为两种模式，一种是正常模式，另一种是 LLI 模式。正常模式下，只需要配置一次源地址或者目的地址，LLI 模式下可以将不同的源地址或者目的地址写入命令 FIFO 中，模块会自动根据 FIFO 中的命令去执行操作(MASTER 读是配置源地址，写配置目的地址)。两种模式下的数据长度寄存器是有区别的，正常模式下按照已配置好的数据长度操作，LLI 模式下按照写入 FIFO 中的命令数据长度操作（FIFO 中命令的数据总长度等于已配置的数据长度）。

a) 配置控制寄存器选择密码算法、加/解密、算法模式、数据大小端、密钥类型以及是否使用 LLI 模式，如果选择 CTR 模式，则还需配置步长寄存器

b) 配置中断使能寄存器（根据需求）
c) 配置密钥以及初始值寄存器
d) 如果配置了 LLI 模式，则需要向 LLI 寄存器中写入命令，如果没有则跳过
e) 配置数据流向寄存器，选择一个 AXI SLAVE 端口和一个 AXI MASTER 端口
f) 根据需求配置源地址或目的地址寄存器以及 MASTER 控制寄存器
g) 配置数据长度寄存器
h) 配置开始寄存器，开始进行密钥扩展
i) 等待密钥扩展完成，配置开始寄存器开启数据传输
j) 等待数据传输完成

4) DMA 模式

由一个 MASTER 端口来完成读写操作。CRYPTO 模块会根据已配置的源地址及数据长度取数据进行加解密，然后将加解密后的数据写入对应目的地址。

a) 配置控制寄存器选择密码算法、加/解密、算法模式、数据大小端、密钥类型，如果选择 CTR 模式，则还需配置步长寄存器
b) 配置中断使能寄存器（根据需求）
c) 配置密钥以及初始值寄存器
d) 配置数据流向寄存器，选择一个 AXI MASTER 端口
e) 根据需求配置源地址或者目的地址寄存器以及 MASTER 控制寄存器
f) 配置数据长度寄存器
g) 配置开始寄存器，开始进行密钥扩展
h) 等待密钥扩展完成，配置开始寄存器开启数据传输
i) 等待数据传输完成

5) 哈希算法模式

a) 配置控制寄存器选择加密模式以及大小端
b) 配置中断使能寄存器（根据需求）
c) CPU 向数据寄存器写入数据（512bit）
d) 检测加密核的状态是否处于忙状态
e) 如果检测到加密核的状态处于空闲状态，则继续向数据寄存器写入数据（如果是最后一笔则将控制寄存器的第 5 位使能，再继续向数据寄存器写入数据），重复该步骤直到数据输入完成
f) 等待传输结束，将最终结果从数据寄存器取走

