EDA322Digital Design Lab

LAB 1

Designed by: Anurag Negi, Angelos Arelakis, Ioannis Sourdis

In this lab you will familiarize yourself with two sets of tools – the first, called QuestaSim(or ModelSim – no difference between these two tools for the course requirements) lets you simulate your VHDL hardware descriptions, while the second, called Xilinx ISE,allows you to synthesize your designs and target a variety of hardware platforms. This lab assumes that you are at least aware of what FPGAs are and where they can be used.

We will see how to use the following tools in this lab:

1. QuestaSim/ModelSim: This tool allows you to compile and simulate descriptions of digital logic in VHDL. A student edition is available free for Windows.
2. XilinxISE: This tool allows you to synthesize your VHDL designs and map the resulting logic onto a variety of target devices. We will target the Nexys3 board equipped with a Spartan 6 (XC6LX16-CS324) device. Note the number in parenthesis, as it would be needed to correctly setup your development environment when using Xilinx ISE. Again, a free student license is available for this software.

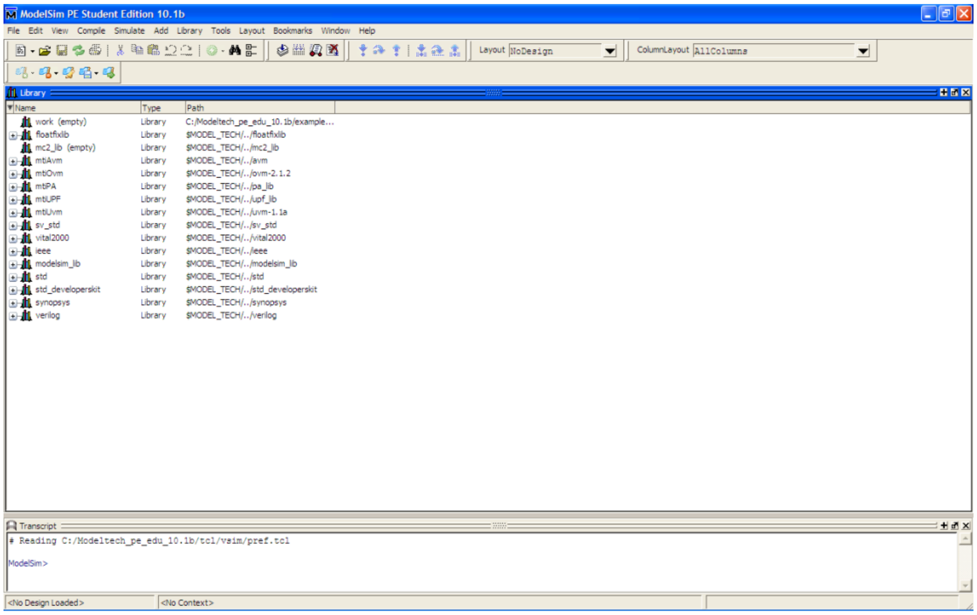
This lab will require you to completethree tasks:

1. Learn to recognize and understand the use of several commonly occurring digital logic components.
2. Simulate a sample VHDL design inQuestaSim/ModelSimand check for correctness of operation using a “do” file.
3. Synthesize the sample VHDL design targeting a Xilinx Spartan-6 FPGA platform and read off some relevant design parameters from reports.

Please download eda322_lab1.zip from PingPong and extract its contents in a local folder which you can access from within QuestaSim/Modelsim and XilinxISE. The .zip archive contains two files: eda322_lab1.vhd and eda322_lab1.do.

Using QuestaSim/ModelSim for simulating VHDL designs – Task 1

Start QuestaSim/ModelSim. You should see a window similar to the one shown below.


ModelSim PE Student Edition 10.1b

Library

Name	Type	Path
work (empty)	Library	C:/Modeltech_pe_edu_10.1b/example...
floatfixlib	Library	$MODEL_TECH/../floatfixlib
mc2_lib (empty)	Library	$MODEL_TECH/../mc2_lib
mtiAvm	Library	$MODEL_TECH/../avm
mtiOvm	Library	$MODEL_TECH/../ovm-2.1.2
mtiPA	Library	$MODEL_TECH/../pa_lib
mtiUPF	Library	$MODEL_TECH/../upf_lib
mtiUvm	Library	$MODEL_TECH/../uvm-1.1a
sv_std	Library	$MODEL_TECH/../sv_std
vital2000	Library	$MODEL_TECH/../vital2000
ieee	Library	$MODEL_TECH/../ieee
modelsim_lib	Library	$MODEL_TECH/../modelsim_lib
std	Library	$MODEL_TECH/../std
std_developerskit	Library	$MODEL_TECH/../std_developerskit
synopsys	Library	$MODEL_TECH/../synopsys
verilog	Library	$MODEL_TECH/../verilog

