УДК 004.272.43

А.В. Бовкун

АВТОМАТИЗИРОВАННЫЕ МЕТОДЫ ПОВЫШЕНИЯ УДЕЛЬНОЙ ПРОИЗВОДИТЕЛЬНОСТИ ПРИКЛАДНЫХ ЗАДАЧ ДЛЯ РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ*

Описаны методы, сокращающие, при комплексном применении, на 5–30 % используемый аппаратный ресурс программируемых логических интегральных схем (ПЛИС). Применение рассмотренных методов повышает удельную производительность решаемой прикладной задачи. Предложен также метод замещения одного аппаратного ресурса ПЛИС другим, позволяющий, при сохранении удельной производительности задачи, сократить время получения прикладного решения. Автоматизированное применение описанных методов позволяет повысить удельную производительность прикладных задач различных предметных областей без участия специалиста-схемотехника.

Реконфигурируемый вычислитель; реконфигурируемая вычислительная система; суперкомпьютер; удельная производительность; редуцирование задержек.

A.V. Bovkun

COMPUTER-AIDED METHODS OF INCREASING OF APPLICATIONS SPECIFIC PERFORMANCE FOR RECONFIGURABLE COMPUTER SYSTEMS

The paper covers methods that provide reduction of FPGA hardware resource by 5-30%. Utilization of the considered methods increases specific performance of the developed solutions of an applied task. The method of FPGA hardware resource replacement reduces time for generating solution of the applied task and keeps the value of specific performance unchangeable. Computer-aided usage of the considered methods provides increasing of specific performance of applied tasks from different fields without any support of circuit engineer.

Reconfigurable computer; reconfigurable computer system; supercomputer; specific performance; delay reduction.

Обеспечение комплексной безопасности сложных технических систем в режиме реального времени характеризуется высокой вычислительной трудоёмкостью, требующей применения многопроцессорных вычислительных систем. Отличительными признаками задач этой области являются большое число битовых операций обработки данных и сильная связность, когда число информационных обменов между функциональными устройствами соизмеримо с числом вычислительных операций. При решении сильносвязанных задач на наиболее распространенных в настоящее время кластерных системах их производительность не превышает 10 % от пиковой, поскольку большие временные расходы на информационный обмен между узлами кластера приводят к тому, что при увеличении числа узлов, задействованных для решения сильносвязанной задачи, наблюдается даже падение производительности [1]. Эффективное решение сильносвязанных задач возможно в случае соответствия вычислительной архитектуры системы информационному графу решаемой задачи, что обеспечивается в концепции реконфигурируемых вычислительных систем (РВС), построенных на программируемых логических интегральных схемах (ПЛИС) [2].

* Исследования выполнены при финансовой поддержке Министерства образования и науки РФ, госконтракт № 14.527.12.0004 от 03 октября 2011 г.

Структурная реализация прикладных задач, характерная для РВС, позволяет достигать высокой удельной производительности, под которой понимается отношение производительности вычислительного устройства к количеству оборудования, необходимого для аппаратной реализации этого устройства. Повышение удельной производительности возможно либо при увеличении производительности прикладной задачи (за счет улучшения параметров её реализации), либо при сокращении аппаратных затрат на реализацию вычислительного устройства при сохранении его производительности. Оптимизация структурной реализации прикладной задачи для повышения удельной производительности выполняется вручную специалистом-схемотехником. Такой подход требует больших временных затрат, поскольку стандартные средства программирования ПЛИС (Xilinx ISE, Quartus II, Libero IDE, Altium Designer и др.) не содержат средств структурной оптимизации, позволяющих сократить используемый аппаратный ресурс при сохранении производительности вычислительной системы.

Таким образом, актуальной является автоматизация методов повышения удельной производительности прикладных задач для РВС, позволяющих сократить время получения эффективного структурного решения.

Методы повышения удельной производительности вычислительных устройств, как правило, связаны с повышением частоты работы устройства и оптимизацией его внутренней структуры [3]. Методы сокращения аппаратных затрат позволяют сократить число эквивалентных вентилей, затрачиваемых на реализацию вычислительного устройства, за счет уменьшения числа выполняемых операций [4] и снижения непродуктивных расходов на синхронизацию информационных потоков [5].

