



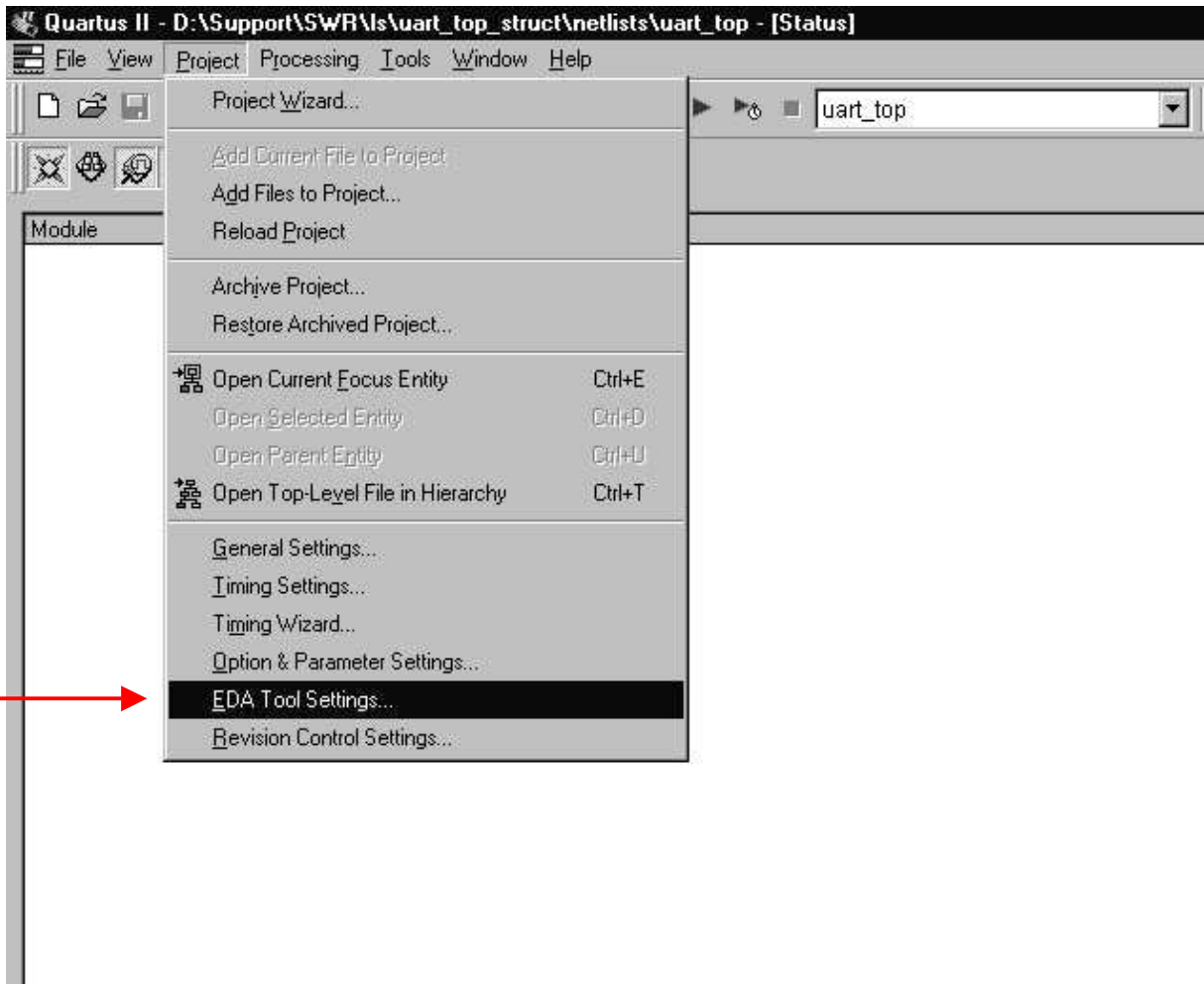
SITAL Technical Notes

Simulating Altera Quartus Verilog Gate-Level file in Modelsim

In order to simulate a Verilog gate-level file that was generated by the Altera Quartus II tool, please do the following steps:

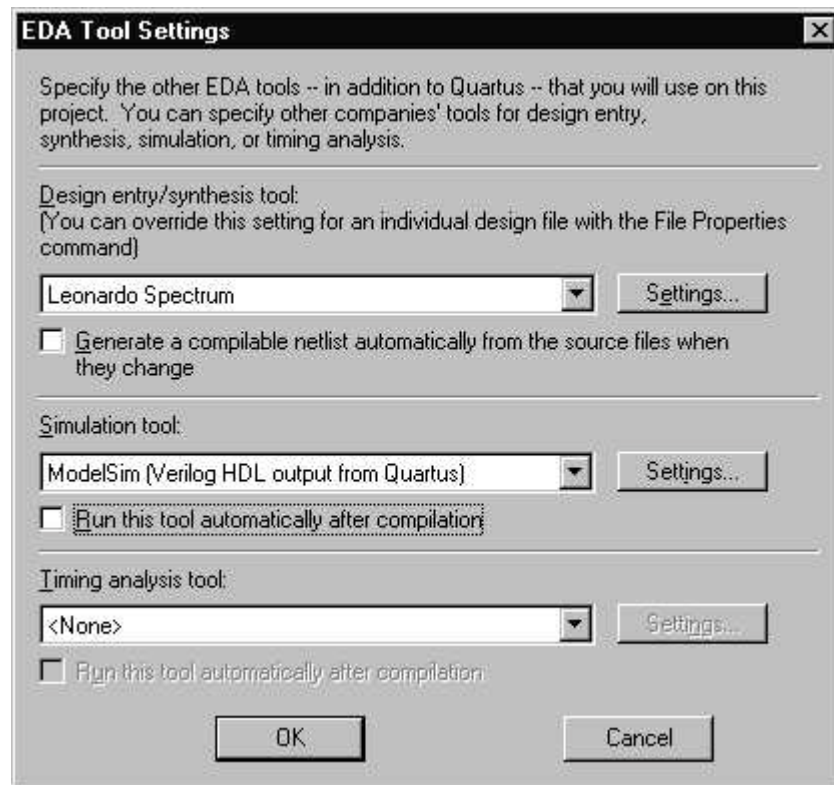
1. Produce the Verilog Output file (.vo) & the Standard Delay Format Output file (.sdo) from the Altera Quartus II .

- a) In the Quartus tool, please create a new project or open an existing project from the file menu.
- b) Choose **EDA Tool Setting** from the **Project** menu .





- c) In the Simulation tool list ,select ‘**Modelsim (Verilog HDL output from Quartus)**’.



- d) Press **OK** .
e) Continue and run your Place & Route process.
f) At the end of the P&R process you will get 2 files: ***.vo** , ***.sdo** .

These files contains the Verilog description and the timing information of your desgin.
The file name of the Standard Delay Format (SDF) Output File (.sdo) is the top-level module name with a ”_v” appended.
(for example, <top-level design name>_v.sdo).

