Синтез частично программируемых схем, ориентированный на маскирование вредоносных подсхем (Trojan Circuits)

А.Ю. Матросова <mau11@yandex.ru>
С.А. Останин <sergeiostanin@yandex.ru>
Е.А. Николаева <nikolaeve-ea@yandex.ru>
Национальный исследовательский Томский государственный университет, 634050, Россия, Томск, пр. Ленина, д. 36

Аннотация. При синтезе современных интегральных схем разработчики все чаще прибегают к услугам сторонних фирм для реализации тех или иных компонент системы (Intellectual Property cores, перепрограммируемых компонент на базе FPGA и т.д.) с целью снижения ее стоимости. В компонентах, изготовленных сторонними фирмами, могут быть спрятаны вредоносные подсхемы (Trojan circuits) с целью разрушения системы или извлечения из нее конфиденциальной информации. Trojan Circuits (TCs) обычно действуют в ситуациях, которые возникают в работающей системе чрезвычайно редко, поэтому они не обнаружимы ни в процессе верификации системы, ни в процессе ее тестирования. В работе предлагается подход к проектированию частично программируемых схем из вентилей, программируемых блоков памяти (LUTs) и программируемых мультиплексоров (MUXs), ориентированный на маскирование TCs. Такой подход к синтезу позволяет либо замаскировать действие TC в случае ее обнаружения, либо получить схему, в которой эффективное введение TCs становится невозможным. Предложен способ перепрограммирования блоков памяти LUTs для маскирования TC. Сформулировано требование к замещающей функции, поступающей на свободный вход программируемого блока, основанное на анализе частичных функций внутренних полюсов комбинационной схемы. Построение частичных функций выполняется с использованием операций над Reduced Ordered Binary Decision Diagrams (ROBDD-графами), строящимися для фрагментов схемы. Операции характеризуются полиномиальной сложностью.

Ключевые слова: частично программируемые схемы; вредоносные схемы (Trojan Circuits); частичные булевы функции; Reduced Ordered Binary Decision Diagrams (ROBDD-графы).

DOI: 10.15514/ISPRAS-2017-29(5)-4

Для цитирования: Матросова А.Ю., Останин С.А., Николаева Е.А. Синтез частично программируемых схем, ориентированный на маскирование вредоносных подсхем (Trojan Circuits). Труды ИСП РАН, том 29, вып. 5, 2017 г., стр. 61-74. DOI10.15514/ISPRAS-2017-29(5)-4

1. Введение

Вредоносные подсхемы (Trojan Circuits) могут вводиться в компоненты интегральной схемы с целью разрушения схемы или извлечения из нее конфиденциальной информации. Trojan Circuits (TCs) обычно действуют в ситуациях, которые возникают в условиях функционирования схемы чрезвычайно редко, поэтому они не обнаружимы ни в процессе верификации, ни в процессе тестирования схемы [1, 2]. Вредоносная подсхема (TC) состоит из двух частей. Триггерная подсхема (Trojan trigger) включается при поступлении на ее входы определенной комбинации значений сигналов. Вторая часть подсхемы (Trojan payload) является исполнительным устройством, включаемым триггерной подсхемой, которое может либо разрушить работу схемы, либо извлечь из нее секретную информацию. Такие вредоносные подсхемы необходимо обнаруживать, и, по возможности, нейтрализовать их действие.

