

Инструкции по подготовке компьютера(ноутбука) для занятий в «Сколковской Школе синтеза цифровых схем»

Содержание

1. Установка и настройка ПО для работы на ОС Windows 10/11	2
1.1 Установка Quartus 21.1 Lite	2
1.2 Установка драйвера для usb-blaster	3
1.3 Проверка правильности установки Quartus.....	6
1.4 Установка Icarus Verilog и gtkwave на Windows	9
2. Установка и настройка ПО для работы на ОС на базе linux (ubuntu 22.04)	11
2.1 Установка Quartus 21.1 Lite	11
2.2 Установка драйвера для usb-blaster	12
2.3 Проверка правильности установки Quartus.....	13
2.4 Установка icarus Verilog и gtkwave на Linux.....	17
3. Установка и настройка ПО для работы с OpenLane.....	17
Дополнительная информация	21

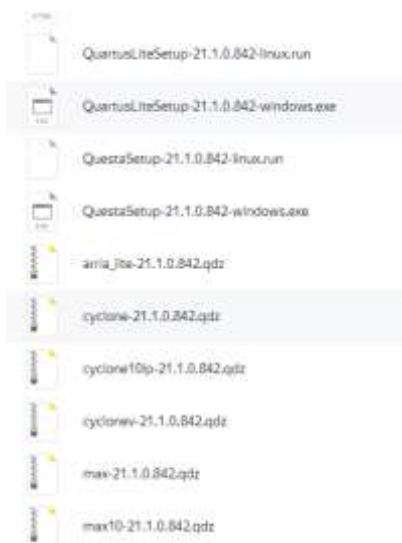
1. Установка и настройка ПО для работы на ОС Windows 10/11

1.1 Установка Quartus 21.1 Lite

Для установки Quartus воспользуйтесь ссылкой на Яндекс Диск <https://disk.yandex.ru/d/CSn1xlo5QHj-HA>, которая содержит установщики и архивы для поддержки ПЛИС серии Cyclone IV.

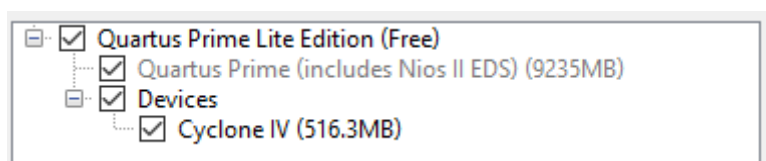
Для установки на ОС Windows 10/11 понадобятся следующие файлы¹:

- Установщик QuartusLiteSetup-21.1.exe
- Архив cyclone-21.1.qdz для поддержки ПЛИС серии cyclone IV используемых в лабораторных работах Школы. (Скриншоты с версией 21.1.0.842, установка не отличается)²



ВАЖНО: Директория установки Quartus не должна содержать символов кириллицы, это может привести к сбоям в его работе.

Так же после выбора директории для установки у вас должно появиться окно выбора установки доп.файлов для ПЛИС серии Cyclone IV:



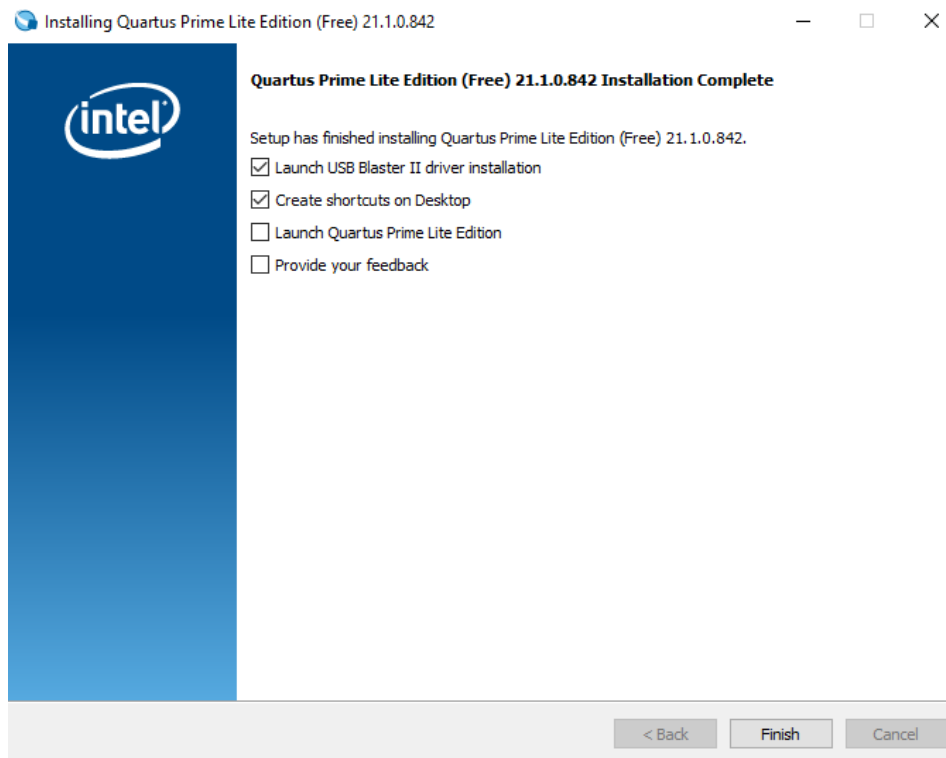
Если этого не произошло, вероятно вы не скачали .qdz архив, или поместили его не в папку с установщиком. Позднее вы сможете установить его с помощью Tools- Install Devices в Quartus

¹ Если у вас стоит приложение Я.Диска, то вы можете сохранить репозиторий в ваш личный диск и через приложение скачать всю папку. Если приложения нет, то нужно скачивать эти файлы по отдельности. Так же обратите внимание на то, что при установке .qdz и .exe файлы необходимо поместить **в одну папку**.

² Если вам потребуется поддержка других серий ПЛИС в своих проектах или необходим симулятор Questa, вы так же можете найти дополнительные файлы для установки на данном диске (Для бесплатной версии Questa требуется лицензия). Например, для поддержки платы De10-Lite, нужен архив max10.qdz.

Quartus займёт не менее 10 ГБ на вашем ЖД.

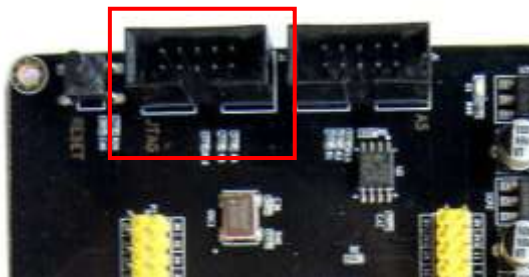
В конце установки так же необходимо установить *USB Blaster II Driver*:



1.2 Установка драйвера для usb-blaster

Для работы с платой необходимо установить драйвер для usb-blaster.

1. Подключите программатор (пластиковый корпус с надписью usb-blaster). Кабель JTAG подключите одним концом к разъёму JTAG (!не AS) на плате OMDAZZ/rzrd (см. рис ниже), а другим – к программатору.



А USB-кабель - от программатора к ПК:



2. Подключите питание через USB B-TYPE (провод в комплекте) к разъёму (плата OMDAZZ/rzrd) (на рис. – правый) и включите плату (кнопка рядом с разъёмом).

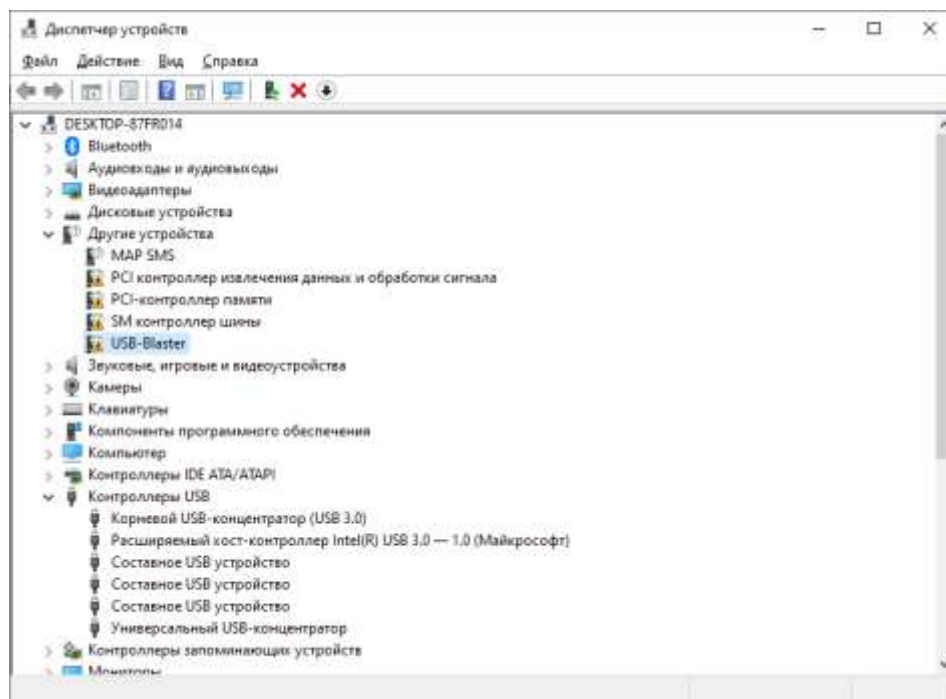


3. Готовая к работе плата должна выглядеть так как на рисунке:



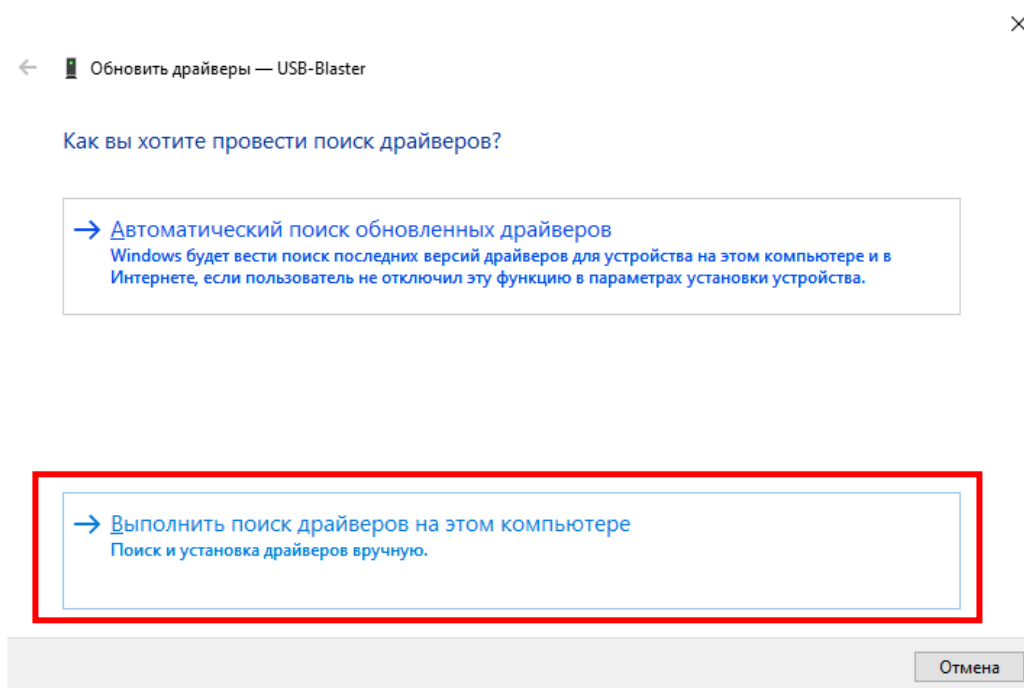
Далее, необходимо зайти в Диспетчер устройств ПК. Самый быстрый способ его найти – с помощью поиска возле кнопки «Пуск». После подключения программатора к ПК у вас должен появиться следующий пункт в диспетчере устройств³:

³ Если он не появится, попробуйте поменять USB-разъем (с 3.0 на 2.0), к которому вы подключали программатор. Не работать может несколько, так что попробуйте все. Также лишний раз проверьте корректность подключения JTAG, работоспособность программатора (на нем должен гореть светодиод) и работоспособность самой платы (Если горят все светодиоды на семисегментном индикаторе одновременно, это говорит о том, что плата неисправна). Если ничего из этого не помогло, обратитесь к модераторам Школы.

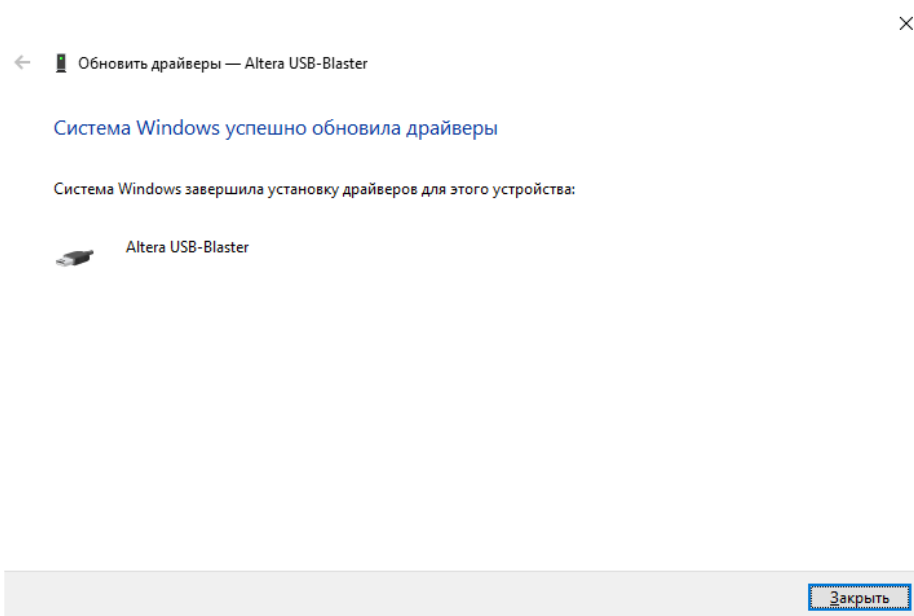
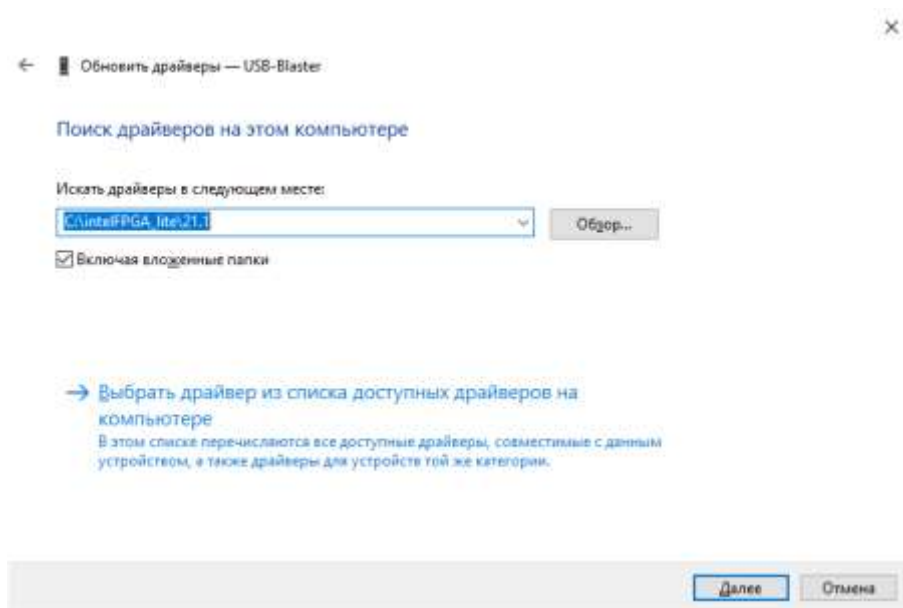


4. Для установки драйвера необходимо нажать *ПКМ* -> *Обновить драйвер*. Во всплывающем окне выбрать опцию «Найти драйверы на этом компьютере» (см. рис ниже)

5. Для поиска нужно выбрать папку установки Quartus 21.1 для поиска драйвера и установить его.⁴



⁴ В корневой папке Quartus содержатся драйвера для usb-blaster и usb-blaster-ii. Если драйвер не установится, обратитесь к модераторам.

