The Abdus Salam
International Centre for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

2065-14

Advanced Training Course on FPGA Design and VHDL for Hardware Simulation and Synthesis

26 October - 20 November, 2009

FPGA Architectures & VHDL
Introduction to FPGAs & FPGA Design Flow

Nizar Abdallah
ACTEL Corp.2061 Stierlin Court Mountain View
CA 94043-4655
U.S.A.

FPGA Architectures & VHDL

Nizar Abdallah
nizar@ieee.org

October 2009

FPGA Architectures & VHDL

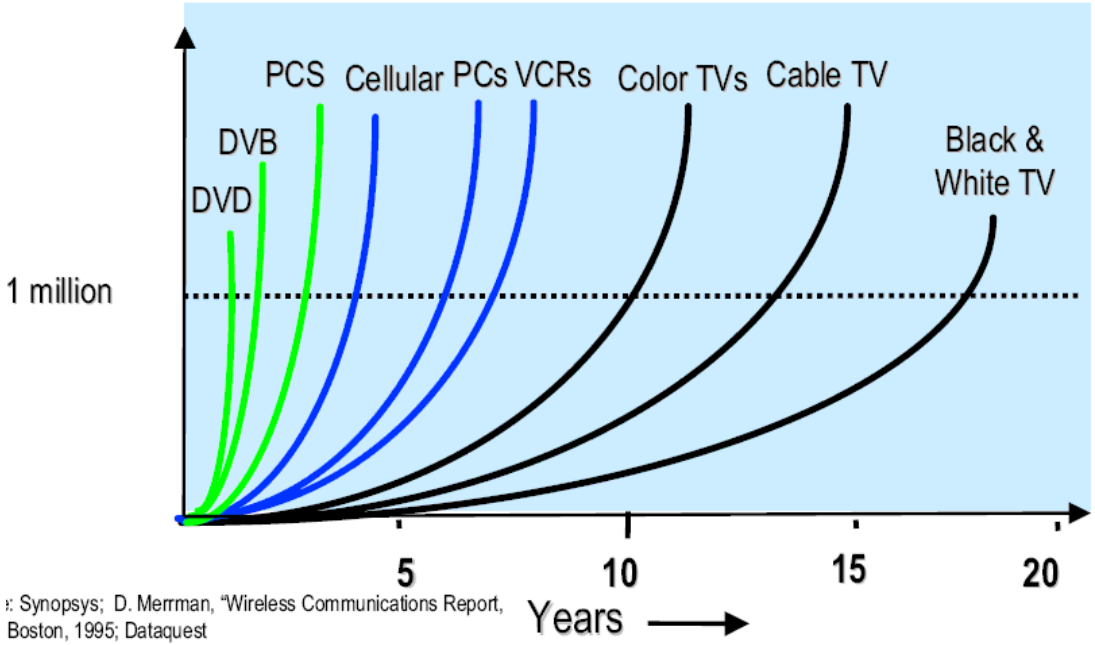
- Introduction to FPGAs & FPGA design flows
- Introduction to Synthesis
- The VHDL hardware description language
- Design verification, validation, and testing

- Programmable logic & FPGA architectures
- Actel's SoC Flash FPGA architectures
- Co-design & co-verification of HW/SW embedded systems

- Emerging technologies and future opportunities.

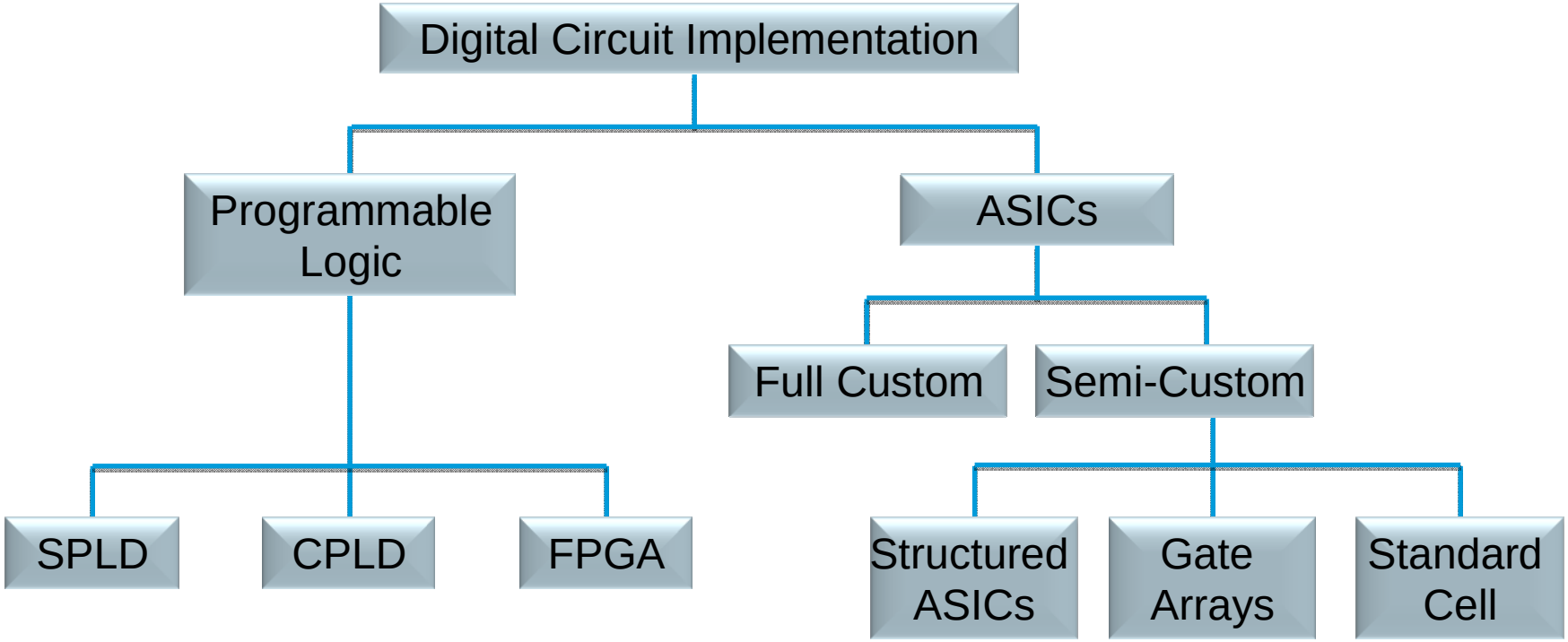
Motivation

- High integration
  - Basic: memory, logic, I/Os
  - Even more: PLL, DSP, A/D, D/A, clock oscillator…
- Accelerated product's time-to-market
  - Flexibility needs
- Design skills
  - System level
  - DSP algorithms
  - SW/HW co-design
  - HDL modeling
  - Design methodology
  - Project management



: Synopsys; D. Merrman, "Wireless Communications Report, Boston, 1995; Dataquest

Digital Logic Technologies



What's a FPGA?

