

МИНОБРНАУКИ РОССИИ

Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

П. Н. БОНДАРЕНКО О. И. БУРЕНЕВА
Л. К. ГОЛОВИНА

УЗЛЫ И УСТРОЙСТВА СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Учебно-методическое пособие

Санкт-Петербург
Издательство СПбГЭТУ «ЛЭТИ»
2017

УДК 004.3(07)
ББК 3973.2-018я7
Б81

Бондаренко П. Н., Буренева О. И., Головина Л. К.

Б81 Узлы и устройства средств вычислительной техники: учеб.-метод. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2017. 64 с.

ISBN 978-5-7629-2163-3

Содержит методические указания к выполнению работ, направленных на получение практических навыков синтеза и анализа комбинационных и последовательностных схем, а также микропрограммных автоматов с использованием САПР Quartus II.

Предназначено для подготовки бакалавров направления «Информатика и вычислительная техника» по профилю «Вычислительные машины, комплексы, системы и сети». Также могут быть использованы специалистами, разрабатывающими электронную аппаратуру.

УДК 004.3(07)
ББК 3973.2-018я7

Рецензент: канд. техн. наук Г. А. Стеклова (СПб ГУПТД).

Утверждено
редакционно-издательским советом университета
в качестве учебно-методического пособия

ISBN 978-5-7629-2163-3

© СПбГЭТУ «ЛЭТИ», 2017

Введение

Приобретение знаний в области разработки цифровых узлов и устройств необходимо для формирования студентов как специалистов, способных решать задачи, связанные с проектированием, созданием и эксплуатацией цифровой техники.

Основной целью курса лабораторных работ по дисциплине «Узлы и устройства средств вычислительной техники» является получение практических навыков в решении задач синтеза комбинационных и последовательностных схем в различных базисах, анализа спроектированных узлов в процессе моделирования их работы, а также формирование умений проведения макетных экспериментов с использованием специализированного оборудования.

В учебно-методическом пособии приводятся краткие теоретические сведения, необходимые при решении задач синтеза и анализа узлов и устройств, а также методика разработки устройств и узлов с использованием систем автоматизированного проектирования (САПР). В качестве САПР выбран программный пакет Quartus II.

Курс состоит из восьми лабораторных работ. Первые две посвящены проектированию комбинационных схем: одна и та же функция реализуется в заданном логическом базисе, а также с использованием дешифратора и мультиплексора. В этих работах затрагивается и понятие модульного проектирования путем создания иерархического проекта. Третья работа предполагает проектирование типового комбинационного узла, четвертая – знакомство с триггерными элементами различного типа. Последние четыре работы ориентированы на разработку последовательностных узлов: регистров, счетчиков, конечных цифровых автоматов. Каждая работа кроме непосредственного проектирования устройства предполагает моделирование и проведение натурного эксперимента с использованием учебного стенда, реализованного на базе программируемой логической интегральной микросхемы (ПЛИС).

Результатом выполнения каждой работы является разработка структурной и функциональной схем цифрового устройства в выбранном или заданном интегральном базисе в соответствии с техническим заданием.

Лабораторная работа 1

ЗНАКОМСТВО С СИСТЕМОЙ ПРОЕКТИРОВАНИЯ QUARTUS II, РЕАЛИЗАЦИЯ КОМБИНАЦИОННОЙ СХЕМЫ

Цель работы – освоение процесса проектирования цифровой схемы в системе автоматизированного проектирования Quartus II, включающего в себя этапы создания цифровой схемы в графическом редакторе, моделирования работы схемы, загрузки результатов проектирования в программируемую логическую интегральную схему и проведение макетного эксперимента. В процессе работы выполняется проектирование простой комбинационной схемы.

Краткие теоретические сведения

Логическая схема называется комбинационной цепью (КЦ), если значения ее выходов y_1, y_2, \dots, y_n описываются системой m булевых функций от входных переменных x_1, x_2, \dots, x_m , т. е. выходные величины КЦ в установившемся режиме определяются только текущим значением входных.

В состав цифровых устройств обычно входят типовые функциональные узлы, а также логические схемы, специфичные для конкретного проекта, – так называемая произвольная логика. Произвольная логика является объектом индивидуального проектирования, которое выполняется по следующим этапам.

На начальном этапе задается характер функционирования КЦ. Это может быть сделано либо с помощью аналитического описания в виде системы булевых функций, либо с помощью таблицы истинности, имеющей 2^n строк (по строке для каждого набора входных переменных) и $n + m$ столбцов. От таблицы легко перейти к совершенной дизъюнктивной нормальной форме (СДНФ), т. е. к дизъюнкциям конститuent единицы искомым функций, составлением логической суммы тех входных наборов, на которых функция принимает единичное значение.

Следующие этапы определяются средствами реализации схемы, в качестве которых могут быть выбраны логические блоки табличного типа, логические блоки в виде последовательности матриц элементов «И» и «ИЛИ», универсальные логические блоки на основе мультиплексоров, логические блоки, собираемые из элементов некоторого базиса.

В случае реализации схемы на логических блоках должна быть выполнена минимизация логической функции и должен быть осуществлен переход к заданному логическому базису.

При этом под минимизацией понимается такое преобразование логических функций, которое упрощает их в соответствии с заданным критерием, также определяемым средствами реализации схемы. Такими критериями могут быть суммарное число входов всех логических элементов схемы (критерий Квайна), минимизация площади кристалла, общее число всех выводов корпусов интегральных микросхем при реализации устройств на печатных платах.

Порядок работы в системе проектирования Quartus II

Программное обеспечение Quartus II предоставляет полный цикл для создания цифровых устройств с дальнейшей их реализацией на базе ПЛИС, объединяя этапы проектирования, синтеза, размещения элементов и трассировки соединений. Анализ работоспособности проектов выполняется в процессе моделирования. В цикле лабораторных работ используется версия Quartus II 13.0.

Этап 1. Создание проекта в САПР Quartus II

1. Понятие проекта. Под термином «проект» в рамках САПР Quartus II понимается набор файлов, описывающих проектируемое устройство, а также ряд вспомогательных файлов. Система поддерживает несколько типов файлов с описанием проекта как графических, так и языковых. В предлагаемом цикле работ будут использоваться файлы описаний блок-схем устройства Block Design File (*.bdf). Вспомогательные файлы содержат дополнительную информацию о проекте, их имена должны совпадать с именем проекта.

2. Создание проекта.

2.1. Перед работой рекомендуется создать в рабочей области жесткого диска папку, в которой будет размещен проект.

2.2. Проект в системе Quartus II создается с использованием встроенного мастера создания проекта ***New Project Wizard***, запускаемого через меню ***File***. Откроется окно ***New Project***. При первом запуске появляется окно, в котором отражена вся возможная последовательность действий при открытии нового проекта. Это окно пропускается нажатием клавиши ***Next***.

В окне первого этапа создания проекта (рис. 1.1) указывается рабочий каталог, созданный в п. 2.1. Если ввести имя несуществующего каталога, то

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?
 ...

What is the name of this project?
 ...

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.
 ...

< Back **Next >** Finish Cancel Help

Рис. 1.1. Окно создания проекта

он будет создан системой Quartus II. В этом же окне задается имя проекта и файла верхнего уровня.

Для продолжения следует нажать кнопку *Next*.

Второе окно предназначено для подключения к проекту ранее созданных файлов. Так как в работе отсутствуют дополнительные файлы, которые должны быть подключены к проекту, выполняется переход к следующему окну нажатием кнопки *Next*.

В третьем окне (рис. 1.2) требуется указать семейство и тип ПЛИС. В работе будет использован макет с установленной микросхемой семейства Cyclone II EP2C5Q208.

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.
 You can install additional device support with the Install Devices command on the Tools menu.

Device family
 Family:
 Devices:

Target device
☐ Auto device selected by the Fitter
☒ Specific device selected in 'Available devices' list
☐ Other: n/a

Show in 'Available devices' list
 Package:
 Pin count:
 Speed grade:
 Name filter:
☒ Show advanced devices ☐ HardCopy compatible only

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PLL	G
EP2C5F256I8	1.2V	4608	158	119808	26	2	8
EP2C5Q208C7	1.2V	4608	142	119808	26	2	8
EP2C5Q208C8	1.2V	4608	142	119808	26	2	8
EP2C5T144C8	1.2V	4608	89	119808	26	2	8

Companion device
 HardCopy:
☐ Limit DSP & RAM to HardCopy device resources

< Back **Next >** Finish Cancel Help

Рис. 1.2. Окно выбора микросхемы

Четвертое окно позволяет подключить к системе Quartus II дополнительные средства EDA. В работе предполагается использовать программные средства моделирования, следует указать средство моделирования (Simulator) ModelSim-Altera.

В последнем окне приводится полная информация о сделанных назначениях. При необходимости к предыдущим окнам можно вернуться клавишей Back. Создание проекта завершается нажатием кнопки **Finish**.

3. *Ввод схемы в графическом редакторе.* На рис. 1.3 показано главное окно САПР Quartus II. Project navigator (1) отображает иерархию проекта, содержащиеся в нем файлы и блоки проекта. В центре окна (2) располагаются основные окна создания и редактирования исходных файлов проекта, временных диаграмм и так далее. В верхней части окна содержится меню и рабочая панель с наиболее часто используемыми кнопками (3). Нижняя часть окна содержит окна, в которые выводятся различные информационные сообщения, а также сообщения об обнаруженных ошибках и проблемах (4).

В лабораторной работе используются следующие пункты меню:

File:

- *New* – обеспечивает переход к окну диалога создания файлов различного типа;
- *Open* – обеспечивает переход к окну диалога открытия файлов различного типа;
- *New Project Wizard* – обеспечивает переход к окну диалога создания нового проекта;
- *Open Project* – обеспечивает переход к окну диалога открытия созданных ранее проектов;

Assignments:

- *Pin Planner* – обеспечивает переход к окну диалога назначения контактов проекта;

Processing:

- *Start Compilation* – запускает полную компиляцию проекта;
- *Analyze current file* – позволяет проверить ошибки текущего файла;
- *Start Simulation* – запускает моделирование проекта;

Tools:

- *Net list viewer* – позволяет просматривать структура проекта на различном уровне;

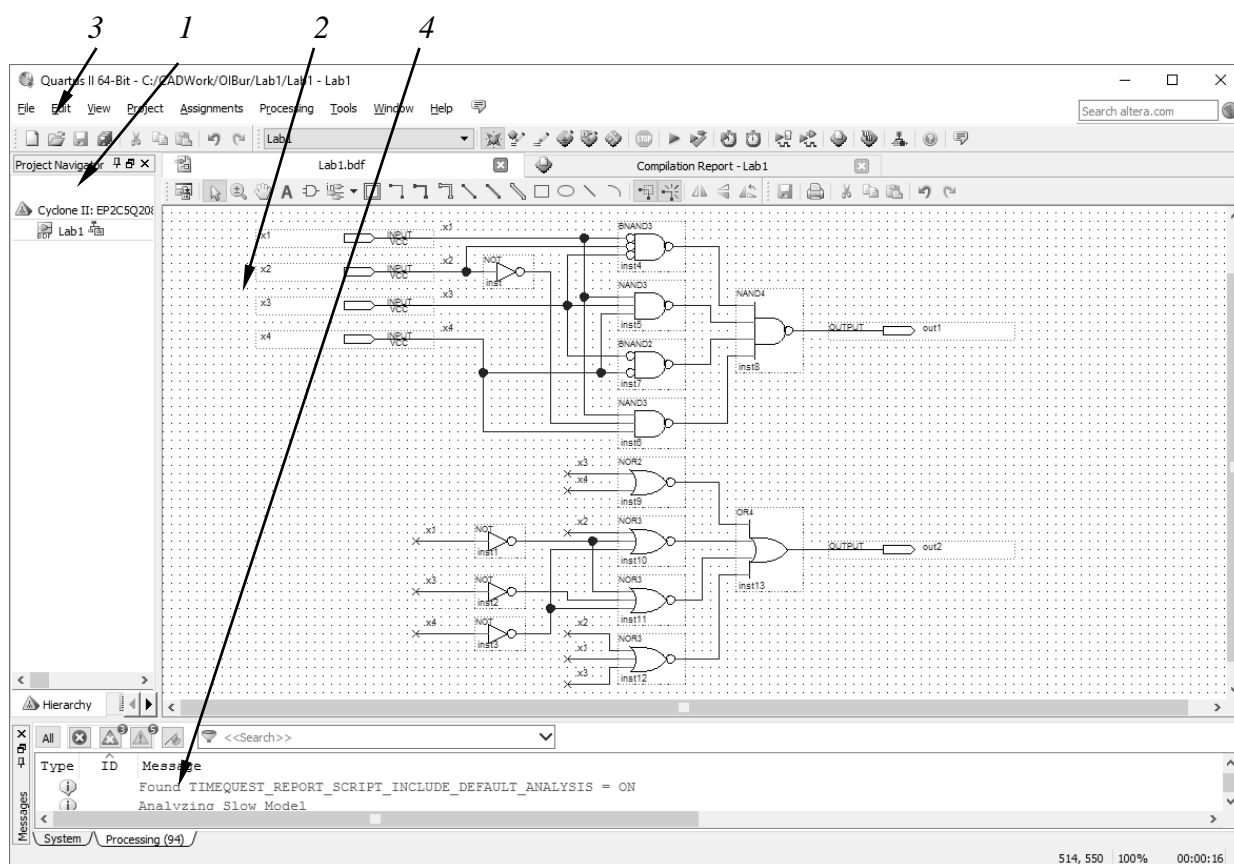



Рис. 1.3. Окно САПР Quartus II

- *Programmer* – запускает программатор, предназначенный для загрузки сформированного проекта в ПЛИС.

Для создания схемы необходимо выполнить следующие действия.

3.1. Запустить графический редактор, для чего выбрать в меню **File** команду **New** и в появившемся окне указать тип создаваемого файла выбором пункта **Block Diagram / Schematic File**. В результате откроется окно графического редактора блок-схемы. Таким же образом может быть создан любой другой файл проекта.

3.2. Сохранить файл как модуль верхнего уровня иерархии, используя команду **Save As** меню **File** и введя имя файла, совпадающее с именем проекта. При использовании другого имени модуль не будет по умолчанию считаться модулем верхнего уровня иерархии, и необходимо будет сделать соответствующие настройки, чтобы он стал таковым.

3.3. Разработать схему, используя библиотеку примитивов САПР Quartus II. Для вставки в рабочую область графического редактора блок-схем элемента необходимо открыть библиотеку пиктограммой  или через контекстное меню **Insert/Symbol**.


При этом появится диалоговое окно *Symbol*, в котором выбирается требуемый элемент. В левом верхнем углу окна находится список библиотечных элементов и мегафункций. Чуть ниже находится поле *Name*, с помощью которого можно быстро найти элемент по его имени или же указать путь к другим элементам, например созданным разработчиком ранее.

При создании предлагаемой в работе схемы используются элементы, сгруппированные в разделе *Primitives / Logic*. Для соединения спроектированного фрагмента с выводами микросхемы или для создания на ее основе символа необходимо добавить порты ввода-вывода; соответствующие элементы находятся в директории *Primitives / Pin*. Портам следует задать собственные имена: двойным щелчком по порту вызывается диалоговое окно, в поле *Pin name(s)* которого вводится новое имя.

Для соединения элементов обычно используются одиночные проводники и шины. Проводникам, шинам, линиям в канале рекомендуется присваивать имена. Например, чтобы не загромождать проект линиями связи можно не соединять их непосредственно, а связывать указанием одинаковых имен. Для задания проводнику имени необходимо щелкнуть по нему правой кнопкой мыши, в выпадающем меню выбрать *Properties* и в появившемся диалоговом окне ввести имя.

3.4. Сохранить разработанную схему командой *File / Save*.

4. *Компиляция проекта* – это преобразование описания проекта в его реализацию на кристалле ПЛИС. На этом этапе выявляются синтаксические ошибки в проекте, проверяется его логическая завершенность и возможность реализации проекта на выбранном кристалле ПЛИС. Результатом компиляции становится оптимизированный и смонтированный проект, используемый для создания файлов, применяемых для временного моделирования, временного анализа, анализа потребляемой мощности и программирования кристалла. Исходными файлами для выполнения этапа являются файлы со схемным представлением *.bdf, на выходе получаются файлы отчета (*.rpt, *.htm) и созданная база данных (*.rdb).

Для запуска компиляции проекта используется пиктограмма  на панели инструментов или команда *Start Compilation* из меню *Processing*.

В окне сообщений менеджера проекта отображается информация о ходе выполнения компиляции включая предупреждения и сообщения об ошибках. Сообщения об успешном выполнении шагов компиляции выводятся зеленым цветом и начинаются со слова *Info*, предупреждения выводятся синим цве-

том и начинаются со слова **Warning**, ошибки выводятся красным цветом и начинаются со слова **Error**. При успешном завершении компиляции на экран выводится соответствующее сообщение.

5. *Моделирование проекта*. Моделирование (**Simulation**) позволяет убедиться в правильности работы спроектированной схемы: анализируется работа устройства в соответствии с заданными входными воздействиями.

Входные воздействия задаются в виде некоторого набора входных сигналов и в ответ на них подсистема моделирования (**Simulator**) среды Quartus II, в соответствии со схемой проекта, синтезирует выходные сигналы.

Модель достаточно точно отображает работу схемы, которая будет реализована непосредственно в микросхеме ПЛИС. Симулятор учитывает особенности реальной работы проекта в ПЛИС: особенности данной микросхемы, задержки распространения сигнала и т. п.

Подсистема моделирования позволяет выполнять:

- функциональное моделирование проекта (**Functional Simulation**) при котором проверяется правильность описания и логического функционирования схемы;

- моделирование с учетом временных параметров реальной ПЛИС (**Timing Simulation**), позволяющее проверить не только правильность логического функционирования проекта, но и его работу с учетом реальных параметров выбранной ПЛИС в самых жестких условиях эксплуатации.

Для моделирования схемы необходимо выполнить следующие действия:

5.1. Создать файл, содержащий входные тестовые воздействия, с помощью редактора временных диаграмм.

Файлы вектора входных воздействий задаются в виде временных диаграмм с использованием специального редактора (**Waveform Editor**). Файлы имеют расширение *.vwf (*Vector Waveform Files*).

Файл создается с использованием команды меню **File / New**, в открывшемся диалоговом окне в группе **Verification / Debugging Files** выбирается **University Program VWF** (файл вектора временных диаграмм). В результате откроется пустое окно редактора временных диаграмм **Simulation Waveform Editor** с именем по умолчанию Waveform1.vwf.

Созданный файл необходимо сохранить, используя команду **Save As** меню **File**. Программа автоматически предложит сохранить файл с именем, совпадающим с именем файла верхнего уровня проекта, присвоив ему расширение *.vwf. Необходимо обратить внимание на наличие флажка **Add file**

to current project (добавить файл к текущему проекту). Если он поставлен, то система автоматически включит созданный файл в текущий проект.

Для удобства на область временных диаграмм нанесена временная сетка, предназначенная для визуальной привязки сигналов к конкретным временным интервалам. Используя команду **Grid Size** меню **Edit**, можно изменить шаг временной сетки. Используя команду **Set End Time** меню **Edit**, можно корректировать длительность процесса моделирования.

5.2. Сформировать файл временной диаграммы введением на диаграмму входных и выходных сигналов проекта. В меню **Edit** следует активизировать команду **Insert Node or Bus** (вставить узел или шину) или же двойным щелчком по области входных и выходных сигналов запускается диалоговое окно, где можно указывать необходимые сигналы.

Для упрощения процесса ввода предусмотрена система поиска цепей (портов) **Node Finder**, запускаемая одноименной кнопкой в этом же окне. Система позволяет ввести цепи и порты в файл временных диаграмм текущего проекта. Общий список вызывается нажатием кнопки **List** в окне **Node Finder** (рис. 1.4), после чего в поле загружаются все узлы проекта с учетом параметра, указанного в поле **Filter**.

