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.

the help of a high-level language COLAMO. The developed parallel program in possession of portability, considered ways of scalability for this program, presented description of achieved performance.

Safety; scalability; hash function, high-level language COLAMO; pipeline parallelization.

Бурное развитие методов и средств обработки данных в последней четверти XX в. привело к созданию новых алгоритмов обработки символьной информации, применяемых в различных областях безопасности больших систем, таких как криптография, защищенный электронный документооборот, системы идентификации и аутентификации, сигнатурный анализ и другое. Современные реализации таких алгоритмов требуют от вычислительных систем скоростей обработки порядка 10^{10}-10^{13} наборов данных в секунду.

В этой связи высокие показатели удельной производительности и энергоэффективности реконфигурируемых вычислительных систем (РВС) [1–2] по сравнению с многопроцессорными вычислительными системами (МВС) кластерной архитектуры обуславливают целесообразность их применения для решения задач символьной обработки, в том числе в составе подсистем безопасности.

Одной из наиболее часто используемых реализаций криптографических алгоритмов при сетевых коммуникациях в Интернете являются хэш-функции, позволяющие создавать цифровые подписи отправляемых сообщений, беспечивающие необходимый уровень конфиденциальности передаваемой информации. Известные на сегодняшний день реализации хэш-функций для РВС разработаны схемотехниками, но такие решения не обеспечивают масштабируемость и переносимость с одной РВС на другую. Преобразовать подобную вычислительную схему таким образом, чтобы ее можно было использовать на другой РВС или же увеличить количество вычислительных конвейеров, – трудоемкая задача и требует временных затрат порядка нескольких недель.

Язык высокого уровня COLAMO [3] является платформонезависимым, что достигается за счет использования в процессе синтеза задачи файла описания архитектуры текущей РВС, в котором отображается информация о типе ПЛИС, их количестве, числе связей между ними. Язык COLAMO позволяет программисту максимально просто описывать различные виды параллелизма в достаточно сжатом виде, оперируя лишь типами данных и индексацией элементов массивов. Языковые средства представлены в виде конструкций операторов условий, циклов и арифметических операций, синтаксис которых аналогичен соответствующему синтаксису в языках Фортран и Паскаль. Расширение языка COLAMO конструкциями, позволяющими осуществлять битовую обработку данных, дает возможность программисту значительно сократить время разработки реализаций задач символьной обработки. Программные средства языка позволяют организовывать распараллеливание по конвейерам, изменяя лишь параметр цикла. Таким образом, использование языка высокого уровня COLAMO для реализаций алгоритмов символьной обработки, а в частности, хэш-функций, позволяет создавать столь же эффективные вычислительные структуры, как и в случае схемотехнического решения, но в то же время преобразования, обеспечивающие перенос с одной РВС на другую, и масштабирование осуществляются в пределах нескольких часов.

Рассмотрим масштабируемую реализацию алгоритма цифровой подписи SHA1 [4] на языке COLAMO. Задача разбита на несколько смысловых блоков, оформленных в виде подкадров (аналог подпрограммы). Подкадры, описывающие нелинейные функции, а также шаг главного цикла, входят в состав подкадров, реализующих каждый из четырех раундов. Основной же подкадр ScSha включает в себя блок расширения исходного сообщения до восьмидесяти 32-разрядных слов и последовательный вызов всех раундов, таким образом, осуществляются вычисления по заданному алгоритму.

В основном теле программы (кадре) описывается распределение потоков данных при помощи регистрового интерфейса (InterfaceRG). Интерфейс обеспечивает информационный обмен между вычислительной схемой в пределах одного кристалла ПЛИС и управляющей ЭВМ. Также в кадре происходит вызов основного подкадра ScSha.

Загрузка данных (длина сообщения в битах, эталонные значения, а также стартовые и конечные адреса для каждой из блочной памяти) осуществляется через пользовательскую шину интерфейса путем заполнения цепочки регистров. В схеме задействовано 16 блоков двухпортовой внутренней памяти шириной 32 бита и глубиной 256 значений каждый. Конкатенация значений (по одному из каждого блока памяти) образует 512 бит исходного сообщения. При этом следует заметить, что если длина входного сообщения меньше, то оно все равно дополняется до 512 бит, так как алгоритм работает с блоками данных длиной, кратной этому числу разрядов. Расширение происходит следующим образом: вначале добавляется 1, затем нули, а в последние 64 бита записывается размер исходного сообщения.

