Применения (опыт разработки и внедрения информационных технологий)

УДК 004.27; 004.274; 004.315

Ю.С. Яковлев

Об оценке эффективности применения ПЛИС в составе *PIM*-систем

Создан метод на основе анализа общих положений по распараллеливанию приложения применительно к *PIM*-системам с использованием ПЛИС, в рамках которого разработаны формулы для оценки их эффективности, и подтверждены выводы и рекомендации численными значениями и графиками зависимости различных параметров *PIM*-системы и ПЛИС, полученных по этим формулам.

Створено метод на основі аналізу загальних положень з розпаралелювання застосування стосовно *PIM*-систем з використанням ПЛІС, в рамках якого розроблено формули для оцінки ефективності застосування ПЛІС у складі *PIM*-системи та підтверджено висновки і рекомендації чисельними значеннями та графіками залежності різних параметрів *PIM*-системи і ПЛІС, отриманих за цими формулами.

Введение. Сегодня при создании супер-ЭВМ наблюдается тенденция использования высокоинтегрированной элементной базы, которая образует с многоядерными процессорами различные гибридные сочетания, например, в узлах суперкомпьютеров используют *PIM*-системы с *FPGA* (*Field – Programmable Gate Array*), 8–16 ядерные центральные процессоры совместно с 50–60 ядерными ускорителями и др. Отметим, что улучшение параметров проектируемой компьютерной системы достигается при использовании высокоинтегрированных *FPGA* совместно с процессорами (*CPU*) [1]: например, в супер-ЭВМ *Maxwell* с использованием чипов *Virtex*-4 [2], супер-ЭВМ *Netezza* на базе фирменных сочетаний *Snipped* чипов *FPGA* и *CPU*, обеспечивающих значительные ускорения обработки данных, реконфигурируемые многопроцессорные вычислительные системы *РВС*-7 на основе микросхемы ПЛИС *Xilinx Virtex*-7 [3, 4] и др.

Анализ архитектур и организации вычислительного процесса систем высокопроизводительных вычислений показал, что их развитие идет в направлении применения более совершенной твердотельной элементной базы и соответствующих архитектурно-структурных решений, включая применение чипов *FPGA* совместно с архитектурами типа «Процессор-в-памяти» (*PIM*-системы), применение многоядерных процессоров, коммутаторов узлов вычислений, ускорителей вычислений, твердотельных накопителей.

В работе [5] рассмотрен один из подходов применения ПЛИС для схемной реализации модулей программы распределения приложений для *PIM*-системы, который позволяет уменьшить время распределения приложения, снизить потребляемую мощность инструментальной ЭВМ, а также уменьшить трудоемкость процесса распараллеливания программы пользователя.

Постановка задачи

Однако, несмотря на перспективность применения *PIM*-систем совместно с ПЛИС, в отечественной и зарубежной литературе не уделяется должного внимания оценке эффективности использования ПЛИС в составе *PIM*-систем. В работах [6, 7] приводятся в общем виде безотносительно к типу используемых компьютерных систем (с рядом существенных допущений) оценки достижимого ускорения и показателей загрузки процессоров для простейших учебных примеров выполнения отдельных задач. Учитывая, что *PIM*-системы существенно отличаются от систем других типов [8], можно утверждать, что тема данной статьи и изложенные в ней материалы весьма актуальны. Цель статьи – на основе анализа общих положений по распараллеливанию приложения при-

менительно к *PIM*-системам с применением ПЛИС создать метод, в рамках которого разработать формулы для оценки эффективности применения ПЛИС в составе *PIM*-системы и подтвердить выводы и рекомендации численными значениями и графиками зависимости различных параметров *PIM*-системы и ПЛИС, которые получены по этим формулам.

