MICROELECTRONIC SYSTEMS DESIGN RESEARCH GROUP

FB Elektrotechnik und Informationstechnik
Prof. Dr.-Ing. Norbert Wehn
Dozent:
Uwe Wasenmüller
Raum 12-213, wa@eit.uni-kl.de

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

# Task 5_B

## Synthesis and Place&Route of DCF-77 Decoder on FPGA und Post-Layout Simulation

## 1. Objective

Objective is to design the top level circuit of DCF-77 decoder and to perform exhaustive validation. Prior to implementation the circuit will be tested by a provided test bench. Subsequently synthesis as well as place and route on Xilinx FPGA will be performed. Post-layout simulation and testing on the board conclude this task.

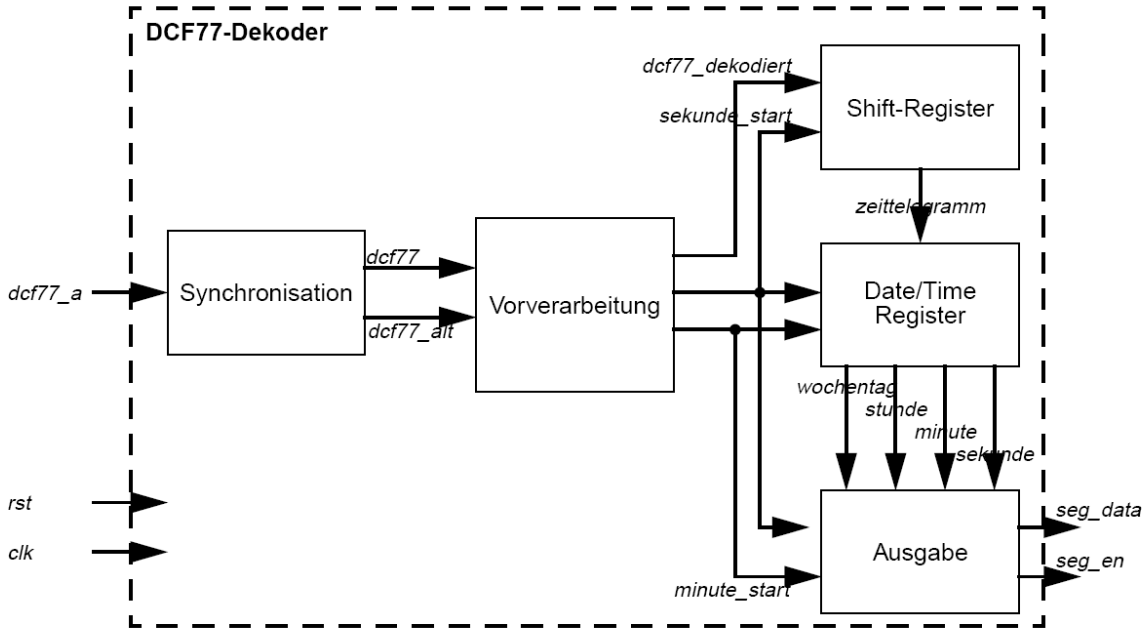
## 2. Approach part 1

Model of the top-level module must be described. Entity declaration is just provided; it may not be changed. For verification use the provided test bench. Steps in detail are:

- Analysis of provided entity declaration
- Analysis of specification
- Design of VHDL model
- Verification with test bench: Analysis of input stimuli; careful checking of expected results

## 3. Specification of Top-Level of DCF-77 Decoder

• Complete the file **src/dcf77_decoder.vhd**
• Specification




Implemented modules of prior task must be instantiated and properly connected. Use the block diagram above. All required component declarations exist just in the mentioned file.

Additionally to the specified outputs of the block diagram the entity declaration exhibits two further output signals you must control. Outputs ***seg_dp(7:1)*** control the decimal point of the 7-segment-displays. Outputs ***led(6:0)*** are connected with the 6 of LED displays on the board.

Decimal point of the rightmost 7-segment-display (Anzeige 0) is physically not connected to the FPGA. Left LED (LED 7) will be used for displaying the signal ***dcf77_a***. With some training you will recognize the impulse patterns.

Your circuit must control output ***seg_dp*** to separate hour-, minute and seconds values on the display by a decimal point on 7-segment-display.

