Шифр «Швидкість реалізації»

СТУДЕНТСЬКА НАУКОВА РОБОТА

на тему:

«Методи та засоби швидкої обробки даних комп’ютерних систем, представлених в системі залишкових класів»

2019 рік

АНОТАЦІЯ

Актуальність використання непозиційної системи числення, а саме системи залишкових класів (СЗК), як одного із способів підвищення швидкодії арифметичних операцій, а в результаті – можливості створення продуктивних та надійних комп'ютерних систем (КС) реального часу, не викликає сумніву. СЗК володіє цінною властивістю незалежності один від одного залишків за прийнятою системі основ. Ця незалежність відкриває широкі можливості для побудови не тільки нової машинної арифметики, а й принципово нової схемної реалізації КС, яка в свою чергу помітно розширює застосування машинної арифметики.

Об'єкт дослідження — процеси обробки даних у системі залишкових класів.

Предмет дослідження — комп'ютерні системи та засоби швидкої обробки даних, що представлені системі залишкових класів.

Метою дослідження є підвищення швидкості обробки даних комп'ютерних систем, що функціонують в системі залишкових класів, з урахуванням кількості необхідного обладнання.

Метод дослідження — теорія надійності та теорія завадостійкого кодування у класі лишків.

Завдання — провести дослідження впливу основних властивостей СЗК на структуру комп'ютерних систем, розглянути принципи реалізації арифметичних операцій у СЗК, та розробити методи швидкої реалізації модульних арифметичних операцій.

Студентська наукова робота складається з вступу, трьох розділів, висновку, списку використаної літератури та одного додатку. Загальний обсяг роботи становить 32 сторінки формату А4 без урахування додатку. Наукова робота містить 17 формул та 8 рисунків. Використано 14 посилань на наукові джерела.

Ключові слова: КОМП'ЮТЕРНА СИСТЕМА, МОДУЛЬНА АРИФМЕТИЧНА ОПЕРАЦІЯ, ОБРОБКА ДАНИХ, СИСТЕМА ЗАЛИШКОВИХ КЛАСІВ, СИСТЕМА ЧИСЛЕННЯ, ШВИДКІСТЬ РЕАЛІЗАЦІЇ.

ЗМІСТ

ВСТУП

Інформаційні технології сучасного світу дуже стрімко розвиваються. Неможливо уявити себе без техніки, яка стала неодмінною часткою нашого життя. Входячи з цього факту інформація стала вельми важливим ресурсом. Уїнстон Черчіль сказав, що «хто володіє інформацією, той володіє світом» і, як можна впевнитись, він мав рацію. З розвитком суспільства інформації стає дедалі більше, тому зростає увага до інформаційних технологій.

Велика сукупність інформації потребує багато чого: захисту зберігання, обробки тощо. Через це було створено багато галузей науки, безліч наукових відкриттів. Проте щороку об’єми інформації продовжують збільшуватися, а тому питання швидкої обробки інформації змушує шукати нові шляхи її вдосконалення.

У сучасних комп’ютерних системах (КС) дії проводяться над числами, представленими у вигляді спеціальних машинних кодів у прийнятій системі числення. Залежно від способу зображення чисел за допомогою цифр, існуючі системи числення умовно ділять на: позиційні, непозиційні та змішані системи. Система числення – сукупність знаків та правил, за допомогою яких можна письмово відобразити будь-яке число [1].

У даній роботі приділяється увага системі залишкових класів (СЗК), яка є непозиційною системою числення. Справа у тому, що позиційні системи числення (ПСЧ) не здатні цілком впоратися з сучасними задачами, серед яких швидкісна обробка даних з високою точністю та надійністю та обробка даних у реальному часі фактично паралельно під час виконання обчислювального процесу.

У свою чергу СЗК відкриває нові можливості у цьому напрямку, оскільки, можливо розпаралелити процес обробки цілочислових даних та забезпечити високу надійність отриманих результатів.

Ще у кінці 50-х років минулого сторіччя радянські вчені (одним з перших був Федір Вікторович Лукін) звернули увагу на СЗК [2]. Першим, хто вважав

доцільним застосування СЗК у обчислювальній техніці були математик А. Свобода та чехословацький інженер М. Валах. Ці вчені зацікавили американців та згодом переїхали туди, де і далі проводили свої праці. Згодом радянські вчені про це дізналися, а Лукін відразу зрозумів перспективність досліджень у цьому напрямі. Працюючи там, де не займалися розробкою комп'ютерних систем та компонент швидкої обробки цілочислових даних (КС), він ознайомив з працею відповідних вчених І. Я. Акушського, Д. І. Юдицького, яких зацікавив даний напрямок, і які фактично першими в СРСР почали працювати в СЗК.

Основним досягненням радянських вчених стала розробка першої робочої модулярної КС Т-340А, яка біла створена для радіолокаційної станції (РЛС) «Дунай-3УП» системи протиракетної оборони А-35. Принципи побудови було запропоновано вченими І. Я. Акушським, Д. І. Юдицьким та Е. С. Андріановим, а сама робота була налагоджена та реально експлуатувалася ще багато років у РЛС. Налагодилося серійне виробництво і К-340А стала базовою для усіх РЛС на той час. Це пояснюється тим, що вперше було організовано принцип незалежних каналів пам'яті, а оперативна пам'ять складалася з 16 блоків, кожний по 1К слів. Кожен блок мав два порти для обміну інформацією: з процесором та абонентами. Прискорювало швидкодію розмежування (почергове) звернень від процесора до блоків. Ще однією особливістю була реалізація двох операційних команд (замість 4), кожна з яких містила у собі по дві команди. Зазначені КС Т-340А та К-340А мали швидкодію 1,2 млн подвійних (2,4 звичайних) операцій у секунду, що на той час вважалося практично неможливим (для порівняння типова швидкодія того часу – десятки, максимум сотні тисяч операцій у секунду). Фактично радянські вчені створили першу у світі КС з швидкодією понад 1 млн оп/с та вартістю одиниці продуктивності – 25 коп. за операцію в секунду. Крім того, КС володіла високою надійністю, через що КС К-340А знаходилася в експлуатації майже 50 років, при цьому не поступаючись сучасним електронним системам.

Проте потім настав час, пов'язаний із занепадом СЗК в СРСР. Почалося все з того, що в США почали цікавитися розробками радянських вчених, які

були у відкритому доступі (наприклад, 5Э53), а радянська влада вирішила припинити будь-які спроби контактів із закордонними країнами. Відповідна реакція була згодом і у США, які у свою чергу також засекретили розробки у цьому напрямі. Це стало поштовхом до занепаду інтересу в напрямку розробок в СЗК, і лише окремі ентузіасти продовжили займатися цим питанням.

Станом на сьогодні відновлено науковий інтерес до СЗК, з'явилося багато публікацій та патентоспроможних пристроїв на які отримано патенти України. Основною характеристикою СЗК, яка привернула до себе увагу стала принципова відмінність від інших ПСЧ, де обробка результату виконується порозрядно і фактично неможливо розпаралелювання цього процесу. У той самий час СЗК завдяки трьом основним властивостям, а саме незалежності, рівноправності та малорозрядності лишків дає змогу швидко та ефективно організовувати процес обчислення результату модульної операції. Але і це ще не все, адже, також на відміну від ПСЧ, у СЗК можливо завдяки введенню додаткових (контрольних) основ отримати коригуючі властивості. Арифметичність коду СЗК дозволяє знайти помилку і більше того – виправити її шляхом введення штучної збитковості. Тобто помилки, які виникають під час обчислень у ПСЧ не знаходяться і не виправляються, навіть можуть розмножитися до інших розрядів, у той час коли в СЗК помилки залишаються кожна у своєму розряді і на загальну швидкодію чи достовірність вони матимуть значно менший вплив, ніж у ПСЧ.

