МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

М. М. Касянчук

ДОСКОНАЛА ФОРМА СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ: МЕТОДИ ПОБУДОВИ ТА ЗАСТОСУВАННЯ

Монографія

Тернопіль
ТНЕУ
2019

УДК 519.7:004
К 28

Рекомендовано до друку
вченою радою Тернопільського національного економічного університету
(протокол №7 від 27.03.2019 р.)

Рецензенти:

Максимович Володимир Миколайович – доктор технічних наук, професор, завідувач кафедри безпеки інформаційних технологій Національного університету «Львівська політехніка»;

М'ясіщев Олександр Анатолійович – доктор технічних наук, професор, завідувач кафедри кібербезпеки та комп'ютерних систем і мереж Хмельницького національного університету;

Грод Іван Миколайович – доктор фізико-математичних наук, доцент, доцент кафедри кібербезпеки Тернопільського національного технічного університету імені Івана Пулюя

Касянчук М. М.
К 28 **Досконала форма системи залишкових класів: методи побудови та застосування :** моногр. – Тернопіль : ТНЕУ, 2019. – 224 с.
ISBN 978-966-654-534-6

У монографії викладено теоретичні основи побудови та програмної реалізації досконалої та модифікованої досконалої форм системи залишкових класів. Уперше зроблена спроба обґрунтувати використання вказаних форм в асиметричних криптосистемах, зокрема трьохмодульній криптосистемі Рабіна, при виконанні операцій модулярного множення та модулярного експоненціювання, побудові розподіленого термо- або п'єзоелектричного сенсора. Особливу увагу приділено методам розробки модифікованої досконалої форми системи залишкових класів на основі перемноження модулів, факторизації, теореми Вієта, розв'язку систем конгруенцій, послідовності Фібоначчі. Наведено приклади використання розроблених методів.

Монографія призначена для науковців та фахівців у галузі знань «Інформаційні технології», може бути корисна для аспірантів, студентів та магістрантів, які навчаються за спеціальностями «Кібербезпека», «Комп'ютерна інженерія», «Комп'ютерні науки», «Інженерія програмного забезпечення», «Інформаційні системи та технології», «Автоматизація та комп'ютерно-інтегровані технології».

ISBN 978-966-654-534-6

УДК 519.7:004

ЗМІСТ

ПЕРЕДМОВА

На сучасному етапі розвитку людської цивілізації практично неможливо уявити будь-яку галузь народного господарства без використання комп'ютерних систем, які виконують функції обробки, збереження, передавання та захисту інформації. Їхньому поширенню сприяє швидкий розвиток відповідних програмно-апаратних засобів. З огляду на це очевидною є актуальність досліджень, присвячених новим принципам побудови обчислювальних систем, раціональним методам організації їхньої роботи, пошукам ефективних шляхів використання і, відповідно, захисту інформаційних потоків при їхньому опрацюванні.

При розробці структури обчислювальної системи одним із найважливіших питань є вибір представлення числової інформації, тобто вибір її кодування або відповідної системи числення. Найбільш поширені нині позиційні системи числення (зокрема, двійкова та десяткова) володіють істотними недоліками: велика розрядність, строга послідовність виконання арифметичних операцій (відсутність можливості розпаралелення), наявність міжрозрядних переносів, які ускладнюють способи реалізації арифметичних операцій, відповідне апаратне забезпечення і призводять до зменшення швидкодії.

Початком досліджень для використання непозиційних систем числення (зокрема, системи залишкових класів) при виконанні арифметичних операцій стали опубліковані в 1955–1957 рр. праці чеських учених М. Валаха і А. Свободи, присвячені

представленню чисел у вигляді сукупності невід'ємних залишків від їхнього ділення на натуральні попарно взаємно прості числа, які називаються модулями. Це дозволило уникнути багатьох труднощів при виконанні арифметичних операцій у позиційних системах числення і привело до створення І. Акушським та Д. Юдіцьким відповідної машинної арифметики в залишкових класах. Істотною перевагою непозиційної системи залишкових класів є можливість розпаралелення процесу обчислень, виконання арифметичних операцій над малорозрядними залишками та відсутність міжрозрядних переносів.

Недоліки системи залишкових класів (визначення знака числа, переповнення розрядної сітки, дробові перетворення, операції ділення та порівняння чисел) знівелювалися з настанням у 1977 р. ери асиметричної криптографії, в якій переважно використовуються цілочисельні модулярні арифметичні операції додавання, множення та експоненціювання.

Запропонована професором Я. М. Николайчуком досконала форма системи залишкових класів є значним внеском у розвиток цієї непозиційної системи, оскільки труднощі при відновленні десяткового числа із його залишків були ще одним недоліком, який стримував розвиток системи залишкових класів. Відповідний підбір модулів дозволяє зменшити часову складність при переведенні чисел із системи залишкових класів у позиційну систему числення шляхом уникнення виконання обчислювально складної операції пошуку мультиплікативного оберненого елемента за модулем і множення на нього.

Однак аналіз вітчизняних та зарубіжних наукових джерел свідчить, що в Україні та світовому просторі практично відсутній узагальнений підхід до проблеми розробки методів побудови

досконалої та модифікованої досконалої форм системи залишкових класів. Вирішення окресленої проблеми є метою цієї монографії.

Автор складає подяку ректору Тернопільського національного економічного університету, доктору економічних наук, професору А. Крисоватому та проректору з наукової роботи Тернопільського національного економічного університету, доктору економічних наук, професору З.-М. Задорожному за фінансову підтримку видання даної монографії; рецензентам доктору технічних наук, професору, завідувачу кафедри безпеки інформаційних технологій Національного університету «Львівська політехніка» В. Максимовичу, доктору технічних наук, професору, завідувачу кафедри кібербезпеки та комп'ютерних систем і мереж Хмельницького національного університету О. М'ясіщеву, доктору фізико-математичних наук, доценту, доценту кафедри кібербезпеки Тернопільського національного технічного університету імені Івана Пулюя І.Гроду за плідну наукову дискусію та цінні зауваження; доктору технічних наук, професору, академіку Міжнародної академії інформатики, завідувачу кафедри спеціалізованих комп'ютерних систем Тернопільського національного економічного університету Я. Николайчуку, доктору технічних наук, професору, завідувачу кафедри інформатики та автоматики Технічно-гуманістичної академії в Бєльсько-Бялій (Польща) М. Карпінському, доктору технічних наук, професору, декану факультету комп'ютерних інформаційних технологій Тернопільського національного економічного університету, М. Диваку, доктору технічних наук, доценту, завідувачу кафедри кібербезпеки Тернопільського національного економічного університету В. Яцківу, колективу

кафедри кібербезпеки Тернопільського національного економічного університету, особливо кандидатам технічних наук, доцентам І. Якименку та С. Івасьєву, за цінні поради та консультації, надані у процесі підготовки монографічного дослідження; дружині Галині, донькам Надії та Христині, своїм батькам Євгенії Григорівні та Миколі Прокоповичу, батькам дружини Оксані Олексіївні та Івану Макаровичу за моральну підтримку.

З повагою, автор М. М. Касянчук

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ, СУЧАСНИЙ СТАН ТА НАПРЯМОК ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ЇЇ ВИКОРИСТАННЯ

1.1 Теоретичні основи алгебри і теорії чисел

Фундаментальною теоретичною основою сучасних асиметричних криптосистем [1–6] є алгебра і теорія чисел, або теорія лишків чи конгруенцій [7–12]. Цілі числа *a* та *b* називаються порівнянними (або конгруентними) за модулем *z*, якщо різниця чисел *a* – *b* націло ділиться на *z.* Співвідношення між *a, b* і *z* запишемо у вигляді:

$$a \equiv b(\mathrm{mod}\, z). \qquad (1.1)$$

Запис mod z буде означати, що $z \geq 1$, числа *a* і *b* – залишки. Запис (1.1) називається конгруенцією.

Відповідно до визначення, запис $a \equiv 0(\mathrm{mod}\, z)$ означає, що *a* націло ділиться на *z.*

Приклад:

$101 \equiv 17(\mathrm{mod}\, 21)$, тому що 101 – 17 = 84, а $84 \vdots 21$ або $135 \equiv 11(\mathrm{mod}\, 4)$, бо при діленні чисел 135 та 11 на 4 є остача 3.

Число *a* конгруентне числу *b* тоді й тільки тоді, коли *a* і *b* мають однакові залишки при діленні на *z*, тому як визначення конгруенцій можна взяти таке: цілі числа *a* і *b* називаються конгруентними за модулем *z*, якщо залишки від ділення цих чисел на *z* рівні.

Основні властивості конгруенцій:

1) рефлексивність: $a \equiv a(\mathrm{mod}\, z)$;

2) симетричність: якщо $a \equiv b(\mathrm{mod}\, z)$, то $b \equiv a(\mathrm{mod}\, z)$;

3) транзитивність: якщо $a \equiv b(\mathrm{mod}\, z)$, $b \equiv c(\mathrm{mod}\, z)$, то $a \equiv c(\mathrm{mod}\, z)$;

4) якщо $a \equiv b(\mathrm{mod}\, z)$ і k_0 – довільне ціле число, то $k_0 a \equiv k_0 b(\mathrm{mod}\, z)$;

5) якщо $k_0 a \equiv k_0 b(\mathrm{mod}\, z)$ й найбільший спільний дільник (НСД) $(k_0, z) = 1$, то $a \equiv b(\mathrm{mod}\, z)$;

6) якщо $a \equiv b(\mathrm{mod}\, z)$ й k_0 – довільне натуральне число, то $k_0 a \equiv k_0 b(\mathrm{mod}\, k_0 z)$;

7) якщо $k_0 a \equiv k_0 b(\mathrm{mod}\, k_0 z)$, де k_0 і z – довільні натуральні числа, то $a \equiv b(\mathrm{mod} z)$;

8) якщо $a \equiv b(\mathrm{mod}\, z)$, $c \equiv d(\mathrm{mod}\, z)$, то $a + c \equiv b + d(\mathrm{mod}\, z)$ й $a - c \equiv b - d(\mathrm{mod}\, z)$;

9) якщо $a \equiv b(\mathrm{mod}\, z)$, $c \equiv d(\mathrm{mod}\, z)$, то $ac \equiv bd(\mathrm{mod}\, z)$;

10) якщо $a \equiv b(\mathrm{mod}\, z)$, то при будь-якому цілому $k>0$, $a^k \equiv b^k(\mathrm{mod} z)$;

11) будь-який доданок лівої або правої частини конгруенції можна перенести із протилежним знаком в іншу частину.

Теорема про ділення з остачею: розділити число $a \in Z$ на число $b \in Z$, $b \neq 0$, з остачею означає знайти пару цілих чисел q та r, тобто таких, що виконують такі умови:

$$a = b\gamma + r, \quad 0 \leq r < |b|. \tag{1.2}$$

Легко доводиться, що для будь-яких цілих чисел a та $b \neq 0$ ділення з остачею можливо і числа γ і r визначаються однозначно.

Один із методів виконання арифметичних операцій за даними цілими числами ґрунтується на простих положеннях

теорії чисел. Ідея цього методу полягає в тому, що цілі числа подаються в одній із непозиційних систем – системі залишкових класів (СЗК), в якій замість операцій над цілими числами оперують із залишками від ділення цих чисел на обрані заздалегідь взаємно прості числа – модулі $p_1, p_2, ..., p_j$. Найчастіше числа $p_1, p_2, ..., p_j$ вибирають із множини простих чисел.

Нехай $A \equiv \alpha_1 (\mathrm{mod}\, p_1),\ A \equiv \alpha_2 (\mathrm{mod}\, p_2), \ldots,\ A \equiv \alpha_j (\mathrm{mod}\, p_j) \ldots$.

Важливо зазначити, що при цьому немає ніякої втрати інформації за умови, що $A < p_1 p_2 \ldots p_j = P$, тому що завжди, знаючи $(\alpha_1, \alpha_2, ..., \alpha_j)$, можна відновити число *A*. Відповідно кортеж $(\alpha_1, \alpha_2, ..., \alpha_j)$ можна розглядати як один зі способів подання цілого числа *A* у комп'ютері – модулярне подання або подання в СЗК.

Мультиплікативно оберненим елементом до числа *a* у модулярній арифметиці є таке число *b*, що виконується конгруенція [13, 14]:

$$ab \bmod z = 1. \qquad (1.3)$$

Умовою існування мультиплікативно оберненого елемента є рівність 1 НСД чисел *a* і *b*, тобто числа *a* і *b* мають бути взаємно прості. Якщо ця умова не виконується, то мультиплікативно обернений елемент до *a* не існує.

Методи пошуку мультиплікативного оберненого елемента можна розділити на дві великі категорії [15–17]: методи, що не ґрунтуються на методах пошуку НСД, і методи, які є похідними від методів пошуку НСД.

Найбільш поширеними методами, які належать до першої групи, є повний перебір всіх можливих варіантів (або брутальна атака) та метод на основі функції Ейлера.

Брутальною атакою слід розуміти метод розв'язання математичних задач, за якого складність повного перебору (брутальної атаки) залежить від кількості всіх можливих варіантів розв'язку задачі. Цей метод належить до класу методів пошуку розв'язку задач із вичерпуванням можливих варіантів розв'язку системи. Це найпростіший і водночас найзатратніший метод. Він характеризується високою обчислювальною складністю, оскільки повний перебір потребує значних часових затрат.

Функція Ейлера $\varphi(z)$, де z – натуральне число, це цілочисельна функція, яка дорівнює кількості натуральних чисел, не більших за z і взаємно простих з ним. Функцію Ейлера можна подати у вигляді так званого добутку Ейлера:

$$\varphi(z) = z \prod_{p_0 | z} \left(1 - \frac{1}{p_0}\right), \tag{1.4}$$

де p_0 – просте число.

При використанні теореми Ейлера $a^{\varphi(z)} \bmod z = 1$ отримується: $a^{\varphi(z)-1} \bmod z = a^{-1} \bmod z$. Така процедура передбачає виконання операції модулярного експоненціювання, що може привести до переповнення розрядної сітки процесора та ускладнює пошук оберненого елемента за модулем для багаторозрядних чисел.

Найбільш ефективними та поширеними є методи другої категорії, зокрема пошук мультиплікативного оберненого елемента за модулем за допомогою розширеного алгоритму Евкліда. Для цього спочатку потрібно записати прямий

алгоритм Евкліда, згідно з яким для будь-якого *z>a=r0*, де *z* і *a* – цілі числа, виконується така система рівнянь:

$$\begin{aligned}
&z = r_0 \times q_1 + r_1,\ q_1 = a,\ 0 \le r_1 < r_0;\\
&r_0 = r_1 \times q_2 + r_2,\ 0 \le r_2 < r_1;\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&r_{j-3} = r_{j-2} \times q_{j-1} + r_{j-1},\ 0 \le r_{j-1} < r_{j-2};\\
&r_{j-2} = r_{j-1} \times q_j + r_j,\ 0 \le r_j < r_{j-1};\\
&r_{j-1} = r_j \times q_{j+1} + 0.
\end{aligned} \qquad (1.5)$$

Оскільки *a* і *z* є взаємно простими, то r_j=1. Далі для реалізації розширеного алгоритму Евкліда описану процедуру необхідно повторити в зворотному порядку:

$$\begin{aligned}
&1 = r_j = r_{j-2} - q_j r_{j-1} = r_{j-2} - q_j \left(r_{j-3} - q_{j-1} r_{j-2}\right) = r_{j-2} - q_j r_{j-3} +\\
&+ q_j q_{j-1} r_{j-2} = -q_j r_{j-3} + \left(1 + q_j q_{j-1}\right) r_{j-2} = -q_j r_{j-3} + \left(1 + q_j q_{j-1}\right) \times \qquad (1.6)\\
&\times \left(r_{j-4} - q_{j-2} r_{j-3}\right) = \ldots
\end{aligned}$$

Такий процес продовжується доти, поки не отримається вираз $v \cdot z + t \cdot r_0 = 1$, де величина $b = t \bmod z = a^{-1} \bmod z$ і буде шуканим оберненим елементом. У табл. 1.1 наведено приклад використання розширеного алгоритму Евкліда, а на рис. 1.1 – його блок-схему.

Таблиця 1.1

Пошук оберненого елемента $41^{-1} \bmod 157$ за допомогою розширеного алгоритму Евкліда

Алгоритм Евкліда	Розширений алгоритм Евкліда
157=41×3+34 41=34×1+7 34=7×4+6 7=6×1+1	1=7–1×6=7–1×(34–4×7)= –1×34+5×7=–1×34+5× (41– –1×34)=5×41–6×34=5×41–6×(157–3×41)= –6×157+23×41; $41^{-1} \bmod 157$=23

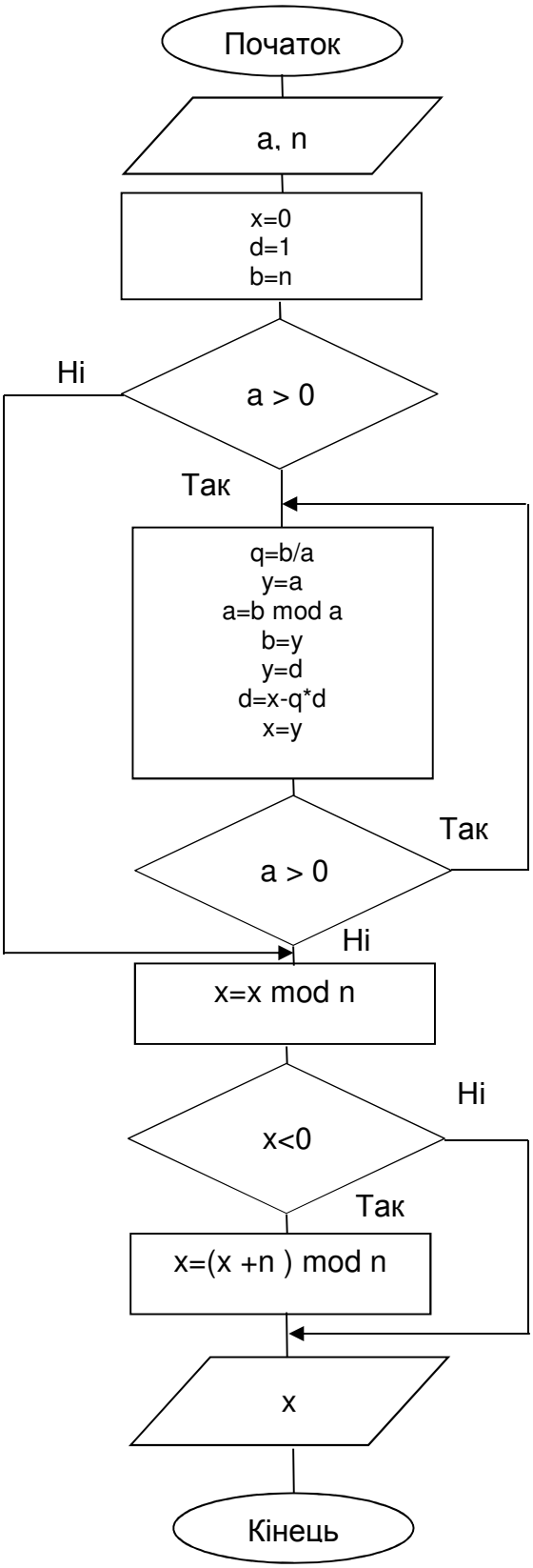


Рис. 1.1. Блок-схема розширеного алгоритму Евкліда

Описаний метод характеризується великою кількістю ділень з остачею, перемножень та підстановок, хоч і виокремлюється найменшою часовою складністю порівняно з двома іншими.

Методи пошуку оберненого елемента на основі алгоритму Евкліда можна поділити на два класи [13, 18]:

– методи, що ґрунтуються на застосуванні класичного алгоритму Евкліда [19];

– методи, що засновані на бінарному алгоритмі Евкліда [20].

Методи другого класу є більш ефективними, оскільки не використовують обчислювально-витратних операцій ділення на довільне число. Ці методи використовують лише елементарні операції, такі як додавання, віднімання та ділення на 2, що є еквівалентним зсуву на один двійковий розряд праворуч.

Поширеним застосуванням розширеного алгоритму Евкліда є китайська теорема про залишки (КТЗ) – один із найдавніших, але доволі важливий нині обчислювальний алгоритм.

У I ст. н. е. китайський математик Сунь-Цзи запропонував цікаву загадку, якою було започатковано модулярну арифметику: потрібно було знайти число, що при діленні на 3 матиме остачу 2, на 5 – 3, на 7 – 2. Крім того, він у частковому випадку показав еквівалентність розв'язку системи модулярних рівнянь та розв'язку одного модулярного рівняння.

Протягом майже двох тисяч років КТЗ постійно вдосконалювалася і розвивалася. Зокрема, у XIII ст. інший китайський математик Цань Цзю-шао розв'язав наведену вище задачу. В XVIII ст. німецький математик Л. Ейлер навів загальне формулювання і доведення КТЗ, а К.-Ф. Гаус істотно розвинув його в праці «Арифметичних дослідженнях» [21].

У середині XX ст. чеські учені М. Валах і А. Свобода запропонували використати давню китайську ідею на новому технічному рівні, створивши перші модулярні електронно-обчислювальні машини «Епос» і «Епос-2» [22].

Слід зазначити, що нині існує декілька еквівалентних формулювань КТЗ. Найбільш поширене з них таке [9]: якщо натуральні числа p_1, p_2, …, p_j попарно взаємно прості, то для будь-яких цілих r_1, r_2, …, r_j, таких що $0 \le r_i < p_i$, існує число A, яке при діленні на p_i має залишок r_i при всіх i=1, 2, ..., j; окрім того, якщо існує два таких числа A_1 та A_2, то $A_1 \bmod P = A_2 \bmod P$, де $P = \prod_{i=1}^{j} p_i$.

Ця теорема переважно має вигляд такої системи порівнянь:

$$\begin{cases} A \bmod p_1 = r_1 \\ A \bmod p_2 = r_2 \\ \ldots\ldots\ldots\ldots \\ A \bmod p_i = r_i \\ \ldots\ldots\ldots\ldots \\ A \bmod p_j = r_j. \end{cases} \qquad (1.7)$$

Шукане число обчислюється за формулою:

$$A = \left(\sum_{i=1}^{j} M_i m_i r_i \right) \bmod P, \qquad (1.8)$$

де $M_i = \dfrac{P}{p_i} = p_1 p_2 \ldots p_{i-1} p_{i+1} \ldots p_j$, $m_i = M_i^{-1} \bmod p_i$.

Пошук мультиплікативного оберненого елемента, необхідний для реалізації КТЗ, характеризується значною

обчислювальною складністю через неможливість розпаралелення процесу обчислень. Відповідно всі операції мають виконуватися над дуже великими числами, що може привести до переповнення розрядної сітки сучасних потужних обчислювальних засобів.

1.2. Теоретичні основи системи залишкових класів

Будь-яка обчислювальна структура взаємопов'язана із системою числення, в якій вона працює. Системою числення слід розуміти сукупність прийомів позначення (або запису) чисел або, спосіб кодування (або подання) елементів деякої кінцевої моделі дійсних чисел словами одного або більше алфавітів. Кодування є ін'єктивним відображенням діапазону системи числення на декартовий добуток його алфавітів, тобто $F : D \to A,$ де $A = A_1 \times A_2 \times ... \times A_j$. Отже, відображення F елемента $x \in D$ ставить у відповідність кодове слово $(x_1, x_2, ..., x_j)$, де $x_i = A_i (i = \overline{1, n})$ – i-та цифра, j – довжина коду. За допомогою зворотного відображення F^{-1}, яке називається декодуванням, також можна визначити систему числення.

У будь-якій кодовій системі мають виконуватись такі вимоги [80–82]:

– можливість подання у заданій системі будь-якої величини в розглянутому діапазоні, який заздалегідь визначено;

– одиничність подання – будь-яка кодова комбінація відповідає одному й тільки одному числу в заданому діапазоні;

– простота виконання операцій із числами в заданій системі числення.

Отже, коди чисел – це імена числових об'єктів, які становлять числовий діапазон. Діапазони як моделі дійсних чисел мають із максимально доступною повнотою та простотою відображати властивості числової множини.

Будь-яке подання чисел робочого діапазону є лише складеним елементом відповідної машинної арифметики і не може розглядатися окремо від неї [26–27]. Арифметичні властивості тої або іншої системи числення насамперед визначаються характером міжрозрядних зв'язків, які з'являються у ході виконання операцій над кодовими словами. Дослідження підтверджують [28–31], що в межах звичайної позиційної системи числення (ПСЧ) значного пришвидшення виконання операцій домогтися неможливо. Це пояснюється тим, що в ПСЧ значення розряду будь-якого числа, крім молодшого, що є результатом двомісної арифметичної операції, залежить не тільки від значення однойменних операндів, а й від усіх молодших розрядів, тобто ПСЧ має строго послідовну структуру. Однак сьогодні перевага надається обчислювальним структурам, що мають здатність до паралельної обробки інформації [32–39]. Такі особливості мають непозиційні коди із паралельною структурою, що дають змогу реалізувати ідею розпаралелювання операцій на рівні виконання елементарних арифметичних операцій. Ця думка сформувалась у середині 1950-х рр., коли М. Валах і Л. Свобода в дослідженнях про непозиційні системи числення [40, 41] розглядали подання чисел у вигляді набору залишків від ділення на обрані натуральні модулі – основи системи. Подібну систему числення почали називати СЗК або модулярною системою числення. Можливість застосування цієї системи в обчислювальних системах розглянута також у дослідженнях таких зарубіжних і вітчизняних учених:

І. Акушського [42, 43], Д. Юдіцького [44], В. Торгашова [45], В. Амербаєва [46, 47], М. Червякова [48–50], В. Краснобаєва [51–54], Я. Николайчука [55–58], А. Омонді [59], Б. Премкумара [60–61], А. Могана [62–63], О. Фінька [64].

Нехай задані додатні взаємно прості числа $p_1, p_2, ..., p_j$, які називають основами або модулями системи. Позначимо $P = \prod_{i=1}^{j} p_i$. Ця величина характеризує величину діапазону системи [44, 59, 63]. СЗК – це така непозиційна система числення, в якій ціле невід'ємне число *N* можна подати у вигляді набору залишків від ділення цього числа на обрані основи системи, тобто

$$A = (\alpha_1, \alpha_2, ..., \alpha_n),\ \alpha_i = A - \left[\frac{A}{p_i}\right] \cdot p_i, \left(i = \overline{1, j}\right). \qquad (1.9)$$

Можливість такого подання числа визначається теоремою про ділення з остачею в кільці цілих чисел [65]: якщо $A \in Z, p \in Z, p \neq 0$, то існують єдині $q_0 \in Z, r_0 \in Z$, тобто такі, що $A = q_0 p + r_0, 0 \le r_0 < |p|,\ \ q_0 = \left[\frac{A}{p}\right]$.

Нескладно помітити, що кожна остача r_i виходить незалежно від інших та містить інформацію про все число.

Установити взаємнооднозначну відповідність між цілими числами з діапазону [0, *P*) та їхніми залишками дозволяє китайська теорема про залишки [66–67].

Можливість застосування СЗК в обчислювальних алгоритмах визначається наявністю певного ізоморфізму між математичними операціями над цілими числами й відповідними операціями над системою цілих невід'ємних залишків за окремими модулями. При цьому додавання, множення,

піднесення до цілого додатного ступеня будь-яких цілих додатних чисел ідентичні відповідним операціям, що виконуються над системою залишків [44].

Нехай операнди *A* і *B*, а також результати операцій додавання та множення *A + B* та *A·B* представлені відповідно залишками α_i, β_i, γ_i, δ_i за модулями $p_i, (i = \overline{1, j})$, причому обидва числа і результати перебувають у діапазоні [0, *P*), тобто

$$\begin{aligned} A &= (\alpha_1, \alpha_2, ..., \alpha_j); \\ B &= (\beta_1, \beta_2, ..., \beta_j); \\ A + B &= (\gamma_1, \gamma_2, ..., \gamma_j); \\ A \cdot B &= (\delta_1, \delta_2, ..., \delta_j). \end{aligned} \qquad (1.10)$$

¡ $0 \le A < P$, $0 \le B < P$, $0 \le A + B < P$, $0 \le A \cdot B < P$.

Вирази (1.10) можна переписати у такому вигляді:

$$\gamma_i = \alpha_i + \beta_i (\mathrm{mod}\ p_i);\ \ \delta_i = \alpha_i \beta_i (\mathrm{mod}\ p_i); \qquad (1.11)$$

$$\gamma_i = \alpha_i + \beta_i - \left[\frac{\alpha_i + \beta_i}{p_i} \right] p_i,\ \ \delta_i = \alpha_i \beta_i - \left[\frac{\alpha_i \beta_i}{p_i} \right] p_i. \qquad (1.12)$$

Справедливість цих правил виконання арифметичних операцій у СЗК випливає безпосередньо з властивостей конгруенцій.

Так, на підставі (1.9) $\gamma_i = A + B - \left[\frac{A + B}{p_i} \right],\ (i = \overline{1, j}).$

З представлення *A* та *B* за теоремою про ділення з остачею випливає, що $A = s_{1i} p_i + \alpha_i$, $B = s_{2i} p_i + \beta_i$, $(i = \overline{1, j})$, $s_{1i} \in Z,\ s_{1i} \ge 0,\ s_{2i} \in Z,\ s_{2i} \ge 0$.

Тоді $A + B = (s_{1i} + s_{2i})\,p_i + \alpha_i + \beta_i, \qquad \left[\frac{A+B}{p_i}\right] = s_{1i} + s_{2i} + \left[\frac{\alpha_i + \beta_i}{p_i}\right] p_i.$

Звідси $\gamma_i = \alpha_i + \beta_i - \left[\frac{\alpha_i + \beta_i}{p_i}\right] p_i.$

У випадку множення $\delta_i = AB - \left[\frac{AB}{p_i}\right] p_i$. Тоді

$$AB = s_{1i} s_{2i} p_i^{\,2} + (\alpha_i s_{2i} + \beta_i s_{2i})\,p_i + \alpha_i \beta_i,$$

$$\left[\frac{AB}{p_i}\right] = s_{1i} s_{2i} p_i + \alpha_i s_{2i} + \beta_i s_{1i} + \left[\frac{\alpha_i \beta_i}{p_i}\right].$$

Отже, $\delta_i = \alpha_i \beta_i - \left[\frac{\alpha_i \beta_i}{p_i}\right] p_i, (i = \overline{1,n)}.$

Приклад:

Дані: $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_4 = 7$. $A = (0, 0, 3, 4)$, $B = (1, 1, 2, 0)$.

Знайти: $A+B$, $A - B$, AB.

Розв'язання: $P = \prod_{i=0}^{n} p_i = 2 \cdot 3 \cdot 4 \cdot 7 = 210.$

$A+B = (0, 0, 3, 4) + (1, 1, 2, 0) = (1, 1, 0, 4)$.

$AB = (0, 0, 3, 4) \cdot (1, 1, 2, 0) = (0, 0, 1, 0)$.

$A - B = (0, 0, 3, 4) - (1, 1, 2, 0) = (1, 2, 1, 4)$.

На відміну від ПСЧ, в якій число A представляється у вигляді

$$A = A_n N^n + A_{n-1} N^{n-1} + \ldots + A_0 N^0 = \sum_{i=0}^{n} A_i N^i, \qquad (1.13)$$

де N – основа ПСЧ, значення числа в модулярному коді не залежить від місця розташування кожного розряду в його представленні, а залежить від значення основи відповідного розряду. З огляду на це модулярний код є непозиційним.

Отже, виконання арифметичних операцій у модулярному коді відбувається незалежно за кожним із модулів, що і вказує на паралелізм цієї системи. Така обставина визначає порозрядне виконання операцій. Ця властивість позбавляє від необхідності «позичати» чи «переносити» одиницю старшого розряду, що приводить до появи кодів із паралельною структурою. Це дозволяє розпаралелити алгоритми при виконанні арифметичних операцій [50, 68–72].

Переведення чисел із ПСЧ у СЗК за допомогою виразу (1.9) пов'язано з реалізацією операції ділення, яка характеризується значною обчислювальною складністю [73–80], тому використання цього методу є неефективним.

Отже, операції додавання й множення над числами, представленими у СЗК, зводяться до відповідних операцій над числами цього подання [81–86]. Це стосується і піднесення і степеня, до обчислення значень многочлена тощо. Операція віднімання в СЗК замінюється додаванням з адитивною інверсією від'ємного числа. Всі ці операції модульні, тобто не потребують позиційних характеристик чисел, які опрацьовуються.

Дослідження СЗК дали змогу виявити такі основні переваги:

1) максимальний паралелізм [87–90]. Для оцінювання рівня паралелізму системи числення вводиться спеціальний показник

$$\prod(v) = \frac{n(v)}{\lambda}, \qquad (1.14)$$

де λ – довжина коду системи; $n(v)$ – кількість порозрядних показників паралелізму $\pi_1, \pi_2, ..., \pi_j$, не менших від заданого порога $v\left(\frac{1}{j} \le v \le 1\right)$, причому $\pi_i = 1 - \frac{n_i}{j} (i = \overline{1, j})$; n_i – максимально

можливе число пар цифр (x_j, y_j), які здійснюють вплив на значення суми $Z = X + Y$ у ході її формування мовою цього коду. Для СЗК показник паралелізму набуває максимально можливого значення 1. Це вказує на відсутність міжрозрядних зв'язків у числі, записаному в СЗК;

2) малоразрядність залишків [71, 91–95]. Через малу кількість можливих кодових комбінацій з'являється можливість побудови табличної арифметики. При цьому більшість операцій перетворюється в однотактові, що здійснюється простою вибіркою з таблиць. За вдосконалення технології виробництва запам'ятовувальних пристроїв із високою щільністю запису інформації, що становлять технічну систему табличного методу обчислень, інтерес до СЗК щораз зростає;

3) реалізація принципу конвеєрної обробки інформації [96–99]. Це означає, що при виконанні обчислень модульні й подальші за ними операції вдається сполучити за часом тільки тоді, коли чергові операції залежать від результатів поточних операцій, що ще не закінчилися. Таким чином, алгоритми модулярної арифметики мають конвеєрну структуру;

4) висока точність, надійність, здатність до самокорекції [100–111]. У СЗК можна побудувати непозиційні коди, що виявляють та виправляють помилки, котрі є повністю арифметичними, тобто у цих кодах інформативна та контрольна частини рівноправні щодо будь-якої операції. Така особливість дає можливість варіювати коригувальною здатністю коду за зміни точності обчислень.

Однак і ця система не позбавлена недоліків. До них належать неможливість візуального порівняння чисел [112–114], відсутність ознак виходу результатів за межі діапазону [115–118], обмеженість дії системи сферою цілих додатних чисел [59, 63, 119–121], одержання у всіх випадках точного

результату операції [122–124], що виключає можливість безпосереднього округлення, а також труднощі виконання немодульних операцій [125–128]. Однак вони не є непереборними.

До модульних операцій у СЗК належать операції додавання, віднімання і множення. Аналіз використання арифметики СЗК вказує, що їх ланки мають однакову структуру, типовим елементом якої є послідовність виду:

$$V = \left| \sum_{i=1}^{j} \left| F_i(\alpha_i) \right|_{p_i} \right|_{p_i}, \qquad (1.15)$$

де $F_i(\alpha_i)$ – цілочисельна функція залишку α_i за деяким модулем; p_i – основа СЗК; $|...|_{p_i}$ – операція визначення найменшого залишку за модулем p_i.

До немодульних операцій належать операції, при яких значення того або іншого результату розряду залежить від його всіх або декількох розрядів вихідного числа.

Пристрої, що реалізують немодульні операції, поділяються на два типи [129–130].

Прикладом пристрою першого типу є пристрій згортки, який забезпечує обчислення:

$$V = \left| \sum_{i=1}^{j} \left| A_i Q_i \right|_{p_i} \right|_{p_i}, \qquad (1.16)$$

де A_i – значення i-го розряду вихідного числа, представленого у позиційній системі числення (ПСЧ); Q_i – ваговий коефіцієнт.

Пристрої згортки широко використовуються у цифрових системах, які функціонують у СЗК і становлять істотну, а інколи

й основну частину устаткування, призначену для реалізації певних способів виконання операцій – переводу чисел із ПСЧ у СЗК, ділення довільних чисел та інших. Крім того, такі пристрої застосовують й у цифрових системах, що функціонують у ПСЧ.

Прикладом пристроїв другого типу є пристрої позиційного перетворення [131], що забезпечують одержання характеристик, які вказують на належність числа, представленого в СЗК, тому або іншому інтервалу діапазону можливого представлення чисел.

Математичною основою для пристроїв першого типу є визначення найменших невід'ємних залишків, які визначаються згортками вихідного числа за кожним модулем. Для визначення згорток за кожним модулем необхідно перевести число з ПСЧ у СЗК.

Переведення числа в СЗК можна здійснити методом ділення. Однак через операцію ділення технічна реалізація такого методу неефективна для машинного використання. Крім того, такий метод потребує застосування арифметичного пристрою в ПСЧ.

Розглянемо метод переведення числа з ПСЧ у СЗК, що не містить операції ділення. Це метод безпосереднього підсумовування модульних значень розрядів позиційного числа.

Нехай число X записане у позиційній системі числення із основою N, тобто

$$X = A_j N^j + A_{j-1} N^{j-1} + ... + A_0 N^0 \text{ або } X = \sum_{i=0}^{j} A_i N^i , \qquad (1.17)$$

де $0 \le A_i \le N-1$.

Представимо степені основи N^i і коефіцієнти A_i у СЗК з основами $p_1, p_2, ..., p_j$, тоді:

$$N^i = (B_1^{(i)}, B_2^{(i)}, ..., B_n^{(i)}), \quad A^i = (A_1^{(i)}, A_2^{(i)}, ..., A_n^{(i)}). \qquad (1.18)$$

Підставивши (1.18) в (1.9), можна одержати:

$$X = (\sum_{i=0}^{j-1} A_1^{(i)} B_1^{(i)} \bmod p_1, \sum_{i=0}^{j-1} A_2^{(i)} B_2^{(i)} \bmod p_2, ..., \sum_{i=0}^{j-1} A_j^{(i)} B_j^{(i)} \bmod p_j). \qquad (1.19)$$

Таким чином, для утворення числа *X* у СЗК потрібно *k* констант, що є степенями p_i й $p_i - 1$ констант, що відповідають значенням A_i.

Маючи в пам'яті процесора масив з $j + p_i - 1$ констант, переведення може бути здійснено процесором, що працює в СЗК [132–133].

Розглянутий метод є основою широкого вибору можливих апаратурних реалізацій цифрових перетворювачів ПСЧ – СЗК, які розрізняються між собою як за складом і кількістю елементів, що використовуються, так і за швидкістю перетворення інформації. Відомими у науковій літературі є цифрові перетворювачі ПСЧ – СЗК, функціонування яких ґрунтується на цьому методі. Їхній аналіз дав змогу зробити важливий висновок, що істотними недоліками подібних перетворювачів є більші апаратурні витрати при переведенні чисел великої розрядності й низька швидкодія. Підвищені вимоги, пов'язані зі зменшенням апаратурних засобів і збільшенням швидкості обробки інформації, привели до необхідності вивчення питань розробки ефективних алгоритмів. Для вирішення цього завдання пропонуються два методи. Розглянемо перший метод. Він ґрунтується на теоремі, що є основою цього методу перетворення чисел із ПСЧ у СЗК як апаратурними, так і програмними способами. Нехай число *X* записане у позиційній системі числення із основою *N*. Якщо

$X_j = \sum_{i=0}^{n_0} A_i^{(j)} C_i^j$, де $C_i \equiv N^i (\mathrm{mod}\ p_i)$, p_i – просте число, n_0 – число розрядів p_i (при i=1, 2,…, n_0), то $X \equiv X_j (\mathrm{mod}\ p_j)$ і $X_j < X$.

Розглянемо другий метод, що набув широкого застосування в науковій літературі. Назвемо його методом послідовного множення й підсумовування. Суть методу полягає в наступному. Нехай число записане у вигляді (1.17). Інакше цей вираз можна записати так:

$$
\begin{gathered}
X = (...(A_j N + A_{j-1})N + A_{j-2})N + ... + A_1)N + A_0 \equiv X_1 \bmod p_j = -\alpha_j \bmod p_j \\
(A_j N + A_{j-1})N \equiv X_j (\mathrm{mod}\ p_j), \\
(X_j \bmod P_j + A_{j-2})N \equiv X_{j-1} (\mathrm{mod}\ p_j), \\
(X_3 \bmod p_j + A_1)N \equiv X_2 (\mathrm{mod}\ p_j), \\
X_2 \bmod p_j + A_0 \equiv X_1 \bmod p_j = \alpha_j \bmod p_j .
\end{gathered}
\qquad (1.20)
$$

Так, значення числа X у СЗК за модулем p_j утвориться шляхом множення старшого розряду на основу системи числення N, потім підсумовування отриманого результату із значенням наступного розряду за модулем p_j, потім множення отриманого результату на основу N за модулем p_j. Такі послідовні множення й підсумовування за модулем виконуються доти, поки при підсумовуванні не буде додане значення молодшого розряду.

Слід зазначити, що розглянутий метод дозволяє реалізувати доволі економічні щодо апаратурних витрат цифрові пристрої перетворення інформації [134].

СЗК має одну особливість, яку можна зарахувати до недоліків цієї системи: не можна визначити візуально величину числа, яке представлене у СЗК, а отже, неможливе виконання

таких операцій, як порівняння чисел, визначення знаку числа. Один зі шляхів вирішення цієї проблеми полягає у перетворенні чисел із СЗК у ПСЧ. Оцінимо існуючі способи переведення: як традиційні (метод ортогональних базисів; переведення числа в узагальнену позиційну систему, так і нові (інтервальні методи переведення).

Основою методу ортогональних базисів відновлення числа за його залишками при переведенні із СЗК у ПСЧ є КТЗ (1.1).

Нехай взаємно прості основи СЗК $p_1, p_2, ..., p_j$; $P = \prod_{i=1}^{j} p_i$ – діапазон системи. З вибором системи визначаються її основні константи – базиси B_i, $i = \overline{1, j}$. Задача переведення числа $A = (\alpha_1, \alpha_2, ..., \alpha_j)$ у ПСЧ полягає у визначенні таких чисел B_i, $i = \overline{1, j}$, щоб $A = \sum_{i=1}^{j} p_i B_i$. Для однозначного визначення p_i на базиси системи B_i накладається ряд обмежень і показується, що таку властивість мають такі базиси:

B_1 = (1, 0, 0, …, 0, 0), B_2 =(0, 1, 0, …, 0, 0), ..., B_j = (0, 0, 0, …, 0, 1),

які називаються ортогональними.

Тоді у випадку ортогональних базисів $P_i = \alpha_i$, $i = \overline{1, j}$. Ортогональні базиси визначаються за такою формулою:

$$B_i = \frac{m_i P}{p_i} = m_i P_i \ , \ i = \overline{1, j}, \qquad (1.21)$$

де

$$P_i = \frac{P}{p_i} = p_1 \cdot p_2 \cdot ... \cdot p_{i-1} \cdot p_{i+1} \cdot ... \cdot p_j, \qquad (1.22)$$

m_i – цілі позитивні числа, що називаються вагами базису. Їх визначають із порівнянь:

$$P_i m_i \left(\mathrm{mod}\ p_i\right)_{=1}. \qquad (1.23)$$

Тоді згідно КТЗ отримаємо число $A=(\alpha_1, \alpha_2,..., \alpha_j)$ $= \sum_{i=1}^{j} \alpha_i B_i (\mathrm{mod}\, P)$.

Таким чином, якщо знайдені ортогональні базиси для системи основ, то для переведення числа $A=$ $(\alpha_1, \alpha_2,..., \alpha_j)$ достатньо обчислити $\sum_{i=1}^{j} \alpha_i B_i$ і ввести цю суму в діапазон [0; P) відніманням величини, кратної P, тобто

$$A = \left| \sum_{i=1}^{j} \alpha_i B_i \right|_p = \sum_{i=1}^{j} \alpha_i B_i - r_A P, \qquad (1.24)$$

де r_A – ранг числа A, який вказує, скільки разів треба відняти величину діапазону P з отриманого числа, щоб повернути його в діапазон.

Розглянемо приклад. Нехай задана система основ $p_1=$ 2, $p_2=$ 3, $p_3=$ 5, $p_4=$7, $p_5=$11, об'єм діапазону P =2·3·5·7·11=2310. Перевести число $A=$ (1, 2, 1, 4, 7) у позиційну систему.

Обчислимо ортогональні базиси. Для цього знайдемо величини P_i:

$$P_1 = \frac{P}{p_1} = 1155,\ P_2 = \frac{P}{p_2} = 770,\ P_3 = \frac{P}{p_3} = 462,\ P_4 = \frac{P}{p_4} = 330,$$

$$P_5 = \frac{P}{p_5} = 210.$$

Шукаємо ваги базисів:

$1155\,m_1 \equiv 1(\mathrm{mod}\ 2),\ m_1 \equiv 1(\mathrm{mod}\ 2).$

$770\,m_2 \equiv 1(\mathrm{mod}\ 3),\ m_2 \equiv 2(\mathrm{mod}\ 3).$

$462\,m_3 \equiv 1(\mathrm{mod}\ 5),\ m_3 \equiv 3(\mathrm{mod}\ 5).$

$330\,m_4 \equiv 1(\mathrm{mod}\ 7),\ m_4 \equiv 1(\mathrm{mod}\ 7).$

$210\,m_5 \equiv 1(\mathrm{mod}\ 11),\ m_5 \equiv 1(\mathrm{mod}\ 11).$

Тоді одержуємо базиси:

$B_1 = m_1 \cdot P_1 = 1 \cdot 1155 = 1155,$

$B_2 = m_2 \cdot P_2 = 2 \cdot 770 = 1540,$

$B_3 = m_3 \cdot P_3 = 3 \cdot 462 = 1386,$

$B_4 = m_4 \cdot P_4 = 1 \cdot 330 = 330,$

$B_5 = m_5 \cdot P_5 = 1 \cdot 210 = 210.$

Обчислимо величину числа A:

$$A = \left|1 \cdot 1155 + 2 \cdot 1540 + 1 \cdot 1386 + 4 \cdot 330 + 7 \cdot 210\right|_{2310} = \left|8411\right|_{2310} = 1481.$$

Оскільки ортогональні базиси B_i повністю визначаються вибором основ системи, то вони можуть бути обчислені заздалегідь, причому єдиний раз.

Недолік розглянутого методу полягає у тому, що доводиться мати справу з великими числами B_i і, крім того, дії додавання та множення треба виконувати в ПСЧ, а отриманий результат необхідно вводити у діапазон відніманням величини, кратної P.

1.3. Сучасний стан і напрями підвищення ефективності використання системи залишкових класів

Використання СЗК у комп'ютерних системах дає змогу істотно підвищити швидкодію реалізації цілочисельних арифметичних операцій [135–136]. Наприклад, у дослідженні [137] проведено розрахунок і порівняльний аналіз продуктивності комп'ютерної системи обробки цілочисельних даних, представлених у СЗК. На основі порівняльного оцінювання продуктивності системи 15Э1235 (розробка алгоритму вибору шляху для комутації повідомлень) та даних наукових джерел [138] сформовано табл. 1.2 з часом виконання відповідних операцій над 32-бітними словами в ПСЧ та СЗК.

Таблиця 1.2

Характеристики системи 15Э1235 в ПСЧ та СЗК

№	Тип операції	К-сть операцій	Час виконання операцій (ум. од.)	
			ПСЧ	СЗК
1	Звертання до ОЗП	360	1080	1080
2	Звертання до ПЗП	100	300	300
3	Додавання	1314	9198	275
4	Множення	1600	320000	320
5	Порівняння	63	441	6

Крім того, з'ясовано, що продуктивність системи під час роботи з алгоритмами вибору шляху у ПСЧ становить 3 еталонних задачі за відносну умовну одиницю часу, а при використанні СЗК – 500 еталонних задач за відносну умовну одиницю часу, тобто майже в 170 разів більше.

У табл. 1.3 показано, що використання технологій паралельної обробки даних, зокрема СЗК, забезпечує більш високу надійність (за ймовірності безвідмовної роботи), ніж при застосуванні двійкової ПСЧ, при меншій кількості додатково введеного обладнання [138].

Таблиця 1.3

Показники продуктивності та надійності системи 15Э1235 в ПСЧ та СЗК

Показник	ПСЧ	СЗК
Продуктивність (ум. од.)	3	500
Надійність (імовірність безвідмовної роботи)	0,966	0,9999
Відносна к-сть обладнання (ум. од.)	64	60
К-сть додаткового обладнання (%)	100	87,5

Відповідно в дослідженні [51] стверджується, що продуктивність модулярної комп'ютерної системи може бути в десятки або сотні разів більшою, ніж у ПСЧ за такої самої тактової частоти.

У працях [139–142] запропоновано та досліджено ефективний метод для паралельного виконання високоточних арифметичних операцій над багаторозрядними числами на основі СЗК. Його суть полягає в перетворенні вихідних операндів у групи незалежних чисел меншої розрядності з подальшою паралельною обробкою елементів цих груп на багатоядерних обчислювачах (рис. 1.2). При цьому можна здійснити налаштування базису обчислень як під роботу з числами конкретної розрядності, так і під конкретну архітектуру системи, що дає можливості для оптимізації обчислювального процесу.

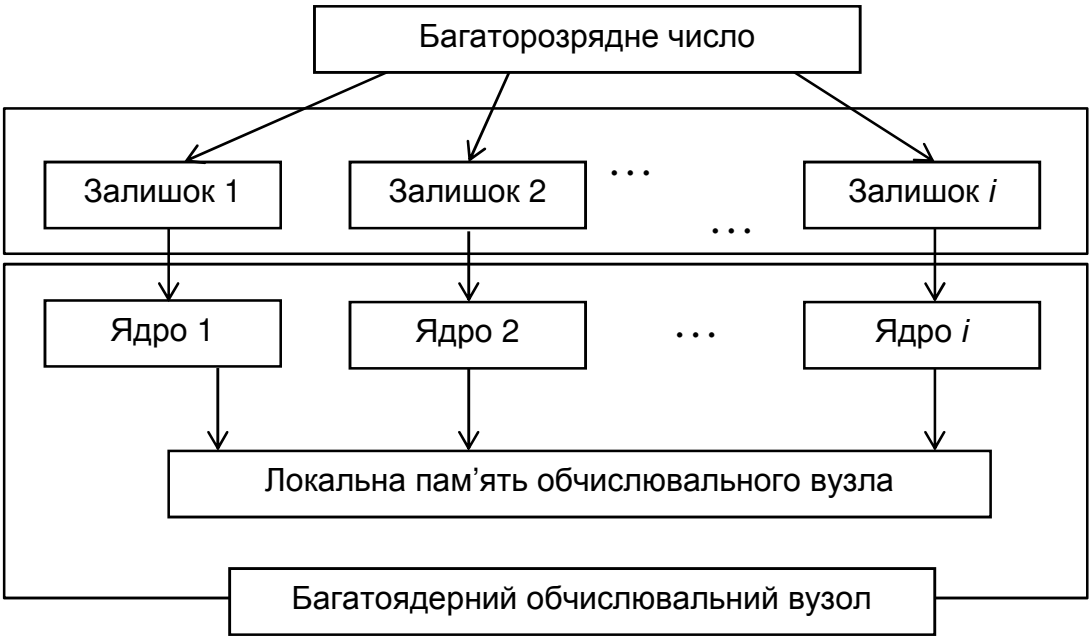


Рис. 1.2. Схема розподілу багаторозрядного числа між ядрами обчислювального вузла

В ході експерименту обчислювався добуток матриць розмірністю 700x700, елементами якої були цілі числа розрядністю від 32 до 248 біт, на основі послідовного та паралельного алгоритмів множення. Схеми розподілу масивів при паралельному алгоритмі для СЗК та ПСЧ представлені відповідно на рис. 1.3 та 1.4.

Графіки залежності часу виконання множення від розрядності числових даних і прискорення, яке досягається при використанні СЗК, подано на рис. 1.5 та 1.6. Так можна простежити, що при 32-бітних операндах час множення матриць з СЗК значно (приблизно в 40 разів при паралельному алгоритмі) менший від часу множення матриць у двійковій системі числення.

