УДК 519.6
DOI: 10.32626/2308-5878.2024-25.70-82

В. К. Задірака, академік НАНУ, д-р фіз.-мат. наук, професор,
А. М. Терещенко, канд. фіз-мат. наук,
І. В. Швідченко, канд. фіз.-мат. наук
Інститут кібернетики імені В. М. Глушкова НАН України, м. Київ

S-СЛІВНА АРИФМЕТИКА ТА ВИСОКОТОЧНІ ОБЧИСЛЕННЯ

Аналізуються тонкощі використання S-слівної арифметики, вплив значення параметру S на оцінку похибки заокруглення; що таке високоточні обчислення і де вони використовуються. Як сфери застосування S-слівної арифметики розглядаються задачі двоключової криптографії, комп'ютерної стеганографії та задачі трансобчислювальної складності. Для розробки алгоритмів S-слівної арифметики використовуються послідовні, паралельні, квантові моделі обчислення, а також використовуються системи залишкових класів. Розглядаються архітектурні особливості обчислювальної системи для реалізації ефективного алгоритму у різних моделях обчислень. Розглядаються особливості перенесення алгоритмів до іншої моделі обчислень. Для паралельної моделі обчислень зазначено важливість зменшення пов'язаних кроків, який може збільшувати кількість оброблюваних даних, але дозволяє залучити більшу кількість паралельних процесорів. Такий підхід є у протиріччі з методом, який зменшує кількість оброблюваних даних, і є необхідність дотримання балансу між двома цими методами у паралельній моделі обчислень. Для квантової моделі обчислень поєднання кубітів є ключовим фактором у визначенні квантового об'єму. Фізична схема визначає, які пари кубітів можуть бути заплутані у квантовому комп'ютері. Проведено аналіз складності реалізації операцій S-слівної арифметики у послідовній, паралельній та квантовій моделях обчислення. Надана апріорна оцінка загальної складності за кількістю однослівних операцій додавання, віднімання, порівняння, бітових операцій у разі реалізації S-слівних операцій у паралельній моделі обчислень. Наводиться інформація про постійно діючий науковий форум «Питання оптимізації обчислень», тематика якого тісно пов'язана з темою (1969-2023 р.р.).

Ключові слова: *S-слівна арифметика, високоточні обчислення, послідовна модель обчислень, паралельна модель обчислень, квантова модель обчислень, S-слівна операція додавання, S-слівна операція віднімання, S-слівна операція порівняння, S-слівна операція знаходження суми.*

Вступ. S-слівна арифметика та тонкощі її використання. У зв'язку з ростом складності задач, які треба розв'язувати, проблемами big data, використанням нелінійних моделей для складних об'єктів, зада-

чами криптографії і криптоаналізу, звичайна однослівна (а також і двослівна) арифметика не справляється з накопиченням похибки заокруглення, яка супроводжує необхідний обчислювальний процес.

Це призводить до того, що комп'ютерні моделі не мають нічого спільного з фізичними. Це говорить про те, що похибка заокруглення стає домінуючою в порівнянні з похибками неусувної та метода.

Вихід є – застосування для обчислень *S*-слівну арифметику. Тобто, коли для запису в комп'ютері застосовується не одна, а *S* комірок пам'яті. Тоді похибка заокруглення буде пропорційна $2^{-S\tau}$, а не $2^{-\tau}$ при однослівній арифметиці. Параметр *S* дозволяє регулювати накопичення похибки заокруглення (класичне, рандомізоване, відсічення) [1]. Правило заокруглення імовірнісне, при якому відбувається менше накопичення похибки заокруглення (більш точне правило заокруглення) – це рандомізоване (ймовірносте), потім іде класичне і саме гірше (яке на сьогоднішній день всюди застосовано) – відсічення.

Виникає питання, чому саме найгірше за якістю правило заокруглення використовується у більшості типів комп'ютерів?

Якщо пригадати історію створення комп'ютерів, то приблизно на початку сімдесятих років відсічення прийшло на заміну класичному правилу заокругленню (згадаємо, наприклад, комп'ютери М-20, М-220) прийшло відсічення для комп'ютерів серії ЕС ЕОМ. Ці комп'ютери орієнтовані, в основному, на задачі типу АСУТП і для їх розв'язання висока точність не була потрібна. Електронщики, користуючись тим, що відсічення просто схемно реалізується і не замислюючись нам тим (і тим самим спрощуючи собі життя), що складність задач буде зростати, продовжували його тиражувати для нових поколінь комп'ютерів.

