

Hardware Trojans (HT)

What is a hardware Trojan?

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

*negative
circuits*

- 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

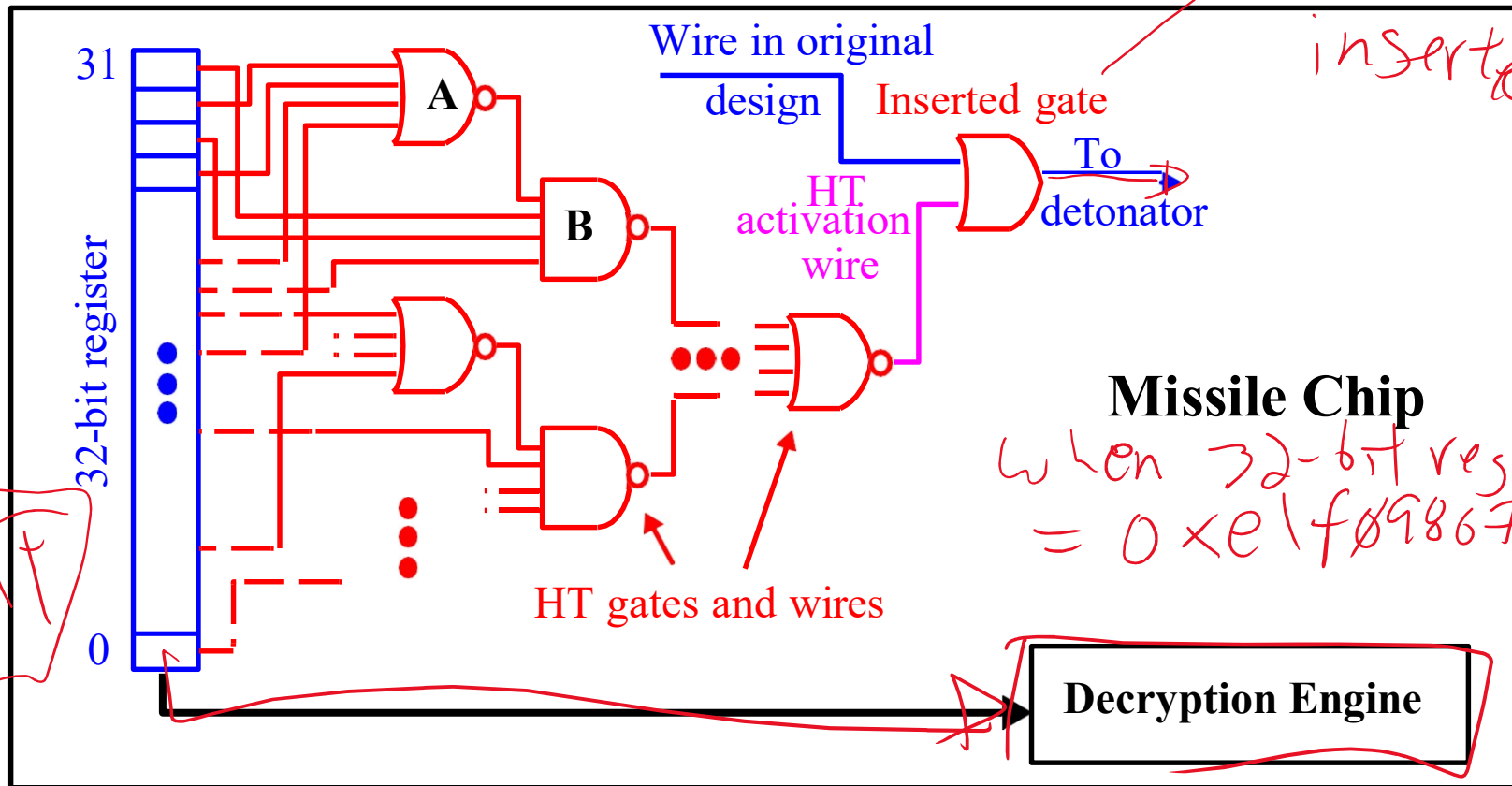
*Motivola
Freescale
AMD
Global
Foundries
2014
IBM sold its fabs to*

