Textbook: Verilog® HDL 2nd. Edition
Samir Palnitkar
Prentice-Hall, Inc.

INSTRUCTOR: CHING-LUNG SU
E-mail: kevinsu@yuntech.edu.tw
Course Web Page:
http://soc.eecs.yuntech.edu.tw
Chapter 1
Overview of Digital Design with Verilog HDL
Outline of Chapter 1

1.1 Evolution of Computer-Aided Digital Design
1.2 Emergency of HDLs
1.3 Typical Design Flow
1.4 Importance of HDLs
1.5 Popularity of Verilog HDL
1.6 Trends in HDLs
1.1 Evolution of Computer-Aided Digital Design

1.2 Emergency of HDLs

1.3 Typical Design Flow

1.4 Importance of HDLs

1.5 Popularity of Verilog HDL

1.6 Trends in HDLs
**Design Complexity for Digital Circuits**

- Vacuum Tubes
- Transistors
- SSI (Small Scale Integration) Chips: Tens of Gates
- MSI (Medium Scale Integration) Chips: Hundreds of Gates
- LSI (Large Scale Integration) Chips: Thousands of Gates
- VLSI (Very Large Scale Integration) Chips: > 100,000 Gates
- ULSI (Ultra Large Scale Integration) Chips: > VLSI
1.1 Evolution of Computer-Aided Digital Design

- **EDA and CAD**
  - EDA: Electronic Design Automation
  - Designers use the EDA circuit and logic simulation technology to verify the designs
  - CAD: Computer-Aided Design
  - CAD Tools: Automatic placement and routing of circuit layout
1.1 Evolution of Computer-Aided Digital Design
1.2 Emergency of HDLs
1.3 Typical Design Flow
1.4 Importance of HDLs
1.5 Popularity of Verilog HDL
1.6 Trends in HDLs
1.2 Emergency of HDLs

- **Hardware Description Language (HDL)**
  - Capacity of HDL: Model the concurrency of processes for hardware elements
  - HDL: Verilog® HDL and VHDL
  - Verilog HDL: Originated in 1983 at Gateway Design Automation
  - RTL: Register Transfer Language
  - RTL Description: Gate level circuits can be automatically generated by synthesis tools from the RTL HDL.
1.3 Typical Design Flow

1.1 Evolution of Computer-Aided Digital Design
1.2 Emergency of HDLs
1.3 Typical Design Flow
1.4 Importance of HDLs
1.5 Popularity of Verilog HDL
1.6 Trends in HDLs
1.3 Typical Design Flow

Typical Design Flow

- Design Specification
- Behavior Description
- RTL Description (HDL)
- Functional Verification and Testing
- Logic Synthesis / Timing Verification
- Gate-Level Netlist
- Logical Verification and Testing
- Floor Planning Automatic Place & Route
- Physical Layout
- Layout Verification
- Implementation

: Design Process
: Description Level
1.4 Importance of HDLs

1.1 Evolution of Computer-Aided Digital Design
1.2 Emergency of HDLs
1.3 Typical Design Flow
1.4 Importance of HDLs
1.5 Popularity of Verilog HDL
1.6 Trends in HDLs
1.4 Importance of HDLs

HDL Advantages over Schematic-based Design

- Design can be described at a vary abstract-level HDLs.
- Designers can write the technology independent RTL description.
- Logic Synthesis tools can automatically convert the RTL design to any fabrication technology.
- Logic synthesis tools can optimize the circuit in area and timing for any technology.
- The function verification can be done early in the design cycle.
- With rapidly increasing complexities of digital circuits and increasingly sophisticated EDA tools, HDLs are now dominant method for large digital designs.
1.5 Popularity of Verilog HDL

1.1 Evolution of Computer-Aided Digital Design
1.2 Emergency of HDLs
1.3 Typical Design Flow
1.4 Importance of HDLs
1.5 Popularity of Verilog HDL
1.6 Trends in HDLs
### Features of Verilog HDL

- Verilog HDL is a general-purpose HDL that is easy to learn and easy to use.
- Verilog HDL allows different levels of abstraction to be mixed in the same model (Switch/ Gate/ RTL/ Behavior level).
- Most logic synthesis tools support Verilog HDL.
- All fabrication vendors provide verilog HDL libraries for postlogic synthesis simulation.
- The programming language interface (PLI) is a powerful feature that allows the user to write custom C code to interact with the internal data structures of Verilog.
1.6 Trends in HDLs

1.1 Evolution of Computer-Aided Digital Design
1.2 Emergency of HDLs
1.3 Typical Design Flow
1.4 Importance of HDLs
1.5 Popularity of Verilog HDL
1.6 Trends in HDLs
1.6 Trends in HDLs

- **Trends in HDL**

- Formal Verification: Formal verification applies formal mathematical techniques to verify the correctness of Verilog HDL and to establish equivalency between RTL and gate-level netlists.

- Assertion checkers allow checking to be embedded in the RTL code.

- Mixed level HDL description for complexity systems.

- For system-level design, the behavior can speedup the simulation time.

- The next generation system description languages: SystemC, System Verilog, Verilog 2000, ...