Тут треба ще згадати, що теорія похибки заокруглення, яка була створена Дж. Уілкінсоном [2], базувалась на класичному правилі заокруглення і її треба обережно використовувати (враховуючи відповідні коригуючі константи [3]) для інших правил заокруглення.

Що таке високоточні обчислення і де вони використовуються. Під високоточними (або підвищеної точності) обчисленнями будемо розуміти обчислення з максимально можливою точністю (яку забезпечить оптимальні за точністю та близькі до них алгоритми при даній інформації про задачу).

Такі задачі виникають в загальної теорії оптимальних алгоритмів [4], апроксимації функцій, чисельному інтегруванні, при розв'язанні задач Коші для систем диференціальних рівнянь, криптографії, стеганографії [5], астрономії і багатьох інших класах задач.

Наприклад, монографія [6] присвячена побудові оптимальних за точністю квадратурних і кубатурних формул обчислення інтегралів

від швидкоосцилюючих функцій підінтегральних функцій для різної степені гладкості. Такого типу інтеграли широко використовуються в цифровій обробці сигналів і зображень, при кодуванні інформації, при розв'язанні задач математичної фізики, механіки, тощо.

S-слівна арифметика також необхідна при розв'язанні задач математичного моделювання, розв'язанні задач поганообумовлених систем алгебраїчних рівнянь, тощо.

Використання послідовної, паралельної, квантової моделей обчислень та обчислень в системі залишкових класів при розробці, тестуванню та реалізації алгоритмів *S*-слівної арифметики. Архітектура комп'ютера (послідовного, паралельного, квантового) визначає арифметику у вигляді елементарних операцій. Тому поділ на послідовну, паралельну та квантову моделі обчислень необхідний для того, щоб максимально враховувати архітектурні особливості обчислювальної системи для реалізації ефективного алгоритму [7, 8].

Під послідовною моделлю обчислень розглядаємо модель, у якій алгоритм виконується на одному ядрі, оброблювані дані знаходяться в основній пам'яті комп'ютера, у разі аналізу складності відбувається оцінка кількості операцій, які виконуються одна за одною послідовно.

Під паралельною моделлю обчислень розглядаємо модель *GPU* (*Graphics Processing Unit*) відеокарти, яка відповідає моделі *SIMD* (*Single Instruction-Multiple Data* – потокові процесори виконують одну інструкцію одночасно, оперуючи різними даними) за класифікацією Флінта, *GPU* має власну пам'ять, яка значно швидше пам'яті основного процесора. Важливою частиною паралельних алгоритмів є команди синхронізації. Наявність команд синхронізації дозволяє алгоритми, які побудовані для моделі *SIMD*, легко адаптувати для моделей *MISD* (*Multiple Instruction-Single Data*) та *MIMD* (*Multiple Instruction-Multiple Data*). Підтримка векторних операцій дає змогу паралельному процесору обчислювати на порядок більше арифметичних операцій у порівнянні з послідовним процесором.

Під моделлю квантового комп'ютера розуміють пристрій, функціонування якого ґрунтується на двох принципах квантової механіки: принципі суперпозиції та явищі квантової заплутаності; якщо класичний комп'ютер оперує двійковими бітами (0 або 1), то квантовий комп'ютер використовує дворівневі квантово-механічні системи (кубіти), які можуть знаходитися у суперпозиції станів; заплутаність стосується станів більш ніж одного кубіта, комбінований стан кубітів містить більше інформації, ніж кубіти окремо.

Перенесення алгоритмів до іншої моделі обчислень. Існує дуже багато алгоритмів реалізації операцій *S*-слівної арифметики для послідовної моделі обчислень. У разі реалізації алгоритмів у парале-

