

# Использование микросхем специальной памяти для обеспечения защиты FPGA от копирования

Дмитрий КОМОЛОВ  
dima@efo.ru  
Роман ЗОЛУТХО  
roman@efo.ru

В статье описывается метод защиты проектов в FPGA от копирования, называемый «определение — друг или враг» (Identification Friend or Foe, IFF). Суть его заключается в том, что функционирование проекта в FPGA не разрешается до тех пор, пока не произойдет совпадения хэш-последовательностей, вычисленных специальным блоком внутри FPGA и во внешней микросхеме специальной памяти. При этом проект остается защищенным даже при перехвате конфигурационного потока, поскольку выделить блок расчета хэш-последовательности в этом случае практически невозможно, а в микросхеме специальной памяти доступ к блоку расчета хэш-последовательности заблокирован. Таким образом, микросхему специальной памяти можно использовать как дополнительную защитную микросхему для FPGA.

## Введение

Большинство семейств СБИС программируемой логики с архитектурой FPGA (Field of Programmable Gate Arrays) выпускаются по технологии статического ОЗУ и требуют конфигурирования после включения питания (для этого служат специализированные внешние ПЗУ). При этом проекты, реализованные на FPGA, уязвимы для копирования, поскольку конфигурационный поток данных может быть перехвачен и использован недобросовестными людьми для несанкционированного повторения проекта. Некоторые семейства FPGA могут использовать закодированный конфигурационный поток для защиты проектов от копирования. Но для этого нужна дополнительная операция программирования декодирующего ключа в энергонезависимую память FPGA, как правило, с использованием дополнительного оборудования. К тому же микросхемы, поддерживающие закодированную конфигурацию, довольно дороги. Большинство семейств FPGA не имеет возможности использования закодированного конфигурационного потока. Для таких семейств эффективным средством защиты проектов от копирования является использование микросхем специальной памяти.

## Реализация

Концепция IFF требует специальной дополнительной микросхемы, в которой реализован хэш-алгоритм. На рис. 1 представ-

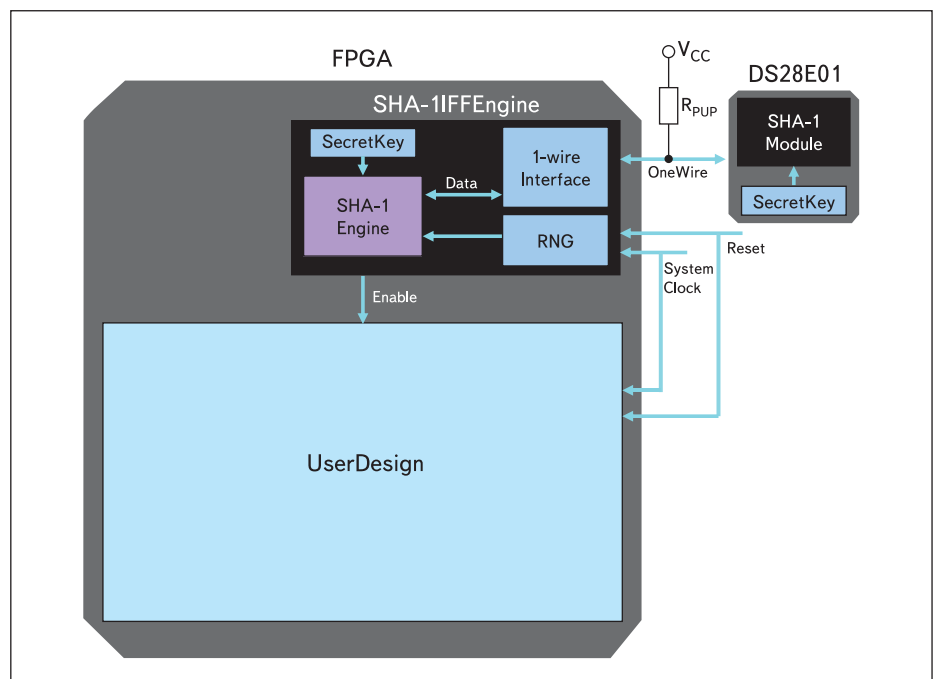


Рис. 1. Реализация концепции IFF

лена реализация IFF с использованием микросхемы специальной памяти DS28E01.

