Санкт-Петербургский государственный университет
Кафедра теории систем управления электрофизической аппаратурой

Нуракунов Алишер

Выпускная квалификационная работа бакалавра

Тестирование цифровых устройств

010900
Прикладная математика и физика

Научный руководитель,
доктор физ.-мат. наук,
профессор Овсянников Д.А.

Санкт-Петербург
2016

Содержание

Введение

XXI век - век технологий. Если посмотреть по сторонам, то можно увидеть, что почти у каждого человека есть одно или более электронное цифровое устройство, будь то телефон, планшет или ноутбук. Каждое из устройств состоит из множества компонентов. А каждый компонент в свою очередь производится серийно. Серийное производство подразумевает изготовление тысячи деталей для различных устройств ежедневно. И несмотря на тот факт, что процесс производства микросхем почти целиком автоматизирован, отстается какой-то процент брака. Для контроля за качеством продукта создаются целые отделы, поэтому из года в год требования к качеству технического контроля работоспособности устройств повышаются.

Проблема тестирования и диагностики появилась при производстве первых микросхем, а актуальность данной проблемы остается высокой по сей день. Для обеспечения надёжности, в процессе производства и эксплуатации применяются средства и методы технической диагностики, позволяющих осуществить проверку функциональности микросхемы и локализацию неисправности.

В настоящее время сложность устройств выросла до такого уровня, что ее тестирование практически невозможно без использования методов автоматизации.

На данном этапе развития области существуют несколько методов тестирования:

1. Визуальный автоматизированный контроль. Метод тестирования, основанный на распознавании изображения схемы. Используется почти на любом промышленном производстве в качестве предварительной проверки качества схемы.

2. Внутрисхемное тестирование. Метод, использующий пробники и набор или матрицу контактов внутри самой схемы. Главным минусом метода является требование использования дорогостоящего обородувания.

3. Граничное сканирование (boundary scan). Метод тестирования, используемый для микросхем с компонентами, поддерживающими стандарт IEEE 1149.

4. Функциональное тестирование. Метод тестирования, проверяющий функциональность схемы или ее компонент отдельно.

Один из этих методов, а именно функциональный, был рассмотрен в данной бакалаврской работе. Функциональный метод диагностики и тестирования цифровых устройств обладает рядом преимуществ, из которых стоит

выделить имитацию фактической работы схемы и невысокую требовательность к дополнительной аппаратуре, т.е. ее дешевизну.

При использовании функционального метода тестирования огромную роль играют способы описания электронной модели устройства и способы моделирования генерации входных воздействий для осуществления контроля.

В рамках данной работы были изучены программная среда Altera QUARTUS II, находящаяся на передовых позициях в области построения моделей микросхем, и САПР SimTest, разработаная в СПбГУ и уже успешно внедренная в ряд предприятий, таких как ОАО "Авангард" .

Данные программные средства были использованы для моделирования и создания тест-программы на примере реальной микросхемы.

Постановка задачи

Рассматривается цифровая схема. Она представляет собой набор элементов с известной функциональностью и связей между этими элементами. Знание общего функционального назначения схемы не является необходимым. С физической точки зрения схема представляет собой устройство с краевыми разъемами, используемыми для подачи и снятия сигналов.

Требуется:

1. Разработать программную модель каждого элемента на основе его функциональности, используя один из языков описания аппаратуры (HDL).
2. На основе моделей элементов и структурного представления объекта контроля реализовать его программную модель, используя программную среду проектирования.
3. Написать тест-программу для данного объекта контроля. Под тестовой программой понимается набор векторов входных воздействий и выходных сигналов реакции схемы. Генерацию сигналов и моделирование их подачи на модель объекта тестового контроля осуществить с помощью САПР SimTest.

Глава 1. Создание тестов для цифровой аппаратуры с помощью САПР SimTest

1.1 Физическое представление устройства

САПР - система автоматического проектирования, используемая для создания и проверки тестов.

Цифровое устройство представляет собой набор элементов с известной функциональностью и связей между этими элементами. Каждый отдельный элемент функционирует по собственному алгоритму. Совокупность алгоритмов элементов, связей между ними и входных сигналов дает представление о функциональности самого устройства.

Физическая модель устройства – это электронная плата с краевыми разъемами, используемыми для подачи и снятия сигналов. На рисунке 1 изображено физическое представление устройства, которое управляется тактируемым сигналом T.

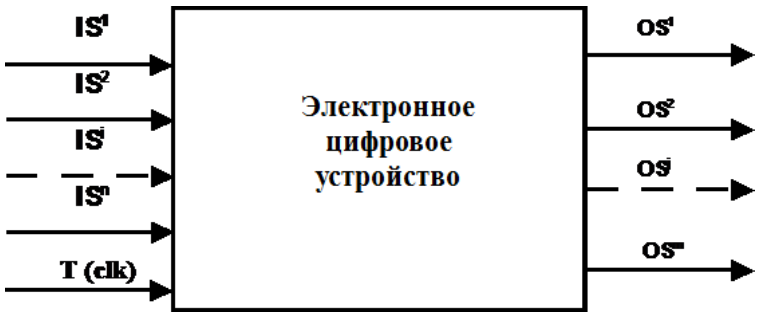


Рис. 1: Физическая модель устройства

В качестве тестирования схемы подразумевается проверка ее правильного функционирования. Составленный соответственно тест в рамках данного подхода проверяет исправность внутренних элементов и связей между ними. В данном подходе считается, что устройство разработано в соответсвии с его функциональностью, не содержит схематических ошибок и работает исправно при его реализации. Таким образом, работоспособным считается устройство без производственных браков и функционируемое в соответствии с составленной аннотацией.

В этом случае проверяется не функциональность цифровой схемы, а отсутствие неисправностей физического характера. Тогда под тестом исходного устройства понимается набор входных и выходных сигналов, зависящих от времени. Получить такой набор можно 2-мя способами:

1. Первый способ заключается в работе с эталонным, заведомо исправным устройством. Подавая наборы входных сигналов к входными контактам устройства и фиксируя выходные сигналы, мы получим искомый тестовый набор.

2. Второй способ состоит в программном моделировании устройства. Построив программную модель устройства, промоделировав подачу входных сигналов и фиксируя выходные, получим искомый тестовый набор.

Данный метод будет рассмотрен в следующем параграфе.

1.2 Программная модель устройства

Под программной моделью устройства понимается программа, моделирующая функциональность устройства. Она принимает входные сигналы и, обработав их в соответсвии с функциональностью устройства, предоставляет набор выходных сигналов. Обработка сигналов происходит путем присваивания каждому входному сигналу какой-либо переменной и работы с самой переменной соответственно. Переменные делятся на 3 типа:

1. Входные. Входные переменные программы описывают входные сигналы на реальной плате.

2. Выходные. Переменные, которые соответсвуют выходным сигналам реальной платы.

3. Внутренние. Переменные, используемые для описания внутреннего взаимодействия между элементами. Отражают связи между элементами и являются аналогами проводов, соединяющих внутренние блоки.

Модель устройства описывается одним из языков описания электронных устройств (HDL = Hardware Dischription Language). В данной работе используется язык Verilog HDL, так как он обладает рядом преимуществ: простой синтаксис, наличие систем проектирования схем, оперирующих с файлами данного типа, в свободном доступе.

Модель устройства на Verilog является главным модулем (top module), из которого вызываются, в соответствии со структурными связями, программные модели компонент (component modules), входящих в устройство (рис. 2).

Построение теста для программной модели, также как и для физической, состоит из экспертного подбора входных сигналов, который бы активировал все элементы устройства и изменял бы выходные сигналы в соответсвии с функционалом схемы. При моделировании устройства появляется возможность отслеживать неисправности внутри схем, что является огромным плюсом для проверки устройства. Это позволяет резко снизить время, требуемое для локализации неисправности схемы, и минимизировать шанс повредить само устройство.