Transcript

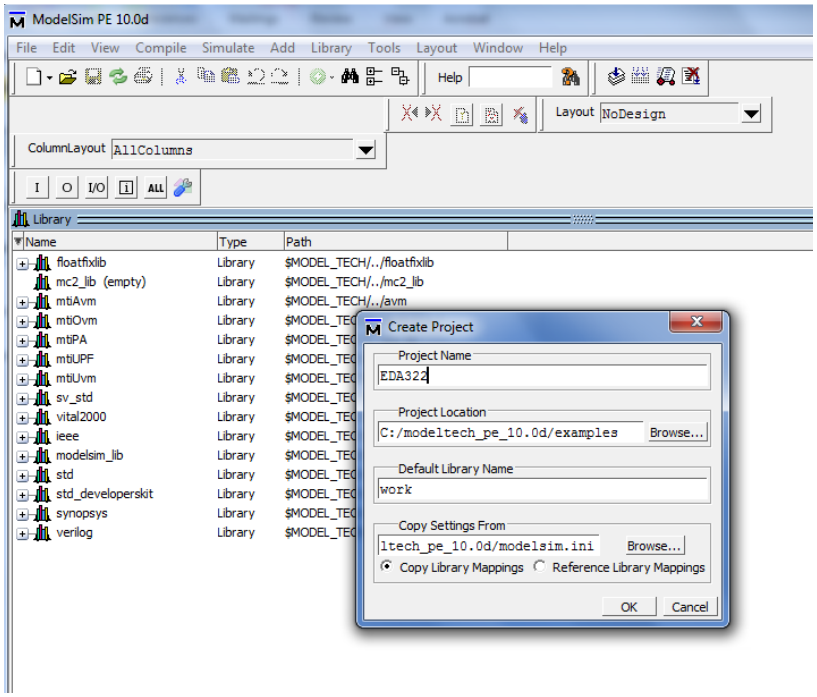
```
# Reading C:/Modeltech_pe_edu_10.1b/tcl/vsim/pref.tcl

ModelSim>
```

<No Design Loaded> <No Context>

File Edit View Compile Simulate Add Library Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns

We will now create a new project, by selecting File->New->Project. Name the project EDA322. All code you write in this course canbe part of this project. Choose a suitable Project Location, so that your code is maintained in a secure place.


ModelSim PE 10.0d

Library

Name	Type	Path
floatfixlib	Library	$MODEL_TECH/../floatfixlib
mc2_lib (empty)	Library	$MODEL_TECH/../mc2_lib
mtiAvm	Library	$MODEL_TECH/../avm
mtiOvm	Library	$MODEL_TEC
mtiPA	Library	$MODEL_TEC
mtiUPF	Library	$MODEL_TEC
mtiUvm	Library	$MODEL_TEC
sv_std	Library	$MODEL_TEC
vital2000	Library	$MODEL_TEC
ieee	Library	$MODEL_TEC
modelsim_lib	Library	$MODEL_TEC
std	Library	$MODEL_TEC
std_developerskit	Library	$MODEL_TEC
synopsys	Library	$MODEL_TEC
verilog	Library	$MODEL_TEC

Create Project

Project Name: EDA322

Project Location: C:/modeltech_pe_10.0d/examples

Default Library Name: work

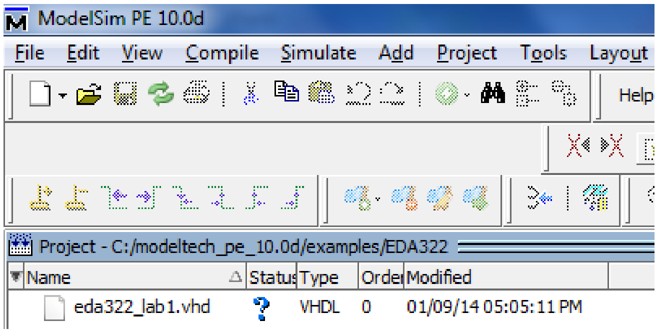
Copy Settings From: ltech_pe_10.0d/modelsim.ini

● Copy Library Mappings ○ Reference Library Mappings

File Edit View Compile Simulate Add Library Tools Layout Window Help
Help
Layout NoDesign
ColumnLayout AllColumns
Browse...
Browse...
OK
Cancel

Once you click OK, the software will ask you to add items to the project. You can select "Add existing file" and add the VHDL file (with the extension ".vhd" or ".vhdl") to the project. You can add files to the project at a later point in time as well. It is not required for the files to be located in the project folder.

The window should now be similar to the image below:

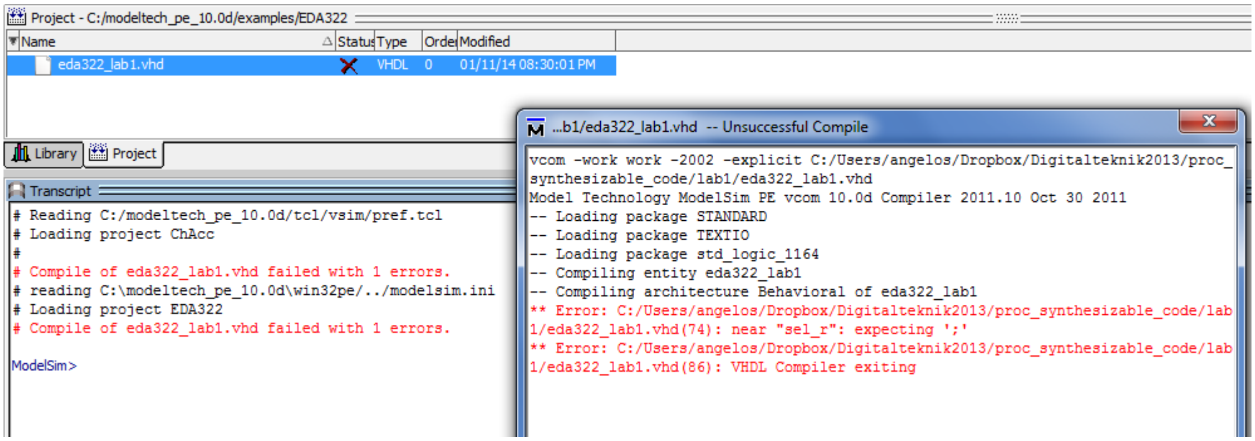


Project - C:/modeltech_pe_10.0d/examples/EDA322

Name	Status	Type	Order	Modified
eda322_lab1.vhd	?	VHDL	0	01/09/14 05:05:11 PM

Double clicking on the file name will open the file for editing in an editing panel. You can now take a look at how the description of a simple hardware component looks like in VHDL.

