Hardware Modeling Overview

Introductory VHDL Methodology

Objective

After completing this module, you will be able to…

- Discuss the VHSIC initiative
- Define the terms ‘Behavioral’ and ‘RTL’
- Define the terms ‘Inference’ and ‘Instantiation’
- Define Hardware Modeling ‘Levels of Abstraction’

What is VHDL?

- **An acronym within an acronym, VHDL stands for *VHSIC Hardware Description Language***
- **Meanwhile VHSIC stands for *Very High Speed Integrated Circuit***
- **With that, we begin to understand both the origin and the intent of the language**

Understanding the Intent?

- **Given the inauspicious origin of the language and standard, it is worth noting that VHDL is first and foremost, a tool for hardware modeling- - that is to say “simulation” as opposed to “synthesis”**

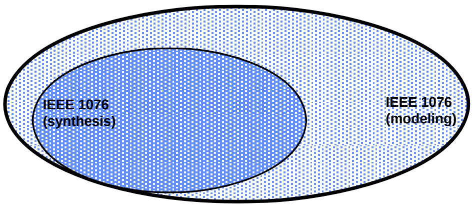
  The IEEE 1076 standard is exhaustive with respect to modeling, but defines only broad parameters for synthesis

  The result - a given hardware module does not necessarily lend itself to a consistent and universal gate-level implementation across various tools and target technologies

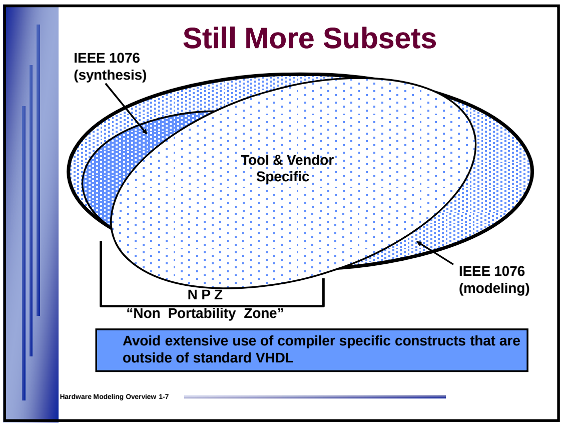
Formalization of VHDL

- **The IEEE formally adopted the language as a standard, ratifying it in 1987, IEEE 1076. Like any IEEE standard there is a minimum five year period for modifications to the original**
- **This actually occurred in 1993 and VHDL-93 is now the official version of the language, however most tools provide support for older modules (VHDL-87) and some are simply still catching up**

Language Subsets



- **Not all VHDL constructs are synthesizable. For example, *wait for 10 ns* is a common modeling construct, but does not generate any corresponding gate-level component**



Still More Subsets

IEEE 1076 (synthesis)

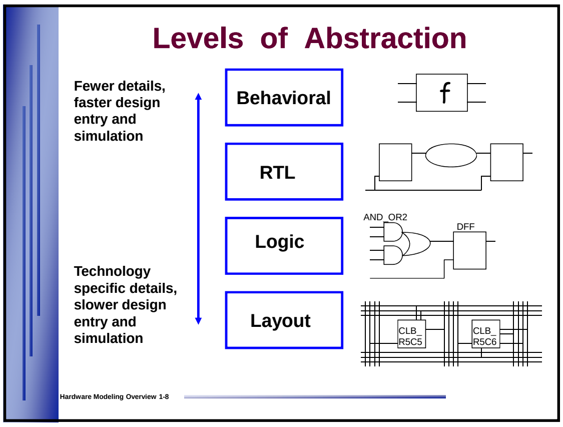
Tool & Vendor Specific

IEEE 1076 (modeling)

N P Z

"Non Portability Zone"

Avoid extensive use of compiler specific constructs that are outside of standard VHDL



Levels of Abstraction

Fewer details, faster design entry and simulation

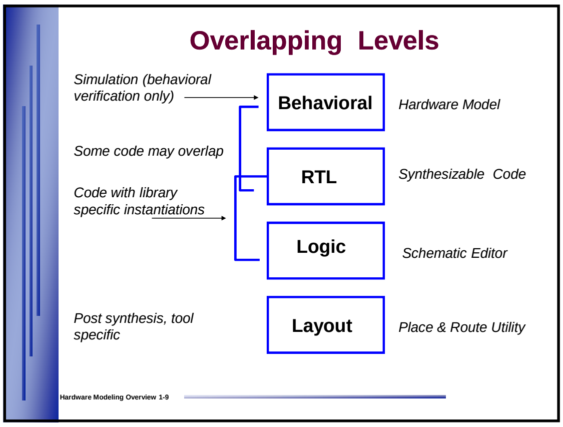
Technology specific details, slower design entry and simulation

