

Самощенко О.В. Державний університет телекомунікацій, Київ

СИНТЕЗ КОМПОЗИТНОЇ СХЕМИ ДОДАВАННЯ ТА ВІДНІМАННЯ ЦІЛИХ ЧИСЕЛ В КОДАХ ІЗ ДОДАТНИМ НУЛЕМ

При синтезі схеми додавання та віднімання алгебраїчних цілих чисел використовується система кодування операндів із додатним нулем. Запропонований метод комп'ютерного подання цілих чисел, згідно з яким коди додатних та від'ємних чисел формуються за єдиною процедурою. Синтез базується на поданні суми та різниці даних у формі залишку по модулю, що дорівнює ваговому коефіцієнту вихідного переносу суматора. Аналітично підтверджено факт формування результату на виходах суматора як в доповняльному коді, так і в коді із додатним нулем. Визначені області припустимих значень результатів для операцій додавання і віднімання коректних вхідних даних. Наявність і тип переповнення розрядної сітки установлюється за результатами аналізу комбінацій сигналів на штатних виходах операційного суматора. Обчислення суми і різниці чисел із використанням зміщеного коду базується на формалізованому описі арифметичних операцій над цілими числами згідно із запропонованим форматом. Аналітично обґрунтована схемотехнічна однорідність операційного суматора. Наведені правила встановлення правильності виконання операцій додавання і віднімання зміщених цілих чисел. Для дійсних значень початкових аргументів отримані діапазони кодів сум і різниць, запропоновані правила ідентифікації позитивних і негативних переповнень. Схеми фіксації переповнення кількості розрядів основних виводів операційного суматора інваріантні відносно операцій додавання та віднімання початкових даних в системі із додатним нулем. Оригінальне використання чисельного зсуву при кодуванні операндів, що проявляється у перевагах технічного забезпечення базових комп'ютерних операцій, зумовлює позитивні властивості при практичних реалізаціях більш складних арифметичних дій.

Ключові слова: суматор, код із додатним нулем, залишок по модулю, ознаки переповнення, доповняльний код.

Samoshchenko O.V. State University of Telecommunications, Kyiv

SYNTHESIS OF COMPOSITE SCHEM ADDITION AND SUBTRACTION OF INTEGERS IN CODES WITH POSITIVE ZERO

Code on the outputs of adder binary numbers described as the remainder of the sum the initial data on the adder module is equal to output carry weight. An original technique for synthesizing a way of operands representation in the addition and subtraction schemes of integers in a code with a positive zero was developed, which is based on the representation the source data in the form a remainder on the adder module. A method of computer representation for integer numbers is proposed, in which the codes of positive and negative numbers are formed by the same procedure. The property of duality the addition and subtraction operations on the initial data in the code with a positive zero have justified analytically, which is based on the fact that certain adder outputs union give the result just as in the two's complement so too in the positive zero codes. Areas of allowable results values for the correct input data addition and subtraction operations are determined. It is identified combination of the adder output signals, which determine the presence and polarity the adder bit grid overflow. It is shown that designed fixing scheme bit grid overflow of adder outputs invariant with respect to operations of addition and subtraction of source data with a positive zero code. For the analytical description of arithmetic operations on integer numbers represented with the proposed encoding method, a technique of calculating the sum and difference of numbers using the biased supplementary code has been proposed. Analytically substantiated, that the technique makes the scheme of the operational adder homogeneous. The rules for establishing the correctness of the addition and subtraction operations of the integers given in the proposed encoding form are given. For true values of the initial arguments, the sums and the differences codes ranges are obtained, and the rules for positive and negative overflows identi-

fication are proposed. The original usage of a common numerical bias during the operands encoding, that evinces itself in the advantages of basic computer operations technical implementation, predetermines positive properties in practical implementations of more complex arithmetical actions.

Keywords: adder binary numbers, code with positive zero, the remainder modulo, signs of overflow, two's complement code.

