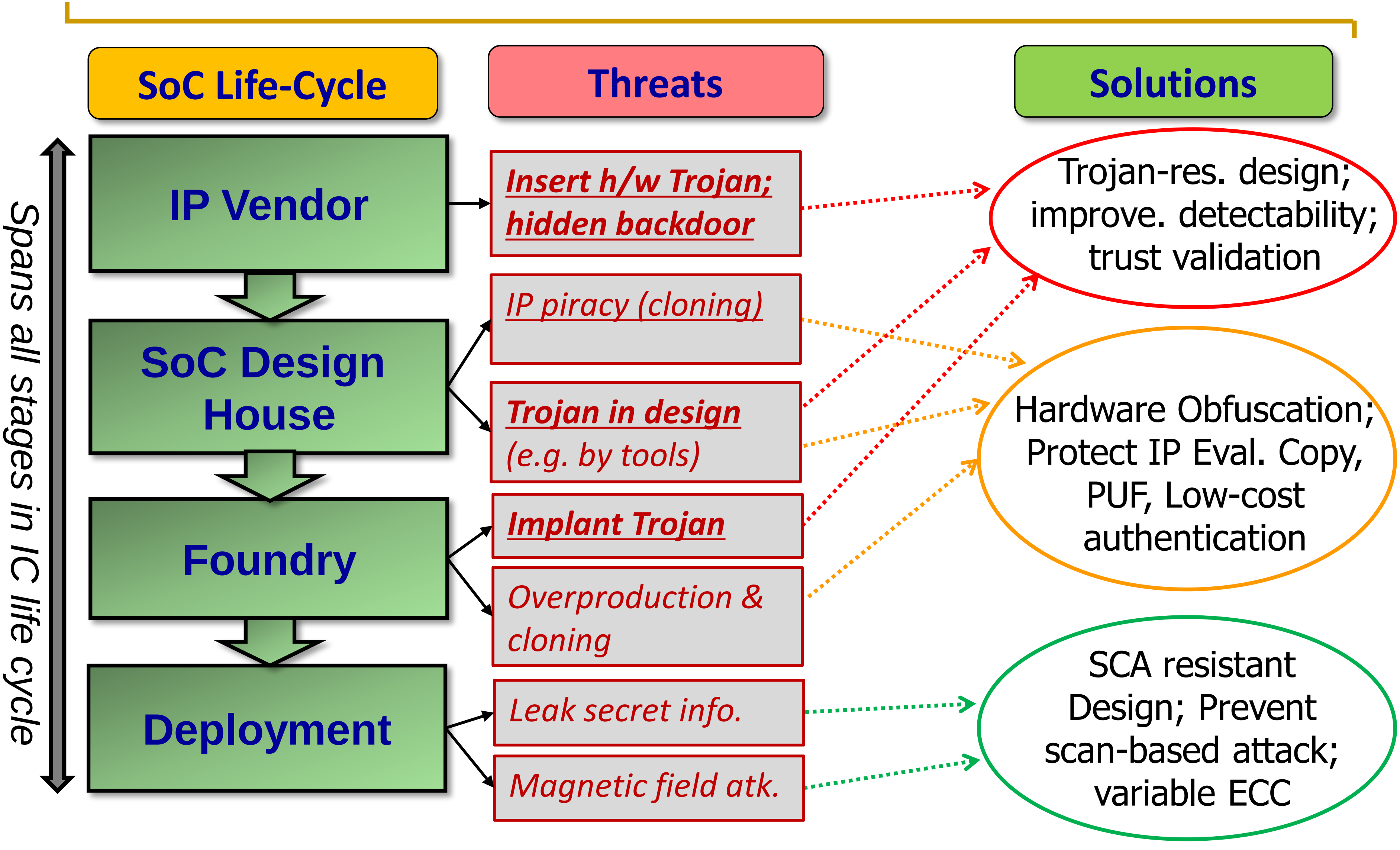
Practical Introduction to Hardware Security

Lecture 7: Hardware Trojan

Instructors: Mehdi Tahoori, Dennis Gnad, Jonas Krautter

INSTITUTE OF COMPUTER ENGINEERING (ITEC) – CHAIR FOR DEPENDABLE NANO COMPUTING (CDNC)

Threats


SoC Life-Cycle
Threats
Solutions
Spans all stages in IC life cycle
IP Vendor
SoC Design House
Foundry
Deployment
Insert h/w Trojan; hidden backdoor
IP piracy (cloning)
Trojan in design (e.g. by tools)
Implant Trojan
Overproduction & cloning
Leak secret info.
Magnetic field atk.
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

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

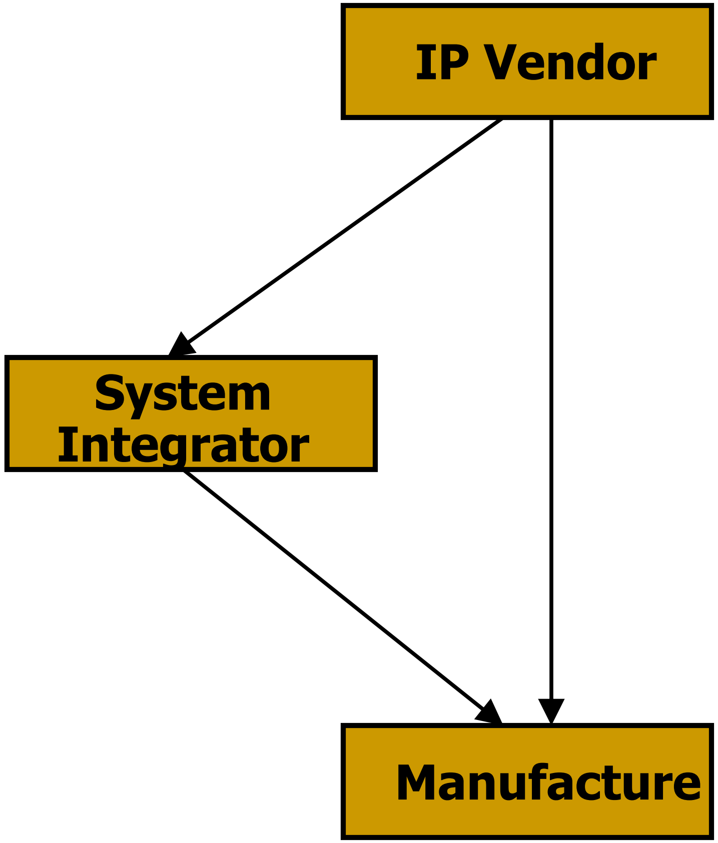
IC/IP Trust Problem

- 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*

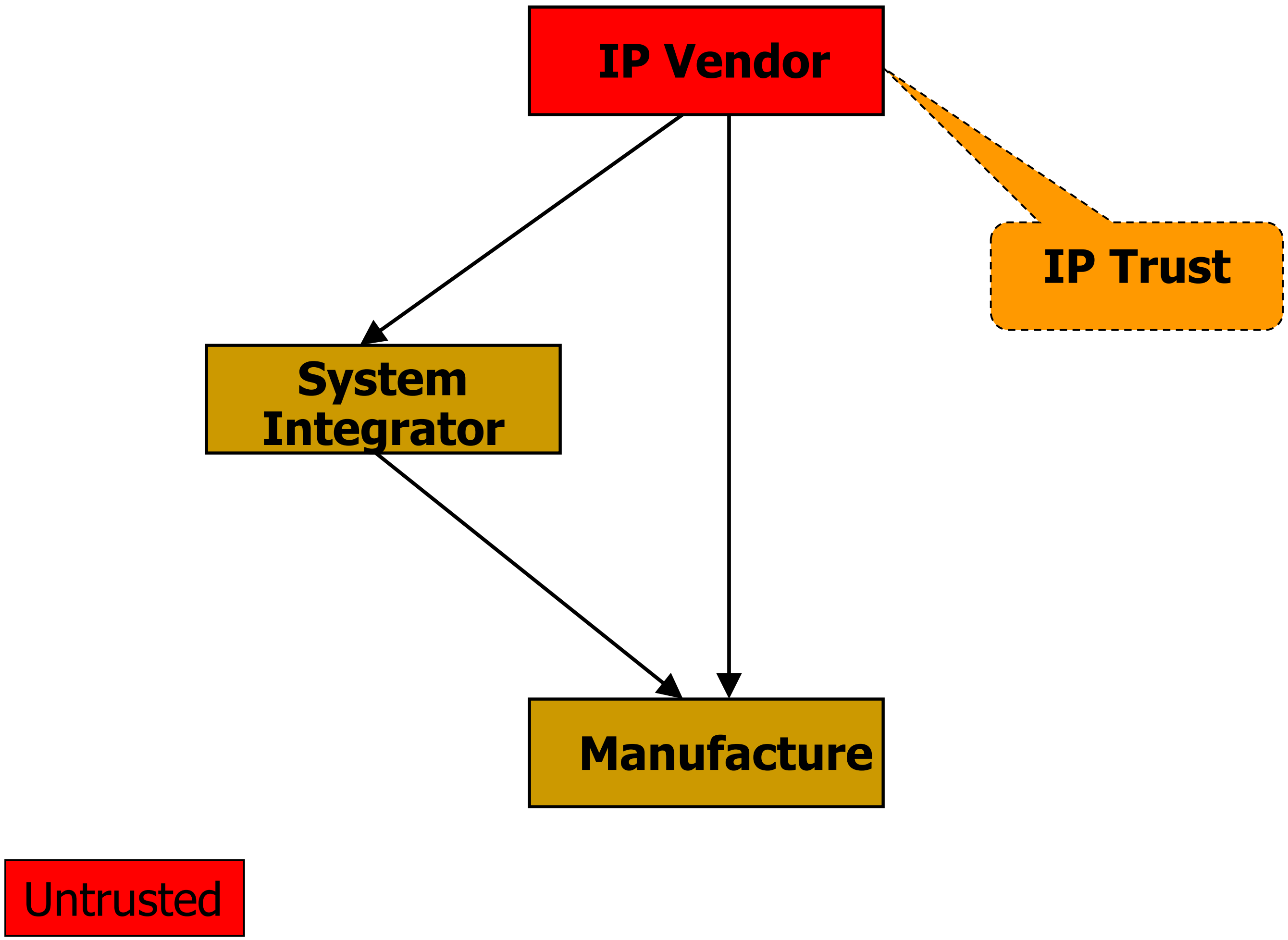
- **Designer and Foundry:**
  - Foundry may place a Trojan in the layout design.
  - *IC Trust problem*

