

2. *Thomas, L.C.* A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers / L. C. Thomas // *International Journal of Forecasting*. 2000. V. 6. P. 49–172.
3. *Малюгин В.И.* Исследование эффективности алгоритмов классификации заемщиков банков на основе балансовых коэффициентов / В.И. Малюгин, О.И. Корчагин, Н.В. Гринь // *Банковский Вестник* № 4 2009. С. 27–32.

РАЗРАБОТКА ТЕСТОВ КОНТРОЛЯ ЦИФРОВЫХ СИСТЕМ НА ВЕРХНИХ УРОВНЯХ ПРОЕКТИРОВАНИЯ

А. В. Ильинкова

Анализируется состояние проблемы контроля сверхбольших интегральных схем (СБИС). Рассматриваются задачи направленного построения тестов контроля по поведенческому описанию объекта на уровне межрегистровых передач (RTL) на языке VHDL. Класс функциональных неисправностей, рассматриваемых при направленном построении теста, соответствует неисправностям константного типа реализаций СБИС на элементах соответствующих библиотек проектирования. Предлагается метод направленного построения теста, который позволяет на ранних этапах проектирования анализировать контролепригодность проекта и ее зависимость от применяемых технологических библиотек проектирования.

Требования к надежности цифровых систем постоянно возрастают не только в областях, в которых отказ может привести к катастрофическим событиям, но и во всех других применениях. Одной из самых важных компонентов систем обеспечения высокой надежности электронных систем является способность определить наличие или отсутствие ошибки функционирования системы. Проблема построения тестов на всем интервале развития интегральной схемотехники является одной из наукоемких проблем, которые до настоящего времени не получили эффективного теоретического и практического решения. Задача построения тестов принадлежит к классу NP- трудных проблем. В связи с высокой сложностью построения тестов контроля и диагностирования функционально-сложных цифровых систем применяются методы контролепригодного проектирования, которые направлены на снижение сложности задачи. В то же время проблема построения тестов остается высоко актуальной, так как все используемые подходы к проектированию не решают по разным причинам задачу контроля цифровых систем.

Тесты необходимы на всех этапах жизненного цикла цифровой системы. Актуальность тестов обусловлена потребностью в тестах на этапе проектирования для анализа корректности проектов на всех этапах про-

ектирования, так как методы формальной верификации не достигли уровня практического применения. На этапе производства тесты необходимы для контроля исправности готовых изделий, на этапе эксплуатации для подтверждения исправности, работоспособности или правильного функционирования устройства. На каждом из этапов жизненного цикла изделия необходимо контролировать объект с учетом моделей неисправностей, соответствующих физическим дефектам, возникающим в связи с рядом конструктивно-технологических факторов и дестабилизирующим воздействием внешней среды.

Основная цель работы - построение тестов контроля цифровых систем на верхних уровнях проектирования для своевременной оценки степени контролепригодности проекта, реализованного в разных технологических библиотеках и сокращения сроков проектирования объекта.

ОСНОВНЫЕ ЭТАПЫ

1. Выбор математической платформы для разработки программной модели объекта, представленного в виде описания поведения или RTL на языке VHDL.

2. Направленное построение теста контроля цифровой системы в целом в классе неисправностей, свойственных физическому объекту.

3. Построение тестов функциональных блоков на основе моделирования и символьных вычислений.

Решение данной задачи в указанной постановке позволяет, во-первых, уменьшить размерность сложной наукоемкой задачи, во-вторых, на ранних этапах в динамике анализировать качество проекта с точки зрения его контролепригодности, в-третьих, оценить контролепригодность проекта применительно к разным технологическим библиотекам на ранних этапах проектирования.

Как математическую платформу для построения поведенческой модели цифровой системы, представленной в виде управляющей и операционной частей, используем DD (диаграмма принятия решений) потоков данных и DD потока управления, которые строятся на основе VHDL - описания объекта. Программный код такого описания содержит операторы языка VHDL, составляющие его синтезируемое подмножество.

Пути данных распространяются через функциональные блоки, которые могут представлять собой комбинационные или последовательностные схемы. При описании системы каждая выходная и внутренняя переменная описывается в виде DD потока данных. Нетерминальные вершины DD потока данных содержат значения сигналов управления, а оконечные, терминальные, узлы описывают соответствующие данные, т.е.

состояния первичных входов, регистров, операции, константы. Управляющая часть описывается конечным автоматом, таблица переходов которого описывается единственной DD. Нетерминальные узлы представляют текущее состояние и входные условия (состояния) управляющей части, а терминальные – следующие логические состояния, которые управляют распространением и обработкой информации в системе.

Описание объекта на языке VHDL на RTL-уровне не несет информации о структурном наполнении объекта, тем не менее, на этапе структурного синтеза каждый фрагмент описания будет реализован в виде некоторой структуры в заданном элементном базисе.

Рассмотрим задачу построения теста в следующей постановке. Цифровая система представлена на RTL-уровне или уровне описания поведения. Структурная реализация системы отсутствует, но известны технологические библиотеки элементов, которые доступны на этапе его синтеза. Необходимо построить тест контроля объекта, при этом функциональные модели рассматриваемых неисправностей должны соответствовать моделям неисправностей константного типа, свойственных реальному физическому объекту. Исходными данными для решения задачи является VHDL –описание объекта. По данному описанию необходимо построить DD потоков данных и DD потока управления, которые затем необходимо перевести в множество предикатов изменения состояния системы, и затем с использованием предикатов будут определяться ограничения при активизации и распространении неисправностей.

