Polynomials Multiplier under Irreducible Polynomial Module for High-Performance Cryptographic Hardware Tools

Maksat Kalimoldayev[1], Sakhybay Tynymbaev[2], Miras Magzom[2], Margulan Ibraimov[2], Serik Khokhlov[2], Akmaral Abisheva[2], Viktoriia Sydorenko[3]

[1]Information and Computational Technologies Institute, Almaty, 050010, Kazakhstan
[2]Al-Farabi Kazakh National University, Almaty, 050040, Kazakhstan
[3]National Aviation University, Kyiv, 03058, Ukraine

s.tynym@mail.ru margulan.ibraimov@kaznu.kz
magzom_miras@gmail.com ak_maral@mail.ru v.sydorenko@ukr.net

Abstract. One of the most popular and effective methods of information security is cryptographic, that can be realized in software as well as in hardware tools. Hardware cryptographic devices are oriented on confidentiality ensuring put some actual problems must be solved. For the purpose to raise the performance of computing devices productivity, it is necessary to use number systems without the disadvantages of the radix numeration system. This is due to the fact that while performing on multi-digit numbers arithmetic operations represented in the positional system, it becomes necessary to take into account inter-bit transfers that far slows down the computation speed and complicates the calculator structure. The new ways search to improve the computing devices performance led researchers to an objective conclusion that in this direction of the positional number system all possibilities have been exhausted. In order to boost productivity of computing devices, it is necessary to use number systems without such disadvantages.

Keywords: Information Security, Cryptosystem, FPGA, Polynomials, Modular Multiplication.

1. Introduction

Topical tendencies of computer equipment and system development require the elaboration of high-performance computing devices, including information security. By information and communication networks and the integrating devices development the need for creating efficient cryptographic transformations hardware solutions will grow. For example, hardware cryptographic devices are few times faster than software cryptographic tools. But hardware tools have some problems that must be solved to provide efficient confidentiality ensuring.

2. Modern Approaches and Problem Definition

There are tasks leading to calculations when the integer values variables far exceed the maximum range of typical computing devices, defined by the hardware-supported machine word length [1, 2]. The hardware implementation deemed to be efficient from the point of view of processing speed and capabilities, solving such issues by traditional approaches is near impossible [3, 4].

For example, concerning ECC or RSA cryptosystems, the main difficulty in cryptographic transformations is first of all due to the need to perform sequential modular multiplication by multi-digit numbers [5]. In such cryptosystem implementation, an important task is to ensure effective modular multiplication [6].

It should not be overlooked, that most of the modern computing equipment operates in a radix numeration system. For multi-digit numbers arithmetic operations represented in the traditional positional system, a need to take into account inter-bit transfers arises, that significantly slows down the computation speed and complicates the calculator structure.

Consequently, relevant researches devoted for searching new ways to improve the computing devices performance are topical. The studies focused on the use of non-traditional methods of coding numerical information and the corresponding parallel variants of computer arithmetic are of great importance.

The value of each numeric character (number) in the designation of a number depends on its position or digit in the traditional radix numeration system. However, besides this, there are also so-called “non - radix numeration system”, one of which is the “residue number system” (RNS) [7]. RNS application is an efficient way of large data calculations. Particularly, the RNS application allows to increase the operations speed due to the lack of transfer when adding, dividing a large block of input data into smaller sub-blocks and parallel processing.

The residue number system is a data representation system in computational arithmetic, where integer is denoted by a set of smaller numbers.

