МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

ЦИФРОВЫЕ УСТРОЙСТВА.
Лабораторный практикум по дисциплине

Учебно-методическое пособие
для обучающихся по направлениям подготовки
11.03.04 — Электроника и наноэлектроника,
11.03.03 — Конструирование и технология электронных средств

Севастополь
СевГУ
2020

УДК 621.396.1
Ц75

Р е ц е н з е н т ы:
Д.Г. Мурзин – доцент кафедры «Электронная техника»,
кандидат технических наук

Ответственный за выпуск:
Ю.П. Михайлюк – канд. техн. наук, доцент,
заведующий кафедрой «Электронная техника»

Составители: И. Б. Широков, Д. В. Начаров, И. В. Скорик

Ц75 Цифровые устройства. Лабораторный практикум по дисциплине : учебно-методическое пособие «» для обучающихся по направлениям подготовки 11.03.04 — Электроника и наноэлектроника, 11.03.03 — Конструирование и технология электронных средств / Сост. И. Б. Широков, Д. В. Начаров, И. В. Скорик ; Министерство науки и высшего образования Российской Федерации, Севастопольский государственный университет, Институт радиоэлектроники и информационной безопасности. – Севастополь: СевГУ, 2020. – 56 с. – Текст : электронный.

Целью методической разработки является оказание помощи обучающимся очной и заочной форм обучения в подготовке и проведении лабораторного практикума по дисциплине «Цифровые устройства», а также в получении ими теоретических знаний и практических навыков синтеза цифровых схем.

УДК 621.396.1

Рассмотрено и утверждено на заседании кафедры «Электронная техника», протокол № 10 от 15 июня 2020 г.

Изд. № 190/2020. Объем 3,5 п.л.
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Дисциплина «Цифровые устройства» изучается обучающимися очной и заочной форм обучения по направлениям подготовки 11.03.04 — Электроника и наноэлектроника, 11.03.03 — Конструирование и технология электронных средств. В методическую разработку включены пять лабораторных работ.

Методические указания ориентированы на приобретение навыков разработки цифровых схем, предполагающему использование современного подхода к проектированию с применением программируемых логических интегральных схем (ПЛИС) фирмы «Altera».

Методические указания по выполнению каждой лабораторной работы содержат следующие разделы: цель работы, теоретическая часть, порядок выполнения работы, содержание отчета, контрольные вопросы. Для выполнения всех лабораторных работ используется один и тот же макет. Описание лабораторного макета проведено в ознакомительной лабораторной работе № 1.

В основу лабораторного практикума положен фронтальный метод проведения работ: группа разделяется на бригады и одновременно выполняет одну и ту же лабораторную работу. Это позволяет синхронизировать процесс изучения теоретического материала с тематикой лабораторного практикума, а также дает возможность преподавателю по ходу занятия давать пояснения, обсуждать методику эксперимента и результаты со всей группой одновременно. Фронтальный метод позволяет также студентам осуществлять предварительную подготовку к лабораторным занятиям, поскольку их тематика заранее известна.

Бригада состоит, как правило, из 2-3 обучающихся. В процессе выполнения работы обучающиеся должны вести подробный протокол, в котором излагают весь ход работы, фиксируют результаты экспериментов. После завершения работы преподаватель анализирует полученные данные и подписывает протокол. На основании протокола каждый обучающийся оформляет отчет, который должен иметь титульный лист и содержать следующие разделы: цель работы, синтезируемые схемы и предварительные расчеты, экспериментальные данные (скриншоты синтезированных схем), выводы по работе.

ЛАБОРАТОРНАЯ РАБОТА № 1
ИЗУЧЕНИЕ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ «QUARTUS II»

1.1. Цель работы

Целью работы является ознакомление с САПР «QUARTUS II». Приобретение практических навыков проектирования логических схем, компиляции проектов, программирования ПЛИС и симуляции результатов проектирования.

1.2. Теоретическая часть

1.2.1. Общие положения

Система автоматизированного проектирования «QUARTUS II» — пользовательская система программирования логики упорядоченных структур) является полнофункциональной системой проектирования устройств на ПЛИС фирмы «Altera». Программное обеспечение этой САПР позволяет осуществить полную разработку проекта, его отладку, а также программирование ПЛИС выбранного типа. Следует отметить, что студенческая версия «QUARTUS II» (в дальнейшем «система») является бесплатной и свободно распространяется. Основные возможности указанной САПР «QUARTUS II» приведены в табл. 1.1.

САПР «QUARTUS II» позволяет использовать разнообразные средства описания проектов с иерархической структурой, осуществлять логический синтез, компиляцию с заданными временными параметрами, выполнять функциональное и временное тестирование (симуляцию проекта).

Создание проекта можно осуществлять с помощью графического редактора «Graphic Editor» системы, в виде текстового файла на языке Altera Hardware Description Language (AHDL) в любом внешнем редакторе или внутреннем «Text Editor», а также в виде временной диаграммы, с помощью «Waveform Editor». При этом для удобства работы со сложными проектами каждый подпроект может быть представлен в виде символа.

Таблица 1.1 — Основные возможности САПР «QUARTUS II»

Поддерживаемые устройства	FLEX 10K, FLEX 8000, FLEX 6000, MAX 9000, MAX 7000, MAX 3000, ACEX 1K, EPLDs.
Средства описания проекта	Схемный ввод, AHDL, интерфейс с САПР третьих фирм, встроенный топологический редактор, иерархическая структура проекта, встроенные библиотеки
Средства компиляции проекта	Логический синтез и трассировка, автоматическое обнаружение ошибок
Средства верификации проекта	Временной анализ, функциональное и временное моделирование, анализ сигналов, возможность использования программ моделирования третьих фирм

1.2.2. Создание проекта и моделирование в Quartus II

После запуска Quartus II приступаем к созданию нового проекта, запустив мастер New Project Wizard… из меню File. В открывшемся окне нажать кнопку Next, далее указываем путь к рабочей папке проекта, название проекта и имя файла верхнего уровня иерархии (имя основного файла) проекта о файлах проекта (рис. 1.1). После этого следует нажать кнопку Next.

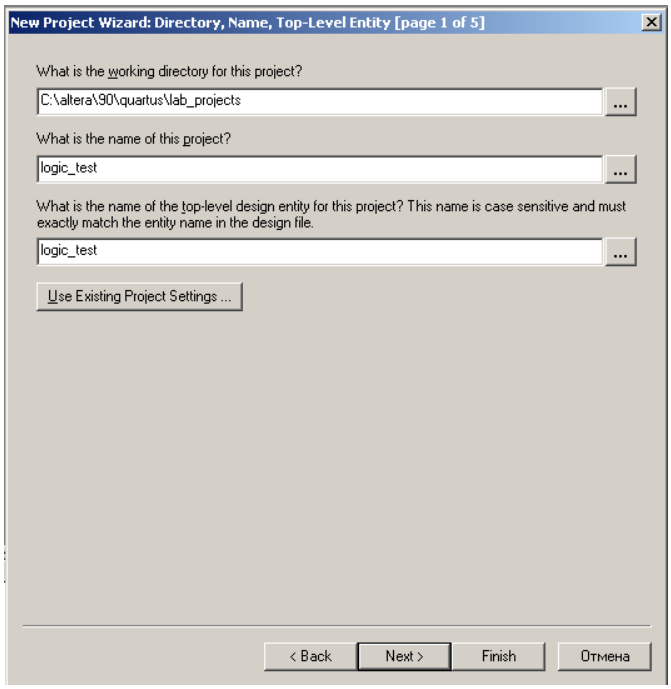


Рис. 1.1 — Создание нового проекта, шаг 1

В открывшемся окне (рис. 1.2) в создаваемый проект можно добавить существующие файлы. Кнопка Add All… позволяет добавить в проект все файлы, находящиеся в выбранной папке проекта. При создании проекта впервые этот шаг можно пропустить, нажав кнопку Next.

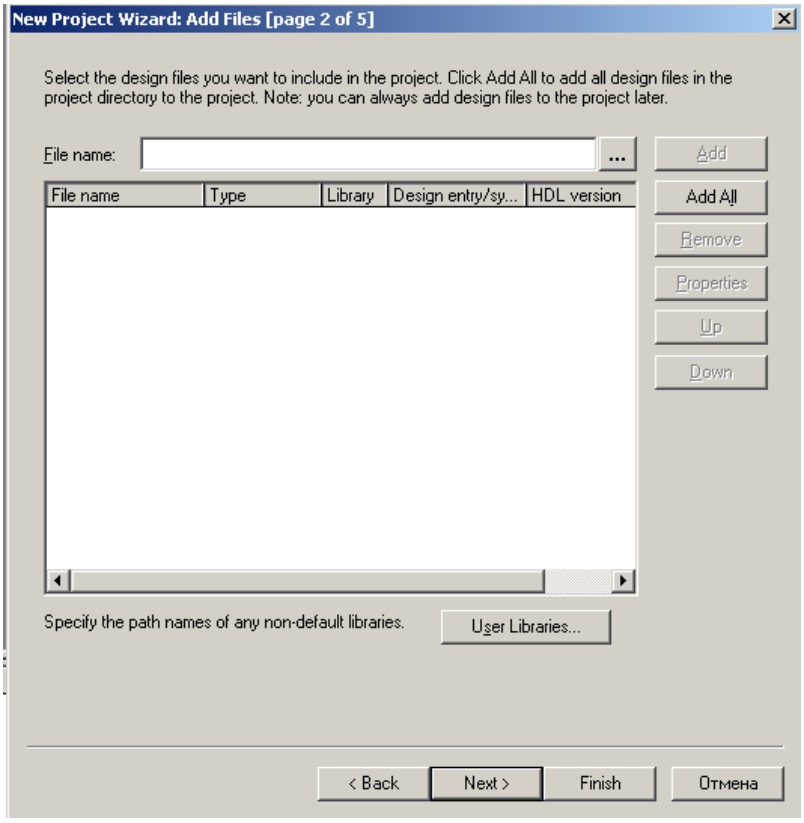


Рис. 1.2 — Создание нового проекта, шаг 2

Следующее окно мастера (рис. 1.3) позволяет выбрать семейство и наименование микросхемы ПЛИС, в которую будет загружен проект. В выпадающем списке Family выбираем семейство микросхем MAX700S, а в поле Available Devices выбираем микросхему EPM7128SLC84-15. После этого нажимаем кнопку Finish.

Чтобы создать новый файл проекта следует выбрать пункт New из меню File. Далее для создания логической схемы выберите тип файла Block Diagram/Schematic File (рис. 1.4) из раздела Design Files. В случае создания текстового описания модуля на языке Verilog следует выбрать тип Verilog HDL File. Нажмите кнопку OK, после чего открывается окно редактора логических схем. Сохраните файл схемы с расширение .bdf, выбрав пункт Save As из меню File.

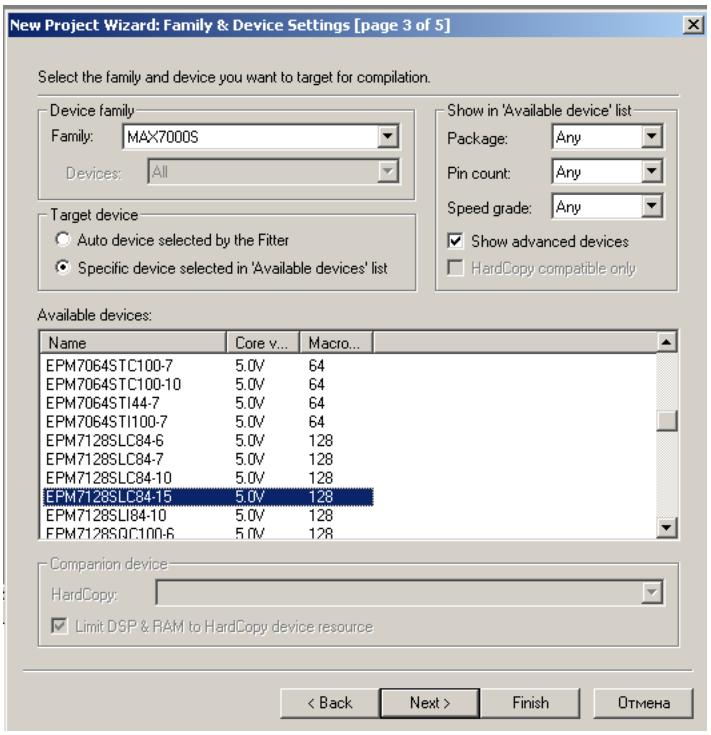


Рис. 1.3 — Создание нового проекта, шаг 3

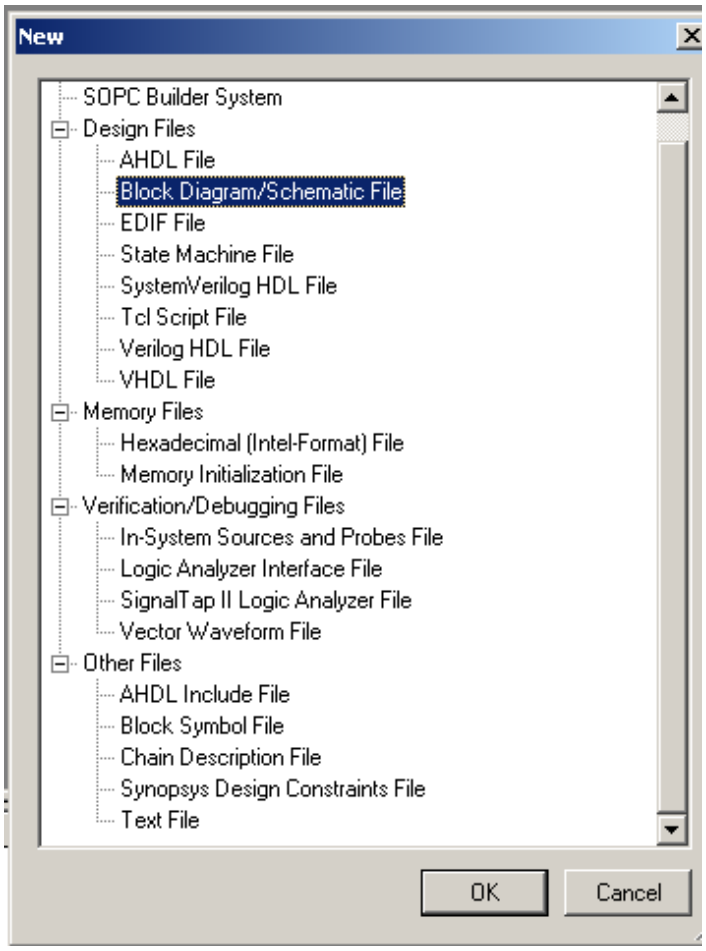


Рис. 1.4 — Выбор тип файла

Для добавления логических элементов в поле схемного редактора следует нажать правую кнопку мыши и в выпадающем меню выбрать пункт Insert → Symbol, либо выполнив двойной щелчок левой кнопкой мыши. В открывшемся окне (рис. 1.5) в поле Libraries из папки primitives выбираем необходимый логический элемент. Символы входных и выходных контактов схемы содержаться в папке pin. Входы и выходы можно переименовать, выполнив двойной щелчок по символу над текстом, либо выбрав пункт Properties из контекстного меню символа.

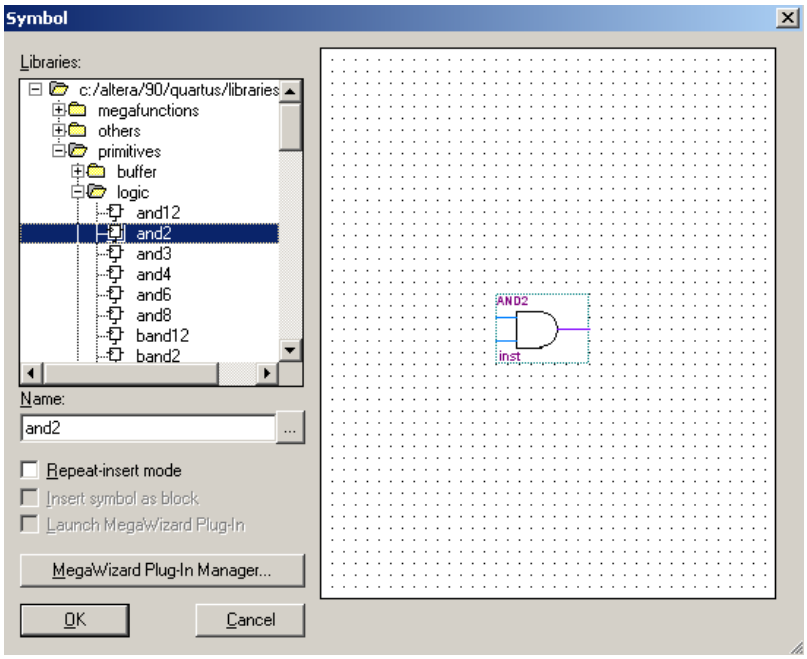


Рис. 1.5 — Выбор необходимого логического элемента

Для соединения элементов схемы можно использовать инструмент Orthogonal Node Tool  , доступный на панели инструментов редактора схем. После того как изображение схемы завершено (рис. 1.6), следует сохранить файл и выполнить компиляцию проекта, выбрав пункт Start Compilation из меню Processing, либо нажав на пиктограмму  .