Самощенко А.В. Государственный университет телекоммуникаций, Киев

СИНТЕЗ КОМПОЗИТНОЙ СХЕМЫ СУММИРОВАНИЯ И ВЫЧИТАНИЯ ЦЕЛЫХ ЧИСЕЛ В КОДАХ С ПОЛОЖИТЕЛЬНЫМ НУЛЕМ

При синтезе схемы сложения и вычитания алгебраических целых чисел используется система кодирования операндов с положительным нулем. Предложен метод компьютерного представления целых чисел, согласно которому коды положительных и отрицательных чисел формируются по единой процедуре. Синтез основан на представлении суммы и разности в виде остатка по модулю, равному весовому коэффициенту выходного переноса сумматора. Аналитически подтверждается факт формирования результата на выходах сумматора как в дополнительном коде, так и в коде с положительным нулем. Определены области допустимых значений результатов для операций сложения и вычитания корректных входных данных. Наличие и тип переполнения разрядной сетки устанавливается по результатам анализа комбинаций значений сигналов на штатных выходах операционного сумматора. Вычисление суммы и разности чисел с использованием смещенного кода базируется на формализованном описании арифметических операций над целыми числами согласно предложенному формату. Аналитически обоснована схмотехническая однородность операционного сумматора. Приведены правила установления правильности выполнения операций сложения и вычитания смещенных целых чисел. Для действительных значений начальных аргументов получены диапазоны кодов сумм и разностей, предложены правила идентификации положительных и отрицательных переполнений. Схемы фиксации переполнения количества разрядов основных выводов операционного сумматора инвариантны относительно операций суммирования и вычитания исходных данных в системе с положительным нулем. Оригинальное использование численного смещения при кодировании операндов, проявляющееся в преимуществах технического обеспечения базовых компьютерных операций, предопределяет положительные свойства при практических реализациях более сложных арифметических действий.

Ключевые слова: сумматор, код с положительным нулем, остаток по модулю, признаки переполнения, дополнительный код.

Вступ.

Використання доповняльних кодів при перетворенні комп'ютерних даних у форматі цілого обґрунтовано відносною простотою схем і алгоритмів відповідних пристроїв [1-3]. Практична реалізація арифметичних дій вимагає обчислення основного результату і обов'язкове формування ознак коректності завершення операцій. Задоволення вказаних вимог пов'язане із необхідністю раціонального використання апаратних ресурсів. Для підвищення ефективності виконання операцій в комп'ютерному обладнанні використовується подання операндів із зміщеними кодами [4-7], згідно із яким передбачається арифметична зміна значення операндів відповідно до деякої константи. Така зміна значення, тобто умовне зміщення розташування змінної на числової висі, дозволяє, наприклад, використовувати алгоритми комп'ютерних перетворювань незначних чисел для обробки алгебраїчних чисел. Одним із варіантів зміщеного кодування є подання комп'ютерних даних кодами із додатним нулем [5,6].

В роботах [8-11] наведені декілька підходів до перетворення чисельних даних, але вони відносяться, зокрема, до схмотехнічної реалізації арифметичних пристроїв [8,9] чи спрямовані на покращення лише специфічних дій [10,11].

Врахування властивостей кодів із додатним нулем дозволяє розробити комп'ютерні арифметичні операційні пристрої із технічними перевагами. Підвищення складності комп'ютерної обчислювальної техніки потребує моделювання операцій перетворення даних, що є невід'ємною частиною процесу успішного проектування апаратно-програмного обладнання. В свою чергу, моделювання ґрунтується на формалізованому опису відповідних операційних структур і процесів перетворення, базовими із яких беззаперечно є додавання та віднімання. Використання для формалізованого опису композитних схем, що дозволяють аналізувати і досліджувати виконання декількох операцій в єдиному операційному пристрої, є доречним.

В роботі з використанням властивостей подання цілих чисел в кодї із додатним нулем та операційних властивостей суматорів цілих чисел запропоновані композитний арифметичний пристрій та ланцюги керування для обробки операндів в кодах із додатним нулем. Схема формує результат, який можна трактувати в системі кодування із додатним нулем, так і в доповняльному кодї. Для всіх режимів роботи схема фіксує наявність та тип переповнення результату.

Загальна методика роботи суматора в кодї із додатним нулем.

Код повного результату на виводах двійкового суматора визначається операційними властивостями згідно співвідношень [12,13]:

$$E \cdot 2^n + S(n, 1) = a(n, 1) + b(n, 1) + e,$$

де $S(n, 1)$ – основні розряди результату на виводах суматора; $a(n, 1)$ та $b(n, 1)$ - коди початкових даних на входах суматора (операнди); n – кількість основних розрядів суматора ($n, 1 = n, n-1, \dots, 2, 1$); e, E – відповідно вхідний та вихідний перенос суматора; $2^n = V$ – ваговий коефіцієнт вихідного переносу суматора;

Двійковий код на основних виводах суматора визначається залежністю:

$$S(n, 1) = (a(n, 1) + b(n, 1) + e)_{mV}, \quad (1)$$

де $(x)_{mV}$ – залишок по модулю V повного коду результату.