Процесс разработки решения прикладной задачи для РВС состоит из следующих этапов:

1) разработка алгоритма решения прикладной задачи;

2) разработка библиотеки схемотехнических блоков, реализующих необходимые математические или программные модели;

3) формирование информационного графа задачи из готовых схемотехнических блоков;

4) разбиение информационного графа на подграфы, каждый из которых структурно реализуется в отдельном кристалле ПЛИС;

5) синхронизация потоков в каждом отдельном подграфе информационного графа задачи и между ними;

6) формирование проектов для каждого задействованного кристалла ПЛИС;

7) формирование файлов временных и топологических ограничений;

8) создание конфигурационных файлов ПЛИС;

9) создание управляющей программы.

Для согласования информационных потоков строится подсистема синхронизации, которая может занимать аппаратный ресурс, соизмеримый с ресурсом самой задачи. При использовании средств автоматической синхронизации информационных потоков подсистема синхронизации, как правило, является неоптимальной. Автоматическая оптимизация подсистемы синхронизации позволит без участия специалиста-схемотехника сократить используемый аппаратный ресурс и повысить удельную производительность РВС на основе ПЛИС.

Для автоматизированного сокращения используемого аппаратного ресурса могут использоваться следующие методы:

- метод эквивалентных преобразований;
- метод поглощения;
- метод высвобождения критического аппаратного ресурса;
- метод ассоциативной перекоммутации.

Метод эквивалентных преобразований заключается в перекоммутации информационных потоков с сохранением информационной эквивалентности.

На рис. 1 проиллюстрированы результаты работы данного метода. На рис. 1,а представлен фрагмент исходного графа, на рис. 1,б – тот же фрагмент после применения одной итерации эквивалентного преобразования, а на рис. 1,в – фрагмент графа после применения двух итераций эквивалентного преобразования. Кружочками обозначены операционные вершины графа, а прямоугольниками – элементы, реализующие задержку цифровых информационных потоков для синхронизации потоков данных, m_1, m_2, m_3 – количество тактов, на которые задерживается поток данных.

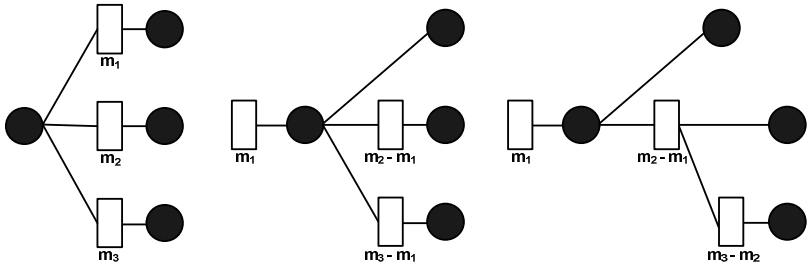


Рис. 1. Пример эквивалентных преобразований: а – фрагмент исходного графа; б – фрагмент графа после одной итерации эквивалентного преобразования; в – фрагмент графа после двух итераций эквивалентного преобразования

Предположим, что $m_3 > m_2 > m_1$. Тогда применение метода эквивалентных преобразований для данного фрагмента дает выигрыш регистров, описываемый следующей формулой:

$$P = (m_1 + m_2 + m_3) - (m_1 + m_2 - m_1 + m_3 - m_1) = 2m_1, \quad (1)$$

где m_1, m_2, m_3 – количество регистров синхронизации.

Для случая n вершин выигрыш по регистрам можно оценить

$$P = (n-1)m_1, \quad (2)$$

где n – количество операционных вершин, участвующих в редукции.

Учитывая, что информационные потоки обычно являются многоразрядными, выигрыш составит:

$$P = (n-1) \cdot m_1 \cdot R, \quad (3)$$

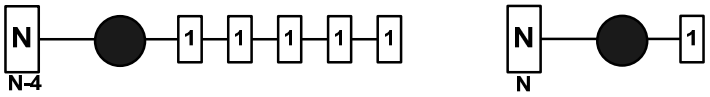
где R – разрядность шины данных.

