3. Наступним етапом налаштовану і верифіковану засобами MatLab модель можна автоматично імплементувати до цільової ПЛІС засобами синтезатора Xilinx AccelDSP та інтегрованої з ним САПР Xilinx ISE. Отож, розроблення апаратних засобів адаптивного еквалайзінга ефективно автоматизується, що прискорює процес і покращує отримані технічні характеристики розробки.

1.Adaptive Equalization System for Data Transmission over Coaxial Cables Jasmine Sai-Ying Cheng A Thesis submitted in conformity with the requirements Department of Electricai and Computer Engineering University of Toronto 1998 Canada National Library Bibliographic Services services bibliographiques, 395 Wellington Street 395. rue Wellington OttawaON KIA ON4. 138 p.2. Haykin, S., Adaptive Filter Theory, Third Ed., Prentice Hall, 1996. 989 p.3. Communications Toolbox User's Guide. The MathWorks, Inc. 2006. 824p.4. AccelDSP Synthesis Tool. User Guide. Release 9.1.01. Xilinx Corp. March, 2007. 228 p. 5. DSP: Designing for Optimal Results High-Performance DSP Using Virtex-4 FPGAs. Xilinx, 2005, 116 p.

УДК 004.383

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

ВДОСКОНАЛЕННЯ АЛГОРИТМУ ОБЧИСЛЕННЯ ОБЕРНЕНОГО ЕЛЕМЕНТА $GF(2^T)$ В НОРМАЛЬОМУ БАЗИСІ

Описано вдосконалення методу Ito–Тічей–Цудзії (Itoh, Teechai, and Tsujii) знаходження оберненого елемента поля Галуа GF(2^m) в оптимальному нормальному базисі. Вдосконалення полягає у зменшенні часу виконання послідовності операцій піднесення до квадрата.

The paper describes Itoh, Teechai, and Tsujii method of GF(2^m) inverse element calculation improvement in optimal normal base. The improvement minimizes the number of squaring.

Вступ. Сучасні стандарти для роботи з цифровими підписами [1, 2] ґрунтуються на використанні полів Галуа та еліптичних кривих.

Елементи $\left\{\theta, \theta^2, \theta^{2^2}, ..., \theta^{2^{m-1}}\right\}$ основного поля Галуа GF(2^m) утворюють нормальний базис (θ – корені полінома p, що утворює поле). Усі інші елементи основного поля Галуа GF(2^m) можуть бути представлені 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*).

Для обчислення оберненого елемента в оптимальному нормальному базисі використовується алгоритм Ito–Тічей–Цудзії (Itoh, Teechai, and Tsujii) [3].

Недоліком алгоритму є велика кількість операцій піднесення до квадрата. У нормальному базисі піднесення до квадрата виконується як циклічний зсув елемента на один двійковий розряд праворуч. У роботі пропонується використовувати зсуви на багато розрядів. Також наведений приклад визначення кількості розрядів, на які треба здійснювати багаторозрядні зсуви для поля Галуа GF(2^{173}).

Аналіз публікацій і окреслення проблеми. Для обчислення оберненого елемента в оптимальному нормальному базисі використовується формула: $x^{-1} = x^{2^m-2} = x^{2(2^{m-1}-1)}$, x≠0. Для обчислення $x^{2^m-2} = x^{2(2^{m-1}-1)}$ існує ефективний алгоритм Ito–Тічей–Цудзії (Itoh, Teechai, and Tsujii) [3]:

нехай m_r, ..., m_0 – двійковий розклад цілого числа m-1. Тоді обчислення оберненого елемента виконують так:

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

(2) Для і від r-1 до 0 обчислюють:

(2.1) $c \leftarrow b$;

(2.2) для j від 1 до k обчислюють $c \leftarrow c^2$;

(2.3) $b \leftarrow bc$;

(2.4) $k \leftarrow 2k$;

(2.5) якщо m_i=1, то $b \leftarrow b^2x$ та $k \leftarrow k$+1.

(3) $x^{-1}=b^2$.