4.2 PKE 引擎

4.2.1 模块概述

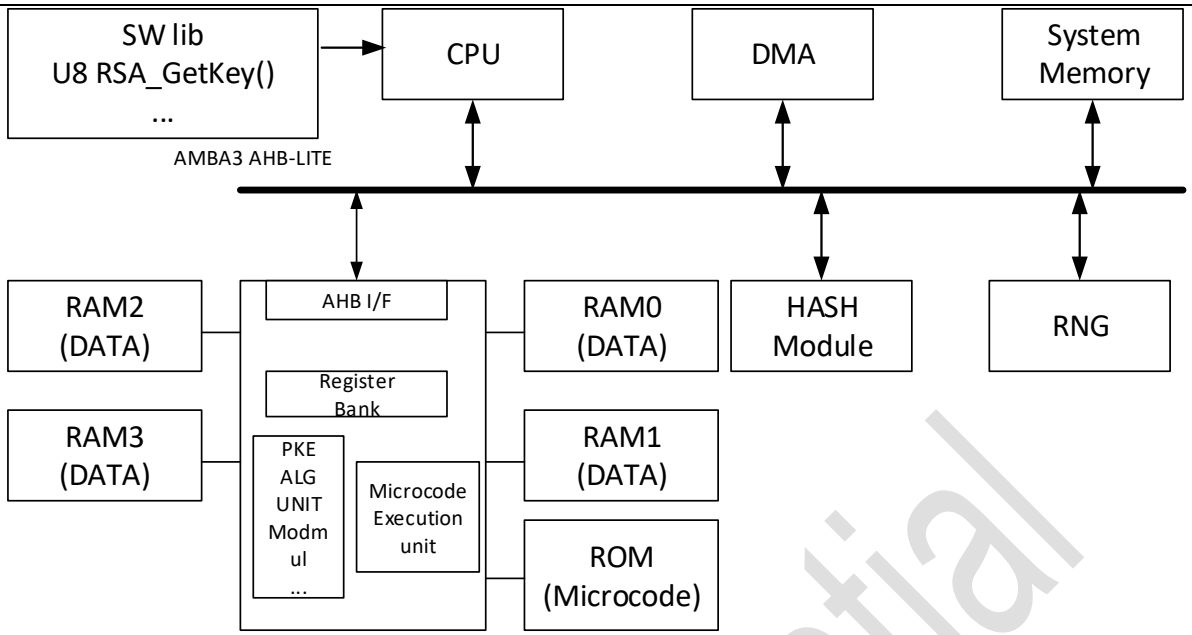


图 4.2 PKE 引擎结构框图

Public Key Engine（PKE）用来加速公钥密码运算中的大数模运算。公钥密钥的运算过程中，存在大量的大数模运算，对于普通的嵌入式 CPU 而言，完成这些大数模运算将会花费大量 CPU 指令，效率极低，因此在大多数支持公钥密码运算的芯片中都会加入公钥密码加速模块来完成公钥密码的运算。PKE 用来加速公钥密码中 RSA 和椭圆曲线（ECC）运算所涉及到的大数模运，RSA 和椭圆曲线密码是目前最为广泛使用的公钥密码。对于硬件而言，这两种加密算法都可以归结到操作数宽度分布在 32~4096 比特的模运算。其中，即使选用操作数位宽最小的 ECC-192，对于大多数 32 位的嵌入式设备而言，完成一次签名操作也会花费大量的 CPU 资源。PKE 模块将 CPU 从复杂的公钥密码运算中解放出来，CPU 只需要将输入参数配置好，PKE 会根据配置完成指定操作。目前，PKE 可以支持直接完成 RSA 中的模幂运算和 ECC 中的点乘运算。CPU 可以通过轮询或者中断方式来查询 PKE 的工作情况。

PKE 包含 AHB 接口模块（AHB I/F）、寄存器组模块、大数运算单元、微码运行单元（MEU）。另外，PKE 模块需要四块 RAM 和一块 ROM，可根据不同寄存器配置完成不同精度的运算。

4.2.2 模块特性

- RSA（可选 CRT）：512~4096 比特
- ECC（素数域）：192、224、256、384 和 521 比特
- 支持一路 AMBA 3 AHB-Lite 接口

4.2.3 工作方式

PKE 的运算通过微码（Microcode）形式完成，微码存储在程序存储单元中。因此通过向程序存储单元中灌入不同微码来实现不同要求的公钥密码运算。例如，在一个安全性要求较高

的 SoC 中，可以向 PKE 模块中的程序存储单元灌入高安全性的公钥算法指令。在一些性能优先的设计中，可以向 PKE 模块中的程序存储单元灌入性能优化的公钥算法指令，实现性能优先的目的。在程序存储单元容量较大的设计中，可以将这些运算指令都写入 ROM，由 CPU 根据不同的使用场景进行实时调用，完整的微码大小大约为 2KB。

PKE 接口被映射到 7KB 地址空间内。这一块地址映射空间内主要包含 CPU 可以访问的所有操作数，这些操作数包含了模数、幂指数、部分中间变量等。除此之外，该地址映射空间内也包含控制和状态寄存器。CPU 可以通过控制寄存器和状态寄存器来配置、监控 PKE 模块。

PKE 支持的运算中，运算数最小为 192 比特，因此，CPU 或 DMA 将数据放入数据 RAM 中会遇到字间大小端问题。在 PKE 模块中，字与字之间都是按照小端进行排列的，下一个部分会给出具体的例子。

PKE 中，最小的操作数为 256 比特（4 个双字），因为目前 ALU 的输入位宽为 256 比特，如果操作数不是字对齐的，需要将高位补零。

PKE 接到开始命令后，开始进行运算，运算过程中，上位机可以通过状态寄存器查询目前的运行状态，也可以通过控制寄存器来中断目前的运行。另外，通过访问数据 RAM 地址可以获得部分中间运算结果。

上位机可以通过轮询或中断的方式来获取 PKE 是否完成目标运算的结果。数据 RAM 都是双字（64-bit）对齐，不支持字节对齐。

4.3 TRNG

4.3.1 模块概述

TRNG 模块通过物理随机源产生随机序列，后经 SM4 均衡处理，生成真随机数，为 SM2、RSA 等非对称算法提供密钥对。

4.3.2 模块特性

- 符合 GM/T 0005-2012《随机性检测规范》
- 符合 NIST SP800-90 a/b/c 的要求
- 集成 4 路物理随机源
- 具有在线健康检测功能
- 随机数生成速率≥30Mbps

5 USB Device 接口

5.1 模块概述

USB300 是一个通用串行总线设备控制器，符合 USB3.0 协议。主要包含 DMA 控制器，AHB 从接口和 AHB 主接口等主要组成。支持 SS 模式，速度最快可达 5Gbps，支持 HS 模式，速度最快可达 480Mbps；支持 FS 模式，速度最快可达 12Mbps。

