doi: 10.7690/bgzdh.2025.07.003

基于国产 FPGA 与 SM4 算法的总线通信加解密 IP 核设计

李　森，袁　强，唐　建
（中国兵器装备集团自动化研究所有限公司特种计算机事业部，四川 绵阳 621000）

摘要：针对某终端设备之间总线通信的加密需求以及加密硬件程序便捷移植开发的需求，设计基于 SM4 加解密算法的 IP 核。采用轻量级单轮循环迭代结构，在满足吞吐率要求下降低对 FPGA 逻辑资源的消耗，其中查找表（look-up table，LUT）的消耗为 2 447 个，触发器（flip-flop，FF）的消耗为 2 914 个；采用加密、解密独立运行的方式使通信过程中的加密运算与解密运算可同时进行互不干扰；使用 Vivado2022.2 对 IP 核进行封装；采用一种“线性变换参数可控”的机制，使加密安全性由单一的密钥作为保证变为“密钥+线性变换参数”的双重保证，提高加密的安全性。测试结果表明：在满足吞吐率要求的条件下，该 IP 核可应用于 UART（485/422/232）、CAN、1553B 等总线的数据加密，且性能稳定。

关键词：FPGA；SM4 算法；IP 核；线性变换参数可控
中图分类号：TN918.4　　**文献标志码：**A

Design of Bus Communication Encryption and Decryption IP Core Based on Domestic FPGA and SM4 Algorithm

Li Sen, Yuan Qiang, Tang Jian
(*Department of Special Computer*, *Automation Research Institute Co*., *Ltd*. *of China South Industries Group Co*., *Ltd*., *Mianyang* 621000, *China*)

Abstract: An IP core based on SM4 encryption and decryption algorithm is designed to meet the encryption requirements of bus communication between terminal devices and the requirements of convenient transplantation and development of encryption hardware programs. A lightweight single-round loop iteration structure is adopted to reduce the consumption of FPGA logic resources while meeting the throughput requirements, wherein the consumption of a lookup table (LUT) is 2 447, and the consumption of a flip-flop (FF) is 2 914; The independent operation mode of encryption and decryption is adopted, so that the encryption operation and the decryption operation in the communication process can be carried out at the same time without mutual interference; A “linear transformation parameter controllable” mechanism is used to change the encryption security from a single key as a guarantee to a double guarantee of “key + linear transformation parameter”, which improves the security of encryption. The test results show that the IP core can be applied to the data encryption of UART(485/422/232), CAN, 1553B and other buses under the condition of meeting the throughput requirements, and the performance is stable.

Keywords: FPGA; SM4 algorithm; IP core; controllable parameters of linear transformation

0　引言

随着物联网、大数据、人工智能的发展，在数据爆炸的时代，数据安全变得越来越重要，为保证数据安全，各国推出了自己的加密体系与算法。我国相继推出了 SM1～SM4、SM7、SM9 等国密算法。2012 年，国家商用密码管理局将 SM4 算法确定为密码行业标准（GM/T000—2012），2016 年其成为国家标准（GB/T32907—2016）。目前，SM4 密码算法主要应用于无线局域网产品及终端设备[1-2]。目前，密码算法主要分为对称密码算法与非对称密码算法 2 类，其中对称密码算法具有高效简洁、部署便捷、加密/解密速度高等优势；因此，对称密码算法在数据加密领域得到了广泛应用[3]。SM4 算法是一种分组对称密码算法，具有安全性强、效率高、硬件易于实现等优点。目前，对 SM4 密码算法的研究主要集中在高吞吐率、优化实现、可重构等方面。王泽芳等[1]对 CTR 模式下的 SM4 算法 ASIC 实现进行了研究；何诗洋等[3]对 SM4 在 FPGA 上的各种实现方式进行了优化研究；武小年等[4]对 SM4 的门限掩码方案进行了研究；薛煜骞等[5-7]对 SM4 分组密码算法的可重构与硬件实现进行了深入研究；王文静等[8-9]对 SM4 算法在光通信与 RFID 射频通信中的加密应用进行了深入研究。