On LEDs from left to right signals ***dcf77_dekodiert***, ***sekunde_start***, ***minute_start*** as well as bit pattern „1100“ must be shown. These signals can be used for debugging in case of problems.

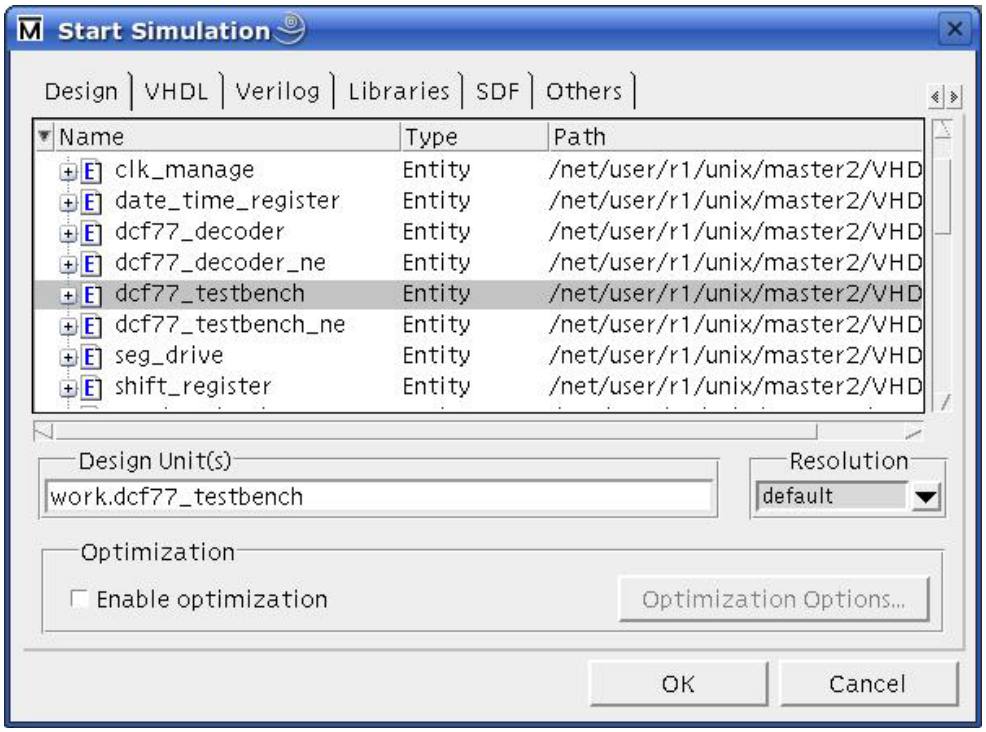
• Implementation

Describe your synthesizable VHDL model according to specification

• Pre Layout Simulation

Compile additionally VHDL-file **src/dcf77_testbench**. This file contains a test bench for validation of the whole circuit. This test bench instantiates your DCF77-decoder and generates all required input stimuli.

Start simulator; use menu ***„Simulate → Start Simulation“***. Choose *Design Unit* **dcf77_testbench;** keep *Simulator Resolution* on „**default**“.




Test bench creates all required input; you don't need any manual force statements. For verification of correct functionality the Signal ***display*** is given, which models in **hexadecimal radix** the 7-segment-display. Test bench automatically stops simulation after applying five test time frames. Thus you can perform simulation with the simulator command **„run -all".**

On signal ***display*** the following values are expected:

```
Wednesday 23:45:00 (time:  60.010 ms)
Monday    09:00:00 (time: 120.010 ms)
Tuesday   11:35:00 (time: 180.010 ms)
Wednesday 17:23:00 (time: 240.010 ms)
Thursday  23:59:00 (time: 300.010 ms)
```

Please check carefully all results including functionality of the seconds. Keep in mind, that the display shows sequentially from left to right new values.

**In case of incorrect behavior update the relevant modules und carry out the described steps again.**

# 4. Part 2

For part 2 of this exercise it is important to obtain the described use of Xilinx ISE tools. Before you synthesize your design with XST (part of ISE), analyze manually your modules regarding number and type of described registers and flip-flops. Also analyze manually the type and number of operators you have described in your VHDL models. After synthesizing the circuit compare your analysis carefully with the messages of the synthesis tool XST. For verification of your circuit after place & route same hints as for the pre layout verification must be taken into account.

