**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ**

отчет

**по лабораторной работе №1**

**по дисциплине «****Элементная база цифровых систем»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1335 |  | Максимов Ю Е |
| Преподаватель |  | Буренева О И |

Санкт-Петербург

2024

**Лабораторная работа 1.**

**ЗНАКОМСТВО С СИСТЕМОЙ ПРОЕКТИРОВАНИЯ QUARTUS II,**

**РЕАЛИЗАЦИЯ КОМБИНАЦИОННОЙ СХЕМЫ**

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

**Задание на работу**

Выполнить проектирование комбинационной схемы, реализующей функцию от четырех переменных, заданную набором входных данных, на которых она принимает единичные значения. Необходимо составить таблицу истинности функции, выполнить минимизацию функции с использованием карт Карно или метода Квайна – Мак-Класки, основанного на применении операций склеивания и поглощений.

Компонент *x*4 входного вектора (*x*4, *x*3, *x*2, *x*1) следует считать старшим двоичным разрядом.

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

Оба варианта реализации реализовать в одном проекте, предусмотрев два соответствующих выхода.

**Вариант**

Десятичные значения векторов входных переменных (*x*4, *x*З, *x*2, *x*1), на которых переключательная функция *y*(*x*4, *x*З, *x*2, *x*1) равна логической «1»:

0, 1, 2, 5, 6, 10, 13, 14.

На других входных наборах функция равна логическому «0».

**Выполнение работы**

1. Синтез логической схемы.

Составим таблицу истинности, имеющую 24 строк (по строке для каждого набора входных переменных) и 4 + 2 столбцов (табл. 1).

Таблица 1 – Таблица истинности функции y(*x*4, *x*З, *x*2, *x*1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №10 | Входные переменные | | | | Функция  *y*(*x*4, *x*З, *x*2, *x*1) |
| *x*4 | *x*3 | *x*2 | *x*1 |
| 0 | 0 | 0 | 0 | 0 | **1** |
| 1 | 0 | 0 | 0 | 1 | **1** |
| 2 | 0 | 0 | 1 | 0 | **1** |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | **1** |
| 6 | 0 | 1 | 1 | 0 | **1** |
| 7 | 0 | 1 | 1 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 | **1** |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | **1** |
| 14 | 1 | 1 | 1 | 0 | **1** |
| 15 | 1 | 1 | 1 | 1 | 0 |

От таблицы истинности перейдем к совершенной дизъюнктивной нормальной форме (СДНФ), т. е. к дизъюнкции конституент единицы искомой функций, путем составления логической суммы тех входных наборов, на которых функция принимает единичное значение.

СДНФ:



Выполним минимизацию функции с помощью карты Карно (диаграммы Вейча), которая представляет собой развертку гиперкуба на плоскости. Элементы в соседних ячейках отличаются лишь в одном разряде. Карте Карно соответствует циклический код Грея, в котором каждая следующая комбинация отличается от предыдущей значением одного разряда. При представлении функции с помощью карты Карно необходимо учитывать, что крайние столбцы и строки считаются соседними. Если свернуть карту в пространстве, соединив ее края, получим тор с такими же свойствами.

Алгоритм нахождения МДНФ по карте Карно:

1) необходимо выделить на карте контуры так, чтобы были соблюдены следующие условия:

- контуры должны содержать в ячейках внутри себя только единицы,

- контуры должны быть прямоугольными или квадратными,

- они должны включать число ячеек, равное степени 2: 1, 2, 4, 8 или 16;

- крайние столбцы, крайние строки и угловые ячейки считаются соседними;

- каждый контур должен охватывать по возможности наибольшее число ячеек,

- контуры могут пересекаться,

- не должно быть контуров, все ячейки которых входят в другие контуры;

- все единицы в ячейках должны быть покрыты контурами.

2) Затем необходимо по контурам составить элементарные конъюнкции, соответствующие им. Для этого при рассмотрении контура выделяются переменные, которые постоянны в контуре, они входят в элементарную конъюнкцию, переменные же, входящие в контур вместе со своими инверсиями, исключаются из нее.