Актуальність використання непозиційної системи числення, а саме системи залишкових класів (СЗК), як одного із способів підвищення швидкодії арифметичних операцій, а в результаті – можливості створення продуктивних та надійних комп'ютерних систем (КС) реального часу, не викликає сумніву. СЗК володіє цінною властивістю незалежності один від одного залишків за прийнятою системі основ. Ця незалежність відкриває широкі можливості для побудови не тільки нової машинної арифметики, а й принципово нової схемної реалізації КС, яка в свою чергу помітно розширює застосування машинної арифметики.

РОЗДІЛ1.
ДОСЛІДЖЕННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РЕАЛІЗАЦІЇ МОДУЛЬНИХ АРИФМЕТИЧНИХ ОПЕРАЦІЙ

1.1. Огляд основних властивостей СЗК

Як відомо, системою числення (СЧ) називають спосіб позначення чисел за допомогою символів. Різні символи (у цьому випадку це числа) мають певну вагу, через що можливо описати значення усього числа. А залежно від форми запису числа системи числення поділяють на позиційні та непозиційні.

Позиційні СЧ характеризуються тим, що в них від розташування (позиції) кожного символу залежить значення всього числа. Прикладом такої системи з основою q = 10 є арабська система числення. Позначимо за A саме число, а за a_i цифри (символи) які є в його записі, тоді число має вигляд:

$$A=(a_{z-1},a_{z-2},...,a_1,a_0), \tag{1.1}$$

де z – розрядність операндів числа, а $0 \le a_i \le q-1$.

Непозиційною СЧ називається така, в якій символи мають фіксовану вагу і вона не залежить від його розташування в числі. Як приклад можна навести римську СЧ, де число може бути записане, відповідно, у вигляді $A = XI$ (10 – 1 = 9) [1].

Дещо детальніше слід розглянути СЗК та її властивості. Введемо поняття класу еквівалентності (1.2) – множина усіх натуральних чисел, що складається з елементів, еквівалентних a_i. Припустимо що у нас є $m_i = 5$, тоді при знаходженні залишку будь-якого натурального числа за модулем $m_i = 5$ можна отримати наступні остачі: 0, 1, 2, 3, 4. Сам клас еквівалентності позначається як $\overline{a_i}$ і для заданого модулю $m_i = 5$ буде 5 таких класів:

$$\begin{aligned}
&\bar{0}\,\{..., -10, -5, 0, 5, 10, ...\} \\
&\bar{1}\,\{..., -9, -4, 1, 6, 11, ...\} \\
&\bar{2}\,\{..., -8, -3, 2, 7, 12, ...\} \\
&\bar{3}\,\{..., -7, -2, 3, 8, 13, ...\} \\
&\bar{4}\,\{..., -6, -1, 4, 9, 14, ...\}
\end{aligned} \qquad (1.2)$$

Це позначення класів еквівалентності в загальному вигляді. Можливо також позначити тільки невід'ємні класи еквівалентності, а якщо взяти з кожного класу по одній еквівалентності, то можна отримати три варіанти систем еквівалентності за модулем $m_i = 5$:

1. 0, 1, 2, 3, 4 – повна система найменших невід'ємних еквівалентностей
2. 0, 1, 2, –2, –1 – повна система найменших за абсолютно величиною (модулем) еквівалентностей
3. 5, 1, 2, 3, 4 – повна система найменших додатних еквівалентностей.

Відповідно до цього можна сформувати саме поняття системи залишкових класів (СЗК), де задано набір попарно взаємно простих чисел $m_1, m_2, ..., m_n$. Очевидно що будь-які 2 з них повинні мати найбільший спільний дільник одиницю. Таким чином, число у СЗК може бути представлено шляхом формування еквівалентностей (лишків) $\{a_i\}$ і виконувались наступні еквівалентності:

$$\begin{aligned}
A &= a_1 (\mathrm{mod}\, m_1), \\
A &= a_2 (\mathrm{mod}\, m_2), \\
&\vdots \\
A &= a_n (\mathrm{mod}\, m_n),
\end{aligned} \qquad (1,3)$$

Слід зауважити, що вибір лишків a_i можуть однозначно визначати число лише в інтервалі $[0, M)$, де $M = \prod_{i=1}^{n} m_i$, отже $A \geq M$ не підходить для однозначного визначення числа у СЗК. Відповідно до цього число A у СЗК може бути записане у вигляді (фактично – сукупність лишків):

$$A_{СЗК} = (a_1, a_2, ..., a_n). \qquad (1.4)$$

Варто буде також розглянути перевід числа із СЗК у ПСЧ. Один з методів полягає у тому, що знаходяться константи $B_i = \overline{m_i}M / m_i$, де m_i визначається з порівняння $\overline{m_i}M / m_i \equiv 1(\mathrm{mod}\, m_i)$ і далі визначається саме число у ПСЧ наступним шляхом:

$$A_{ПСЧ} = \left(\sum_{i=1}^{n} a_i b_i \right) \mathrm{mod}\, M$$

Очевидно, що усі алгоритми будуть виконуватися незалежно один від одного і паралельно у часі, що дозволяє створити структуру операційного пристрою, де кожний тракт функціонує за своєю основою. Використання принципів паралельності та незалежності формування лишків a_i обумовлює три основні властивості СЗК [2]:

1) Незалежність кожного лишку у структурі НКС СЗК. Ця властивість дає змогу сформувати сукупність інформаційно незалежних малорозрядних обчислювальних трактів (ОТ), що функціонують в структурі комп'ютерної системи (КС) у СЗК незалежно один від одного та паралельно у часі.

Варто зауважити, що:

– у загальному випадку час виконання арифметичних операцій КС у СЗК визначається часом виконання операцій за найбільшою розрядною сіткою ОТ;

– якщо у деякому ОТ виникає помилка за основою m_i, ця помилка не поширюється на інші ОТ. При цьому не має значення кратність помилки (одноразова чи багаторазова або сукупність помилок не більш значення $\left[\log_2(m_i - 1) + 1\right]$ двійкових розрядів), адже вона збережеться в цьому ОТ КС за основою m_i до кінця обчислень чи самостійно усунеться в результаті подальших обчислень (у випадку якщо помилкове значення лишку a_i за основою m_i одного числа множиться на нульовий лишок іншого числа за такою самою m_i основою СЗК);

– відокремлення незалежних ОТ КС у СЗК дозволяє проводити операції контролю, діагностики та корекцію помилок даних, а також здійснювати ремонт і технічне обслуговування без зупинок обчислення по всіх ОТ, фактично без припинення процесу рішення задачі [3].

2) Рівноправність лишків у НКС СЗК. Довільний лишок a_i числа $A_{СЗК} = (a_1, a_2, ..., a_n)$ у СЗК містить інформацію щодо усього вихідного числа. За умови виконання нерівності $m_i < m_j$ можливо програмними методами замінити ОТ КС за модулем m_i, який не працює на працездатний ОТ за модулем m_i фактично без зупинки рішення задачі.

Ця властивість дозволяє обмінюватися, на основі застосування програмних методів, такими характеристиками КС, як точність виконання арифметичних операцій, швидкодія виконання арифметичних операцій та надійність виконання арифметичних операцій фактично без зупинки обчислень в процесі рішення задачі. Відповідно, КС у СЗК може мати різну надійність функціонування в залежності від вимог, наприклад, до точності або швидкодії обчислень. Крім цього у ситуації, де відмовили певні означені елементи структури КС у СЗК це дає можливість фактично продовжувати функціонувати, щоправда або без деякої (чи деяких) функцій КС, або функціонувати з погіршеною якістю, наприклад, зі зниження швидкодії, або зі зниженням точності обчислень тощо [4].

