УДК 681.3.04

І.А. Дичка, В.І. Голуб, М.В. Онай

АПАРАТНА РЕАЛІЗАЦІЯ ПРОЦЕДУР МНОЖЕННЯ І ДІЛЕННЯ БАГАТОЧЛЕНІВ У СКІНЧЕННИХ ПОЛЯХ

In this paper we prove it is necessary to implement hardware or hardware-software\operations in Galois fields. Specifically, we demonstrate that hardware implementation is preferable for multiplication and division operations on polynomials with coefficients that belong to the finite field. It is also feasible to run these operations on separate functional units. We develop formulas that allow skipping summation cycles if the bars being summed contain zero values. Algorithms for evaluating the coefficients as well as the function unit diagram for multiplying and dividing the polynomials in the field GF(N) are provided. We show that the hardware implementation of these operations in finite fields significantly increases computational efficiency.

Вступ

Теорія скінченних полів розроблена в наукових працях видатних математиків – Ферма, Ейлера, Лежандра, Гаусса, Галуа [1–3]. До останньої чверті 20-го ст. вона розвивалась як галузь класичної математики. Але у зв'язку з розвитком завадостійкого кодування та криптографії в наш час активно розвиваються прикладні аспекти теорії [3–5].

Обчислення в скінченних полях (які часто називають полями Галуа) мають свою специфіку, і, з урахуванням подальшого розвитку зазначених галузей, з'являється потреба в удосконаленні структур обчислювальних засобів, які реалізують арифметику скінченних полів. Програмна реалізація обчислень з використанням універсальних комп'ютерних засобів є не завжди ефективною з точки зору швидкодії, зокрема за необхідності обчислень у реальному часі. Тому актуальною є проблема апаратної або апаратно-програмної реалізації обчислень у скінченних полях.

Забезпечення високої ефективності обчислень у скінченних полях можливе лише на основі застосування спеціалізованих обчислювальних засобів. З розвитком інтегральної схемотехніки з'являються нові можливості для реалізації обчислень у полях Галуа з потрібною швидкістю, досягти якої можливо лише за рахунок апаратної реалізації операцій [6, 7].

У різних сферах застосування теорії скінченних полів виникає необхідність виконувати над елементами поля такі операції: додавання та віднімання елементів поля, знаходження адитивно оберненого (протилежного) та мультиплікативно оберненого елементів, множення та ділення елементів поля, піднесення до степеня та обчислення значення багаточлена у заданій точці.

Аналіз класів задач, що мають місце при завадостійкому кодуванні та при криптографічному захисті інформації, показує, що найбільш складними з точки зору часових витрат є операції множення та ділення багаточленів з коефіцієнтами, що належать основному полю. Це означає, що операцію ділення та множення елементів поля слід реалізовувати апаратними, а не програмними засобами.

Постановка задачі

Існуючі способи та алгоритми множення і ділення багаточленів у скінченних полях орієнтовані або на програмну реалізацію зазначених операцій, або на їх апаратну реалізацію на основі традиційних функціональних обчислювальних вузлів.

Сучасною елементною базою обчислювальної техніки є ПЛІС-програмовані логічні інтегральні схеми, які завдяки високому ступеню інтеграції елементів дають змогу реалізовувати на одному напівпровідниковому кристалі складні обчислювальні процедури.

Метою статті є розроблення способів виконання операцій множення та ділення багаточленів у скінченних полях, придатних для їх реалізації на ПЛІС. Це істотно скорочує апаратні витрати на реалізацію операцій та підвищує швидкодію функціональних блоків множення та ділення багаточленів.

Множення багаточленів у скінченних полях

Розглянемо множення багаточлена-множеного $a(x) = \sum_{s=0}^{k-1} a_s x^s$ степеня $k-1$ на багаточлен-множник $g(x) = \sum_{j=0}^{r} g_j x^j$ степеня r. Ре-

зультатом має бути багаточлен-добуток $c(x) = \sum_{i=0}^{n-1} c_i x^i$ степеня $n-1$, де $n = k + r$.

