**IRSTI 81.93.29**
**UDC 004.056**

# PIPELINE MODULAR MULTIPLIER

**S. Tynymbayev[1], R. Sh. Berdibayev[2], A. A. Shaikulova[3], S. Adilbekkyzy[4], T. A. Namazbayev[5]**
[1,2,3]Almaty University of Power Engineering and Telecommunication, Almaty, Kazakhstan
[4]L.N. Gumilyov Eurasian National University, Nur-Sultan, Kazakhstan
[5]Al-Farabi Kazakh National University, Almaty, Kazakhstan
[1]s.tynym@mail.ru, [2]r.berdybaev@aues.kz, [3]shaikulova_ak_al@mail.ru,
[4]sairan.02.95@mail.ru, [5]tirnagog@mail.ru
[1]ORCID ID: 0000-0002-9326-9476
[2]ORCID ID: 0000-0002-8341-9645
[3]ORCID ID: 0000-0001-9634-143X
[4]ORCID ID: 0000-0002-3929-7070
[5]ORCID ID: 0000-0002-2389-2262

**Abstract.** Various approaches of multiplying multi-bit numbers modulo are considered. The algorithm of multiplication of numbers is presented, where the process of multiplication modulo is divided into steps and at each step the operation of multiplication is combined with the operation of reducing numbers modulo forms a partial remainder. The circuit solutions for pipeline multiplication of numbers modulo with the analysis of the least significant bits of the multiplier are considered. The proposed modular multiplier does not require preliminary calculations and the calculation results do not go beyond the bit grid of the module. To evaluate the effectiveness, the ratios are used, by which the timing parameters of the multipliers are determined without a pipeline and using a pipeline. Algorithmic validation and verification of the pipeline modular multiplier was carried out on a Nexys 4 board based on the FPGA Artix-7 from Xilinx. Verilog HDL is used to describe the circuit of the pipeline multiplier. The results of a timing simulation of the device are presented in the form of time diagrams, confirming the correct operation of the device.

**Keywords:** public key cryptosystem, hardware encryption, modular multiplication, remainder former, pipelined multiplier.

## Introduction

In asymmetric cryptosystems, the encryption and decryption of data is carried out by exponentiation the numbers $a$ to the power $x$ modulo $P$ ($a^x \bmod P$), which can be implemented in software, hardware-software and hardware [1, 2]. Hardware encryption has a number of significant advantages over software, which has a higher speed and guarantees its integrity [3]. At the same time, the generation and storage of keys, as well as encryption, are carried out in the encoder board itself, and not from a portion of the computer's RAM, it provides security for the implementation of the algorithm itself. Thus, the implementation of the algorithm is protected, which is also an important advantage. Therefore, the development of high-speed operating units of hardware cryptoprocessors for asymmetric encryption, despite their high cost, is an urgent task.

## Modular multiplication approaches

The multiplication of numbers modulo can be done in two ways. In the first method, the operation is divided into two stages. At the first stage, n-bit numbers $A$ and $B$ are multiplied and a 2n-bit number C is formed. At the second stage, the product $C = A * B$ is reducing modulo $P$.

Nowadays, much experience has been gained in the development of high-speed integer multipliers and a squaring device. These include Braun and Wallace multipliers, Dadd multipliers, systolic and Vedic multipliers and quadrants, where the computational complexity is $O(n^2)$ bit operations. But these multipliers are very effective in computing "low-bit" numbers, which are widely used in the construction of operating units of computers of various classes [4].

In cryptography for multiplying multidigit numbers, which make it possible to calculate the required product faster than in $O(n^2)$ steps (bitwise operations), the Karatsuba method [5] is widely used, the complexity of which is $O(n^{\log_2 3})$, Toom-Cook algorithm [6] with complexity of order $O(n2^{\sqrt{2\log_2 n}})$ bit operations. Besides the Schönhage-Strassen algorithm [7] allows you to multiply two n-bit numbers in $O(n \log n \log n)$ bit operations.

