

FTx_SrcAddr	0x7D80_000C	读/写	FIFO-Tx 输出缓冲地址寄存器	0x0000_0000
-------------	-------------	-----	-------------------	-------------

FTx_SrcAddr	位	描述	初始状态
SrcAddr	[31: 0]	源输出缓冲的内部存储器地址。FIFO-Tx 可以向此地址转换数据。当设置 FTx_Ctrl 寄存器的 FTX_Reset 位设置时，将复位值重设。	0x0000_0000

5. FIFO_Tx 信息长度计数器

寄存器	地址	读/写	描述	复位值
FTx_MlenCnt	0x7D40_0010	读/写	FIFO-Tx 信息长度计数寄存器	0x0000_0000

FTx_MlenCnt	位	描述	初始状态
MlenCnt	[31: 0]	剩下用于转换的字的数目。	0x0000_0000

6. FIFO_Tx 写缓冲区

寄存器	地址	读/写	描述	复位值
FTx_WrBuf	0x7D80_0040 ~ 0x7D80_007C	写	FIFO-Tx 写缓冲区 (32 字) 注: 这个地址是 CPU 访问地址	0x0000_0000
	0x7DA0_0040 ~ 0x7DA0_007C		FIFO-Tx 写缓冲区 (32 字) 注: 这个地址是 SDMA1 的访问地址。可以在此地址用 SDMA1 进行存储器到 FTx_WrBuf 的数据转换。	

FTx_RdBuf	位	描述	初始状态
RdBuf	[31: 0]	FIFO-Tx 读缓冲区 (32x32 位)	0x0000_0000

注: 读访问 FTx_WrBuf 时, FIFO-Tx 根据已给的地址从 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。

13.6 安全子系统 AES 模块

1. AES_CTRL

寄存器	地址	读/写	描述	复位值
AES_Rx_CTRL	0x7D10_0000	读/写	AES 控制/状态寄存器	0x0000_0200

SKEY_IDx	位	描述	初始状态
WrPrivMismatch	[31]	SFR 写访问权限不匹配状态位。如果此位设置为 ‘1 ‘，会发生 SFR 写访问权限不匹配现象。	0b
RdPrivMismatch	[30]	SFR 读访问权限不匹配状态位。如果此位设置为 ‘1 ‘，会发生 SFR 读访问权限不匹配现象。	0b
Reserved	[29;11]	保留	0x000000
AesOutReady	[10]	如果设置为 ‘1 ‘，AES 输出缓冲为满置状态, 允许 ARM 或 Rx FIFO 读取当前 128 位结果数据。	0b
AesInReady	[9]	如果设置为 ‘1 ‘，AES 输出缓冲为空状态, 允许 ARM 或 Rx FIFO 读取下一个 128 位结果数据。	1b
AesContDecOn	[8]	继续解密启用位： 0: 改变解密密钥 1: 不改变解密密钥	0b
CtrWidth	[7: 6]	计数器模式计数器宽度位 00: 16 位计数器 01: 32 位计数器 10: 64 位计数器 11: 保留	00b
AesOpMode	[5: 4]	AES 运行模式选择位 00=保留 01=ECB 模式 10=CBC 模式 11=CTR 模式	00b
AesOpDirection	[3]	AES 操作方向选择位 00: 加密 01=解密	0b
AesKeyMode	[2: 1]	AES 密钥模式选择位	0b

		00: 128 位 01: 192 位式 10: 256 位 11: 保留	
AesOpEnable	[0]	如果设置 ‘1 ‘, AES 开始通过 ARM 运行。如果产生 AES_OP_DONE, 此位变成 ‘0 ‘	0b

2. AES_RX_DIN_01~ AES_RX_DIN_04

寄存器	地址	读/写	描述	复位值
AES_Rx_DIN_01	0x7D10_0010	读/写	AES 数据输入寄存器 01	0x0000_0000
AES_Rx_DIN_02	0x7D10_0014	读/写	AES 数据输入寄存器 02	0x0000_0000
AES_Rx_DIN_03	0x7D10_0018	读/写	AES 数据输入寄存器 03	0x0000_0000
AES_Rx_DIN_04	0x7D10_001C	读/写	AES 数据输入寄存器 04	0x0000_0000

