Digital Simulations

Dr Basel Halak

Learning Outcomes

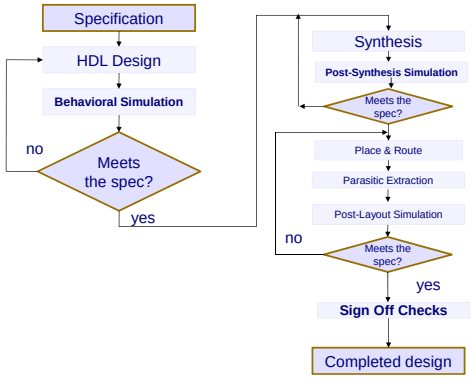
After completing this unit, you should be able to:

- To describe the different Sign –off stages in the digital design process
- To compile and simulate HDL models in Modelsim
- To perform different types of digital simulations

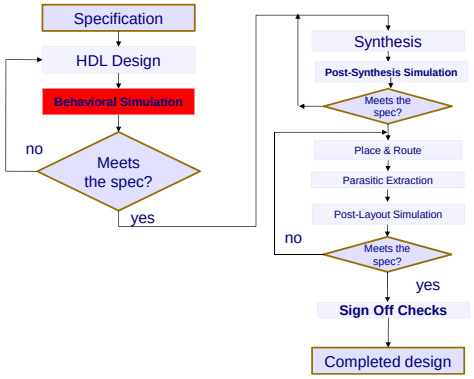
After completing this unit, you should be able to:

- To describe the different Sign–off stages in the digital design process
- To compile and simulate HDL models in Modelsim
- To perform different types of digital simulations

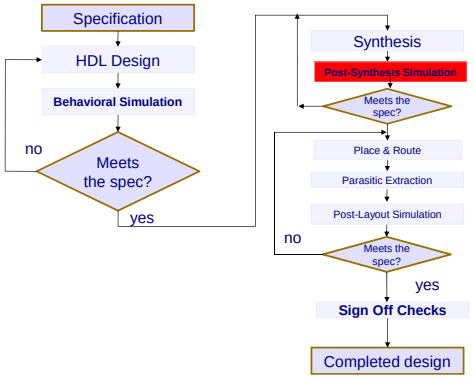
Digital Design Flow



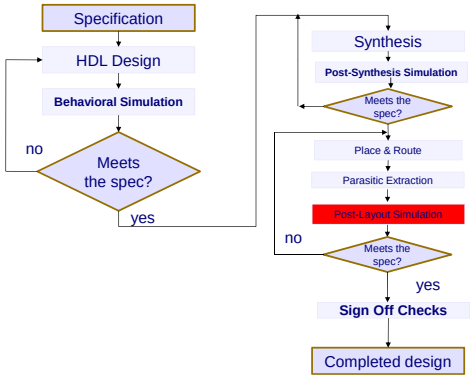
Digital Design Flow



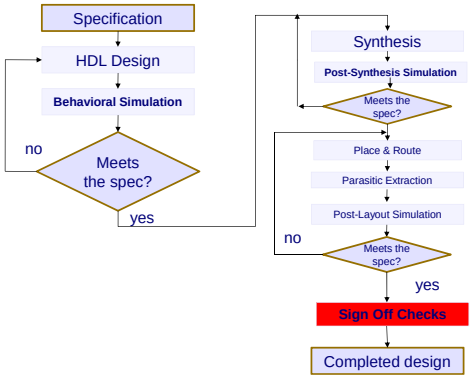
Digital Design Flow



Digital Design Flow



Digital Design Flow

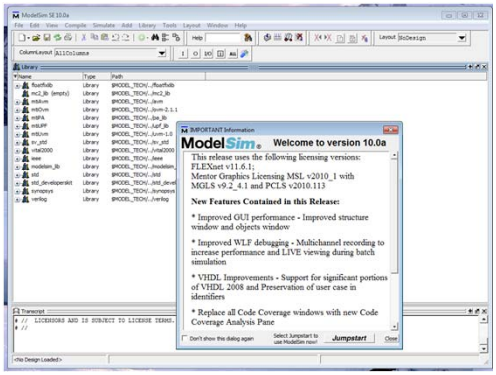


Learning Outcomes

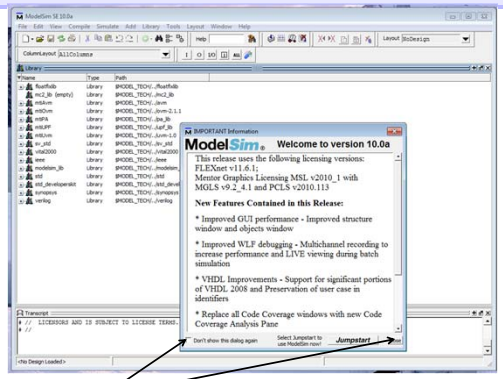
After completing this unit, you should be able to:

- **T describe the different Sign –off stages in the digital design process**
- **To compile and simulate HDL models in Modelsim**
- **To perform different types of digital simulations**

Load Modelsim: CAD->Modelsim

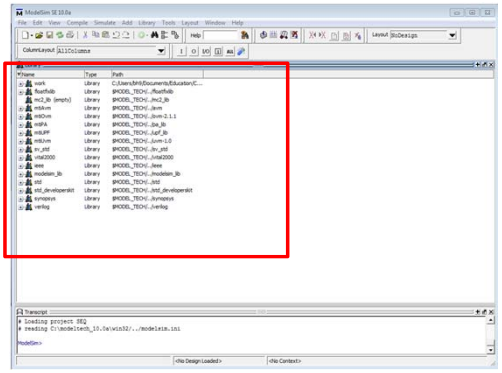


Load Modelsim: CAD->Modelsim

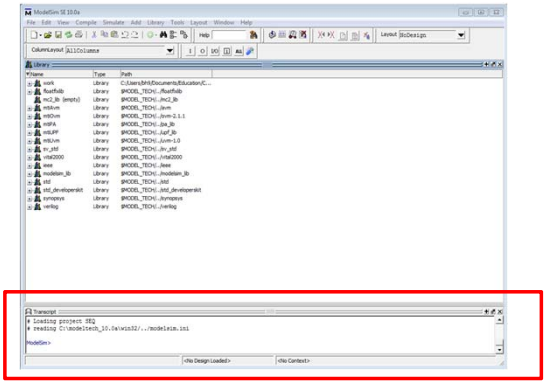


Click this and get rid of it

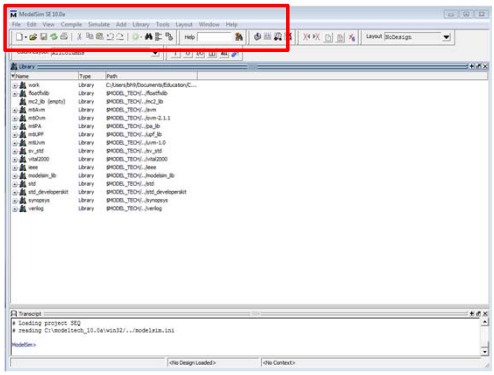
Libraries



Transcript -> commands

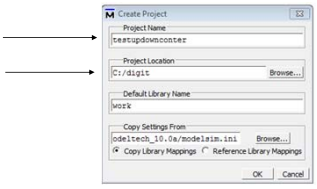


Usual Windows Style menus

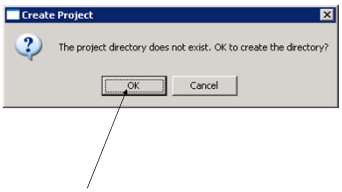


Step 1 : Create a Project

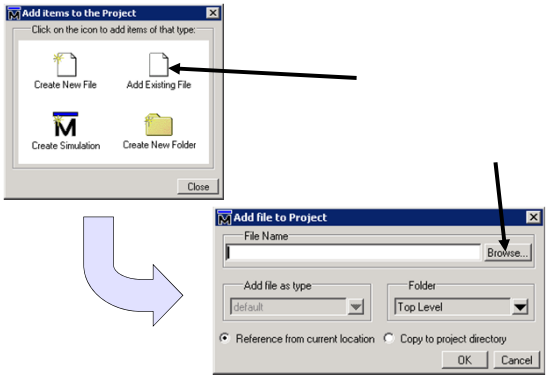
- File -> New -> project



Create the new directory

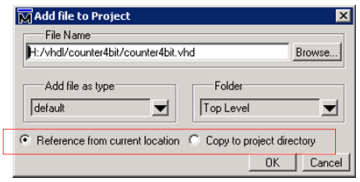


Add the files

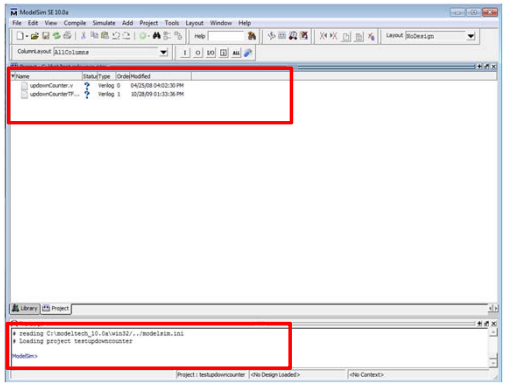


Add the files

- You can reference the files, or you can copy into the directory – this is a good idea if you are going to modify things

