		每个中断源都有一个寄存器位。	

12.5.9.保护使能寄存器，VICPROTECTION

寄存器	地址	读/写	描述	复位值
VIC0PROTECTION	0x7120_0020	读/写	保护使能寄存器(VIC0)	0x0000_0000
VIC1PROTECTION	0x7130_0020	读/写	保护使能寄存器(VIC1)	0x0000_0000

名称	位	描述	复位值
Reserved	[31:1]	保留，作为 0 读取，不要修改	0x0
IntEnable	[0]	使能或禁止保护寄存器访问： 0=保护模式禁止（复位） 1=保护模式使能 当保护模式使能时，只有特权模式可以访问（进行读和写）中断控制寄存器。当保护模式禁止时，用户模式和特权模式都可以访问寄存器。 当保护模式禁止时，这个寄存器只能在特权模式下被访问。	0

12.5.10 软件优先级屏蔽寄存器，VICSWPRIORITYMASK

寄存器	地址	读/写	描述	复位值
VIC0SWPRIORITYMASK	0x7120_0024	读/写	软件优先级屏蔽寄存器(VIC0)	0x0000_FFFF
VIC1SWPRIORITYMASK	0x7130_0024	读/写	软件优先级屏蔽寄存器(VIC1)	0x0000_FFFF

名称	位	描述	复位值
Reserved	[31:16]	保留，作为 0 读取，不要修改	0x0
SWPriorityMask	[15:0]	控制 16 位中断信号优先级软件屏蔽	0xFFFF

		0=中断优先级被屏蔽 1=中断优先级未被屏蔽 寄存器的位于 16 位中断优先级相适应。	

12.5.11.菊花链矢量优先寄存器

寄存器	地址	读/写	描述	复位值
VIC0PRIORITYDAISY	0x7120_0028	读/写	菊花链矢量优先寄存器(VIC0)	0x0000_000F
VIC1PRIORITYDAISY	0x7130_0028	读/写	菊花链矢量优先寄存器(VIC1)	0x0000_000F

名称	位	描述	复位值
Reserved	[31:16]	保留，作为 0 读取，不要修改	0x0
SWPriorityMask	[15:0]	选择矢量中断优先级。可以选择 16 进制数 0~15 范围内的任何一个矢量中断优先级值运行寄存器。	0xF

12.5.12. 矢量地址寄存器，VICVECTADDR[0-31]

寄存器	地址	读/写	描述	复位值
VIC0VECTADDR[31:0]	0x7120_0100 ~0x7120_017C	读/写	矢量地址[31:0]寄存器(VIC0)	0x0000_0000
VIC1 VECTADDR[31:0]	0x7130_0100 ~0x7130_017C	读/写	矢量地址[31:0]寄存器(VIC1)	0x0000_0000

名称	位	描述	复位值
VectorAddr	[31:0]	包含 ISR 矢量地址	0x0000_0000

12.5.13.矢量优先寄存器，VICVECTRPRIORITY[0-31]

寄存器	地址	读/写	描述	复位值
VIC0VECTPRIORITY[31:0]	0x7120_0200 ~0x7120_027C	读/写	矢量优先[31:0]寄存器(VIC0)	0x0000_000F
VIC1 VECTPRIORITY[31:0]	0x7130_0200 ~0x7130_027C	读/写	矢量优先[31:0]寄存器(VIC1)	0x0000_000F

名称	位	描述	复位值
Reserved	[31:4]	保留，作为 0 读取，不要修改	0x0
VectorAddr	[3:0]	选择矢量中断优先级。可以选择 16 进制数 0~15 范围内的任何一个矢量中断优先级值运行寄存器。	0x0000_0000

12.5.14.矢量地址寄存器，VICADDRESS

寄存器	地址	读/写	描述	复位值
VIC0ADDRESS	0x7120_0F00	读/写	矢量地址寄存器(VIC0)	0x0000_0000
VIC1ADDRESS	0x7130_0F00	读/写	矢量地址寄存器(VIC1)	0x0000_0000

