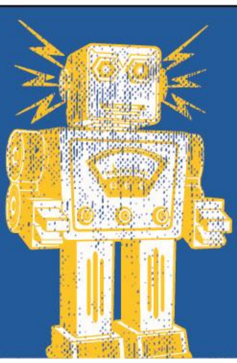
Elektronik



Digital System Design

Distance Learning Letter
Introduction to ModelSim-Intel FPGA Starter Edition

Gefördert von
MA23
Wirtschaft, Arbeit Statistik
StaDt Wien

iPLAT – Competence team for Innovative Platforms for Mixed-Hardware/Software Systems
MA23 Project 18-06

iPLAT

Version: 0.8

Author: A. Gröblinger, R. Höller, A. Puhm, C. Reisner

Copyright Notice

Introduction

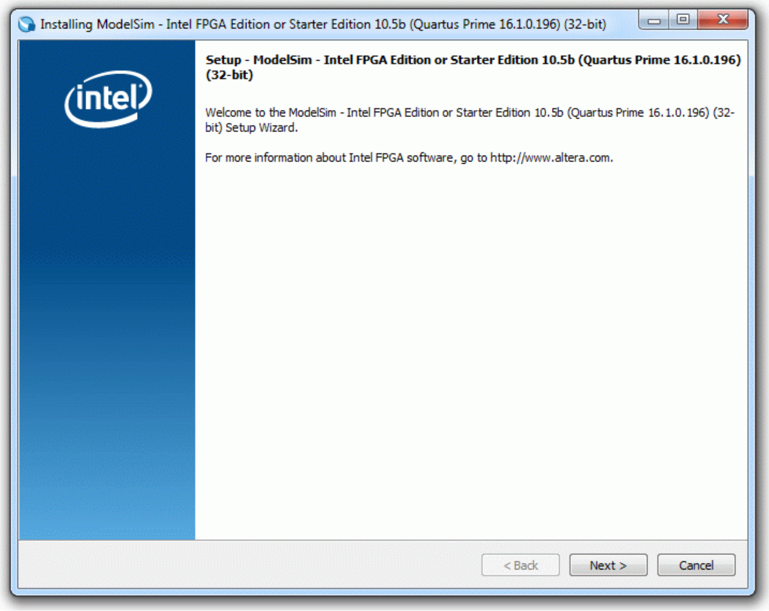
This Distance Learning Letter will describe the steps that are necessary in order to install the VHDL simulator ModelSim-Intel FPGA Starter Edition, Release 10.5b. During the course of this lecture, ModelSim will be the tool used for simulation. This document also gives a short introduction to the simulator. Furthermore, the most common commands as well as their usage during simulation will be discussed. This introduction finishes with a short example.

Note: ModelSim has many different versions. The fully-fledged fee-based version is called QuestaSim. QuestaSim is installed on all PCs of the Department of Embedded Systems. Operation and user interface of ModelSim and QuestaSim are identical. All that is said about ModelSim in this document is also valid for QuestaSim.

Installation of ModelSim-Intel FPGA Starter Edition

The installation procedure is described for a Windows 7 64-bit host machine. The process is similar on a Linux machine. Follow these steps for the installation procedure:

1. Download the installation executable from the CIS website of this lecture (see upper right corner of webpage). Extract the ZIP file (ask your lecturer for the password) and execute the EXE-file. Click “Next >” on the screen shown in Figure 1.