Компоненти векторів коефіцієнтів: $\mathbf{A} = (a_{k-1}, ..., a_0)$ – багаточлена-множеного $a(x)$, $\mathbf{G} = (g_r, ..., g_0)$ – багаточлена-множника $g(x)$ і $\mathbf{C} = (c_{n-1}, ..., c_0)$ – багаточлена-добутку, є елементами поля $\mathrm{GF}(N)$, де N – просте число.

Така операція виникає при завадостійкому кодуванні даних, при цьому отримують кодове слово **C** у несистематичній формі.

Для реалізації множення багаточленів над $\mathrm{GF}(N)$ під час кодування інформації традиційно використовують кола регістрів зсуву, які називають також фільтрами. Такі схеми реалізують множення на фіксований багаточлен $g(x)$.

Але в загальному випадку степінь багаточлена множника (параметр r) може бути різним (може варіюватись). Тому застосовувати фільтри для апаратної реалізації операції множення багаточленів у загальному випадку неможливо.

Отже, необхідний інший підхід до створення схем множення багаточленів з коефіцієнтами з поля $\mathrm{GF}(N)$.

Процедуру множення багаточленів доцільно реалізувати у вигляді окремого функціонального блока, що має інформаційні входи A і G, на які послідовно надходять компоненти векторів **A** і **G** відповідно, інформаційний вихід C, на якому послідовно формуються компоненти вектора **C** – коефіцієнти багаточлена-добутку $c(x)$, а також входи K і R, на які подають параметри k – довжина вектора **A** та r – степінь багаточлена-множника відповідно.

Черговий коефіцієнт c_i при змінній x^i багаточлена-добутку $c(x)$ дорівнює сумі попарних добутків, один із співмножників яких є відповідним коефіцієнтом багаточлена-множеного, а інший – відповідним коефіцієнтом багаточлена-множника (рис. 1).

З метою прискорення виконання операції множення багаточленів за рахунок відкидання тактів підсумовування з нульовими значеннями формування коефіцієнтів c_i багаточлена-добутку здійснюватимемо так (див. рис. 1):

$$c_i = \begin{cases} \sum_{j=0}^{i} a_{i-j} g_j, & 0 \le i < r, \\ \sum_{j=0}^{r} a_{i-j} g_j, & r \le i < k, \\ \sum_{j=i-(k-1)}^{r} a_{i-j} g_j, & k \le i < n, \end{cases}$$

де $i = 0, 1, ..., n-1$.

$n-1$	$n-2$	…	k	$k-1$	…	r	$r-1$	…	1	0
x^{k-1+r}	$x^{k-1+(r-1)}$	…	x^k	x^{k-1}	…	x^r	x^{r-1}	…	x	x^0
$a_{k-1}g_r$	$a_{k-2}g_r$	…	$a_{k-r}g_r$	$a_{k-(r+1)}g_r$	…	$a_0 g_r$	0	…	0	0
0	$a_{k-1}g_{r-1}$	…	$a_{k-r+1}g_{r-1}$	$a_{k-2}g_{r-1}$	…	$a_1 g_{r-1}$	$a_0 g_{r-1}$	…	0	0
0	0		$a_{k-2}g_{r-2}$	$a_{k-(r-1)}g_{r-2}$		$a_2 g_{r-2}$	$a_1 g_{r-2}$	…	0	0
·	·		·	·		·	·		·	·
·	·		·	·		·	·		·	·
·	·		·	·		·	·		·	·
0	0	…	$a_{k-1}g_1$	$a_{k-2}g_1$	…	$a_{r-1}g_1$	$a_{r-2}g_1$	…	$a_0 g_1$	0
0	0	…	0	$a_{k-1}g_0$	…	$a_r g_0$	$a_{r-1}g_0$	…	$a_1 g_1$	$a_0 g_0$
$c_{k+r-1} = a_{k-1}g_r = \sum_{j=r}^{r} a_{k+r-1-j}g_j$	$c_{k+r-2} = \sum_{j=r-1}^{r} a_{k+r-2-j}g_j$	…	$c_k = \sum_{j=1}^{r} a_{k-j}g_j$	$c_{k-1} = \sum_{j=0}^{r} a_{k-1-j}g_j$	…	$c_r = \sum_{j=0}^{r} a_{r-j}g_j$	$c_{k-1} = \sum_{j=0}^{r-1} a_{r-1-j}g_j$	…	$c_1 = \sum_{j=0}^{1} a_{1-j}g_j$	$c_0 = a_0 g_0$
r				k			r			

