

## Работа № 1. Логические операции

1. Для заданных логических функций сформируйте в Quartus Prime BDF-файл (block diagram file), произведите для отладочной платы ПЛИС семейства Cyclone IV сопоставление пинов в Pin Planner. Составьте таблицу истинности, продемонстрируйте работу схемы на отладочной плате.
2. Для этих же функций сформируйте два файла описания на языке Verilog: с применением логических операций и в виде списка соединений.
3. Для п. 2 напишите общий testbench, произведите моделирование в ModelSim (или iVerilog + GTKWave). Проясните идентичность результатов моделирования.

№ бригады	Логические выражения
1	$y_0 = \overline{x_0} + x_1,$ $y_1 = \overline{x_0} \oplus x_1,$ $y_2 = \overline{x_0} + \overline{x_1} + x_2.$
2	$y_0 = \overline{x_0} \oplus x_1,$ $y_1 = \overline{x_2} (x_0 + \overline{x_1}),$ $y_2 = x_0 \cdot \overline{x_1} + x_2.$
3	$y_0 = x_0 \oplus \overline{x_1},$ $y_1 = \overline{x_0} \cdot (x_1 + x_2),$ $y_2 = (x_0 \oplus x_1) \cdot x_2.$
4	$y_0 = \overline{x_0} \cdot (x_1 \oplus x_2),$ $y_1 = x_1 \oplus (\overline{x_0} \cdot x_2),$ $y_2 = x_0 \cdot x_1 + \overline{x_2}.$
5	$y_0 = x_0 \oplus x_1 \oplus \overline{x_2},$ $y_1 = \overline{x_0} + \overline{x_2},$ $y_2 = x_0 + \overline{x_1} \cdot x_2.$
6	$y_0 = \overline{x_0} + \overline{x_1},$ $y_1 = \overline{x_0} \oplus x_1,$ $y_2 = \overline{x_0} + \overline{x_1} \oplus x_2.$
7	$y_0 = \overline{x_1} + x_2,$ $y_1 = \overline{x_0} \oplus x_2,$ $y_2 = \overline{x_0} \oplus \overline{x_1} + x_2.$

Указания:

Для утилит iVerilog + GTKWave используйте следующие команды в терминале:

```
> iverilog <file.v>
```

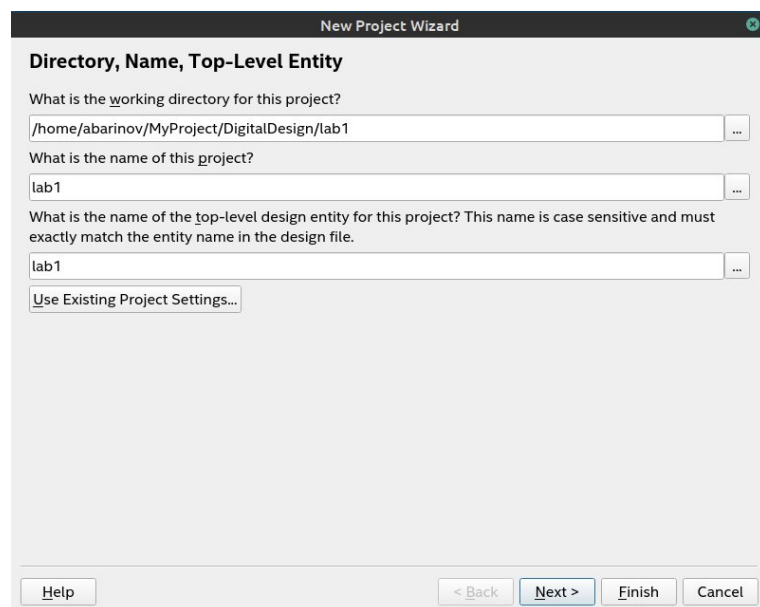
```
> vvp a.out
```

> gtkwave dump.vcd

## Выполнение задания № 1

### Создание проекта

1. Откройте Quartus Prime.
2. Создайте проект в Quartus Prime **File** > **New....**
3. В открывшемся окне выберите *New Quartus Prime Project*. Нажмите *OK*.
4. Теперь необходимо указать директорию, в которой будут расположены файлы проекта, а также задать имя проекта и модуля верхнего уровня (по умолчанию они совпадают). Рекомендуется для каждого проекта формировать отдельную папку. В качестве имени директории, проекта и модуля верхнего уровня укажите **lab1**. Нажмите *Next*.



5. В следующем окне выберите *Empty Project* и нажмите *Next*.
6. В окне добавления файлов ничего не выбираете и нажмите *Next*.
7. В окне настройки устройства ПЛИС выберите семейство устройств Cyclone IV E. Из списка доступных устройств выберите EP4CE6E22C8. Нажмите *Next*.

New Project Wizard

### Family, Device & Board Settings

Device Board

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

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: Cyclone IV E

Device: All

Target device

☐ Auto device selected by the Fitter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Core speed grade: Any

Name filter:

☒ Show advanced devices

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit elements	PLLs	Global Clocks
EP4CE6E22C6	1.2V	6272	92	92	276480	30	2	10
EP4CE6E22C7	1.2V	6272	92	92	276480	30	2	10
EP4CE6E22C8	1.2V	6272	92	92	276480	30	2	10
EP4CE6E22I7	1.2V	6272	92	92	276480	30	2	10

Help < Back Next > Finish Cancel

8. В следующем окне укажите инструмент моделирования ModelSim, а язык Verilog HDL. На текущем этапе моделирование проводится не будет, а пройдёт в задании 3, но так как проект для заданий один и тот же, то выбрать следует сейчас. Нажмите *Next*.