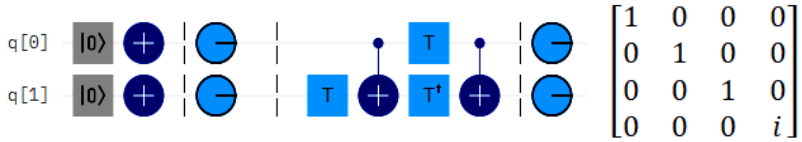
льній моделі обчислень є необхідність перенесення існуючих алгоритмів з послідовної моделі обчислень з метою пришвидшення реалізації або порівняння нових алгоритмів чи їх модифікацій з існуючими алгоритмами. Наступним кроком реалізації операції у паралельній моделі обчислень є оптимізація. Одним із підходів оптимізації є зменшення зв'язаних кроків за рахунок заміни їх простішими та більш однотипними, але незв'язаними операціями, що зазвичай збільшує кількість задіяних процесорів, але дає можливість зменшити загальну кількість операцій, виконуваних кожним з паралельних процесорів. Прикладами такого підходу є реалізації *S*-слівної операції множення на основі швидкого перетворення Фур'є або методом Монтгомері у системі залишкових класів. Іншим підходом є зменшення обсягу оброблюваних даних, що дає змогу зменшити кількість задіяних паралельних процесорів, зберігаючи загальну кількість операцій, виконувану кожним з паралельних процесорів. Прикладом такого підходу є обчислення обох співмножників за рахунок одного прямого швидкого перетворення Фур'є. Оптимізація також може використовувати резерви оптимізації обчислень, які не збільшують кількість кроків алгоритму. Підходи, щодо зменшення зв'язних кроків та зменшення обсягу оброблюваних даних, є у протиріччі між собою. Для ефективного алгоритму необхідне дотримання балансу між цими підходами. Наведений перелік підходів оптимізації не є вичерпним.

Особливістю квантового комп'ютера є те, що він не підтримує класичні операції такі, як додавання, віднімання, множення, тощо. Теоретично доведено, що будь-який класичний алгоритм може бути перенесено у квантову модель обчислень. Універсальними інструментами для задачі перенесення класичних алгоритмів, включаючи реалізації операцій *S*-слівної арифметики, у квантову модель обчислень є вентилі Тоффолі, Переса, Фредкіна та інші. Широкий набір реалізацій універсальних вентилів з використанням різних базових вентилів дозволяє для різних частин квантової схеми отримувати кращу ефективність.

Авторами була досліджена схема реалізації, яка надана на рис. 1. З таблиці видно, що початкові стани кубітів $|0\rangle$ та $|1\rangle$ після обчислення схеми будуть також $|0\rangle$ та $|1\rangle$. Якщо у разі виконання реалізації схеми з'являються інші стани, наприклад, такі, як $|i\rangle$, $|-i\rangle$, $|-1\rangle$ та інші, то кажуть, що у результаті обчислення схеми присутній поворот фази, який є основною відмінністю реалізацій квантових алгоритмів від класичних алгоритмів. Поворот фази має вплив на результат всієї схеми, що не є бажаним. Отримання результату для потрібних початкових станів у необхідній фазі значно додає складності. На прикладі

лем 1 та 2 та рис. 1 та 2 показано методи повороту фази для стану $|11\rangle$ з метою компенсації фази.

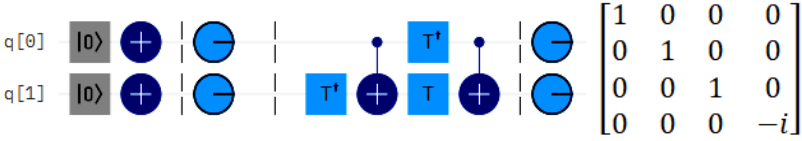
Лема 1. *Для двокубітної схеми для початкового стану* $|11\rangle$ *поворот фази на* $\pi/2$ *може бути отримано за допомогою двох вентилів CX, двох вентилів T (поворот фази на* $\pi/4$*) та одного вентиля T′ (поворот фази на* $-\pi/4$*).*



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$$

Рис. 1. *Схема повороту фази на* $\pi/4$ *для початкового стану* $|11\rangle$

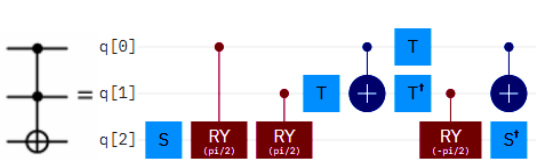
Лема 2. *Для двокубітної схеми для початкового стану* $|11\rangle$ *поворот фази на* $3\pi/2$ *може бути отримано за допомогою двох вентилів CX, одного вентиля T (поворот фази на* $\pi/4$*) та двох вентилів T′ (поворот фази на* $-\pi/4$*).*



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -i \end{bmatrix}$$

Рис. 2. *Схема повороту фази на* $3\pi/2$ *для початкового стану* $|11\rangle$

На науковому семінарі «Квантові обчислення» розглянуто метод побудови вентиля Тоффолі на чотирьох і більше кубітах. Частиною цього методу є покроковий метод корегування повороту фази модифікації реалізації вентиля Марголуса. Реалізація на рис. 3, яка побудована на основі п'яти двокубітних вентилів вважається оптимальною реалізацією. На рис. 3 вентилі *S*, *S′*, *T*, *T′* використовуються для компенсації повороту фази результату обчислення схеми (див. рис. 1).



