DOI:10.19304/j.cnki.issn1000-7180.2015.09.011

基于 FPGA 的高性能 3DES 算法实现

朱欣欣[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：传统 3DES 算法需要 48 轮迭代周期，存在吞吐率低的问题，提出二合一的循环迭代结构，该结构完成一次加解密运算需要 25 个时钟周期，兼容了 ECB 和 CBC 两种工作模式. 在 Altera 公司的 Quartus II 13.0 软件上进行 FPGA 实现，选用器件 EP4SGX530NF45C3，延时为 3.61 ns，吞吐率达到了 709.1 Mb/s，面积为 650 ALUTs，性能优于同类设计.

关键词：3DES；吞吐率；循环迭代结构；FPGA
中图分类号：TN49 **文献标识码**：A **文章编号**：1000－7180(2015)09－0054－06

High-Performance 3DES Algorithm Implement Based on FPGA

ZHU Xin-xin[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: Traditional 3DES algorithm requires 48 clock cycles of iterations, which exists the problem of low throughput, while it proposes double-combined-iteration-structure, which costs 25 clock cycles to complete an encryption or decryption calculation, compatible with ECB and CBC operating modes. On Altera's Quartus II 13.0 software for FPGA implementation, the choice of the device EP4SGX530NF45C3, its delay is 3.61 ns, and throughput reaches 709.1 Mb/s, using an area of 650 ALUTs, achieving higher performance compared with similar implementations.

Key words: 3DES; throughput; double-combined-iteration-structure; FPGA

1 引言

在如今网络飞速发展的时代，商业数据的安全性越来越受到人们的重视，加解密技术也在发挥着重大的作用. 主流的加解密方式分为软件加密和硬件加密两种. 传统的软件加密除了运行速度较慢外，安全性较差，还会占用一定的主机资源. 而硬件加密技术使用硬件加密设备进行加密，独立于主机系统之外使其具有较高的安全性，不占用任何主机资源，运行速度也较快[1]. 根据加解密是否使用同一密钥，加密算法分为对称算法和非对称算法两种，其中对称加密算法已被广泛使用. IBM 公司发布的 DES 算法虽然只有 56 bit 密钥长度而被证实不太安全，但是密钥长度增至 168 bit 的 3DES 算法仍然具有较高的安全性. 另一方面，卫星通信、视频传输、网关服务器及其他大量数据传输业务中，DES 算法已经得到了广泛应用，使用 3DES 算法来加强其安全性可使原系统不做大的改动，所以研究 3DES 算法具有很大的现实意义[2].

在许多应用场合，数据加密的速度快慢，直接影响到系统的处理能力，因此高吞吐率加密算法的实现尤为重要. 当需要对大批数据信息进行加解密处理时，高吞吐率的实现方案能够在较短时间内完成数据操作，将加解密结果输送至下级模块. 如果实现

收稿日期：2014－11－26；修回日期：2015－01－13
基金项目：国家“八六三”计划(2012AA012402)；清华大学自主研发计划(2011Z05116)；清华信息科学与技术国家实验室(2015 年立项)

方案的吞吐率低下，势必会成为制约整个系统处理能力的瓶颈. 在另一方面，在集成电路设计中，芯片面积的大小也是一个很重要的问题. 只有在使用尽量小的芯片面积的基础上，尽可能地提高加解密吞吐率的实现方案，才是能够满足系统要求的.

传统 3DES 算法在每次加解密运算中运算使用 48 个时钟周期，运算速度较慢. 本文提出了二合一的循环迭代结构，并对多种循环迭代结构进行了纵向比较，同时在密钥拓展和 S 盒实现方案方面进行了性能优化. 使用配置灵活、性能优越的 FPGA（现场可编程门阵列）对 3DES 算法进行实现，与同类设计相比，本文的实现方案具有更高的加解密速度和更小的电路面积，有效提升了性能.

2　3DES 算法描述

3DES 算法[1]是在 DES 算法的基础上发展起来的. DES 算法使用 56 bit 密钥，进行 16 轮迭代运算. 而 3DES 算法使用 168 bit 密钥，进行 48 轮迭代运算.

