

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА ТЕОРИИ СИСТЕМ УПРАВЛЕНИЯ ЭЛЕКТРОФИЗИЧЕСКОЙ АППАРАТУРОЙ

Гусев Олег Александрович

Выпускная квалификационная работа бакалавра

**Эмуляция и тестовый контроль цифровых
устройств**

Направление 010900

Прикладные математика, физика и процессы управления

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

Рецензент,
кандидат тех. наук,
доцент
Гришкин В. М.

Санкт-Петербург

2017

Оглавление

Введение.....	3
Постановка задачи.....	4
Глава 1. Развитие схемотехники.....	5
1.1 История	5
1.2 Типы тестирования.	7
Глава 2. Построение программной модели	10
2.1 Язык описания логики элементов Verilog HDL.....	10
2.2 Проектирование цифрового устройства с помощью мультиплатформенной среды проектирования Altera Quartus II	10
2.3 Система автоматизированного проектирования «SimTest»	11
Глава 3. Составление программной модели объекта контроля.....	13
3.1 Моделирование компонентов	13
3.2 Электронная модель схемы.....	18
Глава 4. Тестирование.....	20
4.1 Тестирование логики элементов.....	20
4.2 Тестирование Субблок 714ПЭЗ	23
Заключение	28
Список литературы	29

Введение

После создания первой микросхемы в 1960 году радиотехника и цифровая техника начали входить во все сферы человеческой деятельности. Сейчас сложно представить жизнь без телефона, компьютера, микроволновой печи — все эти устройства включают в себя микросхемы, резисторы, транзисторы. Однако, как и в любой сфере серийного производства, при изготовлении цифровых устройств присутствует брак, в связи с этим для их надежного функционирования необходима техническая диагностика и контроль каждого компонента.

Поиск неисправных элементов еще полностью не автоматизирован и является актуальной проблемой на сегодняшний день. Использование систем автоматизированного проектирования (САПР) тестов таких как «SimTest» (САПР, разработанный в СПбГУ [1,2]) и «Yastek», а также аппаратно-программного комплекса тестового контроля и диагностики УТК-512 позволяют существенно ускорить время проверки работоспособности цифровых устройств.

Цифровым устройством называют техническое приспособление, которое состоит из набора электронных компонентов и, основываясь на алгоритмах их работы, получает и обрабатывает входящую информацию. Под тестом устройства понимают набор входных и соответствующих им выходных данных, проверяющих компоненты устройства и связи между ними. После создания тест загружается в устройство тестового контроля и диагностики и используется для отбраковывания неисправных объектов контроля.

В работе представлен алгоритм построения программной модели на примере реальной схемы Субблок 714ПЭЗ в среде ALTERA QUARTUS II и создание тестовой программы с использованием САПР «SimTest».

Постановка задачи

Задачей этой работы является создание теста для контроля работоспособности цифрового устройства.

В этом исследовании функциональность схемы устройства является неизвестной и считается полностью правильной, в нашем же распоряжении находятся лишь контакты входов и выходов, названия элементов, входящих в ее состав, а также связи между ними. Таким образом проверяется работоспособность элементов, находящихся на данной плате, и корректность связей между ним. В процессе выполнения работы необходимо выполнить следующие пункты:

1. Разработать программную модель каждого элемента устройства на языке описания аппаратных средств. В качестве среды разработки будет использоваться ALTERA QUARTUS II.
2. Осуществить создание программной модели самого цифрового устройства посредством объединения его отдельных элементов.
3. Создать тест для контроля работоспособности объекта контроля. Разработка теста будет проводиться с использованием САПР «SimTest».

Глава 1. Развитие схемотехники

1.1 История

Схемотехника — направление в науке, которое занимается созданием и откладкой устройств, в состав которых обычно входят электронные схемы.

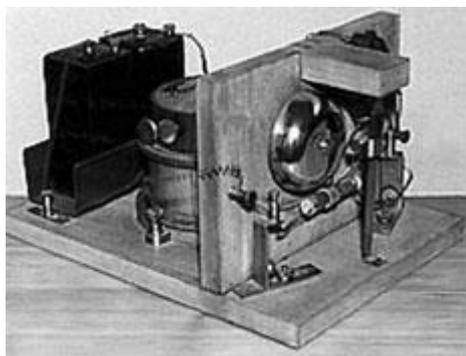


Рис. 1: Радио

7 мая 1895г. можно считать отправной точкой развития схемотехники. В этот день Попов А.С. впервые представил радио (Рис. 1).

Сейчас можно выделить следующие этапы развития:

1.1.1 Первое поколение электроники. Ламповая электроника.

Началом развития ламповой электроники принято считать создание английским инженером Д.А.Флемингом первого диода в 1904 г. (Рис. 2).

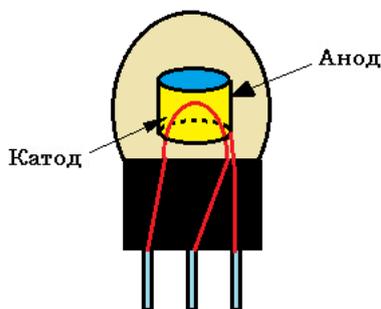


Рис. 2: Диод

Принцип ее работы был основан на трудах Т.А. Эдисона, который описал явление термодинамической эмиссии, и состоял в том, что ток мог проходить лишь в одном направлении. На практике был применен в качестве детектора радиоволн. Спустя несколько лет в 1906 г. Ли де

Форест поместил сетку между катодом и анодом, основная особенность данной конструкции: при подаче положительного напряжения на сетку электроны ускоряются и большая их часть достигает анода, при подаче отрицательного напряжения электроны отталкиваются от сетки и возвращаются к катоду. Это изобретение получило название трехэлектродная лампа или триод. Используется в качестве усилителя и генерации электрических колебаний различной частоты. Позже появились

четырёхэлектродные (тетроды) и пятиэлектродные (пентоды) лампы, их задача заключалась в корректировании характеристик ламп, обычно частотных [3]. В это же время инженеры пытались сделать лампы более миниатюрными, улучшить их качества, снизить потребление на «разогрев» и добиться лучшего охлаждения. Сейчас они применяются в мощных высокочастотных передатчиках и в высококлассной аудиотехнике.

1.1.2 Второе поколение электроники. Полупроводниковая электроника.

Начало эры полупроводников было положено созданием первого транзистора (Рис. 3) в 1947 г. Д. Бардиным, У. Баттейном и У. Шокли.



Рис. 3: Транзистор

Транзистор — полупроводниковый прибор, который усиливает, преобразует и генерирует электрические колебания [3-4]. Данное новшество, благодаря возможности работать при низких напряжениях, быстро пришло на смену ламповой электронике и открыло возможность существенно уменьшить электронные устройства.

В 1962 г. на основе полупроводника был создан светодиод.

1.1.3 Третье поколение электроники. Интегральные микросхемы.

Первоначальное уменьшение размера плат методом микромодулей: электронные модули печатались на тонких керамических пластинах, которые затем спаивались вместе стопкой, таким образом получалась нужная

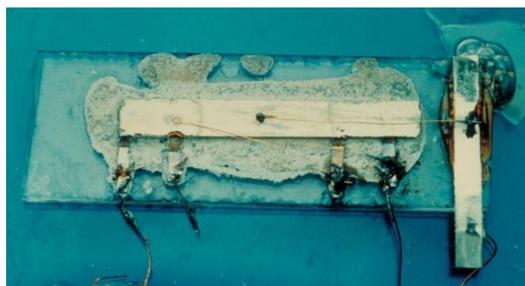


Рис. 4: Первая интегральная схема (11.1x1.6 мм)

микросхема. Джек Килби счел этот метод слишком сложным и решил изготавливать сразу несколько полупроводниковых элементов на одной пластине. Первый микромодуль был выпущен в 1959г. Основной особенностью перехода на

интегральные микросхемы является не только уменьшение массы и размера плат, но и увеличение быстродействия, повышение надежности и уменьшение потребляемой мощности. Так в 1966 г. начался выпуск средних интегральных схем, число элементов в одном кристалле колебалось от 100 до 1000, а в 1975 г. в один кристалл стали уместать более 10000 элементов, такие микросхемы называют сверхбольшими [3-4].