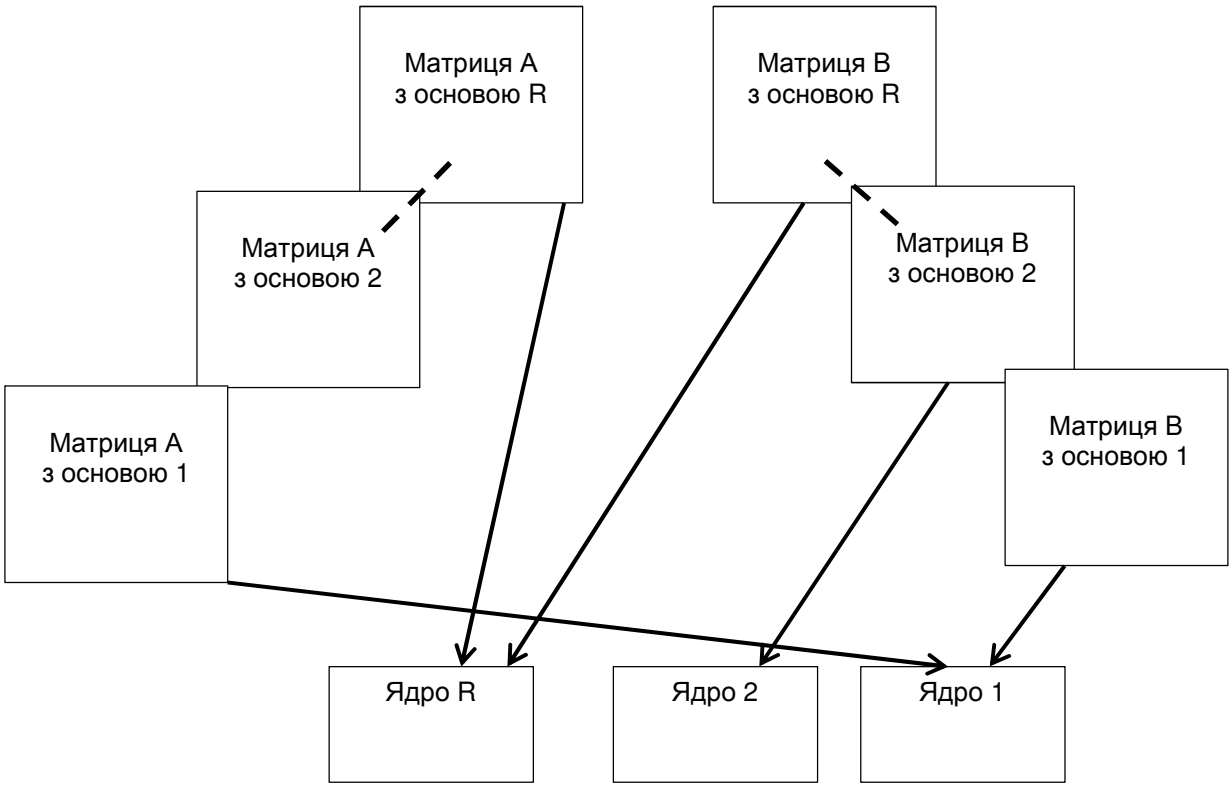


Рис. 1.3. Схема розподілу масивів у паралельному алгоритмі СЗК

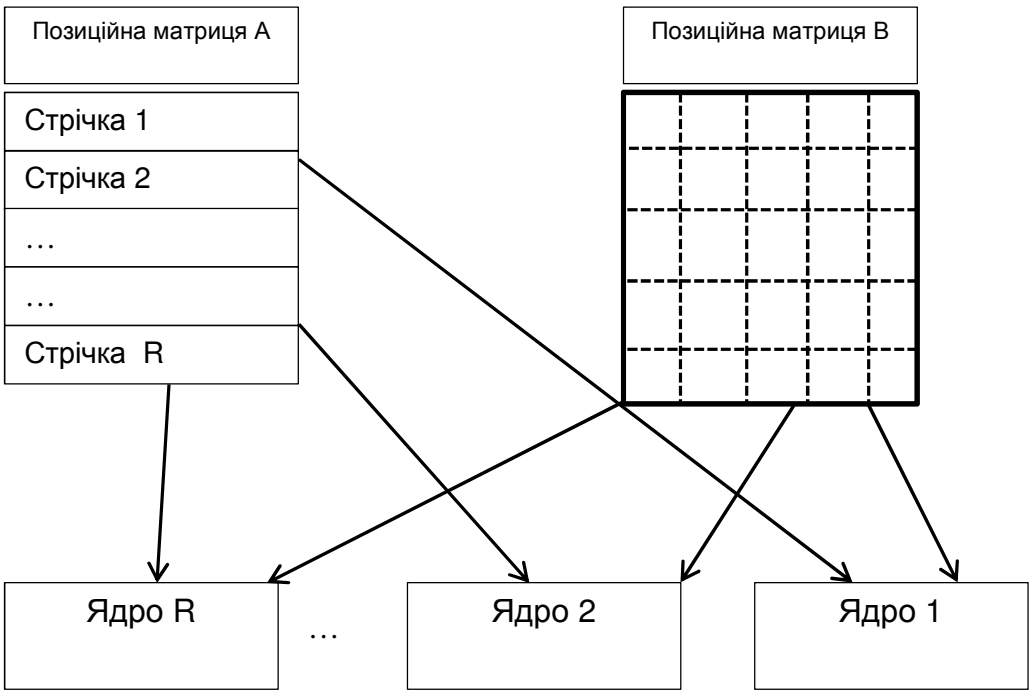


Рис. 1.4. Схема розподілу масивів у паралельному алгоритмі ПСЧ

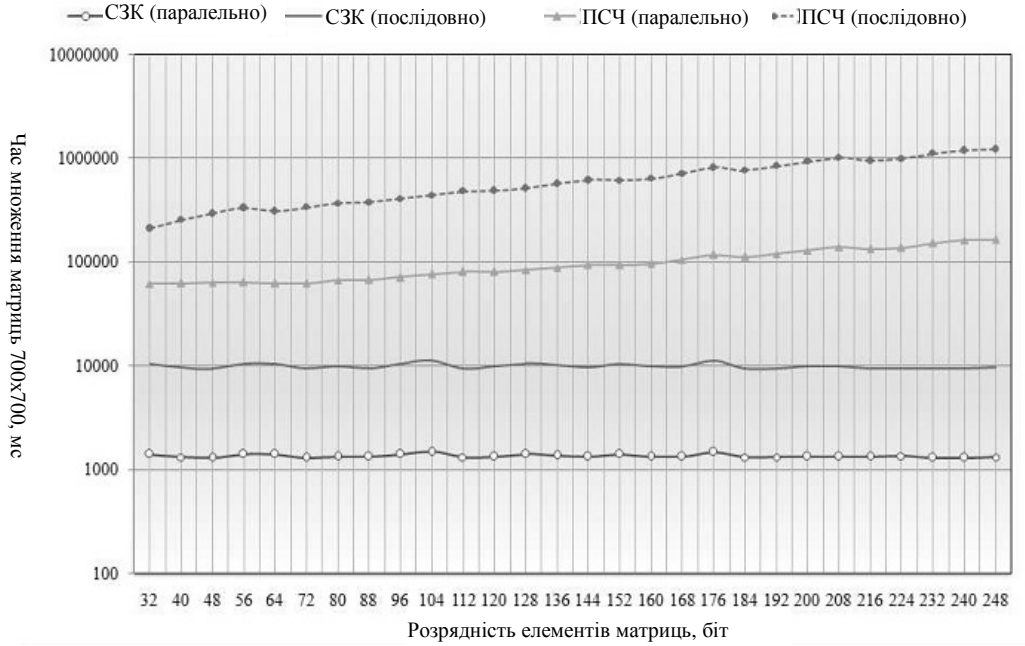


Рис. 1.5. Залежність часу множення матриць від розрядності елементів

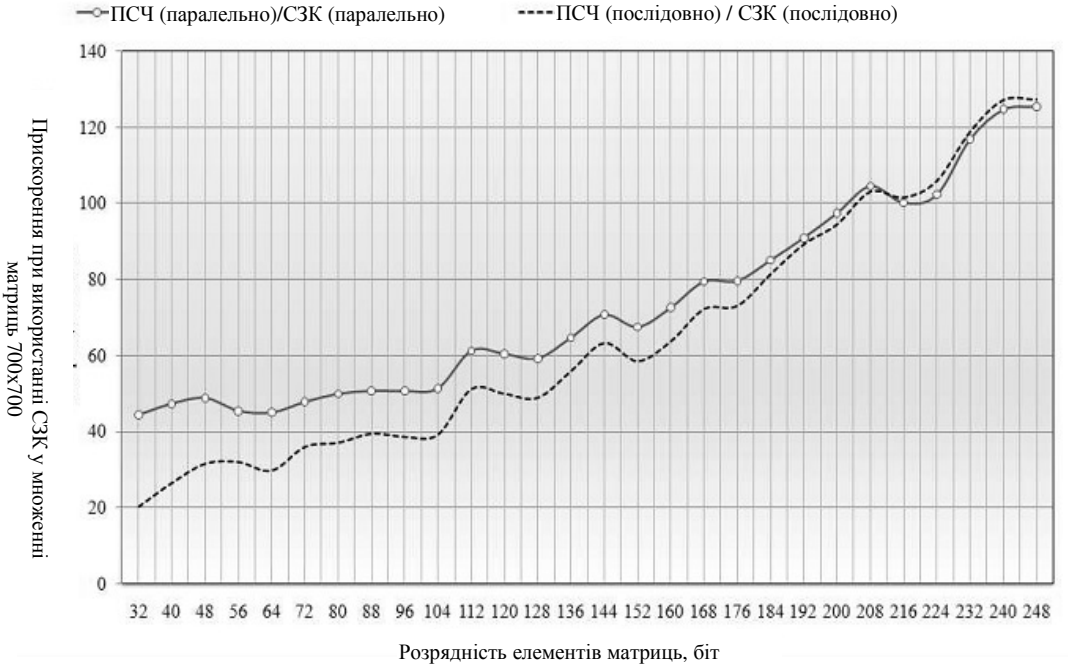


Рис. 1.6. Залежність прискорення СЗК від розрядності елементів

При збільшенні розрядності чисел прискорення з використанням СЗК постійно зростає і для 248-бітних елементів досягає

125 разів. Це пояснюється тим, що із збільшенням розрядності позиційні методи істотно сповільнюються, а в СЗК час обчислень не залежить від розрядності через замикання арифметичних операцій відносно кільця лишків за вибраними модулями.

Одним із шляхів підвищення швидкодії обчислювачів, які працюють у СЗК, є вибір спеціалізованих наборів модулів, від чого істотно залежить час виконання як модульних, так і немодульних операцій. Відповідно в СЗК запропоновано багато різних наборів модулів різного виду та різної кількості для визначених застосувань, які істотно впливають на всі частини апаратної реалізації, в т. ч. прямі перетворювачі, модульні арифметичні канали, зворотні перетворювачі [143–145]. Наприклад, для цифрової обробки сигналів потребується менше модулів, ніж для криптографії. У більшості праць розглядаються модулі виду 2^k, $2^k \pm 1$, що дає можливість раціонального використання регістрів розрядної сітки [146–148]. Найгірший модуль, тобто з найбільшою складністю виконання (може бути найбільшим або модулем комплексного типу), визначає загальний параметр прямого перетворювача або арифметичного каналу.

Крім того, реалізація схем для модулів 2^k+1 набагато складніша, ніж для 2^k або 2^k-1. Затримка зворотного перетворення для популярних модулів СЗК представлена в працях [149–152]. У табл. 1.4 наведено деякі набори спеціалізованих модулів і вказано їх характеристики [153].

Прямий перетворювач і арифметичні канали складаються з незалежних схем для кожного модуля, тому для більшої кількості модулів це забезпечує простіші схеми реалізації. Набір з трьох модулів типу $(2^k-1, 2^k, 2^k+1)$, $(2^k, 2^{k-1}-1, 2^k-1)$ і $(2^k, 2^{k+1}-1, 2^k-1)$ називається збалансованим і забезпечує обмежений динамічний діапазон та паралелізм.

Таблиця 1.4

Набори модулів СЗК та їх характеристики

Л-ра	Набір модулів	Рік	Характеристика
[154]	2^k-1, 2^k, 2^k+1	1967	Конверсійний
[155]	$2k-1$, $2k$, $2k+1$	1992	Неефективний
[156]	$2^{2k}+1$, 2^k+1, 2^k-1	1997	Конверсійний
[157]	2^k-1, 2^k, $2^{k-1}-1$	1998	Арифметичні
[158]	2^k-1, 2^k, $2^{k+1}-1$	1999	Арифметичний
[159]	2^k-1, 2^k, $2^{2k+1}-1$	2008	Арифметичний
[160]	$2^{2k}-1$, 2^k, $2^{2k}+1$	2008	Конверсійний
[161]	2^α, $2^\beta-1$, $2^\beta+1$	2008	Гнучкий, конверсійний
[162]	2^k-1, 2^k, 2^k+1, $2^{k+1}+1$	1999	Арифметичний
[163]	2^k-1, 2^k, 2^k+1, $2^{k+1}-1$	2000	Арифметичний
[146]	2^k-1, 2^k, 2^k+1, $2^{2k}+1$	2003	Конверсійний
[164]	2^k-1, 2^k+1, 2^k-3, 2^k+3	2004	Збалансований
[165]	2^k-1, 2^k+1, $2^{2k}-2$, $2^{2k+1}-3$	2008	Великий діапазон
[147]	2^k-1, 2^k+1, 2^{2k}, $2^{2k}+1$	2010	Конверсійний
[147]	2^k-1, 2^k, 2^k+1, $2^{2k+1}-1$	2010	Арифметичний
[166]	2^k-1, 2^k+1, 2^{2k}, $2^{2k+1}-1$	2010	Великий діапазон, арифметичний
[167]	2^α, 2^k-1, 2^k+1, $2^{k+1}+1$	2014	Арифметичний
[167]	2^α, 2^k-1, 2^k+1, $2^{k-1}-1$	2014	Арифметичний
[168]	2^k-1, 2^k, 2^k+1, $2^k-2^{(k+1)/2}+1$, $2^k+2^{(k+1)/2}+1$	2005	Конверсійний
[169]	2^k-1, 2^k, 2^k+1, $2^{k-1}-1$, $2^{k+1}-1$	2007	Збалансований, арифметичний
[170]	$2^{k/2}-1$, 2^k, $2^{k/2}+1$, 2^k+1, $2^{2k-1}-1$	2009	Незбалансований
[171]	2^k-1, 2^k, 2^k+1, $2^k-2^{(k+1)/2}+1$, $2^k+2^{(k+1)/2}+1$, $2^{k\pm1}+1$	2013	Дуже великий діапазон і паралелізм
[171]	2^k-1, $2^{k+\beta}$, 2^k+1, $2^k-2^{(k+1)/2}+1$, $2^k+2^{(k+1)/2}+1$, $2^{k\pm1}+1$	2013	Дуже великий діапазон, гнучкий
[172]	$2^{k+\beta}$, 2^k-1, 2^k+1, $2^k-\beta_1$, $2^k+\beta_1$, …, $2^k-\beta_i$, $2^k+\beta_i$	2013	Дуже великий діапазон, збалансований

Варто зазначити, що існують деякі схожі набори модулів до тих, які визначені у дослідженні [173]. Для розширення паралелізму можна використовувати чотири модулі типу (2^k–1, 2^k, 2^k+1, 2^{k+1}+1), (2^k–1, 2^k, 2^k+1, 2^{k+1}–1), (2^k–1, 2^k, 2^k+1, 2^{k-1}–1), (2^k–3, 2^k–1, 2^k+1, 2^k+3), а також 5 модулів – (2^k–1, 2^k, 2^k+1, 2^{k-1}–1, 2^{k+1}–1). Ці типи модулів називають арифметичними, оскільки вони є результатом ефективних арифметичних операцій.

На відміну від прямих перетворювачів і модульних арифметичних каналів, зворотне перетворення та складні операції СЗК потребують складних і немодулярних паралельних структур. Збалансовані набори модулів, які дуже популярні для арифметичної частини СЗК, не придатні для зворотних перетворювачів, оскільки призводять до складних мультиплікативних операцій і відповідно до зниження продуктивності зворотного перетворювача. Для забезпечення швидкого зворотного перетворення пропонуються такі набори (конверсійні) (2^k, 2^{2k}–1, 2^{2k}+1), (2^k–1, 2^k, 2^k+1, 2^{2k}+1), (2^k–1, 2^k, 2^k+1, 2^{2k+1}–1), (2^k–1, 2^k+1, 2^{2k}, 2^{2k}+1) і (2^k–1, 2^k+1, 2^{2k}, 2^{2k+1}–1). Серед них четвертий набір має найкращий зворотний перетворювач, що актуально для програм, які часто потребують зворотного перетворення. Водночас третій набір із заміною 2^{2k+1}–1 на 2^{2k}+1 є найкращим для додатків, які потребують великої кількості додавань та множень.

У наборі з більшою кількістю модулів (2^k–1, 2^k+β, 2^k+1, 2^k–$2^{(k+1)/2}$+1, 2^k+$2^{(k+1)/2}$+1, $2^{k\pm1}$+1) використовується незбалансований модуль 2^k+β для збільшення динамічного діапазону. Крім того, вводиться також узагальнення цього модуля, встановленого за допомогою модулів форм $2^k\pm\beta$. Цей тип модулів особливо підходить для криптографічних додатків, що потребують великої кількості модулів з ефективними арифметичними операціями, а також продуктивним зворотним

перетворювачем. Важливим моментом при цьому є розробка спеціальних арифметичних схем, що дає змогу підвищити ефективність.

Отже, для кожного конкретного застосування з вказаними арифметичними операціями, апаратними компонентами та обмеженнями необхідно вибирати відповідний набір модулів.

У табл. 1.5 [174] представлено результати апаратної реалізації СЗК на інтегральній схемі спеціального призначення (ASIC) для різних модулів (Converter–1: 2^k–1, 2^k+1, 2^{2k}, 2^{2k+1}–1 та Converter–2: 2^k–1, 2^k+1, 2^{2k}, 2^{2k}+1) для k=4, 8, 12 і 16 на базі технології TSMC. Досліджувалася площа чіпа (мкм^2), його корисна площа (мкм^2), часова затримка (нс), потужність (мВт).

Таблиця 1.5

Результати дослідження апаратної реалізації СЗК

Параметри схеми	**Converter-1**				**Converter-2**			
k	**4**	**8**	**12**	**16**	**4**	**8**	**12**	**16**
Площа чіпа, мкм^2	4639	9858	13710	17901	5040	9006	13478	17793
Корисна площа, мкм^2	1575,7	3223,8	4347,7	5638	1591,9	2787,5	4098,2	5353,9
Часова затримка (нс)	0,693	1,056	1,33	1,736	0,5	0,732	0,91	1,093
Потужність (мВт)	3,615	4,528	5,485	5,761	4,446	5,42	6,382	6,997

Так, що зворотний перетворювач для Converter-2 має кращий параметр затримки, ніж Converter-1, який характеризується швидшим арифметичним блоком СЗК та прямим

перетворювачем. Таким чином, використання певного набору модулів залежить від визначеного завдання.

Отже, відповідний набір модулів та арифметика СЗК при апаратній реалізації дають змогу суттєво зменшити споживання енергії, особливо в мультиплікаторах [175] та мультиплікаторах-накопичувачах [176–177]. Проте реалізація інших арифметичних операцій (наприклад, ділення, виявлення знаків, порівняння чисел, виявлення переповнення розрядної сітки) ускладнює обчислення в СЗК і цього слід уникати. Відповідно СЗК особливо корисні в алгоритмах, домінуючими операціями в яких є множення та додавання [178]. Наприклад, у дослідженнях [175, 179] запропонована програма обробки цифрових зображень на основі СЗК, у працях [177, 180] описуються фільтри з обмеженим імпульсним відгуком на основі СЗК. Огляд потенціалу та застосунків СЗК представлений у працях [181–182]. Загальна схема апаратної реалізації СЗК подана на рисю 1.7.

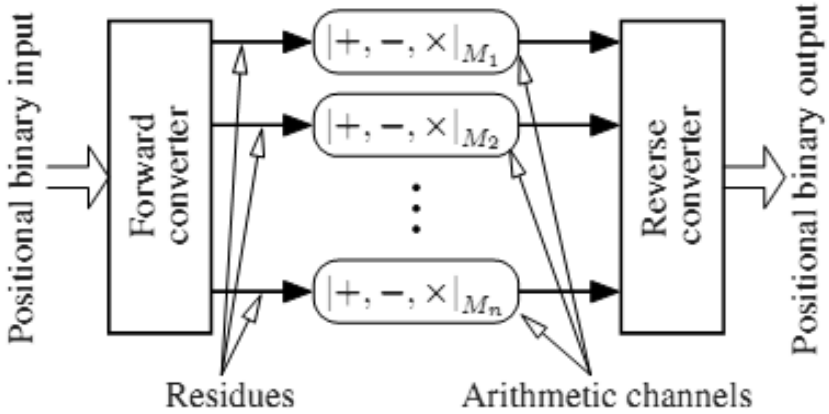


Рис. 1.7. Загальна схема апаратної реалізації СЗК

Схема складається з трьох основних частин: прямого перетворювача, арифметичних каналів СЗК та зворотного перетворювача. Перехідний перетворювач переводить вхідні

двійкові числа в набір значень залишків. Додавання, віднімання та множення виконуються за допомогою незалежних схем, що здійснюють обчислення за відповідним модулем. Кінцевою частиною схеми є зворотний перетворювач, який обчислює вихідне двійкове число із залишків, що отримуються в кожному арифметичному каналі. Перетворювачі потребують відповідних витрат на апаратне забезпечення, тому використання СЗК виправдане лише тоді, якщо економія арифметичних каналів перевищує вартість конвертації.

Основними проблемами у СЗК є вибір індивідуальної системної бази, щоб отримати відповідний динамічний діапазон, швидкість та складність схеми. Для необхідного динамічного діапазону слід знайти компроміс між швидкістю арифметичних операцій і складністю перетворення. З одного боку, малі модулі дають змогу створювати невеликі та швидкі арифметичні одиниці, але кількість модулів у базі СЗК і, отже, складність перетворювачів зростають. З іншого боку, арифметичні операції для великих модулів можуть бути доволі повільними. Поєднання малих модулів, великого динамічного діапазону та простих перетворювачів можливе в ієрархічних СЗК, в яких значення всіх або лише деяких залишків представлено в СЗК з динамічними знаками, меншими за діапазон головної системи [44, 183, 184]. Система, яка використовується для відображення значень залишків, називається нижчим рівнем СЗК, тоді як система, числа якої представлені на наступному рівні, називається СЗК більш високого рівня. Таким чином отримується багаторівнева ієрархічна СЗК. Найвища система в ієрархії називається верхнім рівнем СЗК. Динамічні діапазони СЗК нижчого рівня можна вибрати двома способами. У першому підході [44, 183] динамічні діапазони нижнього рівня доволі великі, щоб

відобразити проміжні результати. Наприклад, для множення діапазон нижнього рівня СЗК має дорівнювати квадрату модуля вищого рівня. Перевага цього розв'язку полягає в тому, що ті самі модулі можуть бути використані для відображення різних значень залишків. Наприклад, для множення при найвищому рівні СЗК (17, 19, 20, 21) максимальні значення залишків будуть 16, 18, 19 і 20. Тоді динамічні діапазони для нижнього рівня мають, як мінімум, становити $16^2+1=257$, $18^2+1=325$, $19^2+1=362$ і $20^2+1=401$. Таким чином, СЗК (3, 4, 5, 7) з діапазоном 420 можна використовувати для всіх чотирьох чисел.

Отже, можна побудувати ієрархічну СЗК з діапазоном $17\cdot19\cdot20\cdot21>217$ з 3-бітними модулями. Однак головним недоліком цього розв'язку є швидке зростання динамічних діапазонів СЗК нижчого рівня. Крім того, перетворювачі між рівнями мають використовуватися після невеликої кількості арифметичних операцій.

У другому підході [184] база СЗК верхнього рівня вибирається з чисел, що розкладаються на малі множники. Нижчий рівень СЗК будується з коефіцієнтів для відповідних модулів. У цьому методі діапазон нижчого рівня дорівнює модулям від базового вищого рівня. Таким чином, виконання обчислень у нижчому СЗК ідентичні з їх формуванням за модулем вищих рівнів. Перетворення між послідовними рівнями може бути здійснене раз для всіх арифметичних операцій, що приводить до низьких витрат на апаратне забезпечення. Проте основним недоліком цієї ідеї є труднощі з виявленням базової СЗК найвищого рівня. У науковій праці [184] ця база вибирається з модулів $2^{2k}-1$, а нижній рівень – з 2^k-1, 2^k+1. Це дозволяє реалізовувати перетворювачі та арифметичні одиниці як прості структури. Однак деякі залишки можуть бути близькими до динамічного діапазону системи, тому переваги втрачаються.

У дослідженні [93] запропоновано новий метод побудови бази ієрархічної СЗК. База містить модулі $2^k\pm1$, які розкладаються на малі множники. Цей підхід дає змогу побудувати перетворювачі вводу/виводу як дворівневі схеми, представлені на рис. 1.8. У верхньому рівні здійснюються перетворення між великими числами (близькими до системного діапазону) та залишками за модулем $2^k\pm1$.

У нижньому рівні проводиться перетворення між залишками за модулями $2^k\pm1$ та модулями, що є множниками $2^k\pm1$. Завдяки ефективному виконанню операцій за модулем $2^k\pm1$ для великих чисел область затримки критичного шляху запропонованих дворівневих перетворювачів невелика. Крім того, арифметичні операції виконуються над невеликими числами і, отже, суматори та перемножувачі можуть бути реалізовані відносно простими та швидкими схемами.

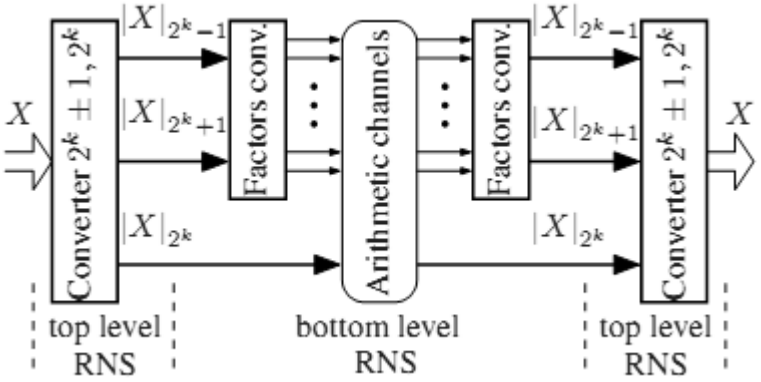


Рис. 1.8. Структура апаратної реалізації ієрархічної СЗК

Через збільшення обсягу обчислень в останній період дослідники почали цікавитися застосуванням СЗК у сучасній асиметричній криптографії [185–187]. Зокрема, в дослідженнях науковців [188–191] представлено безпечні та ефективні підходи щодо застосування СЗК у криптографії на еліптичних кривих. Вони є особливо ефективні як протидія від атак через побічний канал витоку та під час введення несправностей в роботу обчислю-

вальної системи. У працях [192–194] розроблено ефективні алгоритми реалізації RSA-криптографічної системи на основі СЗК, експериментальні дослідження яких підтвердили, що вони мають вищі швидкодію та стійкість до атак грубої сили порівняно з класичними. В дослідженнях [195–200] розроблено та описано методи швидкого виконання арифметичних операцій додавання, множення та піднесення до степеня за модулем у модулярній системі числення при реалізації криптографічних перетворень, в яких показано істотне зменшення часу виконання основних базових операцій криптоалгоритмів.

1.4. Перспективи використання різних форм системи залишкових класів

Використання різних форм СЗК також дає змогу підвищити продуктивність обчислювальних систем при зменшенні їх апаратної складності. У працях [56, 58, 92] запропоновано та обґрунтовано використання чотирьох аналітичних моделей прямих і зворотних перетворень СЗК (табл. 1.6), у т. ч. описаної вище цілочисельної форми.

У табл. 1.6 використано такі позначення: K_k – число у позиційній (двійковій) системі числення; $(b_1, b_2, \ldots, b_i, \ldots, b_k)$ – представлення числа в СЗК; $(p_1, p_2, \ldots, p_i, \ldots, p_k)$ – набір натуральних взаємно простих модулів СЗК; b_i – найменший невід'ємний залишок; P – діапазон кодування чисел у СЗК; a_i – ранг; k – кількість модулів СЗК; B_i – базисні числа СЗК; res – символ операції пошуку найменшого невід'ємного залишку; int – символ операції виділення цілої частини дробового числа; mod – символ операції пошуку залишку за модулем; m_i – ранговий коефіцієнт СЗК; δp – дробова частина в нормалізованій формі СЗК; $[K_k]_0$, $[b_i]_0$ – відповідно позначення числа та залишку в нормалізованій формі СЗК.

Таблиця 1.6

Аналітичні моделі прямих і зворотних перетворень залишкових класів

№	Пряме перетворення форми СЗК	Зворотне перетворення форми СЗК
1.	Цілочисельна форма СЗК	
	$N_k = (b_1 b_2 ... b_i ... b_k)_{(p_1 p_2 ... p_i ... p_k)}$ $b_i = N_k \bmod p_i$, $P = \prod_{i=1}^{k} p_i$; $0 \le N_k \le P-1$	$N_k = \left(\sum_{i=1}^{k} b_i \cdot B_i \right) \bmod P$, $B_i = \frac{P}{p_i} \cdot m_i$; $m_i = \left(\frac{P}{p_i} \right)^{-1} \bmod p_i$
2.	Нормалізована форма СЗК	
	$\frac{N_k}{P} = res \sum_{i=1}^{k} \frac{b_i \cdot B_i (\bmod P)}{P}$, $[N_k]_0 = res \sum_{i=1}^{k} b_i \cdot \frac{B_i}{P} (\bmod 1)$, $0 \le [N_k]_0 \le P-1$; $\frac{B_i}{P} = \frac{1}{p_i}$, $\delta_p \le \frac{1}{P}$, $\frac{1}{p_i} = 0.\overbrace{gggg}^{n_i}\overbrace{gggg}^{\delta_p}$	$[N_k]_0 = res \sum_{i=1}^{k} b_i \cdot \frac{m_i}{p_i} (\bmod 1)$, $[N_k]_0 = res \sum_{i=1}^{k} [b_i]_0 \cdot m_i (\bmod 1)$, $[b_i]_0 = \frac{b_i}{p_i}$, $0 \le [b_i]_0 \le 1$. $N_k = \text{int}[N_k]_0 \cdot P$
3.	Розмежована форма СЗК	
	$N_k = N_{1k} + N_{2k} + ... + N_{ik} +$ $+ ... + N_{nk}$	$N_k =$ → $b_1 = (b_{11} + b_{21} + ... +$ $+ b_{r1} + ... + b_{n1}) \bmod p_1$ $b_2 = (b_{12} + b_{22} + ... +$ $+ b_{r2} + ... + b_{n2}) \bmod p_2$ ... $b_i = (b_{1i} + b_{2i} + ... + b_{ri} +$ $+ ... + b_{ni}) \bmod p_i$ ... $b_k = (b_{1k} + b_{2k} + ... +$ $+ b_{rk} + ... + b_{nk}) \bmod p_k$
4.	Досконала форма СЗК	
	$N_k = (b_1 b_2 ... b_i ... b_k)_{(p_1 p_2 ... p_i ... p_k)}$ $b_i = N_k \bmod p_i$	$N_k = \left(\sum_{i=1}^{k} b_i \cdot B_i \right) \bmod P$, $B_i = \frac{P}{p_i}$; $m_i = 1$

Недоліком звичайної цілочисельної форми СЗК є виникнення значних труднощів при виконанні арифметичної операції порівняння двох чисел, що суттєво ускладнює програмну та апаратну реалізації алгоритмів і відповідних процесів ділення. Крім того, необхідність пошуку мультиплікативного оберненого елемента при переведенні чисел із СЗК у ПСЧ викликає труднощі для застосування цілочисельної СЗК в асиметричних криптосистемах.

Водночас переваги однотактної матричної реалізації інших арифметичних операцій забезпечують широкі перспективи для застосування теоретичних основ цілочисельного перетворення СЗК для створення і широкомасштабного впровадження супершвидкісних процесорів у комп'ютерних системах [71, 72].

Перевагою нормалізованої СЗК є спрощення реалізації процесорів за вилучення нелінійних операцій пошуку залишку за кожним із модулів та заміни операції знаходження залишку mod P на операцію mod 1, яка виконується шляхом відкидання цілої частини результату згідно із операцією int. Однак використання нормалізованої СЗК пов'язане із заокругленням під час ділення на P, що ускладнює її використання в асиметричних криптосистемах.

Математичні операції над числами в розмежованій СЗК можуть бути розподілені за кожним з фрагментів процесора, що забезпечує глибший рівень розпаралелювання обробки інформації і відповідно підвищення швидкодії процесора СЗК. Реалізований такий процесор може бути з суттєвим зменшенням апаратних засобів за кожним із модулів [98, 201]. Ця форма успішно може використовуватися в асиметричних криптосистемах, зокрема при реалізації алгоритму Евкліда, модулярному множенні та експоненціюванні.

Очевидно, що наявність коефіцієнтів $m_i = P_i^{-1} \bmod p_i$ у цілочисельній формі СЗК (формула (1.23)) ускладнює реалізацію відповідного алгоритму. Дослідження різних наборів модулів p_i, яким відповідають коефіцієнти m_i, у теоретико-числовому аспекті підтвердили, що існують такі набори модулів $p_1, p_2, ..., p_k$, яким відповідають одиничні коефіцієнти m_i ($m_1=m_2=...=m_i=...=m_k=1$) (наприклад, 2, 3, 5) або

$$P_i \bmod p_i = 1. \qquad (1.25)$$

Така форма СЗК названа досконалою (ДФ СЗК). Нині ДФ СЗК є особливо перспективною для застосування у сучасних асиметричних криптосистемах. Пошук наборів модулів, що формують ДФ СЗК, є окремою актуальною задачею.

У працях науковців [202–209] розглянуті теоретичні основи побудови ДФ СЗК, яка дає змогу уникнути виконання громіздкої операції пошуку оберненого елемента за модулем та множення на нього згідно з китайською теоремою про залишки. Однак її недоліком є те, що модулі ДФ СЗК дуже швидко зростають, що неприпустимо за необхідності вибору модулів однакової розрядності, зокрема в задачах завадостійкого кодування.

У дослідженнях [210–216] розроблена модифікована ДФ СЗК (МДФ СЗК), в якій виконується умова:

$$P_i \bmod p_i = 1, \ p_i\text{-}1 \ \text{або} \ P_i \bmod p_i = \pm 1, \qquad (1.26)$$

тобто $m_i=1$ або p_i-1 ($m_i=\pm 1$). Це дає змогу усунути недолік ДФ СЗК, а також зменшує ймовірність перевищення діапазону обчислень у формулі (1.8).

Як приклад розглянемо модулярне експоненціювання $a^x \bmod p$, розклавши основу степеня на залишки з

використанням трьохмодулної цілочисельної та МДФ СЗК. Діапазон обчислень $P=p_1 \cdot p_2 \cdot p_3 > p$. Вважаємо, що розрядність модулів p_1, p_2, p_3 приблизно дорівнює третині розрядності p, тобто $\left(\frac{n_0}{3}\right)$, де n_0 – розрядність модуля p. Після пошуку залишків $a \bmod p_1 = \alpha_1$, $a \bmod p_2 = \alpha_2$, $a \bmod p_3 = \alpha_3$ обчислюються значення $\alpha_1^x \bmod p_1$, $\alpha_2^x \bmod p_2$, $\alpha_3^x \bmod p_3$, з яких на основі КТЗ методом виділення квадратів визначається результат модулярного експоненціювання.

Основними операціями при використанні цілочисельної СЗК є знаходження залишків багаторозрядних чисел, пошук оберненого елемента за модулем і піднесення до квадрата за модулем. Відповідно загальна часова складність набуває такого вигляду:

$$O\left(\left(\log_2 3 \cdot (2 \cdot \log_2{}^2 \frac{n_0}{3} + \frac{n_0}{3}\right) + \frac{n_0^2 \cdot 3}{2} + \left(\log_2 {}^{n_0}\!/_{2}\right) + \frac{3n_0^2}{2}\left(\log_2 {}^{n_0}\!/_{6}\right)\right).$$

Якщо вибрані модулі утворюють МДФ СЗК, то часова складність відповідно зменшується:

$$O1\left(\left(\log_2 3 \cdot (2 \cdot \log_2{}^2 \frac{n_0}{3} + \frac{n_0}{3}\right) + \left(\log_2 {}^{n_0}\!/_{2}\right) + \frac{3n_0^2}{2}\left(\log_2 {}^{n_0}\!/_{6}\right)\right).$$

Це відбувається за рахунок усунення операції пошуку оберненого елемента.

РОЗДІЛ 2
ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ ДОСКОНАЛОЇ ФОРМИ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ

2.1. Теоретичні основи аналітичного пошуку коефіцієнтів базисних чисел системи залишкових класів

Як зазначалося в п. 1.3, найбільш поширені та зручні для апаратної реалізації набори спеціалізованих модулів мають вигляд 2^u, $2^u \pm 1$ (табл. 1.4). Однак за умови великої кількості модулів пошук оберненого елемента залишається найбільш обчислювально складною задачею при переведенні чисел із СЗК у позиційну систему числення. Для спрощення цієї операції розглянемо набір модулів у такому вигляді [217]:

$$\begin{cases} p_1=2^u-1; \\ p_2=2^u+1; \\ p_3=2^{2u}+1; \\ p_4=2^{4u}+1; \\ \quad .... \\ p_i = 2^{u\cdot 2^{i-2}}+1; \\ \quad .... \\ p_{k-1} = 2^{u\cdot 2^{k-3}}+1; \\ p_k = 2^{u\cdot 2^{k-2}}+1, \end{cases} \qquad (2.1)$$

де u – степінь двійки в модулі p_1, k – кількість модулів.

Із системи (2.1) можна визначити, що кожен наступний модуль на дві одиниці більший від добутку всіх попередніх. Цим визначається взаємна простота модулів, оскільки всі вони є непарними. Крім того, діапазон розглянутих десяткових чисел для можливих розрахунків обмежується виразом $P = 2^{u \cdot 2^{k-1}} - 1$.

Для пошуку оберненого елемента $m_i = P_i^{-1} \bmod p_i$ запишемо систему рівнянь у такому вигляді:

$$\begin{cases} P_1 \bmod\left(2^u - 1\right) = \left(2^u + 1\right)\left(2^{2u} + 1\right)\left(2^{4u} + 1\right)\ldots\left(2^{u \cdot 2^{i-2}} + 1\right)\ldots\left(2^{u \cdot 2^{k-3}} + 1\right)\left(2^{u \cdot 2^{k-2}} + 1\right) \bmod\left(2^u - 1\right); \\ P_2 \bmod\left(2^u + 1\right) = \left(2^u - 1\right)\left(2^{2u} + 1\right)\left(2^{4u} + 1\right)\ldots\left(2^{u \cdot 2^{i-2}} + 1\right)\ldots\left(2^{u \cdot 2^{k-3}} + 1\right)\left(2^{u \cdot 2^{k-2}} + 1\right) \bmod\left(2^u + 1\right); \\ P_3 \bmod\left(2^{2u} + 1\right) = \left(2^{2u} - 1\right)\left(2^{4u} + 1\right)\ldots\left(2^{u \cdot 2^{i-2}} + 1\right)\ldots\left(2^{u \cdot 2^{k-3}} + 1\right)\left(2^{u \cdot 2^{k-2}} + 1\right) \bmod\left(2^{2u} + 1\right); \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ P_i \bmod\left(2^{u \cdot 2^{i-2}} + 1\right) = \left(2^{u \cdot 2^{i-2}} - 1\right)\ldots\ldots\left(2^{u \cdot 2^{k-3}} + 1\right)\left(2^{u \cdot 2^{k-2}} + 1\right) \bmod\left(2^{u \cdot 2^{i-2}} + 1\right); \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ P_{k-1} \bmod\left(2^{u \cdot 2^{k-3}} + 1\right) = \left(2^{u \cdot 2^{k-3}} - 1\right)\left(2^{u \cdot 2^{k-2}} + 1\right) \bmod\left(2^{u \cdot 2^{k-3}} + 1\right); \\ P_k \bmod\left(2^{u \cdot 2^{k-2}} + 1\right) = \left(2^{u \cdot 2^{k-2}} - 1\right) \bmod\left(2^{u \cdot 2^{k-2}} + 1\right). \end{cases} \quad (2.2)$$

У першому рівнянні (2.2) для кожного множника правої частини отримується залишок 2, тому $P_1 \bmod\left(2^u - 1\right) = 2^{k-1} \bmod\left(2^u - 1\right)$. В другому рівнянні (2.2) залишок від першого множника дорівнює –2, всі інші становлять 2, тому $P_2 \bmod\left(2^u + 1\right) = -2^{k-1} \bmod\left(2^u + 1\right)$.

У всіх інших рівняннях, аналогічно до другого, перший залишок становить –2, інші дорівнюють 2, причому із збільшенням номера рівняння на 1 кількість множників (відповідно і двійок) зменшується також на 1. У результаті таких обчислень отримаємо систему:

$$\begin{cases} P_1 \bmod \left(2^u - 1\right) = 2^{k-1} \bmod \left(2^u - 1\right); \\ P_2 \bmod \left(2^u + 1\right) = -2^{k-1} \bmod \left(2^u + 1\right); \\ P_3 \bmod \left(2^{2u} + 1\right) = -2^{k-2} \bmod \left(2^{2u} + 1\right); \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ P_i \bmod \left(2^{u \cdot 2^{i-2}} + 1\right) = -2^{k-(i-1)} \bmod \left(2^{u \cdot 2^{i-2}} + 1\right); \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ P_{k-1} \bmod \left(2^{u \cdot 2^{k-3}} + 1\right) = -4; \\ P_k \bmod \left(2^{u \cdot 2^{k-2}} + 1\right) = -2. \end{cases} \quad (2.3)$$

Тепер шукаємо величини $m_i = P_i^{-1} \bmod p_i$. У першому рівнянні степінь двійки в правій частині запишемо в такому вигляді: $k - 1 = a_1 u + k_1$, де $0 \le k_1 < u$.

Тоді $2^{k-1} \bmod (2^u - 1) = (2^u)^a \cdot 2^{k_1} \bmod (2^u - 1) = 2^{k_1} \bmod (2^u - 1)$. Звідси

$$m_1 = \frac{2^u}{2^{k_1}} = 2^{u-k_1}. \quad (2.4)$$

Аналогічно отримуємо з другого рівняння: $k - 1 = a_2 u + k_2$. Тоді:

$$-2^{k-1} \bmod (2^u + 1) = -(2^u)^{a_2} \cdot 2^{k_2} \bmod (2^u + 1) = -(-1)^{a_2} \cdot 2^{k_2} \bmod (2^u + 1) =$$

$$= (-1)^{a_2+1} \cdot 2^{k_2} \bmod (2^u + 1) = \begin{cases} 2^{k_2} \bmod (2^u + 1), \ a_2 - \text{непарне}; \\ -2^{k_2} \bmod (2^u + 1), \ a_2 - \text{парне}. \end{cases}$$

Розглянемо два можливих випадки:

а) a_2 – непарне; до 2^u+2 потрібно додати модуль 2^u+1 стільки разів, щоб сума ділилась на 2^{k_2} (тобто другий доданок має дорівнювати 2^{k_2}). Отже,

$$2^u+2+\left(2^u+1\right)\left(2^{k_2}-2\right)=2^u+2+2^{u+k_2}+2^{k_2}-2\cdot 2^u-2=2^{u+k_2}-2^u+2^{k_2}.$$

Поділивши на 2^{k_2}, отримаємо обернений елемент: $m_2=2^u-2^{u-k_2}+1$;

б) a_2 – парне; отримане значення m_2 потрібно записати з протилежним знаком і додати модуль: $m_2=-\left(2^u-2^{u-k_2}+1\right)\bmod\left(2^u+1\right)=2^{u-k_2}$.

Отже, остаточна формула матиме вигляд:

$$m_2=\begin{cases}2^u-2^{u-k_2}+1, & a_2-\text{непарне};\\ 2^{u-k_2}, & a_2-\text{парне}.\end{cases} \tag{2.5}$$

Аналогічно отримуємо з третього рівняння:

$$m_3=\begin{cases}2^{2u}-2^{2u-k_3}+1, & a_3-\text{непарне};\\ 2^{2u-k_3}, & a_3-\text{парне},\end{cases} \tag{2.6}$$

де k_3 і a_3 визначаються з рівності $k-2=2ua_3+k_3$.

З i-го рівняння:

$$m_i=\begin{cases}2^{u\cdot 2^{i-2}}-2^{u\cdot 2^{i-2}-k_i}+1, & a_i-\text{непарне};\\ 2^{u\cdot 2^{i-2}-k_i}, & a_i-\text{парне},\end{cases} \tag{2.7}$$

де k_i та a_i визначаються з рівності $k-(i-1)=2^{i-2}ua_i+k_i$.

Розглянемо (k-1) -е рівняння. Немає необхідності розписувати k_{i-1}, оскільки $2^2<2^{u\cdot 2^{k-3}}$. До $2^{u\cdot 2^{k-3}}+2$ двічі потрібно додати модуль і поділити на -4. В результаті отримаємо:

$$m_{k-1}=2^{u\cdot 2^{k-3}-2}. \tag{2.8}$$

Аналогічно отримуємо для останнього, *k*-го рівняння $\left(2^{u\cdot 2^{k-2}}+2\right)\Big/(-2)=-\left(2^{u\cdot 2^{k-2}-1}+1\right)$. Додавши модуль, отримаємо:

$$m_k = 2^{u\cdot 2^{k-2}-1}. \qquad (2.9)$$

У табл. 2.1 наведені значення p_i, P_i, m_i, а також діапазон можливих розрахунків для k=4 та різних значень u.

Таблиця 2.1

Значення p_i, P_i, m_i, а також діапазон можливих розрахунків для k=4 та різних значень u

u	p_1	P_1	m_1	p_2	P_2	m_2	p_3	P_3	m_3	p_4	P_4	m_4	P
2	2^2-1	1	1	2^2+1	4	4	2^4+1	2^4-7	2	2^8+1	2^8-3	2^6	$2^{32}-1$
3	2^3-1	2	4	2^3+1	2	5	2^6+1	2^6-7	2^3	$2^{12}+1$	$2^{12}-3$	2^{10}	$2^{48}-1$
4	2^4-1	1	1	2^4+1	1	1	2^8+1	2^8-7	2^5	$2^{16}+1$	$2^{16}-3$	2^{14}	$2^{64}-1$
5	2^5-1	16	2	2^5+1	2^5-15	2	$2^{10}+1$	$2^{10}-7$	2^7	$2^{20}+1$	$2^{20}-3$	2^{18}	$2^{80}-1$
6	2^6-1	16	2^2	2^6+1	2^6-15	2^2	$2^{12}+1$	$2^{12}-7$	2^9	$2^{24}+1$	$2^{24}-3$	2^{22}	$2^{96}-1$
…	…	…	…	…	…	…	…	…	…	…	…	…	…
i	2^i-1	16	2^{i-4}	2^i+1	2^i-15	2^{i-4}	$2^{2i}+1$	$2^{2i}-7$	2^{2i-3}	$2^{4i}+1$	$2^{4i}-3$	2^{4i-2}	$2^{16i}-1$

За даними табл. 2.1. величини P_1, m_1, P_2, m_2 при малих u можуть набувати різних значень, що залежить від парності або непарності коефіцієнта a_i. Інші значення P_i, m_i мають вигляд відповідного степеня двійки.

На рис. 2.1 показано логарифмічну залежність степеня u для модуля p_1 від кількості модулів k для 512-розрядного процесора згідно з виразом $u = 2^{10-k}$.

Відповідно можна зробити висновок, що $\log_2 u$ лінійно спадає із збільшенням кількості модулів k.

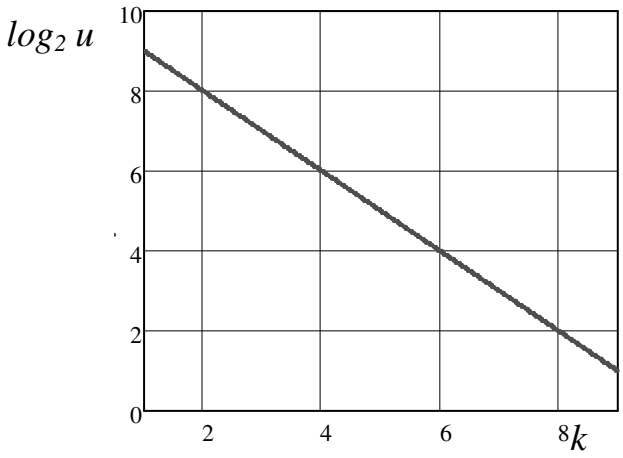


Рис. 2.1. Графік логарифмічної залежності степеня *u* для модуля p_1 від кількості модулів *k*

2.2. Метод побудови досконалої форми системи залишкових класів на основі дробових перетворень

Для побудови набору модулів ДФ СЗК запишемо вираз (1.25) у вигляді такої системи:

$$\begin{cases} P_1 \bmod p_1 = 1 \\ \dots \\ P_k \bmod p_k = 1 \end{cases}. \qquad (2.10)$$

Домноживши кожне рівняння на відповідний модуль, отримаємо:

$$\begin{cases} P \bmod p_1^2 = p_1 \\ \dots \\ P \bmod p_k^2 = p_k. \end{cases} \qquad (2.11)$$

Розв'язуючи систему (2.11) стандартними методами теорії чисел згідно з китайською теоремою про залишки, матимемо:

$$P = \left(\sum_{i=1}^{k} p_i P_i^2 m_i^2 \right) \bmod M \,, \qquad (2.12)$$

де $M = \prod_{i=1}^{k} p_i^2 = P^2$.

Врахувавши, що у ДФ СЗК коефіцінти m_i=1, та скоротивши модуль, ліву та праву частини виразу (2.12) на їхній спільний дільник $P=\prod_{i=1}^{k} p_i$, запишемо рівність (2.12) таким чином:

$$\left(\sum_{i=1}^{k} P_i \right) \bmod P = 1. \qquad (2.13)$$

Вираз (2.13) еквівалентний рівності:

$$\sum_{i=1}^{k} P_i = \gamma P + 1, \qquad (2.14)$$

де γ=1, 2, 3, ...

Поділивши ліву та праву частини рівності (2.14) на P, отримаємо остаточний вираз для пошуку набору модулів у ДФ СЗК:

$$\sum_{i=1}^{k} \frac{1}{p_i} = \gamma + \frac{1}{\prod_{i=1}^{k} p_i}. \qquad (2.15)$$

Розглянемо систему з трьох модулів. Подамо модулі p_2, p_3 у такому вигляді: $p_2=ap_1+b$, $p_3=cp_1p_2+d=cp_1(ap_1+b)+d$, де a і c – натуральні, b і d – цілі числа, причому $|b| < p_1$, $|d| < p_2$. Тоді з формули (2.10) отримуємо:

$$\begin{cases} \left(\left(ap_1+b\right)\left(cp_1\left(ap_1+b\right)+d\right)\right)\bmod p_1=1 \\ \left(p_1\left(cp_1\left(ap_1+b\right)+d\right)\right)\bmod\left(ap_1+b\right)=1. \\ \left(p_1\left(ap_1+b\right)\right)\bmod\left(cp_1\left(ap_1+b\right)+d\right)=1 \end{cases} \quad (2.16)$$

З третього рівняння (2.16) добуток $p_1(ap_1+b)$ має або дорівнювати 1, або бути на одиницю більшим за значення модуля $cp_1(ap_1+b)+d$. Звідси випливає, що *c*=1, *d*=–1. З другого рівняння визначаємо, що $(p_1(ap_1+b)-1)\bmod(ap_1+b)=-1$, тому має виконуватися умова $p_1 \bmod(ap_1+b)=-1$. Це можливо тільки у разі, коли *a*=1, *b*=1. Тоді система (2.16) набуде такого вигляду:

$$\begin{cases} \left(\left(p_1+1\right)\left(p_1\left(p_1+1\right)-1\right)\right)\bmod p_1=1 \\ \left(p_1\left(p_1\left(p_1+1\right)-1\right)\right)\bmod\left(p_1+1\right)=1. \\ \left(p_1\left(p_1+1\right)\right)\bmod\left(p_1\left(p_1+1\right)-1\right)=1 \end{cases} \quad (2.17)$$

З першого рівняння системи (2.17) визначаємо, що $(p_1(p_1+1)-1)\bmod p_1=-1$. Це приводить до умови (p_1+1) mod p_1= –1. Такий випадок можливий тільки тоді, коли p_1=2. Це число є унікальним для ДФ СЗК, оскільки –1 mod 2=1.

Це підтверджує, що єдино можливим набором для трьох модулів ДФ СЗК можуть бути тільки числа 2, 3, 5. При збільшенні будь-якого p_i ліва частина рівності (2.15) стає меншою від 1.

Обчислення рівняння (2.15) для великої кількості модулів, враховуючи, що сума ряду $\sum_{i=1}^{k}\frac{1}{p_i}$ розбіжна, тобто значення γ може бути наскільки можливо великим, є доволі громіздкою задачею. Вона значно спрощується, коли взяти p_1=2, p_2=3, оскільки в цьому разі, згідно з рівністю (2.10), для взаємно простих модулів $P_1 \bmod 2 = 1$, $P_2 \bmod 3 = \pm 1$. Крім того, будь-який модуль можна представити у вигляді $p_i = 6t \pm 1$, де t – натуральне число. Припустивши, що γ=1, перепишемо рівність (2.15) у такому вигляді:

$$\sum_{i=3}^{k}\frac{1}{p_i} = \frac{1}{6} + \frac{1}{\prod_{i=3}^{k} p_i}. \qquad (2.18)$$

Модуль p_3 виберемо так, щоб при відніманні $\frac{1}{p_3}$ у правій частині рівності (2.18) в чисельнику отримати 1. Визначаємо, що p_3=7.Тоді маємо:

$$\sum_{i=4}^{k}\frac{1}{p_i} = \frac{1}{42} + \frac{1}{\prod_{i=4}^{k} p_i}. \qquad (2.19)$$

Аналогічно звідси отримаємо p_4=43:

$$\sum_{i=5}^{k}\frac{1}{p_i} = \frac{1}{1806} + \frac{1}{\prod_{i=5}^{k} p_i}. \qquad (2.20)$$

Для останнього модуля p_k справедлива рівність:

$$\frac{1}{p_k} = \frac{1}{\prod_{i=1}^{k-1} p_i} + \frac{1}{p_k \cdot \prod_{i=1}^{k-1} p_i}. \qquad (2.21)$$

Звідси отримуємо:

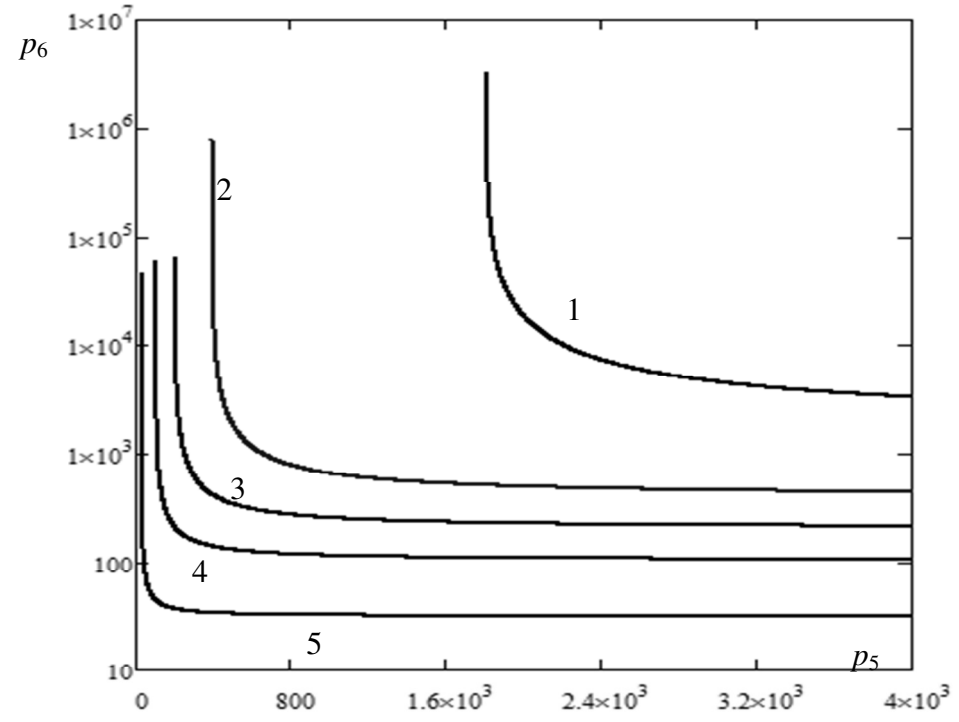
$$p_k = \prod_{i=1}^{k-1} p_i - 1. \qquad (2.22)$$

На основі цього визначаємо закономірність побудови системи модулів ДФ СЗК:

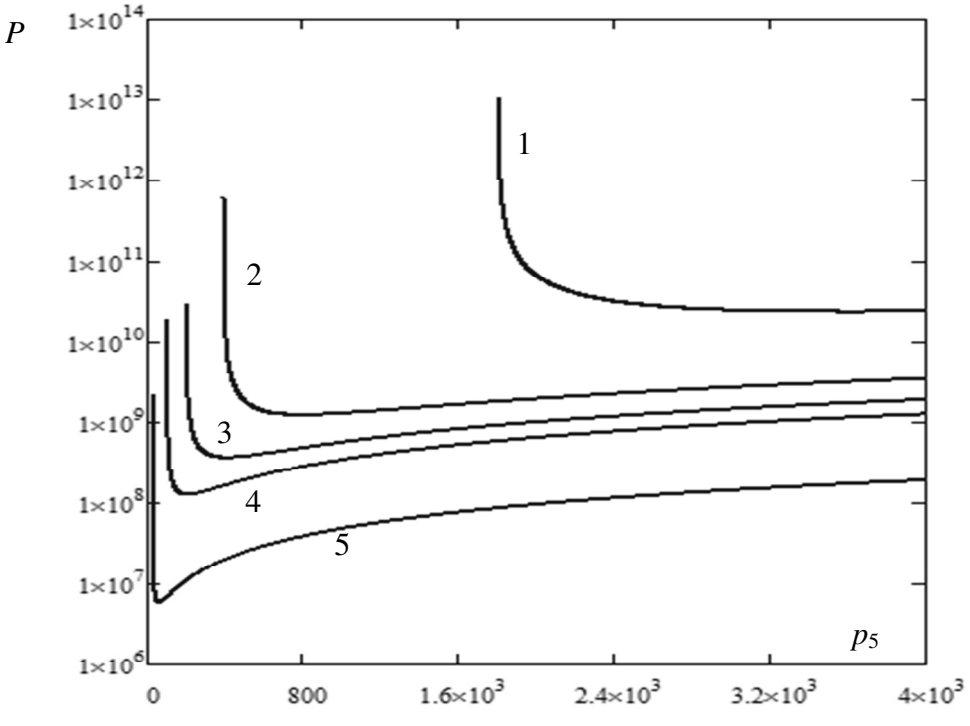
$$\begin{cases} p_1 = 2 \\ p_i = p_1 p_2 \ldots p_{i-1} + 1,\ 1 < i < k. \\ p_k = p_1 p_2 \ldots p_{k-1} - 1 \end{cases} \qquad (2.23)$$

На рис. 2.2а відповідно до формули (2.18) подано графіки залежності модуля p_6 від p_5, для яких потрібно вибрати цілочисельні значення, при різних p_3 та p_4. На рис. 2.2б зображена відповідні діапазони, в межах яких можна виконувати арифметичні операції над десятковими числами.

