

Санкт-Петербургский государственный университет

ГУСЕВ Олег Александрович

Выпускная квалификационная работа

**Построение последовательности тестовых сигналов для комбинационных
цифровых схем и схем с памятью**

Уровень образования: Магистратура

Направление *03.04.01 «Прикладные математика и физика»*

Основная образовательная программа *ВМ.5521.2017 «Математические и
информационные технологии»*

Научный руководитель:

Профессор кафедры ТСУЭФА,
заведующий кафедрой ТСУЭФА,
доктор физ.-мат. наук

Овсянников Д. А.

Рецензент:

Ведущий инженер, АО «Научно-
исследовательский институт
точной механики»,

кандидат технических наук

Антонов В. В.

Санкт-Петербург

2019

Оглавление

Введение.....	4
Постановка задачи.....	6
Глава 1. Развитие электроники и схемотехники.....	8
1.1 Первое поколение электроники. Отправная точка развития.....	8
1.2 Второе поколение электроники. Разработка и совершенствование электровакуумных приборов.	9
1.3 Третье поколение электроники. Кремниевый век.....	10
1.4 Четвертое поколение электроники. Интегральные схемы	10
Глава 2. Обзор современных методов решения задачи.....	14
2.1 Визуальный автоматизированный контроль.....	14
2.2 Внутрисхемное тестирование	16
2.3 Периферийное/граничное сканирование.....	17
2.4 Функциональное тестирование	18
Глава 3. Построение программной модели	22
3.1 Язык описания логики элементов	22
3.2 Проектирование цифрового устройства с помощью мультиплатформенной среды проектирования Altera Quartus II.....	23
3.3 Система автоматизированного проектирования «SimTest»	24
Глава 4. Составление программной модели объекта контроля.....	26
4.1 Моделирование компонентов	26
4.2 Электронная модель схемы.....	41
Глава 5. Тестирование.....	44
5.1 Привязка логических интерфейсов	45
5.2 Генерация теста.....	46

5.2 Анализ покрытия.....	47
Выводы	48
Заключение	49
Список литературы	50

Введение

В 1947 году учёные Джон Бардин, Уильям Брэдфорд Шокли и Уолтер Хаузер Браттейн изобрели транзистор — основной элемент микроэлектроники. Данное открытие имело революционный характер для дальнейшего развития радиоэлектроники [1-3].

Первыми данным изобретением воспользовались производители слуховых аппаратов. Так в 1953 году был создан первый транзисторный слуховой аппарат. Военные так же активно использовали данный прибор, он помогал уменьшить вес и габариты радиоэлектронных приборов [4].

В 1960 году появилась первая микросхема — чаще всего типовая, электронная схема, выполненная на едином кристалле. После этого события радиотехника и цифровая техника начали входить во все сферы человеческой жизни. Сложно представить нашу жизнь без смартфонов, ноутбуков, автомобилей – все эти предметы включают в себя различные микросхемы. Однако, как и в любой сфере серийного производства, при изготовлении цифровых устройств присутствует брак, поэтому для конвейерного производства необходима быстрая и качественная диагностика каждого компонента устройства.

Поиск неисправных элементов еще полностью не автоматизирован и является актуальной проблемой на сегодняшний день. Существует несколько подходов выявления неисправностей, в данной работе будет использована функциональная система тестирования, которая включает систему автоматизированного проектирования (САПР) тестов «SimTest» (САПР, разработанный в СПбГУ), «Yastek» и аппаратно-программный комплекс тестового контроля и диагностики УТК-512. Данные системы помогают произвести не только качественную, но еще и достаточно быструю диагностику [5-19]

Цифровое устройство — техническое устройство или приспособление, состоящее из набора электронных модулей с заданным алгоритмом работы, которое предназначено для получения, обработки и хранения информации в цифровой форме, используя цифровые технологии [20]. Под тестом устройства понимают вектор входных и соответствующих им выходных данных, который проверяет компоненты устройства, а также корректность связей между ними. После создания тест загружается в устройство тестового контроля и диагностики и используется для отбраковывания неисправных объектов контроля.

В данной работе представлен алгоритм проведения функционального тестирования на реальном объекте.

Постановка задачи

Задачей данной работы является построение тестовых взаимодействий на устройство тестового контроля проверки его работоспособности.

В исследовании предназначение и функциональность самого устройства неизвестны, нам дана лишь графическая схема устройства, название элементов, входящих в ее состав, и непосредственно сама плата (совокупность плат). Таким образом заказчик может сохранить особенность функционирования своего устройства, а также передавать на проверку только отдельные его части. Из чего следует, что ведется проверка работоспособности устройства, его компонент и корректность связей между ними.

Для выполнения работы необходимо:

1. Разработать программную модель всех элементов данного устройства:
 - a. Описать логику работы элемента на языке описания аппаратных средств. В работе будет использоваться язык Verilog HDL.
 - b. Создать графическую модель элемента. В качестве среды разработки будет использоваться ALTERA QUARTUS II.
 - c. Произвести проверку работоспособности полученной программной модели. Проверка будет производиться в САПР «SimTest».
2. Создать программную и графическую модель самого устройства контроля:
 - a. Произвести «сборку» модели устройства, согласно графической схеме устройства, которая показывает связь между объектами (используется ALTERA QUARTUS II).

3. Осуществить тестовый контроль работоспособности объекта исследования, путем создания проверяющей тестовой последовательности.

Глава 1. Развитие электроники и схемотехники

Электроника — это область наук, которая изучает взаимодействие частиц с электронными полями и занимается созданием и практическим использованием различных устройств и приборов, используемых в основном для передачи, обработки и хранения информации [21].

Схемотехника — это научно-техническая область, которая занимается проектированием, созданием и отладкой различных радиоэлектронных устройств.

Историю развития электроники разделяют на 4 этапа:

1.1 Первое поколение электроники. Отправная точка развития.

Началом развития ламповой электроники принято считать изобретение электрической лампочки выдающимся русским ученым Лодыгиным А. Н. в 1873 году. Данное открытие помогло Эдисону Т. А. обнаружить явление термоэлектронной эмиссии и описать прохождение тока через вакуум [22-23].



Рис. 1: Лампа Лодыгина

Однако важнейшую роль в развитии электроники и схемотехники сыграло создание устройства передачи и приема звуков на расстоянии при помощи электронномагнитных волн (радио) в 1895 г. ученым А. С. Поповым. Именно в этот момент появился спрос на электронные приборы [24].

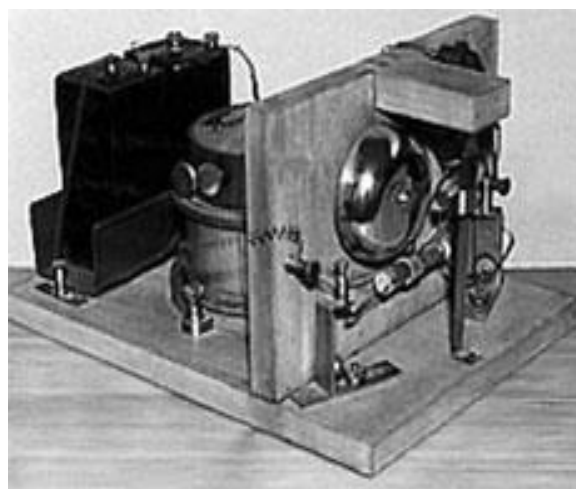


Рис. 2: Радио

1.2 Второе поколение электроники. Разработка и совершенствование электривакуумных приборов

В 1904 г. была изобретена первая двухэлектродная электронная лампа — диод, которая представляет собой вакуумную электронную лампу с двумя электродами - анодом и катодом. Принцип ее работы был основан на трудах Т. А. Эдисона и состоял в том, что ток мог проходить лишь в одном направлении. На практике диод был применен в качестве детектирования электрических колебаний в радиотехнике.

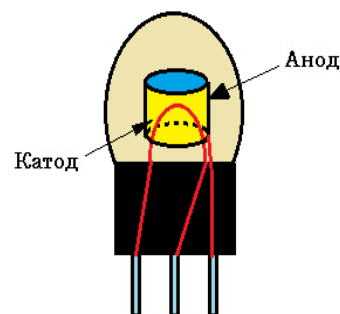


Рис. 3: Диод

Спустя несколько лет в 1907 г. была изготовлена трехэлектродная лампа. Основным ее отличием от диода являлось наличие между катодом и анодом третьего электрода, выполненного в виде проволочной спирали, который называется сеткой. 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. Используется в качестве усилителя и генерации электрических колебаний различной частоты.

Далее стали появляться четырехэлектродные (тетроды) и пятиэлектродные (пентоды) лампы, разрабатывались лампы для дециметровых и сантиметровых волн, инженеры пытались уменьшить размеры ламп, снизить энергопотребление и улучшить охлаждение.

Эра ламповой техники продлилась недолго. В 50-ые годы началось бурное развитие полупроводниковых элементов, однако есть и такие области, где до сих пор используют ламповую технику. Например, высококачественные High-End усилители звука собираются исключительно на ламповых элементах [22-23].

1.3 Третье поколение электроники. Кремниевый век

Развитию полупроводниковой электроники послужили достижения в области физики твердого тела, также большое влияние оказали теоретические и экспериментальные исследования в области электрических свойств проводников академиком Иоффе А. Ф.

В 1947 г. появляется третье поколение электроники, созданной на базе полупроводников. Д. Бардин, У. Баттейн и У. Шокли создали полупроводниковый трехэлектродный усилитель (транзистор) — полупроводниковый прибор, который усиливает, преобразует и генерирует электрические колебания. Таким образом начался новый виток электроники, так называемый «кремниевый век». Новые устройства потребляли значительно меньше электроэнергии, стали более миниатюрными и доступными [22-23].



Рис. 4: Транзистор

1.4 Четвертое поколение электроники. Интегральные схемы

Производители различными способами пытались уменьшить устройства. Изначально размер плат уменьшали методом микромодулей: электронные модули печатались на тонких керамических пластинах, а затем спаивались вместе - таким образом получали нужную микросхему.

Американский учёный Джек Килби решил изготовить полупроводниковые элементы на одной пластине. Первая такая микросхема была выпущена в 1959 г. благодаря этому изобретению микросхемы стали более легкими и миниатюрными, а также увеличилось время быстродействия, повысилась надежность и снизилось энергопотребление.

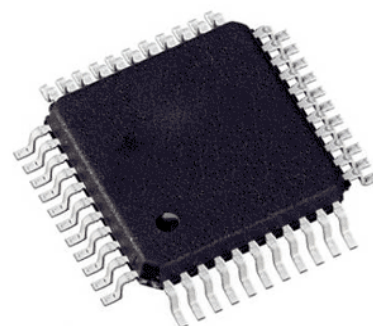


Рис. 5: Интегральная схема

Первые схемы имели малую плотность, однако сейчас можно встретить микросхемы, у которых количество элементов на кристалл превосходит 1 миллиард (гигабольшая интегральная схема) [25].

