УДК 621.396

А.О. Левченко, к.т.н., доц.,
Р.М. Войтенков
Військова академія (м. Одеса), Україна

ГРАНИЧНІ ТОЧНОСТІ ОБЧИСЛЕНЬ В ІНФОРМАЦІЙНИХ СИСТЕМАХ З ПРЕДСТАВЛЕННЯМ ЧИСЕЛ ІЗ ПЛАВАЮЧОЮ КОМОЮ

Розглянуто правила додавання, віднімання, множення й діленням чисел у форматі з плаваючою комою, певним стандартом IEEE. Окреслено особливості реалізації алгоритмів додавання і віднімання, множення, ділення , що впливають на точність обчислень., в зв'язку з чим наведені поняття розрядів захисту й усікання. Зроблено висновок о необхідності впровадження використання довгої арифметики представлення чисел з плаваючою комою.

Ключові слова: *представлення чисел, точність обчислень, пам'ять і адреси, байтова адресація, довге число.*

Точність обчислень визначається кількістю цифр (двійкових або десяткових розрядів) з яким даний комп'ютер виконує обчислення. Звичайна для сучасних комп'ютерів точність близько 15 десяткових розрядів (64 двійкових). Для особливо точних обчислень, наприклад обчислення числа "Пі", ще в 60-ті роки були розроблені машини серії "Мир", які могли обчислювати з точністю до 999 десяткових цифр.

Точність представлення числа залежить від кількості розрядів, що відводяться під число: 8 розрядів – точність становить два десяткові знаки, 32 розряди – точність 10 десяткових знаків.

Самою більшою перевагою збільшення розрядності ЕОМ є використання оперативної пам'яті збільшеного обсягу. 8-розрядний процесор дозволяє звертатися до ОЗУ обсягом до 64 кб, 16-розрядний процесор може працювати з ОЗУ до 1024 кб, 32-розрядний до 4 Гб.

Актуальною стаю завдання визначити граничні точності обчислень в інформаційних системах з представленням чисел із плаваючою комою

Виклад основного матеріалу. Правила додавання, віднімання, множення й діленням чисел у форматі з плаваючою комою регулюють тільки базові кроки при виконанні операцій. Можливість переповнення або втрати значимості в них не врахована. Більше того, для проміжних значень мантиси й порядку може знадобитися більше 24 і 8 біт відповідно. Ці й інші особливості операцій з плаваючою комою потрібно прийняти до уваги в процесі розробки арифметичних пристроїв, відповідних до стандарту IEEE. Усі особливості ми, звичайно, не розглянемо, але про найважливіші, такі як округлення, поговоримо обов'язково.

Якщо порядок двох операндів із плаваючою комою різний, їх мантиси перед додаванням або відніманням повинні бути зрушені відносно один одного. Як приклад розглянемо додавання чисел 2,9400 x 102 і 4,3100 x 104. Представимо 2,9400 x 102 як 0,0294 x 104 і складемо мантиси, внаслідок чого одержимо 4,3394 x 104. Послідовність операцій при додаванні і відніманні можна описати в такий спосіб.

Безпосередньо відповідні правила додавання і віднімання, множення, ділення наведені в [1]. Реалізація алгоритмів додавання і віднімання, множення, ділення має ряд особливостей. Хоча розмір мантис вихідних операндів і кінцевого результату обмежено 24 розрядами, включаючи яку мається на увазі провідну 1, важливо, щоб у ході проміжних обчислень зберігалося кілька додаткових розрядів, названих розрядами захисту або сторожовими розрядами. Це дозволяє забезпечити необхідну точність кінцевого результату [2].

Коли розряди захисту віддаляються з кінцевого результату, мантиса усікається до 24 розрядів. Усікання проводиться й в інших ситуаціях, зокрема, при перетворенні десяткового числа у двійковий формат. Звичайно, для позначення даної операції вживається також термін округлення, але ми будемо використовувати його в більш вузькому змісті, говорячи про одне зі способів усікання значення.

В [1] наведені кілька способів усікання. Найпростіший з них полягає у видаленні розрядів захисту без зміни інших розрядів. Саме така операція називається усіканням. Припустимо, що необхідно скоротити дробове значення із шести розрядів до трьох. Будь-які значення, що лежать у діапазоні від 0,b-1b-2b-3000 до 0,b-1b-2b-3111, усікаються до 0,b-1b-2b-3. Помилка усікання до 3-розрядного результату перебуває в діапазоні 0 - 0,000111, тобто від 0 і майже до 1 у молодшому з розрядів, що залишилися. В нашому прикладі це розряд b-3. У результаті усікання виходить зміщене наближення, оскільки діапазон помилки не симетричний нулю.

Ще одним простим методом усікання є фоннеймановське округлення. Якщо всі розряди, що видаляються, містять нулі, останні відкидаються без зміни розрядів, що залишилися. Але якщо хоч один з розрядів, що видаляються, містить 1, молодший розряд значення, що залишилося, установлюється в 1. У нашому прикладі усікання дробового значення із шести розрядів до трьох будь-які 6-розрядні значення, у яких b-4b-5b-6 не рівні 000, зменшується до 0,b-1b-21. Величина помилки цього методу лежить у діапазоні від –1 до +1 молодшого з розрядів, що залишилися. І хоча при такому способі усікання значення діапазон помилки більше, чим при простому усіканні, її максимальна величина залишається тією ж, а наближення виходить незміщеним, тому що діапазон помилки симетричний щодо нуля [3].