За графіками визначаємо, що модуль p_6 різко зменшується, після чого набуває постійного значення. Діапазон обчислень при певному значенні p_5 має характерний мінімум, глибина якого збільшується із зменшенням діапазону обчислень.



а)



б)

Рис. 2.2. Графіки залежності p_6 від p_5: а) та відповідні діапазони десяткових чисел; б) 1 – p_3=7, p_4=43; 2 – p_3=7, p_4=47; 3 – p_3=7, p_4=53; 4 – p_3=7, p_4=71; 5 – p_3=11, p_4=23

2.3 Узагальнення методу дробових перетворень

Узагальнюючи вираз (2.23) і враховуючи, що для зменшення складності обчислень якомога більша кількість коефіцієнтів m_i має дорівнювати 1, представимо набір модулів у такому вигляді:

$$\begin{cases} p_1; \\ p_2 = p_1 + 1; \\ \ldots\ldots\ldots \\ p_i = p_1 \cdot p_2 \ldots p_{i-1} + 1; \\ \ldots\ldots\ldots \\ p_k = p_1 \cdot p_2 \ldots p_{k-1} - 1. \end{cases} \tag{2.24}$$

Для знаходження m_i використаємо таку систему рівнянь:

$$\begin{cases} P_k \bmod p_k = (p_1 \cdot p_2 \ldots p_{k-1}) \bmod (p_1 \cdot p_2 \ldots p_{k-1} - 1) = 1 = m_k; \\ P_{k-1} \bmod p_{k-1} = (p_1 \cdot p_2 \ldots p_{k-2} \cdot p_k) \bmod (p_1 \cdot p_2 \ldots p_{k-2} + 1) = \\ = (-1) \cdot (-1) = 1 = m_{k-1}; \\ \ldots\ldots\ldots\ldots\ldots \\ P_i \bmod p_i = \left(p_1 \cdot p_2 \ldots p_{i-1} \cdot p_{i+1 \ldots} \cdot p_k \right) \bmod (p_1 \cdot p_2 \ldots p_{i-1} + 1) = \\ = (-1) \cdot 1 \cdot 1 \cdot \ldots \cdot (-1) = 1 = m_i; \\ \ldots\ldots\ldots\ldots\ldots \\ P_2 \bmod p_2 = (p_1 \cdot p_3 \ldots \cdot p_k) \bmod (p_1 + 1) = (-1) \cdot 1 \cdot 1 \cdot \ldots \cdot (-1) = 1 = m_2; \\ P_1 \bmod p_1 = (p_2 \cdot p_3 \ldots \cdot p_k) \bmod p_1 = 1 \cdot 1 \cdot \ldots \cdot (-1) = -1 = m_1. \end{cases} \tag{2.25}$$

Із системи (2.25) визначимо, що всі m_i, крім m_1=–1, дорівнюють 1. Якщо вибрати p_1=2, то система (2.25) переходить у систему (2.23), оскільки $1 \bmod 2 = -1 \bmod 2$.

Слід зазначити, що запропонований у системі (2.23) метод не вичерпує всіх можливих наборів для ДФ СЗК при заданих k. Наприклад, при k=5 набір модулів, отриманий за допомогою системи (2.23), буде такий: 2, 3, 7, 43, 1805. Однак відомі також набори 2, 3, 7, 83, 85 та 2, 3, 11, 17, 59.

Усі можливі набори модулів для ДФ СЗК при k=6, відповідні їм діапазони десяткових чисел і розрядності в двійковій системі числення (в дужках) подані в табл. 2.2.

Таблиця 2.2

Можливі набори модулів при k=6 для ДФ СЗК і відповідні їм діапазони десяткових чисел (в дужках – розрядність)

№	p_1, p_2	p_3	p_4	p_5	p_6	P
1	2, 3 (2)	7 (3)	43 (6)	1807 (11)	3263441 (22)	$1{,}0650050423922\times10^{13}$ (44)
2				1811 (11)	654133 (20)	$2{,}139450562578\times10^{12}$ (41)
3				1819 (11)	252701 (18)	$8{,}30151592914\times10^{11}$ (41)
4				1825 (11)	173471 (18)	$5{,}7175174245\times10^{11}$ (40)
5				1871 (11)	51985 (16)	$1{,}7565866661\times10^{11}$ (38)
6				1901 (11)	36139 (16)	$1{,}24072631634\times10^{11}$ (37)
7				1945 (11)	25271 (15)	$8{,}876868357\times10^{10}$ (37)
8				2053 (12)	15011 (14)	$5{,}5656554898\times10^{10}$ (36)
9				2167 (12)	10841 (14)	$4{,}2427359282\times10^{10}$ (36)
10				2501 (12)	6499 (13)	$2{,}9354722194\times10^{10}$ (35)
11				3041 (12)	4447 (13)	$2{,}4423128562\times10^{10}$ (35)
12				3611 (12)	3613 (12)	$2{,}3562056658\times10^{10}$ (35)
13			47 (6)	395 (9)	779729 (20)	$6{,}0797809317\times10^{10}$ (36)
14				481 (9)	2203 (12)	$2{,}091735282\times10^{9}$ (31)
15			53 (6)	271 (9)	799 (10)	$4{,}81993554\times10^{8}$ (29)
16			71 (7)	103 (7)	61429 (16)	$1{,}8867671634\times10^{10}$ (35)
17		11(4)	23 (5)	31 (5)	47057 (16)	$2{,}214408306\times10^{9}$ (32)

Згідно з даними табл. 2.2. розрядність чисел, над якими виконуються обчислення, зменшується приблизно в 2–3 рази. Величина *P* є максимальною при використанні набору модулів, отриманого за допомогою системи (2.23), що дозволяє розглядати найбільший діапазон десяткових чисел. Розрядність чисел при цьому зменшується вдвічі.

2.4. Побудова досконалої форми системи залишкових класів методом факторизації

У загальному випадку при $\gamma \neq 1$ вираз (2.15) матиме вигляд:

$$\sum_{i=1}^{k} \frac{1}{p_i} = 1 + \frac{1}{\prod_{i=1}^{k} p_i}. \qquad (2.26)$$

Вважаючи невідомими два останні модулі, після відповідних математичних перетворень отримаємо таку формулу:

$$\left(p_{k-1} + p_k\right) \prod_{i=1}^{k-2} p_i + p_{k-1} p_k \left(\sum_{i=1}^{k-2} \frac{P}{p_i} - \prod_{i=1}^{k-2} p_i \right) = 1. \qquad (2.27)$$

Введемо заміну:

$$p_{k-1}, p_k = \frac{a,b - \prod_{i=1}^{k-2} p_i}{\sum_{i=1}^{k-2} \frac{P}{p_i} - \prod_{i=1}^{k-2} p_i}. \qquad (2.28)$$

Після підстановки рівності (2.28) у формулу (2.27) отримується умова, яка має виконуватися для визначення набору модулів ДФ СЗК:

$$ab = \sum_{i=1}^{k-2} \frac{P}{p_i} - \prod_{i=1}^{k-2} p_i + \left(\prod_{i=1}^{k-2} p_i \right)^2. \qquad (2.29)$$

Це означає, що ліва частина рівності (2.29) має бути факторизована, на основі чого визначаються параметри *a* та *b*. Крім того, як випливає з формули (2.28), модулі p_k та p_{k-1} мають бути цілими числами, тобто

$$\left(a,b - \prod_{i=1}^{k-2} p_i \right) \bmod \left(\sum_{i=1}^{k-2} \frac{P}{p_i} - \prod_{i=1}^{k-2} p_i \right) = 0. \qquad (2.30)$$

Отже, вирази (2.29) та (2.30) визначають умови для знаходження будь-якої кількості модулів ДФ СЗК, два з яких невідомі.

Нехай n=6. Відповідно відомі тільки такі набори модулів ДФ СЗК, в яких p_1=2, p_2=3. Тоді вираз (2.26) можна переписати так:

$$\sum_{i=3}^{6} \frac{1}{p_i} = \frac{1}{6} + \frac{1}{\prod_{i=3}^{6} p_i}. \qquad (2.31)$$

Далі з виразів (2.27) та (2.31) маємо:

$$6p_3p_4(p_5 + p_6) = (p_4(p_3 - 6) - 6p_3)p_5p_6 + 1. \qquad (2.32)$$

Введемо позначення:

$$p_{5,6} = \frac{6p_3p_4 + a,b}{p_4(p_3 - 6) - 6p_3}. \qquad (2.33)$$

Після підстановки рівності (2.33) у формулу (2.32) матимемо:

$$(6p_3p_4)^2 - (p_4(p_3 - 6) - 6p_3) = ab. \qquad (2.34)$$

Ліва частина рівності (2.34) має бути факторизована, на основі чого визначаються параметри *a* та *b*. Крім того, як випливає з формули (2.33), модулі p_5 та p_6 є цілочисельними, тобто:

$$(6p_3p_4 + a,b) \bmod (p_4(p_3 - 6) - 6p_3) = 0. \qquad (2.35)$$

Отже, вирази (2.34) та (2.35) визначають умови знаходження будь-якого варіанта набору з шести модулів ДФ СЗК.

2.4.1. Часткові випадки

Перевіривши можливі значення p_3, можна визначити, що цей модуль може дорівнювати 7 або 11. Розглянемо ці випадки детальніше:

1) p_3=7. Вирази (2.34) та (2.35) відповідно трансформуються:

$$(42p_4)^2 - (p_4 - 42) = ab; \qquad (2.36)$$

$$(42p_4 + a,b) \bmod (p_4 - 42) = 0. \qquad (2.37)$$

Модуль p_4 має бути не менший від 43, оскільки набір 2, 3, 7, 41 утворює ДФ СЗК. Тоді умова (2.37) виконується завжди, а з першої можна отримати:

$$ab = (42 \cdot 43)^2 - 1 = 1805 \cdot 1807 = 5 \cdot 19 \cdot 19 \cdot 13 \cdot 139. \qquad (2.38)$$

Використавши всі можливі перестановки множників у (2.38), можна отримати 12 варіантів наборів з 6 модулів ДФ СЗК при заданих модулях 2, 3, 7, 43, які представлені в таблиці 3.4.

Таблиця 2.3

Можливі варіанти наборів з 6 модулів ДФ СЗК при заданих модулях 2, 3, 7, 43

№	*a*	*b*	p_5	p_6
1	1	5·19·19·13·139	1807	3263441
2	5	19·19·13·139	1811	654133
3	13	5·19·19·139	1819	252701
4	19	5·19·13·139	1825	173471
5	5·13	19·19·139	1871	51985
6	5·19	19·13·139	1901	36139
7	139	5·19·19·13	1945	25271
8	19·13	5·19·139	2053	15011
9	19·19	5·13·139	2167	10841
10	5·139	19·19·13	2501	6499
11	5·13·19	19·139	3041	4447
12	5·19·19	13·139	3611	3613

На рис. 2.3 відображено зміни значень модулів p_5 та p_6 залежно від номера модуля згідно з даними табл. 2.3 у логарифмічній шкалі. Як можна визначити, модуль p_5 зростає повільно. Водночас крива графіка для p_6 істотно спадає із збільшенням номера модуля.

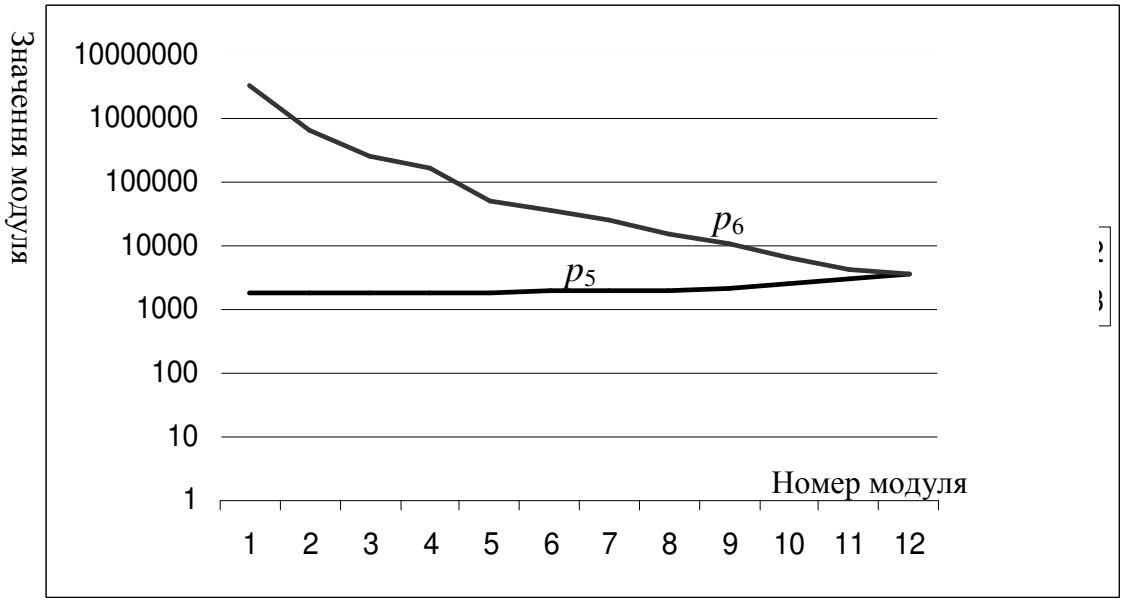


Рис. 2.3. Характер зміни значень модулів p_5 та p_6 залежно від номера модуля (згідно з даними табл. 2.3)

При p_4=47 отримаємо: $p_{5,6} = \dfrac{1974 + a,b}{5}$,

$ab = (42 \cdot 47)^2 - 5 = 9041 \cdot 431$. Можливі два варіанти, які подано у табл. 2.4.

Таблиця 2.4

Можливі варіанти наборів з 6 модулів ДФ СЗК при заданих модулях 2, 3, 7, 47

№	*a*	*b*	p_5	p_6
1	1	9041·431	395	779729
2	431	9041	481	2203

При p_4=53 рівності (2.29) та (2.30) набудуть вигляду: $p_{5,6} = \dfrac{2226 + a,b}{11}$, $ab = (42 \cdot 53)^2 - 11 = 5 \cdot 151 \cdot 6563$. Оскільки 2226 mod 11=4, 5 mod 11=5, 151 mod 11=8, 6563 mod 11=7, то

умову (2.30) задовольняє тільки значення a=5·151. Відповідно b=6563 і p_5=271, p_6=799.

Аналогічно можна знайти тільки один набір модулів: p_4=71, p_5=103, p_6=61429;

2) p_3=11. Вирази (2.29) та (2.30) набудуть відповідно такого вигляду:

$$(66p_4)^2 - (5p_4 - 66) = ab; \qquad (2.39)$$

$$(66p_4 + a, b) \bmod (5p_4 - 66) = 0. \qquad (2.40)$$

Умови (2.39) і (2.40) задовольняють такі значення: p_4=23, p_5=31, p_6=47057.

Отже, всі значення елементів у табл. 2.2, отримані методом підбору, обчислені за допомогою аналітичних розрахунків.

2.5. Застосування досконалої форми системи залишкових класів у китайській теоремі про залишки

ДФ СЗК успішно може використовуватись в асиметричних криптосистемах, зокрема у криптосистемі Рабіна, яка ґрунтується на застосуванні КТЗ, яка зводиться до розв'язання системи конгруенцій (1.7).

Розв'язок такої системи подано у формулі (1.8). Як зазначалося, пошук обернених елементів для коефіцієнтів $m_i = P_i^{-1} \bmod p_i$ становить значну обчислювальну складність. Однак якщо модулі p_1, p_2, … , p_k утворюють ДФ СЗК, тоді можна уникнути цієї громіздкої операції.

Нехай p_1=2, p_2=3, p_3=7, p_4=43, p_5=3611, p_6=3613 і потрібно розв'язати таку систему конгруенцій:

$$\begin{cases} x \bmod 2 = 1; \\ x \bmod 3 = 2; \\ x \bmod 7 = 5; \\ x \bmod 43 = 20; \\ x \bmod 3611 = 100; \\ x \bmod 3613 = 1000. \end{cases} \quad (2.41)$$

У загальному випадку $x = \left(\sum_{i=1}^{6} r_i P_i m_i \right) \bmod P$, де $m_i = P_i^{-1} \bmod p_i$. У ДФ СЗК $m_i = 1$, звідси: x=(3·7·43·3611·3613·1 + +2·7·43·3611·3613·2 + 2·3·43·3611·3613·5 + 2·3·7·3611·3613·20 + +2·3·7·43·3613·100 + 2·3·7·43·3611·1000) mod 5225745 = = (11781028329 + 15708037772 + 16830040470 + 10959096120 + +652507800 + 6521466000) mod 23562056658 = =(62452176491) mod 23562056658 = 15328063175.

Отже, шукане значення x отримане за допомогою КТЗ без виконання громіздкої операції пошуку оберненого елемента за модулем, а використовуючи додавання та множення.

При перетвореннях згідно з КТЗ використовуються такі основні модульні операції: пошук оберненого елемента; пошук залишків; операції множення і додавання. Відповідно при визначенні обчислювальних складностей відомого і запропонованого методів, які дозволяють виконувати перетворення згідно з КТЗ, потрібно врахувати складності всіх вищезазначених операцій, які наведені в табл. 2.5 (f – кількість взаємно простих модулів).

Таблиця 2.5

Часові складності основних операцій КТЗ

№	Основні операції	Часова складність операцій у запропонованому методі	Часова складність операцій у класичному методі
1.	Оберне-ний елемент	Відсутня	$O1(17{,}5f\cdot((n_0+1)^2+n_0^2+n_0)$
2.	Залишки	$O\left(\log_2 {n_0}/{2}\right)$	$O1((n_0+1)^2+n_0)$
3.	Множен-ня і дода-вання	$O\left(\log_2 f\cdot(2\cdot\log_2^2 n_0+n_0\right)$	$O1(f\cdot(2n_0^2+n_0))$

Враховуючи дані табл. 2.5, часова складність КТЗ із використанням запропонованого методу становитиме $O\left(\left(\log_2 f\cdot(2\cdot\log_2^2 n_0+n_0\right)+\left(\log_2 {n_0}/{2}\right)\right)\approx O(\log_2 f\cdot(2\cdot\log_2{}^2 n_0+n_0))$, а із застосуванням класичного методу – $O1(37f\cdot n_0^2+53{,}5f\cdot n_0+17{,}5f+n_0^2+3n_0+1))\approx O1(37f\cdot n_0^2)$. На рис. 2.4 зображено графіки залежності визначених часових складностей від розрядності чисел n_0 у логарифмічній шкалі.

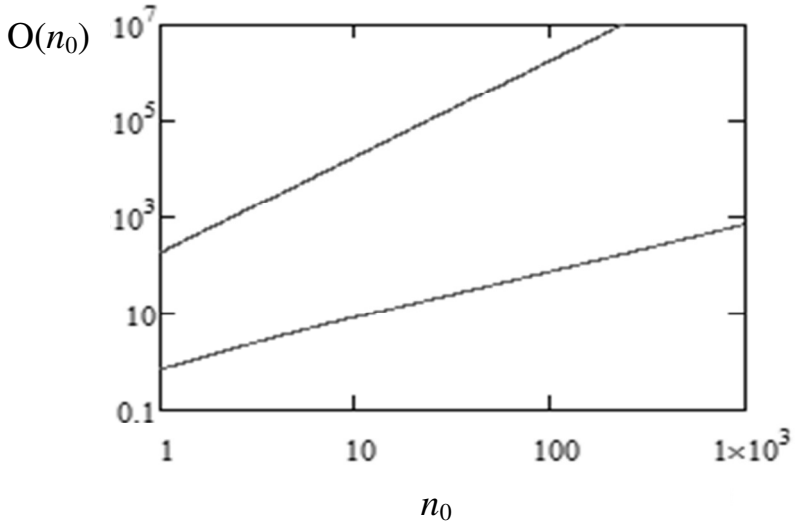


Рис. 2.4. Графіки залежності обчислювальних складностей від розрядності n_0 запропонованим методом $O(n_0)$ та класичним $O1(n_0)$

Отже, використання запропонованого методу, який дає змогу аналітично обчислювати модулі ДФ СЗК та уникати операції пошуку оберненого елемента за модулем, істотно зменшує обчислювальну складність КТЗ відносно класичного.

РОЗДІЛ 3
МЕТОДИ ПОБУДОВИ ТРЬОХМОДУЛЬНОЇ МОДИФІКОВАНОЇ ДОСКОНАЛОЇ ФОРМИ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ

3.1. Метод перемноження модулів

У разі обмеженої кількості модулів і необхідності розгляду великих чисел, згідно з формулою (1.26), зручно підібрати такий набір модулів, щоб вони утворювали МДФ СЗК:

$$m_i = P_i^{-1} \bmod p_i = \pm 1. \qquad (3.1)$$

Порівняно із ДФ СЗК це збільшує обчислювальну складність, але вона менша, ніж при пошуку оберненого елемента за модулем.

Запропонований метод дає змогу побудувати систему із двох модулів, що неможливо в ДФ СЗК. Для цього необхідно вибрати будь–які два послідовні числа p_1 і $p_2=p_1+1$, які будуть завжди взаємно простими, і для них завжди виконується умова:

$$\begin{cases} \left(p_1+1\right) \bmod p_1 = 1; \\ p_1 \bmod \left(p_1+1\right) = -1. \end{cases} \qquad (3.2)$$

Для дослідження набору з трьох модулів запишемо систему, аналогічну до системи (2.16):

$$\begin{cases} \left(\left(ap_1+b\right)\left(cp_1\left(ap_1+b\right)+d\right)\right) \bmod p_1 = \pm 1; \\ \left(p_1\left(cp_1\left(ap_1+b\right)+d\right)\right) \bmod \left(ap_1+b\right) = \pm 1; \\ \left(p_1\left(ap_1+b\right)\right) \bmod \left(cp_1\left(ap_1+b\right)+d\right) = \pm 1. \end{cases} \quad (3.3)$$

Аналізуючи систему (3.3), подібно до виразу (2.16), і вважаючи, що $|p_1|<|p_2|<|p_3|$, отримаємо: $a=c=b=1$, $d=\pm 1$. В цьому разі ліва частина останньої системи спрощується, а права залежатиме від знака біля коефіцієнта *d*. Можна простежити, що

$$\begin{cases} \left(\left(p_1+1\right)\left(p_1\left(p_1+1\right)\pm 1\right)\right) \bmod p_1 = \mp 1; \\ \left(p_1\left(p_1\left(p_1+1\right)\pm 1\right)\right) \bmod \left(p_1+1\right) = \mp 1; \\ \left(p_1\left(p_1+1\right)\right) \bmod \left(p_1\left(p_1+1\right)\pm 1\right) = \pm 1. \end{cases} \quad (3.4)$$

Отже, якщо коефіцієнт $d=1$, то $m_1=m_2=-1$, $m_3=1$. При $d=-1$ маємо: $m_1=m_2=-1$, $m_3=1$. Знову наголосимо, що оскільки $-1 \bmod 2 = 1$, то отримаємо єдино можливий набір з трьох модулів для ДФ СЗК: $p_1=2$, $p_2=3$, $p_3=5$.

Розглянемо такі приклади:

1) нехай модулі $p_1=10$, $p_2=11$, $A=83<P=110$. Запишемо 83 у СЗК: $(83)_{10}=(3,\ 6)_{10,\ 11}$. Згідно з виразом (3.2), зворотне перетворення із СЗК у десяткову систему числення матиме такий вигляд: $(-6\cdot 10 + 11) \bmod 110 = -27 \bmod 110 = 83$;

2) нехай p_1=5, p_2=6, p_3=31, A = 802 < P = 930. Запишемо 802 у СЗК: $(802)_{10}=(2, 4, 27)_{5, 6, 31}$. Для зворотного перетворення класичним методом потрібно виконати дії у такій послідовності: $m_1=(6\cdot31)^{-1}\text{mod } 5=1$; $m_2=(5\cdot31)^{-1}\text{mod } 6=5$; $m_3=(5\cdot6)^{-1}\text{mod } 31=30$; $A=(1\cdot6\cdot31\cdot2+5\cdot5\cdot31\cdot4+30\cdot5\cdot6\cdot27) \text{ mod } 930 = 27772 \text{ mod } 930 = 802$. При використанні МДФ СЗК вказані обчислення значно спрощуються: $m_1=(6\cdot31)^{-1}\text{mod } 5=1$; $m_2=(5\cdot31)^{-1}\text{mod } 6=-1 \text{ mod } 6$; $m_3=(5\cdot6)^{-1}\text{mod } 31=-1 \text{ mod } 31$; $A=(1\cdot6\cdot31\cdot2-1\cdot5\cdot31\cdot4-1\cdot5\cdot6\cdot27) \text{ mod } 930 = -1058 \text{ mod } 930 = 02$;

3) нехай p_1=5, p_2=6, p_3=29, A = 802 < P = 870. Запишемо 802 у СЗК: $(802)_{10}=(2, 4, 19)_{5, 6, 29}$. Для зворотного перетворення класичним методом потрібно виконати дії у такій послідовності: $m_1=(6\cdot29)^{-1}\text{mod } 5=4$; $m_2=(5\cdot29)^{-1}\text{mod } 6=1$; $m_3=(5\cdot6)^{-1}\text{mod } 29=1$; $A=(4\cdot6\cdot29\cdot2+1\cdot5\cdot29\cdot4+1\cdot5\cdot6\cdot19) \text{ mod } 870 = 2542 \text{ mod } 870 = 802$. При використанні МДФ СЗК обчислення знову спрощуються: $m_1=(6\cdot29)^{-1}\text{mod } 5=-1 \text{ mod } 5$; $m_2=(5\cdot29)^{-1}\text{mod } 6=1$; $m_3=(5\cdot6)^{-1}\text{mod } 29=1$;

$A=(-1\cdot6\cdot29\cdot2+1\cdot5\cdot29\cdot4+1\cdot5\cdot6\cdot19) \text{ mod } 930 = 802$.

При використанні МДФ СЗК усувається необхідність пошуку оберненого елемента та множення на m_i при відновленні десяткового числа A.

Система (3.4) дозволяє записати загальну формулу для визначення різноманітних наборів будь-якої кількості модулів, в яких коефіцієнти $m_i=\pm1$. Вважаючи p_1 найменшим у наборі модулів, можна отримати:

$$\begin{cases} p_2 = p_1 + 1; \\ p_i = p_1 p_2 \ldots p_{i-1} \pm 1, \end{cases} \tag{3.5}$$

де i = 3, ... , k.

Зауважимо, що умові m_i=1 відповідають додатні значення модулів p_i, а умові m_i=–1 – від'ємні.

3.2. Побудова трьохмодульної модифікованої досконалої форми системи залишкових класів на основі факторизації

Для демонстрації запропонованого методу обмежимо розрахунки трьома модулями і запишемо вираз (1.26) у вигляді системи:

$$\begin{cases} p_2 p_3 \bmod p_1 = \pm 1; \\ p_1 p_3 \bmod p_2 = \pm 1; \\ p_1 p_2 \bmod p_3 = \pm 1. \end{cases} \tag{3.6}$$

Здійснюючи обчислення, аналогічні до розрахунків у п. 2.2, отримаємо таке рівняння:

$$\sum_{i=1}^{3} \frac{1}{p_i} = \gamma \pm \frac{1}{\prod_{i=1}^{3} p_i}, \tag{3.7}$$

де γ=±1, ±2, ±3, ...

На відміну від ДФ СЗК, коефіцієнт γ можна вибрати таким, що дорівнює 0, що при заданій кількості модулів відповідає найбільшому значенню P. Тоді останню рівність можна переписати у такому вигляді:

$$\frac{1}{p_1} + \frac{1}{p_2} + \frac{1}{p_3} = \pm \frac{1}{p_1 p_2 p_3}. \tag{3.8}$$

У ДФ СЗК найменші модулі набувають строго визначених значень (p_1=2, p_2=3), а у МДФ СЗК найменші модулі можуть бути будь-якими. Домноживши вираз (3.8) на P, отримаємо:

$$p_1 p_2 + p_2 p_3 + p_1 p_3 = \pm 1. \qquad (3.9)$$

Представивши вираз (3.9) у такому вигляді:

$$p_2 p_3 + p_1 (p_2 + p_3) = \pm 1, \qquad (3.10)$$

введемо позначення:

$$p_{2,3} = a, b - p_1. \qquad (3.11)$$

Після підстановки виразу (3.11) в (3.10) та відповідних математичних перетворень матимемо умову, яка має виконуватися для визначення набору з трьох модулів для МДФ СЗК:

$$\pm 1 + p_1^2 = ab. \qquad (3.12)$$

Це означає, що ліва частина виразу (3.12) має бути факторизована, на основі чого визначаються шукані параметри a та b. Отже, вираз (3.12) визначає умову для побудови МДФ СЗК, яка складається з трьох модулів.

Нехай p_1=7. Тоді з формул (3.11) та (3.12) отримаємо: $p_{2,3} = a, b - 7$ і $ab = \pm 1 + 49 = \begin{cases} 50 = 2 \cdot 5 \cdot 5 \\ 48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 \end{cases}$. Усі можливі варіанти з трьох модулів для МДФ СЗК при p_1=7 в табл. 3.1.

Для відображення графіка залежності модулів їх потрібно перенумерувати в порядку зростання абсолютної величини p_3 (табл. 3.2).

На рис. 3.1 зображено зміни значень модулів p_3 та p_4 залежно від номера модуля згідно з даними табл. 3.2.

Як можна простежити, модуль p_2 відносно повільно зростає. Водночас крива графіка для значення модуля p_3 зростає інтенсивніше, доходить до максимуму приблизно посередині номерного діапазону модулів, а потім спадає майже до значення модуля p_2.

Слід зазначити, що найбільший діапазон обчислень буде в тому разі, коли кожен наступний модуль є на одиницю більшим від добутку абсолютних величин попередніх модулів.

Крім того, згідно з даними табл. 3.1, при застосуванні цих модулів МДФ СЗК розрядність чисел, над якими виконуються арифметичні операції, зменшується в 2–3 рази.

Таблиця 3.1

Можливі варіанти систем з трьох модулів для МДФ СЗК при p_1=7 (в дужках – розрядність в двійковій системі числення).

№	p_1	ab	a	b	p_2	p_3	P
1	7 (3)	48	1	48	-6 (3)	41 (6)	1722 (11)
2			-1	-48	-8 (4)	-55 (6)	3080 (12)
3			2	24	-5 (3)	17 (5)	595 (10)
4			-2	-24	-9 (4)	-31 (5)	1953 (11)
5			3	16	-4 (3)	9 (4)	252 (9)
6			-3	-16	-10 (4)	-23 (5)	1610 (11)
7			4	12	-3 (2)	5 (3)	105 (7)
8			-4	-12	-11 (4)	-19 (5)	1463 (11)
9			6	8	-1 (1)	1 (1)	7 (3)
10			-6	-8	-13 (4)	-15 (4)	1365 (11)
11		50	1	50	-6 (3)	43 (6)	1806 (11)
12			-1	-50	-8 (4)	-57 (6)	3192 (12)
13			2	25	-5 (3)	18 (5)	630 (10)
14			-2	-25	-9 (4)	-32 (6)	2016 (11)
15			5	10	-2 (2)	3 (2)	42 (6)
16			-5	-10	-12 (4)	-17 (5)	1428 (11)

Таблиця 3.2

Впорядкування модулів

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p_2	1	2	3	4	5	5	6	6	8	8	9	9	10	11	12	13
p_3	1	3	5	9	17	18	41	43	55	57	31	32	23	19	17	15

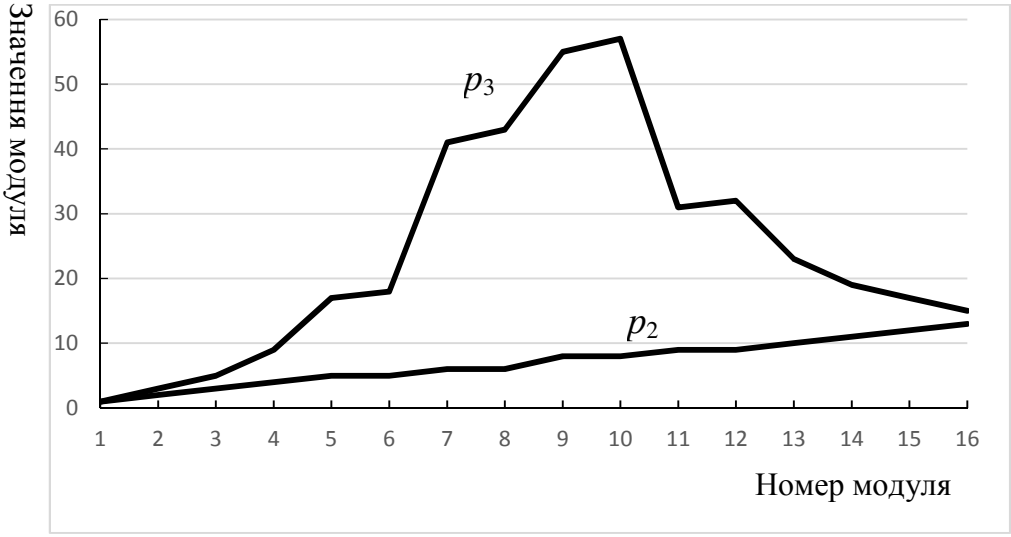


Рис. 3.1. Зміни значень модулів p_2 та p_3 залежно від номера модуля (згідно з даними табл. 4.2)

3.3. Побудова трьохмодульної модифікованої досконалої форми системи залишкових класів за допомогою теореми Вієта

У рівнянні (3.10) є добуток і сума невідомих модулів p_2 та p_3. Для їхнього пошуку введемо позначення $p_2 p_3 = \chi p_1 \pm 1$. Тоді відповідно $p_2 + p_3 = -\chi$. За допомогою теореми Вієта можна побудувати квадратне рівняння, цілочисельними коренями якого будуть значення шуканих модулів:

$$x^2 + \chi x + \chi p_1 \pm 1 = 0. \qquad (3.13)$$

Розв'язавши рівняння (3.13), невідомі модулі можна записати таким чином:

$$p_{2,3} = \frac{1}{2}\left(-\chi \pm \sqrt{\chi^2 - 4(\chi p_1 \pm 1)}\right). \qquad (3.14)$$

З формули (4.14) можна визначити, що розв'язки рівняння (3.13) будуть цілочисельні, коли його дискримінант дорівнюватиме повному квадратові деякого числа, яке зручно представити в такому вигляді:

$$\chi^2 - 4(\chi p_1 \pm 1) = (\chi - 2(p_1 + \rho))^2. \qquad (3.15)$$

Парність другого доданка в дужках у правій частині виразу випливає з дискримінанта у лівій частині формулі (3.15). Після відповідних перетворень рівності (3.15) отримуємо:

$$\chi = 2p_1 + \rho + \frac{p_1^2 \pm 1}{\rho}. \qquad (3.16)$$

Отже, МДФ СЗК з трьох модулів існує, коли виконується умова $\left(p_1^2 \pm 1\right) \bmod \rho = 0$. Це означає, що параметр ρ обмежується інтервалом $\left[-p_1^2 - 1; p_1^2 + 1\right]$, ρ≠0. Розв'язавши рівняння (3.16), можна простежити, що екстремуми χ визначаються з умови $\rho = \pm\sqrt{p_1^2 \pm 1}$. На рис. 3.2 зображено графік залежності величини χ від ρ, що змінюється від –26 до 26, при p_1=5.

Графік та побудоване на основі формули (3.15) відносно ρ квадратне рівняння $\rho_2^2 + \rho_2(2p_1 - \chi) + p_1^2 \pm 1 = 0$ підтверджують, що фіксованій величині χ відповідають два значення параметра ρ, причому, згідно з теоремою Вієта, якщо одне з них є цілочисельним, то й інше також має бути цілим числом. Це дає

змогу зменшити діапазон дослідження ρ до таких меж: $[-p_1+1; p_1-1]$, ρ≠0. На рис. 3.3 для прикладу зображено поверхню, яка, згідно з виразом $\chi = 2p_1 + \rho + \frac{p_1^2 - 1}{\rho}$, характеризує залежність параметра χ від значень p_1=2…10 та ρ=1… p_1–1.

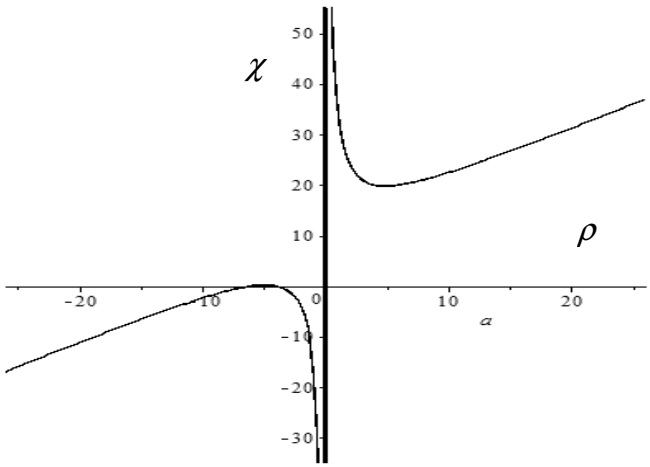


Рис. 3.2. Графік залежності величини χ від параметра ρ

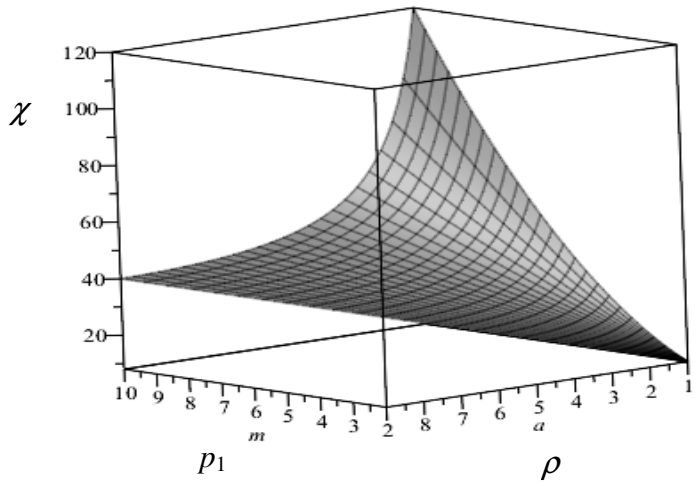


Рис. 3.3. Графік залежності параметра s_1 від значень p_1=2…10 та ρ=1… p_1–1

Як видно з графіка, величина χ досягає максимуму при найбільшому значенні модуля p_1 та найменшому значенні параметра ρ. Мінімум χ отримуємо при мінімальних значеннях p_1 і ρ.

На рис. 3.4, 3.5 зображено відповідно графіки залежності модулів $p_2 = \frac{1}{2}\left(-\rho + \sqrt{\rho^2 - 4(\rho p_1 - 1)}\right)$ та $p_3 = \frac{1}{2}\left(-\rho - \sqrt{\rho^2 - 4(\rho p_1 - 1)}\right)$ від значень модуля p_1=2…10 і параметра ρ=1…p_1-1.

З рис. 3.4 і 3.5 простежуємо, що графіком залежності модуля p_2 від значень p_1 і ρ є площина, а модуля p_3 – гіперболоїд.

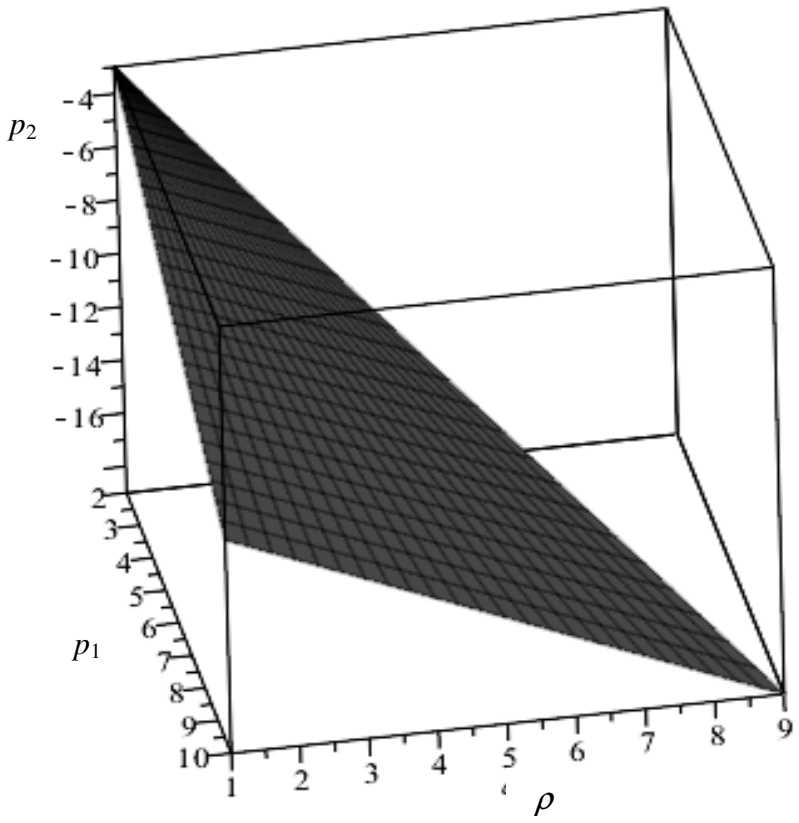


Рис. 3.4. Графік залежності модуля p_2. від значень модуля p_1 і параметра ρ

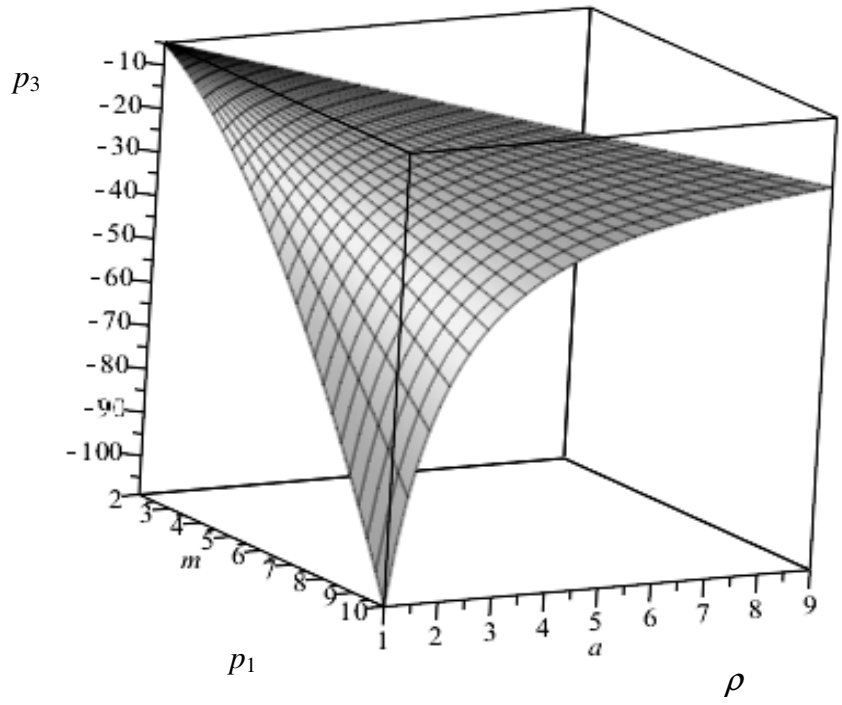


Рис. 3.5. Графік залежності шуканого модуля p_3 від значень модуля p_1 та параметра ρ

Мінімального абсолютного значення модулі p_2 та p_3 набувають при найменших величинах p_1 і ρ. Максимальне абсолютне значення p_2 буде у разі, коли $p_1 \to \max$, $\rho \to \max$, відповідно $|p_3| \to \max$, якщо $p_1 \to \max$, $\rho \to \min$.

У табл. 3.3 подано можливі значення p_2, p_3, відповідних їм параметрів ρ, χ для p_1=7.

Згідно з даними табл. 3.3, модуль p_3 набуває тільки від'ємних значень. При умові ρ<0 модуль p_2 додатний і, навпаки, якщо ρ>0, то $p_2 < 0$.

Таблиця 3.3

Можливі значення p_2, p_3, відповідних їм параметрів χ, ρ для p_1=7

№	p_1	ρ	χ	p_2	p_3
1	7	-6	0	1	-1
2		-5	-1	3	-2
3		-4	-2	5	-3
4		-3	-5	9	-4
5		-2	-12	18	-5
6		-2	-13	17	-5
7		-1	-35	41	-6
8		-1	-37	43	-6
9		1	65	-8	-57
10		1	63	-8	-55
11		2	41	-9	-32
12		2	40	-9	-31
13		3	33	-10	-23
14		4	30	-11	-19
15		5	29	-12	-17
16		6	28	-13	-15

3.4 Побудова трьохмодульної модифікованої досконалої форми системи залишкових класів на основі розв'язку систем конгруенцій

Нехай $p_1=a$, $p_2=a+b$ (причому a і b взаємно прості). Згідно з умовами МДФ СЗК (1.26), має виконуватися така система:

$$\begin{cases} ap_3 \bmod (a+b) = \pm 1; \\ (a+b)p_3 \bmod a = \pm 1; \\ a(a+b) \bmod p_3 = \pm 1. \end{cases} \qquad (3.17)$$

Розглянемо спочатку перші два рівняння системи (3.17), які з урахуванням перетворень $ap_3 \bmod(a+b) = -bp_3 \bmod(a+b)$ та $(a+b)p_3 \bmod a = bp_3 \bmod a$ подамо таким чином:

$$\begin{cases} y \bmod(a+b) = \mp 1, \\ y \bmod a = \pm 1, \end{cases} \tag{3.18}$$

де $y = bp_3$.

Далі потрібно використати розширений алгоритм Евкліда та КТЗ для чисел *a* та (*a*+*b*). Обернені елементи для аналітичних розрахунків зручно шукати методом додавання модуля, запропонованим у працях науковці [218–220], коли до 1 додається модуль і перевіряється, чи ділиться націло отриманий результат на відповідне число. Якщо ні, то знову додається модуль і виконується вказана перевірка. Отже:

$$a^{-1} \bmod(a+b) = -b^{-1} \bmod(a+b) = -\frac{k_1(a+b)+1}{b}; \tag{3.19}$$

$$(a+b)^{-1} \bmod a = b^{-1} \bmod a = \frac{k_2 a + 1}{b}, \tag{3.20}$$

де k_1, k_2 дорівнюють кількості доданих модулів (*a*+*b*) та *a* відповідно.

Рівність коефіцієнтів $k_1 = k_2 = k$ в обох виразах підтверджується записом результату для розширеного алгоритма Евкліда:

$$\frac{(a+b)(ka+1)}{b} - \frac{a(k(a+b)+1)}{b} = 1. \tag{3.21}$$

Далі потрібно використати КТЗ для чотирьох випадків, записаних у системі (3.18): (1; 1), (1; –1), (–1; 1), (–1; –1). Отримаємо:

1) $y \bmod a(a+b) = 1;$

2) $y \bmod a(a+b) = \dfrac{2ka^2 + 2kab + 2a + b}{b} \bmod a(a+b) =$

$$= \frac{2ka(a+b) + 2a + b}{b} \bmod a(a+b) = \left(2ka + 1 + \frac{2a(ka+1)}{b}\right) \bmod a(a+b);$$

3) $y \bmod a(a+b) = -\dfrac{2ka^2 + 2kab + 2a + b}{b} \bmod a(a+b) =$

$$= -\frac{2ka(a+b) + 2a + b}{b} \bmod a(a+b) = -\left(2ka + 1 + \frac{2a(ka+1)}{b}\right) \bmod a(a+b);$$

4) $y \bmod a(a+b) = -1.$

Другий і третій випадки складні для аналізу та, як засвідчує практика, для них отримується або $p_3 < p_2$, або взагалі третє рівняння системи (3.17) не виконується. За аналогією до рівностей (3.19) і (3.20), перший та четвертий випадки дають змогу отримати такі результати:

$$p_3 = b^{-1} \bmod a(a+b) = \frac{k_3 a(a+b) + 1}{b}; \qquad (3.22)$$

$$p_3 = -b^{-1} \bmod a(a+b) = -\frac{k_3 a(a+b) + 1}{b} = \frac{(b - k_3) a(a+b) - 1}{b}, \qquad (3.23)$$

де k_3 дорівнює кількості доданих модулів $a(a+b)$.

З третього рівняння системи (3.17), враховуючи рівності (3.22) і (3.23), отримуємо:

$$a(a+b)\bmod\frac{k_3a(a+b)+1}{b}=\pm1; \tag{3.24}$$

$$a(a+b)\bmod\frac{(b-k_3)a(a+b)-1}{b}=\pm1. \tag{3.25}$$

Вираз під функцією mod у формулах (3.24), (3.25) потрібно помножити на *b* і цей добуток має бути на 1 більший або менший від *a*(*a*+*b*). Це можливо при $k_3=1$ або $b-k_3=1$. Отже, отримуємо остаточний вираз для знаходження третього модуля:

$$p_3=a+\frac{a^2\pm1}{b}, \tag{3.26}$$

який за абсолютною величиною на $\frac{a^2\pm1}{b}$ більший від першого.

Звідси випливає умова, при виконанні якої існує третій модуль для МДФ СЗК:

$$a^2\bmod b=\pm1. \tag{3.27}$$

Рівняння (3.26) і (3.27) пояснюють, чому при заданому *a* не для всіх *b* можна підібрати третій модуль для МДФ СЗК. Крім того, з формули (3.26) можна визначити, що *b*<*a*, інакше $p_3<p_2$.

На рис. 3.6 зображено графік залежності p_3 від значень модулів p_1 та p_2 згідно з формулою (3.26) при умові $p_2>p_1$.

Як видно з графіка, p_3 досягає максимуму, коли p_1 та p_2 приблизно однакові (p_1 при цьому є максимальним). Із збільшенням p_2 модуль p_3 зменшується і при максимальному значенні другого модуля p_2 та p_3 набувають приблизно однакових значень.

У табл. 3.4 подано можливі набори для трьох модулів МДФ СЗК при p_1=11 та їхній добуток (у дужках – розрядність цих чисел), який вказує на умову переповнення розрядної сітки процесора.

Згідно з даними табл. 3.4 можна зробити висновок, що при використанні трьох модулів МДФ СЗК розрядність чисел, над якими виконуються арифметичні операції, зменшується в 2–2,5 разу.

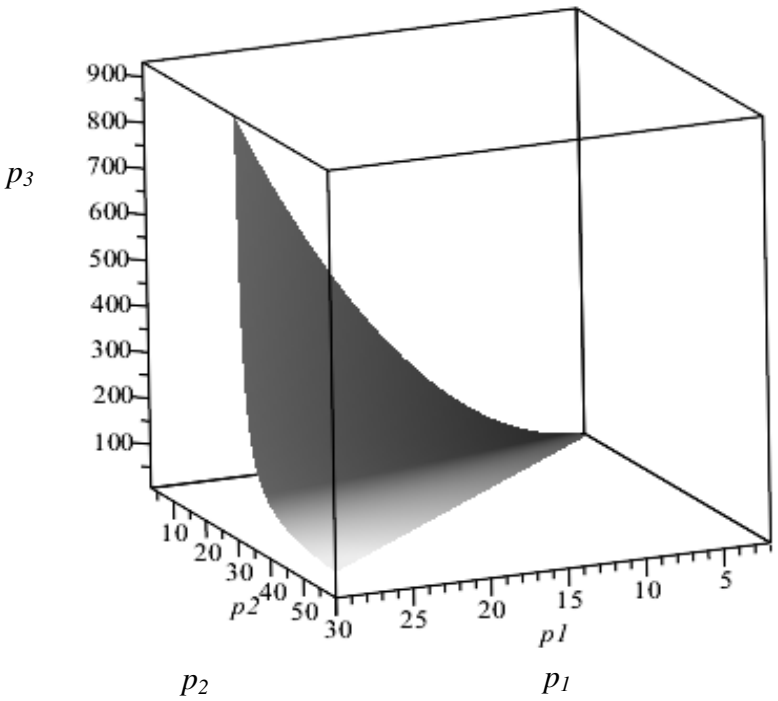


Рис. 3.6. Поверхня, яка характеризує залежність p_3 від значень модулів p_1 та p_2

Крім того, значеню b=1 відповідає два значення p_3, які на 1 відрізняються від добутку двох попередніх модулів. Це

випливає з виразу (3.26), оскільки в цьому разі $p_3 = a(a+1)\pm 1$. Два значення p_3 також отримується при b=2, оскільки p_1 непарне, тому обидва числа $p_1^2 \pm 1$ діляться націло на 2. При b=7 та 9 цілого значення p_3 не існує, оскільки $11^2 \bmod 7 \neq \pm 1$ і $11^2 \bmod 9 \neq \pm 1$.

Таблиця 3.4

Набори для трьох модулів МДФ СЗК при p_1=11 та їхній добуток (у дужках вказана розрядність чисел)

№	$p_1=a$	b	p_2	p_3	P
1	11 (4)	1	12 (4)	133 (8)	17556 (15)
2		1	12 (4)	131 (8)	17292 (15)
3		2	13 (4)	72 (7)	10296 (14)
4		2	13 (4)	71 (7)	10153 (14)
5		3	14 (4)	51 (6)	7854 (13)
6		4	15 (4)	41 (6)	6765 (13)
7		5	16 (5)	35 (6)	6160 (13)
8		6	17 (5)	31 (5)	5797 (13)
9		7	18 (5)	відсутній	-
10		8	19 (5)	26 (5)	5434 (13)
11		9	20 (5)	відсутній	-
12		10	21 (5)	23 (5)	5313 (13)

Для подальшого дослідження модулів p_2 та p_3 дані табл. 3.4 необхідно трансформувати (табл. 3.5).

Таблиця 3.5

Впорядкована система модулів

№	1	2	3	4	5	6	7	8	9	10
p_2	12	12	13	13	14	15	16	17	19	21
p_3	133	131	72	71	51	41	35	31	26	23

На рис. 3.7 зображено графік значень модулів p_2 та p_3 залежно від їхнього номера згідно з даними табл. 3.5.

Як видно з графіка, модуль p_2 повільно збільшується із зростанням його номера. Водночас значення p_3 зменшується набагато інтенсивніше і в кінці розглянутого діапазону обидва модулі приблизно однакові.

