



T680

网络存储安全芯片 技术手册

修改记录

版本号	描述	日期
v0.1	草稿版	2018/11/8
v1.0	初版发布版本	2019/4/3
v1.1	修改部分细节说明	2019/4/25
v1.2	增加 QSPI 描述	2019/6/4



Table of Content

1	概述.....	1
1.1	产品简介	1
1.2	应用产品	1
1.3	芯片架构	2
1.4	芯片特性	2
1.4.1	CPU 资源	2
1.4.2	USB3.0 OTG 接口	2
1.4.3	SATA3.0（主/从）接口	3
1.4.4	eMMC 接口	3
1.4.5	GMAC 接口	3
1.4.6	安全引擎	3
1.4.7	存储资源	4
1.4.8	其他资源	4
1.4.9	安全特性	4
1.4.10	物理规格	4
1.5	地址映射	5
1.6	中断源	5
2	硬件特性.....	7
2.1	芯片封装	7
2.2	管脚分布	8
2.3	管脚描述	11
2.4	管脚复用	18
2.5	上电时序	20
2.6	电性能参数	20
2.7	功耗	21
2.8	PCB 设计建议	21
3	CPU 子系统	22
3.1	CK803S 处理器	22
3.1.1	简介	22
3.1.2	特性	22
3.1.3	架构	23
3.1.4	矢量中断控制器	23
3.1.5	系统计时器	24
3.2	存储	24
3.3	DMA	25
3.3.1	模块概述	25
3.3.2	模块特性	25
3.4	定时器	26
3.4.1	模块概述	26
3.4.2	模块特性	26



3.5	看门狗.....	27
3.5.1	模块概述.....	27
3.5.2	模块特性.....	27
3.6	SCU.....	27
3.6.1	模块概述.....	27
3.6.2	模块特性.....	28
3.6.3	时钟树.....	29
3.6.4	复位树.....	30
4	安全引擎.....	31
4.1	CRYPTO 引擎.....	31
4.1.1	模块概述.....	31
4.1.2	模块特性.....	31
4.1.3	工作方式.....	32
4.2	PKE 引擎.....	34
4.2.1	模块概述.....	34
4.2.2	模块特性.....	34
4.2.3	工作方式.....	35
4.3	TRNG.....	35
4.3.1	模块概述.....	35
4.3.2	模块特性.....	35
5	网络接口.....	37
5.1	GMAC0 控制器.....	37
5.1.1	模块概述.....	37
5.1.2	模块特性.....	38
5.1.3	工作方式.....	38
5.2	GMAC1 控制器.....	39
6	USB OTG 接口.....	40
6.1	模块概述.....	40
6.2	模块特性.....	40
7	SATA 接口.....	42
7.1	SATA Host 控制器.....	42
7.1.1	模块概述.....	42
7.1.2	模块特性.....	43
7.2	SATA Device 控制器.....	43
7.2.1	模块概述.....	43
7.2.2	模块特性.....	44
8	存储接口.....	45
8.1	eMMC0 控制器.....	45
8.1.1	模块概述.....	45
8.1.2	模块特性.....	45
8.1.3	工作方式.....	46
8.2	eMMC1 控制器.....	46



9	外围设备接口.....	47
9.1	MUXIO 接口.....	47
9.1.1	模块概述.....	47
9.1.2	模块特性.....	47
9.1.3	工作方式.....	49
9.2	I2C 控制器.....	56
9.2.1	模块概述.....	56
9.2.2	模块特性.....	57
9.2.3	工作方式.....	57
9.3	QSPI 控制器.....	57
9.3.1	模块概述.....	57
9.3.2	模块特性.....	58
9.4	SPI 控制器.....	58
9.4.1	模块概述.....	58
9.4.2	模块特性.....	59
9.5	UART0 控制器.....	59
9.5.1	模块概述.....	59
9.5.2	模块特性.....	60
9.6	UART1 控制器.....	61
9.7	GPIO0 控制器.....	61
9.7.1	模块描述.....	61
9.7.2	模块特性.....	61
9.8	GPIO1 控制器.....	62
10	安全特性.....	63
10.1	电压检测.....	63
10.1.1	模块概述.....	63
10.1.2	模块特性.....	63
10.2	温度检测.....	64
10.2.1	模块概述.....	64
10.2.2	模块特性.....	64
10.2.3	模块时序.....	65
10.3	物理探测防护.....	65
10.3.1	金属屏蔽层.....	65
10.3.2	后端设计防护.....	65
10.4	芯片 ID.....	65
10.4.1	模块概述.....	65
10.4.2	模块特性.....	66

1 概述

1.1 产品简介

T680 是由方寸微电子自主开发的新一代 Soc 网络终端存储安全芯片，具有功能丰富、性能强劲、功耗低、安全性高等特点，可广泛适用于加密移动硬盘、加密固态硬盘、视频链路加密机、VPN 终端网关、安全网关、网闸、单向导入导出设备、USB 安全网卡、密码卡、密码机、USB 接口芯片等众多安全领域产品。

该芯片集成国产 32 位高性能 RISC CPU，可支持 USB3.0、SATA3.0、GMAC、eMMC 等多种超高速接口，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

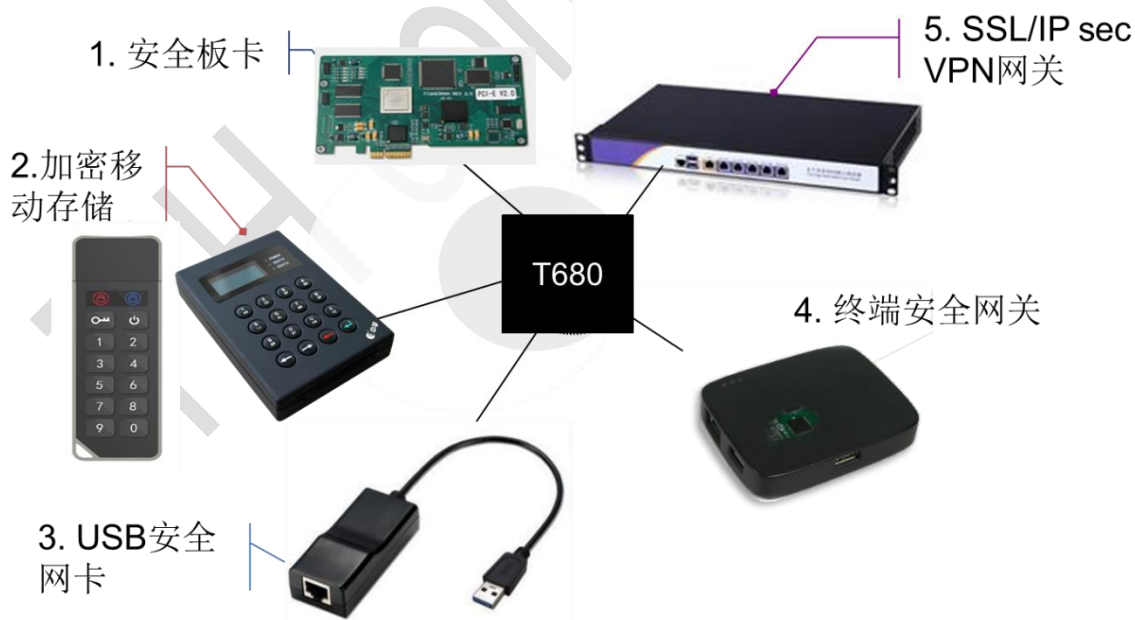


图 1.1 产品应用方案

1.3 芯片架构

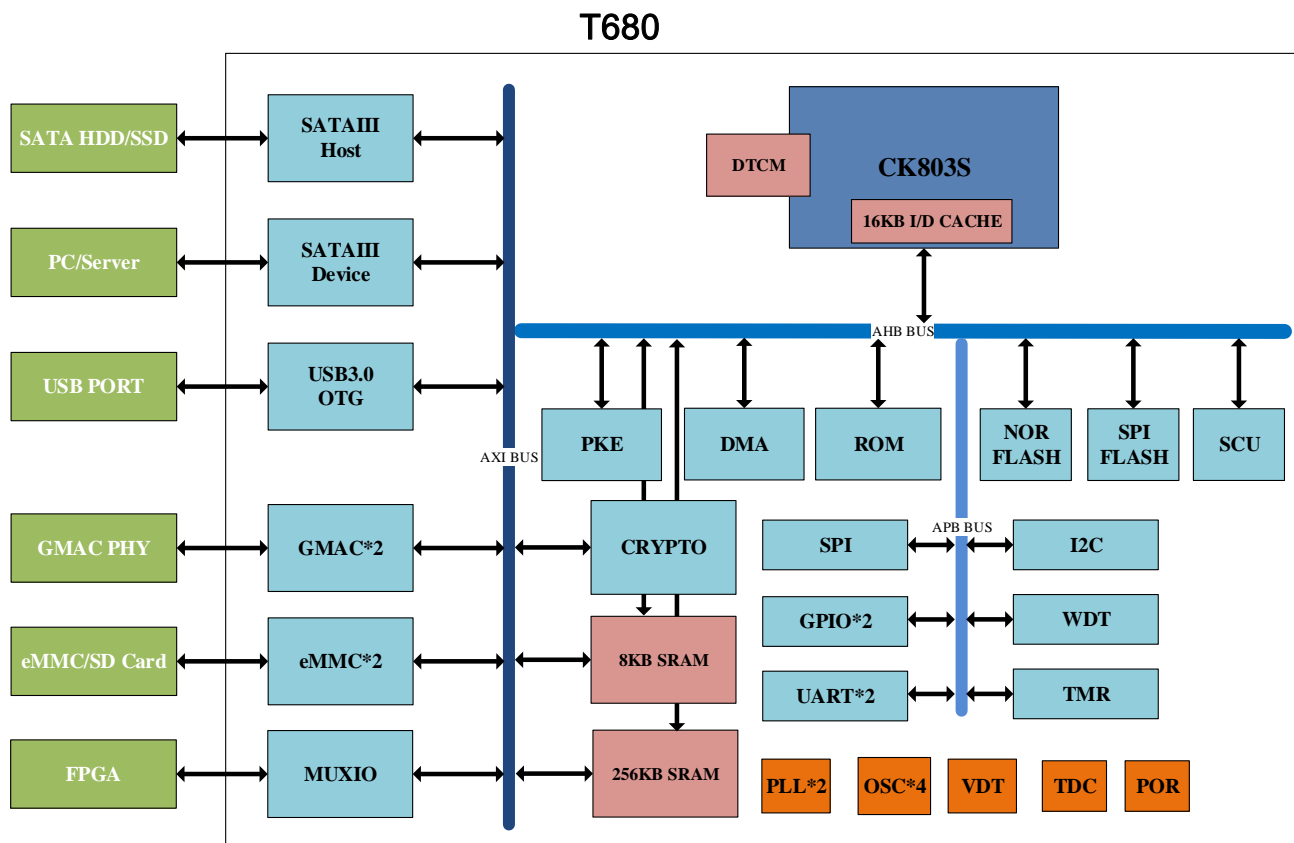


图 1.2 芯片系统架构框图

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 260Mhz
- 内置 16KB I/D Cache
- 内置 DTCM 32KB

1.4.2 USB3.0 OTG 接口

- 支持一路 USB3.0 OTG 接口速率 5Gbps，向下兼容 USB2.0/USB1.1
- 静态角色转换（主机/设备选择）



- 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 SATA3.0 (主/从) 接口

- 支持一路 SATAIII device 接口速率 6Gbps, 向下兼容 3Gbps/1.5Gbps
- 支持一路 SATAIII host 接口速率 6Gbps, 向下兼容 3Gbps/1.5Gbps
- 符合 Serial ATA Revision 3.0 标准协议
- 支持 NCQ 32 命令队列

1.4.4 eMMC 接口

- 支持 2 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

1.4.5 GMAC 接口

- 支持 2 路 GMAC RGMII 接口
- 外挂 PHY 可支持 10/100/1000Mbps 网络接口速率

1.4.6 安全引擎

- 支持 SM4、AES256 数据加密, 加密性能 800MB/s@200Mhz
- 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 500 对/s
- 支持 SM2 签名验签, 性能 $\geq 1200/600$ 次/s@200Mhz
- RSA1024 密钥对生成时间 $<0.1s$
- 支持 RSA1024 签名验签, 性能 $\geq 1200/12000$ 次/s@200Mhz
- RSA2048 密钥对生成时间 $<1s$
- 支持 RSA2048 签名验签, 性能 $\geq 200/4000$ 次/s@200Mhz
- 支持 SM3/SHA1/SHA224/SHA256 算法
- 支持一路 TRNG 发生器, 生成速率 $\geq 30Mbps@50Mhz$

*以上为硬件引擎性能, 非最终产品性能

1.4.7 存储资源

- 32KB ROM
- 256KB SRAM
- 8KB SRAM（系统专用）
- 512KB 片内 flash

1.4.8 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 QSPI 主接口（仅用于连接 SPI Flash/SPI Ram 等）
- 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 32 位数据位宽 MUXIO 接口
- 支持 48 位 GPIO 接口
- 支持在线调试

1.4.9 安全特性

- 支持电压检测
- 支持温度检测
- 支持物理探测防护
- 每颗芯片具备全球唯一 ID

1.4.10 物理规格

- Core 电压为 1.0V
- IO 电压为 3.3V
- 支持 BGA220 11mm x 11mm x1.2mm 封装
- 工作温度 0~70°C, -40~85°C

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	
0x1000_0000	1MB	SATA Device 寄存器端口	
0x1010_0000	1MB	CRYPTO 寄存器端口	
0x1020_0000	1MB	DMA 寄存器端口	
0x1040_0000	1MB	QSPI 寄存器端口	
0x1050_0000	1MB	SCU 寄存器端口	
0x1060_0000	1MB	eMMC0 寄存器端口	
0x1070_0000	1MB	eMMC1 寄存器端口	
0x1080_0000	1MB	GMAC0 寄存器端口	
0x1090_0000	1MB	GMAC1 寄存器端口	
0x10A0_0000	1MB	MUXIO 寄存器端口	
0x10B0_0000	1MB	INTC 寄存器端口	
0x10C0_0000	1MB	TRNG 寄存器端口	
0x10D0_0000	1MB	PKE 寄存器端口	
0x1110_0000	1MB	8KB SRAM	8KB SRAM 在 AHB 总线地址
0x1120_0000	1MB	256KB SRAM	256KB SRAM 在 AHB 总线地址
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	
0x1270_0000	1MB	GPIO0 寄存器端口	
0x1280_0000	1MB	GPIO1 寄存器端口	
0x12A0_0000	1MB	IIC 寄存器端口	
0x2000_0000	32MB	MUXIO_S2 数据端口	MUXIO AXI 总线地址
0x2200_0000	1MB	USB OTG 寄存器端口	
0x2210_0000	1MB	SATA Host 寄存器端口	
0x2220_0000	1MB	8KB SRAM	8KB SRAM 在 AXI 总线地址
0x2230_0000	1MB	256KB SRAM	256KB SRAM 在 AXI 总线地址
0x2240_0000	1MB	eMMC0_S 数据端口	
0x2250_0000	1MB	eMMC1_S 数据端口	
0x2300_0000	8MB	CRYPTO_S1 数据端口	
0x2380_0000	8MB	CRYPTO_S2 数据端口	
0x3000_0000	32MB	MUXIO_S1 数据端口	MUXIO AHB 总线地址

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
26	Core_Timer	
25	VDT	
24	Reserved	
23	TRNG	
22	PKE	
21	AXIC	
20	AHBC	
19	GPIO1	
18	GPIO0	
17	WDT	
16	IIC	
15	UART_1	
14	UART_0	
13	QSPI	
12	Nor-flash	
11	SPI	
10	TIMER	
9	DMA	
8	MUXIO	
7	GMAC_1	
6	GMAC_0	
5	eMMC_1	
4	eMMC_0	
3	CRYPTO	
2	SATA3_Host	
1	SATA3_Slave	
0	USB3_OTG	

