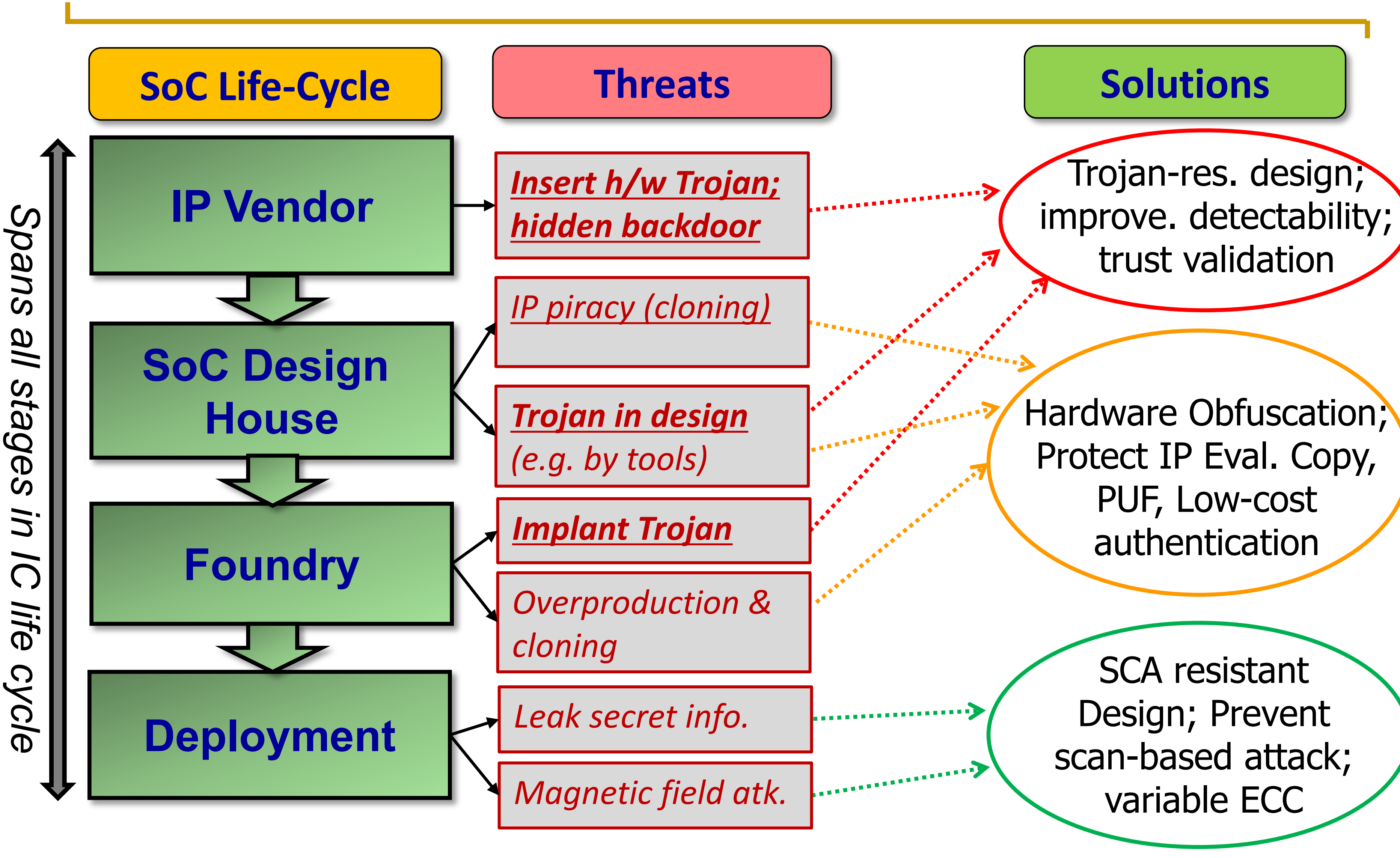
Hardware Trojan

Mark Tehranipoor

Introduction to Hardware Security & Trust
University of Florida

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

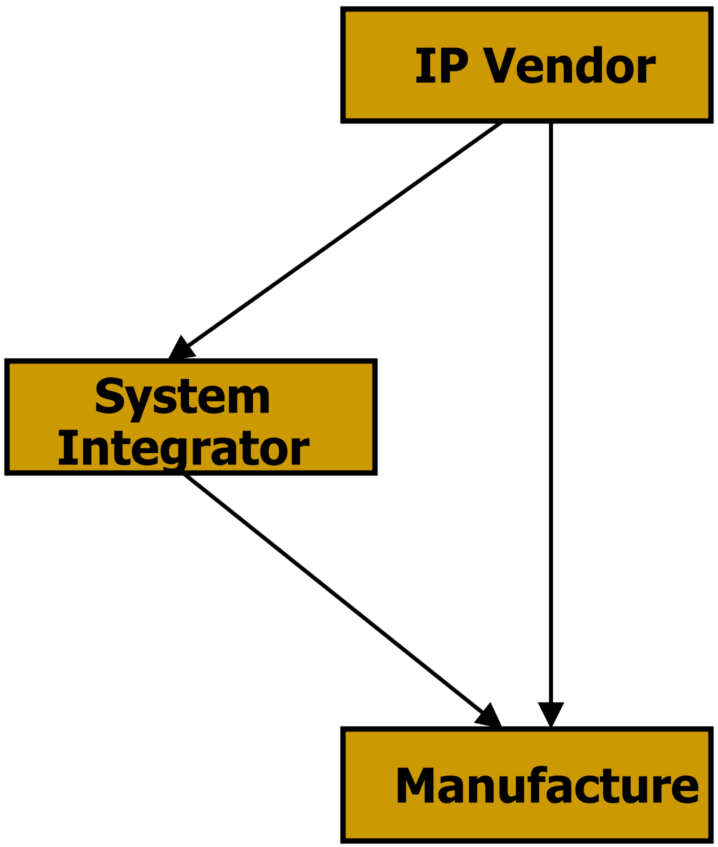
IC/IP Trust Problem

- 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*

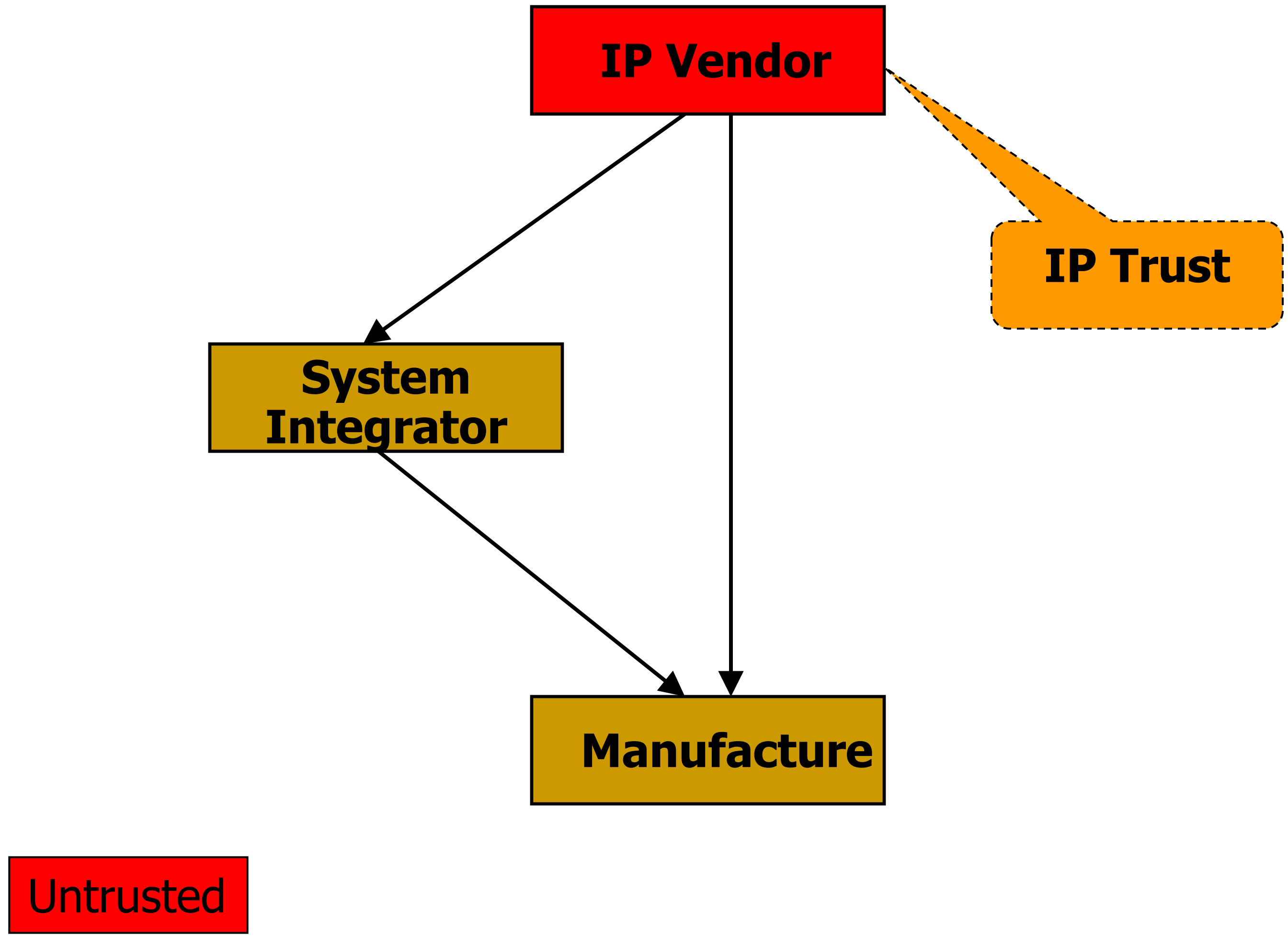
- **Designer and Foundry:**
  - Foundry may place a Trojan in the layout design.
  - *IC Trust problem*

Hardware Trojan Threat

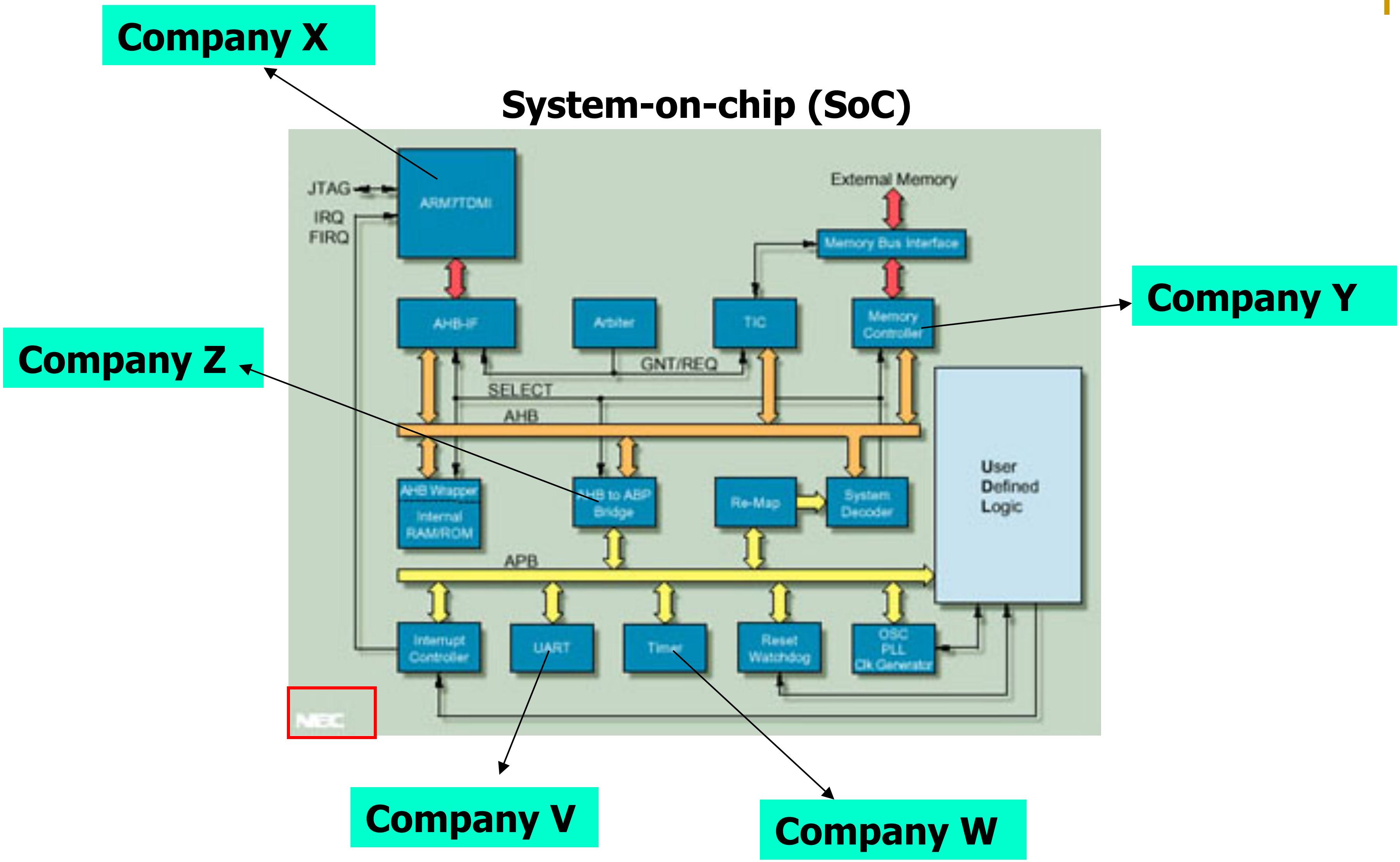


Any of these steps can be untrusted

Hardware Trojan Threat



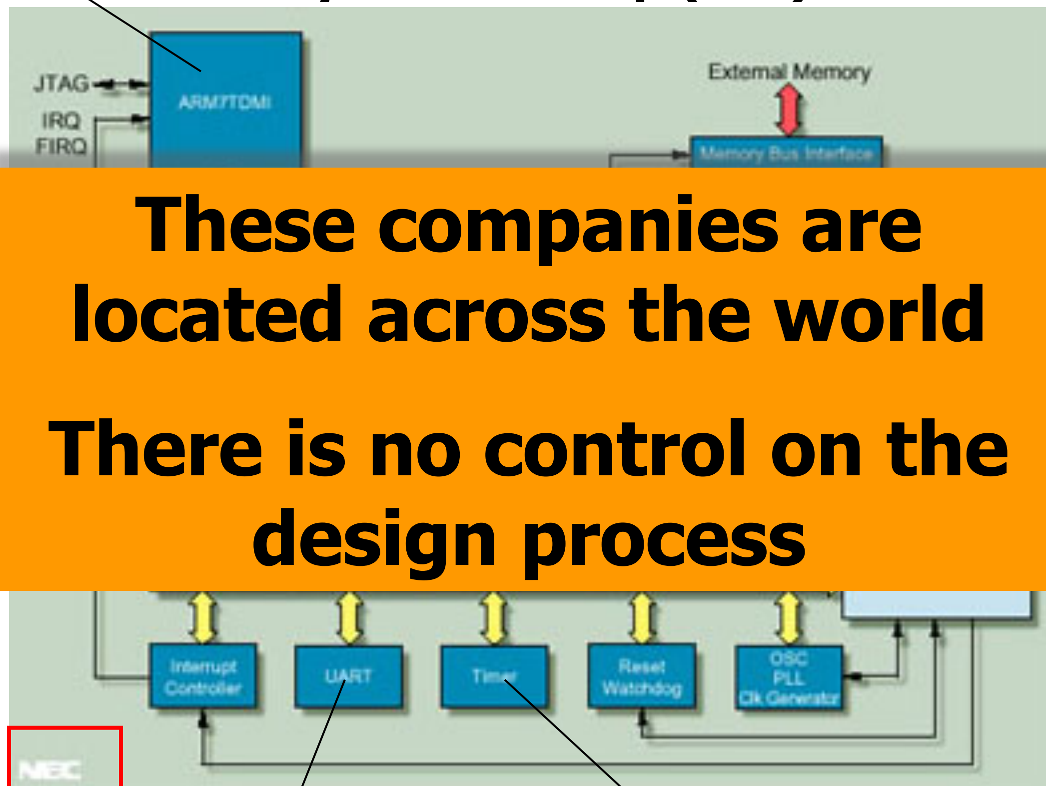
Issues with Third IP Design



Issues with Third IP Design

Company X

System-on-chip (SoC)