Стоит упомянуть процессоры:

За последние 40 лет скорость компьютеров растет экспоненциально: у сегодняшних CPU тактовая частота в тысячу раз выше, чем у их братьев начала 80-ых, объем оперативной памяти на компьютере вырос в десять тысяч раз, а ёмкость жёсткого диска увеличилась более чем в сто тысяч раз. Что соответствует закону Мура — количество транзисторов, размещаемых на кристалле интегральной схемы, удваивается каждые 24 месяца. Так в 2012 году размер резистора процессоров Intel достигал 22 нанометра, а уже сейчас стал возможен переход на 7 нанометров благодаря успехам полупроводниковых гигантов Global Foundries и TSMC. Однако не стоит забывать главное, существует предел — размер атома кремния равен 0,2 нанометра. На данный момент ученые пытаются отойти от кремния в пользу графена — самого тонкого проводника электрического тока, который при этом очень прочен. Когда случится переход пока не известно, по мнению AMD — мирового гиганта по производству компьютерных комплектующих, в ближайшие десять лет основным материалом для процессоров останется кремний и будут осваиваться нормы в 5 и 3 нанометров [26-27].

Затрагивая тему компьютеров невозможно не рассказать о памяти, рассмотрим постоянное запоминающее устройство (ПЗУ) и оперативное запоминающее устройство (ОЗУ), которые так же используются в объекте контроля [29-34].

ПЗУ:

Постоянное запоминающее устройство — устройство памяти, используется для хранения неизменяемых данных. Является энергонезависимым [29-31].

Основным узлом запоминающего устройства (ЗУ) является накопитель, имеющий матричную структуру. Функция элементов памяти накопителя здесь выполняется программированием перемычек (полупроводниковых диодов или транзисторов, включенных между строками и столбцами матрицы). При отсутствии перемычки в данной ячейке хранится ноль, а при ее наличии единица. Таким образом память не требует питания для хранения данных [29].

ПЗУ бывает следующих видов:

- ROM — постоянное запоминающее устройство (невозможно изменить данные). Еще его называют масочным, поскольку данные заносятся при изготовлении микросхем с помощью маски (шаблона) на завершающем этапе технологического процесса [31].
- PROM — программируемое ПЗУ. Изначально это устройство имеет плавкие перемычки для всех ячеек (т.е. все ячейки памяти имеют значение один), изготовитель сам выбирает какие из них обходимо удалить (расплавить). Этот процесс называется программированием ПЗУ. Стоит отметить, что перемычки невозможно восстановить, таким образом программирование ПЗУ возможно лишь один раз [34].
- EPROM — перепрограммируемое ПЗУ. Отличительной особенностью данного запоминающего устройства является окошко из кварцевого стекла, через которое при необходимости можно стереть данные посредством сильного ультрафиолетового света [34].
- EEPROM — электрически стираемое перепрограммируемое ЗУ. Память в этом устройстве может стираться и перезаписываться большое количество раз [31].
- Магнитоэлектронные запоминающие устройства — электронное устройство, в которых на доменном уровне используются электромагнитные процессы [30].

ОЗУ

Оперативное запоминающее устройство — устройство памяти с произвольным доступом, запись информации производится довольно просто и может производиться пользователем огромное количество раз (ограничено только временем жизни микросхемы). Информация в памяти пропадает при выключении её питания [35].

Выделяют следующие виды современных ОЗУ:

- DRAM — динамическая память с произвольным доступом. Для хранения значения в ячейке разработана схема, состоящая из одного конденсатора и одного (иногда двух) транзистора. Такая конструкция позволяет сделать память более миниатюрную и дешевую, однако время работает медленнее SRAM (состояние триггера меняется быстрее, чем конденсатора), а также требует постоянной подзарядки конденсаторов [32].
- SRAM — статическая память с произвольным доступом. Благодаря использованию триггеров эта память обладает высокой скоростью, однако такое решение существенно увеличило ее стоимость и размер [33].

Глава 2. Обзор современных методов решения задачи

С развитием электронной промышленности производители столкнулись со следующей вполне ожидаемой проблемой: быстрого и эффективного тестирования готовой продукции. Сейчас выделяют следующие виды тестирования:

- визуальный автоматизированный контроль
- внутрисхемное тестирование
- периферийное сканирование
- функциональное тестирование

2.1 Визуальный автоматизированный контроль

Под данным видом контроля подразумевают автоматический оптический — АОІ (используется для выявления «видимых» дефектов) и рентгеноскопический контроль — АХІ (используется для выявления «невидимых» дефектов, например, качества пайки). Первые системы технического зрения стали появляться в начале 80-х годов прошлого века. Данный метод тестирования используется на многих этапах производства для определения качества «сборки» изделий.

Использование оборудования, после которого используется автоматический оптический и рентгеноскопический контроль:

- трафаретный принтер или автомат диспенсорного нанесения паяльной пасты
- установщик чипов
- установщик компонентов с малым шагом выводов
- печь оплавления припоя
- установка для монтажа компонентов в отверстия / установка пайки волной

Благодаря использованию данного вида тестирования можно избежать большого количества неисправностей (Таблица 1) [36-37].

Тип дефекта	После принтера трафаретной печати	После установки компонентов	После операции оплавления припоя	После операции пайки волной
Недостаточное нанесение паяльной пасты	X	X		
Блокированные отверстия	X	X		
Короткое замыкание (паста)	X	X		
Смещение пасты	X	X		
Пропущенный компонент		X	X	X
Перекошенный компонент		X	X	X
Полярность		X	X	X
Смещение компонента		X	X	X
Установлен не тот компонент		X	X	X
Отогнутый или сломанный вывод		X	X	X
Избыточное количество пасты			X	X
Вздыбливание компонентов			X	X
Непропаянное соединение			X	X
Короткое замыкание после операции пайки			X	X

Таблица 1: Виды дефектов при печати

Стоит отметить, что при использовании визуального автоматизированного контроля нет необходимости содержать большой штат сотрудников, однако оборудование достаточно дорогое. В связи с этим каждая

организация сама рассчитывает риски появления брака и выбирает этапы, на которых будет проходить такой вид тестирования.

Основным недостатком визуального автоматизированного тестирования является невозможность определить работоспособность элементов, входящих в состав устройства, поэтому данный вид контроля зачастую используется в тандеме с другими.

2.2 Внутрисхемное тестирование

Данный вид тестирования часто называют игольчатым, поскольку он подразумевает присоединение платформ к отдельным частям платы посредством контактных пробников похожих на иголки: стационарного поля («ложе гвоздей») или «летающих» щупов / матриц. Благодаря

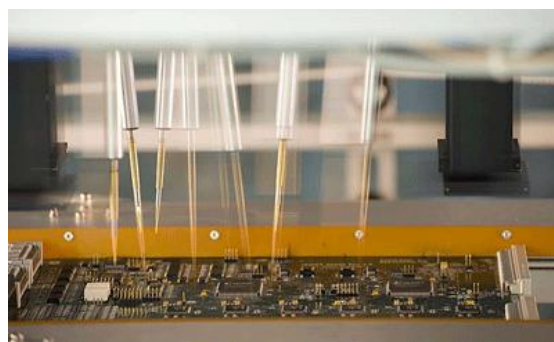


Рис. 6: Тестирование «летающими щупами»

внутрисхемному тестированию проверяются отдельные компоненты или электрические соединения на плате. Таким образом, выявляются дефекты связей (разрывы и замыкания дорожек), а также дефекты пайки или неисправные компоненты, таким образом можно провести тестирования без их демонтажа и проверять объект не подавая питание на все устройство, что уменьшает риск испортить тестируемый объект. Так же существует тандем этого способа с функциональным (будет описан дальше), в этом случае происходит тестирование функциональности отдельных частей платы [38 -39].

Внутрисхемное тестирование постепенно теряет свою актуальность, поскольку современные производители всеми способами пытаются уменьшить размер устройств: создают многоуровневые платы, располагают компоненты слишком близко друг к другу, а также блокируют доступ к разъемам для защиты от влаги и пыли.

2.3 Периферийное/граничное сканирование

Периферийное сканирование — это внутрисистемное тестирование, суть которого заключается в тестировании платы с микросхемами стандарта IEEE 1149 (тестовая логика входит в само устройство) через специальный разъем с помощью специализированного тестового оборудования. Часто этот метод называют JTAG в честь компании, которая занималась его созданием. Как говорилось ранее, в состав устройства должны входить микросхемы, поддерживающие стандарт IEEE 1149, тогда можно тестировать не только эти микросхемы и связи между ними, но и некоторые другие элементы: резисторы, логику, память.

Сейчас огромное количество микросхем поддерживают нужный формат: микросхема уже содержит избыточную тестовую логику. Разработкой тестового оборудования занимается множество компаний, уже можно не только генерировать тестовые программы, но и производить оценку тестового покрытия устройства, что помогает еще в процессе создания устройства добиваться высокой тестопригодности. Также следует отметить, что JTAG сканирование можно проводить на системном уровне: несколько объединенных в одну системную плату устройств, могут иметь общую JTAG шину и тестироваться в составе сложного изделия через один единственный разъем [40-44].

Тестовый контроль происходит по следующему плану:

- подключение объекта тестирования
- подача на вход последовательности тестовых значений
- анализ полученных на выходе результатов
- вынесения результатов проверки

Конечно, тестирование устройств без «долгой» подготовки, обеспечивающее поиск повреждения дорожек, отпада элементов, коротких

замыканий, нерабочих микросхем, делает этот способ контроля достаточно актуальным, но стоит помнить, что не все микросхемы поддерживают такой вид сканирования, что может повлиять на тестопригодность, нет возможности определить неисправности, являющиеся той или иной функцией времени, а также наличие включенной тестовой логики увеличивает не только цену, но и размер самого устройства.

2.4 Функциональное тестирование

Тестирование, основанное на проверке функциональности и спецификационных параметров целого устройства или его компонентов. Может проводиться в ручном и автоматическом режиме (в основном оператору требуется лишь подключить устройство и на основе полученных данных произвести контроль годности). В процессе тестирования создается программная модель самого устройства, т.е. имитируется фактическая работа его схем, после чего создаются тестовые последовательности, которые имитируют работу устройства, и на основе «ответов» устройства дается заключение о работоспособности.

Существует несколько способов создания тестовой последовательности для данного вида тестирования [45]:

- ручной способ;
- псевдослучайные числа;
- алгоритмические методы.

Ручной способ — в данном случае тестовая последовательность создается вручную, т.е. разработчику необходимо вручную «придумывать» сигналы для изменения состояния каждого компонента.

Основным минусом данного метода является время разработки, однако на сегодня он остается самым популярным методом.

Псевдослучайные числа — в данном случае создание тестовой последовательности происходит посредством генерации псевдослучайных величин.

Данный метод создания тестовой последовательности несомненно является довольно быстрым, однако такой подход в большинстве случаев не дает высокого тестового покрытия.

Алгоритмические методы — как может быть понятно из названия данный метод поиска тестовой последовательности основан на генерации паттернов с помощью какого-либо из алгоритмов, которые действуют по следующей схеме:

1. выбор неисправности, которую необходимо обнаружить;
2. поиск последовательности, которая установит данный узел в состояние неисправности;
3. распространение сигнала «назад» до краевых разъемов.

