УДК 004.3

ЛУЦЬКИЙ Г. М.,
БЛІНОВА Т.О.,
ПОРЄВ В.М.

ВИКОРИСТАННЯ ДЕЯКИХ СПОСОБІВ КОДУВАННЯ ДЛЯ ПІДВИЩЕННЯ ТОЧНОСТІ ОБЧИСЛЕНЬ

При здійсненні в обчислювальних системах операцій, особливо векторних, над числами з плаваючою точкою дуже важливі питання точності. Описується алгоритм роботи обчислювального пристрою, орієнтованого на виконання над числами із плаваючою точкою операції послідовності додавання парних добутків. Для забезпечення підвищеною точністю проміжні результати виконання арифметичних операцій представлені в системі числення зі збільшеною основою, причому цифри представлені двійковим кодом. Це дає можливість отримувати результати обчислень, що містять, у загальному випадку, тільки помилку, що виникає через представлення мантиси кінцевим числом розрядів

At realization in the computer systems of operations, especially vector operations, on floating point numbers the questions of exactness are very important. The work algorithm of floating point number computing device, which oriented to performance of add sequence of paired products, is described. For providing enhanceable exactness the intermediate arithmetic results represent in system with scaled-up basis and with digits are represented by binary code. It enables to get the calculation results which contain, in general case, only error which arises up through mantissa represent of a finite number of digits.

Через обмежену, навіть дуже велику, розрядність цифрових обчислювальних машин, питання точності дуже важливі, особливо для обчислень із плаваючою точкою.

Обчислення над числами у формі з плаваючою точкою неточні по своїй природі, можуть спостерігатися порушення закону асоціативності, закону дистрибутивності, що пов'язує операції × та +: $u \times (v + w) \neq (u \times v) + (u \times w)$ тощо. Віднімання майже рівних величин, наприклад, може значно збільшити відносну похибку. Вважається, що для виконання арифметичних дій над числами із плаваючою точкою подвійна точність необхідна майже завжди (у відмінність від фіксованої точки). Для знаходження, наприклад, точного співвідношення для додавання величин із плаваючою крапкою, потрібно виконати сім операцій додавання [1].

Виконання рекомендації по зменшенню похибок, зокрема, попереднього сортування чисел за розміром, рутинного перетворення формул і т.п. програмістом або транслятором утруднюється через ряд причин, наприклад, через незнання до початку роботи програми можливих значень змінних. Забезпечення цих правил, наприклад, зміна порядку подачі операндів, створює труднощі при реалізації паралельних обчислень.

У роботі [2] показано, що при виконанні векторних операцій особливу увагу необхідно приділяти точності операцій, які пов'язані з виконанням послідовності операцій алгебраїчного додавання, а саме: віднімання-додавання, додавання парних добутків тощо.

Отже, уявляються актуальними дослідження, спрямовані на реалізацію виконання арифметичних операцій над числами з плаваючої точкою, що забезпечують зберігання максимально можливого числа стандартних математичних законів.

У роботах [3,4] описаний спосіб додавання послідовності чисел із плаваючої точкою, що дозволяє одержати результат обчислень із заданою точністю. При цьому передбачається, що на вході пристрою кожне число x із плаваючою точкою має вигляд:

$$x = M_x \cdot 2^{Px}, \qquad (1)$$

де M_x – n-розрядна нормалізована ($\frac{1}{2} \le |M_x| < 1$, при $x \neq 0$) дробова частина числа x (мантиса); Px – порядок числа (ціле число з інтервалу $[Pmax, Pmin]$); M_x та Px представлено додатковим двійковим кодом.

Відображення довільного дійсного числа у вигляді (1) містить, у загальному випадку, помилку, що виникає через представлення мантиси кінцевим числом розрядів. За винятком цієї помилки, результат додавання послідовності чисел може бути отриманий із заданою точністю, що досягається через відсутність нормалізації й округлення проміжних сум, а також унаслідок того, що молодші розряди

мантис доданків при вирівнюванні їхніх порядків не втрачаються.

