Hardware & Embedded Security

Prof Daniele Rossi


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


daniele.rossi1@unipi.it


050 221 7611

1

Design for Hardware Trust

Hardware Trojans Detection Methodologies

Lecture 5-Part 1 - DR

2

Brief Outline

- **Hardware Trojan detection methodologies**
  - Logic testing
  - Statistical approach
  - Side-channel analysis (power and delay)

3

Design for Hardware Trust

- Verifying the trustworthiness of manufactured ICs requires a post-manufacturing step to validate the conformance of the fabricated ICs to the original functional and performance specifications
- Current design methodologies provide an adversary with multiple opportunities to insert Trojans that can go undetected
- It is important to develop **design-for-hardware-trust** (DFHT) strategies
  - (i) to **prevent** Trojan insertion into a design and
  - (ii) to **detect** the Trojan if inserted

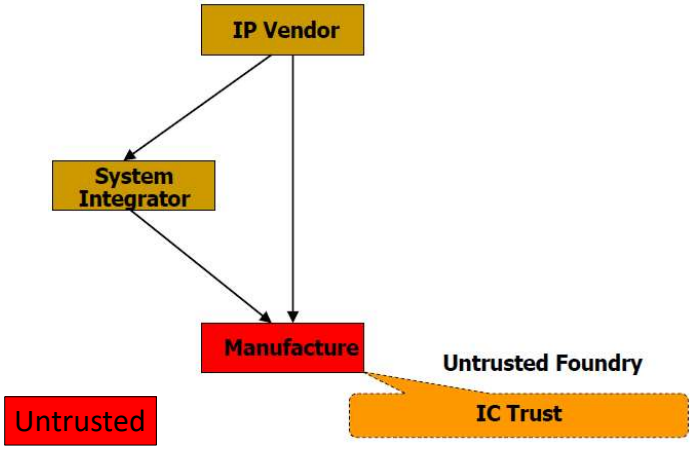
→ ICs must be designed in such a way that **undetected changes** to a circuit are near **impossible**

4

How can we identify hardware trojan?

1. Of course logic testing
2. Statistical approach, but we will focus mostly on
3. Side channel analysis.

IC (System) Trust



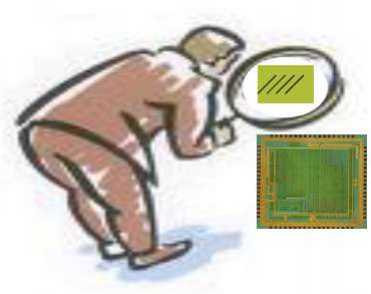
- **IC Trust Objective**
  - Ensure that the fabricated chip/system will carry out only our desired function and **nothing more**

We will focus on IC trust for the reasons we discussed previously (IP)

13

IC (System) Trust

- **Challenges:** SMP S
  - **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 additional functionality in a design

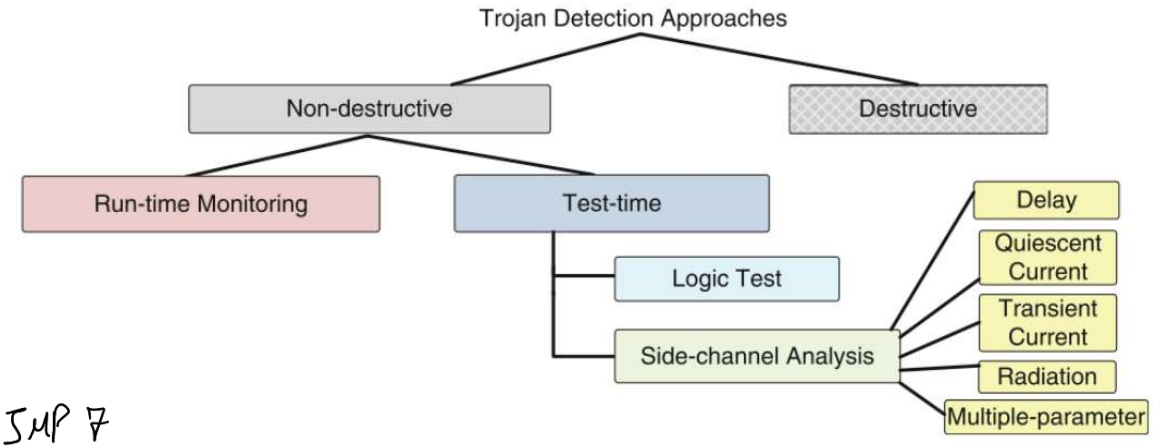


14

1. HWTrs are tiny compared to our actual system (Millions of gates)
2. HWTrs are mostly quiet before activation and occurrence of activation event as rare (you can also have timers): not simple to detect them.
3. Hard to model: they can be anything really.

- You can run a conventional test, however: an activated Trojan makes HW behave in anomaly ways. You could detect anomalies. But: you still need to activate it. So approach is not so effective, simple and tests don't scale very well. You will need some kind of side channel analysis.

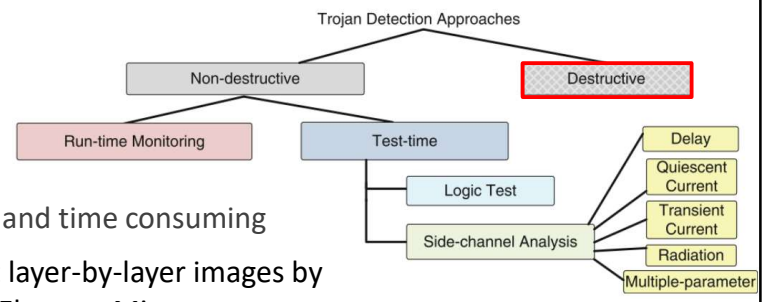
Classification of Trojan Detection Approaches for IC Trust



JMP 7

15

Classification of Trojan Detection Approaches for IC Trust



- **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

16

- You have two main approaches for detection:
  1. Destructive:
     - Reverse Engineering: you open the chip up and layer by layer you analyze the chip. This approach looks at the very physical design of the chip, but for it to be effective you need a golden model for comparison.
  2. Non Destructive:
     - Runtime monitoring: idea is observing behavior of chip at runtime to scan for abnormal behavior. Sometimes called anomaly detection. Also used for maintenance detection or intrusion detection. To do this you need to be able to model normal behavior to know how to recognise abnormal behavior. You could also use machine learning, but you need to retrain it still. Anyway idea is compare behavior of system with normal behavior. To do this, you can exploit pre-existing redundancy in the circuit. You can perform, for ex., a test in multiple cores and see if they behave the same way (as long as HW Trj is not everywhere, IP Trj case). Basic idea is that.
     - Test-Time (no runtime):
       - Logic Testing-based approach to try and activate trojans CALL 21 (PAG. 10)
       - Side channel analysis-based approach: we will focus mainly on delay and power (can be dynamic power but also static power), but also radiation.

