

УДК 004.031.43–044.962

А. С. Кобайло

Белорусский государственный технологический университет

**ОСОБЕННОСТИ АРХИТЕКТУРНОЙ ОРГАНИЗАЦИИ
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ.
ПАРАЛЛЕЛЬНО-КОНВЕЙЕРНЫЕ ВЫЧИСЛИТЕЛИ**

Проведен анализ ограничений применения классических архитектурных решений задачи повышения производительности вычислительных систем – конвейеризации и параллелизма. Рассмотрен один из архитектурных подходов к решению задачи обеспечения режима реального времени для высокопроизводительных вычислительных систем. Предложенный метод основан на сочетании параллелизма и конвейеризации с помощью так называемых параллельно-конвейерных и конвейерно-параллельных вычислителей. Приводятся расчетные соотношения для определения эффективности вычислительных систем, организованных на основе предложенных методов. Рассмотрены примеры использования соответствующих архитектур вычислительных систем для решения конкретных типовых задач, в частности векторных операций.

Применение параллельно-конвейерных вычислителей окажется целесообразным при построении вычислительных систем в условиях отсутствия в распоряжении проектировщика функциональных устройств с временем выполнения операций алгоритма, не превышающим длительности цикла обработки данных, требуемой в соответствии с заданными условиями реального времени.

Ключевые слова: производительность, вычислительная система, параллелизм, конвейеризация, реальное время, цикл конвейера, параллельно-конвейерный вычислитель.

A. S. Kobaylo

Belarusian State Technological University

**PECULIARITIES OF ARCHITECTURAL ORGANIZATION
OF COMPUTER SYSTEMS OF REAL TIME.
PARALLEL-CONVEYOR COMPUTERS**

The analysis of the limitations of the application of classical architectural solutions to the problem of increasing the performance of computational systems – pipelining and concurrency is carried out. One of the architectural approaches to solving the problem of providing real-time mode for high-performance computing systems is considered. The proposed method is based on a combination of parallelism and pipelining with the help of so-called parallel-conveyor and pipeline-parallel calculators. The calculated relationships are used to determine the efficiency of computer systems organized on the basis of the proposed methods. Examples of the use of the corresponding architectures of computer systems for solving specific typical problems, in particular vector operations, are considered.

The use of parallel-conveyor computers will prove to be useful in the construction of computing systems in the absence of a functional device at the designer's disposal with an algorithm execution time that does not exceed the processing cycle time required in accordance with specified real-time conditions.

Key words: performance, computational system, parallelism, pipelining, real time, conveyor cycle, parallel-conveyor computer.

Введение. Достижение требуемой производительности вычислительных систем (ВС) при ограниченных предельных возможностях элементной базы обеспечивается использованием архитектурных подходов к проектированию, основанных на совмещении операций или фрагментов алгоритма во времени и пространстве – конвейеризации и параллелизма.

Конвейеризация – это такой метод организации вычислительного процесса, в результате применения которого в вычислительной системе обеспечивается совмещение разных дейст-

вий по выполнению базовых функций путем их разбиения на подфункции. При этом традиционно считается, что конвейеризация позволяет повысить производительность ВС по сравнению с последовательной системой примерно в r раз, т. е. на величину, равную количеству ступеней конвейера (глубине конвейера) [1]. Однако по ряду причин и в первую очередь в силу возможного значительного различия во времени выполнения операций конвейеризируемого алгоритма реальное ускорение конвейерной ВС может оказаться кроме этого

(при необходимости получения данных с частотой, превышающей частоту формирования выходных данных конвейера, в частности при построении некоторых типов систем реального времени (РВ)) скорость конвейера может оказаться недостаточной.

Второй альтернативный подход к развитию вычислительных архитектур с целью увеличения их быстродействия – параллелизм – предусматривает одновременное выполнение независимых операций задачи разными блоками ВС или одновременную обработку разных данных по однотипным алгоритмам. При этом архитектура, имеющая N параллельных каналов, дает преимущество в быстродействии в сравнении с последовательной обработкой в N раз.

Применение параллелизма требует одновременного наличия данных для всех параллельных ветвей, что существенно лимитирует область использования данного метода.

Таким образом, два классических подхода к решению задачи построения высокопроизводительных вычислительных систем имеют очевидные ограничения, так как первый из них может не обеспечить выполнения требований по производительности, в том числе по обеспечению РВ, второй может оказаться принципиально непригодным при последовательном поступлении данных.

Вышесказанное обуславливает следующие рекомендации по применению рассмотренных классических методов повышения производительности.