The modular reduction operation, which is performed in the second stage, is to obtain the remainder of dividing the product $C = A * B$ by the module $P$. In [8], various modular reduction methods were analyzed. It is shown that the most effective construction tool is a modular reduction device based on a dividing device. The composition of such a dividing device includes a shaper of partial remainders. High-performance matrix and pipeline units for modulating numbers of numbers are easily implemented on the basis of shapers of partial remainders [9, 10, 11, 12, 13].

In the second method, by using the algorithms of Barrett or Montgomery [14, 15, 16] accelerates the process of multiplying large numbers modulo. However, these algorithms require preliminary constant calculations related to the need to use the algorithm for dividing large numbers. Barrett's algorithm requires precomputation constant

$$\mu = \left\lfloor \frac{d^{2m}}{N} \right\rfloor,$$

where $d = 2^k$, $k$ is word size in bits, $m$ is number of words in module $N$.

The Montgomery algorithm requires precomputing the constant "$r^2 (\mathrm{mod}\, N)$", using division with remainder.

In the third method, the process of multiplying numbers modulo is performed in many steps, where at each step the multiplication operation is combined with the operation of casting numbers modulo, forming partial remainders. The number of steps is determined by the bit depth of the multiplier.

Multipliers of numbers modulo sequential actions with the analysis of the least significant and highest bits of the multipliers are considered in [17, 18]. Matrix multipliers of numbers modulo were considered in [19, 20].

The main disadvantages of these multipliers are their low speed.

In this paper we consider pipelined device for modular multiplication with analysis of low-order bits, which is an architectural technique for increasing productivity for multiplying of numbers modulo.

**Pipeline modular multiplier**

With pipeline multiplication of numbers, the entire process of multiplying numbers modulo breaks down into a sequence of completed stages. Each of the stages of the multiplication procedure is performed by the logic gate blocks of its stage, and these logic gate blocks work in parallel.

The results obtained at the i-th stage are transferred to the (i+1)-th stage for further processing. Information is transferred from one stage to another through a buffer memory, which is located between them.

The synchronization of the pipeline is provided by clock pulses, the period of which is determined by the slowest stage of the pipeline and the delay in the buffer memory element.

For simplicity of explanation, we consider the operation of the pipeline using the example of a 4-stage pipeline. The number of pipeline steps is determined by the bit depth of the multiplier. Its structure is shown in fig. 1. The composition of the pipeline is the register PrA, PrB and PrP, where, before the start of multiplication, the multiplier A and factor B and module P are

respectively taken.

Each pipeline stage consists of blocks of logic circuits and buffer registers. The number of pipeline steps is determined by the bit depth of the multiplier.

The logic gate blocks of the 1-st stage of the pipeline include the block of the And1 circuit and the former of the partial residuals of the PRF.1. Buffer registers of the 1-st stage of the pipeline are RegB.1, Regr.1, RegR.0, RegP.1.

The logic gate blocks of the 2-nd stage include the block of logic gate circuits And2, PRF.2 and the adder modulo P AddMP.1 registers RegB.2, Regr.2, Regr.1 RegP.2 are buffer registers of the 2-nd stage pipeline.

The block of logic gate And3. PRF.3 and AddMP.2 comprise logic gate blocks of the 3-rd stage pipeline. The registers RegB.3, Regr.3, Regr.2 RegP.3 are buffer registers of the 3-rd stage.

The logic gate blocks of the 4-th stage include the block of logic circuits And4, AddMP.3, the buffer register of the 4-th stage is the register R.

When transferring bits, the multiplier B from the buffer register of the i-th stage of RegB.i+1 to the register (i+1)-th stage of RegB.i+1 only those bits that have not yet entered the operation are transmitted.

When each clock pulse is applied, the corresponding modules P are moved from the buffer register of the i-th stage RegP.i to the buffer register of the i+1-th stage.

After the supply of the first clock pulse CP1 along its leading edge, the contents of the input registers that make up the first three numbers A1, B1, P1 are transferred to the buffer registers of the first stage. In this case, the logical operation A1 & b3 is performed in the block of circuits And.1 and, with b3 = 1, the remainder R0 = A1 is formed at the outputs of the block of circuits And1, which is received in the RegR.0 buffer register.