Для выработки входной последовательности, загружаемой в вычислительную схему описанной реализации SHA1, на основе множеств значений, хранящихся в памяти, используются функции: Generator, инициализирующая счетчики адресов для каждой памяти, и Connect, определяющая зависимость между генерируемыми адресами. То есть изменение адреса чтения каждой памяти на каждом такте задается однозначно и обеспечивает полный перебор данных. Пример использования функций приведен ниже:

```
for iConv := 0 to Count_Conv - 1
    for i := 0 to 15 do
        for j := begin_addr[iConv,i] to end_addr[iConv, i] do
            addr[ iConv, i] := Generator(j);
        for k := 1 to 15 do
            Connect(addr[iConv, k-1] , addr[ iConv, k]).
```

Как видно из отрывка COLAMO-программы, массивы данных являются двумерными, где первое измерение – параметр цикла, отвечающий за распараллеливание по конвейерам.

После этого формируется массив из шестнадцати 32-разрядных слов, который подается на вход ScSha. В результате работы основного подкадра получается 160-битное хэш-значение входной последовательности в виде набора пяти 32-разрядных слов. Каждое из слов сравнивается с ранее загруженным и хранящимся в регистре эталонным значением. В случае, когда хэш-значение полностью совпадает с эталонным, формируется так называемый бит совпадения, который далее передается в интерфейс и выставляется на соответствующую шину статуса вместе с индексом совпавшего сообщения, само же значение не выгружается, что позволяет сократить объем информационного потока, а значит, сократить оборудование и увеличить скорость обмена данными. Статус схемы обрабатывается управляющей ЭВМ, что позволяет осуществлять контроль правильной работы созданной реализации. Линейная схема полученной реализации представлена на рис. 1.

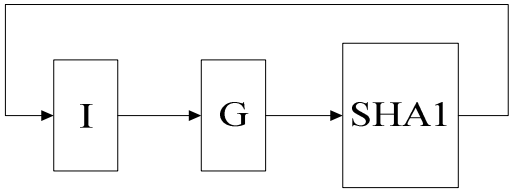

Рис. 1. Конвейер SHA1

Блок I соответствует работе интерфейса, G – генератора адресов чтения памяти, последний блок осуществляет вычисления по алгоритму. Обратная связь показывает возможность распараллеливания за счет использования большего числа конвейеров.

Полученная в результате трансляции и синтеза текста программы вычислительная структура представляет собой конвейер. Программные средства языка позволяют параллельно подключать несколько конвейеров, что ведет к увеличению производительности пропорционально их количеству. Число конвейеров задается параметром цикла и может принимать любые значения, ограниченные лишь доступным ресурсом, что позволяет максимально задействовать оборудование, а также обуславливает масштабируемость.

Параллельная программа, решающая задачу символьной обработки, была реализована на языке COLAMO для реконфигурируемого вычислителя, содержащего 8 ПЛИС Vertex 6 фирмы Xilinx с 24 млн элементарных вентилей и позволила разместить 6 конвейеров на кристалле (48 конвейеров всего). Работая на частоте 250МГц, программа показала реальную производительность $1{,}2\times10^{10}$ наборов данных/с.

Использование языка высокого уровня COLAMO с неявным описанием параллелизма позволяет создавать эффективные масштабируемые реализации прикладных задач символьной обработки, при этом время их разработки составляет всего несколько дней, что во много раз быстрее схемотехнических реализаций, не обеспечивающих к тому же масштабируемость и переносимость с одной РВС на другую.

БИБЛИОГРАФИЧСКИЙ 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. *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: ТРИУМФ, 2003. – 816 с.
4. *Гудков В.А., Левин И.И., Дордопуло А.И.* Ресурсонезависимое программирование многопроцессорных систем на языке COLAMO // Искусственный интеллект. – Донецк: Наука і освіта, 2006. – № 4. – С. 211-219.

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

Семерникова Евгения Евгеньевна – Научно-исследовательский институт многопроцессорных вычислительных систем им. А.В. Каляева Южного федерального университета; e-mail: semernikova_e@mail.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 88634315491; программист.

Semernikova Eugenia Evgen'evna – Scientific-Research Institute Multiprocessing Computing Systems after Kalyaev of South Federal University; e-mail: semernikova_e@mail.ru; 2, Chekhov street, Taganrog, 347928, Russia; phone: +78634315491; programmer.