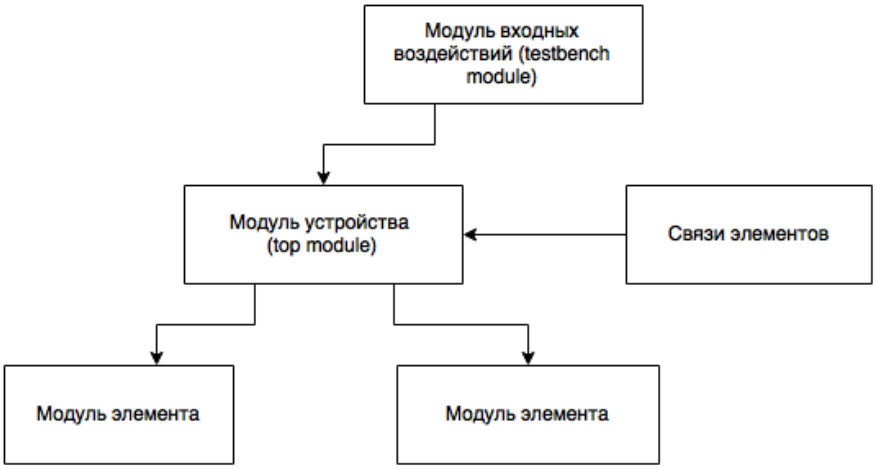


Рис. 2: Программная модель устройства

1.3 Критерий качества теста

В качестве основного критерия качества теста выступает тестовое покрытие. Покрытием называют отношение активированных линий сигналов устройства N_{act} к общему числу сигнальных линий N устройства

$$P = \frac{N_{act}}{N} \cdot 100\%$$

Активированной считается линия сигнала, которая в течение работы программы поменяла свое значение хотя бы один раз.

При разработке теста требуется добиться максимально возможного покрытия. Значение покрытия 100% будет означать, что все сигнальные линии переключились хотя бы один раз, т.е. будет свидетельствовать о исправности связей устройства, но не внутренних элементов. Данный критерий не учитывает, что некоторые сигнальные линии могут быть подключены к источникам постоянного напряжения для правильной работы устройства. В таком случае, даже тест с меньшим значением покрытия может являться достаточным.

Т.к. тест на покрытие проверяет лишь исправность связей внутри устройства, то помимо подобранных для данного теста входных воздействий требуется также отыскание специальных входных сигналов, гарантирующих проверку отдельных сложных элементов схемы. Данный поиск производится экспертным путем.

1.4 Разработка теста

Этапы разработки теста представлены в блок-схеме 3.

Разберем более подробно каждый этап построения теста.

1. Разработка моделей элементов производится человеком самостоятель-

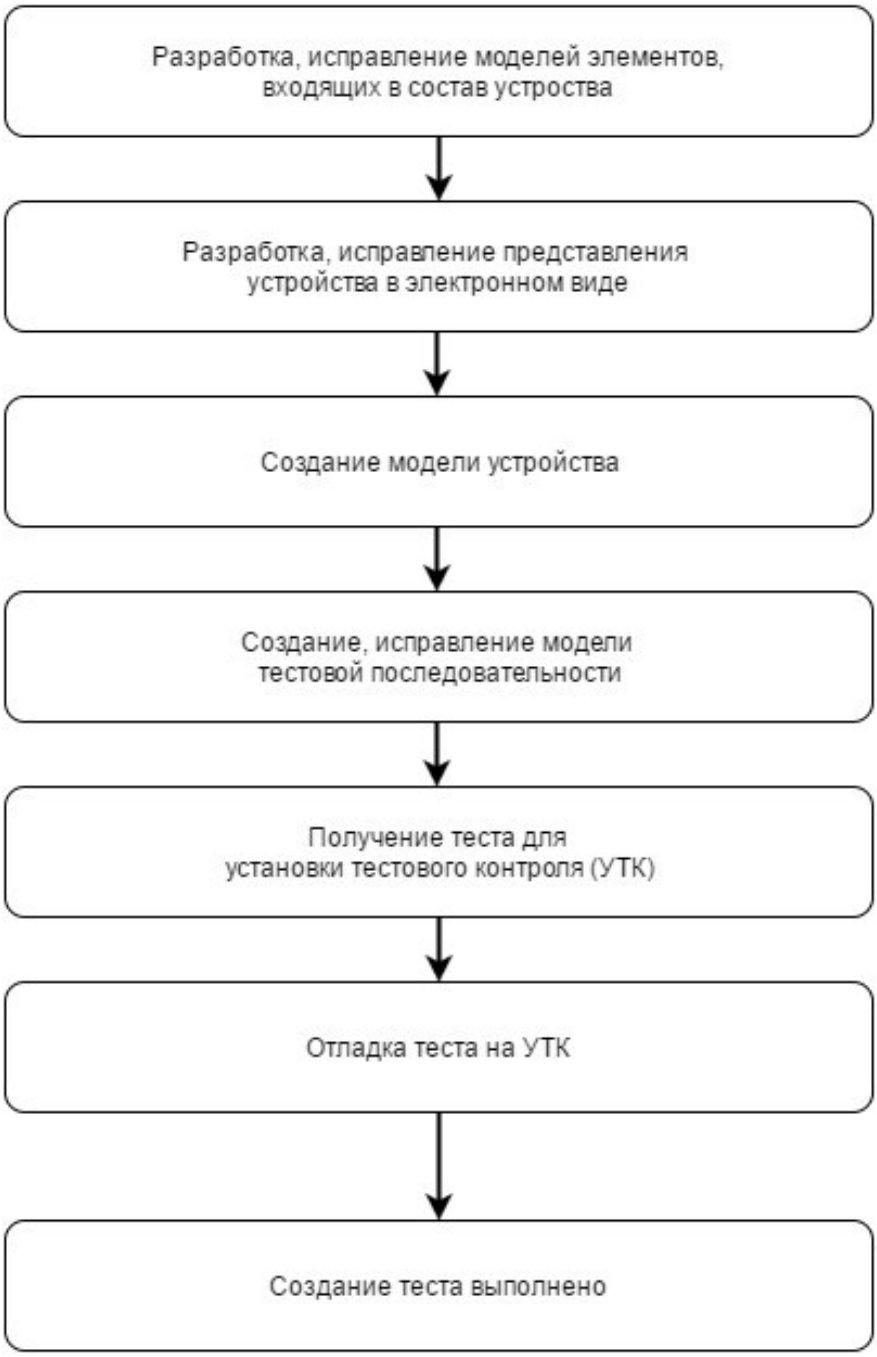


Рис. 3: Этапы формирования теста

но. Для описания моделей элементов используется один из языков HDL, например Verilog HDL. Далее, файл разработанной модели заносится в базу знаний системы САПР.

2. Разработка модели устройства производится с помощью программной среды разработки схем. Одним из примеров таких сред является мультиплатформенная среда проектирования QUARTUS II от компании Altera.

3. Модель устройства создается автоматически, используя электронный вид, сгенерированный в пункте 2.

4. Подбирается набор входных воздействий, которые будут представлены в тестовой последовательности. Моделируется поведение всех компонентов программной модели с использованием тестовой последовательности. Далее происходит анализ покрытия, о достаточности которого судит пользователь. При положительной оценке пользователя формируется тест для дальнейшей работы с устройством тестового контроля.

Оставшиеся пункты выполняются уже на установке тестового контроля непосредственно.

1.5 Построение моделей входных сигналов интерфейсным методом

Основной проблемой создания модели тестовой последовательности является нахождение последовательностей входных сигналов схемы, которые бы активировали при моделировании все элементы устройства, изменяя при этом выходные сигналы схемы.

Метод, используемый в системе, базируется на экспертных знаниях пользователя. При этом не нужно изучать функциональность устройства как целого, а достаточно выделить в нем функциональные узлы (логические интерфейсы), связанные с входными сигналами.

Структура устройства с помощью логических интерфейсов приобретает вид 4.