2.1　DES 算法描述

DES 算法的输入、密钥和输出都是 64 bit，由于密钥有 8 bit 在密钥拓展过程中被舍弃，故有效密钥长度为 56 bit[3].

如图 1 所示，64 bit 的输入明文通过一个初始置换分成左半部分 L_i 和右半部分 R_i，然后进行 16 轮完全相同的迭代运算，最后经过一个末置换得到 64 bit 的输出密文[4].

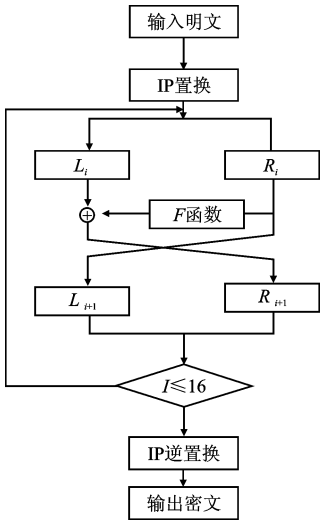

图 1　DES 算法流程图

由图 1 可知，每轮迭代运算包含一个异或运算和 F 函数. 其中 F 函数输入为该轮的右半部分 R_i 和子密钥 K_i，依次进行扩展置换、异或操作、S 盒变换和 P 盒置换，如图 2 所示.

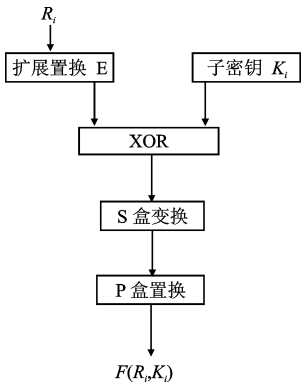

图 2　F 函数操作流程

图 2 中，S 盒变换是使用 8 个六进四出的 S 盒，进行算法中唯一的非线性变换[5]，它极大提高了算法的安全性. 每个 S 盒实际上就是一张 4 行 16 列的置换表. 每轮 F 函数使用的子密钥 K_i 由输入 64 bit 密钥 KEY 经过密钥置换 PC_1、循环左移和密钥置换 PC_2 产生，如图 3 所示.

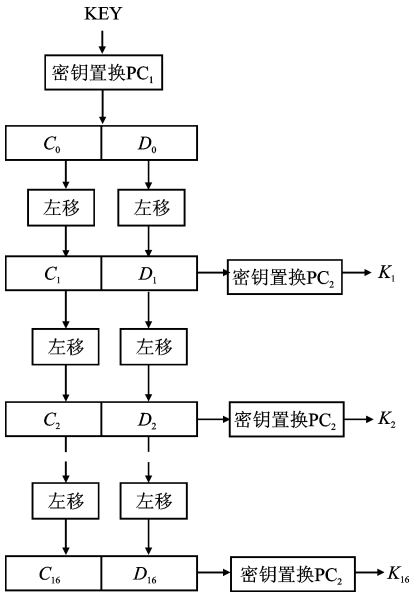

图 3　密钥拓展流程

DES 算法的解密过程和加密类似，不同之处仅仅在于子密钥的使用次序要反过来[6].

2.2　3DES 算法描述

3DES 算法是由 DES 算法改进而来，它的输入和输出同样为 64 bit，不同的是密钥长度为 192 bit（其中有效密钥长度为 168 bit），是由三个不同的 64 bit 密钥 KEY_1、KEY_2 和 KEY_3 构成. 如图 4 所示，加密时，64 bit 明文经过以 KEY_1 为密钥的 DES 加

密、以 KEY_2 为密钥的 DES 解密和以 KEY_3 为密钥的 DES 加密后，得到 64 bit 的密文输出. 解密时与加密过程相反.