Стоит упомянуть о разработке процессоров. В их основы так же входят полупроводниковые элементы. В настоящий момент использование технологии кремния подходит к пределу. Это связано с тем, что уменьшать размеры транзисторов становится все сложнее, а увеличение их скорости переключения влечет за собой экспоненциальное увеличение температуры.

В скором будущем на смену кремния должен прийти графен. Транзистор, на основе которого заменяют три кремниевых.

1.2 Типы тестирования

На данном этапе развития схемотехники выделяют следующие подходы к тестированию радиоэлектронных устройств:

1.2.1 Визуальный контроль.

Контроль качества визуальным методом издавна является одним из способов отбраковки изделий. Вначале люди с помощью различной увеличительной аппаратуры занимались поиском царапин, сколов, изменением размеров изделий — всех отклонений от нормы [5].

По мере уменьшения компонентов электронных устройств человек все больше стал заменяться компьютерами, это связано с качеством и временем обработки данных. В настоящий момент многие производства полностью перешли на визуальный автоматизированный контроль, в котором компьютер сам «просматривает» и отбраковывает электронные устройства и его

компоненты. Данный метод является основополагающим на ранних стадиях производства.

1.2.2 Внутрисхемное тестирование

Метод основан на проверке отдельных частей печатной платы посредством адаптера, называемых контактными иглками. Процесс проверки осуществляется посредством подключения адаптера к отдельной части и подачей/считкой необходимых сигналов [6-7]. Поскольку в прошлом корпуса микросхем были крупные, а сами дорожки менее миниатюрные, любой вывод был доступен для иглолок адаптера, и проблем с доступом не существовало.

В настоящий момент идет процесс миниатюризации, в условиях высокой плотности элементов и переноса части цепей во внутренние слои, оборудование для процесса внутрисхемного тестирования приходится усовершенствовать, что увеличивает его стоимость. Однако это не сокращает область его применения. Основной особенностью такого метода является проверка отдельных компонентов устройства без их демонтажа, а также способность проверять объект не подавая питание на все устройство, что уменьшает риск испортить тестируемый объект.

1.2.3 Периферийное сканирование

Впервые периферийное сканирование было предложено в 1985 году компанией по производству электроники — JTAG (отсюда и появилось название JTAG сканирования). Такой тип тестирования применяется для печатных плат с микросхемами стандарта IEEE 1149 (тестовая логика уже включена в само устройство). На платах элементы соединены в один или несколько каналов (зависит от совместимости компонентов) [8-11].

Основными результатами JTAG сканирования является информация о наличии повреждения дорожек и отпада элементов, коротких замыканий, нерабочих микросхем.

Тестовый контроль происходит по следующему алгоритму:

- подключение объекта тестирования;
- подача на вход последовательности необходимых значений, зависящих от времени;
- анализ специальным программным обеспечением полученных на выходе результатов;
- вынесения результатов проверки.

Здесь, как и во всех видах тестирования, имеются свои плюсы и минусы: периферийное сканирование широко применяется как для проверки, так и для ремонта цифровой техники, помогая более точно указать место поломки, тестирование не зависит от топологии, расположения и наличия тестовых точек и т.д., используются лишь специальные разъемы. Однако для хорошего тестового покрытия необходима поддержка всех компонентов стандарта IEEE 1149, что в свою очередь влечет к увеличению размера и стоимости самого устройства, а также к необходимости проверки самих «тестирующих» дополнений.

1.2.4 Функциональное тестирование

Тестирование, основанное на проверке функциональности и спецификационных параметров целого устройства или его компонентов [1, 11-13]. В ходе тестирования имитируется фактическая работа схемы, а после, на основе входящих в ее состав частей, разрабатывается тест диагностики неполадок. Основные трудности этого вида тестирования связаны с невозможностью на данный момент построения теста автоматически, однако это компенсируется относительно недорогой стоимостью оборудования и возможностью использования практически для всех цифровых устройств.

Глава 2. Построение программной модели

2.1 Язык описания логики элементов Verilog HDL

Verilog HDL — язык описания электронной аппаратуры. Написанная на Verilog HDL программа является моделью определенного устройства. Для автоматического синтеза цифровой схемы необходимо чтобы модель повторяла ее функциональность и реакцию на внешние воздействия. На данный момент существует множество языков, используемых для синтеза цифровых устройств, среди них можно выделить два основных: VHDL и Verilog HDL (HDL - Hardware Description Language) [14-16].

В данной работе использовался язык Verilog. Такой выбор в первую очередь связан с:

- широкой распространенностью. Литература имеется в свободном доступе и на многих языках;
- простотой в освоении. Схож с популярным языком программирования C;
- возможностью использования оборудования и программ, предоставляемых кафедрой.

2.2 Проектирование цифрового устройства с помощью мультиплатформенной среды проектирования Altera Quartus II

Цифровое устройство — набор электронных модулей с определенным алгоритмом работы. Таким образом, работа устройства зависит от входящих в него модулей и связей между ними, а также последовательности входящих сигналов.

Проектирование начинается с создания программной модели каждого компонента. В работе была использована программа Quartus II компании Altera. Создание файла электронной схемы устройства осуществляется по следующей схеме [15]:

1. Поиск необходимой информации по функциональности компонента схемы.
2. Описание функциональности модуля на языке Verilog.
3. Создание программной модели компонента (при необходимости проверки правильности описания есть возможность провести тестирование программной модели этого компонента).
4. Объединение всех компонентов цифрового устройства, путем указания связей и добавления входов/выходов.

Построение программной модели по данному плану более подробно будет разобрано в Главе 3.

2.3 Система автоматизированного проектирования «SimTest»

Построение тестовой программы для современного цифрового устройства невозможно без САПР. В СПбГУ на факультете ПМ-ПУ был разработан свой САПР — «SimTest». «SimTest» используется для автоматического осуществления моделирования входных и выходных сигналов программной модели реального устройства.

Построение тестовой программы осуществляется по следующему плану:

1. Создание проекта и добавление всех компонентов схемы в среду «SimTest».
2. Распределение входных, выходных и двунаправленных краевых сигналов.
3. Выбор и добавление интерфейса и формирование входных воздействий.
4. Анализ полученных данных. При необходимости вернуться в пункт 2.

Рассмотрим подробнее пункт 4. После моделирования система выдает файл, в котором показана реакция устройства на входные воздействия. Так же в ней представлен процент покрытия, вычисленный по формуле: $P = (N_{act}/N) * 100\%$, где N_{act} — число сигнальных линий устройства,

активированных в ходе выполнения теста, N — число линий сигналов устройства [2]. При разработке пытаются достичь наибольшего покрытия, часто называемого качеством теста. Однако даже 100% покрытие еще не является основным критерием окончания построения тестовой программы. Это связано в первую очередь с тем, что многие компоненты могут содержать какие-либо данные в своей памяти или находиться в определенном, неизвестном нам, состоянии. Эти факторы заставляют строить тесты для проверки функционирования и таких сложных компонентов. Решение об окончании разработки тестовой программы остается за оператором, таким образом, при необходимости, после анализа полученных данных он может вернуться в пункт 3 для формирования других входных последовательностей, а возможно и изменения интерфейса.

Глава 3. Составление программной модели объекта контроля

Устройство Субблок 714ПЭЗ включает в себя 84 микросхемы, 67 входных, 57 выходных, 24 контрольных контактов (используются при тестировании реального устройства на установке тестового контроля).