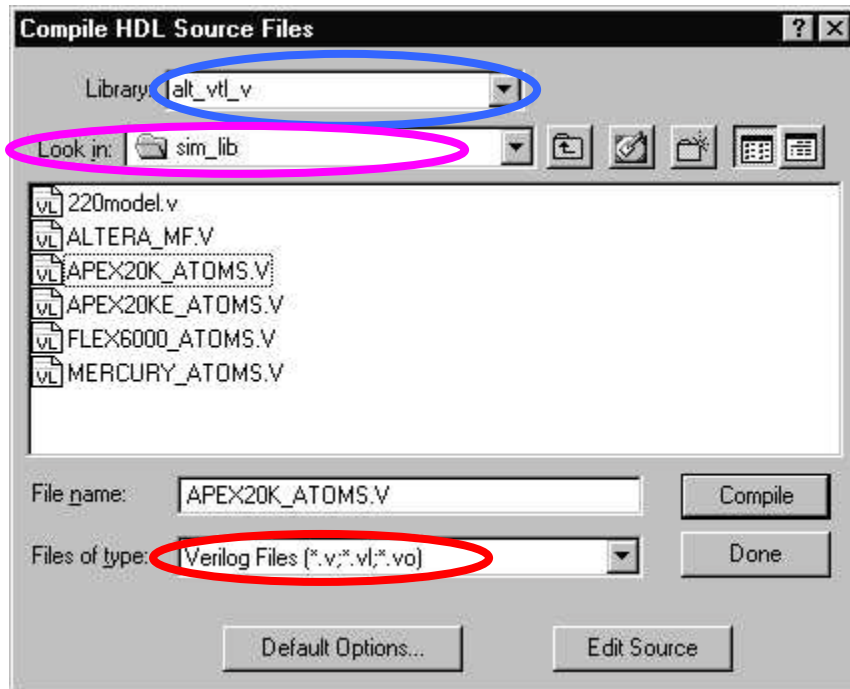
2. Generate the Altera ‘alt_vtl_v’ library.

The Verilog gate-level file (*.vo) is using the Altera primitives library.
Please do the following steps to generate the Altera ‘alt_vtl_v’ library :

- Invoke the ModelSIM Simulation tool.
- Within the Transcript text window do the following:
- Change Directory to the ModelSIM Installation dir, for example C:/modeltech_5.5d.
cd C:/Modeltech_5.5e
- Using windows explorer, change the modelsim.ini file properties (located at **C:/Modeltech_5.5e**) for write permission.
- Generate a new library called **alt_vtl_v** using the Vlib command:
vlib alt_vtl_v



- f) Map the new library using the Vmap command:
vmap alt_vtl_v C:/Modeltech_5.5e/alt_vtl_v
if your ModelSIM Installation directory is different than c:/modeltech_5.5d,
then type:
vmap alt_vtl_v <your_modelsim_installation_directory>/alt_vtl_v
- g) In C:/Modeltech_5.5e change back modelsim.ini properties for read only.
- h) Change Modelsim directory to: ../quartus/eda/sim_lib.
- j) Compile the alt_vtl source files:
Select the '**Design > Compile**' button,
In the '**Compile HDL Source Files**' dialog box set the library to **alt_vtl_v**
using the pull down menu.
Use the '**Look in**' selection to locate the '**alt_vtl_v**' source files.



Compile the sources according to your <device family>.
For example if your <device family> is **APEX20K**, you need to compile the **APEX20K_ATOMS.V** file .
You can compile all the necessary <device family> to this library.

- k) The '**alt_vtl_v**' Library is ready.

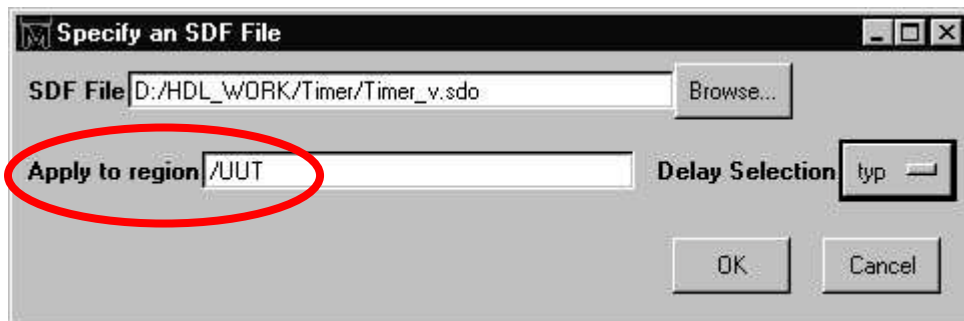


3. Compilation & Simulation of the Gate-Level file.

- a) Copy the *.vo and the *.sdo files into your project directory.
- b) Change ModelSIM directory to your project directory.
- c) Compile the Verilog gate-level file (*.vo) into your work library, using the vcom command (**vcom <top-level design name>.vo**) or from the Compile window.
- d) Select **'Design > Load Design'**.
- e) In the **'SDF'** folder select the **'Add'** option. Specify the full path to the sdo file -> **<top-level design name>_v.sdo**.
- e) Insert the **Instance** name of your module in the **'Apply to region'** field.

