Hardware & Embedded Security

Prof Daniele Rossi


Via G. Caruso 16, room B-1-03


daniele.rossi1@unipi.it


050 221 7611

Hardware Trojans

Taxonomy and Examples

Lecture 4 - DR

Brief Outline

- **Hardware Trojans**
  - Implementation
  - Taxonomy
  - Examples

3

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, …

4

HT is a piece of HW in your IC that shouldn't be there. It's been put there with some malicious intentions. HWT can:

1) Change functionality of system } RELATED. CAUSE SOME DISRUPTION TO IMPACT PROVIDER'S REPUTATION (DONE BY COMPETITORS) ①
2) Reduce reliability
3) leak sensitive information

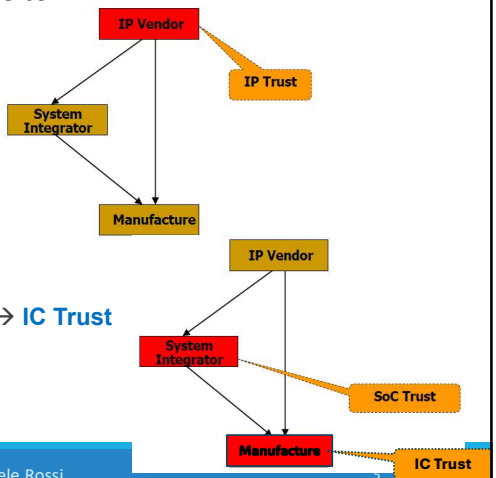
① Best case: attack to user satisfaction. But this can attack reliability of a system too. Imagine attacking a component of a security critical application like autonomous vehicles.

HWT reasonably target military, transportation systems, or any kind of security-critical applications, financial applications etc.

But how can those attacks be carried out? [SLIDE 5]
Where does this weak point come from? Depends on the design manufacturing process.
1 countermeasure could be control the whole manufacturing process. Today, with fabless companies, this is just not possible.
This is related to IC trust, SOC trust etc:
If you have manufacturer not under your control, you have ways to determine if you have what you wanted. Ideally you could compare what you get with what you expect. Functional analysis is difficult though. Side channel analysis, including power or path delay analysis. But in those cases, you need a golden model.
What about IP trust? Computing when you have just IP is no longer possible. For IP is not a matter of manufacturing; if a hw trojan is in the IP, it is going to be present in every component you produce.

IC/IP Trust Problem

- Chip design and fabrication has become very vulnerable to malicious activities and alterations with globalization
- IP Vendor and System Integrator:
  - IP vendor may place a Trojan in the IP → **IP Trust**
- Designer and Foundry:
  - Foundry or integrator may place a Trojan in the layout design → **IC Trust** or **SoC Trust**



5

Issues with Third-Party IP Design

- Hardware Trojans can be inserted into **third-party IPs** by IP vendors during IP design to steal security information/data from other IPs in the system
- Detection of such Trojans is extremely difficult since there is **no known golden model** for third-party IPs as IP vendors usually provide specification and source code, both of which may contain Trojans
- The conventional side-channel techniques for IC trust are not applicable to IP trust: When a Trojan exists in an IP core, **all the fabricated ICs will contain Trojans**
- A Trojan can be very well hidden during the normal functional operation of the third-party IP supplied as register transfer level (RTL) code
- A large industrial-strength IP core can include thousands of lines of code → Identifying the few lines of **RTL code in a soft IP** core that represent a **Trojan** is an extremely **challenging** task

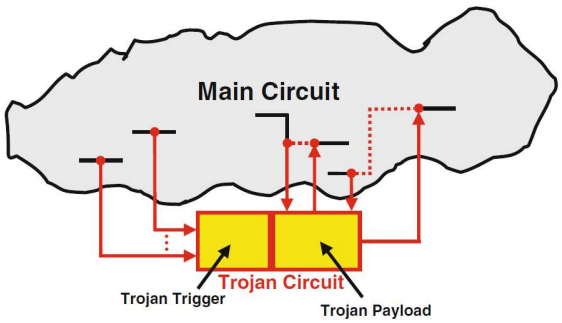
6

But IP Trojans are less likely to be present because you work with trusted partners. With the other two, you might want to speed things up and find shortcuts to save time. So here reliability is a big problem.

HT Implementation

7

Functional Trojan implementation



- A hardware Trojan (HT) consists of: ① Two main blocks
  - ***Trojan trigger*** and
  - ***Trojan payload***
- A functional Trojan takes inputs from some internal nets of the main circuit to the **Trojan payload** and restitches some other nets of the main circuit through **Trojan payload** to **modify design functionality**
- The **Trojan trigger** determines the **activation condition(s)** under which the Trojan payload can propagate erroneous values into the main circuit

8 You have nodes that are used as input of HWT. ①
- When receiving a specific pattern the trigger triggers the T payload that changes the functionality of system. It can alter behavior or introduce reliability issues.

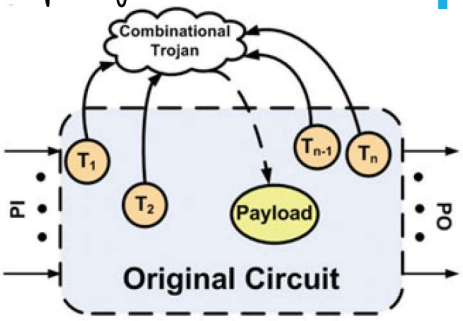
As inputs of trigger you usually don't have primary inputs but just internal nodes. Why? You can test system by applying sets of input. It would be easier to test and detect presence of bugs connected to inputs because you have direct control over them.

