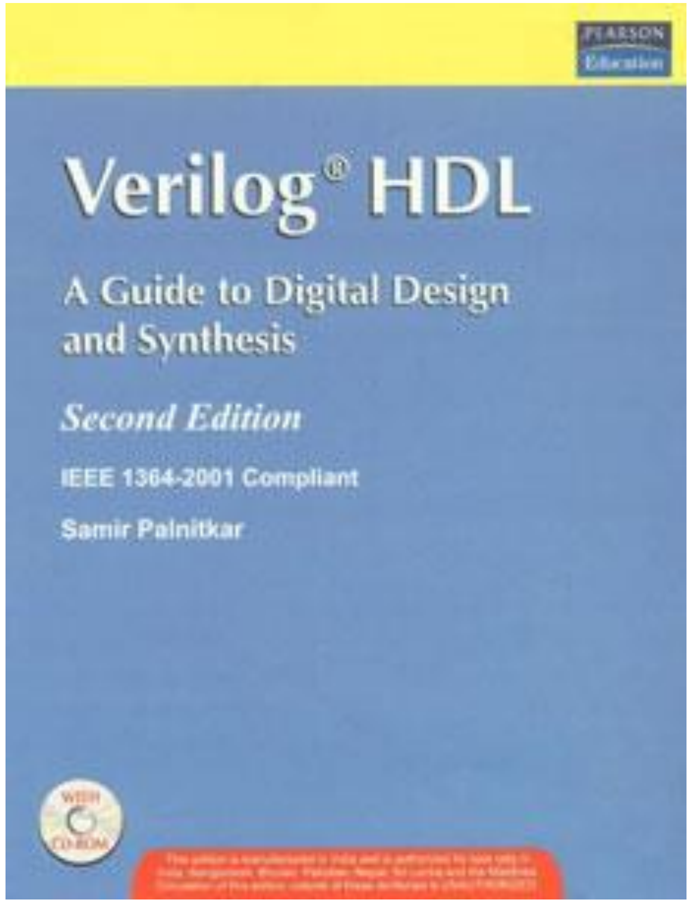
# Verilog HDL

OVERVIEW OF DIGITAL DESIGN WITH VERILOG HDL

# Reference

Verilog HDL: A Guide to Digital Design and Synthesis, 2e, Samir Palnitkar

**Chapters 1**



PEARSON Education

Verilog® HDL

A Guide to Digital Design and Synthesis

*Second Edition*

IEEE 1364-2001 Compliant

Samir Palnitkar

WITH CD-ROM

## Evolution of Computer-Aided Digital Design

Small circuits with just hundreds of transistors :

- Design the layout on paper or by hand on a graphics computer terminal.
- Test on a breadboard.