9. Ознакомьтесь с информацией в окне с итогами и нажмите *Finish*.

New Project Wizard

### EDA Tool Settings

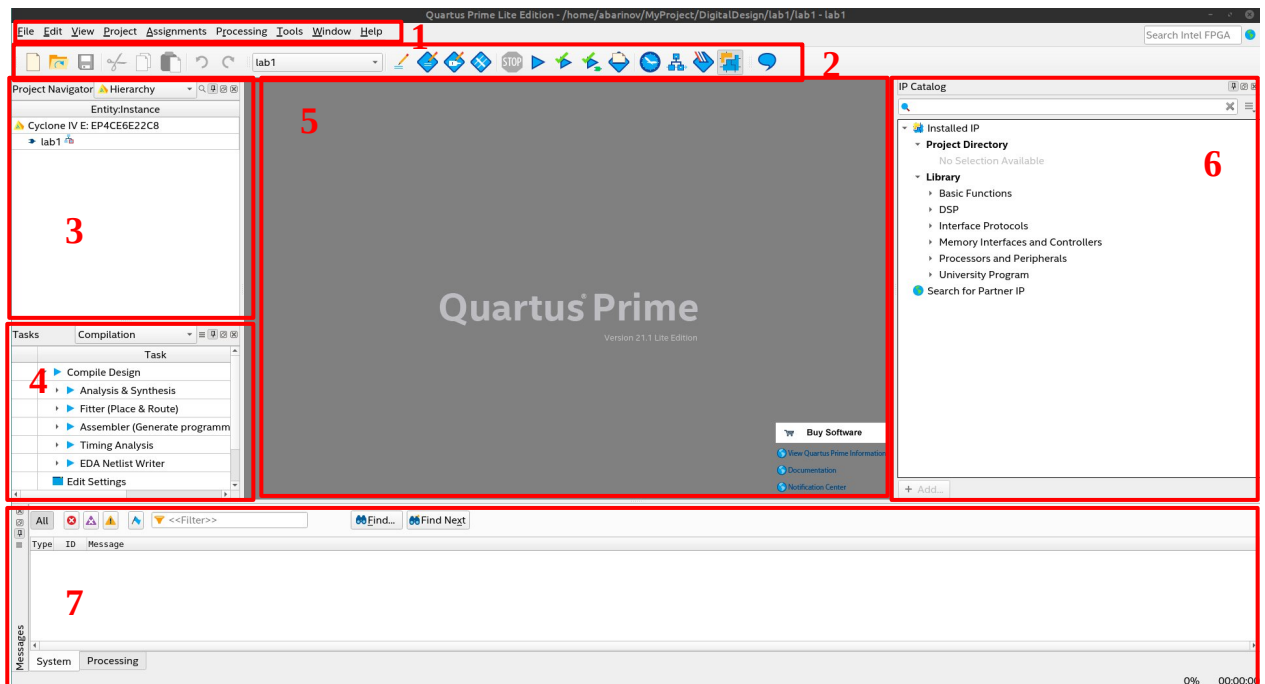
Specify the other EDA tools used with the Quartus Prime software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/...	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	ModelSim	Verilog HDL	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

Help < Back Next > Finish Cancel

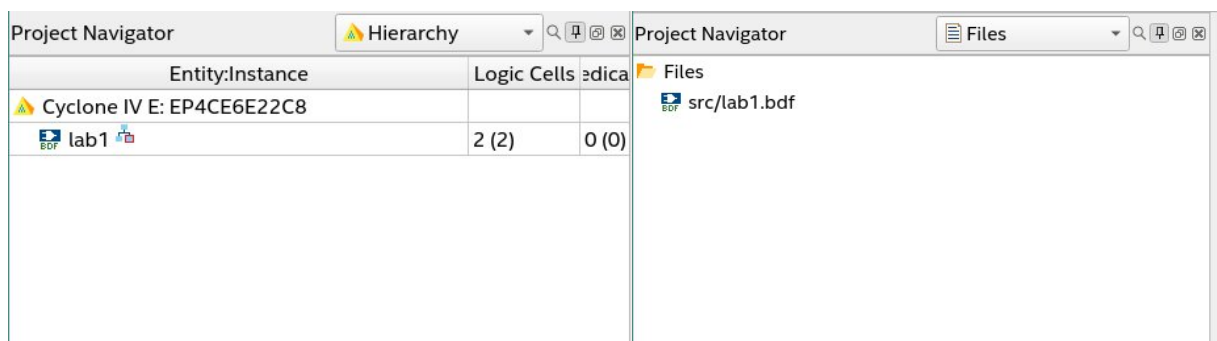
10. В результате появится окно Quartus с созданным проектом.