In the residue number system, a positive integer is represented as a sequence of residues or deductions:

$$A = (a_1, a_2, \ldots, a_n). \tag{1}$$

From dividing to set positive integers $p_1, p_2, \ldots, p_n$, that are called as system basis. α_i numbers are derived in such a way:

$$\alpha_i = A - \left[\frac{A}{p_i} \right] p_i, \ i = \overline{1,n}, \tag{2}$$

where $[A / p_i]$ means whole part from dividing A by p_i. From (2) it follows, that number i-bit α_i of number A is the least positive remainder of the dividing A by p_i and $\alpha_i < p_i$. In this case, the digits formation of each bit is carried out independently of each other. In accordance with the Chinese theorem on remainder number representation A in the form (1) will be unique if the numbers are p_i pairwise simple. The range volume of representable numbers in this case is equals to $P = p_1, p_2, ..., p_n$. In this case, similarly to the radix numeration system, the range of representable

numbers grows as a product of bases, and the digit bits of numbers grows as the sum of the digit capacity of the same bases.
The main privileges that make it possible to effectively use modular arithmetic in some fields of computing technology are: a high level of natural parallelism at the number system level, that is related to the absence of digits transfer in addition and multiplication, as well as the absence of error propagation. In contrast to the radix numeration system, all vector elements are equivalent, and an error in one of them leads only to a dynamic range reduction. This fact allows you to design devices of high fall-over protection and error correction [8].
These features ensure good advantages for RNS over the radix number system at modular operations of addition, subtraction and multiplication. This is especially true if multi-digit numbers act as operands.
Strategic pathway in RNS application in computing is the development of cryptographic information security tools. The research team headed by R.G. Biyashev proposed modular encryption and digital signature generation algorithms based on the nonpositional polynomial number system (NPNS) [9-11]. The purpose of research is the development, research and implementation of information security cryptographic algorithms, developed on the basis of non-positional polynomial number systems, in information and communication systems and networks for various purposes. The block symmetric encryption algorithms developed by them are built on the basis of this approach and are the research results analyzing the possibility of using the non-positional encryption algorithm in practice [12]. Also in this direction there are important works devoted to parallel computation [13-14] as well as papers [15-16].
Taking into account the above stated, the development of computation hardware for the NPNS is an urgent task, the solution of which will provide opportunities for creating efficient cryptosystems hardware implementations based on polynomial RNS.

3. Results and Discussion

Nowadays, the residual numbers system (RNS) is often applied for the development of efficient and high-performance special-purpose processes. RNS is widely applied in cryptography. For example, modular arithmetic allows to create an effective cryptographic systems hardware implementation. The non-positional number systems application allows us to accelerate slow computations in asymmetric encryption algorithms and increase reliability.
The developed non-positional encryption systems, as a cryptographic strength criterion, applies not the key length, but the cryptographic strength of the cryptoalgorithms themselves. The use of non-positional polynomial number systems (NPNS) also makes possible to increase the algorithms efficiency, as in accordance with NPNS rules, all arithmetic operations can be performed in parallel using the modules of the NPNS bases.
For the implementation of the developed algorithms in the form of modules combined into a cryptographic security system (CSS) works on program efficiency are carried out. As well as, work is being carried out to build software and hardware and hardware implementations of cryptographic information security symmetric algorithms based on the NPNS.

As hardware-software and hardware implementation has the best speed characteristics, the cryptographic algorithm integrity is guaranteed and allows to optimize many of the mathematical operations adopted in encryption algorithms. For developed algorithms software and hardware implementation, parts of the procedures are implemented in hardware.

The basic device for non-positional polynomial number systems is a device for multiplying polynomials modulo an irreducible polynomial, where data encryption and decryption routine calculations are performed.

In this research, we consider an approach to polynomials multiplying $A(x)$ and $B(x)$ modulo an irreducible polynomial $P(x)$, that is, $[A(x)*B(x)] \bmod P(x)$, where $\deg A(x),\ \deg B(x) < \deg P(x)$.

In each multiplication process step, the partial remainder r_i is calculated by the former partial remainder shaper by adding modulo two double the previous partial remainder $2r_{i-1}$, with the result of the logical multiplication of the polynomial $A(x)$ (multiplicand) by the next high bit of the polynomial $B(x)$ – multiplier modulo irreducible polynomial $P(x)$.