У загальному випадку, операція додавання (віднімання) цілих чисел зводиться до розв'язання задачі

$$C = A \pm B, \quad (2)$$

де A, B – початкові числові данні (операнди); C – код суми (різниці) операндів.

Для апаратного розв'язання задачі (2) у системі подання операндів і результату у системі кодування із додатним нулем використовується суматор (рис.1) [6].

При додаванні ($SUB=0$) та відніманні ($SUB=1$) цілих чисел в кодї із додатним нулем робота операційного суматора визначається формулами:

$$\begin{aligned} a(n,1) &= A^{DH}(n,1); \\ b(n,1) &= B^{DH}(n,1) \oplus SUB = \begin{cases} B^{DH}(n,1) & \text{при } SUB=0, \\ \overline{B^{DH}(n,1)} & \text{при } SUB=1; \end{cases} \\ e = SUB &= \begin{cases} 0 & \text{при } SUB = 0, \\ 1 & \text{при } SUB = 1. \end{cases} \end{aligned}$$

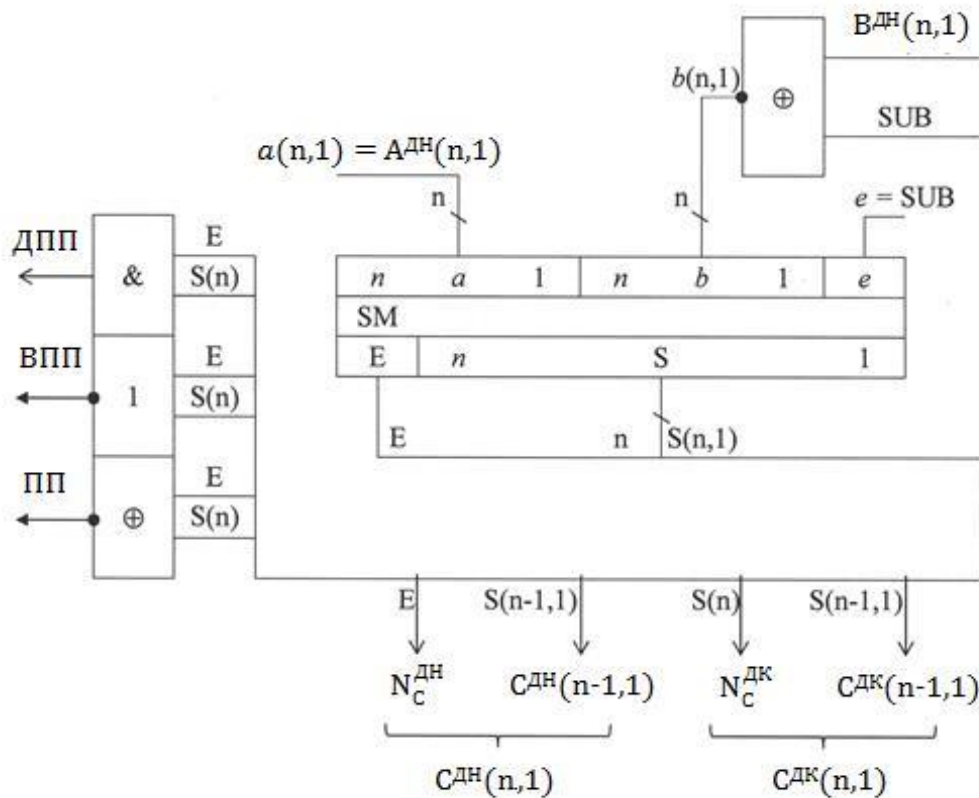


Рис. 1. Композитний суматор цілих чисел в коді із додатним нулем

Особливості подання операндів в коді із додатним нулем

Результат на основних виходах операційного суматора $S(n,1)$ розраховується відповідно до правила вирахування додатного зміщеного коду:

$$C^{cm}(n, 1) = V/2 + C, \tag{3}$$

де $V/2 = 2^{n-1}$ – код зміщення в n -розрядному операційному суматорі; $C^{cm}(n,1)$ – зміщений код алгебраїчної суми (різниці) операндів A та B .

