

УДК 004.274

В.М. ОПАНАСЕНКО, О.М. ЛІСОВИЙ

Інститут кібернетики ім. В.М. Глушкова НАН України, Україна

РЕАЛІЗАЦІЯ ПРОБЛЕМНО-ОРІЄНТОВАНИХ ЦИФРОВИХ  
ПРИСТРОЇВ НА КРИСТАЛАХ FPGA

В роботі розглянуті існуючі та розроблені власні засоби для розв'язання погано обумовлених лінійних систем. Розроблені пристрої для арифметики в форматі з плаваючою точкою з підвищеною точністю обчислень (за рахунок збільшення розрядності даних) дозволяють отримувати достовірний результат для деякої множини задач, які при їх розв'язанні класичним способом (використовуючи найширший тип даних дійсних чисел – Extended) на персональному комп'ютері дають недостовірний результат. Запропоновані апаратні реалізації на базі кристалів ПЛІС типу FPGA цифрових пристроїв (розрядності 64-, 80-, 128 біт) для розв'язання системи лінійних алгебраїчних рівнянь по методу Гауса.

**Ключові слова:** ПЛІС, FPGA, VHDL, формат з плаваючою точкою, арифметика з плаваючою точкою, система лінійних алгебраїчних рівнянь.

## Вступ

Однією з задач, яка стала актуальною перед кібернетиками з середини минулого століття – забезпечити достатню точність обчислень на комп'ютері. Вирішенням цієї задачі було збільшення розрядності даних та блоків обчислень дійсних чисел. Чіткий розподіл процесорів за призначенням виділяв процесори для наукових обчислень, особливістю яких була підтримка обчислень з високою точністю, що гарантувало більшу оцінку достовірності по відношенню до інших комп'ютерів.

У 90-ті роки процесори стали масовим продуктом і найбільшого розвитку стали набувати комп'ютери загального призначення (персональні комп'ютери), яким не потрібні обчислення високої точності, а потрібна швидкодія та невисоке енергоспоживання, що й спричинило зупинку розвитку 128-бітної арифметики. Оптимальною було обрано розрядність 80 біт, що й сьогодні має місце в переважній більшості процесорів в яких розрядність FPU (Floating point unit) 80 біт. Це можна пояснити тим, що ринок насамперед орієнтований на бізнес-процеси, мультимедійні застосування, де такої розрядності обчислень достатньо.

У той же час спеціалізовані процесори для наукових обчислень розвивалися значно повільніше і в результаті втратили навіть свою власну нішу – високопродуктивні обчислення.

У 90-ті роки були зупинені роботи по вітчизняним процесорам. Зараз у світі апаратну підтримку 128-бітних обчислень мають лише декілька компаній, таких як IBM на базі операційної системи AIX, Hewlett-Packard на базі операційної систе-

ми HP-UX, AMD-64 сімейства 10h (процесори Phenom), Sun Microsystems процесорна архітектура SPARC-V9 [1].

Деякі системи серед них Solaris, IBM, мають програмно реалізовану підтримку 128-бітної арифметики з плаваючою точкою, яка надається операційною системою.

Сьогодні розрядності бракує як на робочих станціях науковців, так і на більшості суперкомп'ютерів із кластерною архітектурою, які використовують стандартні масові процесори.

В програмуванні такі компілятори, як gcc, intel ісс й інші використовують 128-бітне представлення для дійсних чисел, а потім для виконання операцій ці числа передаються у 80-бітний FPU, де відбувається суттєве погіршення точності.

Проблема обчислень задач з високою оцінкою достовірності має декілька шляхів вирішення, які можна розділити на **апаратні** та **програмні**. До *апаратних* відноситься: комп'ютерні засоби на базі процесорів з підтримкою FPU 128 біт [2]; процесорів IBM POWER6, з підтримкою десяткового FPU 128 біт (DFP) [3]; комп'ютерні засоби, які використовують арифметичні пристрої з розрядністю обчислень 128 біт або більше і піденуються до комп'ютера в одну з стандартних шин (PCI, PCI-e, або інші), такі пристрої можуть бути розроблені на реконфігурованих платах на базі ПЛІС. До *програмних* відноситься: використання сучасних пакетів математичних обчислень, котрі підтримують символну арифметику (Maple, Matlab, інші); використання в програмуванні бібліотек, які дозволяють програмно забезпечити обчислення з довільно заданими форматами (бібліотеки MPFR, GMP) [4]. Кожний

з шляхів має свої переваги й недоліки, однак всі вони мають право на існування і обирати потрібно після аналізу поставленої задачі.