2 硬件特性

2.1 芯片封装

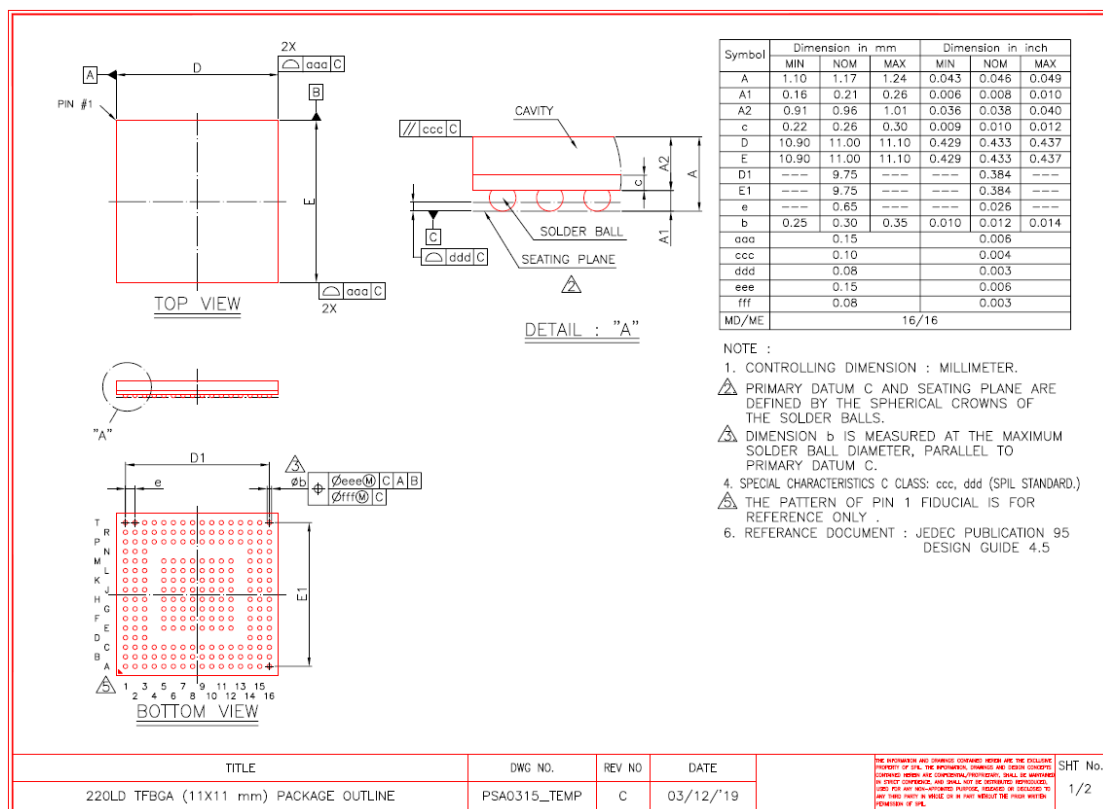


图 2.1 芯片封装尺寸图

2.2 管脚分布

	1	2	3	4	5	6	7	8
A	X_RSTN_ O	X_TM	X_JTG_T CK	X_SD_T XN	GNDA_ SD	X_SD_R XN	GNDA_ SD	X_SH_ TXP
B	X_DM	X_DP	X_JTG_T MS	X_SD_T XP	GNDA_ SD	X_SD_R XP	GNDA_ SD	X_SH_ TXN
C	GNDA_US B	GNDA_U SB	X_JTG_T RST_N	VCC33A _TDC	X_SD_ REXT	GNDA_ SD	VCC33 A_SD	VCC33 A_SH
D	X_SSTXB	X_SSTX A	VCC33A_ USB	NC	NC	NC	NC	NC
E	GNDA_US B	GNDA_U SB	VCC10A_ USB	NC	VCC33I O	GND10_ PLL0	VCC10_ PLL0	VCC10 A_SD
F	X_SSRXB	X_SSRX A	X_I2C_SC L	NC	GNDK	GNDK	GNDK	GNDK
G	GNDA_US B	GNDA_U SB	X_I2C_SD A	NC	VCCK	GNDK	GNDK	GNDK
H	X_ETH0_ TXCTL	X_ETH0_ TXD3	X_ETH0_ MDIO	NC	VCCK	GNDK	GNDK	GNDK
	9	10	11	12	13	14	15	16
A	GNDA_ SH	X_SH_R XN	GNDA_ SH	X_XSCI_I	GNDK	X_ICE_T DO	X_GPIO 0_10	X_GPIO 0_8
B	GNDA_ SH	X_SH_R XP	GNDA_ SH	X_XSCI_O	X_ICE_ TMS	X_ICE_N TRST	X_GPIO 0_11	X_GPIO 0_9
C	X_SH_R EXT	GNDA_ SH	GND_O SC	VCC33_OS C	X_ICE_ TCK	X_ICE_T DI	X_GPIO 0_14	X_GPIO 0_12
D	NC	NC	NC	NC	NC	X_GPIO0 _24	X_GPIO 0_15	X_GPIO 0_13

E	VCC10A _SH	VCC11 A_SD	VCC11 A_SH	VCC33IO	NC	X_GPIO0 _25	X_GPIO 0_18	X_GPIO 0_19
F	GNDK	GNDK	GNDK	VCC3318_ GPIO0	NC	X_GPIO0 _26	X_GPIO 0_22	X_GPIO 0_21
G	GNDK	GNDK	GNDK	VCC3318_ GPIO0	NC	X_GPIO0 _27	X_GPIO 0_23	X_GPIO 0_17
H	GNDK	GNDK	GNDK	VCCK	NC	X_GPIO0 _28	X_GPIO 0_5	X_GPIO 0_16

1 2 3 4 5 6 7 8

J	X_ETH0_ TXD2	X_ETH0_ _TCK	X_ETH0_ MDC	NC	GNDK	GNDK	GNDK	GNDK
K	X_ETH0_ TXD1	X_ETH0_ _TXD0	X_ETH0_ RXCTL	NC	VCCK	GNDK	GNDK	GNDK
L	X_ETH0_ RXD3	X_ETH0_ _RXD2	X_ETH0_ RXD0	NC	VCC331 8_ETH	GNDK	GNDK	GNDK
M	X_ETH0_ RCK	X_ETH0_ _RXD1	X_ETH1_ MDC	NC	VCC331 8_ETH	VCC10_DL L_EMMC	M_RE F	GNDK
N	X_ETH1_ TXCTL	X_ETH1_ _TXD3	X_ETH1_ MDIO	NC	NC	NC	NC	NC
P	X_ETH1_ TXD2	X_ETH1_ _TCK	X_ETH1_ RXCTL	X_ETH1_ _RXD0	VCC331 8_ETH	X_M0_DS	X_M0_ _CLK	X_M0_ CMD
R	X_ETH1_ TXD1	X_ETH1_ _TXD0	X_ETH1_ RCK	X_M0_D 4	X_M0_ D5	X_M0_D6	X_M0_ _D7	X_M1_ D4
T	X_ETH1_ RXD3	X_ETH1_ _RXD2	X_ETH1_ RXD1	X_M0_D 3	X_M0_ D0	X_M0_D1	X_M0_ _D2	X_M1_ D3



	9	10	11	12	13	14	15	16
J	GNDK	GNDK	GNDK	VCCK	NC	X_GPI O0_29	X_GPI O0_2	X_GPI O0_20
K	GNDK	GNDK	GNDK	VCCK	NC	X_GPI O0_30	X_GPI O0_0	X_GPI O0_7
L	GNDK	GNDK	GNDK	VCC3318 _GPIO0	NC	X_GPI O0_31	X_GPI O0_6	X_GPI O0_3
M	VCC3318_ EMMC	VCC3318_ EMMC	VCC3318_ EMMC	VCC3318 _GPIO1	NC	X_GPI O1_6	X_GPI O0_4	X_GPI O0_1
N	NC	NC	NC	NC	NC	X_GPI O1_8	X_GPI O1_9	X_GPI O1_7
P	X_M1_CM D	X_M1_CL K	X_M1_DS	X_RST_N	X_GPI O1_0	X_GPI O1_1	X_GPI O1_13	X_GPI O1_10
R	X_M1_D5	X_M1_D6	X_M1_D7	X_UART0 _TXD	X_GPI O1_2	X_GPI O1_3	X_GPI O1_15	X_GPI O1_11
T	X_M1_D0	X_M1_D1	X_M1_D2	X_UART0 _RXD	X_GPI O1_4	X_GPI O1_5	X_GPI O1_12	X_GPI O1_14

图 2.2 TFBGA220 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
A			
A1	X_RSTN_O	O	芯片复位输出
A2	X_TM	I	测试模式使能信号： 0: 正常功能模式 1: 测试模式
A3	X_JTG_TCK	I	CK803S JTAG 测试输入时钟
A4	X_SD_TXN	O	SATA Device 接口差分输出 N
A5	GNDA_SD		模拟地
A6	X_SD_RXN	I	SATA Device 接口差分输入 N
A7	GNDA_SD		模拟地
A8	X_SH_TXP	O	SATA Host 接口高速差分输出 P
A9	GNDA_SH		模拟地
A10	X_SH_RXN	I	SATA Host 接口高速差分输入 N
A11	GNDA_SH		模拟地
A12	X_XSCI_I	I	系统晶振输入, 30Mhz \pm 30ppm
A13	GNDK		数字地
A14	NC		悬空
A15	X_GPIO0_10	IO	GPIO0 通用输入输出端口 10
A16	X_GPIO0_8	IO	GPIO0 通用输入输出端口 8
B			
B1	X_DM	IO	USB2.0 高速差分输入输出
B2	X_DP	IO	USB2.0 高速差分输入输出
B3	X_JTG_TMS	IO	CK803S JTAG 测试控制信号输入输出接口
B4	X_SD_TXP	O	SATA Device 接口高速差分输出 P
B5	GNDA_SD		模拟地
B6	X_SD_RXP	I	SATA Device 接口高速差分输入 P
B7	GNDA_SD		模拟地
B8	X_SH_TXN	O	SATA Host 接口高速差分输出 N
B9	GNDA_SH		模拟地
B10	X_SH_RXP	I	SATA Host 接口高速差分输入 P
B11	GNDA_SH		模拟地
B12	X_XSCI_O	O	系统晶振输出
B13	NC		悬空
B14	NC		悬空
B15	X_GPIO0_11	IO	GPIO0 通用输入输出端口 11
B16	X_GPIO0_9	IO	GPIO0 通用输入输出端口 9
C			



位置	名称	I/O	功能描述
C1	GNDA_USB		模拟地
C2	GNDA_USB		模拟地
C3	X_JTG_TRST_N	I	CK803S JTAG 测试复位输入
C4	VCC33A_TDC		TDC 3.3v 模拟电源
C5	X_SD_REXT		SATA Device 匹配电阻，需下拉 18K 欧姆到接地信号
C6	GNDA_SD		模拟地
C7	VCC33A_SD		SATA Device 3.3v 模拟电源
C8	VCC33A_SH		SATA Host 3.3v 模拟电源
C9	X_SH_REXT		SATA Host 匹配电阻，需下拉 18K 欧姆到接地信号
C10	GNDA_SH		模拟地
C11	GND_OSC		数字地
C12	VCC33_OSC		系统晶振 3.3v 数字电源
C13	NC		悬空
C14	NC		悬空
C15	X_GPIO0_14	IO	GPIO0 通用输入输出 14
C16	X_GPIO0_12	IO	GPIO0 通用输入输出 12
D			
D1	X_SSTXB	O	USB3.0 超速差分输出 B
D2	X_SSTXA	O	USB3.0 超速差分输出 A
D3	VCC33A_USB		USB 3.3v 模拟电源
D4	NC		悬空
D5	NC		悬空
D6	NC		悬空
D7	NC		悬空
D8	NC		悬空
D9	NC		悬空
D10	NC		悬空
D11	NC		悬空
D12	NC		悬空
D13	NC		悬空
D14	X_GPIO0_24	IO	GPIO0 通用输入输出 24
D15	X_GPIO0_15	IO	GPIO0 通用输入输出 15
D16	X_GPIO0_13	IO	GPIO0 通用输入输出 13
E			
E1	GNDA_USB		模拟地
E2	GNDA_USB		模拟地
E3	VCC10A_USB		USB 1.0v 模拟电源
E4	NC		悬空
E5	VCC33IO		3.3v 数字电源



位置	名称	I/O	功能描述
E6	GND10_PLL0		数字地
E7	VCC10_PLL0		PLL 1.0v 数字电源
E8	VCC10A_SD		SATA Device 1.0v 模拟电源
E9	VCC10A_SH		SATA Host 1.0v 模拟电源
E10	VCC11A_SD	O	SATA Device 1.2v 模拟电源输出, 需外挂 2.2uF 电容
E11	VCC11A_SH	O	SATA Host 1.2v 模拟电源输出, 需外挂 2.2uF 电容
E12	VCC33IO		3.3v 数字电源
E13	NC		悬空
E14	X_GPIO0_25	IO	GPIO0 通用输入输出接口 25
E15	X_GPIO0_18	IO	GPIO0 通用输入输出接口 18
E16	X_GPIO0_19	IO	GPIO0 通用输入输出接口 19
F			
F1	X_SSRXB	I	USB3.0 超速差分输入 B
F2	X_SSRXA	I	USB3.0 超速差分输入 A
F3	X_I2C_SCL	IO	I2C 时钟输入输出接口
F4	NC		悬空
F5	GNDK		数字地
F6	GNDK		数字地
F7	GNDK		数字地
F8	GNDK		数字地
F9	GNDK		数字地
F10	GNDK		数字地
F11	GNDK		数字地
F12	VCC3318_GPIO0		GPIO0 3.3v 或 1.8v 数字电源, 供给 GPIO0[0~31],GPIO1[7~15]信号
F13	NC		悬空
F14	X_GPIO0_26		GPIO0 通用输入输出接口 26
F15	X_GPIO0_22		GPIO0 通用输入输出接口 22
F16	X_GPIO0_21		GPIO0 通用输入输出接口 21
G			
G1	GNDA_USB		模拟地
G2	GNDA_USB		模拟地
G3	X_I2C_SDA	IO	I2C 数据输入输出
G4	NC		悬空
G5	VCCCK		数字电源
G6	GNDK		数字地
G7	GNDK		数字地
G8	GNDK		数字地
G9	GNDK		数字地



位置	名称	I/O	功能描述
G10	GNDK		数字地
G11	GNDK		数字地
G12	VCC3318_GPIO0		GPIO0 3.3v 或 1.8v 数字电源， 供给 GPIO0[0~31],GPIO1[7~15]信号
G13	NC		悬空
G14	X_GPIO0_27	IO	GPIO0 通用输入输出接口 27
G15	X_GPIO0_23	IO	GPIO0 通用输入输出接口 23
G16	X_GPIO0_17	IO	GPIO0 通用输入输出接口 17
H			
H1	X_ETH0_TXCTL	O	GMAC0 TX 端控制信号输出
H2	X_ETH0_TXD3	O	GMAC0 TX 端数据输出端口 3
H3	X_ETH0_MDIO	IO	GMAC0 PHY 的数据输入输出
H4	NC		悬空
H5	VCCK		数字电源
H6	GNDK		数字地
H7	GNDK		数字地
H8	GNDK		数字地
H9	GNDK		数字地
H10	GNDK		数字地
H11	GNDK		数字地
H12	VCCK		数字电源
H13	NC		悬空
H14	X_GPIO0_28	IO	GPIO0 通用输入输出接口 28
H15	X_GPIO0_5	IO	GPIO0 通用输入输出接口 5
H16	X_GPIO0_16	IO	GPIO0 通用输入输出接口 16
J			
J1	X_ETH0_TXD2	O	GMAC0 TX 端数据输出端口 2
J2	X_ETH0_TCK	O	GMAC0 TX 端时钟
J3	X_ETH0_MDC	O	GMAC0 PHY 控制时钟
J4	NC		悬空
J5	GNDK		数字地
J6	GNDK		数字地
J7	GNDK		数字地
J8	GNDK		数字地
J9	GNDK		数字地
J10	GNDK		数字地
J11	GNDK		数字地
J12	VCCK		电源
J13	NC		悬空
J14	X_GPIO0_29	IO	GPIO0 通用输入输出 29



位置	名称	I/O	功能描述
J15	X_GPIO0_2	IO	GPIO0 通用输入输出 2
J16	X_GPIO0_20	IO	GPIO0 通用输入输出 20
K			
K1	X_ETH0_TXD1	O	GMAC0 TX 端数据输出端口 1
K2	X_ETH0_TXD0	O	GMAC0 TX 端数据输出端口 0
K3	X_ETH0_RXCTL	I	GMAC0 RX 控制信号输入
K4	NC		悬空
K5	VCCK		数字电源
K6	GNDK		数字地
K7	GNDK		数字地
K8	GNDK		数字地
K9	GNDK		数字地
K10	GNDK		数字地
K11	GNDK		数字地
K12	VCCK		数字电源
K13	NC		悬空
K14	X_GPIO0_30	IO	GPIO0 通用输入输出 30
K15	X_GPIO0_0	IO	GPIO0 通用输入输出 0
K16	X_GPIO0_7	IO	GPIO0 通用输入输出 7
L			
L1	X_ETH0_RXD3	I	GMAC0 RX 端数据输入端口 3
L2	X_ETH0_RXD2	I	GMAC0 RX 端数据输入端口 2
L3	X_ETH0_RXD0	I	GMAC0 RX 端数据输入端口 0
L4	NC		悬空
L5	VCC3318_ETH		GMAC0 和 GMAC1 3.3v 或 1.8v 数字电源
L6	GNDK		数字地
L7	GNDK		数字地
L8	GNDK		数字地
L9	GNDK		数字地
L10	GNDK		数字地
L11	GNDK		数字地
L12	VCC3318_GPIO0		GPIO0 3.3v 或 1.8v 输入数字电源， 供给 GPIO0[0~31],GPIO1[7~15]信号
L13	NC		悬空
L14	X_GPIO0_31	IO	GPIO0 通用输入输出端口 31
L15	X_GPIO0_6	IO	GPIO0 通用输入输出端口 6
L16	X_GPIO0_3	IO	GPIO0 通用输入输出端口 3
M			
M1	X_ETH0_RCK	I	GMAC0 RX 时钟输入
M2	X_ETH0_RXD1	I	GMAC0 RX 端数据输入端口 1
M3	X_ETH1_MDC	O	GMAC1 PHY 时钟输出



