

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

## Содержание

Ручная установка ПО.....	2
1. Установка и настройка ПО для работы на ОС Windows 10/11.....	2
1.1 Установка Quartus 21.1 Lite(Обязательно).....	2
1.2 Установка драйвера для usb-blaster(Обязательно).....	4
1.3 Проверка правильности установки Quartus.....	7
1.4 Установка Icarus Verilog 12 и gtkwave на Windows(Обязательно).....	11
1.5 Установка ModelSim на Windows (Опционально).....	12
1.6 Установка Questa Lite на Windows(Опционально).....	12
1.7 Установка Git на Windows(Обязательно).....	12
1.8 Установка RARS (Обязательно).....	13
2. Установка и настройка ПО для работы на базе ОС linux (на примере ubuntu 22.04).....	14
2.1 Установка Quartus 21.1 Lite(Обязательно).....	14
2.2 Установка драйвера для usb-blaster(Обязательно).....	16
2.3 Проверка правильности установки Quartus.....	16
2.4 Установка Icarus Verilog 12 и gtkwave на Linux(Обязательно).....	20
2.5 Установка ModelSim на Linux(Опционально).....	21
2.6 Установка Questa Lite на Linux(Опционально).....	21
2.7 Установка RARS на Linux(Обязательно).....	21
3. Установка и настройка ПО для работы с OpenLane (Опционально).....	21
4. Установка виртуальной машины для занятий по функциональной верификации (обязательно).....	24
Создание загружаемого SSD с предустановленным ПО при помощи операционной системы Linux.....	25
1. Запрос образа и первоначальная настройка SSD.....	25
2. Загрузка с SSD.....	25
3. Обновление репозитория.....	28
4. Проверка работы с FPGA платой.....	29
Использование виртуальной машины с предустановленным ПО.....	31
Дополнительная информация.....	36
FAQ.....	36

# Ручная установка ПО

**Данная инструкция описывает процесс установки ПО для работы на отладочных платах с ПЛИС компании Altera/Intel (САПР Quartus).**

**Если у вас используется плата с ПЛИС Xilinx или Gowin или плата вообще отсутствует, Вы можете пропустить шаги 1.1-1.3 (Win) или 2.1-2.3 (Linux) (Список плат и производителей вы можете найти в FAQ).**

**Для установки Vivado или Gowin IDE вы можете воспользоваться доступной в интернете информацией. Если будут возникать проблемы с установкой этих САПРов инструкцию можно дополнить.**

Все представленное ниже ПО бесплатное и свободно распространяется. Из-за того что могут возникнуть проблемы со скачиванием дистрибутивов с официальных сайтов Intel, с помощью VPN, их копии размещены в облаке Я.Диск. **Вполне допустима регистрация аккаунта и самостоятельное скачивание дистрибутивов.**

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

### 1.1 Установка Quartus 21.1 Lite(Обязательно)

Для установки Quartus воспользуйтесь любой доступной из ссылок на Яндекс Диск:

<https://disk.yandex.ru/d/CSn1xlo5QHj-HA>

<https://disk.yandex.ru/d/4EU4zcyj4Gj06rQ>

<https://disk.yandex.ru/d/yNZtfNM49scWrA>

<https://disk.yandex.ru/d/XfSZde5KrGnSig>

<https://disk.yandex.ru/d/EqLtpcOccHUm6Q>

которые содержит установщики и архивы для поддержки ПЛИС серии Cyclone IV, Cyclone V, MAX II, MAX 10.

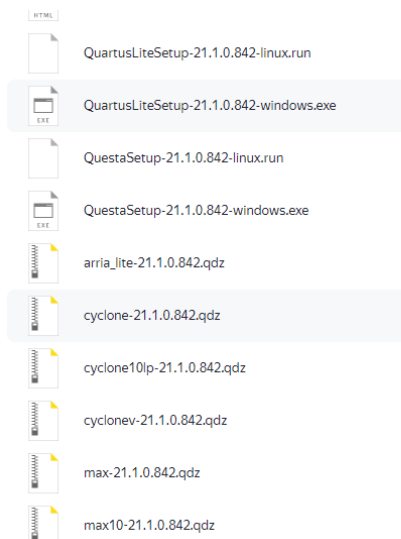
Для установки на ОС Windows 10/11 понадобятся следующие файлы<sup>1</sup>:

- Установщик QuartusLiteSetup-21.1.exe
- Архивы \*.qdz (В данном примере для cyclone IV) для поддержки ПЛИС серий Cyclone IV, Cyclone V, MAX II, MAX 10 используемых в лабораторных работах Школы. (Скриншоты с версией 21.1.0.842 , установка не отличается)<sup>2</sup>

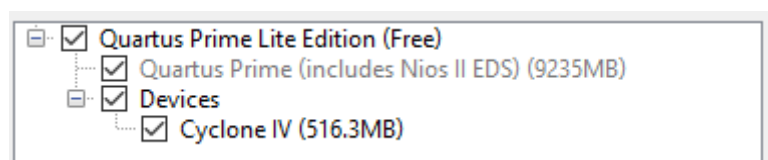
---

<sup>1</sup> 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**.

<sup>2</sup> 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 Я.Диске ( Для бесплатной версии Questa требуется лицензия). Например, для поддержки платы De10-Lite, нужен архив max10.qdz.



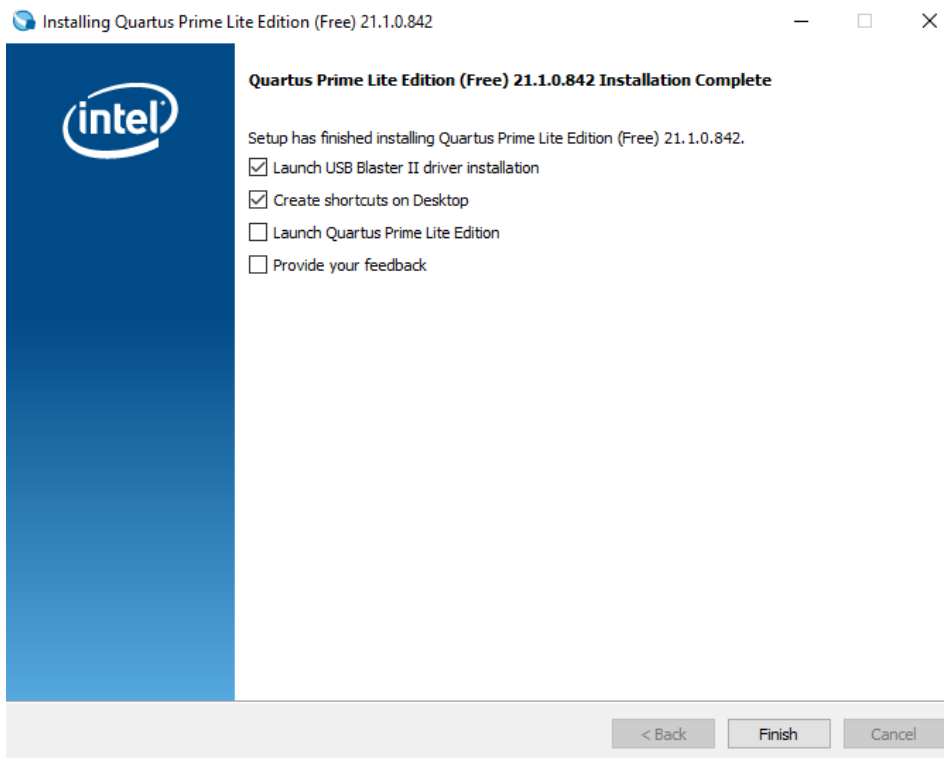
**ВАЖНО:** Директория установки Quartus не должна содержать символов кириллицы, это может привести к сбоям в его работе. Так же после выбора директории для установки у вас должно появиться окно выбора установки доп.файлов для ПЛИС серии Cyclone IV или других серий ПЛИС:



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

Quartus займёт не менее 10 ГБ на вашем ЖД.

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

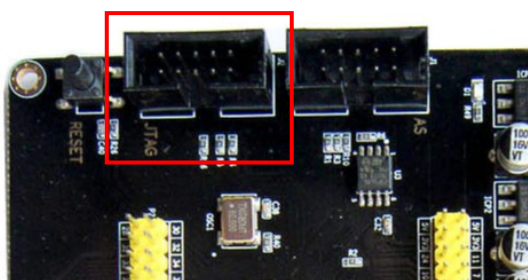


## 1.2 Установка драйвера для usb-blaster(Обязательно)

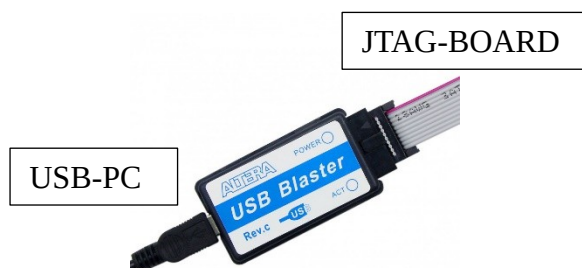
**В качестве примера будет рассматриваться плата OMDAZZ/RZRD с DDS2022-24**

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

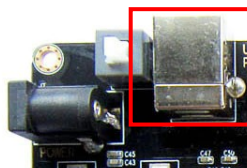
1. Подключите программатор (пластиковый корпус с надписью usb-blaster). Кабель JTAG подключите одним концом к разъёму JTAG (!не AS) на плате OMDAZZ/rzrd (см. рис ниже), а другим – к программатору. (Обратите внимание на выемку в разъёме)



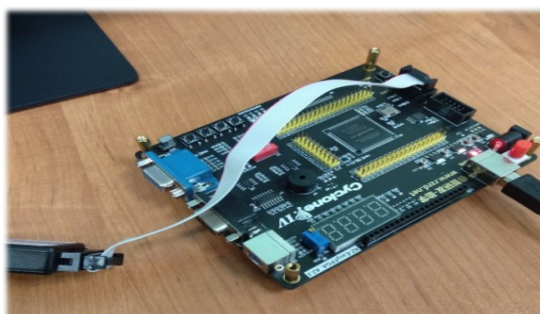
А USB-кабель - от программатора к ПК:



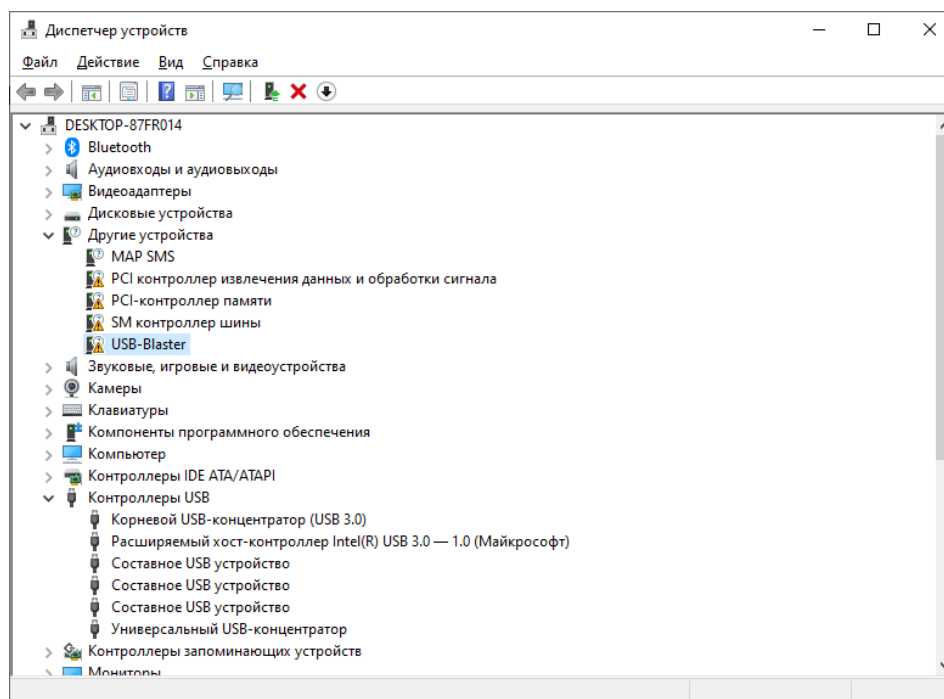
2. Подключите питание через USB B-TYPE (провод в комплекте) к разъёму (плата OMDAZZ/rzrd) (на рис. – правый) и включите плату (кнопка рядом с разъёмом).



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



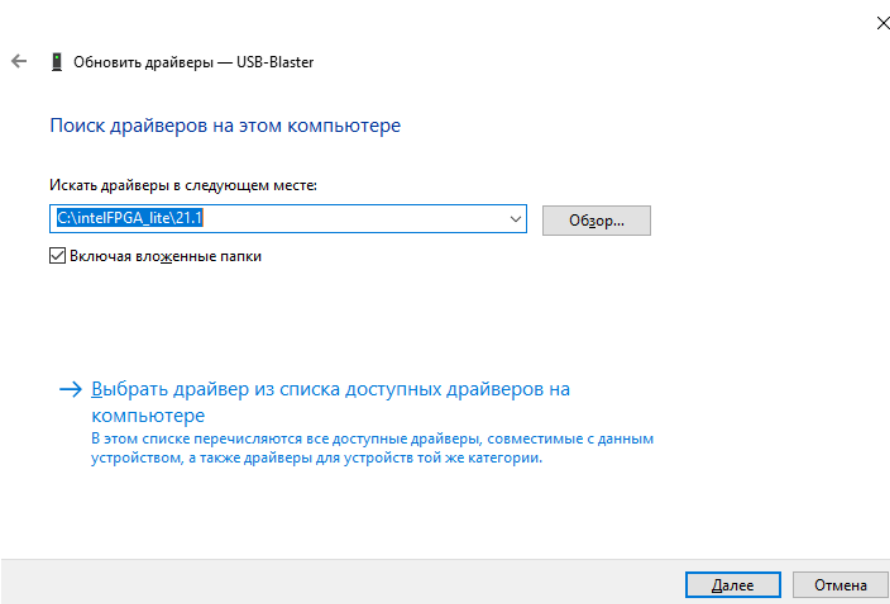
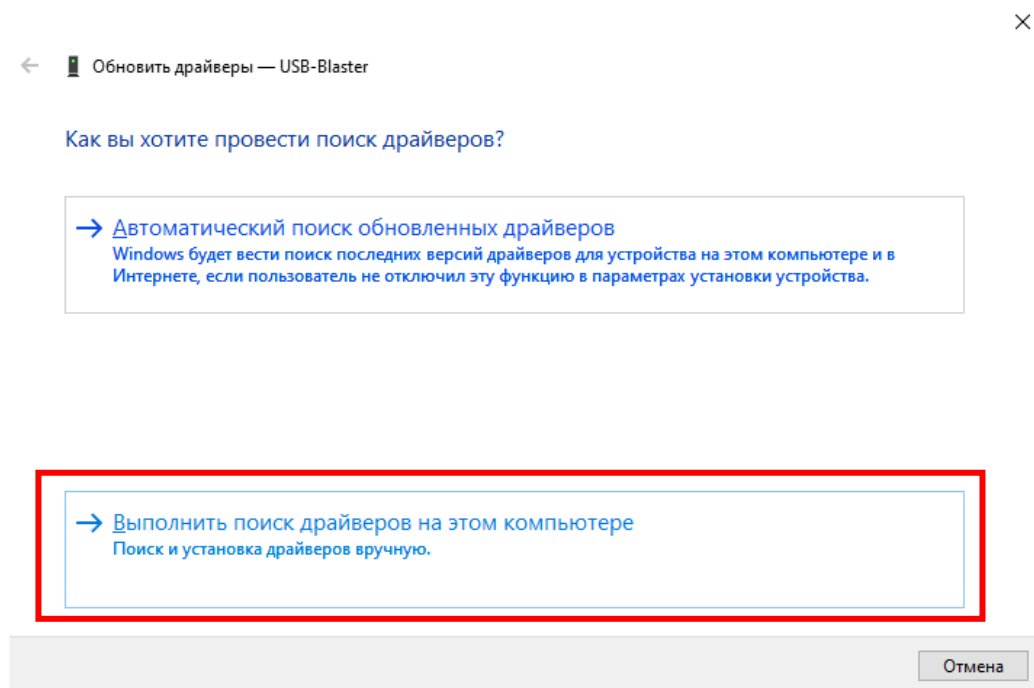
Далее, необходимо зайти в Диспетчер устройств ПК. Самый быстрый способ его найти – с помощью поиска возле кнопки «Пуск». После подключения программатора к ПК у вас должен появиться следующий пункт в диспетчере устройств<sup>3</sup>:



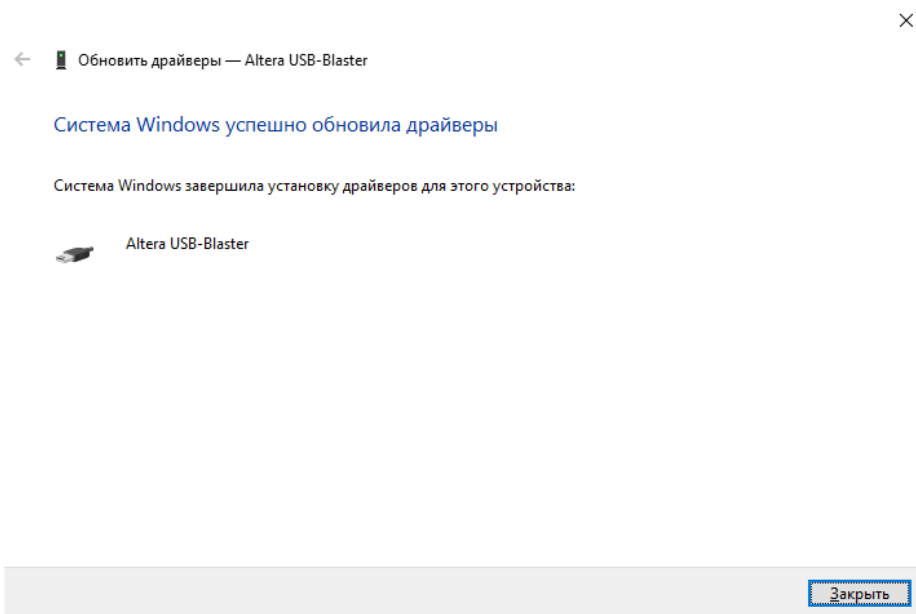
<sup>3</sup> Если он не появится, попробуйте поменять USB-разъем (с 3.0 на 2.0), к которому вы подключали программатор. Не работать может несколько, так что попробуйте все. Также лишний раз проверьте корректность подключения JTAG, работоспособность программатора (на нем должен гореть светодиод) и работоспособность самой платы (Если горят все светодиоды на семисегментном индикаторе одновременно, это говорит о том, что плата неисправна OMDAZZ). 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.<sup>4</sup>

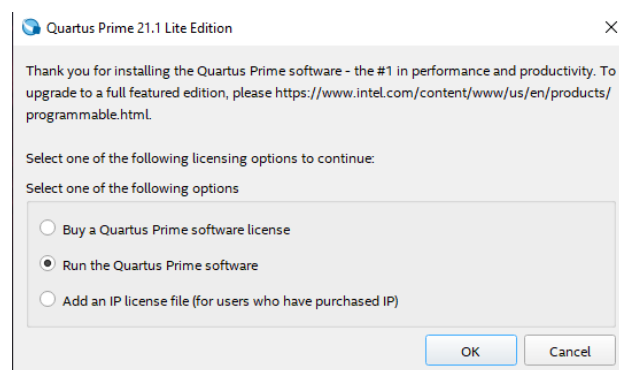


<sup>4</sup> В корневой папке Quartus содержатся драйвера для usb-blaster и usb-blaster-ii. Если драйвер не установится, обратитесь к модераторам.

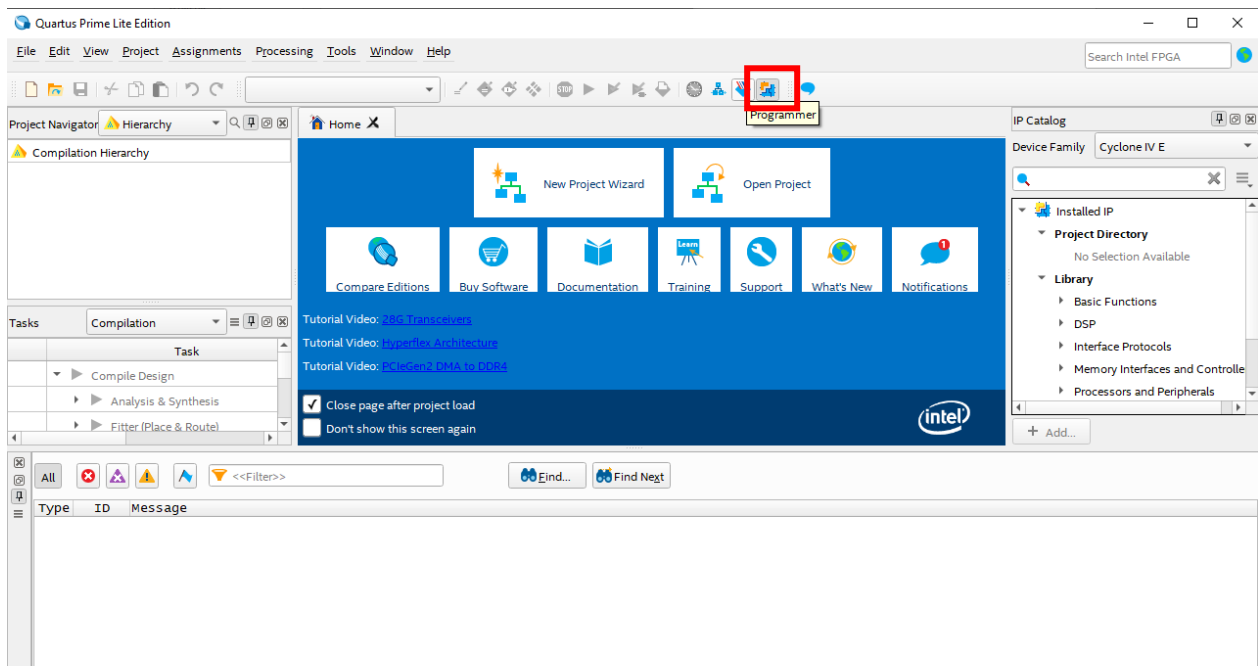


### 1.3 Проверка правильности установки Quartus

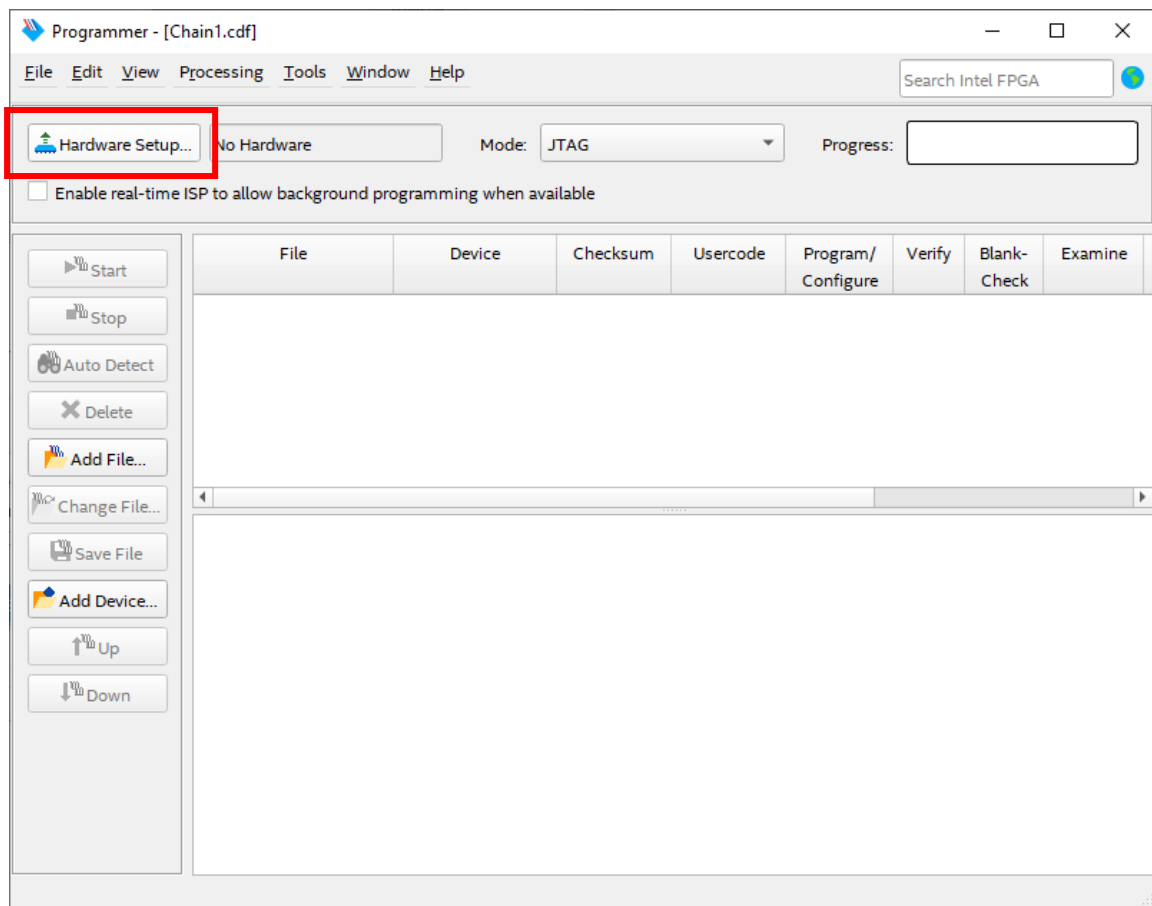
1. При запуске Quartus должно появиться такое окно, запускаем через *Run the Quartus Prime Software*:



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

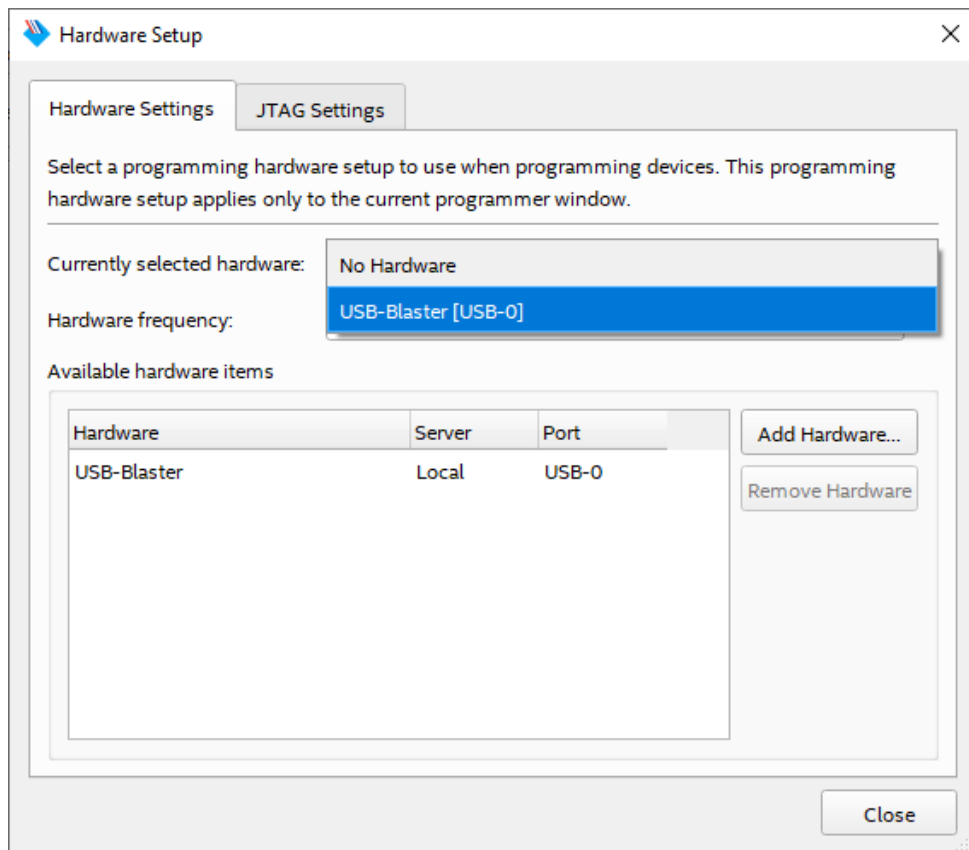


3. Нажимаем *Hardware setup*:

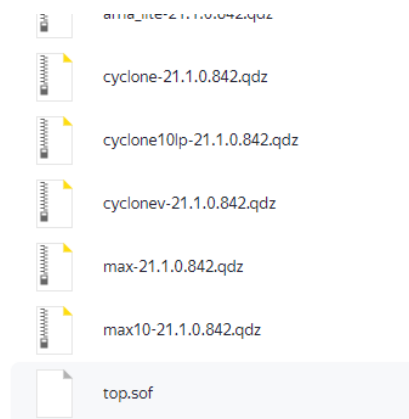


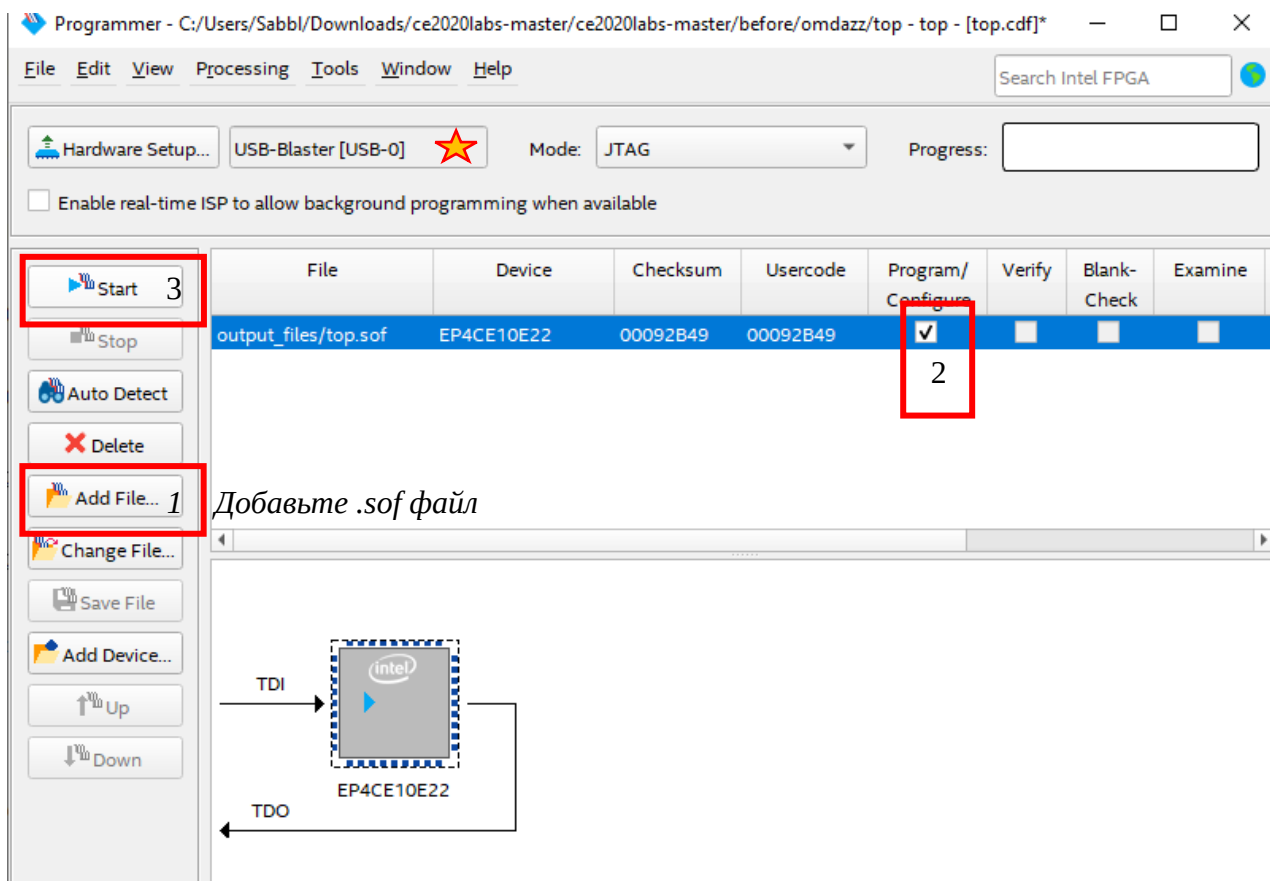
4. Если при выборе устройства в выпадающем списке присутствует USB-Blaster установку драйвера можно считать успешно завершённой:





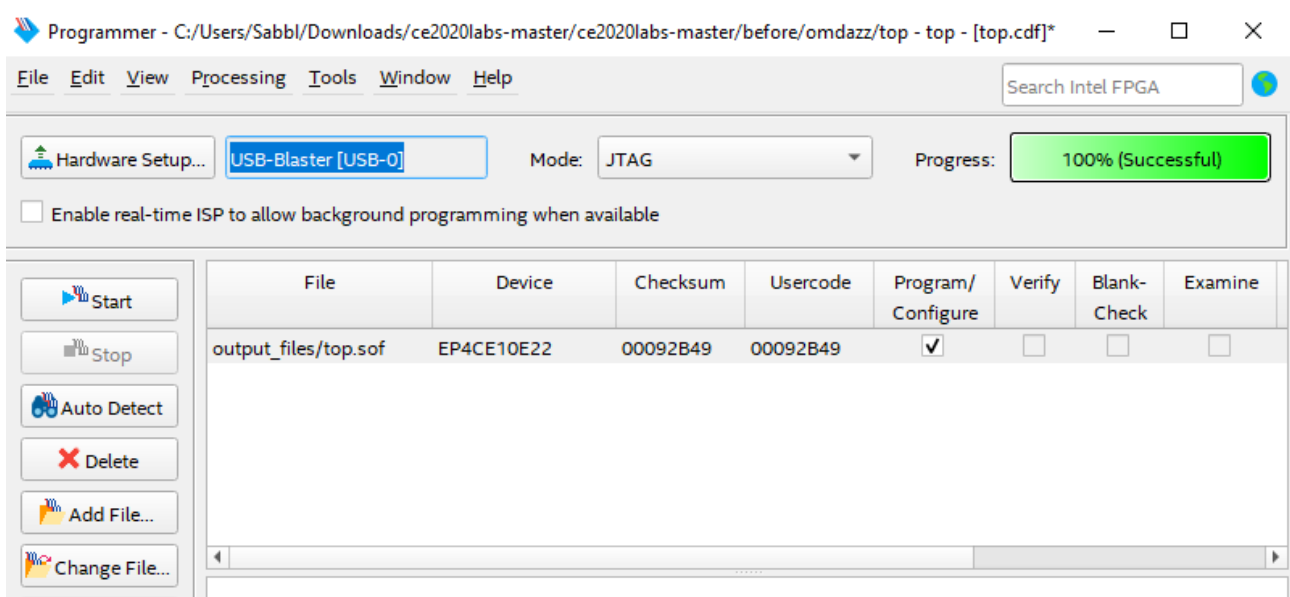
5. Загрузите файл прошивки top.sof с Yandex диска и попробуйте загрузить его на плату OMDAZZ или rzrd с помощью того же окна programmer через *Add File* (Не забудьте про **Hardware setup -> USB-Blaster**):





Е

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



**Вы так же можете работать с Quartus и выполнять некоторые базовые упражнения без платы, например с помощью симулятора Icarus Verilog.**

## 1.4 Установка Icarus Verilog 12 и gtkwave на Windows(Обязательно)

Icarus Verilog – открытый симулятор, который подойдет для моделирования простых дизайнов.

GTKwave – утилита для просмотра временных диаграмм в формате .VCD

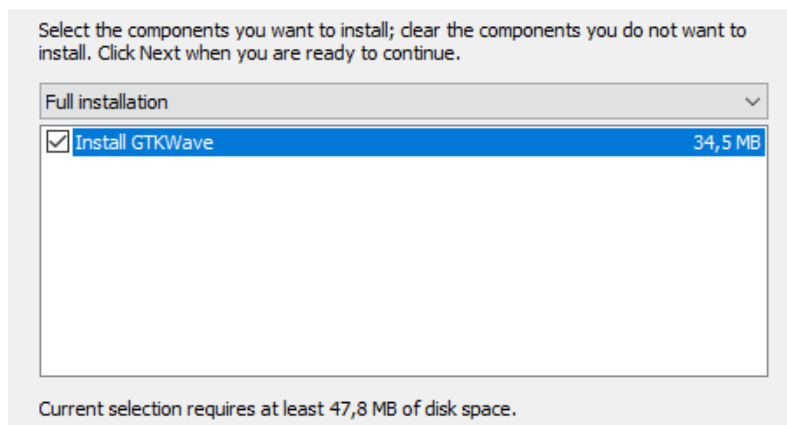
1. Перейдите на сайт с Icarus Verilog 12: <https://bleyer.org/icarus/> и скачайте последнюю версию:

### Download

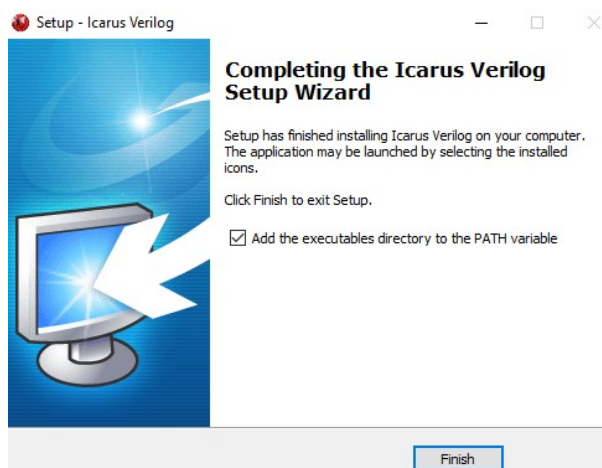
You can find Icarus Verilog sources and binaries for most platforms at the [Icarus site FTP](#). The source

- [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 \[588kB\]](#)

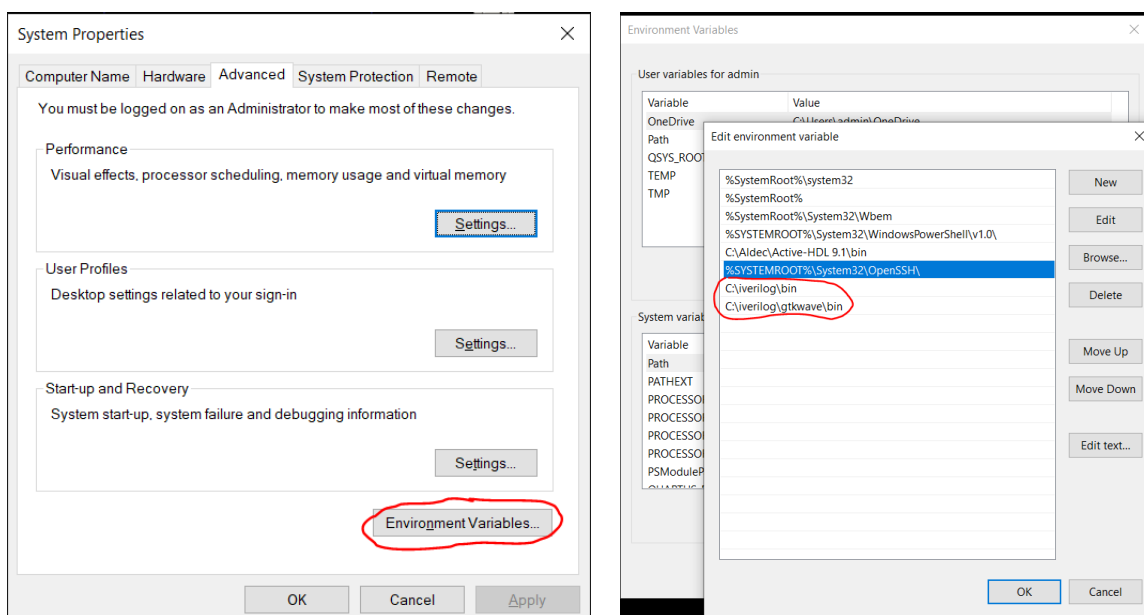
2. Активируйте установщик и следуйте инструкциям.
3. Дополнительно установите GTKWave:



4. После установки добавьте переменную среды (Скрипты автозапуска тестов в лабораторных работах первого дня используют полный путь до директории с установленным компилятором Icarus, но при упрощенном доступе к Icarus из командной строки этот шаг необходим):



Под учетной записью Администратора:



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

## 1.5 Установка ModelSim на Windows (Опционально)

В качестве другого симулятора в лабораторных работах так же может использоваться ModelSim, его можно скачать по ссылке: <https://disk.yandex.ru/d/zuN-mJTungHEDg>

← ModelSim :



## 1.6 Установка Questa Lite на Windows(Опционально)

Ссылка на скачивание дистрибутива Questa Lite [https://disk.yandex.lt/d/2\\_fm-KMoYTL87](https://disk.yandex.lt/d/2_fm-KMoYTL87)

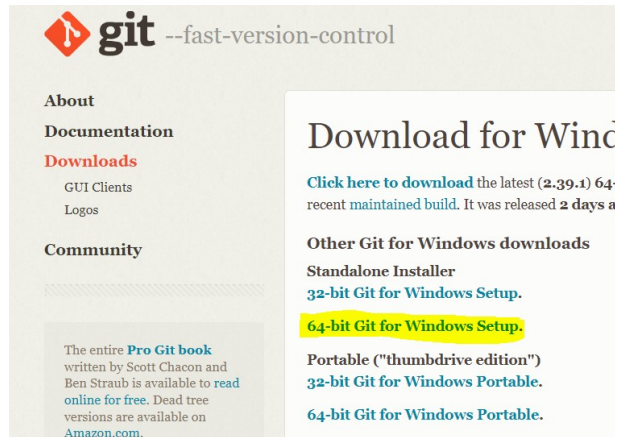
Q

Для использования **потребуется** зарегистрировать аккаунт и активировать бесплатную лицензию на сайте Intel с помощью VPN:: [Fix "Unable to checkout a license" for Questa Intel FPGA Starter Edition - YouTube](#)

## 1.7 Установка Git на Windows(Обязательно)

Git необходим для скачивания репозитория с обучающим материалом с gitflic.ru. Так же git имеет bash shell необходимый для запуска bash скриптов под windows.

1. Скачиваем и устанавливаем Git с сайта: <https://git-scm.com/download/win>



2. При установке под Windows нужно выбрать такую опцию:

#### Adjusting your PATH environment

How would you like to use Git from the command line?



☐ **Use Git from Git Bash only**

This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

☐ **Use Git from the Windows Command Prompt**

This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.

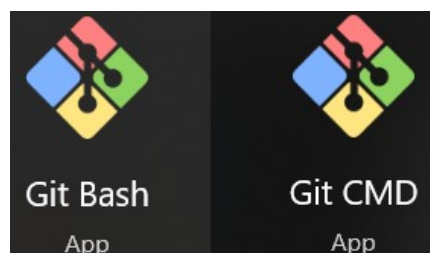
☒ **Use Git and optional Unix tools from the Windows Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.  
**Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.**

<https://gitforwindows.org/>

< Back   **Next >**   Cancel

3. После установки в системе должны быть приложения GitBash и GitCMD:



## 1.8 Установка RARS (Обязательно)

RARS – Это симулятор архитектуры команд RISC-V.

