

УДК 004.031.43–044.962

ПРИМЕНЕНИЕ КОНВЕЙЕРНОЙ ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ

Жиляк Н.А.

Белорусский государственный технологический университет

E-mail: gznadya@gmail.com

Аннотация:

Жиляк Н.А. Применение конвейерной организации вычислительного процесса в системах реального времени. Проведен анализ ограничений применения классических архитектурных решений задачи повышения производительности вычислительных систем – конвейеризации и параллелизма. Рассмотрены некоторые архитектурные подходы к решению задачи обеспечения режима реального времени для высокопроизводительных вычислительных систем.

Ключевые слова: производительность, вычислительная система, параллелизм, конвейеризация, реальное время, цикл конвейера.

Annotation:

Zhilyak N.A. Application of the conveyor organization of the computing process in real-time systems. 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. Some architectural approaches to solving the problem of real-time mode provision for high-performance computing systems are considered.

Keywords: Performance, computational system, parallelism, pipelining, real time, conveyor cycle.

Общая постановка проблемы

Одной из основных проблем, возникающих при проектировании вычислительных систем реального времени (ВС РВ), является достижимость требуемого быстродействия (производительности). Для последовательного выполнения алгоритма из j операции необходим ресурс времени, который определяется:

$$T = \sum_{i=1}^I \tau'_i \cdot \tau_c, \quad (1)$$

где τ'_i – код продолжительности выполнения i -й операции, τ_c – такт или время цикла, которому пропорциональны моменты начала или завершения любых действий в системе (более строгое определение термина будет дано в следующем разделе). Повышение быстродействия элементной базы или, что то же самое, уменьшение значения τ_c имеет свой предел, ограниченный скоростью света. Поэтому для решения названной проблемы более перспективными являются пути поиска архитектурной организации ВС, связанные, в первую очередь, с совмещением операций. Два основных подхода в этом направлении – конвейеризация и параллелизм.

Исследования

Напомним особенности этих подходов для организации вычислительных архитектур. Принцип конвейеризации. Конвейеризация – это такой метод организации вычислительного процесса, в результате применения которого в вычислительной системе обеспечивается

совмещение разных действий по выполнению базовых функций путем их разбиения на подфункции.

Применение конвейерной организации вычислительного процесса позволяет увеличить скорость обработки данных в ρ раз, где ρ – количество ступеней.

Можно выделить два уровня конвейеризации – на базе конвейерных функциональных устройств и на основе конвейерных вычислителей (КВ).

Первый из выделенных уровней предусматривает выполнение некоторой функции в соответствии с приведенными выше правилами и ориентирован на возможность реализации этой функции одним функциональным устройством (ФУ), организованным по конвейерному принципу. Примеры конвейеризации на первом уровне приведены в [1]. Так, N – разрядный сумматор имеет диагональную структуру, построен на основе N одноразрядных полных сумматоров с информационными входами X и Y и входом переноса Z , на элементах задержки, позволяющих осуществить задержку промежуточных результатов вычислений на один цикл, количество ступеней равно требуемой разрядности. В результате такой организации скорость обработки потока данных увеличивается в N раз.

Второй уровень конвейеризации является более общим и ориентирован на конвейеризацию последовательности операций, которые могут быть самыми разными как по своей сущности, так и по времени реализации, причем каждая из операций может выполняться собственным ФУ [2].

Такой подход к конвейеризации вычислительного процесса был предложен автором данного исследования и наиболее подробно рассмотрен в [3, 4].

Сущность принципа проиллюстрирована на рисунке 1, соответствующая временная диаграмма приводится на рисунке 2.

