

УДК 681.3.042

И.Н. Федотова-Пивень

Черкасский государственный технологический университет, Черкассы

## СОВМЕЩЕННОЕ ВО ВРЕМЕНИ СУММИРОВАНИЕ 50-ТИ ЦЕЛЫХ ПОЛОЖИТЕЛЬНЫХ ЧИСЕЛ В РЕКУРРЕНТНОЙ СИСТЕМЕ СЧИСЛЕНИЯ

В статье в рамках программной модели рассмотрено сравнение быстродействий совмещенного во времени сложения 50-ти целых положительных 16-разрядных чисел в линейной избыточной рекуррентной системе счисления третьего порядка с алфавитом  $\{0, 1\}$ , образованной рекуррентным соотношением  $V_n = V_{n-1} + 3V_{n-3} + 2V_{n-4}$  с начальными значениями 1 1 1 2 4 8 и поочередного сложения 50-ти целых положительных 16-разрядных двоичных чисел по стандартному алгоритму Уоллеса. Из работы программной модели следует, что совмещенное во времени сложение 50-ти 16-разрядных целых положительных чисел в указанной рекуррентной системе счисления выполняется в 8,6 раза быстрее и требует в 7,3 раза меньше ячеек памяти для хранения промежуточных результатов вычислений, чем последовательное сложение по классическому алгоритму Уоллеса.

**Ключевые слова:** избыточность, структурно-блочные коды, рекуррентные системы счисления, совмещенное во времени многооперандное сложение, вертикальные разрядные срезы.

### Введение

**Постановка проблемы.** Проблема повышения производительности современных вычислительных устройств (ВУ) является одной из наиболее актуальных проблем в современной электронике. Для этого создаются более быстрые аппаратные устройства и алгоритмы. Последние возможно создать путем применения новых избыточных систем счисления [1] и структурно-блоковых кодов [2]. В вычислительной математике операция сложения многих чисел является достаточно часто используемой арифметической операцией (например, при умножении больших чисел складываются промежуточные результаты умножения, при вычислении с заданной точностью трансцендентных функций находится сумма определенного количества членов разложения в ряд этих функций [3] и др.). На аппаратном уровне многооперандное сложение является фундаментальной проблемой [4, 5], поскольку с ростом числа операндов возрастает логическая сложность сумматора [6] и одним из возможных способов решения этой проблемы является использование алгоритмов совмещенного во времени выполнения операций [7]. Таким образом, разработка ВУ для совмещенного во времени многооперандного сложения в избыточных системах счисления является актуальной задачей.

**Анализ последних исследований и публикаций.** В предыдущих исследованиях рассматривалось многооперандное совмещенное во времени сложение в рекуррентных избыточных линейных системах счисления третьего порядка [8 – 10]. Полученные программные модели совмещенного во времени сложения 5-ти и 20-ти слагаемых на основе свойств линейных избыточных рекуррентных систем счисления

третьего порядка на 73% (для 20-ти 32-разрядных слагаемых) и на 79% (для 20-ти 16-разрядных) соответственно быстрее, чем аналогичное сложение с использованием алгоритма Уоллеса [11].

Вопрос совмещенного сложения более 20-ти двоичных слагаемых в линейной избыточной рекуррентной системе счисления (ЛИРССЧ) не рассматривался.

**Целью данной работы** является сравнение с помощью программной модели времен вычисления суммы 50-ти целых положительных чисел в обычной двоичной системе счисления с применением алгоритма Уоллеса и в линейной избыточной рекуррентной системе счисления 3-го порядка  $V_n = V_{n-1} + 3V_{n-3} + 2V_{n-4}$  с алфавитом  $\{0, 1\}$  и начальными значениями 1 1 1 2 4 8 с применением правил совмещенного во времени сложения в последней системе счисления.

### Изложение основного материала

Для конструирования теоретически самого быстрого многооперандного сумматора хорошо известен и используется алгоритм Уоллеса (дерева Уоллеса) [12]. Однако сложные межсоединения деревьев Уоллеса сильно затрудняют практическую реализацию деревьев Уоллеса [12]. Это алгоритм поочередного сложения многих операндов с применением сумматоров с сохранением переноса (CSA – carry save adder) и одного быстрого сумматора с распространением переноса (CPA – carry propagate adder).

