A2: Analog Malicious Hardware

Kaiyuan Yang, Matthew Hicks, Qing Dong, Todd Austin, Dennis Sylvester Department of Electrical Engineering and Computer Science University of Michigan

IEEE Symposium on Security and Privacy (SP) 2016

Jascha Krattenmacher

Motivation

Current trend: smaller transistors

- Beneficial for performance, power usage
- Expensive to build new production lines
    - Most hardware companies outsource fabrication
        - Vulnerable to fabrication time attack

Key idea

Create an undetectable dopant-level trojan to get superuser privileges*

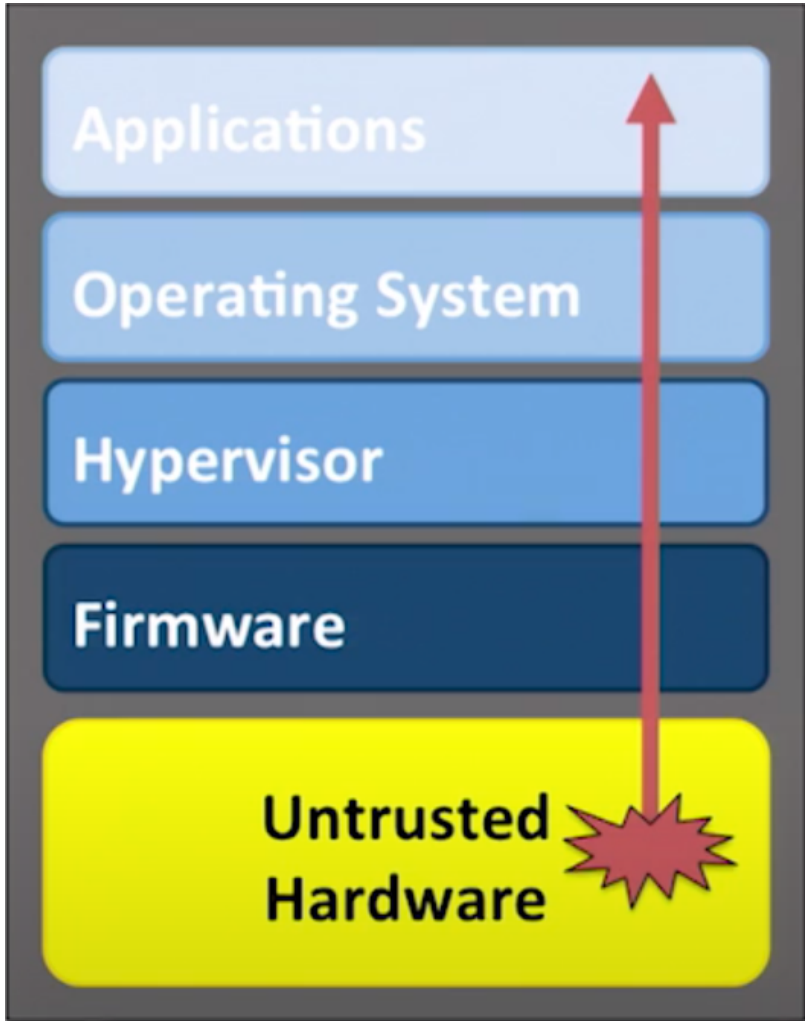
* S. T. King, et al. “Designing and implementing malicious hardware,” in LEET 2008

Fabrication time attack

Why is it dangerous?

Every software implementation is dependent on the hardware.