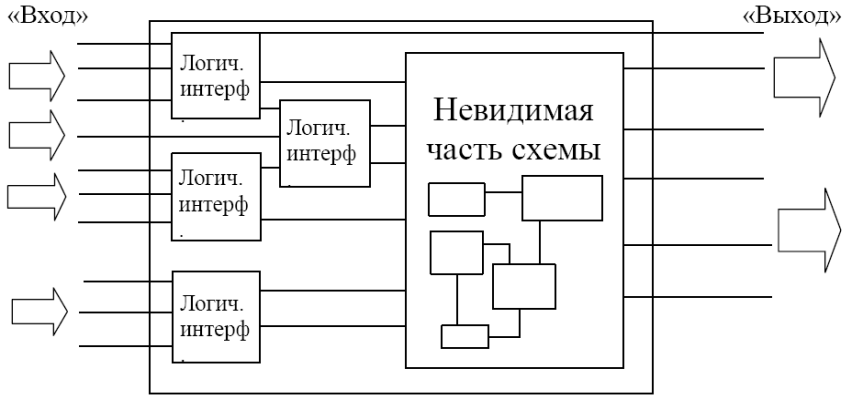


Рис. 4: Интерфейсная структура устройства

На рисунке 4 показана общая структура любого исследуемого устройства. За 'вход' обозначены входные воздействия на устройство, за 'выход' - выходные сигналы, являющиеся результатом реакции устройства на входные воздействия. Логические интерфейсы получаются путем выбора элементов, непосредственно соединненых с входными сигналами. Невидимая часть - та часть схемы, к которой нет прямого доступа.

Представление краевых элементов схем в виде логических интерфейсов выводит нас на новый, более высокий уровень абстракции для генера-

ции входных воздействий. Новый уровень абстракции дает возможность создания тестовых сигналов, на уровне команд интерфейса, а не на более низком уровне сигналов, что упрощает создание тестов. Приведем несколько примеров логических интерфейсов:

1. Генератор тактовых сигналов (ClockInterface). Используется для моделирования входных сигналов на счетчик, запись на котором срабатывает по приходу положительного фронта сигнала, т.е. при переходе сигнала из "логического 0" в "логическую единицу" . Этот интерфейс позволяет задавать изменение значения входного сигнала с требуемым промежутком времени вместо того, чтобы подавать отдельный сигнал на каждом временном шаге.

2. Стробируемый импульс (StrobeInterface). Используется для смены значения сигнала на определенном временном шаге, при этом сохраняет значение входного сигнала на остальном промежутке времени.

Программу, осуществляющую генерацию входных сигналов на языке логических интерфейсов, называют скриптом. Скрипт является текстовым файлом, содержащим набор команд логических интерфейсов. После обработки скрипта автоматически составляется модель входных сигналов для всех краевых сигналов устройства, связанных с логическими линиями интерфейсов.

Результаты моделирования работы схемы анализируются для нахождения набора сигналов, которые давали бы достаточный уровень покрытия. На этом заканчивается этап формирования теста для цифрового устройства.

Глава 2. Описание используемых программных средств

2.1 Язык описания электронных устройств Verilog HDL

Язык Verilog предназначен для описания цифрового оборудования. Каждая отдельная программа на языке Verilog представляет собой модель соответствующего цифрового устройства. Основной принцип поведенческого проектирования электронных схем можно сформулировать следующим образом: если модель полностью определяет функциональность устройства и его реакцию на любые допустимые внешние воздействия, то на основе такой модели можно автоматически синтезировать цифровую логическую схему, реализующую моделируемое устройство. Для использования САПР проектирования схем для построения электронной модели требуется описание схем, а следовательно формальный язык описания. В настоящее время разработан ряд таких языков, получивших в англоязычной научно-технической литературе название HDL (Hardware Description Language – язык описания оборудования). Самыми распространенными считаются два языка Verilog HDL и VHDL.

Для описания электронной модели элементов схемы был выбран язык HDL Verilog, так как он обладает рядом преимущест.

1. Во-первых, это один из самых распростарненных HDL языков. Поэтому имеется большое количество литературы для его изучения.
2. Во-вторых, этот язык С подобен, что облегчает его изучение в виду того, что обучение языку программирования С включено в программу студенческого курса.
3. В-третьих, Verilog поддерживается приложениями и САПР, которые требуются для написания тест-программы для цифрового устройства, что является целью данной бакалаврской работы.

Первым этапом создания тест-программы устройства служит описание логики всех ее внутренних элементов на языке Verilog. В качестве примера обратимся к рисунку 5. На нем изображен код, описывающий работу D-триггера (элементарной ячейки памяти), на языке Verilog HDL.

2.2 Мультиплатформенная среда проектирования схем Altera QUARTUS II

Программная среда QUARTUS II одна из основных разработок фирмы Altera. Она позволяет получить файл схематики устройства (top-module),

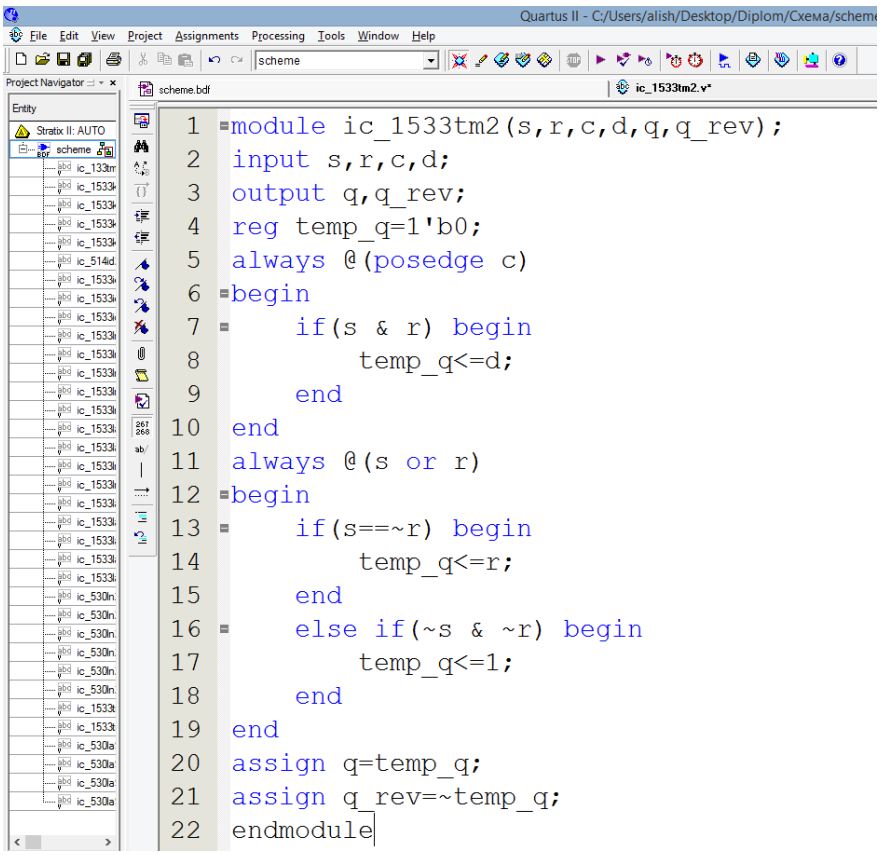


Рис. 5: Логика работы D-триггера

который в дальнейшем используется для написания тест-программы.

Второй этап создания тест-программы - формирование электронной схемы устройства. Электронная схема элемента 1533ТМ2 изображенная на рисунке 6 сохраняется в файле формата bdf и используется для построения электронной схемы всего устройства. Создав bdf-файл всей схемы и переведя его в формат "*.v"с помощью QUARTUS, мы получим исходный файл для загрузки в САПР SimTest, о которой далее.

