VLSI Design

Chapter 6

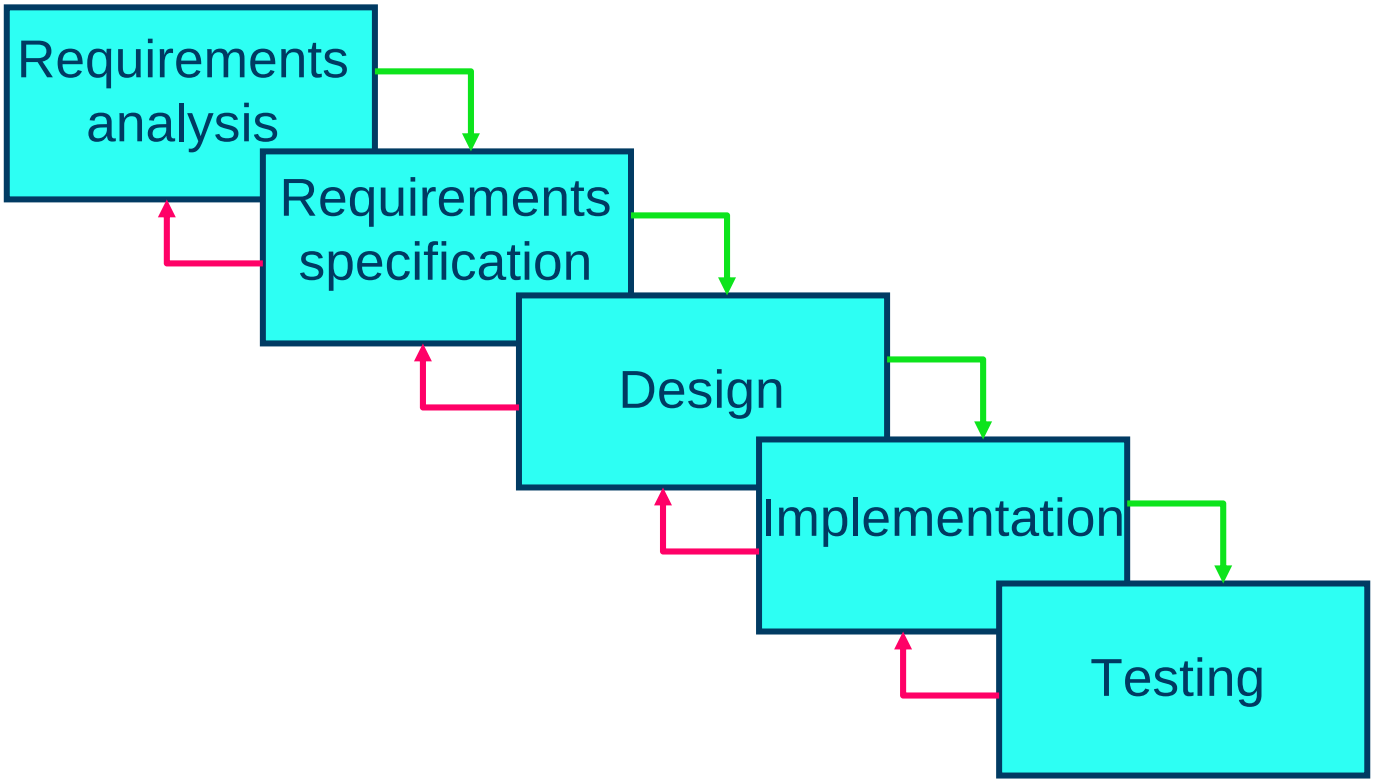
CMOS Design Methods

Jin-Fu Li

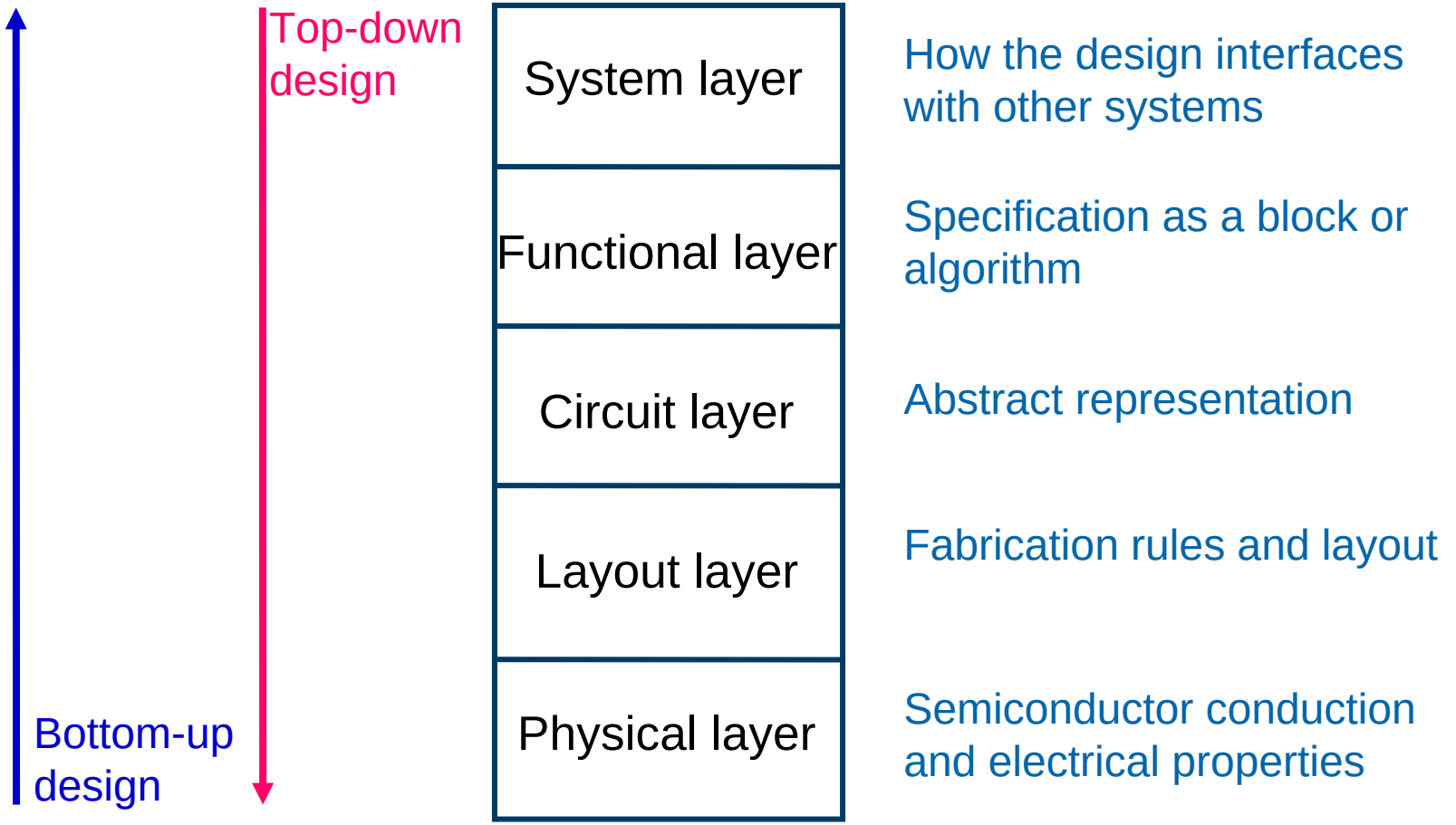
Chapter 6 CMOS Design Methods

- Introduction
- VLSI Design Flows
- Design Strategies
- VLSI Design Styles
- Design Verification