Company Y

Company Z

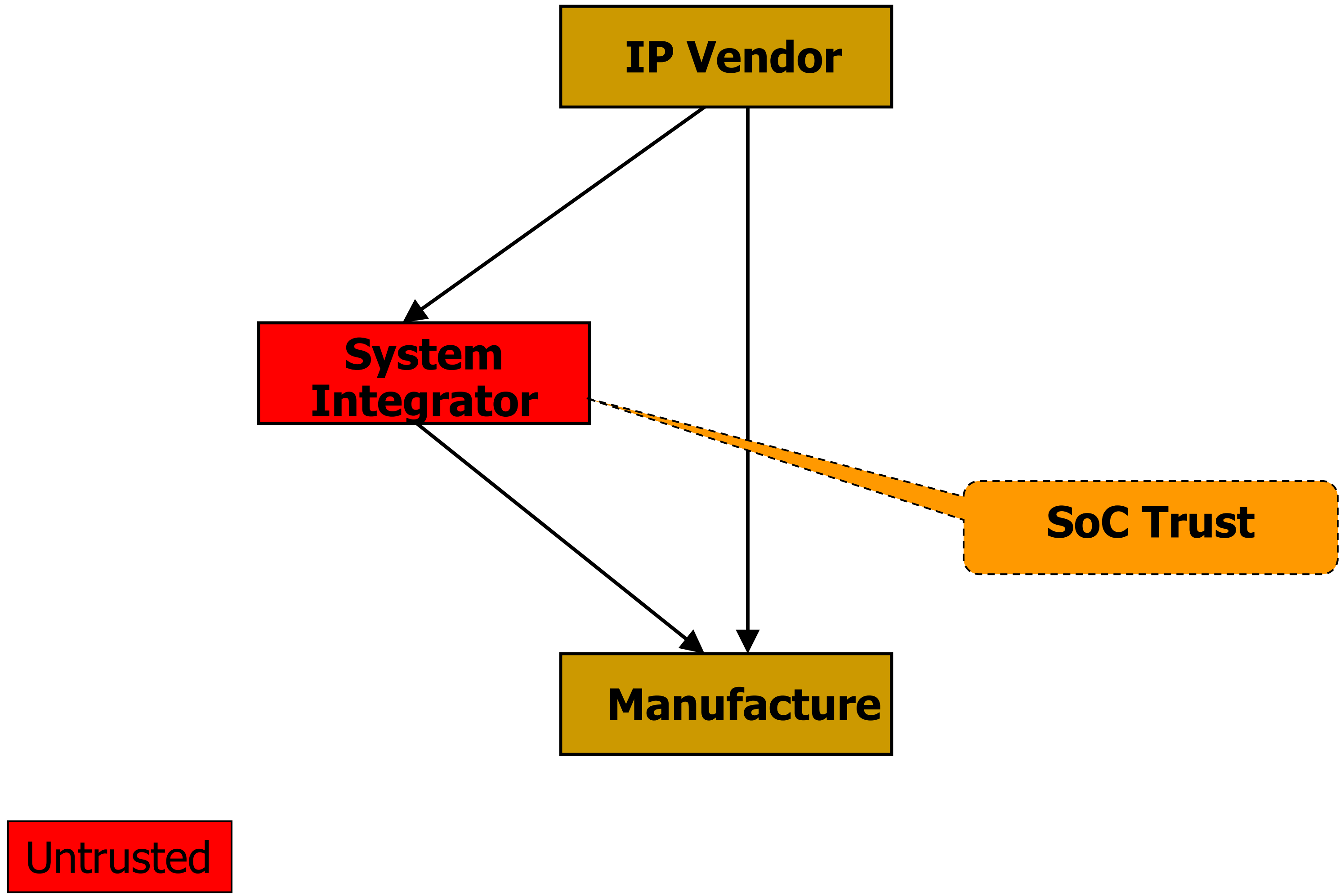
These companies are located across the world

