

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК БЕЛАРУСИ

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ ДИСКРЕТНЫХ СИСТЕМ

*Материалы
Шестой Международной конференции
14–15 ноября 2007 г., Минск*

В 2 томах

Том 2

COMPUTER-AIDED DESIGN
OF DISCRETE DEVICES
CAD DD'07

*Proceedings
of the Sixth International Conference
Minsk, November 14–15, 2007*

Volume 2

Минск 2007

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ БАЗОВЫХ МЕТОДОВ МОДЕЛИРОВАНИЯ В СИСТЕМЕ MODELSIM

Л.А. Золоторевич, А.В. Ильинкова
Белорусский государственный университет, Минск

Рассматриваются особенности применения фирменных средств моделирования цифровых систем на структурном уровне. Приводятся некоторые результаты исследования точности моделирования цифровых структур в программной системе ModelSim 5.7 f фирмы Mentor Graphics.

Введение

Одним из наиболее критичных экономических показателей в современной микроэлектронике является время проектирования изделия, которое существенно определяет время выхода изделия на рынок. Более половины временных затрат на проектирование приходится на решение задач верификации проектов. Задача верификации проектов на основе методов формальной верификации осложняется чрезмерной сложностью спецификаций функционально сложных цифровых систем. Как правило, под формализацией какого-то описания системы понимается построение спецификации, позволяющей получать численные или символические однозначно интерпретируемые характеристики протекающих в системе процессов. Формальная верификация проектов цифровых систем – это возможность получения однозначного ответа о корректности системы, имея лишь некоторое формальное описание результатов проектирования и исходной спецификации. При этом под «формальностью» понимается то, что между символами, которые описывают спецификацию, существуют только явно описанные по тем или иным формальным правилам связи. Иными словами, в основе всех алгоритмов формальной верификации лежат те или иные алгоритмы анализа логических условий, в терминах которых описаны условия корректности. К сожалению, алгоритмы формальной верификации связаны с большим перебором и чаще всего при их реализации не удается избежать «комбинаторного взрыва». Кроме того, задача формализации описания зачастую представляет собой отдельную тяжело решаемую на практике задачу. Поэтому до настоящего времени верификация проектов основывается на моделировании. В связи с этим для повышения эффективности верификации проектов сложных цифровых систем на разных уровнях проектирования требуется развитие как практически пригодных методов и средств моделирования, так и теоретической базы для формальной верификации.

Используемые для верификации проектов средства должны позволять при переходе от одного уровня к другому проверить соответствие поведения проектируемого объекта желаемому (а также соответствие проектов по некоторым другим заданным критериям). При проверке «правильности поведения» необходимо учитывать все те аспекты функционирования объекта, которые влияют на детерминированность переходов и функциональную устойчивость устройства.

Разработка маршрута проектирования предполагает наличие не только программных средств моделирования, но и результатов их исследования как с точки зрения затрат вычислительных ресурсов, так и точности получаемых моделей. При выборе и/или разработке средств моделирования для решения проблем верификации цифровых сис-

тем необходимо обеспечить основной аспект – требуемую точность получаемых результатов моделирования с учетом конструктивно-технологических факторов и дестабилизирующего воздействия внешней среды, а также получить множество моделируемых входных состояний, способных обеспечить полную проверку правильности функционирования системы относительно некоторой имеющейся спецификации.

При разработке технологии проектирования, а также при разработке конкретного проекта важно знать возможности базовых методов и механизмов, реализованных в используемых программных системах моделирования, чтобы определить область их наиболее эффективного применения и сократить сроки проектирования. В литературе известны методы и алгоритмы моделирования, разработанные для применения на функционально-логическом уровне [1–11]. Но для того чтобы осознанно применять определенное программное средство и получать требуемую точность моделирования, необходимо знать, какие методы и алгоритмы реализованы в данном средстве.

В работе приводятся некоторые особенности эксплуатации фирменных систем моделирования на примере программной системы ModelSim 5.7f, предназначенной для моделирования цифровых объектов на языке VHDL [12].

1. Особенности моделирования цифровых устройств в системе ModelSim 5.7f

Применение приобретаемых для использования фирменных систем проектирования изделий микроэлектроники сопряжено с рядом трудностей, которые необходимо преодолевать на этапе разработки собственной технологии (маршрута) проектирования. Указанные трудности обусловлены тем, что система проектирования представляет собой наукоемкий продукт, описание которого, как правило, расплывчато, многословно и мало пригодно для быстрого практического освоения и применения. Кроме того, в документации практически отсутствует описание реализованных методов и алгоритмов построения моделей. Поэтому для определения областей эффективного применения фирменных зарубежных систем требуется их тщательное исследование с целью определения базовых методов, выяснения точности получаемых моделей и уточнения маршрута проектирования.