At the same time, the partial remainder r1 is formed at the outputs of the PRF.1 by performing the operations $r_1 = 2A_1 \bmod P = 2A_1 + \overline{P}_1 + 1$. For this, the multiplier $A_1$ shifted by one bit to the left is fed to the first input of the PRF.1, and the inverse code of the $\overline{P}_1$ module and signal +1 are fed to the PRF.1 other inputs. The value of $r_1$ is received in the buffer register Regr.1 of the first stage of the pipeline.

At the same time, the contents of RegB without bit b0 are transferred to the buffer register RegB.1, and the contents of RegP are transferred to the buffer register RegP.1 of the first stage.

On the trailing edge of the pulse CP1 input registers take the second three numbers $A_2$, $B_2$, $P_2$.

After the second clock pulse CP2 has been applied, the contents of the input registers RegA, RegB and RegP, $A_2$, $B_2$, $P_2$ are transferred to the buffer registers of the first stage and during the transfer of the block of circuits And1, the operation $A_2 \,\&\, b_0$ is performed and at $b_0 = 1$, the remainder $R_0 = A_2$. $R_0$ is formed at the outputs of the circuit And1. $R_0$ is received in the RegR.0 buffer register of the first stage. In addition, a partial remainder $r_1 = 2A_2 \bmod P_2 = 2A_2 + \overline{P}_2 + 1$ is formed in PRF.1, which is stored in the Regr.1 buffer register.

At the same time, the CP2 pulse transfers the contents of the first stage buffer registers to the second stage buffer register. In this case, in the block of circuits And2 and the adder modulo $P$, an operation is performed to calculate the intermediate remainder $R_1$ according to the formula

$$R_1 = \left[\left(r_1 \,\&\, b_1\right) + R_0\right] \bmod P_2$$

which is fixed in the buffer register of the second stage Regr.1.

At the same time, the former of the second stage of PRF.2 forms a partial remainder r2, which is fixed in the Regr.2 buffer register.

On the trailing edge of the clock signal CP2, also in the input registers RegA, RegB and

RegP the values of the numbers of the third three A3, B3, P3 are received.

After the arrival of the third clock pulse CP3, the contents of the input registers (A3, B3, P3) are transferred to the buffer registers of the first stage, and the contents of the second stage are transferred to the buffer register of the third stage.

