

SM4 算法 CBC 模式的高吞吐率 ASIC 实现

符天枢^{1,2}, 李树国^{1,2}

(1 清华大学 微电子学研究所, 北京 100084; 2 清华信息科学与技术国家实验室, 北京 100084)

摘 要: 由于 SM4 算法在 CBC 模式下存在从电路的输出端到输入端的反馈路径, 所以流水线技术难以提高电路的吞吐率. 针对这一问题, 提出一种逻辑化简方法, 使 SM4 加解密算法中每一个轮函数的关键路径减少 1 级异或门延时. 基于这种方法, 实现了一种 4 轮合 1 的 SM4 电路, 在该电路的关键路径中可以减少 4 级异或门延时, 且该电路与本文的其他方案相比有更高的单位面积吞吐率. ASIC 实现的综合结果表明, 4 轮合 1 的 SM4 电路在 CBC 模式下的吞吐率达到 5.24 Gb/s, 高于已发表的同类设计.

关键词: SM4; CBC 模式; 高吞吐率; ASIC 实现

中图分类号: TN49

文献标识码: A

文章编号: 1000-7180(2016)10-0013-06

A High-Throughput ASIC Implementation of SM4 Algorithm in CBC Mode

FU Tian-shu^{1,2}, LI Shu-guo^{1,2}

(1 Institute of Microelectronics, Tsinghua University, Beijing 100084, China;

2 Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China)

Abstract: In CBC mode, pipeline technique does not work on SM4 algorithm to increase throughput, as there is a feedback path from output to input. Over this problem, a logic simplifying method is proposed, which can reduce delay of one stage of XOR gates in each round function of encryption algorithm of SM4. Based on this method, SM4 with a 4-round-in-1 structure is designed, in which, delay of 4 stages of XOR gates can be reduced in the critical path, and this design has a higher throughput per unit area than the other schemes in this paper. Synthesis results show that the ASIC implementation of SM4 with a 4-round-in-1 structure can achieve 5.24 Gb/s in throughput, which is higher than that of reported designs.

Key words: SM4; CBC mode; high-throughput; ASIC implementation

1 引言

SM4 是我国国家密码管理局公布的分组对称密码算法, 广泛应用于加密通信、加密存储等领域^[1]. SM4 算法常见的工作模式有密文分组链接(CBC)模式、电子密码本(ECB)模式等, 其中 CBC 模式比 ECB 模式有更高的安全性, 并且有着更大的应用需求. 但是, 如何提高 SM4 算法在 CBC 模式下的吞吐率是一个困难的问题, 因为在 CBC 模式下, 电路中有从输出端到输入端的反馈路径, 使得当前得到的密文分组与之前得到的密文分组有数据相关

性, 导致无法使用流水线技术来提高电路的吞吐率. 相反, 在 ECB 模式下电路中没有反馈路径, 各个密文分组之间没有数据相关性, 可以使用流水线技术来提高电路的吞吐率. 一些已发表的 SM4 实现方案使用了流水线技术^[2], 使得电路在 ECB 模式下能够达到比较高的吞吐率. 但是, 缺少对 CBC 模式下 SM4 的吞吐率进行提高的研究.

AES 是美国国家标准和技术研究所公布的分组对称密码算法, 该算法也存在着 CBC 模式下难以提高吞吐率的问题. 文献[3、4]采用一种改进方法, 将电路中 S 盒以外的其他逻辑结构进行预计算, 把

预计算的结果与 S 盒的逻辑融合,构成新的查找表,从而提高 AES 电路在 CBC 模式下的吞吐率. 这种将 S 盒改造成新查找表的思想同样可以用于 SM4 算法中.

本文提出一种 SM4 算法在 CBC 模式下的逻辑化简方法,可以使 SM4 加解密算法的每个轮函数的关键路径减少 1 级异或门延时. 基于这种方法设计出一种 4 轮合 1 的组合逻辑数据通路,在这一数据通路的关键路径中,可以减少 4 级异或门延时,从而提高 SM4 算法在 CBC 模式下的吞吐率.

