Generalized ASIC Design Flow

- **High Level Design**
  - Specification Capture
  - Design Capture in C, C++, SystemC or SystemVerilog
  - HW/SW partitioning and IP selection

- **RTL Design**
  - Verilog/VHDL

- **System, Timing and Logic Verification**
  - Is the logic working correctly?

- **Physical Design**
  - Floorplanning, Place and Route, Clock insertion

- **Performance and Manufacturability Verification**
  - Extraction of Physical View
  - Verification of timing and signal integrity
  - Design Rule Checking/LVS
RTL Synthesis Flow

1. Specification
2. RTL Behavioral Description
3. Verification by Description
   - y: OK
   - n: go back
4. RTL Synthesis
5. Library Data
6. Library Mapping
7. Netlist
8. Simulation or Formal Verification
   - y: go back to library mapping
   - n: debug
9. Floorplan and Library Data
10. Timing Analysis
11. Power Analysis
   - y: debug
   - n: go back to power analysis
12. To Physical Synthesis

Logic Design and Verification

- **Design starts with a specification**
  - Text description or system specification language
    - Example: C, SystemC, SystemVerilog

- **RTL Description**
  - Automated conversion from system specification to RTL possible
    - Example: Cadence C-to-Silicon Compiler
  - Most often designer manually converts to Verilog or VHDL

- **Verification**
  - Generate test-benches and run simulations to verify functionality
  - Assertion based verification
  - Automated test-bench generation
ASIC Design Flow

RTL Synthesis and Verification

- **RTL Synthesis**
  - Automated generation of generic gate description from RTL description
  - Logic optimization for speed and area
  - State machine decomposition, datapath optimization, power optimization
  - Modern tools integrate global place-and-route capabilities

- **Library Mapping**
  - Translates a generic gate level description to a netlist using a target library

- **Functional or Formal Verification**
  - HDL ambiguities can cause the synthesis tool to produce incorrect netlist
  - Rerun functional verification on the gate level netlist
  - Formal verification
    - Model checking: prove that certain assertions are true
    - Equivalence checking: compare two design descriptions

Static Timing Analysis

- Checks temporal requirements of the design
- Uses intrinsic gate delay information and estimated routing loads to exhaustively evaluate all timing paths
- Requires timing information for any macro-blocks e.g. memories
- Will evaluate set-up and hold-time violations
- Special cases need to be flagged using timing constraints (more later)
- Reports “slack time”
- Re-synthesize the circuit or re-design to improve delay
Test Insertion and Power Analysis

- Insert various DFT features to perform device testing using Automated Test Equipment (ATE) and system level tests
  - Scan enabled flip-flops and scan chains
    - Automatic Test Pattern Generation (ATPG) tools generate test vectors to perform logic and parametric testing
  - Built-in Self Test
    - Logic: Based on LFSR (random-patterns) and MISR (signature) (LBIST)
    - Memory: Implements various memory testing algorithms (MBIST)
  - Boundary-Scan/JTAG
    - Enables board/system level testing
  - More on DFT and test insertion later

- Power Analysis
  - Power analysis tools predict power consumption of the circuit
  - Either test vectors or probabilistic activity factors used for estimation
Standard Cell Place and Route Flow

- **Netlist**
  - Library Description
  - Technology Constraints
  - Technology Parameters

- **Placement**

- **DEF**

- **Route**

- **Parasitic Extract**

- **ESPF**

- **Timing Analysis**
  - Library SDF

- **Timing OK**

- **Noise and Reliability**

- **OK**

- **Database to Manufacturer**

- **Tool error**

Floorplanning /Placement/Routing

- Manually place major modules in the chip depending on connections with other modules
- Standard cell rows are defined next and the gates are placed
  - No routing channels between rows in newer technologies
- Timing driven placement tries to minimize delay on critical paths
- Routing
  - Route special nets
    - Power, Ground
  - Clock tree synthesis/ routing
    - Minimize skew
    - Insert buffers
  - Global and detailed routing of signal nets
Verification Steps

- Parasitic Extraction
  - Detailed parasitic extraction after routing
  - 2D, 2.5D and 3D extraction possible, output is SPEF, RSPF, ESPF etc.

- Timing Analysis
  - Static timing analysis, iterate if there are problems
  - If possible, full chip simulation at transistor-level using “fast spice simulators”
    - E.g. Nanosim, Ultrasim

- Crosstalk, \(V_{DD}\) Drop, Electromigration Analysis

- Power Analysis
  - More detailed wire capacitance available for power analysis, iterate if there are problems
Timing Driven Placement

Library SDF
Technology Constraints DEF
Technology Attributes Wire Capacitance and Resistance
Library LEF

Placement
Routing Engine
Timing Directed Placement Engine
Extract
Timing Analysis
Timing OK
Final Checks
OK
Finished