becoming fabless

HT Example

Missile control system

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



Missile Chip

When 32-bit res = 0xe1f09867

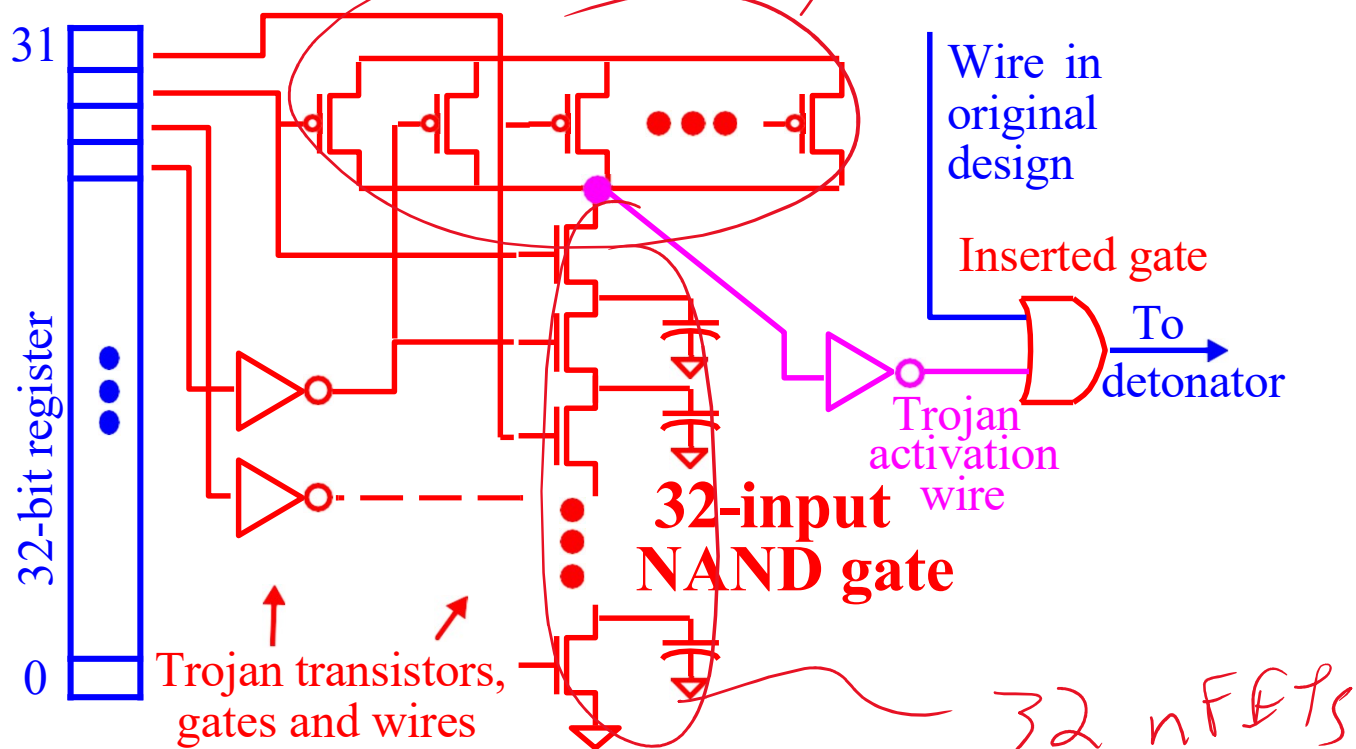
Adversary transmits "code" that causes activation - missile detonates before reaching its target

