Літава Ґ. Метод обчислення точок еліптичної кривої з базисом Радемахера-Крестенсона / Літава Ґ. // Вісник ТНТУ. — 2012. — Том 66. — № 2. — С.207-213. — (приладобудування та інформаційно-вимірювальні технології).

УДК 539.3

Litawa G.

State Higher Vocational School in Nowy Sacz, Poland

ELLIPTIC CURVE POINTS CALCULATION METHOD WITH THE RADEMACHER–KRESTENSON'S BASES

Summary. *The method of calculation for increasing the speed of performance of the basic operation on the elliptic curves, has been proposed. Calculation models using the Rademacher- Krestenson's bases of specially selected points on the elliptic curves, have been presented. The concept of functioning of elliptic curve GF(P) points adder is based on the calculations realized within the Krestenson's bases and parallel summing. FPGA (valve matrix programmed by the user) for the performance of operstions on the elliptic curves the Krestenson's bases and their testing, has been presented.*

Key words: *elliptic curve, GF(p), Rademacher–Krestenson's bases, FPGA, modulo multiplier, points adder.*

Літава Ґ.

Державна вища технічна школа в Новому Сончі, Польща

МЕТОД ОБЧИСЛЕННЯ ТОЧОК ЕЛІПТИЧНОЇ КРИВОЇ З БАЗИСОМ РАДЕМАХЕРА–КРЕСТЕНСОНА

Резюме. *Запропоновано метод збільшення швидкості виконання основних операцій в еліптичних кривих. Наведено моделі обчислень із використанням базису Радемахера-Крестенсона спеціально підібраних точок на еліптичних кривих. Концепція функціонування суматора точок еліптичної кривої GF(p) ґрунтується на обчисленнях, які реалізуються в базисі Крестенсона і паралельним сумуванням. Також представлено ПКВМ (програмована користувачем вентильна матриця) для виконання операцій на еліптичних кривих із використанням базису Крестенсона та висвітлено його тестування.*

Ключові слова: *еліптична крива, GF(p), базис Радемахера-Крестенсона, ПКЛМ, перемножувач за модулем, додавання точок.*

Problem formulation. Over the recent years cipher algorithms relying on elliptic curves have become more and more popular while a rising safety demand still requires longer keys. Growing length of keys calls for more efficient methods and faster calculations on elliptic curves.

Evaluation of recent publications in the explored issue. Available scientific publications suggest various ways to obtain higher calculation rates within elliptic curves operations. Thesis [1] is one notable example. In order to speed up the process of point summation a mixed representation were used. Normal bases were used for representing elements in field GF(2m). Multiplication of elements was based on a multiplication matrix which, in FPGA systems, allowed a completely parallel multiplier, which further on resulted in obtaining a product in one clock cycle. A detailed description of this procedure may be found in thesis [1]. Article [2] describes three algorithms for increasing the pace of basic operations on elliptic curves based on Hybrid Binary-Ternary Number System (HBTNS) invented by Dimitrov and Cooklev in 1995 and described in thesis [3]. A further work published recently is article [4] whose authors are O. Al-Khaleel, Ch.Papachristou, F. Wolff z Case Western Reserve University Cleveland Ohio and K. Pekmestzi of the National Technical University Greece. The system, presented by them, carries out operations on elliptic curves over a field of higher order GF(p). For the summing operations points were represented in projective coordinates which allowed to abandon inverse calculations. Huge numbers operations rely on module addition, subtraction and multiplication.

Short formulation of paper's purpose. Development of elliptic curve points calculation method with Rademacher-Krestenson's bases.

Description of proposed method (algorithm); implementation and testing.

Increase the speed perform basic operations on elliptic curves. In order to make use of calculation method relying on Krestenson's bases it is required to create a point adder in the first place. Elliptic curve point addition or point doubling are basic calculations for this type of cryptography. In the further work we will present a model of an elliptic point adder exploiting projective or mixed coordinates as well as performance summary [5], [6] of such a device in a programmable FPGA unit. The process of adding two points on an elliptic curve GF(p) represented in a mixed [5] way comprises the following array of steps described in table 1.