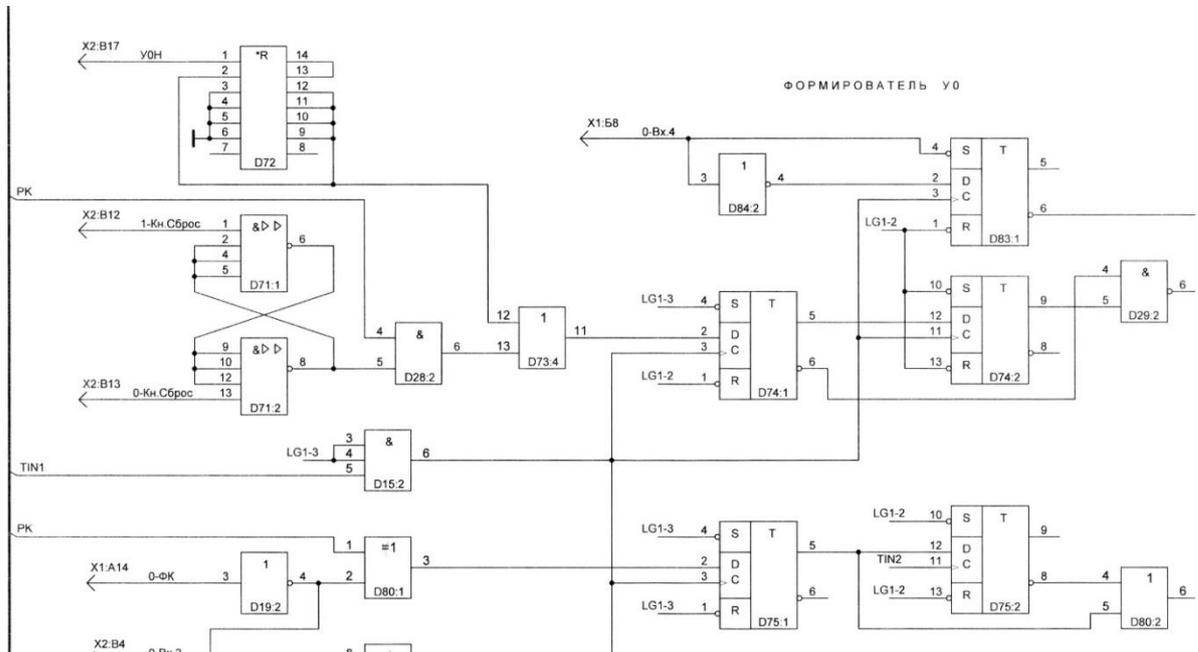


Рис.5: Фрагмент схемы Субблок 714ПЭЗ

Следуя плану, необходимо начать с разработки программной модели каждой микросхемы устройства, что и будет сделано далее.

3.1 Моделирование компонентов

Рассмотрим несколько объектов моделирования с различной логикой работы.

3.1.1 КР530ЛА1

Микросхема КР530ЛА1 [17] включает два одинаковых логических элемента с 4 входами и 1 выходом. Принцип работы можно описать функцией: $Y = \overline{D1} + \overline{D2} + \overline{D3} + \overline{D4}$. Где $D1, D2, D3, D4$ — входы, и Y — выход логических элементов. Н — высокий уровень сигнала, L — низкий уровень сигнала, X — произвольный уровень сигнала.

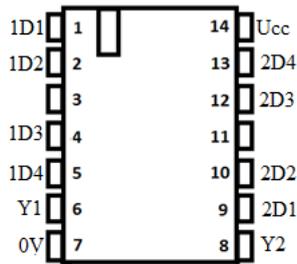


Рис. 6: Расположение выводов

D1	D2	D3	D4	Y
H	H	H	H	L
L	X	X	X	H
X	L	X	X	H
X	X	L	X	H
X	X	X	L	H

Рис. 7: Таблица истинности

01	1D1	Вход
02	1D2	Вход
03		
04	1D3	Вход
05	1D4	Вход
06	Y1	Выход
07	0V	Общий вывод
08	Y2	Выход
09	2D1	Вход
10	2D2	Вход
11		
12	2D3	Вход
13	2D4	Вход
14	Ucc	Напряжение питания

Рис. 8: Назначение выводов

После поиска необходимой информации по работе компонента перейдем к реализации функциональной модели (Рис. 9), а после созданию электронной модели (Рис. 10).

```

1  `timescale 10 ns/ 1 ps
2  module o1_ic_530la1 (d1, d2, d3, d4, q);
3  input d1;
4  input d2;
5  input d3;
6  input d4;
7  output q;
8  assign q = ~(d1&d2&d3&d4);
9  endmodule

```

Рис. 9: Код KP530LA1

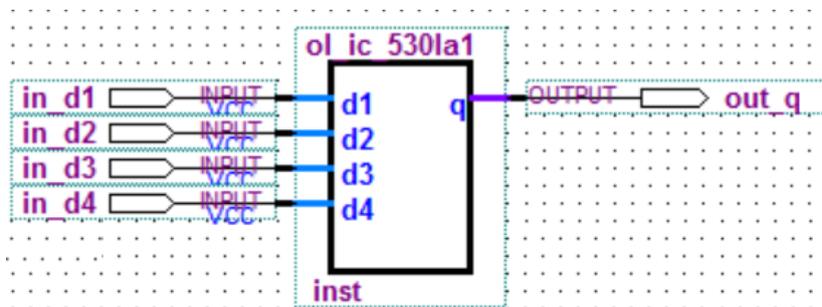


Рис. 10: Программная модель элемента KP530LA1

3.1.2 КР530ТМ2

Микросхема КР530ТМ2 [18] имеет память, состоит из двух срабатывающих по положительному фронту D-триггеров, независимых друг от друга. D-триггер имеет 4 входа: вход сброса (r), вход установки (s), тактовый вход (c) и информационный (d), и 2 выхода q и $q_$. При подаче на вход сброса низкого (высокого) уровня сигнала, а на вход установки высокого (низкого) независимо от оставшихся входов получим низкий (высокий) сигнал на выходе q и высокий (низкий) на $q_$. При подаче низких сигналов на входы r и s и любых сигналов на (c) и (d) значения выходов не определены. Когда на поданы высокие уровни сигнала, смена выходных сигналов будет происходить после подачи положительного фронта сигнала на тактовый вход, в таком случае выход q приобретет значение входа d , а $q_$ противоположный. При подаче высоких сигналов на r и s и низкого на c , независимо от входа d , выходы приобретут сигналы предыдущего состояния.

```
1  `timescale 10 ns/ 1 ps
2  module ol_ic_530tm2 (r,s,c,d,q,q_);
3  input r;
4  input s;
5  input c;
6  input d;
7  output q;
8  output q_;
9  reg q;
10 reg q_;
11 always @( posedge s)
12     begin
13         if (r==1'b0)
14             begin
15                 q=1'b0;
16                 q_=1'b1;
17             end
18     end
19 always @( negedge s)
20     begin
21         if (r==1'b1)
22             begin
```

Рис. 11: Фрагмент кода КР530ТМ2

3.1.3 KP133ID3

Микросхема KP133ID3 [19] представляет собой дешифратор с 2 стробирующими входами управления (C1, C2), 4 информационными входами (D1, D2, D3, D4) и 16 выходами (Y0, Y1, Y2, ... Y15). При подаче хотя бы на один стробирующий вход высокого уровня сигнала, независимо от сигналов на информационных входах, все выходы приобретут высокий уровень. При подаче на C1, C2 низкого уровня, результаты выходов будут зависеть от комбинации сигналов на информационных входах (подробно на Рис. 12.) Н — высокий уровень сигнала, L — низкий уровень сигнала.