Заполним ячейки карты значениями из последнего столбца таблицы истинности (табл. 1). Выделим контуры и для каждого запишем выражение (рис. 1).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x*2 *x*1  *x*4 *x*З |  |  |  |  |
|  | **1** | **1** | 0 | **1** |
|  | 0 | **1** | 0 | **1** |
|  | 0 | **1** | 0 | **1** |
|  | 0 | 0 | 0 | **1** |



Рисунок 1 – Минимизация функции с помощью карты Карно

Объединим полученные конъюнкции логической суммой (дизъюнкция) и получим МДНФ:



2. Функциональная схема.

Преобразуем полученное выражение МДНФ к заданным базисам, используя правило де Моргана ( и ) и двойное отрицание ().

1) not (НЕ), band\* (\*-НЕ И), nand\* (\*-И-НЕ):



2) not (НЕ), bor\* (\*-НЕ-ИЛИ), nor\* (\*-ИЛИ-НЕ),



По полученным выражениям составим функциональную схему, реализующую функцию от четырех переменных (рис. 2). Для каждого базиса предусмотрим отдельный выход (*у*1 и *y*2).

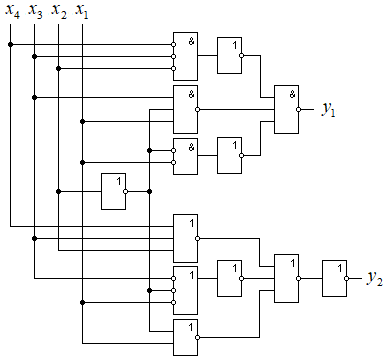


Рисунок 2 – Функциональная схема

3. Схема узла в САПР Quartus II.