Общие положения распараллеливания приложения применительно к *PIM*-системе с применением ПЛИС

Отметим, что классический вариант *PIM*-системы с учетом структурной организации выполнен полностью на отдельном кристалле и содержит один ведущий процессор *VP* и несколько так называемых процессорных ядер *PYa*, объединенных общей шиной в процессорную линейку, образующую *PYa**, которая подключена к *VP* (рис. 1). При этом распараллеливание реализуемого алгоритма внутри одного кристалла происходит на двух уровнях: между *VP* и эквивалентным *PYa**, а также между *PYa* внутри *PYa** [9] с соблюдением следующих правил.

1. Распределение каждого q-го блока реализуемого алгоритма между *VP* и эквивалентным *PYa** выполняется по совпадению кодов их команд с кодами операций q-го блока, при этом в случае их совпадений вычисляется время выполнения всех кодов операций q-го блока алгоритма, и этот блок передается на *VP* или на *PYa**, время выполнения с помощью которого наименьшее.

2. При распределении между всеми *PYa* линейки *PYa** и в дальнейшем при распараллеливании прежде всего используют стандартные подпрограммы для конкретных задач, например, при выполнении операций над матрицами, при решении систем уравнений большого порядка и т.д. Если таковые применить невозможно, то сначала определяют базовое множество $M_б$ кодов операций q-го блока программы как множество, время выполнения которого приближено к среднеарифметическому времени выполнения для множеств $M_{\min}$ и $M_{\max}$. Далее для получения равномерного баланса загрузки для всех *PYa* используют стандартные методы распределения и соответственно распараллеливания (например, метод итераций), принимая во внимание принципы распределения по данным. При этом:

- один и тот же процессор не может назначаться разным операциям в один и тот же момент времени;
- к назначаемому моменту выполнения очередной операции все необходимые данные для последующих операций должны быть вычислены. Например, операции на *PYa* не могут быть выполнены, пока для этого не будут подготовлены все необходимые данные с помощью *VP*;
- время выполнения параллельного алгоритма определяется максимальным значением времени, полученным одним либо несколькими параллельно работающими процессорами.

Для оценки эффективности применения ПЛИС в составе *PIM*-системы рассмотрим отдельные варианты структурной организации *PIM*-систем, показанные на рис. 1 – 4.

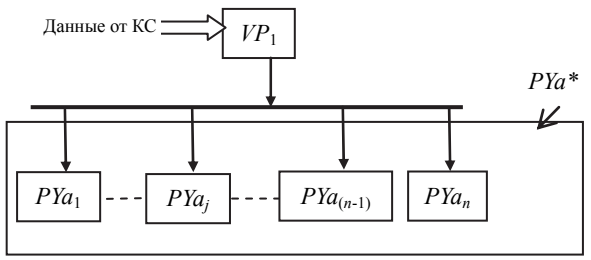


Рис. 1. Упрощенная структурная схема *PIM*-системы

Схема на рис. 2 не представляет особого интереса ее реализации, поскольку процессорные элементы ПЛИС подключены к *PYa** и участвуют в общем процессе распараллеливания аналогично PYa_j , при этом целесообразно, чтобы система команд этих процессорных элементов была аналогична системе команд *PYa*.

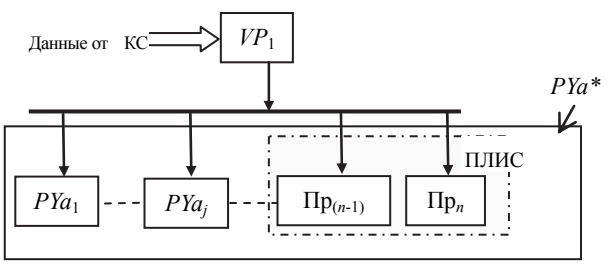


Рис. 2. Схема *PIM*- системы с подключением ПЛИС к *PYa**

Наибольший интерес представляет схема, приведенная на рис. 3, где процессорные элементы ПЛИС используются в качестве *VP*, образуя тем самым линейку *VP** с эквивалентными параметрами по аналогии с эквивалентными параметрами *PYa** при условии, что все процессорные элементы, входящие в *VP**, одинаковы. Тогда система команд эквивалентного *VP** равна системе команд одного процессорного элемента ПЛИС, а эквивалентная емкость памяти равна сумме емкостей памяти, подключенных к каждому процессорному элементу. При этом на данном уровне распределение приложения происходит сначала между эквивалентными *VP** и *PYa**, а затем – внутри линейки *VP** между процессорными элементами этой линейки по аналогии с распределением между процессорными элементами *PYa* внутри *PYa**.

