Lab Lecture 3
Verilog Simulation Mapping
2/6/2004

Michael Liao
Greg Gibeling
Sandro Pintz

Motivation
- Basic Introduction to HDL Design Entry (Covered in Main Lecture)
- Architectural Definition and Modularization (Partitioning)
- What are “Good Verilog Techniques”
- Behavioral vs Structural Verilog

Top Down Architecture (1)
- Top Down Refinement Process
- Start Here:

Top Down Architecture (2)
- End Here:

Partitioning
- Define modules that:
  - Have clean, well defined interfaces to other modules
  - Are manageable in size
  - Can be verified independently
  - Might be designed by different people
  - Functionally makes sense
Practical Verilog (1)

- Remember:
  - "Think Hardware" as you write
  - Meaningful Signal Names:
    - clockEnable, busSync (good)
    - A, B&*_ (not good)
  - Choose a naming style and STICK TO IT
  - Match modules and filenames, 1 module per file

Practical Verilog (2)

- Clear Comments:
  - Comment a whole section
  - Comment for special cases (exceptions)
  - Comments for special signals
- Choose a Code Structure and STICK TO IT
  - State machine style
  - Special section for flops
  - Special section for structural stuff (instantiations)

Think Hardware (1)

- You are NOT programming
- Easiest way to waste time
- Programming will only confuse you
- You are describing hardware
- You should have a circuit in mind
- Know what you want BEFORE YOU WRITE IT
- Don’t try to hack it together
- You will not be happy with the results

Think Hardware (2)

- Hardware isn’t like software
- Things happen all at once
- There is no sequential execution model
- You are not writing a set of instructions
- You are basically creating a schematic in text form
- ALWAYS REMEMBER THIS!

Example

```
module adder (
    // Behavioral Code
    // Module Instantiations (Structural)
    // Instantiate Gates
    // Instantiate Modules
    // Instantiate Primitives
    // Most levels are mixed
```
**Behavioral vs Structural (2)**

- Structural
  - Structural
  - Structural
  - Behavioral
  - Behavioral

**Lab #2 Part I**

- Behavioral Only
- No Instantiations

**Lab #2 Part II**

- Behavioral:
  - Adder
  - Register
  - Structural:
  - Top
  - Two Instantiations

**Lab #2 Part III**

**Lab #2 Part IV**

- Half Adder
  - [Diagram of half adder]

- Full Adder
  - [Diagram of full adder]