Актуальным на данный момент является алгоритм FAN.

Минусом такого подхода можно назвать время разработки, а также потребность в значительных вычислительных ресурсах. При разработке тестового вектора таким способом в некоторых сложных схемах использование таких алгоритмов более неосуществимо и непрактично [46].

Стоит отметить использование нейронных сетей для таких целей. Данные виды программ не только используют генерацию псевдослучайных величин и алгоритмические подходы для проверки отдельных компонентов, но и «вспоминают» подобные части из уже протестированных схем для более быстрого построения тестовой последовательности [47].

Данный вид построения тестового вектора кажется очень перспективным, однако все еще не существует «хороших» нейронных сетей, которые могут справиться со схемами любой сложности.

Основным преимуществом функционального тестирования является хорошее тестовое покрытие и короткое время отбраковывания изделий. Однако затраты как по времени, так и по ресурсам на разработку теста иногда заставляют производителя отказаться от данного вида тестирования.

На данный момент многие компании предлагают свою аппаратуру и для такого вида тестирования. Рассмотрим некоторые из них.

ТЕСТ-6408

Данный прибор представляет собой систему контроля цифровых узлов и блоков бортовой аппаратуры. Он включает в себя 512 независимых каналов и на его базе можно проводить разнообразные алгоритмы контроля цифровых устройств: статическое тестирование (позволяет вручную формировать логические уровни на выходах и контролировать реакцию тестируемого объекта путем регистрации логических уровней на входных разъемах), функциональный динамический контроль (прибор позволяет запускать тестовые вектора для проверки выполнения объектом контроля своего функционала, есть возможность сравнивать выходные вектора как с объектом контроля, так и с его эмуляцией), логический анализ (позволяет реализовывать алгоритмы асинхронного и синхронного анализатора логических состояний), сигнатурный анализ (проверяет тестовыми кодами, которые записаны в ОЗУ вывода) [48].

Как можно заметить, данный прибор обладает «хорошей подготовкой» для функционального тестирования, однако его главным недостатком является зависимость между программным обеспечением и диагностирующим оборудованием (можно использовать только софт производителя). Здесь так

же стоит отметить комплексы фирмы Teradyne, Flex и др, которые обладают похожими возможностями, но содержат большой запас готовых компонентов.

Данная работа подразумевает тестирование на аппарате «УТК-512», которую так же можно отнести к классическим приборам функционального тестирования, однако ее использование не ограничивает разработчика своим программным обеспечением.

FT-17 IR – тестовая система функционального контроля с интегрированным тепловизором.

Данная система содержит «классический» алгоритм функционального тестирования: создается программная модель объекта контроля, строится тестовая последовательность, однако здесь также прописываются эталонные или рассчитанные температурные значения компонентов [49]. Таким образом при тестировании тепловизор считывает температурные параметры каждого компонента, выявляя компоненты с резким перепадом температур или большим отклонением от заданных показателей. Это позволяет определять неблагоприятные места и снижать вероятность будущих поломок, так как резкое изменение температуры в большей степени влияет на их рост.

Как можно увидеть, данный комплекс может не только найти, но еще и предугадать проблему, однако его стоимость, «нераспространенность» ПО зачастую заставляют производителя отказаться от данного прибора.

Глава 3. Построение программной модели

3.1 Язык описания логики элементов

Язык описания аппаратуры (HDL) — язык программирования, используемый для описания структуры и поведения электронных схем. Существует два основных HDL языка: Verilog HDL (создан Gateway Design Automation в 1985 г.) и VHDL (разработан по заказу Министерства обороны США в 1987 г.) [50-51]. Изначально данные языки использовались для документирования и моделирования решений, сделанных в виде бумажных схем, введение HDL позволило работать на более высоком уровне абстракции и упростило моделирование больших проектов. Стоит отметить, что они имеют некоторые ограничения: не подходят для смешанной цифро-аналоговой и полностью аналоговой симуляции, не имеют языковых конструкций для описания рекурсивно-генерируемых логических структур, в связи с этим, они не могут полностью вытеснить другие языки описания аппаратуры.

В данной работе будет использоваться Verilog HDL. Такой выбор можно объяснить следующим:

- данный язык более распространен: многие знакомы с Verilog HDL и не сталкивались с VHDL, — это поможет лучше разобраться в работе, так же существует большое количество литературы (в том числе русскоязычной) и примеров;
- простотой в освоении, имеет большую схожесть с языком C;
- возможностью использования оборудования и программ, предоставляемых кафедрами;
- моим опытом в разработке программных моделей на этом языке.

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

Система автоматизированного проектирования (САПР) — система, которая предназначена для автоматизации процесса проектирования, включающая в себя персонал, а также технических, программных и других средств автоматизации его деятельности.

В данной работе под проектированием понимается создание программной модели существующего цифрового устройства. Для этой цели была использована среда разработки Quartus II компании Altera. Выбор данной среды обусловлен тем, что она удовлетворяет всем необходимым условиям [20]:

- существует бесплатная версия (использована Version 9.0);
- поддерживает язык Verilog HDL;
- имеет удобный интерфейс;
- является одной из ведущих САПР в мире (на данный момент выкуплена INTEL), в связи с этим существует огромное количество соответствующей литературы.

План создания программной модели цифрового устройства:

1. по каждому элементу, входящему в состав устройства, найти документацию, содержащую принцип его работы;
2. создать программную модель, на языке Verilog HDL каждого элемент (в качестве компилятора можно использовать IcarusVerilog);
3. создать поведенческую модель самого цифрового устройства

Подробное описание с примерами будет изложено в следующей главе.

3.1 Система автоматизированного проектирования «SimTest»

Для построения тестовой программы будет использоваться САПР «SimTest». Данная среда была разработана в СПбГУ на факультете ПМ-ПУ [5-19]. Использование этой САПР позволяет в несколько раз сократить время разработки теста.

Для разработки тестовых программ цифровых узлов и модулей формируются поведенческие модели их функционирования (см. глава 3, пункт 2). SimTest использует на входе структуру объекта контроля и программы его компонентов, т.е. выходной файл после разработки в Altera Quartus II. На основе этой информации формируется полная модель устройства и генерируются последовательности входных воздействий на объект контроля.

План построения теста для объекта контроля:

1. создание и добавление файла с поведенческой моделью объекта тестирования в систему САПР «SimTest»;
2. выбор интерфейса для создание тестовой последовательности;
3. формирование входных воздействий;
4. анализ полученного теста, решение вопроса об окончании тестирования или добавления новых тестовых последовательностей.

Более детально рассмотрим пункт 4. Окончанием тестирования могут служить разные аспекты: сроки, финансирование, но как правило основополагающую роль играет процент покрытия: после моделирования программа выдает файл, в котором показана реакция устройства, а также всех его компонентов на входные воздействия, и представлен процент покрытия, вычисляемый по формуле:

$$P = (N_{act}/N) * 100\%,$$

где N_{act} — число сигнальных линий устройства, активированных в ходе выполнения теста, N — число линий сигналов устройства. Разработчики пытаются достичь 100% покрытия, однако, как показывает практика, в большинстве случаев такого результата не достигается: некоторые разъемы компонентов запитаны на постоянный сигнал, для переключения какого-либо сигнала требуется много времени. Так же стоит помнить, что 100% покрытие не является гарантом окончания формирования тестовых воздействий, поскольку некоторые компоненты могут содержать в своем составе память, тем самым находясь в неизвестном состоянии (необходимо сначала направить такие сигналы, чтобы состояние объектов стало известным, в уже после производить тестирование). Все эти факты помогают оператору создать оптимальный тест.

Глава 4. Составление программной модели объекта контроля

Объект контроля ТС3.016.524.Э7 включает в себя 149 микросхем, 31 входных, 61 выходных разъемов.

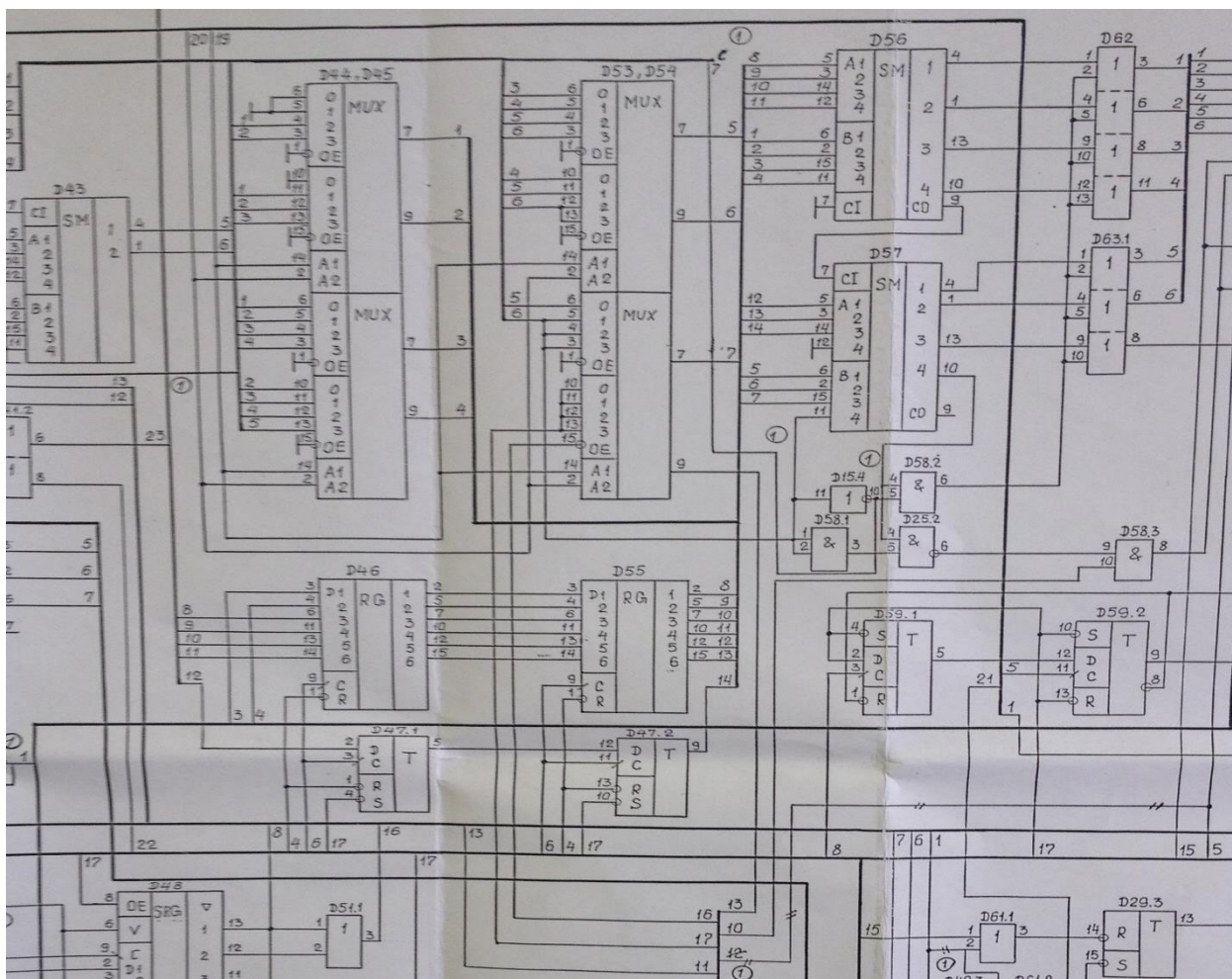


Рис.7: Фрагмент схемы ТС3.016.524.Э7

4.1 Моделирование компонентов

В работе будет рассмотрена часть микросхем:

4.1.1 КР1533IP5

Подробно рассмотрим следующую микросхему: КР1533IP5 — четырехразрядный, синхронный счетчик пульсаций. Микросхема имеет 9 входов и 2 выхода [53].