Автоматизированное применение этого метода для решения задач обеспечения безопасности сложных технических систем в режиме реального времени позволяет сократить используемый аппаратный ресурс на 5–20 %.

Метод поглощения заключается в максимальном использовании программируемых элементов задержки цифровых информационных потоков данных. При создании подсистемы синхронизации на кристалле ПЛИС используются как «элементарные» задержки (задерживающие информационный поток на 1 такт), так и более сложные – программируемые элементы, способные задерживать информационный поток данных на произвольную величину от 1-го до 16 (32)-ти тактов. Максимальную величину программируемой задержки обозначим N.

В процессе оптимизации подсистемы синхронизации нередко возникает ситуация, когда можно осуществить «поглощение» регистров синхронизации за счет программируемых элементов, величина которых меньше их максимальной вели-

чины. Пример поглощения представлен на рис. 2. На рис. 2,а представлена операционная вершина, входной информационный поток которой задерживается программируемым элементом задержки. Величина задержки на данном элементе на 4 меньше максимально возможной величины. Выходной информационный поток проходит через каскад элементарных задержек. В данном случае программируемый элемент задержки может «поглотить» 4 элементарных задержки (рис. 2,б), что приведет к сокращению аппаратного ресурса.



а б

Рис. 2. Сокращение задержек методом поглощения: а – до поглощения; б – после поглощения

Возможно поглощение без сокращения аппаратного ресурса. Пример подобного поглощения приведен на рис. 3. Как видно из рис. 3,б, применение метода поглощения не привело к сокращению аппаратного ресурса, однако подобное преобразование позволяет воспользоваться эвристическим методом «высвобождения критического аппаратного ресурса» ПЛИС.

а б

Рис. 3. Поглощение, не приводящее к сокращению аппаратного ресурса: а – исходный фрагмент информационного графа; б – фрагмент графа после поглощения

Как было показано на рис. 3, применение метода поглощения не гарантирует сокращение аппаратного ресурса, необходимого для построения подсистемы синхронизации.

Автоматизированное применение этого метода для решения задач обеспечения безопасности сложных технических систем в режиме реального времени позволяет сократить используемый аппаратный ресурс на 0–2 %.

В процессе решения прикладной задачи может возникнуть ситуация, когда после применения методов оптимизации не хватает аппаратного ресурса определенного вида для отображения подграфа в кристалл ПЛИС. Это означает, что необходимо заново решать задачу разбиения информационного графа, изменив коэффициент максимального заполнения кристалла ПЛИС. После решения задачи разбиения нужно повторить все этапы разработки решения прикладной задачи, следующие за этапом разбиения. Однако, если превышение аппаратного ресурса незначительно (порядка 0,1–0,5 % данного ресурса в кристалле ПЛИС) и в структурной реализации решения прикладной задачи имеются схемотехнические элементы, реализованные на данном аппаратном ресурсе, то можно высвободить часть необходимого ресурса, заменив имеющиеся схемотехнические элементы эквивалентными элементами, реализованными на другом аппаратном ресурсе.

Анализ занимаемого аппаратного ресурса при использовании задержек различной аппаратной реализации не требует больших временных затрат. Поэтому целесообразно производить данный анализ в тех случаях, если происходит незна-

чительное превышение одного из аппаратных ресурсов ПЛИС. Применение данного метода не позволяет увеличить удельную производительность ПЛИС, однако позволяет в некоторых случаях сократить время на получение решения прикладной задачи.

Автоматизированное применение этого метода для решения задач обеспечения безопасности сложных технических систем в режиме реального времени не позволяет сократить аппаратный ресурс, однако данный метод позволяет сократить время получения решения прикладной задачи.

Метод ассоциативной перекоммутации основан на перекоммутации информационных потоков на смежных вычислительных блоках, реализующих ассоциативные операции (например, целочисленное сложение).

Пусть необходимо реализовать следующую рекурсивную функцию: $a_i = a_{i-1} + b + c, i = 1...N-1$. Вычислительная структура, реализующая данную функцию, состоит из n вычислительных блоков G, каждый из которых состоит из двух сумматоров, обладающих одинаковой задержкой D. При конвейеризации вычислительной структуры необходимо выполнять синхронизацию информационных потоков (рис. 4).

