

МЭС-2006

II Всероссийская научно-техническая конференция

**ПРОБЛЕМЫ РАЗРАБОТКИ
ПЕРСПЕКТИВНЫХ
МИКРОЭЛЕКТРОННЫХ
СИСТЕМ – 2006 (МЭС-2006)**

СБОРНИК ТРУДОВ



Институт проблем проектирования в микроэлектронике
Российской академии наук

09-13 октября 2006

II Всероссийская научно-техническая конференция
**ПРОБЛЕМЫ РАЗРАБОТКИ ПЕРСПЕКТИВНЫХ
МИКРОЭЛЕКТРОННЫХ СИСТЕМ – 2006**

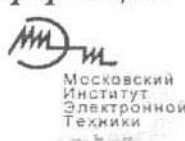
СБОРНИК ТРУДОВ

Учредители конференции



Организатор и проводящая организация –
**Институт проблем проектирования
в микроэлектронике Российской академии наук**

Соорганизаторы конференции



Спонсоры конференции



Информационная поддержка через научные журналы:
"Информационные технологии", "Нано- и микросистемная техника"
"Информационно-управляющие системы"

Исследование методов и средств верификации проектов и генерации тестов МЭС

Л.А. Золоторевич

Белорусский государственный университет, zolotorevichla@bsu.by

Аннотация — Приводится характеристика и результаты эксплуатации собственной системы VLSI_SIM логического проектирования и генерации тестов цифровых СБИС и программы моделирования ModelSim фирмы Mentor Graphics. Дается анализ эффективности реализованных в системе VLSI_SIM методов моделирования и генерации тестов. Рассматриваются полученные теоретические результаты в области моделирования неисправностей СБИС как на структурном уровне, так и на уровне поведения, а также направления развития функциональных возможностей системы.

I. ВВЕДЕНИЕ

Определенные трудности, связанные с применением фирменных зарубежных систем проектирования в микроэлектронике, обусловлены не только их высокой первоначальной стоимостью, но и необходимостью постоянного приобретения новых версий, многотомными англоязычными техническими описаниями, отсутствием результатов глубоких исследований функционального наполнения этих систем и практически необходимых инструкций по технологии их применения. В связи с этим имеется небольшое число специалистов, лишь в определенной мере владеющих заложенными в этих системах технологиями проектирования, что негативно сказывается на уровне отечественной микроэлектроники. Кроме того, на сегодняшний день на практике отсутствует сквозная САПР микроэлектроники, которая обеспечивала бы в полной мере решение требуемых задач с учетом тенденций развития микроэлектроники. Поэтому целесообразно создавать отечественную технологию проектирования на основе глубокого понимания базовых механизмов разработки и интерпретации моделей, генерации тестов и решения др. задач проектирования, с учетом своего опыта и имеющихся практических разработок.

Опыт в области моделирования и тестового диагностирования электронных систем, в том числе, опыт широкого экспериментального исследования в учебном процессе наиболее распространенной программы ModelSim фирмы Mentor Graphics позволил сделать следующие выводы:

- язык VHDL не имеет средств для описания динамики процессов в реальном спектре значений задержек компонентов, так как не позволяет учитывать

задержки компонентов в интервальном виде, что весьма целесообразно при верификации проектов на высоких уровнях проектирования до разработки топологического проекта;

- программа ModelSim не имеет эффективных функционально наполненных средств для задания моделируемых входных воздействий, что сказывается на эффективности верификации проектов; она не позволяет моделировать неисправности и оценивать полноту тестовой последовательности, а формат языка VHDL, как языка описания структуры объекта, ограничивает возможности реализации эффективных методов моделирования неисправностей, а также моделирования объектов на уровне переключательного представления;

- несмотря на большое число публикаций [1]-[5], направленных на решение задач генерации тестов, моделирования неисправностей как на уровне поведенческого представления объекта, так и на переключательном уровне, эффективные средства решения данных задач практически не известны;

- известные фирменные программы генерации тестов, как правило, полностью привязаны к технологии проектирования объекта и не позволяют работать в условиях отсутствия возможности перепроектирования.