AES_Rx_DIN	位	描述	初始状态
AESDIN	[31:0]	AES 第一个到第四个 32 位数据输入寄存器	0x0000_0000

3. AES_RX_DOUT_01~ AES_RX_DOUT_04/ AES_TX_DOUT_01~ AES_TX_DOUT_04

寄存器	地址	读/写	描述	复位值
AES_RX_DOUT_01	0x7D10_0020	读/写	AES 数据输出寄存器 01	0x0000_0000
AES_RX_DOUT_02	0x7D10_0024	读/写	AES 数据输出寄存器 02	0x0000_0000
AES_RX_DOUT_03	0x7D10_0028	读/写	AES 数据输出寄存器 03	0x0000_0000
AES_RX_DOUT_04	0x7D10_002C	读/写	AES 数据输出寄存器 04	0x0000_0000
AES_TX_DOUT_01	0x7D10_0020	读/写	AES 数据输出寄存器 01	0x0000_0000
AES_TX_DOUT_02	0x7D10_0024	读/写	AES 数据输出寄存器 02	0x0000_0000
AES_TX_DOUT_03	0x7D10_0028	读/写	AES 数据输出寄存器 03	0x0000_0000
AES_TX_DOUT_04	0x7D10_002C	读/写	AES 数据输出寄存器 04	0x0000_0000

AES_Tx_DOUT	位	描述	初始状态
AESDout	[31:0]	AES 第一个到第四个 32 位数据输出寄存器	0x0000_0000

4. AES_RX_KEY_01~ AES_RX_KEY_08

寄存器	地址	读/写	描述	复位值
AES_RX_KEY_01	0x7D10_0080	读/写	AES 密钥输入寄存器 01	0x0000_0000
AES_RX_KEY_02	0x7D10_0084	读/写	AES 密钥输入寄存器 02	0x0000_0000
AES_RX_KEY_03	0x7D10_0028	读/写	AES 密钥输入寄存器 03	0x0000_0000
AES_RX_KEY_04	0x7D10_008C	读/写	AES 密钥输入寄存器 04	0x0000_0000
AES_TX_KEY_05	0x7D10_0090	读/写	AES 密钥输入寄存器 05	0x0000_0000
AES_TX_KEY_06	0x7D10_0094	读/写	AES 密钥输入寄存器 06	0x0000_0000
AES_TX_KEY_07	0x7D10_0098	读/写	AES 密钥输入寄存器 07	0x0000_0000
AES_TX_KEY_08	0x7D10_009C	读/写	AES 密钥输入寄存器 08	0x0000_0000

AES_Rx_KEY	位	描述	初始状态
AESKey	[31:0]	AES 第一个到第八个 32 位密钥输入寄存器	0x0000_0000

5. AES_RX_IV_01~ AES_RX_IV_04

寄存器	地址	读/写	描述	复位值
AES_Rx_IV_01	0x7D10_00A0	读/写	AES IV 输入寄存器 01	0x0000_0000
AES_Rx_IV_02	0x7D10_00A4	读/写	AES IV 输入寄存器 02	0x0000_0000
AES_Rx_IV_03	0x7D10_00A8	读/写	AES IV 输入寄存器 03	0x0000_0000
AES_Rx_IV_04	0x7D10_00AC	读/写	AES IV 输入寄存器 04	0x0000_0000

AES_Rx_IV	位	描述	初始状态
AESIV	[31:0]	AES 第一个到第四个 32 位 IV 输入寄存器	0x0000_0000

6. AES_RX_CTR_01~ AES_RX_CTR_04

寄存器	地址	读/写	描述	复位值
AES_Rx_CTR_01	0x7D10_00B0	读/写	AES 计数器预先输入寄存器 01	0x0000_0000
AES_Rx_CTR_02	0x7D10_00B4	读/写	AES 计数器预先输入寄存器 02	0x0000_0000

AES_Rx_CTRL_03	0x7D10_00B8	读/写	AES 计数器预先输入寄存器 03	0x0000_0000
AES_Rx_CTRL_04	0x7D10_00BC	读/写	AES 计数器预先 V 输入寄存器 04	0x0000_0000