Цей алгоритм застосовують при реалізації криптографічних пристроїв [4, 5], що виконують перетворення елементів поля Галуа [6] і точок еліптичних кривих [7] під час виконання операцій над цифровими підписами відповідно до стандартів [1, 2].

Недоліком алгоритму є велика кількість операцій $c \leftarrow c^2$ на етапі 2.2, яка зростає із зростанням r. У нормальному базисі піднесення до квадрата виконується як циклічний зсув елемента на один двійковий розряд праворуч. У роботі пропонується використовувати зсуви на багато розрядів. Також наведений приклад визначення кількості розрядів, на які треба здійснювати багаторозрядні зсуви для поля Галуа $GF(2^{173})$.

Мета роботи. Метою роботи є зменшення часу обчислення оберненого елемента поля Галуа $GF(2^m)$ із використанням методу Іто–Тічей–Цудзії. Цей алгоритм зменшує кількість операцій множення, але кількість операцій піднесення до квадрата (на кроці 2.2) залишається великою, час їхнього виконання приблизно дорівнює часу виконання ще одного множення.

У роботі ставиться задача зменшення часу виконання операцій піднесення до квадрата на кроці 2.2.

Модифікація методу Іто–Тічей–Цудзії. Оригінальний алгоритм передбачає використання регістра з двовходовим мультиплексором на вході для виконання занесення початкового значення до регістру і подальшого його циклічного зсуву на один розряд за кожний такт (рис. 2).

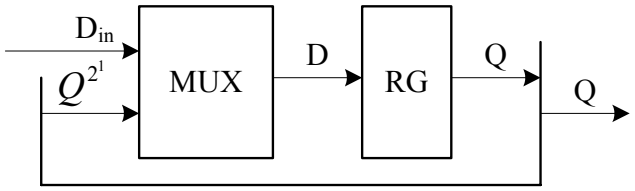

Рис. 1. Функціональна схема регістра зсуву на багато розрядів

В основу модернізації алгоритму покладене використання регістрів зсуву з багатовходовим мультиплексором на вході, що дає змогу виконувати зсув з програмованою величиною зсуву за один такт ([8], рис. 2).

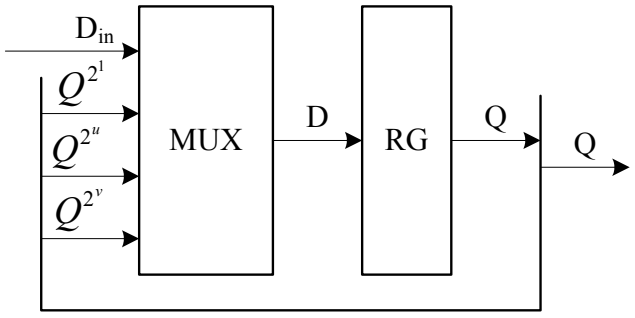

Рис. 2. Функціональна схема регістра зсуву на багато розрядів

На рис. 2 позначено:

D_{in} – дані для початкового завантаження регістра зсуву;

Q^{2^j} – вихід Q регістра, циклічно зсунутий праворуч на j двійкових розрядів.

Сигнали керування на рис. 1 та рис. 2 не позначені.

Циклічний зсув праворуч за один такт на *j* двійкових розрядів елемента

$$\alpha = (a_0, a_1, ..., a_{m-j-1}, a_{m-j}, a_{m-j+1}, ..., a_{m-1}, a_0, a_1, ..., a_{m-j-1})$$

призводить до піднесення його до степеня 2^j:

$$\alpha^{2^j} = (a_{m-j}, a_{m-j+1}, ..., a_{m-1}, a_0, a_1, ..., a_{m-j-1}).$$

У табл. 1 показані апаратні витрати на реалізацію багатовходових однорозрядних мультиплексорів на прикладі ПЛІС ф. Xilinx. Як видно, найкраще співвідношення апаратних витрат на один вхід мають мультиплексори з кількістю інформаційних входів 2, 4, 8 (загалом 2^n).

