

# КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

*E. Eliseeva*

## **ABOUT MODIFICATION OF THE ALGORITHM USER DISTRIBUTION PROGRAM FOR IMPLEMENTATION ITS ON CPLD**

*It was considered the user program allocation algorithm for PIM-processor system and proposed a modification of the algorithm modules for further implementation on CPLD.*

*Key words: PIM-system, CPLD, scheduling algorithm.*

*Розглянуто алгоритм розподілу програми користувача по процесорам PIM-системи і запропонована модифікація модулів даного алгоритму з метою подальшої реалізації на ПЛІС.*

*Ключові слова: PIM-система, ПЛІС, алгоритм розподілу.*

*Рассмотрен алгоритм распределения программы пользователя по процессорам PIM-системы и предложена модификация модулей данного алгоритма для дальнейшей реализации на ПЛИС.*

*Ключевые слова: PIM-система, ПЛИС, алгоритм распределения.*

© Е.В. Елисеева, 2016

УДК 004.2; 004.272.43

Е.В. ЕЛИСЕЕВА

## **О МОДИФИКАЦИИ АЛГОРИТМА РАСПРЕДЕЛЕНИЯ ПРОГРАММЫ ПОЛЬЗОВАТЕЛЯ ДЛЯ РЕАЛИЗАЦИИ ЕГО НА ПЛИС**

Предпосылками для оптимизации алгоритма посредством реализации отдельных его частей на ПЛИС являются: многочисленное выполнение одних и тех же операций над множеством данных; возможность выделить в алгоритме отдельные фрагменты, несвязанные по данным, которые могут выполняться независимо друг от друга.

Для оптимизации рассмотрен алгоритм распределения блоков программы пользователя по процессорам PIM-системы [1–5] и выделены ряд фрагментов алгоритма, которые можно модифицировать для последующей реализации на ПЛИС.

В частности модуль для оценки веса блоков содержит многократные обращения к системе команд ведущего процессора (СК VP) и системе команд процессорного ядра (СК PУA) с целью найти совпадение кода операции из блока программы пользователя и кода команды из СК VP и СК PУA для вычисления времени выполнения операции или, чтобы сделать вывод о невозможности выполнения этой операции на некотором процессоре системы. С другой стороны информация о СК VP и СК PУA хранится отдельно, следовательно, такой поиск в каждой из них можно организовать независимо друг от друга (отдельно для СК PУA и СК VP). Кроме того, поскольку программа пользователя разделена на независимые по данным блоки, т. е. возможность вести оценку веса параллельно сразу для нескольких блоков.

Исходный фрагмент алгоритма распределения приложения по процессорам РІМ-системы (модуль оценки веса каждого блока программы пользователя) приведен в [4].

Укрупненная блок-схема алгоритма работы модуля оценки веса одного блока программы пользователя после модификации с целью параллельной реализации частей данного алгоритма на ПЛИС показана на рис. 1. Закрашенными стрелками отмечены части алгоритма, где осуществляется обмен данными и результатами вычислений с ПЛИС.

Здесь рассмотрен алгоритм оценки веса одного блока программы пользователя. Результатом работы данного алгоритма будет: вес блока для VP, вес для PУа, если возможно выполнение на PУа, и сигналы TEG\_1 и TEG\_2, которые необходимы для работы последующих модулей программы распределения. Их возможные значения приведены в табл. 1.

Для каждого блока программы пользователя предварительно проводится лексический анализ содержимого блока, он заключается в посимвольном считывании кода и выделении лексем (последовательностей допустимых символов, понятных компилятору языка программирования). В результате лексического анализа выдается список токенов, входящих в данный блок. Токены представляются в виде структуры, состоящей из кода токена, указывающего на тип лексемы (например, число, операция и т. д.) и самой лексемы. Осуществляется просмотр всего списка токенов, и для каждого токена выполняется проверка, соответствует ли этот токен какой-либо операции.

Если рассматриваемый токен соответствует операции, то на входы ПЛИС передается значение кода этой операции COPS\_BPr\_q(7:0) и выполняются подпрограммы А (рис. 2) или В (рис. 3), в зависимости от значения flagPУа, который указывает может ли рассматриваемый блок программы пользователя выполниться на PУа. Изначально, flagPУа равен 1, и сбрасывается в 0 в подпрограмме В сразу, как только будет найдена хотя бы одна операция, код которой отсутствует СК PУа и, следовательно, она не может быть выполнена на PУа. Это значит, что и весь блок программы пользователя, содержащий данную операцию, не может выполняться на PУа, поэтому нет смысла проводить поиск для остальных операций из блока в СК PУа.

