

GF(2^m)域上的低功耗可配置 ECC 点乘算法 ASIC 设计实现

车光宁^{1,2,3}, 张钊锋¹

(1 中国科学院 上海高等研究院, 上海 201210; 2 中国科学院大学, 北京 100049;

3 上海科技大学 信息科学与技术学院, 上海 201210)

摘 要: 针对射频识别(RFID)和无线传感网(WSN)等领域的高安全、低功耗、轻量化和可扩展等应用需求,设计一种 GF(2^m)域上实现的椭圆曲线标量乘法电路. 通过对椭圆曲线标量乘法整体架构实现进行逐层分级优化改进,尤其对标量乘核心模块——模乘和模逆进行了低功耗设计改进. 经过 Xilinx FPGA 工具仿真和 Synopsys Design Compiler 工具验证,该椭圆曲线标量乘法架构灵活可拓展,GF(2¹⁶³)域上一次点乘运算仅需要 138 k 个时钟周期并且在 TSMC 0.13 μm 工艺下等效面积仅为 11.9 k,相比较同类设计,面积和执行速度都有着效果显著的优势,可以胜任像 RFID 以及 WSN 一样资源受限的应用场合使用.

关键词: 椭圆曲线点乘算法;有限域算术;低功耗;FPGA;ASIC

中图分类号: TN47

文献标识码: A

文章编号: 1000—7180(2018)01—0015—06

Low Power Configurable ASIC Based Elliptic Curve Scalar Multiplication Over GF(2^m)

CHE Guang-ning^{1,2,3}, ZHANG Zhao-feng¹

(1 Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China;

2 University of Chinese Academy of Science, Beijing 100049, China;

3 School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China)

Abstract: Aiming at the design requirements of high security, low power consumption, light implementation and configurable requirements for radio frequency identification (RFID) and wireless sensor networks (WSN) applications, an elliptic curve scalar multiplication circuit is designed. Through the overall optimization of the whole architecture of the elliptic curve scalar algorithm layer by layer, especially for the lower power design of the core scalar module: modular multiplication and inversion. Through the Xilinx FPGA tool simulation and the comprehensive verification of Design Compiler tools of Synopsys, the scalar multiplication architecture is proved flexible and expandable, one elliptic curve scalar point multiplication over GF(2¹⁶³) is only 138 k clock cycles and the equivalent gate area with TSMC 0.13 μm is only 11.9 k, compared with similar designs, the area and the implementation of the speed has a significant advantage, it can be used for RFID, WSN and other resource-constrained applications.

Key words: elliptic curve point multiplication; finite field arithmetic; low power; FPGA; ASIC

1 引言

随着 RFID 和 WSN 等技术的普及应用,越来越多的无线终端设备接入到网络环境中,低功耗、高安全性的认证和加密算法实现不可避免地成为相关

领域内关注的热点. 公钥加密算法,尤其是椭圆曲线加密算法因为其高安全性、密钥短、存储和带宽要求低等优点,在资源受限的应用领域内有着广阔的应用前景. 在椭圆曲线二进制域 GF(2^m)^[1]上设计实现的椭圆曲线硬件加密系统,因为其实现高效灵活、

功耗低等优点,近年来被应用于设计各种轻量化的硬件加密方案.针对硬件实现消耗的资源 and 实现效率方面的综合权衡,本文针对 ECC 点乘算法进行分层优化设计,逐级优化有限域上的点乘算法、模运算算法,同时权衡功耗、面积、速度和灵活性,设计实现了低功耗可配置的的椭圆曲线点乘算法 ASIC 电路.

2 椭圆曲线加密处理器设计

2.1 ECC 处理器总体结构设计

ECC 标量乘法在整个椭圆曲线加密系统中的作用如图 1 所示^[1],最高层是椭圆曲线加密算法协议层,通过调用点乘算法实现椭圆曲线密钥交换算法(ECDH)和椭圆曲线数字签名算法(ECDSA).第二层是椭圆曲线的点乘运算,第三层是支撑点乘运算的椭圆曲线点加和倍点运算,最底层是有限域的算术模运算,也是顶层协议和算法的具体实现执行层.