Як наслідок, можливість використання перших двох властивостей СЗК обумовлює наявність у КС одночасно трьох видів резервування: функціонального, структурного та інформаційного.

3) Малорозрядність лишків, сукупність яких визначає НКС у СЗК. Ця властивість у першу чергу дозволяє суттєво підвищити швидкодію реалізації арифметичних операцій. Це можливо як за рахунок малорозрядності представлення лишків числа в СЗК, так і за рахунок можливості (на відміну від позиційних систем числення) застосування табличної арифметики; останнє

дозволяє реалізувати фактично за один такт роботи КС основні арифметичні операції у СЗК.

Малорозрядність лишків чисел в СЗК обумовлює широкий вибір варіантів системотехнічних рішень задач реалізації швидких арифметичних модульних операцій. Принципи їх реалізації:

– кільцевого зсуву (застосування регістрів зсуву).

– суматорний принцип (застосування малорозрядних двійкових суматорів);

– табличний принцип (використання елементів пам’яті);

Таким чином, засновуючись на перевагах використання властивостей СЗК (у порівнянні с позиційними системами числення (ПСЧ)) можна виділити наступні:

– можливість організації виконання арифметичних операцій за допомогою табличного принципу і вибіркою результату модульної операції за один такт;

– можливість створення системи контролю та корекції помилок даних безпосередньо під час обчислень в КС;

– можливість реалізації асинхронних арифметичних обчислень на рівні декомпозиції чисел;

– можливість синтезу відмовостійких цифрових пристроїв та використання пасивної та активної відмовостійкості;

– можливість створити КС з ефективним виявленням і виправленням помилок у реальному часі без зупинок в процесі обчислення;

– забезпечення високої активної відмовостійкості за рахунок можливості реконфігурації структури КС;

Отже, основні властивості СЗК дозволяють підвищити швидкодію виконання арифметичних операцій та відмовостійкість функціонування КС. Підвищення швидкодії можливе за рахунок організації паралельної обробки даних і використання табличного принципу реалізації арифметичних модульних операцій. Стає можливо створити систему контролю, діагностики та

корекції помилок у реальному часі фактично без зупинок обчислень, що неможливо у ПСЧ.

1.2. Аналіз існуючих методів реалізації арифметичних операцій

На сьогодні відомо чотири принципи реалізації арифметичних операцій:

1) Суматорний принцип (СП) (використання двійкових суматорів)

2) Табличний принцип (ТП) (застосування ПЗП)

3) Принцип кільцевого зсуву (ПКЗ) (застосування кільцевих регістрів зсуву (КРЗ)

4) Прямий логічний принцип реалізації арифметичних операцій (опис модульних операцій на рівні систем перемикаючих функцій)

Оскільки в ПСЧ під час виконання арифметичної операції обчислювального тракту (ОТ) проводять послідовну обробку розрядів, обчислення не може бути завершеним до моменту послідовного визначення усіх результатів проміжних обчислень. Тому сучасні КС мають суттєвий недолік – наявність міжрозрядного зв’язку, що призводить до часткової втрати швидкодії а також зниження надійності обчислень.

Завдяки тому, що у СЗК відсутній зв'язок між двійковими розрядами операційного пристрою (ОП) засновуючись на ТП або ПКЗ, властивість малорозрядності лишків фактично дає можливість вибору системотехнічного рішення для реалізації модульних операцій.

Операційний пристрій КС в СЗК принципово може бути виконаний в суматорному варіанті чи в табличному. Якщо ОП будується використовуючи малорозрядні суматори, кожен розряд числа оброблюється незалежно і паралельно у часі, при цьому час виконання арифметичної операції визначається часом, необхідним для отримання результату за найбільшою основою СЗК [5].

Варто звернути увагу на деякі недоліки суматорного принципу реалізації арифметичних операцій:

– складність реалізації деяких арифметичних операцій (наприклад, множення);

– великий час перетворення інформації для значних розрядних сіток КС за найбільшою основою СЗК;

– складність синтезу двійкових суматорів;

– неефективне використання двійкових елементів в ОП КС, пов’язане з наявністю збитковості максимальних чисел, що можуть буть представлені на суматорі у порівнянні з основою СЗК m_i.

Великим резервом підвищення надійності функціонування КС є використання схем матриць ПЗП, ПЛМ та ПЛІС. Доведено, що питання, пов’язані з виконанням арифметичних операцій табличними методами має сенс розглядати лише у розрізі їх застосування в КС в СЗК. При цьому не є очевидним найкращий метод з точки зору кількості обладнання ОП, оскільки при табличному методі може бути не більша кількість обладнання, ніж при реалізації суматорного принципу побудови ОП КС в СЗК. Тож пошук шляхів спрощення структури КС обумовлює необхідність створення алгоритмів реалізації модульних операцій, які дозволять збільшити ефективність застосування табличної арифметики.

Якщо роздивитися один з табличних методів реалізації модульного множення [2-4], можна побачити деякі закономірності, за рахунок чого можливе зменшення кількості обладнання. Таблиця симетрична відносно діагоналі, вертикалі та горизонталі, якщо брати числа $\frac{(m_i-1)}{2}=5$ та $\frac{(m_i+1)}{2}=6$. Симетричність пояснюється тим, що операція множення комутативна, та виконуються співвідношення

$$\begin{aligned}
&(m_i-a_i)(m_i-b_i)\equiv a_i\cdot b_i(\bmod m_i),\\
&a_i\cdot b_i+a_i(m_i-b_i)\equiv 0(\bmod m_i),\\
&a_i\cdot b_i+b_i(m_i-a_i)\equiv 0(\bmod m_i).
\end{aligned} \qquad (1.5)$$

Відповідно, для реалізації операції модульного множення фактично необхідно визначити тільки восьму частину таблиці. Проте зменшення таблиці призведе до необхідності попереднього аналізу величин вхідних операндів. Саме тому застосовуються методи спеціального кодування, що дозволяють зменшити розмір таблиці у чотири рази. Існуючий метод (використання коду табличного множення (КТМ) полягає у наступному.

Нехай дано два операнди a_i та b_i .Повтору значень можна уникнути, якщо значення у діапазоні $\left[\frac{m_i+1}{2}, m_i-1\right)$ закодувати, відповідно, як $(m_i - a_i)$ чи $(m_i - b_i)$. При цьому значення $\left[0, \frac{m_i-1}{2}\right)$ кодуються довільно. Відповідно до цього методу, для розпізнавання зазначених діапазонів вводиться так званий індекс

$$\gamma_a, \gamma_b = \begin{cases} 0, \textit{якщо } 0 \le a_i(b_i) \le \frac{m_i-1}{2}, \\ 1, \textit{якщо } \frac{m_i+1}{2} \le a_i(b_i) \le m_i - 1 \end{cases} \tag{1.6}$$

Сам алгоритм визначення результату операції модульного множення за допомогою КТМ має наступні кроки: задано $a_i = (\gamma_a, a_i')$ та $b_i = (\gamma_b, b_i')$. Для визначення результату множення двох операндів за m_i достатньо знайти $a_i' \cdot b_i' (\mathrm{mod}\, m_i)$ та інвертувати індекс γ_i за умови

$$a_i \cdot b_i (\mathrm{mod}\, m_i) = (\gamma_i, a_i' \cdot b_i' (\mathrm{mod}\, m_i)),$$

у цьому виразі

$$a_i' = \begin{cases} a_i, \textit{ якщо } \gamma_a = 0, \\ m_i - a_i, \textit{ якщо } \gamma_a = 1. \end{cases} \tag{1.7}$$

$$\gamma_i = \begin{cases} \overline{\gamma_i}, \textit{ якщо } \gamma_a \ne \gamma_b, \\ \gamma, \textit{ якщо } \gamma_a = \gamma_b. \end{cases} \tag{1.8}$$

Використовуючи цей метод ПЗП фактично зменшується у чотири рази. Зазначається також, що у деяких випадках можливо додатково зменшити

обладнання за рахунок побудови k таблиць (k – розрядність регістру для зберігання інформації щодо числа за основою СЗК). Кожна з цих таблиць дає відповіді за всіма k розрядами результату.