Таблиця 1

Апаратні витрати на реалізацію мультиплексорів на ПЛІС

Входів	2	3	4	5	6	7	8
LUT (Xilinx, Virtex)	8	16	16	25	32	40	32
LUT/вхід	4	5,3	4	5	5,3	5,6	4

Алгоритм Іто–Тічей–Цудзії передбачає використання двовходового мультиплексора. Модифікація алгоритму полягає в використанні 4- або 8-входових мульитплексорів.

Використання 4-входового мультиплексора дає змогу завантажити початкове значення у регістр зсуву, виконувати зсуви на 1, *u* та *v* двійкових розрядів (за один такт). Для різних поліномів значення *u* та *v* знаходяться методом перебору.

Нижче наведений приклад для $m=173_{10}$. Тоді $m\text{-}1 = 172_{10} = AC_{16} = 10101100_2$.

Під час обчислення оберненого елемента зміна значень *k* (що дорівнює кількості однорозрядних зсувів на етапі 2.2 алгоритму) відбувається згідно з табл. 2. Загальна кількість зсувів дорівнює:

$$k(6)+ k(5)+…+ k(0)=168.$$

Таблиця 2

Кількість однорозрядних зсувів

i	7	6	5	4	3	2	1	0
m_i	1	0	1	0	1	1	0	0
k(i)		1	2	5	10	21	43	86

Очевидно, що значення *u* та *v*, які визначають, на скільки розрядів повинен виконуватися багаторозрядний зсув, повинні дорівнювати одному із значень *k* (різному для *u* та *v*). Найкращі результати дає варіант *u*=43, *v*=5, що ілюструє табл. 3. Загальна кількість багаторозрядних зсувів (тобто, загальна кількість тактів виконання етапу 2.2 алгоритму) дорівнює $S_u+S_v+S_1=14$. Отже, тривалість виконання етапу 2.2 алгоритму зменшується з 168 до 14, тобто, приблизно на порядок.

Для цього прикладу можливе використання 8-входового мультиплексора, який забезпечуватиме зсув на 1, 2, 5, 10, 21, 43 та 86 розрядів (при цьому один вхід мультиплексора використовуватимуть для занесення початкового значення). Але, як видно з табл. 1, такий мультиплексор вимагає удвічі більших апаратних витрат порівняно з 4-входовим мультиплексором.

Пункт 2.2. модифікованого алгоритму для 4-входового мультиплексора можна записати у такому вигляді:

(2.2)

(2.2.1) $S_u(i)$ разів обчислити $c \leftarrow c^{2^u}$ (тобто, за один *i*-й такт виконати циклічний зсув на *u* двійкових розрядів, переславши інформацію через 3-й вхід мультиплексора MUX);

(2.2.2) $S_v(i)$ разів обчислити $c \leftarrow c^{2^v}$ (тобто, за один *i*-й такт виконати циклічний зсув на *v* двійкових розрядів, переславши інформацію через 2-й вхід мультиплексора MUX);

(2.2.3) $S_1(i)$ разів обчислити $c \leftarrow c^{2^1}$ (тобто, за один *i*-й такт виконати циклічний зсув на 1 двійковий розряд, переславши інформацію через 1-й вхід мультиплексора MUX).

Значення $S_u(i)$, $S_v(i)$ та $S_1(i)$ попередньо обчислюються та зберігаються у табл. 3 разом з номерами *i* відповідних входів мультиплексора.

Структуру додаткового ПЗП, де зберігається табл. 3, визначають за табл. 4.

Таблиця 3

Кількість багаторозрядних зсувів

i	*k(i)*	$S_u(i)$ – кількість зсувів на 43 розряди (через 3-й вхід MUX)	$S_v(i)$ – кількість зсувів на 5 розрядів (через 2-й вхід MUX)	$S_1(i)$ – кількість зсувів на 1 розряд (через 1-й вхід MUX)
6	1	0	0	1
5	2	0	0	2
4	5	0	1	0
3	10	0	2	0
2	21	0	4	1
1	43	1	0	0
0	86	2	0	0
	Разом зсувів: 14	3	7	4

Пункт 2.2. модифікованого алгоритму для 8-входового мультиплексора можна записати у такому вигляді:

(2.2) обчислити $c \leftarrow c^{2^{k(i)}}$ (тобто, за один *i*-й такт виконати циклічний зсув на *k(i)* двійкових розрядів, переславши інформацію через *i*-й вхід мультиплексора MUX).