Hardware Trojan Threat

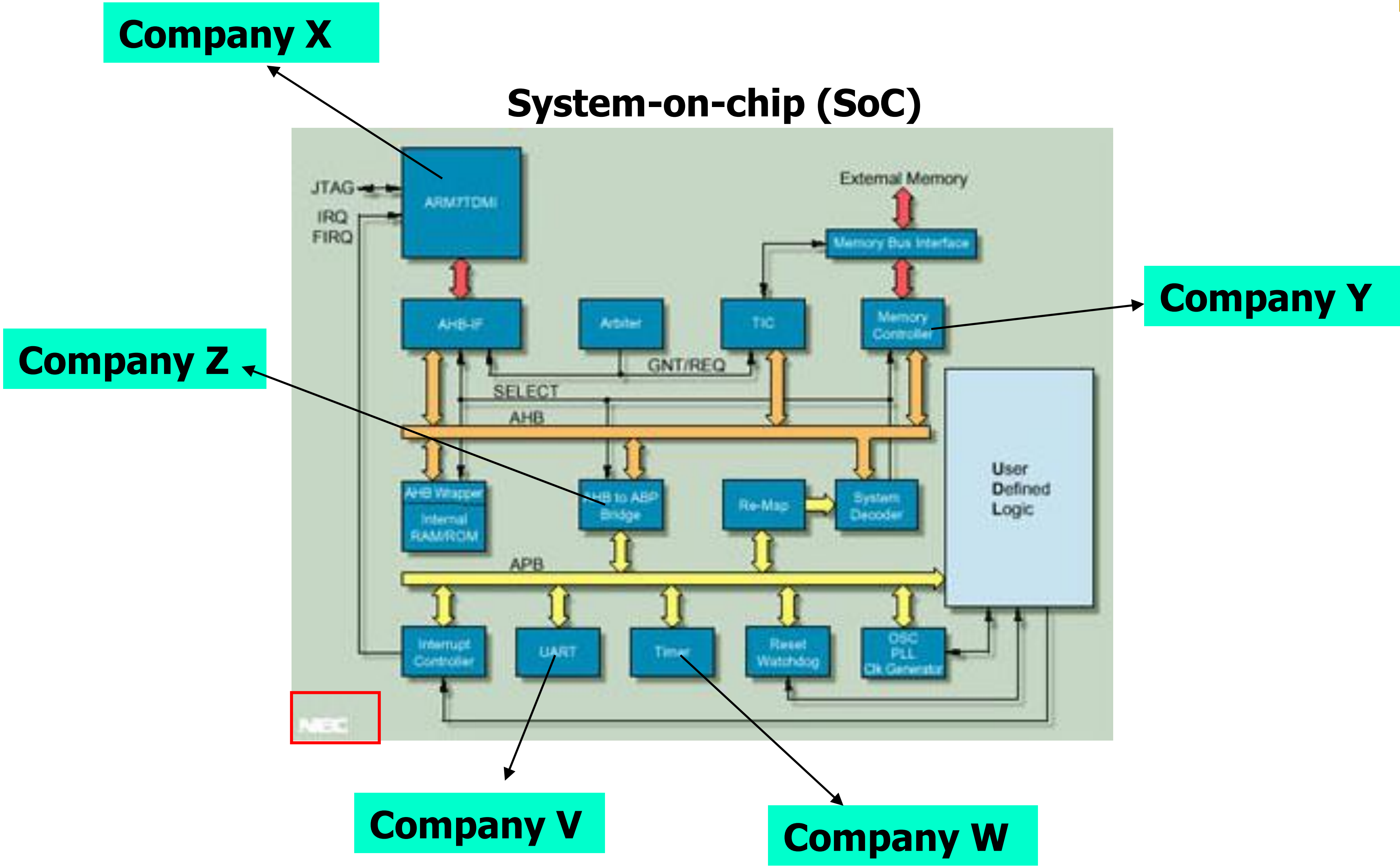


Any of these steps can be untrusted

Hardware Trojan Threat



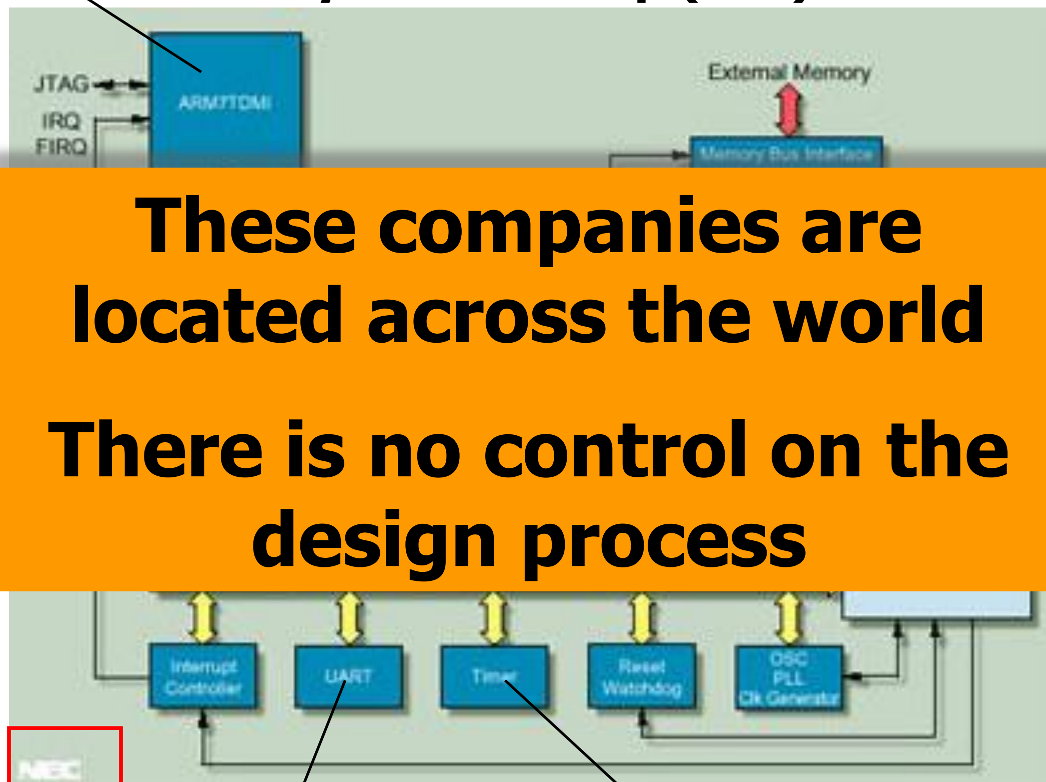
Issues with Third IP Design



Issues with Third IP Design

Company X

System-on-chip (SoC)



Company Y

Company Z

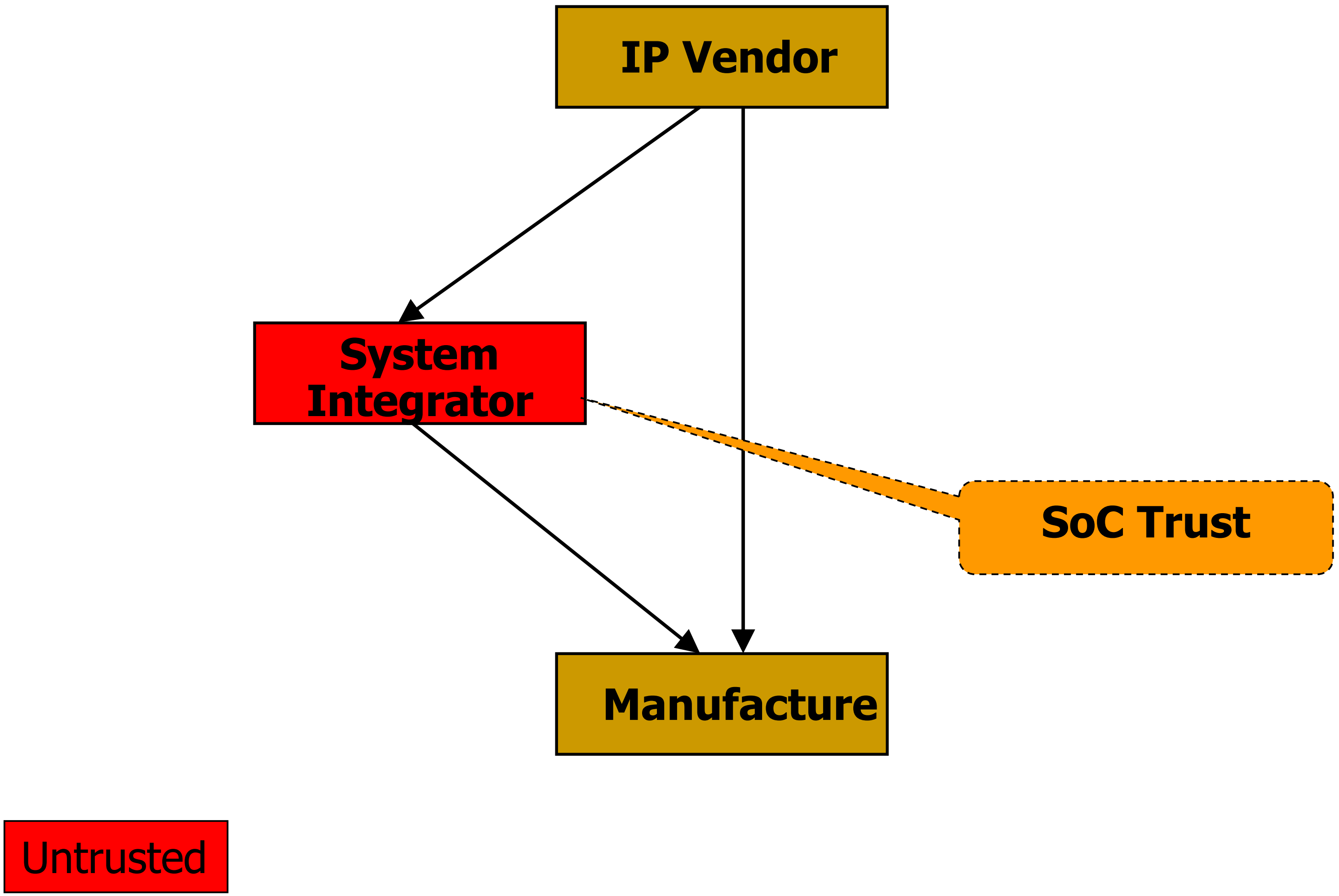
These companies are located across the world

