Модели неисправностей при верификации проектов и контроле цифровых систем

Золоторевич, Л. А.
zolotorevichla@bsuir.by
Белорусский государственный университет информатики и радиоэлектроники,
Минск, Беларусь

Анализируются вопросы формирования общего подхода к рассмотрению задач контроля и верификации при проектировании современных интегральных схем, основанного на совмещении представлений и анализе моделей неисправностей структурных реализаций цифровых устройств, ошибок, возникающих в процессе проектирования, а также преднамеренных искажений на этапах проектирования и изготовления.

Ключевые слова: верификация проектов, построение тестов, контроль СнК, неисправности, ошибки проектирования, блокирование преднамеренных искажений

Введение

Развитие всех отраслей в 21 веке определяющим образом зависит от развития микро- и наноэлектроники. Акцент на цифровую экономику, преимущественное развитие цифровых технологий требует постоянного совершенствования теории и практики проектирования цифровых систем.

Особенность развития микро- и наноэлектроники заключается в том, что оно существенно зависит от методов и применения средств автоматизированного проектирования. Разработка САПР микроэлектроники началась вместе с появлением первых интегральных схем в 1958 году.

Начиная с 1964 года, когда была проведена первая научная конференция по автоматизации проектирования, ежегодно проводится большое число международных симпозиумов, семинаров по разным аспектам теории и практики автоматизированного проектирования. Тематика конференций непосредственно связана с проблемами контроля, верификации, построения тестов. Сложность решения задач контроля постоянно растет из-за возрастания сложности проектируемых объектов, отсутствия общего подхода к рассмотрению ошибок, вносимых в проект при проектировании, их корреляции с реальными неисправностями структурных реализаций объектов. Но все эти трудно решаемые проблемы, в первую очередь, в области верификации, являются естественными, возникающими непреднамеренно и должны решаться в режиме благоприятствующего проектирования. Но в настоящее время возникла потребность в дополнительном контроле проектов на предмет несанкционированного внедрения с целью искажения проекта с разными основополагающими целями. Подобные действия являются преднамеренными и тщательно скрываемыми, что препятствует прямому применению существующих методов тестирования СБИС.

В связи с этим очевидна необходимость развития таксономии нарушений и отклонений, с моделями которых приходится работать при проектировании и организации контроля на всех этапах жизненного цикла цифровой системы. Как развитие теории контролепригодного проектирования (Design-for-Testability - DfT) в работе [1] предлагается подход к проектированию Design for-Trust - DfTr, который дополнительно включает средства для контроля и предотвращения аппаратных атак при проектировании и изготовлении СБИС.

Впечатляющие достижения в области производства интегральных схем, развитие систем на кристалле (СнК) объемом порядка 10 миллиардов транзисторов, реально работающие цифровые СнК и СнК смешанного типа на пластине размером порядка 450 мм. являются следствием больших успехов в смешанной системной интеграции. Одновременно с несомненными высокими достижениями в области производства СБИС имеет место существенное отставание методов и средств проектирования.

Наиболее узким местом в решении этой задачи является анализ функциональной корректности проектов на каждом из этапов процесса иерархического проектирования. Следует заметить, что применение отработанных в плане проектной корректности многократно используемых блоков повторного применения (IPs) при проектировании современных СнК не решает и даже существенно не упрощает задачу верификации проекта в целом. Объединение отлаженных отдельных функциональных блоков не дает никакой гарантии корректности полученного функционала вследствие возникающих несогласованностей, которые должны быть найдены и устранены на этапе верификации RTL- проекта. Включая в проект определенный IP необходимо иметь уверенность в полноте поставляемого теста контроля. Но более сложной задачей является согласование условий корректного совместного взаимодействия блоков внутри системы в целом.