В связи с вышеизложенным, на данном этапе для эффективного решения всего спектра задач проектирования СБИС целесообразно комплексирование средств различных систем проектирования.

В разделе 2 приведены некоторые особенности проектирования при переходе на применение языка VHDL.

В разделе 3 описываются функциональные возможности системы проектирования VLSI_SIM, разработанной в БГУ и доведенной до уровня промышленного применения. Сравнительной особенностью данной системы является ее широкое функциональное наполнение, обеспечивающее решение задач верификации проектов и разработки систем тестового диагностирования.

В разделе 4 рассмотрена возможность повышения точности моделирования цифровых устройств с повторно сходящимися ветвлениями и, соответственно, решения задачи сквозного моделирования неисправностей.

В разделе 5 предлагается один из путей конструктивного решения задачи моделирования неисправностей на уровне поведенческого описания объекта.

В настоящее время в литературе рассматриваются разные подходы к решению данной задачи [1],[2]. Однако практически все они предлагают недостаточно обоснованные решения.

Приводятся практические результаты в области разработки методов и средств верификации проектов и предлагаются новые пути повышения эффективности решения задачи моделирования неисправностей, которая лежит в основе решения ряда задач по обеспечению надежности устройства.

II. ВЕРИФИКАЦИЯ ПРОЕКТОВ И ГЕНЕРАЦИЯ ТЕСТОВ НА ОСНОВЕ VHDL

По мере развития интегральной схемотехники острота проблемы верификации проектов СБИС и построения тестов не ослабевает. Ощущается потребность в новых методах и поддерживающих их инструментальных средствах проектирования «толерантных к неисправностям» электронных систем. Широкое внедрение в практику проектирования интегральных схем языка VHDL как основного стандарта языка описания и моделирования, к сожалению, не позволило избавиться от ряда проблем верификации проектов на всех этапах проектирования, разработки тестов контроля, анализа контролепригодности объекта и контролирующих свойств тестов. Во многом некоторые проблемы еще более усугубились при переходе к нисходящему проектированию на основе VHDL.

В рамках нового подхода проект СБИС начиная с системного, функционально-логического и на последующих уровнях является результатом автоматического синтеза и представляет собой автоматически полученное описание устройства на языке VHDL. В этом случае исходными данными для моделирования и верификации проекта, генерации средств тестового диагностирования, анализа контролепригодности проекта является программа, описывающая функционирование схемы на языке VHDL. В этих условиях необходимо переосмысление и решение указанных выше задач проектирования с учетом новых подходов, так как известные методы решения данных задач в условиях применения VHDL зачастую оказались непригодными для применения на практике.

Постоянное повышение степени интеграции, структурной сложности СБИС, переход на проектирование и изготовление СБИС по принципу «система на чипе» требуют решения указанных выше задач на разных уровнях проектирования с целью сокращения сроков проектирования и запуска в серию. Желательно на самом высоком уровне поведенческого представления объекта уметь делать оценку контролепригодности проекта, анализировать тест на полноту контроля,

строить тест контроля. Такой подход позволяет определять и решать ряд потенциальных проблем на раннем этапе проектирования.

III. ФУНКЦИОНАЛЬНОЕ НАПОЛНЕНИЕ VLSI_SIM

В Белгосуниверситете разработана, доведена до уровня промышленного применения и широко используется в учебном процессе на факультете прикладной математики и информатики программная система VLSI_SIM (*Very Large Scale Integration - Simulation*) для решения задач верификации проектов и генерации тестов СБИС и устройств цифровой электроники. Система предназначена для иерархического моделирования СБИС на функционально-логическом и переключательном уровнях для верификации проектов, решения задач по созданию средств тестового диагностирования, контроля цифровых блоков. Система работает с проектами цифровых СБИС, разработанных по МОП и БикМОП технологиям, и устройств цифровой электроники.

