основы информационной техники: Учеб. пособие для вузов / Ф.Е. Темников, В.А. Афонин, В.И. Дмитриев. – 2-е изд., перераб. и доп. – М.: Энергия, 1979. – 512 с., Wr W.W., Haccoun D., Peile R., Hirata Y. Coding for Satellite Communication // IEEE Journal on Selected Areas in Communications. – May, 1987. – VoI. SAC-5, № 4. 4. Берлекэмп Э.Р. Техника кодирования с исправлением ошибок // ТИИЭР. – 1980. – № 5. – С. 24–58. 5. Бояринов И.М. Помехоустойчивое кодирование числовой информации. – М.: Наука, 1983. – 189 с. 6. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки / Пер. с англ.; Под ред. Добрушина. – М.: Радио и связь, 1985. – 248 с. 7. Кларк Дж., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи. – М.: Радио и связь, 1987. – 392 с. 8. Дмитриев В.И. Прикладная теория информации: Учеб. для студ. вузов по спец. "Автоматизированные системы обработки информации и управления". – М.: Высш. шк., 1989. – 320 с. 9. Цымбал В.П. Теория информации и кодирование: Учебник. – 4-е изд., перераб. и доп. – К.: Вища. шк., 1992. – 263 с. 10. Муттер В.М. Основы помехоустойчивой телепередачи информации. – М.: Радио и связь, 1994. – 293 с. 11. Витерби А.Г. Границы ошибок для сверточных кодов и ассимптотический оптимальный алгоритм декодирования // Некоторые вопросы теории кодирования. – М.: Мир, 1976. – 360 с. 12. The Art of Error Correcting Coding. Robert H. Morelos-Zaragoz. Copyright © 2002 John Wiley & Sons Ltd. 13. А. с. СССР 492872. Устройство для декодирования линейных сверточных кодов / В.В. Золотарев. 14. Волошок В.О., Ліпіч К.Л., Юрчук А.В. Порогове декодування згорткового коду // Матеріали наук.-техн. конф. ІППТ при Нац. ун-ті "Львівська політехніка". – Львів, 2006.

УДК 681.3, 004.3

В.С. Глухов
Національний університет "Львівська політехніка",
кафедра електронних обчислювальних машин

ОБЧИСЛЮВАЛЬНИЙ ПРИСТРІЙ ДЛЯ ОПЕРАЦІЙ НАД ЕЛІПТИЧНИМИ КРИВИМИ

Описано особливості побудови обчислювальних пристроїв для операцій над еліптичними кривими.

The aspects of dedicated processors arcitecture for operations over elliptic curves are described.

Вступ

З 1 січня 2004 року в Україні офіційно дозволено користуватися електронним цифровим підписом замість звичайного. Тому актуальними є розроблення й порівняння різних архітектур спеціалізованих процесорів, які можуть апаратними способами реалізовувати потрібні для отримання і перевірки цифрового підпису операції. Цьому питанню, недостатньо розкритому у вітчизняній літературі, присвячена ця стаття.

Аналіз публікацій та окреслення проблеми

В Україні діють два стандарти на цифровий підпис: міждержавний стандарт ГОСТ 34.310-95 та національний стандарт України ДСТУ 4145-2002 [1]. Існують національні стандарти в інших країнах, а також міжнародний стандарт IEEE1363-2000 [2]. Математичним основам застосування еліптичних кривих присв'ячені роботи [3–8].

Стандарт [1] визначає використання для отримання цифрового підпису еліптичних кривих і полей Галуа $GF(2^p)$, елементи яких можна подати як у поліноміальному, так і у нормальному

базисах. В основу процедур отримання і перевірки цифрового підпису згідно з стандартом ДСТУ 4145-2002 [1] покладено операції над елементами поля Галуа $GF(2^p)$ (p – просте число, далі такі поля, які відповідають вимогам стандарту [1], у цій статті називатимуться просто полями Галуа). Елементи поля Галуа можуть утворювати поліноміальний і нормальний базиси. Про виконання операцій у нормальному базисі (особливо про множення) в стандарті говориться досить загально. Також не окреслено області застосування поліноміального і нормального базисів, їхні переваги та недоліки, а також вплив обраного базису на подальші дії над еліптичними кривими, необхідними для отримання та перевірки цифрового підпису. Ці питання недостатньо розкрито у вітчизняній літературі.