Рис. 1. Схема обчислення коефіцієнтів c_i багаточлена-добутку $c(x) = a(x)g(x) = \sum_{i=0}^{n-1} c_i x^i$

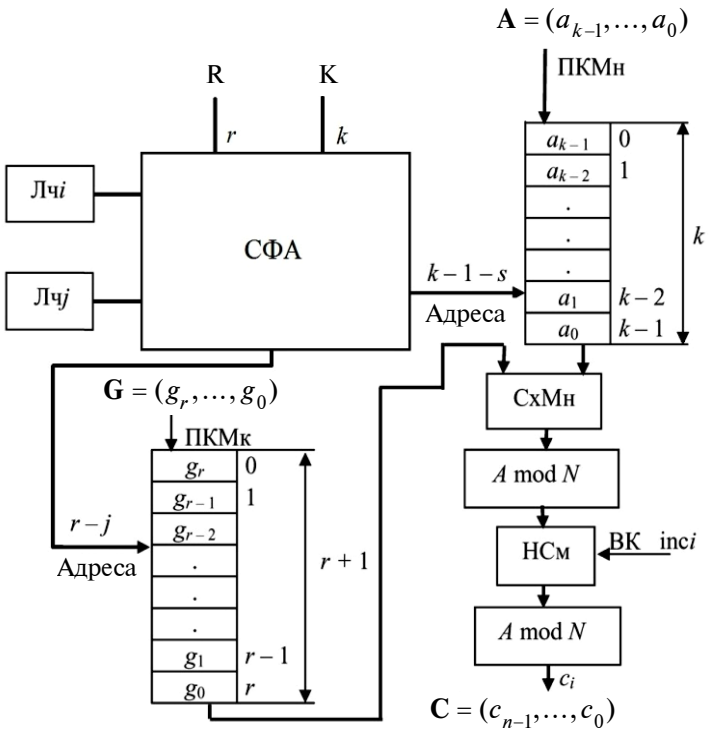

Рис. 2. Схема функціонального блоку для реалізації процедури множення багаточленів у полі $GF(N)$

Блок множення багаточленів містить два лічильники – Лч j, який формує індекс j $(j = 0, 1, \ldots, n)$, та Лч i, який формує індекс i $(i = 0, 1, \ldots, n)$ (рис. 2). Лічильники працюють у режимі інкременту.

За допомогою схеми формування адрес (СФА) формуються три піддіапазони зміни індексу i: $0 \le i < r$, $r \le i < k$, , $k \le i < n$.

Перед початком операції множення коефіцієнти багаточлена $a(x)$ містяться в пам'яті коефіцієнтів множеного (ПКМн), а коефіцієнти багаточлена $g(x)$ – в пам'яті коефіцієнтів множника (ПКМк). Адреси ПКМк $(r - j)$ і ПКМн $(k - 1 - s)$ – в СФА. За відповідною адресою з ПКМк зчитується коефіцієнт g_j, а з ПКМн за відповідною адресою – коефіцієнт a_{i-j}. У схемі множення (СхМн) формується добуток $a_{i-j}g_j$, який за допомогою схеми $A \bmod N$ замінюється лишком за модулем N. У накопичувальному суматорі (НСм) формується значення суми: $\sum_j a_{i-j}g_j$.

При переході до наступного значення i накопичене в НСм значення, що є значенням c_i, надходить на вихід схеми.

