ECE 4514
Digital Design II
Spring 2007

Lecture 1: Introduction

Patrick Schaumont

Course Staff

- Instructor:
  - Patrick Schaumont (schaum@vt.edu)
  - Durham Hall 361
- TA:
  - Sandesh Prabhakar (sandeshp@vt.edu)
- Office Hours
  - Tuesday 9:00AM - 10:00 AM
  - Friday 11:00 AM - 12:00 PM
  - by e-mail appointment

Digital Design II is not a sequel !!

Digital Design I	Digital Design II
Write Boolean Algebra	Hardware Description *Language*
Write Truth Tables	Use Logic Simulators
Draw Karnaugh Maps	Use Logic Synthesis *Tools*
Synchronous Circuits (**20** gates) Counters, Shifters, ALU (maybe)	Synchronous Circuits (**20,000** gates) RISC Processors, Channel-Coding Unit, Encryption Unit

Digital Design II combines three aspects

- Design of Digital Circuits using the Verilog Language
  - Digital circuits are not captured in a schematic, they are expressed in a *hardware description language*
- Use tools to simulate and implement circuits in the Verilog Language
  - Make use of commercial CAD software
  - Make use of a hands-on kit
- Implement designs and learn from the pros.
  - Investigate the design trajectory of complex digital designs starting from algorithm to implementation.
  - Hands-on projects to map complex digital circuits

Digital Design II has 3 types of lectures

(L)anguage: focusing on Verilog

(T)ools: focusing on methodology and tools

(D)esign: focusing on a design example

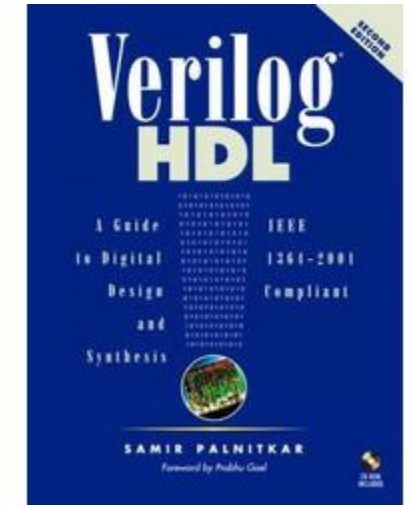
Wk	Date	Lec	Topic	HW
1	15-Jan-08	L1	Introduction and Overview	HW1
	17-Jan-08	L2	(L) Hierarchical Modeling	
2	22-Jan-08	L3	(L) Modeling Elements in Verilog	HW2
	24-Jan-08	L4	(T) HDL Simulators	
3	29-Jan-08	L5	(L) Gate-level Modeling	HW3
	31-Jan-08	L6	(D) Design and Simulation of an RNG	
4	05-Feb-08	L7	(L) Dataflow Modeling	HW4
	07-Feb-08	L8	(D) Synthesis of an RNG	
5	12-Feb-08	L9	(L) Testbench Design	HW5
	14-Feb-08	L10	(L) Behavioral Modeling 1	

Course Website

- The central resource for this course:
  - http://learn.vt.edu
  - Announcements, slides, syllabus, homework and project assignments, bulletin board.

Course Text

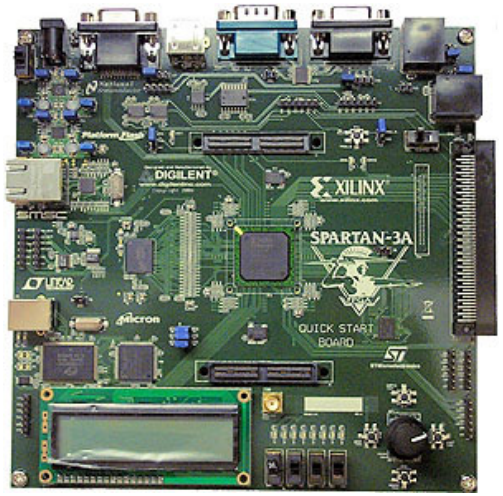
- Textbook
  Verilog HDL: a guide to Digital Design and Synthesis,
  Second Edition,
  Palnitkar



- Additional reading and text material will be posted on Blackboard

Course Equipment

- Spartan 3E Starter Kit
  - $109, Digilent Inc
  - Purchase instructions are on Blackboard



- Software
  - DVD with ISE 9.2, Chipscope and other tools will be distributed in class during week 2. CEL will not distributed DVDs.
  - Software installation keys will be distributed in class during week 2.