Мета дослідження

У статті робиться спроба визначити переваги та недоліки застосування різних способів подання елементів поля Галуа шляхом аналізу стандартів та літературних джерел, а також шляхом експерементальної перевірки деяких неведених у стандарті [1] і літературі алгоритмів та методів. Особливу увагу приділено виконанню операції множення елементів поля Галуа.

Метою роботи є проведення аналізу операцій, які необхідні для виконання множення двох елементів поля Галуа у нормальному та поліноміальному базисах, визначення особливостей використання цих базисів, які впливають на виконананя подальших операцій над еліптичними кривими, розробка і порівняння структур помножувачів та операційних пристроїв, які будуть виконувати вказані операції.

Синтез помножувача елементів поля Галуа у нормальному і поліноміальному базисах

Елементи $\left\{t^{m-1},...,t^2,t,1\right\}$ основного поля Галуа утворюють поліноміальний базис, елементи $\left\{\theta,\theta^2,\theta^{2^2},...,\theta^{2^{m-1}}\right\}$ основного поля Галуа утворюють нормальний базис (t і θ – корені полінома, що утворює поле). Усі інші елементи основного поля Галуа можна подати як у поліноміальному базисі (у вигляді $a_{m-1}t^{m-1}+...+a_2t^2+a_1t+a_0$), так і у нормальному базисі (у вигляді $a_0\theta+a_1\theta^2+a_2\theta^{2^2}+...+a_{m-1}\theta^{2^{m-1}}$), де a_i – двійкові розряди (i = 0, 1, …, m-1). Стандарт [1] рекомендує використовувати поля з поліноміальними базисами, які утворюються примітивними многочленами з трьома або п’ятьома членами (для спрощення обчислень), але рекомендує брати з цих многочленів такі, для яких $m \geq 163$. Також стандарт [1] рекомендує використовувати поля з оптимальними нормальними базисами (для спрощення обчислень), які утворюються примітивними многочленами, для яких $m \geq 173$.

Додавання двох елементів у полі Галуа виконують як порозрядне додавання за модулем 2.

Під час множення двох елементів поля Галуа у поліноміальному базисі:

1) виконують опрацію додавання за модулем 2 (xor). При цьому можливе також використання різних методів прискорення множення (методу Карацуби [3], методу Монтгомері (Montgomery) [4]);

2) множення виконують за модулем p. За модулем p береться або весь результат множення, або кожний проміжний результат (так званий помножувач Мастровіто (Mastrovito, рис. 1) [5].

Нижче наведено фрагмент опису комбінаційної частини Mul помножувача Мастровіто (рис. 1), коли m=173:

– початок фрагмента

```
signal p : STD_LOGIC_VECTOR(172 downto 0):= "0..0100100101";
        g0:for i in 172 downto 0 generate
                o(i)<= (b(i) and a) xor r(i) xor (p(i) and c);
        end generate;
```

– кінець фрагмента.

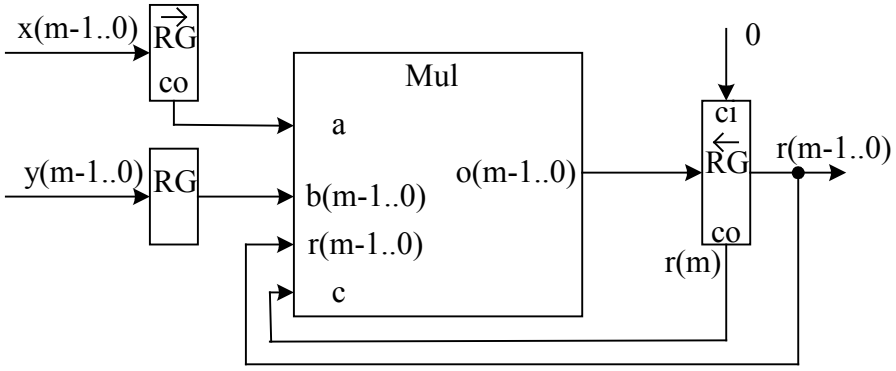


Рис. 1. Помножувач Мастровіто

Під час множення двох елементів (x_N та y_N) поля Галуа у нормальному базисі (далі множення у нормальному базисі) потрібно виконати такі операції:

скласти систему рівнянь

$$t = a_{0,0} + a_{0,1}t + a_{0,2}t^2 + ... + a_{0,m-1}t^{m-1} (\mathrm{mod}\, p(t))$$

$$t^2 = a_{1,0} + a_{1,1}t + a_{1,2}t^2 + ... + a_{1,m-1}t^{m-1} (\mathrm{mod}\, p(t))$$

$$t^4 = a_{2,0} + a_{2,1}t + a_{2,2}t^2 + ... + a_{2,m-1}t^{m-1} (\mathrm{mod}\, p(t))$$

……

$$t^{2^{m-1}} = a_{m-1,0} + a_{m-1,1}t + a_{m-1,2}t^2 + ... + a_{m-1,m-1}t^{m-1} (\mathrm{mod}\, p(t))$$

з системи рівняннь утворити матрицю A з елементами $a_{i,j}$ (у разі правильно обраного полінома, що утворює поле, детермінант матриці A $\det A \neq 0$);

у полі Галуа знайти матрицю B, обернену до A: $B=A^{-1}$, $\det B \neq 0$.

утворити допоміжну матрицю C, де c_i – коефіцієнти полінома $p(t) = t^m + c_{m-1}t^{m-1} + ...c_1 t + c_0$, що утворює відповідне поле Галуа;

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,m-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m-1,0} & a_{m-1,1} & \cdots & a_{m-1,m-1} \end{bmatrix}, \; C = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ c_0 & c_1 & c_2 & \cdots & c_{m-1} \end{bmatrix};$$

обчислити допоміжну матрицю D = ACB;

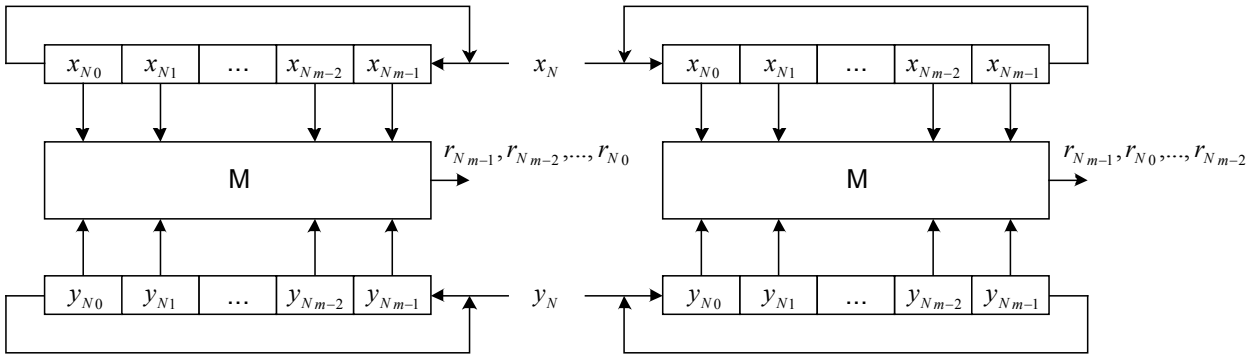
з матриці D утворити помножувальну матрицю M, з елементами $\mu_{i,j} = d_{j-i,-i}$.

Тоді старший розряд результату $r_{N(m-1)} = x_N * M * y_N^t$.

Наступні розряди результату ($r_{N(m-2)}$, …, $r_{N(0)}$) обчислюють за цією самою формулою, тільки замість самих векторів x_N та y_N використовуються їхні послідовні циклічні зсуви на один розряд вліво. Цю схему множення ілюструє рис. 2.

У полі Галуа елементами матриці M будуть тількі 0 та 1, за використання оптимального нормального базису кількість 1 у матриці буде мінімально можливою і дорівнюватиме 2*m-1.

На практиці операції з матрицями перетворюються на обчислення згідно з відомими формулами множення матриць, велика кількість 0 у матриці дає змогу істотно спростити ці формули.