Для зменшення сумарних апаратурних витрат запропоновано використовувати для представлення чисел у систем числення з основою більше двох.

Через те, що при розробці пристроїв обчислювальної техніки рідко використовують багатопозиційні цифрові елементи використовуються системи числення з основами 2^k (k – ціле позитивне число), і кодування кожної цифри цих систем числення визначеною комбінацією станів декількох двопозиційних елементів. Причому, як показують дослідження, приведені в книзі, шуканий мінімум апаратурних витрат досягається при $k=n$-1, тобто при $\varepsilon = 2^{n-1}$.

Вибір таких систем числення обумовлений тим, що вони є найбільше економічними з погляду необхідного устаткування для кодування цифр за допомогою двопозиційних елементів.

Іншою важливою властивістю систем числення з основою 2^k є той факт, що для перекладу числа з фіксованою точкою з однієї системи числення в іншу не потрібно ніяких дій. У зв'язку з цим, видача операндів на входи перетворювача інформації, що працює в системі числення з основою 2^k і зчитування результату з його виходів може супроводжуватися мінімальними перетвореннями.

Таким чином, вираз (1) для довільного числа x с плаваючої точкою у системі числення з такою основою ε буде мати наступний вигляд:

$$x = M_x(\varepsilon)\cdot \varepsilon^{Px(\varepsilon)} = M_x\, 2^{Px \bmod (n-1)} \varepsilon^{\lfloor Px/(n-1)\rfloor}.$$

Зазначимо, що тут і далі $\lceil X \rceil$ – найменше ціле число, що більше або дорівнює X; $\lfloor X \rfloor$ – найбільше ціле число, що менше або дорівнює X; $X \bmod Y$ – залишок від ділення X на Y.

Отже, для переводу довільного числа з плаваючою точкою з двійкової системи числення у систему числення з основою $\varepsilon = 2^{n-1}$ необхідно обчислити $P_x \bmod (n\text{-}1)$ і відповідно зсунути мантису, формуючи дві цифри числа у системі обчислення з основою ε, а також обчислити $\lfloor Px/(n\text{-}1)\rfloor$, щоби визначити, на які позиції числа, представленого у цій системі числення, поставити отримані дві цифри (інші цифри числа дорівнюватимуть нулю).

Аналогічно, для зворотного переводу довільного числа x із плаваючою точкою з системи числення з основою $\varepsilon = 2^{n-1}$ у двійкову систему числення необхідно обчислити, скільки нульових старших цифр у числі, це і буде величина $\lfloor Px/(n\text{-}1) \rfloor$, потім відповідно зсунути уліво число на $\lfloor Px/(n\text{-}1) \rfloor$ позицій; проаналізувати двійковий код старшої цифри зсунутого числа, визначаючи, скільки нульових (уже двійкових) розрядів вона має, це і буде $P_x \bmod (n\text{-}1)$; потім зсунути уліво на $P_x \bmod (n\text{-}1)$ розрядів (двійкових) усе число, формуючи тим самим нормалізовану мантису M_x числа x, представленого у двійковій системі числення, а також обчислити порядок $P_x = (P_x \bmod (n\text{-}1)) + (n\text{-}1)\lfloor Px/(n\text{-}1)\rfloor$ нормалізованого числа x.

Роботи [5,6] присвячені побудові та моделюванню конвеєрного суматора чисел із плаваючою точкою, що забезпечує підвищену точність, робота [7] присвячена побудові пристрою для додавання послідовності добутків пар чисел із плаваючою точкою.

Мета даної статті – надати опис алгоритму роботи розробленого пристрою для виконання векторної операції додавання послідовності добутків пар чисел із плаваючою точкою.

Пристрій для виконання векторної операції додавання послідовності добутків пар чисел із плаваючою точкою (ПДПД) містить блок множення, блок керування, блок підсумовування, блок формування результату (рис.1). Функціональна схема ПДПД представлена на рис.2.

