

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА ТЕОРИИ СИСТЕМ УПРАВЛЕНИЯ ЭЛЕКТРОФИЗИЧЕСКОЙ
АППАРАТУРОЙ

Пушко Федор Александрович

Выпускная квалификационная работа бакалавра

**Моделирование и тестирование радиоэлектронных
устройств**

Направление 010900

Прикладные математика и физика

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	5
История развития схемотехники	6
Современные методы тестирования радиоэлектронных устройств	7
Алгоритм создания программной модели и написание теста для основных классов цифровых микросхем.....	10
1533ЛА2	10
1533ИДЗ	13
1533ТМ2.....	16
1533ИР23.....	19
1533ИЕ7	22
Программное моделирование и написание теста сложного радиоэлектронного устройства Субблок ПЭЗ.....	25
Выводы.....	28
Заключение	29
Литература	30

Введение

Сейчас сложно представить наше существование без технических устройств, которые окружают нас в повседневной жизни. Все вокруг, к чему мы так сильно привыкли, будь то чайник или телефон, так или иначе связано с электроникой. Функциональная электроника пересекается со многими научными отраслями. Благодаря ее развитию, не за горами те времена, когда наборы функциональных микрокомпонентов смогут автоматически управлять искусственными органами: сердцем, почками, желудком, что, безусловно, сделает прорыв в медицине. Наука не стоит на месте и современные исследования требуют огромных вычислительных мощностей, которые невозможны без развития электроники. Отдельно стоит отметить электронику для космических аппаратов: необходимо не просто спроектировать элементы, но и сделать их пригодными для работы в экстремальных условиях.

Именно поэтому так стремительно развивается схемотехника. И если раньше тестированием на этапе производства можно было заниматься вручную, то вместе со стремительным развитием сложных цифровых устройств возникает необходимость в системном и комплексном подходе к организации процессов технической диагностики, анализа и контроля каждого отдельного элемента и устройства в целом. Для этого создаются и используются системы автоматизированного проектирования тестов, развиваются, совершенствуются и внедряются новые комплексы тестового контроля. Одной из таких систем является разработка СПбГУ «SimTest», которая принимает на входе структуру объекта контроля (ОК), описания алгоритма работы его компонентов, а также тестовые воздействия. На основе этой информации формируется полная модель устройства, генерируются последовательности входных воздействий и реакции ОК.

В данной работе рассмотрены основные принципы разработки программных моделей в «Quartus II», а также создание тестов для отдельных микросхем и сложного цифрового радиоэлектронного устройства, с использованием системы «SimTest».

Постановка задачи

Целью работы является создание модели радиоэлектронного устройства Субблок ПЭЗ, а также составление тестовой программы, предназначенной для контроля работоспособности устройства. Субблок ПЭЗ состоит из 73 элементов, представленных 18 различными типами микросхем, а также содержит 134 внутренние связи. Таким образом, список задач, необходимых для достижения поставленной цели, выглядит следующим образом:

- Создать программную модель каждого элемента, входящего в состав радиоэлектронного устройства, на языке описания аппаратных средств в среде разработки ALTERA QUARTUS II.
- Объединить полученные модели для создания модели радиоэлектронного устройства Субблок ПЭЗ.
- Смоделировать тест для проверки работоспособности объекта контроля в САПР «SimTest».

При написании теста стоит учитывать, что необходимо добиться максимально возможного покрытия радиоэлектронного устройства, чтобы проверка компонентов и внутренних сигнальных линий была полноценной.

История развития схемотехники

Первым этапом становления схемотехники, как и электроники в целом, можно считать 1906 год, когда Д.А.Флеминг изобрел первый электронный прибор с односторонней проводимостью, имеющий более общее название – ламповый диод. Следующим шагом был эксперимент Ли де Фореста в 1907 году, в котором он ввел в диод между катодом и анодом третий электрод «управляющую сетку», таким образом был получен триод.

Второй этап развития датируется 1930 годом, когда Ю.Лиленфельдом была придумана концепция первого полевого транзистора, а в дальнейшем, в 1948 году был изобретен первый полупроводниковый триод, что открыло новые возможности для реализации множества идей и подтолкнуло к более широкому развитию схемотехники в мире.

Третьим этапом стал прорыв, осуществленный в конце 1958 – начале 1959 года Джеком Килби, Куртом Леговцем и Робертом Нойсом, сумевшие решить три фундаментальные проблемы, которые позволили создать первую в мире интегральную схему.

По мере развития схемы становились сложнее и функциональней, появилась возможность запускать их серийное производство. Сейчас мы умеем производить большие интегральные схемы, на основе которых можно создать удивительные и многофункциональные устройства. Но вместе с этим многократно увеличивается их сложность. При этом растет риск появления брака(микросхемы с дефектом) в следствие человеческого фактора. Всё это приводит к необходимости применять автоматизацию контрольно-диагностических работ на каждом этапе производства.

Современные методы тестирования радиоэлектронных устройств

В настоящее время неотъемлемой частью серийного производства радиоэлектронных устройств является контроль функциональности и тестирование сборки. Существует несколько видов автоматического тестирования, а именно:

- Визуальный автоматизированный контроль
- Внутрисхемное тестирование
- Периферийное/граничное сканирование
- Функциональное тестирование

Рассмотрим каждый из них подробнее.

Визуальный автоматизированный контроль – это предварительная проверка качества, которая проходит на разных стадиях монтажа печатных плат. Для достижения результата используются как стандартные оптические системы, так и рентгеновское излучение, что позволяет проверять места, недоступные глазу. К сожалению недостатком этого метода является относительная сложность, и, в следствие чего, дороговизна оборудования.

Внутрисхемное тестирование – это технология проверки фрагментов схем либо отдельных компонентов. Для этого вида тестирования используется специальное оборудование и оснастки. Очевидным плюсом этого метода является высокая скорость тестирования, что позволяет успешно применять его в массовом производстве. Из минусов стоит отметить, что этот метод подразумевает физический контакт иглок с контактами тестируемых компонентов, что вызывает очевидные трудности в случае многослойных печатных плат, либо при слишком миниатюрных компонентах[1-2]. Одним из путей решения этой проблемы является использование «летающих щупов»[3], но данный подход значительно

уменьшает производительность, что недопустимо для массового производства.

Периферийное/граничное сканирование – метод, позволяющий выявлять брак еще до стадии функционального тестирования. Другое его название JTAG-тестирование. Он позволяет выявить замыкания, обрывы, «непропаи». В следствие большого распространения стандарта JTAG, этот метод становится популярней в последнее время. Кроме того, многие производители сопровождают схемы BSDN-файлами, которые значительно упрощают подготовку и использование периферийного тестирования. Из минусов стоит отметить низкую скорость тестирования, а также возможность проверки только целостности связей, но не их качество[4-5].

Функциональное тестирование позволяет проверить устройство на выполнение определенно заданной функциональности в автоматическом режиме. При правильно построенном тесте, этот метод способен охватить всю функциональность устройства, при достаточно высокой скорости тестирования[6-12]. Из минусов стоит отметить необходимость разработки тестового программного обеспечения, а также невозможность точного указания на дефектные цепи или выводы проверяемого устройства.