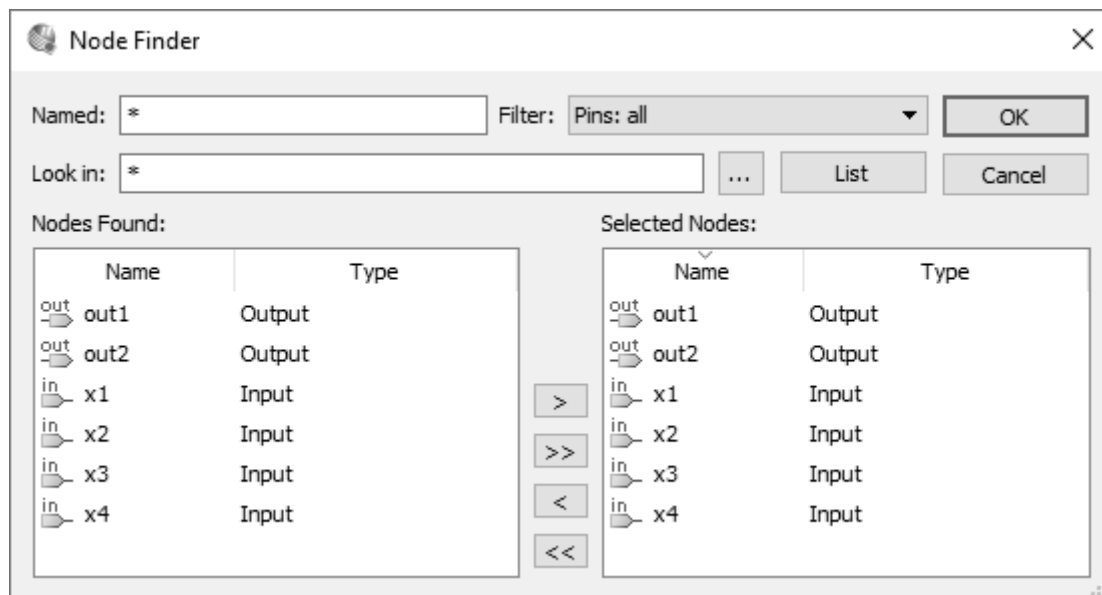


Рис. 1.4. Окно выбора цепей моделирования

Интересующие разработчика в процессе моделирования цепи переносятся в поле **Selected Nodes** нажатием соответствующих кнопок в этом окне. В результате в поле временных диаграмм редактора появляются оси для всех выбранных сигналов. Осям временных диаграмм входных сигналов задаются требуемые значения с помощью инструментария, сгруппированного в меню

пиктограмм. Оси временных диаграмм выходных сигналов имеют неизвестное на данный момент состояние, поэтому им по умолчанию присвоено значение X (forcing unknown). Окно редактора векторных временных диаграмм приведено на рис. 1.5.



Рис. 1.5. Окно редактора векторных временных диаграмм

5.3. Запуск функционального моделирования проекта осуществляется активизацией команды **Simulation / Run Functional Simulation**, а временного – команды **Simulation / Run Timing Simulation** или соответствующих пиктограмм в окне **Simulation Waveform Editor**.

После завершения моделирования его результаты будут выведены в дополнительное окно **Simulation Waveform Editor** (рис. 1.6).

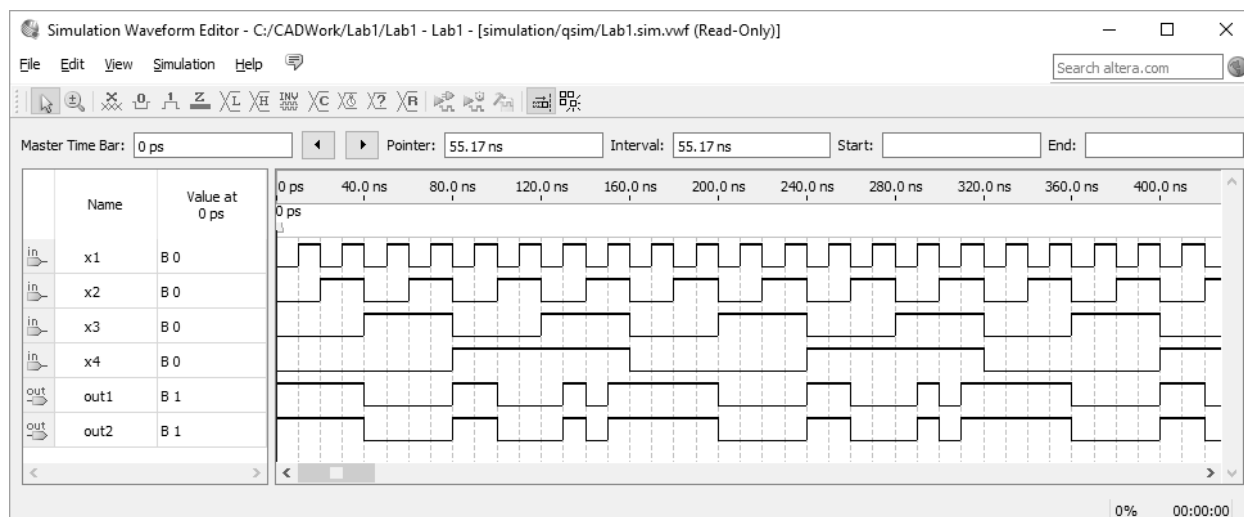


Рис. 1.6. Окно результатов симуляции

Результирующая временная диаграмма доступна только для чтения. Файл с результатами <имя проекта>.sim.vwf расположен в папке ...\\simulation\\qsim\\, созданной системой моделирования в рабочей папке проекта.

Задание на работу

Спроектировать комбинационную схему, реализующую функцию от четырех переменных, заданную набором входных данных, на которых она принимает единичные значения. Необходимо составить таблицу истинности функции, выполнить минимизацию функции с использованием карт Карно или метода Квайна–Мак-Класки, основанного на применении операций склеивания и поглощения.

Компонент x_4 входного вектора (x_4, x_3, x_2, x_1) следует считать старшим двоичным разрядом.

Необходимо разработать два варианта реализации комбинационной схемы, отличающихся элементным базисом. В первом случае в качестве базиса выбрать примитивы not (НЕ), band* (*-НЕ-И), nand* (*-И-НЕ), а во втором – not (НЕ), bor* (*-НЕ-ИЛИ), nor* (*-ИЛИ-НЕ), где * – количество входов элемента. Соответствующие примитивы расположены в библиотеке САПР Quartus II в разделе Primitives / Logic.

Оба варианта реализации собрать в одном графическом файле проекта, предусмотрев два соответствующих выхода. Пример реализации функции, принимающей единичные значения на наборах 0, 1, 2, 3, 8, 9, 13, 15, приведен на рис. 1.3, а временная диаграмма, полученная в результате функционального моделирования – на рис. 1.6.

Таблица 1.1

Номер	Функция	Номер	Функция
1	0, 1, 2, 3, 10, 11, 14	7	0, 2, 4, 5, 6, 7, 9
2	0, 1, 2, 9, 10, 14	8	1, 2, 3, 12, 13
3	0, 1, 2, 5, 6, 10, 13, 14	9	1, 3, 4, 5, 6, 7, 8, 12, 13
4	0, 1, 3, 6, 7, 11, 14, 15	10	1, 3, 4, 5, 6, 7, 10, 14
5	0, 1, 2, 4, 6, 8, 9, 10, 11	11	1, 3, 5, 7, 8, 12, 13, 15
6	0, 1, 5, 8, 9, 13, 14, 15	12	4, 6, 9, 10, 11, 12, 13

Варианты заданий приведены в табл. 1.1, где указаны десятичные значения векторов входных переменных (x_4, x_3, x_2, x_1) , на которых переключательная функция $y(x_4, x_3, x_2, x_1)$ принимает значения логической «1». На других входных наборах функция равна логическому «0».

Последовательность выполнения работы

1. Разработать комбинационную схему в соответствии с заданием. Оформить разработанную схему с учетом требований ГОСТ.

2. Создать проект в САПР Quartus II, подготовить описание схемы, опираясь на средства графического редактора. Схема должна состоять из двух частей, реализующих одинаковую функцию, но отличающихся использованным базисом. Соответственно, проект должен иметь четыре входа и два независимых выхода, как показано на рис. 1.3. При составлении схемы следует использовать библиотечные примитивы.

3. Компилировать проект, исправить ошибки, если они есть.

4. Выполнить функциональное моделирование проекта, убедиться в правильности работы схемы. При моделировании обеспечить полный перебор возможных значений векторов входных сигналов.

5. Выполнить временное моделирование, провести анализ временной диаграммы. Обратить внимание на возможное различие выходных сигналов фрагментов схемы, реализованных в разных базисах, а также отличия от временной диаграммы, полученной при функциональном моделировании.

6. Загрузить проект в учебную плату в соответствии с рекомендациями прил. 1. Входные сигналы подключить к движковым переключателям, а выходные – к светодиодам светодиодной линейки. Проверить работу схемы при полном переборе входных данных.

Содержание отчета

Отчет должен быть оформлен в соответствии с принятыми в СПбГЭТУ требованиями и содержать следующую информацию:

1. Описание процесса синтеза логической схемы.
2. Функциональную схему, оформленную с учетом требований ГОСТ.
3. Схему узла, представленную соответствующим файлом проекта САПР Quartus II.
4. Результаты функционального и временного моделирования, их сравнительный анализ.
5. Список и назначение внешних контактов, описание процесса макетирования.
6. Выводы по работе.

Лабораторная работа 2

ПРОЕКТИРОВАНИЕ КОМБИНАЦИОННОГО УЗЛА НА ОСНОВЕ ДЕШИФРАТОРА И МУЛЬТИПЛЕКСОРА

Цель работы – получить практические навыки в использовании дешифратора и мультиплексора для воспроизведения произвольных логических функций.

Краткие теоретические сведения

Дешифратор с прямыми выходами формирует на своих выходах полную систему конъюнктивных термов от аргументов, подаваемых на информационные входы. Дополнив схему элементом ИЛИ, соединенным с выходами дешифратора, соответствующими конституентам «1», можно получить комбинационный узел, реализующий переключательную функцию в совершенной дизъюнктивной нормальной форме.

Если переключательная функция имеет меньше нулевых значений, чем единичных, то выгоднее использовать дополнительный элемент ИЛИ-НЕ, на входах которого собирают сигналы с выходов дешифратора, соответствующих конституентам «0».

Если использован дешифратор с инверсными выходами, то во втором каскаде комбинационного узла сигналы собирают на элементе И-НЕ или на элементе И. Если заданная функция имеет меньше единичных значений, то применяют элемент И-НЕ, на который подают инверсные сигналы конституент «1». Если переключательная функция имеет меньше нулевых значений, то используют элемент И и передают на него инверсные сигналы конституент «0».

Мультиплексор соединяет логически со своим выходом y тот информационный вход d_i , номер i которого задан кодом на входах настройки X . Мультиплексор реализует переключательную функцию

$$y = \bigvee_{i=0}^{i=r} d_i \& k_i, \quad (2.1)$$

где k_i – конституента «1» для i -го набора настроечных переменных d_1, d_2, \dots, d_n ; $r = 2^n - 1$ – максимальное значение индекса i .

Если на входы d_i мультиплексора подавать константы «0» и «1» в соответствии со значениями заданной переключательной функции y , то выражение (2.1) становится совершенной дизъюнктивной нормальной формой функ-

ции y от аргументов x_1, x_2, \dots, x_n . Получающаяся комбинационная схема имеет структуру «константа – мультиплексор».

Более экономична по затратам оборудования структура «функция – мультиплексор». В ней на входы настройки мультиплексора подают только часть входных переменных x_1, x_2, \dots, x_n , а из остальных формируют промежуточные переменные.

Декомпозицию функции y выполняют либо аналитически, пользуясь разложением по Шеннону, либо графически на картах Карно, либо таблично перестановкой и соединением строк исходной таблицы.

Аналитические преобразования основаны на разложении функции по Шеннону:

$$y(x_0, \dots, x_i, \dots, x_n) = \overline{x_i} \& y(x_0, \dots, 0, \dots, x_n) \vee x_i \& y(x_0, \dots, 1, \dots, x_n).$$

Функция разлагается по тем аргументам x_i , которые предполагается подать на настроечные входы мультиплексора. Подфункции-множители реализуются отдельно и подаются на информационные входы мультиплексора.

Выделение подфункций по карте Карно дает лучшие результаты, так как вследствие обозримости всей функции удастся найти группировку аргументов, которая максимально упрощает схему каскада «функция».

Задание на работу

Лабораторная работа состоит из двух частей.

Часть 1. Реализовать комбинационную схему в соответствии с заданием к лаб. раб. 1 с использованием дешифратора и мультиплексора. Сравнить варианты реализации.

В библиотеке системы Quartus II наряду с примитивами имеются некоторые заранее спроектированные и включенные в отдельную библиотеку более сложные цифровые узлы. Такие функционально завершенные компоненты более высокого уровня сложности, допускающие настройку параметров пользователем, называются мегафункциями. В библиотеке мегафункций имеются типовые узлы мультиплексоров и дешифраторов, которые и следует использовать при выполнении работы.

Спроектированную в лаб. раб. 1 схему следует использовать в текущей работе как библиотечный блок. Схема верхнего модуля проекта должна включать в себя три функционально законченных независимых узла: блок,

спроектированный в лаб. раб. 1; блок, реализующий функцию с использованием мультиплексора; блок, реализующий функцию на базе дешифратора.

Часть 2. Разработать комбинационную схему управления индикацией, построенной на светодиодах.

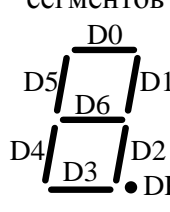
Светодиоды изготавливаются на основе специальных полупроводниковых материалов, пропускание тока через такой диод вызывает его свечение. Цвет свечения зависит от материала, из которого изготовлен диод, яркость свечения зависит от величины тока, протекающего через него. Из нескольких диодов составляются индикаторы и матрицы, отображающие буквы, цифры и служебные символы.

Широко применяются семисегментные индикаторы, в которых семь сегментов-диодов расположены так, что при зажигании определенной их комбинации высвечивается тот или иной символ.

Выпускаются семисегментные индикаторы (ССИ) с общим анодом или общим катодом. В зависимости от типа ССИ зажиганию светодиодного сегмента соответствует единичное или нулевое значение логического управляющего сигнала.

Формат отображения цифр и пример формирования кода для изображения цифр для схемы включения сегмента единичным значением приведены на рис. 2.1.

Нумерация
сегментов



Цифра	D0	D1	D2	D3	D4	D5	D6	DP
0	1	1	1	1	1	1	0	0
1	0	1	1	0	0	0	0	0
2	1	1	0	1	1	0	1	0
...						

Рис. 2.1. Семисегментный индикатор

На вход разрабатываемой комбинационной схемы управления семисегментным индикатором должен подаваться 4-разрядный двоичный код, а на 8-разрядном выходе – формироваться код, обеспечивающий необходимое изображение в соответствии с расположением сегментов на индикаторе.

Комбинационная схема может быть реализована любым известным способом: с помощью элементарного логического базиса, с использованием мультиплексоров или на основе дешифраторов.

Последовательность выполнения работы

Часть 1:

1. Создать проект. Подготовить схему, разработанную в лаб. раб. 1 для включения в проект. Для этого необходимо выполнить следующие действия:

1.1. Скопировать схемный файл *.bdf в директорию проекта, открыть его в редакторе.

1.2. Создать символ для схемы верхнего уровня, для чего воспользоваться командой **File / Create / Update / Create Symbol File**. После выполнения этой команды при обращении к библиотеке элементов командой меню **Insert / Symbol** наряду со стандартной библиотекой САПР QuartusII станет доступен и этот компонент. Он будет расположен в папке Project (рис. 2.2).

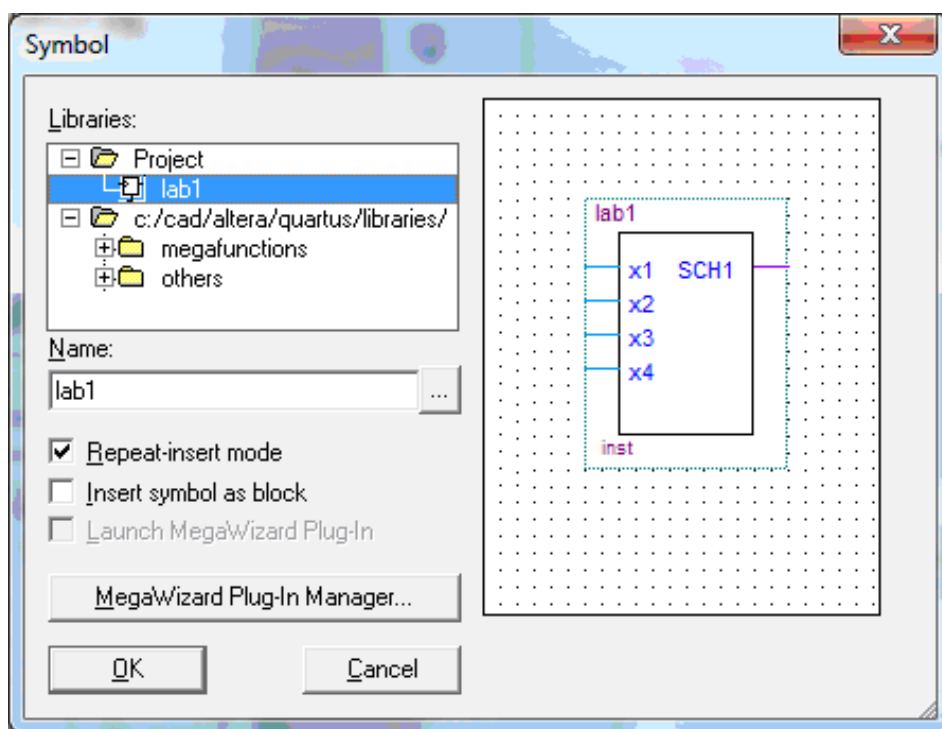


Рис. 2.2. Библиотечный элемент Lab1

2. Подготовить схему в соответствии с заданием, опираясь на средства графического редактора. Схема должна состоять из трех частей, выполняющих одинаковую функцию, но отличающихся реализацией. Первая часть связана с непосредственным использованием созданного блока Lab1, вторая – схема, реализующая логическую функцию с использованием мультиплексора, третья – реализующая эту же функцию на базе дешифратора. Соответственно проект должен иметь три независимых выхода. При разработке схемы следует использовать библиотечные мегафункции.

Мегафункция дешифратора. В библиотеке LPM-модулей дешифратор представлен мегафункцией LPM_DECODE.

Порты и параметры этой мегафункции: вход данных data [], сигнал разрешения enable, сигнал синхронизации для конвейерного использования clock, выходы – eq[]. Настройка мегафункции предполагает указание только тех входных и выходных линий, которые необходимы в разрабатываемой схеме. Очевидно, что в данном случае будут использованы порты data [3..0] и eq[7..0]. Утилита настройки параметров запускается нажатием кнопки ОК в окне *Symbol* (рис. 2.3).

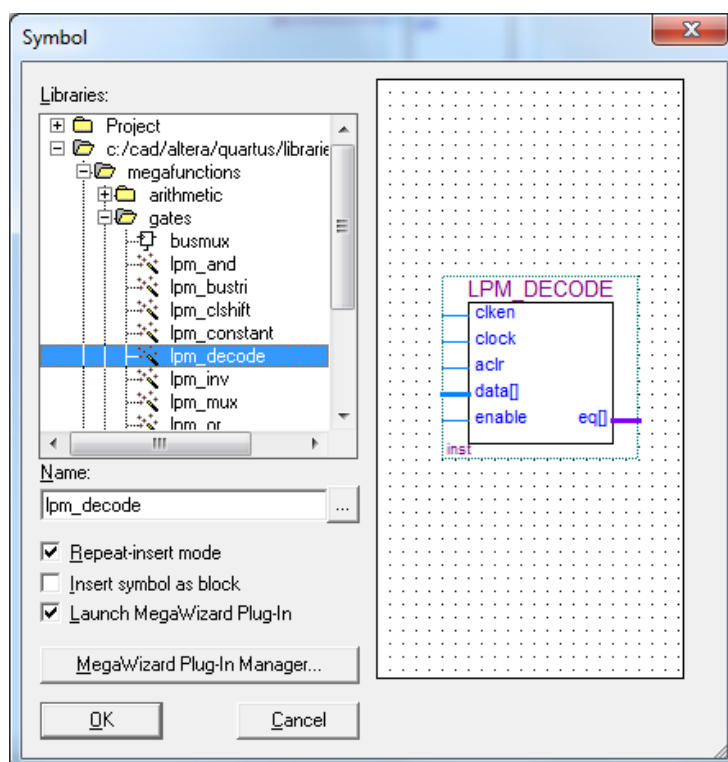


Рис. 2.3. Библиотека мегафункций

Мегафункция мультиплексора. Мегафункция LPM_MUX предполагает указание количества входов данных data [] и указание разрядности данных по каждому входу. Очевидно, что в данном случае будут использованы 16 одно-разрядных портов data.

