Hardware Trojans

Paolo PRINETTO

Director
CINI Cybersecurity National Laboratory

Paolo.Prinetto@polito.it

Mob. +39 335 227529

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it

License & Disclaimer

License Information

This presentation is licensed under the Creative Commons BY-NC License

To view a copy of the license, visit:

http://creativecommons.org/licenses/by-nc/3.0/legalcode

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Prerequisites

- Lectures:
  - *HS_1.1 - The role of Hardware in Security*
  - *HS_1.2 - Hardware Vulnerabilities*

Acknowledgments

- The presentation includes material from
  - Giorgio DI NATALE
  - Nicolò MAUNERO
  - Gianluca ROASCIO

  whose valuable contribution is here acknowledged and highly appreciated.

Goals

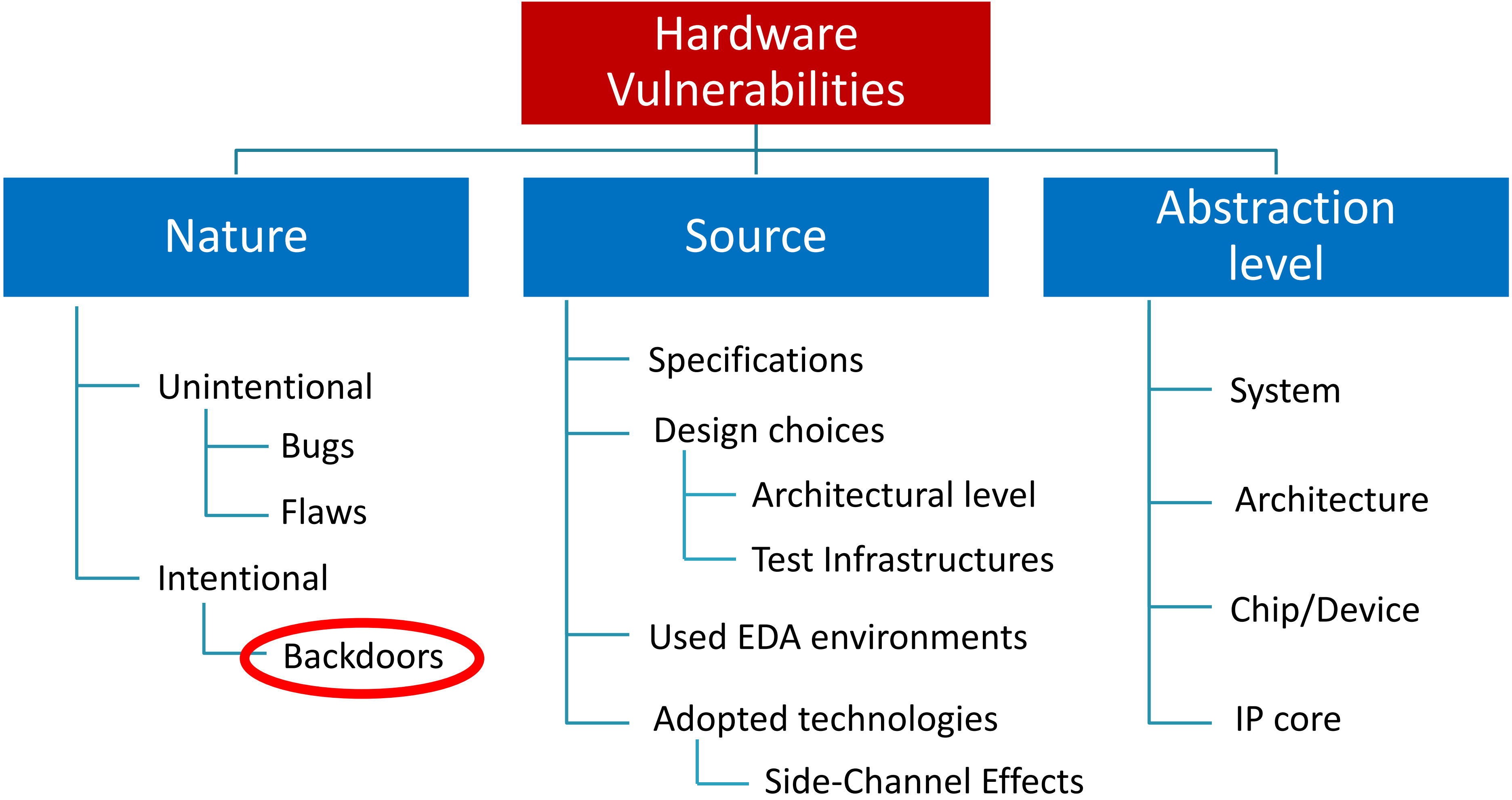
- Presenting an overview on the threat that Hardware Trojan pose today, providing a proper taxonomy.

Outline

- Introduction
- Trojans Taxonomy
- Trojans Detection

Outline

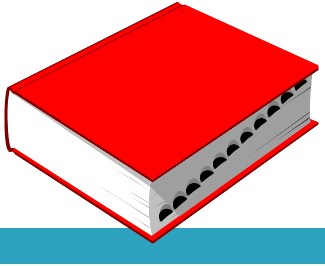
- Introduction
- Trojans Taxonomy
- Trojans Detection



Hardware Vulnerabilities

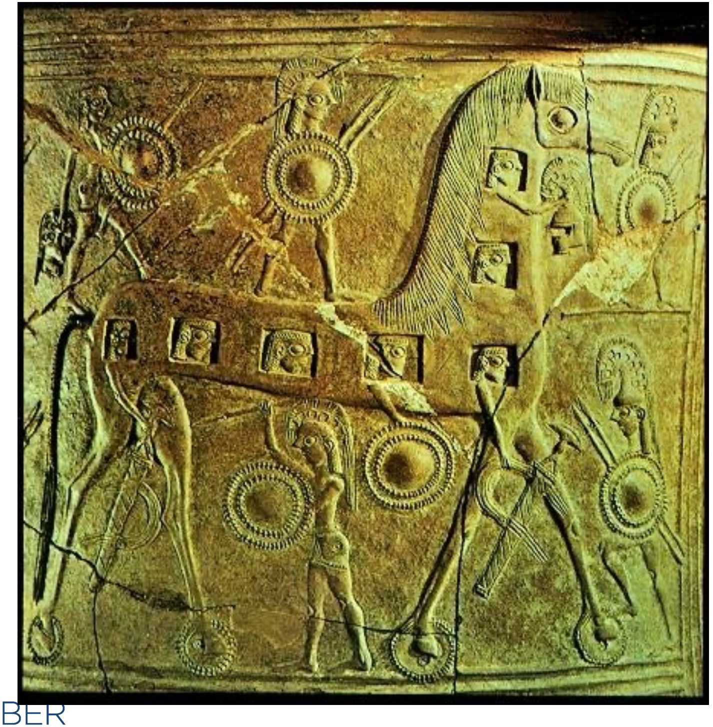
- Nature
  - Unintentional
    - Bugs
    - Flaws
  - Intentional
    - Backdoors
- Source
  - Specifications
  - Design choices
    - Architectural level
    - Test Infrastructures
  - Used EDA environments
  - Adopted technologies
    - Side-Channel Effects
- Abstraction level
  - System
  - Architecture
  - Chip/Device
  - IP core

