

# 基于流水线技术的并行模幂算法硬件实现

黄世伟, 王云峰

(厦门大学电子工程系, 福建 厦门 361005)

**摘 要:** 针对 R-L 模幂算法并行硬件实现成本高的问题, 提出一种流水线形式的模幂运算结构。采用流水线技术对模幂算法中 Montgomery 模乘运算进行硬件设计, 并由此构建模幂运算结构, 实现并行模幂运算, 降低硬件成本。同时对模幂算法中预处理和后处理步骤进行优化, 以减少迭代次数。Virtex-2 系列现场可编程门阵列原型的实现结果表明, 在保证并行模幂运算速度的前提下, 该结构的硬件实现成本近似为传统并行结构的 1/2, 且数据吞吐率更高, 可达 14 Mb/s。

**关键词:** 蒙哥马利算法; 模乘; 模幂; RSA 公钥密码体制; 流水线技术; 现场可编程门阵列原型

## Hardware Implementation of Parallel Modular Exponentiation Algorithm Based on Pipelining Technique

HUANG Shi-wei, WANG Yun-feng

(Department of Electronic Engineering, Xiamen University, Xiamen 361005, China)

**【Abstract】** An efficient pipelined architecture is presented in this paper for solving the problem of high hardware cost of R-L modular exponentiation algorithm, which is formed of Montgomery modular multiplication built by using pipelining technique. The parallel calculation of algorithm can be executed and the hardware cost can be also reduced in the new architecture. Besides, two extra pre-processing and post-processing for converting an integer to its N-residue format in the conventional modular exponentiation algorithm are avoided to reduce the iteration time. The result shows that the new architecture can achieve high data throughput rate of more than 14 Mb/s on Xilinx Field Programmable Gata Array(FPGA) of Virtex-2 series when performs modular exponentiation, while occupy only about half hardware resources when compared with the conventional parallel architecture.

**【Key words】** Montgomery algorithm; modular multiplication; modular exponentiation; RSA public-key cryptosystem; pipelining technique; Field Programmable Gata Array(FPGA) prototype

**DOI:** 10.3969/j.issn.1000-3428.2013.07.004

### 1 概述

在 RSA 公钥密码体制中, 传统的模幂算法主要采用蒙哥马利(Montgomery)模乘算法, 通过串行或并行结构实现。串行结构节省面积但运算速度慢, 并行结构虽提高运算速度却占用较大面积, 而且 Montgomery 模乘算法中加法超长进位链问题是影响其最高运算速度的关键因素<sup>[1]</sup>。文献[1]提出的保留进位加法器(Carry Save Addition, CSA)结构虽能优化加法超长进位链, 但在每次模乘运算过程中需对中间结果进行额外的求和运算, 增加了时钟周期数, 降低了运算速度。文献[2]提出一种改进的串行模幂结构, 并采用串行的 L-R 模幂算法, 降低了硬件成本, 但运算速度仍有待提高。文献[3]对此做了优化, 虽提高了运算速度, 却同时增加了硬件实现成本。文献[4]提出一种基 4 的 Montgomery 模乘算法及其优化的硬件结构, 将传统的基 2 模乘运算迭代次数减少近一半, 但由于采用的是并行的模幂结构, 硬

件实现成本较大。文献[5]提出二维脉动阵列结构, 但由于采用全并行方式, 因此需要更高的硬件实现成本。

本文在对传统 R-L 模幂算法和 Montgomery 模乘硬件结构分析比较的基础上, 采用流水线技术对 Montgomery 模乘以及其组成的模幂运算结构进行设计实现, 并优化 R-L 模幂算法。

### 2 R-L 模幂和模乘算法及其结构分析

本文所用的 R-L 模幂算法如算法 1 所示, 其中, 步骤 3.1 和步骤 3.2 没有依赖关系, 故为并行算法, 其核心运算为 Montgomery 模乘运算。

#### 算法 1 传统 R-L 模幂算法

已知:  $X; N; E = \sum_{i=0}^{H-1} e_i \cdot 2^i; e_i \in \{0, 1\}$

计算:  $P = X^E \bmod N$ ; 其中,  $R = 2^{(n+2)}$

1.  $P_0 = \text{MontMul}(1, R^2)$ ;

