Using Mentor ModelSim Simulator with Lattice iCEcube2

November 24, 2020 (1.7) Application Note AN006

Summary

The Lattice iCEcube2 development software provides a complete FPGA implementation environment for today's FPGA designers. This application note details the basic design and simulation flow using Mentor® ModelSim® simulator.

Introduction

The sample design used in this application note is a simple 4-bit binary up-counter with an associated testbench. The counter design, counter.vhd, is presented first:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity counter is port (
     clk   : in   std_logic;
     reset : in   std_logic;
     count : out  std_logic_vector (3 downto 0));
end counter;

architecture behavioral of counter is
     signal q : std_logic_vector (3 downto 0);
begin

process(clk, reset)
begin
     if(reset = '1') then
          q <= (others=>'0');
     elsif(clk'event and clk = '1') then
          q <= q + 1;
     end if;
end process;

count <= q;

end behavioral;
```

The testbench design, count_tb.vhd, is presented next:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity count_tb is
end count_tb;
```

```
architecture testbench_arch of count_tb is

     component counter
     port (
          clk   : in std_logic;
          reset : in std_logic;
          count : out std_logic_vector (3 downto 0));
     end component;

     signal clk : std_logic := '0';
     signal reset : std_logic := '0';
     signal count : std_logic_vector (3 downto 0) := "0000";

     constant period : time := 100 ns;
     constant duty_cycle : real := 0.5;
     constant offset : time := 100 ns;
begin

     uut : counter
     port map (
          clk => clk,
          reset => reset,
          count => count);

     process -- clock generation
     begin
     wait for offset;
     clock_loop : loop
          clk <= '0';
          wait for (period - (period * duty_cycle));
          clk <= '1';
          wait for (period * duty_cycle);
     end loop clock_loop;
     end process;

     process -- reset generation
     begin
          reset <= '0';
          -- -------------  Current Time:  0ns
          wait for 100 ns;
          reset <= '1';
          -- -------------  Current Time:  100ns
          wait for 35 ns;
          reset <= '0';
          -- -------------  Current Time:  135ns
          wait for 1865 ns;
          -- -------------  Current Time:  2000ns
     end process;

end testbench_arch;
```

Using Mentor ModelSim Simulator with Lattice iCEcube2

This document explains the following scenarios:

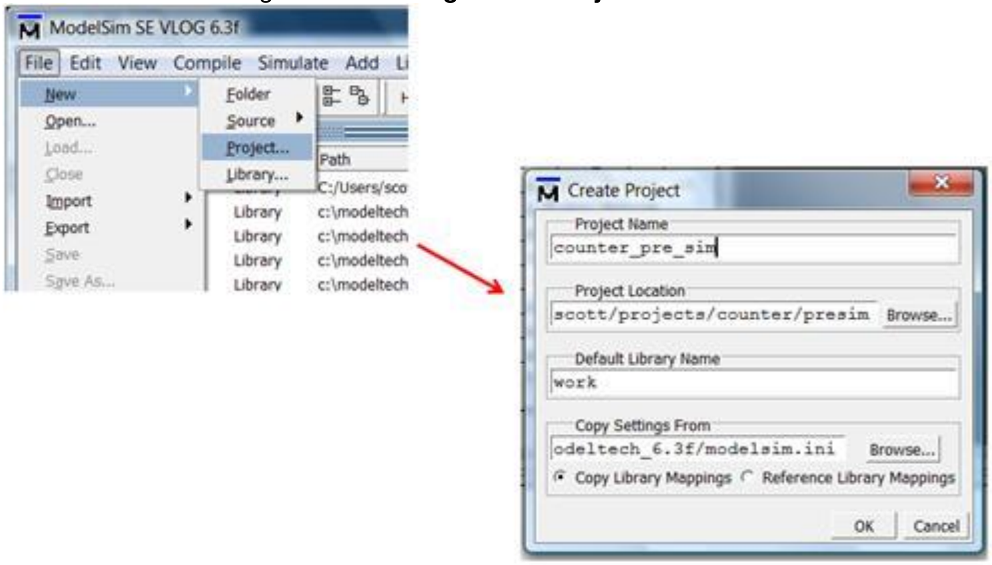
1. Pre-Synthesis Simulation.
2. Post-Synthesis Functional Simulation (Verilog/VHDL).
3. Place-and-Route Functional Simulation (Verilog).
4. Place-and-Route Timing Simulation (Verilog).
5. Place-and-Route Functional Simulation (VHDL).
6. Place-and-Route Timing Simulation (VHDL)

Pre-Synthesis Simulation

This section details the steps required for pre-synthesis simulation.

1. Create a new folder and copy the design (counter.vhd) and testbench (count_tb.vhd).
2. Start ModelSim and create a new project by choosing File→New→Project. Browse to the newly created project folder and enter a project name. Click “OK”.

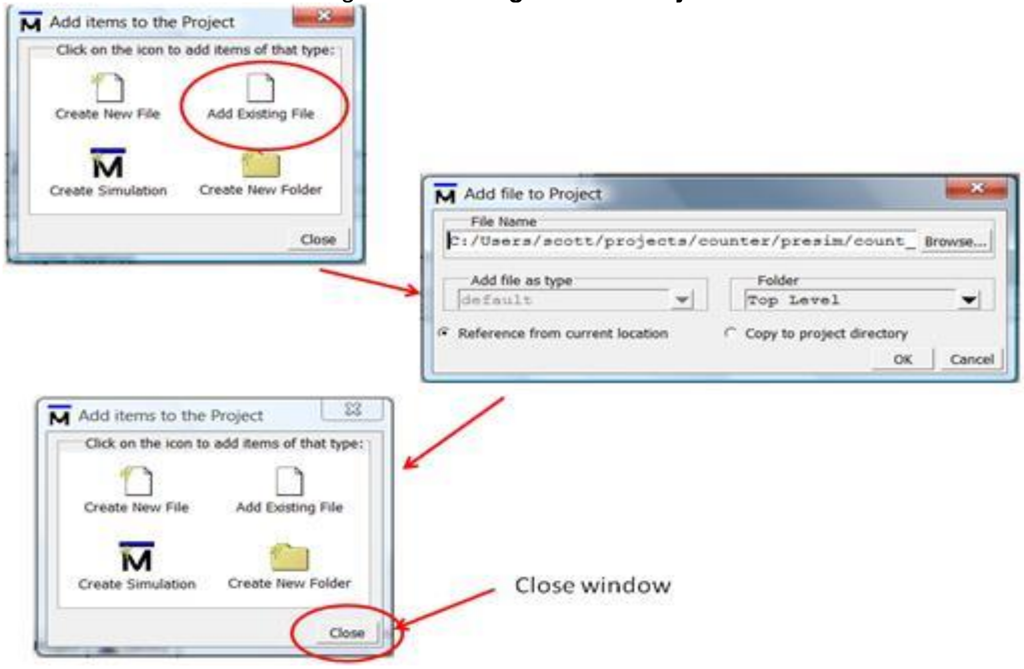
Figure 1: **Creating a New Project in ModelSim**



Project Name: counter_pre_sim
Project Location: scott/projects/counter/presim
Default Library Name: work
Copy Settings From: odeltech_6.3f/modelsim.ini
[✓ Copy Library Mappings] Reference Library Mappings

3. Click “Add Existing Files” and add counter.vhd and count_tb.vhd to the project. Close the “Add items to the Project” window.

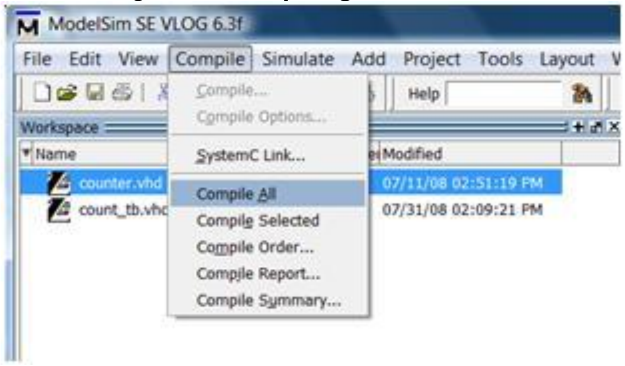
Figure 2: **Adding Files to Project**



File Name: C:/Users/scott/projects/counter/presim/count_
Add file as type: default
Folder: Top Level
[✓ Reference from current location] Copy to project directory

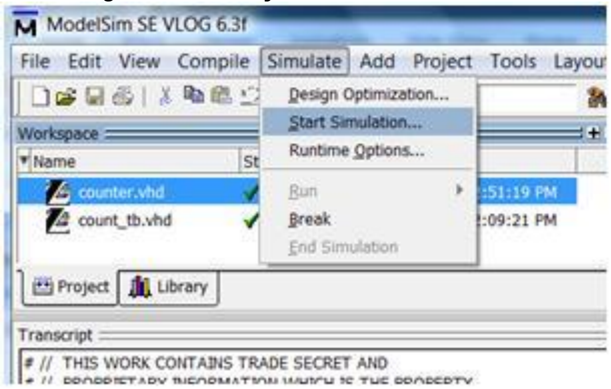
4. Choose Compile→Compile All.

Figure 3: **Compiling HDL Files**



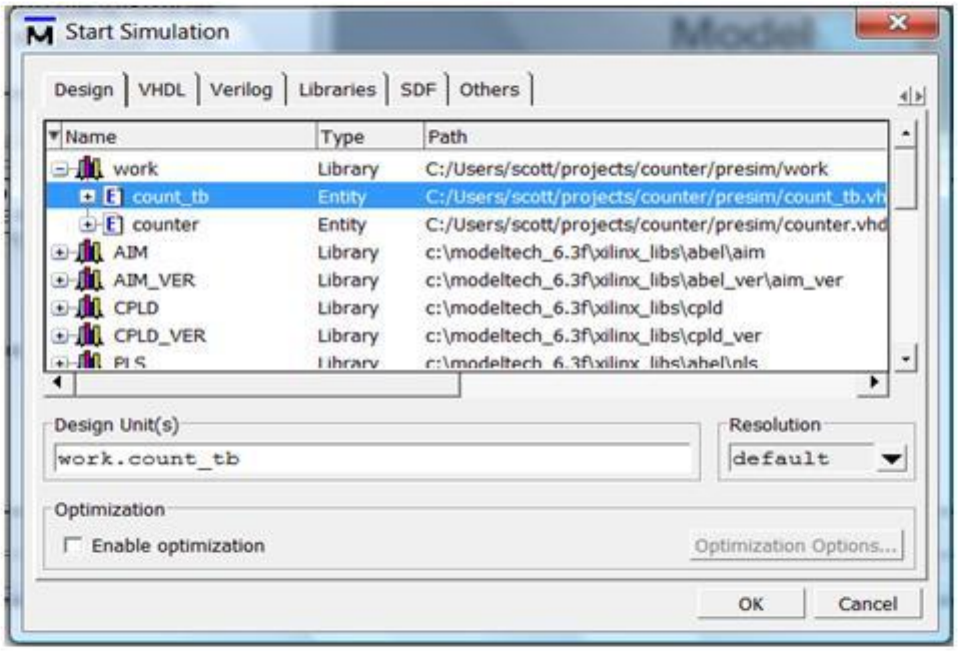
5. Choose Simulate→Start Simulation.