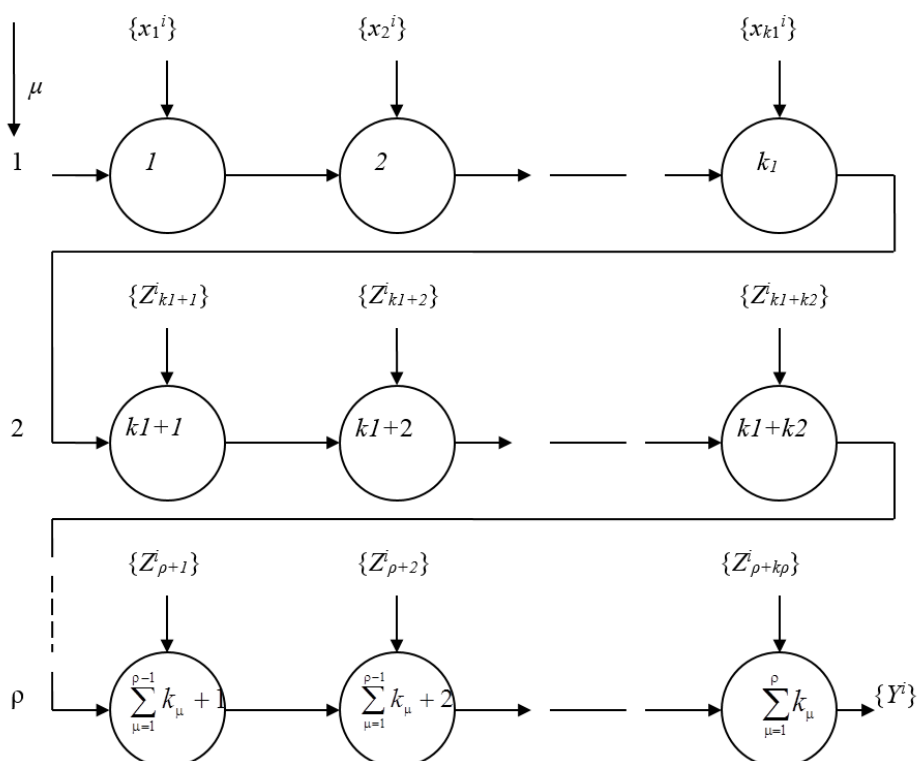


Рис. 1. Конвейеризация на основе КВ

Основные особенности этого подхода заключаются в следующем:

- для уменьшения разницы во времени выполнения операций разными ступенями на реализацию одной ступенью ориентируется последовательность операций (а не одна подфункция);
- каждая из ступеней КВ строится как последовательность разных ФУ;
- входные данные ступени конвейера можно получать как от предыдущей ступени, так и от других технических средств системы, в том числе и от других КВ.