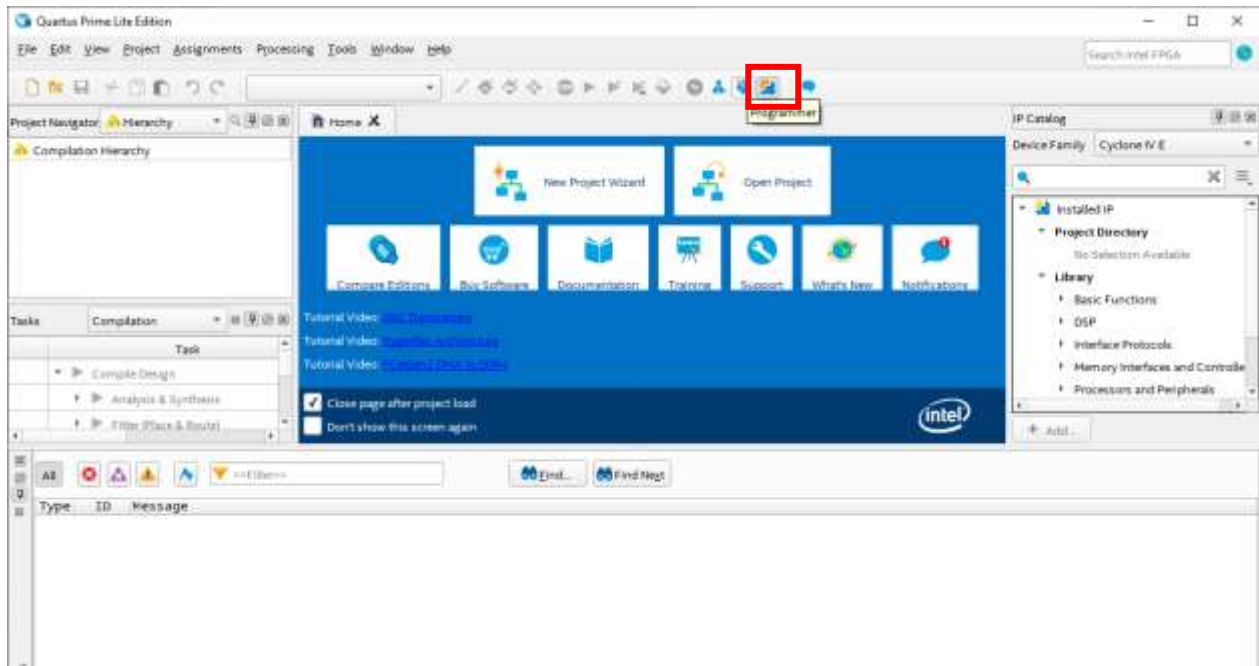


1.3 Проверка правильности установки Quartus

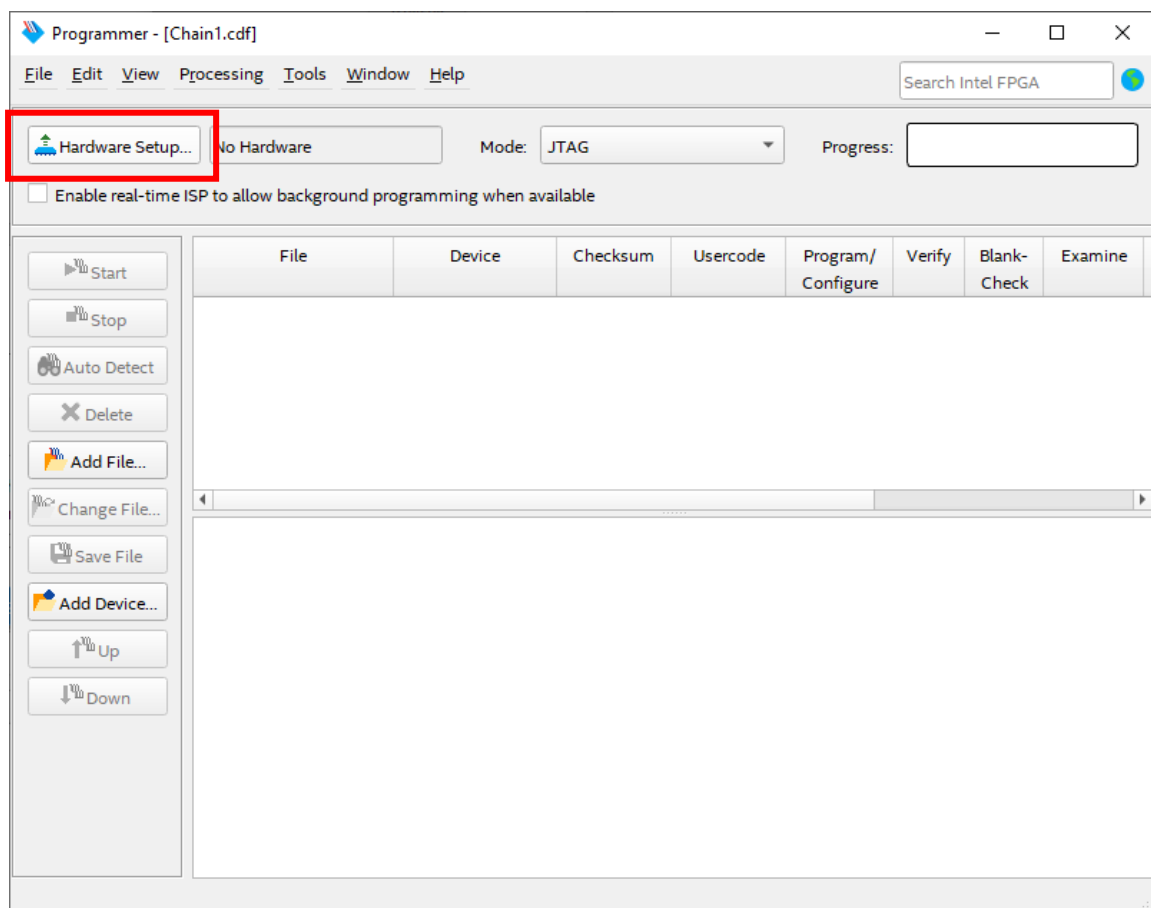
1. При запуске Quartus должно появиться такое окно, запускаем через *Run the Quartus Prime Software*:



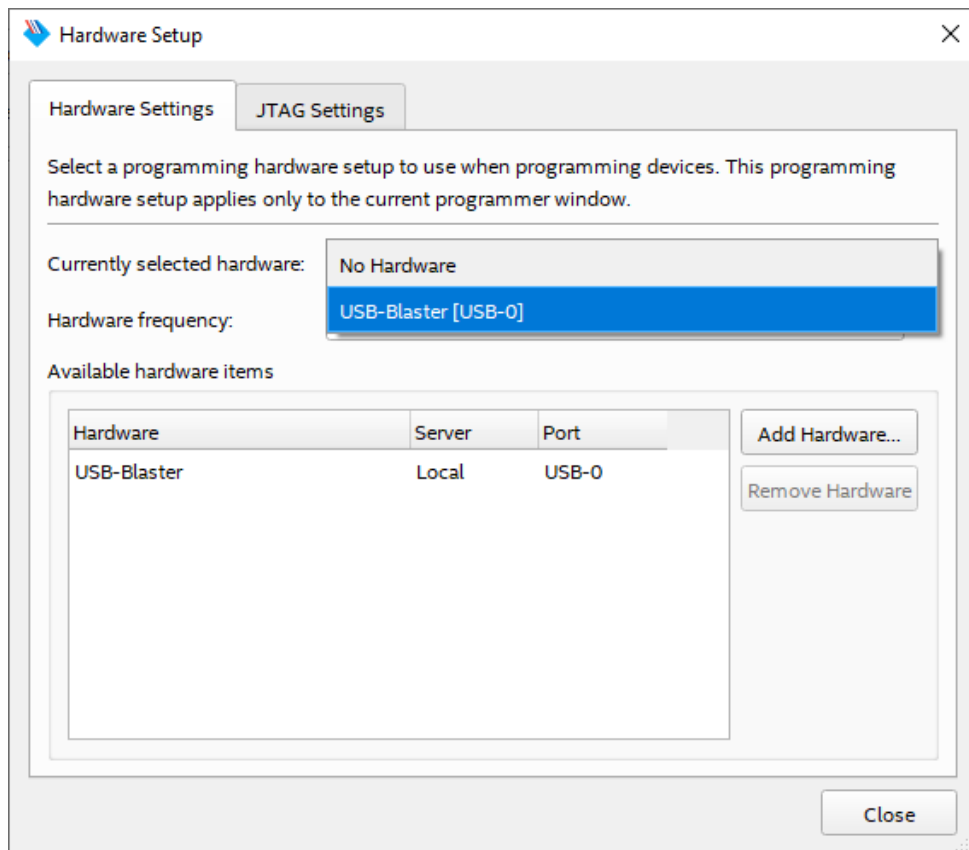
2. Если у вас есть плата открываем окно *programmer*:



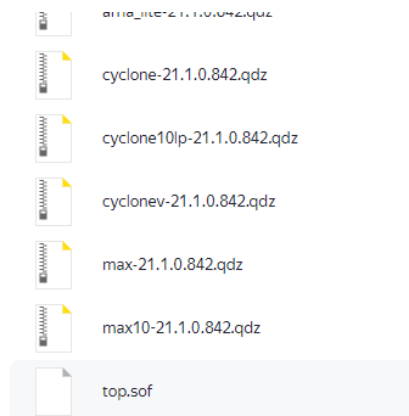
3. Нажимаем *Hardware setup*:

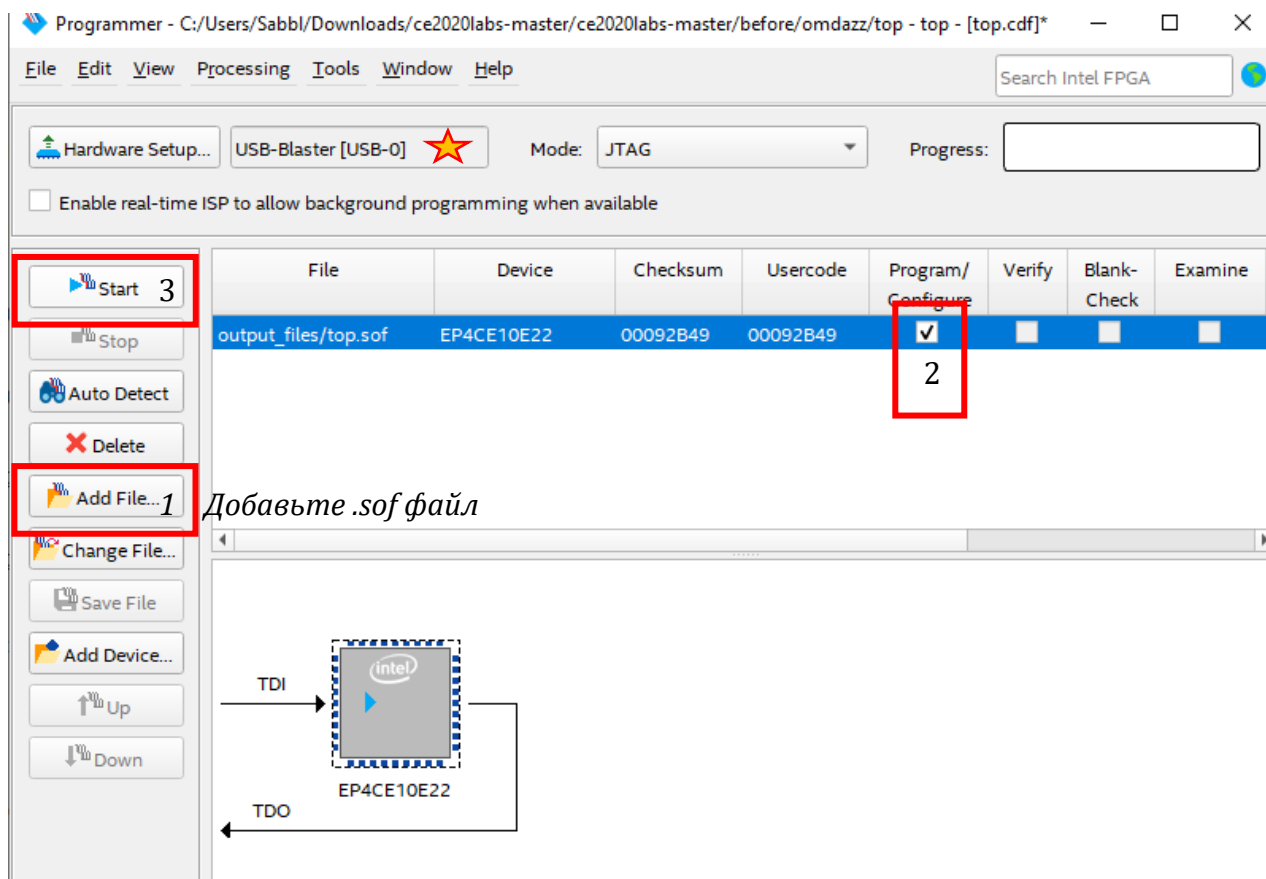


4. Если при выборе устройства в выпадающем списке присутствует USB-Blaster установку драйвера можно считать успешно завершенной:

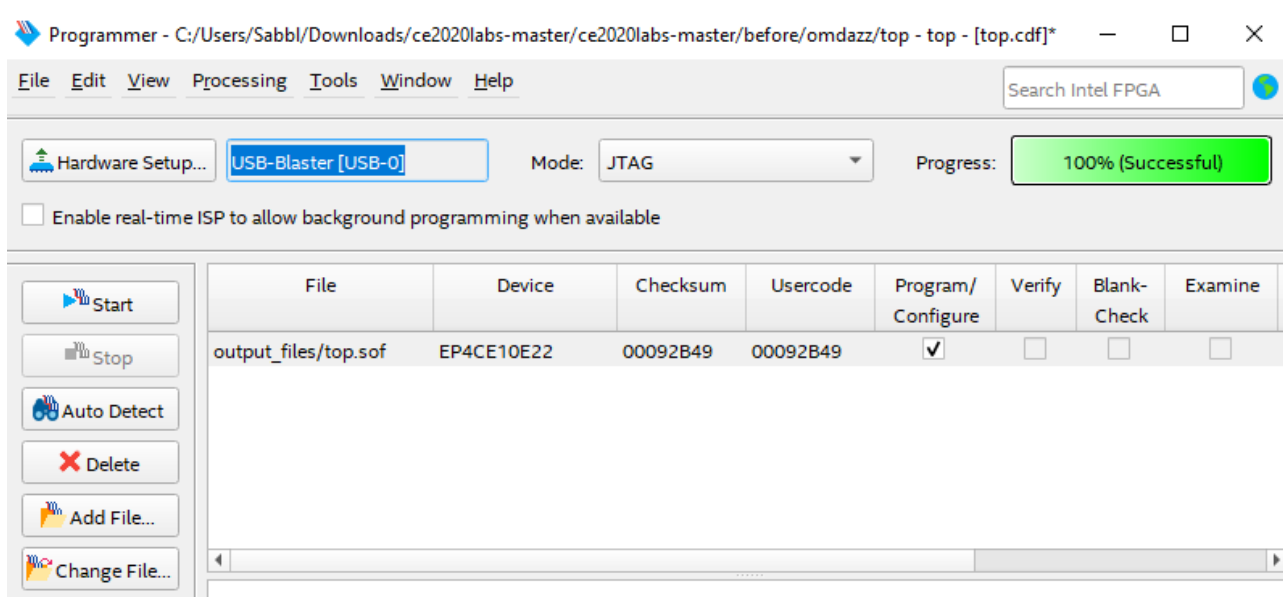


5. Загрузите файл прошивки top.sof с Yandex диска и попробуйте загрузить его на плату OMDAZZ или rzrd с помощью того же окна programmer через *Add File* (Не забудьте про **Hardware setup -> USB-Blaster**):





Если вы увидели надпись **Successful**, установку и проверку работы Quartus можно завершить:



1.4 Установка Icarus Verilog и gtkwave на Windows

1. Перейдите на сайт с Icarus Verilog: <https://bleyer.org/icarus/> и версию 11:

Icarus Verilog for Windows

Icarus Verilog is a free compiler implementation for the IEEE-1364 Verilog hardware description language. Icarus is maintained by the Icarus Verilog project. In this page you will find easy to install Icarus Verilog packages compiled with the MinGW toolchain for the Windows environment.

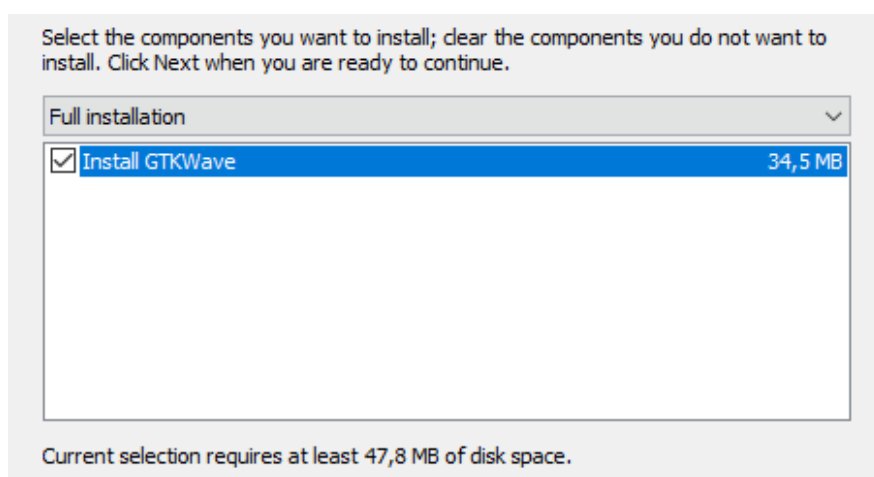
Download

You can find Icarus Verilog sources and binaries for most platforms at the [Icarus site FTP](#). The sources available here have been tested and are ready to use.

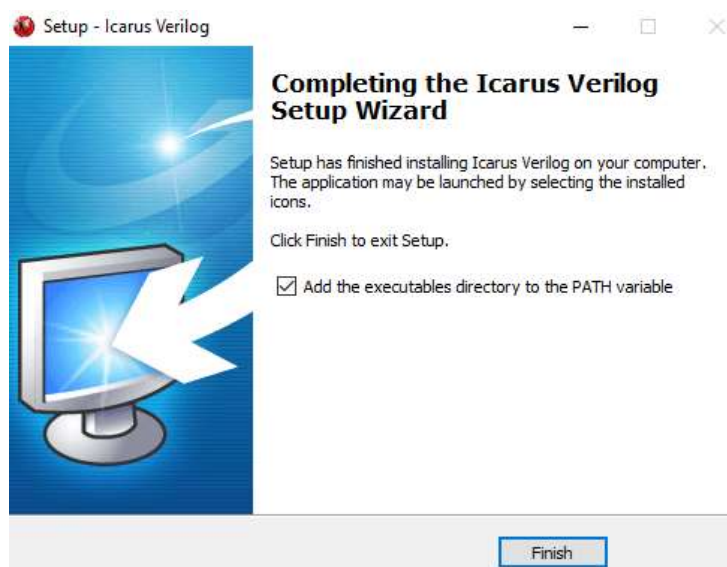
- [iverilog-v12-20220611-x64_setup \[18.2MB\]](#)
- [iverilog-v11-20210204-x64_setup.exe \[44.1MB\]](#)
- [iverilog-v11-20201123-x64_setup.exe \[18.1MB\]](#)
- [iverilog-10.1.1-x64_setup.exe \[9.77MB\]](#)
- [iverilog-10.0-x86_setup.exe \[11.1MB\]](#)
- [iverilog-20130827_setup.exe \(development snapshot\) \[11.2MB\]](#)
- [iverilog-0.9.7_setup.exe \(latest stable release\) \[10.5MB\]](#)
- [iverilog-0.9.6_setup.exe \[10.4MB\]](#)
- [iverilog-0.8.6_setup.exe \(latest release 0.8 series\) \[1.29MB\]](#) [iverilog-0.8.6.7z \[800kB\]](#)
- [iverilog-0.7-20040706_setup.exe \[1.09MB\]](#) [iverilog-0.7-20040706.7z \[588kB\]](#)

2. Активируйте установщик и следуйте инструкциям.