Table 1. The sequence operations summation points in the mixed coordinate.

$\lambda_1 = X_1 Z_2^2$	$\lambda_7 = \lambda_1 + X_2$
$\lambda_3 = \lambda_1 - X_2$	$\lambda_8 = \lambda_4 + Y_2$
$\lambda_4 = Y_1 Z_2^3$	$Z_3 = Z_2 \lambda_3$
$\lambda_6 = \lambda_4 - Y_2$	$X_3 = \lambda_6^2 - \lambda_7 \lambda_3^2$
	$\lambda_9 = \lambda_7 \lambda_3^2 - 2X_3$
	$Y_3 = \left(\lambda_9 \lambda_6 - \lambda_8 \lambda_3^3\right)/2$

Very important in this case is the speed of these operations. Therefore, further work will devote part of the issue increase the speed of these operations through the use of calculations in the Rademacher-Krestenson's bases.

GF(p) curve point adder model. A model of an elliptic point adder, theoretically, could be made of 11 independent multipliers, 2 adders and 5 subtractors, which may turn out impossible for example in reprogrammable structures as it would most probably lack the necessary logical part. The other way exploits a logical unit controlling the sequence of processes realization. It is important to bear in mind the fundamental assumption that he adder and multiplier are independent systems working on their own. The model of such elliptic point adder is outlined in the picture fig. 1.

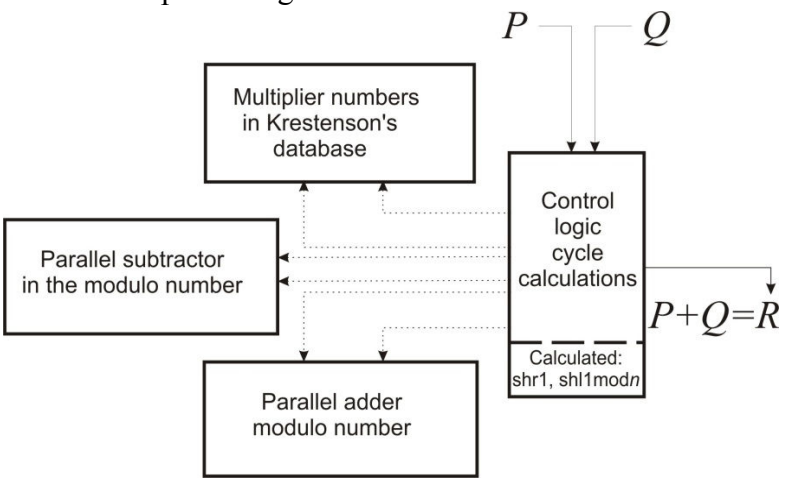

Figure 1. General model summation points on the elliptic curve GF (p).

The adder is derived from foundation developed in thesis [7] with some modifications allowing multiple numbers addition necessary for construction of a multiplier model based on Krestenson's bases. It is assumed that the numbers are fed to the adder in the form of binary sequence. The main task of the model is splitting huge integers into words of specified length m in base $\delta=p^m$ in this case $p=2$ according to $X = x_n\delta^n + x_{n-1}\delta^{n-1} + \ldots + x_1\delta^1 + x_0$. While adding two words $X+Y=Z$, added are two words x_i+y_i represented by integers, where $(x_i + y_i) \bmod \delta$ remains on position i and $(x_i + y_i) \operatorname{div} \delta$ is passed into the older word z_{i+1}. Thanks to the fact that the basis of the division is 2 a binary sequence form of numbers allows an uncomplicated *div* and *mod* operations. The summing process is carried out along with calculation of modulus of the summing result $Z \bmod n$. In our discussion numbers X i Y are smaller than modulus n so $X+Y<2n$. Thus the calculation of modulus Z being sum of X and Y comes down to checking whether $Z > n$ and further, if the condition is true than it is enough to calculate subtraction $Z-n$. Modulus calculation is carried out along with calculation of each word z_i. Numbers subtraction is also carried out simultaneously. Realization of the mentioned above adding model in FPGA system allows to obtain a modulus of sum of 2 numbers in 7 clock ticks. The subtractor's operations are analogical to adder's and need not be described.

