

Левин Илья Израилевич – Общество с ограниченной ответственностью «НИЦ супер-ЭВМ и нейрокомпьютеров»; e-mail: levin@superevm.ru; 347900, г. Таганрог, пер. Итальянский, 106; тел.: +78634612111; директор; д.т.н.; профессор.

Федоров Александр Михайлович – e-mail: fedorov@superevm.ru; начальник отдела.

Доронченко Юрий Иванович – e-mail: doronchenko@superevm.ru; технический директор; к.т.н.

Раскладкин Максим Константинович – e-mail: raskladkin@superevm.ru; начальник отдела; к.т.н.

Levin Ilya Izrailevitch – ООО “SRC of supercomputers and neurocomputers”; e-mail: levin@superevm.ru; 106, Italyanskiy street, Taganrog, 347900, Russia; phone: +78634612111; director; dr. of eng. sc.; professor.

Fedorov Alexander Mikhailovitch – e-mail: fedorov@superevm.ru; head of department.

Doronchenko Yuriy Ivanovitch – e-mail: doronchenko@superevm.ru; technical director; cand. of eng. sc.

Raskladkin Maxim Konstantinovitch – e-mail: raskladkin@superevm.ru; head of department; cand. of eng. sc.

УДК 681.3+004.32

DOI 10.18522/2311-3103-2020-7-19-34

Е.А. Титенко, Е.В. Талдыкин

КОММУТАЦИОННАЯ МОДЕЛЬ ПАРАЛЛЕЛЬНЫХ СРАВНЕНИЙ ЭЛЕМЕНТОВ ДЛЯ ПРОДУКЦИОННЫХ СИСТЕМ, УПРАВЛЯЕМЫХ ПОТОКОМ ДАННЫХ

В статье достигается цель – сокращение временных затрат на генерацию сочетаний элементов множества. Элементы множества формируются из образцов (левых частей) продукционных правил. Основная задача заключается в построении эффективных по времени схем (алгоритмов) параллельной генерации сочетаний элементов массива. Применительно к продукционным системам такие схемы необходимы для активации подмножества продукции, применимых к символьным данным на текущем шаге. За основу взят и развит известный алгоритм параллельного пузырька. Схема коммутации «параллельный пузырьрек» состоит из двух чередующихся вариантов коммутации элементов в пары. Эти коммутации основаны на локальном объединении в пары элементов массива, имеющих смежные индексы. Такое локальное объединение элементов в пары приводит к «малым» перемещениям элементов по длине массива и регулярному характеру генерации пар. В каждой паре выполняется операция сравнения-обмена операндов. Для продукционных систем операция сравнения сводится к поиску пересечений образцов и формированию списка конфликтных слов. Сокращение времени генерации сочетаний основывается на построении вариантов коммутации с распределенным объединением элементов в пары с шагом, равным 4. Разработанная схема коммутации содержит на нечетных шагах коммутации с локальным объединением элементов в пары. На четных шагах выполняется коммутация-ускоритель с распределенным объединением элементов в пары. Моделирование работы разработанной схемы коммутации осуществлялось на типовых задачах сортировки и полного перебора пар элементов. Установлено сокращение временных затрат по сравнению с четно-нечетной сортировкой на 15–18%. В работе определена линейная зависимость времени сортировки с углом наклона меньше 1. Это позволяет использовать схему коммутации для продукционных систем большого размера. Локальные и распределенные связи в схеме коммутации сохраняют свойство регулярности. Эта особенность определяет аппаратную реализацию схемы в виде параллельного коммутатора с естественным масштабированием. Данная схема может использоваться в специализированном продукционном устройстве для декомпозиции продукционной системы на независимые подмножества продукции.

Параллельная сортировка; схема коммутаций; однородное устройство.

E.A. Titenko, E.V Taldykin

SWITCHING MODEL OF PARALLEL COMPARISONS FOR A DATA FLOW RULE BASED SYSTEMS

The article is show the reduction of time spent on generating combinations of elements of the set. The elements of the set are formed from samples (left parts) of the production rules. The main task is to build time-efficient schemes (algorithms) for parallel generation of combinations of array elements. With regard to production systems, such schemes are necessary for the activation of a subset of products applicable to character data in the current step. The basis is taken and developed the well-known algorithm of the parallel bubble. The switching circuit "parallel bubble" consists of two alternating variants of switching elements in pairs. These commutations are based on local union into pairs of array elements with adjacent indices. Such a local combination of elements into pairs leads to "small" displacements of elements along the length of the array and the regular nature of the generation of pairs. In each pair, the operation of comparison-exchange of operands is performed. For production systems, the comparison operation is reduced to the search for sample intersections and the formation of a list of conflicting words. The reduction in the generation time of combinations is based on the construction of switching options with distributed combining of elements in pairs with a step equal to 4. The developed switching scheme contains on odd switching steps with a local combination of elements in pairs. In even-numbered steps, a switching accelerator is performed with a distributed combination of elements in pairs. The simulation of the work of the developed switching scheme was carried out on typical tasks of sorting and complete enumeration of pairs of elements. The reduction of time costs compared with the scheme "parallel bubble" by 15-18%. A linear dependence of the sorting time with a slope angle less than 1 was determined. This allows the use of a switching circuit for large-scale production systems. Local and distributed communications in the switching scheme preserve the property of regularity. This feature determines the hardware implementation of the circuit in the form of a parallel switch with natural scaling. This scheme can be used in a specialized production device for decomposing a production system into independent subsets of products.

Parallel sorting; commutation circuit; multi unit.

Введение. Однородные вычислительные устройства, состоящие из регулярно соединенных ячеек, модулей или блоков, с единым устройством управления представляют перспективный путь создания высокопроизводительных вычислительных устройств (ВУ), совмещающих в операционной части функциональные блоки и коммутационную схему передачи промежуточных результатов [1–3].

Как правило, такие ВУ эффективно применимы для проблемно-поисковых, комбинаторных задач принятия решений, задач обработки символьной информации (ОСИ) с генерацией множества решений [4–6]. В них массово значимыми являются операции множественной генерации и оценки вариантов, тасования, преобразований «бабочка», сортировки, приоритетного выбора, множественного сравнения данных, ассоциативного поиска, поиска и модификации по образцу (шаблону) и др. [7–9]. Такие операции основаны на множественных обменах промежуточных данных, нерегулярных соединениях операндов между собой в информационном графе задачи [10].

Достаточно эффективным подходом обеспечения производительности ВУ на таких задачах является структурное соответствие информационного графа задачи и графа операционной части ВУ с реконfigurацией связей между вершинами графа ВУ [3, 7]. Однако такой путь проектирования ВУ приводит к их специализации, что ограничивает область применения.

Структурное несоответствие графов приводит к функционально и аппаратно избыточным техническим схемам с низкими однородностью и масштабированием. Как следствие, время решения таких задач имеет полиномиальную, степенную сложности, поэтому последовательные алгоритмы и аппаратные решения, разрабатываемые для микропроцессоров и систем общего назначения [11, 12], неэффективны для них.