There is no control on the design process

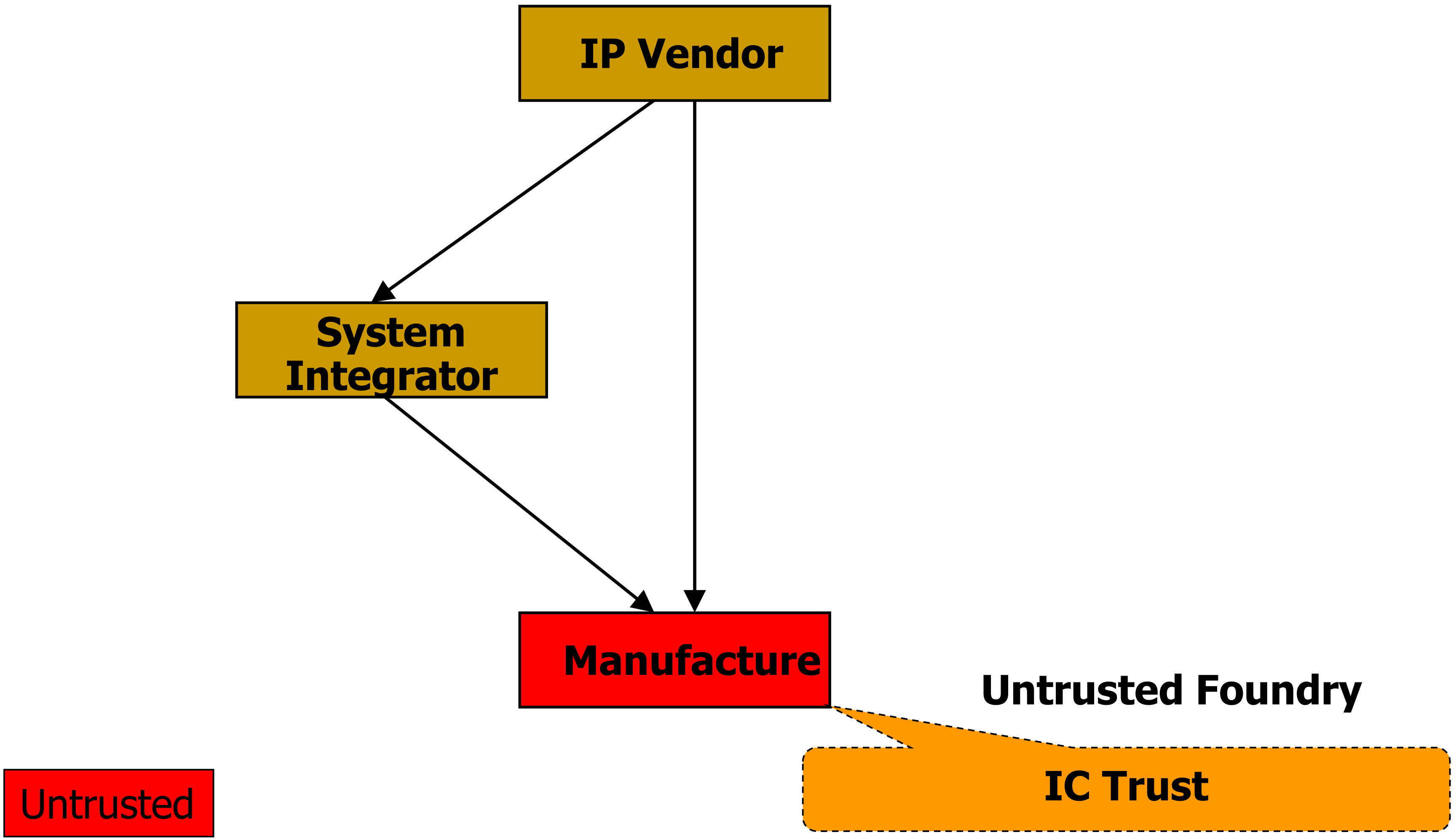
Company V

Company W

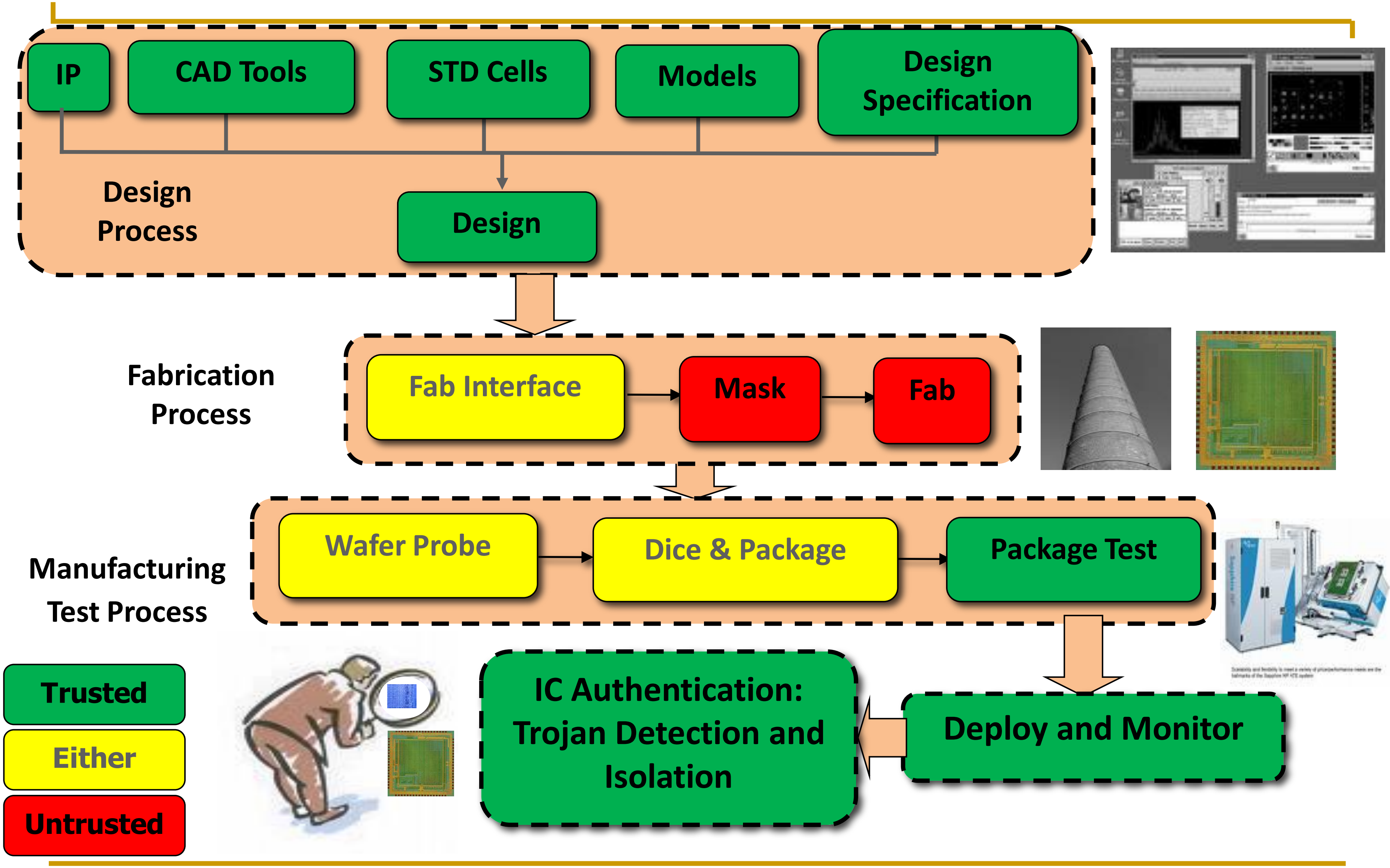
Hardware Trojan Threat



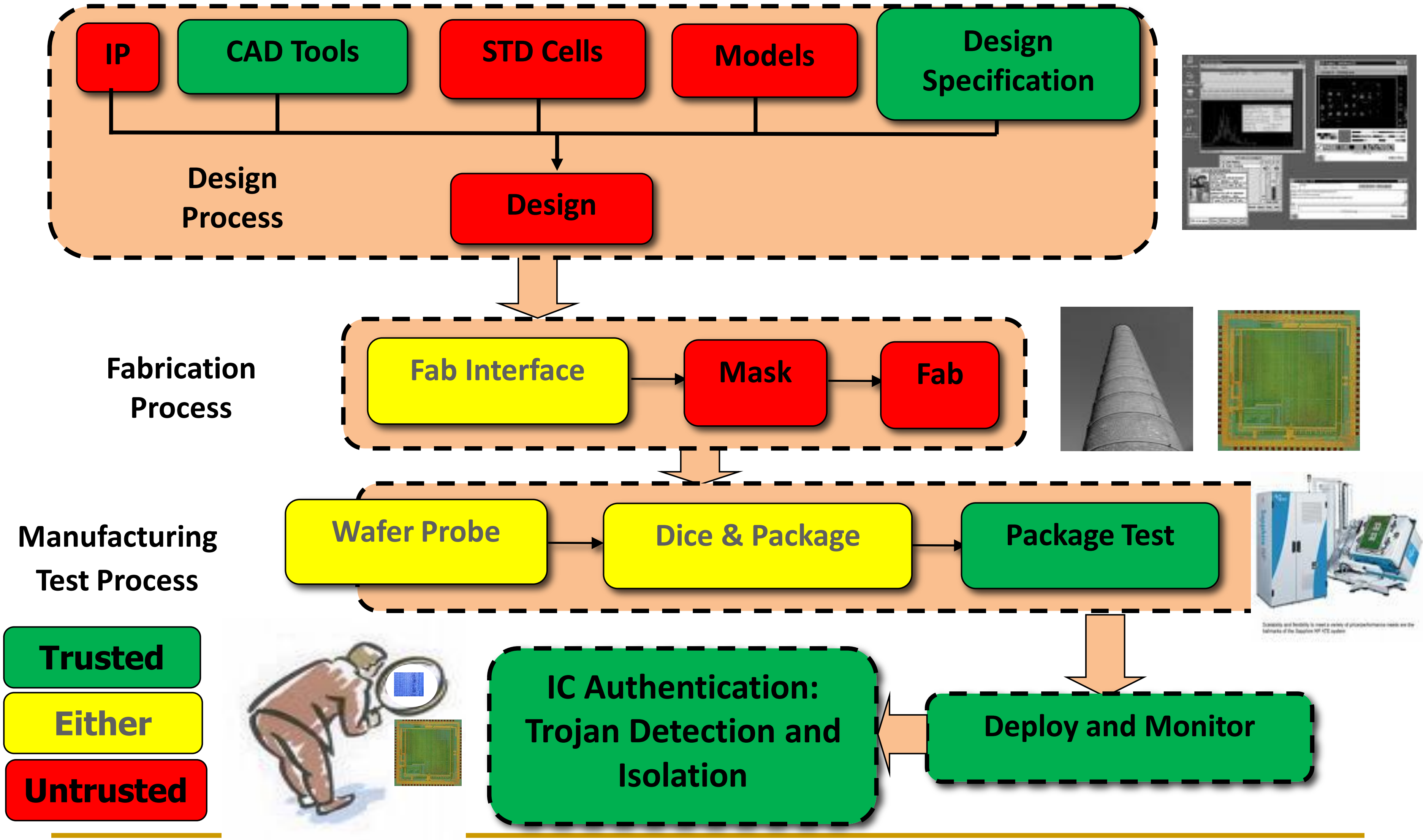
Hardware Trojan Threat



ASIC Design Process – Untrusted Foundry

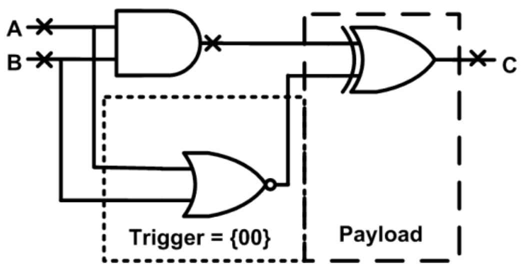


Untrusted Designer and Foundry

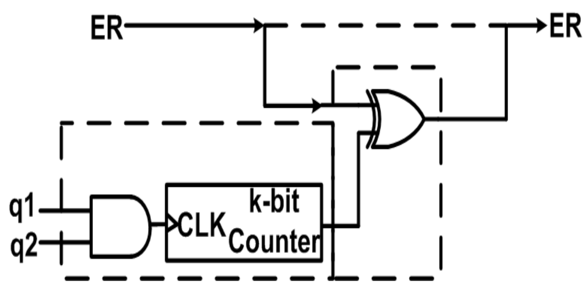


HW Trojan Examples / Models

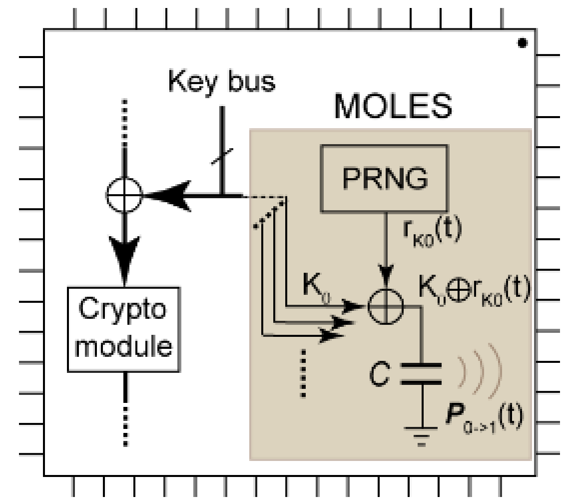
Comb. Trojan Example



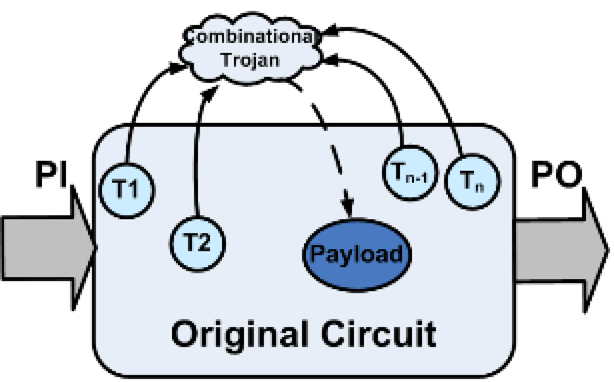
Seq. Trojan Example



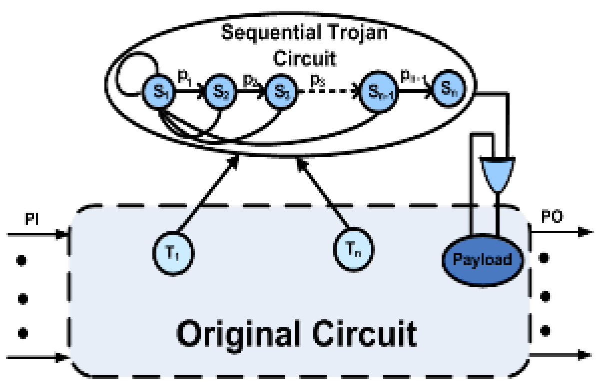
MOLES: Info Leakage Trojan*



Comb. Trojan model



Seq. Trojan Model



HW Trojan evidence!

*Lin et al, ICCAD 2009

Why is detection of hardware Trojans very difficult?

Bug vs. Malicious Change

Verification (Traditional)

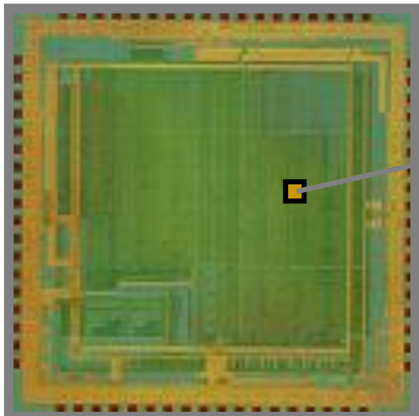
- Bugs (Unintentional)
- Bounded by Spec

Trust Verification

- Malicious change (**<u>Intentional</u>**)
- Unwanted functionality (**<u>Unbounded</u>**)

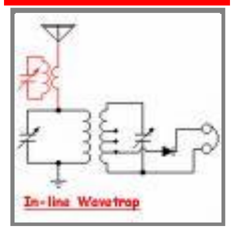
Trojan Attacks → BIGGER verification challenge!