RARS распространяется в виде исполняемого jar-файла. Вам понадобится как минимум Java 8, чтобы запустить его. Последний стабильный релиз можно найти здесь:

<https://github.com/TheThirdOne/rars/releases/tag/v1.6>

## 2. Установка и настройка ПО для работы на базе ОС linux (на примере ubuntu 22.04)

### 2.1 Установка Quartus 21.1 Lite(Обязательно)

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 любой ссылкой на Яндекс Диск :

<https://disk.yandex.ru/d/CSn1xlo5QHj-HA>











<https://disk.yandex.ru/d/4EU4zcj4Gj06rQ>

<https://disk.yandex.ru/d/yNZtfNM49scWrA>

<https://disk.yandex.ru/d/XfSZde5KrGnSig>

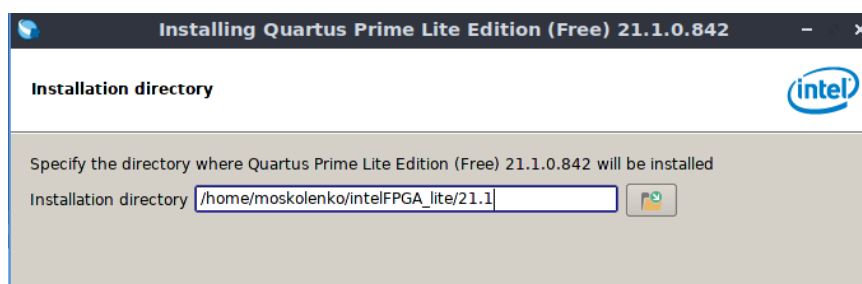
<https://disk.yandex.ru/d/EqLtpcOccHUm6Q>

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, Cyclone V, MAX II, MAX 10 используемых в лабораторных работах Школы. (Версии на скриншотах могут отличаться)<sup>5</sup>

	QuartusLiteSetup-21.1.0.842-linux.run	07.08.2022	11:08	1,88 ГБ
	QuartusLiteSetup-21.1.0.842-windows.exe	07.08.2022	11:08	1,6 ГБ
	QuestaSetup-21.1.0.842-linux.run	07.08.2022	11:08	1,82 ГБ
	QuestaSetup-21.1.0.842-windows.exe	07.08.2022	11:08	961 МБ
	arria_lite-21.1.0.842.qdz	07.08.2022	11:08	499 МБ
	cyclone-21.1.0.842.qdz	07.08.2022	11:08	465 МБ
	cyclone10lp-21.1.0.842.qdz	07.08.2022	11:08	265 МБ
	cyclonev-21.1.0.842.qdz	07.08.2022	11:08	1,34 ГБ
	max-21.1.0.842.qdz	07.08.2022	11:08	11,3 МБ
	max10-21.1.0.842.qdz	07.08.2022	11:08	285 МБ

2. 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
```

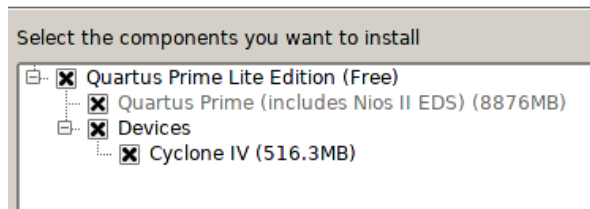


<sup>5</sup> Если вам потребуется поддержка других серий ПЛИС в своих проектах или необходим симулятор Questa, вы так же можете найти дополнительные файлы для установки на данном диске. (Для бесплатной версии Questa тоже требуется лицензия)

**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 вкладки Tools-Install Devices.

Так же необходимо добавить следующие переменные среды в файл ~/.bashrc для возможности быстрого запуска ПО через командную строку:

```
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
```

Вы можете сделать это с помощью команды или любого текстового редактора:

```
vi ~/.bashrc
```

## 2.2 Установка драйвера для usb-blaster(Обязательно)

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
```

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"

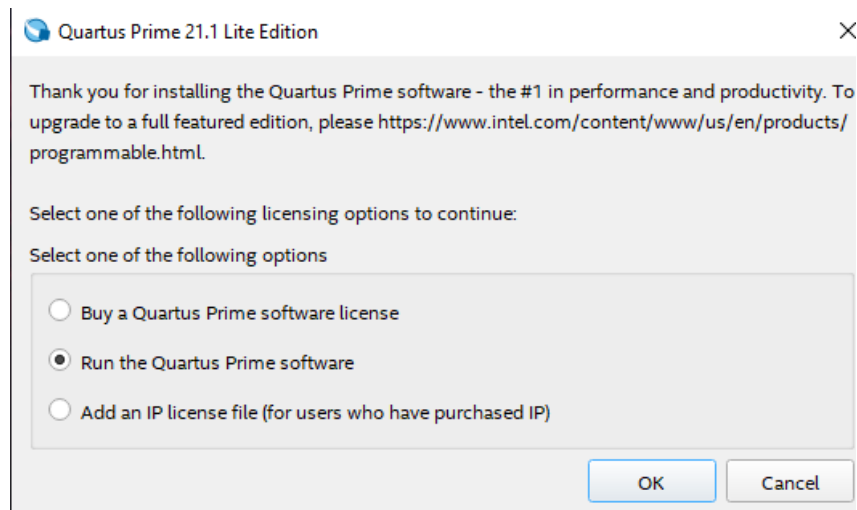
SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6810",
MODE="0666", NAME="bus/usb/${ENV{BUSNUM}}/${ENV{DEVNUM}}", RUN+="/bin/chmod 0666 %c"
```

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

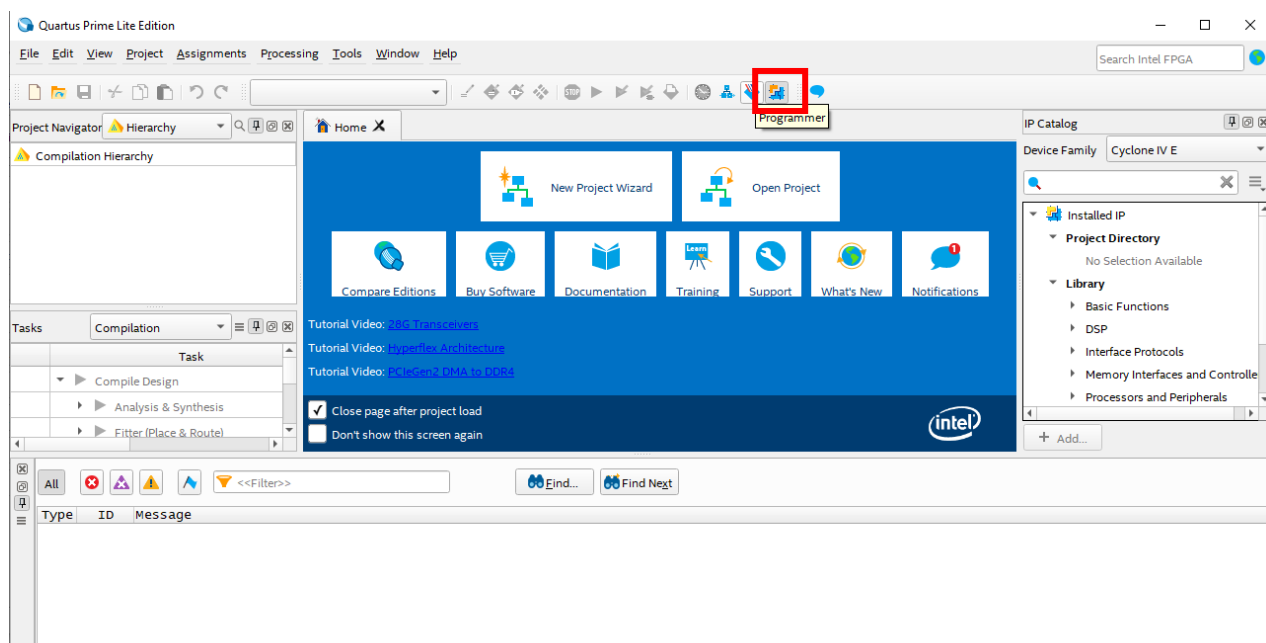
## 2.3 Проверка правильности установки Quartus

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

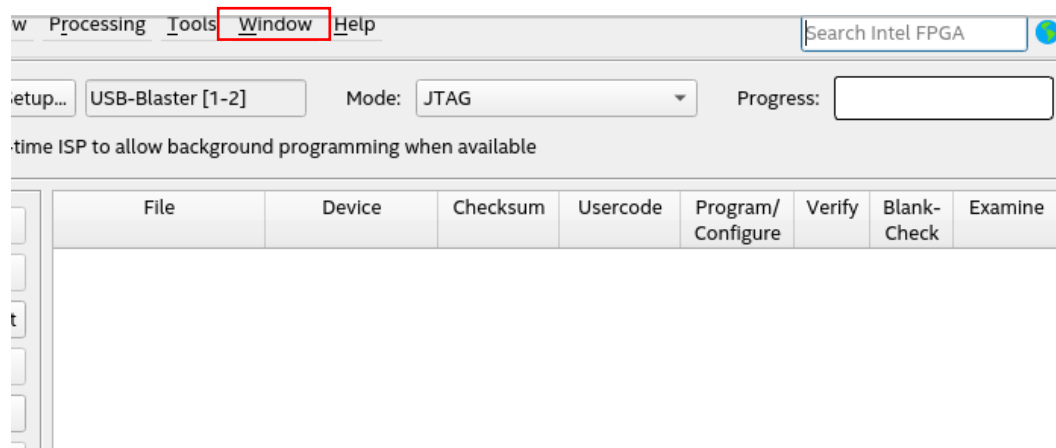




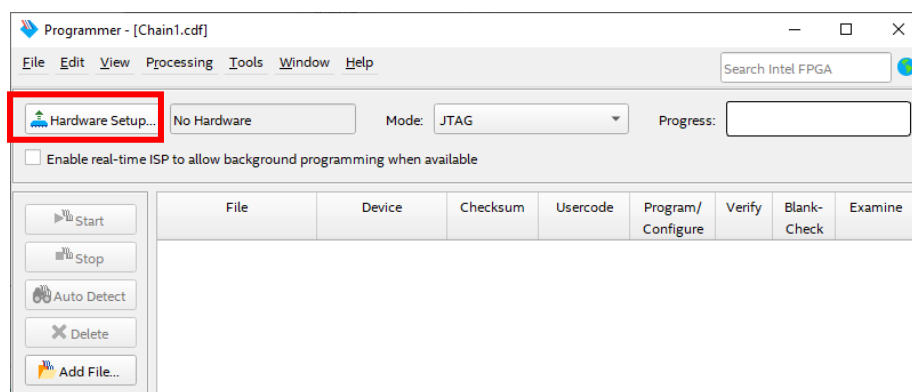
## 2. Открываем окно *programmer*:



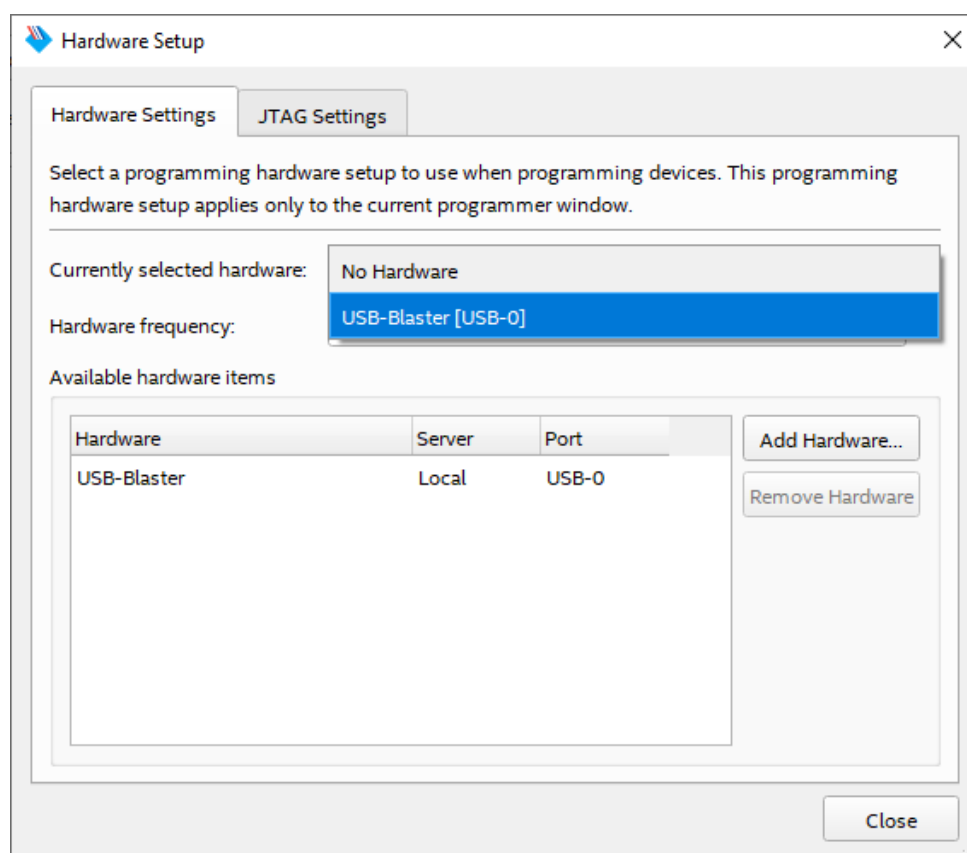
Иногда может возникнуть проблема с прикреплением окна к краю экрана(Lubuntu), эту проблему можно решить с помощью Window-Attach Window:



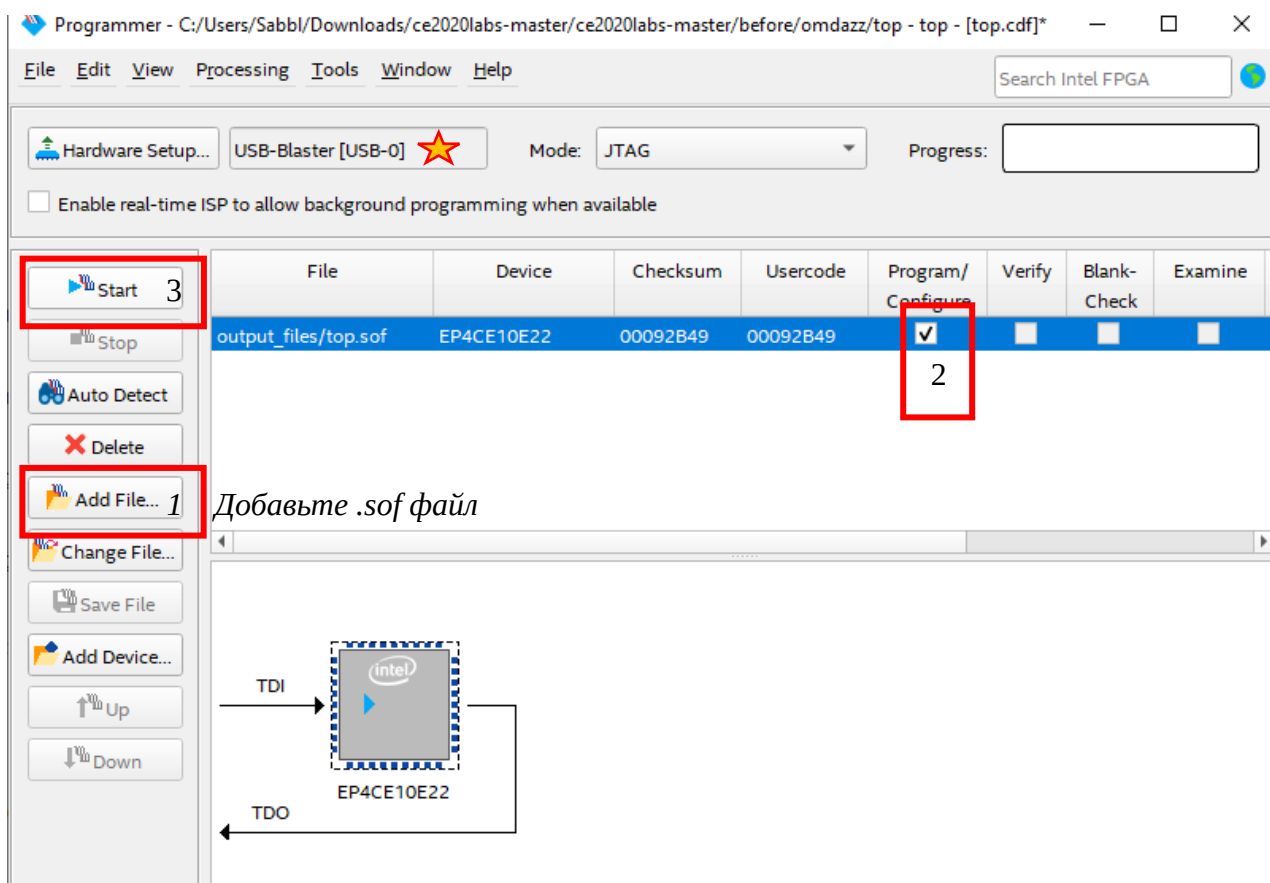
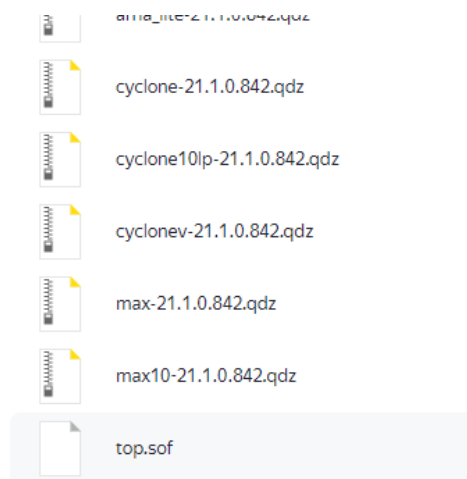
3. Далее нажимаем *Hardware setup*:



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

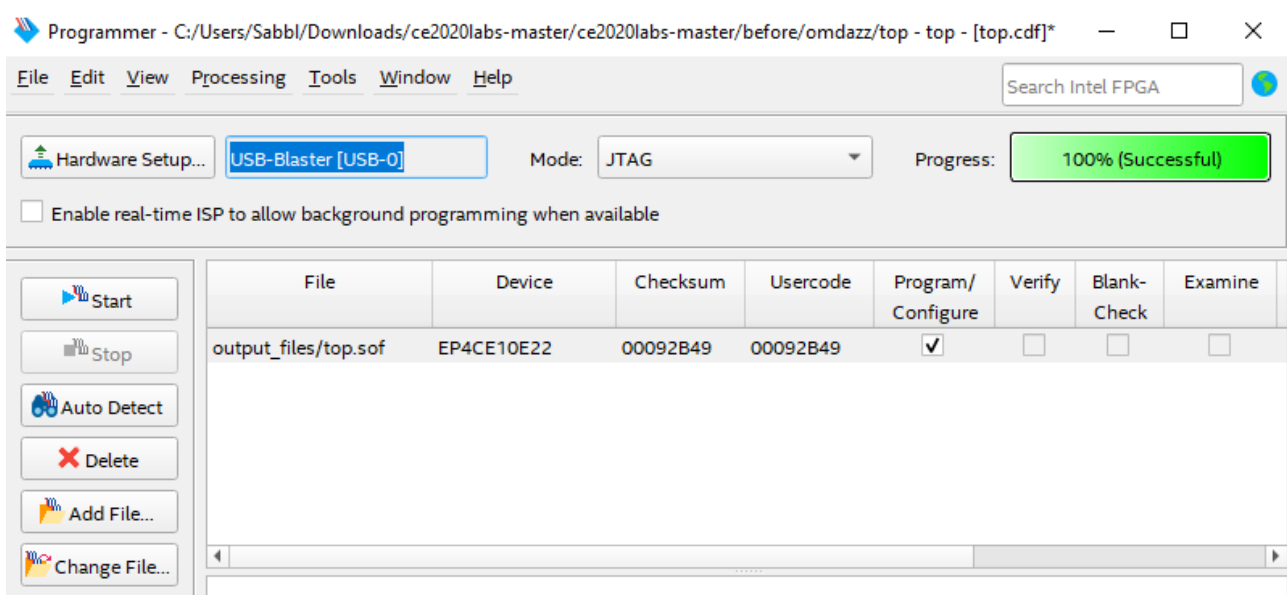


5. загрузите файл прошивки *top.sof* с Yandex диска и попробуйте загрузить его на плату **OMDAZZ** или **rzrd** с помощью того же окна programmer через *Add File* (Не забудьте про Hardware setup -> USB-Blaster). Для других плат вы можете сгенерировать новый .sof файл самостоятельно.



Е

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



## 2.4 Установка Icarus Verilog 12 и gtkwave на Linux(Обязательно)

Icarus Verilog – открытый симулятор, который подойдёт для моделирования простых дизайнов.

GTKwave – утилита для просмотра временных диаграмм в формате .VCD

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

<https://github.com/steveicarus/iverilog> - git-репозиторий для ручной установки Icarus последней версии (нам необходима 12 и выше)

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

