**FPGA CAD TOOL FLOW**

Lab Lecture 2

September 5, 2003

Aaron P. Hurst
(ahurst@eecs.berkeley.edu)

---

**What is the Flow? (generic)**

- Design Entry verification
- Synthesis verification
- Place and Route verification
- Hardware Programming verification

---

**What is the Flow? (specific)**

---

**Design Entry**

- Schematics (not in CS 150)
- Hardware Description Language (HDL)
  - Human readable
  - Hierarchical
  - Meaningful naming
  - "Think Hardware"

---

**Why use an HDL?**

**Digital Design Productivity, in Gates/Week**

<table>
<thead>
<tr>
<th>HDL Type</th>
<th>Productivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavioral HDL</td>
<td>2K-10K</td>
</tr>
<tr>
<td>RTL HDL</td>
<td>1K-2K</td>
</tr>
<tr>
<td>Gates</td>
<td>100-200</td>
</tr>
<tr>
<td>Transistors</td>
<td>10-20</td>
</tr>
</tbody>
</table>

Source: DataQuest
Design Verification

- develop a test bench
  - Drive Inputs (a.k.a. test vectors)
    1. random vectors – need many
    2. targeted vectors – check anticipated problems
  - Check Outputs

Key concept: coverage
- how many of the potential problems have we tested?

Synthesis

- converts HDL into a netlist of simple gates

Place and Route

- FPGA is like a giant grid...
  - find an optimal arrangement of gates (LUTs) and paths to connect them
Accurate Timing Simulation

- ModelSim initially uses ideal values
  - not necessarily very accurate
- At the end of place and route, the Xilinx tools
  know the exact wirelengths and delays
- We can feed this exact timing back into ModelSim
- Simulate with real, accurate timing
- Tricky process: TAs will help

Hardware Verification

- Careful verification and testing after every step of the flow
  makes this last step easy
- However, debugging errors here can be incredibly tedious
  - incredibly expensive in industry (millions of $)
- If it works, you’re done!