Мегафункция константы. В библиотеке LPM-модулей генератор константы представлен мегафункцией LPM_CONSTANT. Значение константы задается в процессе настройки модуля.

Информационные входы и выходы мегафункций, как правило, объединены в шины. Примеры подключения к таким шинам других элементов устройства иллюстрирует вариант схемы, приведенный на рис. 2.4. Входная

4-битная шина $x[4..1]$ подключена ко входу дешифратора, представленного мегафункцией LPM_DECODE0, а также ко входу блока Lab1, причем на вход блока подаются отдельные линии шины. Имя шины обязательно записывается с использованием квадратных скобок, в которых указываются номера линий шины.

При работе с графическим редактором системы Quartus II шины обязательно должны обозначаться жирными линиями.

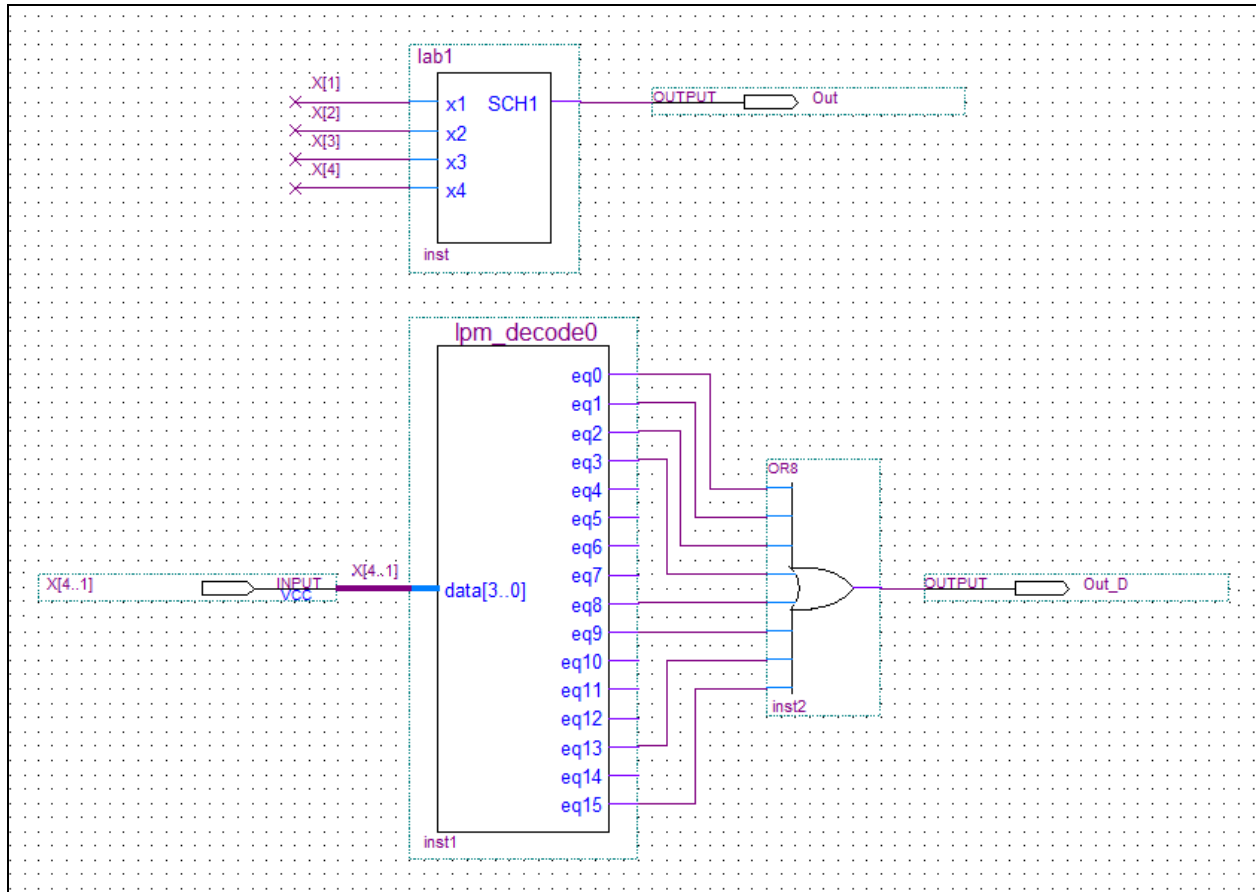


Рис. 2.4. Реализация комбинационной схемы с использованием дешифратора

3. Компилировать проект, исправить ошибки, если они есть.

4. Выполнить временное моделирование, провести анализ временной диаграммы. Объяснить возможное различие выходных сигналов фрагментов схемы, реализованных на стандартном и самостоятельно разработанном модулях. При моделировании обеспечить полный перебор возможных комбинаций входных сигналов.

5. Загрузить проект в учебную плату, подключив входные сигналы $x[4..1]$ к переключателям, а выходные – к светодиодам светодиодной линейки. Проверить работу схемы.

Часть 2:

1. Разработать функциональную схему преобразователя 4-разрядного двоичного кода в код семисегментного индикатора, обеспечивающую включение светодиода единичным сигналом и вывод данных в шестнадцатеричной форме (цифры от 1 до 9, буквы A, b, C, d, E, F).

2. Создать новый проект в САПР Quartus II. С использованием средств графического редактора подготовить схему преобразователя. Оформить ее в виде отдельного библиотечного элемента для включения в последующие проекты.

3. Выполнить загрузку спроектированного модуля в учебную плату и протестировать его работу. Входные сигналы модуля подключить к движковым переключателям; выходные сигналы – к сегментам семисегментного индикатора.

Содержание отчета

Отчет должен содержать следующую информацию.

По части 1:

1. Функциональную схему, реализующую заданную функцию с использованием мультиплексора и дешифратора, оформленную с учетом требований ГОСТ.

2. Схему узла, представленную соответствующим файлом проекта САПР Quartus II.

3. Результаты функционального и временного моделирования, сравнительный анализ выходных сигналов по трем выходам.

4. Список и назначение внешних контактов, описание процесса макетирования.

По части 2:

1. Функциональную схему преобразователя 4-разрядного двоичного кода в код семисегментного индикатора.

2. Схему узла, представленную соответствующим файлом проекта САПР Quartus II.

3. Список и назначение внешних контактов, описание процесса макетирования.

Также отчет должен содержать общие выводы по работе.

Лабораторная работа 3

ПРОЕКТИРОВАНИЕ ТИПОВОГО КОМБИНАЦИОННОГО УЗЛА

Цель работы – закрепить навыки синтеза комбинационных схем в процессе проектирования логической схемы типового комбинационного узла с заданным функциональным назначением в среде Quartus II.

Краткие теоретические сведения

К типовым функциональным узлам комбинационного типа относятся: дешифраторы, шифраторы, мультиплексоры, демультиплексоры, преобразователи кодов, арифметико-логические устройства и другие.

Исходная информация для проектирования представляется описанием функционирования узла, на основании которого может быть построена таблица истинности.

Рассмотрим формирование таблицы истинности для пятиразрядного указателя старшей единицы, имеющего также вход разрешения работы E . Количество входов X и выходов Y у указателя старшей единицы одинаково. При наличии на входах нескольких возбужденных линий (запросов) на выходе будет возбуждена лишь одна, соответствующая старшему запросу. Таблица истинности для пятиразрядного узла приведена в табл. 3.1.

Таблица 3.1

E	x_4	x_3	x_2	x_1	x_0	y_4	y_3	y_2	y_1	y_0
1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
1	0	0	0	1	*	0	0	0	1	0
1	0	0	1	*	*	0	0	1	0	0
1	0	1	*	*	*	0	1	0	0	0
1	1	*	*	*	*	1	0	0	0	0
0	*	*	*	*	*	0	0	0	0	0

Разработка схемы указателя старшей единицы сводится к реализации логических схем, обеспечивающих формирование функций y_i . Процесс проектирования выполняется в соответствии с этапами, перечисленными в лаб. раб. 1. Так, для выходных функций будут получены следующие значения:

$$\begin{aligned}y_0 &= E \overline{x_4} \overline{x_3} \overline{x_2} \overline{x_1} \overline{x_0}; & y_1 &= E \overline{x_4} \overline{x_3} \overline{x_2} x_1; \\y_2 &= E \overline{x_4} \overline{x_3} x_2; & y_3 &= E \overline{x_4} x_3; & y_4 &= E x_4.\end{aligned}$$

Вариант схемной реализации устройства приведен на рис. 3.1, а результаты функционального моделирования – на рис. 3.2.

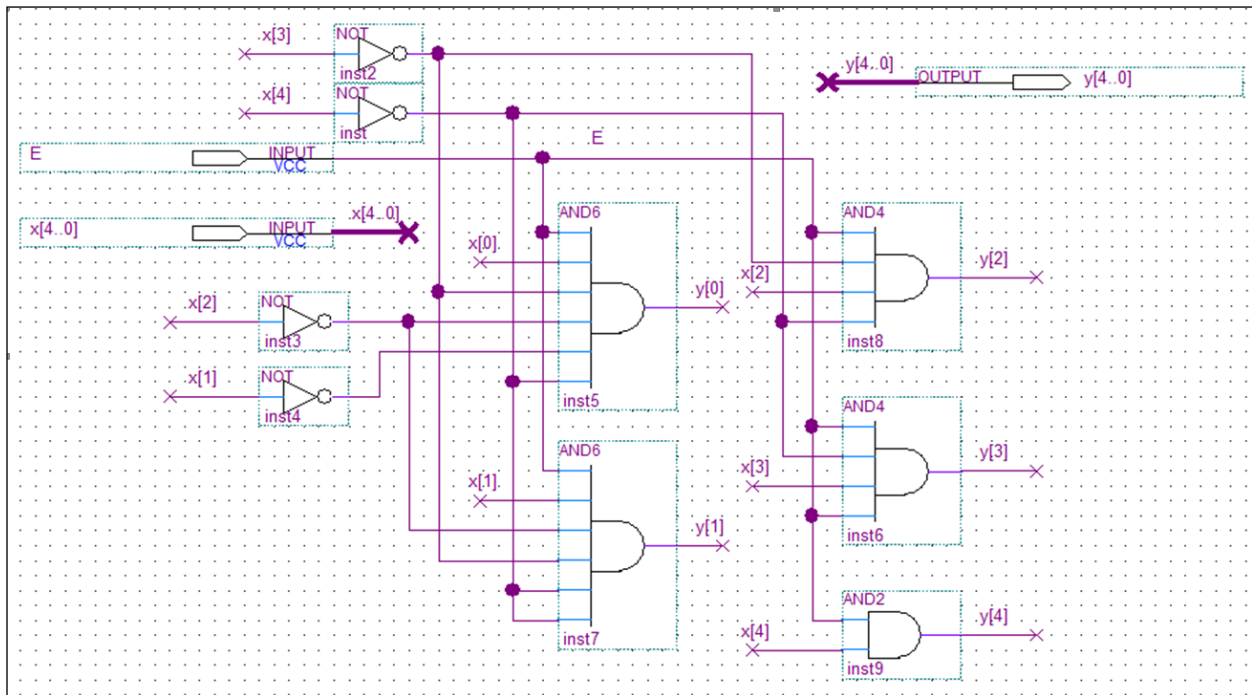


Рис. 3.1. Реализация указателя старшей единицы

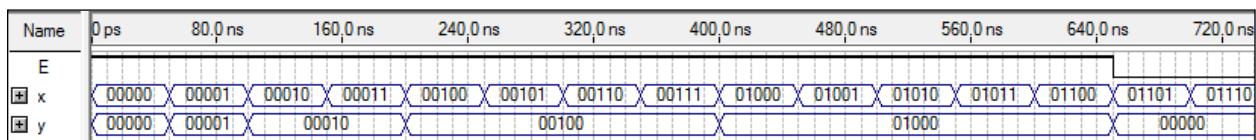


Рис. 3.2. Временная диаграмма работы указателя старшей единицы

Устройство реализовано на логических элементах И и НЕ. В библиотеке примитивов САПР Quartus II представлены примитивы только с четным количеством входов, для реализации функции И с нечетным количеством входов используется дублирование линии одного из аргументов.

Задание на лабораторную работу

Разработать схему типового комбинационного узла в соответствии с заданием, в качестве базовых элементов использовать примитивы библиотеки САПР Quartus II. Варианты заданий:

1. Дешифратор с тремя информационными входами и входом разрешения работы E_n .
2. Приоритетный шифратор с 8 входами и входом разрешения работы E_n .
3. Мультиплексор с двумя адресными входами.
4. Демультимплексор с двумя адресными входами.
5. Схема сравнения двух 3-разрядных кодов на «равно» и «больше».
6. Полный двухразрядный сумматор комбинационного типа.
7. Преобразователь 4-разрядного двоичного кода в код Грея.

Таблица 3.2

Двоичный код	Код Грея	Код Хемминга	Двоичный код	Код Грея	Код Хемминга
0000	0000	0000 000	1000	1100	1000 101
0001	0001	0001 011	1001	1101	1001 110
0010	0011	0010 110	1010	1111	1010 011
0011	0010	0011 101	1011	1110	1011 000
0100	0110	0100 111	1100	1010	1100 010
0101	0111	0101 100	1101	1011	1101 001
0110	0101	0110 001	1110	1001	1110 100
0111	0100	0111 010	1111	1000	1111 111

8. Преобразователь 4-разрядного кода Грея в двоичный код.

9. Преобразователь 4-разрядного двоичного кода в код Хемминга (4–3).

10. Мажоритарный элемент на три входа.

Соответствие двоичного кода, кода Грея и кода Хемминга приведены в табл. 3.2.

Последовательность выполнения работы

1. Разработать функциональную схему узла в соответствии с заданием.
2. Создать проект в САПР Quartus II и с использованием средств графического редактора подготовить схему комбинационного узла в соответствии с заданием. Компилировать проект, исправить ошибки, если они есть.
3. Провести функциональное моделирование проекта, обеспечив полный перебор значений входных сигналов, убедиться в корректности работы схемы.
4. Выполнить временное моделирование, провести анализ полученной временной диаграммы.
5. Загрузить проект в учебную плату, подключив входные сигналы к переключателям, а выходные – к светодиодам светодиодной линейки. Проверить работу схемы.

Содержание отчета

Отчет должен содержать следующую информацию.

1. Функциональную схему узла в соответствии с заданием, оформленную с учетом требований ГОСТ.
2. Схему узла, представленную схемным файлом проекта САПР Quartus II.
3. Временные диаграммы, полученные в ходе функционального и временного моделирования, результаты их анализа.
4. Описание процесса макетирования.
5. Выводы по работе.

Лабораторная работа 4

ТРИГГЕРНЫЕ УСТРОЙСТВА

Цель работы – исследовать особенности функционирования основных типов триггерных устройств: одноступенчатых D-триггеров со статическим и динамическим управлением, RS-триггера, JK-триггера с динамическим управлением, T-триггера.

Краткие теоретические сведения

Триггер – бистабильная ячейка, состоянием которой можно управлять, введя для этого специальные входы. По логическому функционированию выделяют RS-, D-, T- и JK-триггеры. Существуют и комбинированные триггеры, в которых совмещаются свойства триггеров различных типов.

Триггер RS имеет два входа – установки в единицу (S) и установки в ноль (R). Одновременная подача сигналов установки S и сброса R не допускается, комбинация сигналов установки $S = 1$ и сброса $R = 1$ запрещена.

D-триггер имеет один вход, его состояние повторяет входной сигнал, но с задержкой, определяемой тактовым сигналом.

T-триггер имеет один информационный вход и изменяет свое состояние каждый раз при поступлении входного сигнала.

JK-триггер – универсальный триггер, входы установки J и сброса K подобны соответствующим входам RS-триггера. Однако JK-триггер допускает одновременную подачу сигналов на оба эти входа $J = K = 1$. В этом случае он работает как T-триггер.

Логическое функционирование триггеров может быть описано различными способами: таблицами истинности, картами Карно, характеристическими уравнениями, диаграммами состояний, словарями.

Словари триггеров (табл. 4.1) отражают наличие у триггера двух устойчивых состояний и условия перехода из одного состояния в другое и являются удобными инструментами при проектировании схем, содержащих триггеры.

По способу записи информации различают асинхронные (неактируемые) и синхронные (актируемые) триггеры. В асинхронных триггерах переход в новое состояние вызывается изменениями входных информационных сигналов. В синхронных триггерах имеется специальный тактовый вход, и переход происходит только при подаче на этот вход тактовых сигналов.

Таблица 4.1

Переход	RS-триггер		JK-триггер		D-триггер	T-триггер
	R	S	J	K	D	T
0→0	*	0	0	*	0	0
0→1	0	1	1	*	1	1
1→0	1	0	*	1	0	1
1→1	0	*	*	0	1	0

По способу восприятия тактовых сигналов триггеры делятся на управляемые уровнем и управляемые фронтом. При управлении уровнем, или статическом управлении, триггер воспринимает входные информационные сигналы при одном уровне тактового сигнала и остается в неизменном состоянии при другом. При управлении фронтом (при динамическом управлении) переключение триггера происходит только в момент перепада тактового сигнала (на его фронте или спаде).

В САПР Quartus II доступны примитивы триггеров, приведенные на рис. 4.1.

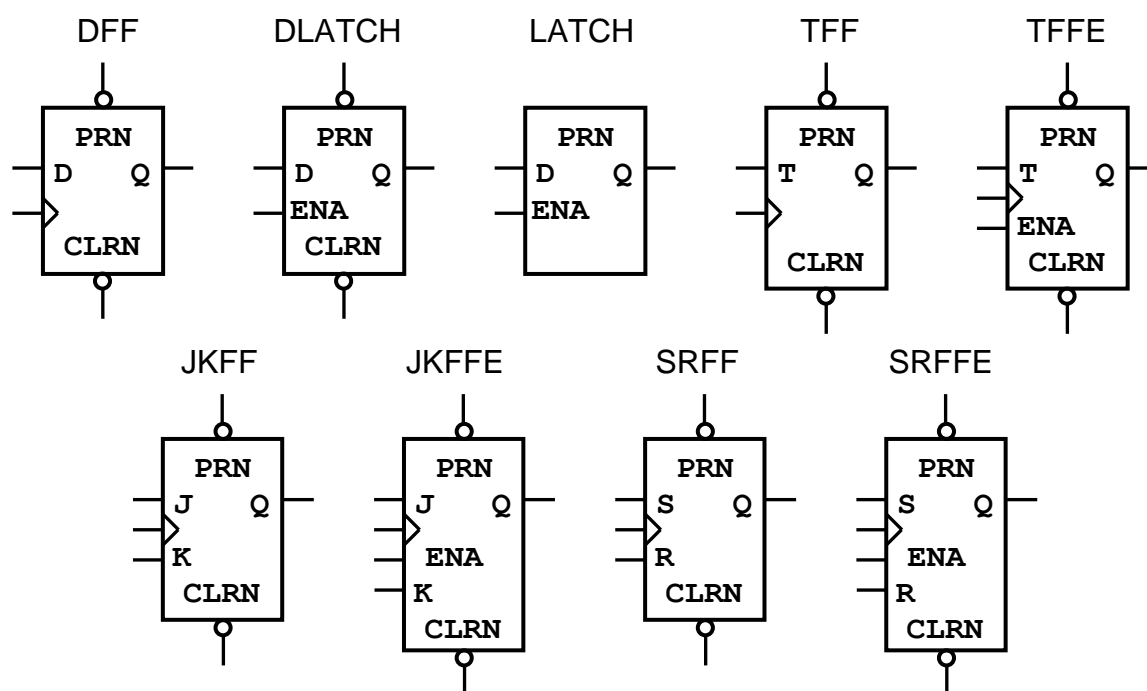


Рис. 4.1. Примитивы триггеров САПР Quartus II

Входы CLRN и PRN являются входами асинхронного сброса и установки соответственно, активный уровень сигнала – «0»; ENA – вход разрешения работы, активный уровень сигнала – «1»; входы, помеченные как срабатывающие по переднему фронту, – входы синхронизации. Примитивы DLATCH и LATCH управляются уровнем сигнала на входе ENA.