Вход						Выход																
C1	C2	D1	D2	D3	D4	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15	
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L

Рис. 12: Принцип работы KP133ID3

```

1  `timescale 10 ns/ 1 ps
2  module o1_ic_133id3(c1,c2,d4,d3,d2,d1,y0,y1,y2,y3,y4,y5,y6,y7,y
3  input c1,c2,d4,d3,d2,d1;
4  output y0,y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15;
5  reg y0,y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15;
6  always @ *
7  begin
8      y0=1'b1;
9      y1=1'b1;
10     y2=1'b1;
11     y3=1'b1;
12     y4=1'b1;
13     y5=1'b1;
14     y6=1'b1;
15     y7=1'b1;
16     y8=1'b1;
17     y9=1'b1;
18     y10=1'b1;
19     y11=1'b1;
20     y12=1'b1;
21     y13=1'b1;
22     y14=1'b1;
23     y15=1'b1;

```

Рис. 13: Фрагмент кода KP133ID3

3.1.4 KP1533SP1

Микросхема KP1533SP1 [20]— схема сравнения для четырехразрядных двоичных чисел. Имеет 3 входа и 3 выхода сравнения и 8 информационных входов. Сравнение чисел производится со старших разрядов. Если два числа не равны, то, независимо от входов сравнения, на выходы поступают результаты сравнения. При равенстве чисел А и В принцип работы элемента указан на Рис. 14. Схема имеет все средства для сравнения чисел большей разрядности без использования дополнительных элементов. Для этого необходимо выходы сравнения микросхемы, сравнивающей младшие разряды, соединить с соответствующими входами микросхемы, производящей сравнение старших разрядов. А на вход равенства (один из входов сравнения) микросхемы, производящей сравнение первых младших разрядов, подать высокий уровень сигнала.

Вход сравнения			Выход (результат)		
A>B	A<B	A=B	A>B	A<B	A=B
H	L	L	H	L	L
L	H	L	L	H	L
X	X	H	L	L	H
H	H	L	L	L	L
L	L	L	H	H	L

Рис. 14: Принцип работы KP1533SP1

```

1  module ol_ic_1553sp1(BIG_IN, SMALL_IN, EQUAL_IN, A0, A1, A2, A3,
2  input BIG_IN, SMALL_IN, EQUAL_IN, A0, A1, A2, A3, B0, B1, B2, B3;
3  output BIG_OUT, SMALL_OUT, EQUAL_OUT;
4  wire [3:0]A;
5  wire [3:0]B;
6  assign A[0] = A0; assign A[1] = A1; assign A[2] = A2; a:
7  reg BIG_OUT; reg SMALL_OUT; reg EQUAL_OUT;
8  always @ *
9  begin
10 if (A == B)
11     begin
12         if (EQUAL_IN == 1'b1)
13             begin
14                 BIG_OUT = 1'b0;
15                 SMALL_OUT = 1'b0;
16                 EQUAL_OUT = EQUAL_IN;
17             end
18         else if (BIG_IN == SMALL_IN)
19             begin
20                 BIG_OUT = ~(BIG_IN);
21                 SMALL_OUT = ~(SMALL_IN);
22                 EQUAL_OUT = EQUAL_IN;
23             end

```

Рис. 15: Фрагмент кода KP1533SP1

3.2 Электронная модель схемы

После создания электронных моделей всех компонентов осуществляется моделирование объекта контроля. После добавления моделей, установления связей между ними, а также подписей всех необходимых элементов получим электронную модель Субблок 714ПЭЗ (Рис. 16), которую необходимо сохранить в нужном формате (.v) для работы в САПР «SimTest».

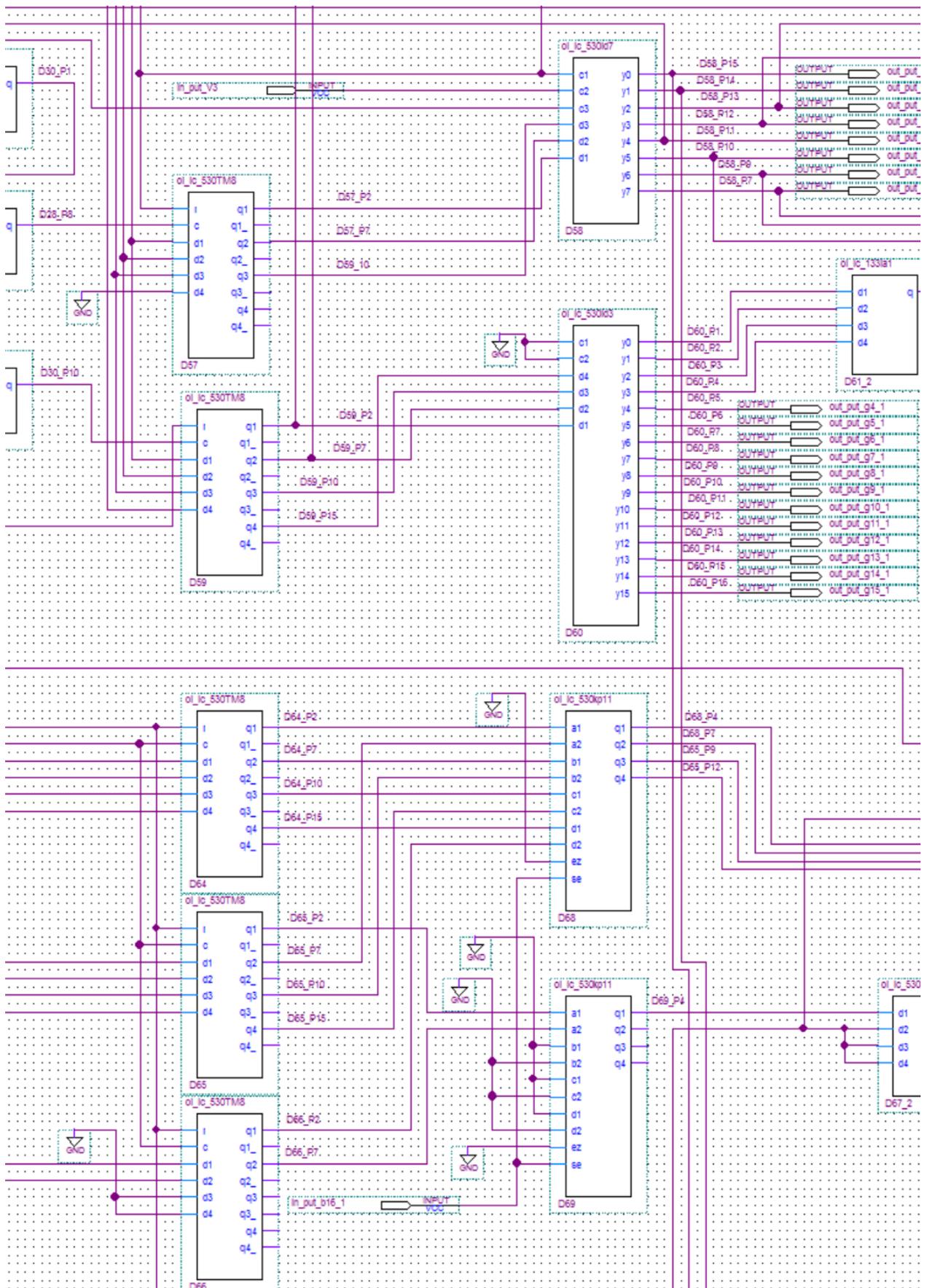


Рис. 16: Фрагмент программной модели Субблок 714ПЭЗ

Глава 4. Тестирование

После составления программной модели Субблок 714ПЭЗ необходимо провести ее тестирование. Вначале будет проведено тестирование отдельных компонентов для проверки правильности описания логики их работы, а после уже самого объекта контроля. Стоит отметить, что тестирование компонентов можно проводить и до, и после «сборки» схемы, так как ее вид не зависит от логики работы элементов.

4.1 Тестирование логики элементов

Рассмотрим тесты элементов, описанных в предыдущей главе.

4.1.1 KP530LA1

После того, как программная модель была создана, необходимо открыть ее в САПР «SimTest», распределить краевые разъемы на входные и выходные и привязать логический интерфейс. В данном случае используется «группа сигналов» (Рис. 17).