Вельми важливо приділити увагу принципу кільцевого зсуву (ПКЗ), який використовує так звані кільцеві регістри зсуву (КРЗ) [2, 9]. При використанні двійкових суматорів присутній вплив (хоча і значно менший за той, що є в ПСЧ) міжрозрядного зв'язку у межах m_i. Табличний варіант має дещо інший недолік: в ньому відсутній міжрозрядний зв'язок, але за умови збільшення фактичної величини модуля СЗК доволі різко зростає кількість обладнання операційного пристрою. Відома теорема Келі про ізоформність підгруп перестановок дозволяє організувати процес виконання модульних арифметичних операцій в СЗК.

Серед методів, які використовують принцип кільцевого зсуву можна визначити наступні [2, 8]:

1) *Метод двійкового позиційно-залишкового кодування.* Числа у КРЗ представлені у двійковому коді. При цьому реалізація може бути виконана за допомогою наступних методів: метод прямого зсуву; метод оберненого зсуву; метод середнього значення (засновується на підвищенні швидкодії за рахунок зменшення значення ПОКЗ); метод залишкового зсуву (шляхом використання деяких тотожностей домогтися однорідності структури реалізації модульних операцій, більшої швидкодії порівняно з операцією модульного додавання та алгоритмічної простоти реалізації); метод множинних контурів (за значенням операнду b встановлюється відповідна строчка шляхом зсуву за спеціальними контурами); метод керуючих матриць (за допомогою спеціальних матриць скорочується час виконання операції $(a_i + \beta_i) \bmod m_i$ завдяки тому, що пошук може бути здійснено спеціальним чином не за максимальну, а за необхідну кількість кроків, що стає можливо завдяки наявності збитковості у розрядах КРЗ).

2) *Метод унітарного позиційно-залишкового кодування даних.* Цей метод усуває головний недолік попереднього методу пов'язаний з тим, що

представлення чисел у двійковому коді затримує рівень швидкодії обробки інформації у КС в СЗК. У цьому методі значення представляються в унітарному коді. Порівняно з методом двійкового позиційно-залишкового кодування швидкодія зростає у $k = [\log_2(m_i - 1) + 1]$ разів.

Отже, використання такої властивості як модульність структури обчислювального процесу в СЗК дозволяє ефективно реалізовувати модульні арифметичні операції додавання за допомогою табличного принципу та принципу кільцевого зсуву.

РОЗДІЛ 2.
АНАЛІЗ ГАЛУЗЕЙ НАУКИ ТА ТЕХНІКИ З ЗАСТОСУВАННЯМ КОМП'ЮТЕРНОЇ СИСТЕМИ, ЩО ФУНКЦІОНУЄ В СЗК

2.1. Використання СЗК для підвищення швидкодії обчислювальних пристроїв

Для сучасних галузей науки та техніки важливо знаходити шляхи вдосконалення продуктивності та надійності функціонування систем. Деякі методи засновані на застосуванні складних багатомашинних комплексів, інші намагаються розмежувати процес та обчислення для паралельного виконання. Проте вони мають або низький результат у сфері користувацької продуктивності, або застосування лише для певного класу задач. Використання великих і надвеликих інтегральних схем посприяло збільшенню інтересу до табличних методів обробки даних. Проте вони теж мають свій суттєвий недолік – значна кількість необхідного обладнання [6].

Саме тому досліджено [7] СЗК на предмет можливості застосування у сучасних швидкодіючих та відмовостійких спецпроцесорах (СП) для швидкої обробки інформації. СЗК може ефективно використовуватися під час рішень дискретного перетворення Фур'є, в матричних та векторних процесорах, під час вирішення задач цифрової фільтрації та розмежування обчислень, проведених у криптографічних системах та полях Галуа. Можливість реалізації пов'язане з модульністю операцій, необхідністю підвищення користувацької продуктивності та особливістю використання модульного множення чи додавання, яке досить ефективно дозволяє використовувати можливості СЗК для вирішення поставлених практичних завдань.

Розглянемо спробу синтезу процесора в СЗК [10]. Вихідна структура СП наведена на рис. 2.1

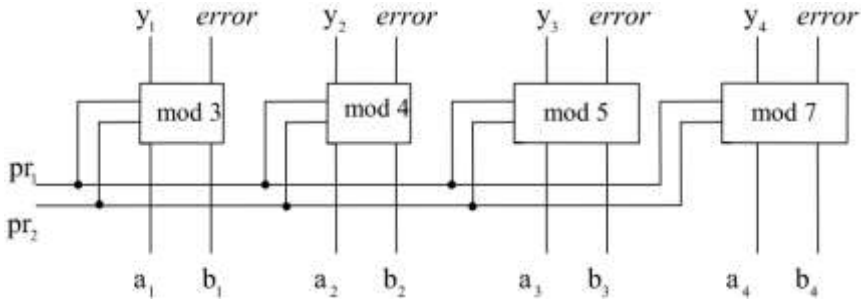

Рис. 2.1. Структурна схема спецпроцесора СЗК для 8 двійкових розрядів

Структурна схема складається з чотирьох незалежних один від одного обчислювальних трактів, що функціонують паралельно у часі за модулями $m_1 = 3, m_2 = 4, m_3 = 5, m_4 = 7$. a_i, b_i – інформаційні входи за кожним модулем; y_i – інформаційний вихід; *error* – вихід, який сигналізує про операцію, яку неможливо реалізувати (некоректні вхідні ситуації); pr_i – вхід операції, яку неможливо реалізувати.

Спецпроцесор реалізовано на ПЛІС MAX3000A. В її основі закладено макрокомірки, логічні блоки та розширювачі (паралельний та розподільний), елементи вводу – виводу та програмована матриця з'єднань. ПЛІС має чотири виходи, що закріплені за глобальними ланцюгами. У свою чергу ці ланцюги відповідають за встановлення кожної макрокомірки у третій стан. Кожний логічний блок складається з шістнадцяти макрокомірок, які з'єднані між собою за допомогою програмованої матриці. Відповідно, кожний логічний блок містить 36 входів з цією матрицею [11].

Макрокомірка ПЛІС складається з наступних основних вузлів:

– програмований регістр;
– матриця розподілу термів;
– локальна програмована матриця.

Логічні розширювачі використовуються для реалізації логічних функцій численних змінних. Розподільний розширювач містить велику кількість входів для логічної функції та з'єднує макрокомірки одного логічного блоку, а паралельний (для реалізації функцій від 5 термів) застосовує локальні матриці макрокомірок, суміжних між собою.

Всі сигнали з різних джерел надходять до програмованої матриці, серед яких сигнали зворотного зв’язку блоків, елементів вводу – виводу та спеціалізовані виводи. На кожний логічний блок подаються лише визначені сигнали під час програмування, а елементи вводу – виводу дають змогу створити режим роботи з третім станом та відкритим колектором.