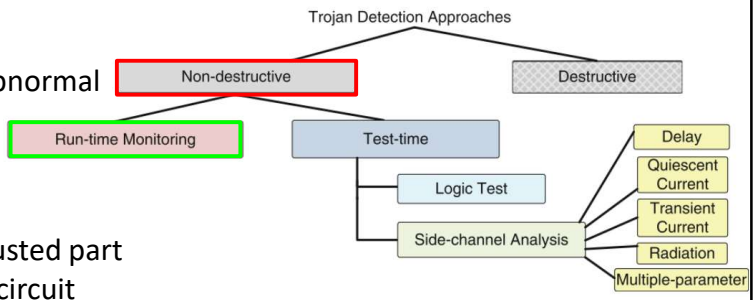
STEP BACK ON LOGIC TESTING: Consolidated technique. Equipment needed is easily available in big companies. To run logic testing you need to identify a set of test inputs and be able to control behavior of specific areas of your system and observe effects of control. You focus on activating certain ops in a specific area (control). But in a complex system you also need to propagate effect of the control (observability, another important property).

Test is straightforward and easy. BUT you don't know what HWTrs is going to look like. Plus in a complex system you have a lot of inputs, internal states. It is impossible to check all the possible states and input configurations. Plus, HWTrs activate on very rare events. And you still cannot focus on traces that activate externally.

SMP 22 (Slide pag. 10)

Classification of Trojan Detection Approaches for IC Trust

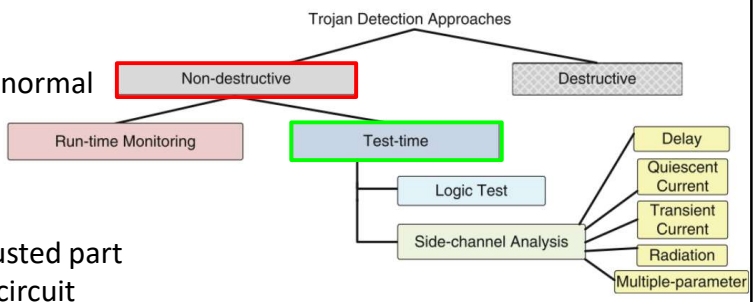
- **Non-destructive Approach:**
  - **Run-time monitoring**: Monitor abnormal behaviour during run-time
    - Exploit pre-existing redundancy in the circuit
    - Compare results and select a trusted part to avoid an infected part of the circuit



17

Classification of Trojan Detection Approaches for IC Trust

- **Non-destructive Approach:**
  - Run-time monitoring: Monitor abnormal behaviour 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

18

Logic Testing Approach

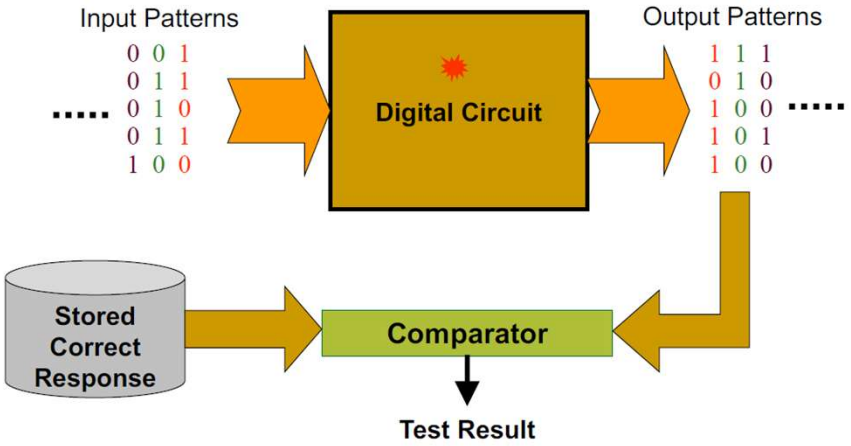
- **Logic-testing approach** focuses on test-vector generation for
  - Activating a Trojan circuit
  - Observing its malicious effect on the payload at the primary outputs
  - Both **functional** and **structural** test vectors are applicable
  - **Pros**:
    - Straight-forward and easy to differentiate
  - **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

21

- This is the idea

Fare slide analoga su structural test, se possibile

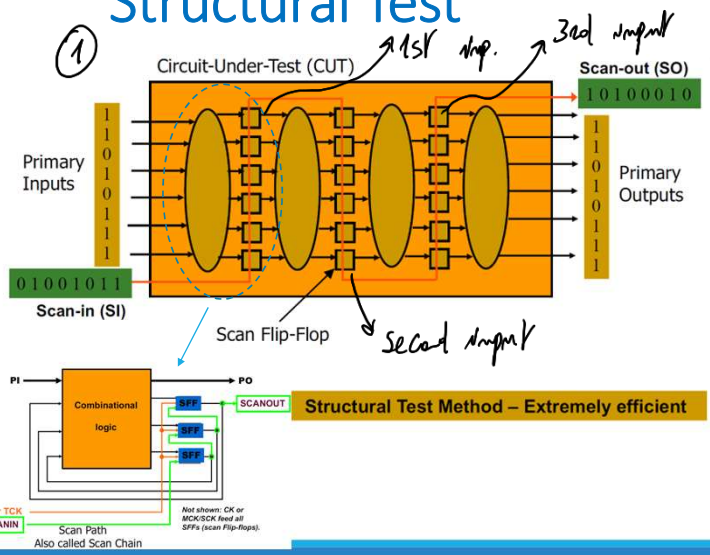
Functional Test



22

SMP 12

Structural Test



Structural Test Method – Extremely efficient

Attractive aspects of structural test: ability to fit quickly and easily into the IC development process. It relies on:

- **DFT (Design for Test) methods**, which help engineers make designs more testable through the use of specific design techniques → additional test access points, on-chip test structures like **scan** and **built-in self-test** (BIST) and path or transition delay test methods
- These techniques **increase observability** and **controllability** of circuit elements that otherwise require increased test development times to reach, hence increasing cost-of-test and delaying time-to-volume.

