ИСПОЛЬЗОВАНИЕ ГЕНЕРАТОРОВ КОНСТАНТ ДЛЯ ФОРМИРОВАНИЯ НЕОПТИМИЗИРУЕМЫХ VHDL ОПИСАНИЙ

Видничук В.Н.

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

Иванюк А.А. – доктор технических наук

Рассматриваются генераторы констант как один из методов внутрисхемной обфускации за счёт формирования не оптимизируемых VHDL описаний аппаратуры.

В связи с быстрым ростом производства аппаратуры и цифровых устройств растут и убытки связанные с пиратством. По последним данным ущерб составляет 17 млрд долларов в год. Также данная проблема влияет и на безопасность и надежность данных устройств. Следует выделить следующие угрозы: пиратство, внедрение аппаратной прослушки, аппаратные трояны, пиратство, реверс инжиниринг. Для борьбы с этими угрозами разрабатываются различные методы. Одним из главных методов – обфускация.

Обфускация это один из методов запутывания понимания функционирования схемы с целью защиты от обратного проектирования. Сложность и затраты на обратное проектирование схемы должно стремиться к максимальному значению и должны быть сопоставимы с затратами на разработку схемы с нуля. Также обфускацию можно использовать для внедрения водяных знаков.

В случае обфускации VHDL описаний рассматривают следующие виды обфускации: лексическая и функциональная.

Лексическая обфускация это изменение исходных проектных описаний для затруднения понимания их злоумышленниками, основным недостатком которой является то, что результат синтеза такого описания не изменится и при обратном проектировании злоумышленник может получить необфусцированное проектное описание[1].

Функциональная обфускация используется для получения более сложных схем путём применения функционально эквивалентных описаний. Функционально эквивалентное описание это описание которое по своей логике работает также но отличается по результатам синтеза. Основной проблемой при таком описании является оптимизатор. Оптимизация это процесс минимизации логических уравнений и приведения их к их минимальным формам. Проблема заключается в том что если оптимизатор видит одинаковые таблицы истинности то он приводит эти функции к одинаковым и результат синтеза(схема) будет одинаковой. Одним из решений проблемы может быть замена константных ‘0’ и ‘1’ генераторы констант. Так как оптимизатор не будет знать точную входную таблицу истинности он не сможет оптимизировать описание.

Генератор констант это разновидность непрозрачных предикатов, значение которых известны на этапе обфускации но требуют вычисления при анализе[2]. Они заменяют значения ‘0’ и ‘1’ что во время синтеза и оптимизации присоединяется к GND и VCC на схему сочетающую в себе последовательную и комбинационную логику. Данная схема разработана таким образом, чтобы генерировать константный ‘0’ или ‘1’ но при этом выглядеть как схема с памятью. Пример такой схемы можно увидеть на рис. 1.

На данном рисунке видно, что константный ‘0’ заменён на схему с памятью, которая при любых входных значениях будет генерировать ‘0’. Генераторы констант можно использовать для замены простых логических примитивов на мультиплексор, повторяющий логическое уравнение данного примитива. Пример такой замены логического примитива AND представлен на рисунке 2. Где U1 это генератор константного ‘0’ а U2 мультиплексор.

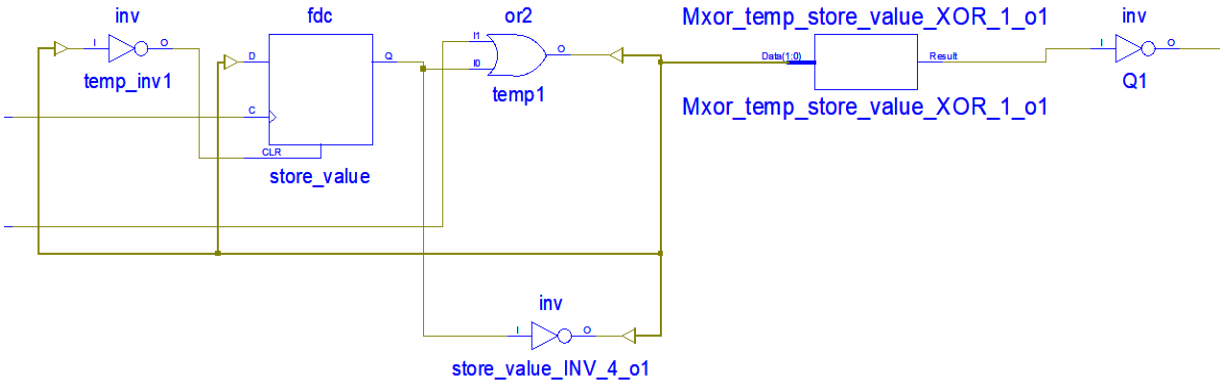

Рисунок 1 – схема не оптимизируемого генератора константного ‘0’.

