E. G. Vorobiev
Saint-Petersburg state electrotechnical university «LETI»

THE CONCEPT OF PROVIDING A CONTINUITY OF FUNCTIONING OF THE DISTRIBUTED INFORMATION AND TELECOMMUNICATION SYSTEMS ON THE BASIS OF QUANTUM METHODS OF SUBMISSION OF INFORMATION

The possibility of use of the quantum theory of information for providing a continuity of functioning of global information systems is analyzed, the new concept of display of mobile and stationary objects and their passports in a common information space is offered.

Quantum theory of information, common information space, providing of information systems functioning continuity, spatial model

УДК 004.3’144

А. Х. Мурсаев, В. О. Ameленко
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Высокопроизводительные устройства шифрации информации

Рассматриваются вопросы аппаратной реализации устройств шифрации информации. Представляется ряд структурных решений, отличающихся по затратам и производительности.

Устройства шифрации информации, аппаратная реализация

Шифрация информации является одним из важнейших средств обеспечения информационной безопасности. До последнего времени зашифровывали в основном текстовые и звуковые сообщения, а со скоростями передачи подобной информации вполне справляются процессорные средства общего назначения. Но в последнее время возникла потребность и возможность обработки более интенсивных потоков данных, в том числе передача в реальном времени результатов секретных испытаний, закрытые видеоконференции или хотя бы кодирование телевизионных сигналов в закрытых платных кабельных сетях и т. п. Здесь необходима скорость обработки до нескольких десятков миллионов символов в секунду, поэтому требования по производительности шифраторов/дешифраторов непрерывно растут и вызывают к жизни аппаратную реализацию этих задач.

Большинство современных алгоритмов шифрации, таких, как Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), Advanced Encryption Standard (AES) Рейндаля, а также ГОСТ 28147–89 [1]–[4] используют разбиение потока исходных данных на блоки по нескольку десятков бит (поэтому их называют блочными). Блоки кодируются независимо, последовательно друг за другом. Алгоритмы объединяет то, что для каждого блока выполняется многократное повторение двух базовых операций – перестановки бит в шифруемом блоке и поэлементная замена фрагментов шифруемых блоков (байт, тетрад и т. п.) на соответствующие им значения, содержащиеся в памяти ключей шифрации. В ряде алгоритмов операция замены интерпретируется как арифметическое преобразование исходного данного и соответствующего подключа или сочетания замены и арифметического преобразования. Подключом называют отрезок кода секретного ключа, используемого для шифрования.

В качестве примера на рис. 1 приведена схема алгоритма IDEA. Выполняется последовательность из нескольких шагов, на каждом из которых осуществляются двухместные операции с отрезками блока данных и подключами, а также соот-