При обчисленні суми послідовності добутків пар чисел на вхід блока множення на кожному такті роботи ПДПД подаються представлені у прямому коді n-розрядні нормалізовані мантиси обох співмножників (у регістри РгМ1 та РгМ2), їхні порядки (у регістри РгП1 та РгП2) та знаки (у тригери Тг31 та Тг32).

Нехай на i-му такті роботи на вхід пристрою подано i-у пару співмножників і у блоці множення відбувається обчислення n-розрядного добутку цих співмножників (за допомогою вузла Пам'ять добутків), порядок (за допомогою суматора) та знак (за допомогою логічного вузла ЛогВ, що обчислює функцію нерівнозначності).

На $(i+1)$-му такті роботи у блоці керування відбувається перетворення цього добутку (далі він виступає у ролі доданка) із двійкової системи обчислення у систему обчислення з основою $\varepsilon = 2^{n-1}$.

А саме, з приходом переднього фронту тактового імпульсу мантиса M_x доданка (добутку), що представлений в прямому коді, порядок P_x доданка (добутку), знак доданку (добутку) записуються відповідно у регістри мантиси

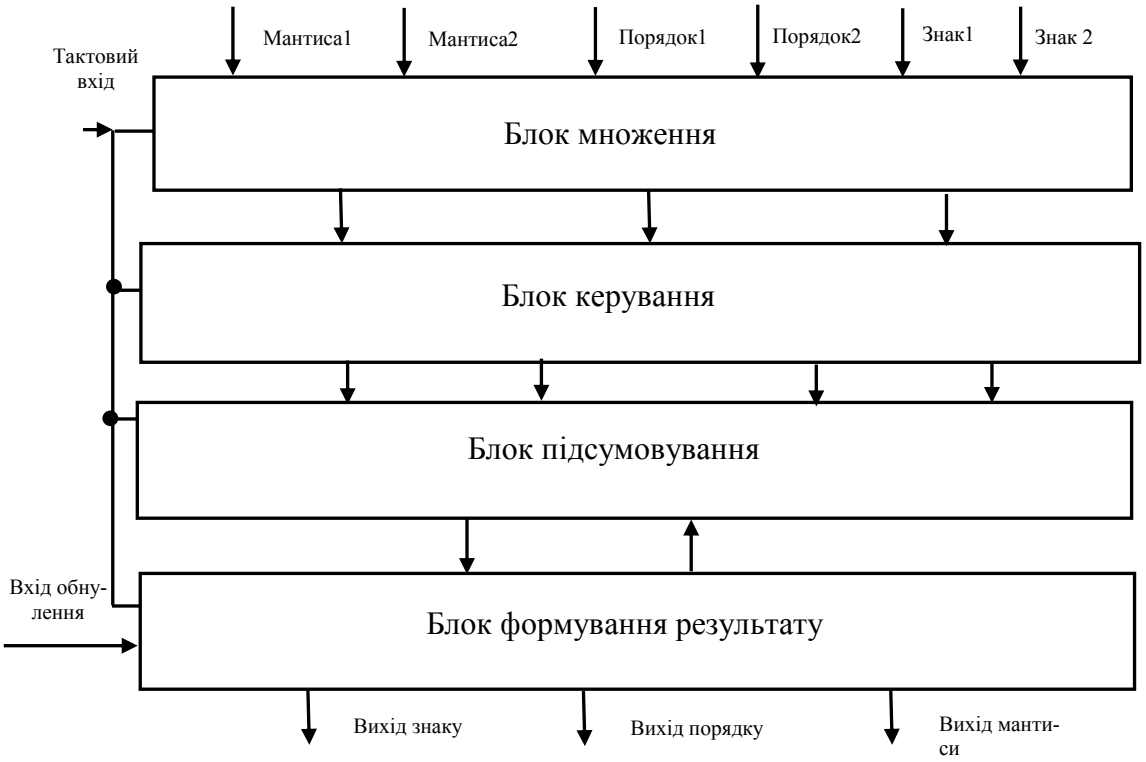

Рис.1. Структурна схема пристрою

(РгМ) та порядку (РгП) і тригер керування (ТгК).