位置	名称	I/O	功能描述
M4	NC		悬空
M5	VCC3318_EHT		3.3v 或 1.8v 数字电源
M6	VCC10_DLL_EMMC		EMMC DLL 1.0v 数字电源
M7	M_REF		悬空
M8	GNDK		数字地
M9	VCC3318_EMMC		EMMC 3.3 或 1.8v 数字电源
M10	VCC3318_EMMC		EMMC 3.3 或 1.8v 数字电源
M11	VCC3318_EMMC		EMMC 3.3 或 1.8v 数字电源
M12	VCC3318_GPIO1		GPIO1 3.3 或 1.8v 数字电源，供给 GPIO1[0~6]信号
M13	NC		悬空
M14	X_GPIO1_6	IO	GPIO1 通用输入输出 6
M15	X_GPIO0_4	IO	GPIO0 通用输入输出 4
M16	X_GPIO0_1	IO	GPIO0 通用输入输出 1
N			
N1	X_ETH1_TXCTL	O	GMAC1 TX 端控制信号输出
N2	X_ETH1_TXD3	O	GMAC1 TX 端数据输出 3
N3	X_ETH1_TXD0	O	GMAC1 TX 端数据输出 0
N4	NC		悬空
N5	NC		悬空
N6	NC		悬空
N7	NC		悬空
N8	NC		悬空
N9	NC		悬空
N10	NC		悬空
N11	NC		悬空
N12	NC		悬空
N13	NC		悬空
N14	X_GPIO1_8	IO	GPIO1 通用输入输出端口 8
N15	X_GPIO1_9	IO	GPIO1 通用输入输出端口 9
N16	X_GPIO1_7	IO	GPIO1 通用输入输出端口 7
P			
P1	X_ETH1_TXD2	O	GMAC1 TX 端数据输出端口 2
P2	X_ETH1_TCK	O	GAMC1 TX 端参考时钟输出
P3	X_ETH1_RXCTL	I	GMAC1 RX 端控制信号输入
P4	X_ETH1_RXD0	I	GMAC1 RX 端数据输入端口 0
P5	VCC3318_ETH		GMAC1 3.3 或 1.8v 数字电源
P6	X_M0_DS	I	EMMC0 HS400 模式 data strobe 输入
P7	X_M0_CLK	O	EMMC0 接口时钟输出
P8	X_M0_CMD	IO	EMMC0 命令信号输入输出端口



位置	名称	I/O	功能描述
P9	X_M1_CMD	IO	EMMC1 命令信号输入输出端口
P10	X_M1_CLK	O	EMMC1 接口时钟输出
P11	X_M1_DS	I	EMMC1 HS400 模式 data strobe 输入
P12	X_RST_N	I	系统复位输入
P13	X_GPIO1_0	IO	GPIO1 通用输入输出端口 0
P14	X_GPIO1_1	IO	GPIO1 通用输入输出端口 1
P15	X_GPIO1_13	IO	GPIO1 通用输入输出端口 13
P16	X_GPIO1_10	IO	GPIO1 通用输入输出端口 10
R			
R1	X_ETH1_TXD1	O	GMAC1 TX 端数据输出端口 1
R2	X_ETH1_TXD0	O	GMAC1 TX 端数据输出端口 0
R3	X_ETH1_TCK	O	GMAC1 TX 端参考时钟输出
R4	X_M0_D4	IO	EMMC0 数据输入输出端口 4
R5	X_M0_D5	IO	EMMC0 数据输入输出端口 5
R6	X_M0_D6	IO	EMMC0 数据输入输出端口 6
R7	X_M0_D7	IO	EMMC0 数据输入输出端口 7
R8	X_M1_D4	IO	EMMC1 数据输入输出端口 4
R9	X_M1_D5	IO	EMMC1 数据输入输出端口 5
R10	X_M1_D6	IO	EMMC1 数据输入输出端口 6
R11	X_M1_D7	IO	EMMC1 数据输入输出端口 7
R12	X_UART0_TXD	O	UART TX 端数据输出
R13	X_GPIO1_2	IO	GPIO1 通用输入输出端口 2
R14	X_GPIO1_3	IO	GPIO1 通用输入输出端口 3
R15	X_GPIO1_15	IO	GPIO1 通用输入输出端口 15
R16	X_GPIO1_11	IO	GPIO1 通用输入输出端口 11
T1	X_ETH1_RXD3	I	GMAC1 RX 端数据输入端口 3
T2	X_ETH1_RXD2	I	GMAC1 RX 端数据输入端口 2
T3	X_ETH1_RXD1	I	GMAC1 RX 端数据输入端口 1
T4	X_M0_D3	IO	EMMC0 数据输入输出端口 3
T5	X_M0_D0	IO	EMMC0 数据输入输出端口 0
T6	X_M0_D1	IO	EMMC0 数据输入输出端口 1
T7	X_M0_D2	IO	EMMC0 数据输入输出端口 2
T8	X_M1_D3	IO	EMMC1 数据输入输出端口 3
T9	X_M1_D0	IO	EMMC1 数据输入输出端口 0
T10	X_M1_D1	IO	EMMC1 数据输入输出端口 1
T11	X_M1_D2	IO	EMMC1 数据输入输出端口 2
T12	X_UART0_RXD	I	UART0 RX 数据输入端口
T13	X_GPIO1_4	IO	GPIO1 通用输入输出端口 4
T14	X_GPIO1_5	IO	GPIO1 通用输入输出端口 5

位置	名称	I/O	功能描述
T15	X_GPIO1_12	IO	GPIO1 通用输入输出端口 12
T16	X_GPIO1_14	IO	GPIO1 通用输入输出端口 14

2.4 管脚复用

在芯片内部，GPIO0、GPIO1、MUXIO、UART1、QSPI 等模块复用 48 根 IO 线，复用模式如下表所示：

X_TM 是系统测试模式控制信号，0 为功能模式，1 为测试模式；FUNC_MODE 通过 SCU 寄存器，可配置为 0、1、2 三种模式。当 FUNC_MODE 为 2 时，通过配置 MUXIO 模块寄存器实现工作模式选择，可分为 FIFO 从、FIFO 主、SRAM 主、ADMUX 主四种模式。

GPIO1[5:0]由 SCU 寄存器控制，在任何 FUNC_MODE 下均可进行模式选择。

表 2.2 管脚复用表

X_TM	0						1
FUNC_MODE [1:0]	0	1	2				
接口信号	功能模式 0	功能模式 1	FIFO 从	FIFO 主	SRAM 主	ADMUX 模式	
GPIO0[31:0]	GPIO0[31:0]	GPIO0[31:0]	DQ[31:0]	DQ[31:0]	ADQ[31:0]	ADDR[7:0](GPIO0[23:16]) ADQ[15:0](GPIO0[15:0])	测试模式 信号
GPIO1[15]	GPIO1[15]	QSPI_SCK	CLK	CLK	CLK	CLK	
GPIO1[14]	GPIO1[14]	QSPI_CS#	CS#	CS#	CS#	CS#	
GPIO1[13]	GPIO1[13]	QSPI_TX (IO0)	RE#	RE#	RE#	RE#	
GPIO1[12]	GPIO1[12]	QSPI_RX (IO1)	WE#	WE#	WE#	WE#	
GPIO1[11]	GPIO1[11]	QSPI_WP# (IO2)	empty	empty	BE0#	BE0#	
GPIO1[10]	GPIO1[10]	QSPI_HOLD# (IO3)	full	full	BE1#	BE1#	
GPIO1[9]	GPIO1[9]	PWM0	CFIFO_empty	PWM0	BE2#	ADV#	
GPIO1[8]	GPIO1[8]	GPIO1[8]	CFIFO_full	GPIO1[8]	BE3#	CRE	
GPIO1[7]	GPIO1[7]	GPIO1[7]	CFIFO_SEL#	GPIO1[7]	ALE#	WAIT#	
GPIO1[6]	GPIO1[6]	GPIO1[6]	PKGEND#	GPIO1[6]	GPIO1[6]	GPIO1[6]	
GPIO1[5]	GPIO1[5]/SPI_CS	GPIO1[5]/SPI_CS	GPIO1[5]/SPI_CS	GPIO1[5]/SPI_CS	GPIO1[5]/SPI_CS	GPIO1[5]/SPI_CS	
GPIO1[4]	GPIO1[4]/SPI_CK	GPIO1[4]/SPI_CK	GPIO1[4]/SPI_CK	GPIO1[4]/SPI_CK	GPIO1[4]/SPI_CK	GPIO1[4]/SPI_CK	
GPIO1[3]	GPIO1[3]/SPI_TXD	GPIO1[3]/SPI_TXD	GPIO1[3]/SPI_TXD	GPIO1[3]/SPI_TXD	GPIO1[3]/SPI_TXD	GPIO1[3]/SPI_TXD	
GPIO1[2]	GPIO1[2]/SPI_RX D	GPIO1[2]/SPI_RX D	GPIO1[2]/SPI_RX D	GPIO1[2]/SPI_RX D	GPIO1[2]/SPI_RX D	GPIO1[2]/SPI_RXD	
GPIO1[1]	GPIO1[1]/ UART1_TXD	GPIO1[1]/ UART1_TXD	GPIO1[1]/ UART1_TXD	GPIO1[1]/ UART1_TXD	GPIO1[1]/ UART1_TXD	GPIO1[1]/ UART1_TXD	
GPIO1[0]	GPIO1[0]/ UART1_RXD	GPIO1[0]/ UART1_RXD	GPIO1[0]/ UART1_RXD	GPIO1[0]/ UART1_RXD	GPIO1[0]/ UART1_RXD	GPIO1[0]/ UART1_RXD	

2.5 上电时序

为确保芯片内部逻辑与外部器件通讯正常，VCCCK（1.0V）电源和 VCCIO（3.3V）电源对上电时序有以下要求，如下图所示：

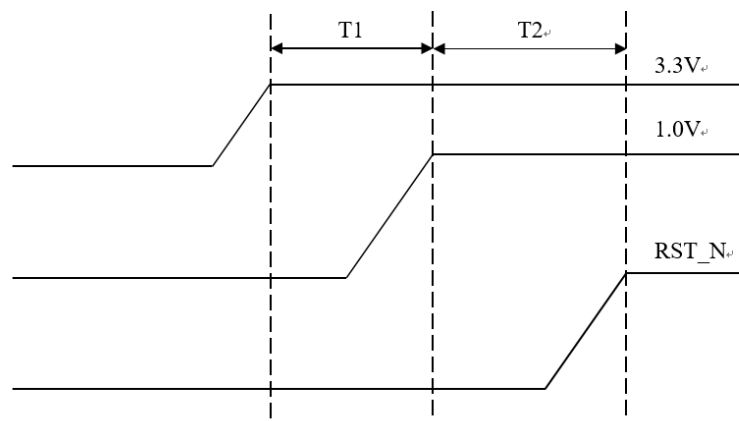


图 2.3 电源上电顺序 1 图

其中， $0 < T1 \leq 10\text{ms}$ 。

T2 为系统复位要求， $40\text{ms} \leq T2$

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	普通 IO 口电源	3.0	3.3	3.6	V
VCCCK	Core 电源	0.9	1.0	1.1	V
VCC10_PLL0	PLL 模拟电源	0.9	1.0	1.1	V
VCC33A_USB	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
VCC3318_ETH	网络接口 IO 数字电源	1.62/3.0	1.8/3.3	1.98/3.6	V
VCC10_DLL_EMMC	EMMC DLL 内核数字电源	0.9	1.0	1.1	V
VCC3318_EMMC	EMMC 接口 IO 数字电源	1.62/3.0	1.8/3.3	1.98/3.6	V
VCC3318_GPIO1	GPIO1 接口 IO 数	1.62/3.0	1.8/3.3	1.98/3.6	V



	字电源				
VCC3318_GPIO0	GPIO0 接口 IO 数字电源	1.62/3.0	1.8/3.3	1.98/3.6	V
VCC33_OSC	系统晶振数字电源	3.0	3.3	3.6	V
VCC10A_SH	SATA HOST PHY 内核模拟电源	0.9	1.0	1.1	V
VCC11A_SH	SATA HOST PHY 内核模拟电源输出	1.0	1.1	1.2	V
VCC33A_SH	SATA HOST PHY IO 模拟电源	3.0	3.3	3.6	V
VCC10A_SD	SATA DEVICE PHY 内核模拟电源	0.9	1.0	1.1	V
VCC11A_SD	SATA DEVICE PHY 内核模拟电源输出	1.0	1.1	1.2	V
VCC33A_SD	SATA DEVICE PHY IO 模拟电源	3.0	3.3	3.6	V
VCC33A_TDC	TDC 模拟电源	3.0	3.3	3.6	V

2.7 功耗

- 静态功耗<0.1W
- 动态功耗<1.5W

2.8 PCB 设计建议

请参考《T6x0 硬件设计用户指南》

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 260Mhz
- 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
- 32KB DTCM
- 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 和 2:1 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

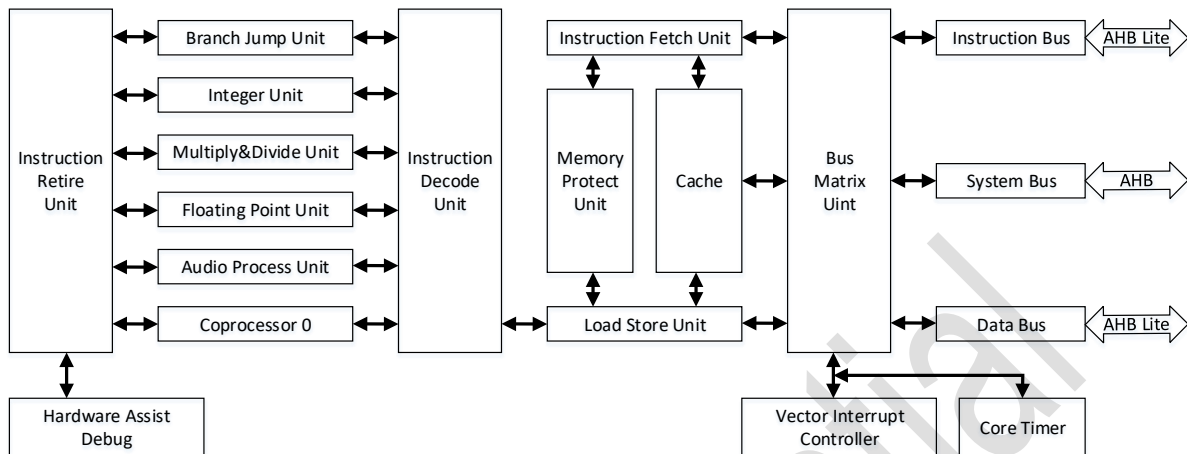


图 3.1 CK803S 系统架构图

*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。

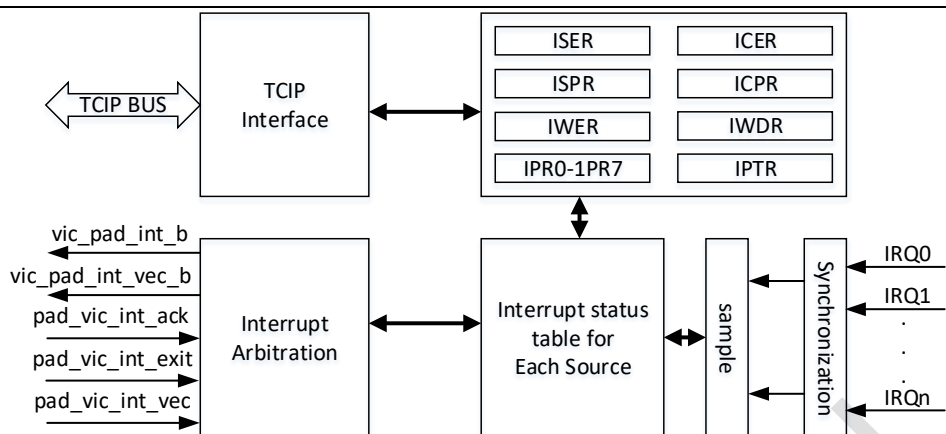


图 3.2 中断控制器结构框图

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：

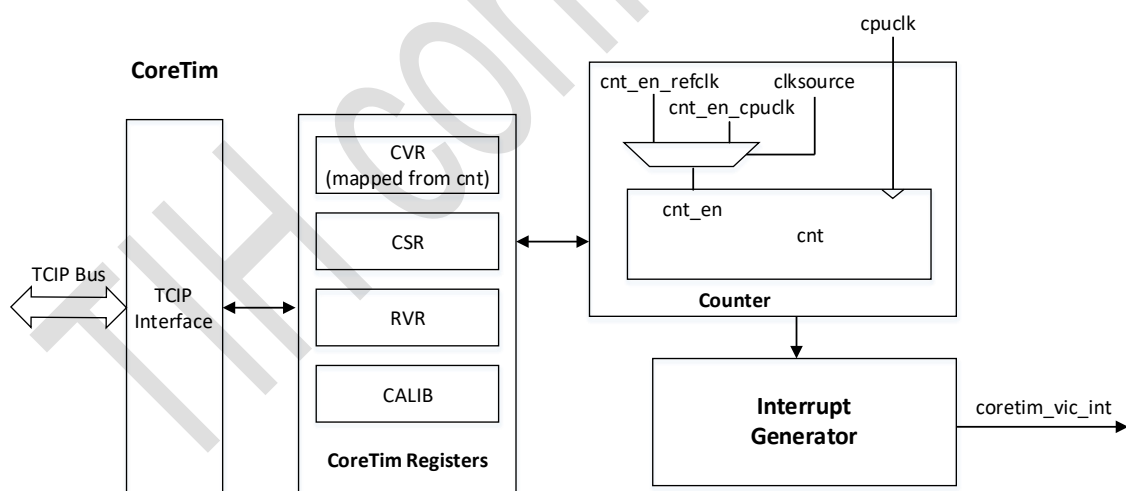


图 3.3 CoreTimer 结构框图

3.2 存储

芯片内部包含 4 块存储单元：ROM、Nor flash 和 2 块 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 8KB SRAM 和 1 片 256KB SRAM，8KB SRAM 用于存储 SATA Host、SATA Device 和 GMAC 等模块的命令链表，用户不能使用该存储空间；256KB SRAM 可供用户使用，用于高速固件代码执行、临时数据存储和算法运算等。8KB SRAM 和 256KB SRAM 各自拥有独立 AHB 和 AXI 接口访问通道，可大幅提升 AHB 和 AXI 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：