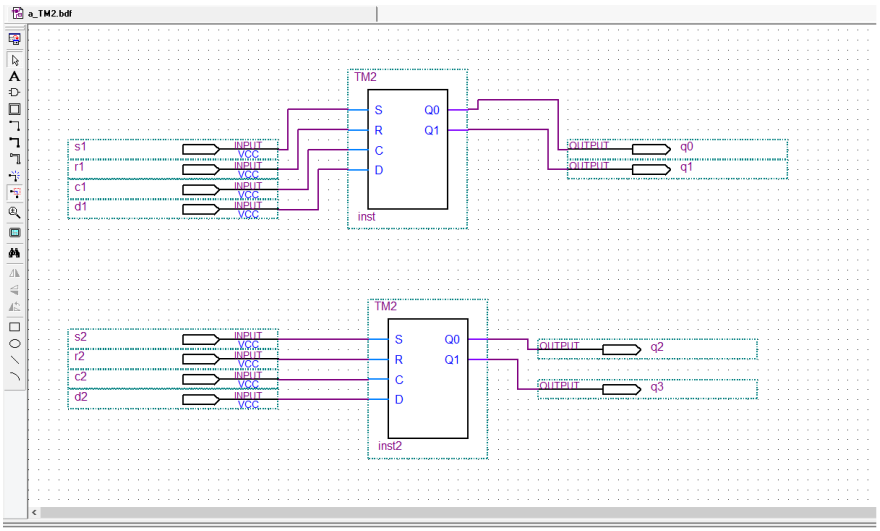


Рис. 6: Электронная схема 1533ТМ2

2.3 САПР SimTest

Современные цифровые устройства содержат сложные схемотехнические решения, основой которых являются контроллеры и процессоры. Построение тест-программ для таких устройств ручным способом является трудоемким процессом. С целью автоматизации процесса построения тестовых программ в СПбГУ на факультете ПМ-ПУ разработана система автоматизированного проектирования (САПР) тестов SimTest. Ее использование позволяет сократить время создания тест-программ в несколько раз.

Для разработки тестовых программ цифровых узлов и модулей формируются поведенческие модели их функционирования. Система SimTest принимает на входе структуру объекта контроля (ОК) и описания его компонентов, что является выходным файлом после работы в QUARTUS. На основе этой информации формируется полная модель устройства и генерируются последовательности входных воздействий и реакции ОК.

Построение автоматизированной тестовой программы с помощью САПР SimTest описано в параграфе 1.4.

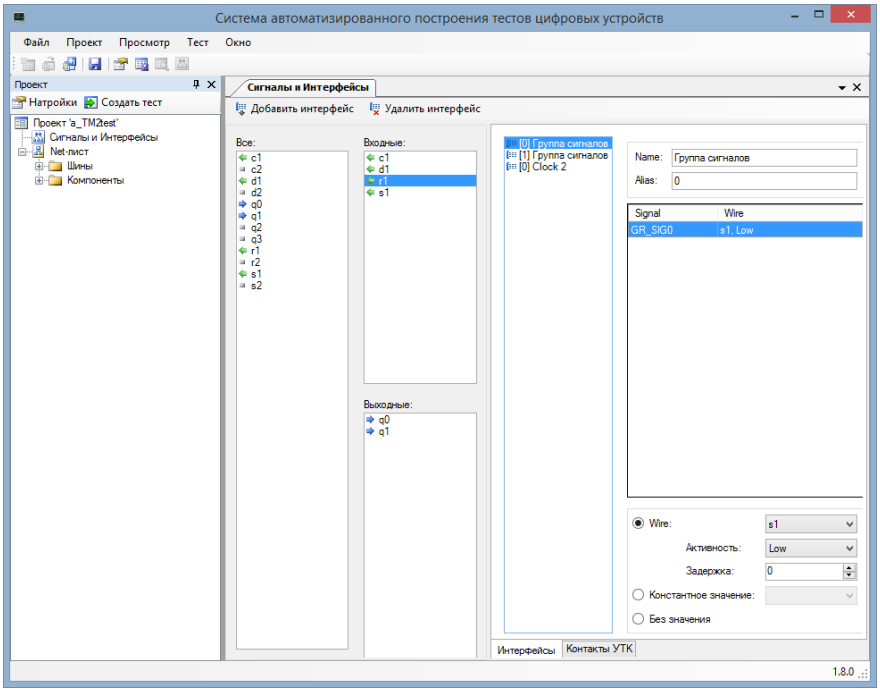


Рис. 7: Интерфейс САПР SimTest

Глава 3. Моделирование и создание тест-программы на примере объекта контроля

Цифровой объект контроля (микросхема) ИЗД.085.882Э3, используемый для моделирования, частично приведен на рисунке 8.

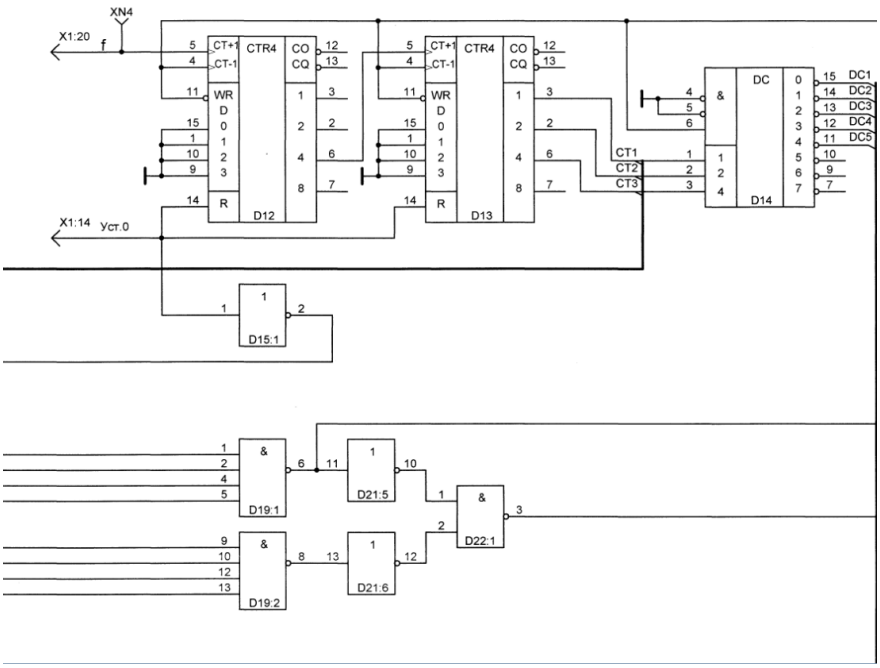


Рис. 8: Фрагмент микросхемы ИЗД.085.882Э3

Объект контроля состоит из цифровых и аналоговых элементов. Аналоговые элементы при построении программной модели устройства исключаются путем корректировки схемы. Для цифровых элементов создаются программные модели, которые используются при создании программной модели всего объекта контроля.

3.1 Описание элементов схемы

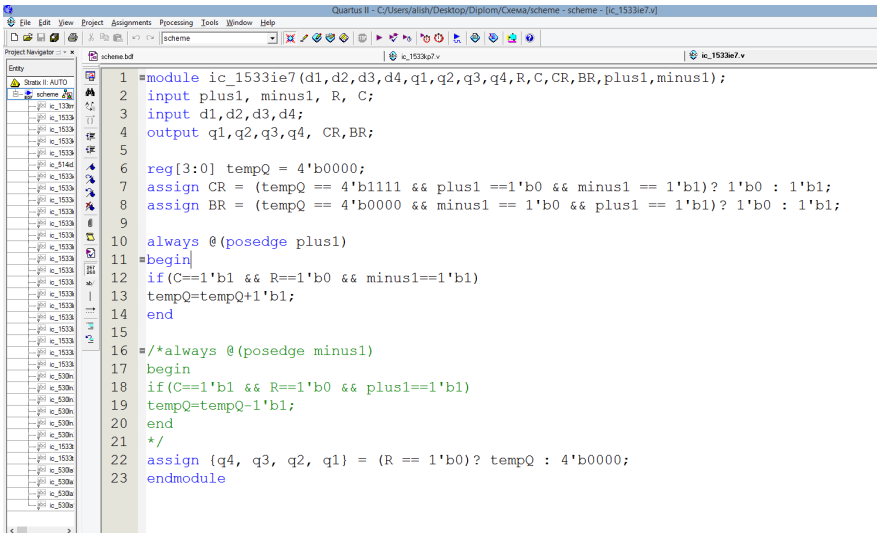
В данном параграфе будут рассмотрены самые большие элементы, входящие в состав моделируемой микросхемы.

