

SNOW 3G 算法和 ZUC 算法在 FPGA 上的性能对比

韩 越 黑 勇 乔树山

(中国科学院微电子研究所, 北京 100029)

摘 要 SNOW 3G 算法和 ZUC 算法都是 3GPP LTE 安全性算法的核心, 已成为第四代移动宽带通信加密标准。SNOW 3G 算法由于提出时间较早, 现已被广泛采用。ZUC 算法是我国自主设计的流密码算法, 于 2011 年被采纳为国际加密标准。为了对比两种加密算法在 FPGA 上的性能, 在分析两种算法结构的基础上, 提出两种算法的 FPGA 设计方案并实现, 使用 Modelsim 软件进行仿真, 将仿真结果与 C 语言模型对比验证设计的正确性, 最后使用 Quartus II 软件进行综合。仿真和综合的结果表明, ZUC 算法的 FPGA 设计的数据吞吐率小于 SNOW 3G 算法的 FPGA 设计, 然而前者资源开销仅为后者的 1/3, 并且在功耗开销方面, 对于相同的数据吞吐率, 前者比后者小 63%。

关键词 SNOW 3G 算法 ZUC 算法 现场可编程逻辑器件 性能对比

中图法分类号 TP311.52; **文献标志码** A

目前, 许多场合都使用流式加密算法来保证数据的安全性, 尤其在无线通信领域。因此高校和工业界提出了大量的流式加密算法。与块式加密算法相比, 同步的流式加密算法可以避免误差传递的影响。LTE 是 3GPP 启动的高速无线数据通信标准计划。LTE 网络不仅满足了互联网应用的移动需求, 而且具有处理数据吞吐率爆炸的能力。为了在 LTE 中提供安全性保护, 3GPP 指定了三套安全性算法, 用来保护通信数据的机密性和完整性; 包括 128—EEA1/128—EIA1、128—EEA2/128—EIA2, 和 128—EEA3/128—EIA3。其中, 第二套算法基于 AES 算法, 是一种块式加密算法。另外两套算法分别基于 SNOW 3G 算法和 ZUC 算法, 它们都是流式加密算法^[1-3]。

SNOW 3G 算法由于提出时间早, 现已被广泛采

用。ZUC 算法是我国自主设计的流密码算法, 于 2011 年被采纳为国际加密标准, 近日已有基于 ZUC 算法的移动设备问世。为了对比两种加密算法在 FPGA 上的性能, 本文在分析两种算法结构的基础上, 提出两种算法的 FPGA 设计方案并实现, 使用 Modelsim 软件进行仿真, 将仿真结果与 C 语言模型对比验证设计的正确性, 最后使用 Quartus II 软件进行综合。仿真和综合的结果表明, ZUC 算法 FPGA 设计的数据吞吐率小于 SNOW 3G 算法的 FPGA 设计, 然而前者资源开销仅为后者的 1/3, 并且在功耗开销方面, 对于相同的数据吞吐率, 前者比后者小 63%。

1 SNOW 3G 算法和 ZUC 算法

SNOW 3G 算法和 ZUC 算法在结构上具有一定的相似性。两种算法的结构如图 1 和图 2 所示。首先, 两者都包含线性反馈移位寄存器 (LFSR) 模块。其次, SNOW 3G 结构中的有限状态机 (FSM) 和 ZUC 结构中的非线性函数 (F_function) 模块在功能上类似。最后, 两者的密钥生成过程均包括初始化阶段和密钥生成阶段。但是, 两者也存在着很大的区别, 在下文将详细描述。

2012 年 9 月 25 日收到, 10 月 31 日修改 核高基重大专项基金
(2009ZX01034-002-004-007(002)) 资助
第一作者简介: 韩 越, 女。中国科学院电子研究所二室硕士研究生。研究方向: 微电子学与固体电子学, 助理研究员。E-mail: hanyue@ime.ac.cn。