```
sudo add-apt-repository ppa:team-electronics/ppa
sudo apt-get update
sudo apt install iverilog gtkwave
```

2. Проверьте версию Icarus:

```
iverilog -v
```

**Если не возникает проблем, то это - все что нужно сделать. Иначе лучше установить icarus с помощью git и сборки.**

3. Проверить работоспособность Icarus можно на файле с расширением «.v». Просто выберите любой Verilog-модуль и напишите команду

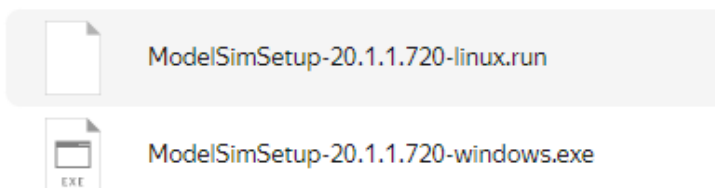
```
iverilog [имя модуля].v
```

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

## 2.5 Установка ModelSim на Linux(Опционально)

В качестве другого симулятора в лабораторных работах так же может использоваться ModelSim, его можно скачать по ссылке: <https://disk.yandex.ru/d/zuN-mJTungHEDg>

## ← ModelSim :



Предварительно установите библиотеки с помощью команд:

```
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
```

Введите следующие команды в терминале:

```
chmod +x ModelSimSetup-20.1.1.720-linux.run
./ ModelSimSetup-20.1.1.720-linux.run
```

## 2.6 Установка Questa Lite на Linux(Опционально)

Ссылка на скачивание дистрибутива Questa Lite <https://disk.yandex.lt/d/p92jXpEjIwPrBg>

Для использования **потребуется** зарегистрировать аккаунт и активировать бесплатную лицензию на сайте Intel с помощью VPN: <https://www.intel.com/content/www/us/en/docs/programmable/683472/21-4/specifying-the-license-for-the-software.html>

## 2.7 Установка RARS на Linux(Обязательно)

RARS – Это симулятор архитектуры команд RISC-V.

RARS распространяется в виде исполняемого jar-файла. Вам понадобится как минимум Java 8, чтобы запустить его:

```
sudo apt-get --yes install default-jre
```

Последний стабильный релиз можно найти здесь:

<https://github.com/TheThirdOne/rars/releases/tag/v1.6>

## 3. Установка и настройка ПО для работы с OpenLane (Опционально)

**Данный способ установки Docker+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, последовательность приведена: (<https://docs.docker.com/engine/install/ubuntu/>):

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/
apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Add the repository to Apt sources:
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

**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 docker-ce-cli containerd.io docker-buildx-plugin docker-
compose-plugin
```

```
sudo docker run hello-world
```

**В случае успешного завершения установки docker, должно появиться похожее 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 дать себе права!**

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

**После установки необходимо перелогиниться или перезагрузить машину!!!**

**Скачиваем git-репозиторий с OpenLane:**

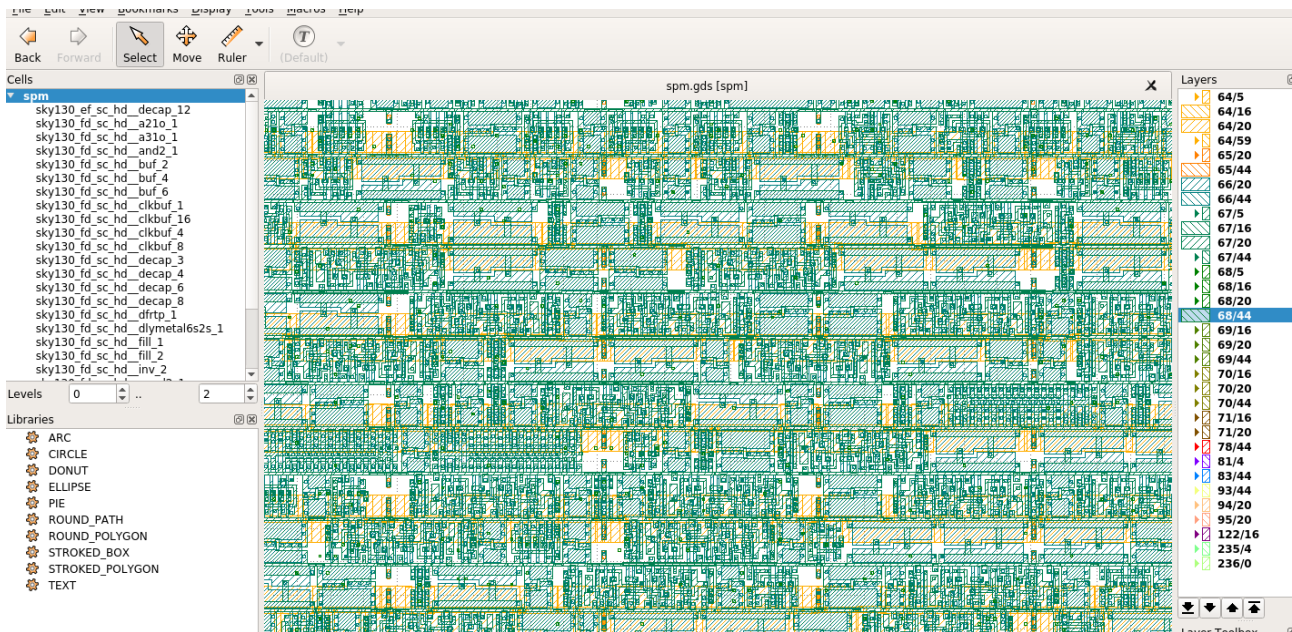
```
mkdir -p ~/github
cd ~/github
git clone https://github.com/The-OpenROAD-Project/OpenLane.git
cd OpenLane
make
```