Рассматривается проблема маскирования вредоносных подсхем (Trojan Circuits) в логических схемах, состоящих из вентилей, путем введения в нее программируемых блоков памяти LUTs (Look up Tables) и программируемых мультиплексоров (MUXs). В LUT могут быть свободные входы (в данной работе допускается один свободный вход), которые можно использовать для коррекции схемы с целью маскирования в ней воздействия вредоносной подсхемы. Исследуются возможности использования свободного входа в условиях перепрограммирования LUT и MUXs либо для маскирования TC в случае ее обнаружения (в этой ситуации перепрограммируется только один соответствующий LUT), либо для перепрограммирования нескольких LUTs (не обязательно всех) таким образом, что включение в схему вредоносных подсхем оказывается неперспективным. Здесь речь идет о превращении исходной логической схемы из вентилей в схему, защищенную от вредоносных подсхем. Это значит, что при сохранении спецификации схемы введение в нее TCs оказывается неэффективным – они с большой вероятностью обнаруживаются в процессе верификации схемы или в результате ее тестирования. Типы вредоносных подсхем и способы их включения в схему могут быть различными. В работе для удобства рассматриваются TCs, вводимые в линию связи между элементами схемы, так что значение на выходе этой линии в условиях активизации входа TC заменяется противоположным. Однако предлагаемый подход годится для произвольного включения TC, важно лишь, что выход вредоносной подсхемы подключен к некоторой линии. Активизация TC обеспечивается достижением соответствующего входного состояния комбинационной схемы (полного состояния, если речь идет о комбинационном

эквиваленте последовательностной схемы). Перепрограммируемые мультиплексоры применяются с целью маскирования нескольких TCs одним и тем же перепрограммируемым LUT.

Предполагается, что исходная схема состоит только из вентилей и реализует заданную спецификацию – систему полностью определенных булевых функций. Требуется покрыть некоторые ее подсхемы программируемыми блоками памяти (LUTs) таким образом, чтобы обеспечить либо маскирование TC в случае ее обнаружения, либо сделать схему более защищенной от вредоносных подсхем. Задача сводится к маскированию линий в схеме, константные неисправности которой трудно обнаружимы, то есть множества тестовых наборов для таких неисправностей малы и не превышают некоторого заданного порога. Предполагается, что именно такие линии удобно использовать для подключения к ним TCs.

В работах [3, 4] схема сначала строится из вентилей. Затем некоторые ее подсхемы покрываются программируемыми блоками (LUTs), так что один вход LUTs, не обязательно всех, остается неиспользованным. Выделяется множество линий, константные неисправности которых необходимо замаскировать одним из двух способов. Возможность маскирования конкретным LUT конкретной линии сводится к определению выполнимости соответствующей Quantified Boolean Formula (QBF). Речь идет о квантифицированной конъюнктивной нормальной форме. Используются различные решатели (QBF-solvers), автоматизирующие процесс. Проблема выполнимости такой формулы относится к PSPACE-полным. Зарубежные исследователи отмечают, что решатели для QBF формулы работают гораздо медленнее, чем решатели, анализирующие конъюнктивную нормальную форму (КНФ) на выполнимость. В случае возможности маскирования определяется способ перепрограммирования маскирующего LUT. В проектируемой схеме проводится дополнительная линия, связывающая свободный вход LUT с выходом элемента схемы (вентиля или другого LUT), который совместно с перепрограммируемым LUT может маскировать неисправность соответствующей линии.

Недостатком такого подхода является, на наш взгляд, отсутствие стратегии выбора подсхем для покрытия их программируемыми блоками (LUTs) и отсутствие формальных критериев возможностей маскирования. Данная работа ориентирована на преодоление этих недостатков. С этой целью используются и анализируются частичные функции, сопоставляемые внутренним полюсам и линиям схемы. Частичные функции вычисляются путем выполнения операций над ROBDD-графами, построенными для фрагментов заданной комбинационной схемы. Такие операции, как известно, характеризуются полиномиальной сложностью. Использование QBF solvers не требуется.

Мы предлагаем сначала найти линии в схеме, неисправности которых трудно обнаружимы. В эти линии могут быть включены вредоносные подсхемы. Поведение активированной TC аналогично проявлению константной

неисправности линии на подмножестве входных наборов схемы (возможно, только на одном), которые не обязательно являются тестовыми наборами этой неисправности в условиях отсутствия в схеме вредоносных подсхем. Выделенные линии маскируются LUTs с фиксированным числом входов, один из которых является свободным. Если маскируется одна из линий множества при подключении к ней TC, то действие подключенной к ней вредоносной подсхемы нейтрализуется. В этой ситуации соответствующий программируемый блок (LUT) перепрограммируется. Если маскируются все или почти все линий выделенного подмножества, то в результате строится схема, устойчивая относительно введения в нее вредоносных подсхем.