1.1 SNOW 3G 算法

SNOW 3G 算法分为两层结构,分别是 LFSR 和 FSM。如图 1 所示。它的工作模式包括初始化阶段和密钥生成阶段。在初始化阶段,LFSR 模块使用来自 FSM 模块的数据 F 进行异或运算,进而产生更新寄存器 S_{15} 的计算结果;在密钥生成阶段,LFSR 模块不再使用 F 。

LFSR 模块中包含 16 个 32 位宽的寄存器 ($S_0 \sim S_{15}$), 3 个异或运算单元, 1 个 MULalpha 运算单元, 和 1 个 DIValpha 运算单元。FSM 模块中使用来自 LFSR 模块的 S_{15} 和 S_5 的值。FSM 模块包含 3 个 32 位宽的寄存器 ($R_1 \sim R_3$), 2 个 32 位宽的加法运算单元, 2 个 32 位宽的异或运算单元, 以及对 B_1 、 B_2 的查表运算单元。FSM 模块产生的输出 F , 与来自 LFSR 模块的寄存器 S_0 进行异或运算, 产生 SNOW 3G 算法的 32 位宽密钥 $Z^{[4]}$ 。

1.2 ZUC 算法

ZUC 算法分为三层结构,分别是 LFSR, 比特重组 (BR) 和 F_function, 如图 2 所示。它的工作模式同 SNOW 3G 算法一样, 包括初始化阶段和密钥生成阶段。在初始化阶段, LFSR 模块使用来自 F_function 模块的输出 W 与来自 BR 模块的 X_3 的异或运算的结果 Z , 向右移一位值 $Z \gg 1$, 进行 $2^{31} - 1$ 模加运算, 进而产生更新寄存器 S_{15} 的计算结果; 在密钥生成阶段, LFSR 模块不再使用 $Z \gg 1$ 。

LFSR 模块采用定义在素域 $GF(2^{31} - 1)$ 的线性反馈移位寄存器, 包含 16 个 32 位宽的寄存器 ($S_0 \sim S_{15}$), 5 个乘法运算单元以及乘积结果的 $2^{31} - 1$ 模加运算单元。BR 模块是 ZUC 算法不同于 SNOW 3G 算法的模块。该模块将来自 LFSR 模块的 8 个寄存器的 128 比特数据重组成 4 组 32 位宽数据 $X_0 \sim X_3$ 。进一步破坏了 LFSR 在素域 $GF(2^{31} - 1)$ 上的线性结构, 使密码攻击变得更加困难。F_function 模块使用来自 BR 模块的 $X_0 \sim X_2$ 的值。F_function 模块包含 2 个 32 位宽的异或运算单元, 2 个 32 位宽的加法运算单元, 2 个 32 位宽的寄存器 (R_1 和 R_2), 1 个循环移位单元, 对 B_1 和 B_2 的查表运算单元, 以及对查表结果进行非线性运算的单元 (L_1 和 L_2)。F_function 模块的输出 W 与来自 BR 模块的 X_3 进行