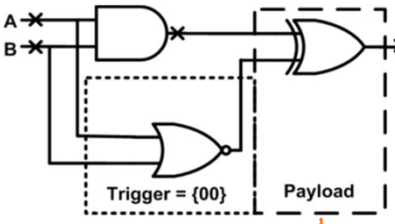
My honest reaction to hardware bugs

HT Combinational Model and Example

Output only depends on the input, not on any internal state

- The **combinational Trojans** do not contain any sequential elements and depend only on the **simultaneous occurrence of a set of rare node conditions** (e.g., on nodes T1 through node Tn) to trigger a malfunction



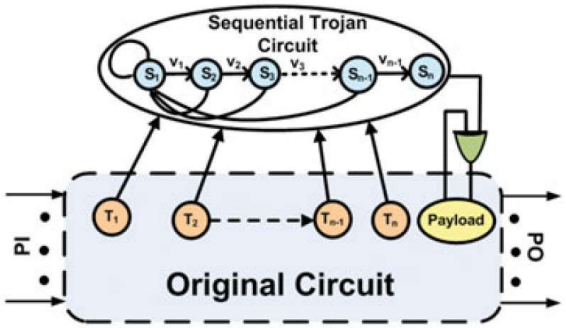


Why "rare"? Because it would be too easy to test otherwise.

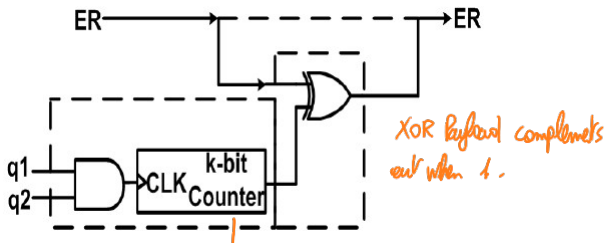
A	B	T	AND	OUT
0	0	1	0	1
0	1	0	0	0
1	0	0	0	0
1	1	0	1	1

Output is not anymore A·B but (A·B) ⊕ Trigger

HT Sequential Model and Example



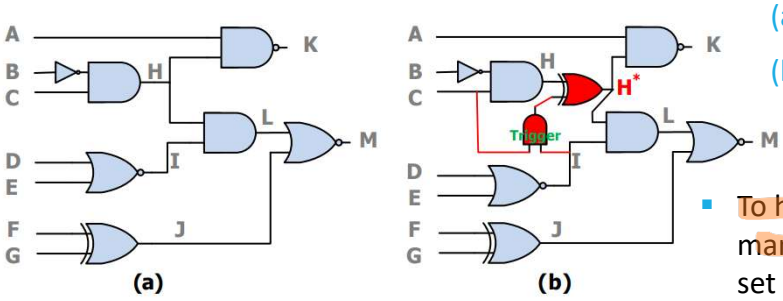
- The **sequential Trojans** undergo a **sequence of state transitions** (S1 through Sn) before triggering a malfunction.



XOR keyboard complements out when 1.

Counter on K bits. Upon AND going from 0 to 1 for 2^K times, imagine our is overflow of counter, you change ER value.

Example: Combinational HT



(a) A simple combinational circuit

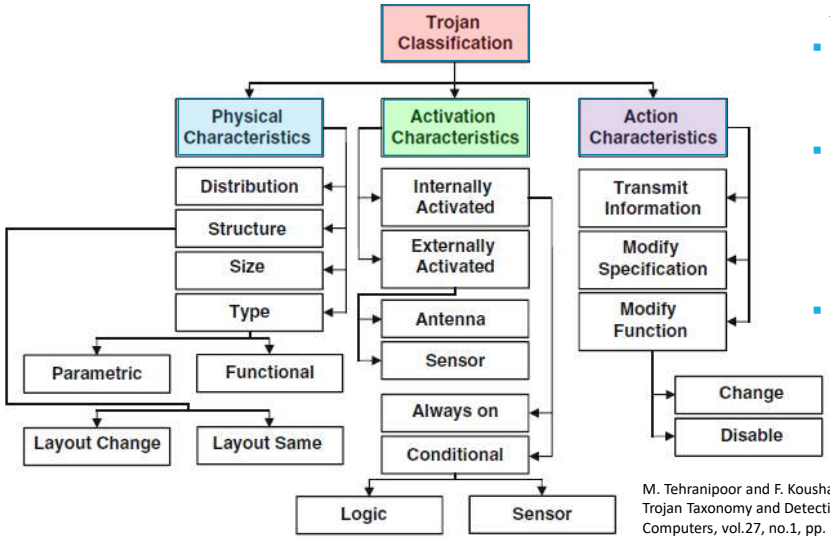
(b) Trojan with trigger and payload added to the original circuit by an external adversary

- To hide the HT from detection during manufacturing test, the adversary has set the **Trojan to be triggered only when nodes C and I are both at logic 1**
- The **trigger AND** gate excites the **payload XOR** gate that causes the node H to be modified to H* leading to an incorrect circuit operation

11 NECESSARY CONDITION: C=1, D=0,E=0.8V not necessarily that induces a fictional modification. The change needs to propagate and that depends on circuit conditions. To observe effects you have to look at primary output of your circuit. You should be able to CONTROL THE NODE and propagate it to the OUTPUT.
So A should be 1 to have K depending on H*.

HT Taxonomy

Hardware Trojan Taxonomy

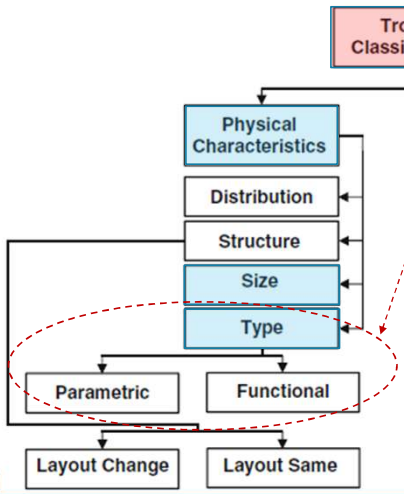


- Currently, the industry **lacks metrics** to evaluate the effectiveness of methods in detecting Trojans
- Such metrics and a comprehensive **taxonomy** could help analyse Trojan detection techniques and let researchers examine their methods against different Trojan types
- Because malicious alterations to a chip's structure and function can take many forms, Trojan taxonomy is decomposed into **three main categories**, according to their **physical**, **activation**, and **action** characteristics