Intentional Vulnerabilities



- A vulnerability inserted intentionally inside a hardware device can be referred to as a *backdoor*, as the person who inserts them wants to guarantee her/himself (or someone else) the possibility of a later access or use that is *outside* the set of intended use cases.

Trojan Horse



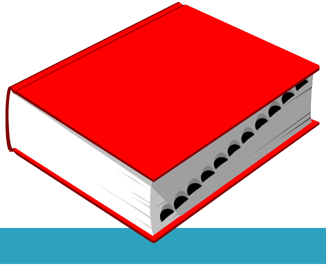
Trojan Horse

Publio Virgilio Marone - Eneide (Libro II, 49)

Hardware Trojan

- A rogue piece of circuitry fraudulently inserted during the design or production phase, which can carry out unauthorized actions when its *triggering conditions* are satisfied.

Hardware Trojan



Trigger

- The activation mechanism of the Trojan (e.g., always on, input condition, …)

Payload

- The harmful effect of Trojan activation (e.g., alter functionality, DoS, destruction, …)

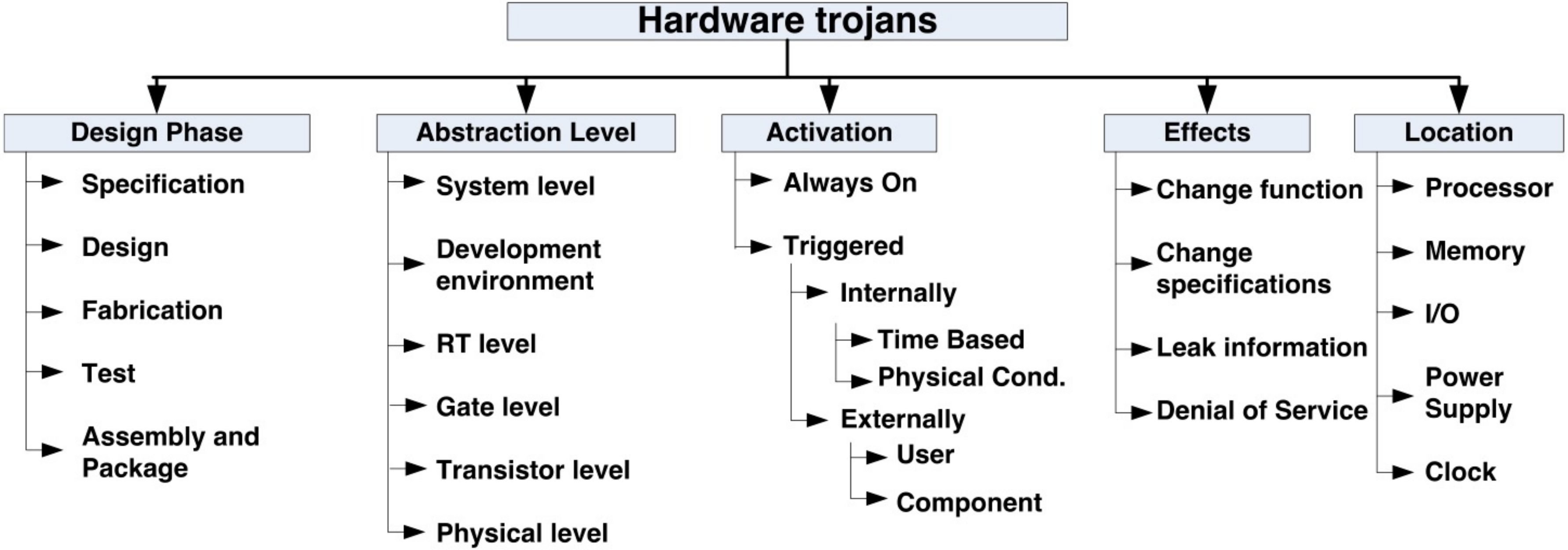
Outline

- Introduction
- Trojans Taxonomy
- Trojans Detection

Hardware Trojan Taxonomy

- HW Trojans can be clustered according to several criteria:
  - When the Trojan is inserted
  - Where the Trojan is inserted
  - How the Trojan can be activated
  - Which effects the Trojan may have