Таким образом, становится очевидным, что при производстве необходимо учитывать плюсы и минусы всех методов автоматического тестирования и пытаться подобрать такое их сочетание, чтобы получить максимальное покрытие каждого устройства за минимальное время.

Наиболее перспективным представляется функциональное тестирование, которое обладает следующими преимуществами перед другими типами тестирования радиоэлектронных устройств:

- Не требует поддержки интерфейса JTAG и имеет высокую скорость тестирования в отличие от периферийного сканирования.

- Не требует сложных, дорогостоящих устройств в отличие от систем автоматизированного визуального контроля.

- Не зависит от размеров и многослойности радиоэлектронного устройства в отличие от внутрисхемного тестирования.

Разработанная в СПбГУ САПР «SimTest» позволяет автоматизировать процесс создания тестовой программы, необходимой для функционального тестирования цифрового устройства[13-15].

Алгоритм создания программной модели и написание теста для основных классов цифровых микросхем

1533ЛА2

Микросхема представляет собой элемент на восемь входов и один выход, выполняющий элементарную Булеву функцию $y = \sim(d1 \& d2 \& d3 \& d4 \& d5 \& d6 \& d7 \& d8)$ [16]. Для моделирования и дальнейшего тестирования элемента, нам необходимы определенные данные, а именно:

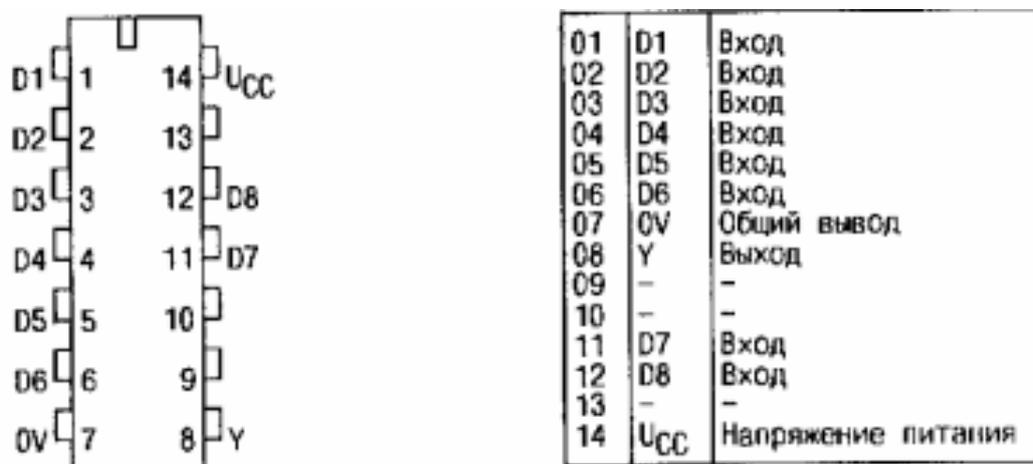


Рисунок 1 Расположение и назначение выводов 1533ЛА2

Из данных таблиц мы узнаем, сколько входов и выходов содержит микросхема и какое назначение имеет каждый из них.

D1	D2	D3	D4	D5	D6	D7	D8	Y
H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	H
X	L	X	X	X	X	X	X	H
X	X	L	X	X	X	X	X	H
X	X	X	L	X	X	X	X	H
X	X	X	X	L	X	X	X	H
X	X	X	X	X	L	X	X	H
X	X	X	X	X	X	L	X	H
X	X	X	X	X	X	X	L	H

Рисунок 2 Таблица истинности 1533ЛА2

Данная таблица показывает алгоритм работы микросхемы, что позволяет перейти непосредственно к написанию кода, с которым будет продолжаться дальнейшая работа:

```
always@*
begin
    if (d1==1'b0 || d2==1'b0
        || d3==1'b0 || d4==1'b0
        || d5==1'b0 || d6==1'b0 ||
        d7==1'b0 || d8==1'b0)
        y=1'b0;
    else
        y=1'b1;
end
```

Рисунок 3 Фрагмент кода 1533ЛА2 в среде разработки «Quartus II»

После написания и компиляции полученного кода, он преобразуется в блок-схему, которая используется в дальнейшем для написания теста:

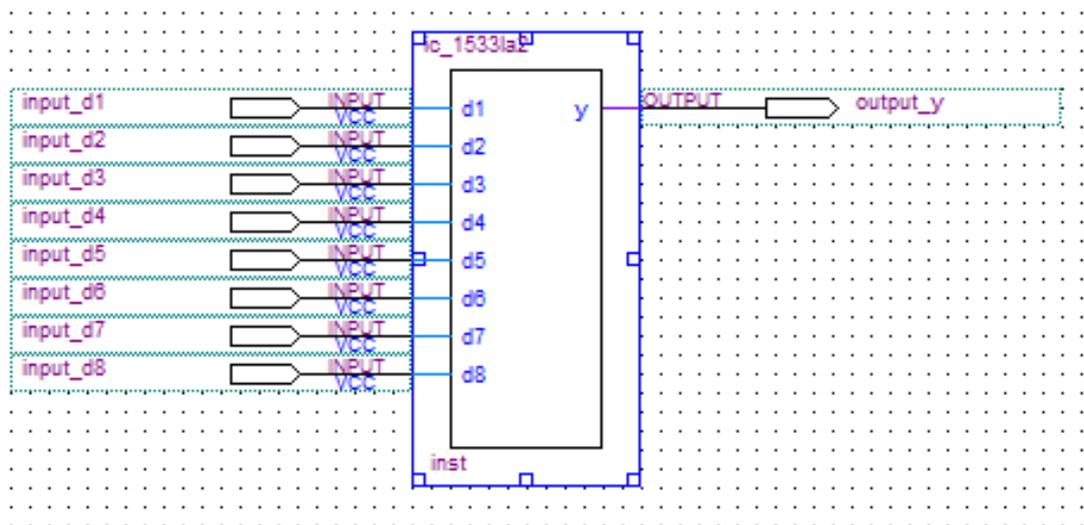


Рисунок 4 Блок-схема 1533ЛА2

Последним шагом будет написание теста и сравнение полученных результатов с таблицей истинности, как видно на рисунке все сходится:



Рисунок 5 Результаты моделирования в "SimTest"

Сам по себе элемент достаточно простой в реализации, но из-за большого количества входов и выходов, код получается достаточно громоздким, состоящим из огромного кол-ва блоков «if», ниже представлена часть кода:

```

if (d4==1'b0)
begin
  if(d3==1'b0 && d2==1'b0 && d1==1'b0)
  begin
    y0=1'b0;y1=1'b1;y2=1'b1;y3=1'b1;
    y4=1'b1;y5=1'b1;y6=1'b1;y7=1'b1;
    y8=1'b1;y9=1'b1;y10=1'b1;y11=1'b1;
    y12=1'b1;y13=1'b1;y14=1'b1;y15=1'b1;
  end

  if(d3==1'b0 && d2==1'b0 && d1==1'b1)
  begin
    y0=1'b1;y1=1'b0;y2=1'b1;y3=1'b1;
    y4=1'b1;y5=1'b1;y6=1'b1;y7=1'b1;
    y8=1'b1;y9=1'b1;y10=1'b1;y11=1'b1;
    y12=1'b1;y13=1'b1;y14=1'b1;y15=1'b1;
  end
end

