# 部分并行的蒙哥马利模乘法器实现研究

刘 强[1,2], 佟 冬[1,2], 程 旭[1,2]
(1. 北京大学计算机科学技术系, 北京 100871; 2. 北京大学微处理器研究开发中心, 北京 100871)

摘 要: 通讯技术的高速发展需要更高性能的密码处理设备. 本文系统的研究了利用部分并行的脉动阵列结构实现模乘法器的问题, 在分析了各类结构的性能结果的基础上, 找到性能最高的和性能-代价比最高的结构配置. 结果表明, 如果以关键路径延迟×单元面积作为时间-面积代价指标, 那么每个运算单元处理一位数据的设计, 在 0.18μm 工艺和 0.25μm 工艺条件下, 具有最高的运算性能, 同时也是性价比最高的设计之一.

关键词: 蒙哥马利算法; 模乘法; 模乘幂; RSA; 公开密钥加密; 脉动阵列
中图分类号: TP309.7; TN47 文献标识码: A 文章编号: 0372-2112 (2006) 08-1537-05

# Implementation Study of Partly Parallel Montgomery Modular Multiplier

LIU Qiang[1,2], TONG Dong[1,2], CHENG Xu[1,2]
(1. Department of Computer Science and Technology, Peking University, Beijing 100871, China;
2. Microprocessor Research and Development Center, Peking University, Beijing 100871, China)

**Abstract** The rapid advance in communication technology brings a request for cryptography systems of higher performance. We systematically implement and compare several variants of partly parallel systolic architecture for Montgomery multiplier with different bit length as well as with different micro architectural approaches. The optimal options are chosen to take advantage of the underlying technology. The result analysis shows that the fully serial systolic architecture, in which one cell processes one bit, achieves the best performance. When the resource overhead is represented as area-time product, it is one of the most cost-efficient designs as well.

**Key words** Montgomery algorithm; modular multiplication; modular exponentiation; RSA; public-key cryptography; systolic array

## 1 引言

随着社会信息化的发展, 越来越多的敏感数据通过网络传送. 这使得对可靠的高性能安全产品的需求越来越大. 信息安全已成为世人关注的重大问题, 而密码技术的飞速发展为信息安全提供了强有力的保护手段. 公开密钥密码体制, 比如 RSA 算法[1], ECC 算法[2], DSA 算法以及 Diffie-Hellman 密钥交换算法[3], 在现代安全系统中占据了重要的位置. 随着密码系统的应用越来越广泛, 人们对加密速度和加密强度的要求也越来越高. 密码算法的硬件实现, 其运算速度远高于软件实现, 但是由于加密算法的计算比较复杂, 硬件开销相当大. 因此如何提高密码处理器的性能-代价比, 在面积、速度和强度之间选取适当的折衷方案, 已成为一个亟待解决的问题.

大多数公开密钥密码算法依赖于长整数的有限域乘法或者模乘法. 1985 年提出的蒙哥马利算法[4], 是应用最为广泛的模乘法算法之一, 该算法不需要通常算法所用到的除法操作, 而是采用硬件上易于实现的加法和移位操作, 因而非常适合于 VLSI 实现.

我们详细探讨了采用部分并行的脉动阵列结构实现蒙哥马利算法的问题, 实现了一系列不同长度和不同内部结构的运算单元, 从中选择具有最高性能的和最高性价比的结构. 分析结果表明, 如果以关键路径延迟×单元面积作为时间-面积代价指标[7,10], 那么全部串行的脉动阵列结构, 即每个运算单元处理一位数据的设计, 在 0.18μm 工艺和 0.25μm 工艺条件下, 具有最高的运算性能, 同时也是性价比最高的设计之一.

本文下面的部分组织如下: 第二节介绍部分并行的脉动阵列结构, 第三节对各类设计的实验结果进行详细分析, 并从中选择出性能最高的和性价比最高的设计, 第四节展示性能结果, 并与现有设计进行性能比较, 第五节是总结.

