面向多曲线的通用高性能ECC处理器设计

刘志伟[1],刘雷波[1,2],黄 海[1],张 琦[1],于 斌[1],赵石磊[1],崔健博[1]
(1. 哈尔滨理工大学,黑龙江哈尔滨 150080;2. 清华大学集成电路学院,北京 100084)

摘 要: 该文针对广泛应用的TLS1.3协议,提出了一种高性能的椭圆曲线密码处理器. 该处理器支持TLS1.3协议中定义的两类素数域椭圆曲线的通用模数. 通过对高基蒙哥马利算法的改进,提出了一种支持521 bit及以下位宽的模乘运算单元,并提出了一种双模乘单元并行结构的标量乘法器. 基于该结构在两类椭圆曲线下设计了雅阁比坐标系下并行的点运算时序排布,使模乘单元的利用率在不同点运算情况下达到100%,95.4%和86.5%. 与现有设计相比,本文中标量乘法运算消耗的周期更少,运算单元利用率更高,在相似的时间面积乘积前提下,具有更强的通用性和可配置性的优势. 在TSMC 55 nm CMOS工艺下达到454 MHz的时钟频率,等效逻辑门数851k,Secp256r1曲线的标量乘运算速度为31 230 times/s.

关键词: 椭圆曲线密码;多曲线;通用模数模乘;蒙哥马利模乘;安全传输层协议

基金项目: 国家重点研发计划"光电子与微电子器件及集成"重点专项子课题(No.2018YFB2202100);黑龙江省自然科学基金优秀青年项目(No.YQ2019F010);黑龙江省普通高校基本科研业务费专项资金(No.2019KYYWF0214)

中图分类号: TN492;TP309 **文献标识码:** A **文章编号:** 0372-2112(2023)06-1562-10

电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210967

Multi-Curve-Oriented General High-Performance ECC Processor Design

LIU Zhi-wei[1], LIU Lei-bo[1,2], HUANG Hai[1], ZHANG Qi[1], YU Bin[1], ZHAO Shi-lei[1], CUI Jian-bo[1]
(1. *Harbin University of Science & Technology*, *Harbin*, *Heilongjiang* 150080, *China*;
2. *School of Integrated Circuits*, *Tsinghua University*, *Beijing* 100084, *China*)

Abstract: This paper proposes a high-performance elliptic curve cryptographic processor for the widely used TLS1.3 protocol. The processor supports two types of elliptic curve with general modulus in prime field defined in TLS1.3 protocol. Firstly, by modifying the high-radix Montgomery algorithm, a modular multiplication unit is proposed, which supports less than 521-bit width operators. Secondly, a parallel scalar multiplier structure with dual-modular multipliers is proposed. Based on this structure, to make full use of the two modular multipliers, a series of point operation timing arrangement is proposed in Jacobian coordinate, which supports two types of elliptic curves, makes the utilization rate of the modular multiplication unit reach 100%, 95.4% and 86.5% under different types of point operations. Compared with the existing design, the scalar multiplier in this work has a less cycle cost and less time consumption, as well as stronger versatility and configurability with similar time-area products. Under TSMC 55 nm CMOS technology, the clock frequency reaches 454 MHz. The scalar multiplication costs 851k equivalent logic gates, and the calculation speed of Secp256r1 curve is 31 230 times/s.

Key words: elliptic curve cryptography; multi-curve; general modulus multiplication; montgomery modular multiplication; transport layer security

Foundation Item(s): National Key Research and Development Program (No.2018YFB2202100); Natural Science Foundation of Heilongjiang Province of China (No.YQ2019F010); Special Fund for Basic Scientific Research Business Expenses of Ordinary Colleges and Universities in Heilongjiang Province of China (No.2019KYYWF0214)

收稿日期:2021-07-26;修回日期:2022-06-10;责任编辑:孙瑶

1 引言

椭圆曲线密码算法借助其更高的安全性,成为公钥体制密码算法中最为重要的算法之一. 更高的安全性则需要更为复杂的运算复杂度来支撑. 服务器端通常负载着大量的公钥签名、验签及密钥生成、密钥交换等任务,需要高性能的椭圆密码加速芯片以满足其大规模的运算需求.

从应用层面来看,椭圆曲线主要用于互联网传输层安全协议(Transport Layer Security, TLS). 最新的TLS1.3版本[1]包含了由美国国家标准与技术研究所(National Institute of Standards and Technology, NIST)定义的secp256r1, secp384r1和secp521r1[2], RFC-7748协议[3]中定义的Curve25519和Curve448,以及我国的商用密码标准SM2等. 可见,仅为兼容TLS1. 3协议即需对7种以上的椭圆参数进行支持. 椭圆曲线密码处理器中,模乘运算单元的设计是关键. 模乘运算单元分为针对特定椭圆曲线参数(固定P值)和通用曲线参数(通用P值)两种. 针对特定曲线参数的模乘运算单元可通过快速模约减运算降低运算量,从而获得更高的性能. 然而椭圆曲线密码的应用往往需同时对多种曲线进行支持. 为了获得较高的兼容性和通用性,选用通用P值的模乘单元是最佳的选择. 然而,通用P值的模乘单元会在性能、面积上有一定的损失. 因此,为了满足服务器端大量的签名、验签需求,需设计具有高性能的通用椭圆曲线密码处理器.

提升椭圆曲线密码处理器中标量乘法的运算速度并对多种曲线进行兼容设计一直是本领域的研究重点. 然而近年来的设计都尽力降低面积时间积,而未对高性能的服务器应用进行优化. Ananyi等人[4]使用乘法器和快速模约减的结构对NIST下的素数域椭圆曲线进行了兼容设计,并在FPGA中实现. Loi等人[5]同样使用FPGA对NIST兼容的5条椭圆曲线标量乘法电路进行了优化,设计了一个小面积的ECC处理器. 虽然其模运算单元的利用率较高,但为了节省面积使用了单一的乘法单元,限制了运算性能. Shah等人[6]的设计使用了4个radix-4的模乘单元,造成了较大的面积,而利用率却不能达到较高的水平;RSD和CSA结构组成的乘法单元并没有使模乘电路的关键路径延迟降低很多. Ding等人[7]使用Karatsuba乘法器和改进的快速模约减进行模乘运算,实现了对NIST曲线的支持,文献[8]在文献[7]的基础上进行改进,使用了Toom-Cook乘法器进一步节省了小位宽乘法次数,并结合点加倍点的计算公式对乘法器和约减加法阵列进行了较为优化的时序排布,使其并行度得到提高. 但现有的研究多数为了提高速度仅支持特定模数的运算,而对其他椭圆无法兼容,严重地限制了其应用范围,通用性较差. 此外,已有的设计绝大多数为节省面积的设计,很难适用于高性能服务器.