За допомогою вузла Пам'ять 1 визначається $P_x \bmod (n\text{-}1)$, і відповідно на $P_x \bmod (n\text{-}1)$ розрядів зсовується мантиса M_x, формуючи дві цифри числа $M_x(\varepsilon)$ у системі обчислення з основою ε, а також обчислюється $\lfloor Px /(n\text{-}1) \rfloor$, щоби визначити, на які позиції числа $M_x(\varepsilon)$ поставити отримані дві цифри (інші цифри числа дорівнюватимуть нулю). Отже, на перший вихід вузла Пам'ять 1 подається цифра числа $M_x(\varepsilon)$, що повинна потрапити на якусь непарну позицію, а на другий вихід вузла Пам'ять 1 подається цифра числа $M_x(\varepsilon)$, що повинна потрапити на якусь парну позицію числа $M_x(\varepsilon)$. Третій вихід вузла Пам'ять 1 визначає, на які саме позиції будуть записані нулі, а отже у ті позиції, що залишилися, будуть записані ці дві цифри.

На $(i+2)$-му такті роботи у блоці підсумовування відбувається перевід мантиси $M_x(\varepsilon)$ у доповняльний код та підсумовування її з накопиченою сумою, що по ланцюгу зворотного зв'язку поступає з регістра блока формування результату (на початку обчислення нової послідовності ця накопичена сума дорівнює нулю, тому що на $(i+2)$-му такті роботи регістр був встановлений у нуль).

Це здійснюється наступним чином. З приходом $(i+2)$-го тактового імпульсу по його передньому фронту відбувається запис числа $M_x(\varepsilon)$ у $(n\text{-}1)\bullet m$- розрядний регістр мантиси РгММ, де n – розрядність мантиси, $m = \left\lceil \frac{P_{макс}}{n-1} \right\rceil + 1$, а $P_{макс}$ – максимальний порядок доданків. Перевід мантиси $M_x(\varepsilon)$ у доповняльний код проходить два етапи. Спочатку комутатор формує обернений код числа (інвертує число, якщо знак доданка дорівнює одиниці), потім при виконанні підсумовування на вхід переносу в молодший розряд суматора мантиси подається знак доданка.

На $(i+3)$-му такті роботи у блоці формування результату відбувається перевід накопиченої суми у прямий код, перевід її у двійкову систему числення та починається формування нормалізованого n - розрядного результату.

Перевід результату $M_x(\varepsilon)$ у прямий код здійснюється у два етапи. Спочатку комутатор результату інвертує число, якщо знак результату дорівнює 1, потім при виконанні підсумовування на вхід переносу в молодший розряд суматора результату подається знак доданка, а на другу групу входів суматора результату подаються сигнали "Логічний нуль".

Формування нормалізованого n- розрядного результату теж проходить у декілька етапів. Спочатку за допомогою першої групи елементів "АБО" із $m\ (n\text{-}1)$- входових елементів "АБО", виявляються всі нульові $(n\text{-}1)$- розрядні