Ділення багаточленів у полі GF(N)

Процедура ділення багаточленів у скінченних полях зазвичай застосовується при декодуванні даних, закодованих завадостійким коректувальним кодом.

Особливістю процедури є те, що операції над коефіцієнтами багаточленів (діленого та дільника) виконуються в полі $GF(N)$.

На рис. 3 зображено функціональний блок для реалізації процедури ділення багаточленів (ФБДМ) у полі $GF(N)$, де $\mathbf{B} = (b_{n-1}, \ldots, b_0)$ – вектор коефіцієнтів багаточлена-діленого $b(x) = \sum_{i=0}^{n-1} b_i x^i$; $\mathbf{G} = (g_r, \ldots, g_0)$ – вектор коефіцієнтів багаточлена-дільника $g(x) = \sum_{j=0}^{r} g_j x_j$; $\mathbf{D} = (d_{n-r-1}, \ldots, d_0)$ – вектор коефіцієнтів багаточлена-частки $d(x) = \sum_{s=0}^{n-r-1} d_s x^s$; $\mathbf{Z} = (z_{r-1}, \ldots, z_0)$ – вектор коефіцієнтів багаточлена-остачі $z(x) = \sum_{p=0}^{r-1} z_p x^p$.

Операцію ділення запишемо у загальному випадку:

$$b(x)/g(x) = d(x) + R_{g(x)}[b(x)],$$

де $R_{g(x)}[b(x)]$ – остача від ділення $b(x)$ на $g(x)$.

Позначимо $R_{g(x)}[b(x)] = z(x)$. Остача $z(x)$ є результатом ділення, частка $d(x)$ відкидається.

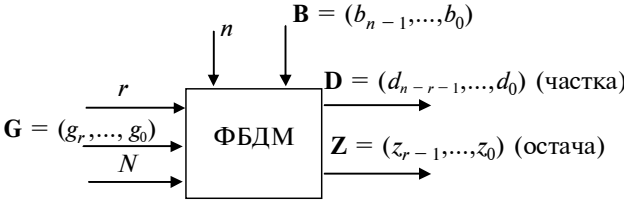

Рис. 3. Функціональний блок для реалізації процедури ділення багаточленів (ФБДМ) у полі $GF(N)$