32 位宽的异或运算, 得到 32 位宽密钥 $Z^{[5,6]}$ 。

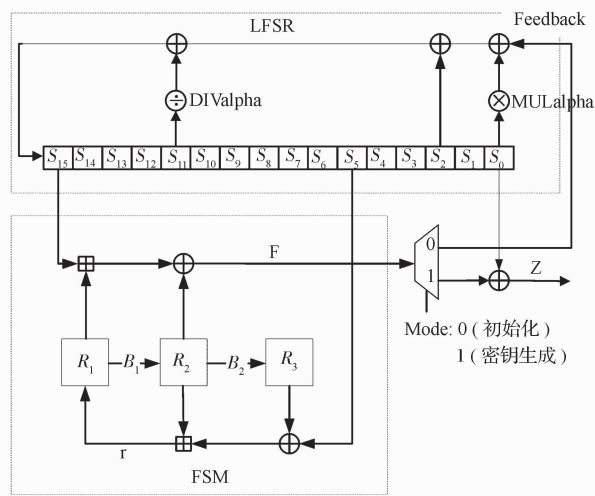


图 1 SNOW 3G 算法结构

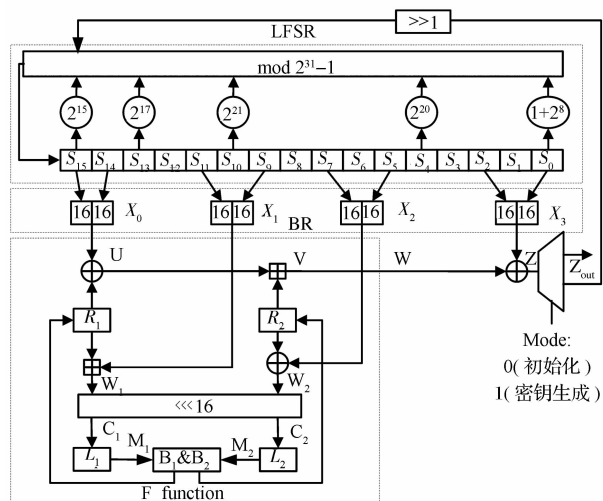


图 2 ZUC 算法结构

2 两种算法的 FPGA 设计方案

根据两种算法的结构, 本文提出了相应的硬件实现方案, 即 FPGA 设计方案。使用 Verilog HDL 语言进行编程实现。

对下文用到的特殊运算符进行如下定义:

“^”表示抑或运算;

“a<<k”表示 a 向左移 k 位;

“a>>k”表示 a 向右移 k 位;

“mod”表示取模运算;

“a||b”表示 a 和 b 的拼接, a 为高位;

“S[a:b]”表示取 S 的第 a 到 b 位, a 为高位;

“S[a]”表示输入为 a 时, S 对应的输出。

2.1 SNOW 3G 算法的 FPGA 设计方案

2.1.1 LFSR 模块

在初始化阶段,更新 S_{15} 的计算值 v 可以表示为:

$$v = (S_0 \ll 8) MUL_{\alpha} [S_0(31:24)] \times S_2(S_{11} \gg 8) DIV_{\alpha} [S_{11}(7:0)] F \quad (1)$$

在密钥生成阶段, v 值的计算不包括式(1)中的 F 项。式(1)中,函数 MUL_{α} 和 DIV_{α} , 均需要从一个递归运算得到。对于 FPGA 实现,复杂度较高,运算时间长。2 个函数均是输入 8 位数据,得到 32 位数据输出,因此可以使用查找表结构实现。FPGA 器件中有 RAM 模块,可以将查找表的内容存入 RAM 中。但是 RAM 中读出的数据在本时钟周期内是不可用的,在下一时钟周期才有效,这样就会降低系统的吞吐率。所以,本文中采用组合逻辑实现查找表结构。

2.1.2 FSM 模块

此模块需要完成两条较长路径的运算,分别包含 1 个 32 位宽加法运算和 1 个 32 位宽异或运算,计算得到寄存器 R_1 的更新值 r 和 FSM 模块的输出 F ,实现如下:

$$\begin{cases} F = (S_{15} + R_1) R_2 \\ r = R_3 S_5 + R_2 \end{cases} \quad (2)$$

B_1 和 B_2 是 2 个 32 位输入到 32 位输出的置换盒,输入 R_1 到 B_1 可以得到 R_2 的更新值;输入 R_2 到 S_2 可以得到 R_3 的更新值。将 B_1 和 B_2 分别表示成 4 个并置的 8 位输入到 8 位输出的置换子盒 ($B_1_T_0 \sim B_1_T_3$ 和 $B_2_T_0 \sim B_2_T_3$),然后把输入的 32 位宽数据 (w) 分解成 4 组 8 位宽数据 ($w_0 \sim w_3$),用 8 位宽数据访问子盒,再将输出结果组合成 32 位宽输出,实现如下:

$$\begin{cases} B_1 = B_1_T_0 || B_1_T_1 || B_1_T_2 || B_1_T_3 \\ B_2 = B_2_T_0 || B_2_T_1 || B_2_T_2 || B_2_T_3 \\ w = w_0 || w_1 || w_2 || w_3 \\ B_1[w] = B_1_T_0[w_0] || B_1_T_1[w_1] || B_1_T_2[w_2] || B_1_T_3[w_3] \\ B_2[w] = B_2_T_0[w_0] || B_2_T_1[w_1] || B_2_T_2[w_2] || B_2_T_3[w_3] \end{cases} \quad (3)$$