M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," IEEE Design & Test of Computers, vol.27, no.1, pp. 10–25, 2010

14

HT Taxonomy: Physical Characteristics (I)



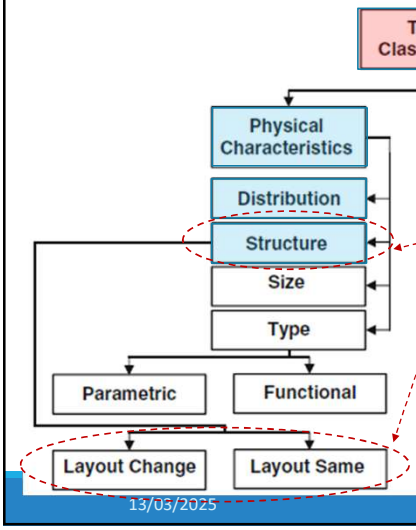
- The physical characteristics category describes the various hardware manifestations of Trojans.
- **Type category** partitions HTs into functional and parametric classes:
  - **Functional class**: HTs that are physically realized through the addition or deletion of transistors or gates
  - **Parametric class**: HTs that are realized through modifications of existing wires and logic
- **Size category**: **number of components** in the chip that have been added, deleted, or compromised

15

First classification:

- Physical characteristics
  - Size: the bigger the trojan is, the easiest it is to detect it (not even for functional tests, but for side channel ones too.
  - Type: functional try to alter the functionality. Parametric: purpose is to change some kind of electric characteristics. Maybe modify propagation delay of a system. Thus by changing (or not) the layout of the system. Changing layout makes detection easier. EX: modification of non functional cells that do not affect functionality and layout.
  - Distribution: refers to location in the physical layout
  - Structure: see below.

HT Taxonomy: Physical Characteristics (II)

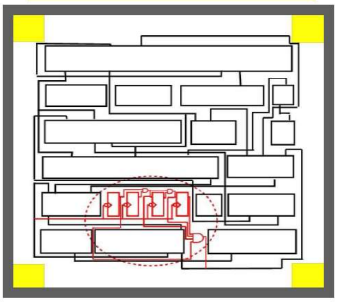


- The physical characteristics category describes the various hardware manifestations of Trojans.
- **Distribution category**: **location** of the Trojan in the chip's physical layout
- **Structure category**: the case when an adversary is forced to **regenerate the layout** to insert a Trojan → the chip's physical form factor could change → such changes could result in different placement for some or all design components
  - Any malicious changes in physical layout that could change the chip's delay and power characteristics would **facilitate Trojan detection**

16

Examples for Layout Level Trojans: Type

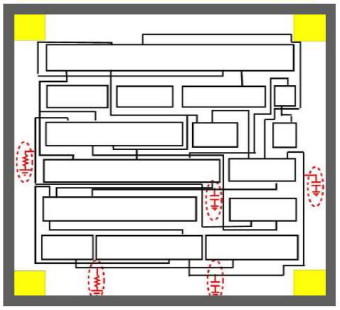
Functional



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

You might have empty space available that you can use.

Parametric



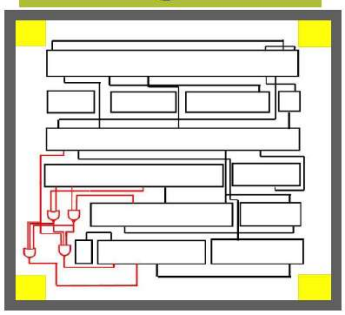
- Modifications of existing components
  - Wire: e.g. thinning of wires
  - Logic: Weakening of a transistor, modification to physical geometry of a gate
  - Modification to power distribution network
- Sabotage reliability or increase the likelihood of a functional or performance failure

We can modify power consumption, propagation delay. Make components less conductive

17

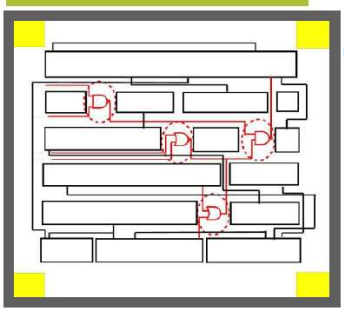
Examples for Layout Level Trojans: Distribution

Tight



- Trojan components are topologically close in the layout

Loose



- Trojan components are dispersed across the layout of a chip

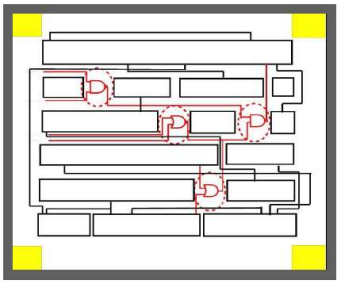
Effects of those pieces is less important so it might be harder to detect them, but whenever you test, you are likely to test an area that has a Trojan.

- Distribution of Trojans depends on the availability of dead spaces on the layout

19

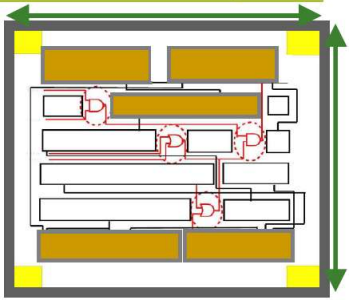
Examples for Layout Level Trojans: Structure

No-change



- 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

Modified Layout

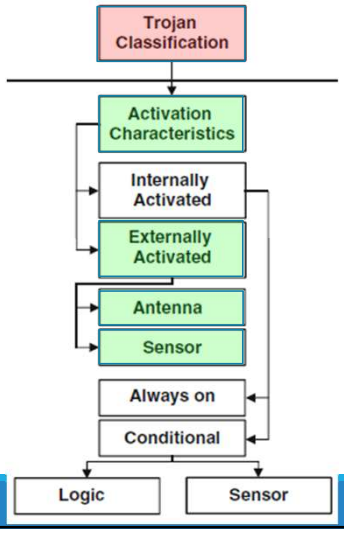


- A change in physical layout can change the delay and power characteristics of chip
  - It is easier to detect the Trojan