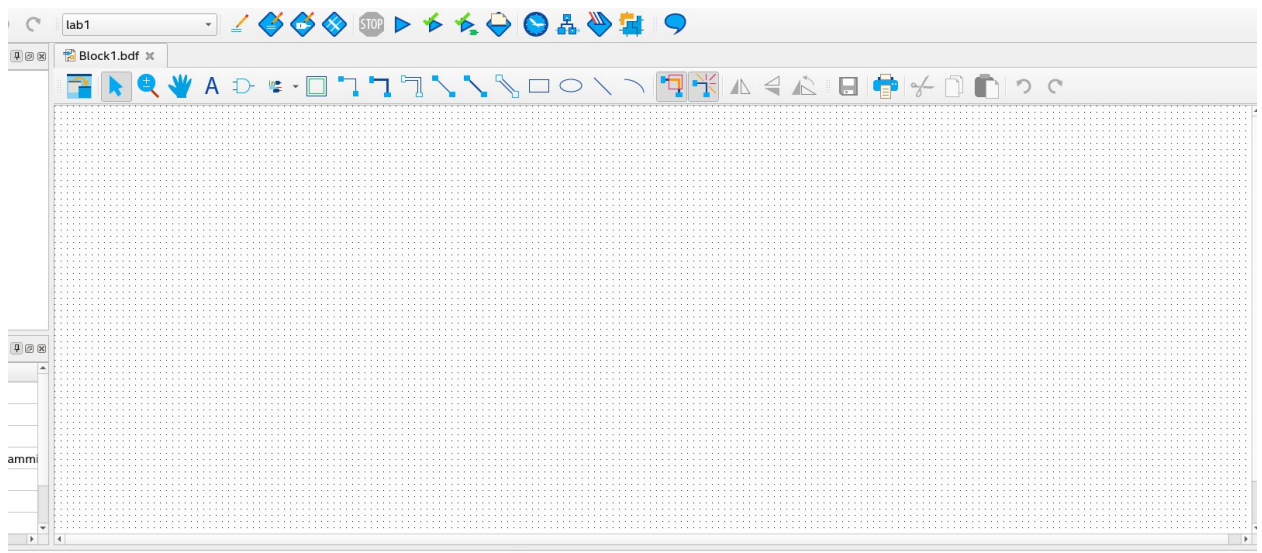
1 – панель меню, 2 – панель с иконками, 3 – навигатор проекта, 4 – список выполняемых задач в процессе компиляции проекта, 5 – основная область, 6 – список IP-блоков (можно закрыть), 7 – журнал с сообщениями о текущем статусе, предупреждениях и ошибок в процессе компиляции проекта




## Создание BDF-файла и компиляция проекта

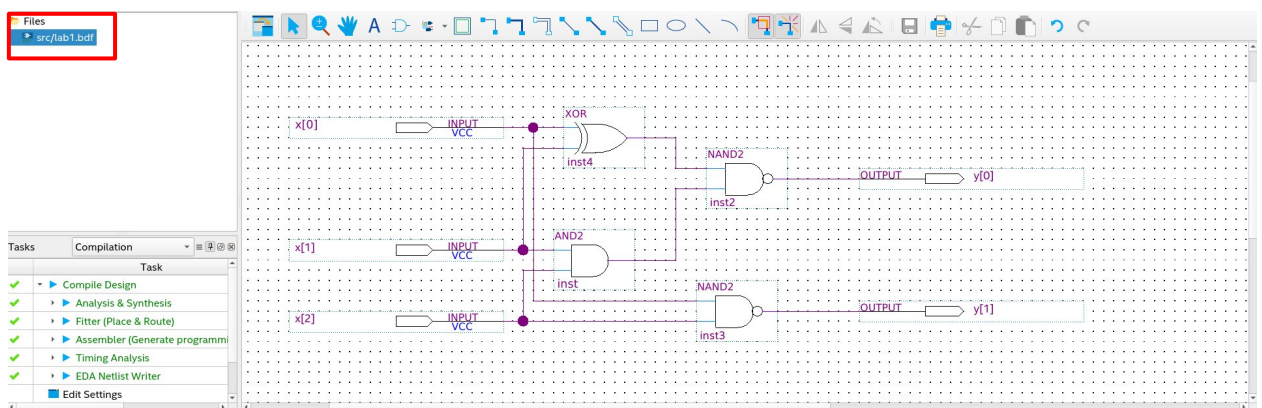
1. В навигаторе проекта Project Navigator измените представление с Navigator на Files.




2. Создайте файл схемного представления **File > New...**, выберите *Block Diagram / Schematic File*. Нажмите **OK**. В результате в главном окне появится область рисования схем с сеткой под именем **Block1.bdf**.



3. Инструментом Symbol Tool  добавьте из библиотеки primitives > logic необходимые логические вентили. При расположении элементов убедитесь, что их границы (штриховые области) не накладываются друг на друга.
4. Инструментом Orthogonal Node Tool произведите  соединение логических элементов проводниками.
5. Инструментом Pin Tool добавьте  входные и выходные пины. Двойным щелчком мыши по пину произведите переименование pin\_name входных пинов на x[0], x[1], x[2], y[0], y[1] и y[2].
6. Сохраните файл **File > Save**. При работе с проектом удобно файлы одинакового назначения сохранять в определённые папки (например, файлы HDL-описания в папку /src, файлы с тестами в папку testbench, результаты моделирования в папку simulation и так далее). Создайте папку в папке с проектом **src** и сохраните файл **lab1.bdf** в ней. Этот файл должен появиться в списке Project Navigator.