A model of huge integers modulo multiplier based on Rademecher-Krestenson's bases and its functioning in programmable structures. A multiplier model based on Krestenson's remaining classes allows multiplication modulo of extremely huge natural numbers without traditional multiplication [8]. Exploiting Krestenson's remaining classes allows a matrix form of multiplied numbers. Finding the product comes down to summing specific elements of the matrix.

Assume two numbers *X* and *Y* and modulus *n*:

$$Z = X * Y \bmod n. \tag{1}$$

In the multiplier model *X* and *Y* are represented as binary sequences

$$\begin{aligned} X &= x_{r-1}2^{r-1} + x_{r-2}2^{r-2} + x_i 2^i + \ldots + x_1 2^1 + x_0 2^0 \\ Y &= y_{r-1}2^{r-1} + y_{r-2}2^{r-2} + y_j 2^j + \ldots + y_1 2^1 + y_0 2^0 \end{aligned} \tag{2}$$

In order to find multiplication result of the above a matrix, shown in table 2, where $m_{ij} = 2^{i+j} / \bmod n$, is constructed.

Table 2. Krestenson's matrix.

…	…	…	…	…	2^{r-1}
…	$(2^{j+i})\bmod n$	…	…	…	2^i
…	…	…	…	…	|
…	…	…	$(2^{1+1})\bmod n$	…	2^1
…	…	…	…	…	2^0
2^{r-1}	2^j	….	2^1	2^0	

The product of the numbers, that is coordinates *X* i *Y* is calculated according to the formula:

$$X \cdot Y \bmod n = \sum_{s,k=1}^{r-1} m_{sk} \bmod n \tag{3}$$

where $x_s, y_k = 1$, that means m_{sk} lies at the intersection of column and row for which respectful x_i i y_j equal 1.

Numbers put in the table are smaller than the given modulus *n*. The sum of numbers within one row or a column, minding the prime assumptions, is smaller than a double modulus, therefore modulo calculations only require comparison and subtracting activities.

Calculation sequence takes the following shape:

1. Generation of Krestenson's matrix according to table 2 and putting it into a 3-dimentional table, where the third dimension depends on the number of words into which numbers were divided, fig. 2.

2. Summing modulo n rows of Krestenson's Matrix according to $\sum_{i;j=1}^{j=r-1} m_{sk} \bmod n$.

The row summing processes are executed parallel, each of *i* rows are summed simultaneously.

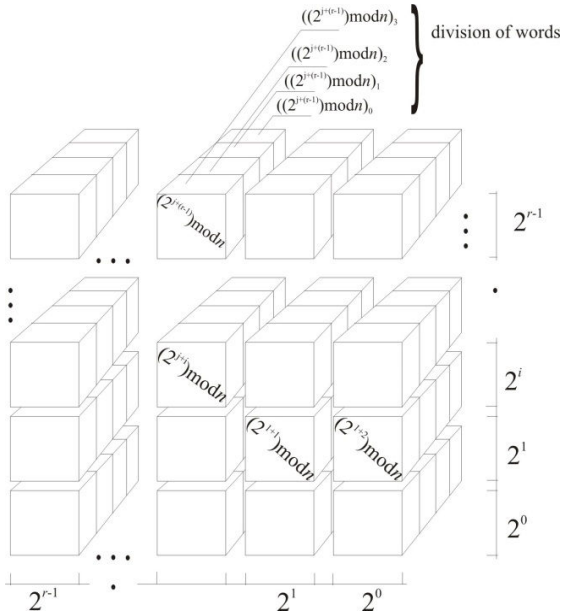

Figure 2. Krestenson's matrix in the model.