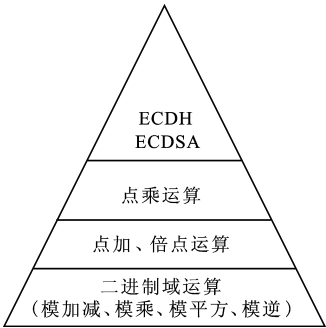


图 1 ECC 点乘算法运算层次图

本文所设计的低功耗 ECC 处理器电路主要由算术逻辑运算模块(ALU)、椭圆曲线处理器控制模块(ECCM)和寄存器存储电路(REG)三部分构成,如图 2 所示. ALU 执行的是图 1 中第三层和第四层的有限域模运算;ECCM 执行顶层点乘算法实现,负责整体计算资源的分配. REG 部分寄存椭圆曲线参数及计算过程中的中间值,配合 ECCM 的数据调用. 本文以下部分将自顶向下逐级优化实现 ECC 点乘算法的低功耗设计.

2.2 点乘算法设计优化

椭圆曲线上的一个点 P 被一个整数 k 相乘称为标量乘或者点乘,它被定义为 k 个 P 的相加,即 $kP = P + P + \dots + P$. 本文所涉及的二进制域 $GF(2^m)$ 上通常有两种通用的算法去实现 ECC 点乘,一种是二进制算法(如算法 1 所示);另一种广泛使用的标量算法则是 Montgomery Ladder 算法^[2],相比较二进制算法每次点加和倍点操作的不同,该算法每次循环均执行一次点加和一次倍点操作,可以有

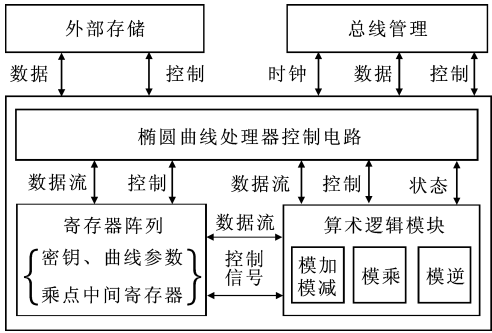


图 2 ECC 处理器的架构图

效的抵抗简单功耗分析攻击(SPA),更具有安全性. 实际应用中基于仿射坐标系的 Montgomery Ladder 算法由于每轮循环中都要执行至少 1 次的模逆操作,计算复杂度高,并不适用于资源受限的应用场合. 为此本文采用射影坐标系下实现的 Montgomery Ladder 算法,这样每次点乘只需要 1 次模逆操作完成最后的坐标值转换即可,同时计算循环中也只需要基点的 x 坐标值参与运算,可以大幅度的减少计算复杂度和降低计算资源消耗.

算法 1 二进制算法

(a)from left to right

input: k and P $k = (k_{m-1}, \dots, k_0), k_i \in \{0, 1\}$

output: $Q = kP$

1. set $Q \leftarrow 0$
2. for i from $m-1$ to 0 do
3. $Q \leftarrow 2Q$
4. if $k_i = 1$ then $Q \leftarrow Q + P$
5. return Q

(b)from right to left

input: k and P $k = (k_{m-1}, \dots, k_0), k_i \in \{0, 1\}$

output: $Q = kP$

1. set $Q \leftarrow 0$
2. for i from 0 to $m-1$ do
3. if $k_i = 1$ then $Q \leftarrow Q + P$
4. $P \leftarrow 2P$
5. return Q

算法 2 射影坐标系下的 Montgomery Ladder 算法

input: $k, PP = (x_P, y_P) \in GF(2^m)$

$k = (1, k_{m-2}, \dots, k_i, \dots, k_0), k_i \in \{0, 1\}$

output: $Q = kP = (x_Q, y_Q)$

- 1: if $k = 0$ or $x_P = 0$ then
return $Q = (0, 0)$ and stop
- 2: set $X_A \leftarrow x_P, Z_A \leftarrow 1, X_B \leftarrow x_P^4 + b, Z_B \leftarrow x_P^2$
- 3: for $i = m-2$ down to 0 do
if $k_i = 1$

```

R ← ZA, ZB ← (XA ZB + XB ZA)2
XA ← xP ZA + XA XB R ZB
R ← XB, XB ← XB4 + b ZB4, ZB ← R2 ZB2
else
R ← ZB, ZA ← (XA ZB + XB ZA)2
XB ← xP ZB + XA XB R ZA
R ← XA, XA ← XA4 + b ZA4, ZA ← R2 ZA2
4: xQ ←  $\frac{X_A}{Z_A}$ 
5: yQ = yP +  $\frac{(x_P Z_A + X_A) \times T}{x_P Z_B Z_A^2}$ 
T = (XA XB + xP (XA ZB + XB ZA) + yP ZA ZB)
6: return Q(xQ, yQ)