Installing ModelSim - Intel FPGA Edition or Starter Edition 10.5b (Quartus Prime 16.1.0.196) (32-bit)

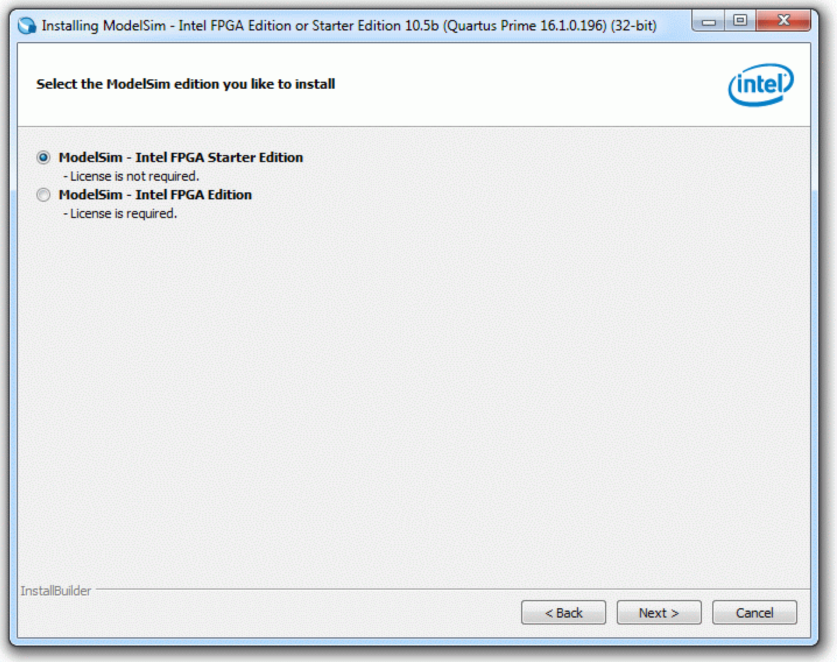
Setup - ModelSim - Intel FPGA Edition or Starter Edition 10.5b (Quartus Prime 16.1.0.196) (32-bit)

Welcome to the ModelSim - Intel FPGA Edition or Starter Edition 10.5b (Quartus Prime 16.1.0.196) (32-bit) Setup Wizard.

For more information about Intel FPGA software, go to http://www.altera.com.

Figure 1: ModelSim installation (Part 1)

2. Select ModelSim-Intel FPGA Starter Edition and click “Next >” (Figure 2).



Installing ModelSim - Intel FPGA Edition or Starter Edition 10.5b (Quartus Prime 16.1.0.196) (32-bit)

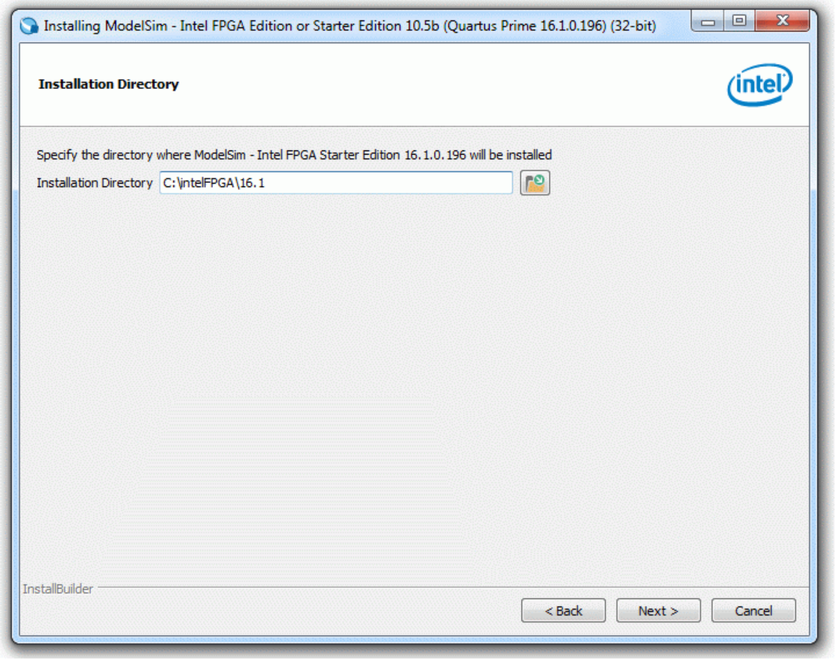
Select the ModelSim edition you like to install

● **ModelSim - Intel FPGA Starter Edition**
- License is not required.