You can compile it by selecting Compile->Compile All from the menu. If there are errors or warnings, you can see them by double clicking on the error message (appeared in red) in the Transcript window. A new window appears where the errors/warnings are listed, like in the figure below. Using this information you can fix them and recompile.



Project - C:/modeltech_pe_10.0d/examples/EDA322

Name	Status	Type	Order	Modified
eda322_lab1.vhd	✗	VHDL	0	01/11/14 08:30:01 PM

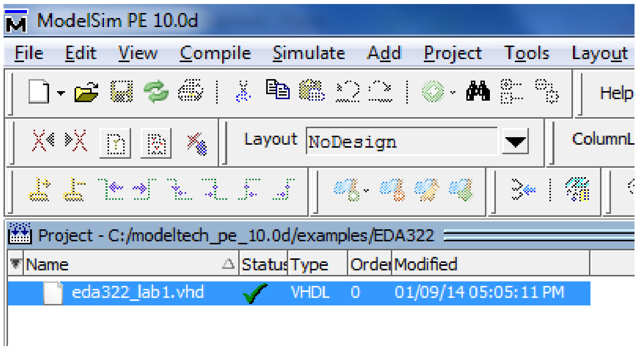
```
# Reading C:/modeltech_pe_10.0d/tcl/vsim/pref.tcl
# Loading project ChAcc
#
# Compile of eda322_lab1.vhd failed with 1 errors.
# reading C:\modeltech_pe_10.0d\win32pe/../modelsim.ini
# Loading project EDA322
# Compile of eda322_lab1.vhd failed with 1 errors.

ModelSim>
```

...b1/eda322_lab1.vhd -- Unsuccessful Compile

```
vcom -work work -2002 -explicit C:/Users/angelos/Dropbox/Digitalteknik2013/proc_
synthesizable_code/lab1/eda322_lab1.vhd
Model Technology ModelSim PE vcom 10.0d Compiler 2011.10 Oct 30 2011
-- Loading package STANDARD
-- Loading package TEXTIO
-- Loading package std_logic_1164
-- Compiling entity eda322_lab1
-- Compiling architecture Behavioral of eda322_lab1
** Error: C:/Users/angelos/Dropbox/Digitalteknik2013/proc_synthesizable_code/lab
1/eda322_lab1.vhd(74): near "sel_r": expecting ';'
** Error: C:/Users/angelos/Dropbox/Digitalteknik2013/proc_synthesizable_code/lab
1/eda322_lab1.vhd(86): VHDL Compiler exiting
```

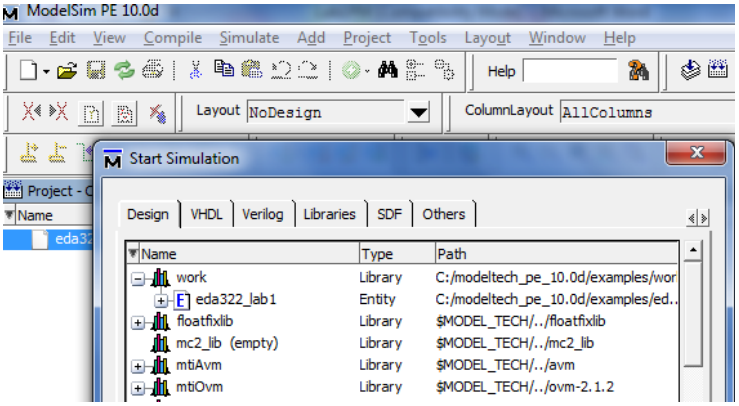
When compilation is successful, you should be able to see a green tick mark in the status field beside the file name.



Project - C:/modeltech_pe_10.0d/examples/EDA322

Name	Status	Type	Order	Modified
eda322_lab1.vhd	✓	VHDL	0	01/09/14 05:05:11 PM

You can simulate the design by selecting Simulate -> Start Simulation. It will present a dialogue box asking you to choose the design entity that you want to simulate. In the "work" library, choose the name of the entity (or the top level entity if you have several entity descriptions comprising your design).In this lab, the entity name "eda322_lab1"is the top level entity in your default "work" library.

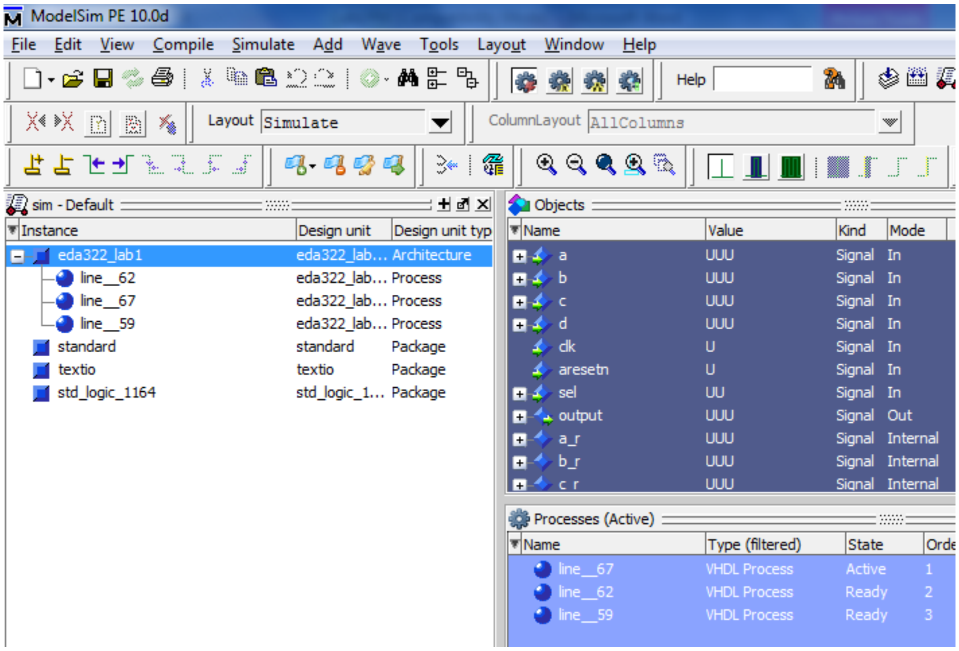

ModelSim PE 10.0d
File Edit View Compile Simulate Add Project Tools Layout Window Help
Help
Project - C
Name
eda32
Start Simulation
Design VHDL Verilog Libraries SDF Others

