**Hardware Trojans**

What is a hardware Trojan?

A deliberate and malicious change to an IC that adds or removes functionality or reduces reliability

- The modifications may be designed to leak sensitive information, personal or corporate
- The modifications may be designed to cause a system to fail at a critical time while operating in mission mode
- The modification may be designed to reduce the reliability of the IC

What makes this a challenging problem?

Adversary makes purposeful discovery highly improbable & physical inspection is very expensive

Trojan circuits are a big concern for US military

IC Trust Program initiated to investigate solutions to the problem defined by a Defense Science Board Task Force report:

Hardware Trojan Example

Missile control system

Assume a chip receives encrypted commands from an RF channel and stores the value in a register for subsequent decryption.

Adversary transmits "code" that causes activation - missile detonates before reaching its target.
Hardware Trojan Example

Adversary may try a ’stealthier’ strategy, e.g., a ’monolithic NAND gate

Many other implementations are possible, e.g. pass-gate versions, some better than others at minimizing power supply anomalies
IC Trust

Consider the vulnerabilities in a typical ASIC design flow

Steps 1-4 and 7-9 are particularly susceptible to malicious insertions and deliberate process modification because these steps include significant third party content.
IC Trust

ASIC designs may use dozens of EDA tools, whose verification of TRUST is formidable.

Insertions during the fabrication process (steps 7-9) would produce dice with differing properties.

Without testing each die, the assumption of trust based on the test results of sampled dice may be misleading.

Process modifications can cause premature aging are particularly difficult to detect.

After deployment (step 13), the IC is largely defenseless against physical tamper and side channel attacks.

In general, in order to develop a strategy to ensure IC trust, it will be necessary to make assumptions about which steps are ’trusted’ and which are ’untrusted’.

The Trojan insertion point, e.g., Soft-IP, GDS, package, etc., will determine what detection strategies are possible, and what assumptions are necessary.
TRUST: ASICs vs. COTs

ASIC
- Design
- IP Theft
- Malicious Circuits
- UNControlled

COTS
- Spec
- Controlled
- Substitution
- Hidden Features
- UNControlled

GDS
- Wafer
- Foundry (Mask & Fab)
- UNControlled

Deply & Monitor
- Package Test
- Controlled

Tamper
- Foundry (Mask & Fab)
- Dice & Package
- UNControlled
Trojan Scenarios: Soft-IP Trojans

Adversary can compromise **soft-IP** by inserting extra, hidden functionality into the netlist

Implications

- No golden model is available
- Every IC has the Trojan

Detection strategies include

- **Formal verification methods**
  
  Prove that the functionality of the IP is equivalent to some higher-level, more abstract ‘trusted’ specification

  Unfortunately, formal verification is a difficult problem (NP-complete) and can only be applied to small circuits

  There has been some recent work in this area that offers more attractive solutions
Trojan Scenarios: Hard- and Soft-IP Trojans

- Monitoring the IC using a ’trusted companion IC’

  Trusted companion IC has access to the internal state of the untrusted IC through extra pins/scan chain

  Trusted companion IC is ’programmed’ such that it knows the legal state space of the untrusted IC, and sets off an alarm and/or shuts down the IC if violated

  (This technique can be used for a second class of GDS-based Trojans)

High security applications would likely use only IP developed in-house or from trusted sources

  Following the advice from a old adage "Doctor, doctor it hurts when I do this!" -
  - Doctor replies, ’well then, don’t do that!"

Hard-IP (GDS) Trojans

  The insertion point here is the layout (GDS)

  We mentioned earlier that several modifications are possible, e.g., those designed to 1) disable/destroy, 2) to leak information and 3) decrease reliability
Trojan Scenarios: Hard-IP Trojans

Changes to the IC’s function can be implemented by

• Using existing ’white-space’ (places in the layout where there are no transistors or
  where there is a by-pass capacitor)

• ’Nudging’ gates to make space for the Trojan gates
  
  In case you were thinking about adding some type of verifiable white space filler
to prevent the first case from occurring

Parametric changes designed to reduce reliability can be implemented by thinning
wires to accelerate EM effects, by thinning oxide to accelerate TDDB, etc.

