



УДК 621.396

ARITHMETIC OPERATIONS FOR BINARY NUMBERS REPRESENTED AS ARRAYS**БАЗОВІ АРИФМЕТИЧНІ ОПЕРАЦІЇ ДЛЯ ДВІЙКОВИХ ЧИСЕЛ ПРЕДСТАВЛЕНИХ ЯК МАСИВИ**

Levchenko A.A. / Левченко А.О.

с.т.с., ас.проф. / к.т.н., доц.

Odessa National University, Odessa, Dvoryanska, 2, 65029

Одесский национальный университет, Одесса, Дворянская, 2, 65029

Аннотация. В работе рассматривается метод представления чисел как двоичных массивов. Для соответствующего подання наведено зміст сукупності елементарних операцій виконання точних комп'ютерних обчислень. Особливо показана доцільність представлення двоичних чисел у вигляді масивів для практичних завдань.

Ключевые слова: арифметичні операції, різнорозмірні дані, представлення довгих чисел в двоичному коді, масиви двоичних чисел, розрядність двоичного числа.

Вступ.

Числові значення діагностичних параметрів, які використовуються в праграмно-апаратних комплексах (ПАК) складних технічних систем підвищеної небезпеки (СТС ПН), відрізняються різнорозмірністю. В цьому випадку під різнорозмірністю розуміється різна кількість елементів розрядної сітки представлення числових значень сукупності параметрів в комп'ютері [1, 2]. Ситуація унеможливорює достовірний прогноз часу виходу параметрів за межі допусків відомими методами та вимагає розвитку обчислювального забезпечення ПАК систем діагностики з прогнозуванням стану СТС ПН.

Сучасні мови програмування реалізують типи даних для роботи з великими раціональними й цілими числами. Типом даних, що описує найбільший діапазон значень цілих чисел доступних нам (C++ і Pascal) є тип даних `__int64`, який вміщує 8-байтове ціле число і може представляти цілі числа від 0 до 18 446 744 073 709 551 615 (2^{64}), або від - 9223372036854775808 до + 9223372036854775807. Такого типу даних вистачає, якщо мова не йде про малі раціональні числа (наприклад розрахунки похибок в системах технічної діагностики), або вирішення математичних задач з дуже високою точністю (наприклад реалізація чисельних методів в інформаційних технологіях СППР з прогнозуванням) [3].

Подання в пам'яті ЕОМ раціональних типів даних обумовлює обмеження на точність їх представлення. Наприклад, тип, `double` має діапазон значень від $1.7e-308$ до $1.7e+308$. Однак він може зберігати лише 16 значущих цифр числа і його порядок. Саме тому цей тип представлення даних не придатний для згаданих задач, де необхідний точний результат.

При реалізації відомих алгоритмів проведення арифметичних операцій в ЕОМ внаслідок неминучих округлень результату обчислень вже при складанні системи рівнянь та обчисленні поліномів 4 ступеню починають накопичуватися помилки комп'ютерних розрахунків які приводять до виродження матриць і унеможливають отримання результату обчислень [3, 4]. Ці помилки пов'язані



з обмеженою довжиною розрядної сітки ЕОМ (що зазвичай дорівнює 32 або 64 біта)

Для вирішення зазначеної проблеми був запропоновано метод представлення чисел як масивів [5, 6]. Під використанням довгої арифметики, в цьому випадку, розуміється арифметичні дії над числами, які за поданням чисел обходять обмеження подання стандартних типів даних та відповідно унеможливають накопичення похибок округлень. Використання довгої арифметики, в свою чергу, дозволяє вирішувати за допомогою простих відомих методів систем лінійних алгебраїчних рівнянь великої розмірності на ЕОМ для завдань прогнозу стану СТС [7].

Саме складність ПАК автоматизованих систем метрологічного супроводження з прогнозуванням (АСМСП), вбудованих систем технічної діагностики з прогнозуванням (ВСТДП) та систем підтримки прийняття рішень (СППР) про стан СТС з числовим вимірювальним контролем фактично стала причиною припинення відповідних робіт в 90-х роках ХХ сторіччя [8, 9]. Подальші дослідження набули теоретичного характеру й найбільш активно проводились у Московському інституті експертизи та випробувань.

