

Д.А. Сорокин, И.И. Левин

СУММАТОР С ПЛАВАЮЩЕЙ ЗАПЯТОЙ В ЦИФРОВЫХ ФОТОННЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВАХ

В рамках предлагаемой авторами концепции структурной организации вычислений в цифровых фотонных вычислительных устройствах необходимо использовать последовательную обработку информации, что позволяет минимизировать скважность подачи операндов из внешней памяти или других электронных источников на фотонное устройство. Это становится возможным, когда обработка операндов не превышает число тактов, равное их разрядности. Кроме того, при последовательной поразрядной обработке значительно снижаются аппаратные затраты на синхронизацию потоков данных. Устранение скважности и снижение накладных расходов на реализацию вычислительных структур в значительной степени способны повысить эффективность цифровых фотонных вычислительных устройств относительно электронных. Однако для создания фотонных вычислительных структур, ориентированных на решение различных трудоёмких задач из таких областей, как математическая физика, линейная алгебра, нейросетевая обработка и многих других, необходимы устройства, реализующие базовые арифметические функции в формате плавающей запятой. Большинство таких арифметических функций содержит элементарную операцию целочисленного сложения. При последовательной обработке операндов младшими разрядами вперёд в двоичной форме представления устройства целочисленного сложения не могут начать выдавать результат до тех пор, пока не будут обработаны все биты информации для учёта переноса, что увеличивает в два раза скважность подачи операндов и латентность устройства. Поэтому для устранения скважности и сокращения латентности предлагается использовать четверичную знакоразрядную форму представления чисел и подавать операнды старшими разрядами вперёд. Применение знакоразрядной формы представления чисел позволяет выполнять немедленную передачу старших разрядов результата операции для дальнейшей обработки в следующие устройства, не дожидаясь получения младших разрядов. В статье рассматриваются вопросы построения всех компонент знакоразрядного сумматора с плавающей запятой: блока определения разности порядков, блока денормализации мантиссы меньшего числа, сумматора мантисс, блока нормализации мантиссы результата и блока коррекции порядка результата. Приведены алгоритмы функционирования данных блоков. Оценка эффективности предлагаемого знакоразрядного сумматора выполнена на макете, разработанном в базе цифровой фотонной логики на реконфигурируемом компьютере «Терциус». Показано, что за счёт величины тактовой частоты работы цифровые фотонные вычислительные устройства способны обеспечить производительность почти на два десятичных порядка больше по сравнению с микроэлектронными устройствами.

Цифровое фотонное вычислительное устройство; сумматор с плавающей запятой; знакоразрядная система счисления.

D.A. Sorokin, I.I. Levin

FLOATING-POINT ADDER IN DIGITAL PHOTONIC COMPUTING SYSTEMS

Within the structural computation paradigm proposed by the authors, digital photonic computing systems are expected to employ sequential data processing, which allows for the minimization of operand duty cycle gaps when data is supplied from external memory or other electronic sources to the photonic device. This becomes feasible when the processing time per operand does not exceed the number of clock cycles corresponding to the operand's bit width. Moreover, sequential digit-wise processing significantly reduces hardware costs associated with dataflow synchronization. The elimination of duty cycle gaps and reduction in structural overhead can substantially enhance the efficiency of digital photonic computing systems relative to their electronic counterparts. However, to enable photonic computational architectures capable of solving complex and computation-intensive problems in domains such as mathematical physics, linear algebra, neural network processing, and others, it is necessary to implement core arithmetic functions in floating-point format. Most of these functions are built around elementary integer addition. In binary systems with sequential processing in least-significant-digit-first order, integer adders are unable to begin producing results until all bits have been processed and carry propagation is complete, thereby doubling the operand duty cycle and increasing latency. To address these issues, this paper proposes the

use of a quaternary signed-digit number representation with operands processed in most-significant-digit-first order. This representation enables immediate transmission of the most significant digits of the result to downstream processing units, without waiting for the completion of lower-order digit computation. This paper addresses the design of all components of the signed-digit floating-point adder: the exponent difference unit, the mantissa denormalization unit for the operand with the smaller exponent, the mantissa adder, the mantissa normalization unit for the result, and the exponent correction unit. Operational algorithms for these units are presented. The performance of the proposed signed-digit adder has been evaluated on a prototype implemented in a digital photonic logic framework on the reconfigurable "Terzius" computing platform. It is demonstrated that, due to the high clock frequency achievable by digital photonic computing devices, their performance can exceed that of microelectronic devices by nearly two orders of decimal scale.

Digital photonic computing; floating-point adder; signed-digit number representation.

Введение. Своевременное и качественное решение трудоемких задач различных областей науки и техники невозможно без развития эффективных высокопроизводительных вычислительных систем. Однако в течение последних двух десятилетий стало очевидным, что экспоненциальный рост тактовых частот и степени интеграции далее невозможен, и лишь эволюция транзисторов и технологий их изготовления позволит ещё некоторое время масштабировать производительность [1, 2]. При этом определяющую роль в построении высокопроизводительных систем начинает играть отвод тепла с кристалла [3]. Постоянный рост потребляемой мощности с сохранением геометрических размеров полупроводниковых устройств сводит на нет возможность постоянного увеличения производительности современных систем.

Поэтому над задачей создания альтернативной вычислительной техники в настоящее время работают многие учёные и исследователи. Одним из перспективных направлений в данной области является создание фотонных вычислителей, в основе которых лежат эффекты взаимодействия когерентных систем световых волн, порождаемых лазерным излучением [4]. Класс задач, которые смогут решать фотонные вычислители, сопоставим с задачами для электронных вычислительных машин, но при этом фотонные технологии потенциально способны в сотни раз уменьшить потребляемую энергию, необходимую для достижения одинаковой производительности с нынешними вычислительными системами.