There is no control on the design process

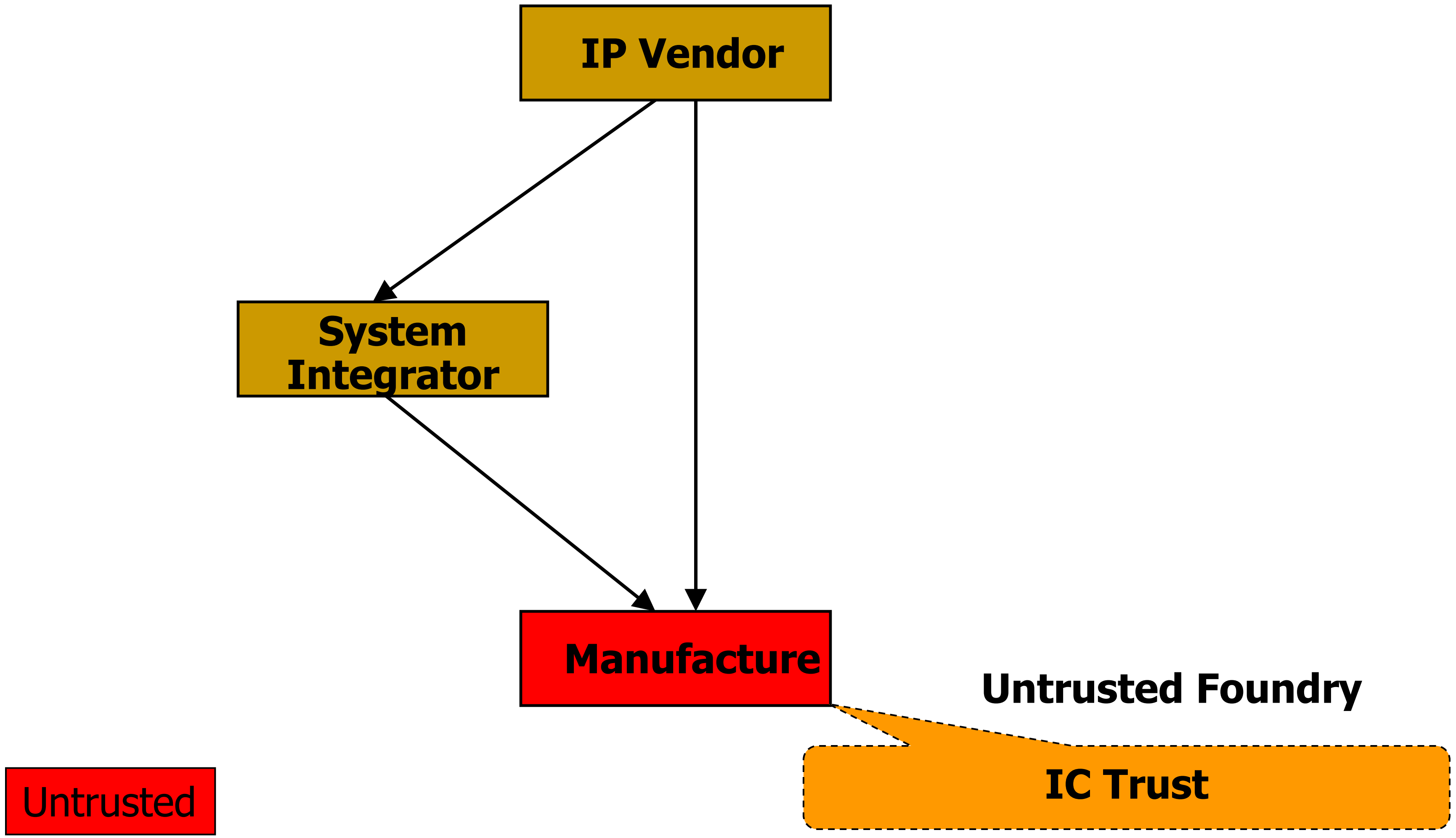
Company V

Company W

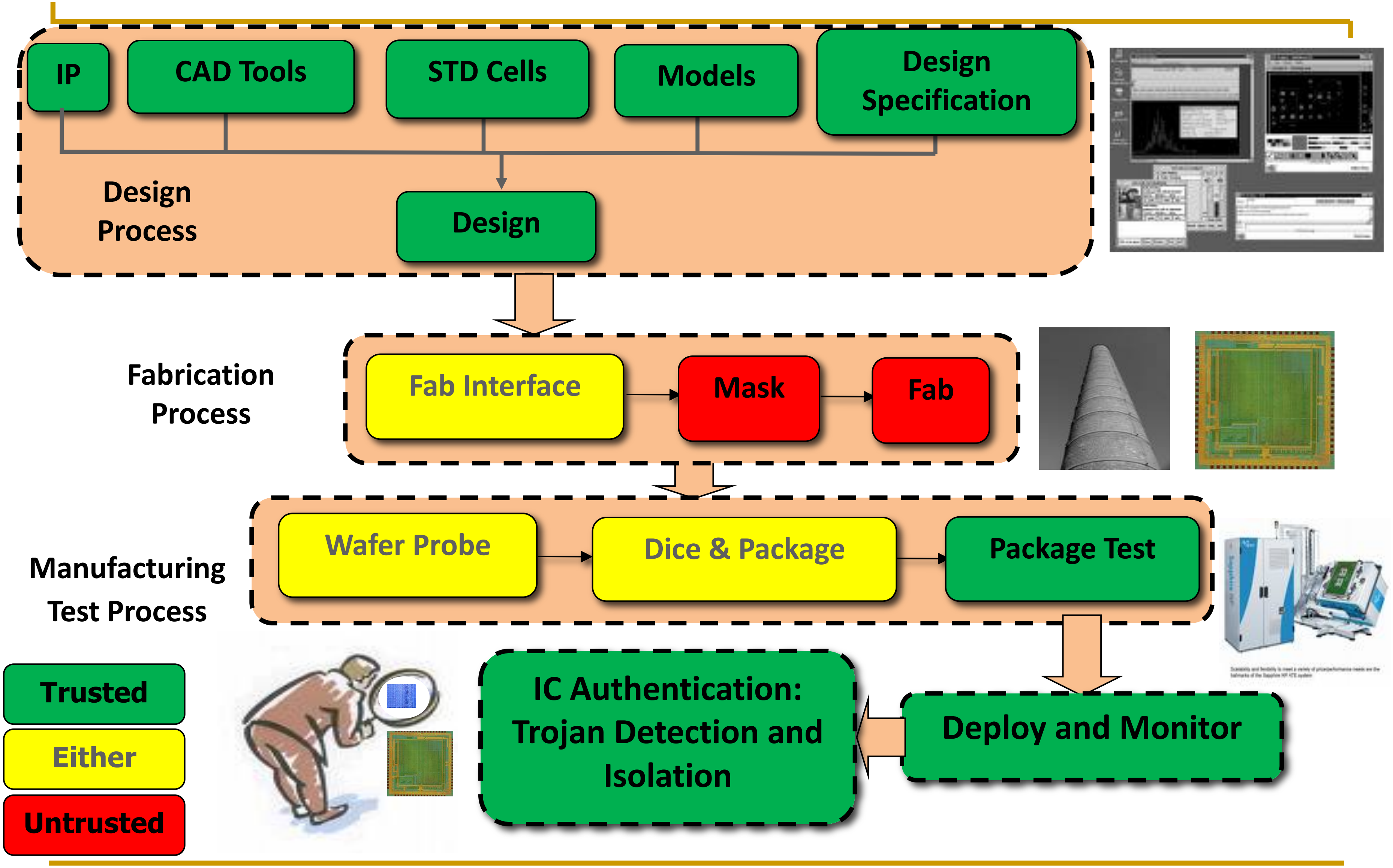
Hardware Trojan Threat



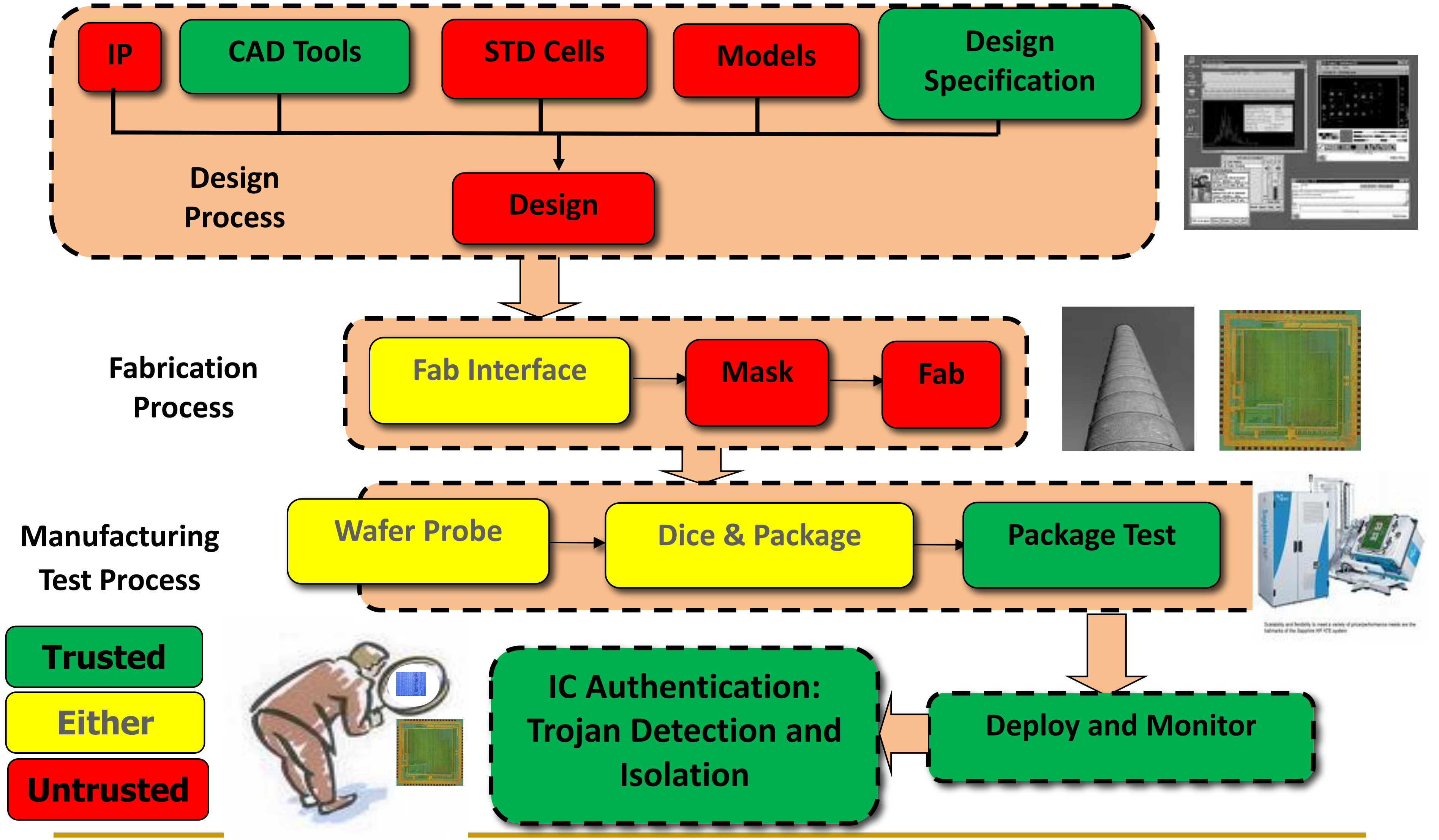
Hardware Trojan Threat



ASIC Design Process – Untrusted Foundry

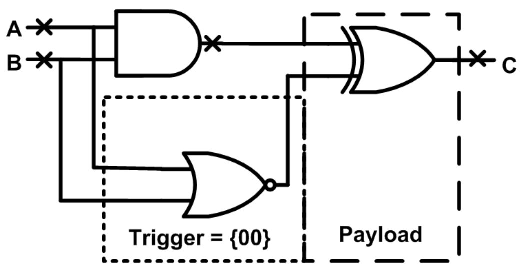


Untrusted Designer and Foundry

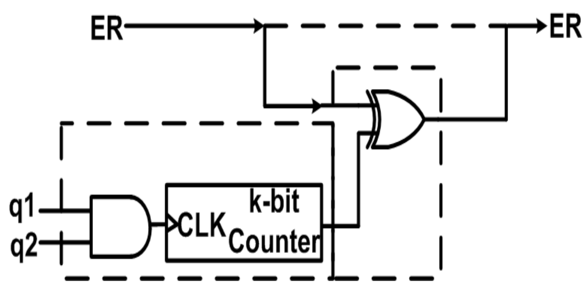


HW Trojan Examples / Models

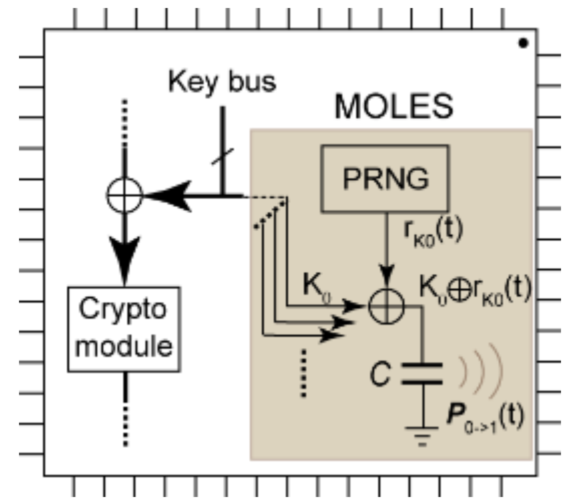
Comb. Trojan Example



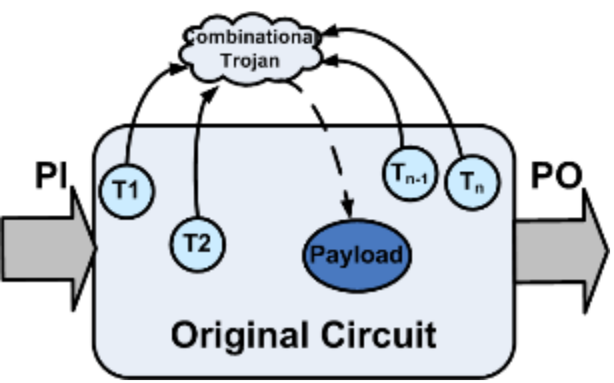
Seq. Trojan Example



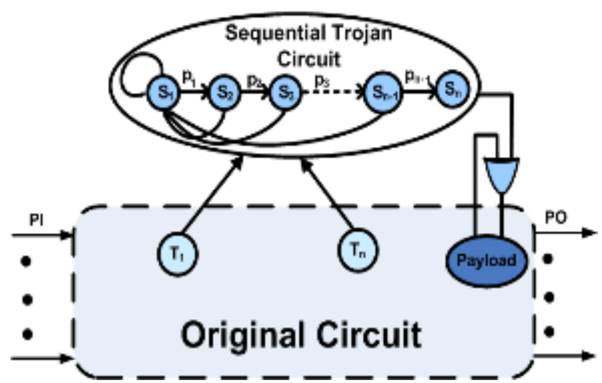
MOLES: Info Leakage Trojan*



Comb. Trojan model



Seq. Trojan Model



**Lin et al, ICCAD 2009*


Fishy Chips: Spies Want to Hack-Proof Circuits

By Adam Rawnsley

In 2010, the U.S. military had a problem. It had bought over 59,000 microchips destined for installation in everything from missile defense systems to gadgets that tell friend from foe. The chips turned out to be counterfeits from China, but it could have been even worse. Instead of crappy Chinese fakes being put into Navy weapons systems, the chips could have been hacked, able to shut off a missile in the event of war or lie around just waiting to malfunction.

HW Trojan evidence!

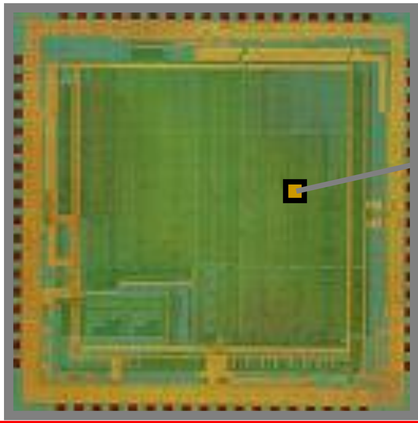
Why is detection of hardware Trojans very difficult?

Bug vs. Malicious Change

- **Verification (Traditional)**
  - Bugs (Unintentional)
  - Bounded by Spec
- **Trust Verification**
  - Malicious change (Intentional)
  - Unwanted functionality (Unbounded)

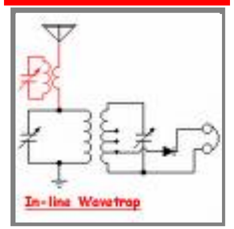
Trojan Attacks → BIGGER verification challenge!

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)

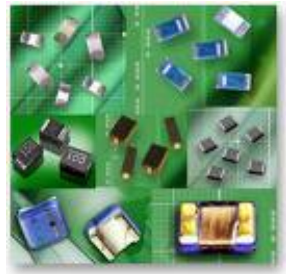
Comparator to monitor key data

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

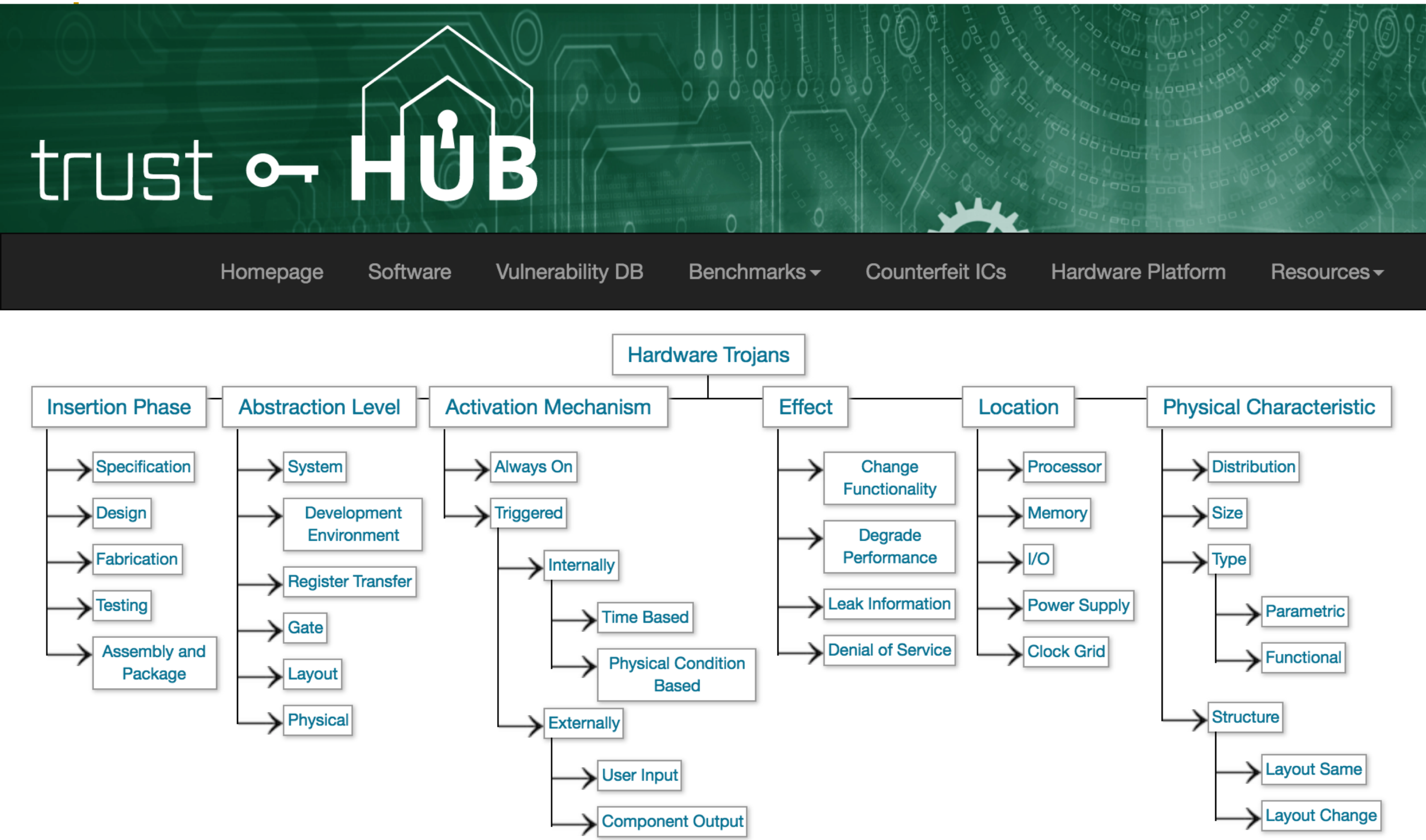
Thousands of chips are being fabricated in untrusted foundries