Обработка символьной информации (ОСИ), связанная с генерацией, анализом и отбором вариантов на основе исчислительных продукционных моделей, является значимым классом проблемно-поисковых задач, использующих схему вычислений «условие→действие» [13]. Главное отличие таких продукционных систем (ПС) или моделей – это естественная ориентация на управление потоком данных [13]. Управление потоком данных основывается на параллельном исполнении такого подмножества правил (продукций), для которых выполнено условие их активации (положительный поиск по образцу) [14–16]. Вторым условием параллельного исполнения потока продукций является наличие достаточного количества функциональных блоков (ячейки, блоки, модули) в операционной части [17, 18].

Распараллеливание потока данных для исчислительных продукционных моделей имеет особенности, связанные с обеспечением полного или направленного перебора и анализа продукций, входящих в ПС, на предмет их систематизации, перестроения, т.е. предобработки ПС. [19, 20]. Сущность управления в такой ПС сводится к заданию ветвящихся вычислительных процессов, отображаемых в виде графа вывода. Недетерминированный характер правил из ПС, управляемых потоком данных, уточняется как возможность их объединения и параллельного выполнения. Это требование объединения набора правил из ПС в текущую активацию обуславливает высокую гибкость коммутационных схем, комбинаторно реализующих выборку правил на выполнение. Основная проблемная ситуация связана с сокращением возвратно-переборных шагов на синтез и выполнение подмножеств активированных продукций.

В связи с этим задача создания эффективных по времени аппаратно-ориентированных коммутационных схем в рамках продукционной парадигмы вычислений является общезначимой задачей теоретической информатики и вычислительной техники.

Постановка задачи. Пусть заданы $n \in \mathbb{N}$, рабочий алфавит A , метасимволы $\rightarrow \notin A$, $*$ $\notin A$, слова $O_i, P_i \in A^*$, $i=1-n$. Пусть задано обрабатываемое слово $S \in A^*$.

Продукция – это правило в алфавите $A \cup \{\rightarrow\}$ вида [9, 13]

$$O \rightarrow P. \quad (1)$$

где O – слово-образец в алфавите A ; P – слово-модификатор в алфавите A .

Продукционная система U задается как система-тройка:

$$U = \{A, \Theta, C\}, \quad (2)$$

где C – схема (стратегия) управления продукциями, Θ – определяющее множество продукций.

Упрощенно, ПС представляется множеством продукций, объединенных на решение задачи стратегией Θ управления потоком данных (потоком активированных продукций).

$$\Theta : \begin{cases} O_1 \rightarrow P_1 \\ \dots \\ O_i \rightarrow P_i \\ \dots \\ O_n \rightarrow P_n \end{cases} \quad (3)$$

Для снижения ресурсных затрат на работу ПС (объем памяти, время перебора продукций) предлагается ее декомпозиция на обособленные подмножества, замкнутые относительно операций алгебры продукций. В этом случае работа ПС разбивается на автономные вычисления по каждому подмножеству с сохранением корректности конечного результата [19, 26].

Декомпозиция ПС основана на алгебре продукций, включающей операции конструктивной логики пересечения, объединения, дополнения слов и др. Эти операции попарно устанавливают свойства структурной связи между образцами и подстановками продукций вида (1). Над ПС выполняются подготовительные вычислительные действия, позволяющие исследовать связи между продуктами и разбить их на обособленные подмножества. Тем не менее характер подготовительных действий имеет комбинаторный характер, а их вычислительная сложность – полиномиальную зависимость, что определяет востребованность специализированных устройств или функциональных узлов, поддерживающих как базовые операции поиска вхождений, пересечений, объединений слов (образцы, подстановки), так и комбинирования образцов и подстановок в пары.

Количество проверок на парные пересечения слов определяется как число сочетаний $C_n^2 = \frac{n(n-1)}{2}$. Квадратичная сложность выражения C_n^2 , при больших n или интенсивных модификациях ПС является достаточной критичной для работы ПС.

В связи с этим требуются аппаратные средства обеспечения параллельных продукционных вычислений в виде схем параллельных коммутаций для перебора пар образцов (образцов и подстановок) ПС с линейной временной сложностью. Далее задача перебора рассматривается в рамках задачи сортировки массива [21].

Метод решения. Задача сортировки является классической benchmark-задачей, на которой проверяется эффективность алгоритмов перебора, сравнения и обмена сортируемых элементов, а также работоспособность схемотехнических решений, реализующих последовательные или параллельные вычисления. Структуры данных, применяемые операции и схемы коммутаций элементов, образующие промежуточные данные, составляют основу алгоритмов сортировки.

Пусть задан алфавит $A = \{a_1, a_2, \dots, a_l\}$, мощностью $|A| = n$ букв. На алфавите A задана функция отношения $F(x_1, x_2)$, $x_1 \in A$, $x_2 \in A$, определяющая порядок пары любых элементов алфавита A :

$$F(x_1, x_2) = \begin{cases} 1, & \text{если } (x_1, x_2) \text{ упорядоченная пара} \\ 0, & \text{если } (x_2, x_1) \text{ упорядоченная пара} \end{cases} \quad (1)$$

Функция (1) характеризуется свойствами:

- ◆ транзитивности – если $F(x_1, x_2) = 1$ и $F(x_2, x_3) = 1$, то $F(x_1, x_3) = 1$;
- ◆ рефлексивности – $F(x_1, x_1) = 1$;
- ◆ антисимметричности – если $F(x_1, x_2) = F(x_2, x_1)$, то $x_1 = x_2$.

В качестве функции отношения будут использованы строгие неравенства вида «<>» или «>». Эти неравенства обеспечивают возможность сортировки массива данных по возрастанию или убыванию.

Упорядоченным (отсортированным) массивом $M_{SORT} = \{m_1, m_2, \dots, m_n\}$, $|M|=n$, $m_i \in A$, $i=1..n$ является массив, в котором для любой пары (m_i, m_j) в случае $i \leq j$ выполняется равенство $F(m_i, m_j) = 1$. Алгоритм сортировки преобразует исходный массив $M = \{m_1, m_2, \dots, m_n\}$ в отсортированный массив M_{SORT} [21].

Среди всего многообразия алгоритмов сортировки в данном исследовании выделяются алгоритмы сортировки с генерацией множества параллельно выполняемых элементарных процессов сравнения над простейшими массивами - парами элементов. Образование пар элементов основано на динамически изменяемых связях в схемах коммутаций. Это свойство что приводит к образованию наборов пар элементов массива, над сформированными парами элементов массива выполняется элементарное упорядочение согласно (1), а схемы коммутаций обеспечивают перебор всевозможных пар и перемещают элементы в соответствии с отношением упорядочения букв алфавита.