Field PROGRAMMABLE Gate Array

What's a FPGA?

A simplistic old definition:

- a high capacity programmable logic device
- An array of programmable basic logic cells surrounded by programmable interconnects
- Can be configured (programmed) by end-users (field-programmable) to implement specific applications
- Capacity up to multi-millions logic gates and speed up to 500MHz
- Popular applications: prototyping, on-site hardware reconfiguration, DSP, logic emulation, network components, etc…

CPLDs versus FPGAs

CPLDs

More Predictable Timing

Simple Architecture

Fewer Registers

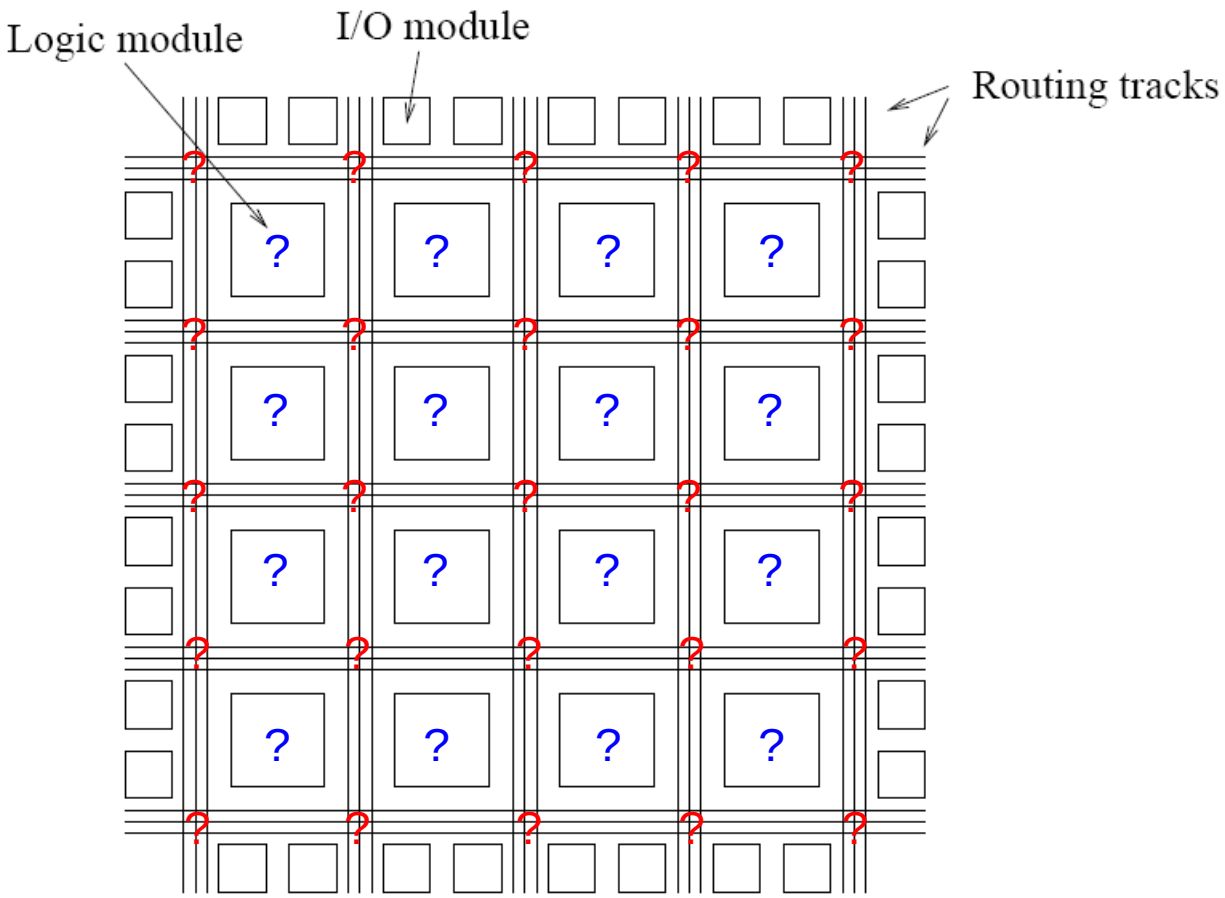
FPGAs

Larger Capacity

Complex Architecture

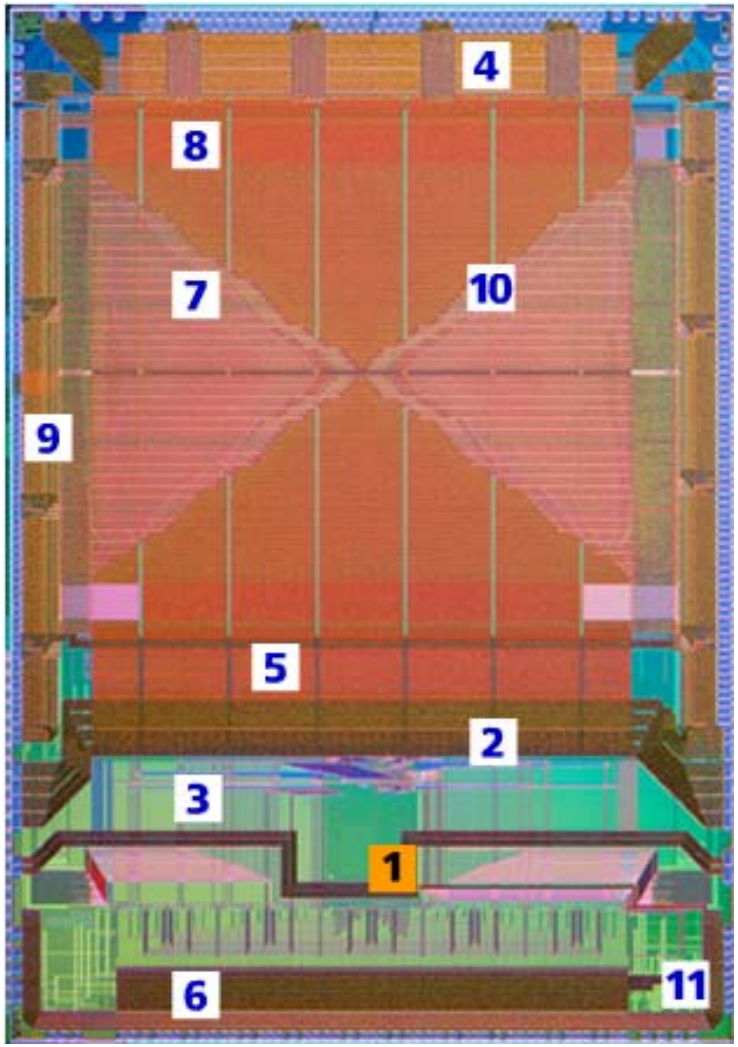
More Registers

Basic FPGA Block Diagram



Generic FPGA Architecture