VLSI_SIM ориентирована на цифровые устройства общего вида, представленные в иерархическом виде как структурная взаимосвязь функциональных компонентов, заданных их автоматными моделями, и фрагментов на уровне транзисторного представления.

Система разработана на языках Delphi, Visual C++, имеет стандартный интерфейс Windows.

В системе обеспечиваются

- подготовка исходных описаний, их синтаксический и семантический анализ, трансляция во внутреннее представление; поддерживается международный формат обмена данными EDIF;
- создание и ведение библиотек компонентов;
- логико-динамическое моделирование иерархических структур БИС;
- моделирование неисправностей;
- генерация тестов структур общего вида БИС;
- анализ результатов диагностического эксперимента.

VLSI_SIM широко применяется в учебном процессе факультета прикладной математики и информатики для изучения цифровых блоков и систем, получения их моделей с разной степенью точности отражения происходящих процессов, разработки тестов контроля, изучения влияния неисправностей на функционирование устройств, организации процессов поиска неисправностей во время проведения диагностического эксперимента, при выполнении курсовых и дипломных проектов студентов по специальностям «информатика» и «прикладная математика», а также при проведении научных исследований.

Система доведена до уровня производственного применения. В системе реализованы следующие основные методы:

метод многозначного моделирования функционально-логических схем с учетом задержек компонентов (подсистема SCA_TIME);

метод многозначного динамического моделирования переключаемых структур (подсистема A_SIMULATE);

метод конкурентно-дедуктивного (параллельно-дедуктивного) моделирования неисправностей функционально-логических структур, представленной сетью компонентов, заданных автоматными моделями, на основе многозначного моделирования исходного объекта с учетом задержек компонентов (подсистема SCA_FAULT);

метод случайной генерации тестов с элементарной экспертной поддержкой (подсистема SCA_ENER);

метод анализа тестов на полноту контроля в классе неисправностей константного типа.

Система VLSI_SIM ориентирована на решение задачи верификации проектов и генерации тестов для объектов, представленных в структурном виде.

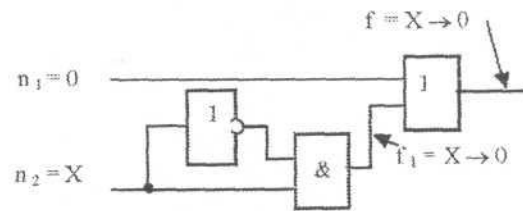
Реализованный в системе метод параллельно-дедуктивного моделирования неисправностей является наиболее эффективным из известных на практике методов. Он основан на моделировании только исправной схемы и не требует собственно моделирования ее неисправных модификаций. Метод основан на применении теоретико-множественных соотношений для вычисления обнаруживаемых неисправностей константного типа на выходах вентиляльных элементов и функционально-сложных блоков, представленных на автоматном уровне таблицами переходов-выходов. При этом используется многозначная алгебра, в которой, к сожалению, отсутствует логическое состояние, инверсное состоянию неопределенности, что может существенно уменьшить контролируемую способность теста.

Покажем, как можно улучшить результат расширения значности применяемой алгебры моделирования.

IV. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ РЕШЕНИЯ ЗАДАЧИ МОДЕЛИРОВАНИЯ НЕИСПРАВНОСТЕЙ

Рассмотрим некоторые особенности применения состояния неопределенности сигнала в линии схемы при организации моделирования неисправностей. При трюичном моделировании схемы с повторно сходящимися ветвлениями, как показано на рис. 1, результат моделирования зачастую оказывается «пессимистичным», так как в реальном режиме схема ведет себя вполне детерминированно, а при моделировании на ее выходе определяется состояние неопределенности. По правилам трюичного моделирования на выходах линий f_1 и f получается при указанных на рисунке входных сигналах состояние неопределенности X , что не позволяет вычислить методом дедуктивного

моделирования множество обнаруживаемых на выходе схемы неисправностей. Это происходит из-за того, что трюичная алгебра не имеет логического состоя-