В настоящее время ведутся разработки технологий построения элементной базы цифровых фотонных микросхем и организации эффективной обработки данных на их основе. В работах [5–7] авторами предложен вариант перспективной архитектуры для цифровых фотонных вычислительных устройств (ЦФВУ), поддерживающей структурную парадигму вычислений [8]. Для обеспечения равенства темпа обработки операндов темпу их поступления на вход ЦФВУ, вне зависимости от формата представления данных, предложены принципы построения подсистем синхронизации и коммутации, обеспечивающих статическое и динамическое согласование потоков операндов. Показано, что если предполагаемая тактовая частота ЦФВУ будет превосходить тактовую частоту электронной памяти на три порядка, то для наиболее эффективного использования аппаратного ресурса ЦФВУ при реализации вычислительных структур решаемых задач необходимо применять последовательный способ передачи информации.

Последовательная обработка позволяет минимизировать или даже полностью устранить скажность подачи данных, если число тактов обработки операндов в каждой операции вычислительной структуры не будет превышать число тактов их поступления. Однако в базовой функции многих арифметических операций – суммировании – традиционно применяемые алгоритмы обработки чисел в двоичном представлении определяют порядок распространения переноса от младших разрядов к старшим. Поэтому при последовательной обработке операндов младшими разрядами вперёд скажность поступления данных будет равна двум, и результирующая латентность операции в два раза больше разрядности операндов. Очевидно, что такой способ обработки данных при решении задач значительно снижает эффективность ЦФВУ.

В отличие от данного подхода, если для обработки последовательных кодов применить знакоразрядную форму представления чисел [9] и подавать операнды старшими разрядами вперёд, то на латентность арифметических операций суммирования, умножения, деления и других перестаёт влиять разрядность операндов [10]. Её величина становится зависимой только от числа тактов обработки разрядов плюс несколько дополнительных тактов синхронизации, поскольку в знакоразрядной системе счисления распространение переноса при выполнении операции сложения выполняется не далее соседнего разряда. Также скважность обработки данных становится минимальной, так как выполняется немедленная передача старших разрядов результата операции для дальнейшей обработки в следующие операционные блоки, не дожидаясь получения младших разрядов. Это особенно важно для минимизации накладных расходов на триггеры ЦФВУ при синхронизации функциональных узлов в вычислительных структурах.

В настоящей статье рассматриваются подходы к построению базовой арифметической операции ЦФВУ с использованием четверичной знакоразрядной системы счисления. Базовой операцией для реализации вычислительных структур большинства различных трудоёмких задач является операция сложения в формате плавающей запятой. Рассмотрены наиболее важные особенности реализации данной операции через призму эффективности вычислительных структур на её основе, синтезируемых в базисе цифровой фотонной логики.

Анализ знакоразрядного формата обработки операндов. Число в знакоразрядной системе счисления, как и в любой позиционной системе, можно описать формулой

$$Z = \sum_{i=-m}^n x_i \cdot 2R^{-i},$$

где $x_i = \{-R, -(R+1), \dots, -1, 0, 1, \dots, R\}$.

Однако, при таком подходе возникает избыточность представления по сравнению с двоичной системой счисления [11]. В табл. 1 указана избыточность для оснований $S = 2^k$, ($k = 1, 2, 3, 4, 5$), выраженная в относительной величине ΔS , %. Из представленных в таблице 1 данных видно, что самой избыточной является двоичная знакоразрядная система, четверичная имеет вполтину меньшую избыточность и т.д. При использовании оснований $S = 2^k - 3$ ($k = 4, 5$) избыточность минимизируется, однако основания $S = 2^k$ обеспечивают простоту совмещения знакоразрядной системы с двоичной системой счисления, поэтому на практике рационально использовать именно их.

Таблица 1

Избыточность знакоразрядной системы при различных основаниях

Основание $S(2^k)$	$2(2^1)$	$4(2^2)$	$8(2^3)$	$16(2^4)$	$32(2^5)$	$13(2^4-3)$	$29(2^5-3)$
ΔS , %	100	50	33	25	20	8	3

Стоит отметить, что при увеличении основания знакоразрядной системы счисления нелинейно растет сложность алгоритмов выполнения элементарных арифметико-логических операций, а их аппаратная реализация требует неприемлемо много ресурсов. Это в том числе связано с неоднозначностью представления чисел и сложностью обработки исключительных ситуаций. Исследования показали, что аппаратные затраты в ЦФВУ на реализацию автомата управления в операции целочисленного деления чисел, представленных в знакоразрядной системе счисления по основанию $S = 2^3$, настолько велики, что пропадает принципиальная целесообразность ее использования по сравнению с обычной двоичной системой счисления. Наиболее оптимальной для реализации является знакоразрядная система счисления с основанием $S = 2^2$ (четверичная знакоразрядная система). Выбор данного основания системы счисления является компромиссом между избыточностью представления знакоразрядных чисел и сложностью реализации элементарных арифметико-логических операций в базисе ЦФВУ, а также гарантирует сравнительную простоту совмещения знакоразрядной системы счисления с двоичной системой.

Для реализации последовательного поразрядного суммирования четверичных знакоразрядных кодов старшими разрядами вперёд [12, 13] необходимо разряды операндов $X = 0, x_1, \dots, x_i, \dots, x_n$ и $Y = 0, y_1, \dots, y_i, \dots, y_n$ представить в виде $x_i, y_i \in \{\overline{3}, \overline{2}, \overline{1}, 0, 1, 2, 3\}$. Сумма $Z = X + Y$ также является знакоразрядным числом: $Z = 0, z_1 z_2 \dots z_i \dots z_n$.

Разряды z_i вычисляются следующим образом. Чтобы избежать возникновения переноса из младших разрядов суммы в старшие, что не позволяет использовать старшие разряды в дальнейших вычислениях без получения всех младших разрядов, сначала вычисляется предварительная сумма разрядов операндов по формуле

$$S_i = x_{i+2} + y_{i+2} + 2S_{i-1} - 8z_{i+1},$$

где $i = 0, 1, 2, \dots, n$.