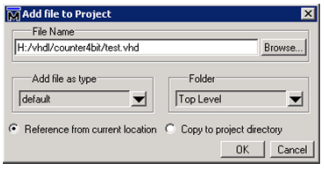


Note the transcript & workspace…

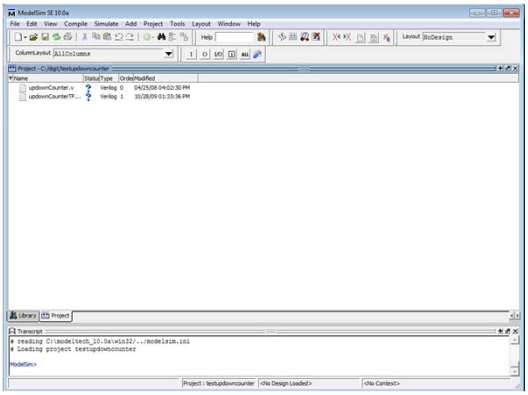


You can add more files

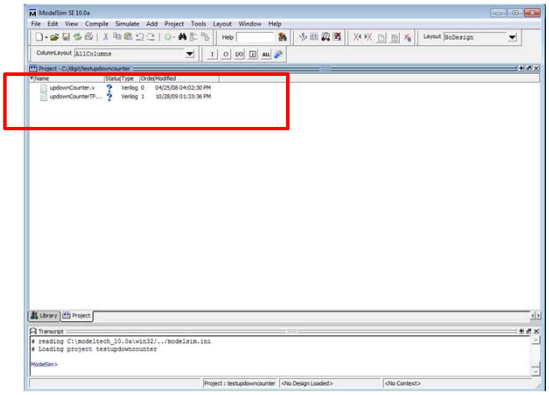
- File -> Add to project -> Existing Files



Now we have the complete project



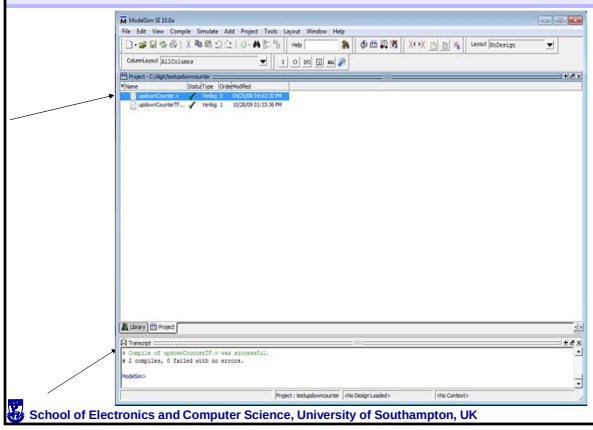
Note the status and type



What Now – Compile

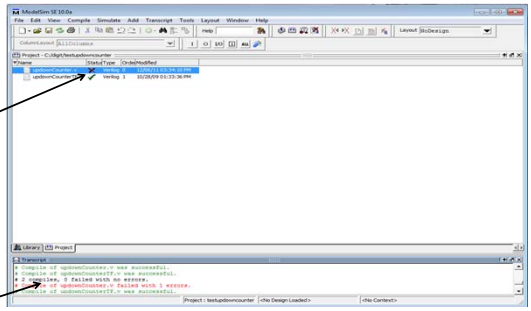
- **We can compile individual files**
  - Useful for large designs
- **We can compile ALL the Verilog files**
  - What we usually do when all the individual files have been checked
- **Compile -> Compile All (from the menus)**
  - Or use the menu bar icon

Compile Status



What if it goes wrong?