Задания на лабораторную работу

1. Исследовать D-триггер.

1.1. Исследуются: одноступенчатый D-триггер со статическим управлением и D-триггер с динамическим управлением (примитивы DLATCH и DFF соответственно). Разработать комбинационную схему, обеспечивающую формирование сигналов в соответствии с временными диаграммами, приведенными на рис. 4.2. Номер диаграммы соответствует номеру бригады.

Дополнительно предусмотреть возможность подачи на исследуемый триггер сигнала асинхронного сброса (для четных вариантов заданий) или установки (для нечетных вариантов).

1.2. Выполнить функциональное моделирование схемы. Сравнить выходные сигналы D-триггеров со статическим и динамическим управлением. Объяснить различия во временных диаграммах, если они обнаружены.

1.3. Проверить работу на макетной плате. Перед выполнением макетного эксперимента добавить в схему счетчик, понижающий частоту тактового сигнала. Разрядность счетчика определяется исходя из частоты генератора, установленного на макете. Работу исследуемых триггеров наблюдать с помощью осциллографа.

2. Исследовать RS-триггер.

2.1. Для исследования RS-триггера (примитив SRFF) подготовить временную диаграмму на 16 тактов, демонстрирующую особенности его работы. Входные сигналы должны быть такими, чтобы исследовать все режимы работы: синхронный прием по информационным входам, хранение информации, асинхронное переключение триггера по сигналам на входах CLRN и PRN.

2.2. Разработать комбинационную схему, обеспечивающую формирование сигналов в соответствии с подготовленной временной диаграммой. Выполнить функциональное моделирование работы RS-триггера.

2.3. Проверить работу на макетной плате, понизив частоту тактового сигнала с использованием счетчика. Работу триггеров наблюдать с помощью осциллографа.

3. Исследовать JK-триггер.

Для исследования JK-триггера (примитив JKFF) реализовать временные диаграммы, приведенные на рис. 4.3, выбрав вариант в соответствии с номером бригады.

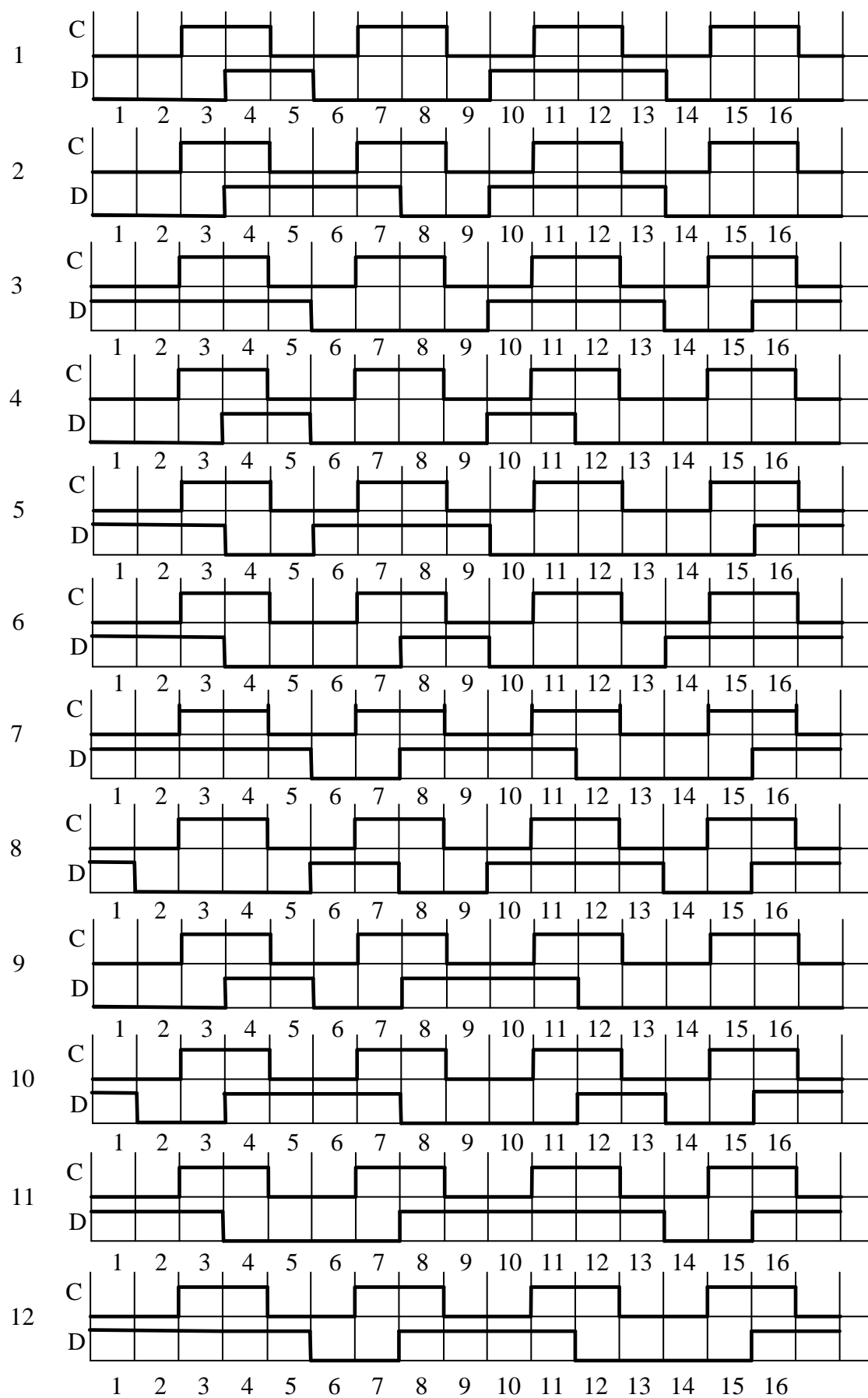


Рис. 4.2. Временные диаграммы для исследования D-триггера

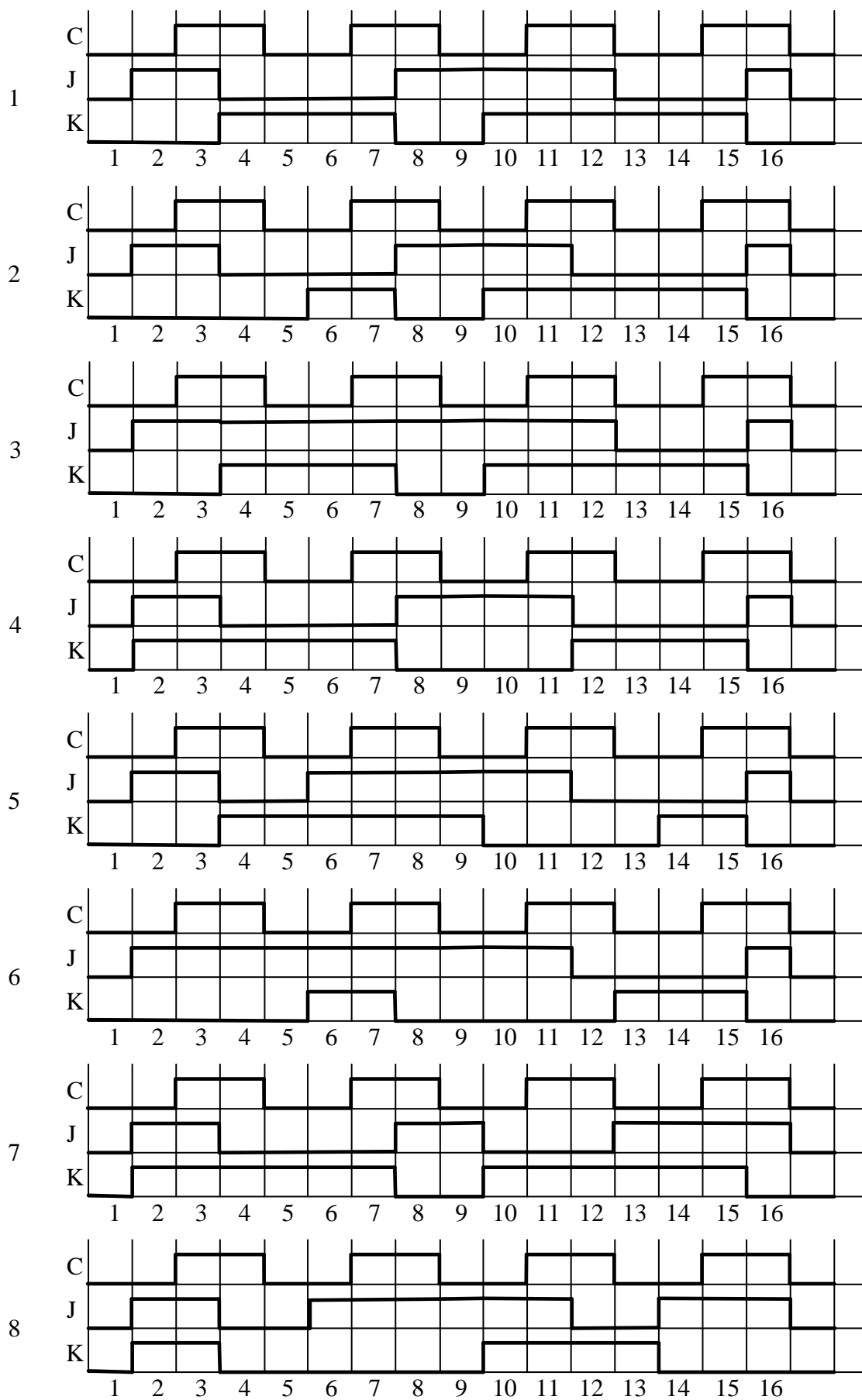


Рис. 4.3. Временные диаграммы для исследования JK-триггера (часть 1)

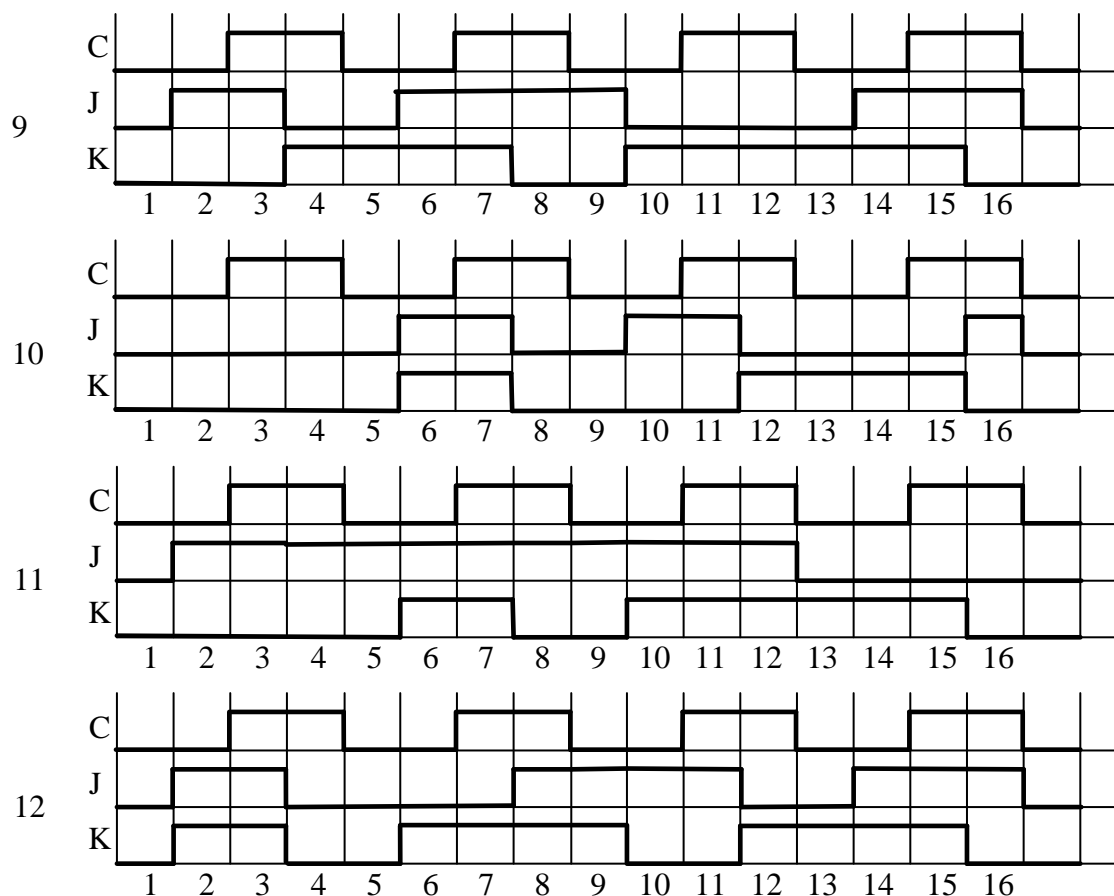


Рис. 4.3. Временные диаграммы для исследования JK триггера (часть 2)

Для исследования JK-триггера выполнить функциональное моделирование схемы. Проверить работу триггера на макетной плате, наблюдая ее с помощью осциллографа.

Последовательность выполнения работы

По каждому пункту задания на работу выполнить следующие действия.

1. Создать проект в системе Quartus II и с использованием средств графического редактора подготовить схему для исследования заданного типа триггера. Схема должна включать в себя комбинационный узел, формирующий тестовую последовательность сигналов, и примитив исследуемого триггера. Такой узел может быть построен любым способом, изученным при выполнении лаб. раб. 1 и 2.

При выполнении макетного эксперимента для автоматического формирования тестовой последовательности сигналов в схему дополнительно вводится 4-разрядный счетчик, вход которого подключается к генератору тактовых сигналов, а выходы подключаются ко входам комбинационного узла.

В библиотеке LPM-модулей счетчик представлен мегафункцией LPM_COUNTER, находящейся в разделе *megafunction / arithmetic*. В этой

мегафункции используются порты: вход clock и выход q[], разрядность выходного сигнала – 4.

Пример реализации схемы, обеспечивающей непрерывную генерацию функции F с использованием мультиплексора, приведен на рис. 4.4, а сформированная последовательность тестового сигнала – на рис. 4.5.

Код на выходе счетчика x[3..0] соответствует номеру такта работы устройства.

2. Провести исследования схемы в режиме функционального моделирования. Время моделирования устройства должно быть не менее длительности двух периодов его работы.

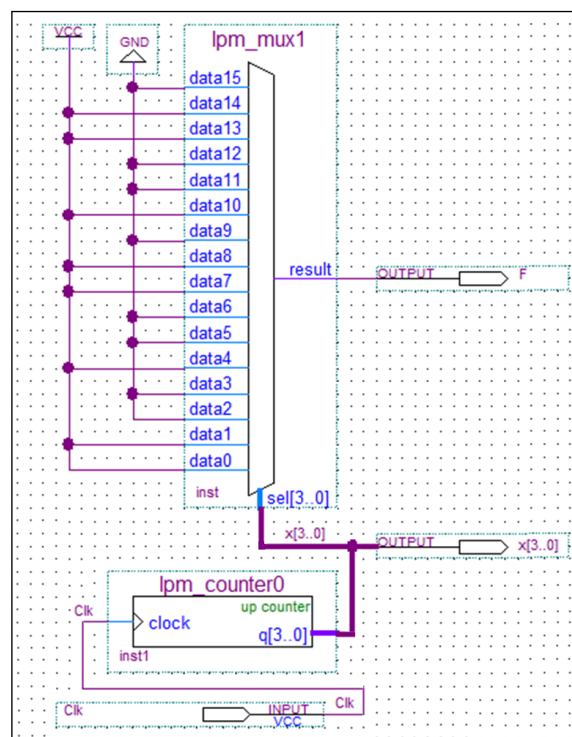


Рис. 4.4. Схема формирования тестовой последовательности сигналов

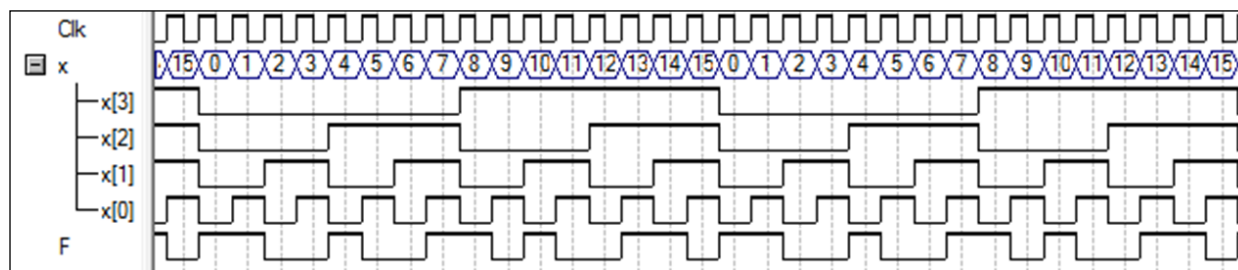


Рис. 4.5. Последовательность тестового сигнала F

3. Провести исследования триггера в режиме макетирования. В качестве тактового сигнала использовать сигнал внешнего генератора. Сигналы со входов и выхода триггера и тактовый сигнал вывести на контакты разъема.

Содержание отчета

Отчет должен содержать следующую информацию:

1. Схему комбинационного узла, формирующего последовательности тестовых сигналов, оформленную с учетом требований ГОСТ.
2. Схемные файлы проектов исследования D-, RS-, JK- и T-триггеров. Результаты их функционального моделирования.
3. Описание процесса макетирования. Скриншоты полученных осциллограмм.
4. Выводы по работе.

Лабораторная работа 5

ИССЛЕДОВАНИЕ РЕГИСТРОВ

Цель работы – исследовать особенности проектирования регистров различного типа, закрепить навыки синтеза и экспериментального исследования узлов в среде Quartus II.

Краткие теоретические сведения

Регистры – это узлы последовательностного типа, выполняющие операции приема, выдачи, хранения, сдвига, поразрядные логические операции. Регистры строятся на базе триггеров и логических элементов.

Главным классификационным признаком является способ приема и выдачи данных. По этому признаку различают:

- параллельные (статические): прием и выдача слов производятся по всем разрядам одновременно);
- последовательные (сдвигающие): слова принимаются и выдаются разряд за разрядом, перемещение слов по разрядной сетке выполняется по тактовому сигналу;
- параллельно-последовательные: имеют входы-выходы последовательного и параллельного типа. Имеются варианты с последовательным входом и параллельным выходом (SIPO, Serial Input – Parallel Output), параллельным входом и последовательным выходом (PISO, Parallel Input – Serial Output), а также варианты с возможностью любого сочетания способов приема и выдачи слов.

Общими для разрядов регистров обычно являются цепи тактирования, сброса/установки, разрешения выхода или приема, т. е. цепи управления.

Регистр, реализующий функцию сдвига, может быть нереверсивным (с однонаправленным сдвигом) или реверсивным (с возможностью сдвига в обоих направлениях). Направление сдвига в регистре – не геометрическое понятие и определяется сдвигом в сторону старших или младших разрядов. Варианты реализации функции сдвига приведены на рис. 5.1.

