CS/EE 3710: Computer Design Laboratory

Lecture 1: Introduction to VHDL and Simulation

(Chapters 1 and 2: Yalamanchili)

Fall Semester 1999

Chris Myers

---------

VHDL

- VHSIC Hardware Description Language.
- Developed in early 80s from the Very High Speed IC program sponsored by the DoD.

---------

What’s its Purpose?

- Originally, a standard, technology- and vendor-independent way to document designs.
- Developed into a language for building simulation models.
- Now frequently used as the input for synthesis.

---------

What is VHDL?

- It’s a documentation system
  - But it’s definitely not just text!
- It’s a programming language
  - But it will definitely confuse you if you treat it like C!
- It’s a design-capture system
  - But you’re missing the point if you think it’s just schematic capture without the graphics!
- It’s a large and complex language
  - But can start simple and add advanced features later.
VHDL vs. Typical Programming Language

* Time is explicit
* Event scheduling
* Concurrent statement evaluation
* Simulation environment
* Component instantiation
* Multiple behaviors for an entity
* Very strong type checking

VHDL vs. Schematics

* Behavioral descriptions
* Multiple representations at different levels
* Potential for synthesis
* Potential for much faster RT-level simulation
* Incremental refinement of program into circuit
* Schematics depict block diagrams well, so many VHDL tools combine schematics and VHDL.

VHDL vs. Proprietary Languages

* PALASM, ABEL, CUPL, and Altera’s AHDL were developed for PLD design.
* Advantages:
  * Simple feature sets and easier to master.
  * Low cost for tools.
* Problems:
  * Not portable,
  * Limited selection of design tools, and
  * Typically cannot express the test environment.
* VHDL beginning to replace these languages.

VHDL vs. Verilog

* Verilog:
  * Wider availability of simulation models.
  * Has a programming language interface.
* VHDL:
  * IEEE standard.
  * Has higher-level design management features.
* Mostly the same and easy to transition between.
* “Arguing about language features is pointless and best left to late night sessions at your favorite watering hole.”
How is VHDL Used?

- Specify performance and interface requirements.
- Capture block level design schematics.
- Validate using functional and timing simulation.
- Document large designs.
- Implement design with synthesis.

Why Should You Use VHDL?

- “It will improve your productivity.”
- Enhances communication between designers.
- There are powerful tools for simulation.
- Allows design at a more abstract level than gates.
- Easy to build and use libraries of commonly-used VHDL modules.
- Eases portability to new tools and technologies.

Why Should You Not Use VHDL?

- Steep learning curve?
  - Getting started is easy.
- Scarcity of low-cost tools?
  - Recently lower cost tools have become available.
- Lack of technology-specific features?
  - Developers have published conventions for synthesis.
- Lack of VHDL applications expertise?
  - PLD and FPGA vendors are adopting VHDL.
Describing Systems

- System - “an assemblage of objects united by some form of regular interaction or interdependence.”
- What do engineers need to know to evaluate a design?
  - The system’s interface.
  - A structural description - i.e., chips and interconnections.
  - A behavioral description - i.e., what a system should do.
- VHDL allows both structural & behavioral descriptions

Signals, Events, and Components

- Digital systems are fundamentally about binary signals.
- Changes of signal values are called events.
- Digital circuits are composed of components (i.e. gates).
- Components are interconnected by wires and transform events on input signals to events on output signals.

Half-adder Circuit

FIGURE 2.2

Propagation Delay

Concurrency

Signal Values -IEEE Standard 1164

- Introduced in late 80s to create a standard nine-valued type to accurately model digital behavior.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'U'</td>
<td>Uninitialized</td>
</tr>
<tr>
<td>'X'</td>
<td>Forcing Unknown</td>
</tr>
<tr>
<td>'0'</td>
<td>Forcing 0</td>
</tr>
<tr>
<td>'1'</td>
<td>Forcing 1</td>
</tr>
<tr>
<td>'Z'</td>
<td>High impedance</td>
</tr>
<tr>
<td>'W'</td>
<td>Weak unknown</td>
</tr>
<tr>
<td>'L'</td>
<td>Weak 0</td>
</tr>
<tr>
<td>'H'</td>
<td>Weak 1</td>
</tr>
<tr>
<td>'D'</td>
<td>Don't care</td>
</tr>
</tbody>
</table>
Discrete Event Simulation Model

- Advance simulation time to that of the event with the smallest timestamp in the event list.
- Execute all events at this timestep.
- Execute the simulation models of all components affected by the new signal values.
- Schedule any future events.
- Repeat until the event list is empty.

Summary

- VHDL can be used to model digital systems at multiple levels of abstraction.
- Descriptions can be both structural and behavioral.
- The behavior of a digital system can be described by events which cause signals to change value.
- A discrete event simulation can be used to analyze a model of a digital system.
- NEXT: Basic language constructs (chapter 3).