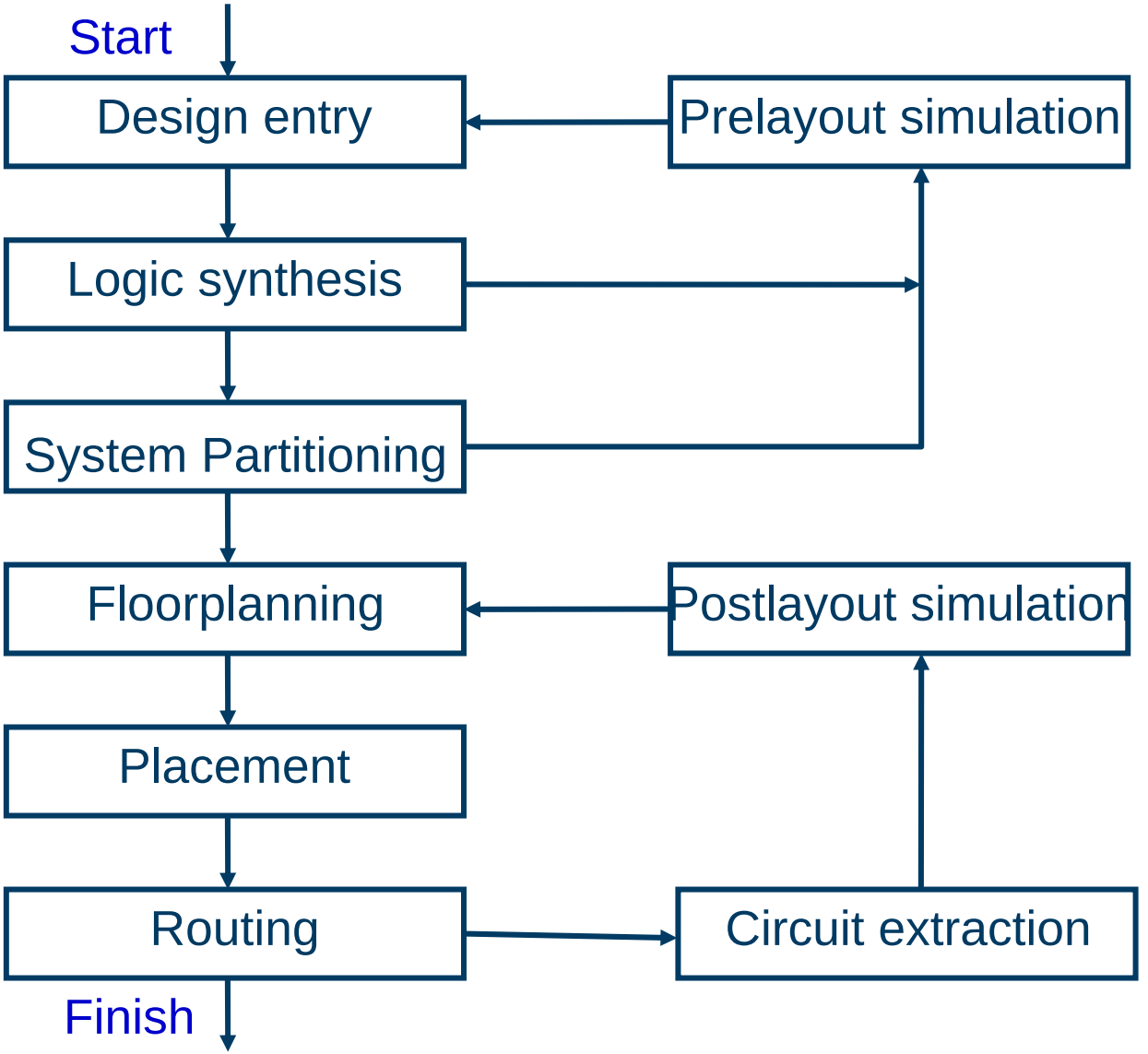
System Development Process



Design Abstraction



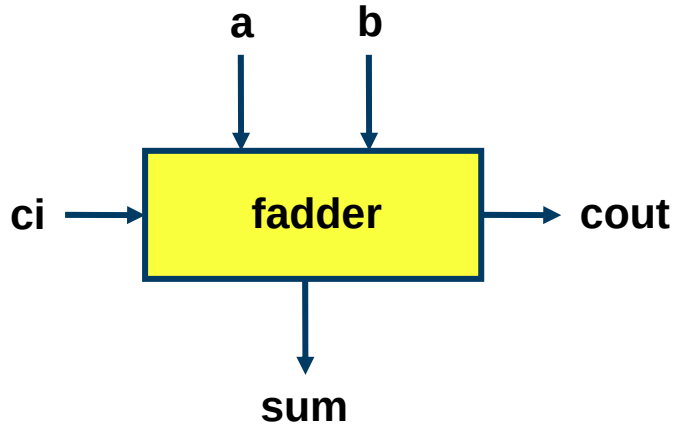
Design Flow – *ASIC Design Flow*



Design Flow – *Design Entry*

- Enter the design into an ASIC design system, either using a *hardware description language* (HDL) or *schematic entry*