3. Дополнительно установите GTKWave:



4. После установки добавьте переменную среды:



Если возникнут доп. вопросы: <https://www.youtube.com/watch?v=5Kync4z5VOw> – видео по установке на Windows.

2. Установка и настройка ПО для работы на ОС на базе linux (lubuntu 22.04)

2.1 Установка Quartus 21.1 Lite

Данный способ проверялся на дистрибутиве **Lubuntu 22.04**.

Аналогичным образом установка будет работать для **Ubuntu** последних версий.

1. Для установки Quartus на Linux можно воспользоваться той же ссылкой на Яндекс Диск : <https://disk.yandex.ru/d/CSn1xlo5QHj-HA>.

Для установки на дистрибутив lubuntu 22.04 вам понадобятся следующие файлы: Установщик QuartusLiteSetup-21.1-linux.run и архив cyclone-21.1.qdz для поддержки ПЛИС серии cyclone IV используемых в лабораторных работах Школы. (Версии на скриншотах могут отличаться)⁵

QuartusLiteSetup-21.1.0.842-linux.run	07.08.2022	11 KB	1.88 TB
QuartusLiteSetup-21.1.0.842-windows.exe	07.08.2022	11 KB	1.81 TB
QuartaSetup-21.1.0.842-linux.run	07.08.2022	11 KB	1.82 TB
QuartaSetup-21.1.0.842-windows.exe	07.08.2022	11 KB	901 MB
aria10-21.1.0.842.qdz	07.08.2022	11 KB	490 MB
cyclone-21.1.0.842.qdz	07.08.2022	11 KB	405 MB
cyclone10p-21.1.0.842.qdz	07.08.2022	11 KB	285 MB
cyclonev-21.1.0.842.qdz	07.08.2022	11 KB	1.34 TB
max-21.1.0.842.qdz	07.08.2022	11 KB	11.3 MB
max10-21.1.0.842.qdz	07.08.2022	11 KB	201 MB

2. Предварительно установите библиотеки с помощью команд⁶:

```
sudo dpkg --add-architecture i386
sudo apt update
sudo apt-get --yes install libc6:i386 libncurses5:i386 libxtst6:i386 libxft2:i386
libc6:i386 libncurses5:i386 libstdc++6:i386
```

3. Для установки откройте терминал (для lubuntu F4) в папке с установщиком .run и введите следующие команды:

```
chmod +x QuartusLiteSetup-21.1.0.842-linux.run

./QuartusLiteSetup-21.1.0.842-linux.run
```

⁵ Если вам потребуется поддержка других серий ПЛИС в своих проектах или необходим симулятор Questa, вы так же можете найти дополнительные файлы для установки на данном диске. (Для бесплатной версии Questa тоже требуется лицензия)

⁶ Если какие-то библиотеки не установятся в виде одной команды, можно попробовать их установить по отдельности



ВАЖНО: Директория установки Quartus не должна содержать символов кириллицы, это может привести к сбоям в его работе.

Так же после выбора директории для установки у вас должно появиться окно выбора установки доп.файлов для ПЛИС серии Cyclone IV.

Select Components



Если этого не произошло, вероятно вы не скачали .qdz архив, или поместили его не в папку с установщиком. Вы можете добавить этот архив позже в самом Quartus с помощью Tools-Install Devices.

Так же необходимо добавить следующие переменные в файл ~/.bashrc: Сделайте это с помощью команды `sudo vi ~/.bashrc` (версии на скриншоте и другие компоненты отлич.) :

```
if [ shept -eq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export QSYS_ROOTDIR="/home/user/intelFPGA_lite/20.1/quartus/sopc_builder/bin"
export QUARTUS_ROOTDIR="/home/user/intelFPGA_lite/20.1/quartus"
PATH=$PATH:/home/user/intelFPGA_lite/20.1/modelsim_ase/bin:/home/user/intelFPGA_lite/20.1/quartus/bin:/usr/local/bin:/opt/riscv/bin:/home/user/Downloads/RVFpga_v2-1/SweRV-ISS/build-Linux
```

```
export QSYS_ROOTDIR="/home/user/intelFPGA_lite/21.1/quartus/sopc_builder/bin"
```

```
export QUARTUS_ROOTDIR="/home/user/intelFPGA_lite/21.1/quartus"
```

```
PATH=$PATH:/home/user/intelFPGA_lite/20.1/quartus/bin:/usr/local/bin
```

2.2 Установка драйвера для usb-blaster

Для работы с платой необходимо установить драйвер для usb-blaster.

1. Подключаем питание платы и программатор в ПК, как описано в шаге 1.2.

2. После подключения программатора к ПК, для начала проверим, определился ли он в системе с помощью команды `lsusb`:

```
edamc@edamc-virtualbox:~/Downloads$ lsusb
Bus 001 Device 005: ID 09fb:6001 Altera Blaster
Bus 001 Device 004: ID 80ee:0021 VirtualBox USB Tablet
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

3. Создайте файл `/etc/udev/rules.d/51-usbblaster.rules` с помощью команды:

```
sudo touch /etc/udev/rules.d/51-usbblaster.rules
```

Откройте его через `sudo vi /etc/udev/rules.d/51-usbblaster.rules` и скопируйте в него следующее:

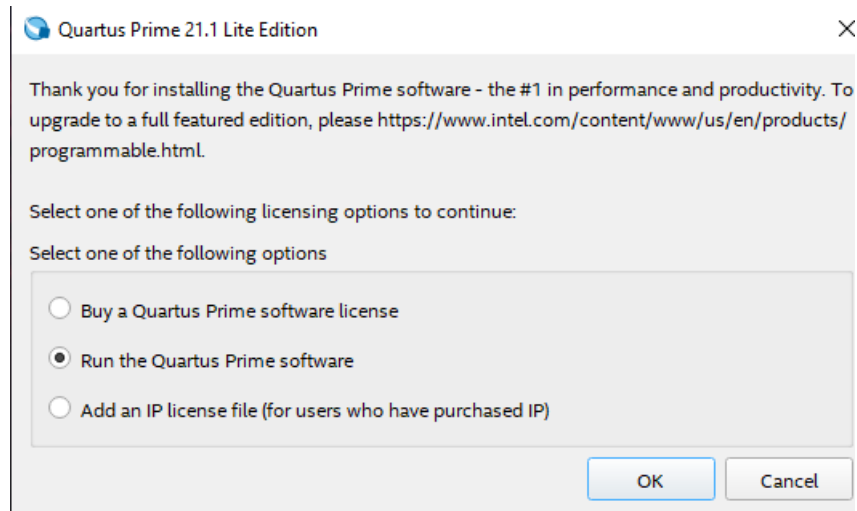
```
# USB Blaster
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6001",
MODE="0666", NAME="bus/usb/${env{BUSNUM}}/${env{DEVNUM}}", RUN+="/bin/chmod 0666 %c"
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6002",
MODE="0666", NAME="bus/usb/${env{BUSNUM}}/${env{DEVNUM}}", RUN+="/bin/chmod 0666 %c"
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6003",
MODE="0666", NAME="bus/usb/${env{BUSNUM}}/${env{DEVNUM}}", RUN+="/bin/chmod 0666 %c"