Предлагаемый подход основан на вычислении частичных булевых функций внутренних полюсов и линий комбинационной схемы с помощью операций над ROBDD-графами, построенными для ее подсхем.

2. Постановка задачи

Рассматривается комбинационная схема C (комбинационная составляющая последовательностной схемы), состоящая из вентилей. Используются способы (рис. 1, рис. 3) маскирования одиночных константных неисправностей на линиях связей между логическими элементами с помощью LUTs и MUXs, предложенные в работах [3, 4]. Число входов программируемого блока (LUT) фиксировано. Один из входов может быть свободным, не используемым при покрытии вентилей схемы C. Для заданного множества линий требуется выполнить покрытие фрагментов схемы C из вентилей программируемыми блоками, так чтобы при последующем их перепрограммировании замаскировать как можно большее количество линий, к которым возможно подключение вредоносных подсхем (TCs). Договоримся в дальнейшем схему из вентилей и схему, в которой некоторые подсхемы покрыты программируемыми блоками, обозначать одним и тем же символом C.

Будем иметь в виду, что каждому внутреннему полюсу v комбинационной схемы (схема реализует функцию f, являющуюся спецификацией) в общем случае сопоставляется частичная функция f_v от входных переменных схемы C, представляемая множествами $M_1(f_v)$ ($M_0(f_v)$) нулевых и единичных наборов значений входных переменных. Полностью определенную функцию внутреннего полюса v будем обозначать символом $f(v)$. Полюс v является выходом подсхемы C_v, входы этой подсхемы совпадают с входами схемы C. Частичность функции подсхемы C_v возникает за счет окружения ее элементами схемы C.

Пусть C является одно выходной комбинационной схемой. Рассмотрим способ маскирования, представленный на рис. 1.

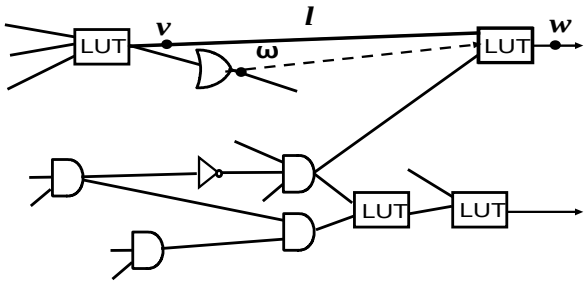


Рис. 1. Маскирование линии l, подключенной к входу LUT
Fig. 1. Masking line l that is LUT input

Полюс v соединен линией l с входом u_i программируемого блока LUT, покрывающего подсхему C_{LUT} из вентилей. На рис. 1 линия l выделена жирным. Входы подсхемы C_{LUT} являются либо входными, либо внутренними переменными схемы C. Число входных переменных LUT фиксировано, одна из переменных свободна, а функция от оставшихся переменных реализует подсхему из вентилей C_{LUT}, покрывающую соответствующий фрагмент из вентилей схемы C. Пусть в линию l включена вредоносная подсхема. Будем маскировать ее, используя свободный вход LUT. В дальнейшем этот LUT будем называть корректирующим. Обозначим его выход символом w. Отметим, что позиция пунктирной линии на рис. 1 заранее не определена, она может соединять свободный вход LUT с выходом другого элемента схемы, при выполнении определенных условий.

Обозначим символом m число входов корректирующего программируемого блока (LUT), тогда $(m-1)$ – число используемых входов в предположении, что в схеме отсутствуют вредоносные подсхемы. Среди используемых входов находится вход u_i, соединенный линией l с полюсом v. Для определенности будем считать, что свободный вход имеет номер m. Пусть в линию l включена вредоносная подсхема. Она изменяет подсхему C_w и функцию, вычисленную по структуре этой подсхемы. Напомним, что входами подсхемы C_w являются входы схемы C. Частичная функция f_w^* , сопоставляемая полюсу w в присутствии ТС, также изменяется, поскольку вредоносная подсхема в условиях активации изменяет реакцию схемы на некоторых ее входных наборах, возможно, только на одном. В этом случае необходимо маскировать линию l за счет использования свободного входа корректирующего LUT. В дальнейшем покажем, что маскирование линии l способом, представленным на рис. 3, аналогично. Сначала введем ряд необходимых понятий.

