

УДК 004.3

А.В. Греков¹, С.Ф. Тюрин²

¹Пермский военный институт войск национальной гвардии Российской Федерации, Пермь, Россия

²Пермский национальный исследовательский политехнический университет, Пермь, Россия

МЕТОДИЧЕСКИЕ ПРИЕМЫ ПРОВЕДЕНИЯ ЛАБОРАТОРНОГО ЗАНЯТИЯ ПО СИНТЕЗУ КОМБИНАЦИОННОЙ СХЕМЫ НА «ЖЕСТКОЙ» ЛОГИКЕ В СИСТЕМЕ QUARTUS II

В настоящее время программируемые логические схемы (ПЛИС) активно применяются при разработке отказо- и сбоеустойчивых цифровых систем. ПЛИС используют разработчики высоконадежных систем управления летательными аппаратами, например, на Пермском предприятии «Стар». Знания и умения в области синтеза систем на ПЛИС необходимы современному высококвалифицированному специалисту-инженеру. В статье приводится методика проведения лабораторного занятия по синтезу цифровых автоматов на «жесткой» логике с применением системы автоматизированного проектирования Quartus II фирмы Altera.

Лабораторное занятие – одна из форм учебной работы, при которой обучаемые по заданию и под руководством преподавателя самостоятельно проводят опыты, измерения, элементарные исследования на основе специально разработанных заданий в условиях лаборатории. Содержание лабораторного занятия определяется перечнем профессиональных умений по конкретной учебной дисциплине (модулю), а также характеристикой профессиональной деятельности выпускников, требованиями к результатам освоения основной профессиональной образовательной программы. Дидактическими целями лабораторных занятий являются: овладение техникой эксперимента; формирование умений решать практические задачи путем постановки опыта; экспериментальное подтверждение изученных теоретических положений, экспериментальная проверка формул, расчетов. Благодаря лабораторным занятиям достигается цель – формирование следующих умений и навыков: наблюдать, сравнивать, сопоставлять, анализировать, делать выводы и обобщения; самостоятельно вести исследования; пользоваться различными приемами измерений, оформлять результат в виде таблиц, схем, графиков; получать профессиональные умения и навыки работы с различными приборами, аппаратурой, установками и другими техническими средствами при проведении опытов. Особое внимание уделяется определению алгоритма проведения эксперимента.

Ключевые слова: лабораторное занятие, комбинационная схема, Quartus II, «жесткая» логика, проект.

A.V. Grekov¹ S.F. Tyurin²

¹Perm Military Institute of the National Guard Forces of the Russian Federation,
Perm, Russian Federation

²Perm National Research Polytechnic University, Perm, Russian Federation

THE METHODOICAL TECHNIQUE OF THE CONDUCTION OF LABORATORIAL CLASSES ON THE SYNTHESIS OF COMBINATIONAL CIRCUITS ON THE "HARD" LOGIC IN THE QUARTUS II

Currently, programmable logic (FPGAs) are used extensively in the development of fault-tolerant digital systems. FPGA designers use a highly reliable aircraft control systems, for example, in the Perm company "Star". Modern highly qualified engineer needs the knowledge and skills in the field of synthesis of systems on FPGAs. The article is devoted to the methods of conducting of laboratory lesson on the synthesis of digital automata "hard" logic with computer-aided design system Quartus II from Altera.

Laboratory lesson – one of the forms of study, in which the trainees on the instructions and under the supervision of the teacher carry out their own experiments, measurements, basic research on the basis of specially developed tasks in the laboratory. The contents of the laboratory exercises is determined by the list of professional skills in specific academic disciplines (modules), as well as the characteristic of the professional activity of graduates, the requirements for the results of development of the basic professional educational program. Didactic objectives labs are mastering the technique of the experiment; formation of skills to solve practical problems by raising expertise; experimental confirmation of theoretical positions studied, experimental testing of formulas, calculations. As a result of laboratory studies achieved the formation of the following skills: to observe, compare, compare, analyze, draw conclusions and generalizations; conduct independent research; using different measurement techniques, draw up the result in the form of tables, charts, graphs; get professional skills to work with different devices, equipment, installations and other technical means in experiments. The special attention is given to the definition of the algorithm of the experiment.

Keywords: laboratory lesson, combinational circuit, Quartus II, "hard" logic, project.

Введение. Существует множество программ-симуляторов, заменяющих реальную радиоэлектронную аппаратуру (РЭА) виртуальными моделями [1, 2]. Такие программы позволяют без проведения сборки реального устройства выполнить отладку работы схемы [3], найти ошибки, допущенные на стадии проектирования, снять требуемые параметры и многое другое [4, 5].