**作者简介:** 黄世伟(1987 -), 男, 硕士研究生, 主研方向: 集成电路设计, 密码学; 王云峰, 副教授、博士

**收稿日期:** 2012-08-10 **修回日期:** 2012-09-11 **E-mail:** win\_hshiw@163.com

2.  $Z_0 = \text{MontMul}(X, R^2)$ ;
3. for  $i=0$  to  $H-1$  do
  - 3.1.  $Z_{i+1} = \text{MontMul}(Z_i, Z_i)$ ;
  - 3.2. if( $e_i=1$ ) then
    - $P_{i+1} = \text{MontMul}(P_i, Z_i)$ ;
  - else
    - $P_{i+1} = P_i$ ;
4.  $P = \text{MontMul}(P_H, 1)$ ;
5. return  $P$ ;

文献[4]提出一种基 4 的 Montgomery 模乘算法, 如算法 2 所示, 即一次迭代使用操作数中的 2 个位。

**算法 2** 基 4 的 Montgomery 模乘算法

已知:  $A = \sum_{i=0}^{n-1} a_i \cdot 2^i$ ;  $B_C = \sum_{i=0}^{n-1} b_{C_i} \cdot 2^i$ ;  $B_S = \sum_{i=0}^{n-1} b_{S_i} \cdot 2^i$   
 $N = \sum_{i=0}^{n-1} n_i \cdot 2^i$ , 其中,  $a_i, b_{C_i}, b_{S_i}, n_i \in \{0, 1\}$

计算:  $\text{MontMul}(A, (B_C+B_S)) = A \cdot (B_C+B_S) \cdot R^{-1} \bmod N$   
 其中,  $R = 2^{(n+2)}$

1.  $S_C^0 = 0; S_S^0 = 0$ ;  
 $B_C' = B_C \cdot 2^2 = \{B_C, 0, 0\}$ ;  
 $B_S' = B_S \cdot 2^2 = \{B_S, 0, 0\}$ ;
2. for  $j=0$  to  $(n/2+1)$  do
  - 2.1.  $q_0^j = S_{C_0}^j \oplus S_{S_0}^j$ ;  
 $q_1^j = S_{C_1}^j \oplus S_{S_1}^j \oplus S_{C_0}^j S_{S_0}^j \oplus (S_{C_0}^j \oplus S_{S_0}^j) n_1$ ;
  - 2.2.  $\{S_{CY}, S_{SUM}\} = (S_C^j + S_S^j + q_0^j N + q_1^j 2N) / 4$ ;  
 $\{Z_{CY}, Z_{SUM}\} = a_{2j+1} 2B_C + a_{2j+1} 2B_S + a_{2j} B_C + a_{2j} B_S$ ;
  - 2.3.  $\{S_C^{j+1}, S_S^{j+1}\} = \{S_{CY}, S_{SUM}\} + \{Z_{CY}, Z_{SUM}\}$ ;
3. return  $\{S_C^{n/2+2}, S_S^{n/2+2}\}$ ;

这种算法相对于基 2 的 Montgomery 模乘算法(1 次迭代只使用 1 个位)来说, 可以减少一半的迭代次数, 使运算速度提高 1 倍。本文研究也基于该算法。为使结果不大于  $N$ , 该算法采用增加一次迭代来避免额外的比较和减法运算<sup>[6]</sup>; 为简化  $q_1^j$  和  $q_0^j$  的组合逻辑, 缩短其关键路径, 该算法先对操作数  $B$  乘以  $2^2$ , 然后再增加一次迭代来消除对结果的影响<sup>[7]</sup>。因此, 完成该算法需要  $(n/2+2)$  个时钟周期。然而由于此算法的结果中含有参数  $R (R = 2^{(n+2)})$ , 在传统 R-L 模幂算法中需要步骤 1、步骤 2 和步骤 4 对模乘结果中的参数  $R$  作相应的预处理和后处理, 以保证模幂结果正确。因此完成一次模幂运算需要  $(H+2)$  次并行模乘运算(指最坏复杂度情况下, 下同), 共计  $(n/2+2)(H+2)$  个时钟周期。

