Hardware Trojan

Mark Tehranipoor

Introduction to Hardware Security & Trust
University of Florida
**Threats**

<table>
<thead>
<tr>
<th>IP Vendor</th>
<th>SoC Design House</th>
<th>Foundry</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Insert h/w Trojan; hidden backdoor</strong></td>
<td><strong>Trojan in design</strong> (e.g. by tools)</td>
<td><strong>Implant Trojan</strong></td>
<td><strong>Leak secret info.</strong></td>
</tr>
<tr>
<td><strong>IP piracy (cloning)</strong></td>
<td><strong>Overproduction &amp; cloning</strong></td>
<td></td>
<td><strong>Magnetic field atk.</strong></td>
</tr>
</tbody>
</table>

**Solutions**
- Trojan-res. design; improve. detectability; trust validation
- Hardware Obfuscation; Protect IP Eval. Copy, PUF, Low-cost authentication
- SCA resistant Design; Prevent scan-based attack; variable ECC

**SoC Life-Cycle**
- Spans all stages in IC life cycle
What is Hardware Trojan?

Hardware Trojan:
- A malicious addition or modification to the existing circuit elements.

What hardware Trojans can do?
- Change the functionality
- Reduce the reliability
- Leak valuable information

Applications that are likely to be targets for attackers
- Military applications
- Aerospace applications
- Civilian security-critical applications
- Financial applications
- Transportation security
- IoT devices
- Commercial devices
- More
IC/IP Trust Problem

- Chip design and fabrication has become increasingly vulnerable to malicious activities and alterations with globalization.

**IP Vendor and System Integrator:**
- IP vendor may place a Trojan in the IP
- *IP Trust problem*

**Designer and Foundry:**
- Foundry may place a Trojan in the layout design.
- *IC Trust problem*
Hardware Trojan Threat

Any of these steps can be untrusted
Hardware Trojan Threat

Untrusted
Issues with Third IP Design

System-on-chip (SoC)

Company X

Company Y

Company Z

Company V

Company W
Issues with Third IP Design

These companies are located across the world.
There is no control on the design process.
Hardware Trojan Threat

IP Vendor

System Integrator

Manufacture

SoC Trust

Untrusted
Hardware Trojan Threat

1. Untrusted IP Vendor
2. System Integrator
3. Manufacture
4. Untrusted Foundry
5. IC Trust
ASIC Design Process – Untrusted Foundry

Design Process
- IP
- CAD Tools
- STD Cells
- Models
- Design Specification

Fabrication Process
- Fab Interface
- Mask
- Fab

Manufacturing Test Process
- Wafer Probe
- Dice & Package
- Package Test

IC Authentication: Trojan Detection and Isolation

Deploy and Monitor

Trusted
Either
Untrusted
Untrusted Designer and Foundry

Design Process
- IP
- CAD Tools
- STD Cells
- Models
- Design Specification
- Design

Fabrication Process
- Fab Interface
- Mask
- Fab

Manufacturing Test Process
- Wafer Probe
- Dice & Package
- Package Test

IC Authentication:
- Trojan Detection and Isolation

Deploy and Monitor

Trusted
- Either
- Untrusted
HW Trojan Examples / Models

Comb. Trojan Example

Seq. Trojan Example

MOLES*: Info Leakage Trojan

Comb. Trojan model

Seq. Trojan Model

*Lin et al, ICCAD 2009
Why is detection of hardware Trojans very difficult?
Trojan Attacks → BIGGER verification challenge!
Silicon Back Door

- Adversary can place an Antenna on the fabricated chip
- Such Trojan cannot be detected since it does not change the functionality of the circuit.
- Adversary can send and receive secret information
- Adversary can disable the chip, blowup the chip, send wrong processing data, impact circuit information etc.
Silicon Time Bomb

Untrusted Hardware

Counter

Finite state machine (FSM)

Comparator to monitor key data

Wires/transistors that violate design rules

- Such Trojan cannot be detected since it does not change the functionality of the circuit.
- In some cases, adversary has little control on the exact time of Trojan action
- Cause reliability issue
Thousands of chips are being fabricated in untrusted foundries.
Comprehensive Attack Model

3.1. Comprehensive Attack Models

Hardware Trojans can be injected at any phase during design or fabrication by different adversaries, which leads to different adversarial models. Typically, the entire design and fabrication procedure of a SoC chip can be divided into three main phases: IP core development, SoC development, and fabrication. Therefore, three types of companies, third-party IP vendors, SoC developer, and foundry, have opportunities to insert hardware Trojans. Only two attack scenarios are presented in [Rostami et al. 2013], while Table I illustrates all seven possible attack models for the potential hardware Trojan threats.