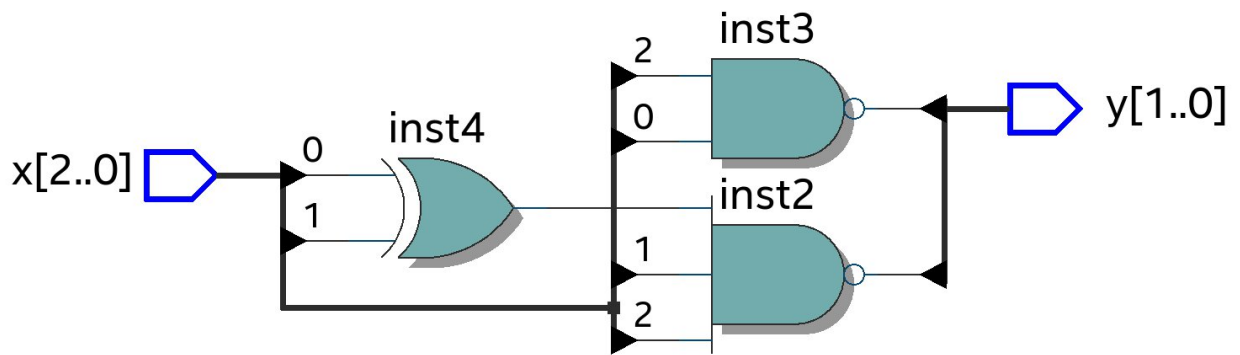
7. Произведите компиляцию проекта выбрав **Processing > Start Compilation**, либо нажав на  панели иконок. В процессе

компиляции в области задач будут отображаться выполняемые операции. Если всё прошло хорошо, то в журнале сообщений не должно быть ошибок, однако будет порядка 14 предупреждений, что на текущем этапе не мешает работе.

8. После компиляции откроется окно с отчётом. Наиболее часто используемая информация из отчёта – это используемые ресурсы ПЛИС, то есть количество логических элементов, регистров, количество пинов, объёма памяти, количество умножителей и прочее. Обратите внимание, что для используемого в описании примера схема состоит из 4 логических элементов, а в отчёте указано только 2.

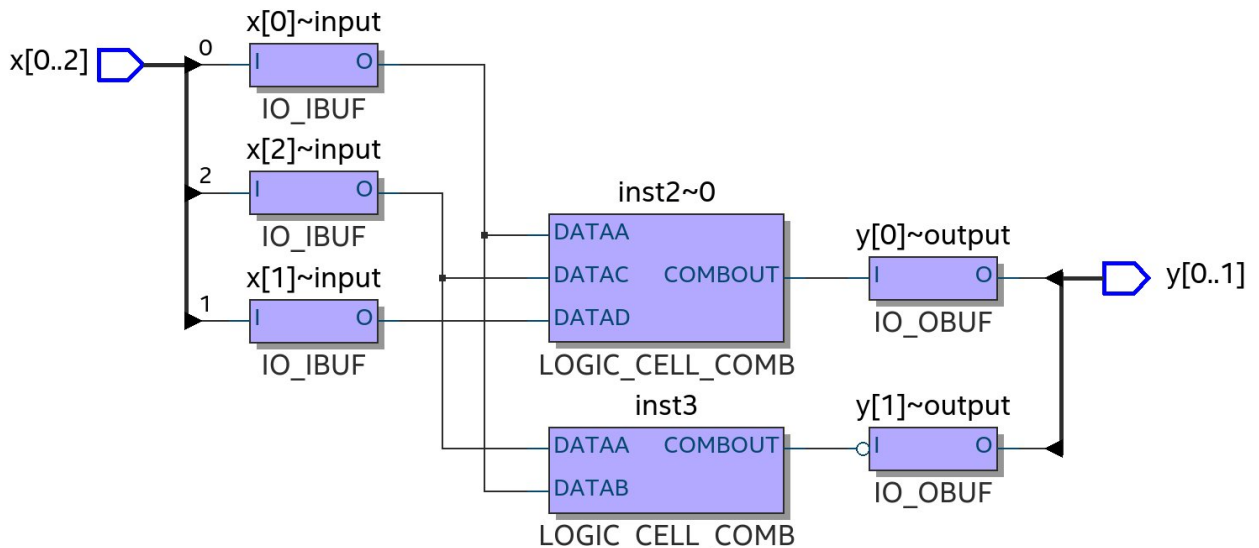
Flow Summary	
Flow Status	Successful - Wed Sep 7 09:48:44 2022
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	lab1
Top-level Entity Name	lab1
Family	Cyclone IV E
Device	EP4CE6E22C8
Timing Models	Final
Total logic elements	2 / 6,272 (< 1 %)
Total registers	0
Total pins	5 / 92 (5 %)
Total virtual pins	0
Total memory bits	0 / 276,480 (0 %)
Embedded Multiplier 9-bit elements	0 / 30 (0 %)
Total PLLs	0 / 2 (0 %)

9. Для просмотра RTL-представления схемы откройте **Tools > Netlist Viewers > RTL Viewer**. Обсудите с преподавателем, возникшее несоответствие в количестве элементов.



RTL-представление схемы

10. Аналогично выберите **Tools > Netlist Viewers > Tecnology Map Viewer (Post-Mapping)**.