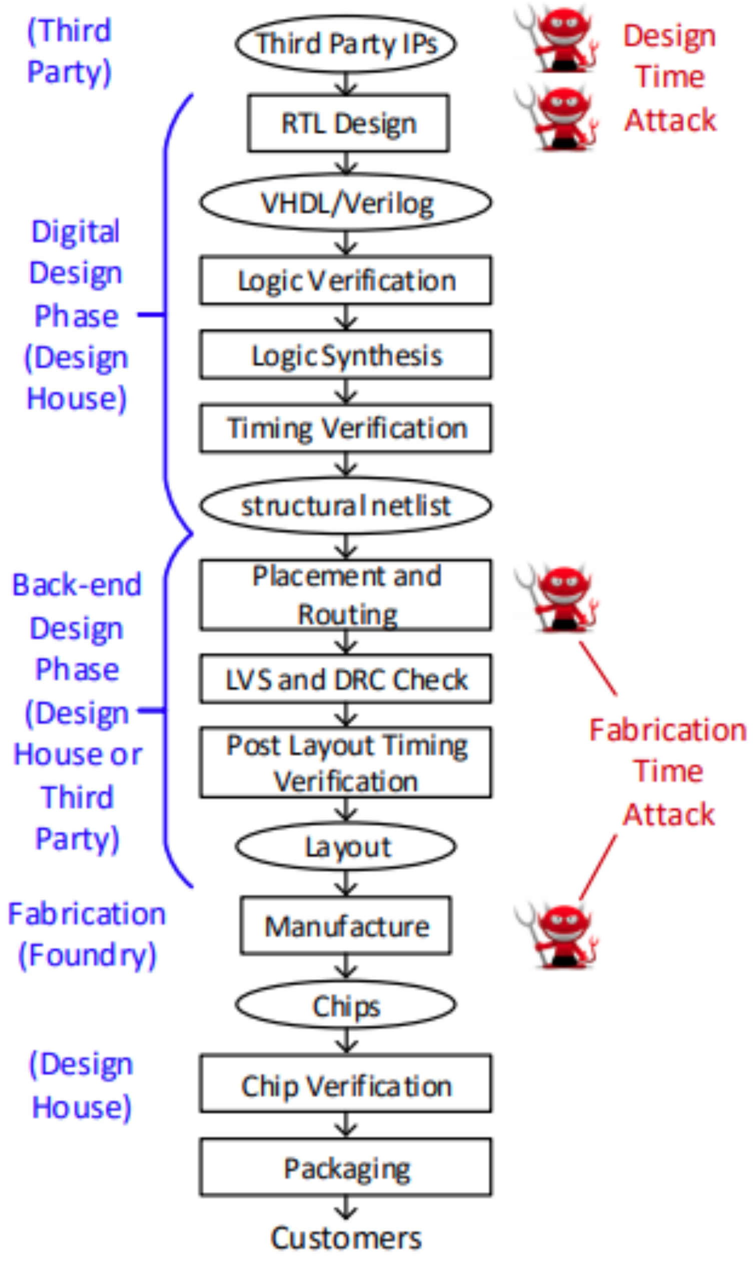
Software has almost no way to check, if the hardware works as intended



Fabrication time attack

Limitations for attacker:

- Cannot increase dimension of the Chip
- Cannot change position of existing parts
- Can use free space and add anything he wants



Dopant-level trojan

- Opens possibility to alter security critical information in the hardware
- When activated it sets a specific pin to 1 or 0 (or multiple)
- Implemented in hardware
- Camouflaged as ordinary hardware

Defenses against dopant-level trojan

1. Visual inspection
- Measures increase in temperature, power usage etc.
- Measures propagation delay on chip

2. Dynamic & static analysis
- Use of benchmark tests

Dopant-level trojan

A good implementation need to fulfill the following:

- Functional
- Small
- Low Power
- Negligible timing perturbation
- Standard cell compatibility

