УДК 004.272, 004.31

РЕАЛИЗАЦИЯ НА FPGA КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ С БОЛЬШИМ КОЛИЧЕСТВОМ ИТЕРАЦИЙ

М.В. Качинский, А.В.Станкевич, А.И.Шемаров
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», Минск, Беларусь

Аннотация. В статье рассматриваются возможные архитектурные решения реализации на базе FPGA криптографических алгоритмов с большим количеством итераций однотипных вычислений, обеспечивающие высокую производительность при обработке блоков входных данных. Проведен анализ производительности. Обосновывается выбор для таких реализаций параллельно-итеративной или конвейерно-итеративной архитектуры разрабатываемых специализированных процессоров. Количество ступеней конвейера, количество параллельных подсистем предлагается выбирать, исходя из параметров криптографического алгоритма, ограничений аппаратных ресурсов конкретного кристалла FPGA, а также возможности размещения в кристалл и трассировки соединений итогового проекта полученного специализированного процессора используемыми инструментальными средствами проектирования. Даются рекомендации по выбору архитектуры.
Ключевые слова: реализация на FPGA; криптографический алгоритм; итерация; блок данных; итеративная, параллельная, конвейерная архитектура процессора; ступень конвейера; аппаратные ресурсы кристалла FPGA; производительность.

FPGA IMPLEMENTATION OF CRYPTOGRAPHIC ALGORITHMS WITH A LARGE NUMBER OF ITERATIONS

M.V. Kachinsky, A.V. Stankevich, A.I. Shemarov
Educational Institution "Belarusian State University of Informatics and Radioelectronics", Minsk, Belarus

Abstract. The article analyzes the usage of architectural solutions for FPGA implementations of cryptographic algorithms, with a significant number iterations of uniform calculations that provide high performance in processing of input data blocks. A performance analysis was performed. The choice of parallel-iterative or pipeline-iterative architecture for specialized processors is substantiated by the evidence. The number of pipeline stages and the number of parallel subsystems are to be chosen based on the parameters of the cryptographic algorithm and the limitations of hardware resources of a particular FPGA device, as well as the possibility of place and route of the final project of specialized processor. Recommendations are provided for the optimal selection of architectural design.
Keywords: FPGA implementation; cryptographic algorithm; iteration; data block; iterative, parallel and pipelined processor architecture; pipeline stage; FPGA hardware resources utilization; performance.

Введение

В настоящее время в задачах обеспечения конфиденциальности, целостности данных и аутентификации широко используются различные криптографические алгоритмы, с помощью которых обрабатывается некоторая последовательность блоков данных. При этом криптографический алгоритм может иметь большое количество итераций однотипных вычислений. В качестве примера можно рассмотреть задачу хэширования блока входных сообщений одинаковой длины алгоритмами SHA-1 (RFC3174) или семейства SHA-2 (RFC6234). В этом случае каждое входное сообщение разбивается на блоки, размер которых определяется алгоритмом, и далее эти блоки последовательно обрабатываются вычислительным ядром алгоритма SHA столько раз, сколько имеется этих входных блоков одного сообщения. При этом сам алгоритм SHA имеет некоторое фиксированное количество однотипных итераций (раундов). Другим примером является использование алгоритмов формирования ключа на основе пароля для блока паролей. В алгоритмах PBKDF2

(RFC2898) и Argon2 [1] имеется параметр, задающий количество итераций алгоритма верхнего уровня итерации. При этом алгоритмы, использующиеся на более низких уровнях иерархии, также имеют свое определенное количество итераций.

Если требуется высокопроизводительная реализация таких алгоритмов, то необходимо использовать аппаратную реализацию. Одним из таких возможных вариантов является реализация на базе FPGA (Field Programmable Gate Array). При реализации на конкретном кристалле FPGA разработчик имеет ограничение по имеющимся аппаратным ресурсам. Рассмотрим возможные варианты архитектурных решений таких реализаций с учетом этого ограничения и оценим возможную производительность таких решений.

Основная часть

В качестве базового блока архитектуры специализированного процессора будем рассматривать процессор одной итерации алгоритма. Этот процессор реализует одну базовую совокупность повторяющихся операций алгоритма и содержит комбинационную схему для реализации требуемых вычислений, а также выходной регистр для фиксации результата итерации. Например, для алгоритмов семейства SHA-2 в качестве такого процессора можно рассматривать устройство, реализующее один раунд алгоритма. Сравним возможные реализации.

На рис. 1 приведена параллельно-итеративная реализация криптографического алгоритма с итерациями.