Then the i-th partial remainder is determined by the formula: $r_i = (2r_{i-1} \oplus A(x)*b_i) \bmod P(x)$, where b_i is the i-th high bit of the binary image of the polynomial $B(x)$, $(b_i = \{0,1\})$. A is the binary image of the polynomial $A(x)$. P is binary image of the irreducible polynomial $P(x)$.

The considered multiplier functional diagram is shown in Fig.1.

The device includes RgA for binary image storing of the polynomial $A(x)$ (multiplicative), shifting the RgB register for the binary image of the polynomial storing $B(x)$ (multiplier), the RgR register for storing the binary image of an irreducible polynomial (module), adder AD1, where modulo 2 sum up the doubled previous remainder $2r_{i-1}$ with the multiplicand $A(x)$ with $b_{i-1} = 1$, forming $C_i = 2r_{i-1} + b_{i-1} * A(x)$. Modulo-two adder (AD2), together with a multiplexer (MS), modifies C_i modulo $P(x)$. Register RgR serves to store intermediate remainders.

Additionally, the multiplier contains a subtracting timing pulse (COUNT), where, at the end of the operation, the “End of Operation” signal is generated. T Trigger, that allow the passage of timing pulses into the circuit.

We consider the multiplier operation. By “START” signal, the binary $A(x)$, $B(x)$ and $P(x)$ polynomials coefficients are received by the blocks of the I1, I2, I3 diagrams, respectively, in the registers $RgA(x)$ and $RgB(x)$, $RgP(x)$. Besides this, by “START” signal, the binary code (k) of the multiplier digits number is received in the TP count. The “Start” signal prior to reaching at the single trigger input T is delayed on the DL.1 delay line. The delay on LZ.1 is determined by the total delay time on $RgA(x)$, I6, AD1, AD2, MS and the recording time of the remainder in the RgR register and the delay time is shifted by *Shf* (L1).