by the time the decryption engine fails, too late

32 pFETs

HT Example

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

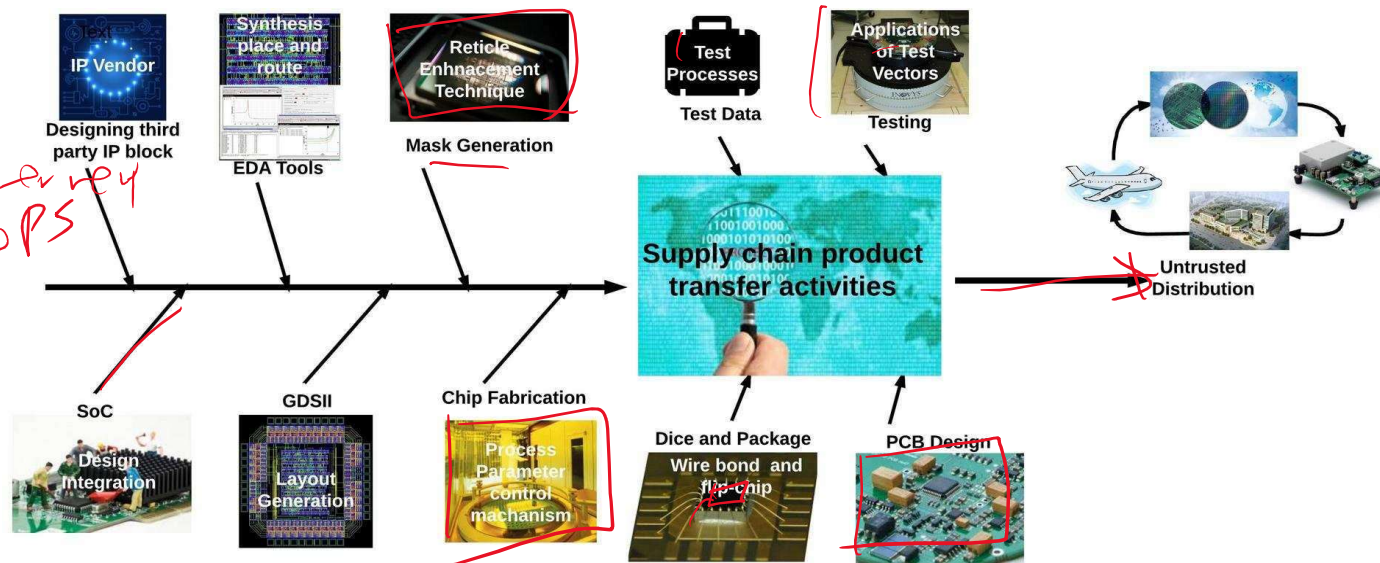


Many other implementations are possible, e.g. pass-gate versions, some better than others at minimizing power supply anomalies

Typically test a tiny subset of possible data vectors
 This HT could have another input as a countermeasure to detection
 100 Billion vectors

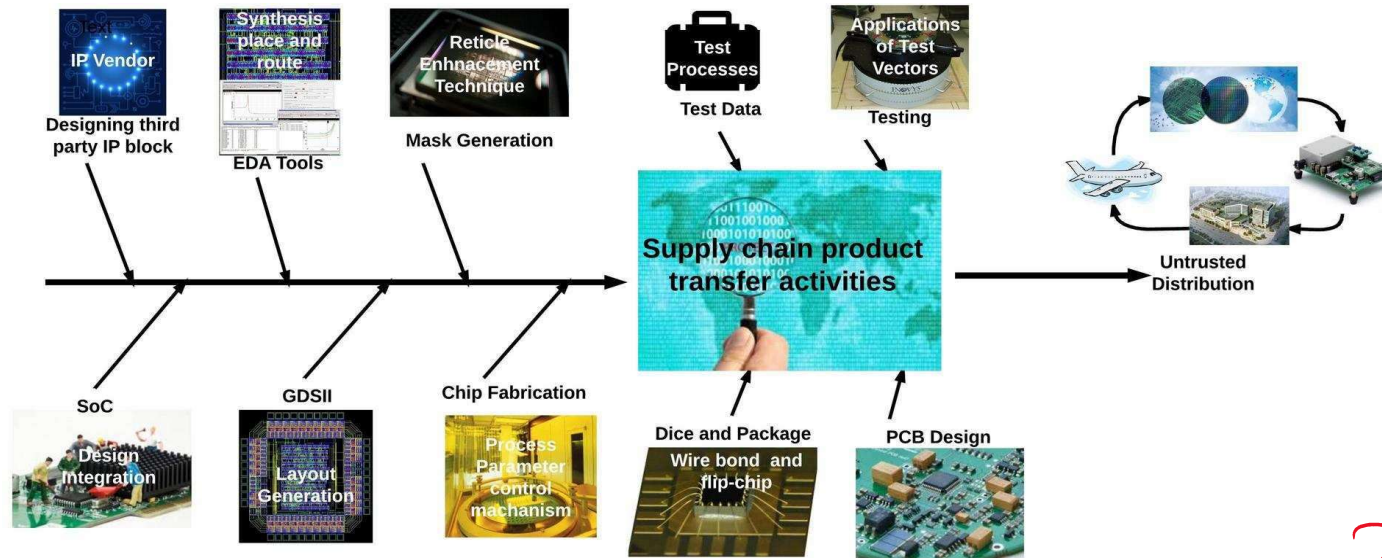
Insertion Points

The horizontal dissemination of the IC design, fabrication and test processes to many distinct companies has dramatically increased the potential for malicious activities