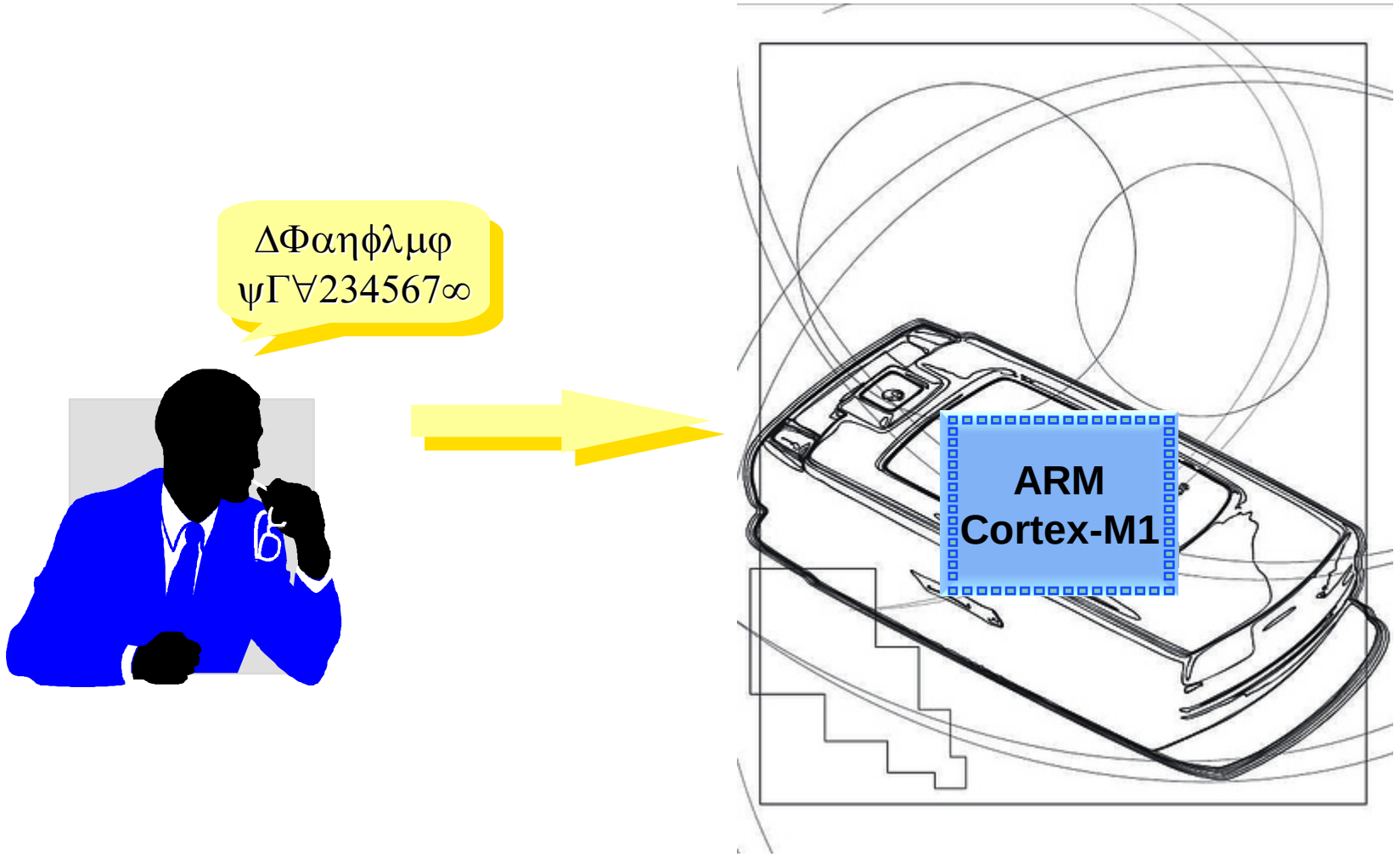
Rich FPGA Block Diagram



Rich FPGA Architecture

1. Integrated ADC
2. Low-Power Support
3. Embedded Flash Memory
4. Advanced I/O Standard
5. Charge Pumps
6. Analog Quads
7. Flash FPGA VersaTile
8. SRAM and FIFOs
9. Integrated Oscillators
10. Routing Structure
11. JTAG

DESIGNER'S DREAM

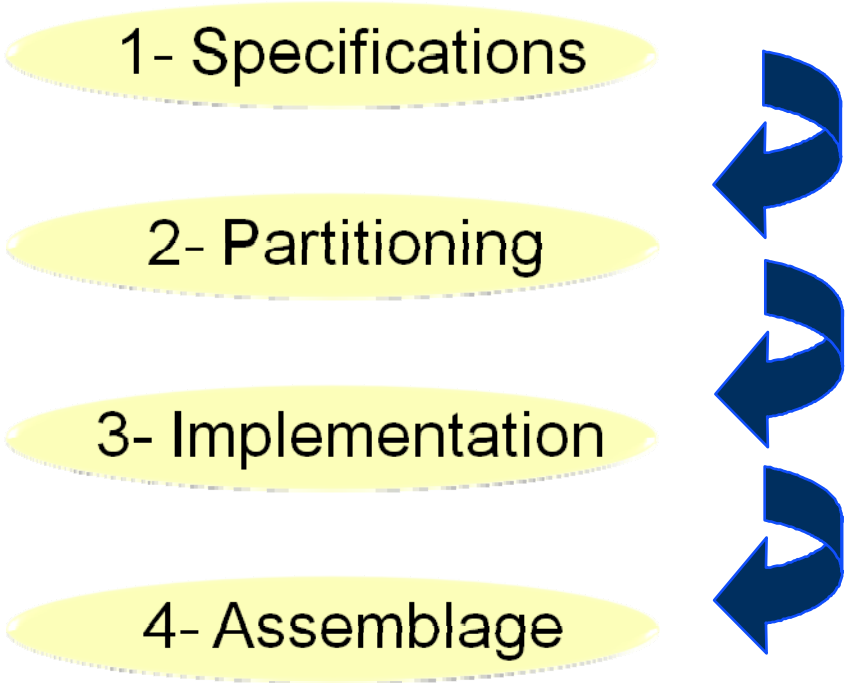


Design Principles

- Hierarchy
  - Divide & conquer
  - Simplification of the problem
- Regularity
  - Divide into identical building blocks
  - Simplifies the assemblage verification
- Modularity
  - Robust definition of all components (entity)
  - Allows easy interfacing
- Locality
  - Ensuring that interaction among modules remains local
  - Makes designs more predictable and re-useable