а б

Рис. 2. Помножувач за стандартом [1] (а) та за методом Мессі–Омури [6] (б)

На відміну від стандарту [1] за відомим методом Мессі–Омури (Massey–Omura, [6]) використовують правий зсув векторів x_N та y_N. Розряди добутку на тій самій матриці M отримують при цьому в іншому порядку. При проведенні деяких наступних операцій над еліптичними кривими краще використовувати саме правий зсув, який у нормальному базисі рівноцінний піднесенню цього елемента до квадрата.

Можна отримати розряди добутку, використовуючи правий зсув у тому самому порядку, що і з використанням лівого, але при цьому необхідно трансформувати матрицю M.

Нижче наведено фрагмент опису послідовності обчислення одного розряду результату $r_{N(m-1)} = x_N*M*y_N^t$ для примітивного полінома з p=173 з використанням правого зсуву, де позначено:

$x_N(i)$, $y_N(i)$, s(i) – i-й розряд операнда x_N, y_N та проміжного результату;

o – вихід матриці (1 біт).

-- початок фрагмента обчислення

$$r_{N(m-1)} = x_N*M*y_N^t \quad (1)$$

s(172) <= y_N (172) and (x_N(171));

s(171) <= y_N (171) and (x_N(172) xor x_N(20));

…

s(1) <= y_N (1) and (x_N(70) xor x_N(22));

s(0) <= y_N (0) and (x_N(21) xor x_N(0));

$r_{N(m-1)}$ <= s(172) xor s(171) xor … xor s(0);

-- кінець фрагмента.

Як видно з наведеного опису, математична матриця M реалізується у вигляді логічної матриці (рис. 3), яка складається з:

матриці з m-1 двовходових суматорів за модулем 2 (xor2);

матриці з m двовходових елементів 2I;

1-го m-входового суматора за модулем 2 (xor_m);

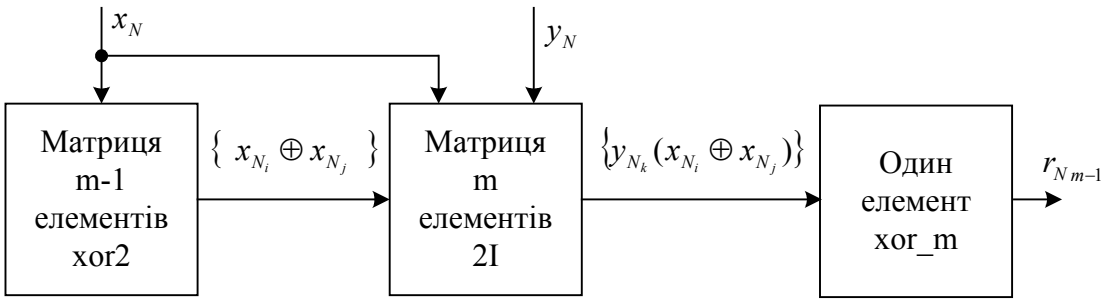


Рис. 3. Логічна матриця M

На відміну від множення елементів поля Галуа у поліноміальному базисі, де усі дії виконують над словами, під час множення у нормальному базисі дії виконують над окремими розрядами операндів, причому над різними розрядами двох операндів. Тому програмна реалізація множення у нормальному базисі буде повільнішою за програмну реалізацію у поліноміальному базисі.

У табл. 1 наведено результати порівняння часу виконання множення в полі Галуа $GF(2^{173})$ програмними способами.

Таблиця 1

Порівняння часу виконання множення програмним способом

Спосіб множення	Час виконання, %
Множення у поліноміальному базисі	100
Елементи подано у нормальному базисі, множення здійснюють у поліноміальному базисі	240
Множення у нормальному базисі з використанням нескороченої матриці M	4500

Для ефективної програмної реалізації множення у нормальному базисі необхідно створювати та використовувати спеціалізований однорозрядний процесор з побітовою організацією пам'яті.

Апаратна реалізація не дає явних переваг жодному з базисів (табл. 2). Менше значення комплексного показника відповідає кращому варіанту (LUT, slices – функціональні елементи ПЛІС ф. Xilinx).

