УДК 004.056.53

ИСПОЛЬЗОВАНИЕ ГЕНЕРАТОРОВ КОНСТАНТ ДЛЯ ВНЕДРЕНИЯ ВОДЯНЫХ ЗНАКОВ В ПРОЕКТНЫЕ ОПИСАНИЯ

Видничук В.Н., аспирант

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Иванюк А.А. – д-р техн. наук, профессор

Представлен способ внедрения водяных знаков в функционально эквивалентные описания аппаратуры на основе генераторов константных значений.

Ключевые слова: генератор константных значений, функционально эквивалентные описания, водяные знаки.

Функционально эквивалентные описания – это один из основных методов функциональной обфускации проектных описаний аппаратуры. Обфускация – это один из методов усложнения понимания исходных кодов проектного описания и разработанного аппаратного обеспечения. Функциональная обфускация – это один из главных видов внутрисхемной обфускации, которая позволяет усложнять понимание проектных описаний и аппаратуры, а также выполнять функцию защиты от несанкционированного доступа и копирования. Для защиты от несанкционированного доступа и копирования используют много различных методов. В данном исследовании предложен метод внедрения водяных знаков и ключей в функционально эквивалентные описания.

Основной проблемой функционально эквивалентных описаний является логический синтез, представляющий собой процесс, посредством которого абстрактное указание желаемой схемы поведения превращается в реализацию конструкций с точки зрения логических вентилей. На данном этапе проводится логическая оптимизация, представляющая собой процесс поиска эквивалентного представления указанной логической схемы с заданными ограничениями. Данный процесс позволяет минимизировать проектные описания, под которые подходят какие-то простейшие логические элементы. Для решения данной проблемы существуют генераторы констант, которые не могут быть распознаны и минимизированы синтезатором, что позволяет применять функционально эквивалентные описания, которые не будут приведены к стандартным.

Генератор констант – это разновидность непрозрачных предикатов, значения которых известны на этапе обфускации, но требуют вычисления при анализе. Они заменяют значения ‘0’ и ‘1’, которые во время синтеза и оптимизации присоединяются к GND и VCC, на схему, сочетающую в себе последовательную и комбинационную логику.

Суть метода внедрения заключается в том, что при подаче на константный генератор ключа, отличающегося от правильного, генератор констант вместо константных значений ‘0’ или ‘1’ может выдавать противоположное или начинать осцилляцию выходного сигнала. Данный метод достигается за счёт добавления фиктивных состояний в конечный автомат, в которые он переходит при неправильном указании ключа. Это позволит при неправильно введённом ключе или изменении внутренних компонентов, в результате несанкционированного доступа и копирования исходной схемы, нарушать её работоспособность. Также поиск причины этой неработоспособности будет затрудняться другими методами функциональной и лексической обфускации. На рисунке 1 представлен один из генераторов таких констант на базе FSM и с входным значением ключа.