图 1 为 Montgomery 模乘算法所对应的模乘单元结构<sup>[7]</sup>, 操作数  $A$  和  $B$  分别保存在寄存器  $A^{\text{shift}}$  与  $\{B_C, B_S\}$  内。其中, 灰色表示寄存器组(下同);  $A^{\text{shift}}$  为移位寄存器, 用于产生每次迭代所需的操作数  $A$  中的 2 个位:  $a_{2j+1}$  和  $a_{2j}$ 。 $\{S_C^j, S_S^j\}$  为中间数据暂存寄存器, 用于暂存模乘每次迭代的中间结果。

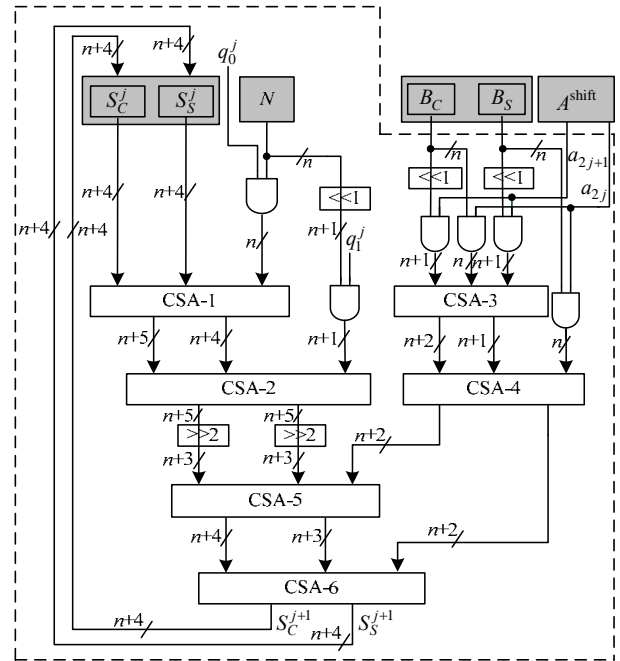


图 1 基 4 的 Montgomery 模乘算法硬件结构

传统的并行模幂运算结构通常采用 2 个 Montgomery 模乘单元来构建其运算单元, 此外还需要控制单元来控制运算单元实现模乘和模幂运算。根据上述 R-L 模幂算法和 Montgomery 模乘单元结构, R-L 并行模幂运算单元的寄存器传输级(Register Transfer Level, RTL)框图如图 2 所示。其中,  $Z_{load}$  和  $P_{load}$  为控制单元产生的控制信号。根据 R-L 算法,  $Z_{i+1} = \text{MontMul}(Z_i, Z_i)$  和  $P_{i+1} = \text{MontMul}(P_i, Z_i)$  的计算同时进行, 结果通过  $Z_{load}$  和  $P_{load}$  信号分别保存到  $Z$ (包括  $Z^{\text{shift}}$ ) 和  $P$  寄存器中, 如此完成 R-L 算法的一次迭代。但同时由图 2 可知, R-L 并行模幂运算单元主要通过 2 个 Montgomery 模乘单元并行实现, 包含 5 组数据寄存器, 2 个 CSA 链(每个由 6 个 CSA 构成), 其硬件实现成本较高。

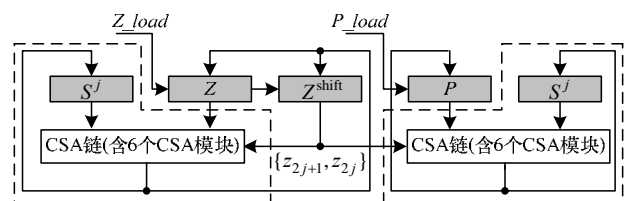


图 2 R-L 并行模幂运算单元的 RTL 级结构

**3 改进的并行模幂算法及其硬件设计**

针对传统 R-L 模幂算法中预处理和后处理步骤会增加运算时间的问题, 本文对此进行优化, 如算法 3 所示。优化后每次模幂运算减少 2 次模乘运算, 可提高模幂运算的数据吞吐率。

