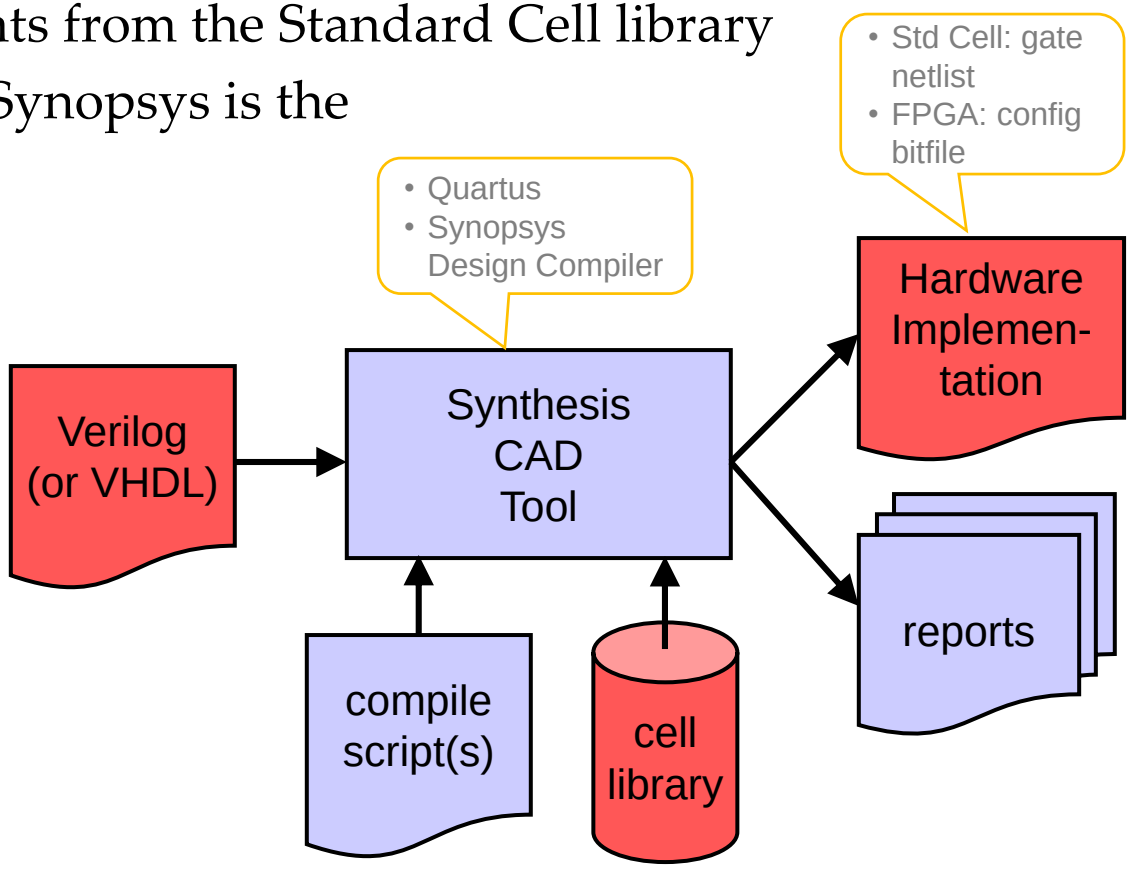
FROM THE HARDWARE DESCRIPTION LANGUAGE (HDL) DESIGN TO HARDWARE

Typical *Standard Cell, Gate Array, or FPGA* Design Flow

- Involves synthesizing Verilog source code to generate a *gate netlist* made up of elements from the Standard Cell library
- *Design Compiler* (DC) by Synopsys is the most popular synthesis tool used in industry
- The same Verilog design could be synthesized to various libraries; for example:
  - Standard cell (NAND, NOR, Flip-Flop, etc.)
  - FPGA library (CLBs, LUTs, etc.)



Synthesis Cell Library

- The Cell Library contains the basic building blocks of the final design ("hardware implementation")
- Standard Cell Design
  - AND, OR, INV, XOR, Flip-Flop, AOI, Full-Adder, etc.
  - Many sizes/strengths of each—e.g., X1, X2, X4, X8
- FPGA
  - LUTs
  - Available structures such as multipliers, DSP slices, block memories, etc.

HDL → Hardware Implementation

- Standard Cell Design
  - Hardware is described by a gate netlist made up of gates from the cell library, plus interconnection specifications
- FPGA
  - Hardware is described by a configuration *bitfile* that specifies how configurable elements from the cell library (LUTs) are configured, plus interconnection specifications
- In both cases, digital hardware to be implemented on a chip has been designed by an HDL description