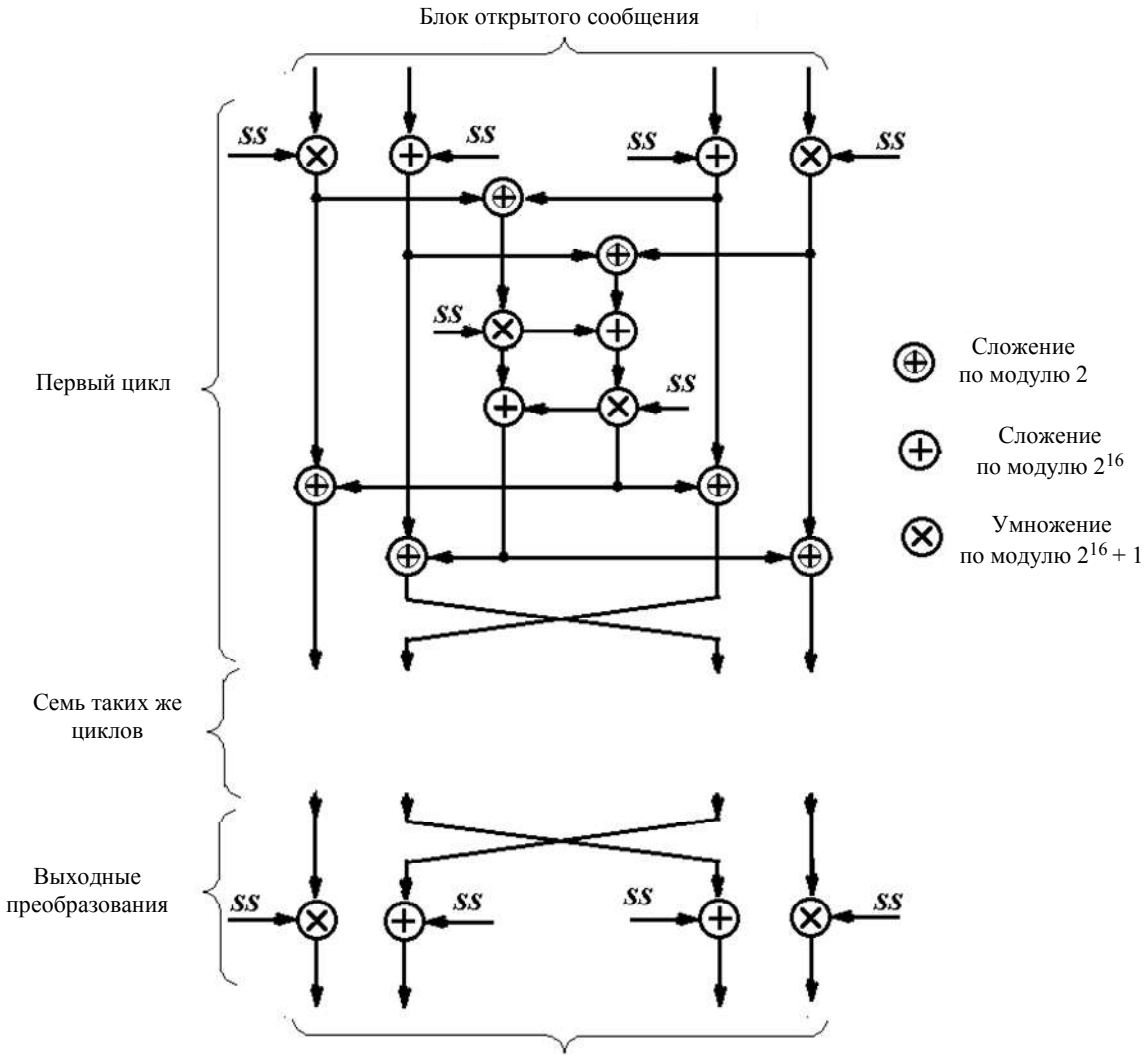

Рис. 1

ветствующие перестановки. Единственная разница между в остальном идентичными шагами состоит в значениях подключей для каждого. Другими словами, выход каждого этапа преобразования – это не что иное, как подфункция предыдущих этапов. Сначала таким преобразованиям подвергается исходный блок данных (блок открытого сообщения), а каждый следующий шаг подобным же образом преобразует данные предыдущего шага.

По такому же принципу строятся остальные алгоритмы, отмеченные в начале статьи.

Один шаг алгоритма по ГОСТ 28147–89 в режиме простой замены предполагает:

– 64-разрядный блок разделяется на две равные части (в дальнейшем обозначим их как n2 – старшие биты и n1 – младшие);

– сложение по модулю 2 кода n1 с одним из восьми 32-разрядных подключей (номер подключа однозначно связан с номером шага алгоритма);

– каждая группа из четырех разрядов полученной суммы (тетрада) заменяется другой четырехбитовой комбинацией. Функции подстановки задаются индивидуально для каждой тетрады кода и в ГОСТе не определены. Эти функции не могут быть произвольными; существует ограниченный набор комбинаций функций замены, гарантирующих обратимость процедуры шифрации/дешифрации. Функции подстановки являются общими для больших групп пользователей (в отличие от индивидуальных секретных ключей пользователя), и таблицы подстановок определяются для групп пользователей соответствующими компетентными органами;

– полученный в предыдущей операции код циклически сдвигается вправо на 11 разрядов;

– сдвинутый код поразрядно суммируется по модулю 2 со старшими 32 разрядами входного блока данных (n2). Результат формирует старшие