Затем формируется перенос по формуле

$$p_i = \begin{cases} 1, & \text{если } S_i \geq 3 \\ 0, & \text{если } |S_i| < 3 \\ \overline{1}, & \text{если } S_i \leq -3. \end{cases}$$

Далее вычисляется конечный результат суммы по формуле

$$z_i = S_i - 4p_i + p_{i+1}.$$

В базе ЦФВУ рациональной является реализация последовательных элементарных арифметико-логических операций, обрабатывающие данные по битам в темпе поступления информации. При подаче операндов старшими разрядами вперёд в знакоразрядной системе счисления появляется возможность обработки старших бит информации, не дожидаясь прихода младших разрядов, что в свою очередь значительно уменьшает латентность операций и позволяет снизить накладные расходы на синхронизацию потоков операндов в ЦФВУ. Кроме того, минимизируется скважность поступления данных и сохраняется точность результатов вычислений не ниже точности вычислений в двоичной системе счисления. Это особенно критично при выполнении таких операций, как суммирование и умножение чисел в формате с плавающей запятой, а также при формировании на их основе в ЦФВУ более сложных вычислительных структур решения различных трудоёмких задач.

Последовательный знакоразрядный сумматор в формате с плавающей запятой. Операцию сложения чисел в четверичной знакоразрядной системе счисления [14] можно описать формулой

$$MZ \times 4^{lZ} = MX \times 4^{lX} \pm MY \times 4^{lY} = (\hat{M} + \tilde{M} \times 4^{-l\delta l}) \times 4^l,$$

где MX, MY, MZ – мантиссы чисел X, Y, Z соответственно; lX, lY, lZ – порядки чисел X, Y, Z ; \hat{M} – мантисса большего числа; \tilde{M} – мантисса меньшего числа; l – порядок результата; δl – разность порядков;

Структурная схема сумматора, реализуемого в базе ЦФВУ приведена на рис. 1.

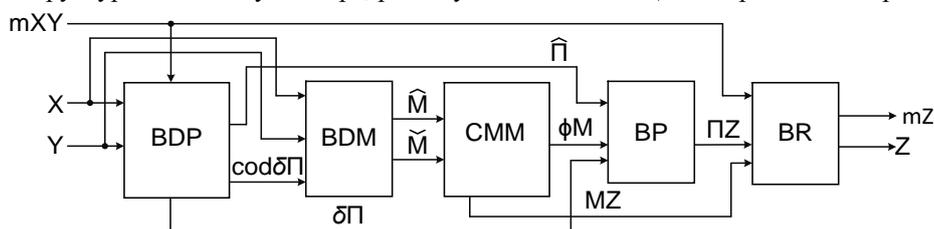


Рис. 1. Структурная схема последовательного знакоразрядного сумматора ЦФВУ в формате с плавающей запятой

Сумматор содержит: блок вычисления разности порядков BDP, блок денормализации меньшей мантиссы BDM, сумматор мантисс CMM, блок коррекции порядка результата BP, блок формирования результата BR.

Вычисление разности порядков $\delta\Pi = \Pi X - \Pi Y$ выполняется в соответствии с описанным ранее знакоразрядным алгоритмом суммирования.

Структурная схема блока BDP показана на рис. 2.

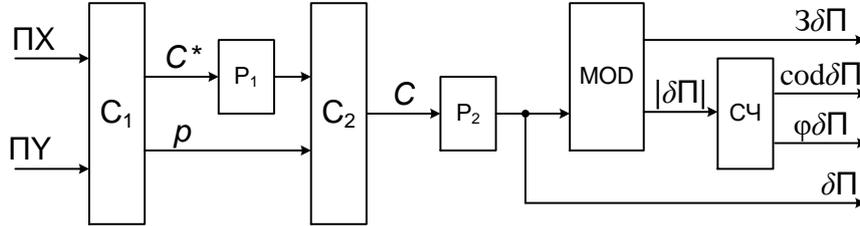


Рис. 2. Структурная схема BDP

На сумматоре C_1 вычисляются промежуточный результат $C^* = \{C_2^*, C_1^*, C_0^*\}$ и перенос $p = \{p^+, p^-\}$. Алгоритм суммирования C_1 имеет следующий вид:

- 1) $C_0^* = a_0 \cdot \overline{b_0} \vee \overline{a_0} \cdot b_0; \alpha = \overline{a_0} \cdot b_0;$
- 2) $C_1^* = (a_1 \cdot \overline{b_1} \vee \overline{a_1} \cdot b_1) \cdot \overline{\alpha} \vee (a_1 \cdot \overline{b_1} \vee \overline{a_1} \cdot b_1) \cdot \alpha;$
- 3) $\beta = \overline{a_1} \cdot b_1 \vee (\overline{a_1} \vee b_1) \cdot \alpha;$
- 4) $\gamma = (a_2 \cdot \overline{b_2} \vee \overline{a_2} \cdot b_2) \cdot \overline{\beta} \vee (a_2 \cdot \overline{b_2} \vee \overline{a_2} \cdot b_2) \cdot \beta;$
- 5) $\eta = a_2 \overline{b_2} \vee (a_2 \vee \overline{b_2}) \cdot \beta;$
- 6) $C_2^* = (\eta \vee C_0^*) \cdot C_1^*;$
- 7) $p^+ = \eta^-(\gamma \vee C_0^* \cdot C_1^*);$
- 8) $p^- = \eta(\gamma \vee C_1^*);$
- 9) $p = p^+ \vee p^-.$

Здесь и далее все приведённые алгоритмы функционирования блоков описаны в базисе примитивных операций ЦФВУ, таких как инверсия « $\overline{}$ », логическое И « \wedge », логическое ИЛИ « \vee ».

Полученное значение C^* задерживается на такт в регистре P_1 поступает на сумматор C_2 , где складывается с переносом p , и получается значение разности порядков $C = \{C_2, C_1, C_0\}$. Алгоритм суммирования C_2 имеет вид:

- 1) $C_0 = C_0^* \cdot \overline{p} \vee C_0^* \cdot p;$
- 2) $C_1 = C_1^* \cdot \overline{p} \vee (C_1^* \cdot \overline{C_0^*} \vee \overline{C_1^*} \cdot C_0^*) p^+ \vee C_1^* \cdot \overline{C_0^*} \vee \overline{C_1^*} \cdot C_0^* \cdot p^-;$
- 3) $C_2 = (\overline{C_2^*} \vee p) \cdot (\overline{C_2^*} \vee C_0^*) \cdot p^- \vee (C_1^* \cdot C_0^* \vee \overline{C_1^*} \cdot \overline{C_0^*}).$