В качестве потенциально возможных алгоритмов, основанных на элементарных процессах сравнения-обмена элементов, рассматриваются алгоритмы [21-24]:

- ◆ четно-нечетных перестановок;
- ◆ параллельный вариант алгоритма Шелла;
- ◆ алгоритм параллельной сортировки Бэтчера.

Также задача сортировки как частный случай перестановки также решается на основе оригинальных настраиваемых коммутационных сетей. Среди наиболее приспособленных под аппаратную сортировку выделяются сеть Каутца, сеть Стоуна [8].

В качестве показателей сравнения алгоритмов принимаются следующие показатели:

- ◆ аппаратная реализация;
- ◆ однородность загрузки;
- ◆ масштабируемость схемы коммутаций;
- ◆ степень распараллеливания;
- ◆ устойчивость сортировки;
- ◆ адаптивность сортировки;
- ◆ количество инверсий перестановок;
- ◆ требование дополнительной памяти;
- ◆ сложность управления.

Под однородностью загрузки понимается количество образуемых пар элементов по итерациям работы алгоритма. С аппаратной точки зрения легче обеспечить управление постоянным количеством коммутируемых пар. Переменное количество коммутируемых пар требует большего количества управляющих сигналов. Масштабируемость схемы основана на однородности схемы коммутаций и заключается в естественном наращивании схемы при увеличении размера исходных данных. Устойчивость – показатель качества сортировки, заключающийся в сохранении отношений следования между одноименными буквами в отсортированном множестве. Адаптивность – показатель, устанавливающий зависимость количества шагов алгоритма от состава исходных данных. В этом смысле используется еще термин чувствительность (чувствительность к данным) алгоритма. Инверсия перестановок – показатель, показывающий количество выполняемых перестановок для расположения элементов в отсортированные позиции. Общеизвестно, что среднее число инверсий перестановок равно $(n^2 - n)/4$ [22]. Соответственно, чем меньше шагов алгоритм тратит на инверсии, тем он эффективнее. Сложность управления оценивается как количество управляющих элементов (бит), приходящихся на 1 информационный элемент.

В табл. 1 приведены качественные оценки алгоритмов в шкале значений {Низкая, Средняя, Высокая}.

Таблица 1

Качественные оценки алгоритмов сортировки

№	Показатели	Алгоритмы сортировки		
		Четно-нечетная	Параллельная Шелл	Бэтчер
1	Аппаратная реализация	Высокая	Низкая	Средняя
2	Однородность загрузки	Высокая	Средняя	Средняя
3	Масштабируемость схемы	Высокая	Низкая	Средняя
4	Степень распараллеливания	Высокая	Средняя	Высокая
5	Устойчивость	Высокая	Низкая	Низкая

6	Адаптивность	Низкая	Низкая	Высокая
7	Количество инверсий	Низкая	Среднее	Высокая
8	Дополнительная память	Низкая	Средняя	Низкая
9	Сложность управления	Низкая	Высокая	Высокая

В табл. 1 первые 7 качественных показателей являются стимулирующими, последние 2 показателя регрессивные. Анализ трех алгоритмов сортировки выявил, что предпочтительным для аппаратной реализации является алгоритм четно-нечетных сортировки как имеющий большинство лучших значений в предлагаемом 9-мерном пространстве показателей.

Тем не менее, низкое значение показателя «Количество инверсий» определяет необходимость модификации схемы четно-нечетных перестановок для сокращения промежуточных шагов на устранение инверсий в образуемых парах.

Разработка модифицированной схемы коммутации. Классическая схема четно-нечетной сортировки состоит из чередующих коммутаций K1, K2, в которых соседние элементы массива M объединяются в пары (рис. 1). Массив элементов понимается как линейная структура.

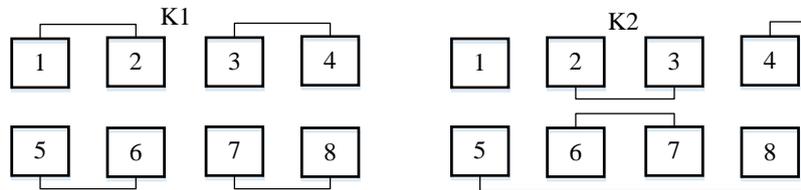


Рис. 1. Коммутации K1, K2 для четно-нечетной сортировки для $n=8$

Пусть в ПС имеется $n=2w$ правил, где $n \in \mathbf{N}$. Пусть каждый элемент массива ассоциируется с правилом вида (1). Тогда правила коммутации K1, K2 обеспечивают перебор пар продукции ПС для проверки истинности конструктивной дизъюнкции пересечения образцов [10] и последующей декомпозиции ПС на обособленные подмножества продукции.

Правила коммутации K1:

$$\forall i \forall j ((O_i^H = O_j^K) \vee (O_j^H = O_i^K) \vee (O_i \subset O_j) \vee (O_j \subset O_i)) | (i = 1, 3, \dots, n-1, j = i+1). \quad (3)$$

Правила коммутации K2:

$$\forall i \forall j ((O_i^H = O_j^K) \vee (O_j^H = O_i^K) \vee (O_i \subset O_j) \vee (O_j \subset O_i)) | (i = 2, 4, \dots, n-2, j = i+1). \quad (4)$$

Коммутация K1 формирует $n/2$ пар элементов массива M . Коммутация K2 формирует $(n/2-1)$ пар элементов массива M .

Образуемые по коммутациям K1, K2 пары элементов (i, j) имеют адреса, отличающиеся между собой на 1, что определяет связность с соседними парами $(i-1, i)$ и $(j, j+1)$. Вместе с тем такая связность уменьшает количество инверсий между элементами не более, чем на 1, что отражается в последовательных перемещениях элементов в необходимые позиции. Как следствие, наибольшее количество шагов для устранения инверсий коммутациями (3) и (4) равно n .

На рис. 2 показана четно-нечетная сортировка по убыванию массива, упорядоченного по возрастанию (один из «худших» наборов).

Для уменьшения количества шагов (сравнений-обменов) четно-нечетной сортировки предлагается модификация с кольцевой трактовкой массива M . В этом случае из граничных элементов массива формируется дополнительная пара элементов, способствующая сокращению количества инверсий (рис. 3).