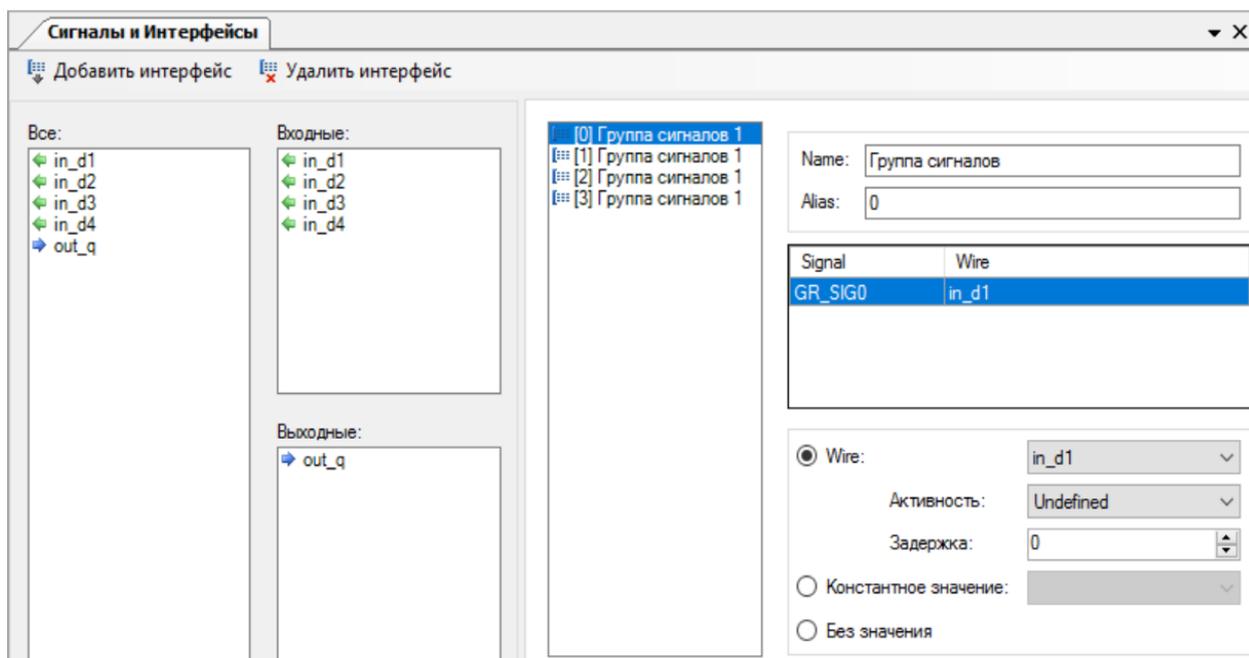


Рис. 17: Привязка и распределение сигналов элемента KP530LA1

Процесс изменения входов описывается вручную (Рис. 18). (val a b, a — такт, на котором происходит смена сигнала, b — значение, поступающие на вход).

После получим сам тест (Рис. 19), по результатам которого можно убедиться в правильности написания программной модели KP530LA1.

```
//d1
#GROUP_SIG 0
val 0 0
val 1 1
val 2 0
val 5 1
val 7 0
val 11 1
val 16 0
//d2
#GROUP_SIG 1
```

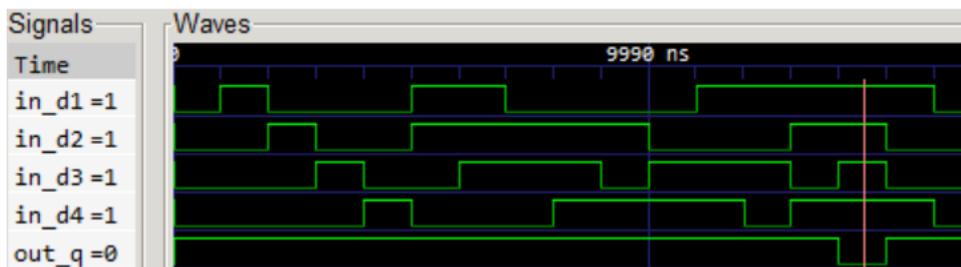


Рис. 19: Тест элемента KP530LA1

Рис. 18: Фрагмент тестовой последовательности элемента KP530LA1

4.1.2 KP530TM2

Аналогично элементу микросхемы KP530LA1 для элемента KP530TM2 привязываем краевые разъемы. Однако поскольку этот элемент содержит память, то первым делом необходимо подать на него такие сигналы, чтобы хранящиеся в нем данные стали известными. В связи на первом такте были поданы сигналы: d — низкий, s — высокий. Как видно из Рис. 21 элемент работает правильно.

```
//c
#GROUP_SIG 0
val 0 0
val 2 1
val 3 0
val 5 1
val 6 0
val 8 1
val 9 0
val 11 1
val 12 0
val 14 1
... 15 0
```

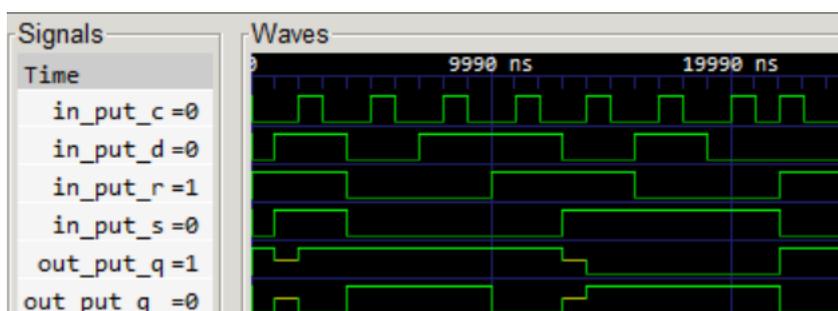


Рис. 21: Тест элемента KP530TM2

Рис. 20: Фрагмент тестовой последовательности элемента KP530TM2

4.1.3 KP133ID3

Поскольку на входы были поданы все комбинации и схема не содержит памяти, то покрытие будет составлять 100%. Анализируя Рис. 22 можно сделать вывод о правильной работе элемента.

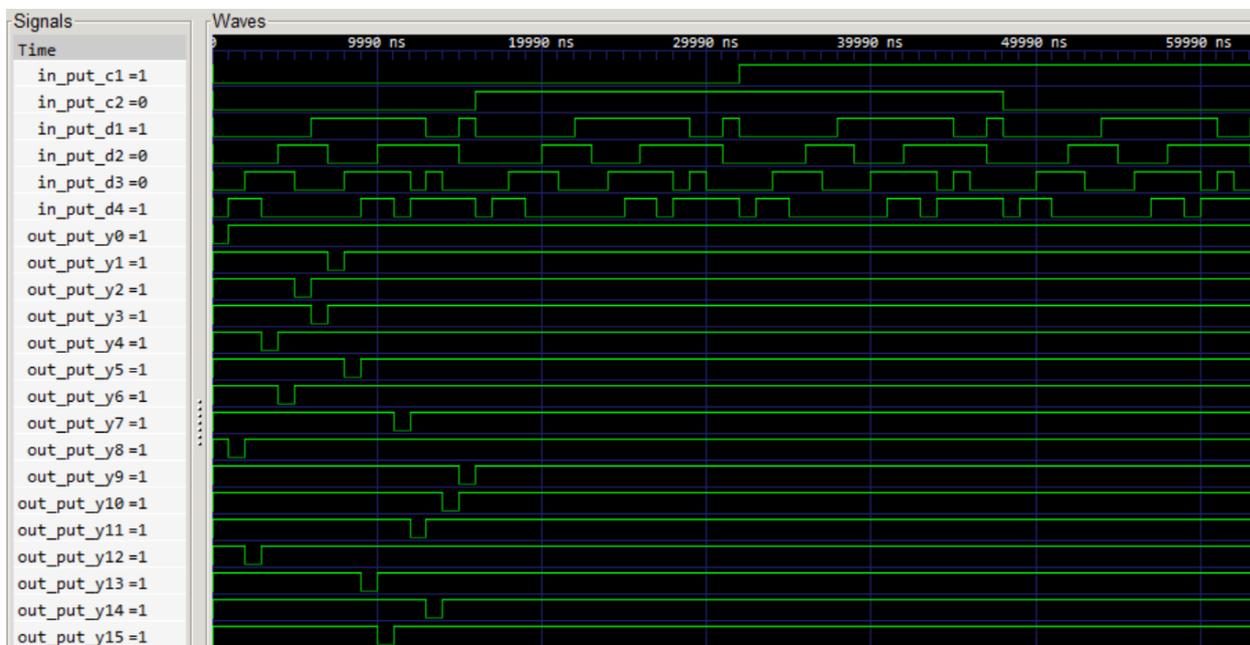


Рис. 22: Тест KP133ID3

4.1.4 KP1533SP1

Данная схема также не содержит памяти. Проверим сравнение по всем разрядам, а также комбинации для наращивания числа разрядов (Рис. 23).