2 SM4 算法描述

SM4 算法的分组长度为 128 位,密钥长度也是 128 位. SM4 算法的结构分为密钥扩展算法和加解密算法两部分.

2.1 SM4 的密钥扩展算法

SM4 的密钥扩展算法结构如图 1 所示,密钥扩展的过程主要包括 32 轮密钥扩展的轮函数,该轮函数与加解密算法中的轮函数结构相似,但不完全相同. 密钥为 128 位,FK 为 SM4 标准中规定的一个 128 位的常数, $K_s (s = 0, 1, 2, \dots, 35)$ 为 32 位数据, $rk_i (i = 0, 1, 2, \dots, 31)$ 为密钥扩展算法产生的 32 个子密钥,并且有 $rk_i = K_{i+4}$.

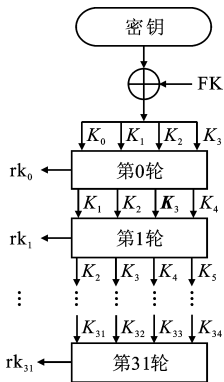


图 1 SM4 的密钥扩展算法结构

2.2 SM4 的加解密算法

SM4 的加解密算法的结构如图 2 所示,128 位的输入数据经过 32 轮轮函数的变换及 1 次反序变换,得到 128 位的输出数据. 图中的 $I_s (s = 0, 1, 2, \dots, 35)$ 为 32 位的数据, rk_s 为第 s 轮轮函数所对应的子密钥,由密钥扩展算法产生.

SM4 算法的解密过程与加密过程基本相同,只需要将 32 个子密钥的使用顺序颠倒即可.

加解密算法的轮函数结构如图 3 所示. 图中 $\oplus$ 代表异或运算, I_s 至 I_{s+3} 为第 s 轮的输入, I_{s+1} 至 I_{s+4}

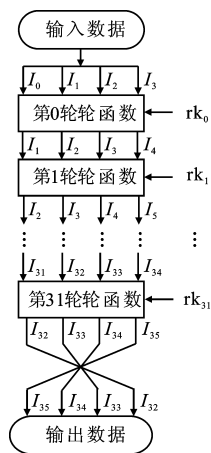


图 2 SM4 的加解密算法结构

为第 s 轮的输出. rk_s 为第 s 轮对应的子密钥. T 函数包括非线性的 τ 函数及线性的 L 函数. τ 函数由并行的 4 个完全相同的 S 盒组成,每个 S 盒是一个 8 位输入、8 位输出的查找表. S 盒是 SM4 算法中唯一的非线性变换,对算法的安全性起着至关重要的作用. L 函数由一些循环左移变换和异或操作组成. 图 3 中的加粗箭头表示轮函数结构的关键路径.

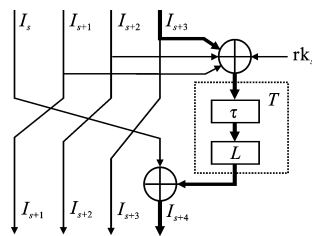


图 3 SM4 加解密算法的轮函数结构

T 函数的内部结构如图 4 所示. 图中的 $\ll\ll$ 表示对 32 位的数据进行循环左移操作,如“ $\ll\ll 2$ ”表示将 32 位数据循环左移 2 位. 图中加粗的箭头表示 T 函数中的关键路径,包括 1 个 S 盒及 3 级异或门. 由于在 ASIC 实现时,循环左移操作可以用硬连线来实现,所以图 4 中的循环左移操作不影响关键路径延时.

3 对 SM4 加解密算法关键路径的化简

函数 T 在加解密算法轮函数的关键路径上. 故减小函数 T 的关键路径延时可以使得加解密算法轮函数的关键路径延时变小.

