ETH zürich

Paper Review of ‘A2: Analog Malicious Hardware’

Bastian Schildknecht
Seminar in Computer Architecture 2019
ETH Zürich

A2: Analog Malicious Hardware

Kaiyuan Yang, Matthew Hicks, Qing Dong, Todd Austin, Dennis Sylvester

University of Michigan

Distinguished paper at

IEEE Symposium on Security and Privacy 2016

A2: Analog Malicious Hardware

Kaiyuan Yang, Matthew Hicks, Qing Dong, Todd Austin, Dennis Sylvester
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI, USA
{*kaiyuan, mdhicks, qingdong, austin, dmcs*}*@umich.edu*

***Abstract*—While the move to smaller transistors has been a boon for performance it has dramatically increased the cost to fabricate chips using those smaller transistors. This forces the vast majority of chip design companies to trust a third party—often overseas—to fabricate their design. To guard against shipping chips with errors (intentional or otherwise) chip design companies rely on post-fabrication testing. Unfortunately, this type of testing leaves the door open to malicious modifications since attackers can craft attack triggers requiring a sequence of unlikely events, which will never be encountered by even the most diligent tester.**

In this paper, we show how a fabrication-time attacker can leverage analog circuits to create a hardware attack that is small (i.e., requires as little as one gate) and stealthy (i.e., requires an unlikely trigger sequence before effecting a chip's functionality). In the open spaces of an already placed and routed design, we construct a circuit that uses capacitors to siphon charge from nearby wires as they transition between digital values. When the capacitors fully charge, they deploy an attack that forces a victim flip-flop to a desired value. We weaponize this attack into a remotely-controllable privilege escalation by attaching the capacitor to a wire controllable and by selecting a victim flip-flop that holds the privilege bit for our processor. We implement this attack in an OR1200 processor and fabricate a chip. Experimental results show that our attacks work, show that our attacks elude activation by a diverse set of benchmarks, and suggest that our attacks evade known defenses.

***Keywords*-analog; attack; hardware; malicious; security; Trojan;**

I. Introduction

Hardware is the base of a system. All software executes on top of a processor. That software must trust that the hardware faithfully implements the specification. For many types of hardware flaws, software has no way to check if something went wrong [1], [2]. Even worse, if there is an attack in hardware, it can contaminate all layers of a system that depend on that hardware—violating high-level security policies correctly implemented by software.

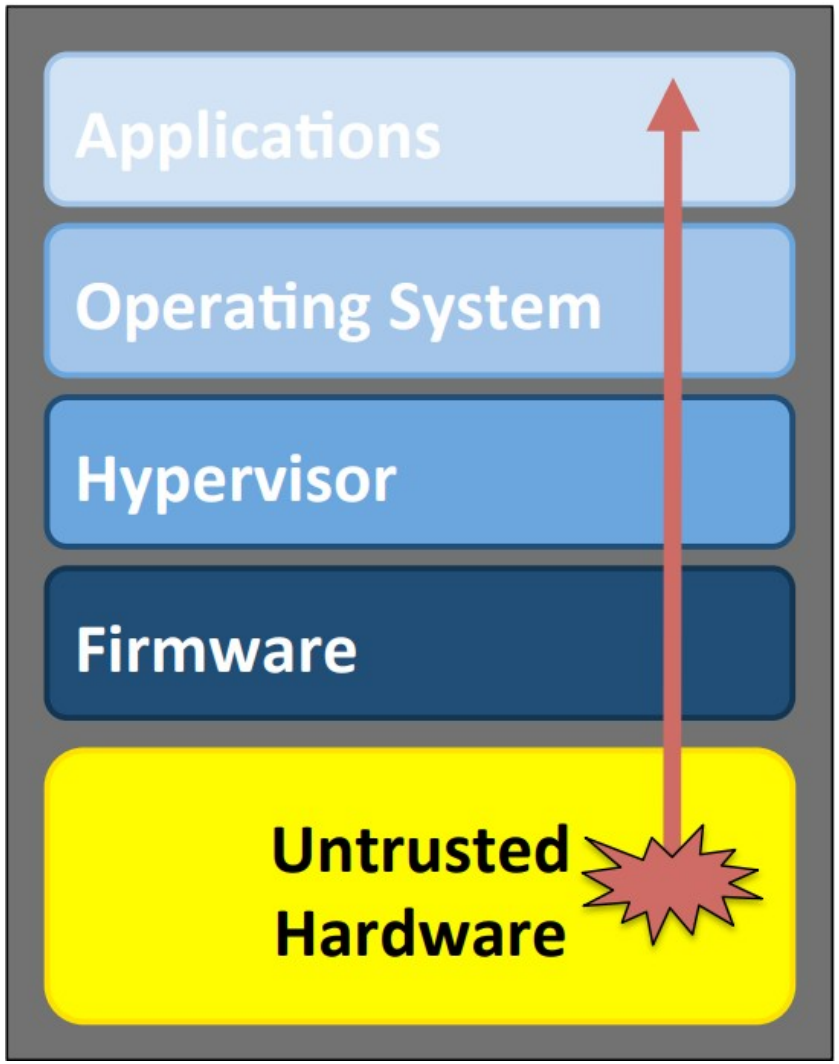
The trend of smaller transistors while beneficial for increased performance and lower power, has made fabricating a chip expensive. With every generation of transistor comes the cost of retooling for that smaller transistor. For example, it costs 15% more to setup the fabrication line for each successive process node and by 2020 it is expected that setting-up a fabrication line for the smallest transistor size will require a $20,000,000,000 upfront investment [3]. To amortize the cost of the initial tooling required to support a given transistor size, most hardware companies outsource fabrication.

Outsourcing of chip fabrication opens-up hardware to attack. The most pernicious fabrication-time attack is the dopant-level Trojan [4], [5]. Dopant-level Trojans convert trusted circuitry into malicious circuitry by changing the dopant ratio on the input pins to victim transistors. This effectively ties the input of the victim transistors to a logic level 0 or 1—a short circuit. Converting existing circuits makes dopant-level Trojans very difficult to detect since there are no added or removed gates or wires. In fact, detecting dopant-level Trojans requires a complete chip delayering and comprehensive imaging with a scanning electron microscope [6]. Unfortunately, this elusiveness comes at the cost of expressiveness. Dopant-level Trojans are limited by existing circuits, making it difficult to implement sophisticated attack triggers [5]. The lack of a sophisticated trigger means that dopant-level Trojans are more detectable by post-fabrication functional testing. Thus, dopant-level Trojans represent an extreme on a tradeoff space between detectability during physical inspection and detectability during testing.

To defend against malicious hardware inserted during fabrication, researchers have proposed two fundamental defenses: 1) use side-channel information (e.g., power and temperature) to characterize acceptable behavior in an effort to detect anomalous (i.e., malicious) behavior [7]–[10] and 2) add sensors to the chip that measure and characterize features of the chip's behavior (e.g., signal propagation delay) in order to identify dramatic changes in those features (presumably caused by activation of a malicious circuit) [11]–[13]. Using side channels as a defense works well against large Trojans added to purely combinational circuits where it is possible to test all inputs and there exists a reference chip to compare against. While this accurately describes most existing fabrication-time attacks, we show that it is possible to implement a stealthy and powerful processor attack using only a single added gate. Adding sensors to the design would seem to adapt the side-channel approach to more complex, combinational circuits, but we design an attack that operates in the analog domain until it directly modifies processor

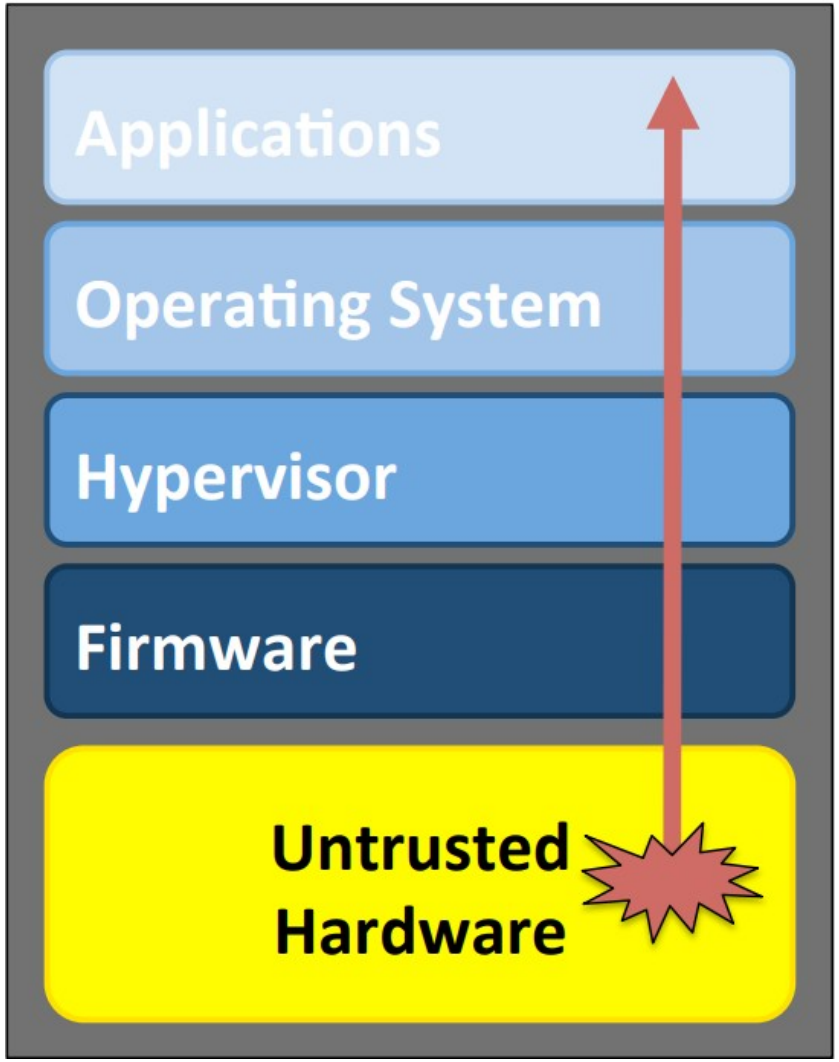
Background

Why Secure Hardware Matters



A system with insecure hardware means an insecure system

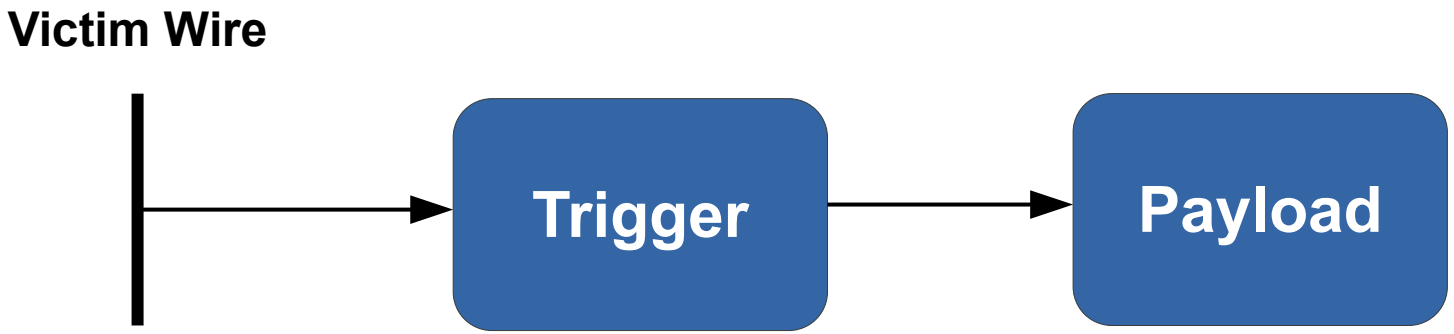
Why Secure Hardware Matters



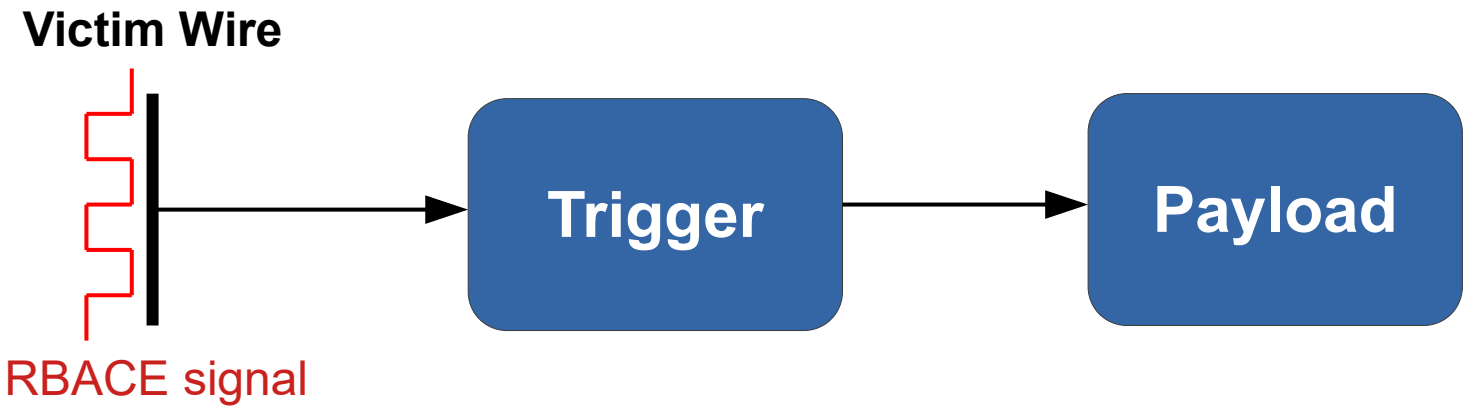
A system with insecure hardware means an insecure system

How does a hardware attack work?

Typical Trigger Based Hardware Attacks

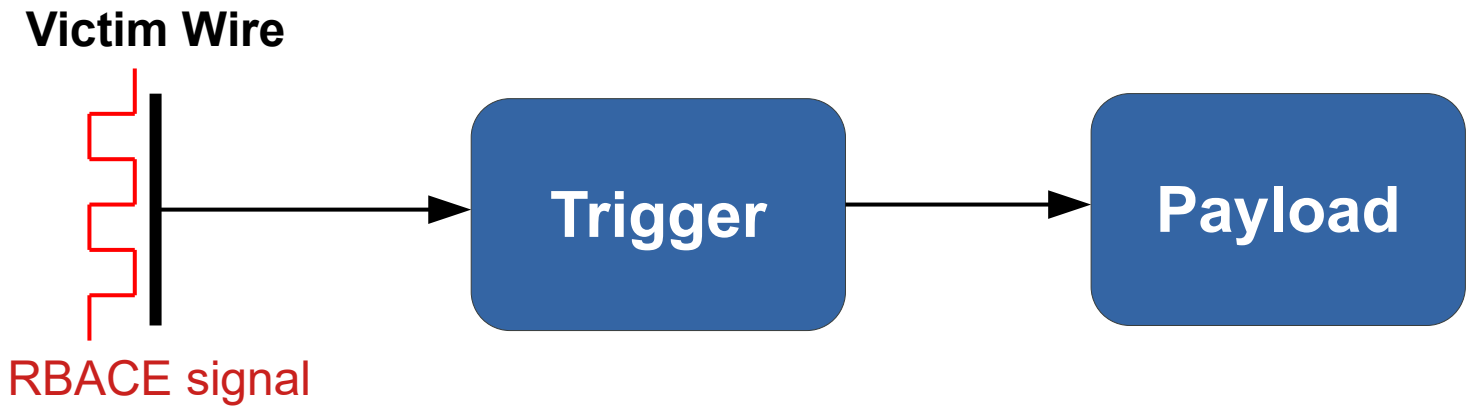


Typical Trigger Based Hardware Attacks



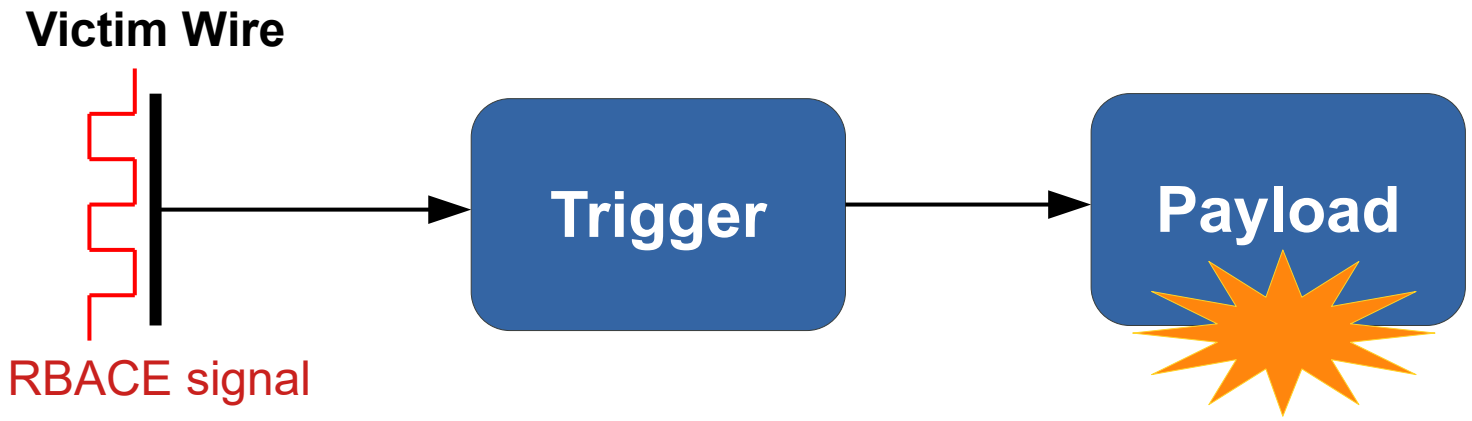
Rare, **B**ut **A**ttacker **C**ontrollable **E**vent

Typical Trigger Based Hardware Attacks



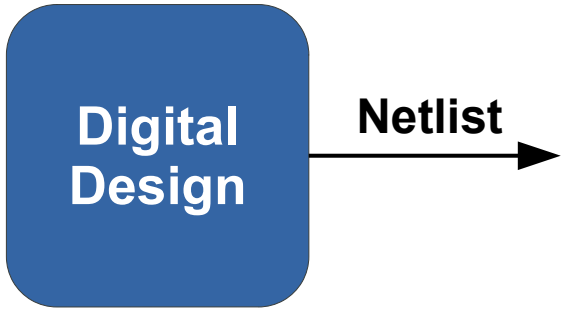
Rare, **B**ut **A**ttacker **C**ontrollable **E**vent

Typical Trigger Based Hardware Attacks

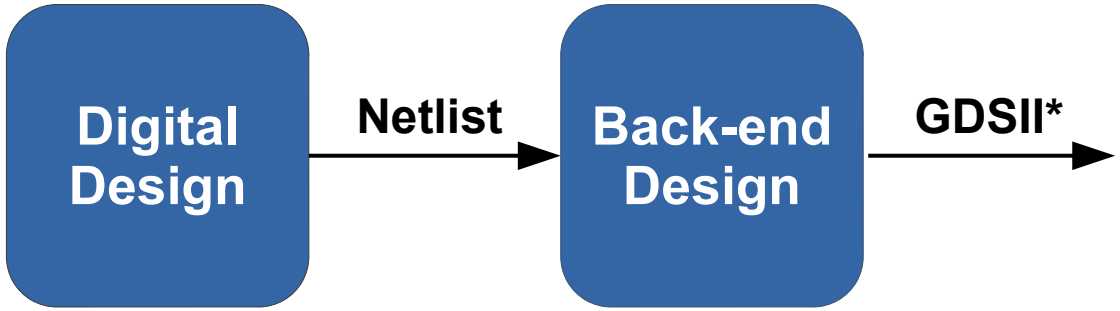


Rare, **B**ut **A**ttacker **C**ontrollable **E**vent

When to Implement a Hardware Attack

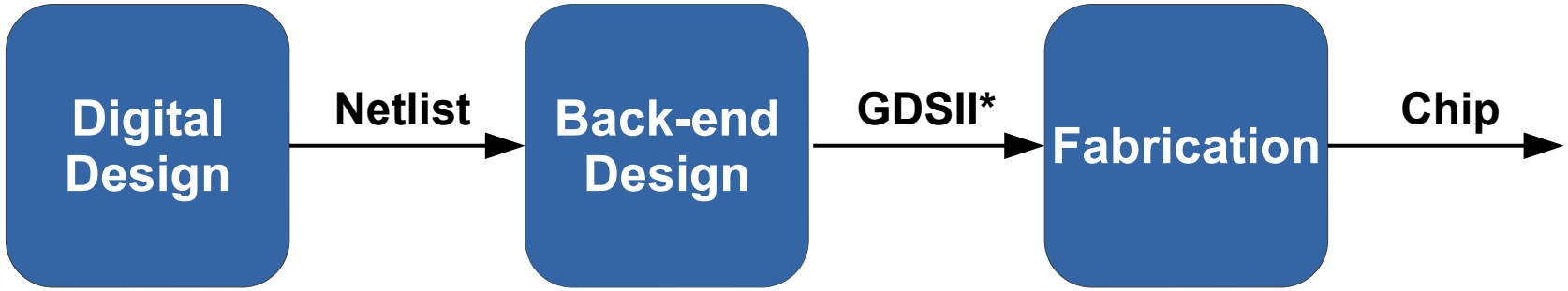


When to Implement a Hardware Attack



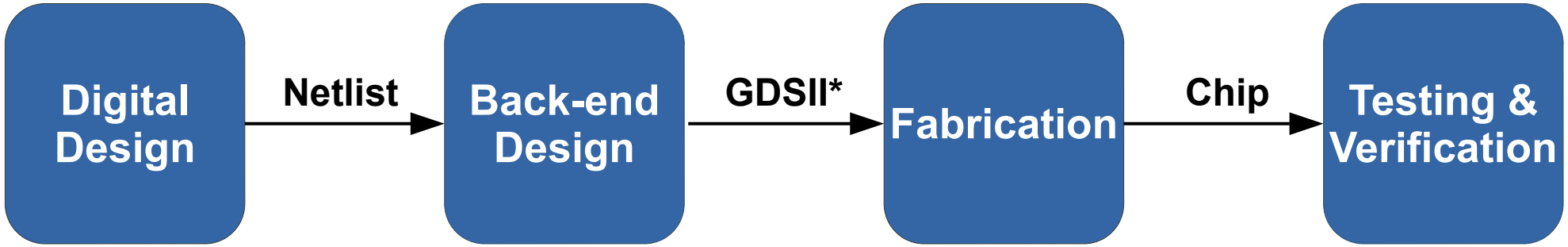
***G**raphic **D**atabase **S**ystem **II** file

When to Implement a Hardware Attack



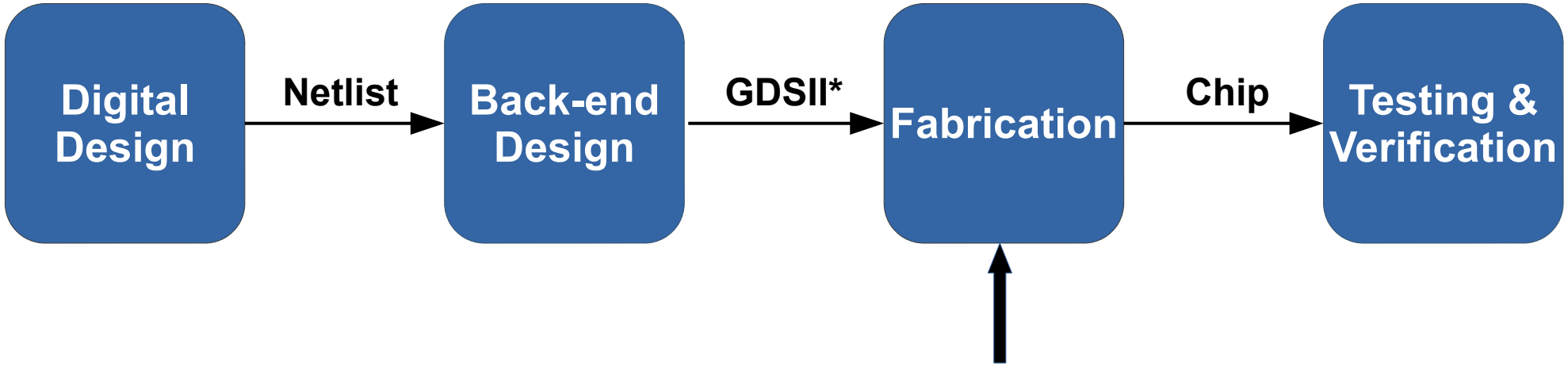
***G**raphic **D**atabase **S**ystem **II** file

When to Implement a Hardware Attack



***G**raphic **D**atabase **S**ystem **II** file

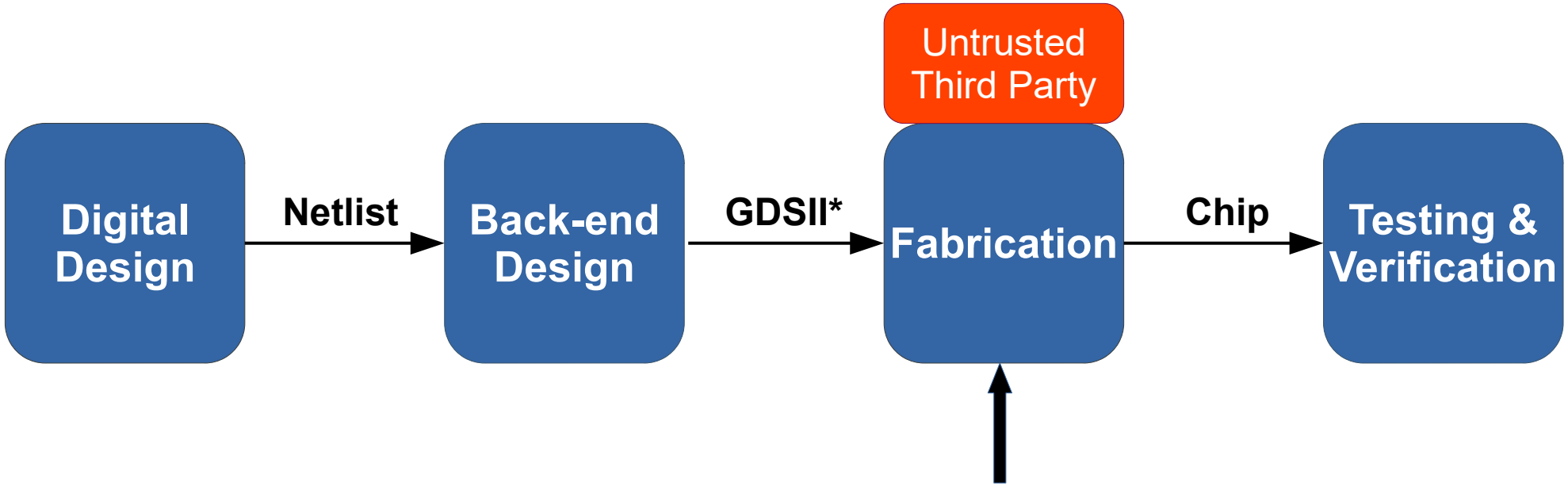
When to Implement a Hardware Attack



Expected to cost **$20,000,000,000** by **2020** for the smallest technology node

***G**raphic **D**atabase **S**ystem **II** file

When to Implement a Hardware Attack



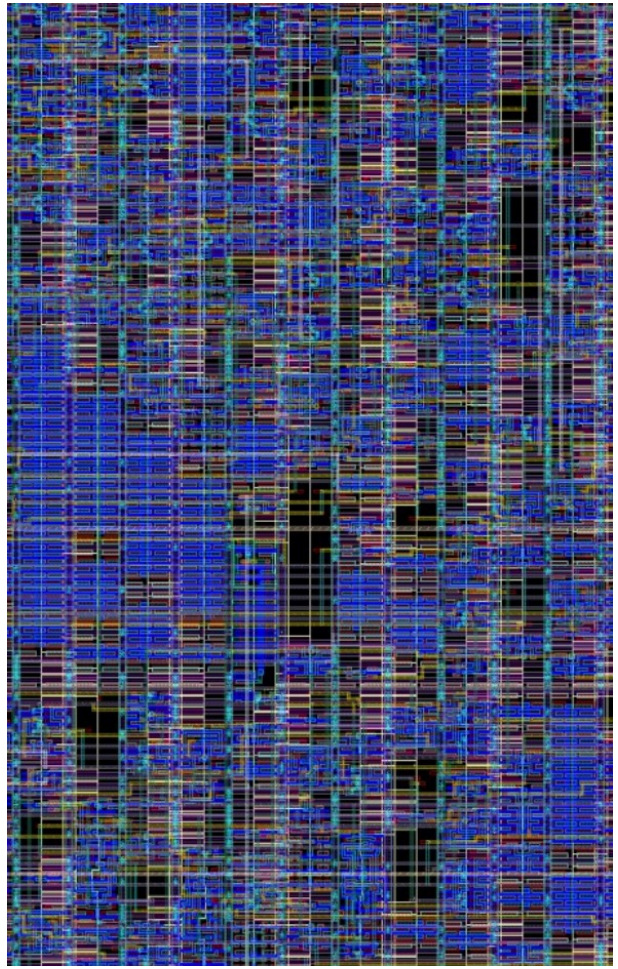
Expected to cost **$20,000,000,000** by **2020** for the smallest technology node