名称	位	描述	复位值
VectAddr	[31:0]	包含当前激活的 ISR 地址，复位值是 0x00000000 寄存器的读取操作可以返回 ISR 的地址，设置当前中断处于正在服务状态。只有当有激活中断的时候可以进行读操作。 向寄存器写入任何值都可以清除当前中断。只有在终端服务快要结束的时候才可以进行写入操作。	0x0

13 安全子系统

13.1 概述

安全子系统是加密功能的加速器。安全子系统架构提供高速数据处理功能，有双层的 AHB 总线和 FIFOs。可以对 FIFO-Rx 和 FIFO-Tx 进行编程，监测 AES 或 DES/3D3ES 或 SHA-1/PRNG 模块。FIFO-Rx 和 FIFO-Tx 从目标模块内自动转换输入输出数据。安全子系统不需要 CPU 便可以完成高速数据处理。

图 13-1 为 安全子系统模块图

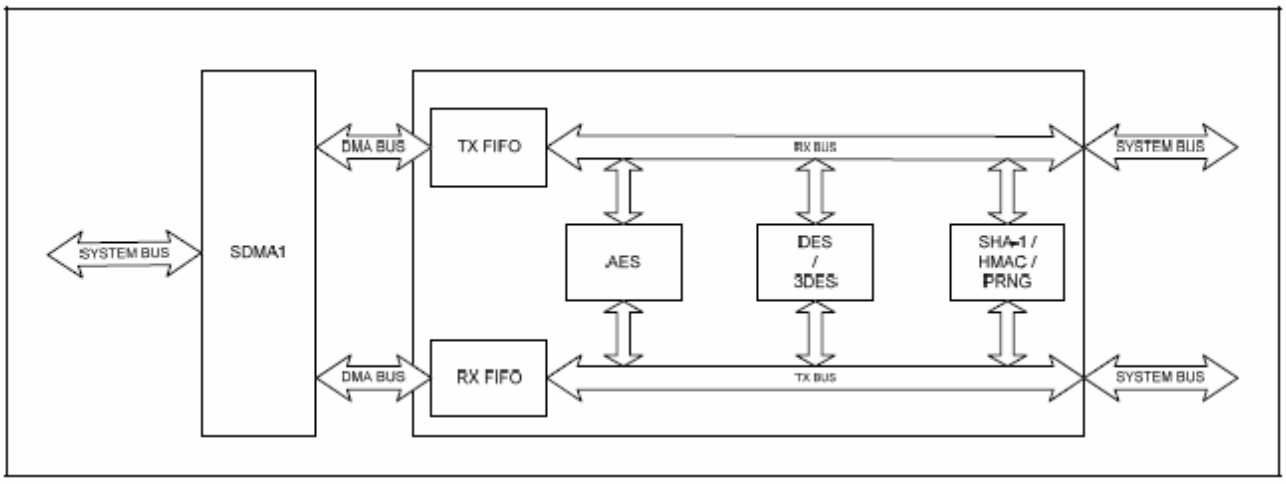

图 13-1 安全子系统模块图

安全子系统的主要性能有：

（1）对称密钥加速器：AES: 支持 ECB, CBC, CTR 模式

DES/3DES: 支持 ECB, CBC 模式

（2）Hash 引擎：支持 SHA-1

支持 H/W HMAC

（3）随机数发生器：每 160 周期产生 PRNG 320 位

（4）FIFO-Rx/Tx：两个 32 字的输入输出流

13.1.1 AES 编程向导

可以根据数据和 Meta 数据转换方式，通过 FIFO 操作或 ARM 操作执行 AES。这里描述 FIFO 基本操作。

1.在寄存器内写入密钥和 Meta 数据。根据密钥大小向密钥寄存器内写入密钥。

2.在 AES 控制寄存器内，设置 AES 操作的 AES 配置密钥长度，设置 AES 操作的防线，操作模式及计数器大小。

3.Rx/Tx　FIFO 相关寄存器的设置遵循以下规则：

（1）FIFO-Rx 信息长度寄存器设置：AES 操作的整体信息的长度（32 位）。

（2）FIFO-Rx 模块尺寸寄存器设置：AES 运算法则特殊模块尺寸（32 位）

（3）FIFO-Rx 目标寄存器设置：AES 输入缓冲区开始地址。

（4）FIFO-Tx 信息设置：AES 操作的整体信息的长度。

（5）FIFO-Tx 模块尺寸寄存器设置：AES 特殊模块尺寸（32 位）

（6）FIFO-Tx 目标寄存器设置：AES 输入缓冲区开始地址。

（7）FIFO-Rx 控制寄存器设置：1）FRx_Host_Module (FRx_Ctrl[7:6]: 选择特殊算法目标模块（AES:00）)，2）FRx_Host_Rd_En(FRx_Ctrl[5]:主机 FRx_Ctrl[31:16]和 FTx_MlenCnt 区域的读使能)，3）FTx_Host_Wr_En(FRx_Ctrl[4]:主机 FRx_Rx 写使能), 4)FRx_Sync_Tx(FRx_Ctrl[3]: FTx_Rx 写等待使能)，5）FTx_Start(FRx_Ctrl[0]:目标模块 FRx_Rx 数据转换器开始)