```

Рисунок 8 Фрагмент кода 1533ИДЗ

Остальная часть кода делается по аналогии, таким образом, на выходе после компиляции получаем нужную нам блок-схему:

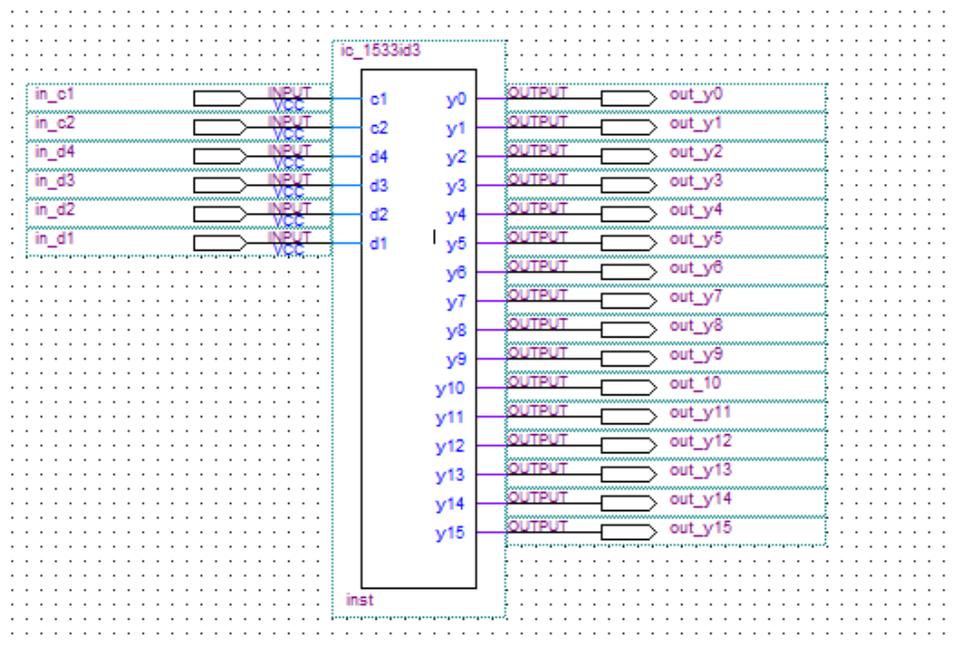


Рисунок 9 Блок-схема 1533ИДЗ

Создаем тест в «SimTest» и получаем его результат:

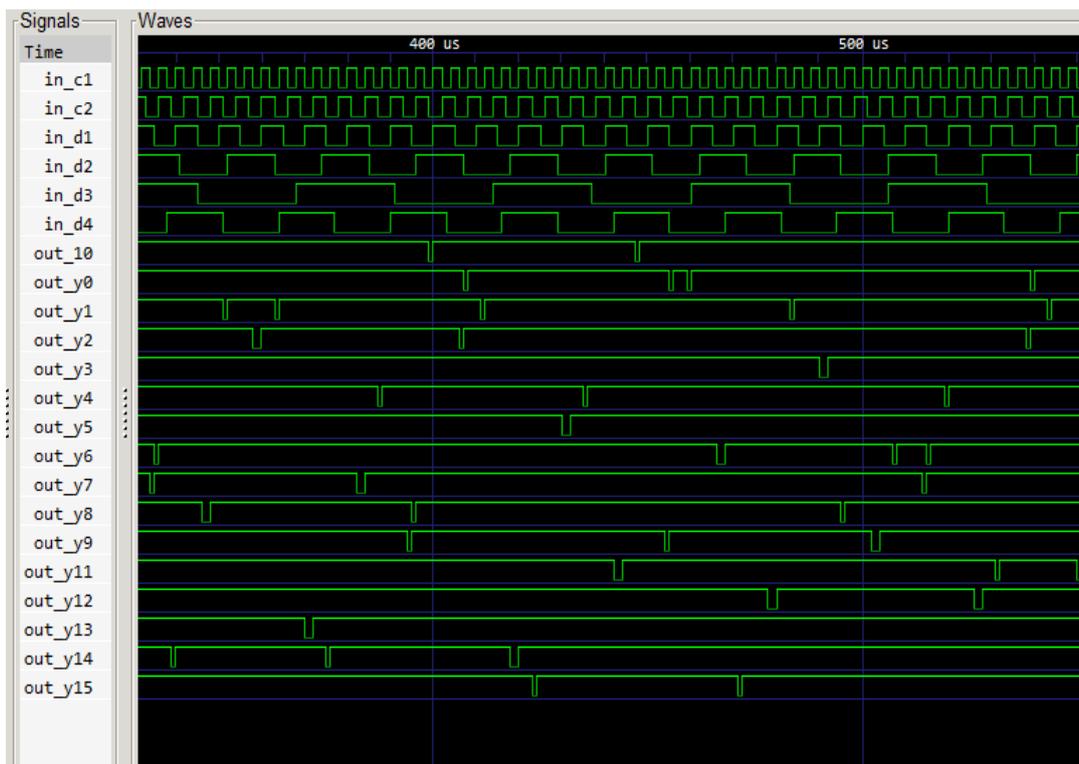


Рисунок 10 Результаты моделирования в "SimTest"

Как видно из теста, в любой момент времени только один выход имеет напряжение низкого уровня. Сравнив с таблицей истинности, можно судить о том, что логика элемента описана верно.

1533TM2

Микросхема представляет собой два независимых D-триггера, на четыре входа и два выхода каждый. Триггер - это элемент, который используется для запоминания одного разряда двоичного кода. Триггер имеет два устойчивых состояния, соответствующих логической единице и логическому нулю соответственно[18].

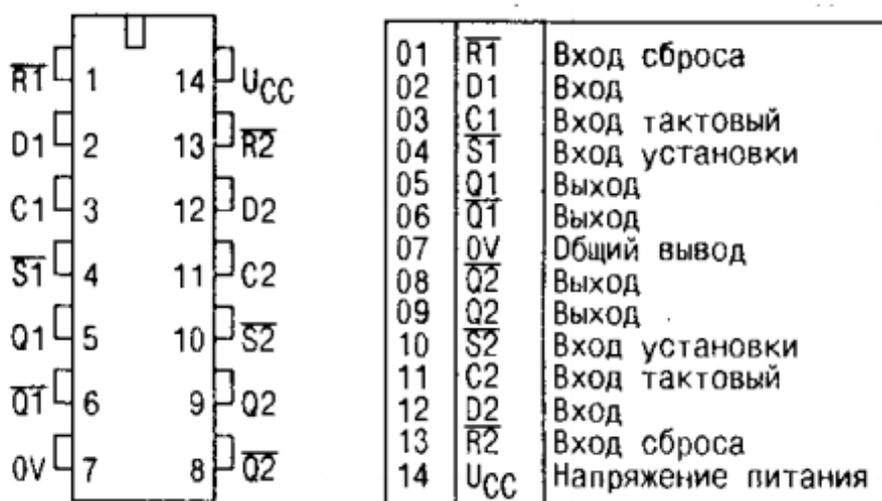


Рисунок 11 Расположение и назначение выводов 1533TM2

На данном рисунке показано, сколько входов и выходов имеет микросхема, а также их назначение.