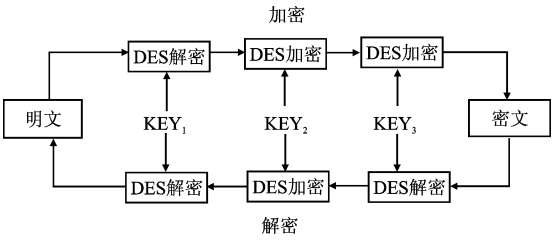

图 4 3DES 算法流程图

三次 DES 算法的进行和密钥长度的增加，使得 3DES 算法比 DES 算法具备更高的安全性. 三个密钥 KEY_1、KEY_2 和 KEY_3 是互不相同的，在安全性要求不太高的情形下，KEY_1 可以等于 KEY_3[7].

由于 DES 算法和 3DES 算法都属于对称加密算法，故都支持 ECB(电子密码本)和 CBC(密码分组链接)两种工作模式.

3 FPGA 设计实现

使用硬件来实现密码算法的基本流程如下：首先要用 Verilog 或者 VHDL 等硬件描述语言[8]对所设计的系统进行编码，然后使用 Modelsim、NC、VCS 等仿真工具进行功能仿真，进而根据设计需求，设计成专用集成电路(ASIC)或者现场可编程逻辑门阵列(FPGA). 进行 ASIC 设计时，所需的设计周期较长，灵活性差，而且设计费用较高[1]. 而进行 FPGA 设计时，设计者只需对芯片内部单元进行配置即可完成设计，设计周期较短，而且可以根据设计需求，灵活改变芯片配置以实现不同的功能，这也大大节省了设计费用. 本设计采用 FPGA 来实现 3DES 密码算法，使用的是 Altera 公司的 Quartus II 13.0 软件.

在传统 3DES 算法的实现方案中，每次迭代运算都需要消耗一个时钟周期，一次加解密操作共需 48 个周期，运算速度较慢. 本文提出循环迭代结构的方案，设计出多轮迭代运算的电路，只需较少的时钟周期即可完成一次加解密操作，有效提高了加解密速度.

3.1 设计方案的选取

本设计的目标是硬件实现高性能的 3DES 算法，即在尽可能减少 FPGA 资源消耗的前提下，最大限度地提高吞吐率，即加解密速度. 由算法描述部分可知，DES 算法的主要数据通路是 16 轮迭代运算，而 3DES 算法则是 48 轮迭代运算，所以设计方案的关键是如何高效地硬件实现这 48 轮迭代运算.

主流的 3DES 对称密码算法的硬件实现方案有两种. 一种是流水线结构设计[3]，即把这 48 轮迭代运算结构完全展开，形成一个 48 级的流水线. 每过一个时钟周期，就计算出一个加解密结果，这种设计能达到较高的吞吐率，但是使用的电路面积也是很可观的. 特别地，可以设计出只有 3 级流水线的 DES 模块级联结构(每级流水线完成 16 轮迭代运算)，或者只有 6 级流水线的结构(其中每级流水线完成 8 轮迭代运算)等方案，这些方案可以减少部分电路面积的消耗，但是电路面积依然很大. 更为重要的是，在当今对称密码算法的很多应用场合中，除了支持电子密码本(ECB)工作模式外，还需支持安全性更高的密码分组链接(CBC)工作模式. CBC 模式要求在加密当前明文分组时，必须先将其与前一明文分组的加密结果进行异或后，再将异或结果送至加密模块，这就要求将数据通路末端的结果送回首端，而且在前一明文分组加密完成前，无法进行当前分组的加密. 流水线结构要求每个时钟周期都有新的明文分组输入，显然无法实现 CBC 模式. 本设计要求 3DES 算法能同时支持 ECB 和 CBC 两种工作模式，故流水线结构不予采用.

另一种实现方案是循环迭代结构设计. 后者将 n 轮迭代运算的组合电路展开(其中 n 是 48 的因子，该电路称为“n 合一电路”)，每个时钟周期将 n 轮运算的结果送回输入端，实现对迭代结构的循环利用，这样经过若干个周期后即可得到 48 轮迭代运算后的加解密结果. 这种实现方案能有效地提高吞吐率，节省电路面积，而且兼容了 ECB 和 CBC 两种工作模式，故本设计采用循环迭代结构.