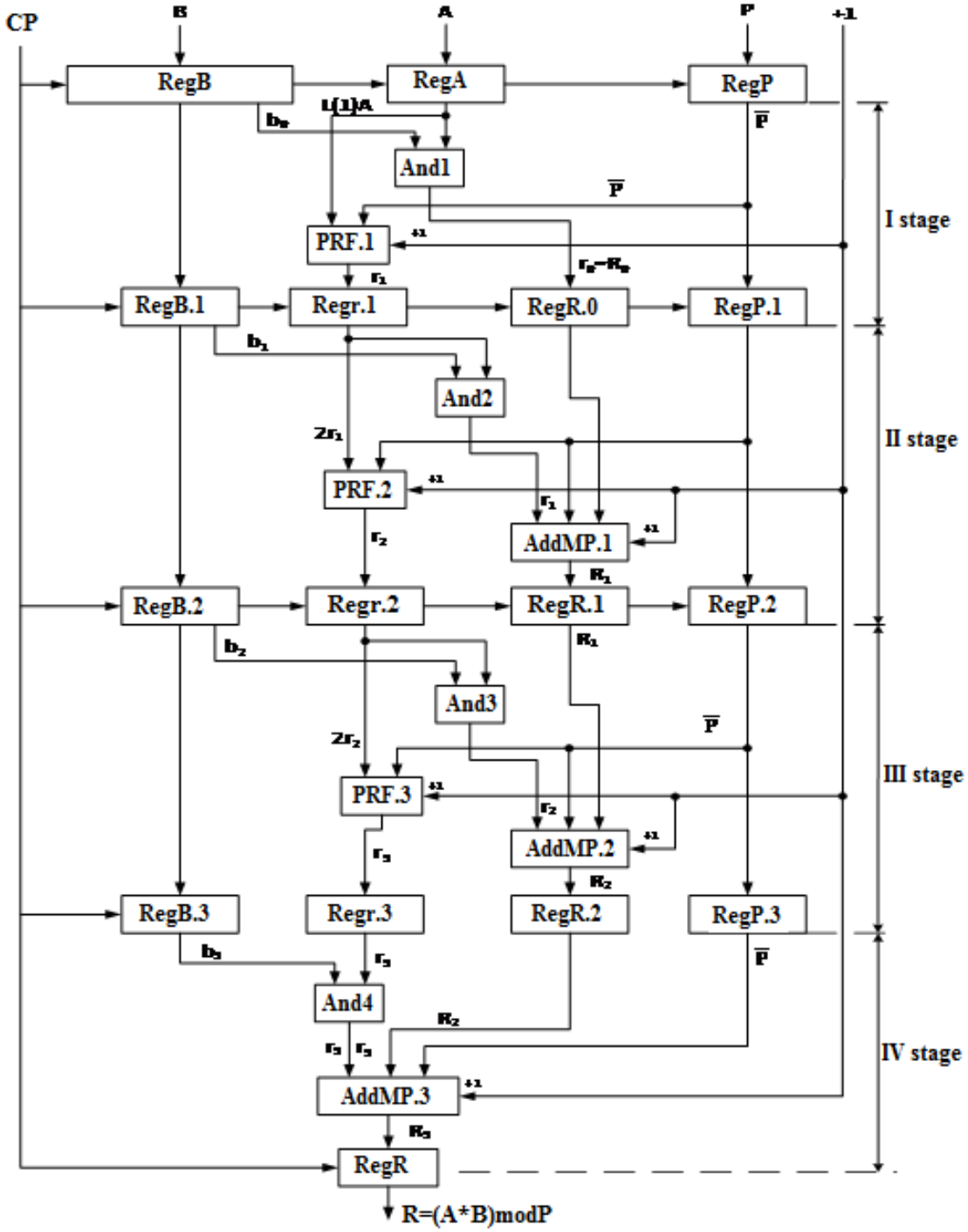


**Figure 1** – 4-stage pipelined device for multiplying numbers modulo, where the multiplication begins with the analysis of the least significant bits of the multiplier

In this case, in the first stage the value of the intermediate residue $R_0 = A_3 \,\&\, b_0$ is formed, which is fixed in the RegR.0 buffer register and the PRF.1 former computes $r_1 = 2A_3 + \overline{P}_3 + 1$, which is stored in the Regr.1 buffer register.

In the second stage R1, the adder modulo P AddMP.1 calculates where the operation

$R_1 = [(r_1 \,\&\, b_1) + R_0] \bmod P_2$ is performed, which is stored in the buffer register Regr.1 of the second stage of the pipeline. In addition, the value $r_2 = 2r_1 + \overline{P}_2 + 1$, which is stored in the Regr.2 buffer register of the second stage of the pipeline, is calculated by the PRF.2.

Under the action of CP3 in the third stage of the pipeline, the value of the partial remainder r3 is calculated by the PRF.3 shaper, where the operation $r_3 = 2r_2 + \overline{P}_1 + 1$ is performed, which is received in the buffer register Regr.3 of the third stage of the pipeline. In addition, an intermediate remainder $R_2 = [(r_2 \,\&\, b_2) + R_1] \bmod P_1$ is formed by the adder AddMP.2, which is stored in the buffer register of the third stage Regr.2.

On the trailing edge of CP3, the input registers take the fourth three-number sets A4, B4, P4.

After the fourth CP4 clock pulse is fed into the pipeline, the contents of the input registers (A4, B4, P4) are transferred to the buffer register of the first stage, and the contents of the buffer registers of the first stage are transferred to the buffer registers of the second stage, the contents of the second stage are transferred to the register of the third stage, and the contents of the third step are transferred to the buffer register of the fourth step.

With a clock pulse CP4 in the first stage of the pipeline, the logic gate block And1 calculates the partial remainder R0 by logically multiplying A4 with b0. The value of R0 of outputs And1 is recorded in the buffer register RegR.0 of the first stage, at the same time the partial remainder $r_1 = 2A_4 \bmod P_4$, which is written in the buffer register Regr.1, is calculated by the PRF.1 shaper.