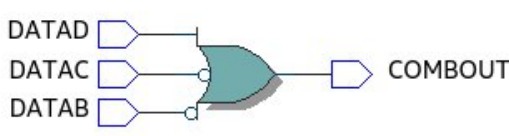
Использованные логические элементы ПЛИС

11. Проанализируйте каждую из LOGIC\_CELL\_COMB: правой кнопкой нажмите на логическую ячейку и в контекстном меню выберите **Properties**, затем в боковой панели выберите снизу вкладку **Truth Table**. Убедитесь, что таблица истинности совпадает с определённой Вами для каждого выхода у. Кроме того, на вкладке **Equation** можно увидеть логическое выражение, соответствующее данной таблице истинности (в нём знак # означает операцию ИЛИ, ! – инверсию, & – И). На графическом изображении сверху можно увидеть RTL-представление реализованной логической функции.

**Properties**

Block: |lab1|inst2~0

Lutmask: (FF3F)



inst2~0

	DATAD	DATAC	DATAB	OUT
1	0	0	0	1
2	0	0	1	1
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

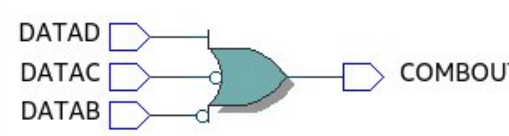
in-out Ports Truth Table Equation

Netlist Navigator Properties

**Properties**

Block: |lab1|inst2~0

Lutmask: (FF3F)



inst2~0

```

<c> inst2~0
= DATAD # !DATAC # !DATAB

DATAB = x[2]~input:O
DATAC = x[1]~input:O
DATAD = x[0]~input:O

```

in-out Ports Truth Table Equation

Netlist Navigator Properties

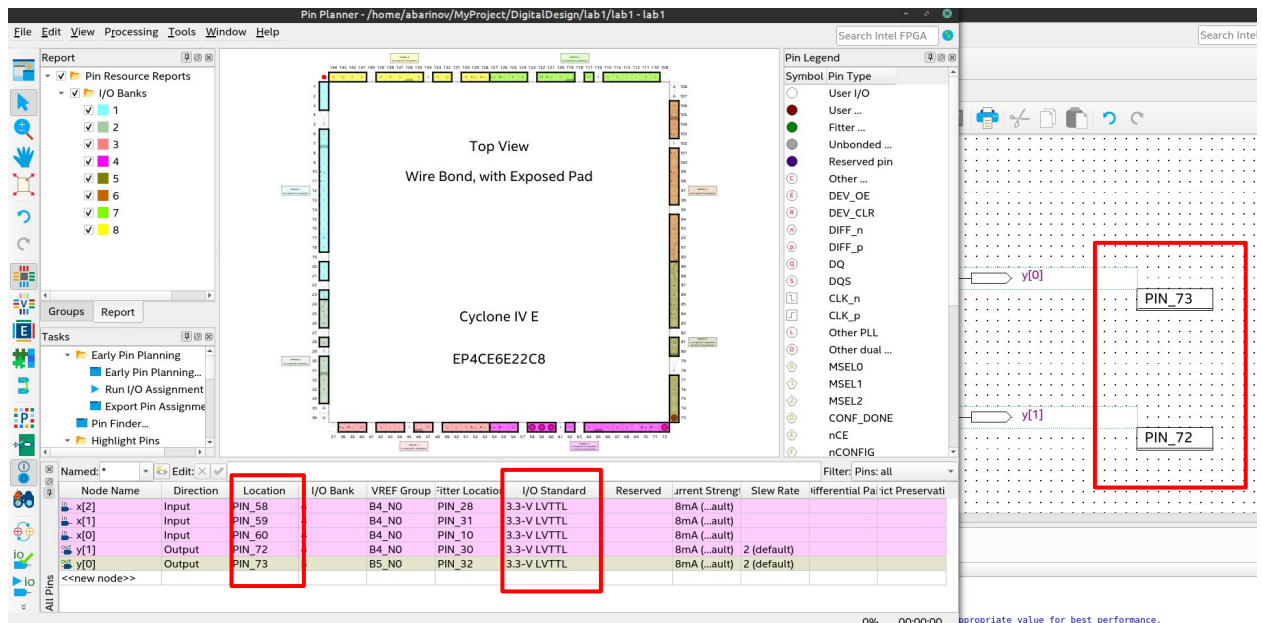
## Прошивка ПЛИС

1. Инструментом **Assignments > Pin Planner** откройте окно назначения пинов. Для входных и выходных пинов поставьте в соответствие пины с ПЛИС, выбрав нужный в столбце Location. Тип контакта (I/O Standart) укажите 3.3-V LVTTL для всех контактов. Для входных сигналов рекомендуется использовать DIP переключатели. Для выходных -