- An example of Verilog HDL

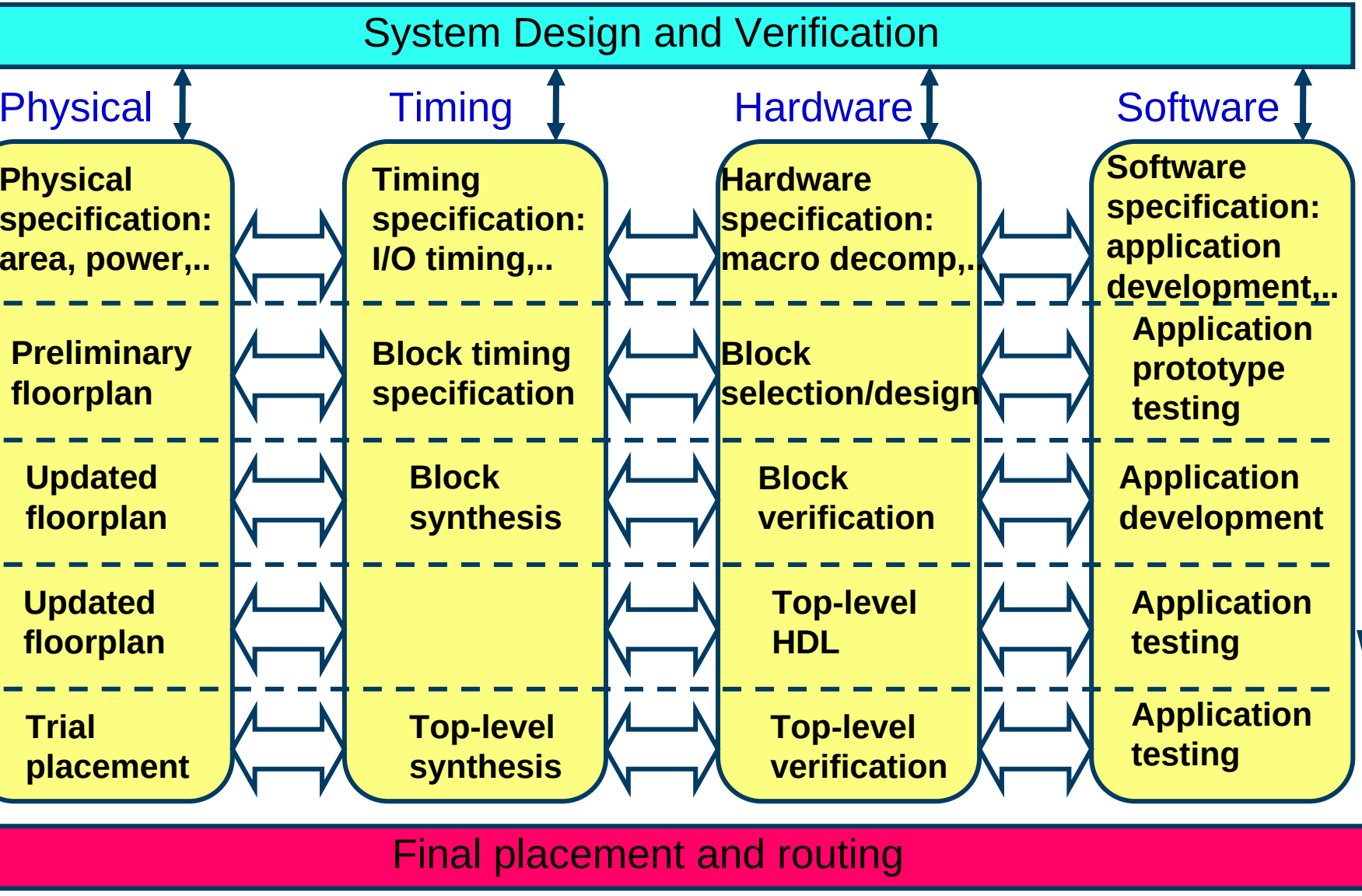
```
module fadder(sum,cout,a,b,ci);
output sum, cout;
input a, b, ci;
reg sum, cout;

always @(a or b or ci) begin
    sum = a^b^ci;
    cout = (a&b)|(b&ci)|(ci&a);
end
endmodule
```



