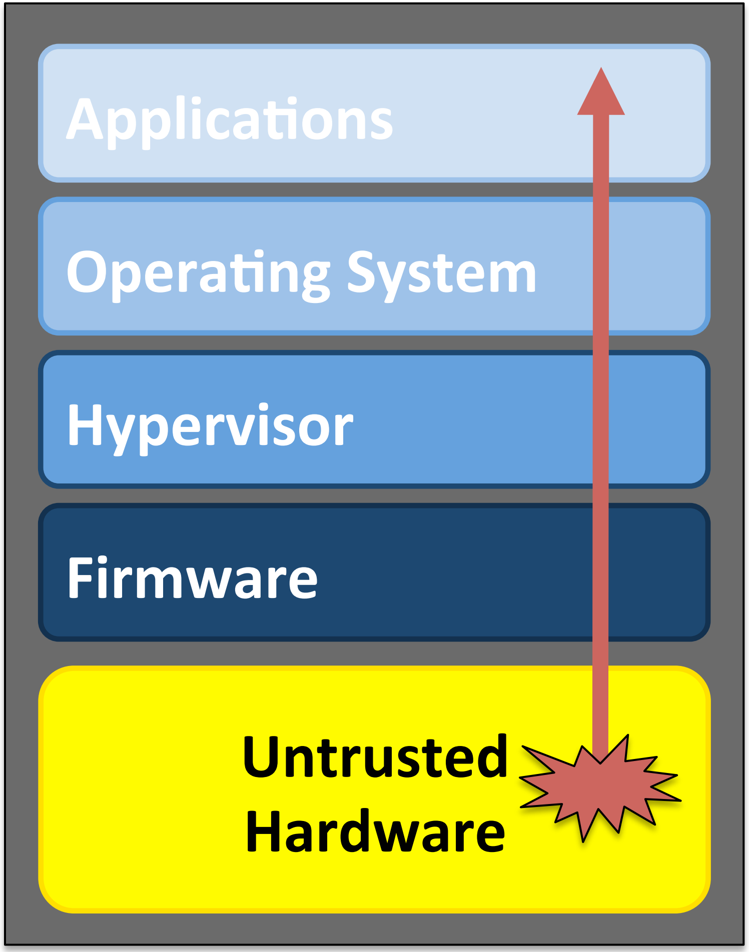
A2: Analog Malicious Hardware

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

University of Michigan

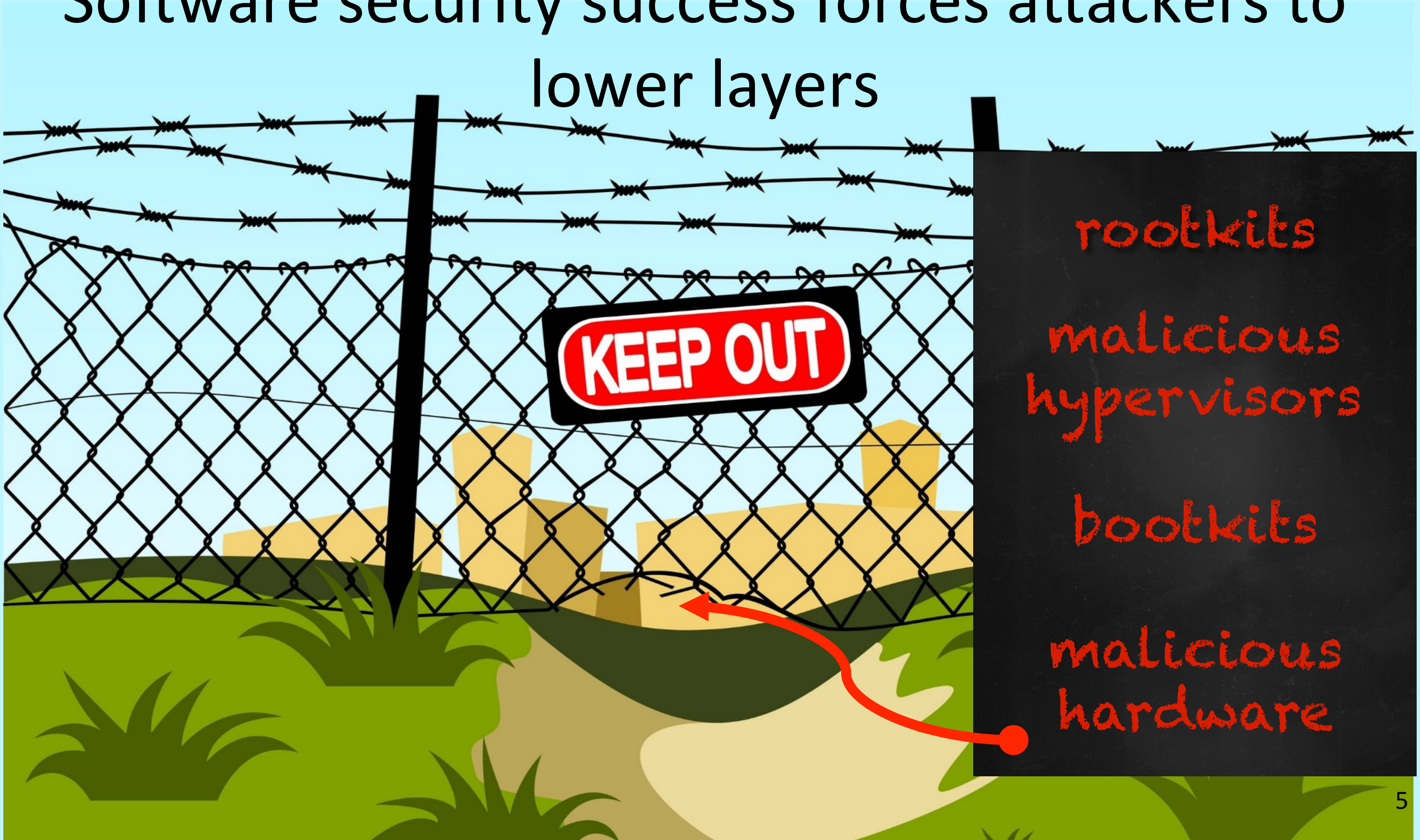
Foundations are important



Weakened hardware weakens the entire system

Software security success forces attackers to lower layers

Software security success forces attackers to lower layers



rootkits

malicious hypervisors

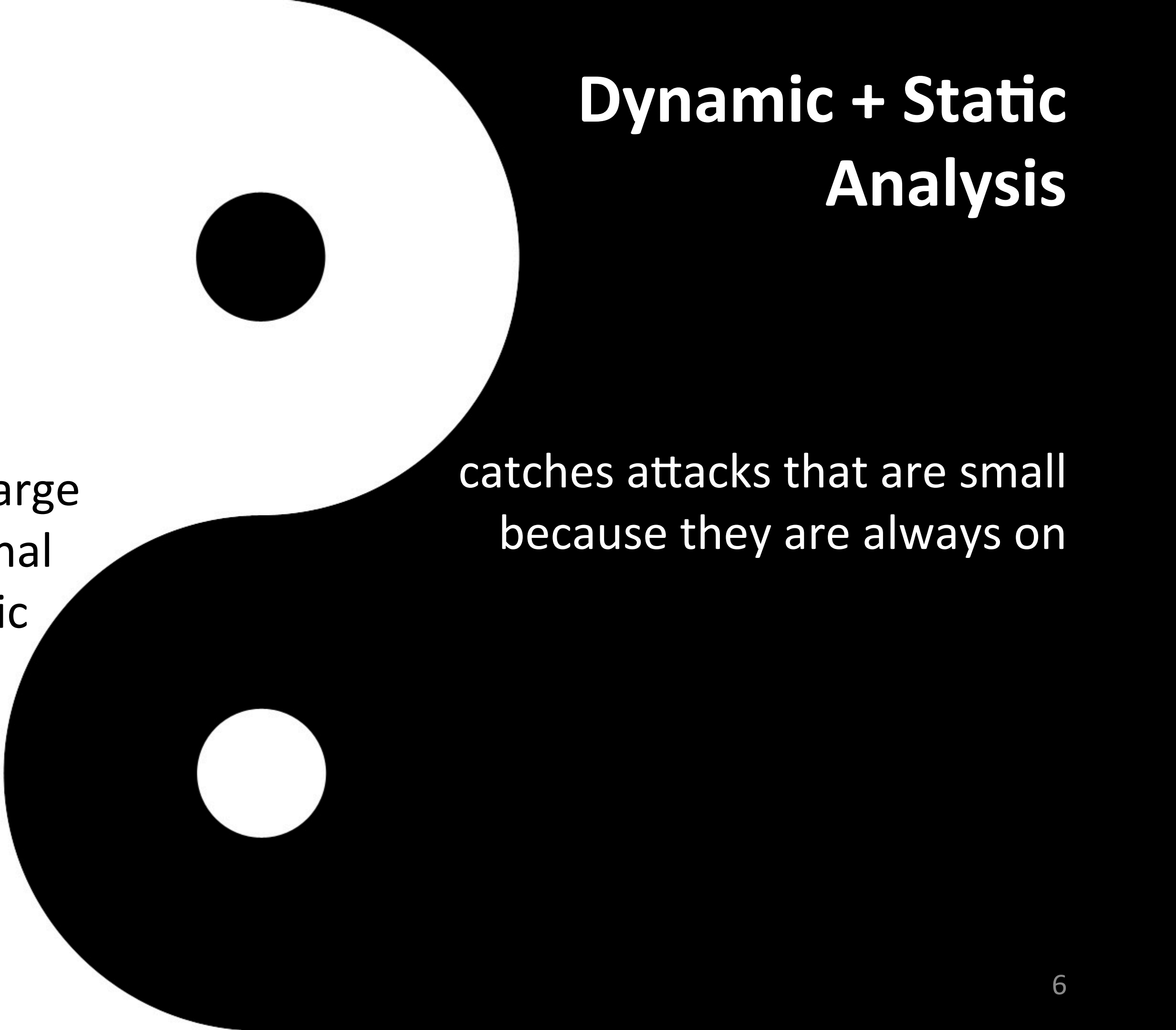
bootkits

malicious hardware

Visual Inspection

Side Channels

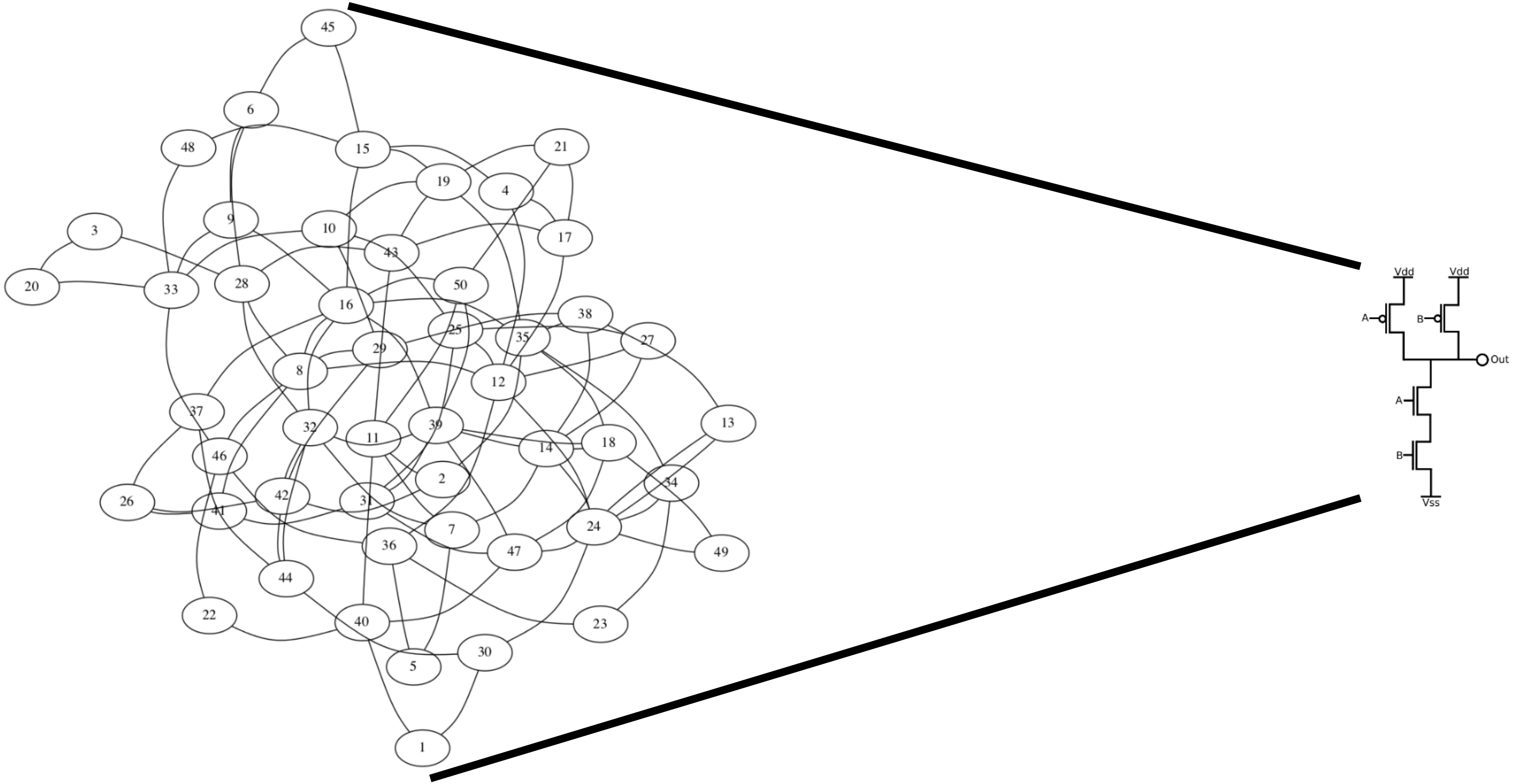
catches attacks that are large because they use additional logic to hide from dynamic analysis