Course Objectives

- Learn to use Verilog to design complex digital circuits, using simulators and synthesis tools
  - Structural, Dataflow, Behavioral Design
  - Test-bench Design and Digital Simulation
  - Design of Combinational & Sequential artifacts
  - Digital Synthesis using tools
  - Optimizations: Time-Area Trade-offs
  - FPGA Board Hands-on
- Prerequisite:
  - ECE 3504 (C- or better)

Course Organization

I

- Verilog Modeling: Structural/ Dataflow/ Behavioral
- Verilog Simulation
- Design: Random Number Generation, SHA-1
- 5 Homeworks + 1 Project

← Drop Deadline

- Review Lecture + Midterm

- Spring Break

II

- Verilog for Digital Synthesis
- Optimization: A/T Tradeoffs, Timing Analysis, Floorplanning
- Design: FIFO's, Reed-Solomon Coder, RISC
- 4 Projects (last two projects are bigger)

- Review Lecture + Final Exam

Course Work

- Weekly Assigned Reading
  - Sections from textbook and other documents, announced on course website
- 5 Homeworks
  - Individual assignments (Mostly tool-based)
  - Approximately one week to solve
  - Solutions on paper - turn in at start of class
  - Solutions on Blackboard - post before deadline
- 5 Projects
  - Larger individual or team-based tool-based assignments
  - One or two weeks to solve
  - Post solutions in dropbox on course website
- 2 Exams
  - Each exam is preceded by a review lecture

Grading

- Steady and persistent effort is rewarded
  - Homeworks: 15%
  - Projects: 45%
  - Midterm: 20%
  - Final Exam: 20%

- Late policy:
  - Late assignments will not be accepted unless prior arrangements have been made with the instructor.
  - Technical issues are not a valid reason for a late assignment.

Honor Code

- The Virginia Tech Honor Pledge is: "*I have neither given nor received unauthorized assistance on this assignment*."
  - This applies to Homeworks, Projects and Exams

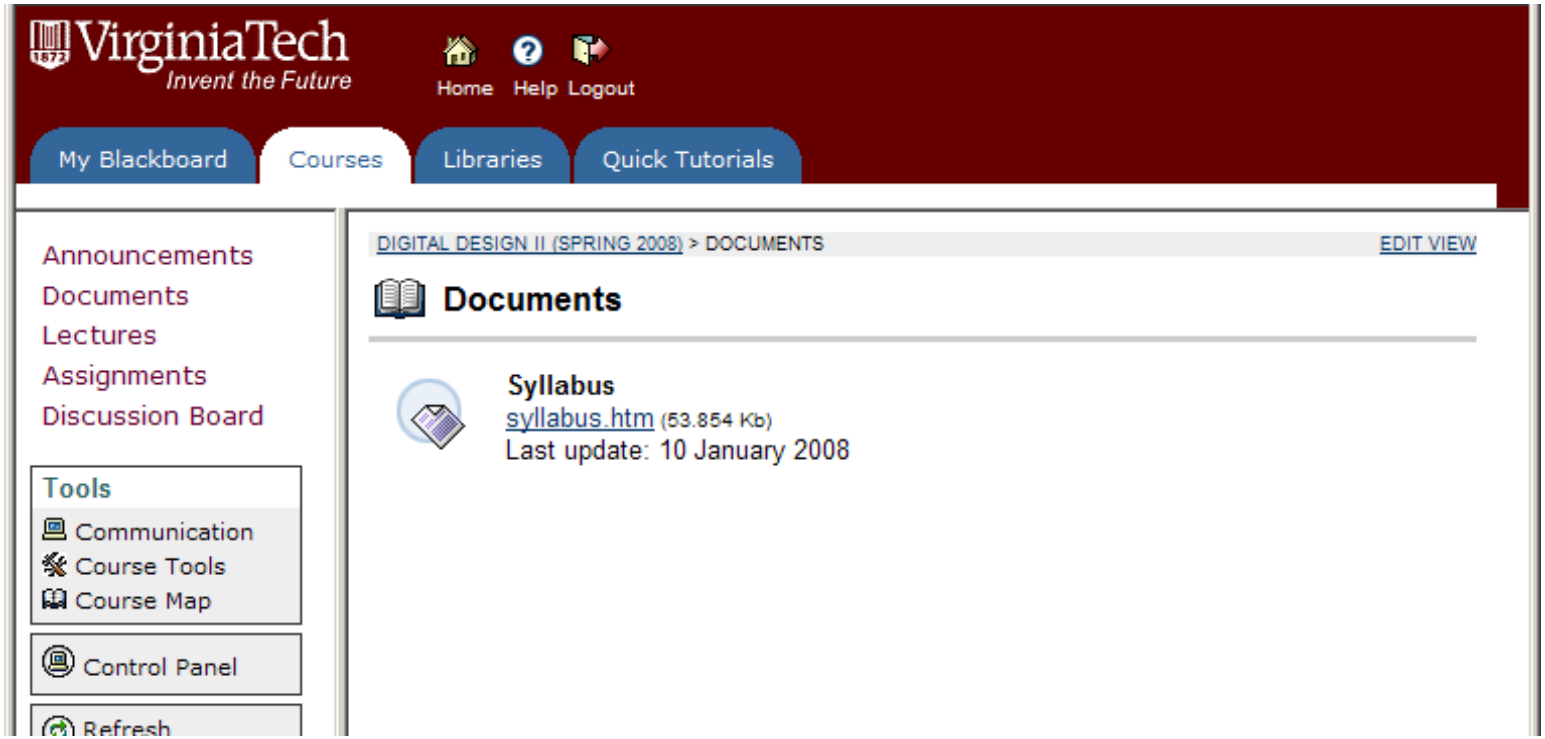
- Violations are reported to the Office of the Honor System.