	K1	K2														
1	2	4	6	8	10	12	14	16	15	13	11	9	7	5	3	1
2	1	3	5	7	9	11	13	15	14	12	10	8	6	4	2	
3	4	1	6	2	8	4	10	6	12	8	14	10	16	12	15	14
4	3	6	1	8	2	10	4	12	6	14	8	16	10	15	12	13
5	6	3	8	1	10	2	12	4	14	6	16	8	15	10	13	12
6	5	8	3	10	1	12	2	14	4	16	6	15	8	13	10	11
7	8	5	10	3	12	1	14	2	16	4	15	6	13	8	11	10
8	7	10	5	12	3	14	1	16	2	15	4	13	6	11	8	9
9	10	7	12	5	14	3	16	1	15	2	13	4	11	6	9	8
10	9	12	7	14	5	16	3	15	1	13	2	11	4	9	6	7
11	12	9	14	7	16	5	15	3	13	1	11	2	9	4	7	6
12	11	14	9	16	7	15	5	13	3	11	1	9	2	7	4	5
13	14	11	16	9	15	7	13	5	11	3	9	1	7	2	5	4
14	13	16	11	15	9	13	7	11	5	9	3	7	1	5	2	3
15	16	13	15	11	13	9	11	7	9	5	7	3	5	1	3	2
16	15	15	13	13	11	11	9	9	7	7	5	5	3	3	1	1
	1	2	3	4	5	6	7	8	9	#	11	12	13	14	15	16

Рис. 2. Пример параллельной четно-нечетной сортировки

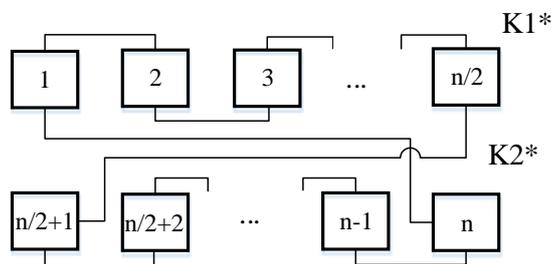


Рис. 3. Модифицированная схема четно-нечетных коммутаций

Теперь коммутации K1* и K2* обеспечивают образование n/2 пар элементов каждая.

Правила модифицированной четно-нечетной коммутации имеют вид

$$K1^*: \forall i \forall j (i, j) | i = 1, 3, \dots, n-1, j = (i+1) \bmod n \quad (5)$$

$$K2^*: \forall i \forall j (i, j) | i = 2, 4, \dots, n, j = (i+1) \bmod n. \quad (6)$$

Сравнение двух схем четно-нечетных коммутаций (рис. 1 и 3) при «матричном» представлении массива M выявило свойство замыкания элементов массива относительно операции обмена при добавлении пары (n,1). За счет дополнительной пары элементов реализуется обход массива с любой позиции, например, {8, 7, 6, 5, 4, 3, 2, 1}. Кроме того, модифицированная схема четно-нечетных коммутаций формирует параллельные пары элементов, над которыми бесконфликтно реализуются операции «сравнение-обмен»:

$$K1^*: \{(1,2), (3,4), (5,6), (7,8)\},$$

$$K2^*: \{(2,3), (4,5), (6,7), (8,1)\}.$$

Разработка специализированного устройства модифицированной четно-нечетной сортировки. Разрабатываемое специализированное устройство сортировки (СУС) строится на основе прямого отображения информационного графа задачи в граф функциональных узлов.

Информационный граф задачи описывается модифицированной схемой четно-нечетных перестановок на n элементов с правилами коммутации (5), (6). Операционная часть СУС состоит из n блоков памяти элементов (БПЭ), соединения которых образуют кольцевую структуру исходного массива M. (рис. 4). На рис. 4

соединения соседних БПЭ имеют перекрестный характер, он нужен для реализации параллельных обменов элементов в пределах каждой пары, включая пару (БПЭ_н, БПЭ₁).

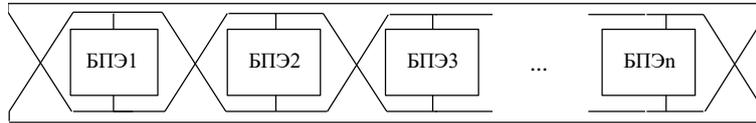


Рис. 4. Общая схема операционной части устройства сортировки

Общая структурная схема СУС (рис. 5) состоит из операционной части, блока управления (БУ) операциями «сравнение-обмен» и БУ коммутациями. БУ «сравнение-обмен» получает от операционной части осведомительные сигналы **X**, выдавая в ответ сигналы **Y**, управляющие микрооперациями записи, чтения, адресации, сравнения и др. БУ коммутациями получает от операционной части осведомительные коммутационные сигналы **KX**, выдавая в ответ управляющие коммутационные сигналы **KY** для проверки и реализации коммутационных соединений БПЭ в пары.

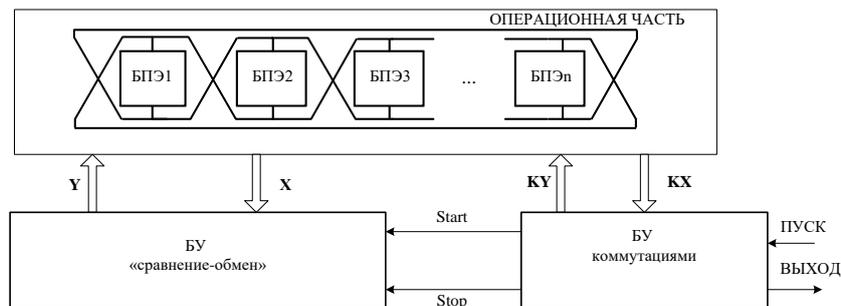


Рис. 5. Общая структурная схема СУС

Наличие двух управляющих автоматов (блоков управления) является отличительной особенностью специализированных устройств данного класса. Блок управления коммутациями является главным, он принимает внешний сигнал ПУСК из внешней среды (от хост-машины) и выдает обратно сигнал окончания работы ВЫХОД. Также БУ коммутациями:

- ◆ запускает БУ операциями «сравнение-обмен» (сигнал Start), получая от него, при необходимости, осведомительные сигналы о состоянии процессов коммутации;
- ◆ завершает работу БУ «сравнение-обмен» (сигнал Stop).

Детализированная структурная схема специализированного устройства сортировки представлена на рис. 6. Однородный состав СУС и регулярные соединения между блоками памяти элементов позволяют получить общую картину о составе операционной части устройства, его информационных, коммутационных и управляющих входах и выходах.

Каждый БПЭ_к ($k=1-n$) содержит регистр RG_k для хранения k -го элемента из массива M и буферный регистр $BuffRG_k$ для реализации обмена. Обмен реализуется на основе мультиплексора MX с однобитовым адресным входом A . Информационные входы мультиплексора (D_0, D_1) подключены к соседним БПЭ. В целом, сортировка четно-нечетными коммутациями обеспечивается подачей чередующейся последовательности на вход A в виде $\{0,1,0,1 \dots\}$ и выполнением последовательности микроопераций до тех пор, пока не будет получен отсортированный массив.