**算法 3** 改进后的 R-L 模幂算法

已知:  $X; N; E = \sum_{i=0}^{H-1} e_i \cdot 2^i$ ;  $e_i \in \{0, 1\}$

计算:  $P = X^E \bmod N$ ; 其中,  $R = 2^{(n+2)}$

1.  $P_0 = 1$ ;





### 4 性能分析与比较

本文在 Xilinx 公司 XC2V6000 的开发平台上实现了改进算法的现场可编程门阵列(Field Programmable Gata Array, FPGA)原型, 并且采用参数化的设计方法来满足不同密钥长度的需求, 使其可以实现 128 bit、256 bit、512 bit、1 024 bit 等甚至更长密钥的 RSA 加解密系统。

#### 4.1 仿真结果

一个 128 bit 模幂运算的仿真结果见图 8, 其表达式为:

$$26AC36AFDA6F2574B8F2DBA355BEFD5 \\ 26AC36AFDA6F2574B8F2DBA355BEFD5 \text{ mod} \\ 15D26E05E4594B726B77CE7D87C1CF9D = \\ 7F8E91770643CFA6D7F149CBE02BE42$$

图中最终结果为左下方的  $Pc\_reg+Ps\_reg$ , 或右下方的串行输出(从左到右为数据的低位到高位)。可见结果是正确的。在 270 MHz 的时钟频率下, 其最坏情况下( $H=128$ )的运算时间为 63.97  $\mu\text{s}$ 。

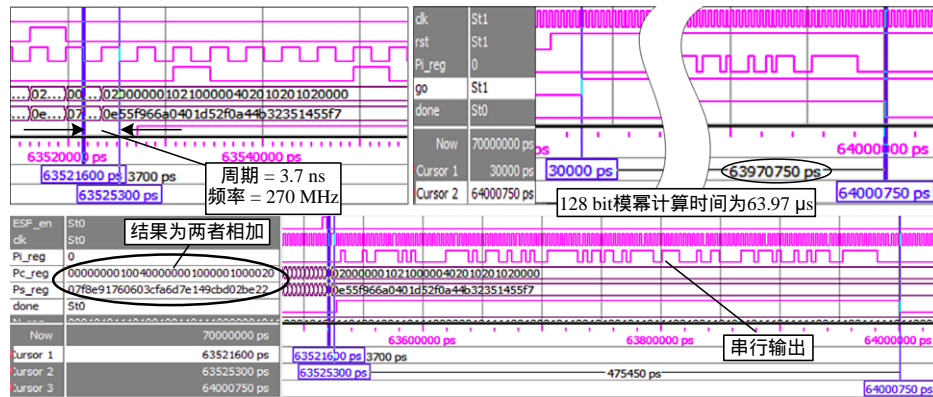


图 8 128 bit 模幂运算的仿真结果

#### 4.2 性能分析

表 2 显示了本文设计的模幂运算结构在最坏情况下 ( $H=n$ ) 的最高时钟频率、数据吞吐量、运算时间、所占资源等相关数据。表 3、表 4 为与其他文献的比较情况。其中, 文献[2-3]方案采用串行结构设计。

表 2 并行模幂运算的 FPGA:XC2V6000 原型实现结果

位长/bit	最高时钟频率 /MHz	吞吐量 / $(\text{Kb}\cdot\text{s}^{-1})$	运算时间 /ms	资源占用 /Slices
256	272.257	1 035.044	0.247	3 374
512	272.257	524.560	0.976	6 709
1 024	271.003	262.852	3.896	13 379

表 3 512 bit 模幂运算的 FPGA 实现结果比较

方案	所选器件	工作频率 /MHz	吞吐量 / $(\text{Mb}\cdot\text{s}^{-1})$	运算时间 /ms	资源占用 /Slices
文献[8]方案	XC2V3000	102.310	5.100	0.100	11 304
文献[9]方案	XC4VSX35	138.293	6.900	-	13 676
文献[2]方案	XC2V3000	168.380	9.280	-	6 294
文献[3]方案	XC2V3000	218.720	11.870	-	7 832
文献[4]方案	XC2V3000	143.700	14.850	-	11 558
本文方案	XC2V6000	272.257	14.191	0.036	6 709

