Testing & Verification of Digital Circuits

ECE/CS 5745/6745

Hardware Verification using Symbolic Computation

- Instructor: Priyank Kalla (kalla@ece.utah.edu)
- 3 Credits
- Mon, Wed, 1:25-2:45pm, WEB L105
- Office Hours: Mon: 3-4pm, or by appointment
- HWs and Programming Assignments + 1 term-project
- Textbook: None required, reference books listed on class website

Agenda for Today

- Introduction to Test, Validation & Verification
- Course Syllabus Overview
- Course Organization
- Requirements, Responsibilities, grading policy, etc.
- Discussions....

Course Organization

- Grading Policy: HW: 50%; Project - 50%
- Course Materials: Lecture notes, slides, tutorials, and papers
- `http://www.ece.utah.edu/~kalla/index_6745.html`
- Pre-requisites: Basic Digital Design + programming experience - I'll teach the rest!
- HW: some problem solving + some experiments with tools + some CAD tool programming and design.
- Projects: Theoretical study, experiments, CAD tool implementation, preliminary investigations on a research problem...

VLSI Circuit Realization Process

- Customer's requirements
- Design Conceptualization by customers and designers
- Design Specifications (usually in English)
- Design Descriptions - VHDL, Verilog, SystemC, . . .
- Circuit realization: Synthesis, Custom Design
- Marketing.....
- Where do Test and Verification fit-in?

VLSI Circuit Realization Process

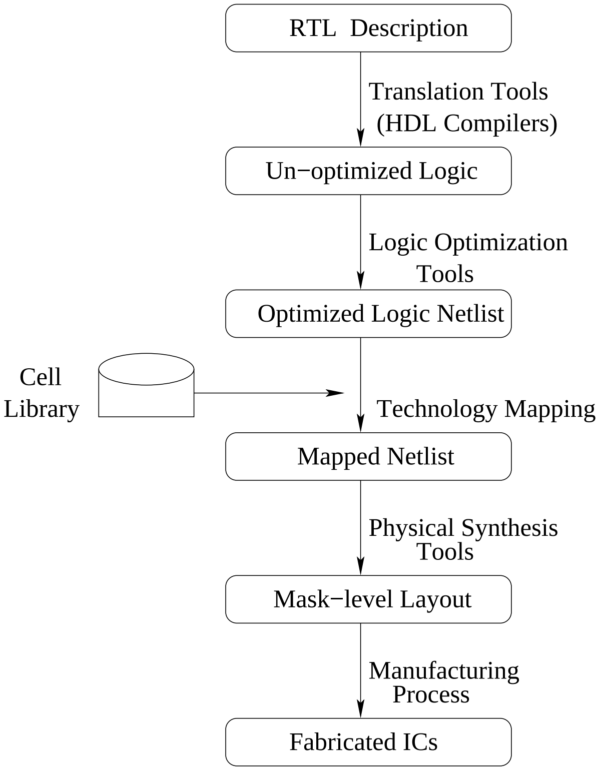
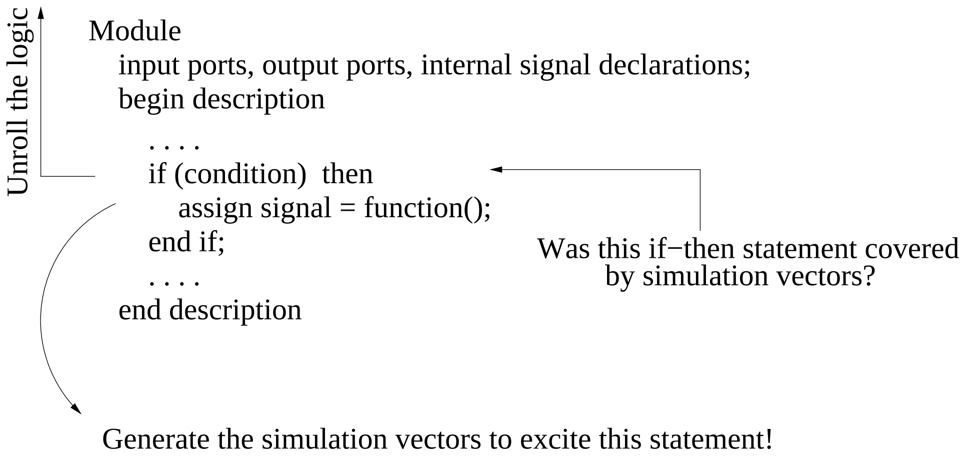


Figure 1: The VLSI Realization Process: A Typical design synthesis flow.

Design Validation

- Is the **specification** fool-proof?
  - Bugs: mis-understanding, incorrect modeling, oversight...
  - How to ascertain validity of the specs?
  - Simulate (randomly?) and observe ... too slow
  - Mathematically infer validity..... computationally complex
  - Not everyone knows/likes "math"
  - Today's solutions: little-bit of math + little-bit of simulation + little-bit of luck!
- Topics on design validation: Simulation, coverage metrics, property verification, model checking, among others..