```

本文同时将点加和倍点的有限域操作进行进一步分析,如图3所示^[3],可以看到每轮操作都要进行6次模乘、5次模平方和3次模加操作。考虑到模加和模平方在设计中可以在1个时钟周期内实现完成,本文合理规划点加部分的状态机执行序列,结合点加最长计算路径的分析,在进行模乘操作时交叉进行模平方和模加的运算,保证在使用1个串行乘法器的资源消耗下,完成每一轮计算所要求的全部操作。针对上述点加和倍点的运算执行,本文仅仅采用了6个寄存器变量(X_A, Z_A, X_B, Z_B, R, T)就能完成点乘算法所有数据输入及中间结果寄存转移,其中X_A, Z_A, X_B, Z_B这4个寄存器用来初始化点乘算法的坐标初值并完成仿射坐标系到射影坐标系的转化, R寄存器用来暂存点乘算法中的中间值, T寄存器用来寄存椭圆曲线参数a、b值和相关的密钥k值。

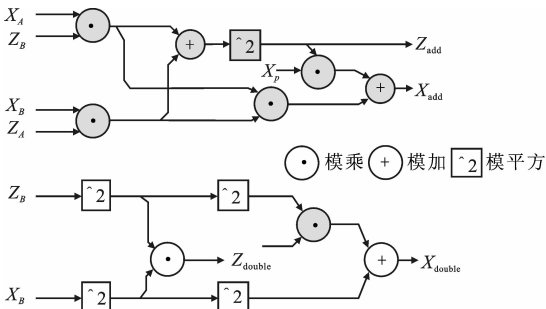


图3 点加(a)和倍点(b)的计算分解

2.3 有限域模运算单元设计优化

2.3.1 模加单元的设计

对于二进制域上的任意两个采用多项基表示的元素:

$$A = (a_{m-1}, a_{m-2}, \dots, a_0) \in \text{GF}(2^m)$$

$$B = (b_{m-1}, b_{m-2}, \dots, b_0) \in \text{GF}(2^m)$$

模加运算的实现可以简单地通过将二进制表示

的元素A和B的对应比特位进行异或求得:

$$C = A + B = \sum_{i=0}^{m-1} a_i \oplus b_i \in \text{GF}(2^m)$$

因此模加单元仅仅需要使用m位的异或门在1个时钟周期内就可以完成。

2.3.2 模平方单元的设计

对于二进制域GF(2¹⁶³)上的一个多项式基表示的元素:

$$A(x) = \sum_{i=0}^{m-1} a_i x^i, a_i \in \{0, 1\} \quad (1)$$

GF(2^m)上的模平方运算执行的是两个多项式的乘法操作,然后进行模约减操作,因为系数都要进行模2操作,所以可通过如下的线性组合方式来快速完成:

$$\begin{aligned} C(x) &= A(x)^2 \bmod f(x) \\ &= a_{m-1} x^{2m-2} + \dots + a_2 x^4 + a_1 x^2 + a_0 \end{aligned} \quad (2)$$

在传统的模平方算法中,都是针对特定的m值来进行特定的模平方操作。例如二进制域GF(2¹⁶³)中, m = 163, 域约减多项式 $f(x) = x^{163} + x^7 + x^6 + 1$, 依据此约减多项式, 可以将次数高于162的x多项式元素分别进行约减操作, 逐项进行约减后可以得到如下的结果:

$$\begin{aligned} c_{162} x^{162} &= a_{81} x^{162} + a_{159} x^{162} + a_{161} x^{162} \\ c_{161} x^{161} &= a_{159} x^{161} + a_{162} x^{161} \\ &\dots\dots \\ c_1 x^1 &= a_{82} x^1 + a_{160} x^1 + a_{162} x^1 \\ c_0 &= a_0 + a_{160} \end{aligned}$$

根据上述的约减结果分析, 可以用247个异或门在1个周期内完成二进制域GF(2¹⁶³)上的一次模平方运算。实际应用中不同的椭圆曲线算法需要处理不同位长m的二进制域模运算, 上述的模平方运算并不能满足使用需求, 因此就需要设计一个可配置的灵活模平方单元电路。本文在传统模平方单元电路的基础之上设计了改进的二进制模平方运算单元电路^[4], 实现方式同样也是基于异或相加这样的线性操作, 但相比传统模平方电路可以满足不同的m取值要求, 大大增加了模平方单元应用的灵活性。