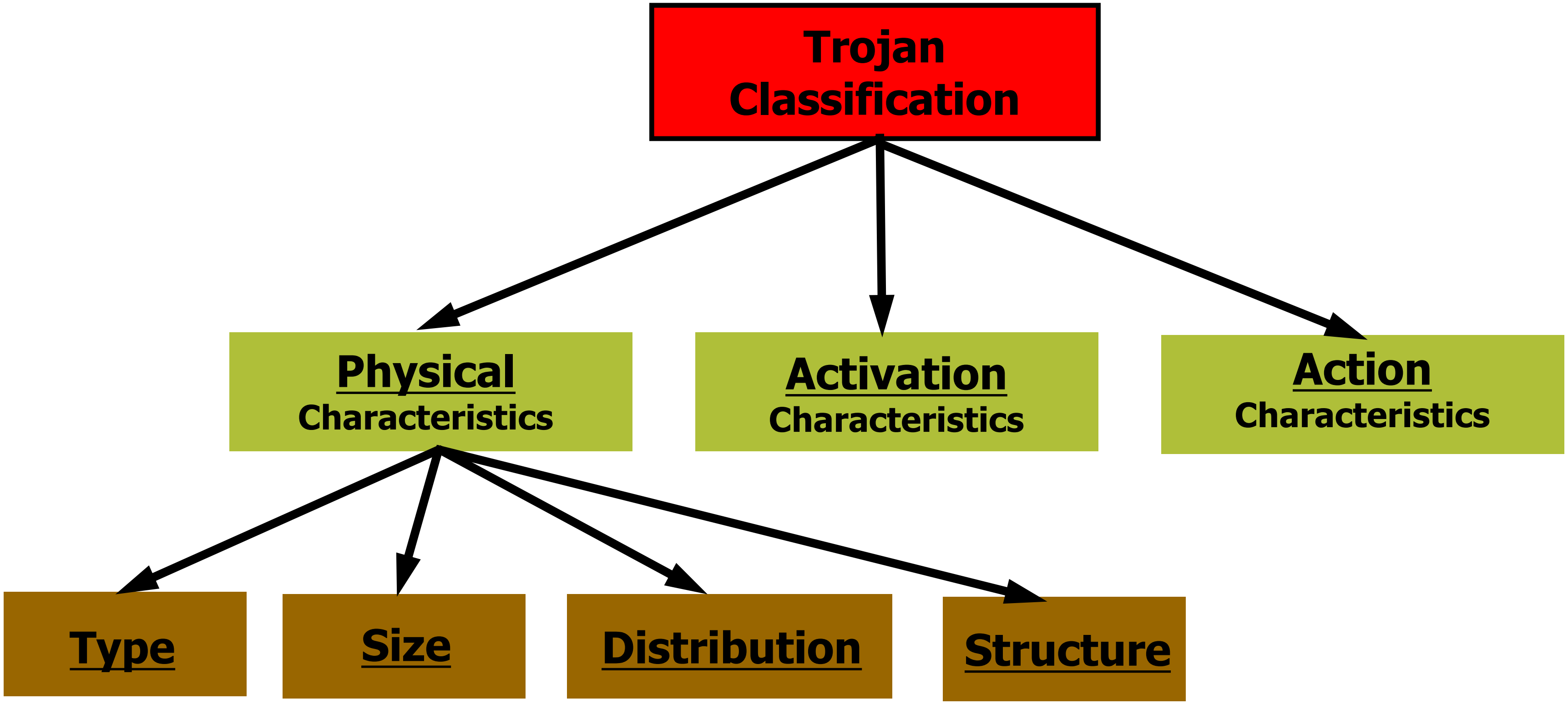
Comprehensive Attack Model

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

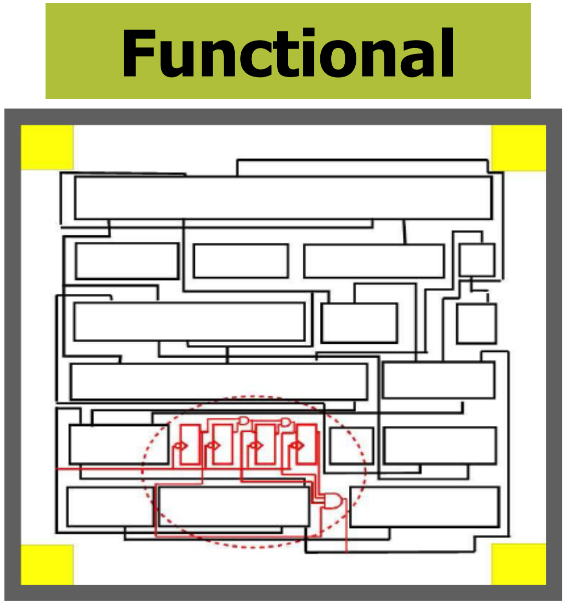


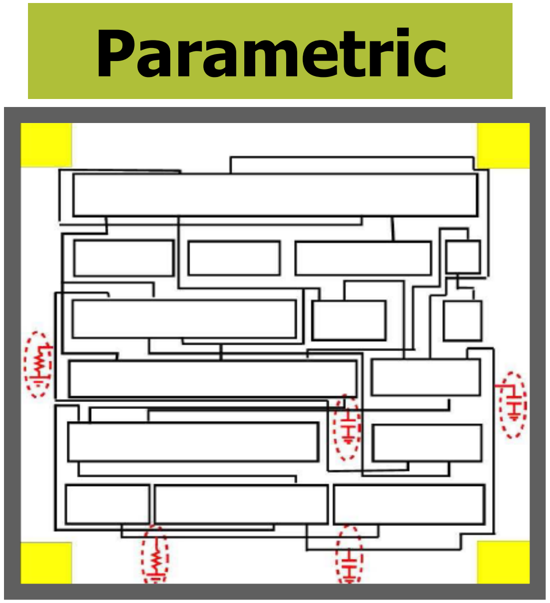
Trojan Taxonomy



Examples for Layout Level Trojans

Example: Type

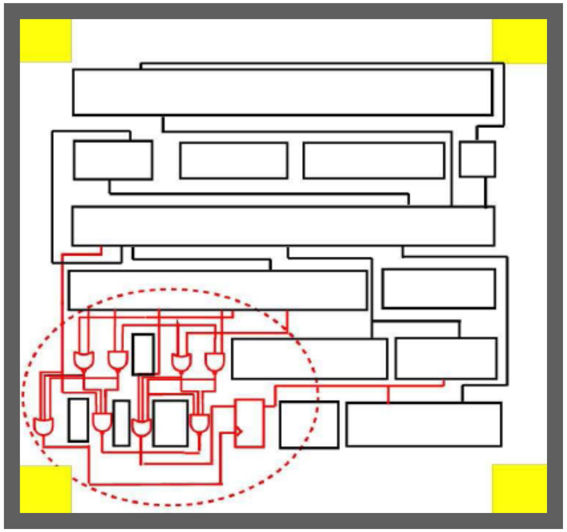




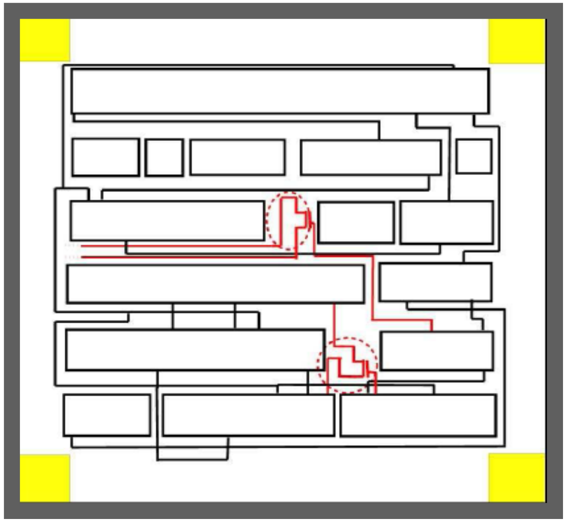
- **Functional**
  - **Addition or deletion of components**
  - **Sequential circuits**
  - **Combinational circuits**
  - **Modification to function or no change**
- **Parametric**
  - **Modifications of existing components**
    - **Wire: e.g. thinning of wires**
    - **Logic: Weakening of a transistor,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



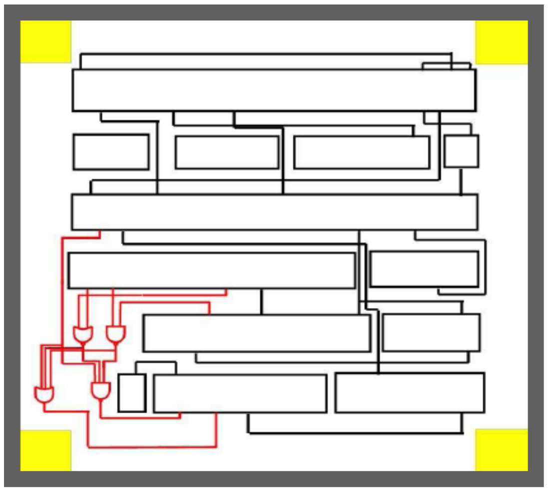
Small



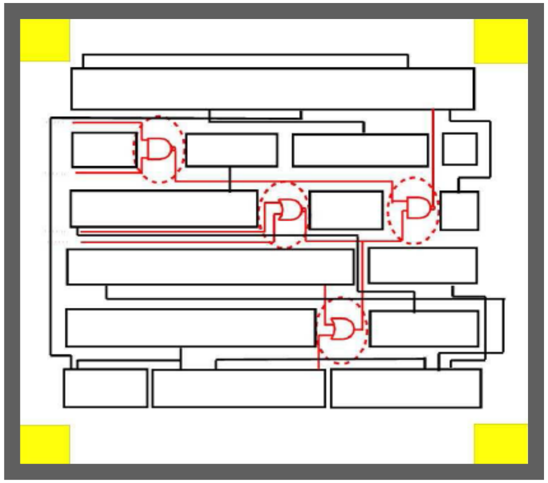
- **Size:**
  - **Number of components added to the circuit**
    - **Small transistors**
    - **Small gates**
    - **Large gates**
- **In case of layout, depends on availability of:**
  - **Dead spaces**
  - **Filler cells**
  - **Decap cells**
  - **Change in the structure**

Example: Distribution

Tight



Loose

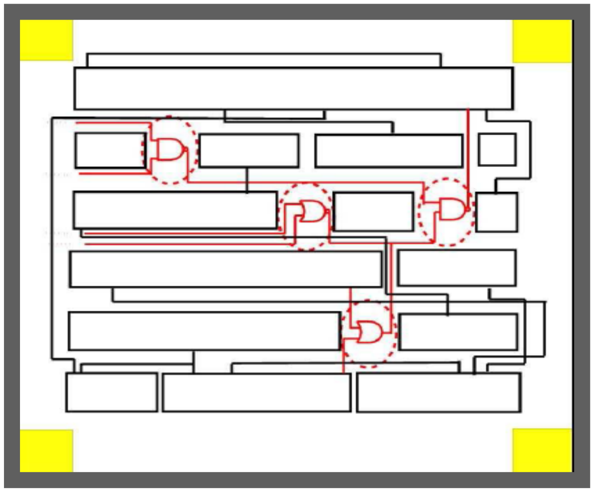


- **Tight Distribution**
  - **Trojan components are topologically close in the layout**
- **Loose Distribution**
  - **Trojan components are dispersed across the layout of a chip**

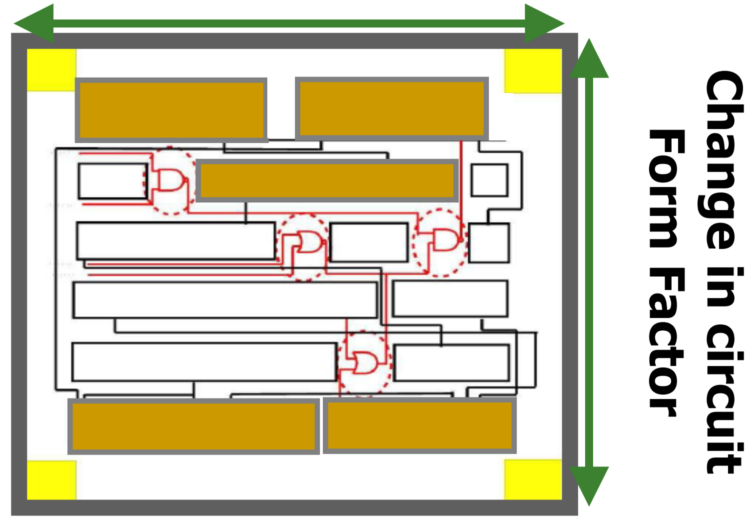
- **Distribution of Trojans depends on the availability of dead spaces on the layout**

Example: Structure

No-change



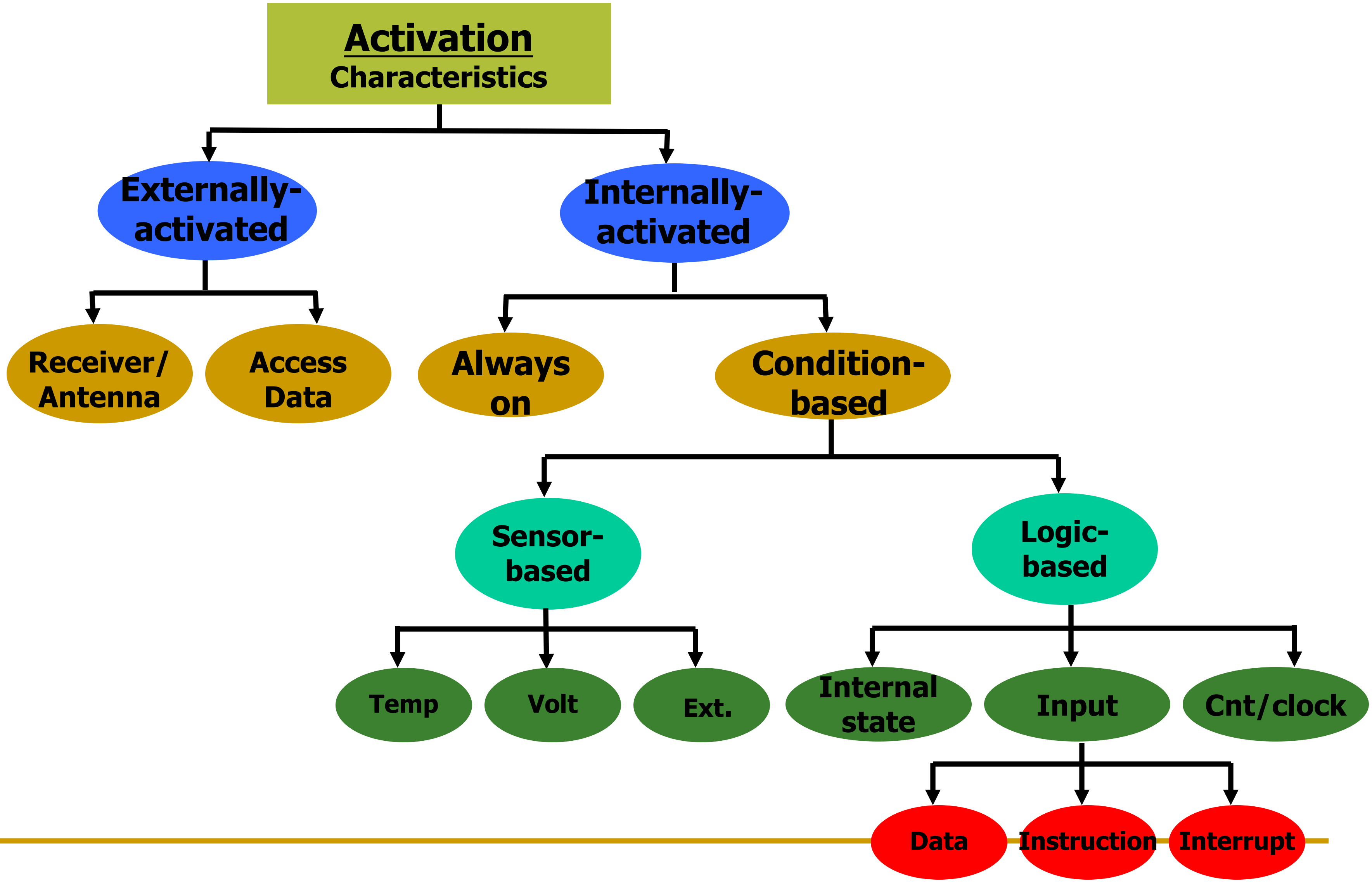
Modified Layout



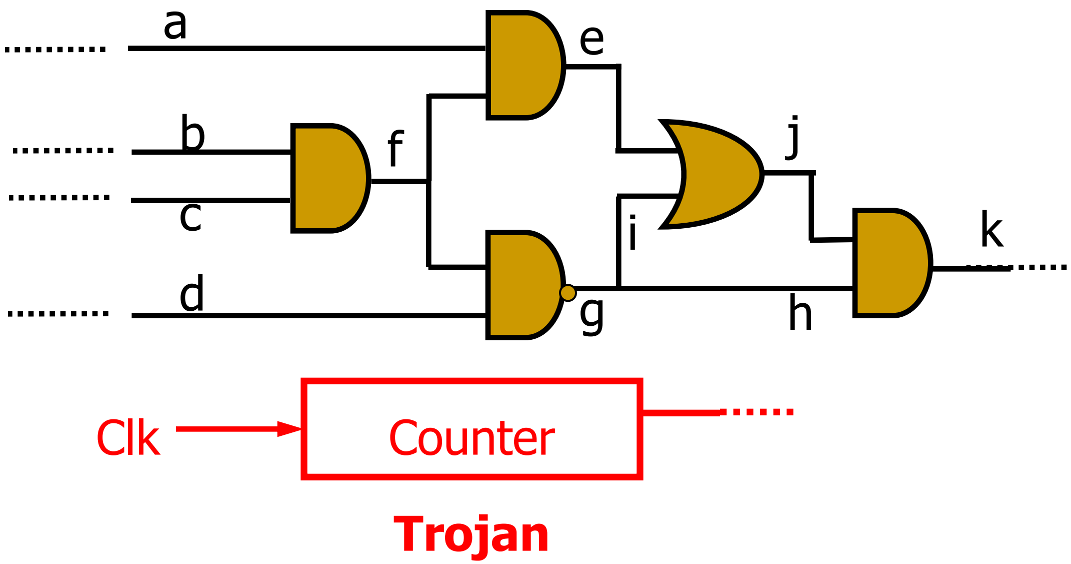
- **The adversary may be forced to regenerate the layout to be able to insert the Trojan, then the chip dimensions change**
  - **It could result in different placement for some or all the design components**

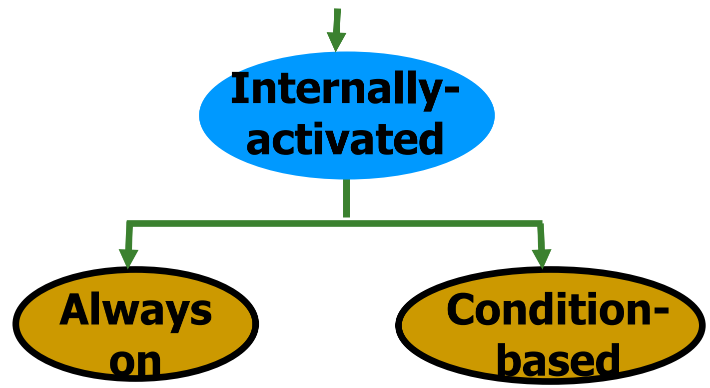
- **A change in physical layout can change the delay and power characteristics of chip**
  - **It is easier to detect the Trojan**