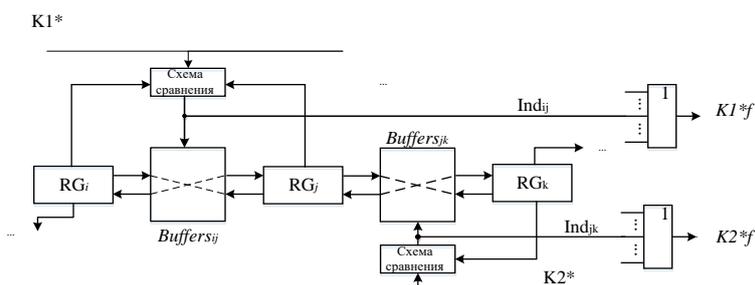


Рис. 6. Структурная схема СУС

Здесь $Buffers_{ij}$, $Buffers_{jk}$ – буферные пары регистров для промежуточного хранения обмениваемых элементов из RG_i , RG_j , RG_k соответственно. Выходы Ind_{ij} , Ind_{jk} необходимы для проверки условия завершения сортировки. Это условие формируется на основе двух выходных сигналов $K1*f$ и $K2*f$, подаваемых в БУ коммутациями. БУ коммутациями прекращает сортировку в соответствии с условием

$$STOP = K1 * f \ \& \ K2 * f = 1. \quad (7)$$

Выходные сигналы завершения коммутаций $K1*$, $K2*$ формируются на основе многовыходных элементов ИЛИ, которые проверяют нулевые значения от выходов схем сравнения – индикаторы Ind .

На рис. 7 приведена функциональная схема ij -й пары БПЭ.

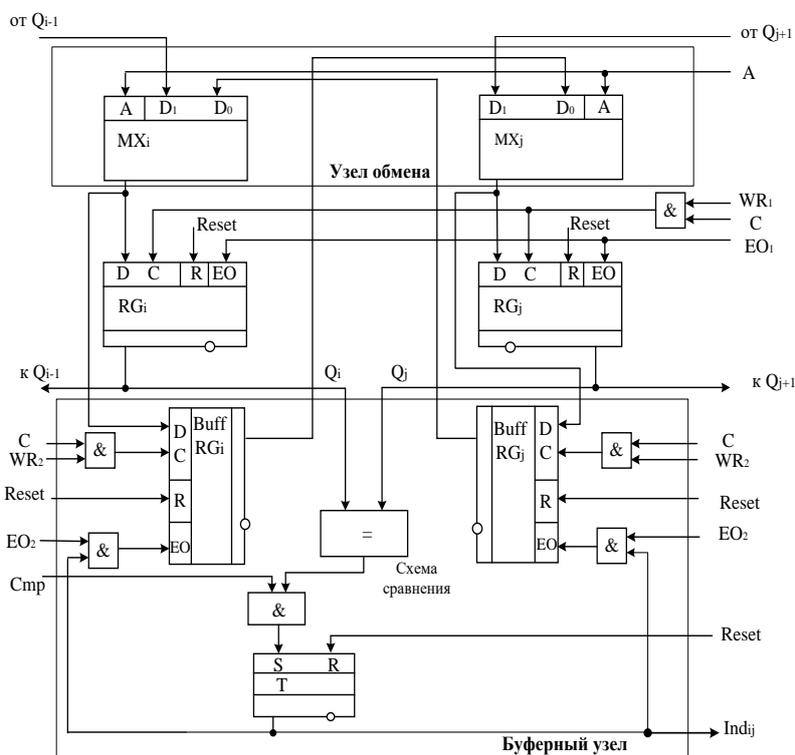


Рис. 7. Функциональная схема пары БПЭ специализированного устройства сортировки

Схема состоит из регистров RG_i и RG_j с высоко импедансными выходами Q_i , Q_j , буферного узла, включая схему сравнения (компаратор), и узла обмена пар элементов. Буферный узел содержит пару буферных регистров $BuffRG_i$ $BuffRG_j$ и асинхронный RS-триггер для фиксации результата сравнения элементов. Схема содержит информационные входы и выходы от соседних БПЭ – Q_{i-1} , Q_{j+1} соответственно.

Адресный вход A , управляемый коммутационными входами $K1^*$ и $K2^*$, обеспечивает в зависимости от варианта коммутации (5) или (6) подключение на вход регистров RG_i и RG_j одной из пар значений:

$$(RG_i, RG_j) = \begin{cases} (RG_j, RG_i), & \text{если } A = 0 \\ (RG_{i-1}, RG_i) \& (RG_{j+1}, RG_j), & \text{если } A = 1 \end{cases} \quad (8)$$

Функциональная схема (рис. 7) содержит следующие управляющие входы:

- ◆ Reset- вход сброса;
- ◆ WR_1, WR_2 – входы записи в регистры и буферные регистры;
- ◆ EO_1, EO_2 – входы разрешения (подключения) выходов регистров;
- ◆ C – вход синхронизации;
- ◆ $Стр$ – вход разрешения обменной записи Q_j и Q_i в буферные регистры.

Аппаратная поддержка микроопераций коммутации, сравнения пар элементов, обменных записей и проверки условия окончания работы позволяет синтезировать алгоритм работы БУ «сравнение-обмен». Алгоритм представляет собой цикл с постусловием, в нем самая длительная операция связана с обменной записью элементов в буферные регистры (функциональные схемы начальной загрузки значений $Q_1 - Q_n$ в БПЭ1 – БПЭn не показаны).

Логическое условие разбиения БПЭ на пары (обозначается как «!») формируется в БУ коммутациями на основе комбинации коммутационных сигналов $K1^*$ и $K2^*$:

$$! = K1^* \& \bar{K}2^* \quad (9)$$

На рис. 8 показана блок-схема алгоритма работы БУ «сравнение-обмен». Фактически по управляющему сигналу вида (9) реализуется одна из коммутаций - $K1^*$ или $K2^*$, объединяющая в пары элементы в четно-нечетных позициях массива и при необходимости реализующая перестановки в регистрах БПЭ1 – БПЭn. Алгоритм блока управления «сравнение-обмен» представляет собой цикл с постусловием. Условие выхода из цикла ($Stop=1$) формируется в БУ коммутациями.

Для корректности подсчета количества шагов сортировки данный алгоритм синхронизируется с алгоритмом управления коммутациями по переменной $CountRes$. Начальное значение $CountRes=0$ устанавливается в алгоритме управления коммутациями, а инкрементируется эта переменная в алгоритме «сравнение-обмен». Такое распределение по синхронизируемой переменной $CountRes$ позволяет контролировать состояние работы 2-х блоков управления. При $CountRes=2$ алгоритм «сравнение-обмен» приостанавливается для проверки условия окончания работы СУС. Проверка выполняется в БУ коммутациями.

Алгоритм управления коммутациями (рис. 9) проверяет значение логической переменной $flag$. Если выполнена хотя бы 1 микрооперация обмена по коммутациям $K1^*$ или $K2^*$, то остается $flag=0$. Тогда сбрасывается переменная $CountRes=0$, и работа алгоритма «сравнение-обмен» продолжается с сохранением $Stop=0$.