3. О построении частичной функции и ее реализации

Выделим в схеме C внутренний полюс v, являющийся выходом вентиля схемы или некоторого LUT и сопоставляемый исходящей из него линии l. Построим

по схеме C подсхему C_l, объявив линию l входом этой подсхемы наряду с переменными $x_1,\ldots, x_n$. Схема C_l получается из схемы C обрывом линии l и устранением всех элементов схемы C, связанных с линией l и не связанных с выходом схемы C. Сопоставим подсхеме C_l ROBDD-граф $R(C_l)$ от переменных $x_1, \ldots, x_n, l$. В графе $R(C_l)$ переменная l выбирается первой при разложении Шеннона с целью построения этого графа.

Будем иметь в виду, что полюсу v, из которого исходит линия l, сопоставляется полностью определенная функция $f(v)$, реализуемая подсхемой C_v (она представляется ROBDD-графом $R(C_v)$), выходом которой является полюс v, а входами – переменные $x_1, \ldots, x_n$. Этому полюсу сопоставляется также частичная функция $\mu(x_1, \ldots, x_n)$, реализуемая на полюсе v от тех же входных переменных, в условиях, когда подсхема C_v является частью схемы C. Частичная функция определена на некоторых единичных $M_1(f_v)$ (нулевых $M_0(f_v)$) наборах полностью определенной функции $f(v)$. Только на этих наборах изменение значения функции $f(v)$ влияет на значение функции f, реализуемой схемой C.

Множества единичных и нулевых наборов частичной функции предложено представлять двумя ROBDD-графами $R_1(v)$ и $R_0(v)$. Построение частичной функции сводится к поиску всех тестовых наборов для константных неисправностей полюса v. Множество всех тестовых наборов для неисправности константа 0 есть множество $M_1(f_v)$, а множество всех тестовых наборов для неисправности константа 1 есть множество $M_0(f_v)$ частичной функции $\mu(x_1, \ldots, x_n)$. Метод построения ROBDD-графов для множеств единичных и нулевых наборов частичной функции внутреннего полюса схемы представлен в работах [5, 6].

Обозначим символами $R_l, R_{\bar{l}}$ ROBDD-графы, полученные из графа $R(C_l)$ следующим образом. Корнем графа R_l, является вершина, в которую заходит дуга, сопоставляемая переменной l графа $R(C_l)$, а корнем графа $R_{\bar{l}}$ является вершина, в которую заходит дуга, сопоставляемая инверсии этой переменной.

Будем иметь в виду, что ROBDD-граф $\overline{R}$ получается из ROBDD-графа R переименованием терминальных вершин: 1 – терминальная вершина становится 0 – терминальной вершиной и наоборот.

Тогда граф $R_1(l)$, представляющий множество тестовых наборов для неисправности константа 1 на линии l, вычисляется по формуле:

$$R_1(l) = (R_l \wedge \overline{R_{\bar{l}}} \vee \overline{R_l} \wedge R_{\bar{l}}) \wedge \overline{R(C_v)}\,,$$

а граф $R_0(l)$, представляющий множество тестовых наборов для неисправности константа 0 на линии l, вычисляется по формуле:

$$R_1(l) = (R_l \wedge \overline{R_{\bar{l}}} \vee \overline{R_l} \wedge R_{\bar{l}}) \wedge R(C_v)\,.$$

Оба графа задают частичную функцию линии l. Если полюс v не является точкой ветвления, то частичные функции линии l и полюса v совпадают.

Рассмотрим частичную функцию f_1 и полностью определенную функцию f_2, представленные парами множеств единичных и нулевых наборов значений своих переменных: $M_1(f_1)$, $M_0(f_1)$; $M_1(f_2)$, $M_0(f_2)$. Будем говорить, что полностью определенная функция f_2 реализует частичную функцию f_1 , если выполняются условие: пересечения множеств $M_1(f_1)$, $M_0(f_2)$ и множеств $M_0(f_1)$, $M_1(f_2)$ пусты. Это значит, что $M_1(f_2)$ содержит $M_1(f_1)$ и $M_0(f_2)$ содержит $M_0(f_1)$.

