

Реконфигурируемая вычислительная модульная система

С.С. Шевелев

ФГБОУ ВО «Юго-Западный государственный университет», Россия, 305040, Курск, ул. 50 лет Октября, 94

✉ e-mail: schewelew@mail.ru

Резюме

Цель исследования. Реконфигурируемая вычислительная система имеет вычислительные комплексы и специализированные ЭВМ, которые используются при решении задач векторной и матричной алгебры, распознавания образов. Различают матричные и ассоциативные системы, нейронные сети. Матричные вычислительные системы состоят из множества процессорных элементов, соединенных через коммутационное устройство с многомодульной памятью. Они предназначены для решения задач над векторами, матрицами и массивами данных. Ассоциативные системы содержат большое число операционных устройств, способных одновременно вести обработку нескольких потоков данных. Нейронные сети и нейрокомпьютеры имеют высокую производительность при решении задач экспертных систем, распознавания образов за счет параллельной обработки нейросети.

Методы. Построен информационный граф вычислительного процесса модульной системы с перестраиваемой структурой. Разработаны структурные и функциональные схемы, алгоритмы, реализующие построение специализированных модулей для выполнения арифметических и логических операций, поисковых операций и функций замены вхождений в обрабатываемых словах. Разработано программное обеспечение моделирования работы арифметико-символьного процессора, специализированных вычислительных модулей, систем коммутаций.

Результаты. Разработана структурная схема реконфигурируемой вычислительной модульной системы, которая состоит из совместимых функциональных модулей, она способна к статической и динамической реконфигурации, имеет параллельную структуру соединения процессора и вычислительных модулей за счет использования интерфейсных каналов. Система состоит из арифметико-символьного процессора, специализированных вычислительных модулей и систем коммутаций, выполняет специфические задачи символьной обработки информации, арифметические и логические операции.

Заключение. Системы с перестраиваемой структурой представляют собой высокопроизводительные и высоконадежные вычислительные системы, которые состоят из объединенных процессоров в многомашинные и многопроцессорные комплексы. Перестраиваемость структуры обеспечивает высокую производительность системы за счет ее адаптации к вычислительным процессам и составу обрабатываемых задач.

Ключевые слова: информационный граф; параллельно-конвейерная структура; модульная система; специализированные вычислительные блоки.

Конфликт интересов: Автор декларирует отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Для цитирования: Шевелев С.С. Реконфигурируемая вычислительная модульная система // Известия Юго-Западного государственного университета. 2019. Т. 23, № 2. С. 137-152. DOI: 10.21869/2223-1560-2019-23-2-137-152.

Reconfigurable Modular Computing System

Sergey S. Schevelev

Southwest State University, 94, 50 Let Oktyabrya str., Kursk, 305040, Russian Federation

✉ e-mail: schewelev@mail.ru

Abstract

Purpose of research. A reconfigurable computer system consists of a computing system and special-purpose computers that are used to solve the tasks of vector and matrix algebra, pattern recognition. There are distinctions between matrix and associative systems, neural networks. Matrix computing systems comprise a set of processor units connected through a switching device with multi-module memory. They are designed to solve vector, matrix and data array problems. Associative systems contain a large number of operating devices that can simultaneously process multiple data streams. Neural networks and neurocomputers have high performance when solving problems of expert systems, pattern recognition due to parallel processing of a neural network.

Methods. An information graph of the computational process of a reconfigurable modular system was plotted. Structural and functional schemes, algorithms that implement the construction of specialized modules for performing arithmetic and logical operations, search operations and functions for replacing occurrences in processed words were developed. Software for modelling the operation of the arithmetic-symbol processor, specialized computing modules, and switching systems was developed.

Results. A block diagram of a reconfigurable computing modular system was developed. The system consists of compatible functional modules and is capable of static and dynamic reconfiguration, has a parallel connection structure of the processor and computing modules through the use of interface channels. It consists of an arithmetic-symbol processor, specialized computing modules and switching systems; it performs specific tasks of symbolic information processing, arithmetic and logical operations.

Conclusion. Systems with a reconfigurable structure are high-performance and highly reliable computing systems that consist of integrated processors in multi-machine and multiprocessor systems. Reconfigurability of the structure provides high system performance due to its adaptation to computational processes and the composition of the processed tasks.

Keywords: block diagram; parallel-pipeline structure; modular system; special-purpose computing units.

Conflict of interest. The Author declare the absence of obvious and potential conflicts of interest related to the publication of this article.

For citation: Schevelev S. S. Reconfigurable Modular Computing System. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta* = *Proceedings of the Southwest State University*. 2019, 23(2): 137-152 (In Russ.). DOI: 10.21869/2223-1560-2019-23-2-137-152.

Введение

Идея концепции построения реконфигурируемых вычислительных систем заключается в аппаратной реализации всех операций, предписанных вершинами информационного графа за-

дачи, всех каналов передачи данных между вершинами, соответствующими дугам графа, и всех информационных каналов, соответствующих входным и выходным вершинам. Задача, определенная информационным графом, будет

выполнена максимально быстро, поскольку обеспечивается максимально возможное распараллеливание вычислений. Информационный граф большой задачи сегментируется на фрагменты – непересекающиеся базовые подграфы, физически реализуемые в аппаратуре реконфигурируемых вычислительных систем.

Основными вычислительными блоками в реконфигурируемой вычислительной системе являются макропроцессоры, которые позволяют реализовать операции, предписанные вершинами информационного графа. Макропроцессор представляет собой некоторый набор элементарных процессоров, объединяемых в единый программно-неделимый вычислительный ресурс с помощью локального пространственного коммутатора. Для реализации информационного графа решаемой задачи макропроцессоры должны иметь возможность соединения в вычислительные параллельно-конвейерные структуры с помощью системного коммутатора, который обеспечивает различные варианты соединения макропроцессоров друг с другом. В состав системы входит распределенная память, обеспечивающая возможности параллельной выдачи массивов входных данных на входы макропроцессоров и записи результатов вычислений с их выходов [1].