Алгебраїчна сума (різниця) C на основних виводах операційного суматора визначається множиною чисел

$$C \in [-V/2; V/2 - 1]. \tag{4}$$

Таким чином,

$$C^{cm}(n, 1) \in [0; V - 1].$$

Максимальне та мінімальне значення зміщеного коду (3) обмежені й визначаються нерівностями [5,6]:

$$C^{cm}(C > 0, \in [0; V/2 - 1]) \geq 2^{n-1},$$

$$C^{cm}(C < 0, \in [-V/2; -1]) < 2^{n-1}.$$

Значення розряду $C^{cm}(n)$ зміщеного коду однозначно характеризує знак алгебраїчної суми:

$$C^{cm}(n) = \begin{cases} 1 \text{ при } C > 0 : C \in [1; V/2 - 1], \\ 1 \text{ при } C = 0, \\ 0 \text{ при } C < 0 : C \in [-V/2; -1]. \end{cases} \tag{5}$$

Згідно (5), в старшій розрядній позиції зміщеного коду $C^{cm}(n)$ знак «+» при нульовому та додатному значенні алгебраїчної суми (різниці) C вказується цифрою «1», а знак «-» – цифрою «0». Нульове значення суми кодується як додатне число. Через це, зміщений код (3)

тракується як код із додатним нулем.

Згідно (3), значення (2) в коді із додатним нулем визначається співвідношенням:

$$C^{DH}(n,1) = V/2 + C, \quad (6)$$

де $C^{DH}(n) = N_C^{DH}$ – розряд знаку алгебраїчної суми (різниці) в коді із додатним нулем:

$$N_C^{DH} = C^{CM}(n) = \begin{cases} 1 & \text{при } C > 0, \\ 1 & \text{при } C = 0, \\ 0 & \text{при } C < 0. \end{cases}$$

При поданні алгебраїчної суми (різниці) в коді із додатним нулем розв'язання задачі (2), згідно (6), зводиться до вираховання співвідношення:

$$\begin{aligned} C^{DH}(n,1) &= V/2 + C = V/2 + (A \pm B) = (V/2 + (V/2 + A) \pm (V/2 + B))_{mV} = \\ &= (V/2 + A^{CM} \pm B^{CM})_{mV} = (V/2 + C^{AB}(n,1))_{mV}, \end{aligned} \quad (7)$$

де $A^{CM} = V/2 + A$; $B^{CM} = V/2 + B$.

При обчисленні алгебраїчної суми $(A + B)$ в системі із додатним нулем

$$C^{AB}(n,1) = (V/2 + A + V/2 + B)_{mV} = (V + (A + B))_{mV} = (V + C)_{mV}. \quad (8)$$

При обчисленні алгебраїчної різниці $(A - B)$ в коді із додатним нулем

$$C^{AB}(n,1) = (A^{CM} - 1 - B^{CM} + 1)_{mV} = (A^{CM} + (V - 1 - B^{CM}) + 1)_{mV}. \quad (9)$$

За визначенням [5] на множині значень (4) права частина виразів (8,9) подає алгебраїчну суму (різницю) в системі доповняльних кодів:

$$(V + C)_{mV} = C^{DK}(n,1) = C^{AB}(n,1),$$

де $C^{DK}(n,1)$ – зображення алгебраїчної суми (різниці) в доповняльному коді; $C^{DK}(n-1,1)$ – основні розряди доповняльного коду алгебраїчної суми (різниці); $C^{DK}(n) = N_C^{DK}$ – розряд знаку алгебраїчної суми (різниці) C в форматі доповняльного коду довжиною n біт:

$$N_C^{DK} = \begin{cases} 0 & \text{при } C \geq 0, \\ 1 & \text{при } C < 0. \end{cases}$$

Вираз (7) набуває вигляду:

$$C^{DH}(n,1) = (V/2 + C^{DK}(n,1))_{mV}, \quad (10)$$

де

$$C^{DK}(n,1) = \begin{cases} ((V/2 + A) + (V/2 + B))_{mV} \\ (A^{CM} + (V - 1 - B^{CM}) + 1)_{mV} \end{cases}. \quad (11)$$