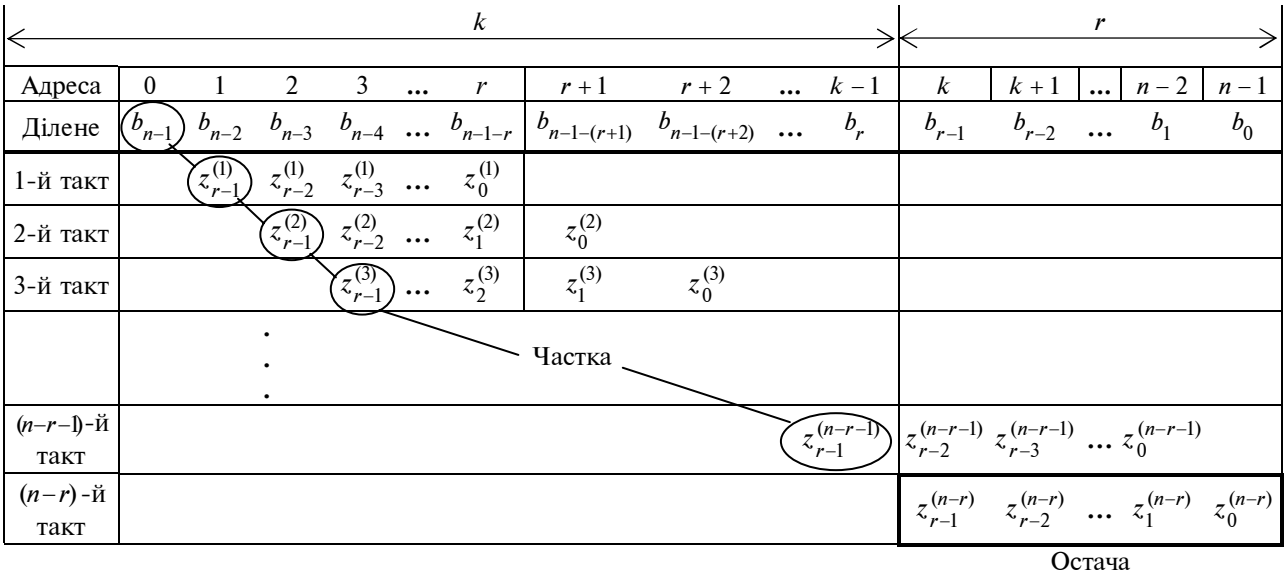

Рис. 4. Схема обчислення частки та остачі під час ділення багаточленів у полі GF(N)

Нехай коефіцієнти b_i багаточлена-діленого $b(x)$ розміщуються в комірках пам'яті діленого (ПДн), а коефіцієнти g_j багаточлена-дільника $g(x)$ – у комірках пам'яті дільника (ПДк).

Обчислювальний процес організуємо в такий спосіб, щоб частка та остача були розміщені в ПДн: перші $k = n - r$ комірок займали коефіцієнти багаточлена-частки, а решту r комірок – коефіцієнти багаточлена-остачі.

Особливістю багаточлена-дільника $g(x)$ є те, що старший коефіцієнт g_r дорівнює одиниці. Це означає, що старший коефіцієнт поточного багаточлена-остачі можна вважати черговим коефіцієнтом багаточлена-частки (рис. 4).

Справді, оскільки $g_r = 1$, то у першому циклі маємо:

ділене:	b_{n-1}	b_{n-2}	b_{n-3}	…	$b_{n-1-(r-1)}$	b_{n-1-r}	$b_{n-1-(r+1)}$	…
–	$b_{n-1}g_r$	$b_{n-1}g_{r-1}$	$b_{n-1}g_{r-2}$	…	$b_{n-1}g_1$	$b_{n-1}g_0$		
остача:	0	$z_{r-1}^{(1)}$	$z_{r-2}^{(1)}$	…	$z_1^{(1)}$	$z_0^{(1)}$		

де $z_{r-j}^{(1)} = b_{n-1-j} - b_{n-1}g_{r-j}$, $j = 1, 2, \ldots, r$.

Отже, b_{n-1} – старший коефіцієнт багаточлена-частки $d(x)$: $d_{n-r-1} = d_{k-1} = b_{n-1}$. Для знаходження $z_{r-j}^{(1)}$ необхідно виконати обчислення $z_{r-j}^{(1)} = [0 + j] - [0] \times g_{r-j}$, де $[0 + j]$ – слово, що зберігається в ПДн за адресою $0 + j$, а $[0]$ – слово, що зберігається в ПДн за адресою 0 (за цією адресою зберігається слово b_{n-1}). Результат $z_{r-j}^{(1)}$ слід записати в ПДн за адресою $0 + j$.

У другому циклі маємо:

ділене:	$z_{r-1}^{(1)}$	$z_{r-2}^{(1)}$	$z_{r-3}^{(1)}$	…	$z_0^{(1)}$	$b_{n-1-(r+1)}$	$b_{n-1-(r+2)}$
–	$z_{r-1}^{(1)}g_r$	$z_{r-1}^{(1)}g_{r-1}$	$z_{r-1}^{(1)}g_{r-2}$	…	$z_{r-1}^{(1)}g_1$	$z_{r-1}^{(1)}g_0$	
остача:	0	$z_{r-1}^{(2)}$	$z_{r-2}^{(2)}$	…	$z_1^{(2)}$	$z_0^{(2)}$	

де $z_{r-j}^{(2)} = [1 + j] - [1]g_{r-j}$, а $[1 + j]$ – слово, що зберігається в ПДн за адресою $1 + j$ (якщо $j = 1$, то за цією адресою в другому циклі зберігається $z_{r-2}^{(1)}$, $[1]$ – слово, що в другому циклі зберігається за адресою 1 (це $z_{r-1}^{(1)}$).