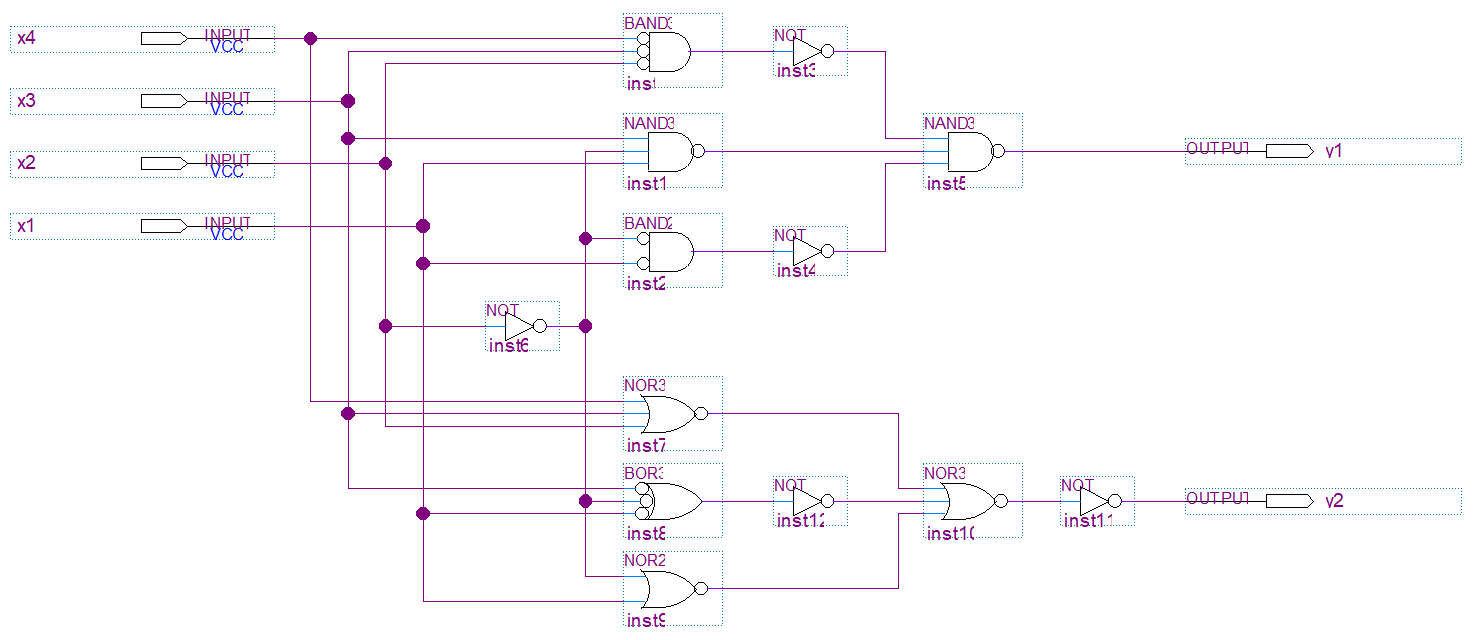


Рисунок 3 – Схема узла в Quartus II

4. Результаты функционального и временного моделирования.

