文章编号:1006 - 9348(2008)05 - 0101 - 04

运算精简的蒙哥马利算法模乘器设计

蒋晓娜,段成华

(中国科学院研究生院信息科学与工程学院,北京 100049)

摘要:针对 Montgomery算法的可伸缩脉动阵列模乘协处理器的硬件实现中,速度和面积没有取得很好平衡的问题,结合 Walter等学者对 Montgomery算法的分析,利用 EDA仿真分析工具,提出一种运算精简的蒙哥马利算法模乘器设计方法。该方法通过先分析已有 Montgomery算法,得到运算精简蒙哥马利算法,然后将该算法映射到可伸缩脉动阵列结构,使模乘器在速度和面积上能够取得很好的平衡。最后进行仿真实验验证,结果证明该方法解决了模乘器速度和面积平衡的问题。通过该方法设计的模乘器,用 TSMC 0.18μm标准单元库综合,核心运算单元最高时钟频率可达 385MHz,等效单元 1.2k等效门。与现有其他方法相比,该模乘器在平衡方面取得较好性能,可以拓展其在移动通信领域的应用。

关键词:蒙哥马利算法;可伸缩脉动阵列;公钥密码体制

中图分类号:TN47 **文献标识码**:B

Operation Reduced Montgomery's Algorithm Modular Multiplier Design

JIANG Xiao - na, DUAN Cheng - hua

(School of Information Science and Engineering, GUCAS, Beijing 100049, China)

ABSTRACT: In view of the problem that area ×time factor of modular multiplication coprocessor, when using Scalable Systolic Array circuit scheme to implement Montgomery algorithm, can not get the trade - offs, and by combining the available typical Montgomery Modular algorithms and using EDA simulation tools, an operation reduced radix 2 - Montgomery algorithm Modular Multiplier design scheme is developed. In this scheme, in order to get the trade - offs of area ×time factor, map an operation reduced radix 2 - Montgomery algorithm, which is developed by combining the available typical Montgomery Modular algorithms, to a Scalable Systolic Array. And the simulation result verifies the scheme. The result shows that based on the TSMC 0.18μm CMOS technology, area of the Modular Multiplier is about 1.2k equivalent - gate, the key cell frequency can up to 385MHz. Compared with other existing solutions, this modular multiplication coprocessor has advantage in terms of area ×time factor, and can be used widely in mobile communication field.

KEYWORDS: Montgomery algorithm; Scalable systolic array; Public key schemes

1 引言

随着电子商务、电子政务、金融电子化和网络通信的蓬勃发展,信息安全问题成为一个热点话题,为了保证信息保密性、完整性、可用性和抗抵赖性,公钥加密系统被广泛应用。模乘运算是公钥加密系统中的核心运算,因此模乘运算实现是公钥加密系统的关键所在,模乘器实现的性能直接决定了系统的性能。

目前,基于 MMM (Montgomery Modular Multiplication)算法的脉动阵列的硬件实现是普遍采用的一种实现大数模乘运算的方法。其硬件实现主要有两种:一种是高基 MMM算法的脉动阵列模乘器设计的研究;另一种是基 2 - MMM算法的脉动阵列模乘器设计的研究。其中前者由于实现时硬件结构非常复杂,在某些应用中不适合,没有受到广泛的重视;后者是由英国学者 Walter[1]于 1993年提出的,由于将 MMM算法很好地映射到二维脉动阵列结构上,有效提高了模乘运算的速度,同时硬件实现简单,因此受到众多学者的关注。但是在以往的研究中,主要集中在模乘器的快速实现上,因此大多数现有基 2 - MMM算法脉动阵列模乘器虽然速度可以满足加、解密系统的要求,但是硬件实现的规模很大,很难满足对速度和面积都有要求的应用的需要。

本文首先对 MMM算法进行了介绍,通过结合 Walter等学者[1][3][4]对 MMM算法的分析,得到运算精简基 2 - MMM算法,并提出一种基于运算精简基 2 - MMM算法可伸缩脉动阵列大数模乘协处理器的设计方法;随后利用仿真工具,对改进的算法和模乘协处理器的性能进行了分析。结果表

基金项目:部分受国家"863计划"项目资助 (2002AA141041)