Figure 4: **Pre-Synthesis Simulation**



6. Expand the work folder and highlight count_tb. Clock OK.

Figure 5: **Start Simulation Window**



Name	Type	Path
work	Library	C:/Users/scott/projects/counter/presim/work
count_tb	Entity	C:/Users/scott/projects/counter/presim/count_tb.vh
counter	Entity	C:/Users/scott/projects/counter/presim/counter.vhd
AIM	Library	c:\modeltech_6.3f\xilinx_libs\abel\aim
AIM_VER	Library	c:\modeltech_6.3f\xilinx_libs\abel_ver\aim_ver
CPLD	Library	c:\modeltech_6.3f\xilinx_libs\cpld
CPLD_VER	Library	c:\modeltech_6.3f\xilinx_libs\cpld_ver
PLS	Library	c:\modeltech_6.3f\xilinx_libs\abel\pls

Design Unit(s): work.count_tb

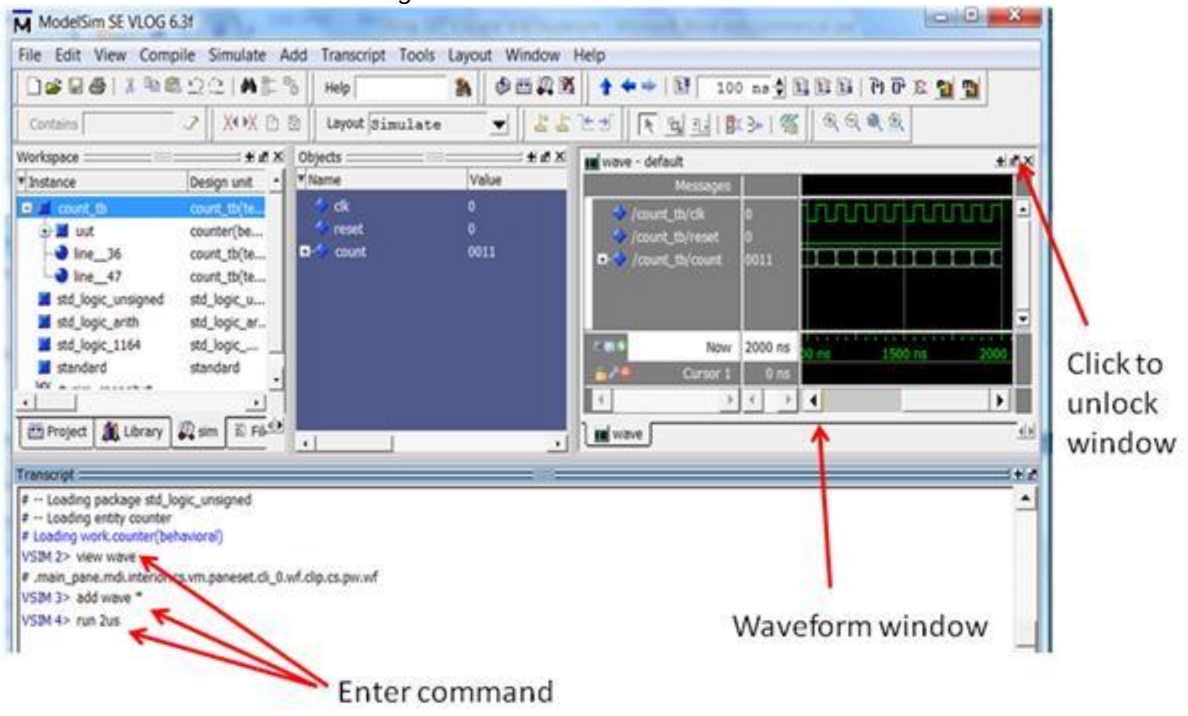
Resolution: default

Optimization

☐ Enable optimization

7. Type the command “view wave” and press the enter key. A new waveform window will show up.
8. Type the command “add wave *” and press the enter key. All I/O signals in the design will show up in the waveform window.
9. Type the command “run 2us” and press enter. The 2uS simulation waveform will show up in the waveform window.
10. Click the “unlock” button to unlock the waveform window for a better view.

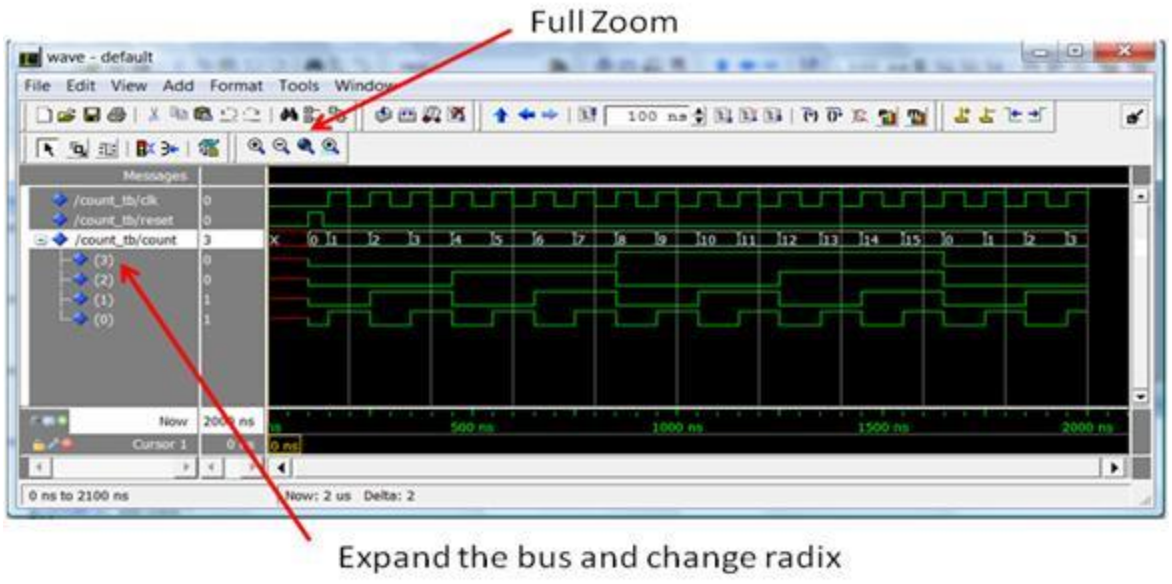
Figure 6: **View Waveform Window**



```
# -- Loading package std_logic_unsigned
# -- Loading entity counter
# Loading work.counter(behavioral)
VSIM 2> view wave
# .main_pane.mdi.interior.cs.vm.paneset.cli_0.wf.clip.cs.pw.wf
VSIM 3> add wave *
VSIM 4> run 2us
```

11. Click “count” to expand this bus. Right-click “count” and choose Radix→unsigned to change the bus number radix. Click “Zoom Full” to have a better view.

Figure 7: **Waveform Window**

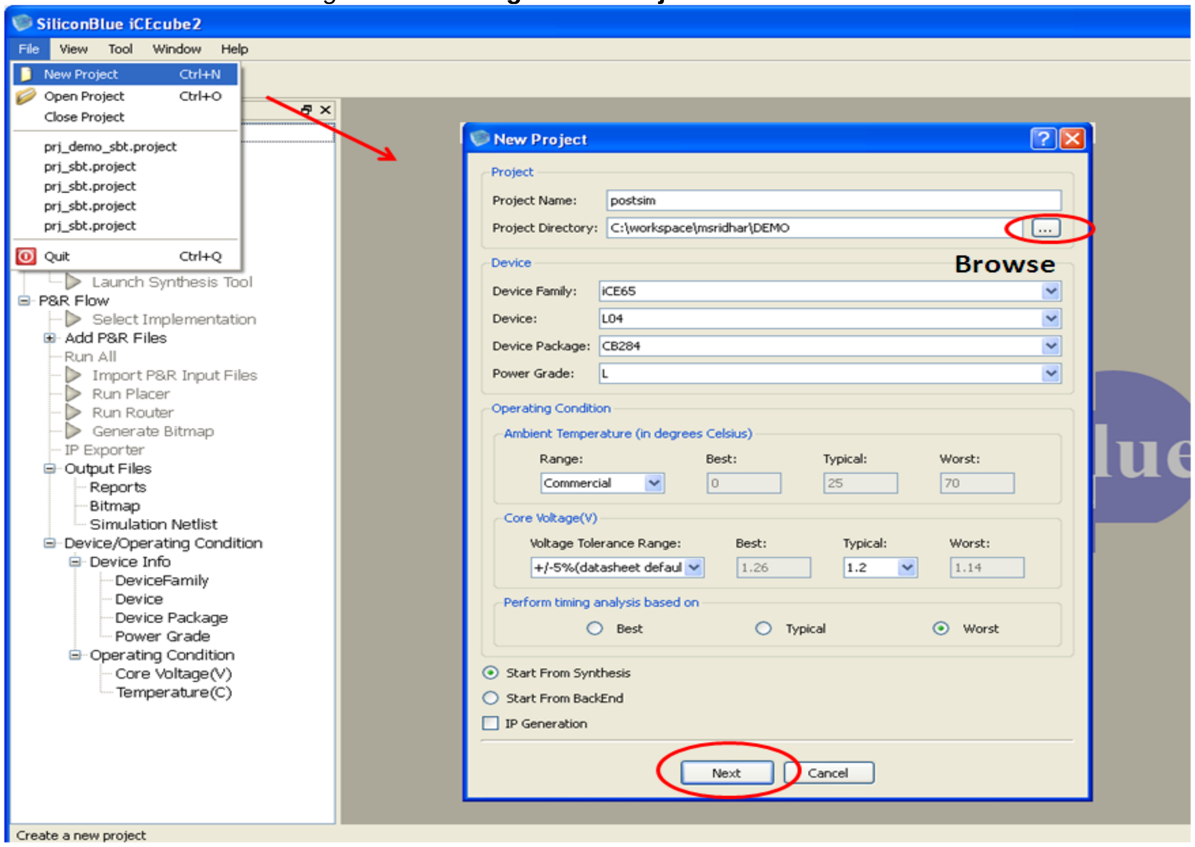


Generating Files Required for Post-Synthesis and Post-Route Simulations

This section details the steps for generating the files that are required for performing post-synthesis and post-route simulations.