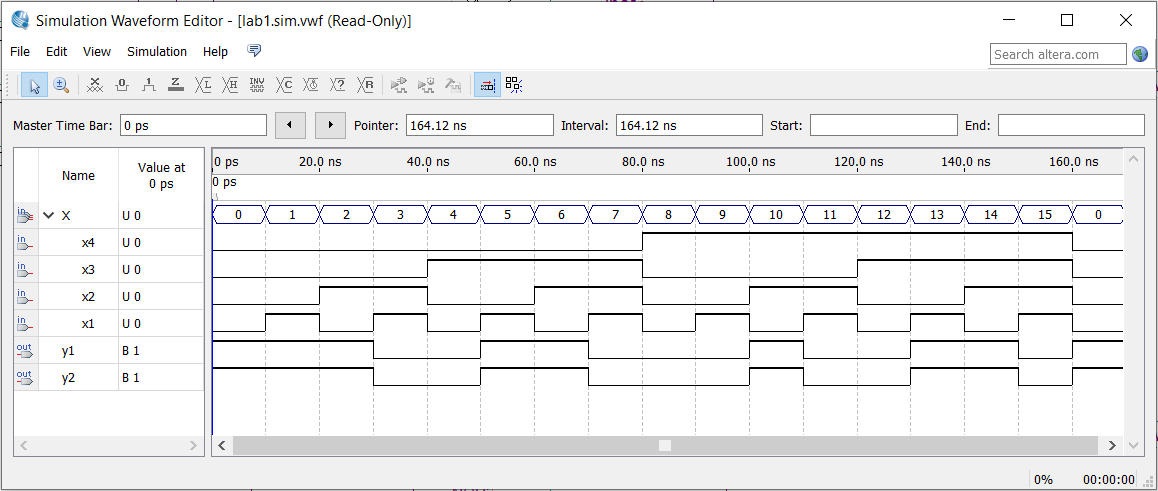


Рисунок 4 – Результаты функционального моделирования

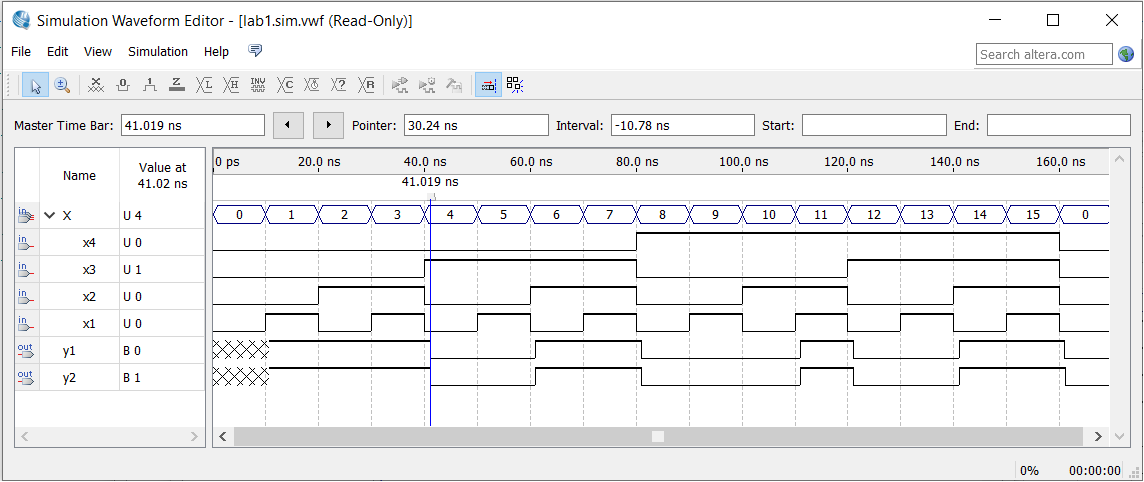


Рисунок 5 – Результаты временного моделирования

Результаты функционального моделирования полностью совпадают с таблицей истинности заданной функции для двух вариантов реализации схемы. Функциональное моделирование проверяет логику работы схемы, а временное – сигналы с учетом всех возникающих задержек сигналов по схеме. В нашем случае, временное моделирование показало задержку выходного сигнала на один такт и tз y1 = 11,019ns, tз y2 = 11,029ns. Т.е. сигналы на выходе “отстают” от сигналов на входе на один такт, причем сигнал на выходе *y*1 появляется на 0,01 ns раньше, чем на выходе *y*2 (рис. 6). Это связано с тем, что число последовательно соединенных элементов во второй схеме на один больше (инверсия на выходе), чем в первом варианте, что требует дополнительного времени на прохождение сигнала.

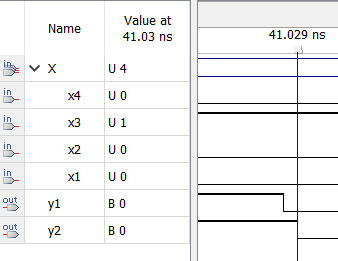


Рисунок 6 – Задержка сигнала *y*2

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

Для проведения макетных экспериментов при выполнении лабораторной работы используется учебная плата, основу которой составляет ПЛИС класса FPGA Altera Cyclone II EP2C5Q208C8N. В состав платы также входят 8-ми разрядный переключатель, 8-ми раз рядная светодиодная линейка, тактовая кнопка, порты расширения, интерфейс USB для подключения к ПК, интерфейс Active serial. Исходными данными для загрузки являются файлы, созданные САПР Quartus II в процессе компиляции проекта, содержащие конфигурационную информацию.

Для загрузки проекта в учебную плату выполним следующие действия.

1. Для подготовки к программированию кристалла выберем пункт Device в меню Assignments, затем, проверив в открывшемся основном окне имя микросхемы EP2C5Q208C8, нажмем кнопку Device and Pin Options и установим параметры, приведенные на рисунке 7, то есть Configuration scheme – Active Serial, Use configuration device – EPCS1, включим опцию сжатия Generate compressed bitstreams.

