УДК 004.31
Діденко А.Є.[1], Зеленьова І.Я.[2]
[1] студ. гр. КНТ-111м НУ «Запорізька політехніка»
[2] канд. техн. наук, доц. НУ «Запорізька політехніка»

АНАЛІЗ СПОСОБІВ ПРИСКОРЕННЯ ОПЕРАЦІЇ ДІЛЕННЯ ДВІЙКОВИХ ЧИСЕЛ В БАЗИСІ ПЛІС

Реалізовані на ПЛІС апаратні прискорювачі нейронних мереж, обчислень цифрової обробки сигналів або зображень, а також інших обчислювальних задач, часто оперують числами з рухомою комою. Необхідність розробки швидких таефективних апаратних модулів арифметичних операцій над числами з рухомою комою пояснюється тим, що використання таких модулів покращує продуктивність роботи апаратних прискорювачів. У роботі [1] було розглянуто методи підвищення ефективності множення двійкових чисел. Продовження досліджень в цьому

напрямку стосуєтьсяметодів підвищення ефективності схемного ділення двійкових чисел, що наразі розглядаютьсяданій роботі.

Операція ділення є найбільш складною для реалізації у порівнянні з операціями складання, віднімання та множення.

Звичайне ділення виконується за наступною формулою:

$$z = q \cdot d + r, \tag{1}$$

де q – частка; d– дільник; r – залишок.

Реалізація цього алгоритму передбачає покрокове виконання операцій віднімання та зсуву, що призводить до стандартного та найпростішого послідовного алгоритму ділення. Якщо представити q та d у вигляді k-бітних двійкових чисел, а число z як $2k$-бітне, то послідовний або так званий «bit-at-a-time» алгоритм ділення на кожному кроці j виконує зберігання часткового залишку $s^{(j)}$ і відніманні від нього значення $q_{k-j}d$, що зсунуте задля вирівнювання. Даний алгоритм може бути представлений наступною формулою [2]:

$$s^{(j)}=2s^{(j-1)}-q_{k-j}(2^k d), \tag{2}$$

де $s^{(j)}$– частковий залишок; $s^{(0)}=z$.

Проаналізувавши звичайний алгоритм ділення, що представлений формулою (2), можна побачити, що час виконання операції ділення залежить від довжини операндів: чим довше ділене, тим більше кроків необхідно виконати. У синхронних схемах це призводить до великої кількості тактів.

Основна причина того, що ділення є повільним процесом, полягає в необхідності завершення j-го кроку алгоритму перед початком виконання j+1-го. Процес множення легко може бути прискорений за рахунок одночасного обчислення часткових добутків, оскільки вони не впливають добутків, оскільки вони не впливають однин на одного. Однак, при діленні результати кроків залежать один від одного і не можна розпочати наступний крок до того, як розраховано залишок та визначено біт частки [3].

Можна визначити чотири основні методи підвищення швидкості виконання операції ділення [4]:

– завчасна зупинка обчислення: виконується здогадка про значення біту частки і, якщо вона неправильна, то подальші обчислення припиняються, оскільки окремий біт може приймати тільки два значення;

– відсутність відновлення залишку: якщо здогадка про значення біту частки виявилася хибною, то значення часткового залишкустає від'ємним. Відсутність відновлення часткового залишку означає, що подальші операції можуть виконуватися і з від'ємним залишком;

– використання кратного дільника: якщо використовувати не сам дільник, а кратне йому число, то можна за один раз знайти значення декількох бітів частки;

– нормалізація: обчислення з використанням дільника і залишку у нормалізованому вигляді дозволяє визначити декілька бітів частки без виконання арифметичних дій.

Завчасна зупинка обчислення досягається наявністю асинхронного суматора, завдяки якому постійно спостерігається значення знакового біту часткового залишку. У випадку коли знак дорівнює одиниці, тобто результат від'ємний, відбувається лише зсув, а операція віднімання пропускається.

У покроковому алгоритмі ділення перед відніманням дільника припускається, що відповідний біт частки дорівнює одиниці. Якщо при цьому різниця від'ємна, то відповідний біт частки насправді дорівнює нулю і необхідно виконати відновлення часткового залишку. Даний алгоритм називається алгоритмом з відновленням.

Відсутність відновлення полягає у продовженні ділення незалежно від того, чи стало значення часткового залишку від'ємним. У випадку, якщо різниця від'ємна, наступний біт частки приймається рівним нулю та замість віднімання дільника виконується його додавання. Таким чином зменшується кількість кроків, оскільки нема необхідності у відновленні часткового залишку.

Ділення з використанням кратного дільника, або дільника з основою *r*описується наступною формулою [2]:

$$s^{(j)} = rs^{(j-1)} - q_{k-j}(r^k d), \tag{3}$$

Ділення двійкових чисел за основою 2^b зменшує кількість операційних тактів у b разів, але очевидно збільшує складність виконання кожного такту.

Нормалізація діленого та дільника подібна до нормалізації чисел з рухомою комою, де число не повинно мати нулів після коми. У цьому методі ділене та дільник представляються у вигляді чисел з фіксованою комою. При нормалізації дільника виконується зсув вліво на кількість нулів, що стоять після коми. При цьому виконується зсув частки вліво на таку ж кількість бітів, при цьому у молодші біті зсувається нулі. Якщо після виконання віднімання дільника від діленого останній став від'ємним, то відбувається зсув діленого на кількість одиниць, що стоять після коми. При цьому також виконується зсув частки на таку ж кількість бітів, але до молодших бітів зсуваються одиниці.Таким чином зменшується кількість операцій віднімання.

Ідею нормалізації використовує алгоритм ділення SRT. Перед початком обчислення дільник нормалізується, щоб його значення належало інтервалу [1/2; 1]. Таким чином, порівняння часткового залишку з дільником

відбувається швидше, оскільки не потрібно порівнювати між собою всі біти. Двійковий дріб більший чи дорівнює 1/2 лише у тому випадку, якщо починається з 0.1 та менший за -1/2, якщо починається з 1.0. Алгоритм SRT описується формулою (4) [3].

$$q_j = \begin{cases} 1, & якщо \quad 2r_{j-1} > 1/2 \\ 0, & якщо \quad -1/2 < 2r_{j-1} < 1/2 \\ \bar{1}, & якщо \quad 2r_{j-1} < -1/2 \end{cases} \tag{4}$$

Отже, прискорення операції ділення двійкових чисел полягає у зменшенні кількості арифметичних операцій у кожному кроці алгоритму, що скорочує часову затримку одного такту, або зменшенні кількості кроків, що дозволяє виконати обчислення за меншу кількість тактів.

Отримані в даній роботі результати аналізу методів підвищення швидкості операції ділення та результати роботи [1] надалі можуть бути використані для побудови та реалізації в ПЛІС ефективного прискорювача арифметичних операцій з числами з рухомою комою.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1.Діденко А.Є. Аналіз методів підвищення ефективності множення двійкових чисел / Діденко А.Є., Зеленьова І.Я. // Тиждень науки НУ «Запорізька політехніка». – 2021. – С. 28-30.

2. Parhami, B. Computer Arithmetic: Algorithms and Hardware Designs, 2nd edition / B. Parhami. – New York : Oxford University Press, 2010. – 672 p.

3. Koren, I. ComputerArithmeticAlgorithms 2nd Edition / I. Koren. – Natick : A K Peters/CRC Press, 2001. – 296p. – ISBN 978-1568811604.

4. Flores I. Thelogicofcomputerarithmetic / I. Flores – EnglewoodCliffs, N. J. :Prentice-Hall, Inc., 1963. – 493 p.