Layout NoDesign

ColumnLayout AllColumns

Name	Type	Path
work	Library	C:/modeltech_pe_10.0d/examples/wor
eda322_lab1	Entity	C:/modeltech_pe_10.0d/examples/ed..
floatfixlib	Library	$MODEL_TECH/../floatfixlib
mc2_lib (empty)	Library	$MODEL_TECH/../mc2_lib
mtiAvm	Library	$MODEL_TECH/../avm
mtiOvm	Library	$MODEL_TECH/../ovm-2.1.2

You should now see a window that looks similar to the one shown below:


ModelSim PE 10.0d
File Edit View Compile Simulate Add Wave Tools Layout Window Help
Help
sim - Default
Objects
Processes (Active)

Layout Simulate

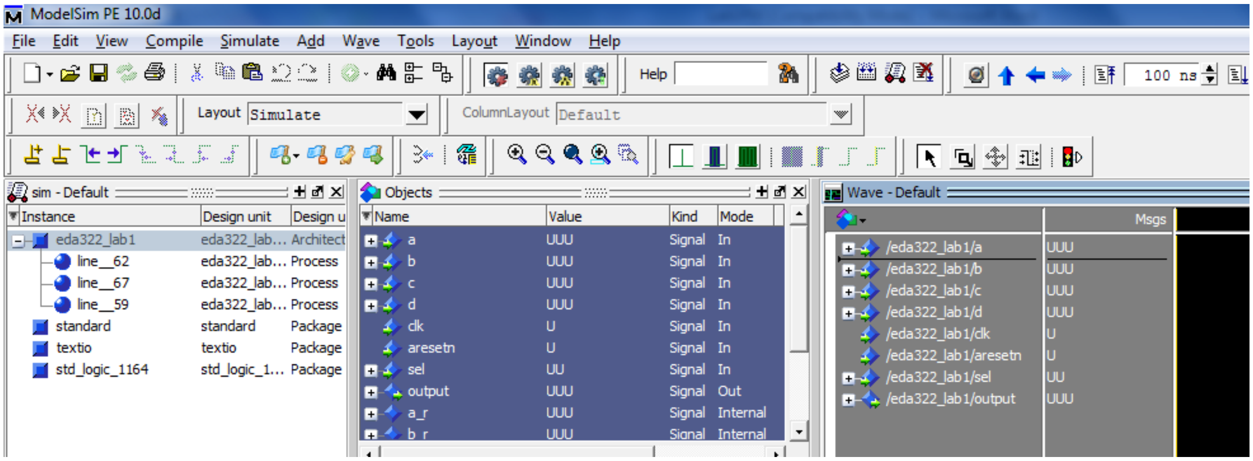
ColumnLayout AllColumns

Instance	Design unit	Design unit typ
eda322_lab1	eda322_lab...	Architecture
line__62	eda322_lab...	Process
line__67	eda322_lab...	Process
line__59	eda322_lab...	Process
standard	standard	Package
textio	textio	Package
std_logic_1164	std_logic_1...	Package

Name	Value	Kind	Mode
a	UUU	Signal	In
b	UUU	Signal	In
c	UUU	Signal	In
d	UUU	Signal	In
clk	U	Signal	In
aresetn	U	Signal	In
sel	UU	Signal	In
output	UUU	Signal	Out
a_r	UUU	Signal	Internal
b_r	UUU	Signal	Internal
c_r	UUU	Signal	Internal

Name	Type (filtered)	State	Orde
line__67	VHDL Process	Active	1
line__62	VHDL Process	Ready	2
line__59	VHDL Process	Ready	3

Signal names in the simulated entity appear in the Objects panel. You can right click on signal names you want to view graphically in the panel and select "Add Wave". A new waveform panel will appear with the signal(s) shown. In alternative, you can select the signals (in Objects) you would like to add in the waveform and then drag and drop them in the signal area of it.

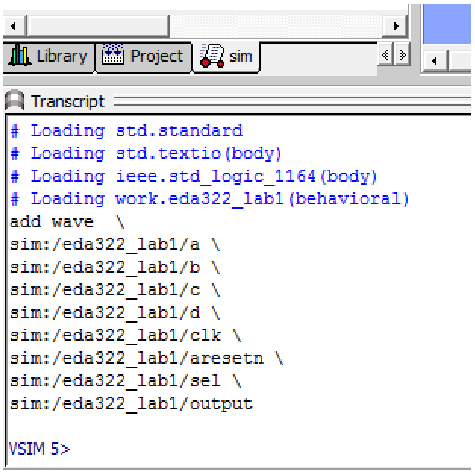


Instance	Design unit	Design u
eda322_lab1	eda322_lab...	Architect
line__62	eda322_lab...	Process
line__67	eda322_lab...	Process
line__59	eda322_lab...	Process
standard	standard	Package
textio	textio	Package
std_logic_1164	std_logic_1...	Package

Name	Value	Kind	Mode
a	UUU	Signal	In
b	UUU	Signal	In
c	UUU	Signal	In
d	UUU	Signal	In
clk	U	Signal	In
aresetn	U	Signal	In
sel	UU	Signal	In
output	UUU	Signal	Out
a_r	UUU	Signal	Internal
b_r	UUU	Signal	Internal

	Msgs
/eda322_lab1/a	UUU
/eda322_lab1/b	UUU
/eda322_lab1/c	UUU
/eda322_lab1/d	UUU
/eda322_lab1/clk	U
/eda322_lab1/aresetn	U
/eda322_lab1/sel	UU
/eda322_lab1/output	UUU

Also notice the Transcript panel at the bottom of the QuestaSim/Modelsim window (see Figure in the next page), which also displays a command line.