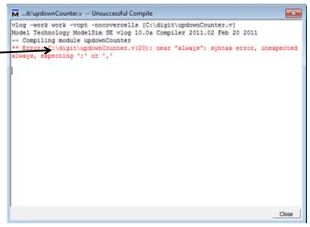
- E.g. syntax error



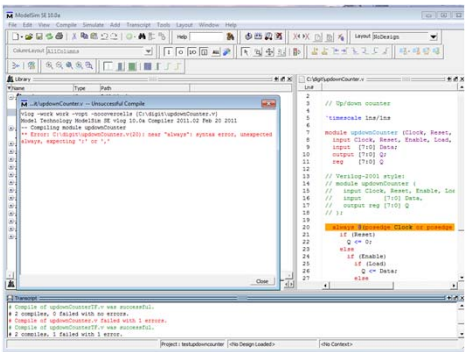
Edit a Verilog File

- Right click on the Verilog file and choose Edit from the pop up menu to bring up the editor
- Double click on the error in the transcript to see where the error is…

Double Click Again



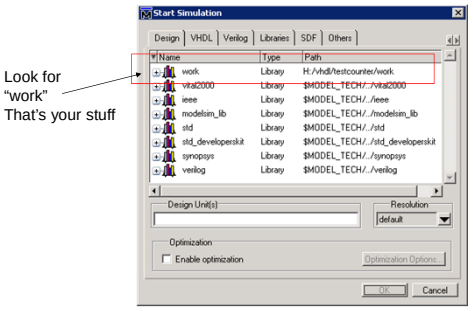
Debug the model



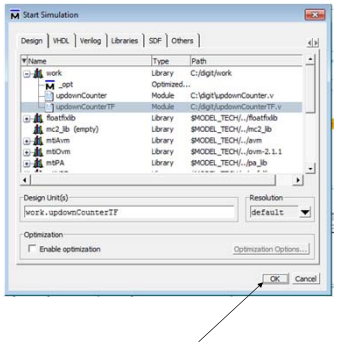
Simulate

- Simulate -> Start Simulation
- Or click on the icon:

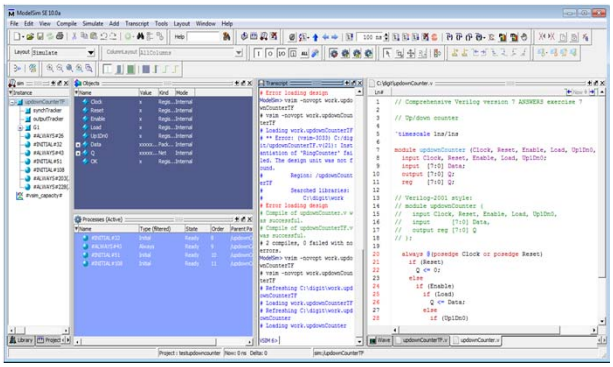
List of compiled Units



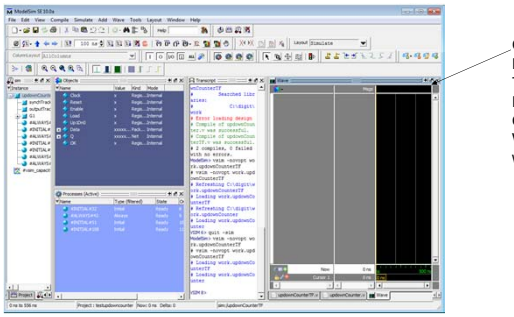
Choose the test circuit



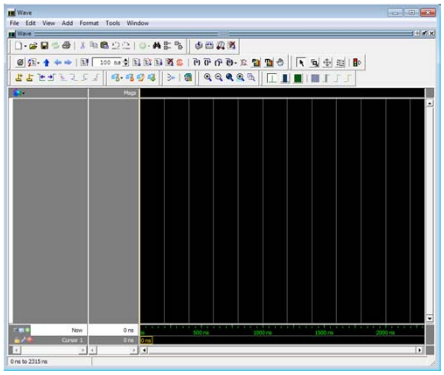
The Design loads up…



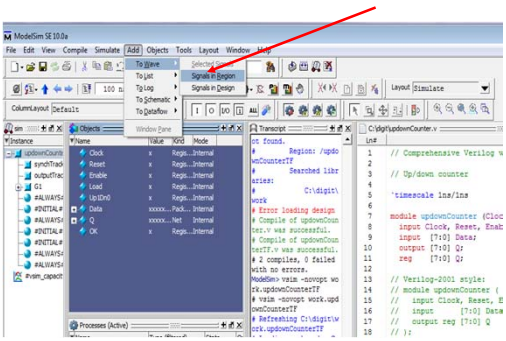
Get wave viewer



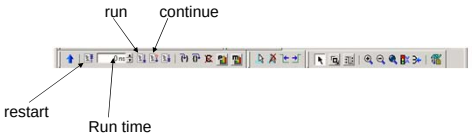
Waveform Window



Add Signals to the waveform window

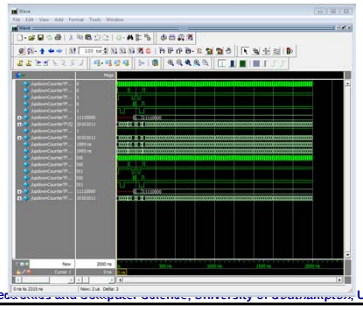


Run Commands



Run a 50 us simulation

- **Type the following command:**
  - Run 2 us



Learning Outcomes

After completing this unit, you should be able to:

- **T describe the Different Sign –off stages in the digital design process**
- **To compile and simulate HDL models in Modelsim**
- **To perform different types of digital simulations**

Types of Digital Simulations

- **Behavioural Simulations:**
- It is used to check the functionality of the design **before synthesis**
- You will need the following files:

1. HDL description of your design
2. A test bench

Types of Digital Simulations

- **Behavioural gate level simulation**
  - **It is a post-synthesis simulation**
  - **You will need** :

1. HDL description of the synthesized design
2. HDL models of all cells in your design (e.g. Verilog models), and
3. A test bench

  - **How to perform Behavioural gate level simulation**

Copy all the required files into your simulation directory and follow the same procedure for behavioural simulation.

Types of Digital Simulations

- **Behavioural gate level simulation with SDF timing**
  - **It is a post-synthesis simulation**
  - **You will need:**

1. HDL description of the synthesized design
2. SDF(standard delay format) file (which can be obtained for the synthesis tool and from the layout tool)
3. HDL models of all cells in your design (e.g. Verilog models), these models should contain specify blocks in their behavioural view that have delay information
4. A test bench

Note: you can generate SDF files in Design Compiler using this command:

write_sdf filename.sdf

Types of Digital Simulations

- **How to do Behavioural gate level simulation with SDF timing**

1. Copy all the required files into your simulation directory
2. Edit the HDL description of the synthesised design by adding the following statement: ***initial $sdf_annotate ("filename.sdf")*** as shown in figure 1
3. Save changes to your HDL
4. Follow the same procedure for behavioural simulations

```
module alu ( Clock, A, B, Op, F, Cout, Equal );
  input [7:0] A;
  input [7:0] B;
  input [3:0] Op;
  output [7:0] F;
  input Clock;
  output Cout, Equal;
  wire   N62, N63, N64, N68, N69, N70, N72, N73, N79, N81, 
         N99, N100, N106, n47, n48, n49, n50, n51, n52, n53,
         n188, n189, n190, n191, n192;
  wire   [7:0] Tmp;

initial $sdf_annotate ("design.sdf");

DF1 F_reg_7_ ( .D(Tmp[7]), .C(Clock), .Q(F[7]) );
DF1 F_reg_6_ ( .D(Tmp[6]), .C(Clock), .Q(F[6]) );
DF1 F_reg_5_ ( .D(Tmp[5]), .C(Clock), .Q(F[5]) );
DF1 F_reg_4_ ( .D(Tmp[4]), .C(Clock), .Q(F[4]) );
DF1 F_reg_3_ ( .D(Tmp[3]), .C(Clock), .Q(F[3]) );
```

Figure 1

Types of Digital Simulations

- **Why use SDF file?**

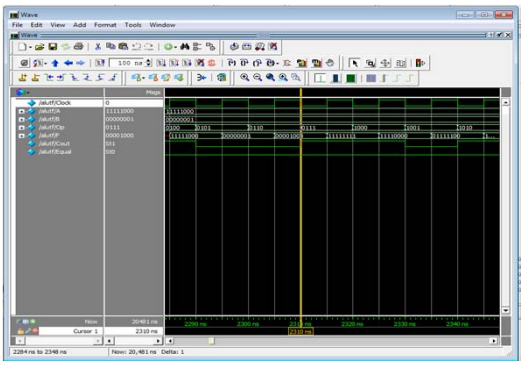
```
(CELL
  (CELLTYPE "NOR40")
  (INSTANCE U214)
  (DELAY
    (ABSOLUTE
    (IOPATH A Q (0.310:0.311:0.311) (0.401:0.401:0.401))
    (IOPATH B Q (0.287:0.288:0.288) (0.255:0.255:0.255))
    (IOPATH C Q (0.260:0.262:0.262) (0.263:0.270:0.270))
    (IOPATH D Q (0.204:0.263:0.263) (0.226:0.243:0.243))
    )
  )
)
```