## 1. Постановка задачі

Розв'язок прикладних задач, як правило, починається зі створення фізичних і математичних моделей. Для їх побудови використовують різні гіпотези. Якщо ці гіпотези вірні (похибка гіпотези відсутня або досить мала), то фізична модель правильно відображає закономірності, які мають місце в прикладній задачі. Фізична модель може бути описана за допомогою математичного апарату, наприклад, деякою системою лінійних алгебраїчних рівнянь. При описі фізичних моделей вкрай рідко виникають системи  $\tilde{A}\tilde{x} = \tilde{b}$  з точними вихідними даними. Найбільш типовою є система  $Ax = b$  з наближеними вихідними даними та зазначенням похибки у вихідних даних [5].

Відомо, що в ряді випадків при розв'язанні наукових та інженерних задач на комп'ютерах, користувачі отримують машинні розв'язки, які не мають

фізичного змісту. Це виникає через похибки вихідних даних, відмінність властивостей дискретних та машинних моделей задач, а також відмінність арифметики та машинної арифметики та інші.

Розв'яжемо СЛАР (вираз 1), коефіцієнт обумовленості якої  $1,34 \cdot 10^7$  та СЛАР (вираз 2), коефіцієнт обумовленості якої  $3,56 \cdot 10^{16}$ . Великі коефіцієнти обумовленості вказують на те, що система (1) вироджується на ФПТ (форматі з плаваючою точкою) Single а система (2) вироджується на ФПТ (Single, Double, Extended), тому її розв'язки отримані класичним способом (використовуючи найширший тип даних дійсних чисел – Extended) на персональному комп'ютері дають недостовірний результат, а для її розв'язання потрібно використовувати інші програмні або апаратні засоби.

Система  $Ax = b$  (1) має точні математичні розв'язки  $x$ . Система  $Ax = b$  (2) не має точних математичних розв'язків  $x$ , однак за допомогою символічної арифметики з заданою надвисокою точністю 100 десяткових символів розв'язуємо СЛАР та отримуємо розв'язки  $x$  з точністю  $10^{-87}$ , яку підраховуємо підстановкою  $x$  в рівняння  $Ax - b$ .

$$A = \begin{bmatrix} 0.135 & 0.188 & 0.191 & 0.178 \\ 0.188 & 0.262 & 0.265 & 0.247 \\ 0.191 & 0.265 & 0.281 & 0.266 \\ 0.178 & 0.247 & 0.266 & 0.255 \end{bmatrix}, b = \begin{bmatrix} 0.3516 \\ 0.4887 \\ 0.5105 \\ 0.4818 \end{bmatrix}, x = \begin{bmatrix} 0.4 \\ 0.5 \\ 0.6 \\ 0.5 \end{bmatrix} \quad (1)$$

$$A = \begin{bmatrix} 0.1348531574394464 & 0.1878970588235294 & 0.1909117647058824 & 0.1779264705882353 \\ 0.1878970588235294 & 0.262 & 0.265 & 0.247 \\ 0.1909117647058824 & 0.265 & 0.281 & 0.266 \\ 0.1779264705882353 & 0.247 & 0.266 & 0.255 \end{bmatrix} \quad (2)$$

$$b = \begin{bmatrix} 0.3516 \\ 0.4887 \\ 0.5105 \\ 0.4818 \end{bmatrix}, x = \begin{bmatrix} 0.6662162162161606738798064490430572\dots e13 \\ -0.4016891891890723506952166298245477\dots e13 \\ -0.1665540540539970051894018639784241\dots e13 \\ 0.9797297297302797072574940696301115\dots e12 \end{bmatrix}$$

## 2. Проектування арифметичних блоків

Розглянемо формати даних із плаваючою точкою відповідно стандарту IEEE-754 [6]: одинарної точності (Single) представлений 32 бітами: 1 біт для знака, 8 бітів для порядку й 23 біта для мантиси; подвійної точності (Double) представлений 64 бітами: 1 біт для знака, 11 бітів для порядку й 52 біта для мантиси; розширеної точності (Extended) представлений 80 бітами: 1 біт для знака, 15 бітів для порядку й 64 біта для мантиси; розширеної точності (Quadruple) представлений 128 бітами: 1 біт для знака, 15 бітів для порядку й 112 бітів для мантиси. Відзначимо, що

всі розглянуті формати за винятком Extended мають схований старший біт мантиси, отже, ширина їхніх мантис на одиницю більше.

Узагальнена формула, яка відображає число в ФПТ має вигляд:

$$A = (-1)^S \times 2^{E-E_c} \times 1.M,$$

де  $S$  – знак числа;

$E$  – порядок;

$E_c$  – зсув порядку;

$1.M$  – нормалізована мантиса.