Таким чином в роботі поставлена мета: обґрунтування змісту сукупності елементарних операцій для випадку представлення двійкових чисел як масивів. Саме досягнення цієї мети створює передумови розвитку бібліотеки програмних продуктів точних обчислень, а завдання практичної реалізації відповідних інформаційних технологій в ПАК прогнозування стану СТС ПН набуває нового змісту [10, 11].

Основний текст.

Говорячи про представлення довгих чисел у вигляді масивів масив, потрібно в першу чергу визначити кількість елементів масиву. Можливі два варіанти: або розмір масиву в програмі не фіксований, та пам'ять виділяється під нього по мірі потреби; або розмір масиву в програмі фіксований, та пам'ять виділяється однаково для всіх чисел. Для спрощення опису алгоритмів в методі подання чисел пропонується використовувати другий варіант.

Так як максимальну кількість цифр довгого числа або задано в постановці завдання, або його неважко оцінити, константу $MaxN$ оголосимо і передбачимо зберігання довгого числа в масиві розміру $[1...MaxN]$.

Це не означає, що потрібно заповнити всі незайняті елементи нулями й надалі всі арифметичні дії використовувати зі всіма $MaxN$ цифрами, не відрізняючи зайняті від незайнятих. Виконуючи дії тільки над потрібними розрядами, можна істотно виграти в часі роботи програми (наприклад, якщо при порівнянні чисел в одному з них більше знаків, то воно й буде більше - можна навіть не дивитися на цифри).

Потрібно вирішити, що робити з незайнятими старшими елементами масиву: розміщати там примусово нулі або вважати, що там можуть бути будь-які числа. Розумніше допускати будь-які числа, оскільки багаторазові заповнення всіх $MaxN$ розрядів нулями потребує багато ресурсів. Тому потрібно не забувати заповнювати нулями окремі цифри або діапазони там, де без цього виникає ризик прийняти некоректні числа за реальне значення.



Порядок зберігання цифр в масиві краще, коли в першому елементі масиву зберігається молодша цифра (кількість одиниць), у другому - передостання по старшинству (кількість "десятків" системи числення) і т.д. При такому порядку цифри однакових розрядів знаходяться в елементах з однаковими індексами. Крім того, якщо після арифметичної дії змінюється кількість розрядів (наприклад, $98+5 = 103$), то змінюється тільки вміст старших розрядів і зсувати цифри не доводиться.

Таким чином: пропонується представляти довге ціле не просто масивом, а записом із додатковими полями: масивом і цілочисельною змінною – кількістю реально зайнятих елементів масиву. Також доцільно в окремому елементі масиву зберігати знак числа (наприклад 1 позначає від'ємне число, 0 додатне) та положення коми в числі.

У найпростішому випадку кожний елемент масиву байтів може містити одну десяткову цифру числа, але це неекономно по пам'яті: байт може містити 256 значень, і лише 10 з них будуть використовуватися. Інша крайність - використовувати масив байтів як число в системі числення з основою 256: будь-яке значення байта використовується і є значенням 256-ричної цифри, і пам'ять використовується максимально ощадливо. До того ж, вдається прискорити обробку за рахунок використання спеціальних побітових команд. Але ввід-вивід такого числа в десятковій формі виявляється нетривіальним завданням. Тому систему числення з основою 256 (або 65 536) використовують, коли є жорсткі вимоги по пам'яті або відомо, що майже вся робота відноситься до "внутрішніх" дій над числами і лише мала частина - до вводу-виводу.

Часто розумним виявляється компромісний варіант: масив елементів типу long, що представляють числа від 0 до 999 999 999 (цифри по основі 1 000 000 000). У порівнянні зі зберіганням однієї десяткової цифри в unsigned char, вдається заощадити 25 % пам'яті, майже не ускладнивши ввід-вивід. При цьому ще й прискорюються деякі арифметичні дії за рахунок того, що в числі, записаного в 1 000 000 000-ричній системі, вдев'ятеро менше цифр, ніж у числі записаного в десятковій.