Design Methodology

- Top-Down design methodology in 4 steps



Step 1: Specifications

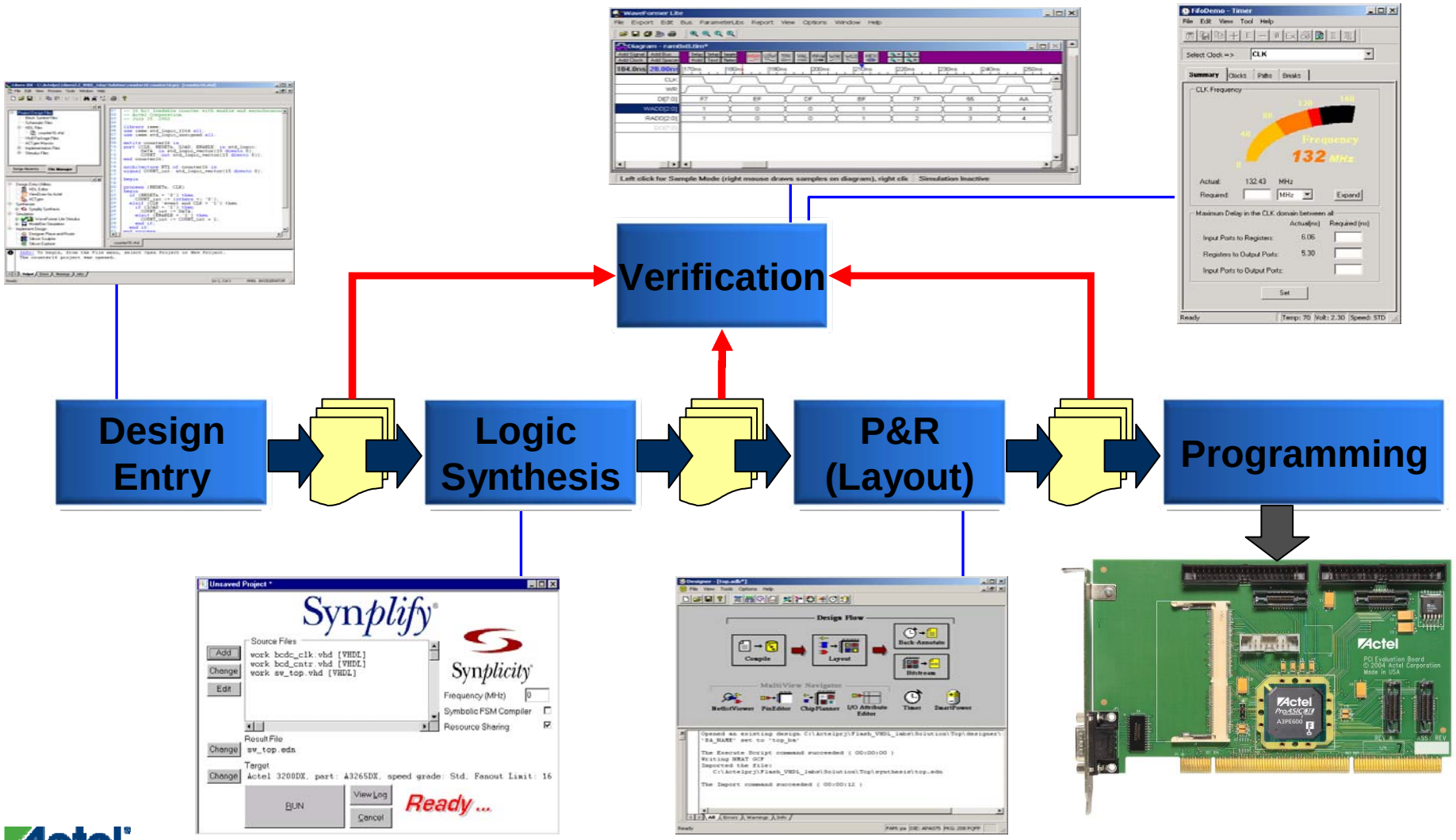
- Put down the circuit concept
  - Easy verification
  - A reference manual for communication
    - Between people
    - Between people and computers
  - How?
    - No Ordinary language
    - Accurate language
    - A language that can be simulated
- Put down the requirements
  - Timing budget
  - Power budget
  - Area budget
  - Financial budget

Step 2: Partitioning

- Divide and conquer strategy
  - Very difficult step: Relays on the know-how of the designer
  - Main idea: To split into several small parts