Conventional attacks in malicious hardware

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

Register to store count

Additional functional units to add and compare

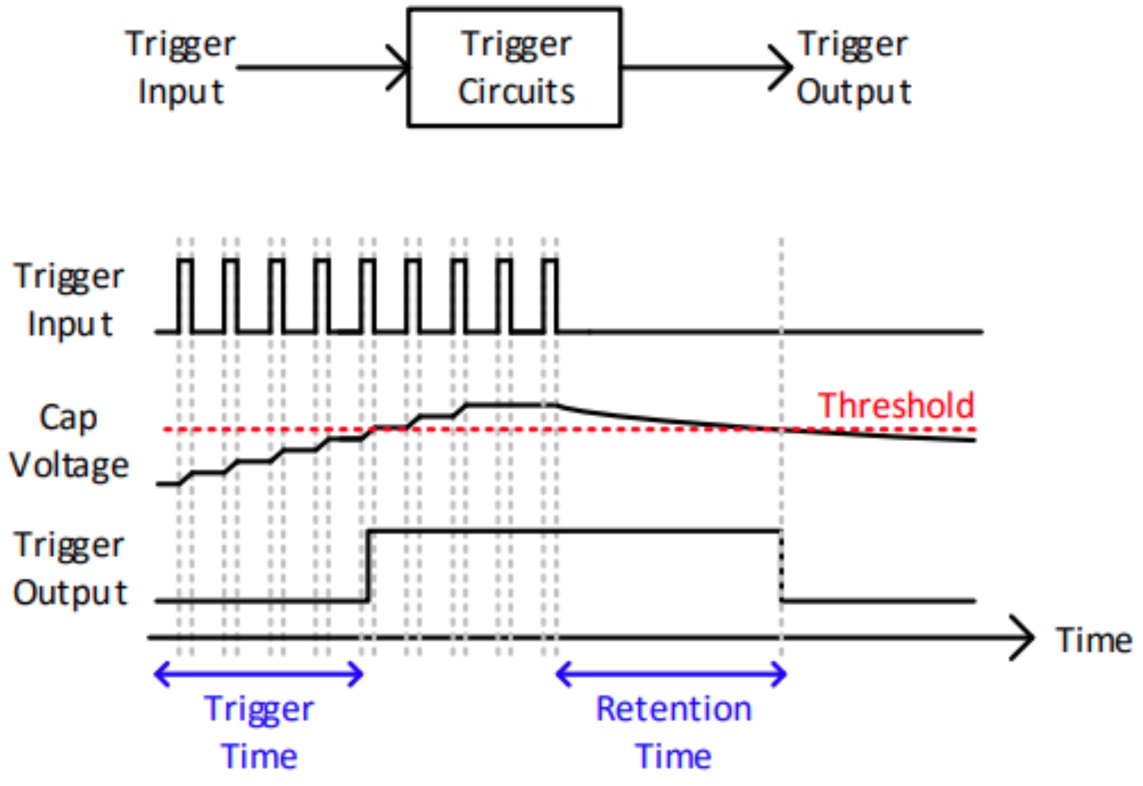
Control unit

RBACE = rare, but attacker controllable event

