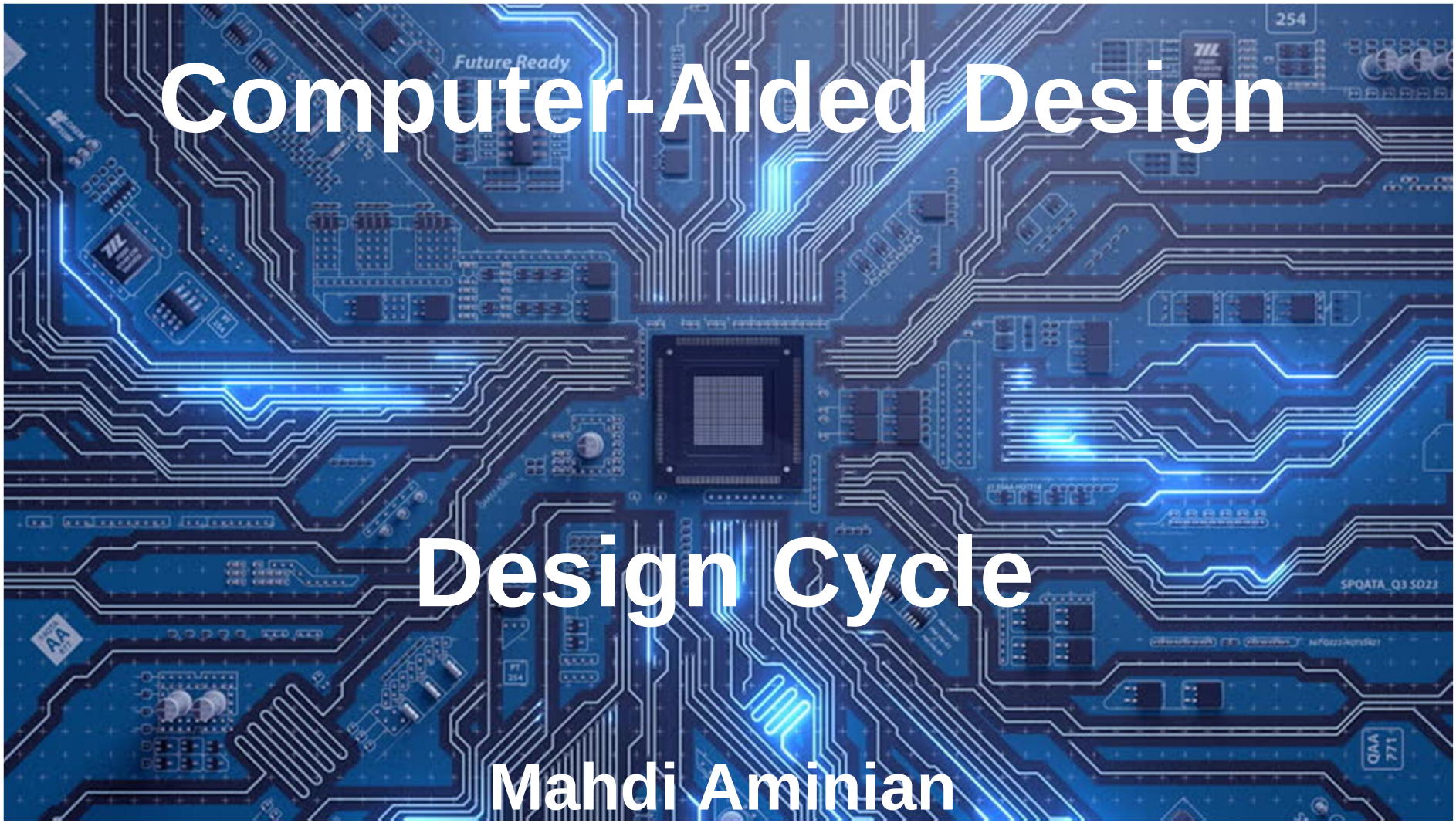


Computer-Aided Design

Design Cycle