Step 3: Implementation

- Simplified FPGA design implementation flow

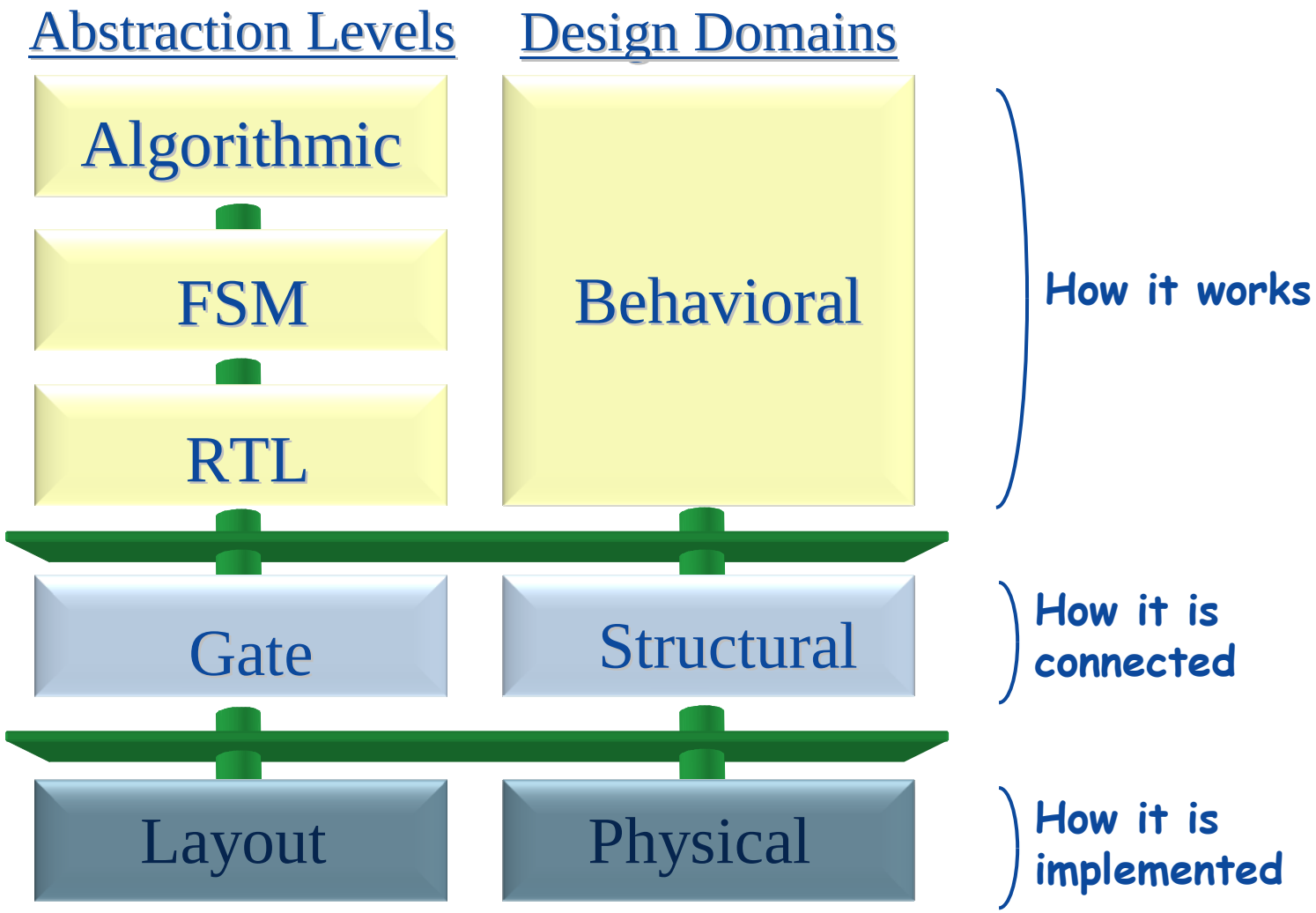


Step 4: Assemblage

- Hierarchical way
- Start from the lowest level
- Final product validation is now possible
  - Compare to original specifications
  - Simulate
  - On-board verification

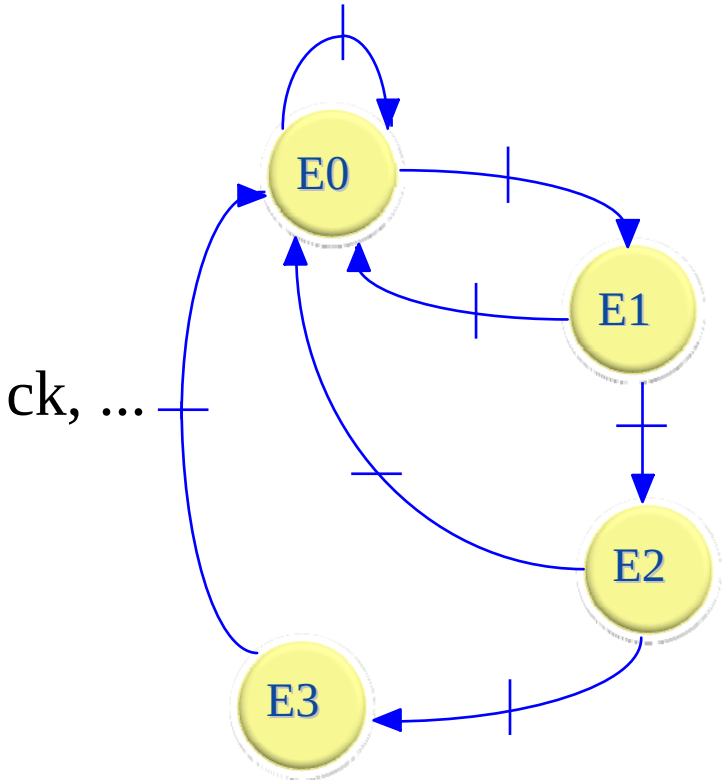
Design Abstraction & Design Domains

- Allow dealing with design complexity



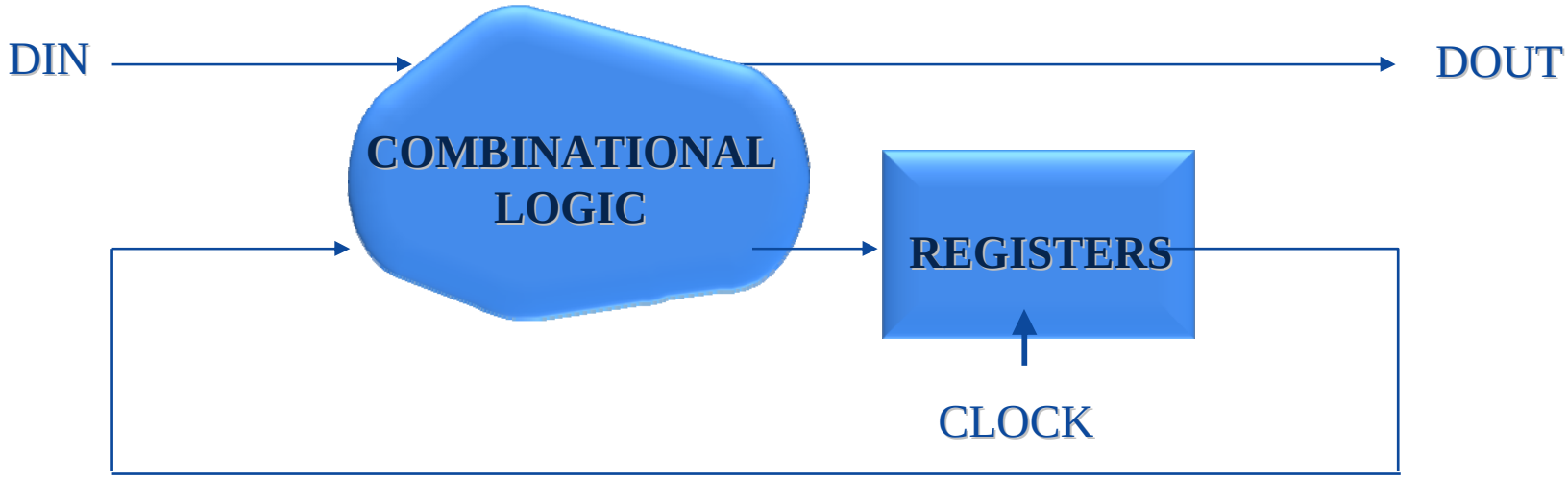
FSM Abstraction Level

- Finite State Machine
- Controller part of a digital design
- Internal states
- State changes driven by:
  - Status information
  - Clock and other external inputs...