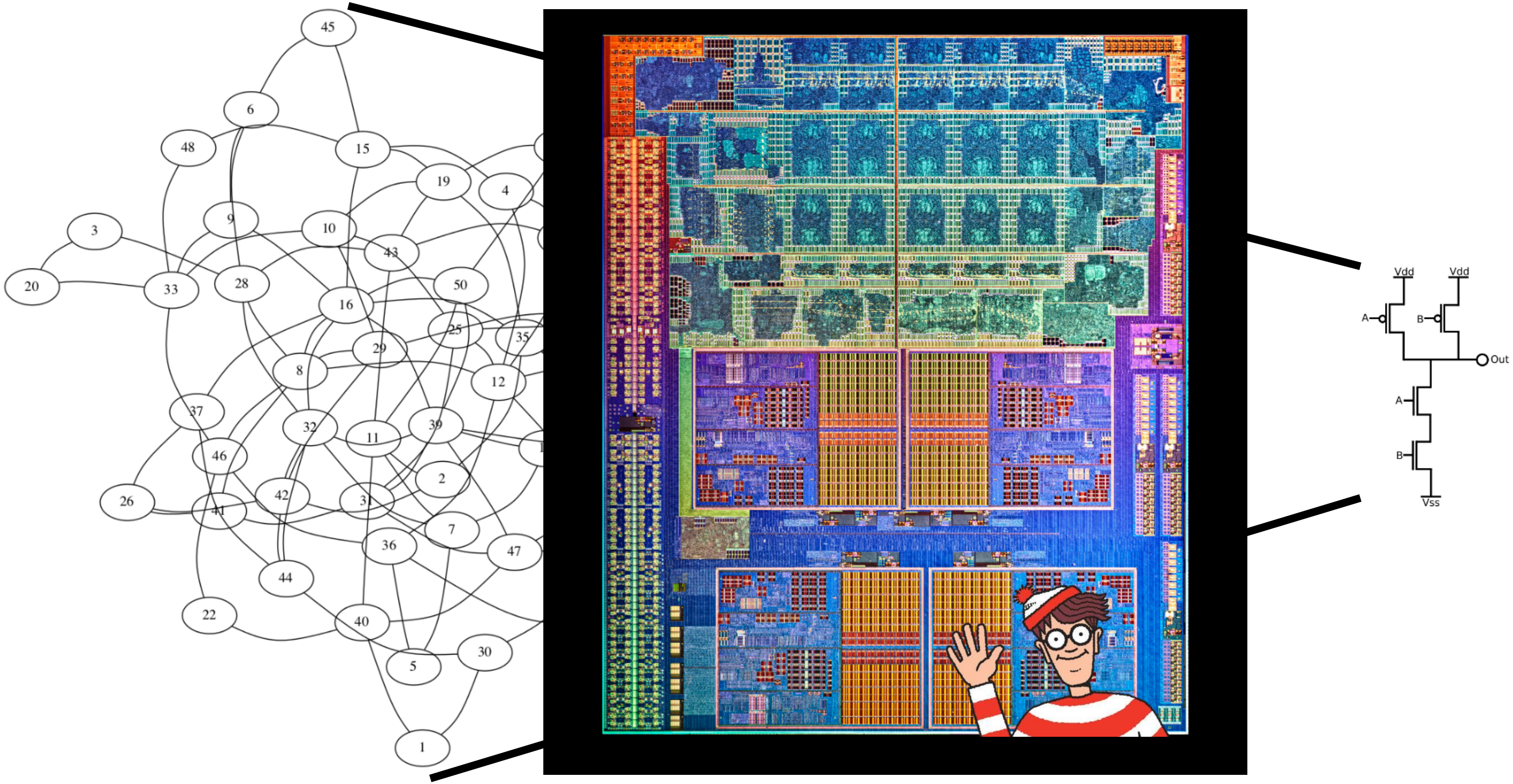
Dynamic + Static Analysis

catches attacks that are small because they are always on

Challenge: construct an attack that is stealthy and small

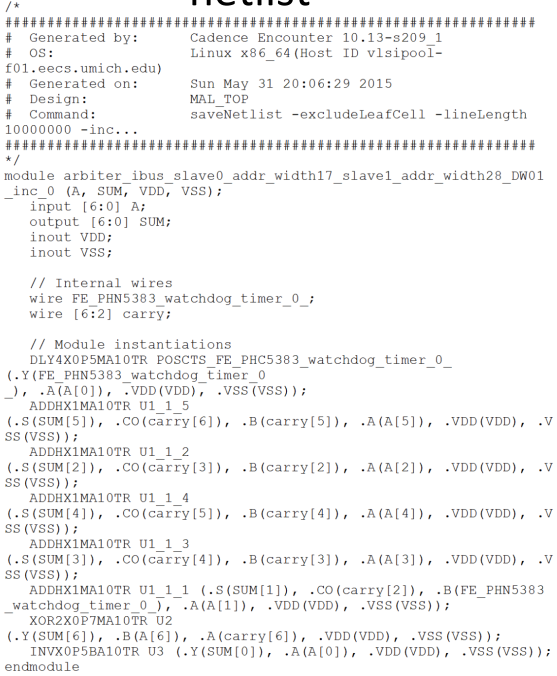


Challenge: construct an attack that is stealthy and small



Two threats, we focus on the stage that restricts the attacker the most

Back-end house netlist

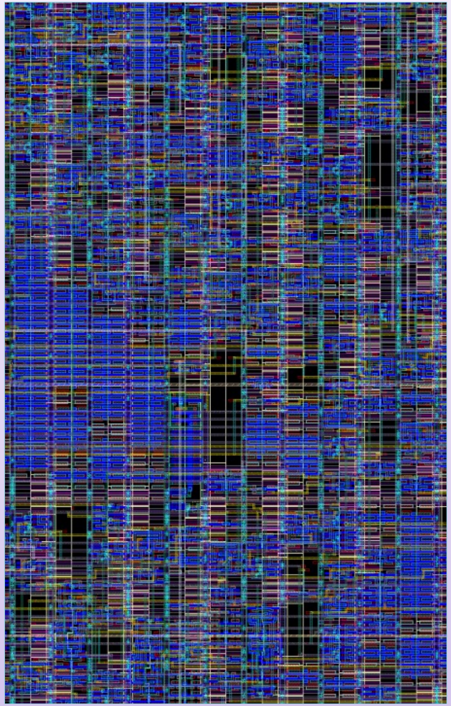


```
/*
################################################################
#  Generated by:      Cadence Encounter 10.13-s209_1
#  OS:                Linux x86_64(Host ID vlsipool-
f01.eecs.umich.edu)
#  Generated on:      Sun May 31 20:06:29 2015
#  Design:            MAL_TOP
#  Command:           saveNetlist -excludeLeafCell -lineLength
10000000 -inc...
################################################################
*/
module arbiter_ibus_slave0_addr_width17_slave1_addr_width28_DW01
_inc_0 (A, SUM, VDD, VSS);
   input [6:0] A;
   output [6:0] SUM;
   inout VDD;
   inout VSS;

   // Internal wires
   wire FE_PHN5383_watchdog_timer_0_;
   wire [6:2] carry;

   // Module instantiations
   DLY4X0P5MA10TR POSCTS_FE_PHC5383_watchdog_timer_0_
(.Y(FE_PHN5383_watchdog_timer_0
_), .A(A[0]), .VDD(VDD), .VSS(VSS));
   ADDHX1MA10TR U1_1_5
(.S(SUM[5]), .CO(carry[6]), .B(carry[5]), .A(A[5]), .VDD(VDD), .V
SS(VSS));
   ADDHX1MA10TR U1_1_2
(.S(SUM[2]), .CO(carry[3]), .B(carry[2]), .A(A[2]), .VDD(VDD), .V
SS(VSS));
   ADDHX1MA10TR U1_1_4
(.S(SUM[4]), .CO(carry[5]), .B(carry[4]), .A(A[4]), .VDD(VDD), .V
SS(VSS));
   ADDHX1MA10TR U1_1_3
(.S(SUM[3]), .CO(carry[4]), .B(carry[3]), .A(A[3]), .VDD(VDD), .V
SS(VSS));
   ADDHX1MA10TR U1_1_1 (.S(SUM[1]), .CO(carry[2]), .B(FE_PHN5383
_watchdog_timer_0_), .A(A[1]), .VDD(VDD), .VSS(VSS));
   XOR2X0P7MA10TR U2
(.Y(SUM[6]), .B(A[6]), .A(carry[6]), .VDD(VDD), .VSS(VSS));
   INVX0P5BA10TR U3 (.Y(SUM[0]), .A(A[0]), .VDD(VDD), .VSS(VSS));
endmodule
```

Foundry GDSII



We leverage analog behavior to construct an attack that is stealthy and small

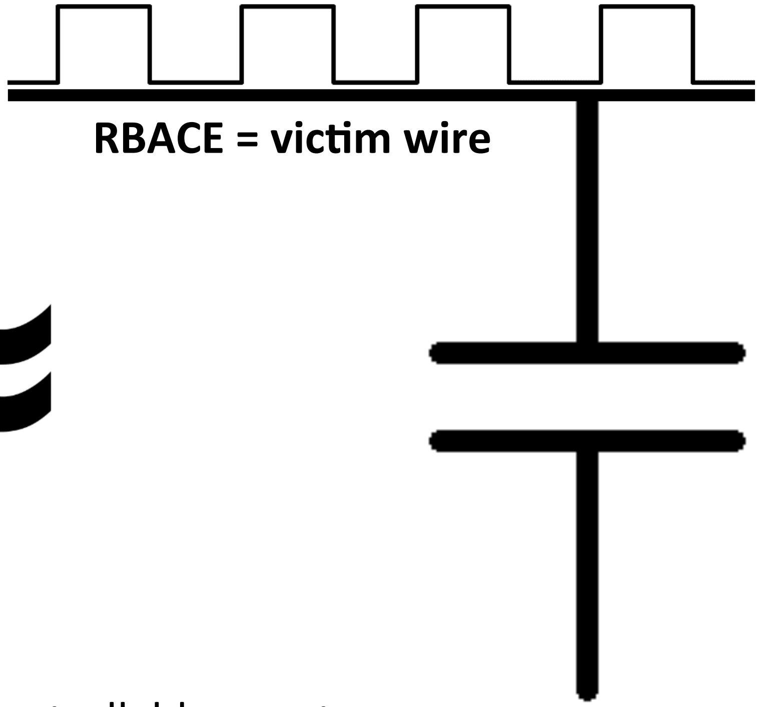
```
on_every(RBACE) do
    if(count == 12345) then
        do_attack()
    else
        count = count + 1
done
```

RBACE = rare, but attacker controllable event

We leverage analog behavior to construct an attack that is stealthy and small

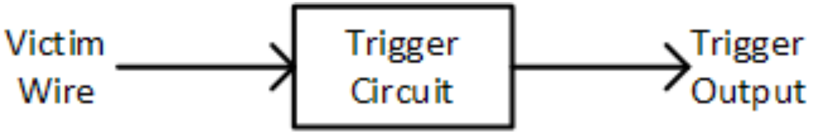
```
on_every(RBACE) do
    if(count == 12345) then
        do_attack()
    else
        count = count + 1
done
```

