Hardware Description Languages

Modeling Digital Systems
HDL coding Styles

- Register Transfer Level
- Structural
- Behavioral

Be careful NOT everybody gives the same meaning to the term BEHAVIORAL!
HDL applications

- High Level Modeling (Behavioral style)
- Design Entry (Structural & RTL styles)
- Simulation (Behavioral style)
  - Validation by mean of a test bench

![Diagram showing the process of HDL applications]

- generate stimuli
- instantiate design to test
- observe responses

dut_tb.vhd

dut.vhd

TESTBENCH
Levels of Abstraction

- Behavioral
- RTL
- Structural
Behavioral Level

- Describe behavior (functionality and performances)
- All language features can be used
• Only a small subset of the Language statements can be mapped into “Silicon”.

SYNTHESIS

HDL code

generic technology

unoptimized generic boolean netlist

optimization & mapping

target technology

optimized gate level netlist

X <= (A AND B) OR X1;
X1 <= (X3 XOR G)
X2 <= (E OR NOT F)
X3 <= (C AND X2)
Structural Level

- Sub-Modules interconnection
- Primitive cells interconnection (net-list)
- The code describes a bunch of port mappings.
Describing Systems

- What aspects do we need to consider to describe a digital system?
  - Interface
  - Function
  - Performance (delay/area/costs/…)

What elements should be in a VHDL description? (1)

- VHDL was conceived for the description of digital systems
- Keeping in mind the pragmatic issues of design re-use and portability of descriptions
  - Portability across technologies
- Attributes of digital systems served as the starting point
  - Language features were designed to capture the key attributes
What elements should be in a VHDL description? (2)

- Descriptions should support multiple levels of abstraction
- The elements should enable meaningful and accurate simulation of hardware described using the elements
  - Elements should have attributes of time as well as function
- The elements should enable the generation of hardware elements that realize a correct physical implementation
  - Existence of a mapping from elements to VLSI devices
Attributes of Digital Systems

Digital systems are about:

- **signals and their values**
- **events, propagation delays, concurrency**
- Time ordered sequence of events produces a waveform
Attributes of Digital Systems: Timing

- **Timing**: computation of events takes place at specific points in time
- Need to **wait for** an event: in this case the clock
- Timing is an attribute of both synchronous and asynchronous systems
Asynchronous communication does not have a global clock
Still need to wait for events on specific signals
Attributes of Digital Systems: Signal Values

- We associate logical values with the state of a signal
- Signal Values: IEEE 1164 Value System

<table>
<thead>
<tr>
<th>Value</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>Uninitialized</td>
</tr>
<tr>
<td>X</td>
<td>Strong Unknown</td>
</tr>
<tr>
<td>0</td>
<td>Strong 0</td>
</tr>
<tr>
<td>1</td>
<td>Strong 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>-</td>
<td>Don’t Care</td>
</tr>
</tbody>
</table>
Attributes of Digital Systems: Multiple Drivers

- **Shared Signals**
  - multiple drivers

- **How is the value of the signal determined?**
  - arbitration protocols
  - wired logic *(not recommended)*
Tristate Inverter

(a) $\overline{EN}$

(b) $EN = 0$

(c) $EN = 1$

(d) $Y = \overline{A}$

FIG 1.26 Tristate inverter
Modeling Digital Systems

- We seek to describe attributes of digital systems common to multiple levels of abstraction
  - events, propagation delays, concurrency
  - waveforms and timing
  - signal values
  - shared signals
Modeling Digital Systems

- Hardware description languages must provide constructs for naturally describing these attributes of a specific design
  - simulators use such descriptions for “mimicking” the physical system
  - synthesis compilers use such descriptions for synthesizing manufacturable hardware specifications that conform to this description
Execution of VHDL models

- For Simulation
  - Discrete event simulator

- For Synthesis
  - Hardware inference
  - The resulting circuit is a function of the building blocks used for implementation and the optimization goal
    - Primitives: e.g. NAND vs. NOR
    - Cost/performance objectives
Simulation of Digital Systems

- Digital systems are modeled as the generation of events (= value transitions) on signals.
- Discrete event simulations manage the generation and ordering of events.
  - Correct sequencing of event processing.
  - Correct sequencing of computations caused by events.
**Discrete Event Simulation: Example**

**Simulation Time**
- **0ns**
- **5ns**
- **10ns**

**Event List Head**
- **U→1**
  - **carry@5ns**
  - **sum@5ns**
- **1→0**
  - **a@5ns**

**Initial state**: a = b = 1, sum = carry = U

**Update event generated from input**
- New event generated from input
- Update signal values, execute, generate new events, update time

**Update signal values, execute, generate new events**
- a@15ns
Discrete Event Simulation

- Management of simulation time: ordering of events

- Two step model of the progression of time
  - Evaluate all affected components at the current time: events on input signals
  - Schedule future events and move to the next time step: the next time at which events take place
Simulation Modeling

- VHDL programs describe the generation of events in digital systems
- Discrete event simulator manages event ordering and progression of time
- Accuracy vs. time trade-offs
  - Greater detail → more events → greater accuracy
  - Less detail → smaller number of events → faster simulation speed
Synthesis and Hardware Inference

Design Specification → Synthesis compiler → HDL
Summary

• VHDL is used to describe digital systems and hence has language constructs for the following key attributes
  • Events, propagation delays, and concurrency
  • Timing, and waveforms
  • Signal values and use of multiple drivers for a signal
Summary

- VHDL has an underlying discrete event simulation model
  - Model the generation of events on signals
  - Built in mechanisms for managing events and the progression of time
  - Designer simply focuses on writing accurate descriptions
VHDL Design Organization

- **Entity**
  the “symbol” interface (input/output ports)

- **Architecture**
  one of the several possible implementation of the design

- **Configuration**
  binding between the symbol and one of the many possible implementation. Can be used to express hierarchy.