收稿日期:2007 - 04 - 24 修回日期:2007 - 05 - 24

明运算精简基 2 - MMM 算法与可伸缩脉动阵列结合,使模乘器在速度和面积性能上取得平衡。最后介绍了基于该方法的设计在实际中的应用。

2 Montgomery算法及其改进

P. L. Montgomery提出的 MMM 算法采用模加和右移的方法避免了通常求模算法中费时的除法操作,被认为是计算大数模乘最有效的算法。

2.1 Montgomery算法

MMM 算法给出求解 $A \times B \times R^{-1} \bmod M$ 的快速方法,通过一定的预运算和后运算得到真正形如 $S = A \times B \bmod M$ 模乘运算结果。

算法 1为原始的 MMM 算法,其中 $T = A \times B$ 且有 $0 \leq T < RM$,M为模数且 $M > 1$,R是满足式 (1)的一个基 (通常取 R为 2^n, n为输入大数的位数):

$$\gcd(M, R) = 1 \tag{1}$$

R^{-1}和 M 是满足式 (2)和式 (3)的数:

$$R R^{-1} \pmod M = 1 \tag{2}$$

$$R R^{-1} - MM' = 1 \tag{3}$$

算法 1: *function REDC(T)*

$REDC(T, M) = T \times R^{-1} \bmod M$

$m = (T \bmod R) M' \bmod R;$

$P = (T + mM) / R;$

if $P > M$ *then* $P - M$ *else return* $P;$

算法 1中每次模乘运算总有 $A \times B$, TM' 和 mM三次大数乘法运算,当 A, B和 M均为 1024 - bit以上的大整数时,带有大数乘法的模乘硬件实现仍然是十分困难的,因此 Montgomery提出了变形的 MMM 算法。

在变形的 MMM 算法中,大整数 A, B和 M以 r为基表示,通常 $r = 2^w$, w称为字长,则对于输入是 n位的大整数有:

$$A = \sum_{i=0}^{n-1} a_i r^i, B = \sum_{i=0}^{n-1} b_i r^i, M = \sum_{i=0}^{n-1} m_i r^i$$

在变形的 MMM 算法中,对于式 (3) 相应地有:

$$R R^{-1} - MM' \equiv 1 \pmod{r^n} \tag{4}$$

由式 (4) 易得, $M' \bmod r = (-M)_r^{-1} = (r - M[0])_r^{-1}$。在变形的 *MMM* 算法中,使用了循环,每次循环对大数中字长的 1位进行运算,因而较之算法 1利于硬件实现。

2.2 基 2 - MMM 算法

目前,大数模乘运算的硬件实现普遍采用脉动阵列结构,当选取上述变形算法中的基 r为 2时,因为 M是奇数 (由式 (1) 易知),所以 $M[0] = 1$,则式 (4) 中的 $(r - M[0])_r^{-1} = 1$,得到基 2 - *MMM* 算法,见算法 2。

$$A = \sum_{i=0}^{n-1} a_i 2^i, B = \sum_{i=0}^{n-1} b_i 2^i, M = \sum_{i=0}^{n-1} m_i 2^i$$

算法 2: function 2 MontPro(A, B, M)

MontPro$(A, B, M) = A \times B \times 2^{-n} \bmod M$

$P = 0;$

For $i = 0$ *to* $n - 1$

$Q[i] = (P[0] + a_i \times b_0) \bmod 2;$

$P = (P + a_i \times B + Q[i] \times M) \operatorname{div} 2;$

采用基 2 - MMM 算法的脉动阵列模乘器,每个 PE完成的是 1 - bit数据的运算,同时右移一位的操作就是除 2。因而,模乘器不仅可以工作在高时钟频率下,而且降低了硬件设计复杂度。

2.3 改进的基 2 - MMM 算法

MMM 算法作为求解模乘运算普遍采用的算法,被众多学者研究和改进。为了解决可伸缩脉动阵列 Montgomery模乘器面积和速度平衡问题,在重点研究了 Walter 等学者[1][3][4]对 MMM 算法理论分析结果基础之上,得到运算精简基 2 - MMM 算法。

运算精简基 2 - MMM 算法对 MMM 算法做了以下三点改进:

改进点一、简化求解过程。MMM 算法引入求 Q[i],用于保证 P是整除的结果,原始算法中该步运算降低了 MMM 算法的效率。在满足计算 P时序要求的同时,为了有效提高算法效率,运算精简算法根据 Eldridge和 Walter[4] 提出的简化求解 $Q[i]$的方案,将乘数 B左移一位,且 $b_0 = 0$,这样 $Q[i]$值仅与 $P_{[i]}[0]$有关,初始时 $P_{[0]}[0] = 0$,故 $Q[0] = 0$。