Таблиця 2

Порівняльні характеристики апаратних помножувачів для m=173

Базис	Апаратні витрати, slices	Апаратні витрати, LUT	Максимальна тактова частота, МГц	Комплексний показник, LUT/МГц
Поліноміальний, рис. 1	275	526	146	3,6
Нормальний, рис. 2, *а*	383	577	169	3,4

І за методом Мессі–Омури, і за стандартом [1] для отримання всіх розрядів добутка використовують одну матрицю протягом m тактів роботи. Прискорення множення полягає в одночасному знаходженні усіх розрядів добутка за один такт роботи. Для цього можна використати:

1) m однакових матриць M згідно з описом (1), на входи кожної наступної матриці подаються циклічно зсунуті операнди з входів попередньої матриці (рис. 4, rot – операція циклічного зсуву ліворуч або праворуч);

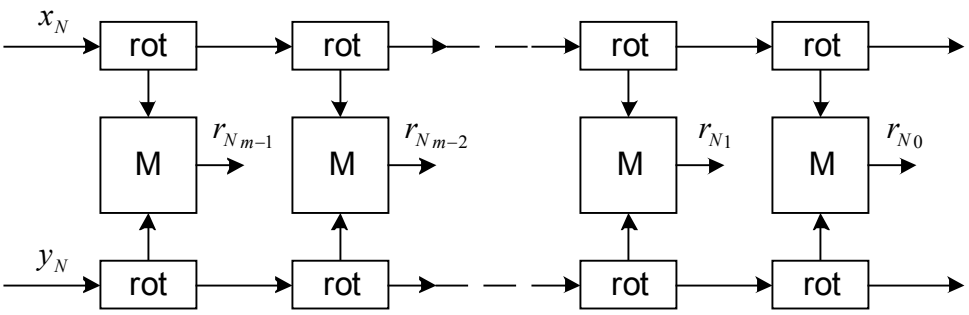


Рис. 4. Одночасне визначення усіх розрядів добутку (варіант 1)

2) m різних матриць M_0, …, M_{m-1}, на входи усіх матриць подаються одні й ті самі операнди (рис. 5);

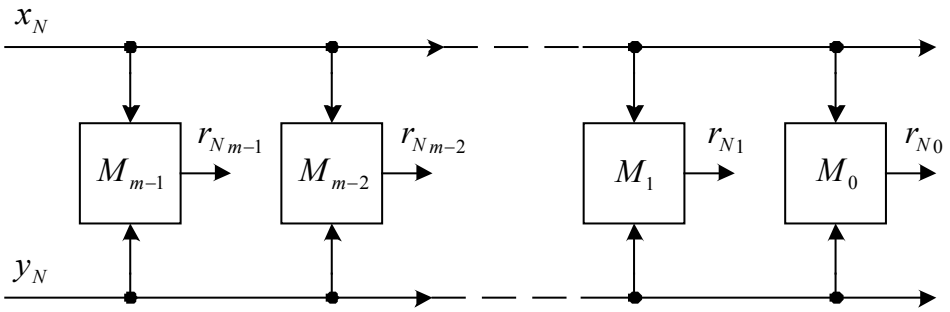


Рис. 5. Одночасне визначення усіх розрядів добутку (варіант 2)

3) одна матриця, яка складається з набору двовходових суматорів за модулем 2 (xor) з усіх m матриць попереднього варіанта 2. Це усуває дублювання елементів у матрицях, що зменшує апаратні витрати (рис. 6).

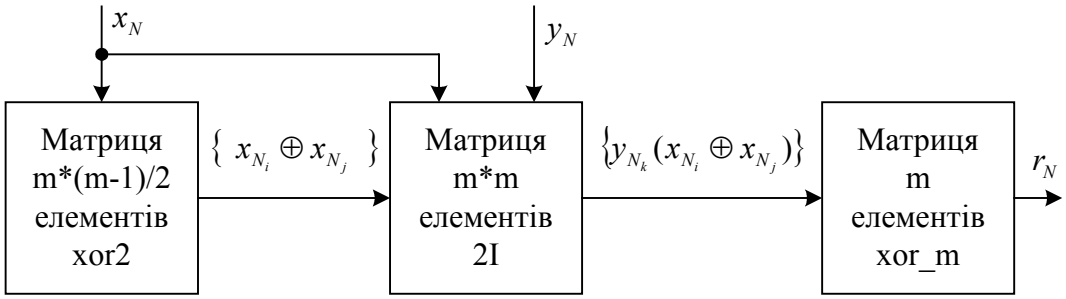


Рис. 6. Одночасне визначення усіх розрядів добутку (варіант 2)

Порівняння апаратних витрат помножувачів у нормальному базисі наведено у табл. 3.