Из определения следует, что полностью определенная функция $f(v)$ является реализацией функции $\mu(x_1, \ldots, x_n)$.

Утверждение 1. Подстановка вместо переменной v любой реализации функции $\mu(x_1, \ldots, x_n)$ сохраняет функцию f схемы C.

Доказательство следует из определения реализации частичной функции.

Частичная функция f_2 поглощает частичную функцию f_1, если выполняются следующие условия:

1) $M_1(f_2)$ содержит $M_1(f_1)$ и $M_0(f_2)$ содержит $M_0(f_1)$;

2) пересечение множеств $M_1(f_1)$, $M_0(f_2)$ и множеств $M_0(f_1)$, $M_1(f_2)$ пусто.

Отношение поглощения частичных функций иллюстрируется на рис. 2.

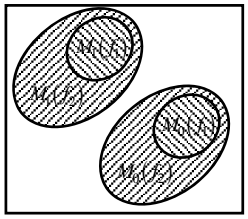


Рис. 2. Функция f_2 поглощает f_1
Fig. 2. Function f_2 covers *f_1*

Может оказаться, что при покрытии схемы из вентилей программируемыми блоками нет возможности подключить линию l к входу LUT. Тогда воспользуемся способом маскирования (рис. 3), предложенным в работах [3, 4]. При таком способе маскирования линия l лежит на пути, проходящем через элементы схемы, связывающем эту линию с входом u_i корректирующего LUT. На рис. 3 неисправная линия выделена жирным шрифтом.

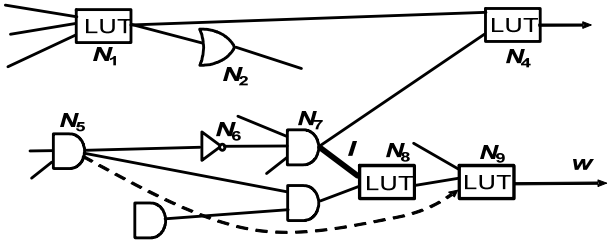


Рис. 3. Маскирование линии l, соединенной через элементы пути с входом корректирующего LUT
Fig. 3. Masking line l that is connected by path with input of correcting LUT

Для обоих способов маскирования линии l (рис. 1, рис. 3) справедливо следующее утверждение.

Утверждение 2. Спецификация схемы C (функция f) сохраняется если и только если частичная функция f_w^* поглощает частичную функцию f_w.

Доказательство очевидно, и следует из определения частичной функции.

Коррекцию будем выполнять, следуя утверждениям 1 и 2, причем, коррекция не должна менять функций элементов схемы C, входящих в окружение подсхемы C_w. Будем иметь в виду, что способы маскирования, представленные на рис. 1, 3, удовлетворяют этому условию.

4. Маскирование неисправности

Для способа, предложенного на рис. 1, и способа, предложенного на рис. 3, будем записывать в корректирующий программируемый блок единичные наборы полностью определенной функции, реализуемой подсхемой из вентилей, покрытой этим блоком и зависящей от его входных переменных, а именно от его $(m-1)$ переменной. Каждому единичному набору сопоставляется два набора в пространстве m переменных: один с единичным значением переменной u_m, другой – с нулевым значением этой переменной.

При перепрограммировании LUT формируем функцию

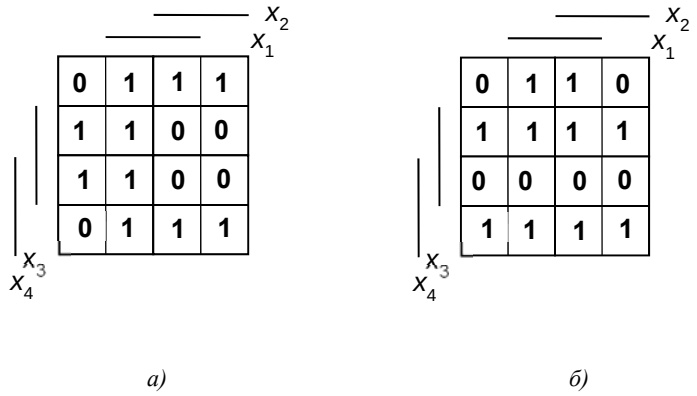
$$f_{LUT}(u_i = 1) \wedge u_m \vee f_{LUT}(u_i = 0) \wedge \overline{u_m}$$