Слід зазначити у якому порядку зберігати цифри в масиві. На перший погляд – як пишуться на папері, так і зберігаються. Однак краще порядок, коли в першому елементі масиву зберігається молодша цифра (кількість одиниць), у другому – передостання по старшинству (кількість "десятків" системи числення) і т.д. При такому порядку цифри однакових розрядів знаходяться в елементах з однаковими індексами. Крім того, якщо після арифметичної дії змінюється кількість розрядів (наприклад, $98+5 = 103$), то змінюється тільки вміст старших розрядів і зсувати цифри не доводиться.

Потрібно також вирішити, що робити з незайнятими старшими елементами масиву: розміщати там примусово нулі або вважати, що там можуть бути будь-які числа. Звичайно розумніше допускати будь-які числа, оскільки багаторазові заповнення всіх MaxN розрядів нулями потребує багато ресурсів. Тому потрібно не забувати заповнювати нулями окремі цифри або діапазони там, де без цього виникає ризик прийняти некоректні числа за реальне значення.

Отже в остаточному варіанті наше довге число (наприклад число -18 446



744 073 709 551 615,1) буде записане в масиві у вигляді (табл.1):

Таблиця 1

Приклад представлення десятичного числа як масиву чисел

Номер елемента масиву					
1	2	3	4	5	6
3	1	1	95516151	467440737	184
Розмір числа	Знак числа	Положення коми	1	2	3

Авторська розробка

В наведеному представленні в 1-й елемент масиву записується розмір числа; в 2-й елемент масиву записується знак числа (1 – від'ємне, 0 – додатне); в 3-й елемент масиву записується положення коми в числі; в 4,5 та 6-й елемент масиву – саме число з основою системи числення 1 000 000 000, при цьому число записується навпаки, тобто в першому елементі масиву зберігається молодша цифра (кількість одиниць), у другому – передостання по старшинству (кількість "десятків" системи числення) і т.д.

Розглянемо зміст виконання базових операцій на рівні архітектури мікрокоманд комп'ютера з двійковими числами у вигляді масивів. В якості базових пропонується окреслити наступну сукупність операцій: ввід та вивід довгого числа, порівняння довгих чисел з врахуванням знаку числа, порівняння довгих чисел без врахуванням знаку числа, нормалізація довгих чисел, додавання довгих чисел без врахуванням знаку числа, додавання та віднімання довгих чисел з врахуванням знаку числа, отримання різниці довгих чисел без врахуванням знаку числа. В роботі останній не розглядається в зв'язку з громіздкістю опису процедур.

В реалізації способу вводу довгого числа у вигляді алгоритму пропонується наступні елементи системи мікрокоманд для реалізації в машинних кодах: *A* – масив вхідного числа, що представлено як масив з елементами типу *long* довжиною по 4 байти розмірністю в 100 елементів. В змінній *inStroka* з типу *AnsiString* представлено вхідне число. В алгоритмі передбачено введення змінних *i*, *j*, *lenOsn*, *startPos*, *endPos*, *PozZnaka* описаних як дані цілого типу *int* довжиною 4 байти. Проміжний масив *buf* в алгоритмі представлено як масив з елементами типу *char* довжиною по 1 байту, розмірністю в 10 елементів. В алгоритмі введено процедуру *Pos(subStr)*, що повертає індекс першого символу входу в рядок *inStroka* підрядка *subStr*. Процедура *Delete(index, count)* видаляє з рядка *inStroka*, починаючи з позиції *index* кількість символів, що дорівнює *count*. Процедура *Length()* повертає кількість символів в рядку *inStroka*. Саме результат роботи алгоритму на основі методу є масив довгого двійково-десятичного числа записаний у вигляді, що представлений в таблиці 1.

Авторська розробка

Проблема виводу довгих чисел полягає в тому, що виводити число необхідно з кінця масиву, так як в кінці масиву зберігаються старші розряди числа. Крім того виводити необхідно фіксоване число розрядів (залежить від



основи системи обчислення), за винятком першого виведеного елемента.