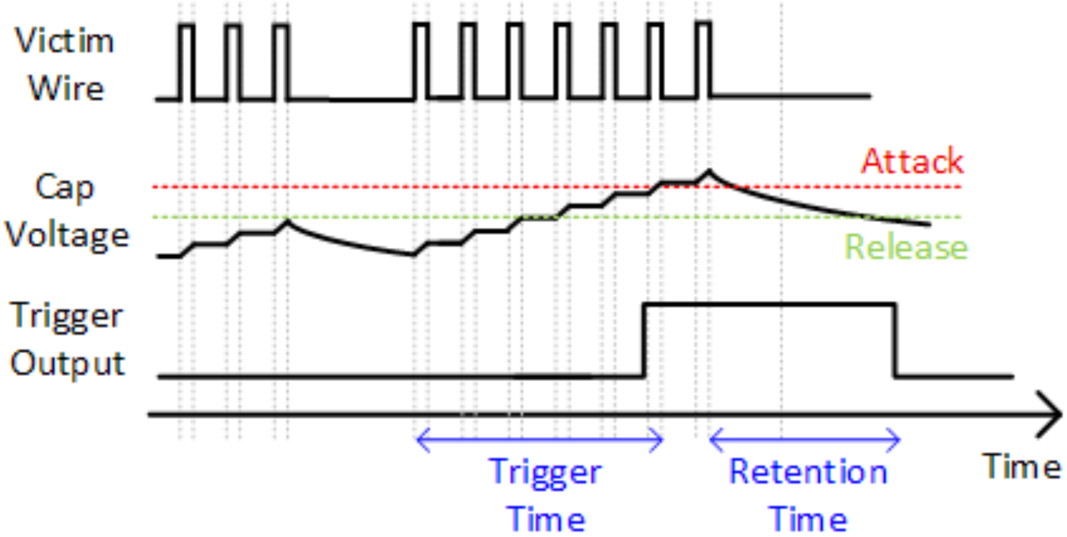
≈



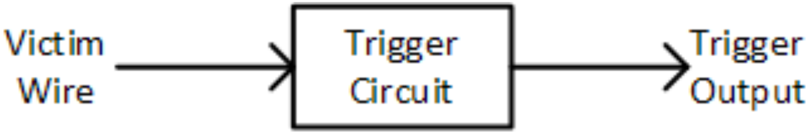
RBACE = rare, but attacker controllable event