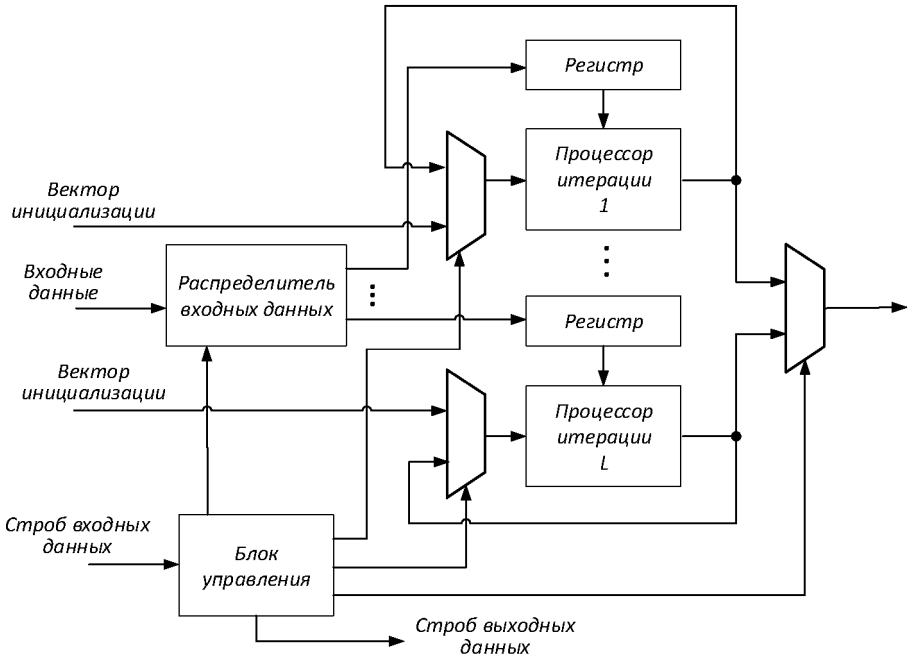

Рис. 1. Параллельно-итеративная архитектура
Fig. 1. Parallel-iterative architecture

Одно процессорное ядро такой архитектуры будет включать процессор одной итерации, регистр входных данных алгоритма и мультиплексор данных итерации. Этот мультиплексор обеспечивает передачу в процессор итерации вектора инициализации алгоритма (на первой итерации), либо промежуточного значения после текущей итерации. По завершении N итераций, определяемых соответствующим алгоритмом, на выходе процессора итераций будет получено требуемое значение алгоритма. При наличии в алгоритме каких-то однократных вычислений они могут быть

выполнены последовательно до или после рассмотренного процессорного ядра. Подсчет числа итераций организуется счетчиком блока управления.

Предположим, что вычисления одной итерации выполняются за один такт частоты синхронизации. В этом случае на каждое следующее процессорное ядро (рис. 1) входные данные должны подаваться со сдвигом на такт. Общий размер обрабатываемого блока данных равен L слов, что соответствует числу процессорных ядер. Порядок подачи входных данных на итеративные процессорные ядра обеспечивает распределитель входных данных. Данные вычислительных ядер объединяются в общий выходной поток с помощью выходного мультиплексора. Число процессорных ядер L зависит от имеющихся ресурсов кристалла FPGA. В простейшем случае при $L = 1$ распределитель данных и выходной мультиплексор не нужны. Время получения полного блока результата для параллельно-итеративной архитектуры равно $N + L - 1$ тактов (без учета такта на прием входных данных в регистр).

На рис. 2 приведена итеративно-конвейерная реализация криптографического алгоритма с итерациями.

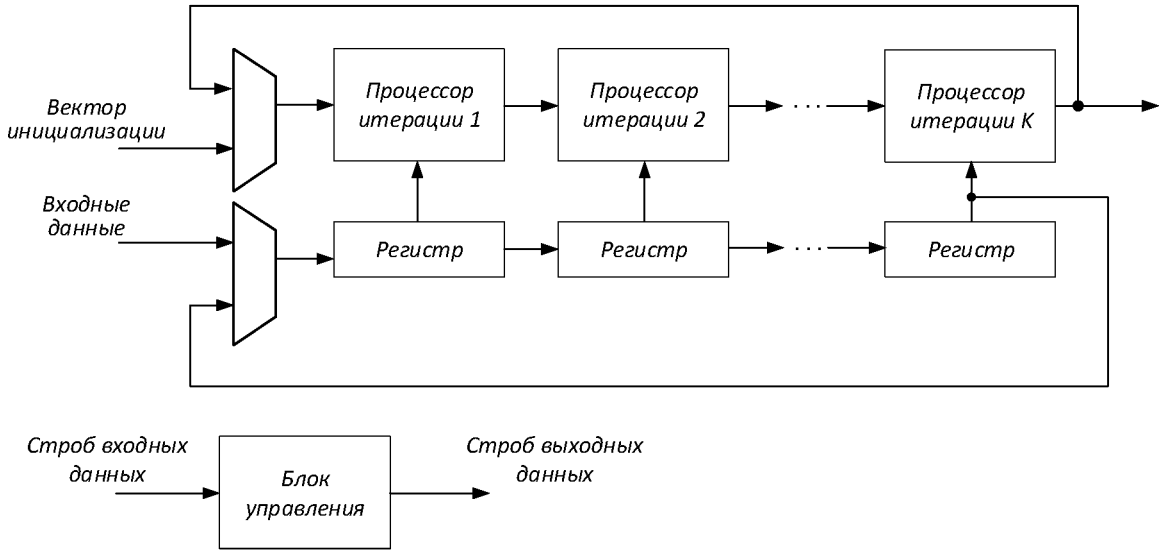

Рис. 2. Конвейерно-итеративная архитектура
Fig. 2. Pipeline-iterative architecture