На Рис. 8. представлена фотография данной микросхемы, так она выглядит на плате или полке магазина. На Рис. 9. представлена условно графическая схема (так называемая цоколевка), она нужна для определения

краевых разъемов, подобным образом часто обозначаются микросхемы в бумажном варианте [54].



Рис. 8: Фото микросхемы KP1533IP5



Рис. 9: Цоколевка KP1533IP5

Данная микросхема обладает следующими параметрами (подобные данные в основном нужны для разработки, однако часто используются и в тестировании, например, из Таблицы 3 можно установить, чему равны логический ноль и логическая единица):

Номинальное напряжение питания	$5\text{В} \pm 10\%$
Выходное напряжение низкого уровня	$\leq 0,4\text{ В}$
Выходное напряжение высокого уровня	$\geq 2,5\text{ В}$
Прямое падение напряжения на антизвонном диоде	$\leq -1,5 \text{ В}$
Ток потребления при $U_{\text{п}} = 5,5\text{ В}$	$\leq 20\text{ мА}$
Входной ток низкого уровня	$\leq -0,2 \text{ мА}$
Входной ток высокого уровня	$\leq 20\text{ мкА}$
Входной пробивной ток	$\leq 0,1\text{ мкА}$
Входной ток	$ -10 \dots -112 \text{ мА}$
Время задержки распространения при включении по выводам 5, 6	$\leq 40\text{ нс}$
Время задержки распространения при выключении по выводам 5, 6	$\leq 45\text{ нс}$
Емкость входа	$\leq 5\text{ пФ}$

Таблица 2: Электрические параметры микросхемы KP1533IP5

Максимальный выходной ток низкого уровня	4 мА
Максимальный выходной ток высокого уровня	-10 мА
Напряжение питания	4,5...5,5 В
Входное напряжение низкого уровня	0...0,8 В
Входное напряжение высокого уровня	2...5,5 В
Максимальное напряжение, подаваемое на выход	5,5 В
Температура окружающей среды	-10...+70 °С

Таблица 3: Предельно допустимые режимы эксплуатации КР1533IP5

Схема работает по следующему принципу: если количество логических единиц четно, то на выходе «четный» (в нашем случае Q) образуется высокий уровень сигнала, а на выходе «нечетный» (в нашем случае Q_̅) образуется высокий сигнал, если количество логических единиц нечетно, то микросхема работает наоборот. Принцип работы микросхемы часто описывается таблицей истинности: таблица 4 (H — высокий уровень сигнала, L — низкий уровень сигнала):

Число входов, на которое подан высокий уровень напряжения	Выход	
	Q	Q _̅
0, 2, 4, 6, 8	H	L
1, 3, 5, 7, 9	L	H

Таблица 4: Таблица истинности КР1533IP5

На основе полученных данных можно приступить к программированию микросхемы, как говорилось ранее, будет использоваться язык Verilog HDL (далее представлен код элемента):

```

module ol_ic_1533ip5(d0,d1,d2,d3,d4,d5,d6,d7,d8,d9,q,q_);
input d0;
input d1;
input d2;

```

```

input d3;
input d4;
input d5;
input d6;
input d7;
input d8;
input d9;
output q;
output q_;
reg q;
reg q_;
reg sum;
always @
begin
    sum<=d1+d2+d3+d4+d5+d6+d7+d8+d9;
    if (sum%2==1'b0)
        begin
            q=1'b1;
            q_=1'b0;
        end
    else
        begin
            q=1'b0;
            q_=1'b1;
        end
end
endmodule

```

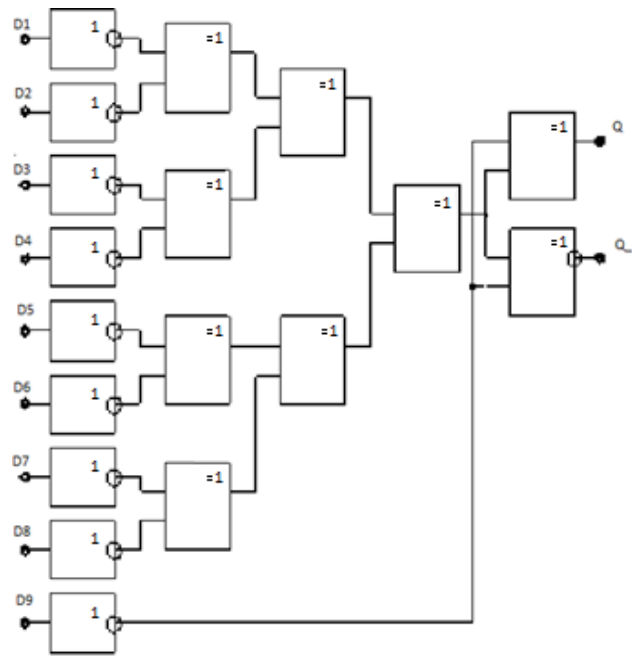


Рис. 10: Функциональная схема микросхемы К1533IP5

4.1.2 КР1533ИД3

Микросхема КР1533ИД3 (Рис. 11) — дешифратор, который преобразует четырех разрядный код, поступающий на входные разъемы ($D1, D2, D3, D4$) в логический ноль на одном из его выходов ($Y0, Y1, Y2, \dots, Y15$). Микросхема имеет два 2-стробирующих входа управления ($C1, C2$), которые могут служить в качестве логических, при использовании дешифратора в качестве мультиплексора данных. Для «разрешения» дешифровки данных на входы $C1$ и $C2$ необходимо подать сигналы низкого уровня (в противном случае все выходы микросхемы приобретут высокий уровень). Для более подробной логики расшифровки следует взглянуть на Таблицу 5 (Н — высокий уровень сигнала, L — низкий уровень сигнала), на Рис. 12 представлено расположение краевых разъемов.

Электрические параметры, которые необходимы для определения логических нуля и единицы, представлены в таблице 6, так же данные параметры могут использоваться для введения задержки и проверки или подачи сигнала на определенную ногу микросхемы [53].

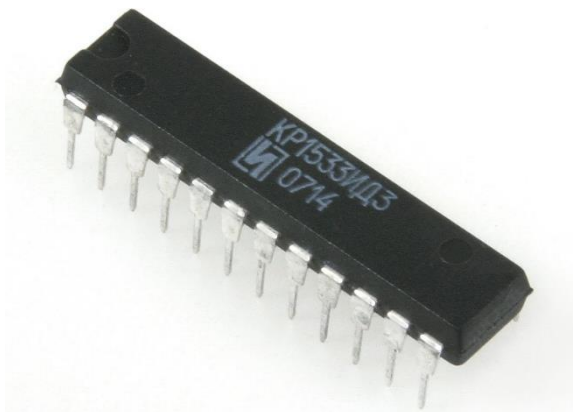


Рис. 11: Фото микросхемы КР1533ИД3

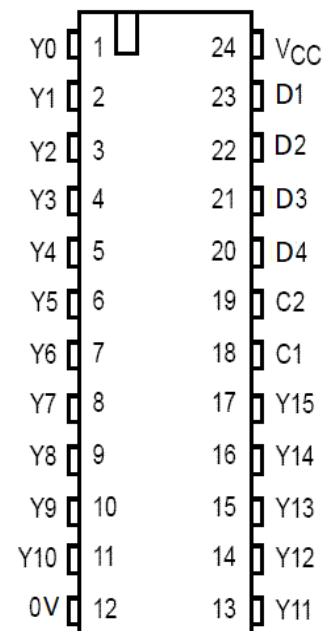


Рис. 12: Цоколевка КР1533ИД3

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	L	H	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	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	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	H	L	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	H	L	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	L	H

Таблица 5: Таблица истинности KP1533ID3

Номинальное напряжение питания	5В ± 10%	
Выходное напряжение низкого уровня	≤ 0,4 В	
Выходное напряжение высокого уровня	≥ 2,5 В	
Входной ток низкого уровня	≤ -0,2 мА	
Входной ток высокого уровня (для одиночного входа)	≤ 20 мкА	
Ток потребления	≤ 15 мА	
Время задержки распространения при включении от входов:	20, 21, 22, 23	≤ 33 нс
	18, 19	≤ 32 нс
Время задержки распространения при выключении от входов:	20, 21, 22, 23	≤ 36 нс
	18, 19	≤ 30 нс
Емкость входа	≤ 5 пФ	

Таблица 6: Электрические параметры микросхемы KP1533ID3

Фрагмент кода элемента:

```
module ol_ic_530id3(c1,c2,d4,d3,d2,d1,y0,y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15);
input c1,c2,d4,d3,d2,d1;
output y0,y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15;
reg y0,y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15;
always @ *
begin
    y0=1'b1;
    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;
    if (c1==1'b0&& c2==1'b0)
        begin
            if(d1==1'b0&&d2==1'b0&&d3==1'b0&&d4==1'b0)
                begin
                    y0=1'b0;
                end
            if(d1==1'b1&&d2==1'b0&&d3==1'b0&&d4==1'b0)
```

4.1.3 КР1533ІЕ7

Микросхема КР1533ІЕ7 (Рис. 13) — двоичный четырехразрядный реверсивный счетчик. Имеет 4 информационных входа: D1, D2, D3, D4, отдельные импульсивные тактовые входы для уменьшения (Td) и увеличения (Tu) счета (состояние счетчика изменяется по перепадам тактовых импульсов (положительных) на данных тактовых входах), вход асинхронной параллельной загрузки С — при подаче низкого уровня сигнала фиксируется код, поданный на информационные входы, вход сброса R — при подаче положительного импульса значение счетчика устанавливаются в состояние логического нуля, а также информационные выходы Q1, Q2, Q3, Q4 и выходы окончания счета на увеличение (P1) и уменьшение (P2), на Рис. 14 представлено расположение краевых разъемов [53, 55].

Более наглядная логика работы микросхемы представлена с помощью Таблицы 7 (Н — высокий уровень сигнала, L — низкий уровень сигнала, X — произвольный сигнал, ↑ — положительный запускающий перепад на тактовом входе).