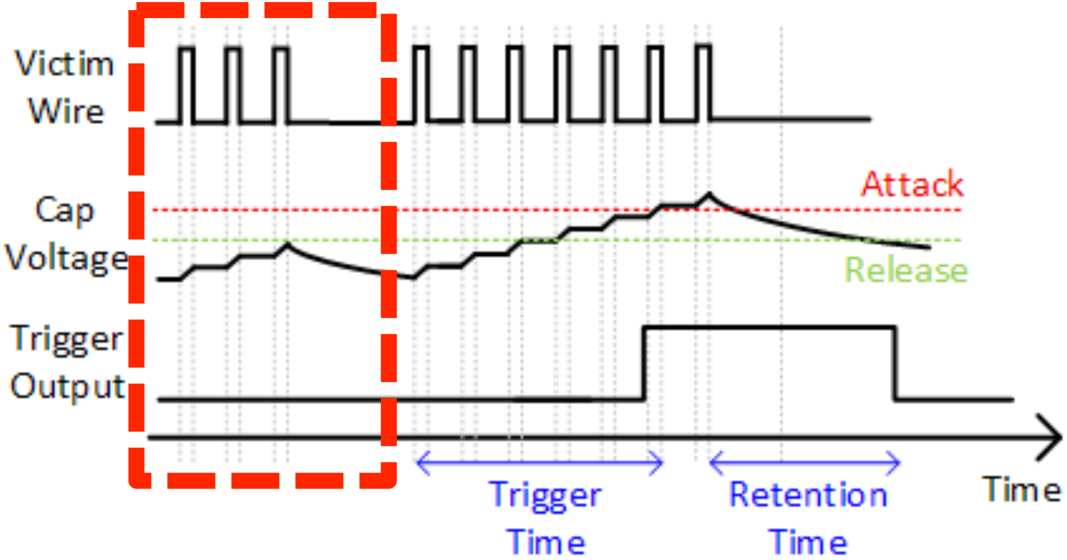
An ideal analog trigger



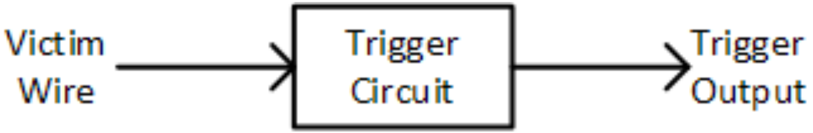


An ideal analog trigger



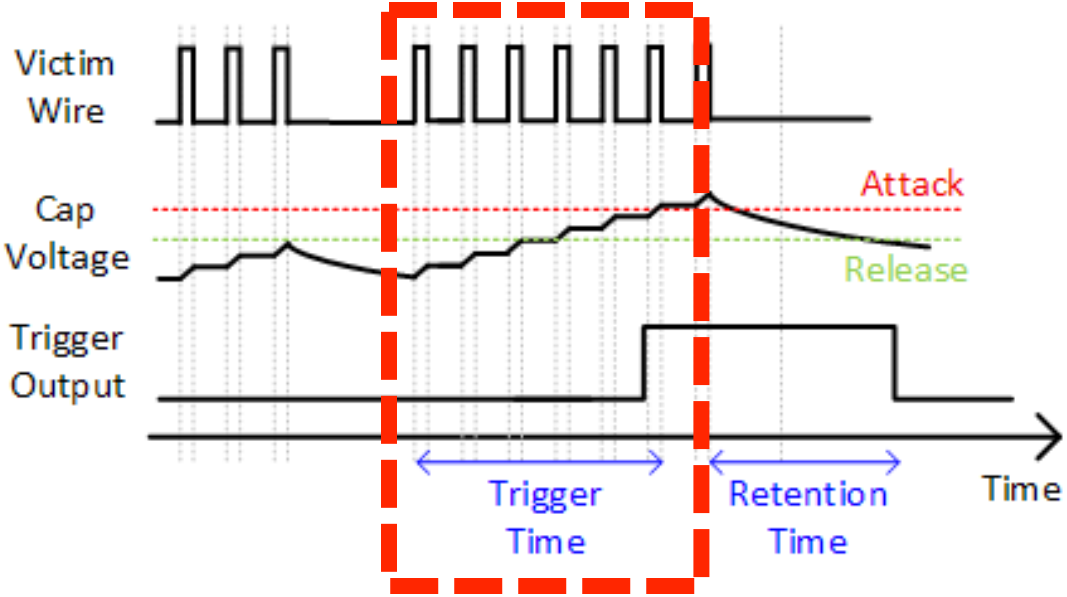


An ideal analog trigger

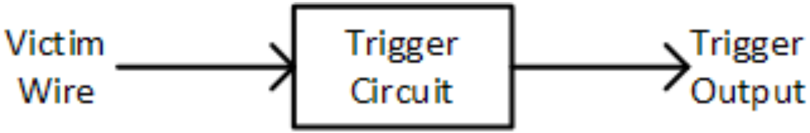

Victim Wire

Trigger Circuit

Trigger Output



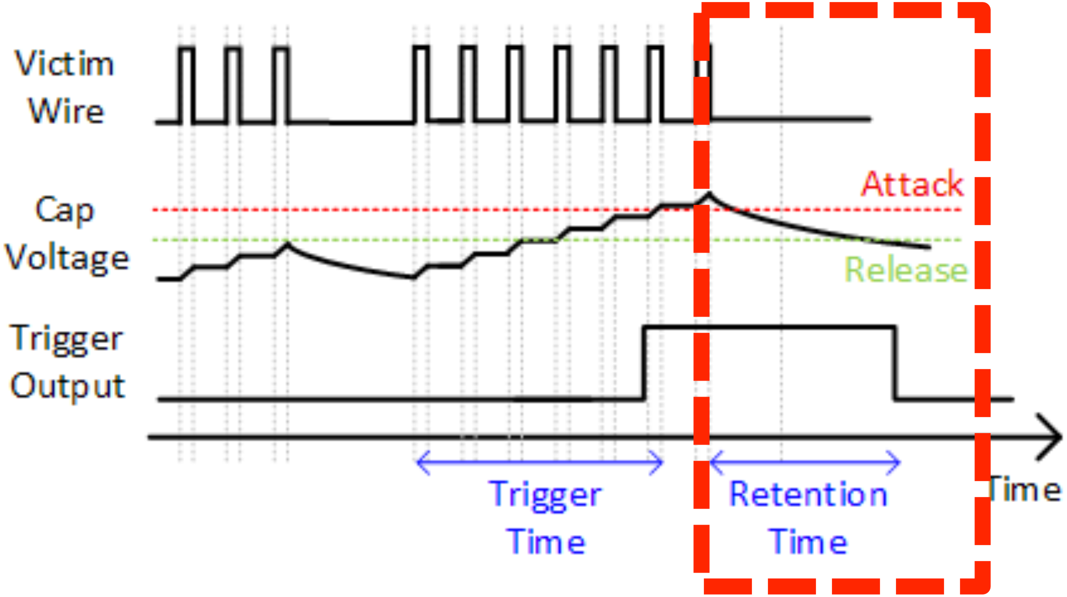
An ideal analog trigger



Victim Wire

Trigger Circuit

Trigger Output



Victim Wire