Silicon Back Door



Untrusted Hardware

Antenna



- Adversary can send and receive secret information
- Adversary can disable the chip, blowup the chip, send wrong processing data, impact circuit information etc.

- Adversary can place an Antenna on the fabricated chip
- Such Trojan cannot be detected since it does not change the functionality of the circuit.

Silicon Time Bomb

Untrusted Hardware

Counter

Finite state machine (FSM)

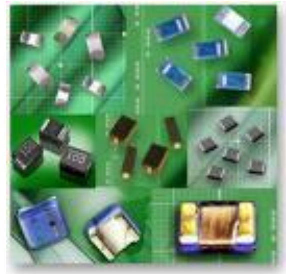
Comparator to monitor key data

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**

Applications and Threats

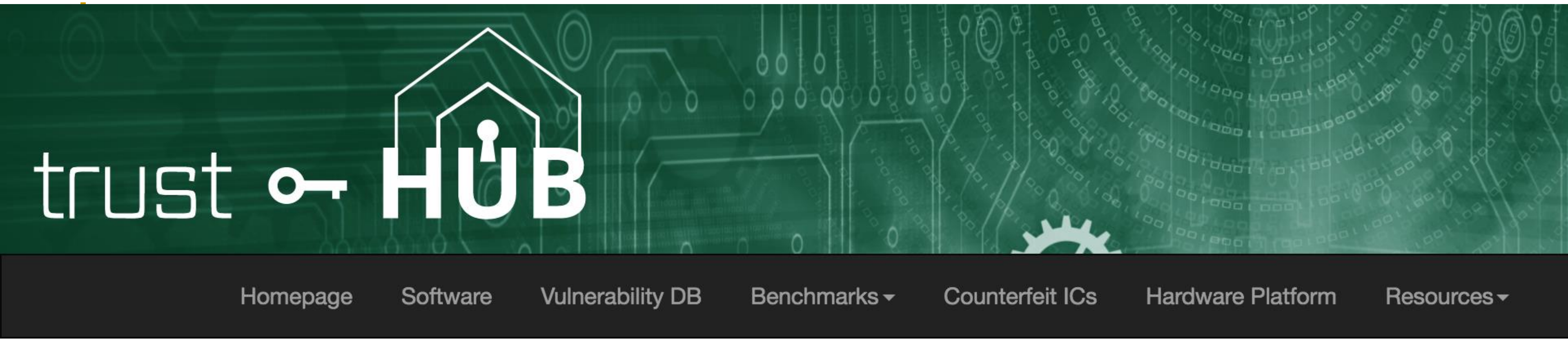
Thousands of chips are being fabricated in untrusted foundries



Comprehensive Attack Model

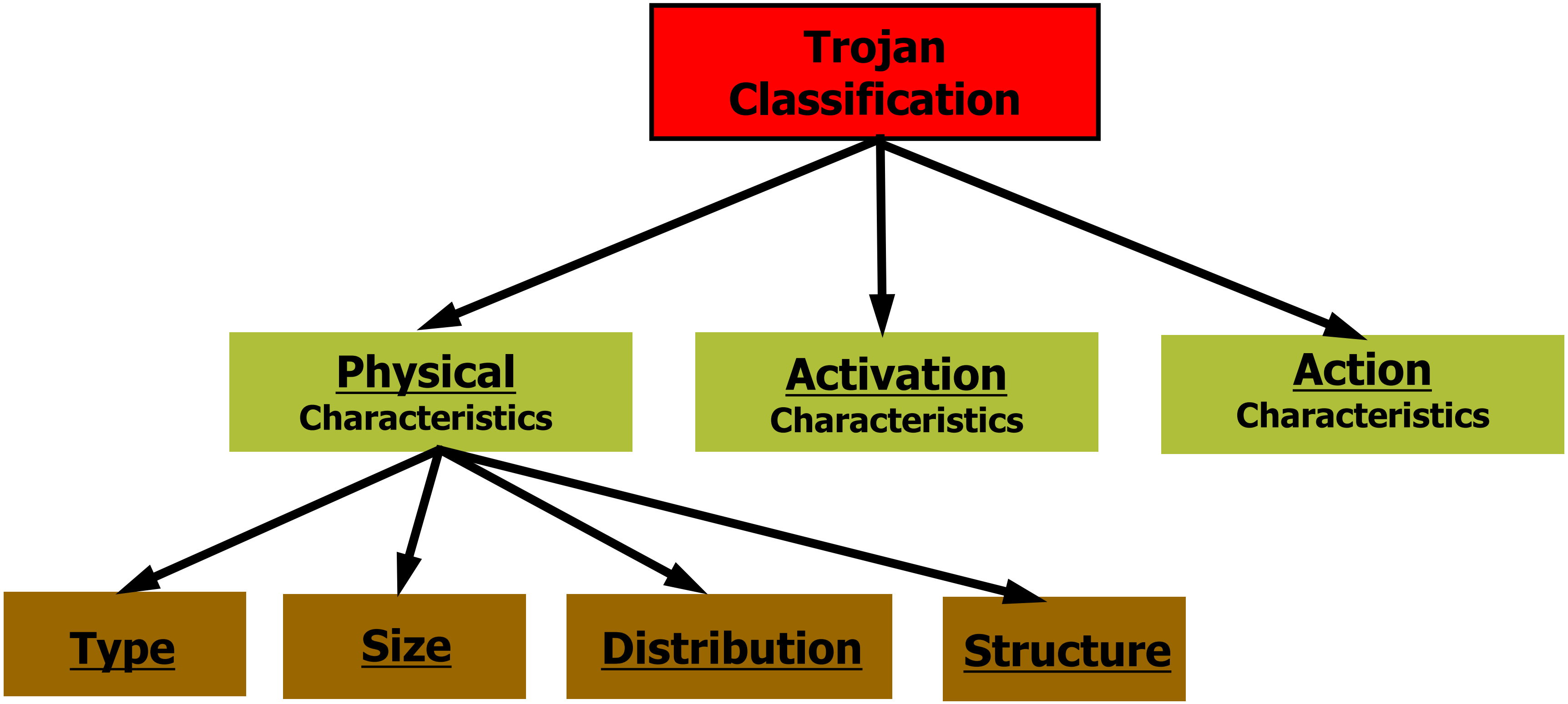
Model	Description	3PIP Vendor	SoC Developer	Foundry
A	Untrusted 3PIP vendor	Untrusted	Trusted	Trusted
B	Untrusted foundry	Trusted	Trusted	Untrusted
C	Untrusted EDA tool or rogue employee	Trusted	Untrusted	Trusted
D	Commercial-off-the-shelf component	Untrusted	Untrusted	Untrusted
E	Untrusted design house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