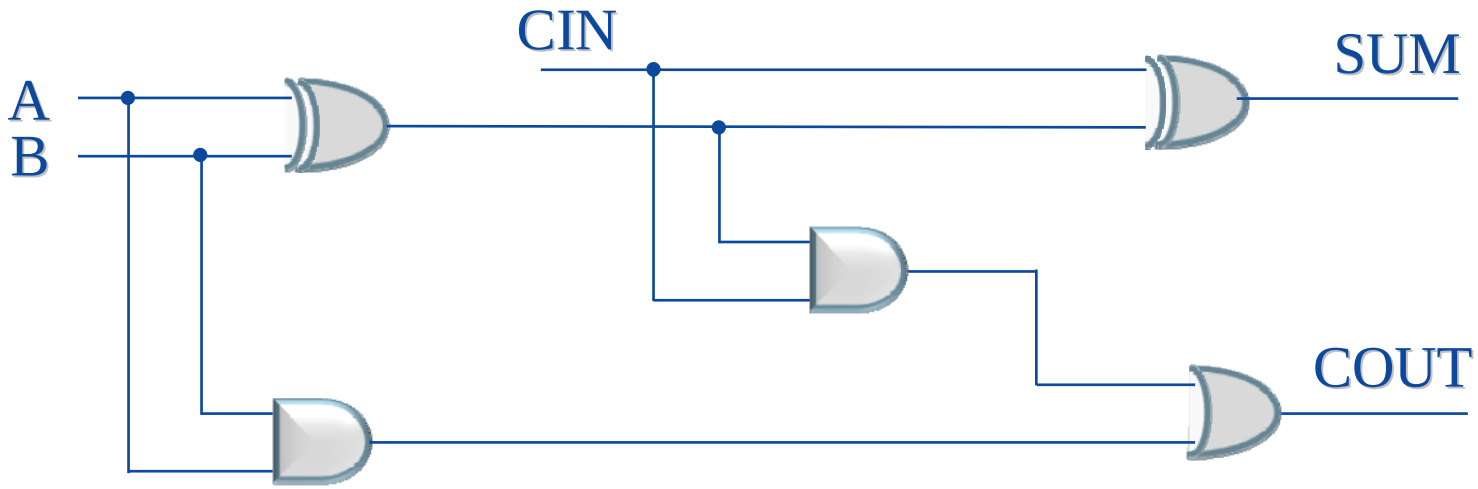
RTL Abstraction Level

- Register Transfer Level
- Registers connected by combinatorial logic
- Very close to the hardware



Gate Abstraction Level

- A gate net-list describing instantiation of models



Questions ?

FPGA Architectures & VHDL

- Introduction to FPGAs & FPGA design flows
- Introduction to Synthesis
- The VHDL hardware description language
- Design verification, validation, and testing

- Programmable logic & FPGA architectures
- Actel's SoC Flash FPGA architectures
- Co-design & co-verification of HW/SW embedded systems

- Emerging technologies and future opportunities.

What's Synthesis?

- The process of converting a design from one abstraction level into a lower abstraction level
- Logic synthesis is mapping an RTL description into a specific target technology
- Includes an optimization step for:
  - Faster speed
  - Smaller area
- Synthesis flow involves multiple steps
  - State minimization
  - State assignment
  - Logic optimization
  - Technology mapping
  - Timing optimization

Logic Synthesis

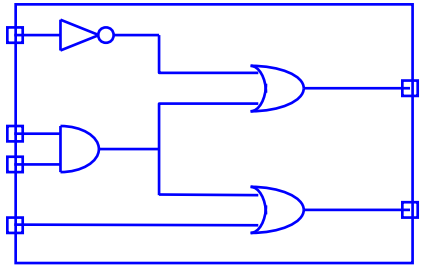
```
ENTITY dec2to4 is
PORT(A,B,enable:in BIT;
        vdd,vss,vdde,vsse:in BIT;
        Y:out bit_vector(0 to 3));
end dec2to4;

architecture dflow of dec2to4 is

signal a_bar,b_bar:bit;
signal a1,a2,a3,a4:bit;

begin
   a_bar <= not a;
   b_bar <= not b;
```

Combinational Logic Synthesis



Gate Netlist = Gate Level Structural Description

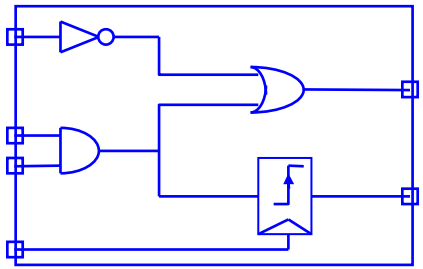
```
ENTITY adder is
PORT(A,B,enable:in BIT;
        vdd,vss,vdde,vsse:in BIT;
        ck: in bit;
        Y:out bit_vector(0 to 3));
end dec2to4;

architecture dflow of adder is

signal regstr:reg_vector(0 to 3)
                        register;
signal a1,a2,a3,a4:bit;

begin
```

