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.

A.G. Kovalenko

MACROPIPELINE IMPLEMENTATION OF AUTHENTICATION ALGORITHMS ON HIGH-LEVEL LANGUAGE COLAMO

Macropipeline implementation on reconfigurable computer systems with a dynamic reconfigurable architecture of authentication algorithms for security subsystems of complex informational systems is considered in this paper. Conditions of the effective realization for reconfigurable computer systems are given for a general scheme of macropipeline tasks. Realization of a typical authentication algorithm SSL on a high-level language COLAMO is considered. Author gives a description of achieved performance of a described macropipeline task for a reconfigurable computer system.

Macropipeline computing; reconfigurable computer system; authentication algorithms; high-level language COLAMO.

Обеспечение безопасности сложных информационных комплексов требует оперативной обработки больших объемов данных по вычислительно-трудоемким алгоритмам. Это обуславливает высокие требования к производительным системам, которые реализуют методы аутентификации (а в случае необходимости – блокирование несанкционированного доступа к информации). Традиционные вычислительные средства не позволяют решать такие задачи в режиме реального времени, что противоречит требованиям к системам безопасности. Применение реконфигурируемых вычислительных систем (РВС) позволит обеспечить высокую скорость обработки данных в системах безопасности сложных технических комплексов.

Для РВС характерна структурная реализация задач [1], подразумевающая аппаратное исполнение всех задействованных вычислительных узлов. Однако существуют задачи, которые невозможно решать структурно из-за больших аппаратных затрат, возникающих при реализации всего вычислительного графа задачи в виде конвейера. Такие задачи, как правило, содержат большое число условных переходов, аппаратная реализация всех альтернативных ветвей которых значительно сокращает удельную производительность РВС.

В таких задачах необходима процедурная реализация всего вычислительного графа или его фрагментов. Данный подход особенно характерен для задач аутентификации в больших системах безопасности, где применяются криптопреобразования различных видов. Одним из наиболее эффективных методов решения такого рода задач, требующих согласования потоков данных между конвейерной и процедурной составляющими, является макроконвейер. Суть макроконвейерного принципа состоит в распределении вычислительных заданий между процессорными элементами таким образом, чтобы каждому процессорному элементу на очередном шаге вычислений выдавались данные, которые будут обрабатываться определенное время независимо от других процессорных элементов.

До настоящего времени макроконвейерные задачи разрабатывались специалистом-схемотехником. Однако схемотехническое программирование реконфигурируемых вычислительных систем, построенных на основе ПЛИС, представляет собой достаточно трудоемкий процесс, требующий от пользователя специальных знаний и навыков. Все это неуклонно приводит к увеличению времени создания и отладки задачи до нескольких месяцев.

Для эффективной реализации вычислительных алгоритмов на РВС был разработан язык высокого уровня COLAMO, позволяющий программисту максимально просто описывать различные виды параллелизма в достаточно сжатом виде и содержащий мощные конструкции, как в традиционных языках программирования [2]. В связи с недавними расширениями языка COLAMO [3], в которых появился инструментарий для описания макроконвейерных вычислений, предлагается

использовать его для сокращения времени создания задач, содержащих разнотипные организации вычислений (конвейерные и процедурные).

Рассмотрим более подробно методы описания макроконвейерных задач на языке программирования высокого уровня для РВС. На рис. 1 представлена типовая структурная схема макроконвейерной реализации задачи.

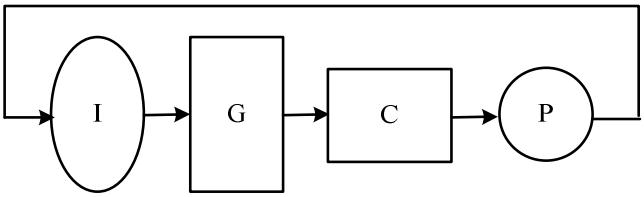

Рис. 1. Структурная схема макроконвейерной реализации задачи

Исходные данные каждого такта поступают с генератора G, управляемого интерфейсом I, на конвейерную схему обработки C, а далее в том же темпе – на блок P, который реализован как макроконвейер, состоящий из множества распараллеленных процедурных блоков, структура которого представлена на рис. 2. Степень распараллеливания процедур определяется временем обработки одного набора данных одной процедурой.