○ **ModelSim - Intel FPGA Edition**
- License is required.

Figure 2: ModelSim installation (Part 2)

3. Accept the license agreement and click “Next >”.
4. Choose the directory for the final installation and click “Next >” (Figure 3). **Do not use spaces or special characters for the name of the folder and/or path!**



Installing ModelSim - Intel FPGA Edition or Starter Edition 10.5b (Quartus Prime 16.1.0.196) (32-bit)

Installation Directory

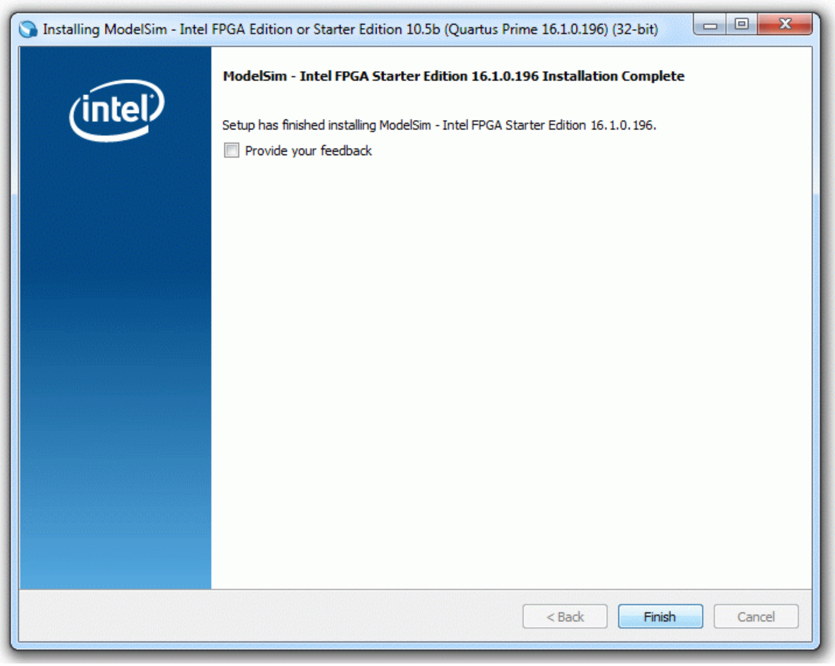
Specify the directory where ModelSim - Intel FPGA Starter Edition 16.1.0.196 will be installed

Installation Directory C:\intelFPGA\16.1

Figure 3: ModelSim installation (Part 3)

5. An installation summary is shown. Click “Next >” to start the installation.

6. Wait until the installation has finished (this usually takes a few minutes). The window shown in Figure 4 should appear. Click “Finish”. ModelSim-Intel FPGA Starter Edition has now been successfully installed.



Installing ModelSim - Intel FPGA Edition or Starter Edition 10.5b (Quartus Prime 16.1.0.196) (32-bit)

ModelSim - Intel FPGA Starter Edition 16.1.0.196 Installation Complete

Setup has finished installing ModelSim - Intel FPGA Starter Edition 16.1.0.196.

□ Provide your feedback

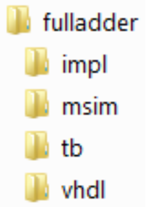
Figure 4: ModelSim installation (Part 4)

VHDL Simulation Example

Download the file `design.zip` from the CIS site of this lecture which contains the Full Adder example. Unpack the archive into e.g. the folder:

```
d:/work/design
```

Note: Omit special characters or blanks in the path name! The directory `design` contains the directory `fulladder` with further subdirectories (see Figure 5).