本文提出一种素数域ECC处理器,适用于运算服务器等高性能应用需求的场景,使其兼容TLS1.3协议中所有素数域曲线,并可通过配置兼容521 bit以下通用的椭圆曲线参数. 设计使用了双模乘运算单元结构,并对点加倍点时序进行了针对此结构的时序排布,以提高两模乘单元的利用率和计算并行度. 设计还使用了改进的蒙哥马利模乘算法,以及雅阁比坐标系下的点加倍点公式和NAF标量乘算法.

2 研究背景

2.1 椭圆曲线密码

椭圆曲线有很多种,使用较多的是NIST标准中使用的Weierstrass曲线,以及RFC-7748协议中使用的Montgomery曲线.

Weierstrass满足曲线方程,即

$$y^2=x^3+ax+b \tag{1}$$

TLS1.3中选用由NIST提出的secp192r1, secp224r1, secp256r1, secp384r1, secp521r1这5条曲线,就属于Weierstrass曲线.

还有一类椭圆曲线是Montgomery曲线,其方程为

$$y^2=x^3+Ax^2+x \tag{2}$$

RFC-7748中涉及的Curve25519和Curve448曲线属于Montgomery曲线,由Bernstein等人[9]于2006年提出,并在2016年成为国际标准[3]后也被TLS1.3采用. 其标量乘运算过程本身具有抗SPA特性,所以一般直接使用蒙哥马利阶梯算法而不会做修改. 不同种类的椭圆曲线对应着不同的点运算公式. 本文为了提高通用性,支持了最常用的两类椭圆曲线,对于每类椭圆曲线参数可配.

2.2 点运算

由于点加和倍点操作各需要进行一次模逆运算,而模逆在模运算当中最为耗时. 引入射影坐标系进行椭圆曲线的点运算可消除计算过程中的模逆运算,而只需在计算结束时,从射影坐标系转换回仿射坐标系时,进行一次模逆运算,从而提升标量乘的计算效率.

基于素数域的椭圆曲线常用的射影坐标系有标准(Standard)射影坐标系$(X,Y,Z)\leftrightarrow(x,y)=(X/Z,Y/Z)$, Jacobian射影坐标系$(X,Y,Z)\leftrightarrow(x,y)=(X/Z^2,Y/Z^3)$, Chudnovsky射影坐标系$(X,Y,Z,Z^2,Z^3)\leftrightarrow(x,y)=(X/Z^2,Y/Z^3)$, Modified Jacobian坐标系,以及二元域常用的López&Dahab坐标系$(X,Y,Z)\leftrightarrow(x,y)=(X/Z,Y/Z^2)$等[10,11]. 不同的坐标系下,点加和倍点运算的计算复杂度不同,需根据标量乘法调用点加和倍点的平均次数考量. 此外,在并行的设计中更需讨论公式的数据依赖

关系以提高并行度. 对于高性能的应用场景,本文使用Jacobian射影坐标系. 仿射坐标系转换至雅阁比坐标系:$(x,y,1)\rightarrow(X,Y,Z)$. 雅阁比坐标系转换至仿射坐标系:$(X/Z^2, Y/Z^3)=(x,y)$.

Jacobian坐标系下的点加公式如下:

$$\begin{aligned} X_3&=-H^3-2U_1H^2+r^2\\ Y_3&=-S_1H^3+r(U_1H^2-X_3)\\ Z_3&=Z_1Z_2H \end{aligned} \tag{3}$$

其中,

$$\begin{aligned} &U_1=X_1Z_2^2,\ U_2=X_2Z_1^2\\ &S_1=Y_1Z_2^3,\ S_2=Y_2Z_1^3\\ &H=U_2-U_1\\ &r=S_2-S_1 \end{aligned}.$$

Jacobian坐标系下的倍点公式为

$$\begin{aligned} X_3&=T\\ Y_3&=-8Y_1^4+M(S-T)\\ Z_3&=2Y_1Z_1 \end{aligned} \tag{4}$$

其中,

$$\begin{aligned} &S=4X_1Y_1^2,\ M=3X_1^2+aZ_1^4\\ &T=-2S+M^2,\ a=-3 \end{aligned}.$$

2.3 模乘算法

模乘电路的优化主要考虑三个方面. 其一是结合ECC处理器的应用场景,选取一个面积、性能适合的标量乘算法. 其二是结合调用模乘单元的点加倍点运算单元,设计符合上层控制器时序排布的模乘电路并行或复用结构,使模乘单元输入数据的依赖关系尽量降低以提升其并行度. 其三是从模乘结构入手,使用不同的乘法器结构和数的表示形式,使模乘单元形成流水结构,降低乘法单元的面积,或提升运算电路的频率.

模乘算法根据模数P是否固定,主要分为两种形式. 一种是针对特定模数P,先进行大位宽的乘法运算,然后进行快速约减[12],常结合Karatsuba和Toom-Cook等算法将大位宽操作数拆分成小位宽,并在拆分的过程中消除部分运算量;另一种是针对通用模数P,在计算乘法的过程中进行约减,有蒙哥马利乘法[13](如算法1)和Barrett乘法[14]等,为使乘法器面积减小,也常将大位宽的操作数拆分成小的分组迭代多次完成. 自从1985年Montgomery[13]提出了蒙哥马利模乘算法,后人又提出了FIPS,FIOS,CIOS等[15]. 基-2的蒙哥马利模乘结构消耗的周期数为操作数位宽L,过多的周期数并不适用于高性能ECC处理器;FIOS(FIPS)结构的蒙哥马利算法经过两层的循环,使周期数增加至$(L/w)^2$,其中L为椭圆曲线操作数位宽,w为乘法器字宽. FIOS同样因其消耗较多的周期数而并不适用. 高基的蒙哥马利结构可在L/w个周期内计算出模乘的结果,消耗的周期数量较少,但为了满足NIST和RFC两个协议的需求,则需要过多的硬件资源.