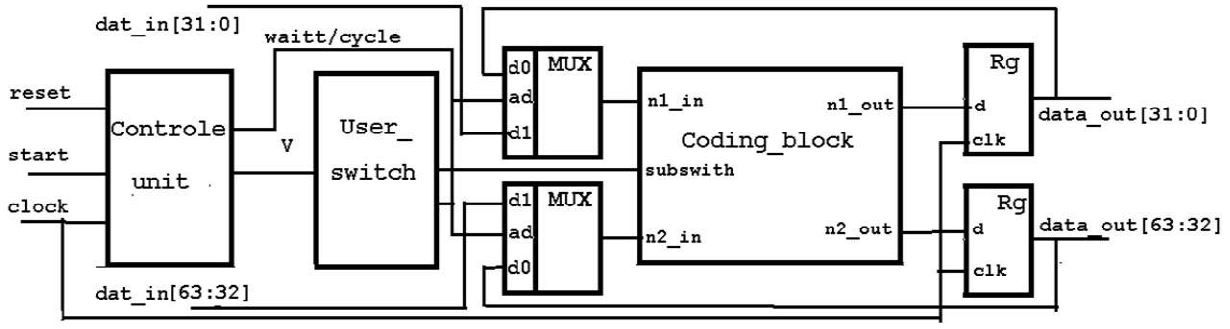

Рис. 2

разряды промежуточного результата, а старшие разряды входных данных на любом этапе представляют младшие разряды результата.

Последовательность преобразований для одного шага шифрации представлена Verilog-программой на листинге 1. Предполагается, что ключ загружается пользователем и хранится в памяти ключа, причем subswitch – это один из элементов ключевой последовательности, а номер выбираемого для каждого шага подключа задается номером шага алгоритма в соответствии с таблицей, определенной ГОСТом.

Конкретные функции замены f0–f7 здесь по понятным причинам не приводятся, и программа базируется на некоторых абстрактных значениях.

Листинг 1.

```
y3 ={ y2[20 : 0], y2[31 : 21]}, //циклический сдвиг на 11 разрядов
n1_out= y3~^n2_in, // исключающее ИЛИ
n2_out=n1_in;
endmodule.
```

В [5] отмечена большая трудоемкость преобразований по алгоритму ГОСТ 28147–89, но аппаратная реализация узла преобразования достаточно экономична. «Сложные» функции замены (четырехразрядный вход -> четырехразрядный выход) реализуются на четырех типичных для микросхем программируемой логики ячейках типа Look_up_table каждая [6], сложения требуют по одной такой ячейке на разряд. Да и в заказных БИС затраты невелики. По этой программе с использованием САПР Quarus^{TM} был выполнен синтез одношагового блока в микросхеме программируемой логики семейства Stratix IV. Потребовалось всего около 100 макроячеек, что займет менее 0.1 % ячеек микросхемы даже минимальной емкости семейства Stratix IV! Время обработки одного блока данных составляет 2.5 нс.

Использовать такой специализированный функциональный узел в зависимости от требований производительности и доступного объема оборудования можно разными способами. При реализации в аппаратуре любого сложного алгоритма каждому действию или группе однотипных действий сопоставляется функционально законченный узел, а исполнение действия в узле сопоставляется с шагом алгоритма. Если на некотором шаге используются результаты, полученные на предыдущих шагах, а также, если какая-либо операция не может быть исполнена одним блоком за приемлемое время, прибегают к «дроблению» алгоритма, заключающемуся во временном или/и пространственном разнесении исполнения элементарных действий [5]–[8]. При пространственном разнесении каждой подзадаче (в рассматриваемой постановке – шагу алгоритма) сопоставляется индивидуальный преобразующий блок. При временном распределении действия выполняются последовательно в одном блоке, функции которого от шага к шагу модифицируются под воздействием сигналов управления.

Сначала рассмотрим построение с многократным использованием одного и того же блока, последовательное во времени. Рис. 2 представляет структуру такого устройства. Кроме схемы комбинационного типа, воспроизводящей арифметические и логические преобразования шифруемых данных, содержание и поведение которой соответствует в программе листингу 1, здесь можно выделить регистровую структуру для запоминания промежуточных результатов на время очередного цикла работы, память ключа пользователя (массив user_switch, который заносится из внешнего файла в начале работы) и управляющий автомат. Отметим, что структура почти инвариантна к большинству используемых алгоритмов шифрации (в том числе IDEA).