Имеющиеся практические результаты в областях синтеза, верификации проектов, построения тестов и организации контроля, во-первых, не достигли требуемого уровня развития, а во-вторых, продолжают оставаться корпоративными достижениями, ориентированными на применение специалистами высокой квалификации. В связи с этим, разработка методов и средств функциональной верификации, разработка тестов и систем контроля остаются наиболее наукоемкими задачами, непосредственно определяющими сроки и стоимость проектов, требующими дальнейшего развития.

1. Применение методов диагностики для защиты цифровых устройств

В связи с высокими темпами роста объемов производства цифровых устройств в настоящее время

особую остроту приобретает проблема нарушения авторских прав [1, 2]. Ущерб от пиратства и других угроз в области производства аппаратного обеспечения составляет около 4 млрд. долларов в год, что примерно в 10 раз превышает ущерб от пиратства в области программного обеспечения [2]. Кроме пиратства появляются новые виды угроз [3]:

- внедрение в проект дополнительных вредоносных несанкционированных операций, изменяющих функциональное наполнение системы;
- внедрение механизмов деградации схемных решений с целью нарушения системы синхронизации, приводящих к нарушению временной согласованности путей распространения сигналов, и, в конечном итоге, к сбою системы;
- включение средств для получения конфиденциальной информации (к примеру, получение криптографических ключей) через порты контроля и к подрыву безопасности и др.

Очевидно, что после изготовления интегральной схемы проверить ее на наличие внесенных искажений, дополненной функциональности можно путем перепроектирования «по прототипу», восстанавливая поэтапно логику устройства и сравнивая с правильным образцом. При этом восстанавливается проект, реализованный в схеме, и сравнивается с моделью исходного проекта. Этот метод обеспечивает высокую вероятность обнаружения искажений, но время и стоимость, необходимые для выполнения перепроектирования, непомерно высоки.

Поэтому основные практические методы контроля развиваются в направлении создания общего подхода к функциональному и тестовому контролю для обнаружения как разного вида ошибок и неисправностей, возникающих в рабочем режиме проектирования, так и для обнаружения внесенных механизмов искажений.

В работе [4] приведены различные модели процесса злонамеренного внедрения, описывающие условия, при которых злоумышленное искажение может внедриться в цифровую систему. Рассматриваются в числе возможных источников искажений поставщики IP's, разработчики СнК, кремниевые фабрики – изготовители СнК.

В модели A вредоносным источником является поставщик IP's, который продает свои изделия разработчикам СнК. Эта модель вполне реалистична, так как разработчики СнК с целью сокращения стоимости и сроков проектирования широко используют привлечение существующих проектов. В рамках данной модели искажение проекта может происходить на RTL-уровне, на функционально - логическом или топологическом.

В модели B угроза исходит от кремниевой фабрики на этапе производства интегральной схемы. Поскольку при изготовлении имеется доступ к топологическому проекту, то возможно восстановление и перепроектирование проекта, добавление элементов аппаратных искажений. Жизненность такой модели очевидна в связи с тем, что со стороны проектировщиков практически отсутствует возможность контроля деятельности в случае, к примеру, оффшорного производства.

В модели C искажения проекта возможны на этапе проектирования вследствие преднамеренных злоумышленных действий конкретного информированного сотрудника, что возможно при использовании ненадежного программного обеспечения САПР.

В моделях D, E, F, G рассматриваются аппаратные искажения, которые возможны в случае ненадежности любых двух или всех трех участников процесса.

Искажения в проекте могут происходить на разных этапах проектирования: на RTL-уровне, на уровне описания схем (уровень netlist), в топологическом проекте. В этих условиях существует потребность в разработке методов и средств обнаружения искажений на разных уровнях абстракции. Поэтому особое значение имеет разработка методов защиты от подобных угроз и борьбы с пиратством. Одной из известных методик защиты исходных кодов программ от обратного проектирования является обфускация, основной задачей которой является затруднение понимания функционирования программы. К сожалению, применение методов обфускации теряют свою актуальность в случае языка VHDL, так как результаты их применения не приводят к изменению конечного результата синтеза, так как структурные реализации устройств до и после обфускации выглядят одинаково [2].