Формати даних, зсув порядку й діапазон допустимих значень представлений на рис. 1.

Незважаючи на те, що діапазон допустимих значень форматів з розширеною точністю однаковий, тобто якщо при рішенні використовується *Quadruple* і результат не буде спеціальним значенням, то для цієї задачі використовуючи *Extended* результат також не буде спеціальним значенням, і навпаки.

У процесі виконання арифметичних операцій над числами у ФПТ може бути отримано результат, який відповідає спеціальним значенням, що встановлено в стандарті *IEEE-754*, у таких ситуаціях формується реєстр прапорів, його структура представлена на рис. 2.

Опис полів реєстра: *RDY* устанавлюється в одиницю, коли визначений результат; *UNDERFLOW* устанавлюється в одиницю, коли в ході виконання операції відбувається втрата значимості; *OVERFLOW* устанавлюється в одиницю, коли в ході

виконання операції відбувається перепоповнення розрядної сітки; *NAN\_RESULT* устанавлюється в одиницю, коли в ході виконання операції результат *NAN-число*; *Inf+\_RESULT* устанавлюється в одиницю, коли в ході виконання операції або під час аналізу вхідних даних визначений результат *Infinity+*; *Inf-\_RESULT* устанавлюється в одиницю, коли в ході виконання операції або під час аналізу вхідних даних визначений результат *Infinity-*; *zero\_RESULT* устанавлюється в одиницю, коли в ході виконання операції або під час аналізу вхідних даних визначений результат *машиинний нуль*; *INVALID\_OPERATION* устанавлюється в одиницю, коли операція над вхідними даними не може бути виконана, наприклад  $(+\infty)+(-\infty)$ ,  $(\pm 0)/(\pm 0)$ ,  $(\pm \text{Infinity})/(\pm \text{Infinity})$ ,  $(\pm \text{Infinity}) \times (0) = \text{NAN-число}$ .

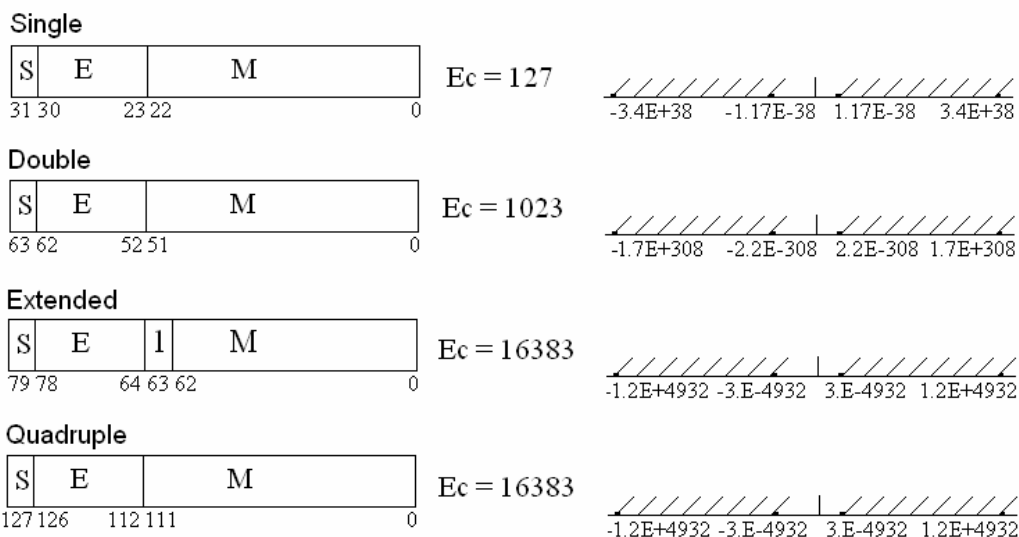


Рис. 1/ Формати даних і діапазон допустимих значень

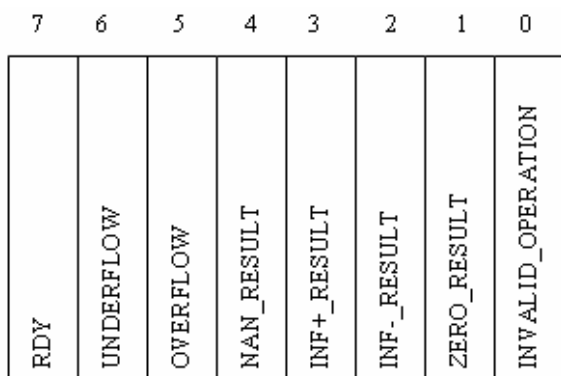


Рис. 2. Регістр прапорів (8 біт)

Модуль додавання-віднімання розробляється шляхом поведінкового опису ступенів конвеєра цього модуля мовою *VHDL*, проектування додаткових вузлів конвеєра й проміжних реєстрів.