（8）FiFo-Tx 控制寄存器设置：1）FRx_Src_Module（FRx_Ctrl[7:6]：选择特殊运算法则源模块（AES:00））,2) FRx_Host_Rd_En(FRx_Ctrl[5]:主机 FRx_Ctrl[31:16]和 FTx_MlenCnt 区域的读使能)， 3）FTx_Host_Wr_En(FRx_Ctrl[4]:主机 FRx_Rx 写使能), 4)FTx_Start(FRx_Ctrl[0]: 源模块 FRx_Rx 数据转换器开始)

4. AES 操作中，向 FiFo_Rx 内写入数据。FiFo_Rx 和 FiFo_Tx 通过以下过程完成数据和 AES 的通信。

（1）向 AES_Rx_DIN_0 寄存器写入数据

（2）开始 AES 运行：根据需要重复下面步骤的第一步到第二步

1）FiFo_Rx 准备轮询 AES 输入

2）FiFo_Rx 向 AES_Rx_DIN_0 寄存器写入数据

3）FiFo_Rx 准备轮询 AEC 输出

4）FiFo_Rx 从 AES_Rx_DIN_0 寄存器内读取数据

5）AES 运行开始

（3）FiFo_Rx 准备轮询 AES 输出

（4）FiFo_Rx 从 AES_Rx_DIN_0 寄存器内读取数据

5.应用 FiFo_Rx 空检检测从 FiFo_Rx 到 AES 的数据传输是否完成。

6.通过检测 FiFo_Rx 的转换运行检测，确认 AES 操作的结果传送到 FiFo_Rx 内，并从 FiFo_Rx 内读取 AES 的运行结果。

13.1.2 TDES

可以根据数据和 Meta 数据转换方式，通过 FIFO 操作或 ARM 操作执行 TDES。这里描述 FIFO 基本操作。

1.向寄存器内写入密钥、Meta 数据和 ECB 模式。设置 TDES 控制寄存器的 DES/TDES 选择、方向、或操作方式。

2. 遵循以下规则设置 Rx/Tx　FIFO 相关寄存器：

（1）FIFO-Rx 信息长度寄存器设置：TDES 操作的整体信息的长度（32 位）。

（2）FIFO-Rx 模块尺寸寄存器设置：TDES 运算法则特殊模块尺寸（32 位）

（3）FIFO-Rx 目标寄存器设置：TDES 输入缓冲区开始地址。

（4）FIFO-Tx 信息设置：TDES 操作整体信息的长度。

（5）FIFO-Tx 模块尺寸寄存器设置：TDES 特殊模块尺寸（32 位）

（6）FIFO-Tx 目标寄存器设置：TDES 输入缓冲区开始地址。

（7）FIFO-Rx 控制寄存器设置：1）FRx_Dest_Module(FRx_Ctrl[7:6]: 选择特殊算法目标模块（TDES/DES:01:00），2）FRx_ Dest _Rd_En(FRx_Ctrl[5]:主机 FRx_Ctrl[31:16]和 FTx_MlenCnt 区域的读使能)，3）FTx_Host_Wr_En(FRx_Ctrl[4]:主机 FRx_Rx 写使能), 4)FRx_Sync_Tx(FRx_Ctrl[3]: FTx_Rx 写等待使能)，5）FTx_Start(FRx_Ctrl[0]:目标模块 FRx_Rx 数据转换器开始)