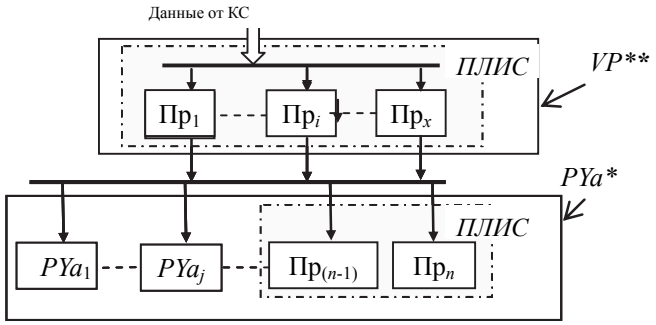


Рис. 3. Схема *PIM*-системы с подключением ПЛИС в качестве *VP* и к *PYa**

Что касается схемы, приведенной на рис. 4, то такая реализация этой схемы наиболее целесообразна, причем возможность реализации *PIM*- системы на ПЛИС показана в работах [1, 3]. Однако сейчас пока отсутствуют ПЛИС, содержащие множество процессоров *VP* с развитой системой команд, а также множество (несколько десятков или даже сотен) процессорных элементов *PYa* (или возможность их построения), при этом логические элементы ПЛИС должны обеспечивать возможность построения соответствующей коммутационной схемы, позволяющей соединять каждый процессорный элемент ПЛИС с *PYa*. Поэтому оценим эффективность применения ПЛИС для схемы, изображенной на рис. 3.

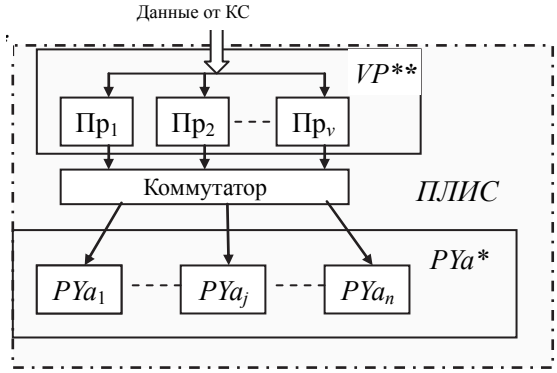


Рис. 4. Схема *PIM*-системы, полностью реализованная на ПЛИС

Оценка эффективности применения ПЛИС при реализации распараллеливания алгоритма приложения для *PIM*-системы

Для этого определим сначала время T_{q1}, необходимое для реализации q-го блока программы с помощью *PIM*-системы без применения ПЛИС (рис. 1). Учитывая, что *PIM*-система содержит множество процессоров *PYa*, которые так или иначе должны выполнять параллельные вычисления, а также п.2 предыдущего подраздела, получим следующее выражение:

$$T_{q1} = F_q \left(\sum_{S=1}^{\delta} t_S + t_{\mu\max} + t_{\text{зад}} + t_{\text{инт}} \right) + T_{q2}, \quad (1)$$