Enhancing the Power of Simulation



Unroll the logic

```
Module
   input ports, output ports, internal signal declarations;
   begin description
      . . . .
      if (condition)  then
         assign signal = function();
      end if;
      . . . .
   end description
```

Was this if−then statement covered
by simulation vectors?

Generate the simulation vectors to excite this statement!

Figure 2: Enhancing RTL code coverage.

Formal Design Verification

- Simulation is incomplete - Remember Pentium Bug?
  - Mathematicians say: “Mathematically prove that your system is fool-proof!”
  - Theorem Proving and Model Checking.....
  - Computationally complex, but good progress
  - While mathematicians argue, industry simulates.....
  - Proof of equivalence is (somewhat) manageable
  - Our focus: Implementation Verification via Equivalence Checking.

Verification: Equivalence Checking

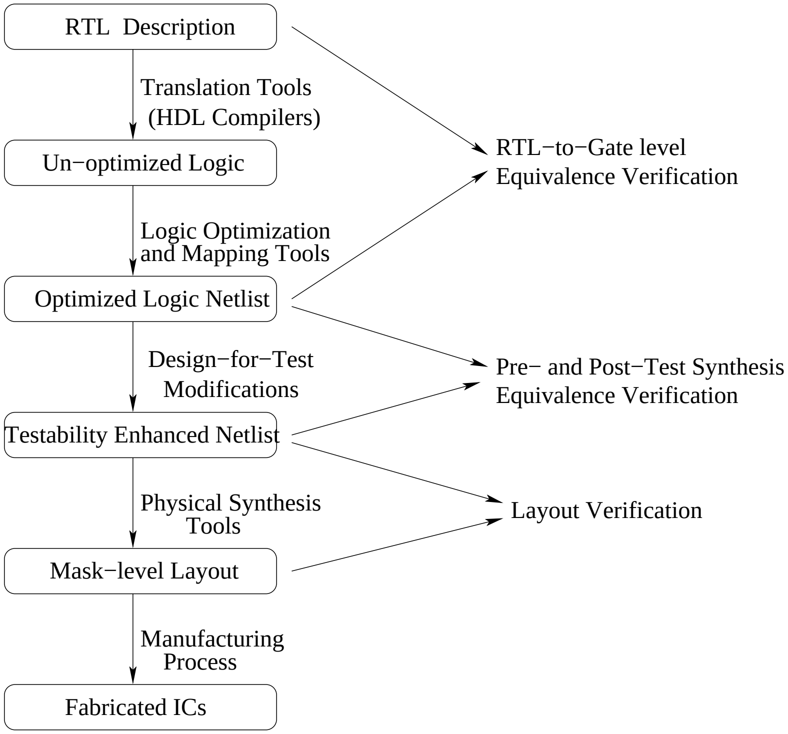


Figure 3: Typical implementation verification flow.