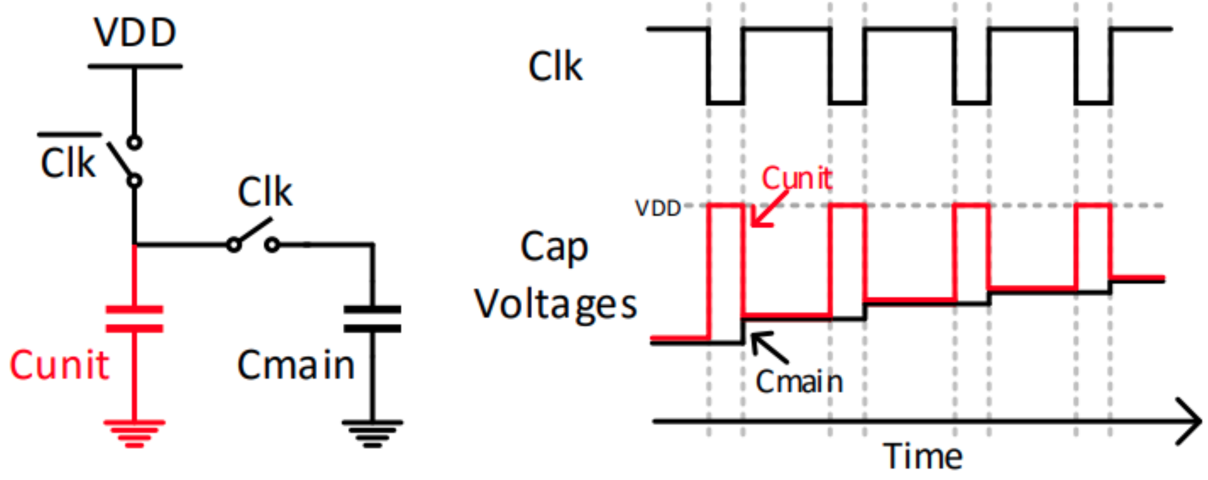
Using Capacitor as counter

Relatively small

Capacitor leak charge



Modified capacitor

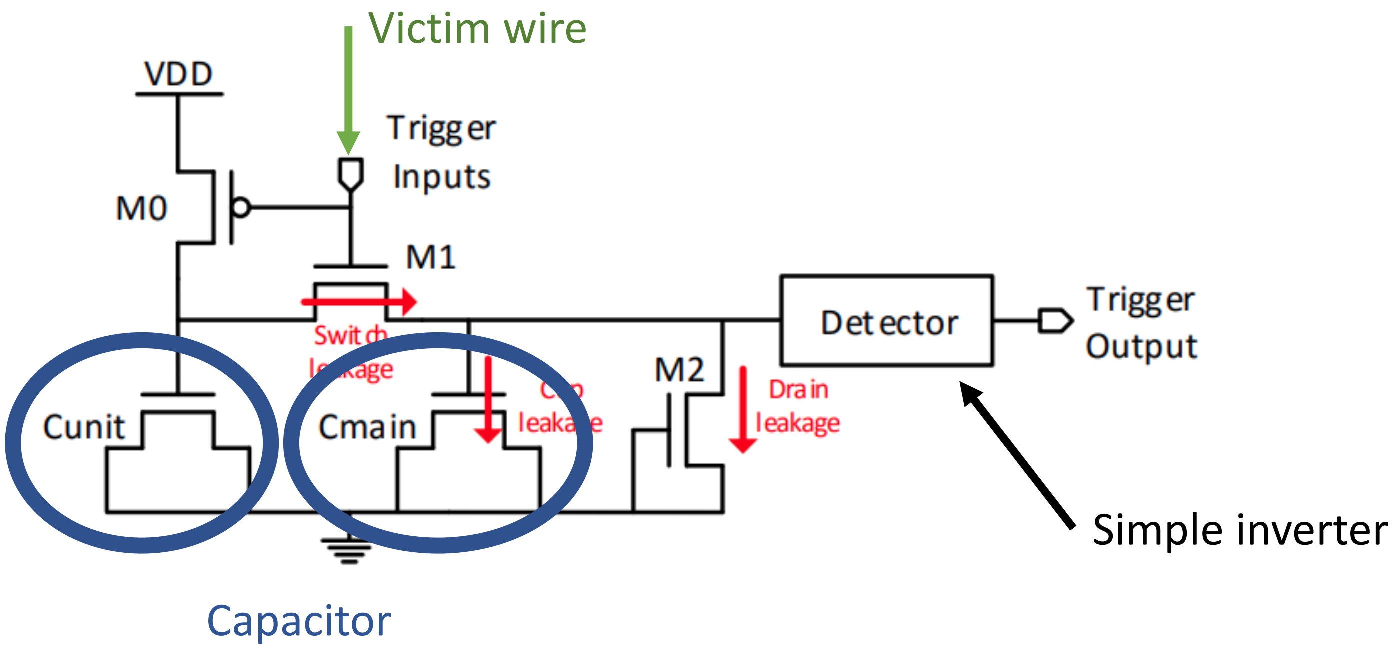


$$\Delta V = \frac{C_{unit} * (VDD - V_0)}{C_{unit} + C_{main}}$$