基于终端设备对总线通信（UART、CAN、1553B）的加密需求，以及便捷移植性的要求，笔者利用国

收稿日期：2024-09-10；**修回日期：**2024-10-21
第一作者：李　森（1993—），男，四川人，硕士。

产 FPGA 对 SM4 加解密算法进行设计实现与 IP 核封装，使用 Vivado2022.2 对其进行封装，使其可以快速应用于基于 Vivado 开发平台的设计中。为进一步简单、高效地提高加密性能，设计一种“线性变换参数可控”的机制，使其在提高加密安全性的同时，保证了轻量化要求。通过对 IP 核在总线通信中的应用进行测试，结果表明满足设计要求。

1 SM4 密码算法概述

SM4 算法由加密算法、解密算法、密钥扩展算法 3 部分组成，采用的都是 32 轮的非线性迭代结构，每轮迭代实现 32 bit 数据的加解密运算。SM4 算法的数据分组与密钥均为 128 bit，加密运算与解密运算的结构完全相同，轮密钥使用顺序相反。

1.1 密钥扩展算法原理

密钥扩展就是由原始长度为 128 bit 的密钥 MK 进行 32 次迭代运算，生成 32 个位宽为 32 bit 的轮密钥 $rk_i(i=0, 1, \cdots, 31)$，其计算流程如下[10]：

输入：密钥 $MK=(MK_0, MK_1, MK_2, MK_3)$，$MK_i \in Z_2^{32}$，$i=0, 1, 2, 3$。

输出：轮密钥 rk_i，$i=0, 1, \cdots, 31$。

令：$(K_0, K_1, K_2, K_3)=(MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3)$；

则：$rk_i=K_{i+4}=K_i \oplus T'(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i)$，进行 32 轮迭代运算，生成 $rk_0, rk_1, \cdots, rk_{31}$。

下面对密钥扩展算法中用到的算子符号做一个说明：

1) Z_2^n 表示 n 位的二进制序列，$(Z_2^n)^4$ 表示 4 个 n 位二进制序列；

2) (FK_0, FK_1, FK_2, FK_3)为 SM4 算法的固定参数，其值分别为：$FK_0=(\text{A3B1BAC6})_H$，$FK_1=(\text{56AA3350})_H$，$FK_2=(\text{677D9197})_H$，$FK_3=(\text{B27022DC})_H$，CK_i 为固定常数，其取值参见 SM4 算法国密标准相关文档，这里不列举赘述；

3) $\oplus$：表示按位异或运算；

4) $T'(\bullet)$为非线性变换算子，由非线性变换 $\tau(\bullet)$ 与线性变换 $L'(\bullet)$2 部分组成，即：$T'(\bullet)=L'(\tau(\bullet))$；

5) 设线性变换 $L'(\bullet)$的输入为 $B \in Z_2^{32}$，则：$L'(B)=B \oplus (B<<<13) \oplus (B<<<23)$；

6) $B<<<i$，表示将 B 循环左移 i 位；

7) 非线性变换 $\tau(\bullet)$由 4 个并行的 Sbox 盒构成，设非线性变换 $\tau(\bullet)$的输入为 $A=(a_0, a_1, a_2, a_3) \in (Z_2^8)^4$，则：$\tau(A)=(\text{Sbox}(a_0), (\text{Sbox}(a_1), (\text{Sbox}(a_2), (\text{Sbox}(a_3))$，其中 Sbox 盒映射表可以查看 SM4 算法国密标准，这里不再给出。

1.2 加解密算法原理

加密运算是将输入的 128 bit 明文$(X_0, X_1, X_2, X_3) \in (Z_2^{32})^4$经过 32 轮迭代运算，变成 128 bit 的密文$(Y_0, Y_1, Y_2, Y_3) \in (Z_2^{32})^4$输出。其计算过程如下：

输入：明文$(X_0, X_1, X_2, X_3) \in (Z_2^{32})^4$；

输出：密文$(Y_0, Y_1, Y_2, Y_3)=(X_{35}, X_{34}, X_{33}, X_{32})$。

加密运算，轮函数的迭代格式为：$X_{i+4}=X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i)$，$i=0, 1, \cdots, 31$；当 32 轮迭代运算完毕后，将最后一次迭代的输出进行反序变换，即为密文(Y_0, Y_1, Y_2, Y_3)。下面对加密轮函数中用到的算子符号做一个说明：

