FPGA Vendor Tools Installation Guide

Version 1.4

Revision History

Revision	Description of Change	Date
v1.1	Initial Release	3/2017
v1.2	Updated for Release 1.2	8/2017
v1.4	Updated for Release 1.4	9/2018

Table of Contents

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	ANGRYVIPER Team	`http://opencpi.github.io/Overview.pdf`

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, Intel Quartus Standard Edition 17.1 and Intel Quartus Pro Edition 17.0.2. Testing has been also been done with Intel Quartus Standard Edition 15.1, but future regression testing will be performed against 17.1. Note that Quartus Standard and Quartus Pro are *different tools*. These two tools support different sets of devices and users should consult Intel's documentation for more information.

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 *and* 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
Intel Quartus Standard 17.1 with License			ALST4	x86/ALST4	ALST4	
Intel Quartus Pro Edition 17.0.2 with License					arria10soc[4]	
Mentor Graphics ModelSim DE 10.6e with License		modelsim			modelsim	

[1]With 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.
[3]Building bitstreams with a WebPACK license is limited to certain Zynq parts. Refer to the vendor's documentation for further information.
[4]There are currently no OpenCPI Board Support Packages developed for Quartus Pro, so there are no ANGRYVIPER-tested platforms to load bitstreams onto, but the arria10soc *target* can be built for.

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 core project (*e.g.*: `<core-project>/rcc/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.

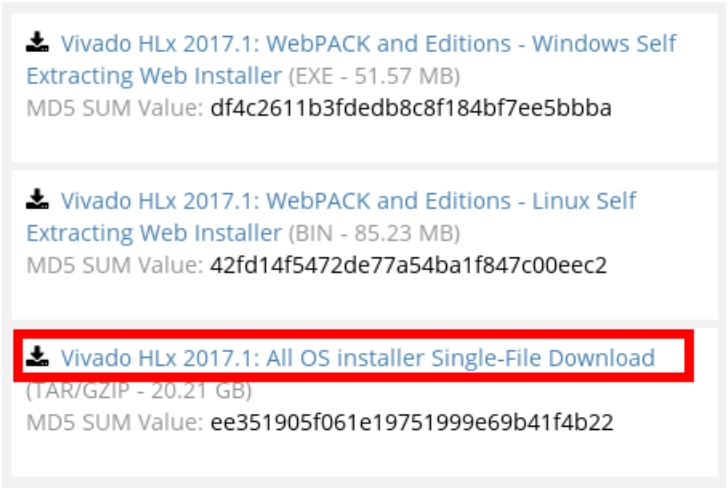

Figure 1: Xilinx Vivado 2017.1 Download

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.

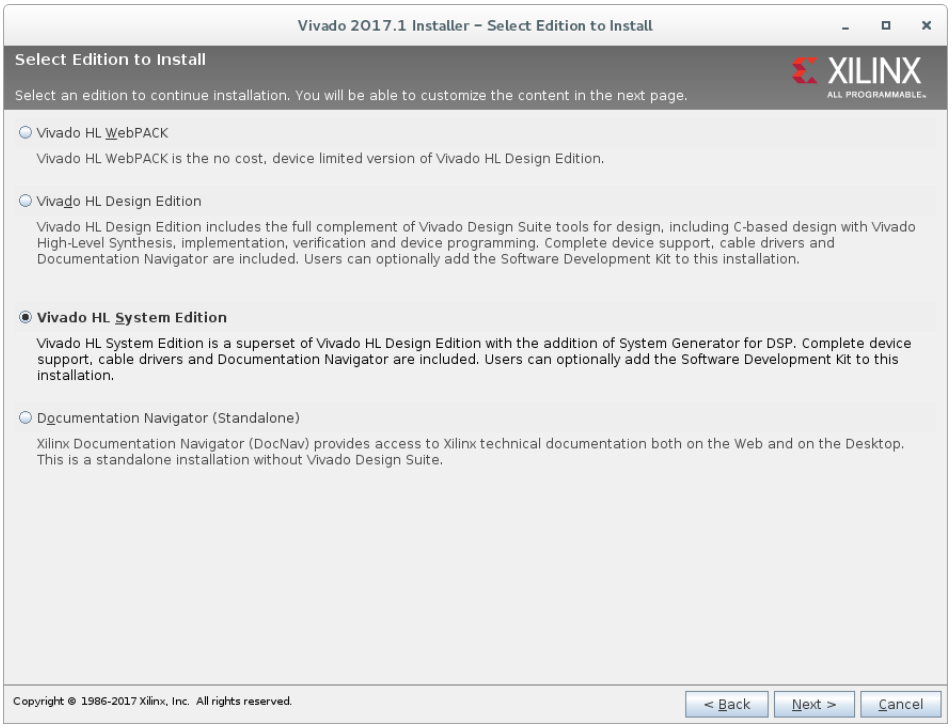

Figure 2: Xilinx Vivado Installer

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.

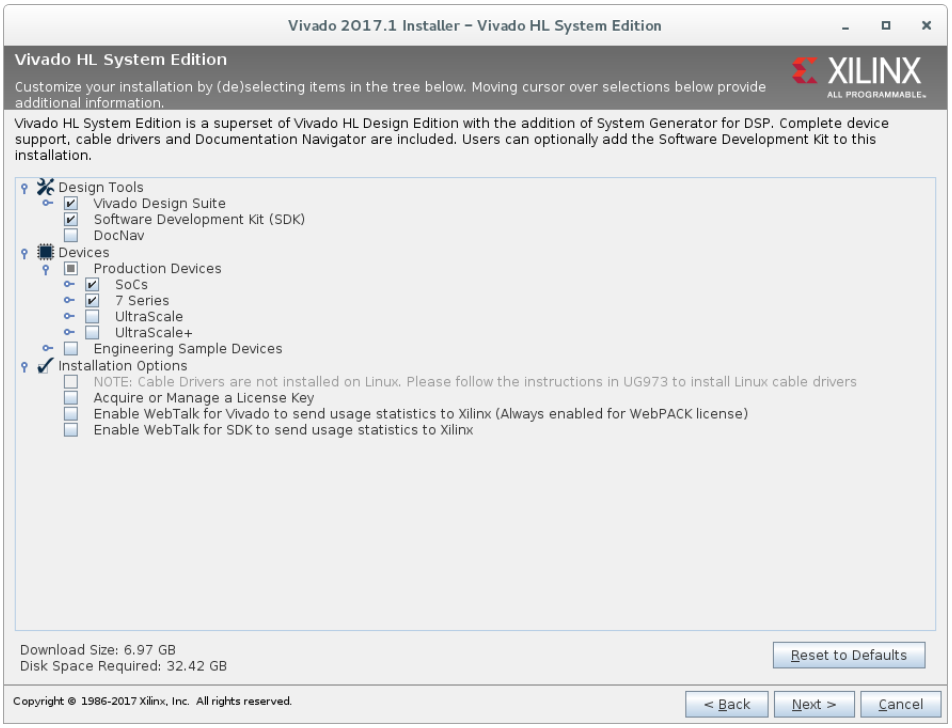

Figure 3: Xilinx Vivado Installation Choice

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

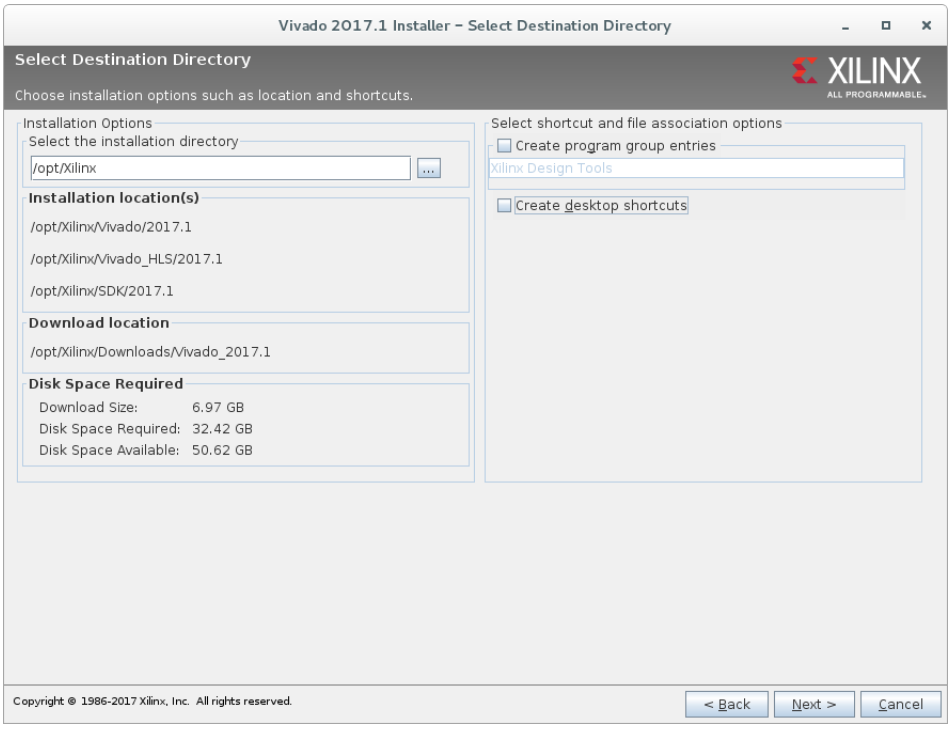

Figure 4: Xilinx Vivado Install Location

3.1.2 OpenCPI Considerations

1. 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*.

2. 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

1. 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.

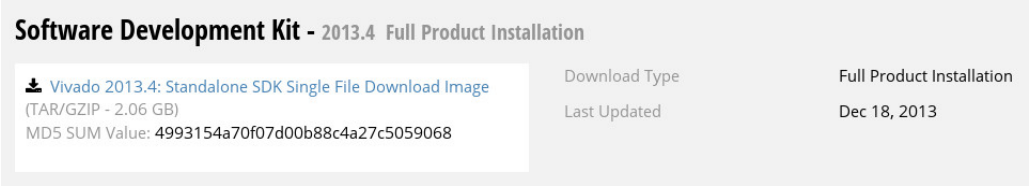

Figure 5: Xilinx Vivado 2013.4 SDK Download

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.