Конвейеризация может быть использована в случаях, когда требуемая скорость обработки потока данных удовлетворяет условию:

$$\Delta t = \tau_{\max},$$

где Δt – требуемый период получения данных на выходе структуры; τ_{\max} – время выполнения самой длинной операции в цепочке вычислений техническими средствами, на которые ориентируется разработчик.

Параллелизм целесообразно применять при возможности накопления или одновременного поступления массива входных данных.

Из вышесказанного следует необходимость разработки принципиально новых подходов к решению проблемы обеспечения высокой производительности проектируемых ВС.

Основная часть. Принцип реализации конвейера на интегрированном конвейерном вычислителе, основанный на возможности объединения на одном уровне ряда последовательно выполняемых подфункций с учетом их реализации в реальном времени, позволяющий максимально загрузить ступени конвейера и приблизить реальное повышение производитель-

ности конвейера по отношению к последовательной вычислительной системе к теоретической, равной глубине конвейера, рассмотрен в статье [2]. В настоящей работе разберем один из подходов к решению поставленной задачи, основанный на совмещении двух вышеуказанных основополагающих принципов создания высокопроизводительных систем.

В тех случаях, когда для реализации некоторых операций алгоритма отсутствуют функциональные устройства (ФУ) с временем выполнения соответствующих операций не большим, чем требуемый цикл обработки данных, построение архитектур реального времени становится невозможным. Для решения данной проблемы предлагается введение в проектируемые системы параллельно-конвейерных вычислителей (ПКВ).

ОПРЕДЕЛЕНИЕ 1. Параллельно-конвейерным вычислителем называется вычислитель, содержащий ρ параллельных ступеней, выполняющих последовательность однотипных операций с одинаковым временным сдвигом, равным периоду формирования очередных результатов на выходе ПКВ [3].

ОПРЕДЕЛЕНИЕ 2. Циклом ПКВ называется интервал времени, равный периоду формирования очередных результатов на выходе ПКВ.

Диаграмма работы ПКВ представлена на рис. 1. На данном рисунке μ – номер ступени ПКВ, ρ – количество ступеней (глубина ПКВ).

Рассмотрим основные особенности проектирования вычислительных структур с использованием ПКВ.

УТВЕРЖДЕНИЕ 1. Длительность цикла ПКВ имеет вид

$$\tau_{\text{ц}} = \frac{t_{\text{п}}}{\rho},$$

где $t_{\text{п}}$ – время выполнения последовательности операций набором простых ФУ; $\tau_{\text{ц}}$ – цикл ПКВ; ρ – количество ступеней ПКВ.

Действительно, при циклической организации вычислительного процесса интервал $t_{\text{п}}$ складывается из ρ интервалов длительностью $\tau_{\text{ц}}$, или $t_{\text{п}} = \rho \tau_{\text{ц}}$. Каждый очередной отсчет переменных на выходе, в соответствии с определением ПКВ, формируется со смещением $\tau_{\text{ц}}$ соответствующей ступенью ПКВ, т. е.: $\tau = \tau_{\text{ц}} \rightarrow \rho = t_{\text{п}} / \tau_{\text{ц}}$, из чего следует справедливость утверждения.

УТВЕРЖДЕНИЕ 2. Применение ПКВ для обработки последовательности операций позволяет ускорить процесс этой обработки в ρ раз по сравнению с реализацией тех же действий на простых ФУ, реализованных на той же элементной базе, что и ПКВ.