Одной из таких систем схемотехнического моделирования является Quartus II [3, 6–13]. Пользователю доступна обширная библиотека моделей элементов, пополнять которую можно самостоятельно. Естественно, для этого нужно досконально знать работу элемента и уметь программировать. С этой целью и проводятся лабораторные занятия.

В структуре проведения лабораторного занятия можно выделить три части.

1. Вводная часть:

- сообщение темы, постановка целей;
- повторение теоретических знаний, необходимых для работы с оборудованием, осуществления эксперимента или другой практической деятельности;
- выдача задания;
- определение алгоритма проведения эксперимента или другой практической деятельности;
- инструктаж по требованиям безопасности;
- ознакомление со способами фиксации полученных результатов;
- допуск к выполнению работы.

2. Самостоятельная работа обучающегося:

- определение путей решения поставленной задачи;
- выработка последовательности выполнения необходимых действий;
- проведение эксперимента (выполнение заданий, задач, упражнений);
- составление отчета;
- обобщение и систематизация полученных результатов (таблицы, графики, схемы и т.п.).

3. Заключительная часть:

- подведение итогов занятия: анализ хода выполнения и результатов работы обучающихся, выявление возможных ошибок и определение причин их возникновения;
- защита выполненной работы.

1. Постановка задачи. Задание: построить схему реализации заданной функции трех переменных в базисе И-НЕ по вариантам. Варианты заданий соответствуют номеру по списку группы.

Пример. Пусть необходимо синтезировать схему реализации переключатальной функции (ПФ) трех переменных №132. Построим таблицу истинности ПФ №132₁₀ (таблица).

Переменные			BC	$f(abc)$	
a	b	c			
0	0	0	0	0	2^0
0	0	1	1	0	2^1
0	1	0	2	0	2^2
0	1	1	3	1	2^3
1	0	0	4	0	2^4
1	0	1	5	1	2^5
1	1	0	6	1	2^6
1	1	1	7	1	2^7

После минимизации заданной функции №132₁₀, например, по кубу соседних чисел, получаем покрытие $(3\vee 7) \vee (5\vee 7) \vee (6\vee 7)$ и функцию в дизъюнктивной нормальной форме (ДНФ):

$$f(abc) = ab \vee bc \vee ac.$$

Представим функцию в базисе И–НЕ:

$$f(abc) = \overline{\overline{ab \vee bc \vee ac}} = \overline{\overline{ab} \overline{bc} \overline{ac}}.$$

Схема в международном стандарте имеет вид (рис. 1).

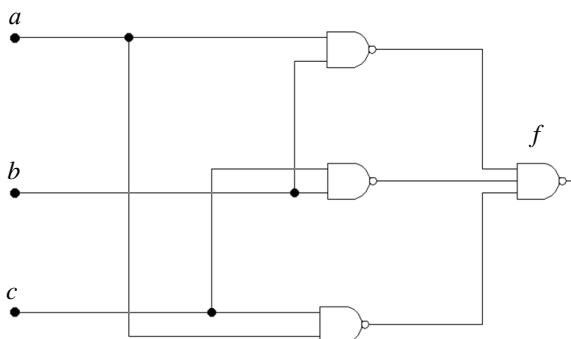


Рис. 1. Схема реализации функции №132₁₀

Построим эту схему в системе Quartus II [3].

2. Проведение эксперимента

2.1. Создание папки проекта. Запустить приложение Quartus II (ярлык показан на рис. 2).

Создадим новый проект. Для этого в меню «File» необходимо выбрать пункт «New project wizard». В появившемся окне «New Project Wizard: Directory, Name, and Top-Level Entity...» («Мастер Создания Проекта: Папка, Имя и Модуль Верхнего Уровня...») щелкнуть по кнопке с тремя точками у верхнего поля ввода. В открывшемся новом окне «Select Directory» («Выбрать Папку») создать новую папку Lab1 для нового проекта устройства (папка необходима, поскольку проект содержит порядка пятидесяти файлов), и открыть ее (рис. 3).