Hardware Trojan Taxonomy

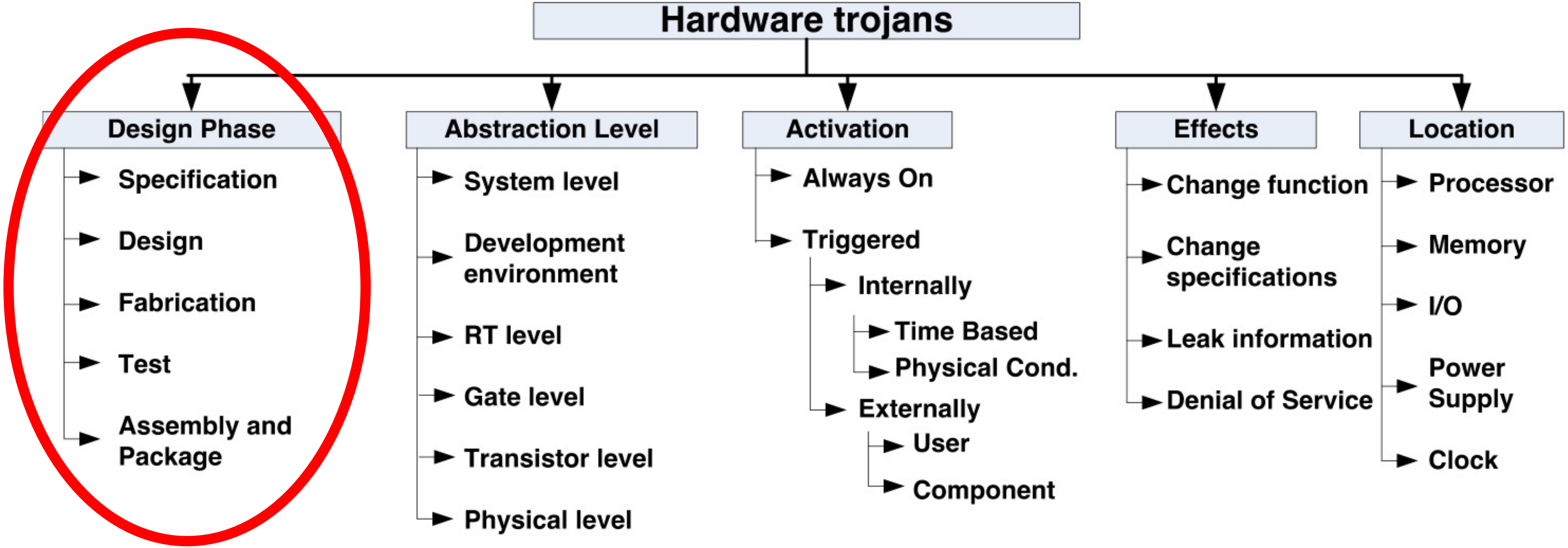


Hardware trojans

- Design Phase
  - Specification
  - Design
  - Fabrication
  - Test
  - Assembly and Package
- Abstraction Level
  - System level
  - Development environment
  - RT level
  - Gate level
  - Transistor level
  - Physical level
- Activation
  - Always On
  - Triggered
    - Internally
      - Time Based
      - Physical Cond.
    - Externally
      - User
      - Component
- Effects
  - Change function
  - Change specifications
  - Leak information
  - Denial of Service
- Location
  - Processor
  - Memory
  - I/O
  - Power Supply
  - Clock

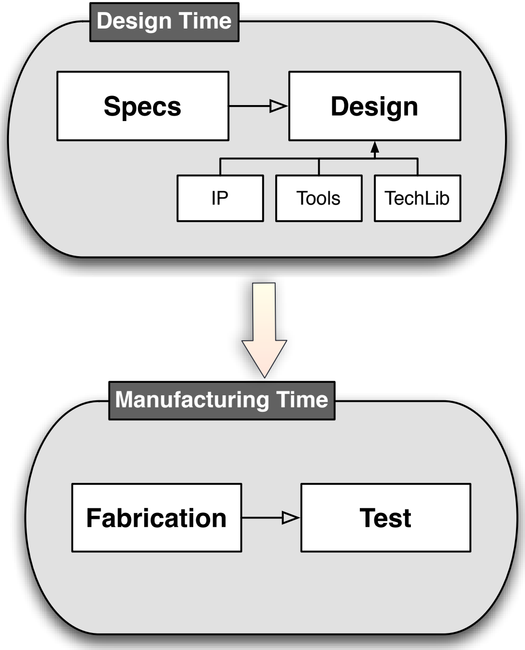
R. Karri et. al “Trustworthy Hardware: Identifying and Classifying Hardware Trojans,”
IEEE Computer, vol. 43, no. 10, pp 39-46, 2010

Hardware Trojan Taxonomy



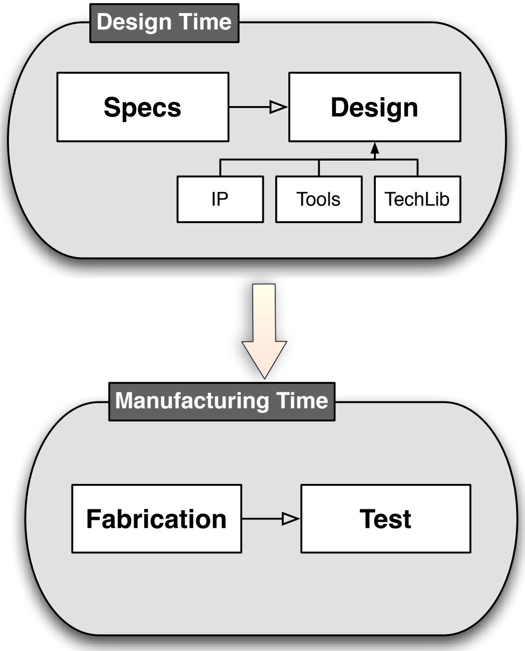
R. Karri et. al "Trustworthy Hardware: Identifying and Classifying Hardware Trojans,"
IEEE Computer, vol. 43, no. 10, pp 39-46, 2010