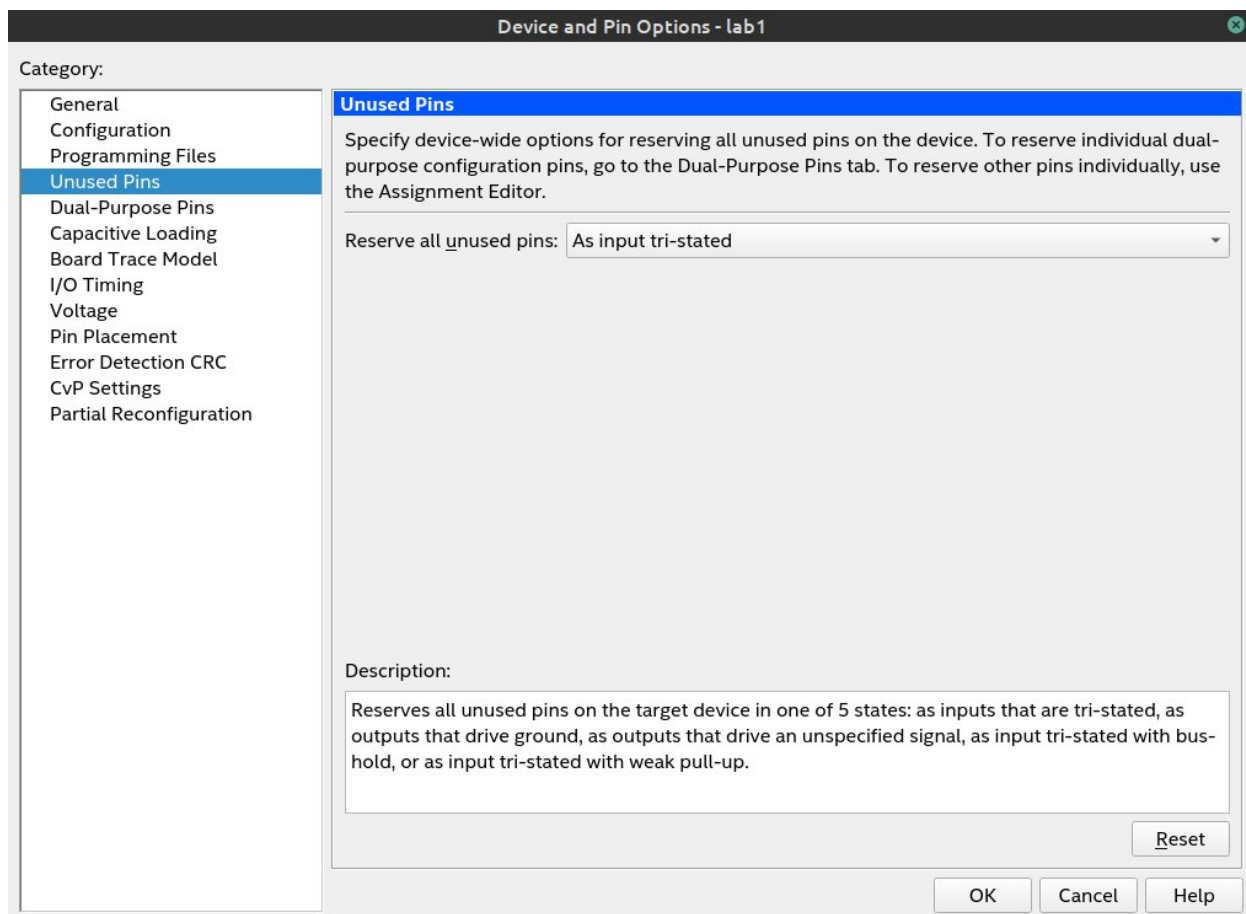


группу светодиодов. Соответствие пинов следует смотреть на отладочной плате. При этом на графическом представлении схемы будут появляться назначение пинов.