где F_q – общее количество циклов, требуемых для реализации q-го блока программы; t_s – время выполнения s-й операции (*COPs*) q-го блока программы с помощью процессора типа *VP* (рис. 1); δ – общее количество команд *VP*, коды команд которых совпали с кодами операций q-го блока программы; $t_{\mu\max}$ – максимальное время выполнения μ-й операции (*COPs*) q-го блока программы, код которой совпал с кодом команды j-го *PYa* при загрузке всех процессоров *PYa* для их параллельной работы с учетом $\Delta_{PYa\max}$ – максимального разбаланса по времени j-го *PYa* по отношению к другим *PYa* и процессорным элементам ПЛИС, включенным в состав *PYa** (согласно приведенному ранее п.2): $t_{\mu\max} = t_\mu + \Delta_{PYa\max}$; $t_{\text{зад}}$ – время задержки прохождения сигнала через коммутационную схему, соединяющую все *PYa*, образуя линейку *PYa**; T_{q2} – время выполнения последовательной части q-го блока программы, не поддающейся распараллеливанию. Эта часть

блока программы реализуется либо инструментальным компьютером, либо процессором *PIM*-системы, при этом время такой реализации можно оценить по выражению:

$$T_{q2} = \sum_{k=1}^{z} t_k \,, \qquad (2)$$

где t_k – время реализации k-й операции последовательной части q-го блока программы; z – общее количество k-х операций последовательной части q-го блока программы.

Далее получим выражения для применения ПЛИС при реализации q-го блока программы (рис. 3). С учетом (1) и (2) выражение для времени реализации q-го блока программы при применении ПЛИС будет иметь вид:

$$T_{q1/\text{ПЛИС}} = F_q\,(t_{y\,\max} + t_{\mu\max} + t_{\text{зад}} + t_{\text{инт}}) + T_{q2}\,, \qquad (3)$$

где: $t_{y\,\max}$ – максимальное время выполнения кодов операций при параллельной реализации на процессорных элементах ПЛИС (вместо *VP*); $t_{\mu\,\max}$ – максимальное время выполнения μ-м процессором *PYa* (с учетом разбаланса его загрузки, в том числе процессорных элементов ПЛИС в составе *PYa** при параллельной реализации кодов операций q-го блока программы; $t_{\text{инт}}$ – время, необходимое для передачи информации через интерфейс ПЛИС↔*PIM*-система и в обратном направлении.

При этом эффективность Θ_q выполнения q-го блока программы с применением ПЛИС для компьютерной *PIM*-системы, принимая во внимание (1) – (3), можно определить по выражению:

$$\Theta_q = T_{q1} / T_{q1/\text{ПЛИС}} = [F_q(\sum_{S=1}^{\delta} t_S + t_{\mu\max} + t_{\text{зад}} + t_{\text{инт}}) + T_{q2}] / [F_q\,(t_{y\,\max} + t_{\mu\max} + t_{\text{зад}} + t_{\text{инт}}) + T_2]. \qquad (4)$$

Представим выражение (1) в более удобном виде для дальнейших преобразований. Для этого можно считать, что $t_s = t_1$ при $s = 1$, усреднив тем самым время выполнения любой s-й операции (*COPs*) q-го блока программы с помощью процессора типа *VP*. Такое допущение возможно, поскольку нас интересует не столько точная количественная оценка эффективности применения ПЛИС в *PIM*-системе, сколько качественная оценка целесообразности применения ПЛИС. К тому же значения параметра t_1 в совокупности с суммой примерно с такими же значениями других параметров в круглых скобках выражения (4) практически слабо оказывает влияние на конечный результат значения Θ_q, что в дальнейшем подтверждено расчетными графиками, приведенными на рис. 5 – 7. Таким образом, выражение (1) можно представить в виде:

$$T_{q1} = F_q \sum_{S=1}^{\delta-1} t_S + [F_q(t_1 + t_{\mu\,\max} + t_{\text{зад}} + t_{\text{инт}}) + T_{q2}]. \qquad (5)$$

Принимая во внимание приведенные допущения для большей наглядности выражений (5) и соответственно (4) введем следующее обозначение:

$$A = F_q\,(t_1 + t_{\mu\,\max} + t_{\text{зад}} + t_{\text{инт}}) + T_{q2}\,. \qquad (6)$$