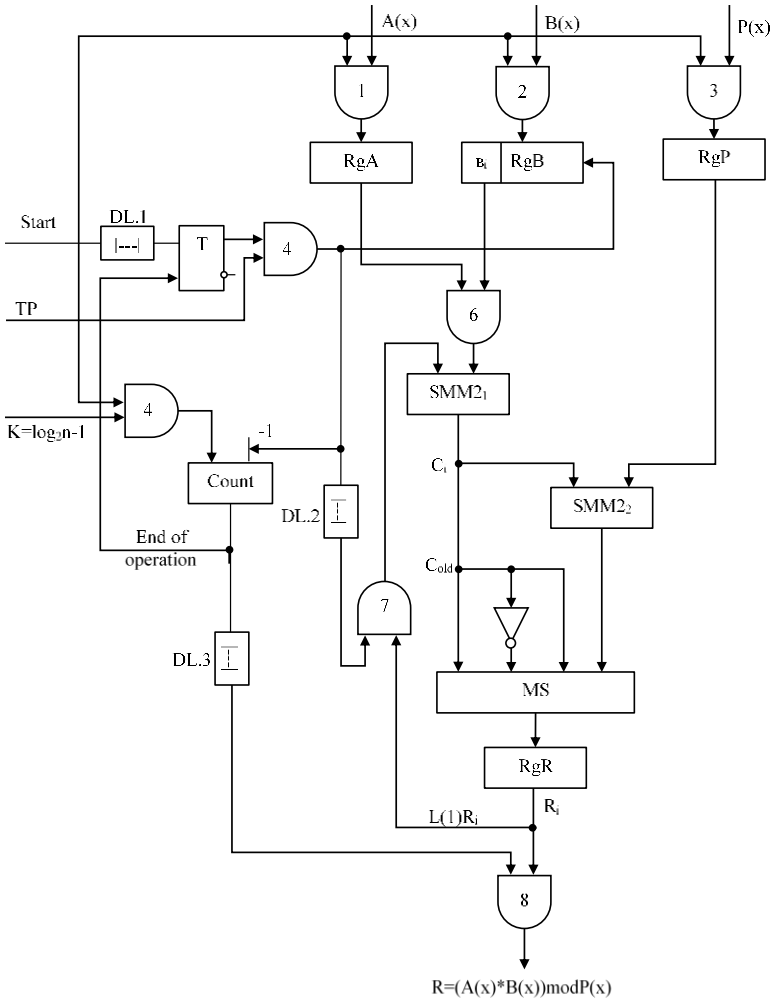

Fig. 1 Functional multiplier diagram of polynomials irreducible polynomials modulo

Upon the “Start” signal is reached the trigger input T and translates it into a single state, that allows the first timing pulse TP1 to pass from the output of the I4 diagram. At this point in the RgR register, the partial remainder is $r_0 = C_0$, with $r_{i-1} = 1$, since $egA(x) < \deg P(x)$.

The first timing signal $RgB(x)$ is shifted to the left by one digit, while in the high order $RgB(x)$ the value of the next coefficient of the polynomial $B(x) - b_{i-2}$ is fixed, provided to the control inputs of the I6 diagrams, and to the other inputs of the polynomial $A(x)$ values coefficients. If, at the same time, $b_{i-2} = 1$, then the polynomial coefficients are provided to the right-hand inputs of the AD1 adder. TSI at the time of the $RgB(x)$ shift is delayed by the delay line DL.2 and is provided to the

control inputs of diagram I7, and the information inputs are supplied by the remainder from the outputs of the Shf diagrams(L1) (L1) $2r_{i-1}$.

From I7 output , the doubled remainder is provided to the left inputs of the AD1 adder. When $b_{i-2}=1,$ the output of this adder is $C_1 2r_{i-1} \oplus A(x)$.

If $b_{i-2}=0,$ $C_1=2r_0$. Next, the C1 value is provided to the left inputs of the adder modulo 2 (AD2). Moreover, if $C_1 < P(x),$ then the multiplexer (MS) outputs the value C_1 and is written to the RgR register forming the value r_1.

If $C_1 \geq P(x),$ then the MS multiplexer outputs the result $C_1 \oplus P(x)$, shaping also the value r_1. Further, the remainder r_1 is shifted one digit to the left by the Shf shifter (L1). At this point, the I4 diagram output of the receives the TP2 timing pulse shifting the contents of the $RgB(x)$ register. AD 1, $RgA(x)$ inputs are provided depending on the value of $b_{i-3},$ and the second inputs are provided with the bits of the residual r_1 multiplied by two. AD1 output, the C2 value is formed and with the help of the adder AD2 and the multiplexer MS, C_2 is modulo, shaping the remainder r_2.

It is noteworthy that when each timing pulse reaches, a unit is subtracted from the TP count. Upon $n-1$ timing signal reaches the RgR register, the result of multiplying the polynomials modulo the irreducible polynomial is generated and the TPC is set to "0" and the counter generates a "end of operation" signal that sets the trigger T to the zero position, preventing the next timing signal from passing output diagram I4. At the time of last remainder shaping signal "end of operation" are delayed on the delay line DL.3. After that, the result is given to the outputs by the diagram I8.

If to consider an example of multiplying polynomials modulo an irreducible polynomial in the multiplier diagram shown in Figure 1.

Let $A(x)=x^4+x+1,\ B(x)=x^4+x^2+1,\ P(x)=x^5+x^2+1.$

Binary representations of polynomials are presented below (see Table 1): $A=100112;\ B=10,012;\ P=1001012.$

Table 1. Example of multiplying

№	RgR bits	AD1	AD2
1	2	3	4
Start	$b_4=1$	$A=010011$ $\oplus$ $\underline{000000}$ $A=\overline{010011}$	$C_0=010011$ $\oplus$ $\underline{P(x)\,100101}$ $\overline{r_1=\ 010011}$
TP1	$b_3=0$	$C_1=2r_1=100110$	$C_1=100110$ $\oplus$ $\underline{P(x)\,100101}$ $\overline{r_2=\ 000011}$

