

18. Natalya F. Noy and Deborah L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology, *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*, March 2001.
19. Gosudarstvennaya informatsionnaya sistema promyshlennosti [State industry information system]. Available at: <http://gisp.gov.ru> (accessed 31 October 2018).
20. Baza dannykh elektronnoy komponentnoy bazy [Electronic components database]. Available at: <http://ekb.mniirip.ru> (accessed 31 October 2018).
21. Biblioteka elektronnykh komponentov [Library of electronic components]. Available at: <https://optochip.org> (accessed 31 October 2018).

Статью рекомендовал к опубликованию д.т.н., профессор И.Е. Тарасов.

Горбунов Виктор Станиславович – Федеральное государственное унитарное предприятие «Научно-исследовательский институт «Квант»; e-mail: gorbunov@rdi-kvant.ru; 125438, г. Москва, 4-ый Лихачевский пер., 15; тел.: +74991531667; к.т.н.; доцент; зам. директора по научной работе.

Тупицын Антон Игоревич – e-mail: atupitsin@rdi-kvant.ru; тел.: +74997457121; зам. начальника отдела.

Gorbunov Victor Stanislavovich – Federal State Unitary Enterprise «Research and Development Institute «Kvant»; e-mail: gorbunov@rdi-kvant.ru; 15, 4-th Likhachevskii lane, Moscow, 125438, Russia; phone: +74991531667; cand. of eng. sc.; associate professor; deputy director for scientific work.

Tupitsyn Anton Igorevich – e-mail: atupitsin@rdi-kvant.ru; phone: +74997457121; deputy head of division.

УДК 681.3+004.32

DOI 10.23683/2311-3103-2018-8-29-38

Е.А. Титенко, А.В. Крипачев, А.Л. Марухленко

КОММУТАЦИОННАЯ СХЕМА ПАРАЛЛЕЛЬНЫХ ПАРНЫХ ПЕРЕСТАНОВОК ДЛЯ СПЕЦИАЛИЗИРОВАННОГО ПРОДУКЦИОННОГО УСТРОЙСТВА

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 взяты и развит известный алгоритм параллельного пузырька. Схема коммутации «параллельный пузырьек» 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 со схемой «параллельный пузырьек» на 15–18 %. В работе определена линейная зависимость времени сортировки с углом наклона меньше 1. Это позволяет использовать схему коммутации для

производственных 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 производственном устройстве для декомпозиции производственной системы на независимые подмножества продукции.

Коммутационная сеть; параллельная сортировка; реконфигурация связей.

E.A. Titenko, A.V. Kripachev, A.L. Marukhlenko

THE COMMUTATION SCHEME OF THE PARALLEL COMPANION TRANSFORMATIONS FOR SPECIALIZED COMPUTING DEVICE

The article shows 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 well-known algorithm of the parallel bubble is taken as the basis and developed. 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 developed switching scheme performance has been carried out on typical tasks of sorting and complete enumeration of pairs of elements. The time costs compared with the scheme "parallel bubble" are reduced by 15–18 %. A linear dependence of the sorting time with a slope angle less than 1 has been 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.

Commutation network; parallel sorting; reconfiguration of connections.

Введение. Массивы однородных дискретных вычислителей (ячейки, модули, блоки) представляет перспективный путь создания вычислительных систем и устройств (ВУ и ВС), совмещающих в операционной части функциональные блоки и коммутационную схему передачи промежуточных результатов [1–6]. Как правило, такие ВУ и ВС имеют специализированный характер, а массово значимыми в них являются операции быстрого преобразования Фурье, тасования, преобразование «бабочка», сортировки, приоритетного выбора, множественного сравнения данных, глобальной проверки и др. [1–7]. Такие операции имеют полиномиальную временную сложность. Аппаратные схемы и алгоритмы их реализации плохо поддаются масштабированию в силу наличия динамически изменяемых связей между коммутируемыми элементами данных [3].