Cap Voltage

Attack

Release

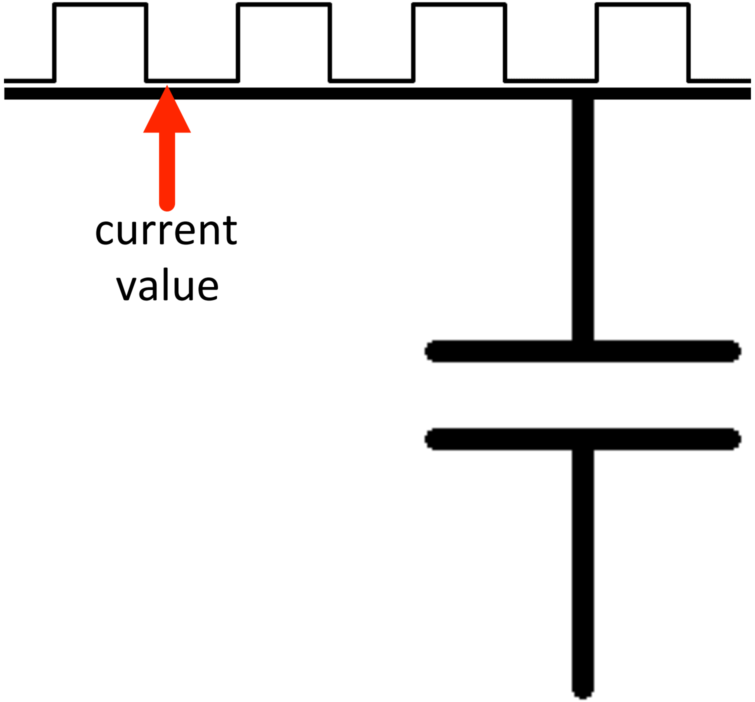
Trigger Output

Time

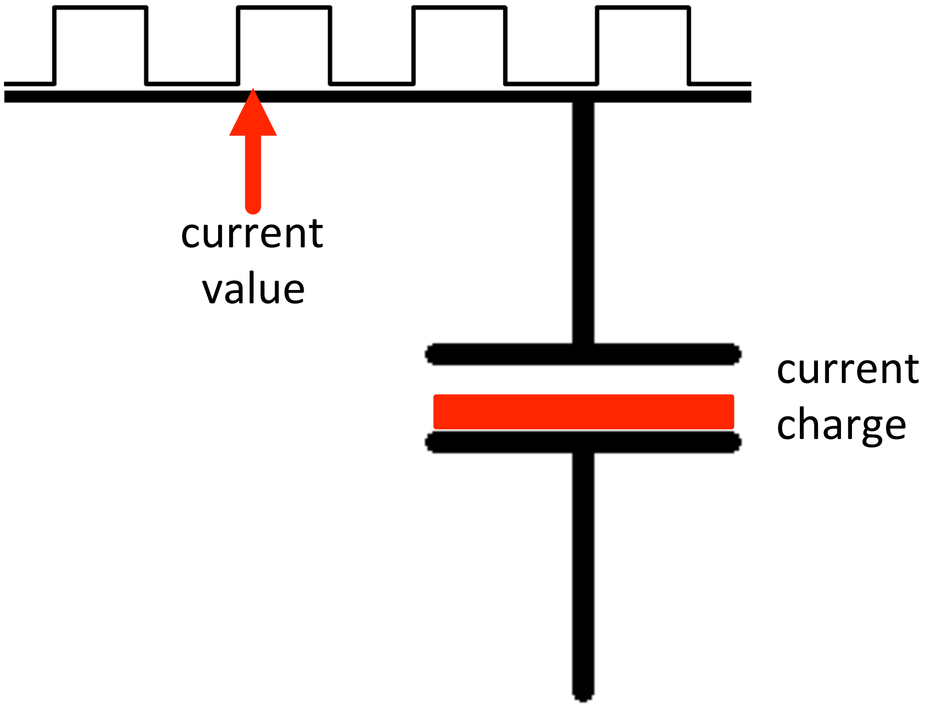
Trigger Time

Retention Time

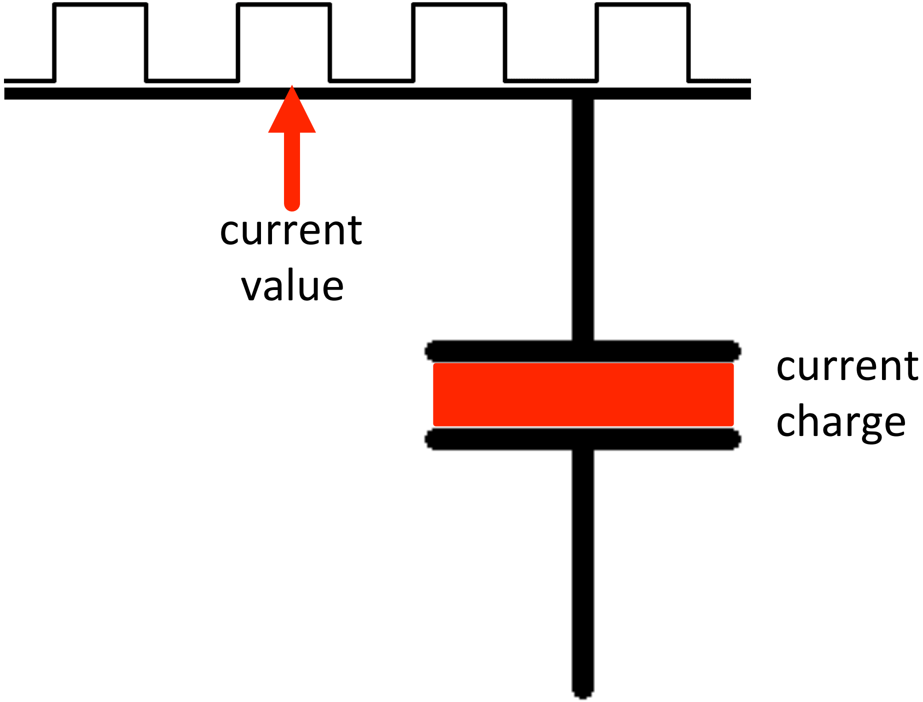
Challenge: small capacitors charge quickly, large capacitors induce current spikes



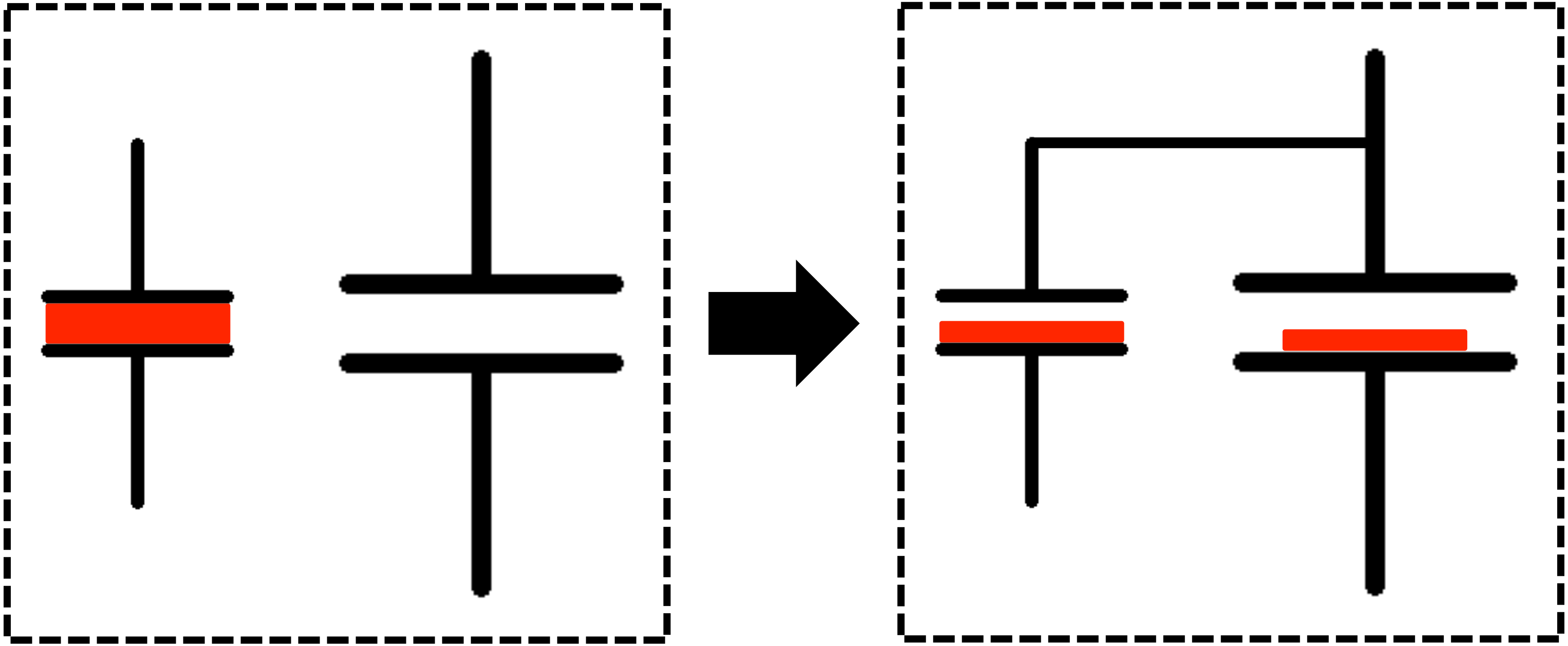
Challenge: small capacitors charge quickly, large capacitors induce current spikes