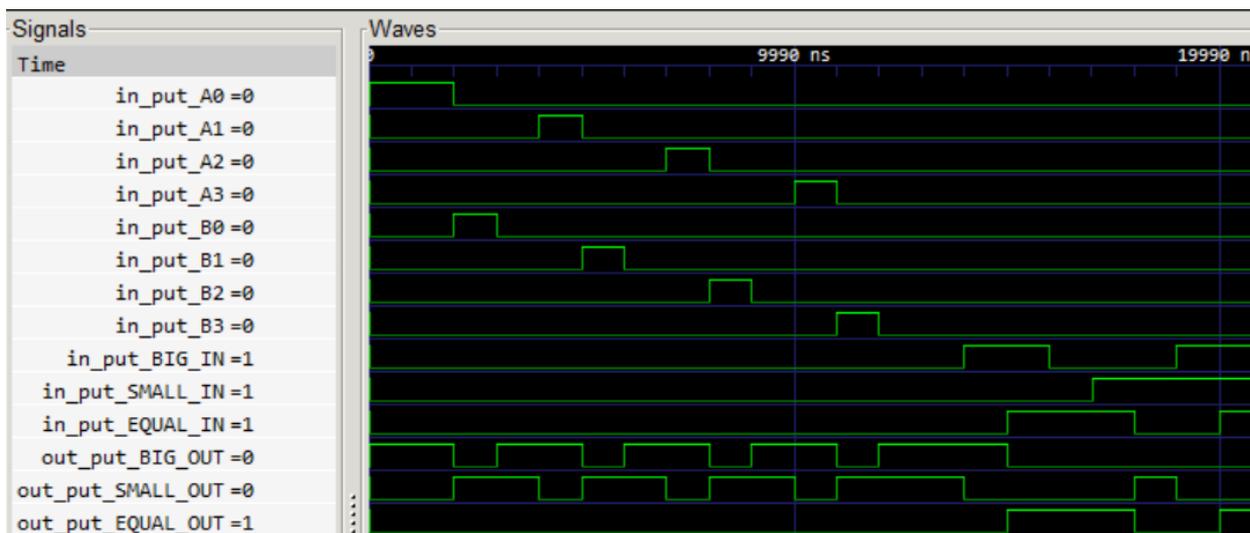


Рис. 23: Тест KP1533SP1

4.2 Тестирование Субблок 714ПЭЗ

Следующим шагом после тестирования логических элементов является создание теста самого устройства. Первым делом необходимо произвести привязку логического интерфейса к разъемам объекта (Рис. 24).

После формирования тестовых последовательностей получили общее тестовое покрытие в 94.16% (Рис. 25). Это можно объяснить тем, что существуют неизменяемые входы: подключенные к источникам постоянного напряжения. Входы и выходы самого устройства были задействованы на 100%, что подтверждает Рис. 25. Фрагмент теста можно увидеть на Рис. 26.