Конвейерная архитектура реализует концепцию проектирования, заключающуюся в развертывании повторяющихся вычислений (unrolling the loop) [2]. В этом случае каждая из повторяющихся операций выполняется последовательно на своем процессоре, после чего полученный результат передается на процессор следующей итерации. В случае, если данные каждой итерации связаны с входными данными первой итерации (например, как в алгоритмах SHA), то параллельно с конвейером вычислительных итераций должен быть организован конвейер данных, который обеспечивает подачу правильных данных для соответствующей ступени конвейера вычислительных итераций (рис. 2).

В случае, если логической емкости кристалла FPGA достаточно для реализации всех N итераций алгоритма, то никакой обратной связи и входного мультиплексора, показанных на рис. 2 не требуется. Если логическая емкость кристалла позволяет разместить для повышения производительности несколько параллельно работающих конвейеров, то можно реализовать решение, подобное рис. 1, где вместо итеративных процессорных ядер будут использоваться параллельно работающие конвейеры.

В случае, если все итерации алгоритма для конвейерной архитектуры не могут быть размещены в кристалле FPGA, то можно реализовать только K из общего числа N

итераций ($K < N$) и полученные после K итераций результаты вычислений с помощью мультиплексора подать на вход первого процессора итераций для продолжения вычислений (рис. 2).

В конвейерно-итеративной архитектуре для удобства и упрощения аппаратной реализации N должно быть кратным числа K, то есть $N = i \times K$, где i – целое число. Размер блока входных данных, загружаемых в конвейер, состоит из K слов. По истечении K тактов (предполагается что одна итерация выполняется за такт) на выходе конвейера появится выходное значение для первого входного слова блока данных после выполнения K итераций. Для получения N итераций выходные данные надо мультиплексировать из K-ой ступени конвейера обратно на вход и пропустить через конвейер этот модифицированный блок промежуточных значений еще $i - 1$ раз.

Время получения полного блока результата для конвейерно-итеративной архитектуры равно $iK + K - 1 = N + K - 1$ тактов. Если L для параллельно-итеративной архитектуры выбрать равным K, то по производительности эти архитектуры будут соответствовать друг другу.

С точки зрения аппаратных затрат предпочтительнее итеративно-конвейерная архитектура, поскольку в ней будет отсутствовать выходной мультиплексор, схема распределения входных данных, вместо L мультиплексоров обратной связи будут только два, проще будет блок управления, поскольку будет необходимо только формирование выходного строба данных. Уменьшение аппаратных затрат создаст лучшие условия для размещения и трассировки проекта в кристалл FPGA, что потенциально может привести к более высокой тактовой частоте полученного проекта, и, следовательно, к более высокой производительности.

Заключение

При реализации на FPGA криптографических алгоритмов с большим количеством итераций могут использоваться различные варианты параллельно-итеративной и конвейерно-итеративной архитектур в зависимости от требуемой производительности и логической емкости используемого кристалла FPGA. Для получения высокой производительности целесообразнее использовать конвейерно-итеративную архитектуру, которая имеет меньшие аппаратные затраты по сравнению с параллельно-итеративной и позволяет создать лучшие условия для последующего размещения и трассировки проекта в кристалл FPGA.

Список использованных источников / References

1. Alex Biryukov, Daniel Dinu, Dmitry Khovratovich. (2016) Argon2: new generation of memory-hard functions for password hashing and other applications. *IEEE European Symposium on Security and Privacy (EuroS&P). IEEE.* 292-302.

2. Kilts S. (2007) Advanced FPGA design: Architecture, Implementation, and Optimization. USA. John Wiley & Sons, Inc.

Сведения об авторах

Качинский М.В., канд. техн. наук, доц., доцент кафедры электронных вычислительных средств, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», kachinsky@bsuir.by.
Станкевич А.В., канд. техн. наук, доц., доцент кафедры электронных вычислительных средств, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», stankevich@bsuir.by.
Шемаров А.И., канд. техн. наук, доц., доцент кафедры электронных вычислительных средств, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», shemarov@bsuir.by.

Information about the authors

Kachinsky M., Ph.D. in Computer Sciences, Associate Professor, department of electrical engineering, Educational Institution "Belarusian State University of Informatics and Radioelectronics", e-mail: kachinsky@bsuir.by.
Stankevich A., Ph.D. in Computer Sciences, Associate Professor, department of electrical engineering, Educational Institution "Belarusian State University of Informatics and Radioelectronics", e-mail:stankevich@bsuir.by.
Shemarov A., Ph.D. in Computer Sciences, Associate Professor, department of electrical engineering, Educational Institution "Belarusian State University of Informatics and Radioelectronics", e-mail:shemarov@bsuir.by.