fulladder
- impl
- msim
- tb
- vhdl

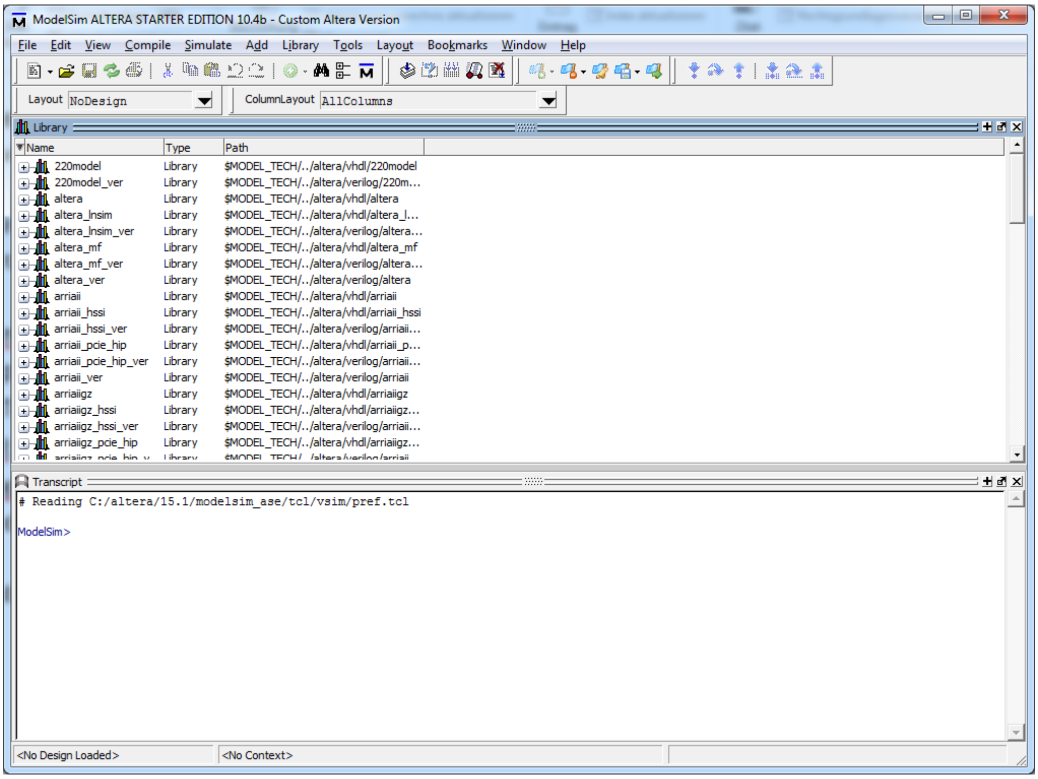
Figure 5: Directory structure

The details regarding the design as well as the VHDL language itself are not relevant for the simulation setup. They will be examined in the following Distance Learning Letters.

The directory structure depicted in Figure 5 is a recommended setup for working with VHDL which has proven useful in practice. There are four subdirectories in this setup:

- `impl`: This directory contains data to implement the design on an FPGA (will be explained in the second part of the lecture and in the distance learning letter “Vivado HL WebPACK Installation/Introduction”).
- `msim`: This directory contains all relevant files which are needed for the simulation with ModelSim, e.g., simulation scripts.
- `tb`: This directory contains all VHDL files of the testbench[1].
- `vhdl`: This directory contains all relevant VHDL design files of the Full Adder example.

Now start the ModelSim VHDL simulator. The window shown in Figure 6 will open (as well an additional welcome window, if you start ModelSim for the first time). It contains the command line of the ModelSim simulator.