由图 4 可知,设 $Sbox_1, Sbox_2, Sbox_3, Sbox_4$ 的输入分别为 $(a_{31}, \dots, a_{24}), (a_{23}, \dots, a_{16}), (a_{15}, \dots, a_8), (a_7, \dots, a_0)$, 输出分别为 $(b_{31}, \dots, b_{24}), (b_{23}, \dots, b_{16}), (b_{15}, \dots, b_8), (b_7, \dots, b_0)$, 其中 a_i 和 $b_i (i = 0, 1,$

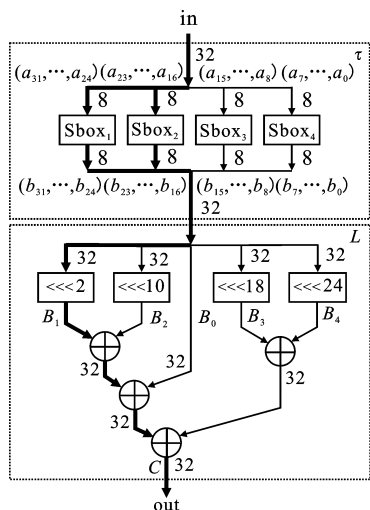


图4 未经过化简的函数T的结构

2, ..., 31) 分别表示1位数据, 则图4中有

$$B_0 = (b_{31}, b_{30}, b_{29}, \dots, b_2, b_1, b_0)$$

对 B_0 分别做循环左移2位、10位、18位和24位的操作, 得到 B_1, B_2, B_3, B_4 , 则有

$$B_1 = (b_{29}, b_{28}, b_{27}, \dots, b_0, b_{31}, b_{30})$$

$$B_2 = (b_{21}, b_{20}, b_{19}, \dots, b_{24}, b_{23}, b_{22})$$

$$B_3 = (b_{13}, b_{12}, b_{11}, \dots, b_{16}, b_{15}, b_{14})$$

$$B_4 = (b_7, b_6, b_5, \dots, b_{10}, b_9, b_8)$$

令

$$C = B_0 \oplus B_1 \oplus B_2 \oplus B_3 \oplus B_4$$

则 C 也是32位数据, 设

$$C = (c_{31}, c_{30}, c_{29}, \dots, c_2, c_1, c_0)$$

其中 $c_i (i = 0, 1, 2, \dots, 31)$ 为1位数据, 则有

$$c_{31} = (b_{31} \oplus b_{29}) \oplus b_{21} \oplus b_{13} \oplus b_7$$

$$c_{30} = (b_{30} \oplus b_{28}) \oplus b_{20} \oplus b_{12} \oplus b_6$$

$$c_{29} = (b_{29} \oplus b_{27}) \oplus b_{19} \oplus b_{11} \oplus b_5$$

.....

$$c_2 = (b_2 \oplus b_0) \oplus b_{24} \oplus b_{16} \oplus b_{10}$$

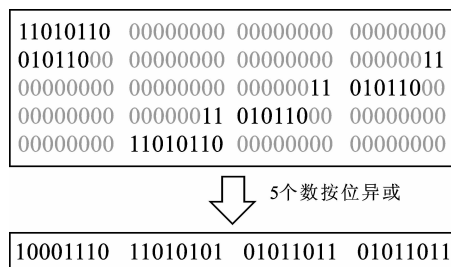
$$c_1 = b_1 \oplus b_{31} \oplus b_{23} \oplus (b_{15} \oplus b_9)$$

$$c_0 = b_0 \oplus b_{30} \oplus b_{22} \oplus (b_{14} \oplus b_8)$$

以上从 c_{31} 到 c_0 的等式中每组括号中的两位数据表明该两位数据来自同一个Sbox的输出, 例如 c_{31} 中 $(b_{31} b_{29})$ 用括号标记是因为 b_{31} 和 b_{29} 都是Sbox₁的输出, $(b_{31} b_{29})$ 的值可以在设计电路时提前预计算, 然后将预计算得到的值直接存放在Sbox₁的查找表中, 这样就使得在计算 c_{31} 时, 异或运算的操作数由5个减少为4个, 则异或门级数可以由3级变为2级。同理 c_{30} 中的 $(b_{30} b_{28})$ 、 c_{29} 中的 $(b_{29} b_{27})$ 、...、 c_{26} 中的 $(b_{26} b_{24})$ 也可以将预计算的值存放在Sbox₁的查找表中, 使得在计算 c_{30} 、 c_{29} 、...、 c_{26} 时, 也能减少1级异