算法3 改进的传统模平方算法

input: $A(x) = (a_{m-1}, \dots, a_1, a_0) \in \text{GF}(2^m)$

output: $C(x) = A(x)^2 \bmod f(x)$

1: set reduction matrix Rd(0) ← A(0)

for i: 1 → m - 1 loop

d(2i - 1) ← 0; d(2i) ← A(i)

end loop

```

2: for  $j: 1 \rightarrow m-1$  loop
temp  $\leftarrow d(j)$ 
for  $i: 0 \rightarrow m-2$  loop
temp = tempX or ( $d(m+i)$  and  $\mathbf{R}(j)(i)$ )
end loop
end loop
3: return  $C(x) \leftarrow \text{temp}$ 

```

算法 3 中用到的约减矩阵 $\mathbf{R}$ 及相关的计算求解采用如下公式定义:

$$\begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_{m-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & R_{0,0} & \cdots & R_{0,m-2} \\ 0 & 1 & \cdots & 0 & R_{1,0} & \cdots & R_{1,m-2} \\ \vdots & \vdots & \ddots & \vdots & R_{0,0} & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & R_{m-1,0} & \cdots & R_{m-1,m-2} \end{bmatrix} \begin{bmatrix} d_0 \\ \vdots \\ d_{m-1} \\ d_m \\ \vdots \\ d_{2m-2} \end{bmatrix}$$

$$R_{j,i} \begin{cases} f_j; & j=0, \cdots, m-1, i=0 \\ R_{j-1,i-1} + R_{m-1,i-1} R_{j,0}; & j=0, \cdots, m-1; i=1, \cdots, m-2 \end{cases}$$

针对不同的模平方操作数位长 m , 仅需要改变输入的模约减矩阵的参数 $f(x)$ 就可以实现不同位长的模平方算术操作, 极大的增加了模平方单元的计算灵活性.

2.3.3 模乘单元的设计

传统的模乘单元实现包括串行和并行, 以及一种介于两者之间的 Digit 串行模乘方式. 串行模乘尽管计算周期相对较长, 但因其实现面积和功耗最小, 是最适合于低功耗实现的方案. 与模平方单元不同, 模乘单元实现的是二进制有限域上的两个不同元素的乘法运算, 并不能通过简单的线性操作来设计实现. 本文根据串行乘法器的实现算法, 结合有限域的数学运算规则, 设计了优化的 MSB 交叉串行算法来实现模乘单元电路. 该算法根据 MSB 模乘计算公式, 将计算中重复出现的部分积 $xa(x)$ 进行约减优化设计, 在一次循环中同时进行模乘和模约减两步操作, 具体的计算分解公式如下所示:

$$\begin{aligned} c(x) &= a(x)b(x) \bmod f(x) \\ &= (b_{m-1}a(x)x^{m-1} + b_{m-2}a(x)x^{m-2} + \cdots + b_1a(x)x + b_0a(x)) \bmod f(x) \\ &= ((b_{m-1}a(x)x^{m-2} + b_{m-2}a(x)x^{m-3} + \cdots + b_1a(x))x + b_0a(x)) \bmod f(x) \\ &= (((\cdots((b_{m-1}a(x))x + b_{m-2}a(x))x + b_{m-3}a(x))x + \cdots + b_1a(x))x + b_0a(x)) \bmod f(x) \end{aligned}$$

与传统的串行算法相比, 在输入参数相同的情况下, 经过 FPGA 软件仿真验证, 硬件资源消耗最多可下降 35%, 同时该模乘单元结构简单高效, 通过改变约减多项式 $f(x)$ 和椭圆曲线域参数 m 值, 可以实现不同参数下的模乘计算, 是一种可扩展、低功耗