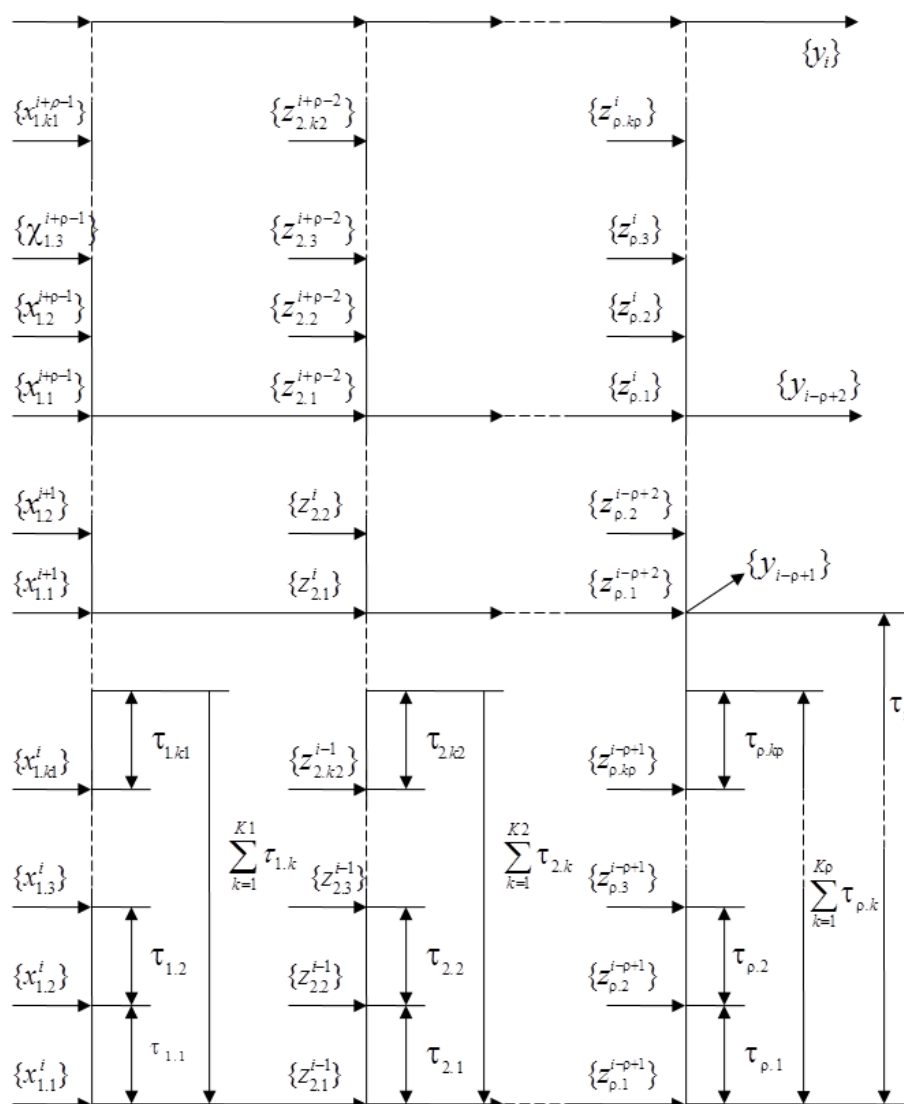


Рис. 2. Диаграмма загрузки КВ: τ_{π} – цикл конвейера

На этих рисунках: $\{X_{\mu k}^j\}$, $j = i, i+1, i+2, \dots$; $\mu = 1, 2, \dots$, $\rho : k = 1, 2, \dots$, K_{μ} входные данные k -о ФУ μ -й ступени на j -м цикле, где K_{μ} – общее количество операций μ -й ступени, обозначения вершин – последовательные номера вершин конвейеризируемого пути.

Конвейеризация второго уровня представляет собой гибкий инструмент при выполнении сложных алгоритмов и построении соответствующих вычислительных структур, является одним из эффективных путей выполнения требований реального времени.

Отметим, что любой из рассмотренных уровней конвейеризации позволяет получить скорость обработки потока данных, определяемую циклом конвейера $\tau_k = \tau'_k \cdot \tau_c$ и $X = I / \tau_k$,

где X – количество машинных слов на выходе конвейера в единицу времени, τ_k – код цикла конвейера. Этот показатель ограничен длительностью самой длинной из операций, выполняемых конвейером:

$$\tau_k \geq \max_{1 \leq i \leq l} \tau_i, \quad (2)$$

где τ_i – время выполнения i -й операции из конвейеризируемой последовательности.

Принцип параллелизма в вычислительных системах. 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 [6, 7].

При этом выделяют следующие виды параллелизма: натуральный параллелизм и параллелизм множества объектов, параллелизм независимых ветвей, параллелизм смежных операций.

Первый из названных видов параллелизма наиболее распространен и относится к задачам, требующим быстрой обработки больших массивов данных, и эта обработка обычно использует методы линейной алгебры, алгоритм свертки, цифровую фильтрацию и т. д.

Параллелизм независимых ветвей предполагает выделение в программе решения объемной задачи независимых ветвей. Критерий независимости ветвей А и В следующий:

- независимость ветвей по данным: это значит, что ни один из выходных данных ветви не является входным данным ветви В;
- выполнение ветвей А и В по разным программам;
- независимость управления ветвями.

При параллелизме множества объектов имеет место отличие по второму критерию; для него все параллельные ветви обрабатываются по одной программе.

На сегодняшний день наиболее разработаны методы распараллеливания первого типа. 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 ограничений, так как первый из них может не обеспечить выполнения требований 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:

$$\Delta t = \tau_{\max}, \quad (3)$$

где Δt – требуемый период получения данных на выходе структуры, $\tau_{\max}$ – время выполнения наиболее длинной операции техническими средствами, на которые ориентируется ветвей, что существенно ограничивает применение этого метода. Кроме того, при расчете быстродействия нужно учитывать также и временные затраты на обработку данных одним каналом.

Параллелизм целесообразно применять при возможности накопления или одновременного поступления массива входных данных. Из вышесказанного следует необходимость разработки принципиально новых подходов к решению проблемы обеспечения функционирования ВС в реальном времени.

Параллельно-конвейерные вычислители. В тех случаях, когда для реализации некоторых операций алгоритма отсутствуют ФУ с временем выполнения соответствующих

операций, не большим, чем требуемый цикл обработки данных, построение архитектур реального времени становится невозможным. Для решения проблемы предлагается введение в проектируемые системы параллельно-конвейерные вычислители (ПКВ).

Параллельно-конвейерным вычислителем называется вычислитель, содержащий p параллельных ступеней, выполняющих последовательность однотипных операций с одинаковым временным сдвигом, равным периоду формирования очередных результатов на выходе ПКВ.

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

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.

Предложенные методы основаны на сочетании параллелизма и конвейеризации либо нетрадиционной конвейерной организации вычислительного процесса в случае существенного различия временных показателей выполнения операций и реализуются в основном с помощью т. н. 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 [5, 6, 7].