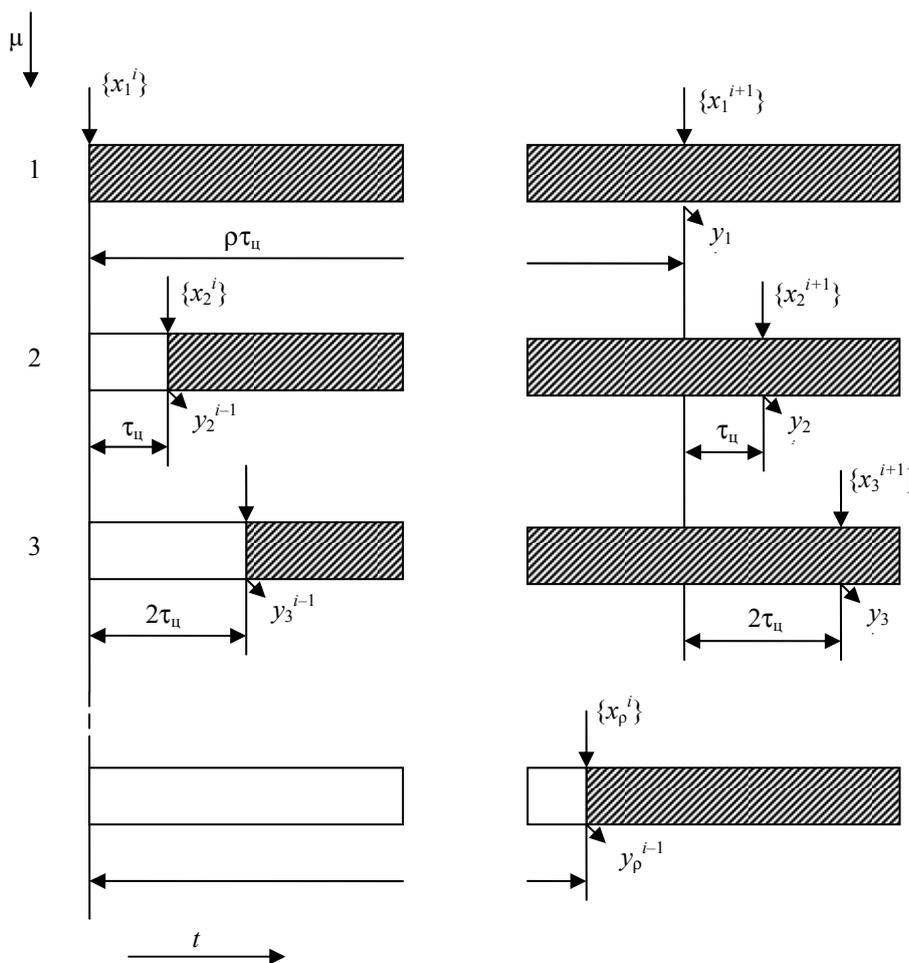


Рис. 1. Диаграмма загрузки ПКВ:
 $\{x_\mu^i\}, \{y_\mu^i\}, \mu = 1, \dots, \rho$ – входные и выходные соответственно данные μ -й ступени i -го цикла

Действительно, пусть некоторая последовательность действий по обработке данных последовательно простых функциональных устройств выполняется за интервал времени t . Каждый следующий отсчет выходных данных в этом случае может быть получен через интервал t после предыдущего. Тогда N -кратное прохождение цикла выполняется за время $T_{пр} = Nt$.

Используя для этой цели ПКВ, реализованный на той же элементной базе, что и простое ФУ, получим на выходе результат также через время t после начала обработки входных данных, но каждый следующий отсчет данных на выходе вычислителя будет получен через $\tau_{ц} = t / \rho$ после предыдущего. Тогда время выполнения N циклов ПКВ будет равно $T_{пр} = t + (N - 1) \tau_{ц} \approx \tau_{ц} N = tN / \rho$, или $T_{пр} / \tau_{ц} = \rho$, что и требовалось доказать.

УТВЕРЖДЕНИЕ 3. Максимально возможная скорость обработки данных ПКВ определяется быстродействием элементной базы и равняется $1/\tau_{ц}$ слов в единицу времени, где $\tau_{ц}$ – такт вычислительной структуры.

Действительно, включение соседней ступени ПКВ возможно через интервал времени, равный $\tau_{ц}$, после включения предыдущей, или

$$t_1^{(i+1)} = t_1^{(i)} + \tau^{(i)},$$

где $t_1^{(i)}$ и $t_1^{(i+1)}$ – моменты начала обработки данных i -й и $(i + 1)$ -й ступенями. Тогда моменты завершения $t_2^{(i)}$ и $t_2^{(i+1)}$ обработки данных этими ступенями определяются как

$$t_2^{(i)} = t_1^{(i)} + \tau^{(i)}, t_2^{(i+1)} = t_1^{(i+1)} + \tau^{(i+1)},$$

где $\tau^{(i)}$ и $\tau^{(i+1)}$ – время обработки данных i -й и $(i + 1)$ -й ступенями соответственно. Тогда минимальный период получения данных на выходе ПКВ имеет вид

$$\tau_{\min} = t_2^{(i+1)} - t_2^{(i)} = \tau_{ц},$$

что и требовалось доказать.

Введение ПКВ, как следует из его определения, предусматривает преобразование графа алгоритма в результате реализации его пути,

отображенного на ПКВ, p параллельными каналами. Преобразование графа вычислительного алгоритма при отображении его пути на ПКВ осуществляется в два этапа, соответствующие введению собственного ПКВ и его ступеней. На первом этапе происходит преобразование, представленное на рис. 2.