Материалы и методы решения задачи

Граф $G(S, F)$ (рис. 1) называется информационным графом вычисли-

тельной открытой развиваемой асинхронной модульной системы, или графом алгоритма решения задачи. Граф $G(S, F)$ содержит множество вершин $s_i \in S$, каждой из которых приписана некоторая операция O_i , принадлежащая множеству допустимых операций O . Дуги $f(s_i, s_{i+1}) \in F$ определяют последовательность выполнения операций, приписанных вершинам графа, причем если две вершины s_i и s_{i+1} соединены дугой $f(s_i, s_{i+1})$, то это означает, что результат операции O_i является входным данным для операции O_{i+1} . Граф $G(S, F)$ имеет также множество входных дуг $f(s_{0i}, s_i)$, определяющих блоком входных данных, а также выходных дуг $f(s_i, s_k)$, определяющих хранение результатов ее решения.

Множество вершин S определяется множеством процессоров вычислительной открытой асинхронной системы. Множество дуг f представляет собой множество каналов коммуникаций между процессорами. Входные и выходные дуги графа $G(S, F)$ определяются каналами связи с источником входных и приемником выходных данных, в частности, с памятью системы. Операции O_i из множества допустимых операций O реализуются в процессорах вычислительной системы в форме последовательных процедур обработки данных.

Вычислительную открытую развиваемую асинхронную модульную систему можно представить как иерархию универсальной ЭВМ, систему специализированных вычислительных модулей и гибкую подсистему интерфейсных кана-

специализированные модули: СПЕЦ МОД1 выполняет операции по символьной обработке, СПЕЦ МОД2 осуществляет сортировку слов входного текста, СПЕЦ МОД3 решает задачу распределения ресурсов со многими параметрами между локализованными центрами, СПЕЦ МОД4 выполняет арифметическую операцию сложения чисел. СПЕЦ МОД5 выполняет сортировку двоичной информации.

Семейство высокопроизводительных многопроцессорных вычислительных систем с динамически перестраиваемой, программируемой архитектурой на основе реконфигурируемой элементной базы является конкурентоспособной альтернативой традиционной организации многопроцессорных систем, основанных на использовании микропроцессоров в качестве основного вычислительного элемента.

Вычислительная открытая развиваемая асинхронная модульная система сочетает в себе как последовательный, так и конвейерный способ обработки информации, поскольку входные данные обрабатываются одновременно по разным конвейерным цепочкам процессоров.

Основными вычислительными блоками в реконфигурируемой вычислительной системе являются макропроцессоры, которые позволяют реализовывать операции, предписанные вершинами информационного графа. Макро-

процессоры соединяются в вычислительные параллельно-конвейерные структуры с помощью системного коммутатора, который обеспечивает различные варианты соединения вычислительных модулей друг с другом. В состав системы входит распределенная память, обеспечивающая параллельную выдачу массивов входных данных на входы макропроцессоров и записи результатов вычислений с их выходов.

Для решения этой задачи можно использовать блок выполнения логических операций с последовательной архитектурой, основными элементами которого являются элементарные вычислители, выполняющие логические операции: $\{\&, \vee, \neg\}$, $\{\&, \neg\}$, $\{\vee, \neg\}$, $\{\&, \oplus, 1\}$, которые относятся к функционально полной системе булевых функций.

К множеству микроопераций $L = \{L_{12}, L, \dots, L_{1z}\}$, выполняемых блоком логических операций S_3 , относятся: инверсия $L_{12} \div g = \bigwedge_{i=1}^n \bar{a}_i$, конъюнкция $L_{12} \div g = \bigwedge_{i=1}^n (a_i b_i)$, дизъюнкция $L_{12} \div g = \bigvee_{i=1}^n (a_i b_i)$, исключающее ИЛИ $L_{12} \div g = \bigoplus_{i=1}^n (a_i b_i)$, $g_i \in \{0, 1\}$; $a_i \in \{0, 1\}$; $b_i \in \{0, 1\}$, эквивалентность $L_{12} \div g = \bigwedge_{i=1}^n \sim (a_i b_i)$, $g_i \in \{0, 1\}$; $a_i \in \{0, 1\}$; $b_i \in \{0, 1\}$.

На рис. 3 представлены: блок ввода данных БВД, система электронных ключей СиК, элементарные вычислители логических операций ЭлВ, блок хранения результатов БХР.