Design & Production Phase

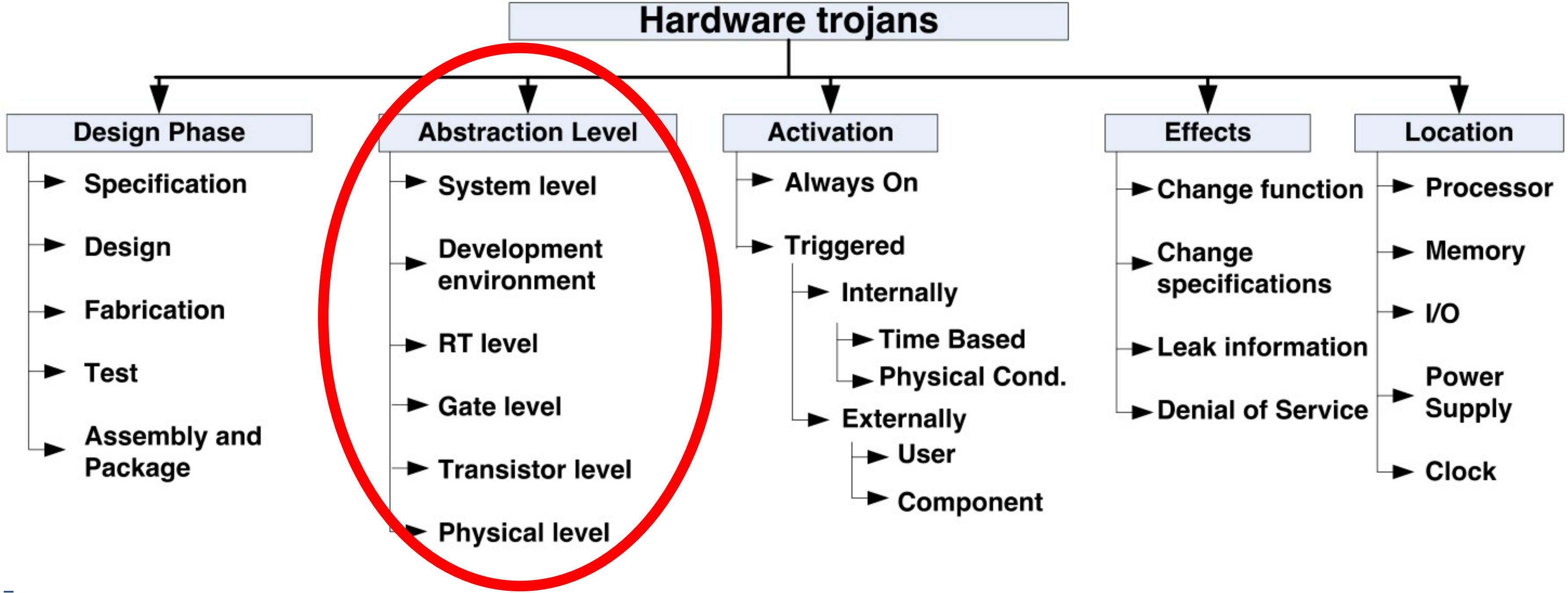


Design & Production Phase

- **Design**
  - Malicious IP core used during the design phase
  - Malicious design tools
  - Malicious designer
- **Fabrication**
  - Modification in the mask geometry and layout
  - Alteration in the chemical composition
- **Test**
  - A Trojan can be either inserted or hidden if already present
  - Untrusted Test Facilities can hide the detection of a Trojan
- **Assembly**
  - Improper termination
  - Improper shielding against phenomena such as electromagnetic interference



Hardware Trojan Taxonomy



R. Karri et. al "Trustworthy Hardware: Identifying and Classifying Hardware Trojans," IEEE Computer, vol. 43, no. 10, pp 39-46, 2010

Abstraction Level

- *System Level*

- Alteration in the interconnections
- Modification of communication protocols
- Alteration of hardware modules
- Exploitation of *active probes* for eavesdropping

Caveat

- Not ALL hardware trojans are exploited by cyber-criminals !!
- Law enforcement agencies are extensively resorting to them

Probes for active eavesdropping

- Active interceptions are mainly conducted via active network probes, i.e., network devices that can be interposed on the user's communication channel and that, in addition to intercepting traffic, can (under specific circumstances) interact with the user pretending to be the recipient.
- This is done in order, for example, to exchange false authentication certificates or to alter the data flow appropriately.

Abstraction Level

- *System Level*
- *Architectural Level*

- The ISA (Instruction Set Architecture) of a processor can include undocumented Machine Instructions, introduced:
  - Fraudulently to enable, for instance, privilege escalations
  - For debugging purposes and then not removed in the final version

Abstraction Level

- *System Level*
- *Architectural Level*
- *RT Level*

- An attacker can more easily gain info about the hardware structure and functionality

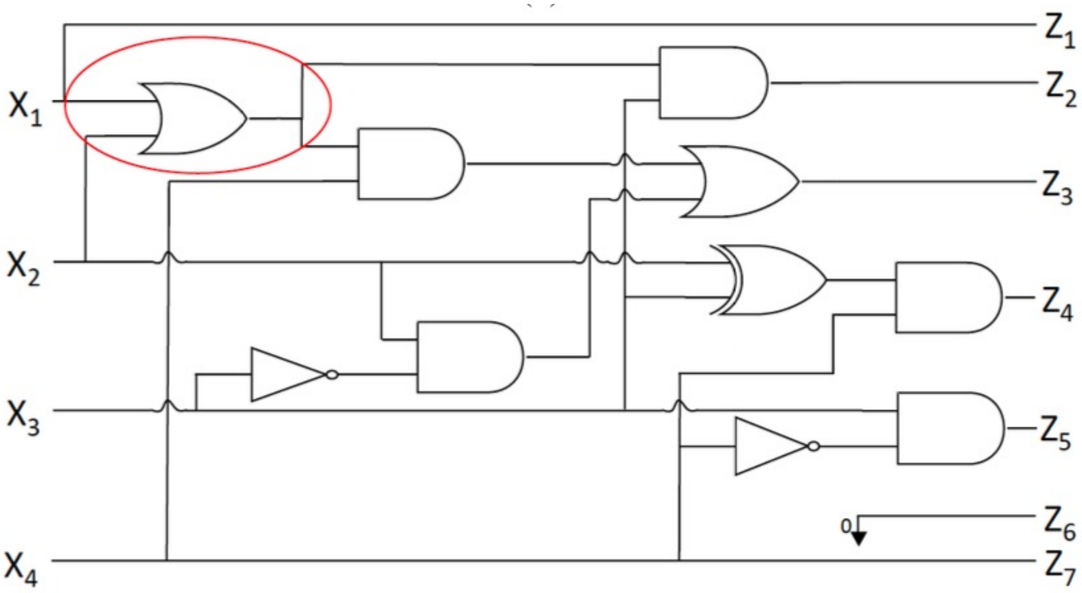
Abstraction Level

- *System Level*
- *Architectural Level*
- *RT Level*
- *Netlist Level*

- Logic gates and flip-flops are added in order to modify or inhibit some of the device functionalities