C unit

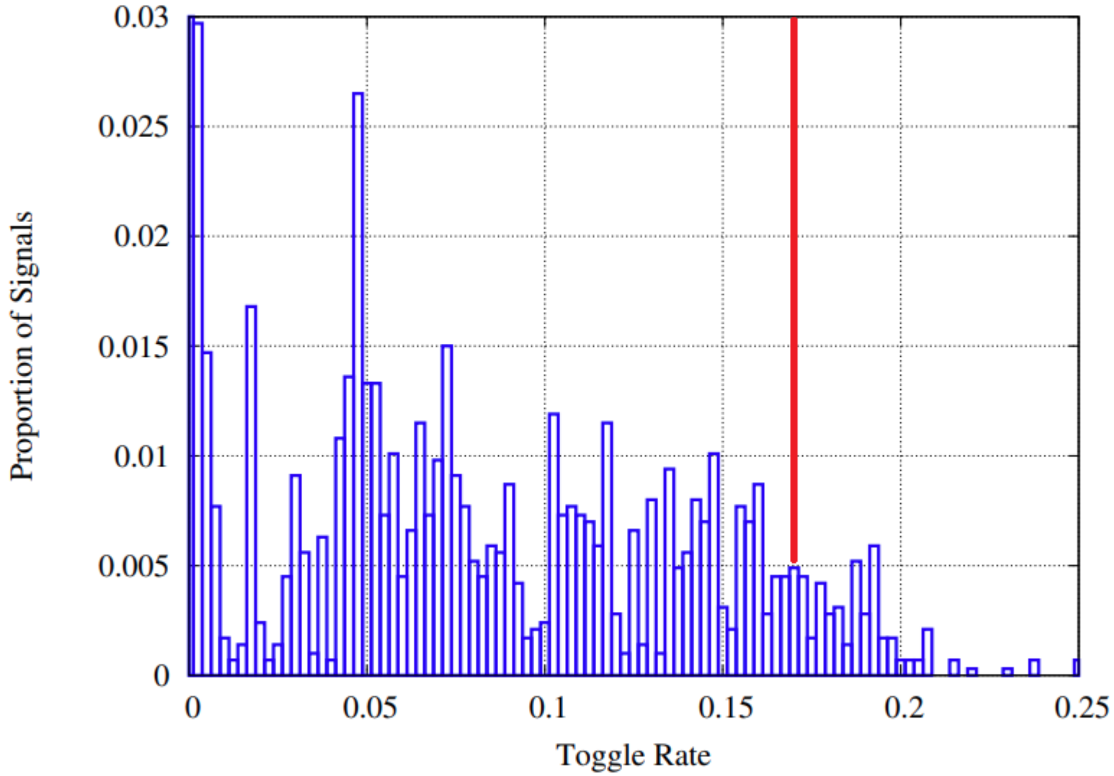
C main

Modified capacitor

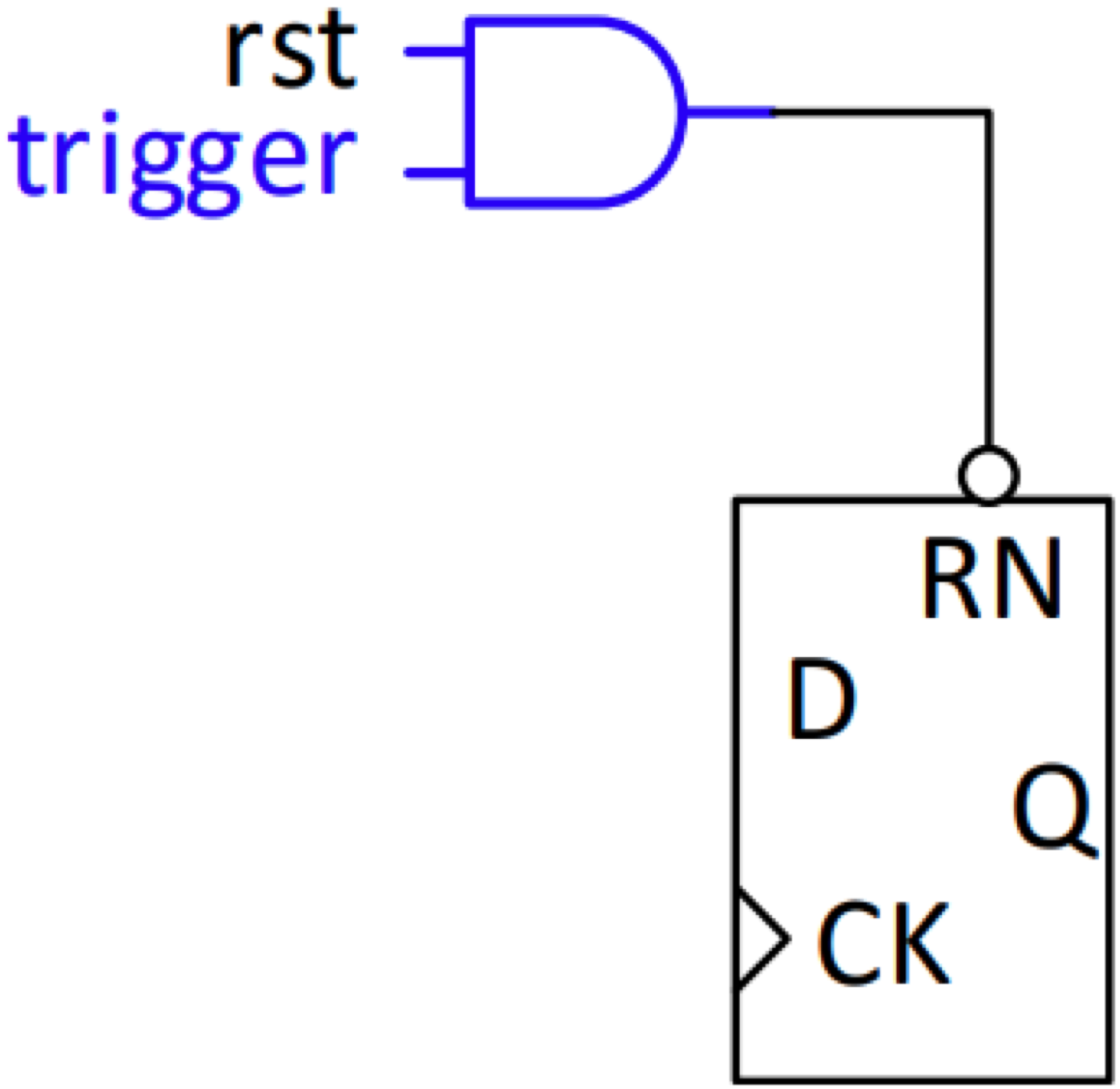


Victim wire

- Good victim wires are rarely used in ordinary use cases
- Can easily be activated with a user program