For example, in the following verilog code UUT is the Instance name:

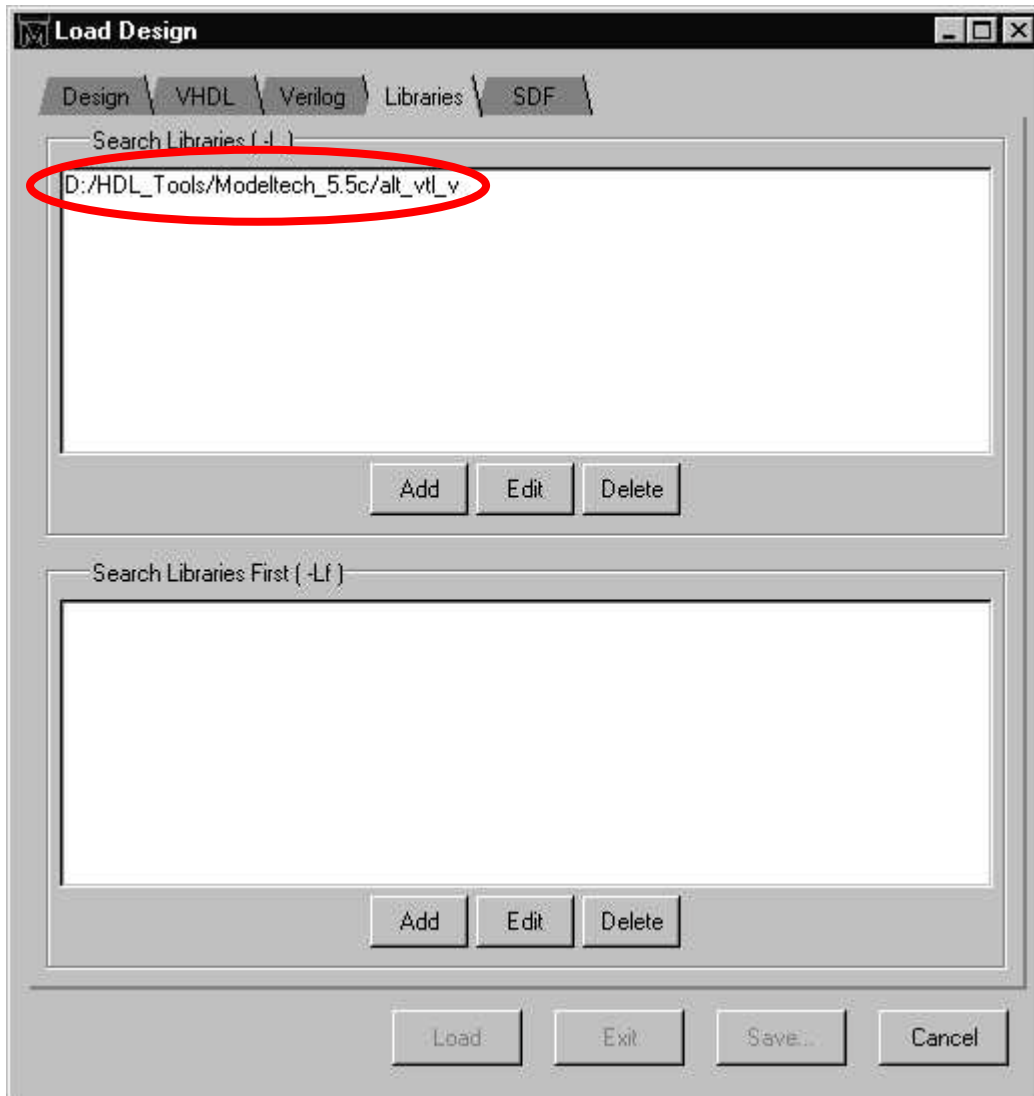
```
Timer UUT(.clk      (clk),  
          .rst      (rst),  
          .data_in  (data_in),  
          .sin      (sin),  
          .data_out (data_out),  
          .sout     (sout)    );
```



- f) Select **OK**.