- Designing third party IP blocks
- Developing CAD tool scripts
- Integration activities where IP blocks and glue logic are assembled into SoCs
- Behavioral synthesis and place and route (PnR) carried out by CAD tools
- Layout mask data generation and mask preparation
- Process parameter control mechanisms used in the multi-step fabrication process
- Supply chain transactions associated with transferring wafers between facilities

Insertion Points



- Generating test vectors using automatic test pattern generation (ATPG)?
- Wafer-probing activities for measuring test structures and detecting defects
- Supply chain transactions associated with creating and transferring dice
- Processes responsible for packaging ICs
- Applying ATPG vectors to packaged ICs using ATE
- Supply chain transactions associated with transferring packaged parts
- Printed circuit board (PCB) design and fabrication
- Processes responsible for installing PCB components (populating PCBs)
- System integration and deployment activities

Insertion Points

The wide range and widely distributed nature of these activities presents an overwhelming opportunity for subversion

Moreover, the diversity among the tasks will require a very sophisticated and complex system to manage the entire set of trust vulnerabilities from start to finish

The research community is tackling these trust challenges one-at-a-time, focusing on those that are the most attractive insertion points for adversaries

- **Subversion of IP blocks** is a serious concern given the ease in which malicious functionalities can be covertly inserted

Moreover, the absence of alternate representations and models for comparison compounds the challenge

- **Layout modifications and IC fabrication** insertion points represent another important focus area

Challenges here include the huge complexity associated with analyzing fabricated ICs and the wide range of opportunities available to the adversary

GSM
cellular
text
IP block

HT 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 HT

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 only applicable to small circuits, i.e., components of the design

There has been a lot of recent work in this area that offers alternative solutions, e.g., circuit obfuscation techniques, which we will cover later in this course

not cover

"Known good"

$$\frac{32}{5} \approx 6$$

state space explosion: $2^n \cdot 2^m = 2^{n+m}$

HT 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

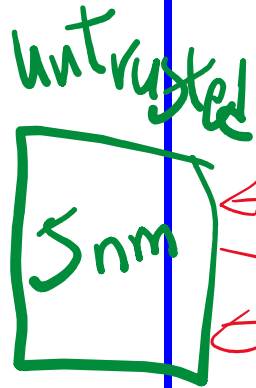
This technique can also be used for GDS-based HT

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

Hard-IP 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



in chip
Xilinx
Altera

HT 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 HT gates

In case you were thinking about adding some type of verifiable white space filler to prevent the first case from occurring

Modifications designed to reduce reliability can be implemented by thinning wires to accelerate EM ^{electron} effects, by manipulating doping concentrations, etc.

Risk 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?

adv. can control more

Let's consider another HT parameter, the size of the HT

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

For HT designed to change functionality, small HT are risky - why?

less likely to be used or work

HT Scenarios: **Hard-IP Trojans**

Therefore, ^{plusyellik} argue that HT designed to change functionality, e.g., disable, enable remote control, are likely to be larger, 10's to 100's of gates to prevent discovery

Unfortunately, the same is not true for information leakage HT

Since they do not, at least in an obvious way, change functionality, they can be very small and remain secure against detection

Information leakage HT can 'leak' information in ways that are not easily detected

- By broadcasting data as EM ^{lectro} radiation using a portion of the power grid
- By inserting data into a ^{magnetic} 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 and in many cases, requiring *non-traditional* testing methods

HT Scenarios: Hard-IP Trojans

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

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

0.01%

This makes it more difficult to develop methods to detect them

Whether the HT is *selectively inserted* or inserted into every copy depends on the application

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

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

Important Considerations

Note that significant differences exist in the HT countermeasures and detection strategies that are applicable

This is true even when only considering only the Soft-IP and Hard-IP insertion points discussed above

For example, golden models are not available at the soft-IP block insertion point, but architectural changes that **obfuscate** the design are available as countermeasures

In contrast, the Hard-IP insertion point allows layout design data to be used to validate the functional and analog behaviors of the IC

But obfuscation is limited to ‘dummy via’ insertion and other nano-level manipulations of the design

Also, *side channel information* is not available or is not accurate enough to be useful for soft-IP blocks but is very powerful for layout-level HT detection

Summary of Challenges of Detecting HT

As discussed, HT detection faces several challenges

- **Hard-IP** Trojans Only: The adversary can choose to **'selectively'** insert the HT into only a subset of the manufactured ICs, making it necessary to **verify all ICs**
- HT designed to **leak information** may not cause a change in the **functional behavior** and therefore, will require **specialized testing methods**

Additional challenges associated with HT detection:

- Task of identifying **an HT** is akin to **finding a needle in a haystack**, i.e., trusted authority must find unknown malicious function in a 'sea' of functions and gates
- HT and **'bugs'** share many of the same characteristics, and it is widely accepted that finding all the bugs is generally infeasible
- Any attempt by the trusted authority to increase the 'ease' of **HT detection** may be visible to **the adversary**, i.e., the adversary can avoid the countermeasures
- The appropriate detection strategy will vary greatly depending on the assumptions made regarding the **"insertion point"**, e.g., **Soft-IP vs Hard-IP** Trojans

Summary of Challenges of Detecting HT

The only advantage afforded to the trusted authority is that his/her detection strategy can be parallelized because the HT needs to be detected only once

Manufacturing tests can be partitioned among multiple ATE

Note this assumes that selective HT insertion is not possible

Is always true for Soft-IP Trojans and is a reasonable assumption for Hard-IP Trojans b/c of mask cost issues

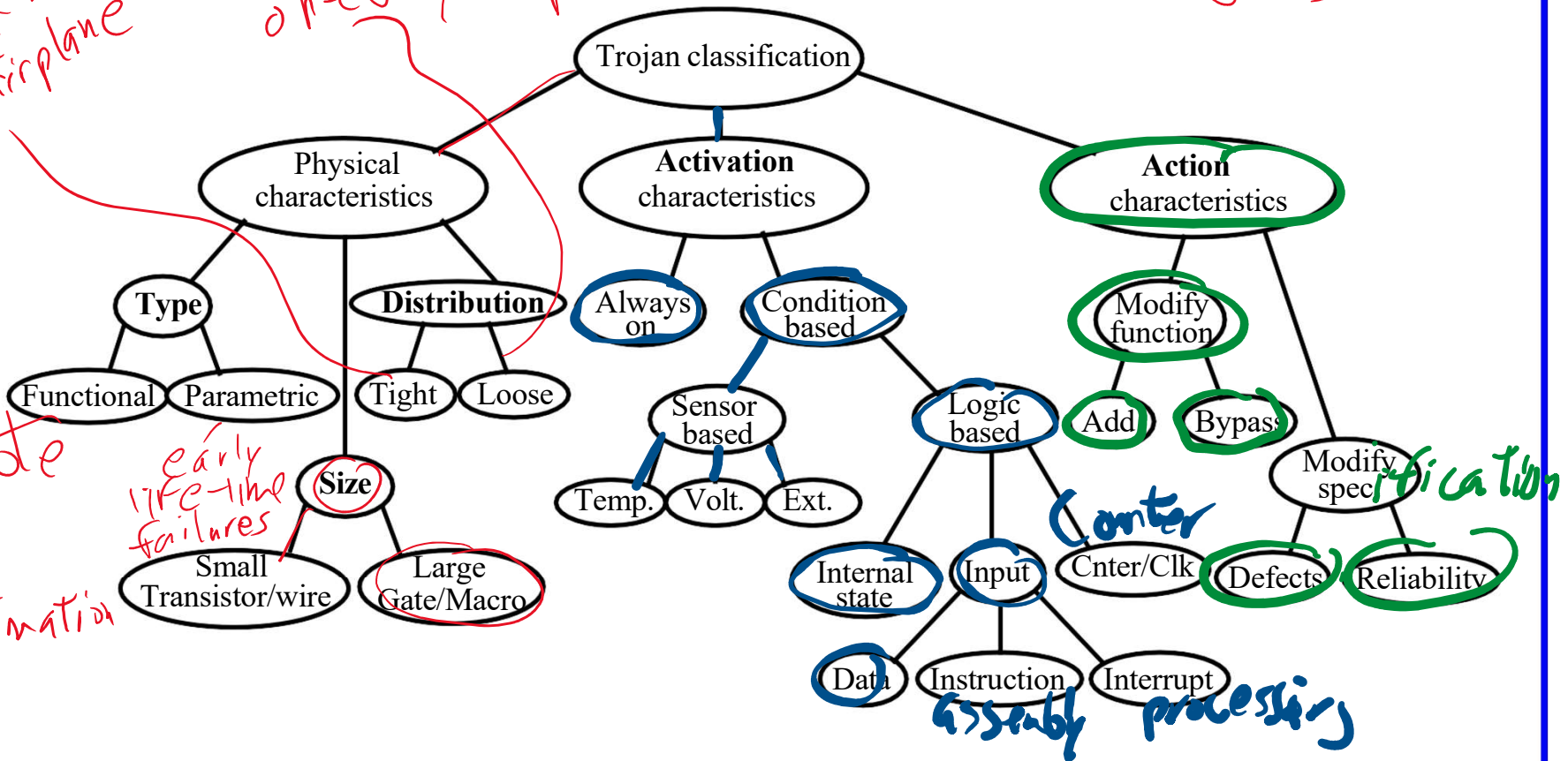
100 X

Unfortunately, even high levels of parallelism 'run-out-of-gas' when the full extent of the search space, both combinational and sequential, is considered

