IEEE802.1AE 中 GCM 的高速硬件实现

赵晶晶[①] 李 丽[①] 潘红兵[①] 许 俊[②] 吴志刚[①] 林 军[①]
[①](南京大学江苏省光电信息功能材料重点实验室 南京 210093)
[②](盛科网络(苏州)有限公司 苏州 215021)

摘 要：该文设计了一种适用于 IEEE802.1AE 协议的 GCM 高速硬件结构。GCM 的核心模块包括 AES 和 Ghash 两部分。该文中 Ghash 模块采用了一种新型的并行乘加器，可以同时处理多组数据，而不需要预先确定等待处理的分组数据总数；为了支持密钥每个时钟周期不断变化，AES 中密钥扩展模块采用了循环展开结构。该文采用二度并行的 Ghash 模块实现了 GCM 高速加密电路，使用 Fujitsu 0.13 μm 1.2 V 1P8M CMOS 工艺进行逻辑综合，得到吞吐率为 97.9 Gbps，面积为 547 k 门，时钟频率达到 764.5 MHz。

关键词：IEEE802.1AE 协议；GCM 算法；AES 算法；密钥扩展；Ghash 函数；硬件实现
中图分类号：TP309.7; TN431.2 **文献标识码**： A **文章编号**：1009-5896(2010)06-1515-05
DOI: 10.3724/SP.J.1146.2009.00651

High-Speed Hardware Implementation for GCM in IEEE802.1AE

Zhao Jing-jing[①] Li Li[①] Pan Hong-bing[①] Xu Jun[②] Wu Zhi-gang[①] Lin Jun[①]
[①](*Key Laboratory of Advanced Photonics and Electronics Materials*, *Nanjing University*, *Nanjing* 210093, *China*)
[②](*Centec Networks*(*Su Zhou*) *Co.*, *Ltd.*, *Suzhou* 215021, *China*)

Abstract: This paper presents a high-speed GCM architecture, which is suitable for IEEE 802.1AE protocol. The core modules of GCM include AES and Ghash. In Ghash module, a new parallel multiply-adder is proposed, which can handle several sets of data at the same time without knowing the total number of data blocks in advance. To support constant key changes in each clock cycle, loop-unrolling structure is used in KeyExpansion module of AES. A GCM encryptor design example with 2-parallel Ghash is implemented and the performance is evaluated by utilizing Fujitsu 0.13 μm 1.2 V 1P8M CMOS technology and a very high throughput of 97.9 Gbps is obtained with 547 Kgates, operating at 764.5 MHz.

Key words: IEEE802.1AE protocol; Galois/Counter Mode(GCM) algorithm; AES algorithm; Key Expansion; Ghash function; Hardware implementation

1 引言

GCM(Galois/Counter Mode)[1]是一个高速的认证加密模式，分别用 AES(Advanced Encryption Sandard)的 CTR(Counter)模式[2]和定义在 GF(2^{128})域上的 Ghash 函数，同时产生密文和认证标签。IEEE802.1AE 协议[3]采用 GCM 算法对帧进行加/解密处理以及完整性校验处理，更好地保证通信安全。

GCM 适合并行运算，但目前尚缺少适用于 IEEE802.1AE 协议的高速 GCM 硬件实现。文献[4]中的设计通过平衡 AES 模块和乘法器单元的关键路径延时，实现了 42.67 Gbps 的吞吐率，面积为 298 k 门。文献[5,6]实现了 34.69 Gbps 和 40 Gbps 的吞吐率。但是文献[4–6]中的设计不符合 IEEE802.1AE 协议。文献[7]提出了一种并行乘加器，使 GCM 的吞吐率突破了 100 Gbps，达到 162.56 Gbps。然而进一步研究可以发现该文中的 GCM 硬件结构也不能适用于 IEEE802.1AE 协议，因为文献[7]提出的 Ghash 模块要在数据传输开始时确定等待处理的分组数据总数以控制并行运算过程，而在实际数据传输中这是不可行的。

本文针对 IEEE802.1AE 协议，设计了一种 GCM 高速硬件结构。其中 Ghash 模块采用一种新型的并行乘加器，可以同时处理多组数据，而不需要预先确定等待处理的分组数据总数。为了支持密钥每个时钟周期不断变化，AES 的密钥扩展模块采用了循环展开结构。本文采用二度并行的 Ghash 模块实现了 GCM 高速加密电路(GCM 的解密部分可复用加密电路)，采用 Fujitsu 0.13 μm 1.2 V 1P8M CMOS 工艺进行逻辑综合，得到吞吐率为 97.9

