

УДК 519.876.5

В.А. КУЛАНОВ, С.А. КУЛАНОВ, А.С. СКРЫННИК

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина***ИНСТРУМЕНТАЛЬНОЕ СРЕДСТВО ЗАСЕВА И МОДЕЛИРОВАНИЯ
КОНСТАНТНЫХ НЕИСПРАВНОСТЕЙ ВСТРОЕННЫХ СИСТЕМ НА ПЛИС**

Проведен анализ существующих техник и инструментально-технических средств засева дефектов встроенных систем на программируемых логических интегральных схемах (ПЛИС). Предложено инструментальное средство засева и моделирования одиночных и множественных константных неисправностей проектов на ПЛИС. Предложенное средство обладает рядом преимуществ по сравнению с существующими программными комплексами: не требует знаний языков описания аппаратуры со стороны пользователя, универсальный формат представления входных данных (EDIF 2.0.0), открытый программный код, возможность формирования файла «прошивки» для ПЛИС с инжектированными дефектами.

Ключевые слова: дефект, засев дефектов, ПЛИС, константная неисправность, stuck-at-fault.

Введение

Благодаря своим характеристикам и функциональным возможностям элементная база программируемых логических интегральных схем (ПЛИС) получила широкое применение в области разработки систем критического применения, т.е. таких систем к которым выдвигаются высокие требования к отказоустойчивости и функциональной безопасности.

Существует множество методов оценки качества, а также верификации программного обеспечения и аппаратных комплексов систем критического применения, в том числе встроенных цифровых систем на микросхемах программируемой логики. Одним из способов верификации, оценки надежности, гарантоспособности и качества проектных решений критических систем является метод засева дефектов [1]. Он закреплен в стандарте международной электротехнической комиссии (IEC 61508), как метод верификации систем критического применения в части анализа характера и последствий отказов (Failure Modes and Effects Analysis) [2].

Метод засева дефектов позволяет проанализировать реакцию системы на ошибку, вызванную внесенным дефектом, не дожидаясь ее естественного проявления при работе системы в штатном режиме. Таким образом, это дает возможность определить, насколько реакция системы, при наличии в ней определенного набора дефектов, соответствует реакции, определенной в самой спецификации проектируемой системы. Другим преимуществом данного метода является возможность прогнозирования и устранения проектных дефектов, способных привести к отказу системы.

На сегодняшний день предложено и разработано множество техник и программно-технических средств засева дефектов в рамках оценки качества и надежности программного обеспечения и аппаратных комплексов критического применения, в том числе проектов на ПЛИС [1]. Все они базируются на определенном подходе при выполнении процедуры засева.

Первый подход основан на имитационном моделировании, когда дефекты вносятся в некоторую модель целевой системы, проводится моделирование и анализируются полученные результаты. При этом разработчик получает преимущество в гибкости настройки процедуры засева и возможности внесения изменений в объект моделирования, но проигрывает в скорости получения результатов. Во втором случае дефекты вносятся непосредственно в исполняющую систему, т.е. рабочий прототип. Эта техника больше подходит под готовые проектные решения, где количество возможных доработок и изменений в функциональность устройства (программного продукта) минимальна.

Для проектов на ПЛИС существует множество техник и программно-технических комплексов засева дефектов. У каждого есть свои достоинства и недостатки.

Выбор определенного инструментального средства в большой степени зависит от решаемой задачи и сложности проектируемой системы.

Таким образом, для верификации проекта разработчик должен учитывать характеристики инструментального средства: поддерживаемый тип дефектов и метод их внесения, способ реализации (программный, аппаратный, смешанный и т.д.),

время получения выходных результатов, масштабируемость, наблюдаемость, а также формат входных и выходных данных и т.д.

Цель статьи – провести анализ существующих методов и программно-технических средств засева дефектов встроенных систем на ПЛИС, рассмотреть средство засева и моделирования одиночных и множественных константных неисправностей проектов на ПЛИС.

1. Анализ методов и инструментальных средств засева дефектов ПЛИС-проектов

На сегодняшний день методы засева дефектов для встроенных систем принято разделять на несколько групп [1]: *аппаратные, программные, имитационные и смешанные (гибридные)*.