При необходимости любой элемент схемы может быть использован как отдельный элемент. Для выполнения этой операции необходимо нарисовать желаемую схему, дать названия входным и выходным портам и сохранить ее с названием, соответствующим названию нового элемента. После этого с помощью команды (File>Create Default Symbol) сохраняем схему в виде элемента. При этом в качестве названия нового элемента будет использовано название

файла схемы, из которого он был создан. Входы нового элемента будут расположены слева, выходы — справа. Для использования нового элемента достаточно набрать его имя, аналогично стандартным элементам QUARTUS II. Двойной щелчок левой кнопкой мыши на таком элементе откроет его внутреннюю схему. Для внесения изменения в схему элемента следует открыть его, внести необходимые изменения, после чего сохранить их (File>Save или пиктограмма 3 на рис. 1.1), после чего повторно создать символ (File>Create Default Symbol). После этого можно закрывать внутреннюю схему символа. При создании нового элемента можно использовать другие элементы. При изменении элемента, для обновления проекта следует использовать команду (Symbol> Update Symbol).

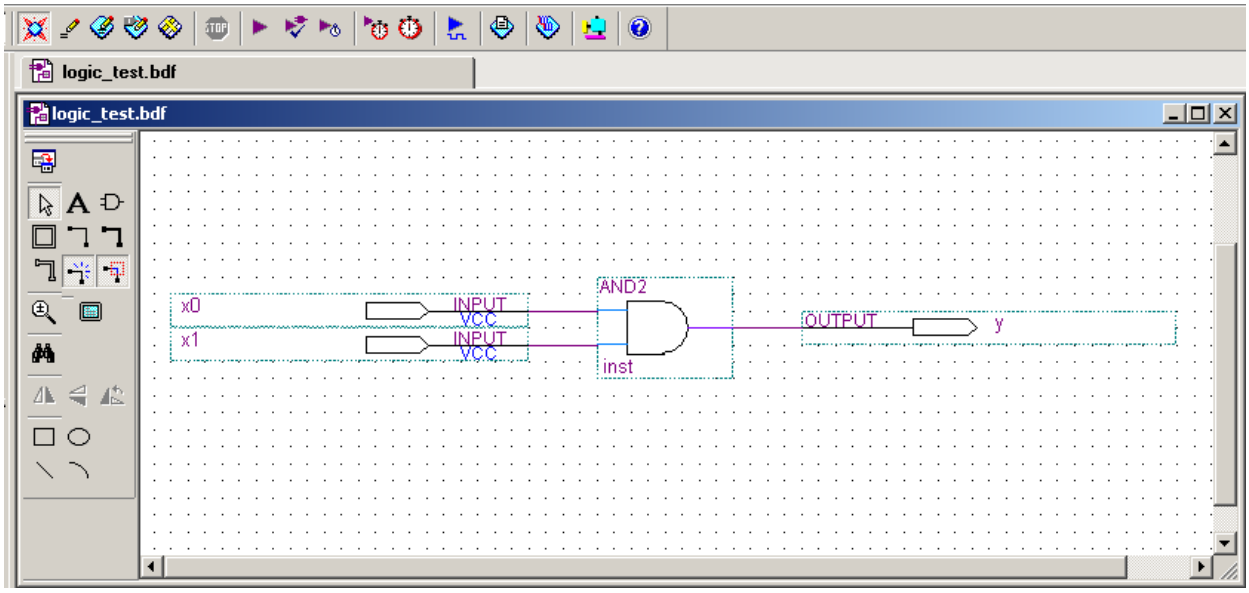


Рис. 1.6 — Соединение выводов элементов

1.2.3. Симулятор среды «QUARTUS II»

Симулятор среды Quartus II позволяет выполнять функциональное и временное моделирование цифровых устройств.

Перед выполнением функциональной симуляции необходимо создать специальный файл, содержащий список соединений логических элементов (netlist), выбрав пункт Generate Functional Simulation Netlist меню Processing.

Для выполнения моделирования создайте файл описания формы входных сигналов (файл симуляции). Для этого выберете пункт New меню File, затем в

появившемся диалоговом окне выберите тип файла Vector Waveform File из раздела Verification/Debugging Files (рис. 1.7).

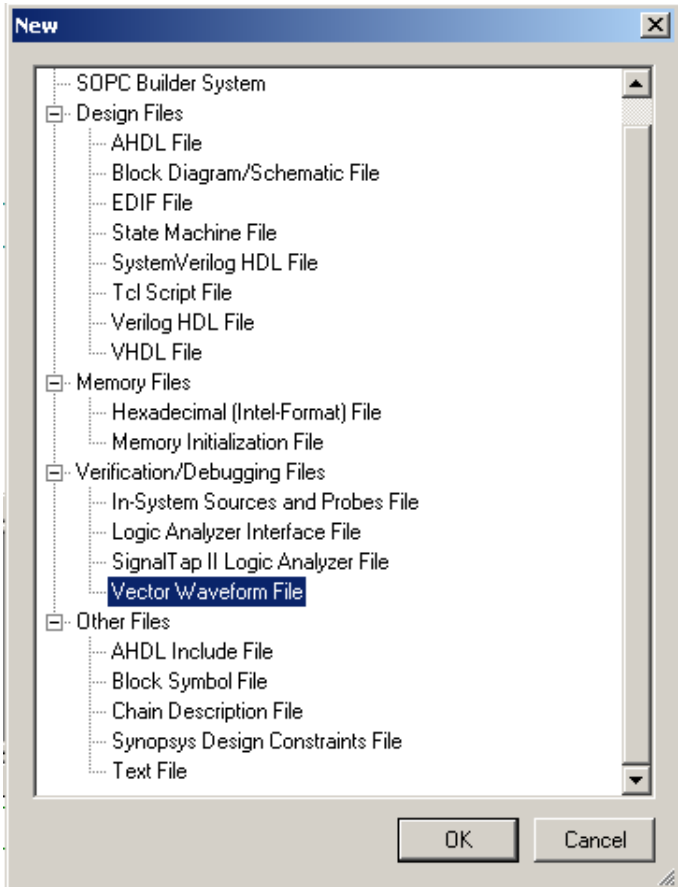


Рис. 1.7 — Выбор симулятора

Вид окна редактора графиков показан на рис. 1.81.

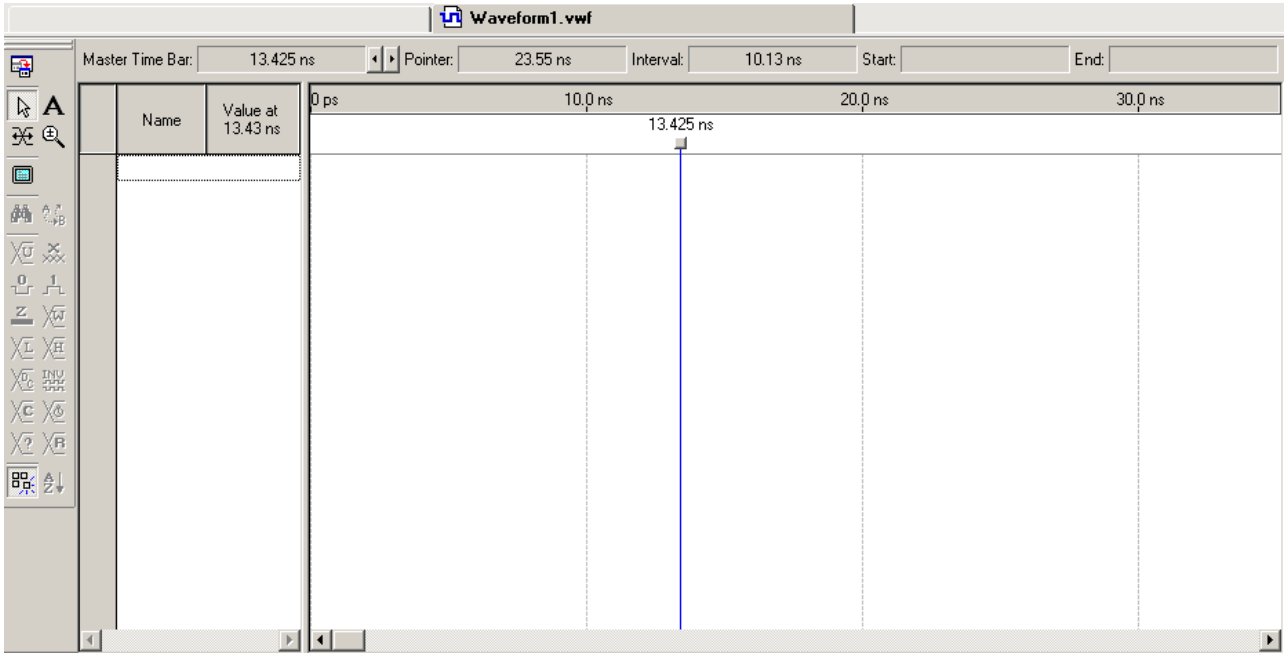


Рис. 1.8 — Окно редактора

Файл симуляции необходимо сохранить с расширением .vwf.

Для добавления сигналов в файл симуляции необходимо нажать правую кнопку мыши в левой панели Names и выбрать пункт Insert / Insert Node or Bus (рис. 1.9).

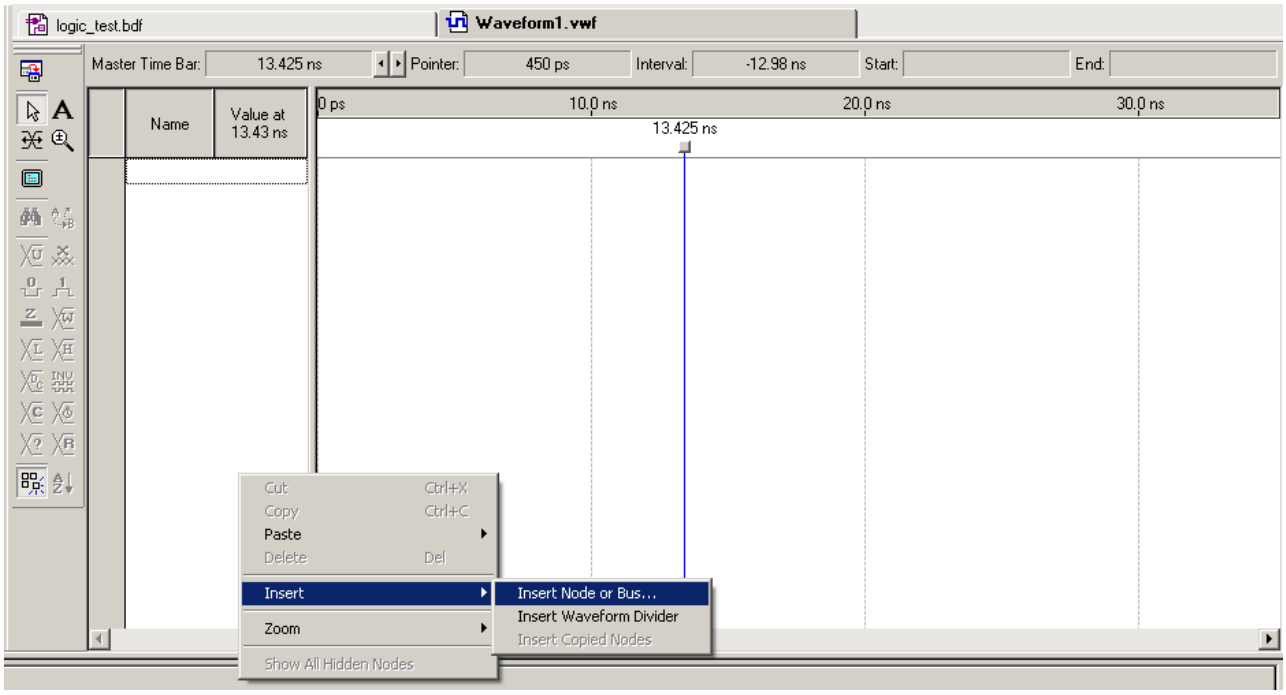


Рис. 1.9 — Вставку узлов контроля

Для выбора сигналов схемы в открывшемся диалоговом окне нажмите кнопку Node Finder (рис. 1.10).

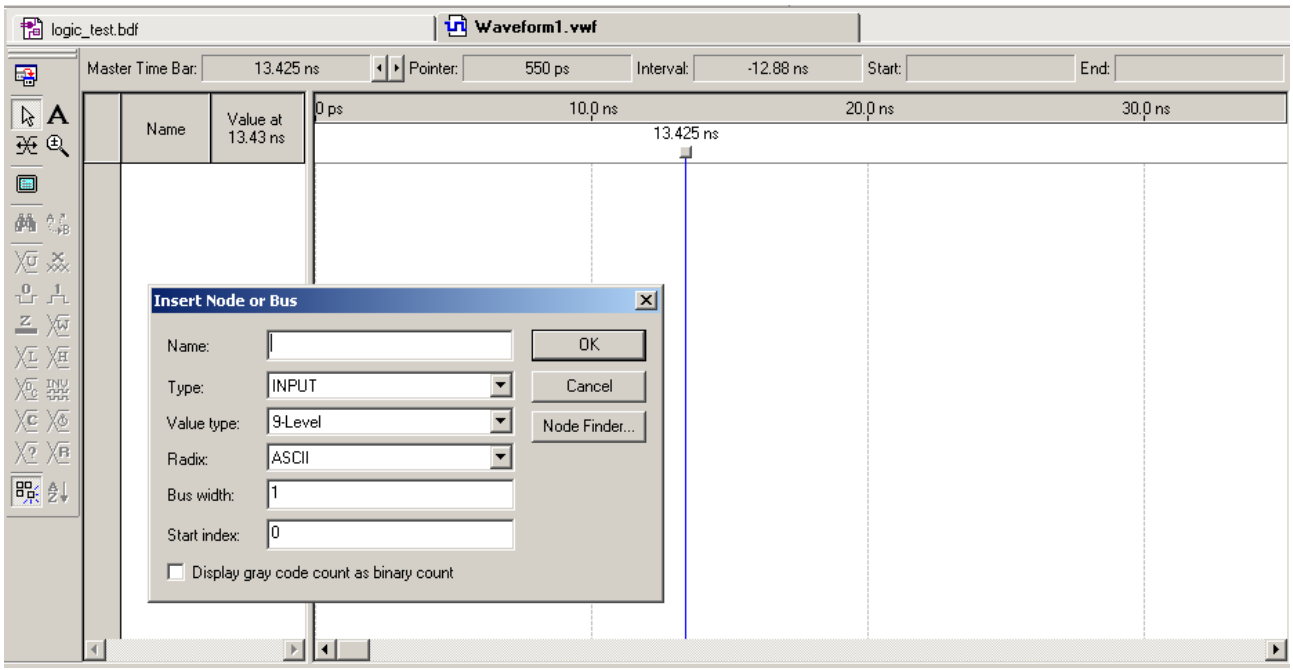


Рис. 1.10 — Формирование сигналов

Рассмотрим структуру диалогового окна Node Finder (рис. 1.11). В поле Look in указано имя проекта, в котором будем производиться поиск сигнала. В поле Named можно задать часть имени сигнала или провода. Например, если необходимо найти выход модуля с именем data, в поле Named можно указать значение *data*. После нажатия кнопки List в поле Nodes Found будет показан список всех сигналов, содержащих такой сигнал слово data в названии. Если указать просто *, то в результате поиска будут выведены все сигналы вне зависимости от имени.

В выпадающем списке Filter можно выбрать тип выводов микросхемы для поиска. Выберите тип Pins: All и нажмите кнопку List.

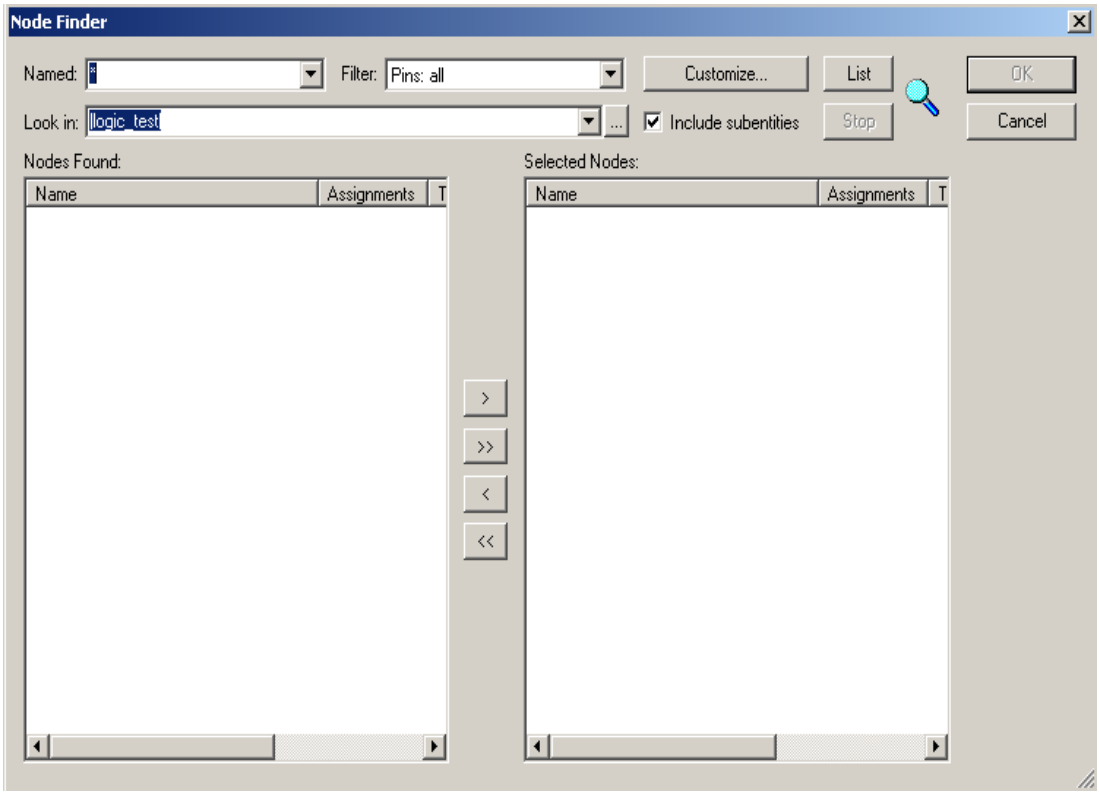


Рис. 1.11 — Структура диалдогового окна

Для выбора сигналов и их добавления на временную диаграмму используются кнопки > и >>. Выберите все сигналы в проекте, нажав кнопку >> (рис.12), далее нажмите кнопку OK.