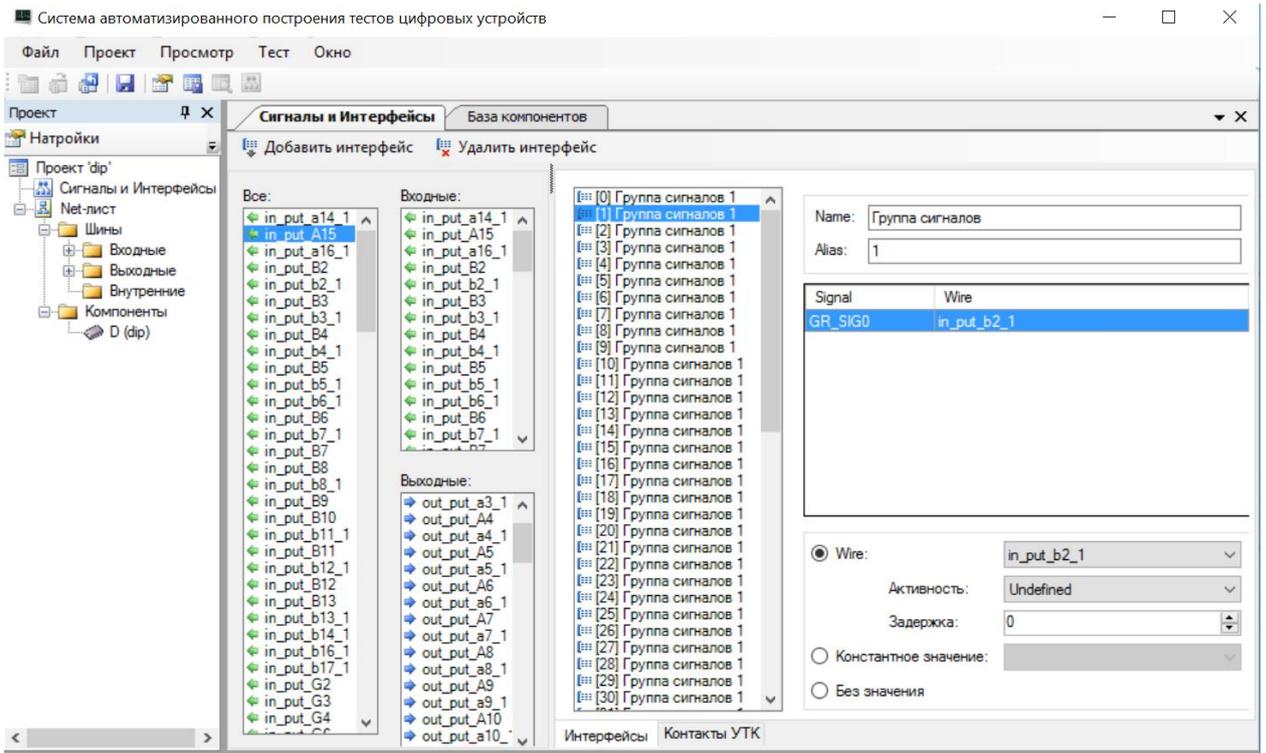


Рис. 24: Привязка логического интерфейса

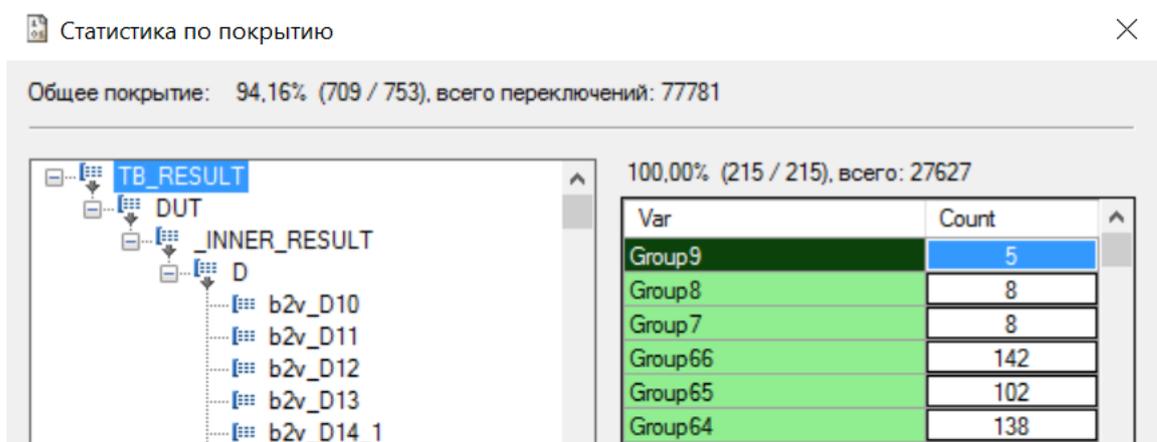


Рис. 25: Тестовое покрытие Субблок 714ПЭ3

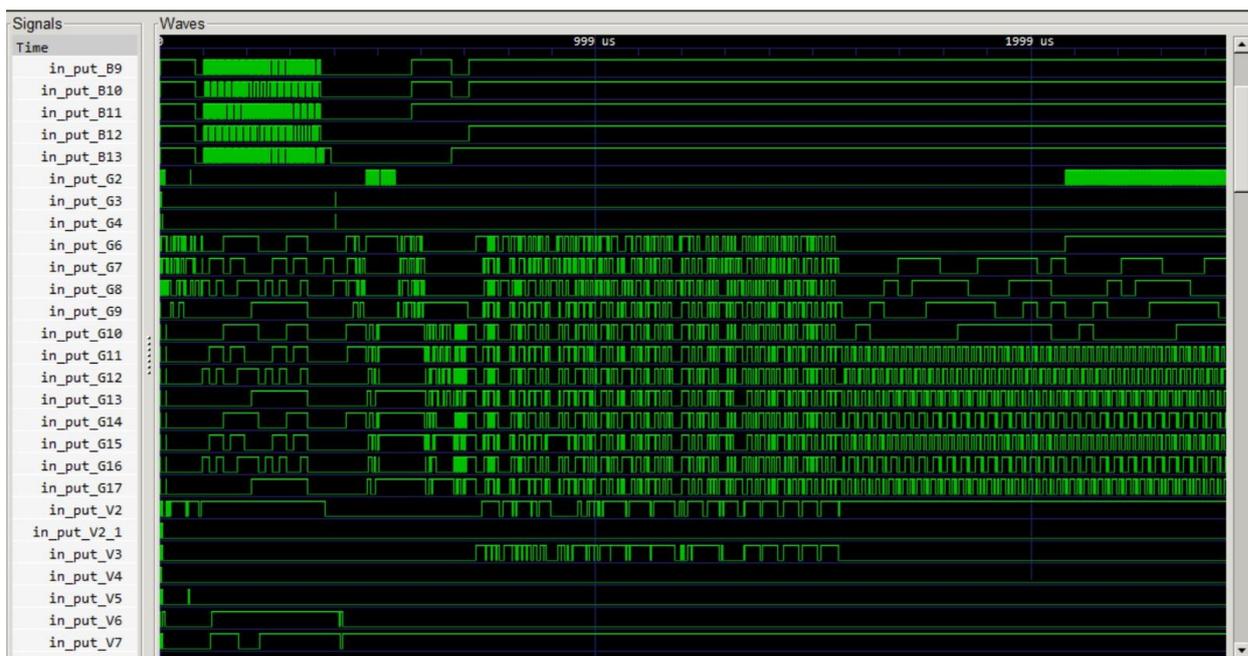


Рис. 26: Фрагмент теста Субблок 714ПЭ3

4.2.1 Проблематика тестирования Субблок 714ПЭ3

При тестировании данного устройства возникло несколько проблем. Первая проблема связана с недостаточно большой разрешающей способностью САПР «SimTest», для этого была введена задержка распространения сигнала на некоторых элементах. С помощью ее можно отследить, например, изменение выхода (Рис. 27) в элементе D76_1 (часть КР530ЛА16). Также задержка «помогла» изменить выходные значения D-триггера в элементе D16_2 (Рис. 28 и Рис. 29). Стоит отметить, что введение задержки не отразится на логике работы всего цифрового устройства.

Другая проблема состояла в необходимости направить все возможные комбинации входных сигналов на все элементы. Например, элемент D68 (КР530КР11) имеет 9 непостоянных входов (Рис. 30) и 512 различных входных комбинаций.

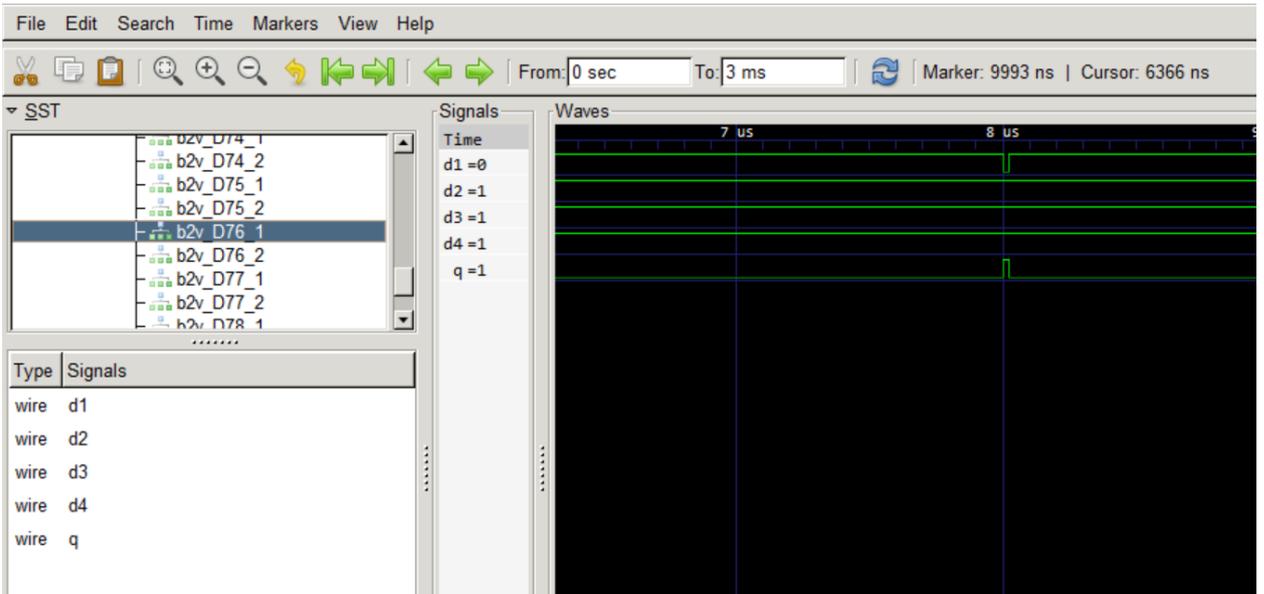


Рис. 27: Тест элемента KP530LA16



Рис. 28: Тест D16_2 без использования задержки (элемент KP530TM2)

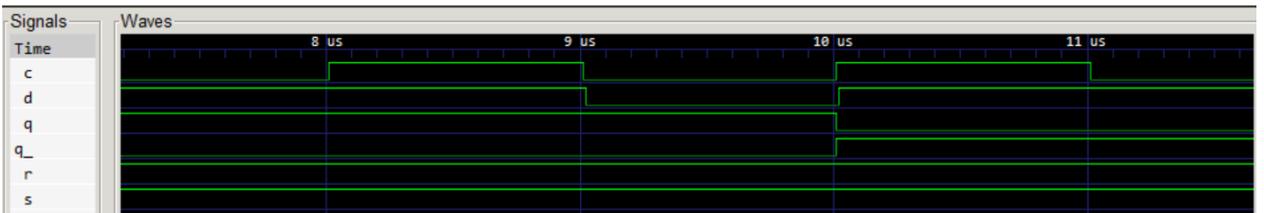


Рис. 29: Тест D16_2 с использованием задержки (элемент KP530TM2)

