

# Упражнение 1

## Введение:

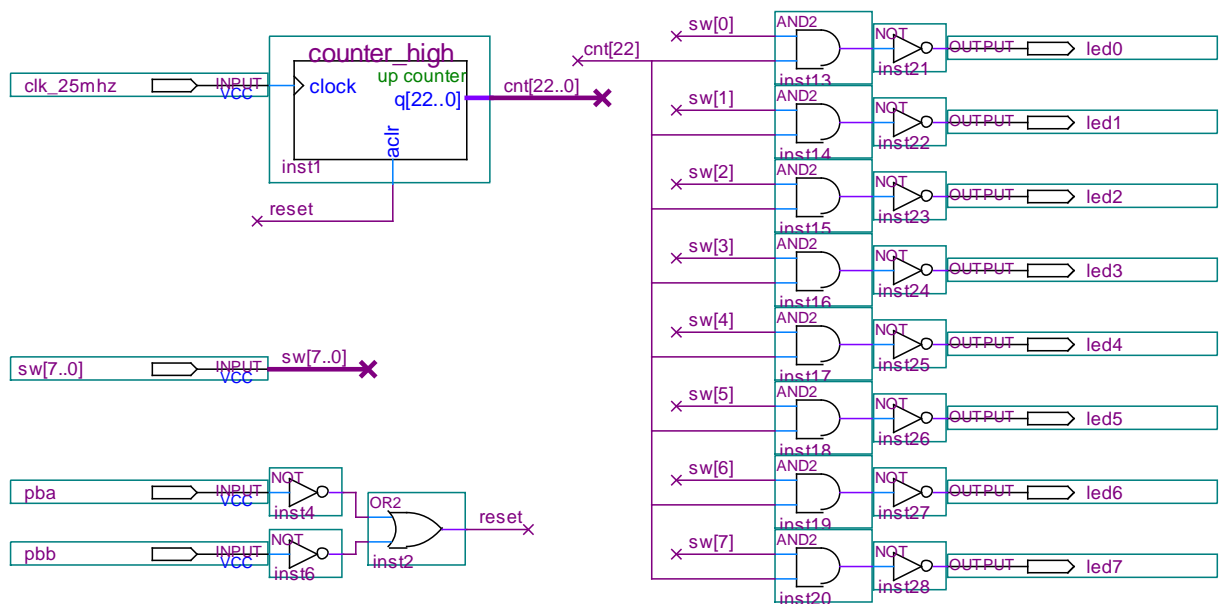
**Цель лабораторной работы** – самостоятельно пройти цикл проектирования в рамках пакета QuartusII, включая следующие этапы:

- ✓ [Создание проекта](#)
- ✓ [Анализ описания проекта \( ввод описания проекта уже выполнен для Вас\)](#)
- ✓ [Анализ назначения выводов СБИС \( опции компилятора и выходы СБИС для платы miniDiLaB-CIV уже назначены и хранятся в файле lab1.qsf\)](#)
- ✓ [Компиляция проекта](#)
- ✓ [Конфигурирование СБИС и проверка проекта на плате](#)

### Что необходимо иметь для выполнения лабораторной работы:

- ✓ Пакет QuartusII (достаточно бесплатной версии пакета, которую можно скачать с сайта фирмы Altera:  
[https://www.altera.com/support/software/download/altera\\_design/quartus\\_we/dnl-quartus\\_we.jsp](https://www.altera.com/support/software/download/altera_design/quartus_we/dnl-quartus_we.jsp) или запросить у официального дистрибьютора фирмы Altera в России – фирмы ЭФО: [www.efo.ru](http://www.efo.ru)
- ✓ Плата miniDiLaB-CIV ( <http://www.altera.ru/postavka-MiniDiLab-CIV.html> )

- ✓ **Рабочая папка проекта (lab1 ) содержит:**
- ✓ **схему верхнего уровня в иерархии проекта – файл lab1.bdf**