**Тестовый запуск:**

```
make test
```

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

```
make mount
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: [В России не любят Линукс, а я не люблю виртуальные машины. Что делать? / Хабр \(habr.com\)](https://habr.com/ru/post/481111/) или написать в группу коммьюнити Школы Синтеза Цифровых схем в telegram: <https://t.me/icDesignCommunity>

#### 4. Установка виртуальной машины для занятий по функциональной верификации (обязательно).

1. Установить Virtual Box:

<https://www.virtualbox.org/wiki/Downloads>

2. Скачать архив:

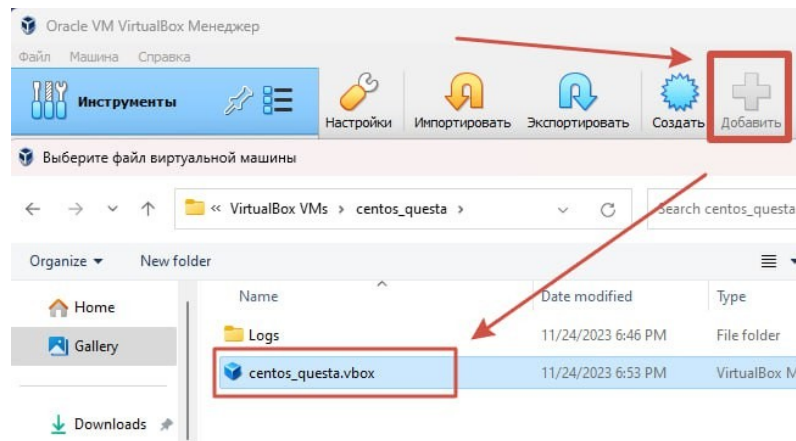
<https://drive.google.com/file/d/1SmqZ7A5l6YUjQrT-830fl8d5rPA2pqvY/view>

3. Распаковать скачанный архив.

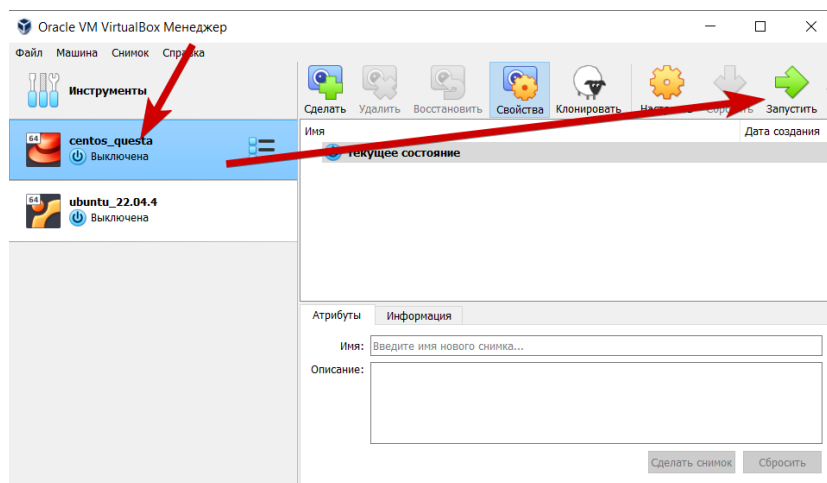
4. Запустить Virtual Box.

5. Добавить виртуальную машину через кнопку “Добавить”, выбрав в файл с расширением .vbox в директории, куда был распакован архив:

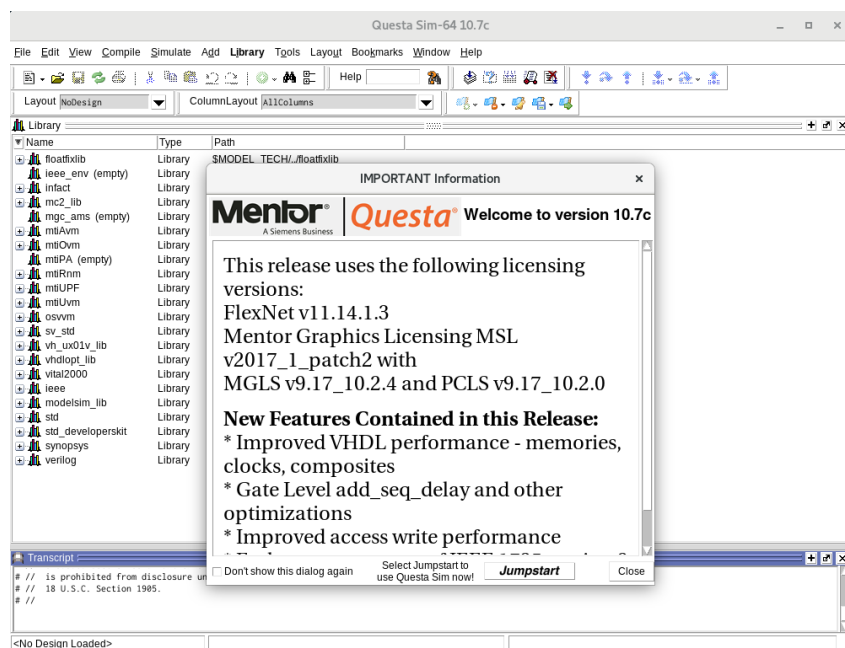




6. Запустить виртуальную машину при помощи кнопки “Запустить”. Пароль от пользователя: 1234:



7. Запустить терминал через меню Applications -> System Tools -> Terminal и ввести команду `vsim` чтобы убедиться в работоспособности симулятора. Должно появиться приветственное окно:



8. В случае возникновения проблем с установкой написать в чат Школы Синтеза Цифровых схем.

# Создание загрузаемого SSD с предустановленным ПО при помощи операционной системы Linux

Оригинал: Статья Юрия Панчула на habr: <https://habr.com/ru/articles/754262/>

**Данный образ содержит ПО для работы на отладочных платах с ПЛИС компании Altera/Intel (САПР Quartus).**

**Для установки Vivado или Gowin IDE вы можете воспользоваться доступной в интернете информацией. Если будут возникать проблемы с установкой этих САПРов инструкцию можно дополнить.**

ПО Questa в образе может требовать наличия лицензии.

## 1. Запрос образа и первоначальная настройка SSD

Прежде всего вам нужно скачать .img образ по одной из доступных ссылок:

<https://disk.yandex.ru/d/vtGdaJxk6Kzr2A>

<https://disk.yandex.ru/d/wfsVMfiAy-aYpg>

[https://disk.yandex.ru/d/\\_kS2HkMZjjEsNg](https://disk.yandex.ru/d/_kS2HkMZjjEsNg)

**ВАЖНО: Могут возникнуть проблемы с GPT!**

GPT - это Global Partition Table. Оказывается этих таблиц на диске две - основная и резервная. Файл .img (~50 GB в распакованном виде) меньше чем размер любого SSD (который сейчас начинается с 120 GB и даже 5 лет назад начинался с 60 GB). Поэтому если на SSD есть две GPT, то после записи .img резервная таблица не будет переписана. Она будет там лежать и мешать первой загрузке.

Такой проблемы нет, если диск размечен без GPT, со старыми DOS Partitions. Но для гарантии вторую GPT нужно стереть. Именно это и делает скрипт `erase_ssd_gpt_and_write_bootable_image.bash`, который находится здесь:

```
https://github.com/yuri-panchul/basics-graphics-music/blob/main/scripts/admin/erase_ssd_gpt_and_write_bootable_image.bash
```

Этот скрипт нужно поместить в директорию с образом SSD, файлом с расширением .img. Или поместить имидж в директорию со скриптом, после чего запустить, причем под `sudo`. Скрипт проверит, что имидж один, спросит, какой диск переписать, проверит, что этот диск не смонтирован, после чего скрипт запишет имидж на выбранный SSD. Операция занимает десятки минут.

## 2. Загрузка с SSD

Нужно подключить SSD к выключенному компьютеру, потом включить компьютер и начать нажимать одну из клавиш для входа в BIOS. [На разных компьютерах эти клавиши разные](#), на многих Esc или Del, но есть и другие:

**ASUS, Acer:** DEL или F2

**Dell:** F12 or F2

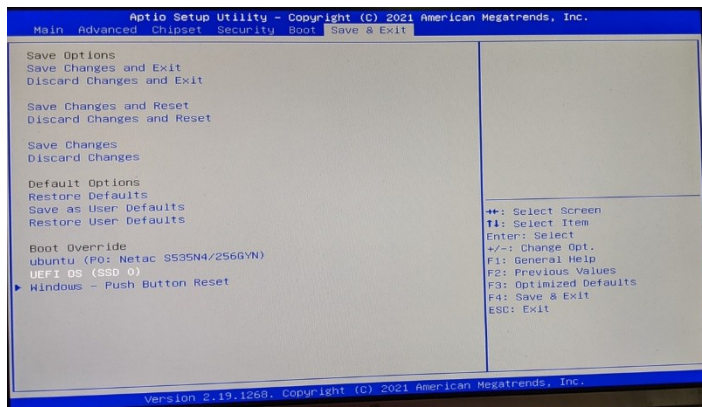
**HP:** F10

**Lenovo:** F2 или Fn + F2 (ноуты), F1 (десктопы), Enter + F1 (ThinkPad)

**MSI:** DEL

**Samsung:** F2

Если такой вход работает, то вы увидите меню вроде этого. С него можно загрузиться с SSD. SSD с Simply Linux, который мы сформировали, поддерживает и загрузку через UEFI, и более старую Legacy Boot загрузку:



**ВАЖНО:** Если клавиши при включении ПК не работают, проблемы могут быть с microsoft, попробуйте следующую опцию с win 10

Параметры

Главная

Найти параметр

Обновление и безопасность

- Центр обновления Windows
- Оптимизация доставки
- Безопасность Windows
- Служба архивации
- Устранение неполадок
- Восстановление**
- Активация
- Поиск устройства
- Для разработчиков

## Восстановление

Вернуть компьютер в исходное состояние

Если ваш компьютер работает нестабильно, попробуйте восстановить его исходное состояние. Вы сможете при этом сохранить или удалить личные файлы, а затем переустановить Windows.

Начать

## Особые варианты загрузки

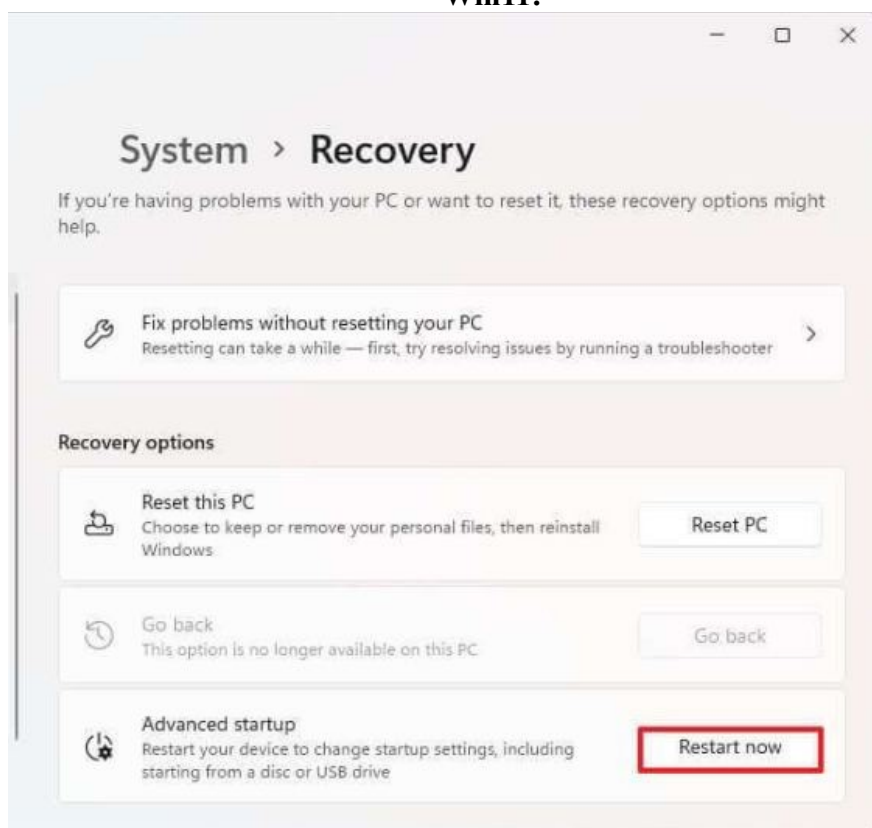
Запустите систему с устройства или диска (например, USB-накопителя или DVD-диска), измените параметры встроенного ПО компьютера, настройте загрузку Windows или восстановите систему из образа. Ваш компьютер перезагрузится.

Перезагрузить сейчас

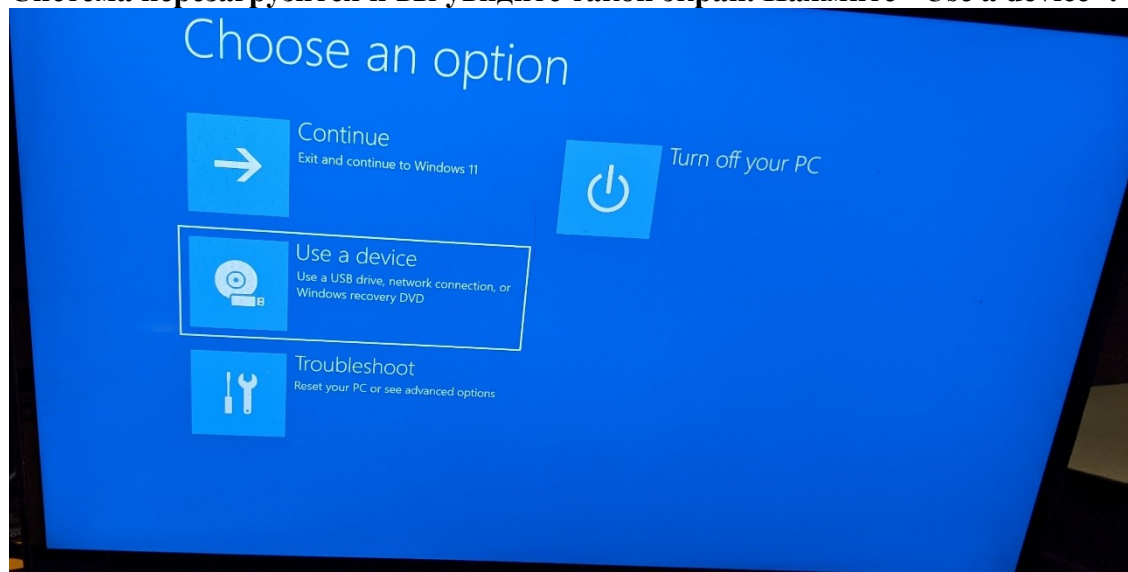
## Дополнительные параметры восстановления

[Узнайте, как начать заново с чистой установкой Windows](#)

## Win11:



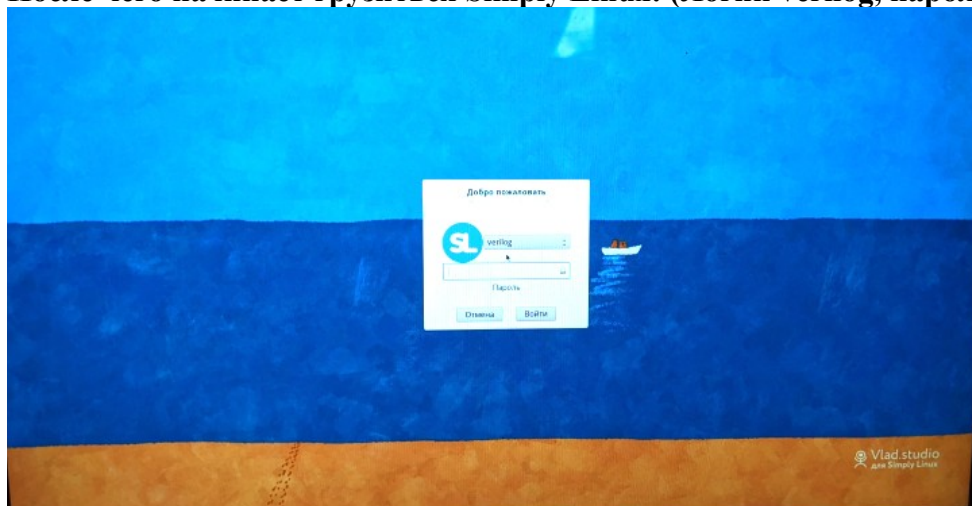
**Система перезагрузится и вы увидите такой экран. Нажмите "Use a device":**



**Теперь нажмите "Linpus lite". "EFI USB Device" тоже может работать, но не всегда:**



**После чего начинает грузиться Simply Linux: (Логин verilog, пароль verilog)**



### 3. Обновление репозитория

После загрузки нужно обновить репозитории с примерами на verilog. Для этого нужно подключиться к интернету, открыть терминал и проверить, каких из нужных пакетов не хватает (их нужно клонировать), а какие уже есть (их нужно обновить). Например если директория `~/projects/basics-graphics-music` уже существует, то нужно ввести команды:

```
cd ~/projects/basics-graphics-music
git pull
```

Но если директории `~/projects/basics-graphics-music` нет, то

```
cd ~/projects
git clone https://github.com/yuri-panchul/basics-graphics-music.git
```

Аналогичным образом нужно поступить с репозиториями:

<https://gitflic.ru/project/yuri-panchul/valid-ready-etc>

<https://gitflic.ru/project/yuri-panchul/systemverilog-homework>



## 4. Проверка работы с FPGA платой

Для начала скажем компьютеру, какая у нас плата. Для этого мы запустим скрипт `check_setup_and_choose_fpga_board.bash`

```
cd ~/projects/basics-graphics-music
./check_setup_and_choose_fpga_board.bash
```

Скрипт выведет список поддерживаемых плат и спросит, какую из них вы хотите использовать. Выберем номер 6, плату Omdazz:

```
[verilog@host-15 basics-graphics-music]$ cd ~/projects/basics-graphics-music
[verilog@host-15 basics-graphics-music]$ ./check_setup_and_choose_fpga_board.bash
check_setup_and_choose_fpga_board.bash: Please select an FPGA board among the following supported:
1) 000_todo      4) de10_lite      7) p1words6      10) tangprimer20k
2) c5gx          5) de2_115        8) rzrd          11) zeowaa
3) de0_cv        6) omdazz         9) saylinx       12) exit
Your choice (a number): 6
```

Скрипт сообщит имя выбранной платы и сохранит ваш выбор в текстовом файле в верхней директории репозитории:

```
Your choice (a number): 6
check_setup_and_choose_fpga_board.bash: FPGA board selected: omdazz
check_setup_and_choose_fpga_board.bash: Created an FPGA board selection file: "/home/verilog/projects/basics-graphics-music/fpga_board_selection"
```

Далее скрипт задаст вопрос, хотите ли вы, чтобы он создал временные рабочие директории для всех примеров репозитория на основе вашего выбора платы. Если не любите скрипты и собираетесь далее работать только с графическим интерфейсом программы синтеза Quartus, ответьте Y ("Да"). Если же вы собираетесь далее использовать скрипты - все равно как отвечать.

