

УДК 681.3: 004.031

МУЛЬТИКОНВЕЙЕРНЫЕ СХЕМЫ ДЛЯ ОБРАБОТКИ УНИТАРНЫХ КОДОВ В ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Е. А. Титенко, к.т.н, доцент, доцент кафедры ПИ ЮЗГУ, Курск, Россия;

orcid.org/0000-0002-5659-4747 e-mail: johnnit@mail.ru

Э. И. Ватутин, д.т.н, профессор, доцент кафедры ВТ ЮЗГУ, Курск, Россия;

orcid.org/0000-0002-7362-7387, e-mail: evatutin@rambler.ru

И. Е. Чернецкая, д.т.н, доцент, заведующий кафедрой ВТ ЮЗГУ, Курск, Россия;

orcid.org/0009-0009-8254-9606, e-mail: white731@yandex.ru

Д. А. Трокоз, д.т.н., доцент, ведущий разработчик Р7, Нижний Новгород, Россия;

orcid.org/0000-0003-0989-3928, e-mail: dmitriy.trokoz@gmail.com

М. А. Титенко, аспирант ЮЗГУ, Курск, Россия;

orcid.org/0009-0004-0819-1707, e-mail: mikhail-titenko@mail.ru

И. И. Пузанов, аспирант ЮЗГУ, Курск, Россия;

orcid.org/0009-0004-8193-7520, e-mail: ivan.puzanov2000@yandex.ru

Рассматривается задача обеспечения работы вычислительных систем, связанная с реализацией параллельных вычислений над битовыми потоками (потоками унитарных кодов). Целью работы является сокращение времени работы конвейерных схем при обработке унитарных кодов. Такие коды являются носителями числовой и символьной информации одновременно. Дуализм данных служит отправочной точкой в организации параллельных вычислений на базе итеративных сетей. Показано, что однородные итеративные сети являются основой для создания различных конвейерных схем. Они могут принимать и обрабатывать несколько входных потоков на основе заложенной в ячейки сети мультифункциональности. Сравнение полученных решений в моно- и мультиконвейерных схемах показало относительный временной выигрыш последних при незначительном повышении аппаратных затрат.

Ключевые слова: итеративная сеть, стартовая точка, многопоточность, вычислительная ячейка.

DOI: 10.21667/1995-4565-2025-93-61-71

Введение

Современные высокопроизводительные вычислительные системы (ВС), вычислительные сети и их элементы [1-4], создаются для эффективного решения вычислительно трудоемких и поисковых задач. Расчетно-модельные или проблемно-поисковые задачи вычислительной физики и химии, генетического моделирования, анализа активности социальных сетей, обработки и распознавания изображений, машинного перевода, математической лингвистики, комбинаторики слов и др. представляют такой класс задач [5-7]. Эти задачи характеризуются естественным параллелизмом данных, в них обрабатываются как числовые, так и символьные данные. Такой подход определяет поиск и разработку нестандартных архитектур и вычислительных структур, ориентированных на управление потоком данных и обработку символьной информации. При этом выполнение независимых операций математического, логического, структурно-лингвистического вида подчиняется модели потока данных (DataFlow Model) [8], а их результаты представляются битовыми потоками, элементы которых описывают количественные и структурные характеристики вычислительных процессов, выполняемых параллельно [9-11].

Структурно битовый поток состоит из набора унитарных кодов (УК), каждый из которых представляет числовую последовательность, количественное значение которого определяется количеством логических «1» в его составе. С другой стороны, биты по отдельности или группами задают структурные свойства УК как строки.

Высокопроизводительная обработка УК возможна в рамках однородных вычислительных структур – конвейеров, позволяющих за счет совмещения операций на различных ступенях вести одновременную обработку нескольких УК из входного потока данных [4, 8, 11]. Под производительностью традиционно понимается количество операций, выполняемых в единицу времени [1, 8].

Исторически над УК в однородных ВС чаще всего выполнялись арифметические операции, связанные с количественными оценками параллельно выполняемых процессов, операции сравнения с числовым эталоном (порогом) и др. Тем не менее, в настоящее время вопросы структурной обработки групп бит (префиксов, суффиксов) в составе унитарного кода являются достаточно значимыми для оценки результатов параллельных процессов проблемно-поисковых задач и задач искусственного интеллекта [12].

Постановка задачи

В таблице 1 приведены наиболее важные операции, выполняемые над УК, где ЧИ – числовая информация, СИ – символьная информация.

Таблица 1 – Базовые операции над УК

Table 1 – Basic operation on unitary codes

№	Операция	Тип данных
1	Нормализация УК	ЧИ
2	Подсчет количества логических «1»	ЧИ
3	Выделение приоритетной логической «1» (арбитраж)	СИ
4	Формирование правой/левой серии логических «1»	СИ
5	Формирование правой/левой серии логических «0»	СИ
6	Вычисление адреса первой правой/левой логической «1»	ЧИ
7	Преобразование «весовой код → нормализованный УК»	ЧИ
8	Преобразование «нормализованный УК → весовой код»	ЧИ
9	Правая/левая инверсия УК	СИ

При выполнении данных операций над УК время T_{conv} работы конвейера рассматривается как целевая функция, зависящая от длины кода ($length$), количества входных потоков данных ($thread$), многофункциональности ячеек (ступеней) конвейера ($multifunctions$), конфигурации самого конвейера ($configuration$)

$$T_{conv}(length, thread, multifunctions, configuration) \rightarrow \min. \quad (1)$$

В качестве ограничений на (1) рассматриваются однородные конвейеры с конфигурацией ($configuration$), использующей двунаправленную обработку УК и соответственно $multifunctions \leq 2$.