Special Needs and Missing Exams

- Exam days
  - Midterm: 28 February 2007
  - Final Exam: 5 May 2008
  - If there is any conflict with the dates above, notify the instructor ASAP
- Accommodations are available for students with disabilities, religious and ethnic holidays, and illness.
- Documentation requirements are listed in the syllabus.

Syllabus

- All of this material is listed in the syllabus



About the Lectures

- Lectures will make use of slides
  - Slides are **great** !
    - Nice pictures to explain concepts
    - Good addition for course text
    - I can annotate them with a tablet PC
    - I can switch to the tools and listings mid-lecture
  - Slides are **horrible** !
    - They make me teach 30% faster (really)
    - They give you the sense that this is all easy stuff (it's not)
    - They make you fall a sleep
    - They make me lazy and teach like a drone
    - They make me waste time looking for clipart. Like this one.

- Slides are a two-edged sword
  - I encourage you to be active and take notes
  - I may fall back to blackboard-based teaching occasionally

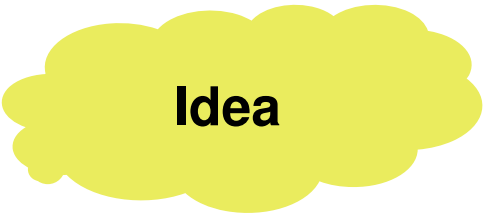
Introduction to Verilog

- This Lecture
  - Why HDL's?
  - Brief history of HDL's
  - Modern Digital Design Flow
  - Homework 1
- Thursday
  - Hierarchical Modeling
  - Review Finite State Machines

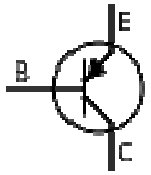
Why HDLs?

1950's

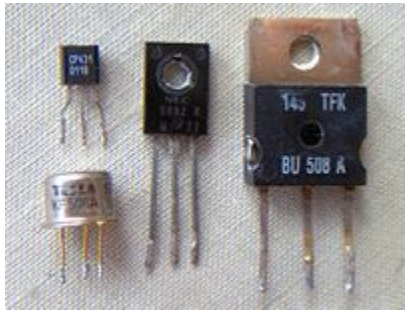
Digital Design
Let's Build An AND!



Draw Schematic



Select Components

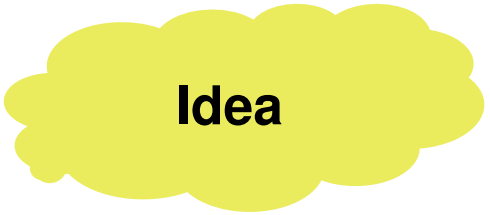


Implement

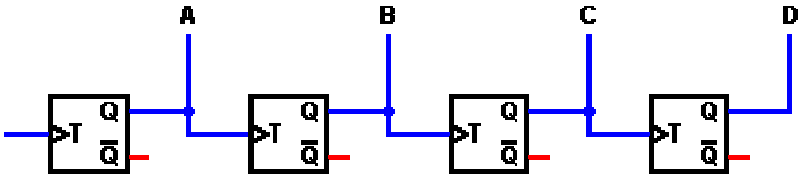
Why HDLs?

1980's

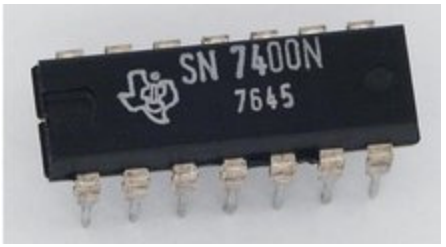
Digital Design
Let's Build A Counter!