2009-04-30 收到，2009-10-08 改回
国家 863 计划项目(2008AA01Z135)和国家自然科学基金(60876017)资助课题
通信作者：李丽 lili@nju.edu.cn

Gbps，面积为 547 k 门，时钟频率达到 764.5 MHz。本设计硬件效率略高于文献[7]的设计，且可更好地适用于 IEEE802.1AE 协议。

2 GCM 算法

GCM 算法对数据进行加密时，有 4 个输入信号：加密密钥 K，初始化向量 **IV**，明文 P，以及附加鉴别信息 A；有两个输出信号：密文 C 和鉴别标识 T。将 P 和 A 按 128 位分组，分别记为：$P_1, P_2, P_3, \cdots, P_{(n-1)}, P_n^*$ 和 $A_1, A_2, A_3, \cdots, A_{(m-1)}, A_m^*$。其中 P_n^* 和 A_m^* 的长度分别为 u 和 $v(1\leqslant u,v\leqslant 128)$；其它分组长度皆为 128 位。

GCM 加密算法定义如式(1)所示。

$$\left.\begin{aligned} &H = E(K, 0^{128}) \\ &Y_0 = \begin{cases} \mathbf{IV} \,||\, 0^{31} \,||\, 1, & \text{len}(\mathbf{IV})=96 \\ \text{GHASH}(H, \{\,\}, \mathbf{IV}), & \text{其他} \end{cases} \\ &Y_i = \text{incr}(Y_{i-1}),\ i=1,\cdots,n \\ &C_i = P_i \oplus E(K, Y_i),\ i=1,\cdots,n-1 \\ &C_n^* = P_n^* \oplus \text{MSB}_u(E(K, Y_n)) \\ &T = \text{MSB}_t(\text{GHASH}(H, A, C) \oplus E(K, Y_0)) \end{aligned}\right\} \tag{1}$$

其中||表示数据串连接；len( )返回 64-bit 的数据串长度；$E(K,Y)$表示用密钥 K 对 Y 进行 AES 加密；函数 incr( )是将参数的最低 32 位看成一无符号数，将其加 1 后并取模 2^{32}。

GHASH 函数定义为：$\text{GHASH}(H,A,C) = X_{m+n+1}$，而 X_i，$i=0,\cdots,m+n+1$ 如式(2)所示。

$$X_i = \begin{cases} 0, & i = 0 \\ (X_{i-1} \oplus A_i) \cdot H, & i = 1,\cdots,m-1 \\ (X_{m-1} \oplus (A_m^* \,||\, 0^{128-v})) \cdot H, & i = m \\ (X_{i-1} \oplus C_{i-m}) \cdot H, & i = m+1,\cdots, m+n-1 \\ (X_{m+n-1} \oplus (C_n^* \,||\, 0^{128-u})) \cdot H, & i = m+n \\ (X_{m+n} \oplus (\text{len}(A) \,||\, \text{len}(C))) \cdot H, & i = m+n+1 \end{cases} \tag{2}$$

乘法是定义在 GF(2^{128})上的运算，约化多项式为式(3)。

$$g(x) = x^8 + x^4 + x^3 + x + 1 \tag{3}$$

3 GCM 高速硬件实现

GCM 加密模块(GCM Encryptor)可以分为 3 个部分：信息提取模块(InfoExtn)，AES 模块和 Ghash 模块，其中 AES 模块又包含密钥扩展(KeyExpansion)和 AES 加密模块(Enc)，如图 1 所示。

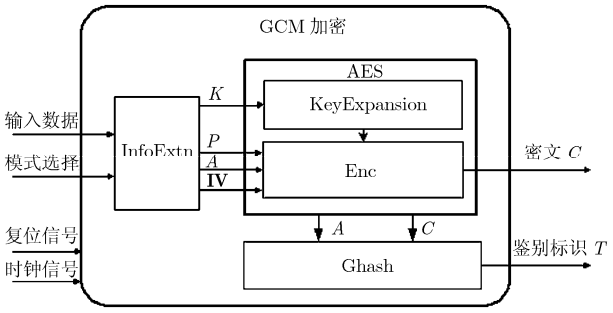

图 1 GCM 的结构框图

InfoExtn 模块用于从输入数据中提取安全协议密钥，安全通道标识，数据包号，目的地址和源地址，并根据不同的工作模式转换成对应的 K，**IV**，A 和 P。AES 模块完成数据加密，输出密文。Ghash 模块通过 Ghash 函数产生鉴别标识 T。