Значення *k(i)* попередньо обчислюються та забезпечують поданням відповідних сигналів на входи мультиплексора. Керує мультиплексором безпосередньо номер такту *i*.

Таблиця 4

Структура ПЗП для збереження табл. 3 у випадку 4-входового мультиплексора

	Діапазон значень	Розрядність, біт	Примітка
Адреса *i*	0…6	3	*p*
Дані $S_u(i)$	0…2	2	
Дані $S_v(i)$	0…4	3	
Дані $S_1(i)$	0..2	2	
Разом даних		7	*q*
Організація ПЗП		8x7=56	$2^p \cdot q$

Функціональна схема пристрою, що реалізує етап 2.2 вдосконаленого алгоритму (для випадку використання 8-розрядного мультиплексора) відповідно до табл. 3, наведена на рис. 3.

Результати порівняння оригінального і вдосконаленого алгоритму наведені у табл. 5.

Як видно, вдосконалений алгоритм забезпечує значний виграш як у часі виконання зсувів, так і при оцінці за комплексним показником, який враховує і часові, і апаратні витрати.

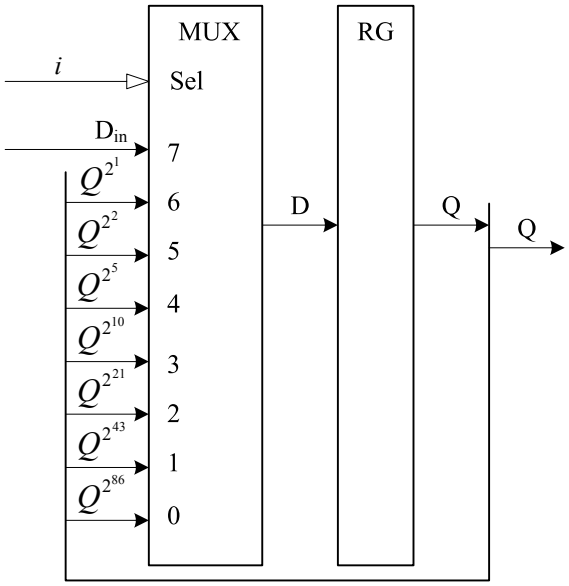

Рис. 3. Функціональна схема пристрою

Аналіз часу виконання алгоритму

Цікаво порівняти кількість тактів виконання зсувів (168, 14 або 7 табл. 5) з кількістю тактів виконання операцій множення під час виконання алгоритму Іто–Тічей–Цудзії.

Кількість *n* операцій множення з використанням немодифікованого алгоритму Іто–Тічей–Цудзії не залежить від елемента, для якого обраховується обернене значення, і дорівнює зменшеній на 1 сумі кількості біт *k* у записі числа *m*-1 *(k=[log(m-1)*]) плюс кількість *e* ненулевих біт (функція *w*) у цьому записі (*e=w(m-1)*).

Таблиця 5

Порівняння оригінального і вдосконаленого алгоритму

	2-входовий мультиплексор	4-входовий мультиплексор	8-входовий мультиплексор
l – Апаратні витрати (кількість LUT)	8	16	32
s – Кількість тактів зсуву	168	14	7
l·s – Комплексний показник, враховує апаратні витрати і кількість тактів зсуву	1344	224	224

Кількість тактів однорозрядних зсувів з використанням немодифікованого алгоритму Іто–Тічей–Цудзії не залежить від елемента, для якого обраховується обернене значення, і дорівнює різниці *m-e*-1.

Кількість тактів багаторозрядних зсувів з використанням модифікованого алгоритму Іто–Тічей–Цудзії і вісьмивходового мультиплексора не залежить від елемента, для якого обраховується обернене значення, і дорівнює *k*.