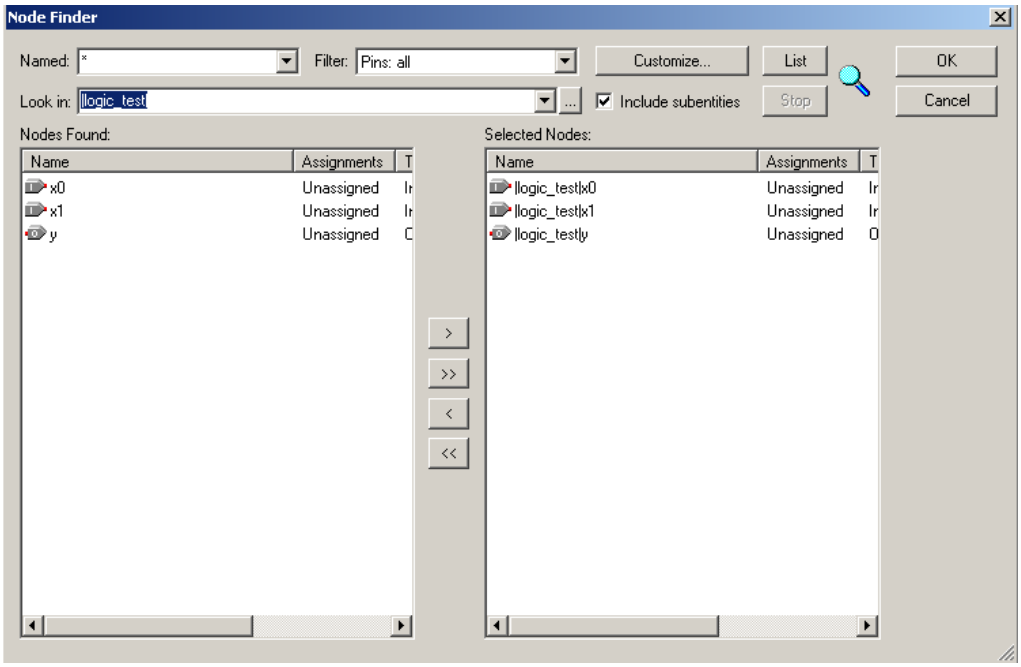


Рис. 1.12 — Добавление сигналов во временную диаграмму

Прежде чем редактировать сигналы необходимо настроить временную шкалу (сетку). Для этого выберите пункт Grid Size меню Edit. Также задайте длительность симуляции, выбрав пункт End Time также из меню Edit (рис. 1.13).

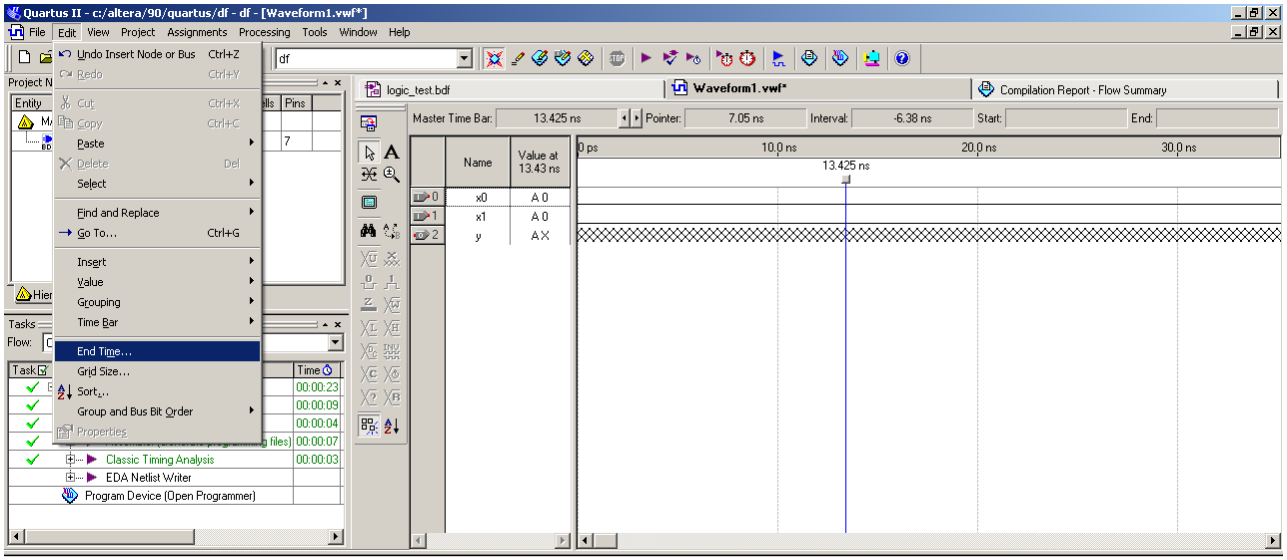


Рис. 1.13 — Задание длительности симуляции

Редактировать входные сигналы проекта можно с помощью кнопок панели инструментов (рис. 1.14). Назначение основных инструментов панели приведено в таблице 1. Для применения инструмента редактирования необходимо выбрать строку сигнала и курсором мыши выбрать временной интервал.

Рис. 1.14 — Редактирование входных сигналов

Таблица 1.2 — Назначение инструментов

Пиктограмма	Инструмент	Назначение
	Waveform Editing Tool	Рисование формы сигнала курсором мыши
	Zoom Tool	Изменение временного масштаба графика (ЛКМ — увеличит масштаб; ПКМ — уменьшить масштаб)
	Forcing Low (0), Forcing High (1)	Установить сигнал в логический 0 либо 1 на выделенном интервале времени
	Invert	Инвертировать значение сигнала
	Overwrite Clock	Сформировать сигнал в виде тактовых импульсов
	Count Value (C)	Сформировать сигнал в виде значений счетчика
	Forcing Unknown (x) High Impedance (Z)	Установить сигнал в неизвестное (x) либо в высокоимпедансное состояние (Z)
	Don't Care (DC)	Установить сигнал в безразличное состояние
	Arbitrary Value	Установить произвольное значение сигнала
	Random Value (R)	Установить случайное значение сигнала

Изобразите вид входных сигналов с помощью инструмента Waveform Editing Tool (рис. 1.15).

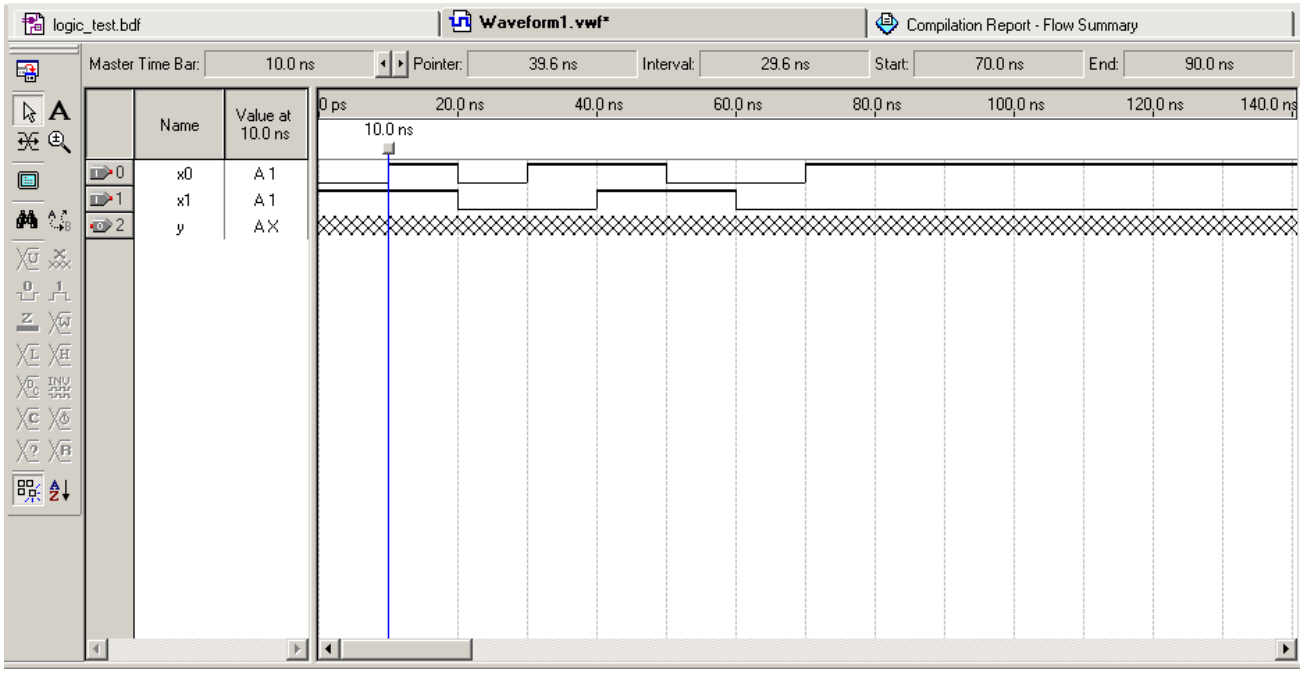


Рис. 1.15 — Формирование вида сигналов

Задайте тип симуляции, выбрав пункт Settings меню Assignment (рис. 1.16).

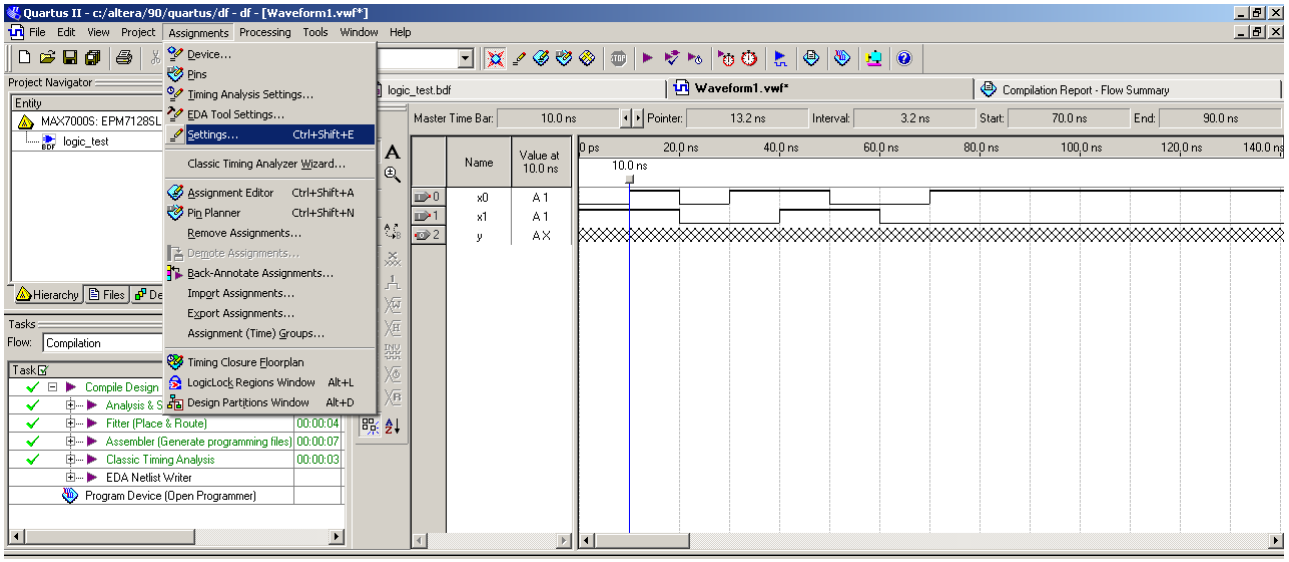


Рис. 1.16 — Выбор типа симуляции

В открывшемся диалоговом окне Settings в поле Category выберите раздел Simulator Settings и в выпадающем списке Simulation mode выберите режим Functional, а в поле Simulation input задайте имя файла симуляции .vwf (рис. 1.17), после этого нажмите кнопку OK.

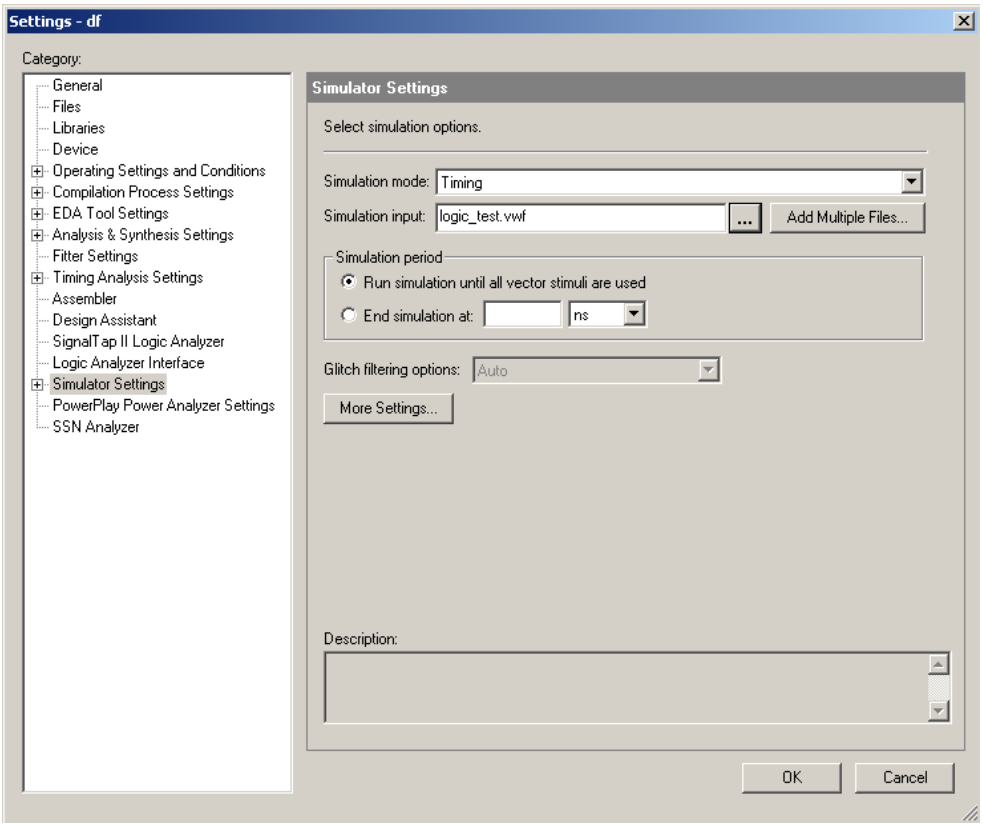


Рис. 1.17 — Выбор параметров симуляции

Запустите моделирование, выбрав пункт Start Simulation меню Processing, либо щелкнув по пиктограмме  (рис. 1.18).

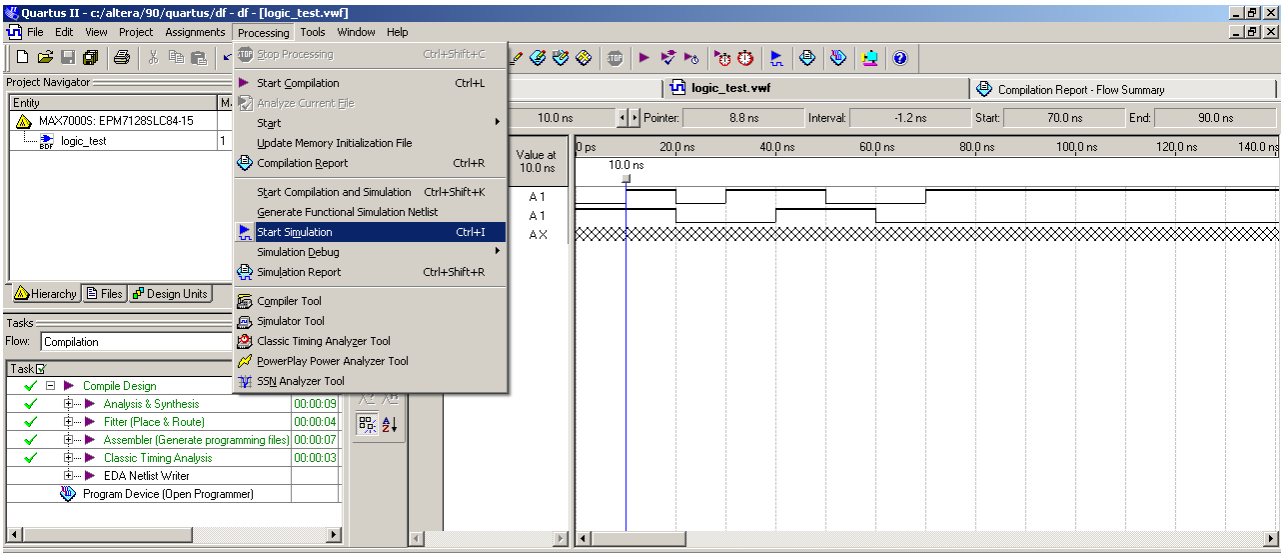


Рис. 1.18 — Запуск симуляции

Результат моделирования показан на рис. 1.19.