3.2 密钥拓展模块的设计

密钥拓展部分是由外部输入的 64 bit 密钥 KEY，得出每轮迭代的子密钥 K_i 的运算电路. 它是独立于 3DES 轮迭代运算的，可以提前产生，以节省每轮运算等待子密钥的时间.

由图 3 的密钥拓展流程可知，最直接的设计思路是将输入的密钥 KEY，经过密钥置换 PC_1 电路，16 轮循环左移电路和密钥置换 PC_2 电路，产生 16 个子密钥 K_i. 但是综合结果显示，这种设计不仅操作繁琐，而且产生了一些寄存器，电路面积相对较大. 由于密钥拓展过程是简单的置换过程，输出子密钥 K_i 各位的值直接来自于输入密钥 KEY 的各位，可事先算好其对应关系，采用直接赋值的方法，使输

入密钥 KEY 的各位直接对应上 16 个子密钥 Ki 的各位. 它操作简单，综合的电路面积较小，本设计采用这个方案.

由于 3DES 算法输入三个密钥 KEY_1、KEY_2、KEY_3，根据轮迭代运算的次序，需要先后进行三次密钥拓展. 本设计采用三个密钥分时复用一个密钥拓展模块的方案，通过系统的控制信号 roundsel 和加解密指示信号 encrypt，来选择当前时刻对哪个密钥进行拓展操作. 分时复用的方法有效减小了电路面积，有利于提高系统性能. 在密钥拓展模块内部，通过直接赋值产生 16 个 48 bit 的子密钥 $k[0]$，$k[1]$，$k[2]$，…，$k[15]$用于加解密. 每个时钟周期输出其中四个子密钥 k_a，k_b，k_c，k_d，输出顺序根据系统控制信号 roundsel 和内部加解密指示信号 des_encrypt 通过表达式的方法实现. des_encrypt 为 1 时加密，k_a，k_b，k_c，k_d 首先输出 $k[0]$，$k[1]$，$k[2]$，$k[3]$，下个时钟周期输出 $k[4]$，$k[5]$，$k[6]$，$k[7]$，依此类推；des_encrypt 为 0 时解密，k_a，k_b，k_c，k_d 首先输出 $k[15]$，$k[14]$，$k[13]$，$k[12]$，下个时钟周期输出 $k[11]$，$k[10]$，$k[9]$，$k[8]$，依此类推. 每次输出两个子密钥，源于系统的二合一电路结构，本设计选取二合一循环迭代结构的原因见下文. 子密钥输出顺序的实现代码如下所示.

```
assign k_a = k[a];
assign k_b = k[b];
assign a = (des_encrypt) ? ({roundsel[2:0],1'b0}) : ({~roundsel[2:0],1'b1});
assign b = (des_encrypt) ? ({roundsel[2:0],1'b1}) : ({~roundsel[2:0],1'b0});
```

3.3　S 盒的设计

由算法描述部分可知，每轮进行迭代时都要进行 F 函数的运算，这也是整个 3DES 算法中最繁琐、最可能增加延时的部分. F 函数中的扩展置换 E 和 P 盒置换很容易通过硬连线的方式实现，关键在于如何设计 8 个 S 盒. 每个 S 盒作为六进四出的非线性结构，是一张 4 行 16 列的置换表，通过查表的方式提高了算法的安全性.

S 盒的实现方法主要有查表法(Look-Up-Table)、双重 case 法、表达式法、双重表达式法和 ROM 实现法. 查表法是最直接的方法，就是通过一个 case 语句，将输入的 64 种(2 的 6 次幂)情况一一列举，分别赋予 4 bit 的输出值. 双重 case 法根据 4 行 16 列置换表的特点，使用双重 case 结构，先对行进行筛选，再对列进行筛选. 表达式法以输入 6 bit 为自变量，输出 4 bit 中的每一位为因变量，利用卡诺图进行化简，得到 S 盒的表达式. 双重表达式法类似于双重 case 法，先用 case 语句对行进行筛选，再对每行进行表达式化简. 由于每行只有 16 个元素，表达式只有四个自变量，故双重表达式法相对于表达式法更加简洁. ROM 实现法是将 S 盒中的数据寄存于 FPGA 内部自带的 ROM 中，每次需要时通过读取 ROM 来获取数据，这样会消耗一定数量的 M9K 资源. 分别使用以上五种 S 盒实现方法，在 Quartus II 13.0 软件上选用器件 EP4SGX530NF45C3 对算法进行综合，结果如表 1 所示.