收稿日期: 2004-09-17; 修回日期: 2006-05-15

## 2 结构设计

### 2.1 用于 RSA 模乘幂运算的脉动阵列乘法器

在深亚微米工艺条件下,集成电路的性能瓶颈已经从逻辑单元转向了单元连线. RSA 算法中的加密解密操作,都是大整数的算术运算,在硬件实现上需要长的(千位以上的)运算结构. 文献中原有的设计都需要做全局数据广播,不仅扇出大、延时长,而且需要的连线资源很大. 脉动阵列结构虽然可以将运算数据的传输限制在局部,但是仍然需要在控制模块、输入输出端口以及长的运算结构之间进行数据传输.

在文[8,9]中,我们提出采用一种模块化的结构进行信号的全局广播(图 1),长的运算结构被划分为一串运算模块,运算模块内部可以在单周期内完成一定的运算,运算模块之间通过流水化的全局总线进行数据传输. 同时,运算模块中的控制信号通过移位寄存器传送. 这样,除了时钟信号之外,其他的信号都被限定一个模块内部或者相邻两个模块之间,从而解决全局数据广播引起的问题.

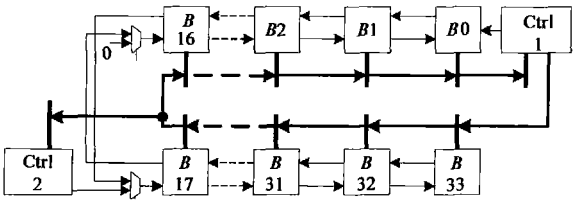


图 1 脉动阵列模乘法器的整体结构

### 2.2 脉动阵列运算单元的结构设计

在图 1中,每个运算模块(BQ, B1, …, B33)用于处理32位的数据,其中包括 32/u 个运算单元(u = 1, 2, 4, …, 32). 图 2(a)显示的是 u 位运算单元的结构(Case-1),其中包括脉动阵列乘法器中的寄存器和控制信号. 寄存器 SP 和 SR 用于存放模乘法的中间运算结果, P和 R 用于存放模乘幂的中间运算结果. 控制信号包括两位的进位信号 ci, 商 qi和乘数 ai, S的重置使能 rst, P和 R 写使能 wp和 wr. 图中的运算单元,每隔一个周期就会有一次空隙,利用率仅为 50%. 为了充分利用硬件资源,两个模乘法可以并行计算. 通常所用到的平方乘法算法,即从右到左的二进制算法[11],可以应用到脉动阵列中来,使得其运算单元在偶数周期计算平方,在奇数周期计算乘法. 这样的策略有一些存储和转换的开销. 控制信号 sel用于选择不同周期的输入数值以及目的寄存器,完成交叉执行策略.

在图 2(a)的结构中, P和 R 寄存器的输入源是 u 位加法器的计算结果,这增加了相关信号,特别是 sel, qi等关键信号的负载,从而增加了延迟. 图 2(b)中的结构(Case-2)将 P和 R 寄存器的输入源改为 SP和 SR寄存器,从而减少了相关信号的负载.

采用交叉执行策略的原因,是为了降低面积,但是由于转换逻辑的原因,增加了计算时间. 在不采用交叉执行策略的位片结构中(图 3(a), Case-3),两个模乘法分别由一个 u 位的加法器进行计算,从而减少电路中转换逻辑的时间开销. 图中两个模乘法分别采用一个 u 位的加法器,它们共用 qi信号,但是 ai, ci都是独立的(ap, ar, cp, cr). ws 是 SP和 SR寄存器的写控制信号.