模块的功能框图如下：

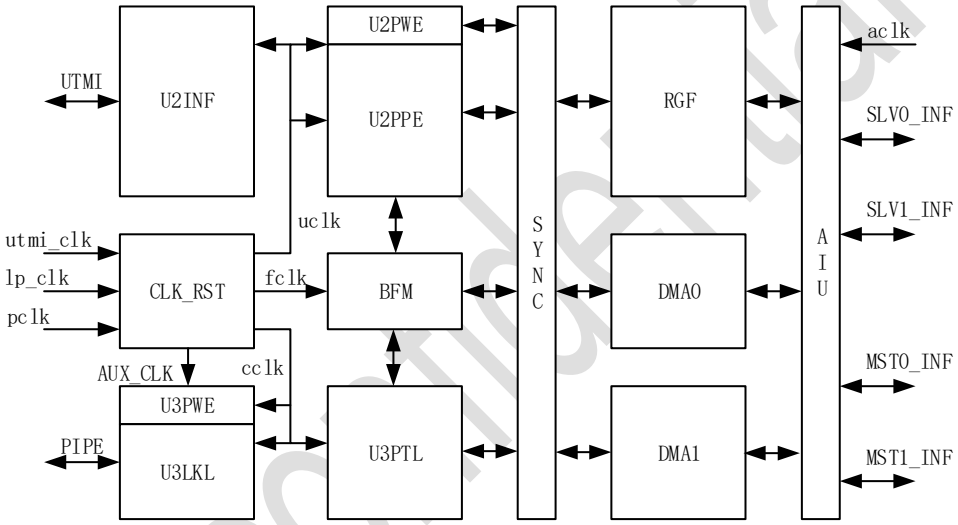


图 5.1 USB 内部结构框图

5.2 模块特性

- 支持 SS/HS/FS 速度模式
- 支持控制/批量/中断/等时传输
- 支持优异的功耗管理，USB3.0 模式下支持 U0/U1/U2/U3，USB2.0 模式下支持 LPM
- 支持内部 DMA
- 支持 9 个端点
- 支持每个端点的 FIFO 深度可配置
- 支持大批量数据流协议
- 支持 USB3.0 和 USB2.0 共享硬件寄存器
- 支持每个端点享有独立的 PRD 列表

6 存储接口

6.1 eMMC 控制器

6.1.1 模块概述

eMMC 控制器（以下简称 eMMC）是嵌入式多媒体设备的主机端控制器，去遵循 eMMC 标准协议，主要用于完成同 eMMC 器件命令及数据的交互，在固件的配合下，该模块可支持 eMMC5.1 协议相关特性，并向下兼容。eMMC 架构框图如下：

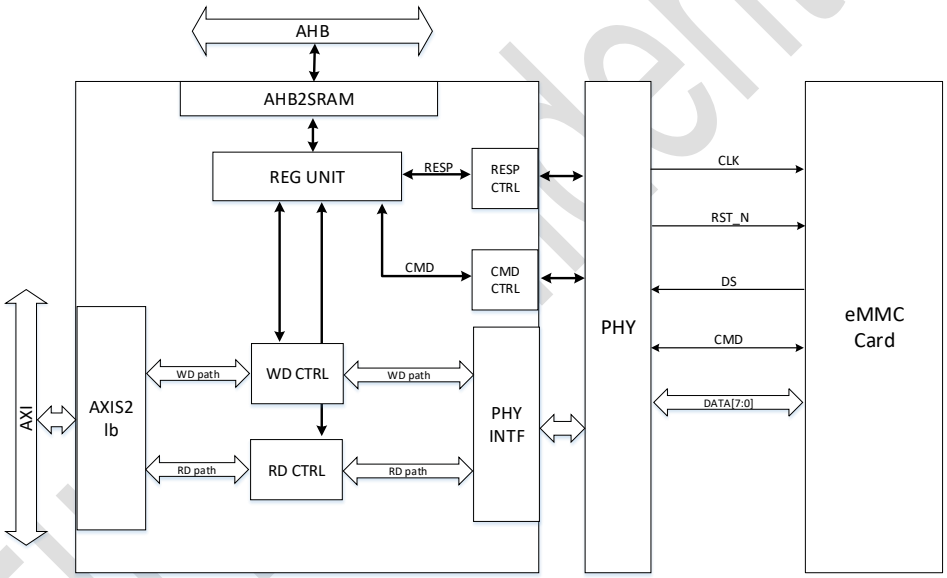


图 6.1 eMMC 控制器结构框图

6.1.2 模块特性

- 支持 1 路 AHB 从配置端口
- 支持 1 路 AXI 从数据传输端口
- 支持 eMMC 5.1 协议标准命令、响应及数据传输格式
- 硬件不支持 QE 操作，其功能由固件驱动实现
- 支持 1/4/8 线传输模式
- 支持 eMMC 协议规定的 HS400/HS200/SDR52/DDR52 模式
- 支持 1 路缓存命令通道
- 支持 1 路直接命令通道

- 支持基于块的数据传输模式，数据块长度为 512B
- 支持时钟流控管理功能
- 支持命令 CRC7 校验及数据 CRC16 校验
- 内置硬件 PHY，集成 DLL 数字锁相环
- 支持 timing 时序可调
- 支持查询和中断两种模式检查命令完成
- 支持超时及错误中断

6.1.3 工作方式

eMMC 寄存器控制模块包含两个命令通道，直接命令通道和缓存命令通道。

缓存命令通道作为一个命令缓存队列，可以缓存多组固定长度的读写命令，并实现响应（response）对比。每一个使用缓存命令通道的命令需要配置四个寄存器，缓存命令参数寄存器，缓存命令寄存器，响应寄存器和响应位使能寄存器。

直接命令通道需要配置两个寄存器，直接命令参数寄存器和直接命令寄存器，另外读写数据还需要配置直接命令字节计数寄存器。直接命令通道的优先级高于缓存命令通道，但无法进行硬件内部的响应对比。直接命令通道的数据传输以字节为单位，字节数由直接命令字节计数寄存器决定，缓存命令通道的数据传输以块（512 字节）为单位，传输长度由缓存命令寄存器高 16 位决定。直接命令通道每次只能发一个命令，需要等到命令（不带响应的）发送完成或响应返回后才能发下一个命令，缓存命令通道可以一次存入多条固定长度块传输的命令，然后 eMMC 控制器会依次执行并进行响应对比。