AES_Rx_CTRL	位	描述	初始状态
AECtr	[31:0]	AES 第一个到第四个 32 位计数器预先输入寄存器	0x0000_0000

13.7 安全子系统 DES/CDES 模块

1. TDES_RX_CTRL

寄存器	地址	读/写	描述	复位值
TDES_Rx_CTRL	0x7D20_0000	读/写	TDES 控制/状态寄存器	0x0000_0200

TDES_Rx_CTRL	位	描述	初始状态
WrPrivMismatch	[31]	SFR 写访问权限不匹配状态位。如果此位设置为 ‘1’，会发生 SFR 写访问权限不匹配现象。	0b
RdPrivMismatch	[30]	SFR 读访问权限不匹配状态位。如果此位设置为 ‘1’，会发生 SFR 读访问权限不匹配现象。	0b
Reserved	[29;8]	保留	0x0000_00
TdesOutReady	[7]	如果设置为 ‘1’，TDES 输出缓冲为满置状态, 允许 ARM 或 Rx FIFO 读取当前 128 位结果数据。	0b
TdesInReady	[6]	如果设置为 ‘1’，TDES 输出缓冲为空状态, 允许 ARM 或 Rx FIFO 读取下一个 128 位结果数据。	1b
DesOrTdes	[5]	DES 或者 TDES 运行选择位 0: DES 方式 1: TDES 方式	0b
TdesMode	[4: 3]	AES 模式选择位 01=ECB 模式 10=CBC 模式	00b
TdesOpDirection	[2]	AES 操作方向选择位 00: 加密 01=解密	0b

TdesIntMode	[1]	TDES 操作结束模式选择位 00: 轮询方式 01=中断方式	0b
TdesOpEnable	[0]	如果设置 ‘1 ‘, TDES 开始通过 ARM 运行。如果产生 DES_ 或者 TDES_OP_DONE, 此位变成 ‘0 ‘	0b

2. TDES_RX_KEY1_0

寄存器	地址	读/写	描述	复位值
TDES_RX_KEY1_0	0x7D20_0010	读/写	TDES 密钥输入寄存器 1_0	0x0000_0000
TDES_RX_KEY1_1	0x7D20_0014	读/写	TDES 密钥输入寄存器 1_1	0x0000_0000
TDES_RX_KEY2_0	0x7D10_0018	读/写	TDES 密钥输入寄存器 2_0	0x0000_0000
TDES_RX_KEY2_1	0x7D10_001C	读/写	TDES 密钥输入寄存器 2_1	0x0000_0000
TDES_RX_KEY3_0	0x7D10_0020	读/写	TDES 密钥输入寄存器 3_0	0x0000_0000
TDES_RX_KEY3_1	0x7D10_0024	读/写	TDES 密钥输入寄存器 3_1	0x0000_0000

TDES_Rx_KEY	位	描述	初始状态
TDESKey	[31:0]	AES 第一个到第六个 32 位密钥输入寄存器	0x0000_0000

3. TDES_RX_INPUT_0/ TDES_RX_INPUT_1

寄存器	地址	读/写	描述	复位值
TDES_RX_INPUT_0	0x7D20_0040	读/写	TDES 数据输入寄存器 0	0x0000_0000
TDES_RX_INPUT_1	0x7D20_0044	读/写	TDES 数据输入寄存器 1	0x0000_0000

TDES_Rx_INPUT	位	描述	初始状态
TDESdinx	[31:0]	AES 第一个到第二个 32 位数据输入寄存器	0x0000_0000

4. TDES_RX_OUTPUT_0/ TDES_TX_OUTPUT_0/ TDES_RX_INPUT_1 TDES_TX_INPUT_1

寄存器	地址	读/写	描述	复位值
TDES_RX_OUTPUT_0	0x7D20_0048	读/写	TDES 数据输出寄存器 0	0x0000_0000

TDES_RX_OUTPUT_1	0x7D20_004C	读/写	TDES 数据输出寄存器 1	0x0000_0000
TDES_TX_OUTPUT_0	0x7D60_0048	读/写	TDES 数据输出寄存器 0	0x0000_0000
TDES_TX_OUTPUT_1	0x7D60_004C	读/写	TDES 数据输出寄存器 1	0x0000_0000

TDES_Rx_OUTPUT	位	描述	初始状态
TDESdout	[31:0]	AES 第一个到第二个 32 位数据输出寄存器	0x0000_0000

5. TDES_RX_IV_0~ TDES_RX_IV_1

寄存器	地址	读/写	描述	复位值
TDES_Rx_IV_0	0x7D20_0050	读/写	TDES IV 输入寄存器 0	0x0000_0000
TDES_Rx_IV_1	0x7D20_0054	读/写	TDES IV 输入寄存器 1	0x0000_0000