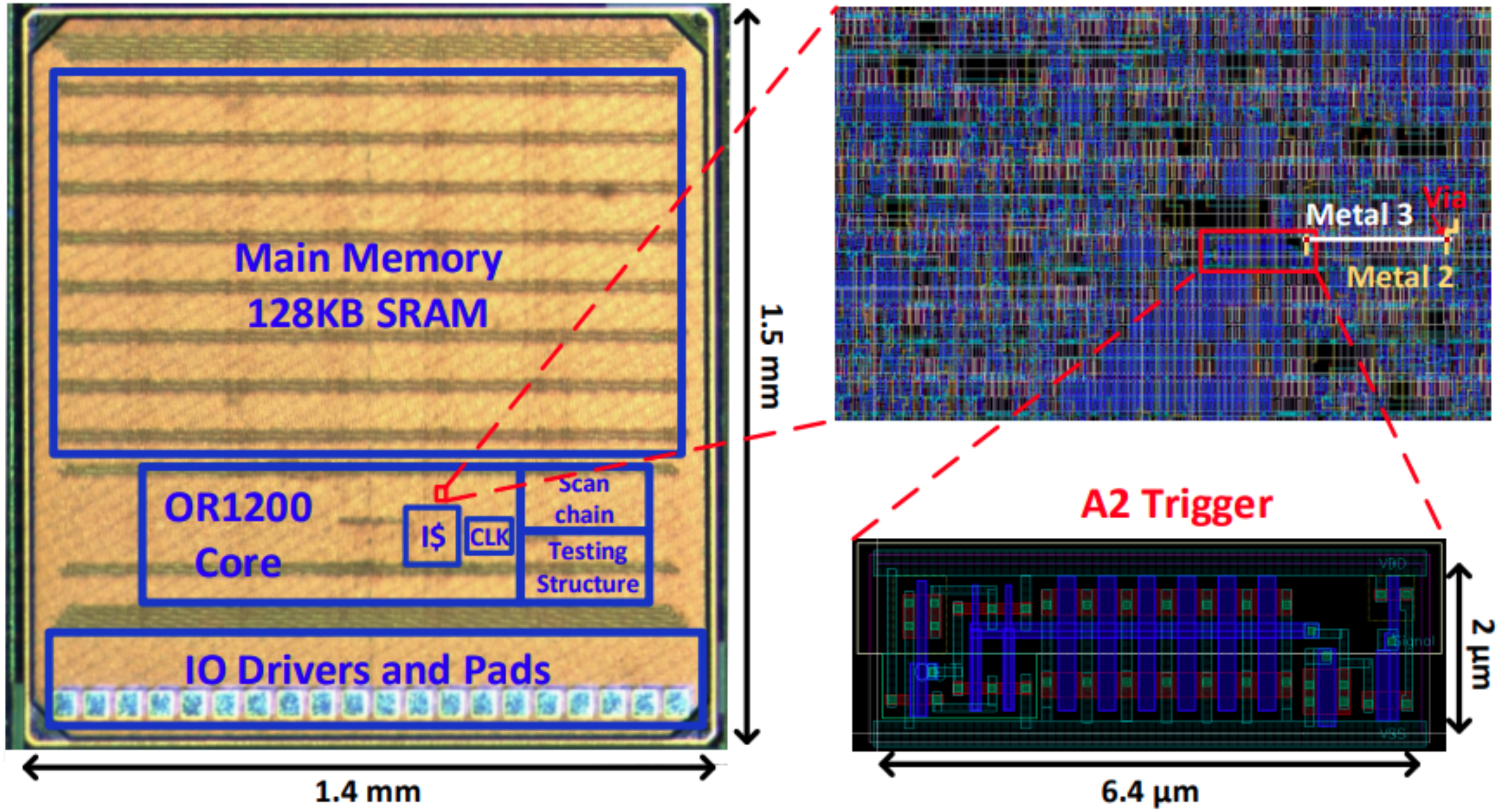
Simulation of benchmark programs



Trigger an attack

- Final goal: Change a security critical pin to a specific value
- Use the already existing set/reset flipflops that are used during startup of a system
- Could use multiple trigger and combine them