Далее на регистре P_2 полученное значение C задерживается на такт и поступает на блок MOD для вычисления модуля разности порядка $|\delta\Pi| = \{|\delta\Pi|_2, |\delta\Pi|_1, |\delta\Pi|_0\}$ и знака разности порядков $3\delta\Pi$. MOD работает по следующему алгоритму:

- 1) Вычисление знака разности порядка
 - 1.1) $3\delta\Pi^- = \overline{3\delta\Pi^-} \cdot \overline{C_2} (C_1 \vee C_0) \vee 3\delta\Pi^+ \cdot \overline{y_0};$
 - 1.2) $3\delta\Pi^+ = \overline{3\delta\Pi^+} \cdot C_2 \vee 3\delta\Pi^- \cdot \overline{y_0};$
- Вычисление модуля разности порядка
 - 2.1) $|\delta\Pi|_0 = C_0;$
 - 2.2) $|\delta\Pi|_1 = \overline{C_1} \cdot C_0 \cdot (C_2 \cdot \overline{3\delta\Pi^+} \vee 3\delta\Pi^-) \vee C_1 \cdot (C_0 \cdot (C_2 \cdot \overline{3\delta\Pi^+} \vee 3\delta\Pi^-));$
 - 2.3) $|\delta\Pi|_2 = C_2 \cdot 3\delta\Pi^+ \vee \overline{C_2} \cdot 3\delta\Pi^- \cdot (C_1 \vee C_0).$

На счётчике СЧ полученное значение преобразуется в значение $cod\delta\Pi$, позволяющий в блоке BDM выполнять сдвиг мантиисы на число разрядов, соответствующее $\delta\Pi$. Работу СЧ можно описать следующими отношениями: $|C^i| = 4 * |C^{i-1}| + |C_i|$, $\phi\delta\Pi = (|\delta\Pi| \geq 16)$, где C^i – текущее значение счётчика; C^{i-1} – предыдущее значение счётчика, $\phi\delta\Pi$ – переполнение разности порядков.

Алгоритм работы схемы формирования $cod\delta\Pi$ имеет вид:

- 1) $cod\delta\Pi_2 = C_2 \cdot (\overline{C_2} \vee 3\delta\Pi^+) \cdot \overline{3\delta\Pi^-}$;
- 2) $cod\delta\Pi_1 = C_1 \cdot (\overline{C_2} \vee 3\delta\Pi^+) \cdot \overline{3\delta\Pi^-}$;
- 3) $cod\delta\Pi_0 = C_0 \cdot (\overline{C_2} \vee 3\delta\Pi^-) \cdot 3\delta\Pi^+$.

Блок BDM предназначен для выбора мантиссы большего числа \hat{M} и сдвига мантиссы меньшего числа \check{M} на $cod|\delta\Pi|$ вправо.

Структурная схема BDM изображена на рис. 3.

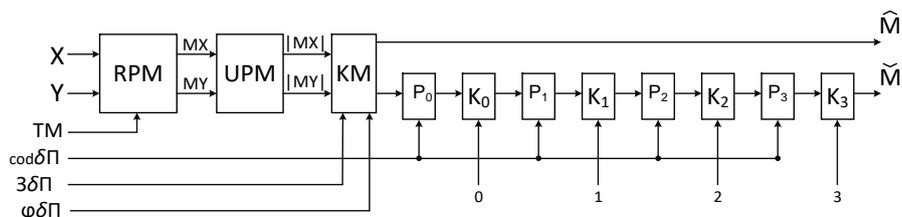


Рис. 3. Структурная схема блока BDM

В блоке RPM из поступающих операндов X и Y выделяются мантиссы (MX , MY) и задерживаются на два такта, что соответствует времени вычисления $3\delta\Pi$ в блоке BDP. Функционально блок RPM состоит из входного и выходного регистров, построенных на D-триггерах и осуществляющих задержку операндов на один такт в каждом, и схемы подготовки мантис, в которой из операндов стробом TM выделяются мантиссы. Строб TM формируется из строба mXY входных операндов X и Y , задержанного на число тактов, равное длине порядка $ПХ(ПУ)$.

В блоке UPM формируются модули мантиссы входных операндов X и Y . Алгоритм формирования модуля мантиссы $|M| = \{M_0, M_1, M_2\}$ описывается следующим алгоритмом:

- 1) $3M^- = 3M^- \cdot Y^0 \vee \overline{3M^+} \cdot MY_2$;
- 2) $3M^+ = 3M^+ \cdot Y^0 \vee \overline{3M^-} \cdot MY_2$;
- 3) $M_2 = 3M^+ \vee \overline{Mi_2} \vee 3M^- \cdot \overline{MY_2} (Mi_1 \vee Mi_0)$;
- 4) $M_1 = (3M^+ \vee \overline{Mi_2} \cdot \overline{3M^-}) Mi_1 \vee (3M^+ \vee \overline{3M^-} \cdot Mi_2) (Mi_1 \neq Mi_0)$;
- 5) $M_0 = Mi_0$,

где $3M^+$, $3M^-$ – соответственно положительный и отрицательный знаки мантиссы, определяемые знаком первого разряда мантиссы; $\{Mi_0, Mi_1, Mi_2\}$ – входные значения разрядов мантиссы.

Далее модули мантиссы операндов $|MX|$, $|MY|$, $3\delta\Pi$ и $\phi\delta\Pi$ поступают в блок KM, где выбираются мантиссы большего и меньшего чисел, чтобы затем на регистрах P_0, P_1, P_2, P_3 и коммутаторах K_0, K_1, K_2, K_3 осуществить сдвиг меньшей мантиссы в соответствии с $cod\delta\Pi$.

В блоке KM определение мантиссы большего и меньшего операнда осуществляется по $3\delta\Pi$ и $\phi\delta\Pi$ в соответствии с алгоритмом:

- 1) $\hat{M} = MX \cdot (3\delta\Pi^+ \vee \overline{3\delta\Pi^-}) \vee MY \cdot \overline{3\delta\Pi^-}$;
- 2) $\check{M} = MX \cdot (\overline{\phi\delta\Pi} \cdot \overline{3\delta\Pi^-}) \vee MY \cdot (3\delta\Pi^- \cdot \overline{3\delta\Pi^+} \vee \overline{3\delta\Pi^+} \cdot \overline{\phi\delta\Pi})$,

где \hat{M} , \check{M} – мантиссы большего и меньшего числа; MX , MY – мантиссы X и Y .