- ✓ **Файл с описанием конфигурируемого счетчика - counter\_high.tdf**
- ✓ **Файл с символом счетчика - counter\_high.bsf**
- ✓ **Файл с настройками проекта –lab1.qsf**

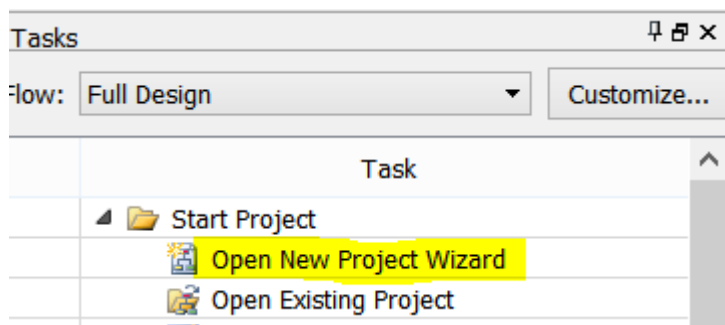
### Алгоритм работы проекта:

Данный проект обеспечивает:

- ✓ **деление на счетчике частоты 25Mhz на  $2^{23}$ ;**
- ✓ **включение (активный уровень – логический нуль)/выключение светодиодов старшим разрядом счетчика-делителя;**
- ✓ **разрешение (активный уровень – логическая единица)/запрещение работы каждого светодиода led[7..0] сигналом от соответствующего переключателя – sw[7..0];**
- ✓ **сброс счетчика при нажатии на кнопку pba или pbb.**

## Часть 1 – Создание проекта

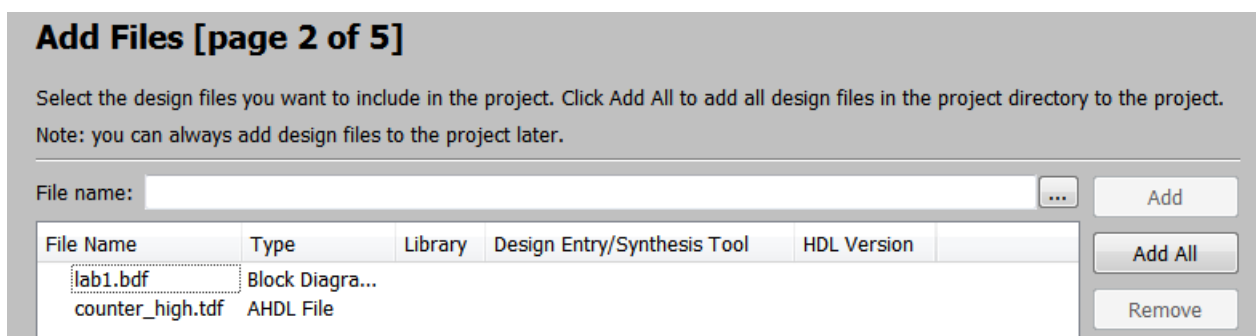
1. Запустите пакет QuartusII
2. В окне **Task** менеджера пакета, выберите **Open New Project Wizard**



3. На экране появится окно введения - **Introduction** (если оно небыло отключено). Нажмите кнопку **next**.
4. В появившемся окне введите следующие данные:

Раздел	Что ввести
<b>What is the working directory for this project?</b> Рабочая папка ( с помощью браузера найдите рабочую папку проекта)	... \ lab1
<b>What is the name of this project?</b> Имя проекта	lab1
<b>What is the name of the top-level design entity for this project?</b> Имя модуля верхнего уровня в иерархии проекта.	lab1

5. Нажмите кнопку **Next**.
6. В окне **Add Files [page 2 of 5]** нажмите кнопку **AddAll** – добавить к проекту все файлы, расположенные в рабочей папке. Затем кнопку **Next**.



7. В окне **Family & Device Setting[page3 of 5]**:

- в разделе **Family** выберите **CycloneIVE**.
- в разделе **Filters**:
  - в графе **Package** выберите **TQFP**,

- в графе **Pin count** выберите **144**,
- в графе **Speed grade** выберите **8**.
- в разделе **Available devices** выберите СБИС **EP4C6E22C8**.

### Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family

Family: Cyclone IV E

Devices: 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: TQFP

Pin count: 144

Speed grade: 8

☒ Show advanced devices

☐ HardCopy compatible only

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit eleme
EP4CE6E22C8	1.2V	6272	92	276480	30


8. Нажмите кнопку **Next**.
9. В окне **EDA Tool Setting [page 4 of 5]** оставьте все без изменения и нажмите кнопку **Next**.
10. Появится окно **Summary [page 5 of 5]**, в котором указаны установки, заданные Вами для создаваемого проекта. Проверьте их. Если все правильно, то нажмите кнопку **Finish**. В противном случае, вернитесь назад, нажав (возможно несколько раз) кнопку **Back**.

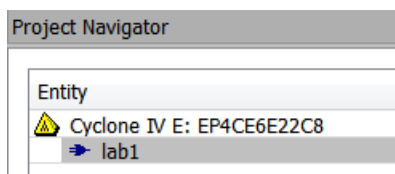
*Проект создан.*

## Часть 2 – Анализ описания проекта

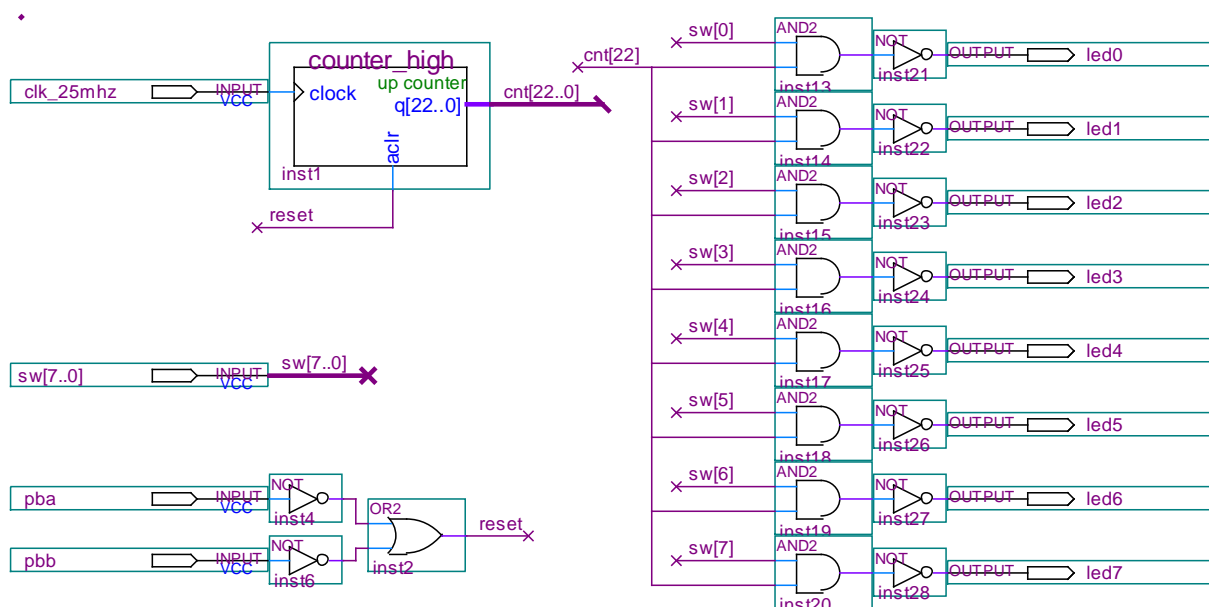
Реализуемый проект иерархический, его описание содержит модуль верхнего уровня иерархии, созданный в схемном редакторе пакета, и модуль нижнего уровня иерархии – счетчик-делитель, созданный с использованием помощника “MegaWizard Plug-In Manager”

1. Откройте модуль верхнего уровня иерархии, представленный в файле lab1.bdf:

- На панели инструментов пакета с помощью иконки  включите (если он не был включен) навигатор проекта - **Project Navigator**
- в окне иерархического отображения проекта – **Project Navigator**, переключитесь на закладку **Hierarchy** и дважды щелкните левой клавишей мыши в поле **lab1**



Откроется окно схемного редактора пакета, в котором будет представлена схема модуля верхнего уровня в иерархии проекта, приведенная ниже.



