轻量级密码算法 LBlock 的 FPGA 优化实现①

邹 祎, 李 浪, 贺位位, 许琼方

(衡阳师范学院 计算机科学系, 衡阳 421002)

摘 要: LBlock 密码算法是我国学者吴文玲和张蕾在 ACNS2011 提出的轻量级分组加密算法. 论文对 LBlock 加密算法的硬件优化实现进行了研究, 一方面将相同运算用一个模块设计完成, 通过主程序重复调用完成加密; 另一方面将轮操作和密钥更新放在同一个模块中并行执行, 而且使用相同寄存器完成 S 盒变换和密钥变换, 这样既可以不影响加密速度, 又不需要将密钥更新中间结果另存, 有效地节省寄存器的使用开销. 然后分模块进行实现并仿真实验, 和进行整体正确性实验验证. 通过实验, 验证论文所用优化方法可以较大幅度减少 LBlock 密码算法的实现面积, slices 占用比减少了 14%, LUT 占用比减少了 32%. 在 VIRTEX 5 下的系统吞吐率为 14.53Gb/s, 更能有效满足较小芯片面积的应用需求, 给当前的物联网加密提供参考.

关键词: LBlock 算法; Verilog HDL; FPGA 实现

Optimal Implementation of Lblock on FPGA

ZOU Yi, LI Lang, HE Wei-Wei, XU Qiong-Fang

(Department of Computer Science, Hengyang Normal University, Hengyang 421002, China)

Abstract: LBlock is a lightweight block cipher designed by Wu Wen-ling and Zhang Lei in ACNS 2011. In this paper, the optimal implementation of LBlock encryption algorithm in hardware is studied. Firstly, the same operation is realized in one module, then the main program calls several times to complete the encryption, especially 32 in Block. Secondly, the same register is used in the subcell and key transformation, the cipher round operation and key update is designed in same module, so that it can release the number of the registers and accelerate the speed of implementation. Last, the sub-modules are combined into a complete program, the correctness of Lblock is tested. It can greatly reduce the area of LBlock algorithm by the optimal method, which can meet the application requirements of smaller chip area and provide reference for the further application of IOT encryption.

Key words: LBlock; Verilog HDL; FPGA implementation

近年来, 随着物联网, 无线传感技术的广泛应用, 如何将加密算法用尽可能小的面积实现, 同时又要保证加密效率的问题成了研究热点, 因此, 轻量级分组密码应运而生并不断发展完善. 近年来提出了一系列轻量级分组密码算法, 有 CHES 2006 年提出的 HIGHT[1]、CHES 2007 年提出的 PRESENT [2]、EUROMICRO 2008 提出的 PUFFIN[3]、CANS 2009 提出的 MIBS[4]、CHES 2011 提出的 PICCOLO[5]和 LED [6], 而 LBlock[7]密码算法是我国学者吴文玲和张蕾在 ACNS2011 提出的.

LBlock 密码算法是 32 轮类 Feistel 结构的轻量级分组密码. 设计上仍采用与传统分组密码类似的迭代结构, 将明文用轮函数在密钥的作用下进行多次迭代得到密文; 密钥扩展算法的设计借鉴了 PRESENT 算法的设计理念, 采用非线性移位寄存器, 利用 S 盒变换和循环移位生成轮密钥[8].

由于 LBlock 密码算法提出时间不长, 目前国内外相关研究文献很少, 因此对于 LBlock 密码算法的硬件优化实现研究具有一定的价值, 可以更好地促进其在面向资源约束的智能卡上的应用[9,10].

① 收稿时间:2014-11-06;收到修改稿时间:2015-01-12

1 LBlock轻量级密码算法简介

LBlock 算法属于 Feistel 结构的分组加密算法, 分组规模为 64 位, 密钥长度为 80 位, 迭代圈数为 32 轮. 其加密过程如图 1 所示.

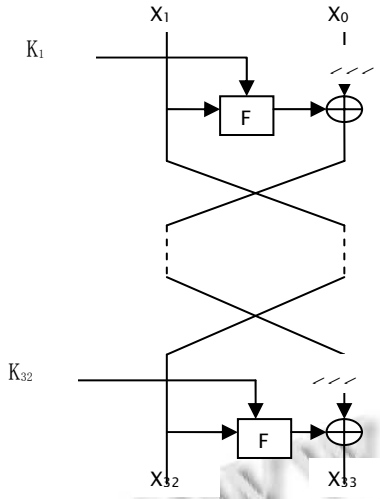

图 1 LBlock 加密过程结构图

其中, 明文 Plaintext 输入为 64 位, Plaintext= X1||X0, X1 等于明文左边的 32 位, X0 等于明文右边的 32 位; 密钥 key 输入为 80 位, K1 位 key 的最左边 32 位, Ki+1 等于第 i 轮密钥更新的最左边 32 位; 最后的密文 ciphertext 输出为 64 位即 ciphertext=X32||X33.