Для увеличения быстродействия вычислительного устройства возможен переход к рекуррентным зависимостям в математическом выражении алгоритма [13], что позволяет уменьшить число операций, заменить сложные операции более простыми и

т.п. Рекуррентные зависимости в математическом выражении алгоритма дает линейная избыточная рекуррентная система счисления (ЛИРССЧ) третьего порядка [9], образованная на основе рекуррентно-соотношения вида

$$V_n = V_{n-1} + 3V_{n-3} + 2V_{n-4} \quad (1)$$

с алфавитом  $\{0, 1\}$  и начальными значениями (НЗ) 1111248, то есть  $V_0 = 1, V_1 = 1, V_2 = 1, V_3 = 1, V_4 = 2,$

$V_5 = 4, V_6 = 8$ . Эта ЛИРССЧ обладает простыми правилами совмещенного сложения (ССЛ) целых положительных чисел [11].

Вследствие простоты этих правил целесообразно применить свойства ЛИРССЧ (1) для ССЛ 50-ти целых положительных чисел. В табл. 1 представлена система правил совмещенного во времени сложения 50-ти целых положительных чисел.

Таблица 1

Система правил совмещенного во времени сложения 50-ти целых положительных чисел

№ правила	Правило	№ правила	Правило	№ правила	Правило
1.	$0=0+0$	18.	$17V_n = V_n + V_{n+4}$	35.	$34V_n = V_{n+1} + V_{n+5}$
2.	$V_n = V_n + 0$	19.	$18V_n = V_{n+1} + V_{n+4}$	36.	$35V_n = V_n + V_{n+1} + V_{n+5}$
3.	$2V_n = V_{n+1}$	20.	$19V_n = V_n + V_{n+1} + V_{n+4}$	37.	$36V_n = V_{n+2} + V_{n+5}$
4.	$3V_n = V_n + V_{n+1}$	21.	$20V_n = V_{n+2} + V_{n+4}$	38.	$37V_n = V_n + V_{n+2} + V_{n+5}$
5.	$4V_n = V_{n+2}$	22.	$21V_n = V_n + V_{n+2} + V_{n+4}$	39.	$38V_n = V_{n+1} + V_{n+2} + V_{n+5}$
6.	$5V_n = V_n + V_{n+2}$	23.	$22V_n = V_{n+1} + V_{n+2} + V_{n+4}$	40.	$39V_n = V_n + V_{n+1} + V_{n+2} + V_{n+5}$
7.	$6V_n = V_{n+1} + V_{n+2}$	24.	$23V_n = V_n + V_{n+1} + V_{n+2} + V_{n+4}$	41.	$40V_n = V_{n+3} + V_{n+5}$
8.	$7V_n = V_n + V_{n+1} + V_{n+2}$	25.	$24V_n = V_{n+3} + V_{n+4}$	42.	$41V_n = V_n + V_{n+3} + V_{n+5}$
9.	$8V_n = V_{n+3}$	26.	$25V_n = V_n + V_{n+3} + V_{n+4}$	43.	$42V_n = V_{n+1} + V_{n+3} + V_{n+5}$
10.	$9V_n = V_n + V_{n+3}$	27.	$26V_n = V_{n+1} + V_{n+3} + V_{n+4}$	44.	$43V_n = V_n + V_{n+1} + V_{n+3} + V_{n+5}$
11.	$10V_n = V_{n+1} + V_{n+3}$	28.	$27V_n = V_n + V_{n+1} + V_{n+3} + V_{n+4}$	45.	$44V_n = V_{n+2} + V_{n+3} + V_{n+5}$
12.	$11V_n = V_n + V_{n+1} + V_{n+3}$	29.	$28V_n = V_{n+2} + V_{n+3} + V_{n+4}$	46.	$45V_n = V_n + V_{n+2} + V_{n+3} + V_{n+5}$
13.	$12V_n = V_{n+2} + V_{n+3}$	30.	$29V_n = V_n + V_{n+2} + V_{n+3} + V_{n+4}$	47.	$46V_n = V_{n+1} + V_{n+2} + V_{n+3} + V_{n+5}$
14.	$13V_n = V_n + V_{n+2} + V_{n+3}$	31.	$30V_n = V_{n+1} + V_{n+2} + V_{n+3} + V_{n+4}$	48.	$47V_n = V_n + V_{n+1} + V_{n+2} + V_{n+3} + V_{n+5}$
15.	$14V_n = V_{n+1} + V_{n+2} + V_{n+3}$	32.	$31V_n = V_n + V_{n+1} + V_{n+2} + V_{n+3} + V_{n+4}$	49.	$48V_n = V_{n+4} + V_{n+5}$
16.	$15V_n = V_n + V_{n+1} + V_{n+2} + V_{n+3}$	33.	$32V_n = V_{n+5}$	50.	$49V_n = V_n + V_{n+4} + V_{n+5}$
17.	$16V_n = V_{n+4}$	34.	$33V_n = V_n + V_{n+5}$	51.	$50V_n = V_{n+1} + V_{n+4} + V_{n+5}$