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Рис. 3. *Схема реалізації вентиля Тоффолі на основі венитлів CRY, CX, S, S′, T, T‘ та представленням вентиля Тоффолі у матричному вигляді*

На рис. 3 представлення вентиля Тоффолі надано у матричному вигляді з урахуванням того, що нумерація кубітів відбувається зверху вниз і кубіт з нульовим індексом є у верхній позиції.

На основі вентиля Тоффолі утворюють універсальний набір елементів для класичних обчислень, як показано на рис. 4.

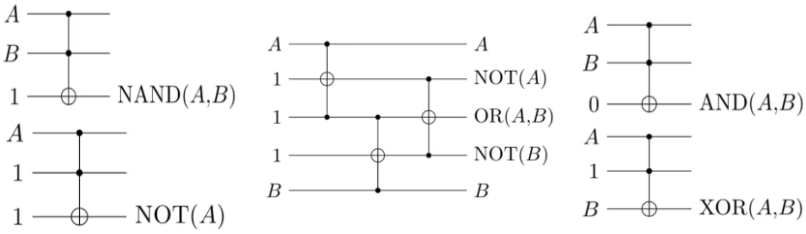


***Рис. 4**. Реалізація елементів NAND, NOT, OR, AND, XOR на основі вентиля Тоффолі*

Аналіз складності реалізації операцій *S*-слівної арифметики у послідовній, паралельній та квантовій моделях обчислення. Для послідовної моделі обчислень найбільший вплив на час виконання алгоритму реалізації операції *S*-слівної арифметики мають операції множення та ділення, що, в свою чергу, впливає на вибір алгоритму реалізації у послідовній моделі обчислень [9]. Так, наприклад, складність реалізації *S*-слівної операції множення методом множення «у стовпчик» за кількістю однослівних операцій множення та однослівних операцій додавання має наступний вигляд:

$$O^*(S) = S^2,\ O^+(S) = 2S^2. \qquad (1)$$

Для аналізу складності у послідовній моделі обчислень характерним є те, що обчислюється загальна складність за кількістю операцій. Для прикладу у табл. 1 надана складність алгоритму знаходження лишку зі кількістю операцій.

Таблиця 1

Операція	**Алгоритм**		
	Стандартний	**Монтгомері**	**Баретта**
Множення	$S(S + 2.5)$	$S(S + 1)$	$S(S + 4)$
Ділення	S	0	0
«Передобчислення»	Нормалізація	$-d_0^{-1} \bmod b$	$[b^{2S}/D]$
Перетворення аргументів	Немає	D_S – дільник	Немає
«Післяобчислення»	Денормалізація	Редукція Монтгомері	Немає
Обмеження	Немає	$W_{2S} < D_S b^S$, НСД(W_{2S}, $D_S b^S$) = 1	$W_{2S} < b^{2S}$

Необхідно зауважити, що у роботі [10] А. В. Анісімов запропонував новий алгоритм обчислення операції модулярної редукції $x \bmod m$, який має кращі показники, ніж алгоритм Монтгомері. Алгоритм не потребує

змін початкових та заключних значень аргументів. Час передобчислень не перебільшує двох множень. Алгоритми обчислення лишку є дуже зручними для перенесення у паралельну модель обчислень.

У 1967 році Амдал у своїй роботі [11] сформулював закон, який визначає верхню межу корисності від збільшення кількості процесорів в обчислювальній системі. Закон стверджує, що невелика частина програми, що не піддається розпаралелюванню, обмежить загальне прискорення від розпаралелення. Закон надано у вигляді формули (2)

$$K_p = 1/\left(\alpha + \frac{1-\alpha}{p}\right), \qquad (2)$$

де α – частина послідовних обчислень, p – кількість процесорів.

На рис. 5 надана залежність коефіцієнту K_p для різних значень α та p.

