

FPGA Vendor Tools Installation Guide

Version 1.2

Revision History

Revision	Description of Change	Date
v1.1	Initial Release	3/2017
v1.2	Updated for Release 1.2	8/2017

Table of Contents

1	References	4
2	OpenCPI Vendor Tool Prerequisites	5
3	Xilinx Toolset Installation and Configuration	6
3.1	Xilinx Vivado Installation in CentOS 6/7	6
3.1.1	Xilinx Vivado 2017.1 Installation in CentOS 6/7	6
3.1.2	OpenCPI Considerations	8
3.1.3	Xilinx Vivado 2013.4 SDK Only Installation in CentOS 6/7	8
3.2	Xilinx ISE 14.7 Installation in CentOS 6/7	11
3.2.1	OpenCPI Considerations	14
3.3	Xilinx LabTools 14.7 Installation in CentOS 6/7	15
3.3.1	OpenCPI Considerations	17
3.4	Xilinx Toolset Licensing	18
3.4.1	Note on node-locked licenses in CentOS 7	19
3.4.2	OpenCPI Considerations	19
3.5	Xilinx Cable Driver Installation in CentOS 6/7	20
3.5.1	Vivado	20
3.5.2	ISE	20
3.5.3	Testing Cable Driver Installation	21
4	Altera Toolset Installation and Configuration	22
4.1	Altera Quartus Prime 15.1 Installation in CentOS 7	22
4.1.1	OpenCPI Considerations	26
4.2	Licensing Notes	27

1 References

This document assumes a basic understanding of the Linux command line (or “shell”) environment. A working knowledge of OpenCPI is required for understanding what vendor tools are necessary to perform various operations. However, no OpenCPI knowledge is required to perform the toolset installation and configuration herein. The reference(s) in Table 1 can be used as an overview of OpenCPI and may prove useful.

Title	Published By	Link
Getting Started	ANGRYVIPER Team	Getting_Started.pdf
Installation Guide	ANGRYVIPER Team	RPM_Installation_Guide.pdf
Acronyms and Definitions	ANGRYVIPER Team	Acronyms_and_Definitions.pdf
Overview	OpenCPI	https://goo.gl/RskxiV

Table 1: References

2 OpenCPI Vendor Tool Prerequisites

One or more third party vendor toolsets are required to perform various OpenCPI operations. Table 2 identifies the various possible vendor tool installation and license requirement combinations. Each combination enumerates the associated functionality that the given combination provides to OpenCPI. OpenCPI supports Xilinx Vivado 2017.1 with Xilinx Vivado (SDK Only) 2013.4 (explained in 3.1). Testing has been done with Vivado 2015.4, but future regression testing will be performed against 2017.1. OpenCPI also supports Xilinx ISE 14.7, Xilinx LabTools 14.7, and Altera Quartus 15.1.

Table 2: Support with Vendor Tools

Tool	Installation	Supported simulators	Load bitstreams onto	Run applications on these platforms	Build bitstreams for	Build software for
No vendor tools			Zynq ¹	Zynq-based ² , x86-only ¹		x86 ¹
Xilinx Vivado	2017.1 with WebPACK License	xsim			Zynq ³	
	2013.4 (SDK only)					Zynq-ARM
	2017.1 <i>and</i> 2013.4 SDK with WebPACK License	xsim			Zynq ³	Zynq-ARM
Xilinx LabTools 14.7			ML605	x86/ML605		
Xilinx ISE 14.7	WebPACK License	isim	ML605	x86/ML605	Zynq ³	Zynq-ARM
	Full License	isim	ML605	x86/ML605	Zynq, ML605	Zynq-ARM
Altera Quartus 15.1 with License		modelsim	ALST4	x86/ALST4	ALST4	

¹With OpenCPI installed, no additional software is required to load bitstreams onto Zynq FPGAs, run applications on Zynq-based or x86-only platforms, or build software for x86.

²“Zynq-based” platform includes both a Zynq’s FPGA and ARM PS. The usage of “Zynq” or “Zynq-based” here does not include Zynq UltraScale devices.

³Building bitstreams with a WebPACK license is limited to certain Zynq parts. Refer to the vendor’s documentation for further information.

3 Xilinx Toolset Installation and Configuration

3.1 Xilinx Vivado Installation in CentOS 6/7

In order to use Vivado with OpenCPI, it is required that you install Vivado 2017.1 *and* Vivado 2013.4's SDK. The 2013.4 SDK is necessary because OpenCPI's "xilinx13.3" software platform requires an SDK with matching glibc/glibc++ versions. An SDK meeting this requirement can be found explicitly in either ISE 14.7 or Vivado 2013.4 SDK. For more information on this requirement you can reference the README for the xilinx13.3 software platform. This is located in the CDK (*e.g.*: /opt/opencpi/cdk/platforms/xilinx13_3).

3.1.1 Xilinx Vivado 2017.1 Installation in CentOS 6/7

1. Download the Vivado 2017.1 installation files from Xilinx's download site: <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2017-1.html>. A Xilinx account will be required.



Figure 1: Xilinx Vivado 2017.1 Download

2. If installing Xilinx tools in a permission-restricted directory, you may need to change the umask temporarily:
% sudo su -
% umask 0002
3. Extract the tarball:
% tar -xf Xilinx_Vivado_SDK_2017.1.0415.1.tar.gz
4. Enter the resulting directory and run the installer:
% cd Xilinx_Vivado_SDK_2017.1.0415.1
% ./xsetup

5. Run through the installation process. Refer to the images below when applicable.

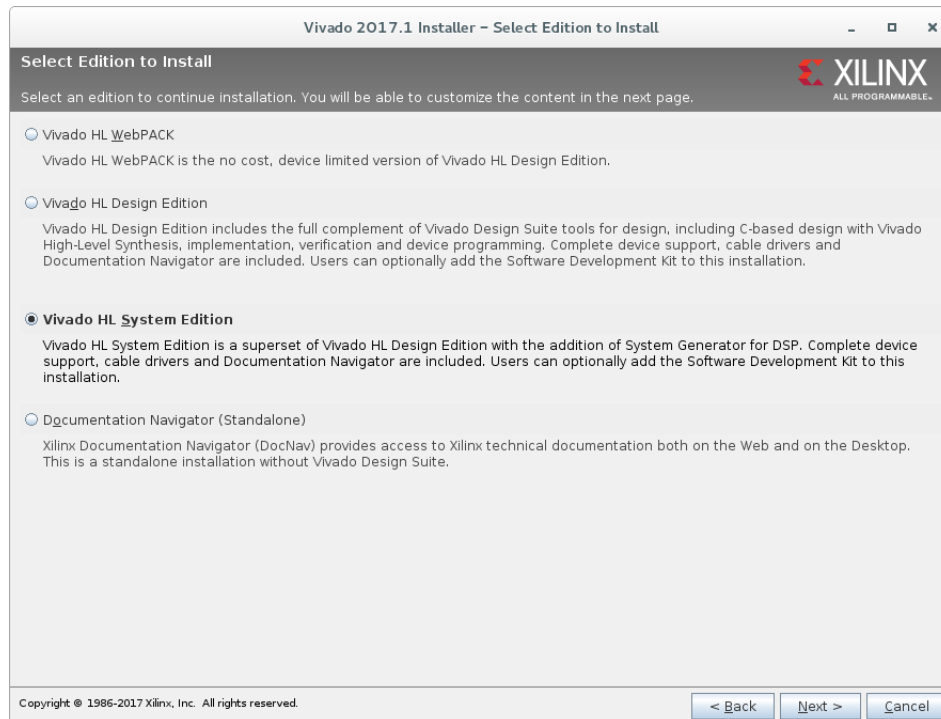


Figure 2: Xilinx Vivado Installer

We do not direct you to acquire a license, but if you do not already have one, you will need to select “Acquire or Manage a License Key” in the image below.