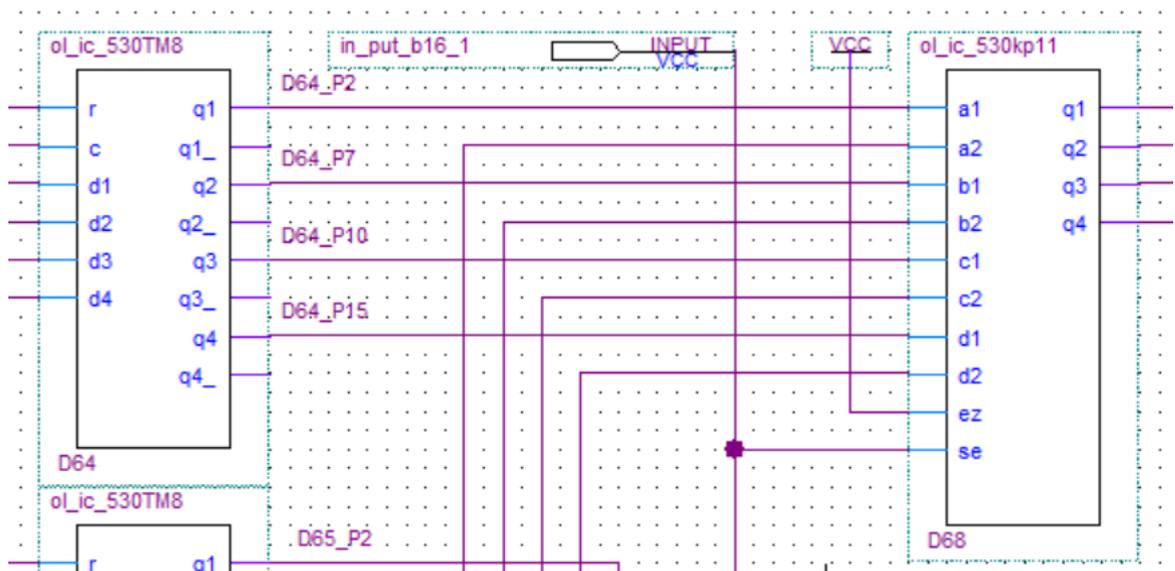


Рис. 30: D68 (KP530KP11)

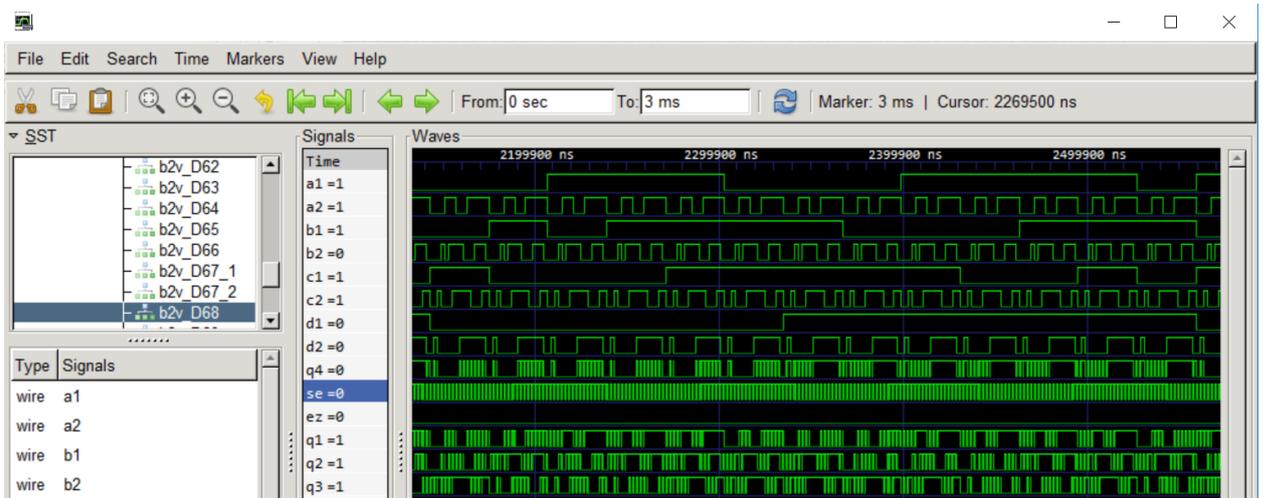


Рис. 31: Тест D68 (КР530КР11)

Заключение

В работе рассмотрены следующие типы тестирования:

- визуальный контроль;
- внутрисхемное тестирование;
- периферийное сканирование;
- функциональное тестирование.

Выполнение необходимых пунктов для написания тестовой программы реальной схемы реализовано на примере Субблок 714ПЭЗ:

1. Описана логика работы входящих в ее состав микросхем: КР530ЛА1, КР530ТМ2, КР133ИД3, КР1533СП1 и др. на языке Verilog HDL;
2. Разработана программная модель всех компонентов цифрового устройства с использованием программы разработки: ALTERA QUARTUS II;
3. Произведена разработка программной модели самого цифрового устройства;
4. С использованием САПР «SimTest» создан тест для контроля исправности цифрового устройства.

Разработанная тестовая программа обладает тестовым покрытием 94%. Это объясняется существованием неизменяемых входов, подключенных к источникам постоянного напряжения.

Список литературы

1. Мельник В., Гришкин В., Михайлов А., Овсянников Д. Тестовый контроль и диагностика радиоэлектронной аппаратуры // Электроника: Наука, технология, бизнес. 2013. Вып. 128. С. 114-117.
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. Вып. 128. С. 118-124.
3. Малашевич Б.М. 50 лет отечественной микроэлектронике. Краткие основы и история развития. М.: Техносфера, 2013. 800 с.
4. Барыбин А.А., Томилин В.И., Шаповалов В.И. Физико-технологические основы макро-, микро- и нанoeлектроники. М.: ФизМатЛит, 2011. 784 с.
5. Задорин А., Захаров Г. Система автоматизированного визуального контроля печатных плат Aplite, 2002. Вып. 10. С. 53-56.
6. Иванов А. Выбор тестовой стратегии при производстве цифровой и аналогово-цифровой техники. // Печатный монтаж. 2010. Вып. 2. С. 48-53.
7. Albee A. J. The evolution of ICT: PCB technologies, test philosophies, and manufacturing business models are driving in-Circuit test evolution and innovations // IPC APEX EXPO Conference and Exhibition 2013. No 1. P. 381–401.
8. Иванов А. Периферийное сканирование в жизненном цикле изделия / Производство электроники: технологии, оборудование, материалы. 2010. Вып. 2. С 1-3.
9. Renbi A., Delsing J. Contactless Testing of Circuit Interconnects // Journal of Electronic Testing: Theory and Applications, 2015, No 31. P. 229–253.
10. Wang R., Chakrabarty K., Eklow B. Scan-based testing of post-bond silicon interposer interconnects in 2.5-D ICs // IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems, 2014. No 33. P. 1410–1423.

- 11.Sangi R., Baranski M., Oltmanns J., Streblow R., Müller D. Modeling and simulation of the heating circuit of a multi-functional building // *Energy and Buildings*, 2016. No 110. P. 13–22.
- 12.Thoulath Begam V. M., Baulkani S. Compact test set method for high fault coverage test pattern generation // *International Journal of Applied Engineering Research*, 2015. Vol. 10, No 55. P. 453–458.
- 13.Ghiduk A. S. Automatic generation of basis test paths using variable length genetic algorithm // *Information Processing Letters*, 2014. Vol. 114, No 6. P. 304–316.
- 14.Кондратенко Ю.П., Мохор В.В., Сидоренко С.А. Verilog-HDL для моделирования и синтеза цифровых электронных схем. Н.: НГГУ им. Петра Могилы, 2002. 207 с.
- 15.Комолов Д.А., Мяльк Р.А., Зобенко А.А. Филиппов А.С. Системы автоматизированного проектирования фирмы Altera MAX+Plus II и Quartus II. Краткое описание и самоучитель. М.:Радиософт, 2002, 361 с.
- 16.Угрюмов Е.П. Цифровая схемотехника. Изд. 3. СПб.: БХВ-Петербург, 2010. 816 с.
- 17.<http://www.datasheet-pdf.ru/1533/1533pdf/1533LA1.pdf>
- 18.<http://www.datasheet-pdf.ru/1533/1533pdf/1533TM2.pdf>
- 19.<http://www.datasheet-pdf.ru/1533/1533pdf/1533ID3.pdf>
- 20.<http://www.datasheet-pdf.ru/1533/1533pdf/1533SP1.pdf>
- 21.Шило В. Л. Популярные цифровые микросхемы. М.: Радио и связь, 1987. 352 с.