6.2 SD Device 控制器

6.2.1 模块概述

SD Device 控制器是一个高度可配置的设备端控制器，支持 SD Physical Layer Specification Version 3.0 并向下兼容。SD 控制器支持 SPI/SD 的 1/4 线模式。控制器最高可支持到 208MHz 的时钟频率。

SD Device 控制器支持 AHB 主/从接口，支持 DMA 模式进行数据传输。系统处理器通过 SD 控制器的 AHB 从接口对控制器内部寄存器进行配置，架构框图如下：

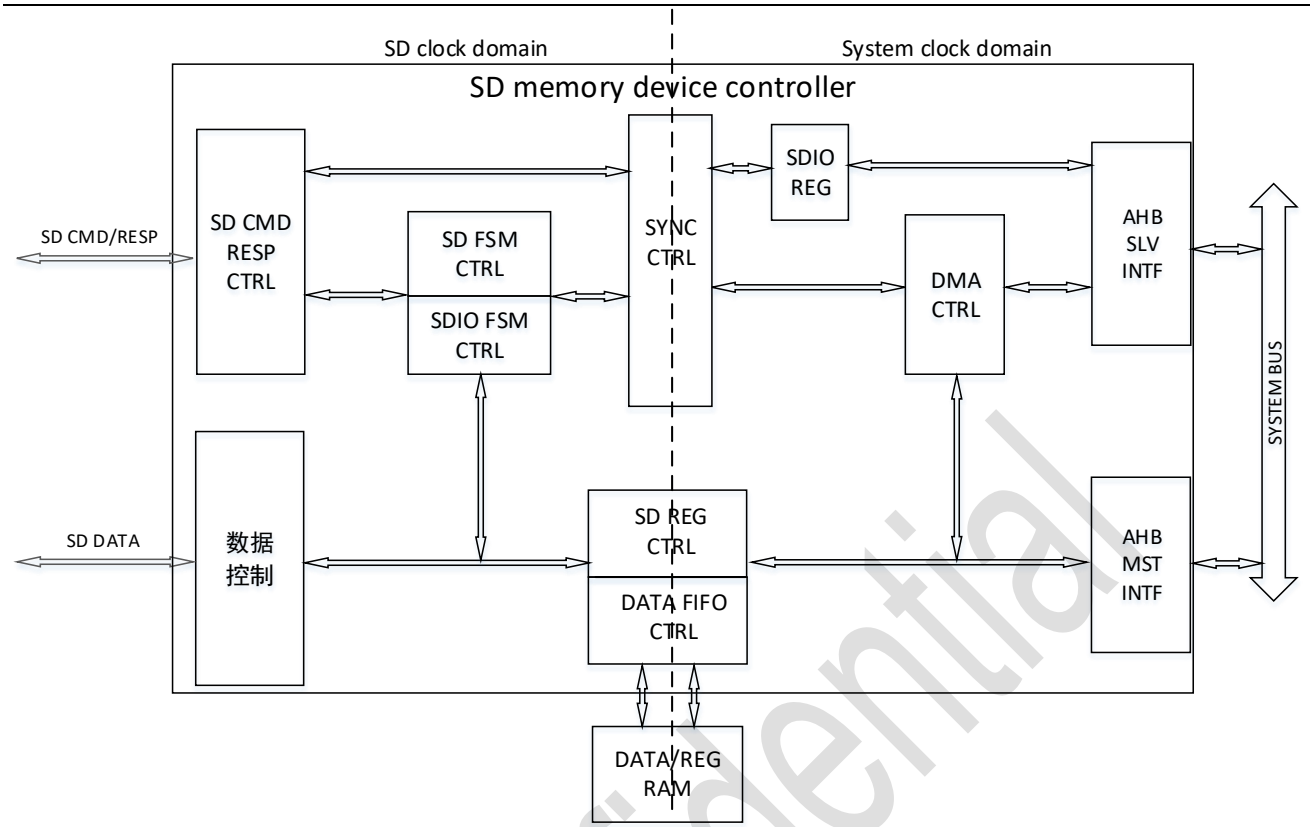


图 6.2 SD 存储设备控制器结构框图

6.2.2 模块特性

- 支持 Part1 SD Physical Layer Specification Ver3.01
- 支持 Part E1 SDIO Specification Version 3.00
- 支持 AMBA Specification 2.0
- 支持 1/4 线 SD 总线，SPI 模式
- 支持 1/4 线 SDR 和 4 线 DDR 模式（SDR12，SDR25，SDR50，SDR104，DDR50）
- SDIO 模式下支持实现 2 个功能
- 向下兼容 SD2.0 和 SDIO2.0
- 支持用于高速数据传输的 DMA 模式
- 支持读等待机制
- SD 时钟最高到 208MHz
- 数据传输速率最高到 832Mbps