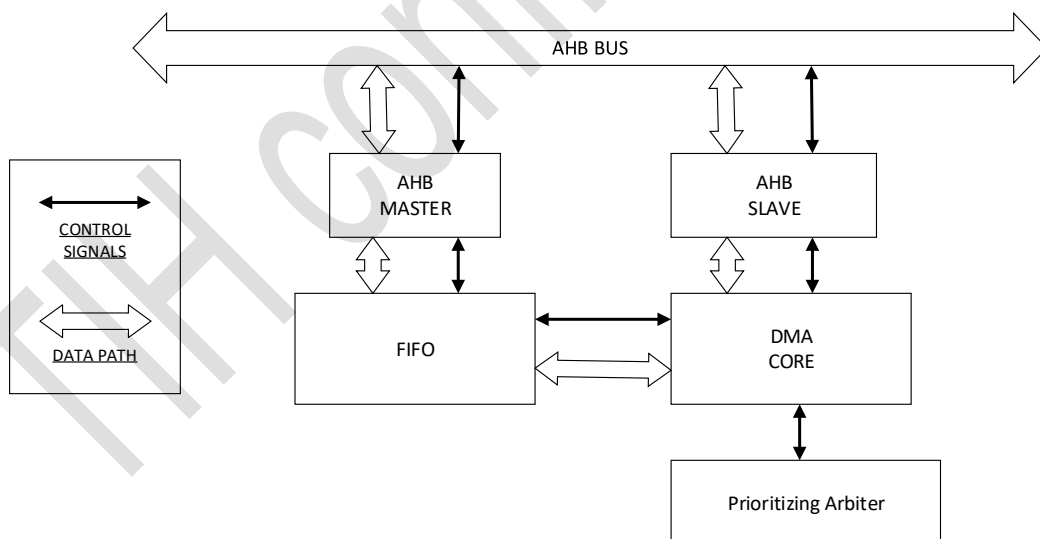


图 3.4 DMA 结构框图

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：

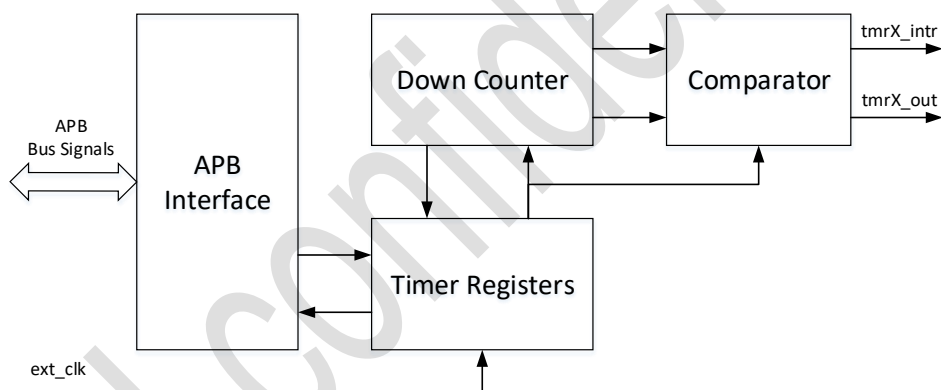


图 3.5 定时器结构框图

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：

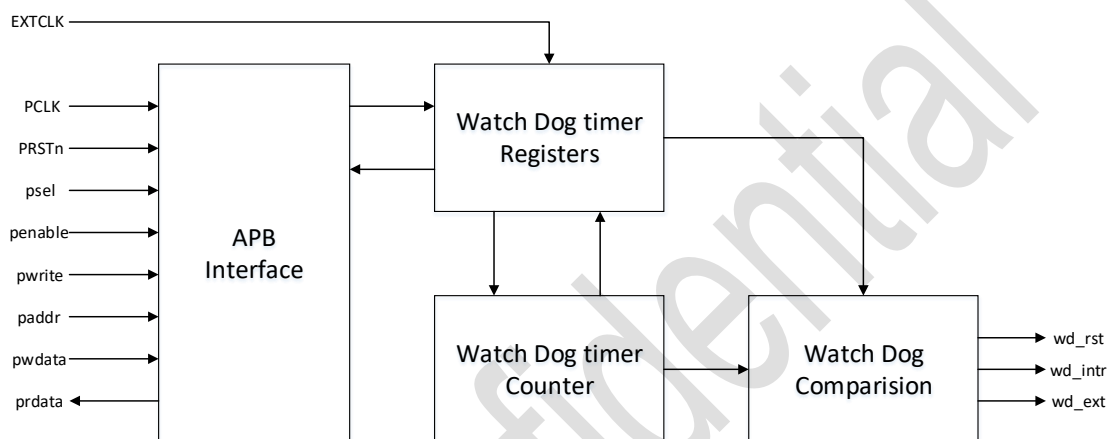


图 3.6 看门狗结构框图

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：

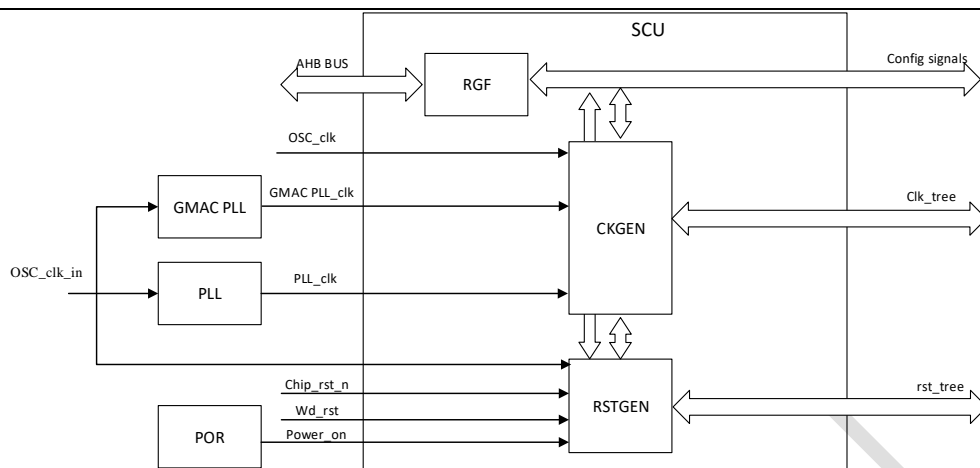


图 3.7 SCU 结构框图

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

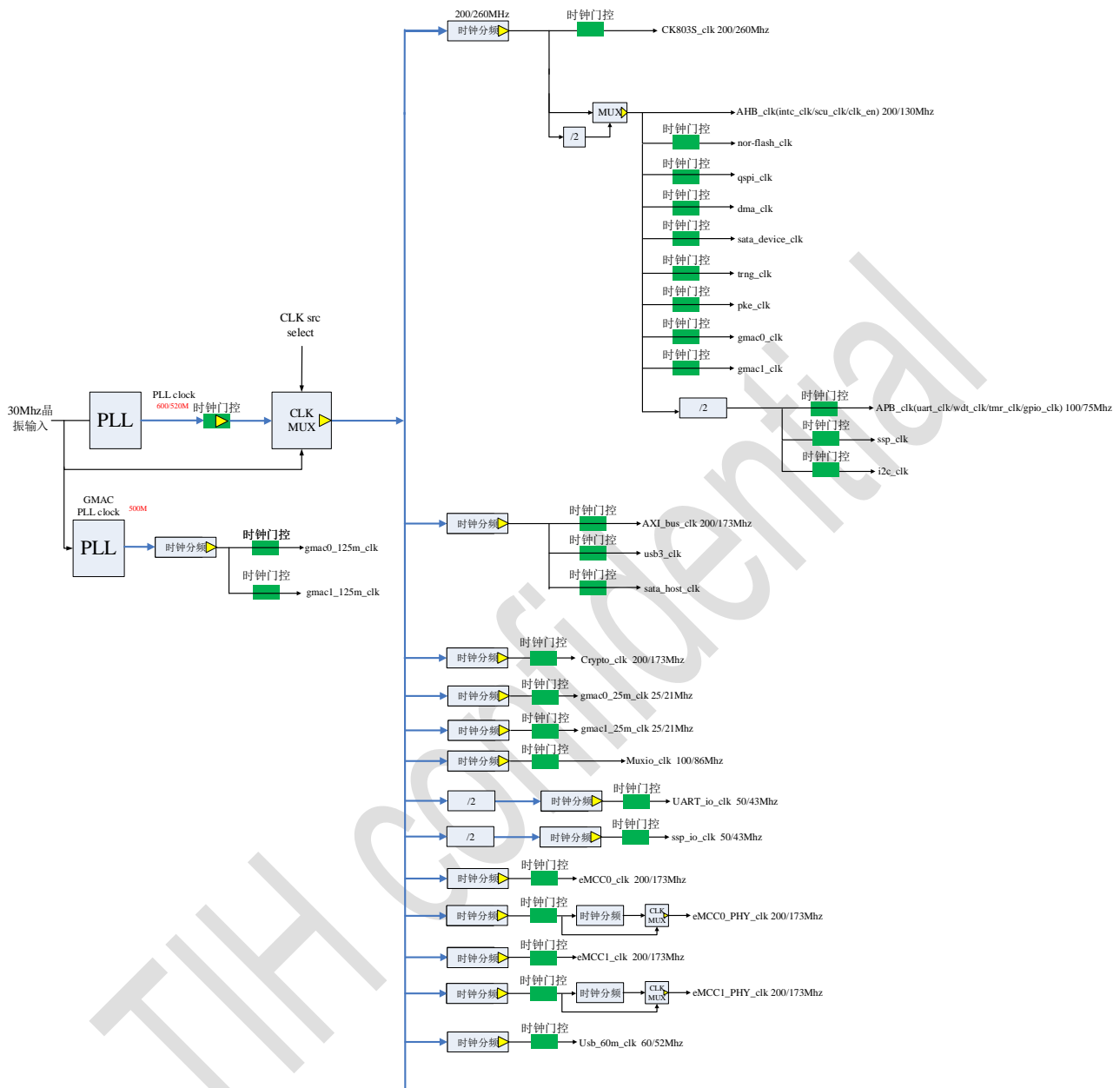


图 3.8 系统时钟树

3.6.4 复位树

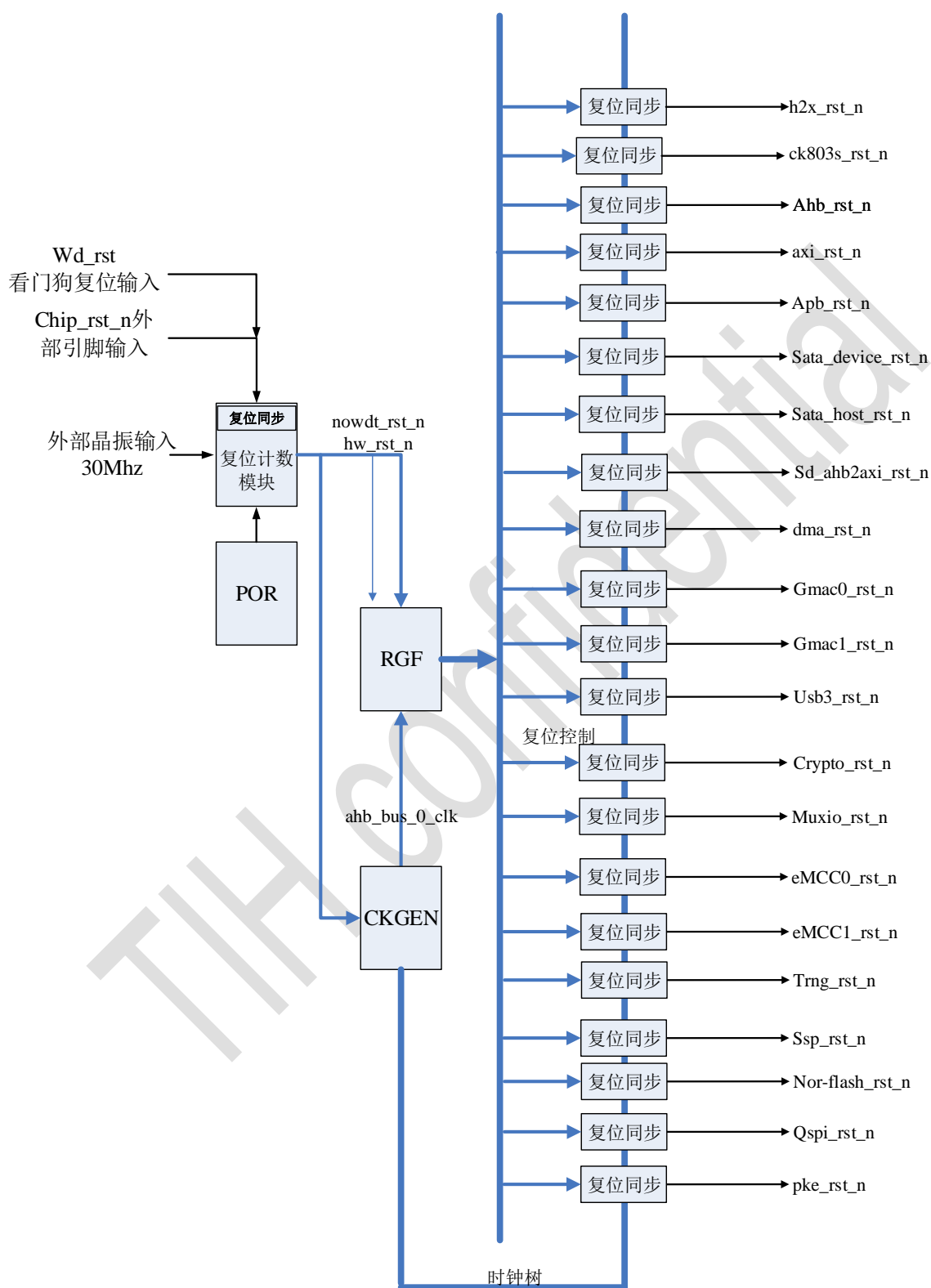


图 3.9 系统复位树

4 安全引擎

4.1 CRYPTO 引擎

4.1.1 模块概述

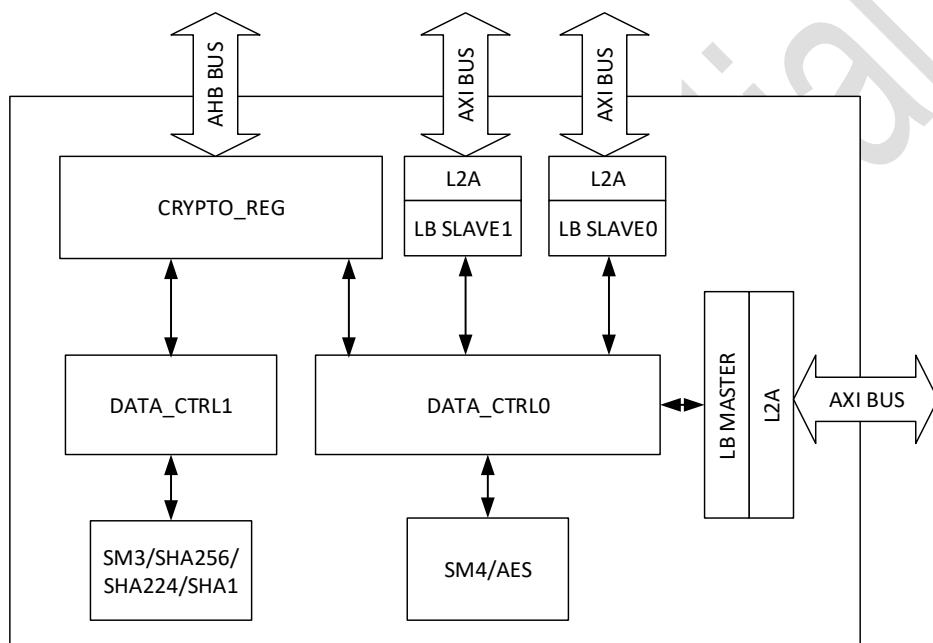


图 4.1 CRYPTO 引擎结构框图

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
- 支持两路 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 $\geq 800\text{MB/s}@200\text{MHz}$ (ECB/CTR/XTS)
- 支持 SM3/SHA1/SHA224/SHA256 $\geq 80\text{MB/s}@200\text{MHz}$

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, 数据从一个 slave 端口写入, 另一个 slave 端口取走 (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, 选择一个 AXI SLAVE 端口或两个 AXI SLAVE 端口 (推荐使用两个, 便于理解)
- 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