Заметим, что при реализации алгоритма предполагалось, что СК PУа является подмножеством СК VP, поскольку в силу архитектурных особенностей РІМ-системы PУа имеют урезанную систему команд по сравнению с VP, и программа пользователя не содержит «недопустимых» операций, т. е. тех, которые не могут быть выполнены на VP.

В подпрограмме А осуществляется поиск соответствующей команды в СК VP, на ПЛИС при этом задействуется для сравнения только канал VP из BLOCK\_ANALIZ [5]. Если найдено совпадение кодов операции из блока программы пользователя и кода команды из СК VP, то в этом случае начинает работу блок суммирования BLOCK\_SUM\_T\_VP/PУа на ПЛИС [5], который перемножает полученные значения Tr\_VP\_sq(7:0) (вес команды) и Repit\_Ts\_BPr\_VP\_q(7:0) (количество повторений), и накапливает результат. Переход к следующим шагам ал-

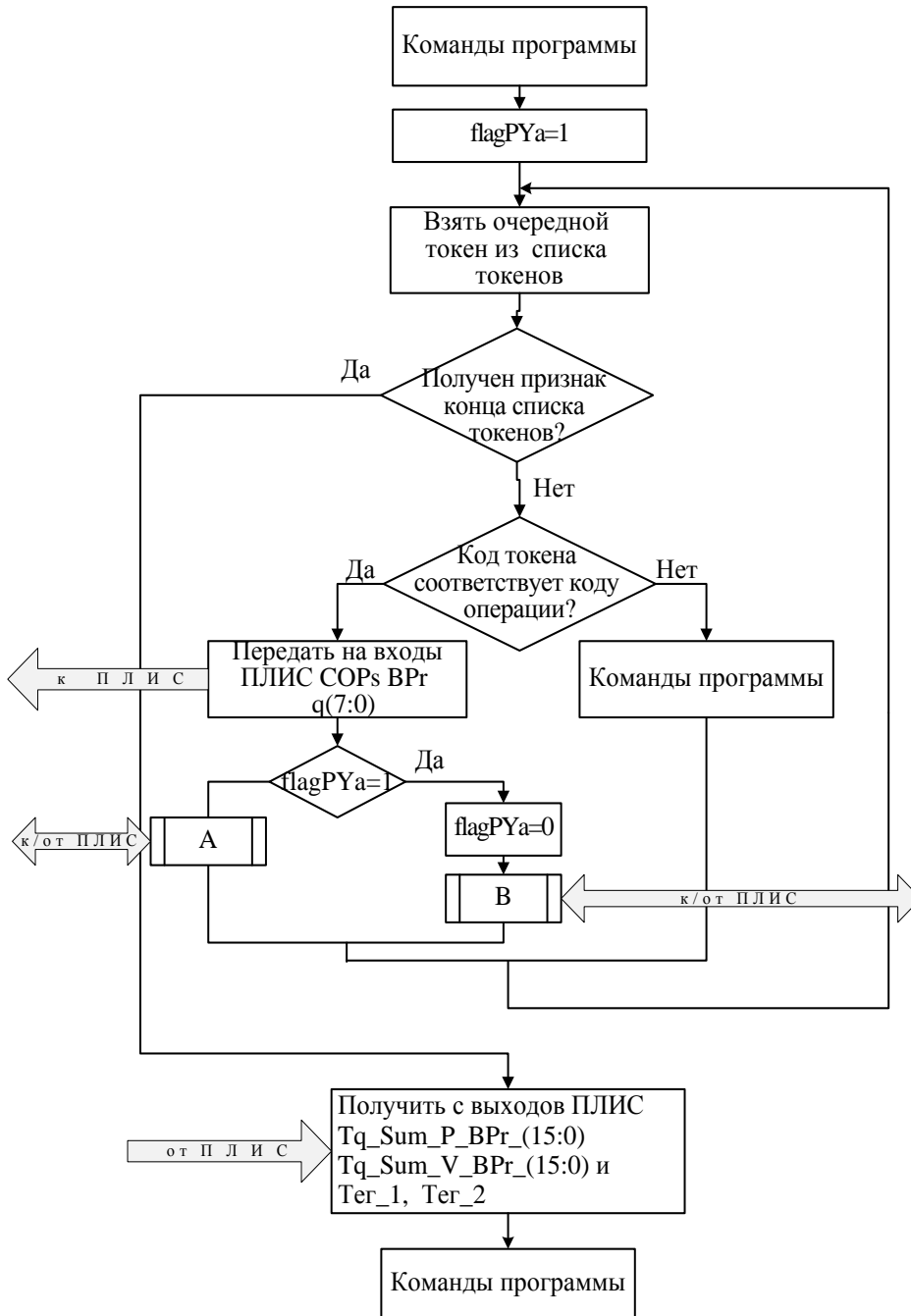


РИС. 1. Укрупненная блок-схема модуля оценки веса модифицированного алгоритма распределения блоков программы пользователя по процессорам PIM-системы с указанием входов и выходов на ПЛИС

ТАБЛИЦА 1. Значения TEG\_1 и TEG\_2

Обозначение	$T(PYa) < T(VP)$	$T(PYa) > T(VP)$	$T(PYa) = T(VP)$
Ter_1	0	1	1
Ter_2	1	0	1

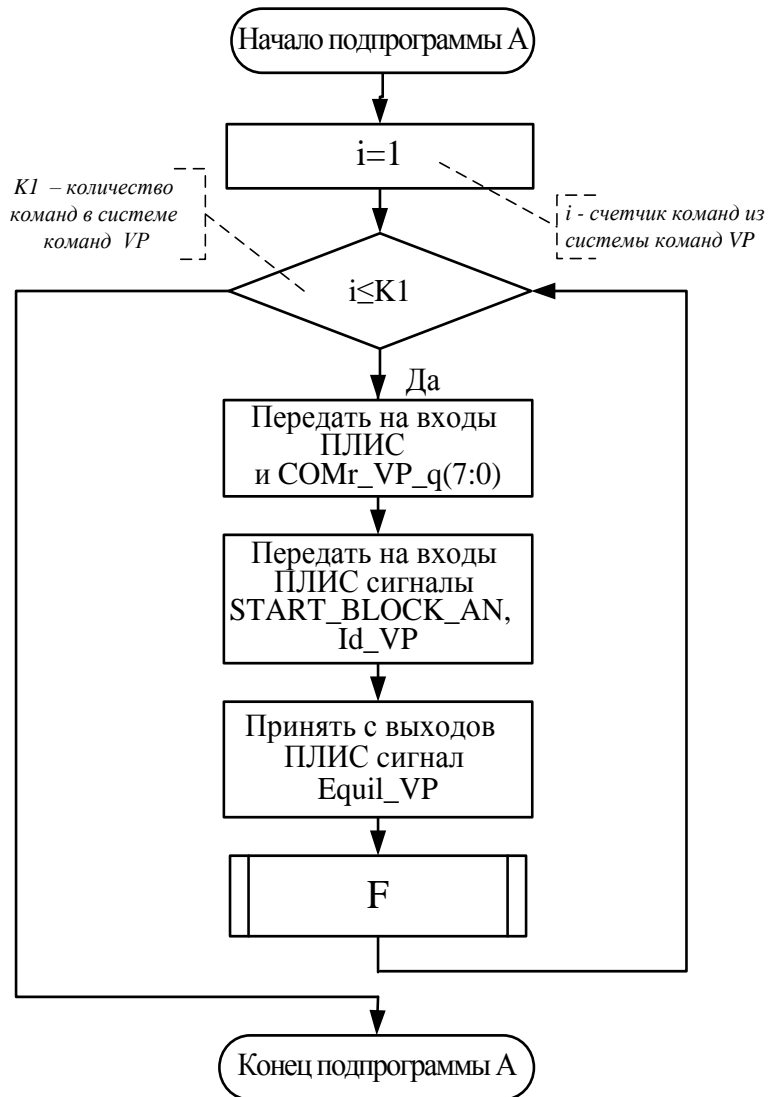


РИС. 2. Подпрограмма А. Определение веса для VP очередной операции в блоке для случая, если блок может выполняться только на VP