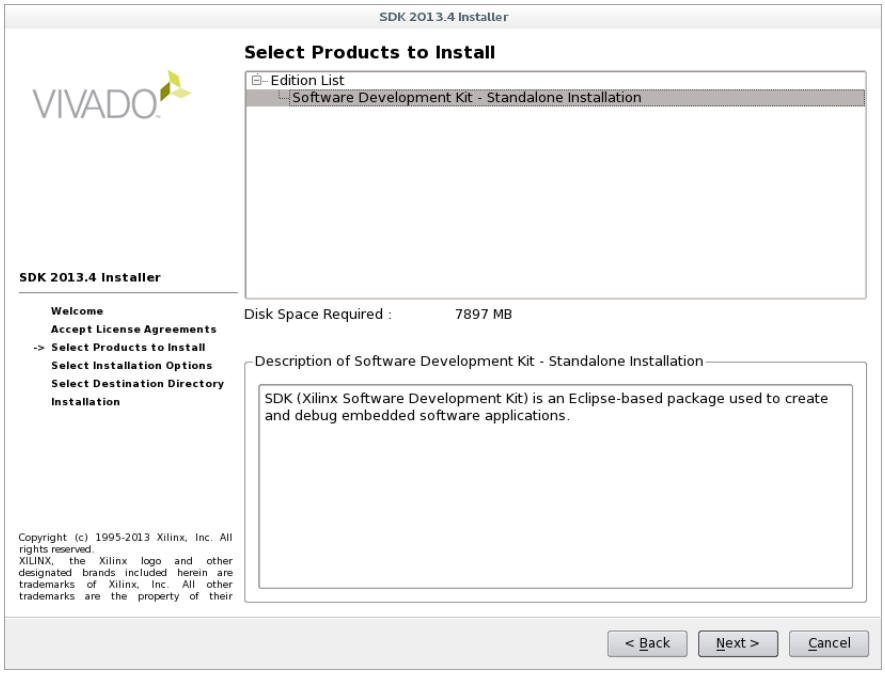

Figure 6: Xilinx Vivado SDK Installer

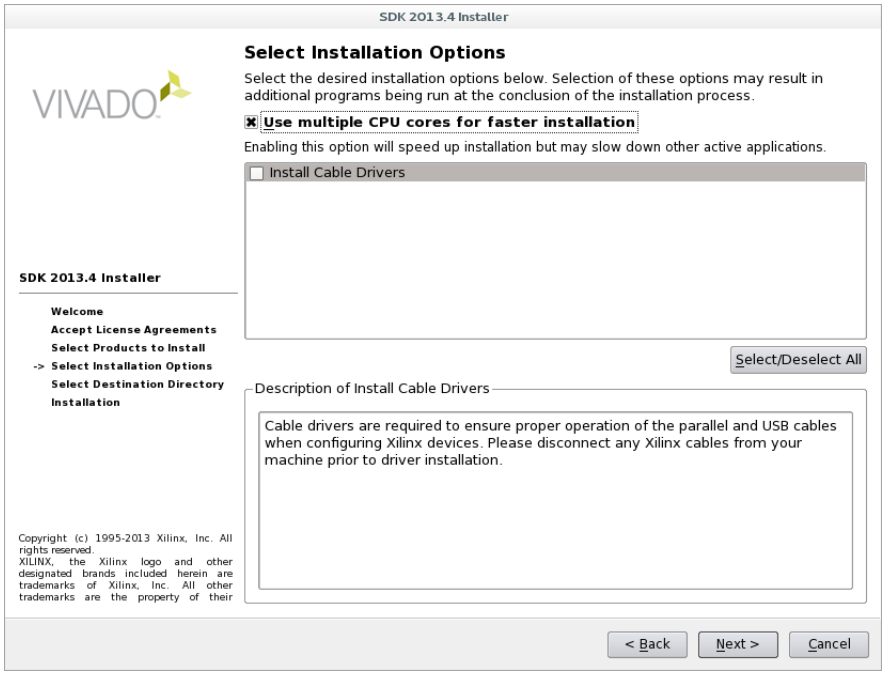

Figure 7: Xilinx Vivado SDK Installation Choice

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

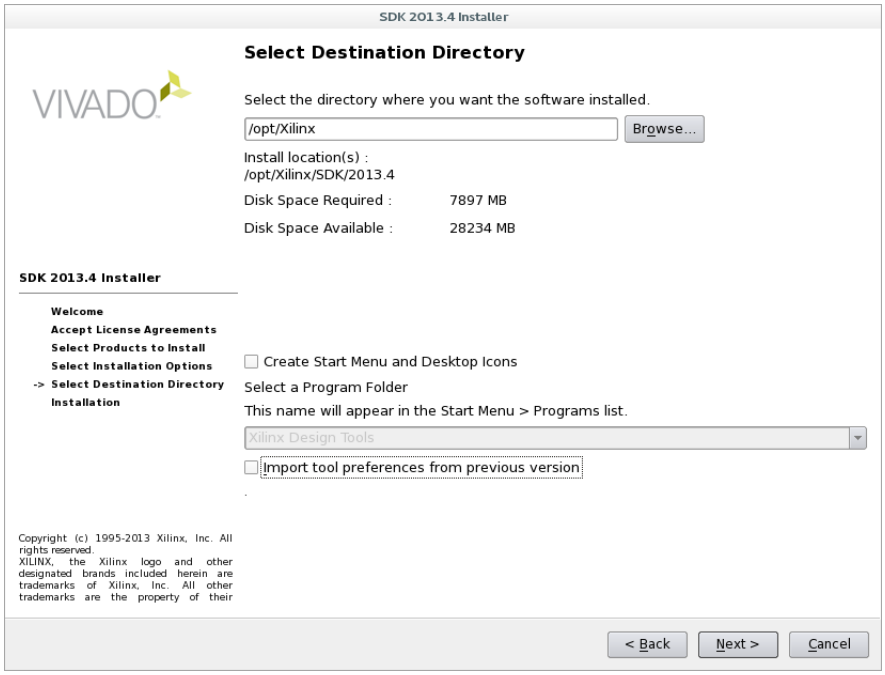

Figure 8: Xilinx Vivado SDK Install Location

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.