3.1.1 КР1533ИЕ7

Микросхема КП1533ИЕ7 [7] представляет собой двоичный четырехразрядный реверсивный счетчик синхронного типа. Имеет 8 входов и 5 выходов. Из 8 входов: 4 информационных, 1 вход установки в состояние "логического нуля" (R), 2 входа прямого('+1') и обратного('-1') счета и вход стробирования предварительной записи(c). При подаче положительного импульса на вход R выходы счетчика устанавливаются в состояние "логического нуля" . При подачи отрицательного импульса напряжения на вход стробирования c осуществляется установка счетчика в необходимое состояние с помощью информационных входов. Для увеличения выходного значения на единицу, т.е. для прямого счета, требуется подавать положительные импульсы ко входу '+1', поддерживая уровень высокого напряжения на входе '-1'. Для "обратного" счета необходимо поменять местами выходы '+1' и '-1' .

После заполнении счетчика на выходе прямого переноса появляется отрицательный импульс.

Реализация функциональности микросхемы представлена на рисунке 9.



```
module ic_1533ie7(d1,d2,d3,d4,q1,q2,q3,q4,R,C,CR,BR,plus1,minus1);
input plus1, minus1, R, C;
input d1,d2,d3,d4;
output q1,q2,q3,q4, CR,BR;

reg[3:0] tempQ = 4'b0000;
assign CR = (tempQ == 4'b1111 && plus1 ==1'b0 && minus1 == 1'b1)? 1'b0 : 1'b1;
assign BR = (tempQ == 4'b0000 && minus1 == 1'b0 && plus1 == 1'b1)? 1'b0 : 1'b1;

always @(posedge plus1)
begin
if(C==1'b1 && R==1'b0 && minus1==1'b1)
tempQ=tempQ+1'b1;
end

/*always @(posedge minus1)
begin
if(C==1'b1 && R==1'b0 && plus1==1'b1)
tempQ=tempQ-1'b1;
end
*/
assign {q4, q3, q2, q1} = (R == 1'b0)? tempQ : 4'b0000;
endmodule
```

Рис. 9: Фрагмент кода схемы КР1533ИЕ7 на языке Verilog

3.1.2 КР1533КП7

Микросхема КР1533КП7 [13] представляет собой селектор-мультиплексор из 8 в 1. Имеет 8 информационных входов, 1 вход стробирования и 3 управляющих входа. В соответствии с установленными на управляющих входах сигналами пропускает сигнал от одного из восьми информационных входов на выходной.

При подаче высоко уровня напряжения на вход стробирования устанавливает выход в состояние "логического нуля" .

Реализация функциональности микросхемы представлена на рисунке 10.

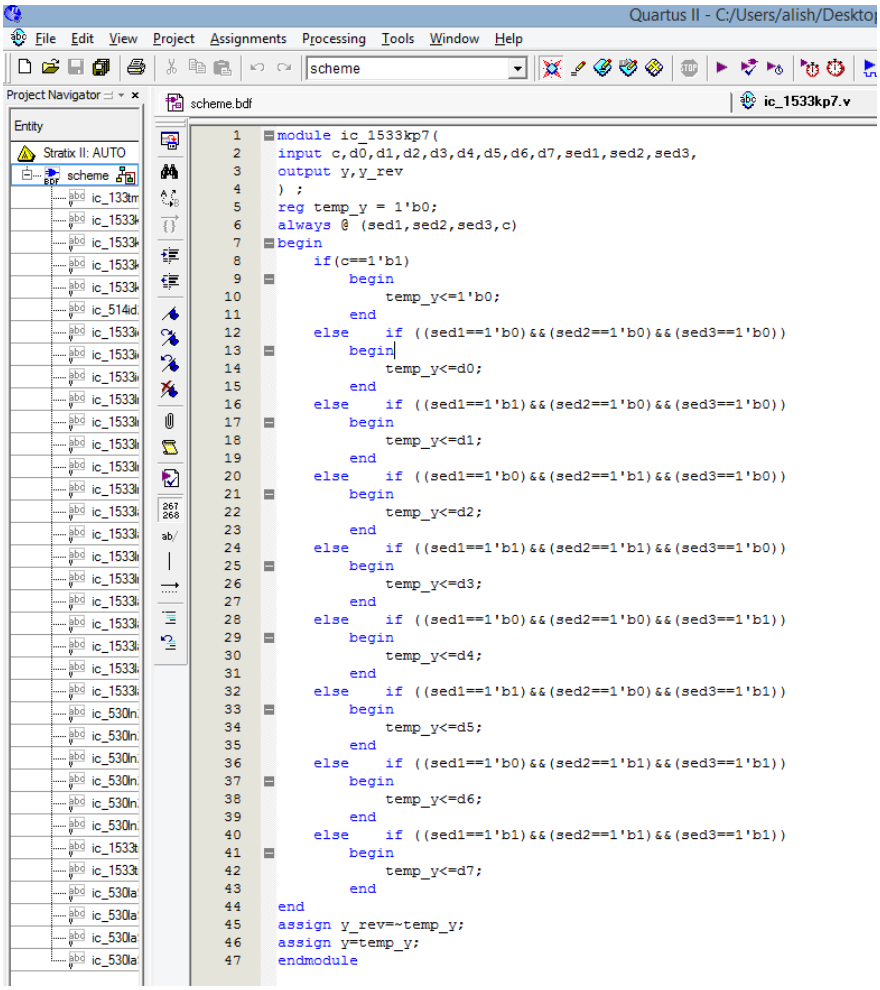


Рис. 10: Фрагмент кода схемы КР1533КП7 на языке Verilog

3.1.3 КР1533ИД7

Микросхема КР1533ИД7 [14] представляет собой дешифратор или демультиплексор 3 на 8. При использовании микросхемы в качестве дешифратора входы $D1, D2, D4$ играют роль информационных, а $C1, C2, C3$ - стробирующих.

В случае использования микросхемы как демультиплексера входы $D1, D2, D4$ разрешающие. В соответствии с поданными на них сигналами входной информационный сигнал проходит к определенному выходу. В данном случае $C1$ используется как информационный, а $C2, C3$ - в качестве стробируемых сигналов.

Реализация функциональности микросхемы представлена на рисунке 11.

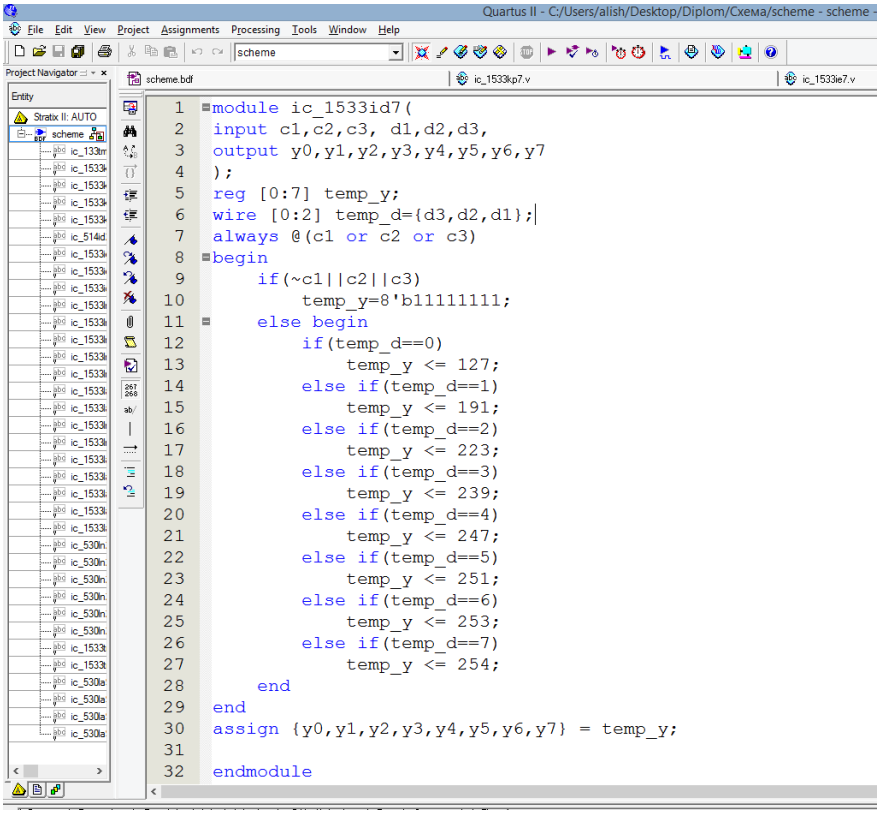


Рис. 11: Код схемы КР1533ИД7 на языке Verilog

3.1.4 КР514ИД2

Микросхема КР514ИД2 [15] представляет собой дешифратор 4-разрядного двоичного кода в сигналы 7-сегментного кода. Используются для управления индикаторами на основе диодных структур.