Для допустимих основних полів з оптимальним нормальним базисом [1] кількість операцій множення *n,* кількість тактів множення $N_m = n*m$ і кількість тактів однорозрядних зсувів $N_s = m-e-1$ та багаторозрядних зсувів *k* наведена у табл. 6. Також наведена частка $N_s/(N_s+N_m)$ тактів зсуву серед загальної кількості тактів виконання алгоритму (%).

Аналізуючи табл. 6, можна зробити висновки:

кількість тактів однорозрядних зсувів не більша за 10 % від кількості тактів множення;

зменшення кількості тактів зсуву зменшує загальну кількість тактів виконання алгоритму не більше ніж на 10 %;

серед основних полів з оптимальним нормальним базисом виділяються декілька з найменшою кількістю множень (10) – це поля зі степенями поліномів m = 173, 179, 281, 293;

якщо кращим вважати поле, в якому для обчислення обернених елементів треба витрачати меншу кількість тактів ніж хоча б в одному полі з меншим порядковим номером (з меншим за m), то найкращими є поля зі степенями поліномів m=281 та m=293 (7-е та 8-е поля, табл. 6 та рис. 4).

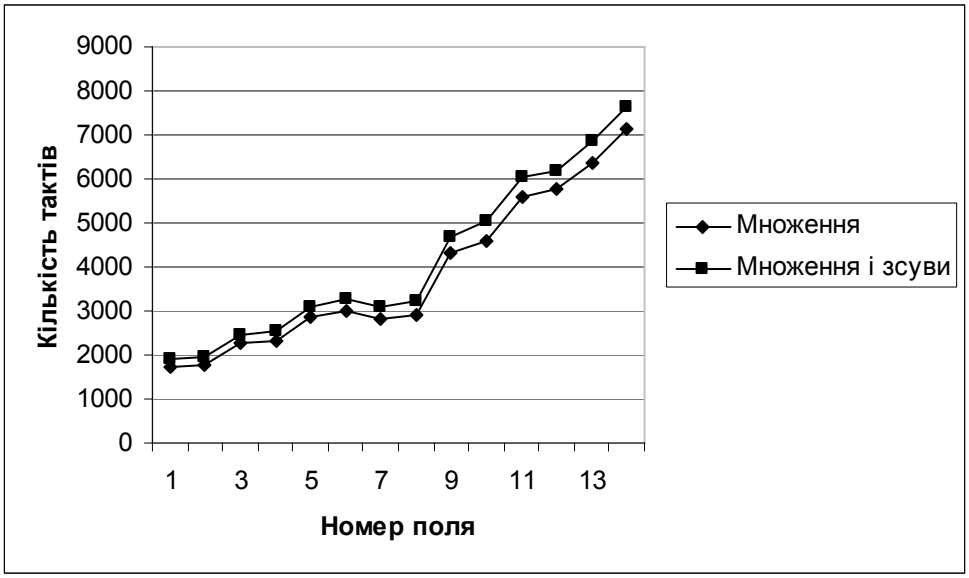

Рис. 4. Кількість тактів множення і зсуву під час обчислення оберненого елемента у нормальному базисі (по осі X – номери полів з табл. 6)

Таблиця 6

Кількість множень під час обчислення оберненого елемента у нормальному базисі

№	m	$(m\text{-}1)_{10}$	$(m\text{-}1)_2$	k= =[log(m-1)]	e= =w(m-1)-1	$n=k+e-1$	$N_m= n*m$	$N_s=m-e-1$	$N_s/(N_s+N_m)$, %
1	173	172	10101100	7	4	10	1730	168	8,9
2	179	178	10110010	7	4	10	1790	174	8,9
3	191	190	10111110	7	6	12	2292	184	7,4
4	233	232	11101000	7	4	10	2330	228	8,9
5	239	238	11101110	7	6	12	2868	232	7,5
6	251	250	11111010	7	6	12	3012	244	7,5
7	281	280	100011000	8	3	10	2810	277	9
8	293	292	100100100	8	3	10	2930	289	9
9	359	358	101100110	8	5	12	4308	353	7,6
10	419	418	110100010	8	4	11	4609	414	8,2
11	431	430	110101110	8	6	13	5603	424	7
12	443	442	110111010	8	6	13	5759	436	7
13	491	490	111101010	8	6	13	6383	484	7
14	509	508	111111100	8	7	14	7126	501	6,6