из единичных наборов функции корректирующего LUT. Эти наборы получены непосредственно по структуре подсхемы, покрытой корректирующим LUT, и теперь представляют функцию этого LUT в пространстве $u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_m$ его входных переменных. Она отличается от исходной функции LUT тем, что в ней переменная u_i заменена переменной u_m, и переменная u_m, (как прежде переменная u_i) теперь является существенной. С целью маскирования ТС делаем переменную u_i несущественной. Это значит, что каждому единичному

набору функции $f_{LUT}(u_i = 1) \wedge u_m \vee f_{LUT}(u_i = 0) \wedge \overline{u_m}$ необходимо сопоставить два набора в пространстве m переменных: один с единичным значением переменной u_i, а другой – с нулевым значением этой переменной. Тогда ТС, подключенная к входу u_i, не может изменить корректного поведения схемы C.

На рис. 4 приведен пример исходной функции корректирующего LUT (рис. 4а) и функции, представляющей результат коррекции (рис. 4б). Здесь вход u_i сопоставляется переменной x_2, а переменная x_4 соответствует свободному входу корректирующего LUT.

В результате перепрограммирования корректирующий LUT реализует функцию от (m – 1) переменной, отличающуюся от его прежней функции тем, что выполнена замена переменных (u_i заменена на u_m). Далее вместо u_m (рис. 1) подставляется полностью определенная функция, реализующая частичную функцию $\mu(x_1, \ldots, x_n)$ полюса v. Полностью определенная функция реализуется на полюсе ω.



а) *б)*

Рис. 4. Перепрограммирование LUT
Fig. 4. Reprogramming LUT

Поскольку функция корректирующего LUT получена из исходной заменой переменной u_i на u_m, а полностью определенная функция, сопоставляемая u_m, соединенным с полюсом ω, реализует частичную функцию, сопоставляемую переменной u_i, то на основании выше приведенных утверждений заключаем, что схема C реализует в результате коррекции ту же функцию f, то есть спецификация схемы C сохраняется. Это значит, что подключение ТС к линии l не изменяет функцию f.

Заметим, что входу u_i в конструкции, представленной на рис. 3, не обязательно сопоставляется слабо определенная частичная функция, как в случае, когда вход u_i инцидентен линии, неисправность которой трудно обнаружима (рис. 1).

5. Синтез схемы

Покрытие исходной вентильной схемы программируемыми блоками памяти (LUTs) с целью маскирования вредоносных подсхем (TCs) выполняем следующим образом. Определяем множество L линий, неисправности которых трудно обнаружимы. Двигаемся в порядке, например, не возрастания мощности тестовых наборов частичных функций. Мощности единичных и нулевых наборов частичной функции линии не должны превышать заданного порога. Находим линию l, сопоставляемую трудно обнаружимым константным неисправностям 0, 1, пытаемся покрыть ее LUT, так что покрываемая линия оказывается внутри покрытой им подсхемы и не является входом LUT. Этот LUT не нуждается в коррекции, а покрытая им линия вычеркивается из списка.

Исчерпав такие возможности, пытаемся покрыть подходящие подсхемы программируемыми блоками (LUTs), так что очередная линия из списка является входом u_i некоторого LUT (рис. 1). Этот LUT корректируется описанным выше способом, если удается найти полюс ω в схеме C, такой что его полностью определенная функция реализует частичную функцию полюса v (входа u_i). Формируется дополнительная (обозначенная пунктиром) линия, связывающая полюс ω с входом u_m.

В случае, если возможно маскирование нескольких линий одним и тем же корректирующим LUT, воспользуемся мультиплексором, как это предложено в работах [3, 4].