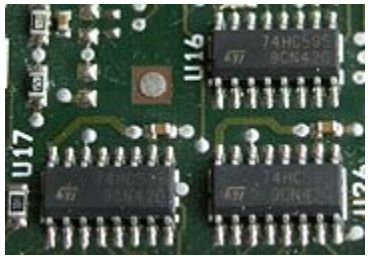
Draw Schematic



Select Components



Implement

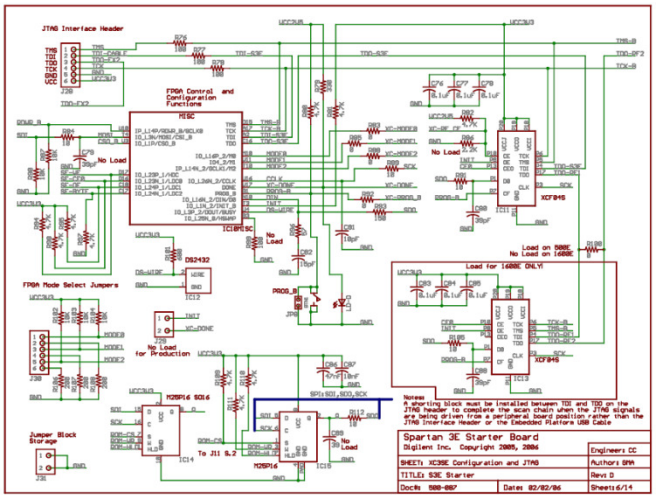


Why HDL's

2008's

Digital Design
Let's Build an iPod!

Draw Schematic



Configuration logic of your Spartan 3E
(so, after drawing this you
haven't even started on the iPod Function ..)

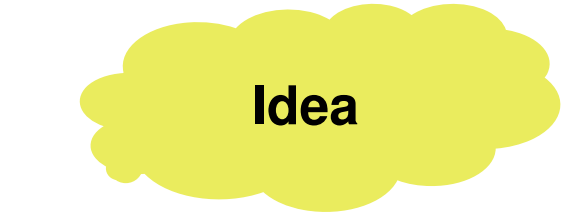
Why HDL's

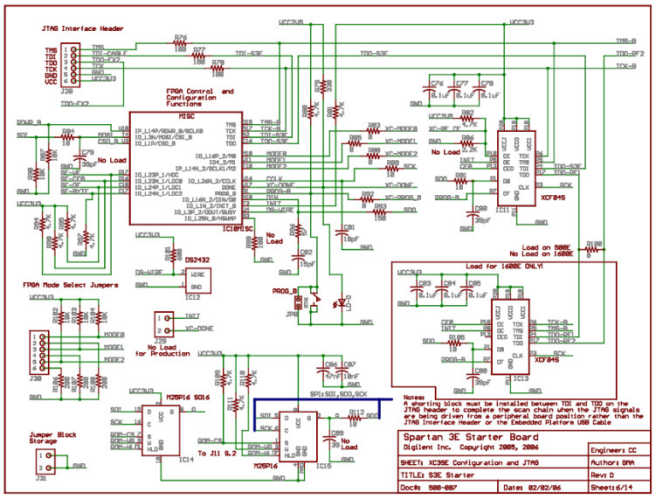
2008's

Digital Design
Let's Build an iPod!

Draw Schematic

Duh ?





Schematic

- A schematic has components and wires
  - Components: Gates, Resistors, (Leds, LCDs), Chips
  - Wires: Interconnecting these components
- A schematic has inputs and outputs
  - So a schematic can also be used as component
  - One schematic can include another one: *Hierarchy*
- A schematic is a representation of a *netlist*

Okay, HDL instead of schematics. Now what?

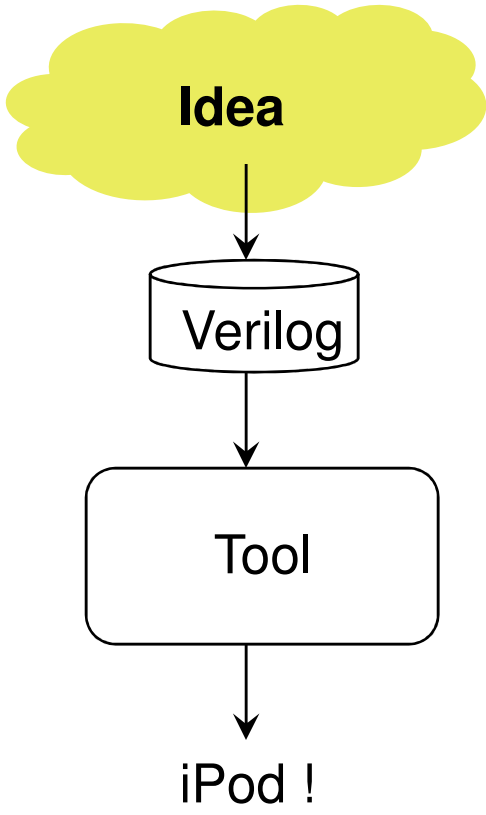
- We can build store HDL's in files
- Files can be parsed and processed by *tools*.