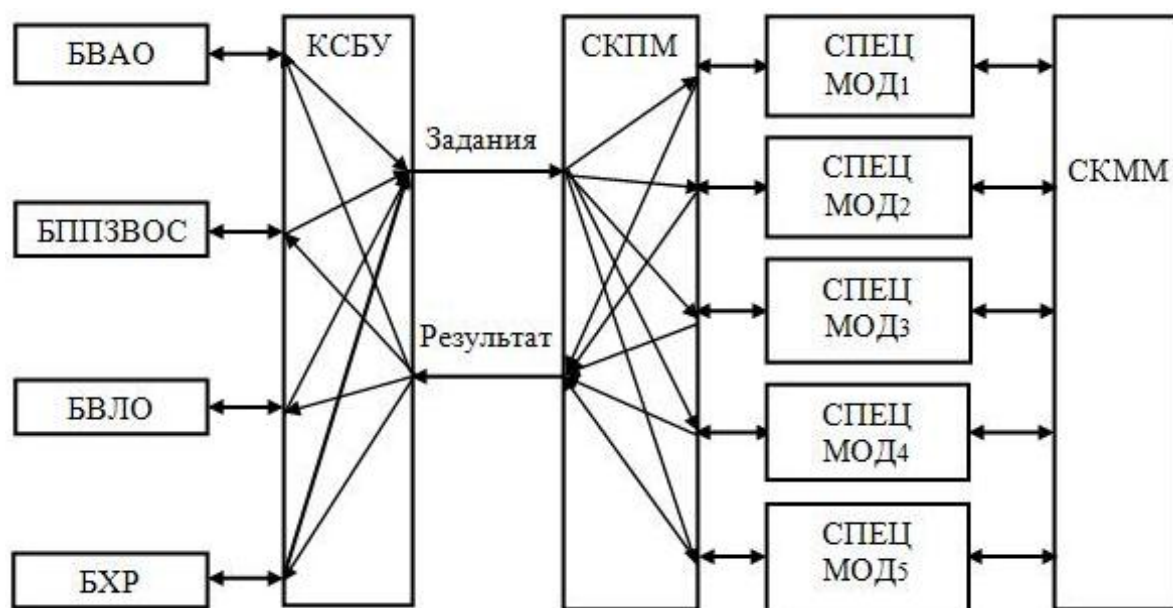


Рис. 2. Граф вычислительного процесса вычислительной открытой развиваемой асинхронной модульной системы ВОРАМС

Время результата $T_{\text{нос}}$ обработки N векторов входных данных определяется по формуле

$$T_{\text{нос}} = N * (t_{\text{БВД}} + t_{\text{Сик}} + t_{\text{ЭЛВ}} + t_{\text{БХР}}) = N \sum_{i=1}^M s(O_i) \cdot \tau, \quad (1)$$

где $t_{\text{БВД}}$, $t_{\text{Сик}}$, $t_{\text{ЭЛВ}}$, $t_{\text{БХР}}$ – время обработки данных блоками устройства; M – число вершин на графе $G(S, F)$; $s(O_i)$ – число тактов работы блока устройства при выполнении операции O_i , соответствующей вершине графа $G(S, F)$, τ – продолжительность такта.

Сокращение времени $T_{\text{нар}}$ обработки потока входных векторов является распараллеливанием процесса обработки. В этом случае параллельный способ подразумевает наличие K процессоров P_i ($i = 1, 2, \dots, K$), каждый из которых

может работать независимо от других процессоров. Каждый из процессоров P_i ($i = 1, 2, \dots, K$) запрограммирован на реализацию графа $G(S, F)$. Множество входных векторов Z_i ($i = 1, 2, \dots, N$) разбивается на (N/K) непересекающихся подмножеств, каждое из этих подмножеств входных векторов может быть обработано на процессоре P_i независимо, то есть параллельно с другими подмножествами.

Для решения этой задачи можно использовать блок выполнения математических операций в форматах с фиксированной и плавающей запятой с параллельной архитектурой (рис. 4), основными элементами которого являются математические вычислители, выполняющие арифметические и сдвиговые операции.

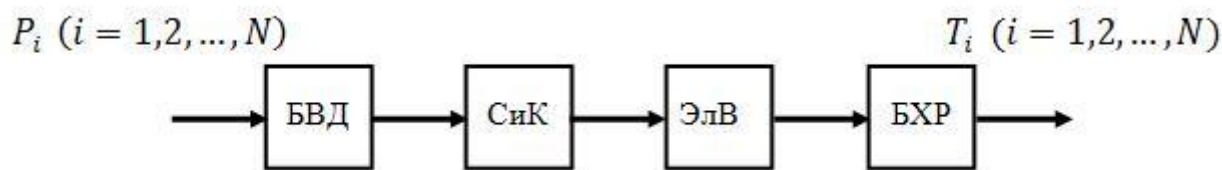


Рис. 3. Блоки выполнения логических операций с последовательной архитектурой

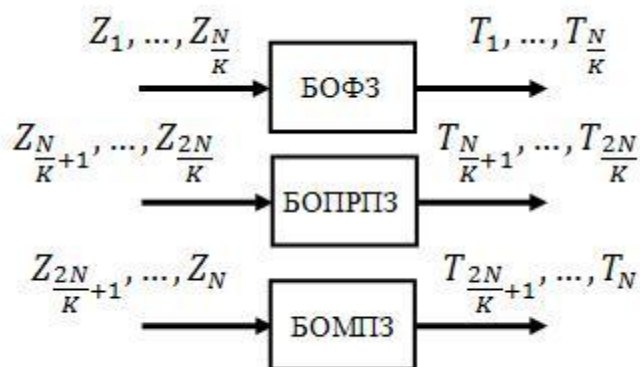


Рис. 4. Параллельный способ обработки векторов входных данных

К множеству микроопераций $R = \{R_{12}, R_{13}, \dots, R_{1k}\}$, выполняемых блоком арифметических операций S_1 , относятся: суммирования $R_{12} = \sum_{i=1}^n (a_i b_i)$, вычитания $R_{13} = \sum_{i=1}^n (a_i - b_i)$, произведения $R_{13} = \prod_{i=1}^n (a_i b_i)$, деления $R_{14} = \prod_{i=1}^n (a_i / b_i)$ в формате с фиксированной запятой и

$$\begin{aligned} R_{12} &= \sum_{i=1}^n (a_i 2^x b_i 2^m), \\ R_{13} &= \sum_{i=1}^n (a_i 2^x - b_i 2^m), \\ R_{13} &= \prod_{i=1}^n (a_i 2^x b_i 2^m), \\ R_{14} &= \prod_{i=1}^n (a_i 2^x / b_i 2^m) \end{aligned}$$

в формате с плавающей запятой.