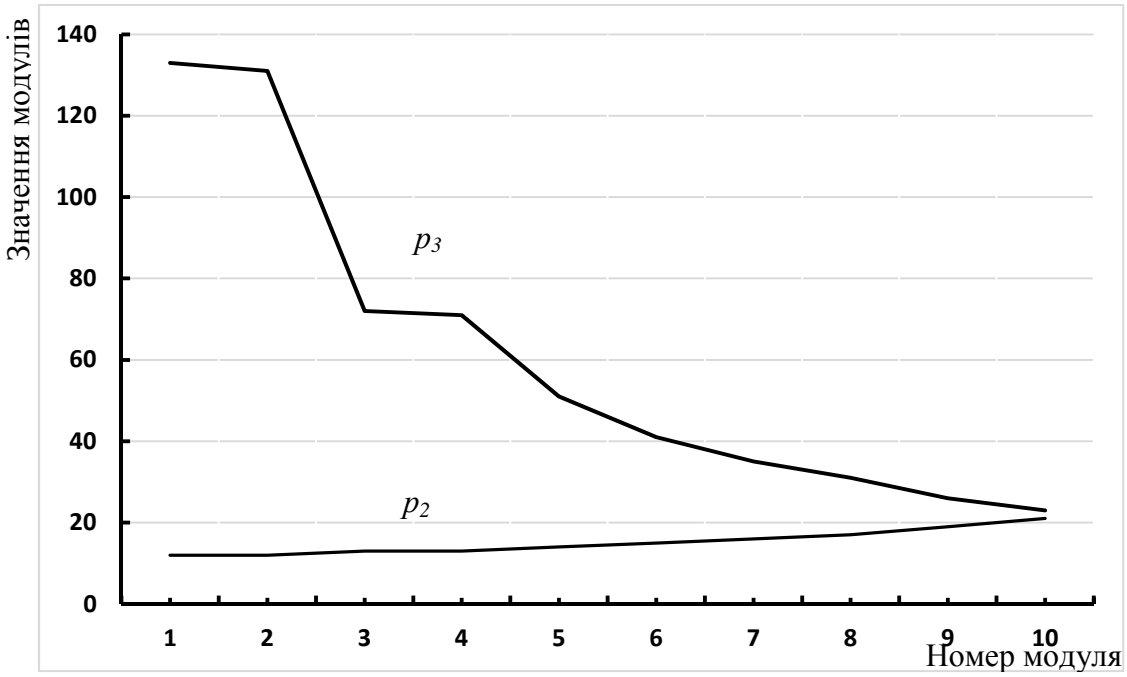


Рис. 3.7. Графік значень модулів p_2 та p_3 залежно від їхнього номера (згідно з даними табл. 4.5)

3.5. Метод побудови системи модулів модифікованої досконалої форми системи залишкових класів із використанням послідовності Фібоначчі

Цікавим є випадок, коли третій модуль дорівнює сумі абсолютних величин двох попередніх. Тоді з виразу (3.26) можна отримати $a + \frac{a^2 \pm 1}{b} = 2a + b$. Переходячи до квадратного

рівняння $b^2+ab-(a^2\pm1)=0$ відносно b і упускаючи його від'ємні корені, знаходимо:

$$b=\frac{-a+\sqrt{5a^2\pm4}}{2}. \qquad (3.28)$$

Це означає, що вираз $(5a^2\pm4)$ має бути повним квадратом деякого числа. Чисельні дослідження підтверджують, що таку умову задовольняє послідовність Фібоначчі: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, … .

Цей ряд виникає не тільки в різних математичних ситуаціях – комбінаторних, чисельних, геометричних, а й у біологічних системах. Використання такої послідовності для обчислювальних систем описано, наприклад, у науковій праці [221].

Аналітично n-й елемент ряду подається за допомогою формули Біне: $F_n=\frac{\left(1+\sqrt{5}\right)^n-\left(1-\sqrt{5}\right)^n}{2^n\cdot\sqrt{5}}$.

Доцільно в загальному випадку розглянути три послідовні елементи ряду Фібоначчі. Враховуючи, що кожен наступний елемент дорівнює сумі двох попередніх і квадрат кожного елемента на одиницю відрізняється від добутку попереднього і наступного членів, отримаємо таку систему:

$$\begin{cases} F_{n+1}F_{n+2}\bmod F_n=F_{n-1}^2\bmod F_n=\pm1; \\ F_nF_{n+2}\bmod F_{n+1}=F_n^2\bmod F_{n+1}=\pm1; \\ F_nF_{n+1}\bmod F_{n+2}=F_{n+1}^2\bmod F_{n+2}=\pm1. \end{cases} \qquad (3.29)$$

Отже, щоб отримати набір з трьох модулів для МДФ СЗК, в якому третій модуль дорівнює сумі абсолютних величин двох попередніх, необхідно вибрати будь-які три послідовні елемента з ряду Фібоначчі, починаючи з третього. На рис. 3.8

зображені графічні залежності F_n, F_{n+1} та F_{n+2} для перших 12 елементів послідовності.

З рис. 3.8 можна визначити, що графіки функцій швидко зростають зі збільшенням числа n. Слід зауважити, що модулі МДФ СЗК знаходяться на перетині вертикальних ліній сітки з побудованими графіками, де значення функції набувають цілочисельних значень і задовольняють умови системи (3.29).

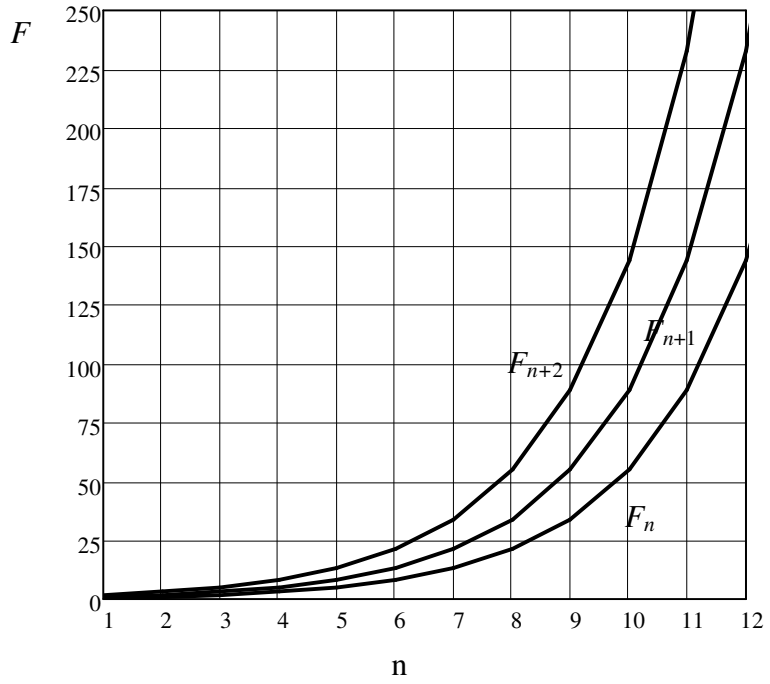


Рис. 3.8. Графічні залежності F_n, F_{n+1} та F_{n+2} для перших 12 елементів послідовності Фібоначчі

3.6. Побудова трьохмодульної модифікованої досконалої форми системи залишкових класів для багаторозрядних чисел

У разі багаторозрядних чисел, коли задана різниця між другим і першим модулями ($p_2-p_1=q<p_1$, причому p_1 та q мають бути взаємно простими), подамо їх у вигляді $p_1=qn-r$, $p_2=qn-r+q$

та побудуємо систему рівнянь для МДФ СЗК, відповідно до системи (3.17):

$$\begin{cases}(qn-r)p_3 \bmod(qn-r+q)=qp_3 \bmod(qn-r+q)=\mp 1;\\ (qn-r+q)p_3 \bmod(qn-r)=qp_3 \bmod(qn-r)=\pm 1;\\ (qn-r)(qn-r+q)\bmod p_3=\pm 1.\end{cases} \quad (3.30)$$

Аналогічно до попереднього, розглянемо спочатку перші два рівняння системи (3.30), які з урахуванням відповідних математичних перетворень набудуть такого вигляду:

$$\begin{cases}z \bmod(qn-r+q)=\mp 1;\\ z \bmod(qn-r)=\pm 1.\end{cases} \quad (3.31)$$

де $z=qp_3$.

Ця система розв'язується на основі КТЗ, яка передбачає пошук оберненого елемента за модулем. Оскільки вираз (3.31) розглядається у загальному вигляді без числових значень, то у цьому разі застосувати розширений алгоритм Евкліда неможливо і обернений елемент потрібно шукати за допомогою додавання модуля. Отже:

$$(qn-r)^{-1} \bmod(qn-r+q)=-q^{-1} \bmod(qn-r+q)=-\frac{k_1(qn-r+q)+1}{q}; \quad (3.32)$$

$$(qn-r+q)^{-1} \bmod(qn-r)=q^{-1} \bmod(qn-r)=\frac{k_2(qn-r)+1}{q}, \quad (3.33)$$

де k_1, k_2 дорівнюють кількості доданих модулів $(qn-r+q)$ та $(qn-r)$ відповідно.

Тоді запис виразу розширеного алгоритму Евкліда

$$\frac{k(qn-r)+1}{q}(qn-r+q)-\frac{k(qn-r+q)+1}{q}(qn-r)=1 \qquad (3.34)$$

вказує на рівність коефіцієнтів $k_1 = k_2 = k$ в обох рівняннях (3.32) і (3.33).

Далі розглядається КТЗ для чотирьох випадків, записаних у системі (3.31): (1; 1), (1; –1), (–1; 1), (–1; –1). Друга і третя пари залишків дають змогу отримати складні для аналізу вирази:

$$z \bmod (qn-r)(qn-r+q) = \\ = \pm\frac{2k(qn-r)(qn-r+q)+2(qn-r)+q}{q} \bmod (qn-r)(qn-r+q), \quad (3.35)$$

з яких, як підтверджують чисельні розрахунки, отримується або $p_3 < p_2$, або не виконується взагалі третє рівняння системи (3.30).

З першої та четвертої пар залишків можна записати:

$$z \bmod (qn-r+q)(qn-r) = \pm 1. \qquad (3.36)$$

Тоді з рівності (3.36), аналогічно до виразів (3.32) і (3.33), шукають відповідні обернені елементи:

$$p_3 = q^{-1} \bmod (qn-r)(qn-r+q) = \frac{k_3(qn-r)(qn-r+q)+1}{q}; \quad (3.37)$$

$$p_3 = -q^{-1} \bmod (qn-r)(qn-r+q) = -\frac{k_3(qn-r)(qn-r+q)+1}{q} = \\ = \frac{(b-k_3)(qn-r)(qn-r+q)-1}{q}, \quad (3.38)$$

де k_3 дорівнює кількості доданих модулів $(qn-r+q)(qn-r)$.

Далі, враховуючи вирази (3.37) і (3.38), з третього рівняння системи (3.30) отримаємо:

$$p_3 = qn(n+1) - r(2n+1) + \frac{r^2 \pm 1}{q}. \qquad (3.39)$$

На рис. 3.9 зображено залежність модуля p_3 від параметрів q і r при n=3, а на рис. 3.10 – залежність p_3 від n і r при сталій різниці $q=p_2-p_1=5$, згідно з формулою (3.39), де у чисельнику останнього доданка взято (r^2–1).

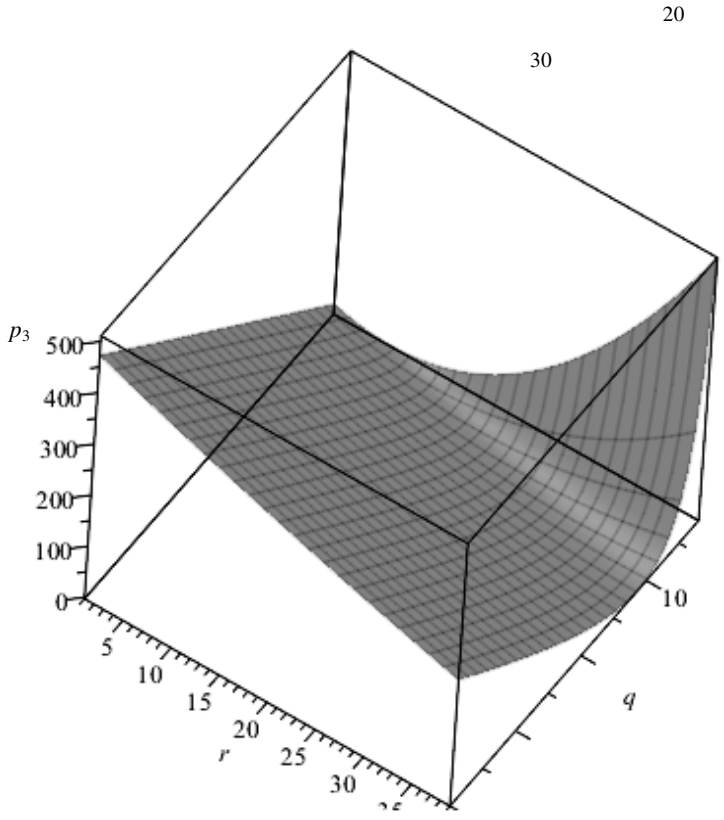


Рис. 3.9. Залежність модуля p_3 від параметрів q і r при n=3

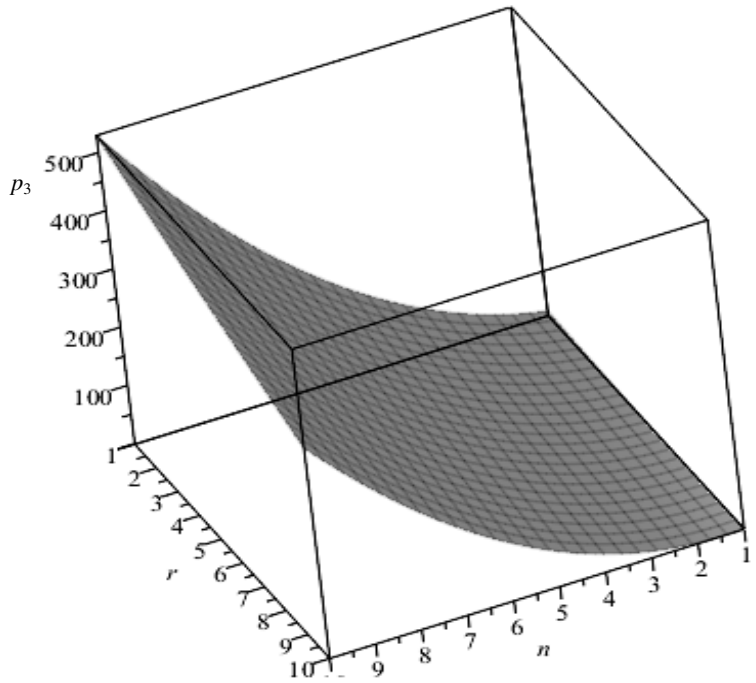


Рис. 3.10. Залежність модуля p_3 від параметрів n і r при сталій різниці модулів $q=p_2-p_1=5$

Як видно із рис. 3.9, при малих q та великих r модуль p_3 змінюється гіперболічно, в інших випадках – лінійно. На рис. 3.10 графіком є параболоїд, модуль p_3 зростає інтенсивніше при малих r та великих n.

З виразу (3.39) випливає умова існування третього модуля, яка має вигляд:

$$r^2 \bmod q = \pm 1. \qquad (3.40)$$

Це, наприклад, пояснює, що для q=7 третій модуль можна знайти тільки при r=1 і r=6. Крім того, коли p_2-p_1=1, тобто q=1, r=0, або p_2-p_1=2 (q=2, r=1), то тоді треті модулі p_3 можуть набувати відповідно два значення: $p_3=n^2+n\pm 1$ та $p_3= 2n^2-(1\pm 1)/2$. У табл. 3.6 наведено аналітичні вирази для третього модуля p_3 і

можливого діапазону обчислень *P* залежно від числа *n* при зміні значення *q* від 1 до 7.

Таблиця 3.6

Аналітичні вирази для модуля p_3 та можливого діапазону обчислень *P* в залежності від числа *n* при зміні параметрів *q* та *r*

№	*Q*	*r*	p_1	p_2	p_3	*P*
1	1	0	n	$n+1$	n^2+n+1, n^2+n-1	$n^4+2n^3+2n^2+n$, n^4+2n^3-n
2	2	1	$2n-1$	$2n+1$	$2n^2$, $2n^2-1$	$8n^4-2n^2$, $8n^4-6n^2+1$
3	3	2	$3n-2$	$3n+1$	$3n^2-n-1$	$27n^4-18n^3-12n^2+5n+2$
4	3	1	$3n-1$	$3n+2$	$3n^2+n-1$	$27n^4+18n^3-12n^2-5n+2$
5	4	3	$4n-3$	$4n+1$	$4n^2-2n-1$	$64n^4-64n^3-12n^2+14n+3$
6	4	1	$4n-1$	$4n+3$	$4n^2+2n-1$	$64n^4+64n^3-12n^2-14n+3$
7	5	4	$5n-4$	$5n+1$	$5n^2-3n-1$	$125n^4-150n^3+27n+4$
8	5	3	$5n-3$	$5n+2$	$5n^2-n-1$	$125n^4-50n^3-50n^2+11n+6$
9	5	2	$5n-2$	$5n+3$	$5n^2+n-1$	$125n^4+50n^3-50n^2-11n+6$
10	5	1	$5n-1$	$5n+4$	$5n^2+3n-1$	$125n^4+150n^3-27n+4$
11	6	5	$6n-5$	$6n+1$	$6n^2-4n-1$	$216n^4-288n^3+30n^2+44n+5$
12	6	1	$6n-1$	$6n+5$	$6n^2+4n-1$	$216n^4+288n^3+30n^2-44n+5$
13	7	6	$7n-6$	$7n+1$	$7n^2-5n-1$	$343n^4-490n^3+84n^2+65n+6$
14	7	1	$7n-1$	$7n+6$	$7n^2+5n-1$	$343n^4+490n^3+84n^2-65n+6$

Знову ж розглянемо випадок, коли третій модуль дорівнює сумі абсолютних величин двох попередніх. Тоді з (3.39) можна отримати $qn(n+1)-r(2n+1)+\dfrac{r^2\pm 1}{q}=2(qn-r)+q$.

Перейшовши до квадратного рівняння $r^2-qr(2n-1)+q^2\left(n^2-n-1\right)\pm 1=0$ відносно *r*, знайдемо:

$$r = \frac{q(2n-1) \pm \sqrt{5q^2 \pm 4}}{2}, \qquad (3.41)$$

тобто вираз ($5q^2 \pm 4$) має бути повним квадратом деякого числа. В табл. 3.7 подано можливі набори модулів, отримані згідно з виразом (3.41), для різних q.

Таблиця 3.7

Можливі набори модулів, отримані за виразом (3.41), для різних q.

№	q	r	$p_1=qn-r$	$p_2=qn-r+q$	$p_3=p_1+p_2$
1	1	$n-2$	2	3	5
2	2	$2n-3$	3	5	8
3	3	$3n-5$	5	8	13
4	5	$5n-8$	8	13	21
5	8	$8n-13$	13	21	34
6	13	$13n-21$	21	34	55
7	21	$21n-34$	34	55	89

Як підтверджують дані табл. 3.7, значення отриманих модулів параметра q, а також відповідних числових величин для r утворюють послідовність Фібоначчі, в якій кожен наступний елемент дорівнює сумі двох попередніх. Крім того, параметр r записується аналітично, однак модулі, які отримуються з його допомогою, набувають конкретних числових значень.

3.7. Приклад застосування розробленого методу

Нехай q=5, r=2.Тоді p_1=5n–2, p_2=5n+3. Перші два рівняння системи (3.30) утворять таку систему:

$$\begin{cases} 5p_3 \bmod(5n-2)=\pm 1; \\ 5p_3 \bmod(5n+3)=\pm 1. \end{cases} \quad (3.42)$$

Обернені елементи $5^{-1}\bmod(5n-2)$ та $5^{-1}\bmod(5n+3)$ шукаємо з алгоритму Евкліда та його наслідку. Наведемо приклад розрахунку тільки для одного з них:

$5n-2=5(n-1)++3$ $\quad$ $1=3-1\cdot 2=3-1\cdot(5-1\cdot 3)=-1\cdot 5+2\cdot 3=-1\cdot 5+$

$5=3\cdot 1+2$ $\quad$ $+2\cdot((5n-2)-5(n-1))=2\cdot(5n-2)-5\cdot(2n-1)=1$

$3=2\cdot 1+1$

Отже, $5^{-1}\bmod(5n-2)=-(2n-1)\bmod(5n-2)=3n-1$. Аналогічно можна зобразити, що $5^{-1}\bmod(5n+3)=-(2n+1)\bmod(5n+3)=3n+2$.

Слід зазначити, що у цьому разі обернений елемент зручніше шукати методом додавання модуля, додаючи до 1 модуль стільки разів, щоб результат ділився на 5. Так, до числа (5n–2)+1 ще два рази додається модуль. Поділивши (15n–6)+1 на 5, отримаємо в результаті (3n–1). Для другого модуля (5n+3)+1+2·(5n+3)= 15n+10. Обернене число буде дорівнювати (3n+2).

Відповідно до цього, система (3.42) трансформується у систему, зручну для застосування КТЗ, в якій необхідно розглянути чотири різних випадки, взявши по одному залишку з кожного рівняння:

$$\begin{cases} p_3 \bmod(5n-2) = 3n-1; & 2n-1; \\ p_3 \bmod(5n+3) = 3n+2; & 2n+1. \end{cases} \quad (3.43)$$

Відповідно до системи (3.43), треба знайти число, яке при діленні на (5*n*–2) дає остачу (3*n*–1) або (2*n*–1), а при діленні на (5*n*+3) – остачу (3*n*+2) або (2*n*+1). Для цього знову потрібно використати алгоритм Евкліда:

$5n+3=(5n-2)++5$	$1=3-1\cdot 2=3-1\cdot(5-1\cdot 3)=-1\cdot 5+2\cdot 3=-1\cdot 5+$
$5n-2=5\cdot(n-1)+3$	$+2\cdot((5n-2)-5(n-1))=2\cdot(5n-2)-5\cdot(2n-1)=$
$5=3\cdot 1+2$	$=2\cdot(5n-2)-(2n-1)\times((5n+3)-(5n-2))=$ $=2\cdot(5n-2)-$
$3=2\cdot 1+1.$	$-(2n-1)(5n+3)+(2n-1)\ (5n-2)=$
	$=-(2n-1)(5n+3)+(2n+1)\ (5n-2).$

Згідно з КТЗ для знаходження можливого значення p_3 потрібно розглянути окремо кожен з чотирьох випадків:

1) $(-(2n-1)(5n+3)(3n-1)+(2n+1)(5n-2)(3n+2))\,mod((5n+3)(5n-2))=$
$=(30n^2+6n-7) \bmod (25n^2+5n-6)= 5n^2+n-1;$

2) $(-(2n-1)^2(5n+3)+(2n+1)^2(5n-2))\,mod((5n+3)(5n-2))=$
$=(20n^2+4n-5) \bmod (25n^2+5n-6)= 20n^2+4n-5;$

3) $(-(2n-1)(5n+3)(2n-1)+(2n+1)(5n-2)(3n+2))\,mod((5n+3)(5n-2))=$
$=(10n^3+31n^2+3n-7) \bmod (25n^2+5n-6)= 10n^3+6n^2-2n-1;$

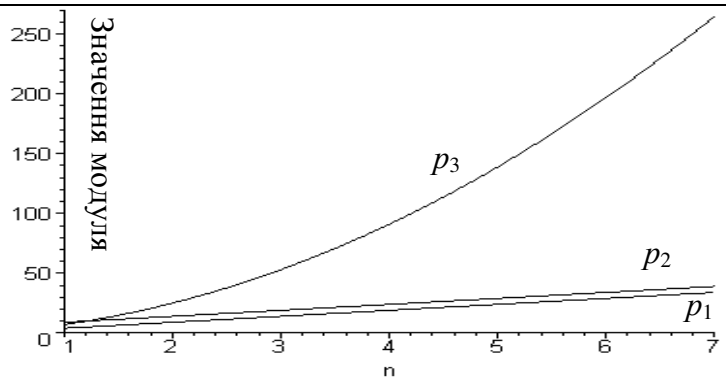
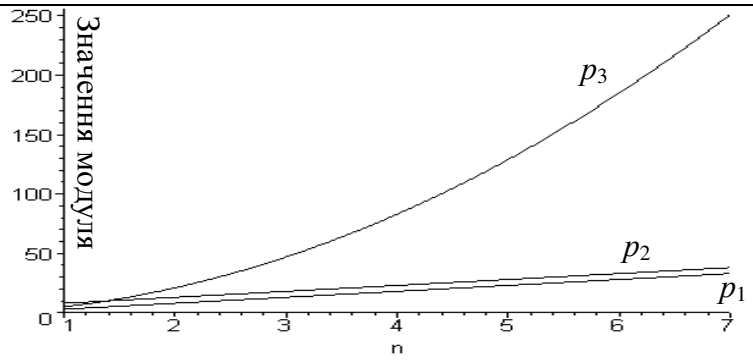
4) $(-(2n-1)(5n+3)(3n-1)+(2n+1)(5n-2)(2n+1))\,mod((5n+3)(5n-2))=$
$=(-10n^3+19n^2+7n-5) \bmod (25n^2+5n-6)= =(-10n^3+19n^2+7n-5).$

Далі необхідно перевірити виконання третього рівняння у системі (3.30). Безпосередньою підстановкою можна переконатися, що за дану умову задовольняє результат, отриманий у першому рівнянні: $(25n^2+5n-6) \bmod (5n^2+n-1)= -1.$

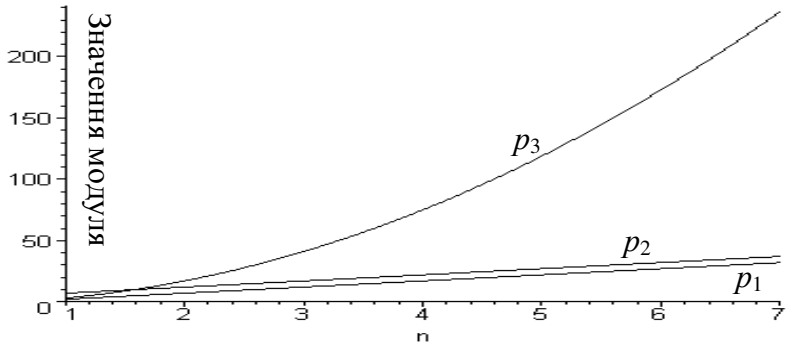
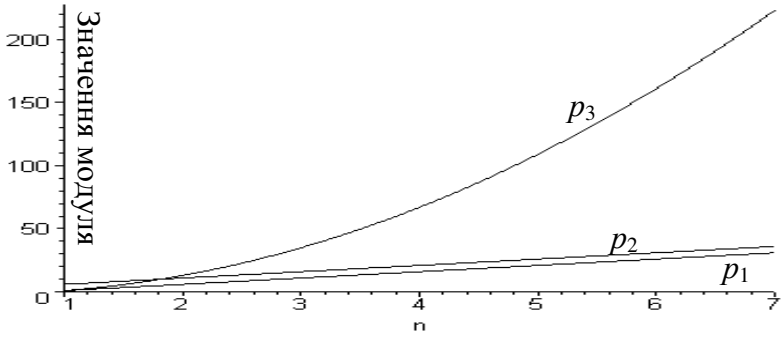
Аналогічно можна знайти модуль p_3 для будь-якого іншого значення *b*. Відповідно в табл. 3.8 подано аналітичні вирази для знаходження модулів і діапазону можливих обчислень, а також графічні залежності всіх модулів при різних *b* та *n*.

Таблиця 3.8

Аналітичні вирази для знаходження модулів і діапазону можливих обчислень, а також графічні залежності модулів при різних *b* та *n*

	p_1, p_2, p_3, P	Графічні залежності
1	2	3
1.	$p_1=5n-1$ $p_2=5n+4$ $p_3=5n^2+3n-1$ $P=125n^4+150n^3-$ $-27n+4$	
2.	$p_1=5n-2$ $p_2=5n+3$ $p_3=5n^2+n-1$ $P=125n^4+50n^3-$ $-50n^2-11n+6$	

Продовження табл. 3.8

1	2	3
3.	$p_1=5n-3$ $p_2=5n+2$ $p_3=5n^2-n-1$ $P=125n^4-50n^3-$ $-50n^2+11n+6$	
4.	$p_1=5n-4$ $p_2=5n+1$ $p_3=5n^2-3n-1$ $P=125n^4-150n^3+$ $+27n+4$	

На рис. 3.11 зображено графік залежності діапазона обчислень *P* від *n* при різних можливих значеннях параметра *b*.

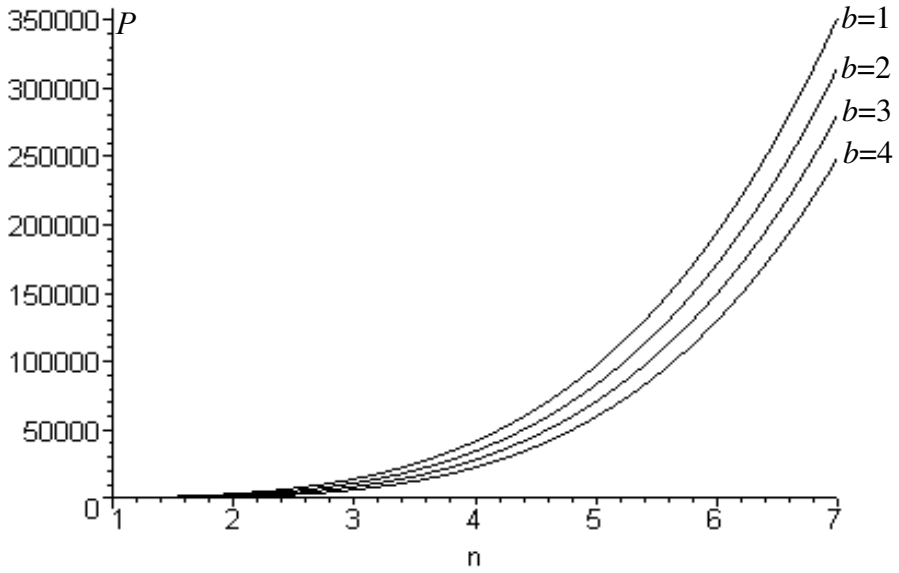


Рис. 3.11. Графік залежності діапазону обчислень *P* від *n* при різних можливих значеннях параметра *b*

Як підтверджують дані табл. 3.8 та рис. 3.11, модуль p_3 та діапазон обчислень P із збільшенням n зростають параболічно й інтенсивніше, якщо є меншим значення параметра b. У табл. 3.9 подано числові значення p_1, p_2, p_3 та P при a=5 для різних значень n та b.

Таблиця 3.9

Числові значення p_1, p_2, p_3 та P при a=5 для різних значень n і b

n	*b*	p_1	p_2	p_3	*P*
1	3	2	7	3	42
	2	3	8	5	120
	1	4	9	7	252
2	4	6	11	13	858
	3	7	12	17	1428
	2	8	13	21	2184
	1	9	14	25	3150

Продовження табл. 3.9

3	4	11	16	35	6160
	3	12	17	41	8364
	2	13	18	47	10998
	1	14	19	53	14098
4	4	16	21	67	22512
	3	17	22	75	28050
	2	18	23	83	34362
	1	19	24	91	41496

За даними табл. 3.9 можна визначити, що при n=1 між модулями виконується таке співвідношення: $p_1 < p_3 < p_2$. В інших випадках модуль p_3 є найбільшим. Крім того, при сталому n і зміні b на 1 модуль p_3 змінюється на величину $2n$, що випливає із даних табл. 3.8.

РОЗДІЛ 4
МЕТОДИ ПОБУДОВИ БАГАТОМОДУЛЬНОЇ МОДИФІКОВАНОЇ ДОСКОНАЛОЇ ФОРМИ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ

4.1 Метод побудови багатомодульної модифікованої досконалої форми системи залишкових класів на основі факторизації

Для побудови МДФ СЗК із будь-якою кількістю модулів запишемо вираз (3.1) у вигляді системи:

$$\begin{cases} P_1 \bmod p_1 = \pm 1; \\ \ldots \\ P_k \bmod p_k = \pm 1. \end{cases} \tag{4.1}$$

Розрахунки, аналогічні до проведених у п. 3.2, дають змогу отримати такий вираз:

$$\sum_{i=1}^{k} \frac{1}{p_i} = \gamma \pm \frac{1}{\prod_{i=1}^{k} p_i}, \tag{4.2}$$

де $\gamma = \pm 1, \pm 2, \pm 3, \ldots$.

На відміну від ДФ СЗК, коефіцієнт γ можна вибрати таким, що дорівнює 0 і при заданій кількості модулів відповідає

найбільшому значенню P. Тоді останню рівність можна переписати у такому вигляді:

$$\frac{1}{p_1}+\frac{1}{p_2}+\frac{1}{p_3}+...+\frac{1}{p_{k-1}}+\frac{1}{p_k}=\pm\frac{1}{p_1p_2p_3..p_{k-1}p_k}. \quad (4.3)$$

У ДФ СЗК найменші модулі набувають строго визначених значень (p_1=2, p_2=3). У МДФ СЗК найменші модулі можуть бути будь-які. Врахувавши це та відповідно трансформувавши рівність (4.3), отримаємо вираз, аналогічний до формули (4.2) при γ≠0:

$$\sum_{i=1}^{k} P_i = \pm 1. \quad (4.4)$$

Нехай невідомими будуть два останні модулі p_{k-1} та p_k. Тоді вираз (4.4) представимо у вигляді:

$$\begin{aligned} &p_{k-1}p_k\left(p_2p_3...p_{k-2}+p_1p_3...p_{k-2}+...+p_1p_2...p_{k-3}\right)+ \\ &+p_1p_2...p_{k-2}\left(p_{k-1}+p_{k-2}\right)=\pm 1. \end{aligned} \quad (4.5)$$

Введемо позначення:

$$p_{k-1,k}=\frac{a,b-p_1p_2...p_{k-2}}{p_2p_3...p_{k-2}+p_1p_3...p_{k-2}+...+p_1p_2...p_{k-3}}. \quad (4.6)$$

Підставивши вираз (4.6) у (4.5), після відповідних математичних перетворень матимемо умову, яка має виконуватися для визначення набору будь-якої кількості модулів МДФ СЗК:

$$\pm\left(p_2p_3...p_{k-2}+p_1p_3...p_{k-2}+...+p_1p_2...p_{k-3}\right)+\left(p_1p_2p_{k-2}\right)^2=ab. \quad (4.7)$$

Відповідно ліва частина вираз (4.7) має бути факторизована, на основі чого визначаються параметри *a* та *b*. Крім того, як випливає з формули (4.6), модулі p_k та p_{k-1} мають бути цілими числами, тобто

$$(a,b-p_1p_2...p_{k-2})\bmod\left|p_2p_3...p_{k-2}+p_1p_3...p_{k-2}+...+p_1p_2...p_{k-3}\right|=0. \quad (4.8)$$

Отже, вирази (4.7) та (4.8) визначають умови для знаходження будь-якої кількості модулів МДФ СЗК, два з яких невідомі.

4.2. Пошук чотирьох модулів модифікованої досконалої форми системи залишкових класів

Як приклад запропонованого методу розглянемо МДФ СЗК, яка складається з чотирьох модулів. Умови формул (4.6) – (4.8) відповідно трансформуються:

$$p_{3,4}=\frac{a,b-p_1p_2}{p_1+p_2};\ \pm(p_1+p_2)+(p_1p_2)^2=ab;\ (a,\ b\text{-}p_1p_2)\bmod(p_1+p_2). \quad (4.9)$$

З формули (4.3) можна визначити, що при k=4 модулі p_1 і p_2 мають мати різні знаки. Очевидно, що, вважаючи модуль p_1 додатним, найбільше варіантів буде, коли $p_2=-(p_1+1)$, оскільки в цьому разі третя умова виразу (4.9) зникає. Перші дві матимуть такий вигляд:

$$p_{3,4}=-\left(a,b+p_1^2+p_1\right);\ a\ \equiv\ c\ (\mathrm{mod}\quad z\ )\ . \quad (4.10)$$

Нехай p_1=7, p_2=–8. Тоді з формули (4.9) отримаємо: $p_{3,4}=-(a,b+56)$ і $ab=\pm1+3136=\begin{cases}3135=3\cdot5\cdot11\cdot19\\3137\ \text{– просте число.}\end{cases}$

Усі можливі варіанти систем з чотирьох модулів для МДФ СЗК при p_1=7, p_2=–8 подано в табл. 4.1 (в дужках вказана розрядність модулів та діапазона обчислень у двійковій системі числення).

Для побудови та дослідження графіка залежності модулів їх потрібно перенумерувати в порядку зростання абсолютної величини p_3, що відображено у табл. 4.2.

На рис. 4.1 відображено зміни значень модулів p_3 та p_4 залежно від номера модуля згідно з даними табл. 4.2 у логарифмічній шкалі з основою 2, на якій відразу можна визначити розрядності отриманих модулів у двійковій системі числення.

Таблиця 4.1

Можливі варіанти систем з чотирьох модулів для МДФ СЗК при p_1=7, p_2=–8 (в дужках – розрядність у двійковій системі числення)

№	p_1, p_2	*ab*	*a*	*b*	p_3	p_4	*P*
1	7 (3), -8 (4)	3135	1	3135	-57 (6)	-3191 (12)	10185672 (24)
2			-1	-3135	-55 (6)	3079 (12)	9483320 (24)
3			3	1045	-59 (6)	-1101 (11)	3637704 (22)
4			-3	-1045	-53 (6)	989 (10)	2935352 (22)
5			5	627	-61 (6)	-683 (10)	2333128 (22)
6			-5	-627	-51 (6)	571 (10)	1630776 (21)
7			11	285	-67 (7)	-341 (9)	1279432 (21)
8			-11	-285	-45 (6)	229 (8)	577080 (20)
9			15	209	-71 (7)	-265 (9)	1053640 (21)
10			-15	-209	-41 (6)	153 (8)	351288 (19)
11			19	165	-75 (7)	-221 (8)	928200 (20)
12			-19	-165	-37 (6)	109 (7)	225848 (18)
13			33	95	-89 (7)	-151 (8)	752584 (20)
14			-33	-95	-23 (5)	39 (6)	50232 (16)
15			55	57	-111 (7)	-113 (7)	702408 (20)
16			-55	-57	-1 (1)	1 (1)	56 (6)
17		3137	1	3137	-57 (6)	-3193 (12)	10192056 (24)
18			-1	-3137	-55 (6)	3081 (12)	9489480 (24)

Таблиця 4.2

Впорядкування модулів

№	1	2	3	4	5	6	7	8	9
p_3	1	23	37	41	45	51	53	55	55
p_4	1	39	109	153	229	571	989	3079	3081
№	10	11	12	13	14	15	16	17	18
p_3	57	57	59	61	67	71	75	89	111
p_4	3191	3193	1101	683	341	265	221	151	113

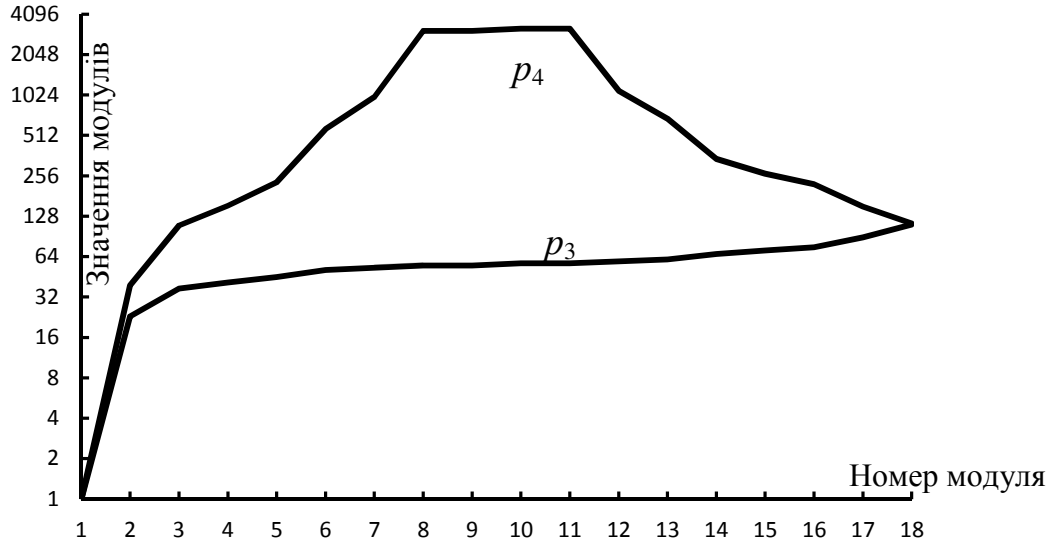


Рис. 4.1 Зміни значень модулів p_3 та p_4 при p_1=7, p_2=–8 залежно від номера модуля (згідно з даними табл. 4.2) у логарифмічній шкалі

Як можна з'ясувати з рис. 4.1, модуль p_3 відносно повільно зростає. Водночас графік для значення модуля p_4 зростає інтенсивніше, досягає плоского максимуму приблизно посередині номерного діапазону модулів, а потім спадає до значення модуля p_3.

Слід зазначити, що найбільший діапазон обчислень матимемо в тому разі, коли кожен наступний модуль є на одиницю більший від добутку абсолютних величин усіх попередніх модулів. Окрім того, за даними табл. 5.1 при

застосуванні за даних модулів МДФ СЗК розрядність чисел, над якими виконуються арифметичні операції, зменшується в 2–3 рази. Набір модулів 7, –8, –1, 1 у табл. 4.1 вказує, що числа 7 та –8 самостійно утворюють МДФ СЗК.

Чисельні розрахунки підтверджують, що для p_1=7 в інших випадках, крім p_2=–8, найбільша кількість варіантів наборів модулів буде при p_2=–9 та p_2=–11. Тоді рівняння (4.9) набудуть відповідно такого вигляду:

$$p_{3,4} = -\frac{a,b+63}{2};\ \pm 2 + 63^2 = ab;\ (a,b-63)\bmod 2 = 0;\quad (4.11)$$

$$p_{3,4} = -\frac{a,b+77}{4};\ \pm 4 + 77^2 = ab;\ (a,b-77)\bmod 4 = 0. \quad (4.12)$$

Звідси випливає, що $ab = \pm 2 + 3969 = \begin{cases} 3967 \\ 3971 = 11 \cdot 19 \cdot 19 \end{cases}$ для p_2=–9 і $ab = \pm 4 + 5929 = \begin{cases} 5925 = 3 \cdot 5 \cdot 5 \cdot 79 \\ 5933 = 17 \cdot 349 \end{cases}$ для p_2=-11. У табл. 4.3 наведені впорядковані значення абсолютних величин модулів, отриманих з формул (4.11) і (4.12) аналогічно до даних табл. 4.2 при p_2=–9 та p_2=–11, а також межа діапазону обчислень P (у дужках вказана розрядність представлених чисел).

Оскільки в розглянутих випадках параметри a та b є непарними числами, то третя умова з формули (4.11) для p_2=–9 виконується при всіх можливих значеннях a та b. Для p_2=–11 третя умова з формули (4.12) виконується тільки для половини з можливих варіантів параметрів a та b.

Таблиця 4.3

Впорядковані значення абсолютних величин модулів p_3 та p_4 при p_2=–9 та p_2=–11.

p_2	p_3, p_4	Номер модуля			
		1	2	3	4
9 (4)	p_3	22(5)	26(5)	31(5)	31(5)
	p_4	73(7)	149(8)	1952(11)	1954(11)
	P	101178 (17)	244062 (18)	3812256 (22)	3816162 (22))
11 (4)	p_3	13(4)	15(4)	18(5)	19(5)
	p_4	40(6)	68(7)	277(9)	1462(11)
	P	40040 (16)	78540 (17)	383922 (19)	2138906 (22)
p_2	p_3, p_4	Номер модуля			
		5	6	7	8
9 (4)	p_3	32(6)	32(6)	37(6)	41(6)
	p_4	2015(11)	2017(11)	212(8)	136(8)
	P	4062240 (22)	4066272 (22)	494172 (19)	351288 (19)
11 (4)	p_3	19(5)	20(5)	23(5)	38(6)
	p_4	1464(11)	513(10)	118(7)	39(6)
	P	2141832 (22)	790020 (20)	208978 (18)	114114 (17)

На рис. 4.2 відображені графіки залежності значень модулів p_3 та p_4 (суцільною лінією – для p_2=–9, пунктирною –для p_2=–11) від номера модуля згідно з даними табл. 4.3 у логарифмічній шкалі з основою 2.

Як можна з'ясувати з рис. 4.2, модуль p_3 відносно повільно збільшується. Графік для значення модуля p_4 збільшується інтенсивніше, досягає плоского максимуму (причому для p_2=–9 максимум ширший) посередині номерного діапазону модулів, а потім спадає.

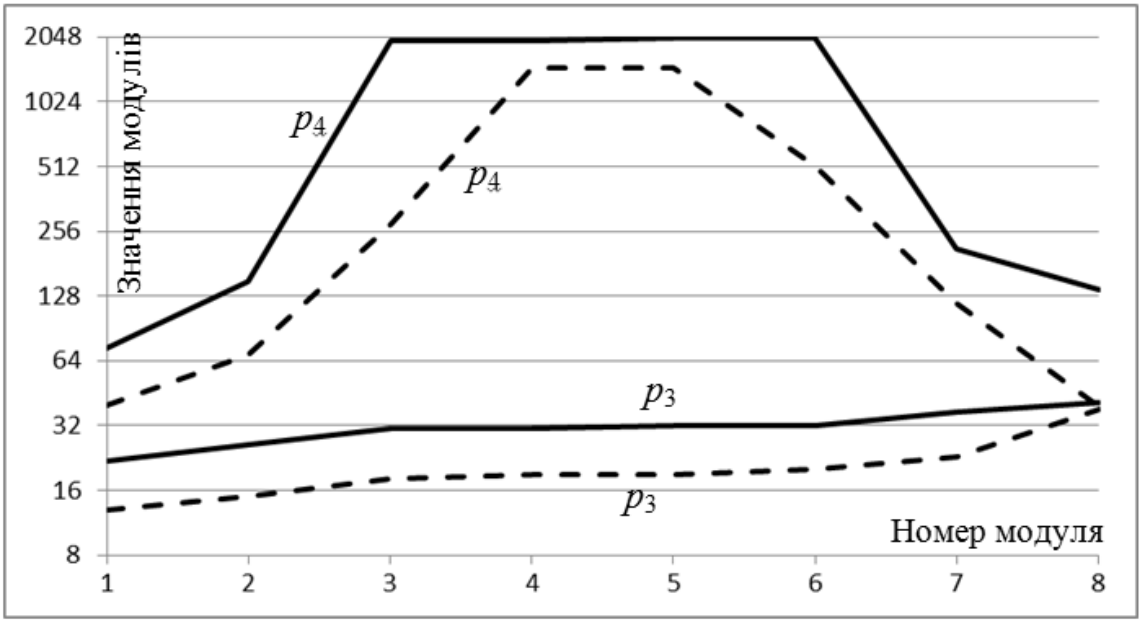


Рис. 4.2. Зміни значень модулів p_3 та p_4 при p_1=7, p_2=–9 (суцільна лінія) і p_2=–11 (пунктирна лінія) залежно від номера модуля (згідно з даними табл. 4.3)

Слід зауважити, що не в усіх випадках множники, на які факторизується добуток *ab*, дають можливість отримати відповідні значення модулів. Це залежить від виконання третьої умови формули (4.9). Розглянемо випадок, коли p_1=7, p_2=–10. Тоді з формули (4.9) отримаємо: $p_{3,4} = -\dfrac{a,b+70}{3}$; ($a$, b+70)mod 3=0; $ab = \pm 3 + 4900 = \begin{cases} 4903 \\ 4897 = 59 \cdot 83. \end{cases}$ Усі можливі варіанти модулів подано в табл. 4.4.

Знову ж розрядність чисел, над якими виконуються операції, зменшується в 2–3 рази. Крім того, в половині з можливих випадків не існує цілочисельних значень p_3 і p_4.

Таблиця 4.4

Можливі варіанти систем з чотирьох модулів для МДФ СЗК при p_1=7, p_2=-10 (в дужках – розрядність у двійковій системі числення)

№	p_1, p_2	ab	a	b	p_3	p_4	P
1	7 (3), -10 (4)	4903 (13)	1	4903	не існує	не існує	не існує
2			–1	–4903	–23 (5)	1611 (11)	2593710 (22)
3		4897 (13)	1	4897	не існує	не існує	не існує
4			–1	–4897	–23 (5)	1609 (11)	2590490 (22)
5			59	83	–43 (6)	–51 (6)	153510 (18)
6			–59	–83	не існує	не існує	не існує

У таблиці 4.5 представлені інші можливі варіанти наборів модулів МДФ СЗК при p_1=7, отримані відповідно до умов (4.9).

Таблиця 4.5.

Можливі варіанти систем з чотирьох модулів для МДФ СЗК при p_1=7, p_2= –12, –13, –15 (в дужках вказана розрядність модулів та діапазону обчислень у двійковій системі числення)

p_2	p_3	p_4	P
–12 (4)	–17 (5)	–1427 (11)	2037756 (22)
	–17 (5)	–1429 (11)	2040612 (22)
	–19 (5)	–145 (8)	231420 (18)
–13 (4)	–15 (4)	1364 (11)	1861860 (21)
	–15 (4)	1366 (11)	1864590 (21)
	–16 (5)	–291 (9)	423696 (19)
–15 (4)	–16 (5)	–73 (7)	122640 (17)

Отже, більшість варіантів отримана при a=±1, коли четвертий модуль на одиницю відрізняється від добутку трьох попередніх, що відповідає максимальній межі діапазону обчислень.

4.3 Приклад побудови п'ятимодульної модифікованої досконалої форми системи залишкових класів

Для спрощення розрахунків обмежимося значенням першого модуля p_1=3, кількістю модулів k=5 [222] і, не зменшуючи загальності рішення, вважатимемо, що

$$p_1=3<|p_2|<|p_3|<|p_4|<|p_5|. \qquad (4.13)$$

Чисельні розрахунки підтверджують, що виконання цієї умови для цілочисельних розв'язків рівності (4.3) можливе тільки тоді, коли p_2, p_3<0.

На рис. 4.3 для прикладу зображено вигляд поверхні, що характеризує залежність модуля p_5 від p_3 та p_4 згідно з виразом:

$$p_5=\frac{-1-3p_2p_3p_4}{p_2p_3p_4+3(p_3p_4+p_2p_4+p_2p_3)}, \qquad (4.14)$$

який отримується з формули (4.3), при p_1=3 та p_2=-5.

Тоді, вважаючи невідомими два останні модулі p_4 та p_5, з формули (4.3) можна отримати діофантове рівняння другого порядку для їхнього пошуку:

$$p_4p_5(p_2p_3+3(p_2+p_3))+3p_2p_3(p_4+p_5)=\pm1. \qquad (4.15)$$

Введемо такі позначення:

$$p_{4,5}=\frac{a,b-3p_2p_3}{p_2p_3+3(p_2+p_3)}. \qquad (4.16)$$

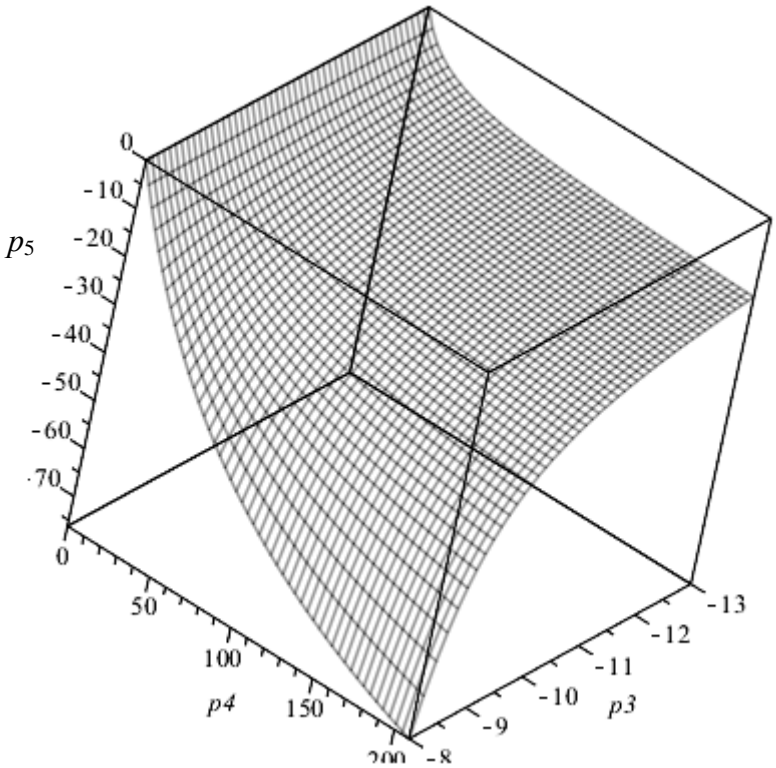


Рис. 4.3. Вигляд поверхні, що характеризує залежність модуля p_5 від p_3 та p_4 при p_2=–5

Після підстановки формули (4.16) у (4.15) та відповідних математичних перетворень можна отримати вираз для цілочисельного розв'язку виразу (4.16):

$$\pm\left(p_2 p_3 + 3(p_2 + p_3)\right) + (3 p_2 p_3)^2 = ab\,. \qquad (4.17)$$

Ліву частину виразу (4.17) потрібно факторизувати, на основі чого визначаються параметри *a* і *b*. Крім того, модулі p_4 та p_5 мають бути цілими числами. Тому з формули (4.16) випливає:

$$(a, b - 3p_2 p_3) \bmod \left(\left| p_2 p_3 + 3(p_2 + p_3) \right|\right) = 0\,. \qquad (4.18)$$

Вирази (4.17) і (4.18) визначають умови для знаходження п'яти модулів МДФ СЗК, два з яких невідомі.

При p_2=–4, p_3=–7 вирази (4.16)–(4.18) перетворюються таким чином: $p_{4,5} = \frac{a, b-84}{-5}$; $(a, b-84) \bmod 5 = 0$; $ab = \pm 5 + 7056 = \begin{cases} 7051 = 11 \cdot 641 \\ 7061 = 23 \cdot 307. \end{cases}$ У табл. 4.5 наведено всі можливі цілочисельні значення *a* і *b*, що визначаються факторизацією добутку *a*·*b*, а також випадки, коли існують набори модулів МДФ СЗК і діапазон відповідних обчислень.

Таблиця 4.6

Можливі варіанти систем із п'яти модулів для МДФ СЗК при p_1=3, p_2=-4, p_3=-7 (в дужках – розрядність у двійковій системі числення).

№	p_1, p_2	*ab*	*a*	*b*	p_4	p_5	*P*
1		7051	1	7051	не існує		
2			–1	–7051	17(5)	1427(11)	2037756(21)
3	p_1=3(2)		11	641	не існує		
4	p_2=–4(3)		–11	–641	19(5)	145(8)	231420(18)
5	p_3=–7(3)	7061	1	7061	не існує		
6			–1	–7061	17(5)	1429(11)	2040612(21)
7			23	23	не існує		
8			–23	–23	не існує		

За даним табл. 4.6 можна визначити, що модулі p_4 та p_5 набувають додатних значень, а розрядність чисел, над якими виконуються арифметичні операції, приблизно зменшується вдвічі. Крім того, у п'яти з восьми можливих випадків, що утворюються при факторизації, цілочисельних наборів модулів не існує. Значенню p_4=17 відповідає два значення модуля p_5,

кожне з яких на одиницю відрізняється від добутку чотирьох попередніх модулів.

Зрозуміло, що найбільше шуканих наборів буде тоді, коли модулі p_1, p_2, p_3 самостійно утворюють МДФ СЗК, оскільки в цьому разі $p_1p_2 + p_2p_3 + p_1p_3 = \pm 1$ і умова рівності (4.18) виконується завжди. Кількість різних варіантів визначатиметься кількістю множників при факторизації лівої частини виразу (4.17). Такі випадки доцільно розглянути детальніше.