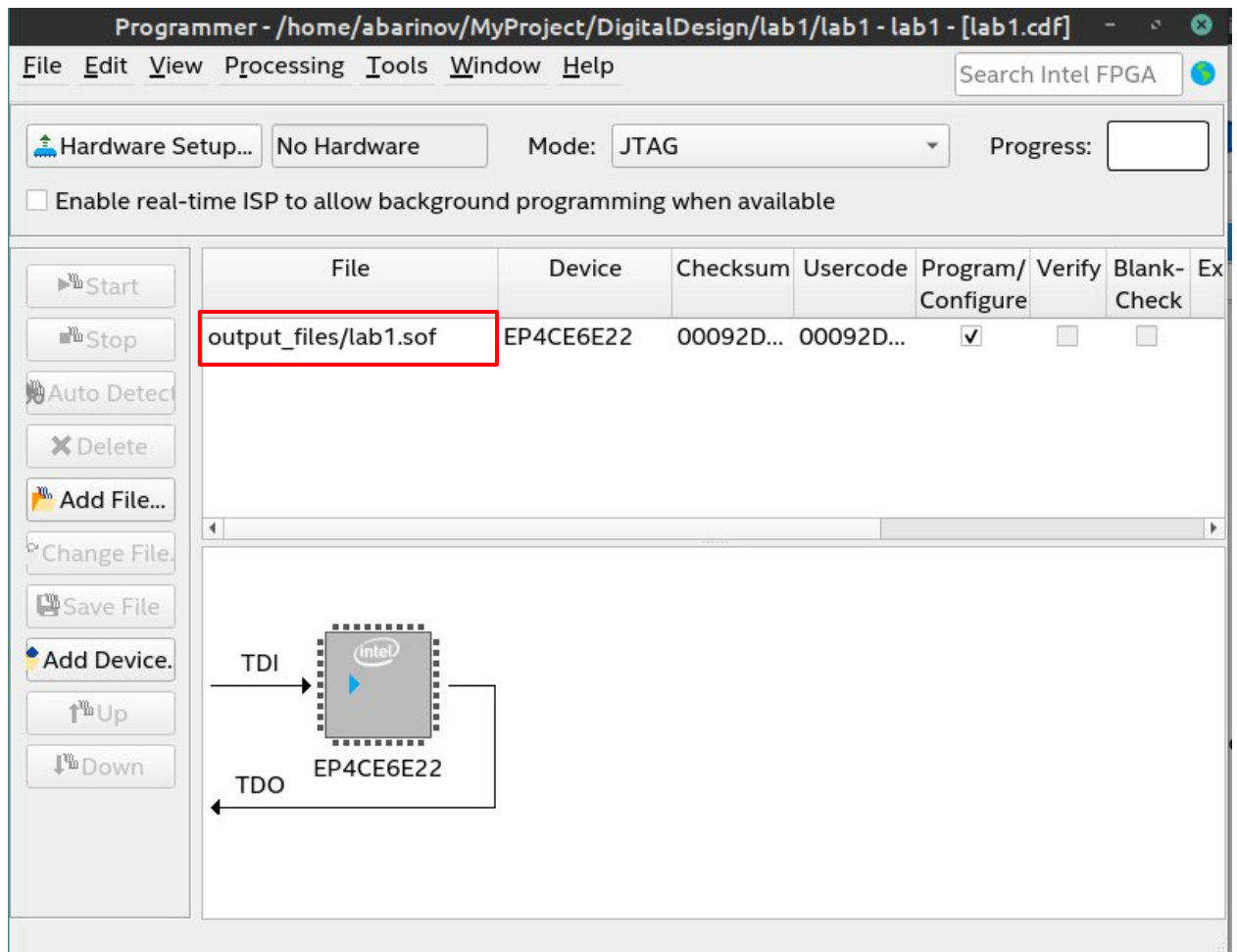
## 2. Закройте окно Pin Planner.



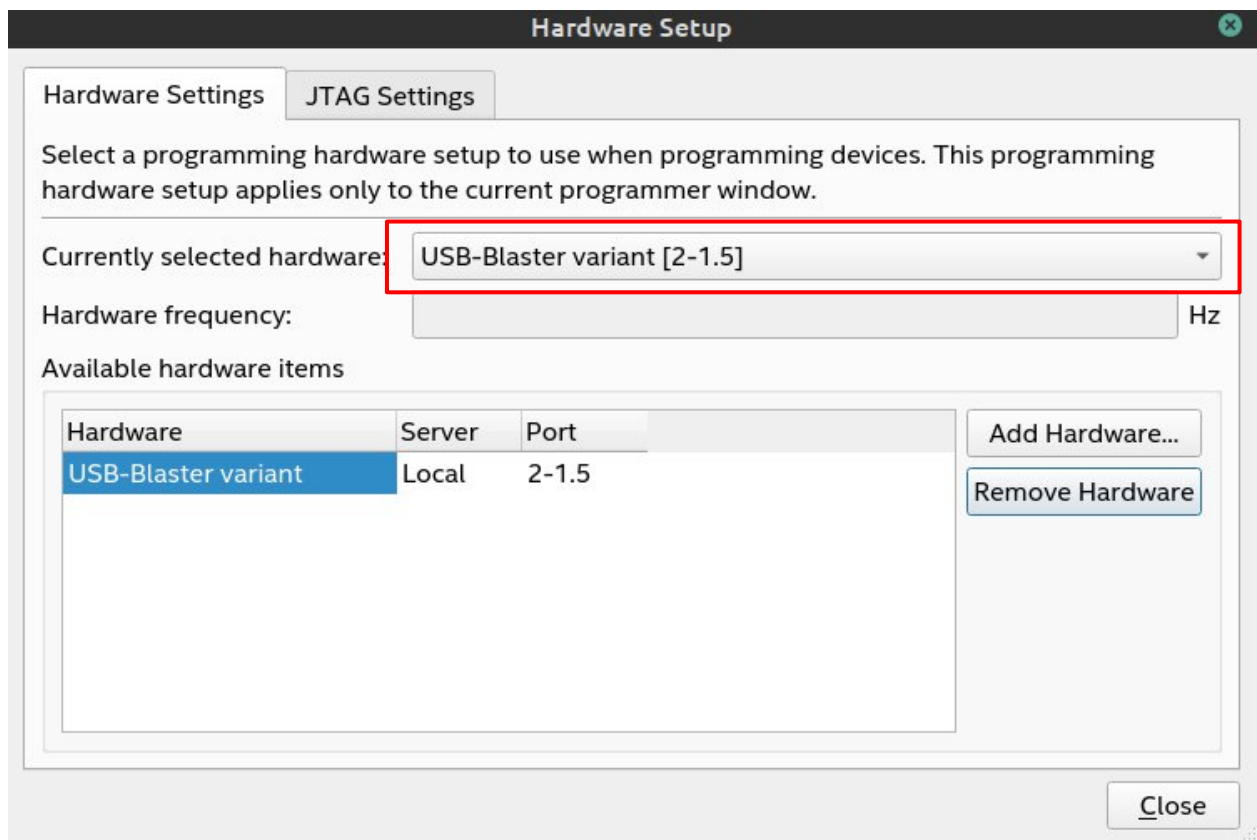
## 3. Инструментом **Assignments > Device** откройте окно настройки ПЛИС для перевода неиспользуемых пинов в третье состояние для избежания случайных коротких замыканий. В нём нажмите кнопку *Device and Pin Options...* В открывшемся окне выберите слева *Unused Pins* и в выпадающем меню укажите *As input tri-stated*. Нажмите *OK*, затем снова *OK*.