В этом случае время получения результата $T_{пар}$, при обработке всего множества векторов входных данных, будет вычисляться по формуле

$$\begin{aligned} T_{пар} &= \left(\frac{N}{K}\right) \cdot (t_{БОФЗ} + t_{БОПРПЗ} + \\ t_{БОМПЗ}) &= \left(\frac{N}{K}\right) \sum_{i=1}^M s(O_i) \cdot \tau. \end{aligned} \quad (2)$$

Время обработки потока сокращается в K раз по сравнению с последовательной обработкой всех входных векторов.

К множеству микроопераций $C = \{C_{12}, C, \dots, C_{1t}\}$, выполняемых блоком параллельного поиска и замены в обрабатываемых словах S_2 относятся: левая конкатенация β и α цепочек $\alpha = d_1 d_2 \dots d_k$ и $\beta = b_1 b_2 \dots b_l$ символы одного этого же алфавита V , $d_i \in V$, $\forall i = 1, \dots, k$ и $b_j \in V$, $\forall j = 1, \dots, l$, равна $\gamma = b_1 b_2 \dots b_l d_1 d_2 \dots d_k$, правая конкатенация цепочек α и β принимает вид $\gamma = d_1 d_2 \dots d_k b_1 b_2 \dots b_l$ над тем же алфавитом V .

Блок выполняет операцию параллельного поиска вхождений в обрабатываемых словах. Требуется найти все вхождения $x = b_1 \dots b_m$ в слове

$w = a_1 \dots a_n$ в качестве подстроки. Определить все смещения s , для которых подстрока $w_s = a_{s+1} \dots a_{s+m}$ совпадает с $b_1 \dots b_m$. Для поиска вычисляется функция

$$F = (a_1 \sim b_1) \& (a_2 \sim b_2) \& \dots \& (a_m \sim b_m).$$

Если функция F равна нулю, то вхождение и фрагмент слова не совпадают. В этом случае формируется сдвиг символов слова влево на один разряд. Операция сравнения продолжается. Если F равно единице, то вхождение найдено в слове. Смещение s определяет позицию вхождения в слове, сдвиг символов слова осуществляется на m позиций влево [3].

В блоке выполняется операция замены вхождений в обрабатываемом слове, которая задается формулой подстановки. Применяя конечный упорядоченный набор формул подстановки к обрабатываемому слову w :

$$\begin{cases} a_{s+1} \rightarrow \beta_1 \\ \dots \\ a_{s+m} \rightarrow \beta_k, \end{cases} \quad (k \geq 1) \quad (3)$$

получаем новое слово $w' = a_1 \dots a_s \beta_1 \beta_2 \dots \beta_k a_{s+m+1} \dots a_n$.

Если это слово взять за исходное и применить те же процедуры, то получа-

ется новое слово w'' и так далее $w \rightarrow w' \rightarrow w'' \rightarrow \dots$.

Быстродействие конвейера при длительной работе будет значительно превышать быстродействие, которое достигается при последовательной обработке команд. Это увеличение будет тем больше, чем меньше длительность такта конвейера и чем больше количество выполненных за исследуемый период команд.

В блоке параллельного поиска и замены в обрабатываемых словах вычислительной системы можно использовать конвейерную обработку. Блок имеет 4 элементарных процессора P_i ($i = 1, 2, 3, 4$), соединенных в цепочку, как показано на рис. 5, каждый из элементарных процессоров P_i запрограммирован на реализацию соответствующего подграфа $G_i(S_i, F_i)$ ($i = 1, 2, 3, 4$) общего графа алгоритма $G(S, F)$. На рис. 5 представлены процессоры: памяти вхождений и обрабатываемых слов БПВС, сравнения и анализа вхождений в обрабатываемых словах БАВС, замены вхождений в обрабатываемых словах БЗАМ, хранения результатов подстановок БХРП.

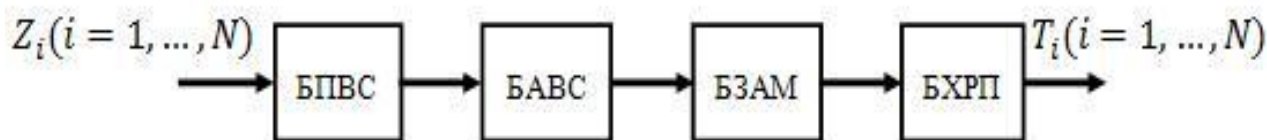


Рис. 5. Конвейерная организация работы блока параллельного поиска и замены в обрабатываемых словах

Длительность такта T_{max} будет равна длительности самого медленного этапа обработки – блока сравнения и анализа вхождений в обрабатываемых словах БАВС.

$$T = \max \{T_{БПВС}, T_{БАВС}, T_{БЗАМ}, T_{БХРП}\} = T_{БАВС}. \quad (4)$$

Время $T_{кон}$ при конвейерной обработке N команд будет вычисляться следующей формулой:

$$T_{кон} = P \cdot T_{БАВС} + (N - 1) \cdot T_{БАВС} = T_{БАВС} \cdot (P + (N - 1)), \quad (5)$$

где P – число элементарных процессоров, $T_{БАВС}$ – самый максимальный по времени этап обработки данных.

Реконфигурируемые вычислительные системы с динамически перестраиваемой архитектурой превосходят по технико-экономическим характеристикам многопроцессорные вычислительные системы с традиционной архитектурой в сегментах рынка суперЭВМ и рабочих станций, а также ускорителей к персональному компьютеру.

Сравнительный анализ времени и скорости конвейерной и последовательной архитектур на примере работы блока параллельного поиска и замены в обрабатываемых словах вычислительной системы.