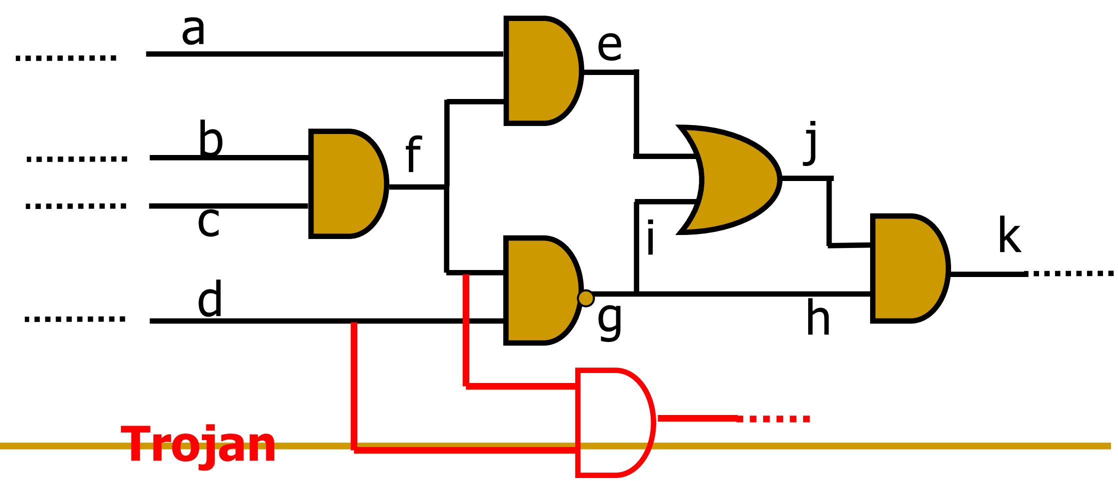
Trojan Taxonomy: Activation



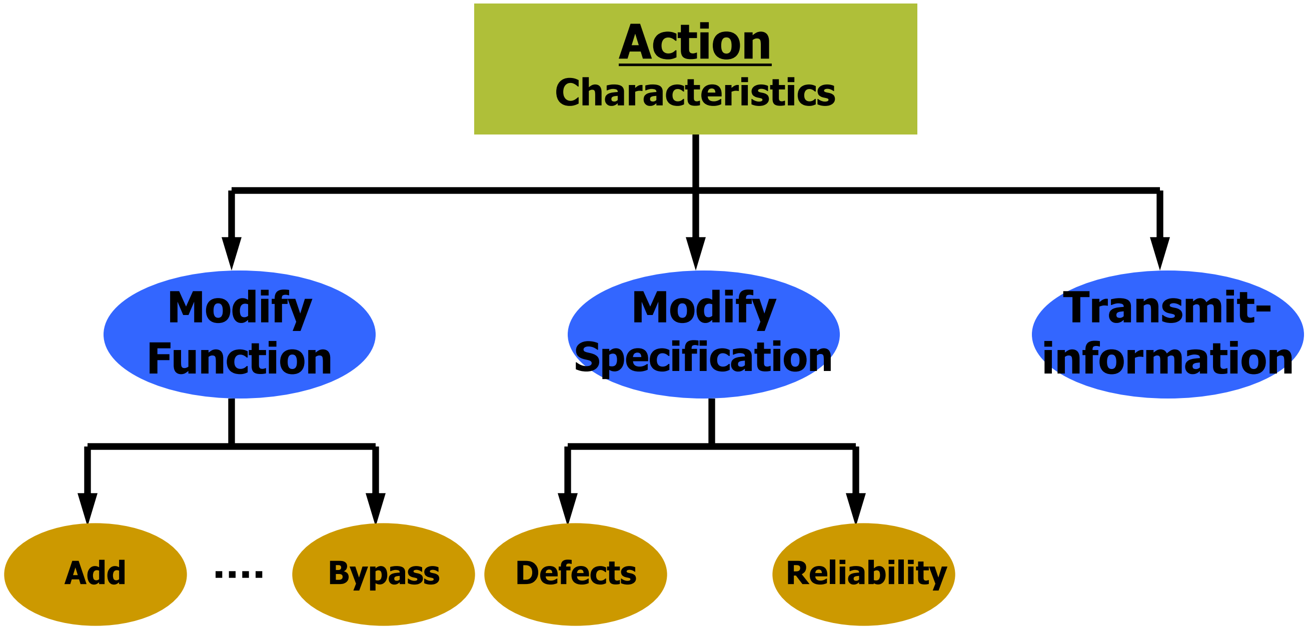
Activation: Internally Activated



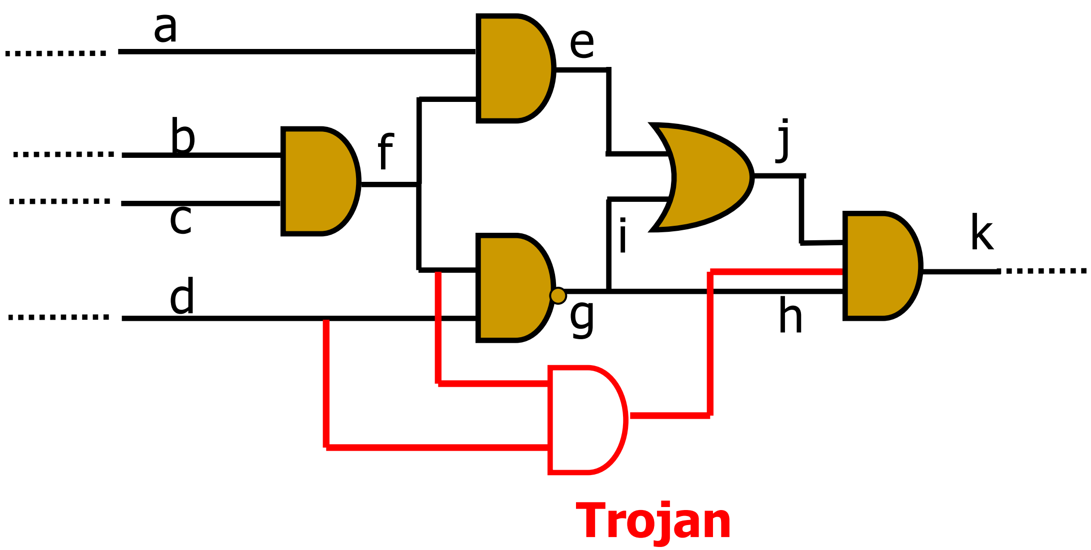




Trojan Taxonomy: Action



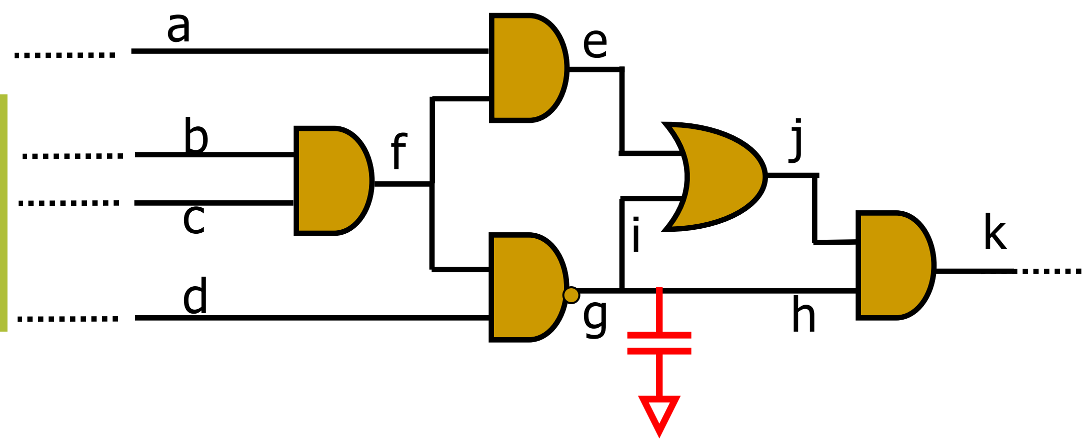
Example: Action



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
  - Focus of this lecture

- **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

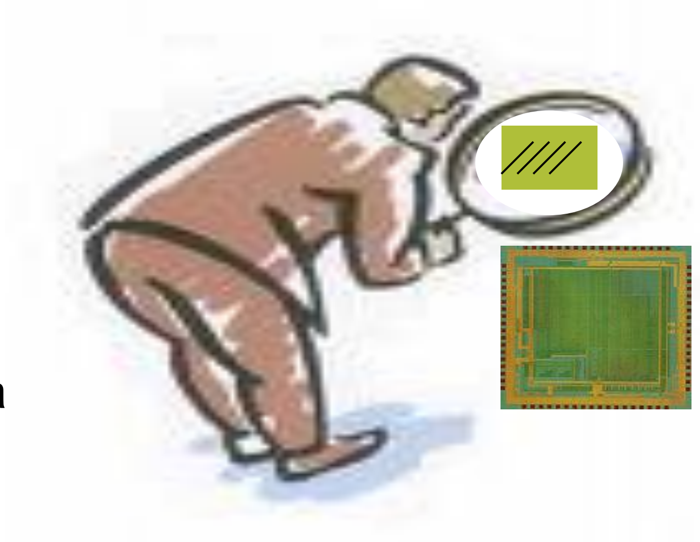
- **If one of the assertions fails, the IP is assumed untrusted.**

- **If coverage is not 100%, *uncovered* parts of the code (RTL, netlist) are assumed suspicious.**

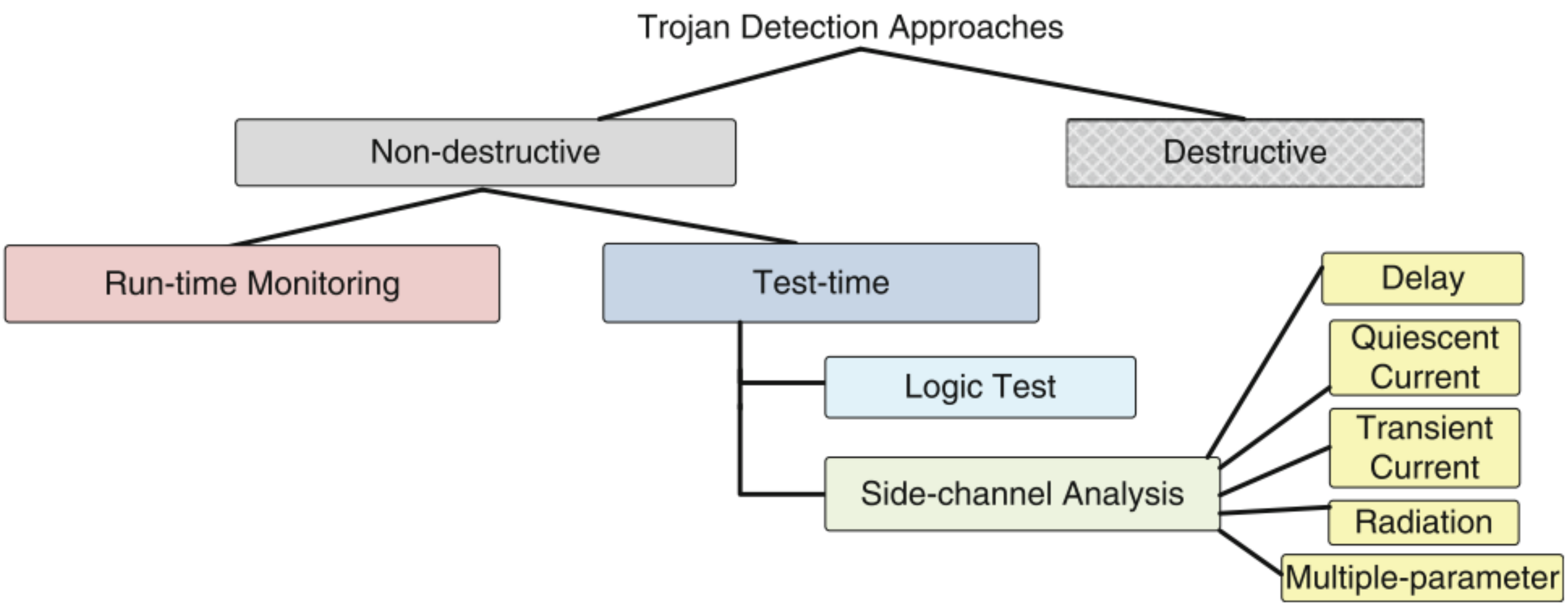
IC Trust

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



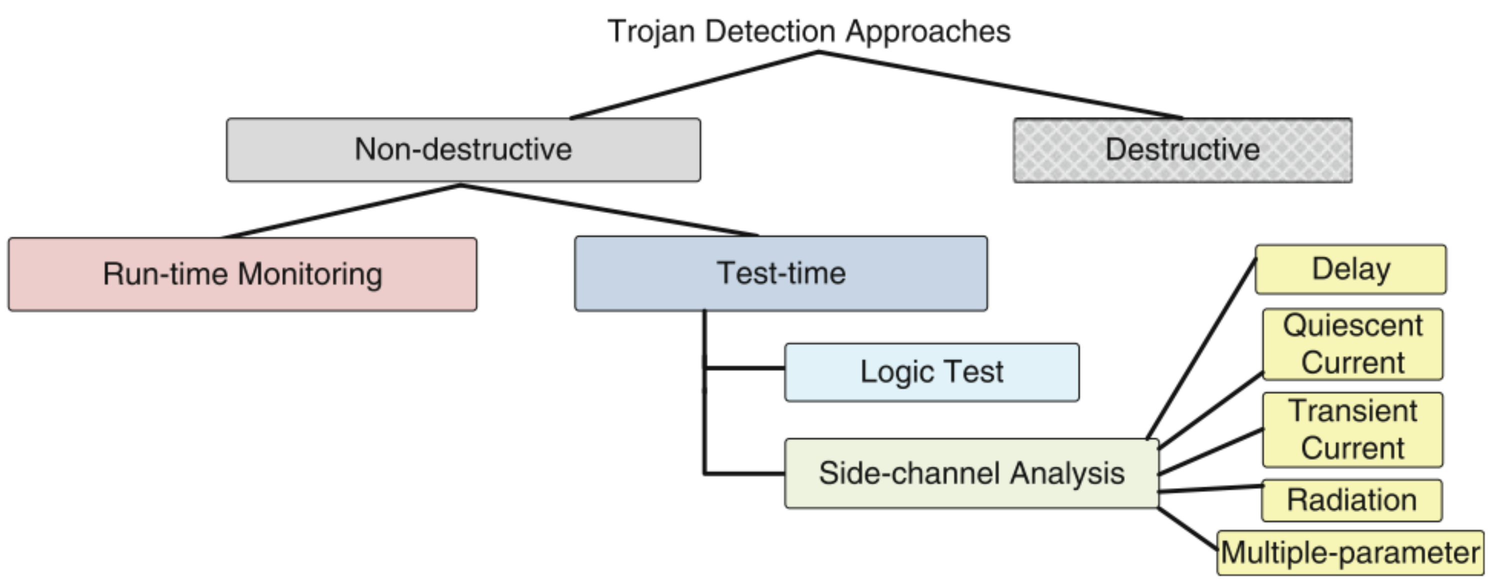
Classification of Trojan Detection Approaches



- **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



Hardware Trojan Benchmarks

- A set of **trust benchmarks** for researchers in academia, industry, and government is needed to
  - Provide a baseline for examining diverse methods developed
  - Establishing a sound basis for the hardness of each benchmark instance
  - Help increase reproducibility of results by others who intend to employ certain methodologies in their design flow
- See NSF supported **Trust-Hub** website (www.trust-hub.org)
  - Complete taxonomy of Trojans
  - More than 120 trust benchmarks available which were designed at different abstraction levels, triggered in several ways, and have different effect mechanisms
  - More than 300 publications used these benchmarks

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.