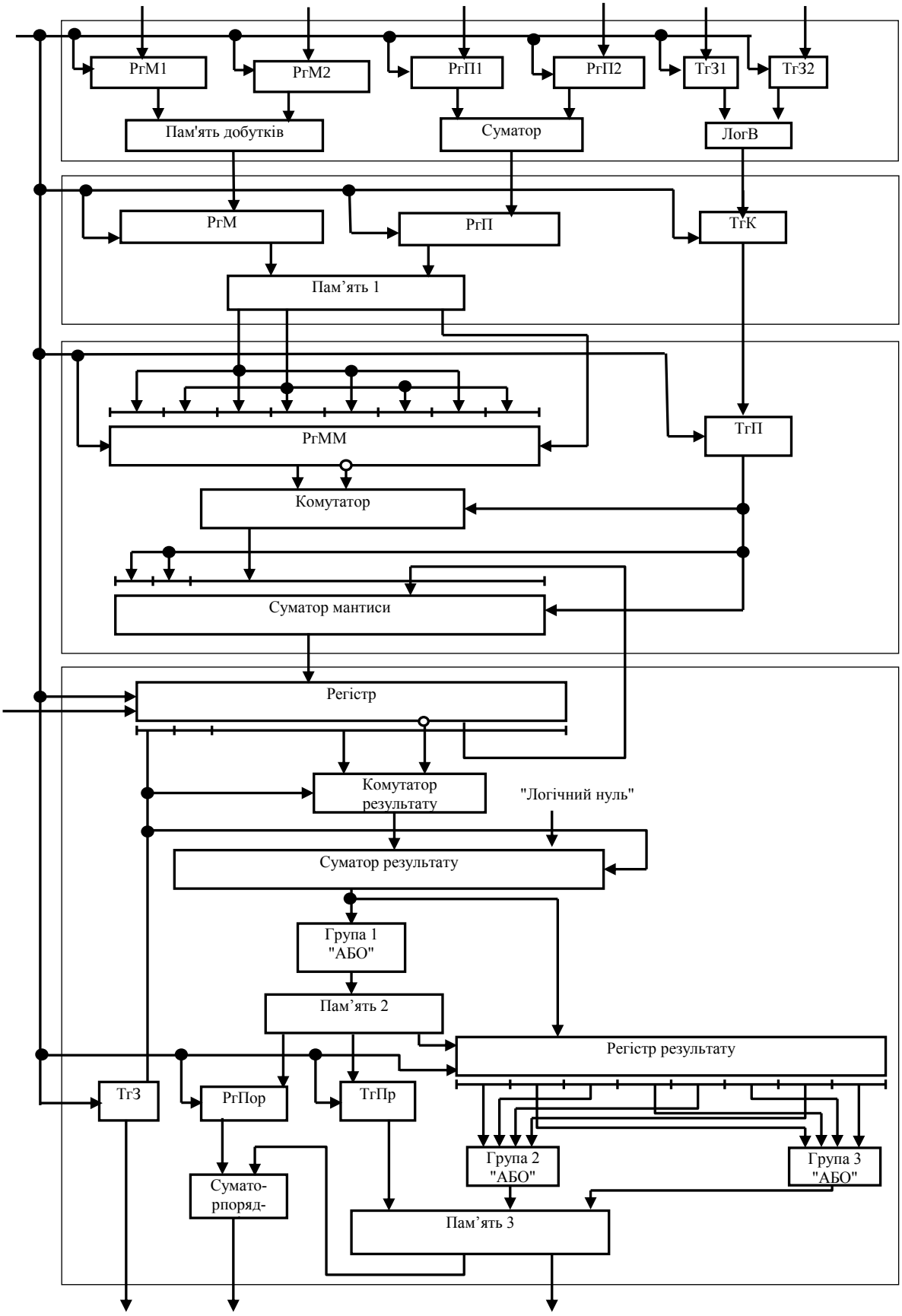

Рис.2. Функціональна схема пристрою

цифри результату. Потім у вузлі Пам'ять2 визначається, скільки старших цифр результату дорівнює нулю, і формується $((n\text{-}1)\lfloor Px/(n\text{-}1)\rfloor)$ – частина порядку числа. Крім того, у вузлі Пам'ять 2 формуються сигнал перестановки та сигнали встановлення в нуль регістра результату.

На $(i+4)$-му такті роботи у блоці формування результату продовжується перевід результату у двійкову систему числення та формування нормалізованого n - розрядного результату.