# 5. Synthesis and Layout with Xilinx ISE

## 5.1 Analysis of required resources

Analyze the components „*vorverarbeitung*“ and „*date_time_register*“ regarding the modeled registers of your VHDL code and the modeled arithmetic operations (adder /subtracter / counter and comparator). Fill out the table below according to the given examples:

***Vorverarbeitung:***

| **Register** in vorverarbeitung | | |
|---|---|---|
| Signal | Definition in Line | Width |
| *Example: minute_start* | *Example: 11* | *Example: 1* |
| | | |

| Arithmetic **Operators** in vorverarbeitung | | |
|---|---|---|
| Type | Definition in Line | Width |
| *Example: Comparator (lessequal)* | *Example: 23* | *Example: 11 Bit* |
| | | |

***Date_time_register***

| **Register** in date_time_register | | |
|---|---|---|
| Signal | Definition in Line | Width |
| | | |

| Arithmetic **Operators** in date_time_register | | |
|---|---|---|
| Type | Definition in Line | Width |
| | | |

### 5.2 Synthesis with ISE

Up to now the circuit **dcf77_decoder.vhd** was used. We use a hardware board of Digilent; documentation of the board can be found in directory doc. Your circuit must be adapted to this lab board. These required changes are realized in **dcf77_decoder_ne.vhd**; this VHDL-model uses your designed components and two additional components designed by our group. Configuration of the new board is only possible with the new top level circuit. Layout as well as post layout simulation must be done with **dcf77_decoder_ne.vhd**

Start Xilinx tool *ISE* by the command

**ise**

**Please follow instructions in appendix 1.**

**After** setting up a project according to appendix 1 start synthesis (Menu: Synthesize -> XST in processes window of XST)

Synthesis run will produce a lot of warnings; most of them can be ignored. In case of error messages check first, whether project setup was correctly done according to appendix 1; e.g. it was forgotten to add all required files to the project. Otherwise typically the rules for synthesizable VHDL descriptions were not fulfilled.

In log file of synthesis run (dcf77_decoder_ne.syr) you find in section „HDL Synthesis“ for all modules of the circuit the registers and operators as well as additional logic, e.g. multiplexers.

**Compare for the components *vorverarbeitung* and *date_time_register* your assumptions in the tables above with the results of XST.**

### 5.3 Implementation and Post-Layout Simulation

Subsequently design is realized by using menu „Implement Design“ in the processes window of ISE. This implementation consists of the steps „Translate“, „Map“ and „Place&Route“.

After implementation once again verification by simulation with real delays of the implemented circuit will be performed. A post-layout simulation model must be generated and simulated. For verification use the VHDL test bench of dcf77_testbench_ne.vhd.

**For generating Post-Layout simulation model follow instructions of appendix 2.**

In post-layout simulation it is checked, whether implemented design shows the same functionality as original VHDL model.

For simulation a new library „back“ is generated; in this library the test bench (dcf77_testbench_ne.vhd) as well as the generated post-layout simulation model (dcf77_decoder_ne_timesim.vhd) will be compiled.

**A script (sim_post_dcf77_ne.do) carries out the required steps.**

Start simulator

**vsim**

Execute in Modelsim the script for post-layout simulation:

**Tools -> Execute Macro -> dofiles -> sim_post_dcf77_ne.do**

Script contains the following commands:

```
vlib back
vmap back ./back
vcom -work back ./dcf77_ne/netgen/par/dcf77_decoder_ne_timesim.vhd
vcom -work back ./src/dcf77_testbench_ne.vhd
vsim -sdftyp /dut=./dcf77_ne/netgen/par/dcf77_decoder_ne_timesim.sdf back.dcf77_testbench_ne
do ./dofiles/wave_post_ne.do
run 75 sec
```

The script creates library back, compiles all requires sources, starts simulation including wave window. You must check, whether simulator shows the following message after loading of design:

```
# ** Note: (vsim-3587) SDF Backannotation Succesfully Completed.
#    Time: 0 ps Iteration: 0 Region: /dcf77_testbench_ne File: ./src/dcf77_testbench_ne.vhd
```

Simulation of the post-layout simulation model (detailed structural VHDL-Model with timing information) takes much longer simulation time than original VHDL-model on register transfer level. Thus simulation is carried out only for a time frame of 75 seconds. After this time first DCF77 time frame will be received.