Входы				Выходы	
\overline{S}	\overline{R}	C	D	Q	\overline{Q}
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	\int	H	H	L
H	H	\int	L	L	H
H	H	L	X	Q ₀	$\overline{Q_0}$

Рисунок 12 Таблица истинности 1533TM2

Обращаясь к описанию и данным рисункам, мы переходим к написанию самого кода:

```
always@(r or s)
begin
    if((s==1'b0) && (r==1'b1))
    begin
        q=1'b1; q_=1'b0;
    end
    if((s==1'b1) && (r==1'b0))
    begin
        q=1'b0; q_=1'b1;
    end
    if((s==1'b0) && (r==1'b0))
    begin
        q=1'b1; q_=1'b1;
    end
end
/*always@(posedge c)
begin
    ((s==1'b1) && (r==1'b1) && (d==1'b1))
    begin
        q=1'b1; q_=1'b0;
    end
    ((s==1'b1) && (r==1'b1) && (d==1'b0))
    begin
        q=1'b0; q_=1'b1;
    end
end*/
```

Рисунок 13 Фрагмент кода 1533ТМ2

Стоит обратить внимание на то, что один из блоков «always» закомментирован. Это связано с тем, что «Quartus II» не дает реализовать несколько блоков always срабатывающих одновременно, поэтому при компиляции мы все кроме одного «скрываем», а уже при создании теста в «SimTest», открываем все блоки обратно.

Далее, опираясь на код, мы создаем блок-схему, подводим входы и выходы, а также подписываем их:

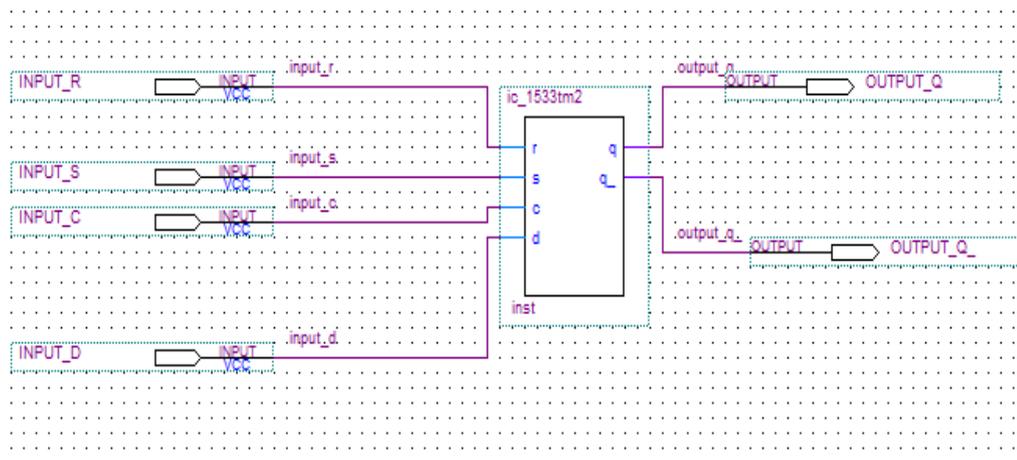


Рисунок 14 Блок-схема 1533TM2

Заходим в «SimTest», создаем новый тест и сверяем результат с таблицей истинности:

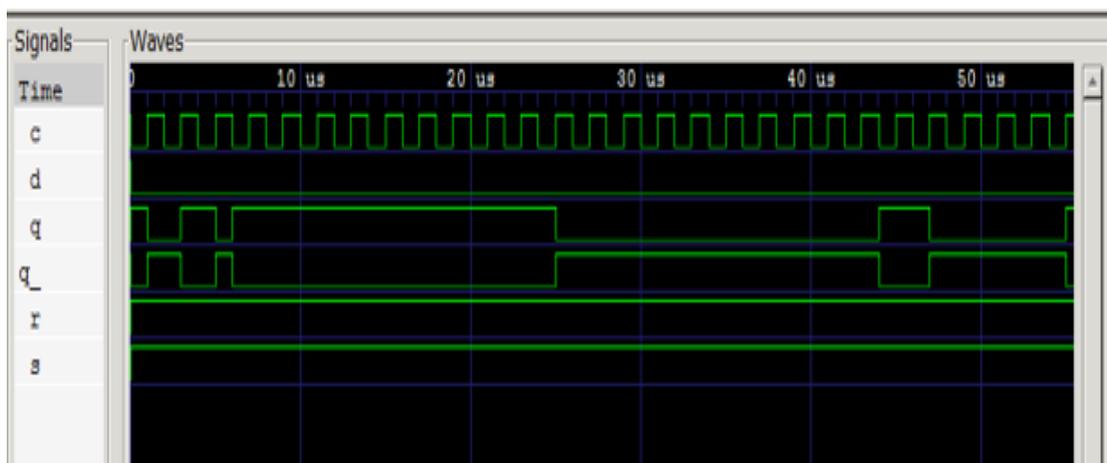


Рисунок 15 Результат теста 1533TM2

1533ИР23

Элемент представляет собой восьмиразрядный регистр на триггерах D-типа с тремя состояниями на выходе. Регистр представляет собой упорядоченный набор D-триггеров, число n которых соответствует числу разрядов в слове. Данная микросхема может быть использована в качестве регистра, буферного регистра, регистра ввода-вывода, магистрального передатчика и др[19].

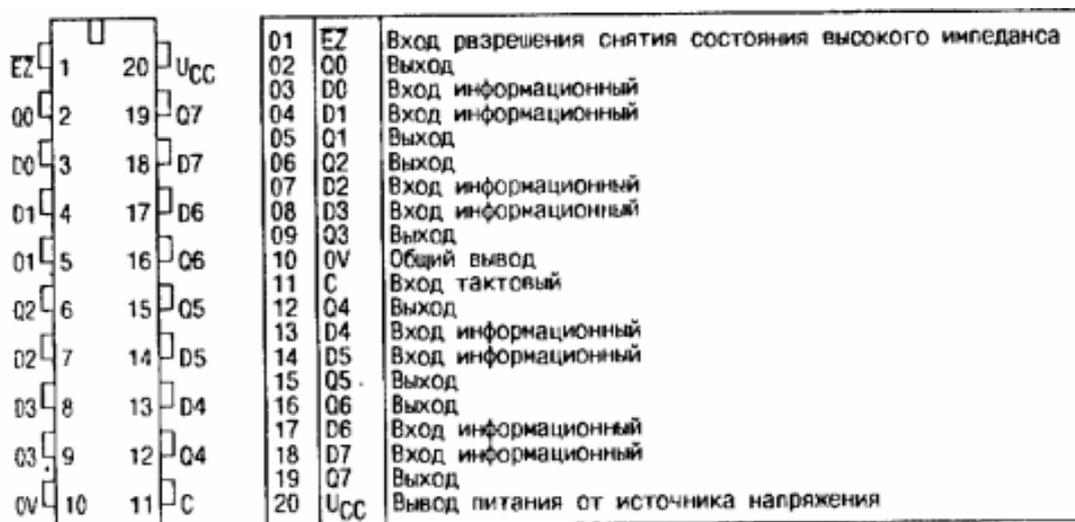


Рисунок 16 Расположение и назначение выводов 1533ИР23

Таким образом, наша схема имеет 10 входов и 8 выходов.