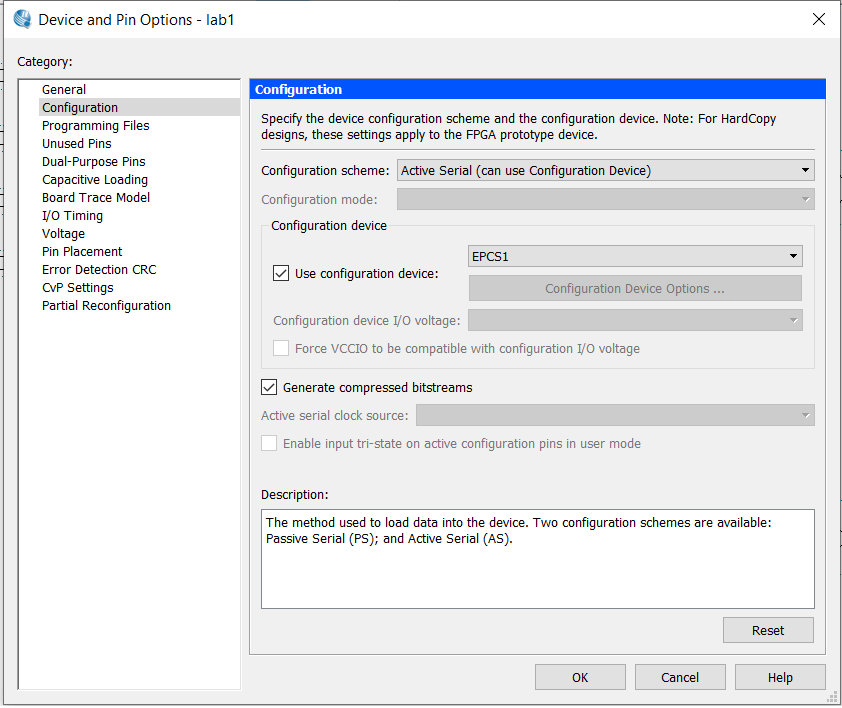


Рисунок 7 - Параметры устройства программирования

2. Программирование начнем с «распиновки» кристалла: для каждого входного и выходного сигнала проекта назначим реальный контакт ПЛИС. Данное действие можно выполнить с использованием утилиты Pin Planer, запускаемой через меню Assignments.

Номера используемых контактов ПЛИС указаны в таблице 2 и 3.

Таблица 2 – Номера контактов ПЛИС учебной платы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер бита 8-разрядного  переключателя | 0 | 1 | 2 | 3 |
| Номер контакта FPGA | 170 | 169 | 168 | 165 |

Таблица 3 – Номера контактов ПЛИС учебной платы

|  |  |  |
| --- | --- | --- |
| Номер светодиода 8-разрядная светодиодная линейка | 0 | 1 |
| Номер контакта FPGA | 171 | 173 |

Вид окна Pin Planner представлен на рисунке 8.