С другой стороны, современные математические, алгоритмические программно-технические средства обработки информации в значительной степени ориентируются на проблемно-поисковые и вычислительные задачи, связанные с принятием решений в условиях априорной неопределенности данных. Обработка символьной информации (ОСИ) на основе производственных моделей является значимым классом таких задач, использующих схему вычислений «условие → действие» [8]. В отличие от традиционного управления потоком команд данная схема

основана на организации вычислений с управлением потоком данных [3]. Управление потоком данных основывается на параллельном исполнении такого подмножества команд (правил), для которых подготовлены все операнды. Для продукционных моделей готовность правил (продукций) к исполнению определяется положительным результатом операции поиска по образцу [9–12]. Распараллеливание потока данных для продукционных моделей имеет особенности, связанные с обеспечением полного или направленного перебора и сравнения элементов символьных данных на предмет их систематизации или перестроения [13].

В связи с этим задача создания коммутационной схемы в рамках продукционной парадигмы вычислений для параллельного выполнения множества операций над парами операндов является общезначимой задачей теоретической информатики и вычислительной техники.

Постановка задачи. Продукционная система (ПС) является типовым формальным аппаратом реализации символьных вычислений, т.е. преобразований над отношениями символов и групп символов. В качестве таких отношений выступают отношения вхождения, пересечения, объединения слов. Продукция – это математическое правило, формализующее данные операции поиска и модификации символьных данных. Аналитически продукция имеет вид

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

где O – слово-образец в рабочем алфавите A ; P – слово-модификатор в рабочем алфавите A ; \rightarrow – метасимвол.

Собственно ПС представляет набор из n продукций, объединенных на решение задачи путем итерационного преобразования (модификации) входных символьных данных до конечного результата

$$\left\{ \begin{array}{l} O_1 \rightarrow P_1 \\ \dots \\ O_i \rightarrow P_i \\ \dots \\ O_j \rightarrow P_j \\ \dots \\ O_n \rightarrow P_n \end{array} \right. \quad (2)$$

Вычислительный процесс в продукционной системе задается как последовательность сработавших подмножеств продукций, одновременно применимых к текущим данным. Активация продукций и их параллельное срабатывание в рамках схемы управления потоком данных осуществляется на основе операции поиска по образцу.

Как известно [8], операция поиска по образцу является наиболее трудоемкой. По мнению экспертов, она занимает до 80-90% времени продукционных вычислений. Операция поиска по образцу является глобальной в пространстве правил и структуре данных. Эта особенность накладывает структурные ограничения на распараллеливание вычислений при реализации данной операции единственным вычислительным устройством.

В связи с этим требуются средства декомпозиции исходной продукционной системы на подмножества с одновременно активируемыми продукциями, удовлетворяющие условию

$$T_{granul}(PS) \rightarrow \min,$$

где T_{granul} – функция времени декомпозиции (грануляции) продукционной системы (PS) на подмножества.

Схемы коммутации. Для ПС из n продукций предлагается способ циклического перебора пар операндов (слов-образцов) с целью формирования всевозможных пар и выполнения над такими парами слов символьных операций. В комбинаторике данный способ известен как сочетания из n элементов – C_n^2 . Число таких сочетаний имеет квадратичную сложность ($C_n^2 = \frac{n(n-1)}{2}$), что является достаточно критичным для ПС большого размера.

В связи с этим необходим алгоритм и его реализация в виде параллельных коммутационных схем генерации множества пар операндов, при этом алгоритм должен иметь линейную сложность. В качестве выполняемой операции между парой операндов определена операция пересечения слов, широко описанная в научной литературе [3].

Из теории и практики преобразования структур данных [6] известен алгоритм параллельного пузырька, генерирующий на каждом шаге $n/2$ ($n/2-1$) пар элементов из исходного массива и реализующий над ними операцию сравнения-обмена.

Графически работа алгоритма описывается схемой коммутации (рис. 1) на примере задачи сортировки при $n=16=2^4$ с исходным массивом $M=\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16\}^T$.