Входы			Выход
EZ	C	D	Q
L	∩	H	H
L	∩	L	L
L	L	X	Q ₀
H	X	X	Z

Рисунок 17 Таблица истинности 1533ИР23

Опираясь на описание элемента и данные рисунки, мы можем перейти к написанию самого кода:

```
input ez, c, d0, d1, d2, d3, d4, d5, d6, d7;
output q0, q1, q2, q3, q4, q5, q6, q7;
reg [0:7] d;

always@(posedge c)
begin
    d <= {d0, d1, d2, d3, d4, d5, d6, d7};
end
assign {q0, q1, q2, q3, q4, q5, q6, q7} = (ez == 1'b1)? 8'bz : d;

endmodule
```

Рисунок 18 Фрагмент кода 1533ИР23

Опираясь на написанный код, моделируется блок-схема в среде разработки «Quartus II»:

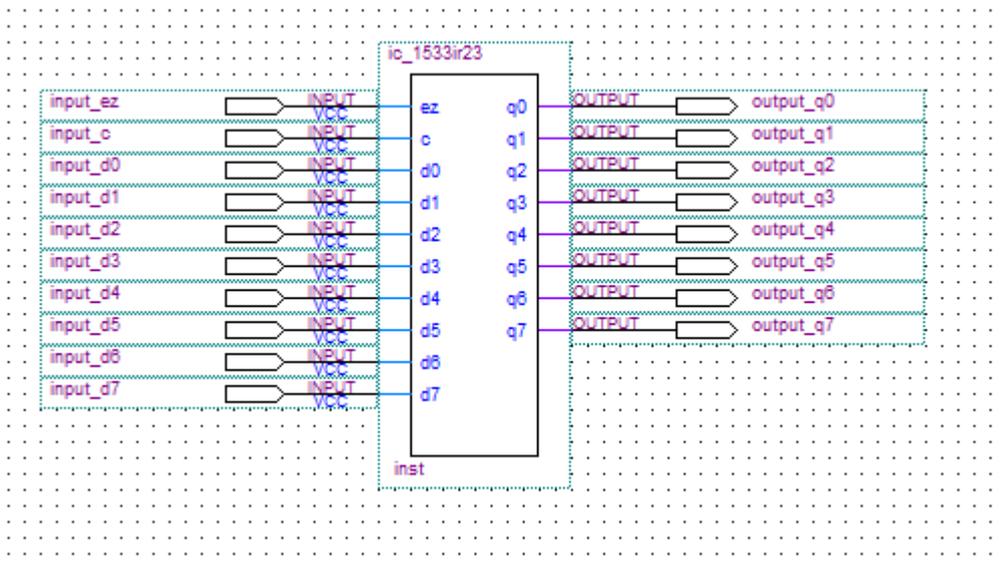


Рисунок 19 Блок-схема 1533ИР23

Финальным шагом будет проектирование тестовой последовательности в «SimTest» для проверки правильности логики работы элемента:

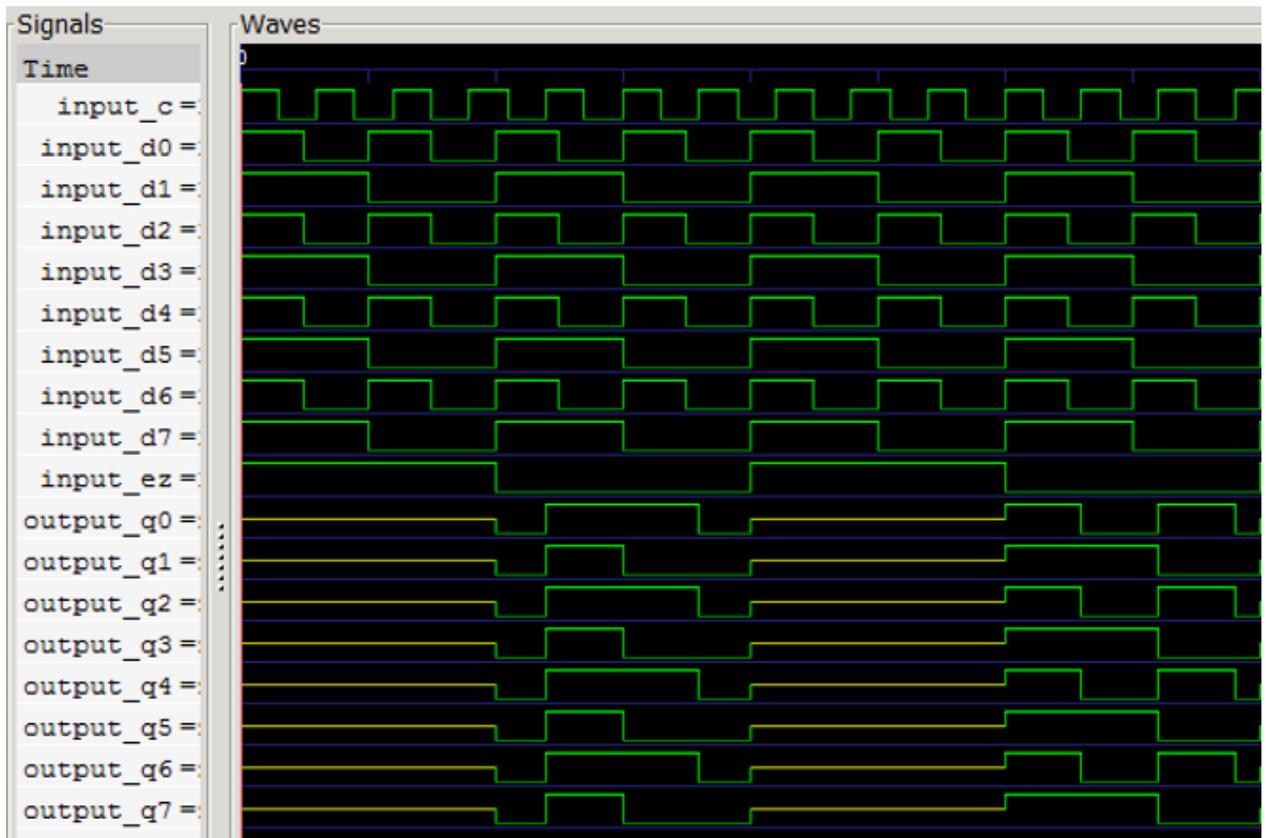


Рисунок 20 Результаты теста 1533IP23

По результатам моделирования убеждаемся, что логика элемента описана верно.

1533ИЕ7

Данная схема – это четырехразрядный двоичный реверсивный счетчик синхронного типа. Для установления выходов в исходное состояние необходимо подать на вход R положительный импульс. Если подать отрицательный сигнал на вход стробирования, то происходит прямая запись сигналов с информационных входов на выходы. При прямом счете необходимо на вход «-1» подать высокий уровень, а на «+1» положительные импульсы, схема работы счетчика для обратного счета аналогична. При заполнении счетчика на выходе прямого переноса появляется отрицательный импульс[20].