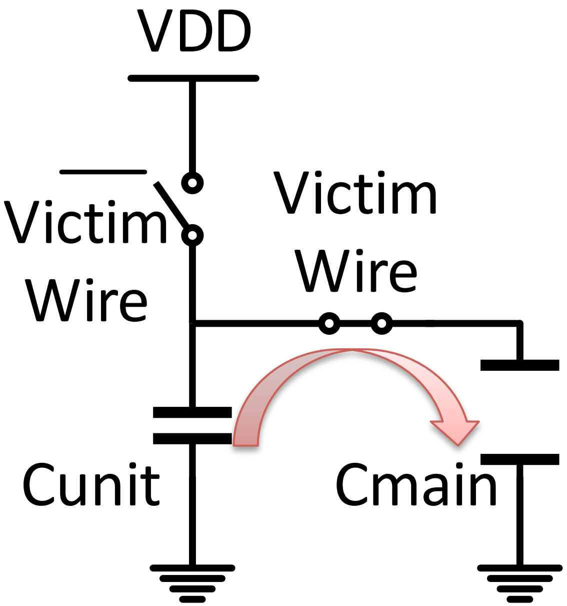
Challenge: small capacitors charge quickly, large capacitors induce current spikes

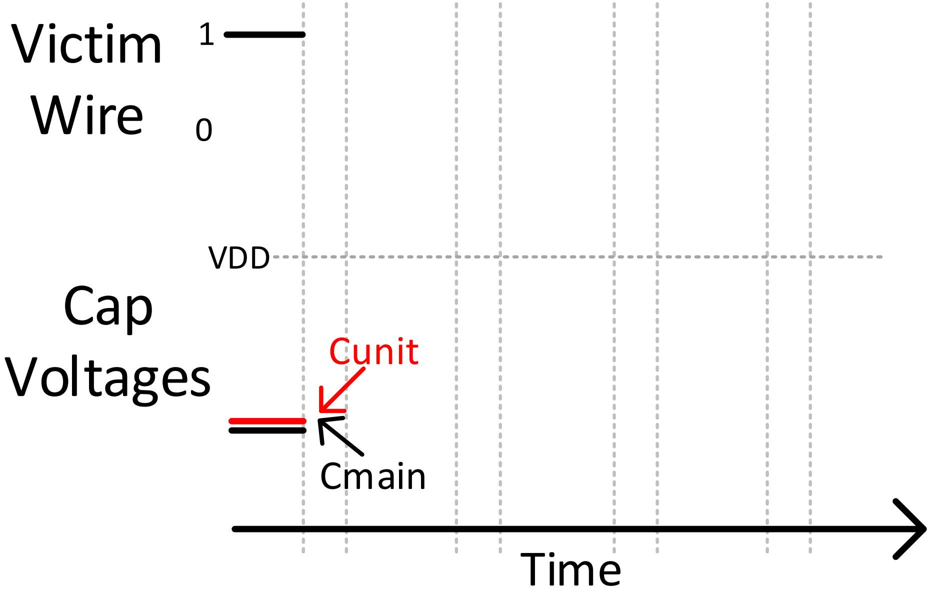


Solution: charge sharing

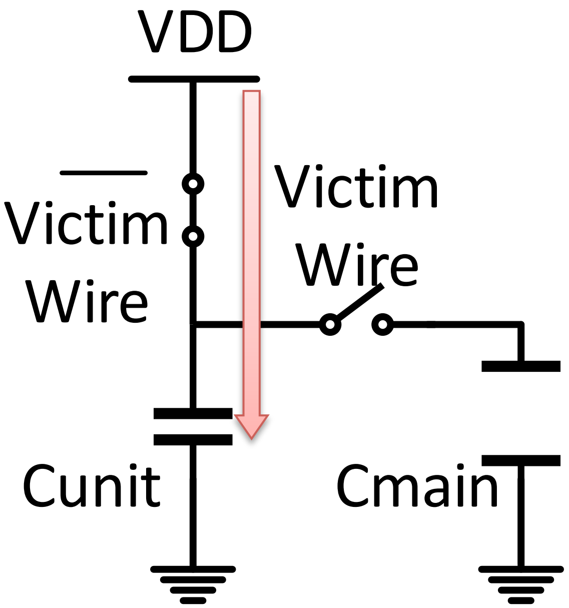


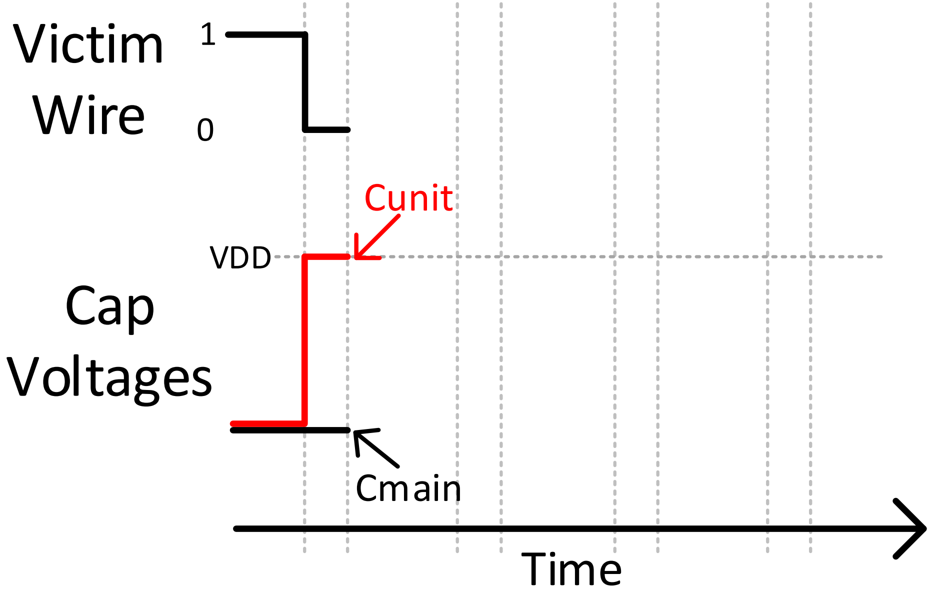
Creating an analog trigger using gated charge sharing



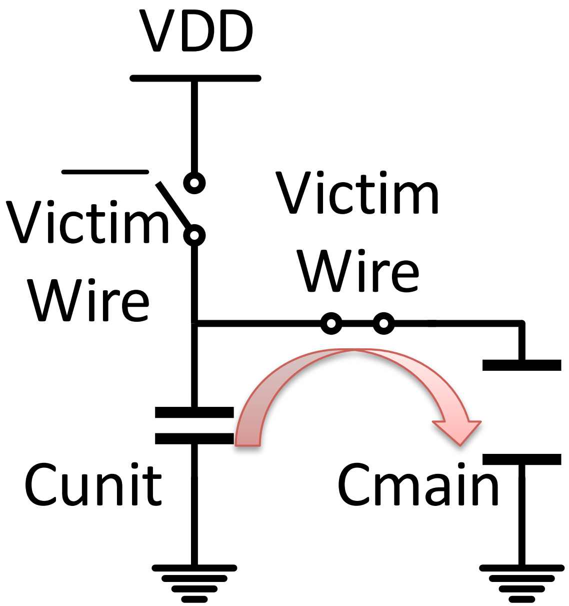


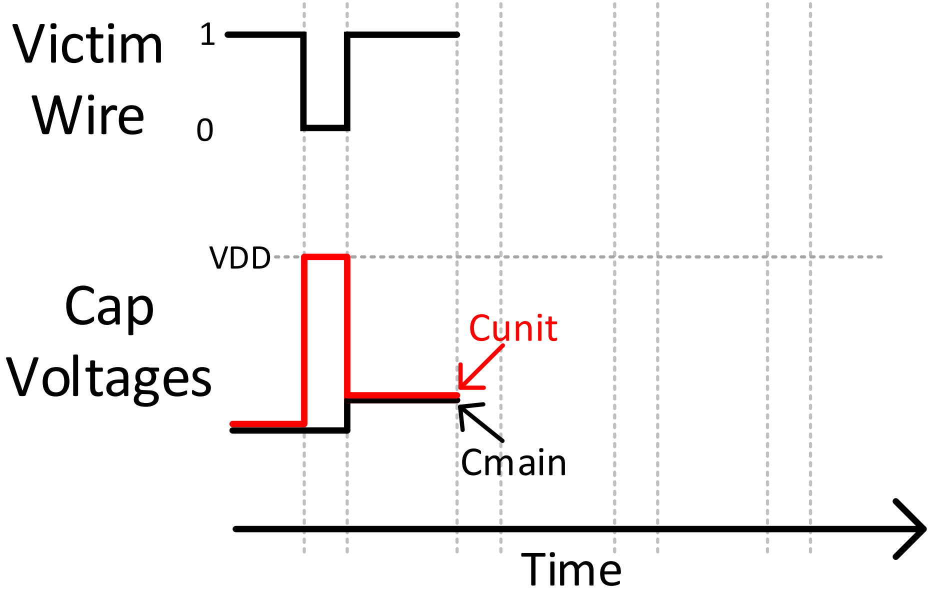
Creating an analog trigger using gated charge sharing