**Check carefully values of signal display as in part 1 of the task.**

Naturally you can perform complete simulation with the command „run –all“.

### 5.4 Validation on hardware board

After successful post-layout simulation configuration data for FPGA must be generated.

Use menu „Generate Programming File“ in Processes window of ISE. The required configuration „bit-File“ will be located after successful execution in the project directory.

Laboratory board is not connected to your terminal. **Thus you will carry out configuration together with your supervisor during the lab hours**.

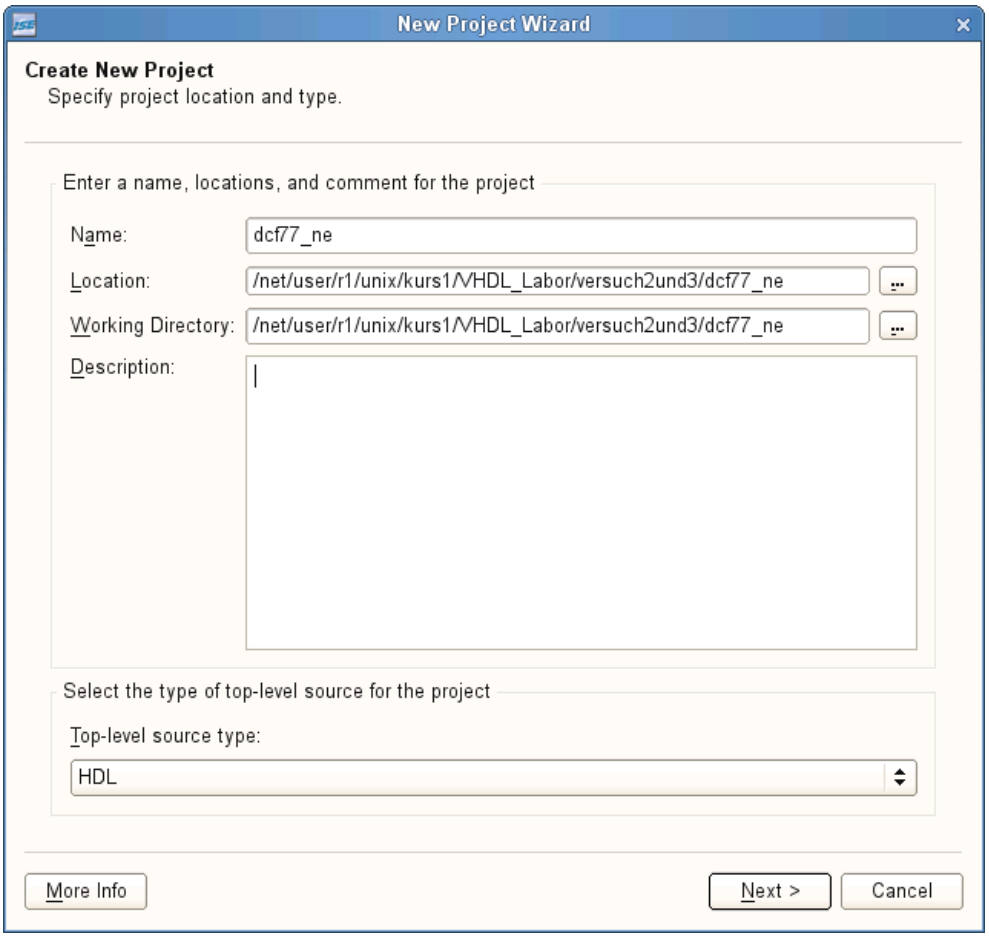
## Appendix 1:
## Create a project for dcf77 with ISE

Before you start with implementation (synthesis and layout of Xilinx-FPGA) check that compilation for simulation is errorless and verification shows the expected results.

All processing steps for configuration of FPGA can be done within ISE. First you must create a new project. Follow the described instructions; especially chosen device and package must be identical to the **XILINX device on board.**

You can control the processing steps (e.g. synthesis, translate, mapping as well place and route) with a lot of options. For the given task it is sufficient to use the given default options.

Start design environment (ISE) by the command **"ise"** in a command shell. After starting use menu **"file > new Project"** to create a new project. Window with procect wizard will come up. Choose location of project directory and identifier (dcf77_ne). Top level source type must be HDL; i.e. a VHDL-code. Identifier for the design will be „dcf77_ne". This implies that a directory with this identifier will be created. All generated files will be located here.