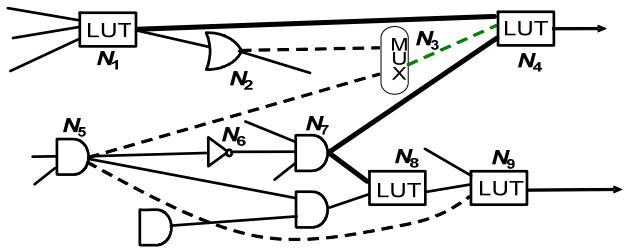


Рис. 5. Маскирование двух линий с помощью мультиплексора
Fig. 5. Two line masking with using MUX

Продолжаем процедуру, до тех пор, пока это возможно. Далее пытаемся маскировать оставшиеся линии, следуя конструкции, представленной на рис. 3, формируя дополнительные линии. В результате получим схему из вентилей, LUTs и MUXs, в которой действия вредоносных подсхем могут быть замаскированы. Имеется возможность либо замаскировать действие ТС в случае ее обнаружения, перепрограммируя соответствующий LUT, либо

получить схему, в которой эффективное введение TCs становится невозможным за счет перепрограммирования соответствующих LUTs.

Для многовыходной схемы *C* предлагается вычислять частичные функции для линии каждой из подсхем, которым она принадлежит. Для каждой из линий получаем систему частичных функций. В результате находим множество линий, неисправности которых трудно обнаружимы. Во множество включаем линию, для которой неисправность трудно обнаружима на каждом из выходов схемы, связанном с этой линией. Аналогичным образом вычисляем частичные функции для внутренних полюсов схем, если они являются точками ветвления. Для остальных полюсов их частичные функции совпадают с частичными функциями исходящих из них линий.

В качестве полюса ω для очередной линии *l* выбираем полюс, такой что его полностью определенная функция реализует каждую из частичных функций системы, сопоставляемых полюсу *v*.

Предварительные эксперименты на контрольных примерах для многовыходных схем показали, что существуют пары полюсов, такие, что частичные функции одного полюса реализуется полностью определенной функцией другого полюса.

6. Заключение

Предложен метод синтеза частично программируемых комбинационных схем (комбинационных составляющих последовательностных схем) из вентилей, программируемых блоков памяти (LUTs) и программируемых мультиплексоров (MUXs), ориентированный на маскирование вредоносных подсхем (TCs) в условиях наличия свободного входа у программируемых блоков памяти. Наряду с маскированием TC за счет перепрограммирования соответствующего LUT в случае обнаружения вредоносной подсхемы предлагается перепрограммирование нескольких LUT таким образом, что включение вредоносных подсхем оказывается нецелесообразным: включение их в линии схемы может с большой вероятностью обнаружено либо в процессе верификации, либо в процессе тестирования. Сформулировано требование к замещающей функции, поступающей на свободный вход программируемого блока, основанное на анализе частичных функций внутренних полюсов комбинационной схемы. Построение частичных функций выполняется с использованием операций над ROBDD-графами, строящимися для фрагментов комбинационной схемы.

Список литературы

[1]. Karri R., Rajendran J., Rosenfeld K., Tehranipoor M. Trustworthy Hardware: Identifying and Classifying Hardware Trojans. Computer, vol. 43, no. 10, 2010, pp. 39-46. DOI: 10.1109/MC.2010.299.

[2]. Yoshimura M., Bouyashiki T., Hosokawa T. A sequence Generation Method to detect Hardware Trojan Circuits. The 16-th IEEE Workshop on RTL and High Level Testing, Proceeding, 2015, pp. 84-89.

[3]. Yamashita S., Yoshida H., Fujita M. Increasing yield using partially-programmable circuits. Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI), Proceeding, 2010, pp. 237-242.

[4]. Jo S., Matsumoto T., Fujita M. SAT-based automatic rectification and debugging of combinational circuits with LUT insertions. IEEE Asian Test Symposium, Proceeding, 2012, pp. 19-24. DOI: 10.1109/ATS.2012.55.