При подаче высоко напряжения на разрешающий вход происходит дешифрация сигналов. В случае установки низкого уровня напряжения на разрешающем входе все выходы переходят в состояние "логического нуля" .

Реализация функциональности микросхемы представлена на рисунке 12.

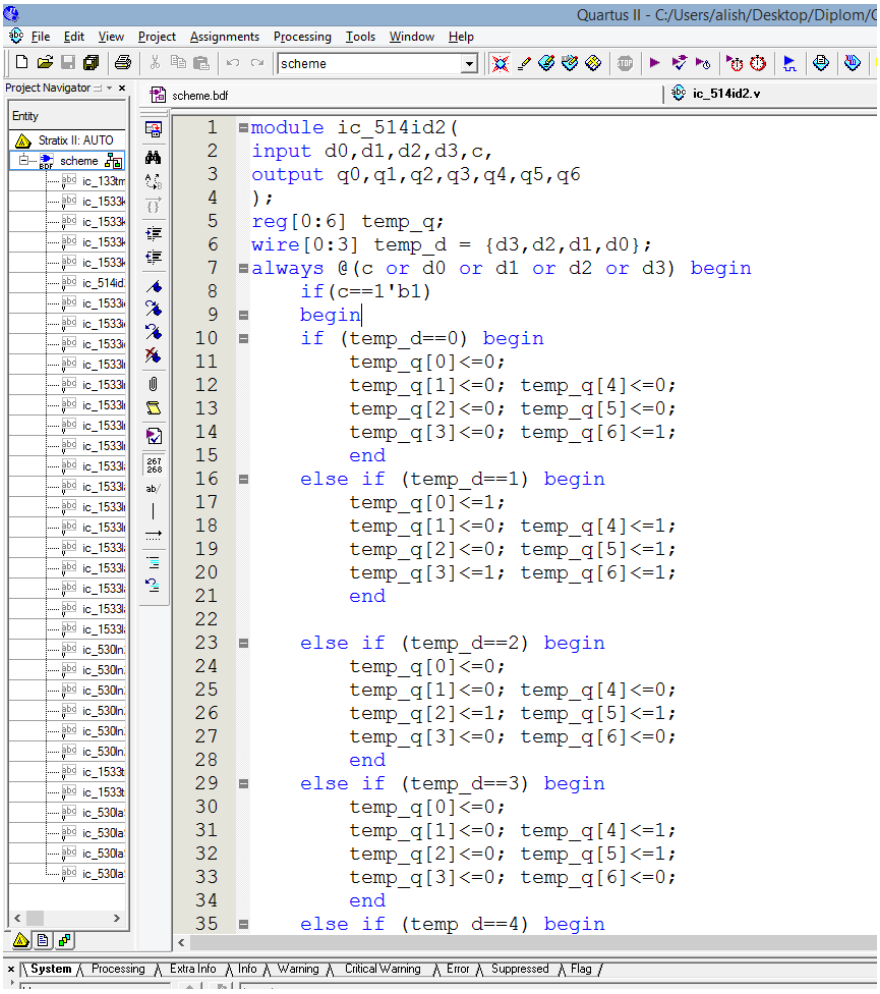


Рис. 12: Фрагмент кода схемы КР514ИД2 на языке Verilog

3.2 Работа в QUARTUS II. Создание электронной модели микросхемы

При создании электронной модели микросхемы требуется создать электронные модели всех ее компонентов и занести их в базу данных системы (рис 12).

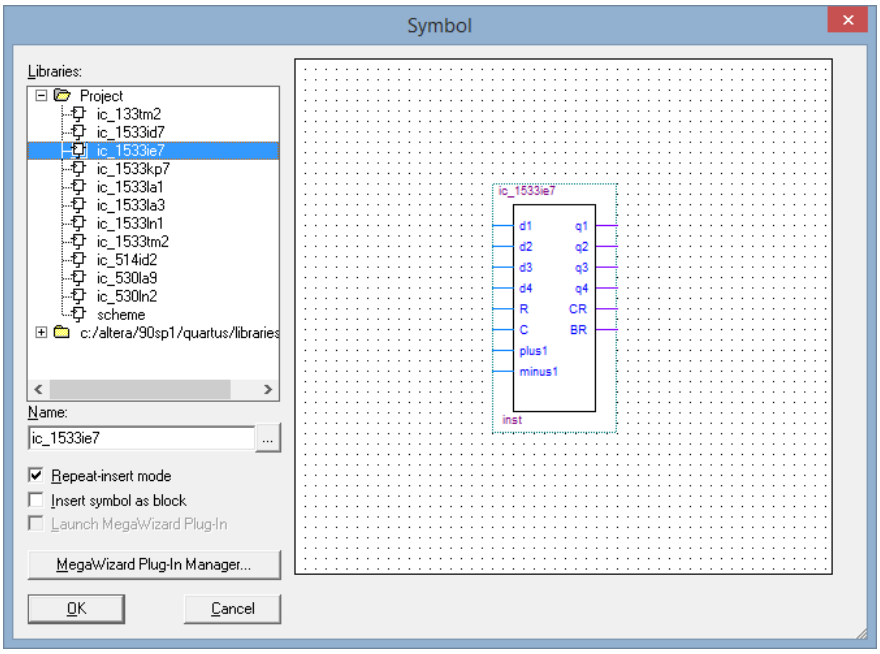


Рис. 13: Визуальное представление элементов в QUARTUS II

Далее, используя все созданные компоненты, составим электронную модель всей схемы. Часть данной модели изображена на рисунке 14.

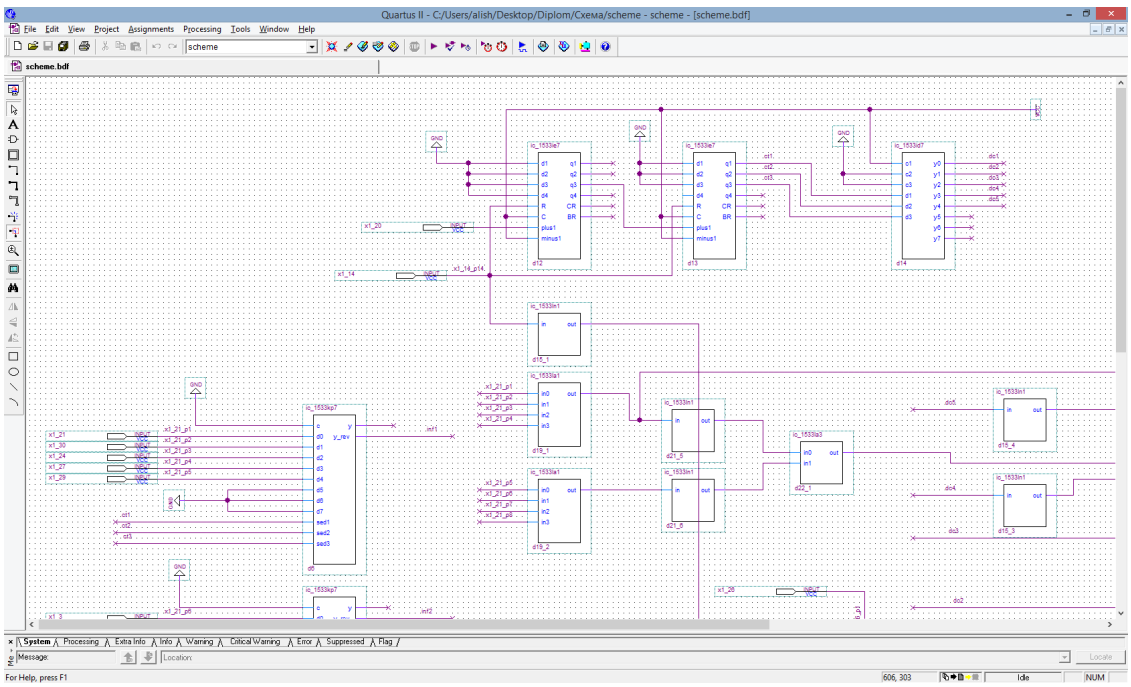


Рис. 14: Часть электронной модели рассматриваемой микросхемы

После создания электронной схемы микросхемы, получим файл формата '*.v', который подается на вход в САПР SimTest.