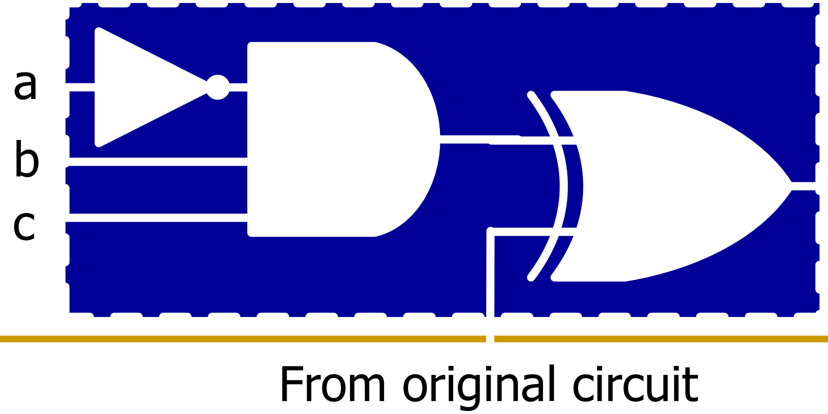
Functional Testing

Feasible Trojan space inordinately large!

Deterministic test generation infeasible

A statistical approach is, more effective

- **MERO: A Statistical Approach**
  - Find the rare events in the circuit
  - Generate vectors to trigger each rare node ***N times***
  - 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

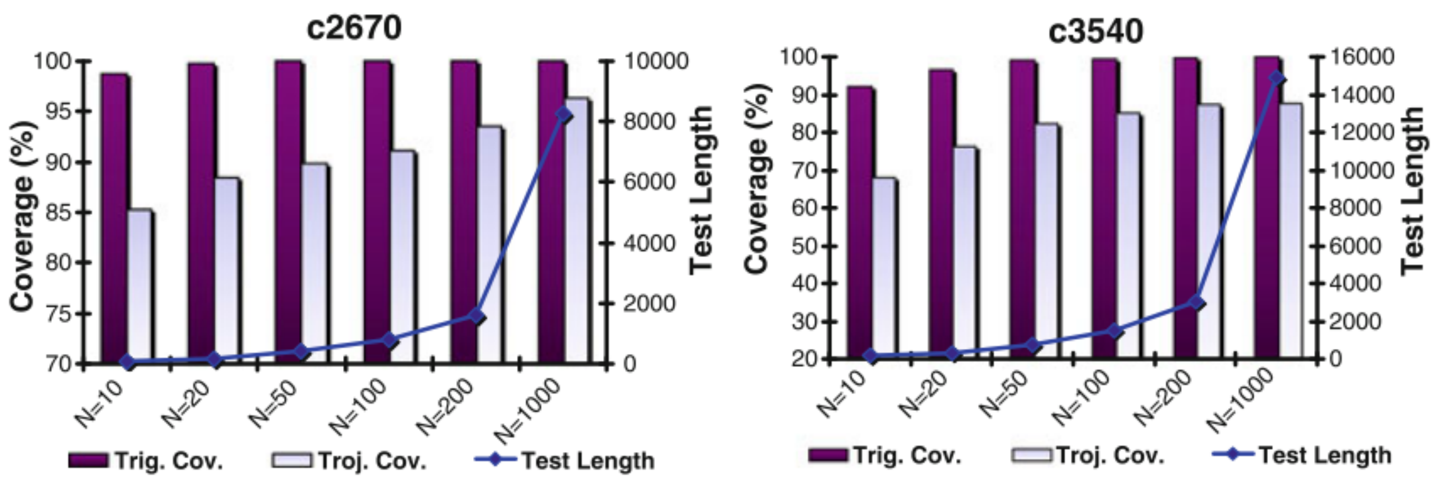


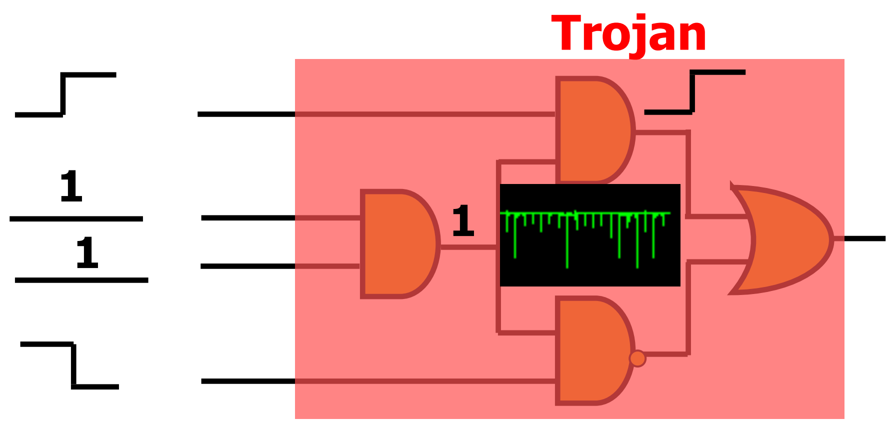
Fig. 15.6 Trigger coverage and Trojan coverage and test length for two ISCAS-85 benchmark circuits for different values of "N," using the MERO approach [8]

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

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.

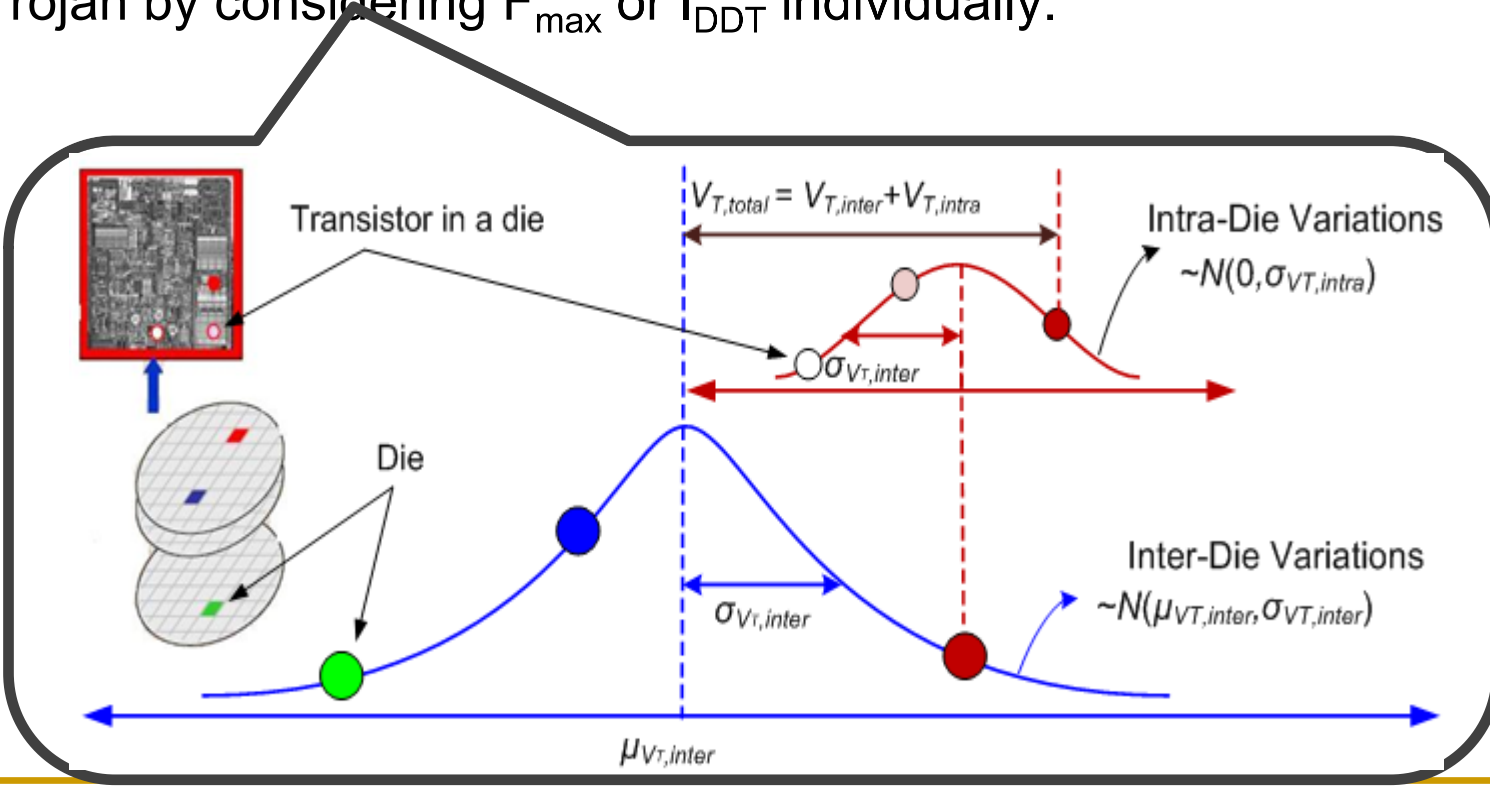
Partial Activation



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 $\downarrow$

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

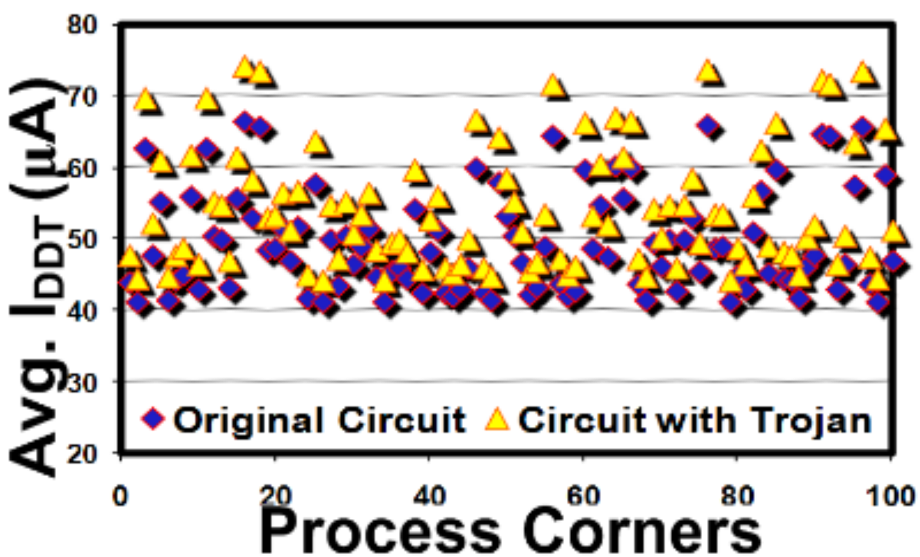
Comparing Approaches

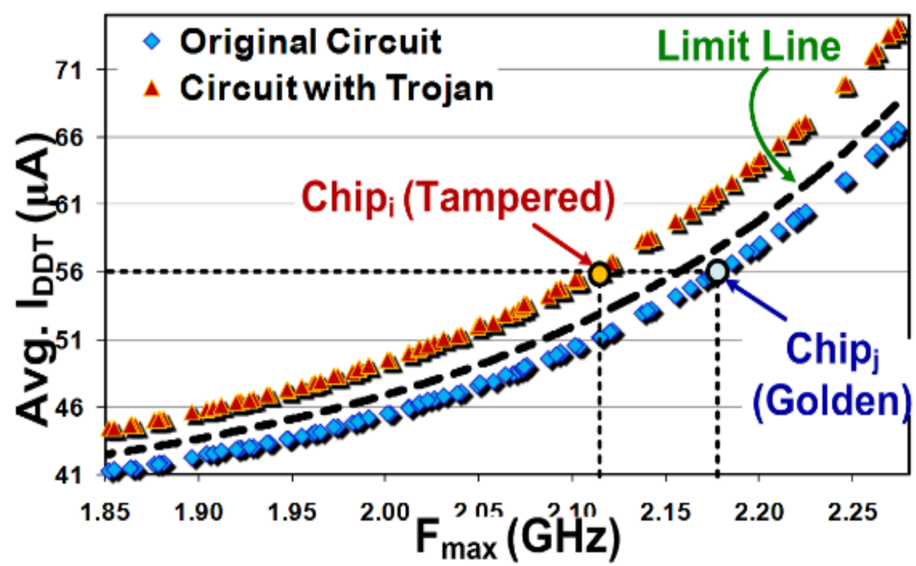
	Logic Testing	Side-Channel Analysis
Pros	• Robust under process noise • Effective for ultra-small Trojans	• Effective for large Trojans • Easy to generate test vectors
Cons	• Difficult to generate test vectors • Large Trojan detection challenging	• Vulnerable to process noise • Ultra-small Trojan Det.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

Side-channel Approach

- *<u>Multiple-parameter</u> Trojan Detection*
  - Due to process variations, Trojan detection by F_{max} or I_{DDT} alone is challenging!