Layout: NoDesign

ColumnLayout: AllColumns

Name	Type	Path
220model	Library	$MODEL_TECH/../altera/vhdl/220model
220model_ver	Library	$MODEL_TECH/../altera/verilog/220m...
altera	Library	$MODEL_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_l...
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_mf	Library	$MODEL_TECH/../altera/vhdl/altera_mf
altera_mf_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_ver	Library	$MODEL_TECH/../altera/verilog/altera
arriaii	Library	$MODEL_TECH/../altera/vhdl/arriaii
arriaii_hssi	Library	$MODEL_TECH/../altera/vhdl/arriaii_hssi
arriaii_hssi_ver	Library	$MODEL_TECH/../altera/verilog/arriaii...
arriaii_pcie_hip	Library	$MODEL_TECH/../altera/vhdl/arriaii_p...
arriaii_pcie_hip_ver	Library	$MODEL_TECH/../altera/verilog/arriaii...
arriaii_ver	Library	$MODEL_TECH/../altera/verilog/arriaii
arriaiigz	Library	$MODEL_TECH/../altera/vhdl/arriaiigz
arriaiigz_hssi	Library	$MODEL_TECH/../altera/vhdl/arriaiigz...
arriaiigz_hssi_ver	Library	$MODEL_TECH/../altera/verilog/arriaii...
arriaiigz_pcie_hip	Library	$MODEL_TECH/../altera/vhdl/arriaiigz...

```
# Reading C:/altera/15.1/modelsim_ase/tcl/vsim/pref.tcl

ModelSim>
```

<No Design Loaded>

<No Context>

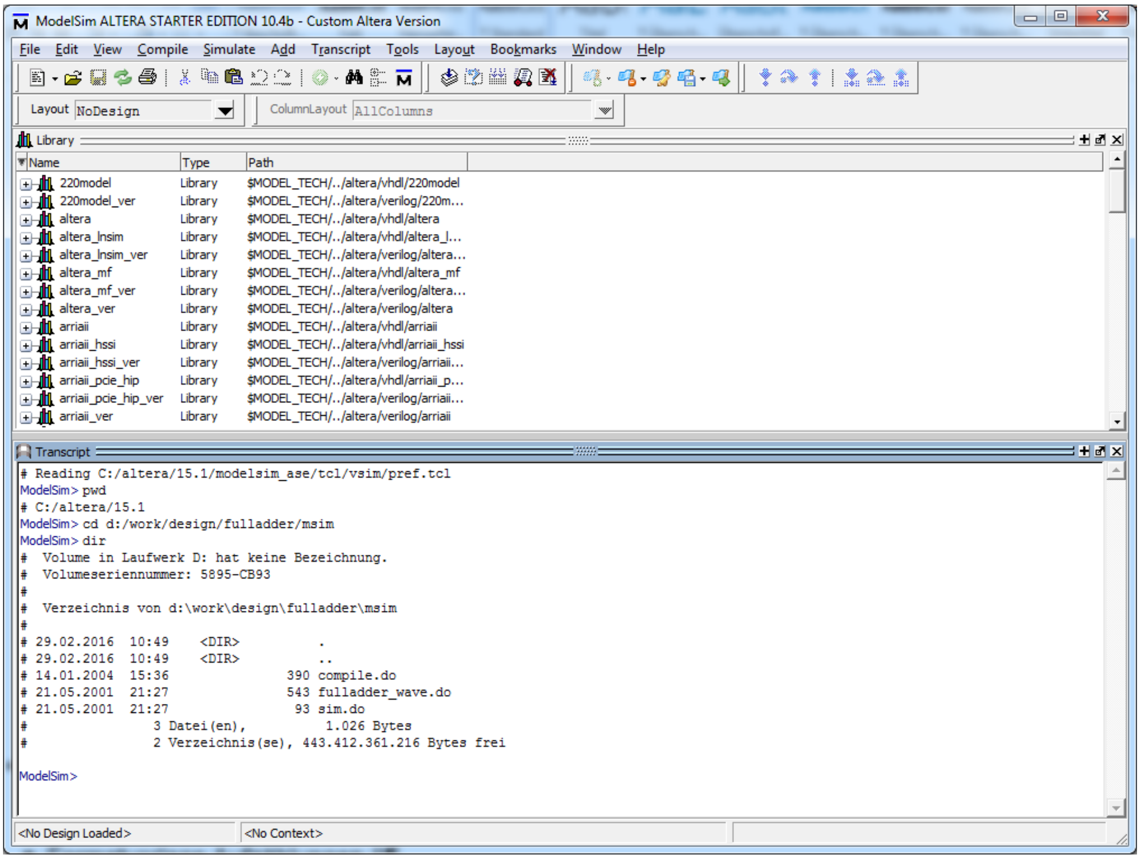
Figure 6: Initial ModelSim window

[1] The term `testbench` will be explained in the following Distance Learning Letters.