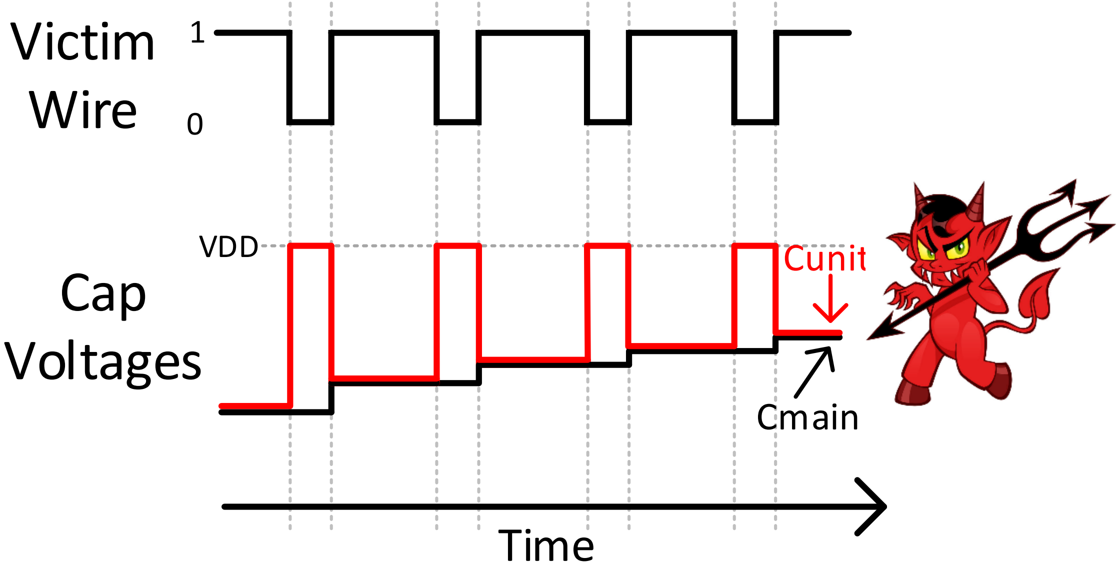


Creating an analog trigger using gated charge sharing



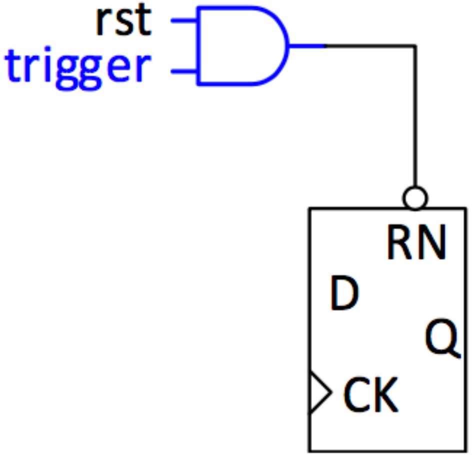


Creating an analog trigger using gated charge sharing

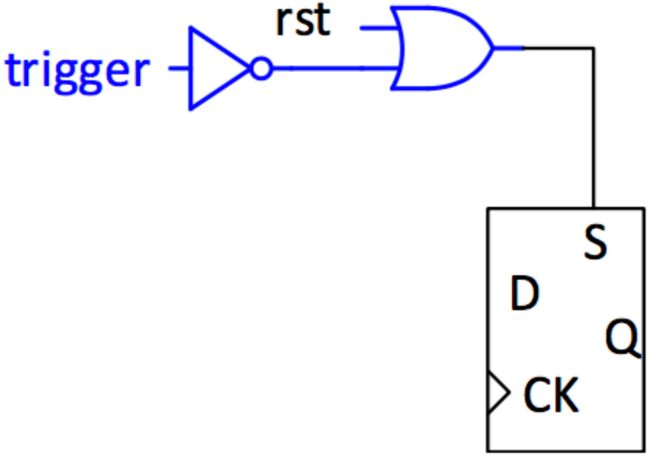


Creating a privilege escalation attack

*Our analog trigger is attack agnostic

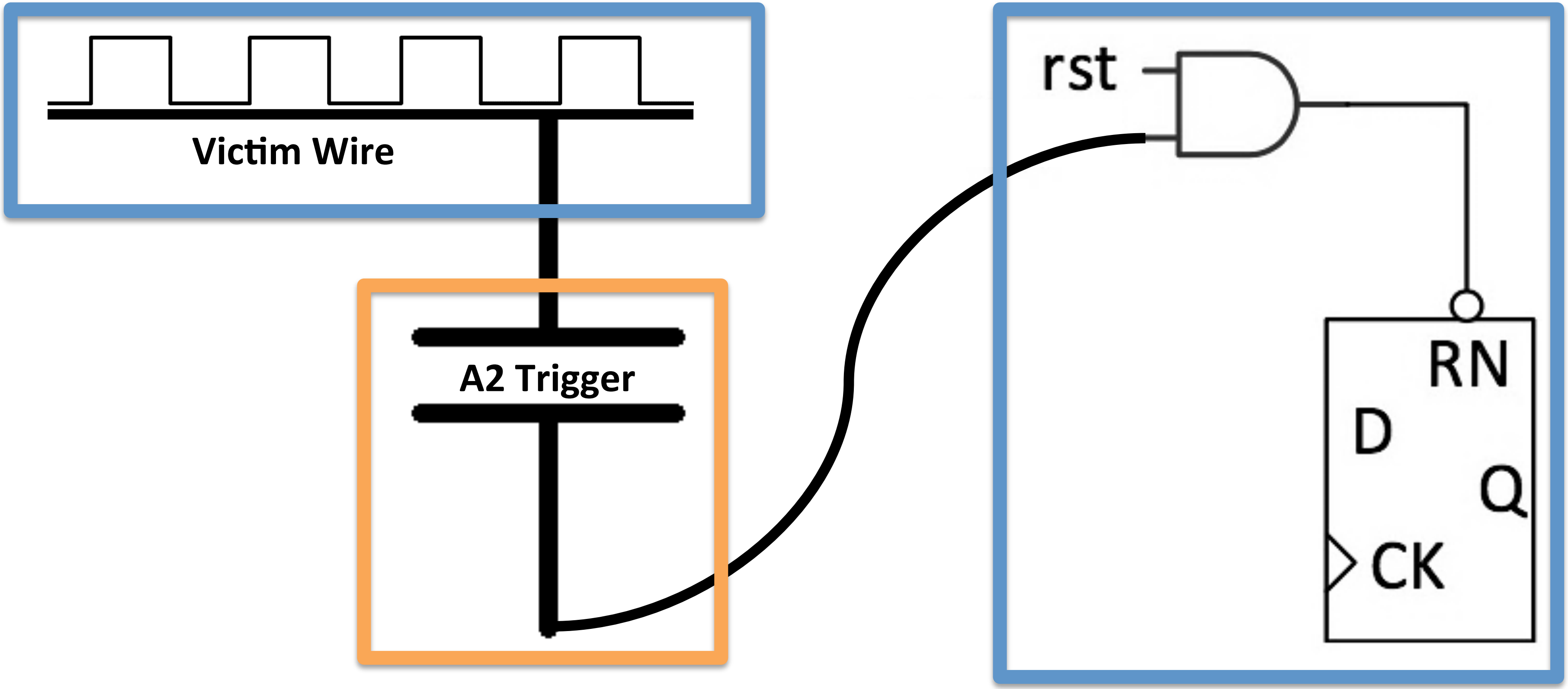


Inverted reset

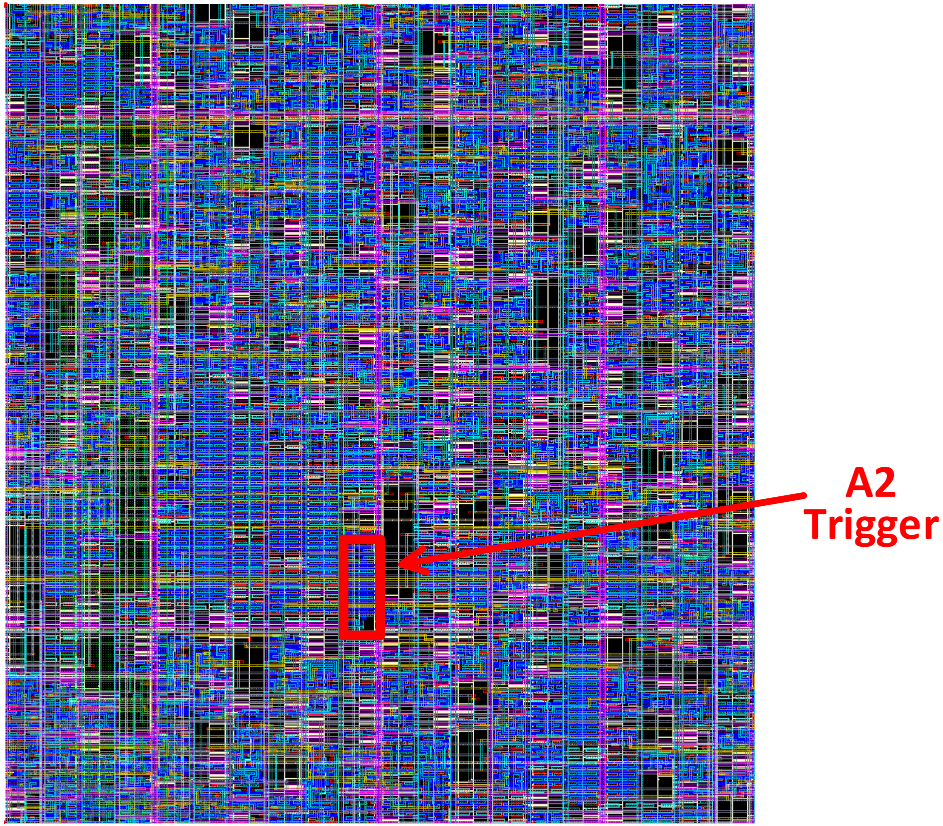


Positive reset

A2