В соответствии с данным преобразованием над каждой из дуг e_l выходной относительно последовательности вершин $\{v_i\}$, отображаемых на ПКВ, выполняется операция добавления вершины v'_j , отождествляемой с операцией мультиплексирования, при этом дуга e_l заменяется парой дуг e'_l и $e'_{\text{ВЫХ}}$.

Второй этап преобразования графа вычислительного алгоритма при отображении его путей на ПКВ базируется на следующем положении.

УТВЕРЖДЕНИЕ 4. Расширенный подграф, соответствующий пути графа вычислительного алгоритма $\{v_1, v_2, \dots, v_j\}$, отображенного на ПКВ, формируется как пересечение подграфов

$$G = G_1 \cap G_2 \cap \dots \cap G_i,$$

где $G_i, i = \overline{1, I}$ – i -й подграф, изоморфный подграфу, соответствующему пути $\{v_1, v_2, \dots, v_j\}$.

Действительно, согласно определению, реализация пути L графа вычислительного алгоритма параллельно-конвейерным вычислителем осуществляет обработку данных по p идентичным параллельным каналам (ступеням), при этом каждому μ -му, $\mu = \overline{1, p}$ каналу соответствует подграф G_μ , изоморфный подграфу пути L . Каждый из этих подграфов получает данные от общей вершины $v'_{\text{ВХ}}$, которая является образом вершины $v_{\text{ВХ}}$ графа алгоритма; $v_{\text{ВХ}}: \exists e_{\text{ВХ}} = (v_{\text{ВХ}}, v_1)$. Тогда $\forall i, \mu: v_{i,\mu}$ является образом $v'_i, e'_{i,\mu}$ – образом e'_i . Кроме того, $t_{i,\mu}^k = t_{i,\mu-1}^k + \tau_{\text{ц}}$ в соответствии с определением ПКВ, где $t_{i,\mu-1}^k, t_{i,\mu}^k$ – время окончания выполнения операций k -го цикла вершинами $v'_{i,\mu-1}$ и $v'_{i,\mu}$ соответственно. Тогда над вершинами $v'_{i,\mu}$ может быть выполнена операция простого элементарного гомоморфизма, таким образом, граф ПКВ получается пересечением подграфов путей, изоморфных подграфу, полученному на первом этапе преобразования, с общей конечной вершиной каждого из подпутей (рис. 3).

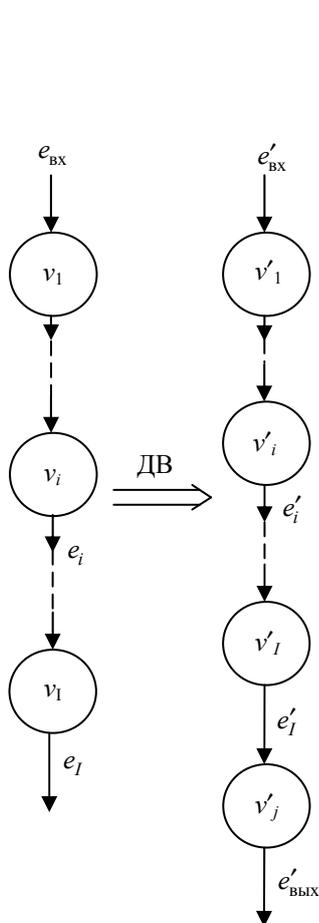


Рис. 2. Первый этап преобразования подграфа при отображении пути на ПКВ: ДВ – добавление вершины