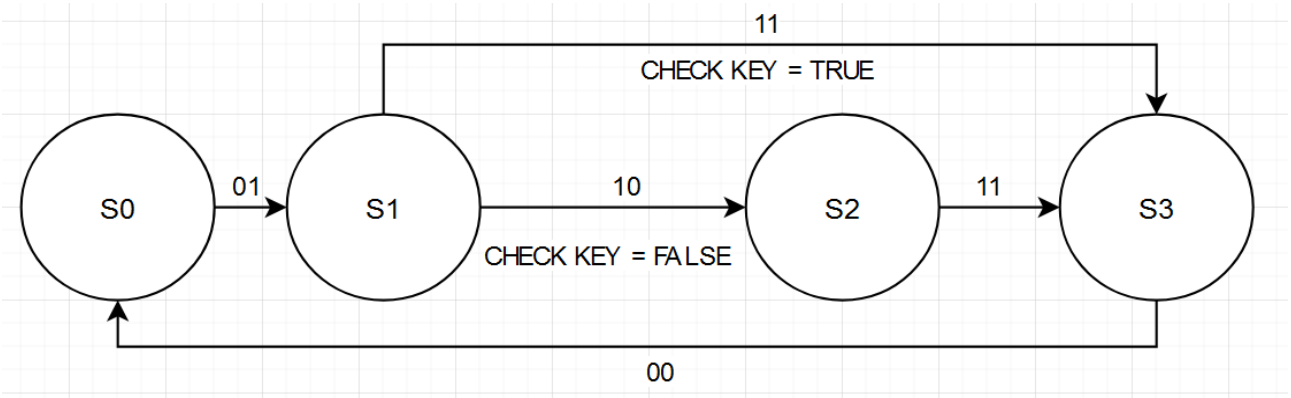

Рисунок 1 – Генератор констант на базе FSM с внедрённым методом проверки ключа

Ключ можно выбирать любой длины, а также его можно формировать различными методами. Например, требовать ввода уникального ключа при первом запуске устройства, хранить данный ключ зашифрованным на самом устройстве или формировать его с помощью других генераторов константных значений. Что позволяет ещё больше затруднить понимание злоумышленниками исходных кодов проектных описаний и схемы аппаратуры. Конечный автомат представлен 4-мя состояниями, каждое из которых генерирует на выход константное значение ‘0’ или ‘1’. Начинается работа данного константного генератора с состояния ‘S0’, в котором генерируется константное значение ‘0’. Далее, если проверка ключа прошла успешно, состояние меняется на ‘S3’, в котором также на выход передаётся значение ‘0’. Если проверка ключа на состоянии ‘S0’ не удалась, конечный автомат переходит в состояние ‘S1’, в котором уже выходное значение становится ‘1’. Далее осуществляется переход в ‘S2’, в котором выходное значение может принимать значение ‘0’ или ‘1’. Отсюда следует, что при использовании данного константного генератора с неправильным ключом начинается последовательная смена значений с ‘0’ на ‘1’ и обратно, что может вызвать неправильную работу схемы. В противном случае на выходе константного генератора всегда константное значение ‘0’.

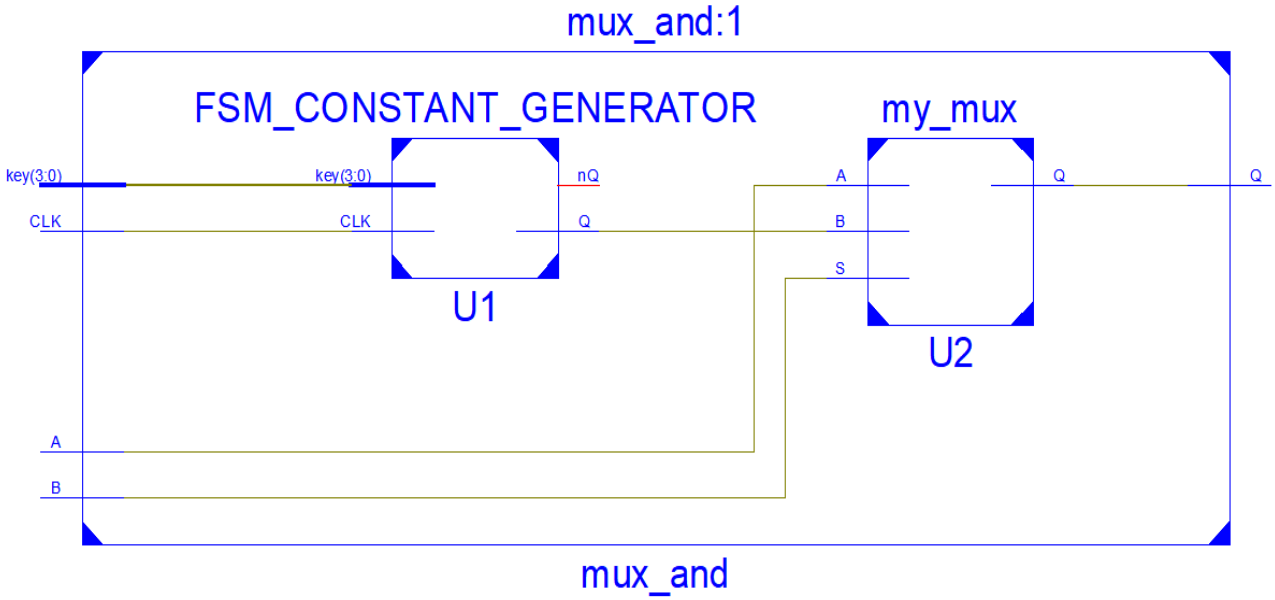

Рисунок 2 – Функционально эквивалентное описание простого логического элемента AND2 с использованием генератора константных значений и мультиплексора

Для примера работоспособности данного константного генератора с внедрённым водяным знаком был использован метод функционально эквивалентных описаний. На рисунке 2 показано

функционально эквивалентное описание простого логического элемента AND2, построенное на мультиплексоре. Элемент ‘FSM_CONSTANT_GENERATOR’ – генератор констант на базе конечного автомата ‘my_mux’ – структурного описания мультиплексора.