Equivalence Checking Setup

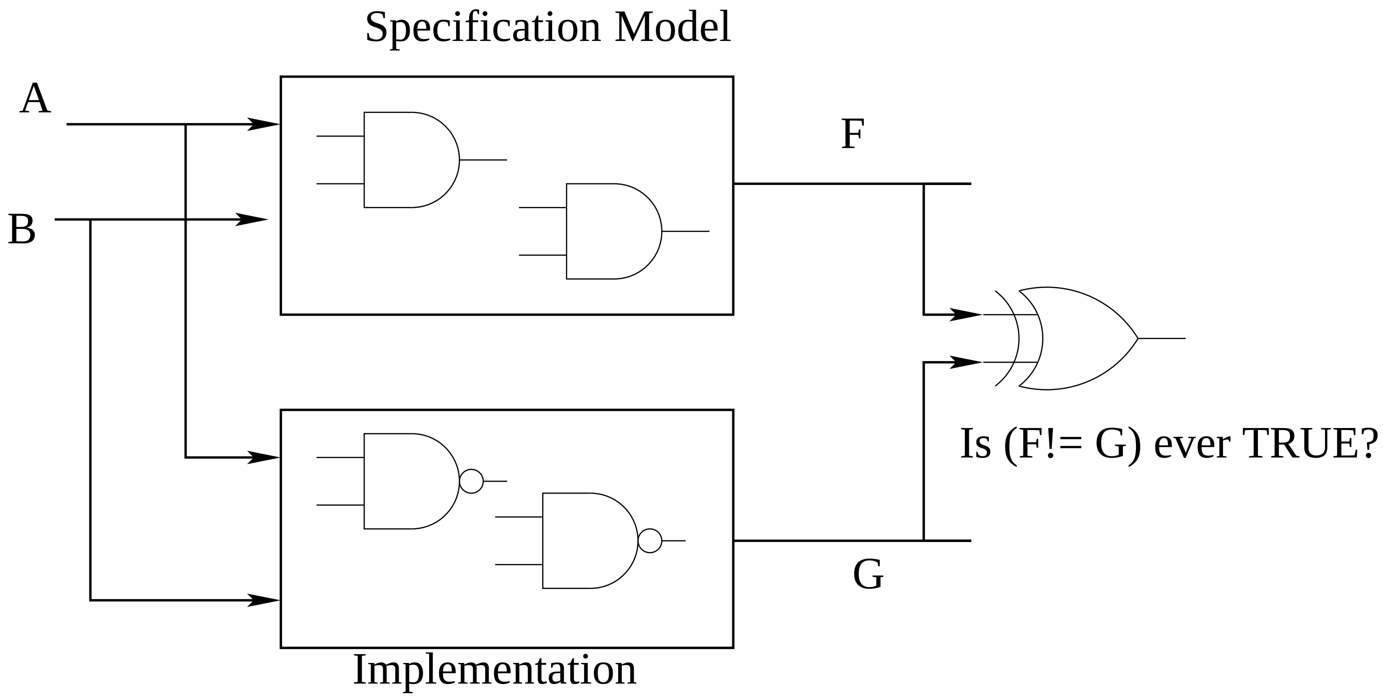


Figure 4: Equivalence Checking Setup: A Miter

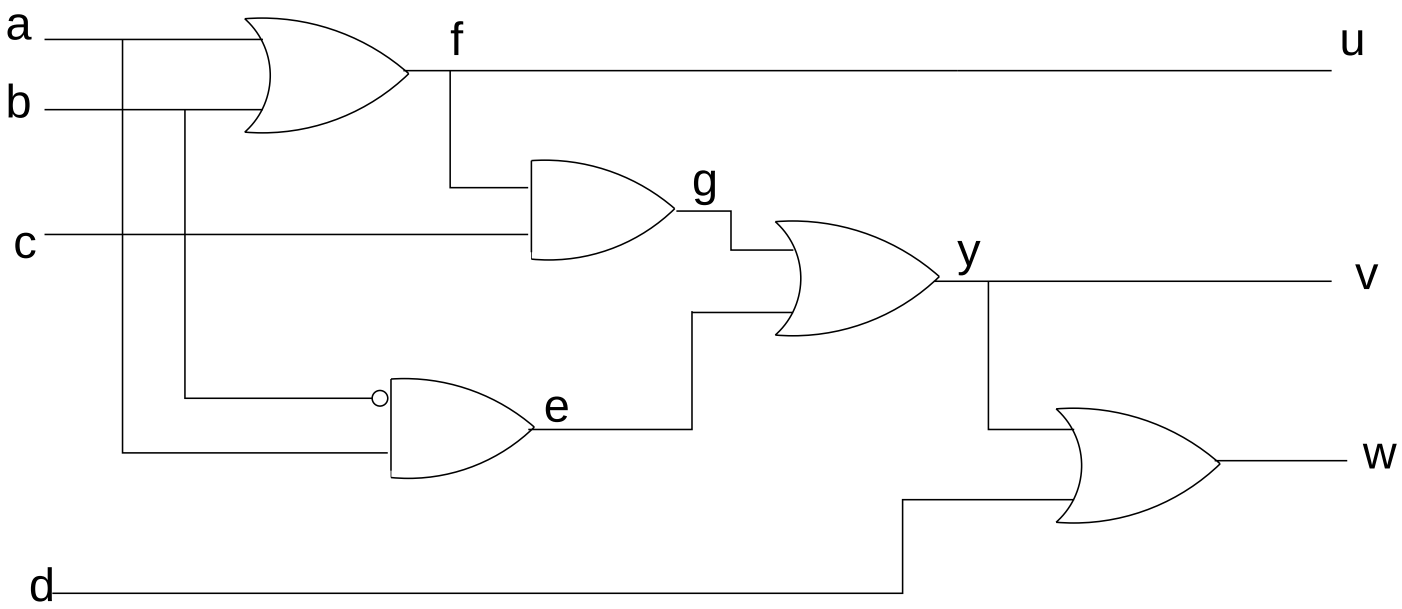
Equivalence Checking Examples

```
module verify(a, b, c, f, y, X, Z, F)
input a, b, c; // Boolean
input [3:0] X; // Bit-vectors
output f, y;
output [3:0] F, Z;

assign f = a'bc + ab'c + abc' + abc;
assign y = ab + ac + bc; // a(b+c) + bc

assign F[3:0] = 8 * X[3:0] * X[3:0] + 8* X[3:0];
assign Z[3:0] = 4'b0000;
```

VLSI Testing

- Testing a fabricated chip for manufacturing defects.
- Physical Failures and Fabrication Defects due to:
  - Shorts or opens on wires.... unreliable metalization
  - Electromigration over a period of time
  - Shorts (fusing) of Source-Drain
  - Improper ion-implantation... slow switching
  - Ever heard of latch-up?
  - Reliability of a manufacturing process in general

Testing: An Example


a
b
c
d
f
g
e
y
u
v
w