Design Flow – *Features of SOC Design*

- Parallel development of H/W and S/W
- Parallel verification and synthesis of modules
- Floorplaning, Placement, and Routing in synthesis process
- Use predesigned Macros (soft/hard)
  - Intelligent Property (IP)
- Planned iteration throughput

Design Flow – *SOC Design Flow*



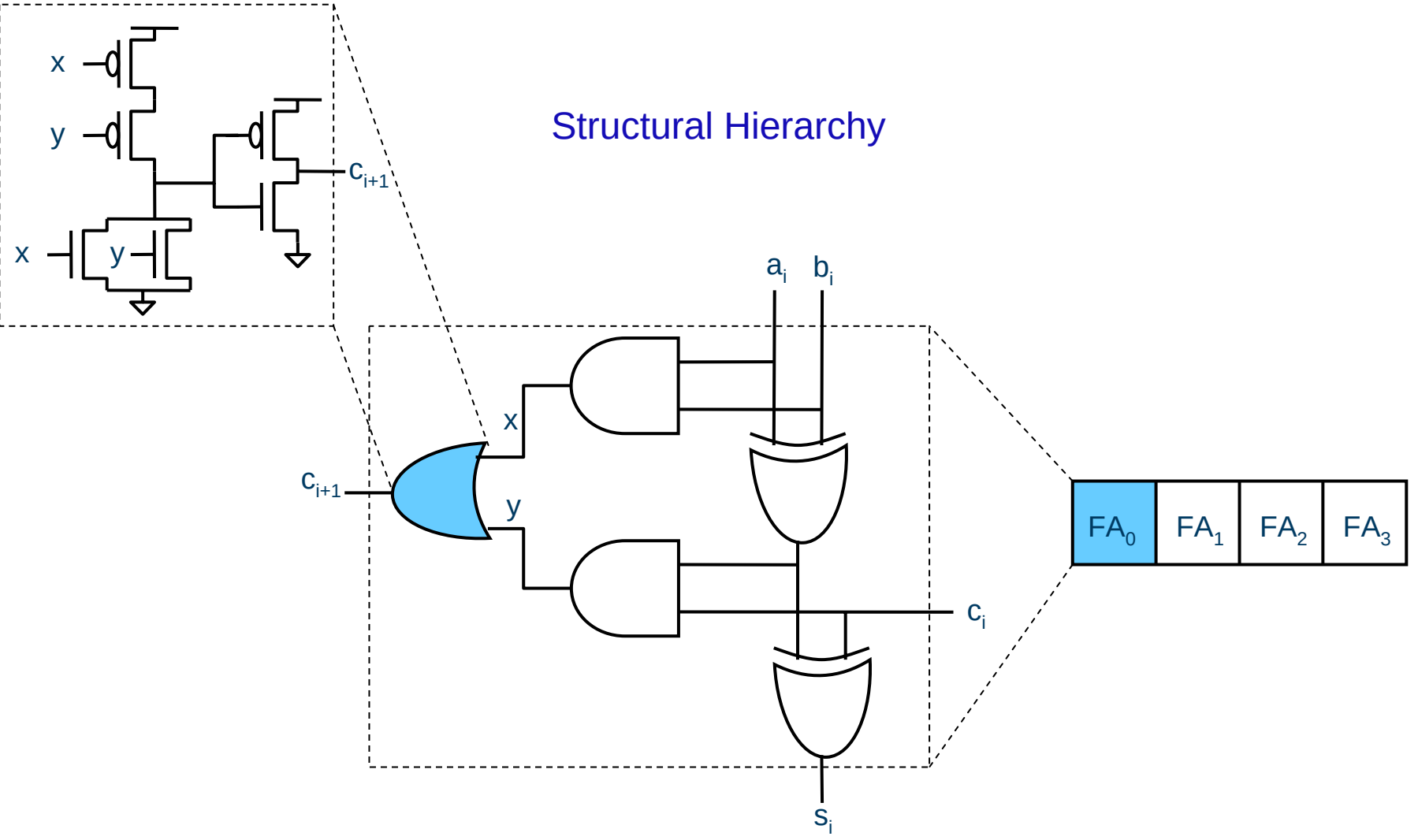
Design Strategies – *Design Parameters*

- Performance
  - Speed, power, function, flexibility
- Size of die (cost of die)
- Time to design (cost of engineering and schedule)
- Ease of test generation and testability (cost of engineering and schedule)