式(3)中的 $B_1_T_0 \sim B_1_T_3$, $B_2_T_0 \sim B_2_T_3$, 8 个 8 位

输入到 8 位输出的子盒,与 LFSR 模块中的 MUL_{α} 、 DIV_{α} 一样,使用组合逻辑实现查找表结构。

2.2 ZUC 算法的 FPGA 设计方案

2.2.1 LFSR 模块

在初始化阶段,更新 S_{15} 的计算值 k 包括 $Z \gg 1$ 参与 $2^{31} - 1$ 模加运算;在密钥生成阶段,不使用 $Z \gg 1$ 。

LFSR 模块中的 $2^{31} - 1$ 模加运算单元实现,将 S_0 、 S_4 、 S_{10} 、 S_{13} 和 S_{15} 分别乘以不同的权值后,进行 $2^{31} - 1$ 模加。伪码表示为:

$$k = [2^{15} S_{15} + 2^{17} S_{13} + 2^{20} S_4 + (1 + 2^8) S_0] \bmod (2^{31} - 1) \quad (4)$$

注意式(4)中模加运算的模不是 2 的整数次幂,因此在不增加位宽的前提下,输出不能直接由输入相加得到。可以增加输出位宽,或设置进位变量,但由于上述 5 个寄存器乘以的权值非常大,这种方法开销太大。最后我们采用的方法是从较低位的寄存器 (S_0) 开始进行乘以 2 的 m 次幂,就是将 S_0 左移 8 位的值与 S_0 右移 (31—8) 位的值进行或运算,然后再取结果的低 31 位与原来的 S_0 相加,这样就完成了 $S_0(2^8 + 1)$ 的 $2^{31} - 1$ 模加运算。按照此方法依次完成其他四个寄存器的模加运算。 $LFSR$ 中的模加运算是此算法的关键路径,为提高系统吞吐率,可以将上述的每一组模加运算中的串行的两个加法器并置,以 $(1 + 2^8 S_0) \bmod (2^{32} - 1)$ 为例,如图 3 所示,实现过程为将 S_0 左移 8 位的值和 S_0 右移 23 位的值进行或运算,然后取或结果的低 31 位与 S_0 进行相加,得到加和结果和进位,如进位为 1,总的运算结果为加和结果再加 1,否则,总的运算结果为加和结果。

2.2.2 BR 模块

通过连线重组比特,实现如下:

$$\begin{cases} X_0 = \{S_{15}[30:15], S_{14}[15:0]\} \\ X_1 = \{S_{11}[15:0], S_9[30:15]\} \\ X_2 = \{S_7[15:0], S_5[30:15]\} \\ X_3 = \{S_2[15:0], S_0[30:15]\} \end{cases} \quad (5)$$

2.2.3 F_function 模块

分别需要计算出:

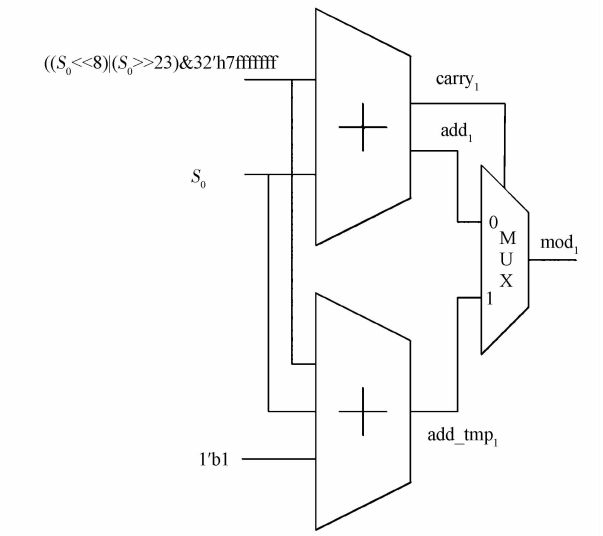


图3 LFSR 模块中的模加运算

$$\begin{cases} W = (X_0 R_1) + R_2 \\ W_1 = R_1 + X_1 \\ W_2 = R_2 X_2 \\ Z = W X_3 \\ C_1 = \{W_1[15:0], W_2[31:16]\} \\ C_2 = \{W_2[15:0], W_1[31:16]\} \end{cases} \quad (6)$$

式(6)中得到的 C_1 和 C_2 进行循环移位后,得到 M_1 和 M_2 ,然后以 M_1 和 M_2 为地址对置换盒 B_1 和 B_2 进行寻址。因为 32×32 的置换盒 B 可以表示成 4 个 8×8 的置子换盒,它的实现过程与式(3)类似,只需将 w 替换成 U_1 和 U_2 即可。特别地,ZUC 算法的置换盒子具有: $B_1 = B_2, B_{1_T_0} = B_{1_T_2}, B_{1_T_1} = B_{1_T_3}$ 。为实现资源优化,置换子盒可以分时复用,在每个时钟周期选择一个寻址逻辑对 $B_{1_T_0}$ 和 $B_{1_T_1}$ 进行访问,分四个时钟周期完成 32 位宽的数据输出。

3 仿真结果与分析

根据两个算法的结构,使用 Visual C/C++ 软件建立 C 语言模型,用来验证 FPGA 设计的正确性。使用 Modelsim 软件对 RTL 代码进行仿真,输入同样的初始密钥和初始向量,仿真结果表明 C 语言模型与 FPGA 设计的结果完全一致。

3.1 ZUC 算法的优化

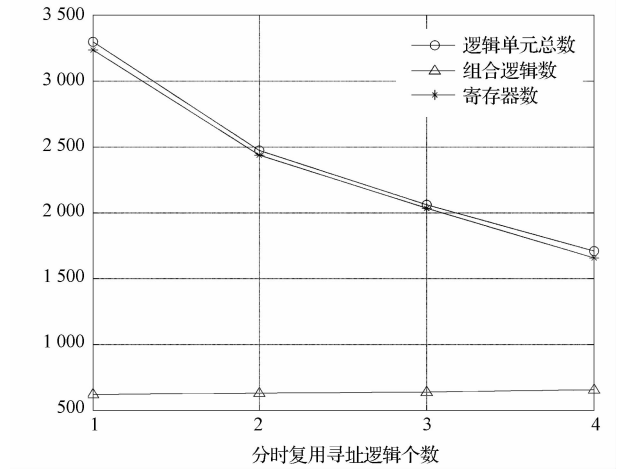


图4 ZUC 算法 FPGA 的综合结果

如上文所述,ZUC 算法可以分时复用 $B_{1_T_0}$ 和 $B_{1_T_1}$ 来减少算法的硬件资源开销。使用 Quartus II 软件对分时复用不同个数寻址逻辑的 FPGA 设计进行综合,选择的器件为 Cyclone II 系列 EP2C15AF484A7,仿真结果如图 4 所示。

图 4 中,分时复用逻辑个数为 1 时代表没有分时复用。从图 4 可以看出,分时复用 4 个寻址逻辑的 FPGA 设计具有最少的资源开销,与未使用分时复用的 FPGA 设计相比,可以降低系统资源开销 45%。分时复用寻址逻辑的实质是分时复用置换盒 $B_{1_T_0}$ 和 $B_{1_T_0}$,减少了构成查找表的组合逻辑单元数,与图 4 中组合逻辑数的变化趋势相符。