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
<th>3PIP Vendor</th>
<th>SoC Developer</th>
<th>Foundry</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Untrusted 3PIP vendor</td>
<td>Untrusted</td>
<td>Trusted</td>
<td>Trusted</td>
</tr>
<tr>
<td>B</td>
<td>Untrusted foundry</td>
<td>Trusted</td>
<td>Trusted</td>
<td>Untrusted</td>
</tr>
<tr>
<td>C</td>
<td>Untrusted EDA tool or rogue employee</td>
<td>Trusted</td>
<td>Untrusted</td>
<td>Trusted</td>
</tr>
<tr>
<td>D</td>
<td>Commercial-off-the-shelf component</td>
<td>Untrusted</td>
<td>Untrusted</td>
<td>Untrusted</td>
</tr>
<tr>
<td>E</td>
<td>Untrusted design house</td>
<td>Untrusted</td>
<td>Untrusted</td>
<td>Trusted</td>
</tr>
<tr>
<td>F</td>
<td>Fabless SoC design house</td>
<td>Untrusted</td>
<td>Trusted</td>
<td>Untrusted</td>
</tr>
<tr>
<td>G</td>
<td>Untrusted SoC developer with trusted IPs</td>
<td>Trusted</td>
<td>Untrusted</td>
<td>Untrusted</td>
</tr>
</tbody>
</table>

Each model is described as follows:

— Model A (i.e., Untrusted 3PIP Trojan model): With semiconductor scaling at very deep submicron levels, more functions (including digital, analog, mixed-signal, and radio-frequency) originally integrated on a board level, are now being placed on a single chip substrate (i.e., System-on-Chip or SoC). It is almost impossible for SoC developers to develop all necessary IPs in-house so they have to purchase some third party IP (3PIP) cores, which could contain hardware Trojans. This adversarial model is very common today as SoC chips are widely used.

— Model B (i.e., Untrusted fab or fabless design house Trojan model): Fabless design houses outsource the fabrication to off-shore third-party foundries with advanced process technologies. An attacker in the foundry can insert Trojans into a design by manipulating the lithographic masks. These Trojans are in the form of addition, deletion or modification of gates. Since foundry has the access to all layers of the design, they can inject either untargeted Trojans to produce random failures or targeted Trojans after carefully reverse engineering to create intended malfunctions. This is a difficult situation for the IC design companies who not only wish to push performance to the edge by using off-shore state-of-the-art technologies but also want to guarantee security for critical applications. The model has been discussed and studied significantly in academia in the last decade.

— Model C (i.e., Untrusted SoC developer Trojan model): Since the complexity of SoC design has increased significantly, more specialized engineers and tools must be involved during SoC design. The hardware Trojan threats could be from untrusted third-party commercial Electronic Design Automation (EDA) tools or rogue designers (also called insider threats).
Trojan Taxonomy
Trojan Taxonomy

Trojan Classification

Physical Characteristics
- Type
- Size
- Distribution
- Structure

Activation Characteristics

Action Characteristics
Examples for Layout Level Trojans
Example: Type

**Functional**
- Addition or deletion of components
- Sequential circuits
- Combinational circuits
- Modification to function or no change

**Parametric**
- Modifications of existing components
  - Wire: e.g. thinning of wires
  - Logic: Weakening of a transistor, modification to physical geometry of a gate
  - Modification to power distribution network
- Sabotage reliability or increase the likelihood of a functional or performance failure
Example: Size

- **Size:**
  - Number of components added to the circuit
    - Small transistors
    - Small gates
    - Large gates

- In case of layout, depends on availability of:
  - Dead spaces
  - Filler cells
  - Decap cells
  - Change in the structure
Example: Distribution

- **Tight Distribution**
  - Trojan components are topologically close in the layout

- **Loose Distribution**
  - Trojan components are dispersed across the layout of a chip

**Distribution of Trojans depends on the availability of dead spaces on the layout**
Example: Structure

- The adversary may be forced to regenerate the layout to be able to insert the Trojan, then the chip dimensions change.
  - It could result in different placement for some or all the design components.