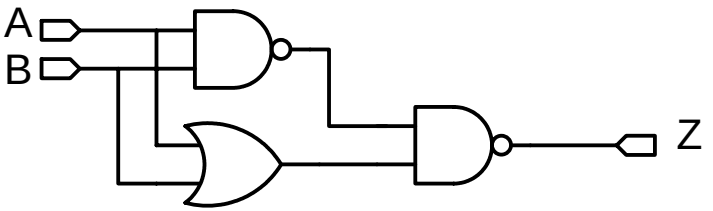
Design Strategies – *Hierarchy*

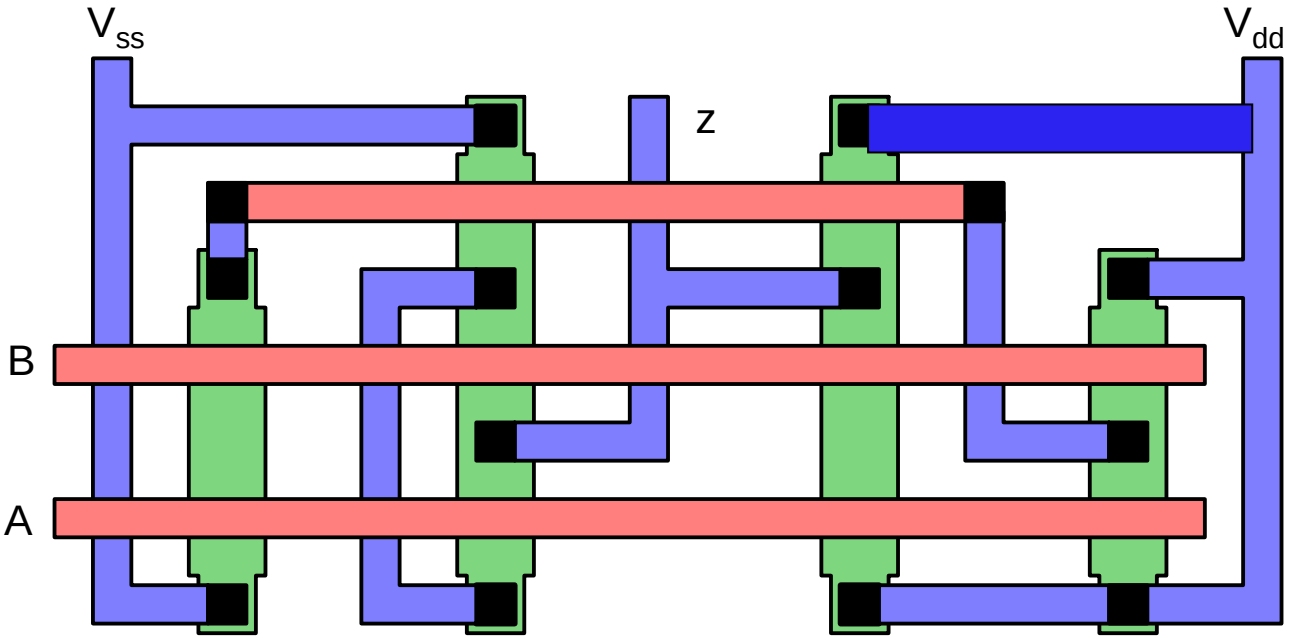
- Hierarchy
  - Dividing a module into submodules and then repeating this operation on the submodules
- Structural hierarchy
  - Reflect functionality, such as the adding, multiplexing, or storing state
- Physical hierarchy
  - An *n*-bit component is built with *n* identical bit-slices