算法1 Montgomery模乘算法

输入: $X,Y,P,R=2^k\geqslant P,k\geqslant\lfloor\log_2P\rfloor+1$.

输出: $XYR^{-1}\pmod P$, 其中, $RR^{-1}=1 \bmod P,\ P'P=-1 \bmod R$.

1. $c=XY$
2. $m=\left((c \bmod R)P'\right) \bmod R$
3. $t=(c+mP)/R$
4. IF $t>P$, THEN $t=t-P$
5. RETURN t

3 算法设计

椭圆曲线密码的运算分为协议层、标量乘法、点运算和模运算四层. 本节对运算的核心即模乘运算和点运算两个层次进行分析和改进. 通过模乘运算单元的优化,实现了更好的通用性;使用两个改进的模乘运算单元进行并行计算,提高了ECC的计算性能. 此外,为了提高两个模乘单元的利用率,将点加和倍点的公式进行了合并优化和时序排布,设计了倍点、点加倍点和蒙哥马利阶梯运算控制器.

ECC处理器的工作流程如图1所示,处理器的输入为常规域数值,由于使用了蒙哥马利模乘算法,因此在计算初始将输入数据转化至蒙哥马利域下. 在标量乘算法中为了避免模逆的运算次数使用了Jacobian坐标系,需在标量乘开始时将坐标转换至雅阁比坐标系下. 默认情况下,只有在计算RFC-7748协议下曲线时使用蒙哥马利阶梯标量乘算法,其余曲线均使用NAF标量乘,但可通过配置切换标量乘算法. 在标量乘计算结束时,将三元Jacobian坐标系转换回二元仿射坐标系,但由于上层的签名验签计算仍需要模乘运算,因此在做完签名验签后,再将数据从蒙哥马利域转换回常规域.

3.1 通用模数的高基蒙哥马利模乘

本节提出了针对多种位宽操作数优化模乘结构. 由于TLS1.3标准中6种位宽的操作数具有最大公约数32,因此本结构基于高基的蒙哥马利算法[16],选取基为$r=2^{32}$,其原因有两个. 第一,对于TLS1.3中所支持的6种宽度(192,224,256,384,448,544)模乘运算均可以整数轮次进行计算(位宽均为32的整数倍),以最高的利用率进行最常用的曲线参数模乘运算. 此外,本算法支持的位宽不仅限于32的倍数,对于非32的整数倍位宽,可通过修改计算轮数,并将不足32倍数位宽的部分补0,支持所有不大于544 bit操作数位宽通用模数P的模乘运算. 第二,本文提出的椭圆曲线处理器面向于ASIC设计,在55 nm工艺库下,272 bit × 32 bit乘法器既能保证较少的模乘周期数量,又使控制电路和运算单元的路径延迟保持均衡,获得较高的时钟频率. 从面积角度讲,272 bit × 32 bit的乘法器面积较大,但从计算单元面积与控制存储单元的面积比来看,可使控制存储

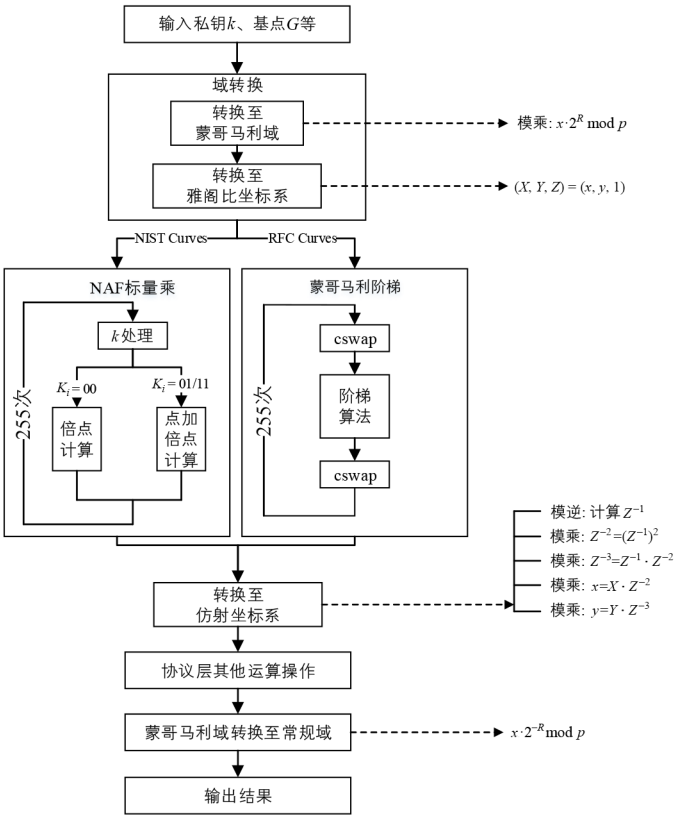


图1　ECC处理器工作流程

单元所占用的面积比例减小,从而获得更高的面积时间积($A\cdot T$).

使用高基蒙哥马利乘法(选取基为2^{32}),对于最常用的256 bit位宽的模乘运算,仅需8个时钟周期即可完成. 然而,若要对544 bit的模乘进行支持,则需要2个544 bit × 32 bit乘法器,以及2个32 bit × 32 bit乘法器,导致电路的面积过于庞大. 然而在椭圆曲线密码的应用中,256 bit位宽的椭圆曲线应用最广,因此优先满足256 bit位宽的模乘运算速度,并在对大于256 bit的椭圆曲线参数兼容的前提下,尽量使面积最小. 因此,本文提出的模乘运算如算法2所示. 其中,L为操作数位宽,$L=\lfloor \log_2 P \rfloor + 1, 32 \leqslant L \leqslant 544$; $m=\lceil L/32 \rceil$; x_i, y_i, z_i, q_i均为32 bit;$t_1, t_{10}, t_{11}, t_2, t_{20}, t_{21}$均为中间变量.