1) $T(\bullet)$为非线性变换算子，由非线性变换 $\tau(\bullet)$与线性变换 $L(\bullet)$ 2 部分组成，即：$T(\bullet)=L(\tau(\bullet))$；

2) 非线性 $\tau(\bullet)$变换与密钥扩展中的 $\tau(\bullet)$完全相同，这里不再赘述；

3) 设线性变换 $L(\bullet)$的输入为 $B \in Z_2^{32}$，则：$L(B)=B \oplus (B<<<2) \oplus (B<<<10) \oplus (B<<<18) \oplus (B<<<24)$，这里的 2、10、18、24 就是轮函数线性变换的参数，在下文的算法实现中，该参数将设计成可控的机制。

解密运算与加密运算采用完全相同的结构进行，只是在进行迭代运算时，使用轮密钥 rk_i 的顺序与加密运算相反，按照 $rk_{31}, rk_{30}, \cdots, rk_1, rk_0$ 的顺序参与轮函数的迭代运算[8]。

2 SM4 密码算法硬件实现

2.1 整体架构设计

IP 核采用加密运算、解密运算独立并行工作的方式进行加密、解密运算，同时采用“线性变换参数可控”的机制实现 SM4 算法的加密、解密运算，整个 IP 核的整体结构如图 1 所示，由加密模块 SM4_encrypt、轮密钥存储矩阵 $\boldsymbol{A}$、轮密钥生成模块、轮密钥存储矩阵 $\boldsymbol{B}$、解密模块 SM4_decrypt 5 部分组成。SM4 密码算法 IP 核的顶层结构如图 2 所示。

端口信号说明如下：

1) clk、reset_n：系统时钟与系统复位，复位低有效。

2) ciphertext_in[127:0]：128 bit 密文数据输入；ciphertext_en 密文输入使能，高有效。

3) key_in[127:0]：128 bit 密钥输入；key_en：密钥输入使能，高有效。

4) plaintext_in[127:0]：128 bit 明文输入；plaintext_en 明文输入使能，高有效。

5) decry_out_en：解密输出数据有效标志，高电平有效；decry_busy：解密模块忙状态标志，高电平表示忙；decry_data_out[127:0]：128 bit 解密数据输出。

6) encry_out_en：加密数据输出有效标志，高电平有效；encry_busy：加密模块忙状态标志，高电平表示忙；encry_data_out[127:0]：128 bit 加密数据输出。