# USB Blaster II
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6010",
MODE="0666", NAME="bus/usb/${env{BUSNUM}}/${env{DEVNUM}}", RUN+="/bin/chmod 0666 %c"
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6810",
MODE="0666", NAME="bus/usb/${env{BUSNUM}}/${env{DEVNUM}}", RUN+="/bin/chmod 0666 %c"
```

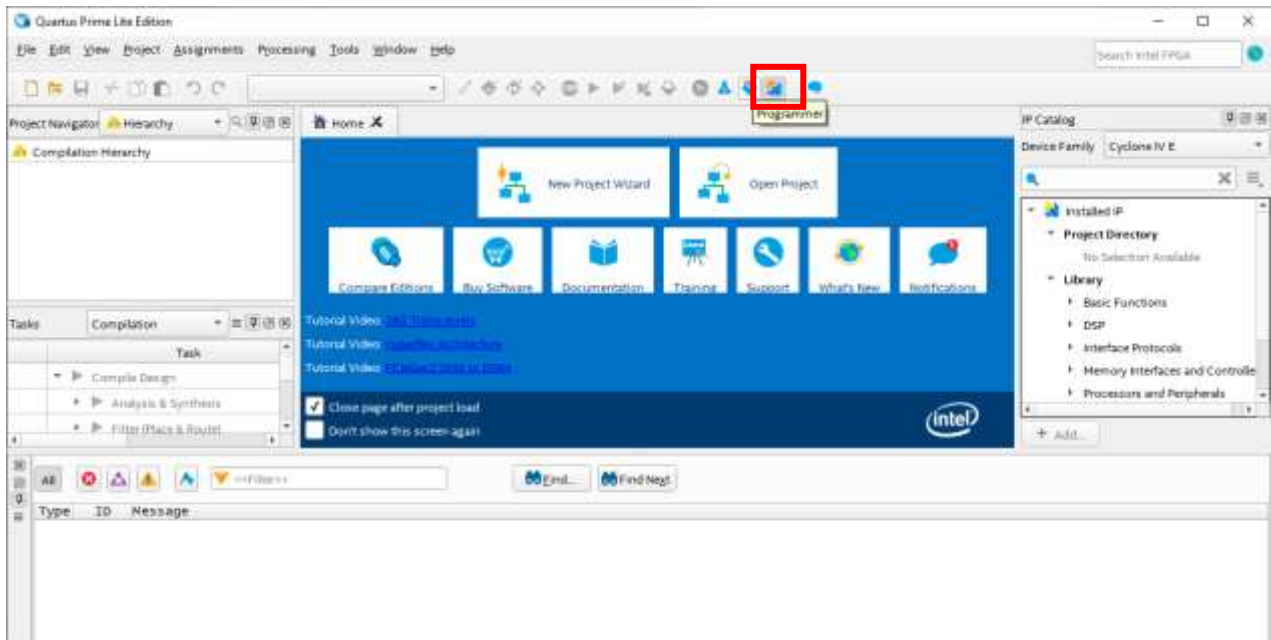
4. Затем, вытащите и снова подключите кабель программатора `usb-blaster`.

2.3 Проверка правильности установки Quartus

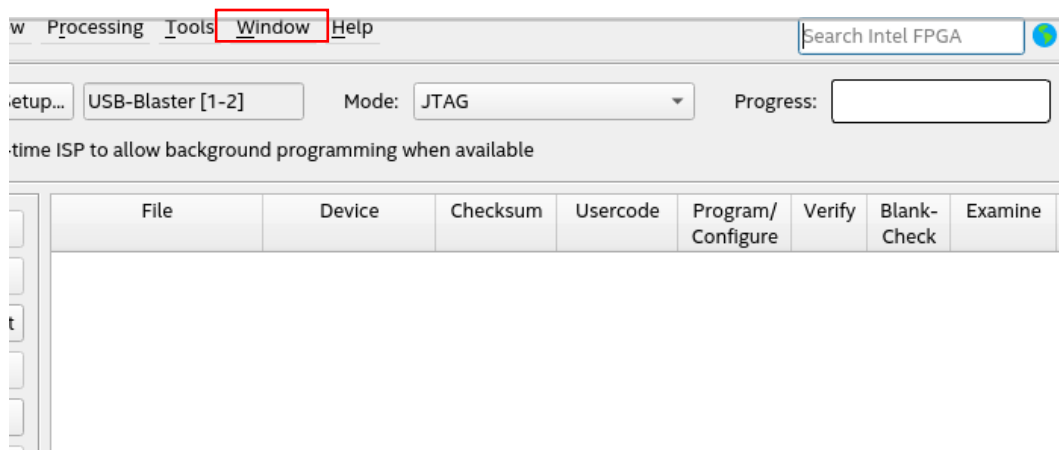
1. При запуске Quartus должно появиться такое окно, запускаем через *Run the Quartus Prime Software*:



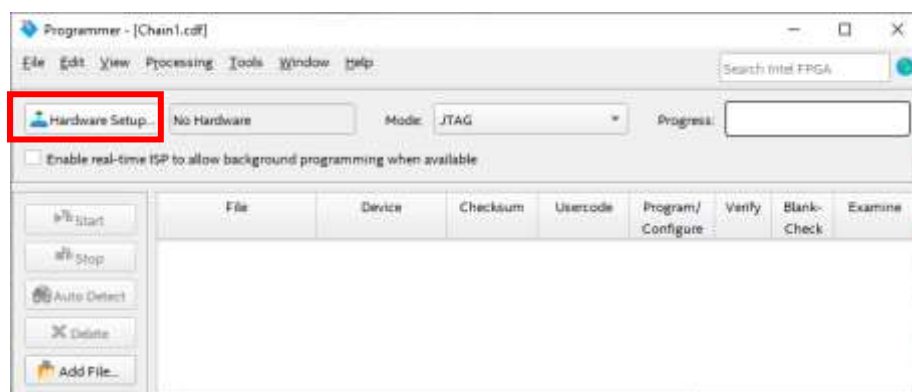
2. Открываем окно *programmer*:



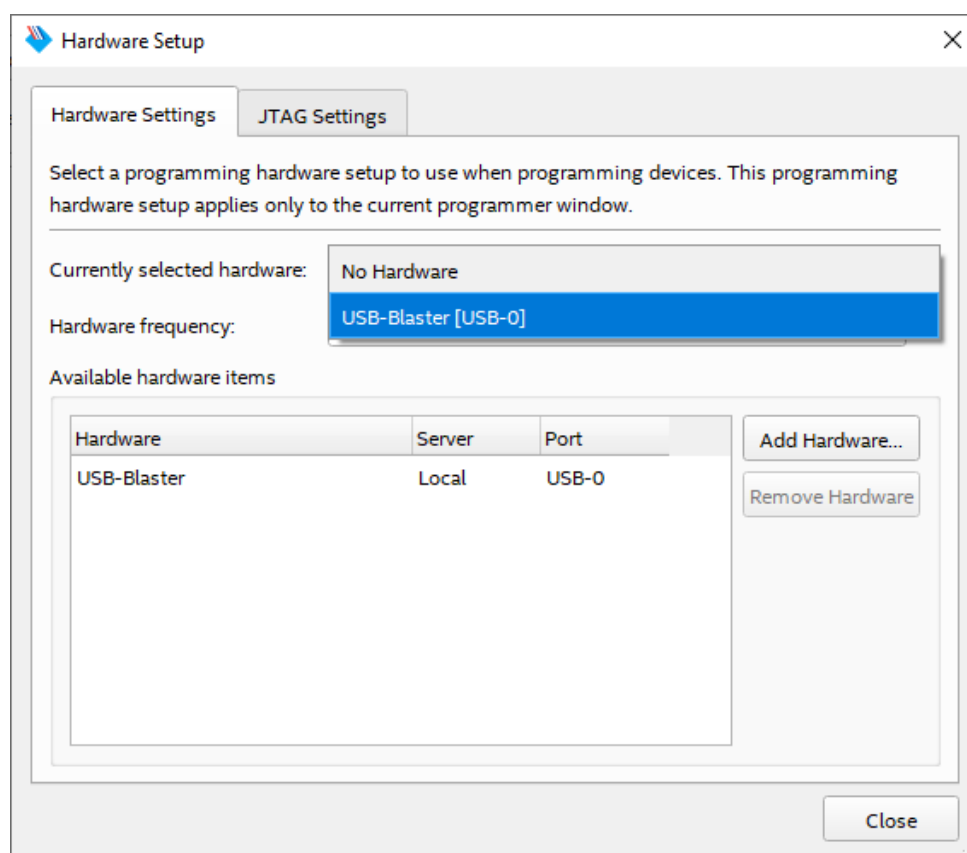
Иногда может возникнуть проблема с прикреплением окна к краю экрана, эту проблему можно решить с помощью Window-Attach Window:



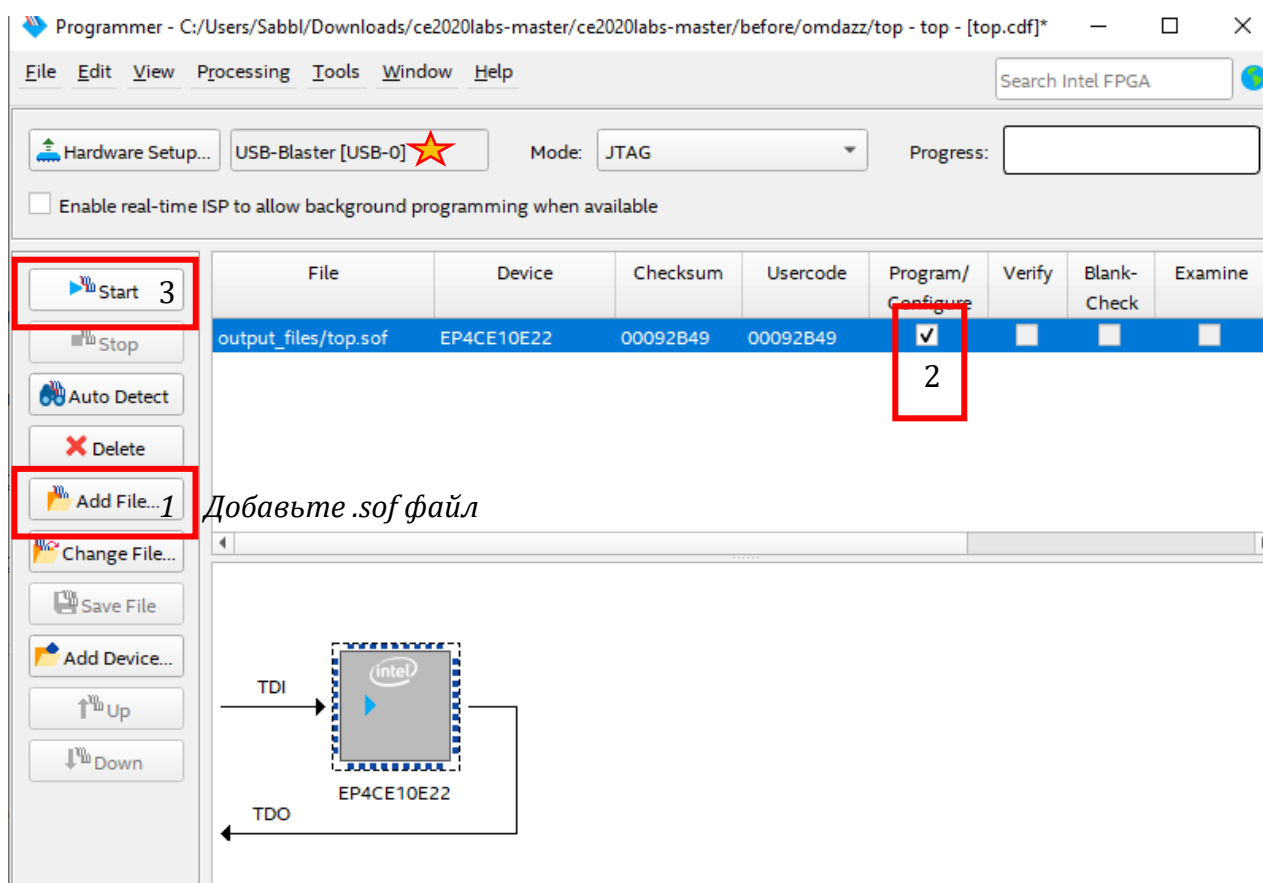
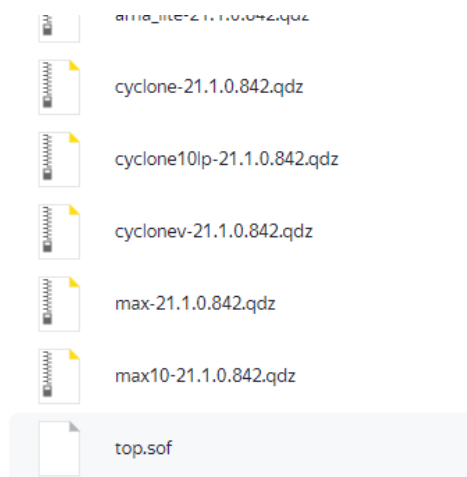
3. Далее нажимаем *Hardware setup*:



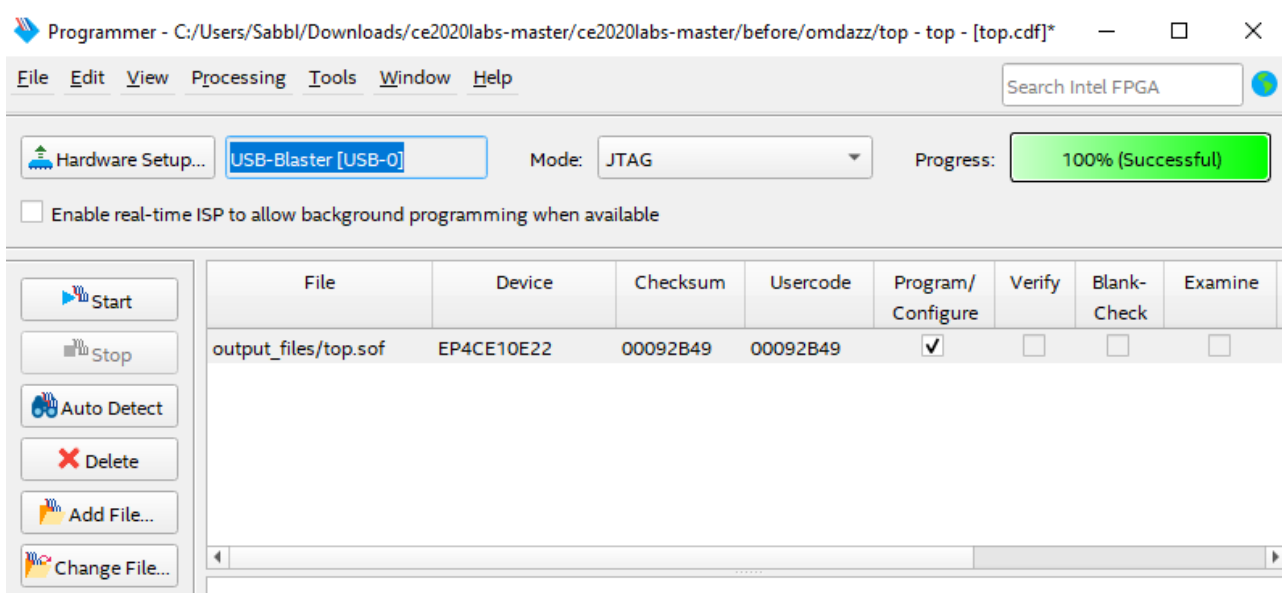
4. Если при выборе подключенного программатора в выпадающем списке присутствует USB-Blaster установку драйвера можно считать завершенной:



5. загрузите файл прошивки *top.sof* с Yandex диска и попробуйте загрузить его на плату *OMDAZZ* или *rzrd* с помощью того же окна programmer через *Add File* (Не забудьте про Hardware setup -> USB-Blaster):



Если вы увидели надпись **Successful**, установку и проверку работы Quartus можно завершить:



2.4 Установка icarus Verilog и gtkwave на Linux

<https://www.youtube.com/watch?v=nY150XXEj6M> – видео по установке.

<https://github.com/steveicarus/iverilog> - git-репозиторий для ручной установки Icarus последней версии

1. Если у вас дистрибутив Ubuntu версии 22.04 введите в консоль команду:

```
sudo apt install iverilog gtkwave
```

2. Проверьте версию Icarus:

```
iverilog -v
```

Если не возникает проблем, то это - все что нужно сделать.

2. (Не обязательно) Проверить работоспособность Icarus можно на файле с расширением «.v». Просто выберите любой Verilog-модуль и напишите команду

```
iverilog [имя модуля].v
```

Если в модуле нет ошибок, сообщений в консоли не появится, а в папке с модулем будет создан файл с расширением «.out».

3. Установка и настройка ПО для работы с OpenLane

Данный способ установки OpenLane проверялся только на Ubuntu/Ubuntu 20.04/22.04. Для других дистрибутивов имеется инструкция на сайте: <https://docs.docker.com/engine/install/ubuntu/>

OpenLane можно поставить только на Linux, поэтому если у вас ОС Windows 10/11 допускается 2 варианта:

1. Вы можете установить ВТОРУЮ операционную систему Linux рядом с текущей. Как это сделать см. статью: [Установка Linux рядом с Windows 10 - Losst](#) .

2. Вы можете установить виртуальную машину с ubuntu и выделить ей достаточно места для всего ПО (~30-40 Гб). Как это сделать см. статью: [Создание виртуальной машины.](#)

Если вы обладатель Lubuntu/Ubuntu, никаких дополнительных шагов предпринимать не придётся.

Открываем терминал.

Для начала нужно установить необходимые python библиотеки, git, make:

```
sudo apt install git
sudo apt install make
sudo apt install build-essential python3 python3-venv python3-pip
```

Устанавливаем docker с помощью последовательности команд в терминале:

```
sudo apt-get remove docker docker-engine docker.io containerd runc

sudo apt-get update

sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

sudo mkdir -p /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Данная команда выведет список доступных версий для вашего дистрибутива Linux:

```
apt-cache madison docker-ce
```

В нашем случае версия докер (5:20.10.17~3-0~ubuntu-focal) может отличаться от вашей, внимательно смотрите инструкцию по установке на сайте <https://docs.docker.com/engine/install/ubuntu/>

```
sudo apt-get install docker-ce=5:20.10.17~3-0~ubuntu-focal docker-ce-cli=5:20.10.17~3-
0~ubuntu-focal containerd.io docker-compose-plugin

sudo docker run hello-world
```

В случае успешного завершения установки docker, должно появиться такое сообщение:

```

user@user-quartus:~/Desktop$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:7d246653d0511db2a6b2e0436cfd0e52ac8c066000264b3ce63331ac66dca625
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

```

Не забудьте добавить себя в группу docker!

```
sudo usermod -aG docker $USER
```

После установки необходимо перелогиниться или перезагрузить машину!!!