Микросхема DS28E01 фирмы Dallas Semiconductor (в настоящее время данным брендом владеет фирма Maxim Integrated Products, Inc.) содержит 1024 бит EEPROM и специальный блок для аппаратного определения хэш-последовательности в соответствии с алгорит-

мом SHA-1 (Secure Hash Algorithm, стандарт ISO/IEC 10118-3). Хэш-последовательность представляет собой 160-битный аутентификационный код (Message Authentication Code, MAC). Модуль SHA-1 в микросхеме DS28E01 аппаратно вычисляет MAC для массива случайных чисел, сгенерированного специальным модулем в FPGA и записанного в область

данных EEPROM. Для вычисления MAC применяется 64-битный ключ, заранее сохраненный в секретной области EEPROM микросхемы DS28E01.

В качестве дополнительной защиты проекта может использоваться проверка уникального для каждой микросхемы DS28E01 идентификационного номера.

В микросхеме DS28E01 [1] реализован однопроводной интерфейс 1-Wire, поэтому для подключения ее к FPGA нужна всего одна линия ввода/вывода. В соответствии со спецификацией 1-Wire эта линия ввода/вывода FPGA должна быть двунаправленной, и ее выходной каскад должен представлять собой буфер с открытым стоком. Для обеспечения уровня логической единицы линия подтягивается внешним резистором к напряжению питания соответствующего банка ввода/вывода.

Компания Altera предлагает пример разработки (Reference Design), в котором используется концепция IFF для защиты проектов в FPGA семейства Cyclone III от несанкционированного копирования (проект *CIII\_Design\_Security\_Enabler*). На рис. 2 представлена блок-схема этого примера разработки.

В этом примере пользовательский проект представляет собой обычный 8-разрядный двоичный счетчик с входом разрешения. Применяя данный пример разработки в качестве шаблона, пользователь может заменить этот счетчик своим проектом.

В FPGA кроме пользовательского проекта располагается блок аутентификации, реализующий проверку IFF (SHA-1 IFF Engine). Структура этого блока представлена на рис. 3.

Блок аутентификации состоит из трех модулей:

- *RNG\_8bits.vhd* — 8-разрядный генератор случайных чисел;
- *small\_micro\_32.vhd* — модуль, вычисляющий MAC сгенерированного массива случайных чисел в соответствии с алгоритмом SHA-1 и сравнивающий его с MAC, считанным из микросхемы специальной памяти DS28E01;
- *One\_Wire.vhd* — модуль, реализующий однопроводной интерфейс между FPGA и микросхемой DS28E01.

Блок имеет следующие порты ввода/вывода:

- *Clock\_In* — входной тактовый сигнал;
- *Resetn\_In* — входной асинхронный сигнал сброса/перезапуска;
- *One\_Wire* — двунаправленный сигнал для обмена данными по однопроводному интерфейсу с микросхемой DS28E01;
- *Enable* — выходной сигнал, который разрешает или запрещает работу пользовательского проекта.

Пользователь может задавать следующие параметры:

- *SECRET\_KEY* — секретный ключ для вычисления MAC массива случайных чисел. Этот ключ должен совпадать с тем, который записан в микросхему DS28E01.

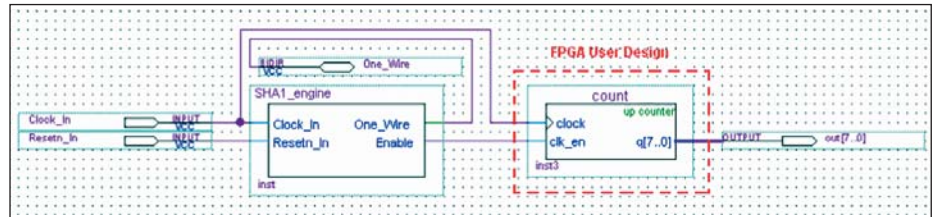


Рис. 2. Блок-схема примера разработки с реализацией защиты проектов в FPGA от копирования