Принцип работы регистров P_0, P_1, P_2, P_3 и коммутаторов K_0, K_1, K_2, K_3 заключается в следующем: четыре последовательных регистра, собранных на D-триггерах, имеют разные задержки, соответствующие весам разрядов: четырехразрядного двоичного числа $cod\delta\Pi$ и равные 1, 2, 4, 8 тактов. Подавая разряды $cod\delta\Pi$ на коммутаторы соответствующих регистров, осуществляется задержка на каждом регистре, и в сумме общая задержка на всех регистрах будет составлять $|\delta\Pi|$. Общий алгоритм функционирования коммутатора приведен ниже:

- 1) $K_0: DP_1 = P_0 \cdot cod\delta\Pi_0 \vee \check{M} \cdot \overline{cod\delta\Pi_0}$;

$$2) K_j: DP_{j+1} = P_j \cdot \text{cod}\delta\Pi_j \vee DP_j \cdot \overline{\text{cod}\delta\Pi_j}, \quad i = 1, 2, 3,$$

где DP_{j+1} – значения на выходе j -го коммутатора;

DP_1 – значение на выходе 1-го коммутатора;

P_j – значение на выходе j -го регистра.

Блок СММ (рис. 4) предназначен для суммирования мантисс, поступающих с блока выравнивания порядков, и описывается следующей формулой:

$$MZ = \hat{M} + \check{M},$$

где MZ – результирующая мантисса;

\hat{M} – мантисса большего числа;

\check{M} – мантисса меньшего числа, приведённая к порядку большего.

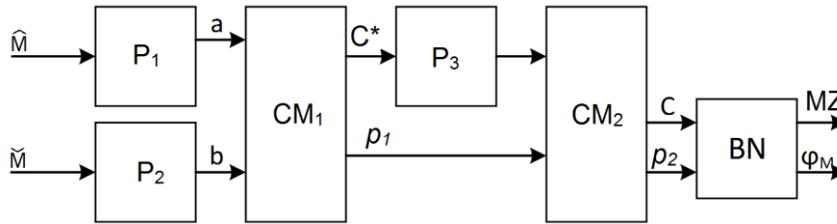


Рис. 4. Структурная схема блока СММ

Мантиссы большего и меньшего операндов на регистрах P_1 и P_2 задерживаются на один такт, а затем поступают на сумматор CM_1 . Результат C^* сумматора CM_1 задерживается на один такт на регистре P_3 , а затем на сумматоре CM_2 складывается с переносом p_1 . Полученный результат C затем нормализуется на блоке нормализации BN переносом p_2 .

Алгоритм сумматора CM_1 подобен алгоритму сумматора C_1 блока BDP , только здесь вместо вычитания выполняется суммирование:

- 1) $C_0^* = (\bar{a}_0 \cdot b_0 \vee a_0 \cdot \bar{b}_0)$; $\alpha = a_0 \cdot b_0$;
 $C_1^* = ((\bar{a}_1 \cdot b_1 \vee a_1 \cdot \bar{b}_1) \cdot \bar{\alpha}) \vee ((\bar{a}_1 \cdot b_1 \vee a_1 \cdot \bar{b}_1) \cdot \alpha)$; $\beta = a_1 \cdot b_1 \vee (\bar{a}_1 \cdot b_1 \vee a_1 \cdot \bar{b}_1) \cdot \alpha$;
- 2) $C_2^* = C_1^* \cdot (C_0^* \vee \bar{\beta} \cdot (a_2 \vee b_2 \vee a_2 \cdot b_2))$;
- 3) $p_1^+ = \bar{a}_2 \cdot \bar{b}_2 \cdot (\beta \vee C_1^* \cdot C_0^*)$;
- 4) $p_1^- = (a_1 \cdot b_1 \cdot \alpha)(a_2 \cdot b_2) + (\bar{a}_1 \cdot \bar{b}_1 \cdot \bar{\alpha})(a_2 \vee b_2)$;
- 5) $p = p^+ \vee p^-$.

Сумматор CM_2 полностью соответствует сумматору C_2 в блоке BDP .

Блок нормализации BN предназначен для нормализации результата на один разряд при положительном и отрицательном переполнении мантиссы. Алгоритм нормализации записывается следующим образом:

- 1) $D\varphi^+ = p_2$;
- 2) $D\varphi^- = (C^i = 0) \cdot \overline{D\varphi^+}$;
- 3) $MZ^i = C^i \cdot D\varphi^- \vee C^{i-1} (\overline{D\varphi^+ \vee D\varphi^-}) \vee C^{i-2} \cdot D\varphi^+$,

где C^i – i -й разряд мантиссы результата C ;

$D\varphi^+$ – признак положительного переполнения мантиссы;

$D\varphi^-$ – признак отрицательного переполнения мантиссы.

Формирование порядка результата в блоке BP выполняется по формуле

$$\Pi Z = \hat{\Pi} + \delta\Pi + \varphi_M.$$

Пока $\delta\Pi_i = 0$, i -му разряду порядка результата ΠZ_i присваивается значение соответствующего разряда порядка большего числа $\hat{\Pi}$. Если $\delta\Pi_i < 0$, то $\Pi Z_i = \hat{\Pi}_i$, а для $i+1$ разряда проверяется условие $(\exists \delta\Pi^- \vee \exists \delta\Pi^+) = 1$ и $\Pi Z_{i+1} = \hat{\Pi}_{i+1}$. Во всех остальных случаях $\Pi Z_i = \hat{\Pi}_i + \delta\Pi_i$. Затем выполняется коррекция порядка разрядом переполнения мантиссы φ_M .

Структурная схема блока ВР представлена на рис. 5.