***G**raphic **D**atabase **S**ystem **II** file

Where to Put a Hardware Attack

Where to Put a Hardware Attack

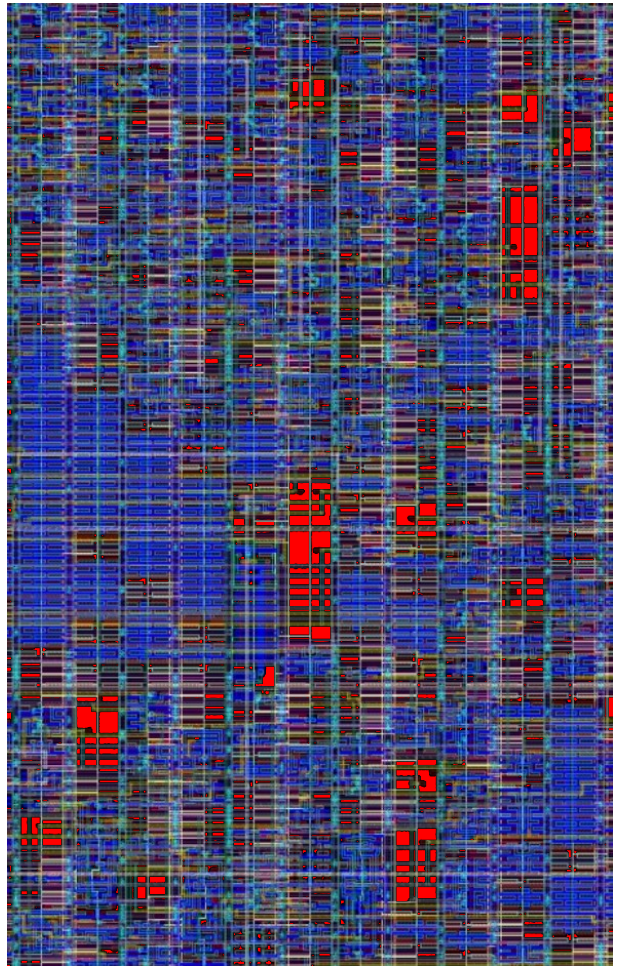
20-30% of chip area is unused



Example GDSII layout with free space

Where to Put a Hardware Attack

20-30% of chip area is unused

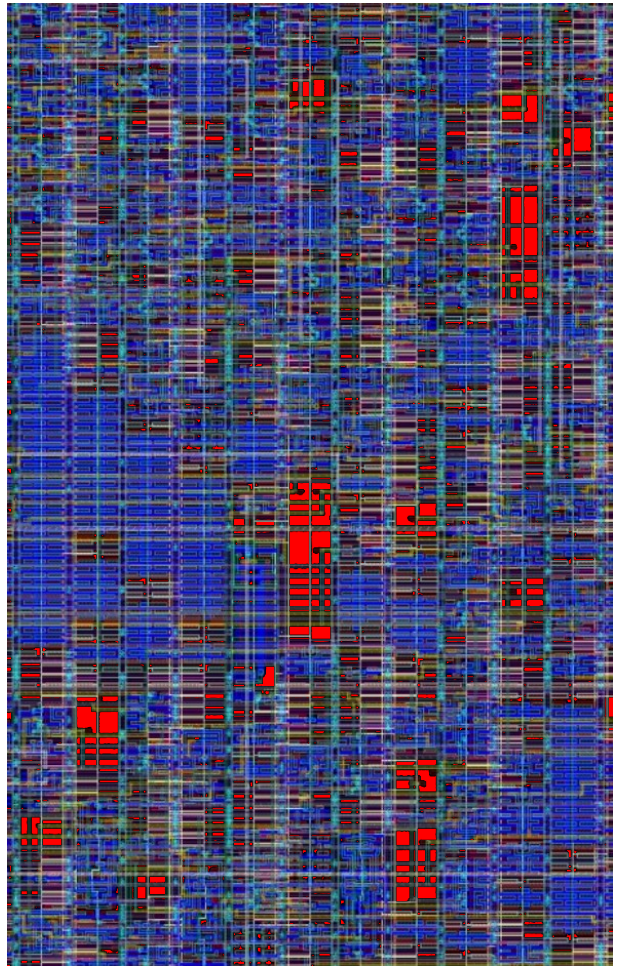


Example GDSII layout with free space

Where to Put a Hardware Attack

20-30% of chip area is unused

Mostly caused by routing constraints



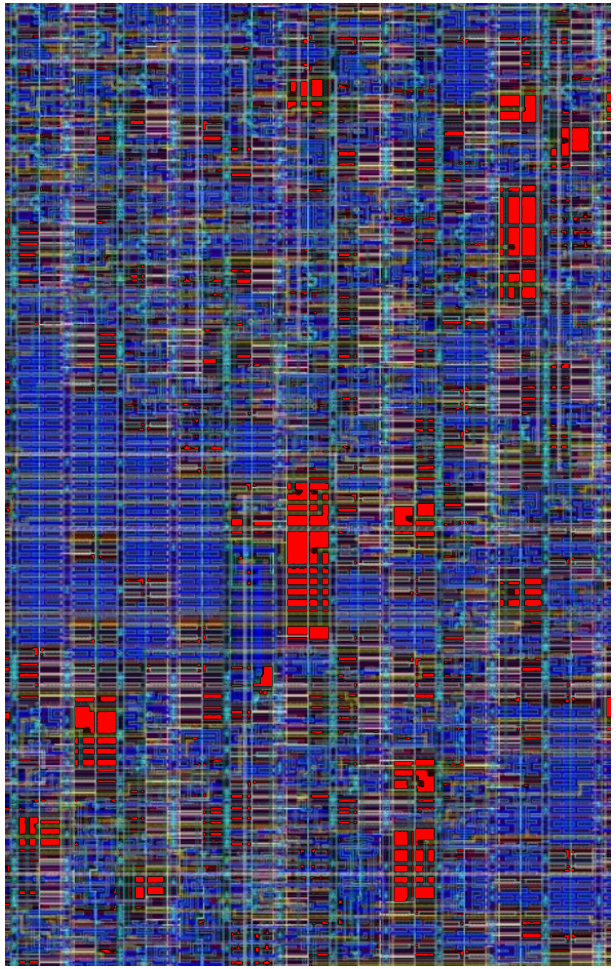
Example GDSII layout with free space

Where to Put a Hardware Attack

20-30% of chip area is unused

Mostly caused by routing constraints

Opens up possibility for attackers to embed malicious hardware



Example GDSII layout with free space

Problem

Issues With Existing Hardware Attacks

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Process Reliability Trojans

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Process Reliability Trojans

Modify the fabrication process to cause the entire chip to fail early

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Process Reliability Trojans

Modify the fabrication process to cause the entire chip to fail early

Not controllable

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Process Reliability Trojans

Modify the fabrication process to cause the entire chip to fail early

Not controllable

Dopant-Level Trojans

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Process Reliability Trojans

Modify the fabrication process to cause the entire chip to fail early

Not controllable

Dopant-Level Trojans

Change behaviour of existing circuits by tying logic gates to logic 0 or 1

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Process Reliability Trojans

Modify the fabrication process to cause the entire chip to fail early

Not controllable

Dopant-Level Trojans

Change behaviour of existing circuits by tying logic gates to logic 0 or 1

Not controllable and not stealthy

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Process Reliability Trojans

Modify the fabrication process to cause the entire chip to fail early

Not controllable

Dopant-Level Trojans

Change behaviour of existing circuits by tying logic gates to logic 0 or 1

Not controllable and not stealthy

Parametric Trojans for Fault Injection

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Process Reliability Trojans

Modify the fabrication process to cause the entire chip to fail early

Not controllable

Dopant-Level Trojans

Change behaviour of existing circuits by tying logic gates to logic 0 or 1

Not controllable and not stealthy

Parametric Trojans for Fault Injection

Same as dopant-level trojans but rely on voltage fluctuations as a trigger

Issues With Existing Hardware Attacks

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Process Reliability Trojans

Modify the fabrication process to cause the entire chip to fail early

Not controllable

Dopant-Level Trojans

Change behaviour of existing circuits by tying logic gates to logic 0 or 1

Not controllable and not stealthy

Parametric Trojans for Fault Injection

Same as dopant-level trojans but rely on voltage fluctuations as a trigger

Not remotely controllable

Can We Do Better?

Is there a better hardware attack that does not suffer from these issues?

Can We Do Better?

Is there a better hardware attack that does not suffer from these issues?

How can it work?

Goal

Goal of the Paper

Goal: Design a hardware attack that is

Goal of the Paper

Goal: Design a hardware attack that is

Very small

Goal of the Paper

Goal: Design a hardware attack that is

Very small

Controllable

Goal of the Paper

Goal: Design a hardware attack that is

Very small

Controllable

Stealthy

Threat Model for the Attack

Goal

Threat Model for the Attack

Goal

- Attack implemented at time of **fabrication**

Threat Model for the Attack

Goal

- Attack implemented at time of **fabrication**
- The attacker has only access to a **correctly implemented GDSII** file

Threat Model for the Attack

Goal

- Attack implemented at time of **fabrication**
- The attacker has only access to a **correctly implemented GDSII** file
- The attacker **cannot change dimensions** or move stuff around

Threat Model for the Attack

Goal

- Attack implemented at time of **fabrication**
- The attacker has only access to a **correctly implemented GDSII** file
- The attacker **cannot change dimensions** or move stuff around
- The attacker has **no knowledge over tests** conducted on the chip

Novelty

Previous Approaches of Hardware Attacks

Novelty

Digital Domain Hardware Attacks

Rely on triggers based on tens to hundreds of logic gates

Not very small and not stealthy

Process Reliability Trojans

Modify the fabrication process to cause the entire chip to fail early

Not controllable

Dopant-Level Trojans

Change behaviour of existing circuits by tying logic gates to logic 0 or 1

Not controllable and not stealthy

Parametric Trojans for Fault Injection

Same as dopant-level trojans but rely on voltage fluctuations as a trigger

Not remotely controllable

Enter The A2 Attack

Novelty

Enter The A2 Attack

The A2 attack uses analog behaviour to mitigate these issues!

Enter The A2 Attack

The A2 attack uses analog behaviour to mitigate these issues!

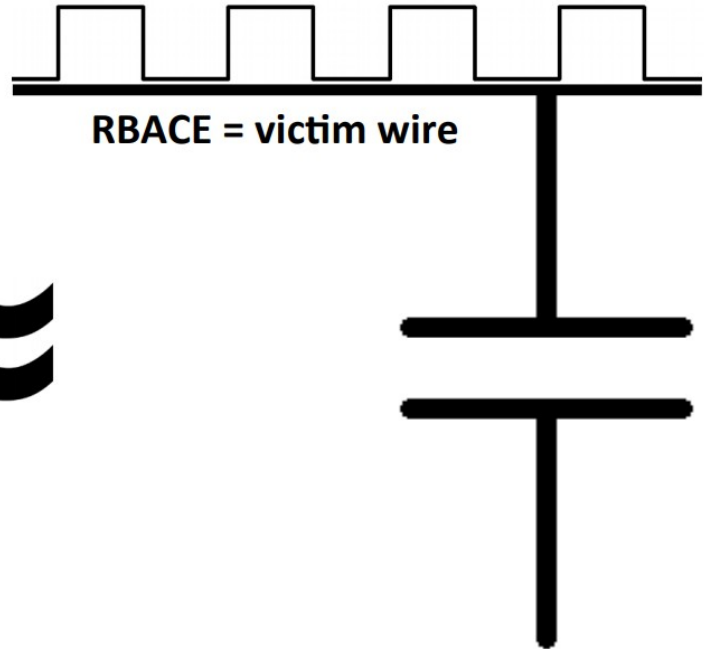
```
on_every(RBACE) do
   if(count == 12345) then
      do_attack()
   else
      count = count + 1
done
```

Enter The A2 Attack

The A2 attack uses analog behaviour to mitigate these issues!

```
on_every(RBACE) do
   if(count == 12345) then
      do_attack()
   else
      count = count + 1
done
```

≈

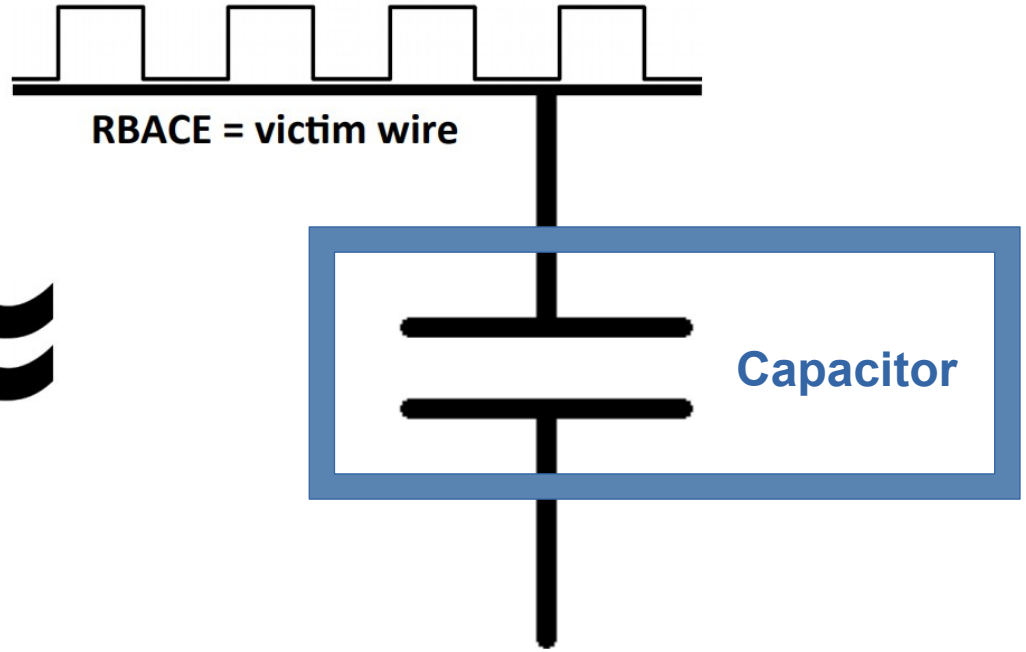


Enter The A2 Attack

The A2 attack uses analog behaviour to mitigate these issues!

```
on_every(RBACE) do
   if(count == 12345) then
      do_attack()
   else
      count = count + 1
done
```

≈

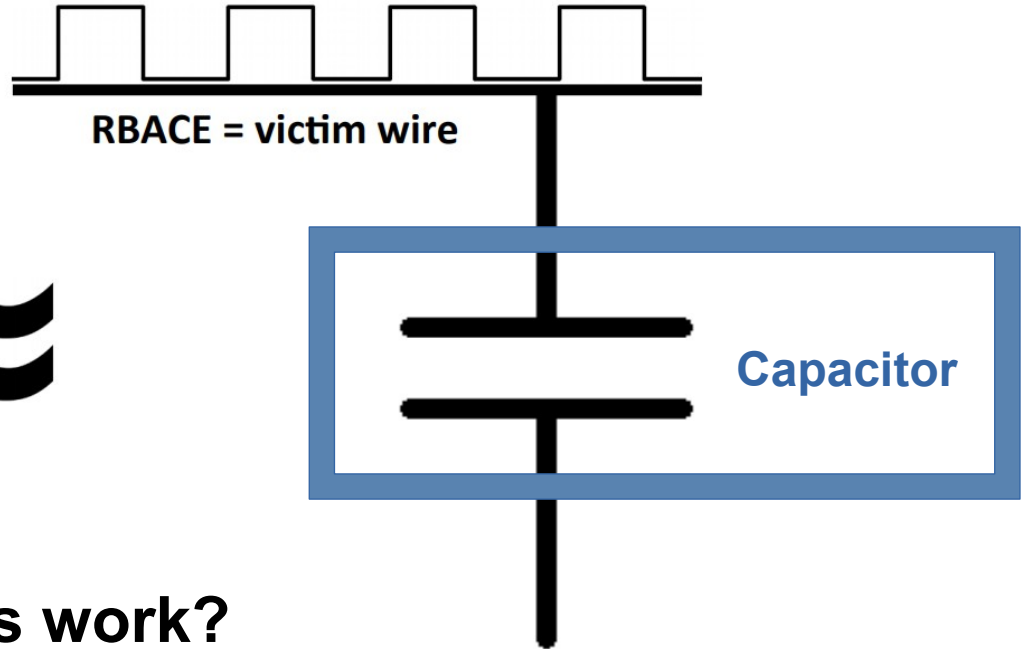


Enter The A2 Attack

The A2 attack uses analog behaviour to mitigate these issues!

```
on_every(RBACE) do
   if(count == 12345) then
      do_attack()
   else
      count = count + 1
done
```

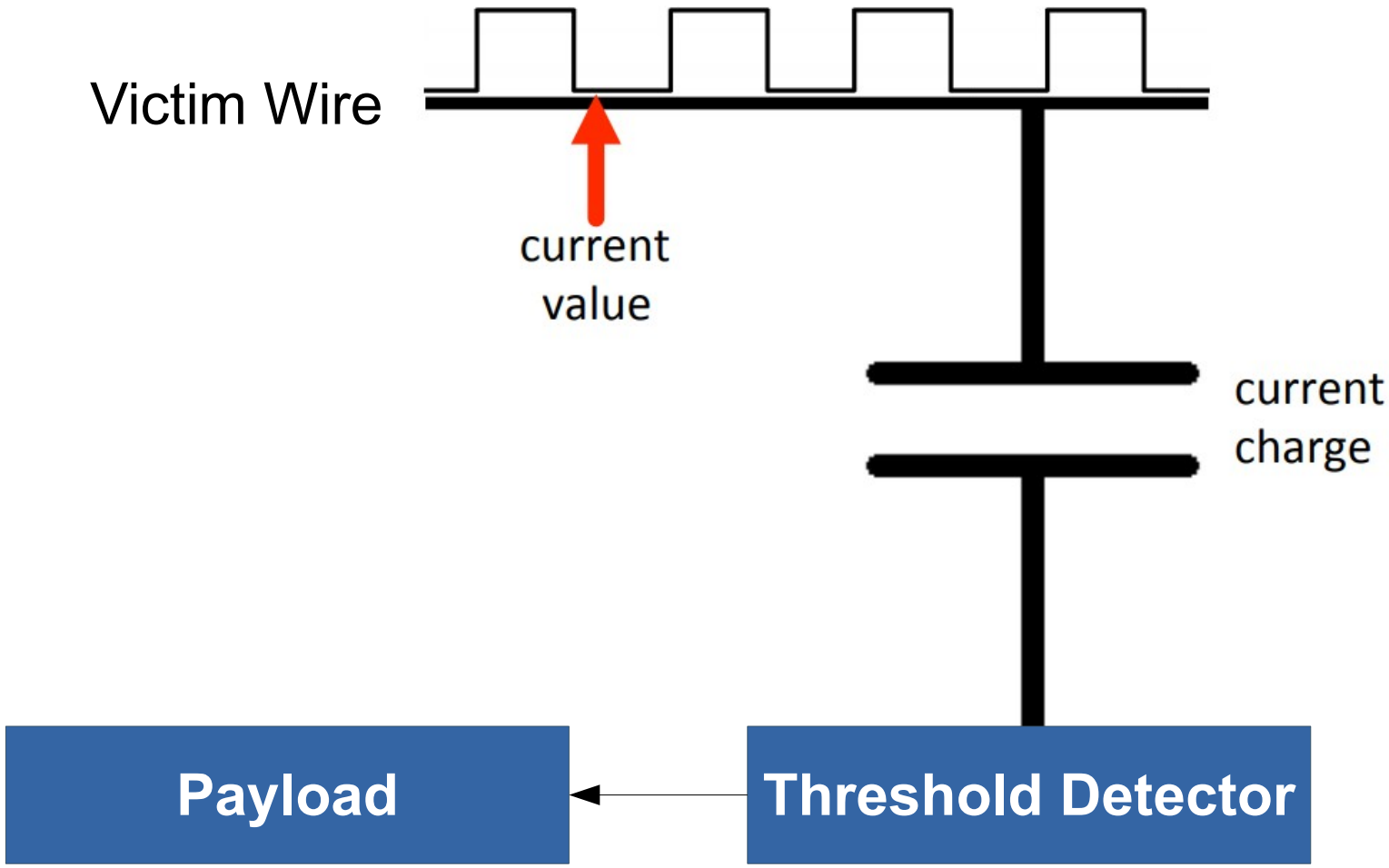
≈



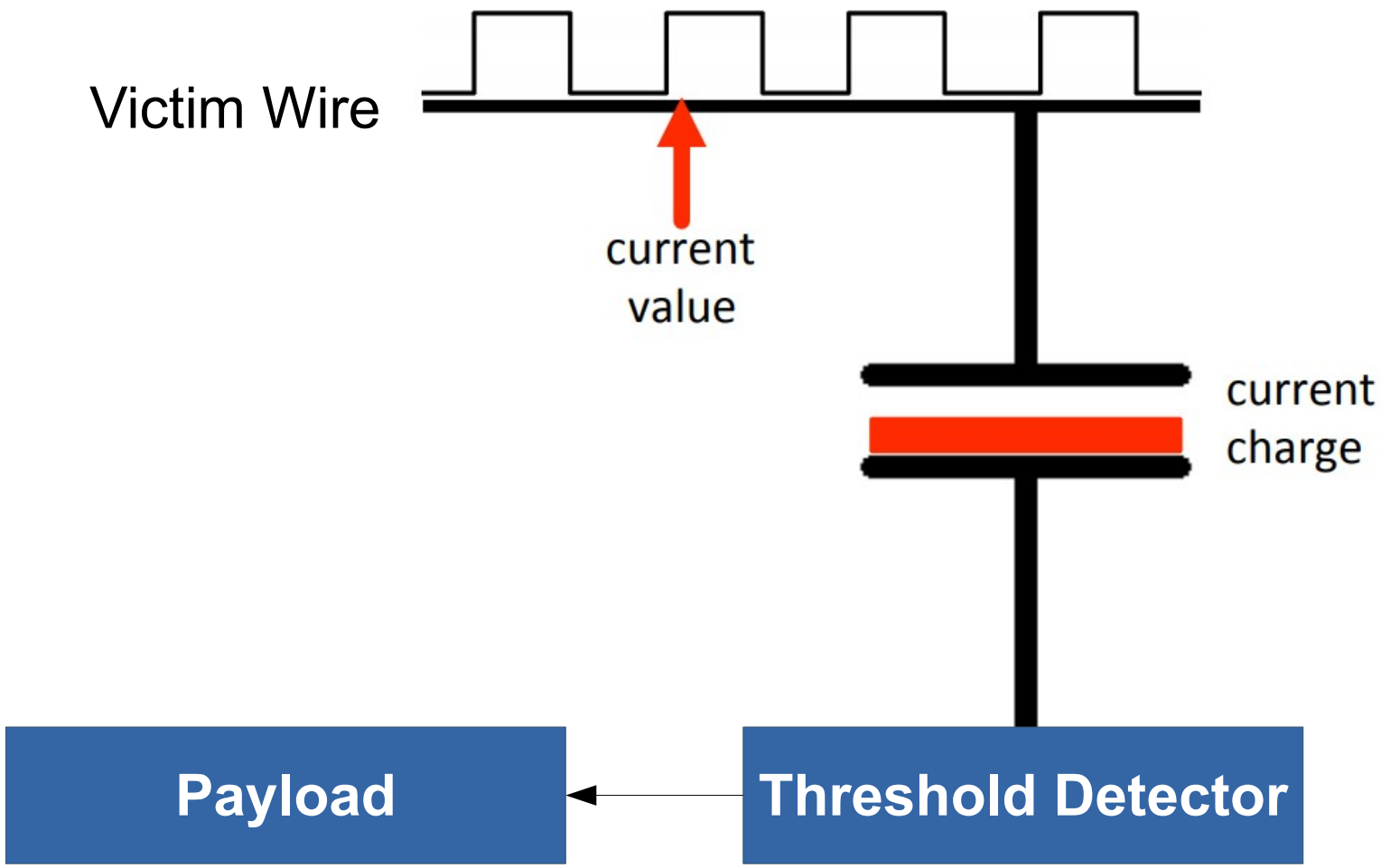
How does this work?

Key Approach & Ideas

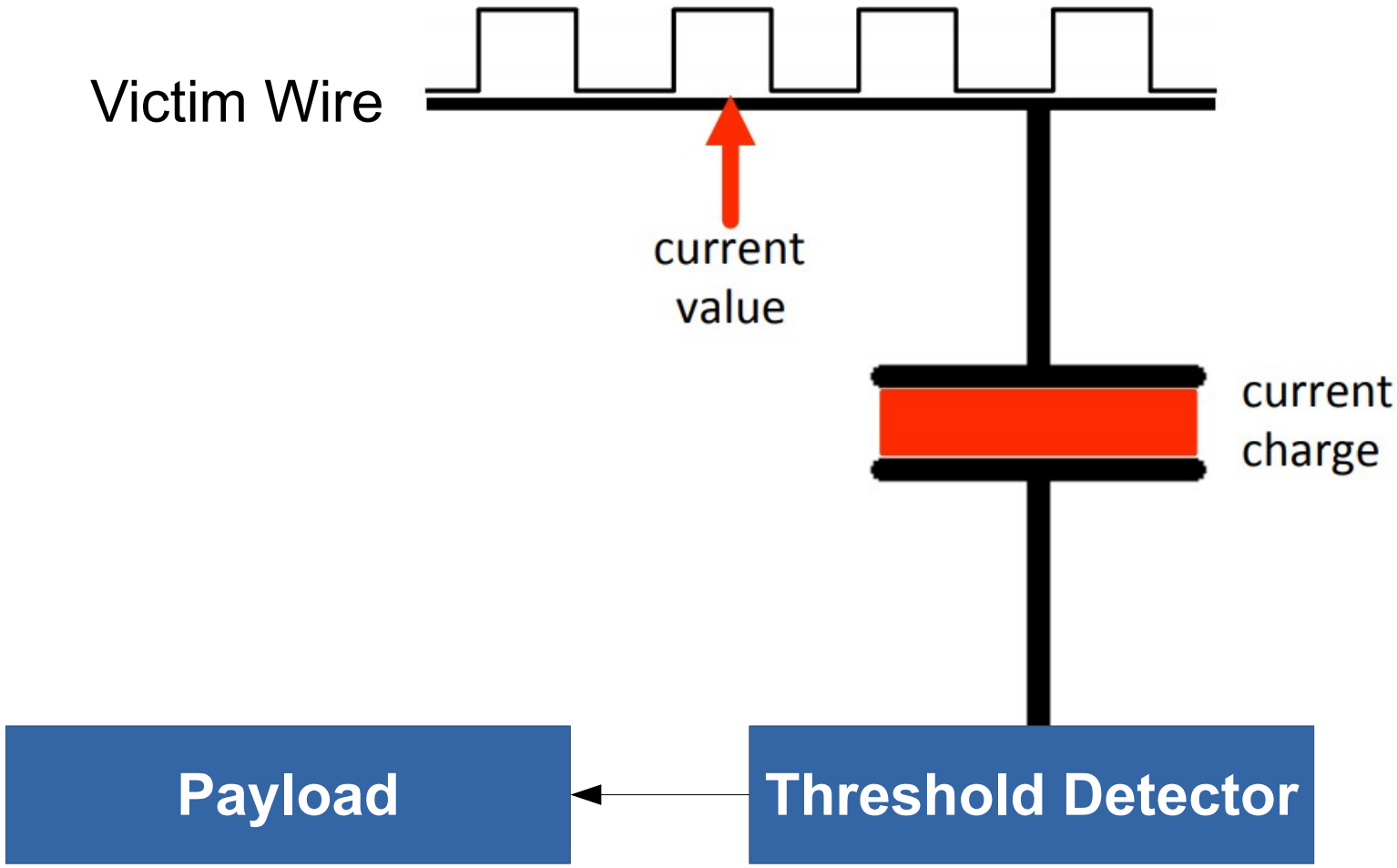
How does A2 work?



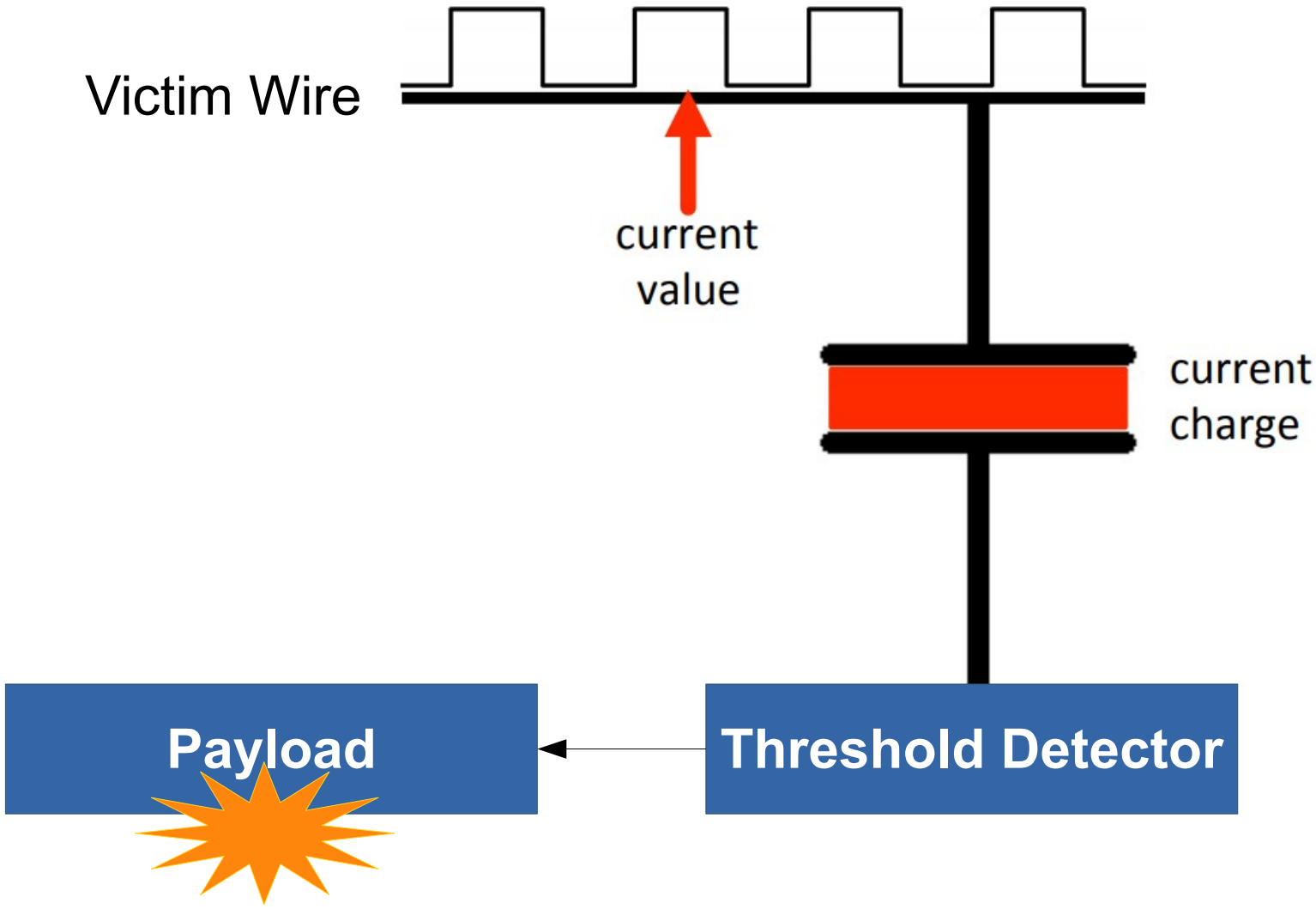
How does A2 work?



How does A2 work?