I would like to argue that when **direct control** is needed by the adversary, then **functional modifications** (the first type) are more attractive

  Why do you think this might be the case?

Let’s consider another Trojan parameter, the **size** of the Trojan

  Trojans can be very small and be effective, e.g., only a couple/three gates

  For Trojans designed to change functionality, small Trojans are risky - why?
Trojan Scenarios: Hard-IP Trojans

Therefore, I argue that Trojans designed to change functionality, e.g., destroy/disable/enable remote control, etc., are likely to be larger, 10’s to 100’s of gates to prevent discovery.

Unfortunately, the same is not true for information leakage Trojans. Since they do not, at least in an obvious way, change functionality, they can be very small and remain secure against detection.

Information leakage Trojans can ‘leak’ information in ways that are not easily detected:

• By broadcasting data as EM radiation
• By inserting data into a communication channel, that appears as error bits to a valid receiver
• By inserting data into a communication channel at a higher frequency, e.g., baud rate, than the valid receiver is expecting

Lot’s of strategies have been proposed -- all of them difficult to detect unless you test at a high level, e.g., the protocol level.
Trojan Scenarios: Hard-IP Trojans

Unlike manufacturing defects, you only need to detect ONE Trojan to yield success!

For example, if a layout-based Trojan is inserted in EVERY copy of a manufactured IC, then alternative test strategies that use MUCH larger test sets can be used.

Unfortunately, layout-based Trojans can be inserted in only a subset of the ICs (unlike soft-IP Trojans which are, by definition, in every copy).

This makes it more difficult to develop a testing method to detect them - why?

Whether the Trojan is selectively inserted or inserted into every copy depends on the application:

I argue that functionally disruptive Trojans, like the missile Trojan, are likely to be inserted into every copy of an IC -- why?

I also argue that information leakage Trojans do not need to be inserted into every copy of the IC to be useful -- why?
**Trojan Taxonomy**

Layout inserted (GDS) Trojans can take many forms but can be broken into the following categories:

- **Physical characteristics**
  - Type: Functional, Parametric
  - Distribution: Tight, Loose
  - Size: Small Transistor/wire, Large Gate/Macro

- **Activation characteristics**
  - Always on
  - Condition based
  - Sensor based: Temp., Volt., Ext.

- **Action characteristics**
  - Logic based: Add, Bypass, Modify spec.
  - Input: Internal state, Input, Cnter/Ck
  - Data: Internal state, Instruction, Interrupt
  - Defects: Reliability

---

Layout-Oriented Trojan Detection Strategies

Three basic approaches:

- **IC Deprocessing and Application of Failure Analysis Methods**
  
  The most straightforward approach is to ’deprocess’ the chip, i.e., remove one layer at a time and compare the wires and transistors with the original layout.

  The comparison can be implemented using image processing methods that compare micro-photographs of IC geometries with layout images.

  Failure analysis (FA) methods traditionally used for identifying the root cause of failure in IC can also be applied to extract geometries:
  
  - Scanning Optical/Electron Microscopy (SOM/SEM)
  - Pico-Second Imaging Circuit Analysis (PICA) and Voltage Contrast Imaging
  - Light-Induced/Charge-Induced Voltage Alternation (LIVA/CIVA)

- **Advantages: IC Deprocessing and Application of Failure Analysis Methods**
  
  Potentially highly sensitive to the presence of Trojans.

- **Drawbacks: IC Deprocessing and Application of Failure Analysis Methods**
  
  Destructive, may miss selectively-inserted Trojans, cost, effectiveness in nano
Layout-Oriented Trojan Detection Strategies

Three basic approaches:

- **Functional activation** through logic testing

  Develop an automatic test pattern generation (ATPG) strategy that generates logic vectors (patterns) to activate the Trojan.

  Given the connectivity characteristics of the Trojan to the original circuit are unknown, ATPG must be based on a heuristic.

  The heuristic chosen by most researchers is that the adversary is likely to connect the Trojan to wires in the original design that are the most difficult to control and observe.

  The rational for this is simple: the adversary wants to make accidental discovery, i.e., through manufacturing test, of the Trojan extremely unlikely.