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

Рис. 1. Схема коммутации «параллельный пузырек»

Схема коммутации «параллельный пузырек» (соответственно алгоритм) состоит из двух чередующихся вариантов коммутации элементов в пары. Коммутация $K1$ основана на локальном объединении в пары элементов массива, имеющих смежные индексы, начиная с начального элемента, т.е. пар $(i, i+1)$, где $i=1-n-1$. Коммутация $K2$ также основана на локальном объединении в пары элементов массива, имеющих смежные индексы, начиная с второго элемента, т.е. пар $(j, j+1)$, где $j=2-n$. Вариант коммутации считается текущим, если хотя бы одна пара элементов среди $n/2$ ($n/2-1$) имеет положительный результат операции сравнения-обмена. Алгоритм не имеет явно заданной команды завершения работы. Применение схемы управления потоком данных для параллельного пузырька означает завершение работы при отсутствии хотя бы одного обмена элементов в паре по обоим вариантам коммутации. Это означает, что варианты коммутаций $K1, K2$ не изменяют состав массива M .

Схема коммутации «параллельный пузырек» имеет линейную сложность вида $T(n)=an + b$, где $a \geq 1$. Вышеописанная закономерность изменения индексов пар элементов в вариантах коммутаций $K1, K2$, обеспечивает перебор необходимых сочетаний элементов массива. Тем не менее объединения в пары смежных элементов массива является сдерживающим ограничением алгоритма «параллельный пузырек».

Для снятия данного ограничения разработаны оригинальные схемы коммутации и параллельного анализа на пересечение пар слов.

Схема коммутации обеспечивает циклическое чередование двух вариантов коммутации – $K1$ и $K2$. Первый вариант коммутации $K1$ аналогичен одному из вариантов коммутации в схеме «параллельный пузырьек». Второй вариант коммутации $K2$ основан на распределенном объединении в пару слов-образцов, имеющих индексы с некоторым шагом Δ . Вариант коммутации $K2$ строится так, чтобы первые элементы пар из коммутации $K1$ входили в новые пары без пропусков. Это условие обеспечивается заданием $\Delta=4$, что позволяет перераспределить элементы пар от коммутации $K1$ в новые пары по коммутации $K2$.

Для $n=2^w$ ($w \in \mathbf{N}$) ниже приведены примеры коммутируемых пар слов по их индексам:

- 1) вариант коммутации $K1$
(1,2), (3,4), (5,6), (7,8), (9,10)(n-1,n);
- 2) вариант коммутации $K2$
(1,4), (5,8), (9,12), ... (n-3,n),
(3,6) (7,10).... (n-(2+4),(n-2)).

Как принято для средств, управляемых потоком данных, условие останова – завершение парных перестановок по обеим вариантам коммутации.

В качестве первой тестовой задачи рассматривается задача сортировки массива M . Пусть исходный массив M размером $n=16=2^4$ по-прежнему имеет вид $M=\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16\}^T$ и в качестве выполняемой операции – сравнение и обмен двух чисел по отношению «больше».

Для предотвращения заикливания процесса сортировки на коммутации $K1$ (кроме нулевого шага) реализуется операция «меньше», что обеспечивает необходимое разнообразие распределения элементов в структуре массива. Коммутация $K2$ обеспечивает целевое продвижение элементов массива в необходимые позиции конечного массива. Тогда чередование коммутаций $K1$, $K2$ обеспечивает достижение результата, осуществляя на каждом шаге $n/2$ или $n/2-1$ сравнений и обменов (рис. 2).

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

Рис. 2. Сортировка локальной и распределенной схемами коммутаций

Анализ перемещений элементов в процессе параллельных парных обменов показывает, что траектория элементов содержит как целевые, так и возвратные перемещения в требуемую позицию. Возвратные перемещения приводят к непродуктивным затратам времени всей схемы коммутации. Они выполняются на шагах варианта коммутации $K1$. Для исключения непродуктивных шагов работы схемы коммутации предлагается в схему коммутации «параллельный пузырьек» ввести коммутацию-ускоритель, основанную на распределенном объединении в пару

слов-образцов, имеющих индексы с шагом $\Delta=4$. Предлагаемая схема коммутации будет содержать на нечетных шагах коммутацию с локальным объединением элементов в пары. На четных шагах будет выполняться коммутация-ускоритель с распределенным объединением элементов в пары (предлагаемый вариант коммутации K2).