Digital Design
Let's Build An XYZ!

"Write HDL" - Designer Activity

"Select Components" - Tool activity (*synthesis)*

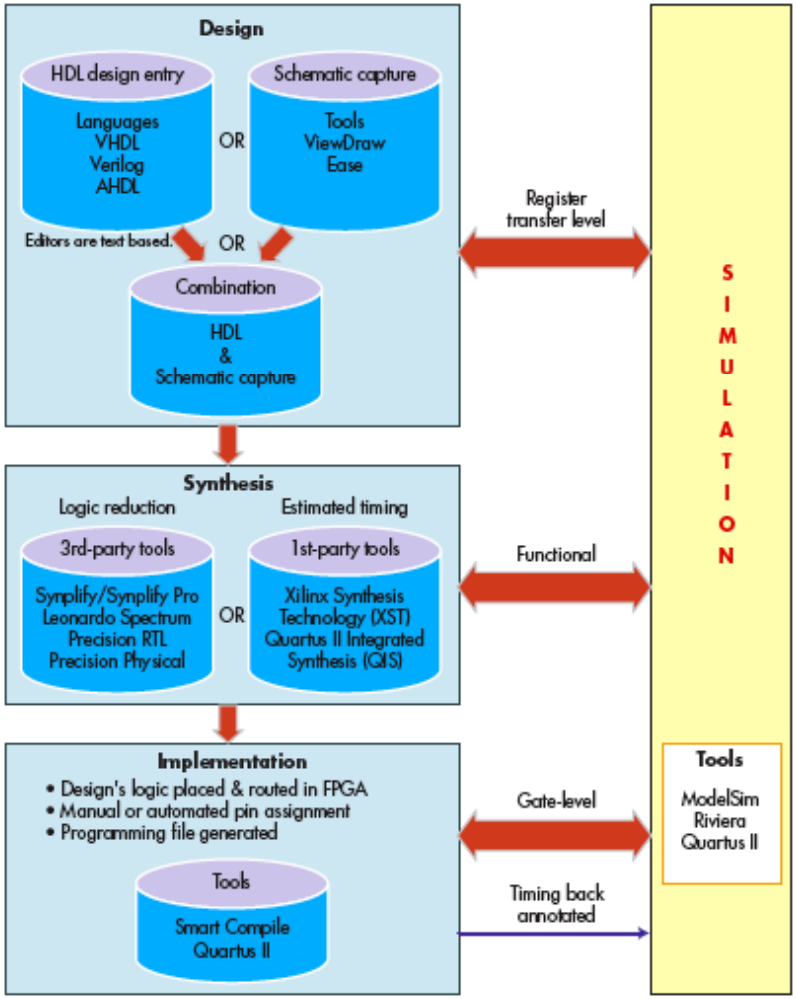
"Implement" - Tool activity



