EC3057D Modeling and Testing of Digital Systems

Winter 2021

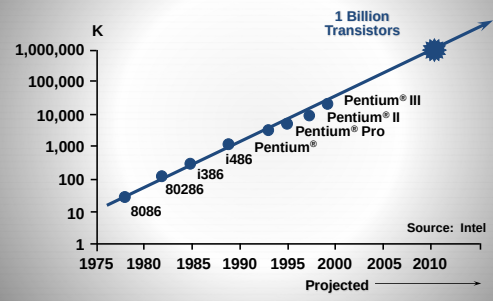
Reference books

- Ciletti M.D., *Advanced digital design with the Verilog HDL*, Second Edition, Prentice Hall, 2010.
- Palnitkar S.,*Verilog HDL: A guide to digital design and synthesis,* Prentice Hall; 2003.
- Charles Roth, Lizy Kurian John, ByeongKil Lee, *Digital systems design using Verilog*, First Edition,Cengage Learning, 2014.
- J. Bhasker, *Verilog HDL Synthesis: A Practical Primer,* B. S. Publications, 2001.
- Bushnell M, Agrawal V., *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits,* Springer Science & Business Media, 2004.

Introduction

- **Moore's Law**
  - In 1965, Gordon Moore noted that the number of transistors on a chip doubled every 18 to 24 months.

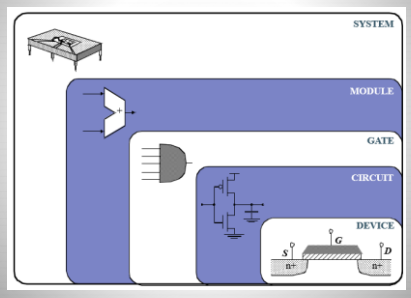
Transistor count



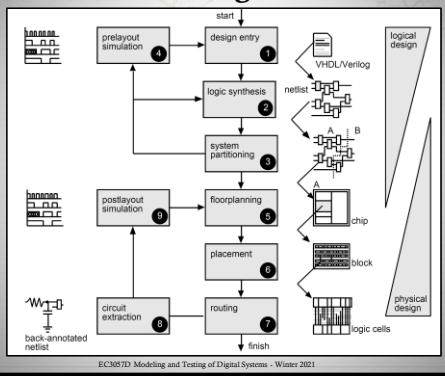
Computer-Aided Design

- Every new generation can integrate 2x more functions per chip
  - Chip price does not increase significantly
  - Cost of a function decreases by 2x
- However,
  - Design engineering population does not double every two years.
  - How to design much more complex chips (with more and more functions)?
- Great need for ultra-fast design methods
  - Design Automation (Computer-Aided Design)

Design Abstraction Enables CAD



VLSI Design Flow



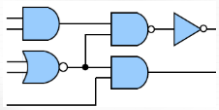
Hardware Description Languages

Evolution of Computer Aided Digital Design

- Digital circuits were designed with vacuum tubes and Transistors
- ICs were then invented
  - SSI, MSI, LSI, VLSI
- Due to complexity of circuits it was not possible to verify circuits on board.
- CAD do verification and design of VLSI circuits, also do automatic placement and routing of circuit layouts.
- Logic simulators came into existence to verify the functionality of these circuits before they were fabricated on Chip

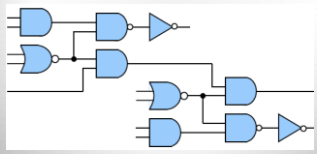
Hardware Description Languages - Introduction

- In the beginning, designs had only a few gates, and thus it was possible to verify these circuits on paper or with breadboards.



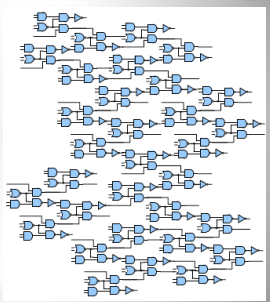
HDL – Introduction (Cont..)

- As designs grew larger and more complex, verification using paper or breadboards became impossible.
- Hence, designers began to use gate-level models described in a Hardware Description Language to help with verification before fabrication



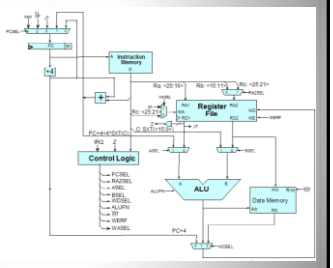
HDL – Introduction (Cont..)

