Терещенко Андрій Миколайович
кандидат фізико-математичних наук, Infopulse Poland

Алгоритм багаторозрядного множення на основі перетворення Мерсенна

Операція множення використовуються для реалізації криптографічних алгоритмів та є однією з операцій, які впливають на швидкодію асиметричних криптографічних операцій таких, як шифрування, дешифрування, верифікація електронного цифрового підпису. Використання швидких методів дозволяє виконувати операції з меншою складністю або на пристроях з меншим набором команд, які не мають операції з плаваючою комою. Такими методами є методи, які використовують перетворення Мерсенна.

У доповіді коротко надана історія виникнення числа Мерсенна та математиків, які брали участь у пошуку наступних чисел Мерсенна. У 2024 році було відкрите 52-е число Мерсенна, яке дорівнює $2^{136,279,841} - 1$ та має 41,024,320 десяткових знаків. 52-е число Мерсенна є найбільшим відомим простим числом. GIMPS є одним із проектів з пошуку чисел Мерсенна, який налічує біля 280,000 користувачів, які беруть участь у цьому проекті.

Пропонується розглянути основні переваги алгоритмів реалізації багаторозрядного множення на основі різних дискретних ортогональних перетворень таких, як Фур'є, косинусного, синусного, Уолша та в інших базисах, у тому числі на основі теоретико-числових перетворень (ТЧП) Мерсенна та Ферма. У доповіді наведені загальні обмеження використання дискретних перетворень (та швидких алгоритмів реалізації таких перетворень), які використовують операції одинарної, подвійної точності та операції з плаваючою комою. Зі збільшенням довжини дискретного перетворення необхідно виконувати обчислення з більшою точністю (більшою кількістю цифр після коми). Використання перетворення Мерсенна вирішує проблему накопичуваної похибки заокруглення за рахунок виконання обчислень над цілими числами.

Надано аналіз деяких методів оптимізації ТЧП та у доповіді автор звертає увагу на те, що навіть використання невеликої кількості операцій множення для заміни операцій додавання та побітового зсуву (які є основними операціями для обчислення прямого та оберненого перетворення Мерсенна) може «поглинути» ефект прискорення. Саме відсутність операцій множення, які замінюються операціями побітового зсуву, дає можливість реалізації швидкого алгоритму багаторозрядного множення.

Розглянуто алгоритм багаторозрядного множення на основі перетворення Мерсенна, де найменший примітивний корінь дорівнює 2, та наведені деякі властивості алгоритмів на основі чисел Мерсенна. Проаналізована межа алгоритму множення з точки зору максимального можливого значення множника та запропонована формула для обчислення такого максимального значення в залежності від довжини p перетворення Мерсенна. Наведено швидкий алгоритм обчислення залишку числа за модулем числа Мерсенна, використання якого значно зменшує складність алгоритму множення. Запропонована формула обчислення оберненого числа, необхідного для обчислення оберненого перетворення Мерсенна великої довжини.

Для перевірки коректності алгоритм було реалізовано на двох різних мовах програмування C# (з використанням бібліотеки програм BigInteger) та Python (з використанням бібліотеки програмування math), які підтримують операції з великими цілими числами. У доповіді наведені деякі методи підготовки великих даних та тестування результатів, що дозволяє зменшити час для реалізації та перевірки коректності алгоритму. Реалізація перетворення Мерсенна та операції модулярної редукції за модулем числа Мерсенна є одними з найпростіших реалізацій, які потребують тільки операції додавання та побітового зсуву. Перетворення Мерсенна має багато однотипних простих операцій додавання та побітового зсуву, які можуть виконуватися незалежно та паралельно, що дає

можливість розпаралелювання операцій у разі його реалізації. В цій доповіді це питання не досліджується.

Результати доповіді можуть бути використані для реалізації швидких алгоритмів багаторозрядної арифметики та для реалізації у мікросхемному виконанні та на ПЛІСах. Ведеться робота по розробці квантового алгоритму множення на основі перетворення Мерсенна.