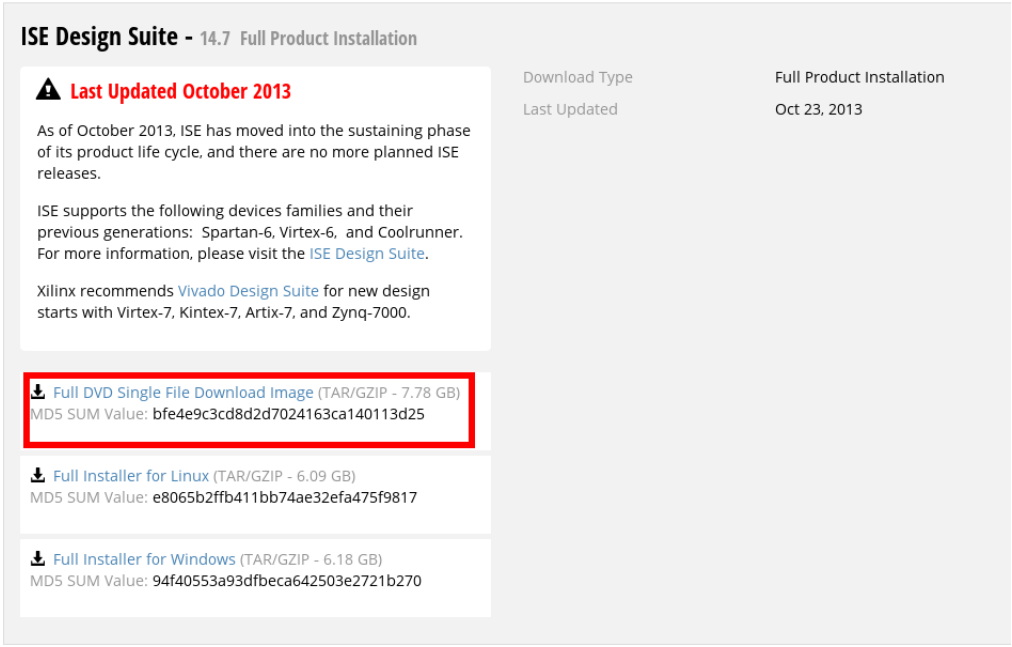

Figure 9: Xilinx ISE Download

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
   ```

5. 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.

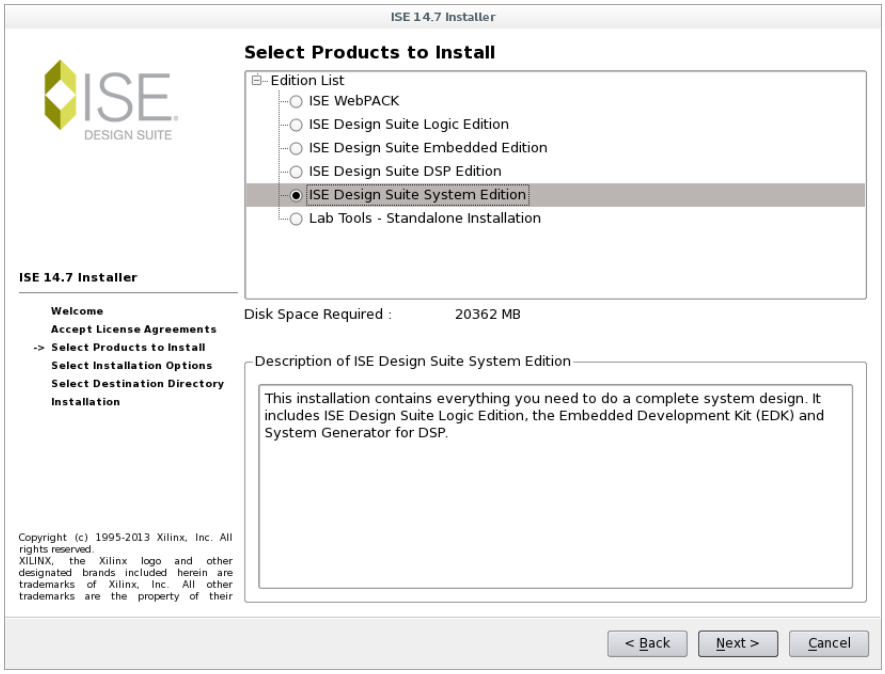

Figure 10: Xilinx ISE Installer

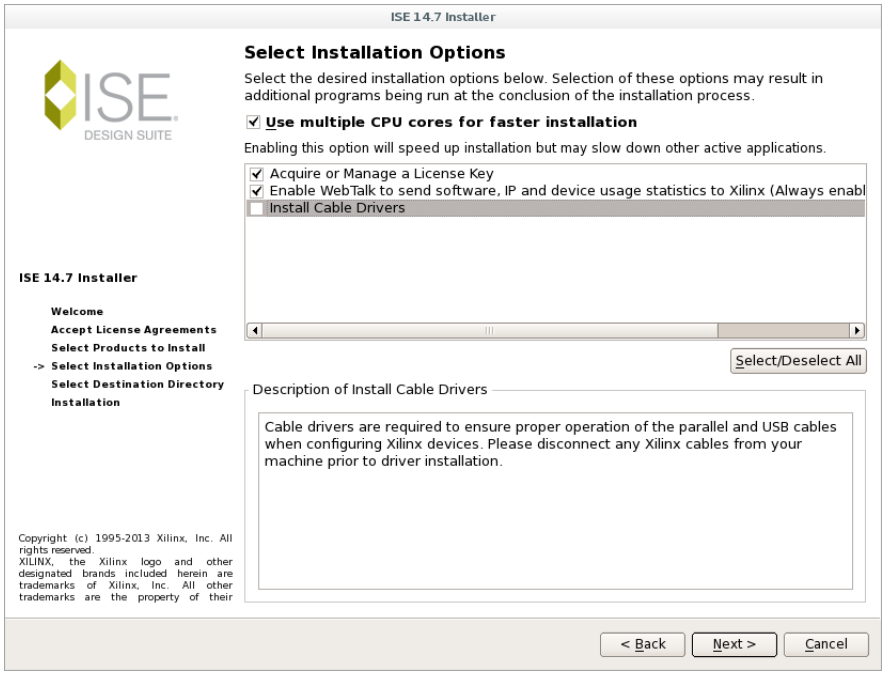

Figure 11: Xilinx ISE Installation Choice

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

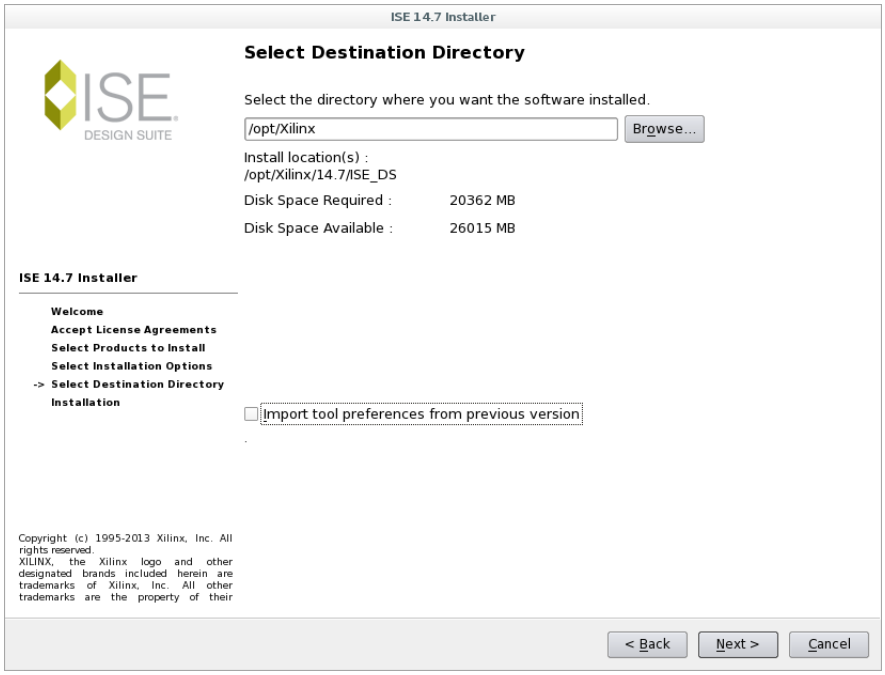

Figure 12: Xilinx ISE Install Location

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/opencpi/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.

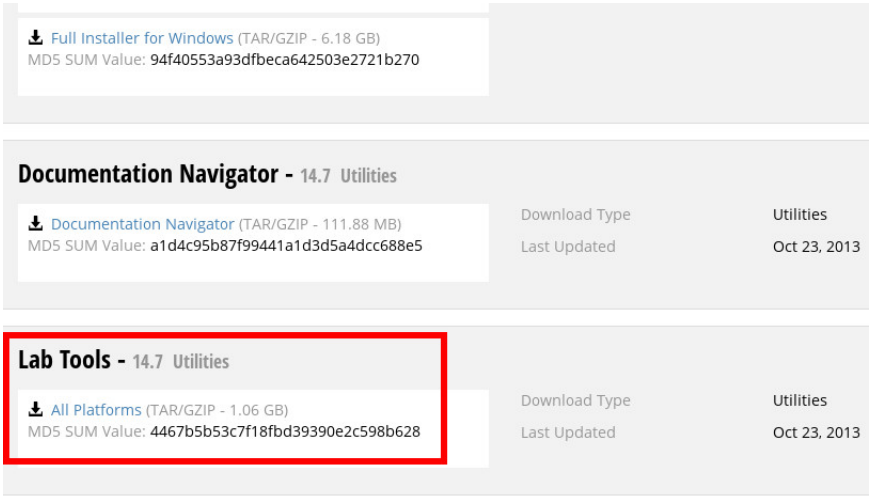

Figure 13: Xilinx LabTools Download

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
   ```

5. 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.