Для рішення задачі (10) при $V = 2^n$ коди $C^{DH}(n,1)$ та $C^{DK}(n,1)$ подаються поліномами степеню $(n-1)$, за результатом чого компоненти полінома $C^{DH}(n,1)$ через складові полінома $C^{DK}(n,1)$ визначаються співвідношеннями [6]:

$$N_C^{DH} = \overline{N_C^{DK}}, \quad C^{DH}(n-1,1) = C^{DK}(n-1,1). \quad (12)$$

Для вираховання складових полінома $C^{DK}(n,1)$ через початкові дані A та B задачі (2), згідно (11,12), на входах операційного суматора необхідно сформувати двійкові коди:

$$a(n,1) = \begin{cases} \frac{V}{2} + A = A^{DH}(n,1) & \text{при } (A + B); \\ A^{CM} = \frac{V}{2} + A = A^{DH}(n,1) & \text{при } (A - B), \end{cases} \quad (13)$$

$$b(n,1) = \begin{cases} \frac{V}{2} + B = B^{DH}(n,1) \text{ при } (A+B); \\ V-1-B^{CM}(n,1) = \overline{B^{CM}(n,1)} = \overline{\left(\frac{V}{2} + B\right)} = \overline{B^{DH}(n,1)} \text{ при } (A-B), \end{cases}$$

$$e = \begin{cases} 0 \text{ при } (A+B); \\ 1 \text{ при } (A-B), \end{cases}$$

де $A^{DH}(n,1)$, $B^{DH}(n,1)$ – відповідно зображення початкових даних A та B задачі (2) в кодї із додатним нулем довжиною n біт; $A^{DH}(n-1,1)$, $B^{DH}(n-1,1)$ – відповідно основні розряди початкових даних A та B , що вказано кодом із додатним нулем; $A^{DH}(n) = N_A^{DH}$, $B^{DH}(n) = N_B^{DH}$ – відповідно знакові розряди початкових даних A та B в кодї із додатним нулем:

$$N_A^{DH} = \begin{cases} 1 \text{ при } A > 0, \\ 1 \text{ при } A = 0, \\ 0 \text{ при } A < 0; \end{cases} \quad N_B^{DH} = \begin{cases} 1 \text{ при } B > 0, \\ 1 \text{ при } B = 0, \\ 0 \text{ при } B < 0. \end{cases}$$

При розрахунку суми (різниці) цілих чисел A та B в системі із додатним нулем в арифметичних ланцюгах пристрою формуються двійкові коди:

$$S(n,1) = C^{DK}(n,1) = \begin{cases} (A^{DH}(n,1) + B^{DH}(n,1) + 0)_{mV} \text{ при } (A+B); \\ (A^{DH}(n,1) + \overline{B^{DH}(n-1,1)} + 1)_{mV} \text{ при } (A-B), \end{cases}$$

$$S(n) = C^{DK}(n) = N_C^{DK}, \tag{14}$$

$$N_C^{DK} = \overline{N_C^{DK}} = \overline{S(n)},$$

$$S(n-1,1) = C^{DK}(n-1,1) = C^{DH}(n-1,1).$$

Формування ознак коректності результату

Область приналежності функцій $A^{DH}(A)$, $B^{DH}(B)$ та $C^{DK}(C)$ однакова в n -розрядному суматорі (13) та визначається, згідно (4,14), інтервалом $[-V/2; V/2-1]$. За коректних значень початкових даних $A \in [-V/2; V/2-1]$ та $B \in [-V/2; V/2-1]$ значення $C = A \pm B$ у загальному випадку визначається більш широким інтервалом. При додаванні результат приналежить області $[-V; V-2]$, а при відніманні різниця операндів визначається інтервалом $[-(V-1); (V-1)]$ (рис.2).