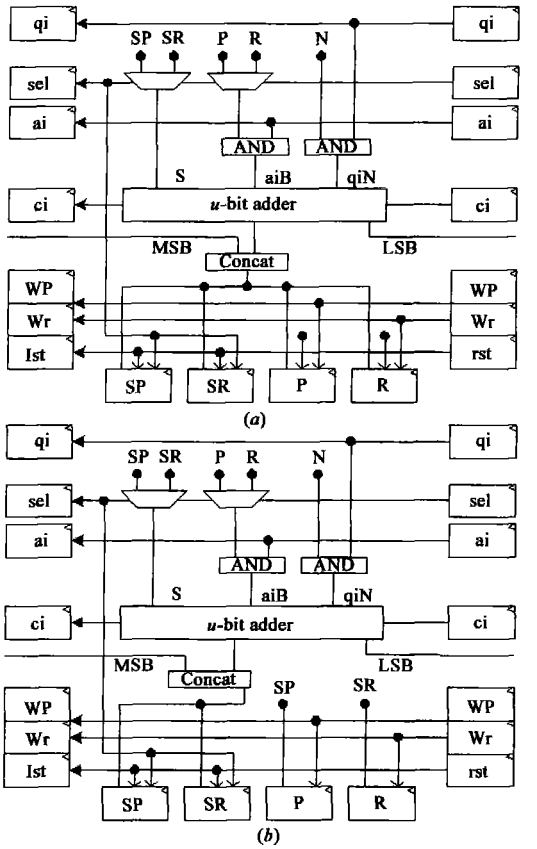


图 2 交叉执行运算单元(Case-1)及其改进结构(Case-2)

对应于 Case-2的结构,图 3(b)中的结构(Case-4)将 P和 R 寄存器的输入源改为 SP和 SR寄存器,从而减少了相关信号的负载.

表 1 时钟周期 = 1.5ns时的单元面积(0.18μm 工艺)

| Block | 1×32 | 2×16 | 4×8 | 8×4 | 16×2 | 32×1 |
|---|---|---|---|---|---|---|
| Case-1 | 50112 | | | | | |
| Case-2 | 47471 | | | | | |
| Case-3 | 63803 | 53744 | 54592 | | | |
| Case-4 | 63803 | 52377 | 55421 | | | |

## 3 性能分析和评测

### 3.1 具有最高运算频率的结构

在后期的实现中我们发现,乘法器控制模块的关键路径延迟在 1.5ns(0.18μm 工艺)和 2.0ns(0.25μm 工艺)左右,所以构成乘法器数据通路的模块要选择延迟在 1.5ns(0.18μm 工艺)和 2.0ns(0.25μm 工艺)以上的结构. 为了

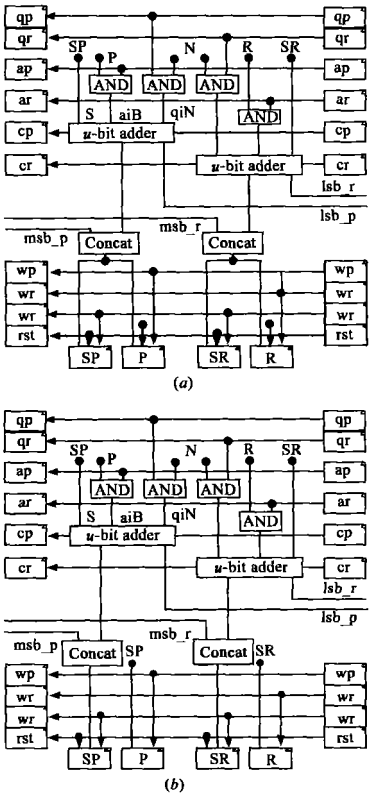


图 3　不采用交叉执行策略的运算单元（Case-3）及其改进结构（Case-4）

在性能(主频)最高的设计中找到面积最小的结构, 我们以控制模块的最小延迟为时序约束条件, 分别对满足条件的配置进行综合, 得到各模块的单元面积($\mu m^2$)见表 1和 2. 表项 M xN 表示运算单元的宽度是 M, 共有 N 个运算单元构成一个 32位的运算模块.