1. Start iCEcube2 and create a new project by clicking Project→New Project. Browse to the project folder, enter a new project name, and select an appropriate part. Click “Next”.

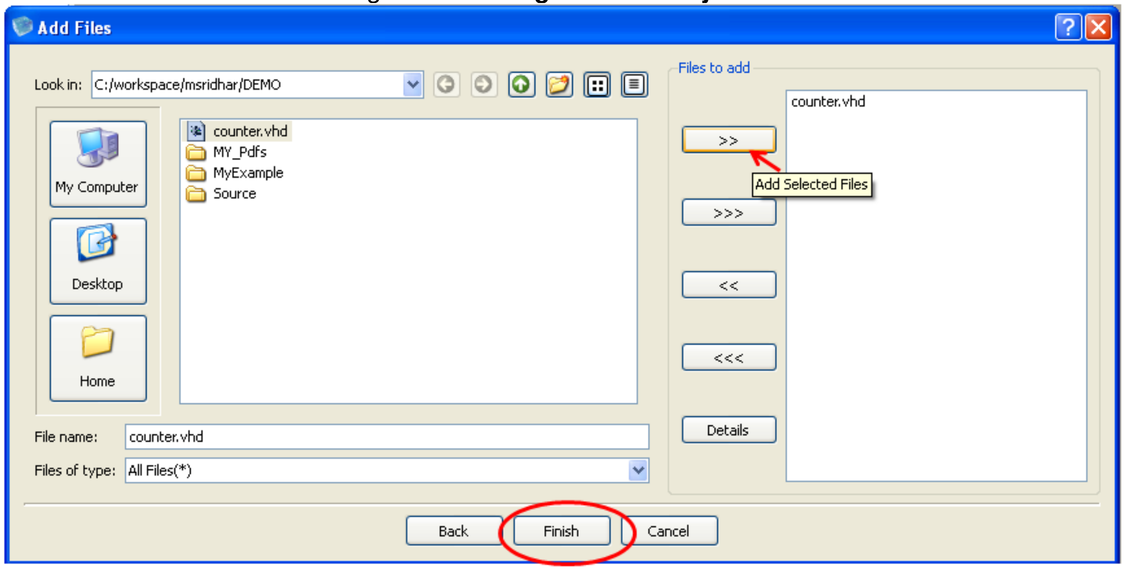
Figure 8: **Creating a New Project in iCEcube2**



Project Name: postsim
Project Directory: C:\workspace\msridhar\DEMO
Device Family: iCE65
Device: L04
Device Package: CB284
Power Grade: L
Range: Commercial Best: 0 Typical: 25 Worst: 70
Voltage Tolerance Range: +/-5%(datasheet defaul Best: 1.26 Typical: 1.2 Worst: 1.14
Best Typical [✓ Worst]
[✓ Start From Synthesis] Start From BackEnd
IP Generation

2. Highlight counter.vhd and click “>>” to add counter.vhd to the right panel. Click on “Finish”.

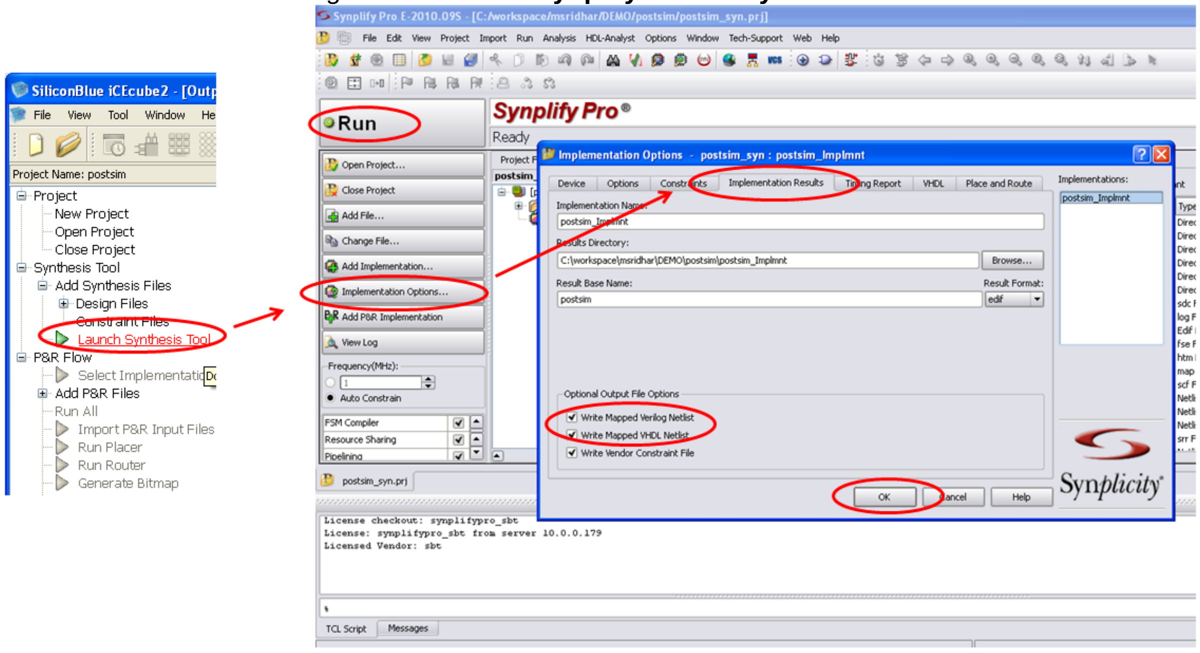
Figure 9: **Adding Files to Project**



Look in: C:/workspace/msridhar/DEMO
counter.vhd
MY_Pdfs
MyExample
Source
File name: counter.vhd
Files of type: All Files(*)
Files to add: counter.vhd

3. Click the “Launch Synthesis Tool” button to launch Synplify Pro® for synthesis. To generate a post-synthesis simulation model in Verilog/VHDL, set the “Write Mapped Verilog Netlist/Write Mapped VHDL Netlist” option in the synthesis tool by going through Implementation Options → Implementation Results, before doing synthesis. Click RUN in Synplify Pro to synthesize the design.

Figure 10: **Launch Synplify Pro for Synthesis**



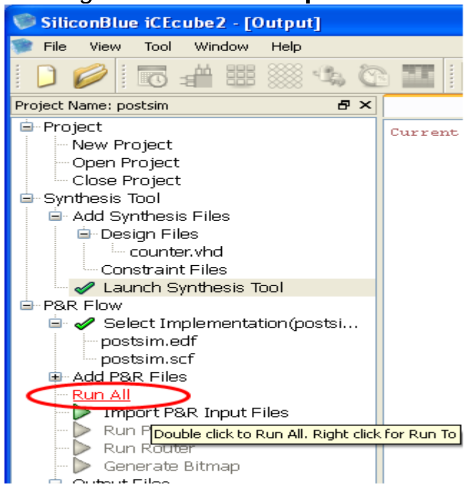
Implementation Name: postsim_Implmnt
Results Directory: C:\workspace\msridhar\DEMO\postsim\postsim_Implmnt
Result Base Name: postsim
Result Format: edif

Optional Output File Options
- [✓] Write Mapped Verilog Netlist
- [✓] Write Mapped VHDL Netlist
- [✓] Write Vendor Constraint File

```
License checkout: synplifypro_sbt
License: synplifypro_sbt from server 10.0.0.179
Licensed Vendor: sbt
```

4. Close Synplify Pro after synthesis. This will bring you back to the iCEcube2 tool window. The synthesis outputs "PRJNAME.edf" and "PRJNAME.scf" are automatically added to "Select Implementation" in "P&R Flow" of iCEcube2. Click "Run All" to run placement, routing and Bitmap Generation.

Figure 11: **Run Complete Flow**



After running the complete flow, the following outputs will be generated.

- PRJNAME.vm, post-synthesis simulation Verilog model
- PRJNAME.vhm, post-synthesis simulation VHDL model
- counter_sbt.v, post-route timing simulation Verilog model
- counter_sbt.vhd, post-route timing simulation VHDL model
- counter_sbt.sdf, for post-route timing Verilog model simulation
- counter_sbt_vital.sdf, for post-route timing VHDL model simulation

Post-synthesis simulation models can be found typically in *PRJNAME/PRJNAME_Implmnt/.* These are generated only if you keep the settings mentioned in Step 3 during synthesis.

The post-synthesis simulation VHDL model PRJNAME.vhm needs to be edited to remove the following two lines for ModelSim Lattice Edition.

```
library synplify;
use synplify.components.all;
```

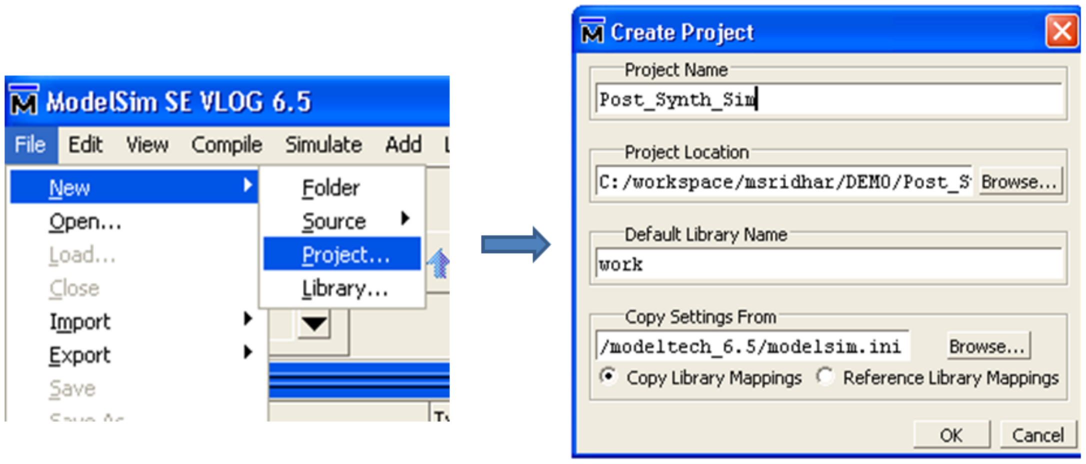
Post-route timing models can be found typically in *PRJNAME/PRJNAME_Implmnt/sbt/outputs/simulation_netlist*

SDF is a standardized representation of timing data commonly used when exchanging timing information between design and simulation tools.