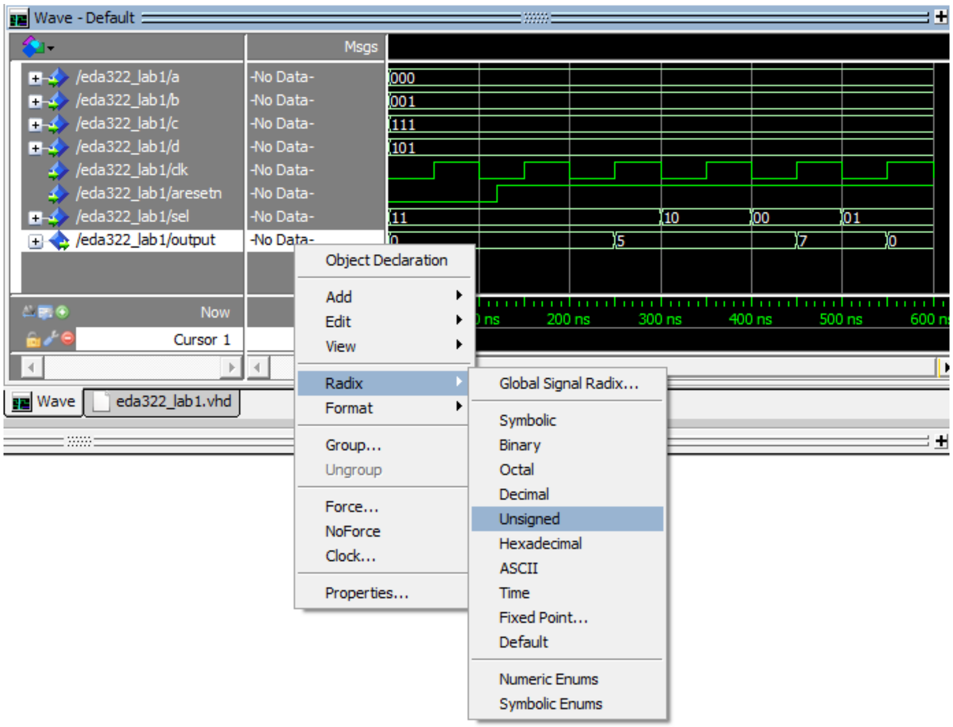


```
# Loading std.standard
# Loading std.textio(body)
# Loading ieee.std_logic_1164(body)
# Loading work.eda322_lab1(behavioral)
add wave  \
sim:/eda322_lab1/a \
sim:/eda322_lab1/b \
sim:/eda322_lab1/c \
sim:/eda322_lab1/d \
sim:/eda322_lab1/clk \
sim:/eda322_lab1/aresetn \
sim:/eda322_lab1/sel \
sim:/eda322_lab1/output

VSIM 5>
```

You can now enter commands to test your module. Type "do eda322_lab1.do" to run the do file you have downloaded. Make sure that the "do" file is placed in the working directory for the project.

You can see that values in the waveform window are updated when the run command is executed. The figure below shows various signal transitions when a VHDL module is simulated. You can zoom in/out using the respective "magnifying glass" tools. You can also change the radix which the signals appear in by right clicking on a signal (or a set of them) and selecting radix. A signal can be added a number of times using different radixes (can be done through the graphic interface or inside the "do" file), which can be practical.


Wave - Default
Msgs
/eda322_lab1/a -No Data- 000
/eda322_lab1/b -No Data- 001
/eda322_lab1/c -No Data- 111
/eda322_lab1/d -No Data- 101
/eda322_lab1/clk -No Data-
/eda322_lab1/aresetn -No Data-
/eda322_lab1/sel -No Data- 11 10 00 01
/eda322_lab1/output -No Data- 0 5 7 0
Now
Cursor 1
0 ns 200 ns 300 ns 400 ns 500 ns 600 ns
Wave
eda322_lab1.vhd
Object Declaration
Add
Edit
View
Radix
Format
Group...
Ungroup
Force...
NoForce
Clock...
Properties...
Global Signal Radix...
Symbolic
Binary
Octal
Decimal
Unsigned
Hexadecimal
ASCII
Time
Fixed Point...
Default
Numeric Enums
Symbolic Enums

The "do" file contains a sequence of commands that can also be entered at the console. A quick guide and a command reference have been provided for your guidance on PingPong. It is strongly recommended to do testing (adding signals to the waveform view, assign signal values, run simulation time)using "do" files.This way makes it very easy to repeat the simulation when debugging and during the lab demonstrations.

Take some time at this point to familiarize yourselves with the syntax of the *.do* file and with the features available in the simulator -- like adding/removing signals from the waveform window, zooming in and out, cursor control, changing the way values are represented in the window (radix). Getting familiar with these controls will help you test and debug code more effectively – a skill which will be valuable throughout the lab assignment.