的模乘算法. 针对不同的约减多项式, 尤其是多项式基表示下的三项式和五项式, 更是可以通过对下图 4 中右侧标注的与门电路进行直接约减, 来进一步提升乘法电路的运算速度和减小相应的乘法电路实现面积. 为了计算上述的串行乘法运算, 本文需要大量计算形如 $xc(x)$ 的中间多项式的值. 对二进制域下的约减多项式两边同时进行 $f(x)$ 取模计算可以得出如下的公式:

$$x^m = f_{m-1}x^{m-1} + \cdots + f_1x + f_0$$

然后将上述公式代入 $xc(x)$ 的计算分解式中能得到如下的详细计算分解公式,

$$\begin{aligned} xa(x) &= x(a_0 + a_1x + \cdots + a_{m-1}x^{m-1}) \\ &= a_0x + a_1x^2 + \cdots + a_{m-1}x^m \\ &= a_{m-1}f_0 + (a_0 + a_{m-1}f_0)x + \cdots + (a_{m-2} + a_{m-1}f_{m-1})x^{m-1} \end{aligned}$$

结合该约减公式并且将串行移位相乘的部分融合在一起, 就可以得到本文所设计的串行乘法器电路, 如图 4 所示^[4].

2.3.4 模逆单元的设计

模逆单元是整个有限域算术单元中资源耗费最大的模块, 实现模逆求解通常有两种计算方法^[1], 一种是基于扩展的欧几里得算法, 此算法的实现原理是通过计算两个多项式之间的最大公约数来求得模逆, 实际应用中该算法更适用于求解素数域内的模逆计算. 本文要求解的是二进制域下模逆计算, 基于费曼小定理 (Fermat's Little Theorem) 的求解方法就为本文所涉及的有限域模逆的计算提供了一个有效的解决方案, 其算法的核心是将模逆运算转换为大量连续的模乘和模平方运算. 以 $m=163$ 为例, 基于费曼小定理的方法求解模逆的计算公式为:

$$\begin{aligned} A^{-1} &= A^{2^m-2} \\ &= A^{2(1+2+2^2+\cdots+2^{m-2})} \end{aligned}$$

可以看出 1 次模逆需要 161 次模乘和 162 次模平方来完成. 尽管如此, 1 次模逆的计算量仍然巨大并且计算周期较长. 本文采用的是 Dimitrov-Jarvien 于 2013 年提出的 DJ 求逆算法, 该算法计算量仅为 9 次模乘和 162 模平方, 大大简化了计算量, 该算法具体实现分解如下述公式所示^[5]:

$$\begin{aligned} 1 + 2 + 2^2 + \cdots + 2^{161} &= \\ (1 + 2 + 2^2) \times (1 + 2^3 + 2^6) \times (1 + 2^9 + 2^{18}) \times \\ (1 + 2^{27} + 2^{54}) \times (1 + 2^{80}) \end{aligned}$$

该算法实现仅需要 2 个 163 bit 的寄存器就可以完成, 可满足节省资源开销、减小面积的目的, 详细的 DJ 算法计算分解过程如算法 4 所示:

算法 4 二进制域下的模拟求解 DJ 算法

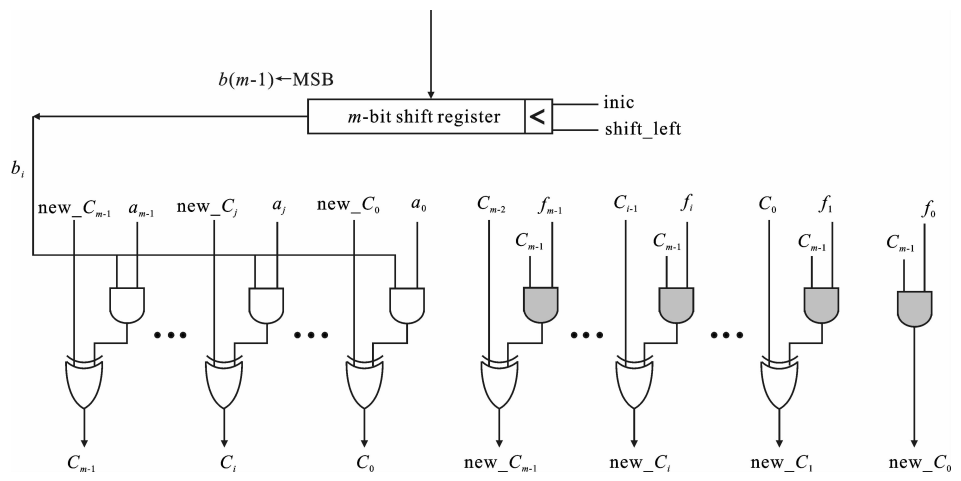


图 4 GF(2^m)域上的 MSB 交叉串行乘法器

INPUT: $A \in GF(2^{163})$
OUTPUT: $C = A^{-1} = A^{2^m-2} (m = 163)$

1. $A \leftarrow A^2$
2. $C \leftarrow A^2$
3. $A \leftarrow A \times C$
4. $C \leftarrow C^2$
5. $A \leftarrow A \times C$
6. $C \leftarrow A^{2^3}$
7. $A \leftarrow A \times C$
8. $C \leftarrow C^{2^3}$
9. $A \leftarrow A \times C$
10. $C \leftarrow A^{2^9}$
11. $A \leftarrow A \times C$
12. $C \leftarrow C^{2^9}$
13. $A \leftarrow A \times C$
14. $C \leftarrow A^{2^{27}}$
15. $A \leftarrow A \times C$
16. $C \leftarrow C^{2^{27}}$
17. $A \leftarrow A \times C$
18. $C \leftarrow A^{2^{81}}$
19. $C \leftarrow A \times C$
20. $C \leftarrow C^2$

μm 工艺下,通过 DC 工具的综合验证,1 次 ECC 点乘仅需要 138 k 个时钟周期完成,等效门数约为 11 900 门,对比同类的 ASIC 实现方案,无论是计算时间和实现面积都有着很大的性能提升,如表 2 所示.

表 1 点乘操作 FPGA 实现测试结果对比

文献	操作域 GF(2 ^m)	实现方式	面积/slices
文献[6]	GF(2 ¹⁶³)	FPGA	10 363
文献[7]	GF(2 ¹⁶³)	FPGA	7 659
本文	GF(2 ¹⁶³)	FPGA	6 428

* 仿真环境:Xilinx FPGA Virtex5

3 实验结果分析

本文选用了GF(2¹⁶³)域上的Koblitz曲线,其中 $a = 1, b = 1, y^2 + xy = x^3 + ax^2 + b$,对该点乘算法进行验证.选用Xilinx公司的Virtex5 FPGA平台进行仿真结果验证、选用ModelSim软件进行波形分析,选用Synopsys公司的Design Compiler软件进行RTL电路综合性能分析.图5是该点乘算法一次点乘运算的仿真结果.

表 2 点乘操作 ASIC 实现测试结果对比

文献	实现方式	计算 周期数	时间 /ms	面积/gates	工艺 / μm
文献[8]	ASIC	275.8k	244.1	12.5k	0.13
文献[9]	ASIC	176.7k	208.5	13.8k	0.13
本文	ASIC	138k	162.9	11.9k	0.13

* 仿真环境:Design Compiler 时钟频率 847.5 kHz 1 次点乘运算 操作域 GF(2¹⁶³)

通过与同类相关方案的对比结果可以看出,本文所设计的点乘算法在实现灵活性、占用面积和计算时间方面,都取得了一定程度的提升和优化.通过采用对Montgomery Ladder点乘算法实现过程的优化改进、MSB 串行乘法器的设计、灵活架构的模平方单元设计和基于DJ算法的模逆单元设计等多种优化方案,本文所提出的点乘设计方案在ECC点乘算法所要求的面积、功耗、计算速度、扩展性等方面取得了有效的平衡.

4 结束语

对于 RFID 和 WSN 等资源受限的应用领域,本

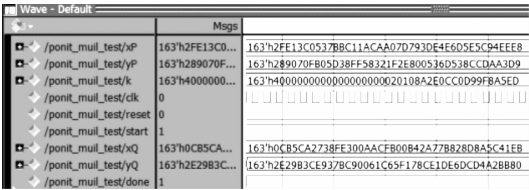


图 5 GF(2¹⁶³)域上一次点乘的计算结果

通过FPGA工具验证及结果分析,本文所设计的ECC点乘法进行1次GF(2¹⁶³)域上点乘运算占用资源为6 428个Slices,相比较同类实现方案,面积优化提升效果显著,如表1所示.在TSMC 0.13

文所设计的 ECC 点乘算法综合权衡了执行速度、面积和功耗等多种指标,采用了以降低功耗为主的主要目标,同时兼顾设计拓展的灵活性,设计出了可配置、低功耗的 ECC 点乘算法硬件实现电路. 综合对比同类的设计方案,无论是 FPGA 实现方式还是 ASIC 实现方式,本文所设计点乘算法都取得了明显的性能提升,可满足资源受限的应用场合的使用需求.