Схема модуля додавання збирається в схемотехнічному редакторі.

Модуль додавання-віднімання можна представити у вигляді чотирьох функціональних блоків, зв'язаних між собою умовними й безумовними переходами [7].

Ділення із плаваючою точкою в загальному випадку включає порівняння мантис діленого й дільника, зсув вправо мантиси діленого й обчислення частки. Порядок (як різниця порядків діленого й дільника) обчислюється паралельно. Процедура виконання операції ділення представляється в наступному виді:

$$C = \frac{A}{B} = \frac{(-1)^{S_1} \times 2^{E_1 - E_c} \times 1, M_1}{(-1)^{S_2} \times 2^{E_2 - E_c} \times 1, M_2} = (-1)^{S_3} \times 2^{E_3 - E_c} \times 1, M_3,$$

де  $E_1, E_2, E_3$  – порядки,

$E_c$  – зсув порядку (наприклад для чисел формату Double  $E_c = 1023$ );

$M_1, M_2, M_3$  – мантиси відповідно першого, другого операнду та результату.

Алгоритм операції ділення з плаваючою точкою (при його поведінковому опису мовою VHDL) [8] представляється у вигляді кінцевого автомата (State machine) з 6 станами ( $R1 - R6$ ).

В стані  $R1$ , ініціалізованим сигналом завантаження даних Load, встановлюється сигнал Rdy в стан логічного нуля й виконується аналіз вхідних операндів  $A$  і  $B$  на рівність спеціальним значенням за допомогою таблиці сумісності вхідних операндів з результатом.

Якщо при аналізі виявлена помилка, то в регістр прапорів записується код помилки й здійснюється перехід у стан  $R6$ , у якому відбувається установка Rdy в стан логічної одиниці й закінчення операції. При відсутності помилки в регістр прапорів записується код "00000000" й автомат переходить у стан  $R2$ .

У стані  $R2$  формуються знак та порядок частки:

$$S_3 = S_1 \oplus S_2, E_3 = E_1 - E_2 + E_c.$$

Якщо  $M_1 < M_2$ , то значення  $E_3$  зменшується на одиницю з установкою в '1' відповідної внутрішньої змінної flag. Далі відбувається перевірка умови  $0 > E_3 > 2 \times E_c + 1$ , при виконанні якої автомат переходить у стан  $R6$ , інакше – у стан  $R3$ .

У станах  $R3$  й  $R4$ , зв'язаних безумовним переходом, здійснюється формування мантиси результату при діленні  $M_1/M_2$ .

У стані  $R3$ : якщо flag = 0, то виконується операція  $(M_1 - M_2)$ , якщо flag = 1 ( $M_1 < M_2$ ), то  $(2M_1 - M_2)$ . Значення отриманої різниці подвоюється й присвоюється змінній  $M_1$ . Змінній, що характеризує кількість циклів віднімання, присвоюємо число рівне ширині мантиси мінус 1 (для формату Quaduple це число дорівнює 113-1) і здійснюється перехід у стан  $R4$ .

У стані  $R4$  в кожному циклі віднімання відбувається порівняння  $M_1$  з  $M_2$ , якщо номер циклу більше або дорівнює 0. Якщо  $M_1$  більше або дорівнює  $M_2$ , то біту  $M_3$ , обумовленому в даному циклі, привласнюється значення 1 й виконується операція  $(M_1 - M_2)$ . У протилежному разі віднімання не відбувається й відповідному біту  $M_3$  присвоюється нульове значення. При виконанні

кожного циклу попереднє значення змінної  $M_1$  подвоюється, а номер циклу зменшується на одиницю. Після того, як необхідна кількість циклів буде вичерпана ( $i < 0$ ), відбудеться перехід автомата в стан  $R5$ .

У стані  $R5$  виробляється операція симетричного округлення мантиси, її нормалізація й запис знаку, порядку й мантиси результату ділення у вихідний порт (якщо отриманий результат не є спеціальним значенням). У протилежному разі здійснюється запис відповідного коду в регістр флагів. Далі, незалежно від проведеного аналізу, автомат переходить у стан  $R6$ .