```
Would you like to create the new run directories for the synthesis of all labs in the package, based
on your FPGA board selection? We recommend to do this if you plan to work with Quartus GUI rather tha
n with the synthesis scripts. [y/n] y

check_setup_and_choose_fpga_board.bash: Setting up: "/home/verilog/projects/basics-graphics-music/lab
s/12_snail_fsm/run"
check_setup_and_choose_fpga_board.bash: Setting up: "/home/verilog/projects/basics-graphics-music/lab
s/03_decoder/run"
```

Теперь заходим в поддиректорию `labs/01_and_or_not_xor_de_morgan` и видим там скрипты для каждого шага работы с примером. Запустим скрипт `03_synthesize_for_fpga.bash` для синтеза с выбранной FPGA платой.

```
cd labs/01_and_or_not_xor_de_morgan
./03_synthesize_for_fpga.bash
```

Оно должно синтезировать код примера на верилоге и прошить память конфигурации FPGA.

Иногда оно не прошивает с первого раза. Выглядит это так:

```
Info: Quartus Prime Shell was successful. 0 errors, 65 warnings
Info: Peak virtual memory: 496 megabytes
Info: Processing ended: Sun Aug 13 19:07:19 2023
Info: Elapsed time: 00:00:20
Info: Total CPU time (on all processors): 00:00:17
03_synthesize_for_fpga.bash: using cable USB-Blaster [1-7.3]
Error (213019): Can't scan JTAG chain. Error code 87.
```

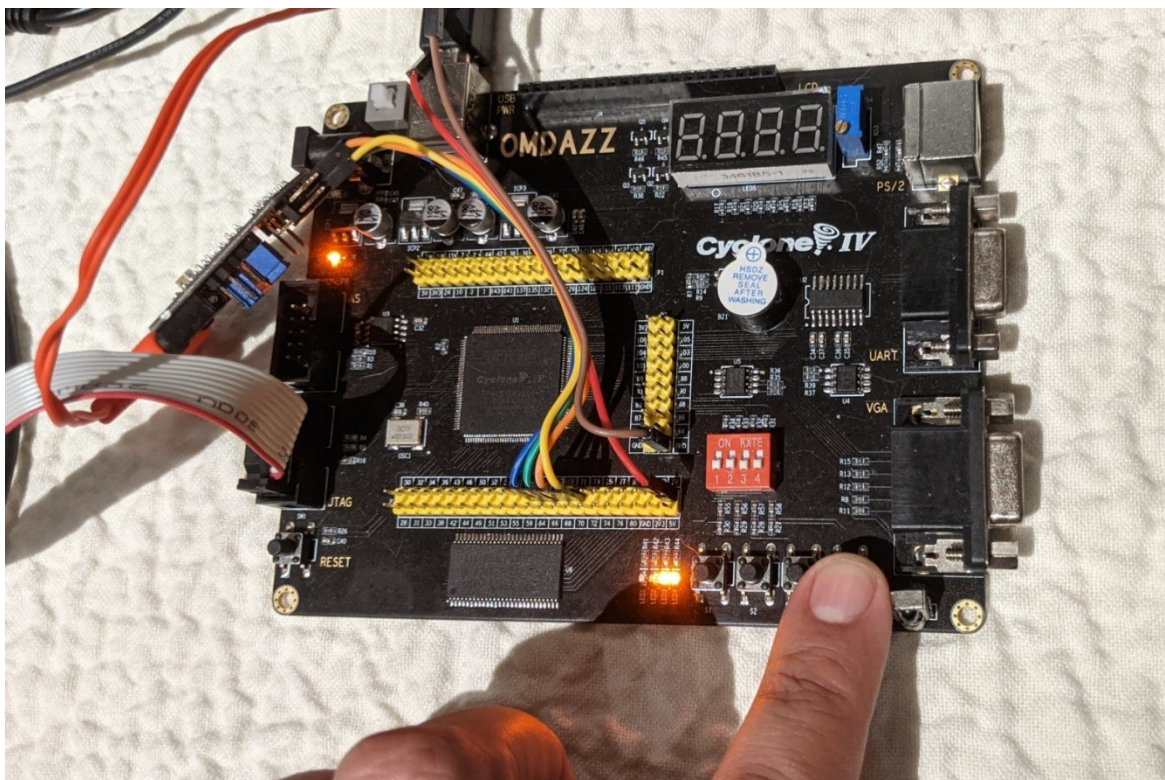
Тогда нужно перемотнуть USB шнур в USB Blaster, после чего запустить скрипт `./04_configure_fpga.bash`. Скорее всего все получится, если выбрана правильная плата:

```

[verilog@host-15 01_and_or_not_xor_de_morgan]$ ./04_configure_fpga.bash
04_configure_fpga.bash: using cable USB-Blaster [1-7.3]
Info: *****
Info: Running Quartus Prime Programmer
Info: Command: quartus_pgm --no_banner -c "USB-Blaster [1-7.3]" --mode=jtag -o P:fpga_project.sof
Info (213045): Using programming cable "USB-Blaster [1-7.3]"
Info (213011): Using programming file fpga_project.sof with checksum 0x00094EB6 for device EP4CE6E22@1
Info (209060): Started Programmer operation at Sun Aug 13 19:08:10 2023
Info (209016): Configuring device index 1
Info (209017): Device 1 contains JTAG ID code 0x020F10DD
Info (209007): Configuration succeeded -- 1 device(s) configured
Info (209011): Successfully performed operation(s)
Info (209061): Ended Programmer operation at Sun Aug 13 19:08:13 2023
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
Info: Peak virtual memory: 325 megabytes
Info: Processing ended: Sun Aug 13 19:08:13 2023
Info: Elapsed time: 00:00:04
Info: Total CPU time (on all processors): 00:00:01

```

Теперь вы можете понажимать кнопки на плате и посмотреть на изменение светодиодов, в соответствии с кодом на verilog (пример плата OMDAZZ):



# Использование виртуальной машины с предустановленным ПО

Данная ВМ содержит ПО для работы на отладочных платах с ПЛИС компании Altera/Intel (САПР Quartus).

Для установки Vivado или Gowin IDE вы можете воспользоваться доступной в интернете информацией. Если будут возникать проблемы с установкой этих САПРов инструкцию можно дополнить.

ПО Questa в образе может требовать наличия лицензии.

Прежде всего вам нужно скачать .img образ по одной из доступных ссылок:

<https://disk.yandex.ru/d/vtGdaJxk6Kzr2A>

<https://disk.yandex.ru/d/wfsVMfiAy-aYpg>

[https://disk.yandex.ru/d/\\_kS2HkMZjjEsNg](https://disk.yandex.ru/d/_kS2HkMZjjEsNg)

После получения образа в формате .img необходимо конвертировать его в virtual disk image (.VDI) формат с помощью qemu-img.

Для этого нужно скачать утилиту с сайта: [qemu-img for WIndows - Cloudbase Solutions](#)

Распаковать архив и с помощью Windows PowerShell сконвертировать `img->vdi` следующим образом:

```
.\qemu-img-win-x64-2_3_0\qemu-img.exe convert -O vdi .\slinux.img .\silinux.vdi
```

После чего необходимо загрузить Virtual Box+Extension pack с сайта [Downloads – Oracle VM VirtualBox](#)

## VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.1 packages, see [VirtualBox 6.1 builds](#). Version 6.1 will remain supported until December 2023.

## VirtualBox 7.0.10 platform packages

- [Windows hosts](#)
- [macOS / Intel hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)
- [Solaris 11 IPS hosts](#)

The binaries are released under the terms of the GPL version 3.

See the [changelog](#) for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

- [SHA256 checksums](#), [MD5 checksums](#)

**Note:** After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

## VirtualBox 7.0.10 Oracle VM VirtualBox Extension Pack

- [All supported platforms](#)

Support VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See [this chapter from the User Manual](#) for an introduction to this Extension Pack. The Extension Pack binaries are released under the [VirtualBox Personal Use and Evaluation License \(PUEL\)](#). Please install the same version extension pack as your installed version of VirtualBox.

## VirtualBox 7.0.10 Software Developer Kit (SDK)

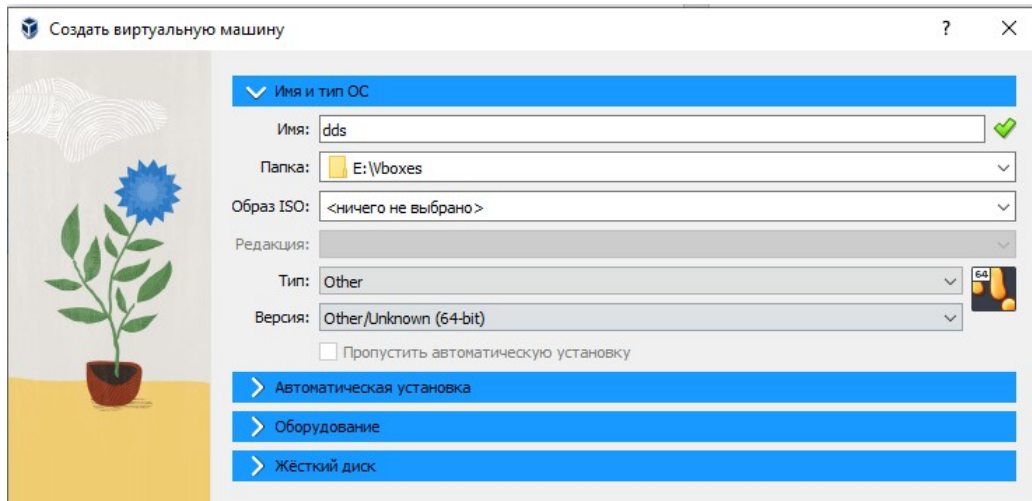
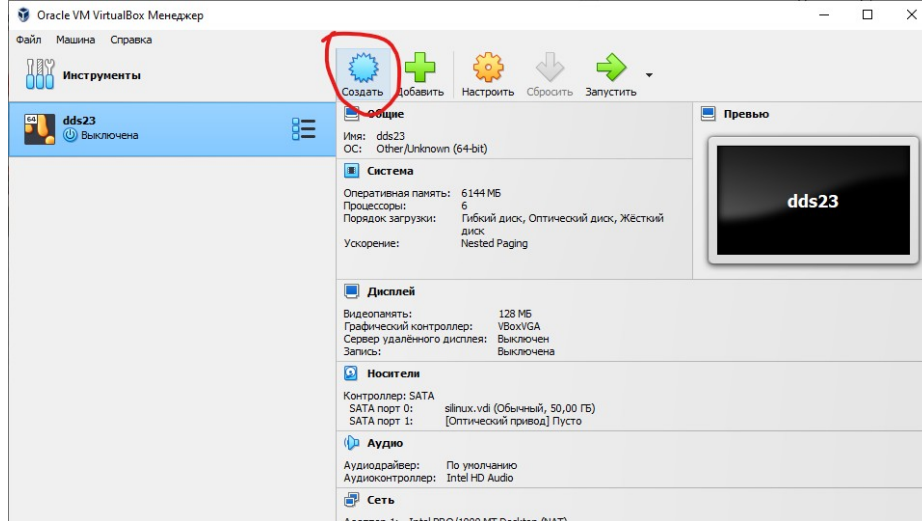
- [All platforms](#)

После установки virtual box, нужно доустановить extension pack сделав двойной клик по нему. Он необходим для возможности использовать буфер обмена между Хост и ВМ и

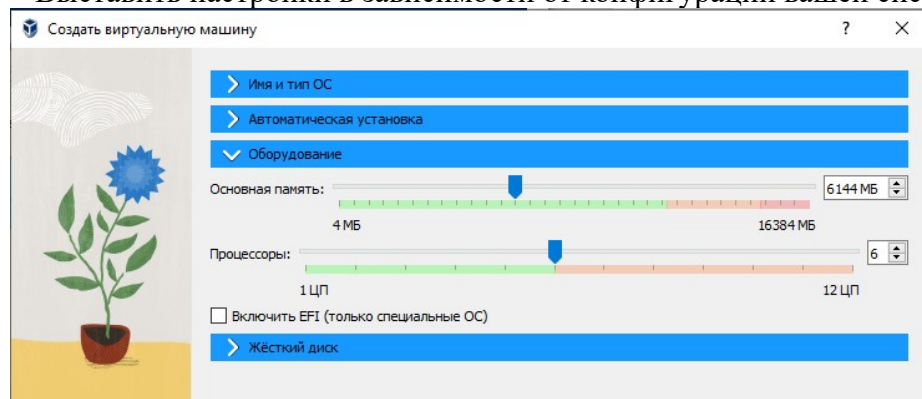


других преимуществ.

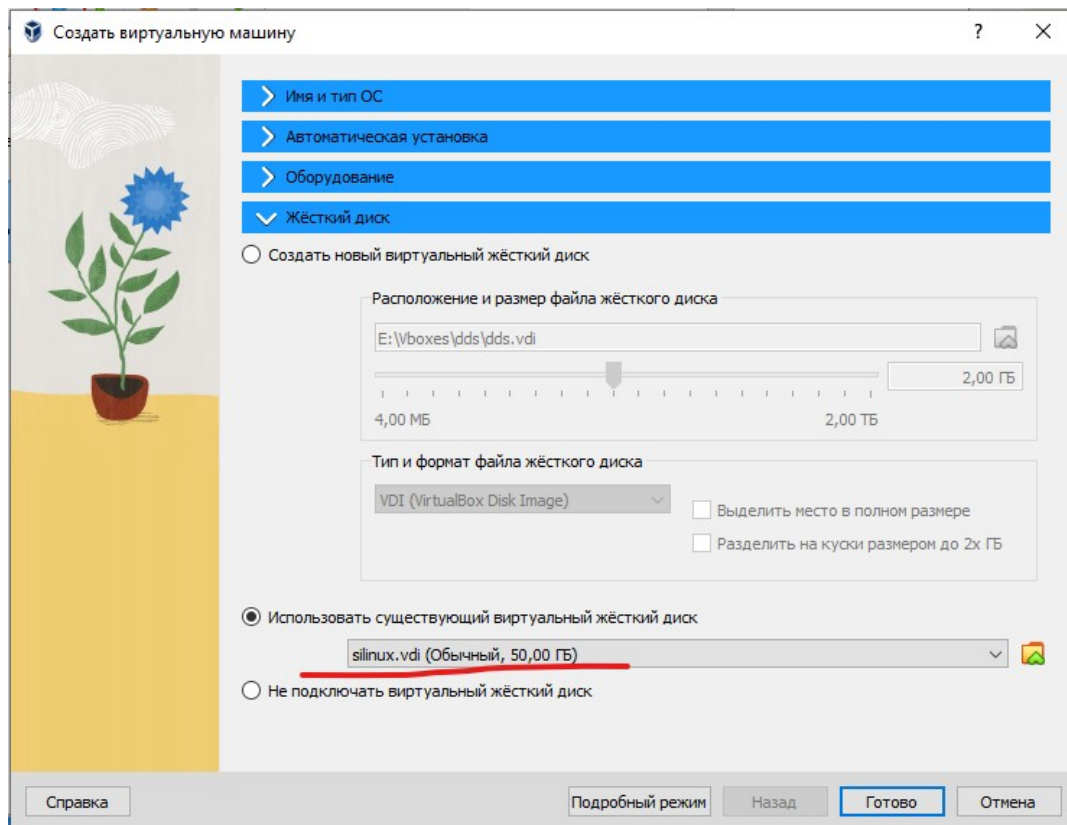
При первом запуске VBox нужно создать виртуальную машину:



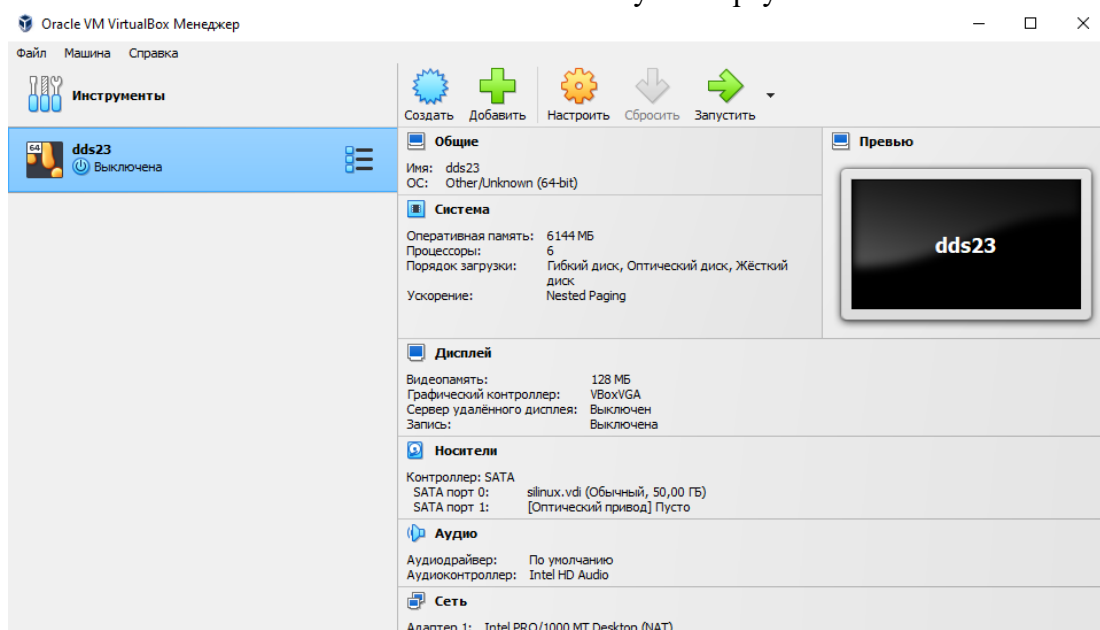
Выставить настройки в зависимости от конфигурации вашей системы:



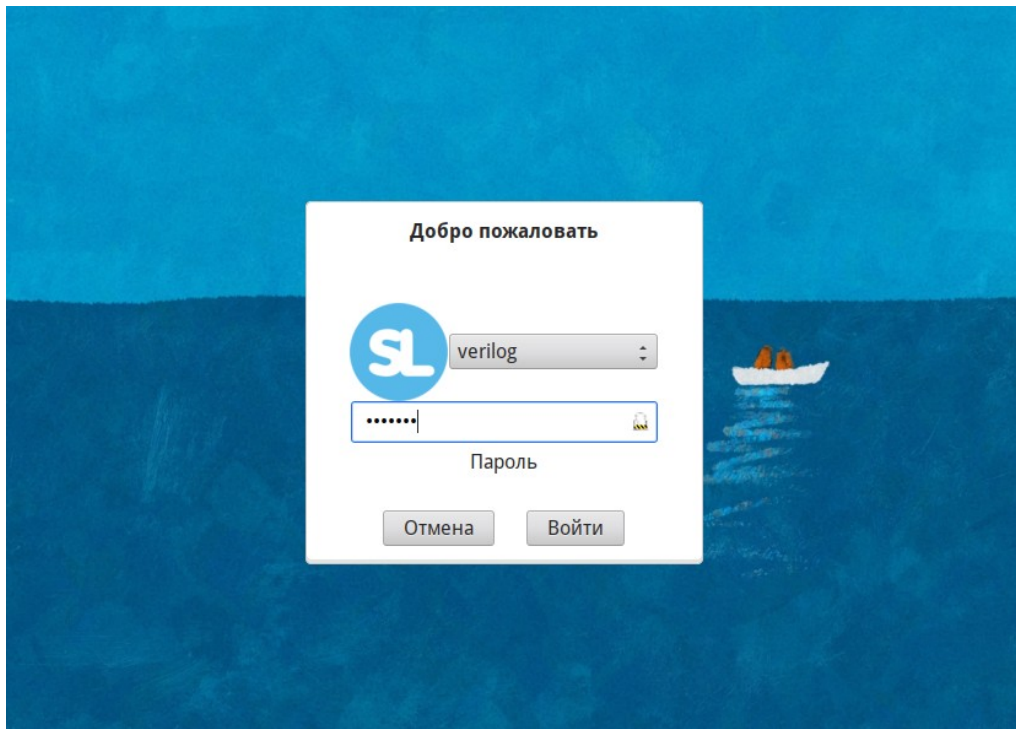
И выбрать существующий файл с расширением .vdi:



После чего появится возможность запуска виртуальной машины:

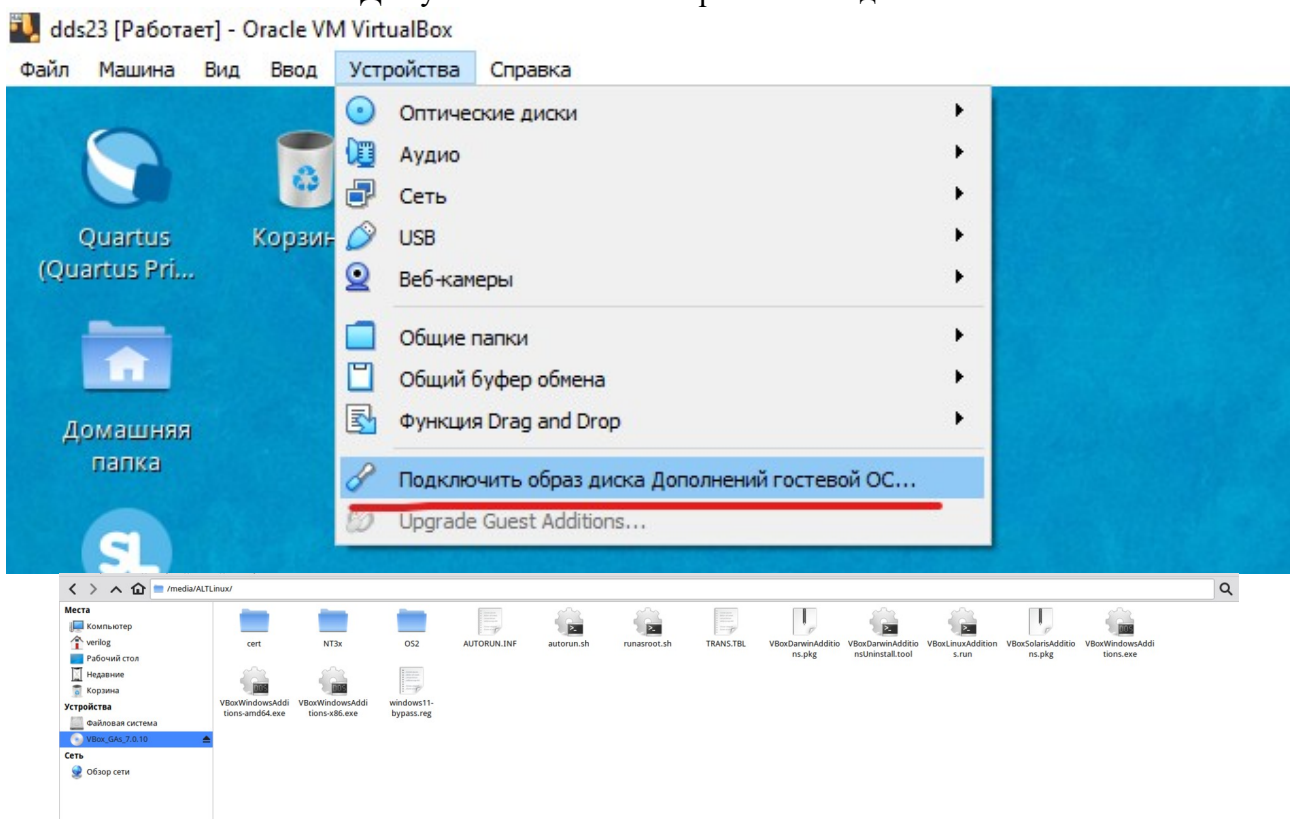


**Логин: verilog, Пароль: verilog**



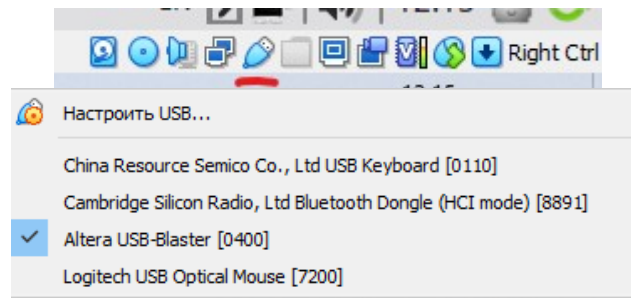
**Опционально можно установить extension pack – расширение для более комфортного использования ВМ(поддержка буфера обмена между хостом и ВМ и т.д)**

Для установки extension pack необходимо:



в консоли: `sh ./autorun.sh`

После чего виртуальная машина готова к использованию.  
Для проброса USB Altera Blaster на виртуальную машину:



Дальнейшие шаги аналогичны работе с образом с SSD.

## Дополнительная информация

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

1. Телеграм-канал коммунити Школы цифрового синтеза (Здесь вы сможете найти ответы на интересующие вас вопросы или задать свой)

<https://t.me/icDesignCommunity>

2. Основные Гит-репозитории Школы синтеза цифровых схем (2023-2025)

git clone <https://github.com/yuri-panchul/basics-graphics-music.git>

- репозиторий с лабораторными работами школы синтеза

git clone <https://gitflic.ru/project/yuri-panchul/systemverilog-homework.git>

- репозиторий с домашними заданиями

## FAQ

1) Список поддерживаемых плат: [basics-graphics-music/boards at main · yuri-panchul/basics-graphics-music](https://github.com/yuri-panchul/basics-graphics-music/tree/main/boards) [GitHub](#)

**САПРы для синтеза: Xilinx(AMD) Vivado, Intel(Altera) Quartus, GowinIDE**  
**На момент сентября 2024:**

- 1) ALINX AX4010 (Altera Cyclone IV)
- 2) Digilent Arty A7 (Xilinx Artix-7)
- 3) Digilent Basys 3 (Xilinx Artix-7)
- 4) Digilent CMod A7 (Xilinx Artix-7), Частичная поддержка
- 5) Digilent CMod S7 (Xilinx Spartan-7), Частичная поддержка
- 6) Digilent Nexys A7 (Xilinx Artix-7)
- 7) Digilent Nexys4 (Xilinx Artix-7)
- 8) Digilent Nexys4 DDR (Xilinx Artix-7)
- 9) Digilent ZYBO Z7 (Xilinx Zynq-7000)
- 10) emooc.cc (Altera Cyclone IV)
- 11) Karnix ASB-254 (Lattice ECP5 25K)
- 12) Olimex ICE40HX8K-EVB (Lattice iCE40 HX FPGA)
- 13) Omdazz (Altera Cyclone IV)
- 14) Orange Crab (Lattice ECP5)

- 15) Piswords-6 (Altera Cyclone IV)
- 16) RzRd (Altera Cyclone IV)
- 17) Saylinx (Altera Cyclone IV)
- 18) Tang Nano 1K (Gowin 1), Частичная поддержка
- 19) Tang Nano 20K (Gowin 2)
- 20) Tang Nano 4K (Gowin 1)
- 21) Tang Nano 9K (Gowin 1)
- 22) Tang Primer 20K Dock (Gowin 2)
- 23) Tang Primer 20K Lite (Gowin 2)
- 24) Tang Primer 25K Dock (Gowin 5)
- 25) Tang Mega 138K Dock (Gowin 5)
- 26) Tang Mega 138K Dock Pro (Gowin 5)
- 27) Terasic Cyclone V GX Starter Kit (Altera Cyclone V)
- 28) Terasic DE0 (Altera Cyclone III)
- 29) Terasic DE0-CV (Altera Cyclone V)
- 30) Terasic DE0-Nano (Altera Cyclone IV)
- 31) Terasic DE0-Nano-SoC (Altera Cyclone V)
- 32) Terasic DE1 (Altera Cyclone II)
- 33) Terasic DE1-SoC (Altera Cyclone V)
- 34) Terasic DE10-Lite (Altera MAX10)
- 35) Terasic DE10-Nano (Altera Cyclone V)
- 36) Terasic DE2 (Altera Cyclone II)
- 37) Terasic DE2-115 (Altera Cyclone IV)
- 38) Terasic DK-DEV-3C120N (Altera Cyclone III), Частичная поддержка
- 39) ZEOWAA (Altera Cyclone IV)