Время и скорость при последовательной организации работы блока вычислительной системы $T_{пос}$ и $V_{пос}$ определяются по формулам:

$$T_{пос} = N \cdot (\tau_1 + \tau_{max} + \tau_3 + \tau_4),$$

$$V_{пос} = \frac{1}{(\sum_{i=1}^4 \tau_i)} = \frac{1}{(\tau_1 + \tau_{max} + \tau_3 + \tau_4)}. \quad (6)$$

Время $T_{кон}$ и скорость $V_{кон}$ конвейерной организации работы блока параллельного поиска и замены в обрабаты-

ваемых словах определяются из отношений:

$$T_{кон} = (N + 4 - 1) \cdot \tau_{max} = (N + 3) \cdot \tau_{max},$$

$$V_{кон} = (\tau_{max})^{-1} = \frac{1}{\tau_{max}}. \quad (7)$$

Отношение времени последовательной организации вычислительного процесса $T_{пос}$ к конвейерной $T_{кон}$ будет определяться как

$$\frac{T_{пос}}{T_{кон}} = \frac{\tau_1 + \tau_{max} + \tau_3 + \tau_4}{\tau_{max}} \approx \frac{4 \cdot \tau_{max}}{\tau_{max}} = 4. \quad (8)$$

Отношение скорости конвейерной $V_{кон}$ к последовательной $V_{пос}$ определится как

$$\frac{V_{кон}}{V_{пос}} = \frac{\tau_1 + \tau_{max} + \tau_3 + \tau_4}{\tau_{max}} \approx \frac{4 \cdot \tau_{max}}{\tau_{max}} = 4. \quad (9)$$

При большом значении числа входных векторов N конвейерная организация вычислительного процесса требует меньше времени, а скорость будет больше, чем последовательная более чем в 4 раза, где 4 – число элементарных процессоров в конвейерной цепочке блока. Эффективность архитектуры конвейерной организации вычислительного процесса по сравнению с последовательной приблизительно в N раз выше, где N – число процессоров в цепочке [4].

Время выполнения операции блоком параллельного поиска и замены в обрабатываемых словах конвейерной и последовательной архитектур приведены в таблице.

Сравнительный анализ времени конвейерной и последовательной архитектур. Изображенный график на рис.6 иллюстрирует разницу в эффективности архитектуры конвейерной организации вычислительного процесса по сравнению с последовательной.

Время определения поиска и замены в словах

Число входных векторов N	1	5	10	15	20
Время $T_{\text{пос}} \text{ нс}$	500	2500	5000	7500	10000
Время $T_{\text{кон}} \text{ нс}$	800	1600	2600	3600	4600

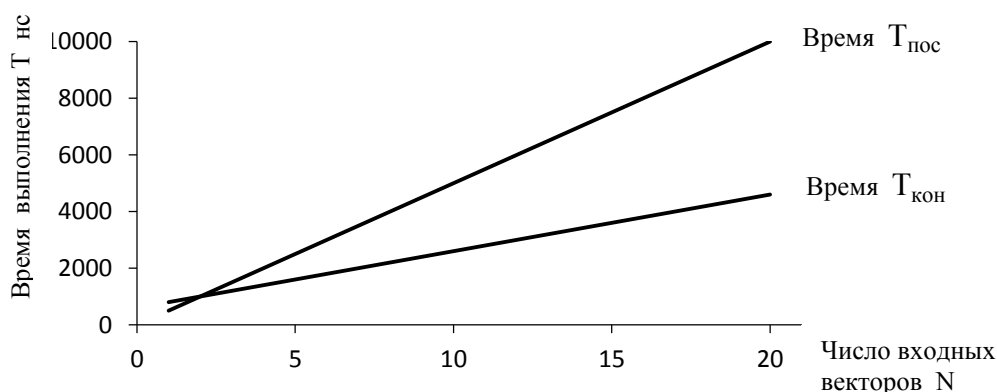


Рис. 6. График сравнения времени конвейерной и последовательной архитектур

Важным компонентом массово-параллельных компьютеров является коммуникационная среда. Это набор аппаратных и программных средств, обеспечивающих обмен сообщениями между процессорами. Она предназначена для достижения высоких скоростей передачи с малым временем задержки и при этом обеспечивает масштабируемую архитектуру, позволяющую строить системы, состоящие из множества блоков.

Система коммутации процессор–модуль обеспечивает передачу информации от арифметико-символьного процессора на входы специализированных вычислительных модулей системы в виде заданий, а также формирует интерфейсный канал процессор-модуль передачи информации с выходов специ-

ализированных вычислительных модулей системы в виде результата для хранения его в памяти арифметико-символьного процессора.

Система коммутации модуль–модуль обеспечивает передачу информации с выходов специализированных вычислительных модулей на входы других специализированных вычислительных модулей системы в виде заданий и результатов, а также формирует интерфейсный канал передачи информации с выходов специализированных вычислительных модулей системы в виде результатов для хранения их в памяти арифметико-символьного процессора [5].

Специализированный вычислительный модуль СпецВычМод1 выполняет операции поиска вхождений в обрабатываемых словах (рис. 7).