## Часть 3 – Анализ назначения выводов СБИС

Назначение опций компилятора (практически все опции имеют значения по умолчанию) и номеров выводов СБИС для платы miniDiLaB-CIV выполнены для Вас и хранятся в файле **lab1.qsf**

Раздел Pin & Location Assignments файла, содержащий привязку выводов проекта к выводам микросхемы EP4C6E22C8, приведен ниже:

```
# Pin & Location Assignments
# =====
set_location_assignment PIN_64 -to pba
set_location_assignment PIN_58 -to pbb
set_location_assignment PIN_72 -to led0
set_location_assignment PIN_71 -to led1
set_location_assignment PIN_70 -to led2
set_location_assignment PIN_69 -to led3
set_location_assignment PIN_68 -to led4
set_location_assignment PIN_67 -to led5
set_location_assignment PIN_66 -to led6
set_location_assignment PIN_65 -to led7
set_location_assignment PIN_88 -to sw[7]
set_location_assignment PIN_89 -to sw[6]
set_location_assignment PIN_90 -to sw[5]
set_location_assignment PIN_91 -to sw[4]
set_location_assignment PIN_49 -to sw[3]
set_location_assignment PIN_46 -to sw[2]
set_location_assignment PIN_25 -to sw[1]
set_location_assignment PIN_24 -to sw[0]
set_location_assignment PIN_23 -to clk_25mhz
```

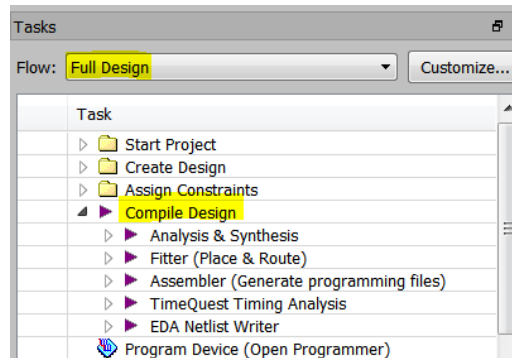
Часть раздела Fitter Assignments, в которой задаются:

- режим работы не использованных выводов СБИС: как входы с pull-up резистором
  - стандарт сигнала для каждого входа/выхода
- приведена ниже:

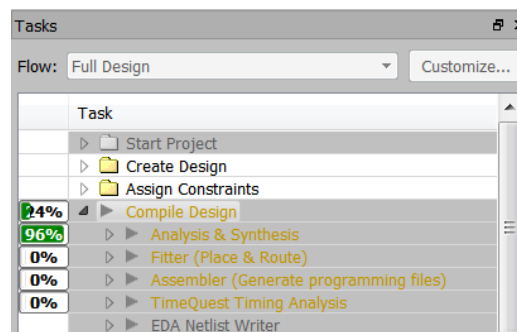
```
set_global_assignment -name RESERVE_ALL_UNUSED_PINS_WEAK_PULLUP "AS
INPUT TRI-STATED"
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to clk_25mhz
set_instance_assignment -name IO_STANDARD "2.5 V" -to led0
set_instance_assignment -name IO_STANDARD "2.5 V" -to led1
set_instance_assignment -name IO_STANDARD "2.5 V" -to led2
set_instance_assignment -name IO_STANDARD "2.5 V" -to led3
set_instance_assignment -name IO_STANDARD "2.5 V" -to led4
set_instance_assignment -name IO_STANDARD "2.5 V" -to led5
set_instance_assignment -name IO_STANDARD "2.5 V" -to led6
set_instance_assignment -name IO_STANDARD "2.5 V" -to led7
set_instance_assignment -name IO_STANDARD "2.5 V" -to pba
set_instance_assignment -name IO_STANDARD "2.5 V" -to pbb
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to sw[7]
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to sw[6]
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to sw[5]
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to sw[4]
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to sw[3]
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to sw[2]
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to sw[1]
set_instance_assignment -name IO_STANDARD "3.3-V LVTTL" -to sw[0]
set_global_assignment -name STRATIX_DEVICE_IO_STANDARD "2.5 V"
```