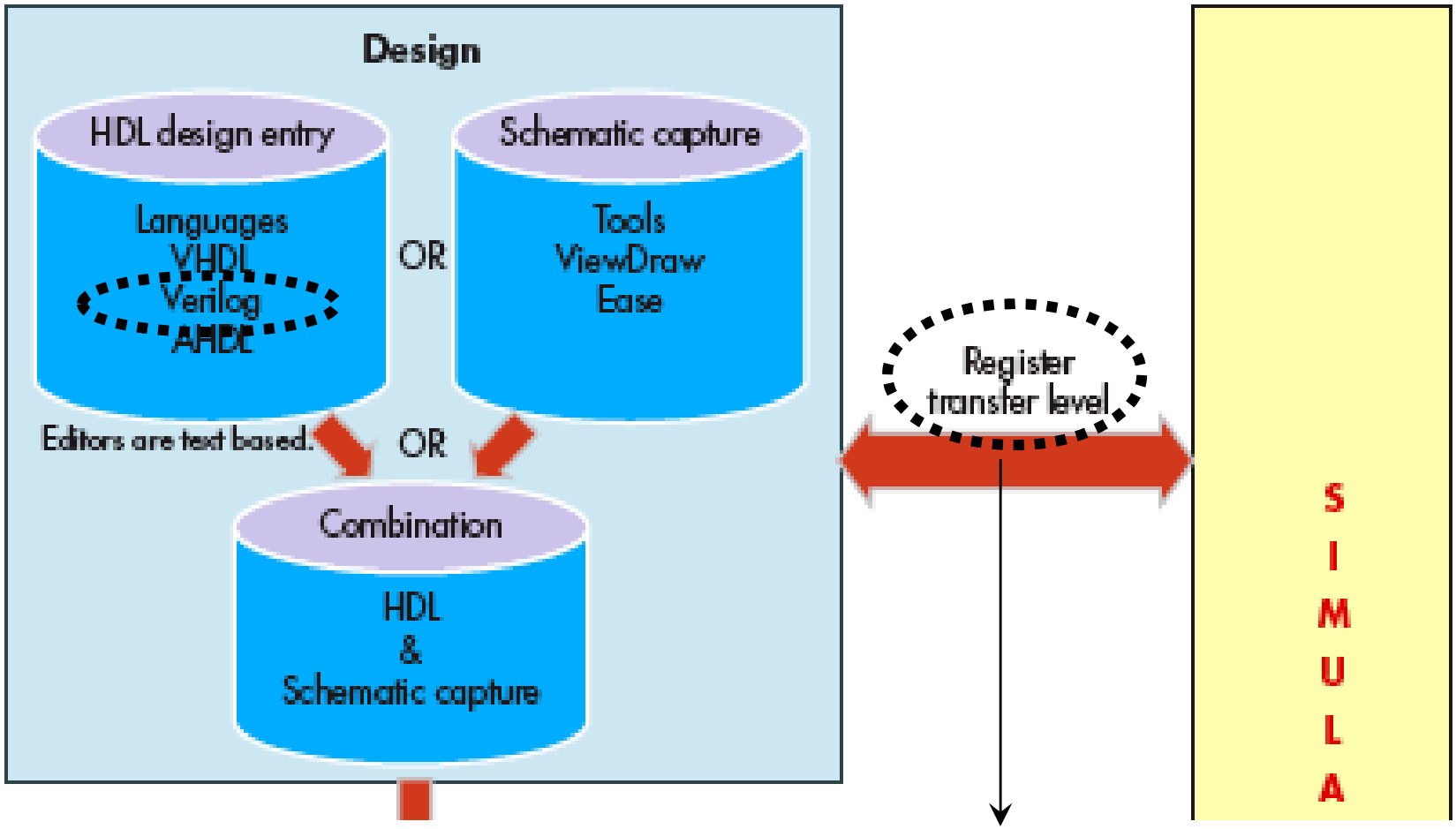
With tools, we can create a *design flow*

- Example of an FPGA Design Flow

"The art of FPGA construction"
Embedded.com
[Smith 08]