Trojan Taxonomy



- Hardware Trojans
  - Insertion Phase
    - Specification
    - Design
    - Fabrication
    - Testing
    - Assembly and Package
  - Abstraction Level
    - System
    - Development Environment
    - Register Transfer
    - Gate
    - Layout
    - Physical
  - Activation Mechanism
    - Always On
    - Triggered
      - Internally
        - Time Based
        - Physical Condition Based
      - Externally
        - User Input
        - Component Output
  - Effect
    - Change Functionality
    - Degrade Performance
    - Leak Information
    - Denial of Service
  - Location
    - Processor
    - Memory
    - I/O
    - Power Supply
    - Clock Grid
  - Physical Characteristic
    - Distribution
    - Size
    - Type
      - Parametric
      - Functional
    - Structure
      - Layout Same
      - Layout Change

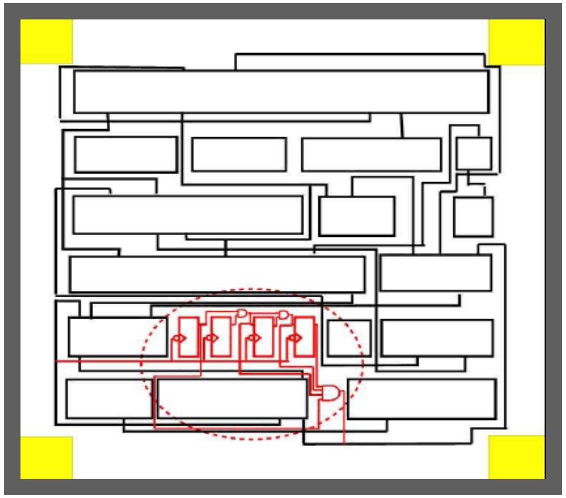
Trojan Taxonomy

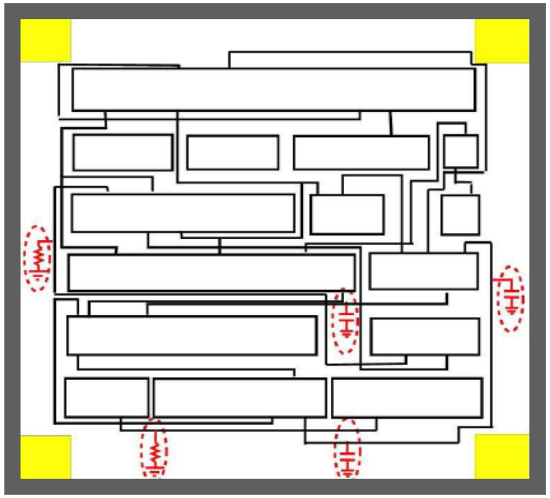


Examples for Layout Level Trojans

Example: Type

Functional | **Parametric**

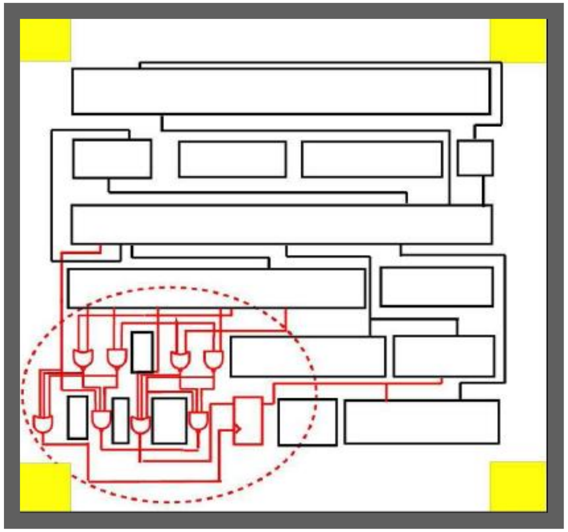




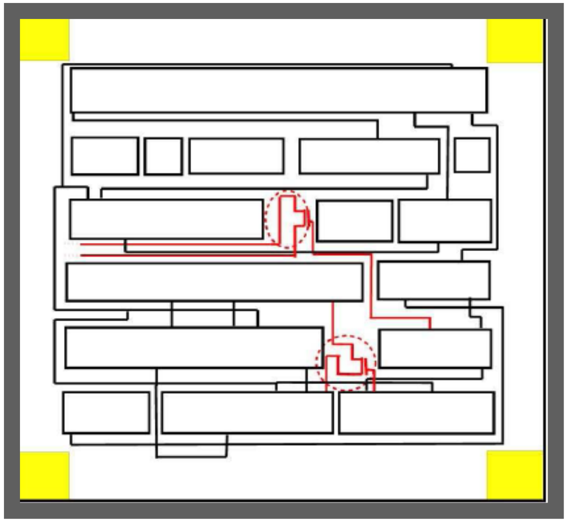
- **Functional**
  - **Addition or deletion of components**
  - **Sequential circuits**
  - **Combinational circuits**
  - **Modification to function or no change**
- **Parametric**
  - **Modifications of existing components**
    - **Wire: e.g. thinning of wires**
    - **Logic: Weakening of a transistor, 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

Big



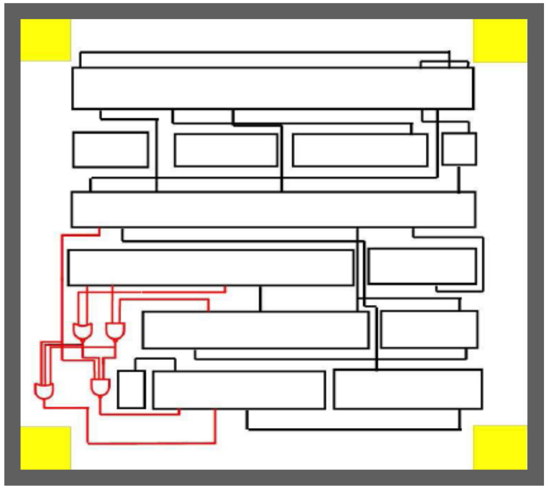
Small



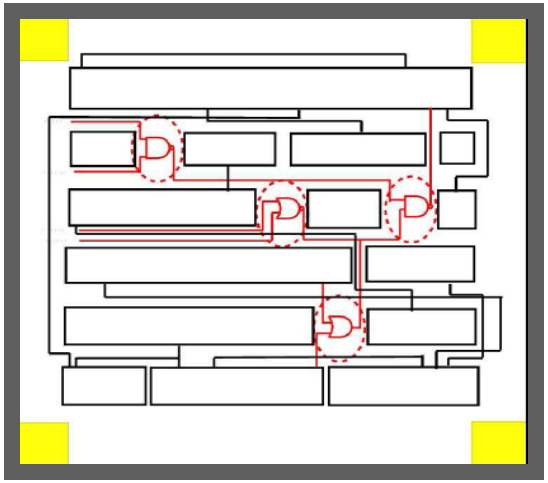
- **Size:**
  - **Number of components added to the circuit**
    - **Small transistors**
    - **Small gates**
    - **Large gates**
- **In case of layout, depends on availability of:**
  - **Dead spaces**
  - **Filler cells**
  - **Decap cells**
  - **Change in the structure**

Example: Distribution

Tight



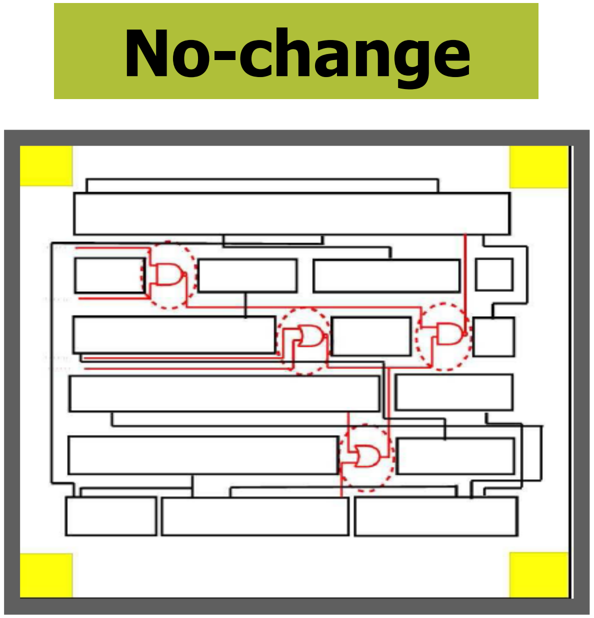
Loose

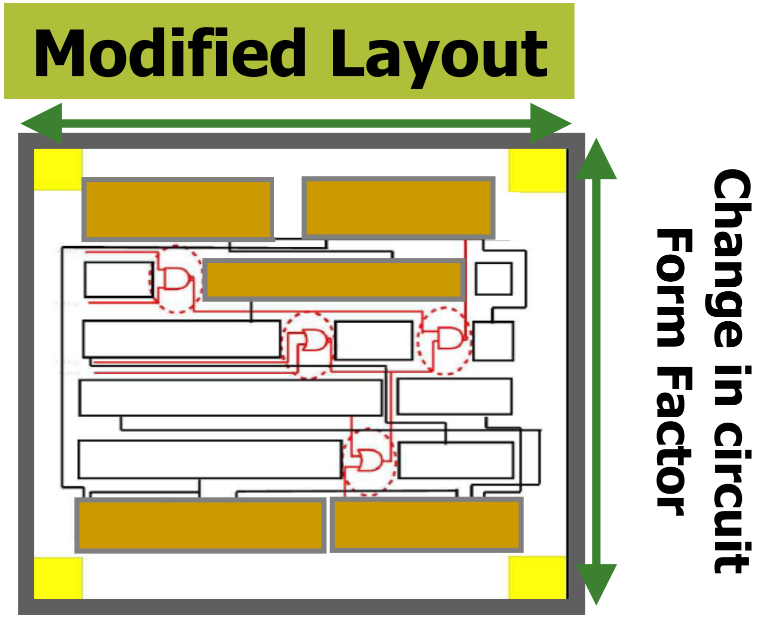


- **Tight Distribution**
  - **Trojan components are topologically close in the layout**
- **Loose Distribution**
  - **Trojan components are dispersed across the layout of a chip**