Try to answer the following questions and check your answers with an assistant:

1. What functionality is modeled in the code?

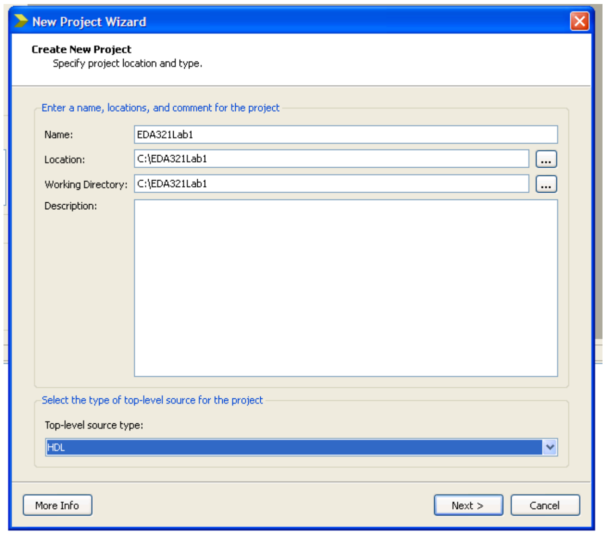
2. How many flip-flops have been used in the code?

3. What happens when the "aresetn" port is set to 0?

Synthesizing your design – Task 2

You can use Xilinx ISE to synthesize your design. Synthesis converts your VHDL description into a netlist of components that can be mapped onto a target device like an FPGA. For this lab we will target a Xilinx Spartan 6 FPGA (XC6SLX16) in a CSG324 package. The design tool also does the mapping onto the FPGA and applies user constraints (like mapping top level design ports onto pins on the FPGA board).

Run XilinxISE and click on File->New Project …
A dialogue box similar to the one shown below should open up. Give your project a name and provide a safe location and a working directory. Choose “Top-level source type” as HDL since you will be writing VHDL code to describe your designs. Notethat when you create a project using XilinxISE, a new folder will be created at the location you have chosen, in contrast to QuestaSimthat doesn´t do this.



New Project Wizard

Create New Project
Specify project location and type.

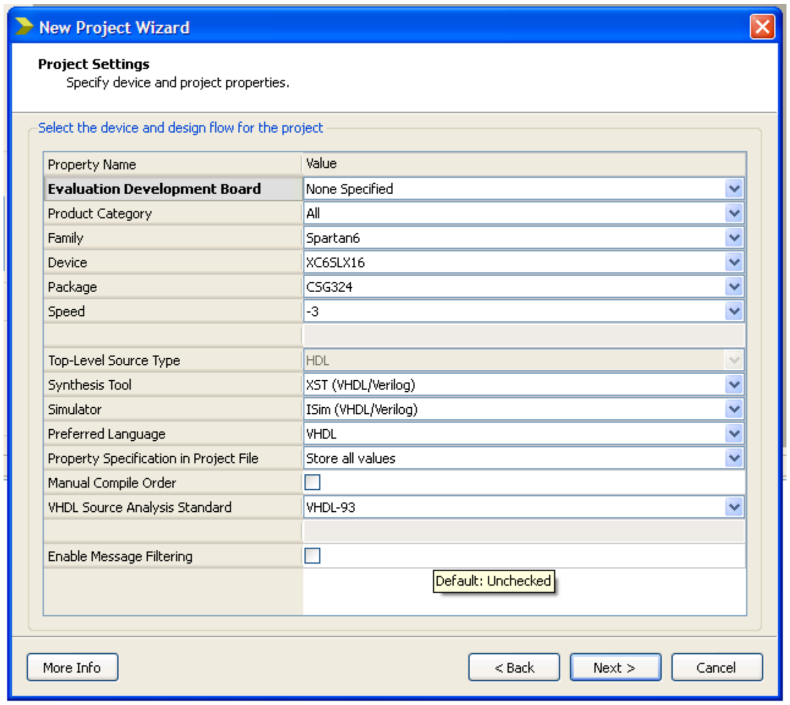
Enter a name, locations, and comment for the project

Name: EDA321Lab1
Location: C:\EDA321Lab1
Working Directory: C:\EDA321Lab1
Description:

Select the type of top-level source for the project

Top-level source type: HDL

Clicking next will bring you to a dialogue box similar to the one shown below:



New Project Wizard

Project Settings

Specify device and project properties.

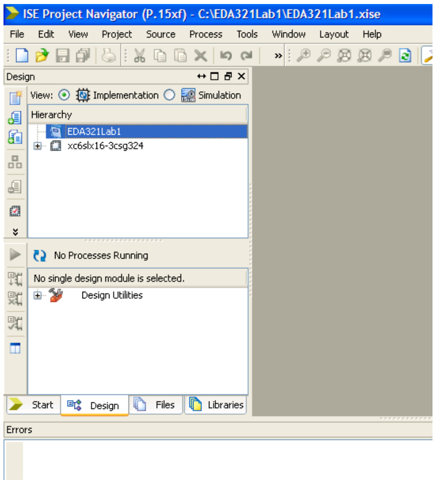
Select the device and design flow for the project

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan6
Device	XC6SLX16
Package	CSG324
Speed	-3
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	□
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	□

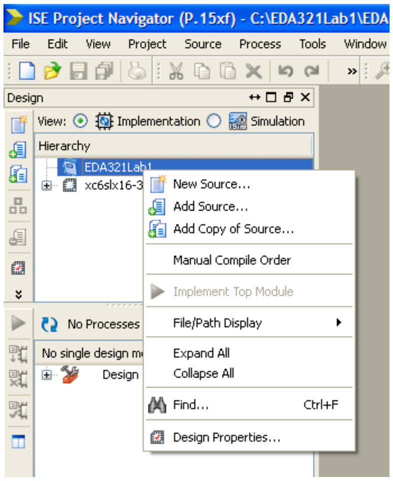
Default: Unchecked

Choose all settings as shown in the image above. Click “Next >” and then “Finish”. Note that these setting include choosing the correct FPGA device (as noted in the introductory section of this document) and the preferred hardware description language, which for this course is VHDL.