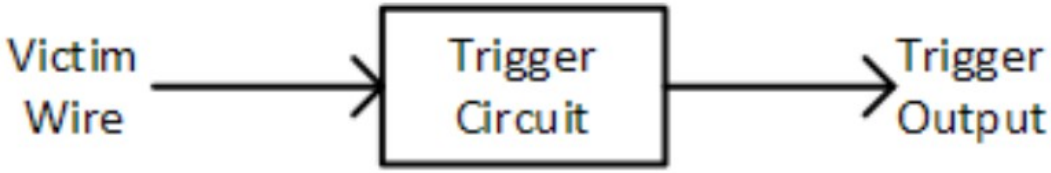


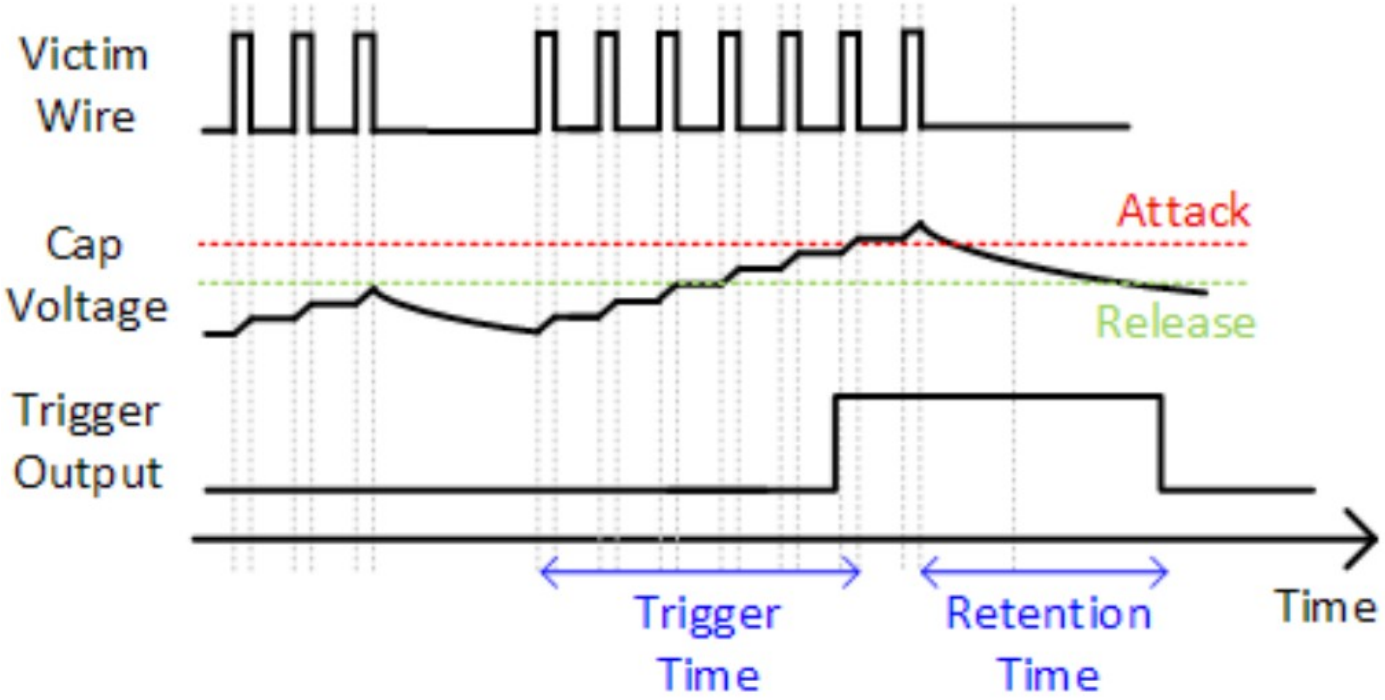
How does A2 work?



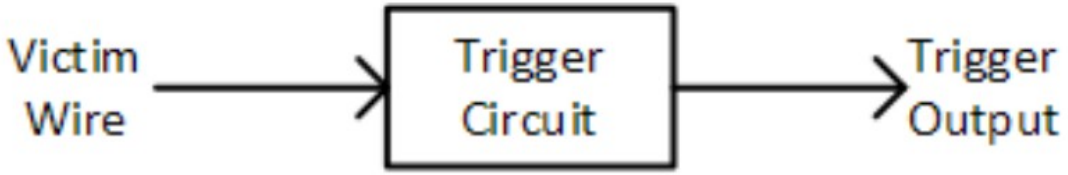
Mechanism in Detail

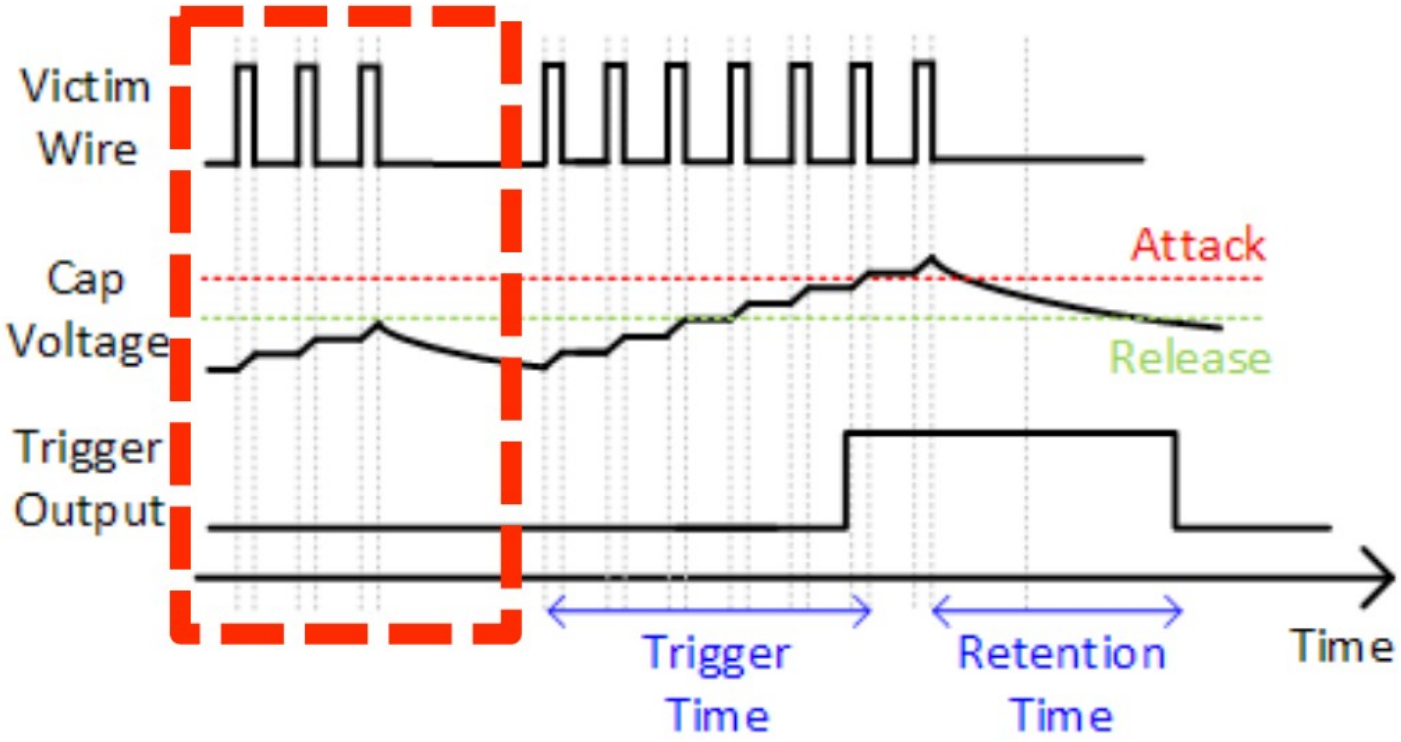
The Analog Trigger Circuit



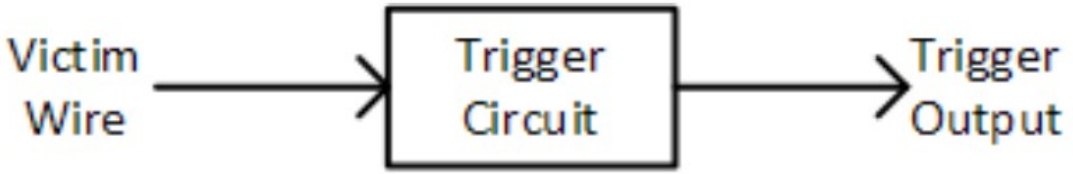


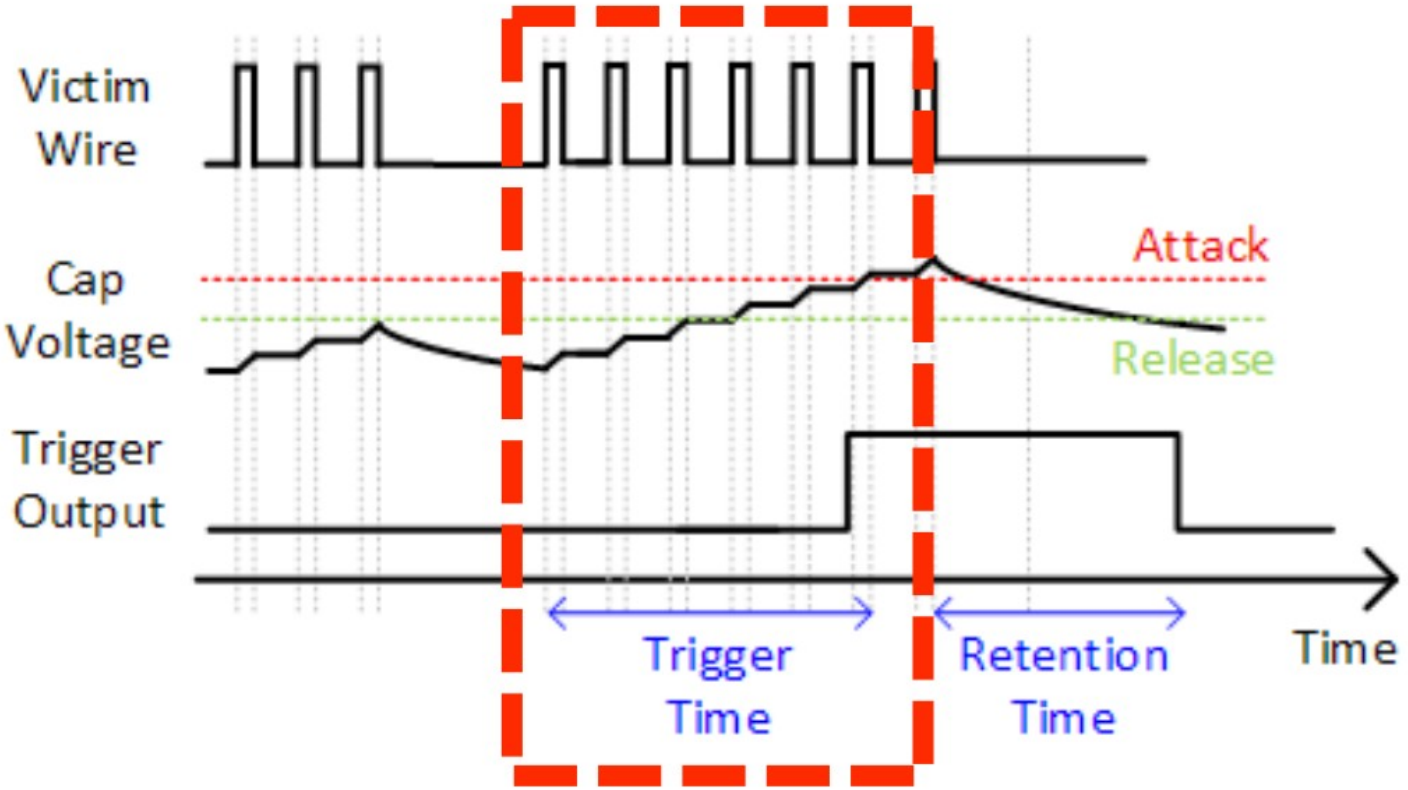
The Analog Trigger Circuit



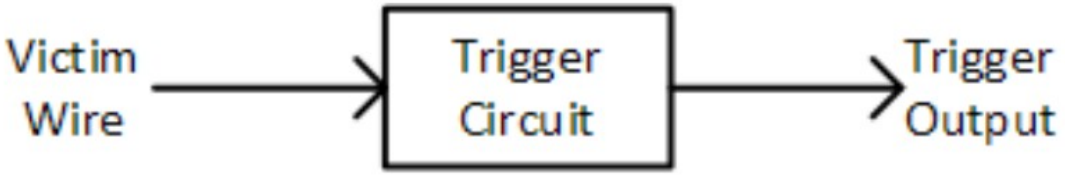


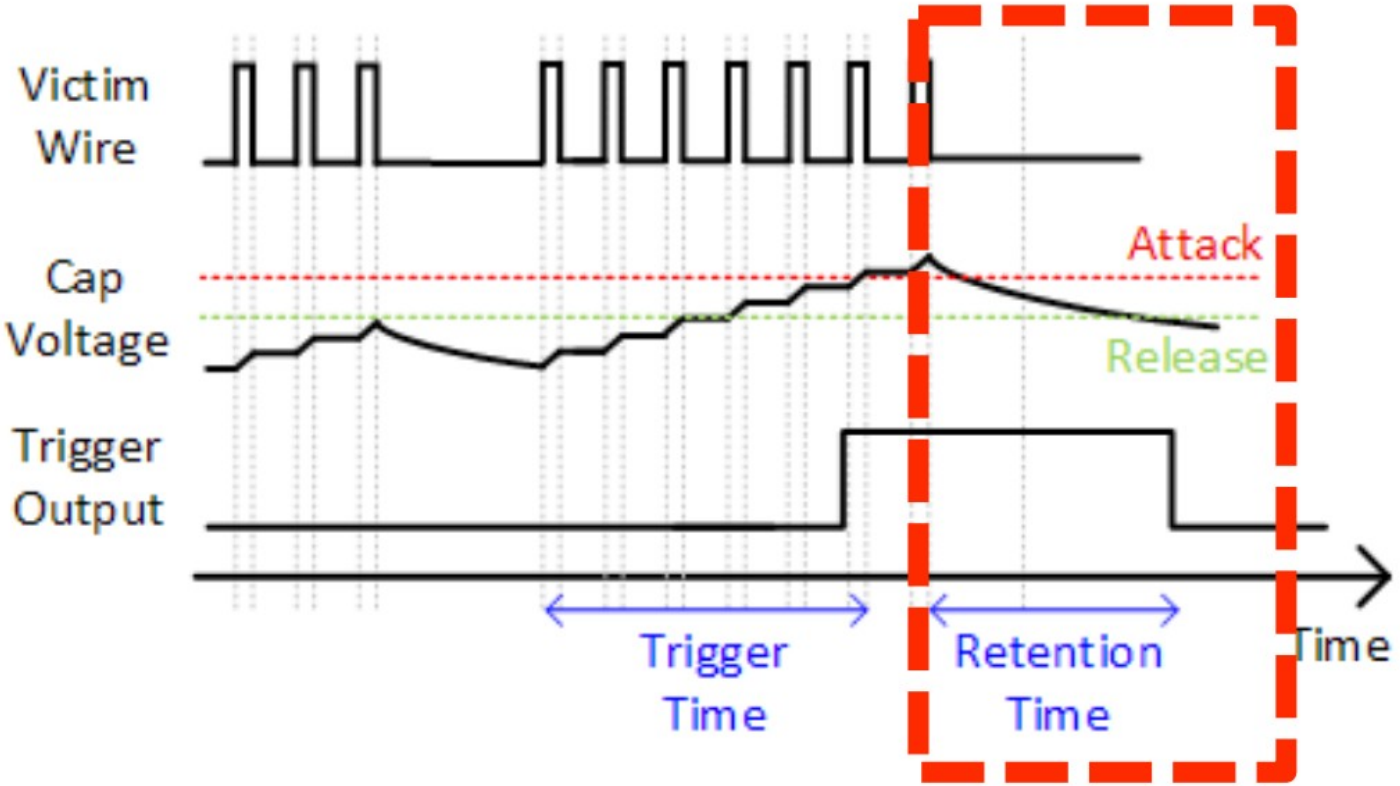
The Analog Trigger Circuit





The Analog Trigger Circuit

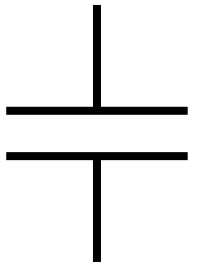




Design Challenge: Single Capacitor

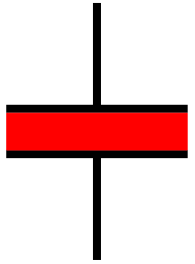
Design Challenge: Single Capacitor

Small capacitors charge up to quickly



Design Challenge: Single Capacitor

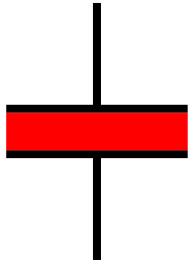
Small capacitors charge up to quickly



Design Challenge: Single Capacitor

Small capacitors charge up to quickly

- This results in the attack being too easy to trigger

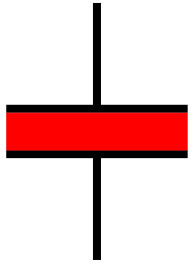


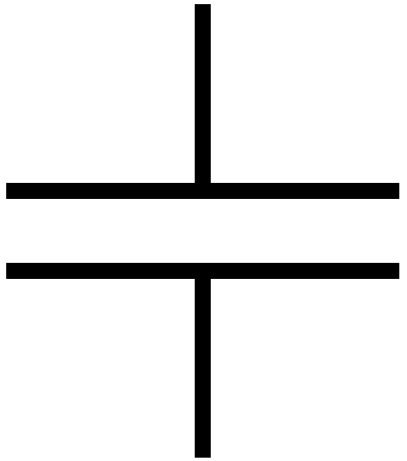
Design Challenge: Single Capacitor

Small capacitors charge up to quickly

- This results in the attack being too easy to trigger

Large capacitors induce current spikes



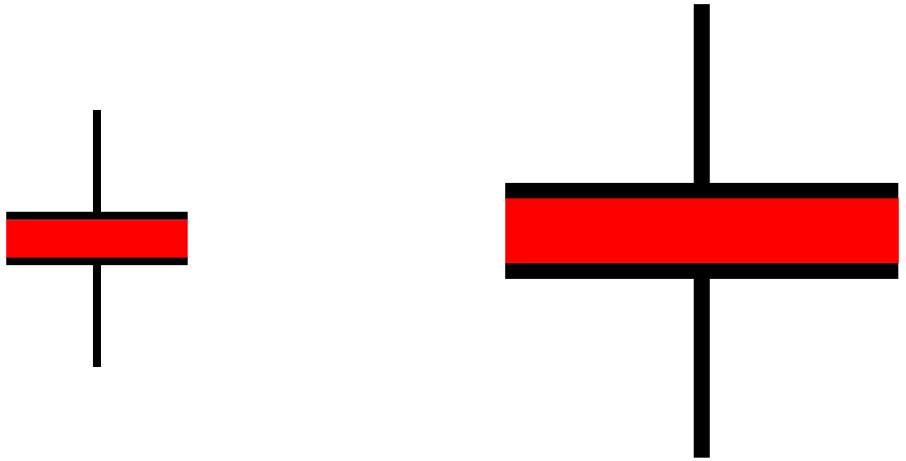


Design Challenge: Single Capacitor

Small capacitors charge up to quickly

- This results in the attack being too easy to trigger

Large capacitors induce current spikes



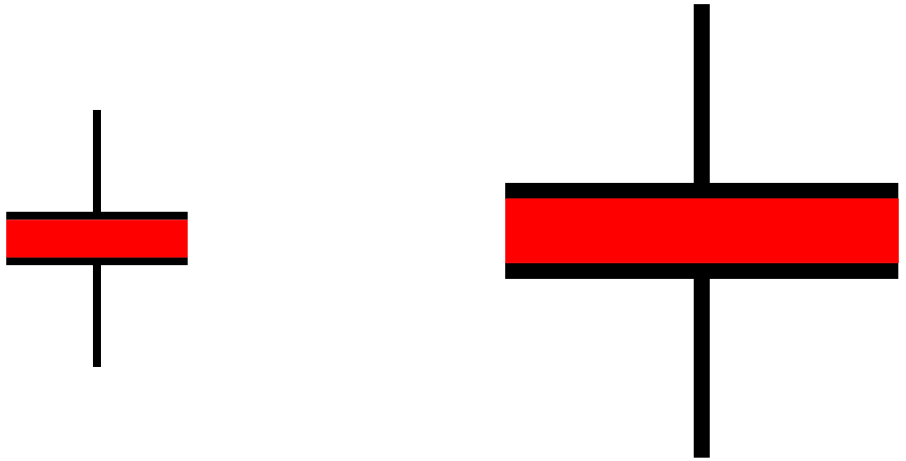
Design Challenge: Single Capacitor

Small capacitors charge up to quickly

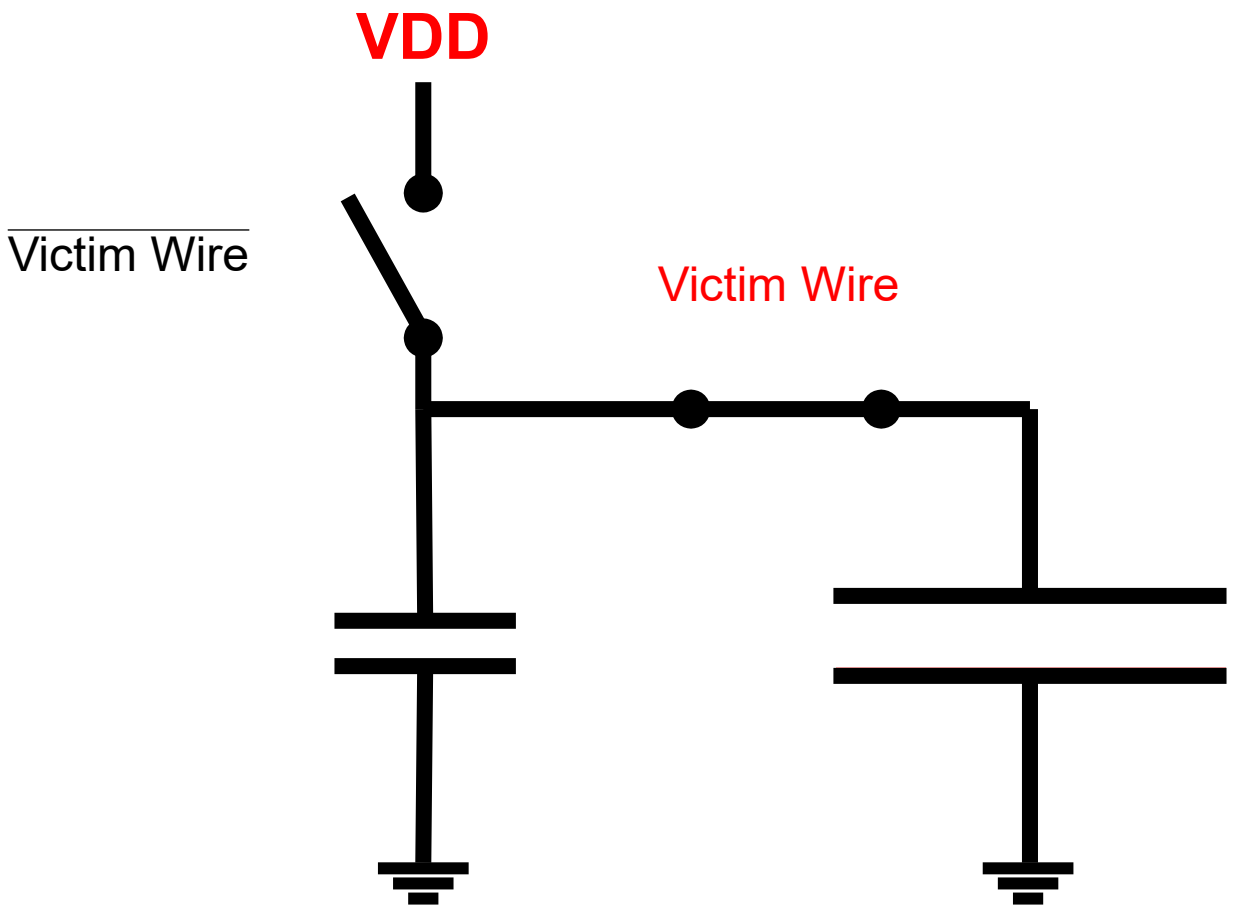
- This results in the attack being too easy to trigger