Post-Synthesis Functional Simulation (Verilog/VHDL)

1. Generate the files that are required for a post-synthesis functional simulation model using the steps described in the section “Generating Files Required for Post-Synthesis and Post-Route Simulations”.
2. Open ModelSim. Create new project using File → New → Project. In the New Project window, give a project name and browse to a folder where the project needs to be saved. Click OK.

Figure 12: **Create New Project**



Project Name: Post_Synth_Sim
Project Location: C:/workspace/msridhar/DEMO/Post_S
Default Library Name: work
Copy Settings From: /modeltech_6.5/modelsim.ini
(● Copy Library Mappings ○ Reference Library Mappings)

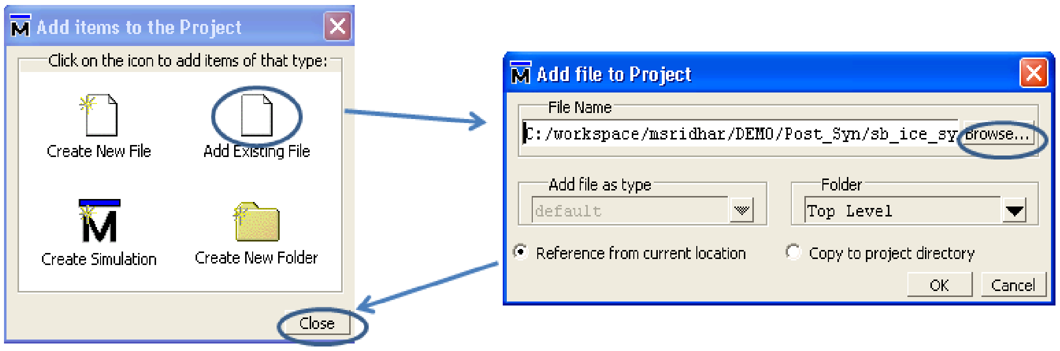
3. Click “Add Existing Files” and add the following files:
   a. PRJNAME.vm, sb_ice_syn.v, counter_tb.vhd for Verilog post-synthesis simulation
   b. PRJNAME.vhm, vcomponent_vital.vhd, sb_ice_lc_vital.vhd, sb_ice_syn_vital.vhd, counter_tb.vhd for VHDL post-synthesis simulation

   sb_ice_syn.v can be found in $INST_DIR/verilog and vcomponent_vital.vhd, sb_ice_lc_vital.vhd, and sb_ice_syn_vital.vhd can be found in $INST_DIR/VHDL.

   If your design contains **PLL,** add **ABIPTBS8.v** and **ABIWTCZ4.v** in $INST_DIR/verilog. To perform post-synthesis simulation on a VHDL design having PLL, you need a mixed-language simulator, since the PLL model (ABIPTBS8.v) is available only in Verilog format. If the design contains **Hardened IP** primitives, add the encrypted Verilog simulation library **sb_ice_ipenc_modelsim.v,** available in $INST_DIR/Verilog.

   Close the “Add items to project” window once all the files are added.

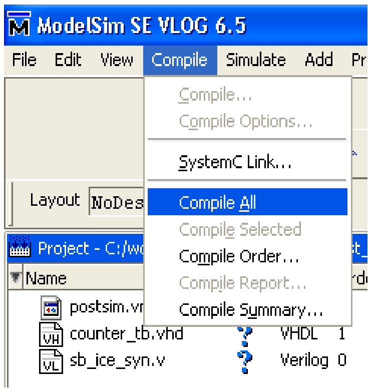
Figure 13: **Add Files to Project**



File Name: C:/workspace/msridhar/DEMO/Post_Syn/sb_ice_sy
Add file as type: default
Folder: Top Level
(● Reference from current location) ○ Copy to project directory

4. Choose Compile → Compile All.
   For VHDL post-synthesis simulation, vcomponent_vital.vhd should be compiled first, then sb_ice_syn_vital.vhd, sb_ice_lc_vital.vhd, and other files.

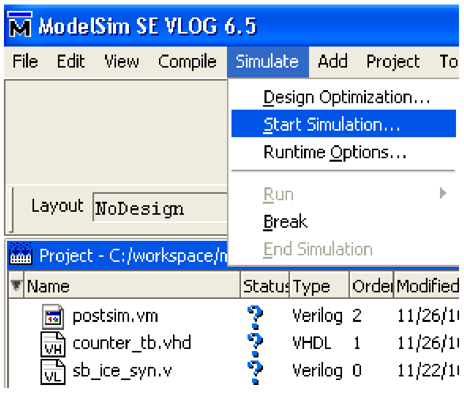
Figure 14: **Compile All**

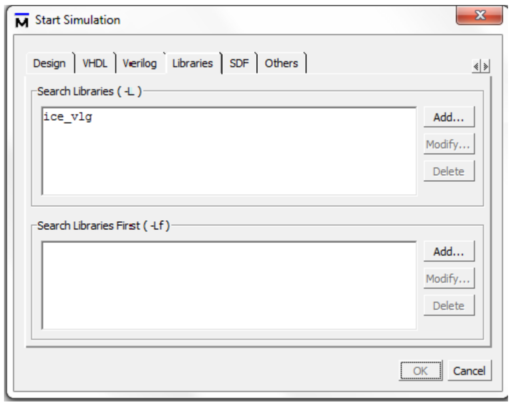


5. Start simulation from Simulate → Start Simulation.

6. In the Simulation window, select the Libraries tab and add the **ice_vlg** resource library for Verilog simulation or **ice** resource library for VHDL simulation in the search libraries path.

Figure 15: **Start Simulation**

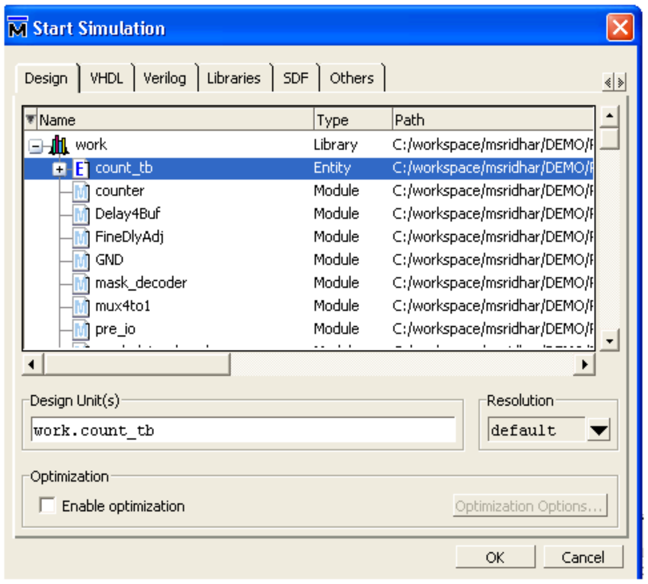




ice_vlg

7. Expand the work folder in the "Start Simulation" window and highlight "counter_tb". Click OK.

Figure 16: **Run Simulation**



Name	Type	Path
work	Library	C:/workspace/msridhar/DEMO/
count_tb	Entity	C:/workspace/msridhar/DEMO/
counter	Module	C:/workspace/msridhar/DEMO/
Delay4Buf	Module	C:/workspace/msridhar/DEMO/
FineDlyAdj	Module	C:/workspace/msridhar/DEMO/
GND	Module	C:/workspace/msridhar/DEMO/
mask_decoder	Module	C:/workspace/msridhar/DEMO/
mux4to1	Module	C:/workspace/msridhar/DEMO/
pre_io	Module	C:/workspace/msridhar/DEMO/

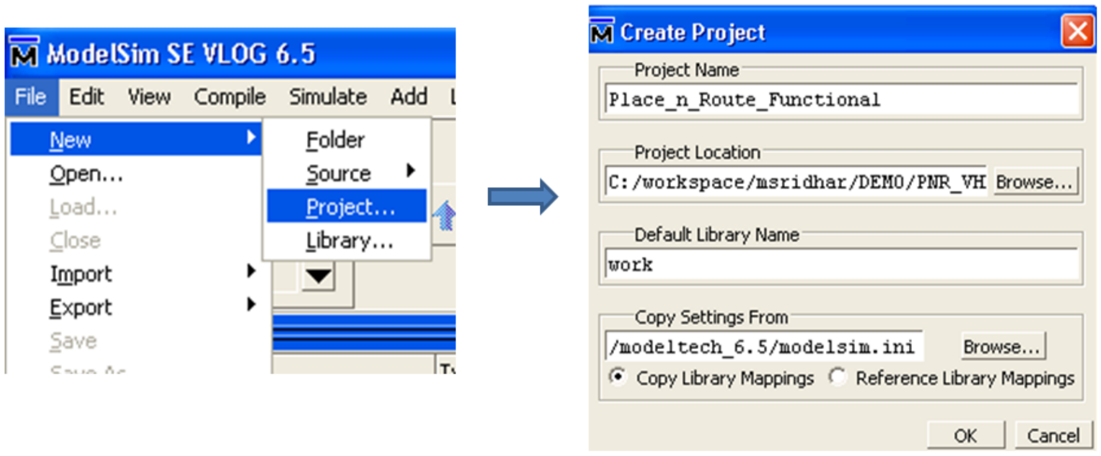
Design Unit(s): work.count_tb
Resolution: default

8. This will open the simulation window. Add the signals wave window and perform simulation as explained in the "Pre-Synthesis Simulation" section from point 7 onwards.

Place-and-Route Functional Simulation (Verilog)

1. Generate the files that are required for post-route functional simulation model using the steps described in the section "Generating Files Required for Post-Synthesis and Post-Route Simulations."
2. Open ModelSim. Create new project using File → New → Project. In the New Project window, give a project name and browse to a folder where the project needs to be saved. Click OK.

Figure 17: **Create New Project**



Create Project

Project Name: Place_n_Route_Functional

Project Location: C:/workspace/msridhar/DEMO/PNR_VH Browse...

Default Library Name: work

Copy Settings From: /modeltech_6.5/modelsim.ini Browse...