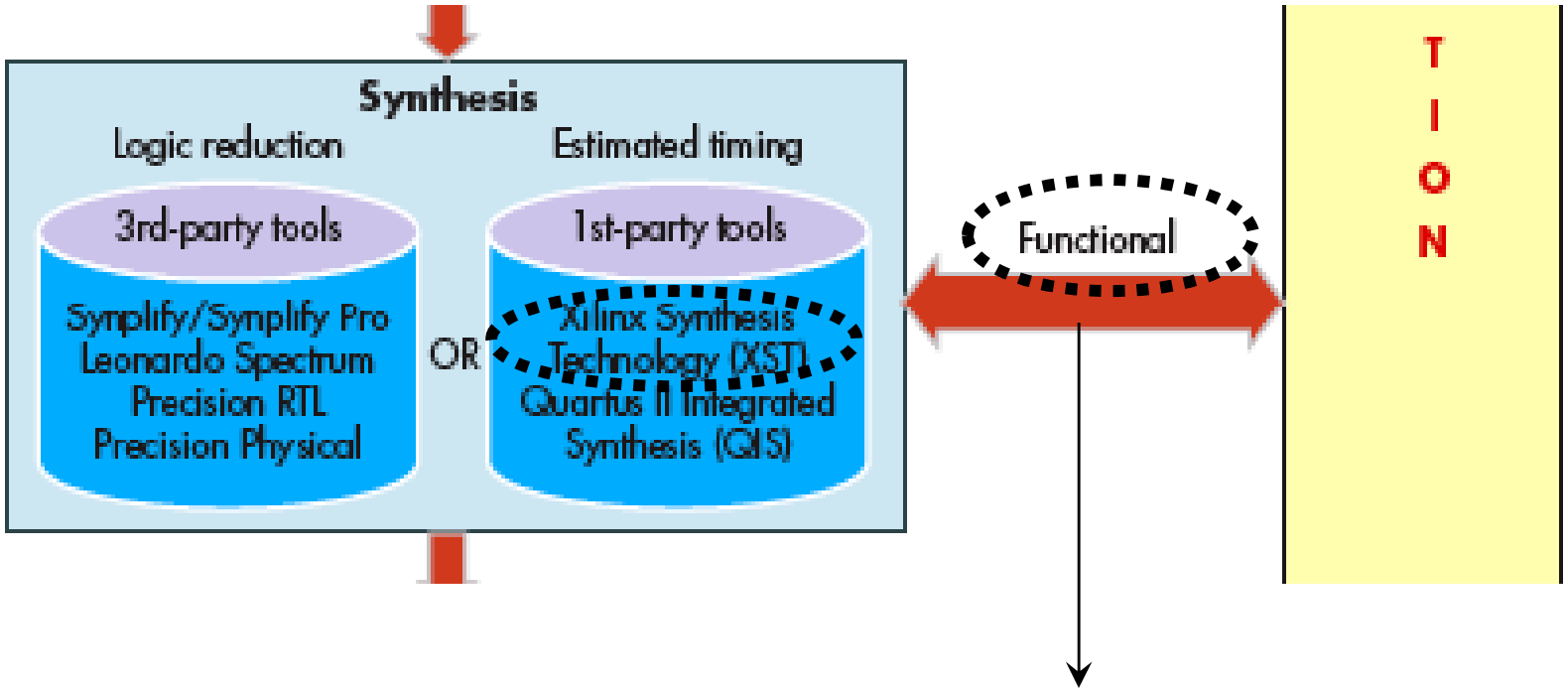


The 'Design' part:



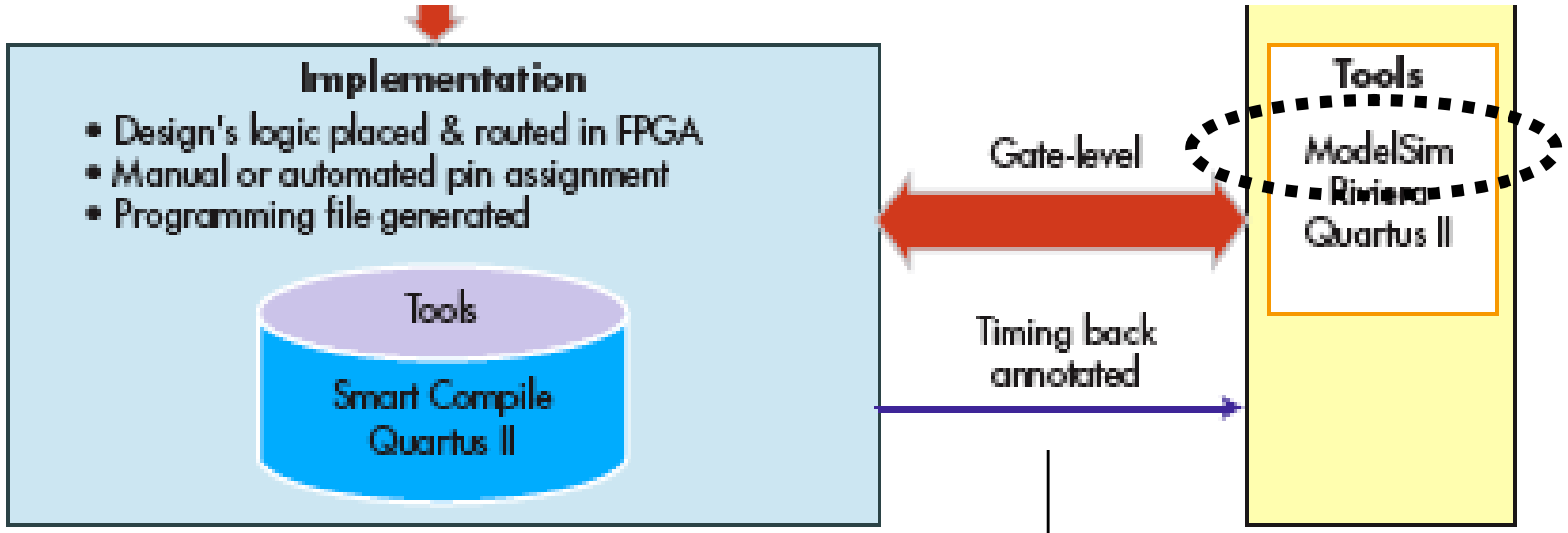
Simulation in terms of expressions and variables
(making abstraction of circuit components)

The 'Synthesis' part:



Simulation in terms of circuit components (making abstraction of time needed to perform computations)