By the fourth clock pulse the PRF.2 shaper calculates the value of the partial remainder $r_2 = 2r_1 + \overline{P}_3 + 1$, which is received in the Regr.2 buffer register. The third clock pulse is also calculated by the adder AddMP.1 intermediate remainder $R_1 = [(r_1 \,\&\, b_2) + R_0] \bmod P_3$, which is formed in the buffer register Regr.1 of the second stage of the pipeline.

The fourth clock pulse in the third stage of the pipeline by the PRF.3 former calculates the value of the partial remainder $r_3 = 2r_2 + \overline{P}_2 + 1$, which is received in the Regr.3 buffer register. The intermediate remainder is also calculated by the adder AddMP.2 $R_2 = [(r_2 \,\&\, b_2) + R_1] \bmod P_2$, which is written to the Regr.2 buffer register.

At the same time, a fourth remainder in the fourth stage of the pipeline generates a partial remainder by the And4 circuits and the adder AddMP.3 calculates the value $R_3 = [(r_3 \,\&\, b_3) + R_2] \bmod P_1$, which is recorded in the Regr.3 buffer register, which is the result of multiplying the number A1 by B1 modulo P1, those. $R = (A_1 \,\&\, B_1) \bmod P_1$.

After feeding the inputs of the pipeline CP5, CP6, etc. at the outputs of Regr.3 we get the results of multiplication $(A_2 \,\&\, B_2) \bmod P_2$, $(A_3 \,\&\, B_3) \bmod P_3$, etc.

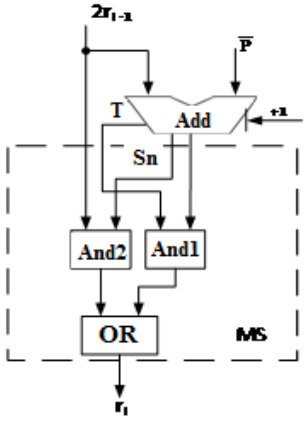


**Figure 2** – The structure of the PRF

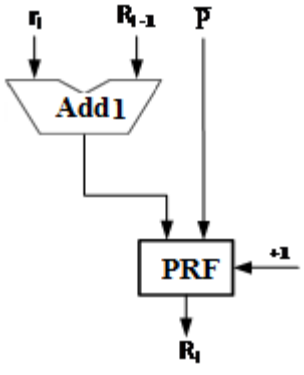


**Figure 3** – The structure of the adder modulo P (AddMP)

Fig. 2 shows the functional diagram of the remainder former (PRF), which consists of a binary adder (Add) multiplexer (MS). The multiplexer, in turn, consists of an And1, And2 and OR circuit. The adder has an output P, which captures the transfer from the sign discharge and the sign output $(Sn - Sign)$. If $T$ (transfer) = 1, then Sn = 0 and vice versa. The value of the previous remainder is fed to the left inputs of the adder with a shift of it by one bit in the direction of the highest bit (2ri-1). The right inputs of the Add adder are supplied with bits of the inverse code of the $\overline{P}$ module. To transfer $\overline{P}$ to an additional code, the level +1 is applied to the low order of the adder.

When $2r_{i-1} \geq P$, then the result of addition at the outputs of the adder by transferring T = 1 is output. When $2r_{i-1} < P$, then the value 2ri-1 is given to the output of the circuit with the signal Sn = 1.

In the first way $r_i = 2r_{i-1} + \overline{P} + 1$, and in the second way $r_i = 2r_{i-1}$.

The structure of the adder modulo P (AddMP) is shown in fig. 3, which consists of the adder Add1, where $r_i$ and $R_{i-1}$ are summed and this sum $(R_{i-1} + r_i)$ is modulo P given by the PRF scheme.

**Efficiency of pipelined multipliers**

To evaluate the effectiveness, it is necessary to determine the ratios by which the time parameters of the multipliers are determined without a pipeline and using a pipeline [4].

The multiplication time of numbers without a pipeline is determined by the formula $NKT_k$, where K is the number of three-number sets to be processed, N is the number of pipeline steps, $T_k$ is the duration of the clock period, which is determined by the ratio Tk = TAddMP + TBReg, where TAddMP is the summation time by module, TBReg is time of writing the result of processing to the buffer registers.