По передньому фронту тактового сигналу у регістр результату записуються тільки дві старші $(n\text{-}1)$-розрядні групи ненульових розрядів, позаяк інші находилися під впливом сигналів встановлення в нуль. За допомогою другої (з $\lceil 0{,}5m\rceil$ $(n\text{-}1)$-входових елементів "АБО") та третьої $((m\text{-}\lceil 0{,}5m\rceil)$ $(n\text{-}1)$-входових елементів "АБО") груп елементів "АБО" їх відокремлено, тим самим відокремлено дві старші цифри мантиси, представленої у системі числення з основою ε. Перша відокремлена цифра може бути як на парній позиції, так і на непарній позиції. Отже, коли на другий адресний вхід вузла Пам'ять 3 поступає непарна цифра, вона може бути молодшою. Для того, щоб знати це, на перший адресний вхід вузла Пам'ять 3 подається сигнал перестановки із тригеру перестановки ТгПР, він дорівнює одиниці, якщо потрібно поміняти місцями старшу і молодшу цифри. Таким чином формується $2(n\text{-}1)$-розрядне двійкове число. За допомогою вузла Пам'ять 3 на основі аналізу того, скільки нульових (уже двійкових) старших розрядів міститься у цьому числі, визначається $P_x \bmod (n\text{-}1)$, це число на $P_x \bmod (n\text{-}1)$ розрядів зсовуються уліво і формується n- розрядний результат, відкидаючи, якщо це потрібно зайві розряди). Частина порядку $P_x \bmod (n\text{-}1)$ результату з першого виходу вузла Пам'ять 3 поступає в суматор порядку, де підсумовується з частиною порядку $((n\text{-}1)\lfloor Px/(n\text{-}1)\rfloor)$. Отже, на $(i+4)$-му такті формується знак суми, порядок та нормалізована мантиса результату.

Таким чином, для підсумовування послідовності добутків із l пар чисел із плаваючою крапкою потрібно $(l+4)$ тактів роботи пристрою. Перед початком обчислення нової послідовності чисел має бути хоча б один холостий такт.

ПДПД легкий у керуванні, тому що виконує обчислення за передбачувану кількість тактів, що залежить тільки від числа доданків.

ПДПД створено з урахуванням вимог IEEE Стандарту на двійкову арифметику із плаваючою точкою (Р754) та Стандарту на арифметику із плаваючою точкою з різними основами системи числення й словами довільної довжини (Р854). Вхідні операнди та остаточний результат нормалізовано й представлено у прямому коді, порядок чисел зміщений. У самому ПДПД виконуються операції з підвищеною точністю. На виходах пристрою формуються результати, що знову представлені у відповідності зі Стандартами Р754 і Р854.

Запропонований пристрій для виконання з підвищеною точністю векторної операції підсумовування послідовності добутків пар чисел із плаваючою точкою може без змін використовуватися, як закінчений вузол, у процесорах.

Список літератури

1. Кнут. Д. Искусство программирования для ЭВМ. В семи томах. Т.2. Получисленные алгоритмы, пер. с англ. − М.: Мир, 1977. − 724 с.
2. Луцький Г.М., Блінова Т.О.Долголенко О.М Пристрій для виконання з підвищеною точністю векторних операцій над числами зі плаваючою точкою //Вісник НТУУ "КПІ". Сер. Інформатика, управління та обчислювальна техніка. – 2002. № 37– С.130-135.
3. АС СССР № 849205. Конвейерное устройство для выполнения арифметических операций над множеством чисел /Вилкул М.А., Долголенко А.Н.,Луцкий Г.М. –1981. Бюл. № 27
4. Самофалов К.Г. Луцкий Г.М. "Основы теории многоуровневых конвейерных вычислительных систем". М.: Радио и связь. – 1989. -272 с.
5. Патент України №50428А, Пристрій для додавання послідовності чисел із плаваючою точкою / Блінова Т.О., Долголенко О.М., Луцький Г.М., Порєв В.М.– 2002. Бюл. № 10.
6. Аксьоненко С.В., Долголенко О.М. VHDL-реалізація конвеєрного пристрою для матричних обчислень високих порядків на FPGA, що реконфігуруються //Вісник НТУУ "КПІ". Сер. Інформатика, управління та обчислювальна техніка. – 2007.№ 46– С. 32-44.
7. Патент України №58920А, Пристрій для додавання послідовності добутків пар чисел із плаваючою точкою / Блінова Т.О., Долголенко О.М., Луцький Г.М., Порєв В.М. -2003.Бюл. № 8.