下面将分别介绍 AES 和 Ghash 模块的硬件设计和 GCM 的总体硬件实现。

3.1 AES 的硬件设计

3.1.1 Enc 模块 AES 的加密模块用来实现 GCM-AES-128 加密功能，输入 128 位明文，经过加密输出 128 位密文。AES-128 加密算法可参考文献[2]，在此不再赘述。GCM-AES-128 加密算法如式(1)所示。

为进一步提高加密速度，本设计的 AES 加密模块运用复合域算法[8]，采用了全流水线结构[9−12]，如图 2 所示。将加密过程的内部循环全部展开，在每轮循环间插入一级流水线，共有 10 级轮间流水线，同时每级流水线内部又采用 7 级子流水线结构，总计包含 70 级流水，最多可以同时处理 70 组数据，经过初始 70 个时钟周期后，每个时钟都能输出 1 组 128 位的密文。

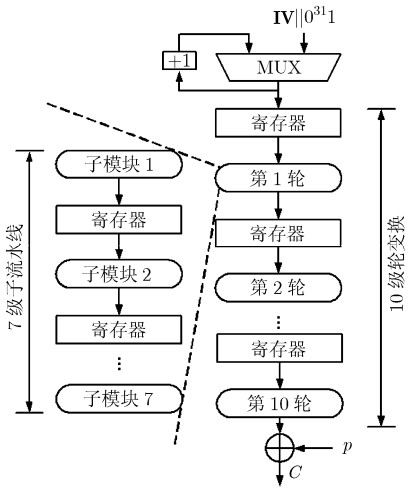

图 2 Enc 模块的电路结构图

3.1.2 KeyExpansion 模块　AES 的密钥扩展模块(KeyExpansion)输入初始密钥 K，经过密钥扩展输出 11 组 128 位的轮密钥(包括 K)。用伪代码写成的密钥扩展程序可见参考文献[2]。

在网络传输过程中，各个用户数据可能使用不同的密钥 K。文献[9]中采用迭代的方法进行密钥展开，密钥扩展模块被锁定 30(或 70)个时钟周期依次产生 11 个轮密钥(包括 K)，然后才能接收下一个密钥 K，即 AES 模块要等待 30(或 70)个时钟周期才能加密下一组数据。本设计中 AES 加密模块采用全流水线结构，最快要求同时处理 70 组数据，显然文献[9]中的结构不能满足本设计的要求。

本文的密钥扩展模块设计了循环展开结构，如图 3 所示。该结构包含 11 个密钥扩展单元，Enc 模块也包含了相同的流水设计，保证了两个模块能够高速协调运算。KeyExpasion 的第零级到第 9 级密钥扩展单元有相同的结构，而第 10 级密钥扩展单元的结构与前 10 级不同，如图 3 所示。其中，SubBytes 表示将 4 个字节数据进行 S 盒运算；ShiftLeft 将每个字节数据循环左移一位。

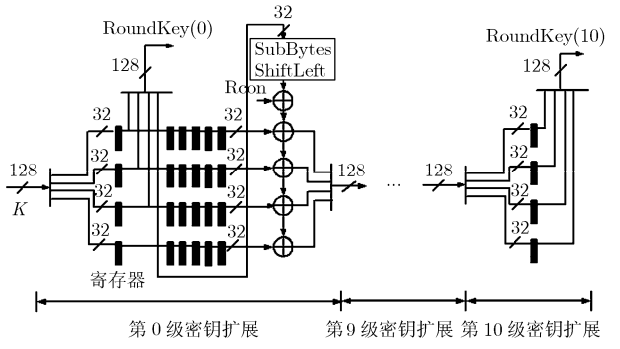

图 3 KeyExpansion 模块的电路结构图

KeyExpansion 模块每个时钟周期输入一个初始密钥 K，经过初始 55 个时钟周期后，每个时钟都能输出 11 组 128 位的密钥，保证了 AES 模块在密钥 K 不断变化时，仍能高速工作。

3.2 Ghash 的硬件实现

Ghash 函数的运算是一个迭代过程，顺序执行乘加操作，如式(2)所示。而乘法器单元具有较大的延时，是提高吞吐率的瓶颈。提高 Ghash 函数运算速度的关键在于采用并行的乘法运算替代顺序运算。