Design Strategies – *Example*

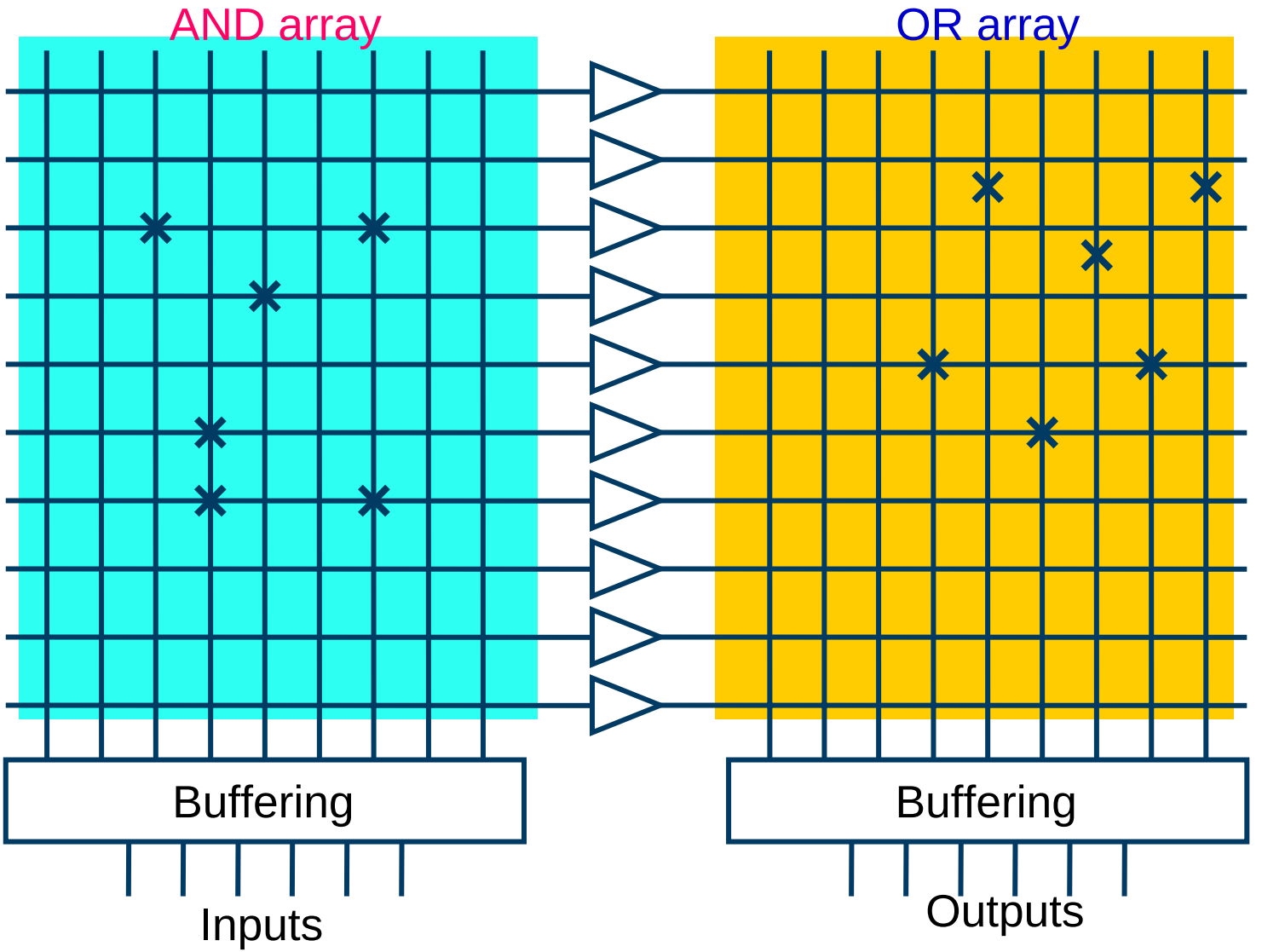


Design Styles – *Full Custom*