В предыдущем исследовании [11], было доказано 21 правило ССЛ. Докажем правила совмещенного сложения до 50-ти слагаемых включительно в ЛИРССЧ (1) и НЗ, то есть последние 30 равенств системы (табл. 1).

**Доказательство.** Двадцать второе правило  $21V_n = V_n + V_{n+2} + V_{n+4}$  в системе правил (табл. 1) получается из 21-го правила  $20V_n = V_{n+2} + V_{n+4}$  так:  $21V_n = V_n + 20V_n = V_n + V_{n+2} + V_{n+4}$ . Аналогично получается правило 24 из 23-го, 26 из 25-го, 28 из 27-го, 30 из 29-го, 32 из 31-го, 34 из 33-го, 36 из 35-го, 38 из 37-го, 40 из 39-го, 42 из 41-го, 44 из 43-го, 46 из 45-го, 48 из 47-го и 50 из 49-го.

Для доказательства остальных правил после 21-го используется правило 3 (табл. 1):

$$2V_n = V_{n+1} \text{ и его модификации } 2V_{n+1} = V_{n+2},$$

$$2V_{n+2} = V_{n+3}, 2V_{n+3} = V_{n+4}, 2V_{n+4} = V_{n+5}.$$

Например, правило 33 получается из 31-го так:

$$32V_n = V_n + 31V_n = V_n + V_n + V_{n+1} + V_{n+2} + V_{n+3} + V_{n+4}.$$

По правилу 3 имеем  $2V_n = V_{n+1}, 2V_{n+1} = V_{n+2}, 2V_{n+2} = V_{n+3}, 2V_{n+3} = V_{n+4}, 2V_{n+4} = V_{n+5}$ . Последовательное применение этих равенств дает:

$$32V_n = 2V_n + V_{n+1} + V_{n+2} + V_{n+3} + V_{n+4} = 2V_{n+1} + V_{n+2} + V_{n+3} + V_{n+4} = 2V_{n+2} + V_{n+3} + V_{n+4} = 2V_{n+3} + V_{n+4} = 2V_{n+4} = V_{n+5}.$$

$$\text{То есть } 32V_n = V_{n+5}.$$

Для сравнения быстродействия совмещенного во времени сложения и последовательного сложения создадим дерево Уоллеса поочередного сложения 50-ти слагаемых по аналогии с рис. 2 из [14] и рис. 1 из [11]. Итак, на рис. 1 изображена схема поочеред-

ного сложения 50-ти слагаемых с использованием дерева Уоллеса. При этом используется 48 цифровых компрессоров типа 3:2 и один быстрый сумматор с распространением переноса (CPA - carry propagate adder) [14]. Как известно, в общем случае для сложения  $n$  операндов по алгоритму Уоллеса

необходимо использовать  $n-2$  цифровых компрессоров и один CPA – сумматор [15].

Исходя из доказанных правил ССЛ в ЛИРССЧ (1) разработана функциональная модель устройства совмещенного во времени сложения 50-ти слагаемых, представленная на рис.2.