При p_2=–4, p_3=–11 з виразів (4.16) і (4.17) можна отримати: $p_{4,5} = \frac{a,b-132}{-1}$; $ab = \pm 1 + 17424 = \begin{cases} 17423 = 7 \cdot 19 \cdot 131 \\ 17425 = 5 \cdot 5 \cdot 17 \cdot 41. \end{cases}$ Усі можливі варіанти модулів та величина діапазону обчислень подані в табл. 4.7.

За даними табл. 4.7 можна визначити, що p_4 набуває тільки додатних значень, а знака p_5 протилежний до знаку *a* і *b.* Розрядність чисел, над якими виконуватимуться арифметичні операції, зменшується у 2–2,5 разу. Рядок 7, в якому p_4, p_5=±1, вказує, що за даний набір з трьох модулів p_1=3, p_2=–4, p_3=–11 утворює МДФ СЗК. Значенням p_4=131 та p_4=133 відповідає по два значення модуля p_5, абсолютні величини яких на одиницю відрізняються від добутку чотирьох попередніх модулів.

Набір p_1=3, p_2=–4, p_3=–13 також утворює МДФ СЗК. Тому з формул (4.16) і (4.17) отримується: $p_{4,5} = \frac{a,b-156}{1}$; $ab = \pm 1 + 24336 = \begin{cases} 24335 = 5 \cdot 31 \cdot 157 \\ 24337 = \text{просте число}. \end{cases}$ Усі можливі варіанти модулів, діапазон можливих обчислень та відповідні розрядності подано в табл. 4.8.

За даними табл. 4.8 можна визначити, що модуль p_4 набуває тільки від'ємних значень, а знак модуля p_5 збігається із

знаком *a* та *b.* Розрядність чисел, над якими виконуватимуться відповідні арифметичні операції, зменшується в 2–2,5 разу. Рядок 7 табл. 4.8 вказує, що за даний набір з трьох модулів утворює МДФ СЗК. Значенням p_4=–155 та p_5=–157 відповідає по два значення модуля p_5, абсолютні величини яких на одиницю відрізняються від добутку абсолютних величин чотирьох попередніх модулів.

Таблиця 4.7

Можливі варіанти систем з п'яти модулів для МДФ СЗК при p_1=3, p_2=–4, p_3=–11 (в дужках – розрядність у двійковій системі числення)

№	p_1, p_2, p_3	ab	a	b	p_4	p_5	P
1	3 (2), –4 (3), –11 (4)	17423	1	17423	131 (8)	–17291 (15)	298995972 (29)
2			–1	–17423	133 (8)	17555 (15)	308195580 (29)
3			7	2489	125 (7)	–2357 (12)	38890500 (26)
4			–7	–2489	139 (8)	2621 (12)	48090108 (26)
5			19	917	113 (7)	–785 (10)	11709060 (25)
6			–19	–917	151 (8)	1049 (11)	20908668 (25)
7			131	133	1 (1)	–1(1)	132 (8)
8			–131	–133	263 (9)	265 (9)	9199740 (24)
9		17425	1	17425	131 (8)	–17293 (15)	299030556 (29)
10			–1	–17425	133 (8)	17557 (15)	308230692 (29)
11			5	3485	127 (7)	–3353 (12)	56209692 (26)
12			–5	–3485	137 (8)	3617 (12)	65409828 (26)
13			17	1025	115 (7)	–893 (10)	13555740 (24)
14			–17	–1025	149 (8)	1157 (11)	22755876 (25)
15			25	697	107 (7)	–565 (10)	7980060 (23)
16			–25	–697	157 (8)	829 (10)	17180196 (25)
17			41	425	91 (7)	–293 (9)	3519516 (22)
18			–41	–425	173 (8)	557 (10)	12719652 (21)
19			85	205	47 (6)	–73 (7)	452892 (19)
20			–85	–205	217 (8)	337 (9)	9653028 (23)

Таблиця 4.8

Можливі варіанти систем з п'яти модулів для МДФ СЗК при p_1=3, p_2=–4, p_3=–13 (в дужках – розрядність у двійковій системі числення)

№	p_1, p_2, p_3	ab	a	b	p_4	p_5	P
1	3 (2), –4 (3), –13 (4)	24335	1	24335	–155 (8)	24179 (15)	584648220 (30)
2			–1	–24335	–157 (8)	–24491 (15)	599833572 (30)
3			5	4867	–151 (8)	4711 (13)	110972316 (27)
4			–5	–4867	–161 (8)	–5023 (13)	126157668 (27)
5			31	785	–125 (7)	629 (10)	12265500 (24)
6			–31	–785	–187 (8)	–941 (10)	27450852 (25)
7			155	157	-1 (1)	1(1)	156 (8)
8			–155	–157	–311 (9)	–313 (9)	15185508 (24)
9		24337	1	24337	–155 (8)	24181 (15)	584696580 (30)
10			–1	–24337	–157 (8)	–24493 (15)	599882556 (30)

У двох останніх випадках для проведення подальших досліджень розподілу абсолютних величин знайдених модулів їх потрібно перенумерувати в порядку зростання $|p_4|$, (табл. 4.9, і 4.10).

Таблиця 4.9

Впорядкування модулів за зростанням $|p_4|$ при p_1=3, p_2=–4, p_3=–11

№	1	2	3	4	5	6	7	8	9	10
p_4	1	47	91	107	113	115	125	127	131	131
p_5	1	73	293	565	785	893	2357	3353	17291	17293
№	11	12	13	14	15	16	17	18	19	20
p_4	133	133	137	139	149	151	157	173	217	263
p_5	17555	17557	3617	2621	1157	1049	829	557	337	265

Таблиця 4.10

**Впорядкування модулів за зростанням $|p_4|$
при p_1=3, p_2=–4, p_3=–13**

№	1	2	3	4	5	6	7	8	9	10
p_4	1	125	151	155	155	157	157	161	187	311
p_5	1	629	4711	24179	24181	24492	24493	5023	941	313

На рис. 4.4 відображено зміни значень модулів p_4 та p_5 залежно від номера модуля згідно з даними табл. 4.9 і 4.10 у логарифмічній шкалі з основою 2, що вказує на розрядності отриманих модулів у двійковій системі числення.

Як видно з рис. 4.4, в обох випадках значення $|p_4|$ відносно повільно зростає. Водночас графік для $|p_5|$ зростає набагато інтенсивніше, досягає до плоского максимуму посередині номерного діапазону модулів, а потім спадає до значення $|p_4|$.

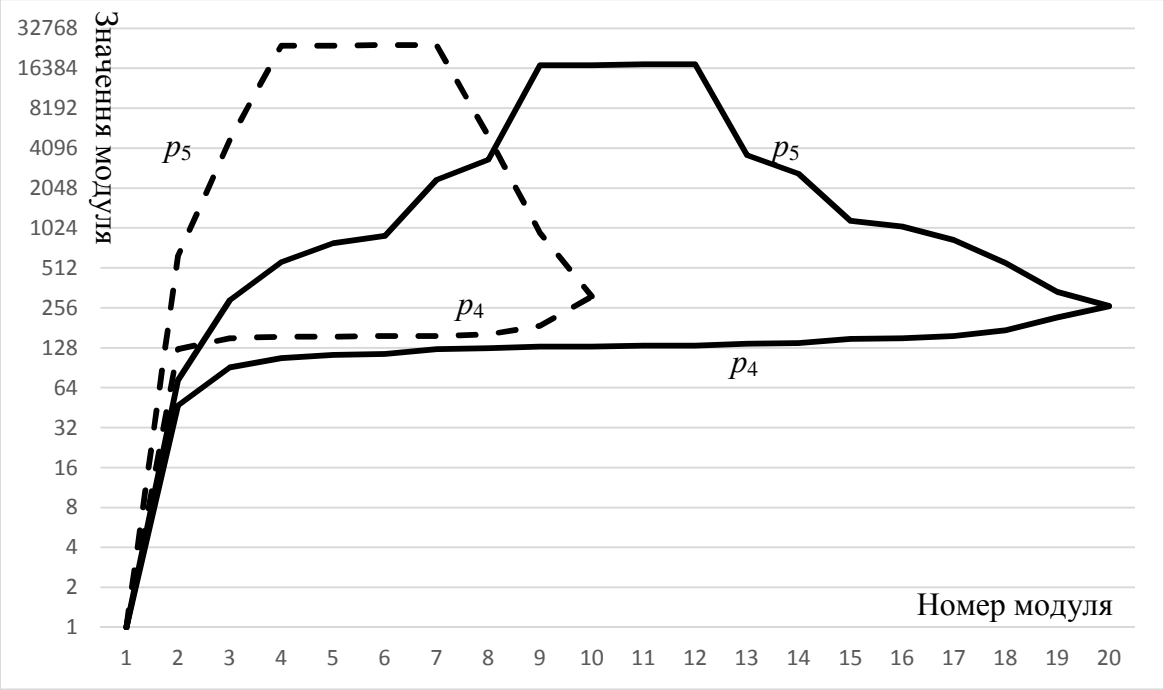


Рис. 4.4. Зміни значень модулів p_4 та p_5 при p_1=3, p_2=–4, p_3=–11 (суцільна лінія) і p_3=–13 (пунктирна лінія) залежно від номера модуля (згідно з даними табл. 4.9 і 4.10)

Набір модулів p_1=3, p_2=–5, p_3=–7 теж утворює МДФ СЗК, тому вирази (4.16) і (4.17) можна записати таким чином: $p_{4,5} = \frac{a,b-105}{-1} = 105 - a,b$ і $ab = \pm 1 + 11025 = \begin{cases} 11024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 13 \cdot 53 \\ 11026 = 2 \cdot 37 \cdot 149. \end{cases}$

Усі можливі варіанти систем з п'яти модулів для МДФ СЗК при p_1=3, p_2=–5, p_3=–7 подано в табл. 4.11.

Результати чисельного експерименту (див. табл. 4.11) підтверджують, що p_4 набуває додатних значень, а знак p_5 протилежний до знака *a* і *b.* Розрядність чисел, над якими виконуватимуться арифметичні операції, зменшується у 2–3 рази. Рядок 17, в якому p_4, p_5=±1, вказує, що за даний набір з трьох модулів p_1=3, p_2=–5, p_3=–7 утворює МДФ СЗК.

Значенням p_4=103, 104, 106, 107 відповідає по два значення модуля p_5, що зумовлене наявністю спільних множників 1 і 2 в обох випадках розкладу добутку *ab*.

Модулі p_1=3, p_2=–5, p_3=–8 теж утворюють МДФ СЗК, тому вирази (4.16) і (4.17) дають змогу отримати такі результати: $p_{4,5} = a,b - 120$ і $ab = \pm 1 + 14400 = \begin{cases} 14399 = 7 \cdot 11 \cdot 11 \cdot 17 \\ 14401 = \text{просте} \end{cases}$ Усі можливі варіанти систем з п'яти модулів для МДФ СЗК при p_1=3, p_2=–5, p_3=–8 подано в табл. 4.12.

За даними табл. 4.12 можна визначити, що p_4 набуває тільки від'ємних значень, а знак p_5 збігається із знаком *a* та *b.* Розрядність чисел, над якими виконуватимуться арифметичні операції, зменшується у 2–3 рази. Рядок 11 вказує, що набір p_1=3, p_2=–5, p_3=–8 утворює МДФ СЗК. Значенням p_4=–119 та p_5=–121 відповідає по два значення модуля p_5, абсолютні величини яких на одиницю відрізняються від добутку абсолютних величин чотирьох попередніх модулів.

Таблиця 4.11

Можливі варіанти систем з п'ятьох модулів для МДФ СЗК при p_1=3, p_2=-5, p_3=-7 (в дужках – розрядність в двійковій системі числення).

№	p_1, p_2, p_3	*ab*	*a*	*b*	p_4	p_5	*P*
1	3 (2), –5 (3), –7 (3)	11024 (14)	1	11024	104 (7)	–10919 (14)	119235480 (27)
2			–1	–11024	106 (7)	11129 (14)	123865770 (27)
3			2	5512	103 (7)	–5407 (13)	58476705 (26)
4			–2	–5512	107 (7)	5617 (13)	63106995 (26)
5			4	2756	101 (7)	–2651 (12)	28113855 (25)
6			–4	–2756	109 (7)	2861 (12)	32744145 (25)
7			8	1378	97 (7)	–1273(11)	12965505 (24)
8			–8	–1378	113 (7)	1483 (11)	17595795 (25)
9			13	848	92 (7)	–743 (10)	7177380 (23)
10			–13	–848	118 (7)	953 (10)	11807670 (24)
11			16	689	89 (7)	–584 (10)	5457480 (23)
12			–16	–689	121 (7)	794 (10)	10087770 (24)
13			26	424	79 (7)	–319 (9)	2646105 (22)
14			–26	–424	131 (8)	529 (10)	7276395 (23)
15			52	212	53 (6)	–107 (7)	595455 (20)
16			–52	–212	157 (8)	317 (9)	5225745 (23)
17			104	106	1 (1)	–1 (1)	105 (7)
18			–104	–106	209 (8)	211 (8)	4630395 (23)
19		11026 (14)	1	11026	104 (7)	–10921 (14)	119257320 (27)
20			–1	–11026	106 (7)	11131 (14)	123888030 (27)
21			2	5513	103 (7)	–5408 (13)	58487520 (26)
22			–2	–5513	107 (7)	5618 (13)	63118230 (26)
23			37	298	68 (7)	–193 (8)	1378020 (21)
24			–37	–298	142 (8)	403 (9)	6008730 (23)
25			74	149	31 (5)	–44 (6)	143220 (18)
26			–74	–149	179 (8)	254 (8)	4773930 (23)

Таблиця 4.12

Можливі варіанти систем з п'ятьох модулів для МДФ СЗК при p_1=3, p_2=-5, p_3=-8 (в дужках – розрядність в двійковій системі числення)

№	p_1, p_2, p_3	ab	a	b	p_4	p_5	P
1	3 (2), –5 (3), –8 (3)	14399 (14)	1	14399	–119 (7)	14279 (14)	203904120 (28)
2			–1	–14399	–121 (7)	–14519 (14)	210815880 (28)
3			7	2057	–113 (7)	1937 (11)	26265720 (25)
4			–7	–2057	–127 (7)	–2177 (12)	33177480 (25)
5			11	1309	–109 (7)	1189 (11)	15552120 (24)
6			–11	–1309	–131 (8)	–1429 (11)	22463880 (25)
7			17	847	–103 (7)	727(10)	8985720 (24)
8			–17	–847	–137 (8)	–967 (10)	15897480 (24)
9			77	187	–43 (6)	67 (7)	345720 (19)
10			–77	–187	–197 (8)	–307 (9)	7257480 (23)
11			119	121	–1 (1)	1 (1)	120 (7)
12			–119	–121	–239 (8)	–241 (8)	6911880 (23)
13		14401 (14)	1	14401	–119 (7)	14281 (14)	203932680 (28)
14			–1	–14401	–121 (7)	–14521 (14)	210844920 (28)

Аналогічно до випадку p_1=3, p_2=–4, в табл. 4.13 і 4.14 представлено впорядкування модулів за зростанням абсолютної величини p_4.

Таблиця 4.13

Впорядкування модулів по зростанню $|p_4|$ при p_1=3, p_2=–5, p_3=–7

№	1	2	3	4	5	6	7	8	9	10	11	12	13
p_4	1	31	53	68	79	89	92	97	101	103	103	104	104
p_5	1	44	107	193	319	584	743	1273	2651	5407	5408	10919	10921
№	14	15	16	17	18	19	20	21	22	23	24	25	26
p_4	106	106	107	107	109	113	118	121	131	142	157	179	209
p_5	11129	11131	5617	5618	2861	1483	953	794	529	403	317	254	211

Таблиця 4.14

**Впорядкування модулів по зростанню $|p_4|$
при p_1=3, p_2=–5, p_3=–8**

№	1	2	3	4	5	6	7
p_4	1	43	103	109	113	119	119
p_5	1	67	727	1189	1937	14279	14281
№	8	9	10	11	12	13	14
p_4	121	121	127	131	137	197	239
p_5	14519	14521	2177	1429	967	307	241

На рис. 4.5 відображено зміни значень модулів p_4 та p_5 залежно від номера модуля згідно з даними табл. 4.13 4.14 у логарифмічній шкалі з основою 2, що вказує на розрядності отриманих модулів у двійковій системі числення.

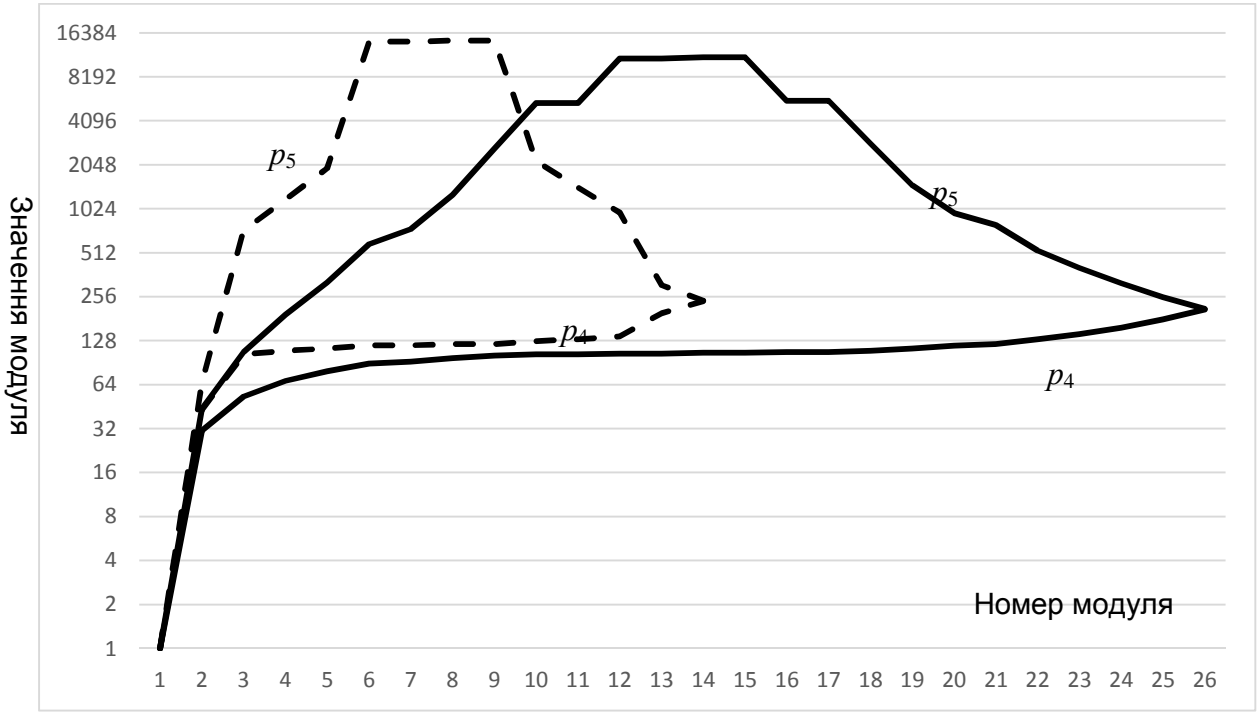


Рис. 4.5 Зміни значень модулів p_4 та p_5 при p_1=3, p_2=–5, p_3=–7 (суцільна лінія) і p_3=–8 (пунктирна лінія) залежно від номера модуля (згідно з даними табл. 4.13 і 4.14)

Характеристики відповідних графіків, зображених на рис. 4.4 і 4.5, є подібними, але плоский максимум на рис. 4.5 має більше значення.

У табл. 4.15 впорядковано значення діапазону обчислень відповідно до даних табл. 4.10, 4.11, 4.13 і 4.14 за зростанням абсолютної величини p_4.

На рис. 4.6 зображено графік залежності діапазону обчислень згідно з номером у табл. 4.15 для різних значень p_2, p_3.

Подібно до графіків для p_5 на рис. 4.4 і 4.5, залежність діапазону обчислень P спочатку різко зростає, посередині номерного діапазону має плоский максимум, який для фіксованого p_2 розміщується вище при більшому $|p_3|$. При подальшому збільшенні номера криві на графіках повільно спадають.

У табл. 4.16 для p_1=3 згідно з аналогічними чисельними дослідженнями наведено інші можливі набори модулів при різних p_2, p_3, які утворюють МДФ СЗК.

У табл. 4.16 відображено, що кількість наборів з п'яти модулів значно зменшується, якщо перші три з них не утворюють МДФ СЗК. Це пов'язано з необхідністю виконання умови рівності (4.18). Розрядність чисел, над якими виконуватимуться арифметичні операції, зменшується у 2–3 рази. Найбільший діапазон обчислень при заданих першому модулю та їхній кількості буде тоді, коли абсолютні величини всіх наступних на одиницю більші від добутку абсолютних величин попередніх.

4.4. Метод розширення набору модулів модифікованої досконалої форми системи залишкових класів

Для демонстрації методу та спрощення розрахунків обмежимо наші міркування п'ятьма модулями, перші три з яких

утворюють ДФ СЗК [223]. Єдиним можливим варіантом є набір p_1=2, p_2=3, p_3=5.

Таблиця 4.15

Впорядкування значення діапазона обчислень *P* по зростанню абсолютної величини p_4 для різних величин p_2, p_3 при p_1=3

№	*P* (p_2=–4, p_3=–11)	*P* (p_2=–4, p_3=–13)	*P* (p_2=–5, p_3=–7)	*P* (p_2=–5, p_3=–8)
1	132	156	105	120
2	452892	12265500	143220	345720
3	3519516	110972316	595455	8985720
4	7980060	584648220	1378020	15552120
5	11709060	584696580	2646105	26265720
6	13555740	599833572	5457480	203904120
7	38890500	599882556	7177380	203932680
8	56209692	126157668	12965505	210815880
9	298995972	27450852	28113855	210844920
10	299030556	15185508	58476705	33177480
11	308195580		58487520	22463880
12	308230692		119235480	15897480
13	65409828		119257320	7257480
14	48090108		123865770	6911880
15	22755876		123888030	
16	20908668		63106995	
17	17180196		63118230	
18	12719652		32744145	
19	9653028		17595795	
20	9199740		11807670	
21			10087770	
22			7276395	
23			6008730	
24			5225745	
25			4773930	
26			4630395	

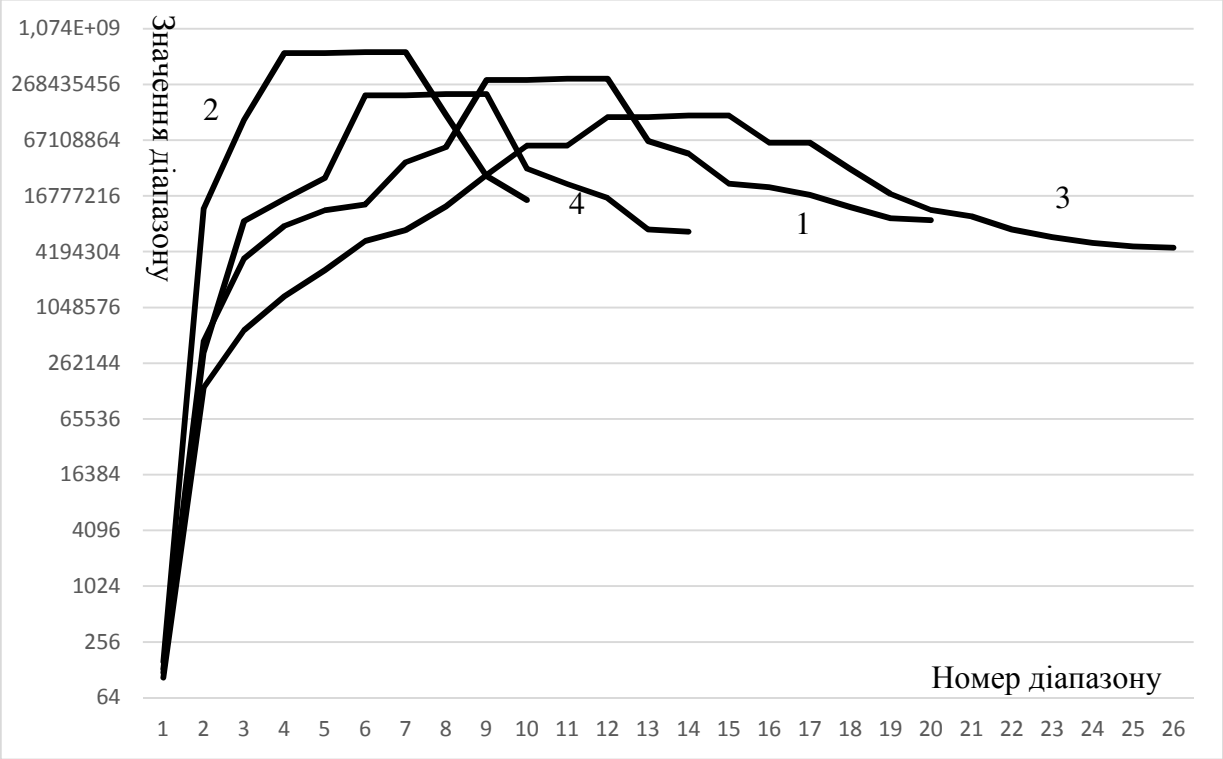


Рис. 4.6. Графік залежності діапазону обчислень згідно з номером у табл. 4.15 для різних значень p_2, p_3 (1 – p_2=–4, p_3=–11, 2– p_2=–4, p_3=–13, 3 – p_2=–5, p_3=–7, 4 – p_2=–5, p_3=–8)

Таблиця 4.16

Інші можливі набори модулів, які утворюють МДФ СЗК при p_1=3

№	p_2, p_3	p_4	p_5	P
1	–4 (3), –17 (5)	–41 (6)	–8363 (14)	69948132 (27)
2		–41 (6)	–8365 (14)	69964860 (27)
3	–5 (3), –11 (4)	–28 (5)	–149 (8)	688380 (20)
4		–23 (5)	949 (10)	3601455 (22)
5		–19 (5)	98 (7)	307230 (19)
6		–17 (5)	61 (6)	171105 (18)
7		–13 (4)	29 (5)	62205 (16)
8	–7 (3), –10 (4)	–11 (4)	2309 (12)	5333790 (23)
9		–11 (4)	2311 (12)	5338410 (23)

Тоді вираз (4.3) набуде такого вигляду:

$$\frac{31}{30}+\frac{1}{p_4}+\frac{1}{p_5}=1\pm\frac{1}{30\,p_4\,p_5}. \qquad (4.19)$$

Після відповідних математичних перетворень (4.19) трансформується в таку умову:

$$(31-30k)p_4\,p_5+30(p_4+p_5)=\pm 1. \qquad (4.20)$$

На рис. 4.7 зображений графік залежності p_5 від p_4 при s=1. Так, при p_4<–30 та p_4>1/30 модуль p_5 набуває від'ємних значень, в інших випадках модуль p_5 додатний.

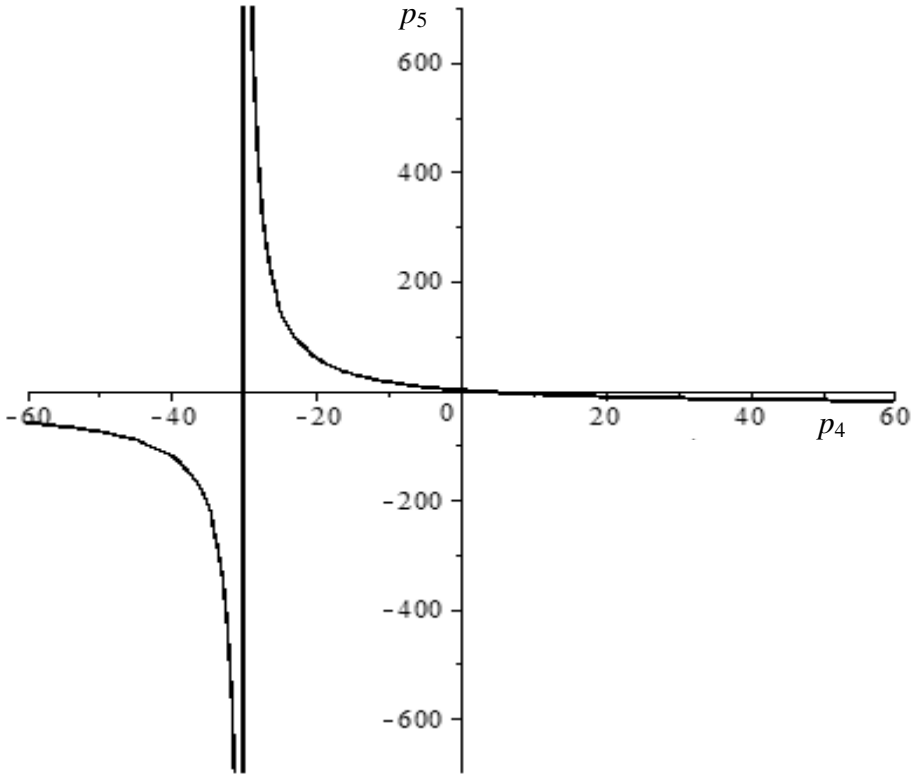


Рис. 4.7. Графік залежності p_5 від p_4 при k=1

Ввівши заміну $p_{4,5} = \dfrac{a,b-30}{31-30s}$, отримаємо вираз для цілочисельного розв'язку рівності (4.20): $ab = \pm(31-30s)+30^2$.

Розглянемо різні значення параметра *s*:

1) *s*=0. Тоді $ab = \pm 31+30^2 = \begin{cases} 931 = 7\cdot 7\cdot 19; \\ 869 = 11\cdot 79. \end{cases}$

У табл. 4.17 подано всі можливі цілочисельні значення *a* і *b*, що визначаються факторизацією, а також випадки, коли набори модулів МДФ СЗК існують, та відповідний їм діапазон обчислень.

Таблиця 4.17

Можливі варіанти систем із п'яти модулів для МДФ СЗК при p_1=2, p_2=3, p_3=5 і *s*=0 (в дужках – розрядність в двійковій системі числення)

№	*ab*	*a*	*b*	p_4	p_5	*P*
1	869	1	869	не існує		
2		–1	–869	–1	–29 (5)	870 (10)
3		11	79	не існує		
4		–11	–79	не існує		
5	931	1	931	не існує		
6		–1	–931	–1	–31 (5)	930 (10)
7		7	133	не існує		
8		–7	–133	не існує		
9		19	49	не існує		
10		–19	–49	не існує		

За даними табл. 4.17 можна визначити, що розрядність чисел, над якими виконуються арифметичні операції, приблизно зменшується вдвічі. Значення p_4=-1 вказує, що набори із чотирьох модулів 2, 3, 5, 29 і 2, 3, 5, 31 утворюють

МДФ СЗК, в яких кожен наступний модуль відрізняється на одиницю від добутку попередніх. Крім того, у восьми із десяти можливих випадків, які утворюються при факторизації, відповідних цілочисельних наборів модулів не існує;

2) s=1. Тоді $ab = \pm 1 + 30^2 = \begin{cases} 901 = 17 \cdot 53 \\ 899 = 29 \cdot 31 \end{cases}$. При заданій умові будь-яким значенням a і b відповідатимуть цілочисельні модулі p_4 та p_5. Результати наведено в табл. 4.18.

Таблиця 4.18

Можливі варіанти систем із п'яти модулів для МДФ СЗК при p_1=2, p_2=3, p_3=5 і s=1 (в дужках – розрядність в двійковій системі числення)

№	*ab*	*a*	*b*	p_4	p_5	*P*
1	899	1	899	–29 (5)	869 (10)	756030 (20)
2		–1	–899	–31 (5)	–929 (10)	863970 (20)
3		29	31	–1 (1)	1 (1)	30 (5)
4		–29	–31	–59 (6)	–61 (6)	107970 (17)
5	901	1	901	–29 (5)	871 (10)	757770 (20)
6		–1	–901	–31 (5)	–931 (10)	865830 (20)
7		17	53	–13 (4)	23 (5)	8970 (14)
8		–17	–53	–47 (6)	–83 (7)	117030 (17)

Дані табл. 4.18 підтверджують, що розрядність чисел, над якими виконуються арифметичні операції, зменшується приблизно у 2–3 рази. Третій рядок у табл. 4.18 вказує, що модулі 2, 3, 5 утворюють ДФ СЗК. Модуль p_4 завжди від'ємний, знак модуля p_5 збігається із знаком параметрів a і b, причому у

цьому випадку $p_4>-30$, що узгоджується з графіком на рис. 4.7. Найбільший діапазон обчислень буде в тому разі, коли абсолютна величина кожного наступного модуля більша на одиницю від добутку абсолютних величин попередніх модулів.

Чисельні розрахунки підтверджують, що при інших значеннях параметра s отримані модулі відрізнятимуться від знайдених при $s=0$, $s=1$ лише знаком. Для проведення подальших досліджень розподілу абсолютних величин усіх отриманих наборів модулів їх потрібно перенумерувати в порядку зростання $|p_4|$ (табл. 4.19).

Таблиця 4.19

Впорядкування модулів по зростанню $|p_4|$ при $p_1=2$, $p_2=3$, $p_3=5$

№	1	2	3	4	5	6	7	8	9	10
p_4	1	1	1	13	29	29	31	31	47	59
p_5	1	29	31	23	869	871	929	931	83	61

На рис. 4.8 відображено зміни значень модулів p_4 та p_5 залежно від номера модуля згідно з даними табл. 4.19 у логарифмічній шкалі, не враховуючи значення $p_4=1$.

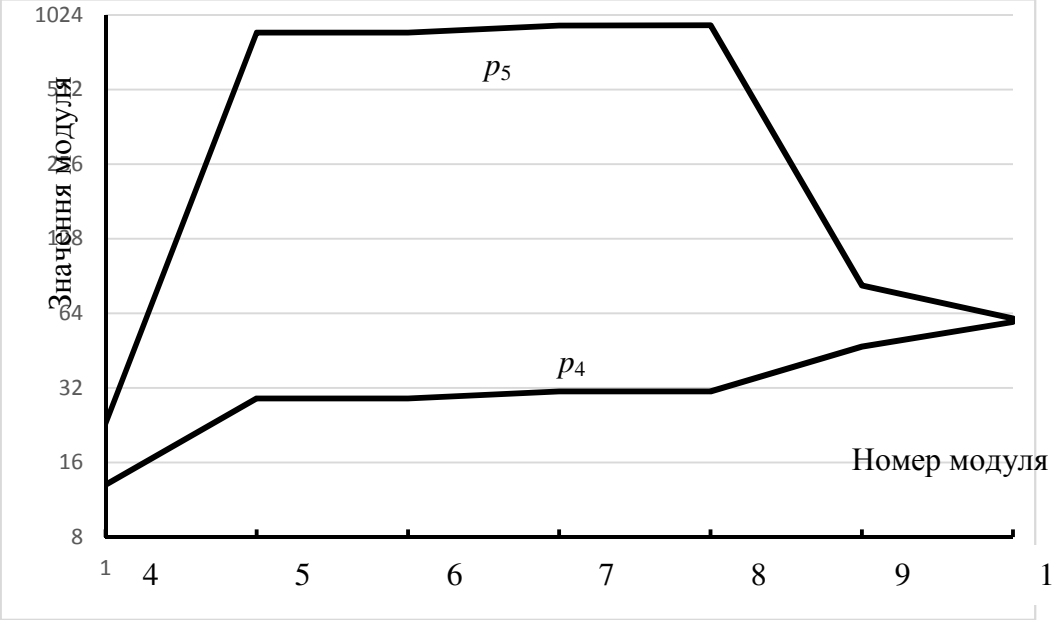


Рис. 4.8. Зміни значень модулів p_4 та p_5 при p_1=2, p_2=3, p_3=5 залежно від номера модуля (згідно з даними табл. 4.19)

Отже, значення $|p_4|$ відносно повільно зростає. Водночас графік для $|p_5|$ зростає набагато інтенсивніше, досягає плоского максимуму, а потім спадає до значення $|p_4|$.

РОЗДІЛ 5
ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ ПОШУКУ МОДУЛІВ ДОСКОНАЛОЇ ТА МОДИФІКОВАНОЇ ДОСКОНАЛОЇ ФОРМ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ ТА ЇХНЄ ЗАСТОСУВАННЯ

5.1. Програмна реалізація методу підбору модулів досконалої форми системи залишкових класів

Програма для підбору модулів ДФ СЗК написана мовою Java. Обрана мова програмування має такі переваги: простий синтаксис; повна підтримка об'єктно орієнтованого підходу; відносно висока продуктивність; архітектурна незалежність і портативність. Для факторизації чисел можна використати одну зі сторонніх бібліотек, наприклад: GMP-ECM. Функція primeFactor виконує факторизацію чисел.

Робота програми складається з таких етапів:

1) знаходження добутку модулів;
2) факторизація отриманого добутку;
3) перебір всіх можливих комбінацій отриманих чисел;
4) перевірка комбінацій на цілочисельнісь модулів;
5) обчислення модулів для чисел, що відповідають умові.

Клас для пошуку модулів містить такі функції:

– calcAB – функція для обчислення добутку a*b. Повертає число, що буде факторизуватися. Параметр функції: m – масив відомих модулів;

– calcP45 – функція для обчислення невідомих модулів P4 і P5. Параметри функції: ab – один з добутків; m – масив відомих модулів;

– test – функція для перевірки модулів. Ця функція повертає true, якщо модуль – ціле число. Параметри функції: ab – добуток факторизованих чисел; m – масив відомих модулів;

– generateModules – виконує перебір усіх комбінацій множників;

– calculate – головна функція, результатом роботи якої є список модулів.

Функції calcP45, calcAB, test працюють відповідно до формул (2.28)–(2.30). Змінна list містить список множників факторизованого числа. У масиві modules зберігаються відомі модулі P0-P3. Алгоритм роботи цієї програми подано у вигляді блок-схеми на рис. 5.1.

Нижче наведено фрагмент програми для генерування модулів досконалої форми:

```
private static TreeSet<Pair> generateModules(LinkedList<Integer>
list,
int[] modules) {
TreeSet<Pair> data = new TreeSet<>(new
Comparator<Pair>() {
@Override public int compare(Pair p1, Pair p2) {
return p1.a - p2.a + p1.b - p2.b;
}
});
// додаємо до множників одиницю
if (!list.contains(1))
list.add(1);
// перебір можливих комбінацій множників for (int count = 0;
count < list.size(); count++) {
```

```
for (int i = 1; i < list.size(); i++) {
int dobA = 1, dobB = 1;
int j = 0;
// обчислити добутки
for (Integer el : list) {
if (j < i) {
dobA *= el;
}
else {
dobB *= el;
}
j++;
}
```

// перевірити, чи задовольняють числа задану умову, і додати в список

```
if (test(dobA, modules) && test(dobB, modules)) {
int p4 = calcP45(dobA, modules);
int p5 = calcP45(dobB, modules);
data.add(new Pair(p4, p5));
}
}
// перемістити останній елемент на перше місце
int last = list.getLast();
list.removeLast();
list.addFirst(last);
}
return data;
}
```

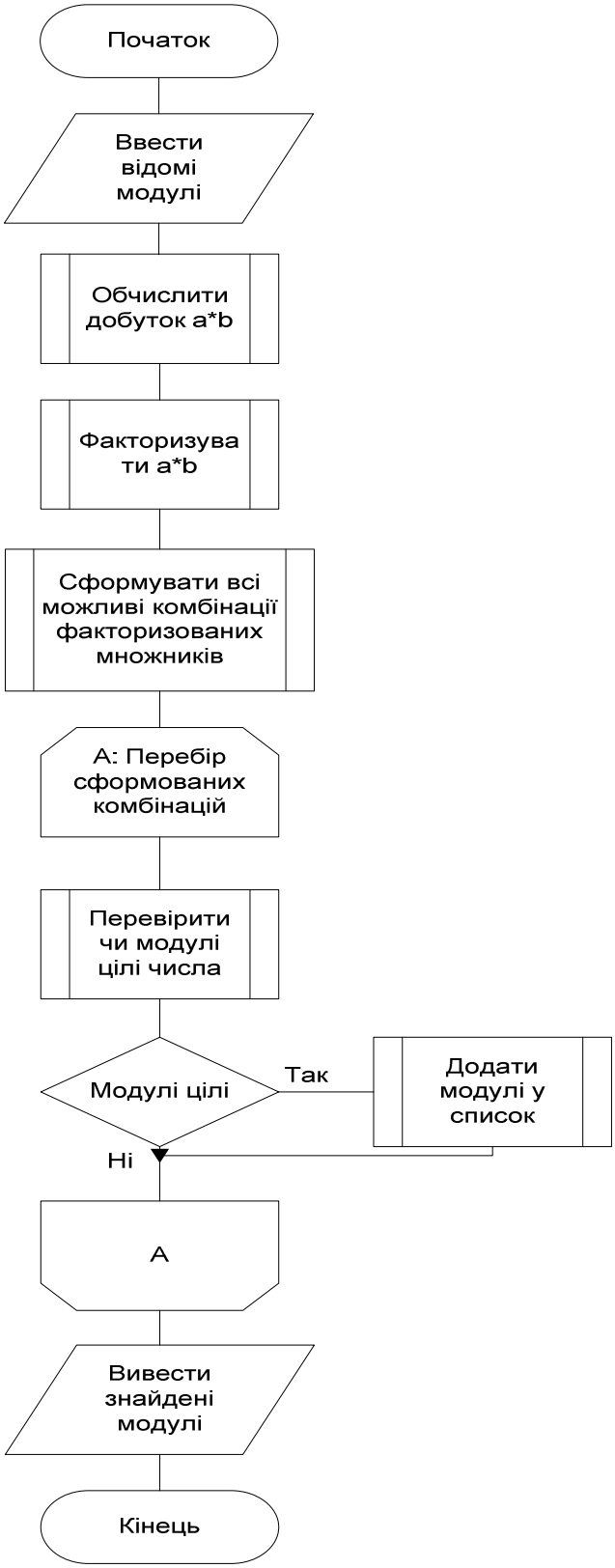


Рис. 5.1. Блок-схема алгоритму програми

На рис. 5.2 зображено головне вікно програми, що призначене для вводу модулів.

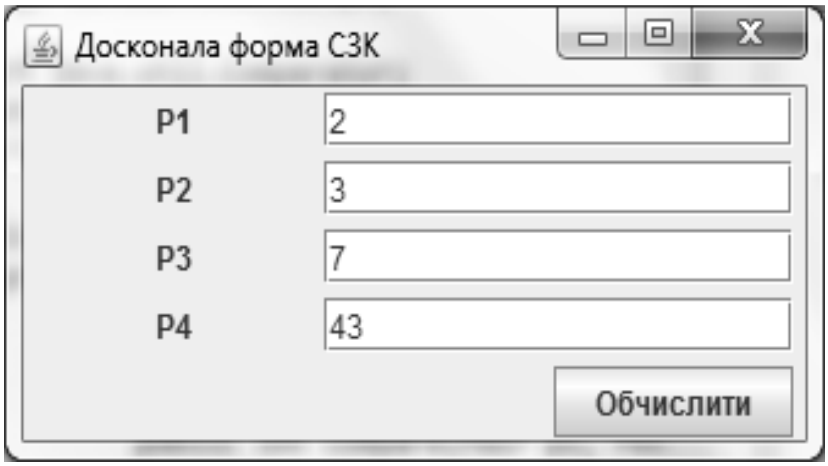


Рис. 5.2. Головне вікно програми

Після натискання на кнопку «Обчислити» програма виводитиме діалогове вікно із знайденими модулями (рис. 5.3), які задовольняють умову ДФ СЗК.

Рис. 5.3. Вивід результатів роботи програми

Для зберігання модулів використовується модуль TreeSet, що є структурою даних для зберігання упорядкованих елементів множини. У цій структурі даних будуть зберігатися об'єкти класу Pair. Щоб уникнути повторюваних елементів, у конструктор передається його власний компаратор. Компаратор знаходить сумарну різницю елементів. Якщо елементи однакові за значенням, то результатом обчислення компаратора буде 0. Необхідно також додати до множників 1, оскільки це число буде частиною результату під час факторизації.

Тепер заповнюється модуль TreeSet можливими комбінаціями множників. При цьому множники у кожному разі мають бути розділені на дві множини A та B, де $B=S/A$, S – це множники факторизованого добутку ab.

Щоб виконати перебір, необхідно в циклі збільшувати кількість елементів однієї з множин і паралельно перелічувати всі можливі варіанти комбінацій множників для заданої кількості елементів у множинах. Для цього на кожній ітерації виконується переставляння останнього множника у множині на перше місце шляхом послідовного виклику методів getLast, removeLast, addFirst.

Для кожної сформованої комбінації обчислюється добуток елементів для кожної множини. Після цього перевіряється цей добуток на його відповідність умові рівності (2.30). Якщо обидва числа задовольняють умову, то обчислюються модулі за формулою (2.28) і зберігаються у список з результатами.

В множині TreeSet зберігаються об'єкти класу Pair. Цей об'єкт містить два цілочисельних поля a і b. Конструктор цього класу написаний так, щоб числа a і b завжди ініціалізувались згідно з умовою $a<b$.

Функція calculate спочатку обчислює добуток ab за допомогою виклику функції calcAB. Далі отриманий добуток

факторизується. Після цього перевіряється кількість елементів отриманої множини множників. Якщо кількість елементів більша від одиниці (число не просте), то викликається функція для генерування модулів generateModules. В іншому разі повертається null.

5.2. Програмна реалізація методу підбору модулів модифікованої досконалої форми системи залишкових класів

Програма також написана мовою Java. Для факторизації чисел можна використати одну зі сторонніх бібліотек, наприклад: GMP-ECM. Функція primeFactor виконує факторизацію чисел.

Робота програми складається з таких етапів: знаходження добутку модулів; факторизація отриманого добутку; перебір всіх можливих комбінацій отриманих чисел; обчислення модулів для чисел, що відповідають умові МДФ СЗК.

При реалізації програми потрібно розв'язати такі задачі:

1) написати ефективний алгоритм перебору всіх можливих добутків *ab*;

2) розробити універсальну архітектуру, яка дасть змогу підтримувати окремі реалізації обчислення невідомих модулів при заданій їхній кількості. Логіку, яка є спільною для всіх реалізацій, необхідно вивести в окремий клас. Для забезпечення універсальності потрібно розробити відповідні інтерфейси;

3) розробити клас, що забезпечить зберігання і сортування модулів, та клас, що буде зберігати значення модулів (Java Bean).

Алгоритм роботи цієї програми поданий у вигляді блок-схеми на рис. 5.4.

Рис. 5.4. Загальна блок-схема алгоритму програми

Відповідно до блок-схеми, обчислення добутку *ab* і невідомих *x*, *y* подані у вигляді наперед визначених процесів. Блок-схема описує загальний алгоритм пошуку для будь-якої кількості модулів. На рис. 5.5 наведені блок-схеми, що описують зазначені процеси обчислення для 4-х модулів.

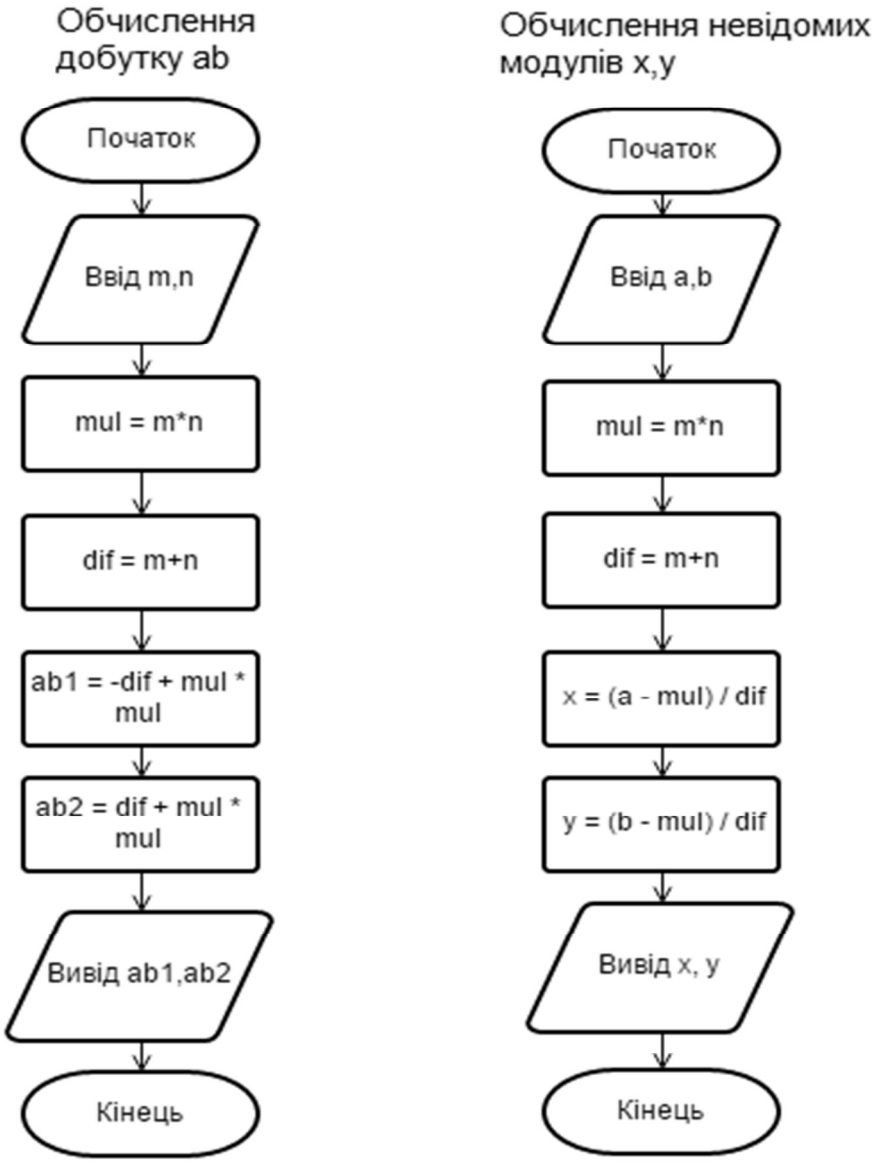


Рис. 5.5. Блок-схеми обчислення *ab* і невідомих *x*, *y* для 4-х модулів

Для зручної заміни реалізації потрібно оголосити такі інтерфейси:

– Analizer містить метод analize для перевірки факторизованого добутку *ab*. Як парамери приймає два цілих числа: *a* і *b* відповідно;

– Permutation є абстракцією класу для генерування всеможливих добутків *ab*, які потрібно факторизувати. Цей

інтерфейс має єдиний метод, що як параметри приймає вектор чисел (факторизований добуток *ab*) і об'єкт, що реалізує інтерфейс Analizer. Метод цього об'єкта буде викликаний для кожного обчисленого добутку.

На основі блок-схеми можна виокремити такі класи:

– HpfModules – Java Bean для представлення об'єктів модулів МДФ СЗК. Цей об'єкт містить поле для зберігання масиву чисел і метод isHalfPerfect. Метод повертає true, якщо модулі, що містяться в об'єкті класу, утворюють МДФ СЗК. Клас також містить статичний метод для перевірки коректності модулів validate, що приймає екземпляр класу HpfModules, і повертає true, якщо модулі не дублюються і не дорівнюють 0;

– ModulesStore – виконує зберігання і сортування знайдених модулів. Найважливішим є метод getSortedModules() – повертає множину Set<HpfModules>;

– PrimeFactor – містить метод calculate(). Цей метод приймає як параметри число, яке буде факторизоване. Результатом роботи методу є масив цілих чисел, на які було розкладено аргумент;

– RecursivePermutation – реалізує інтерфейс Permutation. Виконує перебір можливих комбінацій множників *ab* за допомогою оптимізованого алгоритму рекурсивним шляхом;

– ThreeModulesCalculator, FourModulesCalculator, Five-ModulesCalculator – резалізації інтерфейсу ModulesCalculator. За дані класи містять логіку обчислення для 3-х, 4-х і 5-ти модулів МДФ СЗК відповідно. Ці класи також реалізують інтерфейс Analize. Такий підхід забезпечує універсальність при використанні класу RecursivePermutation із різними реалізація ModulesCalculator. Однак реалізація ModulesCalculator не прив'язана до частини, за яку відповідає клас Analize, тому

метод analize при зростанні складності може бути перенесений в окремий клас;

– Main, ResultDialog – відповідають за графічний інтерфейс налаштування пошуку і відображення результатів.

Нижче наведено фрагмент програми для генерування модулів модифікованої досконалої форми:

```
public void calculate(int n, int m) {
this.m = m;
this.n = n;
dif = n - m;
mul = n * m;

int ab = (-dif + mul * mul);
processAb(ab);
ab = (dif + mul * mul);
processAb(ab);
}

private void processAb(int ab) {
int[] primes = PrimeFactor.calculate(ab);

permutation.permutate(primes, this);
}
@Override
public void analyze(int a, int b) {
simpleAnalize(a, b);
//we should count -1 also
//but multiply should be positive so we have 2 minuses
simpleAnalize(-a, -b);
}
```

```
private void simpleAnalize(int a, int b) {
//maybe here should be a minus before a & b
int x = (a - mul) / dif;
int y = (b - mul) / dif;
Integer[] mods = new Integer[]{n, m, x, y};
if (HpfModules.validate(mods)) {
HpfModules module = new HpfModules(mods);
if (module.isHalfPerfect()) {
store.addModule(module);
}
```

Наведений фрагмент програми ілюструє пошук для 4-х модулів МДФ СЗК. Описані методи належать до класу FourModulesCalculator, який містить логіку обчислення добутку *ab* і невідомих *x, y*. Цей клас містить такі методи:

– calculate – метод, що приймає відомі модулі *m, n*. Цей метод ініціює пошук невідомих *x, y* для заданих *m, n*;

– processAb – призначений для факторизації добутку *a, b* і перебору можливих комбінацій його множників. Як параметри отримує добуток *ab.* Для перебору множинків *ab* використовується допоміжний клас RecursivePermutation, що реалізує інтерфейс Permutation. Інтерфейс Permutation є абстракцією класу, що дає змогу перебирати можливі множники заданого вектора. При цьому передбачається, що всі отримані множники будуть передані екземпляру класу Analizer, який має провести аналіз отриманих даних;

– analyze – фактично є методом, що проводить аналіз всіх отриманих добутків факторизованого *ab*, тобто викликає simpleAnalize для двох можливих випадків [*a*, *b*] і [–*a*, –*b*];

– simpleAnalize - обчислює невідомі *x, y* і перевіряє, чи вектор [*m*, *n*, *x*, *y*] утворює МДФ СЗК.

На рис. 5.6 зображено головне вікно програми, що призначене для налаштування пошуку модулів. Воно містить список вибору кількості модулів і поля для вводу діапазону пошуку для відомих модулів *m*, *n*.

Оскільки заміна дозволяє знайти тільки 2 невідомих модулі, то кількість діапазонів їхнього пошуку, які потрібно задати, буде змінюватися залежно від обраної кількості модулів. Для трьох модулів потрібно задати один діапазон, для чотирьох – два і т. д.