(●) Copy Library Mappings ( ) Reference Library Mappings

3. Click "Add Existing Files" and add the following files:
   a. counter_sbt.v
   b. counter_tb.vhd

   Add the following Verilog simulation libraries available in **$INST_DIR/verilog**

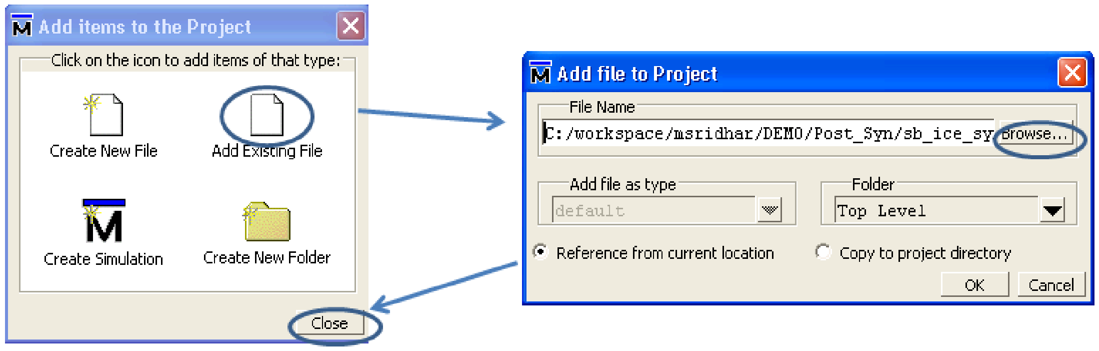
   a. sb_ice_syn.v
   b. sb_ice_lc.v

   If your design contains **PLL,** add **ABIPTBS8.v** & **ABIWTCZ4.v** in $INST_DIR/verilog.

   If the design contains **Hardened IP** primitives, add the encrypted simulation library **sb_ice_ipenc_modelsim.v** available in $INST_DIR/Verilog.

   Close the "Add items to project" window once all the files are added.

Figure 18: **Add Files to Project**



File Name: C:/workspace/msridhar/DEMO/Post_Syn/sb_ice_sy Browse...

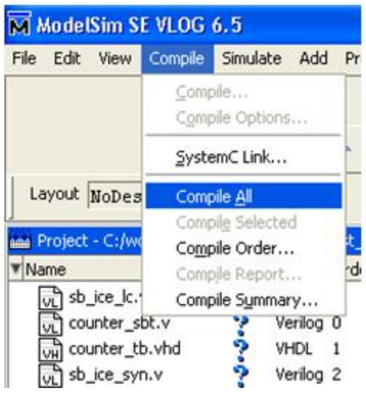
Add file as type: default

Folder: Top Level

(●) Reference from current location ( ) Copy to project directory

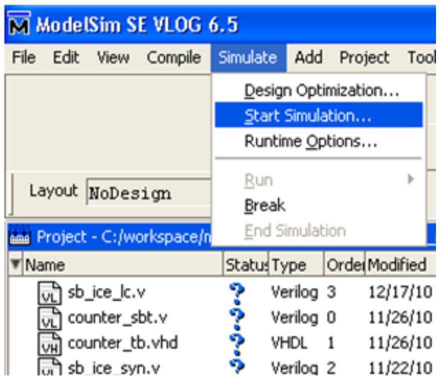
4. Click on Compile → Compile All.

Figure 6: **Compile All**



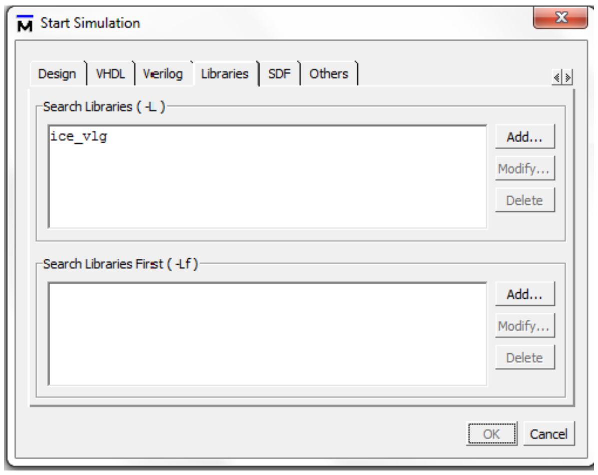
5. Start simulation from Simulate → Start Simulation.

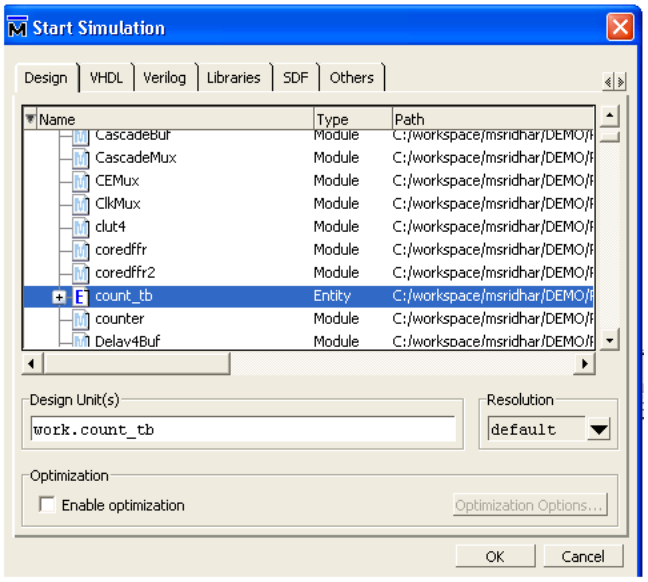
Figure 7: **Start Simulation**


ModelSim SE VLOG 6.5
File Edit View Compile Simulate Add Project Tool
Design Optimization...
Start Simulation...
Runtime Options...
Run
Break
End Simulation
Layout NoDesign
Project - C:/workspace/n
Name Status Type Order Modified
sb_ice_lc.v Verilog 3 12/17/10
counter_sbt.v Verilog 0 11/26/10
counter_tb.vhd VHDL 1 11/26/10
sb_ice_syn.v Verilog 2 11/22/10

6. In the Simulation window, select the Libraries tab and add the **ice_vlg** resource library in the search libraries path.
7. Expand the work folder in the “Start Simulation” window and highlight “counter_tb”. Click OK.

Figure 8: **Run Simulation**





Name	Type	Path
CascadeBuf	Module	C:/workspace/msridhar/DEMO/
CascadeMux	Module	C:/workspace/msridhar/DEMO/
CEMux	Module	C:/workspace/msridhar/DEMO/
ClkMux	Module	C:/workspace/msridhar/DEMO/
clut4	Module	C:/workspace/msridhar/DEMO/
coredffr	Module	C:/workspace/msridhar/DEMO/
coredffr2	Module	C:/workspace/msridhar/DEMO/
count_tb	Entity	C:/workspace/msridhar/DEMO/
counter	Module	C:/workspace/msridhar/DEMO/
Delay4Buf	Module	C:/workspace/msridhar/DEMO/

Design Unit(s): work.count_tb

Resolution: default

Optimization: Enable optimization [ ]   Optimization Options...

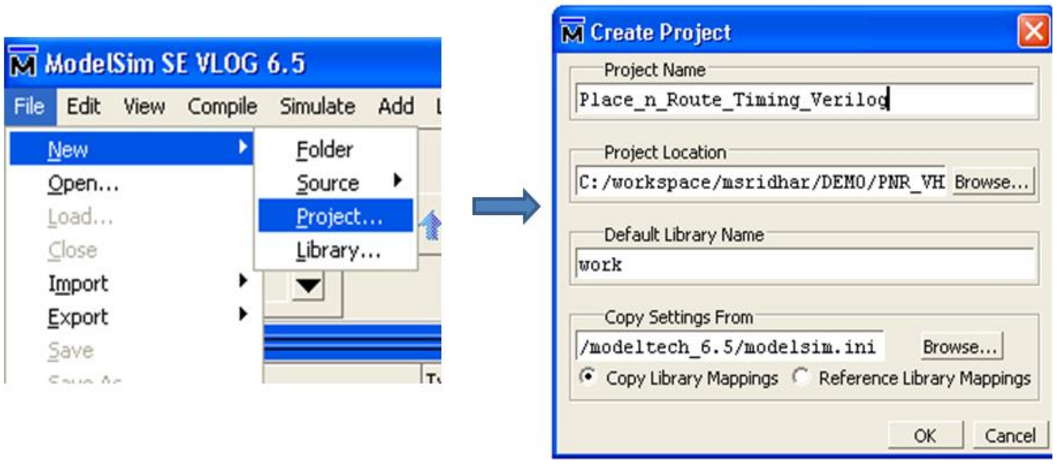
8. This will open the simulation window. Add the signals wave window and perform simulation as explained in the “Pre-Synthesis Simulation” section from point 7 onwards.

Place-and-Route Timing Simulation (Verilog)

1. Generate the files that are required for the post-route timing simulation model using the steps described in the section “Generating Files Required for Post-Synthesis and Post-Route Simulations”.

2. Open ModelSim. Create a new project using File → New → Project. In the New Project window, give a project name and browse to a folder where the project needs to be saved. Click OK.

Figure 9: **Create New Project**



Project Name: Place_n_Route_Timing_Verilog

Project Location: C:/workspace/msridhar/DEMO/PNR_VH

Default Library Name: work

Copy Settings From: /modeltech_6.5/modelsim.ini

[✓ Copy Library Mappings] Reference Library Mappings

3. Click “Add Existing Files” and add the following files:

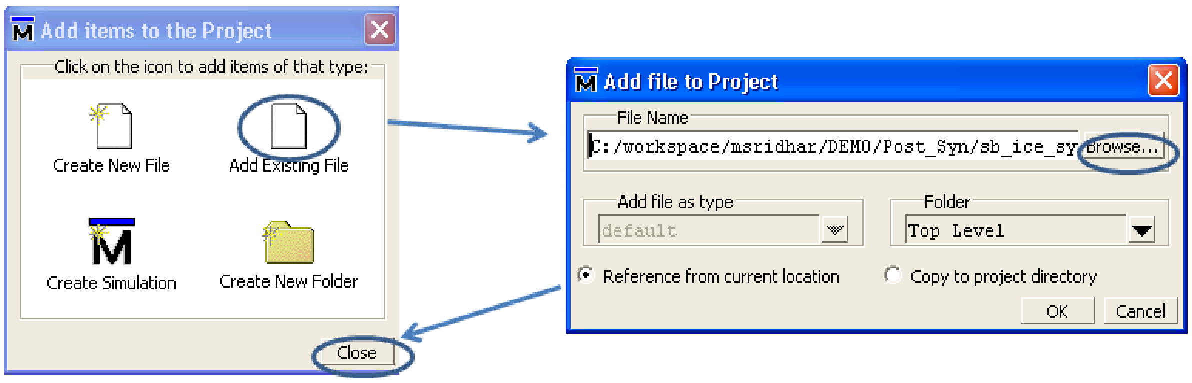
   counter_sbt.v, sb_ice_syn.v, sb_ice_lc.v, counter_tb.vhd for Verilog post-route timing simulation

   The sb_ice_syn.v, sb_ice_lc.v verilog files can be found in $INST_DIR/verilog.