TDES_Rx_IV	位	描述	初始状态
TdesIv	[31:0]	TDES 第一个到第二个 32 位 IV 输入寄存器	0x0000_0000

13.8 安全子系统 SHA-1/PRNG 模块

1. HASH_CONTROL

寄存器	地址	读/写	描述	复位值
HASH_CONTROL	0x7D30_0000	读/写	HASH 引擎控制寄存器	0x0000_0000

HASH_CONTROL	位	描述	初始状态
Reserved	[31:9]	保留	0x0000_00
USE_IV	[8]	使用任意 IV 替代 SHA-1 常数 0: 常数 1:任意 IV	0b
Edn_of_Hash_byte	[7:6]	SHA 文本字节的末端 00: 第一字节 01: 第二字节 10: 第三字节 04: 第四字节	00b

SEED_SETTING_ENABLE	[5]	种子设置使能	0b
Hash_input_finished	[4]	结束 Hash 输入（由硬件清除）	0b
Hash_start	[3]	开始 Hash	0b
Date_Selection	[2]	指明寄存器下一个数值是密钥还是文本 0: 文本 1: 密钥	0b
Engine_Selection	[1:0]	选择使用 SHA-1 或 HMAC 或 PRNG 00:HMAC 01:SHA1 10:PRNG 11=保留	00b

2. HASH-DATA

寄存器	地址	读/写	描述	复位值
HASH_DATA	0x7D30_0004	读/写	HASH 数据	0x0000_0000

HASH_DATA	位	描述	初始状态
HASH_DATA	[31:0]	Hash 数据输入寄存器	0x0000_0000

3. SEED_DATA-01~SEED_DATA_10

寄存器	地址	读/写	描述	复位值
SEED_DATA_01	0x7D30_0008	读/写	PRNG 种子数据 1 ([31:0])	0x0000_0000
SEED_DATA_02	0x7D30_000C	读/写	PRNG 种子数据 2 ([63:32])	0x0000_0000
SEED_DATA_03	0x7D30_0010	读/写	PRNG 种子数据 3 ([95:64])	0x0000_0000
SEED_DATA_04	0x7D30_0014	读/写	PRNG 种子数据 4 ([127:96])	0x0000_0000
SEED_DATA_05	0x7D30_0018	读/写	PRNG 种子数据 5 ([159:128])	0x0000_0000
SEED_DATA_06	0x7D30_001C	读/写	PRNG 种子数据 6 ([191:160])	0x0000_0000
SEED_DATA_07	0x7D30_0020	读/写	PRNG 种子数据 7 ([223:192])	0x0000_0000
SEED_DATA_08	0x7D30_0024	读/写	PRNG 种子数据 8 ([255:224])	0x0000_0000
SEED_DATA_09	0x7D30_0028	读/写	PRNG 种子数据 9 ([287:256])	0x0000_0000
SEED_DATA_10	0x7D30_002C	读/写	PRNG 种子数据 10 ([319:288])	0x0000_0000

SEED_DATA	位	描述	初始状态
SEED_DATA	[31:0]	PRNG 种子数据	0x0000_0000

4. HASH_STATUS

寄存器	地址	读/写	描述	复位值
HASH_STATUS	0x7D30_0030	读	HASH 状态	0x0000_0010

HASH_STATUS	位	描述	初始状态
Reserved	[31:5]	保留	0x0000_00
BUFFER_IN_ENABLE	[4]	1:缓冲区输入使能（缓冲区为空） 0: 缓冲区不能输入数据（缓冲区已满）	1b
HASH_engine_ready	[3]	已准备接收输入数据的下一个 64 字节	0b
Random_Number_Ready	[2]	随机数字	0b
32bit_ready	[1]	只在测试时使用 准备运行下一个 32 位	0b
HASH_output_ready	[0]	已经完成 160 位的 Hash 计算，准备应用	0b