Фактично ділення чисел у ФПТ зводиться до ділення чисел у ФФТ (формат з фіксованою точкою) без відновлення залишку, що реалізовано в стані  $R4$  на базі алгоритму «Ділення з нерухомим дільником без відновлення залишку». Серед алгоритмів ділення у ФФТ реалізовано: ділення з нерухомим діленням і дільником, що зсувається вправо; ділення по SRT-алгоритму; ділення по методу гарвардських ітерацій, які дозволяють змінювати апаратно-часові характеристики блоку, наприклад, алгоритм ділення по методу гарвардських ітерацій скоротить кількість ітерацій алгоритму (наприклад для ділення чисел у форматі Single з 23 ітерацій до 4), але збільшить тривалість синхросигналу й апаратні характеристики блоку.

Модуль множення є найбільш простий і погано дробиться для реалізації, отже реалізуємо його як комбінаційну схему за допомогою поведінкового опису його архітектури мовою VHDL.

Загальні формули для опису операції множення у ФПТ:

$$\begin{aligned} ((-1)^{S_1} \times 2^{E_1 - E_c} \times 1, M_1) \times ((-1)^{S_2} \times 2^{E_2 - E_c} \times 1, M_2) = \\ = (-1)^{S_3} \times 2^{E_3 - E_c} \times 1, M_3, \end{aligned}$$

$$S_3 = S_1 \oplus S_2, E_3 = E_1 + E_2 - E_c, 1, M_3 = 1, M_1 \times 1, M_2,$$

де  $E_1, E_2, E_3$  – порядки,

$E_c$  – зсув порядку (наприклад для чисел формату Extended = 16383),

$M_1, M_2, M_3$  – мантиси відповідно першого, другого операнду і результату.

Так як під час перемноження  $1, M_1 \times 1, M_2$  в частині випадків старша одиниця результату з'являється в останньому біті або передостанньому біті, то в першому випадку необхідно порядок результату збільшити на 1.  $1, M_3$  – формується як старша частина результату перемноження  $1, M_1 \times 1, M_2$ .

В основі блоку множення в ФПТ лежить операція множення двійкових векторів, яку можна реалізувати різними способами: поведінковим описом операції множення, використання функції бібліотеки, формуванням IP-Core за допомогою засобу Core Generator. Крім модулів множення, створених з використанням суматорів на основі LUT, у кристалах серії Spartan-3 є вбудовані перемножувачі Mult18x18, а в кристалах серії Virtex-5 – спеціалізовані секції DSP48E, що реалізують множення 25x18 біт, а також деякі інші функції, необхідні при цифровій обробці сигналів. Ці апаратні засоби дозволяють реалізувати операції множення з різним числом розрядів співмножників, а для збільшення швидкодії при обробці великих масивів чи-

сел є можливість використовувати перемножувачі з різним числом рівнів конвеєризації.

### 3. Пристрій для рішення СЛАР

Для рішення СЛАР методом Гауса розроблено пристрій для наступних форматів даних: Single, Double, Extended, Quadruple. Загальну блок-схему наведено на рис. 3, де ПЗП – програмований запам'ятовуючий пристрій, ОЗП – оперативно-запам'ятовуючий пристрій, СЕЛ – селектор, БОД – блок обробки даних, МУЛЬТ – мультиплексор, СЕ – chip enable (сигнал початок роботи), RDY – ready (сигнал закінчення роботи), WE\_A (WE\_B) – write enable (сигнал дозволу запису).