В сдвигающих регистрах, не имеющих логических элементов в междуразрядных связях, нельзя применять одноступенчатые триггеры, управляемые уровнем, поскольку некоторые триггеры могут за время действия разрешающего уровня синхросигнала переключиться неоднократно, что недопустимо для последовательного соединения: слово или его часть могут сместиться по

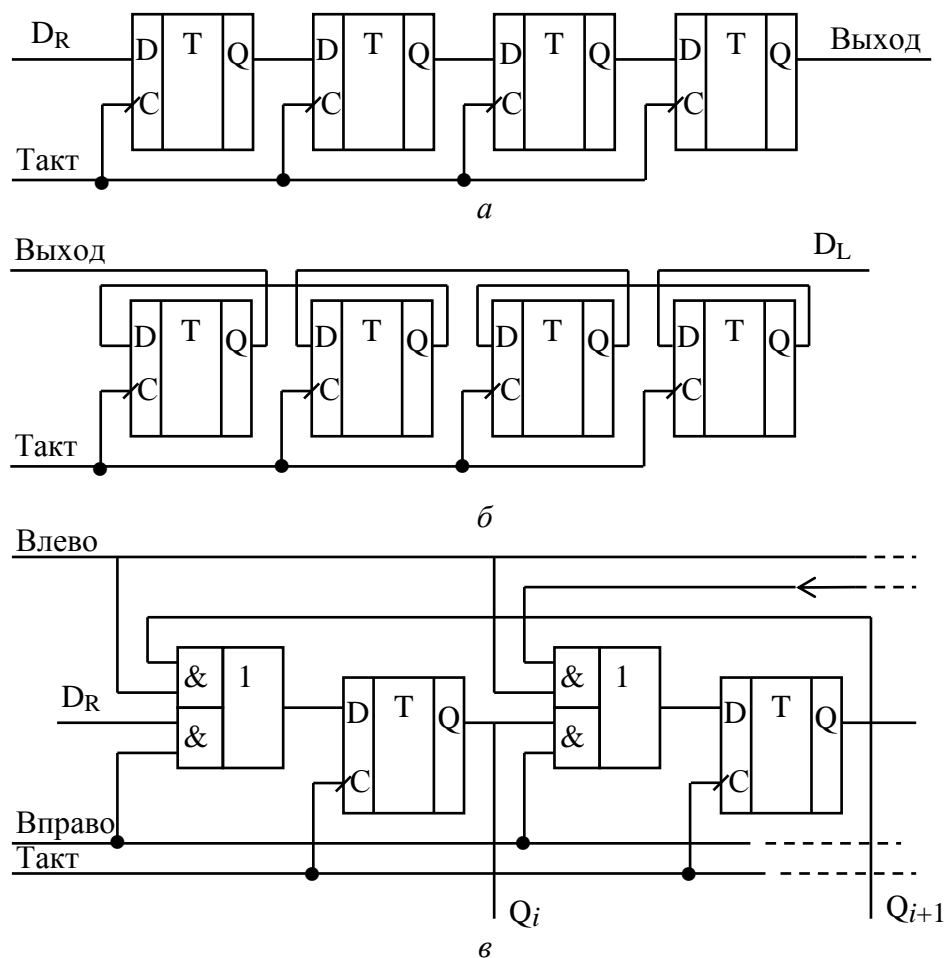


Рис. 5.1. Варианты реализации функции сдвига: *a* – вправо, *б* – влево, *в* – реверсивный

разрядной сетке не на один разряд, как это требуется, а на большее неконтролируемое число разрядов.

Появление в межразрядных связях логических элементов упрощает выполнение условий работоспособности регистров, при этом иногда становится возможным и применение простейших триггеров. Триггеры с динамическим управлением или двухступенчатые обеспечивают работоспособность регистра.

Задания на лабораторную работу

Лабораторная работа состоит из двух частей.

Часть 1. Предполагает разработку 4-разрядного параллельного регистра с использованием триггеров заданного типа. Дополнительно указывается фронт тактового сигнала, по которому происходит срабатывание (\uparrow – подъем тактового сигнала, \downarrow – спад тактового сигнала).

Варианты заданий приведены в табл. 5.1.

Таблица 5.1

Номер задания	Тип триггера	Такт	Асинхронный сброс/установка	Разрешение выдачи	Номер задания	Тип триггера	Такт	Асинхронный сброс/установка	Разрешение выдачи
1	D	↑	Сброс	L	7	D	↑	Установка	H
2	RS	↓	Сброс	L	8	RS	↓	Установка	H
3	JK	↑	Установка	L	9	JK	↑	Сброс	H
4	D	↓	Установка	L	10	D	↓	Сброс	H
5	RS	↑	Сброс	L	11	RS	↑	Установка	H
6	JK	↓	Сброс	L	12	JK	↓	Установка	H

В качестве базовых элементов использовать соответствующие примитивы библиотеки САПР Quartus II.

Часть 2. Предлагается разработать многофункциональный регистр на базе D-триггеров. Условное графическое обозначение (УГО) многофункционального регистра приведено на рис. 5.2.

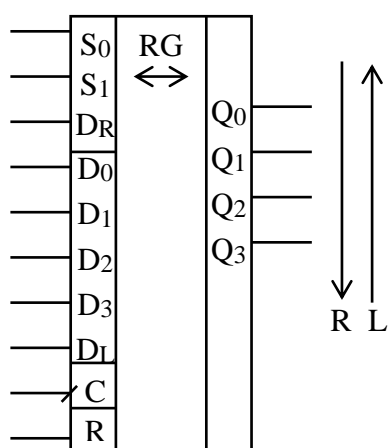


Рис. 5.2. УГО многофункционального регистра

В зависимости от реализуемых операций в регистре должны быть предусмотрены следующие входы:

$D_0 - D_3$ – разряды входного параллельного кода;
 $Q_0 - Q_3$ – разряды выходного параллельного кода;
 C – синхросигнал приема параллельного кода, обеспечивающий срабатывание по переднему фронту;
 R – асинхронный вход сброса регистра;
 S_0, S_1 – входы задания микроопераций;
 DR, DL – последовательные входы при сдвиге вправо и влево соответственно.

Для реализации асинхронного сброса триггера – сброса, не связанного с тактовым сигналом, следует использовать соответствующие входы CLRN примитива D-триггера DFF. Синхронный сброс, который переводит триггер в состояние «сброс» только на активном фронте тактового сигнала, может быть реализован как часть комбинационной логики, генерирующей входные данные на входе D-триггера.

Варианты заданий приведены в табл. 5.2, а выполняемые микрооперации – в табл. 5.3.

Таблица 5.2

Номер задания	Выполняемые микрооперации
1	1, 2, 5, 7
2	1, 3, 6, 8
3	1, 4, 5, 9
4	1, 2, 6, 10
5	1, 3, 5, 11
6	1, 4, 6, 12
7	1, 2, 5, 11
8	1, 3, 6, 10
9	1, 4, 5, 9
10	1, 2, 6, 8
11	1, 3, 5, 7
12	1, 4, 6, 9

Таблица 5.3

Номер МО	Описание микрооперации
1	Параллельная загрузка
2	Синхронный сброс
3	Инвертирование кода
4	Установка в 1 всех разрядов
5	Сдвиг вправо циклический
6	Сдвиг влево циклический
7	Сдвиг влево, заполнение – D_L
8	Сдвиг вправо, заполнение – D_R
9	Сдвиг влево, заполнение – 1
10	Сдвиг вправо, заполнение – 0
11	Сдвиг влево, заполнение – 0
12	Сдвиг вправо, заполнение – 1

В каждом варианте предполагается реализация четырех микроопераций. Кодирование выполняется самостоятельно, код формируется на входах S_0 , S_1 многофункционального регистра.

Последовательность выполнения работы

Часть 1:

1. Создать проект, с использованием средств графического редактора подготовить схему регистра в соответствии с заданием.
2. Компилировать проект, исправить ошибки, если они есть.
3. Выполнить функциональное моделирование проекта, обеспечив полный перебор возможных значений входных сигналов, убедиться в правильности работы схемы.
4. Выполнить временное моделирование, проанализировать полученную временную диаграмму, оценить максимальную частоту работы регистра на основании временных параметров комбинационной логики.

Часть 2:

1. Разработать схему многофункционального регистра в соответствии с заданием.
2. Создать проект с использованием средств графического редактора, ввести в него подготовленную схему многофункционального регистра. Компилировать проект, исправить ошибки, если они есть.

3. Выполнить функциональное моделирование проекта, обеспечив полный перебор возможных значений входных управляющих сигналов, убедиться в правильности работы схемы.

4. Скорректировать схему проекта, обеспечив понижение частоты входного тактового сигнала с помощью специально добавленного счетчика таким образом, чтобы можно было комфортно наблюдать работу регистра при выполнении сдвигов. Проверить работу схемы на макетной плате. Данные, загружаемые в регистр, и код микрооперации формировать на движковых переключателях. Результаты работы проекта наблюдать на светодиодной линейке.

Содержание отчета

Отчет по первой части работы должен содержать:

1. Спроектированную функциональную схему регистра, оформленную с учетом требований ГОСТ.

2. Схемный файл проекта с обоснованием выбора используемых примитивов.

3. Временные диаграммы, полученные в ходе функционального и временного моделирования, результаты их анализа.

Отчет по второй части работы должен содержать:

1. Описание процесса проектирования схемы многофункционального регистра, его функциональную схему, оформленную с учетом требований ГОСТ, и условное графическое обозначение с учетом реализуемых.

2. Схемный файл проекта.

3. Временные диаграммы, полученные в ходе функционального моделирования проекта.

4. Описание процесса макетирования.

Лабораторная работа 6

ПРОЕКТИРОВАНИЕ РАСПРЕДЕЛИТЕЛЕЙ ТАКТОВЫХ СИГНАЛОВ

Цель работы – исследовать особенности функционирования распределителей тактовых сигналов, построенных на сдвигающих регистрах и счетчиках Джонсона.

Краткие теоретические сведения

Распределители тактовых сигналов (РТС) или счетчики в коде «1 из N», формируют импульсные последовательности с заданными временными диаграммами. Для получения таких последовательностей период формируемой временной диаграммы разбивается на части («кванты»), соответствующие минимальному временному интервалу, с использованием задающего генератора с частотой, равной m/T , где m – число «квантов» в периоде диаграммы T . Далее выходные импульсы задающего генератора распределяются во времени и пространстве так, что каждый «квант» появляется в свое время и в своем пространственном канале.

РТС на n выходных каналов легко строится на сдвигающем n -разрядном регистре, замкнутом в кольцо. Для работы распределителя необходимо предварительно занести в регистр слово, содержащее одну единицу. При сдвигах единица перемещается с одного выхода на другой, циркулируя в кольце. Недостаток схемы – возможные нарушения функционирования при сбое. Если в силу каких-либо причин слово в регистре исказится, то возникшая ошибка станет постоянной. Схема не обладает свойством самовосстановления.

Возможны варианты РТС с самовосстановлением работы на кольцевом регистре. Схема такого распределителя с самовосстановлением за несколько тактов основана на том, что на вход регистра подаются нули, пока в нем имеется хотя бы одна единица. Таким образом, лишние возникшие единицы будут устранены. Когда регистр очистится, сформируется сигнал записи единицы на его входе. Таким образом, потеря единственной единицы будет исключена, а выход логического элемента, выполняющего самовосстановление схемы, образует еще один дополнительный канал.

Распределители на кольцевых регистрах находят применение при малом числе выходных каналов. Достоинством распределителей на кольцевых регистрах является отсутствие в их структуре дешифраторов и, как следствие,

высокое быстродействие (задержка перехода в новое состояние равна времени переключения триггера).

Альтернативный подход к построению РТС основан на использовании перекрестной обратной связи (счетчик Джонсона, счетчик Мебиуса, счетчик Либбау–Крейга). Счетчики Джонсона осуществляют счет в коде Джонсона. Состояния счетчика представлены в табл. 6.1.

Таблица 6.1

Номер состояния	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	1	1	1	1
5	1	1	1	0
6	1	1	0	0
7	1	0	0	0

В счетчике Джонсона выход представлен не в коде «1 из N», что требует преобразования кодов для получения выходов РТС, однако такие преобразователи просты, что обуславливает применение счетчиков Джонсона в составе распределителей тактовых сигналов.

Схемотехнически счетчик Джонсона – это сдвигающий регистр с перекрестной обратной связью. Количество внутренних состояний $2n$. Вариант реализации счетчика приведен на рис. 6.1. Для получения нечетного числа состояний в схему дополнительно вводят связь, показанную на рисунке штриховой линией. При возникновении в счетчике комбинации 0111 на выходе элемента ИЛИ-НЕ

появится логический ноль и счетчик перейдет в состояние 1110, т. е. исключается состояние «все единицы». Этот вариант схемы не обладает свойством восстановления после попадания в запрещенные состояния и для правильной работы требует начального сброса.

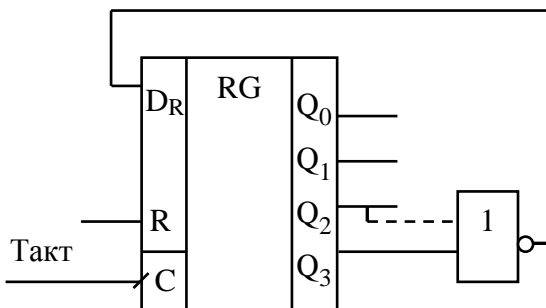


Рис. 6.1. Схема счетчика Джонсона без восстановления после сбоя

появится логический ноль и счетчик перейдет в состояние 1110, т. е. исключается состояние «все единицы». Этот вариант схемы не обладает свойством восстановления после попадания в запрещенные состояния и для правильной работы требует начального сброса.

Указанного недостатка лишена схема счетчика Джонсона, в которой сигнал обратной связи вырабатывается согласно выражению

$$D_R = \overline{Q_{n-1}} \vee \overline{Q_{n-2} Q_{n-3} \dots Q_1 Q_0}. \quad (6.1)$$

Первое слагаемое отражает перекрестную обратную связь, а второе – обеспечивает самовосстановление.

Вариант счетчика Джонсона, схема которого приведена на рис. 6.2, также обладает свойством самовосстановления.

Четырехразрядный регистр реализован трехразрядным, имеющим вход начальной установки, и дополнительном D-триггере. Инверсный выход триггера поступает на вход D_R , образуя перекрестную обратную связь.

Эффект самокоррекции заключается в том, что с помощью элемента «&» выделяется комбинация 10 в старших разрядах и при ее появлении принудительно устанавливаются в 0 все разряды, кроме старшего. Таким образом, счетчик оказывается в разрешенной комбинации и далее функционирует правильно. Очевидно, что дополнительный триггер может быть установлен и со стороны младших разрядов, выявляя комбинацию 01 с обнулением всех разрядов, кроме младшего.

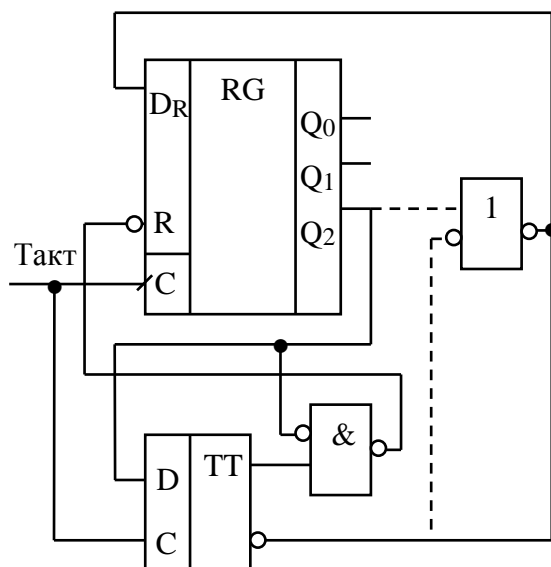


Рис. 6.2. Схема счетчика Джонсона с восстановлением после сбоя

Элемент «1» обеспечивает реализацию счетчика Джонсона с нечетным количеством состояний ($2n - 1$).

Преобразование выходного кода счетчика Джонсона в код «1 из N» осуществляется дешифратором Джонсона, схема которого приведена на рис. 6.3.

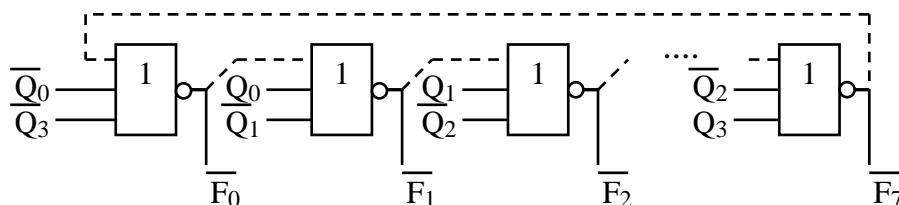


Рис. 6.3. Дешифратор кода Джонсона

Принцип дешифрации состоит в выявлении положения характерной координаты временной диаграммы – границы между зонами единиц и нулей. Эти границы соответствуют закрашенным ячейкам табл. 6.1.

Задания на лабораторную работу

Часть 1. Синтезировать на основе имеющихся в библиотеке САПР Quartus II примитивов распределитель тактовых сигналов на основе сдвигающего регистра в соответствии с заданием, приведенным во второй строке табл. 6.2. Расшифровка вариантов: первая цифра – количество выходных каналов, вторая: 1 – циркуляция единиц, 0 – циркуляция нуля.

Таблица 6.2

Номер варианта	1	2	3	4	5	6	7	8	9	10	11	12
Шифр задания, часть 1	6, 0	7, 1	7, 0	8, 1	8, 0	6, 1	9, 0	9, 0	5, 1	5, 0	10, 0	10, 1
Шифр задания, часть 2	6 0 1	7 1 2	7 0 3	8 1 3	8 0 2	6 1 1	9 0 1	9 0 2	5 1 3	5 0 3	10 0 2	10 1 1

Часть 2. Используя примитивы САПР Quartus II, синтезировать РТС на основе счетчика Джонсона в соответствии с заданием, приведенным в третьей строке табл. 6.2.

Расшифровка вариантов: первая цифра: количество выходных каналов; вторая цифра: 1 – циркуляция единицы, 0 – циркуляция нуля; третья цифра: 1 – счетчик без восстановления, 2 – с самовосстановлением с перекрестной обратной связью по выражению (6.1), 3 – с самовосстановлением, реализованным с дополнительным триггером.

Последовательность выполнения работы

1. Разработать функциональную схему РТС на основе сдвигающего регистра с самовосстановлением после сбоя в соответствии с заданием.
2. Подготовить проект в САПР Quartus II, реализовать схему РТС.
3. Выполнить функциональное моделирование схемы, убедиться в правильности ее работы.
4. Повторить пп. 1–3 для РТС на основе счетчика Джонсона.
5. Добавить в проект счетчик понижающий частоту тактового сигнала. Выполнить компиляцию. Назначить контакты ПЛИС.
6. Загрузить проект на плату, наблюдать работу устройства в автоколебательном режиме.

Содержание отчета

1. Функциональные схемы РТС на основе сдвигающего регистра и счетчика Джонсона, оформленные с учетом требований ГОСТ.
2. Схемные файлы САПР Quartus II для обоих вариантов реализации РТС.
3. Временные диаграммы, полученные в ходе функционального моделирования спроектированных устройств.
4. Описание процесса макетирования РТС на основе счетчика Джонсона.
5. Выводы по работе.

Лабораторная работа 7

ПРОЕКТИРОВАНИЕ ДВОИЧНЫХ СЧЕТЧИКОВ

Цель работы – исследовать особенности функционирования двоичных счетчиков с вырожденными переходами (принудительной установкой в состояния) и различными способами организации переноса.

Краткие теоретические сведения

Счетчиками называются узлы последовательностного типа, осуществляющие счет поступающих на вход импульсов, фиксацию и сохранение результата в заданном коде. Счетчики классифицируются по ряду признаков: направлению счета (суммирующие, вычитающие, реверсивные), способу организации цепей переноса (последовательный, параллельный, сквозной и групповой), используемой системе счисления (двоичная позиционная, двоично-десятичная, прочие не двоичная), модулю счета M – числу состояний счетчика.

В позиционной системе счисления n -разрядный счетчик имеет модуль счета $M = 2^n$. Переходы, реализуемые в таком счетчике, называются невырожденными, а порядок счета – естественным.

Для организации произвольного модуля счета сокращают число внутренних состояний счетчика, запрещая попадание счетчика в ряд из них, т. е. с помощью определенных схемных приемов организуют принудительную установку счетчика – так называемые вырожденные переходы.

Остановимся на двух способах построения счетчиков с произвольным модулем: управлении сбросом (установкой) и модификации межразрядных связей.

При управлении сбросом выявляется момент достижения содержимым счетчика значения $M - 1$. Эта ситуация обеспечивает формирование сигнала сброса счетчика в следующем такте, после чего начинается новый цикл. Такой вариант обеспечивает легкость перестройки счетчика на другие значения модуля, так как требуется изменять лишь код, с которым сравнивается содержимое счетчика для выявления момента сброса.

При построении счетчика с модифицированными межразрядными связями последние, лишние состояния исключаются непосредственно из таблицы функционирования счетчика. Далее обычным для синтеза автоматов способом строится счетчик, специфика которого состоит в нестандартных функ-

циях возбуждения триггеров и, следовательно, в нестандартных связях между триггерами, что и объясняет название способа. Схема получается специализированная, изменение модуля счета требует изменения самой схемы. В то же время реализация схемы счетчика может оказаться простой.