## Часть 4 – Компиляция проекта

1. В окне задач (Tasks) выберите процедуру Full Design и двойным щелчком левой клавиши мыши по команде Compile Design запустите полную компиляцию проекта.



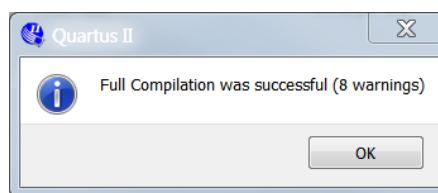
2. Окно задач (Tasks) будет отображать ход выполнения процедуры компиляции.



*В процессе полной компиляции проекта осуществляется:*

- ✓ проверка синтаксиса,
- ✓ синтез с оптимизацией занимаемой площади и быстродействия проекта,
- ✓ трассировка и СБИС с оптимизацией занимаемой площади и быстродействия проекта,
- ✓ получение файла для конфигурирования СБИС – *prof(sof)* файл,
- ✓ получение модели с временными параметрами реализованной СБИС,
- ✓ временной анализ,
- ✓ Формирования файла с детальным отчетом о всех этапах компиляции проекта.

3. При успешном завершении процедуры компиляции Вы получите сообщение (возможно наличие нескольких предупреждений):



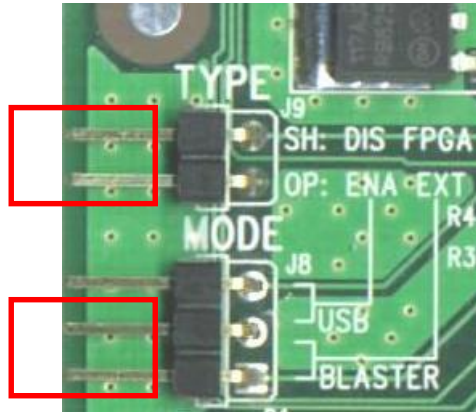
Нажмите кнопку ОК.

***Процедура компиляции завершена.***

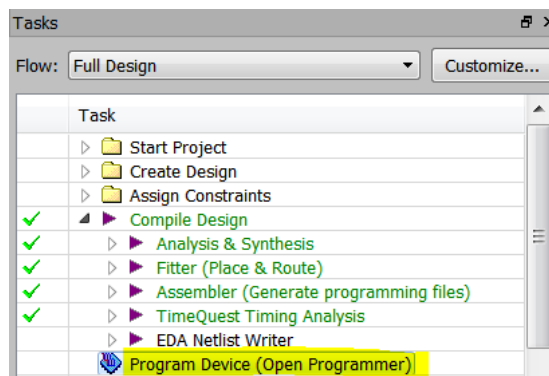


## Часть 5 – Конфигурирование СБИС и проверка проекта на плате

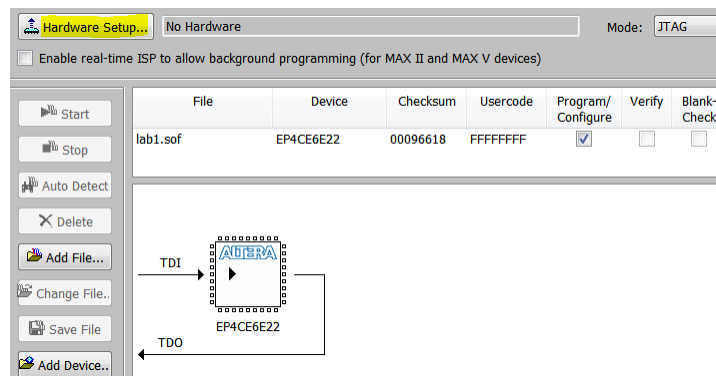
1. На плате miniDiLaB-CIV установите джамперы следующим образом:
  - a. Соедините выводы разъема “TYPE”
  - b. Соедините выводы 1-2 разъема “MODE”