The execution time of operations on the K input streams of numbers (three-number sets) on the N steps of the pipeline with a clock period Tk can be determined by the relation.

$$T_{NK} = (N + (K-1))T_K.$$

This formula reflects that before the output of the pipeline output of the calculation of the first three polynomials, K cycles must pass, and subsequent results will follow in each cycle.

Then the acceleration of computing S due to pipelining can be described by the formula

$$S = \frac{NKT_k}{(N + (K-1))T_k} = \frac{NK}{N + (K-1)}.$$

At $K \to \infty$ the acceleration tends to a value equal to the number of steps K in the pipeline. The gain in time C can be calculated using the formula:

$$C = (NK - (N + K - 1))T_k.$$

Consider an example of the execution of multiplication operations modulo on a four-stage pipeline of four three-number sets:

A1 = 14, B1 = 13, P1 = 15; A2 = 11, B2 = 9, P2 = 15;

A3 = 10, B3 = 11, P3 = 15; A4 = 4, B4 = 7, P4 = 15.

The calculation results in each step on the pipeline steps for all four three-number sets are shown

in table 2.

In this table, Rij are the numbers of intermediate remainders i (i = 0 ÷ 3) and three-number sets j, where j = 1 ÷ 4.

**Table 1**. The calculation results

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Three-number sets and clock pulses / Stages | A1 = 14, B1 = 13, P1 = 15 | A2 = 11, B2 = 9, P2 = 15 | A3 = 10, B3 = 11, P3 = 15 | A4 = 4, B4 = 7, P4 = 15 | – | – | – |
| | CP1 | CP2 | CP3 | CP4 | CP5 | CP6 | CP7 |
| I | R01 = 14 | R02 = 11 | R03 = 10 | R04 = 4 | – | – | – |
| II | | R11 = 14 | R12 = 11 | R13 = 0 | R14 = 12 | – | – |
| III | | | R21 = 10 | R22 = 11 | R23 = 0 | R24 = 13 | – |
| IV | | | | R31 = 2 | R32 = 9 | R33 = 5 | R34 = 13 |

Check: R31 = (A1 * B1) mod P1 = (14 * 13) mod 15 = (182) mod 15 = 2;
R32 = (A2 * B2) mod P2 = (11 * 9) mod 15 = (99) mod 15 = 9;
R33 = (A3 * B3) mod P3 = (10 * 11) mod 15 = (110) mod 15 = 5;
R34 = (A4 * B4) mod P4 = (4 * 7) mod 15 = (28) mod 15 = 13.

For our pipeline: $T_{NK} = \left(N + \left(K - 1\right)\right)T_k = 7T_K$.

Multiplication time without pipeline: $NKT_K = 16T_K$.

The time gain is: $C = \left(NK - \left(N + K - 1\right)\right)T_K = 9T_K$.

Fig. 4 shows the timing diagram and the results of modulo multiplication for the above four three-number sets on a four-stage pipeline. Verilog language was used to describe the circuit of the pipeline multiplier.

Artix-7 from Xilinx companies was chosen as the FPGA board.

As can be seen from fig. 4, the first three numbers A1, B1, P1 are fed to the inputs of the i-th stage of the pipeline after the first clock pulse. In this case the intermediate balance R01 = 14 of the

first three numbers is calculated.