Behavioral

RTL

Logic

Layout



Overlapping Levels

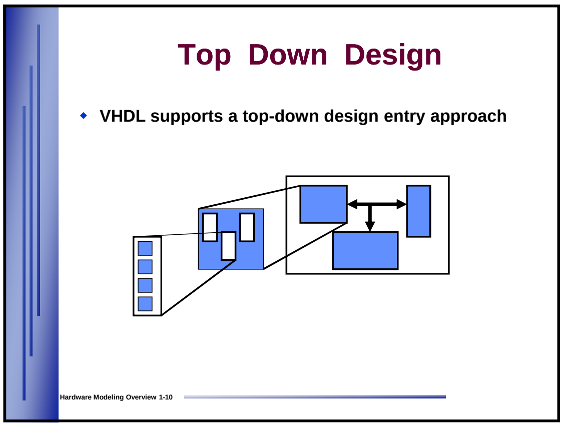
Simulation (behavioral verification only)

Some code may overlap

Code with library specific instantiations

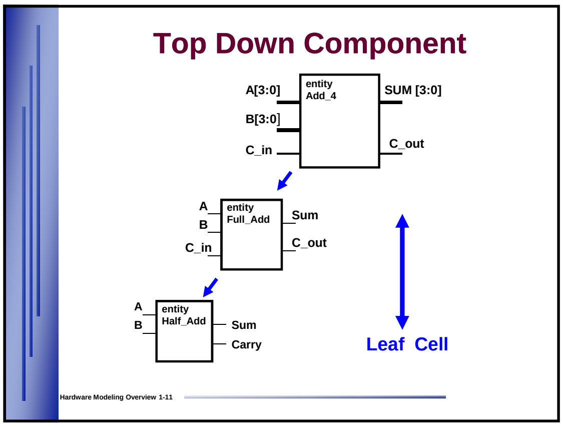
Post synthesis, tool specific

Behavioral	*Hardware Model*
RTL	*Synthesizable Code*
Logic	*Schematic Editor*
Layout	*Place & Route Utility*



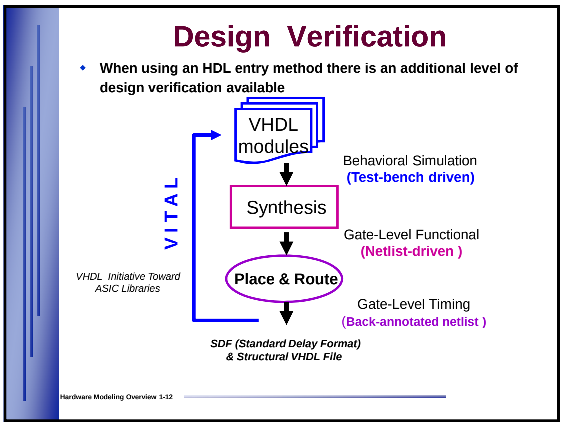
Top Down Design

- VHDL supports a top-down design entry approach



Top Down Component

Leaf Cell



Design Verification

- When using an HDL entry method there is an additional level of design verification available

VHDL modules

Behavioral Simulation (Test-bench driven)

Synthesis

Gate-Level Functional (Netlist-driven)

Place & Route

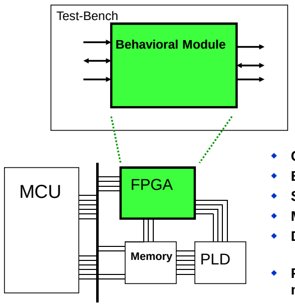
Gate-Level Timing (Back-annotated netlist)

V I T A L

VHDL Initiative Toward ASIC Libraries

SDF (Standard Delay Format) & Structural VHDL File

Why Verify? (Behaviorally)



- Chip Level
- Board Level
- Std Parts Model
- Model Bus Operation
- Discrete Event Driven

- Flexibility over strictly net-list driven

Review Questions

- What does the “V” in VHDL stand for?
- What are four generally recognized ***levels of abstraction***?
- What is the difference between inferring and instantiating?
- In using an HDL design entry, what are three possible stages of design verification?
- What is the benefit of VITAL compliant tools?

Answers

- **What does the “V” in VHDL stand for?**
  - — *VHSIC; Very High Speed Integrated Circuit*
- **What are four generally recognized levels of abstraction?**
  - — *Behavioral, RTL, Logic, Layout*
- **What is the difference between inferring and instantiating?**
  - — *Inference describes only the intended functionality, Instantiation declares the exact component usage*
- **In using an HDL design entry, what are three possible stages of design verification?**
  - — *Behavioral, Gate-Level Functional, Gate-Level w/ Timing, Behavioral w/ Timing*
- **What is the benefit of VITAL compliant tools?**
  - — *Back-annotation of post layout delays in the behavioral verification and it saves time*

Summary

- VHDL is a language for hardware modeling
- Logic synthesis is a subset of the total language
- Tool and technology issues will affect synthesis
- Technology specific resources may require instantiation
- A behavioral model can include back-annotated timing data