Для виводу числа з кількістю цифр меншою від основи СЧ необхідно його попередньо доповнити нулями зліва, при цьому цифри необхідно спочатку записати у деякий буфер, а потім вивести у довге число.

Отже загальні кроки способу (методу) виводу можна визначити наступним чином:

- вивести останній елемент масиву,
- для всіх елементів від передостаннього до першого,
- якщо число розрядів менше, доповнити елемент нулями ліворуч,
- вивести поточний елемент масиву,
- якщо число раціональне, вставити в отримане число кому,
- визначити та вивести знак числа,
- вивести отримане довге число.

В реалізації способу виводу довгого числа у вигляді алгоритму пропонується наступні елементи системи мікрокоманд для реалізації в машинних кодах: *A* – масив вхідного числа представлений як масив з елементами типу *long* довжиною по 4 байти розмірністю в 100 елементів. В змінну *Stroka* з типом *AnsiString* записується вихідне число. Змінні *i*, *j*, *k*, *len*, *DovChys*, *KilCykl* описані як дані цілого типу *int* довжиною 4 байти. Змінна *tmp* типу *AnsiString* використовується як масив проміжних результатів. Метод *Length()* повертає кількість символів в рядку *inStroka*. Метод *Insert (str, index)* вставляє в рядок *Stroka* підрядок *str*, починаючи з індексу *index*.

Авторська розробка

Складність порівняння довгих чисел полягає в тому, що раціональні числа можуть мати різну кількість знаків після коми, що в сою чергу унеможливорює порівняння окремих елементів масиву кожного числа. Для вирішення цієї проблеми пропонується метод нормалізації порівнювальних чисел, який полягає у приведенні дрібної частини числа до однакового розміру. Завжди нормалізується число з меншою кількістю знаків після коми шляхом додавання до нього нулів справа.

Загальні кроки алгоритму порівняння чисел можна визначити наступним чином:

- порівнюється знак чисел, якщо знаки різні, одразу повернути результат порівняння й вийти,
- проводиться нормалізація чисел,
- якщо довжини чисел не рівні, відразу повернути результат порівняння й вийти,
- для всіх елементів від останнього до першого якщо поточні розряди не рівні повернути результат порівняння,
- вийти з алгоритму,
- повернути признак рівності чисел.

В розробленому алгоритмі *A* і *B* – масиви вхідних чисел представлені як масиви з елементами типу *long* довжиною по 4 байти розмірністю в 100 елементів. Змінні *i*, *Roz_A*, *Roz_B*, *Kil* описані як дані цілого типу *int* довжиною 4 байти. Змінна *tmp* типу *AnsiString* використовується як масив проміжних



результатів. Метод Length() повертає кількість символів в рядку. Результатом роботи функції є числа -1, 0 та 1, де 0 – числа рівні, 1 – число A більше числа B, -1 – число B більше числа A.

Авторська розробка

Порядок порівняння довгих чисел без врахування знаку числа такий самий як і при порівнянні довгих чисел з врахування знаку числа. Різниця полягає лише в тому, що знаки числа при порівнянні не враховуються.

Загальні кроки алгоритму порівняння довгих чисел без врахуванням знаку числа можна визначити наступним чином:

- проводиться нормалізація чисел,
- якщо довжини чисел не рівні, відразу повертається результат порівняння й вийти,
- для всіх елементів від останнього до першого якщо поточні розряди не рівні повертається результат порівняння,
- вихід,
- повертання признаку рівності чисел.

Результатом роботи функції є числа -1, 0 та 1, де 0 – числа рівні, 1 – число A більше числа B, -1 – число B більше числа A.

Авторська розробка

Додатковою функцією довгої арифметиці є проведення нормалізації двох довгих чисел.

Алгоритм методу нормалізації двох довгих чисел наступний:

- порівнюється кількість знаків до коми двох чисел. якщо кількість знаків однакова – вихід,
- якщо кількість знаків різна – знаходиться різниця кількості знаків двох чисел,
- менше число доповнюється нулями справа,
- встановлюється положення коми в коротшому з двох чисел.