Рис. 2. Ярлык приложения Quartus II

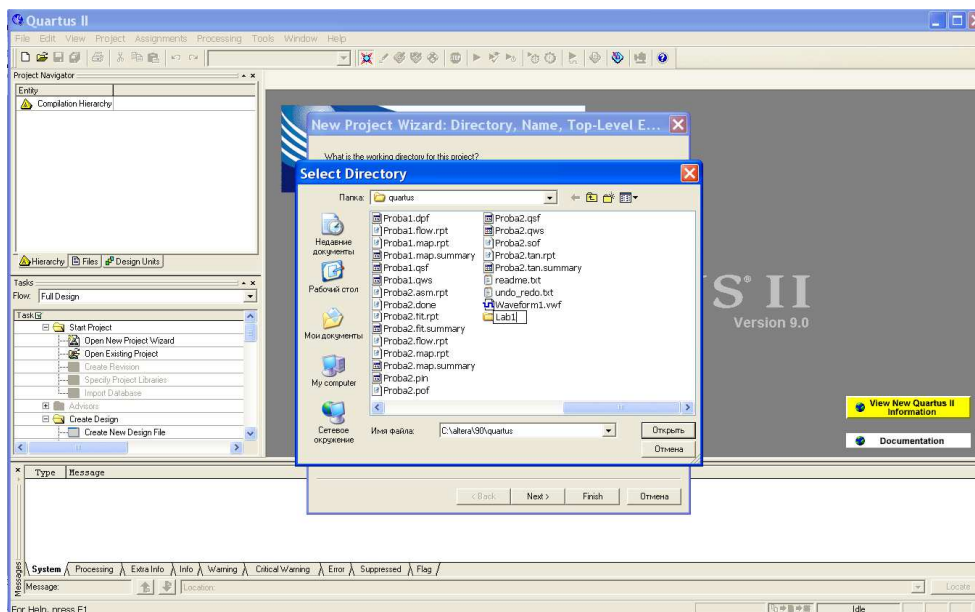


Рис. 3. Создание папки проекта

После возвращения в окно «New Project Wizard: Directory, Name, and Top-Level Entity...» нужно убедиться в том, что в верхнем поле ввода занесен полный маршрут ко вновь созданной папке, во второе поле занести имя проекта Lab1, совпадающее с именем папки, и в третье (нижнее) поле – имя модуля верхнего уровня Lab1, также совпадающее с именем папки. Щелкнуть по кнопке «Finish» («Завершение»). В начале каждого последующего сеанса работы с созданным проектом необходимо выбрать в строке меню «File\Open Project...». В появившемся окне «Open Project» найти папку проекта и, выделив файл проекта, щелкнуть по кнопке «Открыть».

2.2. Создание конструкторского файла схемы. Выбрать в строке меню «File\New...» («Файл\Новый»). В появившемся окне «New» на закладке «Design Files» («Конструкторские файлы») выбрать строку «Block Diagram/Schematic File» («Файл Блок-схемы/Схемный файл»). Щелкнуть по кнопке «ОК» в нижней части окна. На рабочем поле приложения появится окно созданного файла с именем Block1.bdf. Его расширение означает, что это «Block Design File», т.е. файл с графическим представлением проекта в виде схемы (рис. 4).

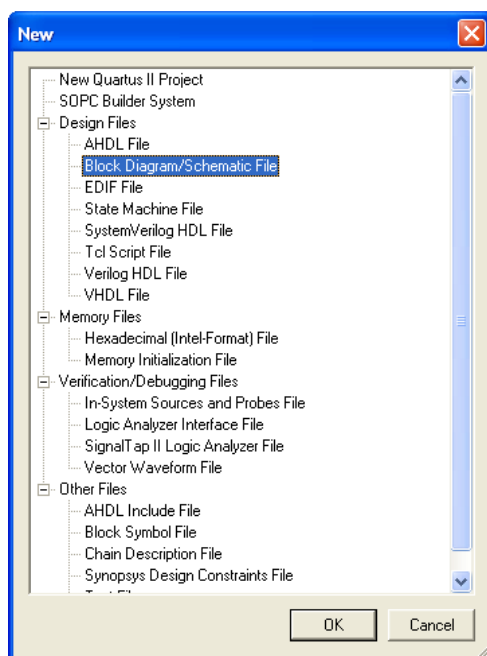


Рис. 4. Выбор файла схемы в окне «New»

Присвоим имя Lab1 этому файлу. Выбрать в строке меню «File\Save As...» («Файл\Сохранить как...»). В появившемся диалоговом окне «Сохранить как» будет предложено сохранить файл с именем нашего проекта и расширением bdf. Следует принять предложение и щелкнуть по кнопке «Сохранить». Получим файл схемы Lab1. bdf.