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

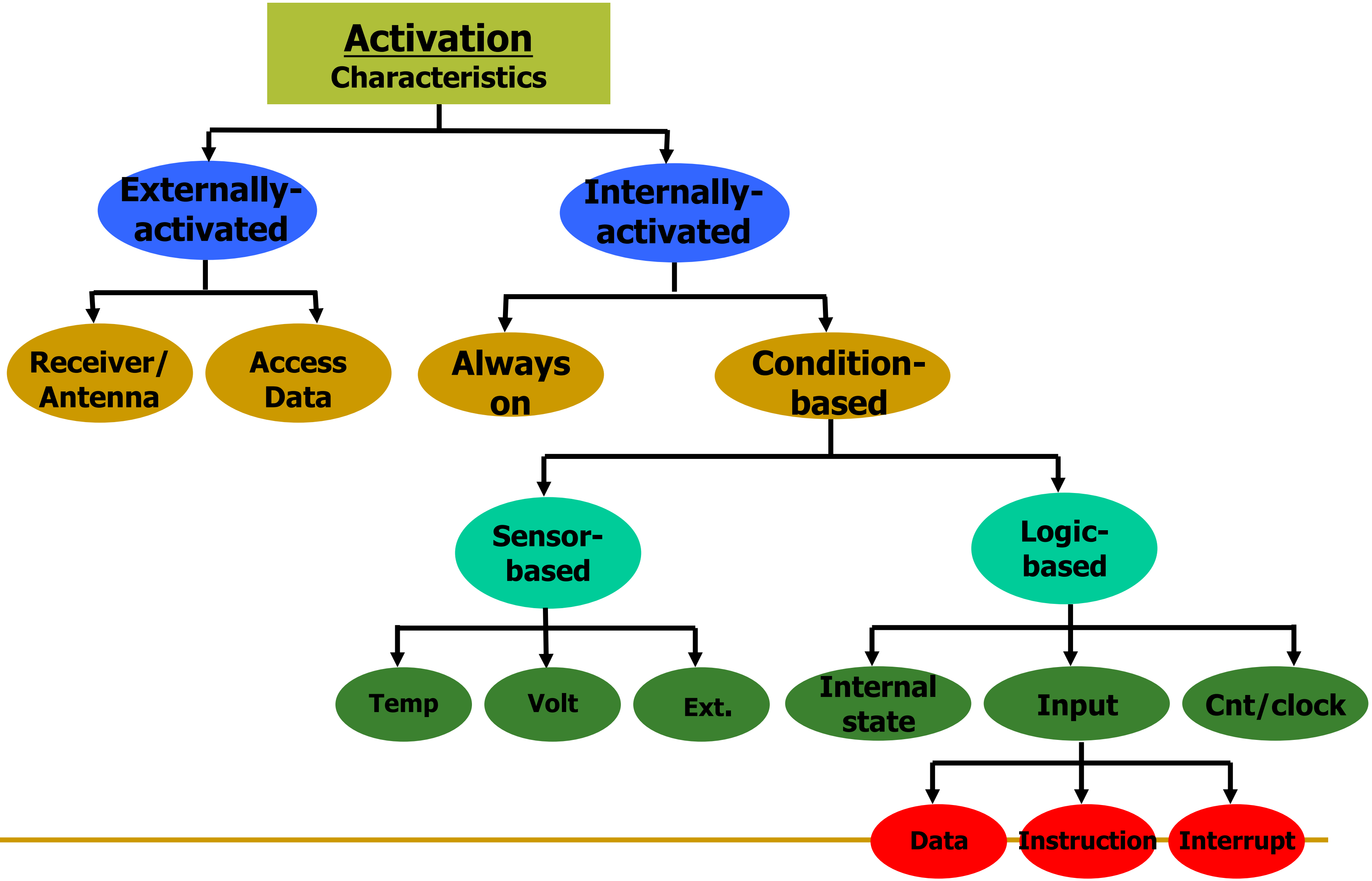
Example: Structure



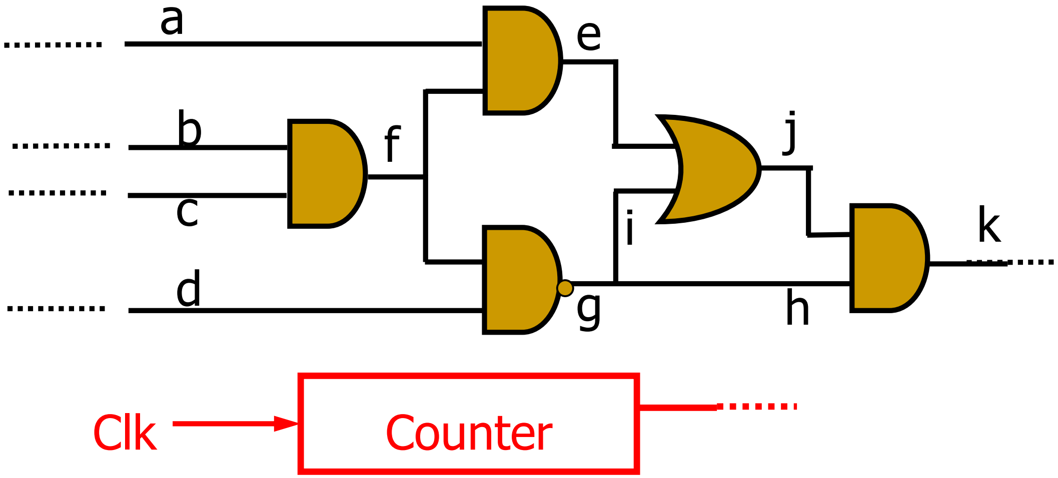


- The adversary may be forced to regenerate the layout to be able to insert the Trojan, then the chip dimensions change
  - It could result in different placement for some or all the design components
- A change in physical layout can change the delay and power characteristics of chip
  - It is easier to detect the Trojan