Для выявления преднамеренных искажений, проникающих в реализации интегральных схем, могут быть применены известные методы и средства технического диагностирования, которые разработаны для обеспечения верификации проектов и контроля объектов проектирования на всех этапах жизненного цикла [5-8].

Ниже рассматривается метод логического шифрования структурных реализаций цифровых устройств, который заключается в изменении логической структуры путем добавления некоторых вентильных элементов и внедрении во входную последовательность встроенных ключей, уникальных для данной схемы.

2. Логическое шифрование цифровых структур

На рис. 1 приведена схема цифрового устройства, реализующего некоторую систему булевых функций. При шифровании в схему добавлены вентили типа XOR, которыми управляют внешние сигналы (ключи) K1, K2 и K3 (рис. 2). Цифровое устройство реализует заданную функцию при установке правильных состояний ключей. Если состояние ключа K1 (рис. 2) на входе вентиля XOR равно 1, то на его выходе будет логическое состояние, инверсное состоянию сигнала на выходе элемента A1, то есть

состояние 0 при 1 в рабочем режиме. Не правильное состояние ключа может быть выявлено по аналогии с выявлением неисправности на линии типа const 0.

В таблице разностных неисправных функций (табл.1) в первом столбце приведены все входные тестовые векторы в классе неисправностей константного типа. В первой строке приведены все неисправности константного типа входных и внутренних переменных схемы. Верхний индекс указывает тип константной неисправности соответствующей линии.

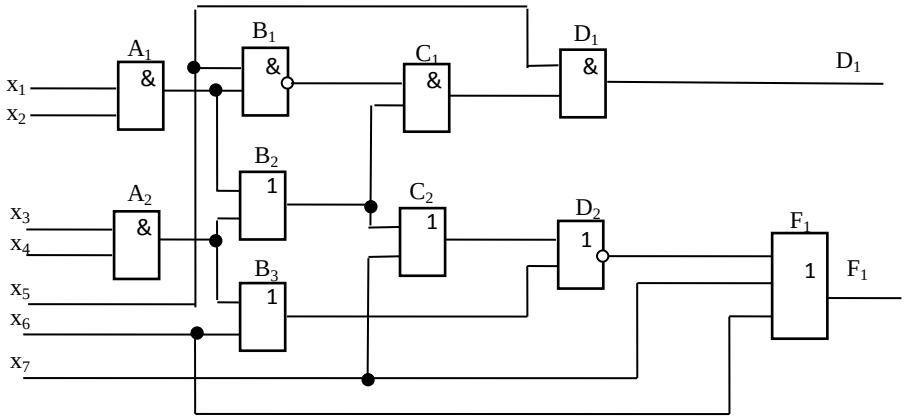


Рис. 1. Логическая схема, реализующая систему булевых функций

$$D_1 = \overline{x_1}x_3x_4x_5 \vee \overline{x_2}x_3x_4x_5; F_1 = \overline{x_1}\,\overline{x_3} \vee \overline{x_2}\,\overline{x_3} \vee \overline{x_1}\,\overline{x_4} \vee \overline{x_2}\,\overline{x_4} \vee x_6 \vee x_7.$$