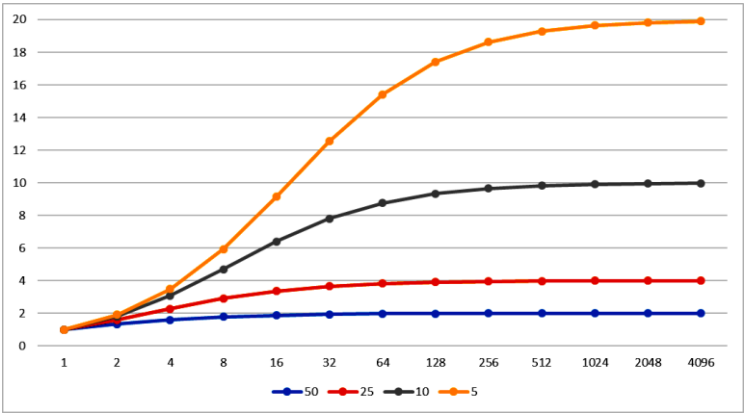


Рис. 5. *Залежність коефіцієнту прискорення K_p від відсотку послідовних обчислень α та кількості задіяних процесорів p*

З рис. 5 можна побачити, що, якщо відсоток послідовних обчислень дорівнює α = 5% (або 0.05), то коефіцієнт прискорення асимптотично наближається до 20 зі збільшенням кількості задіяних процесорів. Подальше збільшення задіяних процесорів, з практичної точки зору, збільшує обсяг накладних витрат, що призводить до зменшення коефіцієнту прискорення.

Важливою характеристикою складності у паралельній моделі обчислень є складність за кількістю операцій для одного процесора, який виконає найбільшу кількість операцій, з розрахунку на те, що не всі процесори можуть бути однаково завантажені на всіх частинах розпаралелення обчислення. Треба зауважити, що у разі розпарале-

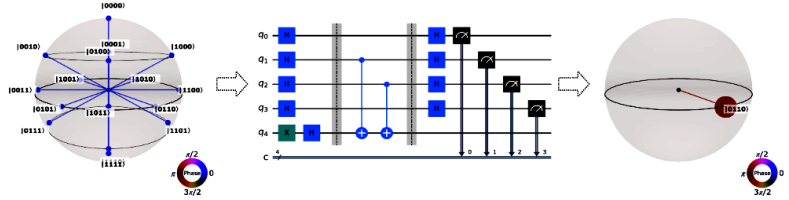
лення на основі зменшення зв'язаних кроків заміною більшою кількістю простих та однотипних команд дозволяє залучити більшу кількість процесорів, зменшуючи кількість операцій на кожен процесор, але може збільшувати загальну складність алгоритму реалізації операції. У цьому випадку треба керуватися лімітами енергоспоживання та кількістю доступних процесорів. В оцінку загальної складності необхідно включати кількість векторних операцій, що можуть зменшувати кількість операцій на порядок.

У табл. 2 надана апріорна оцінка загальної складності за кількістю однослівних операцій додавання, віднімання, порівняння, бітових операцій у разі реалізації *S*-слівних операцій у паралельній моделі обчислень, де h – кількість доданків, $s = km$ – довжина доданку в словах, m – бітів у слові [12].

Таблиця 2

Багаторозрядна операція, Алгоритм	**Загальна кількість операцій**	**Кількість операцій на процесор**	**Кількість операцій процесора (векторних операції довжиною 16)**
Додавання	$14s + 6k$	$14m + 5k + 1$	$5k + 225$
Віднімання	$11s + 6k$	$11m + 5k + 1$	$5k + 177$
Порівняння	$5s + 7k + 2$	$5m + 5k + 4$	$5k + 84$
Знаходження суми	$2sh + 14s + 6k + 1$	$2mh + 14m + 5k + 2$	$5k + 226 + 32h$

Алгоритми та програми для квантових комп'ютерів пишуться у вигляді квантових схем, які складаються з квантових операцій над квантовими даними, що зберігаються у кубітах, та класичних обчислень. Спочатку готується суперпозиція всіх можливих станів квантової системи. Далі квантова схема змінює суперпозиції компонентів. Те, що залишається після скасування відносних амплітуд і фаз вхідного стану, є результатом квантової схеми. Обчислення за схемою відбувається зліва направо. Лівий кінець має початкові квантові дані, а правий – результат квантової схеми.



Суперпозиція всіх можливих станів | *Обчислення за квантовою схемою зі зміною суперпозиції компонентів* | *Результат*

Рис. 6. *Приклад квантової схеми*

З рис. 6 можна побачити, що характеристиками складності у квантовій моделі обчислень є кількість задіяних кубітів та глибина квантової схеми, якій відповідає найбільшій кількості вентилів, які виконується на одному з кубітів.