在文献[7]中提出了一种并行乘加器，但是文献[7]中的设计不能运用于 IEEE802.1AE 协议，因为文献[7]提出的 Ghash 模块要在数据传输开始时确定等待处理的分组数据总数以控制并行运算过程，而在实际数据传输中这是不现实的。针对这一点，本文设计了一种新型的并行乘加器，可完全适用于 IEEE802.1AE 协议。

先将 Ghash 函数的式(2)进行展开，如表达式(4)所示，X_i 不再是变量表达式而是常量表达式，为并行运算提供了可能。

$$\left.\begin{aligned} X_0 &= 0 \\ X_1 &= (X_0 \oplus A_1)\cdot H = A_1 \cdot H \\ X_2 &= (X_1 \oplus A_2)\cdot H = A_1 \cdot H^2 \oplus A_2 \cdot H \\ &\vdots \\ X_i &= (X_{i-1} \oplus A_i)\cdot H = A_1 \cdot H^i \oplus A_2 \\ &\quad \cdot H^{i-1} \oplus \cdots \oplus A_i \cdot H,\ i < m \\ &\vdots \end{aligned}\right\} \tag{4}$$

由式(4)可以进一步总结，Ghash 函数输入(*pq*+*n*)组数据时，输出 X_{pq+n} 的表达式为式(5)，其中 p, q, n 为正整数，$1 \leqslant n \leqslant q$。

$$\begin{aligned} X_{pq+n} &= (X_{pq+n-1} \oplus A_{pq+n})\cdot H \\ &= (A_1 \cdot H^{pq} \oplus A_2 \cdot H^{pq-1} \oplus \cdots \oplus A_{pq+1})\cdot H^n \\ &\quad \oplus A_{pq+2} \cdot H^{n-1} \\ &\quad \vdots \\ &\quad \oplus A_{pq+n} \cdot H \end{aligned} \tag{5}$$

由式(5)可以设计出一种适用于 IEEE802.1AE 协议的新型 q 度并行乘加器。以 q=2 为例，Ghash 模块的电路结构如图 4 所示，该结构针对实际情况设计，不需要预先确定等待处理的分组数据总数，只要对最后输入的一组数据进行判断，控制并行运算过程的控制逻辑简单：如果最后输入两个数据，则 MUX1 输出 H^2，MUX2 输出 H；如果只有一个数据，则 MUX1 输出 H，MUX2 输出 0。

图 5 是输入 5 个数据 A_1–A_5 时，Ghash 模块 5 输入实例图。图 5(a)中，输入两个数据 A_1，A_2，MUX1 取 H^2，MUX2 取 H，经过两个周期输出 X_2

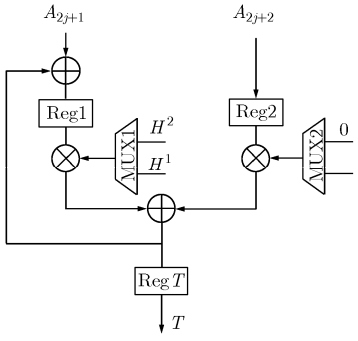

图 4 Ghash 模块的电路结构图(q=2)

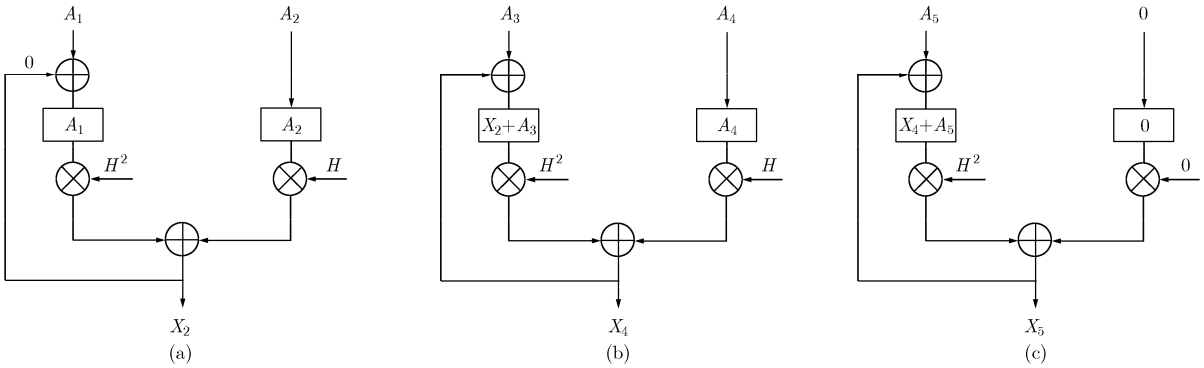

图 5 Ghash 模块 5 输入实例图

如式(6a)；X_2 与输入数据 A_3 作异或运算，结果存入寄存器 Reg1，A_4 存入寄存器 Reg2，MUX1 取 H^2，MUX2 取 H，经过两个周期输出 X_4 如式(6b)，如图 5(b)所示；最后输入一个 128 位数据 A_5，MUX1 取 H，MUX2 取 0，经过两个周期输出 X_5 如式(6c)，如图 5(c)所示。