表 4 1 024 bit 模幂运算的 FPGA 实现结果比较

方案	所选器件	工作频率 /MHz	吞吐量 / $(\text{Mb}\cdot\text{s}^{-1})$	运算时间 /ms	资源占用 /Slices
文献[8]方案	XC2V6000	95.900	4.790	0.210	23 208
文献[7]方案	XC2V6000	97.000	0.184	-	15 670
文献[9]方案	XC4VSX35	123.211	6.160	-	28 891
文献[2]方案	XC2V6000	152.490	8.440	-	12 537
文献[3]方案	XC2V6000	215.830	11.850	-	15 826
文献[4]方案	XC2V6000	126.700	13.190	-	20 040
本文方案	XC2V6000	271.003	14.194	0.072	13 379

串行结构对应于串行的 L-R 算法<sup>[1]</sup>, 虽然硬件实现上只需要 1 个模乘单元, 但 1 次模幂迭代需要计算 2 次模乘, 运算速度慢。因此, 文献[2]方案的硬件资源虽比本文设计的流水线结构略少, 但数据吞吐量近似只有其一半。文献[3]虽对此做了优化, 提高了数据吞吐量, 但硬件所占资源明显比本文的流水线结构多。文献[4]采用了改进的并行结构设计, 数据吞吐量与本文所设计的流水线结构相近, 但随着操作数位数的增加, 本文流水线结构的数据吞吐量优势逐渐体现, 而且硬件资源上只有并行结构的一半。由实现数据可以看出, 本文所设计的流水线形式模幂结构, 比传统的串行结构速度快, 比并行结构资源少, 而且在所有比较的结构中, 其模幂运算的吞吐量最高。

### 5 结束语

本文分析传统模幂运算的实现方式, 并在其基础上提出一种流水线形式的模幂结构。该结构的关键思路在于以流水线的形式来执行并行模幂算法, 这样既能保证并行算法的运算速度, 又能降低硬件成本; 而对并行算法的优化方面, 又进一步提高了模幂运算的速度, 增加了运算的数据吞吐量。数据结果表明, 该结构显著提高了模幂运算的整体性能, 使速度和面积达到了较好的平衡, 可较好地适用于设计高速的不同密钥长度的 RSA 公钥密码系统以及网络应用<sup>[10]</sup>。但本文设计并未考虑功耗的因素, 因此, 如何将其应用于对功耗要求较高的场合, 是下一步研究的重心。

(下转第 25 页)

输成本。给出面向简化路网模型的信息采集方法, 只考虑包括节点延时的路段行程时间, 使得采集内容简单化; 为保证路段行程时间的计算更加精确, 使用在路段节点临界区内估计到达节点时刻的方法, 并考虑了路段中途停车的影响。通过实验验证了该方法的可行性和有效性。但该方法无法与其他采集方法深入比较, 还需更广泛地进行基于GPS终端采集实验, 进一步研究更加准确的地图匹配方法。