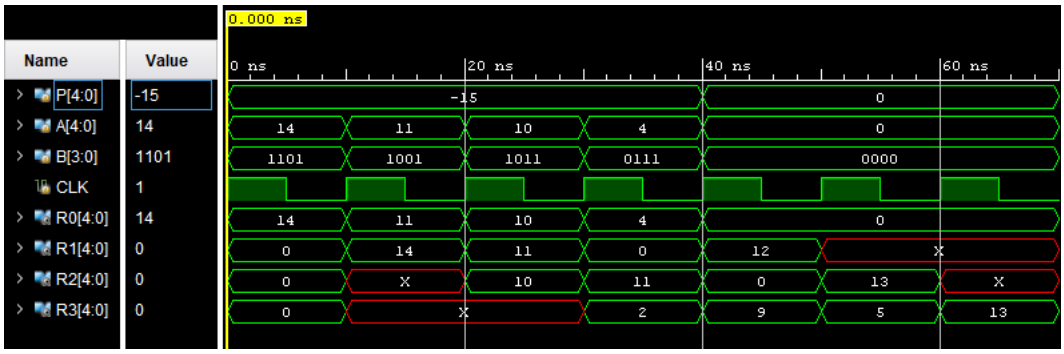


**Figure 4** - Timing diagram of the pipeline circuit

During the action of the second clock pulse, the values of the second three-number sets A2, B2, P2 are received at the pipeline input and the intermediate remainder R02 = 11 is calculated, and at the second stage of the pipeline, the intermediate remainder R11 = 14 from the first three numbers is calculated. Similarly, after the third clock pulse is fed to the pipeline input, the value of the third three of numbers A3, B3, P3 is received and an intermediate balance R03 = 10 is formed, and in the second and third stages, R12 = 11, R21 = 10 are formed, respectively.

After applying the fourth clock pulse to the inputs of the first stage of the pipeline, a triple of numbers A4, B4, P4 are received and the intermediate remainder R04= 4 is calculated on the first stage of the pipeline. And on the remaining steps, intermediate residues R13 = 0, R22 = 11 and R31 = 2 are formed. This R31 = 2 is the result of multiplying numbers modulo (A1 * B1) mod P1.

After applying the fifth clock pulse to the II, III, IV stages of the pipeline, R14 = 12, R23 = 0 and R32 = 9 are calculated respectively. R32 = 9 is the result of multiplying the numbers modulo (A2 * B2) mod P2.

After applying the sixth clock pulse to the III and IV stages of the pipeline, the corresponding residues R24 = 13 and R33 = 5 are calculated. R33 = 5 is the result of multiplying the numbers modulo (A3 * B3) mod P3.

After applying the seventh clock pulse to the IV stage of the pipeline, the remainder R34 = 13 is calculated, which is the result of multiplying the numbers modulo (A4 * B4) mod P4.

## Conclusion

In the proposed pipeline modular multiplier, preliminary calculations are not required. At each stage of the formation of the intermediate remainder, the operations of multiplication and reduction are combined. All calculations do not go beyond the module bit grid.

**Acknowledgement.** The work was carried out by the authors within the program-targeted financing of the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (BR053236757 "Development of software and hardware and software for cryptographic protection of information during its transmission and storage in infocommunication systems and general purpose networks").

## References

[1] Ryabko B. Y., Fionov AI. Osnovy sovremennoy kriptografii dlya spetsialistov v informatsionnykh tekhnologiyakh [“Fundamentals of modern cryptography for information technology professionals”]. M.: Scientific world, 2014. 173 p. (In Russian)

[2] Akhmetov B. S., Korchenko A. G., Sidenko V. V., Drens Y. A., Seilova N. A. Prikladnaya kriptologiya: metody shifrovaniya [“Applied cryptology: encryption methods”]. Almaty: Satbayev

Univercity, 2015. 496 p. (In Russian)

[3] Aitkhozhayeva E. Zh, Tynymbayev S. T. Aspektyi apparatnogo privedeniya po modulyu v asimmetrichnoy kriptografii [Aspects of hardware reduction modulo in asymmetric cryptography], Bulletin of National Academy of Sciences of the Republic of Kazakhstan, 2014. 5. 88–93. ISSN: 1991-349421. (In Russian)

[4] Orlov S. A., Tsilker B. J. Organizaciya EHVM i system ["Organization of computers and systems"], 3rd ed., SPb.: Peter, 2014. ISBN 978-5-496-01145-7. (in Russian)

[5] Karatsuba A. A., Ofman Y. P. Umnozheniya mnogorazryadnykh chisel na avtomatakh ["Multiplications of multi-digit numbers on automata] . DANSSR, 1962. 145. 293–314. (in Russian)

[6] Cook S. A., Aanderaa S. O. On the minimum computation time of functions. Trans. AMS, 142 (1969). 291–314.