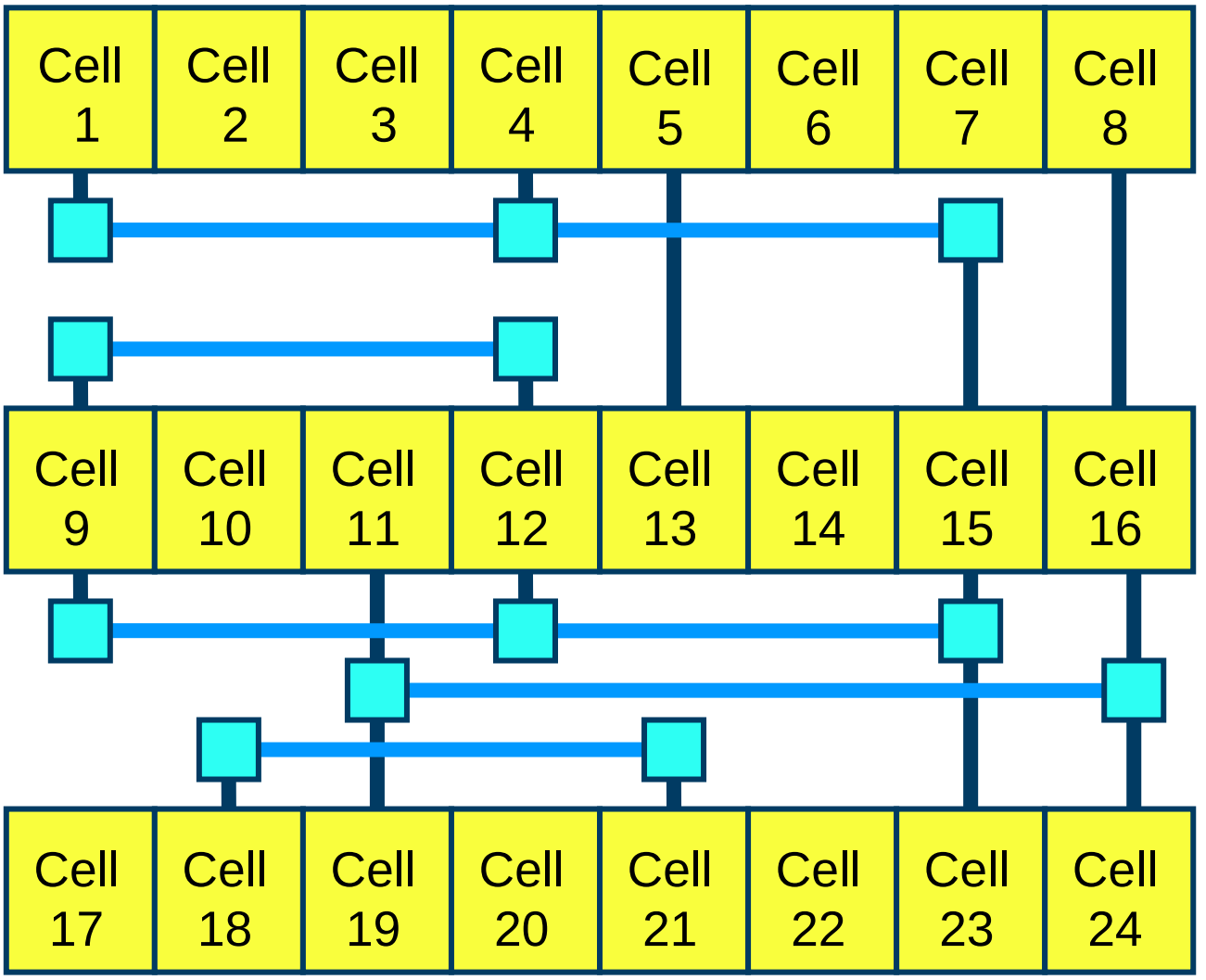




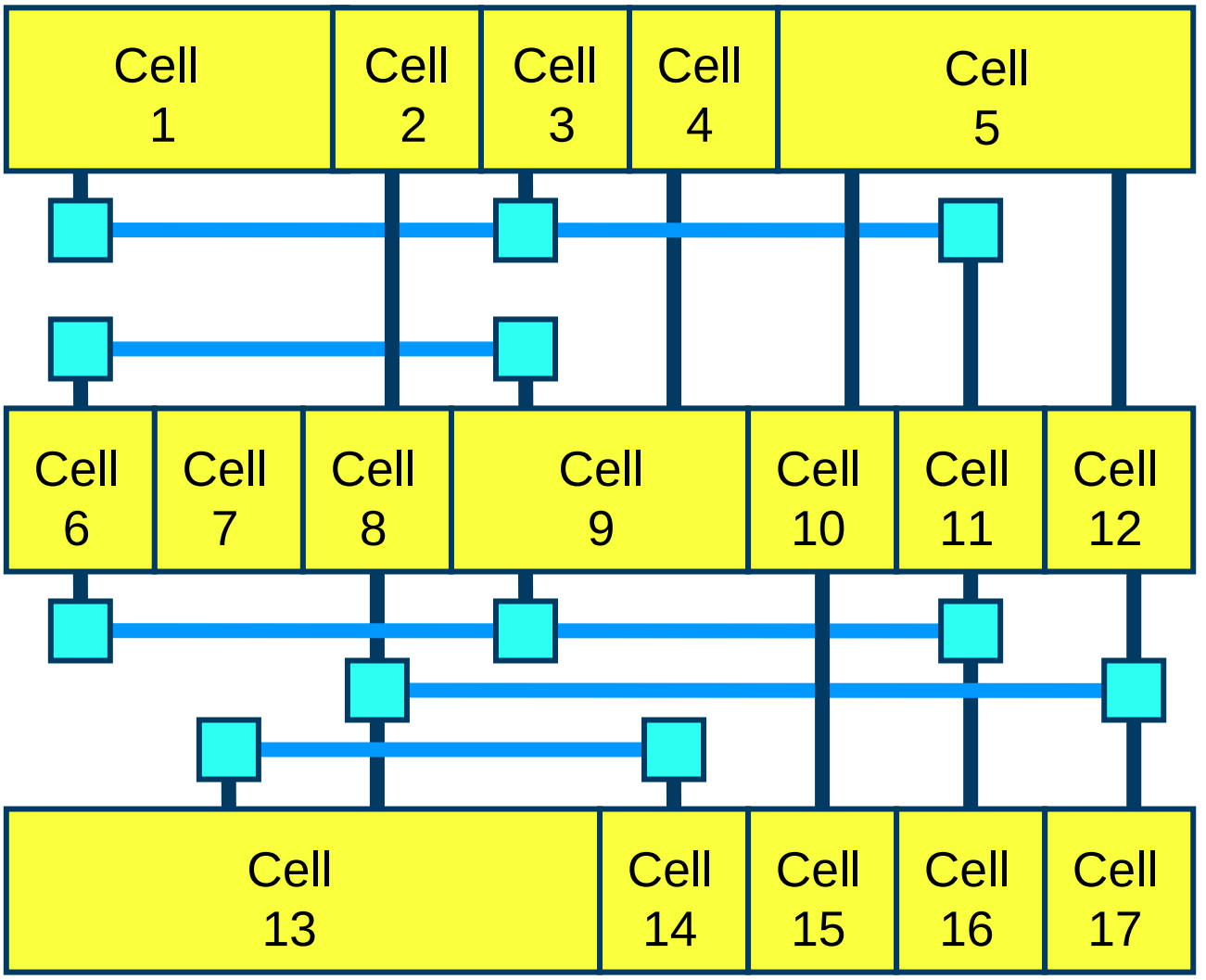
Design Styles – *Programmable Logic Array*