Implementation



OR1200 open-source processor

Single and multi-stage trigger

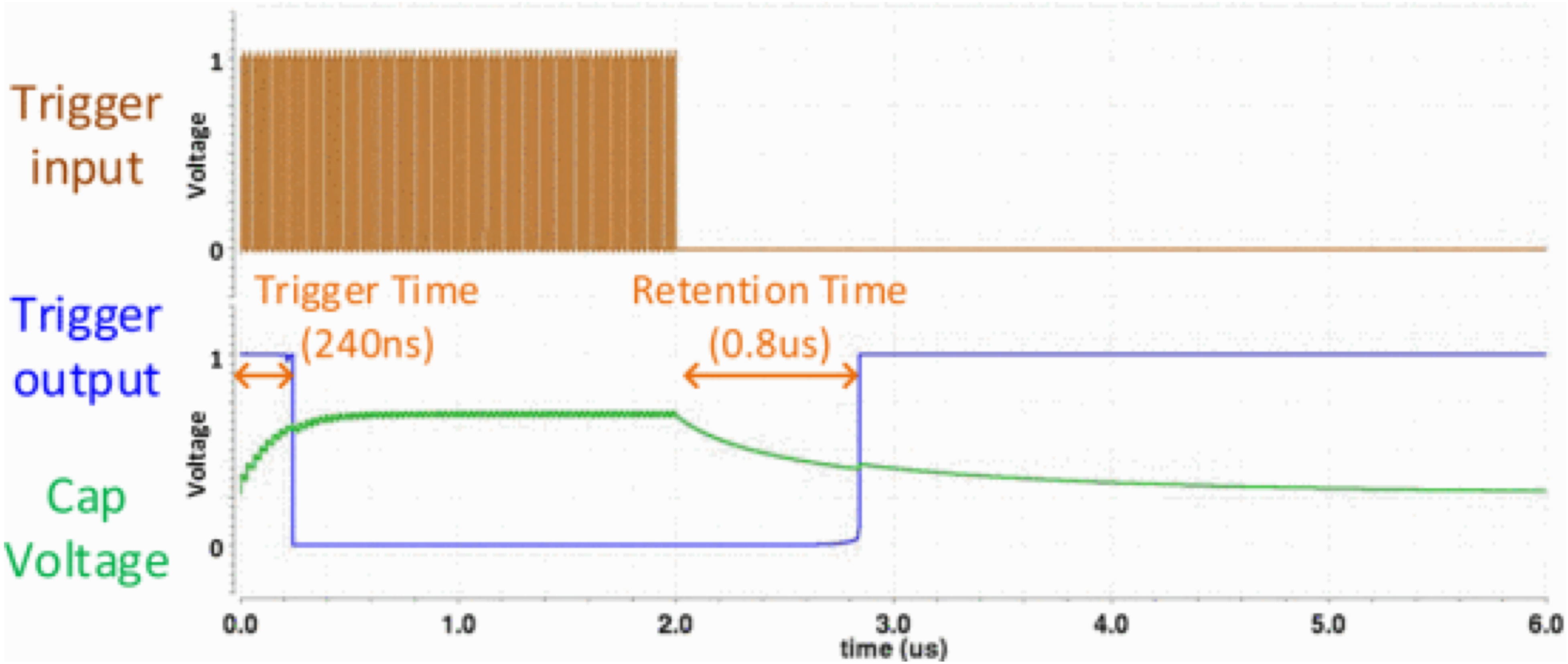
Single trigger

{r0 is a non-zero register but reads as zero in user mode}
Initialize SR[0]=0 {initialize to user mode}
while Attack_Success==0 **do**
  $i \leftarrow 0$
  while $i < 500$ **do**
    $z \leftarrow 1/0$ Victim wire: Division by zero
    $i \leftarrow i + 1$
  end while
  if *read(special register r0)* $\neq 0$ **then**
    $Attack\_Success \leftarrow 1$
  end if
end while

Multi-stage trigger

```
{r0 is a non-zero register but reads as zero in user mode}
Initialize SR[0]=0                      {initialize to user mode}
while Attack_Success==0 do
    i ← 0
    while i < 500 do
        z ← a/b                         {signed division}
        z ← c/d                         {unsigned division}
        i ← i + 1
    end while
    if read(special register r0) ≠ 0 then
        Attack_Success ← 1
    end if
end while
```

Result – SPLICE simulation



Results

- Functional
- Small
- Low Power
- Negligible timing perturbation
- Standard cell compatibility

Temperatures from -25°C to 100°C

Frequencies from 0.5MHz to 120 MHz

None of the 5 benchmark programs triggered an attack

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
w/ IO device	120.00	12.6	14
w/ IO device	9.23	11.6	13
w/ IO device	1.88	13.5	12

Results

- Functional
- Small
- Low Power
- Negligible timing perturbation
- Standard cell compatibility

Used 2.1mm^2 →less than 0.008% of the chip

1 additional gate (previously known 25 gates, 80 μm^2

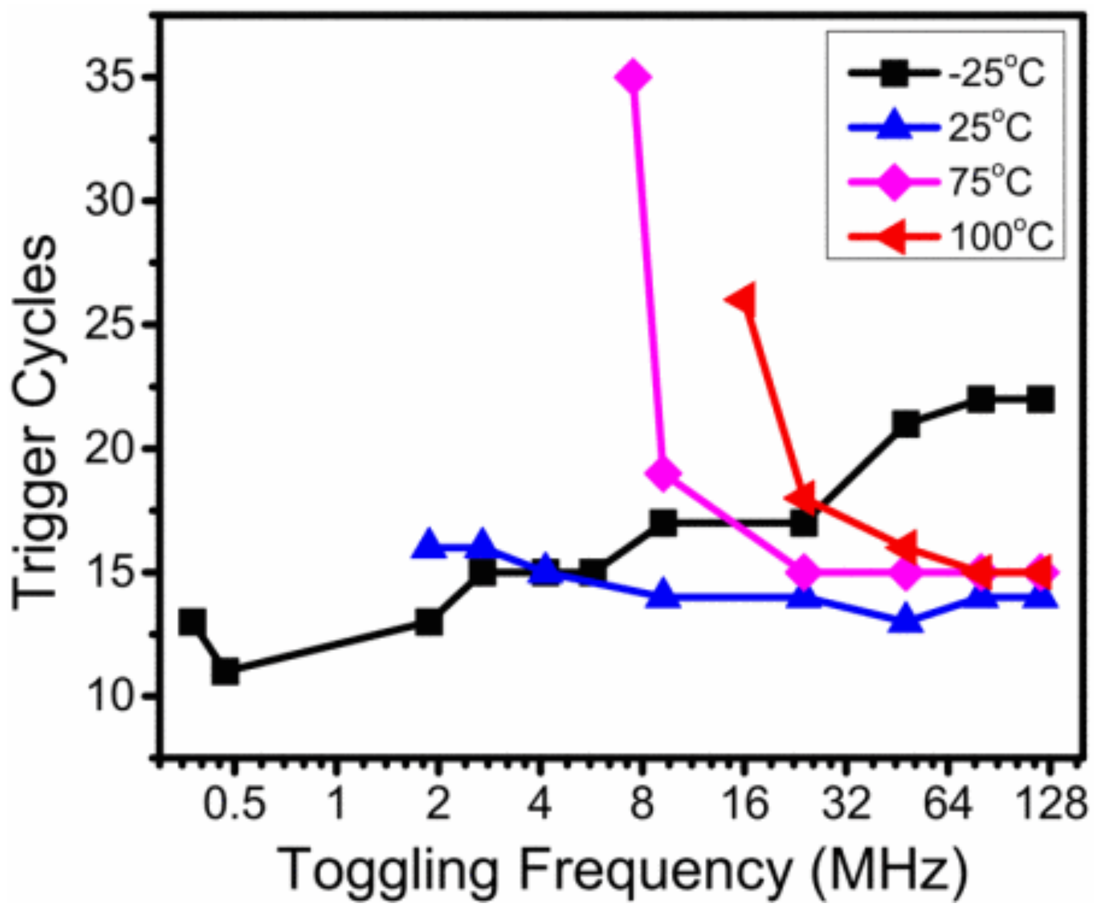
At most 0.5 μW (triggering victim wire all the time)