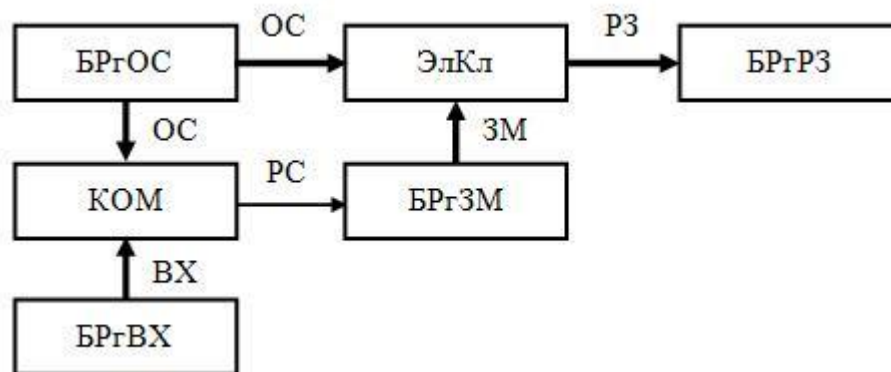


Рис. 7. Специализированный вычислительный модуль СпецВычМод1: БРгОС – блок регистра обрабатываемого слова; БРгВХ – блок регистра вхождений; БРгЗМ – блок регистра замены; КОМ – компаратор; БРгРЗ – блок регистр результата замены; ЭлКл – схема электронных ключей

Первый формат работы устройства определяет вхождения, которые имеют одинаковые части. Это означает, что предыдущие вхождения и последующие имеют общую часть, состоящую из одной буквы или цепочки символов. Вторым форматом работы системы характеризуется определением вхождений, не имеющих общих частей. В этом случае определяется адрес только через n сдвигов, где n – количество букв в регистре вхождения. Если необходимо произвести замену найденного вхождения на подстановку буквы или слова, заранее определенную и записанную в регистр подстановок. В этом случае, вначале осуществляется операция поиска вхождений без общих частей, затем производится операция замены найденного вхождения на подстановку. Процедуры сдвига возможны при помощи реверсивных регистров, которые осуществляют сдвиг информации как влево, так и вправо. В параллельной системе поиска и замены осуществляется процесс обработки информации сразу с не-

сколькими словами и несколькими вхождениями в параллельном формате. Для выполнения параллельной обработки в системе имеются n -блоков поиска и замены [7].

Специализированный вычислительный модуль СпецВычМод2 выполняет операцию сортировки слов с помощью ассоциативного запоминающего устройства (рис. 8). Поиск слов в памяти происходит по содержанию. В ассоциативном запоминающем устройстве для обработки данных используются последовательные алгоритмы, поиск осуществляется параллельно по словам и последовательно – по разрядам. Это позволяет проводить операции на равенство искомых слов, равных заданному признаку. В режиме поиска АЗУ считываются из памяти все слова, подчиняющиеся определенным условиям: равные, больше или меньше, поиск минимального, максимального значения, поиск величин, заключенных в заданном интервале, упорядоченная сортировка [7].

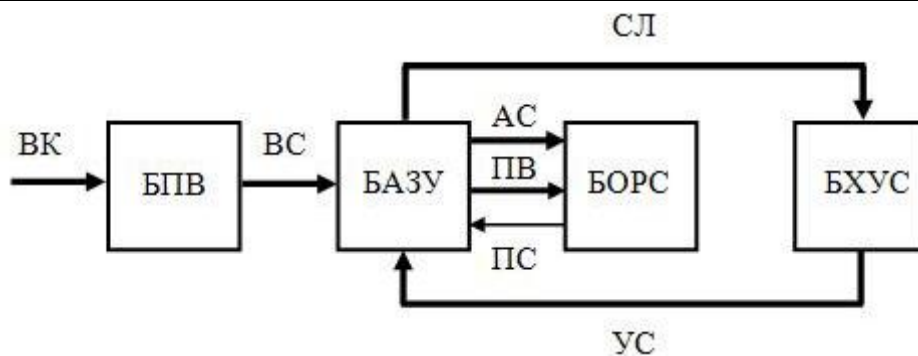


Рис. 8. Специализированный вычислительный модуль СпецВычМод2: БПВ – блок поиска вхождений; БАЗУ – блок ассоциативных запоминающих устройств; БОРС – блок обработки результатов сравнения; БХУС – блок хранения упорядоченных слов

Специализированный вычислительный модуль СпецВычМод3 формирует режимы управления материальными потоками или информацией, которые обеспечивают оптимальное сбалансированное функционирование системы взаимодействующих локализованных узлов по многим ресурсам и параметрам (рис. 9).

Локализованные центры системы служат для формирования избыточного ресурса, который необходимо распределить по ассоциативным запоминающим устройствам других блоков, а также для получения адреса хранения и анализа определенного количества ресурса, полученного от других блоков. Система распределения ресурсов определяет "свободные" места в ассоциативных запоминающих устройствах избыточных ресурсов и их параметров. Процесс передачи избытка на свободные места произойдет тогда, когда объемы избыточных ресурсов и параметров равны или меньше предоставленных объемов "свободных" мест [8].

Специализированный вычислительный модуль СпецВычМод4 выполняет

микрооперацию сложения чисел в троичной системе счисления (рис. 10).

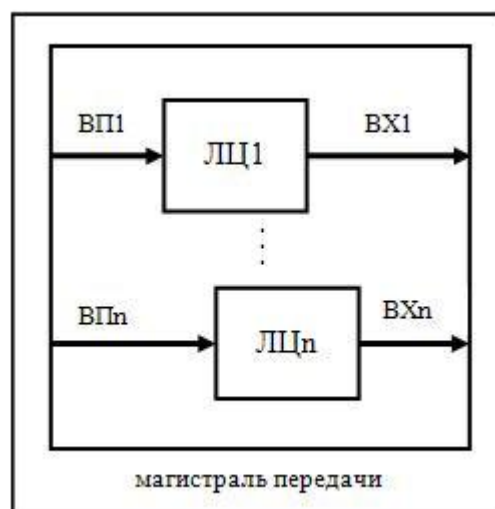


Рис. 9. Специализированный вычислительный модуль СпецВычМод3: ЛЦn – локализованные центры системы распределения; магистраль передачи

Использование троичной системы счисления обосновывается следующим результатом. При сложении младших разрядов A_0 и B_0 , старших A_1 и B_1 разрядов и переносов из младших разрядов в старшие P_{00} , P_{01} формируются суммы Σ_0 , Σ_1 и переносы P_0 , P_1 , $A_1 \in \{0, 1\}$; $B_1 \in \{0, 1\}$. Сигналы Σ_{10} и Σ_{11} представляют младший и старший разряды окончательной суммы [9].

