# ПРОГРАММНО-АППАРАТНАЯ РЕАЛИЗАЦИЯ КОМПОНЕНТОВ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ

Кожевников Д. С.[1], Шатилов Н. П.[2]
Научный руководитель: Торгаев С.Н.
[1]Томский политехнический университет, 634050, г. Томск, пр. Ленина, 30
[2]Томский государственный университет, 634050, г. Томск, пр. Ленина, 36

E-mail: demonicsmile2@yandex.ru

В настоящее время большое внимание исследователей и разработчиков уделяется аппаратным реализациям, в частности, компонентов телекоммуникационных систем, поскольку такие реализации часто могут работать быстрее программных. Тем не менее, программные реализации в некоторых случая оказываются более гибкими и, вместе с тем, легко отлаживаемыми.. Отметим, что в виде «жесткой» логики реализуется большое количество телекоммуникационных компонентов; в том числе, активно исследуются возможности аппаратной реализации телекоммуникационных протоколов высоких уровней. Соответственно, с развитием технологий программирования и проектирования, необходимы четкие критерии сравнения для оценки эффективности аппаратной реализации по сравнению с программной. В настоящей работе такими критериями были выбраны время и скорость обработки входных данных. Соответственно, в данном докладе представлены предварительные результаты по оценке эффективности использования аппаратных реализаций как компонентов телекоммуникационных систем. В частности, поскольку в современном мире на первое место выходит защищенность информации, передаваемой по каналам связи, в данном докладе рассматривается возможность аппаратного шифрования данных. В качестве алгоритма шифрования выбран стандарт США DES.

В 1972 году Национальное бюро стандартов (National Bureau of Standards, NBS) выступило инициатором программы защиты линий связи и компьютерных данных. Одной из целей этой программы была разработка единого, стандартного криптографического алгоритма. К разрабатываемому продукту было выдвинуто следующее основное требование: алгоритм должен быть проверен и сертифицирован, легко доступен, а использующие его различные криптографические устройства должны взаимодействовать. Таким образом появился алгоритм шифрования DES (Data Encryption Standard) [1].

DES представляет собой блочный шифр, которые преобразует данные открытого текста 64-битовыми блоками. Соответственно, на вход алгоритма подаётся 64-битовый блок открытого текста, а с выхода снимается 64-битовый блок шифртекста. Отметим, что DES является симметричным алгоритмом, а именно, для шифрования и расшифрования используются одинаковые алгоритм и ключ (за исключением небольших различий в использовании ключа).

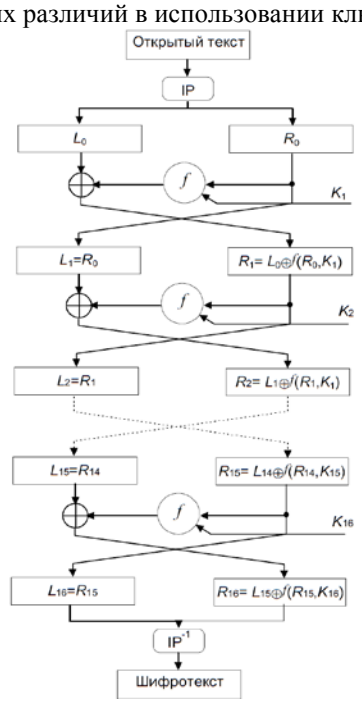


Рис. 1. Алгоритм шифрования DES.

Длина ключа равна 56 битам. Отметим, однако, что ключ, как правило, представляется 64-битовым числом, но каждый восьмой бит используется для проверки четности и игнорируется; Биты четности являются наименьшими значащими битами байтов ключа. Согласно стандарту DES, ключ, который может быть любым 56-битовым числом, можно изменить в любой момент времени. Тем не менее, известно, ряд чисел являются слабыми ключами в смысле стойкости шифра, поскольку первоначальное значение ключа, в ходе алгоритма шифрования, расщепляется на две половины, каждая из которых сдвигается независимо. Примером может служить вектор, наполовину состоящий из единиц и наполовину из нулей. В этом случае для всех этапов алгоритма используется один и тот же ключ. Основными операциями в алгоритме являются: На рисунке 1 приведена схема алгоритма шифрования DES.

Алгоритм DES был реализован аппаратно на ПЛИС типа FPGA Cyclone II Starter Development Board компании Altera (рисунок 2).