或门。

将与Sbox₁的输出 $(b_{31}, \dots, b_{24})$ 相关的异或操作及循环左移操作均提前进行预计算, 并存放在一个新的查找表中, 记为Tbox₁。Tbox₁是一个8位输入, 32位输出的查找表, 表中有 16×16 个元素, 每个元素都是对原Sbox₁中对应位置元素进行预计算的结果。例如Sbox₁中第0行第0列的元素为8位的8'h6, 扩展为32位则为32'h60000000, 如图5所示(图5中为二进制表示), 再分别循环左移2、10、18和24位, 分别得到32'h58000003、32'h00000358、32'h00035800及32'h00d60000, 将以上5个32位的数按位异或, 得到32'h8ed55b5b。则Tbox₁中第0行第0列的元素就是32'h8ed55b5b。通过计算过程中观察得知Tbox₁中每个元素的第三和第四个字节是相同的, 所以虽然Tbox₁的输出是32位, 但在查找表中存储的元素可以只有24位, 只需将24位元素的最后8位复用即可产生32位的输出。比如Tbox₁中第0行第0列的元素可以简化为24'h8ed55b, 这样可以减少Tbox₁查找表所占用的电路面积。

图5 Tbox₁ 中第0行第0列元素的计算过程

同理, Sbox₂, Sbox₃, Sbox₄也可以改进成输入为8位、输出为32位的Tbox₂、Tbox₃、Tbox₄, 其中Tbox₂的每个元素的第一和第四个字节相同, Tbox₃的每个元素的第一和第二个字节相同, Tbox₄的每个元素的第二和第三个字节相同。所以Tbox₂、Tbox₃、Tbox₄中的元素也可以简化为24位, 以节约电路面积, 另外的8位只需将对应的数据进行复用即可。因此图4中的函数T结构可以化简成如图6所示的结构。图6中加粗的箭头表示经过化简后的函数T的关键路径。

对比图4和图6, 图4中函数T的关键路径延时为1个Sbox的延时加3级异或门的延时, 而图6中经过化简后的函数T的关键路径延时为1个Tbox的延时加2级异或门的延时。通过Tbox的构造过程可知, Tbox虽然输出的数据比Sbox多, 但

是 Tbox 输出的每一位仍然是由 28 选 1 的多路选择器产生,这一点与 Sbox 相同,所以 Tbox 的延时与 Sbox 的延时相同. 因此经过化简后的函数 T 的关键路径延时比未化简的函数 T 的关键路径延时减小了一级异或门的延时.

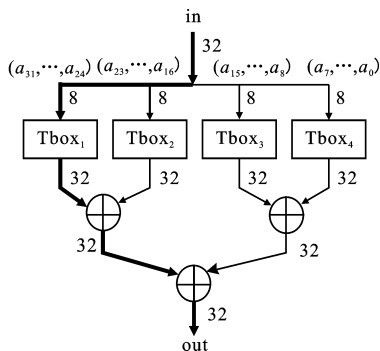


图 6 经过化简后的函数 T 的结构

4 SM4 算法的 ASIC 设计实现

用硬件电路来实现密码算法主要有两种方式,一种是采用现场可编程逻辑门阵列(FPGA),另一种是采用专用集成电路(ASIC). FPGA 实现有设计灵活、开发周期短等优点,但是对电路性能的优化较差,难以满足对电路高吞吐率的要求. 而 ASIC 实现对电路性能的优化较好,适用于追求电路性能的设计. 本设计的目标是提高 SM4 算法在 CBC 模式下的吞吐率,对电路的性能要求较高,所以选择 ASIC 实现.

4.1 设计方案的选取

SM4 的硬件实现主要有两种方案. 一种方案是使用流水线结构. 由于流水线结构不能提高 CBC 模式下电路的吞吐率. 故本设计不采用流水线结构.