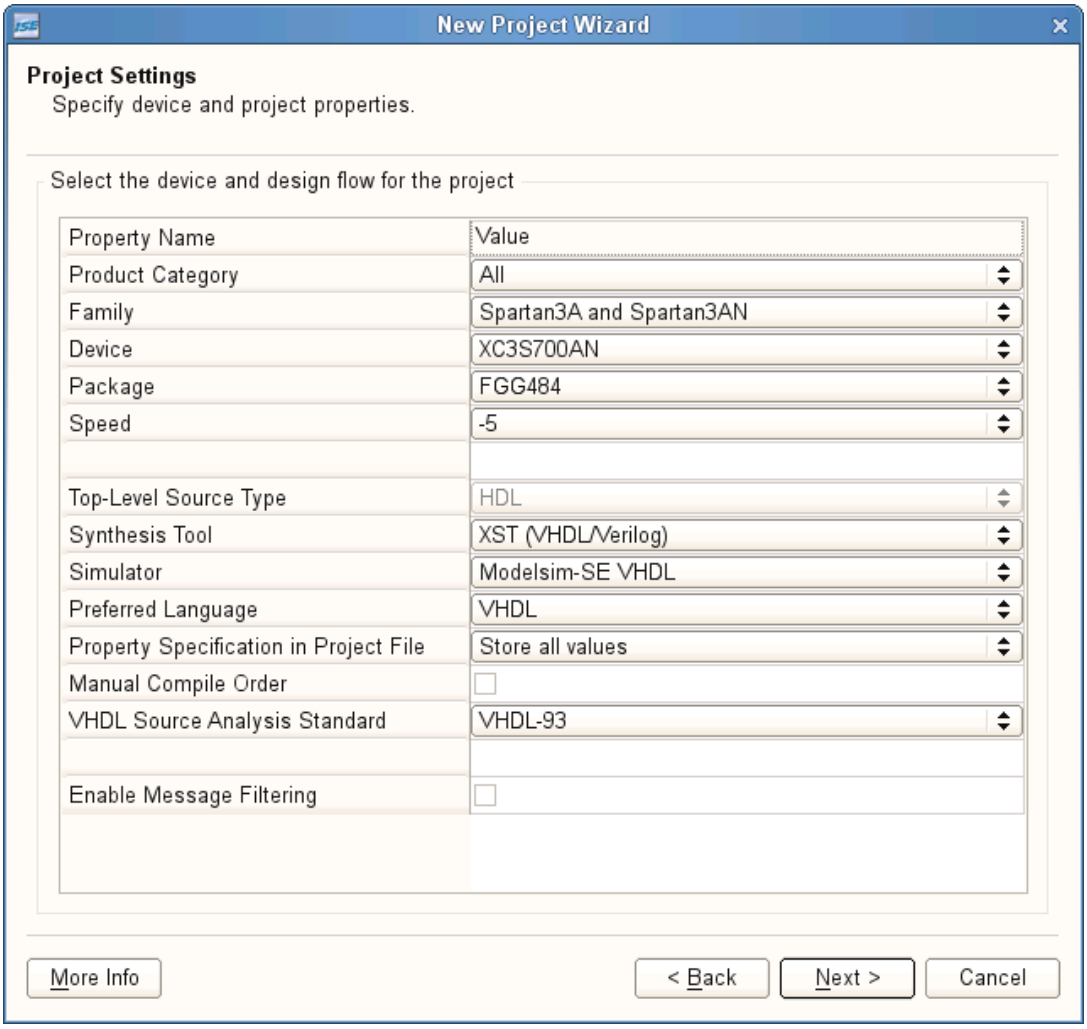



Switch with the „..."-Box to the directory of the task, confirm with OK and write manually the project **name dcf77_ne**. A directory dcf77_ne will be automatically created, where all results of this project will be located.

Choose device family, device, package and speed **correctly** for your project.