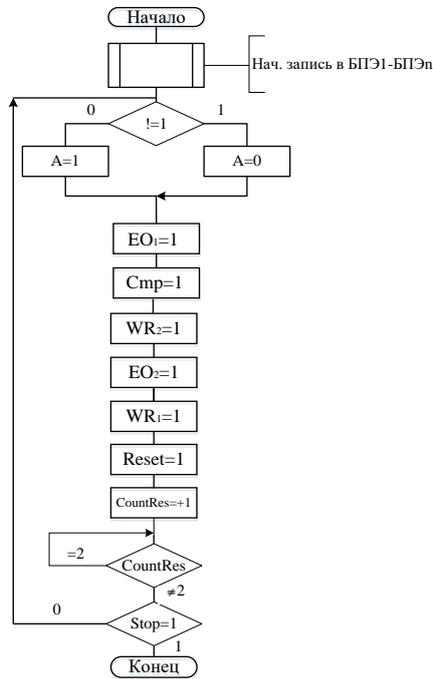


Рис. 9. Блок-схема алгоритма БУ «сравнение-обмен»

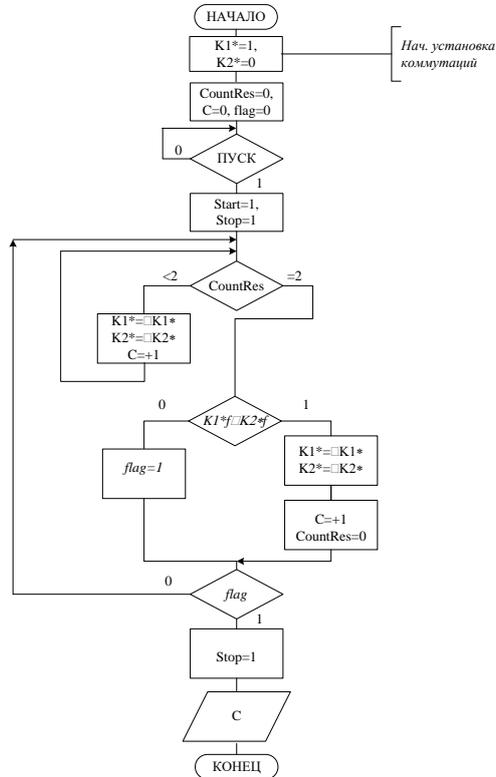


Рис. 9. Алгоритм управления коммутациями

Если на 2-х подряд чередующихся коммутациях $K1^*$ и $K2^*$ не было ни одной микрооперации обмена ($K1^* \vee K2^*$), тогда устанавливается $flag=1$, далее формируется $Stop=1$. Значение $Stop=1$ завершает работу алгоритма «сравнение-обмен». Алгоритм управления коммутациями формирует количество шагов сортировки в переменной S .

Итоговая структурная схема СУС представлена на рис. 10.

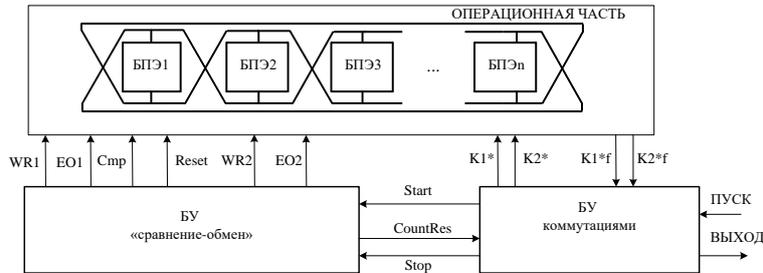


Рис. 10. Итоговая структурная схема СУС

Структурная схема СУС (рис. 10) показывает, что операционная часть устройства состоит из n БПЭ, из которых образуется $n/2$ пар соседних блоков, что на аппаратном уровне позволяет реализовать одну из коммутаций параллельно. Граф вычислительного процесса представляется итерационным переключением двух коммутаций до тех пор, пока на двух подряд коммутациях выполняется хотя бы одна микрооперация обмена.

Моделирование работы. Непосредственное отображение информационного графа задачи в граф операционной части устройства позволило достичь структурного соответствия и выполнять параллельную обработку пар элементов за 1 шаг работы устройства. При этом операция «сравнение-обмен» является аппаратно-ориентированной, она реализуется на буферных регистрах с перекрестными связями. Управление буферными регистрами осуществляется схем сравнения, значения «1» или «0» на выходах которых разрешают или блокируют обменную перезапись элементов соответственно.

Аппаратная поддержка базовых операций «сравнение-обмен» позволяет достичь линейной зависимости времени сортировки. Это подтверждается моделированием работы алгоритмов четно-нечетной сортировки по схемам (3), (4) и (5), (6) соответственно.

На рис. 11 и 12 приведены гистограммы и средние значения шагов работы алгоритмов четно-нечетной сортировки для массива размером $n=4$ элементов и мощности алфавитов $A_1=\{0,1\}$, $A_2=\{0,1,2,3\}$ на полный перебор всех комбинаций соответственно.

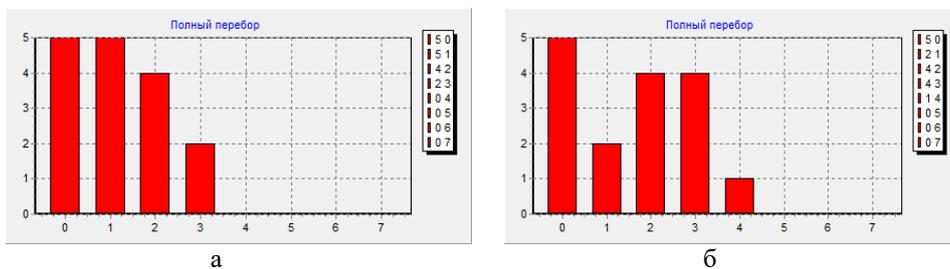


Рис. 11. Гистограммы четно-нечетной сортировки для кольцевой (а) и линейной (б) структуры массива M , $A_1=\{0,1\}$

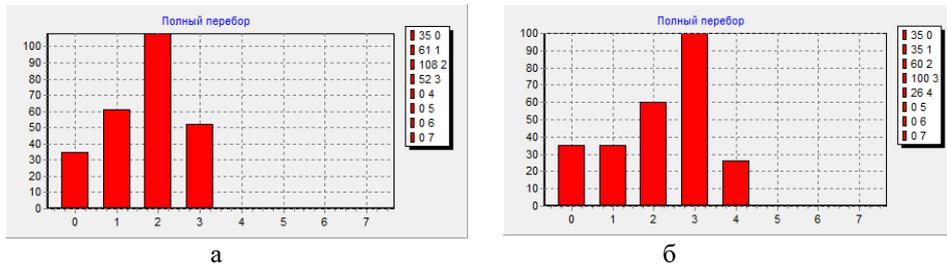


Рис. 12. Гистограммы четно-нечетной сортировки для кольцевой (а) и линейной (б) структуры массива M , $A_1=\{0,1,2,3\}$

Средние значения шагов сортировки по рис. 11 и 12 показаны в табл. 2.