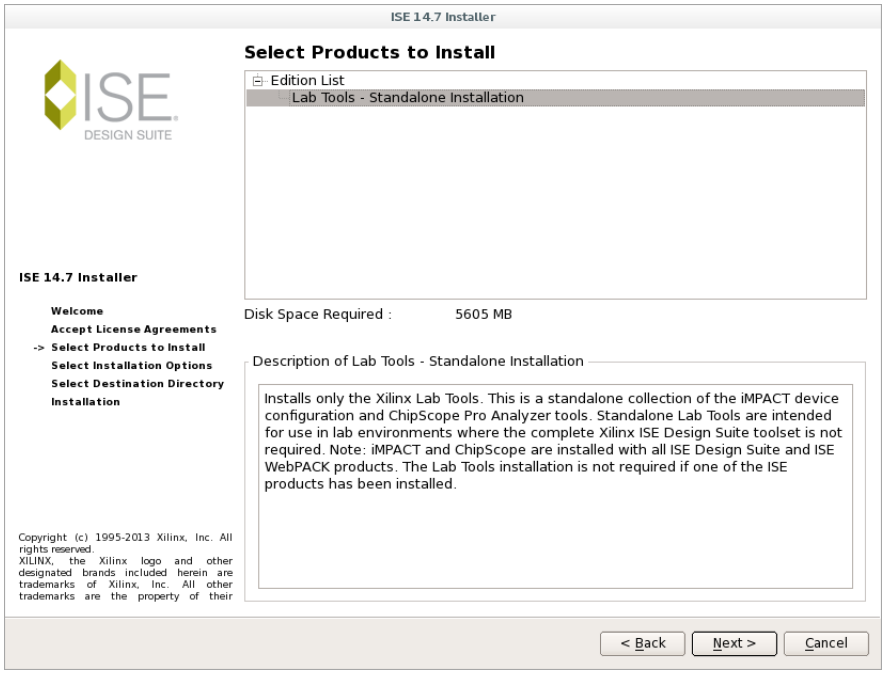

Figure 14: Xilinx LabTools Installer

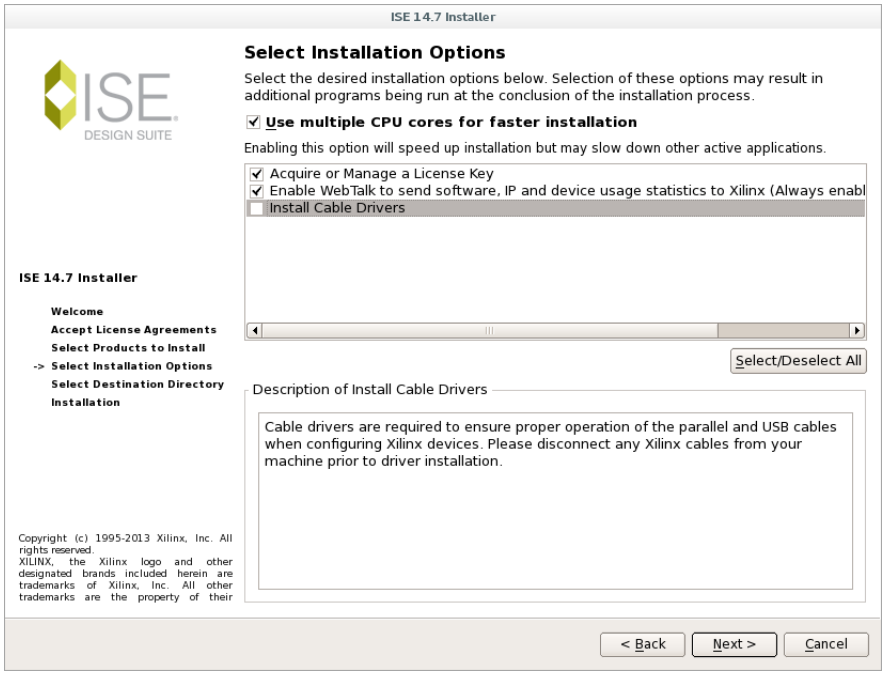

Figure 15: Xilinx LabTools Installation Choice

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

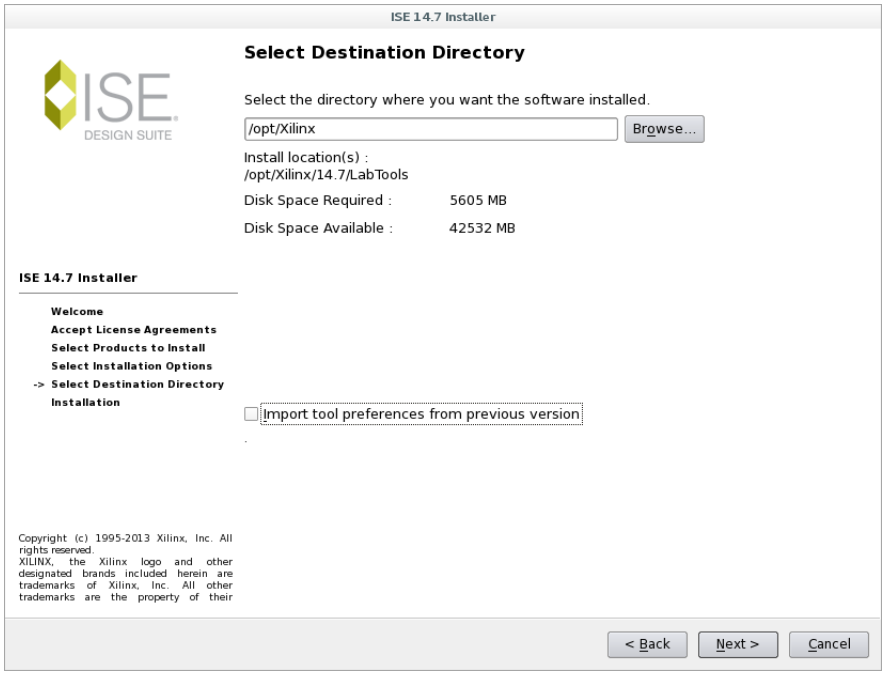

Figure 16: Xilinx LabTools Install Location

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:

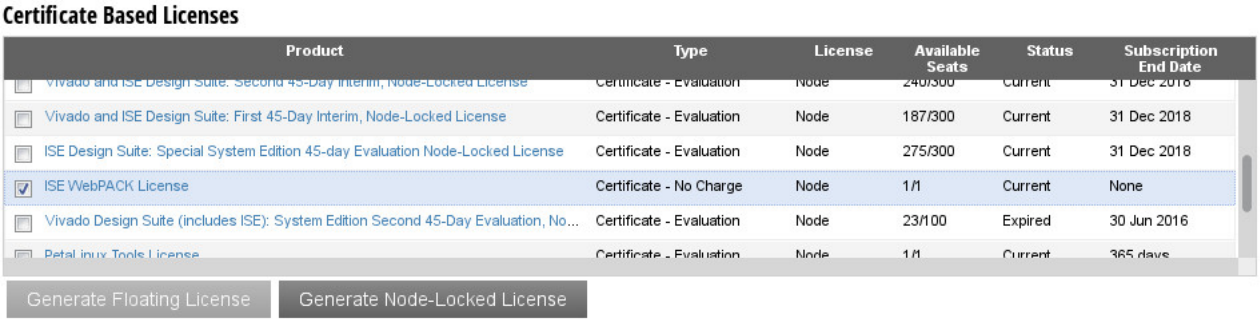

Figure 17: Generate Xilinx license file

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

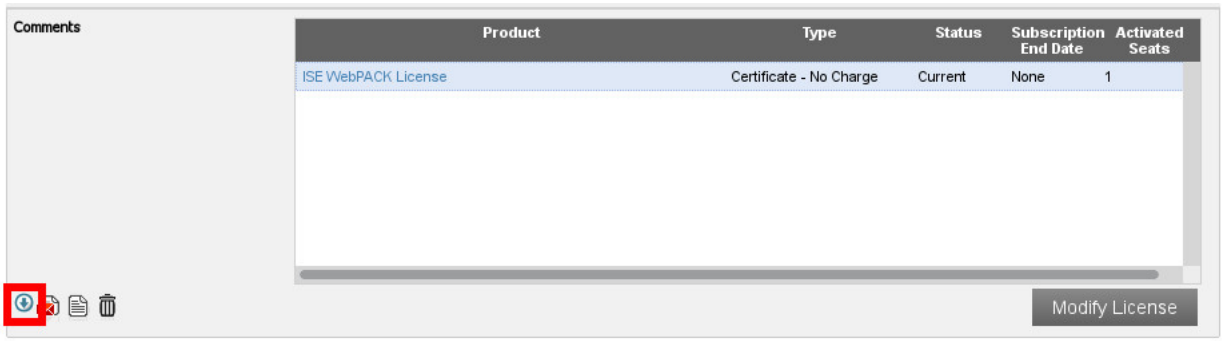

Figure 18: Download Xilinx license file

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".

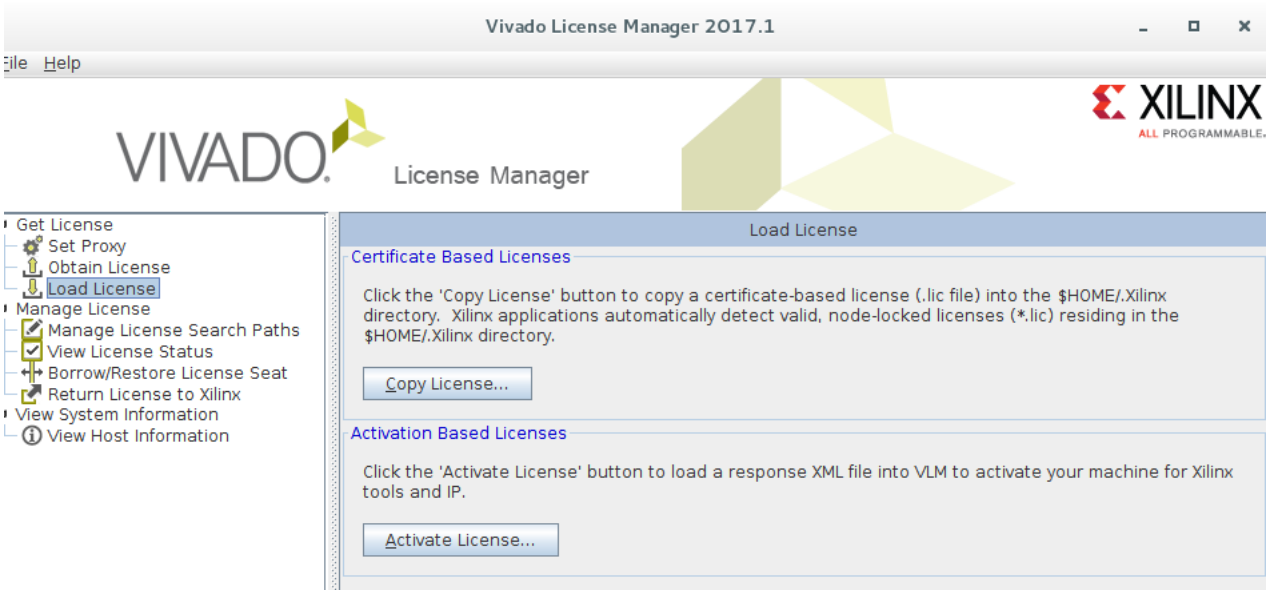

Figure 19: Load Xilinx Vivado license file

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`

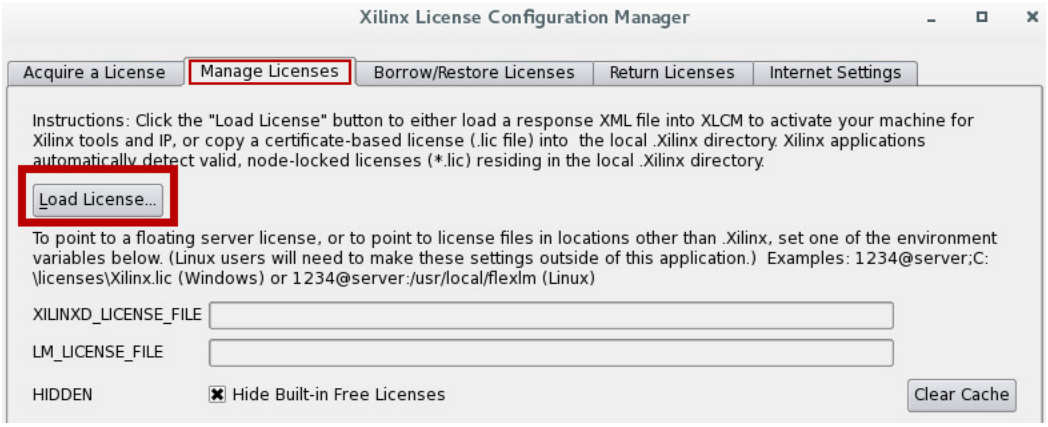

Figure 20: Load Xilinx ISE license file

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 `eth`*N* 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*):
   `cd <YOUR_XILINX_INSTALL>/data/xicom/cable_drivers/<lin64 or lin32>/install_script/install_drivers;`
   `./install_drivers;`

3.5.2 ISE

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

ISE

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

```
cd /opt/Xilinx/14.7/ISE_DS
.  ./settings64.sh
cd
echo listusbcables | impact -batch
```

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

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

4 Intel Quartus Toolset Installation and Configuration

4.1 Intel Quartus Prime Standard Edition 17.1 Installation in CentOS 7

1. Download the Quartus Prime Standard Edition 17.1 installation files from Altera's download site: `https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html`. Choose `Standard Edition 17.1` and either choose the "Complete Download", or the "Multiple File Download" (for this option, make sure to download the device packages of interest). An `Intel Customer` account will be required.
2. If installing Quartus tools in a permission-restricted directory, you may need to change the umask temporarily:
   ```
   % sudo su -
   % umask 0002
   ```
3. Extract the tarball:
   ```
   % tar xvf Quartus-17.1.0*.tar
   ```
4. Run the installer:
   ```
   % ./setup.sh
   ```
5. Run through the installation process and choose your installation directory. Note that OpenCPI will search for Quartus Standard in `/opt/altera` or `~/intelFPGA` without any additional user settings.