如算法2所示,步骤5~7和步骤8~13分别针对模乘操作数位宽在272 bit及以下,和273~544 bit. 对于最常用的272 bit及以下的操作数位宽乘法,可在单周期内完成以保证较少的计算周期数量,获得较快的运算性能;对于使用频率较小的272 bit以上位宽的乘法,使用两个周期完成一次乘法操作,通过复用半字宽的乘法器的方式达到节省面积的目的. 例如,对于256r1曲线而言,步骤6和步骤7分别使用272 bit × 32 bit和256 bit × 32 bit乘法器在同一个时钟周期内完成;对于521r1曲线来说,将步骤9和步骤11同时在第一个时钟周期内完成,用于计算两个乘法的低位部分,将步骤10和12在第二个时钟周期内完成,用于计算乘法的步骤高位部分,并在此步骤中将低位与高位部分相加.

另外,在式中有X_iY_0和X_iY两个乘法,若将X_iY_0和X_iY用同一个乘法器实现,虽然会减小电路面积,但q_i的计算需要等待544 bit × 32 bit的乘法结束后才能继续进行,关键路径过长会大幅降低芯片的工作频率;若使用两个不同的乘法器实现,则存在共同的计算部分造成面积浪费问题. 本文将大位宽乘法器进行拆分:

$$C=X_iY=X_i(Y_H2^w+Y_L)=X_iY_H2^w+X_iY_L.$$

其中,Y_H和Y_L为操作数Y的高位和低位部分;本设计中乘法器字宽w为32.

因此本文提出的结构中,将X_iY_0和X_iY中共同的计算部分分离出来,先计算较快的X_iY_0,并将中间用于后续的计算过程中;随后再进行较大的高位部分计算,虽然这部分的运算较慢,但可与步骤6(273 bit以下)或步骤9、步骤10(272 bit以上)的两个大位宽乘法器同时进行,因此并没有额外增加的延迟.

模乘单元的电路结构如图2所示,在算法2中步骤3、步骤4各包含一个32 bit × 32 bit乘法器,如图2中蓝色乘法单元. 在步骤5~13中,共包含两个大位宽乘法器,分别使用深红色和浅红色表示. 其中,深红色乘法单元用于计算步骤6、步骤9、步骤10,用于计算q_iP,其

算法2 改进的高基Montgomery模乘算法

输入：$X=\sum_{i=0}^{m-1}x_i\cdot(2^{32})^i;\ Y=\sum_{i=0}^{m-1}y_i\cdot(2^{32})^i;$

其中，$0<X,Y<P,\ P<R=2^{32\cdot m},\ \gcd\left(P,2^{32}\right)=1$.

输出：$Z=XYR^{-1}\bmod P$.

1. $Z_0=0$;
2. FOR $i=0$ to $\left\lceil\frac{L}{32}\right\rceil-1$
3. $c=X_iY_0$
4. $q_i=\left(\left(\left(Z_i+c\right)\bmod 2^{32}\right)P'\right)\bmod 2^{32}$
5. IF $L\leqslant 272$
6. $t_1=q_iP$
7. $t_2=\left(X_i\left(Y/2^{32}\right)\right)2^{32}+c$
8. ELSE
9. $t_{10}=q_i(P\bmod 2^{L/2})$
10. $t_{11}=\left(q_i\left(P/2^{L/2}\right)\right)2^{L/2}+t_{10}$
11. $t_{20}=X_i\left(\left(Y/2^{32}\right)\bmod 2^{(L-32)/2}\right)$
12. $t_{21}=\left(X_i\left(Y/2^{(L/2+16)}\right)2^{(L-32)/2}+t_{20}\right)2^{32}+c$
13. $t_1=t_{11},\ t_2=t_{21}$
14. ENDIF
15. $Z_{i+1}=(Z_i+t_1+t_2)/2^{32}$
16. END FOR
17. IF $Z_{i+1}\geqslant$ P THEN $Z_{i+1}=Z_{i+1}-$P; ENDIF
18. RETURN Z_{i+1}

计算位宽为544 bit × 32 bit；浅红色的乘法单元用于计算步骤7、步骤11、步骤12，用于计算XY. 由于与蓝色X_iY_0的乘法器有共用部分，此乘法单元计算位宽为512 bit × 32 bit.

红色的两个特殊乘法单元结构如图2所示，操作数B在位宽大于$L/2$时，由sel信号进行选择，决定着输入为B的高位部分或是低位部分. 当操作数B在位宽小于等于$L/2$时，sel的值恒为0，此时乘法的计算在单周期完成，乘法单元作为$L/2\times 32$ bit乘法器使用. 当操作数B在位宽大于$L/2$时，在计算的第一个时钟周期，sel为0，输入为B的低$L/2$ bit，并由$L/2\times 32$ bit乘法器计算完成后，将结果暂存于寄存器中；在第二个时钟周期，sel由控制器置为1，此时输入为B的高$L/2$ bit，乘法器的计算结果将在左移$L/2$ bit后，上一周期的结果相加，并在第二个周期输出最终乘法结果.

3.2 基于双模乘结构的点运算时序设计

使用单个模乘单元做点运算，其硬件利用率最高，但不满足高性能的处理需求；当使用多个模乘单元时，运算所消耗周期数下降，但对模乘单元的硬件利用率也会相应的下降. 本节针对双模乘运算单元的电路结构，在雅阁比坐标系下的NIST和RFC两种不同曲线的点运算公式计算过程中的数据依赖关系进行分析与优化，以最大化模乘单元利用率，成倍地减少标量乘法的周期数量，提高处理器性能和面积时间积(AT).

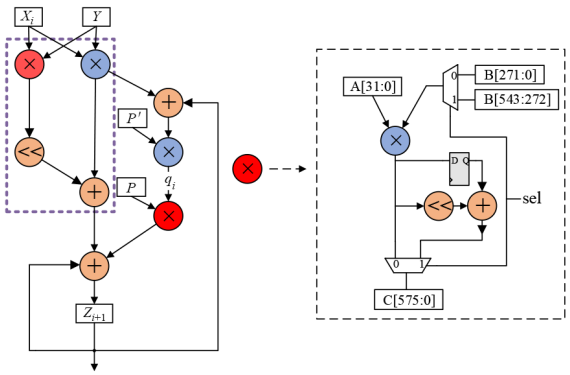


图2 模乘单元结构图

本文对3种情况下的点加倍点公式分别进行了时序排布：Weierstrass曲线下的倍点、点加倍点，以及Montgomery曲线下的蒙哥马利阶梯.