3.2 两种算法的 FPGA 性能对比

从图 3 可以看出,通过并置加法器的方法可以减小模加运算的路径延时,提高系统工作频率,进而提高系统的吞吐率。将 ZUC 算法的 FPGA 设计根据面积优化、路径优化的不同情况,以及 SNOW 3G 算法的 FPGA 设计的综合结果进行比较,如图 5 所示。图 5 中,进行路径优化的 FPGA 设计(情况 1 和 3)具有较高的工作频率;面积最优化的 FPGA 设计(情况 1 和 2)具有最小的资源开销,即需要较少的逻辑单元个数。未进行任何优化的 ZUC 算法的 FPGA 设计(情况 4),具有最低的工作频率,最大的资源开销。SNOW 3G 算法的 FPGA 设计具有较大的资源开销,与 ZUC 算法的最佳设计(情况 1)相比,前者资源开销是后者的 3.5 倍,前者工作频率是

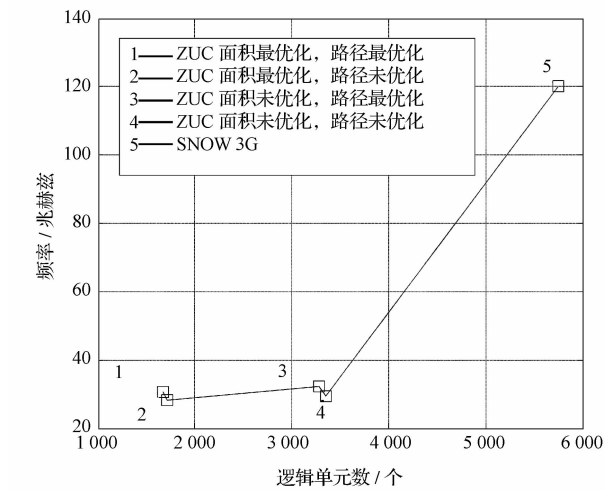


图5 两种算法的性能比较

后者的近 4 倍,即前者的吞吐率是后者的近 4 倍。

为了比较两种算法硬件设计的功耗开销,使用 Synopsys 公司的 Design Compiler 工具对两种算法的 RTL 代码进行 ASIC 综合。采用 0.18 μm 工艺,时钟频率设定为 40MHz,满足时序约束。得到的两种算法的功耗开销结果如图 6 所示。由图 6 可知,SNOW 3G 算法的总功耗开销是 ZUC 算法的 2.2 倍。因此在相同工作频率,相同数据吞吐率的情况下,ZUC 算法比 SNOW 3G 算法功耗开销小 63%。

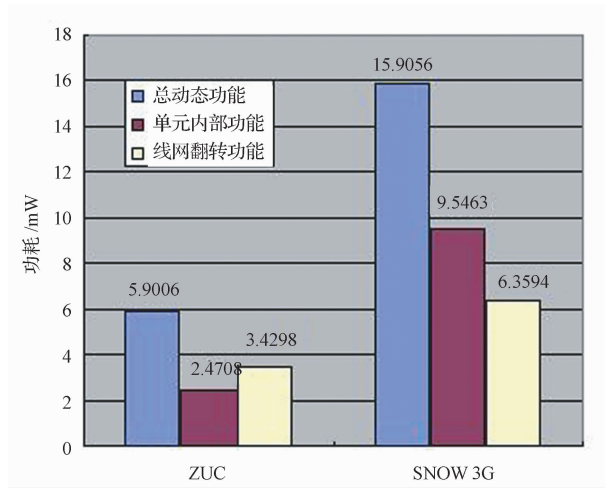


图6 两种算法的功耗比较

4 结束语

本文在分析两种算法结构的基础上,提出两种算法的 FPGA 设计方案并实现,使用 Modelsim 软件进行仿真,将仿真结果与 C 语言模型对比验证设计的正确性,最后使用 Quartus II 软件进行综合。仿真和综合的结果表明,ZUC 算法 FPGA 设计的数据吞吐率小于 SNOW 3G 算法 FPGA 设计,然而前者资源开销仅为后者的三分之一,并且在功耗开销方面,对于相同的数据吞吐率,前者比后者小 63%。