23

Functional Test vs Structural Test

- Until recently, manufacturers have relied on engineers to write test programs able to identify **failures in device function** - **functional test** - and have acquired sufficiently **powerful** automatic test equipment (**ATE**) resources to avoid any bottlenecks in production test and product shipment
- Rising **device complexity**, however, drives a proportionally **faster increase in test complexity** that simply surpasses the capabilities of test engineers and ATE to achieve cost-effective functional test solutions → IC companies have begun to deploy **structural test strategies** designed to uncover **manufacturing defects** within the most complex ICs (https://www.design-reuse.com/articles/5100/what-designers-need-to-know-about-structural-test.html)
- Functional test attempts to answer the question "**Does it function correctly?**"
- Structural test attempts to answer the question "**Was it manufactured correctly?**"

24

- Structural test simplifies our ability to observe behavior of internal nodes of your system. You have a chip modeled as a set of cascading pieces of random logic in slides. You have troubles observing internal nodes (control + propagation is hard). Here, you have internal flip flops that can be arranged as kind of shift registers. So those Flip Flops can also be set to work as shift registers when you want to perform tests. You have a real time going through flip flops. This makes control and observe parts because you can shift in values that you like in the flip flops directly (so you bypass primary input). As you have shifted in the inputs, you apply them in your blocks. First line is input for the 1st one, second for the 2nd one etc. Then you can sample outputs of FF by using them and if you shift out contents of shift registers in output.

This of course increases complexity of system (modify FF, add Multiplexers). But for more efficient than logic tests.

JMP 26 (page 13)

Hardware Trojan Detection: Untrusted Manufactured Integrated Circuits

- Trojan circuits are designed to avoid detection, triggering only under rare conditions
- Trojans are silent most of their lifetimes and have a very small size, relative to their host circuits, and make only limited contributions to circuit characteristics
- These qualities suggest that they most likely connect to nets with low controllability and/or observability
- **Controllability**: The ability to **induce any signal on a line** → **activation** of the signal on a given line
- **Observability**: The ability to **observe** the **changes** on lines at **primary output** → **propagation** of the effect of the signal to output pins

25

Controllability and Observability

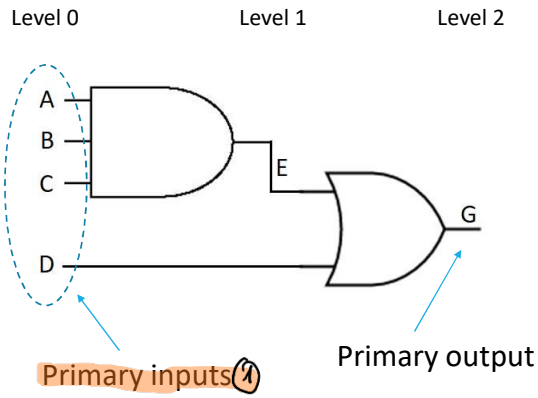
- **Controllability**: The ability to induce any signal on a line → activation of the signal on a given line
- **Observability**: The ability to observe the changes on lines at primary output → propagation of the effect of the signal to output pins
- Combinational logic is usually easy to observe and control
- Sequential logic (finite state machines) can be very difficult to observe and control, requiring many cycles to enter a desired state → Structural tests help
- Good observability and controllability reduces number of test vectors required to testing → make it easier to detect the presence of a HT

26

If you want to check for anomalies at level 1, you can try to force E at 1, so you have to control A, B, C to 1 (you only have access to primary inputs) ①
But even if you set E to 1, you don't necessarily see it. You need to propagate value to primary output. So you need to force primary output to depend only on E. Set D=0.

Controllability and Observability: Example

Controllability

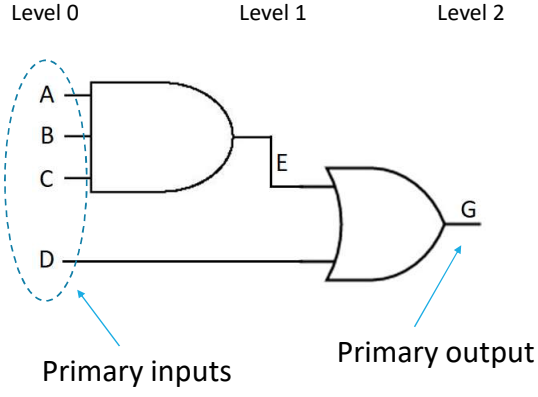


- Controlling the **output** of a multi-input **AND gate** to **logic 1** requires the control of all its input to 1
  - Controlling a node at **level 1** (such as node E) requires controlling the node at level 1 (E itself) and three nodes at level 0 (primary inputs) A, B and C → controllability cost increases from stepping from one level to the next one (it becomes more difficult)

27

Controllability and Observability: Example

Controllability



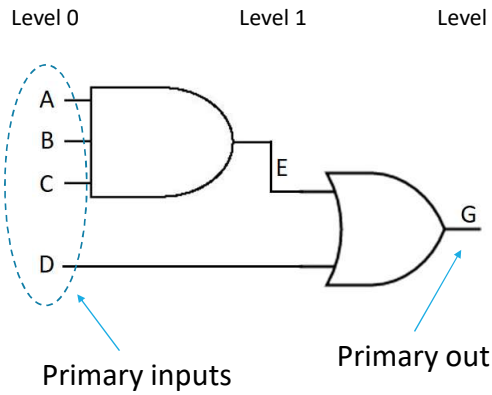
- Controlling a node at **level n** requires controlling the nodes at **levels 0 .. n-1**

  → controllability "cost" increases from stepping from one level to the next one (it becomes more difficult)

28

Controllability and Observability: Example

Observability

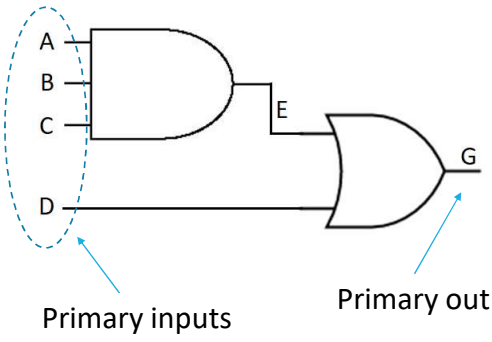


- To **observe** an **internal node** (E), it is necessary to control D to 0 in order to sensitize through the OR gate the primary output G
- Similarly, observability of D at G requires controllability of E to 0.
- The other primary inputs, A, B and C, have equal observability

29

Controllability and Observability: Example

Observability



- Consider primary outputs are at level n Then, observing a node at **level i** usually requires controlling some nodes at **levels i .. n-1**

→ observability "cost" increases from stepping from one level to the previous one (it becomes more difficult)

JMP 18

30

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)
  - $2^{64} > 10^{18}$: This is impractical
  - 100MHz is used → 10^{10} s → 317 years; @10 GHz (exaggerate!) → 3.17 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

31

Any questions so far?