   If your design contains **PLL,** add **ABIPTBS8.v** in $INST_DIR/verilog.
   If the design contains **Hardened IP** primitives, add the encrypted simulation library **sb_ice_ipenc_modelsim.v** available in $INST_DIR/Verilog.

Close the “Add items to Project” window once all the files are added.

Figure 10: **Add Files to Project**



File Name: C:/workspace/msridhar/DEMO/Post_Syn/sb_ice_sy

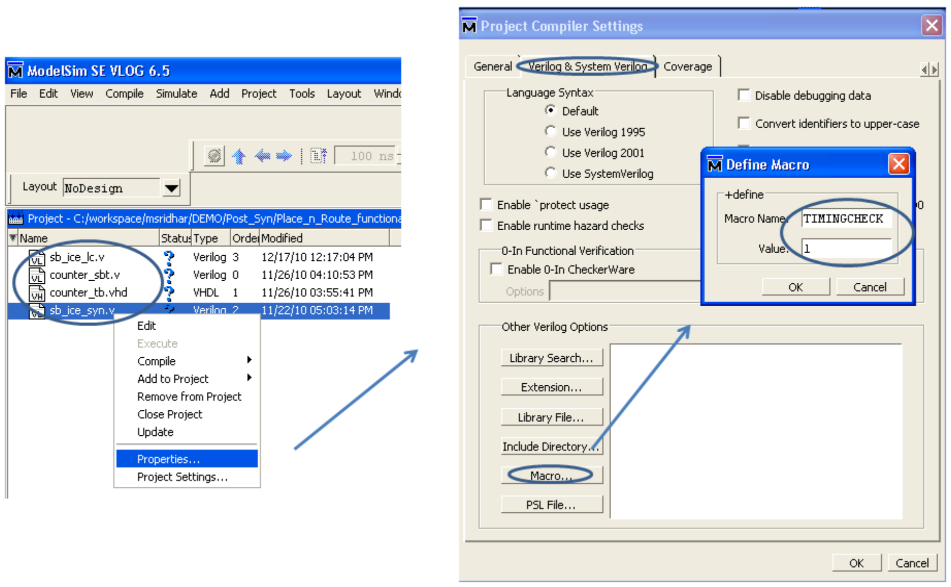
Add file as type: default

Folder: Top Level

[✓ Reference from current location] Copy to project directory

4. In the Project Compiler Settings window, select the “Verilog & System Verilog” tab and click “Macro”. In the Define Macro window, set the macro name to “TIMINGCHECK” and set the value to 1. Click OK.

Figure 11: **Set TIMINGCHECK Macro**



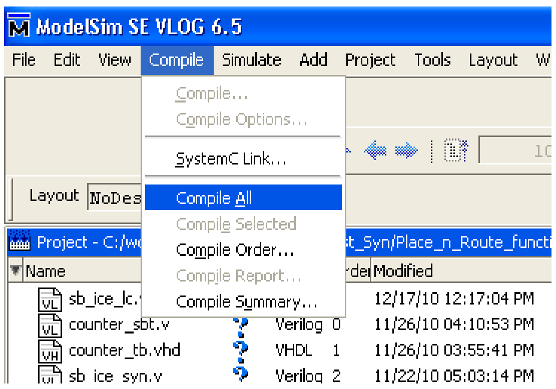
Name	Status	Type	Order	Modified
sb_ice_lc.v	?	Verilog	3	12/17/10 12:17:04 PM
counter_sbt.v	?	Verilog	0	11/26/10 04:10:53 PM
counter_tb.vhd	?	VHDL	1	11/26/10 03:55:41 PM
sb_ice_syn.v	?	Verilog	2	11/22/10 05:03:14 PM

Macro Name: TIMINGCHECK

Value: 1

5. Click on Compile → Compile All.

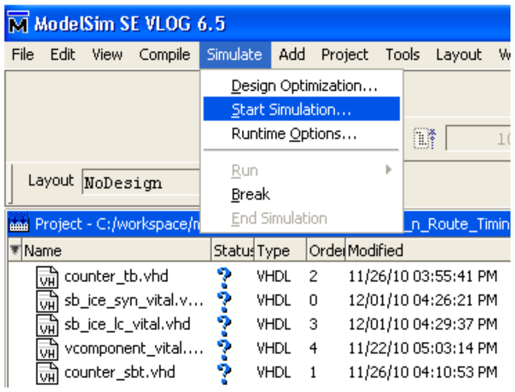
Figure 12: **Compile All**



Name	Status	Type	Order	Modified
sb_ice_lc.v				12/17/10 12:17:04 PM
counter_sbt.v	?	Verilog	0	11/26/10 04:10:53 PM
counter_tb.vhd	?	VHDL	1	11/26/10 03:55:41 PM
sb_ice_syn.v	?	Verilog	2	11/22/10 05:03:14 PM

6. Start simulation using Simulate → Start Simulation

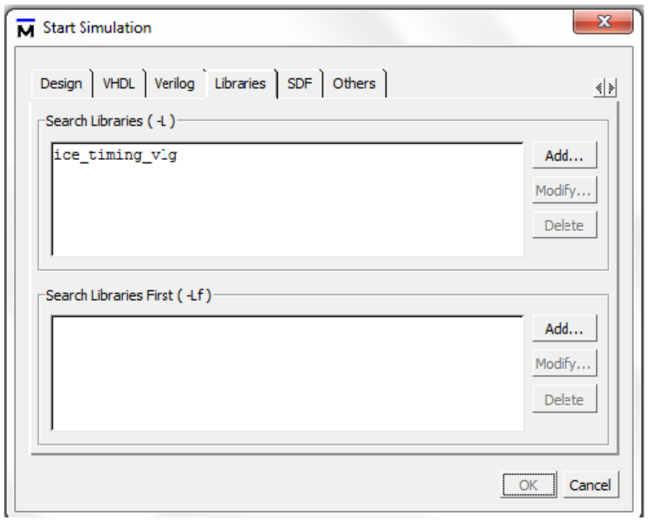
Figure 13: **Start Simulation**



Name	Status	Type	Order	Modified
counter_tb.vhd	?	VHDL	2	11/26/10 03:55:41 PM
sb_ice_syn_vital.v…	?	VHDL	0	12/01/10 04:26:21 PM
sb_ice_lc_vital.vhd	?	VHDL	3	12/01/10 04:29:37 PM
vcomponent_vital.…	?	VHDL	4	11/22/10 05:03:14 PM
counter_sbt.vhd	?	VHDL	1	11/26/10 04:10:53 PM

7. In the Simulation window, select the “Libraries” tab and add the “**ice_timing_vlg**” resource library in the search libraries path.

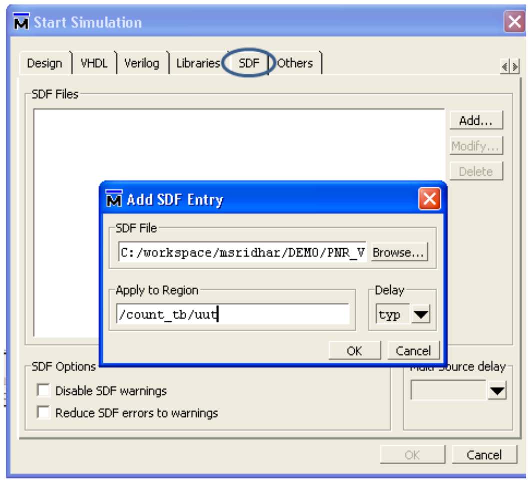
Figure 27: **Add “ice_timing_vlg” resource library to the project**



ice_timing_vlg

8. In the Simulation window, select the SDF tab. Add the SDF file counter_sbt.sdf and type “/count_tb/uut” in “Apply to Region”. Click OK.

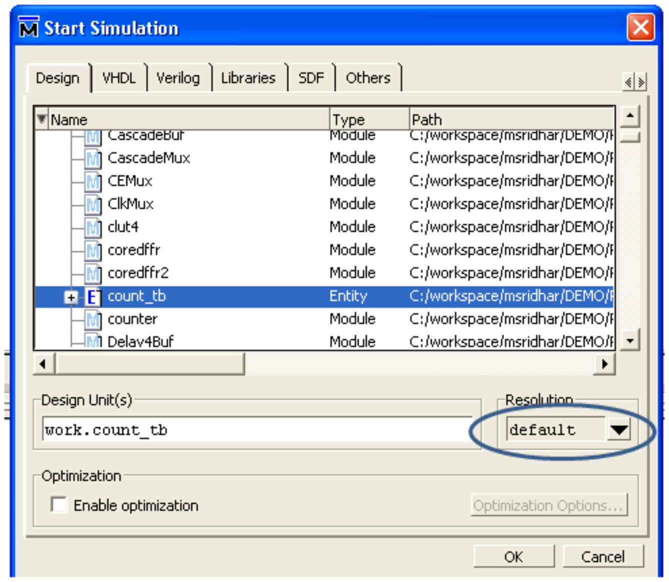
Figure 28: **Add SDF to project**



SDF File: C:/workspace/msridhar/DEMO/PNR_V
Apply to Region: /count_tb/uut
Delay: typ

9. In the Design tab, select "count_tb" under work. Make sure that the "Time Resolution" was set to PS. Click OK.

Figure 29: **Run Simulation**



Name	Type	Path
CascadeBuf	Module	C:/workspace/msridhar/DEMO/
CascadeMux	Module	C:/workspace/msridhar/DEMO/
CEMux	Module	C:/workspace/msridhar/DEMO/
ClkMux	Module	C:/workspace/msridhar/DEMO/
clut4	Module	C:/workspace/msridhar/DEMO/
coredffr	Module	C:/workspace/msridhar/DEMO/
coredffr2	Module	C:/workspace/msridhar/DEMO/
count_tb	Entity	C:/workspace/msridhar/DEMO/
counter	Module	C:/workspace/msridhar/DEMO/
Delay4Buf	Module	C:/workspace/msridhar/DEMO/

Design Unit(s): work.count_tb
Resolution: default

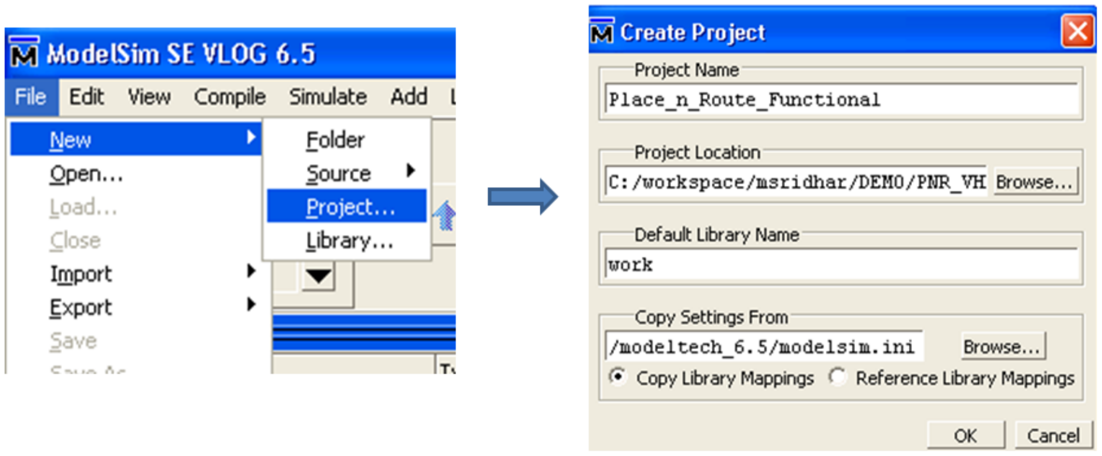
10. This will open the simulation window. Add the signals wave window and perform simulation as explained in the "Pre-Synthesis Simulation" section from point 7 onwards.