В зависимости от выбора первого варианта коммутации (с нечетного или четного начального индекса массива) рассматриваются варианты схем коммутаций с локальным и распределенным объединением элементов пары. Схемы коммутации также основаны на чередовании на нечетных и четных шагах вариантов коммутации с локальным или распределенным объединением элементов пары, что позволяет исключить возвратные перемещения элементов по длине массива. Эта конструктивная особенность приводит к линейной зависимости времени генерации сортировки с углом наклона прямой меньше 1. Фактически это означает, что при длине массива в n элементов число шагов сортировки равно m , где $m < n$.

Первая схема коммутации с локальным и распределенным объединением элементов в пары управляет индексами по следующим законам:

- 1) вариант коммутации K1 (с локальным объединением элементов)
(2,3), (4,5), (6,7), (n-2,n-1);
- 2) вариант коммутации K2 (с распределенным объединением элементов)
(1,4), (5,8), (9,12), ... (n-3,n),
(3,6) (7,10).... ((n-5),(n-2)), (2,n-1).

Пример работы первой схемы коммутации с локальным и распределенным объединением элементов в пары приведен на рис.3. Вариант коммутации K2 обеспечивает согласованное перемещение элементов «прыжками» в соответствующие целевые позиции сортируемого массива, при этом над элементами массива совершаются направленные перемещения в необходимые позиции массива, ведущие в конечные (целевые) позиции.

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

Рис. 3. Работа первой схемы коммутации с локальным и распределенным объединением элементов в пары

Вторая схема коммутации с локальным и распределенным объединением элементов в пары управляет индексами по следующим законам:

- 1) вариант коммутации K1 (с локальным объединением элементов)
(1,2), (3,4), (5,6), (7,8), (9,10)(n-1,n);
- 2) вариант коммутации K2 (с распределенным объединением элементов)
(2,5), (6,9), (10,13), ... , ((n-6),(n-3))
(4,7), (8,11) ((n-4),(n-1)), (3,n).

Пример работы второй схемы коммутации с локальным и распределенным объединением элементов в пары приведен на рис.4. Вариант коммутации K2 обеспечивает согласованное перемещение элементов «прыжками» в соответствующие целевые позиции сортируемого массива, при этом над элементами массива совершаются направленные перемещения в необходимые позиции массива, ведущие в конечные (целевые) позиции. В отличие от первой схемы коммутации после шага 8 получен массив, характеризуемый особым свойством регулярности и однородности пар элементов, начиная с начального индекса. Это свойство показывает, что все элементы пар упорядочены между собой с общим отношением (в данном случае – отношение «больше»). Далее корректными могут быть только варианты коммутации с локальным объединением элементов, аналогичным алгоритму «параллельный пузырьек». Последующая сортировка по алгоритму «параллельный пузырьек» демонстрируется на шагах 9,10,11.

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

Рис. 4. Работа второй схемы коммутации с локальным и распределенным объединением элементов в пары

В качестве второй тестовой задачи рассматривается задача реализации полного перебора пар элементов массива. Задача также решается на основе предлагаемой коммутационной схемы, но результат фиксируется в виде правой треугольной части матрицы, где заполненными должны быть все ячейки. Индекс (k_1, k_2) заполненной ячейки соответствует образованию пары из элементов массива k_1 и k_2 .

Моделирование образования пар элементов массива показано на рис. 5. Оно подтвердило корректность работы всей коммутационной схемы в целом.