- A change in physical layout can change the delay and power characteristics of chip.
  - It is easier to detect the Trojan.
Trojan Taxonomy: Activation

- Externally-activated
  - Receiver/Antenna
  - Access Data
- Internally-activated
  - Always on
  - Condition-based
    - Sensor-based
      - Temp
      - Volt
      - Ext.
    - Logic-based
      - Internal state
      - Input
      - Cnt/clock
      - Data
      - Instruction
      - Interrupt
Activation: Internally Activated

Internally-activated

Always on

Condition-based

Counter

Trojan

Clk

Trojan
Trojan Taxonomy: Action

Action Characteristics

- Modify Function: Add, Bypass
- Modify Specification: Defects, Reliability
- Transmit-information
Example: Action

Modify Specification:
Noise, Delay and Temperature

Trojan

Modify Function
IP Trust & IP Security

**IP Trust**
- Detect *malicious* circuits inserted by IP designers
  - **Goal to Verify Trust**: Protect IP buyers, e.g., SoC integrators
- Focus of this lecture

**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
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 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
Destructive Approach: Expensive and time consuming

- Reverse engineering to extract layer-by-layer images by using delayering and Scanning Electron Microscope
- Identify transistors, gates and routing elements by using a template-matching approach – needs golden IC/layout
Classification of Trojan Detection Approaches

- **Non-destructive Approach**
  - **Run-time monitoring**: Monitor abnormal behavior during run-time
    - Exploit pre-existing redundancy in the circuit
    - Compare results and select a trusted part to avoid an infected part of the circuit.
  - **Test-time Authentication**: Detect Trojans throughout test duration.
    - Logic-testing-based approaches
    - Side-channel analysis-based approaches
Hardware Trojan Benchmarks

• A set of trust benchmarks for researchers in academia, industry, and government is needed to
  — Provide a baseline for examining diverse methods developed
  — Establishing a sound basis for the hardness of each benchmark instance
  — Help increase reproducibility of results by others who intend to employ certain methodologies in their design flow

• See NSF supported Trust-Hub website (www.trust-hub.org)
  — Complete taxonomy of Trojans
  — More than 120 trust benchmarks available which were designed at different abstraction levels, triggered in several ways, and have different effect mechanisms
  — More than 300 publications used these benchmarks
Logic Testing Approach

- **Logic-testing approach** focuses on test-vector generation for
  - Activating a Trojan circuit
  - Observing its malicious effect on the payload at the primary outputs
  - Both functional and structural test vectors are applicable.

**Pros & Cons:**

- **Pros:**
  - Straight-forward and easy to differentiate

- **Cons:**
  - The difficulty in exciting or observing low controllability or low observability nodes.
  - Intentionally inserted Trojans are triggered under rare conditions. (e.g., sequential Trojans)
  - It cannot trigger Trojans that are activated externally and can only observe functional Trojans.
Functional Test Deficiency

- Functional patterns could potentially detect a “functional” Trojan.
  - Exhaustive test would be effective, but certainly not applicable for large circuits
  - E.g. 64 input adder $\rightarrow 2^{65}$ input combination (including carry in)
  - $2^{65} > 10^{18}$ – This is impractical
  - 100MHz is used $\rightarrow 10^{10}$ s $\rightarrow 317$ years
  - Only a few and more effective patterns are used $\rightarrow$ Trojan can escape.
  - The fault coverage is low for manufacturing test

- In practice, structural tests are used.
**Functional Testing**

**Feasible Trojan space inordinately large!**

*Deterministic* test generation infeasible

A statistical approach is, more effective

- **MERO: A Statistical Approach**
  - Find the rare events in the circuit
  - Generate vectors to trigger each rare node *N times*
  - Provides high confidence in detecting unknown Trojans!