6.2.3 工作方式

SD Device 控制器包含命令和数据两个通道。

首先 SD Device 控制器从命令通道接收 SD Host 控制器发过来的命令，给到命令控制单元。

命令控制单元产生相应的命令响应返回给 SD Host 控制器，同时对命令进行解析产生相应的控制信号给控制状态机和同步控制单元，命令状态机根据对应的控制信号进行跳转。同步控制单元负责将 SD 时钟域的信号同步到系统时钟域。DMA 控制单元则根据同步模块发出的控制信号以及系统从接口给出的配置信息来决定 DMA 的启动和状态跳转。

对于数据通路，若当前数据传输写数据操作，数据控制单元负责接收 SD 接口传输的数据，并将数据保存在数据 FIFO 控制单元，然后 DMA 控制单元将数据取出，由 AHB 主接口将数据发送出去；若当前数据传输为读数据操作，命令通道接收到相应的读命令，CPU 通过 AHB 从接口进行配置，由 DMA 控制 AHB 主接口发出读数据命令，将数据读入数据 FIFO 控制单元，再由数据控制单元将数据从 FIFO 中读出，通过 SD Device 接口数据通道发送给 SD Host 控制器。

6.3 SD Host 控制器

6.3.1 模块概述

SD Host 控制器是访问符合包括 SDIO 规范和 SD 存储卡规范的主机控制器。SD Host 为数据传输提供可编程的 I/O 和 DMA 模式。DMA 模式通过 AHB Master 接口在存储器和 SD 卡之间进行数据传输，而不会被 CPU 中断，DMA 还支持单块传输，多块传输和不限长传输方式。

DMA 传输分为 SDMA 和 ADMA 两种模式。SDMA 模式在当前传输结束后需要 CPU 重新配置下一次传输；ADMA 模式采用配置链表的方式，控制器自动读取链表配置，可以实现更高的传输性能。

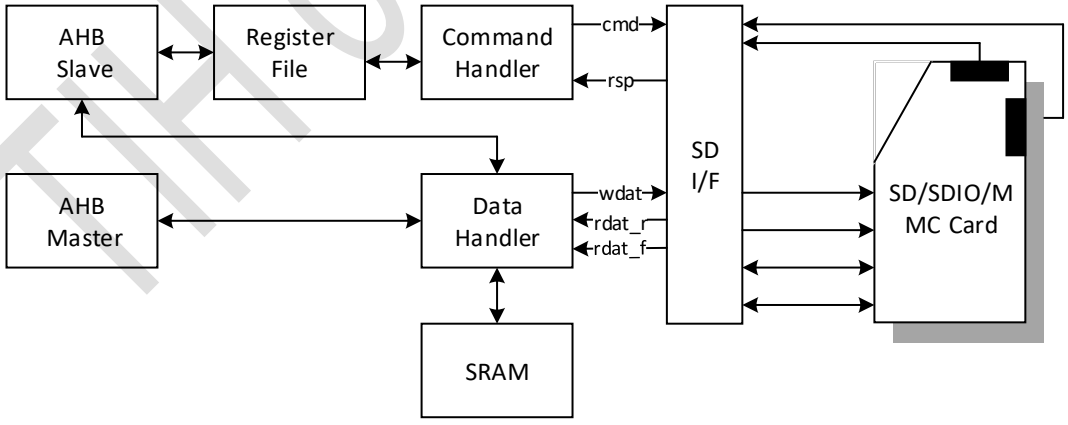


图 6.3 SD Host 结构框图

6.3.2 模块特性

- 支持 SD host controller3.0 标准

- 支持 SD phy layer3.0 标准
- 支持 DMA 和非 DMA 数据传输方式
- 支持 UHS50/UHS104 SD 卡
- 支持 1 线/4 线数据传输模式
- 兼容 SDIO3.0 协议标准
- 兼容 eMMC5.1 协议标准
- 支持插拔检测
- 支持 SDIO 读等待机制

7 外围设备接口

7.1 I2C 控制器

7.1.1 模块概述

I2C 控制器挂载于 APB bus 上，可作为 I2C master 外接 I2C 接口设备，或作为 I2C slave 外接 MCU。

I2C 控制器框图如下：

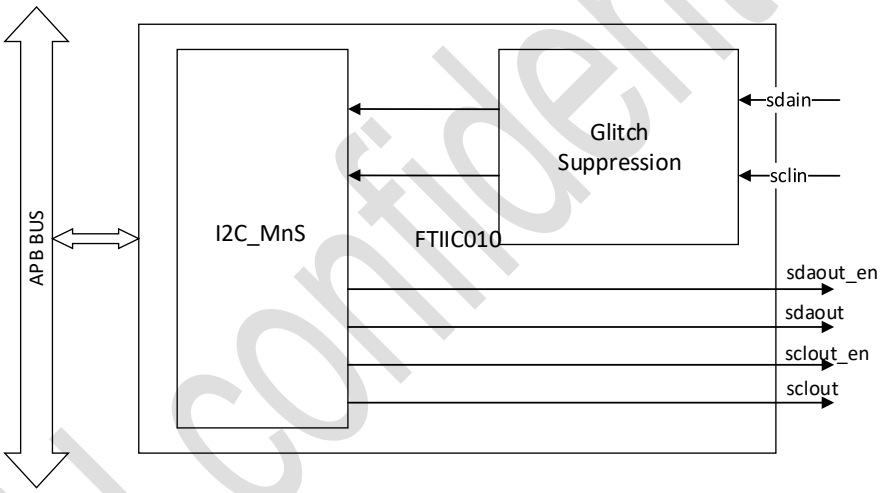


图 7.1 I2C 控制器结构框图

7.1.2 模块特性

- 支持 Standard，Fast and Fast+ modes
- 支持 HS-mode
- 支持 7/10-bit 地址模式
- 支持总线毛刺过滤
- 支持主从模式
- 从模式地址可配
- 支持 master-TX，master-RX，slave-TX，slave-RX 模式
- 集成 32 byte 数据 buffer
- 支持 General Call 和 Start Byte 功能

