CAD for VLSI Design - II

Lecture 11
V. Kamakoti and Shankar Balachandran
Overview of this Lecture

• Synthesis
  – Process of converting one representation of a circuit (source representation) to another functionally equivalent representation (target representation).
  – Normally the target is less abstract (close to device level) than the source
  – Synthesis done across several levels
Synthesis Transformations

Level 1

Level $i$

Level $i+1$

Level N

Specifications

$X = AB; Y = CD; Z = X + Y$
Synthesis of Verilog Code

- Three things are needed
  - Verilog Model
  - Constraints on the circuit
    - Area
    - Delay
  - Library Models
    - What kind of components we have?
    - How are they characterized for area, delay etc?
Two Major Functions

• Translation – Converts the textual representation to a net-list containing target technology cells.
  – Also called mapping.
• Optimization – Transforms the functionality to meet requirements.
• Optimization is an iterative process
  – Enter constraints
  – Run the tools with different options
  – Tune the constraints
• Ideally, the synthesized circuit should simulate the same as the input Verilog code.
Verilog as a Synthesis Language

• Verilog is *primarily* a simulation language

• Features of Verilog
  – Describe hardware at Behavioral, RTL and Gate level
  – Create Test Stimulus
  – Error checking on the model and its usage
  – File I/O
## Subset for Synthesis

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Synthesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Description</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Test Stimulus</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Error Checking</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>File I/O</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

- Synthesis tools should be concerned only about the subset that enables hardware description.
Verilog as a Synthesis Language

- Vendors supply libraries to target ASICs or FPGAs
- Verilog style and synthesizable subset are vendor specific
- Tools come with guidelines
  - Read the guidelines
  - The best route to squeeze the juice out of synthesis tools
Synthesis Tools – Usage

• The documentation does not completely define the operation of tool; in contrast to the Verilog simulator where the LRM tells you exactly what to expect.

• The tool is complex. There are many options. How to exercise these options is the corporate value added.

• Use scripts to develop repeatable experiments
  – Avoid interactive input
Focus on Synthesis

Goal of the course

Synthesis Tool X

Simulatable Verilog

Synthesis Tool Y

Synthesis Tool Z
Verilog Synthesis Flow

1. Create VHDL textual description
2. Compile and simulate to verify functionality
   - If ok, then
     - Compile for synthesis
       - If ok, then
         - Elaborate for synthesis
       - If not ok, then go back to compile for synthesis
     - If not ok, then go back to compile and simulate

3. Optimize for area
   - If ok, then
     - Set Constraints
     - Optimize for Timing/Area
     - Gate-level simulation
       - If not ok, then go back to optimize for area
     - If ok, then layout and resimulation
   - If not ok, then go back to set constraints

4. Revise as required
Synthesis Flow

• Basic synthesis steps are:
  – Analyze
  – Elaborate
  – Compile
  – Report
  – Save

• Basic steps are subject to constraints and options
Analysis

• Source Verilog is analyzed for conformance to the synthesis subset
• Verilog files that will simulate correctly may fail this step because high level constructs are not supported
• Successful analysis will result in the generation of an intermediate form
Elaboration

• A technology independent design is generated from the intermediate form
• The design is expressed in terms of functional blocks and generic gates
Technology Independent Circuit – ONES_CNT

Counting the number of One’s in a 3-bit vector A[2:0]
Compile

- Translates the technology independent design to a library based design
- Compilation is frequently constrained in terms of desired area, delay or power
- Generates a gate level circuit expressed in terms of ASIC library macros
Synthesized Circuit for ONES_CNT

Counting the number of One’s in a 3-bit vector A[2:0]
Questions and Answers

Thank You