SDF delay information for an Instance of the NOR40 cell

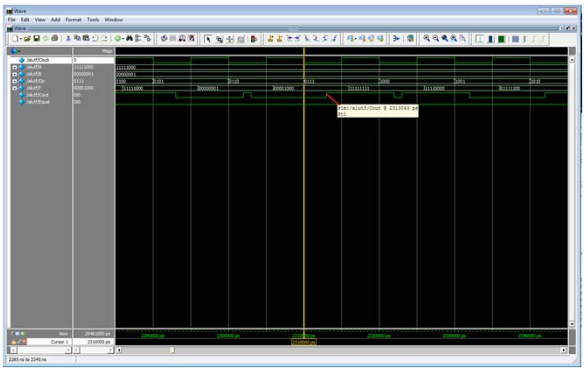
```
module NOR40 (A,B,C,D,Q);

output Q;
input A,B,C,D;

nor (Q,D,C,B,A);

`ifdef functional
`else
specify
  (A => Q) = (1,1);
  (B => Q) = (1,1);
  (C => Q) = (1,1);
  (D => Q) = (1,1);
endspecify
`endif

endmodule
```

Behavioural Verilog Code for NOR40

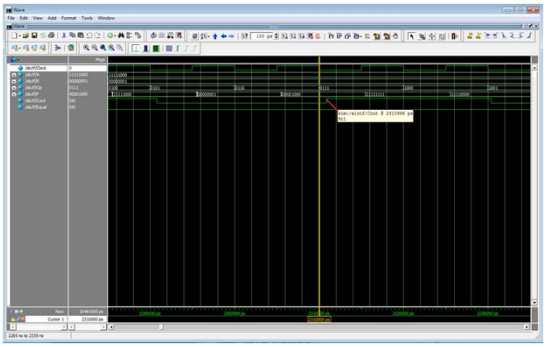
Behavioural Simulations



Behavioural Gate Level Simulation



Behavioural Gate Level Simulation with SDF timing



Digital Simulation Lab Instructions

For this lab you will need:

For Behavioural simulation

1. HDL model of the design (qmults.v)
2. A test bench (test.v)

SOC Encounter Lab Instructions

For this lab you will need:

For post Synthesis simulation

1. A synthesised Verilog net list of the design (this is obtained from the synthesis stage)
2. A test bench
3. Your timing constraints file "design.sdf" (this is obtained from the synthesis stage)
4. HDL models of all cells in technology library

SOC Encounter Lab Instructions

For this lab you will need:

For post Layout simulation

1. Post layout netlist (this is obtained from the layout stage)
2. A test bench
3. Your timing constraints file "design.sdf" (this is obtained from the layout stage)
4. HDL models of all cells in technology library

Behavioural Simulations

1. Create a working directory called BehaviouralSim.
2. Save the qmults HDL file and the test bench in this folder.
3. Open the test bench in a text editor and investigate it.
4. Set the clock period in the test to 2 ns.
5. Create a Modelsim Project.
6. Compile all files.
7. Run the simulations and verify the design function correctly.
8. Does the design simulate correctly if you reduce clock period to 0.5 ns? Why?.
9. Save a printout of your simulations.

Post Synthesis Simulations

1. Create a working directory called PostSynSim.
2. Save the qmults HDL file, the test bench, HDL models of the technology library and SDF timing file in this folder.
3. Open the test bench in a text editor and set the clock frequency in the test to your maximum achievable frequency (from the synthesis stage).
4. Open the SDF file in a text editor and comment out all lines that begin with the word “Removal”.
5. Create a second Modelsim Project.
6. Compile all files.
7. Run the simulations and verify the design function correctly.
8. Does the design simulate correctly if you reduce clock period to 0.5 ns? Why?
9. Save a printout of your simulations.

Post Layout Simulations

1. Create a working directory called PostLaySim
2. Save the qmults HDL file, the test bench, HDL models of the technology library and SDF timing file in this folder.
3. Open the test bench in a text editor and set the clock frequency in the test to your maximum achievable frequency (from the synthesis stage).
4. Open the SDF file in a text editor and comment out all lines that begin with the word “Removal”
5. Create a third Modelsim Project
6. Compile all files
7. Run the simulations and verify the design function correctly.
8. Does the design simulate correctly if you reduce clock period to 0.5 ns? Why?
9. Save a printout of your simulations