Experiment - 1

July 27, 2020

Aim:

Introduction to VHDL and MODELSIM simulator

Software used:

MODELSIM

Theory :

VHDL: Very high speed intergrated circuit Hardware Description Language.

It is a hardware description language that can be used to model a digital system at many levels of abstraction ranging from algorithmic level to the gate level. The complexity of the digital system being modeled coudl vary from that of a simple gate to a complete digital electronic systems, or anything in between. The Digtal Systems can be also be descirbed hierachically. Timing can also be explicitly modeled of the Following languages: - sequential language - concurrent language - net-list language - timeing speciifications - waverform genereation language

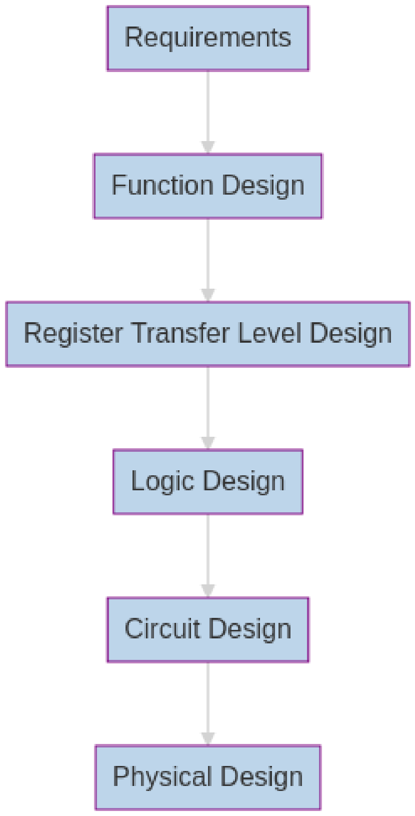
HDL have a specific purpose, rather that being used to design software, an HDL is used to define a computer chip. VHDL can be used to describe any type of circuitry and is frequently used in the design, simulation, and testing of processors, CPUs, motherboards, FPGAs, ASICs, and many other types of types of digital circuitry.

ModelSim: ModelSim is a multi-language HDL simulation environment by Mentor Graphics, for simulation of hardware description languages such as VHDL, Verilog and SystemC, and includes a built-in C debugger. ModelSim can be used independently, or in conjunction with Intel Quartus Prime, Xilinx ISE or Xilinx Vivado. Simulation is performed using the graphical user interface (GUI), or automatically using scripts

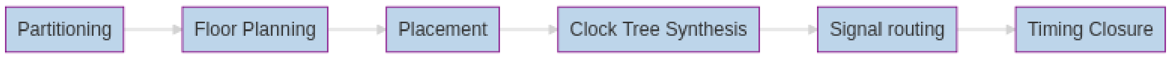
- Unified mixed language simulation engine for ease of use and performance
- Native support of Verilog, SystemVerilog for design, VHDL, and SystemC for effective verification of sophisticated design environments
- Fast time-to-debug, easy to use, multi-language debug environment
- Advanced code coverage and analysis tools for fast time to coverage closure
- Interactive and Post-Sim Debug available so same debug environment used for both
- Powerful Waveform compare for easy analysis of differences and bugs
- Unified Coverage Database with complete interactive and HTML reporting and processing for understanding and debugging coverage throughout your project

Design Flow:

- Requirements
- Function Design: - Behavioral Simulation
- Register Transfer Level Design: - RTL Simulation validation
- Logic Design: - Logic simulation - Verifaction - Fault Simulation
- Circuit Design: - Timing simulation - Circuit Analysis
- Physical Design : - Design Rule Checking - MUXes, Flip-flops, Circuit level Desinging - Copper cloded path - CAD tools are used

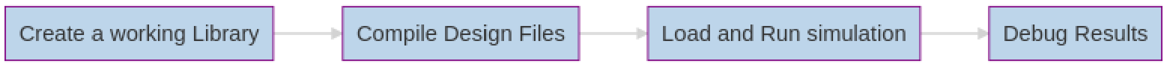


Physical Design Flow

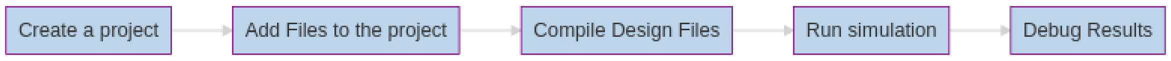


Block Diagram showing ModelSim Software:

Basic Simulation Flow



Project Flow



Steps for using Software:

Usage Flow:

1. Start ModelSim
2. Create a project3.
3. Add VHDL file to project
4. Edit/Compilation:
    1. Compile files into new library by selecting the compile button
    2. Complete the compilation by selecting file and pressing compile
5. Load the design unit. Select the Load design button from the tool bar:
    1. The load design dialog box lets us select the library and top=level design unit to simulate.
6. Select the entity and choose Load to accept these settings
7. Select view > All from the main window menu to open all ModelSium windows
8. From the signal window menu, select view > List > Signal in Region. This commands displaysthe top-level signals in the list window.
9. Add top-level signals to the wave window by selecting view>wave>signals in Region from thesignals window menu

Running the simmulation:

1. Click in the main window and enter the fallowing command at the VSIM prompt: `force clk 1 50, 0 100 -repeat 100`
2. Now we will exercise two different Run functions from the toolbar buttons on either main windowor wave window. Select the Run button first. When the run is complete, select Run All.
3. Select the Break button on either Main or Wave toolbar to interrupt the run. The simulator willstop running as soon as it gets to an acceptable stopping point.
4. Select the Continue Run button to resume the run that we interrupted. ModelSim will hit thebreakpoint, as shown by an arrow in the source window and by a Break message in the Mainwindow.
5. Click the Step button to single-step through the simulation. Notice that the values change in thevariables window.
6. When we're done, quit the simulator by entering the command: quit -force3

Example of VHDL Code:

AND GATE:

```
entity and is
  port (i0, i1 : in bit;
  o0: out bit);
end and;
architecture andgate of and is
begin
  o0 <= i0 and i1;
end andgate;
```

OR GATE:

```
entity or is
  port (i0, i1 : in bit;
  o0: out bit);
end or;
architecture orgate of or is
begin
  o0 <= i0 or i1;
end orgate;
```

NOT GATE:

```
entity not is
  port (i0 : in bit;
  o0: out bit);
end not;
architecture notgate of not is
begin
  o0 <= not i0;
end notgate;
```

NAND GATE:

```
entity nand is
  port (i0, i1 : in bit;
  o0: out bit);
end nand;
architecture nandgate of nand is
begin
  o0 <= not (i0 and i1);;
end nandgate;
```

NOR GATE:

```
entity nor is
  port (i0, i1 : in bit;
  o0: out bit);
end nor;
architecture norgate of nor is
begin
  o0 <= not(i0 or i1);
end norgate;
```

XOR GATE:

```
entity xor is
  port (i0, i1 : in bit;
  o0: out bit);
end xor;
architecture xorgate of xor is
begin
  o0 <= ((not i0) and i1) or ((not i1) and i0);
end xorgate;
```

XNOR GATE:

```
entity xnor is
  port (i0, i1 : in bit;
  o0: out bit);
end xnor;
architecture xnorgate of xnor is
begin
  o0 <= ((not i0) and (not i1)) or ( i0 and i1);
end xnorgate;
```

Result:

Basics of VHDL were studied and Introduction to ModelSim Software. Basic gates were implementedin VHDL.