На вход генератора констант приходит ключ и сигнал синхронизации, и в зависимости от правильности введённого ключа схема генерирует константный ‘0’ или ошибочное значение. Из-за использования генератора константных значений на этапе логического синтеза минимизации данного элемента не произошло. Если подать на вход ‘B’ мультиплексора константное значение ‘0’ и на входы ‘A’ и ‘S’ значения, предназначенные простому логическому AND2, то мультиплексор начнёт себя вести как элемент AND2. При правильно введённом ключе схема будет работать правильно, что показано на рисунке 3.

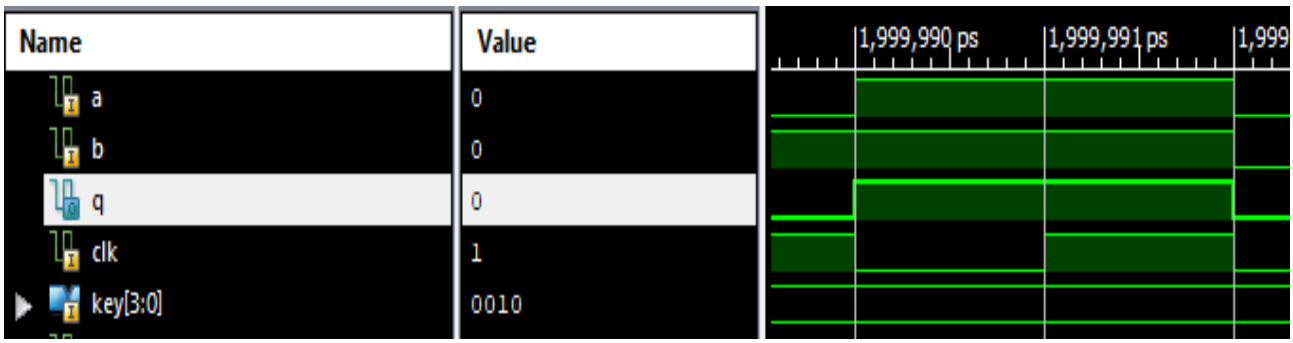

Рисунок 3 – Симуляция работы функционально эквивалентного описания AND2 с использованием генератора констант и правильно введённым ключом

Однако, если ввести неправильный ключ, то значение, подаваемое на вход ‘B’, будет принимать не только константное значение ‘0’, но и ‘1’. Это приведёт к ошибочным результатам работы данной схемы. Результат выхода данной схемы с неправильно введённым ключом показан на рисунке 4. В какой-то момент времени конечный автомат, отвечающий за генерацию константных значений, переходит в фиктивное состояние, которое соответствует неправильному выходу, и схема начинает генерировать ошибочное значение, которое приведёт к неправильному результату работы.

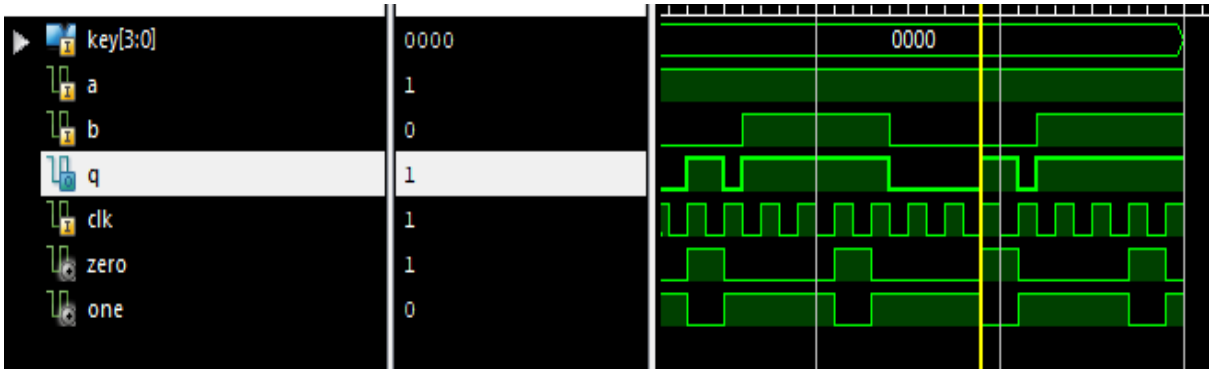

Рисунок 4 – Симуляция работы функционально эквивалентного описания AND2 с использованием генератора констант и неправильно введённым ключом

Данный подход использования генераторов констант можно применить при формировании функционально эквивалентных описаний остальных простых логических примитивов. Например, для простого логического элемента NOT на вход ‘A’ мультиплексора подаётся константное значение ‘0’, на вход ‘B’ подаётся значение ‘1’ и на селектирующий подаётся входное значение элемента NOT. Пример реализации данного элемента показан на рисунке 5.