表 2　时钟周期 = 2 0ns时的单元面积(0 25μm 工艺)

| Block | 1×32 | 2×16 | 4×8 | 8×4 | 16×2 | 32×1 |
|---|---|---|---|---|---|---|
| Case-1 | 85311 | | | | | |
| Case-2 | 87033 | | | | | |
| Case-3 | 113644 | 99740 | | | | |
| Case-4 | 108432 | 97009 | | | | |

表 3　两种乘法器结构的综合结果(单元面积 $\mu m^2$)

| | 0 18μm 工艺 (Delay= 1 5ns) | 0 25μm 工艺 (Delay= 2 0ns) |
|---|---|---|
| Case-1(1×32结构) | 2173090 | 3646051 |
| Case-2(1×32结构) | 2080802 | 3469403 |

在符合约束条件的配置中, Case-1的 1×32结构和 Case-2的 1×32结构单元面积是最小的, 作为乘法器的备选结构, 构成模乘法器, 并最后进行逻辑综合, 得到表 3的结果. 以 Case-2的 1×32结构构成的模乘法器, 与以 Case-1的 1×32结构构成的乘法器相比, 单元面积减少 4 25% (0 18μm 工艺)和 4 84% (0 25μm 工艺), 由 Case-2的 1×32结构构成的模乘法器将会有最高的运算速度和相对小的单元面积.

表 4　时钟周期 = 1 7ns时的单元面积(0 18μm 工艺)

| Block | 1×32 | 2×16 | 4×8 | 8×4 | 16×2 | 32×1 |
|---|---|---|---|---|---|---|
| Case-1 | 47301 | 44869 | | | | |
| Case-2 | 46822 | 45721 | 42814 | | | |
| Case-3 | 63773 | 51725 | 48848 | 47933 | | |
| Case-4 | 63773 | 50072 | 46849 | 46789 | | |

表 5　时钟周期 = 1 9ns时的单元面积(0 18μm 工艺)

| Block | 1×32 | 2×16 | 4×8 | 8×4 | 16×2 | 32×1 |
|---|---|---|---|---|---|---|
| Case-1 | 46652 | 42687 | 39840 | 38212 | | |
| Case-2 | 46399 | 40701 | 38795 | 38499 | | |
| Case-3 | 63803 | 45069 | 41350 | 43023 | 45728 | |
| Case-4 | 63803 | 45415 | 40282 | 42621 | 45232 | |

表 6　时钟周期 = 2 1ns时的单元面积(0 18μm 工艺)

| Block | 1×32 | 2×16 | 4×8 | 8×4 | 16×2 | 32×1 |
|---|---|---|---|---|---|---|
| Case-1 | 46426 | 39564 | 36400 | 34501 | 37169 | 46426 |
| Case-2 | 46370 | 38825 | 35868 | 34920 | 36264 | 38715 |
| Case-3 | 63733 | 44999 | 39304 | 40838 | 43143 | 48296 |
| Case-4 | 63733 | 44999 | 39251 | 39680 | 42411 | 45920 |

表 7　时钟周期 = 2 3ns时的单元面积(0 25μm 工艺)

| Block | 1×32 | 2×16 | 4×8 | 8×4 | 16×2 | 32×1 |
|---|---|---|---|---|---|---|
| Case-1 | 82748 | 82166 | | | | |
| Case-2 | 80732 | 81141 | 77996 | | | |
| Case-3 | 104561 | 86935 | 82183 | 82949 | | |
| Case-4 | 103662 | 85363 | 84988 | 81832 | | |

表 8　时钟周期 = 2 5ns时的单元面积(0 25μm 工艺)

| Block | 1×32 | 2×16 | 4×8 | 8×4 | 16×2 | 32×1 |
|---|---|---|---|---|---|---|
| Case-1 | 77512 | 75335 | 70698 | | | |
| Case-2 | 80127 | 77402 | 67829 | 66216 | | |
| Case-3 | 108489 | 85881 | 76193 | 80939 | | |
| Case-4 | 102101 | 83312 | 75841 | 79292 | | |

表 9　时钟周期 = 2 8ns时的单元面积(0 25μm 工艺)

| Block | 1×32 | 2×16 | 4×8 | 8×4 | 16×2 | 32×1 |
|---|---|---|---|---|---|---|
| Case-1 | 75530 | 69143 | 62277 | 61228 | 65013 | 75732 |
| Case-2 | 75029 | 66965 | 60140 | 58930 | 66211 | 69373 |
| Case-3 | 98876 | 81480 | 74874 | 71205 | 77627 | 98449 |
| Case-4 | 98876 | 79833 | 75168 | 71856 | 76043 | 88139 |