而 F 函数包括轮密钥加(AddKey), S 盒变换(SubCell), P 混合(Permutation)三个过程, 结构如图 2 所示.

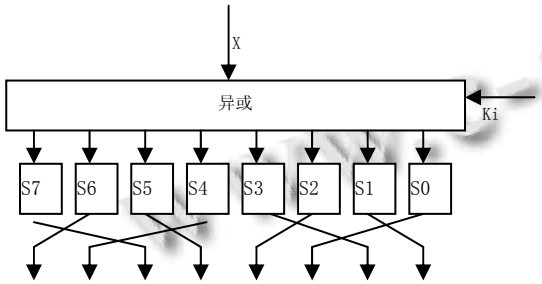

图 2 轮函数结构图

2 LBlock轻量级密码算法的Verilog HDL优化实现

LBlock 主要包括轮密钥加(AddKey), S 盒变换(SubCell), P 混合(Permutation)操作; 同时对密钥扩展 key 进行 S 盒变换(SubCell)、p 置换(Exchange)操作、与轮数异或. 在 LBlock 密码算法的 verilog HDL 实现上主要从面积和速度两个方面来考虑.

2.1 面积优化

在硬件实现上, 对于多次重复计算的过程可设计为一个器件反复使用, 从而节省资源的消耗. LBlock 的 32 轮运算中每轮运算的操作是相同的, 正好符合这个特点. 因此, 从面积上实现是将相同功能模块在硬件上只实现一次, 然后重复调用. 具体到 LBlock 密码算法, 首先在 LBlockRound 模块中把轮密钥加、S 盒变换和 P 混合在硬件上实现单轮加密, 然后通过在主程序(顶层模块)中, 采用面积优先方式, 即通过计数器控制模块 LBlockRound 重复调用 32 次来完成 32 轮运算, 从而实现加密过程. 这种方法可以大大降低硬件开销.

具体的优化方法如下:

(1)相同运算只实现一次, 主程序调用 32 次完成加密.

(2)将轮操作和密钥更新放在同一个模块中并行执行, 另外 S 盒变换和密钥变换使用同一寄存器, 这样既可以不影响加密速度, 又不需要将密钥更新中间结果另存, 有效地节省寄存器的使用开销.

(3) 利用时钟信号控制计数器更新, 完成加密仅需 32 个时钟周期.

其核心代码实现如下所示:

```
module   LBlock(result,state,key,clk);//计数器初始化
always @(posedge   clk)   begin
cnt      <= (cnt^32)? cnt+1: cnt;//计数
res<=ready? ( (cnt)?{te,res[63:32]}:state):res;
k <= ready ? ( (cnt)?t:key):k; //密钥更新
ready   <= (cnt^32)? 1:0;//重复 32 次结束
end
LBlockRound   LR(te,t,res,k,cnt);
//单轮加密运算
//输出 result 值
endmodule
```

2.2 速度优化

加密速度优先方法一般采用全流水进行密码算法实现. LBlock 的 32 轮运算中每轮运算的操作是相同的, 如果不考虑芯片面积, 可将所有 32 个轮运算模块均以硬件实现. 在这种实现方式下, 每一个数据块在完成一次轮运算后, 它可以立即开始下一级流水线的下一轮计算. 不考虑其它因素, 此种流水线实现方式在每个时钟周期有 32 个数据块同时串行处理, 这使得整个

加密运算没有额外的等待时间. 理论上, 这种方法可使数据处理速度比非流水线提高 32 倍, 但它需要大量的硬件资源, 即芯片面积成相应倍增加.

由于 LBlock 是轻量级加密算法, 因此本文在优化实现上主要考虑从面积上实现.

2.3 优化实现流程

在 LBlock 的 32 轮运算中, 从面积上优化的实现方案, 将每轮运算单独模块实现, 称为单轮加密模块. 单轮加密模块内分别实现轮密钥加、S 盒变换和 P 混合三个运算. 单轮加密运算完后, 硬件上不用再提供多余开销, 只需依次重复执行 32 次单轮加密模块, 即可完成整个加密运算, 得到密文. 其实现流程如图 3 所示.