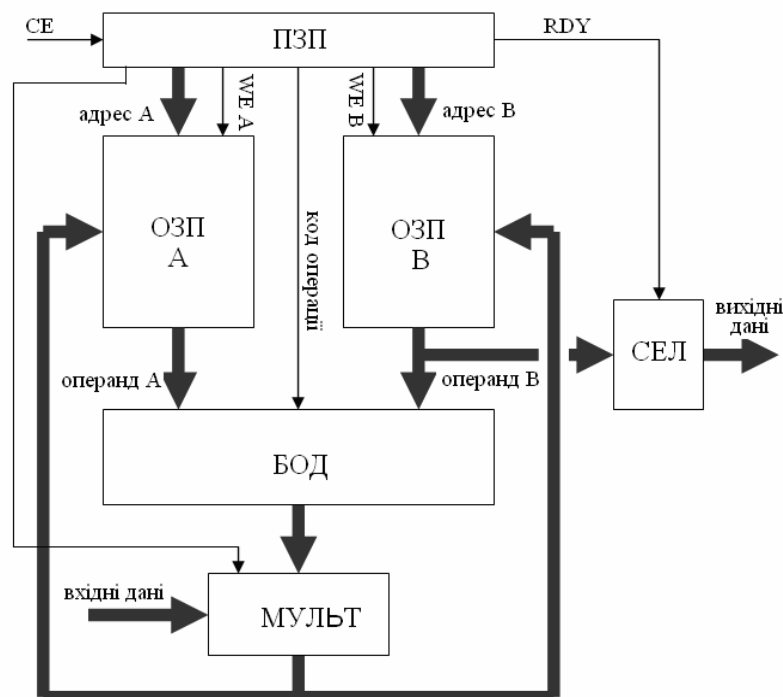


Рис. 3. Загальна блок-схема пристрою

Для кожного з форматів даних всі складові частини схеми окрім ПЗП повинні бути з відповідною розрядністю, для Quadruple – 128 бітні, для Double 64 бітні і так далі. ПЗП – це пам'ять в якій зберігається запрограмований метод Гауса однакова для всіх форматів і програмується на етапі формування ПЗП за допомогою Core Generator. Для розв'язання СЛАР 4-го порядку [9] достатній об'єм 256 слів по 20 бітів. Формат команд, які зберігаються в ПЗП приведено на рис. 4. Для розв'язання систем більших порядків окрім збільшення об'єму ПЗП буде збільшуватися ширина команди, так як збільшиться об'єм ОЗП, а тому поля для їх адресації.

Об'єм ОЗП дорівнює 256 слів, а його ширина відповідна ширині формату даних. ОЗП використо-

вується для зберігання початкових коефіцієнтів, результатів, та проміжних даних під час обчислень. БОД має три необхідні для методу Гауса арифметичні вузли (додавання-віднімання, множення, ділення), відповідної розрядності.

Функціонування пристрою включає три етапи:

1) запис коефіцієнтів системи з навколишнього середовища в ОЗП А та ОЗП В. Мультиплексор ввімкнений на передачу «вхідні данні» на ОЗП. Сигналом «СЕ» запускає виконання програми, команди якої вказують на запис вхідних даних в ОЗП.

2) виконання програми, що описує метод Гауса.

3) сигнал RDY відкриває селектор для видачі інформації. Послідовно читається ОЗП по адресах, в яких записано результати обчислень, дані послідовно віддаються назовні через селектор.

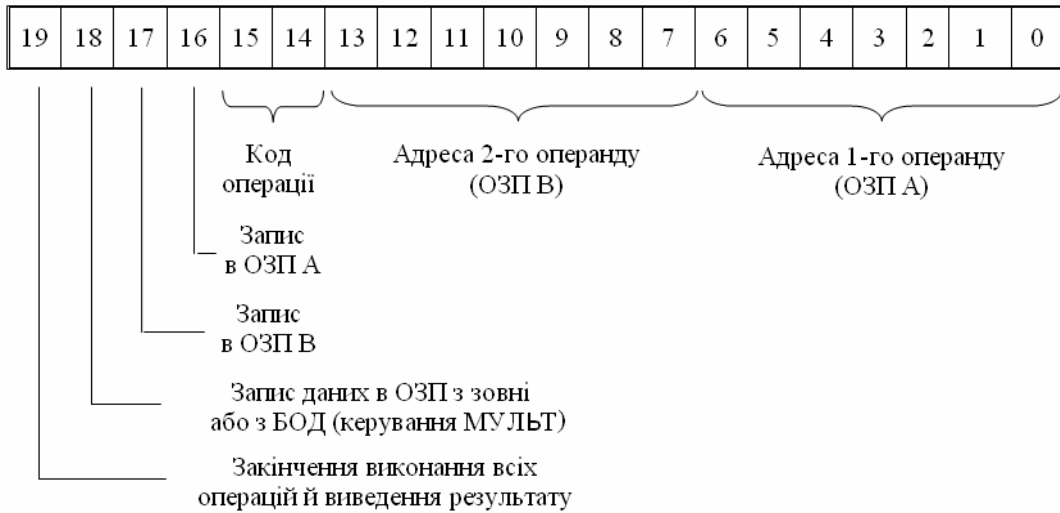


Рис. 4. Формат команди

Тестування пристроїв відбувається на етапах розробки окремих блоків, а також пристрою в цілому. Для тестування використовується система моделювання ModelSim XE3 6,1 яка дозволяє окрім верифікації встановити апаратно-часові характеристики пристроїв, які представлено в табл. 1.

Фірмою Xilinx пропонується ряд арифметичних блоків, у вигляді IP-Core, котрі було використано для побудови копій розроблених пристроїв, для порівняльного аналізу.

Після проведення обчислень СЛАР (вирази 1 та 2) з відповідними коефіцієнтами А та b одержуємо наступні результати (табл. 2, 3).