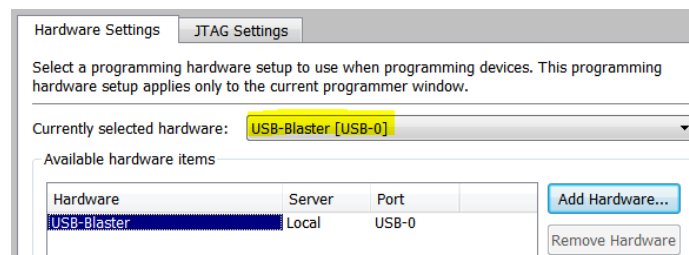
2. Подсоедините входящий в комплект поставки USB кабель A-miniB к USB 2.0 порту компьютера (должен обеспечивать ток до 500мА), а затем к плате miniDiLaB-CIV.
3. Включите плату miniDiLaB-CIV : переключатель Power
4. В окне задач (Tasks) выберите процедуру Full Design и двойным щелчком левой клавиши мыши по команде Program Device запустите приложение, управляющее конфигурированием СБИС.



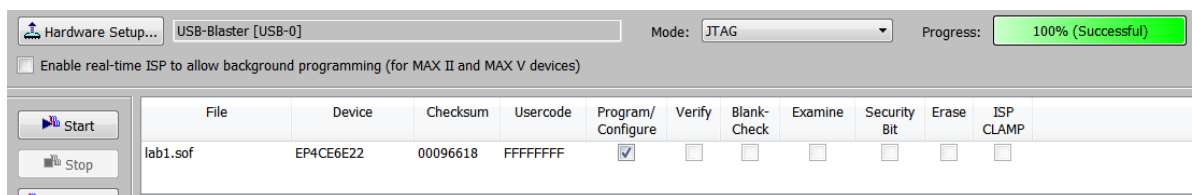
5. Откроется окно управления конфигурированием СБИС.
6. Для установки интегрированного на плату miniDiLaB-CIV средства конфигурирования СБИС нажмите кнопку **Hardware Setup**, откроется окно настроек.



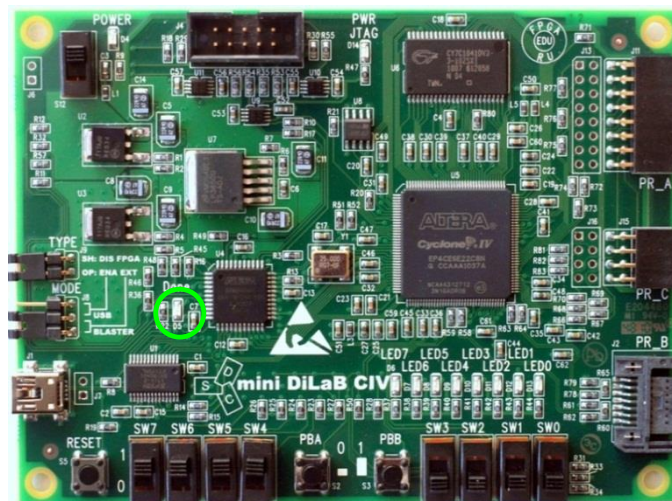
7. В разделе **Available hardware items** выберите (двойным щелчком левой клавиши мыши) **USB-Blaster**. Нажмите кнопку **Close**.



8. Включите опцию **Program/Configure** и нажмите кнопку **Start**. В окне Progress будет отображаться статус процедуры конфигурирования СБИС.



9. Когда СБИС будет запрограммирована на плате miniDiLaB-CIV загорится зеленый светодиод – “Done”.