Picture 4 explains the idea of matrix rows summing with a resulting vector of size equal *r*-1 sums respectful rows. The vector shown in fig. 3

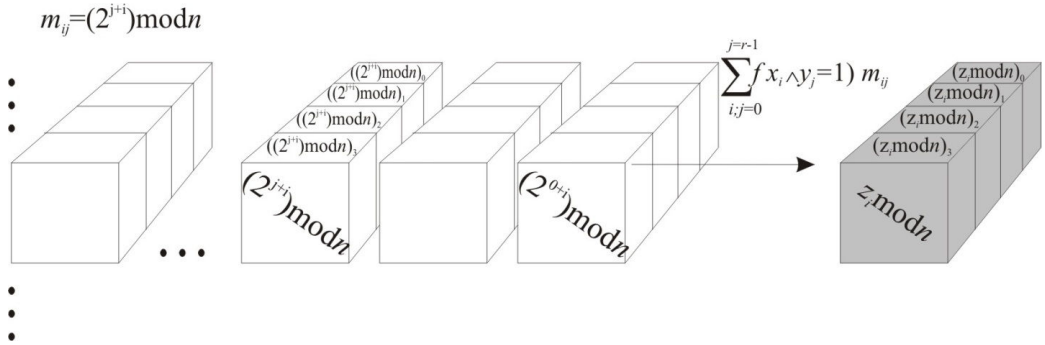

Figure 3. Sum of rows in the Krestenson's matrix.

1. The summing of vector from fig. 4 is executed similarly to the rows summing presented in picture 4. As a result of this operation a product of *X* and *Y* modulo *n* is obtained.

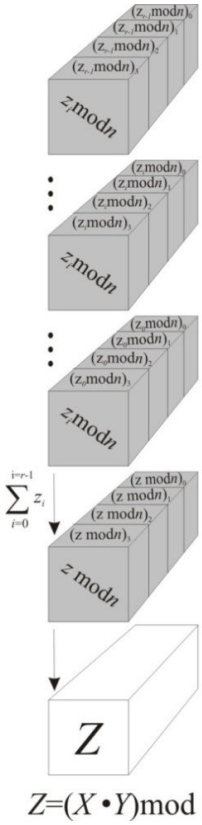

Figure 4. Summing of columns in the Krestenson's matrix.

In order to calculate product of two numbers divided into w words $k = 4w - 2$ steps

The above described multiplying algorithm in FPGA system allows obtaining product of two numbers in $4w + 1$ clock ticks. Table 3 shows performance rates of an adder for numbers of various size.

Table 3. The rate of multiplication for the model-based hardware FPGA for the numbers of different sizes.

Number (bit)	6		1	1	1	1
	9	92	15	38	61	84
Number of multiplications / s	3 538462	25 88235	2 047619	1 640000	1 379310	1 181818

The concept of functioning of elliptic curve GF(P) point adder based on calculations realized within Krestenson's bases and parallel summing. In agreement with prior assumption each of the four calculations may be executed in the same time independently as far as the point summing algorithm structure allows it, see table 1. The whole point summing operation done by the mixed method may be completed in eleven steps. A procedure of calculation sequence choice is presented in table 4.

Table 4. The realization of the calculation summation in two points.

Step	Multiplication	Addition	Subtraction	Shift
1	Z_2^2			
2	$X_1 Z_2^2$			

3	Z_2^3	$\lambda_1 + X_2$	$\lambda_1 - X_2$	
4	$Y_1 Z_2^3$			
5	λ_3^2	$\lambda_4 + Y_2$	$\lambda_4 - Y_2$	
6	λ_6^2			$\frac{\lambda_8}{2}$
7	$\lambda_7 \lambda_3^2$			$\frac{\lambda_6}{2}$
8	λ_3^3	$\lambda_6^2 - \lambda_7 \lambda_3^2$		$\frac{\left(\lambda_7 \lambda_3^2\right)}{2}$
9	$\frac{\lambda_8}{2} \lambda_3^3$		$\frac{\left(\lambda_7 \lambda_3^2\right)}{2} - X_3$	
10	$\frac{\lambda_6}{2} \lambda_9$			
11	$Z_2 \lambda_3$		$\frac{\lambda_9}{2} \lambda_6 - \frac{\lambda_8}{2} \lambda_3^3$	

Simple as it is, a point summing process comes down to execution of operations from table 1. Table 4 shows a way of process grouping so that the operations are executed in the most optimal way. A sensible process grouping together with a right calculation sequence will allow adding two points on an elliptic curve GF(p) in eleven steps. The size of numbers does not influence the number of steps needed to complete the point summing process. The sum of two points on an elliptic curve will be achieved after k=(4w-2)·11, where k – number of steps, w number of words resulting from the division of the original.