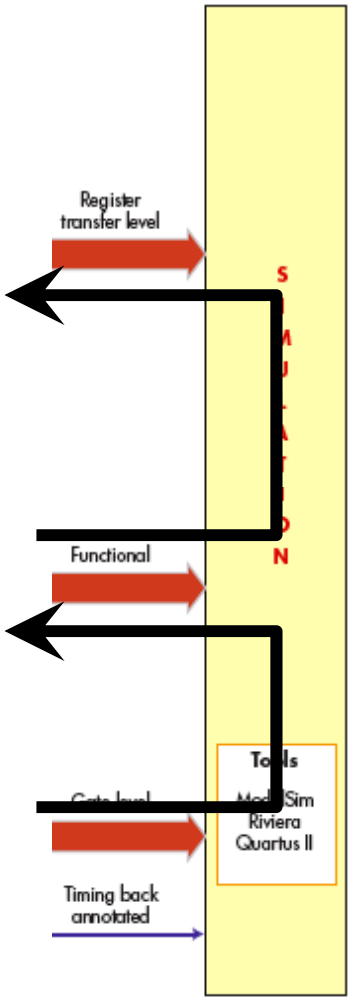
The 'Implementation' part:



Simulation in terms of circuit components and the detailed interconnection pattern and component placement

The 'Verification' Part

- 'go back' to an earlier design level if
  - the circuit is incorrect (RTL, functional)
    - reported by the simulator
  - the circuit is too big (functional, gate-level)
    - reported by the synthesis tool
  - the circuit is too slow (gate-level)
    - reported by the simulator, with additional input from the synthesis tool

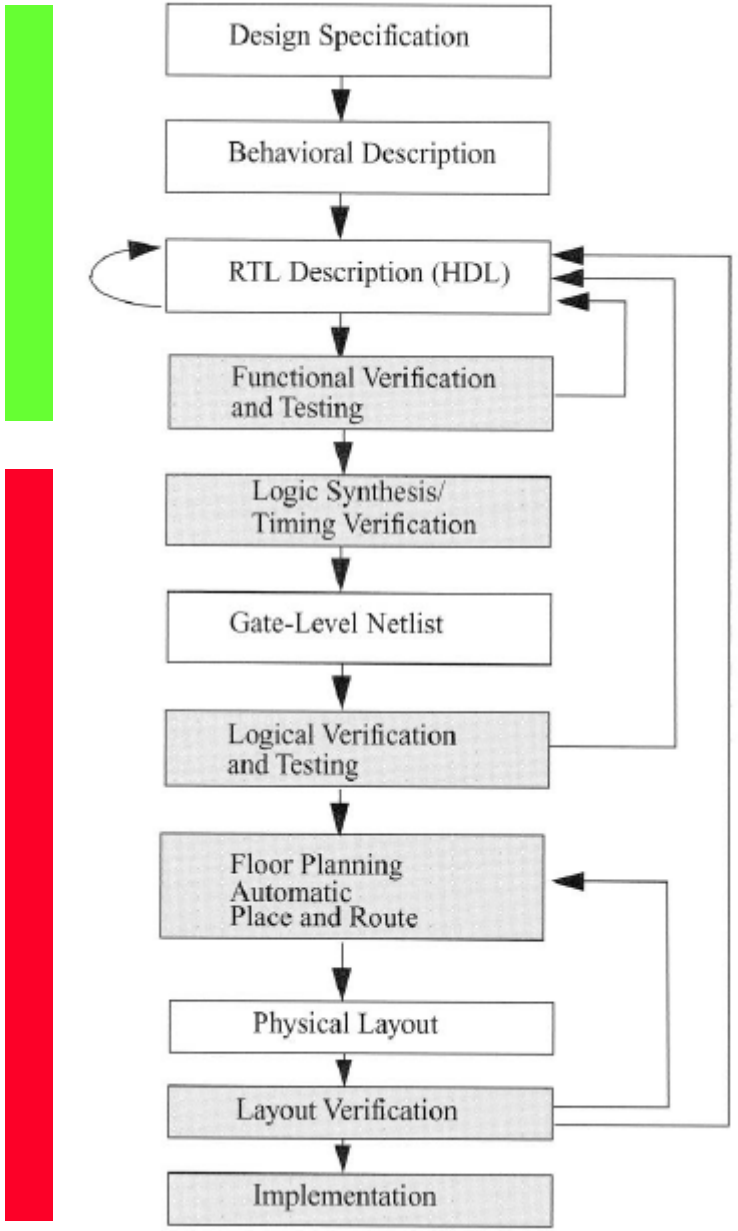


- The design flow works in two directions
  - down: design, refine and implement digital circuits starting from HDL
  - up: verify functionality and performance and correct mistakes

An ASIC Design Flow

This is (almost) the same as for FPGA

This is different than for FPGA



Core area for DD-II

Important for DD-II

Less Important for DD-II (take VLSI for this)

Summary

- A digital designer is concerned with mapping ideas into digital logic components (gates, registers)
- Schematic capture of a netlist is a very convenient way to express the design as a *model*
- HDLs developed as a replacement for schematics
- HDLs are the entry of a modern digital design flow
  - Synthesis: convert textual form to digital logic components
  - Implementation: map digital logic components into technology
  - Verification: use simulation to verify functionality and timing of the implementation