2.3. Набор схемы. Щелкнуть правой кнопкой мыши по рабочему полю файла *.bdf. В появившемся контекстном меню выбрать строку «Insert» («Вставить») и далее «Symbol...» («Символ...»). После выполнения команды контекстного меню «Insert\Symbol...» на экране появится окно «Symbol». В поле «Libraries:» («Библиотеки:») этого окна необходимо щелкнуть по кнопке со знаком «плюс» в строке с маршрутом к папке библиотек пакета. После появления списка библиотек выбрать «primitives». Появится список библиотеки «primitives» («примитивы»). В списке примитивов выбираем раздел «logic» («логические элементы»). Выбираем требуемые элементы и помещаем их на наборное поле. Нажимаем ОК и помещаем элемент на наборное поле. Скопируем элемент и поместим две копии на наборное поле. Выберем из библиотеки трехвходовой элемент 3 И – НЕ (NAND3) и поместим его на наборное поле. Теперь создадим входы-выходы схемы (pin).

Зададим входы (input) a, b, c . Щелкнуть правой кнопкой по символу входного вывода на рабочем поле. В появившемся контекстном меню выбрать строку «Properties...» («Свойства...»).

Входным выводам может быть присвоено значение по умолчанию «GND» или «VCC». Это значение будет использовано пакетом в случае, если данный модуль («entity») не является модулем верхнего уровня иерархии, входит в проект как компонент, и в конструкторском файле верхнего уровня этот вывод оказался неподключенным. Выводы в модуле верхнего уровня являются выводами микросхемы, выводы в модуле нижнего уровня являются точками соединения с другими модулями внутри проекта в целом. Создадим выход input z и поместим выход на наборное поле.

Создадим соединения элементов. Нажать кнопку «Orthogonal Node Tool» («Ввод ортогональных проводников») на вертикальной панели инструментов. Ввести требуемые соединения элементов в соответствии с заданной схемой, нажимая левую кнопку мыши в начальной точке и отпуская ее в конечной точке вводимого проводника. В результате получим схему (рис. 5).

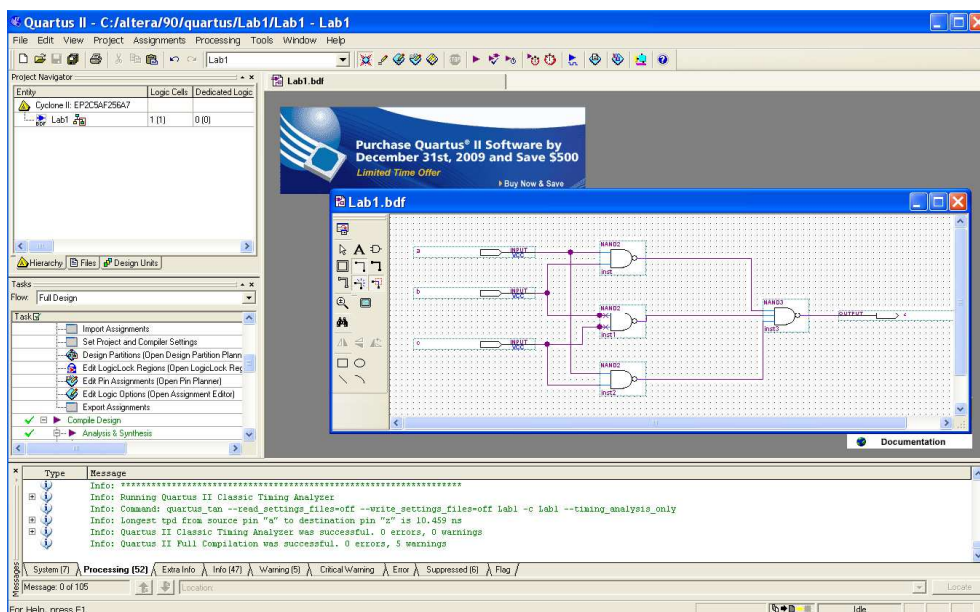


Рис. 5. Схема заданной функции

2.4. Задание ПЛИС. После того, как все проводники будут установлены, проект готов к компиляции – преобразованию введенной схемы в схему из реальных аппаратных блоков какой-либо конкретной микросхемы ПЛИС фирмы Altera.

Записать подготовленную схему в файл проекта, выбрав в главном меню строку «Save» («Сохранить с имеющимся именем»). Выбрать в строке меню «Assignments\Device...» («Назначения\Микросхема...»). В открывшемся окне «Device» в списке «Family:» («Семейство:») выбрать семейство микросхем Cyclone II. В расположенном ниже списке «Available devices:» («Доступные микросхемы:») выбрать микросхему EP2C5F256C7.

2.5. Компиляция проекта. Выбрать в строке меню «Processing\Start Compilation» («Процессы\Запуск компиляции»). Обращаем внимание, что на схеме не должно быть одинаковых имен элементов и контактов, иначе при компиляции будет зафиксирована ошибка.