Таблица 2

Среднее количество шагов сортировки

Размер массива n (структура данных)	Алфавит	
	{0,1}	{0,1,2,3}
4 (кольцо)	1,20	1,69
4 (строка)	1,63	2,20
8 (кольцо)	3,96	4,59
8 (строка)	4,59	5,23

Соотношения среднего времени для фиксированного размера и заданной мощности алфавита показывает преимущество модифицированной схемы сортировки на 15–18 % перед стандартной четно-нечетной сортировкой.

Выводы:

1. Для обеспечения декомпозиции производственной системы на обособленные подмножества создана модифицированная четно-нечетная схема коммутаций, состоящая из двух чередующихся вариантов коммутаций. Для оценки эффективности разработанной схемы коммутаций производственная система отождествлена с массивом, количество элементов которого соответствует количеству продукции в системе. Эта редукция позволяет заменить операцию пересечения слов на типовую операцию сравнения чисел и исследовать коммутационные процессы.

2. Сравнение классической и модифицированной схем коммутаций показало преимущество кольцевой организации массива, в которой пары из граничных элементов позволяют сократить количество инверсий при сортировке. Схема коммутаций позволяет выполнить перебор всех пар за время с линейной зависимостью от количества продукции в системе. Аппаратная поддержка операций «сравнение-обмен» и динамического образования элементов в пары позволила для модифицированной схемы четно-нечетных коммутаций получить среднее время, меньшее по классической четно-нечетной коммутации на 15-18% в зависимости от размера массива и мощности алфавита

3. В рамках теории структурно-процедурных вычислений разработано специализированное устройство сортировки, обеспечивающее декомпозицию производственной системы. Оригинальность устройства определяется наличием двух управляющих автоматов, отвечающих за параллельные сравнения-обмены и коммутации элементов и имеющих общую синхронизируемую переменную, что обеспечивает корректность работы устройства.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Бетелин В.Б., Кушницренко А.Г., Райко Г.О.* Проблемы обеспечения роста производительности отечественных суперЭВМ в период до 2020 года // Информационные технологии и вычислительные системы. – 2010. – № 3. – С. 15-18.
2. *Корнеев В.В.* Следующее поколение суперкомпьютеров // Открытые системы. – 2008. – № 8. – С. 14-19.
3. *Гузик В.Ф., Каляев И.А., Левин И.И.* Реконфигурируемые вычислительные системы: учеб. пособие / под общ. ред. И.А. Каляева. – Таганрог: Изд-во ЮФУ, 2016. – 472 с.
4. *Воеводин, В.В., Воеводин Вл.В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608 с.
5. *Огнев И.В., Борисов В.В., Сутула Н.А.* Ассоциативная память, среды, системы. – М.: Горячая линия – Телеком, 2016. – 416 с.
6. *King A.* Distributed Parallel Symbolic Execution. – B.S., Kansas State University, 2005. – P. 181-197.
7. *Каляев А.В., Левин И.И.* Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: Изд-во Янус-К, 2003. – 380 с.
8. *Фет Я.И.* Параллельные процессоры для управляющих систем. – М.: Энергоатомиздат, 1981. – 160 с.
9. *Ва Б.У., Лоурай М.Б., Ли Гоцзе.* ЭВМ для обработки символьной информации // ТИИЭР. – 1989. – Т. 77, № 4. – С. 5-40.
10. *Артамонов Д.С., Путря М.Г.* Метод оптимизации вычислительного процесса на реконфигурируемых вычислительных средах // Информационные технологии вычислительные системы. – 2010. – № 3. – С. 19-26.
11. *Путря Ф.М.* Архитектурные особенности процессоров с большим числом вычислительных ядер // Информационные технологии. – 2009. – № 4. – С. 2-7.
12. *Бурцев В.С.* Параллелизм вычислительных процессов и развитие архитектуры суперЭВМ. – М.: ТОРУС ПРЕСС, 2006. – 416 с.
13. *Титенко Е.А., Евсюков В.С.* Продукционные системы и теорема о конфликтных словах // Известия Тульского государственного университета. – 2006. – Вып. 15. – С. 92-98.
14. *Титенко Е.А.* Общая структурная схема реконфигурируемого мультипроцессора // Вестник Воронежского государственного технического университета. – 2011. – Т. 7, № 9. – С. 53-57.
15. *Титенко Е.А.* Метод и однородное вычислительное устройство k-приблизительного поиска вхождений по образцу // Вестник Воронежского государственного технического университета. – 2011. – Т. 7, № 7. – С. 70-78.
16. *Титенко Е.А.* Метод параллельного поиска по образцу и матричное устройство для его реализации // Информационные системы и технологии. – 2011. – № 4 (66). – С. 24-30.
17. *Wichert A.* Artificial intelligence and a universal quantum computer // AI Communications. – 2016. – Vol. 29, Issue 4. – P. 537-543.
18. *Titenko E.A., Degtyarev S.V.* Approximate search in the sample on the basis manber-wu method // Journal of Fundamental and Applied Sciences. – 2017. – Vol. 9, No. 2. – P. 914-918.
19. *Титенко Е.А., Зерин И.С.* Исчислительная система продукции и процедура распознавания конфликтов данных // Вестник компьютерных и информационных технологий. – 2012. – № 2. – С. 24-29.
20. *Люгер Дж.Ф.* Искусственный интеллект: стратегии и методы решения сложных проблем. – М.: Изд. дом «Вильямс». 2003. – 864 с.
21. *Титенко Е.А., Крипачев, Марухленко А.Л.* Коммутационная схема параллельных парных перестановок для специализированного продукционного устройства // Известия ЮФУ. Технические науки. – 2018. – № 8 (203). – С. 29-38.
22. *Макконелл Дж.* Основы современных алгоритмов. – 2-е изд. доп.: пер. с англ. / под ред. С.К. Ландо. – М.: Техносфера, 2004. – 386 с.
23. *Кнут Д.* Сортировка и поиск. Т. 3. – М.: Вильямс, 2000. – 500 с.
24. *Batcher К.Е.* Sorting Networks and their Applications // Proc. AFIPS Spring Joint Comput. Conf. – 1968. – Vol. 32. – P. 307314.
25. *Писаренко И.В., Алексеев К.Н., Мельников А.К.* Ресурснезависимое представление сортирующих сетей на языке программирования Set@1 // Вестник компьютерных и информационных технологий. – 2019. – № 9. – С. 3-10.