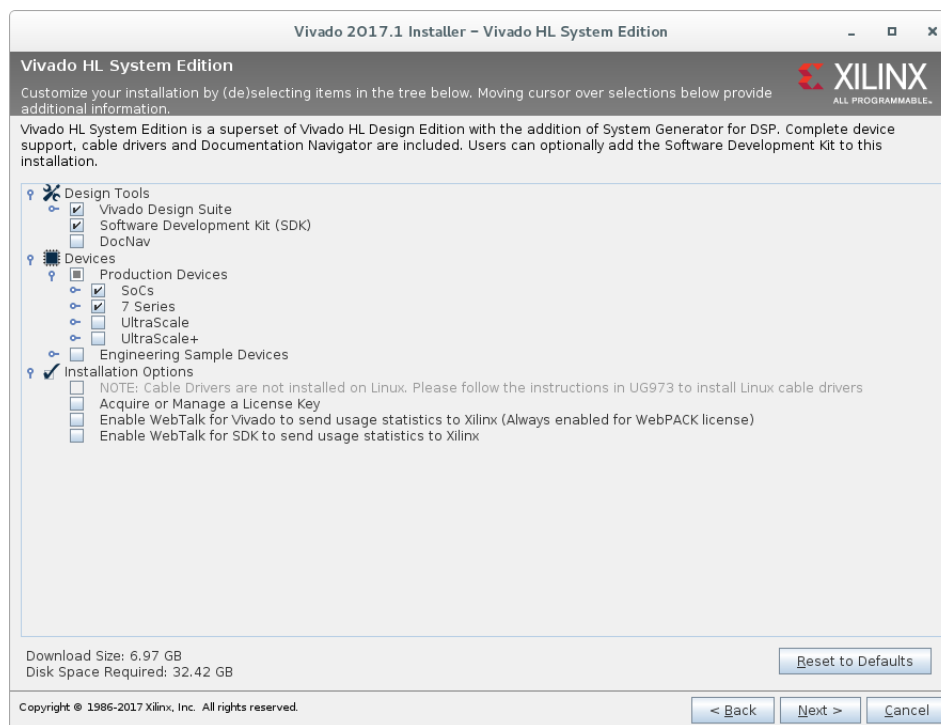


Figure 3: Xilinx Vivado Installation Choice

Take note of the installation directory chosen (e.g. `/opt/Xilinx`) as well as the Vivado version (e.g. 2017.1) for later use.

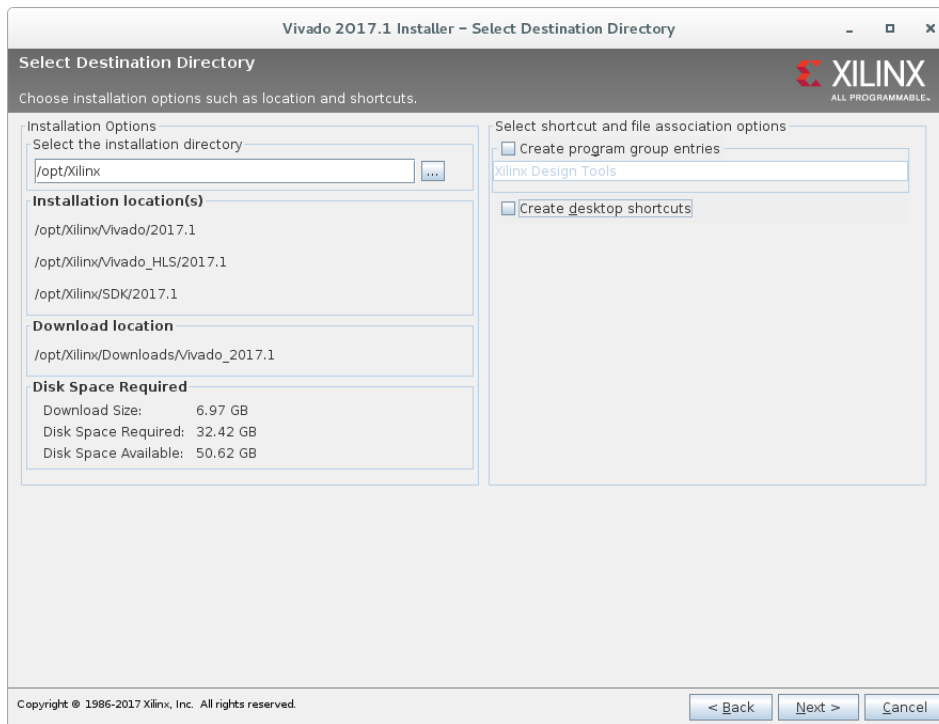


Figure 4: Xilinx Vivado Install Location

3.1.2 OpenCPI Considerations

- Note that sourcing the "`<Vivado-install-dir>/Vivado/<Vivado-version>/settings64.sh`" script will interfere with OpenCPI's environment setup. Accordingly, it is recommended to always source these scripts and execute any follow-on commands in a *separate terminal*.
- To use OpenCPI with any Xilinx Vivado installation, it is required to set the following environment variables before running OpenCPI commands. Note that each of the following **export** statements is only necessary under the following conditions:
 - When using a non-default installation location (i.e. anything other than `/opt/Xilinx`)
 - When Vivado *and* ISE are both being used and are installed in different locations
 - Or when multiple versions of Vivado are installed and you wish to use a version other than the newest.

```
% export OCPI_XILINX_VIVADO_DIR=<Vivado-install-dir>
```

```
% export OCPI_XILINX_VIVADO_VERSION=<Vivado-version>
```

If OpenCPI has been installed prior to the Vivado installation, and it is desired to make the aforementioned environment variables set automatically upon login for all users, the variables should be added in `/opt/opencpi/cdk/env.d/xilinx.sh`. Logging out and logging back into the user account will apply said variables.

3.1.3 Xilinx Vivado 2013.4 SDK Only Installation in CentOS 6/7

- Download the Vivado 2013.4 Standalone SDK installation files from Xilinx's download site: <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html>. Navigate to "2013.4" → "Software Development Kit". A Xilinx account will be required.

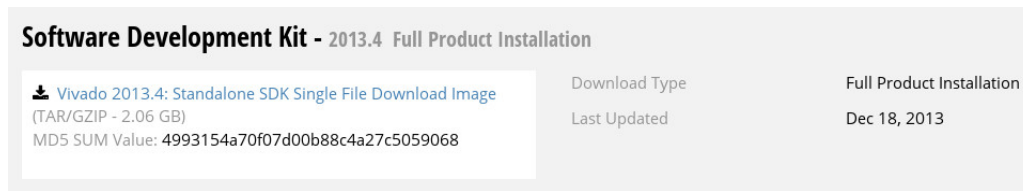


Figure 5: Xilinx Vivado 2013.4 SDK Download

2. If installing Xilinx tools in a permission-restricted directory, you may need to change the umask temporarily:
% `sudo su -`
% `umask 0002`
3. Extract the tarball:
% `tar -xf Xilinx_SDK_2013.4_1210_1.tar`
4. Enter the resulting directory and run the installer:
% `cd Xilinx_SDK_2013.4_1210_1`
% `./xsetup`

5. Run through the installation process. Refer to the images below when applicable.

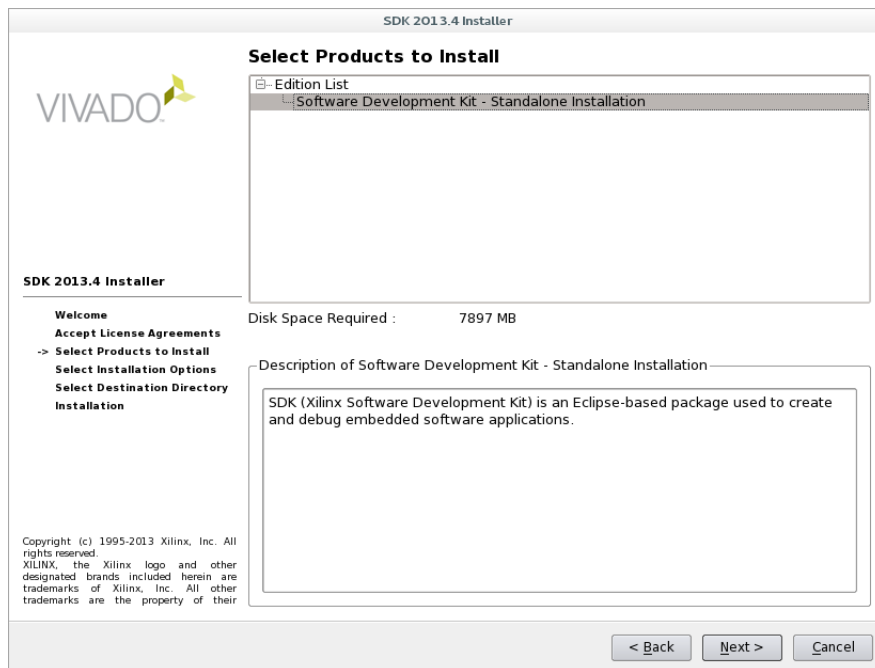


Figure 6: Xilinx Vivado SDK Installer

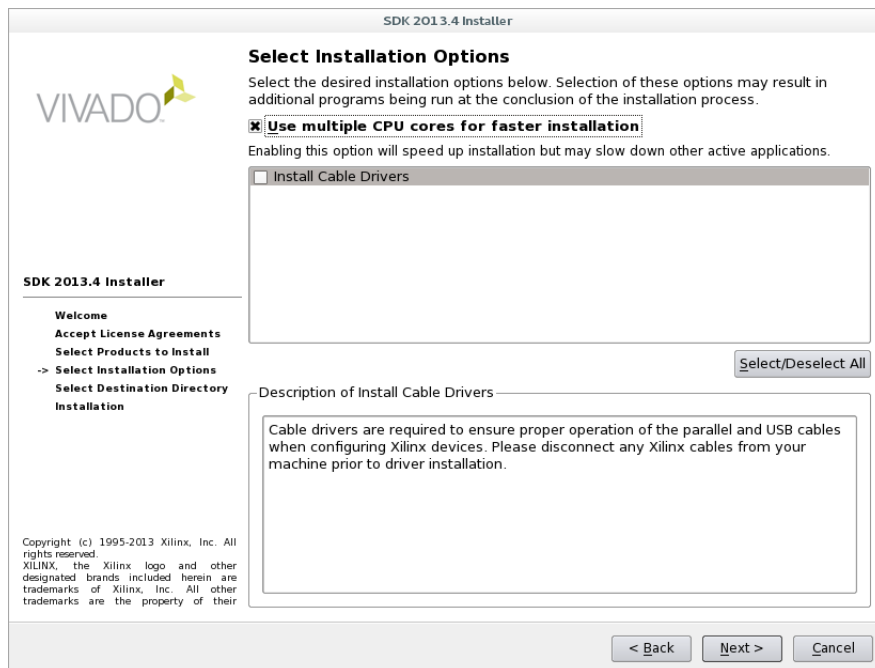


Figure 7: Xilinx Vivado SDK Installation Choice

Take note of the installation directory chosen (e.g. /opt/Xilinx) as well as the Vivado version (e.g. 2013.4) for later use.