Рис. 13: Фото микросхемы КР1533ІЕ7

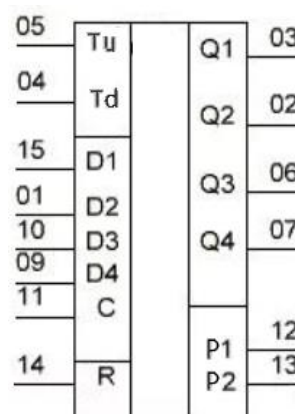


Рис. 14: Условно-графическое обозначение КР1533ІЕ7

Режим	Вход								Выход					
	R	C	Tu	Td	D1	D2	D3	D4	Q1	Q2	Q3	Q4	P1	P2
Сброс	H	X	X	L	X	X	X	X	L	L	L	L	H	L
	H	X	X	H	X	X	X	X	L	L	L	L	H	L
Параллельная загрузка	L	L	X	L	L	L	L	L	L	L	L	L	H	L
	L	L	X	H	L	L	L	L	L	L	L	L	H	H
	L	L	L	X	H	H	H	H	H	H	H	H	L	H
	L	L	H	X	H	H	H	H	H	H	H	H	H	H
Счет на увеличения	L	H	↑	H	X	X	X	X	Счет на увеличения				H	H
Счет на уменьшение	L	H	H	↑	X	X	X	X	Счет на уменьшение				H	H

Таблица 7: Режим счетчика KP1533IE7

Фрагмент кода элемента:

```

module ol_ic_1533ie7(d1,d2,d3,d4,r,c,tu,td,q1,q2,q3,q4,p1,p2);
input d1,d2,d3,d4,r,c,tu,td;
output q1,q2,q3,q4,p1,p2;
reg [3:0] q;
reg p1,p2;
always@ (posedge tu)
begin
    if (q==4'b1111)
        begin
            p1<=1'b0;
        end
    else
        p1<=1'b1;
end
always@ ( r, c, posedge td)
begin
    if (r==1'b1)
        begin
            q<=4'b0000;
        end
end

```

4.1.4 КР1533КР12

Микросхема КР1533КР12 (Рис. 14) — представляет собой двухканальный мультиплексор (содержит два четырехходовых мультиплексора с объединенными входами выбора Рис. 12.). Таким образом данная микросхема имеет 8 сигнальных входов: А1, А2, А3, А4, В1, В2, В3, В4 (по 4 на каждый мультиплексор), 2 выхода: Qa и Qb, входы выбора S0 и S1 — «выбирают» какой из сигнальных входов будет считываться, и входы разрешения Ea и Eb (первого и второго мультиплексора) — переводят в разомкнутое Z-состояние выход соответствующего мультиплексора при подаче высокого уровня сигнала. С более подробным принципом работы отдельного КР1533КР12 можно ознакомиться, используя таблицу 8 (В — высокий уровень сигнала, Н — низкий уровень сигнала, X — произвольный сигнал Z — неизвестное состояние). Расположение разъемов можно узнать, используя Рис. 15 [53].



Рис. 14: Фото микросхемы КР1533КР12

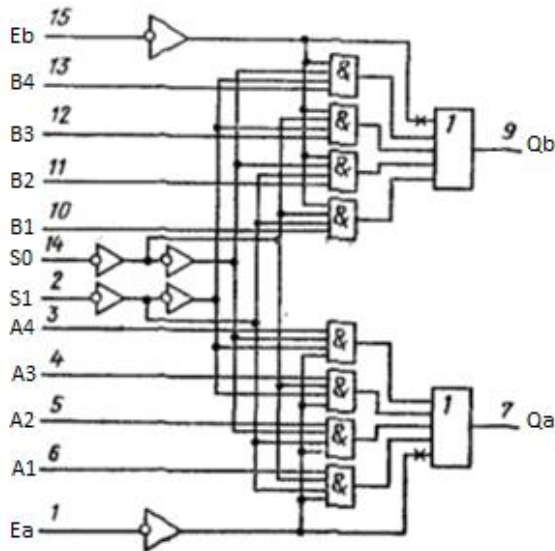


Рис. 15: Мультиплексор КР1533КР12

Выбор данных		Вход				Управление	Выход
		A1	A2	A3	A4		
S1	S0	A1	A2	A3	A4	Ea	y
X	X	X	X	X	X	H	Z
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	X	L	H

Таблица 8: Состояния одного мультиплексора

Фрагмент кода элемента:

```

module ol_ic_1533kp12(s1,s0,ea,eb,a1,a2,a3,a4,b1,b2,b3,b4,qa,qb);
input  s1,s0,ea,eb,a1,a2,a3,a4,b1,b2,b3,b4;
output qa,qb;
reg qa1;
reg qb1;
always @(*)
begin
    if ((s1==1'b0)&&(s0==1'b0)&&(a1==1'b0)
        ||(s1==1'b0)&&(s0==1'b1)&&(a2==1'b0)
        ||(s1==1'b1)&&(s0==1'b0)&&(a3==1'b0)
        ||(s1==1'b1)&&(s0==1'b1)&&(a4==1'b0))
        begin
            qa1=1'b0;
        end
    if ((s1==1'b0)&&(s0==1'b0)&&(a1==1'b1)
        ||(s1==1'b0)&&(s0==1'b1)&&(a2==1'b1)
        ||(s1==1'b1)&&(s0==1'b0)&&(a3==1'b1)
        ||(s1==1'b1)&&(s0==1'b1)&&(a4==1'b1))

```

4.1.5 P556RT7A

Микросхема P556RT7A (Рис. 16) представляет собой программируемое постоянное запоминающее устройство (PROM), емкость 16 кбит (2k x 8), на выходе может иметь 3 состояния. Способ программирования подобных чипов памяти описывался ранее. Имеет 10 адресных входов, 8 выходов и 3 входа разрешения выборки: ESE1, ESE2, ESE3. Считывание из памяти происходит, когда на входы ESE2, ESE3 подан высокий уровень сигнала, а на вход ESE1 низкий (Таблица 9).

В программной модели считывание данных будет происходить из файла «mem.data».



Рис. 16: Фото микросхемы P556RT7A

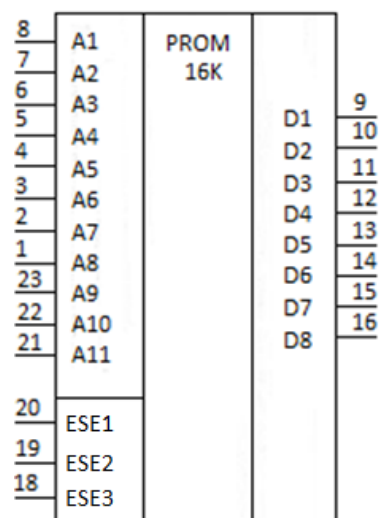


Рис. 17: Условно-графическое обозначение P556RT7A

Наличие перемычки в матрице	Состояние входа			Состояние выхода	Операция
	ESE1	ESE2	ESE3		
Есть	1	1	1	0	Чтение
Независимо от наличия	Любая другая комбинация			Z	Хранение
Нет	1	1	1	1	Чтение

Таблица 9: Режим работы P556RT7A

Код элемента:

```
module ol_ic_rt7(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,c1,c2,c3,d1,d2,d3,d4,d5,d6,d7,d8);
input a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11;
input c1,c2,c3;
inout d1,d2,d3,d4,d5,d6,d7,d8;
reg [7:0] out;
wire [11:0] adr;
reg [7:0] mem [0:2047];
assign adr = { a11, a10, a9, a8, a7, a6, a5, a4, a3, a2, a1};
assign {d8, d7, d6, d5, d4, d3, d2, d1} = out;
initial
begin
    $readmemh("mem.data", mem);
end
always @ (*)
begin
    if( c1 == 0 && c2 == 1 && c3 == 1 )
    begin
        out <= mem[adr];
    end
    else
    begin
        out = 8'bzzzzzzzz;
    end
end
endmodule
```

Объект контроля так же содержит:

- компараторы: 559IP1, 559IP2, 1533SP1;
- логические элементы 533LI3, 533LI6, 533LL1, 533LP8, 1533LA1, 1533LA3, 1533LA4, 1533LE1, 1533LN1, 1533LI1;
- мультиплексоры 1533KP12, 533KP16;
- ram 537RU13;
- регистры 533IR8, 533IR16;
- счетчики/сумматоры: 533IE5, 533IM6;
- триггеры: 1533TM2, 1533TM9, 1533TR2, 533TL2;
- другие арифметическо-логические устройства: 559IP4.

4.2 Электронная модель схемы

После того, как логика всех компонентов описана, приступим к созданию программной модели объекта контроля. Сначала загрузим все его компоненты. На Рис. 17. представлен пример добавления элемента 1533KP11 на программную модель объекта контроля.

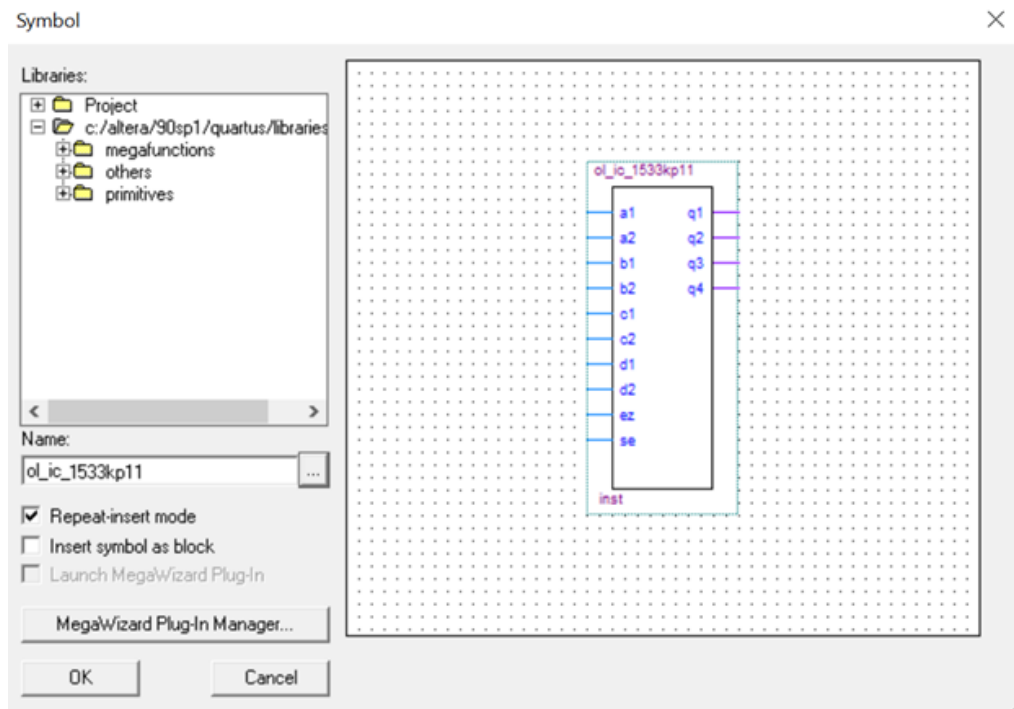


Рис. 17: Добавление элемента 1533KP11

После того, как все компоненты и связи между ними добавлены и подписаны, добавлены входы и выходы (краевые разъемы), необходимо собрать проект и сохранить в формате «.v», который необходим для работы в САПР «SimTest». Программная модель устройства контроля и ее небольшой фрагмент представлены на Рис. 18 и Рис. 19 соответственно.