Теоретические исследования

Теоретической основой создания однородных конвейеров являются итеративные сети [8, 13-15] и вычислительные структуры, использующие принципы однородности состава и регулярности связей между ячейками конвейера [11]. Итеративная сеть – это однородная система вычислительных ячеек для передачи, обработки информационных и управляющих сигналов в конвейере. На основе кода конфигурации конвейера и набора вычисляемых функций итеративная сеть формирует свою вычислительную структуру, далее осуществляет прием входных операндов (унитарных кодов) и реализует рекуррентные вычисления с выдачей выходного УК по ступеням конвейера. Главная особенность конвейеров – последовательное сраба-

тывание вычислительных ячеек, начиная со стартовой точки, которая задается на одной из граничных ячеек сети.

Типовыми примерами организации рекуррентных вычислений по цепочке однородных вычислительных ячеек можно считать схемы параллельного сумматора с последовательными переносами, арбитра, регистра сдвига, счетчика и других комбинационных и автоматных схем [16-19]. При этом в зависимости от типа однородного функционального устройства выбор стартовой точки для рекуррентных вычислений может быть предопределен (для сумматоров) или быть произвольным (для арбитров). Данная особенность существенно влияет на организацию вычислений и общее время работы конвейера. Однако системные принципы назначения/вычисления стартовых точек в организации методов и схемотехнических решений итеративных сетей рассмотрены ограниченно [4, 14]. Конвейерная обработка на основе итеративных сетей подразумевает, в большинстве случаев, единственное фиксированное направление и одну стартовую точку.

Применительно к обработке УК наибольшее распространение получили одномерные итеративные сети с одним направлением связей между вычислительными ячейками. Направление связи между ячейками в итеративной сети, как правило, директивно не регулируется, а определяется исходя из предпочтений разработчика. Тем не менее, для ряда строковых операций направление обхода не влияет на результат, что может служить источником создания наиболее рациональных схем.

На рисунке 1 показаны однонаправленные одномерные итеративные сети, составляющие основу типовых конвейеров. Они преобразуют входной код $X = x_1x_2...x_n$ в выходные коды $Y = y_{11}y_{12}...y_{1n}$ или $Y = y_{21}y_{22}...y_{2n}$ с вычислением соответствующего связующего кода $V = v_1v_2...v_nv_{n+1}$, где v_1 – стартовая точка при направлении обхода ячеек слева направо, а v_{n+1} – стартовая точка при направлении обхода ячеек справа налево. Стартовые точки имеют размер в один бит, они задаются исходя из логики выполняемой функции.

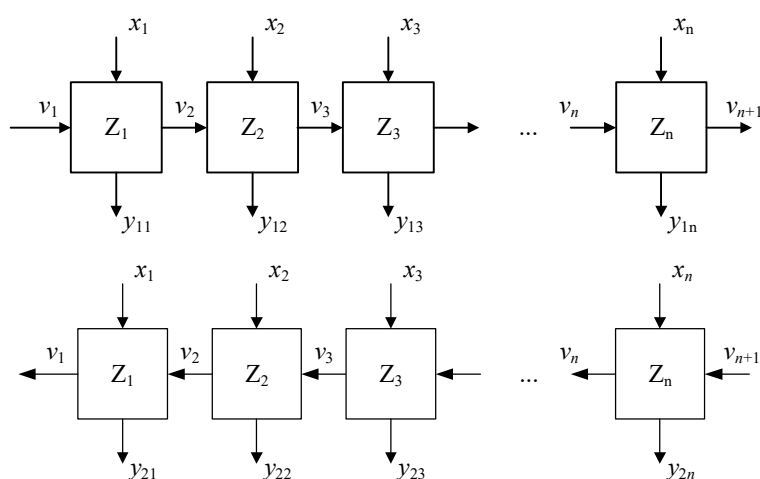


Рисунок 1 – Однонаправленные одномерные итеративные сети
Figure 1 – Directional linear iterative networks

Выбор направления обхода ячеек диктуется особенностями задачи, например, поиск приоритетного значения слева и/или справа в структуре УК (арбитраж слева/справа), выполнение строковых операций над фрагментами строки и других операций, не чувствительных к порядку обхода элементов УК.

Механическое объединение двух сетей влечет лишь удвоение аппаратной сложности без эмерджентного эффекта от такого объединения, так как ячейки каждой итеративной сети не связаны друг с другом.