（8）FiFo-Tx 控制寄存器设置：1）FRx_Src_Module（FRx_Ctrl[7:6]：选择特殊运算法则源模块（TDES/DES:01:00））,2) FRx_Host_Rd_En(FRx_Ctrl[5]:主机 FRx_Ctrl[31:16]和 FTx_MlenCnt 区域的读使能)， 3）FTx_Host_Wr_En(FRx_Ctrl[4]:主机 FRx_Rx 写使能), 4)FTx_Start(FRx_Ctrl[0]: 源模块 FRx_Rx 数据转换器开始)

3.TDES 操作时，主机向 FiFo_Rx 内写入数据。FiFo_Rx 和 FiFo_Tx 通过以下过程完成数据和 TDES

之间的通信。

（1）向 TDES_Rx_INPUT_0 寄存器写入数据

（2）开始 TDES 运行：根据需要重复下面步骤的第一步到第二步

1）FiFo_Rx 准备轮询 TDES 输入

2）FiFo_Rx 向 TDES_Rx_INPUT_0 寄存器写入数据

3）FiFo_Rx 准备轮询 TDEC 输出

4）FiFo_Rx 从 TDES_Rx_INPUT_0 寄存器内读取数据

5）TDES 运行开始

（3）FiFo_Rx 准备轮询 AES 输出

（4）FiFo_Rx 从 TDES_Rx_INPUT_0 寄存器内读取数据

4.应用 FiFo_Rx 空检检测 TDES 操作的所有数据是否从 FiFo_Rx 传输到 TDES。

5.通过检测 FiFo_Rx 的转换运行检测，确认 AES 操作的结果传送到 FiFo_Rx 内，并从 FiFo_Rx 内读取 AES 的运行结果。

13.1.3 SHA-1&PRNG

根据数据和 Meta 数据转换方式，通过 FIFO 操作或 ARM 操作执行 SHA-1 和 PRNG。这里描述 FIFO 基本操作。

1.配置 HASH_CTRL 寄存器(选择引擎（SHA-1，HAMC_SHA-1 ，PRNG），密钥或文本、引擎开始等等)

2. 遵循以下规则设置 Rx/Tx　FIFO 相关寄存器：

（1）FIFO-Rx 信息长度寄存器设置：操作的整体信息的长度（32 位）。

（2）FIFO-Rx 模块尺寸寄存器设置：Hash 运算法则特殊模块尺寸（32 位）

（3）FIFO-Rx 目标寄存器设置：输入缓冲区开始地址。

（4）FIFO-Tx 信息设置：操作的整体信息的长度。

（5）FIFO-Tx 模块尺寸寄存器设置：Hash 运算法则特殊模块尺寸（32 位）

（6）FIFO-Tx 目标寄存器设置：输入缓冲区开始地址。

（7）FIFO-Rx 控制寄存器设置：1）FRx_Dest_Module(FRx_Ctrl[7:6]: 选择特殊算法目标模块（SHA-1/PRNG:01）)，2）FRx_ Host _Rd_En(FRx_Ctrl[5]:主机 FRx_Ctrl[31:16]和 FTx_MlenCnt 区域的读

使能)，3）FTx_Host_Wr_En(FRx_Ctrl[4]:主机 FRx_Rx 写使能), 4)FRx_Sync_Tx(FRx_Ctrl[3]: FTx_Rx 写等待使能)，5）FTx_Start(FRx_Ctrl[0]:目标模块 FRx_Rx 数据转换器开始)

（8）FiFo-Tx 控制寄存器设置：1）FRx_Src_Module（FRx_Ctrl[7：6]：选择特殊运算法则源模块（SHA-1/PRNG:01））,2) FRx_Host_Rd_En(FRx_Ctrl[5]:主机 FRx_Ctrl[31:16]和 FTx_MlenCnt 区域的读使能)， 3）FTx_Host_Wr_En(FRx_Ctrl[4]:主机 FRx_Rx 写使能), 4)FTx_Start(FRx_Ctrl[0]: 源模块 FRx_Rx 数据转换器开始)