4.1.1 OpenCPI Considerations

It may required to set the following environment variables before running OpenCPI commands. Note that `<quartus-version>` should be replaced with the appropriate Quartus version (e.g. `17.1`), and `<quartus-install-dir>` should be replaced with the installation directory (*e.g.* `~/intelFPGA`). Note also that each of the following `export` statements are only necessary when the non-default installation location (*e.g.* anything other than `~/intelFPGA`, `/opt/intelFPGA`, `~/altera` or `/opt/Altera`), or non-default version (*e.g.* anything other than the newest version) of the tools were used.

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

These variables can be set automatically upon login for all users if added in `/opt/opencpi/cdk/env.d/altera.sh`. Logging out and logging back into the user account will apply said variables.

4.2 Intel Quartus Prime Pro Edition 17.0.2 Installation in CentOS 7

NOTE: Do not install Quartus Pro in the same directory as Quartus Standard because OpenCPI cannot differentiate between the two.
NOTE: Quartus Pro and Quartus Standard are *different tools*. The devices supported by each are different, and users should consult Intel documentation before choosing a tool edition.

1. Download the Quartus Prime Pro Edition 17.0 installation files from Altera's download site: `https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html`. Choose `Pro Edition 17.0` and either choose the "Complete Download", or the "Multiple File Download" (for this option, make sure to download the device packages of interest). An `Intel Customer` account will be required.
2. If installing Quartus tools in a permission-restricted directory, you may need to change the umask temporarily:
   ```
   % sudo su -
   % umask 0002
   ```
3. Extract the tarball:
   ```
   % tar xvf Quartus-pro-17.0.0*.tar
   ```
4. Run the installer:
   ```
   % ./setup.sh
   ```

5. Run through the installation process and choose your installation directory. Note that OpenCPI will search for Quartus Pro in `~/intelFPGA_pro` or `/opt/intelFPGA_pro` without any additional user settings.

6. Download the 17.0.2 patch by navigating to the `Updates` tab and downloading "Quartus Prime Software v17.0 Update 2".

7. Run the installer:
   `% ./QuartusProSetup-17.0.2*.run`

4.2.1 OpenCPI Considerations

It may be required to set the following environment variables before running OpenCPI commands. Note that `<quartus-version>` should be replaced with the appropriate Quartus version (e.g. `17.0` not `17.0.2`), and `<quartus-install-dir>` should be replaced with the installation directory (*e.g.* `~/intelFPGA_pro`). Note also that each of the following `export` statements are only necessary when the non-default installation location (*e.g.* anything other than `~/intelFPGA_pro`, `/opt/intelFPGA_pro`, `~/altera` or `/opt/Altera`), or non-default version (*e.g.* anything other than the newest version) of the tools were used.

```
% export OCPI_ALTERA_PRO_DIR=<quartus-install-dir>
% export OCPI_ALTERA_PRO_VERSION=<quartus-version>
% export OCPI_ALTERA_PRO_LICENSE_FILE=<path_to_license_file>
```

These variables can be set automatically upon login for all users if added in `/opt/opencpi/cdk/env.d/altera.sh`. Logging out and logging back into the user account will apply said variables.

4.3 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.

5 ModelSim Installation and Configuration

5.1 ModelSim DE 16.0e Installation in CentOS 7

1. Download the ModelSim installation files for version 10.6e.

2. If installing ModelSim tools in a permission-restricted directory, you may need to change the umask temporarily:
   ```
   % sudo su -
   % umask 0002
   ```

3. Run the installer:
   `% ./install.linux64`

4. Run through the installation process and choose your installation directory. Note that OpenCPI has no default search paths for ModelSim installations.

5.1.1 OpenCPI Considerations

Users will need to set the following environment variables to use ModelSim with OpenCPI. Note that `<modelsim-version>` should be replaced with the appropriate ModelSim version (e.g. `10.6`), and `<modelsim-install-dir>` should be replaced with the installation directory (*e.g.* `~/modelsim_dlx`). The version variable need only be set if multiple ModelSim versions exist in this directory and the user wishes to use a version *other than the most recent.*

```
% export OCPI_MODELSIM_DIR=<modelsim-install-dir>
% export OCPI_MODELSIM_VERSION=<modelsim-version>
% export OCPI_MODELSIM_LICENSE_FILE=<path_to_license_file>
```

These variables can be set automatically upon login for all users if added in `/opt/opencpi/cdk/env.d/modelsim.sh`. Logging out and logging back into the user account will apply said variables.

5.2 Compile Xilinx/Zynq simulation libraries for ModelSim

This section describes how to compile Xilinx simulation libraries of a device(s) for a particular 3rd party simulator, such as ModelSim.

1. Compile Xilinx libraries for ModelSim
2. Modify `modelsim.ini` to include path of compiled Xilinx libraries

5.2.1 Compile Vivado's simulation libraries

This section provides the steps necessary to compile Xilinx Vivado's simulation libraries of the Zynq device, for ModelSim. If using ModelSim 10.4c, note that Vivado 2017.1 does not support compilation of simulation libraries for ModelSim versions earlier than 10.5c. Therefore, if using a ModelSim 10.4c, you will need to use an earlier version of Vivado (*e.g* 2015.4) to compile the simulation libraries. For this example, we use Vivado 2017.1 with ModelSim DE 10.6e.

1. Open a terminal window and switch the user to root:

   ```
   > sudo su -
   ```

2. Configure the terminal for Xilinx Vivado by sourcing the setup script (for bash):

   ```
   > source /opt/Xilinx/Vivado/<version>/settings64.sh
   ```

3. Launch Vivado:

   ```
   > vivado
   ```

4. Select Tools → Compile Simulation Libraries...
5. Select the following:

   Simulator: ModelSim Simulator

   Language: VHDL

   Library: All

   Family: Zynq-7000

   Compiled library location: `/opt/Xilinx/Vivado/<version>/vhdl/modelsim/<version>/lin64`

   Simulator executable path: `/opt/Modelsim/modelsim_dlx/linuxpe`

   Compile 32-bit libraries: Yes

6. Click "Compile"
7. Note that 2017.1 Vivado will result in errors for ModelSim versions earlier than 10.5c. Here, we show the results for Vivado 2017.1 with ModelSim DE 10.6e, and Vivado 2015.4 with ModelSim DE 10.4c.