Отже, $z_{r-1}^{(1)}$ – наступний коефіцієнт багаточлена-частки $d(x)$: $d_{k-2} = z_{r-j}^{(1)}$.

Для обчислення коефіцієнтів остачі в l-му циклі необхідно виконати $z_{r-j}^{(l)} = [l - 1 + j] - [l - 1]g_{r-j}$ і записати результат за адресою $l - 1 + j$, отже $[l - 1 + j] := [l - 1 + j] - [l - 1]g_{r-j}$.

В останньому $l = (n - r)$-му циклі отримаємо кінцевий результат – коефіцієнти остачі $z_{r-j}^{(n-r)} = z_{r-j} = [k - 1 + j] - [k - 1]g_{r-j}$ та молодший коефіцієнт багаточлена-частки $d_0 = z_{r-1}^{(n-r-1)}$.

Процес виконання операції ділення багаточленів потребує $n - r = k$ циклів, а в межах кожного циклу – r тактів. Протягом одного

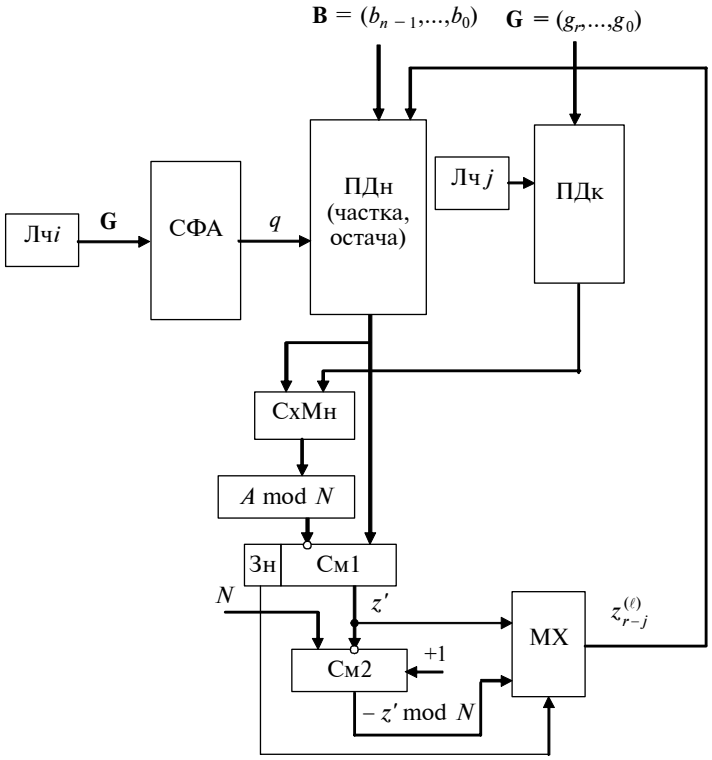

Рис. 5. Схема функціонального блока для реалізації процедури ділення багаточленів у полі GF(N)

такту формується та записується в ПДн одне слово: $z_{r-j}^{(\ell)}$. Тому ФБДМ має у своєму складі два лічильники для роботи з ПДн – лічильник циклів Лчi (рис. 5) та лічильник тактів ЛчТ (на рис. 5 не показаний), який входить до складу схеми формування адрес СФА. Крім того, до складу ФБДМ входить лічильник індекса j (Лчj).

Лчi генерує номери і циклів ($i = 0, 1, 2, \ldots, n - r - 1$); ЛчТ у межах і-го циклу генерує номери q тактів, які є адресами ПДн ($q = i, i + 1, \ldots, i + r - 1$). Початковим станом для Лчi є 0, початковим станом для ЛчТ є i. Інкремент Лчi здійснюється, коли ЛчТ досягає кінцевого номера такту $i + r - 1$.

У кожному такті (за винятком першого такту кожного циклу) з ПДн зчитується та записується слово за адресою q, а з ПДк зчитується слово.