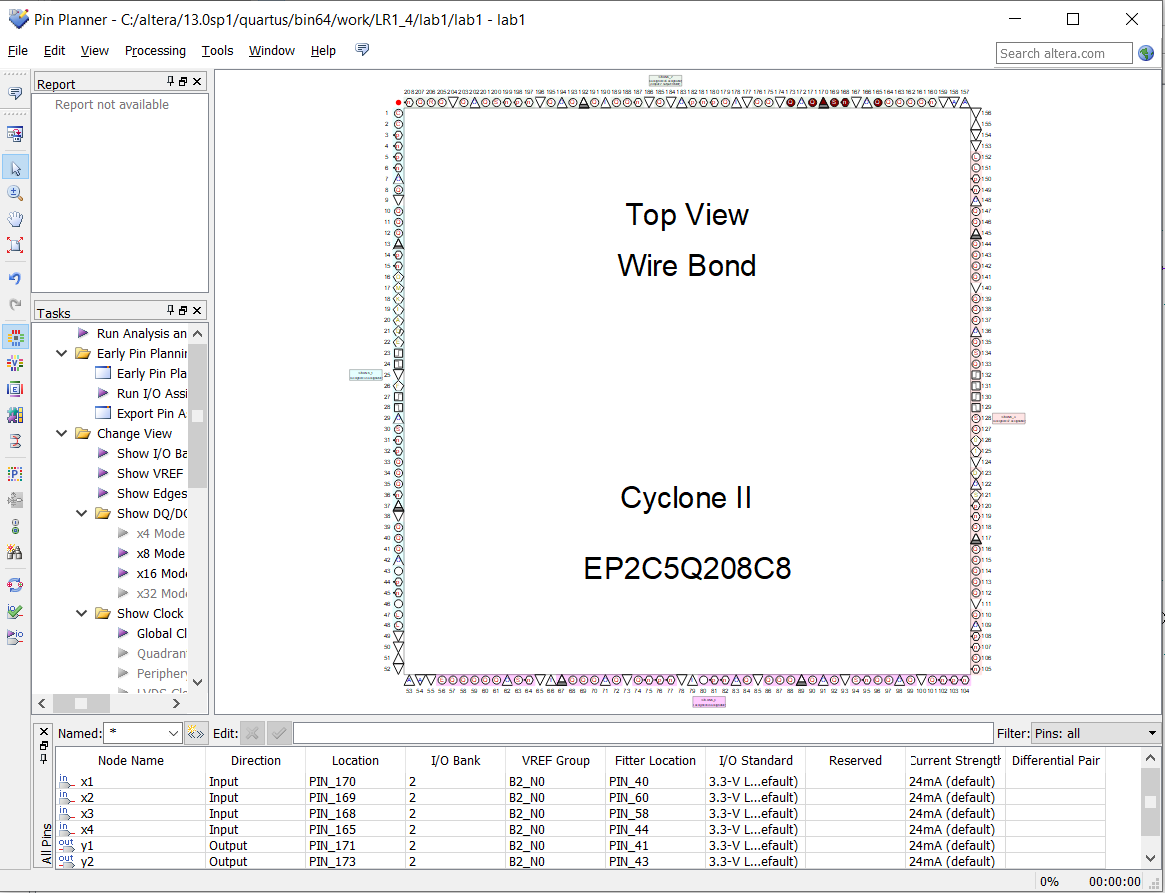


Рисунок 8 – Вид окна Pin Planner

3. После назначения контактов выполним повторную компиляцию проекта. Результатом компиляции является создание в директории проекта файлов конфигурации \*.sof и \*.pof.

4. Выполним программирование кристалла с использованием утилиты Programmer. Она вызывается пиктограммой  на панели инструментов менеджера проекта, либо соответствующей командой меню Tools. Окно программирования представлено на рисунке 9.

В этом окне выберем режим (mode) Active Serial Programming, согласившись с удалением всех устройств из списка при смене режима программирования. Затем с помощью кнопки Add file добавим файл \*.pof, созданный в подкаталоге Output files, и отметим пункты Program/Configure и Verify.

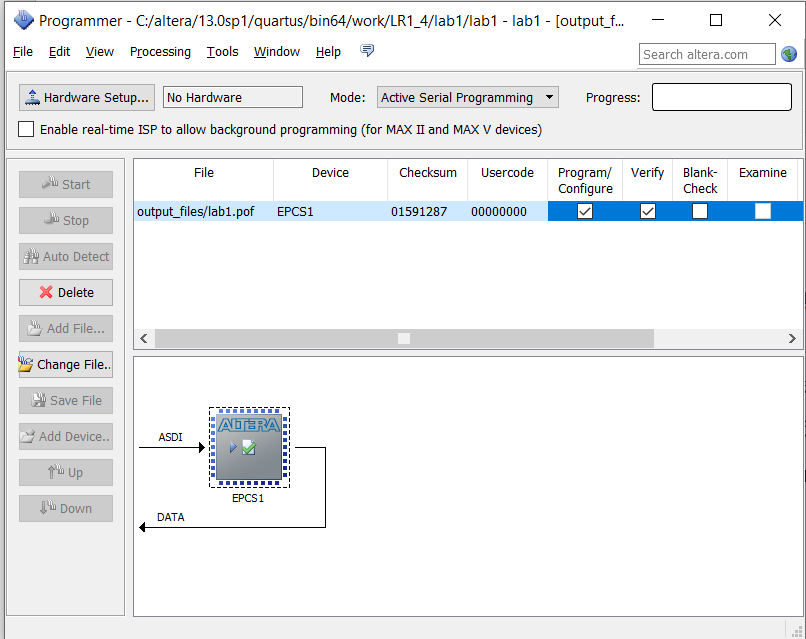


Рисунок 9 – Вид окна Programmer

Непосредственно программирование осуществляется нажатием кнопки Start. После этого в окне Progress будет визуально отображаться процесс выполнения программирования кристалла. После успешного завершения этого процесса в окне Progress установится значение 100%.

**Выводы**

В ходе выполнения работы были получены навыки проектирования цифровой схемы в системе автоматизированного проектирования Quartus II. Выполнены этапы создания цифровой схемы в графическом редакторе, моделирования работы схемы, загрузки результатов проектирования в программируемую логическую интегральную схему и проведение макетного эксперимента.

В результате получена комбинационная схема, реализующая функцию от четырех переменных, заданную набором входных данных, на которых она принимает единичные значения. Схема узла выполнена на логических элементах двух базисов, получены функциональные и временные диаграммы работы. Полученные навыки будут полезны при разработке более сложных цифровых узлов.