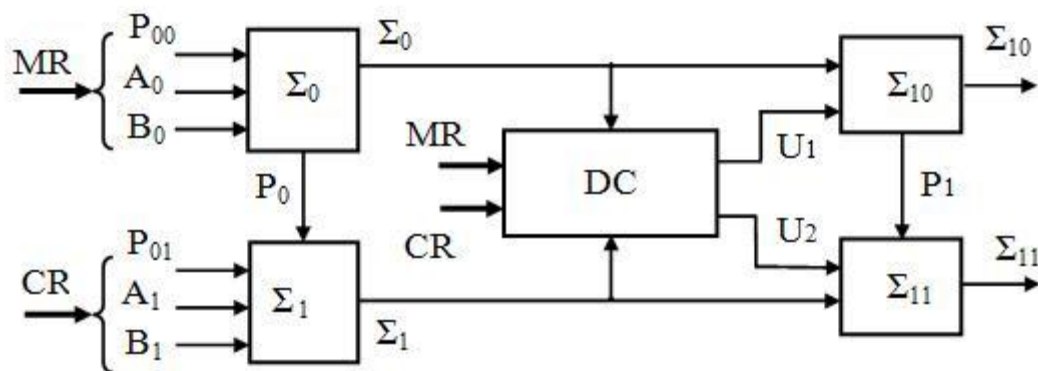


Рис. 10. Специализированный вычислительный модуль СпецВычМод4

$$\Sigma_0 = \bigoplus_{i=1}^3 (A_0 B_0 P_{00}), \quad \Sigma_0 \in \{0, 1\};$$

$$A_0 \in \{0, 1\}; B_0 \in \{0, 1\}; P_{00} \in \{0, 1\}; \quad (10)$$

$$\Sigma_1 = (\bigoplus_{i=1}^2 (P_0 P_{01}) \oplus (\bigoplus_{i=1}^2 (A_1 B_1))), \quad \Sigma_1 \in \{0, 1\};$$

$$P_0 \in \{0, 1\}; P_{01} \in \{0, 1\}; \quad (11)$$

$$\Sigma_{10} = \bigoplus_{i=1}^2 (\Sigma_0 U_1), \quad \Sigma_{10} \in \{0, 1\};$$

$$\Sigma_0 \in \{0, 1\}; U_1 \in \{0, 1\}; \quad (12)$$

$$\Sigma_{11} = \bigoplus_{i=1}^3 (P_1 \Sigma_1 U_2),$$

$$\Sigma_{11} \in \{0, 1\}; P_1 \in \{0, 1\};$$

$$\Sigma_1 \in \{0, 1\}; U_2 \in \{0, 1\}. \quad (13)$$

Специализированный вычислительный модуль СпецВычМод5 выполняет сортировку информации методом преобразования данных в адрес (рис. 11). Входные данные поступают одновременно на информационные и адресные

входы оперативных запоминающих устройств. По сформированным адресам данные и числа одинаковых символов и чисел записываются в оперативную память устройства. Преобразование данных в адрес происходит с помощью хеш-функции, которая отображает каждый ключ из набора S во множество целых чисел без коллизий. Инъективное отображение для целого описывается функцией $h(k): U \rightarrow [m]$, она является k -идеальной хеш-функцией для $S \subseteq U$, если для каждого $j \in [m]$ имеем $|\{x \in S | h(x) = j\}| \leq k$ [10].

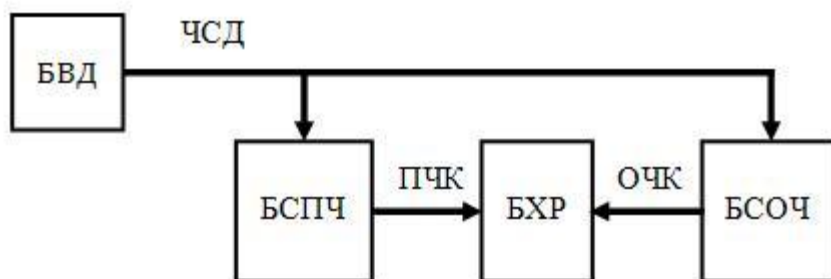


Рис. 11. Специализированный вычислительный модуль СпецВычМод5: БВД – блок ввода данных; БСПЧ – блок сортировки положительных чисел; БСОЧ – блок сортировки отрицательных чисел; БХР – блок хранения результата

Выводы

Архитектура реконфигурируемых вычислительных систем может динамически изменяться в процессе их функционирования. Появляется возможность адаптации архитектуры вычислительной системы под структуру решаемой задачи, создавать проблемно-ориентированные вычислители, структура которых соответствует структуре решаемой задачи. В качестве основного вычислительного элемента в реконфигурируемых вычислительных системах используются не универсальные микропроцессоры, а программируемые логические интегральные схемы, которые объединены с помощью высокоскоростных интерфейсов в единое вычислительное поле. Реконфигурируемые мультимиконвейерные вычислительные системы на основе полей являются эффективным средством для решения потоковых задач обработки информации и

управления. Они могут создавать в базовой архитектуре поля программируемых логических интегральных схем, виртуальные специализированные мультимиконвейерные вычислительные модули, адекватные графу решаемой задачи. Это обеспечивает высокую эффективность мультимиконвейерных вычислений, дает превосходство по технико-экономическим характеристикам над МВС с традиционной архитектурой. Семейство реконфигурируемых вычислительных систем с динамически перестраиваемой архитектурой на основе программируемых логических интегральных схем предназначено для решения вычислительно трудоемких задач, в том числе и большого размера, решение которых на многопроцессорных вычислительных системах традиционной архитектуры либо требует недопустимо больших временных затрат, либо вообще невозможно [11].