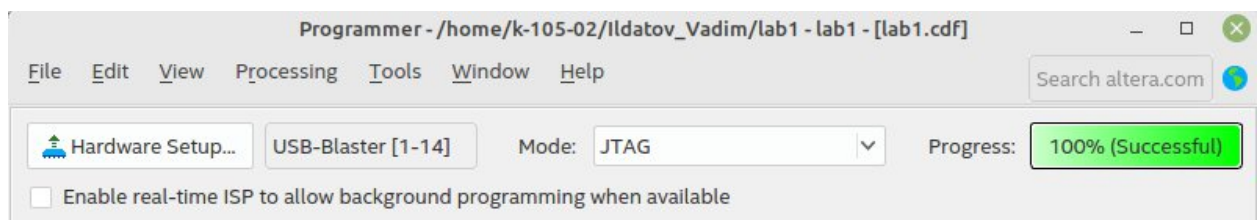
4. Подключите USB питание ПЛИС и программатор к компьютеру, а контакт программатора к контакту JTAG на ПЛИС и mini-USB к контакту mini USB на ПЛИС.
5. Снова произведите компиляцию проекта Processing > Start Compilation.
6. Инструментом **Tools > Programmer** откройте окно прошивки ПЛИС. В нём будет отображён файл прошивки **lab1.sof**, который сформировался в результате последней компиляции.



7. Если прошивки ещё не было, то необходимо указать программатор. Для этого нажмите в открывшемся окне на кнопку *Hardware Settings* и в выпадающем меню выберите USB Blaster. Нажмите *Close*.



8. Для прошивки нажмите *Start*. Если всё прошло хорошо, то в окне *Programmer* должно отобразиться, что процесс завершился удачей, и ПЛИС должна включить светодиоды и начать реагировать на переключатели. В противном случае обратитесь к преподавателю.



Следует отметить, что в зависимости от времени производства отладочной платы состоянию, когда светодиод загорается может соответствовать либо высокий уровень сигнала на выходе, либо низкий. Аналогично для переключателей: верхнему состоянию может соответствовать либо низкий уровень сигнала, либо высокий. Определяется экспериментально.

## Выполнение задания № 2

1. В том же проекте создайте новый файл **File > New....** Выберите *Verilog HDL File*. На рабочей части откроется файл **Verilog1.v**.
2. Опишите в нём модуль `lab1_struct`. Сформируйте в нём структурное описание. Сохраните файл в папку `/src`. Имя файла должно совпадать


с названием модуля **lab1\_struct.v**. Убедитесь, что файл стал отображаться в Project Navigator.

3. Аналогично пп. 1-2 сформируйте файл с описанием поведенческой модели **lab1\_beh**.

Используемые цифровые примитивы для структурного описания и логические выражения для поведенческого описания представлены в таблице 1.

Таблица 1 - Логические операции и вентили, используемые в Verilog-описании

Операция	Логическое выражение	Описание на вентильном уровне
НЕ	$y = \sim a$	<code>not(y, a);</code>
И	$y = a \& b$	<code>and(y, a, b);</code>
ИЛИ	$y = a   b$	<code>or(y, a, b);</code>
И-НЕ	$y = \sim (a \& b)$	<code>nand(y, a, b);</code>
ИЛИ-НЕ	$y = \sim (a   b)$	<code>nor(y, a, b);</code>
ИСКЛЮЧАЮЩЕЕ ИЛИ	$y = a \wedge b$	<code>xor(y, a, b);</code>
ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ	$y = \sim (a \& b)$	<code>xnor(y, a, b);</code>

4. В отличие от задания 1, в случае отсутствия необходимости прошивать ПЛИС и проводить временной анализ разрабатываемого устройства, вместо полной компиляции проекта достаточно произвести только первый этап анализа и синтеза. Для этого выберите в Project Navigator файл **lab1\_struct.v**, вызовите контекстное меню и выберите **Set as Top-Level Entity**. **Компиляция, обработка, анализ и синтез всегда проводятся для модуля верхнего уровня!** На панели иконок выберите **Start Analysis & Synthesis**  для формирования RTL-представления схемы.
5. По окончании в случае отсутствия ошибок проведите анализ RTL- и технологического представлений аналогично заданию 1. Если всё сделано верно, то три описания должны дать идентичные результаты.

При работе с HDL-кодом удобно использовать специальные редакторы с подсветкой синтаксиса (например, NotePad++ для ОС Windows или Kate, Xed и прочие для ОС GNU/Linux). В этом случае созданные файлы необходимо добавить в проект Quartus вручную, вызвав в Project Navigator контекстное меню у папки Files и выбрав **Add/Remove Files in Project....**

## Выполнение задания № 3

1. Сформируйте в проекте файл для тестирования **tb\_lab1.v**, расположите его в папке `/testbench`.