Change in circuit Form Factor

20

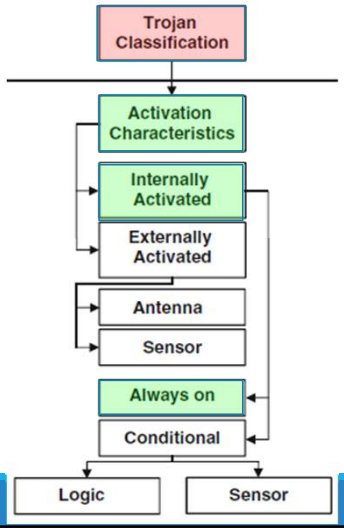
HT Taxonomy: Activation Characteristics (I)



- Activation characteristics refer to the criteria that cause a Trojan to become active and carry out its disruptive function
- Two categories: **Externally Activated** and **Internally Activated**
- **Externally Activated**
  - E.g., HTs activated by an **antenna** or a **sensor** that can interact with the outside world

21

HT Taxonomy: Activation Characteristics (II)



- Activation characteristics refer to the criteria that cause a Trojan to become active and carry out its disruptive function
- Two categories: **Externally Activated** and **Internally Activated**
- **Internally Activated**:
  - **Always on**: the HT is **always active** and can disrupt the chip's function at any time.
    - This subclass covers Trojans that are implemented by **modifying the chip's geometries** such that certain nodes or paths have a higher susceptibility to failure. The adversary can insert the Trojans at nodes or paths that are **rarely exercised**

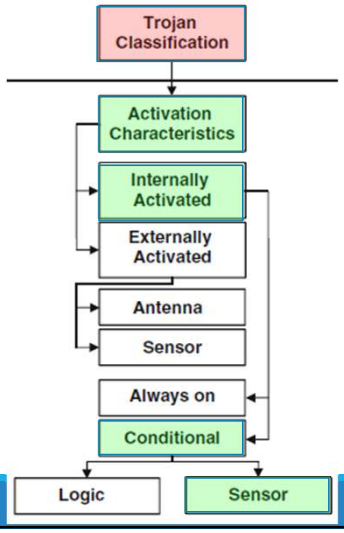
22

What about activation characteristics?

- External activation: by means of antennas or sensors triggered by radio signals.
- Internal activation:
  - Always on (among IA): they are always active. This doesn't mean they are always modifying OP of the system. They need to be triggered first, but they can be triggered at any time. Of course they should activate upon rare conditions.
  - Conditional: inactive until a certain condition is met, and activation can be decided by a sensor output (voltage, temperature). This can be a triggering condition. You can also have any a logical condition.

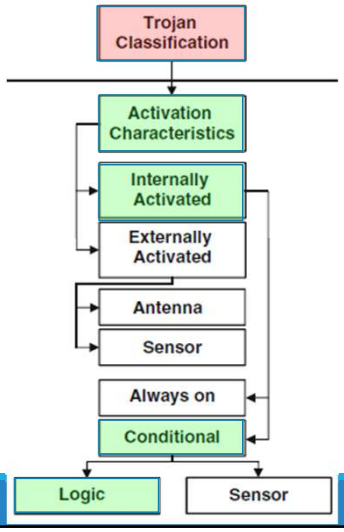
  NOTE: you can have dormant HW that gets activated, and after that activation you can have a trigger. But activation condition can be similar to triggering one too.

HT Taxonomy: Activation Characteristics (III)



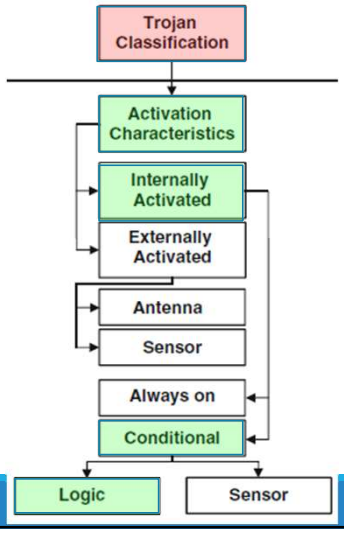
- Activation characteristics refer to the criteria that cause a Trojan to become active and carry out its disruptive function
- Two categories: **Externally Activated** and **Internally Activated**
- **Internally Activated**:
  - **Conditional**: HTs are inactive until a specific condition is met
    - **Sensor**: The activation condition could be based on the **output of a sensor** that monitors temperature, voltage, or any type of external environmental condition (such as electromagnetic interference, humidity, altitude, or temperature

HT Taxonomy: Activation Characteristics (IV)

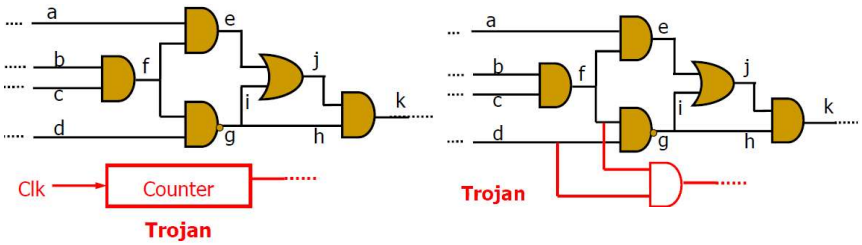


- Activation characteristics refer to the criteria that cause a Trojan to become active and carry out its disruptive function
- Two categories: **Externally Activated** and **Internally Activated**
- **Internally Activated**:
  - **Conditional**: HTs are inactive until a specific condition is met
    - **Logic**: condition could be based on an **internal logic state**, a particular **input pattern**, or an internal **counter value**. The Trojan in these cases is implemented by adding logic gates and/or flipflops to the chip, and hence is represented as a **combinational or sequential circuit**