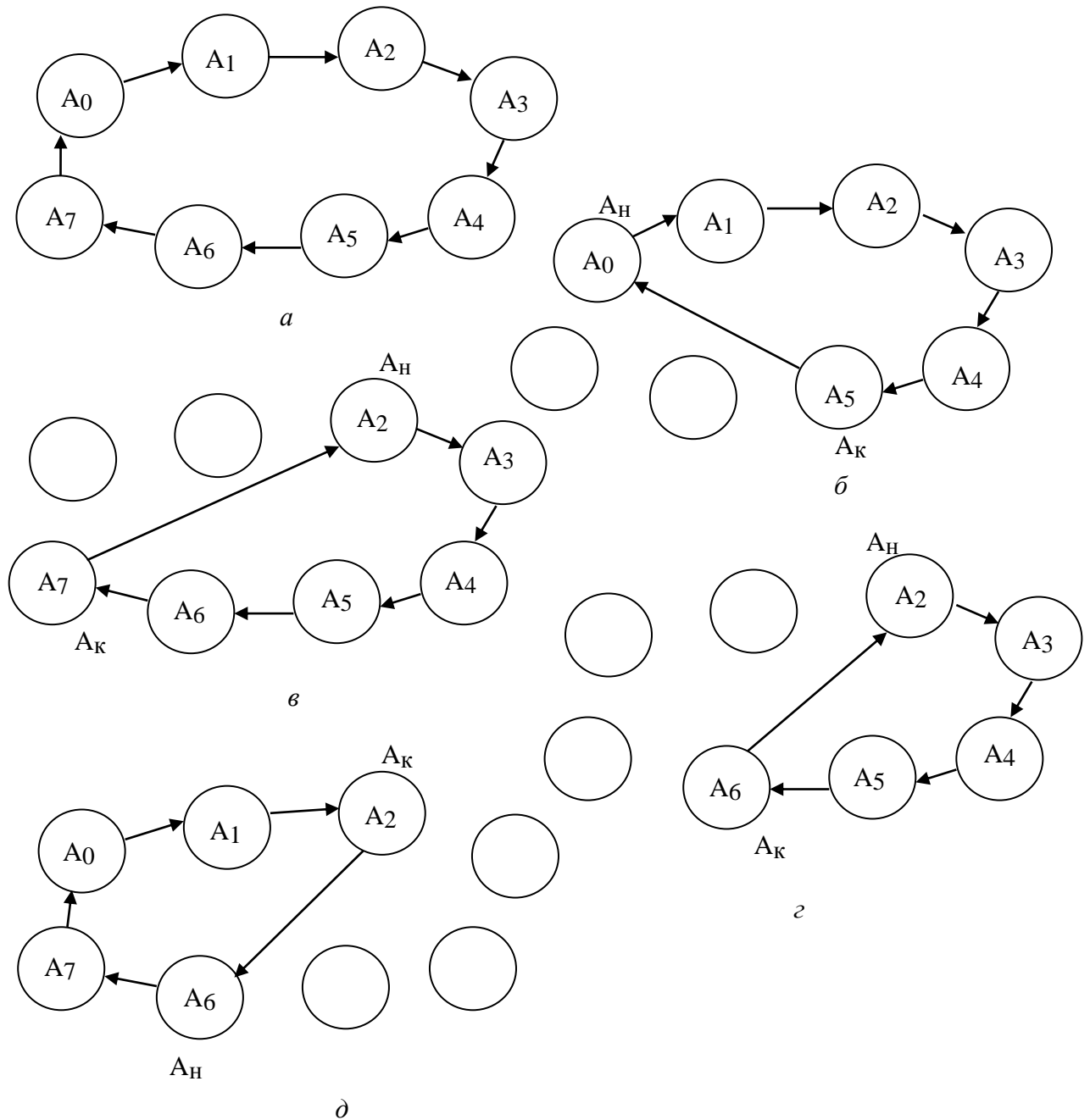


Рис. 7.1. Графы двоичных счетчиков

На рис. 7.1, *a* представлен граф функционирования трехразрядного двоичного счетчика без вырожденных переходов, т. е. с естественным порядком счета; на рис. 7.1, *б* – случай, когда счетчик после конечного состояния A_K переходит в нулевое состояние A_0 .

Рис. 7.1, в соответствует случаю переключения счетчика из конечного состояния A_{2^n-1} (переполнения) в ненулевое начальное A_H . Счетчик, имеющий граф функционирования рис. 7.1, з, считает от кода A_H до состояния A_K , после чего вновь оказывается в состоянии A_H . И, наконец, в счетчике с так называемым принудительным насчетом (рис. 7.1, д) после достижения кода A_K происходит переключение в состояние A_H , а далее – естественный счет вплоть до переполнения.

Один из возможных вариантов схемы счетчика, реализующего граф функционирования рис. 7.1, б, представлен на рис. 7.2, а. До состояния A_K счетчик считает обычным образом. После переключения в состояние A_{K+1} на выходе схемы выделения появляется нулевой сигнал и счетчик сбрасывается в нуль. Состояние A_{K+1} существует кратковременно, только в течение переходного процесса в схеме. Схема выделения может быть реализована с помощью стандартного дешифратора либо на дополнительных логических элементах.

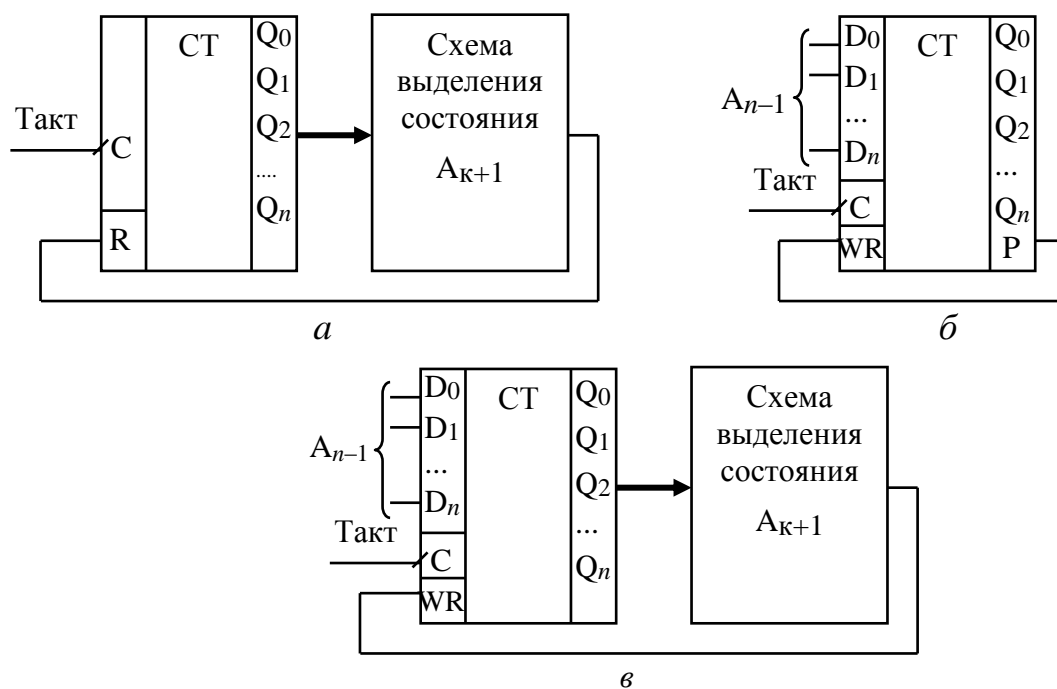


Рис. 7.2. Варианты реализации счетчика

Граф функционирования рис. 7.1, в реализуется схемой счетчика рис. 7.2, б. При переполнении, т. е. при наличии единиц на всех выходах счетчика, во втором полутакте на выходе переноса P появляется нулевой сигнал. Следовательно, во втором полутакте счетчик установится в состояние A_{n-1} , код которого задан на входах счетчика, и по следующему тактовому

импульсу переключится в состояние A_H . Недостатком данного варианта построения счетчика является то, что состояние $(2^n - 1)$ существует лишь в течение полутакта.

Схема счетчика, реализующего граф функционирования, представленный на рис. 7.1, *з*, изображена на рис. 7.2, *в*. После достижения состояния A_{K+1} на входе разрешения приема параллельного кода появляется логический нуль и в счетчик принимается код A_H . Состояние A_{K+1} существует в счетчике кратковременно, лишь на время переходного процесса. Для реализации счетчика с принудительным насчетом (рис. 7.1, *д*) также используется схема рис. 7.2, *в*, однако схема выделения дешифрирует состояние A_{H+1} , а на входы приема параллельного кода подается код A_K .

Построение счетчика с модифицированными межразрядными связями рассмотрим на примере счетчика по модулю $M = 5$. Пусть требуется спроектировать счетчик с выполняющий счет от 0 до 4. Таблица переходов (табл. 7.1) счетчика имеет следующий вид.

При нахождении функций возбуждения триггеров использован «словарь» JK-триггера, приведенный в табл. 4.1.

Таблица 7.1

Исходное состояние Q ₂ Q ₁ Q ₀	Следующее состояние Q ₂ Q ₁ Q ₀	Функции возбуждения триггеров					
		J ₂	K ₂	J ₁	K ₁	J ₀	K ₀
0 0 0	0 0 1	0	X	0	X	1	X
0 0 1	0 1 0	0	X	1	X	X	1
0 1 0	0 1 1	0	X	X	0	1	X
0 1 1	1 0 0	1	X	X	1	X	1
1 0 0	0 0 0	X	1	0	X	0	X

Вместо символа произвольного сигнала X можно подставлять значения 0 или 1, критерием выбора является упрощение получаемой функции. На основании таблицы получим следующие функции:

$$K_0 = 1; K_1 = J_1; K_2 = 1;$$

$$J_0 = \overline{Q_2}; J_1 = Q_0; J_2 = Q_1 Q_0.$$

На основании функций может быть построена схема счетчика рис. 7.3.

В рассмотренном примере лишние состояния при нормальном функционировании счетчика не используются. Однако при сбоях или подаче на схему напряжения питания в начале ее работы лишние состояния могут возникать.

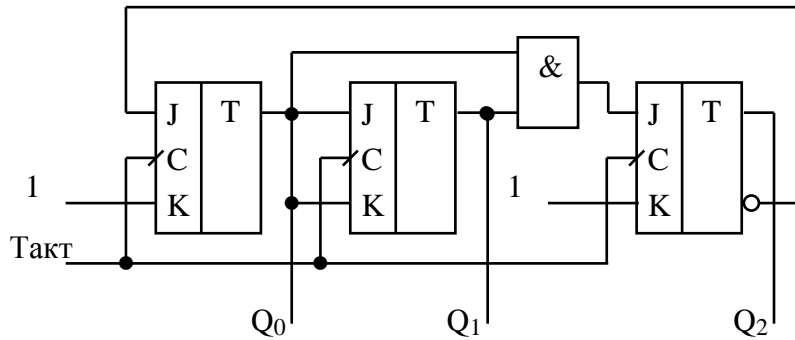


Рис. 7.3. Схема счетчика по модулю 5
с модифицированными межразрядными связями

Поэтому полезно определить поведение схемы при возникновении лишних состояний, т. е. определить переходы, которые происходят из него.

Для полученной схемы счетчика с модулем 5 имеем следующее: состояния 101, 110 и 111 – лишние.

В состоянии 101 $Q_2 = 1$, $Q_1 = 0$ и $Q_0 = 1$ функции возбуждения примут значения $J_0 = 0$, $K_0 = 1$, $J_1 = K_1 = 1$, $J_2 = 0$, $K_2 = 1$. Следовательно, триггеры 0 и 2 сбросятся, а триггер 1 переключится в состояние, противоположное текущему, и из лишнего состояния 101 счетчик перейдет в состояние 010. Аналогично анализируются переходы из состояний 110 и 111.

В итоге будет восстановлен полный граф переходов, представленный на рис. 7.4, в котором учтен и рабочий цикл, и переходы из неиспользуемых состояний (заштрихованные состояния).

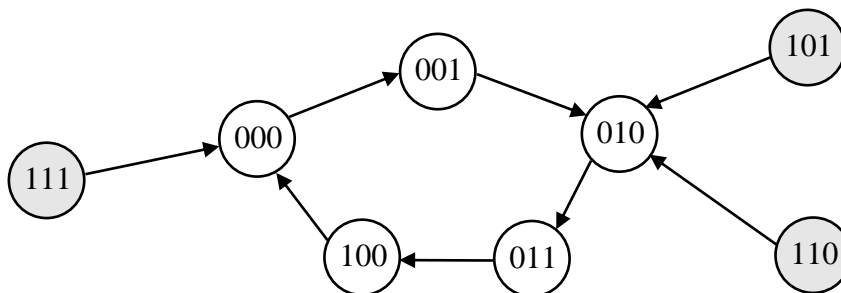


Рис. 7.4. Граф переходов счетчика

Анализ графа показывает, что рассматриваемый счетчик обладает свойством самовосстановления после сбоя – независимо от состояния он вернется в рабочий цикл.

Это свойство можно было обеспечить при разработке схем, добавив в таблицу строки, соответствующие переходам из лишних состояний в любое (любые), соответствующее рабочему циклу.

Задания на лабораторную работу

Часть 1. Синтезировать на основе имеющейся в библиотеке САПР Quartus II мегафункции счетчика LPM_COUNTER (находящейся в разделе *megafunction / arithmetic*) счетчик с вырожденными состояниями. При необходимости в процессе настройки мегафункции установить в счетчике входы **Clear** и / или **Load**. Варианты заданий приведены в табл. 7.2. В столбце «способ реализации 1» приведены задания, реализуемые с помощью выделения конечного состояния с помощью дешифратора, в столбце «способ реализации 2» – с помощью дополнительной логики. В ячейках таблицы указаны разрешенные состояния счетчика.

Таблица 7.2

Номер бригады	Способ реализации 1. Порядок счета	Способ реализации 2. Порядок счета
1	0–12	0–2; 5–13
2	2–14	0–6; 9–14
3	0–13	0–5; 7–15
4	3–12	0–2; 4–14
5	7–14	2–4; 8–15
6	0–14	0–2; 5–15
7	5–11	0–6; 11–15
8	6–12	2–7; 10–15
9	4–13	0–4; 9–15
10	2–10	1–5; 10–15

Таблица 7.3

Номер бригады	Порядок счета	Тип триггера
1	0–8; 10–12	JK
2	0–11	D
3	0–7; 14–15	RS
4	0–2; 5–15	JK
5	3–10	D
6	0–3; 7–15	RS
7	0–4; 6–13	JK
8	5–13	D
9	0–8; 12–15	RS
10	0–5; 7–15	JK

Часть 2. Синтезировать на основе имеющихся в библиотеке САПР Quartus II примитивов триггеров счетчик с вырожденными состояниями с модифицированными межразрядными связями. Варианты заданий приведены в табл. 7.3.

Последовательность выполнения работы

Для каждого из проектируемых устройств выполнить последовательность действий, приведенную далее.

1. Создать проект, с использованием средств графического редактора подготовить схему в соответствии с заданием. Компилировать проект, исправить ошибки, если они есть.

2. Выполнить функциональное моделирование проекта, убедиться в правильности работы схемы.

3. Открыть проект устройства, добавить в проект счетчик, понижающий частоту тактового сигнала, и для заданий пп. 2 и 3 устройство формирования

кодов отображения двоичных кодов в десятичной форме из лаб. раб. 3, при необходимости доработав его.

4. Выполнить компиляцию. Назначить контакты ПЛИС с учетом схемы макетной платы. Загрузить проект в учебную плату.

5. Наблюдать работу устройства, контролируя смену состояний счетчика по числам, отображаемым на индикаторах.

Содержание отчета

Для каждого синтезированного счетчика в отчете должны быть представлены:

1. Описание процесса синтеза.
2. Функциональная схема, оформленная с учетом требований ГОСТ.
3. Схемные файлы САПР Quartus II.
4. Временные диаграммы работы, краткое списание функционирования.
5. Описание процесса и результата макетирования.

Лабораторная работа 8

ПРОЕКТИРОВАНИЕ КОНЕЧНЫХ АВТОМАТОВ

Цель работы – получение навыков в проектировании автомата на основе логических элементов с использованием триггеров заданного типа.

Краткие теоретические сведения

К классу автоматов с памятью (АП) относятся устройства, которые содержат элементы памяти, обеспечивающие свойство АП сохранять некоторое состояние S , определяемое совокупностью состояний всех элементов памяти. АП реагирует входные сигналы сменой состояния и выработкой выходных.

Автоматы с памятью в каноническом представлении разделяют на две части: память и комбинационную цепь. На входы КЦ подаются входные сигналы и сигналы состояния АП, на ее выходе вырабатываются выходные сигналы и сигналы перевода АП в новое состояние.

Далее будут рассматриваться исключительно синхронные автоматы, способные изменять свое состояние только в моменты изменения входных системных сигналов: синхросигнала и сигнала асинхронной (по отношению к синхросигналу) начальной установки (сброса). Сигналы, подаваемые на информационные входы, регистрируются по рабочему фронту синхросигнала, определяющему границы длительности тактовых интервалов.

Функционирование автомата может быть задано различными формализованными способами: таблицами, формулами, диаграммами состояний, графами переходов, схемой микропрограммы функционирования.

В случае, когда автомат задан схемой микропрограммы функционирования, методика его синтеза включает следующие этапы:

- отметку состояний на схеме микропрограммы и их кодирование;
- составление таблицы переходов автомата;
- определение функций возбуждения;
- логический синтез и реализацию в заданном элементном базисе.

АП может иметь структуру автомата Мили или Мура.

Базовым элементом автомата является регистр состояния. Выходные сигналы (S) триггеров этого регистра определяют как следующее состояние автомата (устанавливаемое по фронту тактового сигнала Clk), так и значения выходных сигналов.

Модель Мура описывает вариант автомата, в котором выходные сигналы Y определяются только текущим состоянием S . Модель Мили соответствует варианту автомата, у которого выходные сигналы определяются текущим состоянием и комбинацией входных сигналов X .

Структурное отличие между ними сводится к отсутствию у автоматов Мура подключения входных сигналов X к блоку комбинационной логики, определяющей значения выходных сигналов Y , как показано на рис. 8.1.

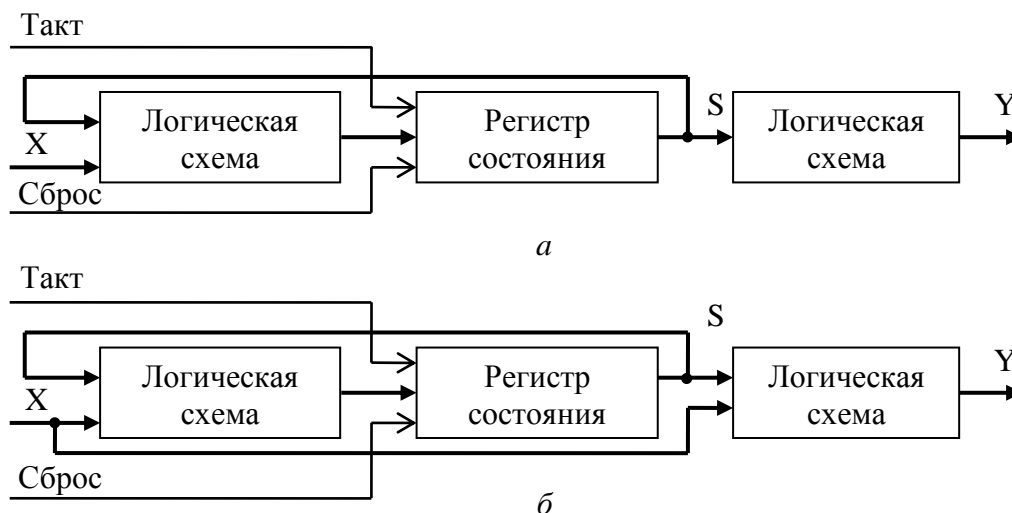


Рис. 8.1. Структурные схемы автоматов Мура и Мили

Рассмотрим методику синтеза на примере синтеза автоматов Мили и Мура, заданных схемой микропрограммы функционирования рис. 8.2.

При синтезе автомата Мура состояниям сопоставляются все операторные, начальная и конечная вершины, причем две последние кодируются одинаково.