**Trojan Trigger Condition**

\[ a=0, \ b=1, \ c=1 \]

From original circuit

*Chakraborty et al, CHES 2009*
MERO

MERO:
- Generates a set of test vectors that can trigger each rare node to its rare value multiple times (N times)
- It improves the probability of triggering a Trojan activated by a rare combination of a selection of the nodes

![Graphs showing coverage and test length for different N values for c2670 and c3540 circuits.]

Fig. 15.6 Trigger coverage and Trojan coverage and test length for two ISCAS-85 benchmark circuits for different values of “N,” using the MERO approach [8]

Challenge: Triggering each net N times in a large circuit is challenging
Hardware Trojans inserted in a chip can change the power consumption characteristics.

**Partial activation** of Trojan can be extremely valuable for power analysis.

The more number of cells in Trojan is activated the more the Trojan will draw current from power grid.

Golden chip required!
Side-Channel Trojan Detection

- Side-Channel Approach for Trojan Detection relies on observing Trojan effect in physical side-channel parameter, such as switching current, leakage current, path delay, electromagnetic (EM) emission.
  - Due to process variations, it is extremely challenging to detect the Trojan by considering $F_{\text{max}}$ or $I_{\text{DDT}}$ individually.
Side-channel Signals

- All the side-channel analyses are based on observing the effect of an inserted Trojan on a physical parameter such as:
  - **IDDQ**: Extra gates will consume leakage power.
  - **IDDT**: Extra switching activities will consume more dynamic power.
  - **Path Delay**: Additional gates and capacitance will increase path delay.
  - **EM**: Electromagnetic radiation due to switching activity

- **Pros & Cons**
  - **Pros**: It is effective for Trojan which does not cause observable malfunction in the circuits.
  - **Cons**: Large process variations in modern nanometer technologies and measurement noise can mask the effect of the Trojan circuits, especially for small Trojan.

Golden chip required!
Sensitivity Metric

**Improving Detection Sensitivity**

\[
Sensitivity = \frac{I_{\text{tampered}} - I_{\text{original}}}{I_{\text{original}}} \times 100\%
\]
## Comparing Approaches

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Testing</td>
<td>Side-Channel Analysis</td>
</tr>
<tr>
<td>Robust under process noise</td>
<td>Difficult to generate test vectors</td>
</tr>
<tr>
<td>Effective for ultra-small Trojans</td>
<td>Large Trojan detection challenging</td>
</tr>
<tr>
<td>Effective for large Trojans</td>
<td>Vulnerable to process noise</td>
</tr>
<tr>
<td>Easy to generate test vectors</td>
<td>Ultra-small Trojan Det. challenging</td>
</tr>
</tbody>
</table>

- A combination of logic testing & side-channel analysis could provide the good coverage!
- Online validation approaches can potentially provide a second layer of defense!
Side-channel Approach

- **Multiple-parameter Trojan Detection**
  - Due to process variations, Trojan detection by $F_{\text{max}}$ or $I_{\text{DDT}}$ alone is challenging!

  ![Graph showing Avg. $I_{\text{DDT}}$ vs. Process Corners and Avg. $I_{\text{DDT}}$ vs. $F_{\text{max}}$]

- Consider the intrinsic relationship between $I_{\text{DDT}}$ and $F_{\text{max}}$

  **Golden chip required!**
Trojan Inserted into s38417 Benchmark

PP: Power Pad

3-Bit Counter Trojan
• Current difference measured from power pad 17 (Trojan-free vs Trojan-inserted)

• There is no change in layout of the circuit. Trojan was inserted in an unused space in the circuit layout.
Current (Charge) Integration Method

- Current consumption of Trojan-free and Trojan-inserted circuits

\[ Q_{\text{trojan-free}}(t) = \int I_{\text{trojan-free}}(t) \cdot dt \]

\[ Q_{\text{trojan-inserted}}(t) = \int I_{\text{trojan-inserted}}(t) \cdot dt = \int (I_{\text{trojan-free}}(t) + I_{\text{trojan}}(t)) \cdot dt \]
Power Analysis -- Challenges

- **Pattern Generation**
  - How to increase switching activity in Trojans?
  - How to reduce background noise?
  - Switching locality
  - Random Patterns
    - No observation is necessary, Similar to test-per-clock

- **Measurement Device Accuracy**
  - Measurement noise

- **Process Variations**
  - Calibration

- **On-Chip Measurement**
  - Vulnerable to attack

- **Authentication Time**
  - Trojans can be inserted randomly
Side Channel Analysis -- Delay

- Hard to detect using power analysis are:
  - Distributed Trojans
  - Hard-to-activate Trojans

- **Path delay**: A change in physical dimension of the wires and transistors can also change path delay.