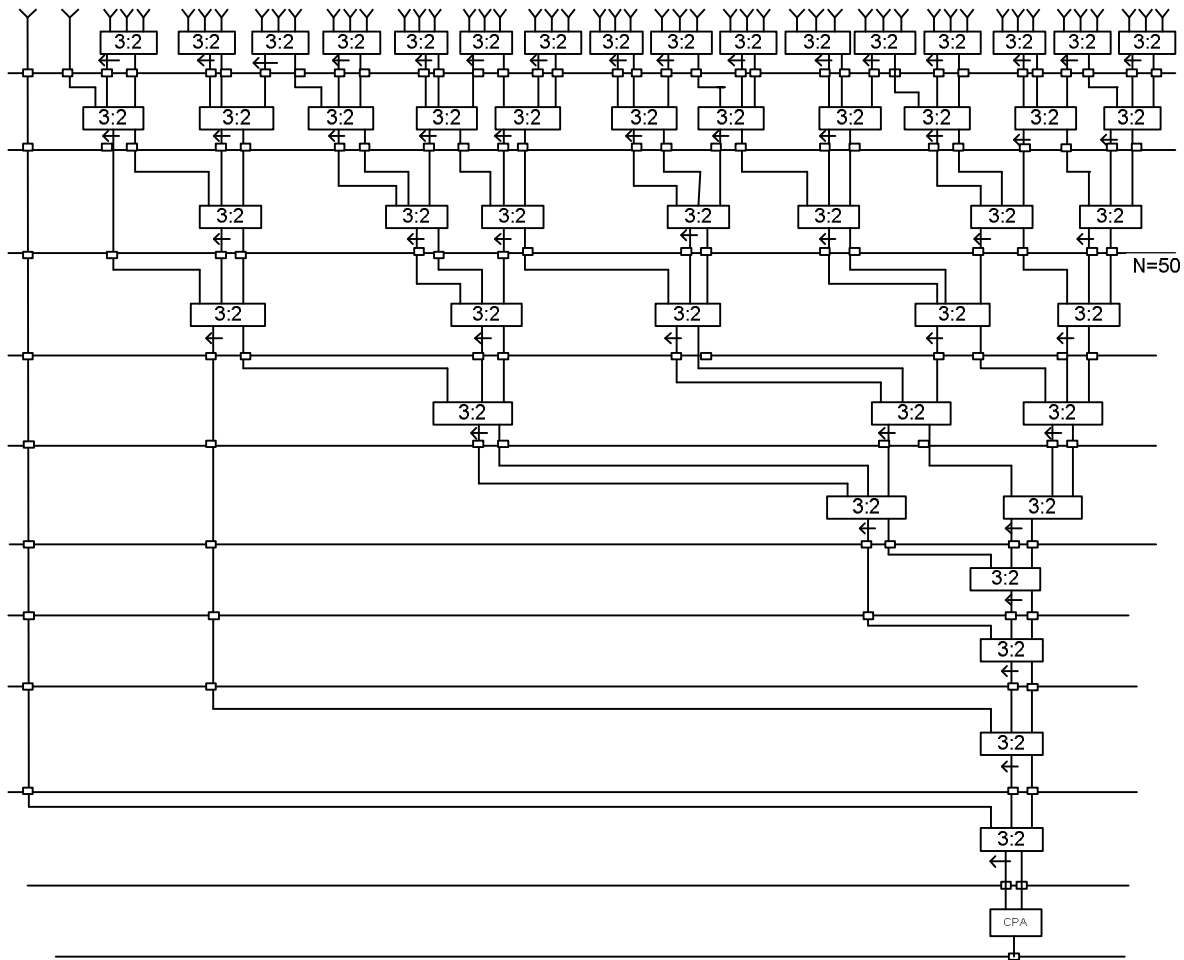


Рис.1. Поочередное сложение 50-ти слагаемых с использованием дерева Уоллеса

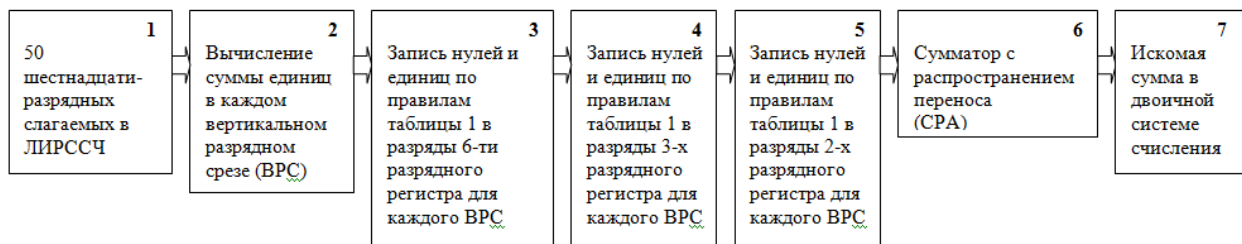


Рис. 2. Функциональная модель устройства многооперандного ССЛ 50-ти слагаемых

Надо отметить, что совмещенное во времени сложение до 50-ти целых положительных  $n$ -разрядных слагаемых включительно в ЛИРССЧ 3-го порядка с алфавитом  $\{0, 1\}$ , образованной ЛИРССЧ (1) с НЗ и по правилам из табл. 1, увеличивает разрядность суммы до  $n + 6$  разрядов.