Place-and-Route Functional Simulation (VHDL)

1. Generate the files that are required for post-route functional simulation model using the steps described in the section "Generating Files Required for Post-Synthesis and Post-Route Simulations".

2. Create a new project using File → New → Project. In the New Project window, give a project name and browse to a folder where the project needs to be saved. Click OK.

Figure 30: **Create New Project**

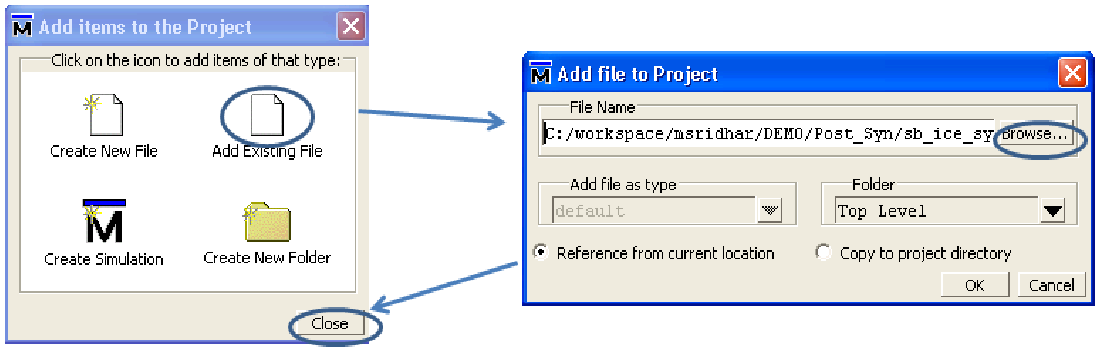


Project Name: Place_n_Route_Functional
Project Location: C:/workspace/msridhar/DEMO/PNR_VH Browse…
Default Library Name: work
Copy Settings From: /modeltech_6.5/modelsim.ini Browse…
[✓ Copy Library Mappings] Reference Library Mappings

3. Click “Add Existing File” and add the following files:
   a. counter_sbt.vhd
   b. counter_tb.vhd

   Close the “Add items to the Project” window once all the files are added.

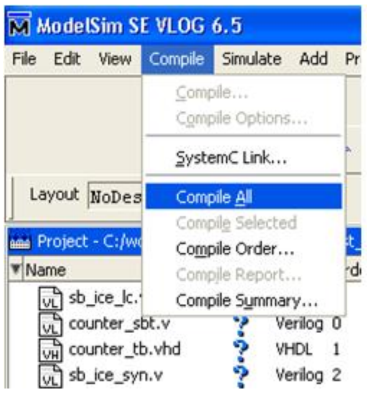
Figure 31: **Add Files to Project**



File Name: C:/workspace/msridhar/DEMO/Post_Syn/sb_ice_sy Browse…
Add file as type: default
Folder: Top Level
[✓ Reference from current location] Copy to project directory

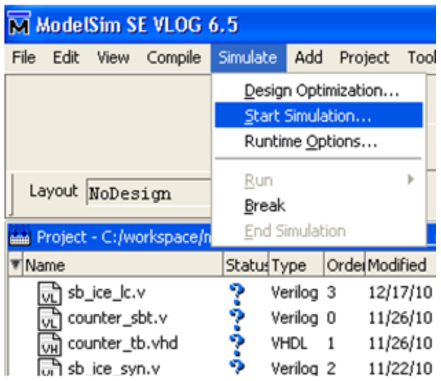
4. Click Compile → Compile All.

Figure 32: **Compile All**



5. Start simulation from Simulate → Start Simulation.

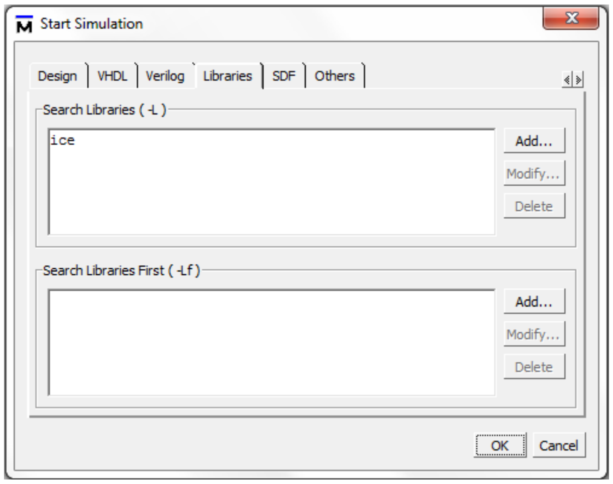
Figure 33: **Start Simulation**



Name	Status	Type	Order	Modified
sb_ice_lc.v	?	Verilog	3	12/17/10
counter_sbt.v	?	Verilog	0	11/26/10
counter_tb.vhd	?	VHDL	1	11/26/10
sb_ice_syn.v	?	Verilog	2	11/22/10

6. In the simulation window, select the tab “Libraries” and add the “**ice**” resource library in the search libraries path.

Figure 34: **Add “ice” resource library to the project**

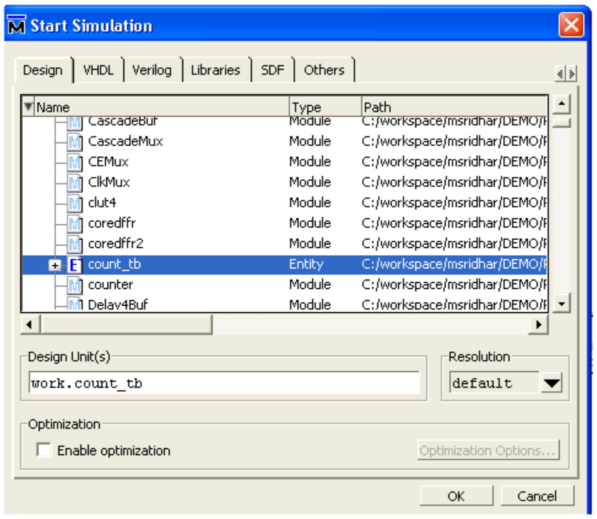


Search Libraries (-L)
ice

Search Libraries First (-Lf)

7. Expand the work folder in the “Start Simulation” window and highlight “counter_tb”. Click OK.

Figure 35: **Run Simulation**



Name	Type	Path
CascadeBuf	Module	C:/workspace/msridhar/DEMO/
CascadeMux	Module	C:/workspace/msridhar/DEMO/
CEMux	Module	C:/workspace/msridhar/DEMO/
ClkMux	Module	C:/workspace/msridhar/DEMO/
clut4	Module	C:/workspace/msridhar/DEMO/
coredffr	Module	C:/workspace/msridhar/DEMO/
coredffr2	Module	C:/workspace/msridhar/DEMO/
count_tb	Entity	C:/workspace/msridhar/DEMO/
counter	Module	C:/workspace/msridhar/DEMO/
Delay4Buf	Module	C:/workspace/msridhar/DEMO/

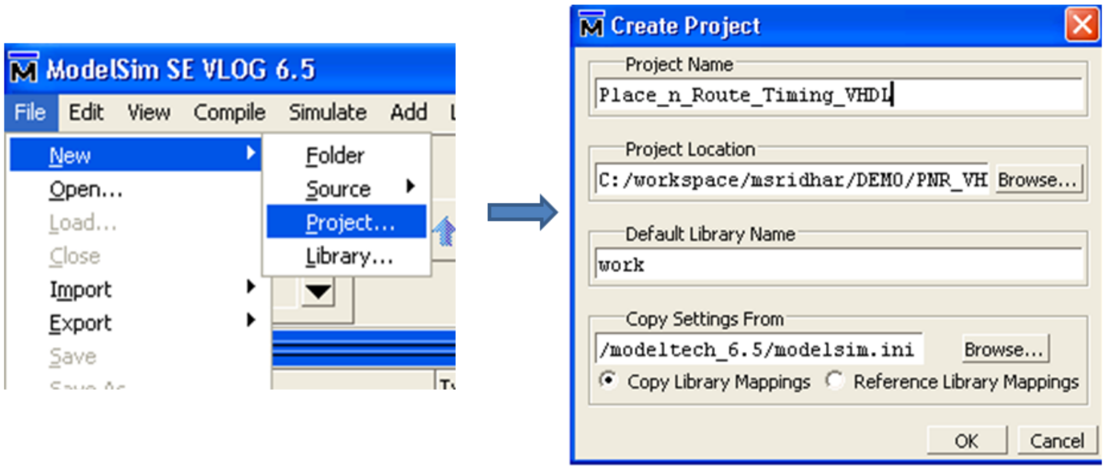
Design Unit(s): work.count_tb
Resolution: default
Optimization
Enable optimization

This will open the simulation window. Add the signals wave window and perform simulation as explained in the “Pre-Synthesis Simulation” section from point 7 onwards.

Place-and-Route Timing Simulation (VHDL)

1. Generate the files that are required for the post-route timing simulation model using the steps described in the section “Generating Files Required for Post-Synthesis and Post-Route Simulations”.
2. Create new project using File → New → Project. In the New Project window, give a project name and browse to a folder where the project needs to be saved. Click OK.