改进点二、减少加法操作次数,简化关键运算。MMM 算法关键运算是 P的求解,可以通过判断 a_i和 $Q[i]$当前周期的值[3],直接得到 $a_i \times B[j] + Q[i] \times M[j]$的运算结果。对于判断结果中的 $M[j] + b_j$的值,保证在模乘运算执行到该处时,此数据已获得前提下,可以在模乘任何时间处理得到且只需运算一次,所以每次循环中求解 P只包括一个选择和一个 1 - bit加法操作,大大减少了模乘运算中的加法次数。

改进点三、消除减法操作。原始算法中,当 $P \geq M$,为使本次模乘运算的结果可以直接作为下次模乘运算的输入,需要多做一次减法运算,使 P满足 $0 < P < M$。该减法操作,既增加了运算步骤,又使模乘运算可能受到差分能量分析等的攻击,安全性将降低。为了消除额外的减法操作,被乘数 A和乘数 B须满足: $A < 2M$且 $a_n = 0, B < 2M$且 $b_{n+1} = 0, M[n+1] = M[n] = 0$[1][4],易证运算结果 $P < 2M$。这里增加 a_n,即增加了一次循环运算,从而保证输出结果 $P < 2M$,而增加 b_{n+1}和 $M[n+1], M[n]$是为了保证运算过程中, P的中间结果中的信息没有丢失,保证最后模乘运算结果的正确性。

function Modi_MontPro_2(A, B, M)

Modi_MontPro_2$(A, B, M) = A \times B \times 2^{-n-1} \bmod M$

$A: a_n, a_{n-1}, \cdots, a_1, a_0$且 $a_n = 0;$

$B: b_{n+1}, b_n, b_{n-1}, \cdots, b_1, b_0$且 $b_0 = b_{n+1} = 0;$

$M: m_{n+1}, m_n, m_{n-1}, \cdots, m_1, m_0$且 $m_0 = 1, m_{n+1} = m_n = 0;$

$S = B + M;$

$P[0] = 0;$

for $i = 0$ *to* n

$Q[i] = (P_{[i]}[0] + a_i \times b_o) \bmod 2;$

$$
\begin{aligned}
&for\ j = 0\ to\ n+1\\
&switch\ a_i, Q[i]\{\\
&\quad 1,1: Mux[j] \rightarrow s_j;\\
&\quad 1,0: Mux[j] \rightarrow b_j;\\
&\quad 0,1: Mux[j] \rightarrow m_j;\\
&\quad 0,0: Mux[j] \rightarrow 0;\\
&\}
\end{aligned}
\tag{5}
$$

$$
P_{[i+1]}[j-1] + Ca2_{[i]}[j] = P_{[i]}[j] + Mux[j] + Ca2_{[i]}[j-1] \tag{6}
$$

运算精简基 2-MMM算法,完成一次 n-bit模乘运算,只需 1次 n-bit和 (n+1)(n+2)次 1-bit加法操作,及 (n+1)(n+2)次选择操作。该算法简化了求解过程,降低了关键运算求解的复杂度,大大减少了加法操作次数。在可伸缩脉动阵列模乘器设计中,需要多次输入 B+M的结果 (如 1024-bit模乘运算,次数为 32次),如果采用改进的模乘算法将减少 31次 1024-bit加法运算,可以提高模乘运算速度,有利于解决模乘器面积和速度的平衡问题。

3 模乘协处理器硬件实现

3.1 可伸缩脉动阵列

将二维退化脉动阵列分割,保持两端的 PE[0]和 PE[n]不变,中间的 PE[1]到 PE[n-1]处理单元分成每 k个处理单元一组,不同的组使用相同的 k个处理单元,构成资源复用的可伸缩脉动阵列[6],迭代地完成 n-bit的 MMM运算。复用单元个数 k的选取,可根据实际应用的不同,在设计初期进行选择。这样,可伸缩脉动阵列既具有脉动阵列的性能优势,又兼有可伸缩的灵活性。