Щодо мікросхеми, то було обрано систему автоматизованого проектування. Вона містить 22 входи та 14 виходів. Склад системи – чотири блоки (A,B,C,D). За результатами дослідження [10] ресурси блоків використовуються на 44%, а системи в цілому – на 68%, що дає запас для покращення інших властивостей, таких як надійність та відмовостійкість.

Таким чином, результати використання СЗК можна реалізовувати для математичних обчислень в несинхронних алгоритмах шифрування у криптографії. Це дозволить збільшити користувацьку продуктивність за умови збільшення розрядної сітки. А основні властивості СЗК (незалежність рівноправність та малорозрядність) роблять її доцільною для використання під час обчислень дискретного логарифму на еліптичних чи гіпереліптичних кривих (завдяки операції модульного множення).

2.2. Задача оптимального резервування у СЗК

Оптимальне резервування у СЗК тісно пов’язане із поняттям надійності та ефективності функціонування систем. Під ефективністю будемо розуміти ступінь досягнення системою мети свого функціонування, яка характеризує спроможність системи виконувати задані функції якомога економнішим шляхом.

Надійність системи – це здатність зберегти у часі та встановлених межах значення усіх параметрів функціонування системи, що характеризують можливість ефективного виконання поставлених функцій за поставлений час і у відповідних умовах технічного обслуговування, транспортування, ремонту та збереження [12].

В цілому задача оптимального резервування має на меті дати відповідь щодо ступені ефективності використання СЗК з метою підвищення відмовостійкості КС. У цій задачі КС представляється як сукупність послідовно з'єднаних каналів обробки інформації (КОІ), до яких для підвищення надійності функціонування додаються резервні тракти обробки інформації (ТОІ). Для вирішення задачі передбачається, що відомо якого рівня надійності потрібно досягнути шляхом додавання x_i збиткових ТОІ. Це означає, що потрібна бути задана функція надійності $R_i(x_i)$ [2]. Крім того, використання ТОІ для i – го КОІ означає додаткові затрати у розмірі c_i, які можуть вимірюватися не тільки у одиницях вартості, а й у різних інших. Нарешті також вважатимуться затрати $V_{дод}^{(1)}$ – обладнання для забезпечення надійності функціонування КС на необхідному рівні [13].

Нехай вектор стану КС має вигляд $X = (x_1, x_2, ..., x_i, ..., x_n)$, де x_i – кількість ТОІ в i – му КОІ. Тоді показник надійності КС $H(t)$ може бути представлений у вигляді:

$$H(t) = \prod_{i=1}^{n} R_i(x_i), \tag{2.1}$$

де $R_i(x_i)$ – загальна кількість ТОІ в i – му КОІ;

n – кількість КОІ в КС.

Розрахунок витрат на організацію КС виконується при цьому за наступним виразом

$$C(X) = \sum_{i=1}^{n} c_i \cdot x_i, \tag{2.2}$$

де c_i – затрати одного ТОІ в i – му КОІ.

Задача оптимального резервування може бути представлена у прямому чи зворотному вигляді.

Пряма задача – знайти $\max\limits_{(X)} H(t) = \max\limits_{X} \prod\limits_{i=1}^{n} R_i(x_i)$ маючи обмеження у вигляді $C(X) = \sum\limits_{i=1}^{n} c_i \cdot x_i \le C_{дод}$ де $C_{дод}$ – максимально можливі затрати, які можна припустити.

Зворотна задача – знайти $\min\limits_{(X)} C(X) = \min\limits_{(X)} \sum\limits_{i=1}^{n} c_i \cdot x_i$ за умови $H(t)[t = const] = \prod\limits_{i=1}^{n} R_i(x_i) \ge H_{дод}(t)[t = const]$, де $H_{дод}(t)[t = const]$ – мінімально можливий показник надійності.

Серед варіантів вирішення сформульованих задач роздивимося метод покоординатного найскорішого спуску. Він дає практично точне, а у ряді випадків абсолютно точне рішення і є достатньо простим.

Можливості СЗК, яка дозволяє суттєво підвищити відмовостійкість та надійність систем обробки інформації, визначаються структурою нерезервованої КС в СЗК, наданої на рис. 2.1.

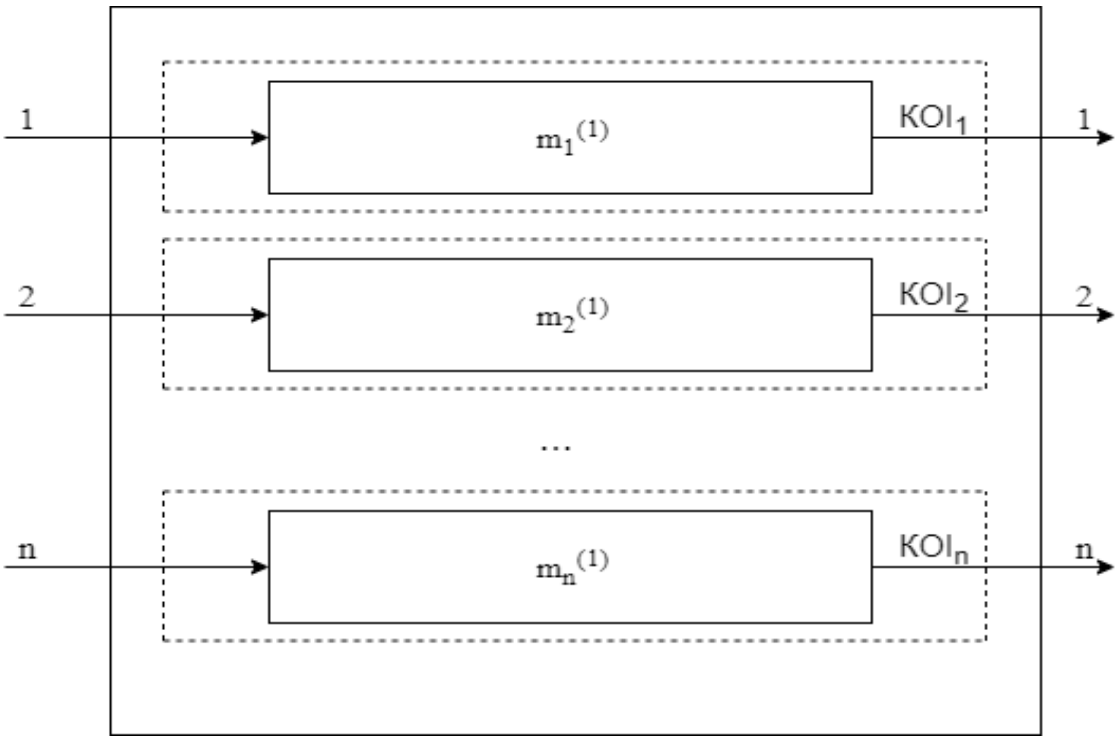

Рис. 2.2. Структурна схема вихідної нерезервованої КС

Результати вирішення поставлених задач за допомогою СЗК можуть показати як забезпечити потрібний рівень надійності $H(t)$ і мінімального значення різноманітних витрат чи навпаки, як забезпечити максимальне значення $H(t)$ за умов обмеженості ресурсів, які задані. При цьому резервована структура КС матиме вигляд (рис 2.3).