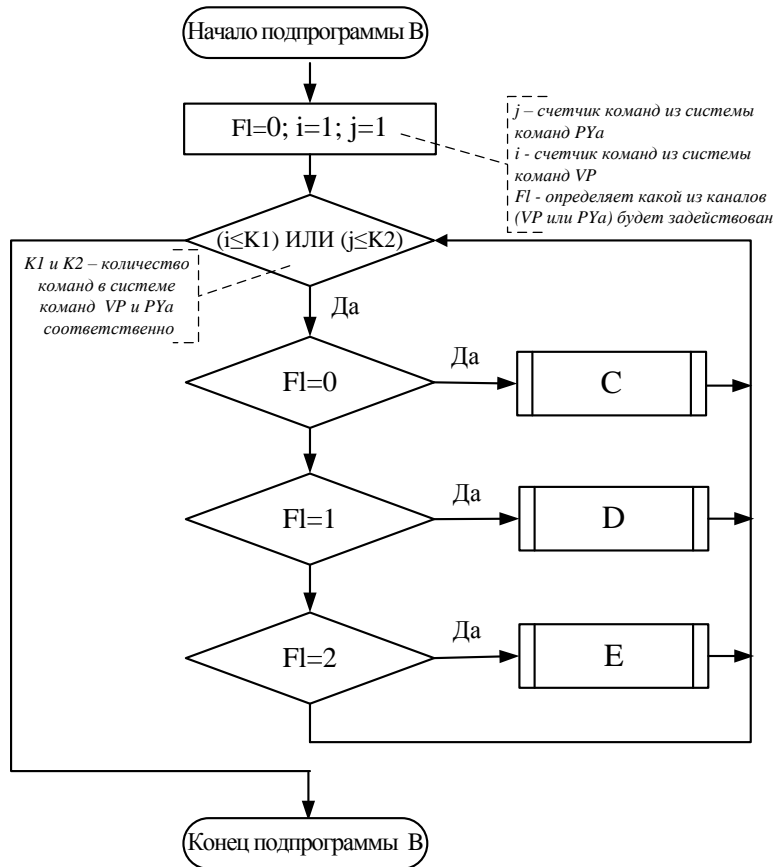


РИС. 3. Подпрограмма В. Поиск соответствия в системах команд соответствующих процессоров и определение веса операции для РУа и VP

горитма происходит без ожидания результата выполнения умножения с накоплением от ПЛИС, тем самым обеспечивается возможность параллельной работы программы оценки веса и вычислений на ПЛИС.

Подпрограмма В будет выполняться в случае, если  $\text{flagPYa}=1$ , т. е. если все рассмотренные ранее операции в блоке могут быть выполнены на РУа или же рассматривается самая первая операция в блоке. Просмотр СК VP, СК РУа и поиск в них совпадении кодов операций блока программы пользователя с кодами команд из СК VP и СК РУа осуществляется в цикле, выход из которого возможен только тогда, когда будут найдены совпадения для обоих процессоров или только для VP, но при этом будет установлена невозможность выполнения некоторой операции на РУа. Здесь возможны следующие варианты: 1) вес рассматриваемой операции еще не оценен ни для одного из процессоров, в этом случае параллельно на ПЛИС можно осуществить сравнение кода операции с очередными кодами команд из СК VP и СК РУа, т. е. возможна параллельная

работа на ПЛИС двух каналов VP и PYa; 2) вес рассматриваемой операции уже оценен для одного из процессоров, тогда поиск совпадений осуществляется только в системе команд другого процессора. Чтобы определить какой из вариантов необходимо выбрать в цикле используется значение флага FI, который можно получить в зависимости от значений сигналов Equal\_VP и Equal\_PYa (табл. 2).

ТАБЛИЦА 2. Значения флага FI в зависимости от значений Equal\_PYa и Equal\_VP

Equal_VP	Equal_PYa	FI = Equal_VP << 1 + Equal_PYa
0	0	0
0	1	1
1	0	2
1	1	3

Если значение FI = 0, это значит, что для оцениваемой операции из блока программы пользователя не найдено пока совпадений ни в СК VP, ни в СК PYa. В этом случае управление переходит к подпрограмме С, в которой параллельно задействуются каналы VP и PYa на ПЛИС для поиска в СК PYa и СК VP.

При необходимости задействуется только один канал PYa или VP. Это происходит в том случае, если блок вообще не может быть выполнен на PYa или уже получен вес команды для одного из процессоров во время предыдущей итерации цикла, показанного на рис. 3. Таким образом, если значение флага FI равно 1 или 2, то выполняется одна из подпрограмм D или E соответственно.