Эмерджентный эффект может быть достигнут при двунаправленной обработке входных данных в рамках одной итеративной сети. В этом случае конвейер будет иметь две стартовые

точки. На рисунке 2 показана двунаправленная итеративная сеть, состоящая из вычислительных ячеек, передающих две встречные связующие функции. Такая организация обеспечивает выполнение двух вычислительных процессов в каждый момент времени, что приводит к уменьшению времени выполнения данных операций за счет одновременной обработки двух бит УК с обеих сторон сети.

Известно, что при однонаправленной обработке УК время работы T итеративной сети линейно зависит от длины кода n и составляет $T = nt_{CELL}$, где t_{CELL} – время срабатывания одной вычислительной ячейки Z_i ($i=1 \dots n$). Итоговый момент времени окончания работы итеративной сети определяется моментом времени срабатывания граничной ячейки сети Z_n или Z_1 .

Напротив, в двунаправленной одномерной итеративной сети (рисунок 2) в каждый момент времени формируется по два бита (y_{1i} , y_{2n-i}) выходных кодов $Y = y_{11}y_{12} \dots y_{1n}$ и $Y = y_{21}y_{22} \dots y_{2n}$.

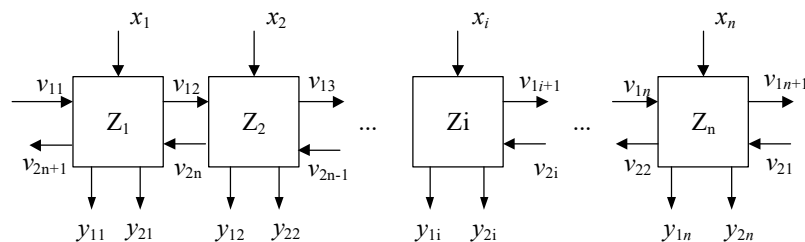


Рисунок 2 – Двунаправленная одномерная итеративная сеть
Figure 2 – Bidirectional linear iterative network

Несмотря на обход всех ячеек итеративной сети по каждому направлению, время, затрачиваемое на вычисления выходных кодов $Y = y_{11}y_{12} \dots y_{1n}$ и $Y = y_{21}y_{22} \dots y_{2n}$, будет меньше по сравнению с однонаправленными конвейерами, так как в двунаправленной итеративной сети на каждом такте выполняется одновременная обработка пары бит из УК. Таким образом, общее время работы конвейера уменьшается за счет одновременного выполнения двух самостоятельных вычислительных процессов, каждый из которых имеет собственную стартовую точку (входные биты на граничных ячейках Z_1 , Z_n).

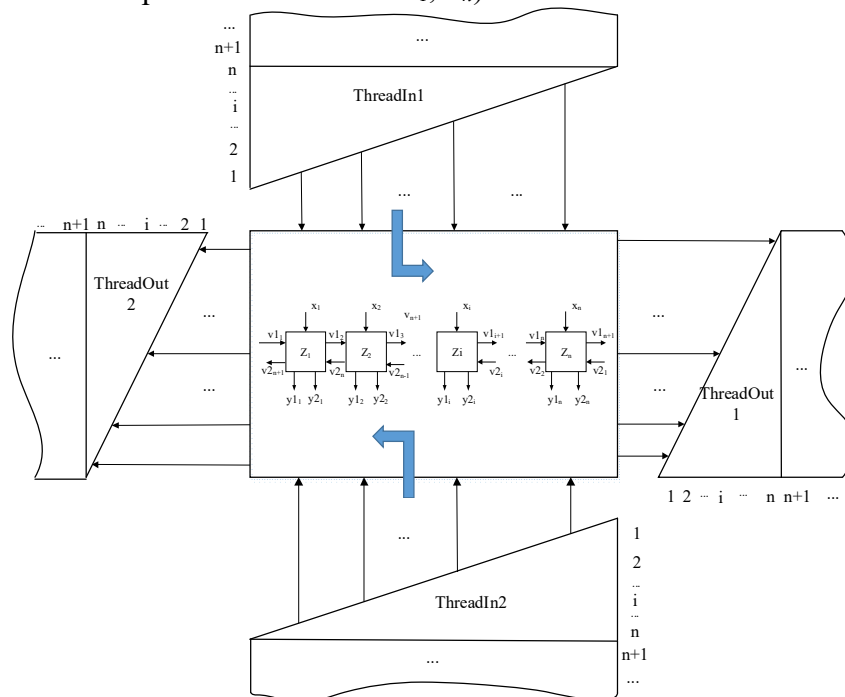


Рисунок 3 – Мультиконвейерная схема подготовки и обработки потоков УК
Figure 3 – Multi-pipeline scheme for unitary threads preparation and processing

В общем случае, работа мультиконвейерной схемы заключается в параллельной подаче двух входных потоков УК – ThreadIn 1, ThreadIn 2, каждый из которых имеет специальную подготовку элементов УК в виде их диагонального размещения в потоке. Такая подготовка обеспечивает параллельную обработку УК в составе каждого из двух входных потоков при полной загрузке конвейера каждым потоком. Другая вычислительная особенность мультиконвейерной схемы – задание в рамках ячеек итеративной сети двух встречных вычислительных процессов путем назначения каждому из них самостоятельной стартовой точки. Каждый вычислительный процесс охватывает все ячейки итеративной сети, но за счет одновременного выполнения нескольких операций в каждом такте выдаются биты-результат в каждом из двух выходных потоков – ThreadOut 1, ThreadOut 2.