3.Hash 操作时，向 FiFo_Rx 内写入数据。FiFo_Rx 和 FiFo_Tx 通过以下过程完成数据和 Hash 的通信。

（1）向 Hash_Rx_DIN_0~ Hash_Rx_DIN_15 寄存器写入数据

（2）开始运行：根据需要重复下面步骤的第一步到第二步

1）FiFo_Rx 准备轮询输入

2）FiFo_Rx 向 Hash_Rx_DIN_0~ Hash_Rx_DIN_15 寄存器写入数据

（3）FiFo_Rx 准备轮询 SHA-1_PRNG 输出

（4）FiFo_Rx 从 HASH_OUTPUT_0 寄存器内读取数据

4.应用 FiFo_Rx 空检检测数据是否从 FiFo_Rx 传输到 SHA-1/PRNG。

5.通过检测 FiFo_Rx 转换器运行确认 SHA-1/PRNG 操作的结果传送到 FiFo_Rx 内，并从 FiFo_Rx 内读取运行结果。

13.2 特殊功能寄存器

1.安全子系统寄存器映射

表 13-2 DMA 和中断控制寄存器映射

地址	读/写	复位值	名称	描述
基本地址+0x00	读/写	0x0000_0000	Dnl_CFG	DMA 和终端配置寄存器
基本地址=0x7D00_0000				

表 13-3 FIFO_Rx 寄存器映射

地址	读/写	复位值	名称	描述
基本地址+0x00	读/写	0x0420_0000	FRx_Ctrl	FIFO-Rx 控制和状态寄存器
基本地址+0x04	读/写	0x0000_0000	FRx_Mlen	FIFO-Rx 信息长度寄存器

基本地址+0x08	读/写	0x0000_0000	FRx_BlkSz	FIFO-Rx 加密算法模块尺寸寄存器
基本地址+0x0C	读/写	0x0000_0000	FRx_DestAddr	FIFO-Rx 输入缓冲地址寄存器
基本地址+0x10	读/写	0x0000_0000	FRx_MlenCnt	FIFO-Rx 信息计数寄存器
基本地址+0x40	写	0x0000_0000	FRx_WrBuf	FIFO-Rx 写缓冲区
…	…	…	…	…
基本地址+0x7C	写	0x0000_0000	FRx_WrBuf	FIFO-Rx 写缓冲区
基本地址=0x7D40_0000 基本地址=0x7D90_0000				

注：写访问 FTx_WrBuf 使 FIFO-Tx 忽略已给的地址向 FIFO 存储器内写入数据。这就意味着在 0x0040 和 0x0080 之间的任何地址将会触发 FIFO 存储器的读操作。这个性能用突发写操作向 FIFO-Tx 产生程序。

表 13-4 **FIFO_Tx 寄存器映射**

地址	读/写	复位值	名称	描述
基本地址+0x00	读/写	0x0420_0000	FTx_Ctrl	FIFO-Rx 控制和状态寄存器
基本地址+0x04	读/写	0x0000_0000	FTx_Mlen	FIFO-Tx 信息长度寄存器
基本地址+0x08	读/写	0x0000_0000	FTx_BlkSz	FIFO-Tx 加密算法模块尺寸寄存器
基本地址+0x0C	读/写	0x0000_0000	FTx_DestAddr	FIFO-Tx 输入缓冲地址寄存器
基本地址+0x10	读/写	0x0000_0000	FTRx_MlenCnt	FIFO-Tx 信息计数寄存器
基本地址+0x40	读	0x0000_0000	FTx_RdBuf	FIFO-Tx 读缓冲区
…	…	…	…	…
基本地址+0x7C	读	0x0000_0000	FTx_RdBuf	FIFO-Tx 读缓冲区
基本地址=0x7D80_0000 基本地址=0x7DA0_0000				