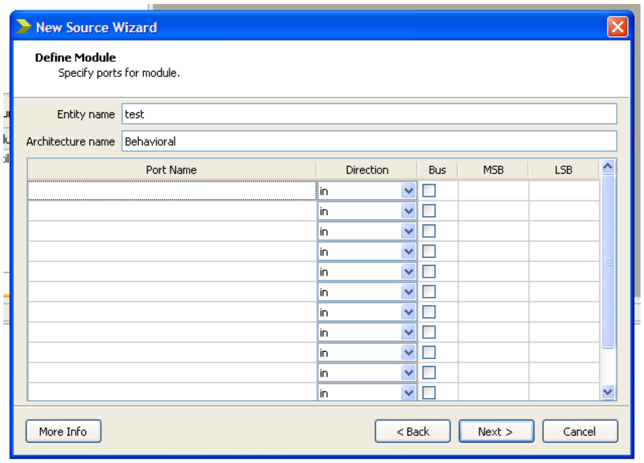
You should now have a Xilinx window than looks similar to the one shown below.



Right-clicking on the project name in the Design tab would provide a menu (shown below) that allows you to add sources to your Xilinx project.



You already have VHDL source provided to you. You can click on "Add Source…" which would allow you to browse and select an existing VHDL file.
You can also choose to use Xilinx ISE as an editor for creating new VHDL codes. In that case choose "New Source…" and provide a filename, selecting "VHDL Module" as source type. This would bring you to a dialogue box similar to the one shown below.



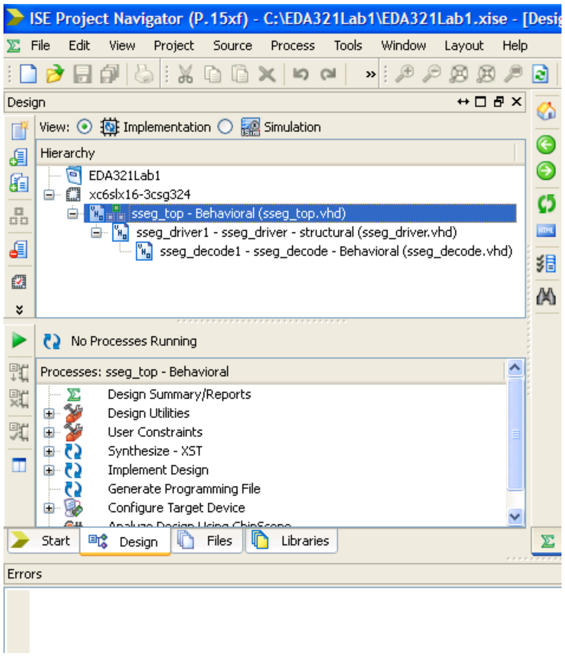
Entity name test

Architecture name Behavioral

Port Name	Direction	Bus	MSB	LSB
	in	□		
	in	□		
	in	□		
	in	□		
	in	□		
	in	□		
	in	□		
	in	□		
	in	□		
	in	□		
	in	□		

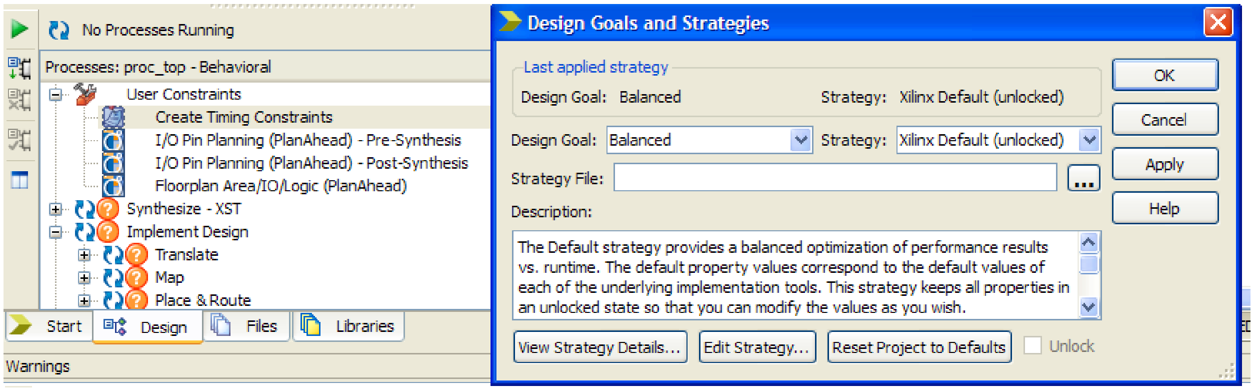
You can describe the VHDL entity using this dialogue box and the tool will automatically add that to the file, which you can then fill in to describe how the entity behaves. As you can see, it allows you to specify port names and directions. Check the "Bus" checkbox if a port is wider than a single bit.

Once you have added VHDL files and/or completed coding, you should have the Xilinx ISE window looking similar to the one shown below:


ISE Project Navigator (P.15xf) - C:\EDA321Lab1\EDA321Lab1.xise - [Desi
File Edit View Project Source Process Tools Window Layout Help
Design
View: ● Implementation ○ Simulation
Hierarchy
EDA321Lab1
xc6slx16-3csg324
sseg_top - Behavioral (sseg_top.vhd)
sseg_driver1 - sseg_driver - structural (sseg_driver.vhd)
sseg_decode1 - sseg_decode - Behavioral (sseg_decode.vhd)
No Processes Running
Processes: sseg_top - Behavioral
Design Summary/Reports
Design Utilities
User Constraints
Synthesize - XST
Implement Design
Generate Programming File
Configure Target Device
Analyze Design Using ChipScope
Start
Design
Files
Libraries
Errors

You may have to identify a top-level module if you have a hierarchy of VHDL modules before you begin synthesis. The “Source” menu allows you do to so.