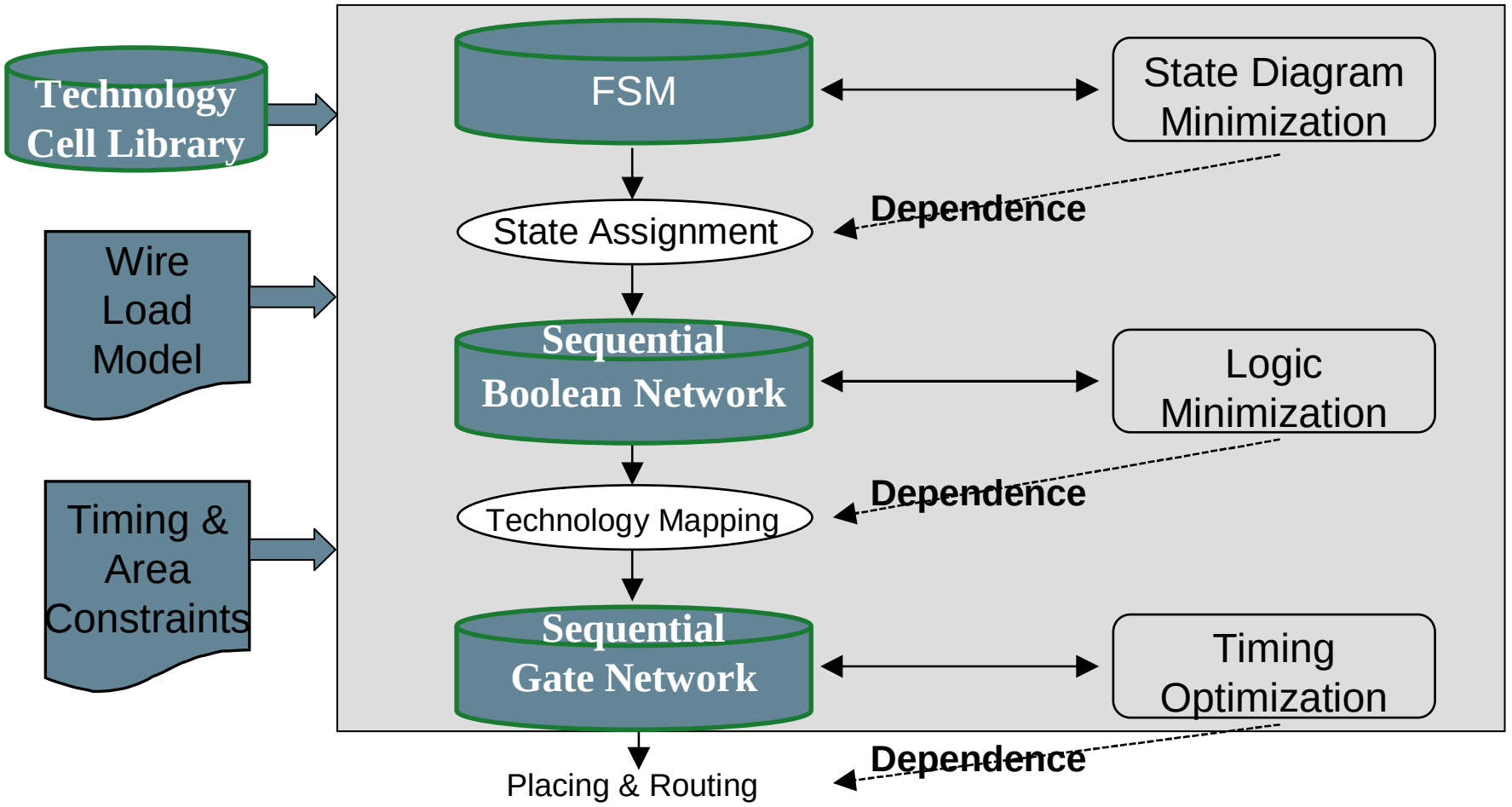
Sequential Logic Synthesis



Gate Netlist = Gate Level Structural Description

Optimization and Logic Synthesis

- Involves multiple internal iterative steps



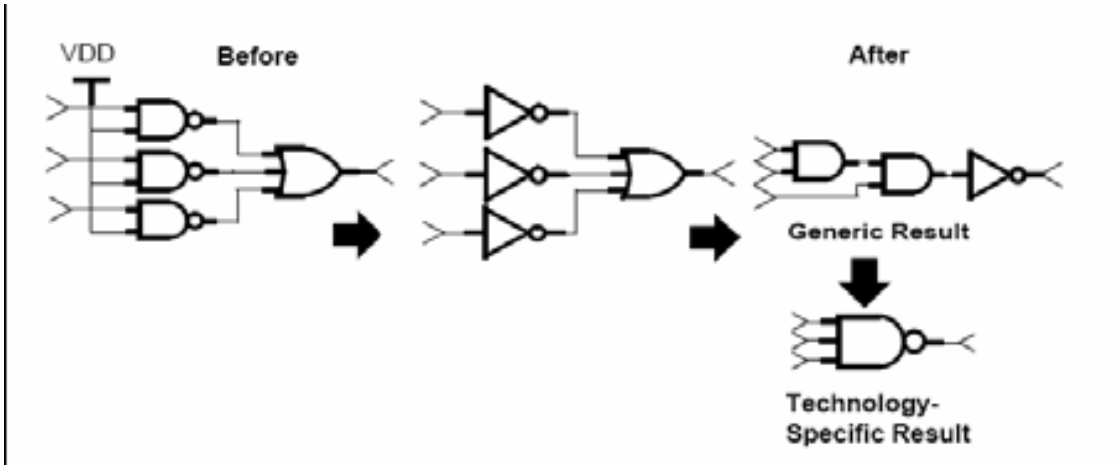
Synthesis Step-by-Step

(Precision Synthesis Reference Manual, Chapter 4)

1. Analyze the Design
   - Check HDL syntax (is it synthesizable?)
   - Locate referenced cells and libraries
   - Resolve parameters and defines
   - Detect design top-level and hierarchy dependencies to determine mapping order
2. Mapping
   - Build hierarchy
   - Infer sequential elements: Flip-flops and latches
   - Infer operators: +, -, *, / (to blackbox models)
   - Infer RAMs
   - Infer Boolean logic
   - Infer finite state machines

Synthesis Step-by-Step (cont'd)

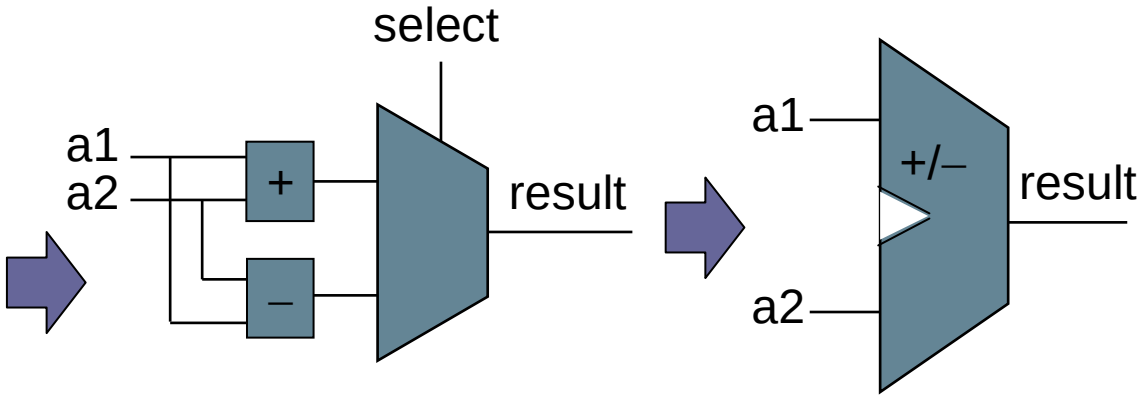
3. Pre-Optimization
   - Component extraction – counters, RAMs, etc., are separated from generic logic
   - Unused logic pruning
   - Boundary optimization
     - Disconnect unused module ports
     - Merge multiple ports connected
   - Constant propagation



Synthesis Step-by-Step (cont'd)

- Resource sharing

```
if (select)
   result = a1 + a2;
else
   result = a1 – a2;
```



Before resource sharing

After resource sharing

Synthesis Step-by-Step (cont'd)

4. Synthesis
   - Maps pre-optimized design into gates and/or FPGA look-up tables
   - Implements operators
   - Generates a complete, but non-optimal, netlist
5. Optimization
   - Reorganizes logic to meet timing or area constraints
   - Calculates estimated interconnect delays using wire load model
   - Resolves design rules such as
     - Maximum fanout
     - Maximum net capacitance
     - Maximum transition time on net
6. Synthesis result is a netlist (circuit) that satisfies
   - Design rules
   - Area constraints
   - Timing constraints based on estimated delays