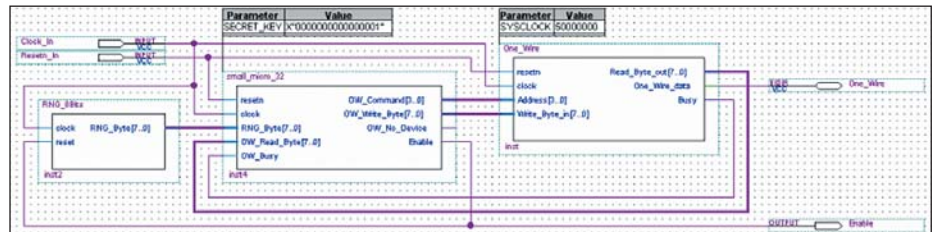


Рис. 3. Структура блока аутентификации (SHA-1 IFF Engine)

- *SYS\_CLOCK* — частота входного тактового сигнала *Clock\_In* (задается в Гц). Этот параметр используется для обеспечения временных требований стандарта 1-Wire. Максимальное значение *SYS\_CLOCK* — 100 МГц.

Блок аутентификации занимает примерно 800 логических элементов и один блок встроенного ОЗУ М9К в микросхеме семейства Cyclone III. По методике, описанной в данной статье, пользователь может разработать