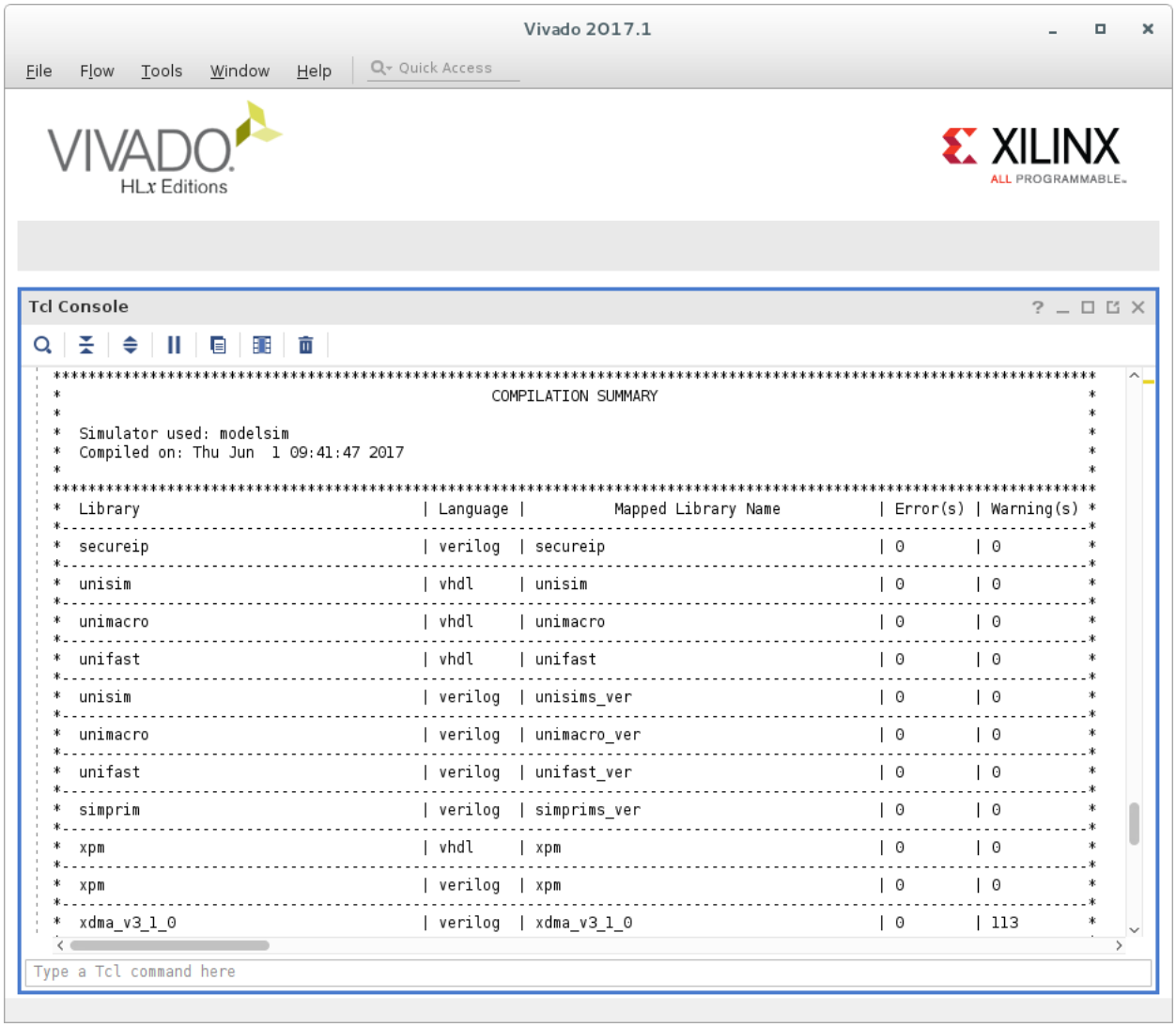

Figure 21: Vivado 2017.1 Compilation Output with ModelSim DE 10.6e

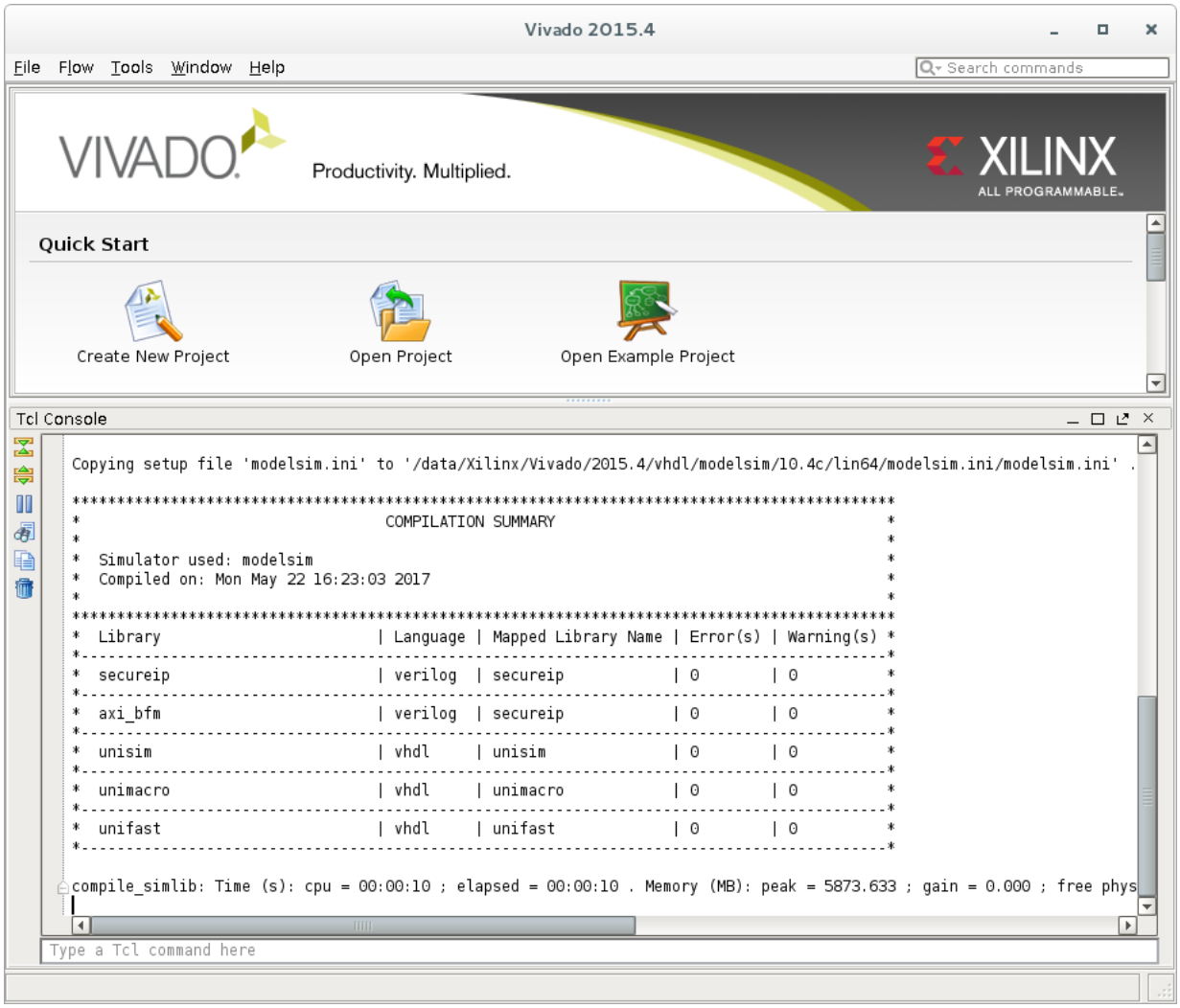

Figure 22: Vivado 2015.4 Compilation Output with ModelSim DE 10.4c

5.2.2 Compile ISE's simulation libraries

This section provides the steps necessary to compile Xilinx ISE's simulation libraries of the Zynq device, for ModelSim.