情况一为NIST定义椭圆曲线下的倍点运算(私钥$k_i=0$的情况). 其中，t_1~t_6表示521 bit通用寄存器堆中的6个. 图3所示为倍点计算的时序排布方法. 设计中使用的两个模乘单元分别用浅蓝色和深蓝色表示，两个模加减单元分别用浅黄色和深黄色表示. 在此时序排布方法中，共有4级模乘，各级之间复用2个模乘单元和2个模加减单元. 由于先后的数据依赖关系，倍点操作共需$4N_{\text{mult}}+5$个周期完成. 其中N_{mult}为一次模乘运算(图中Stage)的周期数. 在最常用的secp256r1曲线下，N_{mult}为8，此时模乘单元的利用率为86.5%；而在secp521r1曲线下，2个模乘单元的利用率高至96.45%.

情况二为NIST椭圆曲线下的点加倍点同时运算(私钥$k_i=\pm1$的情况). 在点加倍点的运算中，用到11个521 bit的通用寄存器. 其时序排布情况如图4所示. 由于点加和倍点中的运算被合在一起进行时序排布，因此所有的模加减操作均可与两个模乘单元并行，从而大大提高了模乘单元的利用率. 在点加和倍点的操作中，共有24个模乘运算，但点加和倍点公式中有一个模乘运算是一样的，因此共需进行23次模乘运算. 此时，secp256r1曲线下的模乘单元利用率为95.45%，其中第一个模乘单元的利用率为100%.

情况三为RFC-7748标准的椭圆曲线下的点运算. 为了考虑抗功耗攻击性能，使用了统一的点加倍点公式. 按照协议推荐，运算在仿射坐标系下进行. 第一次模乘运算的输入数据依赖第一次模加减的结果，使模乘单元出现空闲周期. 因此在本文提出的点加倍点控制器中，将步骤0中的模加减操作与cswap同时进行，使两个模乘单元的利用率均提升至100%. 整个阶梯运算共消耗5次模乘运算的时间，以256 bit的椭圆曲线为例，进行一次点加倍点的计算共需要40个时钟周期.

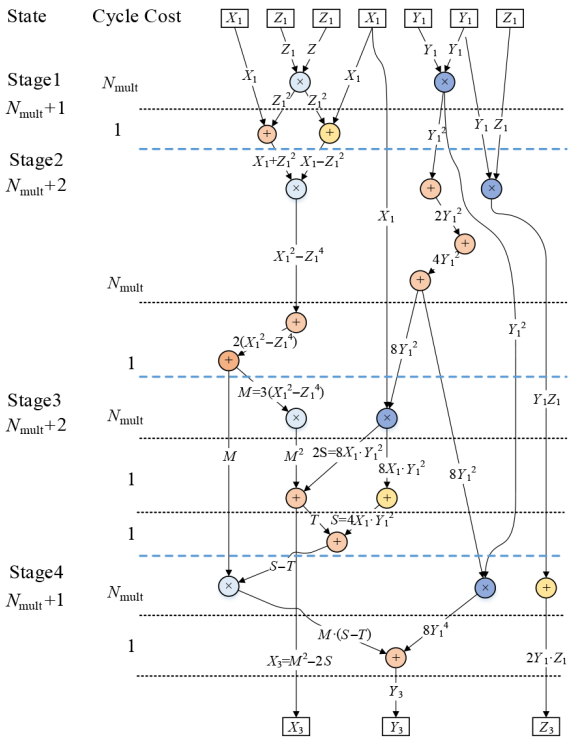


图3　双模乘结构下的倍点运算时序排布图

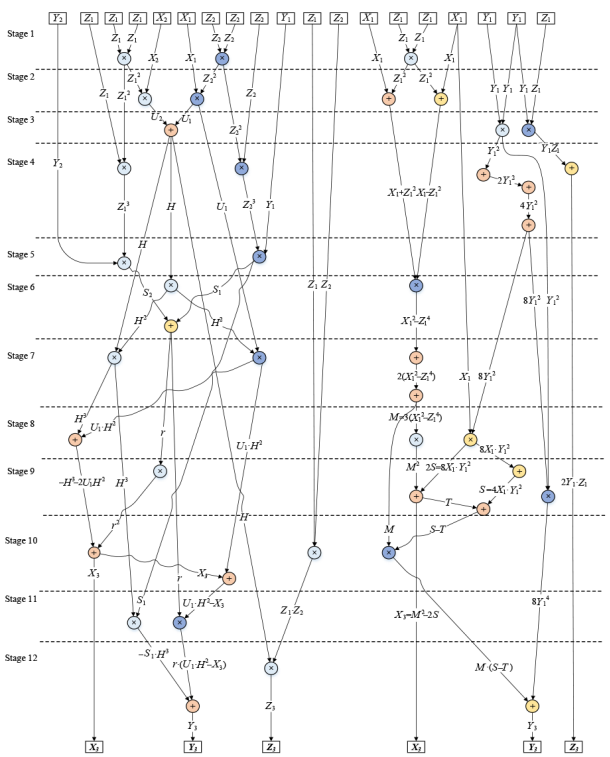


图4　双模乘结构下的点加倍点运算时序排布图

运算中使用到通用寄存器堆中的7个. 其时序排布情况如图5所示.