  Therefore, many of the proposed functional activation strategies are based on deriving tests for nodes that are *random-pattern resistant*. 
Layout-Oriented Trojan Detection Strategies

Three basic approaches:

- **Functional activation** through logic testing

  *Random-pattern resistant* is a term used in the testing community for nodes that have a low probability of being tested using a set of random patterns.

  Controllability and Observability analysis of these nodes indicates that these nodes are controlled and/or observed under very specific conditions.

  These *rare* conditions, i.e., internal circuit states, are the target of many of the proposed Trojan testing strategies.

  The premise is that the adversary can determine these *hard-to-detect* nodes using standard manufacturing test tools and connect the Trojan to them.

- **Advantages: Functional activation** through logic testing

  Existing manufacturing test tools can be leveraged.

  The presence of the Trojan can be validated without deprocessing.
Layout-Oriented Trojan Detection Strategies

Three basic approaches:

- **Drawbacks: Functional activation** through logic testing
  
  Only applicable to small, functional Trojans (least likely) -- information leakage
  
  Trojans may not change the functional behavior of the IC
  
  The adversary may guess that the ATPG strategy may be directed at *hard-to-detect* nodes and can connect some Trojan inputs to *easy-to-detect* nodes
  
  Generating test patterns may be extremely time consuming or impossible for some *hard-to-detect* nodes (impossible to activate)
  
  As the number of inputs to the Trojan increases, the difficulty of generating Trojan activation patterns increases dramatically
  
  This is true because ATPG must generate patterns that check all combinations of *hard-to-detect* nodes (not just one node at a time as is true in manufacturing test for detects)
Layout-Oriented Trojan Detection Strategies

Three basic approaches:

- **Parametric Anomaly Detection Strategies**

  A parametric anomaly is a change in the analog characteristics of the IC, e.g., in its power consumption, delay characteristics, temperature profile, EM radiation characteristics, etc.

  As we indicated earlier with the Trojan missile example, adding a Trojan, i.e., a set of gates, to the layout of an IC changes the analog characteristics of the IC.

  These changes include:
  - Increasing the leakage characteristics of the IC
  - Increasing the dynamic power consumption of the IC
  - Increasing the delay of paths that have nodes connected to the Trojan gate inputs
  - Temperature profile changes in the IC, etc.

  Therefore, Trojan detection methods can be developed that measure an IC’s analog characteristics (*signature*) and compares it against a ’golden device’.
Layout-Oriented Trojan Detection Strategies

Three basic approaches:

- **Parametric Anomaly Detection Strategies**
  The golden device *signature* can be generated from models of the IC using simulation experiments

  The challenge in making this approach effective is accounting for the natural variations that occur between chips that are introduced by
  - Test environment variations, e.g., probe card resistance variations
  - Noise sources, e.g., environmental, instrumentation-related, on-chip sources
  - Process variations

  Detection of selectively-inserted Trojans may be possible by comparing responses among chips

- **Advantages: Parametric Anomaly Detection Strategies**
  Non-destructive and much more cost-effective than FA methods
Layout-Oriented Trojan Detection Strategies

Three basic approaches:

• **Advantages: Parametric Anomaly Detection Strategies**
  Can potentially detect ALL types of Trojans, including functional, information leakage and reliability Trojans
  
  Can be extremely sensitive to small anomalies if the appropriate *calibration* methods are applied to reduce or eliminate test environment and process variations

• **Drawbacks: Parametric Anomaly Detection Strategies**
  Automatic test equipment (ATE) instrumentation may need to be customized to make analog measurements, e.g., transient current and path delay
  
  Data collection may be time consuming and storage intensive, e.g., multiple strobes of the capture cycle to measure *actual* path delays
Layout-Oriented Trojan Detection Strategies

Three basic approaches:

• **Drawbacks: Parametric Anomaly Detection Strategies**
  
  Developing an adequate *golden device* model(s) may be difficult
  
  Models need to include environmental variations and process variations if chip data is not *calibrated*

  Number of *false alarms* may be large if process variations are not accounted for
  
  Validation through deprocessing make false alarms very costly