1.2 ps delay on victim wire → 0.033% of a 4ns clock cycle (250 MHz)

Only uses 2 standard cells to fit

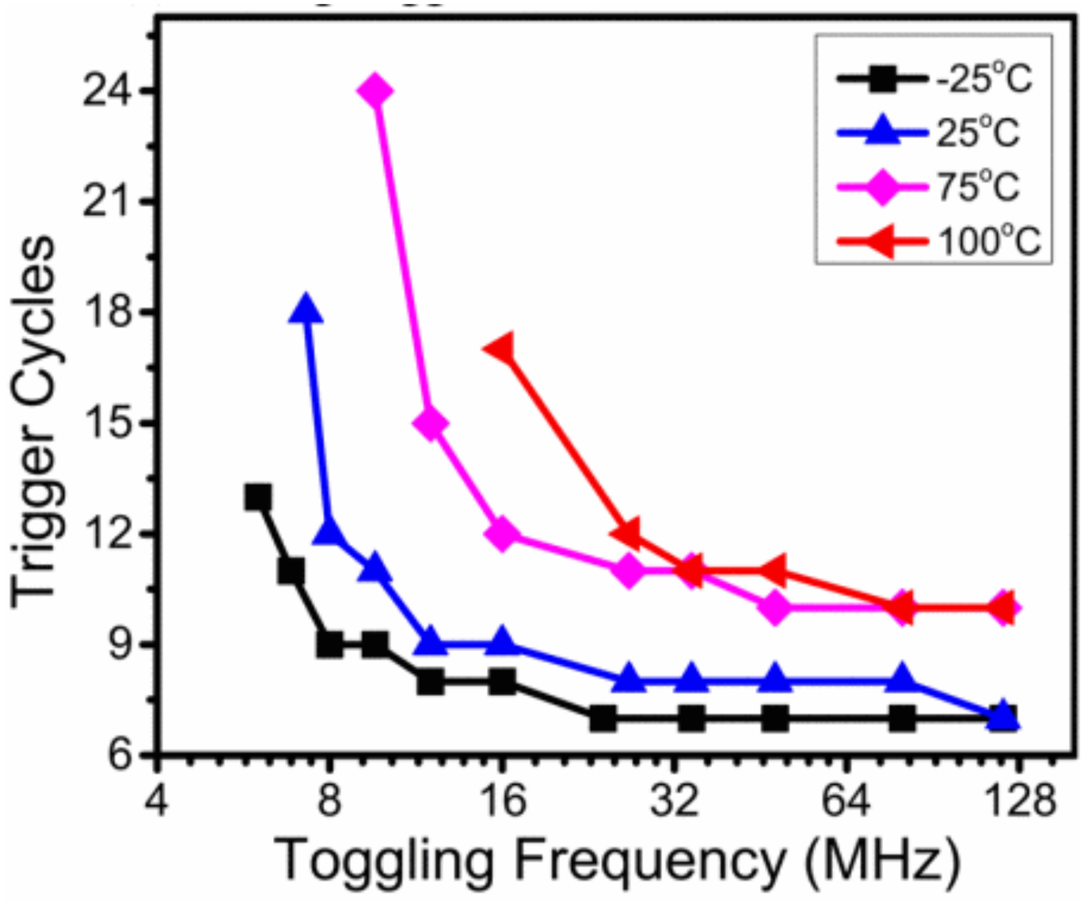
Temperature dependency

Analog trigger circuit with IO device



Temperature dependency

Analog trigger with only core device



OR1200 vs X86

- X86:
- More free space
- More complex operations
- Redundant functional units

Questions?

Strength

Disturbingly easy concept

Well written (easy to understand)

Discusses further implementations (x86)

Weakness

Never mentioned how easy it is to get a software onto a computer with a malicious chip

Not discuss whether such an attack would be noticed after it is done

Discussion

- How easy is it to run such a program on our devices? (Assume Mr. Mutlu has malicious hardware on his computer. How would you get the program to run on his computer?)
- Can you think of other attack that can be done by altering the hardware?

Hardware Trojans in Wireless Cryptographic ICs: Silicon Demonstration & Detection Method Evaluation

Yu Liu*, Yier Jin†, and Yiorgos Makris*
*Department of Electrical Engineering, The University of Texas at Dallas
†Department of Electrical Engineering and Computer Science, University of Central Florida

IEEE International Conference on Computer-Aided Design 2013

Discussion

- How easy is it to run such a program on our devices? (Assume Mr. Mutlu has malicious hardware on his computer. How would you get the program to run on his computer?)
- Can you think of other attack that can be done by altering the hardware?
- Do you think such chips are in use now, we just don't know about it?
- Other ways to protect?

  In the paper:
  two stage manufacture (not trusted vs trusted manufacturer)
  Runtime verification methods