Тогда выражение (4) и соответственно (5) будут иметь вид:

$$T_{q1} = F_q \sum_{S=1}^{\delta-1} t_S + A, \qquad (7)$$

$$\Theta_q = T_{q1}/T_{q1/\text{ПЛИС}} = [F_q \sum_{S=1}^{\delta-1} t_S + [F_q(t_1 + t_{\mu\,\max} + t_{\text{зад}} + t_{\text{инт}}) + T_{q2}]] / [F_q(t_1 + t_{\mu\,\max} + t_{\text{зад}} + t_{\text{инт}}) + T_{q2}] = [A + F_q \sum_{S=1}^{\delta-1} t_S\,] \frac{1}{A} = 1 + \frac{1}{A} F_q \sum_{S=1}^{\delta-1} t_S\,. \qquad (8)$$

При выполнении всей программы, содержащей m блоков, составляющих множество $Q = (1, 2, \ldots, q, \ldots, m)$, эффективность применения *PIM*-системы совместно с использованием ПЛИС согласно рис. 3 можно оценить по формуле:

$$\Theta_{\text{пр}} = \sum_{q=1}^{m} \Theta_q = \sum_{q=1}^{m} (1 + \frac{1}{A} F_q \sum_{S=1}^{\delta-1} t_S\,), \qquad (9)$$

где A определяется по выражению (6).

Графики, построенные по расчетным данным выражений (6), (8) и (9), показаны на рис. 5–7. При этом следует обратить внимание на различие значений параметров по оси абсцисс для $\Theta_q = f(m)$ и б). $\Theta_{\text{пр}} = f(m)$, приведенных на рис. 6.

Анализ выражений (8) и (9) показывает, что применение ПЛИС в составе линейки *PYa** (вместо нескольких параллельно работающих *PYa*) *PIM*-системы практически не эффектив-

но. В то же время применение ПЛИС, которая содержит несколько процессорных элементов вместо *VP* оказывается весьма эффективным, при этом эффективность возрастает с увеличением параметра $t_s \approx t_1$.

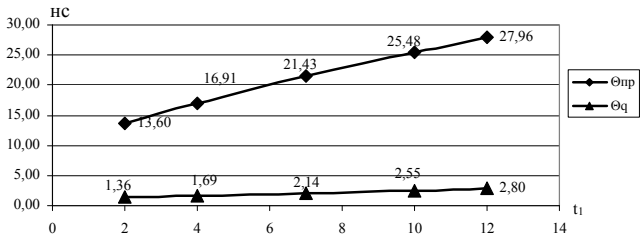
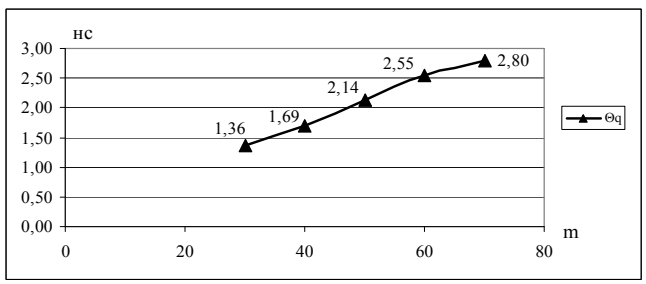
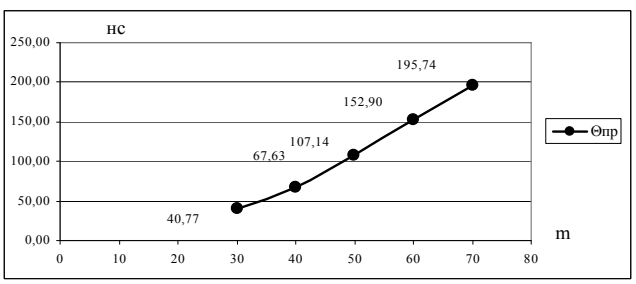