5. HASH_OUTPUT_01 (PRNG_OUTPUT_01_)~HASH_OUTPUT_10 (PRNG_OUTPUT_10)

寄存器	地址	读/写	描述	复位值
HASH_OUTPUT_01	0x7D30_0034	读	HASH 输出 (01) 或 PRNG 输出 [31:0]	0x0000_0000
HASH_OUTPUT_02	0x7D30_0038	读	HASH 输出 (02) 或 PRNG 输出 [63:32]	0x0000_0000
HASH_OUTPUT_03	0x7D30_003C	读	HASH 输出 (03) 或 PRNG 输出 [95: 64]	0x0000_0000
HASH_OUTPUT_04	0x7D30_0040	读	HASH 输出 (04) 或 PRNG 输出 [127: 96]	0x0000_0000
HASH_OUTPUT_05	0x7D30_0044	读	HASH 输出 (05) 或 PRNG 输出 [159: 128]	0x0000_0000
HASH_OUTPUT_06	0x7D30_0048	读	HASH 输出 (06) 或 PRNG 输出 [191: 160]	0x0000_0000
HASH_OUTPUT_07	0x7D30_004C	读	HASH 输出 (07) 或 PRNG 输出 [223: 192]	0x0000_0000
HASH_OUTPUT_08	0x7D30_0050	读	HASH 输出 (08) 或 PRNG 输出 [255: 224]	0x0000_0000
HASH_OUTPUT_09	0x7D30_0054	读	HASH 输出 (09) 或 PRNG 输出 [287: 256]	0x0000_0000

HASH_OUTPUT_10	0x7D30_0058	读	HASH 输出 (10) 或 PRNG 输出 [319: 288]	0x0000_0000
HASH_OUTPUT_01	0x7D70_0034	读	HASH 输出 (01) 或 PRNG 输出 [31:0]	0x0000_0000
HASH_OUTPUT_02	0x7D70_0038	读	HASH 输出 (02) 或 PRNG 输出 [63:32]	0x0000_0000
HASH_OUTPUT_03	0x7D70_003C	读	HASH 输出 (03) 或 PRNG 输出 [95: 64]	0x0000_0000
HASH_OUTPUT_04	0x7D70_0040	读	HASH 输出 (04) 或 PRNG 输出 [127: 96]	0x0000_0000
HASH_OUTPUT_05	0x7D70_0044	读	HASH 输出 (05) 或 PRNG 输出 [159: 128]	0x0000_0000
HASH_OUTPUT_06	0x7D70_0048	读	HASH 输出 (06) 或 PRNG 输出 [191: 160]	0x0000_0000
HASH_OUTPUT_07	0x7D70_004C	读	HASH 输出 (07) 或 PRNG 输出 [223: 192]	0x0000_0000
HASH_OUTPUT_08	0x7D70_0050	读	HASH 输出 (08) 或 PRNG 输出 [255: 224]	0x0000_0000
HASH_OUTPUT_09	0x7D70_0054	读	HASH 输出 (09) 或 PRNG 输出 [287: 256]	0x0000_0000
HASH_OUTPUT_10	0x7D70_0058	读	HASH 输出 (10) 或 PRNG 输出 [319: 288]	0x0000_0000

HASH_OUTPUT	位	描述	初始状态
HASH_OUTPUT	[31:0]	DANG Engine_selection[1:0]==2' b10, PRNG_output, 或者 Rx/Tx 为 Hash_output[31:0]	0x0000_0000

6. HASH_MIDOUT_01~HASH_MIDOUT_05

寄存器	地址	读/写	描述	复位值
HASH_MIDOUT_01	0x7D30_005C	读	HASH_MIDOUT [159: 128]	0x0000_0000
HASH_MIDOUT_02	0x7D30_0060	读	HASH_MIDOUT [127: 96]	0x0000_0000
HASH_MIDOUT_03	0x7D30_0064	读	HASH_MIDOUT [95: 64]	0x0000_0000
HASH_MIDOUT_04	0x7D30_0068	读	HASH_MIDOUT [63:32]	0x0000_0000
HASH_MIDOUT_05	0x7D30_006C	读	HASH_MIDOUT [31:0]	0x0000_0000
HASH_MIDOUT_01	0x7D70_005C	读	HASH_MIDOUT [159: 128]	0x0000_0000
HASH_MIDOUT_02	0x7D70_0060	读	HASH_MIDOUT [127: 96]	0x0000_0000
HASH_MIDOUT_03	0x7D70_0064	读	HASH_MIDOUT [95: 64]	0x0000_0000
HASH_MIDOUT_04	0x7D70_0068	读	HASH_MIDOUT [63:32]	0x0000_0000
HASH_MIDOUT_05	0x7D70_006C	读	HASH_MIDOUT [31:0]	0x0000_0000