У першому такті кожного нового циклу в ПДн слово не записується, а з ПДк слово не зчитується. Це викликано тим, що $g_r = 1$ (тому немає необхідності з ПДк зчитувати одиницю і множити її на черговий коефіцієнт частки) і немає необхідності виконувати мікрооперацію $z_{r-1}^{(\ell)} - z_{r-1}^{(\ell)} g_r$, результат якої заздалегідь дорівнює нулю.

Для виконання мікрооперації $[l - 1 + j] := [l - 1 + j] - [l - 1] g_{r-j}$ слово з виходу ПДн та слово з виходу ПДк (черговий коефіцієнт багаточлена-дільника) перемножуються у СхМн, внаслідок чого отримуємо $[l - 1] g_{r-j}$. Схема $A \bmod N$ обчислює лишок числа $[l - 1] g_{r-j}$ за модулем N, в См2 виконується мікрооперація $[l - 1 + j] - [l - 1] g_{r-j} = z'$. Якщо $z' \geq 0$, то $z_{r-j}^{(\ell)} = z_1$, інакше $z_{r-j}^{(\ell)} = N - z'$ (ця мікрооперація виконується на См2). Отже, мультиплексор МХ пропускає для запису в пам'ять додатну величину $z' \bmod N$.

Після переходу до наступного циклу Лчj, який формує адреси ПДк, встановлюється в нульове значення.

Результат ділення розміщується в ПДн.

Висновки

Реалізація обчислень у скінченних полях може бути ефективною лише за умови створення для цієї мети спеціалізованих обчислювальних засобів. Для досягнення потрібної швидкодії та забезпечення роботи системи в реальному часі реалізовувати найбільш складні процедури та функції арифметики скінченних полів слід апаратними засобами. До таких складних процедур належить множення та ділення багаточленів у полях Галуа. Кожна з цих процедур має бути реалізована у вигляді окремого функціонального блока. Для цієї мети доцільно використовувати ПЛІС. Реалізація на ПЛІС найбільш складних з точки зору витрат часу процедур арифметики скінченних полів дає можливість у 8–10 разів пришвидшити виконання операцій порівняно з їх програмною реалізацією з використанням універсальних обчислювальних засобів.

Функціональні блоки для множення та ділення багаточленів у скінченних полях можуть

бути складовими частинами спеціалізованої обчислювальної системи, що реалізує арифметику скінченних полів.

Такий підхід забезпечує істотне підвищення ефективності обчислень у скінченних полях.

Подальшого дослідження потребує питання ефективної реалізації обчислення значення багаточлена в заданій точці в арифметиці скінченних полів.

1. *V. Patel and K.S. Gurumurthy*, "Arithmetic operations in Multi-Valued logic", VLSICS, vol. 1, no. 1, pp. 21–32, 2010.
2. *P. Kisos et al.*, "An efficient reconfigurable multiplier architecture for Galois field GF(2m)", Microelectronics J., vol. 34, 975–980, 2003.
3. *Ch.-Yng Lee and P.K. Meher*, "Efficient bit-parallel multipliers over finite fields GF(2m)", Comput. and Electrical Eng., vol. 36, pp. 955–968, 2010.
4. *L. Batina et al.*, "Hardware architectures for public key cryptography", Integration, The VLSI J., vol. 34, 2003, pp. 1–64.
5. *M. Morales-Sandoval et al.*, "An area/performance tradeoff analysis of a GF(2m) multiplier architecture for elliptic curve cryptography", Comput. and Electrical Eng., vol. 35, pp. 54–58, 2009.
6. *S. Serdar et al.*, "Erdem Polynomial Basis Multiplication over GF(2m)", Acta Appl. Math., pp. 33–55, 2006.
7. *H. Wu*, "Bit-parallel finite field multiplier and squarer using polynomial basis", IEEE Trans. Comput., vol. 51, no. 7, pp. 750–758, 2002.

Рекомендована Радою
факультету прикладної математики
НТУУ "КПІ"

Надійшла до редакції
4 червня 2012 року