TP 2	$b_2 = 1$	$C_2 = 2r_2 + A =$ 000110 $\oplus$ $\underline{010011}$ 010101	$C_2 = 010101$ $\oplus$ $\underline{P(x)\ 100101}$ $r_3 = 010101$ as $C_3 < P(x)$
TP 3	$b_1 = 0$	$C_3 = 2r_3 + 0 = 101010$	$C_3 = 101010$ $\oplus$ $\underline{P(x)\ 100101}$ $r_4 = 001111$
TP 4	$b_0 = 1$	$C_4 = 2r_4 + A =$ 011110 $\oplus$ $\underline{010011}$ 001101	$C_4 = 001101$ $\oplus$ $\underline{P(x)\ 100101}$ $r_5 = 001101$

Checking: $(x^4 + x + 1) * (x^4 + x^2 + 1) = (x^3 + x^2 + 1)$, accordingly binary display of this polinomial – 01101_2.

This algorithm was tested on Nexys 4 Artix-7 FPGA Board. Figure 2 contains diagram for encoding and decoding the number A in hexadecimal.

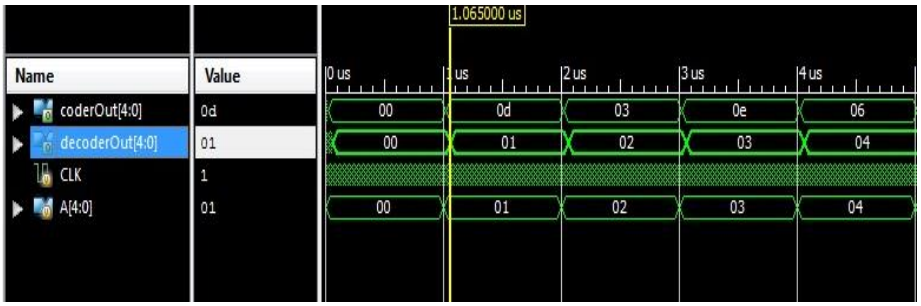

Fig. 2 The timing diagram of the algorithm for an 8-bit number

Table 2. The number of resources used in encoding and decoding as a percentage

Slice Logic Utilization	% of the FPGA resource used when encoding and decoding 4-bit code	% of the FPGA resource used when encoding and decoding 8-bit code	% of the FPGA resource used when encoding and decoding 12-bit code	% of the FPGA resource used when encoding and decoding 24-bit code
Number of Slice Registers	0.02%	0.1%	0.39%	1.25%
Number of Slice LUTs	0.05%	0.33%	1.48%	4.9%
Number of bonded IOBs	7%	13%	19%	36%
Number of BUFG/BUFG CTRLs	3%	3%	3%	3%

Table 2 shows the number of resources used in encoding and decoding processes in percents.

4. Conclusions

The precondition research for is the growing need to create efficient hardware solutions for cryptographic transformations and the difficulties that arise in using the radix numeration system.

As was stated above, the basic privileges of nonpositional number system applying is the absence of transfer of digits in the operations of addition and multiplication, and, consequently, the parallel operations possibility on each of the bases of the system, which significantly speeds up the calculation process. It stands to mention that most modern general-purpose processors are not able to effectively perform nonpositional number system calculations.

For the most effective implementation of computing devices based on the residual number system, it is required to develop non-standard circuit solutions that effectively perform calculations in a nonpositional number system.