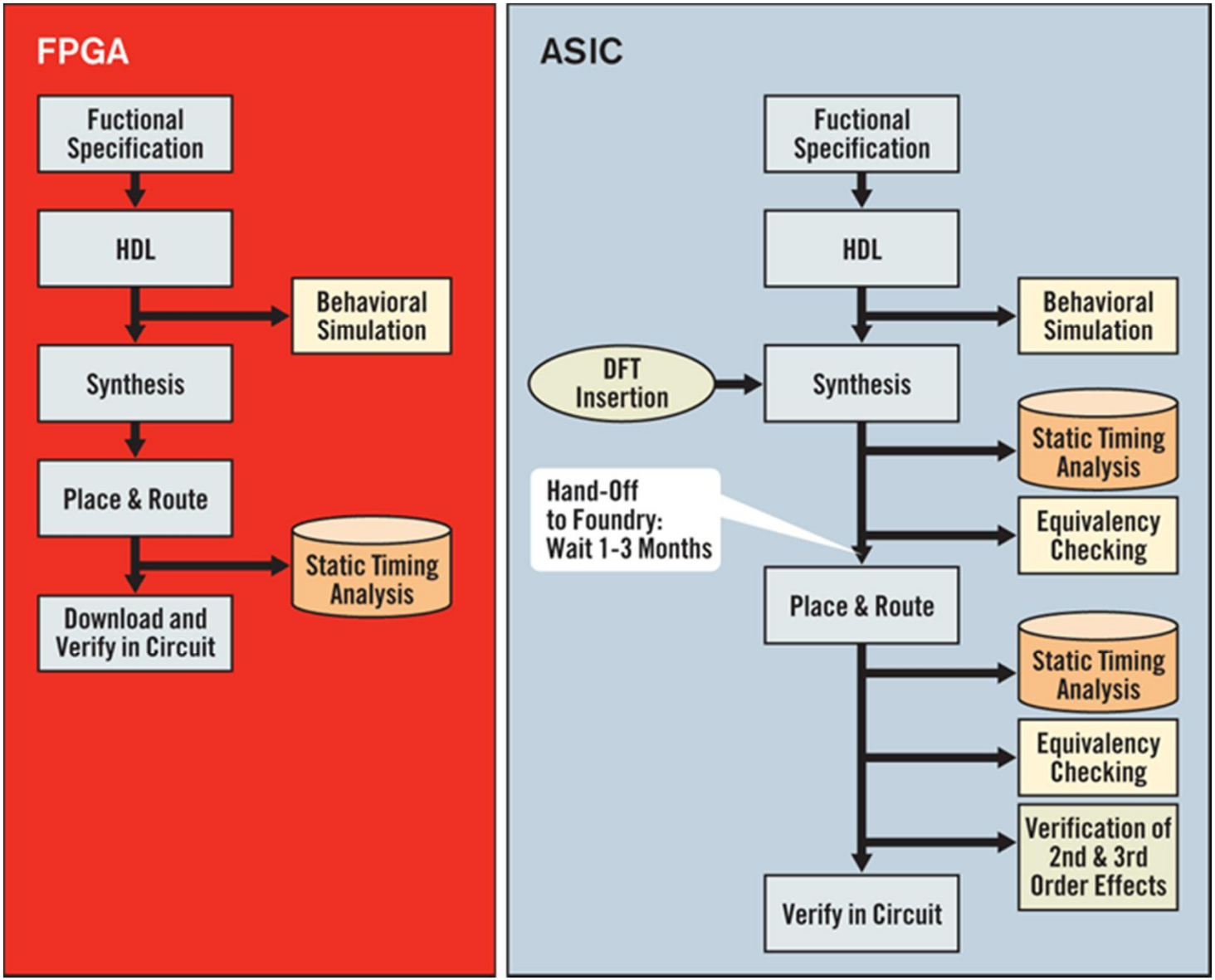
Mahdi Aminian

The University Of Guilan
Rasht - Iran

Some slides courtesy of:

- CAD slides from Dr. Saheb Zamani
- Xilinx & Altera products documentations

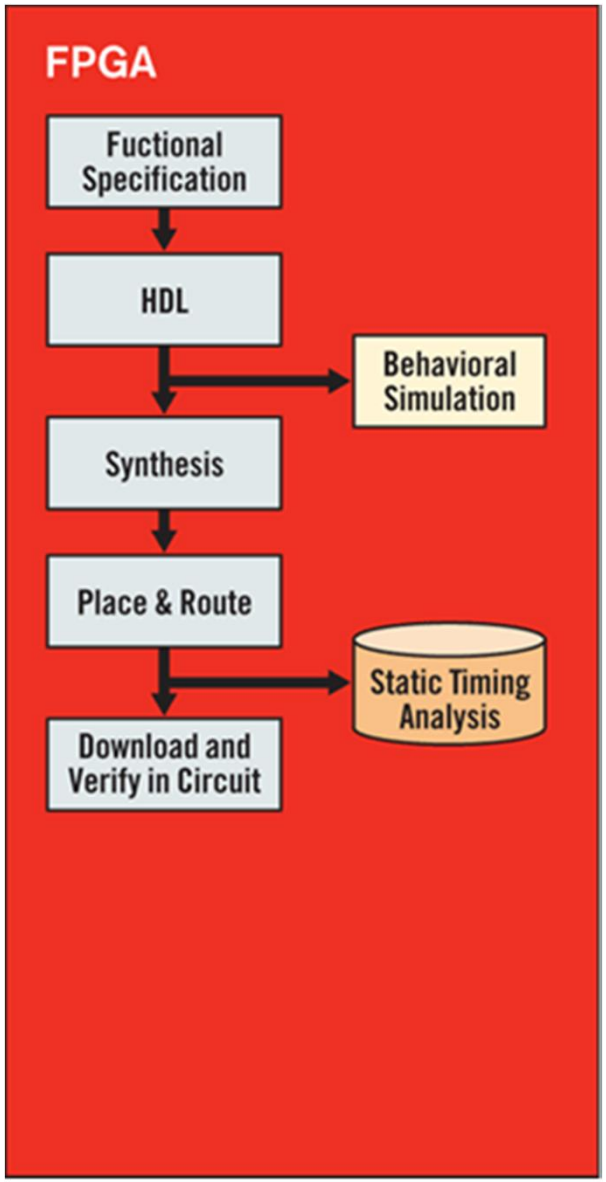
FPGA vs. ASIC Design Flow

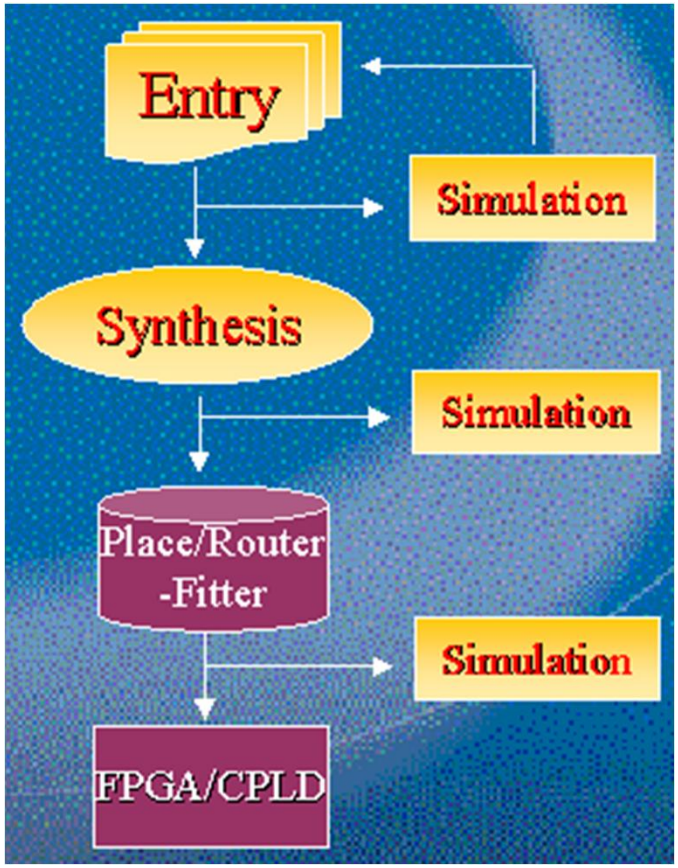


FPGA and Embedded Design

- http://www.xilinx.com/training/free-video-courses.htm#ASIC

چرخه ی طراحی برای FPLDها

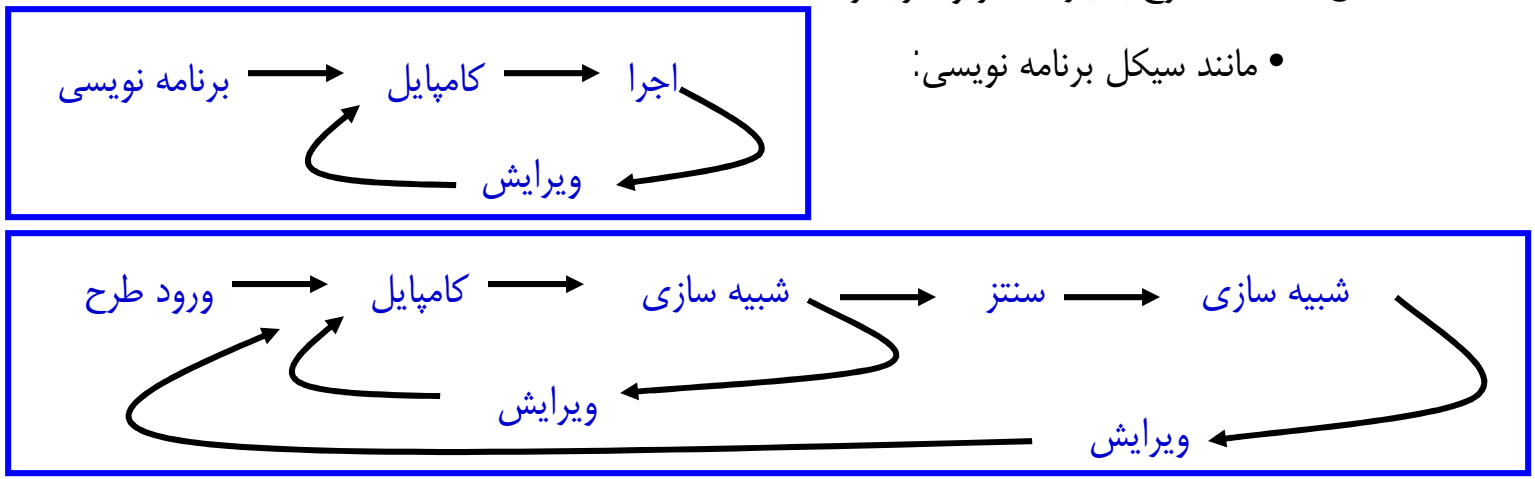




- Design Entry
  - Schematic Netlist
  - HDL
  - Waveform
  - State Diagram

چرخه ی طراحی برای FPLDها

- مزایا:
  - کوتاه شدن پروسه ی طراحی.
  - نوآوری بیشتر (پروسه ی طراحی به مراحل بالاتر رفتاری منتقل می شود) (تشابه با زبانهای سطح بالا)
  - Debug طرح بسیار آسانتر و سریعتر.
  - مانند سیکل برنامه نویسی:



  - تغییرات در طرح بسیار آسانتر.
  - بعضی شرکتها نسخه های جدید سخت افزار خود را روی CD یا از طریق اینترنت در اختیار مشتری خود قرار می دهند تا EPROM را مجددا برنامه ریزی کند.

Entry – HDL Coding

- HDL allows us to describe the functionality of a logic circuit in a language that is:
  - Easy to understand
  - Easy to share
  - Hide complicated implementation details

- Designer more concerned about the design functionality than the detailed circuit design

Simulation by Testbenches

- After HDL coding, the code has to be tested using “Testbenches” (Verification)

- Simulation Tools
  - Modelsim (Mentor Graphics)
  - Simulators of Synthesis Tools