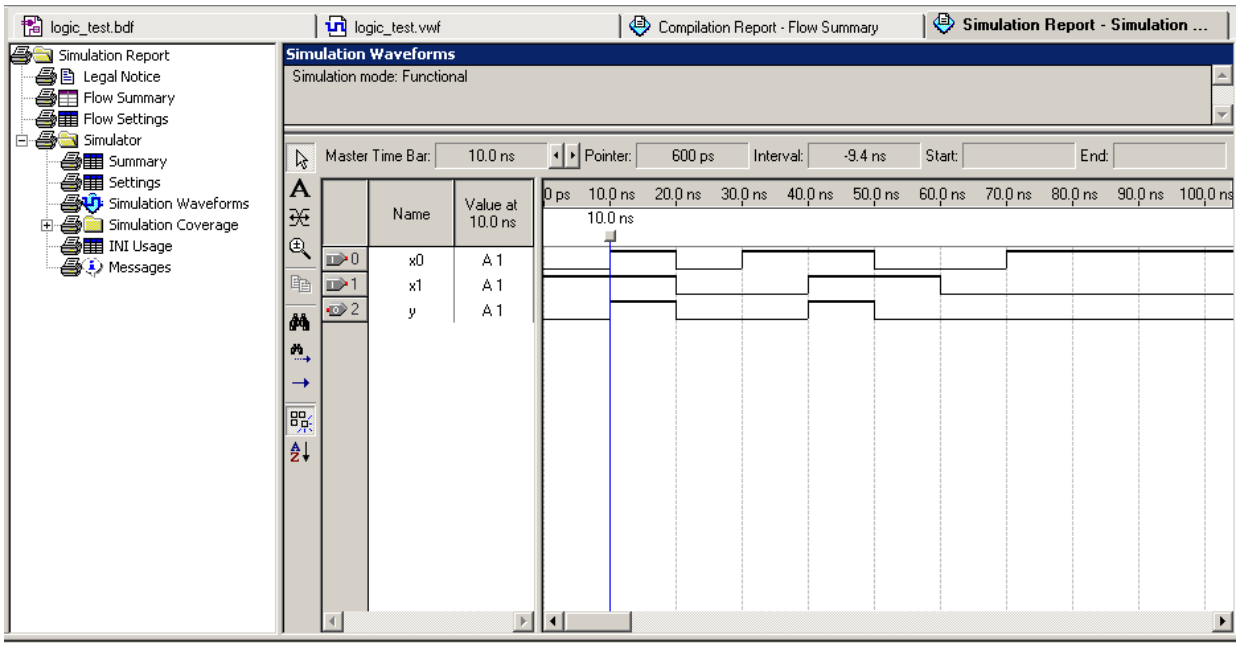


Рис. 1.19 — Результат симуляции

1.2.4. Конфигурирование ПЛИС

Для загрузки проекта на макет и конфигурирования микросхемы ПЛИС в соответствии с разработанной схемой необходимо установить соответствие между входными и выходными портами схемы и выводами микросхемы. Для этого выберите пункт Pins меню Assignments (рис. 1.20).

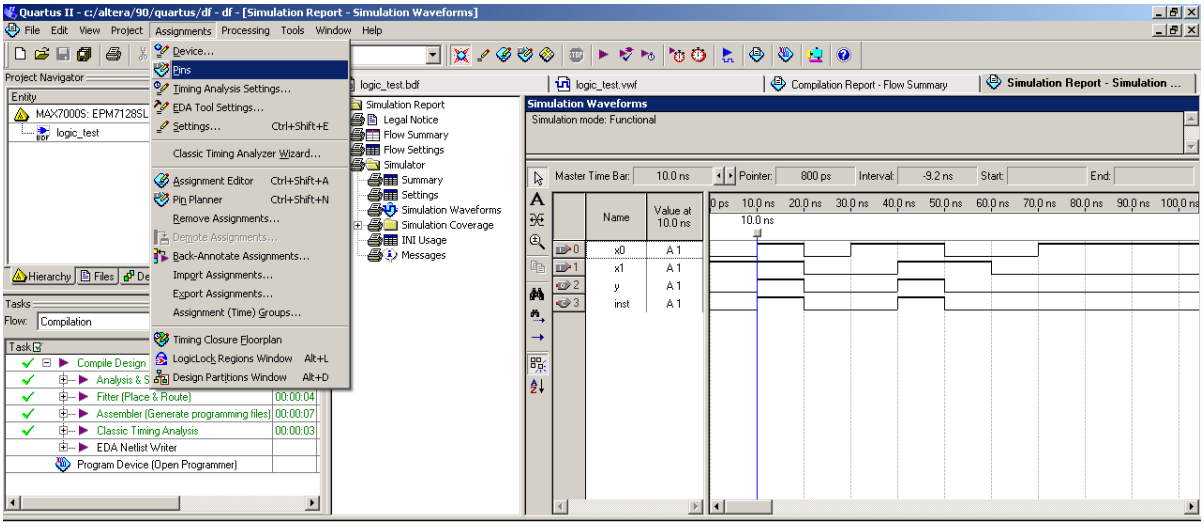


Рис. 1.20 — Конфигурирование ПЛИС

В открывшемся диалоговом окне Pin Planner (рис. 1.21) содержится общий вид микросхемы с указанием расположения и типа выводов. На панели All pins приведена таблица, в которой в столбце Node Name перечислены выводы конфигурационного интерфейса JTAG (сигналы TCK, TDI, TDO, TMS), а также

порты пользовательского проекта. В столбце Location в соответствии с таблицей 2 необходимо назначить выводы микросхемы, соответствующие портам проекта. После этого окно Pin Planner можно закрыть.

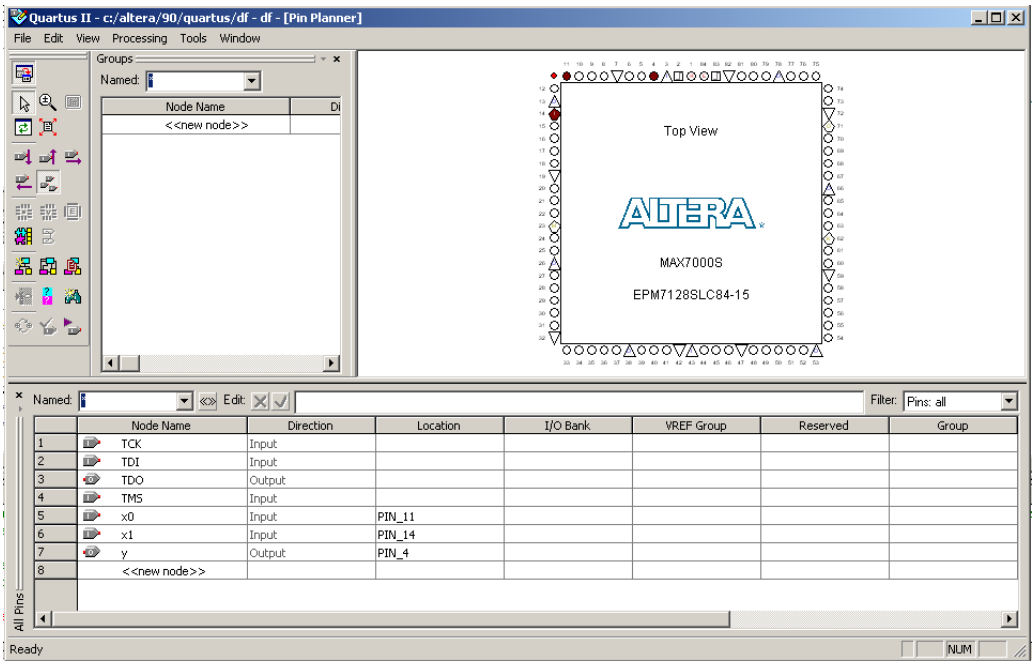


Рис. 1.21 — Выбор пинов ПЛИС

Далее выберем способ конфигурирования неиспользуемых в проекте выводов микросхемы. Для этого выберите пункт Device меню Assignments, далее в открывшемся диалоговом окне (рис. 1.22) нажмите кнопку Device and Pin Options… .

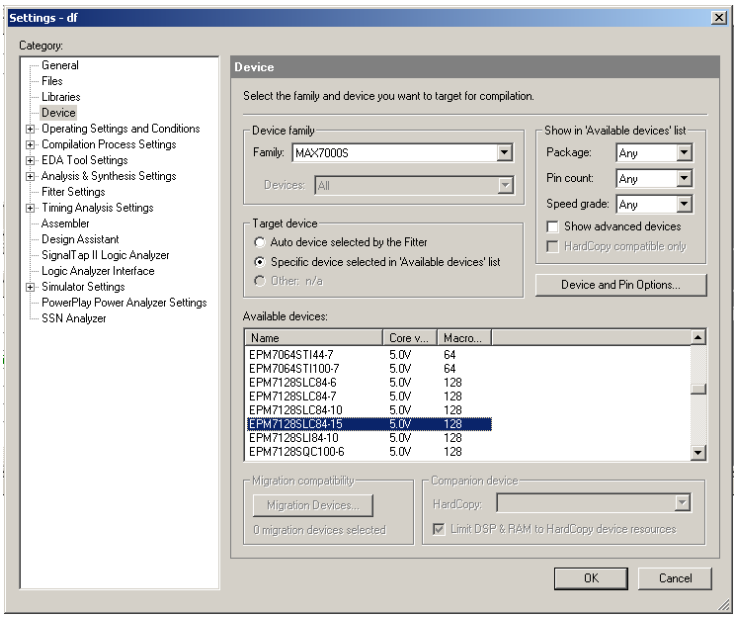


Рис. 1.22 — Выбор устройства

На вкладке Unused Pins в выпадающем списке Reserve all unused pins выберите пункт As input tri-stated (рис. 1.23) и нажмите кнопку OK. Эта настройка настроит все неиспользуемые в проекте выводы микросхемы как входы.

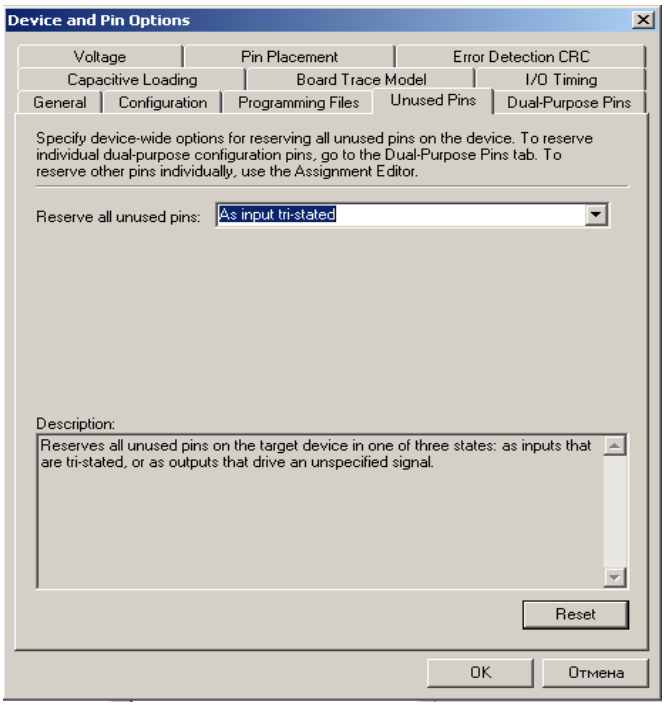


Рис. 1.23 — Назначение неиспользуемых выводов

Выполните компиляцию проекта, щелкнув по пиктограмме  .

В случае успешной компиляции запустите окно программатора (рис. 1.24), выбрав пункт меню Programmer меню Processing. Нажмите кнопку Hardware Setup…, далее кнопку Add Hardware, и затем кнопку OK (рис. 1.25).

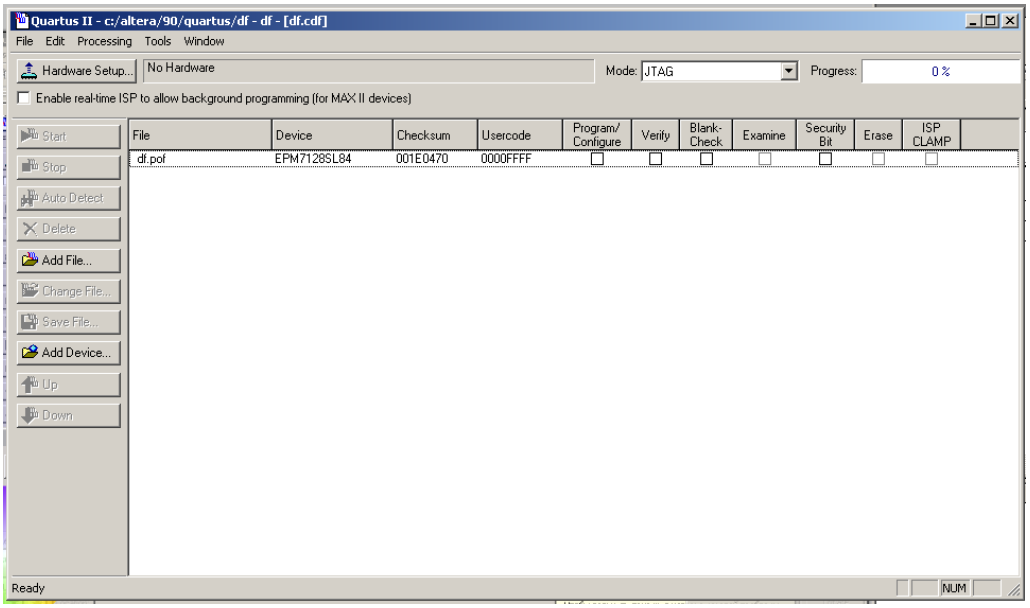


Рис. 1.24 — Окно программатора

После этого в окне программатора установите галочку у пункта Program/Configure (рис. 1.25) и нажмите кнопку Start Start. Статус загрузки отображается в поле Progress в правом верхнем углу окна программатора.

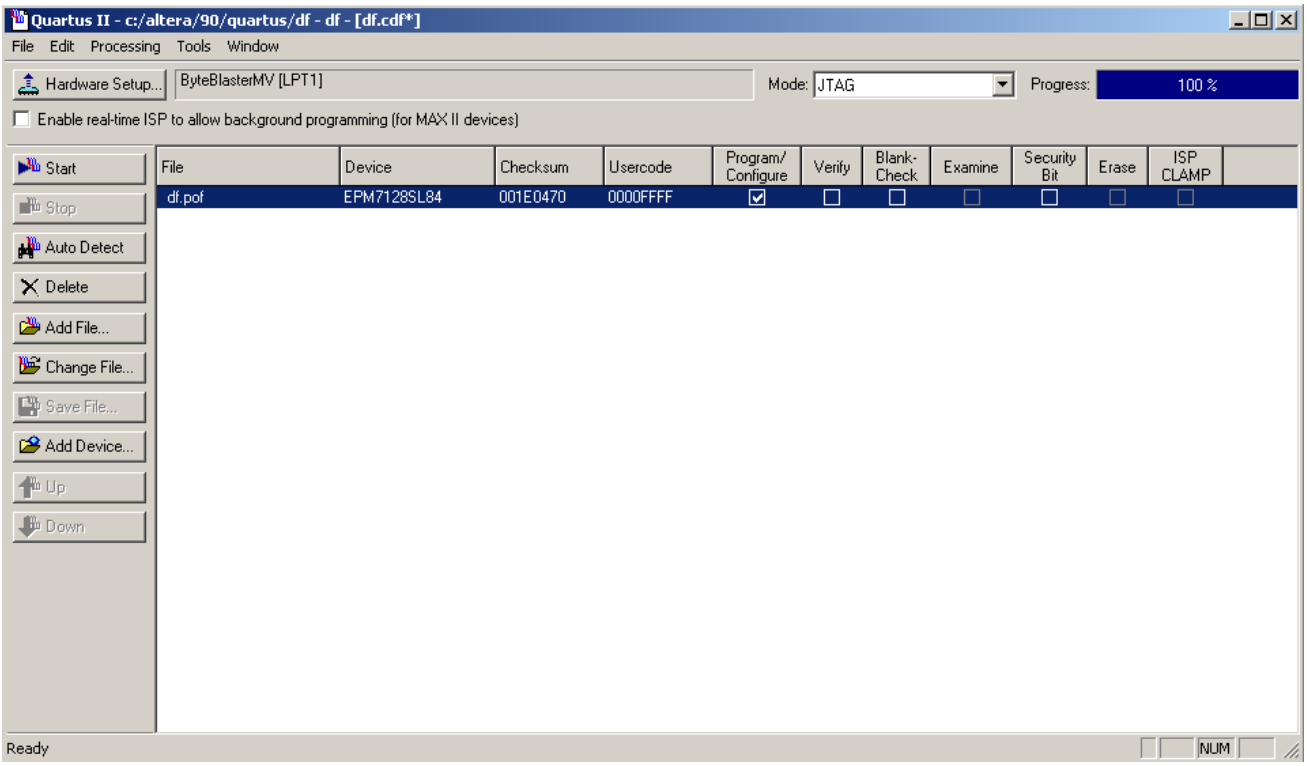


Рис. 1.25 — Установка опции программирования

1.3. Описание лабораторной установки

1.3.1. Расположение и назначение элементов стенда

Лабораторный стенд представляет собой функционально законченное устройство, содержащее в своем составе собственно программируемую логическую интегральную микросхему семейства MAX7000S типа EPM7128SLC84-15, программатор и периферийные устройства. Внешний вид стенда и расположение его составных частей показан на рис. 1.26.