另一种方案是使用循环迭代结构,即取 32 个轮函数中的 n 轮(n 是 32 的因子) 构成一个组合电路,称为“ n 轮合 1”电路. 当 $n < 32$ 时,将 n 轮合 1 电路的输出端通过一个多路选择器后反馈到输入端,使得 n 轮合 1 电路可以循环使用. 比如一个 4 轮合 1 电路循环迭代 8 次即可计算完成 32 个轮函数. 这种方案的优点是可以减小电路的面积;缺点是每个时钟周期 n 轮合 1 电路计算得到的中间结果都需要保存到寄存器中,而且进行循环时还需要经过多路选择器. 当 $n = 32$ 时,将 32 个轮函数构成一个组合电路. 这种结构的优点是节省了中间结果寄存器和多路选择器,从而可以减少 32 轮轮函数所需要的时间;缺点是电路的面积比较大. 本文采用循环迭代结构,对

$n = 4, 8, 16, 32$ 时的 n 轮合 1 结构分别进行电路实现,并对这几种实现的性能和面积进行比较.

SM4 的加解密模块如图 7 所示. 其中 IVen 寄存器用于寄存 CBC 模式加密情况下的 IV 值或参与反馈的输出数据分组, IVde 寄存器用于寄存 CBC 模式解密情况下的 IV 值或参与前馈的输入数据分组,当电路工作在 ECB 模式时, IVen 寄存器和 IVde 寄存器均保持 128'h0. 临时寄存器用于寄存 n 轮合 1 电路计算得到的中间结果. 当 $n = 32$ 时,图 7 中的临时寄存器和选择器可以省略.

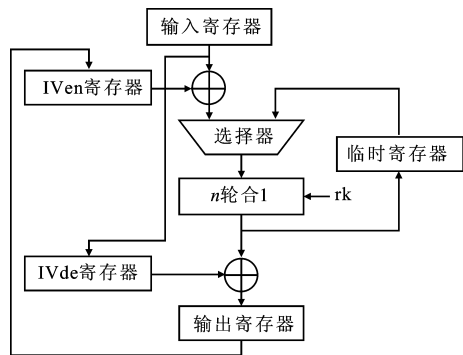


图 7 SM4 的加解密模块

4.2 密钥扩展模块的设计

根据第 2.1 中对 SM4 密钥扩展算法的描述,密钥扩展过程主要包括 32 轮密钥扩展的轮函数. 因为在一次完整的加解密过程中,密钥是不变的,故密钥扩展操作只需要进行一次,因此密钥扩展模块采用一个单轮组合逻辑电路循环 32 次来实现 32 轮轮函数,从而减小密钥扩展模块的面积.

密钥扩展模块的电路结构如图 8 所示. 图中的 FK 为 SM4 标准中规定的一个 128 位的常数,密钥扩展单轮是一个单轮的组合逻辑,寄存器用于寄存单轮电路的输出结果. 图中下方的 $rk_0, rk_1, \dots, rk_{31}$ 表示 32 个 32 位的移位寄存器,用于寄存密钥扩展过程中产生的 32 个子密钥. 电路做加密操作时,移位寄存器的移位路径如图中实线箭头所示,做解密操作时,移位寄存器的移位路径如虚线箭头所示. 移位寄存器在加密、解密的情况下移位方向相反,即可实现 SM4 在加密、解密时子密钥使用顺序相反.

4.3 T 盒的设计

根据第 3 部分中对 Tbox 构成的描述,利用 Sbox 中的每个元素以及 4 种循环左移操作,计算出 4 个 Tbox 中的每个元素,并用查找表法来实现 4 个 Tbox. Tbox 中每个元素包含 24 位,其中有 8 位需要复用,使得输出为 32 位.

表 2 与参考文献中 SM4 实现的性能比较

设计	工艺/ μm	延时/ ns	面积	CBC 吞吐率/ (Mb/s)
文献[5]	0.18	10	78 525 μm^2	100
文献[6]	0.18	20	19.6 k 门	200
文献[7]	0.18	5.4	30 569 μm^2	185
文献[8]	0.18	5	32 k 门	801
本文	0.18	8.91	649 516 μm^2	1 790
文献[9]	0.13	5	22 k 门	800
本文	0.13	6.15	267 361 μm^2	2 600

6 结束语

根据本文中提出的针对 SM4 关键路径进行化简的方法,实现了 4 轮合 1 的 SM4 电路.综合结果表明 4 轮合 1 的 SM4 电路比本文的其他方案有更高的单位面积吞吐率,且 CBC 模式下的吞吐率高于已发表的同类设计.因此,这种对 SM4 关键路径进行化简的方法是有效的,并且对其他分组对称算法提高 CBC 模式下的吞吐率有参考价值.