REFERENCES

1. *Betelin V.B., Kushnirenko A.G., Rayko G.O.* Problemy obespecheniya rosta proizvoditel'nosti otechestvennykh superEVM v period do 2020 goda [Problems of ensuring the productivity growth of domestic supercomputers in the period up to 2020], *Informatsionnye tekhnologii i vychislitel'nye sistemy* [Information technologies and computing systems], 2010, No. 3, pp. 15-18.
2. *Korneev V.V.* Sleduyushchee pokolenie superkomp'yutеров [The next generation of supercomputers], *Otkrytye sistemy* [Open systems], 2008, No. 8, pp. 14-19.
3. *Guzik V.F., Kalyaev I.A., Levin I.I.* Rekonfiguriruemye vychislitel'nye sistemy: ucheb. posobie [Reconfigurable computing systems: textbook], under the general editorship of I.A. Kalyaeva. Taganrog: Izd-vo YuFU, 2016, 472 p.
4. *Voevodin, V.V., Voevodin V.V.* Parallel'nye vychisleniya [Parallel computing]. St. Petersburg: BKhV-Peterburg, 2002, 608 p.
5. *Ognev I.V., Borisov V.V., Sutula N.A.* Assotsiativnaya pamyat', sredy, sistemy [Associative memory, environments, and systems]. Moscow: Goryachaya liniya – Telekom, 2016, 416 p.
6. *King A.* Distributed Parallel Symbolic Execution. B.S., Kansas State University, 2005, pp. 181-197.
7. *Kalyaev A.V., Levin I.I.* Modul'no-narashchivaemye mnogoprotsessornye sistemy so strukturno-protsedurnoy organizatsiyey vychisleniy [Modular-stackable multiprocessor systems with structural and procedural organization of calculations]. Moscow: Izd-vo Yanus-K, 2003, 380 p.
8. *Fet Ya.I.* Parallelnyye protsessory dlya upravlyayushchikh system [Parallel processors for control systems]. Moscow: Energoatomizdat, 1981, 160 p.
9. *Va B.U., Louray M.B., Li Gotsze.* EVM dlya obrabotki simvol'noy informatsii [Computer for processing symbolic information], *TIIEE* [Proceedings of the Institute of Electrical and Radio Electronics Engineers], 1989, Vol. 77, No. 4, pp. 5-40.
10. *Artamonov D.S., Putrya M.G.* Metod optimizatsii vychislitel'nogo protsessa na rekonfiguriruemyykh vychislitel'nykh sredakh [Method of optimization of the computational process on reconfigurable computing environments], *Informatsionnye tekhnologii vychislitel'nye sistemy* [Information Technologies computing systems], 2010, No. 3, pp. 19-26.
11. *Putrya F.M.* Arkhitekturnye osobennosti protsessorov s bol'shim chislom vychislitel'nykh yader [Architectural features of processors with a large number of computing cores], *Informatsionnye tekhnologii* [Information technologies], 2009, No. 4, pp. 2-7.
12. *Burtsev V.S.* Parallelizm vychislitel'nykh protsessorov i razvitiye arkhitektury superEVM [Parallelism of computing processes and the development of the architecture of supercomputers]. Moscow: TORUS PRESS, 2006, 416 p.
13. *Titenko E.A., Evsyukov V.S.* Produktsionnye sistemy i teorema o konfliktnykh slovakh [Productional systems and the theorem on conflict words], *Izvestiya Tul'skogo gosudarstvennogo universiteta* [Izvestiya Tulskego gosudarstvennogo universiteta], 2006, Issue 15, pp. 92-98.
14. *Titenko E.A.* Obshchaya strukturnaya skhema rekonfiguriruemogo mul'tiprotsessora [General block diagram of a reconfigurable multiprocessor], *Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta* [Bulletin of the Voronezh State Technical University], 2011, Vol. 7, No. 9, pp. 53-57.
15. *Titenko E.A.* Metod i odnorodnoe vychislitel'noe ustroystvo k-priblizitel'nogo poiska vkhozhdeniy po obraztsu [A method and a homogeneous computing device for k-approximate search for occurrences in a sample], *Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta* [Bulletin of the Voronezh State Technical University], 2011, Vol. 7, No. 7, pp. 70-78.
16. *Titenko E.A.* Metod parallelnogo poiska po obraztsu i matrichnoe ustroystvo dlya ego realizatsii [Method of parallel search by sample and matrix device for its implementation], *Informatsionnye sistemy i tekhnologii* [Information systems and technologies], 2011, No. 4 (66), pp. 24-30.
17. *Wichert A.* Artificial intelligence and a universal quantum computer, *AI Communications*, 2016, Vol. 29, Issue 4, pp. 537-543.
18. *Titenko E.A., Degtyarev S.V.* Approximate search in the sample on the basis manber-wu method, *Journal of Fundamental and Applied Sciences*, 2017, Vol. 9, No. 2, pp. 914-918.
19. *Titenko E.A., Zerin I.S.* Ischislitel'naya sistema produktsiy i protsedura raspoznavaniya konfliktov dannykh [Calculative system of products and the procedure for recognizing data conflicts], *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Bulletin of Computer and Information Technologies], 2012, No. 2, pp. 24-29.

20. *Lyuger Dzh.F.* Iskusstvennyy intellekt: strategii i metody resheniya slozhnykh problem [Artificial intelligence: strategies and methods for solving complex problems]. Moscow: Izd. dom «Vil'yams». 2003, 864 p.
21. *Titenko E.A., Kripachev, Marukhlenko A.L.* Kommutatsionnaya skhema parallel'nykh parnykh perestановок dlya spetsializirovannogo produktsionnogo ustroystva [Switching scheme of parallel pair permutations for a specialized production device], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2018, No. 8 (203), pp. 29-38.
22. *Makdonell Dzh.* Osnovy sovremennykh algoritmov [Fundamentals of modern algorithms]. 2nd ed.: transl. from engl., ed. by S.K. Lando. Moscow: Tekhnosfera, 2004, 386 p.
23. *Knut D.* Sortirovka i poisk [Sorting and searching]. Vol. 3. Moscow: Vil'yams, 2000, 500 p.
24. *Batcher K.E.* Sorting Networks and their Applications, *Proc. AFIPS Spring Joint Comput. Conf.*, 1968, Vol. 32, pp. 307314.
25. *Pisarenko I.V., Alekseev K.N., Mel'nikov A.K.* Resursonezavisimoe predstavlenie sortiruyushchikh setey na yazyke programmirovaniya Set@1 [Resource-independent representation of sorting networks in the programming language Set@1], *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Bulletin of Computer and Information Technologies], 2019, No. 9, pp. 3-10.

Статью рекомендовал к опубликованию д.ф.-м.н., профессор В.П. Добрица.

Титенко Евгений Анатольевич – Юго-Западный государственный университет; e-mail: johntit@mail.ru; г. Курск, ул. 50 лет Октября, 94; тел.: +79051588904; доцент; в.н.с.

Талдыкин Евгений Владимирович – e-mail: taldykin@mail.ru; аспирант.

Titenko Evgeny Anatolyevich – South-West State University; e-mail: johntit@mail.ru; 94, st. 50 years of October, Kursk, Russia; phone: +79051588904; associate professor, leading researcher.

Taldykin Evgeny Vladimirovich – e-mail: taldykin@mail.ru; post-graduate student.