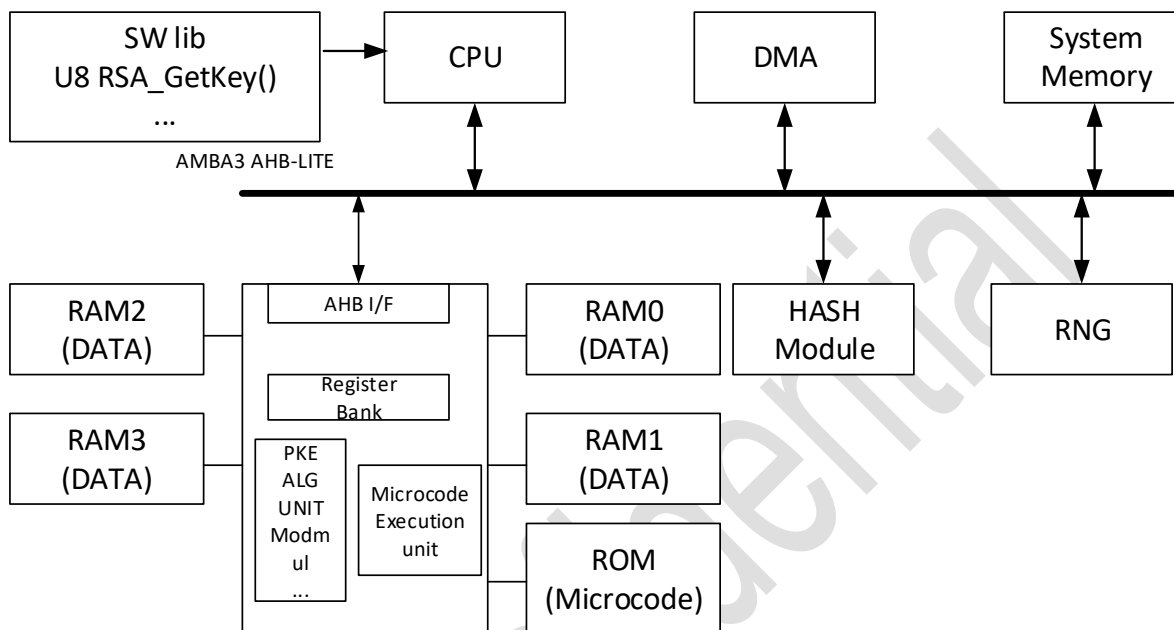


图 4.2 PKE 引擎结构框图

Public Key Engine (PKE) 用来加速公钥密码运算中的大数模运算。公钥密码的运算过程中，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 $\geq 30\text{Mbps}$

TIH Confidential

5 网络接口

5.1 GMAC0 控制器

5.1.1 模块概述

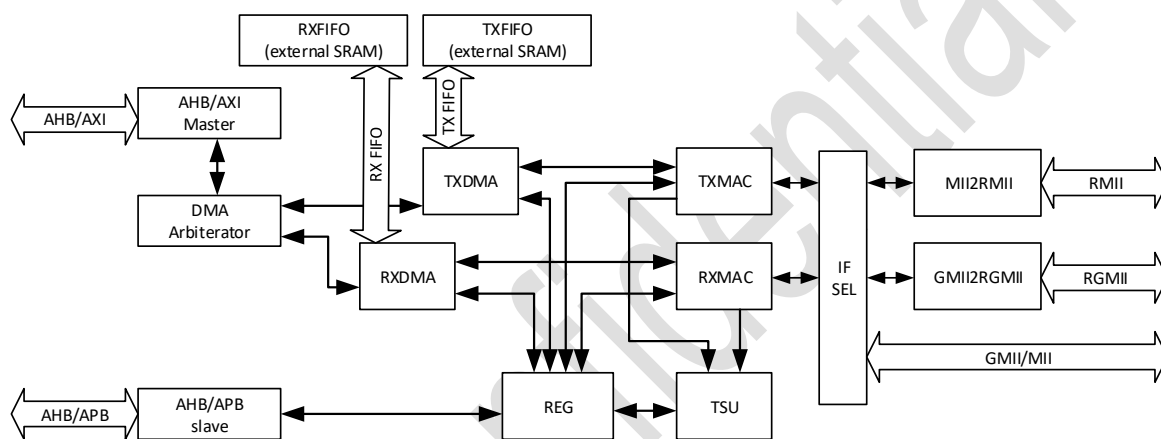


图 5.1 GMAC 控制器结构框图

*注：本芯片中仅支持 RGMII 接口

GMAC 控制器是一个高效能以太网控制器，完全兼容 IEEE 802.3 规范，包括 AHB/AXI 接口、DMA 引擎、数据缓冲区(TXFIFO 和 RXFIFO)、MAC 和接口逻辑等模块。

- 1) DMA 引擎负责系统内存和数据缓冲区之间的所有数据传输，可以减少 CPU 负载，使芯片整体性能更优；
- 2) 数据缓冲区(TXFIFO 和 RXFIFO)用于帧的发送和接收，可降低对系统内存的响应时间要求；
- 3) RGMII 接口基于 HP RGMII 规范 2.0 版本，支持 10Mbps、100Mbps 和 1000Mbps 三种数据速率；
- 4) GMAC 控制器实现了 TCP、UDP、IPv4 和 IPv6 checksum 生成和验证，并支持 VLAN 标记,可有效减少 CPU 运算负载；GMAC 控制器同时支持局域网唤醒功能，支持三个唤醒事件：连接状态更改，魔术包和自定义掩码帧；

5.1.2 模块特性

- 支持 DMA 数据传输
- 支持全双工、半双工操作
- 支持全双工、半双工流控
- 内置 4KB TX FIFO 和 8KB RX FIFO
- 支持最大 4KB 的巨型帧
- 支持 TCP/UDP/IPV4/IPv6 checksum offloads
- 支持 IEEE 802.1Q VLAN 标签植入和移除
- 支持 Wake-On-LAN 功能和三种唤醒方式
- RGMII 接口支持 10Mbps/100Mbps/1000Mbps 传输速率

5.1.3 工作方式

(1) 发送描述符和数据缓冲区

GMAC 控制器使用环形描述符队列来进行发送管理，发送描述符和数据包都存放于系统内存中。GMAC 控制器根据发送描述符内容，将发送数据包从系统内存中转移到 GMAC 控制器的 TX FIFO 中，然后将数据包发送到以太网。

GMAC 控制器支持两个用于发送的描述符环，分别是普通优先级传输环和高优先级传输环，普通优先级传输环用于普通分组传输，高优先级传输环用于高优先级分组传输，具有高优先级的包可以放入高优先级传输环中，以便更快地传输。

(2) 接收描述符和数据缓冲区

GMAC 控制器使用环形描述符队列来进行接收管理，接收描述符和数据接收区都位于系统内存中。GMAC 控制器首先将接收到的数据包存储在 RXFIFO 中，然后根据接收描述符内容，将 RXFIFO 中的数据包移动到系统内存中。

(3) Wake-On-LAN

GMAC 控制器支持三种方式唤醒：

1) 连接状态改变

连接状态改变是指以太网的连接状态发生变化。如果 GMAC 控制器启用了连接状态更改模式，连接状态更改将被视为唤醒事件。

2) 魔术包

一个魔术包包含一个特定序列和 16 个连续的本地 MAC 地址，特定的序列由 6 字节的 0xFF 组成。魔术包的格式如下：

$DA + SA + 6 * (0xFFh) + 16 * (\text{network adaptor's node address}) + \dots$

如果 GMAC 启用了魔术包模式，魔术包会被看做唤醒事件。

3) 自定义掩码帧

GMAC 控制器进入自定义掩码帧模式之前，系统应该将自定义掩码帧列表传递给驱动程序

序，将对应位的字节掩码设置为 1，每个字节掩码决定了将要传入帧的哪些字节应该与相应的自定义掩码帧进行比较，以确定是否为唤醒事件。

此外，自定义掩码帧的数据包长度必须至少为 128 字节。

(4) 流控

GMAC 控制器实现了流控功能，支持 IEEE802.3x 全双工流控模式和半双工流控模式。

全双工模式采用 IEEE802.3x 流控。当 A 和 B 在全双工模式下相互发送和接收数据包时，如果 B 中 RX FIFO 接近满，B 会向 A 发送一个含有暂停时间的帧，以避免数据丢失。然后 A 会在指定时间段内暂停发送数据，B 在指定的时间段内处理已接收数据，暂停时间结束后，A 继续向 B 发送数据包。

半双工操作采用 Back Pressure 方式，当 A 以半双工模式发送和接收数据包时，如果 A 中的 RX FIFO 接近满，A 会发送一个 jam 模式来产生冲突，以避免传入的数据包被保存到 RX FIFO 中，A 需在一段时间内尽快处理接收到的数据，当 RX FIFO 不接近满可以继续接收数据包时 A 不会发送 jam 包。

5.2 GMAC1 控制器

同上，仅基地址不同。

6 USB OTG 接口

6.1 模块概述

USB OTG 接口是通用串行总线双功能设备控制器，支持 USB2.0/USB3.0 及可扩展主机控制器接口（XHCI）协议，可以通过寄存器配置选择来切换不同的功能。模块的功能框图如下：

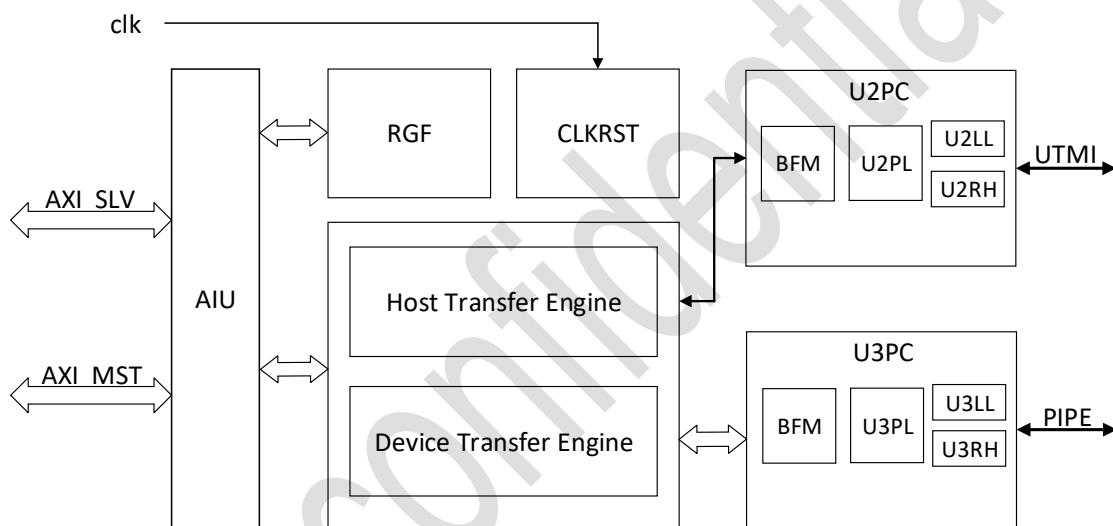


图 6.1 USB 内部结构框图

当 USB OTG 接口作为主机时，控制器根据系统软件准备的数据结构向 USB 设备发出传输请求，控制器支持 XHCI1.0 作为系统软件开发的标准接口。

当 USB OTG 接口作为设备时，控制器响应来自主机的传输请求，内置 9 个端点，端点使用数目可由控制器中的专用缓冲空间配置，端点缓冲空间与系统内存之间的数据传输可以通过内部 DMA 或外部 DMA 完成。

6.2 模块特性

- 支持可扩展主机控制器接口协议 1.0(XHCI1.0)
- 静态角色转换（主机/设备选择）
- 主机和设备模式下支持所有的 USB 传输类型，包括控制/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
- 主机和设备模式都支持 DMA 传输



-
- 主机模式下支持上下文缓存以减少等待时间
 - 设备模式下支持 9 个端点
 - 设备模式下每个端点的 FIFO 深度可配置
 - 设备模式下支持大批量数据流协议

TIH Confidential

7 SATA 接口

7.1 SATA Host 控制器

7.1.1 模块概述

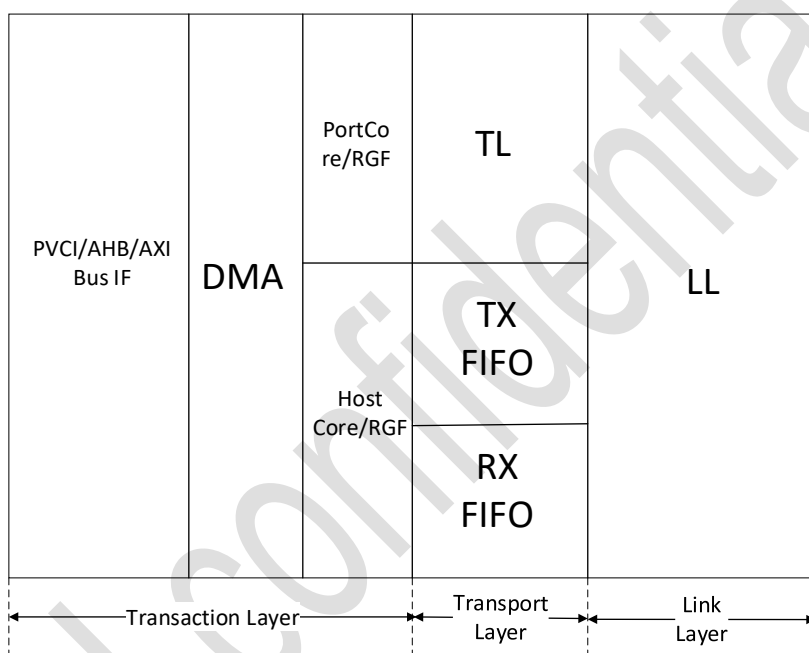


图 7.1 SATA Host 控制器结构框图

SATA host 控制器包含三个协议层：事务层，传输层和链路层。

事务层包含用户总线接口（AHB，AXI，PVCi），DMA 主控，端口/主机寄存器文件，以及端口/主机控制器：

- 1) DMA 控制器负责帧信息结构在系统内存和传输层 TXFIFO/RXFIFO 之间的发送和接收；
- 2) 主机核（Host core）负责全局复位，全局中断以及如 AHCI v1.1 协议中定义的每个端口命令完成合并，无论多少个端口被定义都只有一个主机核；
- 3) 主机寄存器文件（Host register file）是按照 AHCI v1.1 协议实现的，主机寄存器文件会提供实现的特性，比如对 HBA 的控制以及收集所有端口的状态反馈；
- 4) 端口核（Port core）负责处理 HBA 和已连接设备之间的事务。端口核请求 DMA 控制器从设备中获取一条命令或者数据 FIS 并将从设备中获取的 FIS 存储在系统内存中；

- 5) 端口寄存器文件 (Port register file) 与主机寄存器文件都是按照 AHCI v1.1 协议实现, 端口控制器文件负责控制端口并反映端口上的事务和连接状态。

传输层负责构建需要发送的帧信息结构 (FIS) 并解析接收到的帧信息结构, 传输层包含三个主要的部分: FIS 发送器, FIS 接收器和 BIST 控制器模块。

链路层负责发送和接收帧, 发送原语是基于传输层的控制信号, 而接收原语来自物理层并且会转化成控制信号到传输层:

- 1) 数据的发送先经过 CRC 校验, 加扰, 8B/10B 转码, 速度调整, 然后发给物理层;
- 2) 数据的接收则经过数据调整, 10B/8B 转码, 解扰和 CRC 校验然后到传输层。

7.1.2 模块特性

- 符合 Serial ATA Revision 3.0 标准协议
- 符合 AHCI1.1 协议
- 支持数据传输速率 1.5Gbps, 3.0Gbps 和 6.0Gbps
- 支持原生命令队列 (NCQ)
- 支持命令列表覆盖特性
- 支持 PIO 针对多 DRQ 块
- 支持自动局部休眠功能
- 集成 TX FIFO 深度为 256 words
- 集成 RX FIFO 深度为 256 words
- 支持 PHY 数据位宽 20bit 或 40bit 可配

7.2 SATA Device 控制器

7.2.1 模块概述

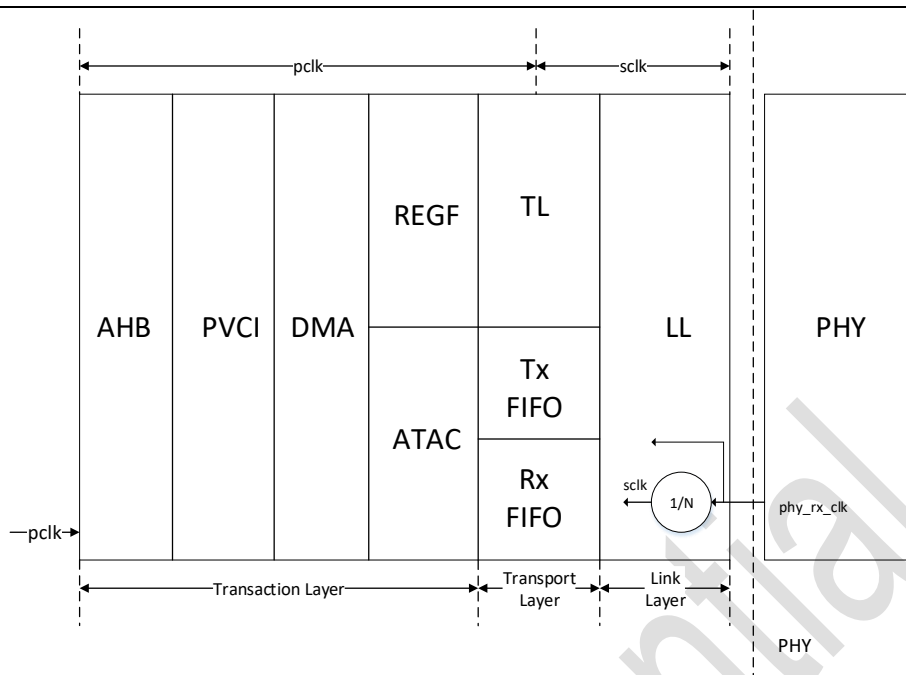


图 7.2 SATA Device 控制器结构框图

SATA device 控制器是一个串行 ATA 从设备控制器，符合 Serial ATA Revision 3.0 标准协议。SATA device 控制器支持 1.5Gbps/3.0Gbps/6.0Gbps 三种数据传输速率，系统总线为 32 位 AHB 总线。SATA device 控制器包含三层：事物层、传输层和链路层，事物层包含 AHB 接口，ATA 控制器，内置 DMA 控制器以及从设备控制寄存器，传输层包含 TX 和 RX FIFO,通过自定义接口与外部 PHY 相连接。