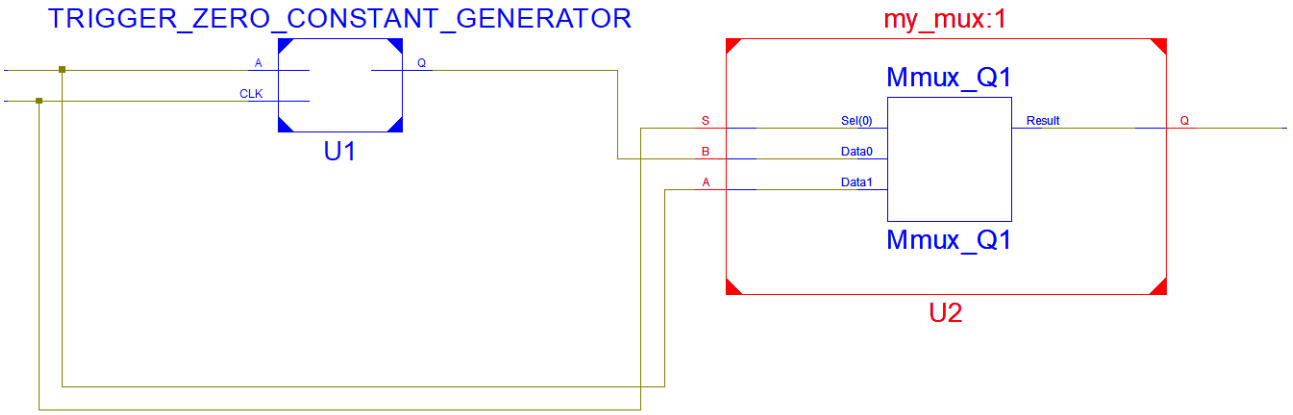

Рисунок 2 – пример схемы обфусцированного логического примитива AND с использованием генератора констант и мультиплексора.

Данная схема является не оптимизируемой из за того, что оптимизатор думает на выходе из генератора констант может выйти не ‘0’ значение. Если использовать обычный константный ноль то оптимизатор минимизирует логическое выражение и на выходе будет обычный AND. На рисунке 3 показан результат технологического синтеза данного описания.

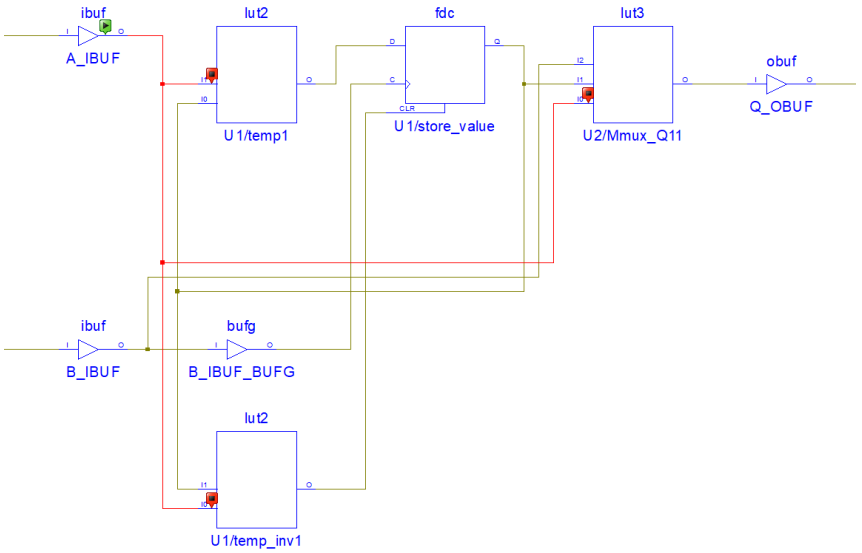

Рисунок 3 – результат технологического синтеза обфусцированного описания AND.

Результат синтеза подтверждает возможность использования функционально эквивалентных описаний для обфускации и помогает усложнить понимание схем, однако он имеет свои недостатки, например в связи с ростом количества элементов возрастают и общие задержки на схеме. Для увеличения сложности понимания имеет смысл использовать лексическую и функциональную обфускацию вместе.

Список использованных источников:

1. Видничук, В. Н. Лексическая обфускация как способ внедрения водяных знаков в исходные коды программ и проектных описаний / Видничук В. Н., Иванюк А. А. // Информационные технологии и системы 2019 (ИТС 2019) = Information Technologies and Systems 2019 (ITS 2019) : материалы международной научной конференции, Минск, 30 октября 2019 г. / Белорусский государственный университет информатики и радиоэлектроники; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2019. – С. 136 – 137.

2. Сергейчик, В. В. Генераторы констант как базовые примитивы схемной обфускации / В. В. Сергейчик // Компьютерные системы и сети: материалы 50-й научной конференции аспирантов, магистрантов и студентов (Минск, 24-28 марта 2014 г.). – Минск : БГУИР, 2014. – С. 78 - 79.