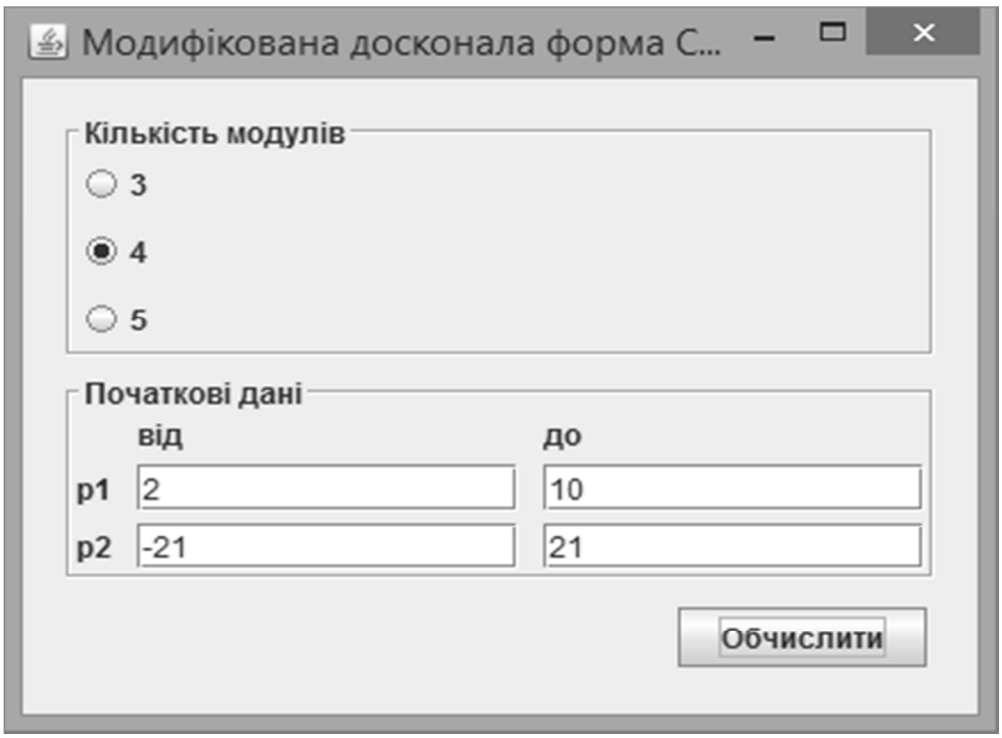


Рис. 5.6. Головне вікно програми

Після натискання на кнопку «Обчислити» програма виводить діалогове вікно із знайденими модулями (рис. 5.7). Далі програма розпочне проходження заданого діапазону чисел і пошук модулів для кожного випадку заданого p_i. Якщо діапазон великий, то обчислення будуть виконуватися значний проміжок

часу. Щоб графічний інтерфейс при цьому відповідав на запити користувача, необхідно всі обчислення вивести в окремий потік:

```
calculateBtn.addActionListener(event -> {
Thread calculateThread = new Thread(() -> {
//обчислення модулів
}).start();
```

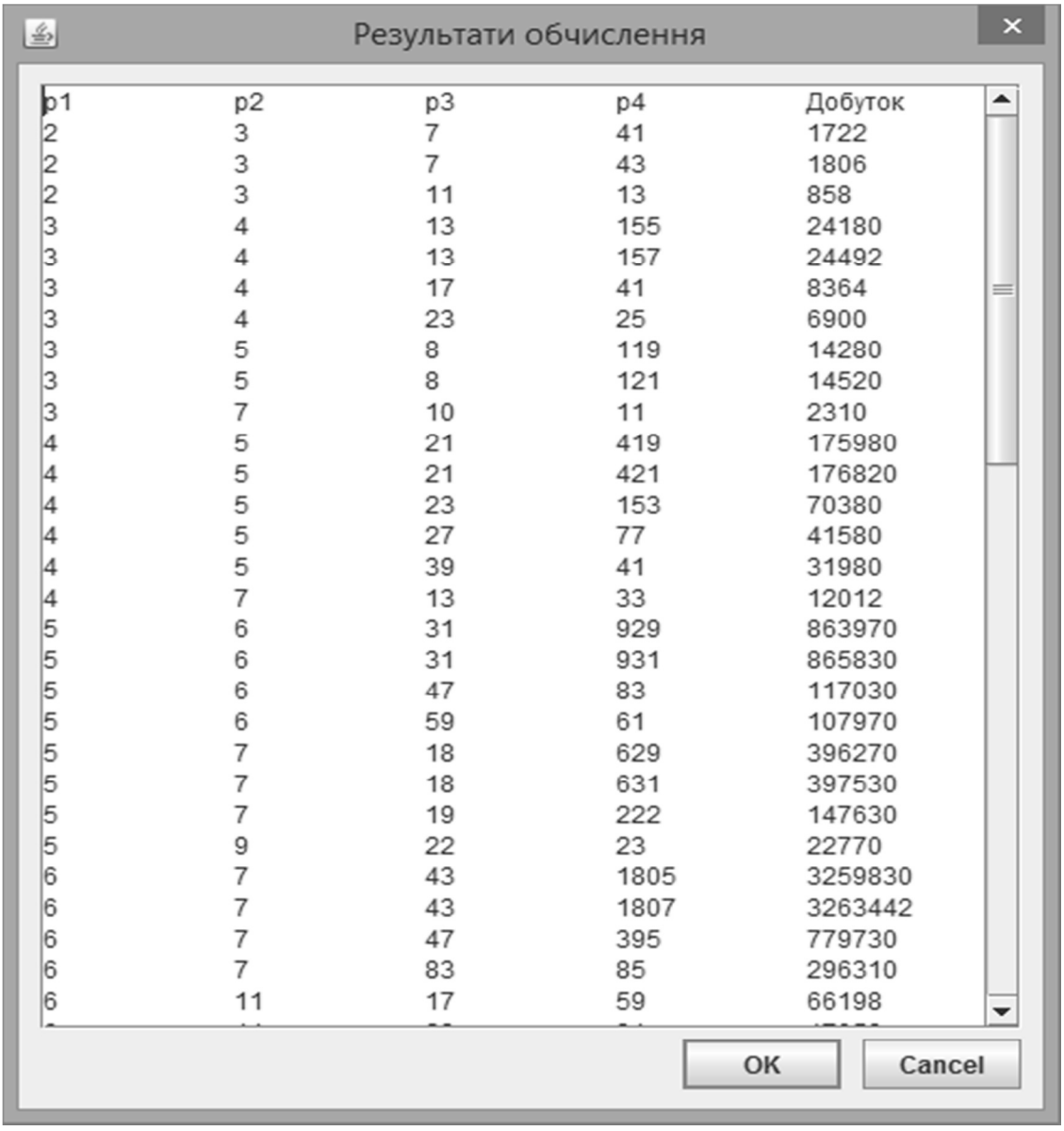


Рис. 5.7. Вивід результатів роботи програми

Для зберігання модулів використовується модуль TreeSet, що є структурою даних для зберігання упорядкованих елементів множини. У цій структурі даних будуть зберігатися об'єкти класу Pair. Щоб уникнути повторюваних елементів, у конструктор передається його власний компаратор. Компаратор знаходить сумарну різницю елементів. Якщо елементи однакові за значенням, результатом обчислення компаратора буде 0. Необхідно також додати до множників 1, оскільки 1 буде частиною результату під час факторизації.

Щоб виконати перебір можливих множників факторизованого *ab*, необхідно в циклі збільшувати кількість елементів однієї із множин і паралельно перелічувати всі можливі варіанти комбінацій множників для заданої кількості елементів у множинах. Для цього на кожній ітерації виконується переставляння останнього множника у множині на перше місце шляхом послідовного виклику методів getLast, removeLast, addFirst.

Для кожної сформованої комбінації обчислюється добуток елементів для кожної множини. Після цього перевіряється цей добуток на відповідність умові рівності (4.8). Якщо два числа задовольняють умову, то обчислюються модулі за формулою (4.5) і зберігаються у список з результатами.

5.3. Побудова трьохмодульної криптосистеми Рабіна на основі різних форм системи залишкових класів

Для шифрування інформаційних потоків з використання трьохмодульної криптосистеми Рабіна вибирається три великих простих числа *p, q і r* [224–225]. Обчислюється значення $n=p\cdot q\cdot r$, де число *n* – відкритий ключ, а *p*, *q* і *r* – закритий.

Шифрування повідомлення *A* (текст) відбувається за допомогою відкритого ключа *n* за формулою $A'=A^2 \bmod n$.

При дешифруванні криптограми *A'* вводяться додаткові допоміжні величини k, l і d:

$$k = A' \bmod p;\ l = A' \bmod q;\ d = A' \bmod r. \qquad (5.1)$$

Значення x, y і z шукають з порівнянь:

$$x^2 \equiv k (\bmod p), \qquad (5.2)$$

$$y^2 \equiv l (\bmod q), \qquad (5.3)$$

$$z^2 \equiv d (\bmod r). \qquad (5.4)$$

Для знаходження *x*, *y* і z необхідно обчислити значення кореня квадратного за модулем. Класичні підходи з використання символів Якобі або Лежандра є працемісткими [226]. З огляду на це пропонується метод, який ґрунтується тільки на операції додавання та потребує перевірки, чи є число повним квадратом, що суттєво зменшує обчислювальну складову методу Рабіна. Отже, для того, щоб знайти значення $\sqrt{k} \bmod p$, необхідно виконати дії у такій послідовності: $k + p$, $k + 2p$, …, $k + i \cdot p$, де i – значення, при якому $k + i \cdot p$ буде повним квадратом. Аналогічно обчислюємо $y^2 \equiv l (\bmod q)$, $z^2 \equiv d (\bmod r)$. Оскільки розв'язками порівнянь (5.2)–(5.4) буде 6 значень, то для дешифрування потрібно розв'язати вісім систем порівнянь, що утворюються як комбінації можливих варіантів пошуку відкритого повідомлення:

$$\begin{cases} A_1 \equiv x(\bmod p); \\ A_1 \equiv y(\bmod q); \\ A_1 \equiv z(\bmod r); \end{cases} \begin{cases} A_2 \equiv -x(\bmod p); \\ A_2 \equiv y(\bmod q); \\ A_2 \equiv z(\bmod r); \end{cases} \begin{cases} A_3 \equiv x(\bmod p); \\ A_3 \equiv -y(\bmod q); \\ A_3 \equiv z(\bmod r); \end{cases} \begin{cases} A_4 \equiv x(\bmod p); \\ A_4 \equiv y(\bmod q); \\ A_4 \equiv -z(\bmod r); \end{cases}$$

$$\begin{cases} A_5 \equiv -x(\bmod p); \\ A_5 \equiv -y(\bmod q); \\ A_5 \equiv z(\bmod r); \end{cases} \begin{cases} A_6 \equiv -x(\bmod p); \\ A_6 \equiv y(\bmod q); \\ A_6 \equiv -z(\bmod r); \end{cases} \begin{cases} A_7 \equiv x(\bmod p); \\ A_7 \equiv -y(\bmod q); \\ A_7 \equiv -z(\bmod r); \end{cases} \begin{cases} A_8 \equiv -x(\bmod p); \\ A_8 \equiv -y(\bmod q); \\ A_8 \equiv -z(\bmod r). \end{cases} \quad (5.5)$$

Одне з розв'язків систем порівнянь (5.5) буде шуканим повідомленням *A*.

Для розв'язання задачі і знаходження всіх розв'язків системи (5.5) потрібно для кожної системи застосувати КТЗ, тобто для першої системи шукане число *A* представлене у вигляді набору $(x, y, z)_{p, q, r}$ найменших невід'ємних залишків від ділення цього числа на фіксовані натуральні попарно взаємно прості числа *p, q, r* ($x = A_1 \bmod p$, $y = A_1 \bmod q$, $z = A_1 \bmod r$), які є відкритими ключами. При цьому має виконуватись умова $0 \le A_1 < n-1$. Знаходження A_1 згідно з КТЗ є доволі складним та громіздким процесом:

$$A_1 = (x \cdot B_1 + y \cdot B_2 + z \cdot B_3) \bmod n, \quad (5.6)$$

де $B_i = S_i m_i$, $i = 1, 2, 3$, $S_1 = \frac{n}{p}, S_2 = \frac{n}{q}, S_3 = \frac{n}{r}$, m_i шукається з виразів $(S_1 m_1) \bmod p = 1$, $(S_2 m_2) \bmod q = 1$, $(S_3 m_3) \bmod r = 1$.

Наведемо приклад. Нехай $p=11$, $q=13$, $r=17$, $n=2431$, відкритий текст $A=1031$. Зашифроване повідомлення: $A' = 1031^2 \bmod 2431 = 614$. Розшифровування буде здійснюватися в такій послідовності: $k = 614 \bmod 11 = 9$, $l = 614 \bmod 13 = 3$,

$d = 614 \bmod 17 = 2$. Далі необхідно обчислити квадратичні лишки $x^2 \equiv 9(\bmod 11)$, $y^2 \equiv 3(\bmod 13)$, $z^2 \equiv 2(\bmod 17)$. Для цього прикладу $\sqrt{9} \bmod 11 = 3$ і 11–3=8, $\sqrt{3+13} \bmod 13 = 4$ та 9, $\sqrt{2+2\cdot 17} \bmod 17 = 6$ та 11. З отриманих коренів можна утворити 8 різних комбінацій: (3, 4, 6), (3, 4, 11), (3, 9, 6), (3, 9, 11), (8, 4, 6), (8, 4, 11), (8, 9, 6), (8, 9, 11).

Для застосування КТЗ потрібно знайти значення m_i. За вибраним прикладом на основі методу додавання модуля можна знайти, що S_1=13·17=221, $(S_1 m_1) \bmod p = 221 \cdot m_1 \bmod 11 = 1 \cdot m_1 \bmod 11 = 1$, звідси m_1=1. Аналогічно S_2=11·17=187, $(S_2 m_2) \bmod q = 187 \cdot m_2 \bmod 13 = 5 \cdot m_2 \bmod 13 = 1$, тоді $m_2 = \frac{(1+3\cdot 13)}{5} = 8$ і S_3=11·13=143,

$(S_3 m_3) \bmod r = 143 \cdot m_3 \bmod 17 = 7 \cdot m_3 \bmod 17 = 1$, $m_3 = \frac{(1+2\cdot 17)}{7} = 5$.

Згідно із системою (5.5) отримано такі результати:

1) A_1 = (3·1·221+4·8·187+6·5·143) mod2431 =
=(663+5984+4290) mod2431= 10937 mod2431 = 1213;

2) A_2=(3·1·221+4·8·187+11·5·143) mod2431=
=(663+5984+7865) mod2431= 14512 mod2431 = 2357;

3) A_3=(3·1·221+9·8·187+6·5·143) mod2431=
=(663+13464+4290) mod2431= 18417 mod2431 = 1400;

4) A_4=(3·1·221+9·8·187+11·5·143) mod2431=
=(663+13464+7865) mod2431= 21992 mod2431 = 113;

5) A_5=(8·1·221+4·8·187+6·5·143) mod2431=
= (1768+5984+4290) mod2431= 12042 mod2431 = 2318;

6) A_6=(8·1·221+4·8·187+11·5·143) mod2431=
=(1768+5984+7865) mod2431= 15617 mod2431 = 1031;

7) A_7=(8·1·221+9·8·187+6·5·143) mod2431=
=(1768+13464+4290) mod2431= 19522 mod2431 = 74;

8) A_8=(8·1·221+9·8·187+11·5·143) mod2431=
=(1768+13464+7865) mod2431=23097 mod2431=1218.

Звідси можна визначити, що зашифрованому повідомленню відповідає A_6.

Задача суттєво спрощується, коли модулі p, q, r підібрано таким чином, що вони утворюють МДФ СЗК, тобто m_i=±1. Це дозволяє уникнути виконання громіздкої процедури пошуку оберненого елемента за модулем та множення в рівності (5.6) на m_i. Нехай p=11, q=17, r=31, n= 5797. Тоді A'=1031^2mod 5797=2110, $k = 2110 \bmod 11 = 9$, $l = 2110 \bmod 17 = 2$, $d = 2110 \bmod 31 = 2$; $\sqrt{9} \bmod 11 = 3$ і 8, $\sqrt{2 + 2 \cdot 17} \bmod 17 = 6$ та 11, $\sqrt{2 + 2 \cdot 31} \bmod 31 = 8$ і 23. З отриманих коренів утворюється 8 різних комбінацій: (3, 6, 8), (3, 6, 23), (3, 11, 8), (3, 11, 23), (8, 6, 8), (8, 6, 23), (8, 11, 6), (8, 11, 23).

Далі S_1=17·31=527, $(S_1 m_1) \bmod p = 527 \cdot m_1 \bmod 11 = 10 \cdot m_1 \bmod 11 = 1$, звідси m_1=10. Аналогічно S_2=11·31=341, $(S_2 m_2) \bmod q = 341 \cdot m_2 \bmod 17 = 1 \cdot m_2 \bmod 13 = 1$, m_2=1; і S_3=11·17=187, $(S_3 m_3) \bmod r = 187 \cdot m_3 \bmod 31 = 1 \cdot m_3 \bmod 17 = 1$, m_3=1.

Тоді згідно із системою (5.5) можна отримати:

1) A_1=(3·10·527+6·1·341+8·1·187) mod5797=
=(15810+2046+1496) mod5797= 19325 mod5797 = 1961;

2) A_2=(3·10·527+6·1·341+23·1·187) mod5797=
=(15810+2046+4301) mod5797= 22157 mod5797 = 4766;

3) A_3=(3·10·527+11·1·341+8·1·187) mod5797=
=(15810+3751+1496) mod5797= 21057 mod5797 = 3666;

4) A_4=(3·10·527+11·1·341+23·1·187) mod5797=
=(15810+3751+4301) mod5797= 22157 mod5797 = 674;

5) A_5=(8·10·527+6·1·341+8·1·187) mod5797=
=(42160+2046+1496) mod5797= 45702 mod5797 = 5123;

6) A_6=(8·10·527+6·1·341+23·1·187) mod5797=
=(42160+2046+4301) mod5797= 48507 mod5797 = 2131;

7) A_7=(8·10·527+11·1·341+8·1·187) mod5797=
=(42160+3751+1496) mod5797= 47407 mod5797 = 1031;

8) A_8=(8·10·527+11·1·341+23·1·187) mod5797=
=(42160+3751+4301) mod5797= 22157 mod5797 = 3836.

Зашифрованому повідомленню відповідає значення A_7. Оскільки відкритий ключ більший, ніж у попередньому разі, то відповідно і проміжні результати набувають більших значень. Однак якщо врахувати, що m_1=10 mod11=-1 mod11, то отримаємо такі розрахунки:

1) A_1=(–3·1·527+6·1·341+8·1·187) mod5797=
=(–1581+2046+1496) mod5797= 1961 mod5797 = 1961;

2) A_2=(–3·1·527+6·1·341+23·1·187) mod5797=
=(–1581+2046+4301) mod5797= 4766 mod5797 = 4766;

3) A_3=(–3·1·527+11·1·341+8·1·187) mod5797=
=(–1581+3751+1496) mod5797= 3666 mod5797 = 3666;

4) A_4=(–3·1·527+11·1·341+23·1·187) mod5797=
=(–1581+3751+4301) mod5797= 6471 mod5797 = 674;

5) A_5=(–8·1·527+6·1·341+8·1·187) mod5797=
=(–4216+2046+1496) mod5797= –674 mod5797 = 5123;

6) A_6=(–8·1·527+6·1·341+23·1·187) mod 5797=
=(–4216+2046+4301) mod5797= 2131 mod5797 = 2131;

7) A_7=(–8·1·527+11·1·341+8·1·187) mod5797=
=(–4216+3751+1496) mod5797= 1031 mod5797 = 1031;

8) A_8=(–8·1·527+11·1·341+23·1·187) mod5797=
=(–4216+3751+4301) mod5797= 3836 mod5797 = 3836.

Отже, проміжні результати набувають менших значень, ніж в попередньому прикладі. Крім того, тільки у двох випадках з восьми потрібно додавати або віднімати *n*, щоб отримати невід'ємне число, менше за *n*.

5.4. HDL-модель трьохмодульної криптосистеми Рабіна та дослідження її харакеристик

Для дослідження часових характеристик модифікованої криптосистеми Рабіна та системи, в якій модулі утворюють МДФ СЗК, було обрано Active-HDL – середовище розробки,

моделювання і верифікації проектів для програмованих логічних інтегральних схем чи програмованих логічних матриць.

Пропонований програмний засіб складається з трьох основних блоків, які загалом формують криптопроцесор, що передбачає роботу із програмованою логічною інтегральною схемою.

Функціональні модулі у запропонованій програмі такі:

– модуль під'єднання необхідних бібліотек;

– модуль оголошення внутрішніх та зовнішніх портів (інтерфейсу проекту);

– модуль опису поведінки розробленої моделі.

Оголошення бібліотек охоплює декілька основних компонент, до яких програма буде звертатись у подальшому. В цьому разі опис доступу до необхідних бібліотек у середовищі Active-HDL має такий вигляд:

– LIBRARY IEEE;

– IEEE.STD_LOGIC_1164.ALL;

– IEEE.NUMERIC_STD.ALL;

– IEEE.MATH_REAL.ALL.

Бібліотека LIBRARY IEEE – це бібліотека вищого рівня, розроблена всесвітньою організацією IEEE, яка постійно оновлюється. За її допомогою виконується доступ і використання бібліотек нижнього рівня.

Бібліотека IEEE.STD_LOGIC_1164.ALL – це стандартизований пакет даних, який використовується при моделюванні й описі бітових типів даних. Логічна система цього пакета передбачає в коді програми роботу над логічними елементами. Проте виконання такого примітивного завдання, як моделювання роботи вентильного транзистора, виходить за межі функціоналу цієї бібліотеки.

Бібліотека IEEE.NUMERIC_STD.ALL – пакет, що передбачає оголошення цифрових і деяких математичних функцій, які використовуються у проектах, що будуть синтезуватись. У цьому пакеті оголошується принцип роботи програмного продукту із числовими масивами і виконання з ними елементарних операцій (+,-,*,/ зсуву ліворуч і праворуч).

Бібліотека IEEE.MATH_REAL.ALL використовується як доповнення до пакета IEEE.NUMERIC_STD.ALL, щоб можна було виконувати із числами більш складні математичні операції – піднесення до степеня, добування коренів різного степеня, знаходження модулів і т. ін.

Модуль оголошення внутрішніх і зовнішніх портів є виділеним блоком, який приймає значення і керує всім програмним продуктом, адже саме в цьому блоці описується, звідки будуть зчитуватись дані та куди їх у подальшому буде записано після обробки. Цей розділ складається із двох блоків:

– generic – записують усі константні значення:

generic (

p_1 : integer := 11;

p_2 : integer := 13;

p_3 : integer := 17

);

– port – описують вхідні та вихідні порти програми:

port (

clk : in std_logic;

$v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8$: out integer

);

У блоці generic описано три основні параметри p_1, p_2, p_3, які відповідають за вхідні числові змінні, що будуть використовуватись як закритий ключ при шифруванні.

У блоці port оголошено два основних елементи – це вхід та вихід у програмі. За вхід використовується тактовий сигнал clk, а вихідними сигналами є значення $v_1,\ldots,v_8$. Блок опису коду програми організований таким чином, що на його початку описано всі внутрішні змінні, які будуть використовуватись у програмному продукті. Ці змінні оголошуються за допомогою зарезервованого слова signal і описуються як дійсні значення у відповідних діапазонах:

```
signal n: real range 1.0 to 9999999.0 :=1.0;
signal w: real range 0.0 to 9999999999999.0 :=1.0;
signal word: real range 1.0 to 9999999.0;
signal g111: real range 0.0 to 5000000.0 :=1.0;
signal g222: real range 0.0 to 5000000.0 :=1.0;
signal g333: real range 0.0 to 5000000.0 :=1.0;
signal sqrt1: real range 0.0 to 5000000.0 :=1.0;
signal sqrt2: real range 0.0 to 5000000.0 :=1.0;
signal sqrt3: real range 0.0 to 5000000.0 :=1.0;
signal g1: real range 0.0 to 5000000.0 :=1.0;
signal g2: real range 0.0 to 5000000.0 :=1.0;
signal g3: real range 0.0 to 5000000.0 :=1.0;
signal p1_g1: real range 0.0 to 5000000.0 :=1.0;
signal p2_g2: real range 0.0 to 5000000.0 :=1.0;
signal p3_g3: real range 0.0 to 5000000.0 :=1.0;
signal k1: real range 0.0 to 5000000.0 :=1.0;
signal k2: real range 0.0 to 5000000.0 :=1.0;
signal k3: real range 0.0 to 5000000.0 :=1.0;
signal m1: real range 0.0 to 5000000.0 :=1.0;
signal m2: real range 0.0 to 5000000.0 :=1.0;
signal m3: real range 0.0 to 5000000.0 :=1.0;
signal m11: real range 0.0 to 5000000.0 :=1.0;
signal m22: real range 0.0 to 5000000.0 :=1.0;
signal m33: real range 0.0 to 5000000.0 :=1.0.
```

Здебільшого всі ці змінні застосовуються для проміжних обчислень. Змінна word позначає блок інформації, який буде шифруватися.

Поведінку модифікованого алгоритму Рабіна описано як процес від clk:

```
read: process(clk)
begin
if clk'event and clk='1' then
n<=p1*p2*p3;
w<=word**2.0 mod n ;
g111<=w mod p1;
g222<=w mod p2;
g333<=w mod p3;
sqrt1<=g111;
sqrt2<=g222;
sqrt3<=g333;
end process.
```

У процесі обчислюються відповідно до запропонованого алгоритму змінні $v_1,\ldots,v_8$. Функціональна схема розробленого пристрою має вигляд, представлений на рис. 5.8.

На рис. 5.9 подана блок-схема HDL-моделі трьох-модульного криптоалгоритму Рабіна.

Для побудови часової діаграми симуляції роботи трьохмодульного криптоалгоритму Рабіна без використання МДФ СЗК був вибраний такий ряд вхідних параметрів: p=11, q=13, r=17 – вхідні таємні ключі; Word = 1024 – блок для шифрування.

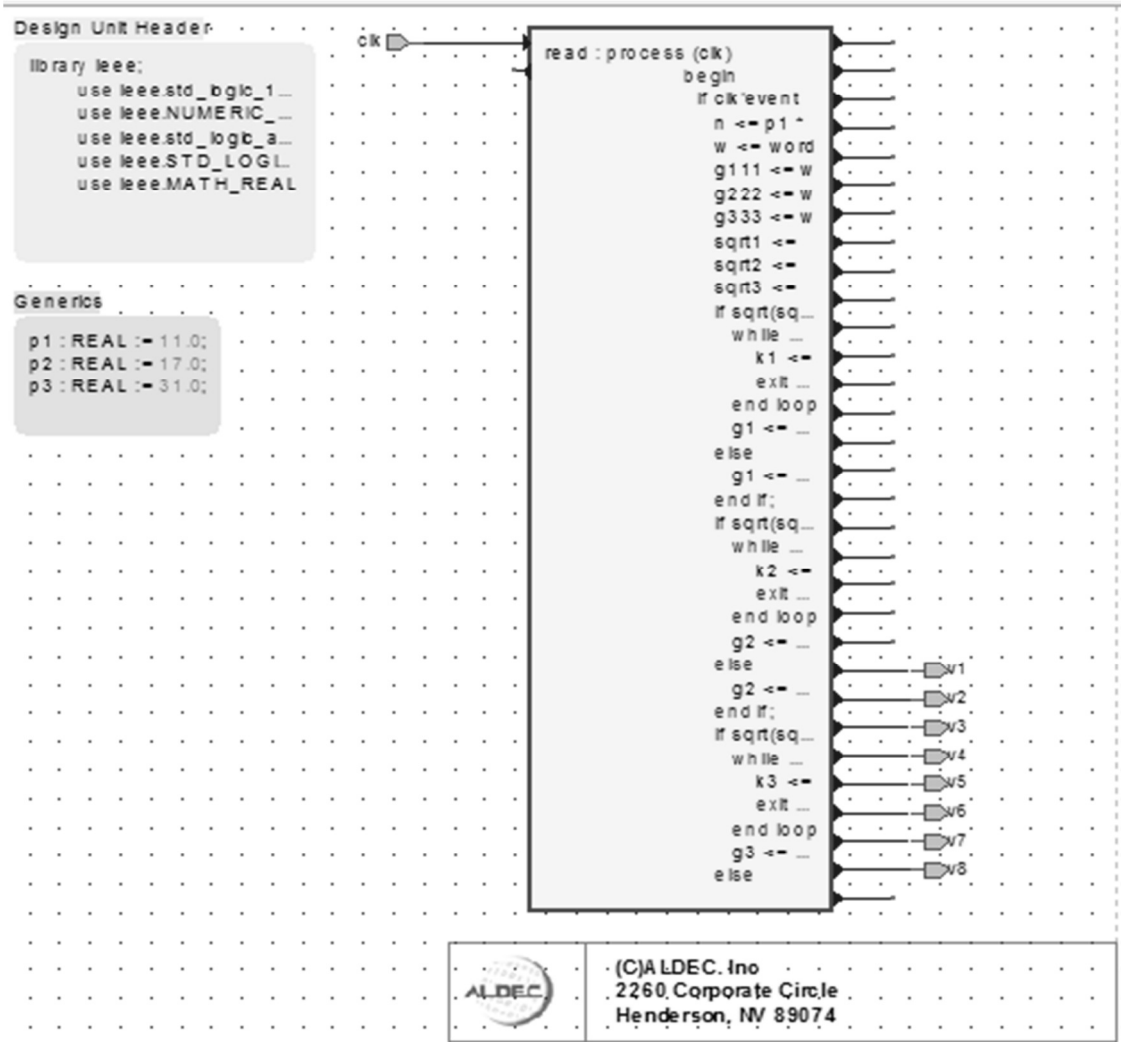


Рис. 5.8. Функціональна схема розробленого пристрою

На часовій діаграмі (рис. 5.10) відображено результати моделювання роботи проектованого пристрою. На основі її аналізу можна визначити, що при проходженні 1,4 мікросекунди (1400 нс) і встановленні всіх проміжних параметрів у статичному стані система виводить результат моделювання на один із вихідних портів.

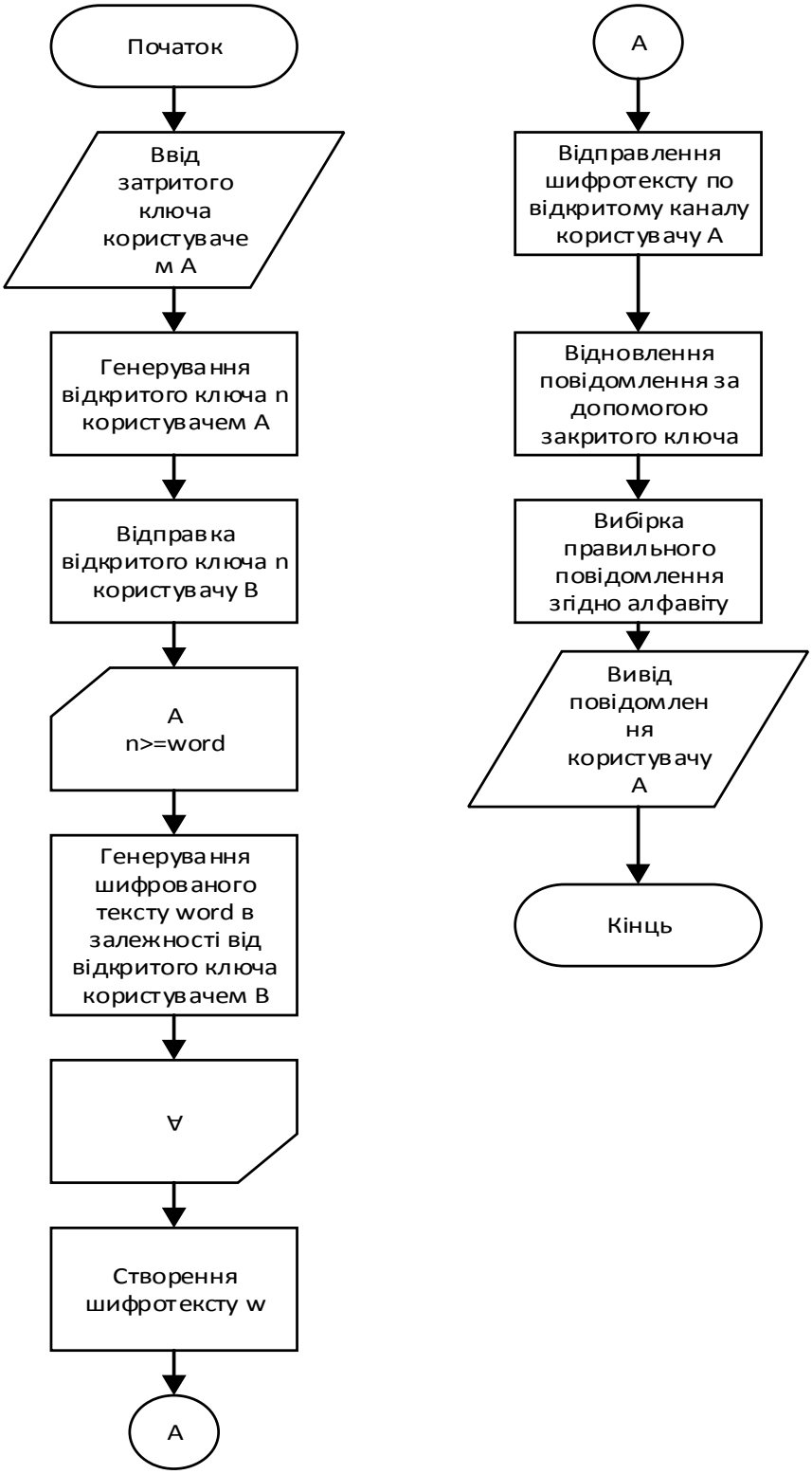


Рис. 5.9. Блок-схема HDL-моделі модифікованого криптоалгоритму Рабіна

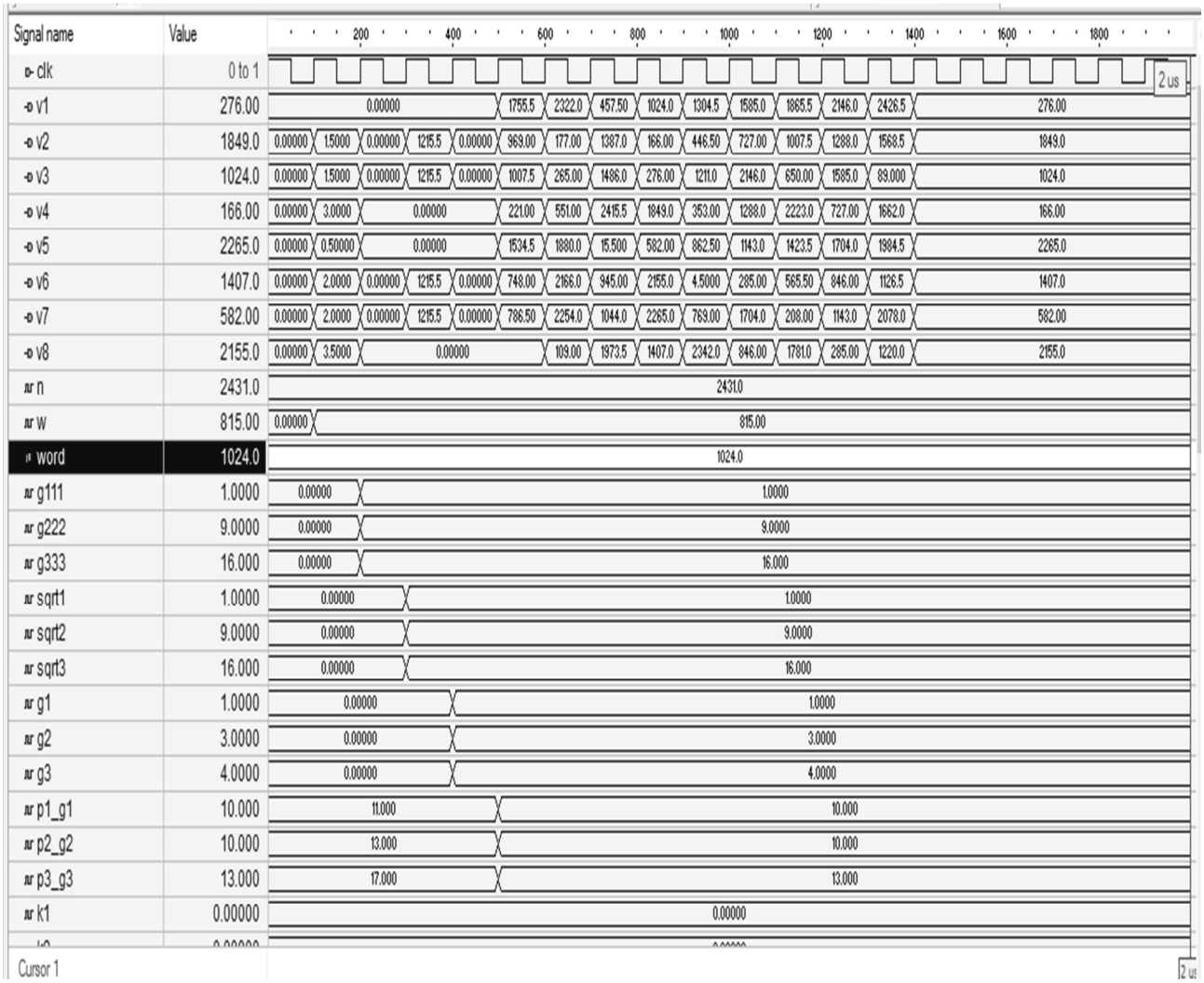


Рис. 5.10. Часова діаграма симуляції роботи трьохмодульного алгоритму Рабіна

Часова діаграма під час симуляції процесу шифрування трьохмодульного криптоалгоритму Рабіна з використанням МДФ СЗК подана на рис. 5.11. Шифрувався такий самий блок відкритого тексту Word = 1024. Були вибрані прості модулі p=11, q=17, r=31, які утворюють МДФ СЗК. Їхній добуток P=5797 більший, ніж у попередньому разі (P=2431), що в загальному має забезпечити збільшення часу роботи алгоритму.

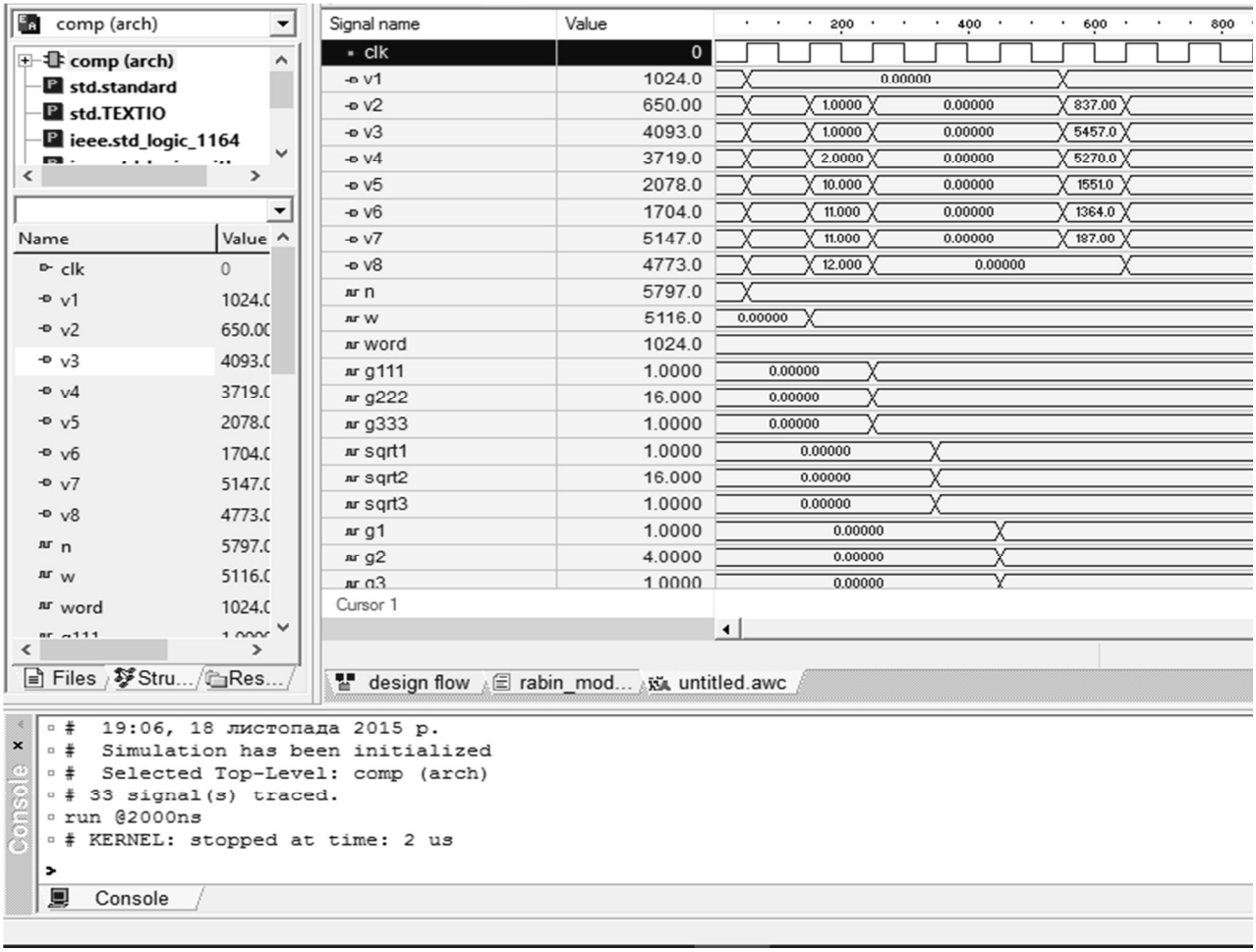


Рис. 5.11. Часова діаграма трьохмодульного криптоалгоритму Рабіна з використанням МДФ СЗК

Однак з рис. 5.11 можна з'ясувати, що час роботи трьохмодульного криптоалгоритму Рабіна з використанням МДФ СЗК, який дає змогу уникнути виконання процедури пошуку оберненого елемента за модулем, становить 650 нс, тобто зменшився приблизно вдвічі порівняно з попереднім.

Крім того, цей метод при виборі модулів одного порядку має перевагу перед класичним у стійкості за рахунок збільшення блоку відкритого тексту для шифрування.

У табл. 5.1 (результати розміщені в порядку зростання діапазону обчислень P) наведено час (у наносекундах) t_{1i} та t_{2i} (i=1, 2) виконання трьохмодульного криптоалгоритму Рабіна на

основі звичайної цілочисельної СЗК та для деяких значень простих модулів, які утворюють МДФ СЗК. Блоки відкритого тексту вибиралися таким чином: Word1=131071, вага Хемінга якого є максимальною для заданої розрядності (131071=2^{17}–1), та Word2=229376, у двійковому записі якого в трьох старших розрядах є одиниці, а в усіх решти розрядах – нулі.

Таблиця 5.1

Час виконання трьохмодульного криптоалгоритму Рабіна

№	*p*	*q*	*r*	*P*	Word=131071		Word=229376	
					t_{11}, нс	t_{12}, нс	t_{21}, нс	t_{22}, нс
1	41	71	97	282367	4100	1000	4100	1300
2	43	71	109	332777	4200	3000	4200	1900
3	41	53	181	393313	4100	900	4100	1800
4	29	31	449	403651	2900	800	2900	1300
5	37	41	379	574943	3700	1500	3700	2200
6	53	89	131	617927	5300	3200	5300	1000
7	59	79	233	1086013	5900	5500	5900	5100
8	67	101	199	1346633	6700	2900	6700	3000
9	59	71	349	1461961	7300	7200	5900	2400
10	41	43	881	1553203	5000	4900	11900	11800
11	67	89	271	1615973	6700	3600	6700	4900
12	53	59	521	1629167	9900	9800	5300	4000
13	71	101	239	1713869	7100	4600	7100	2200
14	61	71	433	1875323	6100	4200	6100	3500
15	79	131	199	2059451	7900	3200	7900	3000
16	97	109	881	9314813	9700	4900	11900	11800

Час для різних чисел позначений відповідно t_{j1} та t_{j2} (j=1, 2). Розшифровування у звичайній цілочисельній СЗК (i=1) відбувається за формулою (5.6), обернені елементи (m_1=p–1, m_2=m_3=1) шукаються на основі додавання модуля. При використанні МДФ СЗК розшифровування здійснено за формулою:

$$M_1 = \left(-x \cdot S_1 + y \cdot S_2 + z \cdot S_3\right) \bmod n . \qquad (5.7)$$

За даними табл. 5.1 можна визначити, що для модулів, які утворюють МДФ СЗК, використання виразу (5.7) зменшує час криптоперетворень. При шифруванні Word1 для модулів p=41, q=53, r=181 досягається максимальна перевага приблизно в 4,6 разу.

Водночас існують модулі криптоперетворень p=53, q=59, r=521, для яких спостерігається мінімальна перевага, тобто прискорення досягається 1,01 разу. Це пояснюється різною кількістю ітерацій при пошуку оберненого елемента.

У криптоперетворенні блоку Word2 максимальна та мінімальна переваги у 5,3 та 1,008 разу відповідно спостерігаються при використанні наборів модулів p=53, q=89, r=131 та p=97, q=109, r=881.

В середньому час виконання операцій при використанні МДФ СЗК для Word1 та Word2 зменшився відповідно у 1,58 і 1,63 разу.

На рис. 5.12 зображено графіки часу виконання криптоперетворень залежно від номера наборів модулів згідно з даними табл. 5.1.

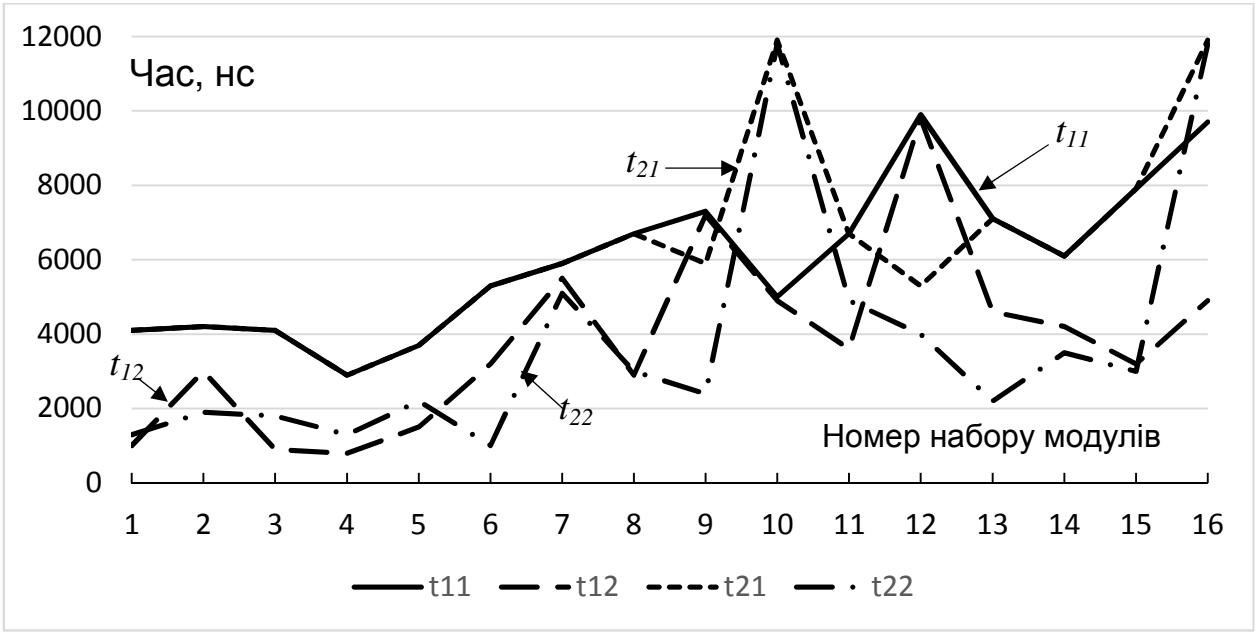


Рис. 5.12. Графіки залежності часу шифрування від номера наборів модулів

За рис. 5.12 можна зробити висновок, що всі графіки мають осцилюючий характер, однак загальний тренд спрямований на збільшення часу виконання криптоперетворень при зростанні можливого діапазону виконання операцій. Запропоновану HDL-модель трьохмодульного криптоалгоритму Рабіна з використанням МДФ СЗК можна успішно реалізувати на сучасних програмованих логічних інтегральних схемах або програмованих логічних матрицях.

5.5 Метод побудови розподіленого термо- або п'єзоелектричного сенсора на основі системи залишкових класів

З курсу фізики відомо, що загальний опір послідовно з'єднаних резисторів дорівнює сумі опорів кожного з них [227]. Зворотна операція за класичного підходу є практично нездійсненою. Розв'язання такої задачі – це важливий крок для розробки засобів автоматизованого управління спеціалізованими комп'ютерними системами, технологічні об'єкти яких мають специфічні особливості, що усувають можливість безпосереднього доступу людини. Однією із задач є також визначення тиску та розподілених температурних полів у різних точках і на різних рівнях досліджуваного середовища (наприклад, для контролю умов зберігання й обліку руху нафто- та газопродуктів). Такі задачі актуальні в геофізиці, нафто-газовидобувній, вугільній, металургійній, космічній та інших галузях промисловості, а також у метеорології.

Важливим аспектом розвитку досліджень у цій галузі є розробка підходів, методів, алгоритмів та комп'ютерних засобів побудови розподілених сенсорів на основі використання СЗК.

Сенсори для вимірювання розподілених значень тиску або температури, як правило, реалізуються двома способами: або багатоточковою структурою з паралельним інформаційним каналом, або на основі позиційних систем числення з моноканальною лінією передачі інформації [228]. Недоліком першого способу є наявність великої кількості автономних ліній зв'язку від сенсорів до мікроконтролерів.

Основою другого методу побудови багатопараметричного сенсора (БПС) є послідовне з'єднання термо- або п'єзорезисторів R_i. Опір кожного з них може змінюватися стрибкоподібно на величину ΔR_i, що визначає точність вимірювання і відповідає основі системи числення. Тоді загальний опір БПС визначається аналітичним виразом $R_x = \sum_{i=1}^{n} R_i$. Недоліком цього методу вимірювання є значна різниця між ΔR_i, яка відповідає $\frac{\Delta R_{i+1}}{\Delta R_i} = F$, де F – основа позиційної системи числення. Наприклад, при F=2 та F=10 відповідно ΔR_i в кожному каналі має змінюватися в 2 та 10 разів. Крім того, кожний наступний сенсор має в F разів більший діапазон вимірювання порівняно з попереднім.

Нехай маємо t однакових послідовно з'єднаних термо- або п'єзорезисторів R_0 (рис. 5.13), опір яких може змінюватися стрибкоподібно з кроком $\Delta R_i = \frac{R_0}{p_i}$, де p_i – взаємнопрості числа або модулі, які визначають точність вимірювання [229–230]. При цьому максимальне значення вимірюваної величини для кожного резистора становить $D_i = R_0\left(1 - \frac{1}{p_i}\right)$, а, відповідно, повне максимальне значення:

$$D = \sum_{i=1}^{t} D_i = R_0 \cdot \sum_{i=1}^{t} \left(1 - \frac{1}{p_i} \right). \qquad (5.8)$$

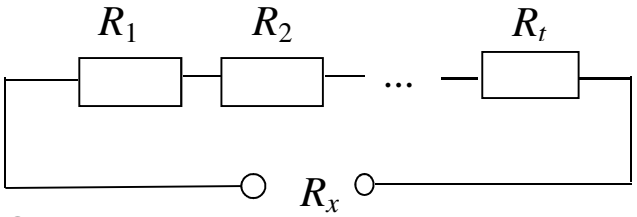


Рис. 5.13. Схема вимірювальної системи

Загальний опір $R_x = \sum_{i=1}^{n} R_i$, який визначається безпосереднім вимірюванням, для системи, зображеної на рис. 5.13, потрібно представити у вигляді суми шуканих опорів R_i. Знайдемо кількість можливих комбінацій $P = \prod_{i=1}^{n} p_i$ та базисні числа m_i. Далі виконується така послідовність дій:

$$X = R_x \bmod R_0,\ Y = \frac{XP}{R_0},\ c_i = (m_i \cdot Y) \bmod p_i. \qquad (5.9)$$

Шукані опори кожного резистора визначаються з виразу $R_i = c_i \cdot \Delta R_i$.

Для прикладу візьмемо три резистори опором R_0=10 Ом, p_1=3, p_2=4, p_3=5. Тоді P=60; ΔR_1=3,33 Ом; ΔR_2=2,5 Ом; ΔR_3=2 Ом; m_1=2, m_2=3, m_3=3. Згідно з формулою (5.8), діапазон вимірювання перебуває в межах від 0 до 22,17 Ом.

Нехай прилад зафіксував величину R_x=12,83 Ом. У результаті отримуємо X=12,83mod10=2,83; $Y = \frac{2{,}83 \cdot 60}{10} \approx 17$; c_1=(2·17)mod 3=1; c_2=(3·17)mod 4=3; c_3=(3·17)mod 5=1. Тоді

шукані величини R_1 =1·3,33 Ом=3,33 Ом; R_2 =3·2,5 Ом=7,5 Ом; R_3 =1·2 Ом=2 Ом.

5.5.1. Метод побудови сенсора на основі системи залишкових класів за допомогою таблиць

Шукані опори R_i можна знайти за даними таблиць. Це пришвидшує отримання результатів, однак потребує застосування більшого об'єму пам'яті. Відповідно до попереднього прикладу, будується табл. 5.2. У першому стовпчику знаходяться всі можливі значення параметра X, розміщені в порядку зростання.

Таблиця 5.2

Побудова БПС табличним методом для p_1=3, p_2=4, p_3=5

X	Y	c_1	R_1	c_2	R_2	c_3	R_3	R_x
1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0
0,17	1	2	6,67	3	7,5	3	6	20,17
0,33	2	1	3,33	2	5	1	2	10,33
0,5	3	0	0	1	2,5	4	8	10,5
0,67	4	2	6,67	0	0	2	4	10,67
0,83	5	1	3,33	3	7,5	0	0	10,83
1	6	0	0	2	5	3	6	11
1,17	7	2	6,67	1	2,5	1	2	11,17
1,33	8	1	3,33	0	0	4	8	11,33
1,5	9	0	0	3	7,5	2	4	11,5
1,67	10	2	6,67	2	5	0	0	11,67
1,83	11	1	3,33	1	2,5	3	6	11,83
2	12	0	0	0	0	1	2	2
2,17	13	2	6,67	3	7,5	4	8	22,17
2,33	14	1	3,33	2	5	2	4	12,33
2,5	15	0	0	1	2,5	0	0	2,5
2,67	16	2	6,67	0	0	3	6	12,67
2,83	17	1	3,33	3	7,5	1	2	12,83
3	18	0	0	2	5	4	8	13
3,17	19	2	6,67	1	2,5	2	4	13,17
3,33	20	1	3,33	0	0	0	0	13,33

Продовження табл. 5.2

1	2	3	4	5	6	7	8	9
3,5	21	0	0	3	7,5	3	6	13,5
3,67	22	2	6,67	2	5	1	2	13,67
3,83	23	1	3,33	1	2,5	4	8	13,83
4	24	0	0	0	0	2	4	4
4,17	25	2	6,67	3	7,5	0	0	14,17
4,33	26	1	3,33	2	5	3	6	14,33
4,5	27	0	0	1	2,5	1	2	4,5
4,67	28	2	6,67	0	0	4	8	14,67
4,83	29	1	3,33	3	7,5	2	4	14,83
5	30	0	0	2	5	0	0	5
5,17	31	2	6,67	1	2,5	3	6	15,17
5,33	32	1	3,33	0	0	1	2	5,33
5,5	33	0	0	3	7,5	4	8	15,5
5,67	34	2	6,67	2	5	2	4	15,67
5,83	35	1	3,33	1	2,5	0	0	5,83
6	36	0	0	0	0	3	6	6
6,17	37	2	6,67	3	7,5	1	2	16,17
6,33	38	1	3,33	2	5	4	8	16,33
6,5	39	0	0	1	2,5	2	4	6,5
6,67	40	2	6,67	0	0	0	0	6,67
6,83	41	1	3,33	3	7,5	3	6	16,83
7	42	0	0	2	5	1	2	7
7,17	43	2	6,67	1	2,5	4	8	17,17
7,33	44	1	3,33	0	0	2	4	7,33
7,5	45	0	0	3	7,5	0	0	7,5
7,67	46	2	6,67	2	5	3	6	17,67
7,83	47	1	3,33	1	2,5	1	2	7,83
8	48	0	0	0	0	4	8	8
8,17	49	2	6,67	3	7,5	2	4	18,17
8,33	50	1	3,33	2	5	0	0	8,33
8,5	51	0	0	1	2,5	3	6	8,5
8,67	52	2	6,67	0	0	1	2	8,67
8,83	53	1	3,33	3	7,5	4	8	18,83
9	54	0	0	2	5	2	4	9
9,17	55	2	6,67	1	2,5	0	0	9,17
9,33	56	1	3,33	0	0	3	6	9,33
9,5	57	0	0	3	7,5	1	2	9,5
9,67	58	2	6,67	2	5	4	8	19,67
9,83	59	1	3,33	1	2,5	2	4	9,83

Величина *Y* у другому стовпчику, отримана за допомогою формули (5.9), вказує на порядковий номер відповідного параметра *X*. В останньому, дев'ятому стовпчику вказані можливі значення повного опору, які може зафіксувати прилад.