Висновки. У роботі описане вдосконалення алгоритму Ito–Тічей–Цудзії (Itoh, Teechai, and Tsujii) знаходження оберненого елемента поля Галуа $GF(2^m)$ в оптимальному нормальному базисі. Вдосконалення полягає у зменшенні часу виконання послідовності операцій піднесення до квадрата

(послідовності операцій циклічного зсуву праворуч на один двійковий розряд). Використання вузлів циклічного зсуву на багато розрядів дає змогу скоротити час виконання послідовності зсувів приблизно на порядок, а час виконання алгоритму загалом приблизно на 7–9 %.

З метою зменшення тривалості обчислень рекомендується під час роботи відповідно до стандарту [1] у нормальному базисі використовувати поля зі степенями поліномів m=281 та m=293

1. Національний стандарт України ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується 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. Itoh T., Teechai O., Tsujii S. A Fast Algorithm for Computing Multiplicative Inverses in GF(2t) Using Normal Bases // J. Society for Electronic Communications. – Japan, 1986. – 44. – P. 31–36. 4. Глухов В., Заіченко Н., Оліярник Б. Шифропроцесор для бортових інформаційно-керуючих систем // Наукові нотатки: Міжвуз. зб. (за напрямом "Інженерна механіка"). – 2007. – Вип. 19. Луцький державний технічний університет. – Луцьк, 2007. – С. 33–43. 5. Глухов В.С., Євтушенко К.С., Заіченко Н.В., Оліярник Б.О. Криптографічні засоби спеціалізованої бортової ЕОМ для бронетехніки // Вісн. Хмельницьк. анц. ун-ту. – 2007. – № 2. Технічні науки. – Хмельницький, 2007. – Т. 2. – С. 29–33. 6. Глухов В.С. Операційний пристрій для роботи з елементами поля Галуа, представленими у нормальній формі // Матеріали наук.-техн. конф. ІППТ при Нац. ун-ту "Львівська політехніка". – Львів, 2007. 7. Глухов В.С. Обчислювальний пристрій для операцій над еліптичними кривими // Вісн. Нац. ун-ту "Львівська політехніка". – 2006. – № 573. – С. 54–61. 8. Hlukhov V. Improvement of Algorithm for Computing Multiplicative Inverses in GF(2^m) Using Normal Bases. Матеріали конференції ACSN'2007. – Львів, 2007.

УДК 681.3, 621.3

В.А. Голембо, О.Ю. Бочкарьов, Х.Р. Попадюк
Національний університет "Львівська політехніка",
кафедра електронних обчислювальних машин

ПРОБЛЕМА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ КОЛЕКТИВНОЇ ПОВЕДІНКИ АВТОНОМНИХ МОБІЛЬНИХ АГЕНТІВ В ЗАДАЧАХ ПРОСТОРОВОЇ САМООРГАНІЗАЦІЇ

Розглянуто підходи до розроблення алгоритмічного забезпечення колективної поведінки автономних мобільних агентів в задачах просторової самоорганізації на основі аналізу особливостей цих задач та різних варіантів комплектації робототехнічної платформи агента.

The approaches to the development of algorithms for collective behaviour of autonomous mobile agents in the tasks of spatial self-organization based on analysis of features of these tasks and various settings of agent's robotic platform are considered.

Вступ. Розглядається актуальна проблема розроблення алгоритмічного забезпечення багатоагентних систем (колективів агентів) в задачах просторової самоорганізації [1, 2]. Зокрема, обговорюється необхідність та можливість створення відповідного набору алгоритмів, який можна використовувати незалежно від типу мобільної робототехнічної платформи агента. Це, насамперед, зумовлено певною універсальністю задач просторової самоорганізації, вирішення яких потрібне для організації цілеспрямованої поведінки переважної більшості систем розподіленої робототехніки