Large capacitors induce current spikes

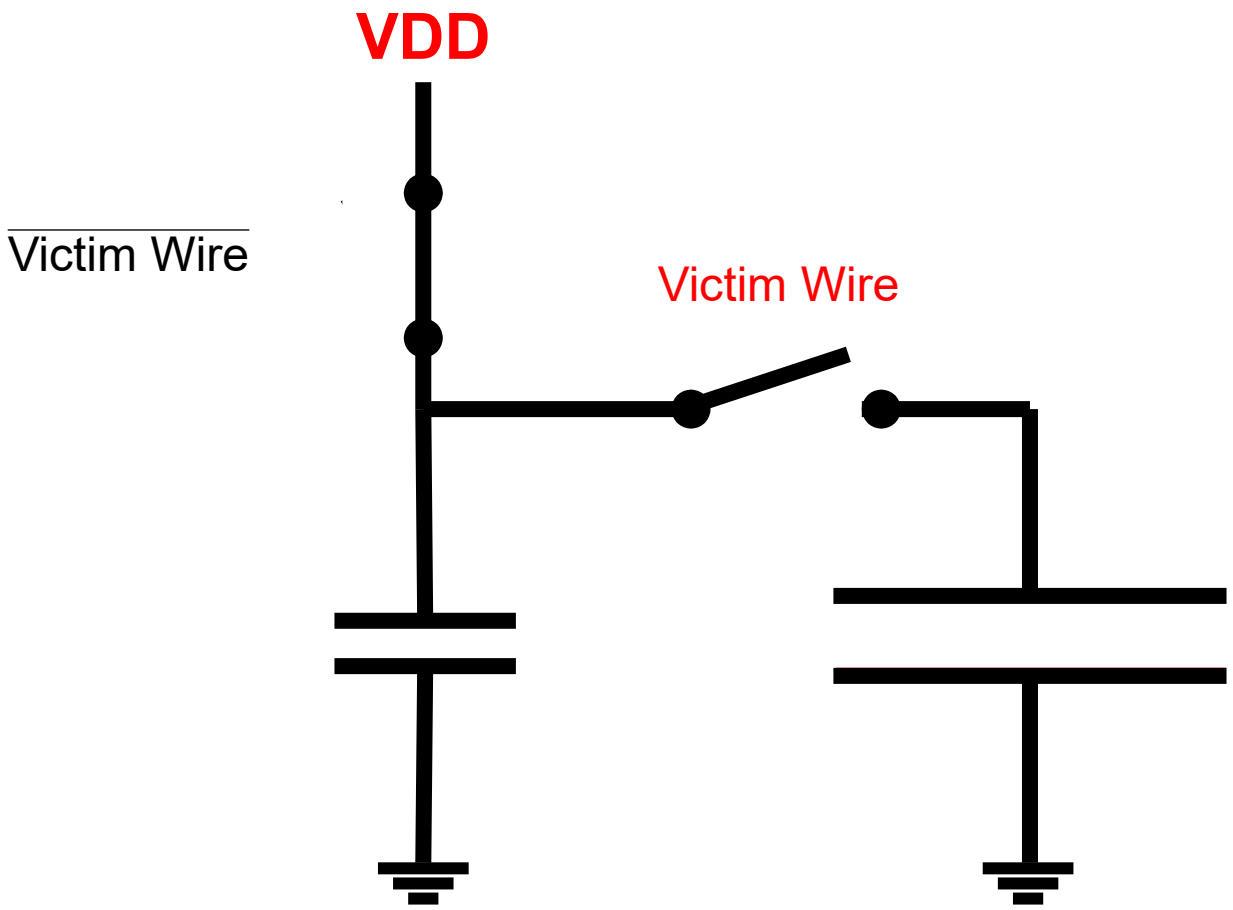
- This makes it also easier to detect



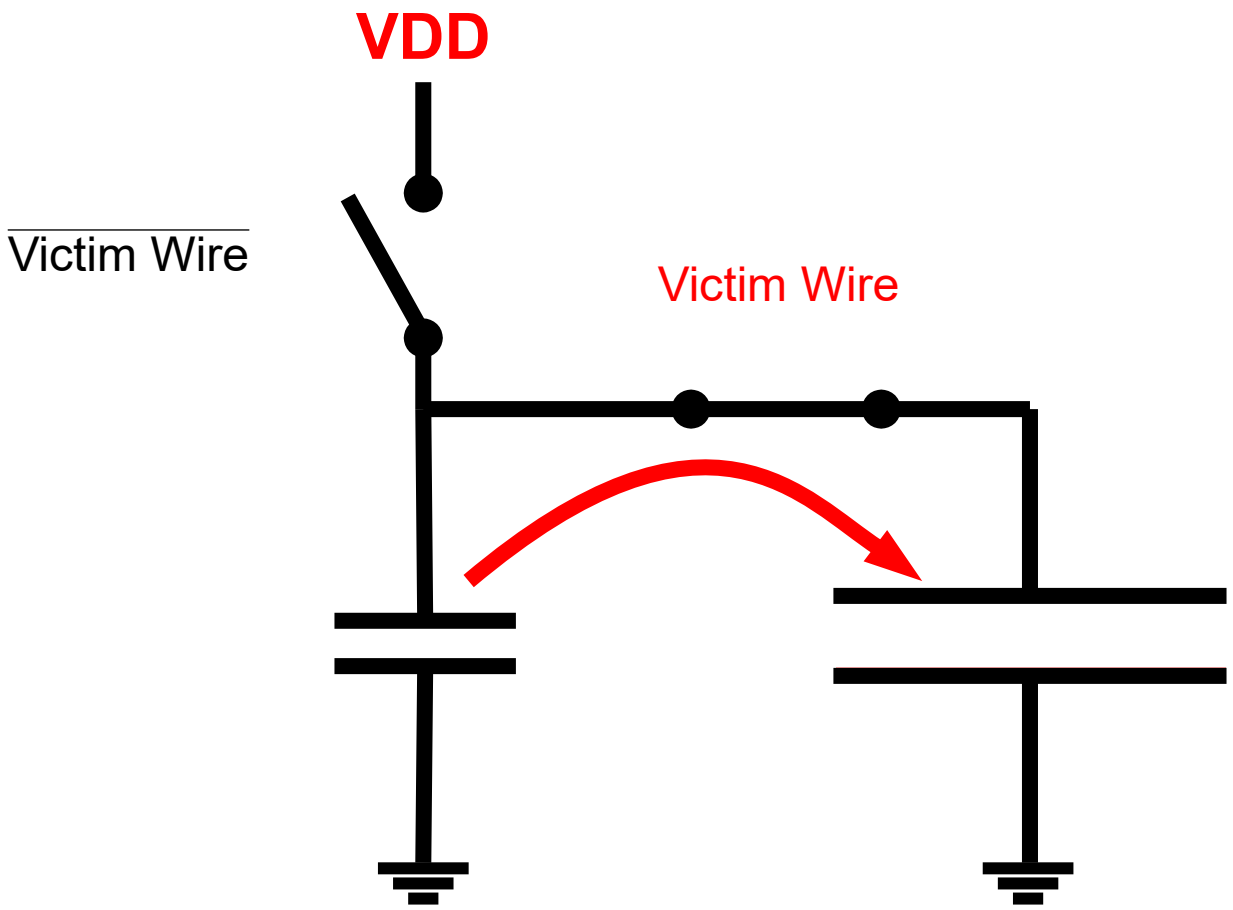
Charge Sharing



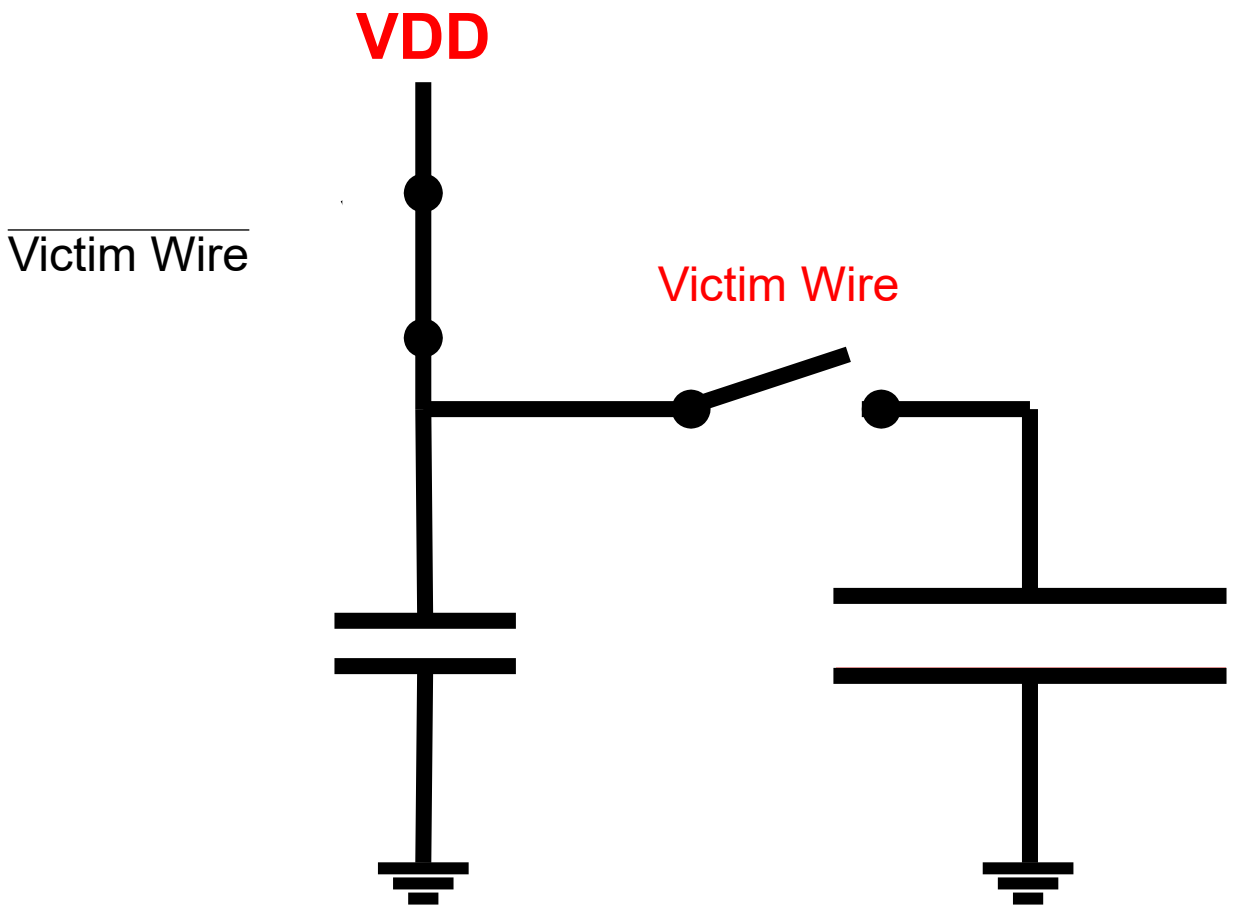
Charge Sharing



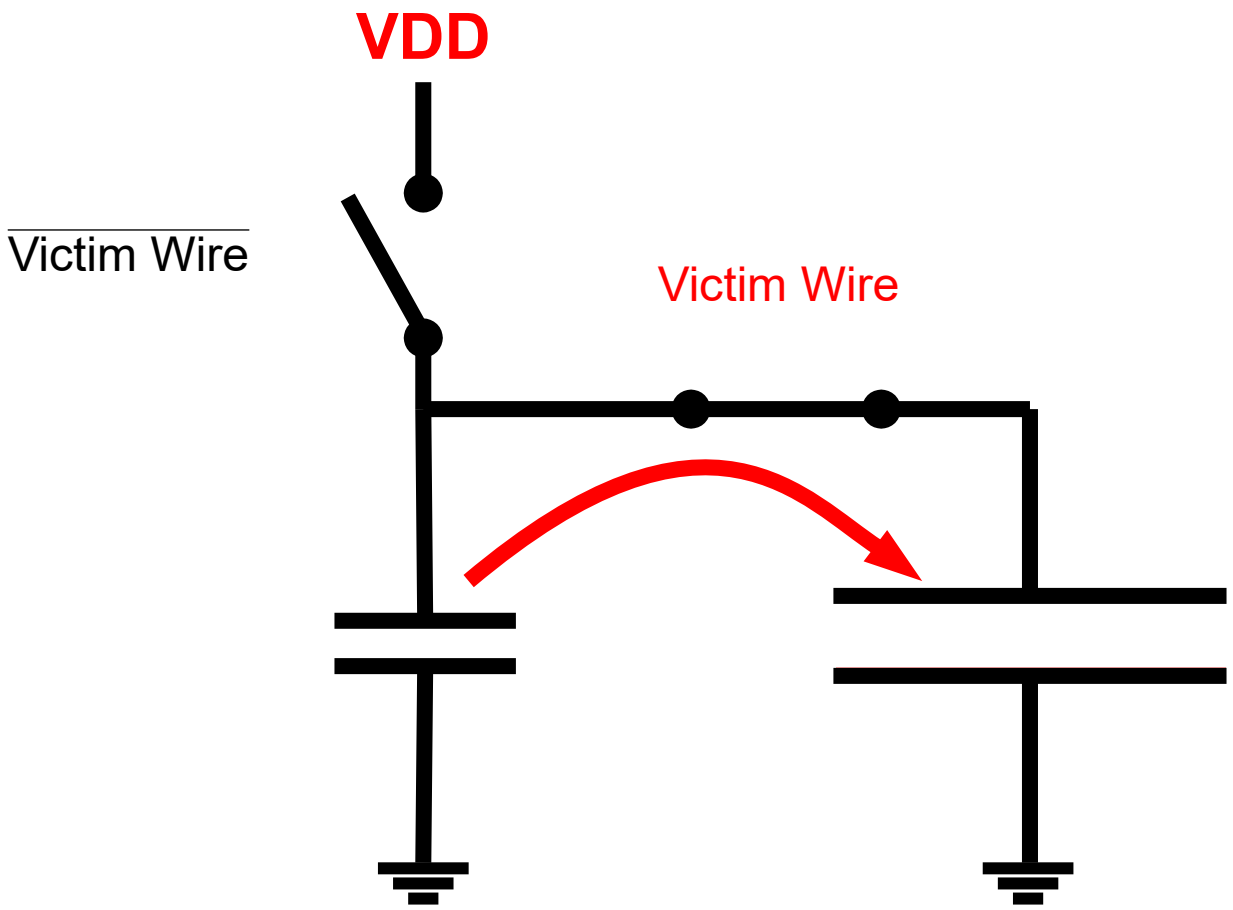
Charge Sharing



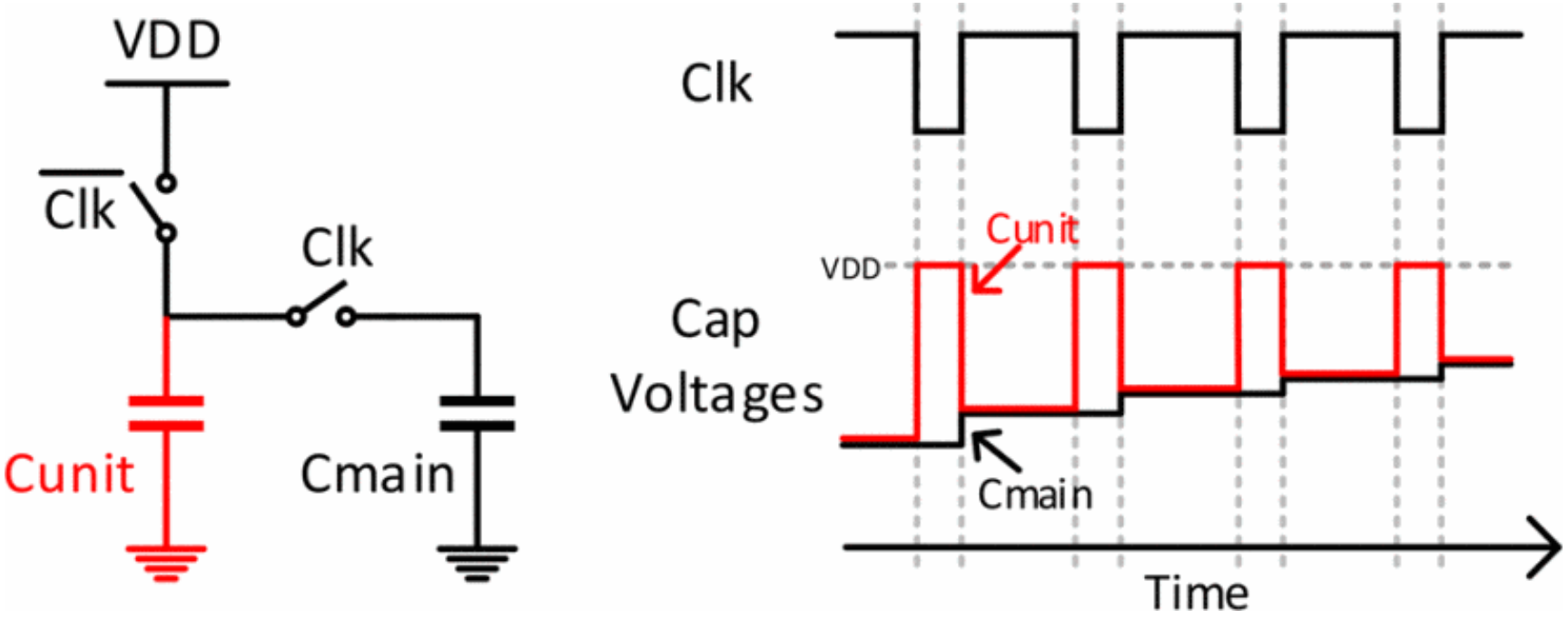
Charge Sharing



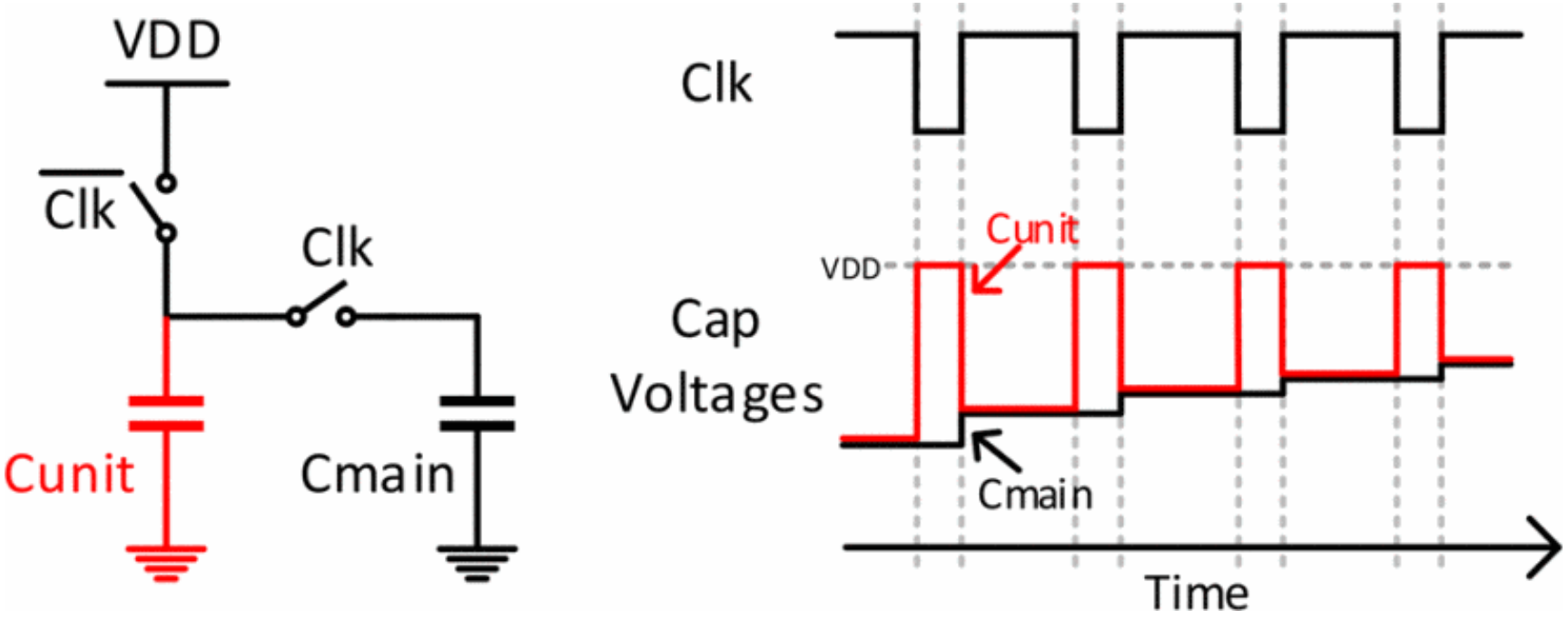
Charge Sharing



The Analog Trigger Circuit (Revised)

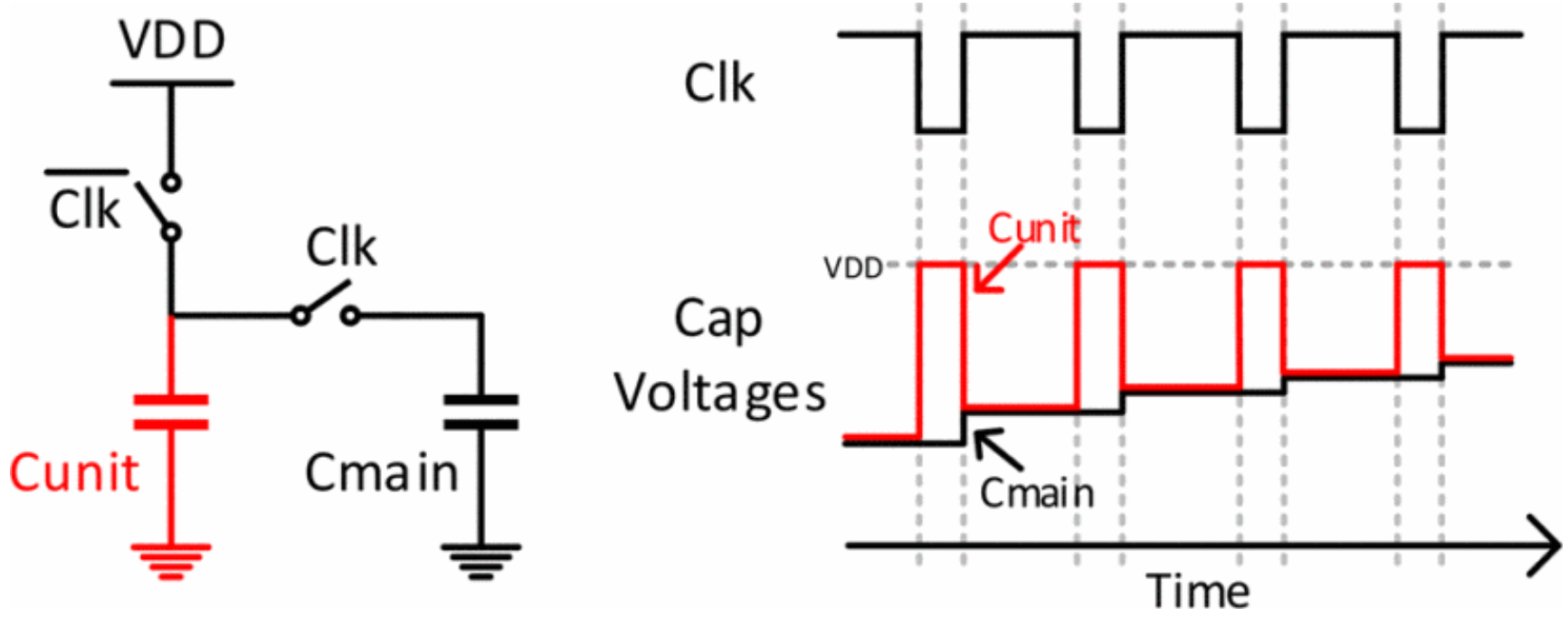


The Analog Trigger Circuit (Revised)



1. What can this trigger be used for?

The Analog Trigger Circuit (Revised)



1. What can this trigger be used for?
2. What do we connect it to?

1. Example Payload Circuit

1. Example Payload Circuit

Observation:

1. Example Payload Circuit

Observation:

Many processors de-escalate privilege stepwise after reset

1. Example Payload Circuit

Observation:

Many processors de-escalate privilege stepwise after reset

Idea:

1. Example Payload Circuit

Observation:

Many processors de-escalate privilege stepwise after reset

Idea:

Tap into reset wires of supervisor mode register

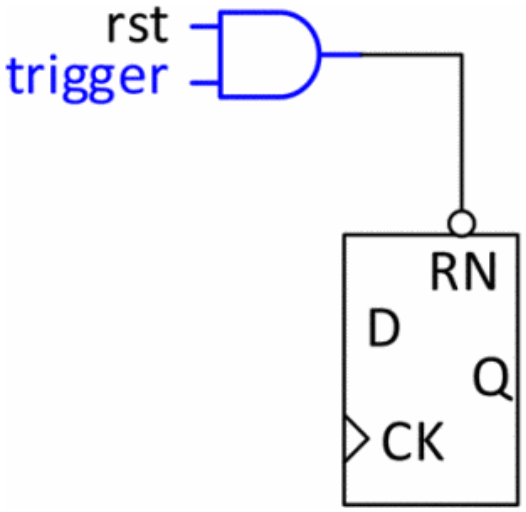
1. Example Payload Circuit

1. Example Payload Circuit

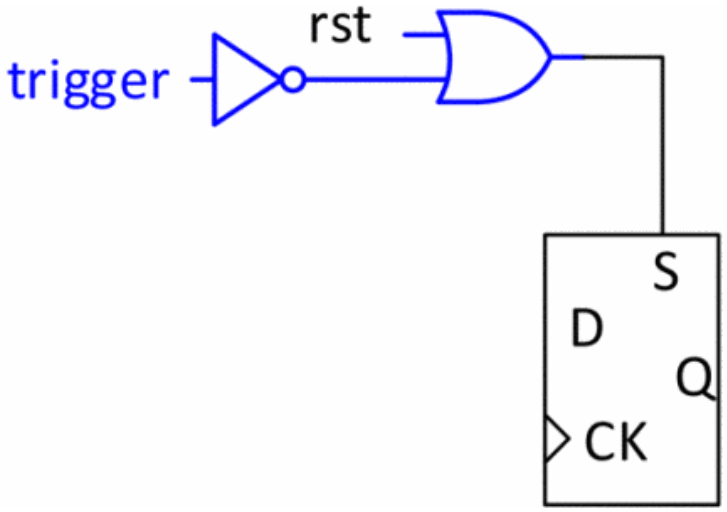
Privilege escalation by flipping the supervisor mode bit

1. Example Payload Circuit

Privilege escalation by flipping the supervisor mode bit



Active-low reset variant



Active-high reset variant

2. How to Find a Victim Wire

2. How to Find a Victim Wire

Observation:

2. How to Find a Victim Wire

Observation:

Need to find a software controllable wire with usually very low toggle rate

2. How to Find a Victim Wire

Observation:

Need to find a software controllable wire with usually very low toggle rate

Idea:

2. How to Find a Victim Wire

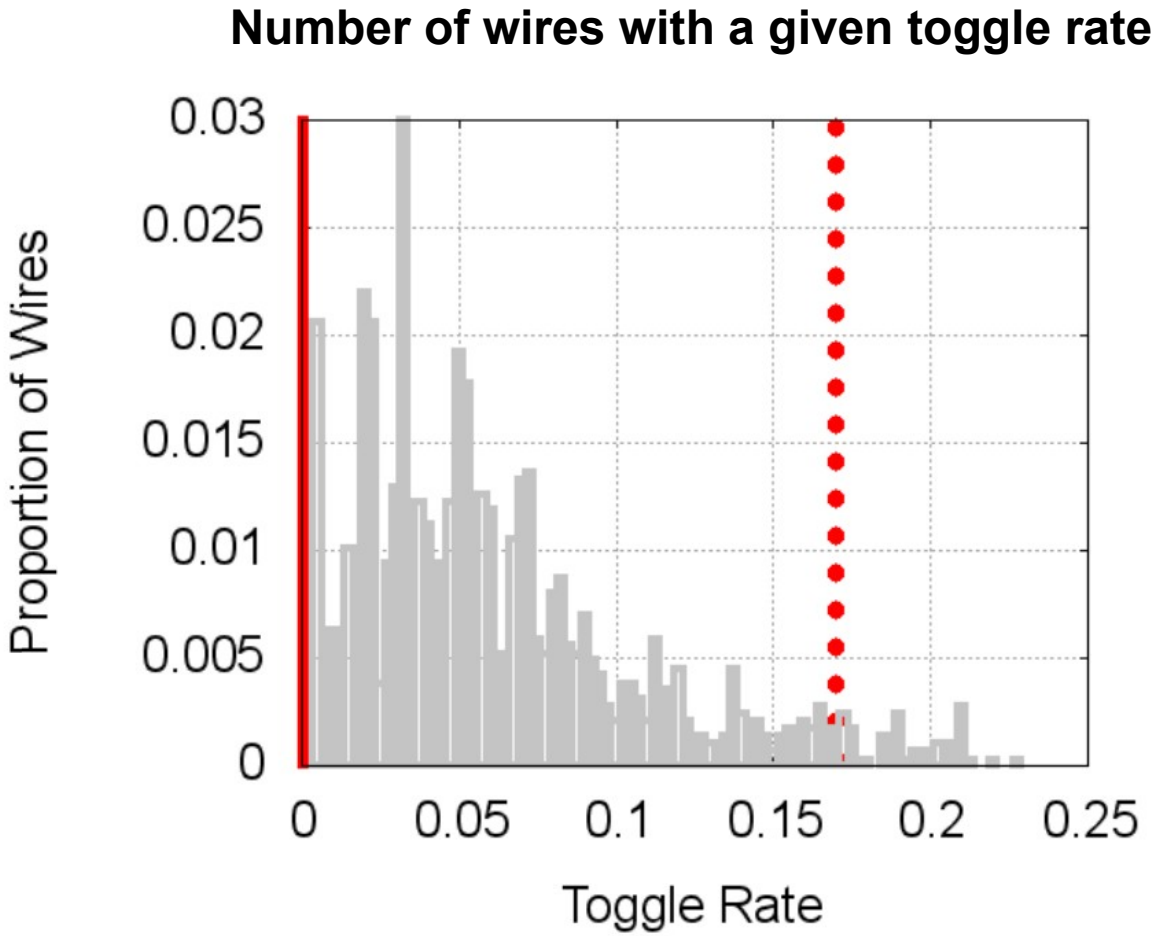
Observation:

Need to find a software controllable wire with usually very low toggle rate

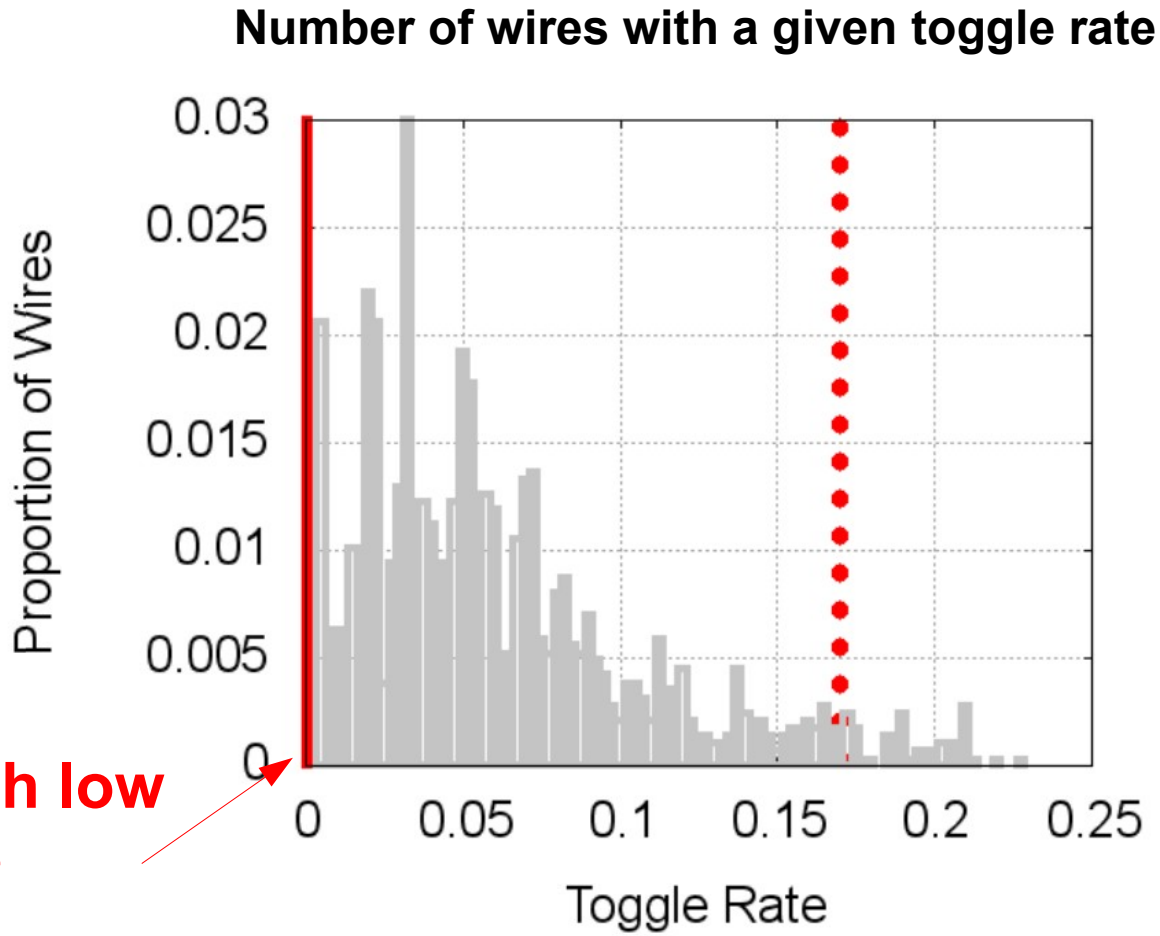
Idea:

Simulate different programs to find wires with low toggle rates

2. How to Find a Victim Wire

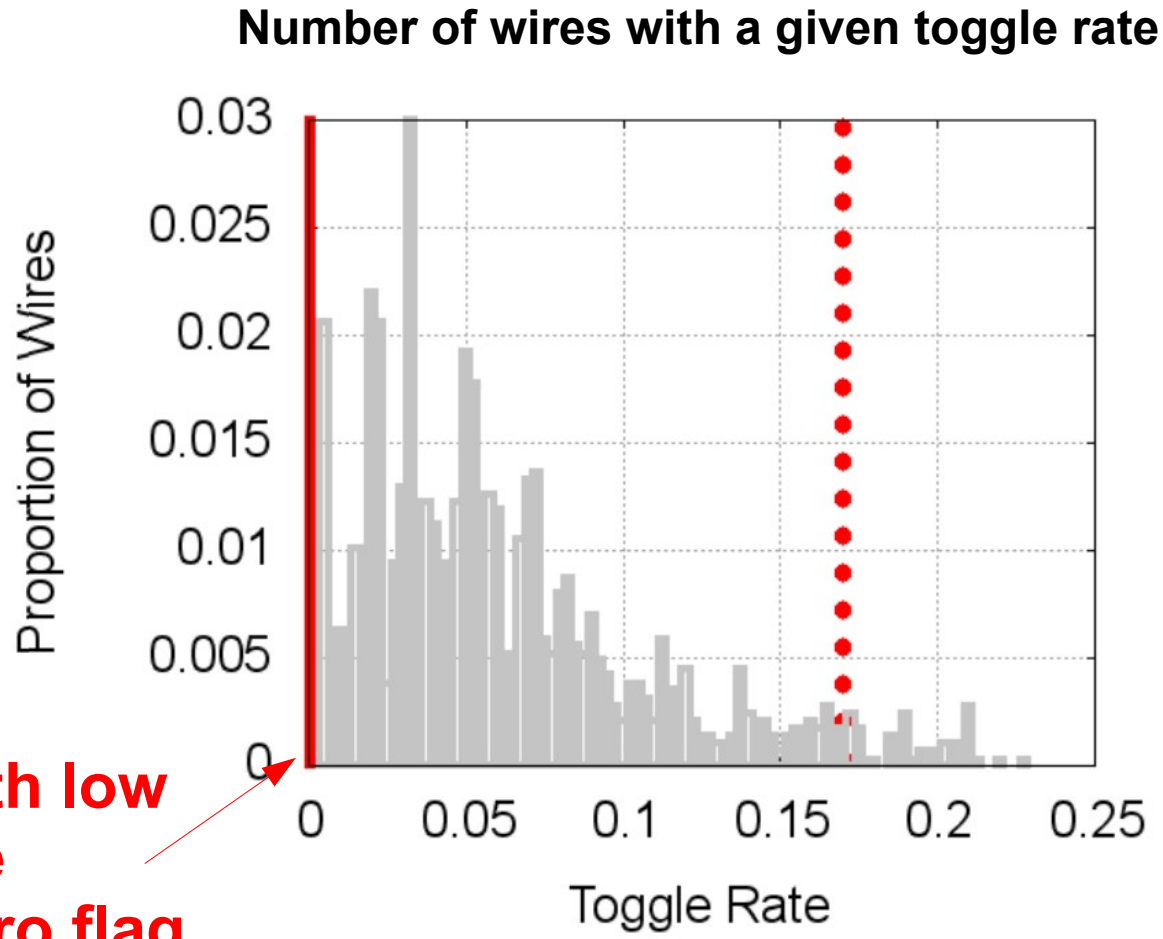


2. How to Find a Victim Wire



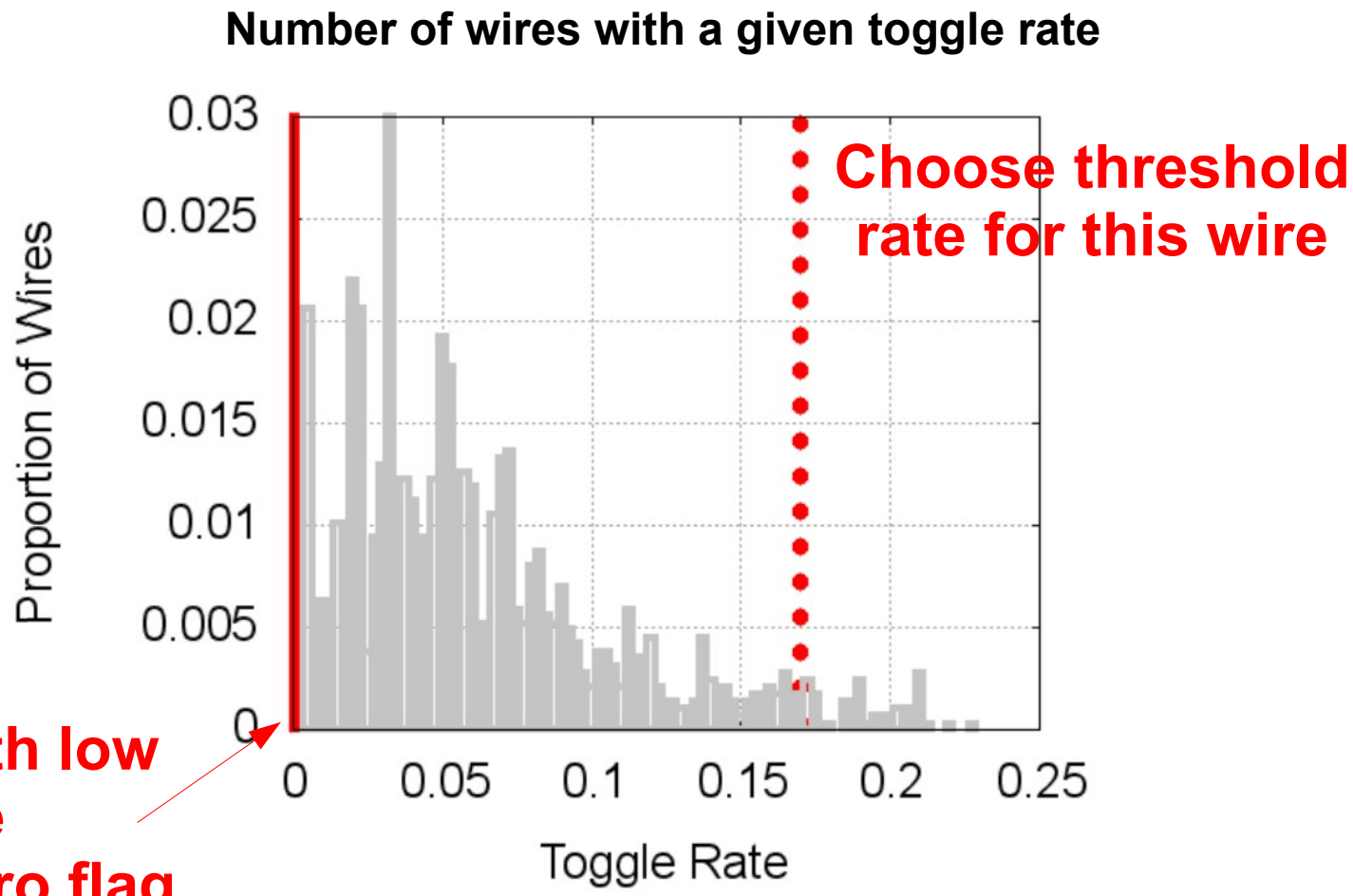
Choose wire with low toggle rate

2. How to Find a Victim Wire



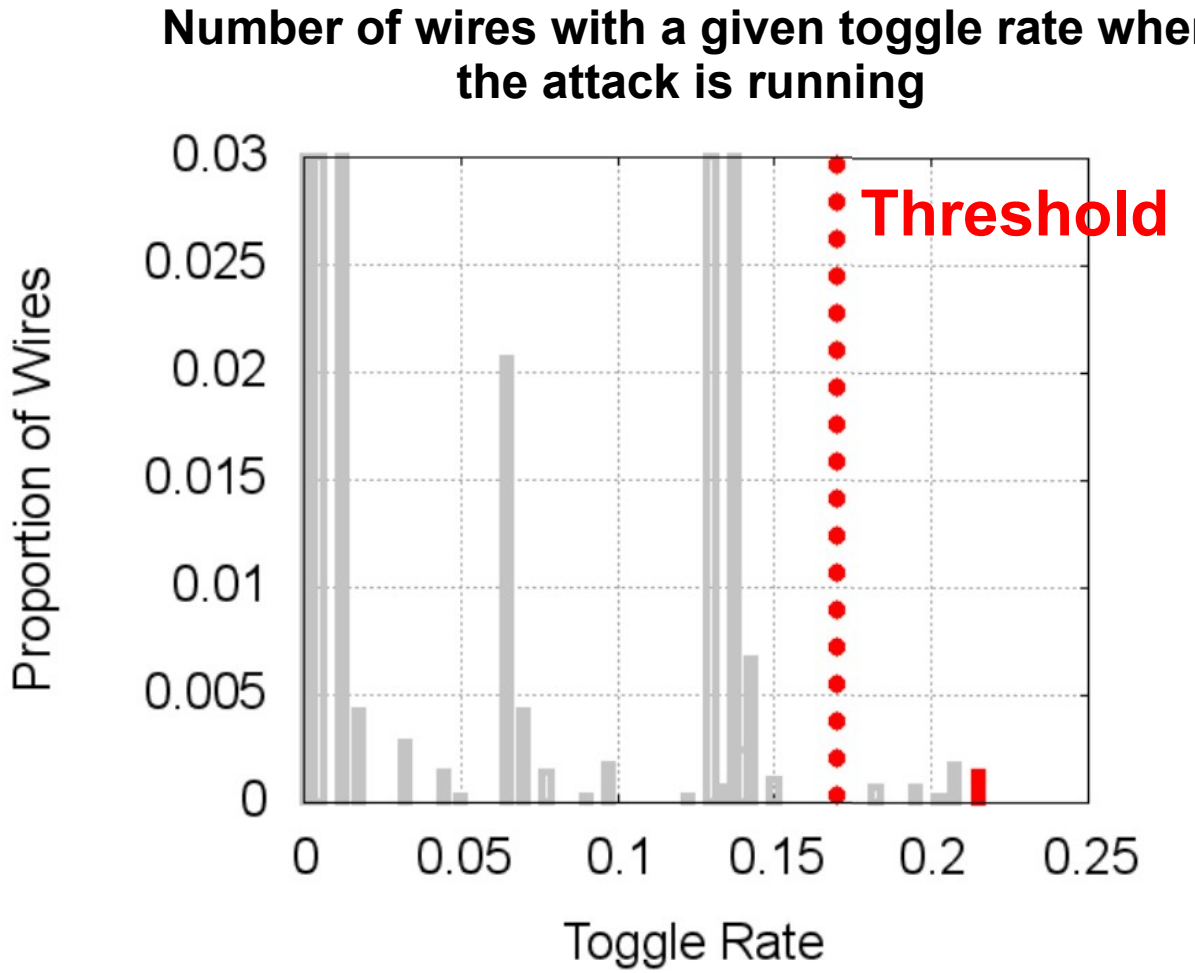
Choose wire with low toggle rate e.g. divide by zero flag

2. How to Find a Victim Wire

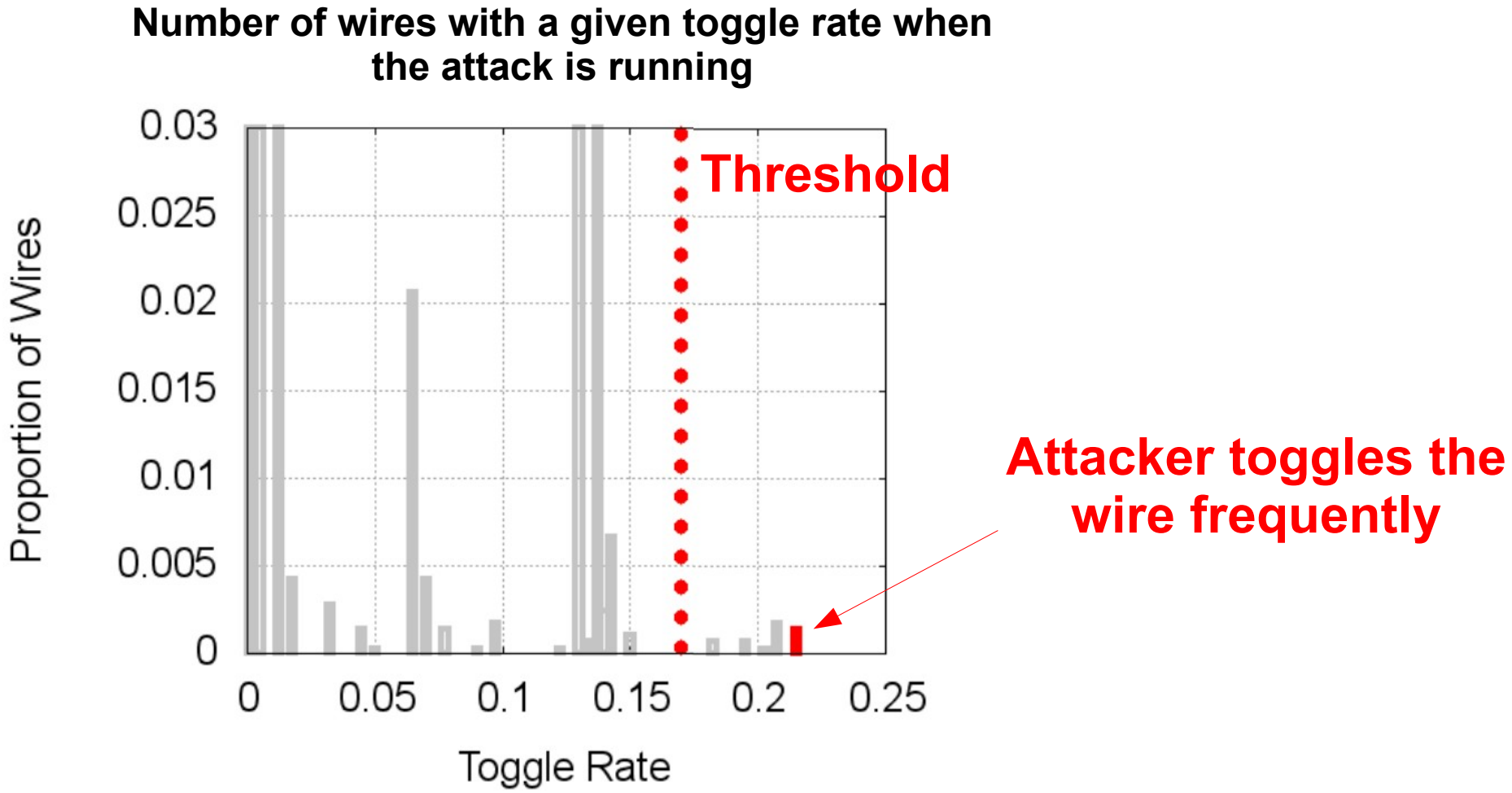


Choose wire with low toggle rate e.g. divide by zero flag

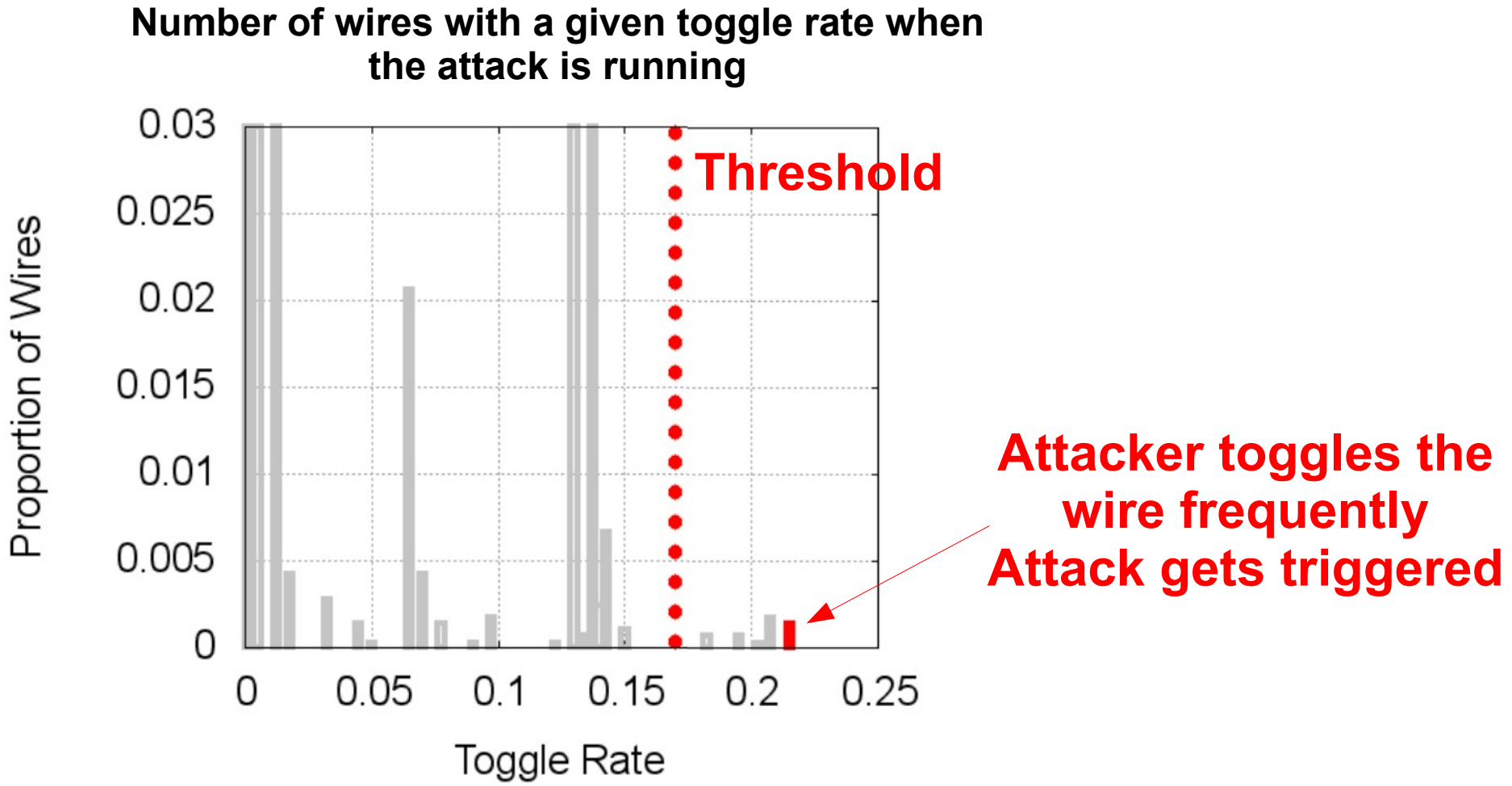
2. How to Find a Victim Wire



2. How to Find a Victim Wire



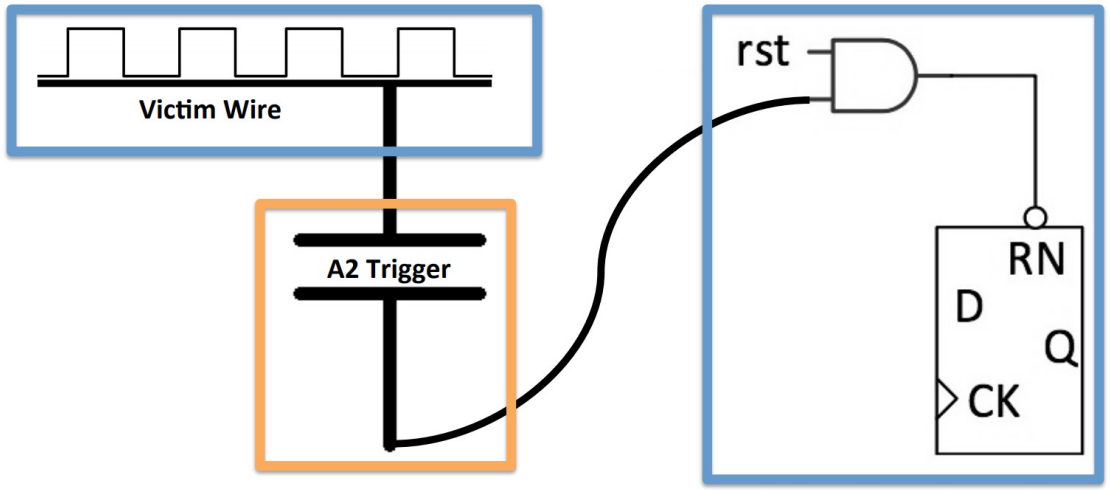
2. How to Find a Victim Wire



Controlling the Attack From Software

Attack Code Example

```
/* Victim wire is divide by zero
flag */
while attack_success == 0 do
  i ← 0
  while i < 500 do
    z ← 1/0
    i ← i + 1
  end while
  if test_privileges() == 1 then
    attack_success ← 1
  end if
end while
```



Analog domain and digital domain of A2

Key Results

Methodology

Methodology

How the attack was evaluated:

Methodology

How the attack was evaluated:

1. Verification of design in simulation on 65nm CMOS in SPICE

Methodology

How the attack was evaluated:

1. Verification of design in simulation on 65nm CMOS in SPICE

2. Implementation and verification of design in a real processor

Methodology

How the attack was evaluated:

1. Verification of design in simulation on 65nm CMOS in SPICE

2. Implementation and verification of design in a real processor

3. Comparison of the results from 1. and 2.

Methodology

How the attack was evaluated:

1. Verification of design in simulation on 65nm CMOS in SPICE

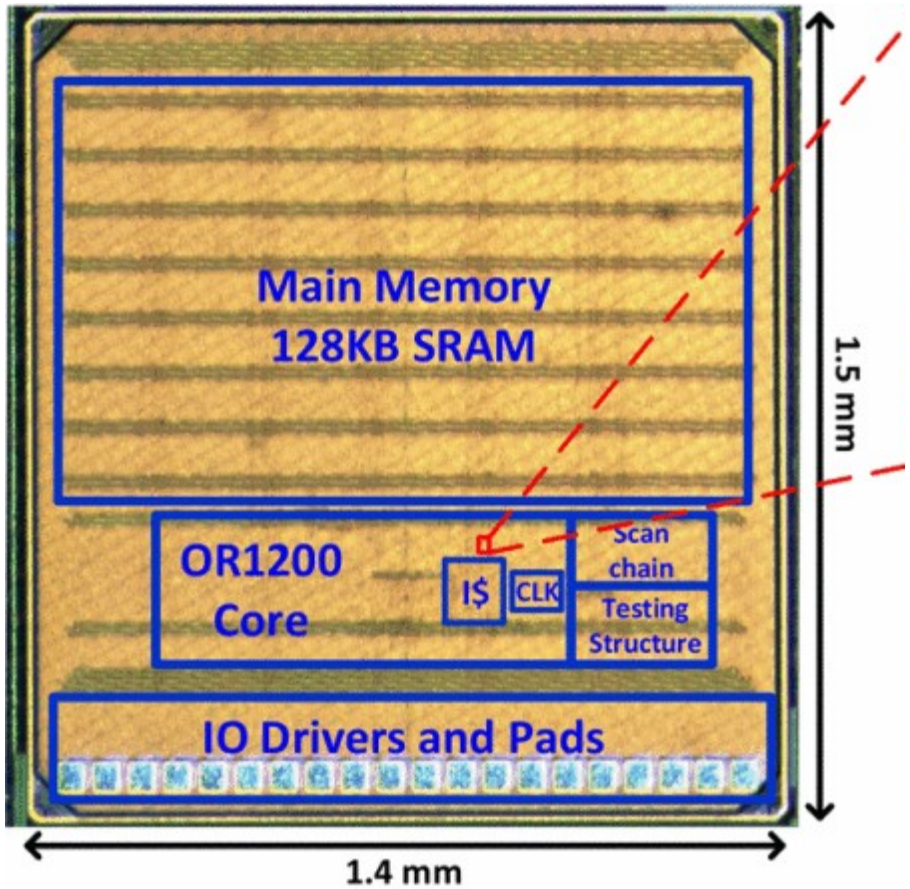
2. Implementation and verification of design in a real processor