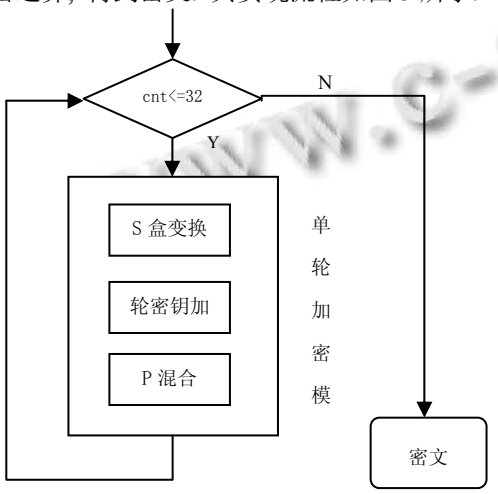

图 3　LBlock 加密算法优化实现流程图

其中计数器 cnt 在 clk 信号的控制下, 每一个周期增 1, 因此, 32 个周期后完成 LBlock 加密算法的所有轮运算, 得到密文.

2.4 优化实现的核心代码与验证

LBlock 单轮运算的核心代码如下所示:

```
module  LBlockRound(res,Up_k,s,k,r);
initial   begin // 初始化 Sbox
s0[ 0]=4'hE;s0[ 1]=4'h9;s0[ 2]=4'hF;s0[ 3]=4'h0;s0
[ 4]=4'hD;s0[ 5]=4'h4;s0[ 6]=4'hA;s0[ 7]=4'hB;
……  //(中间变换代码略去)
end
//完成五个操作: AddKey, SubCell , Permutation,
X0 循环左移, X1 与 X0 异或
assign res={
s6[s[59:56]^k[75:72]]^s[23:20],s4[s[51:48]^k[67:64
]]^s[19:16],
……
};  //完成单轮计算的所有代码
assign  Up_k={      //密钥更新
s9[k[50:47]],s8[k[46:43]],k[42:22],k[21:17]^r[4:0],k
[16:0],k[79:51]
};
endmodule
```

实验中用于加密验证的数据如下:

原文(s):0123456789abcdef

密钥(k): 0123456789abcdeffedc

单轮(取第 1 轮数据)加密后:

加密密文: 151FF1FD

更新密钥: 673579BDBFDB802468AC

利用 modelsim 6.1f 的仿真结果如图 4 所示.

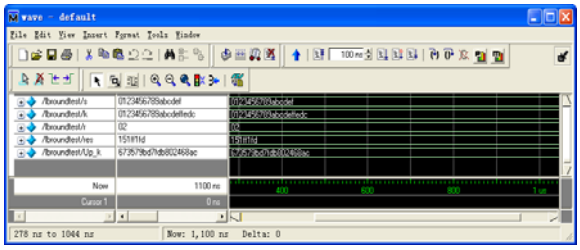

图 4　LBlock 单轮运算仿真实验结果

由图 4 中仿真结果可证, 单轮运算结果与预期相同, 单轮运算模块正确.

顶层模块中, 采用面积优先方式, 即通过计数器控制模块 LBlockRound 调用的次数, 实现加密过程. 核心代码如上 D1-1 所示.

实验中用于加密验证的数据如下:

原文(state):0123456789abcdef

密钥(key): 0123456789abcdeffedc

密文(result):4b7179d8ebee0c26

利用 Modelsim6.1f 仿真结果, 如图 5 所示.

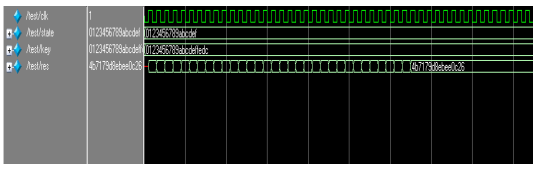

图 5　LBlock 仿真实验结果

从图 5 的实验结果可见, 本文的优化实现后加密运算结果是正确的.

3 基于EDK的FPGA验证及性能评价

嵌入式开发套件(EDK)是用于设计嵌入式可编程系统的全面解决方案. 该套件包括嵌入式软件工具(Platform Studio)以及嵌入式 IBM PowerPC 硬件处理器核和/或 Xilinx MicroBlaze 软处理器核进行 Xilinx 平台 FPGA 设计时所需的技术文档和 IP. EDK 自带了许多工具和 IP, 可以用来设计完整的嵌入式处理器系统, 主要包括 Xilinx 平台工作室 XPS 和软件开发套件 SDK.