### 3.2　具有最高性价比的结构

我们定义一种设计的时间-面积代价指标为: TA = 关键路径延迟 ×单元面积[7, 10]. 关键路径的延迟与主频成反比, 所以 TA 越小表明结构的性能-代价比(简称性价比)越好. 集成电路的面积和时序约束之间并没有线性反比关系, 所以找到性能-代价比最高的结构, 没有系统的方法. 数据通路部分的面积大致占到模乘法器全部面积的 75%, 我们首先对构成数据通路的模块在不同配置(u = 1, 2, 4, 8, 16, 32)、不同时序约束下进行综合, 从中找出可能满足条件的结构. 在 0.18μm 工艺条件下, 分别设置时钟周期为 1.5ns, 1.7ns, 1.9ns, 2.1ns; 在 0.25μm 工艺条件下, 分别设

置时钟周期为 2.0ns, 2.3ns, 2.5ns, 2.8ns, 并进行逻辑综合, 其中只有部分结构满足时序要求, 它们的单元面积($\mu m^2$)列在表 1、2、4~9.

从表 1、2、4~9得到, 在两种工艺条件下, Case-2各类结构的面积基本上都小于 Case-1同类结构的面积, 因此, 我们将对 Case-2的各类结构做进一步的分析, 从中找出具有最小的时间-面积代价指标的结构. 在 0.18μm 工艺条件下, 分别设置时钟周期为 1.5~2.4ns; 在 0.25μm 工艺条件下, 分别设置时钟周期为 2.0ns~3.3ns, 对由 Case-2各类结构构成的数据通路进行逻辑综合, 其中只有部分结构满足时序要求, 它们的时间-面积代价折线图见于图 4(a)、图 4(b).

在 0.18μm 工艺条件下, 时间-面积代价指标最小的是 Case-2的 1×32结构 (时钟周期 = 1.5ns, TA = 2422785). 在 0.25μm 工艺条件下, 时间-面积代价指标最小的是 Case-2的 8×4结构 (时钟周期 = 2.8ns, TA = 5630444.4). 在 0.25μm 工艺条件下, Case-2的 1×32结构时间-面积代价指标最小是在时钟周期 = 2.0ns时, TA = 5936958, 与 8×4结构的最小值相差 5.44%.

## 4 运算性能和比较

从上一节的分析结果可以知道, 以 Case-2的 1×32结构构成的模乘法器, 即每个运算单元处理一位数据的设计, 在 0.18μm 工艺条件下, 具有最高的运算速度和性价比, 在 0.25μm 工艺条件下, 具有最高的运算速度和比较好的性价比. 我们对由这种结构构成的模乘法器进行逻辑综合, 得到结果如表 10所示.

表 **10** 乘法器的综合结果

| | 结构 | 延迟(ns) | 单元面积($\mu m^2$) | TA |
|---|---|---|---|---|
| 0.18μm 工艺 | Case-2 (1×32结构) | 1.5 | 2080802 | 3121203 |
| 0.25μm 工艺 | Case-2 (1×32结构) | 2.0 | 3469403 | 6938806 |