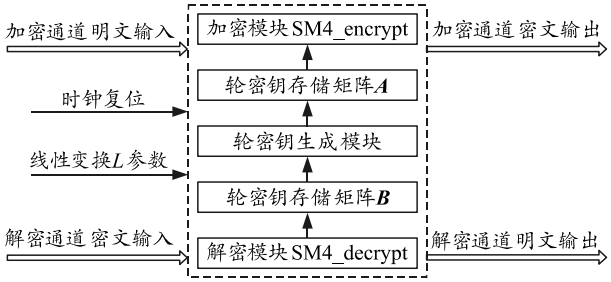

图 1　SM4IP 核整体结构

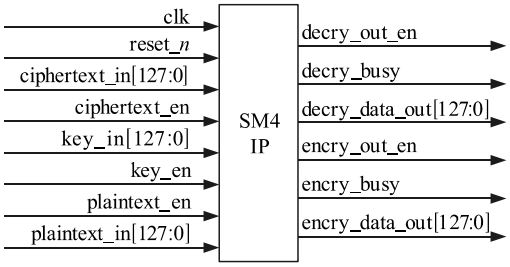

图 2　SM4IP 模块顶层

2.2　轮密钥生成模块设计

轮密钥生成模块用于对密钥进行扩展，在 SM4 密码算法中，轮密钥扩展迭代函数的结构如图 3 所示[11]。

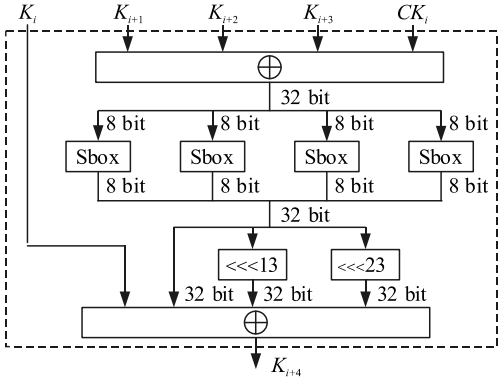

图 3　轮密钥扩展迭代函数结构

由图 3 可知，轮密钥扩展迭代函数，在每一轮迭代运算中：1) 对输入 K_{i+1}, K_{i+2}, K_{i+3} 以及 CK_i 进行异或运算；2) 将结果从高到低每 8 bit 一组送入 Sbox 盒；3) 将 Sbox 盒子的输出进行线性变换，线性变换后与 K_i 异或，得到该次迭代的输出 K_{i+4}。根据图 3 所示的结构，对轮密钥生成模块进行设计，其顶层结构如图 4 所示。

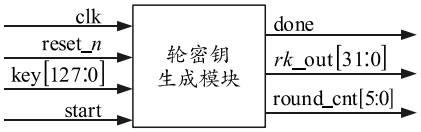

图 4　轮密钥生成模块顶层

端口信号说明如下：

1) clk、reset_n：系统时钟与系统复位，复位低有效。

2) key[127:0]：128 bit 的密钥输入；start：轮密钥扩展计算启动信号，该信号为脉冲信号，高有效，每次有效，轮密生成模块将连续迭代计算 32 次。

3) done：轮密钥每轮计算结束信号，高有效；rk_out[31:0]：每轮 32 bit 轮密输出；round_cnt[5:0]：当前迭代计算的序号。

2.3　加密、解密模块设计

加密、解密模块用于对输入的明文、密文进行加密、解密运算。在加密、解密模块中最核心的结构为加解密轮函数。SM4 为对称密码算法，它的加密、解密轮函数结构完全相同，具体如图 5 所示。

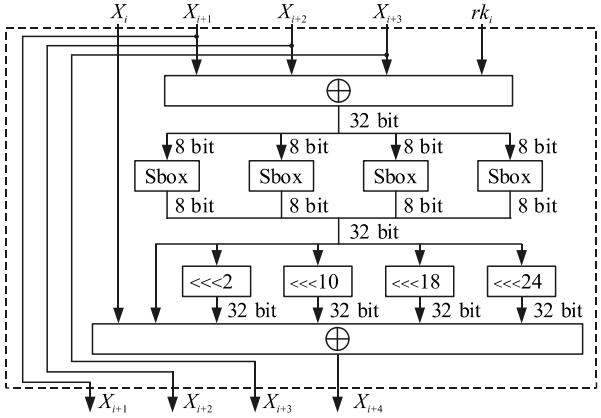

图 5　加密、解密轮函数结构

为降低对逻辑资源的消耗，达到轻量化设计的要求，采用迭代的方式实现对明文、密文的加解密；因此，每个分组的数据进入加解密模块后，至少要经过 32 个时钟周期，才能得到加解密的结果。加解密轮函数每次计算的输出作为下次计算的输入，直到第 32 次迭代计算完毕后，该次的输出经过反序变换后，即为加密、解密的结果。根据图 5 所示的轮函数内部结构，对加解密模块进行设计，其顶层结构如图 6 所示。

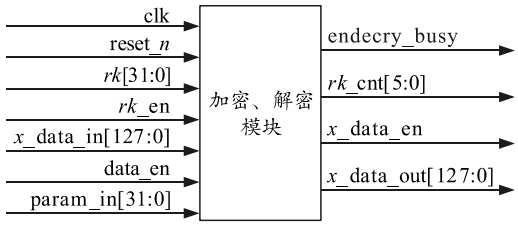

图 6 加密、解密模块顶层

端口信号说明如下：

1) clk、reset_*n*：系统时钟与复位，低有效。

2) *rk*_en：轮密钥使能信号，高有效；*rk*[31:0]：轮密钥输入。

3) *x*_data_in[127:0]：128 bit 明文或密文输入；data_en：明文输入使能脉冲，高有效。

4) param_in[31:0]：线性变换 *L* 的参数输入，从高到低，每 8 bit 一组用于控制轮函数中线性变换的结构，分别对应 *Lt*1Par、*Lt*2Par、*Lt*3Par、*Lt*4Par。