36

HT Detection Using Side-Channel Signal Analysis

- Side channel signals include
  - **Timing signals**
  - **Power signals**
- Trojans typically **change a design's parametric characteristics** for example, by degrading performance, changing power characteristics. This
  - → influences **power** and/or **delay** characteristics of wires and gates in the affected circuit
  - → introduces reliability problems in the chip

37

HT Detection Using Side-Channel Signal Analysis

- **Power-based** side-channel signals provide visibility of the **internal structure** and **activities** within the IC, enabling detection of Trojans without fully activating them You don't need to activate all the Trojan to detect it.
- **Timing-based** side channels can detect a Trojan's presence if the chip is tested using **efficient delay tests** that are sensitive to small changes in the circuit delay along the **affected paths** and that can effectively differentiate Trojans from process variations
  - HTs do not need to be activated RET

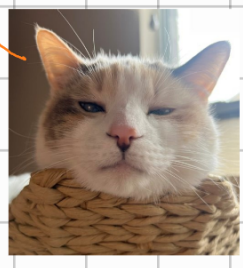
38

SIDE CHANNEL ATTACKS

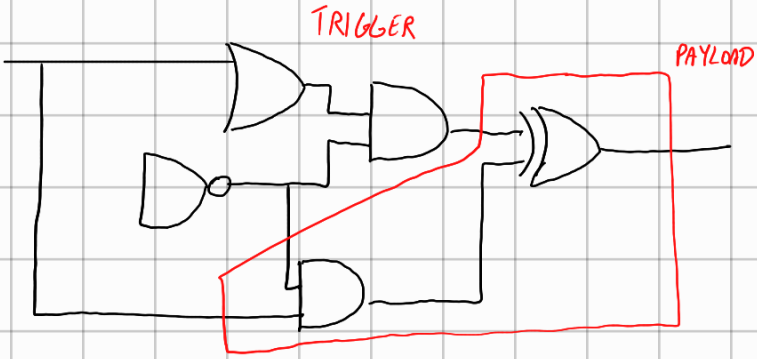
- We need to characterize behavior of system in terms of timing behavior (signal propag) or power consumption of our circuits.
- What is the base assumption for timing and power analysis? We need GOLD MODELS.
  ↳ HWTrs can alter both timing and power behavior of system. Impact is not gonna be massive, but still there. We can detect it.

For delays, we mentioned for counterfeiting that an increased delay can also be something related to reliability and not only performance. Why can we expect that an increase in delay is linked with reliability problems? (More valid for counterfeiting) [SOMETIMES INCRENSE IN DELAY CAN REFLECT MISBEHAVIOR SO RELIABILITY ISSUES]

- Additional delay can be masked electrically by your system up until the delay doesn't go over the constraints set by setup, hold and propagation delay times. But again, this is more related to recycled components. Performance degradation should be so large that it exceeds margins set by system.



- For power analysis what can we expect from HWTrs? In order to detect an HWTrs through power analysis you need to activate it. Easier said than done. But what about delay?

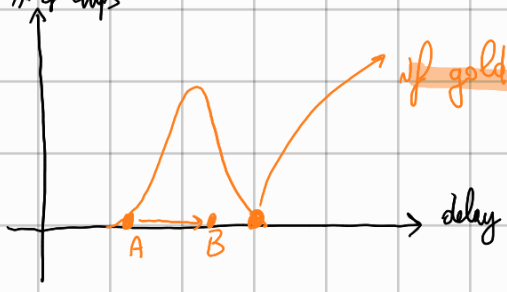


For power analysis it would be better to activate Trj. For path delay nope. Why? Because regardless of activation, extra connections introduce extra load capacitance, which means extra delay.

PANKO REASONS

CALL 38 (slide 17)

In all cases, you are going to need a golden model. Pros and cons for side channel analysis: you don't need to identify a misbehavior of the system (functional problem), just that you have anomaly in delay / power consumption. See if they fall out of your distribution of delay / consumption. You have distribution of parameters because of process variation.



If gold module falls here, you have a slow industry case. If the Trojaned chip was in A and Trojan moves it to B, you cannot tell!

So we still need to take Process variation into account for both power and delay.

NOTE: When we say power, you might focus on the instant power CALL 40 (page 20) but it might not be easy: you might have small differences and need good equipment to measure it.

- You need to consider simulation of your process: not only process variation, you can have measurement noise for Power analysis (and difference is minimal so any noise is bad). To remove noise, replicate measurement multiple times and if noise is pretty much random, it is going to be negligible if you take average. Still, to have an indication of Trj you need differences beyond process variation.

JMP 42 (page 21)

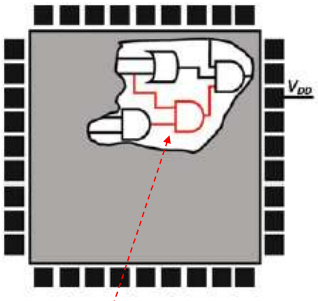
Side-channel Signals

- 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

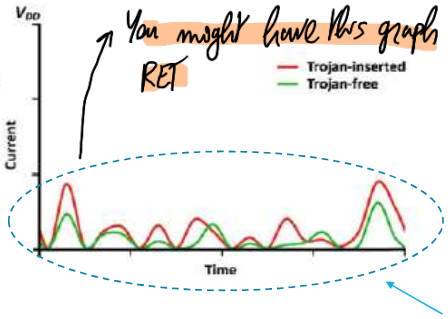
Golden chip required!