Таблиця 3

Порівняння апаратних витрат

Варіант	Час множення, тактів	Кількість елементів xor2	Кількість елементів 2I	Кількість елементів xor_m	Помножувач
1, 2	1	m*(m-1)	m*m	m	Паралельний
3	1	m*(m-1)/2	m*m	m	Паралельний
	M	m-1	m	1	Послідовний

Відомі проміжні варіанти послідовно-паралельних помножувачів [7].

Сучасні засоби проектування ПЛІС містять вбудовані засоби для мінімізації схем. Тому перелічені вище варіанти 1–3 реалізуються в складі однієї ПЛІС з однаковими і апаратними, і часовими характеристикам, оскільки в результаті мінімізації варіантів 1 та 2 автоматично формується варіант 3.

Варіант 3 дає виграш для великих значень m, коли паралельний помножувач доводиться будувати на базі кількох ПЛІС.

Синтез операційного пристрою для виконання операцій над еліптичними кривими

Правий циклічний зсув (піднесення до квадрата, квадрування) елементів поля Галуа використовують:

для обчислення оберненого елемента x^{-1}, такого, що $x*x^{-1} = 1$;

для додавання точок еліптичної кривої, незалежно від координат (афінних чи проективних), в яких виконується операція над точками.

Обернений елемент використовують:

для додавання та подвоєння точок еліптичної кривої, якщо операції з точками виконують в афінних координатах;

під час переходу від проективних координат до афінних, якщо операції з точками виконують в проективних координатах.

Обчислення у проективних координатах не вимагають знаходження обернених елементів і тому виконуються швидше, ніж в афінних. Але вони вимагають виконання операції знаходження оберненого елемента для подання результату в афінних координатах.

Для обчислення оберненого елемента в оптимальному нормальному базисі використовують формулу: $x^{-1} = x^{2m-2}$, $x \neq 0$. Для обчислення правої частини існує ефективний алгоритм [1]: нехай $m_r, \ldots, m_0$ – двійковий розклад цілого числа m-1. Тоді обернений елемент обчислюють так:

$b \leftarrow x$; $k \leftarrow 1$.

Для i від r-1 до 0 обчислюють: $c \leftarrow b$;

Для j від 1 до k обчислюють: ($c \leftarrow c^2$; $b \leftarrow bc$; $k \leftarrow 2k$);

Якщо $m_j=1$, то $b \leftarrow b^2x$ та $k \leftarrow k+1$.

$x^{-1}=b^2$.

При використанні лівого зсуву обчислювальний пристрій повинен містити разом з помножувачем окремий вузол піднесення до квадрата (квадратор) і додатковий мультиплексор на виході пристрою для виведення результату квадрування назовні.

Інколи квадратор міститься і в структурах з використанням помножувача з правим зсувом [7]. У пристроях для поліноміального базису використання квадратора є практично обов'язковим [8].

Використання помножувача з правим зсувом дає змогу виконувати квадрування на цьому ж помножувачі – на одному з його регістрів циклічного зсуву праворуч. У цьому випадку операційний пристрій помножувача складається з:

логічної помножувальної матриці M;

арифметико-логічного пристрою ALU;

двопортового регістрового файла RGF (рис. 7).