В каждой из подпрограмм С, D, E вычисляется новое значение флага FI, чтобы иметь возможность использовать его при следующей итерации цикла, показанного на рис. 3.

В каждой из подпрограмм С, D, E в зависимости от результатов, полученных с ПЛИС, проводится анализ сигналов Equal\_VP и Equal\_PYa. Значение Equal\_VP показывает было ли совпадение кода операции из блока программы пользователя с кодом рассматриваемой команды из СК\_VP. Аналогично, значение Equal\_PYa показывает было ли совпадение кода операции с кодом рассматриваемой команды из СК\_PYa.

Если значение Equal\_VP или Equal\_PYa равно нулю, то совпадения для СК\_VP или СК\_PYa соответственно не было. В этом случае увеличивается на единицу значение соответствующего счетчика *i* (счетчик команд из СК\_VP) или *j* (счетчик команд из СК\_PYa). Значения этих счетчиков используются в качестве критериев для выполнения итераций в цикле, который представлен в подпрограмме В на рис. 3.

Если значение Equal\_VP или Equal\_PYa равно единице, то совпадение в СК\_VP или СК\_PYa имело место и рассматриваемой операции присваивается такой же вес для VP или PYa, как и вес соответствующей ей команды из СК\_VP или СК\_PYa в зависимости от того, где было найдено совпадение. На ПЛИС передаются сигналы запускающие вычисление суммы с накоплением

(добавление веса соответствующей операции к весу всего блока) для разных процессоров. На ПЛИС так же передаются значения найденного веса для VP –  $Tr\_VP\_sq(7:0)$  или же  $Tj\_PYa\_sq(7:0)$ , если речь идет о весе для PYa и значение  $Repit\_Ts\_BPr\_VP(PYa)\_q(7:0)$ , которое соответствует количеству повторений данной операции в рассматриваемом блоке программы пользователя.

При этом не ожидается возвращение результата с ПЛИС и могут выполняться последующие команды алгоритма. Результат в виде суммы с накоплением будет затребован с ПЛИС позже, когда будут рассмотрены и оценены все операции из заданного блока программы пользователя (по завершению работы цикла рис. 3). В этом случае результатом будут значения веса (времени выполнения) заданного блока для различных процессоров VP и PYa – на рис. 1 эти значения обозначены соответственно  $Tq\_Sum\_P\_BPr(15:0)$  и  $Tq\_Sum\_V\_BPr(15:0)$ , что позволяет убрать из алгоритма, приведенного в [4] часть связанную с вычислением значений веса всего блока для VP и PYa. Значения TEG\_1 и TEG\_2, возвращаемые с ПЛИС, дают возможность оценить вес блока для VP и PYa не прибегая к дополнительным сравнениям этих весов при распределении блоков программы пользователя по процессорам PIM-системы, т. е. сокращают время работы последующих модулей алгоритма распределения программы пользователя, который рассмотрен в [4].

Полученная модификация алгоритма оценки веса блока программы пользователя с реализацией отдельных его фрагментов на ПЛИС дает ускорение оценки веса блока программы пользователя приблизительно в 1,5–2 раза (в зависимости от содержимого блока) за счет параллельного выполнения некоторых вычислений на ПЛИС, возможности параллельного выполнения команд программы распределения и вычислений на ПЛИС, а также за счет переноса части вычислений на ПЛИС, где они выполняются быстрее.

1. 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.
2. Пат. на винахід 103535 Україна, МПК G06F 9/44, G06F 9/45. Спосіб розподілу програми користувача для комп'ютерної системи / Сергієнко І.В., Палагін О.В., Боюн В.П., Яковлев Ю.С., Єлісєєва О.В.; Інститут кібернетики імені В.М. Глушкова НАН України; заявл. 03.01.2012; опубл. 25.10.2013, Бюл. № 20.
3. Пат.на винахід 104039 Україна, МПК G06F 9/44, G06F 9/45. Система для розподілу програми користувача / Сергієнко І. В. Палагін О. В., Боюн В. П., Яковлев Ю. С., Єлісєєва О. В.; Інститут кібернетики ім. В.М. Глушкова НАН України; заявл. 27.02.2012; опубл. 25.12.2013, Бюл. № 24.
4. Елисеева Е.В. О распараллеливании пользовательских задач в распределенных компьютерных системах типа “процессор-в-памяти”. *Математические системы и машины*. 2010. № 4. С. 68–81.
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.

Получено 01.11.2016