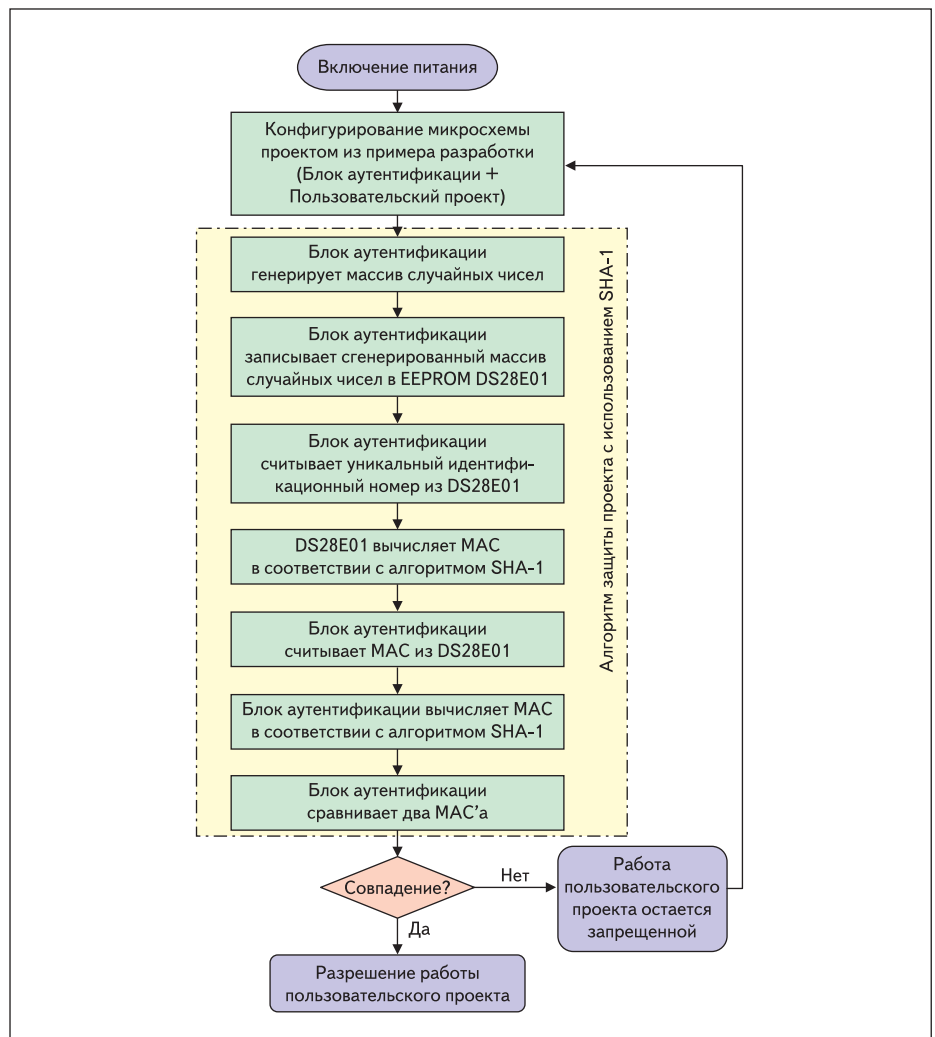


Рис. 4. Алгоритм работы блока аутентификации при реализации концепции IFF\*

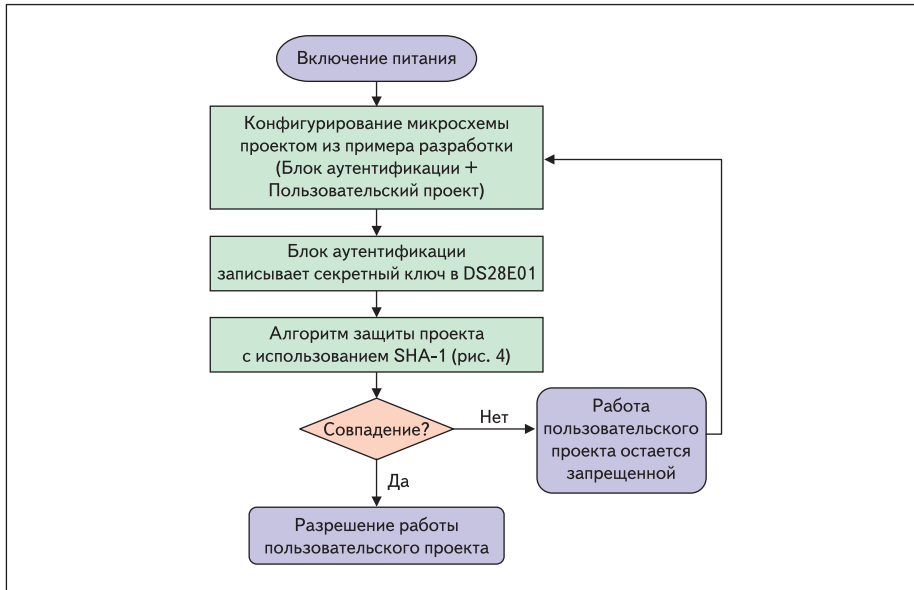


Рис. 5. Алгоритм работы проекта программирования секретного ключа

свой собственный, более компактный блок аутентификации (например, можно использовать компактное процессорное ядро для программной реализации как алгоритма SHA-1, так и протокола 1-Wire).

После включения питания микросхема FPGA конфигурируется потоком данных, содержащим проект пользователя и блок аутентификации. После завершения конфигурирования SHA-1 IFF Engine запрещает работу пользовательского проекта и начинает процесс аутентификации:

- генерирует массив случайных чисел, сохраняет его в памяти FPGA и по однопровод-

ному интерфейсу записывает его в область данных EEPROM микросхемы DS28E01;

- считывает уникальный идентификационный номер из микросхемы специальной памяти;
  - выдает команду микросхеме специальной памяти на вычисление MAC для сохраненного в EEPROM DS28E01 массива случайных чисел;
  - считывает MAC из микросхемы DS28E01 и вычисляет свой MAC для сохраненного в памяти FPGA массива случайных чисел.
- Работа пользовательского проекта разрешается только в том случае, если считанный

из микросхемы DS28E01 и рассчитанный блоком SHA-1 IFF Engine коды совпадают.

На рис. 4 показан алгоритм работы блока проверки при реализации концепции IFF.

После разрешения работы пользовательского проекта блок SHA-1 IFF Engine выключается для снижения потребляемой микросхемой FPGA мощности. Пользователь может повторно запустить блок аутентификации, организовав управление по входу Reset с помощью внешних цепей или управляющего автомата.

Как уже упоминалось, для расчета хэш-последовательности в микросхеме DS28E01 используется ключ, заранее записанный в секретную область памяти. Для записи секретного ключа в DS28E01 в описываемом примере разработки компании Altera имеется специально предназначенный для этого проект *CIII\_Design\_Security\_Load*. Этот проект аналогичен уже рассмотренному *CIII\_Design\_Security\_Enabler*, за исключением того, что блок SHA-1 IFF Engine после завершения конфигурирования FPGA записывает секретный ключ в микросхему DS28E01. Алгоритм работы проекта *CIII\_Design\_Security\_Load* показан на рис. 5.

Проект *CIII\_Design\_Security\_Load* рекомендуется для программирования небольших партий микросхем DS28E01.

Для обеспечения массового производства можно заказать у фирмы-производителя партию микросхем DS28E01 с запрограммированным на фабрике секретным ключом. ■

## Литература

1. [www.maxim-ic.com/DS28E01](http://www.maxim-ic.com/DS28E01)