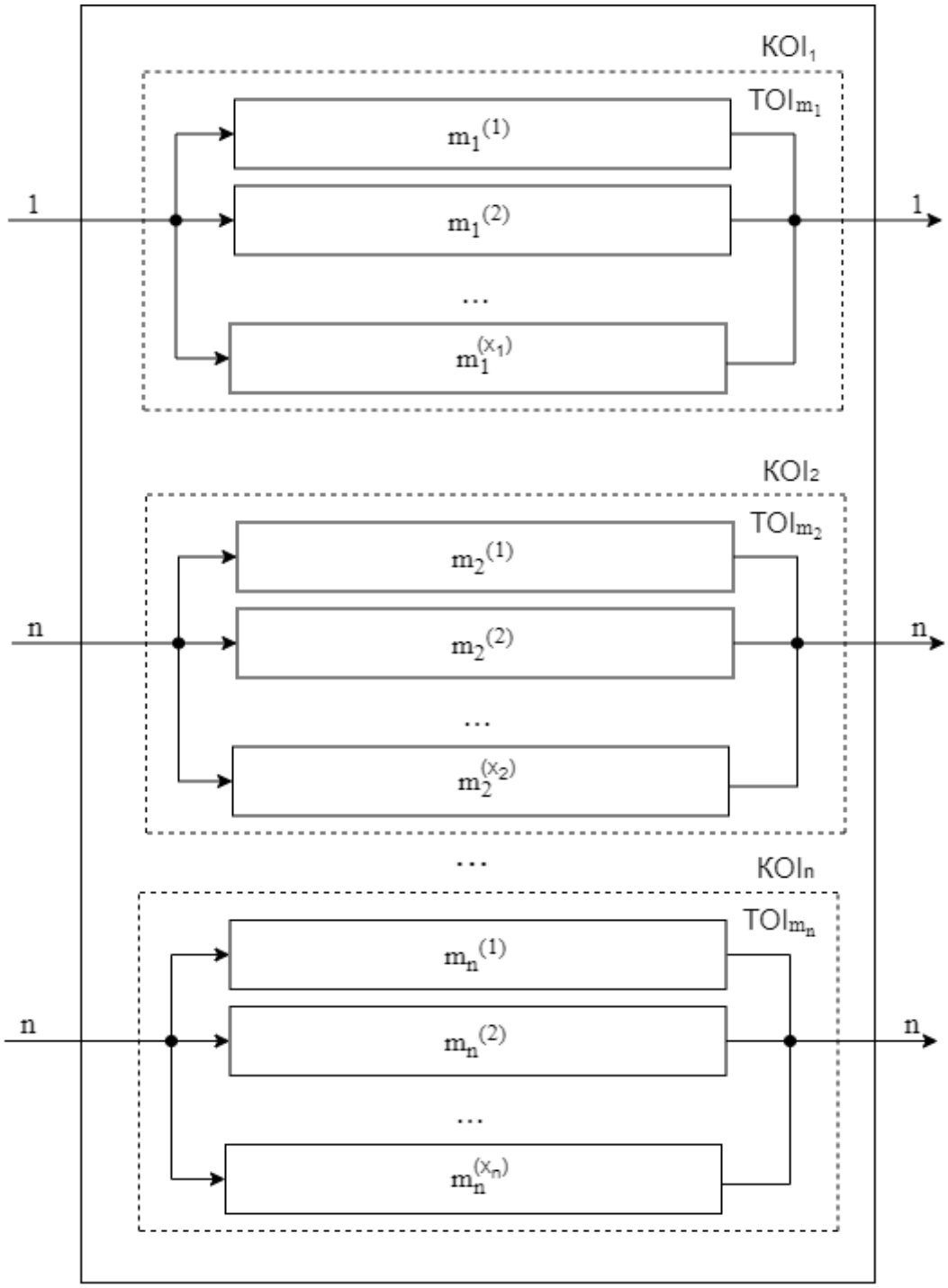

Рис. 2.3. Структурна схема резервованої КС

РОЗДІЛ 3.
РОЗРОБКА МЕТОДУ ШВИДКОЇ РЕАЛІЗАЦІЇ МОДУЛЬНИХ АРИФМЕТИЧНИХ ОПЕРАЦІЇ ТА КОНТРОЛЮ ДАНИХ

Як відомо, у СЗК число, яке представлено сукупністю лишків (1.4) може мати не тільки інформаційні, а й контрольні основи, тоді воно матиме вигляд:

$$A=(a_1,a_2,...,a_n,a_{n+1}) \tag{3.1}$$

Якщо число послідовно привести до вигляду:

$$A^{(H)}=(0,0,...,0,\gamma_{n+1}) \tag{3.2}$$

не виходячи за межі робочого діапазону, можна визначити правильність числа наступним чином: якщо $\gamma_{n+1}=0$, число правильне та лежить у межах $[0,M)$, а випадок $\gamma_{n+1}\neq 0$ вказує на неправильність числа, при цьому воно буде знаходитися в іншому діапазоні $[j\cdot M,(j+1)\cdot M)$, де $j=(1,2,..m_{n+1}-1)$, а $(j+1)$ – власне той інтервал, куди воно потрапило.

Розберемо детальніше процес нульовизації на прикладі. Визначення робочого та повного діапазонів (рис. 3.1).

m1	m2	m3	m4	m5		Mr	Mp
3	4	5	7	13		420	5460

Рис. 3.1. Робочий та повний діапазони

Mr – робочий діапазон, Mp – повний діапазони.

Визначимо кількість інтервалів, що перевіряються (N_i =]M/m_i[ ) для основ $m_i=m_{n+1}=m_5=13$, $m_i=m_1=3$, $m_i=m_n=m_4=7$ (рис. 3.2).

Кількість інтервалів, що перевіряються (Ni =]M/mi[)		
mi=m1	mi=mn	mi=mn+1
140	60	33

Рис. 3.2. Кількість інтервалів

Константи нульовизації визначено на рис. 3.3.

	Залишок	Визначення констант нульовизації				
		m1 = 3	m2 = 4	m3 = 5	m4 = 7	m5 = 13
0	**0000**	00	00	000	000	0000
1	**0001**	01	01	001	001	0001
2	**0010**	10	10	010	010	0010
3	**0011**	00	11	011	011	0011
4	**0100**	01	00	100	100	0100
5	**0101**	10	01	000	101	0101
6	**0110**	00	10	001	110	0110
7	**0111**	01	11	010	000	0111
8	**1000**	10	00	011	001	1000
9	**1001**	00	01	100	010	1001
10	**1010**	01	10	000	011	1010
11	**1011**	10	11	001	100	1011
12	**1100**	00	00	010	101	1100

Рис. 3.3. Константи нульовизації

Виконаємо перевірку чисел (по 5 чисел) з діапазонів $[0, M-10), [M-10, M+10], [M+10, M_0)$ при різних значеннях m_i (рис. 3.4)

Візьмемо $m_i = m_{n+1} = m_5 = 13$

	Перевірка чисел		
	[0, 410)	[410-430]	[430,5460)
Числа	3	412	431
	55	415	500
	67	416	1012
	123	420	3045
	406	429	5400

Рис. 3.4. Вихідні числа з діапазонів

Представимо числа у двійковому вигляді для визначення необхідної константи нульовизації з блоку констант нульовизації.

Після того, як знайдено двійковий вигляд чисел, які здвигнуті на лівий край їх інтервалу, можна перевірити ці числа. Для цього знайдемо їх значення в ПСЧ, та значення інтервалу n_A.