3.3　模逆运算

模逆运算所使用的方法较多,如使用二进制右移

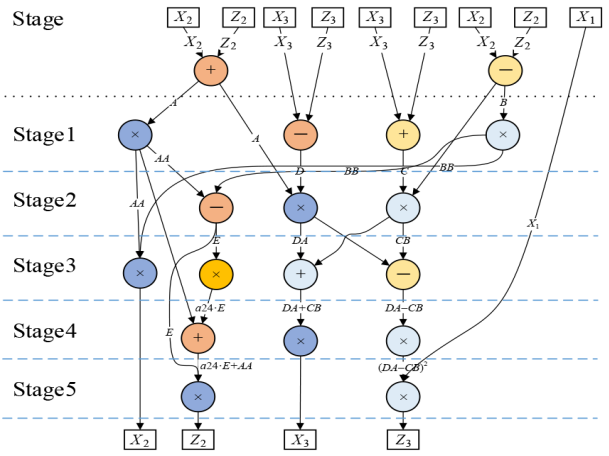


图5　双模乘结构下的蒙哥马利阶梯运算时序排布图

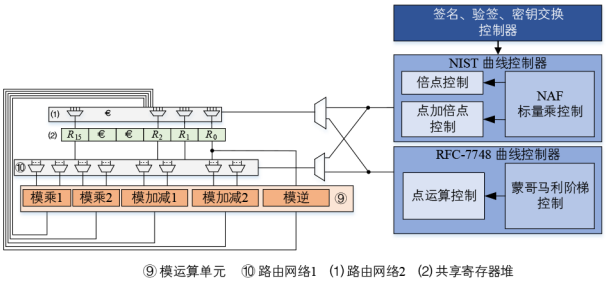
算法[17],该算法理论上需要至多512个周期即可完成运算. 本文使用基于蒙哥马利求逆算法的改进版本Kaliski算法,所以将模逆模块单独引出,设立标志位以供标量乘内部使用或外部调用.

4　硬件实现和比较

4.1　ECC处理器的实现与验证

图6所示为ECC处理器的整体结构. 设计选用了双标量乘控制器结构:一个使用NAF标量乘算法保证NIST曲线下的运算速度,另一个使用蒙哥马利阶梯标量乘算法保证RFC曲线下的抗功耗攻击能力,使用的标量乘算法可通过配置切换. 转换坐标系选取了Jacobian坐标系(Weierstrass曲线)和仿射坐标系(Montgomery曲线). 为了获得最低的计算周期数量,设计使用了双模乘单元(如3.1节所述)和双模加减单元的结构. 对其点加倍点公式中模运算的时序如3.2节所述. 此外,根据点运算的时序排布情况,两个模加减单元既可在单周期内完成两组不同的并行模加减的操作,也可以通过在单周期内完成串联的两次模加减操作,如图7所示. 模逆使用了基于蒙哥马利求逆算法的改进版本Kaliski算法.

本结构使用了16个521 bit的通用存储器,用于点加、倍点和标量乘中间结果的存储. 模运算单元的输入由路由网络2进行多路选择,而模运算单元的输出,通



⑨ 模运算单元　⑩ 路由网络1　(1) 路由网络2　(2) 共享寄存器堆

图6　ECC处理器整体结构

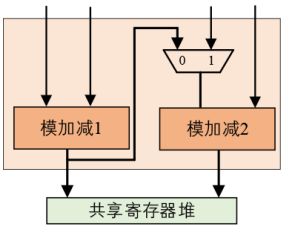


图7 模加减运算单元互联结构

过路由网络1分配至16个通用寄存器中. 路由网络1和路由网络2的选择信号由控制单元生成.

在不同的曲线参数和工作模式下,各部分计算所需的周期数量如表1所示.

表1 不同曲线参数和工作模式下的运算周期数量

曲线	模乘	倍点	点加倍点	模逆	NAF标量乘	蒙哥马利阶梯标量乘
secp192r1	6	29	72	386	8 345	-
secp224r1	7	33	84	450	11 229	-
secp256r1	8	37	96	514	14 539	-
Curve25519	8	-	41	512	-	10 471
secp384r1	24	101	288	770	62 817	-
Curve448	28	-	141	898	-	63 224
P-521r1	34	141	408	1 044	119 967	-

4.2 性能比较

本文选取了5篇支持521 bit位宽椭圆曲线参数的论文设计进行算法上的比较,如表2所示. 在6篇文献中,本设计在支持通用模数且同时支持Montgomery和Weierstrass两种曲线的前提下,标量乘所消耗的周期数量、单次运算时间均少于其他文献.

表2 模乘算法及标量乘周期数量比较

文献	是否通用模数	模乘算法	并行设计	标量乘消耗的周期数量	
				曲线	周期数/(k·Cycle)
本文	是	改进的高基蒙哥马利	2个模乘单元并行	secp256r1	14.539
				secp521r1	119.967
文献[6]	是	基-4蒙哥马利	4个模乘单元并行	secp256r1	144.500
				secp521r1	570.300
文献[4]	仅NIST	特定的快速模约减	无	secp256r1	366.000
				secp521r1	2 394.000
文献[5]	仅NIST	特定的快速模约减	无	secp256r1	991.701
				secp521r1	7 038.040
文献[7]	仅NIST	特定的快速模约减	模乘和约减并行	secp256r1	15.360
				secp521r1	93.780
文献[8]	仅NIST	特定的快速模约减	模加减和模乘并行	secp256r1	15.714
				secp521r1	74.787

在以上文献中,仅有[6]支持通用模数. 使用了4个基-4的模乘单元,但较多的运算单元造成了较大的面积,而其利用率却不能达到较高的水平. 基-4的模乘算法由于迭代次数较多,使整个标量乘法的周期数量变多. 同时,RSD和CSA结构组成的乘法单元并没有使模乘的关键路径延迟降低很多.

大部分兼容NIST曲线的研究,都仅仅支持了协议中所使用到的少数几个特定模数P,如文献[4,5,7,8]和文献[18]. 由于模数固定,因此取余操作的计算十分简单,其全部使用了普通乘法和快速模约减的算法结构. 这种结构虽然可获得较快的速度和较小的面积,但应用范围十分有限. 其中,文献[4]中的模乘单元使用了8个32 bit乘法器(由18 bit × 18 bit DSP48E构成),为了降低控制单元的面积,所设计的架构使用了指令执行的方式,并将模运算单元作为运算处理单元通过总线结构与控制器相连. 这种结构的取指、回写等操作消耗的周期数过于多(如模乘则需78个周期),造成了较低的面积时间积.