Аппаратный засев дефектов подразумевает использование специально разработанного тестового оборудования. Традиционно такой тип дефектов вносился на уровне выводов интегральных схем. С появлением новых технологических решений, например, специализированный аппаратный интерфейс JTAG и метода граничного сканирования, проектировщик получил возможность доступа к внутренним ресурсам кристалла.

Преимущества метода:

1. Доступ к областям кристалла, которые труднодоступны другими методами.
2. Хорошо работает для систем, в которых требуется длительное время выполнения тестовых наборов и их мониторинга.
3. Экспериментальная оценка путем засева в аппаратуру во многих случаях является единственным практическим способом точного оценивания тестового покрытия и задержек на элементах.
5. Лучше подходит для моделей неисправностей низкого уровня.
6. Эксперименты выполняются быстрее. Опыты можно выполнять в режиме реального времени.
8. Не требует моделей проектирования (разработки) или проверки (валидации).

К основным недостаткам метода относятся:

1. Аппаратный засев может представлять высокий риск повреждения целевой системы.
2. Низкая переносимость.
3. Ограниченный набор точек засева и ограниченный набор засеваемых дефектов.
4. Требуется специализированная аппаратура для засева дефектов.

Инструментальные средства, использующие аппаратный способ засева дефектов – AFIT, RIFLE, MARS [1].

Программный засев дефектов позволяет оценивать последствия скрытых ошибок программного

обеспечения встроенных систем. Метод предполагает модификацию программного кода путем внесения дефектов соответствующего типа. Не используется для верификации аппаратных решений на ПЛИС.

Множество современных средств засева дефектов встроенных систем на ПЛИС используют **имитационное моделирование**. Это обусловлено в первую очередь спецификой разработки такого класса проектов.

Здесь начальный этап проектирования – это разработка модели-спецификации цифрового устройства, для чего обычно используются различные языки описания аппаратуры, такие как VHDL, Verilog и т.д. Затем, на уровне этапов верификации и тестирования модели, используется имитационное моделирование полученного устройства в рамках современных систем автоматизированного проектирования (САПР).

Внесение дефектов при имитационном моделировании возможно как за счет изменения поведения модели устройства, либо ее частей, так и применение различных техник засева дефектов в рамках имитационного подхода.

Преимущество метода:

1. Полный контроль над моделью неисправности, моделью целевой системы и механизмом засева дефектов.
2. Невысокая стоимость системы компьютерной автоматизации, не требующая специализированной аппаратуры.
3. Возможность моделирования кратковременных и постоянных дефектов.
4. Возможность внесения дефектов на различных уровнях абстракции модели цифрового устройства.

К основным недостаткам метода относятся:

1. Значительное время разработки модели и получения результатов моделирования.
2. Точность результатов зависит от качества выбора модели дефектов.
3. Модели могут включать не все типы дефектов, которые присутствуют в реальной аппаратной реализации системы.
4. Необходимы специальные знания в области языков описания аппаратуры.

Инструментальные средства засева дефектов, которые используют имитационную модель – VFIT, MEFISTO-C, ALIEN, VERIFY [1].

К **смешанному (гибридному)** подходу относятся программно-технические комплексы, которые сочетают в себе возможности поддержки программно-аппаратного способа внесения неисправностей.

Применение данного подхода особенно эффективно, когда существует необходимость в использовании часто изменяющихся техник засева и моделей

дефектов, а также в тех случаях, когда необходимо провести частичное моделирование системы с определенными типами неисправностей. Гибридный подход сочетает в себе точность аппаратного и гибкость программного метода.

Как было рассмотрено ранее, большинство средств засева дефектов ПЛИС-проектов базируется на имитационном моделировании и использует языки описания аппаратуры, как для представления системы, так и для описания моделей дефектов [1].

С одной стороны достигается максимальный уровень совместимости, с другой – накладываются дополнительные требования к знанию языка описания аппаратуры, специальных языковых конструкций и условий их применения для активации определенного типа дефектов.

Существует ряд других программных средств [1], которые для моделирования дефекта не требуют знаний языков описания аппаратуры, однако используют специальный формат представления входных данных (проекта). Это создает определенного рода трудности, связанные с преобразованием готовых проектных файлов, особенно когда их сложность составляет сотни строк кода, в рабочий формат средств моделирования.