Световые индикаторы HL0 — HL3 представляют собой семисегментные (A, B, C, D, E, F, G) светодиодные индикаторы плюс децимальная точка P. Для зажигания соответствующего сегмента на назначенном выводе ПЛИС необходимо сформировать логическую единицу.

Для зажигания линейки отдельных светодиодов LED1 — LED8, на назначенном выводе ПЛИС необходимо сформировать логическую единицу.

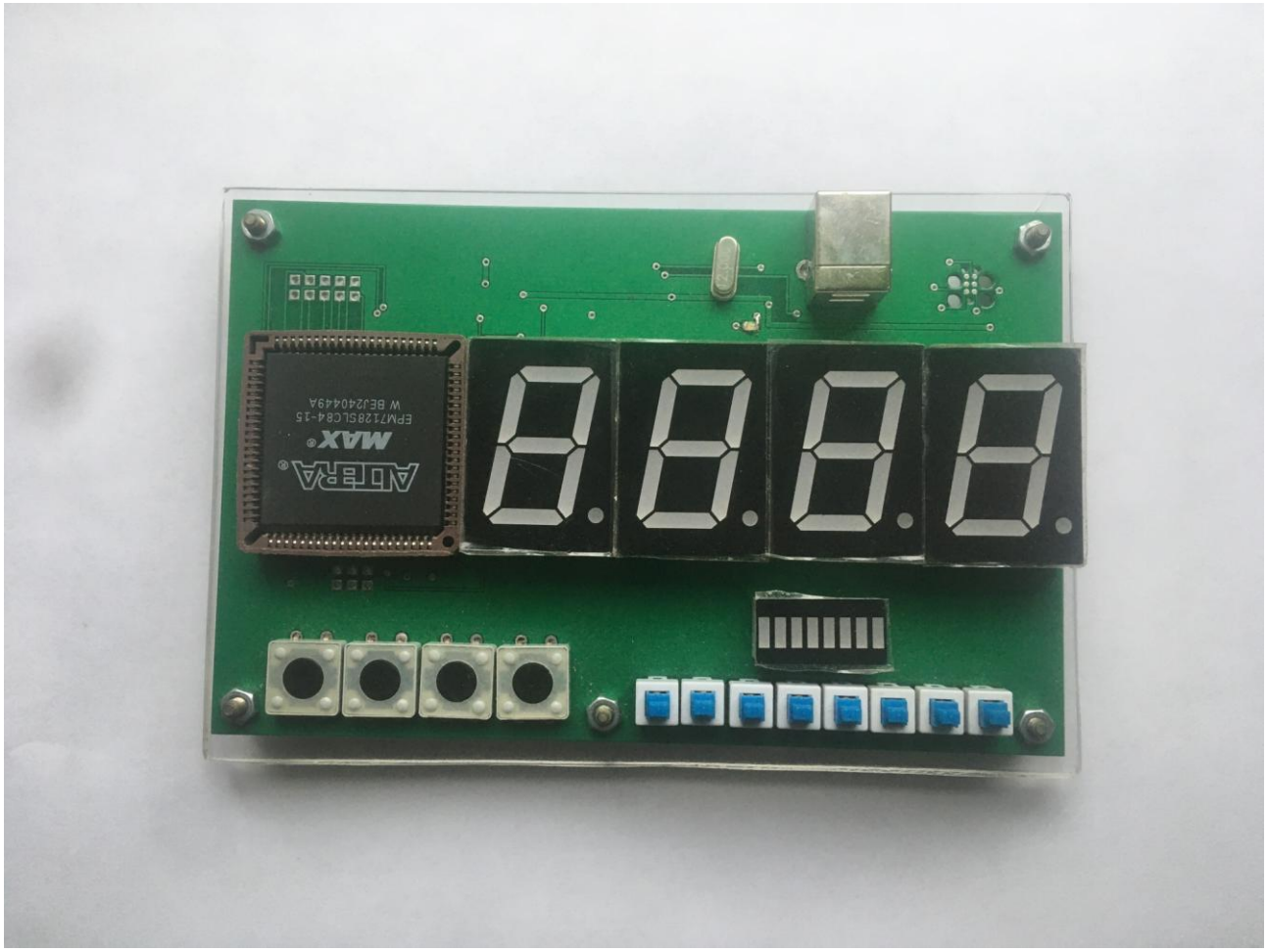


Рисунок 1.26 — Внешний вид стенда

Кнопки SA1 — SA4 позволяют при ее нажатии формировать на соответствующем входе ПЛИС логический ноль (стационарное состояние — логическая единица).

Переключатели SW1 — SW8 позволяют устанавливать на соответствующих входах ПЛИС долговременно логический ноль или логическую единицу.

Тактовый генератор формирует на входе CLOCK ПЛИС последовательность импульсов с частотой следования около 1 кГц.

Программирование ПЛИС осуществляется посредством программатора, подключенного к порту USB компьютера. Через этот же порт осуществляется питание макета.

Назначение выводов ПЛИС в стенде приведены в таблице 1.3.

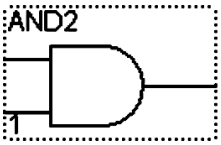
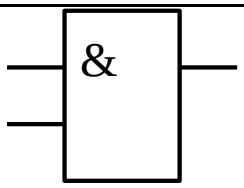
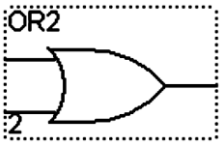
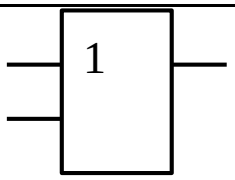
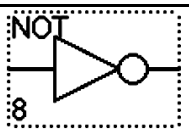
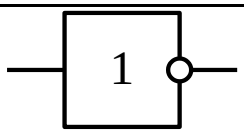
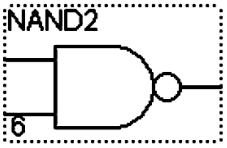
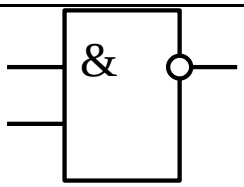
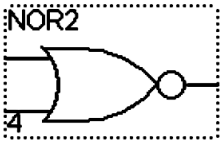
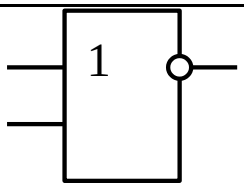
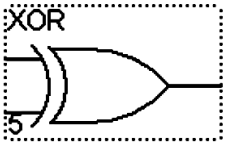
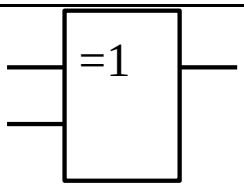
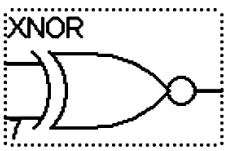
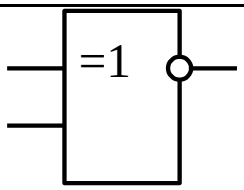
Таблица 1.3 — Назначение выводов ПЛИС в стенде

Номер вывода	Назначение	Номер вывода	Назначение
54	Кнопка_3	31	Сегмент_A1
57	Кнопка _2	28	Сегмент_B1
56	Кнопка _1	16	Сегмент_C1
58	Кнопка _0	15	Сегмент_D1
79	Светодиод_0	12	Сегмент_E1
80	Светодиод _1	30	Сегмент_F1
77	Светодиод _2	34	Сегмент_G1
75	Светодиод _3	17	Сегмент_DP1
76	Светодиод _4		
74	Светодиод _5	36	Сегмент_A2
73	Светодиод _6	33	Сегмент_B2
70	Светодиод _7	10	Сегмент_C2
68	Переключатель_0	9	Сегмент_D2
69	Переключатель _1	8	Сегмент_E2
67	Переключатель _2	35	Сегмент_F2
64	Переключатель _3	37	Сегмент_G2
65	Переключатель _4	11	Сегмент_DP2
63	Переключатель _5		
60	Переключатель _6	39	Сегмент_A3
61	Переключатель _7	40	Сегмент_B3
24	Сегмент_A0	5	Сегмент_C3
25	Сегмент_B0	4	Сегмент_D3
20	Сегмент_C0	81	Сегмент_E3
21	Сегмент_D0	41	Сегмент_F3
18	Сегмент_E0	44	Сегмент_G3
27	Сегмент_F0	6	Сегмент_DP3
29	Сегмент_G0		
22	Сегмент_DP0	83	CLK

1.3.2. Обозначения основных логических элементов в САПР

Соответствие обозначений основных логических элементов, используемых в САПР «QUARTUS II» и принятых по ГОСТ 2.721-74 ЕСКД приведено в табл. 1.5.

Таблица 1.4 — Соответствие обозначений основных логических элементов

Название элемента	Графическое обозначение функции в QUARTUS II	Стандартное графическое обозначение функции	Логическая функция
2И (Конъюнктор)			$y = X1 \cdot X2 = \overline{\overline{X1} + \overline{X2}}$
2ИЛИ (Дизъюнктор)			$Y = X1 + X2 = \overline{\overline{X1} \cdot \overline{X2}}$
НЕ (Инвертор)			$Y = \overline{X}$
2И-НЕ			$Y = \overline{X1 \cdot X2} = \overline{X1} + \overline{X2}$
2ИЛИ-НЕ			$Y = \overline{X1 + X2} = \overline{X1} \cdot \overline{X2}$
Исключающее ИЛИ			$Y = X1 \cdot \overline{X2} + \overline{X1} \cdot X2$
Исключающее ИЛИ-НЕ			$Y = X1 \cdot X2 + \overline{X1} \cdot \overline{X2}$

1.4. Порядок выполнения работы

1.4.1. Создайте в рабочей папке свою директорию.

1.4.2. Откройте ПП QUARTUS II. Создайте проект, воспользовавшись мастером New Project Wizard.

1.4.3. Выберете семейство и наименование микросхемы ПЛИС, в которую будет загружен проект.

1.4.4. Создать новый файл проекта. Сохраните файл схемы с расширением .bdf, выбрав пункт Save As из меню File

1.4.4. В поле схемного редактора нажать правую кнопку мыши и в выпадающем меню выбрать пункт Insert → Symbol. В графе Symbol Name наберите INPUT (без кавычек) и нажмите ОК. На экране появится одноименный элемент. Перемещать элемент можно традиционным для Windows методом.

1.4.5. Введите элемент OR2 (элемент 2ИЛИ). Обратите внимание, что в верхнем левом угле элемента указано его имя. Расположите элемент OR2 напротив предыдущего элемента, перетаскивая его курсором мыши. Наведите курсор мыши на правый конец элемента INPUT, его изображение изменится на «плюс». Нажав и удерживая клавишу мыши проведите черту от выхода INPUT к входу элемента OR2. Линию связи можно сгибать, но только один раз.

1.4.6. Элемент INPUT имеет служебное имя, что указано сбоку от элемента. Замените его на «IN1» дважды щелкнув мышью по данному полю и введя необходимые символы.

1.4.7. Пользуясь полученными навыками, дорисуйте схему, изображенную на рис. 1.27.

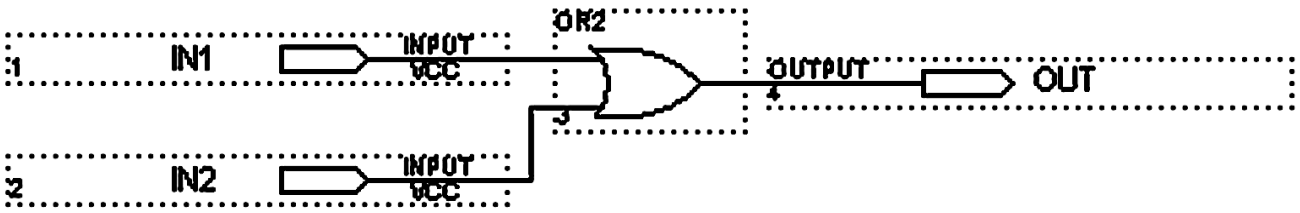


Рисунок 1.27 — Элемент 2 ИЛИ

1.4.8. Командой Save As (меню File) запомните созданный файл с именем вашим именем.

1.4.9. Откройте компилятор (пункт Processing меню QUARTUS II) и откомпилируйте проект, нажав на Start Compilation. Если компиляция завершилась с ошибками, исправьте их, сверяясь с схемой и перекомпилируйте проект.

1.4.10. Создайте специальный файл, содержащий список соединений логических элементов.

1.4.11. Создайте файл описания формы входных сигналов (файл симуляции).

1.4.12. Добавьте сигналы в файл симуляции

1.4.13. Настроить временную шкалу (сетку) и задайте длительность симуляции.

1.4.14. Изобразите вид входных сигналов с помощью инструмента  Waveform Editing Tool (рис. 1.15)

1.4.15. Задайте тип симуляции, выбрав пункт Settings меню Assignment (рис. 1.16) и (рис. 1.16) после этого нажмите кнопку OK.

1.5. Содержание отчета

1.5.1. Цель работы.

1.5.2. Внешний вид макета.

1.5.3. Скриншот главного меню САПР.

1.5.4. Скриншоты синтезируемых элементов и всех ступеней проектирования.

1.5.5. Выводы

1.6. Контрольные вопросы для защиты

1.6.1. Как создать проект?

1.6.2. Поясните функции редактора графики и символов.

1.6.3. Поясните функции компилятора.

1.6.4. Поясните функции симулятора.

1.6.5. Поясните функции анализатора временных диаграмм.

1.6.6. Поясните функции программатора.

1.6.7. Как вводятся элементы на рабочий стол?

1.6.8. Как устанавливаются связи между входами и выходами элементов на рабочем столе?

1.6.9. Как оцениваюся временные задержки сигналов?

1.6.10. Как изменитья временной масштаб сигналов?

ЛАБОРАТОРНАЯ РАБОТА № 2
ИССЛЕДОВАНИЕ БАЗОВЫХ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ

2.1. Цель работы

Целью работы является изучение основных законов двоичной алгебры и базовых логических элементов. Проведение моделирования логических элементов.

2.2. Теоретические сведения

2.2.1. Основные представления о двоичной алгебре и базовых элементах

Все цифровые устройства построены на принципе многократного повторения относительно простых базовых логических схем. Математическим инструментом такого построения служит Булева алгебра, называемая так же алгеброй логики.

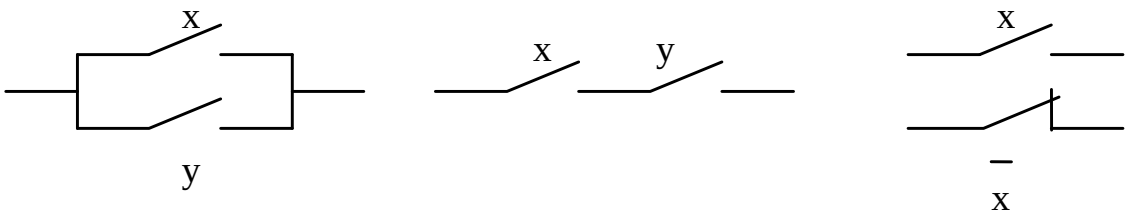
В отличие от переменной в обычной алгебре логическая переменная имеет только два значения, которые обычно называются логическим нулем и логической единицей. В качестве обозначений используют «0» и «1» или просто 0 и 1.

Существует три основных операции между логическими переменными, представленные в табл. 2.1.

Таблица 2.1 — Основные логические операции

Операция	Обозначение
Конъюнкция (логическое умножение)	$y = x_1 \wedge x_2 = x_1 \cdot x_2 = x_1 x_2$
Дизъюнкция (логическое сложение)	$y = x_1 \vee x_2 = x_1 + x_2$
Инверсия (логическое отрицание)	$y = \overline{x}$

Для пояснения основных операций алгебры логики на рис. 2.1 представлена их реализация с помощью контактных цепей.



а) б) в)

Рис. 2.1 — Реализация базовых логических элементов
с помощью контактных цепей

Операция дизъюнкции реализуется параллельным соединением контактов x и y (рис. 2.1, а), операция конъюнкции — последовательным соединением контактов x и y (рис. 2.1, б), а операция отрицания — использованием нормально замкнутого контакта (рис. 2.1, в), если считать, что x — нормально разомкнутый контакт.

Применительно к логическим операциям существуют правила и законы булевой алгебры, приведенные ниже.

Коммутативный закон

$x_1 x_2 = x_2 x_1$ $\qquad$ $x_1 + x_2 = x_2 + x_1$

Ассоциативный закон

$x_1\ x_2 x_3\ =\ x_1 x_2\ \ x_3$ $\qquad$ $x_1 +\ x_2 + x_3\ =\ x_1 + x_2\ \ + x_3$

Дистрибутивный закон

$x_1\ \ x_2 + x_3\ = x_1 x_2 + x_1 x_3$ $\qquad$ $x_1 + x_2 x_3 =\ x_1 + x_2\quad x_1 + x_3$

Правило склеивания

$x_1\ \ x_1 + x_2\ = x_1$ $\qquad$ $x_1 + x_1 x_2 = x_1$

Правило повторения

$xx = x$ $\qquad$ $x + x = x$

Правило отрицания

$x \cdot \overline{x} = 0$ $\qquad$ $x + \overline{x} = 1$

Правило двойного отрицания

$$\overline{\overline{x}} = x$$

Теорема де Моргана

$$\overline{x_1 x_2} = \overline{x_1} + \overline{x_2} \qquad \overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2}$$

Операции с 0 и 1

$$x \cdot 1 = x \qquad x + 0 = x$$

$$x \cdot 0 = 0 \qquad x + 1 = 1$$

$$\overline{0} = 1 \qquad \overline{1} = 0$$

С помощью перечисленных законов можно вычислить результаты конъюнкции и дизъюнкции для всех возможных значений переменных x_1 и x_2. Таблица, включающая результаты вычисления по заданному закону называется таблицей соответствия или таблицей истинности (см. табл. 2.2 и табл. 2.3).