HT Taxonomy: Activation Characteristics (IV)

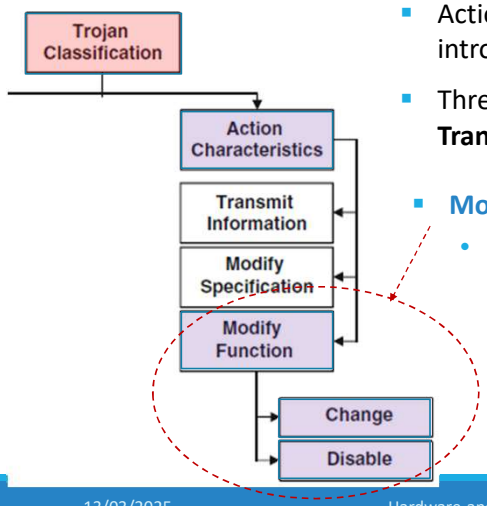


- Activation characteristics refer to the criteria that cause a Trojan to become active and carry out its disruptive function
- Two categories: **Externally Activated** and **Internally Activated**
- **Internally Activated**:
  - **Conditional**: HTs are inactive until a specific condition is met



25

HT Taxonomy: Action Characteristics (I)



- Action characteristics identify the types of disruptive behaviour introduced by the Trojan
- Three categories: M**odify Function, Modify Specification,** and **Transmit Information**
- **Modify Function**:
  - It refers to Trojans that change the chip's function by adding logic (**Change**) or by removing or bypassing existing logic (**Disable**)

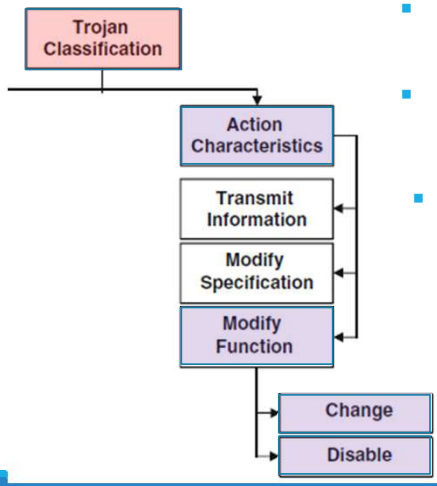
26

What about Active Characteristics?

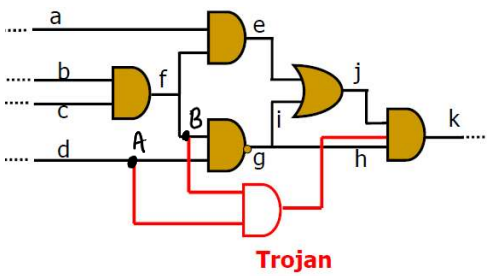
- Function modification: can be changing or disabling functions. CALL 27 for example.
- Modify specification: related to parametric Trojans. We might modify parametric properties such as delay, by for instance modifying wire and transistor geometry. Note that if you change functions in slide 27, you are also applying some sort of specification change: we modify delays by applying extra capacitive loads. (A, B) But you can target specification only, by adding extra capacitors, that modify delay and power consumption. (CALL 29)
- Transmit information: transmit secret keys for cryptography or whatever else to take advantage of some things. In medical devices you can steal health data from patients.

JMP 31

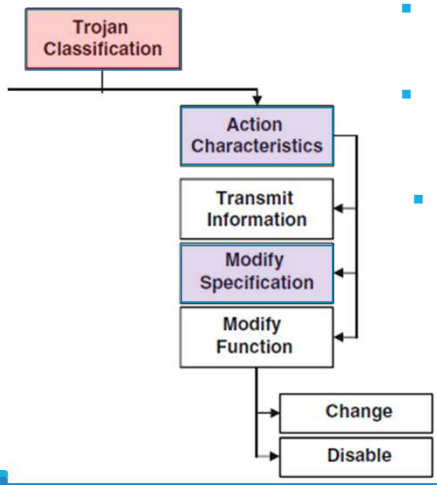
HT Taxonomy: Action Characteristics (I)



- Action characteristics identify the types of disruptive behaviour introduced by the Trojan.
- Three categories: M**odify Function, Modify Specification,** and **Transmit Information**
- **Modify Function**:
  - **Change**

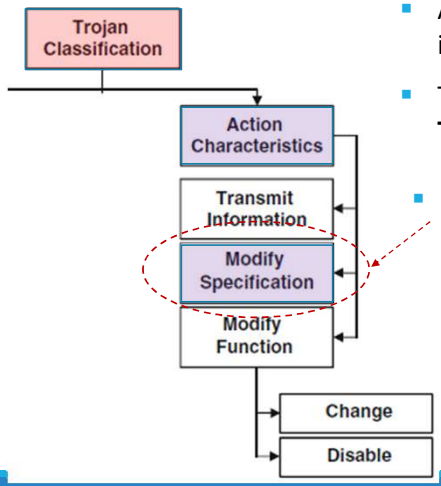


HT Taxonomy: Action Characteristics (II)

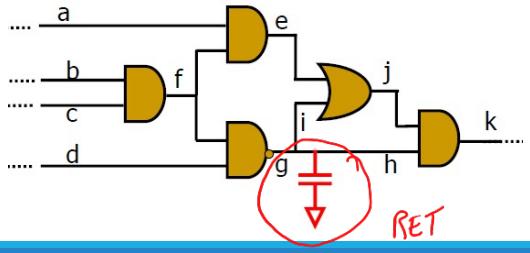


- Action characteristics identify the types of disruptive behaviour introduced by the Trojan
- Three categories: M**odify Function, Modify Specification,** and **Transmit Information**
- **Modify Specification**:
  - It refers to Trojans that focus their attack on **changing** the chip's **parametric properties**, such as **delay** when an adversary modifies existing wire and transistor geometries