Для элемента OR2 на вход ‘A’ подаётся значение ‘1’, на входы ‘B’ и ‘S’ подаются входные значения элемента OR2. Также можно на базе мультиплексора реализовать элементы NAND, NOR, XOR, XNOR, LATCH. Внедрение в эти функционально эквивалентные описания генераторов константных значений, использующие водяные знаки, позволит усложнить анализ исходных проектных описаний и затруднит понимание логики работы схемы, и при неправильно введённом ключе работоспособность схемы нарушается. Можно ещё затруднить анализ схемы путём добавления большего количества фиктивных состояний в генератор констант на базе FSM. Водяной знак может быть встроен в другие виды генераторов констант.

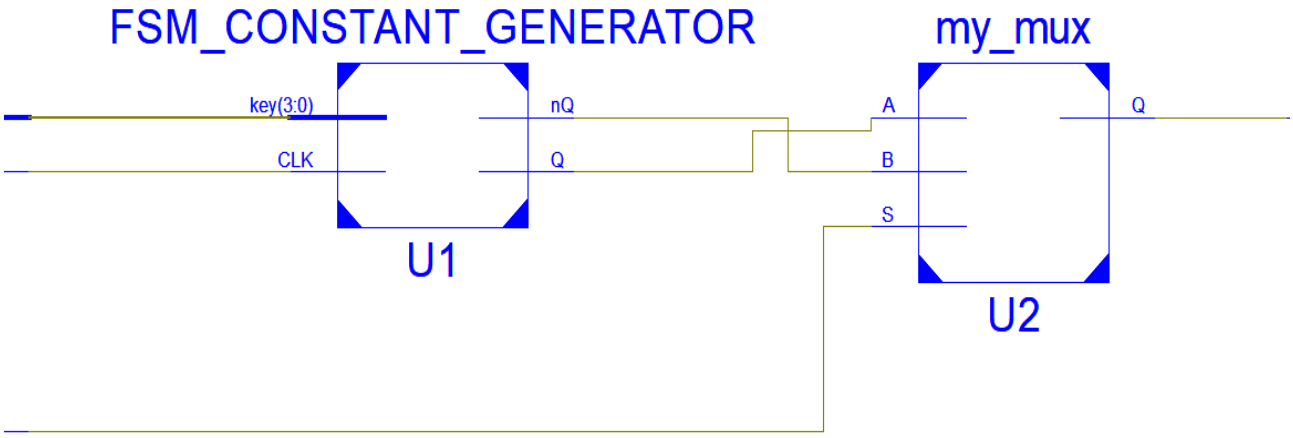

Рисунок 5 – Функционально эквивалентное описание простого логического элемента NOT с использованием генератора константных значений и мультиплексора

По полученным в результате исследования данным можно сказать, что приведённый метод использования генераторов констант и внедрения в них водяных знаков позволяет усложнить анализ исходных кодов проектных описаний аппаратуры, а также при несанкционированном доступе к описаниям нарушить работоспособность схемы. Это позволяет применять данный метод как механизм защиты проектных описаний от несанкционированного использования.

Список использованных источников:

1. Chakraborty, R. S. Hardware Security through Design Obfuscation : Ph. D. diss. / R. S. Chakraborty. – Cleveland, 2010. – 167 p.

2. Collberg, C. A Taxonomy of Obfuscating Transformations / C. Collberg, C. Thomborson, D. Low. – Auckland : Department of Computer Science, 1997. – 36 p.

3. Сергейчик, В. В. Базовые примитивы схемной обфускации цифровых устройств / В. В. Сергейчик // материалы Международной научно-технической конференции, приуроченной к 50-летию МРТИ – БГУИР, Минск, 18–19 марта 2014 г. : в 2 ч. / БГУИР. – Минск, 2014. – Ч. 1. – С. 442–443.

4. Иванюк, А. А. Физическая криптография и защита цифровых устройств / А. А. Иванюк, С. С. Заливако // Доклады БГУИР. – 2019. – №2 (120). – С. 50–58.

UDC 004.056.53

USING A CONSTANT GENERATORS FOR EMBEDDING A WATERMARK INTO PROJECT DESCRIPTIONS

Vidnichuk V.N.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Ivaniuk A.A. – Doctor of Sciences in Technology

A method for introducing watermarks into functionally equivalent descriptions of equipment based on constant value generators is presented.

Keywords: constant value generator, functionally equivalent descriptions, watermarks.