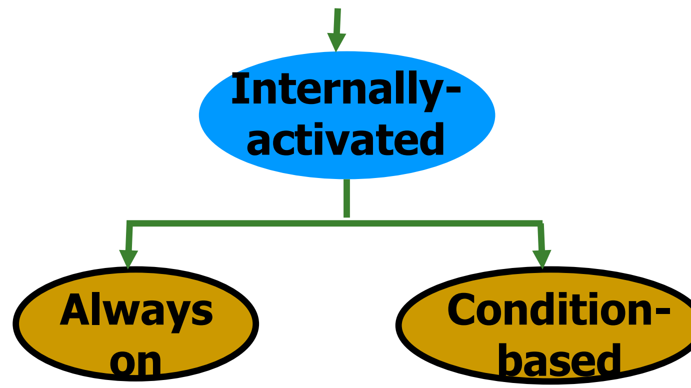
Trojan Taxonomy: Activation

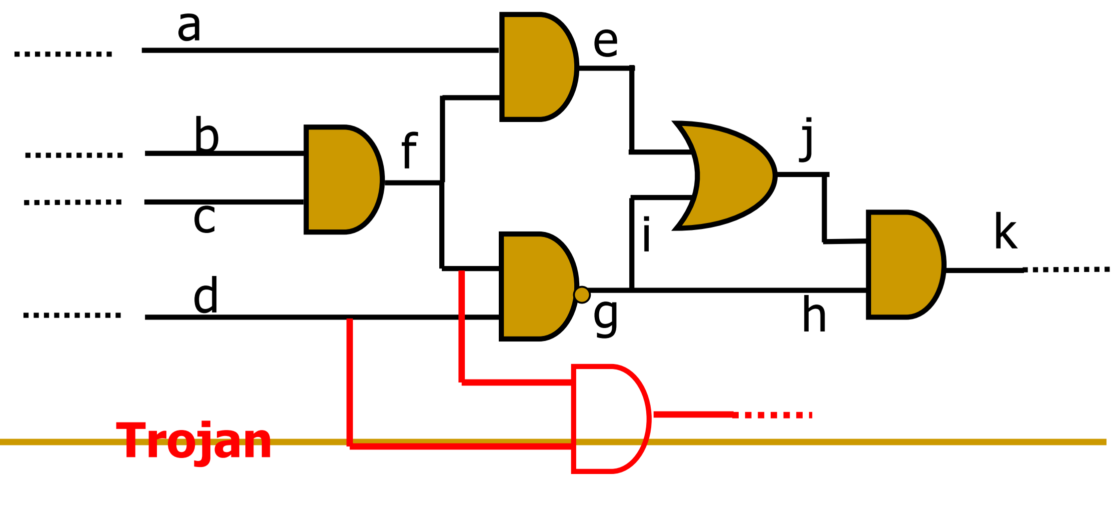


Activation: Internally Activated

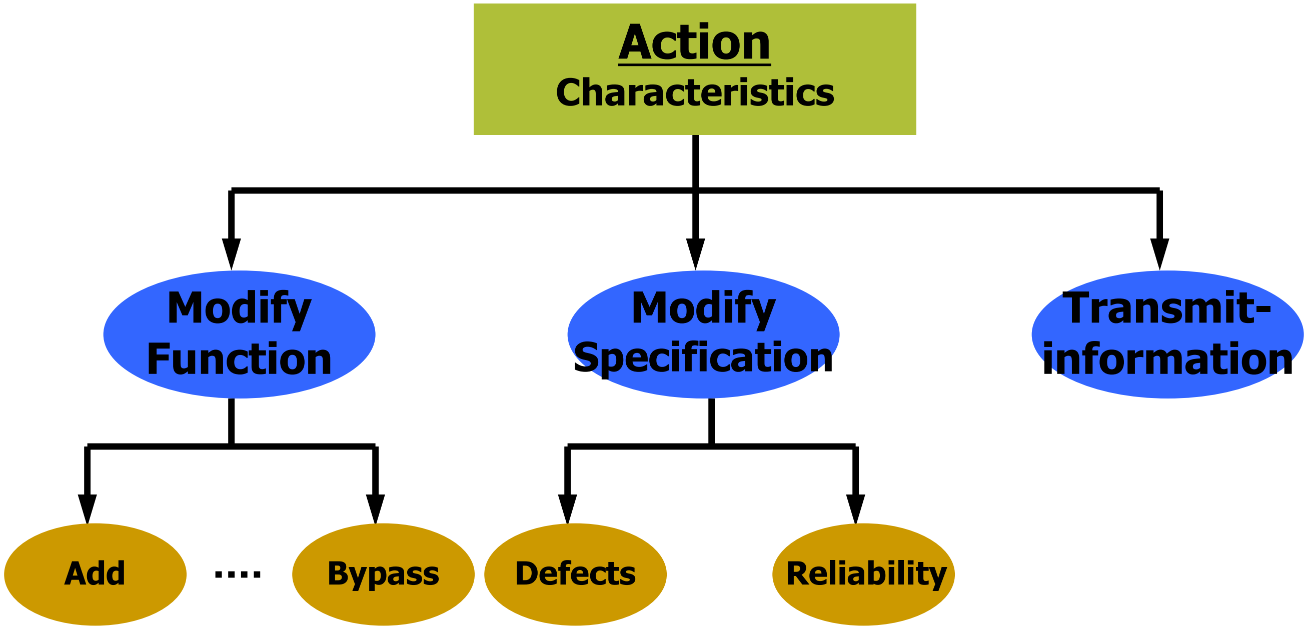


Trojan

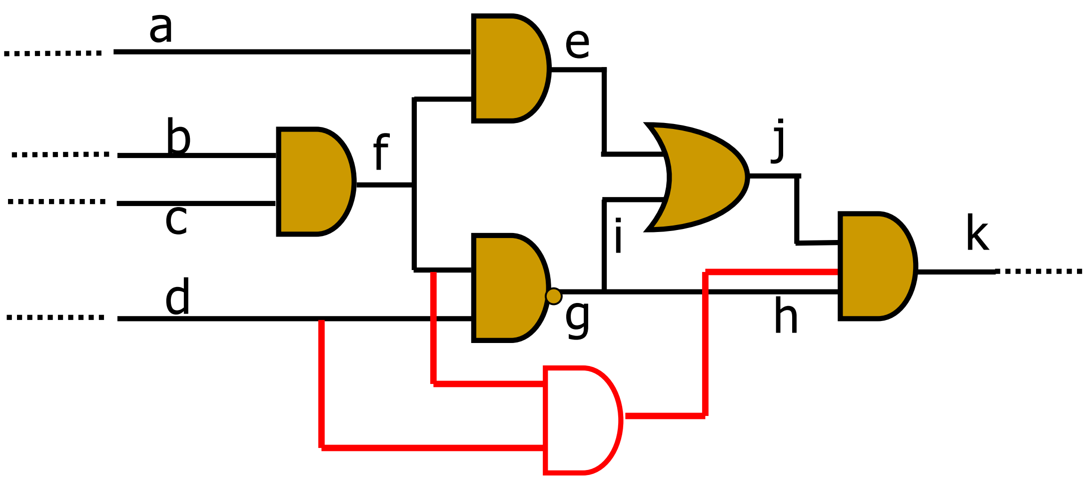




Trojan Taxonomy: Action



Example: Action

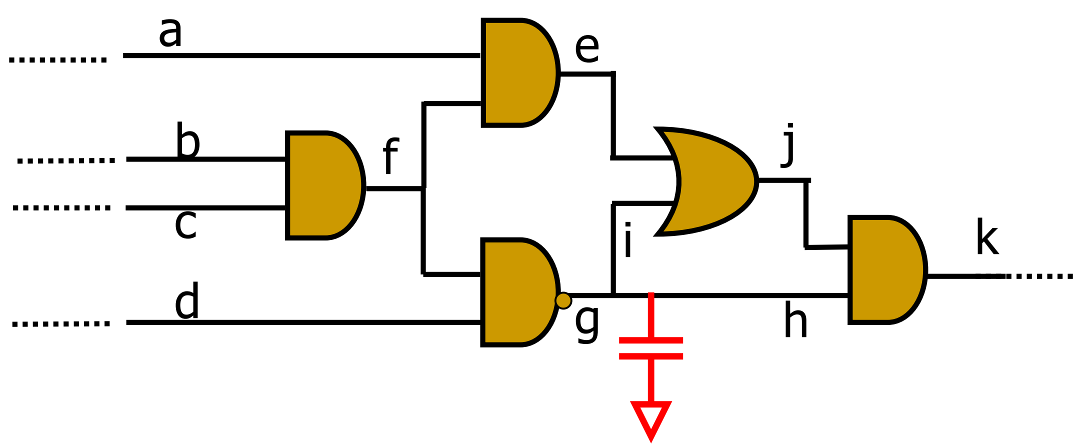


Trojan

Modify Function

Modify Specification:

Noise, Delay and Temperature



IP Trust & IP Security

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

- **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

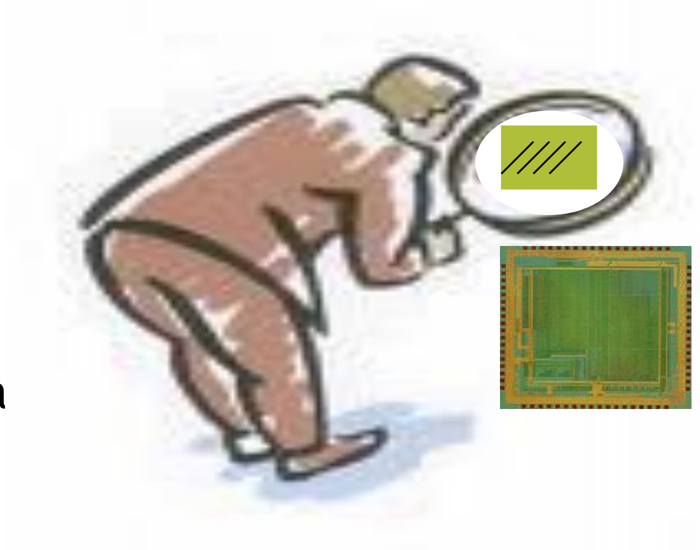
Suspicious Parts

- **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.**

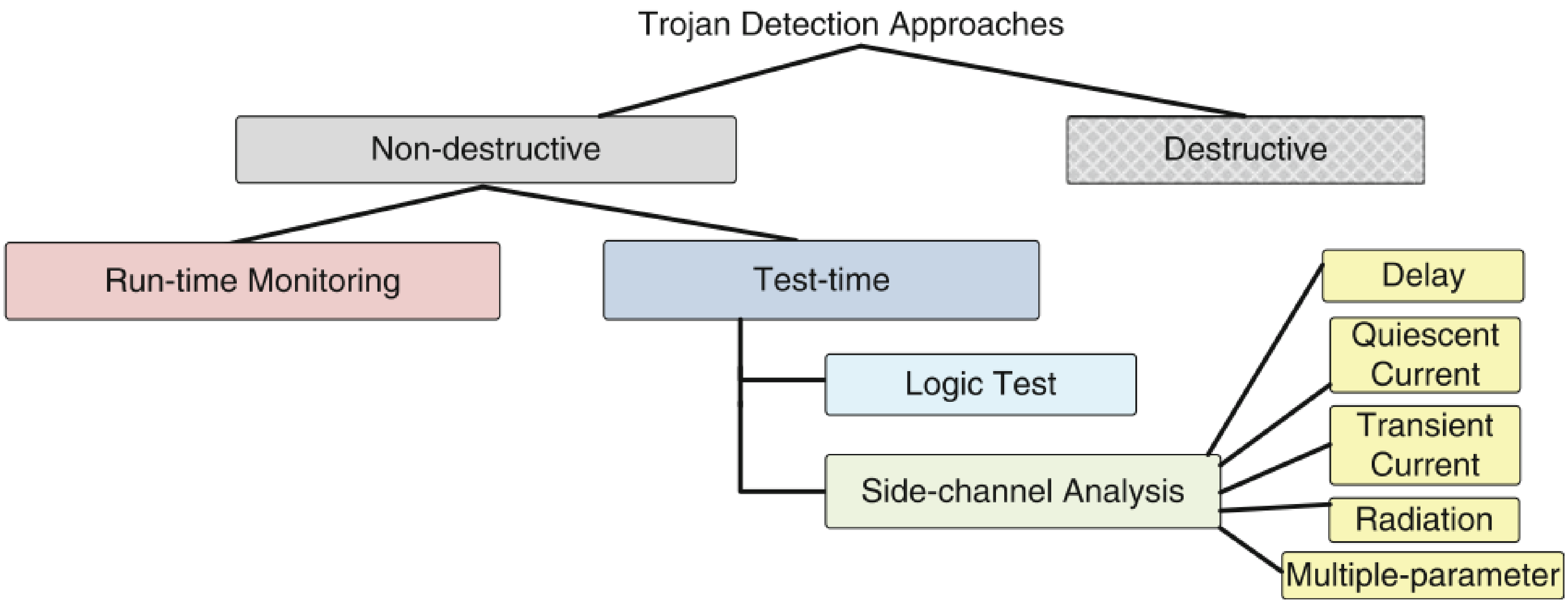
IC Trust

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 additiona functionality in a design



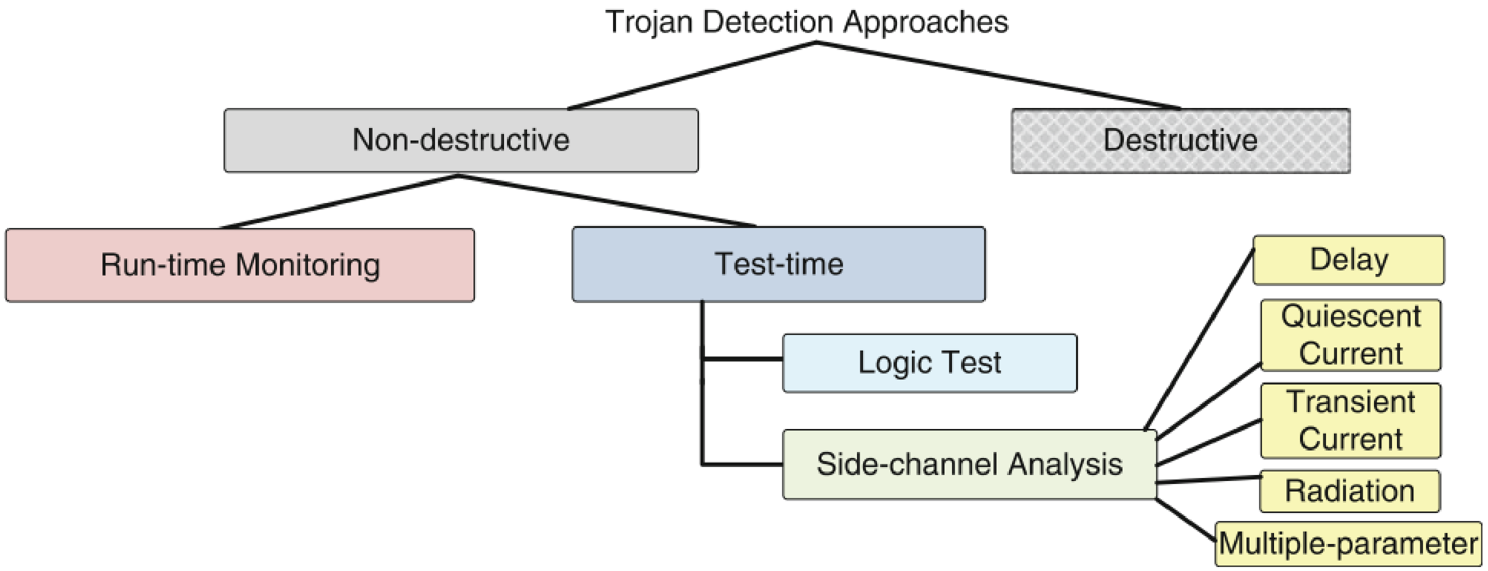
Classification of Trojan Detection Approaches