На рис. 7 показано, що поєднання кубітів є ключовим фактором у визначенні квантового об'єму. Схема визначає, які пари кубітів можуть бути заплутані у квантовому комп'ютері. Деяке обладнання може працювати лише з парами кубітів, які розташовані поруч один з одним, інше обладнання може безпосередньо виконувати операції заплутування до будь-яких двох фізичних кубітів у системі. Повністю з'єднані кубіти можуть виконувати алгоритми ефективніше, з меншою кількістю кроків. У разі практичної реалізації на квантовій схемі квантовий об'єм буде більшим за рахунок використання більшої кількості кубітів і відповідно довших ланцюжків вентилів до виконання.

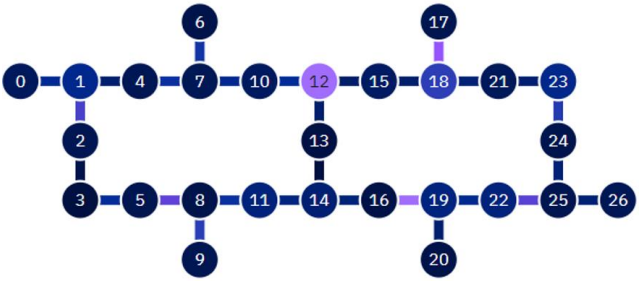


Рис. *7. Схема реалізації квантого компютера на основі 27 кубітів*

Необхідно також зазначити, що кожен квантовий комп'ютер підтримує свій набір базових вентилів і у результаті транспіляції вентилі квантової схеми розкладаються на різний набір базових вентилів для різних квантових комп'ютерів.

Як уже було зазначено раніше використання вентилів Тоффолі полегшує перенесення класичних обчислень до квантової моделі обчислень. У зв'язку з тим, що вентиль Тоффолі є універсальним, дуже зручно робити оцінку складності обчислень за кількістю вентилів. На практиці використовуються інші універсальні вентилі такі, як Фредкіна і Переса, але вентиль Тоффолі є найбільш поширеним та зручним для використання, тому частіше аналіз складності відбувається на основі вентиля Тоффолі.

У табл. 3 наведена складність обчислення операції множення у квантовій моделі обчислення [13]. Класичний метод множення «у стовпчик» потребує $4n+1$ кубітів для множення двох чисел довжиною n бітів, але глибина квантової схеми обчислення має квадратичну залежність від довжині n, що для великих n дуже впливає на час виконання та на рівень помилок. Метод Тоом–3 має кращі значення за кількістю задіяних кубітів (для невеликої кількості кубітів) та має

меншу глибину квантової схеми. Починаючи з $n = 52$, класичний метод множення «у стовпчик» починає вигравати за кількістю кубітів та вентилів Тоффолі, хоча програє за глибиною Тоффолі.

Таблиця 3

Метод множення	Кількість вентилів Тоффолі	Кількість кубітів	Глибина Тоффолі
Класичний	$4n^2 - 3n$	$4n + 1$	$4n^2 - 4n + 1$
Карацуби [14]	$42n^{1,585}$	$n^{1,427}$	$n^{1,158}$
Тоом–2.5	$49n^{1,547}$	$n^{1,404}$	$n^{1,143}$
Тоом–3	$8n^2 + 66n^{1,465} - 72n$	$n^{1,353}$	$n^{1,112}$

Наукове підґрунтя при проведенні досліджень. Стимулом для побудови ефективних алгоритмів *S*-слівної арифметики були і є наукові форуми «Питання оптимізації обчислень», які Інститут кібернетики ім. В. М. Глушкова НАН України почав проводити з 1969 року, по сьогоднішній день їх проведено 48.

З тематикою статті почала з'являтися на шпальтах цих наукових форумів з 1993 року.

Як ми вже згадували, *S*-слівна арифметика широко використовується в двоключової криптографії, для високоточних обчислень та багатьох інших класів задач.

Починаючи, з 1993 року на кожному науковому форумі в секції «Захист інформації», а з 2023 року і в секції «Квантові обчислення» присутні доповіді, присвячені оптимізації алгоритмів виконання операцій над багаторозрядними числами в різних моделях обчислень – послідовній, паралельній, квантовій та в системах залишкових класів.

Робота цих секцій стимульована і стимулює виконання робіт в цьому важливому науковому напрямку, що призвело до виходу двох монографій, а також впровадження результатів у вигляді програмних та програмно-апаратних комплексів для ГРУ МО України, СБ України, Інституту кібернетики ім. В. М. Глушкова, Люблінського політехнічного інституту (Польща).

Зараз учасники наукових форумів активно працюють над розробкою нових ефективних алгоритмів *S*-слівної арифметики для квантової моделі обчислень.