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; конвейерно-параллельные вычислители могут быть использованы при обработке массивов данных большой размерности по одному алгоритму, если параллельная вычислительная система не позволяет обеспечить реальный масштаб времени или требуемую скорость обработки данных.

Выводы

Таким образом, в данном разделе предложены общие требования реализации вычислительного процесса в реальном времени, позволяющие всесторонне охватить проблему синтеза ВС РВ.

2. Постановка задачи синтеза вычислительных структур как классической проблемы отображения вычислительных алгоритмов на архитектуры вычислительных систем предоставляет возможность решать эту проблему методами теории графов, что упрощает переход от вычислительных систем с учетом заданных требований.

3. Использование предложенных автором единичных функций для представления математических моделей вычислительных процессов позволяет:

- представлять единым аналитическим выражением модели, содержащие временные параметры или операции временных преобразований, что упрощает процедуры синтеза вычислительных структур и средств управления, а также позволяет экономить память для хранения исходных данных и промежуточных результатов;
- описывать аналитическими выражениями формирование управляющих сигналов в требуемые моменты времени;
- автоматизировать синтез вычислительных структур для моделей с временными параметрами и преобразованиями и блоков управления ВС РВ;

- упростить ввод исходных данных, уменьшить требуемый объем памяти и трудоемкость алгоритмов при автоматизации проектирования, а также упростить проектируемые вычислительные структуры и уменьшить их аппаратные затраты вследствие отсутствия необходимости ввода в эти структуры средств для анализа попадания параметров процесса в нужные интервалы;

- использовать для реализации временных преобразований простые ФУ.

4. Метод представления математических моделей процессов, заданных дискретными отсчетами параметров времени позволит в N раз снизить трудоемкость проектирования вычислительных структур, а при автоматизации проектирования – объем памяти для хранения исходных данных и промежуточных результатов.

5. Метод синтеза математических моделей с требуемыми свойствами предоставляет возможности:

- существенно уменьшить стоимость и сократить сроки проектирования и изготовления вычислительных структур;

- создавать вычислительные устройства, отличающиеся от структур, построенных на основе базовой модели, меньшими аппаратными затратами, стоимостью, габаритами, энергопотреблением при обеспечении требуемых функциональных возможностей;

- на начальной стадии проектирования сделать вывод о принципиальной возможности обеспечения требуемых свойств [7].

6. Предложенный принцип организации параллельно-конвейерных вычислителей:

а) позволяет:

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

- увеличить быстродействие по сравнению с конвейеризацией для векторных операций в число раз, определяемое соотношением скоростей. выполнения операций умножения и сложения;

- б) предоставляет возможность при выполнении векторных операций получить выигрыш по совокупности технических параметров по сравнению с параллелизмом практически при том же быстродействии. Вопрос о целесообразности замены быстродействующего элемента на ПКВ должен решаться в каждом случае индивидуально.

7. Предложенный принцип организации вычислительных архитектур на базе конвейерно-параллельных вычислителей позволяет увеличить быстродействие по сравнению с параллельными вычислителями в p раз при тех же аппаратных затратах, по сравнению с конвейером в m раз, где p – количество ступеней конвейерной цепи КПВ, m – количество каналов КПВ.

Литература

1. Коуги П. Архитектура конвейерных ЭВМ. М.: Радио и связь, 1985. 567 с.
2. Кобайло А. С. Основы теории синтеза вычислительных структур реального времени. Минск: БГУИР, 2001. 202 с.
3. Воеводин В.В., Воеводин В.В. Параллельные вычисления. Издательство БХВ-Петербург, 2002. 609 с.
4. Кобайло, А. С. Теория синтеза вычислительных систем реального времени / А. С. Кобайло. – Минск: БГТУ, 2010. – 256 с.
5. Кобайло, А. С. Базовый алгоритм синтеза вычислительных структур реального времени / А. С. Кобайло, Н. А. Жилияк // Труды БГТУ. Сер. VI, Физ.-мат. науки и информатика – 2007. Вып. XV. – С.147–150.
6. Жилияк, Н. А. Базовый алгоритм синтеза вычислительных структур реального времени для теплоэнергетических измерений / Н. А. Жилияк, С. И. Акунович // Труды БГТУ. Сер. VI, Физ.-мат. науки и информ. – 2008. Вып. XVI. – С.129–132.
7. Жилияк, Н. А. Синтез вычислительных структур реального времени / Н. А. Жилияк // Автоматический контроль и автоматизация производственных процессов: материалы Междунар. науч.-тех. конф., Минск, 6–8 июня 2006г. // Белорус. гос. технол. ун-т – Минск, 2006. – С. 208–211.