- **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



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 → 2^{65} input combination (including carry in)
  - $2^{65} > 10^{18}$ – This is impractical
  - 100MHz is used → 10^{10} s → 317 years
  - Only a few and more effective patterns are used → Trojan can escape.
  - The fault coverage is low for manufacturing test

- In practice, structural tests are used.

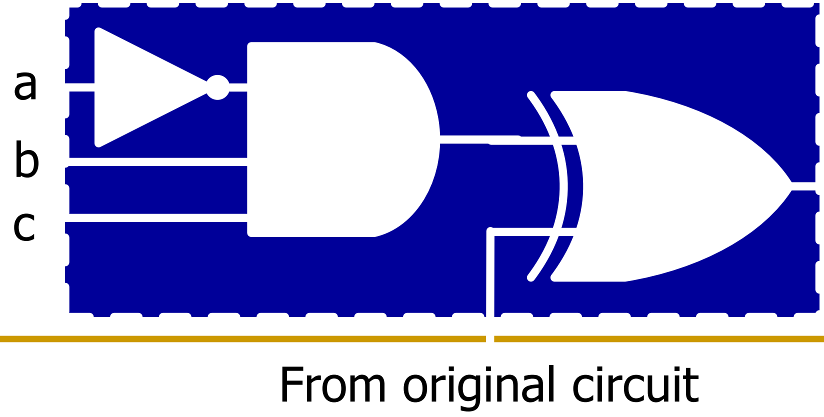
Functional Testing

Feasible Trojan space inordinately large!

<u>Deterministic</u> test generation infeasible

A statistical approach is, more effective

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



Trojan Trigger Condition

a=0, b=1, c=1

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

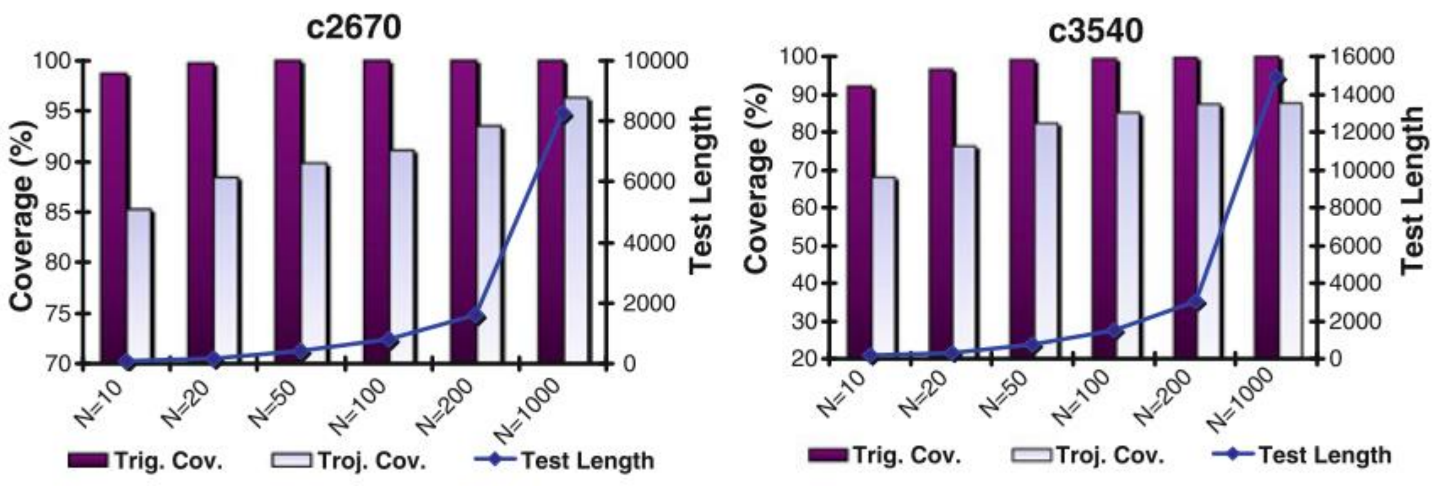


Fig. 15.6 Trigger coverage and Trojan coverage and test length for two ISCAS-85 benchmark circuits for different values of "N," using the MERO approach [8]

- **Challenge:** Triggering each net N times in a large circuit is challenging

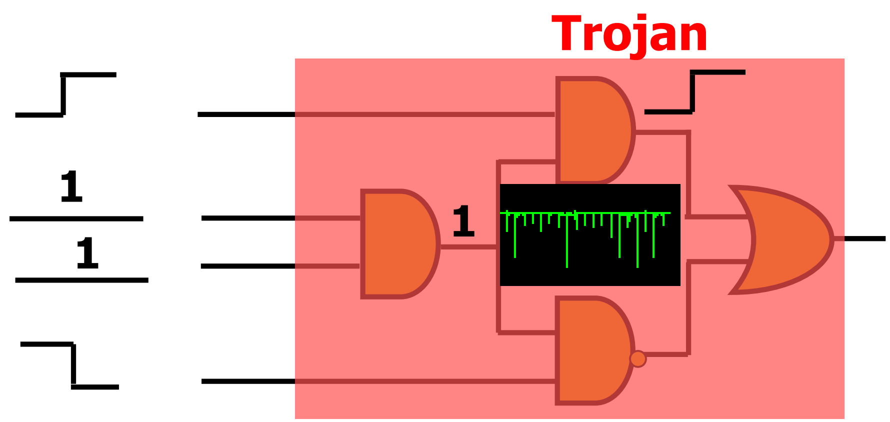
Comprehensive Attack Model

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

Side Channel Signal Analysis -- Power

- 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.

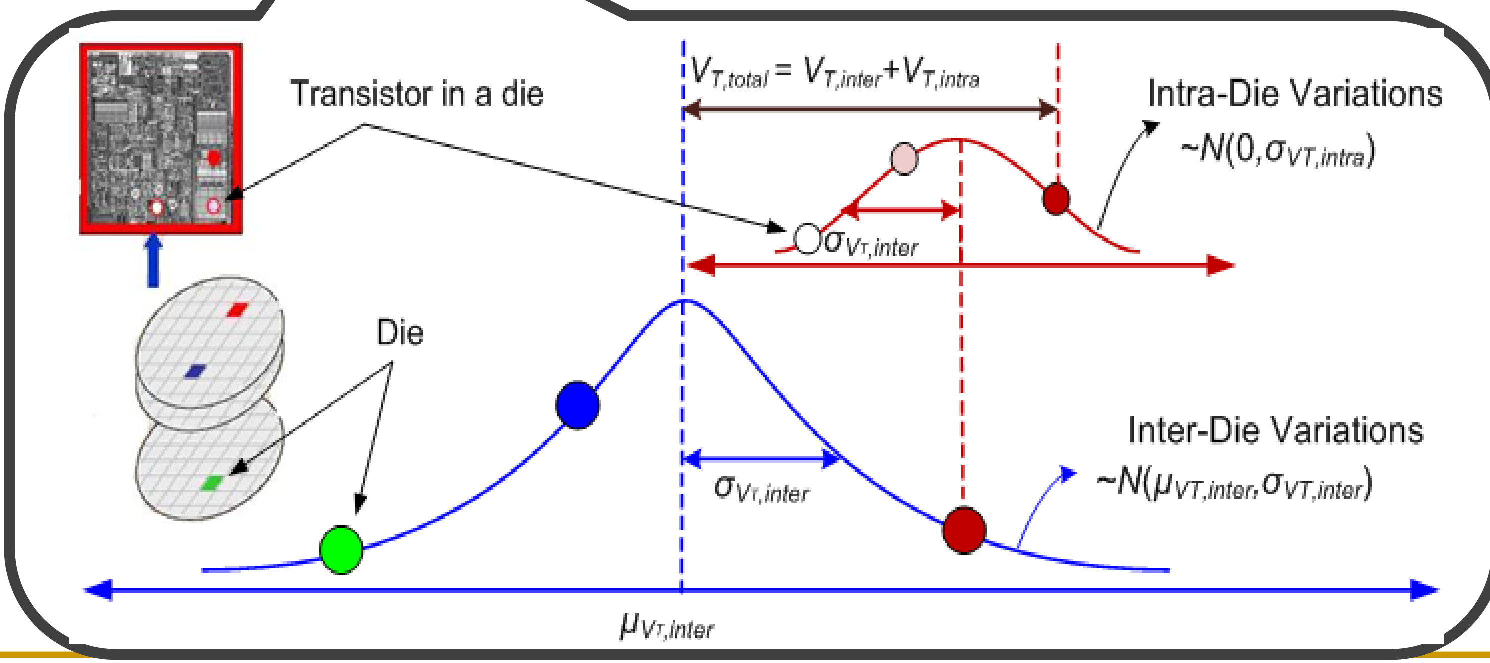
Partial Activation



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_{max} or I_{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**

Trojan Size $\downarrow$ $\Rightarrow$ Sensitivity $\downarrow$

Circuit Size $\uparrow$ $\Rightarrow$ Sensitivity $\downarrow$

$$Sensitivity = \frac{I_{tampered} - I_{original}}{I_{original}} \times 100\%$$

Comparing Approaches

	Logic Testing	Side-Channel Analysis
Pros	• Robust under process noise • Effective for ultra-small Trojans	• Effective for large Trojans • Easy to generate test vectors
Cons	• Difficult to generate test vectors • Large Trojan detection challenging	• Vulnerable to process noise • Ultra-small Trojan Det. challenging

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

Question?