3.3 Работа в САПР SimTest

На последнем этапе создания тест-программы используется САПР SimTest. Производится привязка логических интерфейсов из базы САПР SimTest к краевым разъемам модели объекта контроля. Осуществляется формирование входных воздействий, описанных в скрипте. В результате работы были получены следующие результаты.

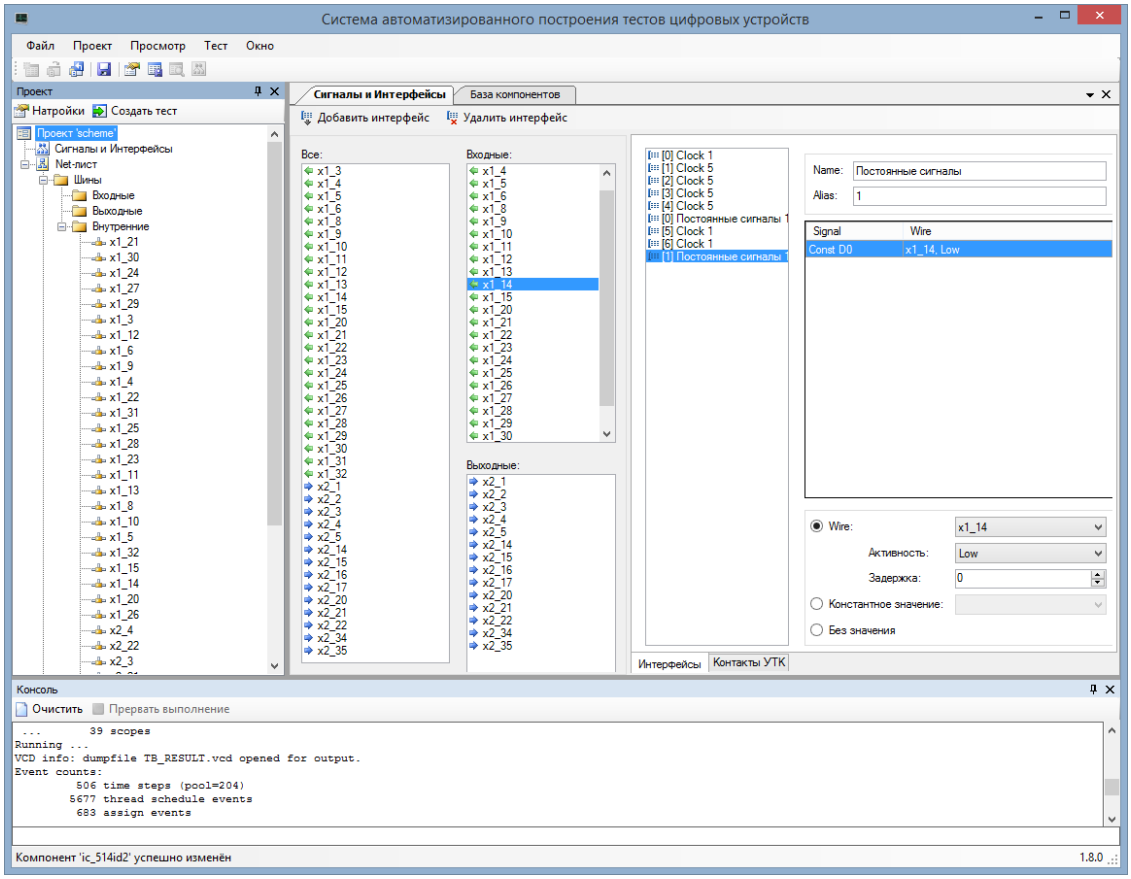


Рис. 15: Привязка логических интерфейсов

Общее покрытие составило порядка 85% (рис 17), что объясняется тем, что не все входы в схеме задействованы. Некоторые входы подключены к источникам постоянного напряжения.