参考文献:

- [1] Hankerson D, Menezes A, Vanstone S. Guide to elliptic curve cryptography[J]. Springer, 2004, 22 (3): 98-101.
- [2] Montgomery P L. Speeding the pollard and elliptic curve methods of factorization [J]. Mathematics of computation, 1987, 177(48): 243-264.
- [3] Li L, Li S. High-Performance pipelined architecture of elliptic curve scalar multiplication over $GF(2^m)$ [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2016, 24(4): 1223-1232.
- [4] Sutter G D, Deschamps J P, Imaña J L. Efficient elliptic curve point multiplication using digit-serial binary field operations[J]. IEEE Transactions on Industrial Electronics, 2013, 60(1): 217-225.
- [5] Azarderakhsh R, Jarvinen K U, Mozaffari-Kermani M. Efficient algorithm and architecture for elliptic

curve cryptography for extremely constrained secure applications[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2014, 61(4): 1144-1155.

- [6] Khan Z U A, Benaissa M. High speed ECC implementation on FPGA over $GF(2^m)$ [C]//Field Programmable Logic and Applications (FPL), 2015 25th International Conference on. London, UK, IEEE, 2015: 1-6.
- [7] 魏国珩, 汪亚, 张焕国. 面向 RFID 应用的 $GF(2^m)$ 域上 ECC 点乘运算的轻量化改进研究[J]. 计算机工程与科学, 2017, 39(1): 81-85.
- [8] Lee Y K, Sakiyama K, Batina L, et al. Elliptic-curve-based security processor for RFID[J]. IEEE Transactions on Computers, 2008, 57 (11): 1514-1527.
- [9] Liu Z, Liu D, Zou X, et al. Design of an elliptic curve cryptography processor for RFID tag chips[J]. Sensors, 2014, 14(10): 17883-17904.

作者简介:

车光宁 男, (1986-), 硕士研究生. 研究方向为 RFID 安全认证协议及加密算法的低功耗设计和实现.

E-mail: cheguangning@163.com.

张钊锋 男, (1972-), 博士, 研究员. 研究方向为 CMOS 射频集成电路及片上系统 SOC、物联网核心芯片和车联网关键技术.

(上接第 14 页)

自设计标准单元库, 通过标准数字流程设计待机电路的方法, 并同时处于常开状态的功耗管理单元 (PMU) 提出了进一步优化方法. 采用自设计标准单元库实现的待机电路通过后端流程及仿真验证, 达到了设计要求, 并有较为理想的功耗表现.

参考文献:

- [1] Tsai-Kan Chien, Lih-Yih Chiou. Low-Power MCU with embedded rram buffers as sensor hub for iot applications [J]. Journal On Emerging and Selected Topics In Circuits and Systems, IEEE, 2016, 6(2): 247-257.
- [2] Atmel. Setting a new low power standard for ARM cortex-M0 + based microcontrollers [EB/OL]. [2017-04-05]. <http://www.atmel.com/products/microcontrol/arm/sam-1.aspx>. 2016.
- [3] Silicon Labs. EFM32TM 32-bit Micro-controllers (MCUs) [EB/OL]. [2017-04-07]. <http://www.silabs.com/products/mcu/32-bit/Pages/32-bit-microcontrollers.aspx>.

[4] ST Microelectronics. STM32L486xx datasheet. Production data[M]. Switzerland, Geneva, ST Microelectronics, 2014.

[5] Texas. Instrumerts. MSP430FR413x 混合信号微控制器[M]. USA, Austin, Texas Instruments, 2014.

[6] 杨紫薇, 朱致玖, 袁甲, 等. 一种针对低功耗 MCU 关断模式的功耗优化方法[J]. 微电子学与计算机, 2017, 34 (6): 26-30.

[7] 梁曼. 适用于结构化 ASIC 的单元库设计[D]. 北京: 中国科学院大学, 2013.

[8] 虞希清. 专用集成电路设计使用教程[M]. 杭州: 浙江大学出版社, 2007.

作者简介:

杨紫薇 女, (1991-), 硕士研究生. 研究方向为低功耗 MCU 关键技术.

袁甲 男, (1985-), 博士, 助理研究员. 研究方向为低功耗数字电路理论、近亚阈值技术、系统容错技术及低功耗数字设计流程.

乔树山 (通信作者) 男, (1981-), 博士, 副研究员. 研究方向为数字集成电路设计、数字电视、绿色无线通信、电力线载波通信. E-mail: qiaoshushan@ime.ac.cn.