Таблиця 1

Характеристики пристроїв на Virtex 4 FX100-ff11

Ресурси	Формати даних			
	Single	Double	Extended	Quadruple
Slices	1586 (3%)	6007(14%)	8316 (19%)	23418 (55%)
DSP48	4 (2%)	13 (8%)	16 (10%)	49 (30%)
RAMB16	3 (1%)	3 (1%)	5 (1%)	5 (1%)
Період Clk, нс	50	60	80	100

Таблиця 2

Результати розв'язку СЛАР (вираз 1) апаратним шляхом

Формат	Пристрій, що використовує власні арифметичні блоки	Пристрій, що використовує IP-Core фірми Xilinx
Single	x1=0,254377186298370361 x2=0,587947368621826172 x3=0,636590600013732910 x4=0,478293210268020630	x1=0,164982497692108154 x2=0,641937494277954102 x3=0,659047245979309082 x4=0,464972347021102905
Double	x1=0,400000000287084000 x2=0,499999999821477000 x3=0,59999999953293000 x4=0,500000000015871000	x1=0,400000000170648000 x2=0,499999999899230000 x3=0,599999999860679000 x4=0,50000000000401000
Extended	x1=0,39999999999906770 x2=0,50000000000056311 x3=0,60000000000023408 x4=0,49999999999986117	x1=0,400000000000005191 x2=0,49999999999996858 x3=0,59999999999998718 x4=0,50000000000000757
Quadruple	x1=0,400000000000000000000000002652091 x2=0,49999999999999999999999998397992 x3=0,59999999999999999999999999334615 x4=0,500000000000000000000000000394574	відсутні арифметичні блоки

Таблиця 3

Результати розв'язку СЛАР (вираз 2) апаратним шляхом

Формат	Пристрій, що використовує власні арифметичні блоки	Пристрій, що використовує IP-Core фірми Xilinx
Single	x1=-64153,664062500000000000 x2= 38681,507812500000000000 x3= 16039,548828125000000000 x4=-9434,256835937500000000	x1=-27856,28125000000000 x2= 16796,2792968750000 x3= 6965,29150390625000 x4=-4096,46044921875000
Double	x1= 2264503194623,3600000000000000 x2=-1365362220253,9000000000000000 x3=-566125798647,2270000000000000 x4= 333015175637,3470000000000000	x1= 3343472363451,5000000 x2=-2015917160209,6700000 x3=-835868090854,30700000 x4= 491687112298,69300000
Extended	x1= 6655584185991,900460000000000000 x2=-4012925759200,165400000000000000 x3=-1663896046497,543230000000000000 x4= 978762380293,558017000000000000	x1= 6657982426662,6836300 x2=-4014371757251,6669800 x3=-1664495606665,2391300 x4= 979115062745,14385500
Quadruple	x1=6662162162161,6067166007879779302717040186 x2=-4016891891890,723493568514283355878205310 x3=-1665540540539,970046344699511658272177594 x4=979729729730,27970399318870014412337350964	відсутні арифметичні блоки

В таблиці 2 обробка даних формату Single взагалі не дає позитивного результату, так як не гарантована навіть перша цифра після точки. Деяка різниця обчислень з використанням власних блоків і блоків Xilinx пояснюється відмінністю алгоритмів і округлень, котрі використовувались під час їх розробки.

Варто зауважити, що підрахувавши похибку отриманих результатів, пристрій в якому використовуємо IP-Core фірми Xilinx дає трохи точніші результати (середньоквадратичні похибки результатів для Double рівні 0,00000000013204625 та 0,00000000010278500 відповідно), однак це не впливає на кінцевий висновок, оскільки обидва пристрої гарантують однакову кількість знаків після точки (для даного прикладу Double гарантує 9, Extended – 13, Quadruple – 27 знаків).

В таблиці 3 обробка даних, що використовує формати Single та Double не дає позитивного результату, а Extended гарантує лише 2 перші цифри результатів. Розв'язання на Quadruple гарантує 4 цифри після точки.

### Висновки

При розв'язанні СЛАР (вираз 2), класичним комп'ютерним способом отримані розв'язки не відповідають дійсності, тому коректне розв'язання СЛАР може бути виконане за допомогою наступних підходів: символічні обчислення, бібліотека MPFR, арифметичні пристрої на ПЛІС.

Для рішення СЛАР методом Гауса розроблено пристрої для наступних форматів даних: Single (32

біти), Double(64 біти), Extended (80 біт), Quadruple (128 біт). Результати обчислень показали, що використання Single та Double не дає позитивного результату, а Extended гарантує лише 2 перші цифри результатів. Розв'язання на Quadruple гарантує 3 цифри після точки.