3.2 可伸缩脉动阵列的处理单元

根据运算精简基 2-MMM算法,得到 PE电路原理图,如图 1所示。其中由 1个 FA和一个 4-1Mux实现 PE的逻辑运算,即实现运算精简算法中的式 (5)和式 (6);移位寄存器用于保存中间运算结果,移位寄存器的长度,对于普通处理单元为 n/(k-1),其中 n为输入数据的位数,k为迭代的 PE的数目;计算结果在时钟上升沿从触发器输出到下一处理单元。不难看出,改进的 PE电路结构简单,关键路径的延时较小。在可伸缩脉动阵列模乘器中,系统时钟频率受限于 PE的关键路径,因此,基于改进的模乘算法的可伸缩脉动阵列模乘器可以工作在较高的时钟频率下,从而可以提高模乘运算的速度。

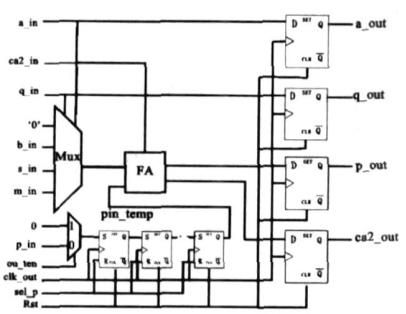

图 1 普通处理单元 PE[i]原理图

3.3 模乘协处理器

模乘协处理器作为安全处理器的协处理单元,用来进行运算强度大的模乘运算。本文提出的模乘协处理的系统架构如图 2所示,主要包括五部分:输入单元、模乘运算单元、输出单元、时钟单元和控制单元。其中:输入单元:输入单元接收并为参与运算的数据提供数据通道,特别包括一个完成 n-bit加法运算的加法处理模块;模乘运算单元:模乘器的核心运算单元,完成大数模乘运算;输出单元:将运算结果串行输出;控制单元:协调各单元按时序工作;时钟单元:为各单元提供需要的时钟。

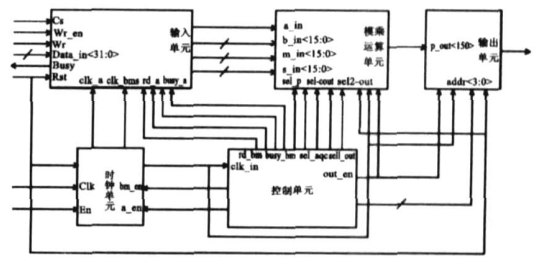

图 2 模乘协处理器系统架构

4 仿真分析

4.1 仿真验证

用 C语言执行扩展欧几里德算法对运算精简基 2-MMM算法做了验证,用 Verilog语言描述了基于运算精简算法的线性脉动阵列模乘器并用 ModelSim SE 6.0d对其进行了功能仿真,与 C语言执行运算精简算法输出结果比较,通过多组数据的比较,结果一致,验证了算法的正确性。

限于篇幅,这里给出 n=4模乘运算的验证过程。欲求 A=11,B=10,M=15的模乘运算结果,根据算法则:输入 A={01011}=11,B={010100}=20,M={001111}=15,验证过程包括以下步骤:

步骤 1:利用扩展欧几里德算法求得:

$R(2^4)$的模 M乘法逆元 R^{-1}为 1,

则 $P = 11 \times 10 \times 1 \bmod 15 = 5$;

$R(2^5)$的模 M乘法逆元 R^{-1}为负 7,

则 $P = 11 \times 20 \times (-7) \bmod 15 = 5$;

步骤 2:利用 C语言实现的算法 4求得的结果:P=20;

步骤 3:利用 ModelSim SE 6.0d对模乘器进行仿真,图 3为仿真结果,从图 3中可以看到,在 T=14时结果 P=20输出并保持一个时钟周期的时间。20 mod 15 =5,所以易知四个输出数据是等价的,同时输出结果满足 P<2M即 20<30,当输入多组不同数据进行仿真验证时,可得到相同的结论,从而验证算法及其线性脉动阵列模乘器功能的正确性。

4.2 仿真性能分析

基于运算精简基 2-MMM算法的可伸缩脉动阵列模乘

图 3　4 - bit模乘运算仿真图

器的设计,即模乘协处理器中模乘运算单元,借助 EDA仿真工具,将改进算法映射到可伸缩脉动阵列结构。仿真结果表明,该模乘器融合了二者的优势,求解过程中简化了求解 Q[i]的运算,避免了最后除法操作,求解过程中只需进行一次 B+M运算,PE仅由 1 - bit FA和 1个 4 - 1 Mux构成,完成一次 n - bit模乘运算过程需 (2n/k) n + k个时钟周期,模乘运算单元规模为 (k + 2) PE(这里 k值可以根据实际应用需要,在设计初期选取)。结果还表明,比较好的配置方案是选择 k的值为主处理器数据总线宽度的一半,对于 32位机,选取 k = 16时,模乘运算单元最高时钟频率可达 385MHz,系统时钟频率可达 170MHz,等效单元 1.2k等效门。因此,基于运算精简算法的大数模乘器设计方法,在面积和时间上取得了很好的平衡,在实现上有很大的灵活性。