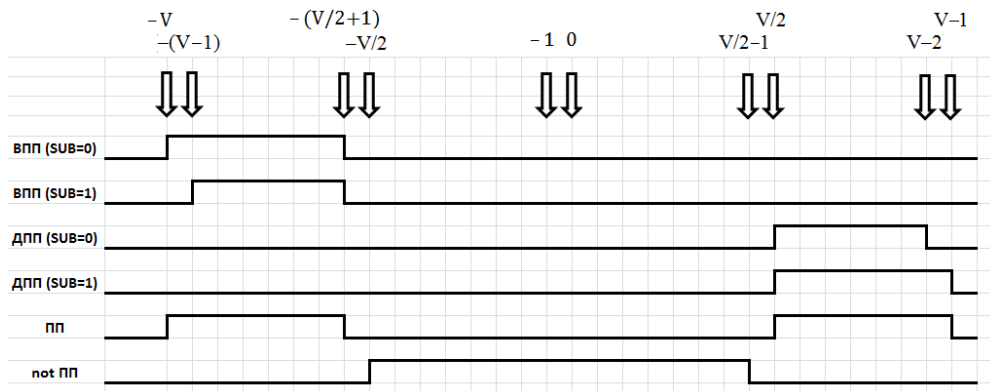


Рис. 2. Области значень ознак переповнення результату в кодї із додатним нулем

Отже, при $C \in [-V; -(V/2+1)]$, в схемі додавання необхідно фіксувати від’ємне, а при $C \in [V/2; V-2]$, – додатне переповнення розрядної сітки суматора. В свою чергу, при розрахунку різниці, якщо $C \in [-(V-1); -(V/2 + 1)]$, то необхідно фіксувати від’ємне переповнення (ВПП), а при $C \in [V/2; V - 1]$ – додатне переповнення (ДПП) розрядної сітки суматора.

Алгоритм фіксації переповнення при додаванні (SUB=0) визначається нерівностями:

$$\begin{aligned} \text{ВПП} &= \begin{cases} 1 \text{ при } C \in [-V; -(V/2 + 1)]; \\ 0 \text{ при } C \in [-V/2; -1]; \end{cases} \\ \text{ДПП} &= \begin{cases} 1 \text{ при } C \in [V/2; V - 2]; \\ 0 \text{ при } C \in [0; V/2 - 1]; \end{cases} \\ \text{ПП} &= \begin{cases} 1 \text{ при } C \in [-V; -(V/2 + 1)] \vee [V/2; V - 2]; \\ 0 \text{ при } C \in [-V/2; V/2 - 1], \end{cases} \end{aligned} \quad (15)$$

де ПП – ознака загального переповнення операційного суматора при додаванні.

При додаванні цілих чисел в системі із додатним нулем алгоритм фіксації переповнення розрядної сітки суматора (1,2) описується логічними виразами [6]:

$$\begin{aligned} \text{ВПП} &= \bar{E} \wedge \overline{S(n)}; \\ \text{ДПП} &= E \wedge S(n); \end{aligned} \quad (16)$$

$$\text{ПП} = \text{ВПП} \vee \text{ДПП} = \bar{E} \wedge \overline{S(n)} + E \wedge S(n) = \overline{E \oplus S(n)}.$$

Комбінації значень ознак переповнення та знакових розрядів при додаванні (відніманні) цілих чисел в кодах із додатним нулем показано в табл.1.

Таблиця 1

Ознаки переповнення та значень знакових розрядів

E	S(n)	ВПП	ДПП	ПП	N _C ^{ДК}	N _C ^{ДН}
0	0	1	0	1	*	*
1	1	0	1	1	*	*
0	1	0	0	0	1	0
1	0	0	0	0	0	1

За відсутності переповнення комбінації старших розрядів повної суми ES(n) та знакових розрядів N_C^{ДК} та N_C^{ДН} утворюють набори, згідно за якими впливає, що розряди знаку результату в системі із додатним нулем та в доповняльному коді визначаються виразами:

$$N_C^{ДК} = S(n), \quad N_C^{ДН} = E. \quad (17)$$

Алгоритм фіксації переповнення при відніманні (SUB=1) визначається нерівностями:

$$\begin{aligned} \text{ВПП} &= \begin{cases} 1 \text{ при } C \in [-(V - 1); -(V/2 + 1)], \\ 0 \text{ при } C \in [-V/2; -1]; \end{cases} \\ \text{ДПП} &= \begin{cases} 1 \text{ при } C \in [V/2; V - 1], \\ 0 \text{ при } C \in [0; V/2 - 1]; \end{cases} \\ \text{ПП} &= \begin{cases} 1 \text{ при } C \in [-(V - 1); -(V/2 + 1)] \vee [V/2; V - 1], \\ 0 \text{ при } C \in [-V/2; V/2 - 1]. \end{cases} \end{aligned} \quad (18)$$

Незважаючи на різні інтервали визначення переповнення, за наявності від’ємного та додатного переповнення різниці (A – B), ознаки переповнення співпадають з відповідними ознаками, що характеризують переповнення суми (A + B). В силу цього, вирази (16) описують алгоритм контролю переповнення як в схемі додавання, так і в схемі віднімання початкових даних в кодах із додатним нулем.