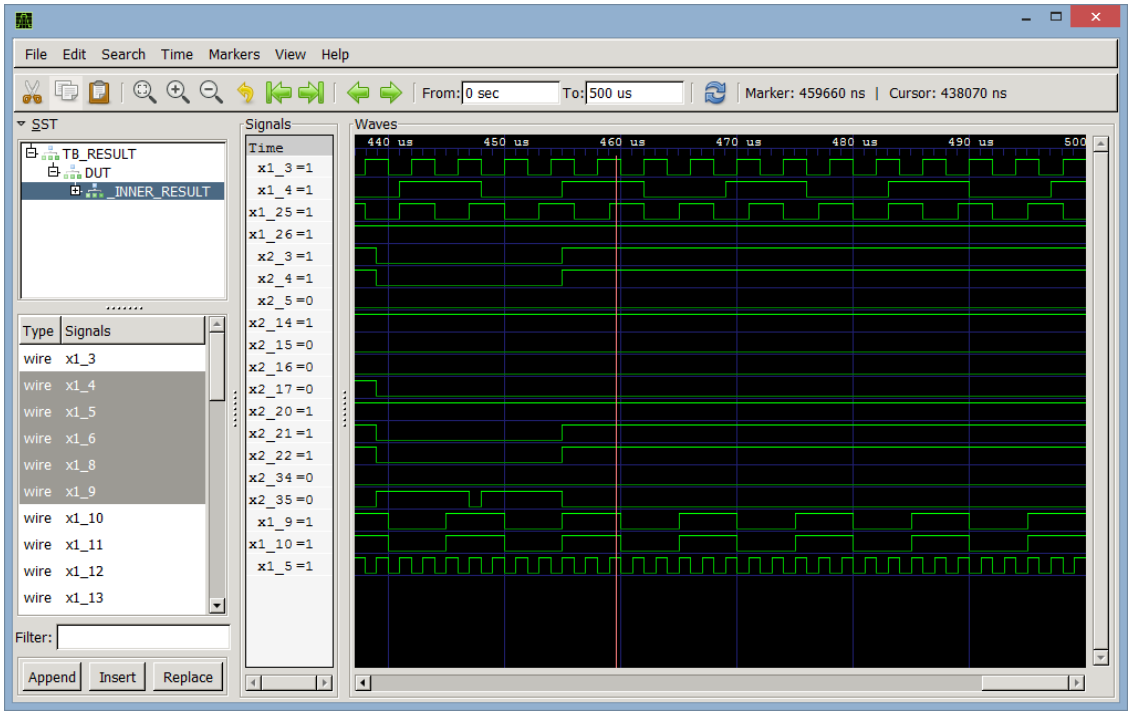


Рис. 16: Фрагмент временной диаграммы работы компонентов

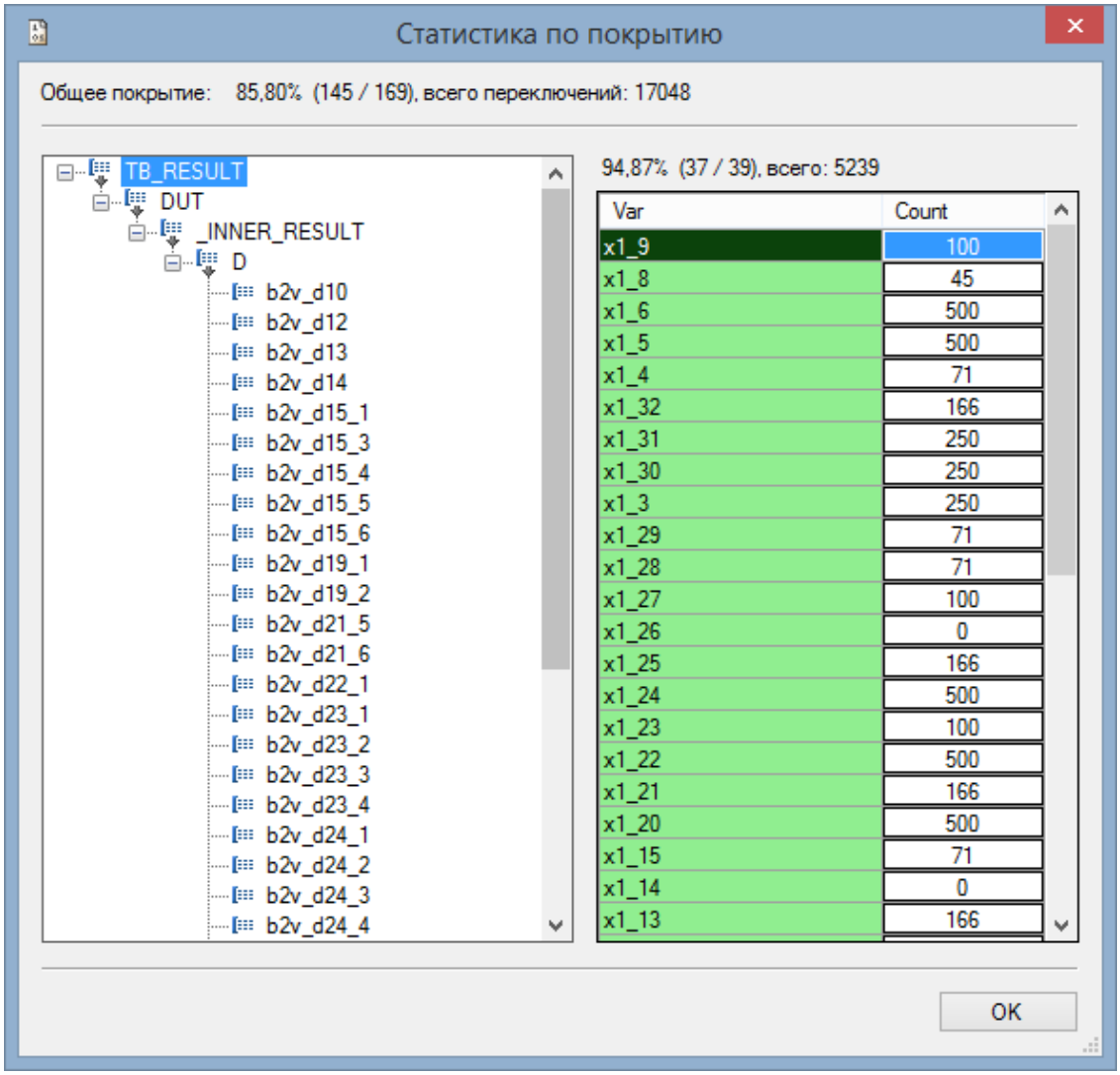


Рис. 17: Покрытие тест-программой

Заключение.

Главным итогом работы можно считать освоение метода построения тестовых программ для цифровой аппаратуры. Для его освоения потребовалось дополнительно освоить следующие технологии:

1. Язык программирования электронных устройств Verilog HDL, который использовался для описания логики компонент схемы.
2. Мультиплатформенная среда проектирования Altera QUARTUS II, используемая для создания электронной модели схемы.
3. САПР SimTest, используемую для автоматизации реакции модели схемы на тестовую последовательность.

Освоение метода было продемонстрировано на примере объекта контроля ИЗД.085.882Э3. Для создания теста для объекта контроля были проделаны следующие шаги:

1. Разработаны программные модели цифровых устройств КР1533ИЕ7, КР1533КП7, КР1533ИД7, КР514ИД2, КР1533ТМ2 и др. на языке Verilog HDL.
2. С помощью среды проектирования Altera QUARTUS II получена программная модель цифрового объекта контроля ИЗД.085.882Э3.
3. Произведено соответствие логических интерфейсов «SimTest» краевым разъемам модели объекта контроля. Подобраны необходимые проверочные последовтельности входных сигналов. Уровень тестового покрытия, полученный в результате моделирования тест-программы для рассматриваемого объекта контроля, оказался приемлимым.

Список литературы

[1] Гришкин В.М., Лопаткин Г.С., Михайлов А.Н., Овсянников Д.А. Интерфейсный метод построения моделей входных воздействий для тестирования электронных цифровых модулей // Вопросы радиоэлектроники. 2013. Т. 1. № 1. С. 80-89

[2] Мельник В., Гришкин В., Михайлов А., Овсянников Д. Методика разработки тест-программ контроля и диагностики цифровых устройств с использованием САПР SimTest // Электроника: Наука, технология, бизнес. 2013. № S (128). С. 118-124.

[3] Melnik V.I., Mikhailov A.N., Grishkin V.M., Ovsyannikov D.A., Yelaev Y.V. Methods of modeling od the test inputs for analysis the digital devices // 2014 International conference on computer technologies in physical and engineering applications (ICCTPEA). Editor: E. I. Veremey. Санкт-Петербургский государственный университет; IEEE (IEEE Catalog number CFP14BDA-USB). 2014. С. 112-113.

[4] Melnik V.I., Mikhailov A.N., Grishkin V.M., Ovsyannikov D.A., Yelaev Y.V. Modeling methods of the test inputs for analysis the digital devices // 2014 2nd International Conference on Emission Electronics, ICEE 2014 Joined with 10th International Vacuum Electron Sources Conference, IVESC 2014, International Conference on Computer Technologies in Physical and Engineering Applications, ICCTPEA 2014, 20th International Workshop on Beam Dynamics and Optimization, BDO 2014 - Proceedings. 2014

[5] Jayapradha V., Ravi S., Kamalakkannan R., Selvakumar S. Test coverage analysis of memory cluster testing using JTAG // (2014) International Journal of Applied Engineering Research, 9 (22), pp. 11861-11870.

[6] Yin X.H., Xu C.F. On a method of getting test data for boundary scan interconnection test in multiple scan chains // (2014) Advanced Materials Research, 986-987, pp. 1531-1535.

[7] Shashidhara H.B., Yellampalii S., Goudanavar V. Board level JTAG/boundary scan test solution // (2014) Proceedings of International Conference on Circuits, Communication, Control and Computing, I4C 2014, art. no. 7057760, pp. 73-76.

[8] Renbi A., Delsing J. Contactless Testing of Circuit Interconnects // (2015) Journal of Electronic Testing: Theory and Applications (JETTA), 31 (3), pp. 229-253.

[9] Bhowmik B., Deka J.K., Biswas S. Beyond test pattern generation: Coverage analysis // (Conference Paper) 2015 International Conference on Industrial

Instrumentation and Control, ICIC 2015 6 July 2015, Article number 7151009, Pages 1620-1625.

[10] Bhar A., Chattopadhyay S., Sengupta I., Kapur R. Small Test Set Generation with High Diagnosability // Journal of Circuits, Systems and Computers 2015 DOI: 10.1142/S0218126616500249

[11] Paltnitkar S. Verilog HDL: A Guide to Digital Design and Synthesis, Second Edition. Prentice Hall PTR, 2003. 496 c.

[12] 1533IE7 datasheet. http://www.datasheet-pdf.ru/1533/1533pdf/1533IE7.pdf

[13] 1533KP7 datasheet. http://www.datasheet-pdf.ru/1533/1533pdf/1533KP7.pdf

[14] 1533ID7 datasheet. http://www.datasheet-pdf.ru/1533/1533pdf/1533ID7.pdf

[15] 514ID2 datasheet. http://ic-info.ru/upload/iblock/5a1/514%D0%98%D0%942.pdf