7.2.2 模块特性

- 符合 Serial ATA Revision 3.0 标准协议
- 支持 1.5Gbps,3.0Gbps,6.0Gbps 数据传输速率
- 支持 EXECUTE DEVICE DIAGNOSTIC, Non-data, DMA, PIO, NCQ 命令协议
- 支持 PIO multiple-DRQ 块
- 集成 TX FIFO 深度为 256 words
- 集成 RX FIFO 深度为 256 words
- 支持 PHY 数据位宽 20bit 或 40bit 可配
- 支持 NCQ 命令协议下 FDMA 声明单个 setup FIS
- 支持远端重定时环路
- 支持局部自动睡眠切换

8 存储接口

8.1 eMMC0 控制器

8.1.1 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：

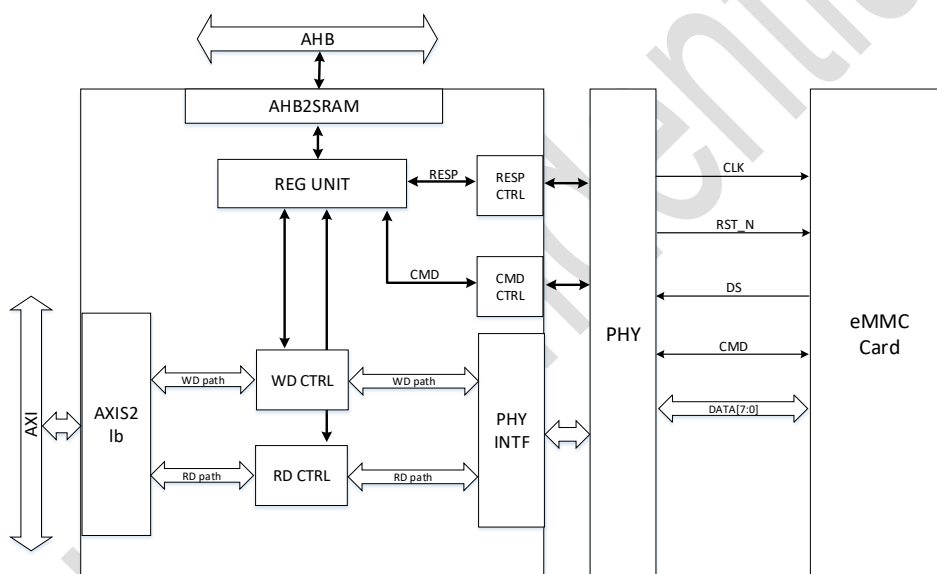


图 8.1 eMMC 控制器结构框图

8.1.2 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

8.1.3 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。

8.2 eMMC1 控制器

eMMC1 控制器与 eMMC0 控制器的内部结构完全相同，仅 eMMC1 和 eMMC0 的基地址不同。

9 外围设备接口

9.1 MUXIO 接口

9.1.1 模块概述

MUXIO 接口是一种多功能接口模块，支持 FIFO slave、FIFO master、SRAM master、ADMUX master 四种工作模式，可外接 FPGA、高速密码芯片、SRAM 颗粒以及 ADMUX 颗粒等外设，用以扩展 T680 芯片的功能。

MUXIO 与 GPIO0 和 GPIO1 模块共用芯片 IO 管脚，通过 SCU 进行模块级选择，通过 MUXIO 内部寄存器进行工作模式的选择。MUXIO 应用框图如下：

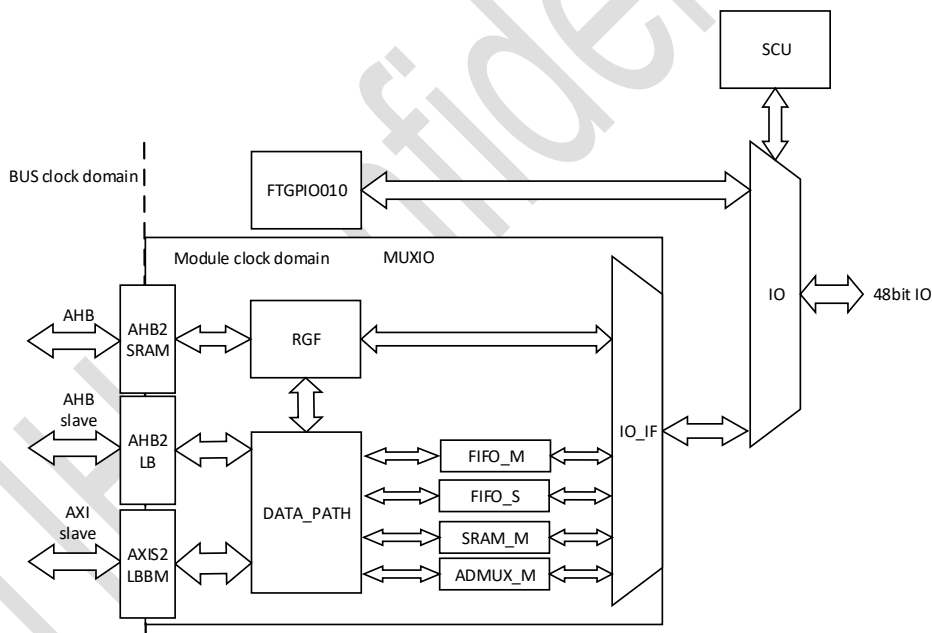


图 9.1 MUXIO 结构框图

9.1.2 模块特性

- 支持一路 AXI 数据通路接口
- 支持一路 AHB CPU 配置接口
- 支持一路 AHB 数据通路接口
- 支持 FIFO 主/FIFO 从/ SRAM 主/ADMUX 主四种模式



- FIFO 从/主/SRAM 主模式时钟最高频率支持 85Mhz
- FIFO ADMUX 主模式时钟最高频率支持 60MHz
- FIFO 从模式特性如下：
 - a) FIFO 从接口端仅支持 16/32 位数据位宽传输
 - b) AXI 端仅支持字对齐读写操作
 - c) FIFO 包含读数据 FIFO 1 个（8KB），写数据 FIFO 1 个（8KB）和命令 FIFO 2 个（32 Words）
 - d) 支持空满信号指示
 - e) 支持半空半满信号中断
 - f) 支持包结束信号
 - g) 支持同步/异步模式
 - h) 支持超时操作中断
- FIFO 主模式特性如下：
 - a) FIFO 主接口端仅支持 16/32 位数据位宽传输
 - b) AXI 端仅支持字对齐读写操作
 - c) 支持空满信号流控
 - d) 只支持同步模式，时序不可配
 - e) 支持超时操作中断
- SRAM 主特性如下：
 - a) 支持 8/16/32 位数据位宽传输
 - b) 支持高低字节数据使能
 - c) 支持 AXI 端 32 位操作自动拆解功能
 - d) 不支持 AXI 端 FIXED 类型传输
 - e) 支持同步/异步模式
 - f) 异步模式支持地址锁存扩展功能，最大寻址范围 64Mb
 - g) 控制信号时序参数可配
- ADMUX master 特性如下：
 - a) 支持 16 位数据位宽传输
 - b) 支持高低字节数据使能
 - c) 支持 AHB/AXI 端 32 位操作自动拆解功能
 - d) 不支持 AHB/AXI 端 FIXED 类型传输
 - e) 支持同步/异步模式
 - f) 支持低 16 位地址数据复用，最大寻址范围 128Mb
 - g) 异步模式控制信号时序参数不可配
 - h) 异步模式下只支持 single 传输，不支持 burst 传输
 - i) Burst mode 只支持 continuous，不支持 fixed length

9.1.3 工作方式

9.1.3.1 FIFO slave 模式

FIFO slave 应用框图如下：

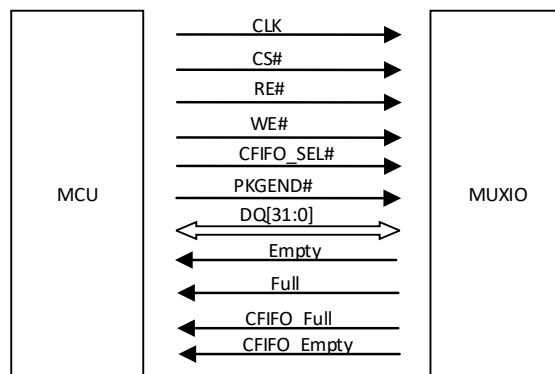


图 9.2 FIFO Slave 应用接口

FIFO 从接口支持同步模式和异步模式。

同步模式下时序如下：

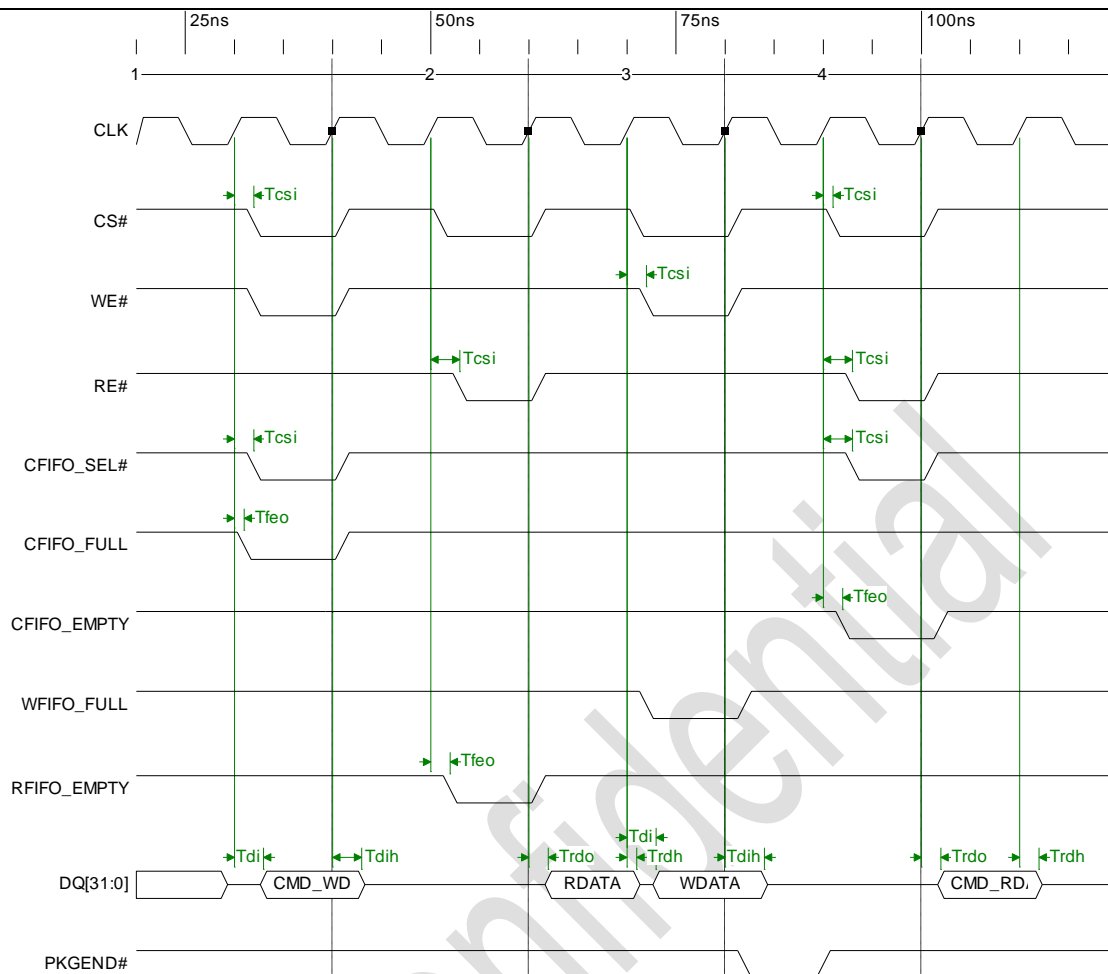


图 9.3 FIFO Slave 同步模式时序图

异步模式下时序如下：

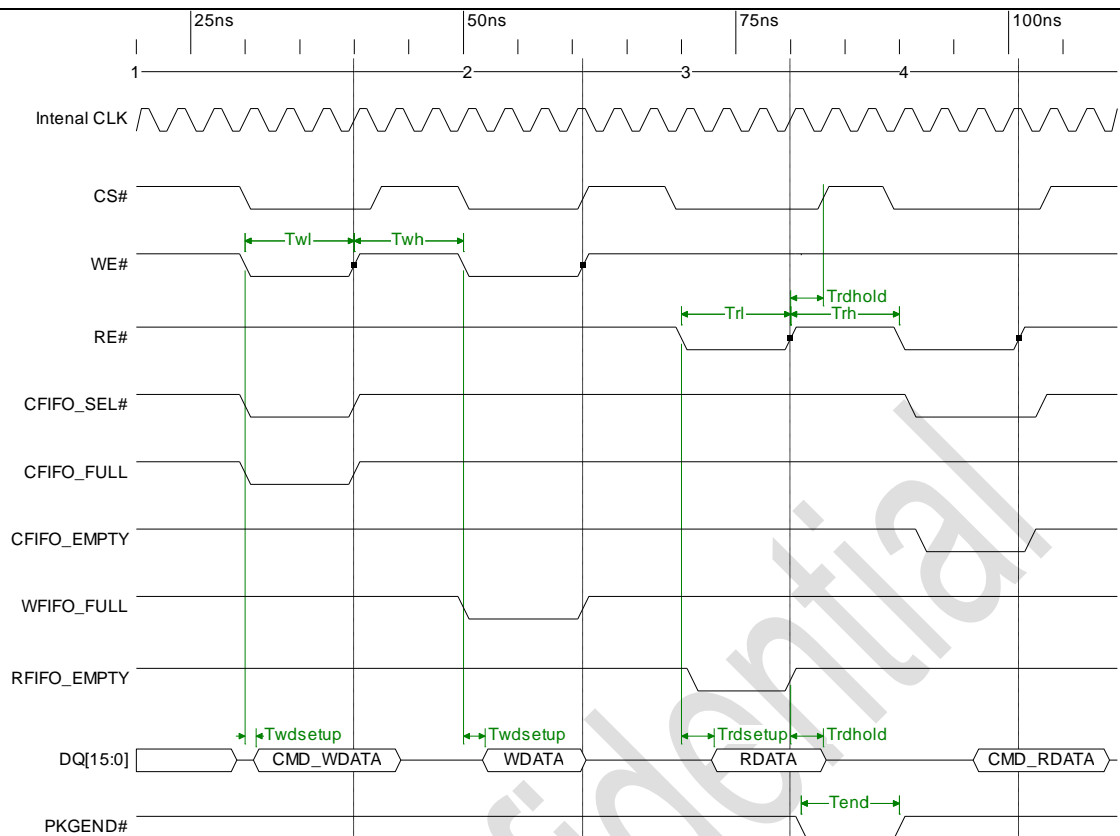


图 9.4 FIFO Slave 异步模式时序图

主机时序参数:

$Twl \geq 3$, $Twh \geq 3$, $Trl \geq 3$, $Trh \geq 3$, $Twsetup \leq 1$, $Tend \geq 3$ 。

FIFO slave 时序参数:

$Trdhold = 0/1$, $Trdsetup \leq 1$

长包传输:

向 PC 端传输数据时, 外挂主机可先传输固定长度数据包。在包结束后, 外挂主机通过 PKGEND# (此时 CFIFO_SEL# 为 1) 产生数据包结束中断通知 CPU。

短包传输:

外挂主机完成短包数据传输后, 命令传输通知 CPU。

命令传输:

外挂主机可通过 CFIFO_SEL# 有效选择命令 FIFO 读写。当写完命令后, 通过 CFIFO_SEL# (此时 CFIFO_SEL# 为 0) 和 PKGEND# 产生命令包结束中断通知 CPU。

9.1.3.2 FIFO master 模式

FIFO 主应用框图如下:

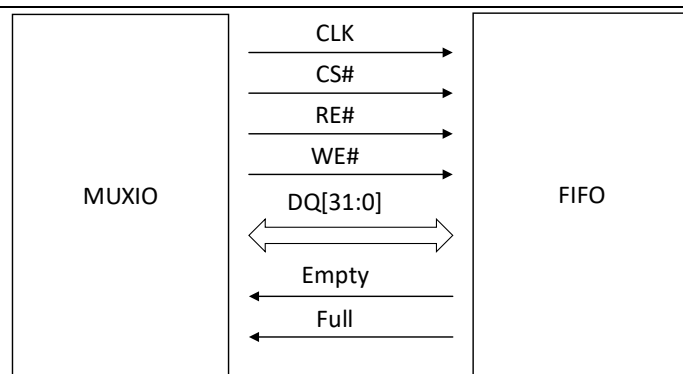


图 9.5 FIFO Master 应用接口

FIFO 主接口只支持同步模式，Timing 如下：

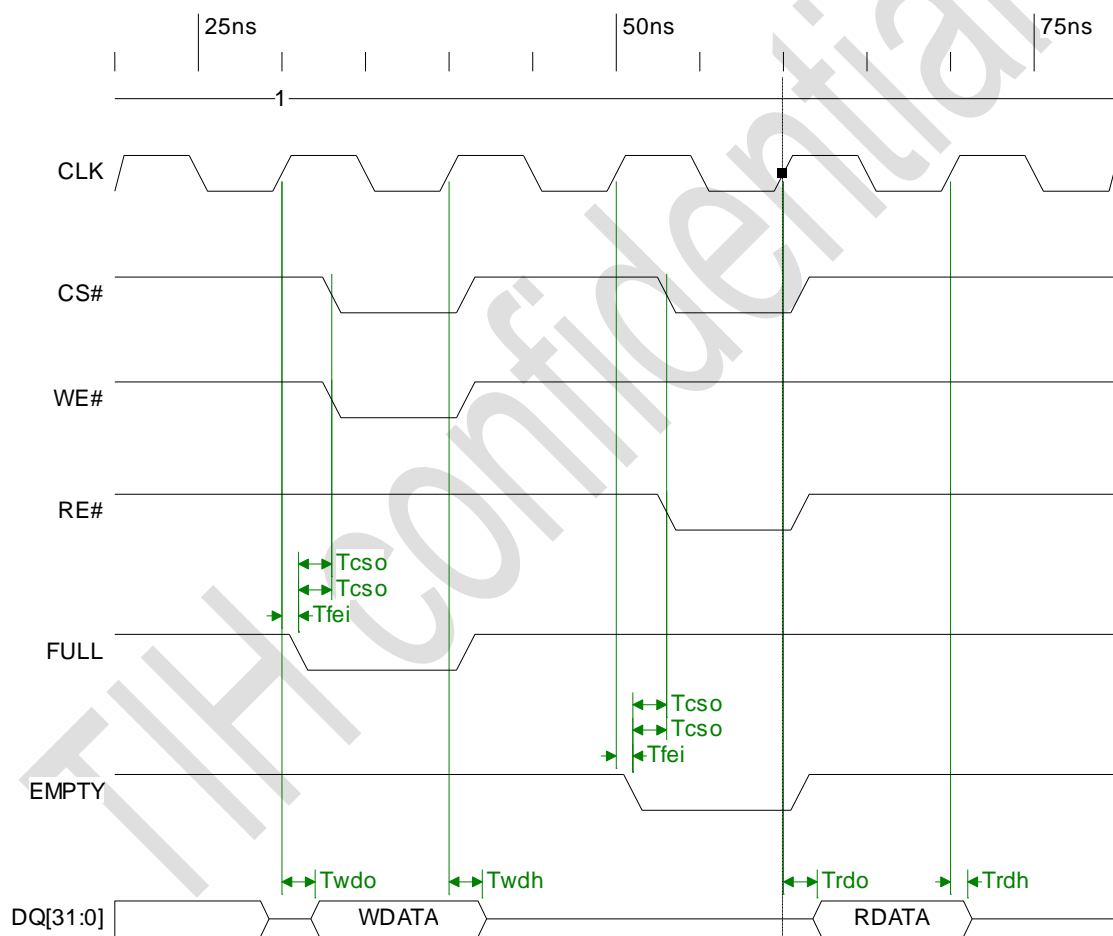


图 9.6 FIFO Master 时序图

该模块中有超时机制，FIFO master 模式下，发生写超时或者读超时，则上报中断给 CPU。

9.1.3.3 SRAM master 模式

SRAM master 应用框图如下：

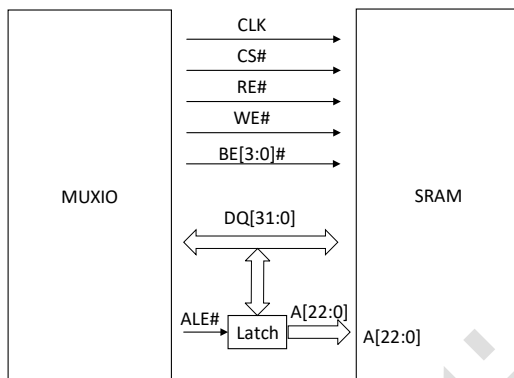


图 9.7 SRAM Master 应用接口

时序图如下：

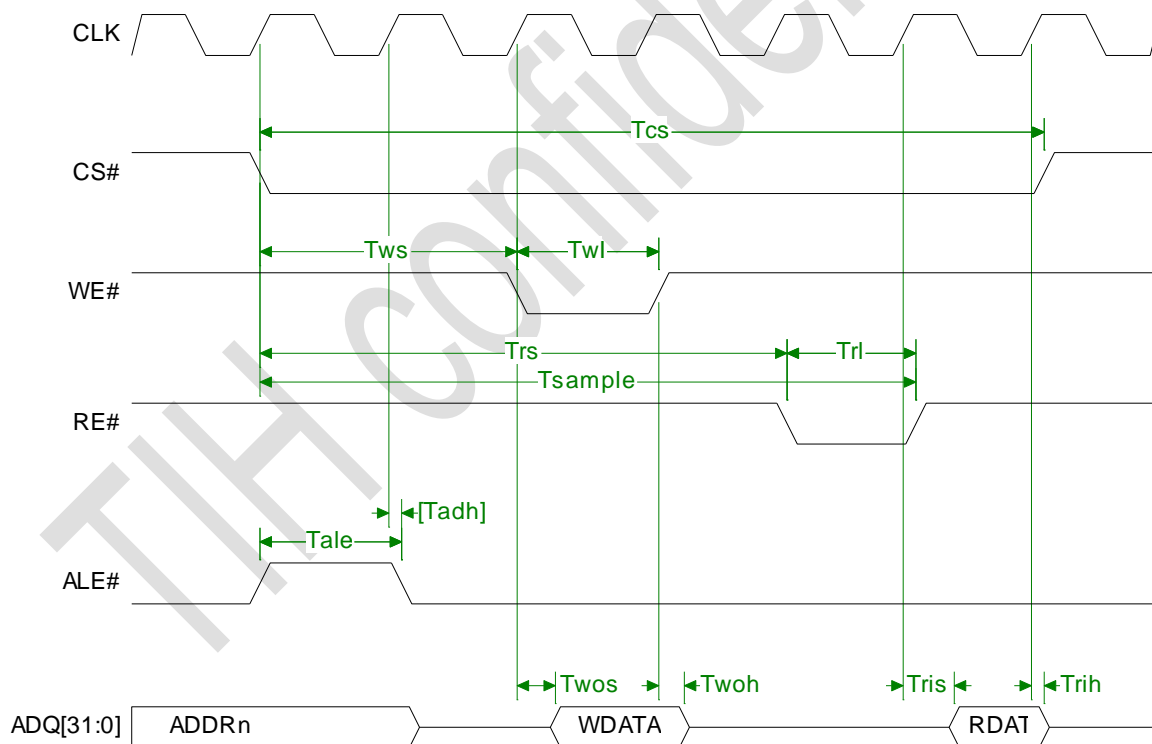


图 9.8 SRAM Master 时序图

上图中 $T_{cs} \geq (T_{ws} + T_{wl}) / (T_{rs} + T_{rl}) + 1$ 。

$T_{ws} / T_{rs} \geq T_{ale}$ 。

$T_{adh} \geq 1ns$ 。

$T_{woh} \geq 1ns$

9.1.3.4 ADMUX master 模式

ADMUX master 应用框图如下：

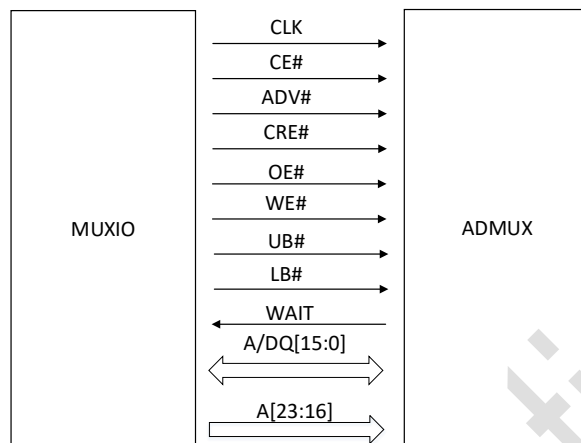


图 9.9 ADMUX Master 应用接口

支持同步异步数据传输，同步模式下支持 burst 数据传输，但只支持 continuous，不支持 fixed-length 模式传输。

异步模式读时序图如下：

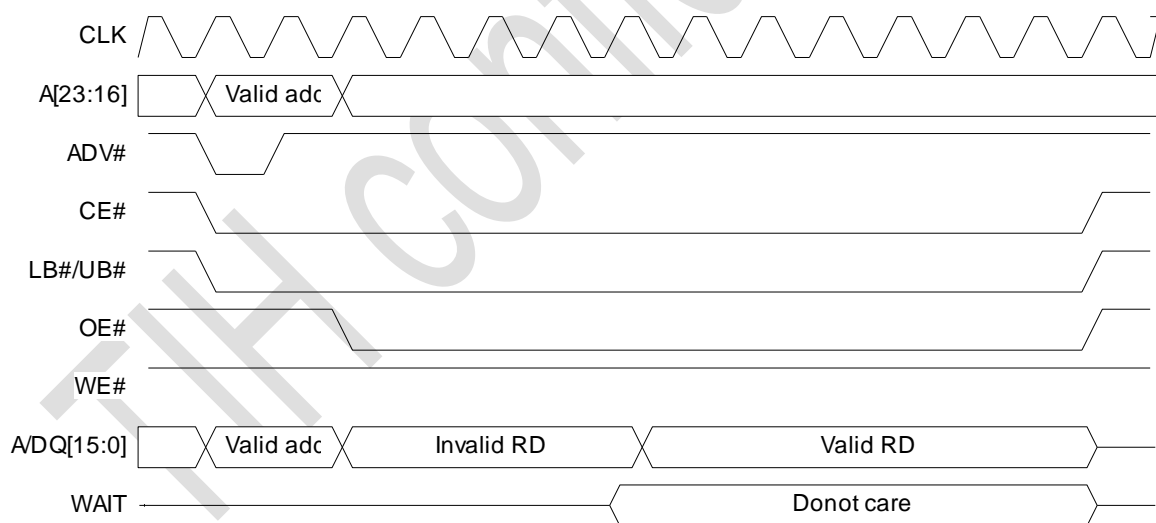


图 9.10 ADMUX 异步模式读时序图

异步模式写时序图如下：

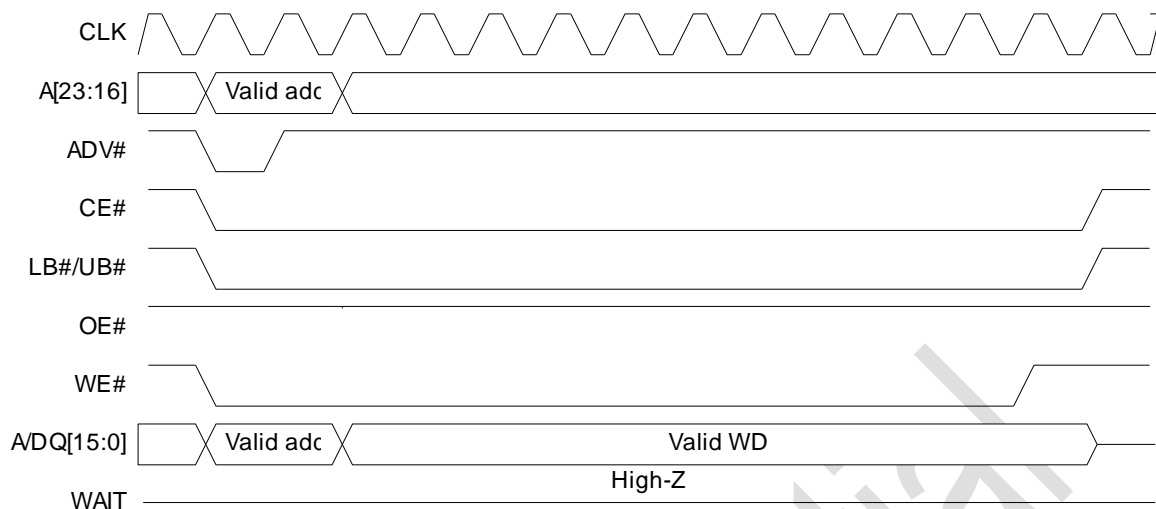


图 9.11 ADMUX 异步模式写时序图

同步 burst continuous 模式读时序图如下：

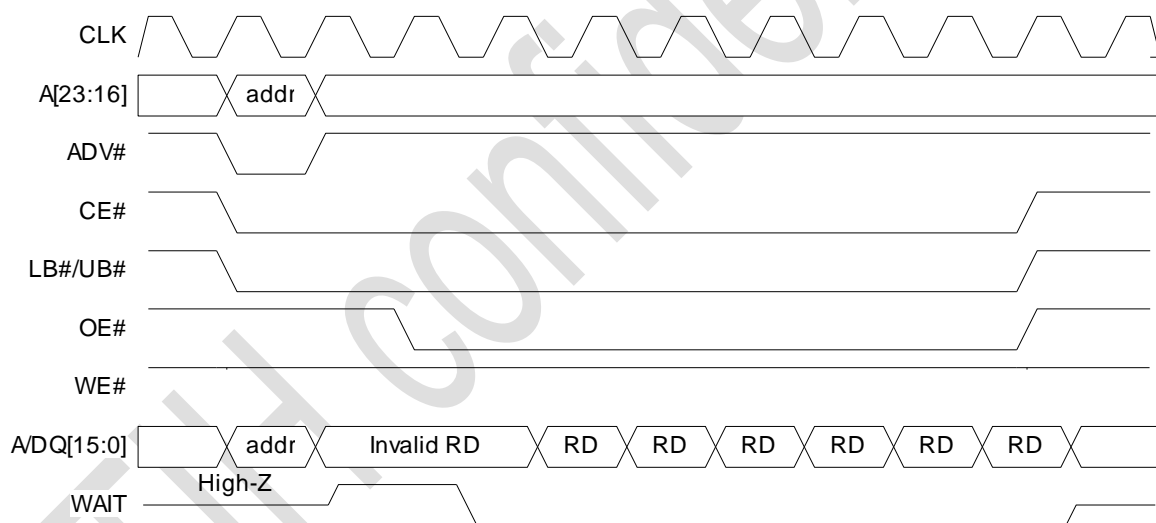


图 9.12 ADMUX 同步模式读时序图

同步 burst continuous 模式写时序图如下：

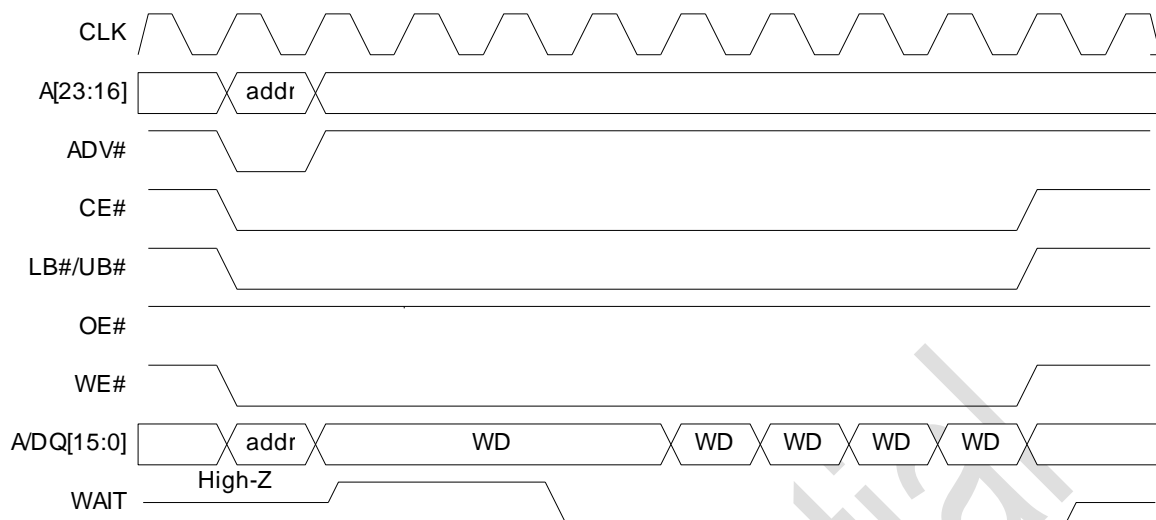


图 9.13 ADMUX 同步模式写时序图

异步模式的时序参数不可配，为保守值，具体时序要求见上面时序图。

同步 burst 模式下 WAIT 为低数据有效传输，device 端 WAIT 采用默认配置，BCR[8]=1。

9.2 I2C 控制器

9.2.1 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：

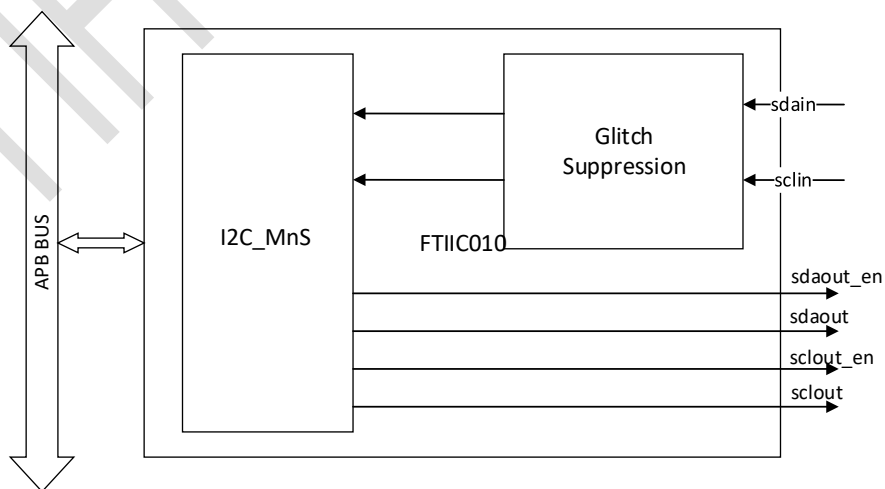


图 9.14 I2C 控制器结构框图

9.2.2 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

9.2.3 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

9.3 QSPI 控制器

9.3.1 模块概述

QSPI 控制器主要用于外扩 SPI SRAM、SPI flash 外设等。
QSPI 控制器架构如下：

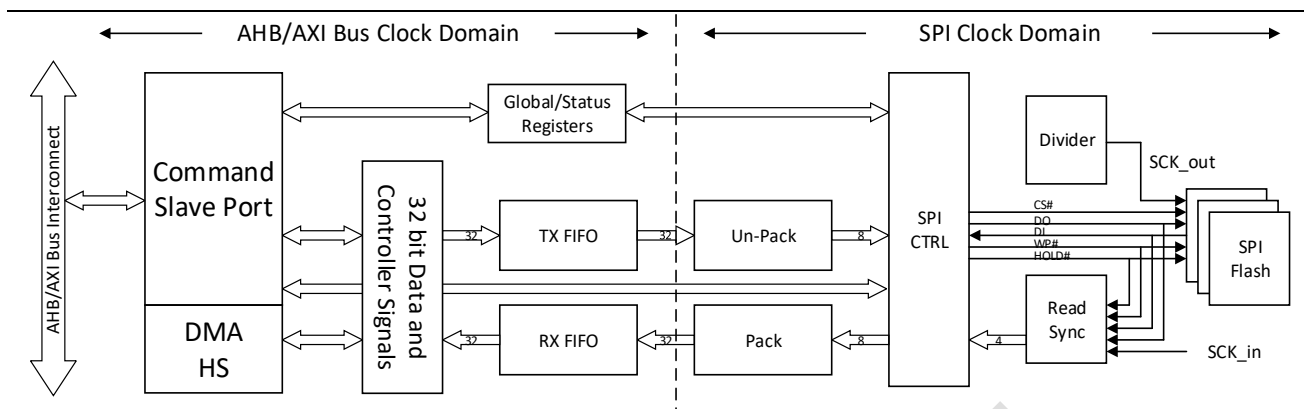


图 9.15 QSPI 控制器结构框图

9.3.2 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

9.4 SPI 控制器

9.4.1 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：

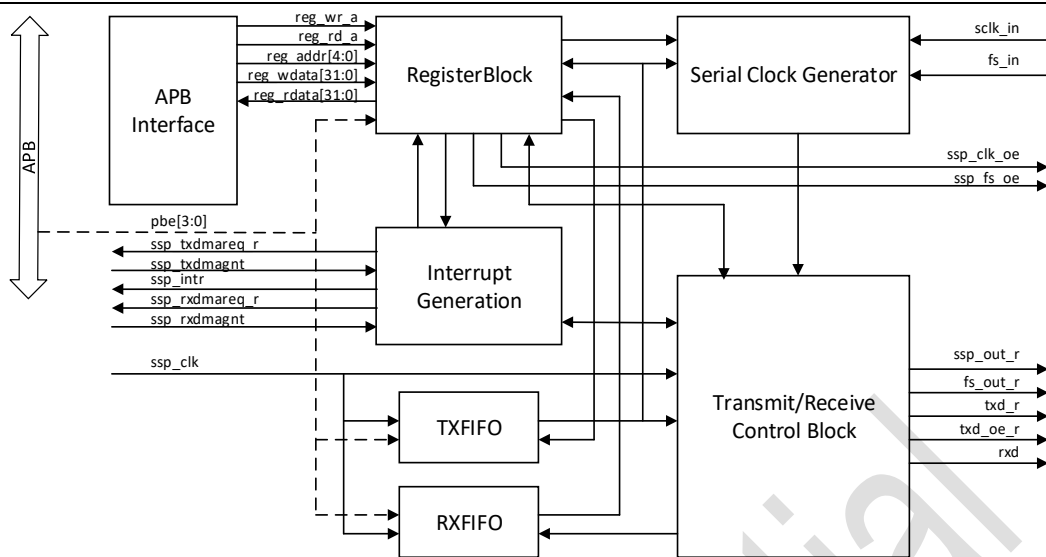


图 9.16 SPI 控制器结构框图

9.4.2 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