HT Taxonomy: Action Characteristics (II)

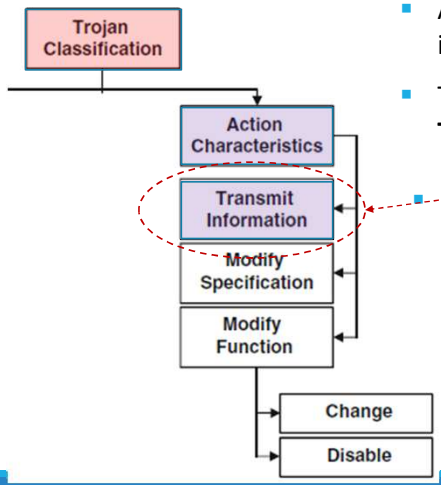


- Action characteristics identify the types of disruptive behaviour introduced by the Trojan
- Three categories: M**odify Function, Modify Specification,** and **Transmit Information**
- **Modify Specification**: delay for paths involving g-h increases



29

HT Taxonomy: Action Characteristics (III)



- Action characteristics identify the types of disruptive behaviour introduced by the Trojan.
- Three categories: M**odify Function, Modify Specification,** and **Transmit Information**
- **Transmit Information**:
  - It includes Trojans that transmit **key information** to an adversary

30

Example 1: MOLES Hardware Trojan (I)

- **MOLES**: **Malicious Off-chip Leakage Enabled by Side-channels** [r1]
- A generic MOLES can be implemented by different side-channels, such as power consumption, electromagnetic radiation and path delay
- In this example, a MOLES circuit is specifically designed to **consume data-dependent power** as a power side-channel to **leak multi-bit secret keys**
- A critical feature of MOLES is the **signal-to-noise ratio (SNR)**, defined as the **power level of side-channel leakage** to **that of the host IC**
- An effective MOLES requires a **low SNR to evade evaluators' detection**, but a **high enough SNR for the attacker to extract the secret key bits**

[r1]. Lin, Burleson, Paar, "MOLES: Malicious Off-Chip Leakage Enabled by Side-Channels", IEEE/ACM International Conference on Computer-Aided Design Jan. 2009 (ICCAD 2009)

32 Power consumed by IC depends on data that is being processed

Example 1: MOLES Hardware Trojan (I)

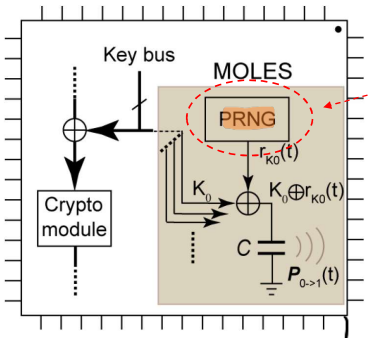
- **MOLES**: **Malicious Off-chip Leakage Enabled by Side-channels** [r1]
- A generic wer consump
- In this ex pendent power as
- A critical ower level of side-char
- An effect high enough SNR for t ime.

Side-Channel Attack - exploit the implementations of cryptographic algorithms or software: When performing a side-channel attack, some **observable behaviour** of the (cryptographic) routine implementation is used to obtain additional information that allows the attacker to decode some cipher text, calculate the cryptographic keys or obtain details of the executed instructions and data within the system

[r1]. Lin, Burleson, Paar, "MOLES: Malicious Off-Chip Leakage Enabled by Side-Channels", IEEE/ACM International Conference on Computer-Aided Design Jan. 2009 (ICCAD 2009)

33

Example 1: MOLES Hardware Trojan (II)

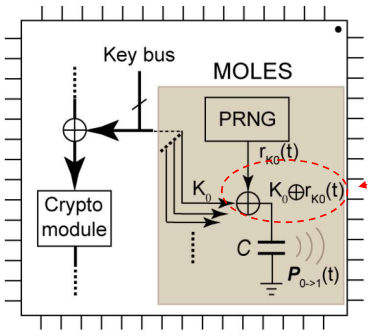


Lin, Burleson, Paar, ICCAD 2009

- Each key bit is modulated with a long pseudo-random number (PN) sequence by an XOR operation. As shown in the figure, a binary pseudo-random number generator (PRNG) can generate a PN sequence $r_{K0}(t)$

Objective: leak secret key to perform operation.
Crypto module gets key as input, and when we have our moles.
PRNG generates a sequence that looks like it is random, but it is not
XOR is of PR Sequence and key bits, connected to a capacitor. You measure electromagnetic signal generated by capacitor, so you can monitor when value on capacitor changes from 0 to 1 (change). Since you know the PRS, you can try to understand bits of the key.
PRNG is there to trigger transitions. So PRNG is good for that because it forces 0→1 transition.

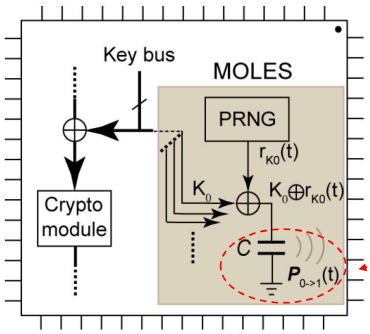
Example 1: MOLES Hardware Trojan (II)



Lin, Burleson, Paar, ICCAD 2009

- Each key bit is modulated with a long pseudo-random number (PN) sequence by an XOR operation. As shown in the figure, a binary **pseudo-random number generator** (PRNG) can generate a PN sequence $r_{K0}(t)$
- The multi-bit key bus is covertly hardwired to the XOR gates by the attacker (only the key bit K0 is shown in the figure)

Example 1: MOLES Hardware Trojan (II)

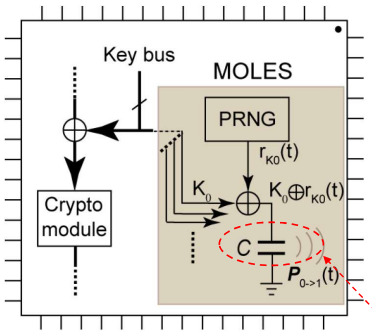


Lin, Burleson, Paar, ICCAD 2009

- Each key bit is modulated with a long pseudo-random number (PN) sequence by an XOR operation. As shown in the figure, a binary pseudo-random number generator (PRNG) can generate a PN sequence $r_{K0}(t)$
- The multi-bit key bus is covertly hardwired to the XOR gates by the attacker (only the key bit K0 is shown in the figure)
- The output node of each XOR gate, with no connection to any I/O pin, is connected to a capacitive load that leaks a small amount of power $P_{0\rightarrow1}(t)$ when a 0→1 logic transition occurs.