Right-clicking on “Create Timing Constraints” under “User Constraints” and selecting “Design Goals and Strategies” brings up a dialogue box as shown below. It allows you to choose among several design goals that allow designers to target different area vs. power vs. performance tradeoffs. The image below shows a design where a “Balanced” approach has been selected.


No Processes Running
Processes: proc_top - Behavioral
User Constraints
Create Timing Constraints
I/O Pin Planning (PlanAhead) - Pre-Synthesis
I/O Pin Planning (PlanAhead) - Post-Synthesis
Floorplan Area/IO/Logic (PlanAhead)
Synthesize - XST
Implement Design
Translate
Map
Place & Route
Start
Design
Files
Libraries
Warnings
Design Goals and Strategies
OK
Cancel
Apply
Help
View Strategy Details...
Edit Strategy...
Reset Project to Defaults
□ Unlock

Last applied strategy

Design Goal: Balanced Strategy: Xilinx Default (unlocked)

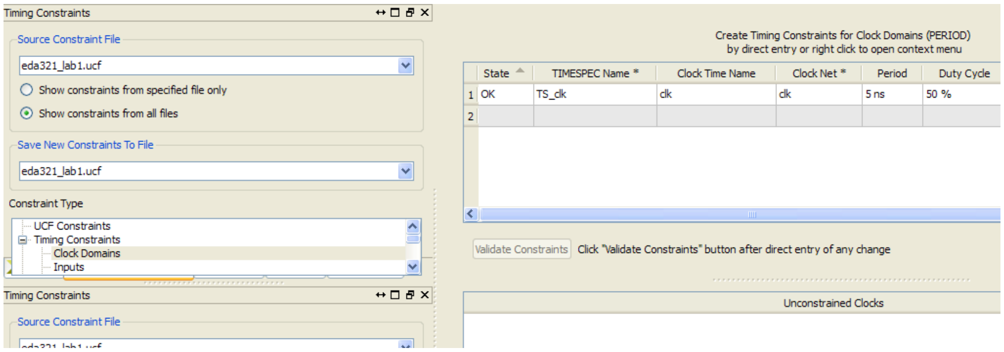
Design Goal: Balanced Strategy: Xilinx Default (unlocked)

Strategy File:

Description:

The Default strategy provides a balanced optimization of performance results vs. runtime. The default property values correspond to the default values of each of the underlying implementation tools. This strategy keeps all properties in an unlocked state so that you can modify the values as you wish.

Create a clock timing constraint by double clicking on “Create timing constraints” and selecting “Clock Domains” constraint type. Apply it to the signal called “clk” and constrain it to a 5ns period. The result would look similar to the image shown below:



Source Constraint File: eda321_lab1.ucf

○ Show constraints from specified file only

● Show constraints from all files

Save New Constraints To File: eda321_lab1.ucf

Create Timing Constraints for Clock Domains (PERIOD)
by direct entry or right click to open context menu

	State	TIMESPEC Name *	Clock Time Name	Clock Net *	Period	Duty Cycle
1	OK	TS_clk	clk	clk	5 ns	50 %
2						

Click "Validate Constraints" button after direct entry of any change

Source Constraint File: eda321_lab1.ucf

Save the constraint. Xilinx ISE creates a UCF (user constraints file) that contains text specifications for each constraint imposed by the user. The particular constraint instructs the tool to do everything possible to implement the circuit in such a way that the clock period does not exceed 5ns. This enables you to "push" the tool to achieve a minimum performance – something you will need to do in (the optional) Lab 7.

Now you are all set to begin synthesis of your design by double clicking on "Synthesize – XST" in the Design tab. Though this sample design will synthesize without errors or warnings, in future labs this is an important stage where you will be able to fix many problems that might exist in more complex designs.

Double click on implement design to invoke "Translate"/Map/PAR algorithms in the tool. These steps convert a synthesized netlist into a design that is suitable for implementation on the targeted device. Several optimizations may happen here to optimize based on resources available on the FPGA and constraints provided by the designer.

Take a look at the **Design Summary** section in the Synthesis report. Also take a look at **Post Place and Route Timing Report**. Try to understand the meaning and implications of various terms and numbers you see in these reports.

Try to answer the following questions and check your answers with an assistant**:**

1. How many flip-flops were used to implement the design?

2. How many LUTs (look-up tables) were used to implement the design?

3. What is the maximum net delay post place and route?

Perform a power analysis by running "Generatetext power report" under "Place and Route" ->"Generate post place and route static timing" in the design tab. What is the reported total design power in mW? Note that this number may be inaccurate for reasons beyond the scope of the present lab. The intention here is to introduce you to the types of analyses that can be performed using Xilinx ISE.

Hints and Tips:

Installing ModelSim and Xilinx on your personal device

The student version of *ModelSim* [1]and *Xilinx-ISE WebPACKDesign Software*[2]are sufficient for the labs in this course. You can download and use these tools onyour own device fromthe links provided at the end of this document.

Forcing values in the .do file

Depending on the version of the ModelSim you use, you are allowed to utilize differentformats forassigning a value to a signal. But, it is strongly recommended that you use the generally accepted explicit format where # symbol should be used to define the radix of a value. For instance:

- force mySignal 2#101

where binary value "101" is assigned to mySignal, or

- force mySignal 10#240

where decimal value "240" is assigned to mysignal.

Problem with generating the text power report:

Sometimes, after running “Generate Text Power Report”, the report is still inaccessible. If this happened, right click on “Generate Text Power Report”, go to “Process Properties” and make sure that “Produce Verbose Report” is checked. Rerun all.

References

[1] ModelSim PE Student Edition. [Online]. Available: http://www.mentor.com/company/higher_ed/modelsim-student-edition

[2] Xilinx ISE WebPACK Design. [Online]. Available: http://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html