На рисунке 3 показана схема подготовки и выдачи потоков УК для мультиконвейерной схемы с двумя встречными вычислительными процессами по всем его ступеням (ячейкам сети).

Экспериментальные исследования

Работа традиционного конвейера с одним направлением вычислений описывается следующей схемой подготовки и подачи различных элементов единственного входного потока УК (рисунок 4).

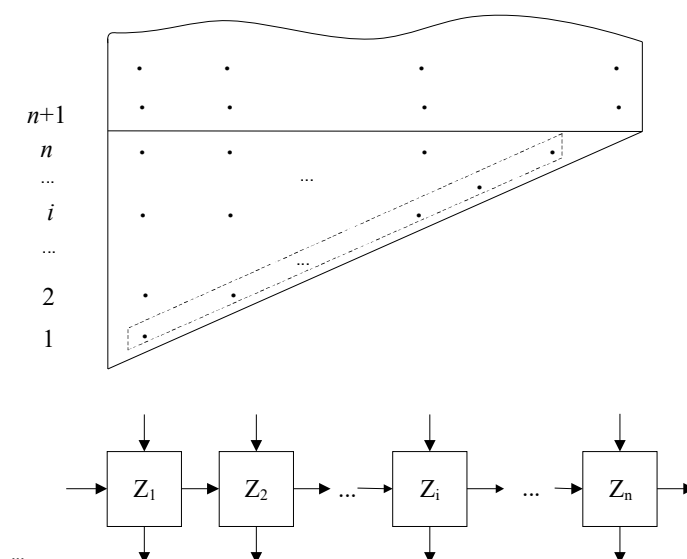


Рисунок 4 – Однонаправленный конвейер обработки потока УК
Figure 4 – Directional pipeline for unitary threads processing

Полное время T_{conv} работы конвейера определяется временем t_{fill} его загрузки данными и временем t_{work} потактовой работы

$$T_{conv} = t_{fill} + t_{work} . \quad (2)$$

Параметрами потока УК являются: k - количество подаваемых УК, n – длина унитарного кода.

«Диагональная» подача элементов УК приводит к тому, что время загрузки всех ячеек конвейера занимает n тактов, а количество элементов УК, обработанных ячейками конвейера за n тактов, составляет $(n+1)*n/2$ элементов. Далее с $n+1$ -го до k -го тактов конвейер осуществляет потактовую обработку n элементов УК в составе потока, выдавая на каждом такте работы выходной УК.

Для конвейера с единственным входным потоком его время T_{conv} работы оценивается как

$$T_{conv} = \underbrace{n}_{t_{fill}} + \underbrace{1+1+\dots+1}_{t_{work}} = n + k . \quad (3)$$

В рамках двух встречных процессов мультиконвейер обрабатывает отдельные входные потоки УК и выдает (вправо и влево) выходные потоки. Для такой конфигурации мульти-

конвейер независимо обрабатывает каждый входной поток, одновременно начиная вычисления с граничных ячеек итеративной сети.

Время загрузки мультиконвейера будет по-прежнему равно $t_{fill} = n$, так как элементы каждого входного УК подаются последовательно по одному. Тем не менее, мультифункциональность позволяет совместить выполнение двух вычислительных функций на конвейере для двух входных потоков ThreadIn 1, ThreadIn 2. Следовательно, для такого конвейера общее время обработки входных потоков составит

$$T_{conv} = n + k / 2. \quad (4)$$

Дальнейшее уменьшение времени работы мультиконвейера достигается не только увеличением числа одновременно выполняемых вычислительных функций над элементами УК, но и созданием схем ускоренной загрузки итеративной сети. Научная новизна мультиконвейерных схем определяется специальной распределенной подачей пар элементов УК. Такой подход позволяет в рамках одной итеративной сети организовывать несколько локальных конвейеров, имеющих дополнительные стартовые точки. Тогда общее время загрузки конвейера и время последующей обработки элементов УК будет кратно уменьшено количеству формируемых локальных конвейеров, в которых выполняются двунаправленные вычислительные процессы. Время вычисления дополнительных стартовых точек зависит от состава УК и его длины. Тем не менее, эти вычисления выполняются один раз над УК и позволяют распараллелить обработку УК по его частям.

На рисунке 5 приведен вариант схемы подачи элементов УК в двунаправленную итеративную сеть, начиная с крайних элементов УК. Подача элементов УК организована таким образом, чтобы две текущие ячейки Z_{q1} , Z_{q2} ($q1=1 \dots n/2$, $q2=n/2+1 \dots n$) сети формировали два бита v_{q1+1} v_{q2-1} связующих кодов для двух следующих вычислительных ячеек Z_{q1+1} , Z_{q2-1} . При этом синхронно с работой двух текущих ячеек Z_{q1} , Z_{q2} сети в конвейере обеспечивается распределенная подача двух следующих элементов из состава УК. Каждая из половин УК (префикс и суффикс) одновременно обрабатывается локальным конвейером с двунаправленными связями ячеек.