The results obtained for implementation in FPGA. Implementation of an elliptic curve *GF*(*p*) point summing unit in FPGA system Stratix III EP3SL150F1152I4SL allowed clock frequency 44MHz for 92bit size and furthermore yielded effectiveness of 234 000 summing operations per second. Effectiveness for other tested sizes is presented in table 5.

Table 5. The speed of the cumulated points GF (p) model for FPGA-based hardware for the numbers of different sizes.

Elliptic Curve *GF*(*p*)	69	92	115	138	161	184
Number of summation / s	319444,4	234042,6	185344,8	148550,7	125000	107142,86

Conclusions. The presented calculation models utilizing Rademacher–Krestenson's bases along with specific representations of points on elliptic curves resulted in a higher pace of basic calculations. Furthermore, real and functioning FPGA systems, operating in accordance with the described theoretical models, proved actual increase in elliptic curve calculation pace. Practical tests outlined possibilities of implementing these calculation models in FPGA systems. Other structures where these models might be implemented are

processors of nVidia video cards supported by CUDA technology or ATI Stream equipped with numerous cores, for example Femi with its 512 cores. Our further research will focus on implementation of the presented models in the mentioned video cards.

References.

1. Majkowski P., Wojciechowski T., Wojdyński M., Rawski M. Realizacja jednostki wspomagającej kryptoanalizę szyfrów opartych na krzywych eliptycznych w strukturach reprogramowalnych, Pomiary Automatyka Kontrola, Vol. 53, Nr 7 2007. – S. 24-26.
2. Adikari J., Dimitrov V., Imbert L. Hybrid Binary-Ternary Number System for Elliptic Curve Cryptosystems, IEEE transactions on computers, VOL. 60, NO. 2, FEBRUARY 2011, (http://www.lirmm.fr/~imbert/pdfs/hybrid_ieeetc_2011.pdf)
3. Dimitrov V.S., Cooklev T.V. Two Algorithms for Modular Exponentiation Based on Nonstandard Arithmetics, IEICE Trans. Fundamentals of Electronics, Comm. and Computer Science, vol. E78-A, no. 1, special issue on cryptography and information security, pp. 82-87, Jan. 1995. (http://eprint.iacr.org/2008/285.pdf)
4. Al-Khaleel O., Papachristou C., Wolff F., Pekmestzi K. An Elliptic Curve Cryptosystem Design Based on FPGA Pipeline Folding, IOLTS '07 Proceedings of the 13th IEEE International On-Line Testing Symposium IEEE Computer Society Washington, DC, USA 2007 (http://bear.ces.cwru.edu/Recent_Papers/iolts07.pdf)
5. Blade I., Seroussi G., Smart N. Krzywe eliptyczne w kryptografii, WNT, Warszawa, 2004.
6. Hankerson D., Menezes A., Vanstone S. Guide to elliptic curve cryptography, Springer, NY 2004. – 332 p. (http://math.boisestate.edu/~liljanab/Crypto2Spring10/GuideToECC.pdf)
7. Makoha A.H., Zuj B.U. The arithmetic of large integers in parallel computer systems, 20.03.2007 (http://revolution.allbest.ru/mathematics/00011260_0.html) (ros.)
8. Yakymenko I., Kasyanchuk M., Nykolajchuk Y.: Matrix algorithms of processing of the information flow in computer systems based on theoretical and numerical Krestenson's basis, TCSET'2010, February 23-27, 2010, Lviv-Slavske, Ukraine. – P. 241.

Отримано 15.02.2012