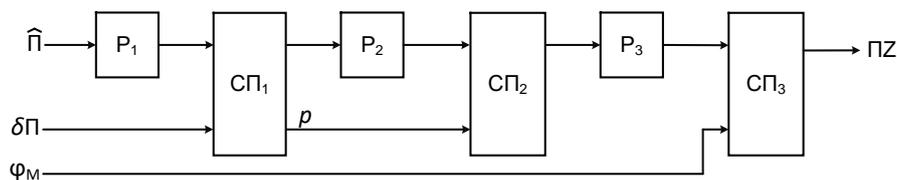


Рис. 5. Структурная схема блока ВР

Порядок \hat{P} задерживается на регистре P_1 на один такт для синхронизации с δP , а затем складывается на сумматоре $СП_1$ с δP . Промежуточный результат суммы задерживается на регистре P_2 на один такт для синхронизации с переносом p , а затем складывается на сумматоре $СП_2$ с p . Результат сумматора $СП_2$ задерживается на три такта на регистре P_3 , а затем на сумматоре $СП_3$ складывается с переполнением мантиссы φ_M , тем самым выполняя коррекцию порядка. Сумматоры $СП_1$, $СП_2$ и $СП_3$ полностью соответствуют сумматору C_2 блока ВDP.

Блок ВR осуществляет синхронизацию MZ и $ПZ$, а также формирует результат Z . Время формирования зависит от разрядности r обрабатываемых операндов X и Y , которая определяется как сумма разрядности порядка результата ($rПZ$) и разрядности мантиссы результата (rMZ). Момент формирования Z определяется стробом входных данных mXY , задержанным в блоке ВR на число тактов, равное латентности формирования $ПZ$. Сначала на выход Z коммутируется шина $ПZ$, а затем – шина MZ по алгоритму:

- 1) $CЧ_Z = \begin{cases} 0, & \text{если } mXY = 1 \\ CЧ_Z + 1, & \text{если иначе} \end{cases}$;
- 2) $mZ = (CЧ_Z=0)$;
- 3) $Z = ПZ$, если $0 \leq CЧ_Z \leq rПZ$;
- 4) $Z = MZ$, если $rПZ \leq CЧ_Z \leq rПZ + rMZ - 1$;
- 5) $Z = 0$, если $CЧ_Z \geq (rПZ + rMZ)$.

На рис. 6 представлена временная диаграмма разработанного сумматора.

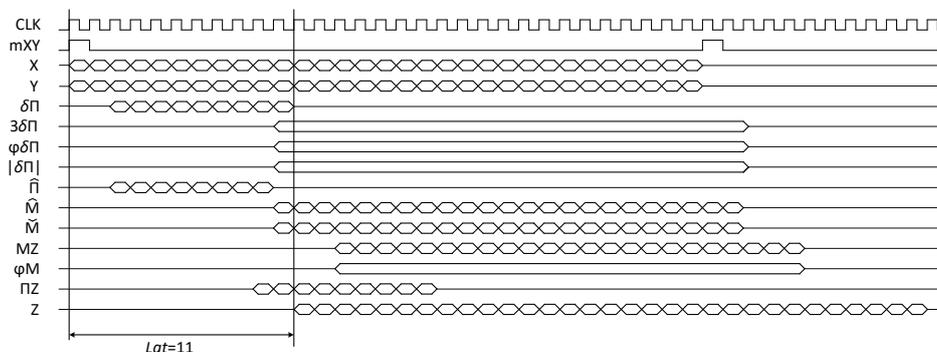


Рис. 6. Временная диаграмма работы последовательного знакоразрядного сумматора ЦФVУ в формате с плавающей запятой

Применение в ЦФVУ последовательных вычислений старшими разрядами вперёд позволяет создать знакоразрядный сумматор в формате с плавающей запятой с латентностью 11 тактов, что, например, меньше латентности параллельного двоичного сумматора (12 тактов). Более того, латентность сумматора можно снизить, если перейти к обработке операндов с порядками меньшей разрядности. Это особенно важно для оптимизации аппаратных затрат на синхронизацию вычислений в ЦФVУ при реализации вычислительных структур решения задач.

Оценка эффективности. Точность представления информации в знакоразрядной системе счисления несколько отличается от точности операций в традиционной бинарной арифметике, как целочисленной, так и с плавающей запятой [15, 16]. Так, в четверичной знакоразрядной системе счисления для обеспечения одинаковых с обычной двоичной системой диапазонов и точности представления чисел требуется в два раза больше бит, которые необходимы как для покрытия избыточности знакоразрядной системы, так и для обеспечения требуемой точности. Однако значительным преимуществом при выполнении знакоразрядных операций является почти в два раза меньшая разрядность кодов промежуточных значений [17]. Кроме того, при представлении операндов, например для рассмотренного в статье знакоразрядного сумматора в формате с плавающей запятой, используются прямая кодировка разрядов для положительных чисел и дополнительная кодировка для отрицательных, за счёт чего достигается большая простота последовательного суммирования по сравнению с параллельным устройством.

Для оценки эффективности применения в ЦФВУ четверичной знакоразрядной системы счисления был реализован 40-разрядный сумматор в базе примитивов ЦФВУ на реконфигурируемом компьютере «Терциус 3», построенном на FPGA XCVU095 [18]. Примитивы ЦФВУ, предположительно, могут содержать только двухместные логические функции и D-триггеры. Также были учтены ограничения ЦФВУ по разветвленности сигналов (fanout) и глубине логики.

На «Терциус 3» тактовая частота реализации составила $\nu = 500$ МГц. Поскольку сумматор обрабатывает 40-разрядные операнды последовательно, то скажность подачи данных составила $S = 40$. Производительность сумматора $P = \nu/S$ составила 12,5 МФлопс.

Сравнение точности вычислений на знакоразрядном сумматоре с двоичным параллельным суммированием в стандарте IEEE754 на случайных числах в диапазоне от 10^{-6} до 10^6 показало среднюю погрешность вычислений 0,015%, а максимальная погрешность составила 0,12%. Таким образом, использование знакоразрядной системы счисления в ЦФВУ потенциально позволяет решать вычислительно-трудоемкие задачи, требующие высокой точности вычислений. Следует отметить, что переход на знакоразрядную систему счисления потребует переработки элементарных арифметико-логических операций, таких как умножение, деление, извлечение корня и других, что влечет за собой необходимые изменения математических алгоритмов решения трудоёмких задач в связи с изменениями в точности выполнения операций.