36

Example 1: MOLES Hardware Trojan (II)



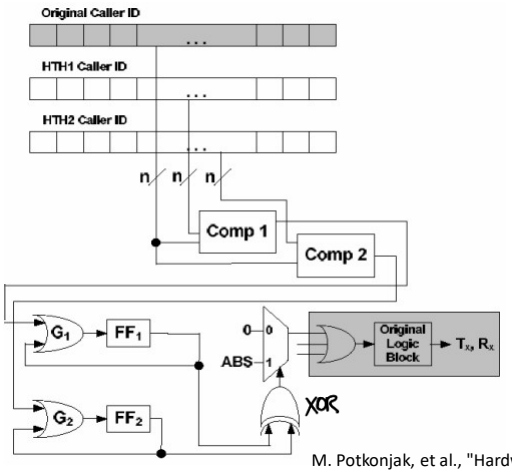
Lin, Burleson, Paar, ICCAD 2009

- Each key bit is modulated with a long pseudo-random number (PN) sequence by an XOR operation. As shown in the figure, a binary pseudo-random number generator (PRNG) can generate a PN sequence $r_{K0}(t)$
- The multi-bit key bus is covertly hardwired to the XOR gates by the attacker (only the key bit K0 is shown in the figure)
- The output node of each XOR gate, with no connection to any I/O pin, is connected to a capacitive load that leaks a small amount of power $P_{0\rightarrow1}(t)$ when a 0→1 logic transition occurs.
- The size of the load capacitor is an adjustable design parameter for MOLES, which determines the amount of side-channel information leakage and thus the SNR But too big is detectable

37 What kind of side channel analysis you need here?

We have this ghost circuitry in order to activate it of the caller.
We want to see whether original caller corresponds to one of the two HTH caller IDs.
We have a comparator. This is the trigger part. When first caller calls, FF1 stores 1.
Then XOR switches to 1. FF1 will always be 1. When second caller calls, FF2 is 1 and XOR goes to 0. You can deactivate Trojans.

Example 2: Cell Phone HT

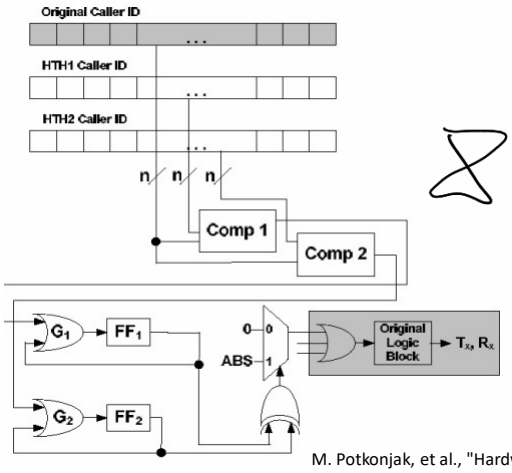


- **Ghost circuitry** can be activated in a cell phone when specific inputs or data are detected at specific memory locations
- The **unshaded portion** of the circuit represents the **Hardware Trojan Horse** (HTH) circuitry when it is activated by a HTH caller ID number.

  Activation: **Original Caller ID = HTHi Caller ID**, I = 1 or 2
- Upon activation, the **attacker bitstream** (**ABS**) is activated and the initial cell phone design is corrupted → HTHs will either cause the cell phones to malfunction or cause confidential information to be leaked.

M. Potkonjak, et al., "Hardware Trojan horse detection using gate-level characterization", DAC 2009, 688-693

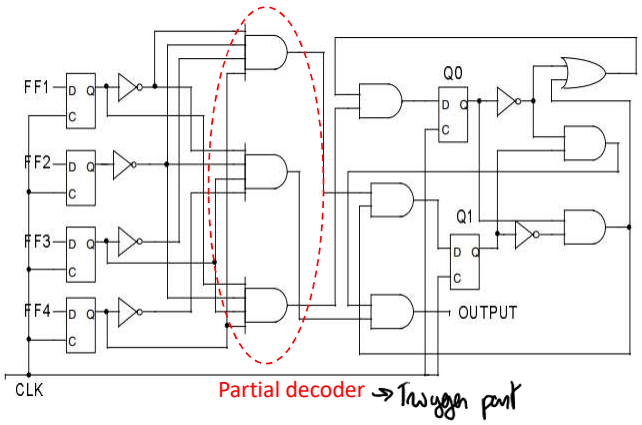
Example 2: Cell Phone (Combinational) HT



- HT may be **difficult to identify by traditional timing/power analysis techniques**.
- To avoid timing analysis-based detection, an attacker only needs to ensure that **no path delays** between the inputs to flip-flops (FFs) or between the outputs to FFs **are increased**.
- Also, the **switching power** can remain stable **until the trigger** of the attacker's caller ID **activates the HTH**