Користаючись нагодою запрошуємо фахівців, які працюють в даній тематиці, залучитись до участі в міжнародних наукових форумах «Питання оптимізації обчислень».

Наступний 49-й науковий форум-міжнародна наукова школа «Питання оптимізації обчислень» відбудеться у вересні 2025 року.

Висновки. Розглянуто шляхи оптимізації арифметичних операцій під час побудови алгоритмів у різних моделях обчислень – паралельній, паралельній, квантовій.

Під час побудови алгоритмів у послідовній моделі обчислень розглянуто шляхи оптимізації арифметичних операцій, які полягають у зменшені кількості однослівних операцій множення та ділення.

Під час побудови алгоритмів у паралельній моделі обчислень розглянуто шляхи оптимізації арифметичних операцій, а саме:

1. Зменшення зв'язаних кроків за рахунок заміни їх простішими та однотипними, але незв'язаними операціями, що зазвичай збільшує кількість задіяних процесорів, але дає можливість зменшити загальну кількість операцій, виконуваних кожним з паралельних процесорів.
2. Зменшення обсягу оброблюваних даних, що дає змогу зменшити кількість задіяних паралельних процесорів, зберігаючи загальну кількість операцій, виконувану кожним з паралельних процесорів.
3. Використання резервів оптимізації обчислень (не збільшуючи кількість кроків алгоритму).

Наведений перелік шляхів оптимізації не є вичерпним.

Розпаралелювання арифметичних операцій можна здійснити переважно тоді, коли вдається зменшити кількість сильно зв'язаних кроків, тому у роботах особлива увага приділялася першому методу оптимізації. Метод оптимізації за рахунок зменшення загальної розрядності вхідних даних використовувався для алгоритму *S*-слівного множення на основі швидкого перетворення Фур'є (ШПФ).

Під час побудови алгоритмів у квантовій моделі обчислень розглянуто шляхи оптимізації арифметичних операцій, які полягають у використанні різних реалізацій вентилів Тоффолі, у разі поєднання яких виникають повторюванні елементи, видалення яких не змінює результату схеми, але зменшує довжину ланцюжку вентилів до виконання.

Список використаних джерел:

1. Задірака В. К., Швідченко І. В. Методи боротьби з накопиченням похибки заокруглення при розв'язанні задач трансобчислювальної складності. *Кібернетика та комп'ютерні технології*. 2024. № 2. С. 47-56. URL: https://doi.org/10.34229/2707-451X.24.2.5.
2. Уилкинсон Дж. Х. Алгебраическая проблема собственных значений. Москва: Наука, 1970. 564 с.
3. Задірака В. К., Швідченко І. В. Використання похибки заокруглення в сучасних комп'ютерних технологіях. *Кібернетика та комп'ютерні технології*. 2021. № 3. С. 43-52. URL: https://doi.org/10.34229/2707-451X.21.3.4.
4. Sergienko I. V., Zadiraka V. K., Lytvyn O. M. Elements of the General Theory of Optimal Algorithms. Springer, 2021. P. 378. URL: https://doi.org/-10.1007/978-3-030-90908-6.
5. Оптимальні алгоритми обчислення інтегралів від швидкоосцилюючих функцій та їх застосування. Т. 2. Застосування / І. В. Сергієнко, В. К. Задірака, О. М. Литвин, С. С. Мельникова, О. П. Нечуйвітер. Київ: Наукова думка, 2011. 348 с.

6. Задірака В. К., Луц Л. В., Швідченко І. В. Теорія обчислень інтегралів від швидкоосцилювальних функцій. Київ: Наукова думка, 2023. 472 с.
7. Задірака В. К., Терещенко А. М. Комп'ютерна арифметика багаторозрядних чисел у послідовній та паралельній моделях обчислень. Київ: Наукова думка, 2021. 136 с.
8. Задірака В. К., Терещенко А. М., Швідченко І. В. Багаторозрядна арифметика у послідовній, паралельній та квантовій моделях обчислень. *Фізико-математичне моделювання та інформаційні технології.* 2023. Вип. 36. С. 87-91. URL: https://doi.org/10.15407/10.15407/fmmit2023.36.087.
9. Задірака В., Олексюк О. Комп'ютерна арифметика багаторозрядних чисел. Київ: Наук. думка, 2003. 263 с.
10. Анісімов А. В. Алгоритмічна теорія великих чисел. Модулярна арифметика великих чисел. Київ: Видавничий дім «Академперіодика». 2001. 153 с.
11. Amdahl G. M. Validity of the single processor approach to achieving large-scale computing capabilities. *AFIPS Conf. Proc.* 1967. Vol. 30. P. 483-485.
12. Tereshchenko A., Zadiraka V. Algorithm for calculation the carry and borrow signs in multi-digit operations in the parallel computational model. *International Journal of Computing.* 2023. Vol. 22. № 1. P. 21-28. URL: https://computingonline.net/computing/article/view/2875.
13. Larasati H. T., Awaludin A.M., Ji J. Kim H. Quantum Circuit Design of Toom 3-Way Multiplication. *Appl. Sci.* 2021. Vol. 11. P. 3752. DOI: 10.3390/app11093752.
14. Карацуба А. А., Офман Ю. П. Умножение многоразрядных чисел на автоматах. *ДАН СССР*. 1962. Вип. 145. С. 293-294.