ния,

Рис. 1. Фрагмент схемы

инверсного состоянию X . В данном случае задачу моделирования неисправностей для повышения точности результатов моделирования предлагается решать следующим образом:

1. На этапе предобработки схемы выделить элементы повторно сходящиеся ветвлений;
2. Ввести множество логических состояний X_i и \bar{X}_i , инверсных соответствующим состояниям X_i , где i – индекс принадлежности к сходящемуся ветвлению с номером i ($i = \bar{1}, k$, где k – число выделенных сходящихся ветвлений).
3. При моделировании элемента – истока сходящегося ветвления i при появлении на его выходе логического состояния X заменить это состояние на состояние X_i и использовать его в последующем при моделировании остальных элементов данного сходящегося ветвления. Таким образом, при моделировании исправной схемы мы осуществили переход от трюичной алгебры к четырехзначной с индексами, где введено множество переменных

$V = \{0, 1, X_i, \bar{X}_i\}$, на которых определены операции И (&), ИЛИ (I), НЕ (\neg) (табл. 1).

Таблица 1

Логические функции в четырехзначном алфавите

&	0	1	X	\bar{X}	I	0	1	X	\bar{X}
0	0	0	0	0	0	0	1	X	\bar{X}
1	0	1	X	\bar{X}	1	1	1	1	1
X	0	X	X	0	X	X	1	X	1
\bar{X}	0	\bar{X}	0	\bar{X}	\bar{X}	\bar{X}	1	1	\bar{X}
\neg	0	1	X	\bar{X}					

	1	0	\bar{X}	X
--	---	---	-----------	-----

Применение такого подхода, при котором состояние неопределенности X , возникающее на выходе элемента истока i -го повторно сходящегося ветвления, трансформируется в состояние X_i , позволяет существенно улучшить точность моделирования исправной схемы, а также точность сквозного вычисления множества обнаруживаемых неисправностей. Применяя данный подход при моделировании фрагмента схемы, приведенного на рисунке 1, сигналы в линиях f и f_1 оказываются равными 0, а не состоянию неопределенности, как в случае применения метода Эйхельбергера, что существенно уточняет результаты моделирования.

Моделирование неисправностей и тестовое диагностирование цифровых систем исторически выполнялось на вентиляльном уровне. Однако размерность данной задачи применительно к проектам современных СБИС в целом на уровне вентиляльного представления ограничивает возможность ее эффективного решения.

V. МОДЕЛИРОВАНИЕ НЕИСПРАВНОСТЕЙ НА УРОВНЕ ПОВЕДЕНИЯ НА ЯЗЫКЕ VHDL

Актуальность проблемы оценки контролепригодности проекта на ранних стадиях проектирования, а также разработки тестов и анализа их полноты активизирует поиск эффективных решений задачи анализа функционирования устройств при наличии неисправностей при нисходящем проектировании на основе VHDL. Решение задачи идет по двум различным направлениям. Одно из направлений исследований основано на применении аппаратных прототипов проектируемых устройств, другое — на моделировании неисправностей.

В литературе имеются сообщения о некоторых подходах к моделированию неисправностей на поведенческом уровне, в том числе основанных на применении специальных инструментариев для введения неисправностей в исходное описание объекта, грамматического анализа VHDL-описаний и моделирования. Имеются инструменты и для моделирования СБИС на структурном уровне на языке VHDL. В то же время отсутствуют эффективные методы и системы для решения задачи в целом. В ряде работ предлагаются относительно сложные модели неисправностей, в кото-

рых делается попытка связать поведенческие модели с реальными аппаратными неисправностями, однако аргументация приводимого соответствия отсутствует. Известны работы, где поведенческие неисправности подразделяются на две категории: неисправности микроопераций (micro-operation faults) и неисправности управления (control faults). Одна из них предполагает замену одних операций другими, например, операцию логического сложения на операцию умножения и наоборот, операцию арифметического сложения на операцию вычитания и т.д. Остается открытым вопрос, насколько подобная замена адекватна реальным неисправностям в соответствующих аппаратных средствах.