Для тих операцій, у яких бере участь багато операндів і виконується значна кількість проміжних дій, переважніше незміщене наближення, при якому позитивні й негативні помилки компенсують один одного. Якщо статистично оцінити результати складних обчислень з використанням незміщеного наближення, їх точність виявиться дуже високою.

Перейдемо до наступного методу усікання значення – округленню. Завдання округлення полягає в максимальному наближенні результуючого значення до вихідного. Воно набагато точніше отриманого шляхом усікання й до того ж дає незміщений результат. Округлення виконується так: якщо старший з розрядів, що видаляються, містить 1, до молодшого з розрядів, що залишилися, числа додається 1. Таким чином, 0,b-1b-2b-31 округляється до 0,b-1b-2b-3+0,001, а 0,b-1b-2b-30... - до 0,b-1b-2b-3. Результат виходить гранично близьким до вихідного числа за винятком ситуації, що коли видаляються розряди, рівні 10...0. У цьому випадку вихідне значення лежить посередині між двома можливими усіченими представленнями. Для забезпечення незміщеного наближення можна вибирати значення молодшого з розрядів, що залишилися, таким чином, щоб завжди виходило найближче парне значення.

У розглянутому прикладі з використанням 6 розрядів значення 0,b-1b-20100 округляється до 0,b-1b-20, а 0,b-1b-21100- до 0,b-1b-21+ 0,001. Цю технологію [1] описує так: округлення до найближчого числа, а у випадку двох однакових помилок округлення – до найближчого парного числа.

Помилка округлення лежить у діапазоні від -1/2 до +1/2 значення молодшого з розрядів, що залишилися. Очевидно, що це найкращий метод. Однак реалізувати його трудніше всього, оскільки буде потрібно додаткова операція й, можливо, нормалізація. Згідно зі стандартом IEEE, для усікання чисел за замовчуванням використовується округлення. У цьому стандарті описані й інші методи усікання; усі вони визначені як режими округлення [1].

Обговорюючи помилки, що виникають через видалення розрядів захисту, розглянуто одну операцію усікання. Якщо програма виконує довгу послідовність обчислень, у якій задіяні числа із плаваючою комою, аналіз діапазонів помилок і кінцевих результатів значно ускладнюється. Цей

аспект числових обчислень в роботі більше обговорюватись не буде, але відповідно до стандарту IEEE розряди захисту й округлення визначаються по іншому.

Результати однієї операції повинні обчислюватися з точністю до половини одиниці в молодшому розряді. У загальному випадку із цією метою результат повинен коротшати шляхом округлення.

Для забезпечення такої точності на проміжних кроках виконання операції досить зберігати три розряди захисту. Перші два – це старші розряди мантиси, які наприкінці обчислень підлягають видаленню. Третій розряд містить результат виконання логічної операції АБО всіх розрядів мантиси, крім зазначених двох розрядів захисту. Підтримувати його на проміжних кроках операції досить просто. Він ініціалізірується нулем, а коли в нього з мантиси висувається 1,стає рівним 1 і зберігає це значення. Тому даний розряд іноді називають другим проміжним бітом округлення (**sticky bit**).

Числові й символьні операнди, так само як і команди, зберігаються в пам'яті комп'ютера. Пам'ять складається з багатьох мільйонів гнізд, у кожній з яких утримується один біт інформації, що має значення 0 або 1. Оскільки один біт здатний представити дуже маленьку кількість інформації, біти рідко обробляються поодинці. Як правило, їх обробляють групами фіксованого розміру. Для цього пам'ять організується таким чином, що групи по n біт можуть записуватися й зчитуватися за одну базову операцію.

Група з n біт називається словом інформації, а значення n – довжиною слова. Схематично пам'ять комп'ютера можна представити у вигляді набору слів.

Довжина слова сучасних комп'ютерів становить від 16 до 64 біт. Якщо довжина слова комп'ютера рівна 32 битам, в одному слові може зберігатися 32-розрядне число в системі доповнення до двох або чотири символи ASCII, що займають по 8 біт. Вісім, що йдуть підряд битів називаються байтом. Для представлення машиною команди потрібно одне або кілька слів. Про кодування машинних команд ми поговоримо далі в цій главі, після того як обговоримо команди на рівні мови.

Для доступу до пам'яті з метою запису або читання окремих елементів інформації, будь то слова або байти, необхідні імена або адреси, що визначають їхнє розташування в пам'яті. У якості адреси традиційно використовуються числа з діапазону від 0 до 2k-1 зі значенням k, достатнім для адресації всієї пам'яті комп'ютера. Усі 2k адреси становлять адресний простір комп'ютера.