В процессе компиляции на экране присутствуют следующие окна:

- окно иерархии проекта (состав и подчиненность модулей) – было на экране до компиляции;

- окно конструкторского файла («Block Design File») – было до компиляции;

- окно состояния процесса («Status window») – было до компиляции незаполненным; в ходе компиляции отражает текущий объем выполнения отдельных этапов в процентах и фиксирует время, затраченное на каждый этап и на весь процесс;

- окно отчета о компиляции («Compilation Report») – появилось в процессе компиляции; отражает статистическую информацию о проекте;

- окно сообщений («Message Window») – было незаполненным до компиляции; выводит информационные сообщения на различных этапах обработки данных проекта, а также предупреждения (warnings) и сообщения об ошибках (error messages), если они имеются;

- информационное окно («Quartus II») – появляется по окончании процесса компиляции; сообщает о его успешном или неудачном завершении.

Наибольший интерес представляют последние два окна. Информационное окно не позволит продолжать работу, пока вы не щелкните по кнопке «ОК» на его поле (предполагается по умолчанию, что разработчик прочтет выведенное в окне сообщение).

В окне сообщений, в случае аварийного завершения процесса компиляции, будут представлены сообщения об ошибках. Щелчок по сообщению об ошибке в логике проекта вызывает выделение на схеме элемента, к которому относится ошибка. (Эту ситуацию легко можно смоделировать, удалив, например, какой-либо необходимый проводник на схеме и запустив провокационный проект на компиляцию.)

3. Систематизация полученных результатов

3.1. Временное моделирование. После успешного завершения компиляции следует убедиться в правильном функционировании разработанного устройства, а также установить его основные временные параметры. Это можно осуществить, используя систему моделирования пакета – симулятор («Simulator»). Моделирование («Simulation») состоит в том, что на входные выходы спроектированного устройства подаются сигналы, представляющие последовательно, в различные промежутки времени, все возможные комбинации аргументов. Эти сигналы задаются с помощью так называемых временных диаграмм («Waveform»), составляемых разработчиком в редакторе временных диаграмм («Waveform Editor») и записываемых в файл *.vwf («Vector Waveform File»). В процессе моделирования симулятор определяет, какие сигналы и с какой задержкой разработанное устройство сформирует на выходных выводах, и представляет результаты также в виде временных диаграмм.

Выбрать в строке меню «Processing\Simulator Tool». В появившемся окне «Simulator Tool» в списке «Simulation mode:» («Режим моделирования:») выбрать строку «Timing» («Временной»).

При установке режима «Functional» («Функциональный») в результате моделирования будут получены идеальные временные диаграммы сигналов выходных узлов без учета реальных аппаратных задержек, что может быть полезно на предварительных этапах проектирования.

В режиме «Временной» будут получены диаграммы с задержками сигналов, определяемыми числом элементов на пути распространения сигнала и выбранным типом микросхемы. Кроме того, в режиме «Временной» симулятором обнаруживаются так называемые «Glitches» («Выбросы», «Иголки») – не обусловленные логикой проекта двукратные и более перепады логического уровня за короткие промежутки времени. Эти паразитные импульсы могут отрицательно повлиять на работоспособность устройства. Щелкнуть по кнопке с тремя точками справа от поля «Simulation input:» («Входные данные для моде-

лирования»). Убедиться в том, что в поле занесено имя подготовленного файла временных диаграмм *.vwf.

На панели «Simulation period» («Период моделирования») включить кнопку «Run simulation until all vector stimuli are used» («Моделировать, пока все входные сигналы не будут использованы»). На панели «Simulation options» («Параметры моделирования») установить «галочку» в строке «Automatically add pins to simulation outputs waveforms» («Автоматически включать выходные узлы в результирующие временные диаграммы»). Установка указанного выше свойства обеспечит автоматическое добавление во временные диаграммы, получаемые в результате моделирования, всех существующих выходных узлов с их диаграммами. Это свойство следует выключать, если вы хотите наблюдать диаграммы только части выходных узлов. В этом случае их необходимо ввести вручную.