39

Power-Based Hardware Trojan Detection



Hardware Trojan



- In power-based techniques, the **power consumption** of **IC under authentication** (IUA) is compared with that of **Trojan-free (golden) circuit**
- Measured current from VDD pin in Trojan-free and Trojan-inserted circuits over a specific time interval

40

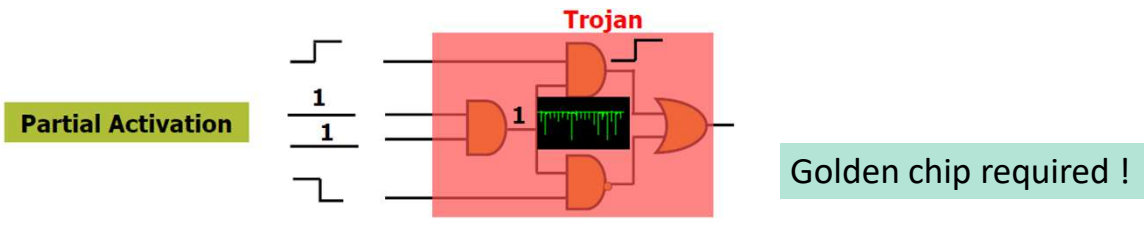
Power-Based Hardware Trojan Detection

- Each current measurement consists of several elements, including:
  1. the **main circuit current** consumption which is the same for all chips
  2. **measurement noise** which can be eliminated by averaging several measurements
  3. **process variations** which are random and cannot be cancelled
  4. **Trojan** contributions if they exist
- Any measurable **difference beyond process variations** can be an indication of **Trojan** existence

41

Power-Based Hardware Trojan Detection

- 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



Golden chip required !

42

Power-Based Analysis: Sensitivity Metric

- Improving Detection Sensitivity

Definition of a sensitivity metric

Trojan Size ↓ → Sensitivity ↓

Original Circuit Size ↑ → Sensitivity ↓

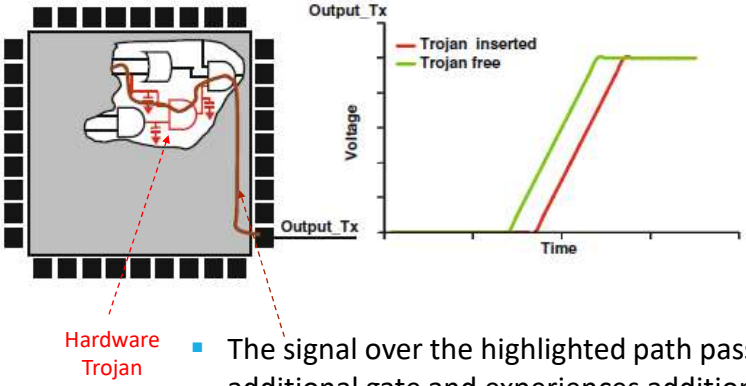
MATTER OF RELATIVE SIZE

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

Is vs the current drawn from power supply by chip.

 JMP 23

Delay-Based Hardware Trojan Detection



- Delay-based techniques analyze the **impact** of Trojans on design **performance**
- Any additional gates or wiring introduces **extra capacitances**, and then any rising or falling on **Trojan-inserted paths** creates extra time for transition
- The signal over the highlighted path passes through extra wiring and an additional gate and experiences additional delay due to the resistance and capacitance of the extra wiring and propagation delay of the Trojan gate

- Delay: by the very fact that something extra is present, you have extra capacitance. Propagation delay depends on how conductive the circuit is, supply voltage, and load capacitance.
  ↳ (Current)

- A higher supply voltage will give a higher current for the mos
- Note that on very low values of current, current doesn't increase very much quadratically with supply voltage, and a higher V_{DD} means a higher logic 1 to reach → more time.

But what matters is that higher load → higher delay. Let's focus on power:

$$P_{dyn} = \alpha C_D f_{CLK} V_{DD}^2$$

↳ activating HW increases α. Higher load.

Note that load capacitance has also effects on power consumption, so even its existance has detectable effects for power consumption, but the difference is going to be minimal if not activated.

PATH DELAY FINGERPRINTING

- You select a few paths to analyze, and of course you need a golden model. Still you have the problem of process variation! Both wafer die than wafer die (inside the chip you still have some process variation).

JMP 47 (pag 25)

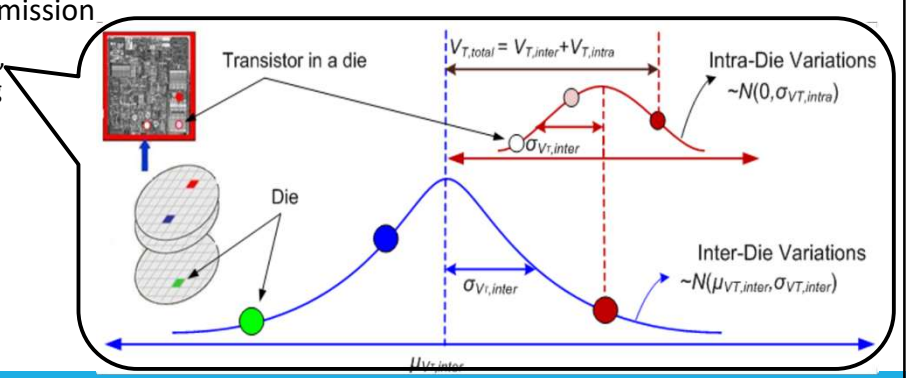
Delay-Based Hardware Trojan Detection

- Possible delay-based technique: **Path delay fingerprinting** (similar to the technique seen for counterfeit detection) (e.g., [3])
- A circuit has many paths, each representing one part of the entire circuit characteristic
- The technique measures the delay of **several nominated paths** on **several chips** to bring process variations into account
- Afterwards, the **chips are reversed engineered to ensure they are genuine**, and their measurements are used as a signature
- The same measurements are performed on other chips and compared with the signature
- Any difference can be an indication of Trojan insertion

45

Process Variations

- 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 Fmax or IDDT individually.