3. Comparison of the results from 1. and 2.

4. Assessing detectability

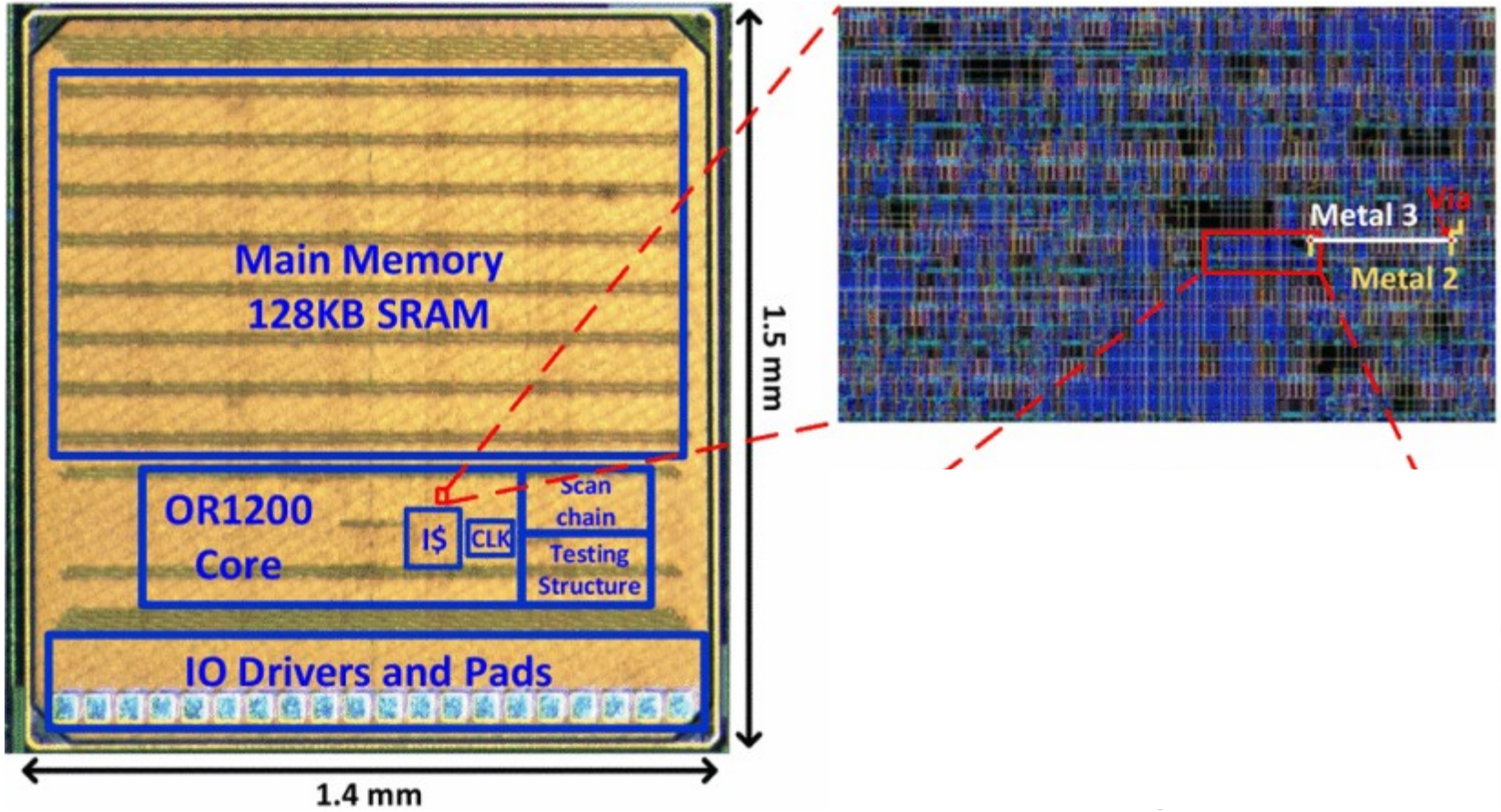
Implementation in a Real Chip

OpenRISC 1200 Processor



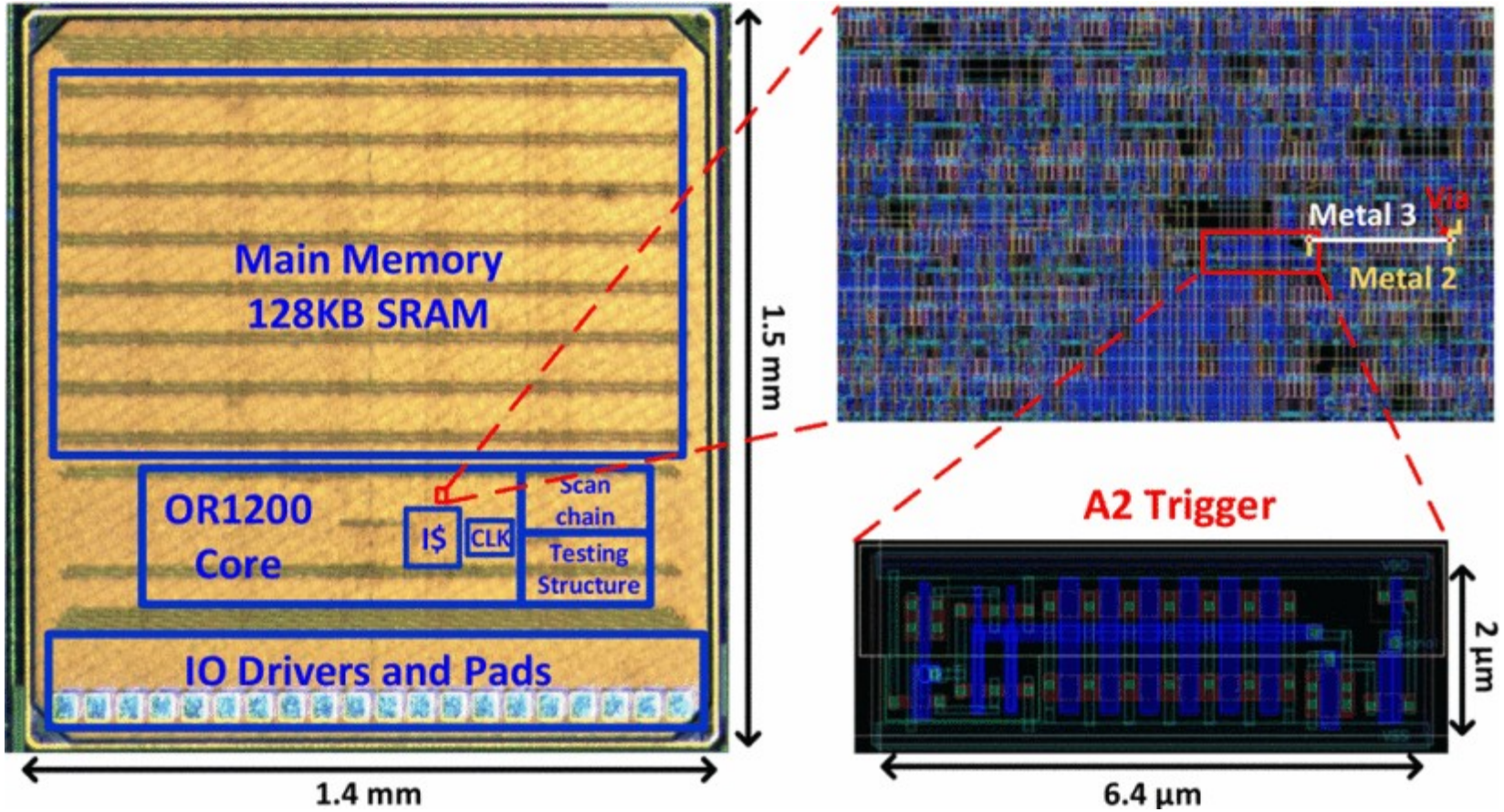
Implementation in a Real Chip

OpenRISC 1200 Processor



Implementation in a Real Chip

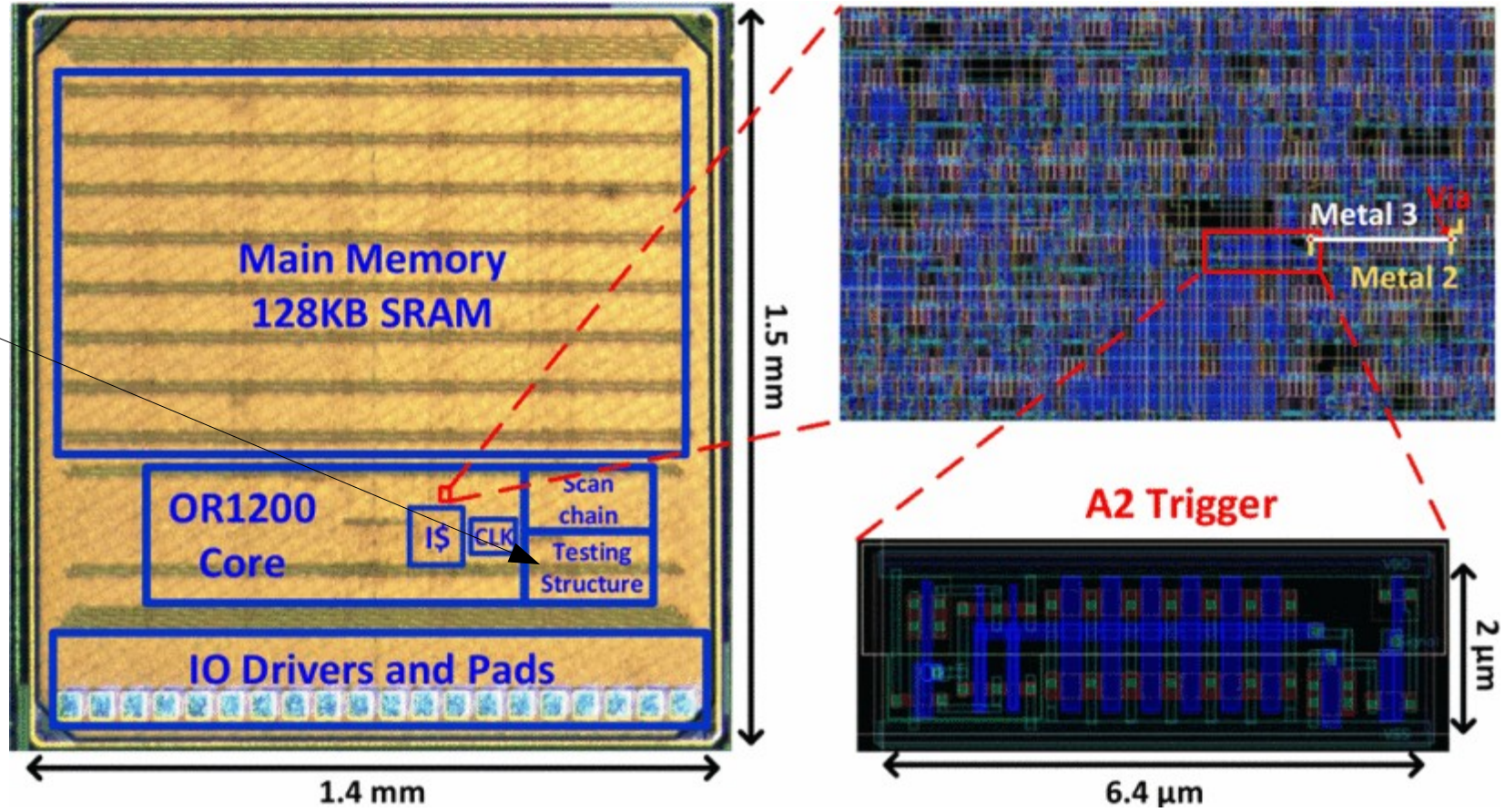
OpenRISC 1200 Processor



Implementation in a Real Chip

OpenRISC 1200 Processor

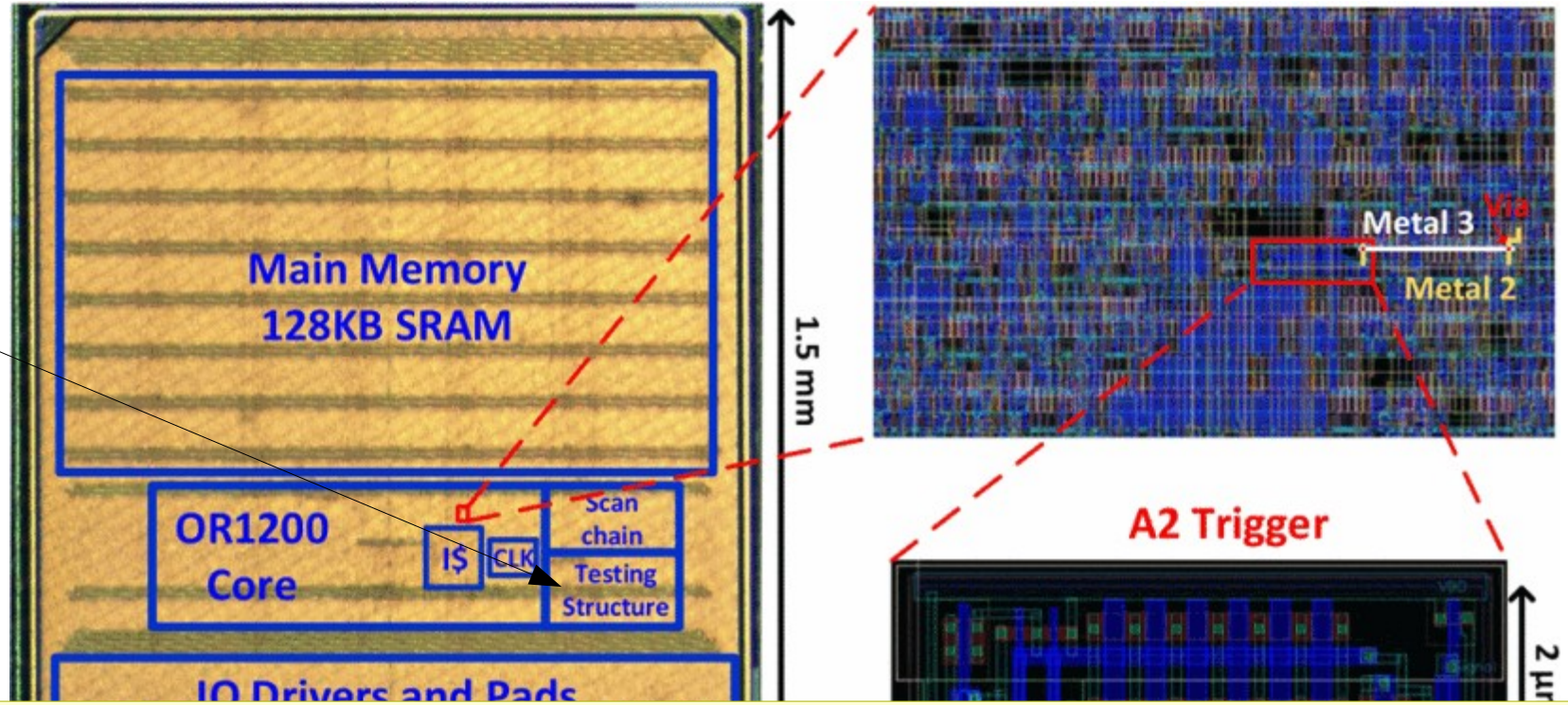
Includes stand-alone trigger testing structure



Implementation in a Real Chip

OpenRISC 1200 Processor

Includes stand-alone trigger testing structure



Uses only 0.08% of the total area!

Goal of the Paper

Goal: Design a hardware attack that is

Very small

Controllable

Stealthy

Goal of the Paper

Goal: Design a hardware attack that is

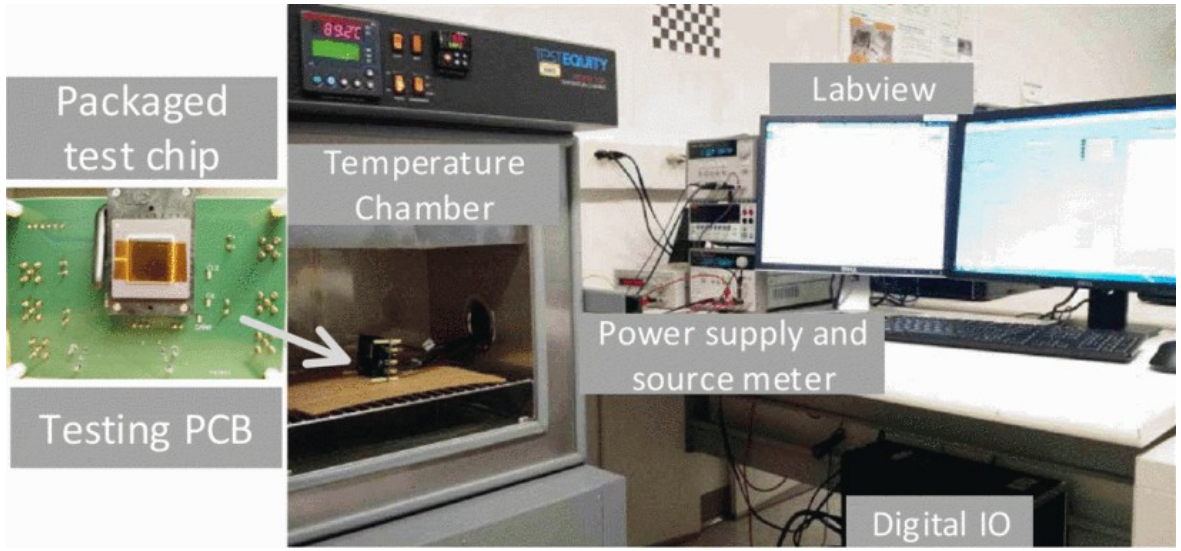
Very small 

Controllable

Stealthy

Verification in the OR1200 Processor

Key Results

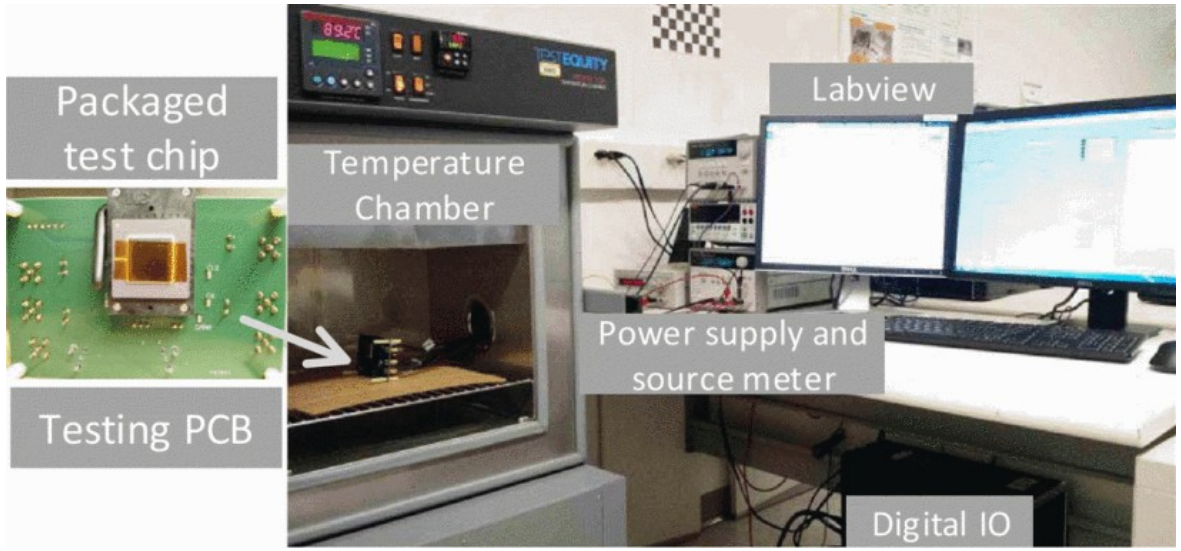


Testing setup

Verification in the OR1200 Processor

Key Results

Circuits tested under temperature, clock frequency and voltage variations



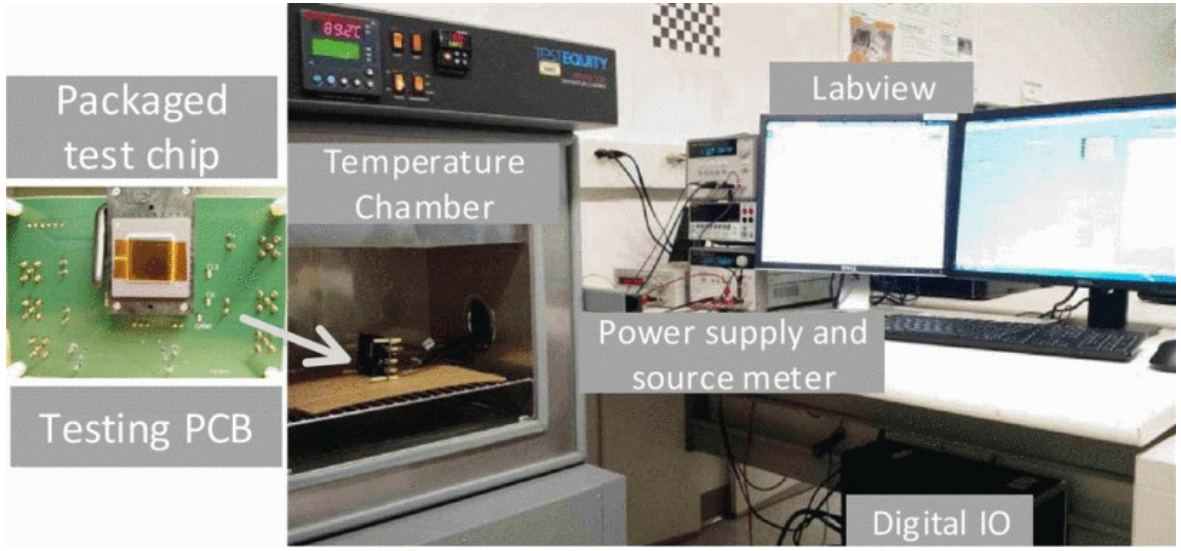
Testing setup

Verification in the OR1200 Processor

Key Results

Circuits tested under temperature, clock frequency and voltage variations

Tested on multiple chips



Testing setup

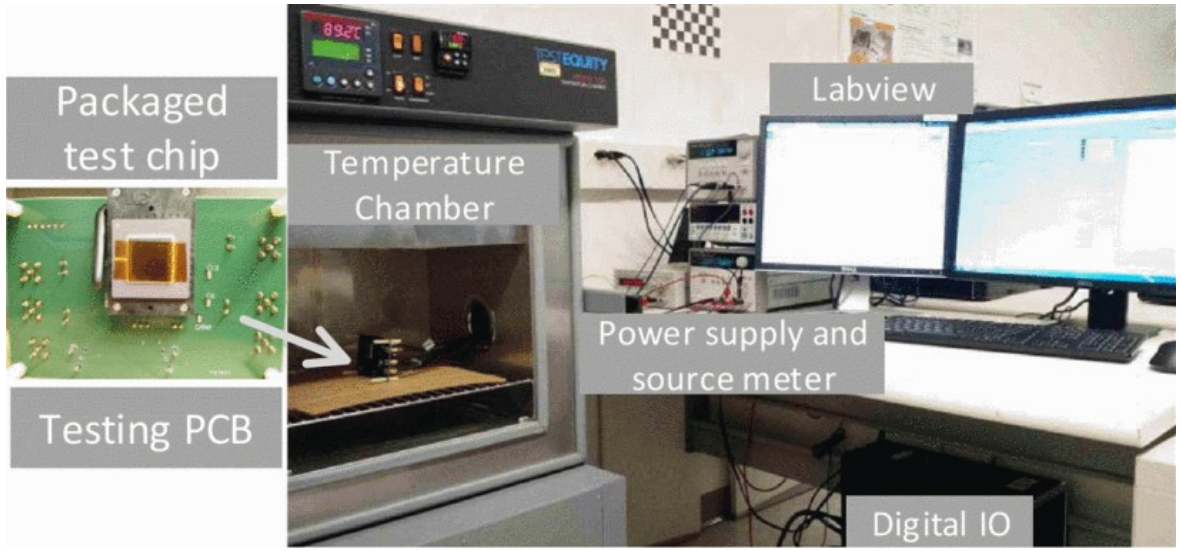
Verification in the OR1200 Processor

Key Results

Circuits tested under temperature, clock frequency and voltage variations

Tested on multiple chips

Trigger and retention times measured using the separate testing structure



Testing setup

Test Results of Real Chip Implementation

Key Results

Test Results of Real Chip Implementation

Attacks in the chips are:

Test Results of Real Chip Implementation

Attacks in the chips are:

Robust against manufacturing variations

Test Results of Real Chip Implementation

Attacks in the chips are:

Robust against manufacturing variations

Robust against supply voltage fluctuations

Test Results of Real Chip Implementation

Attacks in the chips are:

Robust against manufacturing variations

Robust against supply voltage fluctuations

Robust against temperature changes

Comparison to Simulation

Trigger times in cycles

Trigger Circuit	Toggle Rate (MHz)	Measured (10 chip avg)	Simulated (Typical corner)
w/o IO device	120.00	7.4	7
w/o IO device	34.29	8.4	8
w/o IO device	10.91	11.6	10

Comparison to Simulation

Trigger times in cycles

Trigger Circuit	Toggle Rate (MHz)	Measured (10 chip avg)	Simulated (Typical corner)
w/o IO device	120.00	7.4	7
w/o IO device	34.29	8.4	8
w/o IO device	10.91	11.6	10

Comparison to Simulation

Trigger times in cycles

Trigger Circuit	Toggle Rate (MHz)	Measured (10 chip avg)	Simulated (Typical corner)
w/o IO device	120.00	7.4	7
w/o IO device	34.29	8.4	8
w/o IO device	10.91	11.6	10

Comparison shows that simulation has good enough accuracy to fabricate precise and controllable attacks!

Goal of the Paper

Goal: Design a hardware attack that is

Very small 

Controllable

Stealthy

Goal of the Paper

Goal: Design a hardware attack that is

Very small 

Controllable 

Stealthy

How to Detect a Hardware Attack

Key Results

How to Detect a Hardware Attack

Side Channel Information

How to Detect a Hardware Attack

Side Channel Information

Temperature

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

Adding Sensors

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

Adding Sensors

Measure signal propagation delays

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

Adding Sensors

Measure signal propagation delays

Detects attacks that add logic to wires

How to Detect a Hardware Attack

Key Results

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

Visual Inspection

Adding Sensors

Measure signal propagation delays

Detects attacks that add logic to wires

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

Visual Inspection

Delayering the chip

Adding Sensors

Measure signal propagation delays

Detects attacks that add logic to wires

How to Detect a Hardware Attack

Key Results

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

Visual Inspection

Delayering the chip

Inspection via scanning electron microscope

Adding Sensors

Measure signal propagation delays

Detects attacks that add logic to wires

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

Visual Inspection

Delayering the chip

Inspection via scanning electron microscope

Detects attacks that are big

Adding Sensors

Measure signal propagation delays

Detects attacks that add logic to wires

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

Visual Inspection

Delayering the chip

Inspection via scanning electron microscope

Detects attacks that are big

Adding Sensors

Measure signal propagation delays

Detects attacks that add logic to wires

Functional Testing

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

Visual Inspection

Delayering the chip

Inspection via scanning electron microscope

Detects attacks that are big

Adding Sensors

Measure signal propagation delays

Detects attacks that add logic to wires

Functional Testing

Test for unexpected behaviour

How to Detect a Hardware Attack

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Detects attacks that get hot or use much power

Visual Inspection

Delayering the chip

Inspection via scanning electron microscope

Detects attacks that are big

Adding Sensors

Measure signal propagation delays

Detects attacks that add logic to wires

Functional Testing

Test for unexpected behaviour

Detects some attacks that change the circuit behaviour

How Stealthy is the Attack?

How Stealthy is the Attack?

Can the attack be detected by side channels?

How Stealthy is the Attack?

Can the attack be detected by side channels?

Measuring of chip power consumption

How Stealthy is the Attack?

Can the attack be detected by side channels?

Measuring of chip power consumption

Simulating theoretical power usage of trigger circuit

How Stealthy is the Attack?

Can the attack be detected by side channels?

Measuring of chip power consumption

Simulating theoretical power usage of trigger circuit

Answer:

How Stealthy is the Attack?

Can the attack be detected by side channels?

Measuring of chip power consumption

Simulating theoretical power usage of trigger circuit

Answer:

The power requirements of the attack are well below normal fluctuations

Detection Mechanisms Evaded by A2

Side Channel Information

Temperature

Power requirements

Electromagnetic measurements

Adding Sensors

Measure signal propagation delays

Visual Inspection

Delayering the chip

Inspection via scanning electron microscope

Functional Testing

Test for unexpected behaviour

Detection Mechanisms Evaded by A2

Side Channel Information ✓

Temperature

Power requirements

Electromagnetic measurements

Adding Sensors

Measure signal propagation delays

Visual Inspection

Delayering the chip

Inspection via scanning electron microscope

Functional Testing

Test for unexpected behaviour

Maybe With On-Chip Sensors?

Maybe With On-Chip Sensors?

Can the attack be detected by measuring propagation delays?

Maybe With On-Chip Sensors?

Can the attack be detected by measuring propagation delays?

High accuracy simulation of trigger wire delays

Maybe With On-Chip Sensors?

Can the attack be detected by measuring propagation delays?

High accuracy simulation of trigger wire delays

Reset wires are typically asynchronous

Maybe With On-Chip Sensors?

Can the attack be detected by measuring propagation delays?

High accuracy simulation of trigger wire delays

Reset wires are typically asynchronous

Answer:

Maybe With On-Chip Sensors?

Can the attack be detected by measuring propagation delays?

High accuracy simulation of trigger wire delays

Reset wires are typically asynchronous

Answer:

For a **4ns** clock period the delay change is **only 0.33%** and well below process variation and noise

Detection Mechanisms Evaded by A2

Side Channel Information ✓

Temperature

Power requirements

Electromagnetic measurements

Adding Sensors

Measure signal propagation delays

Visual Inspection

Delayering the chip

Inspection via scanning electron microscope

Functional Testing

Test for unexpected behaviour

Detection Mechanisms Evaded by A2

Side Channel Information ✓

Temperature

Power requirements

Electromagnetic measurements

Adding Sensors ✓

Measure signal propagation delays

Visual Inspection

Delayering the chip

Inspection via scanning electron microscope

Functional Testing

Test for unexpected behaviour

And Visual Inspection?

And Visual Inspection?

Can the attack be found by looking at the chip?

And Visual Inspection?

Can the attack be found by looking at the chip?

A2 is as small as **one gate** and is almost identical to the other gates in a design

And Visual Inspection?

Can the attack be found by looking at the chip?

A2 is as small as **one gate** and is almost identical to the other gates in a design

Difficult to distinguish one gate in a sea of hundreds of thousands of gates (or even more)

And Visual Inspection?

Can the attack be found by looking at the chip?

A2 is as small as **one gate** and is almost identical to the other gates in a design

Difficult to distinguish one gate in a sea of hundreds of thousands of gates (or even more)

Requires delayering to very low layers

And Visual Inspection?

Can the attack be found by looking at the chip?

A2 is as small as **one gate** and is almost identical to the other gates in a design

Difficult to distinguish one gate in a sea of hundreds of thousands of gates (or even more)

Requires delayering to very low layers

Answer:

And Visual Inspection?

Can the attack be found by looking at the chip?

A2 is as small as **one gate** and is almost identical to the other gates in a design

Difficult to distinguish one gate in a sea of hundreds of thousands of gates (or even more)

Requires delayering to very low layers

Answer:

A2 is unlikely to be found by visual inspection

Detection Mechanisms Evaded by A2

Key Results

Side Channel Information ✓

Temperature

Power requirements

Electromagnetic measurements

Adding Sensors ✓

Measure signal propagation delays

Visual Inspection

Delayering the chip

Inspection via scanning electron microscope

Functional Testing

Test for unexpected behaviour

Detection Mechanisms Evaded by A2

Side Channel Information ✓

Temperature

Power requirements

Electromagnetic measurements

Adding Sensors ✓

Measure signal propagation delays

Visual Inspection ✓

Delayering the chip

Inspection via scanning electron microscope

Functional Testing

Test for unexpected behaviour

What About Functional Testing?

What About Functional Testing?

Is the attack triggered during normal execution?

What About Functional Testing?

Is the attack triggered during normal execution?

Testing with five selected benchmark programs

What About Functional Testing?

Is the attack triggered during normal execution?

Testing with five selected benchmark programs

Testing over 6 different temperatures from -25°C to 100°C

What About Functional Testing?

Is the attack triggered during normal execution?

Testing with five selected benchmark programs

Testing over 6 different temperatures from -25°C to 100°C

Answer:

What About Functional Testing?

Is the attack triggered during normal execution?

Testing with five selected benchmark programs

Testing over 6 different temperatures from -25°C to 100°C

Answer:

The attack was not activated across all programs and temperatures

Detection Mechanisms Evaded by A2

Side Channel Information ✓

Temperature

Power requirements

Electromagnetic measurements

Adding Sensors ✓

Measure signal propagation delays

Visual Inspection ✓

Delayering the chip

Inspection via scanning electron microscope

Functional Testing

Test for unexpected behaviour

Detection Mechanisms Evaded by A2

Side Channel Information ✓

Temperature

Power requirements

Electromagnetic measurements

Adding Sensors ✓

Measure signal propagation delays

Visual Inspection ✓

Delayering the chip

Inspection via scanning electron microscope

Functional Testing ✓

Test for unexpected behaviour

Detection Mechanisms Evaded by A2

Side Channel Information ✓

Temperature

Power requirements

Electromagnetic measurements

Adding Sensors ✓

Measure signal propagation delays

Visual Inspection ✓

Delayering the chip

Inspection via scanning electron microscope

Functional Testing ✓

Test for unexpected behaviour

Detection Mechanisms Evaded by A2

Side Channel Information ✓

Temperature

Power requirements

Electromagnetic measurements

Adding Sensors ✓

Measure signal propagation delays

Visual Inspection ✓

Delayering the chip

Inspection via scanning electron microscope

Functional Testing ✓

Test for unexpected behaviour

A2 is not easily detectable!

Goal of the Paper

Goal: Design a hardware attack that is

Very small 

Controllable 

Stealthy

Goal of the Paper

Goal: Design a hardware attack that is

Very small 

Controllable 

Stealthy 

Goal of the Paper

Goal: Design a hardware attack that is

Very small ✓

Controllable ✓

Stealthy ✓

Possible Defenses Against A2

Possible Defenses Against A2

One possible defense against A2 could come in the form of split manufacturing:

Possible Defenses Against A2

One possible defense against A2 could come in the form of split manufacturing:

Subset of the chip design is fabricated in a trusted manufacturing facility

Possible Defenses Against A2

One possible defense against A2 could come in the form of split manufacturing:

Subset of the chip design is fabricated in a trusted manufacturing facility

Very expensive

Possible Defenses Against A2

One possible defense against A2 could come in the form of split manufacturing:

Subset of the chip design is fabricated in a trusted manufacturing facility

Very expensive

Difficult to do, as wires can be reverse engineered and flip-flops are typically fabricated by the third party

Possible Defenses Against A2

One possible defense against A2 could come in the form of split manufacturing:

Subset of the chip design is fabricated in a trusted manufacturing facility

Very expensive

Difficult to do, as wires can be reverse engineered and flip-flops are typically fabricated by the third party

Needs a new type of defense!

Summary

Summary

Summary

Problem: Current hardware attacks have some inherent flaws, i.e., they are 1) big, 2) uncontrollable or 3) not stealthy enough

Summary

Problem: Current hardware attacks have some inherent flaws, i.e., they are 1) big, 2) uncontrollable or 3) not stealthy enough

Goal: create a hardware attack that is small (i.e., requires as little as one gate) and stealthy (i.e.,requires an unlikely trigger sequence before effecting a chip's functionality) and controllable.

Summary

Problem: Current hardware attacks have some inherent flaws, i.e., they are 1) big, 2) uncontrollable or 3) not stealthy enough

Goal: create a hardware attack that is small (i.e., requires as little as one gate) and stealthy (i.e.,requires an unlikely trigger sequence before effecting a chip's functionality) and controllable.

Key Idea:

- Construct a circuit that only uses 2 capacitors to siphon charge from nearby wires as they transition between digital values.
- When the capacitors are fully charged, deploy an attack that forces a victim flip-flop to the desired value.

Summary

Problem: Current hardware attacks have some inherent flaws, i.e., they are 1) big, 2) uncontrollable or 3) not stealthy enough

Goal: create a hardware attack that is small (i.e., requires as little as one gate) and stealthy (i.e.,requires an unlikely trigger sequence before effecting a chip's functionality) and controllable.

Key Idea:

-Construct a circuit that only uses 2 capacitors to siphon charge from nearby wires as they transition between digital values.

-When the capacitors are fully charged, deploy an attack that forces a victim flip-flop to the desired value.

Key Results: 1) Implemented this attack in an OR1200 processor and fabricated a chip; 2) Experimental results show that the attack works efficiently; 3) The attack eludes activation by a diverse set of benchmarks; 4) the attack evades known defenses

Strengths

Strengths of the Paper

Strengths of the Paper

+ Shows a new type of hardware attack not seen before

Strengths of the Paper

+ Shows a new type of hardware attack not seen before
+ Real hardware implementation

Strengths of the Paper

+ Shows a new type of hardware attack not seen before
+ Real hardware implementation
+ Shows thorough testing of the attack

Strengths of the Paper

+ Shows a new type of hardware attack not seen before
+ Real hardware implementation
+ Shows thorough testing of the attack
+ Uses a strong and realistic threat model

Strengths of the Paper

+ Shows a new type of hardware attack not seen before
+ Real hardware implementation
+ Shows thorough testing of the attack
+ Uses a strong and realistic threat model
+ Assesses the possibility of an implementation in different architectures

Strengths of the Paper

+ Shows a new type of hardware attack not seen before
+ Real hardware implementation
+ Shows thorough testing of the attack
+ Uses a strong and realistic threat model
+ Assesses the possibility of an implementation in different architectures
+ Well written and relatively easy to understand

Strengths of the Paper

+ Shows a new type of hardware attack not seen before
+ Real hardware implementation
+ Shows thorough testing of the attack
+ Uses a strong and realistic threat model
+ Assesses the possibility of an implementation in different architectures
+ Well written and relatively easy to understand
+ Gives a history on previous work done in the field

Weaknesses & Limitations

Weaknesses & Limitations

Weaknesses & Limitations

- Does not give a concrete defense mechanism

Weaknesses & Limitations

- Does not give a concrete defense mechanism

- Cannot test hypothesis on other architectures due to cost and secrecy

Weaknesses & Limitations

- Does not give a concrete defense mechanism
- Cannot test hypothesis on other architectures due to cost and secrecy
- Contains a few typos

Thoughts & Ideas

Thought and Ideas

Thought and Ideas

- **Can this charge-pump mechanism be used for good purposes?**
  - i.e. avoiding complicated state machines where precision is not as important
  - As was mentioned last week, maybe to prevent Rowhammer attacks?

Thought and Ideas

- **Can this charge-pump mechanism be used for good purposes?**
  - i.e. avoiding complicated state machines where precision is not as important
  - As was mentioned last week, maybe to prevent Rowhammer attacks?

- **Is this attack already used?**
  - I have not found any evidence that this attack is being used yet (please prove me wrong)
  - I have found cases for other hardware trojans though, e.g. [1]
  - Can you think of other cases of hardware attacks being used?

[1] S. Skorobogatov, C. Woods, "**Breakthrough silicon scanning discovers backdoor in military chip**", Proc. 14th Int. Conf. Cryptograph. Hardw. Embedded Syst., pp. 23-40, 2012.

Thought and Ideas

- **Can this charge-pump mechanism be used for good purposes?**
  - i.e. avoiding complicated state machines where precision is not as important
  - As was mentioned last week, maybe to prevent Rowhammer attacks?
- **Is this attack already used?**
  - I have not found any evidence that this attack is being used yet (please prove me wrong)
  - I have found cases for other hardware trojans though, e.g. [1]
  - Can you think of other cases of hardware attacks being used?
- **What has to be considered when applying this attack to other (smaller) technology nodes?**

[1] S. Skorobogatov, C. Woods, "**Breakthrough silicon scanning discovers backdoor in military chip**", Proc. 14th Int. Conf. Cryptograph. Hardw. Embedded Syst., pp. 23-40, 2012.