Скачиваем git-репозиторий с OpenLane:

```

mkdir -p ~/github
cd ~/github
git clone https://github.com/The-OpenROAD-Project/OpenLane.git
cd OpenLane
make

```

После завершения установки нужно провести тестовый запуск:

```
make test
```

```

[INFO]: There are no setup violations in the design at the typical corner.
[SUCCESS]: Flow complete.
[INFO]: Note that the following warnings have been generated:
[WARNING]: Current core area is too small for a power grid. The power grid will be minimized.
[WARNING]: There are max fanout violations in the design at the typical corner. Please refer to 'des
ff/29-rcx_sta.slew.rpt'.

Basic test passed
user@user-quartus:~/Desktop/OpenLane$ █

```

Если тестовый запуск прошёл, можно попробовать посмотреть результат выполнения тестового flow с помощью встроенного в контейнер докера редактора gds файла **klayout**:

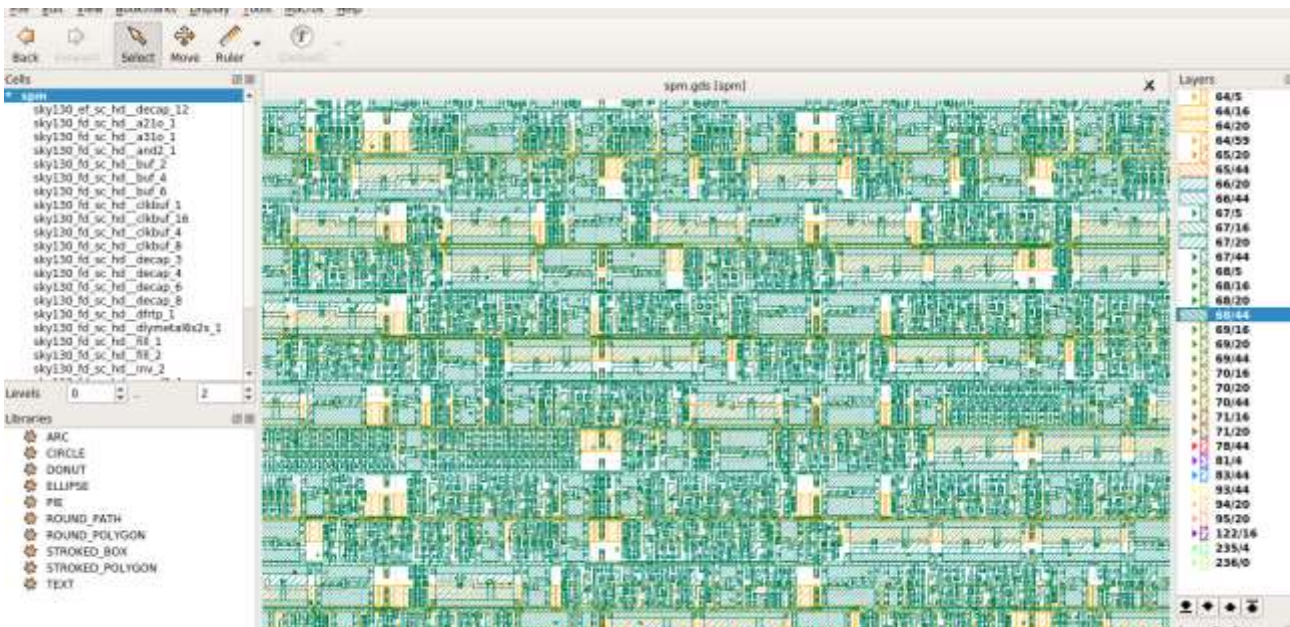
make mount

Должна открыться консоль файловой системы контейнера:

```
user@user-quartus:~/Desktop/OpenLane$ make mount
cd /home/user/Desktop/OpenLane && \
    docker run --rm -v /home/user/Desktop/OpenLane:/openlane -v /home/user/Desktop/OpenLane/designs:/openlane/pdks:/home/user/Desktop/OpenLane/pdks -e PDK_ROOT=/home/user/Desktop/OpenLane/pdks -e PDK=sky130 -c_hd --user 1000:1001 -e DISPLAY=:0 -v /tmp/.X11-unix:/tmp/.X11-unix -v /home/user/.Xauthority:/home/user/.Xauthority:efabless/openlane:6ab944bc23688cae6dc6fa32444891a1e57715c8-amd64
Unable to find image 'efabless/openlane:6ab944bc23688cae6dc6fa32444891a1e57715c8-amd64' locally
6ab944bc23688cae6dc6fa32444891a1e57715c8-amd64: Pulling from efabless/openlane
Digest: sha256:01fe69759428be4f780985ea38474c9fcf9b185a3b1d6f8efe27a838ef425b55
Status: Downloaded newer image for efabless/openlane:6ab944bc23688cae6dc6fa32444891a1e57715c8-amd64
OpenLane Container (6ab944b):/openlane$
```

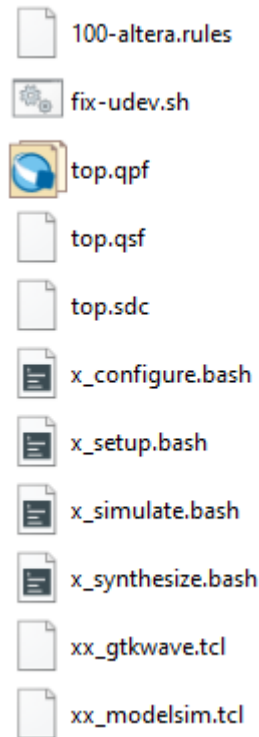
Введите следующую команду для просмотра полученного gds файла:

klayout designs/spm/runs/openlane_test/results/final/gds/spm.gds



Если что-то не получилось можно обратиться к статье Юрия Панчула по установке OpenLane на сайте Хабр: <https://inlnk.ru/20L0od> или написать в группу Школы Синтеза Цифровых схем в telegram: <https://t.me/DigitalDesignSchool>

4. Структура каталога с лабораторными работами в Quartus



Проектные файлы Quartus

- **top.qpf** – Quartus project file. Файл проекта Quartus, который автоматически появляется при создании проекта. Сам по себе не содержит важной информации (только ревизии проекта), основные настройки хранятся в файле с расширением .qsf
- **top.qsf** – Quartus settings file. Файл настроек проекта Quartus. Автоматически подгружается в проект из папки. Содержит все настройки проекта, включая параметры компиляции, тип используемой ПЛИС и т.д. Важной частью файла являются настройки подключения физических пинов ПЛИС к проекту и пути к файлам дизайна .v .sv и т.д.
- **top.sdc** – Synopsys design constraints. Файл проектных ограничений. Необходим для работы статического временного анализатора в Quartus. Содержит информацию о частоте тактового сигнала и задержках входа/выхода.

*Скрипты x**

Необходимы для автоматического запуска Quartus из консоли без использования графического интерфейса. Можно использовать на win, например из git bash при правильной установке переменных среды

- **x_setup.bash** – Скрипт для задания и проверки переменных среды Quartus.
- **x_synthesize.bash** – Скрипт для полной компиляции проекта (от верилога до платы). Результат работы скрипта .sof файл для загрузки в ПЛИС.
- **x_configure.bash** – Скрипт для конфигурирования ПЛИС. Автоматически подгружается после завершения компиляции скриптом **x_synthesize.bash**.

Можно использовать отдельно от `x_synthesize.bash` , предварительно завершив компиляцию.

- `x_simulate.bash` – Скрипт для запуска симуляции, на выбор даётся запуск из icarus+gtkwave, modelsim/questa.
- `xx_gtkwave.tcl` – Скрипт для загрузки сигналов в gtkwave.
- `xx_modelsim.tcl` – Скрипт для загрузки сигналов в modelsim.

Запуск скриптов на windows.

Внимание! Если путь к проекту содержит кириллицу, скрипт не заработает! Так же необходимо использовать все файлы, начиная с каталога `\omdazz\lab_1_and_or_xor\run`, поскольку скрипт ссылается на файлы в верхних каталогах!

```
mosko1enko@DESKTOP-K8JEJ3H MINGW64 /c/labs/day_4/v/omdazz/lab_1_and_or_xor/run
$ ./x_synthesize.bash
```

Запуск скрипта на синтез из git bash из каталога v.

Дополнительная информация

По мере появления новых вопросов у студентов Школы, инструкция может обновляться, не забывайте проверять новости на сайте ChipExpo и наш телеграм-канал!

1. Телеграм-канал Школы цифрового синтеза (Здесь вы сможете найти ответы на интересующие вас вопросы или задать свой)- <https://t.me/DigitalDesignSchool>

2. Основной гит-репозиторий Школы синтеза цифровых схем - <https://github.com/DigitalDesignSchool>

3. Гит-репозиторий OpenLane - <https://github.com/The-OpenROAD-Project/OpenLane/blob/master/README.md>