ЗСК значеня					Добуток	ПСЧ знач	nA
m1 = 3	m2 = 4	m3 = 5	m4 = 7	m5 = 13			
0	0	0	0	0	0	0	0
1	0	2	3	0	21892	52	4
2	1	0	2	0	16445	65	5
0	1	2	5	0	27417	117	9
1	2	3	4	0	31798	4498	346
1	3	3	4	0	33163	403	31
1	3	3	4	0	33163	403	31
2	0	1	3	0	22256	416	32
2	0	1	3	0	22256	416	32
0	2	4	2	0	23634	1794	138
0	1	4	2	0	22269	429	33
2	2	4	4	0	38714	494	38
2	1	1	0	0	11921	1001	77
0	2	2	4	0	24882	3042	234
1	3	0	5	0	27235	5395	415

Рис. 3.5. Перевірка чисел та визначення інтервалу

Оскільки для $m_i = m_{n+1} = m_5 = 13$, $N_i = 33$, то числа n_A більші, ніж 33 і їх значення в ПСЧ більші за робочий діапазон ($M_r = 420$), тому робиться висновок, що присутня помилка даних. Також слід зауважити, що таке число як 420 (а разом із ним 421, 422,…, 428) за рахунок зміщення вважаються правильними, хоча вони більші за робочий діапазон.

Достовірність контролю даних становить 0,979020979. Для інших основ достовірність буде дорівнювати 1.

Проведено контроль даних за допомогою позиційної ознаки непозиційної кодової структури. Було виявлено помилку серед даних, що перевірялись, що свідчить про те, що цей метод не завжди забезпечує достовірний результат контроля. Це пояснюється тим, що деякі числа, більші за робочий діапазон, а отже неправильні, здвигаються до лівого краю інтервалу, який потрапляє у межі робочого діапазону. Тому для них робиться висновок, що ці числа правильні (хоча вони більші за робочий діапазон).

Але цей приклад можна поліпшити з точки зору часу виконання операції нульовизації. Ідея складається у тому, що можна нульовизувати відразу дві основи, що дозволить вдвічі швидше провести дану операцію. Оскільки може бути два варіанти завдяки кількості основ (парна чи непарна кількість основ).

Основи для процедури нульовизації групуються за наступною схемою (рис 3.6)

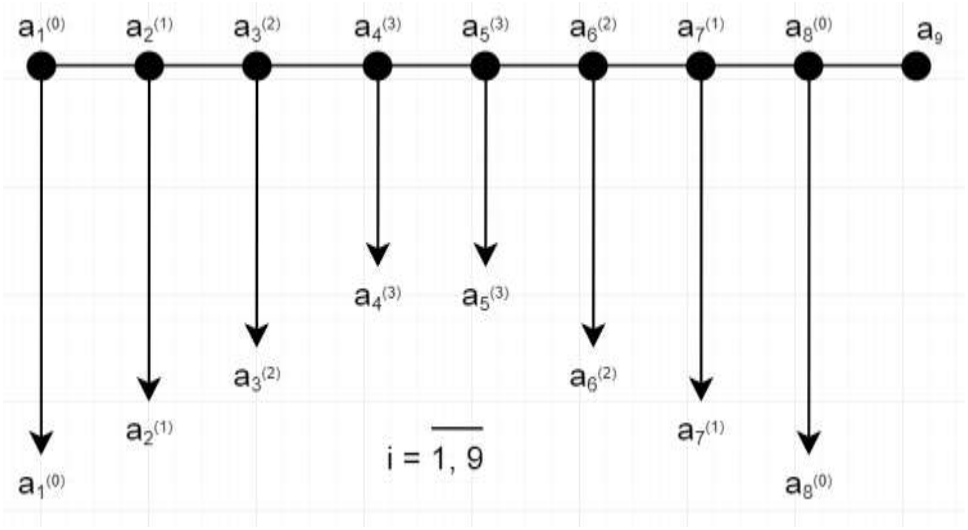

Рис. 3.6. Схема групування основ

Умовний час виконання процедури нульовизації розраховується за наступною формулою:

$$T_{HP} = n \cdot \tau_{скл} = 8$$

Отже, на відміну від послідовної нульовизації, час, витрачений на виконання фактично у 2 рази менше, проте доводиться створювати більше констант нульовизації та вибирати серед них відразу по двом основам необхідну.

Наступний метод, який дозволяє пришвидшити час виконання операції заснований на тому, що виконання операції віднімання константи та її вибір розмежовані у часі. Це дозволяє за рахунок попереднього визначення наступного залишку не звертатися до блоку констант нульовизації а відразу провести операцію віднімання. Тобто поки визначається i константа нульовизації для числа $A^{(i)}$ за значенням залишку $a_{i+1}^{(i)}$ (за основою m_{i+1}) в ОТ за

основою m_{i+2} може бути сформовано $a_{i+2}^{(i+1)}$ значення залишку, за яким буде проводитися наступний етап вибору константи нульвизації.

Аналітично можна пояснити це наступним чином [14]:

$$a_{i+2}^{(i+1)} = [a_{i+2}^{(i)} - \Delta a_{i+2}] \bmod m_{i+2} \tag{3.3}$$

Основна відмінність, яка дозволяє скоротити час виконання – необхідність звернення до блоку констант нульвизації один раз на кожні два такти замість постійного звернення, а отже загальна кількість тактів – $[n/2]$.

Час виконання процедури нульвизації можна знайти за формулою:

$$T_{НЗ} = \left(\left[\frac{n-1}{2} \right] + n \right) \cdot \tau_{скл} \tag{3.4}$$

Запропонований метод можна покращити, використовуючи принцип, за яким було здійснено паралельну нульвизацію – розпаралелити процес на дві основи (за двома залишками $a_{i+1}^{(i)}$ та $a_{n-i}^{(i)}$). Тобто знаходяться два наступні залишки $a_{i+2}^{(i+1)}$ та $a_{n-i-1}^{(i+1)}$ числа $A^{(i+1)}$ а дві операції (віднімання та вибір константи нульвизації) робляться одночасно.

Величини Δa_{i+2} та Δa_{n-i-1} , які віднімаються аналітично знаходяться наступним чином:

$$\begin{aligned} a_{i+2}^{(i+1)} &= [a_{i+2}^{(i)} - \Delta a_{i+2}] \bmod m_{i+2}, \\ a_{n-i-1}^{(i+1)} &= [a_{n-i-1}^{(i)} - \Delta a_{n-i-1}] \bmod m_{n-i-1} \end{aligned} \tag{3.5}$$

Оскільки процес нульвизації проводиться одночасно по двох основах, після кожних двох операцій віднімання потрібен ще один такт для визначення адреси та звернення до блоку констант нульвизації. Відповідно до цього [2] час, необхідний для виконання операції нульвизації визначається:

$$T_{НПЗ} = \left[\frac{n+1}{2} \right] \cdot \tau_{скл} + \left[\frac{\frac{n+1}{2}+1}{2} \right] \cdot \tau_{виб} \tag{3.6}$$

Оскільки $\tau_{скл} = \tau_{виб}$ то:

$$T_{НПЗ} = \left(\left[\frac{n+1}{2} \right] + \left[\frac{\frac{n+1}{2}+1}{2} \right] \right) \cdot \tau_{скл} \qquad (3.7)$$

Розглянуті методи дають змогу прискорити виконання операції нульвизації, але, на жаль, не є принципово новими науковими методами. Пошук нового методу призвів до формулювання концепції, яка може бути використана для зменшення часу виконання операцій. Вона засновується на тому, щоб якомога більше розпаралелити процес виконання арифметичних операцій та зменшити час виконання звернень до готових блоків констант. Перспективним може бути пошук різних варіацій існуючих методів, де б обробка виконувалася, наприклад, одночасно за чотирма чи більше основами, проте відповідний математичний інструмент ще не розроблений.