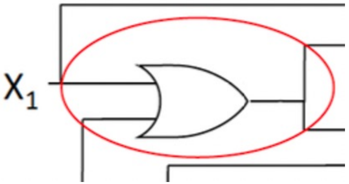
Netlist Level Trojan

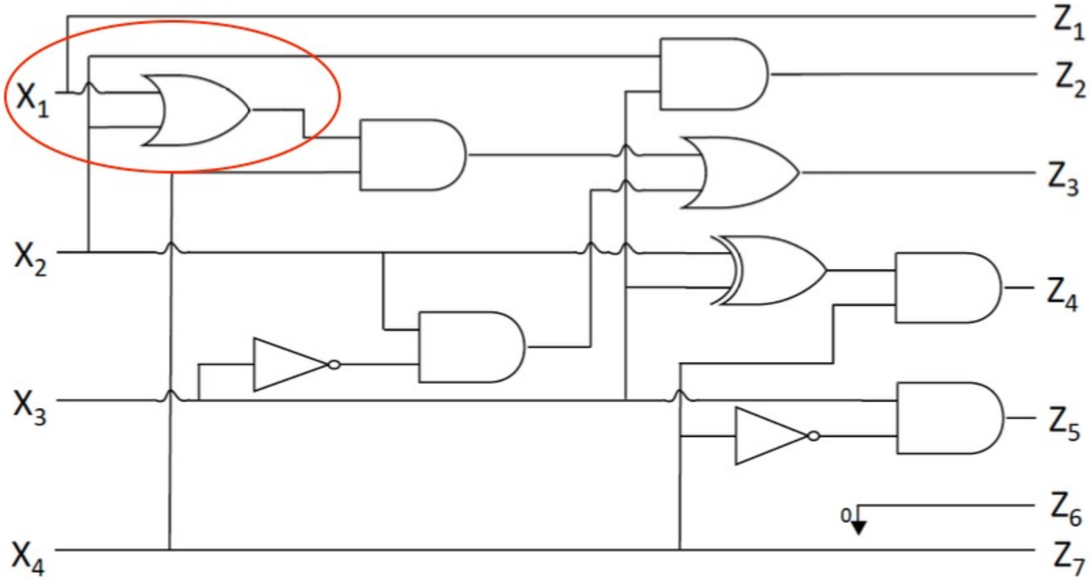
- Circuit without the Trojan



Netlist Level Trojan

- Circuit with the Trojan



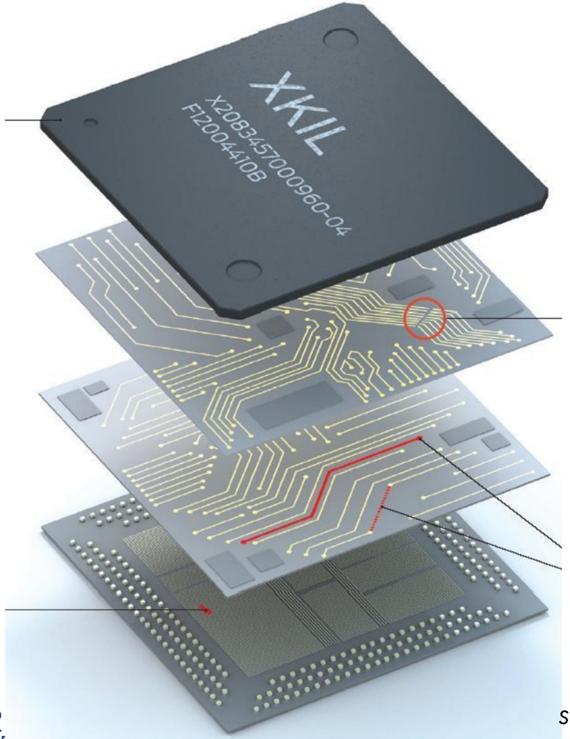


Abstraction Level

- *System Level*
- *Architectural Level*
- *RT Level*
- *Netlist Level*
- *Transistor Level*

- Resizing or deletion of existing transistors

Transistor Level Trojan



Source: *IEEE Spectrum*

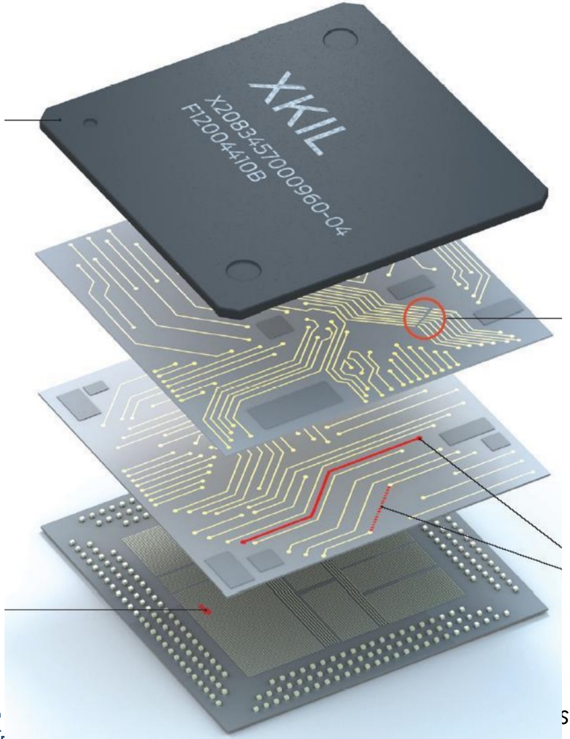
ADD EXTRA TRANSISTORS
Adding just 1000 extra transistors during either the design or the fabrication process could create a kill switch or a trapdoor. Extra transistors could enable access for a hidden code that shuts off all or part of the chip.

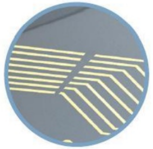
Abstraction Level

- *System Level*
- *Architectural Level*
- *RT Level*
- *Netlist Level*
- *Transistor Level*
- *Layout Level*

- Modification in transistors or layout
- Circuit is altered to affect reliability or correct functionality

Layout Level Trojan

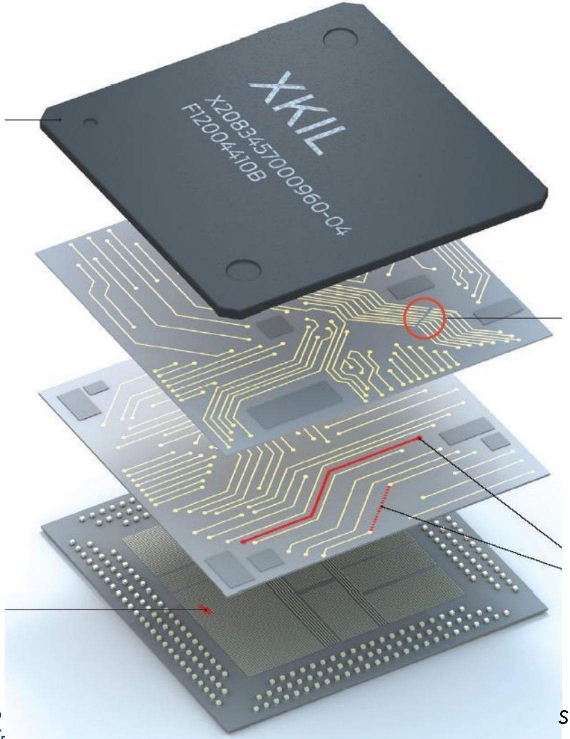


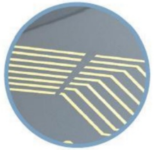


NICK THE WIRE
A notch in a few interconnects would be almost impossible to detect but would cause eventual mechanical failure as the wire became overloaded.