Однако тактовая частота ЦФВУ [19] предполагается в районе 1 ТГц, соответственно производительность знакоразрядного сумматора на ЦФВУ составит около 25 ГФлопс. В то же время производительность эквивалентного по точности двоичного параллельного 32-разрядного сумматора IEEE754 [20] на «Терциус 3» на частоте $\nu = 500$ МГц составляет 0,5 ГФлопс. При этом затраты логических элементов и триггеров FPGA на реализацию знакоразрядного сумматора в базе ЦФВУ примерно в 3,5 раза ниже, чем на реализацию параллельного двоичного сумматора IEEE754. Поэтому реальное ускорение ЦФВУ относительно FPGA составит примерно 175 раз.

Исследование выполнено в рамках научной программы Национального центра физики и математики, направление №1 «Национальный центр исследования архитектур суперкомпьютеров. Этап 2023–2025».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Bérut Antoine.* Information and Thermodynamics: Experimental Verification of Landauer's Principle Linking Information and Thermodynamics. – URL: <https://arxiv.org/pdf/1503.06537.pdf> (дата обращения: 28.10.2022).
2. 10 лет до 10 нм: закон Мура все ещё работает // PCNews, 12.07.2008. – URL: <http://pcnews.ru/news/10-channalweb-intel-pat-gelsinger-100-tsmc-45-2009-1965-33-1971-1978-1989-1997-25-2005-65-pentium-233904.html> (дата обращения: 28.10.2022).
3. *Cerofolini C.F., Mascolo D.* Hybrid Route From CMOS to Nano and Molecular Electronics // Nanotechnology for electronic materials and devices. – Springer Science+Business Media, LLC, 2007. – P. 1-65.
4. *Степаненко С.А.* Фотонный компьютер: структура и алгоритмы, оценки параметров // Фотоника. – 2017. – № 7/67. – DOI: 10.22184/1993-7296.2017.67.7.72.83.

5. *Сорокин Д.А., Левин И.И., Касаркин А.В.* Перспективная архитектура цифровой фотонной вычислительной машины // Известия ЮФУ. Технические науки. – 2022. – № 4 (2022). – С. 200-212. – DOI 10.18522/2311-3103 2022.
6. *Sorokin D.A., Kasarkin A.V., Podoprigora A.V.* Elements of a Digital Photonic Computer // Supercomputing Frontiers and Innovations. – 2023. – Vol. 10, No. 2. – P. 62-76. – DOI: <https://doi.org/10.14529/jsfi230205>.
7. *Сорокин Д.А., Левин И.И., Касаркин А.В.* Обзор моделей коммутационных подсистем цифровых фотонных вычислительных устройств // Известия ЮФУ. Технические науки. – 2024. – № 5 (2024). – С. 173-185. – DOI 10.18522/2311-3103-2024-5-185-194.
8. *Каляев А.В., Левин И.И.* Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: Янус-К, 2003. – 380 с.
9. *Сергеев А.М.* Об особенностях представления чисел при знакоразрядном кодировании и вычислительный эксперимент с ними // Информационно-управляющие системы. – 2006. – № 3 (22). – С. 56-58.
10. *Евстигнеев В.Г.* Недвоичные компьютерные арифметики // Электроника и информатика. – 2005: Междунар. науч.-техн. конф. – М.: Ангстрем, 2006. – 774 с.
11. *Орлов Дмитрий.* Влияние ошибок округления на результаты алгоритмов вычислительной геометрии // Лекция 2 Проблемы организации вычислений. Национальный исследовательский университет «МЭИ» Кафедра Вычислительных машин, систем и сетей.
12. *Каляев А.В.* Многопроцессорные системы с программируемой архитектурой. – М.: Радио и связь, 1984. – 240 с.
13. *Каляев А.В., Левин И.И.* Многопроцессорные системы с перестраиваемой архитектурой: концепции развития и применения // Наука – производству. – 1999. – № 11. – С. 11-19.
14. *Amir Kaivani, Seokbum Ko.* Floating-Point Butterfly Architecture Based on Binary Signed-Digit Representation // IEEE transactions on very large scale integration (VLSI) systems. – March 2016. – Vol. 24, No. 3.
15. *Kung H.T.* Harvard University. High-order-bit First Conversion for Signed-Digit Representations // Annual GOMACTech Conference. – IEEE, 2021.
16. *Arash Eghdamian, Azman Samsudin.* An Improved Signed Digit Representation of Integers // Indian Journal of Science and Technology. – October 2017. – Vol 10 (39). – DOI: 10.17485/ijst/2017/v10i39/119863. – ISSN (Print): 0974-684. – ISSN (Online): 0974-5645.
17. *Andrew G Dempster, Malcolm David Macleod.* Generation of Signed-Digit Representations for Integer Multiplication // Signal Processing Letters, IEEE September 2004. – DOI: 10.1109/LSP.2004.831725.
18. UltraScale FPGA Product Tables and Product Selection Guide. – Режим доступа: <https://docs.amd.com/v/u/en-US/ultrascale-fpga-product-selection-guide> (дата обращения: 18.06.2025).
19. *Степаненко С.А.* Фотонная вычислительная машина. Принципы реализации. Оценки параметров // Доклады Академии наук. – 2017. – Т. 476, № 4. – С. 389-394. – DOI: 10.1134/S1064562417050234.
20. IEEE Standard for Floating-Point Arithmetic. – Режим доступа: <https://ieeexplore.ieee.org/document/8766229> (дата обращения: 18.06.2025).