Рис. 5. Графики зависимостей $\Theta_q = f(t_1)$ и $\Theta_{пр} = f(t_1)$



а



б

Рис. 6. Графики зависимостей: *а* – $\Theta_q = f(m)$; *б* – $\Theta_{пр} = f(m)$

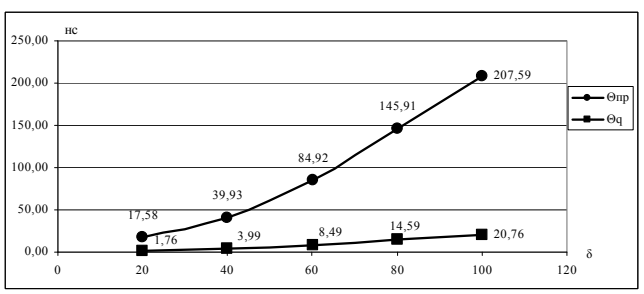


Рис. 7. Графики зависимостей $\Theta_q = f(\delta)$ и $\Theta_{пр} = f(\delta)$

Эффективность применения ПЛИС (вместо *VP*) также существенно увеличивается при увеличении количества блоков программ ***m*** алгоритма реализуемой задачи (рис. 3). Аналогичная картина наблюдается и при увеличении параметра **δ** – общего количество команд *VP*, коды команд которых совпали с кодами операций *q*-го блока программы (рис. 7).

Анализ графиков (рис. 5–7) показывает, что каждый из параметров выражения (6), находящихся в круглых скобках, слабо влияет на значения коэффициентов эффективности Θ_q и $\Theta_{пр}$, изменяя их значения от 1,36 до 2,8 (для Θ_q) и от 13,6 до 28,0 (для $\Theta_{пр}$). Более сильное влияние на конечный результат $\Theta_{пр}$ оказывает изменение количества блоков программ ***m*** алгоритма реализуемой задачи. При исходных значениях параметров от 30 до 70 значения параметра $\Theta_{пр}$ изменяется от 40,77 до 195,7. Аналогично – при изменении параметра δ от 20 до 100 значения параметра Θ_q изменяется от 1,76 до 20,76 в то время, как значения параметра $\Theta_{пр}$ изменяется от 17,58 до 207,59. Отсюда следует, что при распараллеливании алгоритма решаемой задачи, целесообразно оптимизировать размеры *q*-х блоков (параметр δ) реализуемого алгоритма и их количество (параметр *m*), принимая во внимание параллелизм данных.

Дополнительным фактором для уменьшения времени выполнения распараллеленной программы есть исключение из выражений (6) и (9) параметра $t_{инт}$ – задержки по времени при передаче информации через интерфейс между ПЛИС и *PIM*-системой, которые выполнены на отдельных кристаллах, а также уменьшение параметра $t_{зад}$ – времени задержки прохождения сигнала через коммутационную схему, которая соединяет все *PYa* (соответствующие процессоры ПЛИС). Это принципиально возможно, если разместить всю *PIM*-систему на ПЛИС (рис. 4), тем более, что такая возможность в перспективе имеется [1, 3].

Заключение. Таким образом, эффективность применения ПЛИС в составе *PIM*-системы очевидна для ряда архитектурных и структурных решений верхнего уровня распараллеливания, при этом эффективность применения ПЛИС существенно возрастает при увеличении количества анализируемых блоков программ (параметр *m*) и обработки общего количества команд *VP*, коды команд которых сов-

пали с кодами операций q- го блока программы (параметр δ). При этом на нижнем уровне замена процессорных ядер (*PYa*) несколькими процессорными элементами ПЛИС не обеспечивают положительного эффекта, так как параметры процессорных элементов ПЛИС должны иметь идентичность с параметрами *PYa* и участвовать вместе с ними в процессе распараллеливания задачи. Применение процессорных элементов ПЛИС на нижнем уровне распараллеливания может быть целесообразным при крупном масштабировании *PIM*-системы путем увеличения количества *PYa* вместе с процессорными элементами ПЛИС. Однако в этом случае возникает проблема интерфейса между ПЛИС и *PIM*-системой, которые выполнены на отдельных кристаллах, и добавляются соответствующие задержки сигналов при передачи через этот интерфейс информации. Наилучший положительный эффект, как было указано, может быть достигнут при размещении всей *PIM*-системы на ПЛИС.