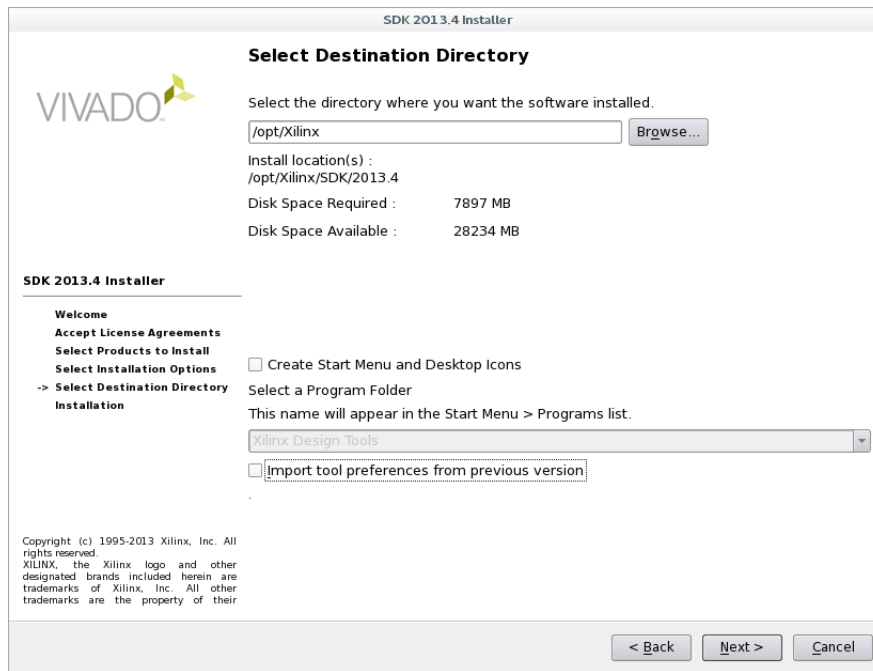


Figure 8: Xilinx Vivado SDK Install Location

3.2 Xilinx ISE 14.7 Installation in CentOS 6/7

1. Download the ISE 14.7 installation files from Xilinx's download site: <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html>. A Xilinx account will be required.

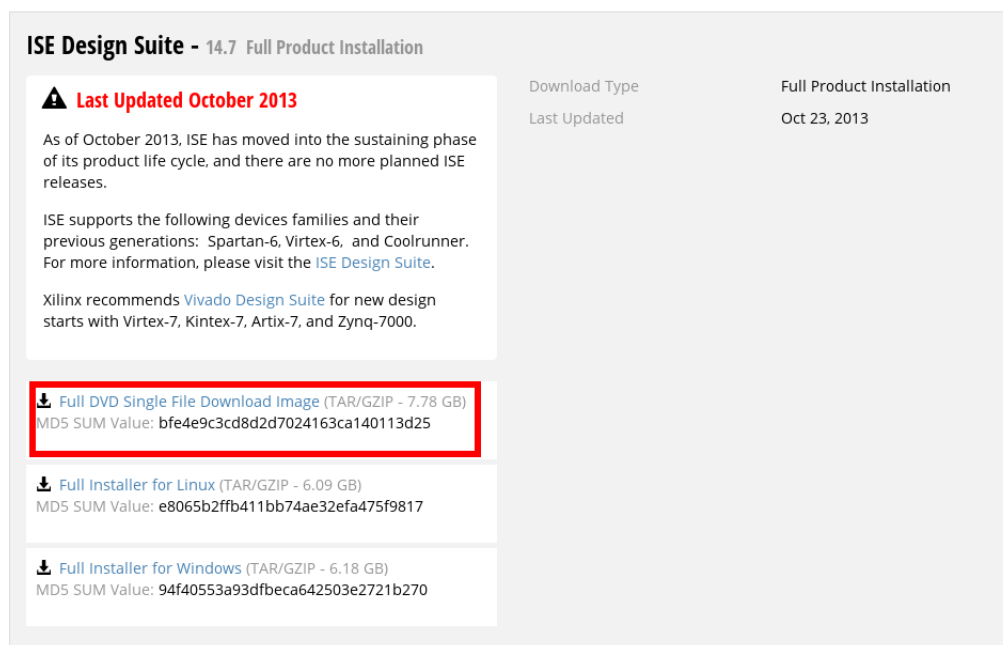


Figure 9: Xilinx ISE Download

2. If installing Xilinx tools in a permission-restricted directory, you may need to change the umask temporarily:
% `sudo su -`
% `umask 0002`
3. Extract the tarball:
% `tar -xf Xilinx_ISE_DS_14.7_1015_1.tar`
4. Enter the resulting directory and run the installer:
% `cd Xilinx_ISE_DS_14.7_1015_1`
% `./xsetup`

- Run through the installation process. Refer to the images below when applicable. Note that the checkbox for cable drivers is left unchecked. Cable driver installation, if necessary, should be handled after this installation is complete. See section 3.5 for more information.