Рекомендуется кодировать начальное и конечное состояния нулями, а остальные состояния, по возможности, – кодами микрокоманд. На схеме рис. 8.2 a_i – отметки состояний автомата Мура, причем применено кодирование

$$a_0 = 00; a_1 = 01; a_2 = 11; a_3 = 10.$$

Для автомата Мили состояниям сопоставляются вход конечной вершины и входы всех вершин, следующих непосредственно за операторными и за начальной.

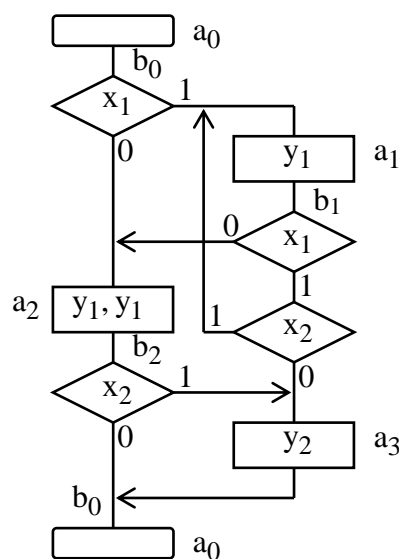


Рис. 8.2. Схема микропрограммы автомата

На схеме рис. 8.2 состояния автомата Мили отмечены b_k , причем $b_0 = 00$; $b_1 = 10$; $b_2 = 11$.

Структурная таблица автомата Мура (табл. 8.1) содержит в каждой строке отметку, код исходного состояния, условие перехода, отметку и код нового состояния. Кроме того, резервируются столбцы для записи функций возбуждения на последующих этапах синтеза. Выходы автомата Мура совпадают с состоянием перехода и записываются в тех же графах, что и эти состояния.

Таблица 8.1

Исходное состояние $Q_i [n]$			Условие	Состояние перехода $Q_i [n+1]$				Функции возбуждения	
Метка	Q_1	Q_0		Метка	Q_1	Q_0	Вых. сигнал	D_1	D_0
a_0	0	0	x_1	a_1	0	1	y_1	0	1
			$\sim x_1$	a_2	1	1	y_1, y_2	1	1
a_1	0	1	$x_1 x_2$	a_1	0	1	y_1	0	1
			$\sim x_1$	a_2	1	1	y_1, y_2	1	1
			$x_1 \sim x_2$	a_3	1	0	y_2	1	0
a_2	1	1	x_2	a_3	1	0	y_2	1	0
			$\sim x_2$	a_3	0	0	–	0	0
a_3	1	0	1	a_3	0	0	–	0	0

Структурная таблица автомата Мили (табл. 8.2) отличается столбцом, в котором записываются выходные сигналы, являющиеся функциями состояния и входного сигнала.

Таблица 8.2

Исходное состояние $Q_i [n]$			Условие	Состояние перехода $Q_i [n+1]$					Функции возбуждения			
Метка	Q_1	Q_0		Метка	Q_1	Q_0	y_1	y_2	J_1	K_1	J_0	K_0
b_0	0	0	x_1	b_1	1	0	1	0	1	*	0	*
			$\sim x_1$	b_2	1	1	1	1	1	*	1	*
b_1	1	0	$x_1 x_2$	b_1	1	0	1	0	*	0	0	*
			$\sim x_1$	b_2	1	1	1	1	*	0	1	*
			$x_1 \sim x_2$	b_0	0	0	0	1	*	1	0	*
b_2	1	1	x_2	b_0	0	0	0	1	*	1	*	1
			$\sim x_2$	b_0	0	0	0	0	*	1	*	1

В столбцы «функции возбуждения» записываются значения логических сигналов на входах элементов памяти, обеспечивающих требуемое в текущем такте изменение состояния. В рассматриваемом примере для автомата Мура использованы D-триггеры, а для автомата Мили – JK-триггеры.

Далее выполняется запись функций возбуждения, а для автомата Мили также формируются функции выходов.

Для автомата Мура на основании данных табл. 8.1 функции возбуждения для триггеров D_1 и D_0 будут иметь следующий вид:

$$D_0 = \overline{Q_1} \overline{Q_0} x_1 \vee \overline{Q_1} \overline{Q_0} \overline{x_1} \vee \overline{Q_1} Q_0 x_1 x_2 \vee \overline{Q_1} Q_0 \overline{x_1},$$

$$D_1 = \overline{Q_1} \overline{Q_0} \overline{x_1} \vee \overline{Q_1} Q_0 \overline{x_1} \vee \overline{Q_1} Q_0 x_1 \overline{x_2} \vee \overline{Q_1} Q_0 x_2.$$

В зависимости от выбранного способа построения комбинационной цепи, например при реализации в базисе логических примитивов, может потребоваться минимизация полученных функций. После минимизации функции возбуждения примут следующий вид:

$$D_0 = \overline{Q_1} \overline{Q_0} \vee \overline{Q_1} x_2 \vee \overline{Q_1} x_1,$$

$$D_1 = \overline{Q_1} \overline{x_1} \vee Q_1 Q_0 x_2 \vee \overline{Q_1} Q_0 \overline{x_2}.$$

Схема автомата Мура приведена на рис. 8.3, а временная диаграмма его работы – на рис. 8.4.

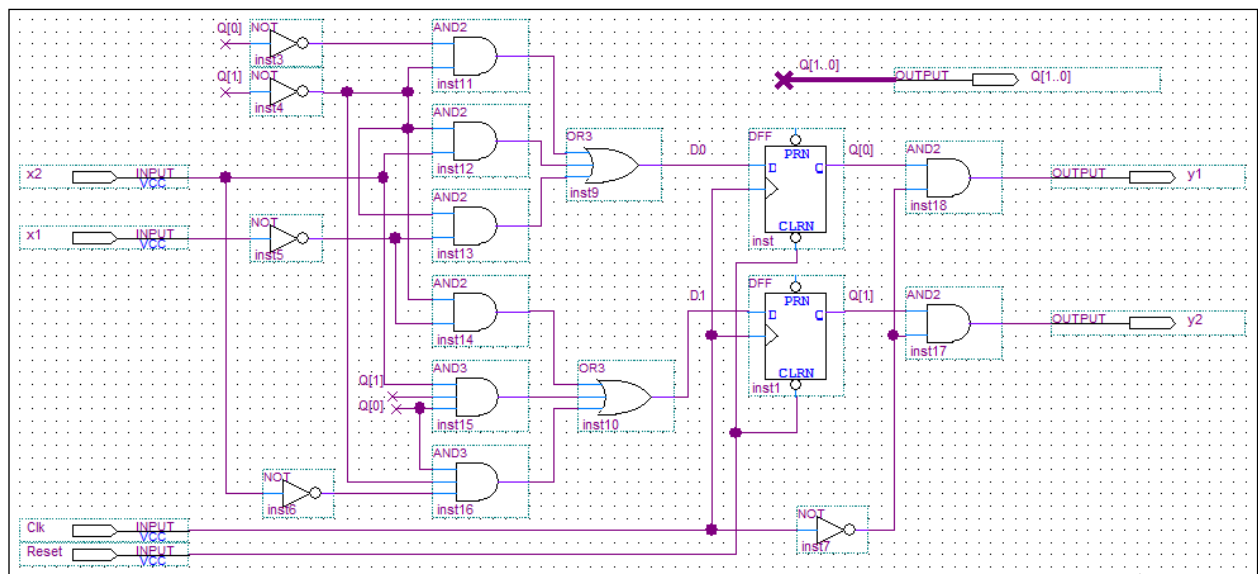


Рис. 8.3. Схема автомата Мура

Функции выхода y_1 и y_2 формируются на основе сигналов триггеров:

$$y_0 = Q_0 \overline{Clk}; \quad y_1 = Q_1 \overline{Clk}.$$

Сигналы микроопераций формируются в фазе нулевого значения тактового сигнала Clk , что исключает возможное влияние процессов переключения триггеров на передний фронт и длительность выходных сигналов.

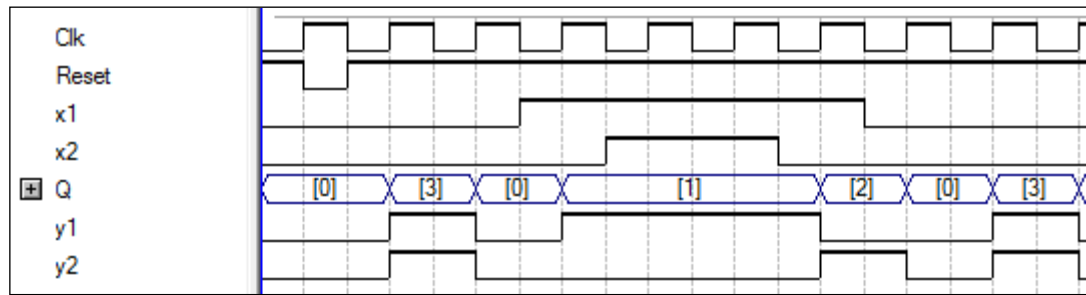


Рис. 8.4. Временная диаграмма автомата Мура

Смена состояний автомата осуществляется при поступлении тактового сигнала Clk, начальная установка выполняется по сигналу Reset, подключенному ко входам асинхронного сброса триггеров.

Для удобства контроля работы спроектированного автомата в схему дополнительно введен выходной порт Q[1..0], что позволяет наблюдать на временной диаграмме изменения состояния автомата.

Для автомата Мили на основании данных табл. 8.2 функции возбуждения для триггеров JK₁ и JK₀ будут иметь следующий вид:

$$J_1 = K_0 = 1,$$

$$K_1 = x_1 x_2 \overline{Q_0} \overline{Q_1} \vee x_2 \overline{Q_0} Q_1 \vee x_2 \overline{Q_0} Q_1 = x_1 x_2 \overline{Q_0} \overline{Q_1} \vee Q_0 Q_1,$$

$$J_1 = x_1 \overline{Q_0} \overline{Q_1} \vee x_1 \overline{Q_0} Q_1 = x_1 \overline{Q_0}.$$

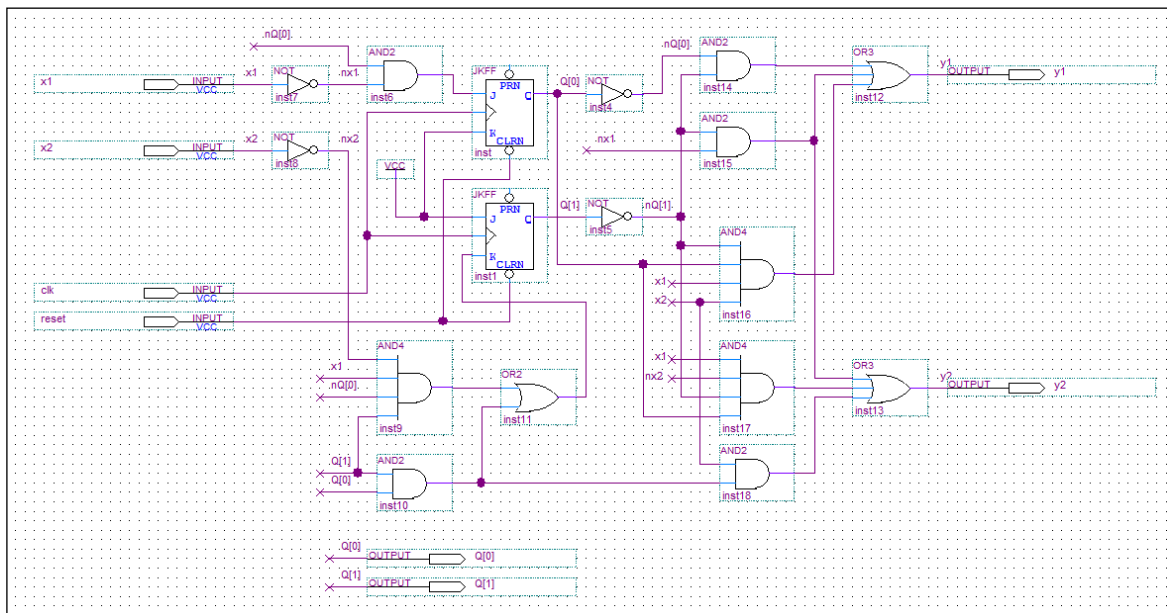


Рис. 8.5. Схема автомата Мили

Функции выходов представлены следующими выражениями:

$$y_1 = x_1 \overline{Q_0} \overline{Q_1} \vee x_1 \overline{Q_0} Q_1 \vee x_1 x_2 \overline{Q_0} \overline{Q_1} \vee x_1 \overline{Q_0} \overline{Q_1},$$

$$y_2 = x_1 \overline{Q_0} \overline{Q_1} \vee x_1 \overline{Q_0} Q_1 \vee x_1 x_2 \overline{Q_0} \overline{Q_1} \vee x_2 \overline{Q_0} Q_1.$$

В результате минимизации функций для выходных сигналов y_1 и y_2 будут получены следующие выражения:

$$y_1 = \overline{Q_0} \overline{Q_1} \vee x_1 x_2 Q_0 \overline{Q_1} \vee \overline{x_1} \overline{Q_1},$$

$$y_2 = \overline{x_1} \overline{Q_1} \vee x_1 \overline{x_2} Q_0 \overline{Q_1} \vee x_2 Q_0 Q_1.$$

Комбинационные логические схемы, формирующие функции переходов и выходов автомата, легко реализуются с использованием логических примитивов. Схема автомата Мили приведена на рис. 8.5.

Автомат меняет состояние при поступлении тактового сигнала Clk, установка в начальное состояние выполняется асинхронно по сигналу Reset.

Задания на лабораторную работу

Спроектировать автомат, реализующий заданный алгоритм функционирования. Варианты заданий приведены в таблице 8.3. В таблице указаны схема микропрограммы из вариантов, приведенных на рис. 8.6, тип автомата и тип используемых триггеров.

Таблица 8.3

Номер	Схема микро- программы	Автомат	Триггер	Номер	Схема микро- программы	Автомат	Триггер
1	а	Мили	RS	8	б	Мили	JK
2	б	Мура	D	9	в	Мура	RS
3	в	Мили	JK	10	г	Мили	D
4	г	Мура	RS	11	а	Мура	JK
5	а	Мили	D	12	б	Мили	JK

Последовательность выполнения работы

1. Для заданной схемы микропрограммы автомата и указанной модели реализации необходимо выполнить разметку схемы микропрограммы, определить кодирование состояний, подготовить структурную таблицу, сформировать функции возбуждения триггеров регистра состояния, а также функции выходов, при необходимости выполнить минимизацию этих функций.

2. Разработать функциональную схему автомата.

3. Разработать проект, реализующий схему автомата в соответствии с заданием.

4. Выполнить функциональное моделирование проекта, убедиться в правильности работы схемы. При моделировании обеспечить полный перебор возможных значений входных векторов и переходов автомата.

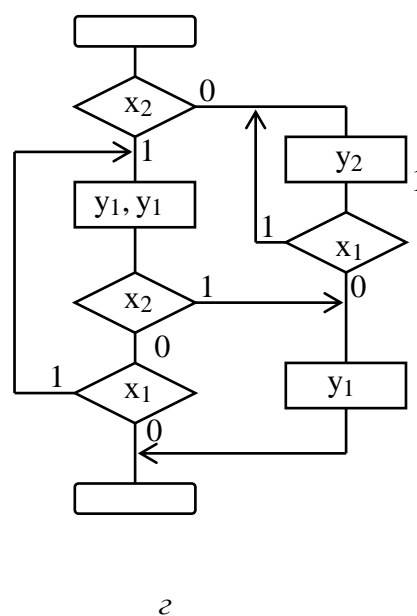
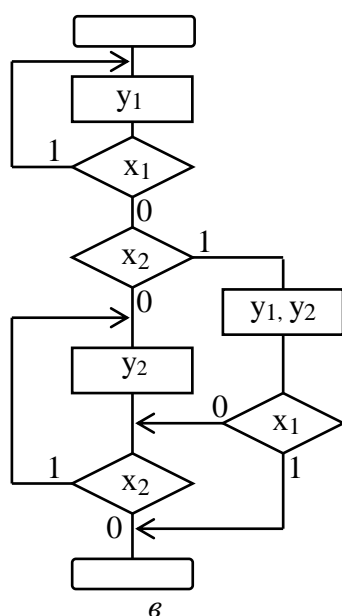
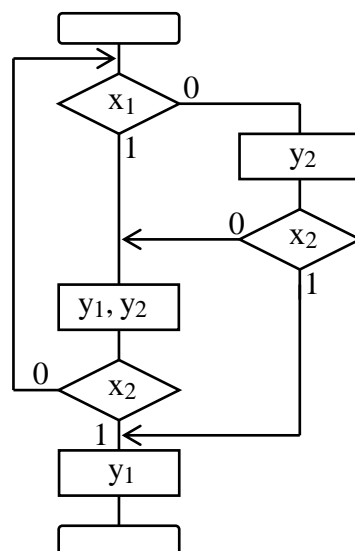
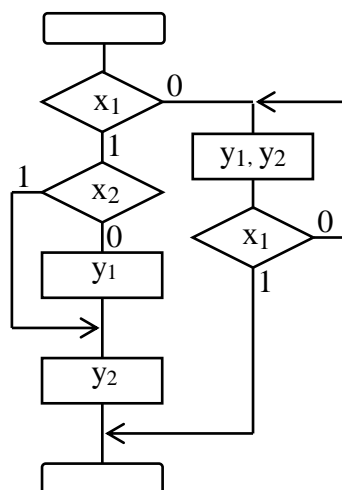


Рис. 8.6. Схемы микропрограмм автоматов

5. Загрузить проект в учебную плату, подключив входные сигналы к переключателям, а выходные – к светодиодам светодиодной линейки. Проверить работу реальной схемы.

Содержание отчета

Отчет должен следующую информацию.

1. Описание процесса синтеза схемы автомата, структурную таблицу автомата, его функциональную схему.
2. Схему спроектированного автомата, представленную соответствующим графическим файлом проекта САПР Quartus II.

3. Временные диаграммы работы автомата, полученные в результате функционального моделирования.
4. Список и назначение внешних контактов, описание процесса макетирования.
5. Выводы по работе.

Список литературы

1. Угрюмов Е. П. Цифровая схемотехника: учеб. пособие для вузов. 3-е изд. перераб. и доп. СПб.: БХВ-Петербург, 2010.
2. Грушвицкий Р. И., Мурсаев А. Х., Угрюмов Е. П. Проектирование систем на микросхемах с программируемой структурой. СПб.: БХВ-Петербург, 2006.
3. Комолов Д. А., Мьяльк Р. А., Зобенко А. А. Системы автоматизированного проектирования фирмы Altera Max Plus II и Quartus II. М.: РадиоСофт, 2002.

1. Порядок проведения макетного эксперимента

Для проведения макетных экспериментов при выполнении цикла лабораторных работ используется учебная плата, основу которой составляет ПЛИС класса FPGA Altera Cyclone II EP2C5Q208C8N.

В состав платы также входят 8-разрядный движковый переключатель, 8-разрядная светодиодная линейка, два 7-сегментных светодиодных индикатора, тактовая кнопка, порты расширения, интерфейс USB для подключения к ПК, интерфейс *Active serial* для конфигурирования платы.

Исходными данными для загрузки являются файлы, созданные САПР Quartus II в процессе компиляции проекта, содержащие конфигурационную информацию.

Для загрузки проекта в учебную плату необходимо выполнить следующие действия.

1. Для подготовки проекта к загрузке конфигурации на кристалл надо выбрать пункт *Device* в меню *Assignments*. Если на этапе создания проекта кристалл ПЛИС не был указан, то в появившемся окне (рис. П1) указать семейство и конкретную микросхему, установленную на плате.