Остальные свойства на панели «Simulation options» обеспечивают следующее:

- «Check outputs» («Контроль выходных диаграмм») – побуждает симулятор сравнивать сформированные проектировщиком в файле *.vwf ожидаемые диаграммы выходных сигналов с реально получаемыми в результате моделирования диаграммами этих сигналов;

- «Setup and hold time violation detection» («Обнаружение нарушений времени предустановки и времени удержания») – контролирует процесс моделирования на наличие нарушений параметров «время предустановки» и «время удержания» и сообщает о них в окне сообщений;

- «Glitch detection» («Обнаружение выбросов») – контролирует процесс временного моделирования на наличие выбросов и сообщает о них в окне сообщений;

- «Overwrite simulation input file with simulation results» («Обновление входного файла моделирования *.vwf с добавлением результатов моделирования») – включает режим обновления файла временных диаграмм.

Закреть окно «Simulator Tool».

Выбрать в строке меню «Processing\Start Simulation» («Процессы\Запуск моделирования»). По умолчанию на экране появится окно «Simulation Report» («Отчет о моделировании»), а в окне состояния процесса начнет отражаться объем выполнения процесса моделирования. После завершения моделирования на экране появится информационное окно с сообщением о завершении процесса. Его следует

закрывать, щелкнув по кнопке «ОК». (Процесс моделирования можно было бы запустить и непосредственно из окна «Simulator Tool», щелкнув по кнопке «Start». Однако в этом случае оказывается несколько более длинным процесс вывода результатов моделирования на экран.) Если моделирование завершилось успешно, то на экране останется окно «Simulation Report», представленное на рис. 6.

Сравниваем с таблицей истинности – все верно, только имеется определенное запаздывание формирования выходного сигнала. Так и должно быть. Определим временную задержку схемы.

3.2. Определение временной задержки схемы. Установить «Master Time Bar» (Главный маркер) на требуемый логический фронт входных сигналов, соответствующий моменту времени, например, на фронт набора 011, на котором формируется первая единица.

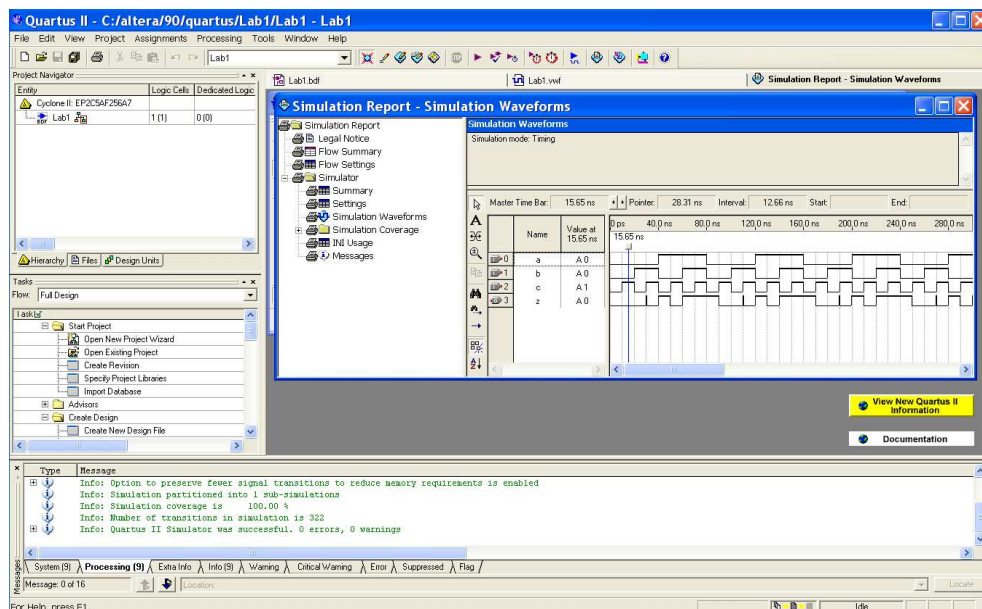


Рис. 6. Результаты моделирования

При однократном щелчке по кнопке перемещения Маркер переходит в указанном на кнопке направлении к положению следующего по времени логического фронта на выделенных диаграммах. Если ни одна диаграмма не выделена, учитываются все диаграммы, расположенные на рабочем поле. Теперь надо установить другой маркер – он называется «штриховой» на требуемый выходной сигнал – первую единицу. Щелкнуть правой кнопкой мыши по движку Главного маркера. В контекстном

меню выбрать строку «Insert Time Bar...» («Установить маркер...», далее – Штриховой маркер). В появившемся окне «Insert Time Bar», не меняя параметров, щелкнуть по кнопке «ОК». В месте установки Главного маркера появится Штриховой маркер (они будут наложены друг на друга).

Однократно щелкнуть по кнопке перемещения Главного маркера вправо. Он перейдет на следующий по времени логический фронт, т.е. фронт изменения функции z . У движка Штрихового маркера, оставшегося на старом месте, появится число, обозначающее его смещение по времени относительно Главного маркера.