$$X_2 = A_1 \cdot H^2 \oplus A_2 \cdot H \tag{6a}$$

$$X_4 = (X_2 \oplus A_3) \cdot H^2 \oplus A_4 \cdot H = A_1 \cdot H^4 \oplus A_2 \cdot H^3 \oplus A_3 \cdot H^2 \oplus A_4 \cdot H \tag{6b}$$

$$X_5 = (X_4 \oplus A_5) \cdot H = A_1 \cdot H^5 \oplus A_2 \cdot H^4 \oplus A_3 \cdot H^3 \oplus A_4 \cdot H^2 \oplus A_5 \cdot H \tag{6c}$$

采用 Synopsys DC 工具，针对 Fujitsu 0.13 μm 1.2 V 1P8M CMOS 工艺库进行综合，得出本设计中乘法器的关键路径延时约是 AES 模块的两倍。因此本设计的 Ghash 模块采用图 4 中二度并行的结构，两个乘法器单元并行工作，使 Ghash 模块的关键路径延时减半，平衡了 Ghash 模块和 AES 模块的关键路径延时，提高了电路速度。GCM 每个时钟输入一组数据，先行缓存，经过两个时钟周期输入 Ghash 模块，Ghash 中的乘法器单元两个时钟周期完成一次乘法运算，从而突破了乘法器单元的较大延时对速度的限制。按照上述分析，用 DC 进行逻辑综合时，注意 Ghash 模块中的乘法路径采用多周期路径的方法，即两个时钟周期完成一次乘法运算。

3.3 GCM 的总体硬件实现

综上所述，GCM 的整体电路结构如图 6 所示，包含密钥扩展模块(KEYEXP)，AES 加密模块(Enc)和二度并行的 Ghash 模块。128 位的寄存器 Y 作为计数器，每个时钟周期加 1。当最后一组数据的长度小于 128 时，128 位的寄存器 MASK 用于数据掩码。Reg1，Reg2 是深度为 128 的寄存器，每个时钟周期输入一个 128 位的数据，Reg1，Reg2 等两个时钟周期依次存满 128 位数据；若先输入一个 128 位的数据存入 Reg1 后，等一个时钟周期没有数据输入，则 Reg2 置零；若输入数据不满 128 位，则添 0 补齐。

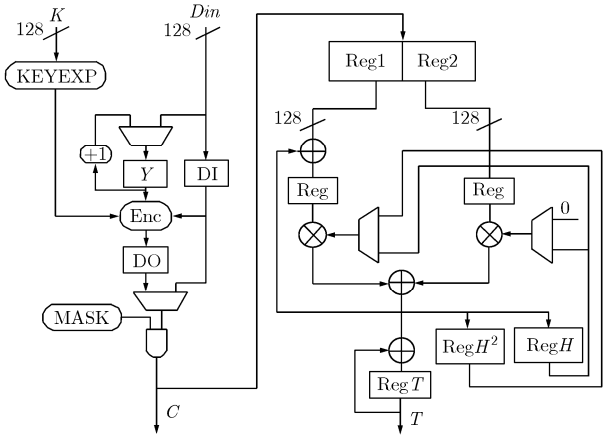

图 6 GCM 的电路结构图

文献[7]中的设计要求同时输入 4 组用户数据，进行 4 度并行运算，而实际情况是用户数据是一组一组输入的。本文的 GCM 加密模块针对数据实际输入情况进行设计，每个时钟周期只需输入一组用户数据，本设计与文献[7]的实验结果比较见文章后续部分。