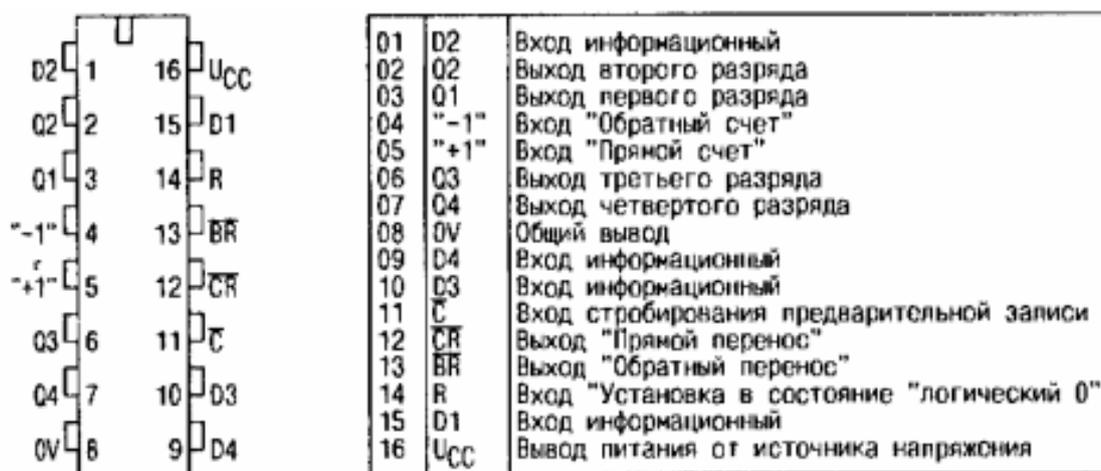


Рисунок 21 Расположение и назначение 1533ИЕ7

Схема имеет 8 входов и 6 выходов.

Режим работы	Вход			
	R	С	"+1"	"-1"
Установка в состояние "логический 0"	H	X	X	X
Запись информации	L	L	X	X
Неактивное состояние	L	H	H	H
Счет прямой	L	H	┌	H
Счет обратный	L	H	H	┌

Рисунок 22 Таблица истинности 1533ИЕ7

Фрагмент кода, описывающий алгоритм работы:

```
always@(negedge C or posedge R or posedge Summ)
begin
  if(R == 1'b1)
    q <= 4'b0000;
  else
    if(C == 1'b0)
      q <= {D1, D2, D3, D4};
    else
      if(Summ == 1'b1)
        begin
          if(Plus == 1'b0 && Minus == 1'b1)
            q = q + 1'b1;
          else
            if(Plus == 1'b1 && Minus == 1'b0)
              q = q - 1'b1;
        end
      end
end
```

Рисунок 23 Фрагмент кода 1533ИЕ7

Полученная после компиляции блок-схема:

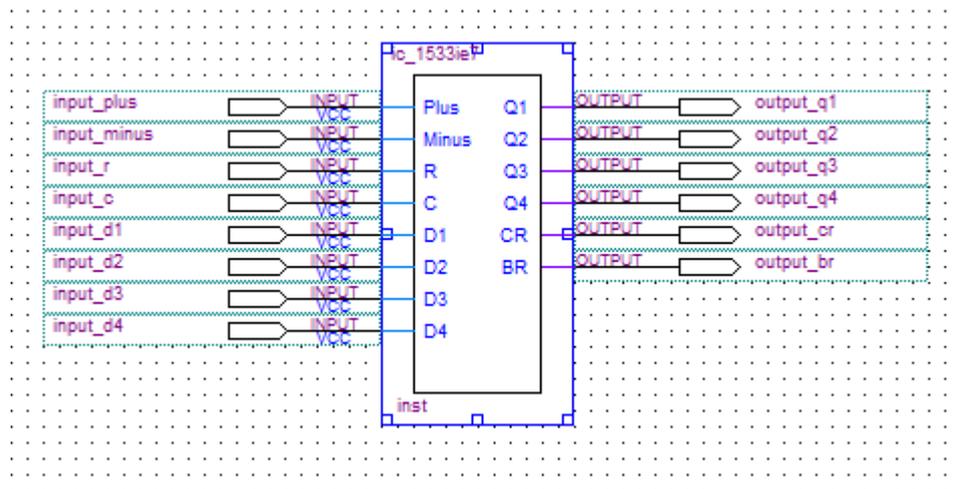


Рисунок 24 Блок-схема 1533ИЕ7

Как и в предыдущих примерах, на основе блок-схемы пишется тест и проверяется правильность работы модели в «SimTest»:

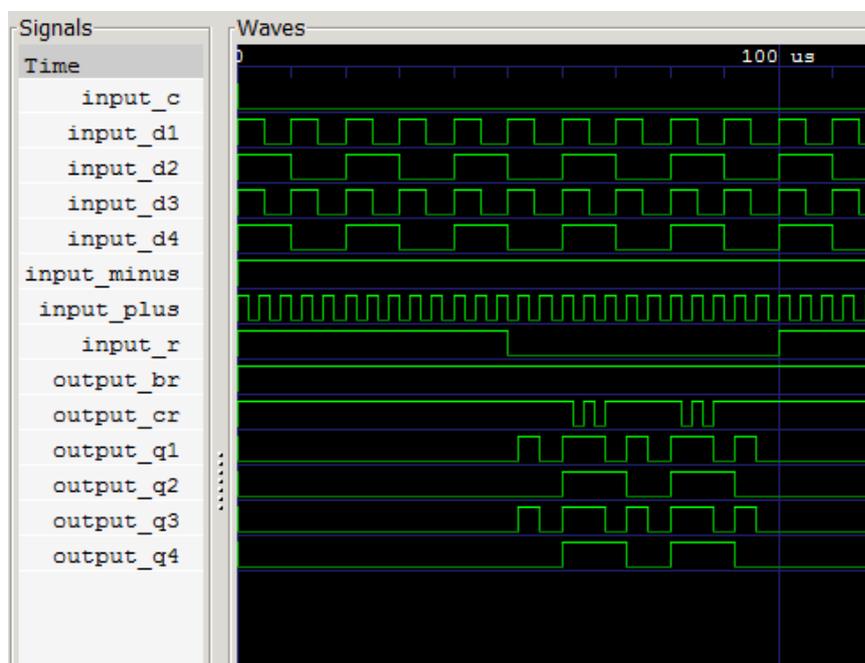


Рисунок 25 Результаты теста 1533ИЕ7

На рисунке показан результат проверки прямого счета, аналогичным образом получается обратный счет. Схема реализована правильно.

Программное моделирование и написание теста сложного радиоэлектронного устройства Субблок ПЭЗ

Субблок ПЭЗ состоит из 18 различных элементов: 530ИЕ17, 530ЛА2, 530ЛА16, 530ЛЕ1, 530ЛЛ1, 530ТВ9, 533ТВ6, 1533ИД3, 1533ИЕ7, 1533ИР23, 1533ИР31, 1533ЛА3, 1533ЛА4, 1533ЛИ1, 1533ЛН1, 1533ЛП5, 1533ТМ2, а также содержит 134 внутренние связи.