Source: *IEEE Spectrum*

Layout Level Trojan

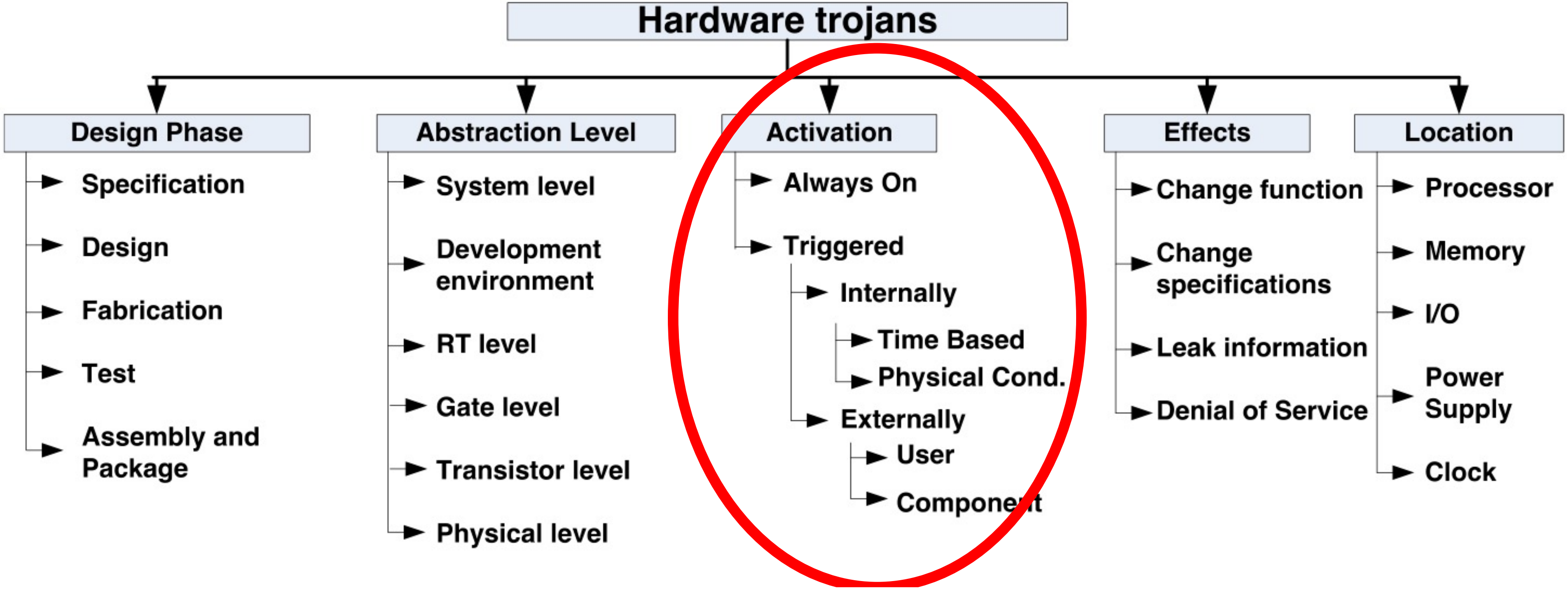




ADD OR RECONNECT WIRING
During the layout process, new circuit traces and wiring can be added to the circuit. A skilled engineer familiar with the chip’s blueprints could reconnect the wires that connect transistors, adding gates and hooking them up using a process called circuit editing.

Source: IEEE Spectrum

Hardware Trojan Taxonomy



Hardware trojans

- Design Phase
  - Specification
  - Design
  - Fabrication
  - Test
  - Assembly and Package
- Abstraction Level
  - System level
  - Development environment
  - RT level
  - Gate level
  - Transistor level
  - Physical level
- Activation
  - Always On
  - Triggered
    - Internally
      - Time Based
      - Physical Cond.
    - Externally
      - User
      - Component
- Effects
  - Change function
  - Change specifications
  - Leak information
  - Denial of Service
- Location
  - Processor
  - Memory
  - I/O
  - Power Supply
  - Clock

R. Karri et. al “Trustworthy Hardware: Identifying and Classifying Hardware Trojans,” IEEE Computer, vol. 43, no. 10, pp 39-46, 2010

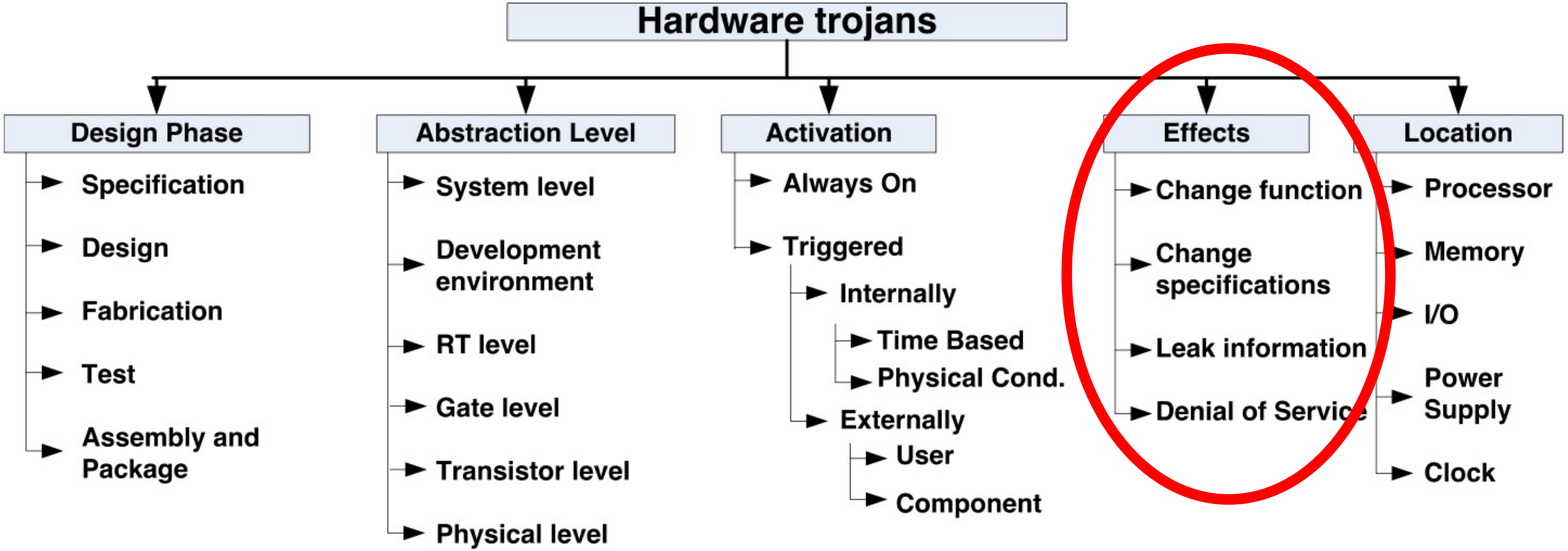
CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