Synthesis Vendors

- Synopsys:
  - Design Compiler
  - FPGA Compiler II
- Mentor Graphics:
  - Exemplar Logic Leonardo Spectrum
  - Precision
- Synplicity:
  - Synplify
  - Synplify Pro

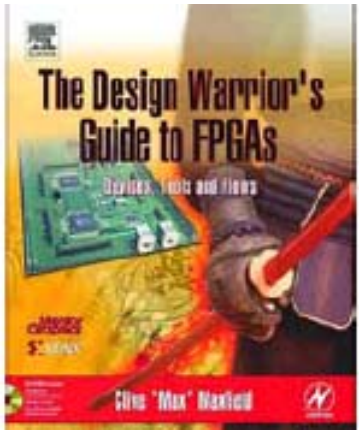
RTL Simulation

- Simulates with a clock-cycle accuracy
  - No timing guarantee
- Allows getting proper function of the design before jumping into details
- We chose VHDL for this course
  - One of the two popular languages used for hardware modeling

VHDL-Vital ’95 Simulation Vendors

- Synopsys:
  - Scirocco
- Mentor Graphics:
  - Model Technology ModelSim
- Cadence:
  - NC-VHDL simulator

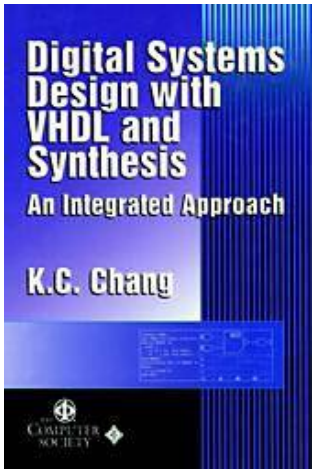
References

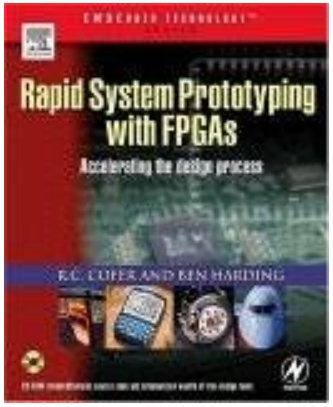


- Clive Maxfield,
  The Design Warrior's Guide to FPGAs: Devices, Tools, and Flows,
  Elsevier Science & Technology, 2004
  ISBN 0750676043

- Smith, D. J.,
  HDL Chip Design,
  Doone Publications, Madison AL, 2001
  ISBN 0965193438

References (cont'd)

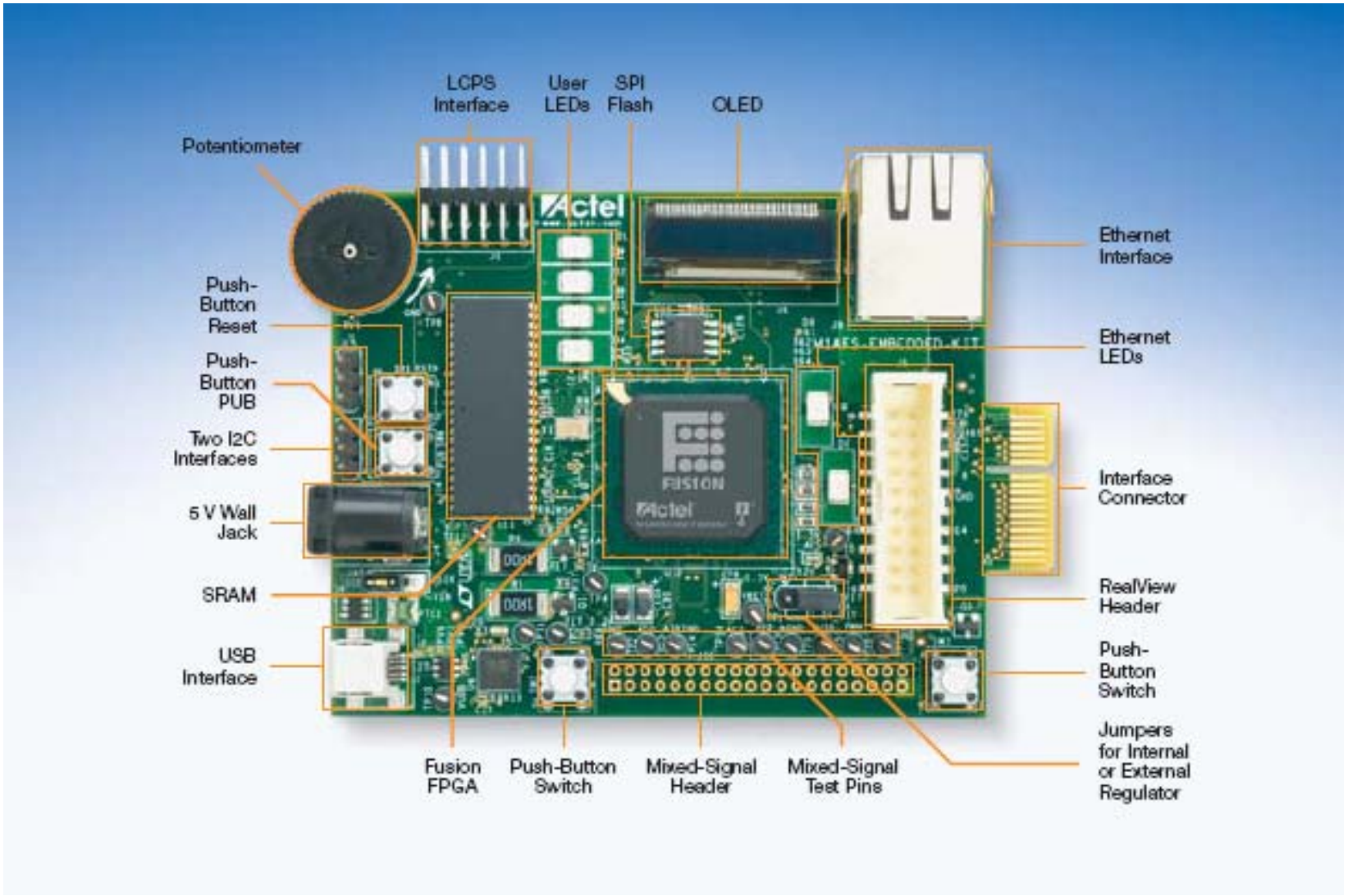




- K.C. Chang,
  Digital Systems Design With VHDL and Synthesis: An Integrated Approach,
  Wiley-IEEE Computer Society Press, First edition 1999
  ISBN 0-7695-0023-4

- RC Cofer, Benjamin F. Harding,
  Rapid System Prototyping with FPGAs: Accelerating the Design Process,
  Newnes; Bk&CD-Rom edition, Sep. 2005
  ISBN 0750678666

Lab Resources: Fusion Embedded Development Kit



Questions ?