Hard-IP Trojan Taxonomy

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

on every chip, e.s.s + HDL IP Trojans



adv makes an extra angle for airplane

disable or leak information

early life-time failures

"Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans", R. M. Rad, W. Xiaoxiao, M. Tehranipour, J. Plusquellic, International Conference on Computer-Aided Design (ICCAD), Nov. 2008, pp. 632 - 639

Hard-IP 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 Alteration (LIVA/CIVA)

- **Advantages**

Potentially highly sensitive to the presence of HT

- **Drawbacks**

Destructive, may miss selectively-inserted HT, cost, effectiveness in nano



A few examples

FM# tools have a high probability to detect HT with very few false positives

time,

Hard-IP 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 HT

Given the connectivity characteristics of the HT to the original circuit are UNknown, ATPG must be based on a heuristic

A common assumption is that the adversary is likely to connect the HT to wires in the original design that are the most difficult to control and observe

The rationale for this is simple: the adversary wants to make accidental discovery of the HT, i.e., through manufacturing test, highly unlikely

Therefore, many of the proposed *functional activation* strategies are based on deriving tests for nodes that are random-pattern resistant

Hard-IP 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

stuck-at fault

Controllability and observability analysis indicates these nodes are controlled and/or observed under very specific conditions, i.e., internal circuit states

These *rare* conditions are the target of many of the proposed HT testing strategies

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

- **Advantages**

Existing manufacturing test tools can be leveraged

The presence of the HT can be validated without deprocessing

chip thus not destroyed

Hard-IP Trojan Detection Strategies

Three basic approaches:

- **Drawbacks**

Only applicable to small, functional HT, which are the least likely

Note that information leakage HT may **not** change the functional behavior of the IC so logic testing will be ineffective

The adversary may guess that the ATPG strategy may be directed at *hard-to-detect nodes* and can connect some HT 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 HT increases, the difficulty of generating HT activation patterns increases dramatically

This is true because ATPG must generate patterns that check all combinations of *hard-to-detect nodes*

change outputs

Hard-IP 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, etc.

As we indicated earlier with the HT missile example, adding a HT to the layout of an IC changes the analog characteristics of the IC

This is true because of the **observer effect** (Heisenberg principle) that observing a system (the HT) changes the behavior of the system

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 HT gate inputs
- Changing the temperature profile of the IC

Therefore, HT detection methods can be developed that measure an IC's analog characteristics (signature) and then compares them to a golden chip or model

infant failures

Tejas
2004

Hard-IP Trojan Detection Strategies

Three basic approaches:

- **Parametric Anomaly Detection Strategies**

The golden *signature* can be generated from models of the IC using simulation experiments or from chips fabricated in a trusted facility

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, both chip-to-chip and within-die

Detection of selectively-inserted HT may be possible by comparing responses among the tested chips

- **Advantages**

Non-destructive and much more cost-effective than FA methods

mother
nature

Hard-IP Trojan Detection Strategies

Three basic approaches:

• Advantages

Can potentially detect ALL types of HT, including functional, information leakage and reliability HT

Can be extremely sensitive to small anomalies if the appropriate calibration methods are applied to reduce/eliminate environmental and process variations

• Drawbacks

Automatic test equipment (ATE) instrumentation may need to be customized to make high precision measurements, e.g., transient current and path delay

Data collection may be time consuming and data storage and processing large, e.g., multiple strobes of the capture cycle to measure *actual* path delays

FWA
Test

1 GHz $\lambda_s = 10^{-9}$ s

$p_s = 10^{-12}$ s

Hard-IP Trojan Detection Strategies

Three basic approaches:

- **Drawbacks: Parametric Anomaly Detection Strategies**

Developing a golden model based on simulations may be difficult because foundry models must match the hardware, and account for noise sources

Developing a golden model based on trusted chips is challenging because the nature of process variations that exist in the chips depends on the foundry

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

False Positives
False Negatives