Enter the command `pwd` into the console window with the title `Transcript`. This command will print the path of the current working directory. Use the command `cd` to change into the subdirectory `msim` of the Full Adder VHDL project. The path has to be specified with "/" instead of "\" since the console window operates in a Tck/Tk[2] environment. A possible command can be e.g.:

```
cd d:/work/design/fulladder/msim
```

After changing the directory, type the command `dir` into the console. The output should look similar to what is shown in Figure 7.


ModelSim ALTERA STARTER EDITION 10.4b - Custom Altera Version
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
Library
Transcript

Layout NoDesign
ColumnLayout AllColumns

Name	Type	Path
220model	Library	$MODEL_TECH/../altera/vhdl/220model
220model_ver	Library	$MODEL_TECH/../altera/verilog/220m...
altera	Library	$MODEL_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_l...
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_mf	Library	$MODEL_TECH/../altera/vhdl/altera_mf
altera_mf_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_ver	Library	$MODEL_TECH/../altera/verilog/altera
arriaii	Library	$MODEL_TECH/../altera/vhdl/arriaii
arriaii_hssi	Library	$MODEL_TECH/../altera/vhdl/arriaii_hssi
arriaii_hssi_ver	Library	$MODEL_TECH/../altera/verilog/arriaii...
arriaii_pcie_hip	Library	$MODEL_TECH/../altera/vhdl/arriaii_p...
arriaii_pcie_hip_ver	Library	$MODEL_TECH/../altera/verilog/arriaii...
arriaii_ver	Library	$MODEL_TECH/../altera/verilog/arriaii

```
# Reading C:/altera/15.1/modelsim_ase/tcl/vsim/pref.tcl
ModelSim> pwd
# C:/altera/15.1
ModelSim> cd d:/work/design/fulladder/msim
ModelSim> dir
#  Volume in Laufwerk D: hat keine Bezeichnung.
#  Volumeseriennummer: 5895-CB93
#
#  Verzeichnis von d:\work\design\fulladder\msim
#
# 29.02.2016  10:49    <DIR>          .
# 29.02.2016  10:49    <DIR>          ..
# 14.01.2004  15:36               390 compile.do
# 21.05.2001  21:27               543 fulladder_wave.do
# 21.05.2001  21:27                93 sim.do
#                3 Datei(en),          1.026 Bytes
#                2 Verzeichnis(se), 443.412.361.216 Bytes frei

ModelSim>
```