Алгоритм работы управляющего автомата описывается листингом 2. Автомат имеет 2 состояния. Первое из них – состояние ожидания waitt,

в котором устройство ожидает новый блок данных, а после получения этого блока по сигналу start переходит в рабочий режим и выполняется 32 цикла, причем каждый следующий шаг преобразует результаты, полученные на предыдущем. Циклы отличаются только значениями подключа, которые считываются из памяти. Номер подключа (переменная v) формируется в управляющем автомате в зависимости от порядкового номера цикла *k*.

Листинг 2. Программа работы управляющего автомата.

```
always @ (posedge clock or posedge reset)
begin
if (reset) begin state <=waitt ; ready<=0; end
else case ( state )
waitt: if (start) begin state <= cycle;
n1_in <=data_in [31:0]; n2_in<=data_in [63:32];
ready = 0; k <= 0; p<=0;
end
cycle: if (k == 32 ) begin ready <= 1;
  state<=waitt ;
  end
 else begin n1_in <=n1_out; n2_in<= n2_out; k<=k+1;
if (p==7 ) p=0; //модификация адреса подключа
else p=p+1;
if (k<24) v=p ;
else v=7-p;
 end
endcase;
end //always
```

Хотя здесь используется временное разделение операций, выигрыш в сравнении с архитектурами общего применения весьма ощутим. Это достигается за счет использования специализированных блоков, включая параллельное выполнение подстановок и аппаратную реализацию сдвигов на постоянное число разрядов.

Дальнейшее повышение производительности обеспечивается каскадным соединением блоков. Предельный случай – полный конвейер. Используются 32 одинаковые секции, содержащие модуль coding_block и выходной регистр (рис. 2), причем выходы регистров каждой секции напрямую подключаются к входам n1_in и n2_in следующей секции. Все регистры синхронизируются общим сигналом, чем обеспечивается потактное продвижение информации в устройстве. Каждой секции сопоставлен индивидуальный подключ шифрации. Задержка получения результата обработки одного блока данных такая же, как в структуре с микропрограммным управлением, но при непрерывном потоке данных одновременно в процессе обработки может находиться до 32 таких блоков.

Следует, однако, заметить, что такая структура относительно затратна, а параметры превосходят все мыслимые на сегодня потребности систем шифрования – до 4 млрд символов ASCII в секунду при реализации в ПЛИС семейства Stratix IV.

Промежуточным вариантом по затратам и производительности может стать каскадное соединение секций, каждая из которых последовательно выполняет несколько циклов кодирования (4, 8, или 16). В этом случае подобным образом (вход следующего с выходом предыдущего) объединяются не комбинационные схемы, а микропрограммно-управляемые модули (рис. 3). Каждая секция почти повторяет рис. 2 – надо лишь модифициро-