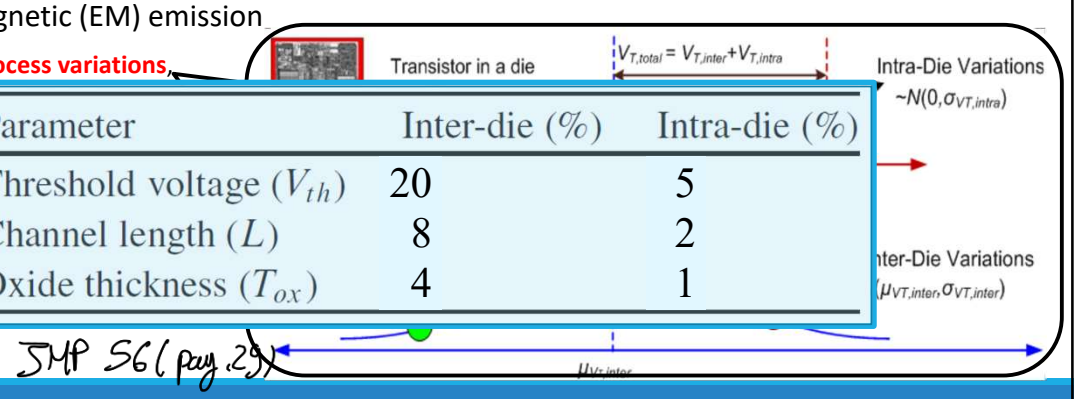
16

46

Process Variations

- 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 ext to dete conside individu

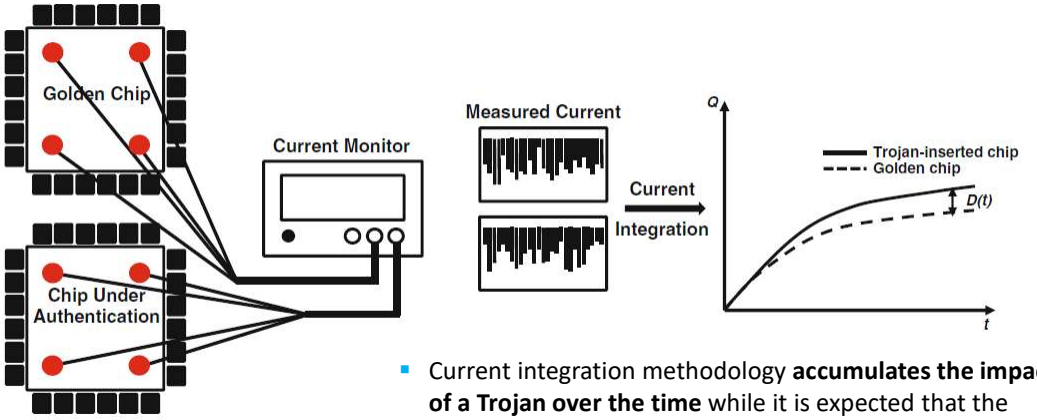
Parameter	Inter-die (%)	Intra-die (%)
Threshold voltage (V_{th})	20	5
Channel length (L)	8	2
Oxide thickness (T_{ox})	4	1



SMP S6 (pag. 29)

47

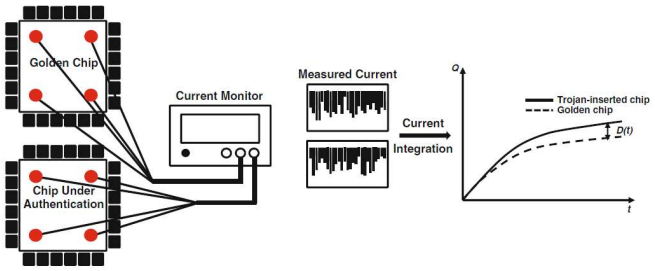
Power-based Side-Channel Analysis: Current (Charge) Integration Method



- Current integration methodology **accumulates the impact of a Trojan over the time** while it is expected that the process variations impact is cancelled out by integration

48

Current (Charge) Integration Method



- First, a **golden chip** is identified
- Next, an average **current waveform** is formed in response to a **pattern set**
- Then, the pattern set is applied to **each CUA**, and the current is **measured locally** via power pads

- The small current consumption difference between Trojan-inserted and Trojan-free circuits can be increased through the integration process
- In the case of a Trojan's existence in a chip, **more current difference** can be **measured** by applying **more patterns to the chip**, making the Trojan detection task easier

Current (Charge) Integration Method: Analysis

- $I_{Trojan\text{-}free}$: current drawn by Trojan free circuit
- $I_{Trojan\text{-}inserted}$: current drawn by Trojan-inserted circuit

- The integration current at time t for Trojan-free and Trojan-inserted circuits is:

$$Q_{Trojan-free} = \int_0^t I_{Trojan-fre}\ (t)dt \qquad Q_{Trojan-inserted} = \int_0^t I_{Trojan-inserted}\,(t)dt =$$

$$= \int_0^t \left[I_{Trojan-free}(t) + I_{Trojan}(t)\right] dt$$

- Since the same pattern set is applied to both a golden chip and a CUA, the difference between $I_{Trojan\text{-}free}$ and $I_{Trojan\text{-}inserted}$ comes from
  I. the additional current drawn by Trojan gates
  II. changes in the circuit current due to process variations

Current (Charge) Integration Method: Analysis

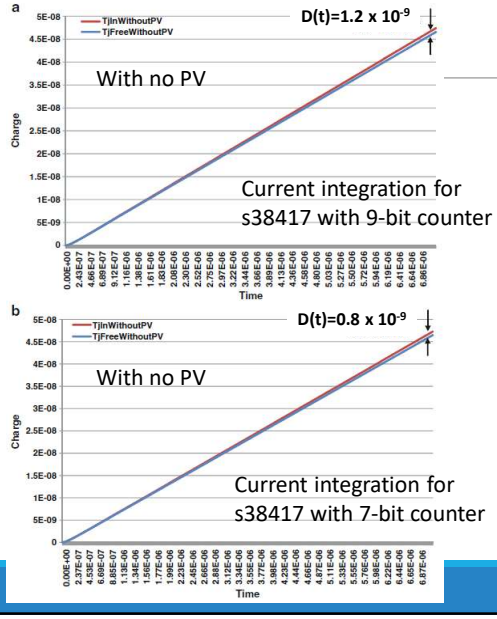
- By **integrating the current along the time axis** for both chips, their **cumulative difference** at time *t*, $D(t)$, can be increased by **applying more patterns**

$$D(t) = Q_{Trojan-inserte} - Q_{Trojan-free} = \int_0^t I_{Trojan}(t)dt$$

- When $D(t) \geq D_{th}$ → the chip is identified as a Trojan-inserted chip
- Threshold D_{th} is determined by the **Trojan detection timing budget** as well as the **current measurement device resolution**
- Current integration capability does not depend on the location of a Trojan in a circuit, since current is measured locally through power pads → **Trojan** circuitry **impacts** at least **one power pad**