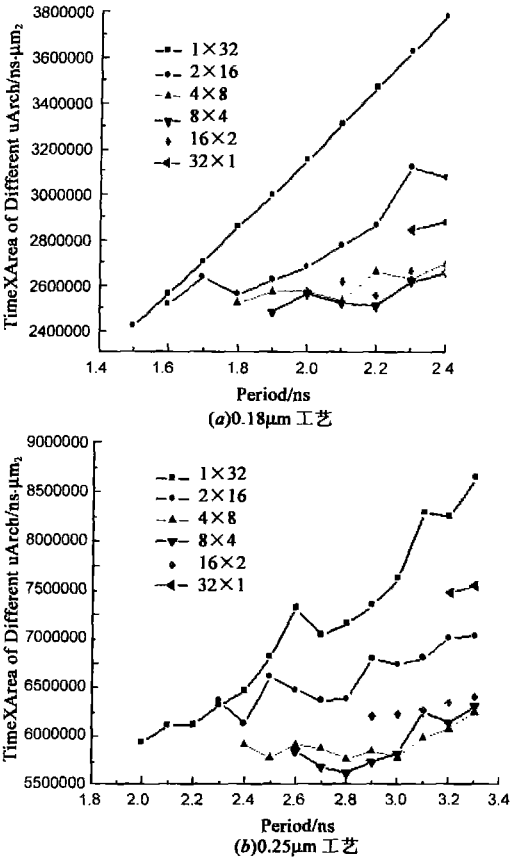


**图 4 数据通路在不同时钟约束下的时间-面积代价折线图**

与文献中已有的代表性设计在解密速度方面的性能比较, 列在表 11. 通过性能比较可以看出我们的设计在计算速度上的优势. 文献[7, 13, 14, 16]采用 FPGA 实现, 没有芯片规模的数据. "波特率"一栏没有数据的设计, 文献[5, 14, 16]在硬件上不直接支持中国剩余定理(CRT), 文献[13]是专门针对 CRT 的设计, 文献[15]没有非 CRT 的数据.

表 **11** 与现有设计的性能比较(解密速度)

| 设计 | 年份 | 工艺(um) | 芯片规模(Kilo Gates) | 主频(MHz) | 位长 | 时钟周期数(非 CRT) | 波特率(Kb/s) 非 CRT | 非 CRT |
|---|---|---|---|---|---|---|---|---|
| [7] | 1999 | FPGA | - | 52 | 1024 | 2(n+4)(n+2) | 25 | 101 |
| [6] | 2001 | 0.6 | 109 | 150 | 512 | 2(n+4)(n+2) | 141 | 578 |
| [5] | 2001 | 0.5 | 156 | 50 | 1024 | (n+36)(n+2) | 47 | - |
| [14] | 2002 | FPGA | - | 50 | 1024 | (n+3)(n+1) | 46 | - |
| [13] | 2003 | FPGA | - | - | 512 | $(n/2+3)^2$(CRT) | - | 91 |
| [15] | 2003 | FPGA | - | 100 | 1024 | (n/2+2)(n/2+3)(CRT) | - | 389 |
| | | 0.18 | 187 | 200 | 1024 | (n/2+2)(n/2+3)(CRT) | - | 778 |
| [8] | 2003 | 0.18 | 148 | 420 | 1024 | 2(n+2)(n+2) | 199 | 794 |
| [9] | 2003 | 0.18 | 137 | 450 | 1024 | 2(n+2)(n+2) | 214 | 851 |
| [12] | 2003 | 0.18 | 137 | 550 | 1024 | 2(n+2)(n+2) | 261 | 1,041 |
| [16] | 2004 | FPGA | - | 78 | 1024 | 2(n+4)(n+2) | 36 | - |
| Now | 2004 | 0.25 | 201 | 500 | 1024 | 2(n+2)(n+2) | 237 | 946 |
| | | 0.18 | 156 | 666 | 1024 | 2(n+2)(n+2) | 316 | 1,260 |

## 5 结论

本文系统的研究了利用部分并行的脉动阵列结构实现模乘法器的问题,针对深亚微米工艺实现的要求提出改进措施,利用模块化策略解决全局信号的广播问题,利用冗余策略解决长乘法器的动态分割问题.在系统的分析了各类结构的性能结果的基础上,找到了性能最高的和性能-代价比最高的结构配置.

参考文献:


[1] RIVEST R L, SHAMIR A, ADLEMAN L. A method for obtaining digital signatures and public-key cryptosystems [J]. Communications of the ACM, 1978, 21(2): 120-126.

[2] MILLER V. Use of elliptic curves in cryptography[A]. Advances in Cryptology-CRYPTO 85, Lecture Notes in Computer Science[C]. New York: Springer-Verlag, 1986. 417-426.

[3] DIFFIE W, Hellman M E. New directions in cryptography [J]. IEEE Transactions on Information Theory, 1976, 22(6): 644-654.