表 1　五种 S 盒实现方法的综合结果

S 盒实现方法	延时/ns	吞吐率/(Mb/s)	面积/ALUTs	M9K/个数
查表法	3.64	703.3	648	0
双重 case 法	3.73	686.3	664	0
表达式法	3.61	709.1	650	0
双重表达式法	3.65	701.4	672	0
ROM 实现法	4.51	567.6	732	8

由表 1 可知，查表法、表达式法比双重 case 法、双重表达式法具有更小的延时和更高的吞吐率，对 ALUTs 资源的消耗也更小. 与查表法相比，表达式法具备更小的延时，性能略优于查表法. 而 ROM 实现法虽然使用了 M9K 资源，但是 ALUTs 依然很大，而且延时增大到 4.51 ns，致使吞吐率明显减小，不宜采用. 综上所述，本设计采用表达式法对 S 盒进行设计，能够获得更高的性能.

3.4　整体结构设计

由上文可知，在设计方案选取时，本设计采用的是循环迭代结构来实现 3DES 算法. 在该方案中，通过循环使用展开的 n 轮迭代运算的“n 合一电路”，经过 $48/n$ 个周期后，得到 48 轮迭代运算结果，如图 5 所示.

由于本设计还需要支持 16 轮迭代运算的 DES 算法，故 n 必须同时是 16 和 48 的因子，可取 1，2，4，8，16. 针对以上五种情况，分别设计出相应的一合一、二合一、四合一、八合一、十六合一电路，经过 Modelsim SE 10.0a 软件的功能仿真验证后，在 Quartus II 13.0 软件上选用器件 EP4SGX530NF45C3 对电路进行综合，结果如表 2 所示.

在表 2 中，每种循环迭代结构所需的周期数为执行 48 轮迭代运算所需周期数加上 1，其中 1 为将

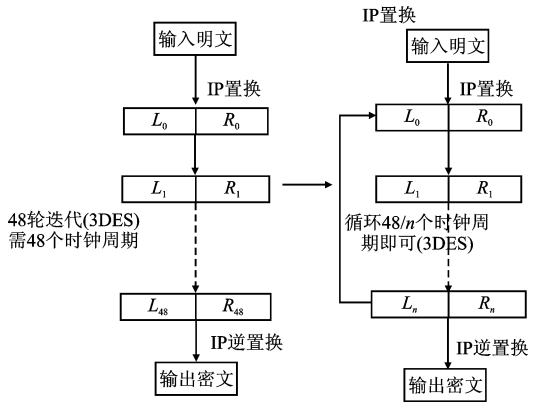

图 5 n 合一循环迭代结构

加解密结果打到输出寄存器所需的时钟周期. 可以

表 2 不同循环迭代结构的综合结果

循环迭代结构	延时/ns	吞吐率/Mbps	面积/ALUTs	周期数
一合一	3.00	435.4	609	49
二合一	3.61	709.1	650	25
四合一	5.90	834.4	779	13
八合一	10.50	870.7	1 222	7
十六合一	20.41	783.9	2 152	4

看出,一合一结构的延时虽然较小,但是由于周期数大,系统的吞吐率较低,不宜采用. 虽然八合一和十六合一结构的吞吐率都达到了 700 Mb/s 以上,但是由于迭代展开的幅度较大,导致延时和电路面积都很大,单位面积的吞吐率性能较低,不宜采用. 二合一和四合一结构都达到了较高的吞吐率性能,而且使用的电路面积较小,由于二合一结构的单位面积吞吐率性能 1.091 Mb·s^{-1}·$ALUT^{-1}$大于四合一结构的 1.071 Mb·s^{-1}·$ALUT^{-1}$,而且时钟频率也较高,故本设计采用二合一循环迭代结构,如图 6 所示.