Важным моментом является вычисление суммы единиц в каждом вертикальном разрядном срезе (ВРС), где не разделяется код на код условной сум-

мы и код переносов, не используется деление на 2 с упорядочением единиц в ВРС для получения остатка и частного от деления. Поэтому на этом этапе многооперандного сложения нет необходимости в использовании сумматоров с сохранением переносов, что уменьшает аппаратные затраты и увеличивает быстродействие устройства.

Для вычисления суммы единиц в каждом ВРС возможно использовать устройство для определения

количества единиц в двоичном числе [16], скорректированное для 50-ти целых положительных 16-ти разрядных чисел. Это устройство имеет на входе числа в двоичном коде и двоичный код количества единиц на выходе. При этом входной двоичный код преобразуется в уплотненный код, формируется итоговый уплотненный код и преобразуется уплотненный код в двоичный код количества единиц. Вычисление суммы единиц в каждом ВРС возможно производить независимо от других ВРС. Применение устройства для определения количества единиц в двоичном числе [16] соответствует блоку 2 на рис.2.

Система правил из таблицы 1 применяется многократно при промежуточных вычислениях. Так, для вычисления суммы 50 целых положительных слагаемых сначала необходимо 6 рядов разрядов под промежуточные значения (блок 3 на рис.2), получаемые с помощью правил 1-51 таблицы 1 (по одному разряду суммы единиц ВРС на один ряд промежуточных значений). Далее эти 6 рядов разрядов рассматриваются как 6 новых чисел увеличенной разрядности, к новым суммам ВРС которых применяются первые 7 правил таблицы 1. Для записи этих полученных сумм ВРС требуются уже 3 новых ряда разрядов под промежуточные значения (блок 4 на рис.2). Далее эти 3 ряда разрядов рассматриваются как 3 новых числа увеличенной разрядности, к очередным суммам ВРС которых применяются первые 4 правила таблицы 1. Для записи этих полученных сумм ВРС требуются уже 2 новых ряда разрядов под промежуточные значения (блок 5 на рис.2). Тем самым сложение 50 целых положительных слагаемых сводится к сложению двух рядов двоичных кодов и получению результирующей двоичной суммы с помощью сумматора с распространением переноса (CRA-adder) (блок 6 на рис.2).

Для сравнения быстродействия совмещенного сложения 50-ти целых положительных 16-разрядных чисел в ЛИРСЧ 3-го порядка (1), с НЗ 1 1 1 1 2 4 8, алфавитом {0, 1} с быстродействием поочередного сложения 50-ти целых положительных 16-разрядных двоичных чисел по алгоритму Уоллеса была разработана программная модель с помощью электронных таблиц Excel 2003 и встроенного языка программирования VBA. Программная модель подобна к описанной в [11], но алгоритм усовершенствован и применен для 50-ти целых положительных 16-разрядных чисел в ЛИРСЧ 3-го порядка (1). Работа программной модели показывает, что сложение в ЛИРСЧ для 16-разрядных целых положительных чисел требует меньше вычислений и выполняется в 8,6 раза быстрее, чем сложение в обычной двоичной ССЧ по алгоритму Уоллеса (определено как среднее значение времени вычисления суммы на 400 наборах по 50 случайных 16-разрядных слагаемых в каждом).

В программной модели совмещенного сложения в ЛИРСЧ 3-го порядка 50-ти целых положительных слагаемых для случая 16-разрядных слагаемых необходимо в 7,3 раза меньше ячеек памяти для хранения промежуточных результатов вычислений, чем в соответствующей программной модели поочередного сложения в обычной двоичной ССЧ по алгоритму Уоллеса.

Итак, применение правил совмещенного во времени сложения позволяет сократить количество промежуточных значений по сравнению с алгоритмом Уоллеса и уменьшить время подсчета результирующей суммы.

## Выводы

В результате проведенных исследований определено, что модель совмещенного во времени сложения 50-ти слагаемых в ЛИРСЧ 3-го порядка с алфавитом {0, 1}, сгенерированной рекуррентным соотношением  $V_n = V_{n-1} + 3V_{n-3} + 2V_{n-4}$  с НЗ 1 1 1 1 2 4 8 для 16-разрядных целых положительных чисел, требует меньше вычислений и в 8,6 раза быстрее, чем модель сложения в обычной двоичной ССЧ по алгоритму Уоллеса.