Коректні значення суми (A + B) та різниці (A – B), згідно (15) та (18), задаються однаковим інтервалом [-V/2; V/2 – 1], а двійковий код суми та різниці визначається тотожно:

$$E \cdot 2^n + S(n,1) = V + (A \pm B) = V + C.$$

Тому умовами відсутності переповнення різниці ($A - B$) є вирази, що характеризують ознаки відсутності переповнення суми початкових даних ($A + B$). Зважаючи на це, розряди знаку різниці ($A - B$) в системі із додатним нулем та в доповняльному коді визначаються логічними виразами (17), що сформовані для схеми додавання початкових даних в коді із додатним нулем.

Висновки

Основні результати досліджень зводяться до наступного:

– технічною основою операційного пристрою є композитний суматор для об'єднаної обробки цифрових і знакових розрядів операндів та аналізу результату виконання операцій додавання і віднімання;

– реалізований метод синтезу в системі обробки операндів із додатним нулем, що ґрунтується на композитному поданні суми та різниці початкових даних як залишку по модулю суматора;

– аналітично доведена властивість подання результату сполученнями на штатних виходах операційного суматора як в доповняльному коді, так і в коді із додатним нулем;

– виявлені комбінації штатних вихідних змінних операційного суматора, що визначають наявність та тип переповнення розрядної сітки основних виводів суматора;

– підтверджено, що розроблені схеми фіксації переповнення кількості розрядів основних виводів суматора інваріантні відносно операцій додавання та віднімання початкових даних в системі із додатним нулем;

– розроблена композитна схема додавання та віднімання цілих чисел в системі із додатним нулем створює передумови для успішного проектування пристроїв виконання більш складних операцій.

Список використаної літератури

1. Patterson, David A. Computer organization and design: the hardware/software interface / David A. Patterson, John L. Hennessy. – 5th ed. – 2014. – (The Morgan Kaufmann series in computer architecture and design).

2. Баби́ч, Н.П. Основы цифровой схемотехники / Н.П.Баби́ч, И.А.Жуков. – ДМК-Пресс, 2016. – 480с.

3. Жмакин А.П. Архитектура ЭВМ. 2-е изд. – СПб.: БХВ-Петербург, 2010. – 352с.

4. Intel® 64 and IA-32 Architectures Software Developer's Manual. - Order Number: 325462-067US, May 2018.

5. Святный В.А. Математическое описание компьютерных операций суммирования и вычитания целых чисел при смещенных кодах операндов / В.А.Святный, В.В.Лапко, А.В.Сабощенко. // Наукові праці Донецького національного університету. Серія «Інформатика, кібернетика та обчислювальна техніка». – Красноармійськ: ДонНТУ. - 2016. – Вип.1(22). – с.75-83.

6. Лапко В.В., Самощенко О.В., Маргієв Г.Е. Синтез та математичне моделювання схем додавання та віднімання цілих чисел в кодах з додатним нулем. // Наукові праці Донецького національного університету. Серія “Інформатика, кібернетика та обчислювальна техніка”. Випуск 1 (24) – Покровськ: ДонНТУ. – 2017. - с.12-21.

7. Самощенко О.В., Мірошкін О.М., Маргієв Г.Е. Синтез та дослідження схем додавання та віднімання цілих чисел в системі з від'ємним нулем. // Наукові праці Донецького національного технічного університету. Серія “Інформатика, кібернетика та обчислювальна техніка”. – Випуск 1(26). - Покровськ: ДонНТУ. – 2018. - с.91-100.

8. Yadav, Priyanka, et al. “Design and Implementation of 4-Bit Arithmetic and Logic Unit Chip with the Constraint of Power Consumption.” IOSR Journal of Electronics and Communication

Engineering, vol. 9, no. 3, 2014, pp. 36–43, 10.9790/2834-09353643. Accessed 31 Mar. 2020.