表 1　MMM算法和运算精简算法可伸缩脉动阵列综合结果

设计方案	PEs	Area	Tmin	Nmul	Tmul
陈强[6]	18	1.4k	4.8ns	128n + 16	629μs
ours	18	1.2k	2.6ns	128n + 16	341μs

其中:Tmul表示完成一次模乘运算所需要的时间;

Area模乘运算单元的硬件实现等效门数。

4.3　实际应用

密码技术,特别是高性能的安全处理器实现,代表着一个国家在信息安全领域的水平,也是保障国家信息安全的关键。公开密钥密码编码学是目前广泛应用的密码技术,而模乘是公开密钥密码编码学中的核心运算。采用本文所述的模乘运算单元的协处理器硬件实现面积很小,同时运算速度也能满足一般应用要求,可广泛应用与各种安全或认证领域,尤其适合对面积和速度综合性能有所要求的场合,如移动通信领域等。

密码技术,特别是加、解密技术是信息安全中的核心技术,国家关键基础设施中不可能引进或采用别人的加密技术,只能自主研发。因此,本文的模乘协处理器硬件实现方法,具有自主知识产权,可广泛应用于关系国家安全的基础设施中。

5　结论和展望

综上所述,本文采用改进的基 2 - MMM算法,得到改进 PE电路。采用改进 PE的可伸缩脉动阵列模乘器与原有 CSSA[6]相比,有效降低了设计复杂度,解决了面积 - 时间的平衡问题。同时看到,模乘器可作为安全处理器的协处理单元可实现 RSA和 ECC两种主流的加密算法,同时具备硬件可配置的特点,这也是下一步研究的重点。

参考文献:

[1] C. D. Walter. Systolic modular multiplication[J]. IEEE Transactions on Computers, 1993, 42(3): 376 - 378

[2] P. Kornerup. A systolic, linear - array multiplier for a class of right - shiftalgorithms[J], Computers, IEEE Transactions on Computers, 1994, 43(8): 892 - 898

[3] Nedjah, N. de Macedo Mourelle, L, Three hardware architectures for the binary modular exponentiation: sequential, parallel, and systolic [J], IEEE Transactions on [C]ircuits and Systems I: Fundamental Theory and Applications 2006, 53(3): 627 - 633

[4] S. E. Eldridge and C. D. Walter. Hardware implementation of Montgomery's modular multiplication algorithm[J]. IEEE Transactions on Computers, 1993, 42(6): 693 – 699.

[5] C. D. Walter. Improved Linear Systolic Array for Fast modular Exponentiation[J]. IEE Proc. - Comput Digit Tech, 2000, 47(5): 323 - 328

[6] 陈强,段成华。CSSA——低功耗 Montgomery模乘的环形脉动阵列[J]。微电子学与计算机,2005, 22(8): 44 - 47

[作者简介]

蒋晓娜(1981 -),女(汉),辽宁人,硕士研究生,主要研究方向:安全协处理器设计;

段成华(1962 -),男(汉),重庆人,教授,主要研究方向:安全处理器,高级综合。

(上接第 80页)

[4] V P Jilkov, X R Li. Online bayesian estimation of transition probabilities for markovian jump systems[J]. IEEE Trans on Signal Processing, 2004, 52(6): 1620 - 1630.

[5] 席裕庚, 王凡. 非线性系统预测控制的多模型方法[J]. 自动化学报, 1996, 22(4): 456 - 461.

[6] N Gordon, D J Salmond, A F M Smith. Novel approach to nonlinear and non - Gaussian Bayesian state estimation[C]. IEE Proceedings - F, 1993, 140(2): 107 - 113.

[作者简介]

陆　可(1980 -),男(汉族),浙江宁波人,博士研究生,主要研究方向:电机控制、状态估计、参数辨识;

肖　建(1950 -),男(汉族),湖南衡阳人,教授,博导,主要研究方向:计算机控制、模糊控制、交流传动控制。