9.5 UART0 控制器

9.5.1 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：

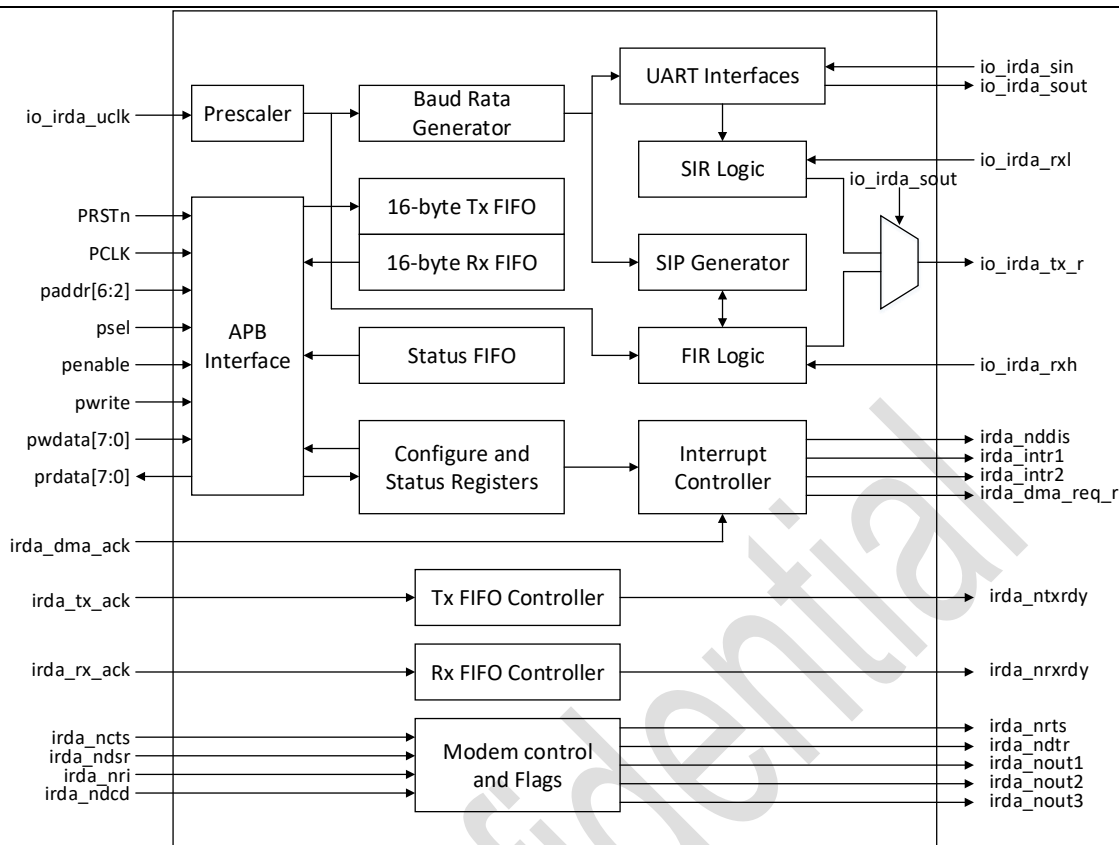


图 9.17 UART 控制器结构框图

*注：上图中 Irda 功能本芯片中未支持

9.5.2 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

9.6 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。

9.7 GPIO0 控制器

9.7.1 模块描述

GPIO0 提供 32 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。

GPIO0 的 32 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。

GPIO0 模块结构图如下：

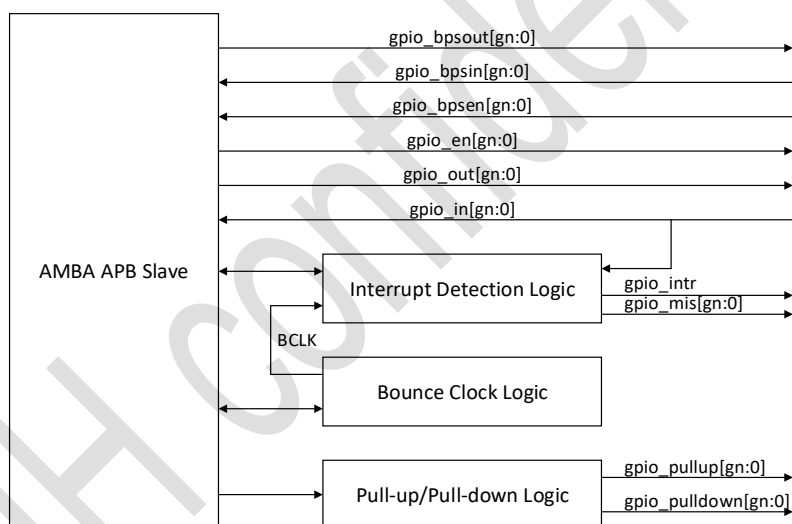


图 9.18 GPIO 控制器结构框图

9.7.2 模块特性

- 32 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

9.8 GPIO1 控制器

GPIO1 控制器与 GPIO0 控制器内部结构完全相同，除基地址不同外，GPIO1 只有低 16 位管脚有效，高 16 位管脚无效。

TIH Confidential

10 安全特性

10.1 电压检测

10.1.1 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：

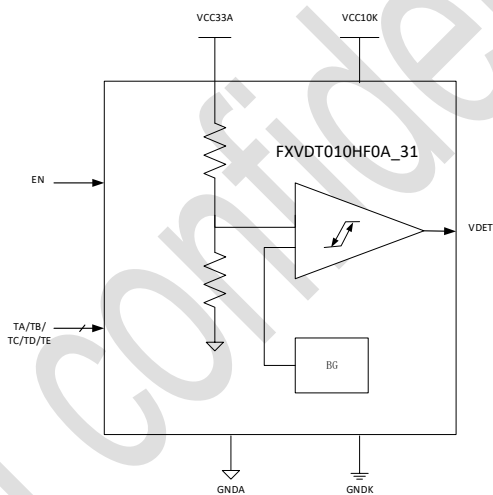


图 10.1 电压检测模块框图

10.1.2 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

10.2 温度检测

10.2.1 模块概述

温度检测 TDC 是一个高速温度转数字信号模块，可以帮助 CPU 进行实时温度监控及报警，当芯片处于极端环境或者瞬时温差较大的情况下，CPU 可以调整各模块配置让芯片进入更加安全的工作模式，以保持芯片工作的稳定性，同时也可以防止外部温度环境的攻击。

TDC 模块结构如下：

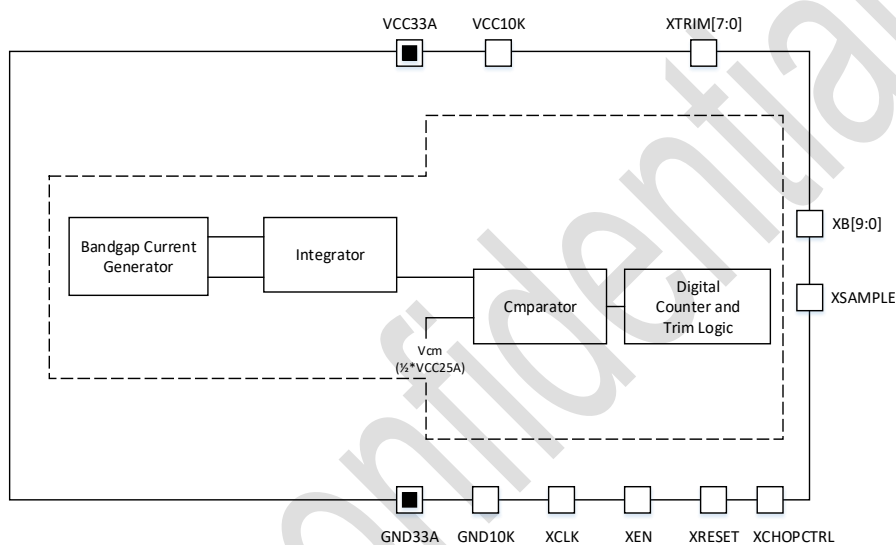


图 10.2 温度检测模块结构框图

10.2.2 模块特性

- 工作温度范围：-40~125℃
- 转换精度 10bit
- 转换速率 1.0KSPS
- 支持低功耗模式

10.2.3 模块时序

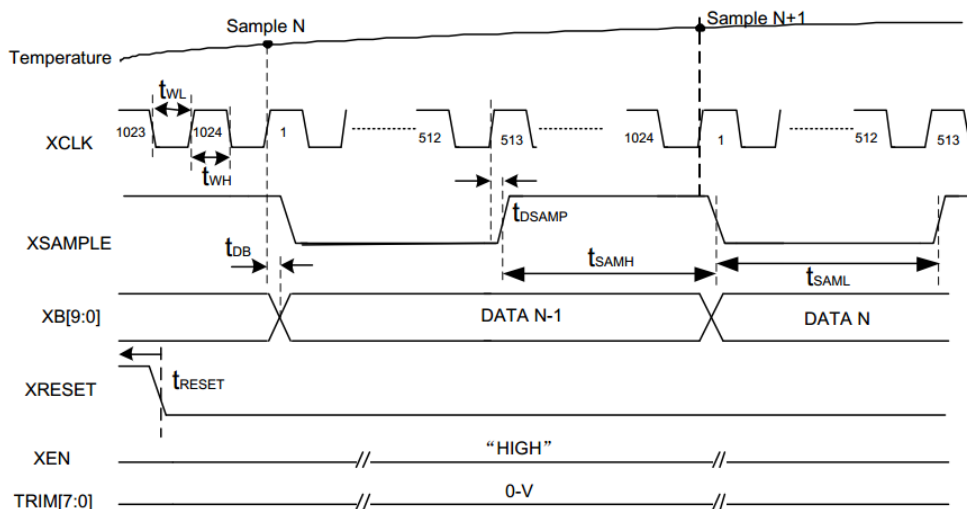


图 10.3 TDC 工作时序图

10.3 物理探测防护

10.3.1 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。

10.3.2 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。

10.4 芯片 ID

10.4.1 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。



10.4.2 模块特性

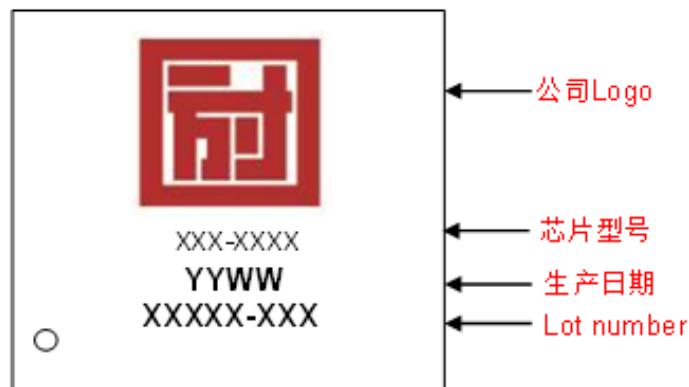
- 有效数据位宽 64bits
- 可支持出厂烧写和用户烧写 2 种模式
- 用户可自定义烧写内容
- 支持低功耗模式

TIH Confidential

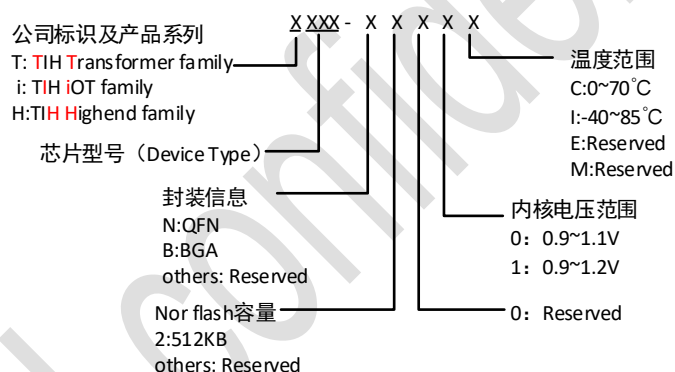
订购信息

芯片名称	flash 容量	封装信息	温度范围	Package Qty
T680-B200C	512KB	BGA220	0~70°C	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