Список литературы

1. Хорошевский В.Г. Архитектура вычислительных систем. М., 2008. 520 с.
2. Гузик В.Ф., Каляев И.А., Левин И.И. Реконфигурируемые вычислительные системы. Таганрог.: Южный федеральный университет, 2016. 472 с.
3. Пат. 2453910 Рос. Федерация Вычислительная открытая развиваемая асинхронная модульная система ВОРАМС / Шевелев С.С.; № 2009113184/08 заявл. 08.04.2009; опубл. 20.06.2012, Бюл. № 17. 81 с.
4. Хокни Р., Джессхоуп К. Параллельные ЭВМ. Архитектура, программирование и алгоритмы. М.: Радио и связь, 1986. 392 с.

5. Шевелев С.С. Вычислительная открытая развиваемая асинхронная модульная система // Проблемы информационной безопасности. Компьютерные системы Санкт-Петербургского государственного политехнического университета. 2013. № 4. С. 86-92.

6. Шевелев С.С. Устройство параллельного поиска и замены вхождений в обрабатываемых словах // Известия Юго-Западного государственного университета. 2012. №4 (43). С. 32-36.

7. Пат. 2223538 Рос. Федерация Устройство сортировки слов. Шевелев С.С.; № 2002109134/09/ заявл. 08.04.2002; опубл. 10.02.2004, Бюл. № 4.

8. Шевелев С.С., Лопин В.Н. Система взаимораспределения ресурсов // Известия Курского государственного технического университета. 2008. №1 (22). С. 74– 78.

9. Пат. 2453900 Рос. Федерация. Параллельный сумматор-вычитатель в троичной системе счисления на нейронах. Шевелев С.С. № 2010108106/08/; заявл. 04.03.2010; опубл. 20.06.2012, Бюл. № 17.

10. Пат. 2382396 Рос. Федерация. Устройство сортировки информации методом преобразования данных в адрес. Шевелев С.С., Кобелев В.Н., Шевелева Е. С., Солодовников Ф.М. № 2008112463/09/; заявл. 31.03.2008; опубл. 20.02.2010, Бюл. № 5.

11. Каляев И.А., Левин И.И., Семерников Е.А. Реконфигурируемые мультиконвейерные вычислительные структуры. Ростов н/Д.: Изд-во ЮНЦ РАН, 2008. 320 с.

Поступила в редакцию 11.03.2019

Подписана в печать 27.03.2019

Reference

1. Khoroshevsky V.G. Architecture of computing systems. Moscow, 2008, 520 p. (In Russ.).

2. Guzik V.F., Kalyaev I.A., Levin I.I. Reconfigurable computing systems. Taganrog, 2016, 472 p. (In Russ.).

3. Pat. 2453910 Ros. Federation. Computational open developed asynchronous modular system VORAMS / S.Shevelev; No. 2009113184/08 declare 04/08/2009; publ. 20.06.2012, Byul. no. 17. 81 p.

4. Hockney R., Jesshope K. Parallel computers. Architecture, programming and algorithms. Moscow, Radio and communication Publ., 1986, 392 p. (In Russ.).

5. Shevelev S.S. Vychislitel'naya otkrytaya razvivaemaya asinkhronnaya modul'naya sistema [Computational open developed asynchronous modular system]. *Problemy informatsionnoi bezopasnosti. Komp'yuternye sistemy Sankt-Peterburgskogo gosudarstvennogo politekhnicheskogo universiteta* = *Problems of information security. Computer systems of the St. Petersburg State Polytechnic University*, 2013, no. 4, pp. 86-92 (In Russ.).

6. Shevelev S.S. Ustroistvo parallel'nogo poiska i zameny vkhozhdenni v obrabatyvaemykh slovakh [The device parallel search and replace occurrences in the processed words]. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*, 2012, no.4 (43), pp. 32-36 (In Russ.).
7. Pat. 2223538 Ros. Federation Word sorting device. S.Shevelev; No. 2002109134/09 / declare 04/08/2002; publ. 10.02.2004, Byul. no. 4.
8. Shevelev S.S., Lopin V.N. Sistema vzaimoraspredeleniya resursov [System of mutual allocation of resources]. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*, 2008, no.1 (22), pp. 74– 78 (In Russ.).
9. Pat. 2453900 Ros. Federation. Parallel adder-subtractor in the ternary number system on neurons / S.Shevelev; No. 2010108106/08 declare 03/04/2010; publ. 20.06.2012, Byul. no. 17.
10. Pat. 2382396 Ros. Federation. Shevelev S.S., Kobelev V.N., Sheveleva E.S., Solodovnikov F.M. A device for sorting information by converting data to an address. No. 2008112463/09; declare March 31, 2008; publ. 20.02.2010, Bull. No. 5
11. Kalyaev I.A., Levin I.I., Semernikov E.A. Reconfigurable multi-pipeline computing structures. Rostov-on-Don, 2008, 320 p. (In Russ.).

Received 11.03.2019

Accepted 27.03.2019

Информация об авторах / Information about the Authors

Сергей Степанович Шевелев, кандидат технических наук, доцент, ФГБОУ ВО «Юго-Западный государственный университет», г. Курск, Российская Федерация, e-mail: schewelew@mail.ru

Sergey S. Shevelev, Candidate of Engineering Sciences, Associate Professor, Southwest State University, Kursk, Russian Federation, e-mail: schewelew@mail.ru