Тестування пристроїв виконано на етапах розробки окремих блоків, а також пристроїв в цілому. Для тестування використано систему моделювання ModelSim XE3 6,1 яка дозволяє окрім верифікації встановити апаратно-часові характеристики пристроїв, які представлено в табл. 1.

Універсальність розробленої схеми пристроїв дозволить використовувати її, в разі необхідності, для розв'язання систем більших порядків та більш широких форматів даних, які можуть бути стандартизовані або ні.

### Література

1. Sun Microsystems [Електрон. ресурс]. – Режим доступу до ресурсу: <http://www.sun.com>.
2. Family 10h AMD Phenom Processor Product Data Sheet [Електрон. ресурс]. – Режим доступу до ресурсу: <http://www.amd.com>.
3. IEEE 754R Decimal Floating-Point Arithmetic [Електрон. ресурс]. – Режим доступу до ресурсу: <http://www.intel.com/technology/itj/2007/v11i1/s2-decimal/1-sidebar.htm>.
4. Mpfr library [Електрон. ресурс]. – Режим доступу до ресурсу: <http://www.mpfr.org>.
5. Химич А.Н. Анализ достоверности компьютерных решений систем линейных алгебраических уравнений с приближенно заданными исходными

данными / А.Н. Химич, Е.А. Николаевская // Кибернетика и системный анализ. – 2008. – №6. – С. 12-18.

6. Hollash S. IEEE Standard 754 Floating Point Numbers [Електрон. ресурс] / S. Hollash. – Режим доступу до ресурсу: <http://IEEE/IEEE Standard 754 Floating-Point.htm>.

7. Палагин А.В. Проектирование реконфигурируемых систем на ПЛИС / А.В. Палагин, В.Н. Опанасенко, А.Н. Лисовый // Технология и конструирование в электронной аппаратуре. – 2007. –

№ 3. – С. 15-19.

8. Палагин А.В. Физическое моделирование цифровых устройств на ПЛИС с помощью моделирующей платы / А.В. Палагин, В.Н. Опанасенко, А.Н. Лисовый // Матеріали 13 міжнародної конференції з автоматичного управління «Автоматика-2006». – 2007. – С. 190-194.

9. Молчанов І.М. Машинні методи розв'язування задач прикладної математики. Алгебра, наближення функцій, звичайні диференціальні рівняння / І.М. Молчанов. – К.:Наук. думка, 2007. – 550 с.

Надійшла до редакції 18.12.2008

**Рецензент:** д-р техн. наук, пров. наук. співр., Н.І. Алішов, Інститут кібернетики ім. В.М. Глушкова, Київ, Україна

### РЕАЛИЗАЦИЯ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ЦИФРОВЫХ УСТРОЙСТВ НА КРИСТАЛЛАХ FPGA

*В.Н. Опанасенко, А.Н. Лисовый*

В работе рассмотрены существующие и разработанные собственные средства для решения плохо обусловленных систем линейных уравнений. Разработаны устройства для арифметики в формате с плавающей точкой с повышенной точностью вычислений (за счет увеличения разрядности данных) позволяют получать достоверный результат для множества задач, которые при их решении классическим способом (используя самый широкий вещественный тип данных – Extended) на компьютере не дает достоверный результат. Предложены аппаратные реализации на базе кристаллов ПЛИС типа FPGA цифровых устройств (разрядностей 64–, 80–, 128 бит) для решения систем линейных алгебраических уравнений по методу Гаусса.

**Ключевые слова:** ПЛИС, FPGA, VHDL, формат с плавающей точкой, арифметика с плавающей точкой, система линейных алгебраических уравнений.

### THE REALIZATION TASK-ORIENTED DIGITAL DEVICE ON CRYSTAL OF FPGA

*V.N. Opanasenko, A.N. Lisovyi*

The ware have analyzed and developed to accomplish ill-conditioned systems of line algebraic equation. The devices for more precision floating point arithmetic have developed (by increasing format wide). It permits get correct result for some part of task, which can't be calculation on classical computers arithmetic with floating point event using type Extended, cause result which we get after calculation is wrong. It's suggested hardware devise (capacity 64–, 80–, 128 bit) on base of PLD (FPGA) for calculation systems of line algebraic equation by Gauss.

**Keywords:** PLD, FPGA, VHDL, floating point format, arithmetic with floating point, system of line algebraic equation.

**Опанасенко Володимир Миколайович** – д-р техн. наук, пров. наук. співр., Інститут кібернетики ім. В.М. Глушкова, Київ, Україна, e-mail: vloranas@ukr.net.

**Лісовий Олександр Миколайович** – мол. наук. співр., Інститут кібернетики ім. В.М. Глушкова, Київ, Україна, e-mail: Lan-Sasha@yandex.ru.