Кроме того, отсутствие «открытого программного кода» для большинства существующих средств засева дефектов не позволяет расширять их функциональность, что приводит к медленному развитию программных продуктов в данном направлении.

2. JFITool - средство засева и моделирования константных неисправностей ПЛИС-проектов

В основе программного средства засева и моделирования одиночных и множественных константных неисправностей JFITool положена технология Java.

Основными составляющими компонентами средства JFITool является библиотека JHDL [3] и Byuediftools [4].

JHDL (Java Hardware Description Language) – низкоуровневый язык описания аппаратуры, основанный на принципах описания цифровых устройств в рамках объектно-ориентированного подхода с использованием языка программирования Java. Представляет собой набор библиотек и утилит для разработки, визуализации и тестирования цифровых проектов на ПЛИС [3].

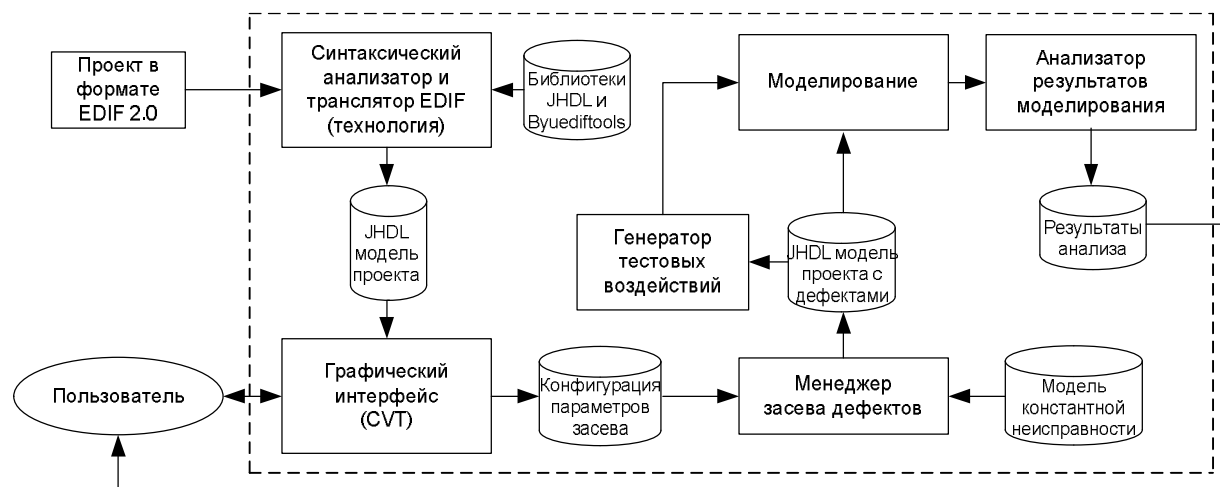


Рис. 1. Архитектура JFITool

Библиотека Byuediftools в основном состоит из классов и методов, реализующих функцию синтаксического анализа (разбора) файлов проектов на ПЛИС, представленных в формате EDIF 2.0.0 [5]. В Byuediftools также существует набор утилит (классов), которые работают с данными, полученными в результате синтаксического разбора файла проекта. Среди прочих, ряд классов, позволяющих использовать возможности библиотеки JHDL.

Преимуществом рассмотренных библиотек является их общедоступность и наличие открытого программного кода, который может модернизи-

роваться и применяться согласно открытому лицензионному соглашению.

Утилита JFITool разработано в рамках инструментального средства расчета метрик диверсности многоверсионных проектов на ПЛИС.

Архитектура JFITool включает следующие компоненты (рис. 1):

- графический интерфейс пользователя;
- синтаксический анализатор;
- менеджер засева дефектов;
- генератор тестовых воздействий;
- анализатор выходных результатов;

Графический интерфейс пользователя. Позволяет визуализировать процедуру засева и моделирования константных неисправностей, отобразить модель целевого устройства в различных состояниях, провести настройку проводимого эксперимента.

Синтаксический анализатор. Выполняет синтаксический анализ и транслирование проекта в формате EDIF 2.0.0 в JHDL представление, используя библиотеки JHDL и Vuuediftools.

Менеджер засева дефектов. Выполняет засев единичных и множественных константных неисправностей в модель проекта. Конфигурируется через графический интерфейс.