10. Светодиоды led[7..0], для которых разрешена работа (соответствующие переключатели sw[7..0] в положении “1”) будут мигать с частотой ~ 3 раза в секунду, остальные светодиоды – выключены. При нажатии на кнопку rba или rbb все светодиоды выключаются.
11. Выключите и включите питание платы. Обратите внимание на то, что конфигурация СБИС ПЛ отсутствует.

12. Создайте jic файл ( EPCS16 – микросхема для хранения конфигурации):

**File=>Convert Programming Files**

Output programming file

Programming file type: JTAG Indirect Configuration File (.jic)

Options... Configuration device: EPCS16 Mode: Active Serial

File name: output\_file.jic

Advanced... Remote/Local update difference file: NONE

☒ Create Memory Map File (Generate output\_file.map)

☐ Create CvP files (Generate output\_file.periph.jic and output\_file.core.rbf)

Input files to convert

File/Data area	Properties	Start Address
Flash Loader		
EP4CE6		
SOF Data	Page_0	<auto>
lab1.sof	EP4CE6E22	

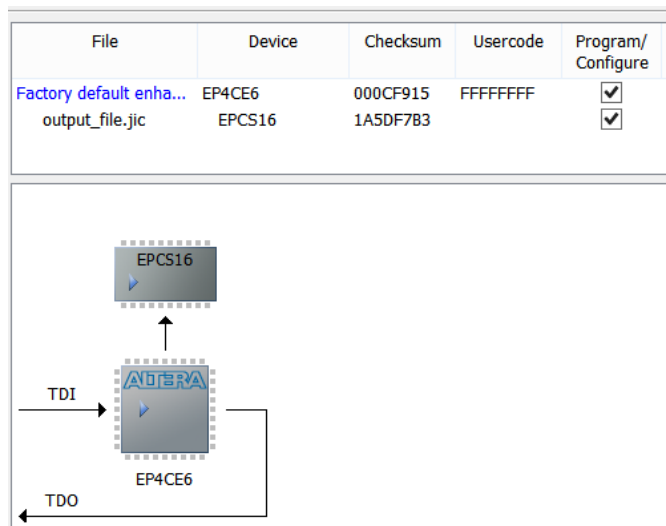
Add Hex Data

Add Sof Page

Add File...

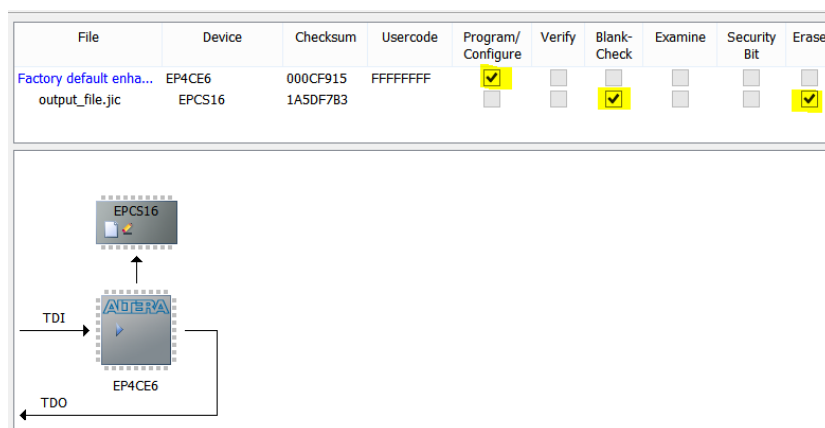
Remove

13. Осуществите программирование микросхемы EPSC с помощью созданного jic файла.



14. Выключите и включите притание платы. Обратите внимание на то, что СБИС ПЛ автоматически сконфигурирована.

15. Осуществите стирание содержимого микросхемы EPCS16:



16. Выключите и включите притание платы. Обратите внимание на то, что СБИС ПЛ не была сконфигурирована (светодиод Done не горит).

*Лабораторная работа завершена.*