参考文献:

[1] 国家密码管理局. GM/T 0002-2012. SM4 分组密码算法[S]. 北京:中国标准出版社,2012.

[2] 李大为,赵旭鑫,武萌. SMS4 密码算法的高速流水线实现[J]. 电子器件,2007,30(2):590-592.

[3] Shastry P V S, Somani N, Gadre A, et al. Rolled architecture based implementation of AES using T-Box [C]// 55th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS). Boise, Idaho,

USA; IEEE, 2012; 626-630.

[4] 刘楠,陈迅,李军伟. 面向 CBC 模式的 AES 高速芯片设计与实现[J]. 电子技术应用,2014,40(1):64-67.

[5] 王晨光,乔树山,黑勇. 分组密码算法 SM4 的低复杂度实现[J]. 计算机工程,2013,39(7):177-180.

[6] 徐艳华,白雪飞,郭立. 适合 SMS4 算法硬件实现的 S 盒构造新方法[J]. 中国科学技术大学学报,2009,39(11):1164-1170.

[7] Shang Ming, Zhang Qinglong, Liu Zongbin, et al. An ultra-compact hardware implementation of SMS4[C]// 3rd IIAI International Conference on Advanced Applied Informatics (IIAI-AAI). Kitakyushu, Japan; IEEE Computer Society, 2014; 86-90.

[8] Bai Xuefei, Guo Li, Huang Lu, et al. A fast VLSI design of SMS4 cipher based on twisted BDD S-Box architecture[C]// 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC 2009). Wuhan, China; IEEE Computer Society, 2009; 345-348.

[9] Yan Weiwei, You Kaidi, Han Jun, et al. Low-cost reconfigurable VLSI implementation of the SMS4 and AES algorithms[C] // 2009 IEEE 8th International Conference on ASIC (ASICON). Changsha, China; IEEE, 2009; 135-138.

作者简介:

符天枢 男,(1989-),硕士研究生.研究方向为信息安全算法的数字大规模集成电路设计与实现.
E-mail:fts13@mails.tsinghua.edu.cn.

李树国 男,(1963-),教授,博士生导师.研究方向为信息安全算法及其大规模集成电路设计与实现.

(上接第 12 页)

[5] Lee J, Park K, Kang S. A Die-selection method using search-space conditions for yield enhancement in 3D memory[J]. ETRI Journal, 2011, 33(6): 904-913.

[6] Rab M T, Bawa A, Touba N. Implementing defect tolerance in 3D-ICs by exploiting degrees of freedom in assembly[C]//Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2012 IEEE International Symposium on. USA,Austin,IEEE, 2012; 178-181.

[7] Kang W, Lee C, Cho K, et al. A die selection and matching method with two stages for yield eehancement of 3-D memories[C]//Test Symposium (ATS), 2013 22nd Asian. Taiwan, Yilan, IEEE, 2013; 301-306.

[8] Huang R F, Li J F, Yeh J C, et al. A simulator for e-

valuating redundancy analysis algorithms of repairable embedded memories[C] // Memory Technology, Design and Testing, 2002. (MTDT 2002). Proceedings of the 2002 IEEE International Workshop on. France, Bendor,IEEE, 2002; 68-75.

作者简介:

刘 军 男,(1978-),博士,教授,硕士生导师.研究方向为嵌入式系统、计算机体系结构.

王堂亚(通讯作者) 男,(1990-),硕士研究生.研究方向为三维存储器故障单元容错技术. E-mail:993973922@qq.com

任福继 男,(1959-),留日博士,教授,博士生导师.研究方向为智能机器人、情感计算、自然语言处理、智能科学、大数据与社会计算、云与健康计算.