9. Yellampalli, Siva. “Basic Operation Performed on Arithmetic Logic Unit (ALU) For 32-Bit Floating Point Numbers: (Initial Results).” *International Journal of Applied Engineering Research*, vol. 12, 2017, pp. 3248–3252, ISSN 0973-4562. Accessed 31 Mar. 2020.

10. Brodal, Gerth Stolting, et al. “Integer Representations towards Efficient Counting in the Bit Probe Model.” *Journal of Discrete Algorithms*, vol. 26, May 2014, pp. 34–44, 10.1016/j.jda.2013.11.001. Accessed 31 Mar. 2020.

11. Calderon, Humberto, et al. “Reconfigurable Universal Adder.” 2007 IEEE International Conf. on Application-Specific Systems, Architectures and Processors (ASAP), July 2007, 10.1109/asap.2007.4429978. Accessed 31 Mar. 2020.

12. Cragon, H. G. (2000) *Computer Architecture and Implementation*. University of Texas at Austin, Cambridge University Press.

13. Tanenbaum, A. S., Austin, T. (2012) *Structured Computer Organization*. University of Michigan, Ann Arbor, Michigan, United States.

References

1. Patterson, David A. (2014) *Computer organization and design: the hardware/software interface* / David A. Patterson, John L. Hennessy. - 5th ed. (The Morgan Kaufmann series in computer architecture and design).

2. Babych, M. P., Zhukov, I. A. (2004) *Computer circuit design*, MK-Pres, Kyiv..

3. Zhmakyn, A. P. *Computer architecture*. 2th ed. (2010), SPB, BHV-Peterburg, 352p.

4. Intel® 64 and IA-32 Architectures Software Developer’s Manual. Order Number: 325462-067US, May 2018.

5. Svyatnyy, V. A., Lapko, V. V., Samoshchenko, O. V. (2016) Mathematical description of computer operations of summation and subtraction of integers with offset operand codes. *Naukovi pratsi DonNTU: Informatyka, Kybernetyka ta obchysliuvalna tekhnika*, # 1 (22), Donetsk National Technical University, Krasnoarmiysk.

6. Lapko, V. V., Samoshchenko, O. V., Marhiev, H. E. (2017) Synthesis and mathematical modeling of addition and subtraction schemes of integers in the codes with positive zero. *Naukovi pratsi DonNTU: Informatyka, Kybernetyka ta obchysliuvalna tekhnika*, # 1 (24), Donetsk National Technical University, Pokrovsk.

7. Samoshchenko, O. V., Мірошкін О.М., Marhiev, H. E. (2018) Synthesis and research of addition and subtraction schemes for negative zero coded integers. *Naukovi pratsi DonNTU: Informatyka, Kybernetyka ta obchysliuvalna tekhnika*, # 1 (26), Donetsk National Technical University, Pokrovsk.

8. Yadav, Priyanka, et al. “Design and Implementation of 4-Bit Arithmetic and Logic Unit Chip with the Constraint of Power Consumption.” *IOSR Journal of Electronics and Communication Engineering*, vol. 9, no. 3, 2014, pp. 36–43, 10.9790/2834-09353643. Accessed 31 Mar. 2020.

9. Yellampalli, Siva. “Basic Operation Performed on Arithmetic Logic Unit (ALU) For 32-Bit Floating Point Numbers: (Initial Results).” *International Journal of Applied Engineering Research*, vol. 12, 2017, pp. 3248–3252, ISSN 0973-4562. Accessed 31 Mar. 2020.

10. Brodal, Gerth Stolting, et al. “Integer Representations towards Efficient Counting in the Bit Probe Model.” *Journal of Discrete Algorithms*, vol. 26, May 2014, pp. 34–44, 10.1016/j.jda.2013.11.001. Accessed 31 Mar. 2020.

11. Calderon, Humberto, et al. “Reconfigurable Universal Adder.” 2007 IEEE International Conf. on Application-Specific Systems, Architectures and Processors (ASAP), July 2007, 10.1109/asap.2007.4429978. Accessed 31 Mar. 2020.

12. Cragon, H. G. (2000) *Computer Architecture and Implementation*. University of Texas at Austin, Cambridge University Press.

13. Tanenbaum, A. S., Austin, T. (2012) *Structured Computer Organization*. University of Michigan, Ann Arbor, Michigan, United States.