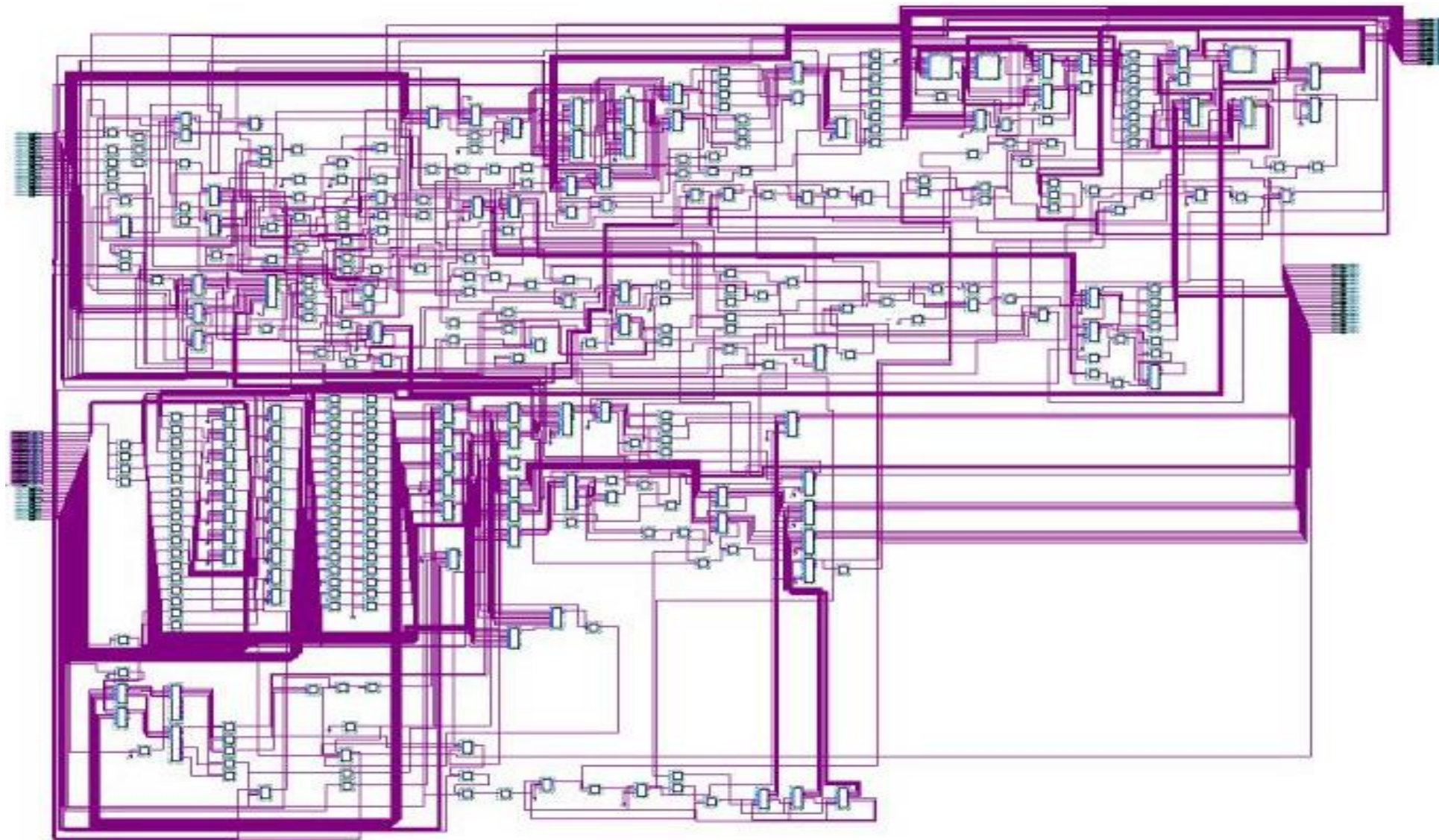


Рис. 18: Программная модель объекта контроля

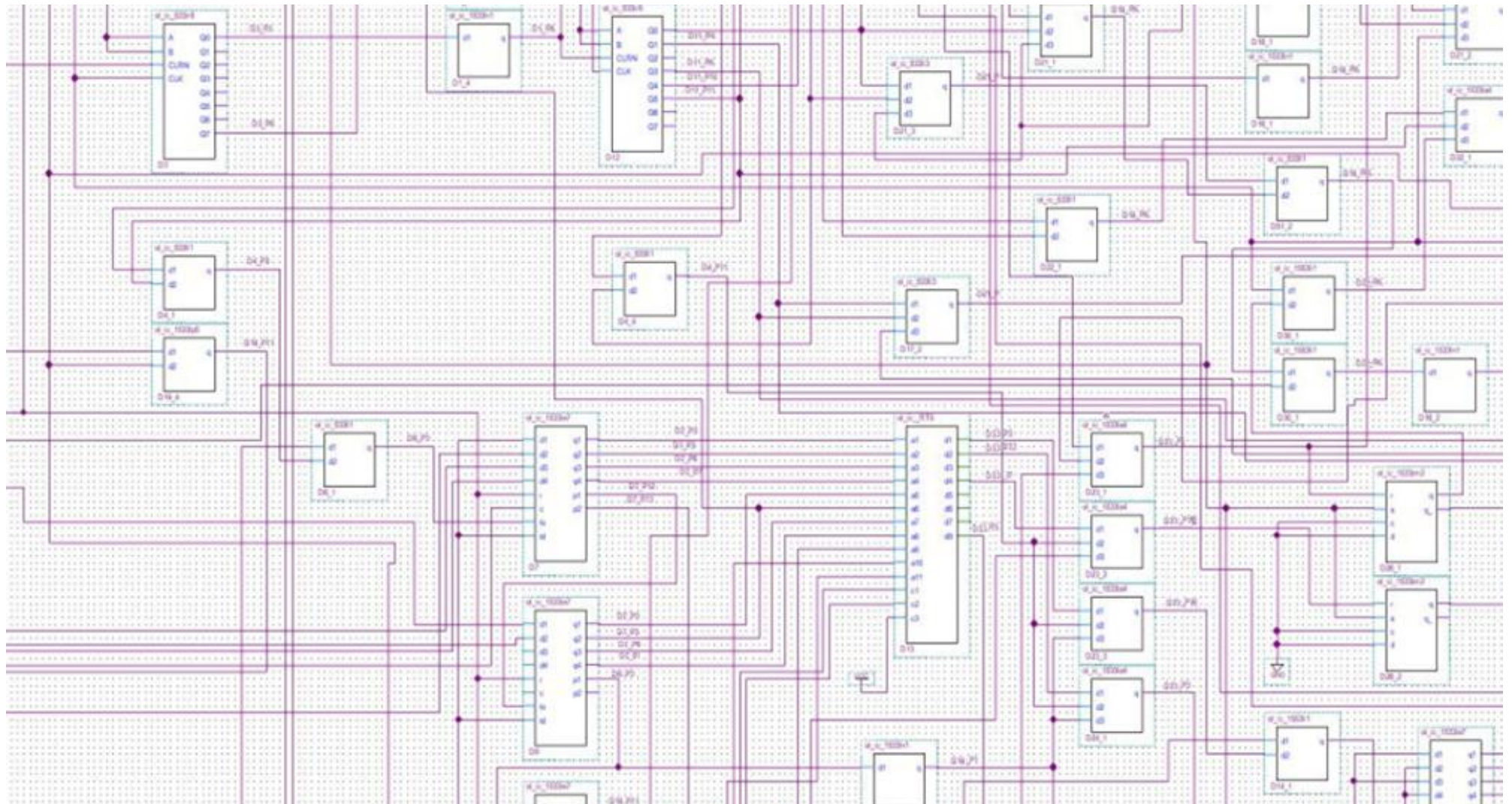


Рис. 19: Фрагмент программной модели объекта контроля

Глава 5. Тестирование

После создания программной модели объекта контроля перейдем к поиску тестового вектора.

На основе файла, сгенерированного программой Altera Quartus II, в САПР «SimTest» создается проект: загружаются все компоненты устройства (каждый элемент хранится в базе данных) и отдельный файл со связями между ними. Для перехода к поиску тестовой последовательности разработчику необходимо лишь указать какие из разъемов являются входными, а какие выходными (Рис. 20).

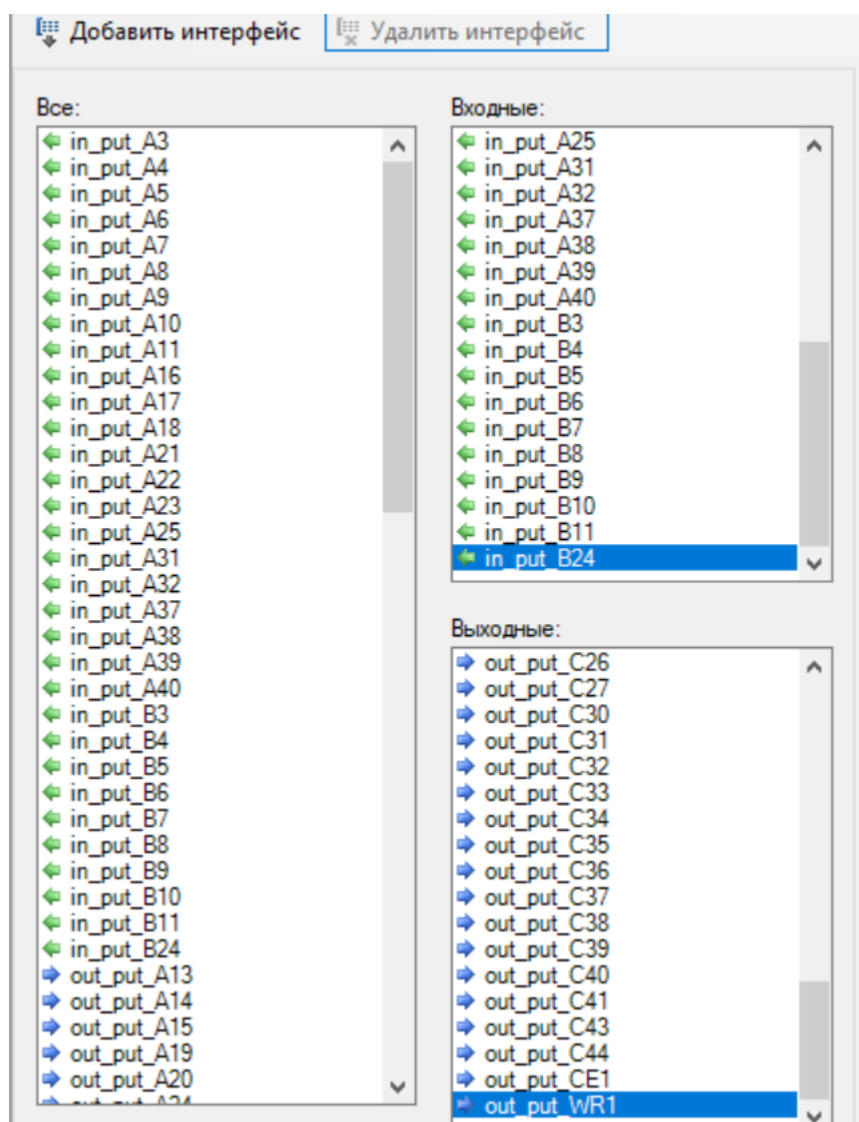


Рис. 20: Распределение на входные и выходные сигналы

5.1 Привязка логических интерфейсов

Сейчас необходимо каждому входу назначить определенный программный модуль — логический интерфейс (Рис. 21). САПР «SimTest» содержит несколько таких модулей, однако в работе будет использован один: «Группа сигналов» — такой интерфейс служит для привязки скрипта, содержащего изменения входных воздействий (описывается вручную).