References

1. A. Poschmann, Lightweight Cryptography – Cryptographic Engineering for a Pervasive World. IACR ePrint archive 2009, 516 p., (2009).
2. J. Guajardo, S. Kumar, C. Paar, J. Pelzl, Efficient software-implementation of finite fields with applications to cryptography, Acta Applicandae Mathematica, Vol. 93, Iss. 1-3, pp 3-32, 2006.
3. Kalimoldayev M., Tynymbayev S., Gnatyuk S., Ibraimov M., Magzom M. The device for multiplying polynomials modulo an irreducible polynomial News of the National Academy of Sciences of the Republic of Kazakhstan, Series of Geology and Technical Sciences, 2 (434), pp. 199-205 (2019).
4. Tynymbayev S., Gnatyuk S.A., Aitkhozhayeva Y.Z., Berdibayev R.S., Namazbayev T.A. Modular reduction based on the divider by blocking negative remainders, News of the National Academy of Sciences of the Republic of Kazakhstan, Series of Geology and Technical Sciences, 2 (434), pp. 238-248, 2019.
5. N. Gura, Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs'' Proc. 6th Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES 04) LNCS 3156. – Springer (2004).
6. D. Schinianakis., T. Stouraitis, RNS-based RSA and ECC cryptography basic operations, algorithms, and hardware, Embedded Systems Design with Special Arithmetic and Number Systems, Springer (2017).
7. Hu Z., Gnatyuk S., Kovtun M., Seilova N. Method of searching birationally equivalent Edwards curves over binary fields, Advances in Intelligent Systems and Computing, Vol. 754, pp. 309-319, 2018.
8. S. Gnatyuk, M. Kovtun, V. Kovtun, A. Okhrimenko, Search method development of birationally equivalent binary Edwards curves for binary Weierstrass curves from DSTU 4145-2002, Proceedings of 2nd Intern. Scientific-Practical Conf. on the Problems of Infocommunications. Science and Technology (PIC S&T 2015), Kharkiv, Ukraine, pp. 5-8 (2015).
9. M. Kalimoldayev, R. Biyashev, S. Nyssanbayeva, Y. Begimbayeva, Modification of the digital signature, developed on the nonpositional polynomial notations, Eurasian Journal of Mathematical and Computer Applications, ISSN 2306–6172, Volume 4 , Issue 2 (2016), pp. 33-38 (2016).

10. R. Biyashev, M. Kalimoldayev, S. Nyssanbayeva, M. Magzom, Development of an encryption algorithm based on nonpositional polynomial notations, Proceedings of the International Conference on Advanced Materials Science and Environmental Engineering pp. 112-118 (2016).

11. S. Gnatyuk, A. Okhrimenko, M. Kovtun, T. Gancarczyk, V. Karpinskyi, Method of Algorithm Building for Modular Reducing by Irreducible Polynomial, Proceedings of the 16th International Conference on Control, Automation and Systems, Gyeongju, Korea, pp. 1476-1479 (2016).

12. S. Gnatyuk, V. Kinzeryavyy, M. Iavich, D. Prysiazhnyi, Kh. Yubuzova, High-Performance Reliable Block Encryption Algorithms Secured against Linear and Differential Cryptanalytic Attacks, CEUR Workshop Proceedings (Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, Kyiv, Ukraine, May 14-17, 2018), vol. 2104, pp. 657-668 (2018).

13. M. Ciet, M. Neve, E. Peeters, J. Quisquater, Parallel FPGA implementation of RSA with residue number systems-can side-channel threats be avoided?, In 2003 46th Midwest Symposium on Circuits and Systems, Vol. 2, pp. 806-810, 2003.

14. R. Hobson, P. McGinn, Co-processor for performing modular multiplication, U.S. Patent No. 6,209,016. 27 Mar. 2001.

15. V. Krasnobayev, S. Koshman, A. Yanko, A. Martynenko, Method of Error Control of the Information Presented in the Modular Number System, Problems of Infocommunications. Science and Technology: International Scientific-Practical Conference, pp. 35-42, 2018.

16. V. Krasnobayev, S. Koshman, A Method for Operational Diagnosis of Data Represented in a Residue Number System, Cybernetics and Systems Analysis, vol. 54, issue 2, pp. 336-344, 2018.