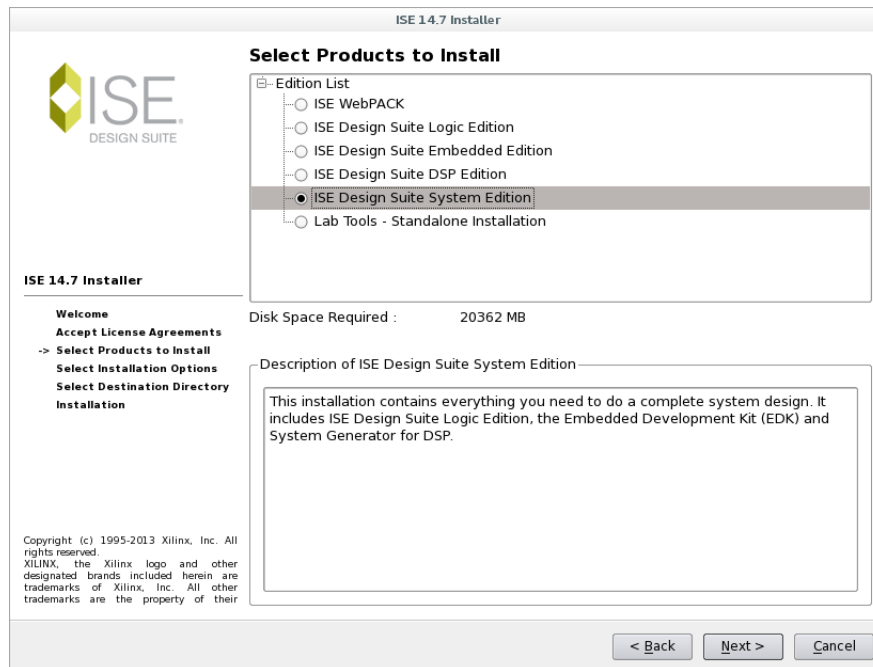


Figure 10: Xilinx ISE Installer

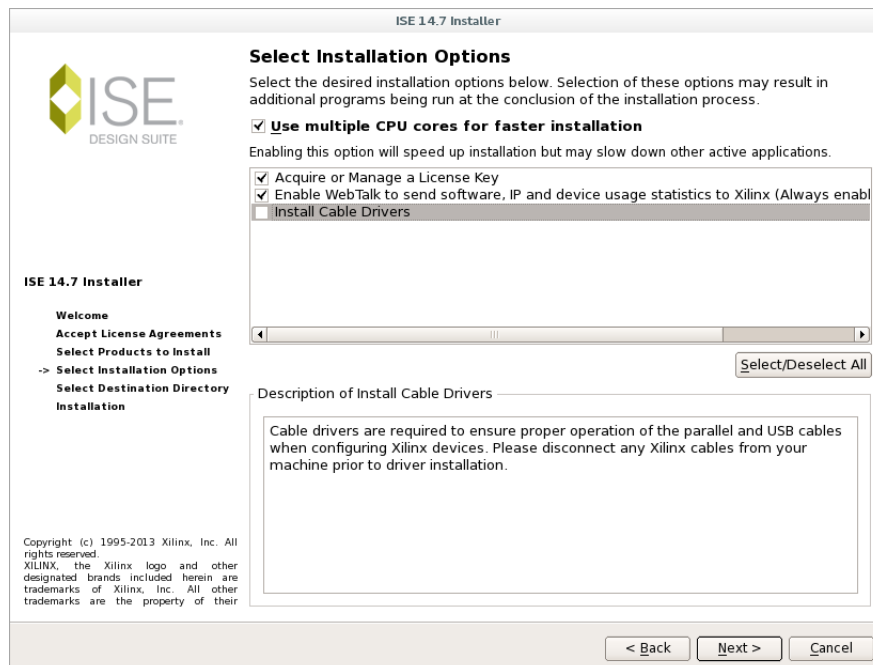


Figure 11: Xilinx ISE Installation Choice

Take note of the installation directory chosen (e.g. `/opt/Xilinx`) as well as the LabTools version (e.g. 14.7) for later use.

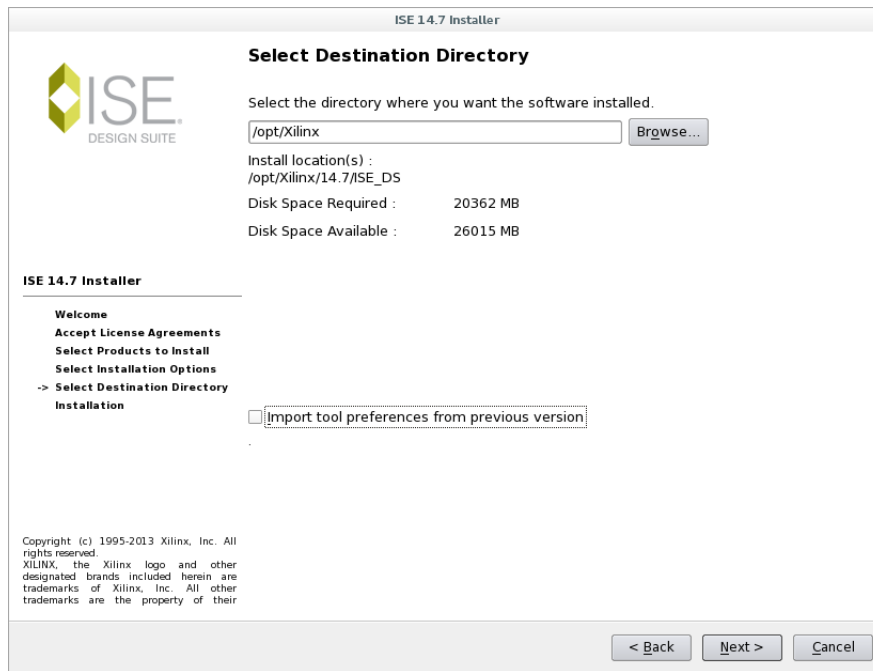


Figure 12: Xilinx ISE Install Location

3.2.1 OpenCPI Considerations

1. Note that sourcing the “`<ISE-install-dir>/<version>/LabTools/settings64.sh`” or “`<ISE-install-dir>/<version>/LabTools/settings32.sh`” scripts will interfere with OpenCPI’s environment setup. Accordingly, it is recommended to always source these scripts and execute any follow-on commands in a *separate terminal*.
2. To use OpenCPI with any Xilinx ISE or LabTools installation, it is required to set the following environment variables before running OpenCPI commands. Note that each of the following `export` statements are only necessary when the non-default installation location (i.e. anything other than `/opt/Xilinx`) or non-default version (i.e. anything other than 14.7) of the tools were used.

If only one of Xilinx ISE or Xilinx LabTools is installed,

```
% export OCPI_XILINX_DIR=<ISE-or-LabTools-install-dir>
% export OCPI_XILINX_VERSION=<ISE-or-LabTools-version>
```

If Xilinx LabTools and ISE are the same version and installed in the same directory,

```
% export OCPI_XILINX_DIR=<ISE-and-LabTools-install-dir>
% export OCPI_XILINX_VERSION=<ISE-and-LabTools-version>
```

If Xilinx LabTools and ISE are the same version and are installed in different directories,

```
% export OCPI_XILINX_DIR=<ISE-install-dir>
% export OCPI_XILINX_LAB_TOOLS_DIR=<LabTools-install-dir>
% export OCPI_XILINX_VERSION=<ISE-and-LabTools-version>
```

If Xilinx LabTools and ISE are different versions (LabTools will be ignored),

```
% export OCPI_XILINX_DIR=<ISE-install-dir>
% export OCPI_XILINX_VERSION=<ISE-version>
```

If OpenCPI has been installed prior to the ISE installation, and it is desired to make the aforementioned environment variables set automatically upon login for all users, the variables should be added in

/opt/opencv/cdk/env.d/xilinx.sh. Logging out and logging back into the user account will apply said variables.

3.3 Xilinx LabTools 14.7 Installation in CentOS 6/7

1. Download the LabTools 14.7 installation files from Xilinx's download site: <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html>. A Xilinx account will be required.

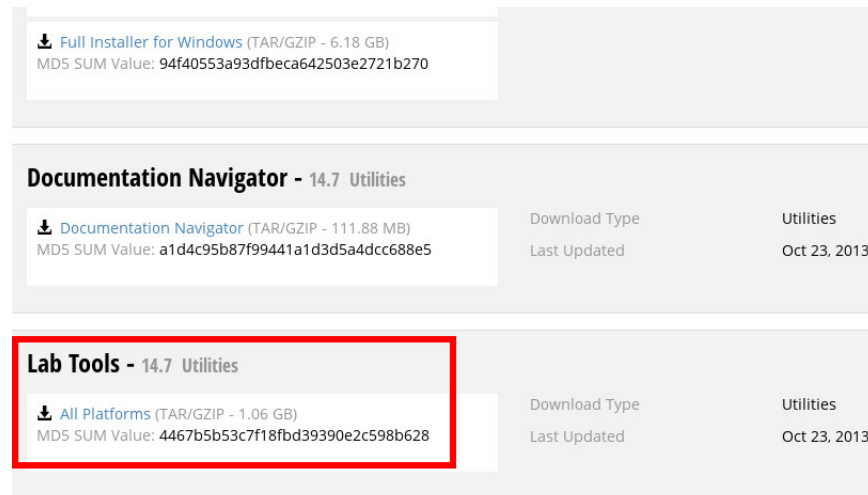


Figure 13: Xilinx LabTools Download

2. If installing Xilinx tools in a permission-restricted directory, you may need to change the umask temporarily:


```
% sudo su -
% umask 0002
```
3. Extract the tarball:


```
% tar -xf Xilinx_LabTools_14.7_1015_1.tar
```
4. Enter the resulting directory and run the installer:


```
% cd Xilinx_LabTools_14.7_1015_1
% ./xsetup
```

- Run through the installation process. Refer to the images below when applicable. Note that the checkbox for cable drivers is left unchecked. Cable driver installation, if necessary, should be handled after this installation is complete. See section 3.5 for more information.