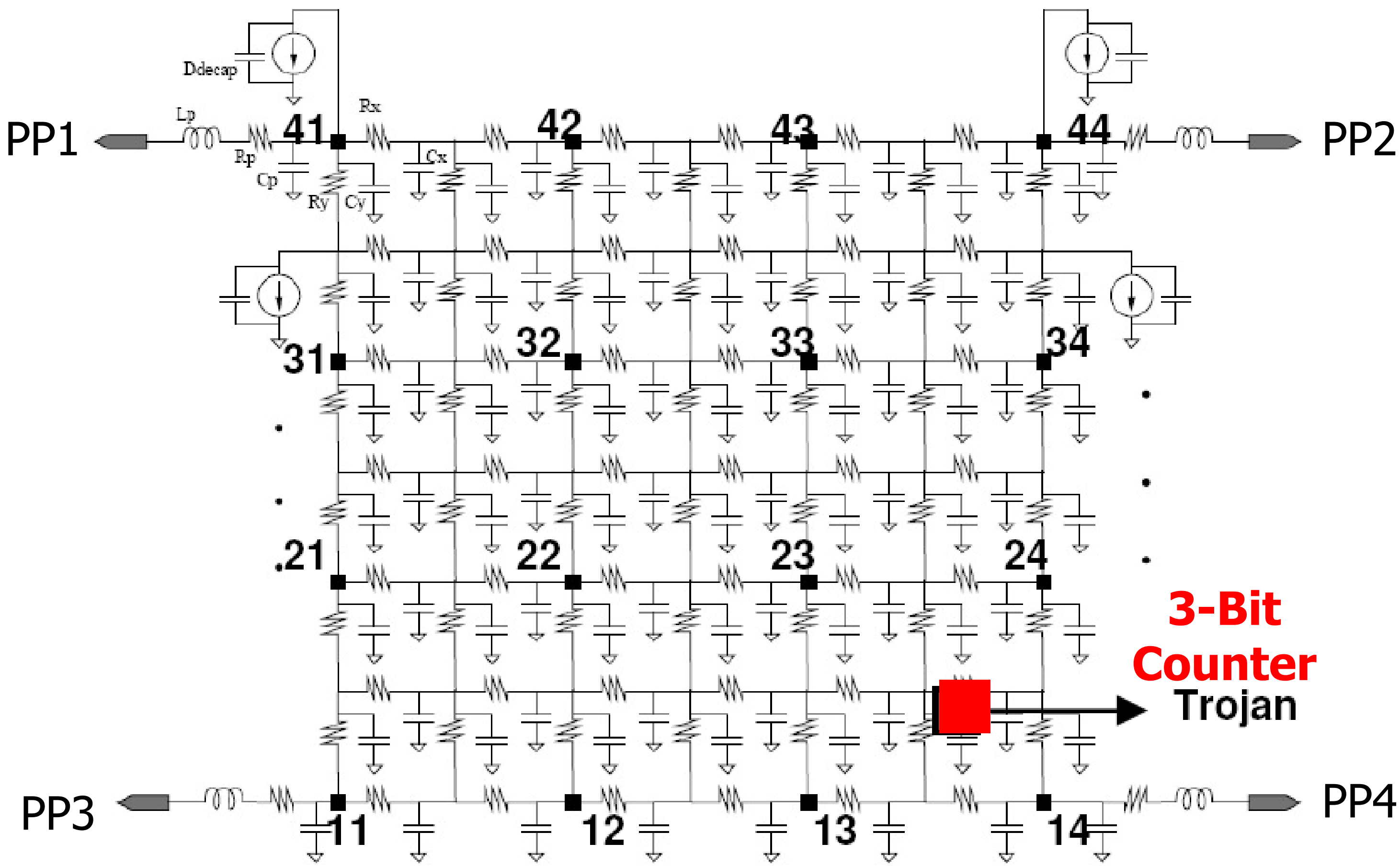




- Consider the intrinsic relationship between I_{DDT} and F_{max}

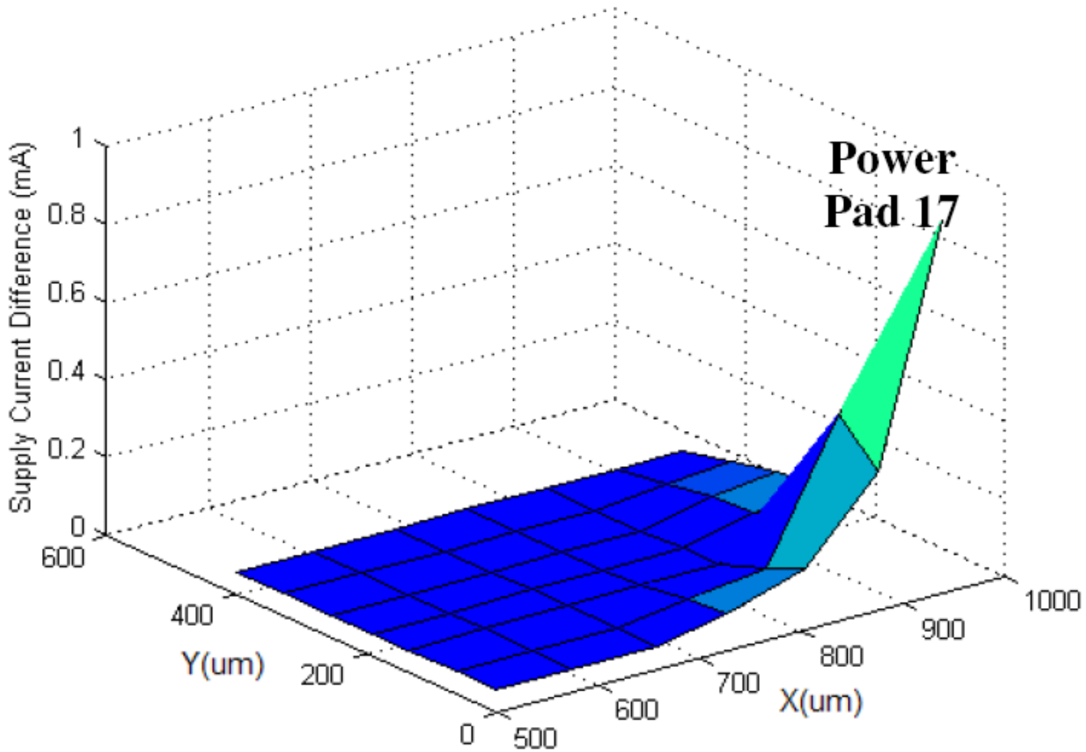
Golden chip required!

Trojan Inserted into s38417 Benchmark



PP: Power Pad

Power Analysis -- Locality



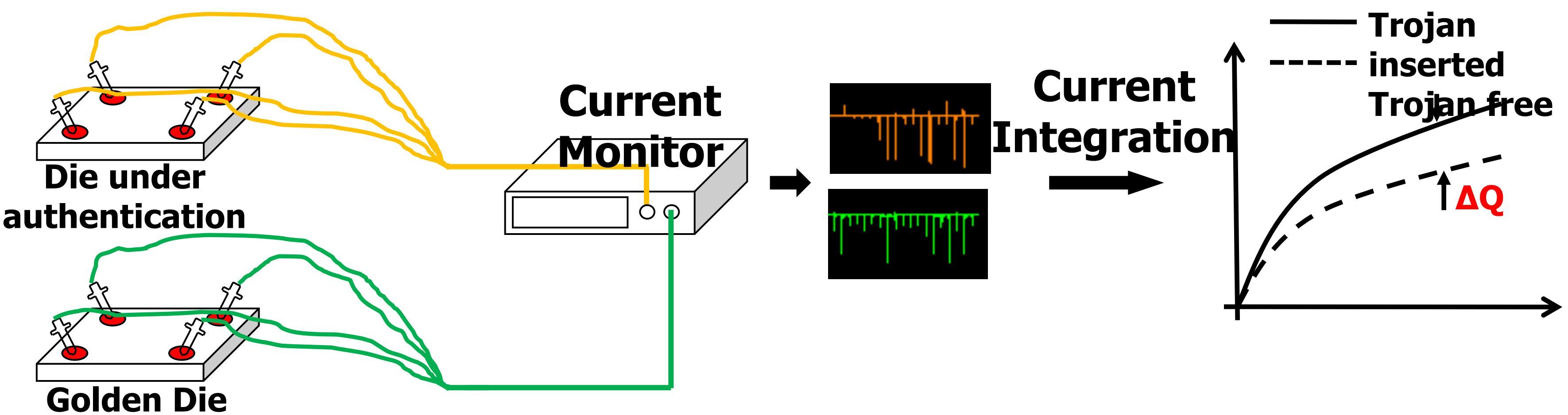
- Current difference measured from power pad 17 (Trojan-free vs Trojan-inserted)
- There is no change in layout of the circuit. Trojan was inserted in an unused space in the circuit layout.

Current (Charge) Integration Method

- Current consumption of Trojan-free and Trojan-inserted circuits

$$Q_{trojan\text{-}free}(t) = \int I_{trojan\_free}(t)\ .\ dt$$

$$Q_{trojan\text{-}inserted}(t) = \int I_{trojan\_inserted}(t)\ .\ dt = \int (I_{trojan\_free}(t) + I_{trojan}(t))\ .\ dt$$

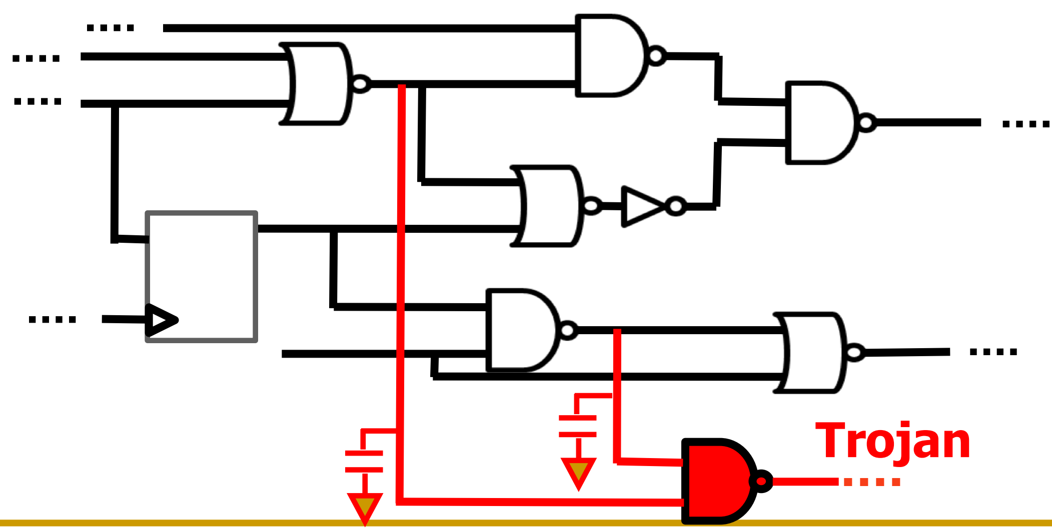


Power Analysis -- Challenges

- **Pattern Generation**
  - **How to increase switching activity in Trojans?**
  - **How to reduce background noise?**
  - **Switching locality**
  - **Random Patterns**
    - **No observation is necessary , Similar to test-per-clock**
- **Measurement Device Accuracy**
  - **Measurement noise**
- **Process Variations**
  - **Calibration**
- **On-Chip Measurement**
  - **Vulnerable to attack**
- **Authentication Time**
  - **Trojans can be inserted randomly**

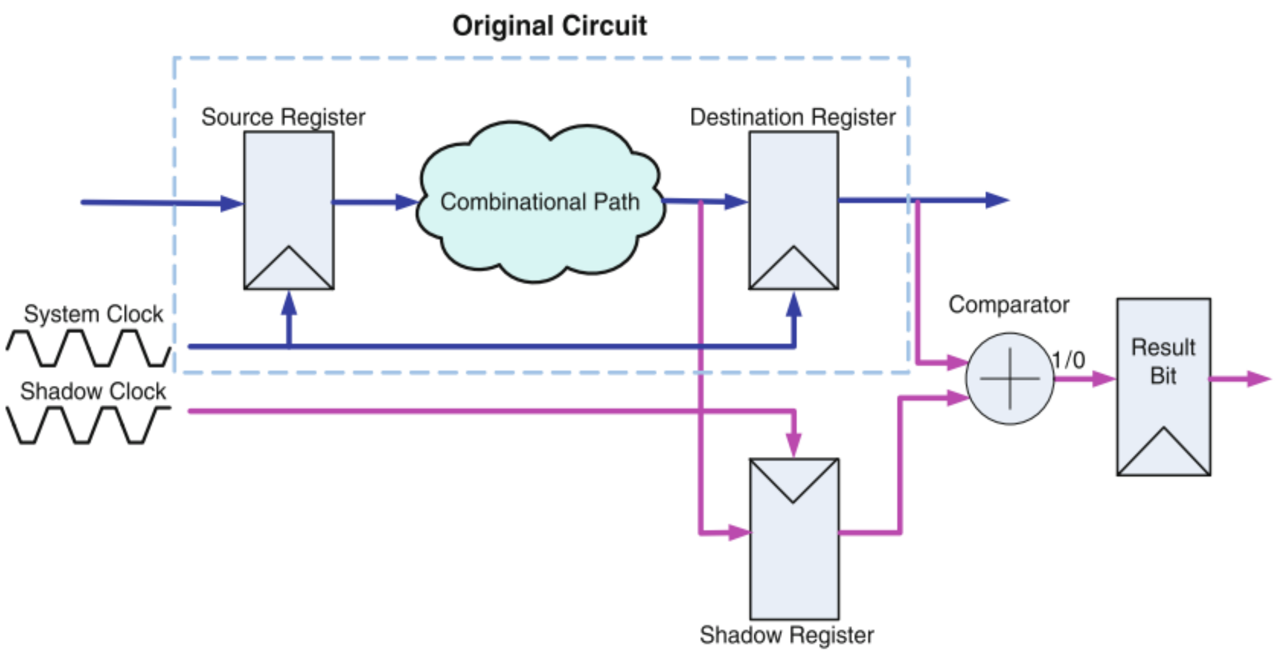
Side Channel Analysis -- Delay

- Hard to detect using power analysis are:
  - Distributed Trojans
  - Hard-to-activate Trojans
- **Path delay**: A change in physical dimension of the wires and transistors can also change path delay.
- We are developing new methods that can detect additional delays on each path of the circuit.



Delay-based Methods

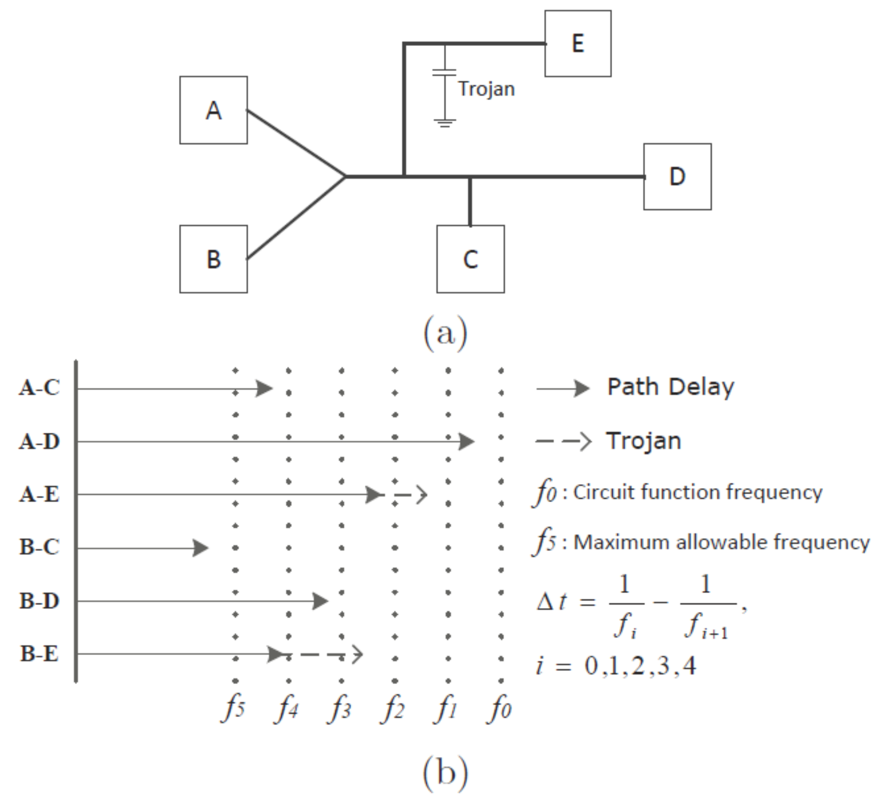
- Shadow-register provides a possible solution for measuring internal path delay.
- From this architecture, it can be seen that the basic unit contains one shadow register, one comparator and one result register.