Генератор тестовых воздействий. Формирует набор тестовых воздействий (последовательностей) для целевой модели.

Анализатор выходных результатов. Проводит анализ результатов моделирования системы и генерирует отчет (выходной результат) для конечного пользователя.

Программное средство JFITool позволяет:

- выполнять синтаксический анализ и трансляцию проектов в формате EDIF 2.0.0 в формат JHDL с возможностью визуализации целевой системы;
- выполнять засев и моделирование множественных и единичных константных неисправностей;
- проводить анализ результатов моделирования и тестирования модели проекта;
- выполнять генерацию наборов тестовых воздействий, в зависимости от требуемого алгоритма;
- формировать выходной файл списка соединений в формате EDIF 2.0.0, в том числе с инжектированными константными неисправностями.

JFITool имеет открытый программный код, доступно для модернизации и функционального расширения сторонними разработчиками.

3. Результаты моделирования в среде JFITool

Одно из основных назначений техники засева дефектов проектных решений на ПЛИС является оценка алгоритмов тестирования целевой системы, при этом для сравнительной характеристики того или иного метода используется метрика полноты покрытия неисправности (Fault Coverage, FC) [6, 7].

Полнотой покрытия неисправностей, достигаемой при тестировании схемы, называется отношение числа неисправностей, обнаружимых тестом, к общему числу возможных неисправностей схемы:

$$FC = \frac{n}{M} \times 100\%, \quad (1)$$

где n – количество обнаруженных тестом дефектов;

M – общее число всех неисправностей в схеме.

В настоящее время считается приемлемым использовать алгоритмы, позволяющие детектировать (покрывать) 95% неисправностей соответствующей модели дефектов [8].

Рассмотрим пример оценки полноты покрытия неисправностей для 4-х битового компаратора. В качестве искомого параметра, определим среднее значение количества случайных тестовых наборов (E), необходимых для покрытия всех возможных единичных константных неисправностей.

При реализации эксперимента введем ряд ограничений:

- проведем засев и моделирование единичных константных неисправностей (single stuck-at-fault);
- рассмотрим класс устройств, относящихся к комбинационным автоматам;
- генерация входных тестовых последовательностей осуществляется случайным образом.

Реализация комбинационного устройства проводилась на языке описания аппаратуры VHDL (ANSI/IEEE Std 1076-1993) в Xilinx WebPack ISE 11.1, с последующей трансляцией файла проекта в формат EDIF 2.0.0.

Для снижения времени моделирования, необходимого для получения достоверных результатов, расчет производился в распределенном вычислительном окружении ГРИД [9].

Результаты моделирования в среде JFITool приведены на рисунке 2, где представлена зависимость E (по оси Y) от количества проведенных экспериментов (по оси X). Очевидно, что увеличение количества экспериментов, позволяет получить более высокую точность показателя E .

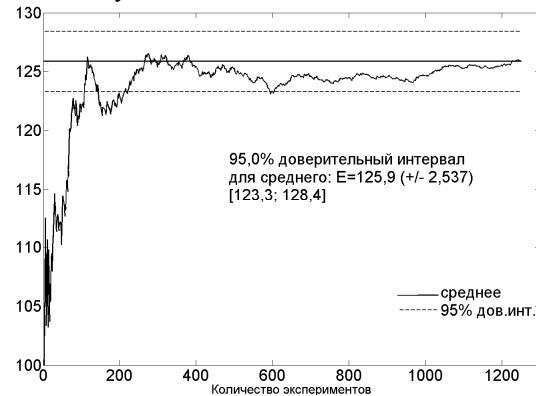
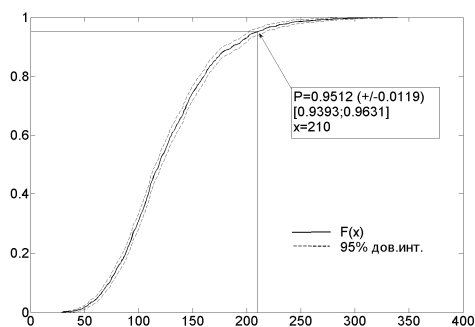


Рис. 2. Среднее количество случайных тестовых наборов, покрывающих 100% неисправностей

Рассмотрим функцию распределения $F(x)$ количества тестов необходимых для покрытия 100% дефектов в схеме 4-х битового компаратора (рис. 3). Таким образом, необходимо провести 210 экспериментов, чтобы с вероятностью 0.95 утверждать о детектировании всех возможных единичных константных неисправностей в данной схеме.