#### 参考文献

- [1] 姜桂艳, 常安德, 吴超腾. 基于GPS浮动车的交通信息采集方法[J]. 吉林大学学报: 工学版, 2010, 40(4): 971-975.
- [2] Byon Y J. GPS-GIS Integrated System for Travel Time Surveys[D]. Toronto, Canada: University of Toronto, 2005.
- [3] Liu Chun, Meng Xiaolin, Fan Yeming. Determination of Routing Velocity with GPS Floating Car Data and Web GIS-based Instantaneous Traffic Information Dissemination[J]. The Journal of Navigation, 2008, 61(2): 337-353.
- [4] 胡明伟. 基于GPS的实时交通信息采集方法的研究[J]. 公路交通技术, 2007, (5): 121-125.
- [5] 张存保, 杨晓光, 严新平. 基于浮动车的交通信息采集系统研究[J]. 交通与计算机, 2006, (5): 31-34.
- [6] Li Q, Zhang T, Yu Y. Using Cloud Computing to Process Intensive Floating Car Data for Urban Traffic Surveillance[J]. International Journal of Geographical Information Science, 2011, 25(8): 1303-1322.
- [7] Zhu Zhengyu, Wang Lina, Li Jinyan, et al. A Simplified Road Network Model Considering Delay Time at Intersection and Its Application on Automobile Navigation[J]. Communications in Computer and Information Science, 2011, 158(1): 66-74.
- [8] Zhu Zhengyu, Liu Wei, Wang Lina, et al. A Simplified Real-time Road Network Model Considering Intersection Delay and Its Application on Vehicle Navigation[J]. Applied Mechanics and Materials, 2011, 58-60: 1959-1965.
- [9] Zhu Zhengyu, Li Jinyan, Liu Lin, et al. A Simplified Time-division Based on Road Network Model Considering Intersection Delay for Vehicle Navigation[J]. Applied Mechanics and Materials, 2011, 52-54: 1226-1232.
- [10] 朱征宇. 一种简化交通路网模型及基于此模型的导航方法: 中国, 201110002841.2[P]. 2011-01-07.
- [11] 朱征宇. 基于简化路网模型的实时路况监控方法: 中国, 201110217798.1[P]. 2011-08-01.
- [12] 张和生, 张毅, 温慧敏. 利用GPS数据估计路段的平均行程时间[J]. 吉林大学学报: 工学版, 2007, 37(3): 533-537.
- [13] 姜桂艳, 李继伟, 张春勤. 城市主干路路段行程时间估计的BPR修正模型[J]. 西南交通大学学报, 2010, 45(1): 124-129.

编辑 索书志

(上接第20页)

#### 参考文献

- [1] Kwon T W, You C S, Heo W S, et al. Two Implementation Methods of a 1 024-bit RSA Cryptoprocessor Based on Modified Montgomery Algorithm[C]//Proc. of ISCAS'01. Sydney, Australia: IEEE Press, 2001.
- [2] Shieh Ming-Der, Chen Jun-Hong, Wu Hao-Hsuan, et al. A New Modular Exponentiation Architecture for Efficient Design of RSA Cryptosystem[J]. IEEE Transactions on Very Large Scale Integration(VLSI) Systems, 2008, 16(9): 1151-1161.
- [3] Shieh Ming-Der, Chen Jun-Hong, Wu Hao-Hsuan, et al. A New Algorithm for High-speed Modular Multiplication Design[J]. IEEE Transactions on Circuits and Systems, 2009, 56(9): 2009-2019.
- [4] 薛念, 潘赞, 张宇弘, 等. 基于Montgomery模乘的RSA加密处理器[J]. 计算机工程, 2010, 36(13): 125-127.
- [5] Walter C D. Systolic Modular Multiplication[J]. IEEE Transactions on Computers, 1993, 42(3): 376-378.
- [6] 王旭, 董威, 戎蒙恬. 基于改进Montgomery模乘算法的RSA加密处理器的实现[J]. 上海交通大学学报, 2004, 38(2): 240-244.
- [7] 张远洋, 李峥, 杨磊, 等. 一种新型的基于Montgomery的模幂器结构[J]. 计算机工程, 2007, 33(16): 211-213.
- [8] Mcivor C, Mcloone M, Mccanny J V. Modified Montgomery Modular Multiplication and RSA Exponentiation Techniques[J]. IEE Proceedings-Computers and Digital Techniques, 2004, 151(6): 402-408.
- [9] Hu Zhengbing, Shboulr R M, Shirochin V P. An Efficient Architecture of 1 024-bits Cryptoprocessor for RSA Cryptosystem Based on Modified Montgomery's Algorithm[C]//Proc. of IDAACS'07. Dortmund, Germany: [s. n.], 2007.
- [10] Stallings W. 密码编码学与网络安全——原理与实践[M]. 3版. 刘玉珍, 译. 北京: 电子工业出版社, 2004.

编辑 金胡考