51

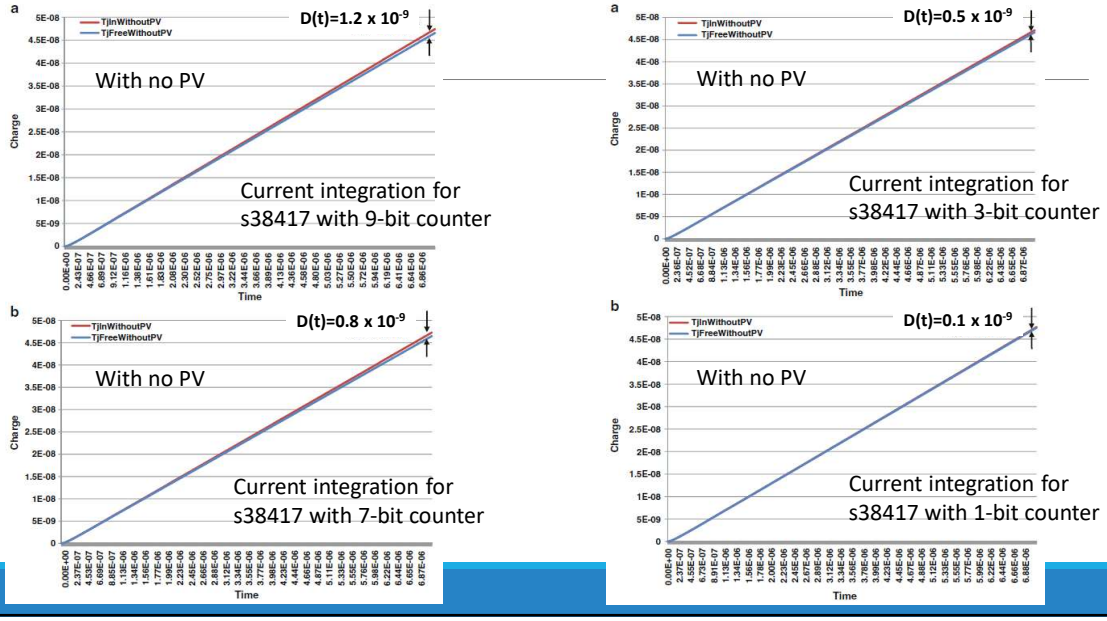
Current (Charge) Integration Method: : Example



S38417 from ISCAS89 Sequential Benchmark Circuits:
28 inputs
106 outputs
1636 D-type flipflops
13470 inverters
8709 gates (4154 ANDs + 2050 NANDs + 226 ORs + 2279 NORs)

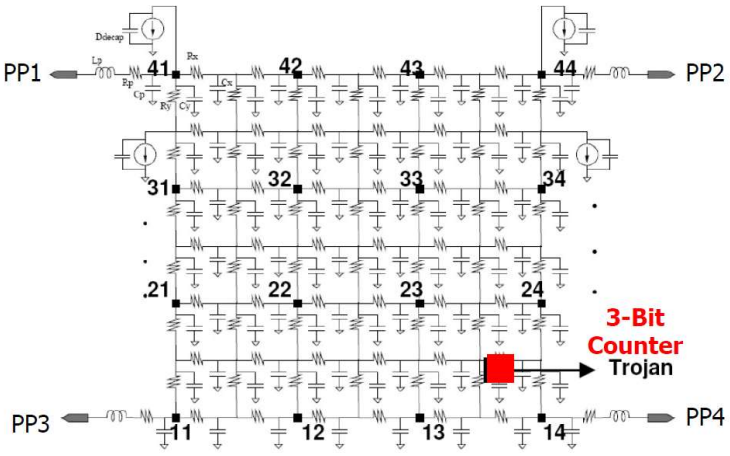
52

Current (Charge) Integration Method: Example

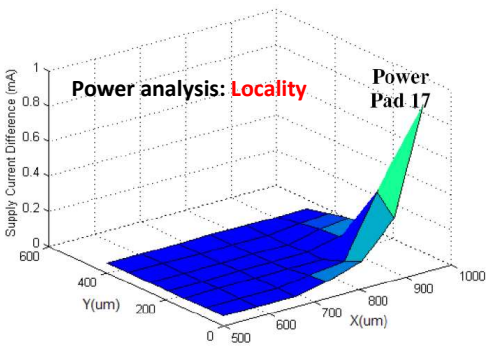


53

Example: Trojan Inserted into s38417 Benchmark



PP: Power Pad



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

Source: M. Tehranipoor, Univ. of Florida, slides from the course Introduction to Hardware Security & Trust, Course Slide

54

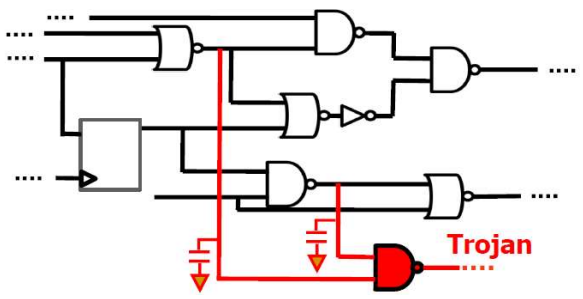
Power Analysis: Challenges

- **Pattern Generation**
  - How to increase switching activity in Trojans?
  - How to reduce background noise?
  - Switching locality
- **Measurement Device Accuracy**
  - Measurement noise
- **Process Variations**
  - Calibration
- **On-Chip Measurement**
  - Vulnerable to attack
- **Authentication Time**
  - Trojans can be inserted randomly

55

Delay-based Side Channel Analysis

- **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



56

Why does delay side channel analysis can be more effective?

1. No need for activation failures detection
2. Can make detection of distributed traces easier: difference in power consumption is minimal, but delay monitoring is easier. Distribution impacts delay of many different parts so it is better.

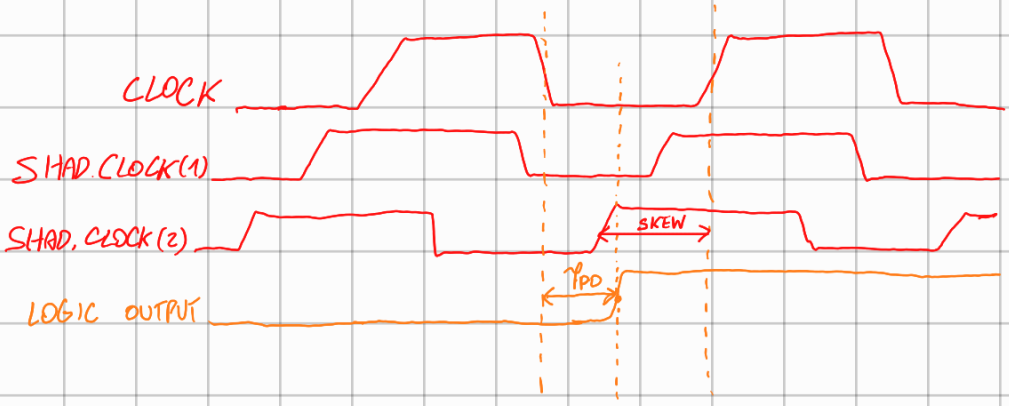
SOLUTION TO ADOPT; REFERENCES TO SLIDE 57. ① ↓

- We have our original circuit (blue) ② triggered by sys clock.