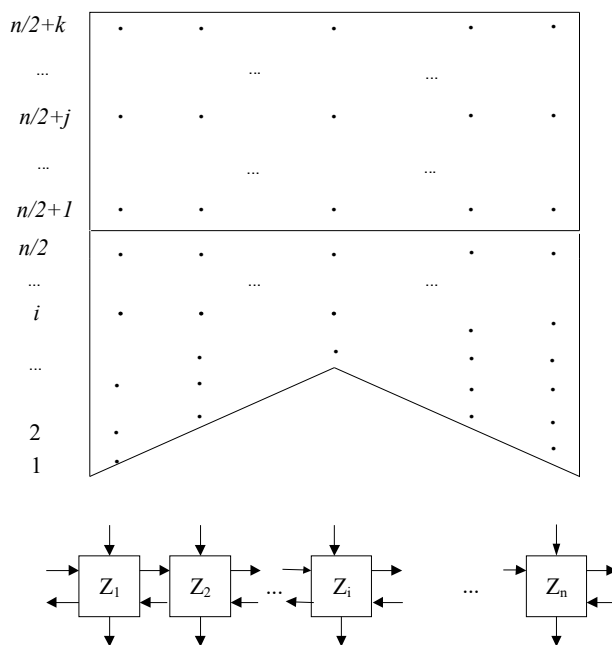


Рисунок 5 – Схема формирования входного потока УК для итеративной сети
Figure 5 – Scheme for preparing initial UC stream for iterative network

Задача арбитража слева/справа является общезначимой задачей для обеспечения параллельных вычислений в многопроцессорных ВС. Фактически эта задача сводится к последовательному просмотру УК как строки с ее обеих границ. Целевая миссия – обнаружение пер-

вой логической «1» слева и справа в составе УК и сброс остальных логических «1». На рисунке 6 показана структура вычислительной ячейки, формирующей на каждом такте работы конвейера две выходные функции $Y_1 = y_{11}y_{12} \dots y_{1n}$ и $Y_2 = y_{21}y_{22} \dots y_{2n}$, содержащие по одной логической «1».

Как итог, для мультиконвейера с распределенной подачей элементов УК общее время обработки входных потоков определяется как

$$T_{conv} = n / (2 * num_thread) + k / num_function, \quad (5)$$

где num_thread – число входных потоков, $num_function$ – число выполняемых функций.

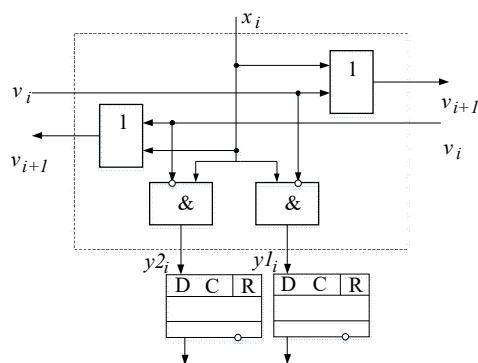


Рисунок 6 – Структура вычислительной ячейки мульти конвейера
Figure 6 – Structure of multi pipeline computational cell

На рисунке 7 показаны временные зависимости времени T работы конвейеров для типовой функции арбитража, где $(num_thread, num_function)$ – параметры конвейера. Для расчета зависимости времени $T(n)$ работы конвейера приняты задержки: время инициализации конвейера – 2τ , время работы ячейки $t_{CELL} = 2,5\tau$, время вычисления дополнительных стартовых точек – $\log_2(n/2)\tau$. В расчетах также принято, что организация много функциональности и много поточности по сравнению с моно- конвейером связана с введением накладных временных расходов на настройку конфигурации мультиконвейеров, подготовку исходных данных.

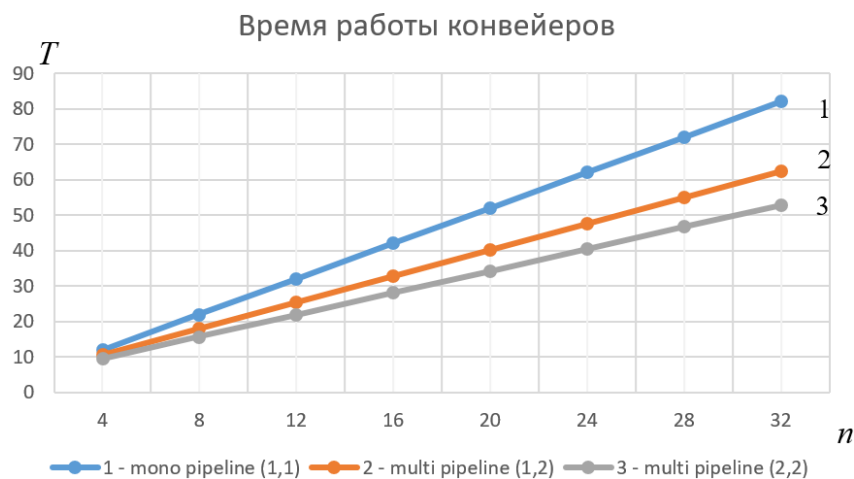


Рисунок 7 – Временные зависимости работы конвейеров
Figure 7 – Time scheme of pipeline operation

Анализ линейных временных зависимостей (1, 2, 3) показал, что с увеличением длины n унитарного кода преимущество мультиконвейерных схем с многопоточной обработкой увеличивается. На интервале длины n унитарного кода от 16 до 32 бит сравнение времени вычисления функций арбитража двунаправленным мультиконвейером показало временной выигрыш от 1,49 до 1,55 раз по сравнению с моно- конвейером. В целом, данные графики подтверждают сокращение времени работы мульти конвейеров при одновременном выполнении нескольких функций над потоками унитарных кодов.