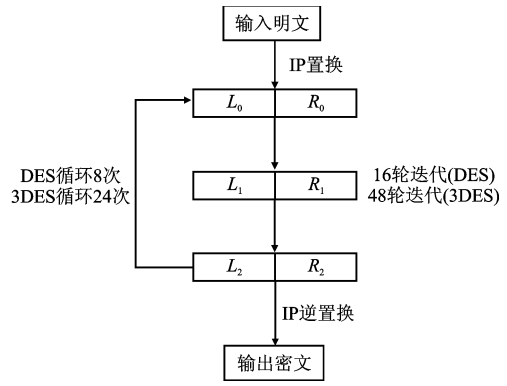

图 6 二合一循环迭代结构

通过以上分析,采用模块化设计来构建 3DES 算法实现的整体结构. 系统主要由有限状态机 FSM、密钥拓展模块 KEY_EXTEN 和二合一循环迭代的数据通路组成,如图 7 所示. FSM 通过信号 roundsel 来控制状态转换,是整个系统的控制核心. 根据 3DES 的迭代轮数和加解密情况,KEY_EXTEN 模块从外部输入的 KEY_1、KEY_2、KEY_3 中选择相应的初始密钥进行拓展操作,得到每轮迭代所需的子密钥,其中运行 DES 算法时以 KEY_1 为初始密钥. 由于采用二合一循环迭代结构,故每次 KEY_EXTEN 模块输出两个子密钥. 在信号 roundsel 控制下,数据通路的输入为 64 bit 的 TEXTIN,经过两轮迭代运算后得到 L_2、R_2,再将其送回 L_0、R_0 处,在下一时钟周期循环迭代运算,直至得到最后 64 bit 的加解密结果 TEXTOUT. 每轮迭代过程中,S 盒采用的都是表达式法.

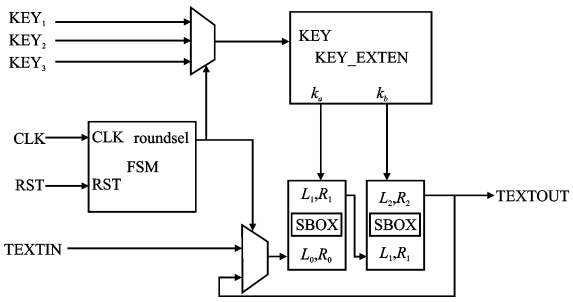

图 7 3DES 算法整体结构图

4 硬件实现结果

在使用 Verilog 对系统完成设计后,要对其进行功能验证和综合,选用 Modelsim SE 10.0a 软件对 Verilog 模型进行验证,选用 Quartus II 13.0 软件来综合,进行性能分析.

4.1 功能验证

为了突出各种仿真情形的典型性,本文选用在 CBC 模式下,使用 3DES 算法进行加解密的波形. 外部输入的三个初始密钥分别为:

KEY_1=64'h0123456789abcdef,

KEY_2=64'h23456789abcdef01,

KEY_3=64'h456789abcdef0123.

CBC 模式下,还需要外部输入初始化向量 IV,IV= 64'h0123456789abcdef. 加密时,输入明文 TEXTIN=64'h167e47ec24f71d63,经过 25 个时钟周期后,输出信号 out_ready 为高电平,输出密文 TEXTOUT=64'hdea16606621225f9,如图 8 所示.

解密时,在同样的初始密钥和初始化向量下,输

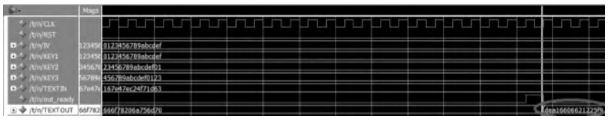

图 8　CBC 模式下 3DES 加密仿真波形

入密文 TEXTIN＝64′hdea16606621225f9，同样经过 25 个周期后 out_ready 信号为高电平，输出明文 TEXTOUT＝64′h167e47ec24f71d63，如图 9 所示，仿真结果正确无误.