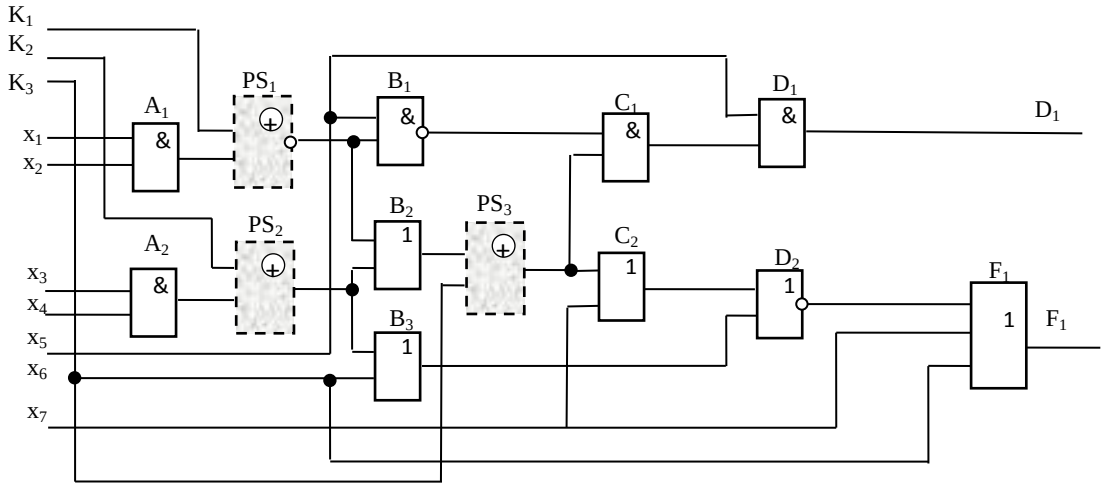
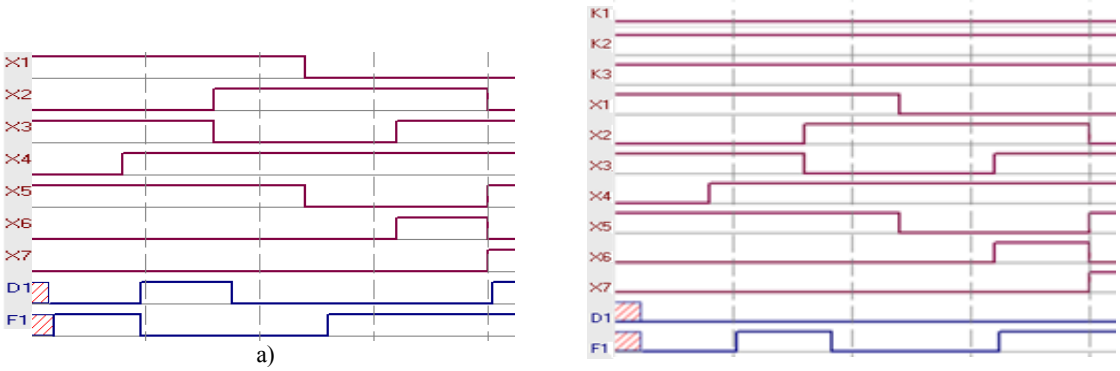


Рис. 2. Схема с вентилями PS1, PS2 и PS3 для логического шифрования



а) б)

Рис. 3. Результаты моделирования; а) для схемы на рис.1; б) для схемы на рис.2.

На рис. 3 приведены временные диаграммы функционирования исходной схемы (рис.1) на тестовом наборе, разработанном для контроля неисправностей константного типа (рис. 3, а), и схемы с введенными вентилями для шифрования (рис. 3, б) с неправильными значениями ключей.

С целью обеспечения недоступности извлечения правильных ключевых кодов необходимо

обеспечить 50% кодовое расстояние Хэмминга между выходными состояниями схемы в условиях применения правильных и ошибочных ключевых кодов [9]. В работе [10] рассмотрен метод случайного выбора мест внедрения дополнительных вентилей. При таком подходе в результате эффекта маскирования неисправностей эффект искажения функции от применения неправильных ключей может быть потерян.