Design Styles – *Gate Array*



Design Styles – *Standard Cell Design*

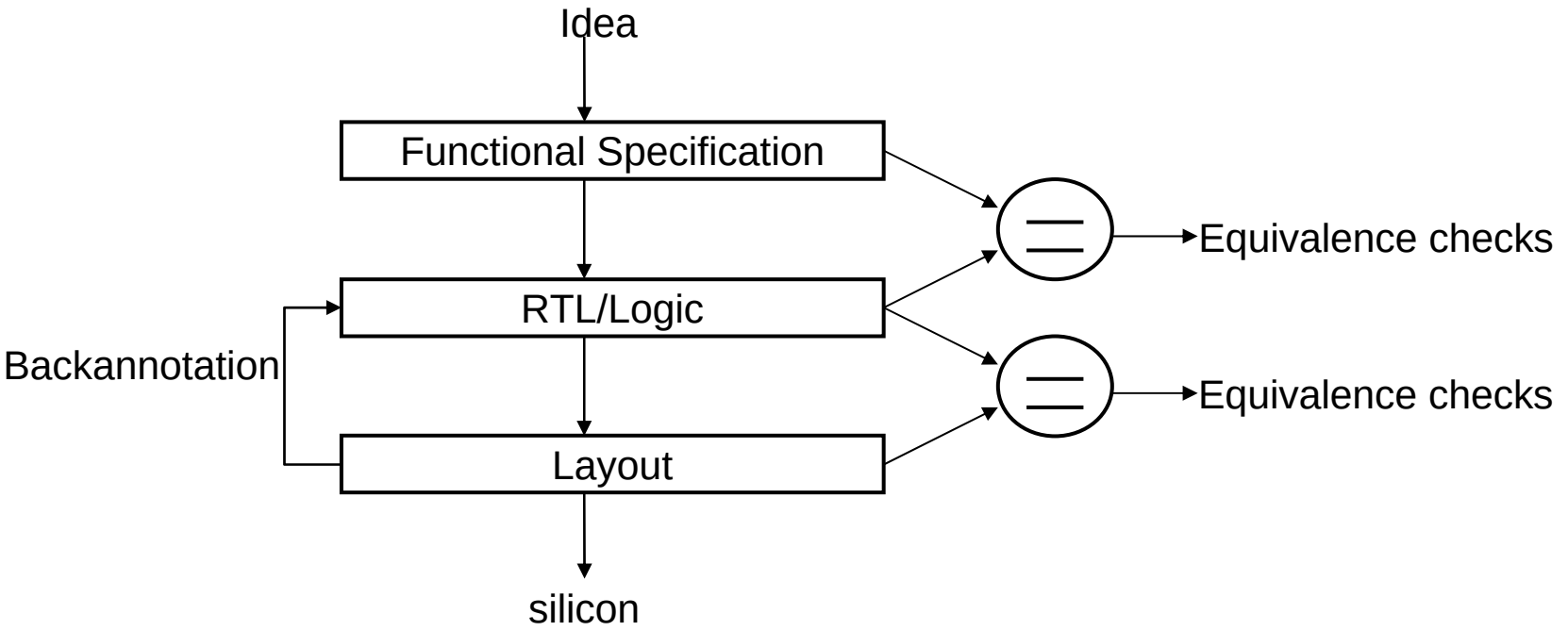


Design Styles – *Comparison*

Design Styles	Advantages	Disadvantages
Full-custom	- Compact designs; - Improved electrical characteristics;	- Very time consuming; - More error prone;
Semi-custom	-Well-tested standard cells which can be shared between users; -Good for bottom-up design;	-Can be time consuming to built-up standard cells; -Expensive in the short term but cheaper in long-term costs;
Gate array	-Fast implementation; -Easy updates; -Only two layers of metal require customization;	-Can be wasteful of space and pin connections; -Relatively expensive in large volumes;

Design Verification – *Basic Concept*

- A conventional flow through a set of design tools to produce a CMOS chip from a functional specification



Design Verification – *Simulation*

- Circuit-level simulation – SPICE
  - High accuracy
  - Long simulation times
  - Basic sources of error
    - Inaccuracies in the MOS model parameters
    - An inappropriate MOS model
    - Inaccuracies in parasitic capacitances and resistancies
- Logic-level simulation
  - Deal with simulation at the logic level
  - Timing
    - Specified with an intrinsic delay and a load dependent delay

Design Verification – *Simulation*

- An event-driven simulator
- Timing is specified with $T_{gate}=T_{intrinsic}+C_{load}XT_{load}$

- Switch-level simulation
  - Switch simulators merge logic-simulator techniques with some circuit simulation techniques by modeling transistors as switches
  - Modeling CMOS gates as either pull-up or pull-down structures

- Mixed-mode simulators
  - Simulators that merge the good points of functional simulation, logic simulation, switch simulation, timing simulation, and circuit simulation

Design Verification – *Summary*

- A good simulator is crucial to modern CMOS design
- Logic simulators are of use at the system level
- Timing simulator are useful for modules into the 100-100K transistors
- Circuit simulators are useful for 10-1000 transistors
- Mixed-mode simulators allow a trade-off in simulation accuracy and time of simulation