Synthesis

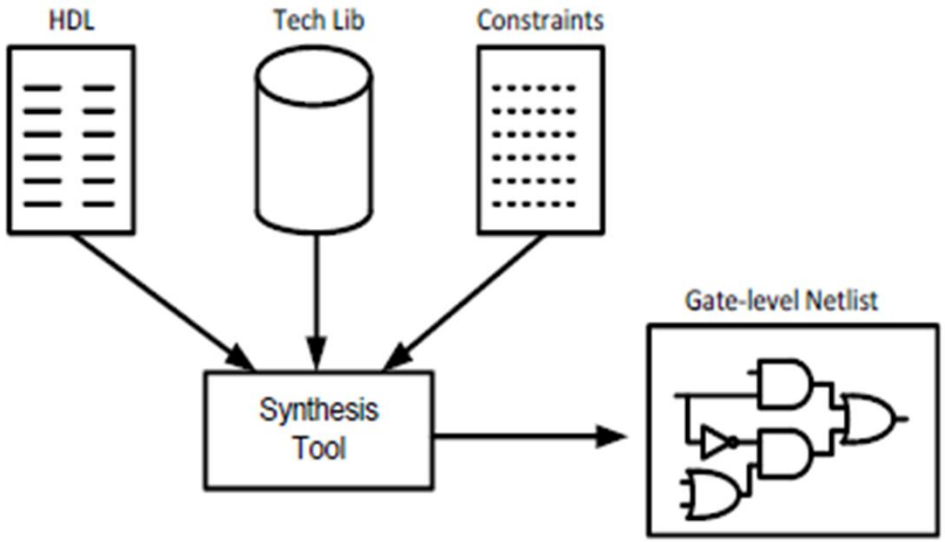
- Synthesis Tool:
  - Analyzes a piece of HDL code and converts it into optimized logic gates
  - This conversion is done according to the **“language semantics”**

⟶ We have to learn these language semantics, i.e., VHDL code

- Why using synthesis tools?
  - It is an important tool to improve designers’ productivity to meet today’s design complexity
  - If a designer can design 150 gates a day, it will take 6666 man’s day to design a 10-million gate design, or almost 20 years for 10 designers! This is assuming a linear grow of complexity when design gets bigger

Synthesis Tools

- Input
  - HDL Code
  - “Technology Library” file
  - Constraint file (Timing, area, power, loading requirement, optimization Alg.)
- Output
  - A gate-level “Netlist” of the design
  - Timing files (.sdf)

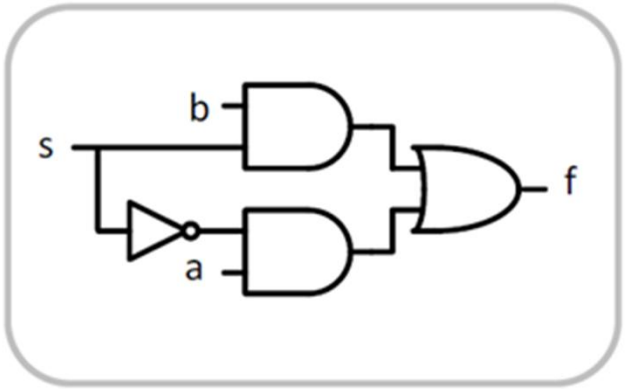


Synthesis Example

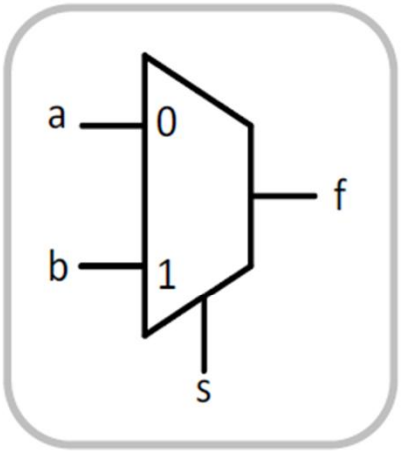
- A 2-to-1 Multiplexer (2x1 Mux)

```
If (s==0)
    f = a;
else
    f = b;
```