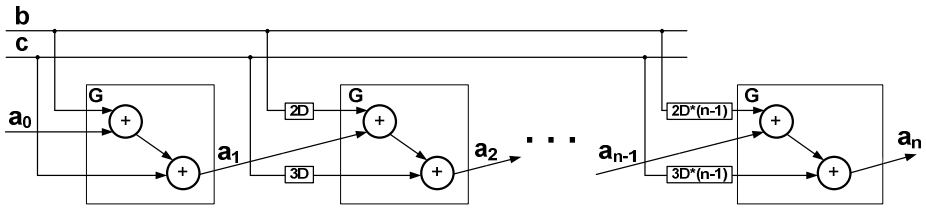


Рис. 4. Вычислительная структура

Если просуммировать величины используемых задержек, то получим следующее выражение:

$$T = 5D \cdot n\left(\frac{n-1}{2}\right). \quad (4)$$

Если к данной структурной реализации вычислительного конвейера применить метод эквивалентных преобразований, то получится структурная реализация, представленная на рис. 5.

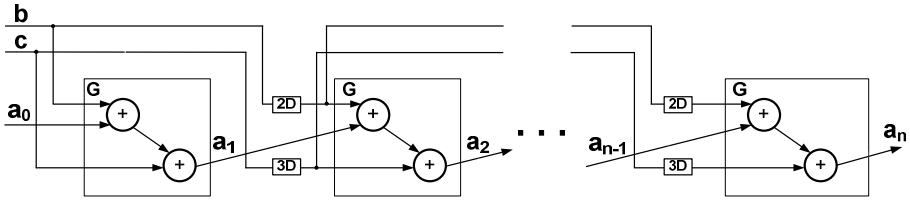


Рис. 5. Вычислительная структура после применения метода эквивалентных преобразований

Нетрудно подсчитать сумму задержек, требуемых для данной реализации:

$$T = 5D(n-1). \quad (5)$$

Если к структурной реализации, изображенной на рис. 4, применить метод ассоциативной перекоммутации, то получим следующий вариант структурной реализации (рис. 6).

Как видно из рис. 6, сумма задержек выражается формулой

$$T = 2D \cdot n\left(\frac{n-1}{2}\right). \qquad (6)$$

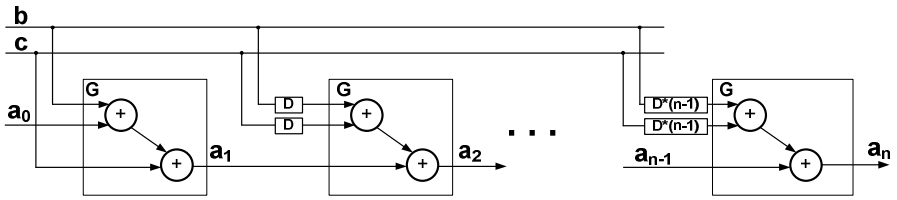


Рис. 6. Вычислительная структура после ассоциативной перекоммутации

Таким образом, при сравнении формул (4) и (6) можно заметить, что применение метода ассоциативных перекоммутаций дает выигрыш в сокращении аппаратных расходов на реализацию $3Dn\left(\frac{n-1}{2}\right)$ задержек.

Если же данный метод применить к структурной реализации, изображенной на рис. 5, то получится структурная реализация, представленная на рис. 7.

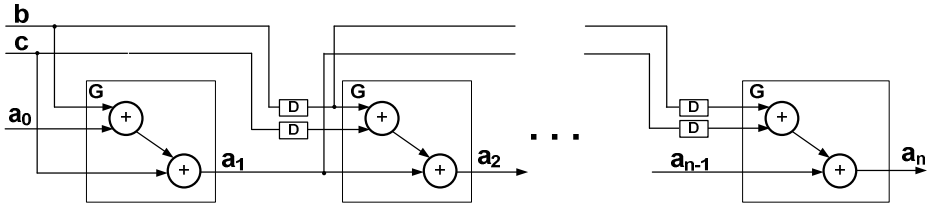


Рис. 7. Вычислительная структура после применения двух методов оптимизации

Как видно из рис. 7, сумма задержек выражается формулой

$$T = 2D(n-1). \qquad (7)$$

Если сравнить формулы (5) и (7), то легко заметить, что применение метода ассоциативных перекоммутаций дает выигрыш в сокращении аппаратных расходов на реализацию $3D(n-1)$ задержек.