本设计选用的 FPGA 器件是 XILINX VIRTEX 5 XC5VLX50T, 首先, 完成 LBlock 加密算法的硬件描述; 然后利用 Modelsim6.1f 仿真, 验证通过; 最后, 在 EDK 13.2 平台下完成硬件平台的搭建与下载, 并结合 SDK13.2 验证其正确性.

在 EDK 平台上, 通过添加自定义 IP 核, 并下载至 FPGA 运行, 在超级终端显示的输出结果如图 6 所示, 由图中数据, 可得出代码描述与实现结果正确.

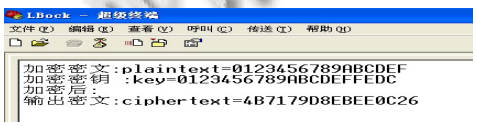

图 6 LBlock 超级终端输出结果图

为了对本文设计的 LBlock 优化实现方法在资源利用方面进行性能评价, 实验中将文中所述两种方式实现的代码进行下载, 其主要逻辑资源消耗情况对比, 如图 7 所示. 左图是从速度上实现的数据, 即未做优化的 LBlock 下载至 FPGA 上时, 占用 slices 为 3986, LUT 为 11004; 右图是从面积上实现的数据, 即优化的 LBlock 下载至 FPGA 上时, 占用 slices 为 2966, LUT 为 7557, 由此数据比较可知, 优化后的 LBlock 在面积资源占用上相比优化前有较大优势, slices 占用比减少了 14%, LUT 占用比减少了 32%. 进一步验证了本文提出的 LBlock 密码算法优化实现的优越性.

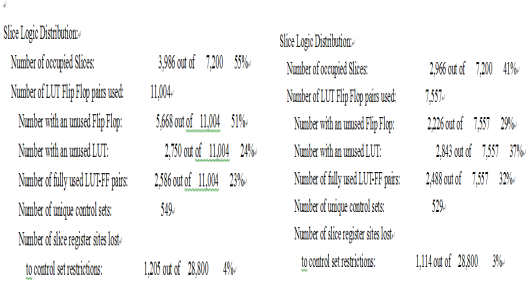

图 7 两种方式 slice 等资源利用对比图

```
Design statistics:
   Minimum period:   9.911ns (Maximum frequency: 100.898MHz)
   Maximum path delay from/to any node:   4.578ns
   Maximum net delay:   0.805ns
```

图 8 系统时钟频率

4 总结

本文对 LBlock 轻量级密码算法进行了面向面积优化的 FPGA 实现, 并通过实验下载, 验证了其正确性和在资源利用上的优势, 后续工作是对其安全性和效率做进一步的研究与分析.

参考文献

1 Hong D, Sung J, Hong S, et al. HIGHT: a new block cipher suitable for low-resource device. Cryptographic Hardware and Embedded Systems 2006, LNCS 4249. 2006. 46–59.

2 Bogdanov A, Knudsen LR, Leander G, et al. PRESENT: an ultra–lightweight block cipher. Proc. of Cryptographic Hardware and Embedded Systems. 2007. 450–466.

3 Cheng H, Heys H, Wang C. PUFFIN: A novel compact block cipher targeted to embedded digital systems. The 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools. University of Parma, Parma (Italy). 2008. 383–390.

4 Izadi M, Sadeghiyan B, Sadeghian SS, et al. MIBS: A new lightweight block cipher. CANS 2009. Kanazawa, Ishikawa, Japan, 2009. 334–348.

5 Shibutani K, Isobe T, Hiwatari H, Mitsuda A, Akishita T, Shirai T. Piccolo: An ultra-lightweight block cipher. Proc. of the CHES 2011. Nara, Japan. LNCS 6917. 2011. 342–357.

6 Guo J, Peyrin T, Poschmann A, Robshaw M. The LED block cipher. Proc. of the CHES 2011.Nara, Japan. LNCS 6917. 2011. 326–341.

7 Wu WL, Zhang L. Lblock: A lightweight block cipher. LNCS 6715. 2011. 327–344.

8 詹英杰,关杰,等.对简化版 LBlock 算法的相关密钥不可能差分攻击.电子与信息学报,2012,34(9):2161–2166.

9 Zhao L, Nishide T, Sakurai K. Differential fault analysis of full LBlock. COSADE 2012. LNCS 7275. 2012. 135–150.

10 Liu Y, Gu D, Liu ZQ, Li W. Impossible differential attacks on reduced–round LBlock. ISPEC 2012. LNCS 7232. 2012. 97–108.