Рис. 3. Функция распределения $F(x)$

Выводы

Предложенное средство засева и моделирования константных неисправностей JFITool обладает рядом преимуществ по сравнению с существующими программными комплексами: не требует знаний языков описания аппаратуры со стороны пользователя; универсальный формат представления входных данных (EDIF 2.0.0); открытый программный код; возможность формирования файла «прошивки» для ПЛИС с инжектированными дефектами;

В дальнейшем планируется расширить функциональность JFITool в части: реализации механизма распределенных вычислений за счет возможностей языка программирования Java; поддержки входных и формирования выходных файлов в формате ISCAS'89; поддержки методов генерации тестовых последовательностей на основе существующих

алгоритмов (D-алгоритм, PODEM, FAN и т.д.) [6 – 8]; поддержки работы через Интернет.

Литература

1. Benso A. *Fault injection techniques and tools for embedded systems reliability evaluation* // A. Benso, P. Prinetto. – Politecnico di Torino, Italy. – 2003.
2. IEC 61508 Standart for Functional Safety of Electrical / Electronic // Programmable Electronic Safety-Related Systems.
3. Bellows P. *JHDL – An HDL for reconfigurable Systems* // P. Bellows, B. Hutchings– Brigham Young University, Provo, USA. – 1997.
4. Reliability.ee.byu.edu [Электронный ресурс] – Режим доступа: <http://reliability.ee.byu.edu>.
5. ANSI/EIA-548-1988, *Electronic Design Interchange Format, Version 2.0.0 Recommended Standart EIA-548, Electronics Industries Associations*. [Электронный ресурс] – Режим доступа: <http://reliability.ee.byu.edu/edif/EIA-548>
6. *A Designer's Guide to Built-In Self-Test* // Kluwer Academic Publishers, 2002. – 319 p.
7. Janusz Rajski, Jerzy Tyszer, *Arithmetic Built-in Self-test for Embedded Systems*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1998
8. Miczo A. *Digital Logic Testing and Simulation, Second Edition* / A. Miczo. – John Wiley, 2003.
9. H. Stockinger. *Defining the Grid: A Snapshot on the Current View. Draft 1.0* / Stockinger H. – 26 June 2006.

Поступила в редакцию 12.02.2010

Рецензент: д-р техн. наук, проф., зав. кафедры компьютерных систем и сетей В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина.

ІНСТРУМЕНТАЛЬНИЙ ЗАСІБ ЗАСІВУ ТА МОДЕЛЮВАННЯ КОНСТАНТНИХ НЕСПРАВНОСТЕЙ ВБУДОВАНИХ СИСТЕМ НА ПЛІС

В.О. Куланов, С.О. Куланов, О.С. Скринник

Проведено аналіз існуючих технік та інструментально-технічних засобів засіву дефектів вбудованих систем на програмованих логічних інтегральних схемах (ПЛИС). Запропоновано інструментальний засіб засіву та моделювання поодиноких та множинних константних несправностей проектів на ПЛИС.

Ключові слова: дефект, засів дефекту, ПЛИС, константна несправність, stuck-at-fault.

STUCK-AT-FAULT INJECTION AND MODELING TOOL FOR EMBEDDED FPGA-BASED PROJECTS

V.O. Kulanov, S.O. Kulanov, O.S. Skrynnik

Modern fault injection tools for embedded PLD-based projects are analyzed. A new single and multiple stuck-at-fault injection and modeling tool for embedded PLD-based projects are proposed.

Keywords: defect, fault injection, PLD, stuck-at-fault.

Куланов Виталий Александрович – ассист. кафедры компьютерных систем и сетей, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина, e-mail: V.Kulanov@csac.khai.edu.

Куланов Сергей Александрович – ассист. кафедры компьютерных систем и сетей, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина, e-mail: Sergey@kulanov.org.ua.

Скринник Александр Сергеевич – студент кафедры компьютерных систем и сетей, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина, e-mail: isken_khai@mail.ru.