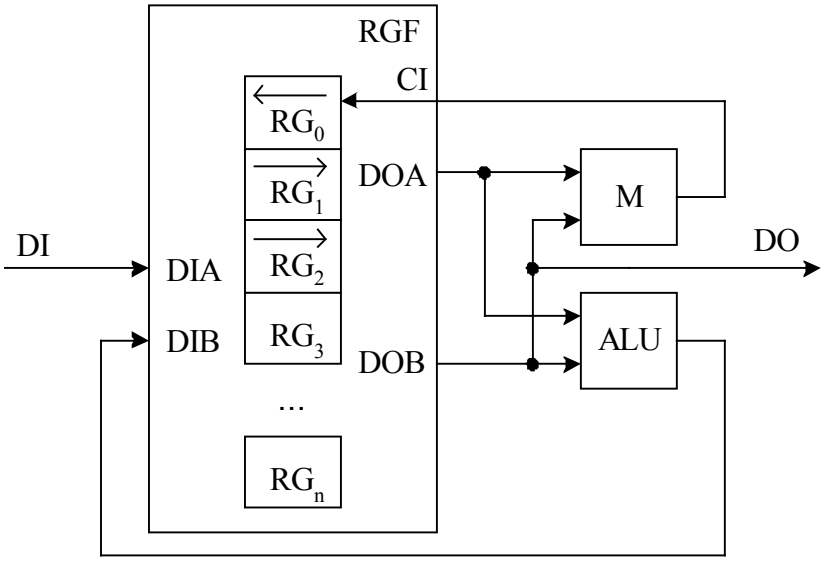


Рис. 7. Пристрій без використання квадратора (нормальний базис)

До складу регістрового файла входять 3 регістри з закріпленими операціями (циклічний зсув праворуч або ліворуч), решта регістрів – регістри загального призначення. Усі регістри можуть приймати дані з двох паралельних входів DIA та DIB, видавати свій вміст на два паралельні виходи DOA та DOB.

Основні операції, які виконує АЛП – додавання за модулем 2, трансляція операндів на вихід без оброблення. Типи операцій, які може виконувати пристрій і тривалість їхнього виконання, наведено у табл. 4.

Таблиця 4

Система команд пристрою

Тип операції	Виконує	Тривалість виконання, тактів	Примітка
a*b	$RG_1*RG_2->RG_0$	m	
a^2b	$RG_1^2*RG_2->RG_0$	m+1	
a^2	$RG_i^2->RG_i$	1	i = {1, 2}
a xor b	RG_i xor $RG_j->RG_k$	1	i,j,k = {0,1, …, n-1}
a	$RG_i->RG_k$	1	i,k = {0,1, …, n-1}

Висновки

У статті проаналізовано стандарти утворення і перевірки цифрового підпису, особливу увагу приділено виконанню операцій множення над елементами поля Галуа, що утворюють нормальний базис. Показано, які операції потрібно виконувати для такого множення; найскладнішою з них є операція утворення оберненої матриці. Запропоновано структури послідовного та паралельного помножувачів, що працюють у нормальному базисі, доведено їхню перевагу.

Також запропоновано структуру операційного пристрою, призначеного для виконання операцій над еліптичними кривими, до складу якого входить послідовний помножувач. Проведено розподіл операцій між його складовими частинами.

1. Національний стандарт України ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що грунтується на еліптичних кривих. Формування та перевіряння. – К.: Державний комітет України з питань технічного регулювання та споживчої політики, 2003. 2. IEEE Std 1363–2000 IEEE Standard Specifications for Public-Key Cryptography Sponsor Microprocessor and Microcomputer Standards Committee of the IEEE Computer Society. Approved 30 January 2000. 3. Карацуба А. и Офман Ю. Умножение многозначных чисел на автоматах // Докл. Академии Наук СССР. – 1962. – Т. 145, 2. – С. 293–294. 4. Montgomery P.L. Modular multiplication without trial division // Mathematics of Computation. – Apr. 1985. – Vol. 44, no. 170. – P. 519–521. 5. Mastrovito E.D. VLSI architectures for multiplication over finite field GF(2m). In T. Mora, editor, Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, 6th International Conference, AAECC-6, Lecture Notes in Computer Science, No. 357, P. 297–309, Rome, Italy, July 1988. New York, NY: Springer-Verlag. 6. U.S. Patent Number 4, 587, 627. Computational method and apparatus for finite fieldarithmetic / J. Omura and J. Massey. – May 1986. 7. FPGA Based Implementation Of An Elliptic Curve Coprocessor Utilizing Synthesizable VHDL code http://www.vlsi.informatik.tu-darmstadt.de/staff/mjung/publications/comprehensive.pdf. 8. Tradeoff analysis of fpga based elliptic curve cryptography / M. Bednara, M. Daldrup, J. Teich J. von zur Gathen, J. Shokrollahi http://www-math.uni-paderborn.de/~aggathen/Publications/beddal02b.pdf .