Как упоминалось ранее, для построения программной модели необходимо отдельно создать каждый элемент данной схемы, затем объединить их в общем проекте:

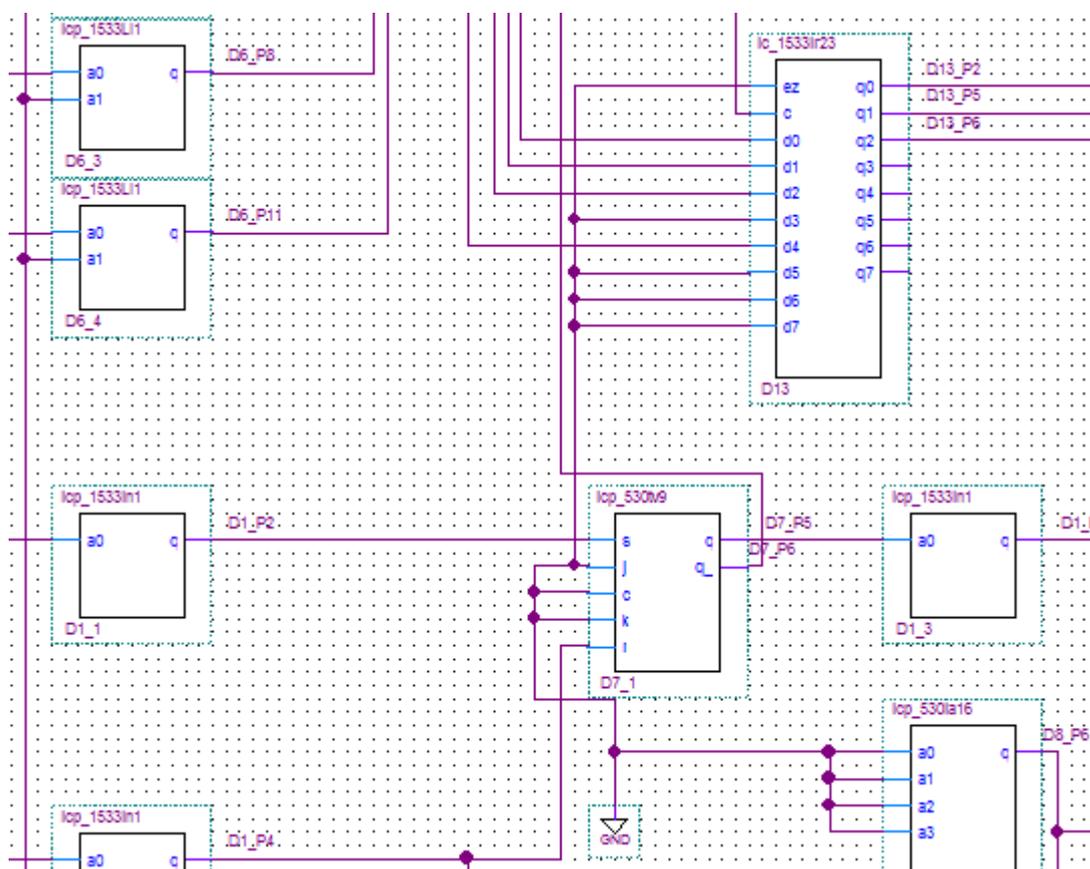


Рисунок 26 Фрагмент блок-схемы Субблок ПЭЗ

Далее, как и в предыдущих примерах, мы компилируем блок-схему, затем преобразуем ее в HDL-файл для дальнейшей работы в «SimTest».

Необходимо добиться максимального покрытия, чтобы проверить каждый компонент устройства, поэтому создаем новый проект и приступаем

к написанию скрипта, с помощью которого шаг за шагом будет моделироваться последовательность тестовых воздействий.

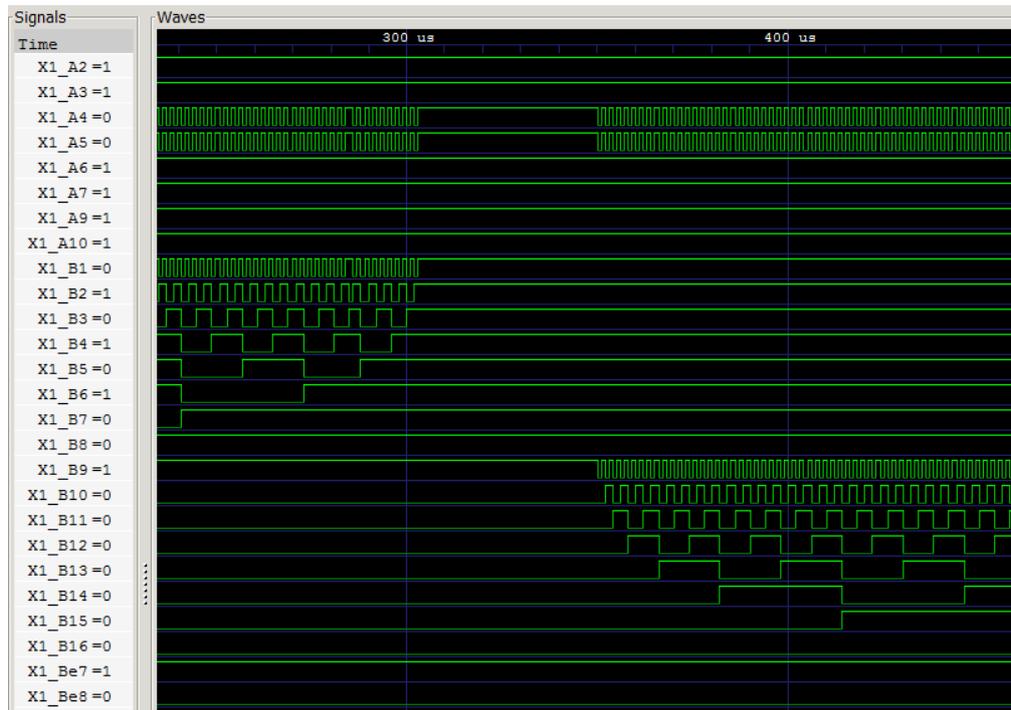


Рисунок 27 Фрагмент результата теста Субблок ПЭЗ

Получаем анализ покрытия:

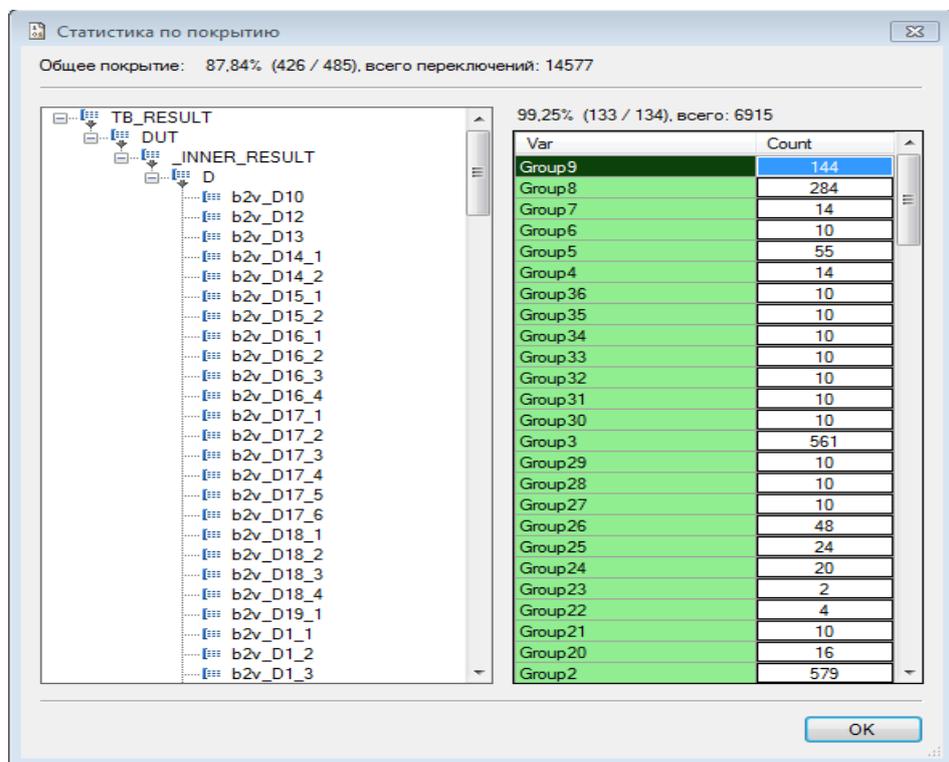


Рисунок 28 Анализ покрытия Субблок ПЭЗ

Таким образом, мы получили покрытие, близкое к 100%. Один выход не изменился, потому что все входы элемента подключены к источнику постоянного сигнала:

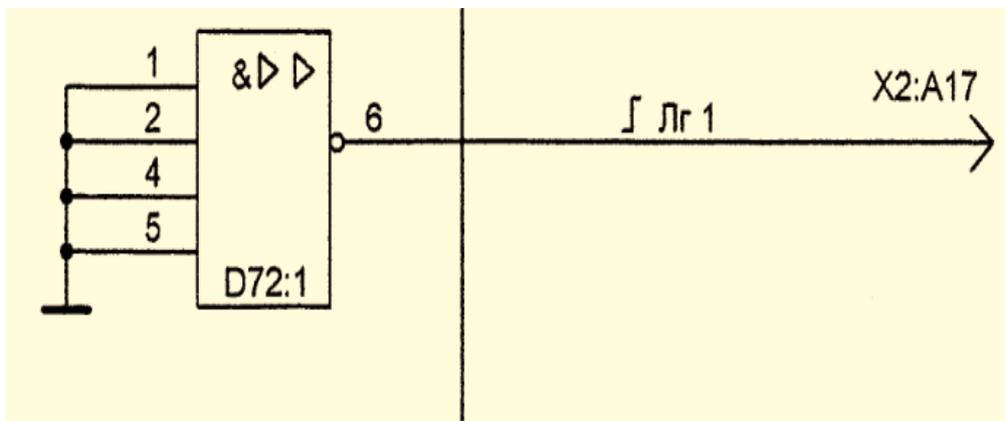


Рисунок 29 Элемент с постоянным сигналом на входах

Стоит заметить, что 59 из 485 внутренних сигнальных линий не изменили своего состояния, потому что подключены к источникам постоянного сигнала. Таким образом, покрытие в 87,84% является полным для объекта контроля.

Выводы

1. Были написаны и протестированы программные модели для всех 18 различных типов микросхем, входящих в состав Субблок ПЭЗ
2. Создана программная модель Субблок ПЭЗ в среде разработки «Quartus II».
3. С помощью САПР «SimTest» смоделирован тест для проверки работоспособности объекта контроля, обладающий полным покрытием.

Таким образом, были выполнены все поставленные задачи и достигнута цель работы.

Заключение

В данной работе рассмотрена история развития схемотехники, а также описаны и проанализированы современные методы автоматического тестирования.

Описанные в работе алгоритмы написания программы для функционального тестирования были применены для создания программной модели и разработки тестовой последовательности для контроля работоспособности цифрового радиоэлектронного устройства Субблок ПЭЗ.

Литература

1. Городецкий А. Снова о внутрисхемном тестировании ICT // Компоненты и технологии. 2011. №7. С. 58–59.
2. 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, 1. P. 381–401.
3. Holtzer M. In-circuit pin testing: An excellent potential source of value creation // SMT Surface Mount Technology Magazine, 2015, 30 (6). P. 68–71.
4. Renbi A., Delsing J. Application of Contactless Testing to PCBs with BGAs and Open Sockets // Journal of Electronic Testing: Theory and Applications, 2015, 31 (4). P. 339–347.
5. Renbi A., Delsing J. Contactless Testing of Circuit Interconnects // Journal of Electronic Testing: Theory and Applications, 2015, 31 (3). P. 229–253.
6. 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, 55. P. 453–458.
7. Елаев Е. В. Интерфейсный метод автоматизированной генерации тестовых воздействий для цифровых радиоэлектронных объектов контроля // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2015. № 4. С. 19–24.
8. Sangi R., Baranski M., Oltmanns J., Streblov R., Müller D. Modeling and simulation of the heating circuit of a multi-functional building // Energy and Buildings, 2016, 110. P. 13–22
9. Fujita M., Taguchi N., Iwata K., Mishchenko A. Incremental ATPG methods for multiple faults under multiple fault models //

Proceedings – International Symposium on Quality Electronic Design, 2015, art. no. 7085420. P. 177–180.

10. Kochte M. A., Elm M. b , Wunderlich H.-J. Accurate X-propagation for test applications by SAT-based reasoning // IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems, 2012, vol. 31, 12, art. no. 6349431. P. 1908–1919.

11. 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, серия ОТ. 2013. № 1. С. 80–88.

12. Гусев О. А., Елаев Е. В., Мащинский Н. С., Нуракунов А. Автоматизация генерации тестовых воздействий для комбинационных цифровых схем // Процессы управления и устойчивость. 2016. Т. 3. № 1. С. 389–393.

13. Мельник В. И., Гришкин В. М., Михайлов А. Н., Овсянников Д. А. Методика разработки тест-программ контроля и диагностики цифровых устройств с использованием САПР «SimTest» // Электроника: Наука, технология, бизнес. 2013. № S (128). С. 118–124.

14. Елаев Е. В., Степанов Ю. Л., Ферсенков В. В. Подходы к моделированию микропроцессоров для построения контрольно-диагностических тестов // Процессы управления и устойчивость, 2015. Т. 2. № 1. С. 398–403

15. Мащинский Н. С., Елаев Е. В., Федюкович П. А. Моделирование сложных цифровых устройств с целью их тестирования // Процессы управления и устойчивость. 2015. Т. 2. № 1. С. 452–457.

16. <http://www.datasheet-pdf.ru/1533/1533pdf/1533LA2.pdf>

17. <http://www.datasheet-pdf.ru/1533/1533pdf/1533ID3.pdf>

18. <http://www.datasheet-pdf.ru/1533/1533pdf/1533TM2.pdf>

19. <http://www.datasheet-pdf.ru/1533/1533pdf/1533IR23.pdf>

20. <http://www.datasheet-pdf.ru/1533/1533pdf/1533IE7.pdf>