Якщо вимірювальний прилад зафіксував R_x=12,83 Ом, то X=12,83 mod 10 = 2,83. Знаходимо цю величину в табл. 5.2. Їй відповідають c_1=1, c_2=3, c_3=1. Отже, шукані значення R_1 =3,33 Ом; R_2 =7,5 Ом; R_3 =2 Ом.

На рис. 5.14 зображено графік залежності зміни загального опору R_x від параметра *X*, який змінюється з кроком 0,1667 при p_1=3, p_2=4, p_3=5.

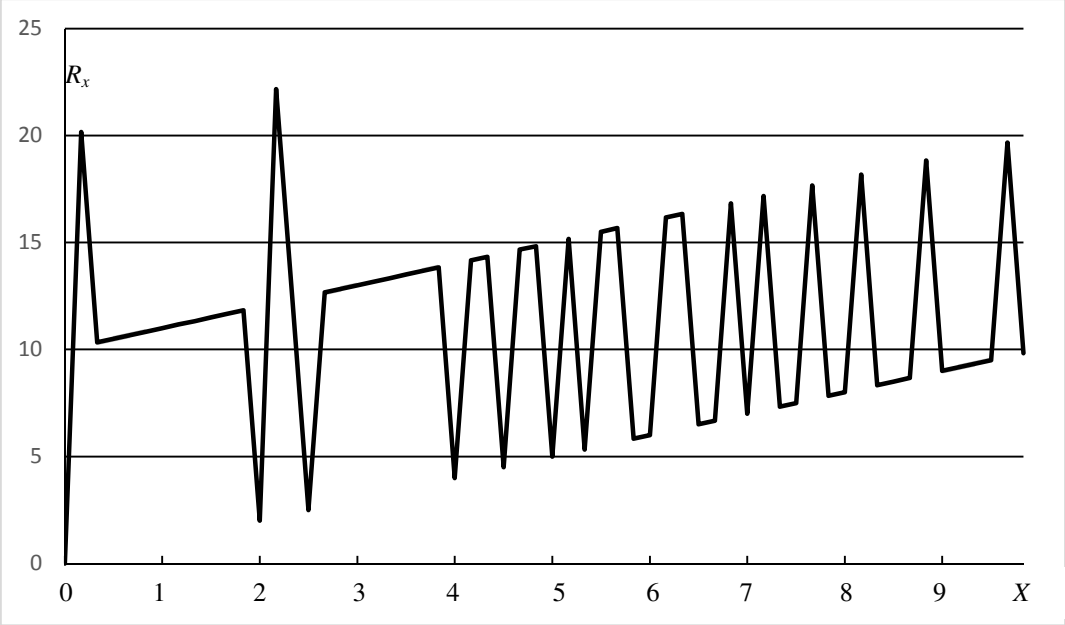


Рис. 5.14. Графік залежності зміни загального опору R_x від параметра *X* при p_1=3, p_2=4, p_3=5

З рис. 5.14 можна визначити, що крива на графіку для R_x має стрибкоподібний характер, причому максимуми (крім двох значень) та мінімуми змінюються лінійно.

5.5.2. Рекомендації щодо вибору наборів модулів з точки зору теорії чисел

Для досягнення приблизно однакової точності вимірювання опору кожного резистора вибрані взаємно прості модулі не мають значно відрізнятися. Прикладом може бути набір з трьох послідовних чисел, зокрема p_1=99, p_2=100, p_3=101 для R_0=100 Ом. За умови більшої кількості модулів їх можна вибрати таким чином: p_1=101, p_2=102, p_3=103, p_4=107, p_5=109, … .

За теорією чисел для істотного зменшення кількості обчислень модулі бажано вибрати так, щоб вони утворювали ДФ СЗК, для яких m_i=1. Це дає змогу уникнути виконання доволі громіздкої операції знаходження оберненого елемента за модулем $m_i = M_i^{-1} \bmod p_i$ та множення Y на базисні числа m_i, оскільки

$$c_i = Y \bmod p_i. \qquad (5.10)$$

У табл. 5.3 подано всі можливі значення відповідних параметрів при модулях, які утворюють ДФ СЗК: p_1=2, p_2=3, p_3=5, опір R_0=10 Ом. Звідси ΔR_1=5 Ом; ΔR_2=3,33 Ом; ΔR_3=2 Ом. Відповідно максимальна величина вимірювання D=19,67 Ом. Нехай прилад зафіксував R_x=13,67 Ом. Тоді X=13,67 mod 10= =3,67; $Y = \frac{3{,}67 \cdot 30}{10} = 11$; c_1=11mod 2=1; c_2=11mod 3=2; c_3=11mod 5=1. Шукані значення R_1 =1·5=5 Ом; R_2=2·3,33=6,67 Ом; R_3=1·2=2 Ом. Такі самі величини можна отримати за даними табл. 5.3.

Недоліком цього методу є те, що у ДФ СЗК модулі дуже швидко зростають і відповідно істотно відрізняються точності вимірювання у різних точках.

Таблиця 5.3

Побудова БПС табличним методом для p_1=2, p_2=3, p_3=5

X	Y	c_1	R_1	c_2	R_2	c_3	R_3	R_x
0	0	0	0	0	0	0	0	0
0,33	1	1	5	1	3,33	1	2	10,33
0,67	2	0	0	2	6,67	2	4	10,67
1	3	1	5	0	0	3	6	11
1,33	4	0	0	1	3,33	4	8	11,33
1,67	5	1	5	2	6,67	0	0	11,67
2	6	0	0	0	0	1	2	2
2,33	7	1	5	1	3,33	2	4	12,33
2,67	8	0	0	2	6,67	3	6	12,67
3	9	1	5	0	0	4	8	13
3,33	10	0	0	1	3,33	0	0	3,33
3,67	11	1	5	2	6,67	1	2	13,67
4	12	0	0	0	0	2	4	4
4,33	13	1	5	1	3,33	3	6	14,33
4,67	14	0	0	2	6,67	4	8	14,67
5	15	1	5	0	0	0	0	5
5,33	16	0	0	1	3,33	1	2	5,33
5,67	17	1	5	2	6,67	2	4	15,67
6	18	0	0	0	0	3	6	6
6,33	19	1	5	1	3,33	4	8	16,33
6,67	20	0	0	2	6,67	0	0	6,67
7	21	1	5	0	0	1	2	7
7,33	22	0	0	1	3,33	2	4	7,33
7,67	23	1	5	2	6,67	3	6	17,67
8	24	0	0	0	0	4	8	8
8,33	25	1	5	1	3,33	0	0	8,33
8,67	26	0	0	2	6,67	1	2	8,67
9	27	1	5	0	0	2	4	9
9,33	28	0	0	1	3,33	3	6	9,33
9,67	29	1	5	2	6,67	4	8	19,67

На рис. 5.15 зображено графік залежності для зміни загального опору R_x від параметра X, який змінюється з кроком 0,33 при значеннях модулів p_1=2, p_2=3, p_3=5.

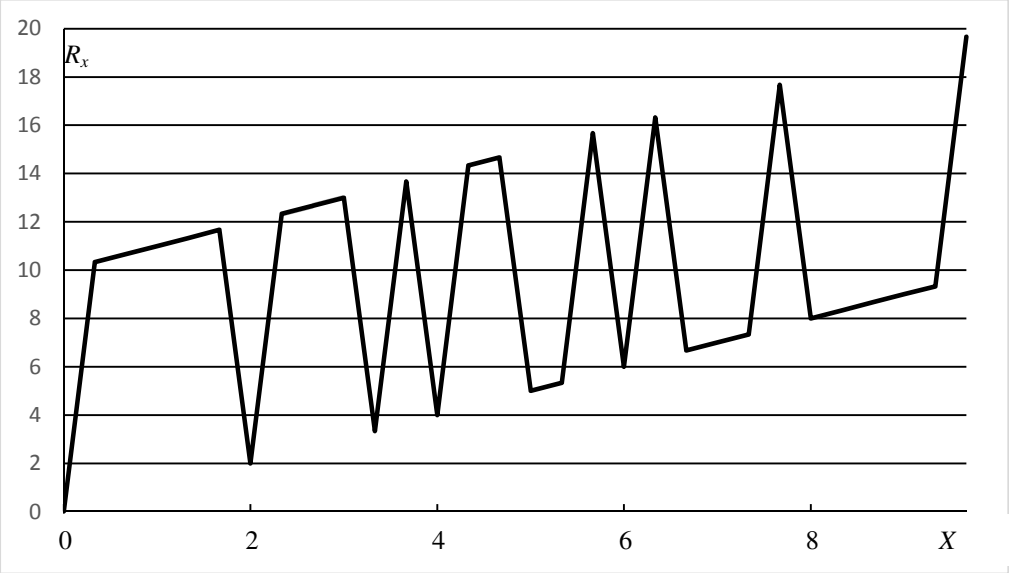


Рис. 5.15.Графік залежності зміни загального опору R_x від параметра X при p_1=2, p_2=3, p_3=5

З рис. 5.15 можна з'ясувати, що крива на графіку для R_x теж має стрибкоподібний характер, однак у цьому разі, на відміну від попереднього, всі максимуми та мінімуми змінюються лінійно. Крім того, графік є обернено симетричним відносно середини діапазону осі абсцис. При малих значеннях X спостерігаються ширші максимуми, при великих – ширші мінімуми.

Якщо вимірювання необхідно виконати лише у двох точках, то набір модулів зручно вибрати у вигляді двох великих послідовних чисел, що забезпечує приблизно однакову високу точність. Такі модулі утворюють МДФ СЗК, для якої виконується умова $m_i = M_i^{-1} \bmod p_i = \pm 1$. Це також дозволяє уникнути пошуку оберненого елемента за модулем та множення Y на базисні числа, оскільки $m_1=p_2 \bmod p_1=(p_1+1) \bmod p_1=1$, $m_2=p_1 \bmod (p_1+1)= =-1 \bmod (p_1+1)=p_1$. Звідси:

$$C_1 = Y \bmod p_1,\ c_2=(p_2 - Y \bmod p_2). \qquad (5.11)$$

У табл. 5.4 подано можливі значення відповідних параметрів при p_1=5, p_2=6, R_0=10 Ом, ΔR_1=2 Ом; ΔR_2=1,67 Ом.

Таблиця 5.4

Побудова БПС табличним методом для p_1=5, p_2=6

X	Y	c_1	R_1	c_2	R_2	R_x
0	0	0	0	0	0	0
0,33	1	1	2	5	8,33	10,33
0,67	2	2	4	4	6,67	10,67
1	3	3	6	3	5	1
1,33	4	4	8	2	3,33	11,33
1,67	5	0	0	1	1,67	1,67
2	6	1	2	0	0	2
2,33	7	2	4	5	8,33	12,33
2,67	8	3	6	4	6,67	12,67
3	9	4	8	3	5	13
3,33	10	0	0	2	3,33	3,33
3,67	11	1	2	1	1,67	3,67
4	12	2	4	0	0	4
4,33	13	3	6	5	8,33	14,33
4,67	14	4	8	4	6,67	14,67
5	15	0	0	3	5	5
5,33	16	1	2	2	3,33	5,33
5,67	17	2	4	1	1,67	5,67
6	18	3	6	0	0	6
6,33	19	4	8	5	8,33	16,33
6,67	20	0	0	4	6,67	6,67
7	21	1	2	3	5	7
7,33	22	2	4	2	3,33	7,33
7,67	23	3	6	1	1,67	7,67
8	24	4	8	0	0	8
8,33	25	0	0	5	8,33	8,33
8,67	26	1	2	4	6,67	8,67
9	27	2	4	3	5	9
9,33	28	3	6	2	3,33	9,33
9,67	29	4	8	1	1,67	9,67

Максимальна величина, яку може вказати вимірювальний прилад, становить D=16,33 Ом. Тоді m_1=1, m_2=5mod 6= =–1 mod 6.

Нехай, наприклад, прилад зафіксував величину R_x=12,67 Ом. Аналітичним методом отримано: X=12,67mod10=2,67; $Y = \frac{2{,}67 \cdot 30}{10} = 8$; c_1=8mod 5=3; c_2=6–(8mod 6)=4. Шукані значення R_1 =3·2=6 Ом; R_2=4·1,67=6,67 Ом. За даними табл. 5.3 отримується той самий результат.

На рис. 5.16 зображено графік залежності зміни загального опору R_x від параметра X, який змінюється з кроком 0,33 при p_1=5, p_2=6.

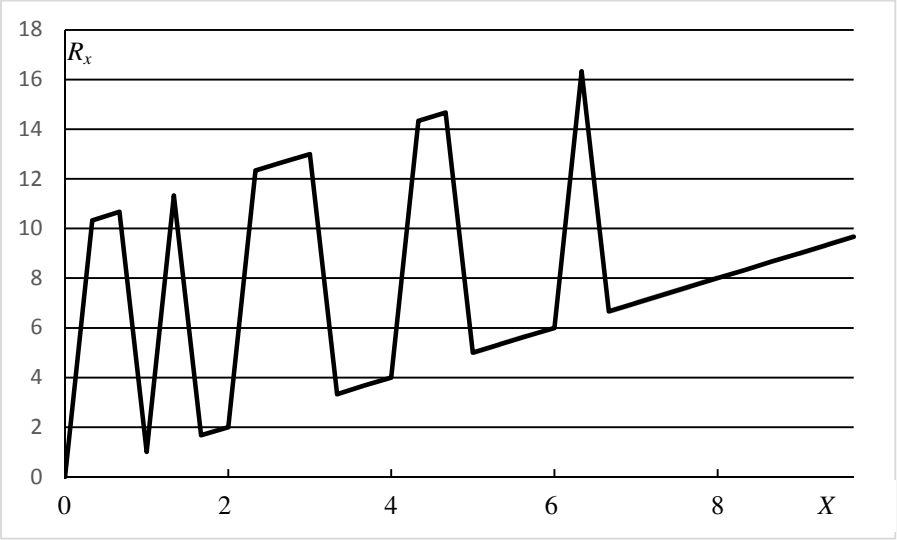


Рис. 5.16. Графік залежності зміни загального опору R_x від параметра X при p_1=5, p_2=6

З рис. 5.16 можна визначити, що кількість максимумів зменшилася, а при збільшенні X мінімуми стають ширшими.

Якщо вимірювання необхідно провести у більш ніж двох точках, то модулі слід вибрати так, щоб вони утворювали МДФ СЗК і неістотно відрізнялися один від одного. Для прикладу

візьмемо чотири резистори, вибравши модулі таким чином: p_1=8, p_2=9, p_3=11, p_4=13 і P=10296. Можна перевірити, що m_1=7mod 8=–1 mod 8, m_2=1, m_3=1, m_4=12mod 13=–1 mod 13, тобто вибрані модулі задовольняють умову МДФ СЗК. Для R_0=10 Ом отримано: ΔR_1=1,25 Ом; ΔR_2=1,11 Ом; ΔR_3=0,91 Ом; ΔR_4=0,77 Ом. Нехай вимірювальний прилад зафіксував величину R_x=11,55 Ом. Тоді аналогічно до формул (5.11) можна отримати шукані значення резисторів: X=11,55mod10=1,55; $Y = \frac{1{,}55 \cdot 10296}{10} \approx 1596$; c_1=8–1596mod 8=8-4=4; c_2=1596mod 9=3; c_3=1596mod 11=1; c_4=13–1596mod 13=13–10=3; R_1=4·1,25 Ом= =5 Ом; R_2 = 3·1,11 Ом=3,33 Ом; R_3=1·0,91 Ом=0,91 Ом; R_4=3·0,77 Ом=2,31 Ом. З отриманих результатів можна визначити, що сума знайдених опорів дорівнює значенню, яке показав прилад.

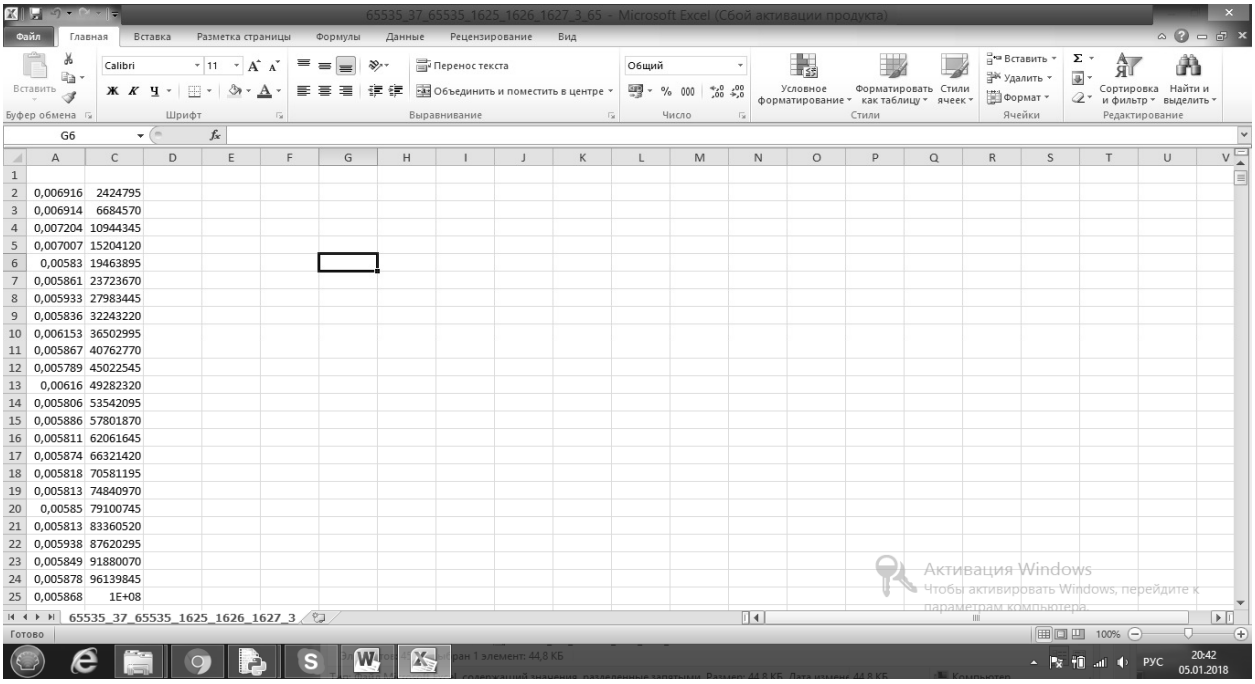
5.6. Експериментальні дослідження програмної реалізації множення у системі залишкових класів та її модифікованій досконалій формі

Для програмної реалізації операції множення в СЗК і МДФ СДК [231] обрана високорівнева мова програмування загального призначення Python [232]. Вибір останньої зумовлений її модульністю та можливістю повторного використання коду. Це є інтерпретована об’єктноорієнтована мова програмування високого рівня із динамічною семантикою. Структури її даних високого рівня разом із динамічною семантикою та динамічним зв’язуванням сприяють швидкій розробці програм, а також є засобом поєднання існуючих компонентів. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формах на всіх

основних платформах. У мові програмування Python підтримується декілька парадигм програмування, зокрема: об'єктноорієнтована, процедурна, функціональна та аспектноорієнтована. Безсумнівною її перевагою є можливість роботи з великорозрядними числами. Приклад введення вхідних параметрів зображено на рис. 5.17.

Рис. 5.17. Головне вікно програми

Результати розміщуються у файл з розширенням csv, ім'я якого записано в останньому рядку головного вікна, що містить усі вхідні параметри. Приклад отриманого файла з часом та результатами множення двох чисел наведений на рис. 5.18. На рис. 5.19 подано часові характеристики виконання операції множення $N=p\cdot q$ у трьохмодульній СЗК.



	A	C
1		
2	0,006916	2424795
3	0,006914	6684570
4	0,007204	10944345
5	0,007007	15204120
6	0,00583	19463895
7	0,005861	23723670
8	0,005933	27983445
9	0,005836	32243220
10	0,006153	36502995
11	0,005867	40762770
12	0,005789	45022545
13	0,00616	49282320
14	0,005806	53542095
15	0,005886	57801870
16	0,005811	62061645
17	0,005874	66321420
18	0,005818	70581195
19	0,005813	74840970
20	0,00585	79100745
21	0,005813	83360520
22	0,005938	87620295
23	0,005849	91880070
24	0,005878	96139845
25	0,005868	1E+08

Рис. 5.18. Приклад отриманого файла

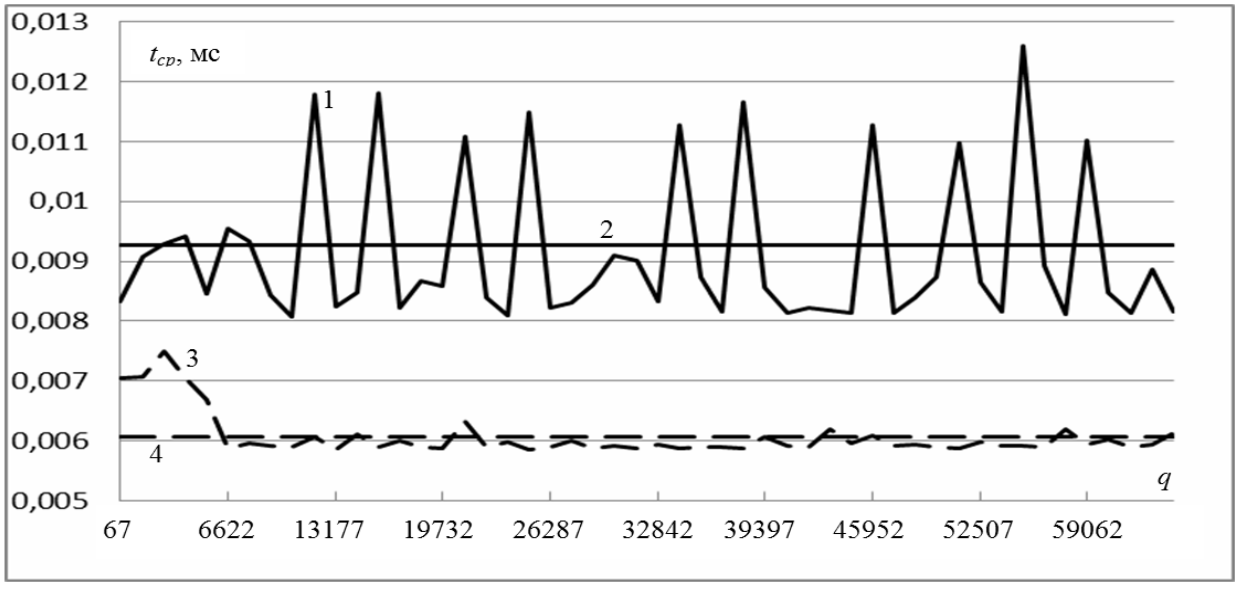


Рис. 5.19. Часові характеристики виконання операції множення в трьохмодульній СЗК

Множник p=65536 фіксований із двома різними системами модулів (перший випадок – модулі незначно відрізняються один від одного: p_1=1625=$\left\lfloor\sqrt[3]{65536^2}\right\rfloor$, p_2=1626, p_3=1627 – пунктирні лінії;

другий випадок – модулі значно відрізняються: p_1=163, p_2=1627, p_3=16381 – суцільні лінії).

Другий множник q змінювався від значення 67 до p з кроком 1311. Останній визначав кількість отриманих обчислень, що дорівнювала 50. Добуток модулів обох систем перевищує 2^{32}.

З рис. 5.19 можна визначити, що графік 1 має осцилюючий характер. Середній час виконання операції множення (лінія 2) дорівнює 0,009259 мс. У другому випадку, крім початкових значень q, час виконання множення (графік 3) не зазнає суттєвих коливань. Середній час (лінія 4) становить 0,006066 мс, що в 1,53 разу менше, ніж у попередньому випадку. Отже, для підвищення швидкодії в СЗК попарно взаємно прості модулі необхідно вибирати так, щоб вони якомога менше відрізнялися один від одного.

Для дослідженні МДФ СЗК система модулів зі значною різницею між ними (p_1=651, p_2=691, p_3=11246) вибиралася за формулою (3.26), яку можна записати таким чином:

$$p_3 = p_1 + \frac{p_1^2 \pm 1}{p_2 - p_1}. \qquad (5.12)$$

При побудові трьохмодульної МДФ СЗК за формулою (5.12) не може бути вибрана система модулів однакової розрядності. Найменша різниця між модулями за такої умови становитиме:

$$p_{2,3} = 2p_1 \pm 1. \qquad (5.13)$$

На основі цього вибрані такі модулі: p_1=1025, p_2=2049, p_3=2051. Добуток модулів в обох випадках перевищує 2^{32}.

Вхідні параметри були ті самі, що і для звичайної СЗК. Обчислення проводилися за таким виразом для МДФ СЗК:

$$A = \left(-b_1 P_1 + b_2 P_2 + b_3 P_3\right) \bmod P, \qquad (5.14)$$

де b_i – залишки.

Отримані результати відображено на рис. 5.20. Суцільна лінія вказує на час виконання множення (крива 1) і середній час (лінія 2) для 50 значень p при p_1=651, p_2=691, p_3=11246, пунктирна (графіки 3, 4) – відповідно для p_1=1025, p_2=2049, p_3=2051.

Отже, в обох випадках при малих значеннях q амплітуда коливань велика, при збільшенні q вона зменшується за винятком невеликого відрізка в другій половині діапазону змін значення q.

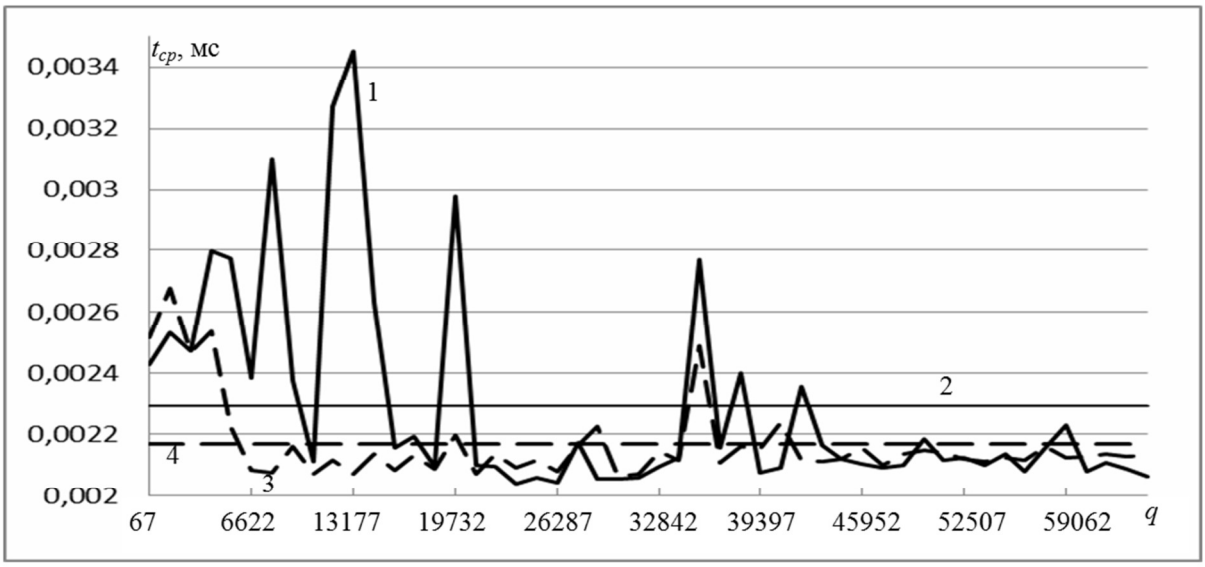


Рис. 5.20. Часові характеристики виконання операції множення в трьохмодульній МДФ СЗК

Середній час для системи модулів p_1=651, p_2=691, p_3=11246 становить 0,002293 мс (лінія 2), а для p_1=1025,

p_2=2049, p_3=2051 – 0,002169 мс (лінія 4), що в 1,057 разу менше, ніж у попередньому разі. Порівняння рис. 5.19 та 5.20 дає змогу встановити суттєве підвищення швидкодії за використання МДФ СЗК. Подальше дослідження проводилося для чисел, розрядність n_0 яких змінювалася від 16 до 24 біт. Були розглянуті чотири випадки побудови системи модулів:

1) модулі СЗК значно відрізняються один від одного;

2) модулями є три послідовних числа, перше і третє з яких непарні: $p_1 \approx \left[\sqrt[3]{2^{2n_0}}\right]$, p_2= p_1+1, p_3= p_1+2;

3) модулі обчислюються за такими формулами: p_2= p_1+1, p_3=p_1(p_1+1)–1;

4) модулі обчислюються за такими виразами: p_2= 2p_1–1, p_3=2p_1+1.

У всіх випадках добуток модулів є мінімальним, але перевищує 2^{2n_0}. У третьому і четвертому випадках системи модулів утворюють МДФ СЗК. Перший множник у добутку N=p·q був фіксованим: p=2^{n_0}–1, що відповідає максимальному числу заданої розрядності. Другий множник q змінювався від початкового значення $q = 2^{n_0} - \left[\frac{2^{n_0}}{1000}\right] + 1$ з кроком $\left[\frac{2^{n_0}}{1000}\right]$. Таким чином, отримано 1000 різних значень числа q і відповідно час виконання 1000 операцій множення A=p·q при фіксованому значенні p і біжучому q. Далі для кожної розрядності визначався середній час виконання операції.

Для нівелювання випадкових впливів на роботу комп'ютера обчислення повторювалися 100 разів. Відповідні набори модулів і середній час обчислень для чисел різних розрядностей подано в табл. 5.5.

Таблиця 5.5

Набори модулів і середній час обчислень для чисел різних розрядностей

n_0		16	17	18	19	20	21	22	23	24
Випадок 1	p_1	163	235	341	501	737	1093	1627	2429	3641
	p_2	1627	2587	4097	6503	10323	16387	26009	41287	65539
	p_3	16381	28413	49165	84541	144523	245807	416147	701881	1179703
$t_{ср}$, мкс		8,154	8,562	8.526	9,319	9,28	9,671	9,749	10,105	10,69
Випадок 2	p_1	1625	2581	4095	6501	10321	16385	26007	41285	65537
	p_2	1626	2582	4096	6502	10322	16386	26008	41286	65538
	p_3	1627	2583	4097	6503	10323	16387	26009	41287	65539
$t_{ср}$, мкс		5,891	5,95	5,962	5,966	5,934	5,93	5,982	7,185	7,304
Випадок 3	p_1	256	362	512	724	1024	1448	2048	2896	4096
	p_2	257	363	513	725	1025	1449	2049	2897	4097
	p_3	65791	131405	262655	524899	1049599	2098151	4196351	8389711	16781311
$t_{ср}$, мкс		5,461	5,98	5,618	5,689	5,65	5,684	5,57	5,732	5,593
Випадок4	p_1	1025	1626	2581	4097	6502	10322	16385	26008	41286
	p_2	2049	3251	5161	8193	13003	20643	32769	52015	82571
	p_3	2051	3253	5163	8195	13005	20645	32771	52017	82573
$t_{ср}$, мкс		5,551	5,351	5,33	5,364	5,365	5,44	5,956	5,959	6,679

На рис. 5.21 зображено графіки залежності середнього часу виконання операції множення від розрядності n_0 чисел, які використовуються згідно з даними табл. 5.5 (номер графіка відповідає номеру випадку в табл. 5.5).

З рис. 5.21 можна визначити, що найбільший час витрачається для звичайної СЗК у першому випадку, коли модулі значно відрізняються один від одного. Відповідно криві на графіку набувають зростання практично лінійно із збільшенням розрядності. Графіки 2 і 4 при малих розрядностях майже лінійні, часові стрибки спостерігаються при n_0=22 та n_0=23 відповідно. І третій графік також лінійний на проміжку розглянутого діапазону. Аналіз рис. 5.21 свідчить про те, що використання модулів, які або утворюють МДФ СЗК, або мало

відрізняються один від одного, дає змогу збільшити швидкодію обчислювальних систем.

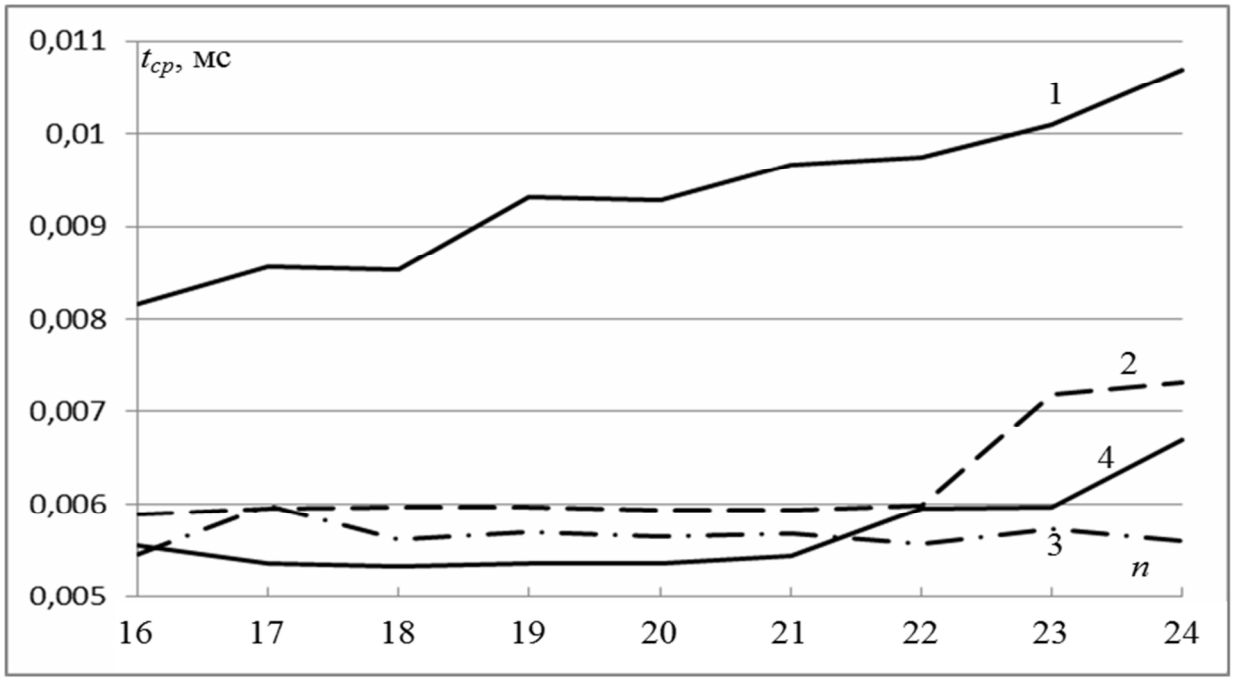


Рис. 5.21. Графіки залежності середнього часу виконання операції множення в СЗК від розрядності множників

На рис. 5.22 зображено графіки залежності середнього часу виконання операції множення від розрядності n_0 для третього (крива 1) і четвертого (крива 2) випадків табл. 5.5, модулі в яких утворюють МДФ СЗК, при тих самих вхідних параметрах з використанням формули (5.14).

Аналіз рис. 5.21, 5.22 підтверджує, що середній час обчислень у МДФ СЗК зменшується приблизно в 2,5–3 рази порівняно із звичайною цілочисельною формою СЗК.

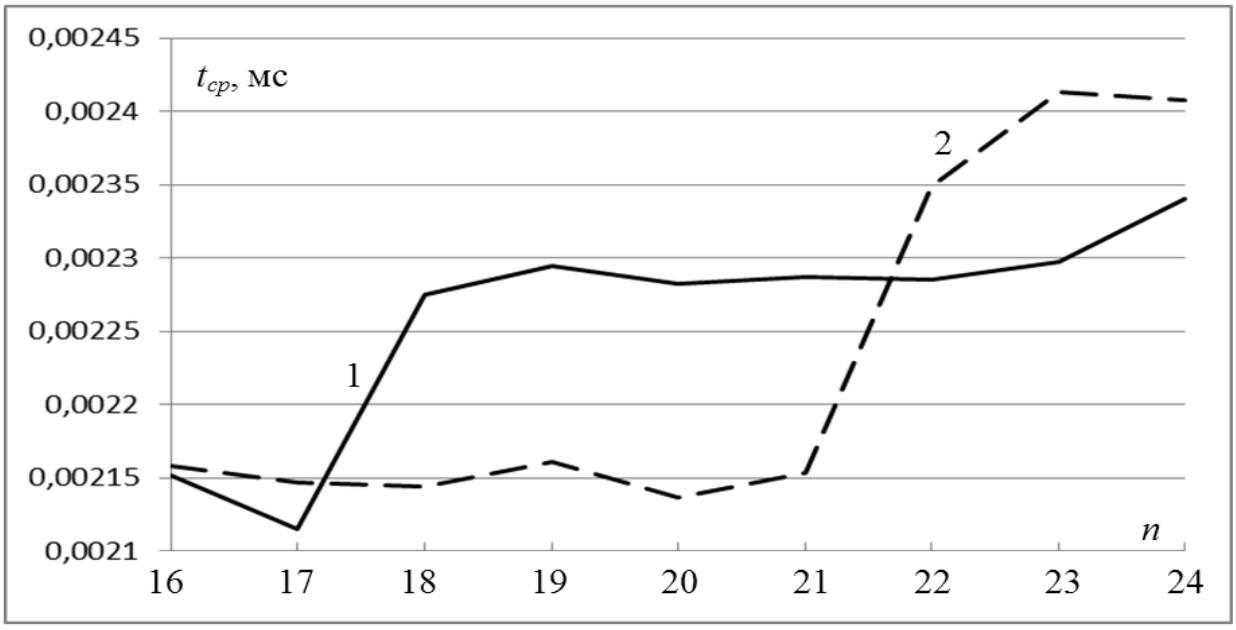


Рис. 5.22. Графіки залежності середнього часу виконання операції множення від розрядності при використанні формули (5.14)

5.7 Експериментальні дослідження програмної реалізації методів модулярного експоненціювання

Для експериментальних досліджень методів модулярного експоненціювання обрано комп'ютер HP ProBook 4540S з процесором Intel i5, тактова частота якого становить 2,3 GHz, оперативна пам'ять – 6 GB, операційна система – Linux Mint 17.1, середовище програмування – Python 3.4.0.

Вибір методів окреслений часом модулярного експоненціювання, який для різних методів мав бути приблизно одного порядку, та можливістю виконання операцій над великорозрядними числами. В зв'язку з цим не досліджувалися методи прямого піднесення до степеня, при якому для 16-бітних чисел відбувалося переповнення розрядної сітки, та множення справа наліво або зліва направо, що потребувало виконання x–1 операцій множення (x – показник степеня) і відповідно

збільшувався час обчислень. З цієї причини не застосовано метод СЗК, коли модуль P є добутком декількох простих співмножників ($P=p_1p_2\ldots p_k$) і його потрібно факторизувати, що призводило до істотного збільшення часової складності. Отже, для дослідження вибрані чотири таких методи [233]:

1) бінарний або метод пониження степеня за допомогою квадратів [234], який описується таким виразом:

$$N = a^x \bmod p = \left(a^{x-2x_1} a_1^{x_1-2x_2} \ldots a_i^{x_i-2x_{i+1}} \ldots \right) \bmod p = \\ = \left(\prod_{i=0}^{\lfloor \log_2 x \rfloor} a_i^{x_i-2x_{i+1}} \right) \bmod p, \tag{5.15}$$

де $a_0=a$, $x_0=x$, $a_i = a_{i-1}^2 \bmod p$; $x_i = \left\lfloor \frac{x_{i-1}}{2} \right\rfloor$;

2) метод пониження степеня за допомогою кубів (3-арний) [224], який можна записати аналогічно до формули (5.15):

$$N = a^x \bmod p = \left(a^{x-3x_1} a_1^{x_1-3x_2} \ldots a_i^{x_i-3x_{i+1}} \ldots \right) \bmod p = \\ = \left(\prod_{i=0}^{\lfloor \log_3 x \rfloor} a_i^{x_i-3x_{i+1}} \right) \bmod p, \tag{5.16}$$

де $a_0=a$, $x_0=x$, $a_i = a_{i-1}^3 \bmod p$; $x_i = \left\lfloor \frac{x_{i-1}}{3} \right\rfloor$;

3) використання СЗК, коли основа степеня розбивається на залишки від ділення на попарно взаємно прості модулі, далі

відбувається пониження степеня бінарним методом і відновлення десяткового запису числа. Загалом цю процедуру можна описати такими виразами:

$$N = a^x \bmod p = \left(\sum_{i=1}^{k} m_i M_i a_i \right) \bmod p, \qquad (5.17)$$

де $a_i = (a \bmod p_i)^x \bmod p_i$, $p = \prod_{i=1}^{k} p_i$, $P_i = \frac{p}{p_i}$, $m_i = P_i^{-1} \bmod p_i$,

k – кількість модулів. Пошук оберненого елемента відбувається за допомогою алгоритму Евкліда та його наслідку;

4) використання МДФ СЗК, згідно з якою модулі підібрані таким чином, що $m_i = P_i \bmod p_i = \pm 1$. Це дає змогу уникнути виконання громіздкої операції пошуку оберненого елемента за модулем та множення у формулі (5.17) на базисні числа m_i, що відповідно приводить до зменшення часу обчислень.

Для модулярного піднесення до степеня $a^x \bmod p$ залежно від розрядності n_0 та кількості одиничок у двійковому записі числа (ваги Хемінга) параметри a, x та p визначалися таким чином: $a=r(n_0–3, \Delta)$, $x=r(n_0, \Delta)$, де цілочисельною є функція

$$r(n_0, \Delta) = 2^{n_0} - 1 - \left[RND \cdot \left[\frac{\Delta \cdot n_0}{100} \right] \right], \qquad (5.18)$$

де $0<RND<1$ – випадкова величина, заданий параметр Δ набуває дискретних значень 0, 10, 30, 50 та 80.

Останній вказує, що при $\Delta=0$ вага Хемінга чисел $a=2^{n_0}-1$ та $x=2^{n_0-3}–1$ дорівнює їхній розрядності. Збільшення цього параметра означає, що Δ% молодших розрядів у двійковому записі a та x може змінитися випадково під дією функції RND.

Для кожного методу піднесення до степеня використовувався один і той самий модуль, який залежить від розрядності числа x і є добутком трьох попарно взаємно простих співмножників $p=p_1 \cdot p_2 \cdot p_3$, що утворюють МДФ СЗК: $p_1 = 2^{\left\lfloor \frac{n_0}{3} \right\rfloor + 1}$, $p_2=2p_1-1$, $p_2=2p_1+1$. Отримані результати наведені в табл. 5.6 (у першому стовпчику 1 – пониження степеня за допомогою квадратів, 2 – за допомогою кубів, 3 – СЗК, 4 – МДФ СЗК).

Слід зазначити, що модуль p перевищує діапазон обчислень, який визначається відповідною розрядністю n_0, приблизно у 8–16 разів. Усі розрахунки для кожного випадку проводилися 10 разів і визначався середній час виконання модулярного експоненціювання (в мікросекундах).

Як підтверджують дані табл. 5.6, швидкість піднесення до степеня за модулем істотно залежить від методу виконання. Так, при малих розрядностях (до 256 включно) методи СЗК та МДФ СЗК істотно поступаються в часі двом іншим (приблизно в 2–4 рази), а найшвидше модулярне експоненціювання виконується методом пониження степеня за допомогою кубів. Це стосується випадку, коли $n_0=512$ і $\Delta=0$. Для інших Δ і тієї самої розрядності мінімальний час буде при використанні бінарного методу, а СЗК та МДФ СЗК відстають приблизно в 1,5 разу і, починаючи з $n_0=1024$, вони забезпечують мінімальний час.

На рис. 5.23 у логарифмічній шкалі зображено графіки залежності часу виконання операції модулярного експоненціювання різними методами від розрядності чисел для $\Delta=0$, тобто для чисел з максимальним значенням ваги Хемінга.

Таблиця 5.6

Час виконання операції модулярного експоненціювання різними способами

Розрядність	8	16	32	64	128	256	512	1024	2048	4096
Δ=0										
1	14	24	55	108	254	746	2251	10727	64628	405196
2	12	23	44	91	200	655	2162	11987	75411	466016
3	58	83	159	299	628	1430	3430	10366	37564	216596
4	62	72	150	266	610	1340	3195	9956	36861	208496
Δ=10										
1	10	18	39	81	179	487	1955	9452	61290	404806
2	9	17	34	66	148	445	1997	10055	68190	458023
3	43	58	114	211	456	997	2858	9266	34831	199364
4	40	53	105	193	451	969	2563	8417	34413	198145
Δ=30										
1	10	18	38	78	177	487	1919	9353	60907	403917
2	9	17	35	68	146	441	2038	10097	68163	457405
3	43	61	115	211	458	998	2848	9009	34768	199534
4	40	53	103	193	443	974	2546	8409	34378	197433
Δ=50										
1	11	18	37	79	177	494	1954	9465	61187	404641
2	8	17	33	67	146	442	1978	10224	68165	457931
3	44	61	113	211	459	1004	2847	9108	34826	199136
4	40	53	103	193	442	965	2553	8443	34361	197780
Δ=80										
1	10	17	37	78	181	492	1918	9419	60949	404217
2	9	18	33	67	146	444	2015	10074	68148	458093
3	42	61	112	209	458	1001	2856	9086	34738	199704
4	40	51	101	194	440	974	2560	8394	34355	197588

Слід зазначити, що при інших значеннях Δ залежності мають приблизно такий самий характер, тому вони не наведені. На рис. 5.24 наведено графіки залежності часу виконання модулярного піднесення до степеня різними методами від значення ваги Хемінга Δ при найбільш поширеній розрядності n_0=1024.

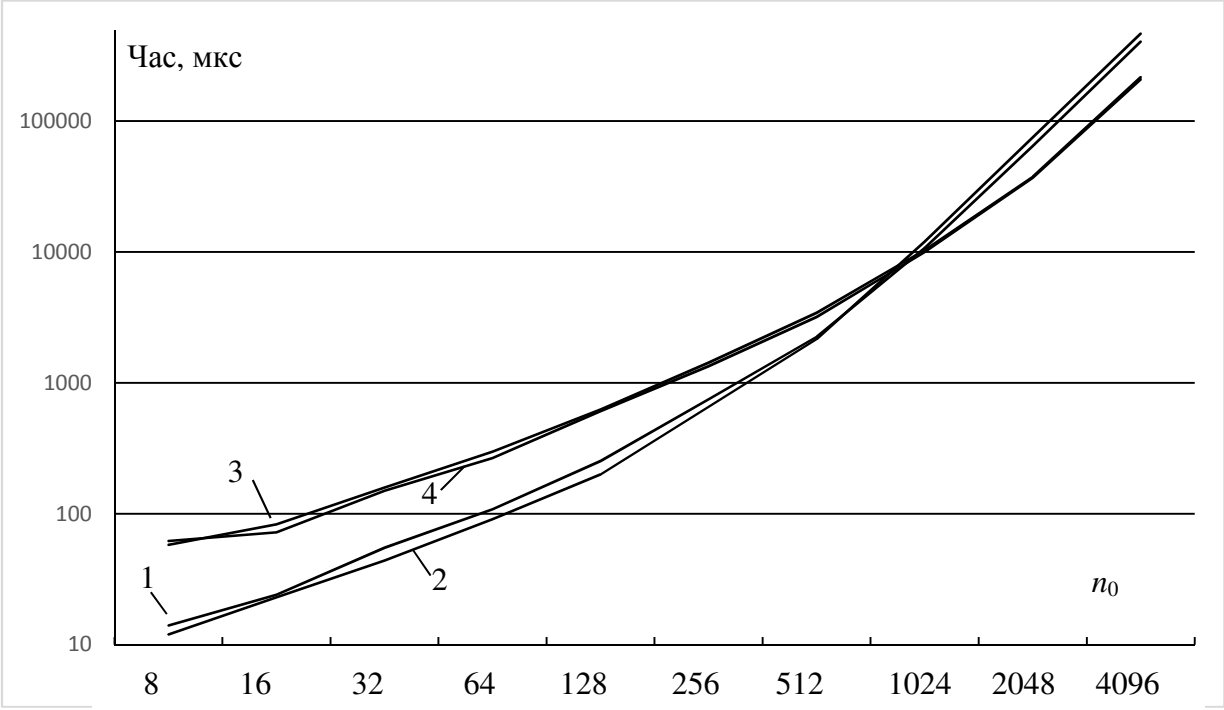


Рис. 5.23.Графіки залежності часу виконання операції модулярного експоненціювання різними методами від розрядності чисел для Δ=0 (1 – пониження степеня за допомогою квадратів, 2 – за допомогою кубів, 3 – використання СЗК, 4 – застосування МДФ СЗК)

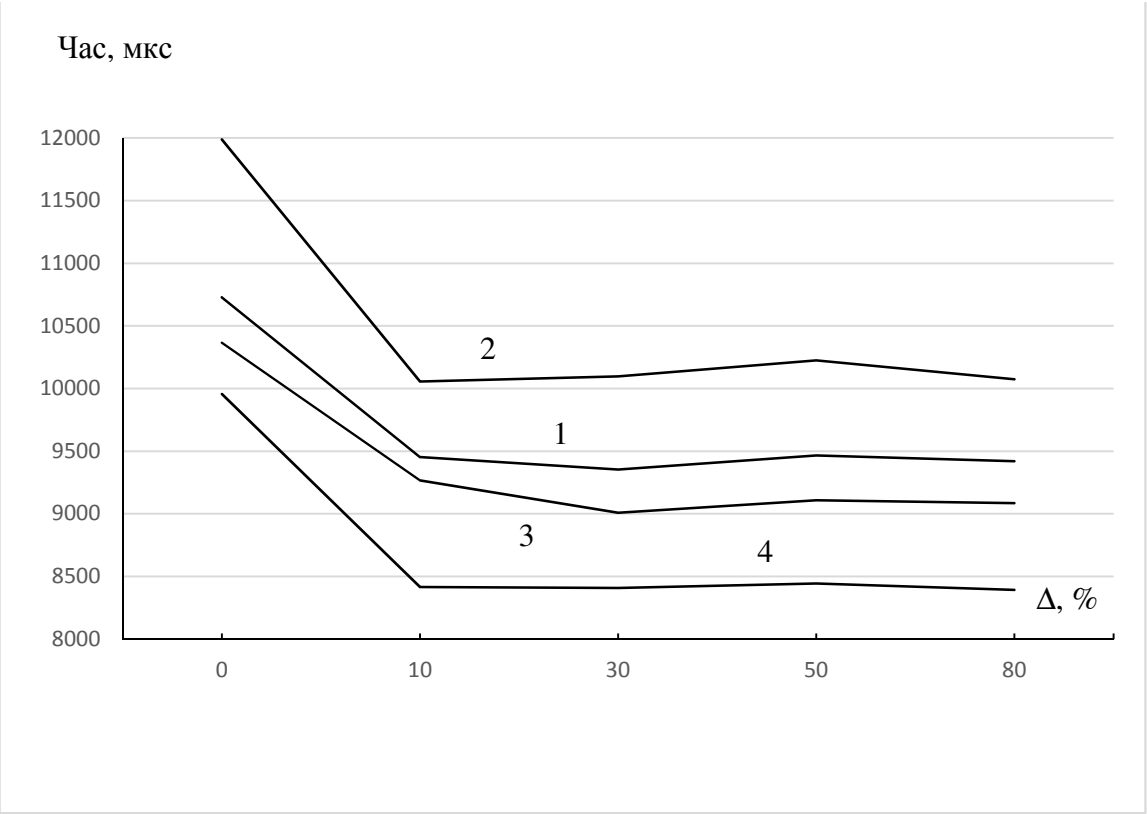


Рис. 5.24. Графіки залежності часу виконання модулярного піднесення до степеня різними методами від значення ваги Хемінга Δ при *n*=1024 (1 – пониження степеня за допомогою квадратів, 2 – за допомогою кубів, 3 – використання СЗК, 4 – застосування МДФ СЗК)

З рис. 5.23 і 5.24 видно, що графіки для СЗК та МДФ СЗК мають однаковий характер. Це пов'язано з тим, що при переведенні чисел із СЗК у десяткову систему числення для модулів, які утворюють МДФ СЗК і використовуються в дослідженні, потрібна незначна кількість ітерацій при застосуванні алгоритму Евкліда, його наслідку та китайської теореми про залишки.

Крім того, з рис. 5.24 можна простежити, що всі графіки мають приблизно однаковий характер. Найбільший час затрачається при максимальному значенні ваги Хемінга (Δ=0). При зменшенні останньої час різко зменшується і надалі залишається майже постійним, здійснюючи невеликі коливання відносно деякого значення.

ЛІТЕРАТУРА

1. Hoffstein J. An Introduction to Mathematical Cryptography / J. Hoffstein, J. Pipher, J. Silverman. – Springer Science+Business Media, LLC, 2008. – 524 p.
2. Jeffrey H. An Introduction to Mathematical Cryptography / H. Jeffrey, P. Jill, H. Joseph. – Berlin : Springer, 2008. – 540 p.
3. Задірака В. К. Комп'ютерна криптологія / В. К. Задірака, О. С. Олексюк. – Тернопіль, К., 2002. – 504 с.
4. Фергюссон Н. Практическая криптография / Н. Фергюссон, Б. Шнайер. – М. : Вильямс, 2005. – 424 с.
5. Вербіцький О. В. Вступ до криптології / О. В. Вербіцький. – Львів : ВНТЛ, 1998. – 247 с.
6. Василенко О. Н. Теоретико-числовые алгоритмы в криптографии / О. Н. Василенко. – М. : МЦНМО, 2006. – 336 с.
7. Shoup V. A Computational Introduction to Number Theory and Algebra / V. Shoup. – Cambridge University Press, 2005. – 517 p.
8. Stillwell J. Elements of Number Theory / Stillwell J. – Springer, 2010 – 256 p.
9. Бородін О. І. Теорія чисел / О. І. Бородін. – К. : Вища школа, 1970. – 275 с.
10. Бухштаб А. А. Теория чисел / А. А. Бухштаб. – М.: Просвещение, 1966. – 384 с.
11. Hardy G. H. An Introduction to the Theory of Numbers / G. H. Hardy, E. M. Wright, A. Wiles. – Oxford University Press, 2008. – 656 p.