7.1.3 工作方式

I2C 控制器可工作在如下模式：

- TX/RX in Slave Mode
- RX in Slave Mode with Repeat-Start
- TX in Master Mode
- RX in Master Mode
- TX in Master Mode with HS-Mode (or START Byte)
- RX in Master Mode with HS-Mode (or START Byte)
- TX/RX in Slave Mode with HS-Mode (or START Byte)
- Master TX Burst Mode
- Master RX Burst Mode

7.2 SPI Flash 控制器

7.2.1 模块概述

SPI Flash 控制器主要用于外扩 SPI SRAM、SPI flash 外设等。

SPI Flash 控制器架构如下：

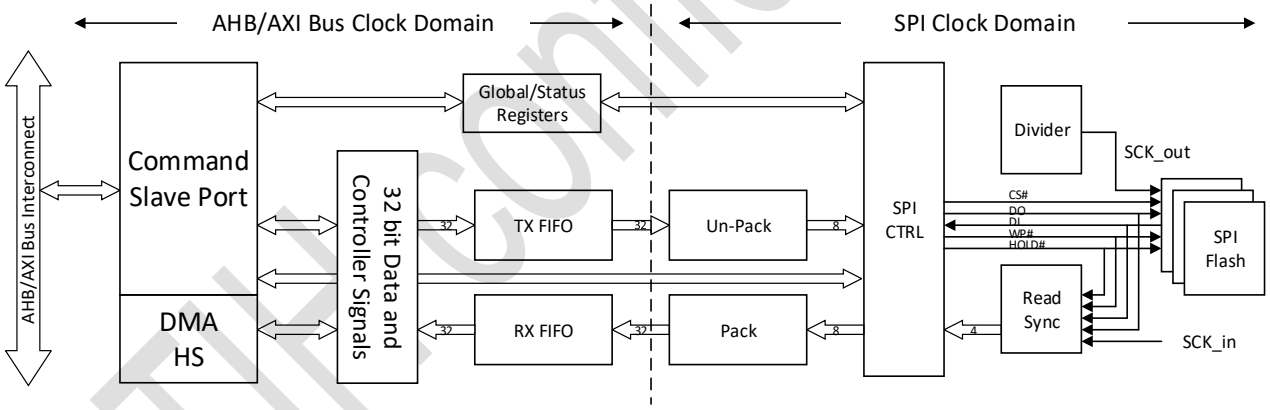


图 7.2 SPI Flash 控制器结构框图

7.2.2 模块特性

- 控制器时钟和接口时钟异步可调
- 支持 SPI 单线/双线/四线模式
- 最高接口工作频率 100 MHz

7.3 SPI 控制器

7.3.1 模块概述

SPI控制器挂载于APB总线上，符合Motorola总线协议，可作为SPI 主从设备进行外设扩展，操作简单、可扩展性强。

SPI模块结构如下：

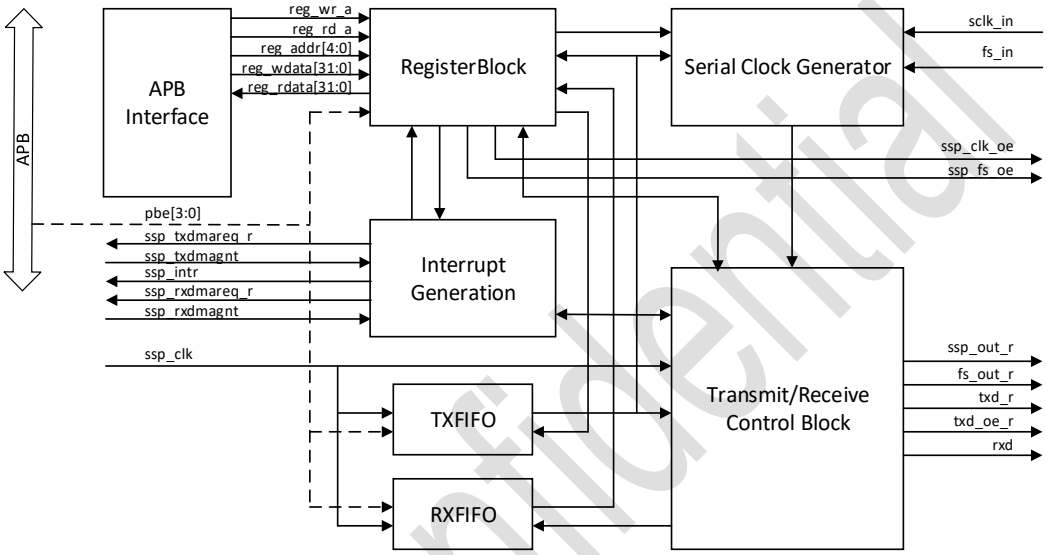


图 7.3 SPI 控制器结构框图

7.3.2 模块特性

- 支持 Motorola SPI 协议标准
- 最高接口工作频率 20 MHz
- 支持主从模式
- 输出时钟的极性、相位、频率可配
- 串行数据支持 MSB 或者 LSB first 模式
- 集成 32bytes TXFIFO
- 集成 32bytes RXFIFO
- TXFIFO/RXFIFO 阈值中断可配
- 支持中断和查询模式
- 独立的 SPI 工作时钟
- 独立可配置的中断使能

