

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ОПЕРАЦІЇ ДІЛЕННЯ НА ПРОГРАМОВНО-ЛОГІЧНІЙ ІНТЕРГАЛЬНІЙ СХЕМІ

Розроблено модулі ділення в елементному базисі програмовно-логічній інтегральній схемі (ПЛІС) типу FPGA, які виконано за допомогою поведінкового опису алгоритмів мовою VHDL. Виконано перевірку

функціонування модулів ділення методом моделювання в системі ModelSim Xilinx Edition — MХЕ ІІІ за допомогою поведінкового стенда.

The dividers blocs on PLD base of type of FPGA were developed. They were realized by behavior description on VHDL. The functioning of dividers blocs was checked by simulation method in ModelSim Xilinx Edition — MХЕ ІІІ and behavior stand.

Вступ

Проектування і реалізацію цифрових пристроїв на сучасній елементній базі може бути виконано на базі ПЛІС типу FPGA. Використання HDL-технології, яка є комплексом інструментальних засобів САПР і методологією проектування, орієнтованих на опис проекту за допомогою мов VHDL і Verilog [1], дає змогу в необхідні терміни розробити і реалізувати пристрій, використовуючи мінімум устаткування.

Найважливішою властивістю готового технічного рішення (IP-core) є його гарантоване відтворення в новому проекті відповідно до специфікації, визначеної розробником цього рішення. Слід зазначити, що опис моделі за допомогою HDL-технології (*Hardware Description Language* — мова опису апаратних засобів) допомагає не тільки зробити її перенастроюваною і незалежною від технології, а й і виконувати її моделювання та синтез з використанням інструментальних засобів різних фірм.

Постановка проблеми

Операція ділення, найскладніша із списку базових арифметичних операцій, може бути реалізована різними алгоритмами. Актуальними залишаються питання оптимізації алгоритмів [2], використання спеціальних алгоритмів прискореного ділення, а також їх апаратних реалізацій. Операція ділення має такі шляхи оптимізації за швидкодією:

- за рахунок заміни знаменника зворотною величиною з подальшим множенням її на чисельник;
- за рахунок скорочення часу виконання операції додавання / віднімання;
- за рахунок зменшення кількості операцій додавання / віднімання, при розрахунку частки;
- обчислення частки в надмірній системі числення.

Одним із способів організації пристрою може бути використання конструкції *Case* (машини станів) [3]. Принцип конструкції полягає в тому, що в кожному стані виконуються задані операції і визначається перехід до наступного стану. Операції в кожному стані виконуються за новим синхроімпульсом. Такий підхід використовується у тому випадку, коли потрібно зменшити тривалість синхроімпульсу, що приводить до збільшення апаратних витрат. В конструкції *Case* легко організувати будь—які цикли, у тому числі зі змінною кількістю ітерацій, що не завжди можливо в конструкції *For*, в якій використовується обмежена кількість ітерацій.

На основі вказаних властивостей розглянемо реалізацію модулів ділення, поведінковий опис яких використовує конструкцію *Case*.

Реалізація модулів ділення

Розглянемо реалізацію модулів ділення в елементному базисі ПЛІС типу FPGA фірми Xilinx. Модуль ділення містить чотири входи: $a(32:0)$ — чисельник; $b(16:0)$ — знаменник; Load — сигнал завантаження операндів (стан «1» відповідає операції завантаження); Clk — послідовність синхроімпульсів, і два виходи: $c(16:0)$ — частка; RG_FLAG(7:0) — регістр прапорів [4].

Алгоритми, що використовуються:

Ділення з нерухомим чисельником і зсувом управо знаменником. Операція віднімання застосовується для 32 біт модуля чисельника і лише тоді, коли чисельник більше знаменника, в цьому випадку в біт частки записується «1», в іншому випадку «0». Спочатку знаменник формується як '0' & b & «0000000000000000» (де b модуль знаменника). Зсув знаменника відбувається на кожній ітерації.

Ділення з нерухомим знаменником і відновленням залишку. Чисельник зсувається вліво на кожній ітерації. Віднімання знаменника з чисельника відбувається у тому випадку, коли старша

частина чисельника $a(31 \text{ downto } 16)$ більше знаменника і результат записується в старшу частину знаменника. Головна відмінність від першого алгоритму полягає в тому, що операція віднімання проводиться тільки над 16 бітами, і зсув відбувається над чисельником уліво.