Заключение

Современные высокопроизводительные ВС (мультипроцессоры, ассоциативные процессоры, многофункциональные процессорные вычислительные поля, реконфигурируемые ВС и др.) [20-23] формируют определенные результаты в виде последовательностей унитарных кодов.

Унитарные коды используются во многих проблемно-поисковых и информационно-аналитических, когнитивных задачах и процессах. Они представляются как числами, так и строками. Для их эффективной количественной и структурной обработки целесообразно использовать многофункциональные мультиконвейеры на базе однородных итеративных сетей. Объединение классических принципов конвейеризации с принципами многопоточной и мультифункциональной обработки позволяет уменьшить время работы конвейера пропорционально длине унитарного кода. Апробация на типовых функциях арбитража показала временной выигрыш до 1,55 раз по сравнению с моно- конвейером.

Таким образом, мультиконвейерные схемы на базе однородных итеративных сетей с двунаправленной обработкой элементов УК позволяют вести параллельно-конвейерные вычисления в однородных вычислительных системах.

Библиографический список

1. Voevodin V.V., Voevodin V.I. Parallel computing. St. Petersburg, BHV-Peterburg Publ., 2002. 608 p.
2. Гузик В.Ф., Каляев И.А., Левин И.И. Реконфигурируемые вычислительные системы. Ростов-на-Дону: Южный федеральный университет. 2016. 472 с.
3. Перепелкин Д.А., Фам А.М. Математическое и компьютерное моделирование процессов планирования и распределения разнородных ресурсов в промышленных сетях // Вестник Рязанского государственного радиотехнического университета. 2021. № 77. С. 68-80.
4. Каляев И. А., Левин И.И. Реконфигурируемые вычислительные системы на основе ПЛИС. Ростов-на-Дону: Южный научный центр РАН, 2022. 475 с.
5. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
6. Lothaire M. Applied Combinatorics on Words. In: Encyclopedia of Mathematics and its Applications. Cambridge: Cambridge University Press. 2005.
7. Титенко Е.А. Parallel Sorting Algorithm for the Operation of Industrial Intelligent Systems // 2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). 2021. P. 65-69.
8. Бурцев В.С. Параллелизм вычислительных процессов и развитие архитектуры суперЭВМ: сб. статей / сост. В.П. Торчигин, Ю.Н. Никольская, Ю.В. Никитин. М.: ТОРУС ПРЕСС, 2006. 416 с.
9. Титенко Е.А, Петрик Е.А., Воронин Д.А., Атакищева И.В. Продукционная модель для параллельной обработки знаний // Информационно-измерительные и управляющие системы. 2011. Т. 9. № 11. С. 81-86.
10. Титенко Е.А, Семенихин Е.А., Петрик Е.А., Воронин Д.А. Структурно-функциональная организация арбитра параллельной обработки запросов // Информационно-измерительные и управляющие системы. 2010. Т. 8. № 11. С. 30-34.
11. Огнев И.В., Борисов В.В., Сутула Н.А. Ассоциативные память, среды, системы. М.: Горячая линия – Телеком. 2016. 420 с.
12. Рыбина Г.В. Основы построения интеллектуальных систем: учеб. пособие. М.: Финансы и статистика, ИНФРА-М, 2010. 432 с.
13. Бандман О.Л, Миренков Н.Н., Седухин С.Г. Специализированные процессоры для высокопроизводительной обработки данных. М.: Радио и связь, 1988. 208 с.
14. Фет Я.И. Параллельные процессоры для управляющих систем / Я. И. Фет. Москва: Энергоиздат. 1981. 160 с.
15. Ватугин Э.И., Зотов И.В., Титов В.С. Использование схемных формирователей и преобразователей двоичных последовательностей при построении комбинаторно-логических акселераторов // Известия Курского государственного технического университета. 2008. № 4 (25). С. 32-39.
16. Зельдин Е.А. Цифровые интегральные микросхемы в информационно-измерительной аппаратуре. Л.: Энергоатомиздат. 1986. 280 с.