ОБЩИЙ АЛГОРИТМ ПОСТРОЕНИЯ ТЕСТА:

1. VHDL-описанию объекта строим DD потоков данных и DD потока управления (используется синтезируемое подмножество языка VHDL).
2. Выбираем очередной оператор программного кода.
3. Выбираем очередную функциональную неисправность оператора.
4. Извлекаем функциональные ограничения для очувствления данной неисправности.
5. Извлекаем функциональные ограничения для распространения эффекта неисправности к выходам системы.
6. Выполняем разрешение ограничений.
7. Если ограничения удовлетворены, то сравниваем реакции системы по выполнению программного VHDL - кода с исходным оператором и с оператором-мутантом. Если реакции различные, то фиксируем найденный тест контроля определенной группы неисправностей.
8. В противном случае, переходим к п.3 или к п. 2, или на выход при окончании процесса или по времени.

Предложенный метод основан на построении моделей функциональных неисправностей элементов программного кода. В работе исследуются известные подходы к построению функциональных неисправностей, предлагается методика построения функциональных неисправностей по структурному представлению механизмов реализации операторов, что представлено схематично на рисунке, где показано соответствие между ошибкой оператора if-then-else (функциональной неисправностью моделируемой системы) и неисправностью константного типа некоторого «механизма» его структурной реализации.

Figure 1 illustrates the transformation of a program fragment into a logic circuit and back into a simplified program fragment.

a) Original Program Fragment:

```

Y <= C and D
if Y = '0' then
  Z <= X1;
else Z <= X2;
endif;
Z <= Z + B;
A <= Z * A;

```

b) Logic Circuit Diagram:

The circuit takes inputs $X1$, $X2$, and Y , and a constant 0 . It uses multiplexers $M1$, $M2$, and $M3$, and an adder (labeled $+$). The output is Z . The circuit implements the logic where Z is updated based on the value of Y and the inputs $X1$ and $X2$.

c) Resulting Program Fragment:

```

if Y = '0' then
  Z <= 0;
else Z <= X2;
end if

```

1. *Золоторевич Л. А.* Построение тестов контроля цифровых систем на уровне межрегистровых передач // Л.А. Золоторевич, А.В. Ильинкова // Информатика. 2010. №1. С. 112–121.
2. *Золоторевич Л. А.* Разработка тестов для анализа контролепригодности СБИС на верхних уровнях проектирования // Л.А. Золоторевич, А.В. Ильинкова // Автоматика и телемеханика. 2010. В печати.

3. *Золоторевич Л.А.* Экспериментальное исследование базовых методов моделирования в системе ModelSim / Л.А. Золоторевич, А.В. Ильинкова // The International Conference Computer– Aided Design of Diskrete Devices (CAD-DD'07). Vol.1. Minsk. 2007. P. 245–251.
4. *Золоторевич Л.А.* Анализ полноты тестов контроля цифровых устройств на основе решения задачи выполнимости // Л.А. Золоторевич, А.В. Ильинкова // Материалы Междунар. науч.-практ. конф. «Современная радиоэлектроника: научные исследования и подготовка кадров». Минск, 23–24 апреля 2008. С. 25–27.
5. *Золоторевич Л.А.* Анализ тестов контроля цифровых устройств по отрицательной нормальной форме реализуемой функции // Л.А. Золоторевич, А.В. Ильинкова // The Fifth International Scientific Conference «Information technologies in industry» ITI* 2008. P. 107–108.
6. *Ibarra O. H., Sahni S.* Polynomially Complete Fault Detection Problems // IEEE Transactions on Computers. 1975. V. C. 24. No. 3. P. 242–249.
7. *Murray B. T., Hayes J. P.* Hierarchical Test Generation Using Precomputed Tests for Modules // International Test Conference. 1988. P. 221–229.
8. *Ubar R.* Test Synthesis with Alternative Graphs // IEEE Design and Test of Computers. 1996. V. 13. No. 1. P. 48–57.
9. *Jervan G., Peng Z., Ubar R.* Test Cost Minimization for Hybrid BIST // IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT'00). 2000. P. 283–291.
10. *Roth J.P.* Diagnosis of automata failures: a calculus and a method // IEEE Trans. on Computers. 1966. V. 15. No. 7. P. 278–291.

ЧИСЛЕННЫЙ АНАЛИЗ ОДНОГО ПАРАРЕАЛЬНОГО АЛГОРИТМА РЕШЕНИЯ СИСТЕМ ОДУ

В. А. Карпейчик

В настоящее время особенно актуальным становится вопрос наличия алгоритмов, способных выполняться параллельно несколькими вычислительными модулями, так как при исполнении последовательного алгоритма на вычислительной системе, использующей несколько вычислительных модулей, вычисления будут производиться только на одном из них. Важной проблемой при моделировании и решении сложных задач является решение дифференциальных уравнений и их систем. В данной работе проведено численное исследование одного парареального метода решения систем ОДУ, впервые представленного в 2001 году Габриэлем Туриничи и Айвоном Мэдэем[1]. Парареальным они называют алгоритм решения исходной задачи, позволяющий производить параллельные вычисления на отрезках временной оси дифференциальной задачи.

Пусть имеем дифференциальную задачу

$$\begin{aligned} u'(t) &= f(u(t)), \\ u(T_0) &= u^0, \end{aligned} \tag{1}$$