参 考 文 献

- 1 ETSI/SAGE TS 35. 221—2011 ,Specification of the 3GPP Confidentiality and Integrity Algorithms 128—EEA3 & 128—EIA3; Document 1;128—EEA3 and 128—EIA3 Specification
- 2 ETSI/SAGE TS 35. 222—2011 ,Specification of the 3GPP Confidentiality and Integrity Algorithms 128—EEA3 & 128—EIA3; Document 2;ZUC Specification
- 3 ETSI/SAGE TS 35. 223—2011 ,Specification of the 3GPP Confidentiality and Integrity Algorithms 128—EEA3 & 128—EIA3; Document 3;Implementors test data
- 4 张洪铭,何登平. 基于 LTE 系统的 SNOW 3G 密算法研究. 宽带网络,2012;12(34):91—93
- 5 冯秀涛. 3GPP LTE 国际加密标准 ZUC 算法. 信息安全与通信保密,2011;12;45—46
- 6 Zhang Lingchen, Xia Luning, Liu Zongbin. Evaluating the optimized implementations of SNOW 3G and ZUC on FPGA. IEEE TrustCom, 2012;436—442

The Performance Comparison between SNOW 3G and ZUC Algorithms on FPGA

HAN Yue, HEI Yong, QIAO Shu-shan

(Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, P. R. China)

[Abstract] SNOW3G and ZUC algorithm are both the heart of secure algorithm sets in 3GPP LTE-Advanced, and have become the 4G mobile broadband communication standard. SNOW 3G algorithm is widely applied because it arose much earlier than ZUC algorithm which is designed by China and adopted as international standard in 2011. In order to compare the implementation performance of two algorithms, the structures of both algorithms are firstly analysed, and then proposed the FPGA schemes and implement them. Modelsim is used to simulate the designs, and Quartus II is used to synthesize the designs. The simulation and synthesis results show that ZUC algorithm is better in area overhead and power consumption, but worse in data throughput rate.

[Key words] SNOW 3G algorithm ZUC algorithm Field Programmable Gate Arrays
performance comparison

(上接第 1646 页)

- 3 Jainak, Robert Pwduin, MAO J. statistical pattern recognition; a review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000; 22 (1) : 4—37
- 4 Duan Wei, Lu Xiang-yang. Weighted Naive Bayes classifier model based on information gain. Changsha, China: Intelligent System Design and Engineering Application, 2010; (2) : 819—822
- 5 任 靖, 李春平. 最小距离分类器的改进算法——加权最小距离

- 分类器. 计算机应用, 2007; 25 (5) : 992—994
- 6 邱潇钰, 张化祥. 基于核的最小距离分类法的参数选择方法. 计算机工程, 2008; 34 (5) : 188—190
- 7 陈慧萍, 林莉莉, 王建东, 等. WEKA 数据挖掘平台及其二次开发. 计算机工程与应用, 2008; 44 (19) : 76—79
- 8 赵 蕊. 基于 WEKA 平台的决策树算法设计与实现. 武汉: 中南大学, 2007

An Improved Algorithm of Decision Tree Based on Information Gain and Minimum Distance Classification

DUAN Wei¹, MA Li¹, LU Xiang-yang²

(School of Math and Computer Science, Jiangxi Science & Technology Normal University¹, Nanchang 330013, P. R. China;

School of Electric and Information Engineering², Zhongyuan University of Technology, Zhengzhou 450007, P. R. China)

[Abstract] The decision tree is the commonly used method of classification. Based on the algorithm of ID3 decision tree, an improved decision tree algorithm is put forward. Aiming at the training sets which have some instances that have the same attributes but belong to different classes, the improved ID3 algorithm using a new method that combines attribute reduction and minimum distance classification which are based on information gain rather than majority vote method to judge classification for leaf node. The experimental results show the improved algorithm has a good effect on optimizing the structure of decision tree and improving accuracy rate of classification.

[Key words] information gain minimum distance decision tree classification