Отже, пам'ять складається з 2k адресуємих елементів. Наприклад, використання 24-розрядних адрес дозволяє адресувати 224 (16777216) елементів пам'яті. Звичайно ця кількість адресуємих елементів позначається як 16 М (16 мега), де 1 М - 220 (1048576). 32-розрядним адресам відповідає адресний простір з 232, або 4 Г (4 гіга), елементів, де 1 Г - 230. Крім того, часто використовуються позначення ДО (кіло), що відповідає 210 (1024), і Т (тера), що відповідає 240.

Слід окремо відзначити три основні одиниці інформації: біт, байт і слово. Байт завжди рівний 8 бітам, а довжина слова звичайно коливається від 16 до 64 біт. Окремі біти, як правило, не адресуються. Найчастіше адреси призначаються байтам пам'яті. Саме так адресується пам'ять більшості сучасних комп'ютерів, і саме цей спосіб адресації ми будемо використовувати в цій книзі. Пам'ять, у якій кожний байт має окремий адрес, називається пам'яттю з байтовой адресацією. Послідовні байти мають адреси 0, 1, 2 і т.д. Таким чином, при використанні слів довжиною 32 біта послідовні слова мають адреси 0,4,8,..., і кожне слово складається з 4 байт.

Такого типу даних вистачає практично завжди, якщо звичайно мова не йде про дуже малі раціональні числа, або вирішення математичних завдань з дуже високою точністю.

Не допомагають і раціональні типи даних тому, що представляють число лише з деякою точністю. Це зрозуміло з їхнього подання в пам'яті ЕОМ. Наприклад, тип, **double** має діапазон значень від 1.7*e*-308 до 1.7*e*+308 [2]. Однак він може зберігати лише 16 значущих цифр числа і його порядок. Саме тому він не придатний у тих задачах, де необхідний точний результат.

Таким чином необхідно використовувати довгу арифметику [4, 5] представлення чисел в вигляді масивів даних для підвищення точності розрахунків, необхідно самостійно реалізувати потрібні типи даних та математичні операції для довгої арифметики представлення чисел.

Список використаних джерел

1. *[Електронний ресурс] Режим доступу: http://kufas.ru/arch2/page32.htm.*
2. *Левченко А. О. Анализ предельных точностей вычислений в информационных системах с представлением чисел с плавающей запятой / А. О. Левченко, Р. М. Войтенков // Збірка тез доповідей 3-го науково-технічного семінару "Перспективні шляхи розвитку інформаційних систем прицілювання та самонаведення високоточного озброєння РВіА"– Львів : АСВ, 2012. – С. 117.*
3. *Левченко А.О. Результати моделюючого експерименту перевірки працездатності обчислювальної системи з представленням чисел з плаваючою комою / А. О. Левченко, Р. М. Войтенков // Збірка тез доповідей 3-го науково-технічного семінару "Перспективні шляхи розвитку інформаційних систем прицілювання та самонаведення високоточного озброєння РВіА"– Львів : АСВ, 2012. – С. 119.*
4. *Левченко А.О. Метод представлення чисел для програмних засобів гарантоздатних інформаційних технологій систем підтримки прийняття рішень для керування станом ОВТ / А. О. Левченко, Р. М. Войтенков // Збірник тез доповідей 19-ї науково-практичної конференції "Проблеми створення, розвитку та застосування інформаційних систем спеціального призначення". – Житомир : ЖВІ НАУ, 2012. – С. 142.*
5. *Левченко А.О. Метод представлення чисел для програмних засобів інформаційних систем / А. О. Левченко, Р. М. Войтенков // Збірник 3-го науково-технічного семінару "Геоінформаційні системи в військових задачах". – Львів : АСВ, 2012. – С. 114–120.*

Рецензент: Скачков В.В., д.т.н., проф., Військова академія (м. Одеса)

ПРЕДЕЛЬНЫЕ ТОЧНОСТИ ВЫЧИСЛЕНИЙ В ИНФОРМАЦИОННЫХ СИСТЕМАХ С ПРЕДСТАВЛЕНИЕМ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

А.О. Левченко, Р.М. Войтенков

Рассмотрены правила прибавления, вычитания, умножения и деления чисел в формате с плавающей запятой, определенные стандартом IEEE. Очерчены особенности реализации алгоритмов прибавления, вычитания, умножения и деления, которые влияют на точность вычислений, в связи с чем приведены понятия: разряды защиты и усечение. Сделан вывод в необходимости использования длинной арифметики представления чисел с плавающей запятой.

Ключевые слова: *представление чисел, точность вычислений, память, адреса, байтовая адресация, длинное число.*

LIMITING ACCURACY OF THE CALCULATIONS IN INFORMATION SYSTEM WITH NUMERATION WITH SAILLING COMMA

A. Levchenko, R. Voitenkov

The Considered rules of the addition, subtractions, multiplying and fissions numbers in format with sailling comma, determined by standard IEEE. The Outlined particularities to realization algorithm additions, subtractions, multiplying and fissions, which influence upon accuracy of the calculations, in connection with than is brought notions: categories of protection and catting. Conclusion is Made in need of the use the long arithmetic of the numeration with sailling comma.

The Keywords: *numeration, accuracy of the calculations, memory, the address, byte addressing, long number.*