Synthesis Tool



Synthesized gate-level



Schematic

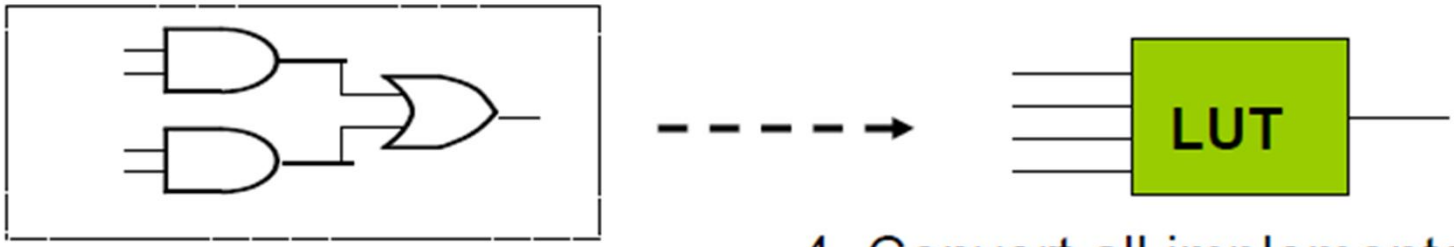
Synthesis Tool

- Infer logic and state elements
- Perform technology-independent optimizations
  - e.g., logic simplification, state assignment
- Map elements to the target technology
- Perform technology-dependent optimizations
  - Multi-level logic optimization
  - Choose gate strengths to achieve speed goals

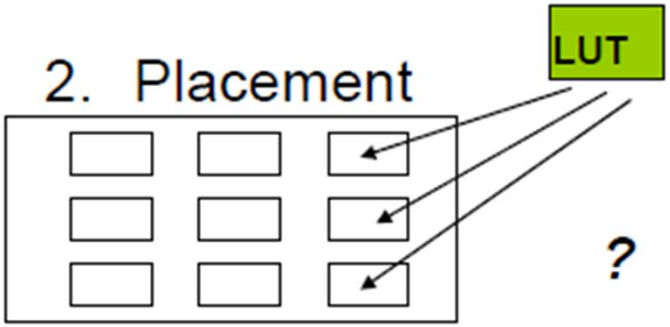
Vendor Name	Product Name	Platform
Altera	Quartus II	FPGA
Xilinx	ISE	FPGA
Mentor Graphics	Modelsim, Precision	FPGA/ASIC
Synopsys	Design Compiler, Galaxy	ASIC
Synplicity	Synplify	ASIC
Cadence	Ambit, BG, RC	ASIC

Implementation for FPGA

1. Technology Mapping

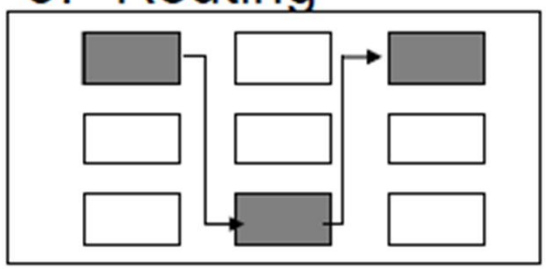


2. Placement



Assign a logical LUT to a physical location.