17. **Титенко Е.А., Скорняков К.С., Бусыгин К.Н.** Методы и сумматоры с параллельными групповыми процессами // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2013. № 1. С. 161-166.
18. **Титенко Е.А., Типикин А.П., Лапин Д.В.** Некоторые пути построения перспективных вычислительных систем для параллельной обработки массивов данных и изображений на ПЛИС // Электромагнитные волны и электронные системы. 2016. Т. 21. № 10. С. 56-59.
19. **Ханис А.Л., Беспалько С.В., Титенко Е.А., Фролов С.Н., Сазонова Н.Н.** Обзор исследований киберстрахования // Интеллектуальные информационные системы: тенденции, проблемы, перспективы: материалы докладов VII всероссийской очной научно-практической конференции «ИИС-2019». Курск: Юго-Западный государственный университет, 2019. С. 108-118.
20. **Кравец О.Я., Подвальный Е.С., Титов В.С., Ястребов А.С.** Архитектура вычислительных систем с элементами конвейерной обработки: учеб. пособие. Воронеж, Курск, СПб.: Политехника, 2009. 151 с.
21. **Зерин И.С., Атакищев О.И., Титенко Е.А., Корольков О.Ф., Атакищев А.О.** Метод, алгоритм и техническое решение параллельного поиска и подстановки на ассоциативной памяти // В мире научных открытий. 2012. № 1-1(25). С. 166-180.
22. **Страбыкин Д.А.** Разработка и экспериментальное исследование функциональной модели ассоциативного запоминающего устройства в Microsoft Excel // Современные наукоемкие технологии. 2023. № 7. С. 97-103. DOI: 10.17513/snt.39701.
23. **Мартышкин А.И.** Реализация узлов на программируемых логических интегральных схемах реконфигурируемой вычислительной системы // Вестник Рязанского государственного радиотехнического университета. 2023. № 85. С. 36-43. DOI: 10.21667/1995-4565-2023-85-36-43.

UDC 681.3: 004.031

MULTI-PIPELINE SCHEMES FOR UNITARY CODES PROCESSING IN HIGH-PERFORMANCE COMPUTING SYSTEMS

E. A. Titenko, Ph.D. in technical sciences, associate professor, associate professor, department of computer engineering, Kursk, Russia;

orcid.org/0000-0002-5659-4747 e-mail: johnntit@mail.ru

E. I. Vatutin, Dr. in technical sciences, full professor, associate professor, department of computer engineering, SWSU, Kursk, Russia;

orcid.org/ 0000-0002-7362-7387, e-mail: evatutin@rambler.ru

I. E. Chernetskaya, Dr. in technical sciences, associate professor, head of the department of computer engineering, SWSU, Kursk, Russia;

orcid.org/0009-0009-8254-9606), e-mail: white731@yandex.ru

D. A. Trokoz, Dr. in technical sciences, associate professor, leading developer in R7; Nizhniy Novgorod, Russia;

orcid.org/ 0000-0003-0989-3928, e-mail: dmitriy.trokoz@gmail.com

M. A. Titenko, postgraduate student, SWSU, Kursk, Russia;

orcid.org/0009-0004-0819-1707, e-mail: mikhail-titenko@mail.ru

I. I. Puzanov, postgraduate student, SWSU, Kursk, Russia;

orcid.org/0009-0004-8193-7520, e-mail: ivan.puzanov2000@yandex.ru

The article considers the problem of ensuring the operation of computing systems associated with the implementation of parallel calculations over bit streams (streams of unitary codes). The aim of the work is to increase specific performance of pipeline circuits for processing unitary codes. Such codes are carriers of both numerical and symbolic information. Data dualism is the starting point while organizing multi-threaded computations based on iterative networks. The authors show that iterative networks are the basis for creating various pipeline schemes. They can accept and process several input streams based on multi functionality embedded in network cells.

The comparison of obtained circuit solutions in mono- and multi-pipeline schemes has shown the time gain of the latter with insignificant increase in hardware costs.

Keywords: *iterative network, starting point, multifunctionality, computational cell.*