Ділення з нерухомим знаменником без відновлення залишку. Ідея алгоритму полягає в тому, що над старшою частиною чисельника зі знаком (17 біт) і знаменником зі знаком може виконувати як додавання, так і віднімання. Операція визначається відповідно до знака результату, що сформувався в попередній операції. У разі додавання в біт частки записується «1», віднімання — «0».

Метод Гарвардської ітерації є одним із ефективних алгоритмів ділення. Цей алгоритм часто використовується для реалізації на апаратних засобах, оскільки основні дії цього алгоритму можна розбити на більш простіші складові, що дає змогу окремо їх реалізувати, після чого об'єднати в одне ціле. Даний алгоритм може бути застосований при знаходженні кореня квадратного. Універсальність методу в тому, що під час його реалізації на апаратному і програмному рівні виконується одна й та сама кількість ітерацій.

Алгоритм Гарвардської ітерації призначений для ділення чисел з плаваючою точкою. В даній праці він був модифікований і використовується для ділення цілих чисел. Складність модифікації в тому, що на кожній ітерації є два числа з плаваючою точкою, відповідно з'являється необхідність їх перетворення.

Алгоритм забезпечує квадратичну збіжність результату. На кожній ітерації множник R одночасно перемножується на чисельник і знаменник. У результаті цих дій знаменник квадратично збігається до одиниці, а чисельник — до частки.

Оскільки має місце квадратична збіжність і кожна ітерація подвоює кількість одиниць після коми в знаменнику, то новий знаменник матиме вигляд $0.11xxx\dots$. У даному методі множення відбувається доти, поки знаменник стане еквівалентним одиниці ($0,1111\dots111$).

Алгоритм ділення за методом Гарвардської ітерації :

1. Нормалізація знаменника і відповідно зсув чисельника.
2. Визначення першого множника R .
3. Множення Y на R , підраховуємо $Y1$.
4. Множення X на R , підраховуємо $X1$.
5. Визначення зайвих бітів з проміжних результатів та занесення тимчасових результатів до початкових X та Y .
6. Повернення до пункту 3.

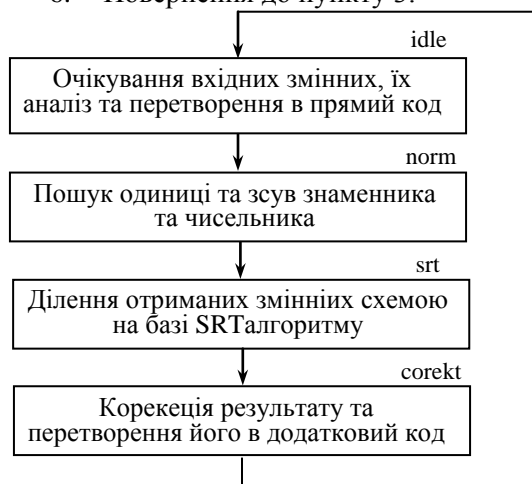


Рис. 1. Алгоритм ділення на базі SRT-алгоритму

SRT-алгоритм є модифікацією алгоритму ділення без відновлення залишку. Відмінність полягає у відсутності необхідності додавання або віднімання на кожній ітерації залежно від значення, що вийшло, в регістрі чисельника. Частка відображається двома регістрами і формується від старшого біта до молодшого порозрядно: «0» — відповідає комбінації «00», коли три біти в регістрі чисельника співпадають; «1» — відповідає комбінації «10», коли три старші біти в регістрі чисельника не співпадають і чисельник позитивний; «-1» — відповідає «01», коли три старші біти в регістрі чисельника не збігаються і чисельник негативний. Частка формується відніманням з регістра 1 значення в регістрі 2 і корекцією результату. Алгоритм роботи зображено на рис. 1.

Запропонована реалізація модуля ділення чисел з фіксованою точкою виконана засобами поведінкового опису SRT-алгоритму мовою VHDL. Архітектура модуля представлена машиною станів, що має чотири стани. В стані «idle» виконується аналіз вхідного операнда і формування додаткових векторів, в «norm» — пошук старшої одиниці знаменника і «нормалізація» знаменника та зсув чисельника, в «srt» — безпосередньо операція ділення, в «corekt» — корекція частки, залишку і формування вихідного результату.