Activation

- **Always On**:
  the Trojan is always active
- **Triggered**:
  the Trojan shows its effects only when activated.
  The activation condition can be
  - **Internal**: the Trojan waits for a sequence of one or more events that occur in the system. This condition is typically an internal logic state or a pattern of input/output signals.
  - **External**: the Trojan is activated by an external signal received, e.g., from an antenna or a sensor.

Hardware Trojan Taxonomy

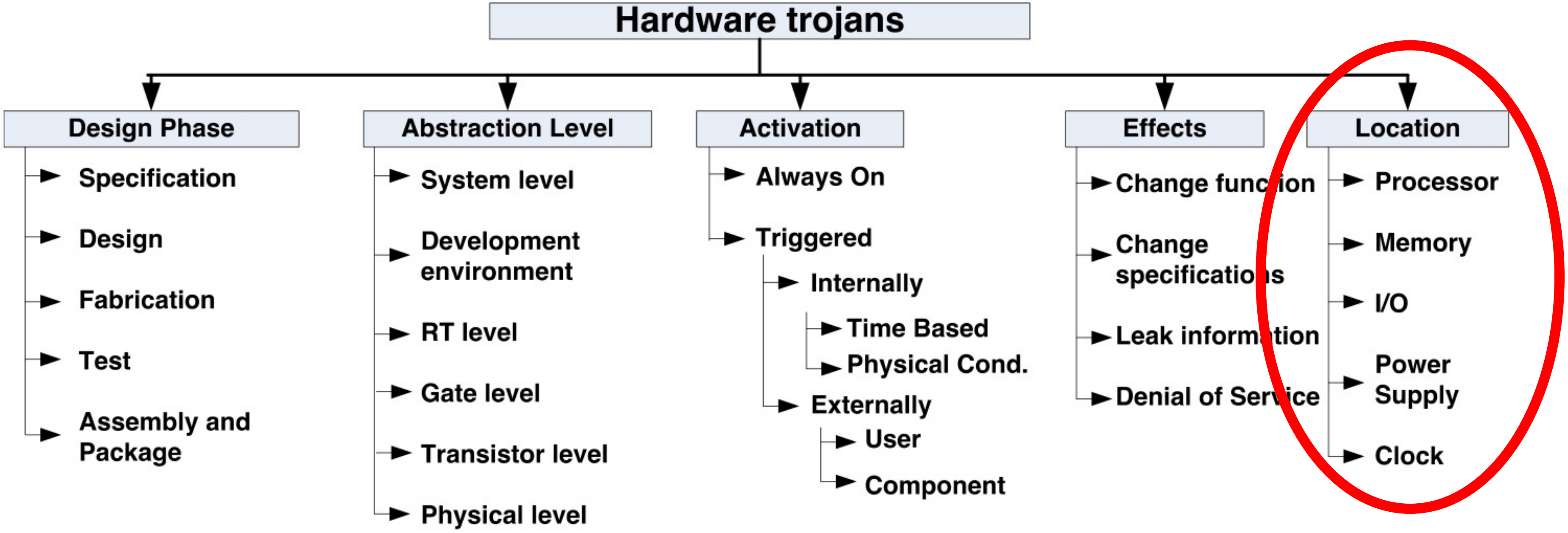


R. Karri et. al “Trustworthy Hardware: Identifying and Classifying Hardware Trojans,” IEEE Computer, vol. 43, no. 10, pp 39-46, 2010

Effects

- **Change in the functionality**:
  The Trojan can bypass, modify or delete existing logic, changing one or more of the device’s functionalities
- **Reduced reliability**:
  The Trojan can alter the reliability of the chip by modifying characteristics of the circuit such as the length of a critical path or the power consumption
- **Denial of Service (DoS)**:
  The Trojan can alter some parameters of a device to exhaust resources or introduce computational delays.

Hardware Trojan Taxonomy



Hardware trojans

- Design Phase
  - Specification
  - Design
  - Fabrication
  - Test
  - Assembly and Package
- Abstraction Level
  - System level
  - Development environment
  - RT level
  - Gate level
  - Transistor level
  - Physical level
- Activation
  - Always On
  - Triggered
    - Internally
      - Time Based
      - Physical Cond.
    - Externally
      - User
      - Component
- Effects
  - Change function
  - Change specifications
  - Leak information
  - Denial of Service
- Location
  - Processor
  - Memory
  - I/O
  - Power Supply
  - Clock

R. Karri et. al "Trustworthy Hardware: Identifying and Classifying Hardware Trojans," IEEE Computer, vol. 43, no. 10, pp 39-46, 2010

Location

- **Processor/microcontroller**:
  can be placed in the power or clock distribution grid to reduce reliability of or cause DoS attacks
- **Memory**:
  can modify address or enable/disable read/write operations
- **Input/output**:
  A Trojan placed here may have access to information exchanged between two devices, modify the communication or change the content of the exchanged data.