[5]. Матросова А.Ю., Останин С.А., Бухаров А.В., Кириенко И.Е. Поиск всех тестовых наборов для неисправности логической схемы и представление их ROBDD-графом. Вестн. Том. гос. ун-та. УВТиИ. № 2(27), 2014, стр. 82-89.

[6]. Matrosova A.Yu., Ostanin S.A., Kirienko I.E. Generating all test patterns for stuck-at faults at a gate pole and their connection with the incompletely specified Boolean function of the corresponding subcircuit. The 14th Biennial Baltic Electronics Conference, Proceeding, 2014, pp. 85-88. DOI: 10.1109/BEC.2014.7320562.

Partially Programmable Circuit Design Oriented to masking Trojan Circuits

A.Yu. Matrosova <mau11@yandex.ru>
S.A. Ostanin <sergeiostanin@yandex.ru>
E.A. Nikolaeva <nikolaeve-ea@yandex.ru>
National Research Tomsk State University
36, Lenin Avenue, Tomsk, 634050, Russia

Abstract. The enhanced utilization of outsourcing services for a part of VLSIs (Intellectual Property cores, reprogramming components based on FPGA and so on) to cut VLSI cost increases risk of inserting Trojan Circuits (TCs) that may destroy VLSI or provide leakage of confidential information. TCs as a rule act in rare operation situations, therefore they are not detectable neither during VLSI verification nor VLSI testing. The approach to partially programmable circuit design from gates, programmable LUTs and MUXs oriented to masking TCs is suggested. The approach allows getting a circuit that masks TC when it has been found or deriving a circuit that is tolerant to TCs actions. The method of reprogramming LUTs for masking TCs is developed. The condition of replacing a function corresponding to free LUT input is formulated. It is based on using incompletely specified Boolean functions of internal nodes of the circuit. The functions are obtained with using operations on ROBDDs corresponding to the circuit fragments. The operations have a polynomial complexity.

Key words: Partially Programmable Circuits, Trojan Circuits (TCs), Incompletely Specified Boolean Functions, ROBDDs.

DOI: 10.15514/ISPRAS-2017-29(5)-4

For citation: Matrosova A.Yu., Ostanin S.A., Nikolaeva E.A. Partially Programmable Circuit Design Oriented to masking Trojan Circuits. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 5, 2017, pp. 61-74 (in Russian). DOI: 10.15514/ISPRAS-2017-29(5)-4

References

[1]. Karri R., Rajendran J., Rosenfeld K., Tehranipoor M. Trustworthy Hardware: Identifying and Classifying Hardware Trojans. Computer, vol. 43, no. 10, 2010, pp. 39-46. DOI: 10.1109/MC.2010.299.

[2]. Yoshimura M., Bouyashiki T., Hosokawa T. A sequence Generation Method to detect Hardware Trojan Circuits. The 16-th IEEE Workshop on RTL and High Level Testing, Proceeding, 2015, pp. 84-89.

[3]. Yamashita S., Yoshida H., Fujita M. Increasing yield using partially-programmable circuits. Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI), Proceeding, 2010, pp. 237-242.

[4]. Jo S., Matsumoto T., Fujita M. SAT-based automatic rectification and debugging of combinational circuits with LUT insertions. IEEE Asian Test Symposium, Proceeding, 2012, pp. 19-24. DOI: 10.1109/ATS.2012.55.

[5]. Matrosova A.Yu., Ostanin S.A., Buharov A.V., Kirienko I.E. Generating all test patterns for a given stuck-at fault of a logical circuit and its ROBDD implementation. Tomsk State University Journal of Control and Computer Science [Vestn. Tom. gos. un-ta. UVTiI], № 2(27), 2014, pp. 82-89 (in Russian).

[6]. Matrosova A.Yu., Ostanin S.A., Kirienko I.E. Generating all test patterns for stuck-at faults at a gate pole and their connection with the incompletely specified Boolean function of the corresponding subcircuit. The 14th Biennial Baltic Electronics Conference, Proceeding, 2014, pp. 85-88. DOI: 10.1109/BEC.2014.7320562.