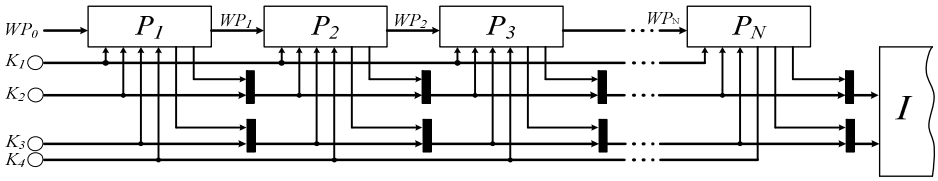

Рис. 2. Структура макроконвейера P

Все секции процедуры объединены общей шиной: входная последовательность подается на вход первой секции и, если она не занята, далее поступает на обработку. Если секция уже обрабатывает порцию данных, входная последовательность без изменений подается на выход, который, в свою очередь, заведен на вход другой секции, где также проверяется занятость текущей секции, и так далее.

Согласованная работа разнородных вычислительных блоков в едином вычислительном контуре обеспечивается использованием на языке высокого уровня COLAMO специальных интерфейсов типа «конвейер-конвейер», «конвейер-процедура» и «процедура-процедура» [4].

Для поддержки процедурных реализаций фрагментов задач в языке COLAMO используется конструкция LocalProc, имеющая следующий формат:

LocalProc <Имя_процедуры>(in: <Список_входных_параметров>; out: <Список_выходных_параметров>) >> <Имя_процессора/блока>;

…

EndLocalProc;

«Имя_процессора/блока» представляет собой имя, на котором выполняется процедура. В качестве процессора могут выступать встроенные в ПЛИС процессорные ядра (например, MicroBlaze), внешние процессорные элементы или реализованные схемотехниками специализированные библиотечные элементы. При этом библиотечный элемент должен удовлетворять следующим требованиям:

- достаточная гибкость программирования, то есть возможность выполнения при помощи элемента определенного набора операций проблемной области;
- минимальное время обработки одного данного в процедуре;
- масштабируемость, позволяющая построить эффективный макроконвейер;
- рабочая частота, обеспечивающая максимальную производительность системы.

Такие элементы создаются в виде IP-ядра. Они содержат свою систему команд и могут программироваться средствами языка COLAMO.

Для организации плотного потока данных с генератора (каждый такт) при согласовании конвейерных и процедурных фрагментов необходимо обеспечить количество локальных процедур (секций), равное времени обработки одного данного одной локальной процедурой.

Рассмотрим реализацию макроконвейера средствами языка COLAMO на примере типового алгоритма аутентификации, содержащего последовательно соединенные хэш-функции и симметричный блочный шифр RC4 [5]. При реализации на РВС такой задачи последовательно подключенные хэш-функции реализуются конвейерно, алгоритм RC4 – процедурно. Время работы процедурной реализации RC4 определялось из соотношения:

$$T_{rc4} = t_u + t_n + t_ш + \lambda,$$

где t_u – время работы цикла инициализации массива S, t_n – время работы цикла начальной перестановки элементов массива S, $t_ш$ – время работы цикла обработки данных, λ – время, затрачиваемое в процедуре на организацию выгрузки результата.

Время работы цикла инициализации определялось по формуле

$$t_u = m \cdot n_u \cdot t,$$

где m – число элементов массива S, n_u – количество операций при инициализации, t – время выполнения одной операции в машинных тактах. При $m = 256$, $n_u = 1$, $t_u = 1$, получим, что $t_u = 256$ тактов. Однако использование двухпортовой памяти позволило вдвое сократить время инициализации. Поэтому $t_u = 128$ тактов.

Время работы цикла начальной перестановки элементов массива S определялось по формуле

$$t_n = m \cdot n_ï \cdot t,$$

где n_n – количество операций при начальной перестановке элементов массива S. При $m = 256$, $n_n = 3$, $t = 1$, получим, что $t_n = 768$ тактов.

Время обработки данных определялось по формуле

$$t_ш = k \cdot n_ш \cdot t,$$

где $n_ш$ – количество операций при обработке данных, k – количество данных. При $n_ш = 4$, $t = 1$, получим что $t_n = 4 \cdot k$ тактов.

Таким образом, время работы процедуры рассчитывалось следующим образом:

$$T_{rc4} = 128 + 768 + 4 \cdot k + \lambda.$$

При $k = 14$ и $\lambda = 10$ было получено $T_{RC4} = 962$ такта машинного времени. Следовательно, для обеспечения разбора плотного потока данных, поступающего из конвейера С (см. рис. 1), в макроконвейере Р потребовалось установить 962 секции процедуры RC4.