Таблица 1. Таблица разностных неисправных функций

Неисправности → / Тест-векторы ↓	X_1^0	X_1^1	X_2^0	X_2^1	X_3^0	X_3^1	X_4^0	X_4^1	X_5^0	X_5^1	X_6^0	X_6^1	X_7^0	X_7^1	A_1^0	A_1^1	A_2^0
1010100				1^2				1^1								1^2	
1011100				1^1	1^1		1^1		1^1			1^2		1^2		1^1	1^1
1101100	1^2		1^2									1^2		1^2	1^2		
0101111		1^2				1^2										1^2	
0111010										1^1	1^2						
0011101					1^1		1^1		1^1				1^2			1^1	1^1
Неисправности → / Тест-векторы ↓	A_2^1	B_1^0	B_1^1	B_2^0	B_2^1	B_3^0	B_3^1	C_1^0	C_1^1	C_2^0	C_2^1	D_1^0	D_1^1	D_2^0	D_2^1	F_1^0	F_1^1
1010100	1^1				1^1		1^2		1^1		1^2		1^1	1^2		1^2	
1011100		1^1		1^1				1^1				1^1			1^2		1^2
1101100			1^1	1^2					1^1	1^2			1^1		1^2		1^2
0101111	1^2				1^2		1^2				1^2		1^1	1^2		1^2	
0111010													1^1			1^2	
0011101		1^1		1^1				1^1				1^1				1^2	

Представим в общем виде алгоритм шифрования структурных реализаций цифровых устройств комбинационного типа:

1) Разработать тест контроля цифрового устройства в классе неисправностей константного типа;
2) Определить неисправности, обнаруживаемые каждым из наборов теста;
3) Построить таблицу разностных неисправных функций;
4) По таблице разностных неисправных функций определить множество неисправностей, которые могут изменить логические состояния порядка 50% выходных линий;
5) В соответствии с множеством неисправностей, полученным в п. 4, произвести зашифрованную структуру схемы, используя по необходимости вентили XOR или XNOR;
6) Путем моделирования неисправностей зашифрованной схемы выявить возможные явления маскирования неисправностей.
7) При получении неудовлетворительного результата перейти к п.4.

Список литературы

1. *Rajendran J., Sam M., Sinanoglu O., Karri R.* Security analysis of integrated circuit camouflaging // ACM SIGSAC conference on Computer & communications security. Germany, Berlin. 04 - 08 November, 2013. P. 709-720.
2. *Сергейчик В. В., А. А. Иванюк.* Методы лексической обфускации VHDL-описаний // Information Technologies and Systems 2013 (ITS 2013): Proceeding of The International Conference, BSUIR, Minsk, 24th October 2013. PP. 198-199.
3. *Shakya B., He. T., Salmani H., Forte D., Bhunia S., Tehranipoor M.* Benchmarking of hardware Trojans and maliciously affected circuits // Hardw. Syst. Secur. (HaSS) 1(1). 2017. PP. 85–102.
4. *Xiao K., Forte D., Jin Y., Karri R., Bhunia S., Tehranipoor M.* Hardware Trojans: Lessons learned after one decade of research/ // ACM transactions on design automation of electronic system. Vol. 22, No.1. Article 6. May 2016.
5. Zolotorevich L.A. Modeling of faults of VLSI at the structural level: problems and solutions // The International Conference Computer- Aided Design of Diskrete Devices (CAD-DD'04). – Vol. 1. – Minsk. – 2004. – PP. 32-40.
6. *Zolotorevich L.A.* Project verification and construction of superchip tests at the RTL level // Automation and Remote Control. USA, NY, Plenum Press 2013. Vol. 74, Issue 1. P. 113-122.
7. *Zolotorevich L.A., Il'inkova A. V.* Development of tests for VLSI circuit testability at the upper design levels Automation and Remote Control. USA, NY, Plenum Press. Vol. 71 Issue 9. September 2010. P. 1888-1898.
8. *Zolotorevich, L.A., Yukhnevich D.I.* Switch-level VLSI quasistatic simulation methods: Comparative accuracy of models // Automation and Remote Control. USA, NY, Plenum Press. 09/1998.-9(9).
9. *Baumgarten A., Tyagi A., Zambreno J.* Preventing IC Piracy Using Reconfigurable Logic Barriers / //IEEE Design and Test of Computers. Vol. 27. No. 1. 2010. PP. 66–75.
10. *Roy J., Koushanfar F., Markov I.* EPIC: Ending Piracy of Integrated Circuits Proceedings of the IEEE //ACM Design, Automation and Test in Europe. 2008. PP. 1069–1074.