**Please check exactly, that your entering coincides with figure below.**

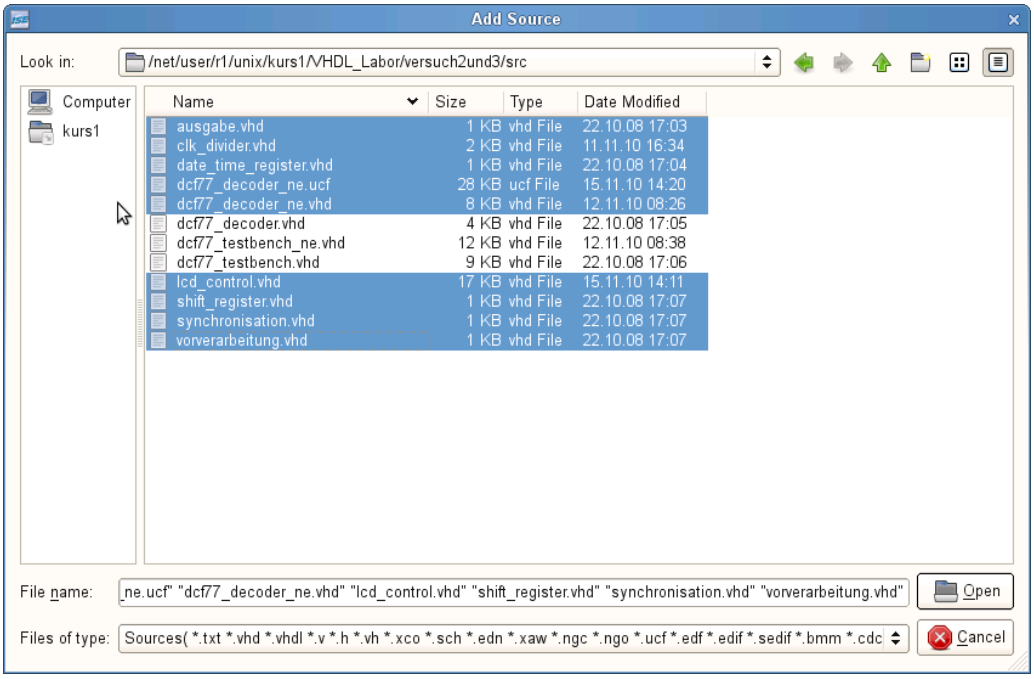
**After this** go ahead with Box „Next >“.



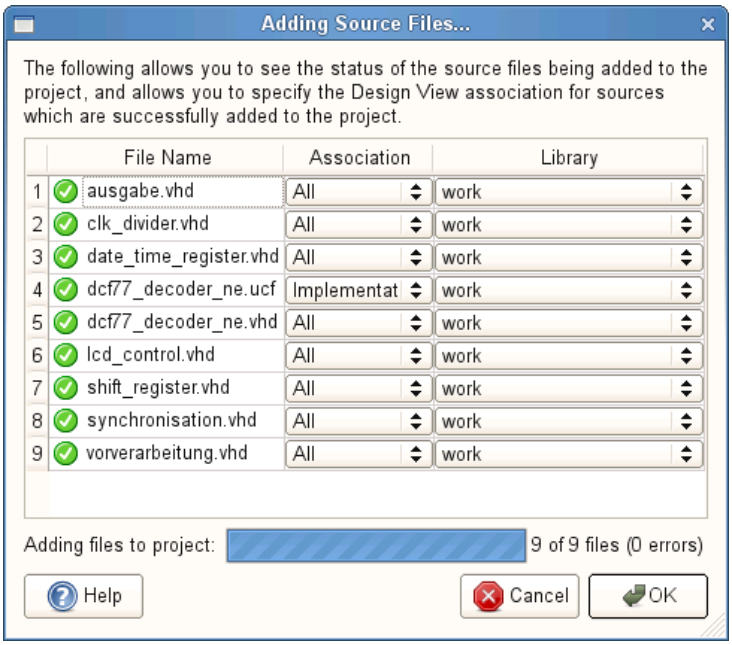

After pressing <Next> button you will see a summary of project settings; you must confirm with button <Finish>. Now you have created a project.

Now all required source codes must be added to the project. Choose according directory (../src) und add as shown in figure the required files (for selecting several files press <STRG> key and select files with the mouse). Confirm the correct selection with <Open> button. If you have forgotten a file you can add it later as just shown.

Keep in mind, that figure shows another file location as in your task.