<No Design Loaded>
<No Context>

Figure 7: Content of the directory `msim`

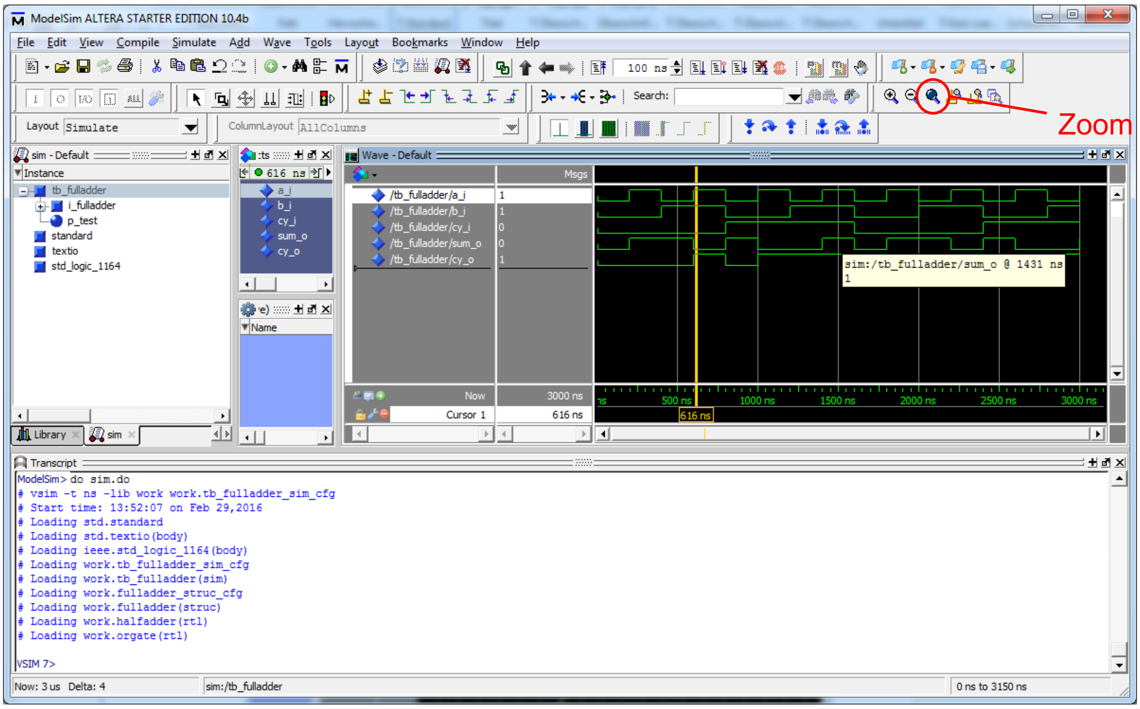
The directory `msim` contains three files with the suffix `.do`. These are script files which automate the simulation process of the Full Adder example. It is recommended to use these script files as templates for the simulations done in this course.

Now enter the following commands in the order listed below:

[2] Tcl/Tk (from "Tool command language / Tool kit") is a platform independent scripting language with a tool kit for fast graphical user interface (GUI) development.

```
vlib work
do compile.do
do sim.do
```

The first command `vlib work` creates a library with the name `work` in the current directory. This library is required for each and every simulation. The second command `do compile.do` executes the `compile.do` script of the current directory. This script compiles all VHDL design and testbench files which are necessary in order to simulate the Full Adder project. The third command `do sim.do` loads the compiled design from the work library, issues a simulation run and stops the simulation after 3000 ns. In addition, several windows are opened and five signals of the Full Adder project are displayed in the `wave` window. Click the “Zoom Full” icon in the menu bar (Figure *8*) to display the full simulation run.



```
ModelSim> do sim.do
# vsim -t ns -lib work work.tb_fulladder_sim_cfg
# Start time: 13:52:07 on Feb 29,2016
# Loading std.standard
# Loading std.textio(body)
# Loading ieee.std_logic_1164(body)
# Loading work.tb_fulladder_sim_cfg
# Loading work.tb_fulladder(sim)
# Loading work.fulladder_struc_cfg
# Loading work.fulladder(struc)
# Loading work.halfadder(rtl)
# Loading work.orgate(rtl)

VSIM 7>
```

Now: 3 us Delta: 4 sim:/tb_fulladder

Figure 8: ModelSim window during simulation

It is also possible to execute all these steps with the help of the GUI. This will be explained in the course of this lecture. However, the simulator can be operated much faster and more comfortably with the help of the commands and scripts mentioned above. Therefore, the following page gives a brief explanation of both scripts.

compile.do

```
vcom ../vhdl/halfadder_.vhd
vcom ../vhdl/halfadder_rtl.vhd
vcom ../vhdl/halfadder_rtl_cfg.vhd
vcom ../vhdl/orgate_.vhd
vcom ../vhdl/orgate_rtl.vhd
vcom ../vhdl/orgate_rtl_cfg.vhd
vcom ../vhdl/fulladder_.vhd
vcom ../vhdl/fulladder_struc.vhd
vcom ../vhdl/fulladder_struc_cfg.vhd
vcom ../tb/tb_fulladder_.vhd
vcom ../tb/tb_fulladder_sim.vhd
vcom ../tb/tb_fulladder_sim_cfg.vhd
```

Path and name of the VHDL file that is to be compiled.