Some Interesting Follow-Up Papers

Some Interesting Follow-Up Papers

- Yumin Hou, Hu He, Kaveh Shamsi, Yier Jin, Dong Wu, Huaqiang Wu, "**R2D2: Runtime reassurance and detection of A2 Trojan**", Hardware Oriented Security and Trust (HOST) 2018 IEEE International

Some Interesting Follow-Up Papers

- Yumin Hou, Hu He, Kaveh Shamsi, Yier Jin, Dong Wu, Huaqiang Wu, "**R2D2: Runtime reassurance and detection of A2 Trojan**", Hardware Oriented Security and Trust (HOST) 2018 IEEE International

- Xiaolong Guo, Huifeng Zhu, Yier Jin, Xuan Zhang, "**When Capacitors Attack: Formal Method Driven Design and Detection of Charge-Domain Trojans**", Design Automation & Test in Europe Conference & Exhibition (DATE) 2019, pp. 1727-1732, 2019. Symposium on, pp. 195-200, 2018.

Some Interesting Follow-Up Papers

- Yumin Hou, Hu He, Kaveh Shamsi, Yier Jin, Dong Wu, Huaqiang Wu, "**R2D2: Runtime reassurance and detection of A2 Trojan**", Hardware Oriented Security and Trust (HOST) 2018 IEEE International

- Xiaolong Guo, Huifeng Zhu, Yier Jin, Xuan Zhang, "**When Capacitors Attack: Formal Method Driven Design and Detection of Charge-Domain Trojans**", Design Automation & Test in Europe Conference & Exhibition (DATE) 2019, pp. 1727-1732, 2019. Symposium on, pp. 195-200, 2018.

- Meng Li, Bei Yu, Yibo Lin, Xiaoqing Xu, Wuxi Li, David Z. Pan, "**A practical split manufacturing framework for Trojan prevention via simultaneous wire lifting and cell insertion**", Design Automation Conference (ASP-DAC) 2018 23rd Asia and South Pacific, pp. 265-270, 2018.

Some Interesting Follow-Up Papers

- Yumin Hou, Hu He, Kaveh Shamsi, Yier Jin, Dong Wu, Huaqiang Wu, "**R2D2: Runtime reassurance and detection of A2 Trojan**", Hardware Oriented Security and Trust (HOST) 2018 IEEE International

- Xiaolong Guo, Huifeng Zhu, Yier Jin, Xuan Zhang, "**When Capacitors Attack: Formal Method Driven Design and Detection of Charge-Domain Trojans**", Design Automation & Test in Europe Conference & Exhibition (DATE) 2019, pp. 1727-1732, 2019. Symposium on, pp. 195-200, 2018.

- Meng Li, Bei Yu, Yibo Lin, Xiaoqing Xu, Wuxi Li, David Z. Pan, "**A practical split manufacturing framework for Trojan prevention via simultaneous wire lifting and cell insertion**", Design Automation Conference (ASP-DAC) 2018 23rd Asia and South Pacific, pp. 265-270, 2018.

- Mohammad-Mahdi Bidmeshki, Angelos Antonopoulos, Yiorgos Makris, "**Information flow tracking in analog/mixed-signal designs through proof-carrying hardware IP**", Design Automation & Test in Europe Conference & Exhibition (DATE) 2017, pp. 1703-1708, 2017.

Open Discussion

Open Discussion

- How would you try to detect A2?
- How bad do you think is this type of attack?
- Can you think of a better attack?
- Do you think the shown follow-up papers solve the problem?
- Can the proposed mechanism be used for good?
- What are your thoughts on this paper?
- What do you think are the most important takeaways here?

Open Discussion

- How would you try to detect A2?
- How bad do you think is this type of attack?
- Can you think of a better attack?
- Do you think the shown follow-up papers solve the problem?
- Can the proposed mechanism be used for good?
- What are your thoughts on this paper?
- What do you think are the most important takeaways here?

Moodle Discussion

https://moodle-app2.let.ethz.ch/mod/forum/discuss.php?d=38995

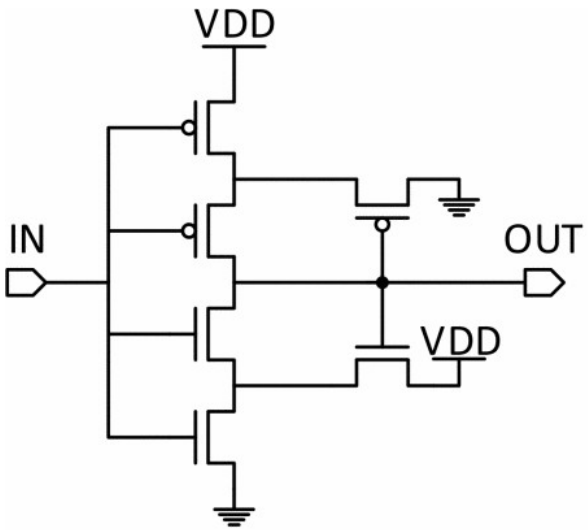
Thank You For Your Attention!

Backup Slides

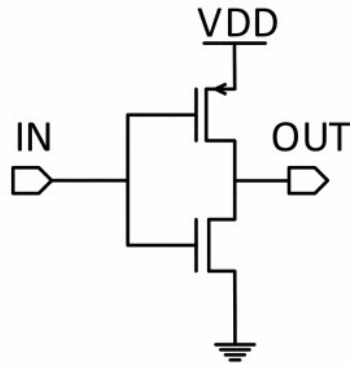
Threshold Detector

Threshold Detector

Two possibilities for threshold detectors



Schmitt Trigger

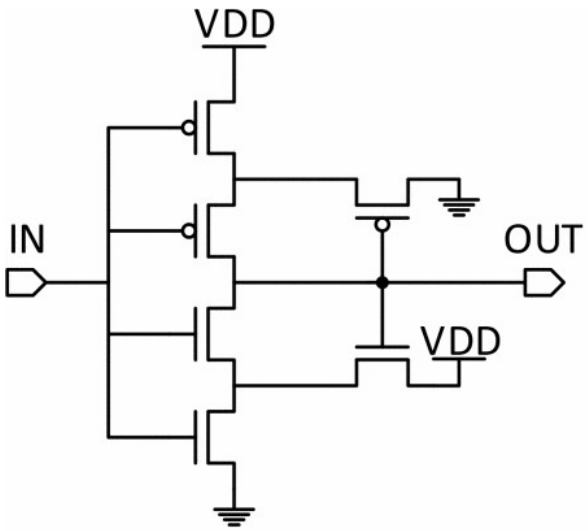


Skewed Inverter

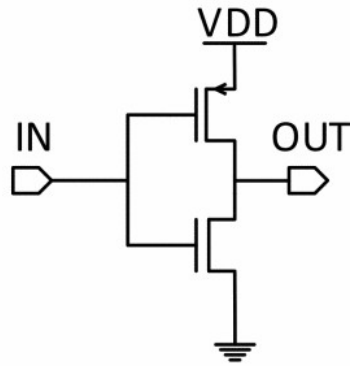
Threshold Detector

Two possibilities for threshold detectors

- Skewed inverter with fixed switching voltage



Schmitt Trigger

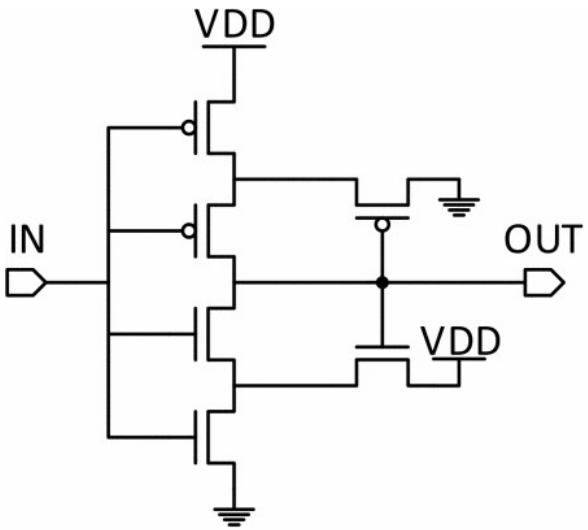


Skewed Inverter

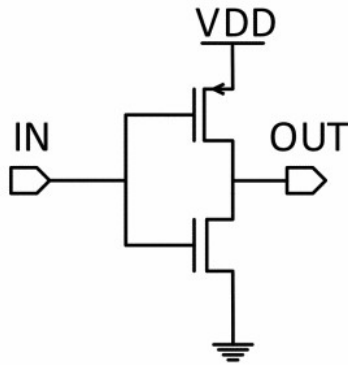
Threshold Detector

Two possibilities for threshold detectors

- Skewed inverter with fixed switching voltage
- Schmitt trigger with hysteresis, i.e. high threshold on rising edge and low threshold on falling edge



Schmitt Trigger



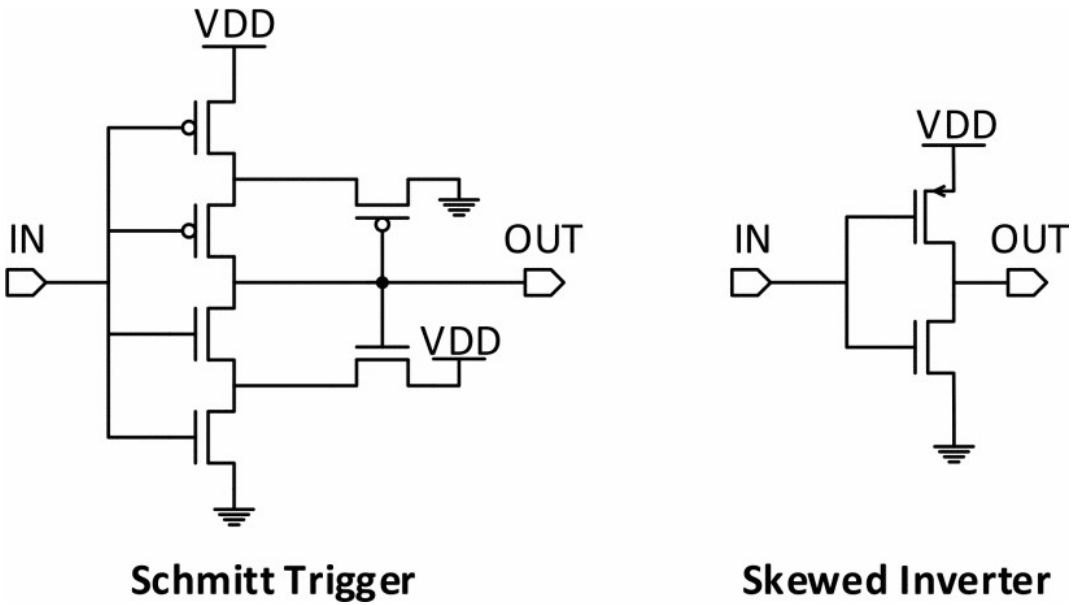
Skewed Inverter

Threshold Detector

Two possibilities for threshold detectors

- Skewed inverter with fixed switching voltage
- Schmitt trigger with hysteresis, i.e. high threshold on rising edge and low threshold on falling edge

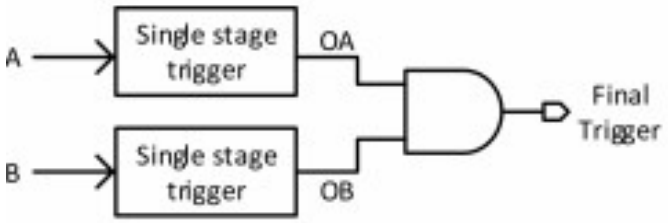
  Paper chooses Schmitt trigger as it extends trigger and retention time



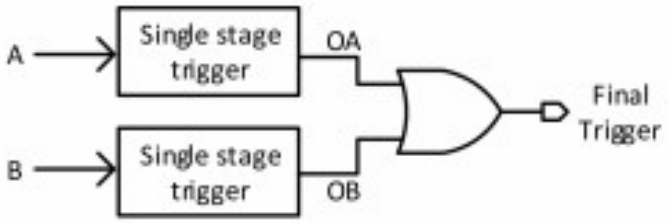
Schmitt Trigger

Skewed Inverter

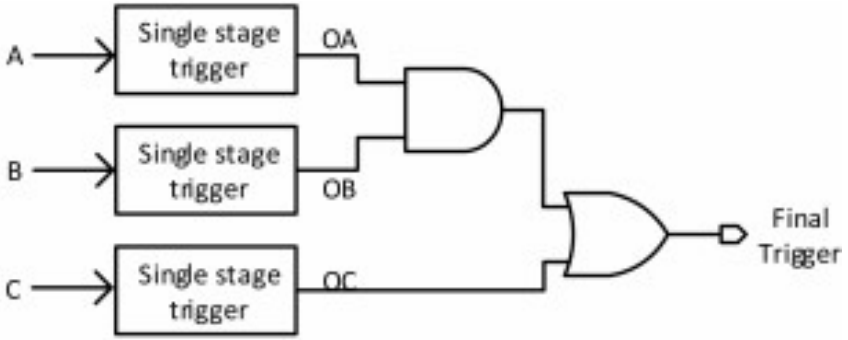
Possibility: Chaining Triggers Together



Final Trigger = OA & OB
Either A or B triggers

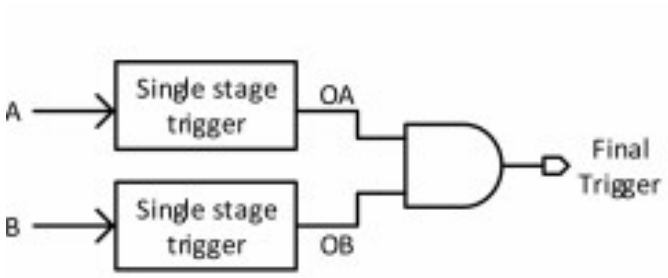


Final Trigger = OA | OB
Both A and B trigger

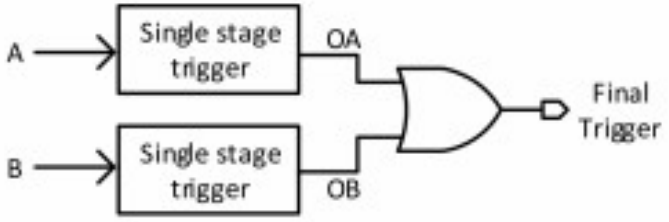


Final Trigger = (OA & OB) | OC
One of A and B trigger, C trigger

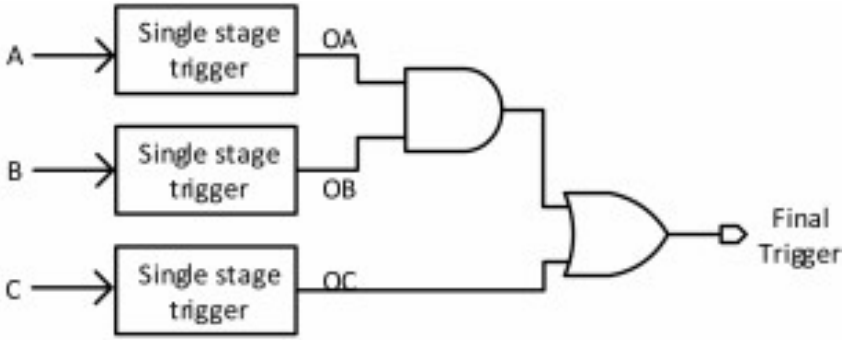
Possibility: Chaining Triggers Together



Final Trigger = OA & OB
Either A or B triggers



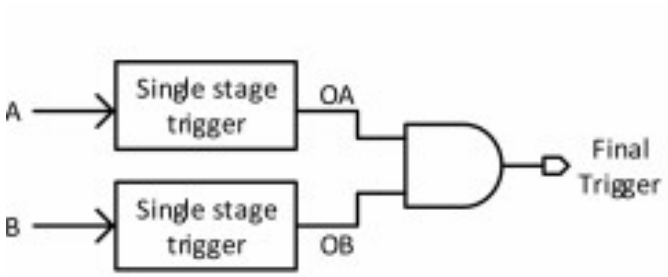
Final Trigger = OA | OB
Both A and B trigger



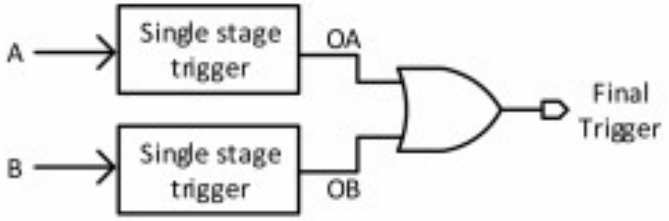
Final Trigger = (OA & OB) | OC
One of A and B trigger, C trigger

- Triggers can be combined to form more complex trigger mechanisms

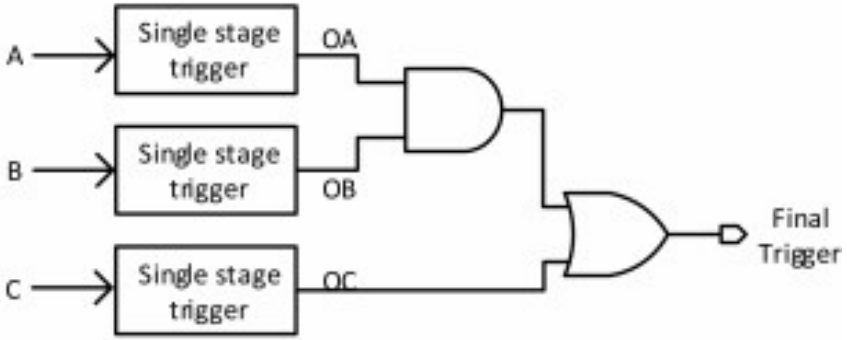
Possibility: Chaining Triggers Together



Final Trigger = OA & OB
Either A or B triggers



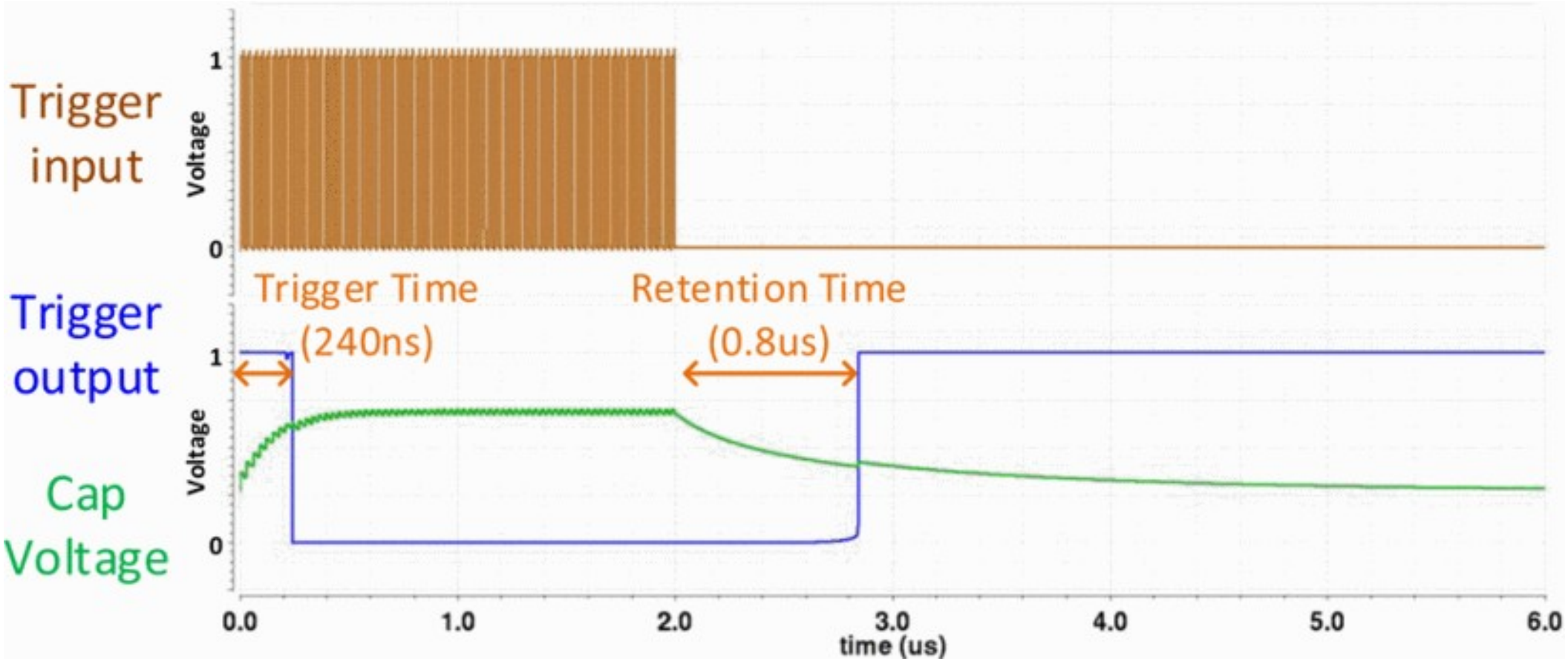
Final Trigger = OA | OB
Both A and B trigger



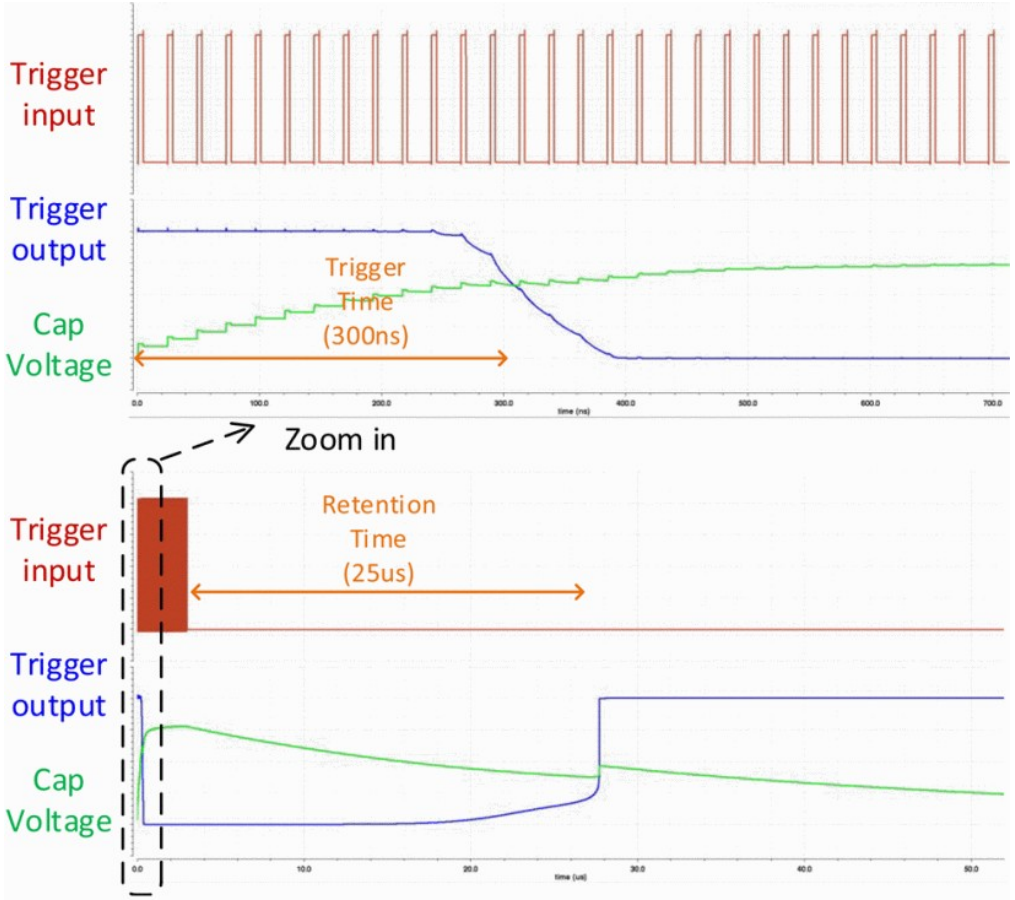
Final Trigger = (OA & OB) | OC
One of A and B trigger, C trigger

- Triggers can be combined to form more complex trigger mechanisms
- Can be used to construct well hidden multi-stage triggers

SPICE Simulation Results

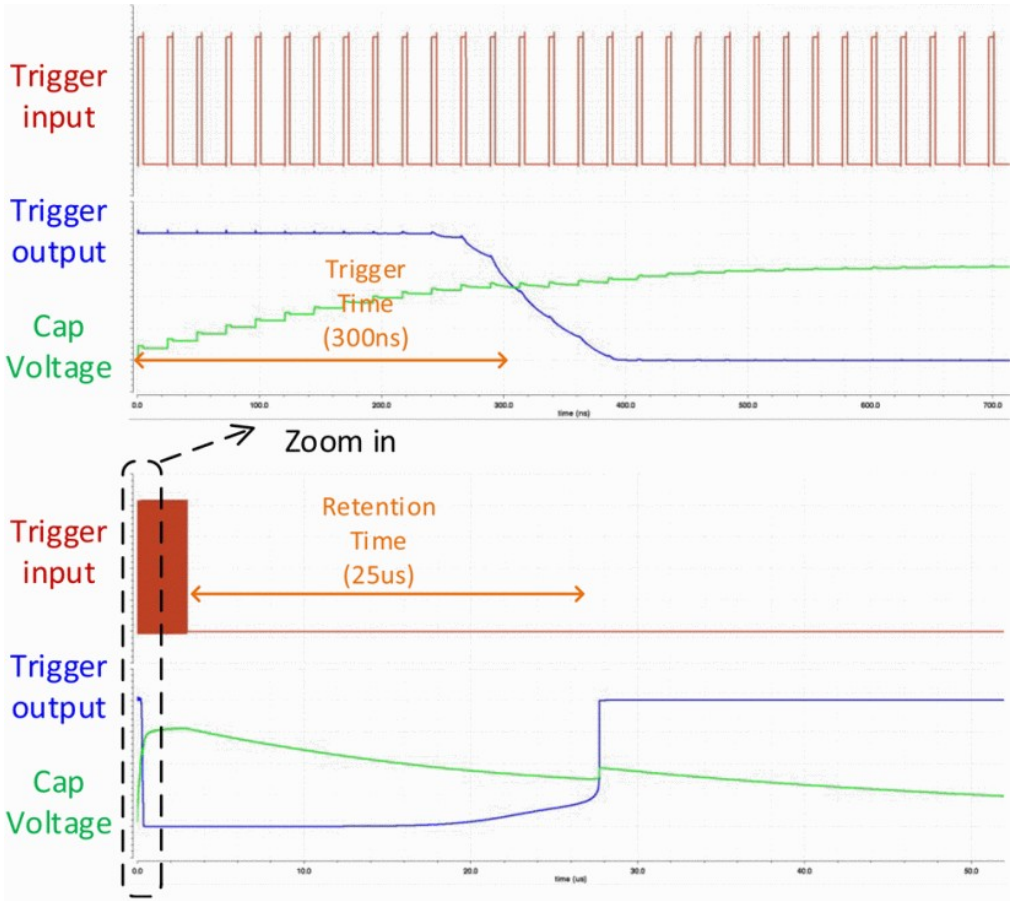


SPICE Using I/O Devices in 65nm CMOS



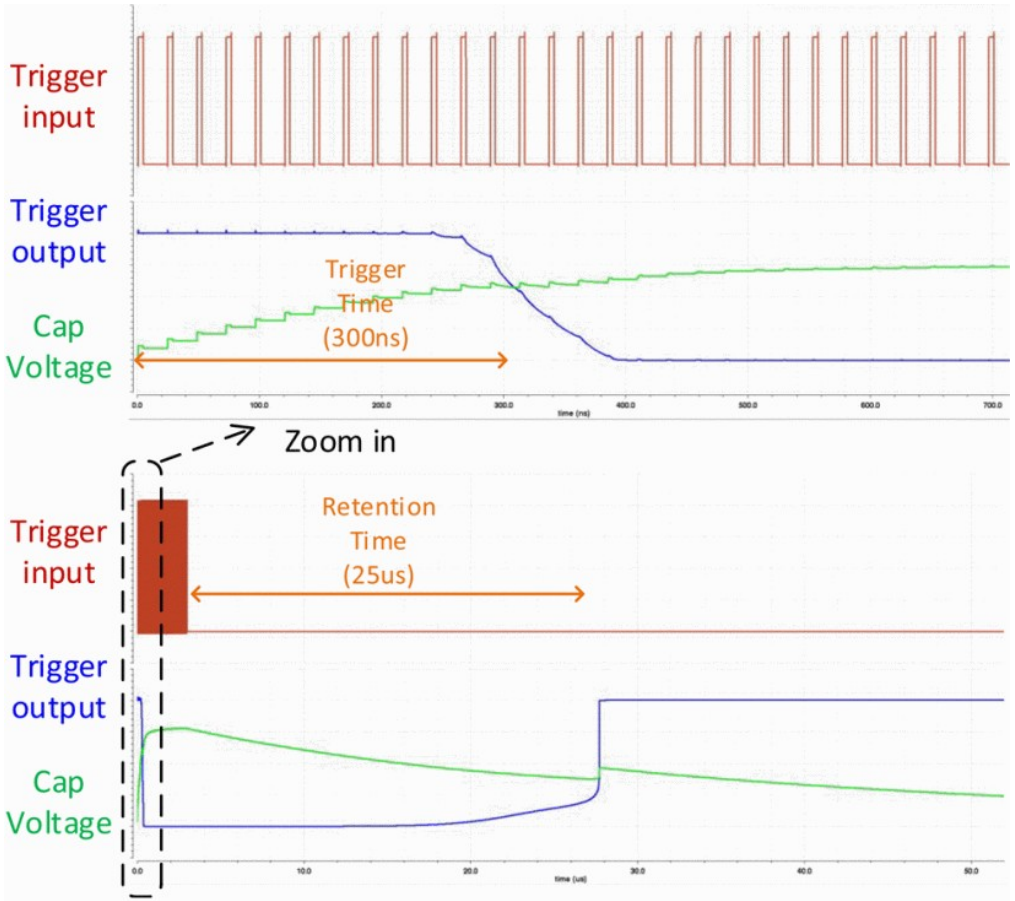
SPICE Using I/O Devices in 65nm CMOS

- To mitigate gate leakage, I/O Device Cells can be used instead of normal standard cells