В настоящее время наиболее распространенной программной системой моделирования цифровых систем на основе использования языка VHDL является система ModelSim фирмы Mentor Graphics. Ниже приведены некоторые особенности, которые необходимо учитывать при отработке проекта в среде данной системы на этапе функционально-логического описания объекта, когда важно правильно интерпретировать результаты моделирования проектируемого объекта с учетом имеющихся возможностей учета задержек компонентов. Очевидно, что описание объекта на любом уровне включает описание блоков разной функциональной сложности. Из теории цифровых систем известно, что существенное влияние на правильность отработки заданной функции оказывают состязания сигналов в линиях связи и их последствия. Известно также, что для успешного решения проблемы моделирования состязаний применяются методы многозначного моделирования с учетом номинальных значений задержек компонентов устройства и линий связи.

При моделировании цифровых объектов на основе использования языка VHDL для описания сигнала применяется восьмизначный алфавит $V = \{U, 0, 1, X, L, H, W, Z\}$. Здесь U – состояние начальной неопределенности, символы $\{0, 1, X\}$ означают состояния сильных сигналов 0, 1 и «не определено», а также символы $\{L, H, W\}$ слабых сигналов соответственно.

Имеется возможность описания задержек сигналов. Однако точность моделирования существенно зависит от того, какие использованы методы моделирования и как реализуются указанные выше языковые возможности в соответствующем инструментальном средстве.

Рассмотрим вначале механизм моделирования в системе ModelSim 5.7f простейшего трехвходового вентиля OR с учетом инерционной задержки t -элемента, равной его простой (транспортной) составляющей; $t = 21$ ns (рис. 1).

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity OR3 is
port ( X1, X2, X3: in STD_LOGIC;
      Y: out STD_LOGIC);
end OR3;
architecture ARCHOR of OR3 is
begin
Y <= X1 or X2 or X3 after 21 NS;
end ARCHOR;
entity test1_OR is
end test1_OR;
architecture ARCH_test1_OR of test1_OR is
component OR3
port ( X1, X2, X3: in STD_LOGIC;
      Y: out STD_LOGIC);
end component;
signal X1, X2, X3, Y: STD_LOGIC;
begin

```

```

X1 <= '0', '1' after 100 ns, '0' after 120 ns;
X2 <= '0', '1' after 121 ns, '0' after 141 ns;
X3 <= '0', '1' after 142 ns, '0' after 162 ns;

```

```

X1 <= '0', '1' after 100 ns, '0' after 107 ns;
X2 <= '0', '1' after 107 ns, '0' after 114 ns;
X3 <= '0', '1' after 114 ns, '0' after 121 ns;

```