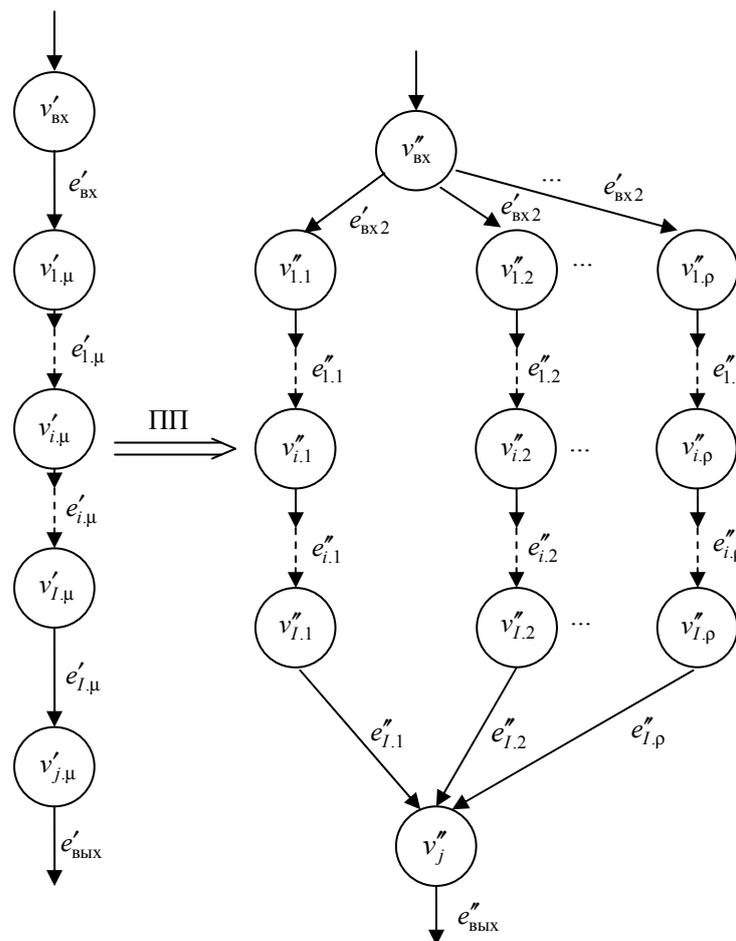


Рис. 3. Второй этап преобразования подграфа при отображении пути на ПКВ: ПП – пересечение подграфов

СЛЕДСТВИЕ. Скорость обработки данных с помощью ПКВ определяется скоростью мультиплексора

$$f = \frac{1}{\tau_{\text{зп}}(Mx)} k,$$

где f – частота поступления данных на вход мультиплексора, $\tau_{\text{зп}}(Mx)$ – время задержки распространения мультиплексора, $k = 2-4$.

Заключение. Предложенный принцип параллельно-конвейерной обработки данных позволяет:

- организовать параллельную обработку данных при их последовательном поступлении на вход системы в условиях отсутствия возможности их накопления;

- достигнуть быстродействия обработки потока данных, определяемого только частотой переключения элементной базы (например, так-

товая частота для микросхем линии Хеоп составляет 3,3 ГГц);

- увеличить быстродействие по сравнению с последовательной системой в число раз, равное количеству параллельных каналов, и по сравнению с конвейерной системой – пропорционально отношению длительности цикла конвейера к длительности цикла ПКВ.

Особенно эффективным окажется применение параллельно-конвейерной обработки информации в автоматизированных системах управления производственным оборудованием и технологическими процессами, системах автоматизации научного эксперимента, цифровой обработки сигналов, при решении задач линейной алгебры.

Особенности применения ПКВ для решения ряда типовых задач требуют особого рассмотрения.

Литература

1. Коуги П. Архитектура конвейерных ЭВМ. М.: Радио и связь, 1985. 567 с.
2. Кобайло А. С. Особенности архитектурной организации вычислительных систем реального времени. Интегрированные конвейерные вычислители // Труды БГТУ. Сер. 3, Физ.-мат. науки и информатика. 2017. № 2. С. 129–132.
3. Кобайло А. С. Основы теории синтеза вычислительных структур реального времени. Минск: БГУИР, 2001. 202 с.
4. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: Издательство БХВ-Петербург, 2002. 609 с.

References

1. Kougi P. *Arkhitektura konveyyernykh EVM* [The architecture of conveyor computers]. Moscow, Radio i svyaz' Publ., 1985. 567 p.
2. Kobaylo A. S. The peculiarities of architectural organization of computer systems of real time. Integrated conveyor calculators. *Trudy BGTU* [Processings of BSTU], series 3, Physics and mathematics. Informatics, 2017, no. 2, pp. 129–132 (In Russian).
3. Kobaylo A. S. *Osnovy teorii sinteza vychislitel'nykh struktur real'nogo vremeni* [Fundamentals of the theory of synthesis of computational structures of real time]. Minsk, BGUIR Publ., 2001. 202 p.
4. Voevodin V. V., Voevodin Vl. V. *Parallel'nye vychisleniya* [Parallel computing]. St. Petersburg, Izdatel'stvo BKhV-Peterburg Publ., 2002. 609 p.

Информация об авторе

Кобайло Александр Серафимович – кандидат технических наук, доцент, доцент кафедры информационных систем и технологий. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: kas151153@gmail.com

Information about the author

Kobaylo Alexandr Serafimovich – PhD (Engineering), Assistant Professor, Associate Professor, the Department of Information Systems and Technologies. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: kas151153@gmail.com

Поступила 28.11.2017