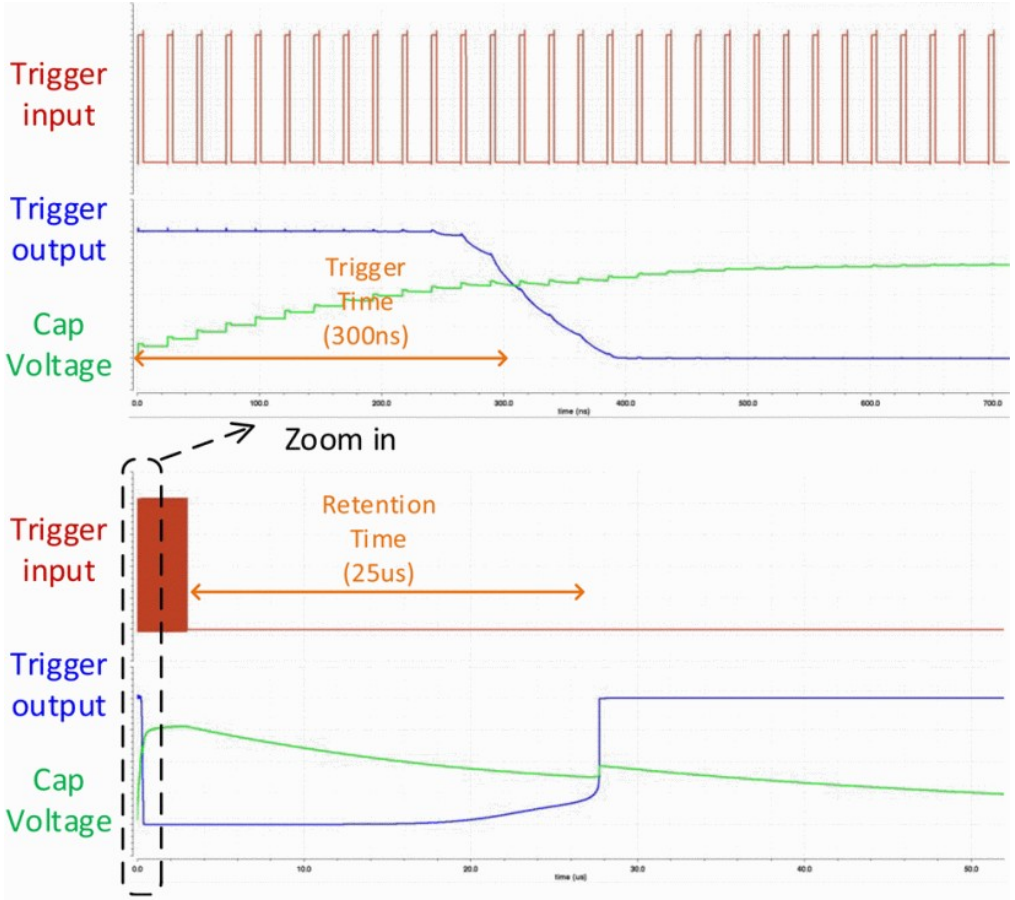
SPICE Using I/O Devices in 65nm CMOS

- To mitigate gate leakage, I/O Device Cells can be used instead of normal standard cells
- Results in more control over trigger and retention times



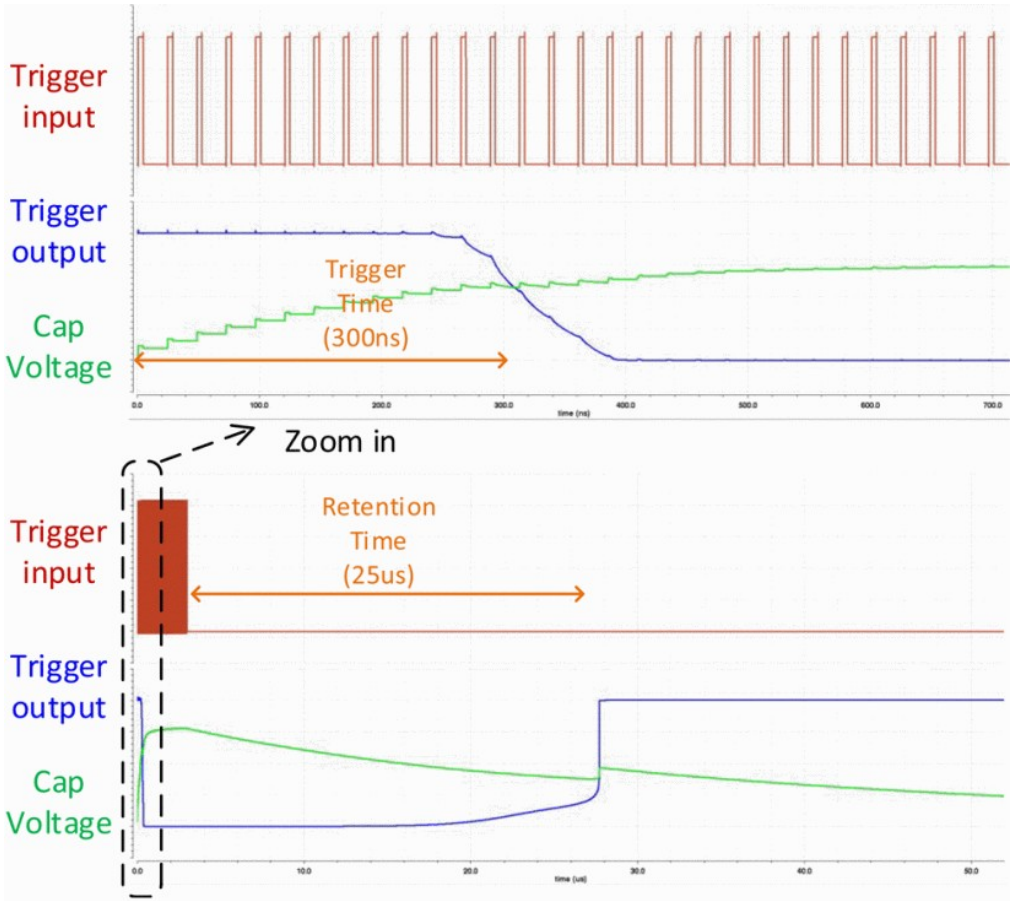
SPICE Using I/O Devices in 65nm CMOS

- To mitigate gate leakage, I/O Device Cells can be used instead of normal standard cells
- Results in more control over trigger and retention times
- Uses slightly more chip area

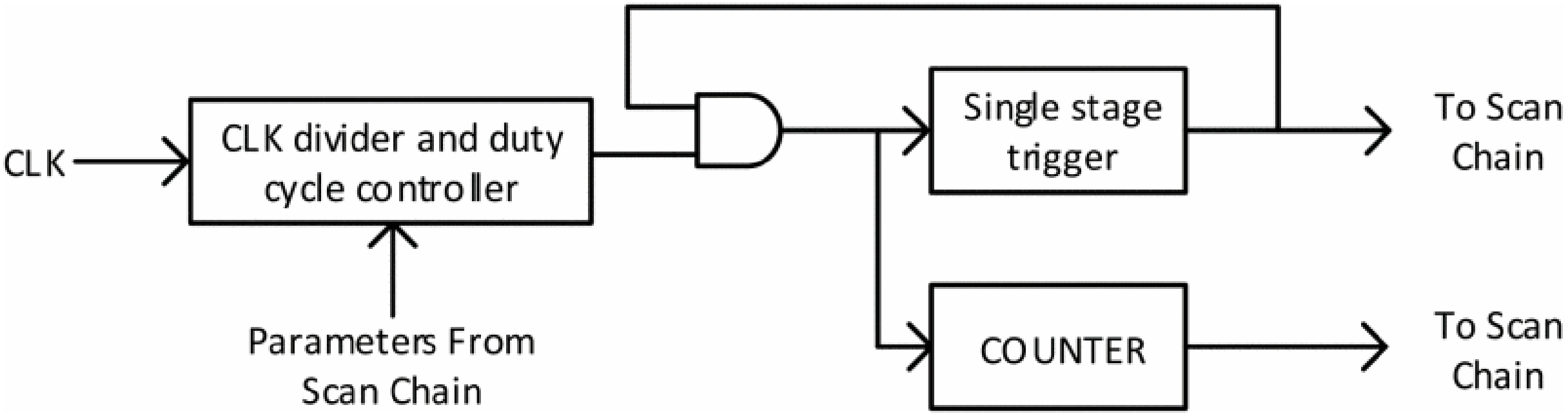


SPICE Using I/O Devices in 65nm CMOS

- To mitigate gate leakage, I/O Device Cells can be used instead of normal standard cells
- Results in more control over trigger and retention times
- Uses slightly more chip area
- Also simulated in 65nm low power CMOS



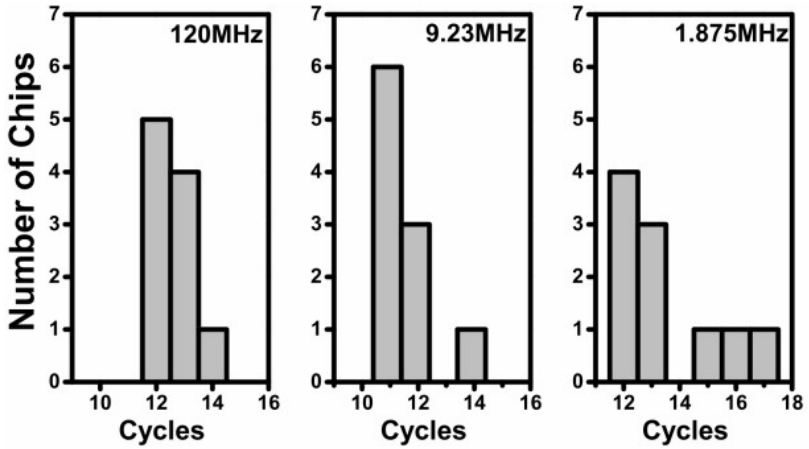
Stand-alone Testing Structure



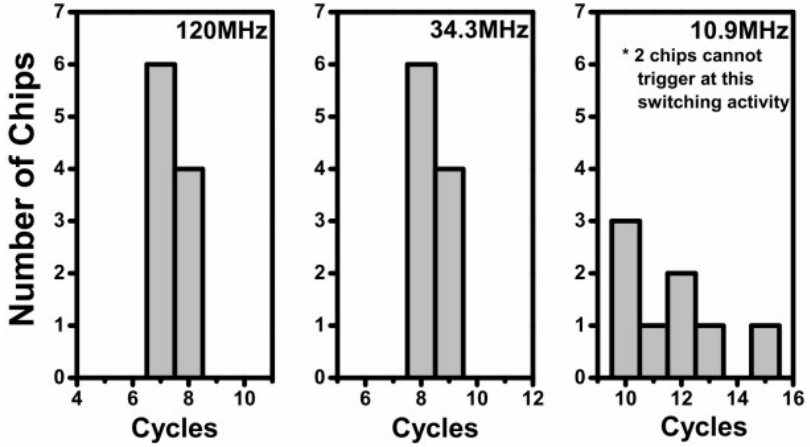
Results Across 10 Chips (1V, 25°C)

Results Across 10 Chips (1V, 25°C)

- Shows number of chips which show a certain trigger time in cycles at different switching frequencies



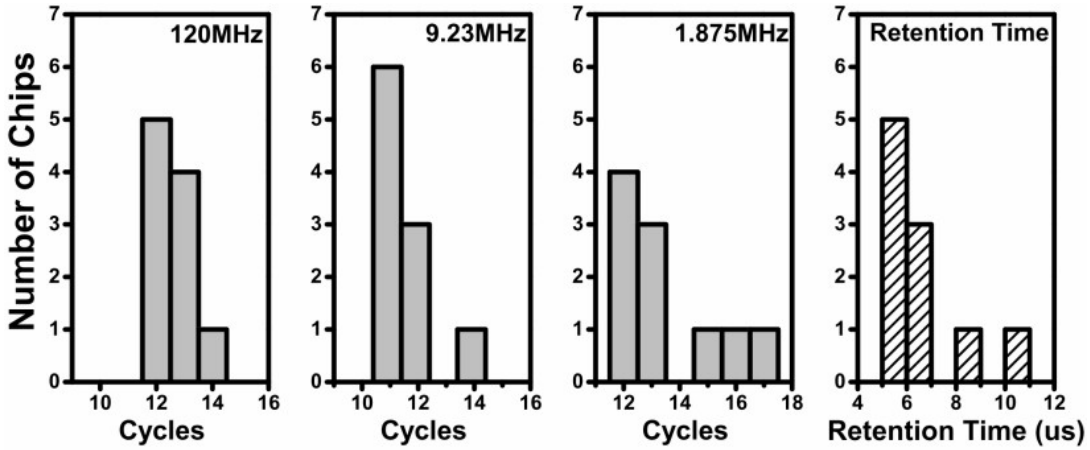
(a) Distribution of analog trigger circuit using IO device



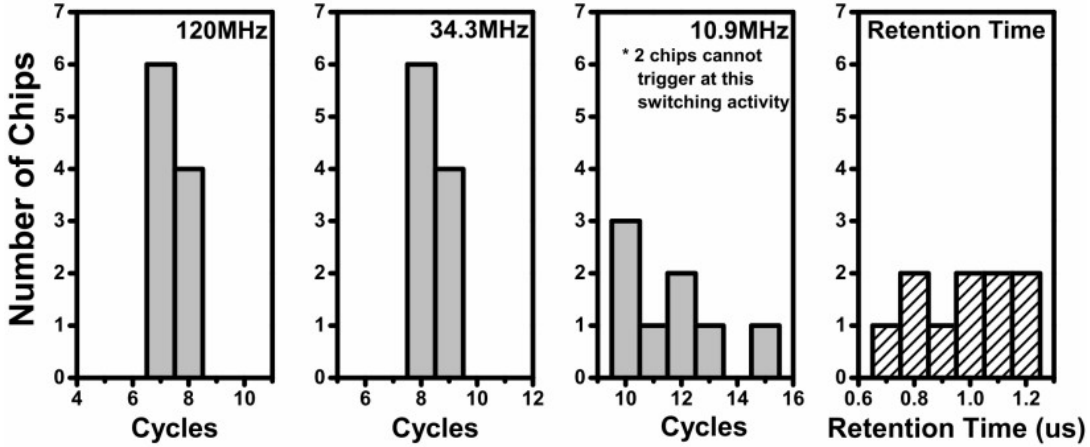
(b) Distribution of analog trigger circuit using only core device

Results Across 10 Chips (1V, 25°C)

- Shows number of chips which show a certain trigger time in cycles at different switching frequencies
- Also shows number of chips which show a certain retention time in µs



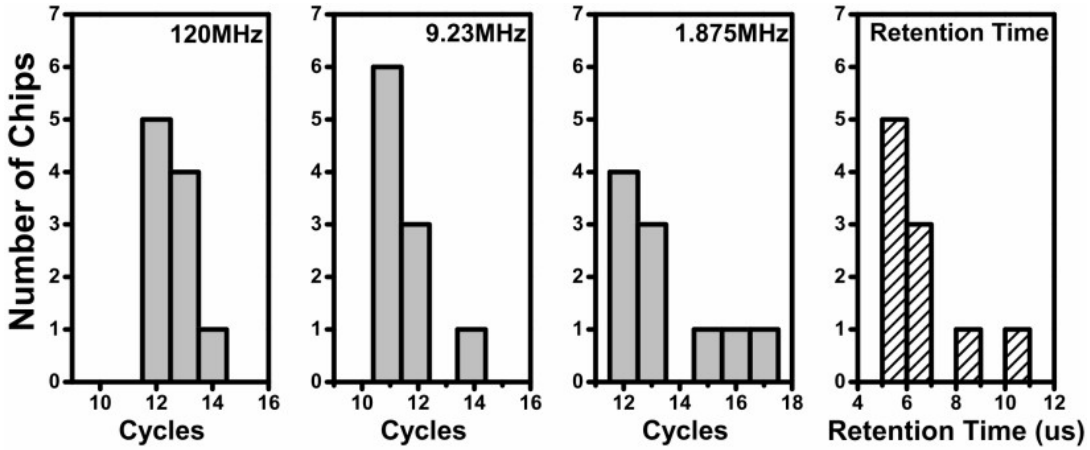
(a) Distribution of analog trigger circuit using IO device



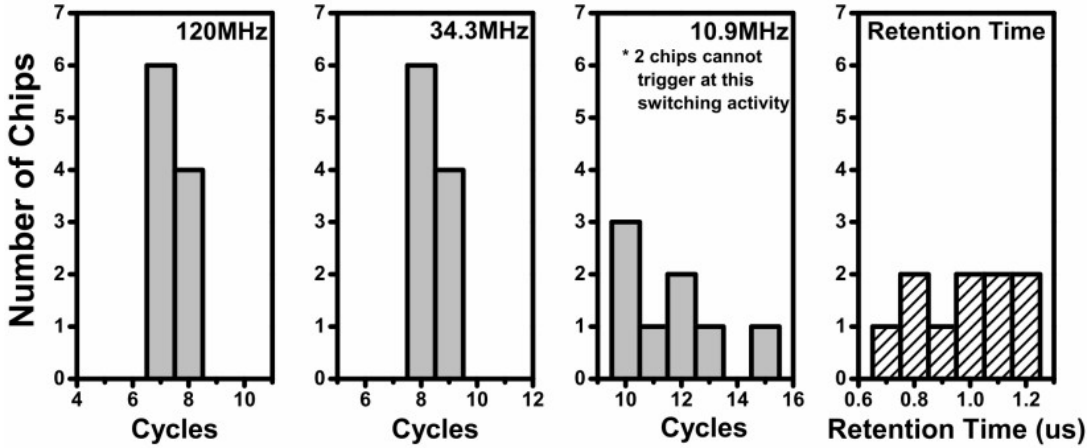
(b) Distribution of analog trigger circuit using only core device

Results Across 10 Chips (1V, 25°C)

- Shows number of chips which show a certain trigger time in cycles at different switching frequencies
- Also shows number of chips which show a certain retention time in µs
- Shows robustness against manufacturing variations



(a) Distribution of analog trigger circuit using IO device

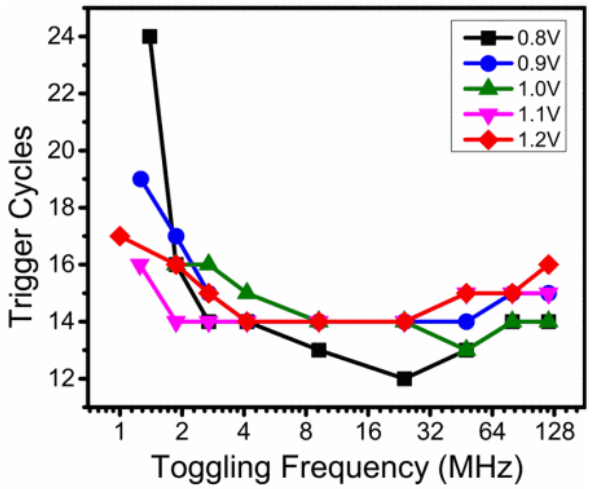


(b) Distribution of analog trigger circuit using only core device

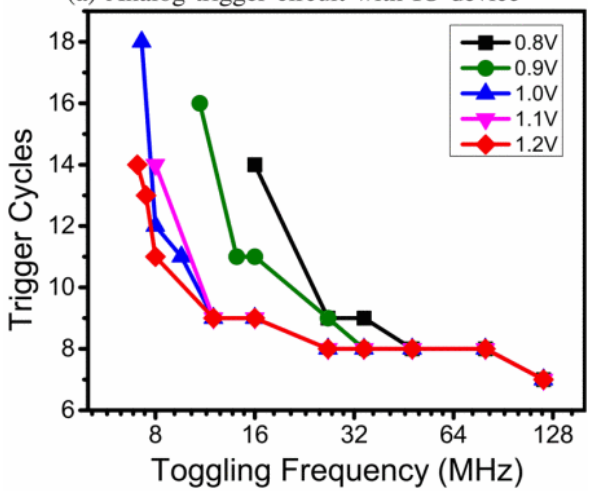
Varying the Voltage

Varying the Voltage

- Shows the trigger time in cycles for a given voltage and frequency



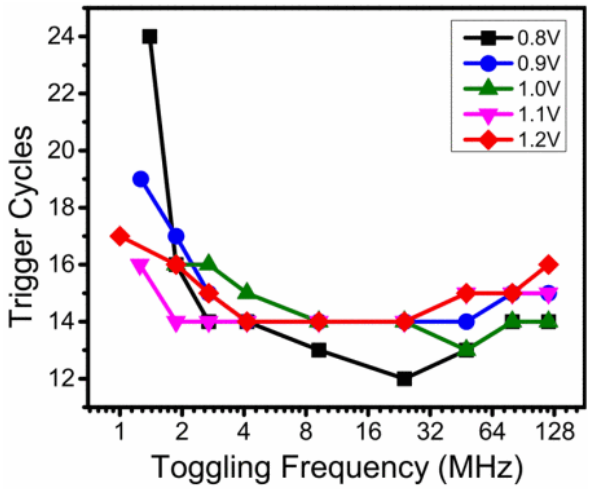
(a) Analog trigger circuit with IO device



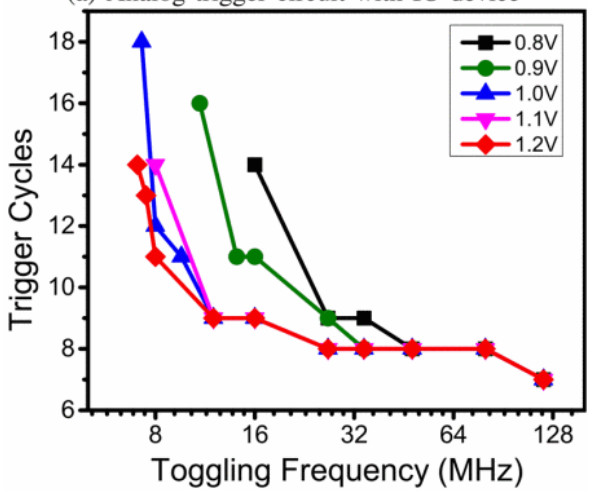
(b) Analog trigger circuit with only core device

Varying the Voltage

- Shows the trigger time in cycles for a given voltage and frequency
- Shows robustness across variations in the supply voltage



(a) Analog trigger circuit with IO device

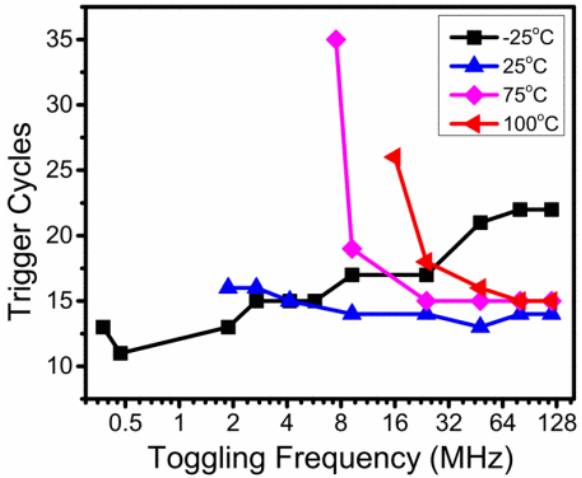


(b) Analog trigger circuit with only core device

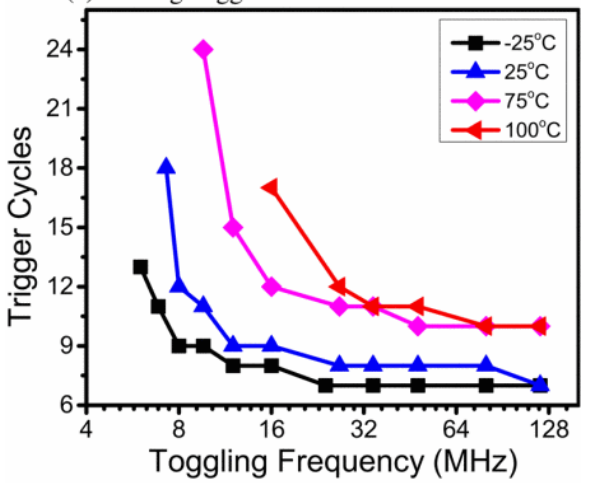
Varying the Temperature

Varying the Temperature

- Shows the trigger time in cycles for a given temperature and frequency



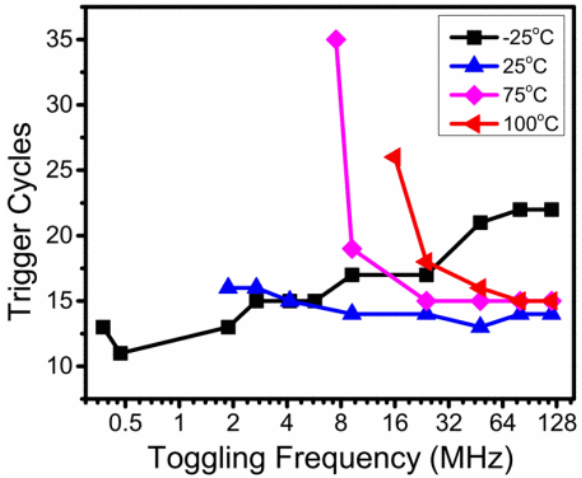
(a) Analog trigger circuit with IO device



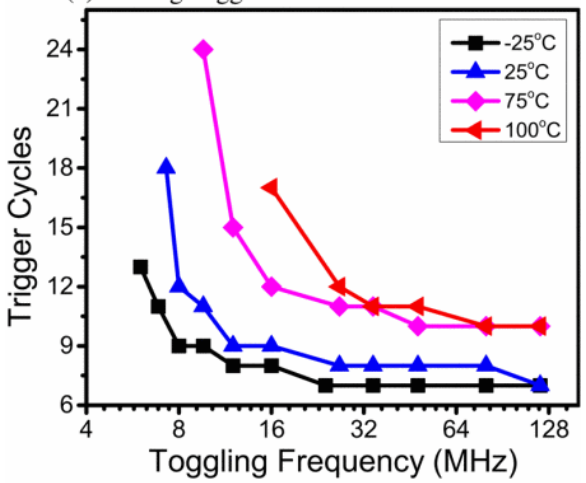
(b) Analog trigger circuit with only core device

Varying the Temperature

- Shows the trigger time in cycles for a given temperature and frequency
- Shows robustness across variations in the ambient temperature



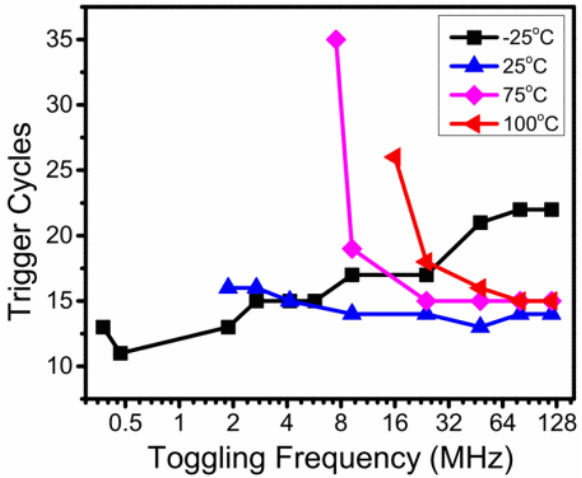
(a) Analog trigger circuit with IO device



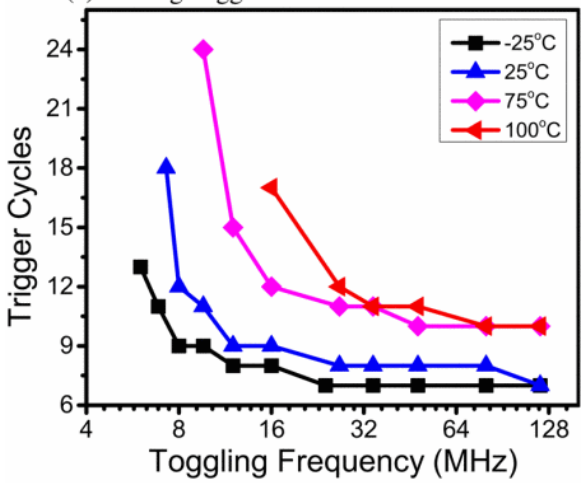
(b) Analog trigger circuit with only core device

Varying the Temperature

- Shows the trigger time in cycles for a given temperature and frequency
- Shows robustness across variations in the ambient temperature
- The paper states that both single and two-stage attacks trigger in all 10 chips over 6 tested temperatures



(a) Analog trigger circuit with IO device



(b) Analog trigger circuit with only core device

Evaluation of Side Channel Information

Backup Slides

Evaluation of Side Channel Information

- Power consumption of the chip measured down to 1 µA at 1V and 25°C

Program	Power (mW)
Standby	6.210
Basic math	23.703
Dijkstra	16.550
FFT	18.120
SHA	18.032
Search	21.960
Single-stage Attack	19.505
Two-stage Attack	22.575
Unsigned Division	23.206

Table III: Power consumption of our test Chip running a variety of benchmark programs.

Evaluation of Side Channel Information

- Power consumption of the chip measured down to 1 µA at 1V and 25°C
- Simulated power consumption of the trigger is 5.3 nW with I/O devices and 0.5 µW without I/O devices at maximum switching activity

Program	Power (mW)
Standby	6.210
Basic math	23.703
Dijkstra	16.550
FFT	18.120
SHA	18.032
Search	21.960
Single-stage Attack	19.505
Two-stage Attack	22.575
Unsigned Division	23.206

Table III: Power consumption of our test Chip running a variety of benchmark programs.

Evaluation of Side Channel Information

- Power consumption of the chip measured down to 1 µA at 1V and 25°C
- Simulated power consumption of the trigger is 5.3 nW with I/O devices and 0.5 µW without I/O devices at maximum switching activity
- Well below normal power fluctuations

Program	Power (mW)
Standby	6.210
Basic math	23.703
Dijkstra	16.550
FFT	18.120
SHA	18.032
Search	21.960
Single-stage Attack	19.505
Two-stage Attack	22.575
Unsigned Division	23.206

Table III: Power consumption of our test Chip running a variety of benchmark programs.

Evaluation of Side Channel Information

- Power consumption of the chip measured down to 1 µA at 1V and 25°C
- Simulated power consumption of the trigger is 5.3 nW with I/O devices and 0.5 µW without I/O devices at maximum switching activity
- Well below normal power fluctuations
- Temperature and propagation delays are nearly unaffected by A2 as it is as small as one gate

Program	Power (mW)
Standby	6.210
Basic math	23.703
Dijkstra	16.550
FFT	18.120
SHA	18.032
Search	21.960
Single-stage Attack	19.505
Two-stage Attack	22.575
Unsigned Division	23.206

Table III: Power consumption of our test Chip running a variety of benchmark programs.

Is X86 Safe From A2?

Backup Slides

Is X86 Safe From A2?

- The authors expect A2 to be easier to implement in X86 as in OR1200

Is X86 Safe From A2?

- The authors expect A2 to be easier to implement in X86 as in OR1200
- X86 has likely more possible target registers

Is X86 Safe From A2?

- The authors expect A2 to be easier to implement in X86 as in OR1200
- X86 has likely more possible target registers
- X86 has also likely more viable victim wires

Is X86 Safe From A2?

- The authors expect A2 to be easier to implement in X86 as in OR1200
- X86 has likely more possible target registers
- X86 has also likely more viable victim wires
- Due to the complexity of X86, A2 should also be more difficult to detect

Is X86 Safe From A2?

- The authors expect A2 to be easier to implement in X86 as in OR1200
- X86 has likely more possible target registers
- X86 has also likely more viable victim wires
- Due to the complexity of X86, A2 should also be more difficult to detect
- The only expected challenge is maintaining controllability over the many redundant functional units in X86