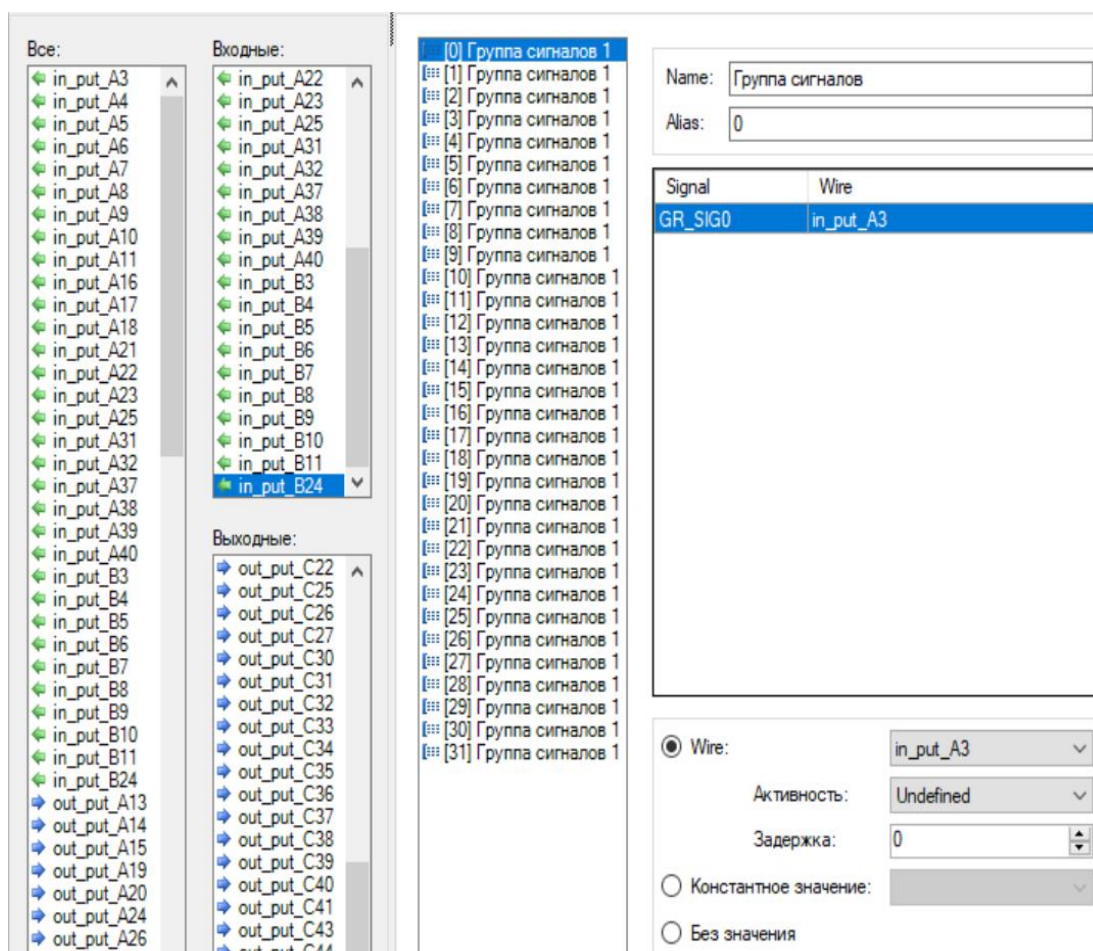


Рис. 21: Привязка логического интерфейса

5.2 Генерация теста

При подаче входного вектора «SimTest», используя элементы из базы, формируется стандартный .vcd-файл результатов, в котором происходит запись не только ответных выходных сигналов, но и состояние каждого отдельного элемента. Результат представлен в виде графика. (Рис. 22). Используя эти данные можно анализировать покрытие отдельных элементов, смотреть недостающие комбинации (время эмуляции составляет не более 10 секунд).

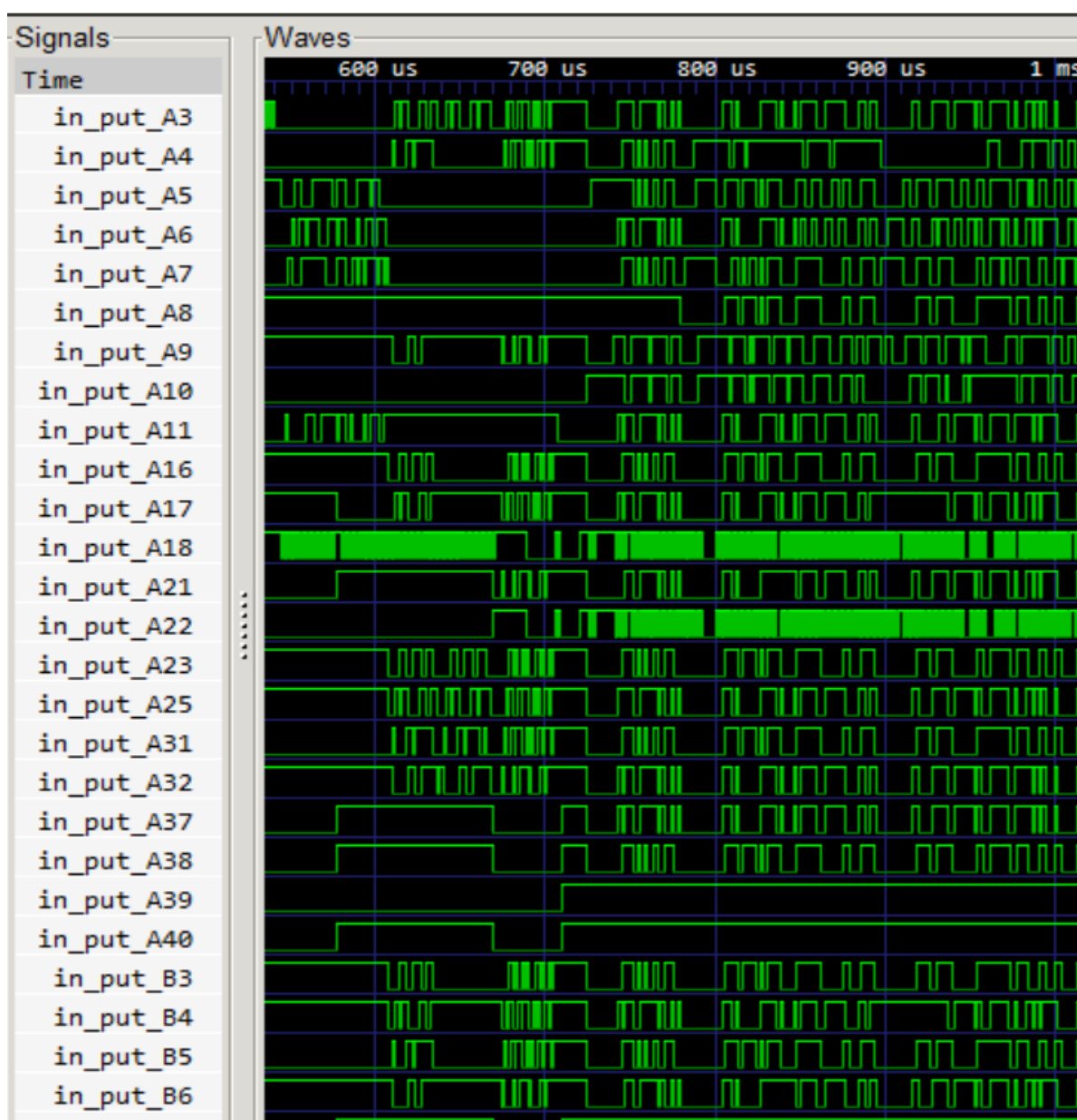


Рис. 22: Фрагмент диаграммы работы объекта контроля

5.2 Анализ покрытия

САПР «SimTest» сам анализирует .vcd-файл и на его основе вычисляет тестовое покрытие всех уровней тестируемого объекта. На Рис. 23 показана статистика по покрытию: левая часть — общее покрытие, в данном случае оно составляет 91,76%, правая часть — статистика по переключению краевых разъемов — 100%.

Покрытие полученного теста является максимальным, т.к. 8,24% составляют линии, которые не могут быть изменены или подключены к постоянному источнику питания (постоянная «1») или земле (постоянный «0»).

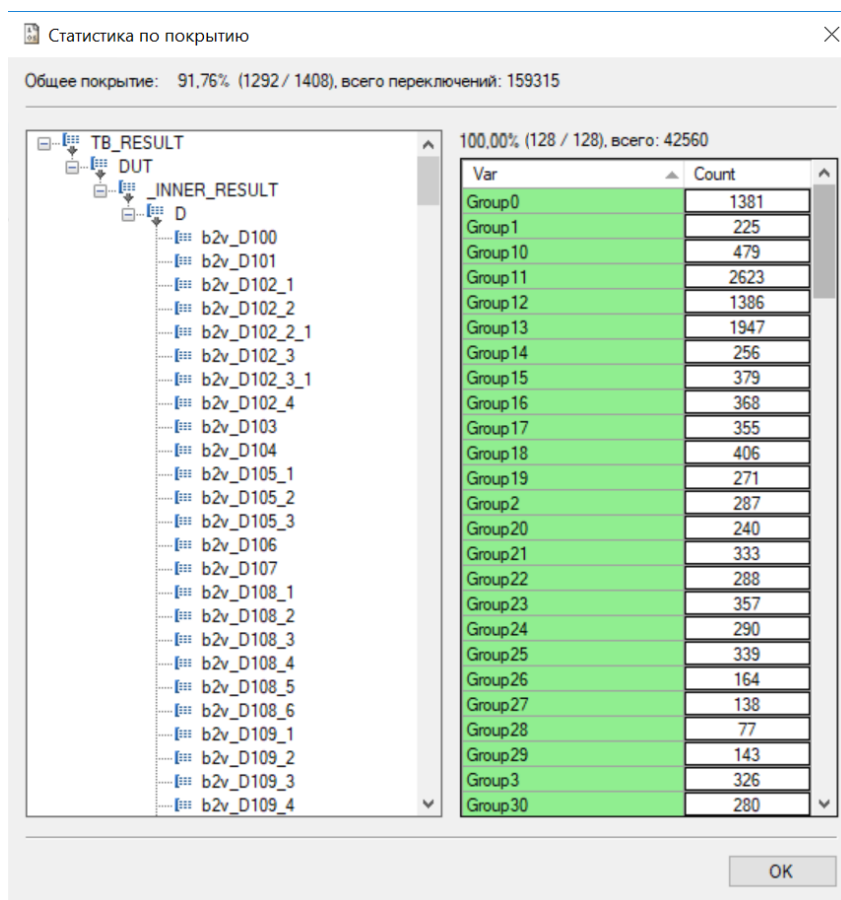


Рис. 23: Тестовое покрытие

Выводы

Разработаны программные модели всех компонентов устройства контроля ТС3.016.524.Э7 на языке описания аппаратуры Verilog HDL. Стоит отметить, что данные компоненты, добавлены в общую базу данных компонент, так как могут использоваться в дальнейшем для других объектов контроля.