[4] Montgomery P L. Modular multiplication without trial division[J]. Mathematics of Computation, 1985, 44(170): 519-521.

[5] Kwon T W, You C S, Heo W S, Kang Y K, Choi J R. Two implementation methods of a 1024-bit RSA cryptoprocessor based on modified Montgomery algorithm[A]. Proceedings of the 2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001)[C]. New York: IEEE Press, 2001. 4: 650-653.

[6] Wu C H, Hong J H, Wu C W. RSA cryptosystem design based on the chinese remainder theorem[A]. Proceedings of the conference on Asia South Pacific Design Automation Conference (ASP-DAC 2001). New York: IEEE Press, 2001. 391-395.

[7] Blum T, Paar C. Montgomery modular exponentiation on reconfigurable hardware[A]. Proceedings of the 14th IEEE Symposium on Computer Arithmetic (ARITH-14)[C]. New York: IEEE Press, 1999. 70-77.

[8] Liu Q, Ma F Z, Tong D, Cheng X. Efficient implementation of the RSA crypto processor in deep sub-micron technology [A]. Proceedings of the ICISA 2nd International Conference on Applied Cryptography and Network Security (ACNS2004)[C]. Beijing: ICISA Press, 2004. Technical Track, 106-114.

[9] Liu Q, Ma F Z, Tong D, Cheng X. A regular parallel RSA processor[A]. Proceedings of the IEEE 47th International Midwest Symposium on Circuits and Systems (MWSCAS 2004)[C]. New York: IEEE Press, 2004. 3: 491-494.

[10] Shang M, Vuillemin J. Fast implementations of RSA cryptography[A]. Proceedings of 11th IEEE Symposium on Computer Arithmetic[C]. New York: IEEE Press, 1993. 252-259.

[11] KOC C K. High-speed RSA implementation[R]. Bedford, MA, USA: RSA Laboratories, 1994.

[12] Liu Q, Tong D, Cheng X. A new systolic architecture without global broadcast[A]. Proceedings of the IEEE 7th International Conference on Signal Processing (ICSP'04)[C]. New York: IEEE Press, 2004. 1: 527-530.

[13] Mcivor C, Mcloone M, Mccanny J V. A high-speed, low latency RSA decryption silicon core[A]. Proceedings of the 2003 IEEE International Symposium on Circuits and Systems (ISCAS'03)[C]. New York: IEEE Press, 2003. 4: 133-136.

[14] Daly A, Marnane W. Efficient architectures for implementing montgomery modular multiplication and RSA modular exponentiation on reconfigurable logic[A]. Proceedings of the Tenth ACM International Symposium on Field-Programmable Gate Arrays (FPGA'02)[C]. New York: ACM Press, 2002. 44-49.

[15] Mcivor C, Mcloone M, Mccanny J V, et al. Fast montgomery modular multiplication and RSA cryptographic processor architectures[A]. Proceedings of the IEEE 37th Asilomar Conference on Signals, Systems, and Computers[C]. New York: IEEE Press, 2003. 1: 379-384.

[16] Cilardo A, Mazzeo A, Romano L, Saggese G P. Carry-save montgomery moduar exponentiation on reconfigurable hardware[A]. Proceedings of IEEE Design, Automation and Test in Europe Conference and Exhibition (DATE'04)[C]. New York: IEEE Press, 2004. 3: 206-211.


作者简介:


刘　强　男,1978年生于山东沂水,博士,现为 IBM 中国研究院高级工程师.2000年毕业于北京大学计算机系获理学学士学位,2005年毕业于北京大学计算机系获理学博士学位.作为主要技术人员参与完成多项国家级有关处理器核心技术的研究课题,主要研究方向为高性能微处理器、VLSI设计与验证、安全芯片、系统芯片、计算机运算、浮点处理.

E-mail: liuqiang@water.pku.edu.cn; qiangliu@cn.ibm.com

佟　冬　男,1971年生,博士,现为北京大学微处理器研究开发中心副教授,主要研究方向为计算机体系结构、可重构计算、互联网、存储系统、系统芯片设计.E-mail: tongdong@mprc.pku.edu.cn