S-WORD ARITHMETIC AND HIGH PRECISION CALCULATIONS

The intricacies of using S-word arithmetic, the influence of the value of the parameter S on the estimation of the rounding error are analyzed; what are high-precision calculations and where they are used. The problems of two-key cryptography, computer steganography and the problem of transcomputational complexity are considered as areas of application of S-word arithmetic. For the development of S-word arithmetic algorithms, sequential, parallel, quantum computing models are used, and systems of residual classes are used. The architectural features of the computer system for the implementation of an effective algorithm in various models of calculations are considered. For the parallel computing model, the importance of reducing the connected steps is indicated, which can increase the amount of processed data, but allows to involve a larger number of parallel processors. This approach is in conflict with a method that reduces the amount of processed data, and there is a need to maintain a balance between these two methods in a parallel computing model. For the quantum computing model, the connection of qubits is a key factor in determining the quantum volume. The physical scheme determines which pairs of qubits can be entangled in a quantum computer. Peculiarities of transferring algorithms to another computing model are considered. An analysis of the complexity of implementing S-word arithmetic operations in sequential, parallel, and quantum computing models is carried out. For the parallel computing model, the importance of reducing the connected steps is indicated, which can increase the

amount of processed data, but allows to involve a larger number of parallel processors. This approach is in conflict with a method that reduces the amount of processed data, and there is a need to maintain a balance between these two methods in a parallel computing model. For the quantum computing model, the connection of qubits is a key factor in determining the quantum volume. The physical scheme determines which pairs of qubits can be entangled in a quantum computer. Information is provided about the ongoing scientific forum «Calculation optimization issues», the subject of which is closely related to the topic (1969-2023).

Key words: *S-word arithmetic, high-precision calculations, sequential computing model, parallel computing model, quantum computing model, S-word operation of addition, S-word operation of subtraction, S-word operation of comparison, S-word operation of sum.*

Отримано: 30.07.2024

УДК 518.85
DOI: 10.32626/2308-5878.2024-25.82-95

Є. В. Івохін, д-р фіз.-мат. наук, професор,
К. Е. Юштін, канд. фіз-мат. наук

Київський національний університет імені Тараса Шевченка, м. Київ

ДВОЕТАПНИЙ МЕТОД РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА НА ОСНОВІ ГЕНЕТИЧНОГО АЛГОРИТМУ

Одним з основних завдань логістики є пошук найбільш ефективного маршруту в задачі комівояжера на заданій транспортній мережі, що дозволяє обслуговувати максимальну кількість споживачів, враховуючи певні критерії. У типовій задачі комівояжера критеріальною функцією найчастіше виступає довжина або тривалість маршруту. Однак, така постановка не враховує суб'єктивність оцінок тривалості переміщень за етапами, що може бути пов'язано з різними об'єктивними та суб'єктивними факторами. Розглянуто постановку задачі комівояжера з нечітко визначеною тривалістю переміщень між вузлами мережі. В основу підходу для розв'язання задачі покладено генетичний алгоритм з вдосконаленими процедурами мутації та формування різноманітності в популяціях. Для формалізації нечітких величин застосовується трапецієподібні нечіткі числа, що подаються в узагальненому випадку, який базується на застосуванні гаусовського розподілу та відповідних характеристик. На основі поєднання генетичного алгоритму та методу кластеризації Варда запропоновано двоетапну схему розв'язування оптимізаційної задачі комівояжера на заданій транспортній мережі. На першому етапі проводиться процедура кластеризації за методом Варда. На другому етапі на основі отриманого набору кластерів та знайдених оптимальних міжклас-