```

L1: OR3 port map (X1, X2, X3, Y);
end ARCH_test1_OR;

```

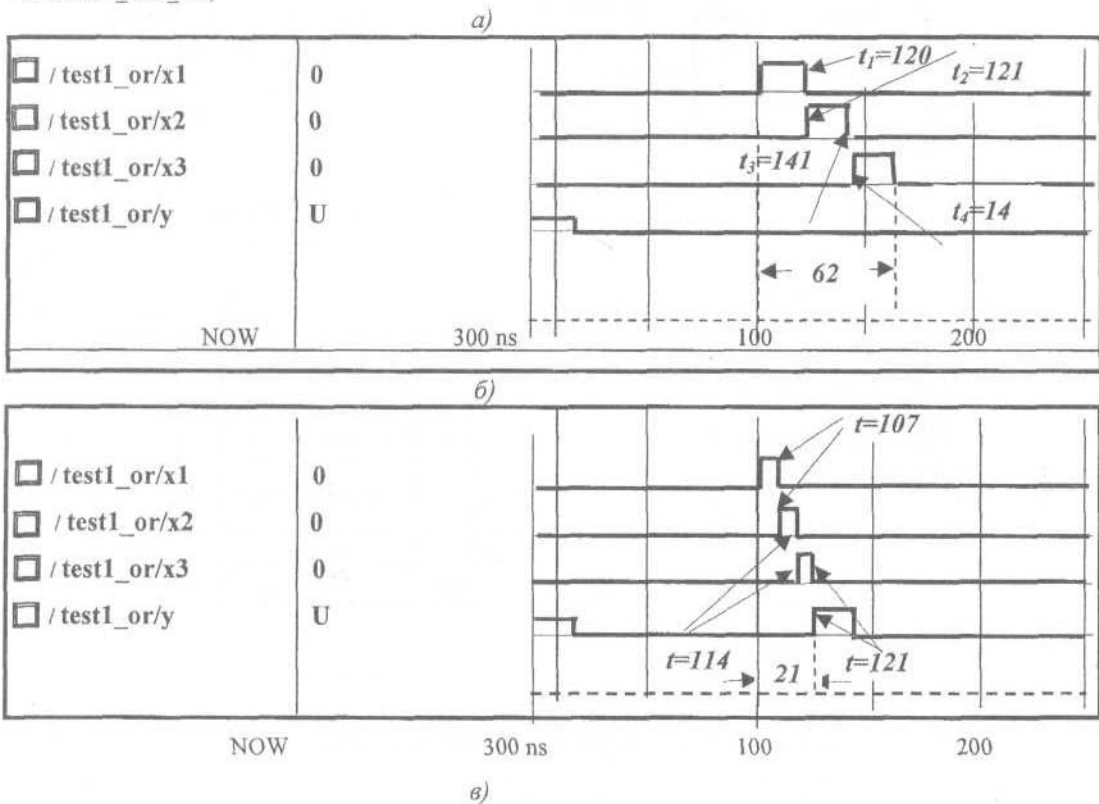
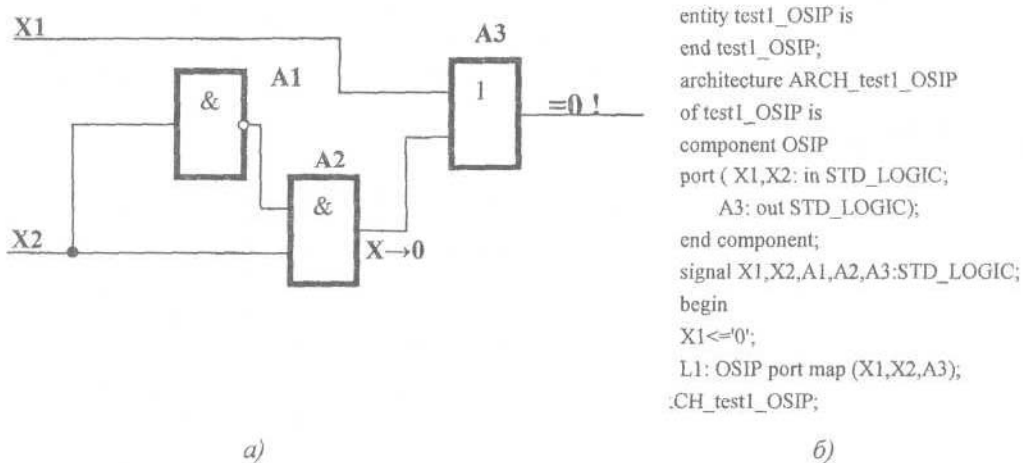


Рис. 1. Моделирование элемента OR с входами X1, X2, X3 и выходом Y в системе ModelSim 5.7f: а) описание на языке VHDL; б) пример 1; в) пример 2

На рис. 1, а приведены моделируемые входные воздействия, описанные на языке VHDL. Результаты моделирования приведены на рис. 1, б, в. В примере 1 сигнал на выходе элемента OR с задержкой 21 ns не изменился, а в примере 2 сигнал переключился дважды. Схемотехнический анализ и физическое моделирование данного эксперимента показывают, что при небольшой (в зависимости от ряда конструктивно-технологических факторов) длительности интервалов времени $[t_1 - t_2]$ и $[t_3 - t_4]$ (рис. 1, б) на выходе элемента произойдет двойное переключение сигнала. Поэтому системы моделирования должны иметь возможность адаптации программных средств к реальным условиям применения. Следовательно, полученные результаты моделирования в примере 1 могут не соответствовать реальному поведению элемента. Для уточнения заложенного механизма при адаптации программы моделирования к конкретным условиям требуется проведение физического эксперимента в рамках применяемой технологии производства. Недооценка данного момента при моделировании может повлечь определенные трудности при работе с проектом.

2. Моделирование устройств с повторно-сходящимися ветвлениями

На рис. 2, а приведен простой фрагмент цифровой сети, в которой имеется два пути распространения сигнала от входа X2 к входам элемента A2.



<input type="checkbox"/>	/ test1_opis/x1	0				
<input type="checkbox"/>	/ test1_opis/x2	U				
<input type="checkbox"/>	/ test1_opis/a1	U				
<input type="checkbox"/>	/ test1_opis/a2	U				
<input type="checkbox"/>	/ test1_opis/a3	U				
	Now	800 ns	250	500		

Рис. 2. Моделирование фрагмента с повторно-сходящимися ветвлениями:
а) логическая схема; б) описание входного воздействия; в) результаты моделирования

Схемотехнический анализ фрагмента показывает, что на выходах элементов A2 и A3 формируется сигнал логической единицы короткой длительности при переключе-

нии сигнала на входе X2 из нуля в единицу и постоянном нуле на входе X1. При переключении на X2 из единицы в ноль на указанных выходах сохраняется сигнал постоянного нуля. Если сигнал на входе X2 имеет любое определенное состояние (нуля или единицы), то на выходе A2 и A3 будет сигнал постоянного нуля. Однако подобное поведение схемы не подтверждается результатами моделирования. Подадим на вход X1 сигнал постоянного нуля. На входе X2 по умолчанию подается состояние начальной неопределенности (левое значение сигнала в расположении элементов типа STD_LOGIC).

На рис. 2, в приведены результаты данного эксперимента. На выходе элемента A3 должно быть состояние логического нуля, а не состояние неопределенности, которое получено при моделировании. Здесь следует отметить, что в программе может возникать неточность результатов из-за того, что не учитываются некоторые особенности моделирования элементов повторно-сходящихся ветвлений. В литературе известны более точные методы моделирования подобных схем, не реализованные в рассматриваемой программной системе [7, 9, 11].

3. Особенности использования «дельта-задержки» сигнала

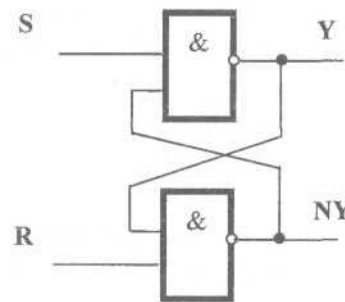
При моделировании цифрового объекта на этапе верификации проекта даже на рабочих входных воздействиях возможны разные сочетания сигналов на входах компонентов из-за несогласованности задержек распространения сигналов в линиях связи и появления критических состязаний.

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity RS is
port ( R,S: in STD_LOGIC;
      Y,NY: inout STD_LOGIC);
end RS;
architecture ARCHRS of RS is
begin
Y<= S nand NY;
NY<= R nand Y;
end ARCHRS;
entity test1_RS is
end test1_RS;
library IEEE;
use IEEE.STD_LOGIC_1164.all;
architecture ARCH_test1_RS of test1_RS is
component RS
port ( R,S: in STD_LOGIC;
      Y,NY: inout STD_LOGIC);
end component;
signal R,S,Y,NY:STD_LOGIC;
begin
S <= '0', '1' after 100 ns, '0' after 120 ns, '0' after 200 ns, '1' after 220 ns,
'1' after 300 ns, '0' after 320 ns, '1' after 400 ns, '0' after 420 ns;
R <= '1', '1' after 100 ns, '0' after 120 ns, '1' after 200 ns, '0' after 220 ns,
'1' after 300 ns, '0' after 320 ns, '1' after 400 ns, '1' after 420 ns;
L1: RS port map (R,S,Y,NY);
end ARCH_test1_RS;