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

Рис. 5. Результат перебора пар элементов массива

Выводы. В работе достигнута цель по сокращению временных затрат на обеспечение параллельных производственных вычислений для обработки символьных данных. Для обеспечения декомпозиции производственной системы на независимые подмножества продукции требуется реализовать полный перебор пар слов-образцов на предмет выявления структурных зависимостей между продуктами каждого подмножества. Разработаны две оригинальные схемы коммутации, соединяющие в пары элементы таким образом, чтобы выполнялись как смежные, так и распределенные объединения элементов в пары. Чередование вариантов коммутаций и параллельная обработка пар в каждом варианте позволили получить линейную зависимость времени полного перебора образцов исходной производственной системы, при этом установлено, что угол наклона линейной зависимости меньше 1. Эта особенность позволяет использовать полученные схемы в специализированных устройствах обеспечения параллельных производственных вычислений [14–20]. В отличие от ранее существующих схем параллельной генерации сочетаний элементов сформировано логическое условие, предотвращения заикливания перебора пар.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Каляев А.В., Левин И.И.* Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: Изд-во Янус-К, 2003. – 380 с.
2. *Левин И.И., Дордопуло А.И., Мельников А.К., Гудков В.А.* Методы и средства архитектурно-независимого программирования высокопроизводительных вычислительных систем // Десятая Всероссийская мультikonференция по проблемам управления (МКПУ-2017). – Ростов-на-Дону – Таганрог: Изд-во ЮФУ, 2017. – С. 145-147.
3. *Бурцев В.С.* Параллелизм вычислительных процессов и развитие архитектуры супер-ЭВМ. – М.: ТОРУС ПРЕСС, 2006. – 416 с.
4. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608 с.
5. *Кравец О.Я., Подвальный Е.С., Титов В.С., Ястребов А.С.* Архитектура вычислительных систем с элементами конвейерной обработки: учеб. пособие. – СПб: Политехника, 2009. – 152 с.
6. *Сергеев С.Л.* Архитектуры вычислительных систем: учебник. – СПб.: БХВ-Петербург, 2010. – 231 с.
7. *Фет Я.И.* Параллельные процессоры для управляющих систем. – М.: Энерго-атомиздат, 1981. – 160 с.
8. *Ва Б.У., Лоурай М.Б., Гоцзе Ли.* ЭВМ для обработки символьной информации // ТИИЭР. – 1989. – Т. 77, № 4. – С. 5-40.
9. *Титенко Е.А.* Общая структурная схема реконфигурируемого мультипроцессора // Вестник Воронежского государственного технического университета. – 2011. – Т. 7, № 9. – С. 53-57.
10. *Довгаль В.М., Титов В.С., Титенко Е.А.* Стратегии быстрых символьных вычислений для исчислительной производственной системы // Известия высших учебных заведений. Приборостроение. – 2008. – Т. 51, № 2. – С. 44-47.
11. *Титенко Е.А., Евсюков В.С.* Производственные системы и теорема о конфликтных словах // Известия Тульского государственного университета. Серия технологическая системотехника. – 2006. – Вып. 15. – С. 92-98.
12. *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.
13. *Титенко Е.А.* Метод и однородное вычислительное устройство k-приближительного поиска вхождений по образцу // Вестник Воронежского государственного технического университета. – 2011. – Т. 7, № 7. – С. 70-78.
14. *Casbeer D.W.* Forest fire monitoring with multiple small UAVs // Proceedings of the 2005 American Control Conference. – 2005. – P. 3530-3535.

15. *Spry S.C., Girard A.R., Hedrick J.K.* Convoy Protection using Multiple Unmanned Aerial Vehicles: Organization and Coordination // Proc. of the 24th American Control Conference, Portland, OR., June 2005.
16. *Tonetti S., Hehn M., Lupashin S., D'Andrea R.* Distributed control of antenna array with formation of UAVs // In World Congress. – 2011, August. – Vol. 18, No. 1. – P. 7848-7853.
17. *Chung J.* Cooperative Control of UAVs Using a Single Master Subsystem for Multi-task Multi-target Operations // Advances in Intelligent Systems and Computing. – 2015. – Vol. 345. – P. 193-212.
18. *Корнеев В.В., Киселев А.В.* Современные микропроцессоры. – СПб.: BHV-СПб. 2003. – 448 с.
19. *King A.* Distributed Parallel Symbolic Execution. – B.S., Kansas State University, 2005.
20. *Hwang K., Briggs F.A.* Computer Architecture and Parallel Processing, 1984. – P. 32-40.