The sequence of compilation (Entity before Architecture before Configuration is important!

The vcom command is used in order to compile VHDL files. It can also be used directly in the command line.

sim.do

```
vsim -t ns -novopt -lib work work.tb_fulladder_sim_cfg
view *
do fulladder_wave.do
run 3000 ns
```

This command starts the simulation.

Optional: Time resolution of the simulation (here: nanoseconds).

Provides compatibility between ModelSim and QuestaSim

Library which contains the compiled testbench.

Opens additional windows for simulation (wave…).

Testbench that is to be simulated.

Run the simulation for 3000 ns.

Loads a previously saved file which contains a list of signals that get displayed in the wave window.

Important commands in ModelSim

Command	Description
`help` *command*	Prints out a short description of the *command*
`pwd`	Current working directory
`dir`	Lists the content of the directory
`cd`	Change directory
`do` *script-file*	Execute the specified scriptfile
`vcom` *filename.vhd*	Compiles the VHDL file *filename.vhd* into the default library, i.e. the library *work*
`vsim` *work.testbench*	Loads the compiled *testbench* from the library *work*
`view` *windowname*	Opens the ModelSim window *windowname* (e.g. *wave, signals*, *variables*, *structure*, …)
`force` …	Forces a signal of the simulation to a specific value
`restart -f`	Restarts the simulation from time zero
`history`	Displays a history of the last issued commands
`quit -sim`	Terminates the current simulation
`quit -f`	Terminates ModelSim
`log -r *`	Records all signals in the design including sub-unit signals (“-r”). They can be added to a *wave* window later on.
`run 100 us`	Runs the simulation for 100 µs

Abbreviations

VHDL	Very High Speed Integrated Circuit Hardware Description Language
CAD	Computer Aided Design
Tcl/Tk	Tool command language / Tool kit
GUI	Graphical User Interface

References

The know-how required to develop digital integrated circuits consists by about 50 % of EDA tool know-how. Therefore, it is of utmost importance to know how to operate these tools. An introductory course can only describe the basic operating principle of such a tool.

The ModelSim simulator is one of the industry's leading tools for HDL simulation (Verilog and VHDL). It is recommended to have a look at the extensive documentation and examples in the directories `docs` and `example` of the ModelSim installation.

Questions

1. Find out what the three different methods of the command `force` are. Explain the differences.
2. Recall the simulation of the Full Adder example. This example shows the necessary three main steps in order to simulate a VHDL design. What are they?
3. In which order do VHDL files have to be compiled?
4. Why is it useful to have a uniform directory structure for VHDL designs and the related CAD tools?
5. How do you asses the ratio between tool-know-how (e.g. simulator) to basic-know-how (e.g. VHDL, Verilog, Digital Circuit Technology …)?

Version

Version 0.1, 2006-08-16.	Dokument angelegt.
Version 0.2, 2010-08-31	Switched to Altera.
Version 0.3, 2011-06-24	Document translated and updated.
Version 0.4, 2013-08-03	Modifications due to changes in ModelSim.
Version 0.5, 2015-08-31	Minor updates.
Version 0.6, 2016-02-29	Document updated.
Version 0.7, 2016-03-04	Proofreading.
Version 0.8, 2017-01-25	Switched from ModelSim-Altera to ModelSim-Intel; -novopt option added

If you find errors or inconsistencies, please report them to the supervisors of this lecture via email. Thank you!