4 实现结果

本文设计了一种适用于 IEEE802.1AE 协议的新型并行乘加器和循环展开结构的密钥扩展模块，同时采用了全流水线结构的 AES 加密模块，构成了 GCM 加密模块的高速硬件结构。使用 Fujitsu 0.13 μm 1.2 V 1P8M CMOS 工艺进行逻辑综合，综合结果以及和最近发表文章的比较见表 1。为了便于比较，这里引入了参数——硬件效率。

$$\text{硬件效率} = \frac{\text{吞吐率}}{\text{面积}} \tag{7}$$

由表 1 可知，本设计的时钟频率为 764.5 MHz，得到最大吞吐率为 97.9 Gbps，面积为 547 k 门。本

表 1 GCM 的 ASIC 实现性能比较

各种设计	GCM 结构	AES 结构	最大频率 (MHz)	吞吐率 (Gbps)	面积 (gates)	硬件效率 (kbps/gate)	单元库	支持 IEEE802.1ae
本文设计	2- Parallel	Pipelined	764.5	97.9	547,233	178.9	0.13 μm	Yes
文献[4]	Sequential	Pipelined	333.3	42.67	297,542	143.40	0.13 μm	No
		4-stage Pipelined loop		10.67	118,645	89.90		
	4-Parallel	Loop		10.67	162,373	65.69		
文献[5]	Sequential	Pipelined	271.0	34.69	498.658	69.57	0.18 μm	
文献[6]				20.00	250,000	80.00		
				40.00	400,000	100.00		
文献[7]	4-Parallel	Pipelined	317.5	162.56	979,348	165.99	0.13 μm	

设计的硬件效率比已经发表的性能最好的设计[7]的硬件效率略高,而且本设计适用于 IEEE802.1AE 协议。在符合 IEEE802.1AE 协议的前提下,本设计的吞吐率最大,硬件效率最高。因此,本设计可满足 IEEE802.1AE 协议高速应用方面的要求。

5 结束语

本文设计了一种适用于 IEEE802.1AE 协议的 GCM 高速硬件结构;提出了一种新型的并行 Ghash 的硬件结构;设计了循环展开结构的密钥扩展模块,以支持密钥的连续变化。此外,本文的设计方法可为同类高速电路设计提供指导。若采用多通道技术,本设计的吞吐率有望得到进一步提高。

参 考 文 献

[1] McGrew D A and Viega J. The Galois/Counter Mode of Operation (GCM). May 2005. http://www.csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-revised-spec.pdf.

[2] Daemen J, Rijmen V 著. 谷大武, 徐胜波译. 高级加密(AES)算法. 北京: 清华大学出版社, 2003: 31–64.

[3] IEEE Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Security, IEEE Standard 802.1ae, 2006.

[4] Satoh A. High-speed hardware architectures for authenticated encryption mode GCM. Proc. IEEE ISCAS, Island of Kos, May, 2006, 4.

[5] Yang B, *et al*.. High speed architecture for Galois/Counter Mode of operation (GCM). Cryptology ePrint Archive: Report 2005/146. Jun., 2005. http://eprint.iacr.org/2005/146.pdf.

[6] Elliptic Semiconductor Inc. CLP-15 AES-GCM Core Product Brief. 2004. http://www.ellipticsemi.com/CLP-15_5027. pdf

[7] Satoh A. High-speed parallel hardware architecture for Galois Counter Mode. Proc. IEEE ISCAS, New Orleans, LA, 2007: 1863–1866.

[8] Nalini C Dr, *et al*.. Compact Designs of SubBytes and MixColumn for AES. IEEE IACC, Seattle, WA, March 6–7, 2009: 1584–1587.

[9] Zhang Xin-miao and Parhi K K. High-speed VLSI architectures for the AES algorithm. . *IEEE Transactions on Very Large Scale Integration* (*VLSI*) *Systems*, 2004, 12(9): 957–967.

[10] Saleh Abdel-hafeez, *et al*.. High performance AES design using pipeling structure over GF((24)2). IEEE ICSPC, Dubai, November 24–27, 2007: 716–719.

[11] Fan Chih-peng and Hwang Jun-kui. Implementations of high throughput sequential and fully pipelined AES processors on FPGA. proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems, Xiamen, China, Nov.28–Dec.1, 2007: 353–356.

[12] Rizk M R M and Morsy M. Optimized area and optimized speed hardware implementations of AES on FPGA. International Design and Test Workshop, 2007 2nd, Cairo, Dec. 16–18, 2007: 207–217.

赵晶晶: 女,1984 年生,硕士生,研究方向为数字集成电路设计.

李 丽: 女,1975 年生,副教授,研究方向为 VLSI 设计技术和设计方法学.

潘红兵: 男,1971 年生,副教授,研究方向为多核处理器、CMOS 传感器.

许 俊: 男,1972 年生,博士后,研究方向为以太网交换路由芯片的研发.