REFERENCES

1. *Kalyaev A.V., Levin I.I.* Modul'no-narashchivaemye mnogoprotsessornye sistemy so strukturno-protsedurnoy organizatsiyey vychisleniy [Modular-scalable multiprocessor system with structural-procedural organization of computing]. Moscow: Izd-vo YAnus-K, 2003, 380 p.
2. *Levin I.I., Dordopulo A.I., Mel'nikov A.K., Gudkov V.A.* Metody i sredstva arkhitekturno-nezavisimogo programmirovaniya vysokoproizvoditel'nykh vychislitel'nykh sistem [Methods and means of architecture-independent programming of high-performance computing systems], Desyataya Vserossiyskaya mul'tikonferentsiya po problemam upravleniya (MKPU-2017) [Tenth all-Russian multiconference on control problems (ICPD-2017)]. Rostov-on-Don – Taganrog: Izd-vo YuFU, 2017, pp. 145-147.
3. *Burtsev V.S.* Parallelizm vychislitel'nykh protsessov i razvitie arkhitektury superEVM [Parallelism of computing processes and development of supercomputer architecture]. Moscow: TORUS PRESS, 2006, 416 p.
4. *Voevodin V.V., Voevodin V.V.* Parallelnye vychisleniya [Parallel computation]. Saint Petersburg: BKHV-Peterburg, 2002, 608 p.
5. *Kravets O.Ya., Podval'nyy E.S., Titov V.S., Yastrebov A.S.* Arkhitektura vychislitel'nykh sistem s elementami konveyernoy obrabotki: ucheb. Posobie [Architecture of computer systems with elements of conveyor processing: tutorial]. Saint Petersburg: Politekhnik, 2009, 152 p.
6. *Sergeev S.L.* Arkhitektury vychislitel'nykh sistem: uchebnik [Architectures of computing systems: textbook]. Saint Petersburg:, 2010, 231 p.
7. *Fet Ya.I.* Parallelnye protsessory dlya upravlyayushchikh system [Parallel processors for control systems]. Moscow: Energoatomizdat, 1981, 160 p.
8. *Va B.U. Louray M.B., Gotsze Li.* EVM dlya obrabotki simvol'noy informatsii [Computer for processing symbolic information], *TIIEP* [Proceedings of the Institute of electrical and radio electronics engineers], 1989, Vol. 77, No. 4, pp. 5-40.
9. *Titenko E.A.* Obshchaya strukturnaya skhema rekonfiguriruемого mul'tiprotsessora [General block diagram of reconfigurable multiprocessor], *Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta* [Bulletin of Voronezh state technical University], 2011, Vol. 7, No. 9, pp. 53-57.
10. *Dovgal' V.M., Titov V.S., Titenko E.A.* Strategii bystrykh simvol'nykh vychisleniy dlya ischislitel'noy produktsionnoy sistemy [Strategies of fast symbolic calculations for the calculus production system], *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie* [Journal of Instrument Engineering], 2008, Vol. 51, No. 2, pp. 44-47.
11. *Titenko E.A., Evsyukov V.S.* Produktsionnye sistemy i teorema o konfliktnykh slovakh [Production systems and the theorem on conflict words], *Izvestiya Tul'skogo gosudarstvennogo universitet. Seriya tekhnologicheskaya sistemotekhnika* [News of Tula state University. A series of technological systems engineering], 2006, Issue 15, pp. 92-98.
12. *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.
13. *Titenko E.A., Evsyukov V.S.* Produktsionnye sistemy i teorema o konfliktnykh slovakh [The method and homogeneous computing device of k-approximate search of occurrences on a sample], *Izvestiya Tul'skogo gosudarstvennogo universitet. Seriya tekhnologicheskaya sistemotekhnika* [Bulletin of Voronezh state technical University], 2006, Issue 15, pp. 92-98.