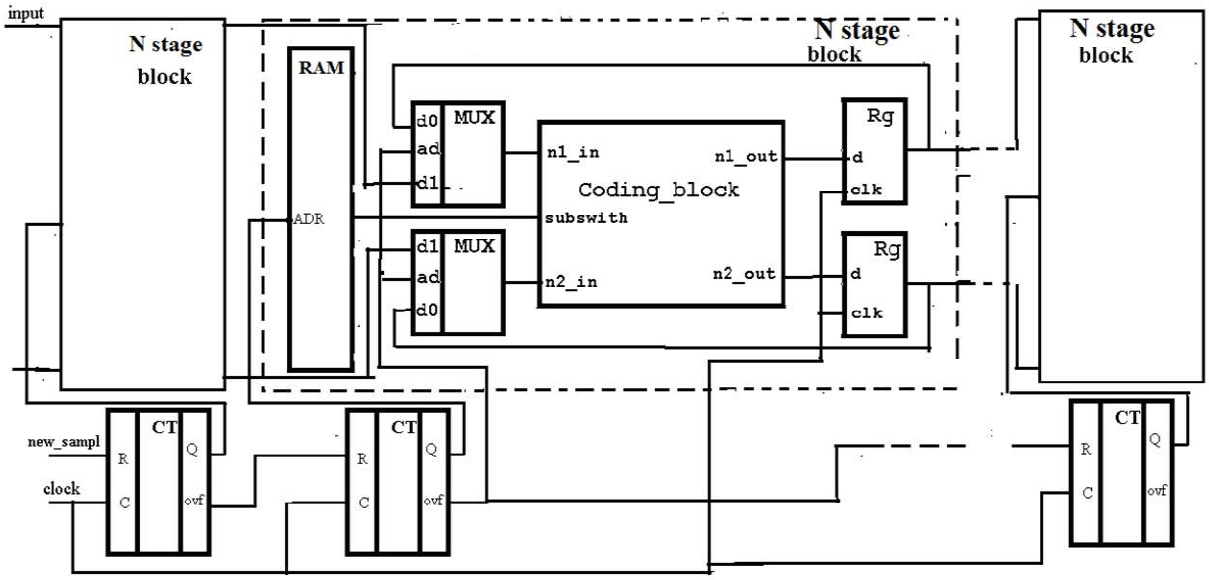

Рис. 3

вать число повторений, а блоки памяти хранят подключи только для тех шагов, которые интерпретируются конкретной секцией.

Каждый **N stage block** реализует N шагов алгоритма шифрации, после завершения которых выдает сигнал готовности на следующий **N stage block,** вызывая в нем исполнение следующих шагов. Поток входных данных не обязательно должен быть непрерывным, необходимо лишь, чтобы интервал поступления на вход блоков данных был не меньше времени обработки блока в одной секции.

Дешифраторы сообщений строятся по таким же схемам, как и шифраторы. Например, для кодирования по ГОСТ 28147–89 таблицы замены и ключи пользователя для процедур шифрования и дешифрования совпадают, различие лишь в порядке выборки подключей.

Экономичность, доступность, высокая производительность, универсальность и повышенная устойчивость к взлому аппаратных устройств шифрации позволяет рекомендовать подход к широкому применению в разнообразных системах.

Предложенные базовые архитектуры при простой модификации схемы исполнителя шага алгоритма позволяют создавать эффективные структуры и устройства для решения задач многих других предметных областей: цифровая обработка сигналов и изображений, функциональные преобразования, решение конечных и дифференциальных уравнений, сжатие данных, анализ сложных структур данных и многое др. В случае возникновения подобных потребностей просим читателей обращаться к авторам.

СПИСОК ЛИТЕРАТУРЫ

1. Основы криптографии / А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин, А. В. Черемушкин. 2-е изд., испр. и доп. М.: Гелиос, 2002.

2. ГОСТ 28147–89. Алгоритм криптографического преобразования. Ввод в действие с 01.01.90.

3. Молдовян А. А., Молдовян Н. А., Советов Б. Я. Криптография. СПб.: Лань, 2001.

4. Schneier B. Appllied cryptology. Jonn Willey & Sons, Inc., 1996.

5. Kaeslin H. Digital Integrated Circuit Design.From VLSI Architectures to CMOS Fabrication. N. Y.: Cambridge University Press, 2008.

6. Угрюмов Е. П. Цифровая схемотехника. 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2004.

7. Грушвицкий Р. И., Мурсаев А. Х., Угрюмов Е. П. Проектирование систем на микросхемах с программируемой структурой. СПб.: БХВ-Петербург, 2006.

8. Мурсаев А. Х. Представление в языках проектирования аппаратуры потоковой, конвейерной и микропрограммной реализации операционных устройств // Изв. СПбГЭТУ «ЛЭТИ». 2010. № 5. С. 73–78.

A. H. Mursaev, V. O. Amelenko
Saint-Petersburg state electrotechnical university «LETI»

HIGH-PERFORMANCE ENCRYPTORS

Hardware implementation of encrypting devices is discussed. Basis approaches to structural organization of operating units and definition of the structure while designing with VerilogHDL are presented in examples.

Encryptors, hardware devices