```

a)



б)

Рис. 3. Моделирование RS-триггера:
а) логическая схема; б) описание входного воздействия

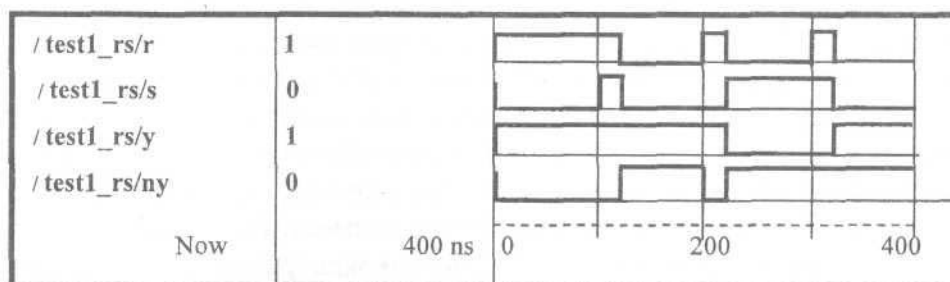


Рис. 4. Результаты моделирования RS-триггера

На рис. 3, 4 приведен пример моделирования асинхронного RS-триггера на разных входных воздействиях при использовании «дельта-задержки» распространения сигнала через элемент. Моделирование заканчивается сообщением об ошибке

<# ** Error: (vsim-3601) Iteration limit reached at time 400 ns>.