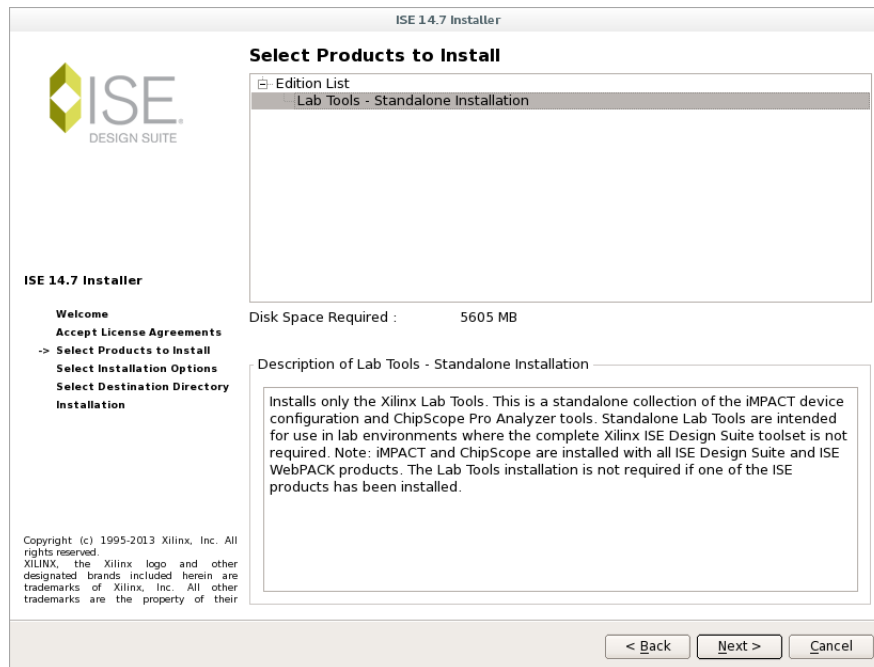


Figure 14: Xilinx LabTools Installer

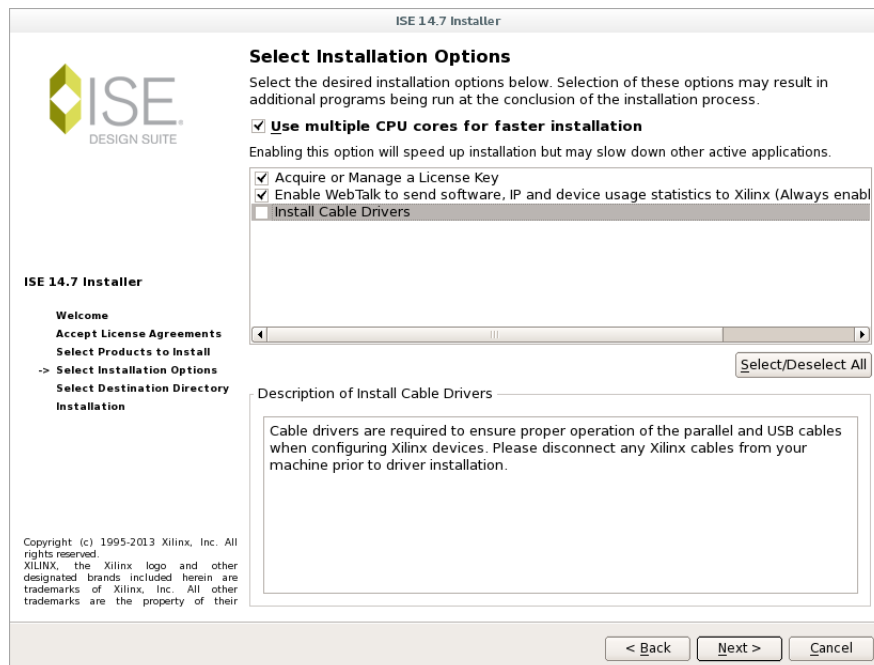


Figure 15: Xilinx LabTools Installation Choice

Take note of the installation directory chosen (e.g. `/opt/Xilinx`) as well as the LabTools version (e.g. 14.7) for later use.

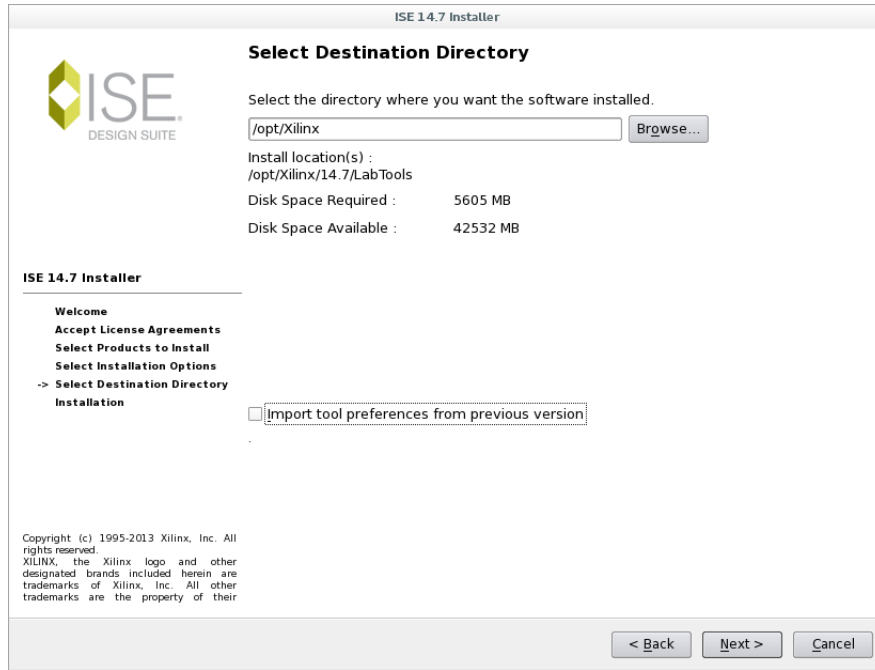


Figure 16: Xilinx LabTools Install Location

3.3.1 OpenCPI Considerations

1. Note that sourcing the “`<LabTools-install-dir>/<version>/LabTools/settings64.sh`” or “`<LabTools-install-dir>/<version>/LabTools/settings32.sh`” scripts will interfere with OpenCPI’s environment setup. Accordingly, it is recommended to always source these scripts and execute any follow-on commands in a *separate terminal*.
2. To use OpenCPI with any Xilinx ISE or LabTools installation, it is required to set the environment variables according to Section 3.2.1 before running OpenCPI commands.

3.4 Xilinx Toolset Licensing

A license, either WebPACK or non-WebPACK, is required for Xilinx Vivado and Xilinx ISE. Xilinx LabTools does not require a license.

1. The following screenshots show an ISE WebPACK license. Refer to 2 to determine which license is necessary. To generate a license, navigate to <http://www.xilinx.com/getlicense> and login (or create an account). Generate a license file:

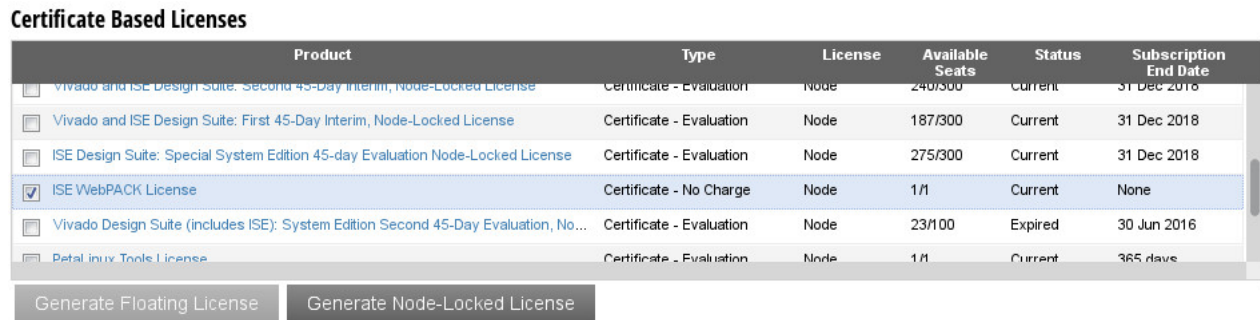


Figure 17: Generate Xilinx license file

2. Download the file and move it to the intended location:

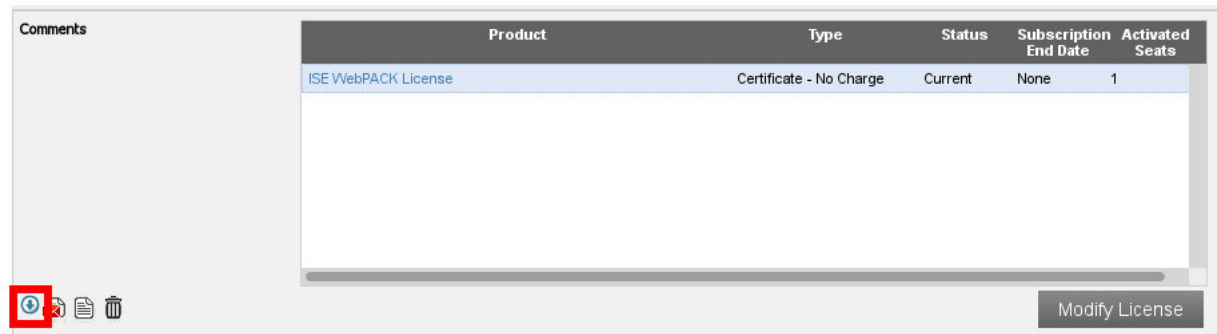


Figure 18: Download Xilinx license file

For use of Xilinx tools separate from OpenCPI, you will need to enable the license through the Xilinx tools.

For Vivado, follow these steps:

1. Run “`source <Vivado-install-dir>/Vivado/<version>/settings64.sh`”.
2. Open up the license manager and load the downloaded license. The license manager can be launched either from the Vivado GUI, or from the command line by running:
`sudo <Vivado-install-dir>/Vivado/<version>/bin/vlm`

Here, you can either navigate to “Load License” and load a copy of the license file, or you can enter the license search paths via “Manage License Search Paths”.