Figure 36: **Create New Project**



Project Name: Place_n_Route_Timing_VHDL

Project Location: C:/workspace/msridhar/DEMO/PNR_VH Browse...

Default Library Name: work

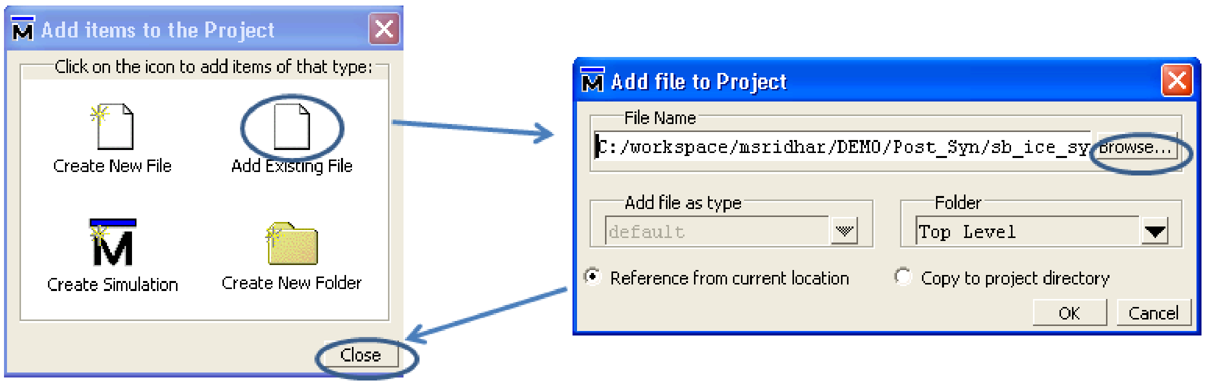
Copy Settings From: /modeltech_6.5/modelsim.ini Browse...

[✓ Copy Library Mappings] Reference Library Mappings

3. Click “Add Existing Files” and add the following files:
   a. counter_sbt.vhd
   b. counter_tb.vhd

   Close the “Add items to the Project” window once all the files are added.

Figure 37: **Add Files to Project**



File Name: C:/workspace/msridhar/DEMO/Post_Syn/sb_ice_sy Browse...

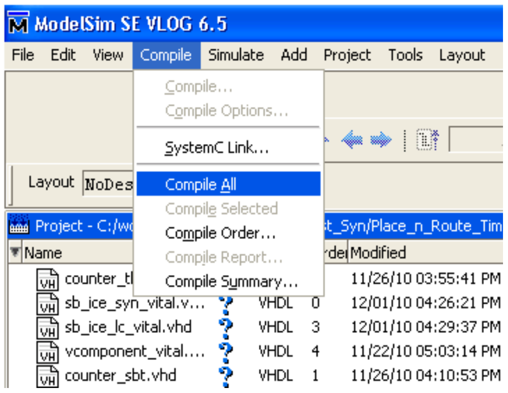
Add file as type: default

Folder: Top Level

[✓ Reference from current location] Copy to project directory

4. Choose Compile → Compile All.

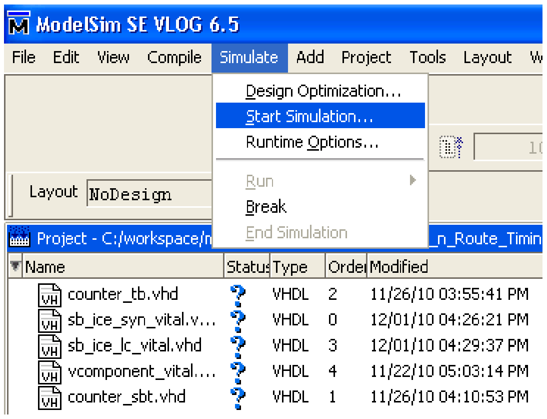
Figure 38: **Compile All**



Name				Modified
counter_tl				11/26/10 03:55:41 PM
sb_ice_syn_vital.v...	?	VHDL	0	12/01/10 04:26:21 PM
sb_ice_lc_vital.vhd	?	VHDL	3	12/01/10 04:29:37 PM
vcomponent_vital....	?	VHDL	4	11/22/10 05:03:14 PM
counter_sbt.vhd	?	VHDL	1	11/26/10 04:10:53 PM

5. Start simulation using Simulate → Start Simulation.

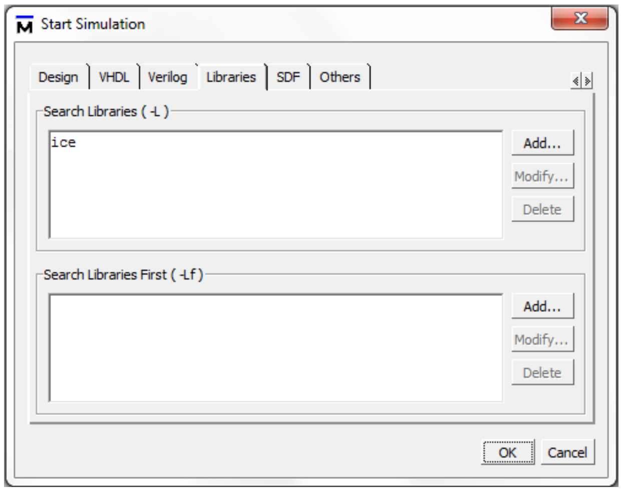
Figure 39: **Start Simulation**



Name	Status	Type	Order	Modified
counter_tb.vhd	?	VHDL	2	11/26/10 03:55:41 PM
sb_ice_syn_vital.v...	?	VHDL	0	12/01/10 04:26:21 PM
sb_ice_lc_vital.vhd	?	VHDL	3	12/01/10 04:29:37 PM
vcomponent_vital....	?	VHDL	4	11/22/10 05:03:14 PM
counter_sbt.vhd	?	VHDL	1	11/26/10 04:10:53 PM

6. In the Simulation window, select the “Libraries” tab and add the “**ice**” resource library in the search libraries path.

Figure 40: **Add “ice” resource library to the project**



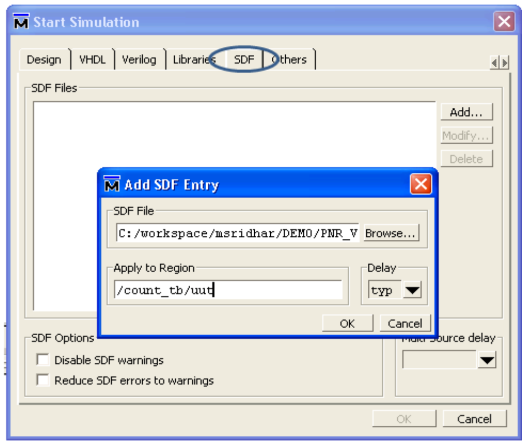
Search Libraries (-L)

```
ice
```

Search Libraries First (-Lf)

7. In the Simulation window, select the SDF tab. Add the SDF file counter_sbt_vital.sdf and type “/count_tb/uut” in “Apply to Region”. Click OK.

Figure 41: **Add SDF to project**



SDF Files

SDF File: C:/workspace/msridhar/DEMO/PNR_V

Apply to Region: /count_tb/uut

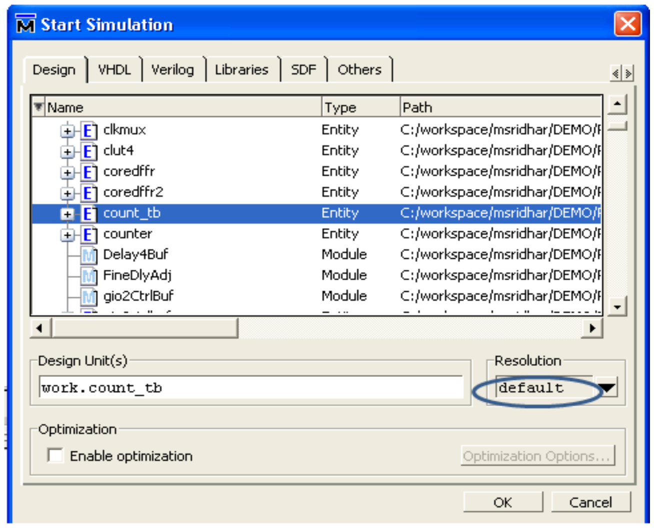
Delay: typ

SDF Options

- [ ] Disable SDF warnings
- [ ] Reduce SDF errors to warnings

8. In the Design tab, select "count_tb" under work. Make sure that the "Time Resolution" was set to PS. Click OK.

Figure 42: **Run Simulation**



Name	Type	Path
clkmux	Entity	C:/workspace/msridhar/DEMO/
clut4	Entity	C:/workspace/msridhar/DEMO/
coredffr	Entity	C:/workspace/msridhar/DEMO/
coredffr2	Entity	C:/workspace/msridhar/DEMO/
count_tb	Entity	C:/workspace/msridhar/DEMO/
counter	Entity	C:/workspace/msridhar/DEMO/
Delay4Buf	Module	C:/workspace/msridhar/DEMO/
FineDlyAdj	Module	C:/workspace/msridhar/DEMO/
gio2CtrlBuf	Module	C:/workspace/msridhar/DEMO/

Design Unit(s): work.count_tb

Resolution: default

Optimization

- [ ] Enable optimization

9. This will open the simulation window. Add the signals wave window and perform simulation as explained in the "Pre-Synthesis Simulation" section from point 7 onwards.

Conclusion

The Lattice iCEcube2 development software provides a complete FPGA implementation environment for today's FPGA designers. For simulation needs, the Mentor ModelSim simulator complements the iCEcube2 development software and is a cost effective, easy to use simulation tool.

References

For more information on products, solutions, and applications enabled by Lattice Semiconductor Corporation, take the next step and visit www.latticesemi.com.

Revision History

Version	Date	Description
1.7	24-Nov-2020	Updated for ModelSim usage changes.
1.6	18-Dec-2013	Added resource library details for iCE40LM primitives.
1.5	26-APR-2013	Added resource library generation details for post route VHDL simulations.
1.4	24-DEC-2010	Added sections for pre/post synthesis, functional/timing, Verilog/VHDL simulations
1.3	23-DEC-2008	Updated corporate contact information.
1.2	20-OCT-2008	Initial release, updated iCEcube screen shots and timing simulation information.
1.1	06-AUG-2008	First draft, functional simulation only.

Copyright

Trademarks

Disclaimers

Lattice Semiconductor Corporation

5555 N.E. Moore Court,

Hillsboro , Oregon 97124-6421
United States of America

Tel: +1 503 268 8000
Fax: +1 503 268 8347
www.latticesemi.com