3. Routing

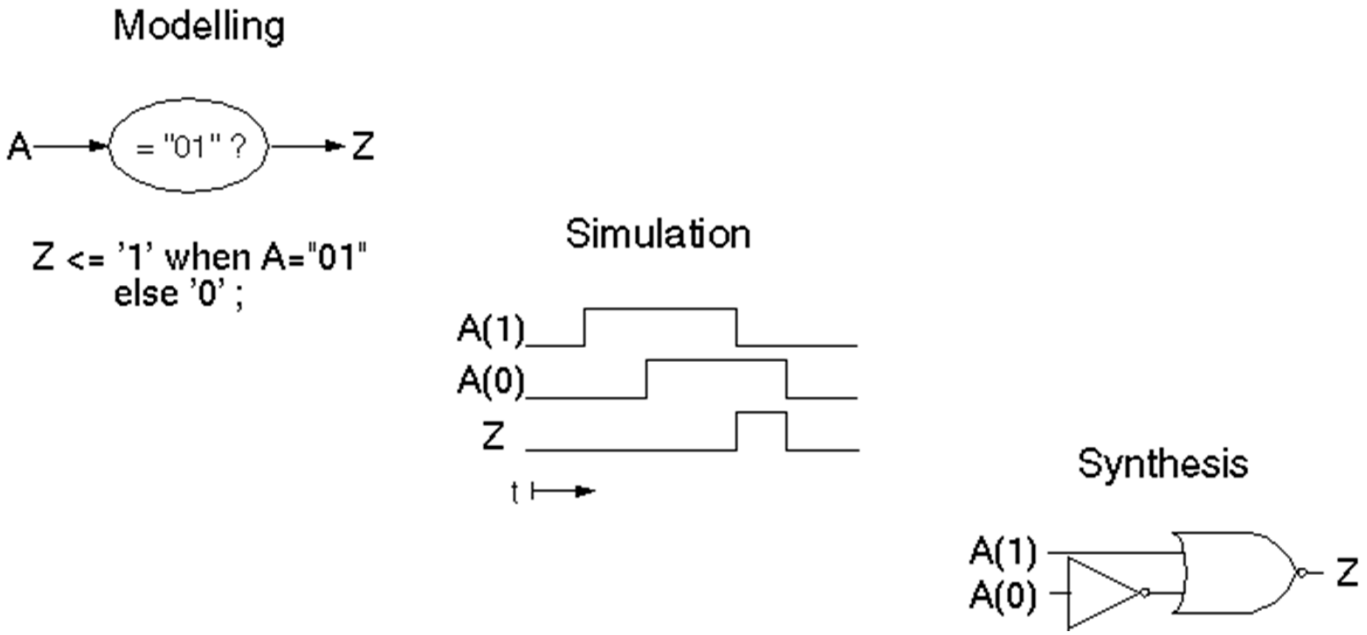


Select wire segments and switches for Interconnection.

4. Convert all implementation "details" to FPGA programming info (configuration bits): LUT RAM bits, CCM & PSM FF/SRAM bits, etc.

- Can store config bits on disk or ROM and load into FPGA as needed
- Can thus use the FPGA to implement multiple digital systems (at different times or sometimes simultaneously in different FPGA partitions)

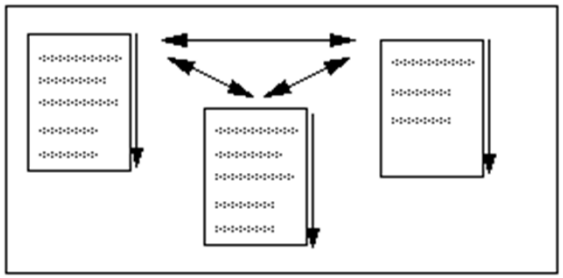
Application of HDLs



- **VHDL به عنوان زبان مستندسازی (توصیف فرمال و بدون ابهام).**
- **سنتز: تبدیل (اتوماتیک یا دستی) یک توصیف به توصیفی با جزئیات بیشتر**

Concepts of HDL

- **Execution of Statements:**
  - **Sequential**
  - **Concurrent**

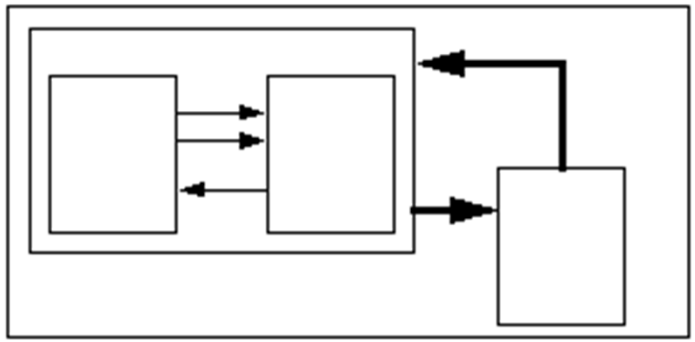


- اجرای دستورات:
  - ترتیبی
  - همزمان (موازی)

Concepts of HDL

- **Abstraction**: طرح را می توان در سطوح مختلفی از جزئیات توصیف کرد:
  - برای مدلسازی، سطوح بالا کافی است.
  - برای سنتز، ممکن است جزئیات بیشتری لازم باشد.

- **Modularity**: می توان بلوک بزرگ پیچیده را به بلوکهای کوچکتر تقسیم کرد و برای هر بخش یک مدل نوشت.
- **Hierarchy**: تشکیل یک درخت سلسله مراتبی
  - هر کدام از نودها ممکن است در سطح متفاوتی از **abstraction** توصیف شده باشد.



Modeling Capability