[7] Schonhage A., Strassen V. Bystroye umnozheniye bol'shikh chisel. Kiberneticheskiy sbornik ["Fast multiplication of large numbers". Cybernetic collection]. 1973. 2.87–98. (in Russian)

[8] Kovtun M., Kovtun V. Obzor i klassifikaciya algoritmov deleniya i privedeniya po modulyu bolshih celyh chisel dlya kriptograficheskih prilozheniy [Review and classification of algorithms for dividing and modulating large integers for cryptographic applications], Cipher Company. URL: http://docplayer.ru/30671408-Obzor-i-klassifikaciya-algoritmov-deleniya-i-privedeniya-po-modulyu-bolshih-celyh-chisel-dlya-kriptograficheskih-prilozheniy.html (date of access: 20.12.2019) (In Russian)

[9] Petrenko V. I, Chipiga A. F. (1995). Umnojitel' po modulu [Modulus multiplexer]. Kombinatsionnyy rekurrentnyy formirovatel' ostatkov [Combination recurrent former of remainders]. Patent of the Russian Federation. No. 2029435 (In Russian)

[10] Petrenko V. I., Sidorchuk A. V., Kuz'minov J. V. (2007) Umnojitel' po modulu [Modulus multiplexer]. Patent of the Russian Federation. No. 2299461 (In Russian)

[11] Kopytov V. V., Petrenko V. I., Sidorchuk A. V. (2009). Ustroystvo dlya formirovaniya ostatka po proizvol'nomu modulu ot chisla [Device for generating remainder from arbitrary modulus of number]. Patent of the Russian Federation. No. 2368942. (In Russian)

[12] Tynymbayev S. T., Aitkhozhayeva Y. Zh., Adilbekkyzy S. High speed device for modular reduction. Bulletin of National Academy of Sciences of the Republic of Kazakhstan, 2018. 6(376). 147–152. ISSN 2518-1467 (Online). ISSN 1991-3494 (Print). doi: https://doi.org/10.32014/2018.2518-1467.38.

[13] Tynymbayev S., Berdibaev R. S., Omar T., Shaikulova A. A., Magauin B. Bystrodeystvuyushchiye ustroystva privedeniya chisla po modulyu ["High-speed devices of reduction"]. Materials of the XIV International Asian School-Seminar "Problems of optimization of complex systems. Almaty, 2018. 2 (In Russian)

[14] Barrett P. Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor. In: Odlyzko A.M. (eds) Advances in Cryptology — CRYPTO' 86. CRYPTO 1986. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. 1986. 263. doi: https://doi.org/10.1007/3-540-47721-7_24.

[15] Montgomery P. L. Modular Multiplication without Trial Division. Mathematics of Computation, 1985. 44(170). 519–521. doi: https://doi.org/10.2307/2007970.

[16] Pisek E., Henige T. M. (2013) Method and apparatus for efficient modulo multiplication. Patent US No.8417756 B2.

[17] Tynymbayev S., Berdibayev R. Sh., Omar T., Gnatyuk S. A., Namazbayev T. A., Adilbekkyzy S. Device for multiplying modulo numbers with analysis of the lower bits of the multiplier. Bulletin of National Academy of Sciences of the Republic of Kazakhstan,2019. 4(380). 38–45.

[18] Tynymbayev S., Aitkhozhayeva Y. Zh., Berdibayev R. Sh., Abilda B. G. A multiplier of numbers modulo with analysis of the two most significant bits of the multiplier per step. Materials of the International scientific-practical conference "Actual problems of information security in Kazakhstan". Almaty, 2020. 236–241.

[19] Tynymbayev S., Berdibayev R. Sh., Aitkhozhayeva Y. Zh., Omar T., Adilbekkyzy S. The matrix matrix of the multiplier of numbers modulo, where the multiplication begins with the analysis of the least significant bits of the multiplier. Certificate of state registration of rights to the object of copyright of the MOJ of the RK, no. 2689, dated: Apr. 8, 2019.

[20] Tynymbayev S., Berdibayev R. Sh., Aitkhozhayeva Y. Zh., Omar T., Shaikulova A. A., Magauin B. The matrix scheme of the multiplier of numbers modulo, where the multiplication begins with the highest digits of the multiplier. Certificate of state registration of rights to the object of copyright of the MOJ of the RK, no. 2690, dated: Apr. 8, 2019.