Авторська розробка

При додаванні двох довгих чисел реалізуємо добре відоме додавання "в стовпчик", в якому складаються окремі цифри, починаючи з молодших. Якщо сума більше основи системи числення, то відбувається перенос у наступний розряд. Для вирішення складності додавання двох різновеликих чисел з різною кількістю значущих знаків після коми перед додаванням проводиться нормалізація чисел.

Загальні кроки алгоритму додавання довгих чисел без врахуванням знаку числа можна визначити наступним чином:

- нормалізація чисел,
- вибір довжини числа-суми як максимум з довжин першого й другого числа,
- прохід від 1 до довжини числа – суми,
- підсумовування поточного розряду двох доданків,
- якщо сума більше або дорівнює основи системи обчислення, то робиться перенос у старший розряд,
- якщо був перенос із останнього розряду числа-суми, то збільшується



на 1 довжина числа-суми.

В алгоритм додавання передаються три параметри: перші два задають доданки, третій — суму. A , B і C - масиви з елементами типу long довжиною по 4 байти розмірністю в 100 елементів. Змінні i , $carry$ описані як дані типу long довжиною 4 байти.

Авторська розробка

При реалізації методу (способу) віднімання довгих чисел як алгоритмів для мікрокоманд необхідно врахувати, що при відніманні кількість цифр результату може бути набагато меншою кількості цифр зменшуваного (наприклад, $10000-9999 = 1$). Крім того, необхідно враховувати знаки чисел, та ситуацію коли зменшуване може бути менше від'ємника.

При відніманні двох довгих чисел реалізуємо віднімання в стовпчик, в якому віднімаються окремі цифри, починаючи з молодших. Якщо різниця в поточному розряді менше нуля, то відбувається займ числа зі старшого розряду. Для вирішення складності віднімання двох великих чисел з різною кількістю значущих знаків після коми перед відніманням проводиться нормалізація чисел.

Загальні кроки алгоритму віднімання довгих чисел без врахуванням знаку числа можна визначити наступним чином:

- проводиться нормалізацію чисел,
- віднімається від більшого числа менше для чого попередньо реалізується алгоритм порівняння двох довгих чисел,
- обирається довжина числа-різниці як максимум з довжин першого і другого числа,
- для проходу від 1 до довжини числа-різниці обчислюється різницю в поточному розряді,
- якщо різниця в поточному розряді менше нуля, то робиться позику із старшого розряду,
- визначається довжина числа-результату.

В алгоритм віднімання передаються три параметри: перші два задають зменшуване та від'ємник відповідно, третій — різницю. A , B і C – масиви з елементами типу long довжиною по 4 байти розмірністю в 100 елементів. Змінні i та $carry$ описані як дані типу long довжиною 4 байти.

Заключення та висновки..

В роботі було розглянуто реалізацію методів виконання базових операцій з двійковими числами поданими як масиви. Було наведено опис алгоритмів частки операцій, що визначено як базові, у випадку згаданого представлення чисел.

Задачі прогнозування стану вирішуються за даними експериментальних досліджень, значення яких завжди є сумішшю: шум, помилка вимірювання. Вже при шумі в 5% неможлива реалізація гарантоздатного програмного забезпечення прогнозування стану СТС [1, 4, 9].

Використання описаних в роботі алгоритмів методів довгої арифметики показує можливість гарантованого прогнозу для ступеню зашумлення до 7% (в деяких випадках зашумлення даних до 9%) [12, 13].