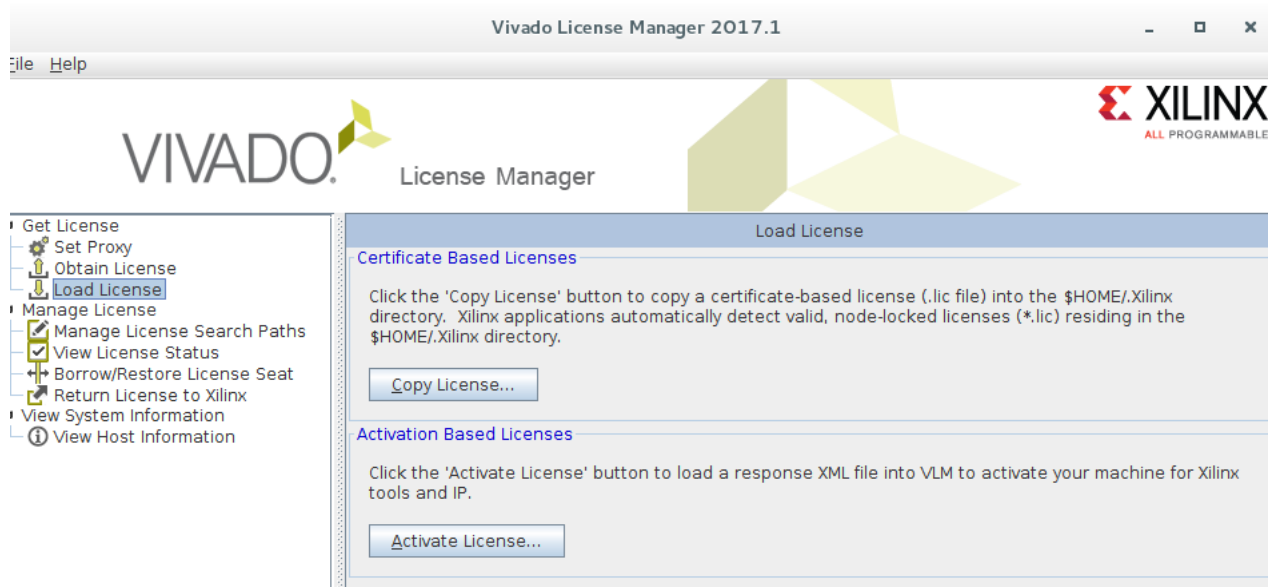


Figure 19: Load Xilinx Vivado license file

For ISE, follow these steps:

1. Run “`source <ISE-install-dir>/<version>/ISE_DS/settings64.sh`” (or `settings32.sh` if the system has a 32-bit architecture).
2. Open up the license manager and load the downloaded license. The license manager can either be launched from the ISE GUI, or launched from the command line by running:
`sudo <ISE-or-LabTools-install-dir>/<version>/ISE_DS/common/bin/lin[64]/xlcm`

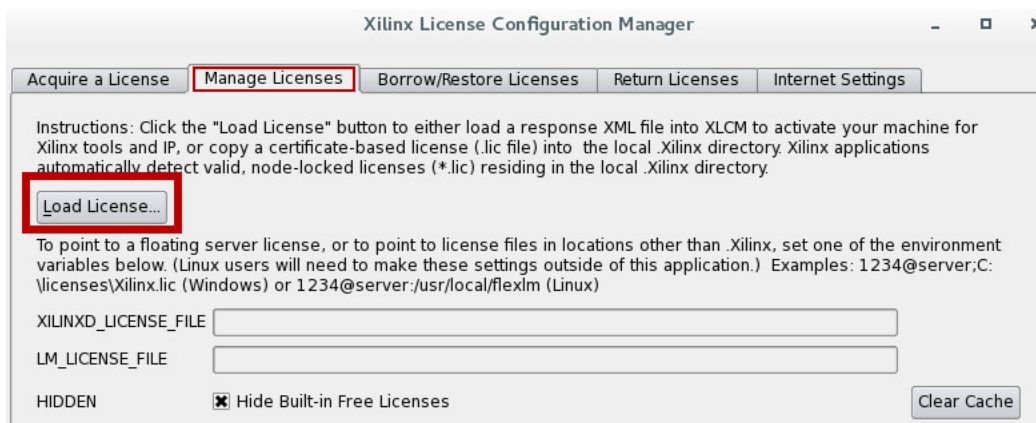


Figure 20: Load Xilinx ISE license file

3.4.1 Note on node-locked licenses in CentOS 7

If using a Xilinx node-locked license under CentOS 7, see the Red Hat Networking Guide to revert to the `ethN` naming convention.

3.4.2 OpenCPI Considerations

1. Note that sourcing the “`settings64.sh`” or “`settings32.sh`” scripts will interfere with OpenCPI’s environment setup. Accordingly, it is recommended to always source these scripts and execute any follow-on commands in a *separate terminal*.
2. To enable a license for use through OpenCPI, the following is required:

- Add `export OCPI_XILINX_LICENSE_FILE=<PATH_TO_LIC>` to `/opt/opencpi/cdk/env.d/xilinx.sh`. Note that this can instead point to a license server `<port>@<server.ip.addr>`. If using a floating license server, it is possible to set `OCPI_XILINX_LICENSE_FILE` to the license server in addition to setting `export XILINXD_LICENSE_FILE=<PATH_TO_LOCAL_LIC>`. This will allow use of a local license, e.g. a local WebPACK license, by default and the served floating license when WebPACK license is not sufficient.¹

3.5 Xilinx Cable Driver Installation in CentOS 6/7

3.5.1 Vivado

The steps herein are a slightly modified subset of those outlined in <https://www.xilinx.com/support/answers/66440.html>.

1. Run the following command : `ls -al /etc/udev/rules.d`
2. Check if the following two files are present : `52-digilent-usb.rules` `52-xilinx-pcusb.rules`
3. If the files above are not present, run the installer (*it is important to have the JTAG cable unplugged while you perform the installation*):
`<YOUR_XILINX_INSTALL>/data/xicom/cable_drivers/<lin64orlin32>/install_script/install_drivers/install_drivers`

3.5.2 ISE

Refer to 2 to determine if cable driver installation is required. Xilinx ISE does not officially support cable driver installation in CentOS. The steps herein are a slightly modified subset of those outlined in <http://www.xilinx.com/support/answers/29310.html>.

Installation of libusb prerequisites

1. Make sure `libudev-devel` is installed. If not, run `"sudo yum install libudev-devel"`.

Installation of the libusb package

1. Download the libusb 0.1.12 package from <http://libusb.info/>
 Make sure this is version 0.1.12. You will probably need to navigate to the legacy libusb:
 Downloads, Other Releases, Parent folder, libusb-0.1 (LEGACY), 0.1.12
2. Open a shell or terminal console.
3. Extract the libusb package script and its support files by typing: `tar -xf libusb-0.1.12.tar.gz`. This will create a directory named `"libusb-0.1.12"` in the current directory.
4. Navigate to the `"libusb-0.1.12"` directory by typing:
`% cd libusb-0.1.12`
5. Run the configure script by typing: `./configure`. Running the `./configure` script without any argument installs the libusb shared libraries to the `"/usr/local"`. Root permission is likely required to be able to write to this directory. If root permission is not available, run the configure script with the `--prefix` argument.
`% ./configure --prefix=<install-dir>`
 where `<install-dir>` is a directory where the libusb shared libraries will be installed, and this directory can be owned by a regular user.
6. Run the following commands to complete the installation (will likely need root permission here as well):
`% make`
`% make install`

¹See Xilinx "AR# 42507: What are the search order and locations..." and "AR# 44024: If a feature is licensed in multiple locations..."

7. Update the `LD_LIBRARY_PATH` environment variable, if necessary, to point to the libusb shared libraries. If the installation was performed from a root account, make sure that the `“/usr/local/lib”` directory is included in the `LD_LIBRARY_PATH` environment variable. If the installation was performed from a regular user’s account, add the `“<install_dir>/lib”` to the `LD_LIBRARY_PATH` environment variable. The directory may instead be `“/usr/local/lib”` or `“<install_dir>/lib”` (whichever contains the libusb 0.1 .so/.a files).

Installation of cable enumeration drivers

NOTE: The user will likely require root permissions to perform this installation.

1. Disconnect the cable and close all applications that use the cable.
2. Log in and open a shell or terminal console.
3. Navigate to the `<XILINX_DIR>/bin/lin` (for 32-bit version) or `<XILINX_DIR>/bin/lin64` (for 64-bit version) directory. Note that the `<XILINX_DIR>` variable should be set to your Xilinx installation directory (i.e. `/opt/Xilinx/14.7/ISE_DS/ISE`).