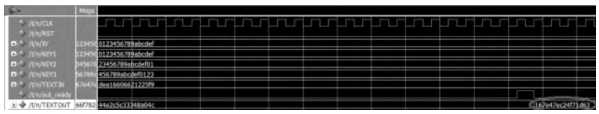

图 9　CBC 模式下 3DES 解密仿真波形

4.2　性能分析

在 Altera 公司的 Quartus II 13.0 软件上，选用 Stratix IV 系列的器件 EP4SGX530NF45C3 对电路进行综合，一次加解密操作需要 25 个时钟周期，3DES 吞吐率达到了 709.1 Mb/s，占用的面积只有 650 ALUTs，综合结果和性能见表 3.

表 3　FPGA 综合结果和性能

延时	频率	吞吐率	面积	周期
3.61/ns	277/MHz	709.1/(Mb/s)	650 ALUTs	25

对比 IP 设计公司 Helion[9] 和 Cast[10] 的同类设计，表 4 中列出了 3DES 算法的延时、吞吐率和电路面积，以及本设计在相同的器件下对应的性能指标. 可见，在同样的条件下，相比于 Helion 公司的方案（参考文献中的面积单位 ALM 已换算为 ALUT），本设计方案虽然吞吐率较低（661 Mb/s＜724 Mb/s），但是使用面积也明显减小（591 ALUTs＜742 A LUTs），单位面积吞吐率性能 1.118 Mb·s^{-1}·$ALUT^{-1}$ 高于 Helion 公司的 0.976 Mb·s^{-1}·$ALUT^{-1}$；相比于 Cast 公司的方案，本设计方案具有更短的延时，更高的吞吐率和更小的电路面积，有效提升了性能.

5　结束语

针对 3DES 算法的 FPGA 实现方案，本文提出了二合一的循环迭代结构，在同时支持 ECB 和 CBC 两种模式的前提下，实现了较高的性能，可广泛应用于商业数据加密、网络安全产品等领域. 对比其他同类设计，本文提出的方案吞吐率较高、电路面积较小，实现了速度与面积的综合权衡，是一个高性能的 FPGA 实现方案.

表 4　不同 FPGA 实现方案的性能比较

设计	器件	延时/ns	吞吐率/Mb/s	面积/ALUTs
Helion 参考文献[2]	EP2AGX190FF35C4	-	724	742
本设计	EP2AGX190FF35C4	3.87	661	591
Cast 参考文献[3]	EP3SL50F484C2	3.66	364	709
本设计	EP3SL50F484C2	3.02	847	591

参考文献：

[1] 常少卿，任芳. 基于 FPGA 的 3DES 加密系统的设计与实现[J]. 现代电子技术，2011，34(18)：114-120.

[2] 邵金祥，何志敏. 基于 FPGA 的 3DES 加密算法高速实现[J]. 现代电子技术，2004，27(21)：55-57.

[3] 黄本雄，鲍跃魁，胡海. 3-DES 算法的一种硬件实现[J]. 计算机与数字工程，2005，33(8)：86-89.

[4] 柳沐璇，张树丹，唐彩彬. 一种基于 AHB 总线的 DES IP 核设计[J]. 微电子学与计算机，2014，31(10)：69-71.

[5] 惠越超，汪一鸣. 基于 S 盒优化的轻量级加密算法设计[J]. 通信技术，2010，43(5)：103-108.

[6] 吴筱，郭培源，何多多. DES 和 SM4 算法的可重构研究与实现[J]. 计算机应用研究，2014，31(3)：853-856.

[7] 张元金. 3DES 分组加密算法模型分析[J]. 计算机与数字工程，2014，42(8)：1468-1471.

[8] 夏宇闻. Verilog 数字系统设计教程[M]. 2 版. 北京：北京航空航天大学，2008.

[9] Helion Technology Limited. Des & 3DES cores[EB/OL]. [2014-11-10]. http://www.heliontech.com/des.htm.

[10] Cast Corporation. Triple data encryption standard corel [EB/OL]. [2014-10-19]. http://www.cast.inc.com/ip∑cores/encryption/des3/index.html.

作者简介：

朱欣欣　男，(1990-)，硕士. 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：490239345@qq.com.

李树国　男，(1963-)，教授，博士生导师. 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.