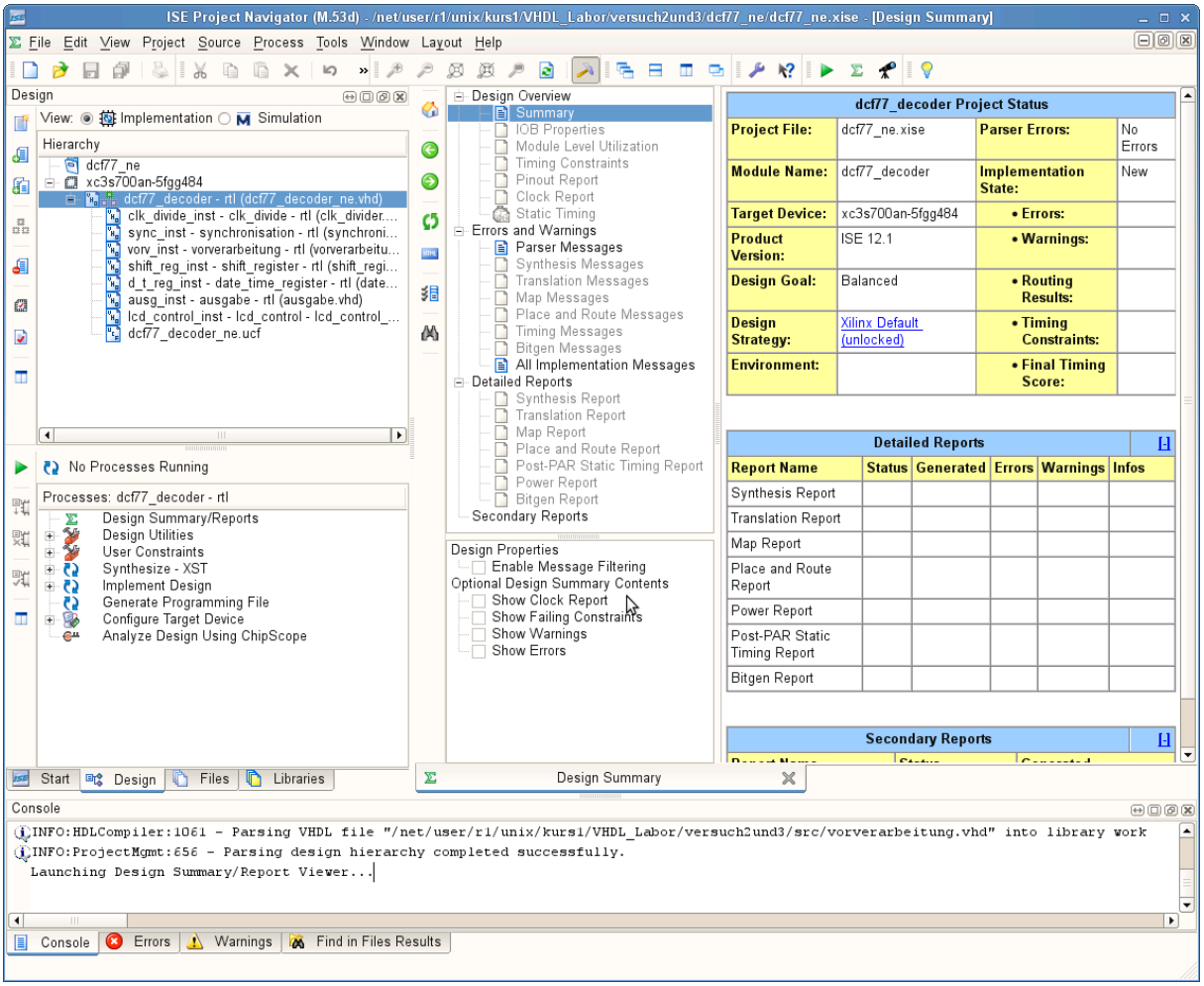



After <Open> you will see following window




**Take care that file with extension** ucf (**U**ser **C**onstraint **F**ile) **is present**. This file does not describe functionality as a VHDL source code, but important constraints for layout of the XILINX chip. Without these constraints, e.g. location of I/O ports, circuit may be correct in simulation but not on board.

Settings for project are now finished; you will see following window:




Top left in hierarchy window the top-level module should be highlighted (**dcf77_decoder_ne**). In window “Processes” required steps for layout can be started.

Clicking on “+” sign contiguous to menu entry shows further details.