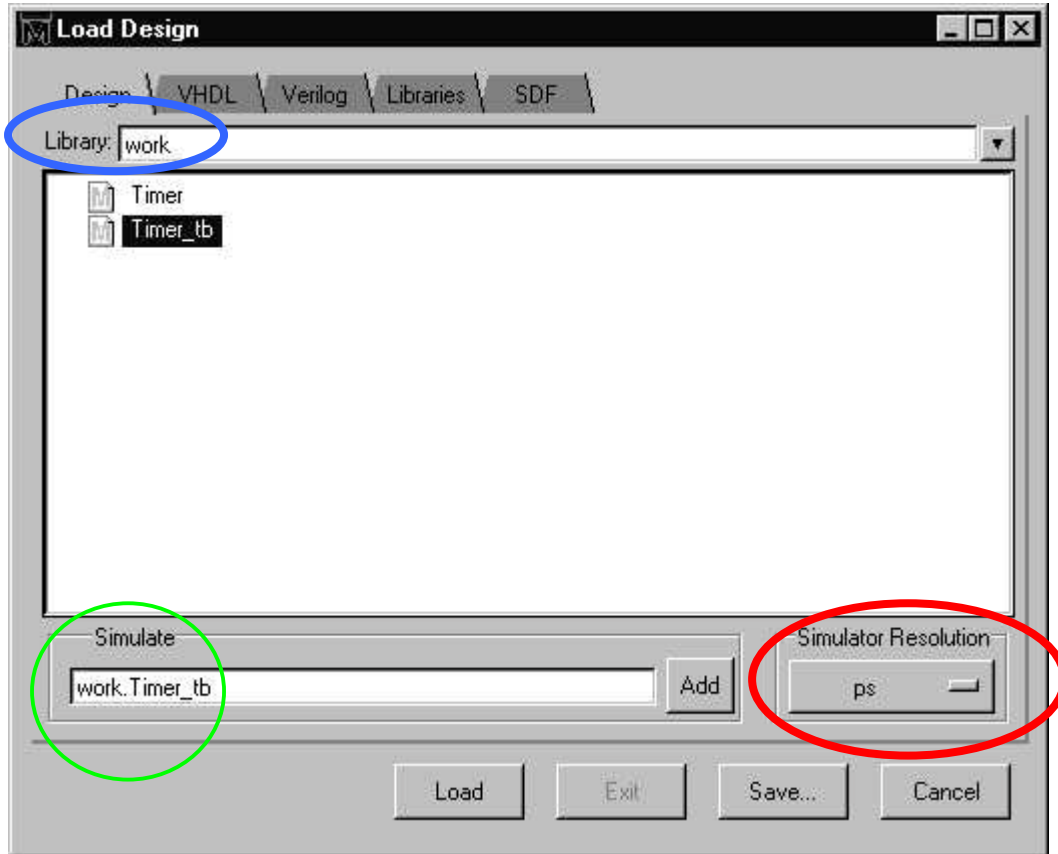


- g) In the '**Libraries**' folder add to the Search Libraries the full path of the alt_vtl_v library that was generated in section 2e.





- h) In the 'Design' folder, Change the 'Simulator Resolution' to 'ps'.
- i) Set the library field to **work** and select your Test-Bench module, click the 'Add' button.



- j) Select the 'Load' button to perform a gate-level timing simulation.

You may also load your design using a text command instead of using the GUI as described in sections 3 d-j by typing the following:

```
vsim -L alt_vtl_v -sdfmax UUT=D:/HDL_WORK/Timer/Timer_v.sdo work.Timer_tb
```