Недостатком предлагаемых решений является то, что отсутствует обоснование соответствия поведенческих моделей и реальных неисправностей аппаратуры. Известно, что средства синтеза объектов, описанных на языке VHDL, ориентированы на некоторое подмножество операторов языка, так называемое синтезируемое подмножество. К тому же они ориентированы на применение соответствующих библиотек компонентов, так что каждая конструкция операторов языка из его синтезируемого подмножества реализуется определенной структурой. Рассмотрим одно из возможных решений задачи построения примитивов неисправностей цифровых объектов комбинационного типа на поведенческом уровне их описания таким образом, чтобы они соответствовали неисправностям их физических реализаций. Для иллюстрации подхода выберем одну из часто применяемых конструкций языка, к примеру, if-then-else, if $Y = '0'$

```

then
  Z <= X1;
else
  Z <= X2;

```

и рассмотрим ее некоторые аппаратные реализации на основе мультиплексора. На рис. 2 приведена реализация функции на основе ДНФ, на рис. 3 та же функция реализована на основе КНФ.

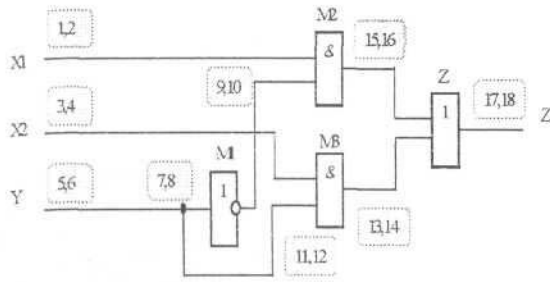


Рис. 2. Реализация конструкции <if-then-else> на основе ДНФ

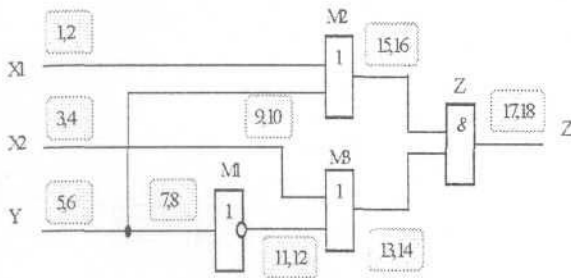


Рис. 3. Реализация конструкции <if-then-else> на основе КНФ

Сгенерируем тесты и определим покрываемые каждым набором множества неисправностей (к примеру, средствами программной системы VLSI_SIM) в классе неисправностей константного типа для обеих реализаций схем (неисправности пронумерованы на рисунках: нечетные номера - неисправности типа "const 0", четные - "const 1"). Результаты сведем в таблицу 2.

Очевидно, таблицу 2 можно значительно сократить за счет объединения одинаковых столбцов, соответствующих эквивалентным неисправностям. В таблице 3 каждый столбец соответствует группе эквивалентных неисправностей, покрываемых соответствующими входными наборами. Анализируя таблицу 3, можно выделить группы 1-4 неисправностей, которые покрываются только единственным входным набором. В то же время эти наборы покрывают и неисправности из групп 5-8. Неисправности, входящие в группы 5-8, можно исключить из дальнейшего рассмотрения, так как они будут обнаружены при обнаружении неисправностей из групп 1-4.

Получим функции неисправностей, соответствующие неисправностям групп 1-4:

для неисправностей ДНФ - реализации для группы

№1 — $Z = X2 \wedge Y$;

№2 — $Z = \bar{Y} \vee X2 \wedge Y$;

$Z = X1 \wedge \bar{Y} \vee X2$;

№3 — $Z = X1 \wedge \bar{Y}$;

№4 — $Z = Y \vee X1 \wedge \bar{Y}$;

$Z = X1 \vee X2 \wedge Y$;

для неисправностей КНФ-реализации для группы

№1 — $Z = X2 \wedge Y$;

$Z = X1 \wedge X2 \vee X2 \wedge Y$;

№2 — $Z = X2 \vee \bar{Y}$;

№3 — $Z = X1 \wedge \bar{Y}$;

$Z = X1 \wedge X2 \vee X1 \wedge \bar{Y}$

№4 — $Z = X1 \vee Y$.