5) endecry_busy：加解密模块忙状态标志，用于表示加解密模块当前的工作状态；*rk*_cnt[5:0]：轮密钥访问序号，用于从轮密钥矩阵中读取相应的轮密钥。

6) *x*_data_en：明文、密文输出有效脉冲，高有效；*x*_data_out[127:0]：128 bit 明文、密文输出。

2.4 仿真分析

为验证设计的正确性，对 SM4 密码算法进行仿真验证。当密钥 MK=(0123456789ABCDEFFEDCBA9876543210)$_H$，线性变换参数为标准 SM4 参数(Lt1Par=2、Lt2Par=10、Lt3Par=18、Lt4Par=24)时，抽取任意 2 组仿真结果如图 7 所示。当 MK=(0123456789ABCDEFFEDCBA9876543210)$_H$，线性变换参数为(Lt1Par=1、Lt2Par=3、Lt3Par=5、Lt4Par=7)时，抽取任意 2 组仿真结果如图 8 所示。由图 7 和 8 可知，向加密模块输入明文后，得到对应的密文，将密文输入解密模块，可以看到，解密出的结果正是输入加密模块的明文；因此，IP 核设计正确。

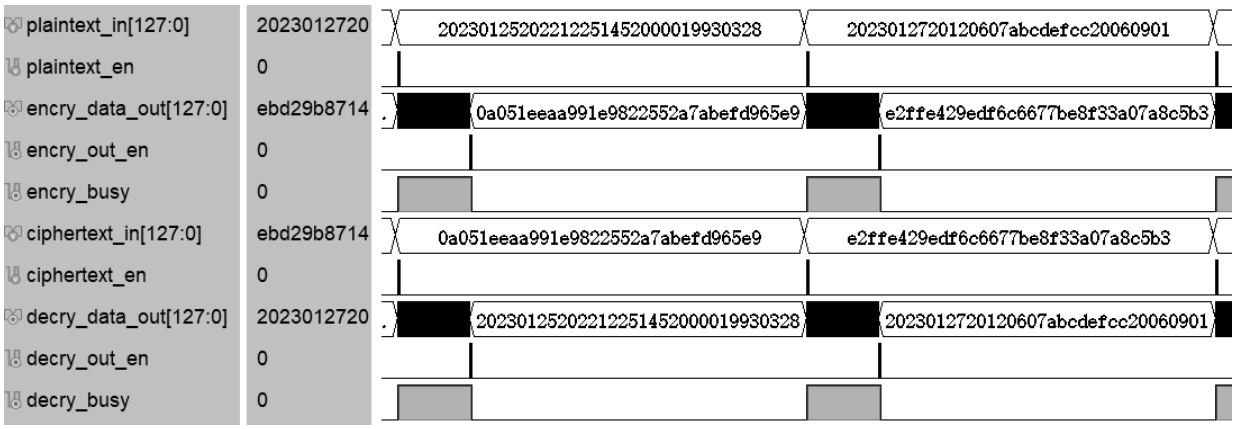

图 7 仿真结果 1

图 8 仿真结果 2

2.5 IP 核封装

为便于在 Vivado 开发环境中快速便捷的使用 IP 核，在设计中使用 Vivado2022.2 对 IP 核进行封装；并用 GUI 配置界面实现对线性变换参数的设置，如图 9 所示。其中 Lt1Par、Lt2Par、Lt3Par、Lt4Par 4 个参数，分别对应解密、加密模块中迭代轮函数线

性变换 L 中循环左移的位数，其取值范围为 0～31，通过调用 IP 核时设置不同的线性变换参数，实现线性变换参数可控；对应的接收方在进行解密运算时，必须使用相同的线性变换配置参数和密钥，才能译码出正确的明文，实现加密由密钥的单一保护，变为“密钥+线性变换参数”的双重保护结构。