Выводы. Таким образом, описанные приёмы обеспечивают качественное проведение лабораторного занятия по синтезу комбинационной схемы в системе Quartus II. При этом педагогические усилия должны быть направлены: на четкую постановку познавательной задачи; подробный инструктаж к работе (осмысление обучаемыми сущности задания, последовательности его выполнения); проверку теоретической и практической готовности обучаемых к занятию; выделение возможных затруднений в процессе работы. Также необходимы установка на самоконтроль и наблюдение за действиями обучаемых, регулирование темпа работы, помощь (при необходимости) и коррекция действий, проверка промежуточных результатов.

Библиографический список

1. Угрюмов Е.П. Цифровая схемотехника: учеб. пособие для вузов. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2007. – 800 с.
2. Потемкин И.С. Функциональные узлы цифровой автоматики. – М.: Энергоатомиздат, 1988. – 320 с.
3. Реализация цифровых автоматов в системе Quartus фирмы Altera: лаб. практикум / сост. С.Ф. Тюрин, А.В. Греков, О.А. Громов – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2011. – 134 с.
4. Инженерия прикладного программного обеспечения: монография / А.Р. Яматов, А.О. Суворов, А.А. Петренко, А.В. Греков; Пермский ВИ ВВ МВД России. – Пермь, 2016. – 89 с.
5. Греков А.В. Современные семейства программируемых логических интегральных схем FPGA фирмы Altera // Инновационные технологии: теория, инструменты, практика (Innotech 2013): матер. V Междунар. интернет-конф. молодых ученых, аспирантов и студентов. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2014. – С. 414–420.

6. Греков А.В., Успенко В.Б. Перспективные программируемые логические интегральные схемы FPGA фирмы Altera // В мире научных открытий. Естественные и технические науки / Научно-инновационный центр. – Красноярск, 2014. – № 6.1(54). – С. 518–534. DOI: 10.12731/wsd-2014-6.1-13

7. Греков А.В., Тюрин С.Ф. Функционально-полный толерантный логический элемент относительно комбинированной модели отказов // В мире научных открытий. Естественные и технические науки. – Красноярск, 2015. – № 12-3(72). – С. 724–741. DOI: 10.12731/wsd-2015-12.3-5

8. Тюрин С.Ф., Греков А.В., Коржев В.С. Скользящее резервирование с восстановлением на основе элементов с избыточным базисом // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. – СПб: Изд-во Санкт-Петербург. гос. политехн. ун-та, 2012. – № 5(157). – С. 38–44.

9. Tyurin S.F., Grekov A.V. Functionally Complete Tolerant Elements // International Journal of Applied Engineering Research. – 2015. – Vol. 10. – № 14. – P. 34433–34442.

10. Tyurin S.F., Grekov A.V. The Checked Logic Element ChLUT FPGA // Scientific Journal «In the World of Scientific Discoveries. Natural & Technical Sciences». – Krasnoyarsk: Publishing House Science and Innovation Center, 2014. – № 10(58). – P. 223–231. DOI: 10.12731/wsd-2014-10-17

11. Presentation on ALTERA'S FPGA Technology [Электронный ресурс]. – URL: <http://www.authorstream.com/Presentation/hsrathore158-1410279-fpga/> (дата обращения: 20.10.2016).

12. Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices [Электронный ресурс]. – URL: https://www.altera.com.cn/content/dam/altera-www/global/zh_CN/pdfs/literature/hb/stx3/stx3_siii51002.pdf (дата обращения: 20.10.2016).

13. Hierarchical Design [Электронный ресурс]. – URL: http://www.labcenter.com/products/pcb/schematic_hierarchy.cfm (дата обращения: 20.10.2016).

References

1. Ugriumov E.P. Tsifrovaia skhemotekhnika [Digital circuitry]. Saint Petersburg: BKhV-Peterburg, 2007. 800 p.

2. Potemkin I.S. Funktsional'nye uzly tsifrovoi avtomatiki [Functional units of digital automation]. Moscow: Energoatomizdat, 1988. 320 p.

3. Tiurin S.F., Grekov A.V., Gromov O.A. Realizatsiia tsifrovyykh avtomatov v sisteme Quartus firmy Altera [The implementation of digital machines in the company's Quartus Altera]. Permskii gosudarstvennyi natsional'nyi issledovatel'skii universitet, 2011. 134 p.