- When the number of gates in the designs are in the ranges of 100,000 gate designs, these gate-level models also became complex for the functional specification.



HDL – Introduction (Cont..)

- Designers again turned to HDLs for help – abstract behavioural models written in an HDL provided both a precise specification and a framework for design exploration.



HDL – Introduction (Cont..)

- Better be **standard** than be proprietary.
- Even though HDLs were popular for logic verification, designers had to manually translate the HDL-based design into a schematic circuit with interconnections between gates.
- Digital circuits could be described at a register transfer level (RTL) by use of an HDL.
- The designer had to specify how the data flows between registers and how the design processes the data.
- The details of gates and their interconnections to implement the circuit were automatically extracted by logic synthesis tools from the RTL description.

HDL - Introduction

- HDLs also began to be used for system-level design.
- HDLs were used for simulation of system boards, interconnect buses, FPGAs (Field Programmable Gate Arrays), and PALs (Programmable Array Logic).
- Can describe a design at some levels of abstraction
- Can be used to document the complete system design tasks
  - Testing, simulation, ..., related activities
- Comprehensive and easy to learn
- Most popular logic synthesis tools support verilog HDL
- All fabrication vendors provide Verilog HDL libraries.

What is Verilog HDL?

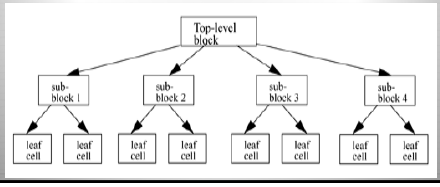
- Invented by Philip Moorby in 1983/84.
- The original standard was IEEE 1364
  - The first version was published in 1995.
  - Revised in 2001 and 2005.
- Allows different levels of abstraction to be mixed in the same design.
- Single language for design and testbench.

Contd..

- Simulated to check functionality.
- Synthesized (netlist generated).
- Built-in primitives, logic function
- User-defined primitives
- Built-in data types
- High-level programming constructs

Hierarchy of design methodologies

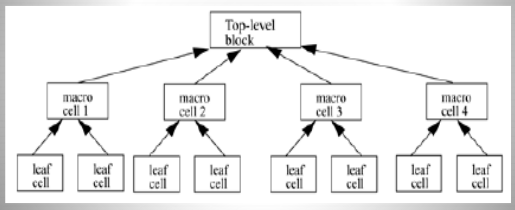
- **Top-Down Design**
  - This style is convenient and efficient.
  - Start with system specification
  - Decompose into subsystems, components, until indivisible
  - Realize the components
  - But it is very difficult to follow a pure top-down design. Hence, most designs are mix of both the methods.



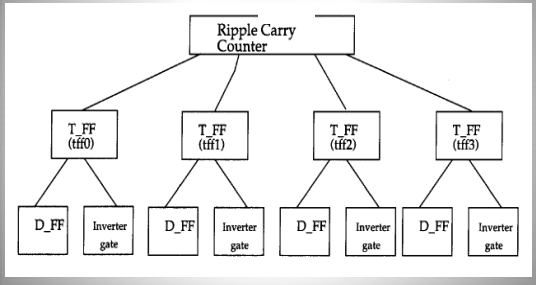
Hierarchy of design methodologies

- **Bottom-Up Design**
  - The traditional method of electronic design is bottom-up (designing from transistors and moving to a higher level of gates and, finally, the system).
  - But with the increase in design complexity traditional bottom-up designs have to give way to new structural, hierarchical design methods.
  - Start with available building blocks
  - Interconnect building blocks into subsystems, then system
  - Achieve a system with desired specification

- Bottom – up Design



Hierrarchy



Levels of Abstraction

- **Behavioral Level :**Module can be implemented in terms of the desired design algorithm without concern for the hardware implementation details. Very similar to C programming
- **Dataflow Level**: Module designed by specifying dataflow. The designer is aware of how data flows between hardware registers and how the data is processed in the design
- **Gate Level:** Module implemented in terms of logic gates like (and ,or) and interconnection between gates
- **Switch Level:** Module implemented with switches and interconnects. Lowest level of Abstraction