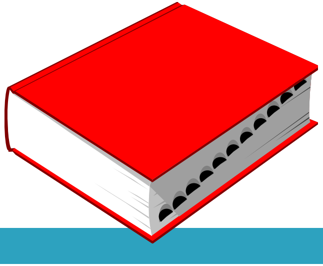
Outline

- Introduction
- Trojans Taxonomy
- Trojans Detection

Trojan Detection

- Detecting Hardware Trojans can be seen as a “usual” *Validation & Verification (V&V) step*

Validation



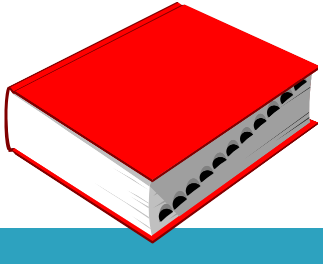
- The process of evaluating the system at the end of the development process, to ensure compliance with system requirements

[IEEE standard glossary of
Software Engineering terminology]

Validation goals

- Checking the correspondence of the intermediate artifacts and the final product to users’ expectations

Verification



- The process of determining whether the product of a given phase of the system development cycle fulfils the requirements established during the previous phase, or not

[IEEE standard glossary of
Software Engineering terminology]

Verification Goals

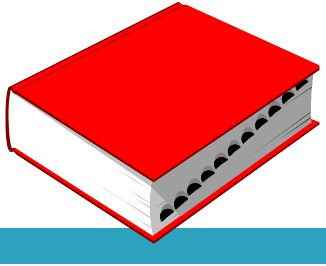
- Steering the process toward the construction of a product that satisfies the requirements by checking the quality of intermediate artifacts as well as the ultimate product

[Mauro Pezzè & Michal Young
“Software Testing and Analysis: Process, Principles and Techniques”
Wiley, 2008]

Trojan Detection Approaches

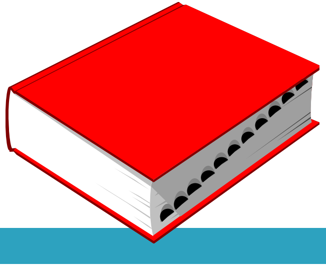
- Real industrial cases:
  - *Simulation*
  - *Functional Verification*
  - *Emulation*
  - *Formal Verification*
  - *Model Checking*
  - …
- Training phase:
  - *Reverse Engineering*
  - *Visual Inspection*

Formal Verification



- Aims at proving, resorting to a mathematical reasoning, once and for all, regardless the system state and the input sequences, the existence of a given relationship between two entities (e.g., Specification vs Implementation)

Model Checking



- Aims at proving whether a system description satisfies a given set of properties, or not

Reverse Engineering

- Rely on your design experiences in order to identify differences between the two entities
- Some possible cases are presented in the sequel

Possible cases

Trojan-free entity

- Informal specs
- Behavioral RT-level description
- Structural RT-level description

Corrupted entity

- Structural RT-level description
- Structural Gate-level description (Netlist)

Possible cases

Trojan-free entity

- Informal specs
- Behavioral RT-level description
- Structural RT-level description

Corrupted entity

- Structural RT-level description
- Structural Gate-level description (Netlist)

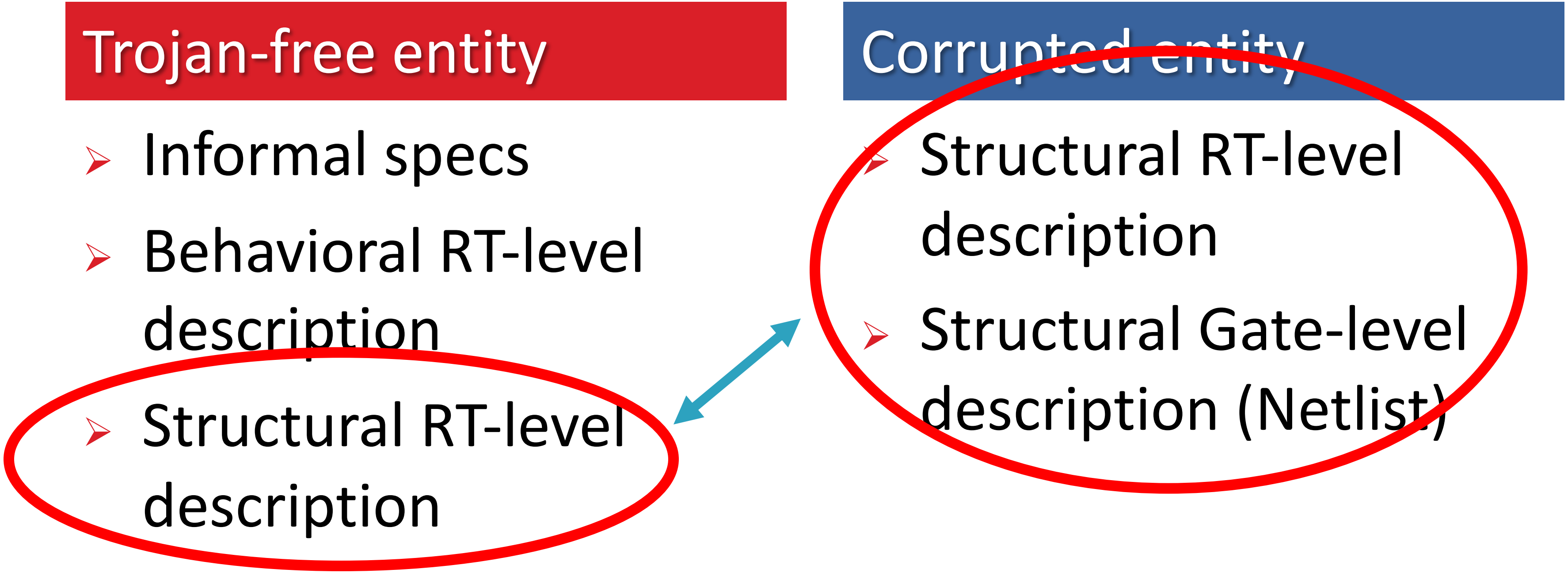
Hints

- Start from the Corrupted description
- Analyze it carefully:
  - identify the functional blocks used to implement the various use cases of the Trojan-free entities
  - mark them
- The component left un-marked most likely are part of the Trojan

Hints

- Try to identify:
  - The activation sequence
  - The payload of the Trojan

Possible cases



Trojan-free entity

- Informal specs
- Behavioral RT-level description
- Structural RT-level description

Corrupted entity

- Structural RT-level description
- Structural Gate-level description (Netlist)

Hints

- Analyze concurrently the 2 descriptions:
  - Find a match between the functional blocks of the 2 descriptions
  - Mark them
- The components left un-marked in the Corrupted entity most likely are part of the Trojan

Hints

- Try to identify:
  - The activation sequence
  - The payload of the Trojan

Малые Автюхи, Калинковичский район
Республики Беларусь

Paolo PRINETTO

Director
CINI Cybersecurity National Laboratory

Paolo.Prinetto@polito.it

Mob. +39 335 227529

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it