4. Iamatov A.R., Suvorov A.O., Petrenko A.A., Grekov A.V. Inzheneriia prikladnogo programmnoho obespecheniia: monografiia [Engineering of the applied software: monograph]. Permskii voennyi institut vnutrennikh voisk Ministerstva vnutrennikh del Rossiiskoi Federatsii, 2016. 89 p.

5. Grekov A.V. Sovremennye semeistva programmiruemykh logicheskikh integral'nykh skhem FPGA firmy Altera [Modern families of programmable logic integrated circuits FPGA by Altera]. *Materialy V Mezhdunarodnoi internet-konferentsii molodykh uchenykh, aspirantov i studentov "Innovatsionnye tekhnologii: teoriia, instrumenty, praktika (Innotech 2013)"*. Permskii natsional'nyi issledovatel'skii politekhnicheskii universitet, 2014, pp. 414-420.

6. Grekov A.V., Uspalenko V.B. Perspektivnye programmiruemye logicheskie integral'nye skhemy FPGA firmy Altera [Perspective programmable logic integrated circuits FPGA from Altera]. *V mire nauchnykh otkrytii. Estestvennye i tekhnicheskie nauki*. Krasnoiarsk, 2014, no. 6.1(54), pp. 518-534. DOI: 10.12731/wsd-2014-6.1-13

7. Grekov A.V., Tiurin S.F. Funktsional'no-polnyi tolerantnyi logicheskii element otnositel'no kombinirovannoi modeli otkazov [The functionally complete tolerant logic element relative to the combined model of failures]. *V mire nauchnykh otkrytii. Estestvennye i tekhnicheskie nauki*. Krasnoiarsk, 2015, no. 12-3(72), pp. 724-741. DOI: 10.12731/wsd-2015-12.3-5

8. Tiurin S.F., Grekov A.V., Korzhev V.S. Skol'ziashchee rezervirovanie s vosstanovleniem na osnove elementov s izbytochnym bazisom [Moving backup with recovery on the basis of the elements with excess basis]. *Nauchno-tekhnicheskie vedomosti Sankt-Peterburgskogo gosudarstvennogo politekhnicheskogo universiteta. Informatika. Telekommunikatsii. Upravlenie*. Sankt-Peterburgskii gosudarstvennyi politekhnicheskii universitet, 2012, no. 5(157), pp. 38-44.

9. Tyurin S.F., Grekov A.V. Functionally Complete Tolerant Elements. *International Journal of Applied Engineering Research*, 2015, vol. 10, no. 14, pp. 34433-34442.

10. Tyurin S.F., Grekov A.V. The Checked Logic Element ChLUT FPGA. *Scientific Journal "In the World of Scientific Discoveries. Natural &*

Technical Sciences". Krasnoyarsk: Publishing House Science and Innovation Center, 2014, no. 10(58), pp. 223-231. DOI: 10.12731/wsd-2014-10-17

11. Presentation on ALTERA'S FPGA Technology, available at: <http://www.authorstream.com/Presentation/hsrathore158-1410279-fpga/> (accessed 20 October 2016).

12. Logic Array Blocks and Adaptive Logic Modules in Stratix III Devices, available at: https://www.altera.com.cn/content/dam/altera-www/global/zh_CN/pdfs/literature/hb/stx3/stx3_siii51002.pdf (accessed 20 October 2016).

13. Hierarchical Design, available at: http://www.labcenter.com/products/pcb/schematic_hierarchy.cfm (accessed 20 October 2016).

Сведения об авторах

Греков Артем Владимирович (Пермь, Россия) – кандидат технических наук, доцент кафедры программного обеспечения вычислительной техники и автоматизированных систем Пермского военного института войск национальной гвардии Российской Федерации (614112, Пермь, ул. Гремячий Лог, 1, e-mail: grekartemvl@mail.ru).

Тюрин Сергей Феофентович (Пермь, Россия) – заслуженный изобретатель Российской Федерации, доктор технических наук, профессор кафедры автоматики и телемеханики Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр., 29, e-mail: tyurinsergfeo@yandex.ru).

About the authors

Grekov Artem Vladimirovich (Perm, Russian Federation) is a Ph.D. in Technical Sciences, Associate Professor at the Department of Software computer technology and automated systems Perm military institute of the National Guard Forces of the Russian Federation (614112, Perm, 1, Gremyachy Log str., grekartemvl@mail.ru).

Tyurin Sergey Feofentovich (Perm, Russian Federation) is a Honored Inventor of the Russian Federation, Doctor of Technical Sciences, Professor at the Department of Automation and Telemechanics Perm National Research Polytechnic University (614990, Perm, 29, Komsomolsky pr., e-mail: tyurinsergfeo@yandex.ru).

Получено 12.10.2016