M. Potkonjak, et al., "Hardware Trojan horse detection using gate-level characterization", DAC 2009, 688-693

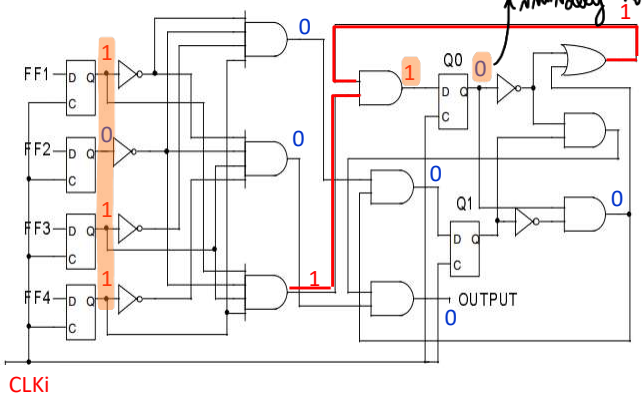
Example 3: Typical Sequential HT



- The four inputs to the Trojan are the state bits of the flip-fops in the original circuit
- Here the sequence that the Trojan is trying to detect is **1011, 0001 and 0010** in this order

M. Banga, M. S. Hsiao, "A Region based Approach for the Identification of Hardware Trojans," in Proc. of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST2008), pp. 40–47, 2008

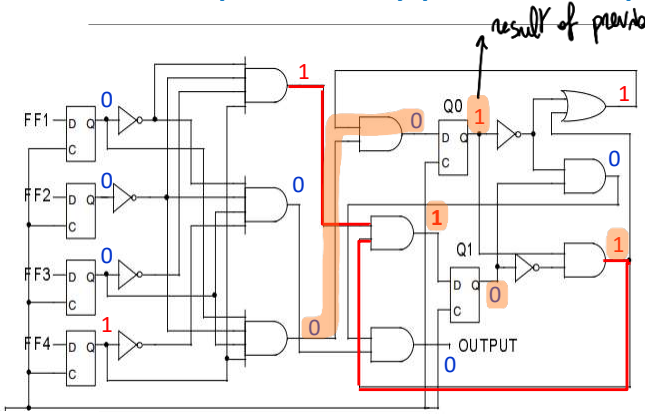
Example 3: Typical Sequential HT



- The four inputs to the Trojan are the state bits of the flip-flops in the original circuit
- Here the sequence that the Trojan is trying to detect is **1011**, 0001 and 0010 in this order

M. Banga, M. S. Hsiao, "A Region based Approach for the Identification of Hardware Trojans," in Proc. of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST2008), pp. 40–47, 2008

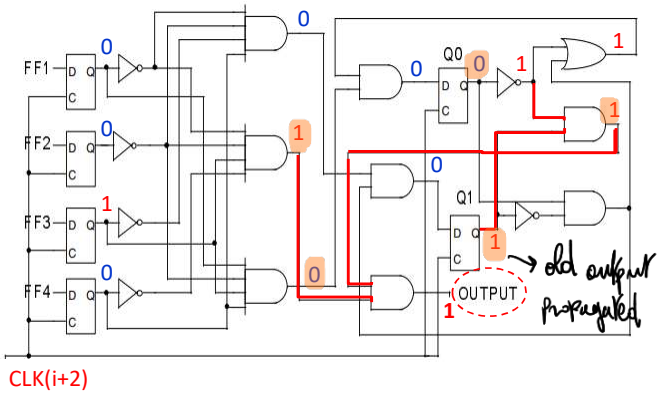
Example 3: Typical Sequential HT



- The four inputs to the Trojan are the state bits of the flip-fops in the original circuit
- Here the sequence that the Trojan is trying to detect is 1011, **0001** and 0010 in this order

M. Banga, M. S. Hsiao, "A Region based Approach for the Identification of Hardware Trojans," in Proc. of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST2008), pp. 40–47, 2008

Example 3: Typical Sequential HT



- The four inputs to the Trojan are the state bits of the flip-fops in the original circuit
- Here the sequence that the Trojan is trying to detect is 1011, 0001 and **0010** in this order
- This sequence triggers the output of the Trojan that affect one or more internal signals.

M. Banga, M. S. Hsiao, "A Region based Approach for the Identification of Hardware Trojans," in Proc. of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST2008), pp. 40–47, 2008

Hardware Trojan Characteristics: Let's Recap

- Trojan circuits are **sly**, triggering only under **rare conditions**
- Trojans are designed to be **silent most of their lifetime**, to have a **very small size** relative to their host designs, and to make only **limited contributions to circuit characteristics**
- Analysing the vulnerabilities of IC development process requires the knowledge of design, fabrication, and test processes
- To ensure a client's IC is authentic, the entire design and fabrication process must be made **trustworthy** or manufactured ICs should be **verified by clients for trustworthiness**
- Having a separate and **secure IC supply chain** is desirable but economically prohibitive

→ Need for **effective methods and techniques for Trojan prevention and detection**

Additional References and Readings

- S. Adee, "The Hunt for the Kill Switch," https://spectrum.ieee.org/semiconductors/design/the-hunt-for-the-kill-switch
- A. Avizienis, J. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, 2004, pp. 11–33.
- M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," in the IEEE Design & Test of Computers, vol.27, no.1, pp. 10–25, 2010
- X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions," in Proc. of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST2008), pp. 15–19, 2008
- J. Markoff, "Old Trick Threatens the Newest Weapons," http://www.nytimes.com/2009/10/27/science/27trojan.html?_r=3&emc=etal
- L. Wang, C. Wu, and N. Touba, "VLSI Test Principles and Architectures: Design for Testability," Morgan Kaufmann Publishers.
- Y. Alkabani and F. Koushanfar, "Consistency-Based Characterization for IC Trojan Detection," in Proc. of the International Conference on Computer-Aided Design (ICCAD09), pp. 123–127, 2009.
- F. Wolff, C. Papachristou, S. Bhunia, R.S. Chakraborty, "Towards Trojan Free Trusted ICs: Problem Analysis and Detection Scheme," in Proc. of the Design, Automation and Test in Europe (DATE08), pp. 1362–1365, 2008.
- I. Verbauwhede and P. Schaumont, "Design Methods for Security and Trust," in Proc. of the Design, Automation and Test in Europe (DATE07), pp. 672–677, 2007