Таблица 2.2 — Таблица истинности для логического умножения (конъюнкции)

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Таблица 2.3 — Таблица истинности для логического сложения (дизъюнкции)

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Из табл. 2.2 следует, что y только тогда равен 1, когда x_1 и x_2 равны 1.

На этом основании операция конъюнкции называется также функцией И. При дизъюнкции двух переменных y равен 1 тогда, когда x_1 или x_2 равны 1, поэтому операцию дизъюнкции называют также функцией ИЛИ. Обе эти функции можно распространить на сколь угодно большое число переменных.

2.2.2. Составление логических функций

Наиболее простым способом задания свойств схемы является таблица истинности. Для реализации схемы требуется найти такую логическую функцию, которая соответствует этой таблице. Далее эту функцию преобразуют в простейшую форму, которую потом реализуют при помощи комбинации базовых логических схем. При условии записи функции в дизъюнктивной нормальной форме выполняется следующая последовательность действий:

В таблице истинности выделяют строки, в которых выходная переменная имеет значение 1;

Для каждой такой строки составляют конъюнкцию всех входных переменных, причем сомножитель x_i записывают в том случае, если рассматриваемая переменная имеет значение 1, в противном случае записывают $\overline{x_i}$. Т.е. составляется столько произведений, сколько имеется строк с $y=1$;

Записывается логическая сумма всех полученных ранее произведений.

Рассмотрим последовательность действий на примере произвольной таблицы истинности (см. табл. 2.4).

Таблица 2.4 — Таблица истинности произвольного элемента

x_1	x_2	x_3	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Переменная y принимает значение «1» в 3, 5 и 7 строках. Составим конъюнкцию для этих строк.

$$K_3 = \overline{x_1} x_2 \overline{x_3}$$
$$K_5 = x_1 \overline{x_2} \overline{x_3}$$
$$K_7 = x_1 x_2 \overline{x_3}$$

Результирующую функцию запишем как логическую сумму произведений:

$$y = K_3 + K_5 + K_7 = \overline{x_1} x_2 \overline{x_3} + x_1 \overline{x_2} \overline{x_3} + x_1 x_2 \overline{x_3}$$

Применяя законы двоичной алгебры, приведенные ранее, получим конечную запись функции:

$$y = x_1 + x_2 \cdot \overline{x_3}$$

Полученная функция легко реализуется при помощи базовых логических элементов.

2.3. Порядок выполнения работы

2.3.1. По заданной таблице истинности (см. таблицу 2.5) получить результирующую логическую функцию. Столбец y_N необходимо выбрать в соответствии с номером бригады.

Таблица 2.5 — Таблица истинности индивидуальных заданий

x_1	x_2	x_3	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
0	0	0	1	0	0	0	0	0	1	0	1	0
0	0	1	0	1	0	1	0	1	0	0	1	0
0	1	0	0	0	1	0	1	0	0	1	1	1
0	1	1	1	0	0	1	0	1	0	1	0	0
1	0	0	0	0	1	0	0	0	1	1	0	1
1	0	1	0	1	0	1	1	0	0	0	0	0
1	1	0	1	0	1	0	0	1	1	0	0	1
1	1	1	0	1	0	0	1	0	0	0	0	0

2.3.2. Составить логическую схему для полученной функции.

2.3.3. Построить данную схему в «MAX +PLUS II», назначив соответственно три входных сигнала и один выходной.

2.3.4. Произвести компиляцию проекта

2.3.5. Провести симуляцию работы схемы при всех сочетаниях входных переменных. Для чего необходимо сформировать три последовательности импульсов входных сигналов типа меандр с периодом последующего сигнала в два раза большим, чем период предыдущего. Это можно сделать вручную, выделяя соответствующие секторы и формируя на заданных временных интервалах лог. 0 или лог. 1. Это можно сделать автоматически, выбрав режим задания тактовых последовательностей для одного или группы узлов (кнопки «1» или «2» рис. 2.2) .

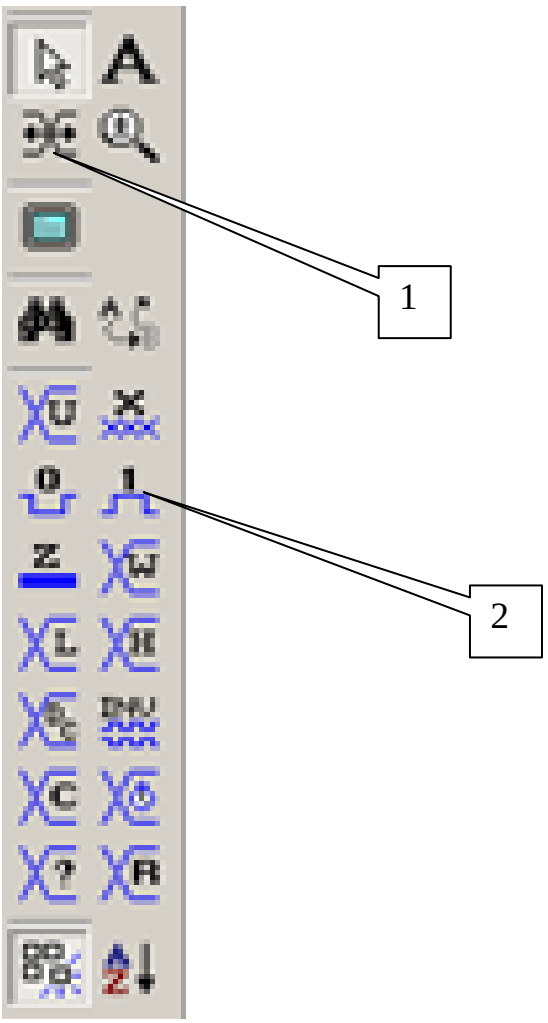


Рисунок 2.2 — Задание временных последовательностей

Выбирая второй способ задания тактовых импульсов, можно формировать последовательности импульсов с периодами, кратными периоду исходной

последовательности (2, 3, 4 и т.д.). Для представления во временной области последовательного перебора всех комбинаций входных переменных, периоды последовательностей должны быть кратны степени числа 2.

По умолчанию среда разработки MAX+PLUS II обеспечивает анализ работы схемы на временном интервале 0 — 1 us с шагом 100 ns, что позволяет пронаблюдать всего 10 возможных комбинаций входных переменных или 10 возможных состояний схемы. Для каждого из входных сигналов можно задавать конечное время анализа в отдельности, воспользовавшись меню File, опцией End Time. По умолчанию End Time = 1us. Причем установку времен анализа необходимо начинать с наименьшего. Например, (см. рис.2.3) время изменения сигнала по входу «3» — 2.2 us. (устанавливается первым), по входу «2» — 5.2 us (вторым), по входу «1» — 10 us. Для примера четвертая временная диаграмма отображает выходной сигнал схемы 3И при подаче на ее входы описанных временных последовательностей.

2.4. Содержание отчета

2.4.1. Цель работы.
2.4.2. Заданная таблица истинности.
2.4.3. Составленная формула.
2.4.4. Полученная логическая схема.
2.4.5. Скриншоты временных диаграмм работы схемы.
2.4.6. Выводы.

2.5. Контрольные вопросы для защиты

2.5.1. Запишите таблицу истинности для элемента 2И.
2.5.2. Запишите таблицу истинности для элемента 2ИЛИ.
2.5.3. Запишите таблицу истинности для элемента 2И-НЕ.
2.5.4. Запишите таблицу истинности для элемента 2ИЛИ-НЕ.

2.5.5. Запишите таблицу истинности для элемента ИСКЛЮЧАЮЩЕЕ ИЛИ.

2.5.6. Запишите таблицу истинности для элемента ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ.

2.5.7. Сформулируйте коммутативный закон.

2.5.8. Сформулируйте правило повторения.

2.5.9. Сформулируйте правило отрицания.

2.5.10. Опишите операции с 0 и 1.

ЛАБОРАТОРНАЯ РАБОТА № 3
ИССЛЕДОВАНИЕ ДЕШИФРАТОРОВ КОДОВ

3.1. Цель работы:

Целью работы является изучение принципов построения дешифраторов на примере десятичного и семисегментного дешифраторов и принципов построения логической схемы по произвольной таблице истинности

3.2. Теоретические сведения

Дешифраторы позволяют преобразовывать одни виды бинарных кодов в другие. Например, преобразовывать позиционный двоичный код в линейный восьмеричный или шестнадцатеричный. Преобразования производятся по правилам, описанным в таблице истинности. Дешифраторы выпускаются в виде отдельных микросхем или используются в составе других микросхем. В настоящее время дешифраторы используются как составная часть других микросхем, таких как мультиплексоры, демультиплексоры, ПЗУ или ОЗУ.

3.2.1. Двоично-десятичный дешифратор

Рассмотрим пример построения дешифратора из двоичного кода в десятичный. Десятичный код обычно отображается одним битом на одну десятичную цифру. В десятичном коде десять цифр, поэтому для отображения одного десятичного разряда требуется десять выходов дешифратора. Сигналы с этих выходов можно подать на десятичный индикатор. Таблица истинности такого дешифратора приведена в таблице 3.1.

Таблица 3.1 — Таблица истинности двоично-десятичного дешифратора

Входы				Выходы									
X8	X4	X2	X1	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

Схема, соответствующая данной таблице истинности, приведена на рис.3.1.

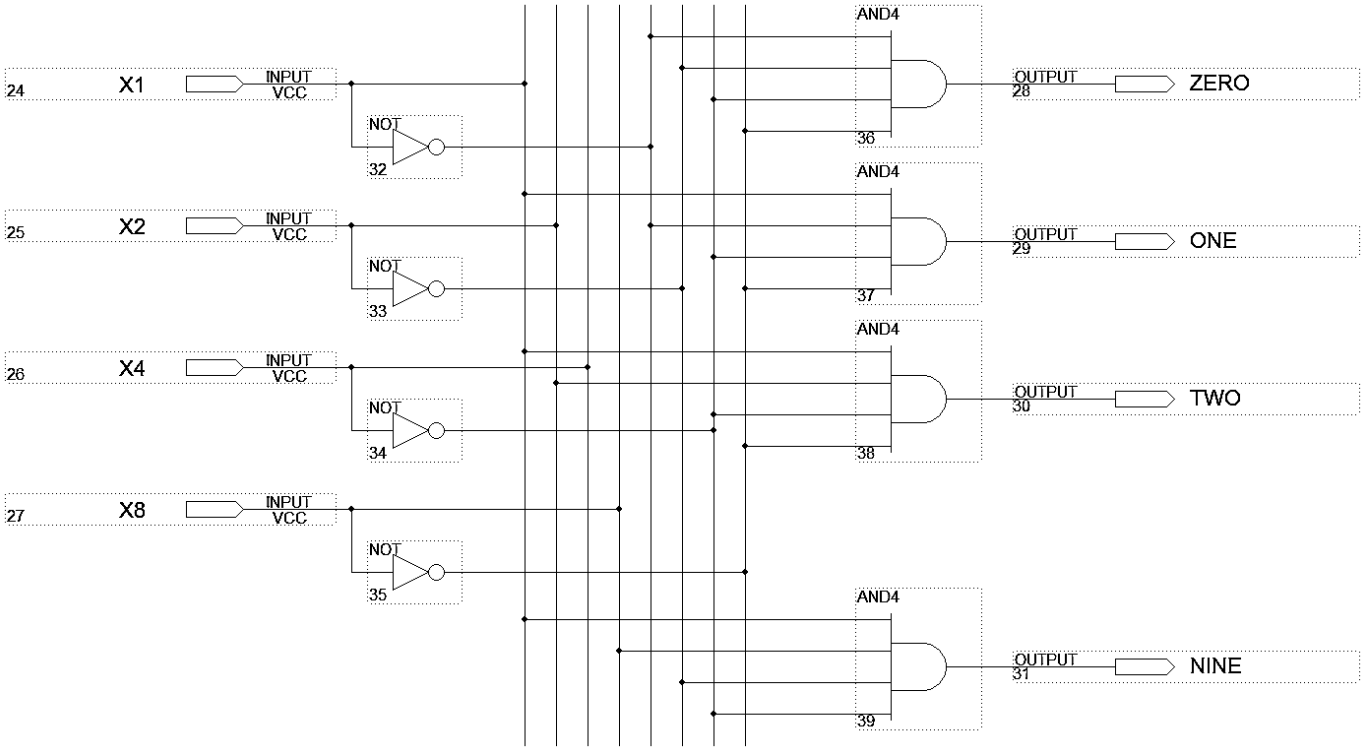


Рисунок 3.1 — Схема двоично-десятичного дешифратора

3.2.2. Семисегментный дешифратор

Для отображения десятичных и шестнадцатеричных цифр часто используется семисегментный индикатор. Изображение семисегментного индикатора и название его сегментов приведено на рис. 3.2.

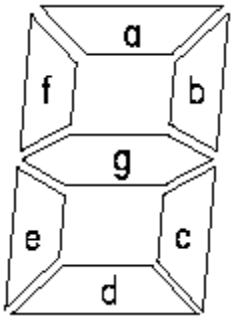


Рисунок 3.2 — Семисегментный индикатор

Для отображения на таком индикаторе цифры ноль достаточно зажечь сегменты a, b, c, d, e, f. Для отображения цифры 1 зажигают сегменты b и c. Точно таким же образом можно получить изображения всех остальных десятичных или шестнадцатеричных цифр. Все комбинации таких изображений получили название семисегментного кода. Таблица истинности семисегментного дешифратора приведена ниже.

Таблица 3.2 — Таблица истинности семисегментного дешифратора

Цифра	X8	X4	X2	X1	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

3.2.3. Составление схемы семисегментного дешифратора

Первым шагом составления схемы является получение логических функций для каждого из выходов a, b, c …g. При анализе таблицы истинности нетрудно заметить, что число нулей в ней гораздо больше числа единиц. Таким образом, составлять функцию можно для нулевых ячеек таблицы, инвертируя результат. Так, например, для выхода a можно записать выражение:

$$a = \overline{\overline{X_1} \cdot X_2 \cdot \overline{X_4} \cdot X_8 + \overline{X_1} \cdot \overline{X_2} \cdot \overline{X_4} \cdot X_8}$$

Аналогично выполняется построение функций для остальных выходов дешифратора.

По полученной функции можно построить схему дешифратора сегмента a.

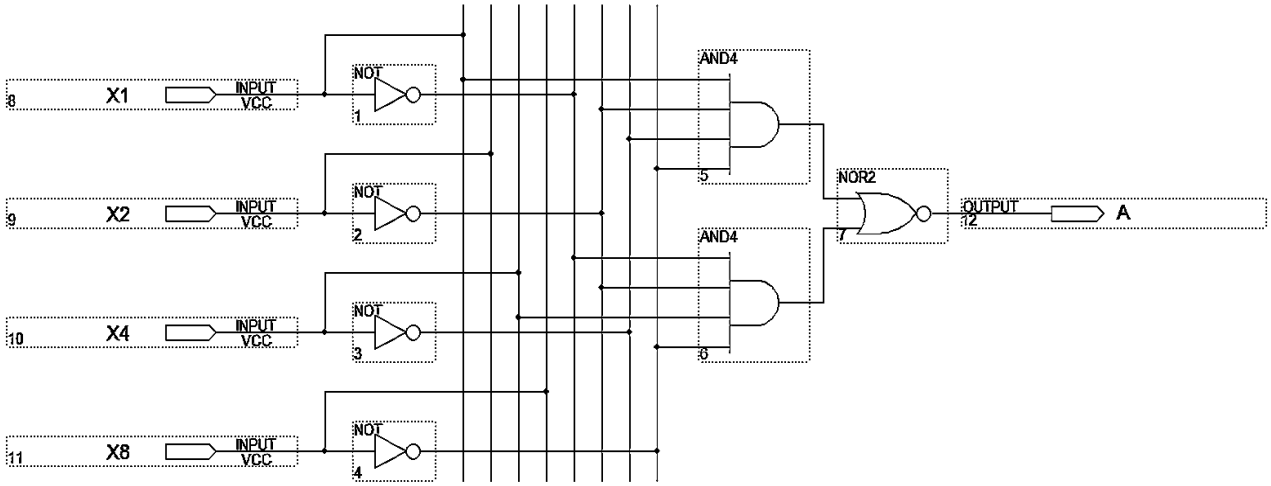


Рисунок 3.3 — Принципиальная схема семисегментного индикатора

3.3. Порядок выполнения работы

3.3.1. По таблице истинности составить логические функции всех выходов.

3.3.2. Составить схему семисегментного дешифратора.

Для студентов, выполняющих работу строго по плану, семисегментный индикатор строится в соответствии с таблицей истинности 3.2.