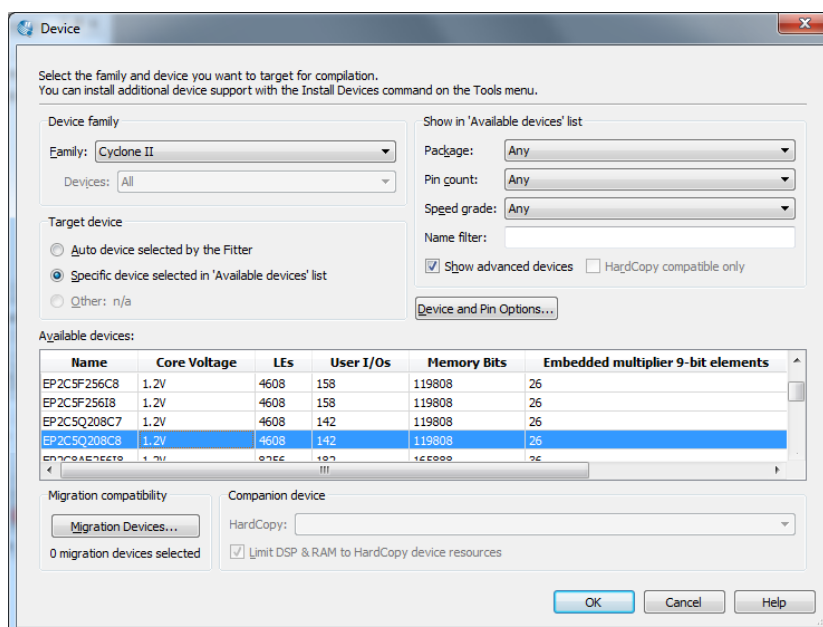


Рис. П1. Выбор микросхемы

Окно выбора режима программирования (рис. П2) запускается кнопкой *Device and Pin Options*. В нем следует выбрать пункт *Configuration* в окне *Category* и установить параметры, приведенные на рис. П2, т. е. *Configuration*

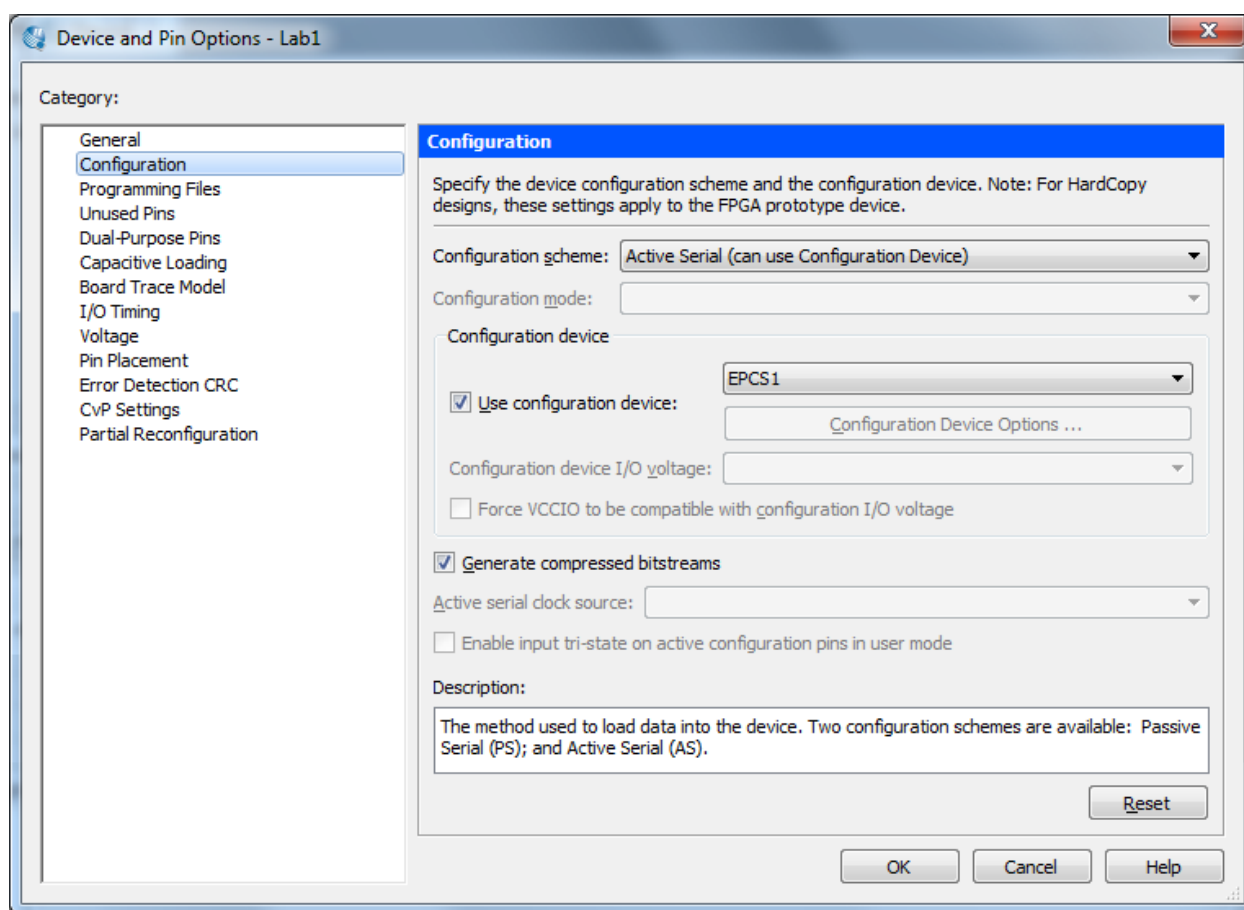


Рис. П2. Параметры устройства программирования

scheme / Active Serial, Use configuration device / EPCS1, включить опцию сжатия *Generate compressed bitstreams*.

В этом окне следует выбрать пункт *Unused Pins* и неиспользуемые контакты перевести в третье состояние *Reserve all unused pins: As input tri-stated*.

2. Загрузка конфигурации в кристалл начинается с «распиновки» микросхемы: для каждого входного и выходного сигнала проекта необходимо назначить реальный контакт ПЛИС. Данное действие можно выполнить с использованием утилиты *Pin Planer*, запускаемой через меню *Assignments*. Номера контактов микросхемы, необходимых для выполнения лабораторных работ, указаны в прил. 3. Вид окна *Pin Planner* представлен на рис. П3. Необходимые для каждого сигнала номера контактов могут быть назначены путем заполнения столбца *Fitter Location* в таблице, расположенной в нижней части окна *Pin Planner*. Дополнительно при назначении контактов необходимо проконтролировать указанный в этой таблице в столбце *I/O Standart* стандарт ввода/вывода. Для используемой учебной платы это – 3,3-V LVTTTL.

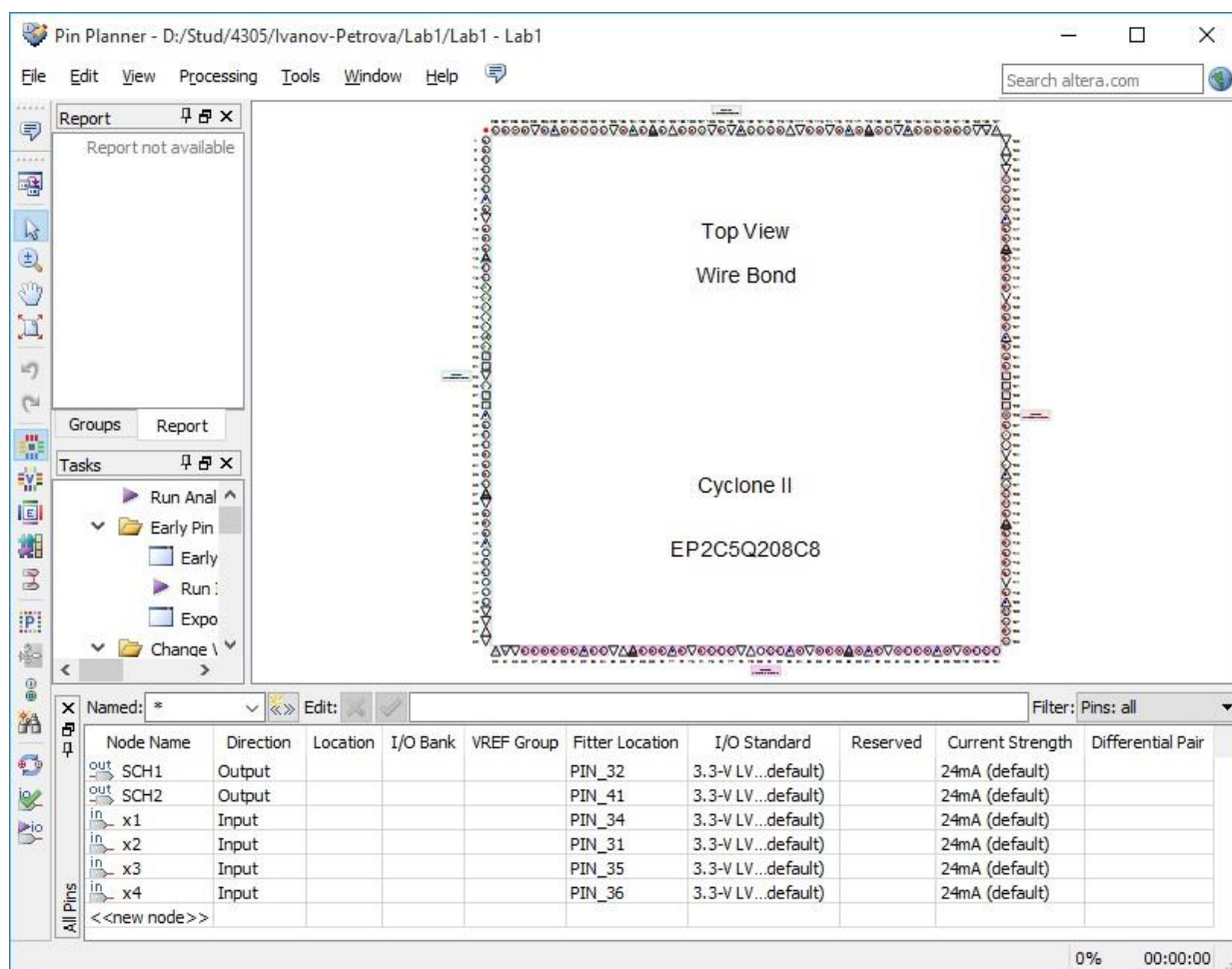



Рис. П3. Вид окна «Pin Planner»

3. После назначения контактов необходимо выполнить повторную компиляцию проекта. По завершении компиляции допустимо появление некоторого количества предупреждений. Предупреждения необходимо тщательно проверить. Возможно их оставление без реакции, если это не приведет к фатальным последствиям. Результатом компиляции является создание в директории проекта файлов конфигурации *.sof и *.pof.

4. Выполнить конфигурирование кристалла с использованием утилиты **Programmer**. Она вызывается пиктограммой  на панели инструментов менеджера проекта либо соответствующей командой меню **Tools**. Окно программирования представлено на рис. П4. В момент запуска утилиты учебная плата должна быть подключена к компьютеру, на нее подано питание и подключен программатор. В этом случае в поле **Hardware** будет указано устройство программирования USB-Blaster [USB-0].

В этом окне необходимо выбрать режим (mode) **Active Serial Programming**, согласившись с удалением всех устройств из списка при смене режима программирования. Затем с помощью кнопки **Add file** добавляется файл *.pof,

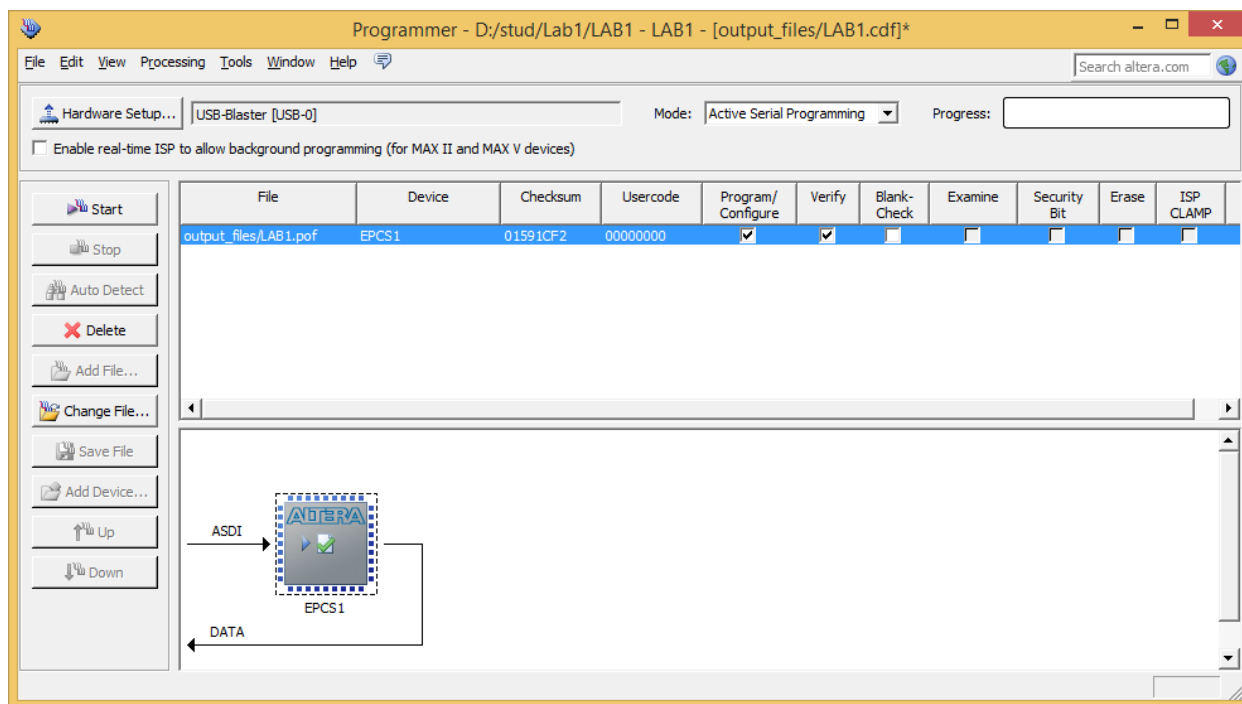


Рис. П4. Вид окна «Programmer»

созданный в подкаталоге *Output files* в результате компиляции проекта, и отметить пункты программирования *Program / Configure* и верификации загруженной конфигурации *Verify*.

Непосредственно программирование осуществляется нажатием кнопки *Start*. После этого в поле *Progress* будет визуально отображаться процесс загрузки конфигурации в кристалл, и при успешном его завершении в поле установится значение 100 %.

2. Порядок работы с осциллографом

1. Для исследования работы схемы выполнить загрузку проекта в учебную плату, используя последовательность действий, приведенную в прил. 1.

2. Включить двухканальный режим осциллографа, в этом режиме на экране можно наблюдать одновременно два сигнала (тактовый сигнал и выходной сигнал триггерного элемента).

3. Настроить каналы осциллографа. Поставить каналы в измерение постоянного напряжения (DC). Регулятор чувствительности вертикального отклонения (масштаб по вертикали) установить на значение, позволяющее вывести на экран без наложений два сигнала с диапазоном 3 В.

4. Настроить развертку (масштаб по горизонтали). При наблюдении одного периода сигнала частота развертки должна быть равна частоте исследуемого напряжения, при наблюдении n периодов – в n раз ниже. Минимальная частота развертки должна быть такой, чтобы изображение не мерцало на экране. Рекомендуется установить одинаковые значения развертки для обоих каналов.

Конкретные параметры развертки определяются частотой работы схемы: частотой генератора тактового сигнала, установленного на плате, и разрядностью счетчика, понижающего частоту, а также количеством периодов наблюдаемых сигналов, выводимых на экран, с учетом соотношения $T_z = 8T_{\text{clock}}$, где T_{clock} – период рабочей частоты схемы.

5. Выбрать тип синхронизации. При выборе внутренней синхронизации, т. е. синхронизации исследуемым сигналом, могут возникнуть сложности с получением устойчивого изображения на экране. Использование внешней синхронизации, позволяющей управлять моментом появления сигнала на развертке, решает эту проблему. В качестве внешнего синхросигнала, подаваемого на специальный вход осциллографа, можно использовать старший разряд блока генерации исходных данных, либо старший разряд счетчика, понижающего частоту тактового сигнала.

6. Подключить щупы осциллографа. Рекомендуется использовать удлинители: одиночные провода с установленными с одной стороны одиночными гнездами (подключаются к контактам разъема отладочной платы) и штырями с другой, подключаемыми к щупам осциллографа. Контакт щупа, подключаемый к земле, через удлинитель соединяется с цепью 0 В, также выведенной на внешние контакты разъема макетной платы.

3. Номера контактов ПЛИС учебной платы

Контакты движковых переключателей

Номер бита	0	1	2	3	4	5	6	7
Номер контакта FPGA	170	169	168	165	164	163	162	161

Диоды светодиодной линейки

Номер светодиода светодиодной линейки	0	1	2	3	4	5	6	7
Номер контакта FPGA	171	173	175	176	179	180	182	185

Сигналы разъема порта А

Номер конт.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Сигнал	0V	-5V	Номер контакта FPGA												+3.3V	+5V
			37	39	35	40	34	41	33	43	31	44	30	46		

Сигналы разъема порта В

Номер конт.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Сигнал	0V	-5V	Номер контакта FPGA												+3.3V	+5V
			60	61	59	63	58	64	57	67	56	68	48	69		

Подключение семисегментного индикатора

Номер индикатора	Сегмент	D0	D1	D2	D3	D4	D5	D6	DP
0	Номер контакта FPGA	191	187	188	197	195	192	193	189
1	Номер контакта FPGA	203	198	199	208	207	205	206	201

Тактовая кнопка – номер контакта 160.

Тактовый сигнал 40 МГц – номер контакта 132.

Оглавление

Введение	3
Лабораторная работа 1. ЗНАКОМСТВО С СИСТЕМОЙ ПРОЕКТИРОВАНИЯ QUARTUS II, РЕАЛИЗАЦИЯ КОМБИНАЦИОННОЙ СХЕМЫ	4
Краткие теоретические сведения	4
Порядок работы в системе проектирования Quartus II	5
Задание на работу	13
Последовательность выполнения работы	13
Содержание отчета	14
Лабораторная работа 2. ПРОЕКТИРОВАНИЕ КОМБИНАЦИОННОГО УЗЛА НА ОСНОВЕ ДЕШИФРАТОРА И МУЛЬТИПЛЕКСОРА	15
Краткие теоретические сведения	15
Задание на работу	16
Последовательность выполнения работы	18
Содержание отчета	21
Лабораторная работа 3. ПРОЕКТИРОВАНИЕ ТИПОВОГО КОМБИНАЦИОННОГО УЗЛА	22
Краткие теоретические сведения	22
Задание на лабораторную работу	23
Последовательность выполнения работы	24
Содержание отчета	24
Лабораторная работа 4. ТРИГГЕРНЫЕ УСТРОЙСТВА	25
Краткие теоретические сведения	25
Задания на лабораторную работу	27
Последовательность выполнения работы	30
Содержание отчета	31
Лабораторная работа 5. ИССЛЕДОВАНИЕ РЕГИСТРОВ	32
Краткие теоретические сведения	32
Задания на лабораторную работу	33
Последовательность выполнения работы	35
Содержание отчета	36
Лабораторная работа 6. ПРОЕКТИРОВАНИЕ РАСПРЕДЕЛИТЕЛЕЙ ТАКТОВЫХ СИГНАЛОВ	37
Краткие теоретические сведения	37

Задания на лабораторную работу	39
Последовательность выполнения работы.....	40
Содержание отчета.....	40
Лабораторная работа 7. ПРОЕКТИРОВАНИЕ ДВОИЧНЫХ СЧЕТЧИКОВ.....	41
Краткие теоретические сведения	41
Задания на лабораторную работу	46
Последовательность выполнения работы.....	46
Содержание отчета.....	47
Лабораторная работа 8. ПРОЕКТИРОВАНИЕ КОНЕЧНЫХ АВТОМАТОВ	48
Краткие теоретические сведения	48
Задания на лабораторную работу	53
Последовательность выполнения работы.....	53
Содержание отчета.....	54
Список литературы	55
ПРИЛОЖЕНИЯ	56
1. Порядок проведения макетного эксперимента	56
2. Порядок работы с осциллографом.....	60
3. Номера контактов ПЛИС учебной платы	61

Бондаренко Павел Николаевич,
Буренева Ольга Игоревна,
Головина Людмила Константиновна

**Узлы и устройства
средств вычислительной техники**

Учебно-методическое пособие

Редактор Н. В. Лукина

Подписано в печать 28.12.17. Формат 60×84 1/16.
Бумага офсетная. Печать цифровая. Печ. л. 4,0.
Гарнитура «Times New Roman». Тираж 65 экз. Заказ .

Издательство СПбГЭТУ «ЛЭТИ»
197376, С.-Петербург, ул. Проф. Попова, 5