1. Open a terminal window and switch the user to root:

   ```
   > sudo su -
   ```

2. Configure the terminal window for Xilinx ISE by sourcing the setup script (for bash):

   ```
   > cd /opt/Xilinx/14.7/ISE_DS/
   > source settings64.sh
   ```

3. Launch the Xilinx CompXLib GUI:

   ```
   > cd /opt/Xilinx/14.7/ISE_DS/ISE/bin/lin64
   > ./compxlib
   ```

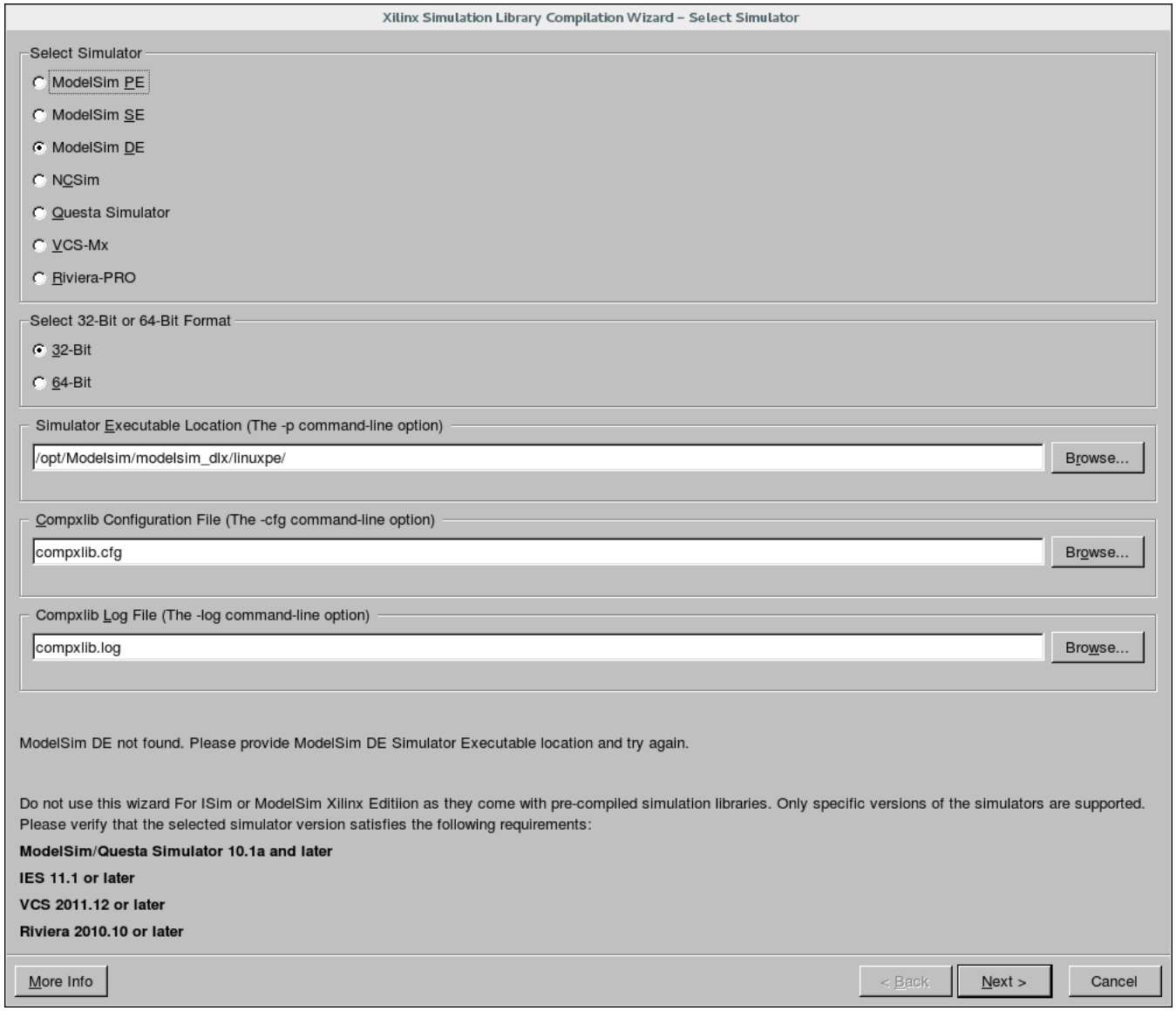

Figure 23: Compilation Wizard - Select Simulator

4. Select ModelSim DE.
5. Set Simulator Executable Location.
6. Click “Next”.

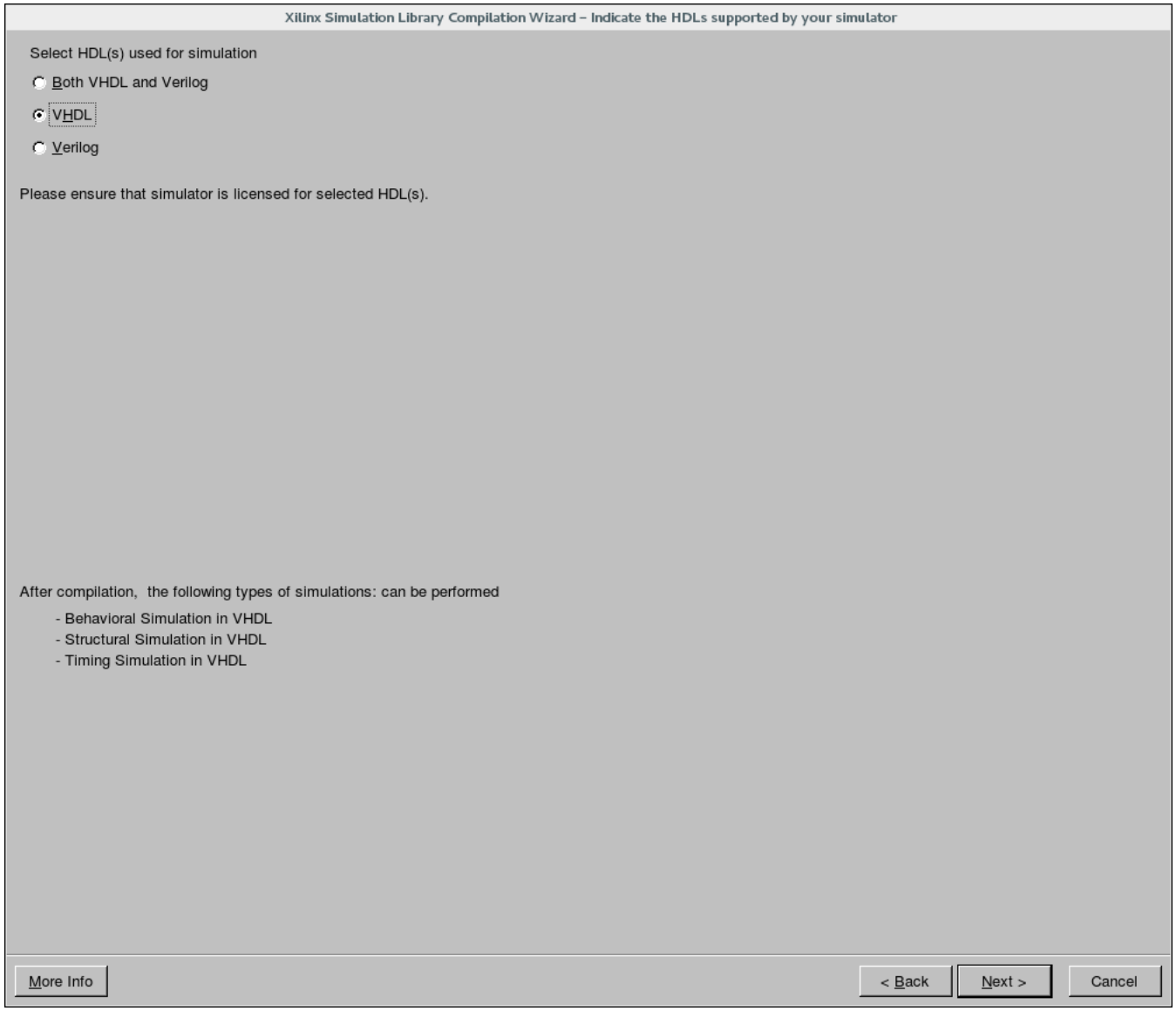

Figure 24: Compilation Wizard - HDLs to support simulator

7. Select “VHDL”.
8. Click “Next”.

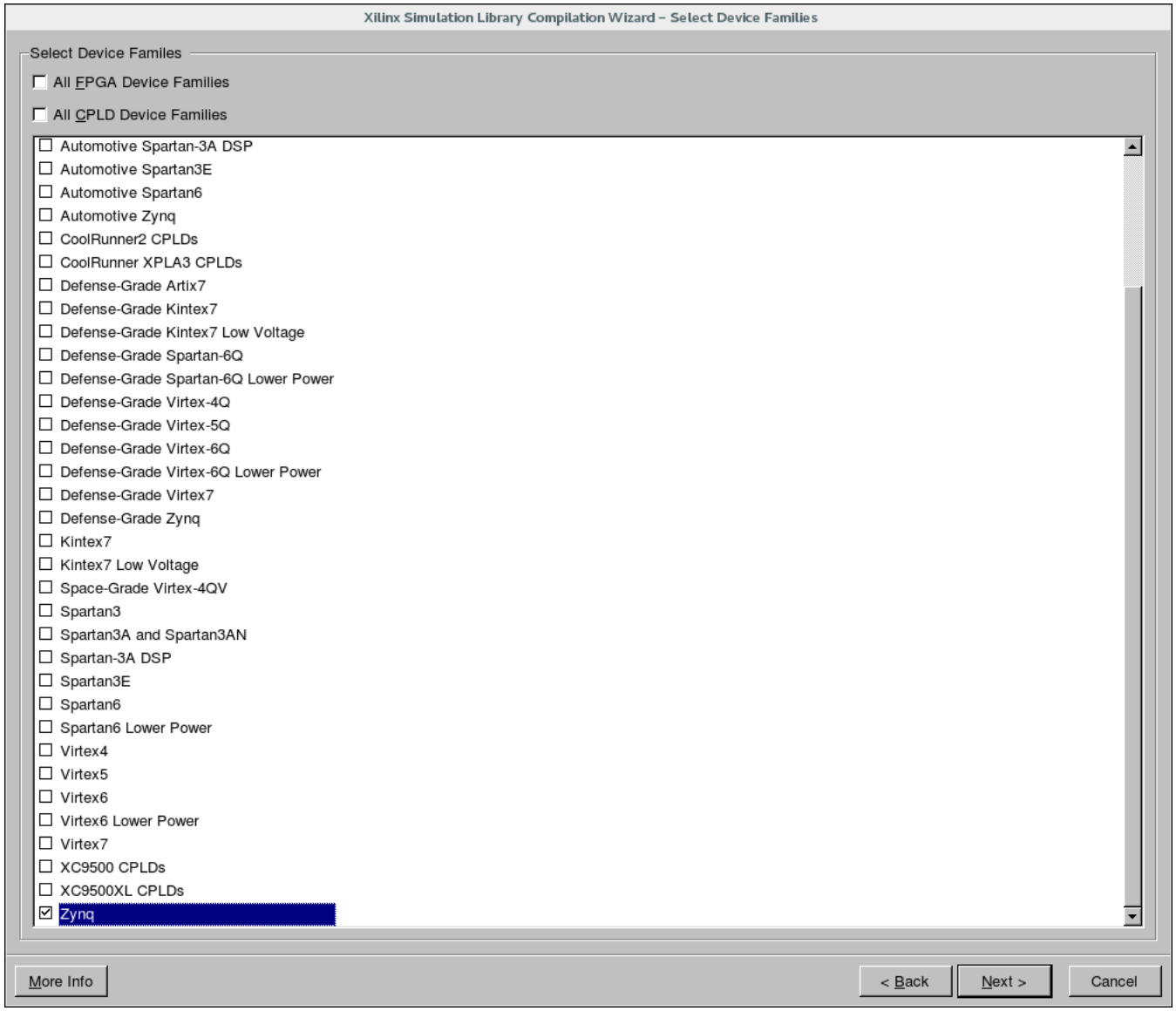

Figure 25: Compilation Wizard - Select Device Families

9. Uncheck “All FPGA Device Families”.
10. Uncheck “All CPLD Device Families”.
11. Check “Zynq”.
12. Click “Next”.

Figure 26: Compilation Wizard - Select Simulation Libraries

13. No change.
14. Click “Next”.

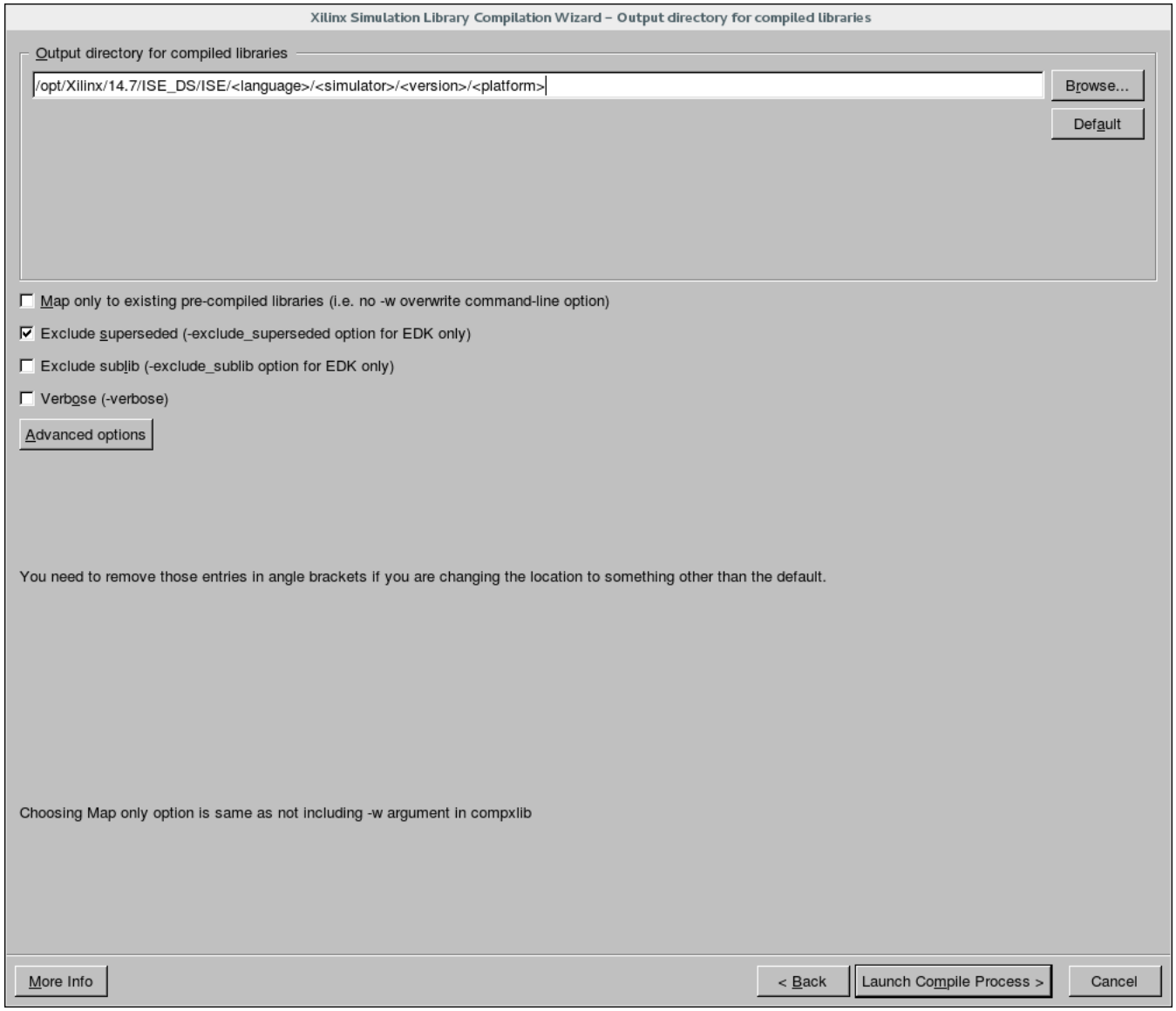

Figure 27: Compilation Wizard - Select Output directory

15. Select defaults.
16. Click “Launch Compile Process”.

    Note: This step will take approximately 20 mins.