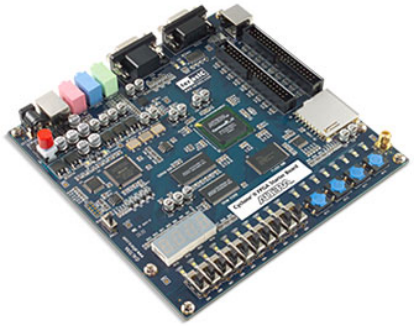

Рис. 2. Внешний вид платы Cyclone II Starter Development Board

Технология FPGA в активно используется в настоящее время и базируется на блоках умножения-суммирования (в частности, для обработки сигналов, DSP), а также логических элементах (как правило, на базе таблиц истинности соответствующих функция) и, вместе с тем, их блоках коммутации [2]. Отметим, что технология FPGA обычно используются для обработки сигналов и имеет более гибкую архитектуру, чем, например, CPLD. Программа для FPGA хранится в распределённой памяти, которая может быть выполнена как на основе энергозависимых ячеек статического ОЗУ (программа не сохраняется при исчезновении электропитания микросхемы), так и на основе энергонезависимых ячеек Flash-памяти или перемычек antifuse (программа сохраняется при исчезновении электропитания). Если программа хранится в энергозависимой памяти, то при каждом включении питания микросхемы необходимо заново конфигурировать её при помощи начального загрузчика, который может быть встроен и в саму плату на базе FPGA. Альтернативой ПЛИС FPGA являются более медленные цифровые процессоры обработки сигналов, тем не менее, исследования по использованию таких процессоров при аппаратном шифровании остаются за рамками данного доклада.

Помимо аппаратной реализации докладчиками была разработана программная реализация алгоритма шифрования DES. Соответствующая программа была написана на языке C++ с использованием интегрированной среды разработки Microsoft Visual Studio 2013 Express Edition.

Программа принимает на вход 64-битовый вектор (блок открытого текста) и случайным образом сгенерированный ключ той же размерности. На выходе получается 64-битный блок шифротекста. Отметим, что в программная реализация использует технологию ООП, а именно, для обработки булевых векторов размерности более 32 бит был реализован класс «Длинный булев вектор».

Результатом работы выступает сравнение полученных реализаций, и одним из критериев оценки было выбрано время, затраченное на шифрование с использованием программной и аппаратной реализаций. В частности, в эксперименте определяется множество последовательностей, которые подаются на каждую из реализаций, и сравнивается общее время обработки входных данных.

Отметим, что для достижения максимальной скорости шифрования требуется обеспечить наиболее быстрый способ считывания информации для шифрования. Оптимальным решением в данном случае является считывание этих данных с SD-карты или организация интерфейса передачи данных, который «сможет» быстро осуществлять передачу данных к ПЛИС. Примером такого интерфейса может служить USB или Ethernet. Отметим, что решение данной задачи, а именно, выбора подходящего интерфейса для вычисления «чистого» времени аппаратного шифрования представляет интерес для дальнейших научных исследований, и на данный момент организована память внутри самого кристалла ПЛИС. Этот факт позволил использовать 64-разрядную память, из которой можно единовременно считывать 64 бита открытого текста. Эксперименты по оценке времени шифрования проводятся авторами в настоящий момент.

Отметим, что помимо скорости и времени обработки данных существуют и другие критерии оценки. В дальнейшем планируется оценить и другие реализации, возможно, сравнить с уже существующими по эффективности, надежности, качеству и другим параметрам. Отдельным вопросом при проектировании аппаратного обеспечения является вопрос тестирования, и в данном случае в дальнейшем авторы предлагают использовать программную реализацию для тестирования аппаратной.

Перечисленные задачи открывают перспективы для дальнейших научных исследований.

Работа выполнена при поддержке Министерства образования *и науки Российской Федерации, Госзадание №5.1307.2014.*

**Литература**

1. Прикладная криптография : Протоколы, алгоритмы, исходные тексты на языке Си : пер. с англ. / Б. Шнайер. — М.: Триумф, 2002. — 816 с.: ил.. — Знания и опыт экспертов. — Библиогр.: с. 741-796.
2. Стешенко В.Б. ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов. — М.: ДОДЭКА, 2000. —128 с.