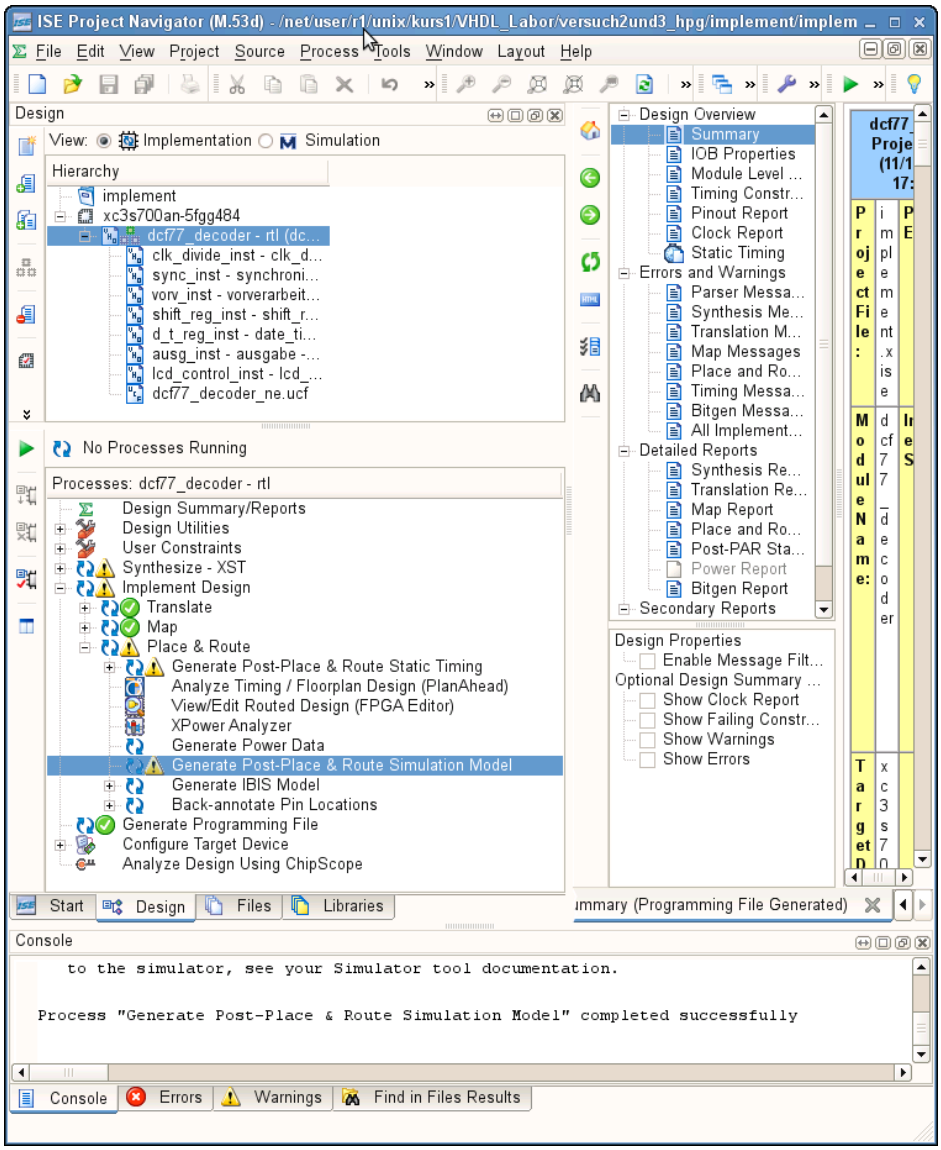
For starting actions press right mouse button on requested menu item; using sub menu entry “run“ execution will be started.

## Appendix 2:
## Generation of post-layout simulation model

After successful synthesis and layout, post-layout simulation model will be generated by menu in process window

Generate Post-Place & Route Simulation Model

as shown in figure below.




For post-layout simulation following files in directory dcf77_ne/netgen/par will be generated:

dcf77_ne_timesim.vhd

dcf77_ne_timesim.sdf.

File dcf77_ne_timesim.vhd is a pure structural VHDL model, which contains instantiations of XILINX specific elements (CLB, LUT) and their interconnection. File dcf77_ne_timesim.sdf provides detailed timing information for simulator.

Now all required data for post-layout simulation are available.