We saw in the past how to work with clock sweeping, changing clock frequencies to identify interval in which prop. delay falls into.

This solution uses a shadow clock that has same frequency of original one, that has been negatively skewed (we add a delay). So only one clock frequency and piece to add skew,

- IDEA: you start skewing clock so you start anticipating the sampling edge of clock and then you sample it,



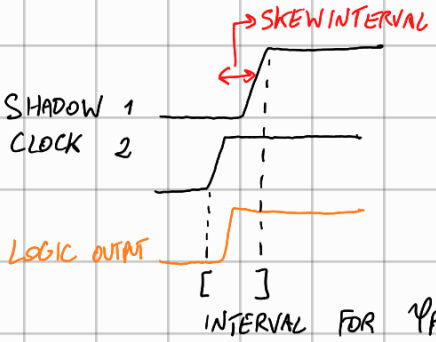
T_{PD} is enough for clock, shad clock ① but not for shad clock ②. You recognise this because comparator will give different values because of different samplings.

clock period

Given skews and T_{CLK} how to find delay?

$T_{PD} \approx T_{CLK} - Skew$ [with some margin of error because I cannot hit the perfect skew]

If you modify skew by adding small chunks (skew-interval), you can give a propagation delay definition in a smaller interval.



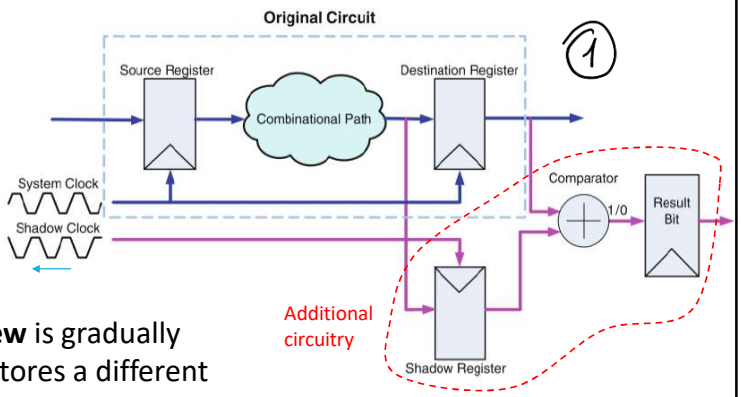
This is a not so expensive approach!

- Limitations:
  - Of course process variation (golden module comparison)
  - With respect to clock sweeping, you have extra area overhead (additional clock, shadow reg., comparator etc.). This for every path to measure.

SMP 59

Delay-based Methods: Shadow Register

- **Shadow-register** provides a possible solution for measuring internal path delay
- The basic unit contains **one shadow registe**r, one comparator and one result register
- **Shadow register clock's negative skew** is gradually increased until the shadow register stores a different (incorrect) value than its main register → we can obtain delay of comb. path as a function of skew amount

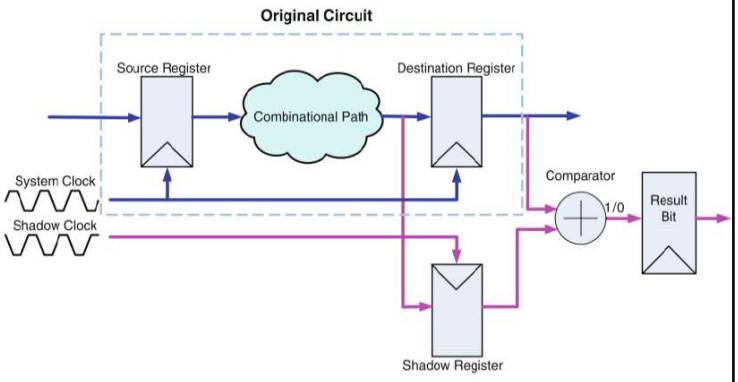


Jie Li, John Lach, "Negative-Skewed Shadow Registers for At-Speed Delay Variation Characterization", 2007 25th International Conference on Computer Design

57

Delay-based Methods: Shadow Register

- Shadow-register provides a possible solution for measuring internal path delay.
- The basic unit contains one shadow register, one comparator and one result register.
- **Limitations**:
  - PV
  - Overhead
  - Shadow-clock (negatively skewed wrt system clock)

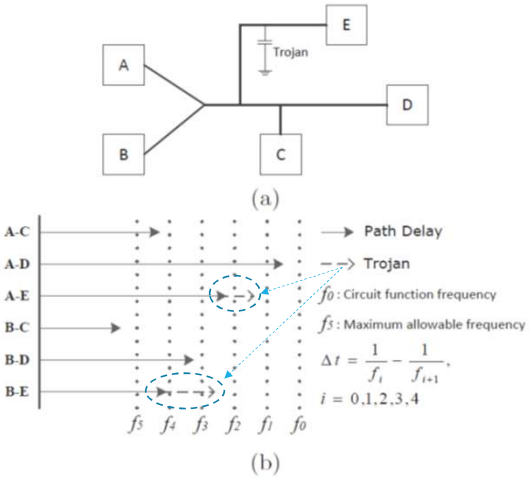


Jie Li, John Lach, "Negative-Skewed Shadow Registers for At-Speed Delay Variation Characterization", 2007 25th International Conference on Computer Design

58

Delay-based Methods: Clock Sweeping

- 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**



59

Delay Analysis -- Challenges

- **Major advantage over power analysis: No activation is required.** } MAIN ADVANTAGE: easier to measure smaller delay differences than power.
- **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 — You still need test inputs to have propagation to measure
  - Path Coverage

You need to stimulate longest path
- **Process Variations (V_{thr}, L, T_{ox})**
  - Impact circuit delay characteristics significantly
  - Differentiate between Trojan and PV
- **Trojan can have impact on multiple paths → advantage** GOOD

60