7.4 UART0 控制器

7.4.1 模块概述

UART0 控制器与通用的 16C550 UART 完全兼容。

UART0 控制器架构如下：

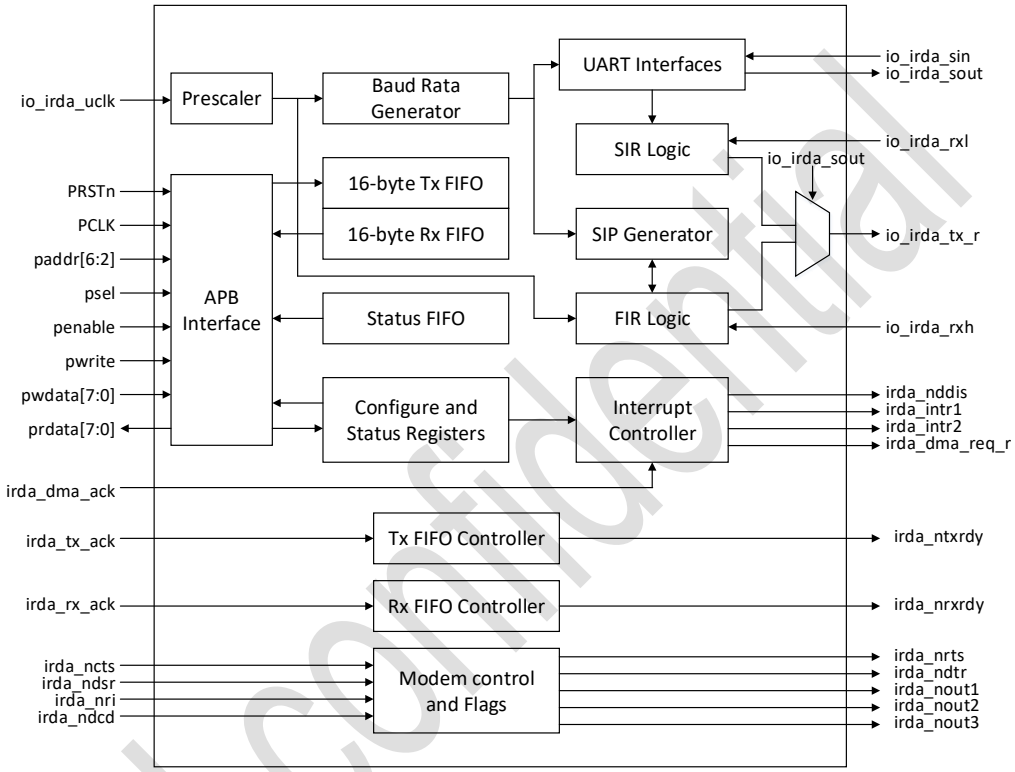


图 7.4 UART 控制器结构框图

*注：上图中 Irda 功能本芯片中未支持

7.4.2 模块特性

- 完全兼容高速 NS 16C550A UART
- 最高波特率为 3Mbit/s
- 集成 32bytes TX FIFO
- 集成 32bytes RX FIFO
- 支持奇偶校验方式或无校验
- 支持帧错误检测
- 支持 FIFO 溢出报警
- 波特率可配置
- 支持数据位和停止位的位宽配置，数据位宽可配置为 5/6/7/8bits，停止位可配置为

1/1.5/2bits

7.5 UART1 控制器

UART1 控制器与 UART0 控制器内部结构及逻辑完全相同，只是基地址不同。

7.6 GPIO 控制器

7.6.1 模块描述

GPIO 提供 16 位可编程的输入输出管脚。每个管脚可配置为输入或输出。管脚用于生成特定应用的输出信号或采集特定应用的输入信号。输入管脚，GPIO 可作为中断源；输出管脚，每个 GPIO 都可以独立地清 0 或置 1。