В программной модели совмещенного сложения в ЛИРСЧ 3-го порядка 50-ти целых положительных слагаемых для 16-разрядных слагаемых необходимо в 7,3 раза меньше ячеек памяти для хранения промежуточных результатов вычислений, чем в соответствующих программных моделях поочередного сложения в обычной двоичной ССЧ по алгоритму Уоллеса.

Данные исследования показывают, что свойства избыточной линейной рекуррентной системы счисления третьего порядка позволяют осуществлять многооперандное совмещенное во времени сложение 50-ти слагаемых, а также можно предположить (учитывая предыдущие исследования), что с увеличением числа слагаемых быстродействие многооперандного сумматора на основе правил сложения в ЛИРСЧ 3-го порядка будет возрастать по сравнению со сложением по алгоритму Уоллеса. Выгодным с аппаратной точки зрения является то, что результат ССЛ 50-ти слагаемых представляется в двоичной системе счисления без дополнительных преобразований. На практике эта ЛИРСЧ 3-го порядка может быть использована для ускорения работы цифровых фильтров.

## Список литературы

1. Брюхович Е.И. Экономическая стратегия разработки вычислительных систем: место и роль счислений / Е.И. Брюхович // *Управляющие системы и машины. Научно-производственный журнал. Ин-т кибернетики имени В. М. Глушкова АН УССР.* – 1990. – № 2 (106). – С. 3-18.
2. Пантелеева Н.М. Теоретичні основи створення природно-надійних комп'ютерних систем / Н.М. Пантелеева

ва, В.М. Рудницький. – Черкаси: Брама-Україна, 2009. – 200 с.

3. W. H. Specker. A Class of Algorithms for  $\ln x$ ,  $\exp x$ ,  $\sin x$ ,  $\cos x$ ,  $\tan^{-1} x$ ,  $\cot^{-1} x$ . – IEEE Trans. On Electronic Computers, Vol. EC-14, No1, pp.85-86, Feb 1965.

4. Chi-Hsiang Yeh, Benrooz Parhami. Efficient pipelined multi-operand adders with high throughput and low latency: designs and applications // Proc. 30th Asilomar Conf. Signals, Systems, and Computers, Pacific Grove, CA, 3-6 November 1996. – P. 894-898.

5. Мартинюк Т.Б. Рекурсивні алгоритми багатоперандної обробки інформації: [Монографія] / Т.Б. Мартинюк. – Вінниця: “Універсум-Вінниця”, 2000. – 216 с. – ISBN 966 – 7199 – 98 - 3.

6. Wallace C.S. A suggestion for a fast multiplier / C.S. Wallace // IEEE Transactions on Electronic Computers, C-13(2), February 1964. – P. 14-17.

7. Лебедев С.А. Электронно-вычислительные машины / С.А. Лебедев // Сессия АН СССР по научным проблемам автоматизации производства. Пленарные заседания. – М.: АН СССР. – 1957. – Т. 1. – С. 162 – 180.

8. Рудницький В.М. Метод підвищення швидкодії арифметичних пристроїв за рахунок суміщеного виконання операцій в структурно-блокових кодах / В.М. Рудницький, І.М. Федотова-Півень // Системи обробки інформації. Збірник наукових праць. – X. : ХУПС, 2009. – Вип. 4 (78). – С. 117-119.

9. Рудницький В.М. Моделювання суміщеного додавання до п'яти доданків включно в надлишковій рекурентній системі числення 3-го порядку / В.М. Рудницький, І.М. Федотова-Півень // Збірник наукових праць “Системи управління, навігації та зв'язку”. – К. : ДП „Центральний науково-дослідний інститут навігації та управління”. – 2011. – Вип. 2 (18). – С. 164 – 166.

10. Рудницький В.М. Програмна модель одночасного додавання п'яти додатних цілих чисел в надлишковій рекурентній системі числення третього порядку / В.М. Рудницький, І.М. Федотова-Півень // Наука і техніка Повітряних Сил Збройних Сил України. – 2011. – № 2(6). – С. 158-161.

11. Федотова-Півень І.Н. Програмное моделирование совмещенного во времени сложения двадцати целых положительных чисел в избыточной рекуррентной системе счисления третьего порядка / И.Н. Федотова-Півень, О.Б. Півень // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, 2013. – № 58. – С. 131-137.