У таблиці наведені дані про апаратні ресурси, максимальну швидкодію розроблених модулів залежно від алгоритму.

Верифікація модулів ділення

Перевірка функціонування модулів здійснена методом моделювання в системі *ModelSim Xilinx Edition* — MXE III.

На рис. 2 зображено схему стенда, виконану засобами схематичного редактора *Engineering Capture System (ECS)*, що входить до складу системи *Xilinx ISE Foundation*.

ОЦІНКИ АПАРАТНИХ РЕСУРСІВ ТА МАКСИМАЛЬНОЇ ШВИДКОДІЇ МОДУЛІВ ДІЛЕННЯ

| Оцінка апаратних ресурсів та швидкодії | (Slices) | Синхроімпульс Clk, період (нс) |
|--|-----------|--------------------------------|
| Ділення з нерухомим чисельником і зсовуванням вправо знаменником | 251 (7%) | 11,5 |
| Ділення з нерухомим знаменником і відновленням залишку | 222 (6%) | 9,8 |
| Ділення з нерухомим знаменником без відновлення залишку | 221 (6%) | 10,8 |
| Ділення на базі SRT-алгоритму | 410 (11%) | 12,6 |
| Ділення на базі методу гарвардської ітерації | 520 (15%) | 25 |

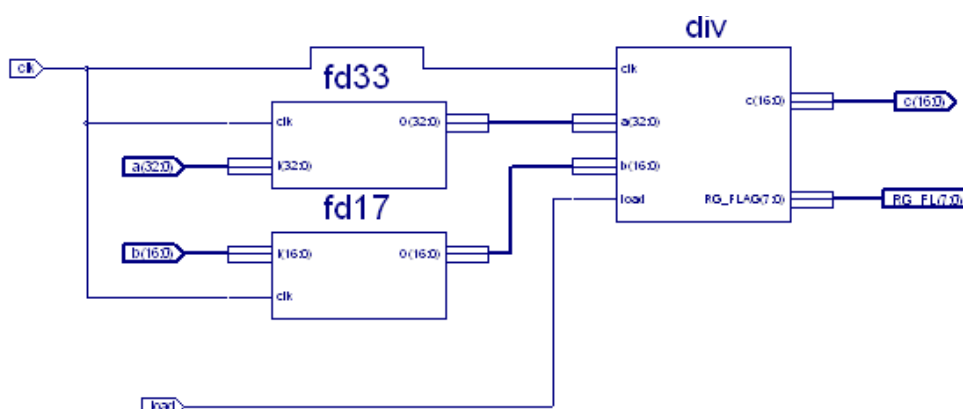


Рис. 2. Стенд для моделювання

Висновки

Модулі розроблені за допомогою інструментальних засобів *Xilinx Foundation ISE 9.2*, протестовані моделюванням в системі *ModelSim Xilinx Edition-III* за допомогою стенду для перевірки, що підтверджує правильність функціонування. Проект реалізований на ПЛІС *Spartan 3 XC3X400*.

На базі аналізу отриманих результатів (див. таблицю) встановлено, що з-поміж розглянутих варіантів реалізацій алгоритмів ділення, оптимальним за критерієм ресурси—швидкодія є алгоритм ділення з нерухомим знаменником і відновленням залишку.

Отримані результати будуть використовуватись при розробці складних обчислювальних пристроїв з використанням типових технічних рішень (у вигляді бібліотечних елементів IP—Core) як складові компоненти для реалізації таких пристроїв на одному кристалі. Такі типові рішення в сукупності є «відкритою бібліотекою» файлів конфігурацій, що входять до складу реконфігуровної обчислювальної системи з віртуальною архітектурою.

ЛІТЕРАТУРА

1. Семенец В. В., Хаханова І. В., Хаханов В. І. Проектирование цифровых систем с использованием языка VHDL. — Харьков : ХНУРЕ, 2003. — 492 с.
2. Карцев М. А. Арифметика цифровых машин. — М. : Наука, 1969. — 576 с.
3. VHDL'93. IEEE Standard VHDL Language Reference Manual. IEEE Std 1076-1993.
4. Опанасенко В. Н., Сахарин В. Г., Лисовый А. Н. Проектирование модулей с плавающей точкой на ПЛИС с использованием языка VHDL // Математические машины и системы 2005. — № 3. — 195 с.