Построена программная модель самого радиоэлектронного объекта контроля в САПР Altera Quartus II, на основе полученной документации, спецификаций и схемы устройства.

Построена тестовая последовательность и с помощью САПР «SimTest» произведена эмуляция теста: тестовая последовательность, которая должна подаваться на входные краевые разъемы объекта контроля, представлена в виде скрипта, отвечающий им вектор выходных сигналов был эмулирован с помощью САПР «SimTest».

Произведен анализ тестовой программы, который показал, что смоделированный тест для цифрового устройства ТС3.016.524.Э7 удовлетворяет необходимым требованиям тестового покрытия.

Заключение

В работе:

- разобраны возможные типы тестирования электронных устройств, также описаны их плюсы и минусы;
- представлены и реализованы все этапы формирования программных моделей отдельных компонентов объекта (69 логических элементов, 12 компараторов, 15 мультиплексоров, 1 ram, 2 rom, 11 регистров, 17 счетчиков, 37 триггера, 2 дешифратора) в среде разработки Altera Quartus II (таких как КР1533IP5, КР1533ID3, КР1533IE7, КР1533КР12, P556RT7A);
- приведены и реализованы все этапы построения тестовой последовательности и эмуляции теста на примере реального объекта контроля ТС3.016.524.Э7 в САПР «SimTest»;
- произведен анализ полученных данных на основе тестового покрытия, предоставленного САПР «SimTest»

Список литературы

1. Новиков, М. А. Олег Михайлович Лосев — пионер полупроводниковой электроники (рус.) // Физика твёрдого тела. — 2004. — Т. 46, № 1. — С. 5—9
2. https://www.kit-e.ru/articles/elcomp/2006_9_198.php
3. <http://radioschema.ru/istoriya/istoriya-izobreteniya-tranzistora.html>
4. Шокли, У. Теория электронных полупроводников: Приложения к теории транзисторов. // М.: Издательство иностранной литературы, 1953. — 714 с.
5. Гришкин В.М., Лопаткин Г.С., Михайлов А.Н., Овсянников Д.А. Интерфейсный метод построения моделей входных воздействий для тестирования электронных цифровых модулей. // Электроника: Вопросы радиоэлектроники. 2013. Т. 1. № 1. С. 80-89.
6. Гришкин В.М., Степанов Ю.Л., Лопаткин Г.С., Большаков А.А. Подход к разработке тестов цифровых электронных модулей для автоматического тестового оборудования. // Электроника: Вопросы радиоэлектроники. 2013. Т. 1. № 1. С. 89-99.
7. Мельник В., Гришкин В., Михайлов А., Овсянников Д. Тестовый контроль и диагностика радиоэлектронной аппаратуры // Электроника: Наука, технология, бизнес. 2013. Вып. 128. С. 114-117.
8. 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.
9. Папулин Д.А., Смирнов А.Д., Гришкин В.М., Степанов Ю.Л. Метод применения устройств тестового контроля в качестве сканера цифровых контрольно-диагностических тестов. // Смирнов Николай Васильевич. Процессы управления и устойчивость. 2018. Т. 5. № 1. С. 287-291.
10. Папулин Д.А., Смирнов А.Д., Гришкин В.М., Степанов Ю.Л. Поведенческие модели для логических элементов с частично определенной

- логикой. // Смирнов Николай Васильевич. Процессы управления и устойчивость. 2018. Т. 5. № 1. С. 292-296.
11. Степанов Ю.Л., Гришкин В.М., Елаев Е.В., Федюкович П.А. Развитие программной среды "ястек" и ее использование при написании тестовых программ для цифровых модулей. // М.: Центральный научно-исследовательский институт экономики, систем управления и информации «Электроника. Вопросы радиоэлектроники. 2015. № 2. С. 198-205.
12. Федюкович П.А., Елаев Е.В., Машинский Н.С., Гришкин В.М. Формирование тестовых последовательностей с помощью sat-решателя. // Смирнов Николай Васильевич. Процессы управления и устойчивость. 2015. Т. 2. № 1. С. 521-526.
13. Grishkin V., Yelaev Y., Lopatkin G., Mikhailov A., Ovsyannikov D. Interface method of digital devices testing. // Institute of Electrical and Electronics Engineers. Tenth International Vacuum Electron Sources Conference (IVESC) & Second International Conference on Emission Electronics (ICEE) 2014. С. 107-108.
14. Melnik V.I., Mikhailov A.N., Grishkin V.M., Ovsyannikov D.A., Yelaev Y.V. Methods of modeling of the test inputs for analysis the digital devices. // Спб: Издательство Санкт-Петербургского государственного университета. 2014 International conference on computer technologies in physical and engineering applications (ICCTPEA). 2014. С. 112-113.
15. 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. // Спб: Издательство Санкт-Петербургского государственного университета. 2nd International Conference on Emission Electronics (ICEE) Selected papers. Proceedings Edited by: N. V. Egorov, D. A. Ovsyannikov, E. I. Veremey. 2014. С. 48-50.
16. Машинский Н.С., Елаев Е.В., Федюкович П.А. Моделирование сложных цифровых устройств с целью их тестирования. // Смирнов Николай

- Васильевич. Процессы управления и устойчивость. 2015. Т. 2. № 1. С. 452-457.
17. Елаев Е.В., Степанов Ю.Л., Ферсенков В.В. Подходы к моделированию микропроцессоров для построения контрольно-диагностических тестов. // Смирнов Николай Васильевич. Процессы управления и устойчивость. 2015. Т. 2. № 1. С. 398-403.
18. Интерфейсный метод автоматизированной генерации тестовых воздействий для цифровых радиоэлектронных объектов контроля. // Спб.: Санкт-Петербургский государственный университет промышленных технологий и дизайна. Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2015. № 4. С. 19-24.
19. Гусев О.А., Елаев Е.В., Машинский Н.С., Нуракунов А. Автоматизация генерации тестовых воздействий для комбинационных цифровых схем. // Смирнов Николай Васильевич. Процессы управления и устойчивость. 2016. Т. 3. № 1. С. 389-393.
20. Комолов Д.А., Мьяльк 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.
21. <https://ru.wikipedia.org/wiki/Схемотехника>
22. Г. И. Веселов. Большая советская энциклопедия // М. : Советская энциклопедия, 3-е изд. 1969—1978.
23. Ульрих Титце, Кристоф Шенк (U. Tietze, Ch. Schenk). Перевод Г.С. Карабашев. Графика Р.В. Салимонов. Полупроводниковая схемотехника. // М.: Издательство «ДМК Пресс», 2008 12-е издание.
24. Кудрявцев С. (Скайф). Рождение радио. — Л.: Молодая гвардия, 1935
25. <http://www.texnic.ru/tools/stud/1/1004.htm>
26. <https://www.ft.com/content/b15a6652-ab37-11e8-94bd-cba20d67390c>
27. Николай Алексеев. Кремниевая эволюция // ComputerBild. 2011. Вып. 22. С. 80—85.

28. <https://occlub.ru/news/hardware/5382-10-nm-tekhprotsess-intel-uvelichivaet-plotnost-tranzistorov-v-2-7-raza-po-sravneniyu-s-14-nm>
29. https://ru.wikipedia.org/wiki/Постоянное_запоминающее_устройство
30. https://ru.wikipedia.org/wiki/Магнитоэлектронные_запоминающие_устройства
31. Угрюмов Е. П. Цифровая схемотехника. — СПб.: БХВ-Петербург, 2005.
32. <https://ru.wikipedia.org/wiki/DRAM>
33. <https://ru.wikipedia.org/wiki/SRAM>
34. <https://studfiles.net/preview/1556140/>
35. <https://lab201.jimdo.com/студентам/лекции-мдк-01-01/34-элементная-база-озу/>
36. Задорин А., Захаров Г. Система автоматизированного визуального контроля печатных плат. // Aplite, 2002. Вып. 10. С. 53-56.
37. Астрицкий. А. Компоненты и технологии. // Общество с ограниченной ответственностью Издательство Файнстрит, : журнал. — 2003, выпуск 3. С. 172—173.
38. Иванов А. Выбор тестовой стратегии при производстве цифровой и аналогово-цифровой техники. // Печатный монтаж. 2010. Вып. 2. С. 48-53.
39. 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.
40. Иванов А. Периферийное сканирование в жизненном цикле изделия / Производство электроники: технологии, оборудование, материалы. 2010. Вып. 2. С 1-3.
41. Renbi A., Delsing J. Contactless Testing of Circuit Interconnects // Journal of Electronic Testing: Theory and Applications, 2015, No 31. P. 229–253.
42. 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.

43. <https://www.xjtag.com/ru/about-jtag/what-is-jtag/>
44. 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.
45. <http://eesemi.com/atpg.htm>
46. https://www.researchgate.net/publication/234044505_FAN_A_fanout-oriented_test_pattern_generation_algorithm
47. https://www.researchgate.net/publication/228556689_Neural_Network_Automatic_Test_Pattern_Generator
48. <http://www.informtest.ru/produktsiya/functional-control-systems/test-6408/>
49. <https://sovtest-ate.com/equipment/ft-17-ir-testovaya-sistema-funktsionalnogo-kontrolya-s-integrirovannym-teplovizorom/>
50. 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.
51. Соловьев В. В. Основы языка проектирования цифровой аппаратуры Verilog. — М.: Горячая линия — Телеком, 2014. — 208 с.
52. Шило В. Л. Популярныe цифровые микросхемы. М.: Радио и связь, 1987. 352 с.
53. <http://ielekt.ru/datasheet/1533ip5.pdf>
54. Нефедов А.В. Интегральные микросхемы и их зарубежные аналоги. М.: ИП. РадиоСофт, 2000. Т. 2. 640 с.