Для студентов, отстающих от графика выполения лабораторных работ, предусматривается индивидуальное задание, которое выдается преподавателем. Согласно этому заданию таблица 3.2 изменяется. В ней меняются местами две любые отображаемые цифры. Например, при поступлении кода числа 1 должна отображаться цифра 7. И наоборот, при поступлении кода числа 7 должна отображаться цифра 1.

3.3.3. Сохранить схему как макрос.

3.3.4. Выполнить симулирование работы схемы, убедиться, что схема составлена верно и работает.

3.3.5. Предъявить результаты моделирования преподавателю. В том случае, если результаты верные, выполнить программирование макета.

3.3.6. Проверить работу дешифратора, изменяя входную комбинацию при помощи тумблеров на макете.

3.4. Содержание отчета

3.4.1. Цель работы.

3.4.2. Таблица истинности семисегментного дешифратора.

3.4.3. Схема семисегментного дешифратора. Скриншот схемы.

3.4.4. Результаты моделирования работы схемы. Скриншот симуляции.

3.4.5. Выводы.

3.5. Контрольные вопросы для защиты

3.5.1. Поясните функции полного дешифратора.

3.5.2. Поясните функции десятичного дешифратора.

3.5.3. Поясните функции семисегментного дешифратора.

3.5.4. Каким образом можно сохранить схему дешифратора как один элемент?

3.5.5. Какое стандартное число входов может быть у многовходовых элементов И/ ИЛИ?

3.5.6. В каких случаях целесообразно использовать выходные вентили с инвертором?

3.5.7. Сформулируйте ассоциативный закон.

3.5.8. Сформулируйте закон де-Моргана.

3.5.9. Сформулируйте правило склеивания

3.5.10. Сформулируйте дистрибутивный закон.

ЛАБОРАТОРНАЯ РАБОТА № 4
ИССЛЕДОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТНЫХ СХЕМ

4.1. Цель работы

Осуществление синтеза асинхронного десятичного счетчика и закрепление навыков создания и анализа работы цифровых схем с помощью САПР «QUARTUS II». Ознакомление с возможностью замены стандартных цифровых элементов и схем микросхемами программируемой логики.

4.2. Теоретические сведения

4.2.1. Построение счетчика

Счетчиком называется последовательностная схема, циклически переходящая из одного устойчивого состояния в другое под действием входных сигналов. При этом можно получить различный коэффициент пересчета (модуль счета) K_C — количество входных переключающих импульсов, которое нужно подать на вход счетчика, чтобы он вернулся в исходное состояние.

Простейшим счетчиком является триггер. В счетном режиме на его вход нужно подать два счетных импульса, чтобы триггер возвратился в исходное состояние.

Для того чтобы получить произвольный модуль счета K_C необходимо $l = \log_2 K_C$ триггеров.

Если $K_C = 2^m$, где m — целое, то такой счетчик называют двоичным.

Если $K_C \neq 2^m$, то для организации такого счетчика потребуется n триггеров, где n — ближайшее целое большее l.

Состояние счетчика определяется состоянием его триггеров

$$Q_1, Q_2, Q_3, ..., Q_n.$$

Правило работы счетчика, т.е. порядок изменения его состояний, задают в виде соответствующей таблицы. В счетчиках с естественным порядком счета каждый входной переключающий сигнал приводит к изменению числа, зафик-

сированного в счетчиках на единицу. В зависимости от знака этого изменения счетчики различают на суммирующие и вычитающие.

В таблице 4.1 заданы переходы трехразрядного двоичного счетчика из одного состояния в другое при подаче на его вход N запускающих импульсов.

Из таблицы видно, что при поступлении на вход счетчика 8 импульсов, последний возвращается в исходное состояние, т.е.

$$K_C = 2^3 = 8,$$

где 3 — число разрядов (триггеров) счетчика.

Если рассматривать таблицу сверху вниз, то каждый счетный импульс увеличивает содержимое счетчика на единицу — реализуется суммирующий режим работы счетчика. Если эту же таблицу рассматривать снизу вверх — получаем вычитающий режим работы счетчика.

Таблица 4.1 — Таблица переходов двоичного счетчика

N	Q3	Q2	Q1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0
9	0	0	1
10	0	1	0
11	0	1	1

Для пояснения счетного режима работы счетчика на рис.4.1 показана временная диаграмма работы суммирующего двоичного счетчика с модулем счета K_C, равным 8.

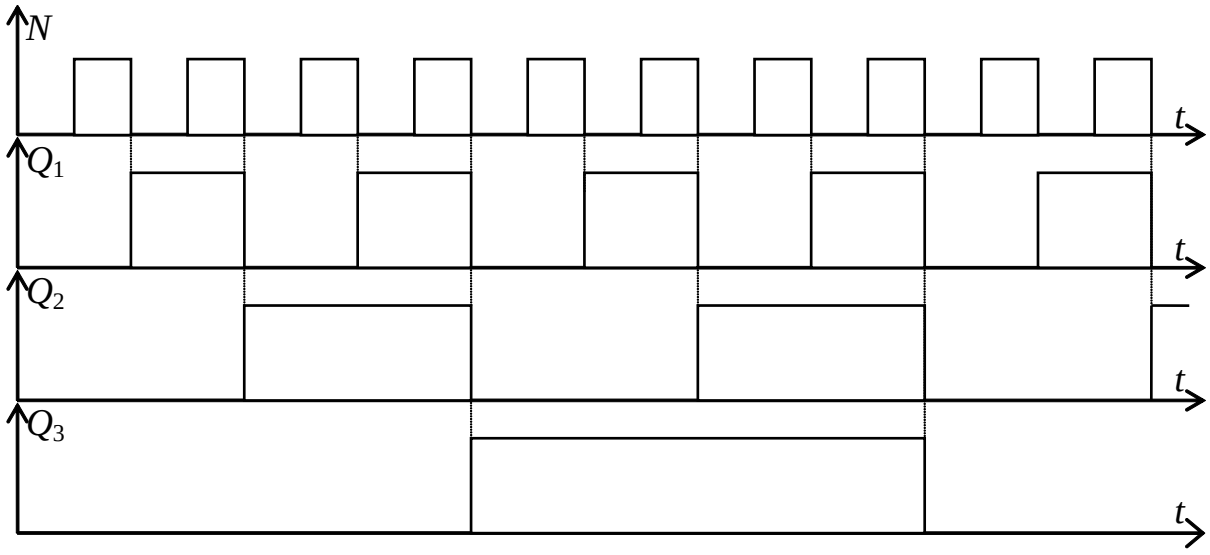


Рис. 4.1 — Временная диаграмма работы двоичного счетчика

Для организации модуля счета асинхронного счетчика не равного степени 2, например 5, необходимо организовать асинхронный сброс всех триггеров счетчика в исходное нулевое состояние при поступлении пятого по счету импульса синхронизации. Пятый по счету импульс синхронизации согласно временной диаграмме, представленной на рис. 4.1 устанавливает первый триггер в состояние «1», второй триггер к этому моменту времени находился в состоянии «0», третий — «1». Таким образом, объединяя выходы первого и третьего триггеров логикой «И», можно синтезировать сигнал, свидетельствующий о том, что счетчик находится в состоянии «101» (число 5) и по этому сигналу организовать асинхронный сброс всех триггеров счетчика в исходное нулевое состояние.

При организации вычитающего счетчика исходным состоянием будет такое, когда все триггеры счетчика находятся в состоянии «1».

Принципиальная схема асинхронного суммирующего счетчика по модулю 5 приведена на рис. 4.2, а временная диаграмма, поясняющая его работу, на рис. 4.3.

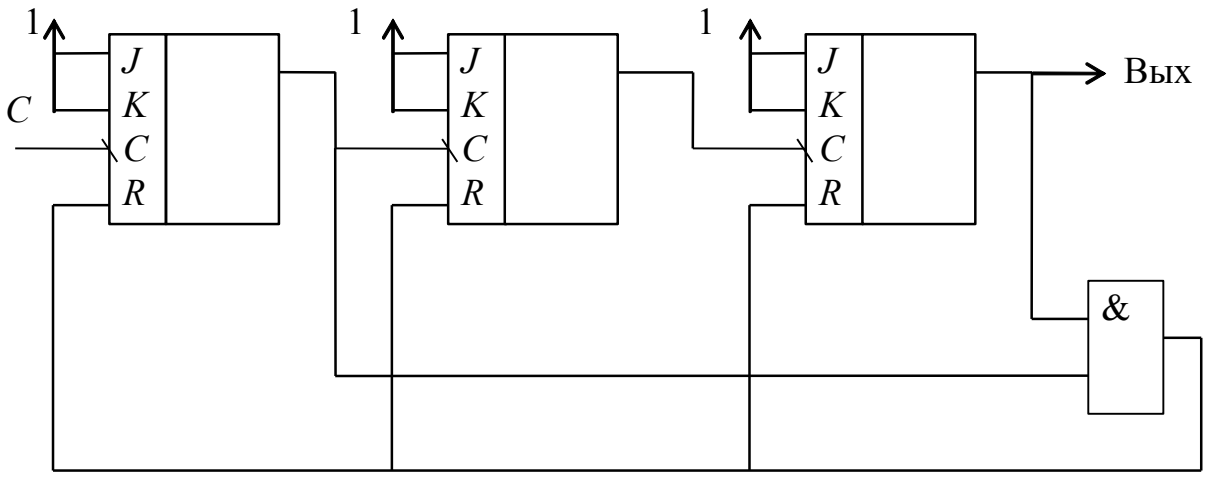


Рис. 4.2 — Принципиальная схема счетчика с произвольным модулем счета

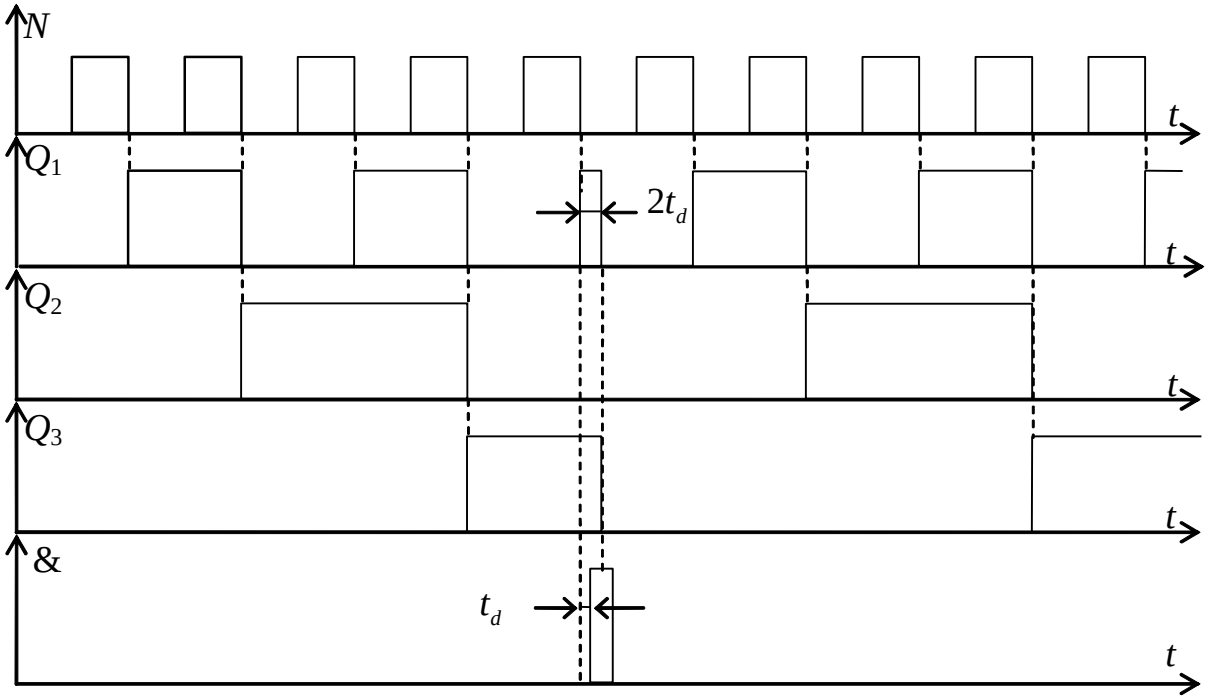


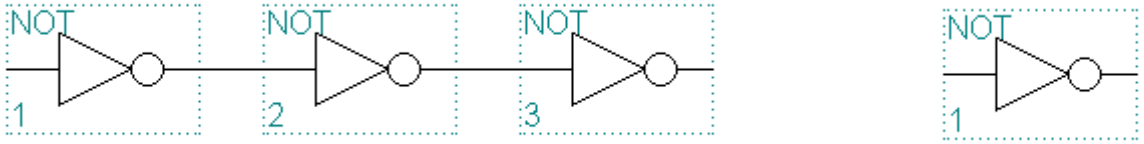
Рис. 4.3 — Временная диаграмма работы счетчика

Особенностью работы счетчика является то, что на выходе схемы «И» формируются очень короткие импульсы, зарегистрировать которые с помощью обычных средств наблюдения за процессом работы, например, осциллографами, как правило, не представляется возможным, так как длительность этих импульсов составляет $2t_d$, где t_d — время задержки переключения триггеров и логической схемы «И», определяемой технологией изготовления этих элементов. Как правило, это время составляет от единиц до сотен наносекунд, и зарегистрировать его достаточно сложно.

Таким образом, модуль счета асинхронного счетчика может быть спроектирован произвольным, для этого достаточно объединить логикой 2- , 3- и более «И» соответствующие выходы триггеров счетчика, синтезируя тем самым сигнал асинхронного сброса всех триггеров счетчика в исходное нулевое состояние.

На практике иногда представляет интерес не сброс всех триггеров счетчика, а их предустановка в состояние «1», либо промежуточный вариант, когда часть триггеров сбрасывается в «0», а часть устанавливается в «1». В каждом конкретном случае необходимо выбирать тот вариант, который дает наименьшее число элементов и их связей.

При синтезе схемы асинхронного счетчика в среде разработки «QUARTUS II» необходимо учитывать еще одну особенность. Элементарный триггер, входящий в состав макроячейки ПЛИС фирмы Altera, не состоит из базовых логических элементов 2И-НЕ или 2ИЛИ-НЕ. Его функционирование основывается на других правилах и приемах, составляющих особенность построения ПЛИС фирмы Altera. В результате чего время задержки переключения триггера практически равно нулю. Эта особенность функционирования триггера в составе ПЛИС вносит определенные коррективы в работу схем, построенных на его основе. Схема счетчика с асинхронным сбросом, показанная на рис. 4.2 и реализованная в среде разработки «QUARTUS II» работает неадекватно и временная диаграмма, показанная на рис. 4.3, не реализуется. Причиной этому служит малая задержка во времени переключения триггеров схемы. Другими словами, для реализации требуемых временных диаграмм необходима некоторая дополнительная задержка во времени формирования импульса сброса, реализуемого на выходе схемы совпадения 2И (см. рис. 4.2). Решить эту проблему простым последовательным соединением нескольких элементарных логических элементов, как это показано на рис. 4.4 а) не дает желаемого результата. Система автоматизированного проектирования «QUARTUS II» производит свою дополнительную «оптимизацию» и в реалии заменяет цепочку, показанную на рис. 4.4 а) более простой, как показано на рис. 4.4 б).



а) б)

Рис. 4.4 — Оптимизация схемы, производимая САПР «QUARTUS II»

Включить дополнительный инвертор на выходе схемы, например 2И, также не дает желаемого результата. САПР заменяет эту комбинацию элементов новым элементом 2И-НЕ.

Решить эту проблему можно добавлением в состав схемы логического элемента, имеющего внешний физический вывод микросхемы ПЛИС, как это показано на рис. 4.5.

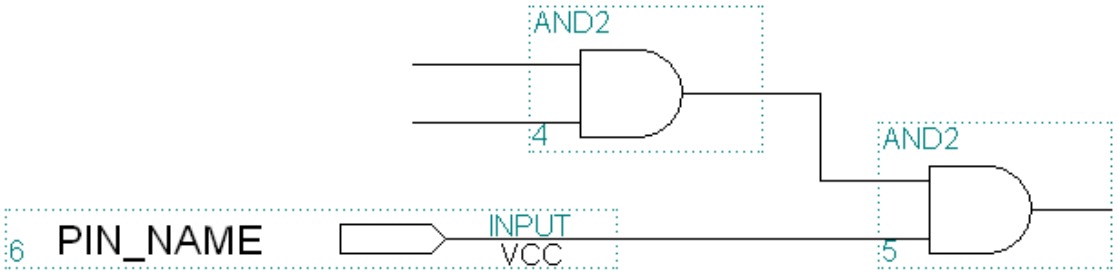


Рис. 4.5 — Реализация требуемой дополнительной задержки

САПР «QUARTUS II» при этом воспринимает этот дополнительный элемент, вводимый нами исключительно для формирования дополнительного времени задержки, как функционально необходимый, управляемый внешними сигналами, и оптимизацию схемы не производит. Наша задача заключается лишь в том, чтобы на этот дополнительный вывод подать сигнал требуемого логического уровня.

4.2.2. Реализация счетчика в системе «QUARTUS II»

Для иллюстрации возможностей программируемой логики рассмотрим создание на базе ПЛИС простейшего счетчика числа импульсов.