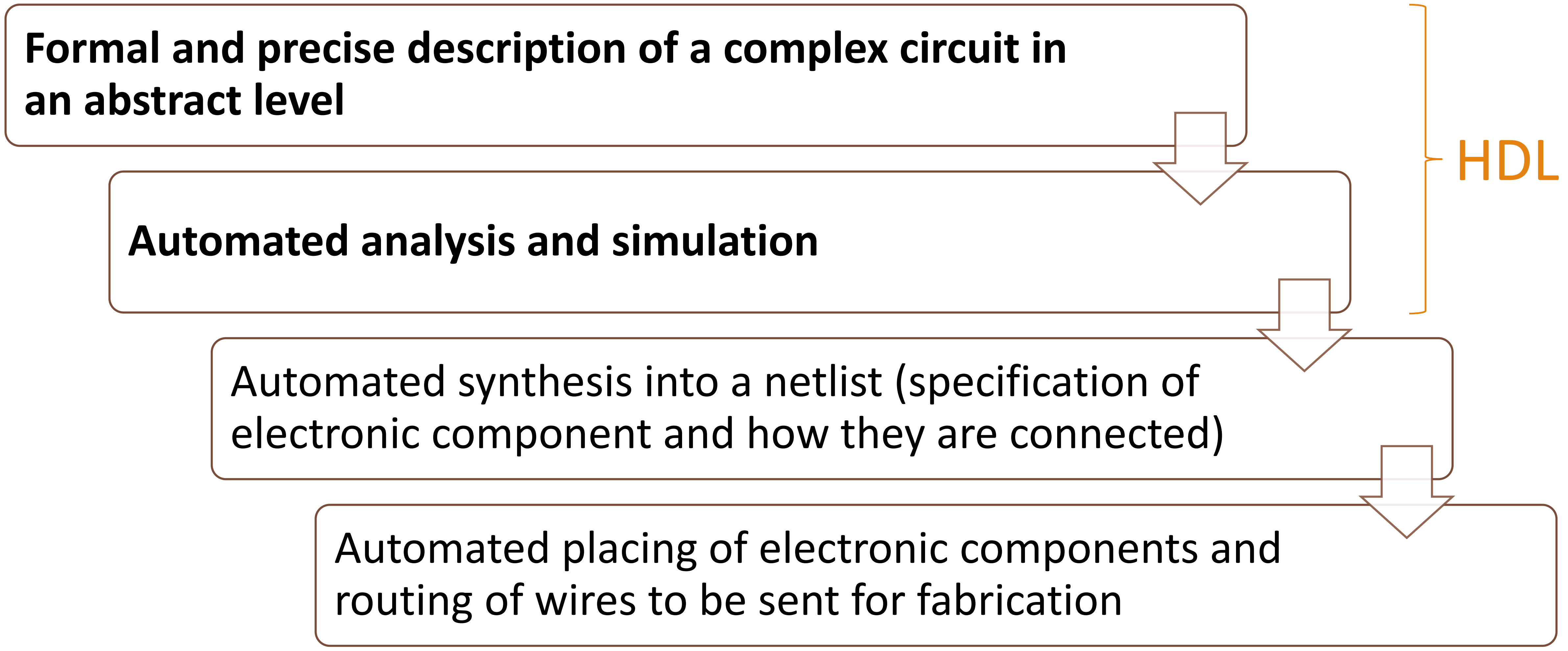
But a modern processors has millions of transistors (eg : Intel i7 has 731,000,000 transistors) :

- Computer aided techniques for design and verification is required.

## Emergence of HDL

- Programming languages like C describe computer programs which are sequential in nature.
- But digital circuits involves concurrency -> general programming languages doesn’t suit.
- Languages that describe digital circuits called Hardware Description Languages (HDL) came into existence.

# Integrated Circuit Design Processes



**Formal and precise description of a complex circuit in an abstract level**

**Automated analysis and simulation**

Automated synthesis into a netlist (specification of electronic component and how they are connected)

Automated placing of electronic components and routing of wires to be sent for fabrication

# Hardware Description Languages (HDL)

Example HDL :

- **Verilog HDL**
- **VHDL**

$\left.\begin{matrix}\\ \\ \end{matrix}\right\}$ Widely used

- AHDL
- AHPL
- Bluespec

## Verilog HDL - History

- Verilog HDL invented by Philip Moorby in 1983 at Gateway Design Automation.
- Verilog- based synthesis tool introduced by Synopsys in 1987
- Gateway Design Automation bought by Cadence in 1989
- Verilog placed in public domain to compete with VHDL
    - Open Verilog International (OVI) IEEE 1364 -1995 and
    - revised version IEEE 1364 -2001
    - revised version IEEE 1364 -2005

# Verilog HDL

- Easy to use : similar to syntax of C programming language
- Mixed level modelling
    - Behavioral - Algorithmic (like high level language)
    - Data-flow - Register transfer (synthesizable)
    - Gate-level - Structural (AND, OR ……)
- Single language for design and simulation
- Built-in primitives, logic functions and data types
- User-defined primitives
- Built-in High-level programming constructs