ВИСНОВКИ

В ході роботи було проаналізовано та перевірено основні властивості СЗК: незалежність лишків, що дає змогу організувати роботу у вигляді паралельно функціонуючих трактів, що працюють незалежно один від одного; рівноправність лишків, завдяки чому в кожному лишку міститься інформація щодо усього вихідного числа та відкривається можливість заміни непрацездатного тракту без переривання обчислень; малорозрядність лишків, що суттєво зменшує час виконання арифметичних операцій. Зазначені властивості є унікальними та не мають аналогів у ПСЧ, що робить СЗК перспективною та привабливою для підвищення швидкодії реалізації модульних арифметичних операцій.

Доведено актуальність використання СЗК для КС на прикладі синтезу спецпроцесора. Дана конструкція складається з чотирьох трактів (8 двійкових розрядів), кожен з яких функціонує незалежно один від одного. Спецпроцесор реалізовано на ПЛІС MAX3000A. Це підтвердило, що результати використання СЗК можна реалізовувати для математичних обчислень в несинхронних алгоритмах шифрування у криптографії, а основні властивості СЗК роблять її доцільною для використання під час обчислень дискретного логарифму на еліптичних чи гіпереліптичних кривих (завдяки операції модульного множення). Приділено увагу для вирішення питання оптимального резервування для збільшення надійності та швидкодії виконання арифметичних операцій у СЗК. Таким чином можливо створити високонадійні та відмовостійкі структури КС в СЗК. Було проаналізовано методи реалізації арифметичних операцій, які засновуються на принципах, таких як: суматорний принцип (СП); табличний принцип (ТП); принцип кільцевого зсуву (ПКЗ); прямий логічний принцип реалізації арифметичних операцій.

Досліджено переваги та недоліки вищезазначених принципів і можливість їх практичного застосування для вирішення питання швидкодії реалізації модульних арифметичних операцій.

На основі дослідження існуючих методів, а також їх практичних реалізацій було здійснено пошук нового методу реалізації арифметичних операцій, але принципово нового методу, на жаль, не було розроблено, проте на практиці було показано яким чином можна досягти збільшення швидкодії та сформульовано загальну концепцію, яка може бути використана для майбутніх досліджень цьому напрямі. Це проаналізовано на прикладі процесу нульовизації, де перший метод послідовної нульовизації мав час реалізації $T_{H1} = 2 \cdot n \cdot \tau_{скл}$, а останній метод, заснований на попередньому визначенні результатів (метод паралельної нульовизації з попереднім визначенням залишків) вже реалізовано за час:

$$T_{НПЗ} = \left(\left[\frac{n+1}{2} \right] + \left[\frac{\frac{n+1}{2}+1}{2} \right] \right) \cdot \tau_{скл}$$

що майже втричі менше за початковий.

На основі розробленого методу швидкої реалізації модульних арифметичних операцій було подано заявку на корисну модел пристрою, який функціонує у СЗК для додавання лишків a_i і b_i чисел за модулем m_i СЗК, особливість якого полягає у тому, що він здійснює операцію додавання лишків значно скоріше існуючого прототипу та виключає зі своєї структури порівняно складу схему порівняння двох двійкових чисел. Крім того, прототип у випадку переповнення розрядної сітки формував сигнал, який подавався на вхідні регістри, що фактично призводило до виконання арифметичної операції додавання ще раз і, відповідно, швидкодія функціонування пристрою зменшувалася фактично в два рази. Розглянуто три приклади функціонування пристрою (три різні можливі варіанти поведінки сигналів) і показано достовірність отриманих результатів. Обґрунтовано підвищення швидкодії функціонування пристрою порівняно з існуючими аналогами та прототипом та доведено його практичну значущість.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Краснобаєв В. А. Застосування системи залишкових класів у машинній арифметиці / В. А. Краснобаєв, С. О. Кошман // Вісник Харківського державного технічного університету сільського господарства. Проблеми енергозабезпечення та енергозбереження в АПК України. – Х. : ХДТУСГ, 2003. – Вип. 19. – С. 134-136.

2. Модели и методы обработки данных в системе остаточных классов: монография / [Краснобаев В. А., Кошман С. А., Мороз С. А. и др.] ; под ред. С. А. Кошмана – Х. : ООО "В деле", 2017. – 197с.

3. Кошман С. А. Табличный метод обработки цифровой информации в классе вычетов / С. А. Кошман, С. Н. Деренько, В. А. Краснобаев // Радіоелектронні і комп'ютерні системи. – 2006. – № 5 (17). – С. 171–175.

4. 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: збірник наукових праць. – Харків: ХУПС ім. І. Кожедуба, 2015. – Вип. 3 (128). – С. 57-61.

5. 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. – № 7 (26). – С. 219–221.

6. Акушский И.Я. Машинная арифметика в остаточных классах / И.Я. Акушский, Д.И. Юдицкий. – М. : Сов. радио, 1968. – 440 с.

7. Янко, А. С. Основные свойства непозиционной системы счисления [Текст] / В. А. Краснобаев, С. В. Сомов, А. С. Янко // Системи управління, навігації та зв’язку : зб. наук. пр. / Полтавський національний технічний університет імені Юрія Кондратюка. – П.: ПолтНТУ, 2013. – Вип. 1(25). – С. 110-113.

8. Мартиненко С. О. Метод технічної реалізації арифметичних операцій у модулярній системі числення на основі використання принципу кільцевого зсуву / С.О. Мартиненко, В.А. Краснобаєв, С.О. Кошман, О.А. Замула, М.С. Деренько // Вісник ХНТУСГ імені Петра Василенка, 2009. – Вип. 87. – С. 71–73.

9. Барсов В. И. Устройства обработки информации в модулярной системе счисления на основе применения метода унитарного кодирования / В. И. Барсов, В. О. Жадан, В. А. Краснобаев, Е. А. Сотник // Системи обробки інформації. - 2011. - Вип. 8. - С. 25-28. - URL: http://nbuv.gov.ua/UJRN/soi_2011_8_7 (дата звернення 15.06.2019).

10. Фурман И. А. Вариант синтеза процессора в системе остаточных классов / И. А. Фурман, С. А. Кошман, В. А. Краснобаев // Радиотехника и Информатика. – 2003. - №2. – С. 94-96.

11. Краснобаев В. А. Расчет и сравнительный анализ производительности компьютерной системы обработки целочисленных данных, функционирующих в системе остаточных классов / В. А. Краснобаев, А. С. Янко, П. Н. Гроза, С.А. Кошман, А. П. Гроза, Ю. П. Бендес // Системи обробки інформації: збірник наукових праць. – Харків: ХУПС ім. І. Кожедуба, 2015. – № 1 (126). – С. 111-117.

12. Янко, А. С. Метод табличной реализации операции умножения в классе вычетов [Текст] / В. А. Краснобаев, А. С. Янко, С. А. Кошман // Системи обробки інформації : зб. наук. пр. / Харківський університет Повітряних Сил імені Івана Кожедуба. – Х.: ХУПС, 2014. – Вип. 4(120). – С. 121-127.

13. Янко, А. С. Исследование производительности компьютерной системы обработки целочисленных данных, функционирующей в системе остаточных классов [Текст] / В. А. Краснобаев, А. С. Янко, С. А. Кошман, С. В. Сомов, Ю. П. Бендес // Збірник наукових праць Харківського університету Повітряних Сил імені Івана Кожедуба. – Х.: ХУПС, 2015. – Вип. 1(42). – С. 48-52.

14. Yanko, A. S. The method of error correction in the system of residual classes [Text] / V. A. Krasnobayev, A. S. Yanko, S. A. Koshman // Chemia i chemiczne technologie techniczne nauki ekologia nauk biologicznyhch geografia i geologia rolnictwo medycyna. – Przemyśl : Nauka i Sdudia, 2015. – 5(136). – P. 51-62.

ДОДАТОК А

Акти впровадження та реалізації