На данном входном воздействии конечное состояние триггера должно быть определенным и Y равно единице. Полученная временная диаграмма не отражает истинного конечного состояния триггера. При задании единичной задержки сигнала на элементе результат моделирования оказывается правильным.

Таким образом, результаты моделирования оказываются качественно разными в зависимости от того, задержка сигнала применяется по умолчанию как бесконечно малая величина или в проекте задается единичное значение задержки. При «поточковом» описании схем с обратными связями необходимо задать хотя бы единичную задержку элементам, если не известно ее номинальное значение, и результат моделирования будет более адекватным, чем при использовании модели дельта-задержки сигнала на элементе.

Заключение

В результате проведенных исследований можно сделать следующие выводы:

1. Система ModelSim 5.7 f позволяет моделировать в девятизначной логике структурные представления цифровых объектов с учетом номинальных значений задержек (но не разбросов задержек) компонентов. Нет возможности моделирования объекта в реальном спектре значений задержек компонентов и линий связи, что не позволяет эффективно решать некоторые задачи верификации до разработки топологического проекта.

2. При моделировании может возникать неточность результатов из-за того, что не учитываются некоторые особенности моделирования элементов повторно-сходящихся ветвлений.

3. Модели компонентов основываются на вынесении сосредоточенной задержки к выходам компонента. Это не позволяет непосредственным образом промоделировать задержку, приписанную к некоторому входно-выходному тракту передачи информации, а также адаптировать механизм высокочастотной отсечки сигнала к разным условиям применения.

4. Как в языке VHDL, так и в системе ModelSim не предусмотрена возможность моделирования устройств на переключательном уровне. Высокая актуальность этой задачи подтверждает целесообразность комплексирования средств проектирования.

5. В системе ModelSim нет возможности простым образом, без написания отдельного программного блока, называемого испытательным стендом, организовать процесс подачи на моделируемую схему входных воздействий (известных или случайных).

6. Система ModelSim предназначена для моделирования при верификации проектов, но не позволяет эффективно моделировать объект на этапе разработки средств диагностики. В ней нет возможностей для эффективного моделирования неисправностей и, значит, для решения задачи генерации и анализа тестов контроля.

Список литературы

1. Eichelberger E.V. Hazard detection in combinational and sequential switching circuits // IBM Journ. Res. and Dev. – 1965. – Vol. 9, no. 2. – P. 90–99.
2. Биргер А.Г. Метод моделирования дискретных устройств // Автоматика и телемеханика. – 1981. – № 1. – С. 138–144.
3. Бондарь С.Б. Девятисимвольное моделирование логических схем // Цифровые устройства и микропроцессоры. – Рига, 1980. – С. 143–151.
4. Бродский М.А. Моделирование логических схем с учетом задержек элементов // Контроль цифровых схем. – Вып. 19. – М.: ИНЭУМ, 1975. – С. 13–20.
5. Гончаровский О.В., Кон Е.Л. Тройное моделирование цифровых автоматов с учетом задержек // Автоматика и вычислительная техника. – 1977. – № 3. – С. 45–47.
6. Золоторевич Л.А. Определение длительности переходного процесса в логических схемах // Автоматика и вычислительная техника. – 1977. – № 6.
7. Золоторевич Л.А. Многозначное моделирование дискретных устройств без сходящихся ветвлений с учетом разбросов задержек на срабатывание логических элементов // Современные вычислительные и автоматизированные системы: сб. науч. тр. – Минск: БГУ, 1980.
8. Золоторевич Л.А. Интервальная временная булева алгебра и ее применение для динамического анализа проектируемых устройств ЭВМ // Автоматика и вычислительная техника. – Рига. – 1984. – № 4. – С. 81–88.
9. Золоторевич Л.А. Интервальное моделирование цифровых устройств со сходящимися ветвлениями // Проблемные вопросы автоматизации производства и обработки информации: сб. науч. тр. – Минск: БГУ, 1987. – С. 93–103.
10. Золоторевич Л.А. Разработка методов верификации СБИС и ППЭВМ на их основе // Тр. Междунар. симпозиума INFO-89, Минск, Беларусь. – Минск, 1989. – Т. 2. – С. 115–120.
11. Золоторевич Л.А. Исследование методов и средств верификации проектов и генерации тестов МЭС // Сб. науч. тр. Всероссийской науч.-техн. конф. «Проблемы разработки перспективных микроэлектронных систем – 2006». – М.: ИППМ РАН, 2006. – С. 163–168.
12. Бибило П.Н. Основы языка VHDL. – М.: Солон-Р, 2000. – 200 с.