If you have installed Xilinx LabTools instead of ISE, you will need to navigate to:

`<XILINX_DIR>/bin/lin64/install_script/install_drivers/linux_drivers/pcusb` (for LabTools, `XILINX_DIR` might be `/opt/Xilinx/14.7/LabTools/LabTools`)

4. Run the installation script by typing: `./setup_pcusb` (you might first need to set the script as executable). This script will likely require root privileges. You might see some errors executing the script relating to `“xusbdfwu.hex”` file. These can be ignored.
5. Reconnect the cable.

Verifying udev rules

1. Run the following command : `ls -al /etc/udev/rules.d`
2. Check if the following file is present : `xusbdfwu.rules`
3. If the file is present, go to step 5. If the files above are not present, open the `setup_pcusb` script and change line 26 from `TP_USE_UDEV="0"` to `TP_USE_UDEV="1"`
4. Rerun the `setup_pcusb` installation script
5. `xusbdfwu.rules` should now be present in `ls -al /etc/udev/rules.d`. Open the file and change (if necessary) `SYSFS` to `ATTRS`
`BUS` to `SUBSYSTEM`
`$TEMPNODE` to `$tempnode`
6. Reload the udev rules by typing `udevadm control --reload-rules`

3.5.3 Testing Cable Driver Installation

Vivado

TODO

ISE

To verify successful cable driver installation, you can run the following:

```
export LIBUSB_INSTALL_DIR=/usr/local/lib/
cd /opt/Xilinx/14.7/ISE_DS
. ./settings64.sh
cd
```

```
echo listusb cables | LD_LIBRARY_PATH=$LIBUSB_INSTALL_DIR:$LD_LIBRARY_PATH impact -batch
```

If the cable driver is successfully installed, `“Using libusb.”` will be included in the text printed to the screen.

4 Altera Toolset Installation and Configuration

4.1 Altera Quartus Prime 15.1 Installation in CentOS 7

1. Download the Quartus Prime 15.1 installation files from Altera's download site:
<https://www.altera.com/downloads/download-center.html>. A myAltera account will be required. This first page is the download-center home page. Select the Download button next to "Quartus Prime software Standard edition" as shown in Figure 21.

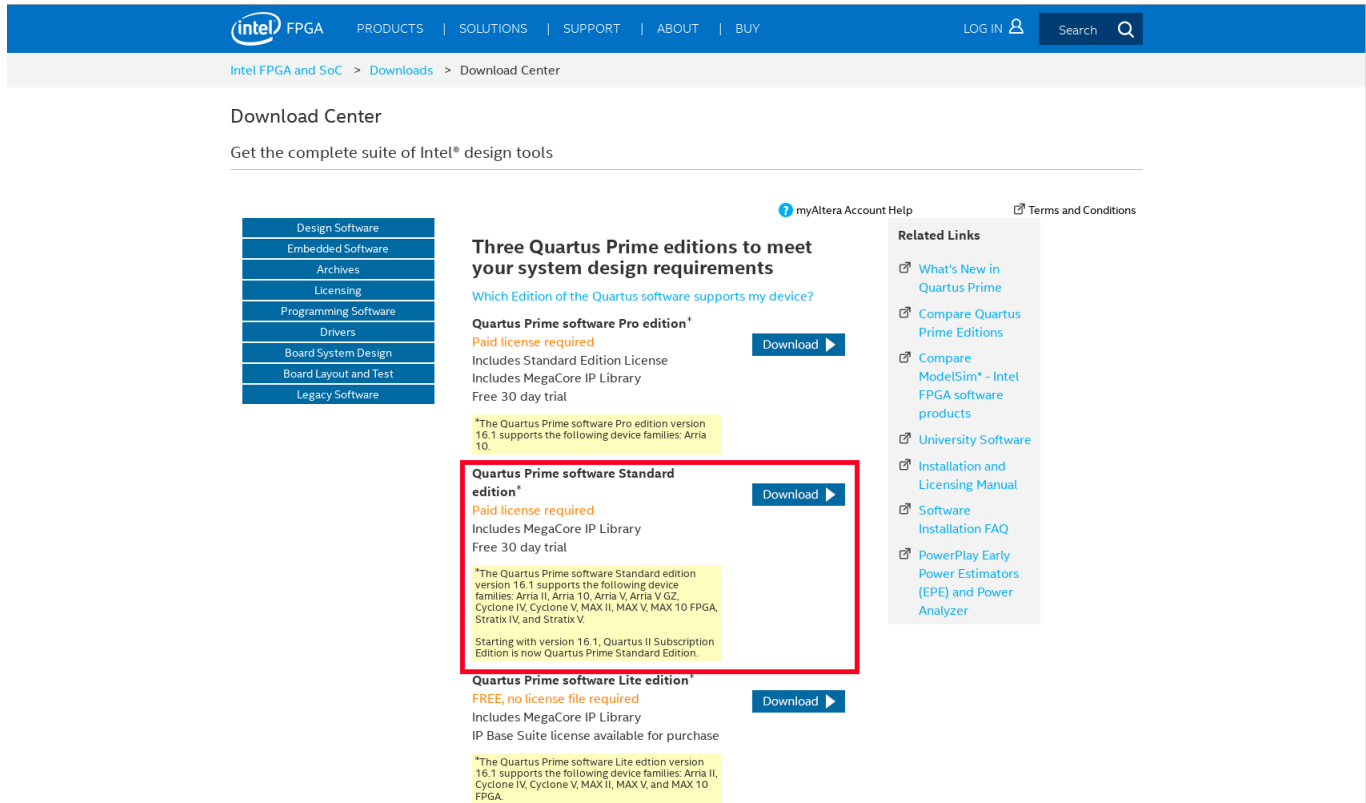


Figure 21: Altera Download Homepage

2. From the "Quartus Prime software Standard edition" page, select 15.1 from the drop-down box next to "Select release" as shown in Figure 22.
3. Select the **Complete Download** version which downloads the file `Quartus-15.1.0.185-linux-complete.tar`. Click the arrow pointing downward to begin the download as shown in Figure 22. Note that the tarball is ~24GB. Ensure that the installation system has enough disk space and correct file write permissions.

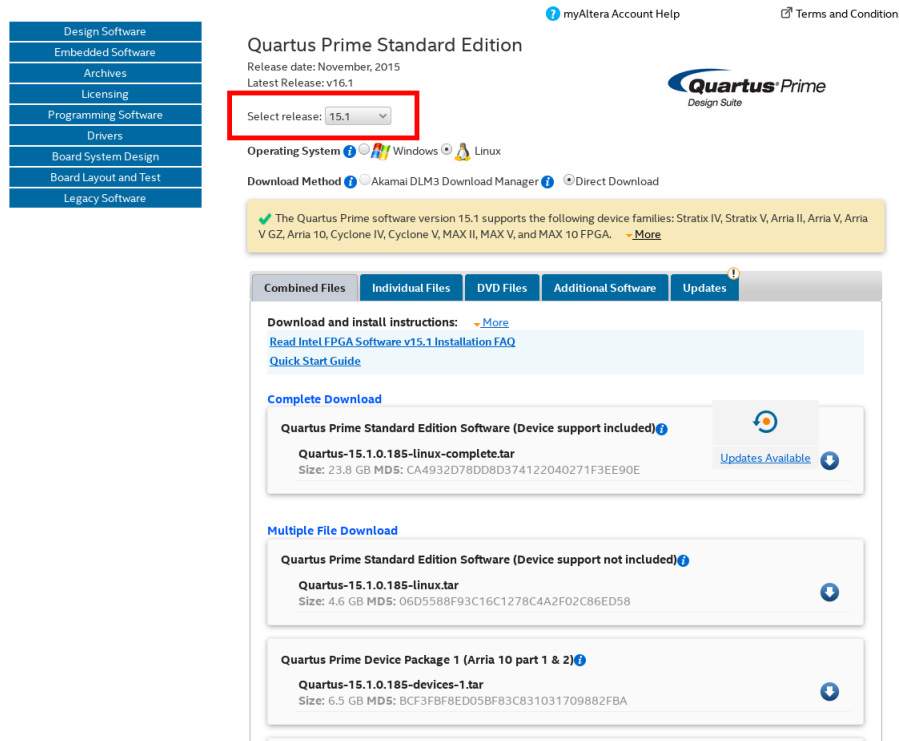


Figure 22: Altera Quartus Prime Standard Edition 15.1 Download

4. If installing Altera tools in a permission-restricted directory, you may need to change the umask temporarily:


```
% sudo su -
% umask 0002
```
5. Extract the tarball:


```
% tar xvf Quartus-15.1.0.185-linux-complete.tar
```
6. Run the installer:


```
% ./setup.sh
```