DOI: 10.21667/1995-4565-2025-93-61-71

References

1. **Voevodin V.V., Voevodin V.I.** *Parallel computing*. St. Petersburg, BHV-Peterburg Publ., 2002. 608 p.
2. **Guzik V.F., Kalyaev I.A., Levin I.I.** Rekonfiguriruyemye vychislitel'nye sistemy (Reconfigurable computing systems). Rostov-na-Donu: Yuzhnyj federal'nyj universitet. 2016. 472 p. (in Russian).
3. **Perepelkin D.A., Fam A.M.** Matematicheskoe i komp'yuternoe modelirovanie processov planirovaniya i raspredeleniya raznorodnykh resursov v promyshlennyykh setyakh. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2021, no.77, pp. 68-80. (in Russian).
4. **Kalyaev I. A., Levin I.I.** *Rekonfiguriruyemye vychislitel'nye sistemy na osnove PLIS* (Reconfigurable computing systems based on FPGA). Rostov-na-Donu: Yuzhnyj nauchnyj centr RAN. 2022. 475 p. (in Russian).
5. **Gehri M., Dzhonson D.** Vychislitel'nye mashiny i trudnoreshaemye zadachi (Computing machines and difficult-to-solve problems). Moscow: Mir.1982. 416 p. (in Russian).
6. **Lothaire M.** Applied Combinatorics on Words. In: *Encyclopedia of Mathematics and its Applications*. Cambridge: Cambridge University Press. 2005.
7. **Titenko E.A.** Parallel Sorting Algorithm for the Operation of Industrial Intelligent Systems. *2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*. 2021, pp. 65-69.
8. **Burcev V.S.** *Parallelizm vychislitel'nykh processov i razvitie arkhitektury superEHVM* (Parallelism of computing processes and the development of supercomputer architecture): collection of articles sb. statej / sost. V.P. Torchigin, Yu.N. Nikol'skaya, Yu.V. Nikitin. Moscow: TORUS PRESS, 2006. 416 p. (in Russian).
9. **Titenko E. A., Petrik E. A., Voronin D. A., Atakishcheva I. V.** Produkcionnaya model' dlya paralel'noj obrabotki znaniy. *Informacionno-izmeritel'nye i upravlyayushchie sistemy*. 2011, vol. 9, no.11, pp. 81-86. (in Russian).
10. **Titenko E. A., Semenikhin E.A., Petrik E. A., Voronin D. A.** Strukturno-funkcional'naya organizatsiya arbitra paralel'noj obrabotki zaprosov. *Informacionno-izmeritel'nye i upravlyayushchie sistemy*. 2010, vol. 8, no. 11, pp. 30-34. (in Russian).
11. **Ognev I.V., Borisov V.V., Sutula N.A.** *Associativnye pamyat', sredy, sistemy* (Associative memory, environments). Moscow: Goryachaya liniya – Telekom. 2016. 420 p. (in Russian).
12. **Rybina G.V.** *Osnovy postroeniya intellektual'nykh sistem* (Fundamentals of Building Intelligent Systems): ucheb. posob. Moscow: Finansy i statistika, INFRA-M, 2010. 432 p. (in Russian).
13. **Bandman O.L., Mirenikov N.N., Sedukhin S.G.** *Specializirovannye processory dlya vysokoproizvoditel'noj obrabotki dannykh* (Specialized Processors for High-Performance Data Processing). Moscow: Radio i svyaz', 1988. 208 p. (in Russian).
14. **Fet YA. I.** Parallelnye processory dlya upravlyayushchikh system (Parallel Processors for Control Systems). Moscow: Ehnergoizdat. 1981. 160 p. (in Russian).
15. **Vatutin Eh. I., Zotov I.V., Titov V.S.** Ispol'zovanie skhemnykh formirovatelej i preobrazovatelej dvoichnykh posledovatel'nostej pri postroenii kombinatorno-logicheskikh akseleratorov. *Izvestiya Kurskogo gosudarstvennogo tekhnicheskogo universiteta*. 2008, no. 4(25), pp. 32-39. (in Russian).
16. **Zel'din E.A.** *Cifrovye integral'nye mikroskhemy v informacionno-izmeritel'noj apparature* (Digital integrated circuits in information-measuring equipment). L.: Ehnergoatomizdat, 1986. 280 p. (in Russian).
17. **Titenko E.A., Skorniyakov K.S., Busygin K.N.** Metody i summatory s paralel'nymi gruppovymi processami. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika, informatika. Medicinskoe priborostroenie*. 2013, no. 1, pp. 161-166. (in Russian).
18. **Titenko E.A., Tipikin A.P., Lapin D.V.** Nekotorye puti postroeniya perspektivnykh vychislitel'nykh sistem dlya paralel'noj obrabotki massivov dannykh i izobrazhenij na PLIS. *Ehlektromagnitnye volny i ehlektronnye sistemy*. 2016, vol. 21, no. 10, pp. 56-59. (in Russian).
19. **Khanis A.L., Bepal'ko S.V., Titenko E.A., Frolov S.N., Sazonova N.N.** Obzor issledovaniy kibernetizatsii. *Intellektual'nye informacionnye sistemy: tendencii, problemy, perspektivy: materialy dokladov VII vserossijskoj ochnoj nauchno-prakticheskoy konferencii «IIS-2019»*. Kursk: Yugo-Zapadnyj gosudarstvennyj universitet. 2019, pp. 108-118. ((in Russian).

20. **Kravec O.Ya., Podval'nyj E.S., Titov V.S., Yastrebov A.S.** *Arkhitektura vychislitel'nykh sistem s ehlementami konvejernoj obrabotki* (Architecture of computing systems with elements of pipeline processing): uchebn. posobie Voronezh, Kursk, Sankt-Peteburg: Politehnika, 2009. 151 p. (in Russian).

21. **Zerin I.S., Atakishchev O.I., Titenko E.A., Korol'kov O.F., Atakishchev A.O.** Metod, algoritm i tekhnicheskoe reshenie parallel'nogo poiska i podstanovki na associativnoj pamyati. *V mire nauchnykh otkrytij*. 2012, no. 1-1(25), pp. 166-180. (in Russian).

22. **Strabykin D.A.** Razrabotka i ehksperimental'noe issledovanie funktsional'noj modeli associativnogo zapominayushchego ustrojstva v Microsoft Excel. *Sovremennye naukoemkie tekhnologii*. 2023, no. 7, pp. 97-103. (in Russian). DOI: 10.17513/snt.39701.

23. **Martyshkin A.I.** Realizatsiya uzlov na programmiruemykh logicheskikh integral'nykh skhemakh rekonfiguriruemoy vychislitel'noj sistemy. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2023, no.85, pp. 36-43. (in Russian). DOI: 10.21667/1995-4565-2023-85-36-43.