Автоматизированное применение этого метода для решения задач обеспечения безопасности сложных технических систем в режиме реального времени позволяет сократить используемый аппаратный ресурс на 5–10 %.

Автоматизированное применение приведенных методов позволит без участия специалиста-схемотехника сократить используемый аппаратный ресурс ПЛИС, что приведет к увеличению удельной производительности РВС. Применение описанных методов на ряде тестов из области обеспечения комплексной безопасности сложных технических систем в режиме реального времени привело к сокращению используемого аппаратного ресурса ПЛИС на 5–30 % без потери производительности системы при решении прикладной задачи по сравнению с использованием существующих средств автоматической синхронизации. В настоящее время описанные методы внедряются в разработанный в НИИ многопроцессорных вычислительных систем ЮФУ им. А.В. Каляева многокристальный синтезатор прикладных структурных программ Fire!Constructor [6].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления / Под ред. В.В. Воеводина. – СПб.: БХВ-Петербург, 2002. – 599 с.
2. *Зотов В.Ю.* Проектирование цифровых устройств на основе ПЛИС фирмы XILINX в САПР WebPACK ISE. – М.: Горячая линия-Телеком, 2003. – 624 с.
3. *Сорокин Д.А., Дордопуло А.И., Бовкун А.В.* Аппаратная реализация докинга лигандов на реконфигурируемых вычислительных системах // Информатика, вычислительная техника и инженерное образование. – 2011. – Вып. 4(6). – С. 30-46. – Эл. № ФС77-39729 от «29» апреля 2010 г. http://digital-mag.tti.sfedu.ru.
4. *Сорокин Д.А., Левин И.И., Дордопуло А.И., Мельников А.К.* Решение задач с существенно-переменной интенсивностью потоков данных на реконфигурируемых вычислительных системах // Вестник компьютерных и информационных технологий. – М.: Машиностроение, 2012. – № 2. – С. 24.
5. *Доронченко Ю.И.* Организация эффективных вычислений для реконфигурируемых вычислительных систем на основе ПЛИС // Известия ТРТУ. – 2006. – № 16 (71). – С. 11-16.
6. *Гуленок А.А.* Методы и алгоритмы отображения графов задач на реконфигурируемые вычислительные системы // Вестник компьютерных и информационных технологий. – М.: Машиностроение, 2011. – № 6. – С. 3-11.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ромм.

Бовкун Александр Викторович – Научно-исследовательский институт многопроцессорных вычислительных систем им. А.В. Каляева Южного федерального университета; e-mail: simans2002@mail.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 88634315491; младший научный сотрудник.

Bovkun Alexandr Viktorovich – Scientific-Research Institute Multiprocessing Computing Systems after Kalyaev of South Federal University; e-mail: simans2002@mail.ru; 2, Chekhov street, Taganrog, 347928, Russia; phone: +78634315491; research assistant.

УДК 004.421.6

А.Г. Коваленко

МАКРОКОНВЕЙЕРНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ АУТЕНТИФИКАЦИИ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ COLAMO*

Рассматривается макроконвейерная реализация алгоритмов аутентификации для подсистем безопасности сложных информационных комплексов на реконфигурируемых вычислительных системах с динамически перестраиваемой архитектурой. Для обобщенной схемы макроконвейерных задач сформулированы условия их эффективной реализации на реконфигурируемых вычислительных системах. Рассмотрена реализация типового алгоритма аутентификации SSL на языке программирования высокого уровня COLAMO. Приведены достигнутые характеристики производительности реконфигурируемой вычислительной системы при решении описанной макроконвейерной задачи.

Макроконвейерные вычисления; реконфигурируемая вычислительная система; алгоритмы аутентификации; язык высокого уровня COLAMO.

* Исследования выполнены при финансовой поддержке Министерства образования и науки РФ, госконтракт № 14.527.12.0004 от 03 октября 2011 г.