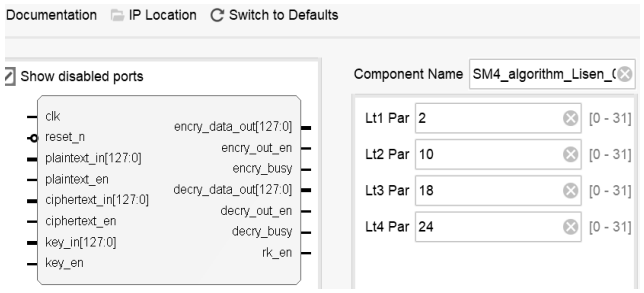

图 9　IP 核封装结果

3　实际验证

3.1　资源与吞吐率

完成 SM4 IP 核的设计后，需要对该 IP 核的实际性能进行测试，利用 Vivado2022.2 IDE 对 IP 进行综合，综合后 IP 核对资源的消耗如表 1 所示。

表 1　IP 资源消耗　　%

Resource	Estimation	Available	Utilization
LUT	2 447	203 800	1.20
FF	2 914	407 600	0.71
IO	650	500	130.00
BUFG	1	32	3.13

由表 1 可知：消耗 LUT 2 447 个，触发器 FF 消耗 2 914 个。由于轻量化要求，该 IP 核并未采用流水线结构实现[12]，而是采用迭代方式实现，在每组处理数据为 128 bit 条件下，其吞吐率最大可达 540.96 Mbps，对于其应用对象(1553B 总线、CAN 总线、UART485/422/232)，该吞吐率满足实际应用需求。

3.2　实际应用测试

为验证 IP 核实际加密、解密的效果，在国产 FPGA 上以 UART 接口作为实际验证对象，验证设计的可行性。

验证加密功能，通过调试 IP 核 VIO 向 FPGA 注入原始明文，并通过串口助手接收密文，验证加密功能；当密钥为$(0123456789ABCDEFFEDCBA9876543210)_H$，线性变换参数为($Lt$1Par, Lt1Par2, Lt1Par3, Lt1Par4)=(2, 10, 18, 24)（标准线性变换参数)时，对大量数据进行加密测试，在大量结果中随机抽取 5 组，如图 10 所示，其中上方 5 组数据为明文，下方为串口接收侧对应的 5 组密文。由图 10 可知：串口接收到的密文，均为对应明文的加密结果，证明加密侧功能正确。

图 10　加密侧实际验证

验证解密功能，通过串口调试助手发送密文，FPGA 接收密文，并通过 VIO 输出解密出的明文数据，其密钥与线性变换 L 的参数与上方加密侧的密钥和参数相同；在大量结果中随机抽取 5 组，如图 11 所示，其中上方 5 组数据为串口发送的密文，下方 5 组数据为对应解密出的明文。由图 11 可知：解密出的明文均为对应密文的解密结果，证明解密侧功能正确。

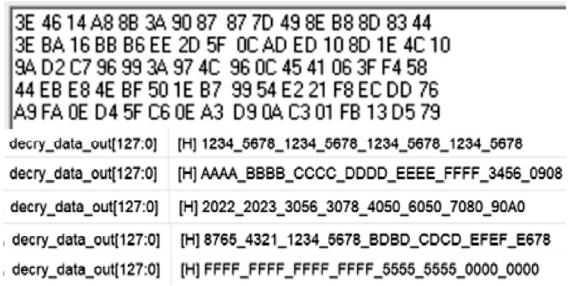

图 11　解密侧实际验证

4　结束语

基于某终端设备总线通信的加密需求，笔者以 SM4 密码算法为核心，设计一种用于总线通信加密的 IP 核。在传统 SM4 密码算法的基础上采用“线性变换参数可控”的机制提高加密的安全性。采用 VivadoIDE 对 IP 核进行封装，使其可以快速在平台上进行复用，并给出了 IP 核逻辑资源消耗与吞吐率参数。最后以国产 FPGA 为验证平台，以 UART 串口为对象，对加密、解密功能进行设计验证，验证结果满足设计需求；该 IP 核的设计可为其他总线加密提供一种设计参考。

参考文献：

[1] 王泽芳，唐中剑. SM4 算法 CTR 模式的高吞吐率 ASIC 实现[J]. 电子器件, 2019, 42(1): 173-177.

（下转第 58 页）