文献[5]同样支持NIST的5条曲线,由于模乘单元使用了单个乘法器迭代运行的结构,设计的面积非常小,而各单元的利用率也达到了很高的水平. 得益于Xilinx Virtex5的DSP运算单元,使其主频可高达182 MHz,十分适合对面积要求高的场景,而极少的乘法器资源使标量乘法的周期数量较大,对吞吐量较高的场景则并不适用,同样的,设计并不支持通用的模数.

文献[7]为了顾及5条曲线的位宽,将乘法单元设计为134 bit,借助Karatsuba算法,使256 bit × 256 bit的乘法运算,由4次半位宽乘法(128 bit乘法,实际使用134 bit实现)缩减到3次. 设计中还借助了Lazy Reduction算法和Naïve adder,同时更改了模约减公式,省去了传统快速模约减算法中的最后一步有限范围内的求模运算. 此外,论文还设计了乘法器的三级流水和模约减的二级流水结构,非常有效地提升了设计的并行度,

但对384 bit和521 bit的优化有限. 文献[8]在文献[7]的基础上,使用了Toom-Cook乘法,进一步将大位宽的乘法(384 bit以上)次数减少. 在硬件设计上,将Toom-Cook乘法所使用的加法阵列与模约减部分共用,节省了芯片面积,并使用了四级流水的乘法器和四级流水的加法矩阵来完成模乘运算,使其并行度提高. 然而,文献[7,8]都使用了较小的运算单元,使控制部分的面积占比较大,有效运算的面积占比较低,利用率下降. 同时,两篇文献为FPGA而优化,对于ASIC实现的电路而言可以使用更大的模运算电路以提高有效运算部分的电路面积占比. 此外,两篇文献仅支持特定的模数,本文则针对高性能的通用模数应用设计了适用于ASIC的ECC处理器,FPGA平台并不能很好地满足性能要求.

ECC处理器使用ASIC设计方法,采用TSMC 55 nm工艺库实现,由于设计目标面向高计算性能服务器,乘法器的设计并不适用于FPGA平台. 设计工具使用了VCS做逻辑验证. 使用Synopsys Design Compiler做逻辑综合,并以此获得电路的工作频率和面积. 55 nm工艺库下电路工作频率最高可到454 MHz,此主频下的面积使用二输入等效与非门(GE)数量评估,为851k. 按表1中平均消耗周期计算,对于secp256r1位宽的椭圆曲线,计算标量乘需32.02 μs,每秒可计算3.123万次. 对于Curve25519曲线,计算一次标量乘需23.06 μs,每秒可计算4.335万次,即为密钥交换的计算速度. 为了与其他文献进行对比,本文同时使用了TSMC 130 nm和SMIC 180 nm工艺库进行逻辑综合.

目前,专门针对TLS1.3的密码加速芯片尚未有较完整的研究,在进行比较的ECC处理器中,使用ASIC实现的文献[8,18~24]如表3所示. 可见,在支持521 bit通用模数P的设计中,本文的运算速度最快,AT最小,且支持多类椭圆曲线,具备更佳的灵活性.

表3　标量乘性能对比

文献	工艺/nm	主频/MHz	门数/(k·GE)	通用P	曲线	周期数量/(k·Cycle)	运算速度/(k·OPs)	单次时间/μs	AT
本文	55.000	454.000	851.000	是	secp192	8.345	54.40	18.38	15.64
					secp224	11.229	40.43	24.73	21.04
					secp256	14.539	31.23	32.02	27.24
					secp384	62.817	7.22	138.36	117.74
					secp521	119.967	3.78	264.24	224.87
					X25519	10.471	43.35	23.06	19.62
					X448	63.224	7.18	139.26	118.51
	130.000	295.000	755.550		secp256	14.539	20.29	49.28	37.23
					secp521	119.967	2.46	406.67	307.26
	180.000	237.000	832.580		secp256	14.539	16.30	61.35	51.08
					secp521	119.967	1.98	506.19	421.44
文献[19]	65.000	549.450	447.000	是	secp256	397.300	1.38	730.00	326.31
文献[20]	90.000	217.000	313.000	是	secp256	164.920	1.71	760.00	237.88
文献[21]	90.000	185.000	540.000	是	secp256	22.200	8.30	120.00	64.80
文献[22]	90.000	250.000	157.300	是	secp192	56.250	4.40	225.00	35.39
文献[23]	130.000	150.000	59.140	是	secp256	475.000	0.32	3 160.00	186.90
文献[24]	130.000	150.000	57.050	是	secp256	610.000	0.25	4 070.00	232.00
文献[8]	180.000	360.000	466.000	仅NIST	secp256	15.834	22.73	43.98	20.49
					secp521	74.802	4.81	207.78	96.82
文献[18]	180.000	200.000	65.400	仅NIST	secp256	148.000	1.35	740.00	48.40

在上述设计中只有文献[8]支持了521 bit位宽的运算,但仅支持特定的模数;文献[18~21]仅支持256 bit位宽并支持通用的模数. 与文献[19~21]相比较,本文的周期数量和单次标量乘耗时均有较大的提升. 原因是其模乘单元使用了较多周期的迭代,使标量乘运算的周期数量明显增多,虽然运算单元的面积有所下降,但控制电路并未减小,导致了AT较大. 而文献[8]仅支持特定的模数,在模乘时可使用快速模约减算法,计算的复杂度上远低于本文通用模数的模乘,因此其周期数量与本文相似;另外本文同时支持NIST和RFC-7748两类椭圆曲线,面积也会有更多消耗,因此文献[8]获得的AT乘积也略好于本文. 文献[23]由于仅支持了192 bit运算,因此有较小的面积. 文献[23,24]都使用了Interleaved模乘算法,并在计算模乘的单元中仅使用了加法器实现,这两种方案都可以有效地降低芯片面积. 但文献[6]运算消耗周期过多,控制电路占

比大导致了电路单位面积的运算效率低. 文献[18]同样使用了相似的256 bit × 32 bit乘法器,把快速模约减合并在了每一次乘法的迭代中,节省了芯片面积,但同样仅支持特定模数的运算.