注：读访问 FTx_WrBuf 使 FIFO-Tx 忽略已给的地址从 FIFO 存储器内读取数据。这就意味着在 0x0040 和 0x007c 之间的任何地址将会触发 FIFO 存储器的读操作。这个性能用突发读操作向 FIFO-Tx 产生程序。

表 13-5 AES 寄存器映射

地址	读/写	复位值	名称	描述
Rx-AES 寄存器映射（RX 部分）				
基本地址+0x00	读/写	0x0000_0200	AES_Rx_Ctrl	AES Rx 控制和状态寄存器

基本地址+0x10	读/写	0x0000_0000	AES_Rx_DIN_01	AES Rx 数据输入寄存器 01
基本地址+0x14	读/写	0x0000_0000	AES_Rx_DIN_02	AES Rx 数据输入寄存器 02
基本地址+0x18	读/写	0x0000_0000	AES_Rx_DIN_03	AES Rx 数据输入寄存器 03
基本地址+0x1C	读/写	0x0000_0000	AES_Rx_DIN_04	AES Rx 数据输入寄存器 04
基本地址+0x20	读	0x0000_0000	AES_Rx_DOUT_01	AES Rx 数据输出寄存器 01
基本地址+0x24	读	0x0000_0000	AES_Rx_DOUT_02	AES Rx 数据输出寄存器 02
基本地址+0x28	读	0x0000_0000	AES_Rx_DOUT_03	AES Rx 数据输出寄存器 03
基本地址+0x2C	读	0x0000_0000	AES_Rx_DOUT_04	AES Rx 数据输出寄存器 04
基本地址+0x80	读/写	0x0000_0000	AES_Rx_KEY_01	AES Rx 密钥输入寄存器 01
基本地址+0x84	读/写	0x0000_0000	AES_Rx_KEY_02	AES Rx 密钥输入寄存器 02
基本地址+0x88	读/写	0x0000_0000	AES_Rx_KEY_03	AES Rx 密钥输入寄存器 03
基本地址+0x8C	读/写	0x0000_0000	AES_Rx_KEY_04	AES Rx 密钥输入寄存器 04
基本地址+0x90	读/写	0x0000_0000	AES_Rx_KEY_05	AES Rx 密钥输入寄存器 05
基本地址+0x94	读/写	0x0000_0000	AES_Rx_KEY_06	AES Rx 密钥输入寄存器 06
基本地址+0x98	读/写	0x0000_0000	AES_Rx_KEY_07	AES Rx 密钥输入寄存器 07
基本地址+0x9C	读/写	0x0000_0000	AES_Rx_KEY_08	AES Rx 密钥输入寄存器 08
基本地址+0xA0	读/写	0x0000_0000	AES_Rx_IV_01	AES Rx IV 输入寄存器 01
基本地址+0xA4	读/写	0x0000_0000	AES_Rx_IV_02	AES Rx IV 输入寄存器 02
基本地址+0xA8	读/写	0x0000_0000	AES_Rx_IV_03	AES Rx IV 输入寄存器 03
基本地址+0xAC	读/写	0x0000_0000	AES_Rx_IV_04	AES Rx IV 输入寄存器 04
基本地址+0xB0	读/写	0x0000_0000	AES_Rx_CTR_01	AES Rx 计数器预算寄存器 01
基本地址+0xB4	读/写	0x0000_0000	AES_Rx_CTR_02	AES Rx 计数器预算寄存器 02
基本地址+0xB8	读/写	0x0000_0000	AES_Rx_CTR_03	AES Rx 计数器预算寄存器 03
基本地址+0xBC	读/写	0x0000_0000	AES_Rx_CTR_04	AES Rx 计数器预算寄存器 04
Tx-AES 寄存器映射（TX 部分）				
基本地址+0x20	读	0x0000_0000	AES_Tx_DIN_01	AES Tx 数据输入寄存器 01
基本地址+0x24	读	0x0000_0000	AES_Tx_DIN_02	AES Tx 数据输入寄存器 02
基本地址+0x28	读	0x0000_0000	AES_Tx_DIN_03	AES Tx 数据输入寄存器 03