14. Casbeer D.W. Forest fire monitoring with multiple small UAVs, *Proceedings of the 2005 American Control Conference*, 2005, pp. 3530-3535.
15. Spry S.C., Girard A.R., Hedrick J.K. Convoy Protection using Multiple Unmanned Aerial Vehicles: Organization and Coordination, *Proc. of the 24th American Control Conference, Portland, OR., June 2005*.
16. Tonetti S., Hehn M., Lupashin S., D'Andrea R. Distributed control of antenna array with formation of UAVs, *In World Congress*, 2011, August, Vol. 18, No. 1, pp. 7848-7853.
17. Chung J. Cooperative Control of UAVs Using a Single Master Subsystem for Multi-task Multi-target Operations, *Advances in Intelligent Systems and Computing*, 2015, Vol. 345, pp. 193-212.
18. Korneev V.V., Kiselev A.V. *Sovremennye mikroprotssory [Modern microprocessor]*. Saint Petersburg: BHV-SPb. 2003, 448 p.
19. King A. *Distributed Parallel Symbolic Execution*. B.S., Kansas State University, 2005.
20. Hwang K., Briggs F.A. *Computer Architecture and Parallel Processing*, 1984, pp. 32-40.

Статью рекомендовал к опубликованию д.т.н., профессор Д.Б. Борзов.

Титенко Евгений Анатольевич – Юго-Западный государственный университет; e-mail: johntit@mail.ru; 305040 г. Курск, ул. 50 лет Октября, 94; тел.: +79051588904; кафедра информационных систем и технологий; к.т.н.; доцент.

Крипачев Александр Владимирович – e-mail: xray-007@mail.ru; зав. отделом компьютерных технологий научной библиотеки.

Марухленко Анатолий Леонидович – e-mail: proxy33@mail.ru; кафедра информационной безопасности; к.т.н.; доцент.

Titenko Evgeny Anatolievich – South-West State University; e-mail: johntit@mail.ru; 94, 50 years of October street, Kursk, 305040, Russia; phone: +79051588904; the department of IT; cand. of eng. sc.; associate professor.

Kripachev Alexander Vladimirovich – e-mail: xray-007@mail.ru; head of the department of computer technologies of the scientific library.

Marukhlenko Anatoly Leonidovich – e-mail: proxy33@mail.ru; the department of information security; cand. of eng. sc.; associate professor.

УДК 004.272.44

DOI 10.23683/2311-3103-2018-8-38-47

О.В. Непомнящий, И.Н. Рыженко, А.И. Легалов

МЕТОД АРХИТЕКТУРНО-НЕЗАВИСИМОГО ВЫСОКОУРОВНЕВОГО СИНТЕЗА СБИС

Одним из актуальных направлений развития технологий проектирования сверхбольших интегральных схем и вычислительных систем на их основе является высокоуровневый синтез. При описании проекта на верхних уровнях закладываются концепции общесистемного взгляда на организацию всего процесса проектирования. Поэтому на первый план выходит развитие маршрутов и технологий, базирующихся на принципах высокоуровневого, архитектурно независимого проектирования, позволяющих осуществлять формирование комплексного подхода к организации всех фаз создания проекта. Требуется создание методов эффективной выработки архитектурных решений для однокристалльных систем параллельной обработки информационных потоков, не зависящих от конечной формы реализации. Необходима разработка инструментальных средств, обеспечивающих эффективный перенос архитектурно-независимого, высокоуровневого описания решаемых прикладных задач на целевую платформу. Авторами предложен новый метод синтеза проекта. Метод базируется на функционально-потоковой парадигме параллельных вычислений, это позволяет осуществлять архитектурно-независимую разработку алгоритмов функционирования СБИС. Предложена модель вычислений, использующая ряд промежуточных