12. Виноградов И. М. Основы теории чисел / И. М. Виноградов. – М. – Ижевск : НИЦ «Регулярная и хаотическая динамика», 2003. – 176 с.
13. Zhengbing Hu. The Analysis and Investigation of Multiplicative Inverse Searching Methods in the Ring of Integers Modulo M / Hu Zhengbing, I. Dychka, M. Onai, A. Bartkoviak // International Journal of Intelligent Systems and Applications (IJISA). – 2016. – Vol. 8, №11. – P. 9–18.
14. Parthasarathy S. Multiplicative inverse in mod(m) / S. Parthasarathy // Algologic Technical Report. –2012. – №1. – P. 1–3.
15. Дичка І. А. Способи знаходження мультиплікативного оберненого елемента в скінченних полях / І. А. Дичка, М. В. Онай // «Наукові вісті НТУУ «КПІ». – 2015. – Вип. 2. – С. 160–165.
16. Дичка І. А. Застосування k-арного методу Евкліда для пошуку мультиплікативного оберненого елемента у кільці лишків за модулем m / І. А. Дичка, М. В. Онай, А. Ю. Бартков'як // «Наукові вісті НТУУ «КПІ». – 2016. – Вип. 5. – С. 248–252.
17. Lorencz R. New Algorithm for Classical Modular Inverse / R. Lorencz // Cryptographic Hardware and Embedded Systems. International Workshop. – 2002. – P. 57–70.
18. Sorenson J. Two fast GCD algorithms / J. Sorenson // Journal of Algorithms. – 1994. – V. 16, № 1. – P. 110–144.
19. Vallee B. Dynamical analysis of a class of Euclidean algorithms / B. Vallee // Theoretical Computer Science. – 2003. – V. 297. – P. 447 – 486.
20. Vallee B. Dynamics of the binary Euclidean algorithm: functional analysis and operators / B. Vallee // Algorithmica. – 1998. – Vol. 22, № 4. – P. 660–685.

21. Гаусс К. Ф. Труды по теории чисел / К. Ф. Гаусс ; общ. ред. акад. И. М. Виноградова. – М. : Изд-во АН СССР, 1959. – 297 с.
22. Valach M. Origin of the code and number system of remainder classes / M. Valach, A. Svoboda // Stroje Na Zpracovani Informaci. – 1955. – Vol. 3. – P. 121–134.
23. Кузьмин И. В. Основы информации и кодирования / И. В. Кузьмин, В. А. Кедрус. – К. : Вища школа, 1986. – 238 с.
24. Курко А. М. Введення в теорію інформації / А. М. Курко, В. Я. Решетник– Тернопіль : Вид-во ТНТУ ім. Івана Пулюя, 2017. – 108 с.
25. Жураковський Ю. П. Теорія інформації і кодування / Ю. П. Жураковський, В. П. Полторак. – К. : Вища школа, 2001. – 255 с.
26. Рабинович З. Л. Типовые операции в вычислительных машинах / З. Л. Рабинович, В. А. Раманаускас. – К. : Техніка, 1980. –264 с.
27. Хетагуров Я. А. Проектирование автоматизированных систем обработки информации и управления / Я. А. Хетагуров. – М. : Высшая школа, 2006. – 223 с.
28. Roy R. Comparative Study and Analysis of Performances among RNS, DBNS, TBNS and MNS for DSP Applications / R. Roy, D. Datta, S. Bhagat, S. Saha, A. Sinha // Journal of Signal and Information Processing. – 2015. – Vol. 6. – P. 49–65.
29. Ghosh A. A New Architecture for FPGA Implementation of a MAC Unit for Digital signal Processors using Mixed number System / A. Ghosh, S. Singha, A. Sinha // Computer Architecture News. – 2012. – Vol. 40. – P. 33–38.
30. Kundu P. An Efficient Architecture of RNS Based Wallace Tree Multi-plier for DSP Application / P. Kundu, O. Bandyopadhyay, A. Sinha // Circuits and Systems. – 2008. – Vol. 48. – P. 221–224.

31. Singha S. A New Architecture for FPGA based Implementation of Conversion of Binary to Double Base Number System (DBNS) Using Parallel Search Technique Singha / S. Singha, A. Ghosh, A. Sinha // Computer Architecture News. –2011. – Vol. 39. – P. 12–18.
32. Deryabin M. High Performance Parallel Computing in Residue Number System / M. Deryabin, N. Chervyakov, A. Tchernykh, M. Babenko, M. Shabalina // International Journal of Combinatorial Optimization Problems and Informatics. – 2018. – Vol. 9 (1). – P. 62–67.
33. Грицык В. В. Распараллеливание алгоритмов обработки информации в системах реального времени / В. В. Грицык. – К. : Наукова думка, 1981. – 216 с.
34. Lin K.-L. Modular low-power high-speed CMOS analog to digital converter for embedded systems / K.-L. Lin, A. Kemma, B. Hosticka. – Boston : Kluwer Academic Publishers, 2008. – 254 p.
35. Гофф М. К. Сетевые распределенные вычисления: достижения и проблемы / М. К. Гофф. – М. : Кудиц-образ, 2006. – 320 с.
36. Гриценко В. И. Распределенные информационные системы. Состояния. Перспективы развития / В. И. Гриценко, А. А. Урсатьев // Управляющие системы и машины. – 2003. – № 4. – С. 11–21.
37. Кондалев А. И. Высокопроизводительные преобразователи формы информации / А. И. Кондалев. – К. : Наукова думка, 2007. – 280 с.
38. Стемпковский А. Л. Особенности реализации устройств цифровой обработки сигналов в интегральном исполнении с применением модулярной арифметики / А. Л. Стемпковский,

А. И. Корнилов, М. Ю. Семенов // Информационные технологии. – 2004. – № 2. – С. 2–9.

39. Корнилов А. И. Методы аппаратной оптимизации сумматоров для двух операндов в системе остаточных классов / А. И. Корнилов, М. Ю. Семенов, В. С. Калашников // Электроника. – 2004. – № 1. – С. 75–82.
40. Svoboda A. Rational numerical system of residual classes / A. Svoboda // Stroje Na Zpracovani Informaci. – 1957. – Vol. 1. – P. 33–48.
41. Svoboda A. The numerical system of residue classes in mathematical machine / A. Svoboda // Information processing. – 1960. – Vol. 2. – P. 81–100.
42. Акушский И. Я. Арифметические операции в системе остаточных классов / И. Я. Акушский // Вопросы радиоэлектроники. – 1960. – Сер. VII, в. 3. – 254 с.
43. Акушский И. Я. Основы машинной арифметики комплексных чисел / И. Я. Акушский, В. М. Амербаев, И. Т. Пак. – Алма-Ата : Наука, 1970. – 248 с.
44. Акушский И. Я. Машинная арифметика в остаточных классах / И. Я. Акушский, Д. И. Юдицкий. – М. : Сов. радио, 1968. – 460 с.
45. Торгашев В. А. Система остаточных классов и надежность ЦВМ / В. А. Торгашев. – М. : Сов. радио, 1973. – 120 с.
46. Амербаев В. М. Теоретические основы машинной арифметики / В. М. Амербаев. – Алма-Ата : Наука, 1976. – 324 с.
47. Амербаев В. М. Быстродействующий согласованный фильтр, построенный по модулярному принципу / В. М. Амербаев, А. Л. Стемпковский, Г. Э. Широ // Информационные технологии. – 2004. – № 9. – С. 5–12.

48. Червяков Н. И. Нейрокомпьютеры в остаточных классах / Н. И.Червяков, П. А.Сахнюк, А. В.Шапошников, А. И. Макоха. – М. : Радиотехника, 2003. – 272 с.
49. Червяков Н. И. Приближенный метод сравнения модулярных чисел и его применение для деления чисел в системе остаточних классов / Н. И. Червяков, М. Г. Бабенко, П. А. Ляхов, И. Н. Лавриненко // Кибернетика и системный анализ . – 2014. – Т. 50 (6). – С. 176–186.
50. Червяков Н. И. Модулярные параллельные вычислительные структуры нейропроцессорных систем / Н. И.Червяков, П. А.Сахнюк, А. В.Шапошников, С. А. Рядnов. – М. : Физматлит, 2003. – 288 с.
51. Краснобаев В. А. Модели и методы обработки данных в системе остаточних классов / В. А. Краснобаев, С. А. Кошман, С. А. Мороз, В. Н. Курчанов, А. С. Янко. – Х. : ООО «В деле», 2017. – 197 с.
52. Барсов В. И. Методология параллельной обработки информации в модулярной системе счисления / В. И. Барсов, Л. С. Сорока, В. А. Краснобаев. – Х. : УИПА, 2009. – 268 с.
53. Krasnobayev V. A. The method of error correction in the system of residual classes / V. A. Krasnobayev, A. S. Yanko, S. A. Koshman // Nauka i studia. – 2015. – № 5 (136). – P. 51–62.
54. Krasnobayev V. A. Method for arithmetic comparison of data represented in a residue number system / V. A. Krasnobayev, A. S. Yanko, S. A. Koshman // Cybernetics and Systems Analysis. – 2016. – Vol. 52, is. 1. – P P. 145–150.
55. Николайчук Я. М. Теоретичні основи базисних перетворень СЗК / Я. М. Николайчук, Ю. С. Федорович // Автоматика

2000 : матеріали Наук. конф. (м. Львів, 2000 р.). – Львів, 2000. – С. 120.

56. Николайчук Я. М. Проектування спеціалізованих комп'ютерних систем / Я. М. Николайчук, Н. Я. Возна, І. Р. Пітух. – Тернопіль : ТзОВ «Терн граф», 2010. – 392 с.
57. Николайчук Я. М. Теорія джерел інформації / Я. М. Николайчук. – Тернопіль : ТзОВ «Терн граф», 2010. – 536 с.
58. Николайчук Я. М. Коди поля Галуа: теорія та застосування / Я. М. Николайчук. – Тернопіль : ТзОВ «Терн граф», 2012. – 576 с.
59. Omondi A. Residue number systems: theory and implementation / A. Omondi, B. Premkumar. – London : Imperial College Press, 2007. –296 p.
60. Vinod A. P. A memoryless reverse converter for the 4-moduli superset $\{2^n–1,\ 2^n\ 2^n+1,\ 2^{n+1}–1\}$ / A. P. Vinod, A. B. Premkumar // Journal of Circuits, Systems and, Computers. – 2000. – Vol. 10 (1-2). – P. 85–99.
61. Premkumar A. B. A format framework for conversion from binary to residue numbers / A. B. Premkumar // IEEE Transactions Sircuit Systems. – 2002. – Vol. 49. – P. 135–144.
62. Ananda Mohan P. V. RNS to binary conversion using diagonal function and Pirlo and Impedovo monotonic function / P. V. Ananda Mohan // Circuits Systems Signal Processing. – 2016. – Vol. 35. – P. 1063–1076.
63. Ananda Mohan P. V. Residue Number Systems: Theory and Applications / P. V. Ananda Mohan. – Birkhäuser, 2016. – 351 p.
64. Финько О. А. Модулярная арифметика параллельных логических вычислений / О. А. Финько. – М. : Ин-т проблем управления РАН, 2003. – 214 с.

65. Завало С. Т. Алгебра і теорія чисел / С. Т. Завало, В. Н. Костарчук, Б. І. Хацет – К. : Вища школа, 1977. – 400 с.
66. Минеев М. П. Об одном применении китайской теоремы об остатках к шифру Виженера / М. П. Минеев, В. Н. Чубариков // Доклады Академии наук. – 2010. – Т. 430, №1. – С. 21–22.
67. Hema V. Data Integrity Checking Based On Residue Number System and Chinese Remainder Theorem In Cloud / V. Hema, M. Ganaga Durga // International Journal of Innovative Research in Science, Engineering and Technology. 2014. – Vol. 3 (3). – P. 2584–2588.
68. Шенец Н. Н. Модуляроное разделение секрета и системы электронного голосования / Н. Н. Шенец // Вестник БГУ. – 2011. – № 1. – С. 101–104.
69. Ростовцев В. С. Модулярные высокоточные параллельные вычисления с использованием нейросетевых технологий / В. С. Ростовцев, Е. И. Зорин, Е. А. Грачёв // Вестник СибГАУ: Математика, механика, информаика. – 2013. – № 4(50). – С. 71–74.
70. Калмыков И. А. Структурная организация параллельного спецпроцессора цифровой обработки сигналов, использующего модулярные код / И. А. Калмыков, М. И. Калмыков // Теория и техника радиосвязи. – 2014. – № 2. – С. 60–66.
71. Pikh V. Synthesis of High-performance Components of Spectral Analyzers and Special Processors for Data Encryption in Rademacher-Krestenson's Theoretical-numerical Basis / V. Pikh, V. Kimak, B. Krulikovskyi // Proceedings of the 13th International Conference of The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2015). – 2015. – P. 182–184.

72. Vozna N. System complexity criteria and synthesis of high-performance multifunctional parallel ADC in Rademacher's and Haar-Krestenson's theoretical and numerical bases / N. Vozna, Ya. Nykolaichuk, O. Zastavnyy, V. Pikh // Proceedings of the 14th International Conference of The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2017). – 2017. – P. 218–221.
73. Chang C. A division algorithm for residue numbers / Chin - Chen Chang, Yeu-Pong Lai // Applied Mathematics and Computation – 2006. – Vol. 172 (1). – P. 368–378.
74. Червяков Н. И. Умножение и деления чисел в системе остаточних клас сов с использованием полей Галуа GP(p) / Н. И. Червяков, М. Г. Бабенко, П. А. Ляхов, И. Н. Лавриненко, А. М. Лягин // Научно-технические ведомости СПбГПУ. – 2014. – Т. 3 (198). – С. 65–76.
75. Labafniya M. Non-iterative RNS Division Algorithm / M. Labafniya, M. Eshghi // Proceedings of the International multiconference of engineers and computer scientists. – 2012. – Vol. 1. – P. 132–135.
76. Smyk R. Pipelined division of signed numbers with the use of residue arithmetic for small number range with the programmable gate array / R. Smyk, Z. Ulman, M. Czyżak // Electrical Engineering. – 2013. – № 76. – P. 117–126.
77. Labafniya M. RNS division algorithm for 2^n–1 and 2^n dividers / M. Labafniya, M. Eshghi // Proceedings of the 22nd Iranian Conference on Electrical Engineering (ICEE). – 2014. – P. 111–114.
78. Lu M. A novel division algorithm for the residue number system / Mi Lu, Jen-Shiun Chiang // IEEE Transactions on Computers. – 1992. – Vol. 41(8). – P. 1026–1032.

79. Bajard J.-C. A new Euclidean division algorithm for residue number systems / J.-C. Bajard, L.-S. Didier, J.-M. Muller // Journal VLSI Signal Processing. – 1998. – Vol. 19 (2). – P. 167–178.
80. Hitz M. A. Integer division in residue number systems / M. A. Hitz, E. Kaltofen // IEEE Transaction Computation – 1995. – Vol. 44 (8). – P. 983–989.
81. Vivek N. Design of RNS Based Addition Subtraction and Multiplication Units / N. Vivek, K Anusudha // International Journal of Engineering Trends and Technology. – 2014. – Vol. 10 (12). – P. 593–596.
82. Beuchat J.-L. Some Modular Adder and Multipliers for Field Programmable Gate Arrays / J.-L. Beuchat // IEEE Proceedings of International Symposium on Parallel and Distributed Processing. – 2010. –Vol. 17. – P. 8–11.
83. Lalitha K. V. High performance adder using residue number system / K. V. Lalitha, V. Sailaja // International Journal of VLSI and Embedded Systems. – 2014. – Vol. 05. – P. 1323–1332.
84. Vergos H. On the design of efficient modular adders / H. Vergos // Journal Circuits, Systems and Computation – 2005. – Vol. 14 (5). – P. 965–972.
85. Jaberipur G. On building general modular adders from standard binary arithmetic components / G. Jaberipur, B. Parhami, and S. Nejati // Proceedings 45th Asilomar Conference Signals, Systems, and Computers. – 2011. – P. 6–9.
86. Angel M. A. Improving system performance by using prefix adders in RNS / M. A. Angel, A. Narendrakumar // International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. – 2016. – Vol. 5 (9). – P. 1–5.
87. Isupov K. RNS-based Data Representation for Handling Multiple-Precision Integers on Parallel Architectures /

K. Isupov, V. Knyazkov // International Conference on Engineering and Telecommunication. – 2016. – P. 76–79.

88. Anitha K. Design and Implementation of Modified Sequential Parallel RNS Forward Converters / K. Anitha, T.S. Arulananth, R. Karthik, P. Bhaskara Reddy // International Journal of Applied Engineering Research. – 2017. – Vol. 12 (16). – P. 6159–6163.
89. Reshadinezhad M. A Novel Low Complexity Combinational RNS Multiplier Using Parallel Prefix Adder / M. R. Reshadinezhad, F. K. Samani // International Journal of Computer Science. – 2013. – Vol. 10 (3). – P. 430–440.
90. Yang L. L. A residue number system based parallel communication scheme using orthogonal signaling: Part II– Multipath fading channels / L. L. Yang, L. Hanzo // IEEE Transactions on Vehicular Technology – 2002. – Vol. 51. – P. 1541–1553.
91. Harvey D. Faster arithmetic for number-theoretic transforms / David Harvey // Journal of Symbolic Computation. – 2014. – Vol. 60. – P. 113–119.
92. Николайчук Я. М. Теорія цифрових перетворень мультибазисного супершвидкодіючого процесора / Я. М. Николайчук // Искусственный интеллект. – 2008. – № 4. – С. 387–394.
93. Tomczak T. Hierarchical residue number systems with small moduli and simple converters / T. Tomczak // International Journal of Applied Mathematics and Computer Science. – 2011. – Vol. 21 (1). – P. 173–192.
94. Краснобаев В. А. Математическая модель процесса табличной реализации операций алгебраического умноження в классе вычетов / В. А. Краснобаев, Е. В. Загуменная,

С.А. Мороз, В.О. Жадан // Радіоелектронні і комп'ютерні системи. – 2012. – Т.11 (2). – С. 281–287.

95. Кошман С. А. Метод реализации арифметических операций в модулярной арифметике на основе использования малоразрядных двоичных сумматоров / С. А. Кошман, Н. С. Деренько // Радіоелектронні і комп'ютерні системи. – 2007. – Т.7 (26). – С. 219–221.
96. Волинський О. І. Швидкодія міжбазисних перетворювачів Радемахера-Крестенсона / О. І. Волинський // Юриспруденція та проблеми інформаційного суспільства (ЮПІС – 2011) : зб. матеріалів Проблемно-наук. міжгалуз. конф. – 2011. – С. 71–75.
97. Jablonski J. Pipeline processing for serial realization of basical arithmetical operations / J. Jablonski // Proceedings of The International Workshop on Discrete-Event System Design (DESDes'01). – 2001. – P. 85–90.
98. Николайчук Я. М. Теоретичні основи побудови та структура спецпроцесорів в базисі Крестенсона / Я. М. Николайчук, О. І. Волинський, С. В. Кулина // Вісник Хмельницького національного університету. – 2007. – № 3 (1). – С. 85–90.
99. Hiasat A. A. Semi-custom VLSI design and implementation of a new efficient RNS division algorithm / A. A. Hiasat, H. A. A. Zohdy // Computer Journal. – 1999. – Vol. 42 (3). – P. 232–240.
100. Краснобаев В. А. Концепция создания отказоустойчивых компьютерных систем обработки информации в системе остаточных классов на основе применения ПЛИС / В. А. Краснобаев, С. А. Кошман, А. И. Тыртышников, Н. С. Гаркавенко // Системи обробки інформації. – 2013. – В. 7(114). – С. 79–82.
101. Краснобаев В. А. Метод повышения достоверности контроля данных, представленных в системе остаточных

классов / В. А. Краснобаев, С. О. Кошман, М. О. Маврина // Кибернетика и системний анализ. – 2014. – Т. 50 (6). – С. 167 –175.

102. Цзюнь С. Спецпроцессор кодирования изображений в системе остаточных классов / Су Цзюнь., В. В. Яцкив, А. А. Саченко, Ху Чежньбин // Современные информационные и электронные технологии (СИЭТ-2012): труды МНПК – Одесса, 2012. – С. 95.
103. Цаволык Т. Г. Метод исправления ошибок на основе модулярных корректирующих кодов / Т. Г. Цаволык, В. В. Яцкив // Вестник Брестского государственного технического университета. Физика, математика, информатика. – 2015. – № 5 (850). – С. 36 – 38.
104. Яцків В. В. Виявлення та виправлення багатократних помилок на основі модулярних коректуючих кодів / В. В. Яцків // Інформаційні технології та комп'ютерна інженерія. – 2015. – Т. 33, № 2. – С. 77–82, 88.
105. Яцків В. В. Двовимірні коректуючі коди на основі модулярної арифметики / В. В. Яцків, Т. Г. Цаволик // Вісник Хмельницького національного університету. Технічні науки. – 2015. – № 4 (227). – С.144–148.
106. Яцків В. В. Метод підвищення надійності передачі даних в безпровідних сенсорних мережах на основі системи залишкових класів / В. В. Яцків // Радіоелектроніка та інформатика. – 2010. – № 2. – С. 32–35.
107. Zhengbing Hu. Increasing the Data Transmission Robustness in WSN Using the Modified Error Correction Codes on Residue Number System / Hu Zhengbing, V. Yatskiv, A. Sachenko // Elektronika ir Elektrotechnika. – 2015. – Vol. 21(1). – P. 76–81.
108. Roshanzadeh M. Error Detection & Correction in Wireless Sensor Networks By Using Residue Number Systems /

M. Roshanzadeh, S. Saqaeeyan // International Journal of Computer Network and Information Security. – 2012. – № 2. – P. 29–35.

109. Aremu I. A. Information encoding and decoding using Residue Number System for $\{2^{2n}-1, 2^{2n}, 2^{2n}+1\}$ moduli sets / I. A. Aremu, K. A. Gbolagade // International Journal of Advanced Research in Computer Engineering & Technology. – 2017. – Vol. 6 (8). – P. 1260–1267.
110. Xiao H. New Error Control Algorithms for Residue Number System Codes / H. Xiao, H. Garg, J. Hu, G. Xiao // Electronics and Telecommunications Research Institute. – 2016. – Vol. 38 (2). – P. 326–336.
111. Lo H. Parallel Algorithms for Residue Scaling and Error Correction in Residue Arithmetic / H. Lo, T. Lin // Wireless Engineering and Technology. – 2013. – Vol. 4 (4). – P. 198–213.
112. Krasnobayev V. A. Method for arithmetic comparison of data represented in a residue number system / V. A. Krasnobayev, A. S. Yanko, S. A. Koshman // Cybernetics and Systems Analysis. – 2016. – Vol. 52 (1). – P. 145–150.
113. Исупов К. С. Об одном алгоритме сравнения чисел в системе остаточных классов / К. С. Исупов // Вестник Астраханского государственного технического университета. – 2014. – № 3. – С. 40–49.
114. Torabi Z. Low-Power/Cost RNS Comparison via Partitioning the Dynamic Range / Z. Torabi, G. Jaberipur // IEEE Transactions on Very Large Scale Integration Systems. – 2016. – Vol. 24 (5). – P. 1849–1857.
115. Keir Y. A. Division and overflow detection in residue number systems / Y. A. Keir, P. W. Cheney, M. Tanenbaum // IRE Transactions on Electronic Computers – 1962. – Vol. EC–11. – P. 501–507.

116. Rouhifar M. Fast Overflow Detection in Moduli Set {2^n-1, 2^n, 2^n+1} / M. Rouhifar, M. Hosseinzadeh, S. Bahanfar, M. Teshnehlab // International Journal of Computer Science Issues. 2011. – Vol. 8 (3). – P. 407–414.
117. Askarzadeh M. A New approach to overflow detection in moduli set {2^n-3, 2^n-1, 2^n+1, 2^n+3} / M. Askarzadeh, M. Hosseinzadeh, K. Navi // Second International Conference on Computer and Electrical Engineering. – 2009. – P. 439–442.
118. Rouhifar M. A new approach to overflow detection in moduli set {2^n–1, 2^n, 2^n+1} / M. Rouhifar, M. Hosseinzadeh, M. Teshnehlab // International Journal of Computational Intelligence and Information Security. – 2011. – Vol. 2 (3). – P. 35–43.
119. Singh N. An overview of Residue Number System / N. Singh // National Seminar on Devices, Circuits & Communication. – 2008. – P. 132–135.
120. Garner H.L. The Residue Number System / L. G. Harvey // IRE Transactions on Electronic Computers. – 1959. – Vol. EC-8 (2). – P. 140–147.
121. Sharoun A. O. Residue number system / A. O. Sharoun // Electrical Engineering. – 2013. – № 76. – P. 265–270.
122. Bavya V. Optimizing the Precision of Digital Signal Processors Using Residue Number System / V. Bavya, R. Uthira Devi // Imperial Journal of Interdisciplinary Research. – 2016. – Vol. 2 (4). – P. 1113–1122.
123. Kornerup P. Finite Precision Number Systems and Arithmetic / P. Kornerup, D. W. Matula. – Cambridge : University Press, 2010. – 699 p.
124. Исупов К. С. Арифметика многократной точности на основе систем остаточных классов / К. С. Исупов,

В. С. Князьков // Программные системы: теория и приложения. – 2016. – № 1(28). – С. 61–97.

125. Кошман С. А. Концепция реализации немодульных операций в модулярной системе счисления / С. А. Кошман // Проблеми інформації : Друга міжнар. наук.-техн. конф. – К., 2014. – С. 94–95.
126. Червяков Н. И. Методы, алгоритмы и техническая реализация основных проблемных операций, выполняемых в системе остаточных классов / Н. И. Червяков // Инфокоммуникационные технологии. – 2011. – № 4. – С. 4–12.
127. Лавриненко А. Н. Исследование немодульных операций в системе остаточных классов / А. Н. Лавриненко, Н. И. Червяков // Научные ведомости Белгородского государственного университета. – 2012. – № 1 (120), вып. 21/1. – С. 110–122. – (Серия : Информатика).
128. Fin'ko O. A. Large Systems of Boolean Functions: Realization by Modular Arithmetic Methods / O. A. Fin'ko // Automation and Remote Control. – 2004. – Vol. 65 (6). – P. 871–892.
129. Vassalos E. SUT-RNS Residue-to-Binary Converters Design / E. Vassalos, D. Bakalis, H. T. Vergos // 15th Euromicro Conference on Digital System Design. – 2012. – P. 65–72.
130. Vassalos E. SUT-RNS forward and reverse converters / E. Vassalos, D. Bakalis, H. T. Vergos // In Proc. IEEE Comput. Society Annual Symp. VLSI. – 2010. – P. 11–16.
131. Cardarilli G. C. Residue Number System for low-power DSP applications / G. C. Cardarilli, A. Nannarelli, A. Re // In Proceedings 41st Asilomar Conference Signals, Systems and Computers. – 2007. – P. 1412–1416.
132. Madhukumar A. Enhanced architecture for Residue Number System-based CDMA for high-rate data transmission /

A. Madhukumar, F. Chin // IEEE Transactions Wireless Communication – 2004. – Vol. 3 (5). – P. 1363–1368.

133. Wei S. A novel residue arithmetic hardware algorithm using a Signed-Digit number representation / S. Wei, K. Shimizu // IEICE Transactions Information Systems. – 2000. – Vol. E83-D (12). – P. 2056–2064.
134. Persson A. Forward and reverse converters and moduli set selection in signed-digit Residue Number Systems / A. Persson, L. Bengtsson // Journal Signal Processing Systems – 2009. – Vol. 56 (1). – P. 1–15.
135. Сиора А. А. Отказоустойчивые системы с версионно-информационной избыточностью в АСУ ТП : моногр. / А. А. Сиора, В. А. Краснобаев, В. С. Харченко. – Х. : МОН, НАУ им. Н. Е. Жуковского (ХАИ), 2009. – 320 с.
136. Jaberipur G. Weighted two-valued digit-set encodings: unifying efficient hardware representation schemes for redundant number systems / G. Jaberipur, B. Parhami, M. Ghodsi // IEEE Transaction Circuits Systems – 2005. – Vol. 52 (7). – P. 1348–1357.
137. Краснобаев В. А. Расчет и сравнительный анализ производительности компьютерной системы обработки целочисленных данных, представленных в системе остаточных классов / В. А. Краснобаев, А. С. Янко, С. А. Кошман, В. Н. Курчанов, Ю. П. Бендес // Системи обробки інформації. – 2015. – В 3 (128). – С. 57–61.
138. Жихарев В. Я. Методы и средства обработки информации в непозиционной системе счисления в остаточных классах / В. Я. Жихарев, Я. В. Илюшко, Л. Г. Кравец, В. А. Краснобаев. – Житомир : Волынь, 2005. – 220 с.
139. Исупов К. С. Параллельные вычисления над много-разрядными числами в системе остаточных классов /

К. С. Исупов // Труды Международной суперкомпьютерной конференции (г. Новороссийск, 19–24 сентяб. 2011 г.). – Новороссийск, 2011. – С. 534–540.

140. Исупов К. С. Программный пакет высокоточных модулярно-позиционных вычислений с плавающей точкой / К. С. Исупов, В. С. Князьков // Advanced Science. – 2013. – № 3. – С. 150–171.
141. Исупов К. С. Система остаточных классов как инструмент для выполнения параллельных высокоточных численных расчетов / К. С. Исупов, В. С. Князьков // Математическое моделирование развивающейся экономики, экологии и биотехнологий (ЭКОМОД-2010) : труды V Всерос. науч. конф. (5–11 июля 2010 г.). – М., 2010. – С. 79–88.
142. Исупов К. С. Инструментальный комплекс для проектирования параллельных масштабируемых программ численных расчетов / К. С. Исупов, В. С. Князьков // Научно-технический вестник СПбГУ. – 2010. – № 6 (70). – С. 68–72.
143. Abdullah M. A systematic approach for selecting practical moduli sets for residue number systems / M. Abdullah, A. Skavantzos // Proceedings on 27th IEEE International Symposium System Theory. – 1995. – P. 445–449.
144. Wang W. M. Moduli selection in RNS for efficient VLSI implementation / W. M. Wang, N. S. Swamy, M. O. Ahmad // Proceedings IEEE International Symposium Circuits Systems. – 2003. – V. 4. – P. 512–515.
145. Liu Y. Moduli Set Selection and Cost Estimation for RNS-Based FIR Filter and Filter Bank Design / Y. Liu, E. Lai // Design Automation for Embedded Systems. – 2004. – V. 9. – P. 123–139.

146. Cao B. An efficient reverse converter for the 4-moduli set {2^n-1, 2^n, 2^n+1, 2^{2n}+1} based on the new Chinese Remainder Theorem / B. Cao, C.-H. Chang, T. Srikanthan // IEEE Transactions Circuits Systems – 2003. – Vol. 50(10). – P. 1296–1303.
147. Molahosseini A. Efficient reverse converter designs for the new 4-moduli sets {2^n–1, 2^n, 2^n+1, 2^{2n+1}–1} and {2^n–1, 2^n+1, 2^{2n}, 2^{2n}+1} based on New CRTs / A. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, S. Timarchi // IEEE Transactions Circuits Systems – 2010. – Vol. 57 (4). – P. 823–835.
148. Gbolagade K. An improved RNS reverse converter for the {2^{2n+1}–1, 2^n, 2^n–1} moduli set / K. Gbolagade, R. Chaves, L. Sousa and, S. Cotofana // In Proceedings International Symposium Circuits Systems – 2010. – P. 2103–2106.
149. Kalampoukas L. High-speed parallel-prefix modulo 2^n–1 adders / L. Kalampoukas, D. Nikolos, C. Efstathiou, H. T. Vergos, J. Kalamatianos // IEEE Transactions on Computers. – 2000. – V. 49 (7). – P. 673–679.
150. Efstathiou C. Fast parallel-prefix modulo 2^n+l adder / C. Efstathiou, H. T. Vergos, D. Nikolos // IEEE Transactions on Computers. – 2004. – V. 53 (9). – P. 1211–1216.
151. Patel R A. Novel power-delay-area-efficient approach to generic modular addition / R. A. Patel, M. Benaissa, N. Powell, S. Boussakta // IEEE Transactions on Circuits and Systems. – 2007. – V. 54. – P. 1279–1292.
152. Hiasat A. A. High-speed and reduced area modular adder structures for RNS / A. A. Hiasat // IEEE Transactions on Computers. – 2002. – V. 51. – P. 84–89.
153. Zarandi A. A. E. Modern Residue Number System Moduli Set: Efficiency vs. Complexity / A. A. E. Zarandi,

A. S. Molahosseini, M. Hosseinzadeh // Нейрокомпьютеры: разработка, применение. – 2014. – № 9. – С. 7–12.

154. Szabo N. Residue Arithmetic and its Applications to Computer Technology / N. Szabo, R. Tanaka. – New York : McGraw-Hill, 1967. – 319 p.
155. Premkumar B. An RNS to binary converter in 2n+1, 2n, 2n–1 moduli set / B. Premkumar // IEEE Transactions on Circuits and Systems. – 1992. – V. 39. – P. 480–482.
156. Pourbigharaz F. A signed-digit architecture for residue to binary transformation / F. Pourbigharaz, H. M. Yassine // IEEE Transactions on Computers. – 1997. – V. 46. – P. 1146–1150.
157. Hiasat A. A. Residue-to-binary arithmetic converter for the moduli set (2^k, 2^k–1, 2^{k-1}-1) / A. A. Hiasat, H. S. Abdel-Aty-Zohdy // IEEE Transactions on Circuits and Systems. – 1998. – V. 45. – P. 204–208.
158. Mathew J. Fast residue-to-binary converter architectures / J. Mathew, D. Radhakrishnan, T. Srikanthan // In Proceedings of IEEE International Midwest Symposium on Circuits and Systems. – 1999. – P. 1090–1093.
159. Molahosseini A. S. A new residue to binary converter based on mixed-radix conversion / A. S. Molahosseini, K. Navi, M. K. Rafsanjani // In Proceedings of IEEE International Conference on Information and Communication Technologies: From Theory to Applications. – 2008. – P. 394–399.
160. Hariri A. A new high dynamic range moduli set with efficient reverse converter / A. Hariri, K. Navi, R. Rastegar // Elsevier Journal of Computers and Mathematics with Applications. – 2008. – V. 55 (4). – P. 660–668.
161. Molahosseini A. S. An efficient architecture for designing reverse converters based on a general three-moduli set / A. S. Molahosseini, K. Navi, O. Hashemipour, A. Jalali //

Elsevier Journal of Systems Architecture. – 2008. – V. 54. – P. 929–934.

162. Bhardwaj M. A reverse converter for the 4-moduli superset $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ / M. Bhardwaj, T. Srikanthan, C. T. Clarke // In Proceedings of IEEE Symposium on Computer Arithmetic. – 1999. – P. 316–321.
163. Vinod A. P. A residue to binary converter for the 4-moduli superset $\{2^n–1, 2^n, 2^n+1, 2^{n+1}+1\}$ / A. P. Vinod, A. B. Premkumar // Journal of Circuits, Systems and Computers. – 2000. – V. 10. – P. 85–99.
164. Sheu M. H. An Efficient VLSI Design for a Residue to Binary Converter for General Balance Moduli $\{2^n–1, 2^n+1, 2^n–3, 2^n+3\}$ / M. H. Sheu, S. H. Lin, C. Chen, S. W. Yang // IEEE Transactions on Circuits and Systems. – 2004. – V. 51 (3). – P. 152–155.
165. Zhang W. An efficient design of residue to binary converter for four moduli set $\{2^n–1, 2^n+1, 2^{2n}–2, 2^{2n+1}–3\}$ based on new CRT II / W. Zhang, P. Siy // Elsevier Journal of Information Sciences. – 2008. – V. 178 (1). – P. 264–279.
166. Molahosseini A.S. A Reverse Converter for the Enhanced Moduli Set $\{2^n–1, 2^n+1, 2^{2n}, 2^{2n+1}–1\}$ Using CRT and MRC / A. S. Molahosseini, K. Vovr // In Proceedings of IEEE Computer Society Annual Symposium on VLSI. – 2010. – P. 456 – 457.
167. Patronik P. Design of Reverse Converters for General RNS Moduli Sets $\{2^k, 2^n–1, 2^n+1, 2^{n-1}–1\}$ / P. Patronik, S. J. Piestrak // IEEE Transactions on Circuits and Systems. – 2014. – P. 143–148.
168. Hiasat A. A. VLSI implementation of new arithmetic residue to binary decoders / A. A. Hiasat // IEEE Transactions on Very Large Scale Integration Systems. – 2005. – V. 13 (1). – P. 153–158.

169. Cao B. A residue-to-binary converter for a new five-moduli set / B. Cao, C.-H. Chang, T. Srikanthan // IEEE Transactions on Circuits and Systems I. – 2007. – V. 54 (5). – P. 1041–1049.
170. Molahosseini A. S. A new five-moduli set for efficient hardware implementation of the reverse converter / A. S. Molahosseini, C. Dadkhah, K. Navi // IEICE Electronics Express. – 2009. – V. 6 (4). – P. 1006–1012.
171. Pettenghi K. RNS Reverse Converters for Moduli Sets With Dynamic Ranges up to (8n+l)-bit / K. Pettenghi, R. Chaves, L. Sousa // IEEE Transactions on Circuits and Systems. – 2013. – V. 60 (6). – P. 1487–1500.
172. Pettenghi H. Method to Design General RNS Reverse Converters for Extended Moduli Sets / H. Pettenghi, R. Chaves, L. Sousa // IEEE Transactions on Circuits and Systems. – 2013. – V. 60 (12). – P. 877–881.
173. Parhami B. On Equivalences and Fair Comparisons among Residue Number Systems with Special Moduli / B. Parhami // In Proceedings of 44th Asilomar Conference Signals, Systems, and Computers. – 2010. – P. 1690–1694.
174. Zarandi A. A. E. Reverse Converter Design via Parallel-Prefix Adders: Novel Components, Methodology and Implementations / A. A. E. Zarandi, A. S. Molahosseini, M. Hosseinzadeh, S. Sorouri, S. Antao, L. Sousa // IEEE Transactions on Very Large Scale Integration Systems. – 2014. – P. 834–838.
175. Chokshi R. Exploiting residue number system for power-efficient digital signal processing in embedded processors / R. Chokshi, K.S. Berezowski, A. Shrivastava, S. J. Piestrak // Proceedings of the 2009 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES ‘09), Grenoble, France. – 2009. – P. 19–28.

176. Piestrak S. Design of residue multipliers-accumulators using periodicity / S. Piestrak, K. Berezowski // Proceedings of the IET Irish Signals and Systems Conference (ISSC 2008), Galway, Republic of Ireland. – 2008. – P. 380–385.
177. Piestrak S. J. Architecture of efficient RNS-based digital signal processor with very low-level pipelining / S. Piestrak, K. Berezowski // Proceedings of the IET Irish Signals and Systems Conference (ISSC 2008), Galway, Republic of Ireland. – 2008. – P. 127–132.
178. Wnuk M. Remarks on hardware implementation of image processing algorithms / M. Wnuk // International Journal of Applied Mathematics and Computer Science. – 2008. – Vol. 18(1). – P. 105–110.
179. Wang W. RNS application for digital image processing / W.Wang, M. N. S. Swamy, M. O. Ahmad // Proceedings of the 4th IEEE International Workshop on System-on-Chip for Real-Time Applications (IWSOC'04), Banff, Alberta, Canada. – 2004. – P. 77–80.
180. Conway R. Improved RNS FIR filter architectures / R. Conway, J. Nelson // IEEE Transactions on Circuits and Systems. – 2004. – Vol. 51(1). – P. 26–28.
181. Mohan A. P .V. Residue Number Systems: Algorithms and Architectures / A. P. V. Mohan. – Kluwer Academic Publishers, Norwell, MA, 2002. – 254 p.
182. Soderstrand M. A. Residue Number System Arithmetic: Modern Applications in Digital Signal Processing / M. A. Soderstrand, W. K. Jenkins, G. A. Jullien, F. J. Taylor. – IEEE Press, Piscataway, NJ, 1986. – 382 p.
183. Yassine H. M. Hierarchical residue numbering system suitable for VLSI arithmetic architectures / H. M. Yassine // Proceedings of the IEEE International Symposium on Circuits

and Systems (ISCAS ‘92), San Diego, CA, USA. – 1992. – P. 811–814.

184. Skavantzos A. Implementation issues of the two-level residue number system with pairs of conjugate moduli / A. Skavantzos, M. Abdallah // IEEE Transactions on Signal Processing. – 1999. – Vol. 47(3). – P. 826–838.
185. Bajard J. Multi-fault attack detection for RNS cryptographic architecture / J. Bajard, J. Eynard, N. Merkiche // In 23nd IEEE Symposium on Computer Arithmetic (ARITH 2016), Silicon Valley, CA, USA. – 2016. – P. 16–23.
186. Perin G. Electromagnetic analysis on RSA algorithm based on RNS / G. Perin, L. Imbert // In 2013 Euromicro Conference on Digital System Design (DSD). – 2013. – Vol. 1. – P. 345–352.
187. Schinianakis D. Hardware-fault attack handling in RNS-based Montgomery multipliers / D. Schinianakis, T. Stouraitis // In 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013). – 2013. – P. 3042–3045.
188. Fournaris A. P. Secure and Efficient RNS Approach for Elliptic Curve Cryptography / A. P. Fournaris, L. Papachristodoulou, L. Batina, N. Sklavos // Proc. Of the 6th Conference on Trustworthy Manufacturing and Utilization of Secure Devices (TRUDEVICE 2016), Barcelona. – 2016. – P. 121–126.
189. Fournaris A. P. Fault and power analysis attack resistant RNS based edwards curve point multiplication / A. P. Fournaris, N. Klaoudatos, N. Sklavos, C. Koulamas // In Proceedings of the 2nd Workshop on Cryptography and Security in Computing Systems, Amsterdam, Netherlands. – 2015. – P. 43–46.
190. Fournaris A. P. Residue number system as a side channel and fault injection attack counter measure in elliptic curve cryptography / A. P. Fournaris, L. Papachristodoulou, L. Batina, N. Sklavos // In 2016 International Conference on

Design and Technology of Integrated Systems in Nanoscale Era. – 2016. – P. 1–4.

191. Guillermin N. A high speed coprocessor for elliptic curve scalar multiplications over Fp / N. Guillermin // Lecture Notes in Computer Science Advances in Cryptology, Cryptographic Hardware and Embedded Systems. – 2010. – P. 48–64.
192. Fadulilahi I. R. Efficient Algorithm for RNS Implementation of RSA / I. R. Fadulilahi, E.K. Bankas, J.B.A.K. Ansuura // International Journal of Computer Applications. – 2015. – Vol. 127 (5). – P. 14–19.
193. Laurent I. A Full RNS Implementation of RSA / I. Laurent, B. Jean-Claude // Transactions on computers. – 2004. – Vol. 53 (5). – P. 1-6.
194. Nobuhiro T. Design of High-Speed RSA Encryption Processor Based on the Residue Table for Redundant Binary Numbers / T. Nobuhiro, I. Teruki // Systems and Computers in Japan. – 2002. – Vol. 33(5). – P. 423–432.
195. Krasnobayev V. A. Method of realization of cryptographic RSA transformations on the basis of application of modular number system / V. A. Krasnobayev, S. A. Koshman // Biomedical Soft Computing and Human Sciences. – 2011. – Vol. 17 (2). – P. 31–36.
196. Краснобаев В. А Метод быстрой реализации криптографических преобразований на основе поразрядной табличной реализации / В. А. Краснобаев, С.А. Кошман // Системи обробки інформації. – 2009. – № 7 (79). – С. 63–68.
197. Краснобаев В. А Метод быстрой обработки криптографической информации в модулярной системе счисления / В. А. Краснобаев, С. А. Кошман, С. В. Сомов, Е. А. Крючко // Системи обробки інформації. – 2013. – № 6 (113). – С. 194–198.

198. Краснобаев В. А/ Метод контроля криптографической информации, представленной в модулярной системе счисления / В. А. Краснобаев, С. А. Кошман, В. Н. Курчанов, А. В. Гарамась // Збірник наукових праць Харківського університету Повітряних сил ім. І. Кожедуба. – 2013. – Вип.3 (36). – С. 104–107.
199. Krasnobayev V. A. The model and the method of implementation of integer arithmetic operations within the RSA crypto algorithms / V. A. Krasnobayev, O. I. Tyrtyshnikov, I. I. Sliusar, V. N. Kurchanov, S. A. Koshman // Системи обробки інформації. – 2014. – № 1 (117). – С. 117–122.
200. Krasnobayev V. A. Mathematical model and tabular method implementation of modular arithmetic operations with crypto transformations in the residue class / V. A. Krasnobayev, O. I. Tyrtyshnikov, S. V. Somov, S. A. Koshman, G. V. Sokol, N. V. Rvachova // Системи обробки інформації. – 2014. – № 2 (118). – С. 119–123.
201. Волинський О. І. Методи міжбазисних перетворень на основі розмежованої системи числення залишкових класів / О. І. Волинський // Вісник національного університету Львівська політехніка, Комп'ютерні системи та мережі. – 2010. – № 688. – С. 53–59.
202. Касянчук М. М. Аналітичний пошук модулів досконалої форми системи залишкових класів та їх застосування в китайській теоремі про залишки / М. М. Касянчук, І. З. Якименко, І. Р. Паздрій, Я. М. Николайчук // Вісник Хмельницького національного університету. Технічні науки. – 2015. – № 1(221). — С. 170–176.
203. Касянчук М. М. Концепція теоретичних положень досконалої форми перетворення Крестенсона та його практичне застосування / М. М. Касянчук // Оптико-

електронні інформаційно-енергетичні технології. – 2010. – № 2 (20). – С. 43–47.

204. Kasianchuk M. Algorithms of findings of perfect shape modules of remaining classes system / M. Kasianchuk, I. Yakymenko, I. Pazdriy, O. Zastavnyy // XIII International Conference «The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-2015)», 23–25 February, 2015, Polyana-Svalyava (Zakarpattya), Ukraine. – P. 168–171.
205. Касянчук М. М. Метод знаходження модулів досконалої форми системи залишкових класів / М. М. Касянчук, І. З. Якименко, А. Я. Давлетова, Т. М. Долинюк, Н. А. Рендзеняк // Теорія прийняття рішень : праці VII Міжнар. школи-семінару (м. Ужгород, 29 верес. – 4 жовт. 2014 р.). – Ужгород, 2014. – С. 122–123.
206. Касянчук М. М. Алгоритми підбору модулів у системі залишкових класів / М. М. Касянчук, Р. П. Сидорчук // Сучасні комп'ютерні інформаційні технології (ACIT–2011) : матеріали І Всеукр. школи-семінару молодих вчен. і студ. (м. Тернопіль, 20–21 трав. 2011 р.). – Тернопіль, 2011. – С. 50–51.
207. Kasianchuk M. Conception of theoretical bases of the accomplished form of Krestenson's transformation and its practical application / M. Kasianchuk // Proceedings of the 4–th International Conference «Advanced Computer Systems and Network: Design and Application» (ACSN–2009).–L'viv, 2009. – P. 299–301.
208. Касянчук М. М. Теорія та математичні закономірності досконалої форми системи залишкових класів / М. М. Касянчук // Питання оптимізації обчислень (ПОО–XXXV) :

праці Міжнар. симпозіуму. Т.1. – К. – Кацивелі, 2009. – С. 306–310.

209. Касянчук М. М. Теорія перетворення досконалої форми системи залишкових класів базису Крестенсона / М. М. Касянчук // Інформаційні проблеми комп'ютерних систем, юриспруденції, економіки та моделювання (ПНМК–2009) : матеріали Проблемно-наук. міжгалуз. конф. – Тернопіль – Бучач, 2009. – С. 133–137.
210. Касянчук М. Алгоритми побудови модифікованої досконалої форми системи залишкових класів / М. Касянчук // Спеціалізовані комп'ютерні технології в інформатиці. – Тернопіль : Бескиди, 2017. – С. 580–604.
211. Nykolaychuk Ya. M. Theoretical Foundations of the Modified Perfect form of Residue Number System / Ya. M. Nykolaychuk, M. M. Kasianchuk, I. Z. Yakymenko // Cybernetics and Systems Analysis. –2016. – Vol. 52, is. 2. – P. 219–223.
212. Kasianchuk M. N. Theory and Methods of Constructing of Modules System of the Perfect Modified Form of the System of Residual Classes / M. N. Kasianchuk, Ya. N. Nykolaychuk, I. Z. Yakymenko // Journal of Automation and Information Sciences. – 2016. – Vol. 48, № 8. – P. 56–63.
213. Касянчук М. Построение модифицированной совершенной формы системы остаточных классов с использованием факторизации / М. Касянчук // Радиоэлектроника, информатика, управление. – 2017. – Вид. 42, № 3. – С. 53–59.
214. Касянчук М. М. Побудова модифікованої досконалої форми системи залишкових класів на основі розв'язку систем конгруенцій / М. М. Касянчук // Науковий збірник Національного лісотехнічного університету України. – 2016. – Т. 26, № 7. – С. 372–377.

215. Касянчук М. М. Побудова трьохмодульної модифікованої досконалої форми системи залишкових класів на основі розв'язку квадратного рівняння / М. М. Касянчук // Інформатика та математичні методи в моделюванні. – 2016. – Т. 6, № 1. – С. 19–25.
216. Николайчук Я. М. Алгоритм знаходження системи модулів модифікованої досконалої форми системи залишкових класів / Я. М. Николайчук, М. М. Касянчук, І. З. Якименко, Л. М. Тимошенко, Т. М. Долинюк // Сучасні інформаційні та електронні технології : матеріали Міжнар. наук.-практ. конф. (Одеса, 26 – 30 трав. 2014 р.). – С. 115–116.
217. Nykolaychuk Ya. M. Theoretical Foundations for the Analytical Computation of Coefficients of Basic Numbers of Krestenson's Transformation / Ya. M. Nykolaychuk, M. M. Kasianchuk, I. Z. Yakymenko // Cybernetics and Systems Analysis. – 2014. – Vol. 50 (5). – P. 649–654.
218. Касянчук М. М. Дослідження часових характеристик апаратної реалізації методів пошуку оберненого елемента за модулем / М. М. Касянчук, І. З. Якименко, С. В. Івасьєв, Н. М. Мандебура, В. М. Неміш // Вісник Хмельницького національного університету. Технічні науки. –2017. – № 6 (255). – С. 191–197.
219. Касянчук М. М. Експериментальне дослідження програмної реалізації методів пошуку оберненого елемента за модулем / М. М. Касянчук, І. З. Якименко, С. В. Івасьєв, О. В. Момотюк // Інформатика та математичні методи в моделюванні. – 2017. – Т. 7, № 3. – С. 178–186.
220. Rajba T. Research of Time Characteristics of Search Methods of Inverse Element by the Module / T. Rajba, A. Klos-Witkowska, S. Ivasiev, I. Yakymenko, M. Kasianchuk // Proceedings of the 2017 IEEE 9th International Conference on

Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS–2017) – Bucharest, Romania. 2017. – V. 1, September. – P. 82–85.

221. Stakhov A. The Generalized Principle of the Golden Section and its Applications in Mathematics, Science and Engineering / A. Stakhov // Chaos, Solitons & Fractals. – 2005. – V. 26 (2). – P. 263–289.
222. Kasianchuk M. High- Theoretical foundations for creating five modular modified perfect form of the system of residual classes / M. Kasianchuk, I. Yakymenko, S. Ivasiev // Projekt Interdyscyplinarny projektem XXI wieku. Monografia. – Bielsko-Biała : Wydawnictwo Naukowe Akademii Techniczno-Humanistycznej w Bielsku-Białej, 2017. – T. 2. – P. 123–130. – Chapter in monograph.
223. Касянчук М. М. Метод розширення набору модулів модифікованої досконалої форми системи залишкових класів / М. М. Касянчук, І. З. Якименко, С. В. Івасьєв, Б. О. Маслияк // Математичне та комп'ютерне моделювання. Технічні науки. – 2017. – В.15. – С. 73–78.
224. Касянчук М. М. Модифікований метод шифрування Рабіна з використанням різних форм системи залишкових класів / М. М. Касянчук, І. З. Якименко, Л. О. Дубчак, Н. А. Рендзеняк, Н. М. Мандебура // Вісник Хмельницького національного університету. Технічні науки. – 2017.№ – 1(245). – С. 127–131.
225. Kasianchuk M. Rabin's modified method of encryption using various forms of system of residual classes / M. Kasianchuk, I. Yakymenko, I. Pazdriy, A. Melnyk, S. Ivasiev // XIV International Conference «The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM-

2017)». 21–25 February, 2017, Polyana-Svalyava. – P. 222–224.

226. Івасьєв С. В. Вдосконалений алгоритм пошуку символів Якобі / С. В. Івасьєв, І. З. Якименко, М. М. Касянчук // Оптико-електронні інформаційно-енергетичні технології. – 2015. – Т. 29, № 1. – С. 45–50.
227. Кучерук І. М. Загальний курс фізики / І. М. Кучерук, І. Т. Горбачук, П. П. Луцик. – К.: Техніка, 2001. – 452 с.
228. Заставний О. М. Методологія побудови автономних сенсорів для розподілених комп'ютерних мереж / О. М. Заставний, Я. М. Николайчук // Вісник Технологічного університету Поділля. – 2002. – Т.1, № 3. – С. 142–146.
229. Iakymenko I. Construction of distributed thermal or piezoelectric sensor based on residue systems / I. Iakymenko, M. Kasianchuk, I. Kinakh, M. Karpinski // Przeglad Elektrotechniczny. – 2017. – № 1. – P. 290–294.
230. Касянчук М. М. Метод побудови розподіленого температурного сенсора на основі системи числення базису Крестенсона / М. М. Касянчук, М. І. Чирка, Я. М. Николайчук, І. З. Якименко // «Інформаційні проблеми комп'ютерних систем, юриспруденції, енергетики, економіки, моделювання та управління ПНМК-2011 : матеріали Проблемно-наук. міжгалуз. конф. (Бучач, Яремча, Карпати, 17–20 трав. 2011 р.). – Бучач – Яремча, 2011. – С. 143–149.
231. Касянчук М. Н. Экспериментальное исследование программной реализации системы остаточных классов и ее модифицированной совершенной формы / М. Н. Касянчук // Вестник Брестского государственного технического университета. Физика, математика, информатика. – 2017. – № 5 (97). – С. 53–57.

232. Stewart J. Python for Scientists / J. Stewart. – Cambridge : Cambridge University Press, 2014. – 230 p.
233. Касянчук М. М. Експериментальне дослідження програмної реалізації методів модулярного експоненціювання / М. М. Касянчук, І. З. Якименко, Т. М. Долинюк, Н. А. Рендзеняк // Інформатика та математичні методи в моделюванні. – 2015. – Т. 5, № 4. – С. 376–382.
234. Задірака В. К. Комп'ютерна арифметика багаторозрядних чисел / В. К. Задірака, О. С. Олексюк. – К., 2003. – 264 с.

Наукове видання

Михайло Миколайович Касянчук

ДОСКОНАЛА ФОРМА СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ: МЕТОДИ ПОБУДОВИ ТА ЗАСТОСУВАННЯ

Монографія

Комп'ютерна верстка Ольги Слимак
Дизайн обкладинки Марії Одобецької

Підписано до друку 26. 04. 2019 р.
Формат 60х84 $^{1}/_{16}$. Гарнітура Times.
Папір офсетний. Друк на дублікаторі.
Умов. друк. арк. 12. Облік.-вид. арк. 11
Зам. № М 019-19. Тираж 300 прим.

Видавець та виготовлювач
Тернопільський національний економічний університет
вул. Львівська, 11, м. Тернопіль, 46004

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців ДК № 3467 від 23.04.2009 р.

Видавничо-поліграфічному центр «Економічна думка» ТНЕУ
вул. Бережанська 2 м. Тернопіль 46004
тел. (0352) 47-58-72
E-mail: edition@tneu.edu.ua