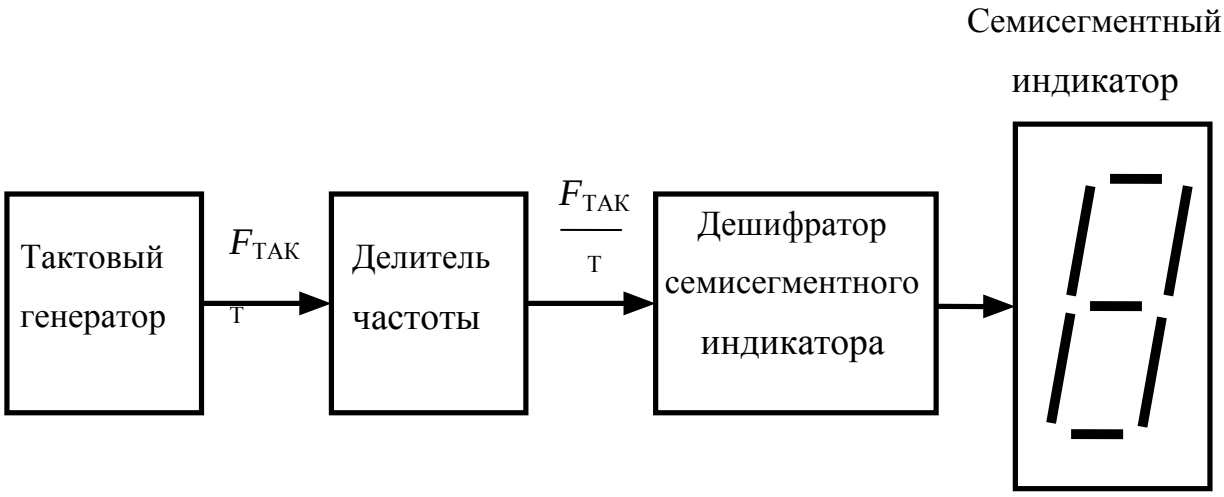


Рис. 4.6 — Структурная схема простейшего счетчика числа импульсов

В схеме, показанной на рис. 4.6, используется тактовый генератор, который формирует сигнал с частотой $F_{ТАКТ}$. Этот сигнал поступает на вход делителя частоты. Частота выходного сигнала делителя частоты меньше частоты входного сигнала в N раз. Коэффициент деления N делителя частоты определяется его внутренней организацией.

С выхода делителя частоты сигнал с частотой $F_{ТАКТ}/N$ поступает на вход дешифратора семисегментного индикатора, который преобразовывает входной сигнал в сигнал управления семисегментным индикатором.

В итоге, при поступлении на вход схемы от тактового генератора N импульсов, показания индикатора изменятся на единицу.

Для выполнения такой схемы на обычных цифровых микросхемах потребовалось бы не менее двух корпусов (при увеличении N и количества индикаторов это число быстро возрастает). При использовании микросхем программируемой логики ситуация упрощается.

Рассмотрим реализацию счетчика с коэффициентом деления 1024. Это число может быть записано как 210. Такой коэффициент деления могут обеспечить 10 последовательно соединенных D-триггеров. На рис. 4.7 показаны символы наиболее распространенных триггеров, а на рис. 4.8 показана принципиальная схема делителя на 1024, выполненная на D-триггерах.

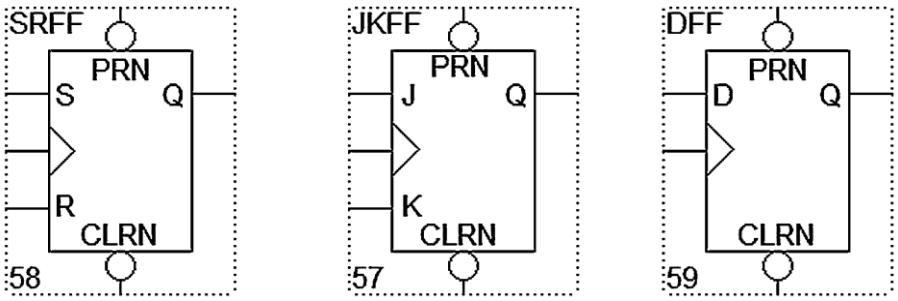


Рисунок 4.7 — Элементы САПР для RS, JK и D-триггеров

Для анализа работы схемы счетчика понадобится дешифратор, который был создан Вами на предыдущем занятии и дополнительный четырехразрядный счетчик импульсов, осуществляющий дополнительное деление полученного значения частоты в 2, 4, 8 и 16 раз. При этом если величина входной частоты будет составлять порядка одного килогерца, показания индикатора будут обновляться примерно через секунду (то есть в 1024 раз медленнее).

4.3. Порядок выполнения работы

4.3.1. Синтезировать схему асинхронного двоичного счетчика с модулем счета, равным 1024. Сохранить сему как один элемент.

4.3.2. К выходу схемы (выход последнего триггера счетчика) подключить один из светодиодов.

4.3.3. Ко входу схемы подключить тактовый генератор. Тактовая частота поступает на вход CLOCK (43 вывод ПЛИС). Величина тактовой частоты составляет порядка 1 кГц. В качестве генератора использован интегральный таймер.

4.3.4. Скомпилировать схему и запрограммировать ПЛИС.

4.3.5. Пронаблюдать работу счетчика, оценить период мигания светодиода.

4.3.6. Синтезировать схему счетчика с модулем счета, определяемым по индивидуальному заданию. Индивидуальные задания приведены в табл. 4.2. Сохранить схему как один элемент.

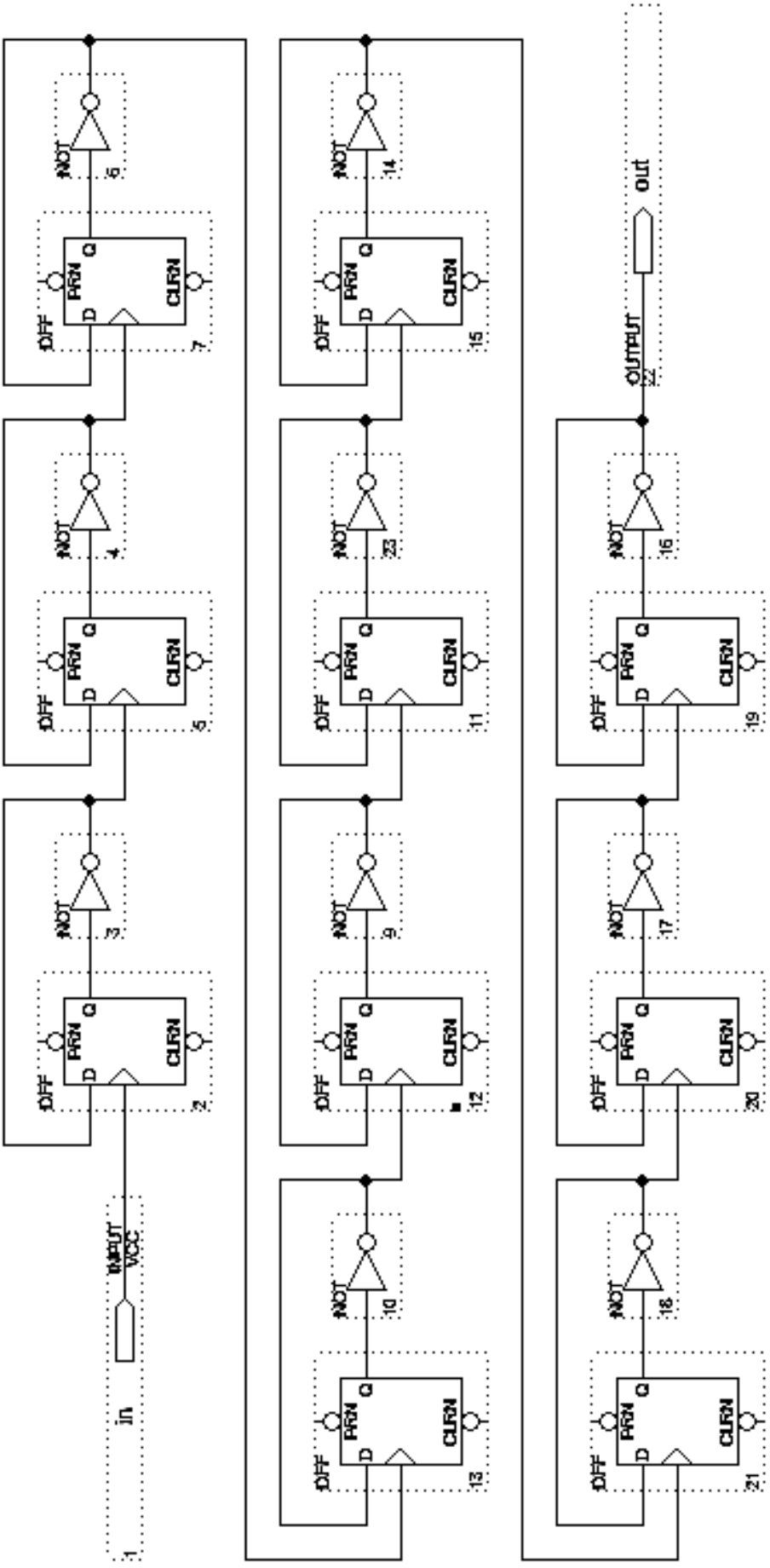


Рис. 4.8 — Предварительный делитель частоты на 1024

Таблица 4.2 — Индивидуальные задания

№ бригады	1	2	3	4	5	6	7	8	9	10
Модуль счета	5	6	7	9	10	11	12	13	14	15

4.3.7. Провести симуляцию работы схемы счетчика, разработанного по индивидуальному заданию.

4.3.8. К выходам триггеров счетчика, подключить синтезированный ранее семисегментный дешифратор.

4.3.9. На вход счетчика подать сформированные секундные импульсы, добавив в схему синтезированный ранее двоичный счетчик с модулем счета 1024.

4.3.10. Скомпилировать схему устройства и запрограммировать ПЛИС.

4.3.11. Пронаблюдать работу счетчика на семисегментном индикаторе.

4.4. Содержание отчета

4.4.1. Цель работы.

4.4.2. Скриншот схемы предварительного делителя на 1024.

4.4.3. Скриншот схемы счетчика импульсов по индивидуальному заданию.

4.4.4. Результаты симулирования работы схемы. Скриншот симуляции.

4.4.5. Скриншот всего проекта.

4.4.6. Выводы.

4.5. Контрольные вопросы для защиты

4.5.1. Дайте определение последовательностной схемы.

4.5.2. Дайте определение двоичному счетчику импульсов.

4.5.3. Дайте определение счетчику импульсов с произвольным модулем счета.

4.5.4. Дайте определение асинхронному счетчику импульсов.

4.5.5. Дайте определение синхронному счетчику импульсов.

4.5.6. Почему нельзя использовать для формирования задержки фронта импульса три включенных подряд инвертора?

4.5.7. Как можно обойти проблему, обозначенную в п. 4.5.6?

4.5.8. Что дает использование макросов по созданию собственного элемента?

4.5.9. Как следует выбирать модуль счета предделителя?

4.5.10. Какова частота следования импульсов задающего генератора?

ЛАБОРАТОРНАЯ РАБОТА № 5
СИНТЕЗ ЛОГИЧЕСКИХ СХЕМ

5.1. Цель работы

Закрепление навыков синтеза схем на основе ПЛИС и анализа работы цифровых схем с помощью САПР «QUARTUS II».

5.2. Теоретические сведения

Теоретические сведения приведены в подразделах 1.2, 2.2, 3.2, 4.2 настоящих рекомендаций.

5.3. Индивидуальные задания

5.3.1. Вариант 1

С использованием макросов, полученных ранее в ходе выполнения лабораторной работы №2, синтезировать схему, обеспечивающую мерцание светодиода LED 1 с частотой 1 Гц и подсчет количества мерцаний с выводом десятичной цифры на индикаторах HL1, HL2. Обнуление счетчика должно происходить по нажатию клавиши SA1.

5.3.2. Вариант 2

С использованием полученных ранее макросов синтезировать схему, обеспечивающую мерцание светодиода LED 1 с частотой 1 Гц и подсчет количества мерцаний с выводом цифры в двоичной системе на индикаторах HL1, HL2. Обнуление счетчика должно происходить по нажатию клавиши SA1.

5.3.3. Вариант 3

С использованием полученных ранее макросов синтезировать схему, обеспечивающую мерцание светодиода LED1 с частотой 1 Гц и подсчет количества мерцаний с выводом цифры в восьмеричной системе на индикаторах HL1, HL2. Обнуление счетчика должно происходить по нажатию клавиши SA1.

5.3.4. Вариант 4

С использованием полученных ранее макросов синтезировать схему, обеспечивающую поочередное мерцание светодиодов, начиная от LED1 и за-

канчивая LED8, с частотой 1 Гц и подсчет количества мерцаний LED3 с выводом цифры в десятичной системе на индикаторах HL1, HL2. Обнуление счетчика должно происходить по нажатию клавиши SA1.

5.3.5. Вариант 5

С использованием полученных ранее макросов синтезировать схему, обеспечивающую мерцание светодиодов, начиная от LED1 и заканчивая LED8, частотой 0,5 Гц и подсчет количества мерцаний LED с выводом цифры в двоичной системе на индикаторах HL1, HL2. Обнуление счетчика должно происходить после 30 мерцаний.

5.3.6. Вариант 6

С использованием полученных ранее макросов синтезировать схему, обеспечивающую подсчет нажатий на кнопку SA1 с индикацией количества нажатий на светодиодном индикаторе HL1, HL2 и индикацией двоичного кода при помощи светодиодов LED1…LED8 согласно весовым коэффициентам.

5.3.7. Вариант 7

С использованием полученных ранее макросов синтезировать схему, обеспечивающую подсчет нажатий на кнопку SA1 с индикацией количества нажатий на светодиодном индикаторе HL1, HL2 и сбросом счетчика по нажатию кнопки SA2.

5.3.8. Вариант 8

С использованием полученных ранее макросов синтезировать схему, работу макета в режиме таймера с периодом 60 секунд. Запуск таймера по нажатию на кнопку SA1, сброс по нажатию на кнопку SA2. По достижению цифры 60 должны быть засвечены все светодиоды от LED1 до LED8.

5.4. Порядок выполнения работы

5.4.1. Синтезировать проектируемую схему по индивидуальному заданию. Сохранить схему как один элемент.

5.4.2. К выходам схемы подключить светодиодные индикаторы по индивидуальному заданию.

5.4.3. К входу схемы подключить тактовый генератор. Тактовая частота поступает на вход CLOCK (43 вывод ПЛИС). Величина тактовой частоты со-

ставляет порядка 1 кГц. В качестве генератора использован интегральный таймер.

5.4.4. Ко входам схемы подключить кнопки по индивидуальному заданию.

5.4.5. Скомпилировать схему и запрограммировать ПЛИС.

5.4.6. Проанализировать работу схемы.

5.4.7. Провести симуляцию работы схемы.

5.5. Содержание отчета

5.5.1. Цель работы.

5.5.2. Индивидуальное задание.

5.5.3. Скриншот проектируемой схемы.

5.5.4. Результаты симулирования работы схемы. Скриншот симуляции.

5.5.5. Выводы.

5.6. Контрольные вопросы для защиты

В качестве контрольных вопросов используются контрольные вопросы всех предыдущих лабораторных работ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Новиков, Ю. В. Введение в цифровую схемотехнику: учеб. пособие / Ю. В. Новиков. — М.: ИНТУИТ: БИНОМ. ЛЗ, 2007. — 344 с.
2. Микушин, А. В. Цифровые устройства и микропроцессоры: учебное пособие / А. В. Микушин, А. М. Сажнев, В. И. Сединин. — СПб.: БХВ-Петербург, 2010. — 832 с.
3. Титце, У. Полупроводниковая схемотехника / У. Титце, К. Шенк.— М.: Мир, 1989. — 384 с.
4. Система проектирования «QUARTUS II» [Электронный ресурс] / ALTERA.— http:// www.altera.com/.— 1.01.2017
5. Точчи, Р. Дж. Цифровые системы: Теория и практика / Р. Дж. Точчи, Н. С. Уидмер. — 8-е изд. — М.; СПб.; К.: Вильямс, 2004. — 1021 с.
6. Клочков, Г. Л. Цифровые устройства и микропроцессоры: Учебник / Г. Л. Клочков. — Воронеж: ВИРЭ, 2005. — 320 с.

Наличие в фонде библиотеки СевГУ электронных изданий учебной литературы

№ п/п	Наименование изданий учебной литературы	Количество экземпляров
1.	Сажнев, А. М. Цифровые устройства и микропроцессоры : учеб. пособие для академического бакалавриата / А. М. Сажнев. — 2-е изд., перераб. и доп. — Москва : Юрайт, 2019. — 139 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-10883-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/432199 (дата обращения: 09.04.2019).	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
2.	Рафиков, Р.А. Электронные сигналы и цепи. Цифровые сигналы и устройства [Электронный ресурс] : учебное пособие / Р.А. Рафиков. — Электрон. дан. — Санкт-Петербург : Лань, 2016. — 320 с. — Режим доступа: https://e.lanbook.com/book/72997. — Загл. с экрана	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
3.	Пухальский, Г. И. Проектирование цифровых устройств [Электронный ресурс] : учебное пособие / Г. И. Пухальский, Т. Я. Новосельцева. — Электрон. дан. — Санкт-Петербург : Лань, 2012. — 896 с. — Режим доступа : https://e.lanbook.com/book/68474. — Загл. с экрана.	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
4.	Китаев, Ю. В. Основы цифровой техники [Электронный ресурс] : учебное пособие / Ю. В. Китаев. — Электрон. дан. — Санкт-Петербург : ИТМО, 2007. — 87 с. — Режим доступа : https://e.lanbook.com/book/43631. — Загл. с экрана.	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