- We are developing new methods that can detect additional delays on each path of the circuit.
Delay-based Methods

- Shadow-register provides a possible solution for measuring internal path delay.
- From this architecture, it can be seen that the basic unit contains one shadow register, one comparator and one result register.

**Limitations:**
- PV
- Overhead
- S-clock
- Output
Clock Sweeping Technique

- Clock sweeping involves applying a pattern at different clock frequencies, from a lower speed to higher speeds.
- Some paths sensitized by the pattern which are longer than the current period start to fail when the clock speed increases.
- The obtained start-to-fail clock frequency can indicate the delays of the paths sensitized by the patterns.
Delay Analysis -- Challenges

- Major advantage over power analysis: No activation is required.

Detection and Isolation
- How significant is the delay inserted by Trojan?
- It depends on Trojan size and type
- Location: on short paths or long paths

Pattern Generation
- Delay test patterns
- Path Coverage

Process Variations ($V_{th}$, $L$, $T_{ox}$)
- Impact circuit delay characteristics significantly
- Differentiate between Trojan and PV

- Trojan can have impact on multiple paths (an advantage over PV)
# Trojan Detection

<table>
<thead>
<tr>
<th>Trojan Classification</th>
<th>Power Analysis</th>
<th>Delay Analysis</th>
<th>Fully Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Physical Characteristics</strong></td>
<td>Type</td>
<td>Functional</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>Parametric</td>
<td>P</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>Size</td>
<td>Small</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Large</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>Distribution</td>
<td>Tight</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loose</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>Structure</td>
<td>Modify Layout</td>
<td>P</td>
</tr>
<tr>
<td><strong>Activation Characteristics</strong></td>
<td>Always-on</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Condition-based</td>
<td>Logic-based</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>Sensor-based</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Action Characteristics</strong></td>
<td>Modify Function</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modify Spec.</td>
<td>Defects</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>Reliability</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

P: Detection is possible  
D: High level of confidence
Self-similarity in Space & Time – for Trust Verification

Uncorrelated switching in time due to a seq. Trojan!

Simultaneously detects Trojan & aged/recycled ICs!

No golden chip required!!!
Since detecting Trojan is extremely challenging, design for hardware trust approaches are proposed to:

- **Improve hardware Trojan detection methods**
  - Improve sensitive to power and delay
  - Rare event removal

- **Prevent hardware Trojan insertion**
  - Design obfuscation
Rare Event Removal

- Intelligent attackers will choose low-frequency events to trigger the inserted Trojans.

- Improving controllability or observability can make rare events scarce, thereby facilitating detecting Trojans inside the design.
  - Design for Trojan test: inserting probing points
  - Inserting dummy scan flip-flops
Increasing Probability of Partial/Full Activation

- Inserting dummy FFs on path with very low activation probability
Increasing Probability of Partial/Full Activation

- Dummy scan flip-flops are inserted to control hard-to-excite nodes.

Usage:
- **Full activation**: increase controllability
- **Power-based**: generate switching activities
- **Delay-based**: activate more paths to improve coverage
Trojan Prevention-Design obfuscation

- The objective is deterring attackers from inserting Trojans inside the design.

- Design obfuscation means that a design will be transformed to another one which is functionally equivalent to the original, but in which it is much harder for attackers to obtain complete understanding of the internal logic, making reverse engineering much more difficult to perform.

- It obfuscates the state transition function to add an obfuscated mode on top of the original functionality (called normal mode).
Design obfuscation

- Specified pattern is able to guide the circuit into its normal mode.
- The transition arc K3 is the only way the design can enter normal operation mode from the obfuscated mode.
BISA: Built-In Self-Authentication

- Filling all unused spaces with a circuit that can easily test itself
Question?