Литература:

1. Блинов А.П., Левин С.Ф. Научно-методическое обеспечение гарантированности решения метрологических задач вероятностно-статистическими методами // Измерительная техника. – 1988. – № 12. – С. 8 – 10.
2. Левин С.Ф. Гарантированность программ обеспечения эксплуатации техники. – Киев: Знание. 1989. – 23 с.
3. Левченко А.О., Войтенков Р.М. Витоки втрати працездатності систем діагностики ОВТ другого роду з представленням чисел з плаваючою комою // Сб. науч. труд. Sword. – Иваново: МАРКОВА АД, 2015. – Вип. № 1(38) Том 3. С. 4 – 11
4. Левин С.Ф. Погрешности измерений и вычислений как причина «катастрофического феномена 1985–1986 годов» в авиационной и ракетно-космической технике. – Контрольно-измерительные приборы и системы. – 2000. – № 3. – С. 21-25
5. Левченко А.О., Войтенков Р.М. Метод представлення чисел для програмних засобів інформаційних систем//Збірник 3-го науково-технічного семінару “Геоінформаційні системи в військових задачах”. – Львів: АСВ, 2012. – С. 114-120
6. Левченко А.О., Войтенков Р.М. Метод представлення чисел для програмних засобів гарантованих інформаційних технологій систем підтримки прийняття рішень для керування станом ОВТ//Збірник тез доповідей 19-ї науково-практичної конференції “Проблеми створення, розвитку та застосування інформаційних систем спеціального призначення”. – Житомир: ЖВІ НАУ, 2012. – С. 142-143.
7. Абрамов О.В., Розенбаум А.Н. Прогнозирование состояния технических систем. – М.: Наука, 1990. – 126 с.
8. Левин С.Ф. Математическая теория измерительных задач. Ч. 1-10. – Контрольно-измерительные приборы и системы. – 1999-2006.
9. С.Ф. Левин. Об основаниях теории измерительных задач [Online]. Available:http://pribory-si.ru/publication/index.php?ELEMENT_ID=5119
10. Левченко А.О. Структура інформаційної технології гарантованого розв’язання задачі прогнозу стану системи озброєння. //Проблеми створення, випробовування, застосування та експлуатації складних інформаційних систем/Збірник наукових праць. ЖВІ НАУ. – 2010.– №3. – С. 96 – 110
11. Левченко А.О. Теоретические вопросы моделирования и оценки качества систем обеспечения эксплуатации сложных технических комплексов// Системы озброєння та військової техніки. – Х.: ХУПС, 2007. – Випуск 3(11). – С. 119 – 124.
12. Левченко А.О., Войтенков Р.М. Результати моделюючого експерименту перевірки працездатності обчислювальної системи з представленням чисел з плаваючою комою// Збірка тез доповідей 3-го науково-технічного семінару “Перспективні шляхи розвитку інформаційних систем прицілювання та самонаведення високоточного озброєння РВіА”– Львів: АСВ,



2012. – С. 119.

13. Левченко А.О. Граничні точності обчислень в інформаційних системах з представленням чисел із плаваючою комою // Збірник наукових праць ВА (м. Одеса), Випуск № 2 (2), 2014 р., С.157-161.

References:

1. Blinov A.P., Levin S.F. (2012). Nauchno-metodicheskoye obespecheniye garantirovannosti resheniya metrologicheskikh zadach veroyatnostno-statisticheskimi metodami [Scientific and methodological support of guaranteed solutions to metrological problems by probabilistic and statistical methods] in *Izmeritel'naya tekhnika* [Measuring technique], vol.12, pp. 8-12
2. Levin S.F. (1989). Garantirovannost' programm obespecheniya ekspluatatsii tekhniki [Warranty of equipment maintenance programs]. Kiyv, Znanie, 223p.
3. Levchenko A.O., Voitenkov R.M. (2015). Vytoky vtraty pratsezdatsnosti system diahnostyky OVT druhoho rodu z predstavlennyam chysel z plavayuchoyu komoyu [The origins of the disability of second-line ATS diagnostic systems with floating point representation] in *Naučnye trudy SWorld* [Scientific works SWorld], issue 38, vol.3, pp. 4-11
4. Levin S.F. (2000). Pogreshnosti izmereniy i vychisleniy kak prichina «katastroficheskogo fenomena 1985–1986 godov» v aviatsionnoy i raketno-kosmicheskoy tekhnike [Errors of measurements and calculations as the cause of the “catastrophic phenomenon of 1985-1986” in aviation and rocket and space technology] in *Kontrol'no-izmeritel'nyye pribory i sistemy* [Instrumentation and systems], vol.3, pp. 21-15
5. Levchenko A.O., Voitenkov R.M. (2012). Metod predstavleniya chysel dlya proqramnykh zasobiv informatsiynykh system [Number representation method for information systems software] in *Zbirnyk 3-ho naukovo-tekhnichnoho seminaru “Heoinformatsiyni systemy v viys'kovykh zadachakh”* [Collection of the 3rd scientific-technical seminar "Geoinformation systems in military tasks"], pp. 114-120
6. Levchenko A.O., Voitenkov R.M. (2012). Metod predstavleniya chysel dlya proqramnykh zasobiv harantozdatnykh informatsiynykh tekhnolohiy system pidtrymky pryynyattya rishen' [A method of representing numbers for software-enabled information technology decision support systems] in *Zbirnyk 19-yi naukovo-praktychnoyi konferentsiyi “Problemy stvorenniya, rozvytku ta zastosuvannya informatsiynykh system spetsial'noho pryznachennya”* [Collection of the 19th scientific-practical conference "Problems of creation, development and application of special purpose information systems"], pp. 142-143.
7. Abramov O.V., Rozenbaum A.N. (1990). Prognozirovaniye sostoyaniya tekhnicheskikh system [Forecasting the state of technical systems]. Moskva, Nauka, 126 p.
8. Levin S.F. (2006). Matematicheskaya teoriya izmeritel'nykh zadach [The mathematical theory of measurement problems] in *Kontrol'no-izmeritel'nyye pribory i sistemy* [Instrumentation and systems], vol.3, pp. 21-15
9. Levin S.F. [Online]. Ob osnovaniyakh teorii izmeritel'nykh zadach [On the foundations of the theory of measurement problems]. Available:http://pribory-si.ru/publication/index.php?ELEMENT_ID=5119
10. Levchenko A.O. (2010). Struktura informatsiynoyi tekhnolohiyi harantovanoho rozv'yazannya zadachi prohnozu stanu systemy ozbroynennya [The structure of information technology of the guaranteed solution of the task of forecasting the status of the weapons system] in *"Problems of creation, testing, application and operation of complex information systems" Naučnye trudy ZVI NAU* [Scientific works of the National aviation academy Military institute], issue 3, vol.3, pp. 96-110.
11. Levchenko A.O. (2007). Teoreticheskiye voprosy modelirovaniya i otsenki kachestva sistem obespecheniya ekspluatatsii slozhnykh tekhnicheskikh kompleksov [Theoretical questions of modeling and quality assessment of systems of operation of complex technical complexes] in



Systemy ozbroynnya ta viys'kovoyi tekhniky [Weapons and military equipment], vol.3 (11), pp. 119 – 124.

12. Levchenko A.O., Voitenkov R.M. (2012). Rezul'taty modelyuyuchoho eksperymentu perevirky pratsездatnosti obchyslyval'noyi systemy z predstavlennyam chysel z plavayuchoyu komoyu [Результати моделюючого експерименту перевірки праездатності обчислювальної системи з представленням чисел з плаваючою комою] in *Zbirnyk 3-ho naukovo-tekhnichnoho seminaru "Heoinformatsiyni systemy v viys'kovykh zadachakh"* [Collection of the 3rd scientific-technical seminar "Geoinformation systems in military tasks"], pp. 129-130.

13. Levchenko A.O. (2014). Hranychni tochnosti obchyslen' v informatsiynykh systemakh z predstavlennyam chysel iz plavayuchoyu komoyu [Marginal accuracy of calculations in information systems with floating point representation] in *Naučnye trudy VA* [Scientific works of the Military academy], issue 2, vo2, pp. 157-161.

Abstract. *The work is aimed at substantiation of a set of elementary operations for representation of binary numbers as arrays, which in the future will create preconditions for the development of a library of software products of exact calculations. The expediency of representing binary numbers as arrays for some practical tasks is shown. Also described are algorithms for performing some basic operations with the above representation of numbers.*

Key words: *arithmetic operations, multidimensional data, representation of long numbers in binary code.*

Статья отправлена: 08.09.2019 г.

© Левченко А.О.