REFERENCES

1. *Bérut Antoine.* Information and Thermodynamics: Experimental Verification of Landauer's Principle Linking Information and Thermodynamics. Available at: <https://arxiv.org/pdf/1503.06537.pdf> (accessed 28 October 2022).
2. 10 let do 10 nm: zakon Mura vse eshche rabotaet [10 years to 10 nm: Moore's Law still works], *PCNews*, 12.07.2008. Available at: <http://pcnews.ru/news/10-channalweb-intel-pat-gelsing-100-tsmc-45-2009-1965-33-1971-1978-1989-1997-25-2005-65-pentium-233904.html> (accessed 28 October 2022).
3. *Cerofolini C.F., Mascolo D.* Hybrid Route From CMOS to Nano and Molecular Electronics, *Nanotechnology for electronic materials and devices*. Springer Science+Business Media, LLC, 2007, pp. 1-65.
4. *Stepanenko S.A.* Fotonnyy komp'yuter: struktura i algoritmy, otsenki parametrov [Photonic computer: structure and algorithms, parameter estimates], *Fotonika* [Photonics], 2017, No. 7/67. DOI: 10.22184/1993-7296.2017.67.7.72.83.
5. *Sorokin D.A., Levin I.I., Kasarkin A.V.* Perspektivnaya arkhitektura tsifrovoy fotonnoy vy-chislitel'noy mashiny [Promising architecture of a digital photonic computing machine], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2022, No. 4 (2022), pp. 200-212. DOI 10.18522/2311-3103 2022.
6. *Sorokin D.A., Kasarkin A.V., Podoprigora A.V.* Elements of a Digital Photonic Computer, *Supercomputing Frontiers and Innovations*, 2023, Vol. 10, No. 2, pp. 62-76. DOI: <https://doi.org/10.14529/jsfi230205>.

7. Sorokin D.A., Levin I.I., Kasarkin A.V. Obzor modeley kommutatsionnykh podsystem tsifrovyykh fotonnykh vychislitel'nykh ustroystv [Review of models of switching subsystems of digital photonic computing devices], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2024, No. 5 (2024), pp. 173-185. DOI 10.18522/2311-3103-2024-5-185-194.
8. Kalyaev A.V., Levin I.I. Modul'no-narashchivaemye mnogoprotsessornye sistemy so strukturno-protsedurnoy organizatsiey vychisleniy [Modularly scalable multiprocessor systems with structural-procedural organization of computations]. Moscow: Yanus-K, 2003, 380 p.
9. Sergeev A.M. Ob osobennostyakh predstavleniya chisel pri znakorazryadnom kodirovani i vychislitel'nyu eksperiment s nimi [On the features of number representation with sign-based coding and a computational experiment with them], *Informatsionno-upravlyayushchie sistemy* [Information and Control Systems], 2006, No. 3 (22), pp. 56-58.
10. Evstigneev V.G. Nedvoichnye komp'yuternye arifmetiki [Non-binary computer arithmetic], *Elektronika i informatika*, 2005: Mezhdunar. nauch.-tekhn. konf. [Electronics and Information Technology – 2005: International Scientific and Technical Conference]. Moscow: Angstrom, 2006, 774 p.
11. Orlov Dmitriy. Vliyanie oshibok okrugleniya na rezul'taty algoritmov vychislitel'noy geometrii [The influence of rounding errors on the results of computational geometry algorithms], *Lektsiya 2 Problemy organizatsii vychisleniy. Natsional'nyy issledovatel'skiy universitet «MEI» Kafedra Vychislitel'nykh mashin, sistem i setey* [Lecture 2. Problems of computing organization. National Research University "MPEI" Department of Computing Machines, Systems and Networks].
12. Kalyaev A.V. Mnogoprotsessornye sistemy s programmiruemoi arkhitekturoy [Multiprocessor systems with programmable architecture]. Moscow: Radio i svyaz', 1984, 240 p.
13. Kalyaev A.V., Levin I.I. Mnogoprotsessornye sistemy s perestraivaemoi arkhitekturoy: kontseptsii razvitiya i primeneniya [Multiprocessor systems with reconfigurable architecture: concepts of development and application], *Nauka – proizvodstvu* [Science – production], 1999, No. 11, pp. 11-19.
14. Amir Kaivani, Seokbum Ko. Floating-Point Butterfly Architecture Based on Binary Signed-Digit Representation, *IEEE transactions on very large scale integration (VLSI) systems*, March 2016, Vol. 24, No. 3.
15. Kung H.T. Harvard University. High-order-bit First Conversion for Signed-Digit Representations, *Annual GOMACTech Conference. IEEE*, 2021.
16. Arash Eghdamian, Azman Samsudin. An Improved Signed Digit Representation of Integers, *Indian Journal of Science and Technology*, October 2017, Vol 10 (39). DOI: 10.17485/ijst/2017/v10i39/119863. – ISSN (Print): 0974-684. – ISSN (Online): 0974-5645.
17. Andrew G Dempster, Malcolm David Macleod. Generation of Signed-Digit Representations for Integer Multiplication, *Signal Processing Letters, IEEE September 2004*. DOI: 10.1109/LSP.2004.831725.
18. UltraScale FPGA Product Tables and Product Selection Guide. Available at: <https://docs.amd.com/v/u/en-US/ultrascale-fpga-product-selection-guide> (accessed 18 June 2025).
19. Stepanenko S.A. Fotonnaya vychislitel'naya mashina. Printsipy realizatsii. Otsenki parametrov [Photonic computing machine. Implementation principles. Parameter estimates], *Doklady Akademii nauk* [Reports of the Academy of Sciences], 2017, Vol. 476, No. 4, pp. 389-394. DOI: 10.1134/S1064562417050234.
20. IEEE Standard for Floating-Point Arithmetic. Available at: <https://ieeexplore.ieee.org/document/8766229> (accessed 18 June 2025).

Сорокин Дмитрий Анатольевич – НИЦ супер-ЭВМ и нейрокомпьютеров; e-mail: jotun@inbox.ru; г. Таганрог, Россия; тел.: +79508668253; начальник отдела; к.т.н.

Левин Илья Израилевич – НИЦ супер-ЭВМ и нейрокомпьютеров; г. Таганрог, Россия; e-mail: levin@superevm.ru; тел.: +78634612111; директор; д.т.н.; профессор.

Sorokin Dmitriy Anatolyevich – Supercomputers and Neurocomputers Research Center; e-mail: jotun@inbox.ru; Taganrog, Russia; phone: +79508668253; chief of Department; cand. of eng. sc.

Levin Ilya Izrailevich – Supercomputers and Neurocomputers Research Center; e-mail: levin@superevm.ru; Taganrog, Russia; phone: +78634612111; director Supercomputers and Neurocomputers Research Center; dr.of eng. sc.; professor.