7. Run through the installation process. Refer to the images here when needed:

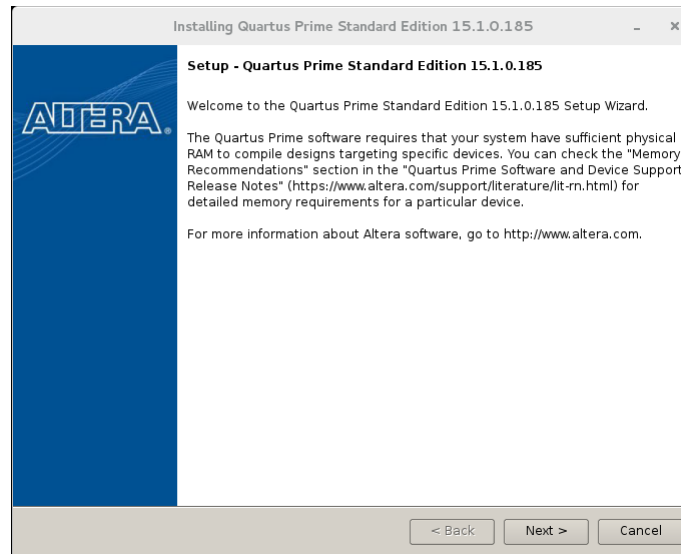


Figure 23: Altera Quartus Prime Setup Wizard

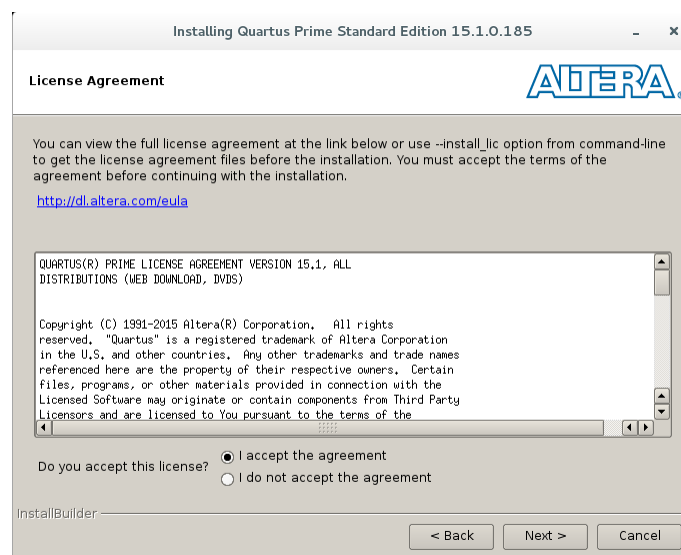


Figure 24: Altera Quartus Prime License Agreement

Take note of the installation directory chosen (e.g. `/opt/altera`) as well as the LabTools version (e.g. 15.1) for later use.

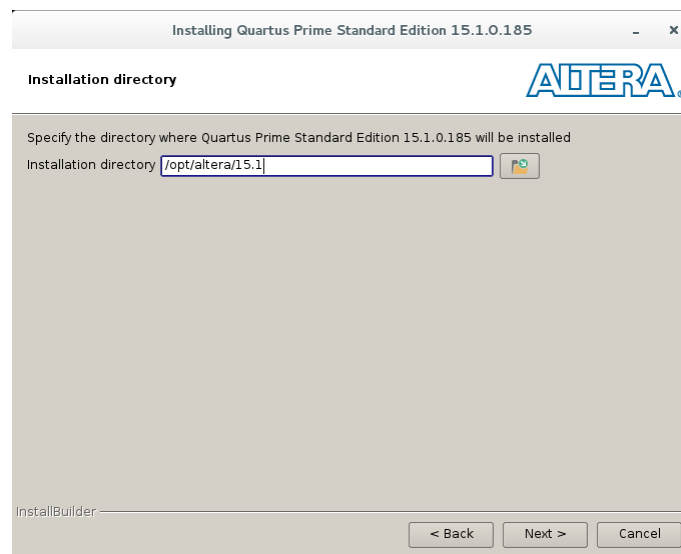


Figure 25: Altera Quartus Prime Installation Directory

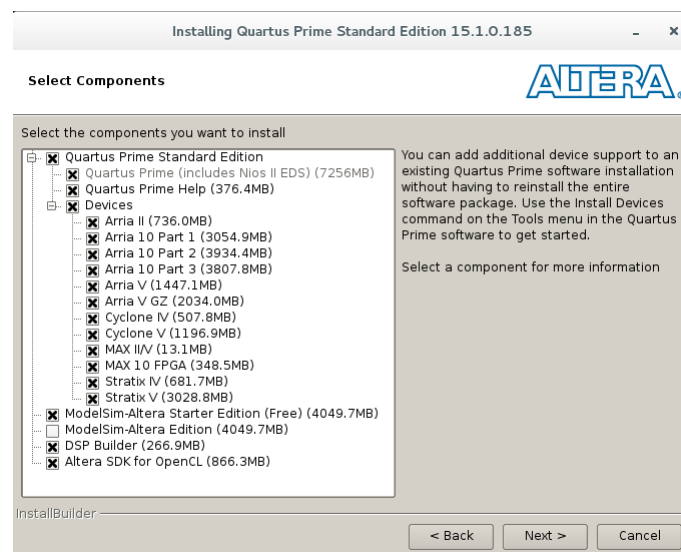


Figure 26: Altera Quartus Prime Select Components

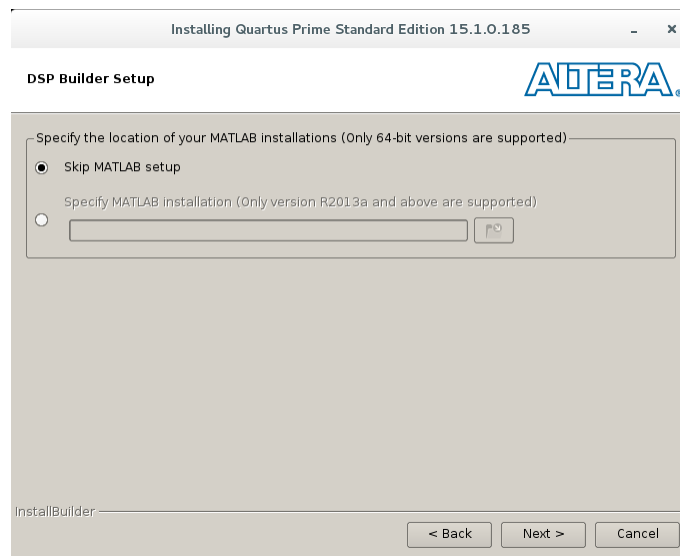


Figure 27: Altera Quartus Prime DSP Builder Setup

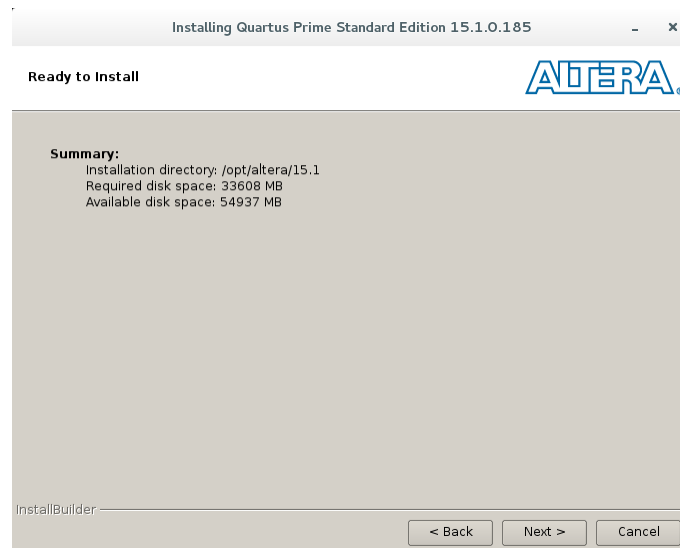


Figure 28: Altera Quartus Prime Installation Summary

4.1.1 OpenCPI Considerations

If OpenCPI has been installed prior to the Quartus installation, it is required to set the following environment variables before running OpenCPI commands. Note that `<altera-version>` should be replaced with the appropriate Altera version (e.g. 15.1), and `<altera-install-dir>` should be replaced with the installation directory (i.e. `/opt/altera`). Note also that each of the following `export` statements are only necessary when the non-default installation location (i.e. anything other than `/opt/altera` or non-default version (i.e. anything other than 14.7) of the tools were used.

```
% export OCPI_ALTERA_DIR=<altera-install-dir>
% export OCPI_ALTERA_VERSION=<altera-version>
% export OCPI_ALTERA_LICENSE_FILE=<path_to_license_file>
```

If OpenCPI has been installed prior to the Quartus installation, and it is desired to make the environment variables set automatically upon login for all users, the variables should be added in `/opt/opencpi/cdk/env.d/altera.sh`. Logging out and logging back into the user account will apply said variables.

4.2 Licensing Notes

If the user runs the Quartus software in its native GUI mode outside of OpenCPI, a license file configuration *might* be stored in the variable `LICENSE_FILE` within `~user/.altera.quartus/quartus2.ini`; this setting overrides the `OCPI_ALTERA_LICENSE_FILE` noted above and may cause confusion.