GPIO 的 16 个管脚输入状态下也可以根据电平或跳变值产生可屏蔽中断。

GPIO 模块结构图如下：

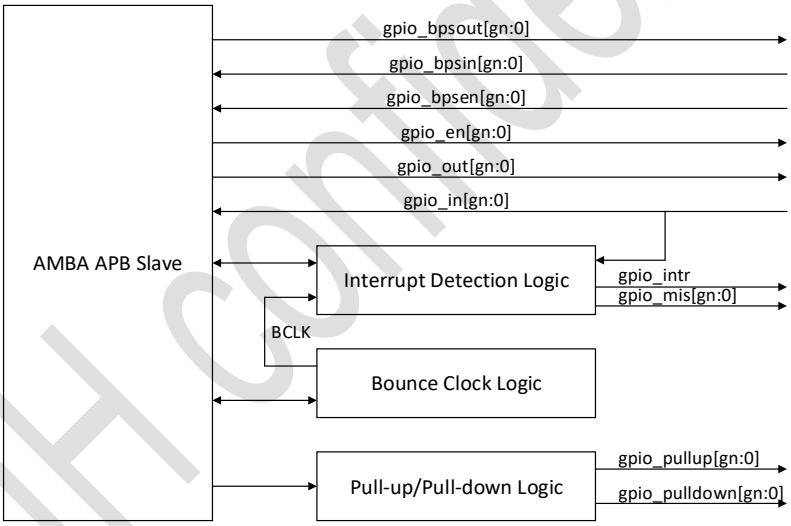


图 7.5 GPIO 控制器结构框图

7.6.2 模块特性

- 16 个管脚可独立设置为输入或输出
- 每个管脚均可以设置为 bypass 模式
- 每个管脚输入状态下可作为中断源
- 输入中断源可以设置为电平触发或边沿触发
- 每个端口可通过 SCU 配置为上拉或下拉
- 输出状态下每个 bit 都可单独设置 0 或 1
- 所有管脚上电复位后默认为输入

8 安全特性

8.1 电压检测

8.1.1 模块概述

电压检测模块 VDT 用于检测当前 IO 电压是否正常，当 IO 电压低于配置电压时，电压检测模块将触发 CPU 中断，可有效防止各种电压攻击手段。

电压检测模块结构如下图所示：

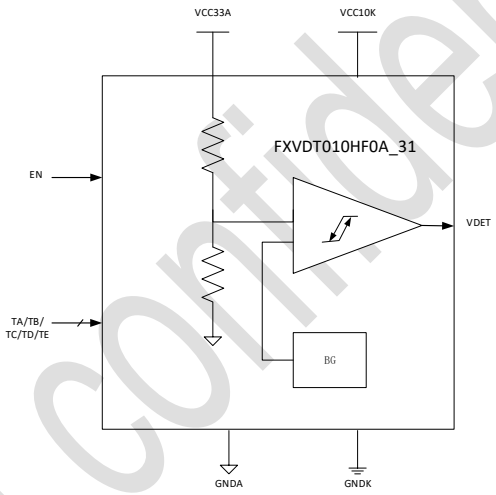


图 8.1 电压检测模块框图

8.1.2 模块特性

- 节点工作温度范围-40~125℃
- 支持低功耗模式
- 支持检测电压阈值微调

8.2 物理探测防护

8.2.1 金属屏蔽层

芯片采用 Power mesh 方法增加了金属屏蔽层，可有效防止芯片外部的电磁攻击。

8.2.2 后端设计防护

采用 Chip Level 层 Flatten 的方法，将接口电路、功能电路、密码算法电路和随机电路等完全进行混合布线，可有效防止后端电路反向分析等外部攻击。

8.3 芯片 ID

8.3.1 模块概述

芯片内置 OTP（One Time Programmable）电路，提供一次性编程机会，可作为芯片全球唯一识别号。

8.3.2 模块特性

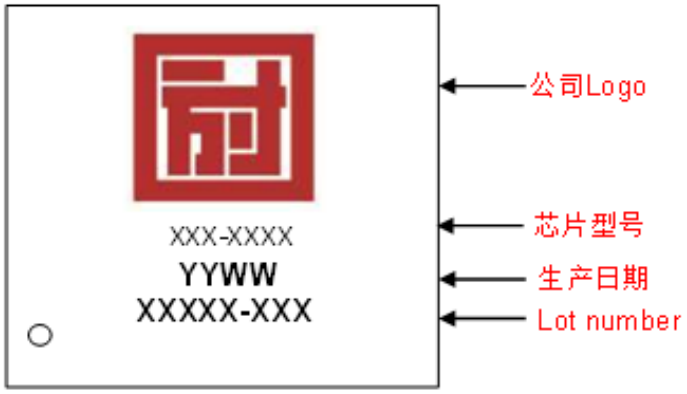
- 有效数据位宽 64bits
- 可支持出厂烧写和用户烧写 2 种模式
- 用户可自定义烧写内容
- 支持低功耗模式

订购信息

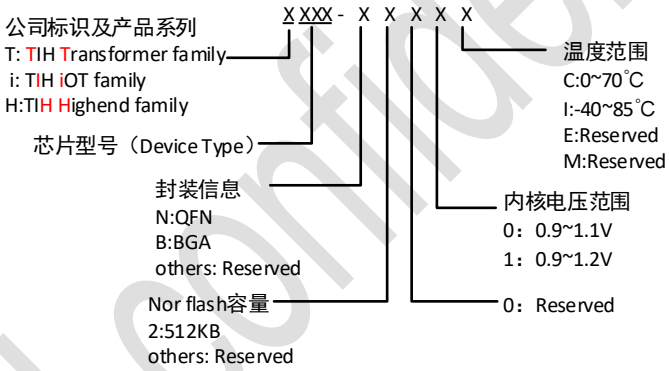
芯片名称	flash 容量	封装信息	温度范围	Package Qty
i560-N200C	512KB	QFN64	0~70℃	1680

*注：Package Qty 表示单包芯片数量。

芯片外部丝印



芯片命名规则



举例