- **Limitations**:
  - PV
  - Overhead
  - S-clock
  - Output

Clock Sweeping Technique

- Clock sweeping involves applying a pattern at different clock frequencies, from a lower speed to higher speeds.
- Some paths sensitized by the pattern which are longer than the current period start to fail when the clock speed increases.
- The obtained start-to-fail clock frequency can indicate the delays of the paths sensitized by the patterns



Delay Analysis -- Challenges

- **Major advantage over power analysis: No activation is required.**
- **Detection and Isolation**
  - **How significant is the delay inserted by Trojan?**
  - **It depends on Trojan size and type**
  - **Location: on short paths or long paths**
- **Pattern Generation**
  - **Delay test patterns**
  - **Path Coverage**
- **Process Variations (V_{th}, L, T_{ox})**
  - **Impact circuit delay characteristics significantly**
  - **Differentiate between Trojan and PV**
- **Trojan can have impact on multiple paths (an advantage over PV)**

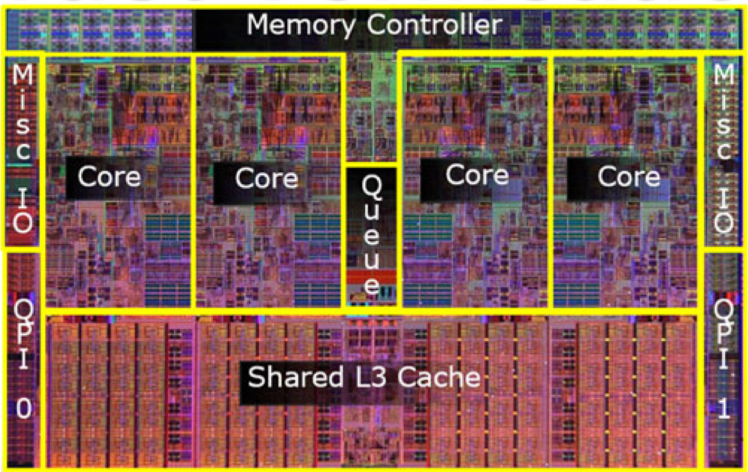
Trojan Detection

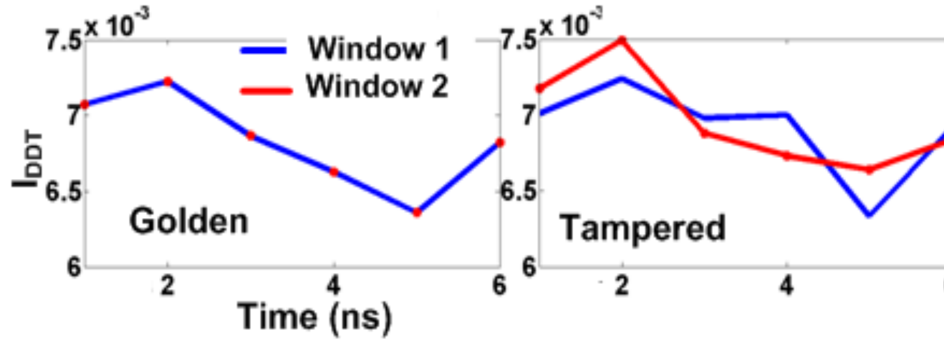
Trojan				Power Analysis	Delay Analysis	Fully Activation
Trojan Classification	Physical Characteristics	Type	Functional	D	P	P
			Parametric	P	D	P
		Size	Small		D	P
			Large	D	P	P
		Distribution	Tight	D	D	P
			Loose	P	D	P
		Structure	Modify Layout	P	D	
	Activation Characteristics	Always-on			D	
		Condition-based	Logic-based	D	P	P
			Sensor-based	D		
	Action Characteristics	Modify Function		D	P	
		Modify Spec.	Defects	P	D	P
			Reliability	P	P	P

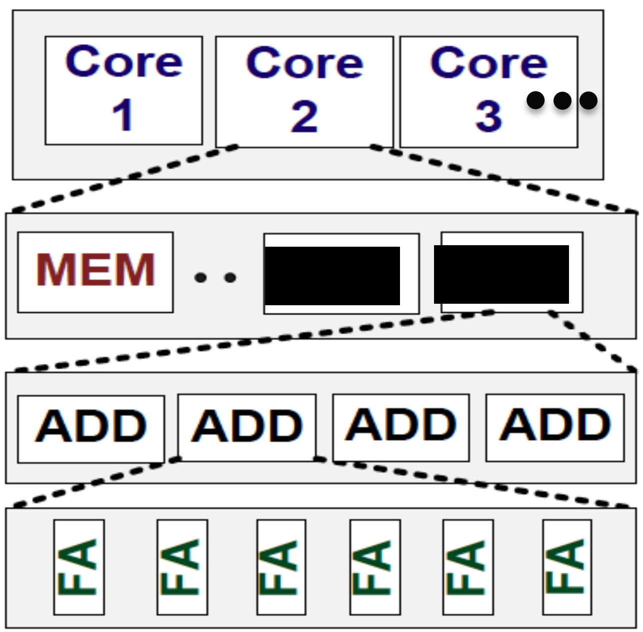
P: Detection is possible **D: High level of confidence**

Self-similarity in Space & Time – for Trust Verification

Image courtesy: Intel







Uncorrelated switching in time due to a seq. Trojan!

Simultaneously detects Trojan & aged/recycled ICs!

No golden chip required!!!

Design for Hardware Trust

- Since detecting Trojan is extremely challenging, design for hardware trust approaches are proposed to
  - **Improve hardware Trojan detection methods**
    - Improve sensitive to power and delay
    - Rare event removal
  - **Prevent hardware Trojan insertion**
    - Design obfuscation

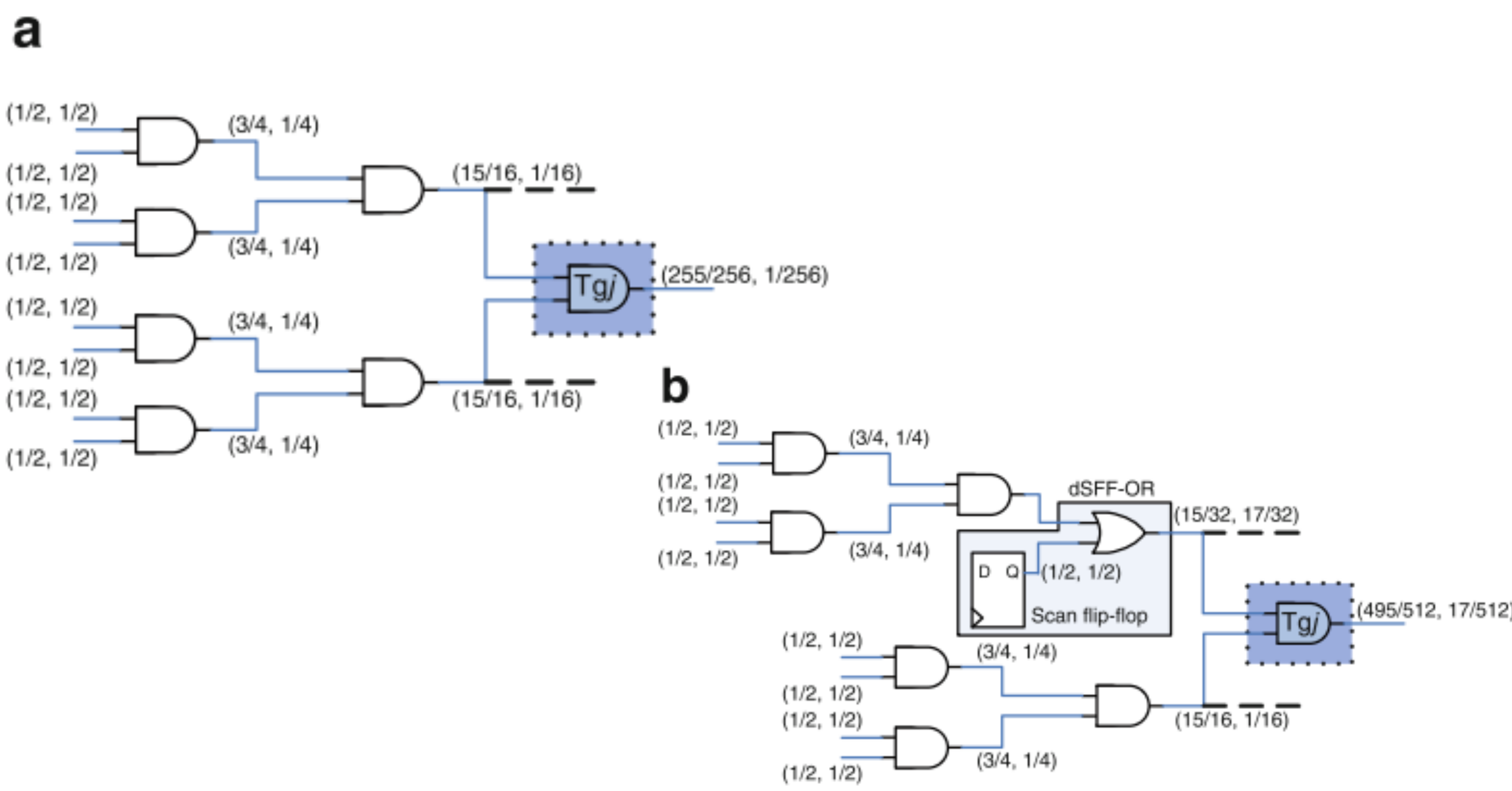
Rare Event Removal

- Intelligent attackers will choose low-frequency events to trigger the inserted Trojans.

- Improving controllability or observability can make rare events scarce, thereby facilitating detecting Trojans inside the design.
  - Design for Trojan test: inserting probing points
  - Inserting dummy scan flip-flops

Increasing Probability of Partial/Full Activation

- Inserting dummy FFs on path with very low activation probability



Increasing Probability of Partial/Full Activation

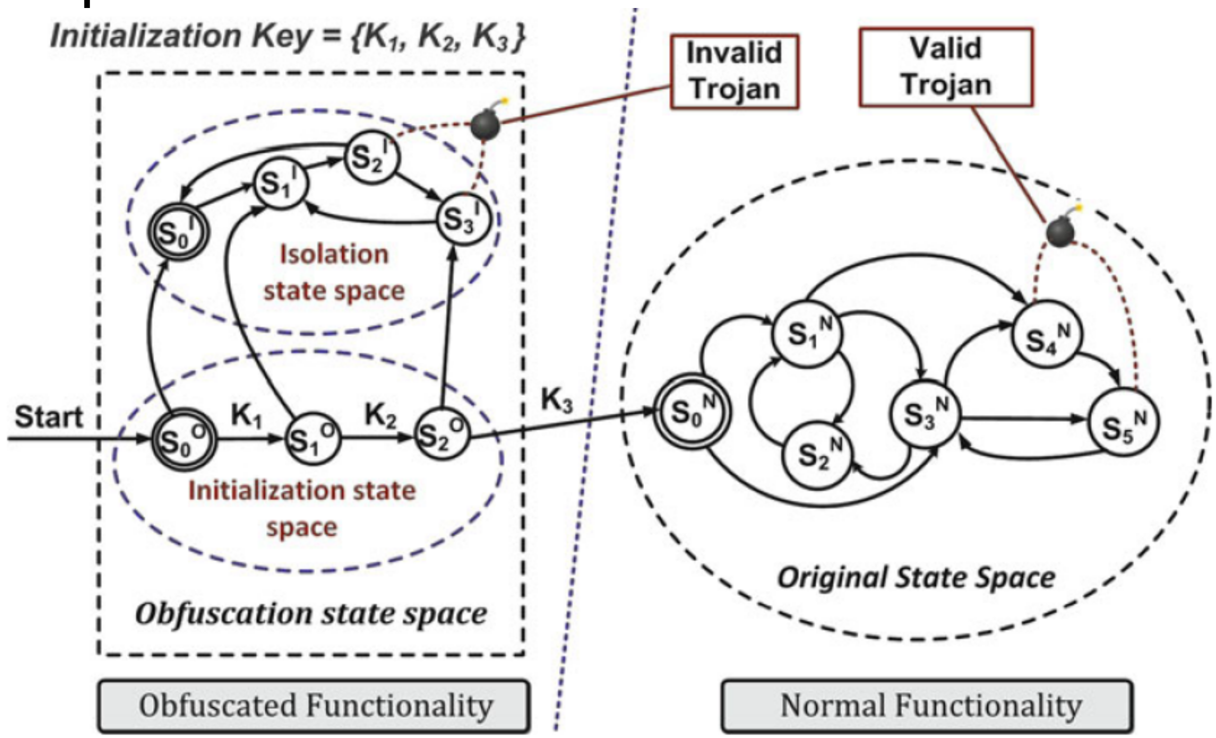
- Dummy scan flip-flops are inserted to control hard-to-excite nodes.
- Usage:
  - **Full activation**: increase controllability
  - **Power-based**: generate switching activities
  - **Delay-based**: activate more paths to improve coverage

Trojan Prevention-Design obfuscation

- The objective is deterring attackers from inserting Trojans inside the design.
- Design obfuscation means that a design will be transformed to another one which is functionally equivalent to the original, but in which it is much harder for attackers to obtain complete understanding of the internal logic, making reverse engineering much more difficult to perform.
- It obfuscates the state transition function to add an obfuscated mode on top of the original functionality (called normal mode).

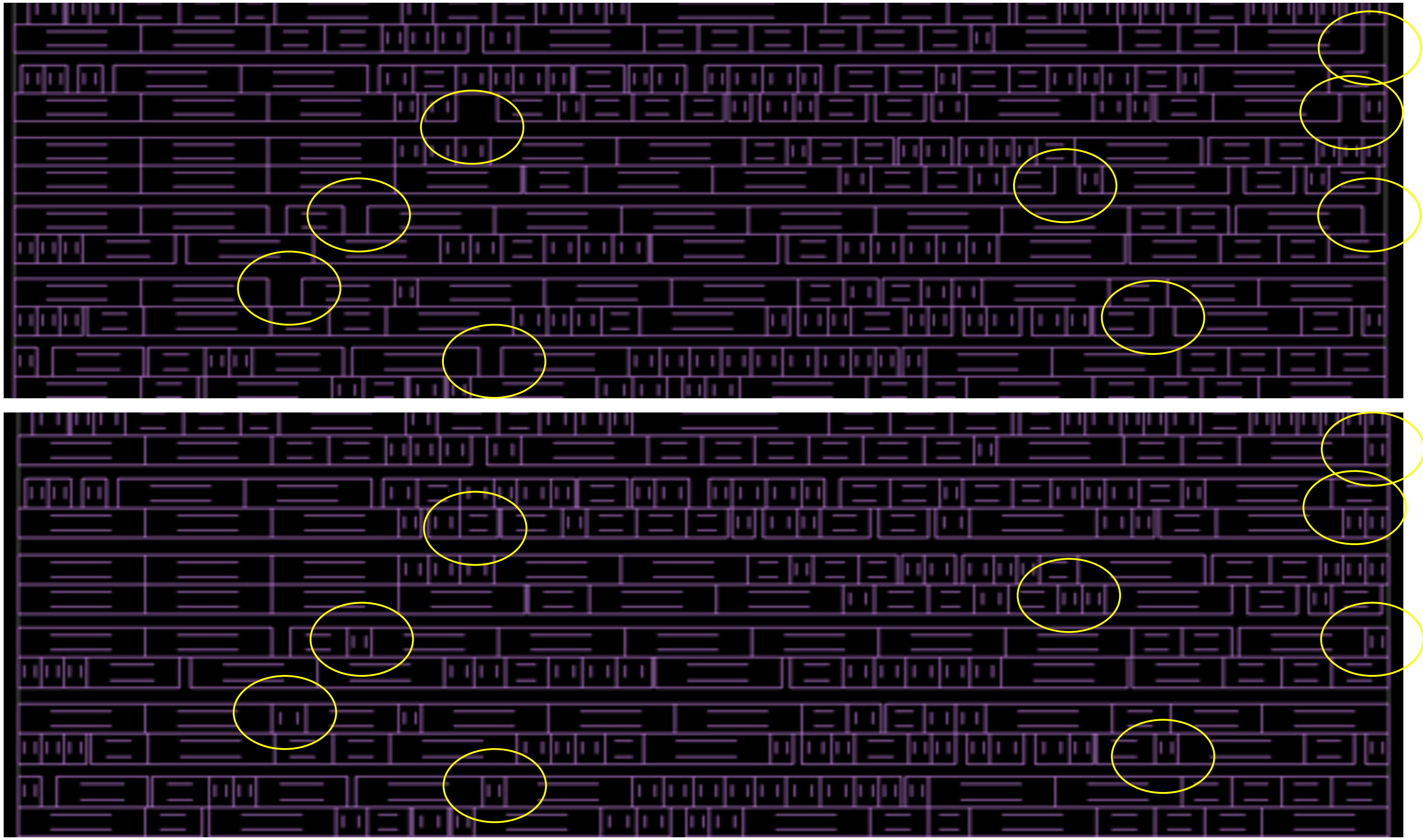
Design obfuscation

- Specified pattern is able to guide the circuit into its normal mode.
- The transition arc K3 is the only way the design can enter normal operation mode from the obfuscated mode.



BISA: Built-In Self-Authentication

- Filling all unused spaces with a circuit that can easily test itself



Question?