Анализируя приведенные функции, получаем множество моделей поведенческих неисправностей (здесь не приводятся). Из полученного множества моделей неисправностей можно выделить четыре поведенческие неисправности, покрывающие все неисправности константного типа в приведенных реализациях (приведены на рис. 4). Обнаружение данных четырех поведенческих неисправностей обеспечивает обнаружение всех неисправностей константного типа, соответствующих реальному объекту, представленному на вентиляльном уровне в виде ДНФ или КНФ реализаций.

Неисправности типа «И»	
if Y = '0'	if Y = '0'
then	then
Z <= X1;	Z <= X1 and X2;
else	else
Z <= X1 and X2;	Z <= X2;
end if	end if
Неисправности типа «ИЛИ»	
if Y = '0'	if Y = '0'
then	then
Z <= X1 or X2;	Z <= X1;
else	else
Z <= X2;	Z <= X1 or X2;
end if	end if

Рис. 4. Неисправности на поведенческом уровне

Таким образом, «искажение» оператора if-then-else, как это показано на рис.3, является обоснованным и обеспечивает полное покрытие реальных физических неисправностей соответствующего объекта.

VI. ЗАКЛЮЧЕНИЕ

Средствами собственной системы VLSI_SIM решается задача генерации тестов для цифровых систем, представленных взаимосвязью функциональных блоков, заданных на автоматном уровне.

Таблица 2

Автоматически сгенерированные тесты и покрываемые неисправности

Неисправности →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Тест-векторы ↓	ДНФ – реализация схемы																		
011	1			0		0					0		0					0	
010	0		1				1						1		1			1	1
101	0				1	1		1			1				1			1	1
100	1	0					0		0	0						0		0	
	КНФ – реализация схемы																		
100	1	0					0		0		0		0		0		0		0
010	0		1				1				1						1		1
001	0				1	1		1					1		1				1
011	1			0		0				0				0		0		0	

Таблица 3

Группы эквивалентных неисправностей

Группы неисправностей →		1	2	3	4	5	6	7	8
Неисправности →	0	1,8,9,15	2,12	3,11,13	4,7,10	5	6	14,16,18	17
Тест-векторы ↓	ДНФ – реализация схемы								
011	1					0			0
010	0		1				1	1	
101	0				1		1	1	
100	1	0						0	0
	КНФ – реализация схемы								
Группы неисправностей →		1	2	3	4	6	7	8	
Неисправности →		1,8,11	2,10,16	3,5,9	4,7,12,14	6	13,15,17	18	
100	1	0				0	0		
010		0	1			1		1	
001		0			1			1	
011	1			0			0		

Предлагаются пути повышения эффективности решения задачи за счет более гибкой обработки состояния неопределенности при моделировании схемы. Рассматривается задача моделирования неисправностей объектов, описанных на уровне поведения на языке VHDL.

ЛИТЕРАТУРА

- [1] Baraza J.C., Gracia J.A. Prototype of a VHDL-based Fault Injection Tool: Descrip. and Application // Journal of System Arch. – 2002. – V. 47. – № 10. – P. 847-867.
- [2] Ferrandi F., Fummi F., Sciuto D. Test Generation and Testability Alternatives Exploration of Critical Algorithms for Embedded Applications // IEEE Trans. on Computers. – 2002. – V. 51. – № 2. – P. 200-215.
- [3] Zolotarevitch L.A., Baturitsky M.A. Deduktive switch-level CMOS-VLSI fault simulation // The International conference computer-aided design of diskrete devices (CAD DD'95). – Minsk-Szczecin, 1995. – V. 2. – P. 157-164.
- [4] Золоторевич, Л.А. Моделирование неисправностей в структурах СБИС на VHDL // Информатика. – 2005. – № 1. – С. 89 – 94.