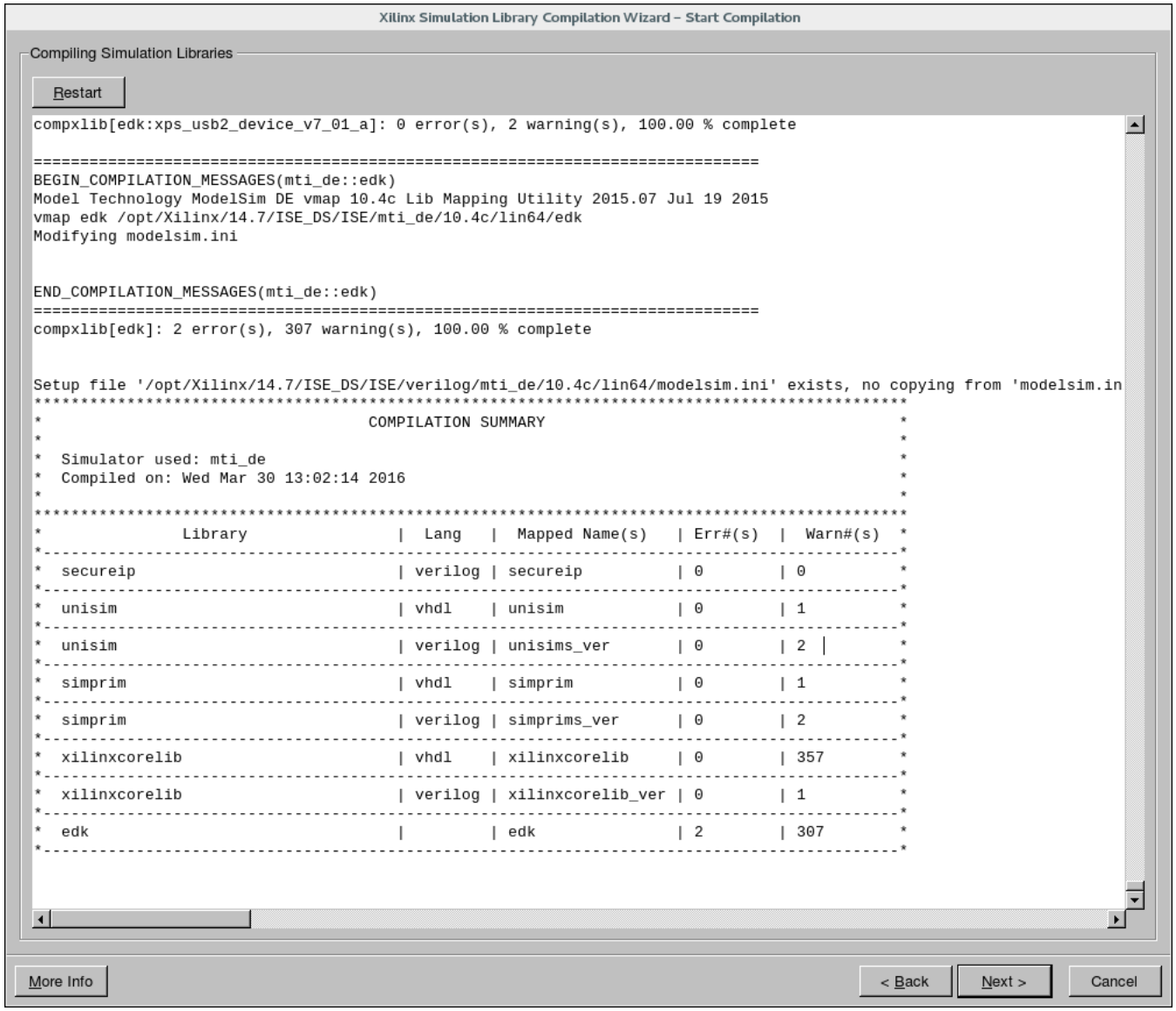

Figure 28: Compilation Wizard - Start Compilation

17. Click “Next”.

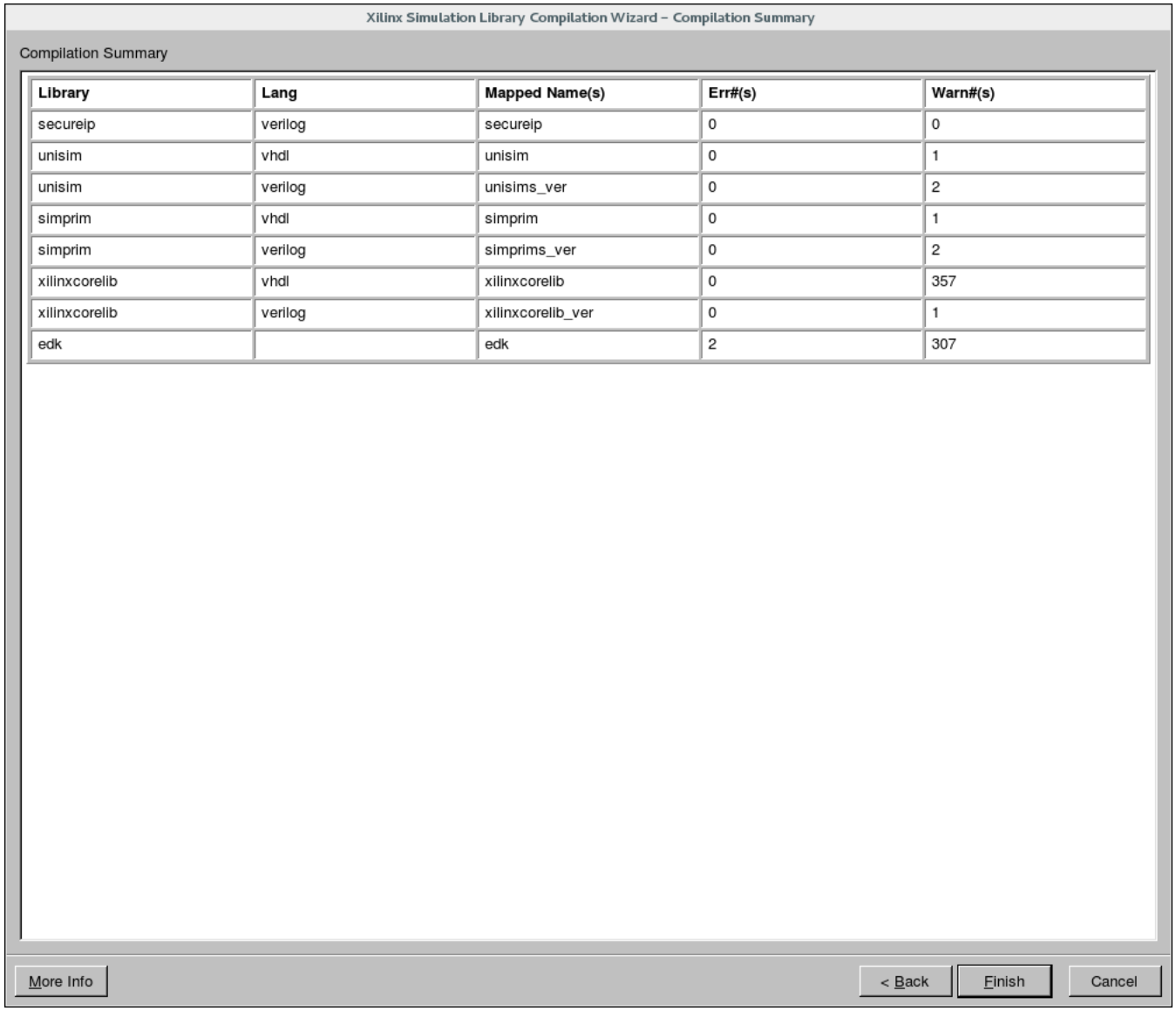

Figure 29: Compilation Wizard - Compilation Summary

18. Click “Finish”.

5.2.3 Modify "modelsim.ini" to include path to built library

This section details the steps to modify the "`modelsim.ini`" file.

1. Browse to the install directory of ModelSim
   ```
   > cd /opt/Modelsim/modelsim_dlx
   ```
2. Open the modelsim.ini file as the root user
   ```
   > vi modelsim.ini
   ```
3. Locate the bottom of the "`[Library]`" section and add the following for Vivado:
   unifast = /opt/Xilinx/Vivado/2017.1/vhdl/modelsim/10.6e/lin64/unifast
   unisim = /opt/Xilinx/Vivado/2017.1/vhdl/modelsim/10.6e/lin64/unisim
4. Or, add the following for ISE:
   xilinxcorelib = /opt/Xilinx/14.7/ISE_DS/ISE/vhdl/mti_de/10.4c/lin64/xilinxcorelib
   unisim = /opt/Xilinx/14.7/ISE_DS/ISE/vhdl/mti_de/10.4c/lin64/unisim