12. Mou Z.-J. A class of close-to-optimum adder trees allowing regular and compact layout / Z.-J. Mou, F. Jutand // IEEE International Conference on Computer Design: VLSI in Computers and Processors. 1990. ICCP'90. Proceedings, pages: 251-254, 17-19 September, 1990, Cambridge, MA, ISBN 0-8186-2079-X.

13. Справочник по цифровой вычислительной технике (электронные вычислительные машины и системы) / [сост. Малиновский Б. Н., Александров В. Я., Боюн В. П. и др.; ред. Малиновский Б. Н.]. – К.: Техніка, 1980. – 320 с.

14. Martinez M. On the design of FPGA-based multi-operand pipeline adders / M. Martinez, J. Valls, E. Boemo // Proceedings of the XII Design of Circuits and Integrated Systems Conference (DCIS'97). – Universidad de Sevilla, Seville, Spain, November 18-21, 1997. – P. 701-706.

15. Balasubramanian P. Self-Timed Multi-Operand Addition / P. Balasubramanian, D.A. Edwards, W. B. Toms // International Journal of circuits, systems and signal processing. – Vol. 6, Issue 1, 2012. – P. 1-11.

16. Пат. RU 2446442 С1 Российская Федерация, МПК G06F 7/50, H03K 21/100. Устройство для определения количества единиц (нулей) в двоичном числе/ Ядыкин И.М.; заявитель: Ядыкин И.М.; патентообладатель: ФГБОУ высшего профессионального образования «Национальный исследовательский ядерный университет «МИФИ» (НИЯУ МИФИ)». - № 2011114163/08; заявл.: 11.04.2011; опублик.: 27.03.2012 Бюл.№9.

Поступила в редколлегию 4.12.2014

**Рецензент:** д-р техн. наук, проф. В.Н. Рудницький, Черкаський державний технологічний університет, Черкаси.

### СУМІЩЕНЕ В ЧАСІ ДОДАВАННЯ 50-ТИ ЦІЛИХ ПОЗИТИВНИХ ЧИСЕЛ В РЕКУРЕНТНІЙ СИСТЕМІ ЧИСЛЕННЯ

І.М. Федотова-Півень

В статті в рамках програмної моделі розглянуто порівняння швидкодії суміщеного в часі додавання 50-ти цілих додатних 16-розрядних чисел в лінійній надлишковій рекурентній системі числення третього порядку з алфавітом  $\{0, 1\}$ , утвореної рекурентним співвідношенням  $B_n = B_{n-1} + 3B_{n-3} + 2B_{n-4}$  з початковими значеннями 1 1 1 2 4 8 і почергового додавання 50-ти цілих додатних 16-розрядних двійкових чисел за стандартним алгоритмом Уолеса. З роботи програмної моделі слідує, що суміщене в часі додавання 50-ти 16-розрядних цілих додатних чисел у вказаній рекурентній системі числення виконується в 8,6 рази швидше і вимагає в 7,3 раз менше комірок пам'яті для зберігання проміжних результатів обчислень, ніж послідовне додавання за класичним алгоритмом Уолеса.

**Ключові слова:** надлишковість, структурно-блочні коди, рекурентні системи числення, суміщене в часі багатоперандне додавання, вертикальні розрядні зрізи.

### THE COINCIDENT IN TIME ADDITION OF FIFTY POSITIVE INTEGERS IN THE RECURRENT NUMERATION SYSTEM

I.M. Fedotova-Piven'

In an article in the framework of program model examined compare the performance of coincident in time addition of 50 positive 16-bit binary numbers in the linear redundant recurrent numeration system of the third order with alphabet  $\{0, 1\}$  created by the linear recurrence relation  $B_n = B_{n-1} + 3B_{n-3} + 2B_{n-4}$  with starting values 1 1 1 2 4 8 and addition by turns of 50 integer positive 16-bit binary numbers on a standard algorithm of Wallace. Work of program model shows that coincident in time addition of 50 16-bit positive 16-bit binary numbers in the specified recurrent system of numeration performed 8.6 times faster and requires 7.3 times less memory to store intermediate results of calculations than the sequential addition for the classical algorithm Wallace.

**Keywords:** redundancy, structurally - sectional codes, recurrent numeration systems, coincident in time multi-operand addition, vertical digital slices.