Программа на языке COLAMO состоит из основного модуля, в котором описаны схема генерации входных последовательностей, алгоритм их обработки, анализ результатов, а также дополнительных модулей, каждый из которых содержит описание соответствующего криптопреобразования (хэш-функции и процедура RC4). Модули, выполняющие хеширование входной последовательности, естест-

венным образом описываются на языке COLAMO. Вычислительные и управляющие команды для процедурного блока описаны в теле конструкции LocalProc. Вызов локальных процедур осуществляется внутри кадра в теле цикла, параметры которого определяются количеством необходимых процедурных блоков для обработки входного потока данных без пауз:

```
Cadr Main;
var data : array uinteger [8:vector,963:vector,4:vector,N:stream] com;
…
for iv:=0 to 7 do
…
   for s:=0 to N-1 do
  …
     for j:=0 to 961 do
        LocalProc_RC4(data[iv,j,0,s],data[iv,j,1,s],data[iv,j,2,s],data[iv,j,3,s],
              data[iv,j+1,0,s],data[iv,j+1,1,s],data[iv,j+1,2,s],data[iv,j+1,3,s]);
…
EndCadr.
```

В силу того, что второй параметр *j* коммутационной переменной *data* является векторным, циклический вызов локальных процедур приводит к каскадному подключению так, как показано на рис. 2, причем производится это автоматически транслятором языка COLAMO. Отметим, что на входы первого процедурного блока заводятся выходы конвейерной части вычислительного графа задачи.

Описываемый типовой алгоритм аутентификации был реализован на вычислительном блоке «Ригель», состоящем из 4 базовых модулей по 8 ПЛИС семейства Virtex-6 на каждом, на частоте 220 МГц (рис. 3). На вычислительном блоке удалось разместить 8 макроконвейеров, производительность вычислительного блока при решении задачи составила $1{,}76\times10^9$ наборов данных/с.

Рис. 3. Вычислительный блок «Ригель»

Достигнутое значение производительности позволяет сделать вывод о перспективности использования языка программирования высокого уровня для РВС COLAMO, существенно повышающем скорость разработки макроконвейерных реализаций алгоритмов аутентификации подсистем безопасности сложных информационных комплексов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Каляев А.В, Левин И.И.* Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: ООО «Изд-во Янус-К», 2003.
2. *Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И.* Реконфигурируемые мультиконвейерные вычислительные структуры. – 2-е изд. перераб. и доп. / Под общ. ред. И.А. Каляева. – Ростов на-Дону: Изд-во ЮНЦ РАН, 2009. – 344 с.
3. *Гудков В.А., Левин И.И.* Расширение языка высокого уровня COLAMO для программирования реконфигурируемых вычислительных систем на уровне логических ячеек ПЛИС // Вестник компьютерных и информационных технологий. – М.: Машиностроение, 2010. – № 12. – С. 10-17.
4. *Раскладкин М.К.* Библиотека масштабируемых интерфейсов для реконфигурируемых вычислительных систем на основе ПЛИС // Материалы 9-й Международной конференции-семинара «Высокопроизводительные параллельные вычисления на кластерных системах». – Владимир: Изд-во ВГУ, 2009. – С. 329-331.
5. *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: ТРИУМФ, 2003 – 816 с.

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

Коваленко Алексей Геннадьевич – Научно-исследовательский институт многопроцессорных вычислительных систем им. А.В. Каляева Южного федерального университета; e-mail: k.a.g@bk.ru; 347928, г. Таганрог, ул. Чехова, 2, тел.: 88634315491; младший научный сотрудник.

Kovalenko Alexey Genad'evich – Scientific-Research Institute Multiprocessing Computing Systems after Kalyaev of South Federal University; e-mail: k.a.g@bk.ru; 2, Chekhov street, Taganrog, 347928, Russia; phone: +78634315491; research assistant.

УДК 004.421.6

Е.Е. Семерникова

РАЗРАБОТКА МАСШТАБИРУЕМЫХ РЕАЛИЗАЦИЙ АЛГОРИТМОВ СИМВОЛЬНОЙ ОБРАБОТКИ ДЛЯ РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ*

Рассмотрена масштабируемая реализация алгоритма на реконфигурируемой вычислительной системе с динамически перестраиваемой архитектурой типовых, для задач аутентификации, алгоритмов символьной обработки. Описана реализация алгоритма криптографического хеширования SHA1 на языке высокого уровня с неявным описанием параллелизма COLAMO. Полученная параллельная программа обладает свойством переносимости с одной реконфигурируемой вычислительной системы на другую, рассмотрены способы ее масштабирования, представлены достигнутые характеристики производительности.

Безопасность; масштабируемость; хэш-функция; язык высокого уровня COLAMO; распараллеливание по конвейерам.

E.E. Semernikova

DEVELOPMENT OF SCALABLE REALISATIONS SYMBOLIC PROCESSING ALGORITHMS FOR RECONFIGURABLE COMPUTER SYSTEMS

The paper describes scalable realization of typical symbolic processing algorithms from authentication tasks for reconfigurable computer system with dynamically reconfigurable architecture. The author suggests description of cryptographic hashing algorithm SHA1 realization with

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