IP Trust & IP Security

- **IP Trust**
  - Detect ***malicious*** circuits inserted by IP designers
    - **Goal to Verify Trust**: Protect IP buyers, e.g., SoC integrators
- **IP Security**
  - Information leakage, side-channel leakage, backdoors, functional bugs and flaws, illegal IP use/overuse, etc.
    - **Goal to Verify Security**: Protect application

IP Trust

- IPs from untrusted vendors need to be verified for trust before use in a system design
- **Problem statement:** How can one establish that the IP does exactly as the specification, nothing less, nothing more?

- **IP Cores**:
  - Soft IP, firm IP and hard IP

- **Challenges**:
  - No known golden model for the IP
    - Spec could be assumed as golden
  - Soft IP is just a code so that we cannot read its implementation

Approaches for Pre-synthesis

- **Formal verification**
  - Property checking
  - Model checking
  - Equivalence checking
- **Coverage analysis**
  - Code coverage
  - Functional coverage

Formal Verification

- **Formal verification**
  - Ensuring IP core is exactly same as its specification
  - Three types of verification methods
    - **Property checking**: Every *requirement* is defined as assertion in testbench and is checked
    - **Equivalence checking**: Check the equivalence of RTL code, gate-level netlist and GDSII (Graphic Design System) file
    - **Model checking**
      - System is described in a formal model (C, HDL)
      - The desired behavior is expressed as a set of properties
      - The specification is checked against the model

Coverage Analysis

- **Code coverage**
  - **Line coverage**
    - **Show which lines of the RTL have been executed**
  - **Statement Coverage**
    - **Spans multiple lines, more precise**
  - **FSM Coverage**
    - **Show which state can be reached**
  - **Toggle**
    - **Each Signal in gate-level netlist**
- **Function coverage**
  - **Assertion**
    - **Successful or Failure**

Suspicious Parts

- **If one of the assertions fails, the IP is assumed untrusted.**
- **If coverage is not 100%, *uncovered* parts of the code (RTL, netlist) are assumed suspicious.**

IC (System) Trust

- **Objective**:
  - Ensure that the *fabricated chip/system* will carry out only our desired function and nothing more.
- **Challenges**:
  - **Tiny**: several gates to millions of gates
  - **Quiet**: hard-to-activate (rare event) or triggered itself (time-bomb)
  - **Hard to model**: human intelligence
  - Conventional test and validation approaches fail to reliably detect hardware Trojans.
    - Focus on manufacture defects and does not target detection of additional functionality in a design