5 结束语

本文首先通过对高基蒙哥马利算法的改进,提出了一种支持多位宽的模乘运算单元,此模乘单元可通过多轮迭代实现521 bit以下任意位宽模数运算. 其次,本文基于双模乘单元设计了并行的点运算时序排布,在该排布下的两个模乘单元可在NIST椭圆曲线下实现95.4%和86.5%的利用率,在RFC-7748定义的曲线下实现100%的利用率. 最后本文设计了基于蒙哥马利阶梯和二进制NAF的双椭圆曲线标量乘法的硬件结构. 在该结构下使用55 nm CMOS工艺实现的电路可工作在最高454 MHz,等效逻辑门851k,运算速度高达31 230 times/s.

参考文献

[1] RESCORLA E. The Transport Layer Security (TLS) Protocol Version 1.3: RFC 8446[S]. Fremont: IETF Trust, 2018.

[2] IEEE. IEEE Standard Specifications for Public Key Cryptography: IEEE 1363-2000[S]. Piscataway: IEEE, 2000.

[3] LANGLEY A. Elliptic Curves for Security: RFC 7748[S]. Fremont: IETF Trust, 2016.

[4] ANANYI K, ALRIMEIH H, RAKHMATOV D. Flexible hardware processor for elliptic curve cryptography over NIST prime fields[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009, 17(8): 1099-1112.

[5] LOI K C C, KO S B. Scalable elliptic curve cryptosystem FPGA processor for NIST prime curves[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2015, 23(11): 2753-2756.

[6] SHAH Y A, JAVEED K, AZMAT S, et al. A high-speed RSD-based flexible ECC processor for arbitrary curves over general prime field[J]. International Journal of Circuit Theory and Applications, 2018, 46(10): 1858-1878.

[7] DING J N, LI S G. A reconfigurable high-speed ECC processor over NIST primes[C]//2017 IEEE Trustcom/BigDataSE/ICESS. Sydney: IEEE, 2017: 1064-1069.

[8] DING J N, LI S G, GU Z. High-speed ECC processor over NIST prime fields applied with toom-cook multiplication [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2019, 66(3): 1003-1016.

[9] BERNSTEIN D J. Curve25519: New Diffie-Hellman speed records[M]//Public Key Cryptography - PKC 2006. Berlin: Springer, 2006: 207-228.

[10] COHEN H, MIYAJI A, ONO T. Efficient elliptic curve exponentiation using mixed coordinates[M]//Lecture Notes in Computer Science. Berlin: Springer, 1998: 51-65.

[11] 仲先海. 并行可配置ECC协处理器关键技术研究[D]. 郑州: 解放军信息工程大学, 2008.
ZHONG X H. Research of Key Techniques on a Parallel and Reconfigurable ECC Coprocessor[D]. Zhengzhou: PLA Information Engineering University, 2008. (in Chinese)

[12] 刘哲, 王伊蕾, 徐秋亮. 最优素数域的优化蒙哥马利算法: 设计、分析与实现[J]. 密码学报, 2014, 1(2): 167-179.
LIU Z, WANG Y L, XU Q L. Optimized Montgomery algorithms for optimal prime fields: Design, analysis and implementation[J]. Journal of Cryptologic Research, 2014, 1(2): 167-179. (in Chinese)

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

[14] BARRETT P. Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor[M]//Advances in Cryptology - CRYPTO'86. Berlin: Springer, 2007: 311-323.

[15] KAYA KOC C, ACAR T, KALISKI B S. Analyzing and comparing Montgomery multiplication algorithms[J]. IEEE Micro, 1996, 16(3): 26-33.

[16] ELDRIDGE S E, WALTER C D. Hardware implementation of Montgomery's modular multiplication algorithm [J]. IEEE Transactions on Computers, 1993, 42(6): 693-699.

[17] HANKERSON D, MENEZES A J, VANSTONE S. Guide to Elliptic Curve Cryptography[M]. New York: Springer, 2004.

[18] CHOI P, LEE M K, KIM J H, et al. Low-complexity elliptic curve cryptography processor based on configurable partial modular reduction over NIST prime fields[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2018, 65(11): 1703-1707.

[19] HOSSAIN M S, KONG Y N, SAEEDI E, et al. High-performance elliptic curve cryptography processor over NIST prime fields[J]. IET Computers & Digital Techniques, 2017, 11(1): 33-42.

[20] LEE J W, CHUNG S C, CHANG H C, et al. Efficient power-analysis-resistant dual-field elliptic curve crypto-

graphic processor using heterogeneous dual-processing-element architecture[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2014, 22(1): 49-61.

[21] CHUNG S C, LEE J W, CHANG H C, et al. A high-performance elliptic curve cryptographic processor over GF(p) with SPA resistance[C]//2012 IEEE International Symposium on Circuits and Systems (ISCAS). Seoul: IEEE, 2012: 1456-1459.

[22] 黎明, 吴丹, 戴葵, 等. 高性能可扩展公钥密码协处理器研究与设计[J]. 电子学报, 2011, 39(3): 665-670.
LI M, WU D, DAI K, et al. Research and design of a high-performance scalable public-key cipher coprocessor [J]. Acta Electronica Sinica, 2011, 39(3): 665-670. (in Chinese)

[23] CUI C, ZHAO Y, XIAO Y, et al. A hardware-efficient elliptic curve cryptographic architecture over GF(p) [J]. Mathematical Problems in Engineering, 2021, 2021: 1-7.

[24] HU X H, ZHENG X, ZHANG S S, et al. A low hardware consumption elliptic curve cryptographic architecture over GF(p) in embedded application[J]. Electronics, 2018, 7(7): 104.

作者简介

刘志伟　男,1987年出生,哈尔滨人. 现为哈尔滨理工大学测控技术与通信工程学院博士研究生. 主要研究方向为可重构计算、高速密码算法、并行加密技术、密码芯片的安全设计等.
E-mail: zwliu@hrbust.edu.cn

刘雷波　男,1976年生,新疆克拉玛依人. 清华大学长聘教授,博士生导师,国家级一流本科课程负责人. 主要研究方向为软件定义芯片、硬件安全和密码芯片、VLSI数字信号处理等.
E-mail: liulb@tsinghua.edu.cn

黄　海(通讯作者)　男,1982年生,内蒙古巴彦淖尔人. 博士,教授,博士生导师,CCF高级会员. 主要研究方向为信息安全、可重构计算、集成电路设计.
E-mail: ic@hrbust.edu.cn