1. *Яковлев Ю.С.*, *Комухаев Є.И.* Развитие элементной базы и соответствующих архитектурно-структурных решений для систем высокопроизводительных вычислений // УСиМ. – 2013 – № 5. – С. 38–45.
2. *Описание* FPGA-суперкомпьютера Maxwell. – *http://parallel. ru/FPGA/maxwell.html*
3. *Яковлев Ю.С.*, *Елисеева Е.В.* Применение ПЛИС для создания высокопроизводительных вычислительных систем и их компонентов // Математичні машини і системи. – 2014. – № 1. С. 22–35.
4. *Левин И.И.* Семейство высокопроизводительных реконфигурируемых вычислительных систем // Вестн. Томского гос. ун-та: Управление, вычислительная техника и информатика. – 2008.– № 2 (3). – С. 77–93.
5. *Яковлев Ю.С.*, *Елисеева Е.В.* Применение ПЛИС для схемной реализации модулей программы распределения приложений для *PIM*-системы // Математичні машини і системи. – 2015. – № 2. – С. 28–39.
6. *Гергель В.П.*, *Фурсов В.А.* Лекции по параллельным вычислениям // Самара: Изд-во СГАУ. 2009. – 166 с.
7. *Гергель В.П.* Теория и практика параллельных вычислений: Учеб. пособие. – М.: Интернет-Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2007. – 423 с.
8. *Яковлев Ю.С.* Однокристальные компьютерные системы высокой производительности. Особенности архитектурно-структурной организации и внутренних процессов. – Винница: ВНТУ, 2009. – 294 с.
9. *Яковлев Ю.С.*, *Елисеева Е.В.* О реализации распределения приложения для параллельного выполнения на *PIM*-системе // Інформаційні технології та комп'ютерна інженерія. – 2011 – № 3. – С. 53–59.

Поступила 09.04.2015

Тел. для справок: +38 044 526-3207, 292-3185 (Киев)

UDC 004.27; 004.274; 004.315

Yu.S. Yakovlev

About the Estimation of Efficiency of Application the PLIS in Structure PIM-System

General provisions of multi sequencing of the application with reference to PIM-system with application the PLIS in correspondence with the formulated rules are observed. Three alternatives of block diagrammes of the PIM-systems using the PLIS are offered:

a) processor units supplement the PLIS for PYa;

b) processor units the PLIS are used instead of VP and supplement PYa;

c) the PIM-system is completely realised on the PLIS.

For alternative *b* expressions are gained and numerical estimations of efficiency of application the PLIS are fulfilled. On these expressions schedules of dependences of a time of implementation of

q^{th} block of the program Θ_q and all program $\Theta_{\text{пр}}$ from execution time of t_s s^{th} operation (COPs) q^{th} block of the program by means of the processor of type VP, total of blocks of the program m and parametre δ – total of VP commands which codes of commands coincide with codes of operations q^{th} the program block are builted. It is shown, that efficiency of application the PLIS (instead of VP) essentially increases at parametre increase m, and also at parametre increase δ. It leads to expediency of mutual optimisation of these parametres, taking into consideration parallelism of data. Output is drawn on expediency of performance of all PIM-system on the PLIS, that in addition allows to reduce a time of implementation of parallel algorithm at the expense of delay exclusion at an information transfer through the interface of link the PLIS with PIM-system which are fulfilled on separate chips, and also to reduce signal passage hold time through the diagramme of connections which connects all PYa and appropriate processor units the PLIS.

●