Implanting A2 into an existing chip layout

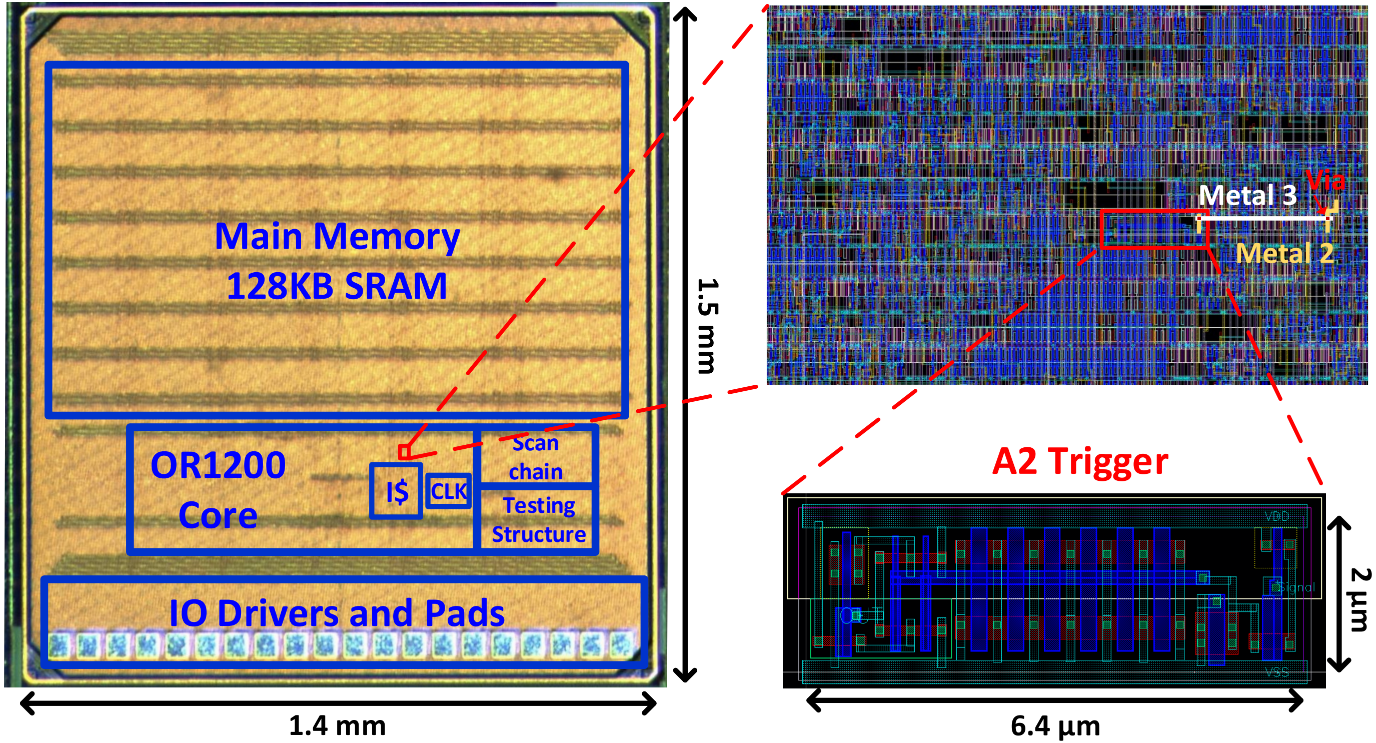


20% to 30%
of chip area
is **unused**

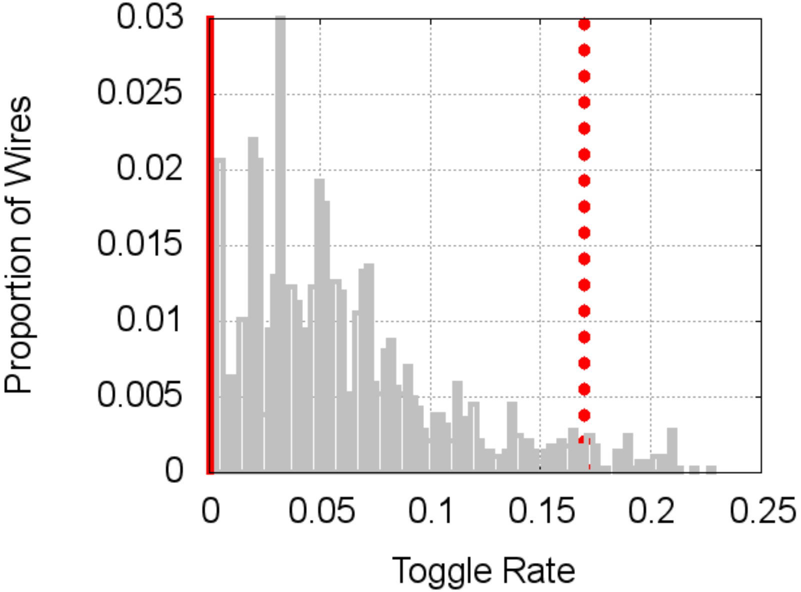
Other challenges in the paper

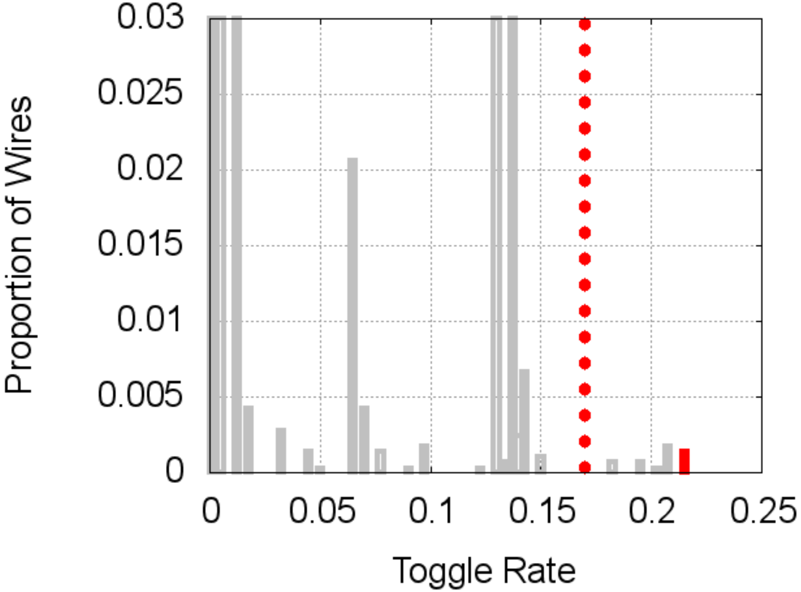
- Analog circuit design process
- Finding a suitable victim wire
- Finding the flip-flop to attack
- Building multi-stage attacks
- Writing trigger activation code
- Covertly testing for attack success

We had to build A2 to know it worked

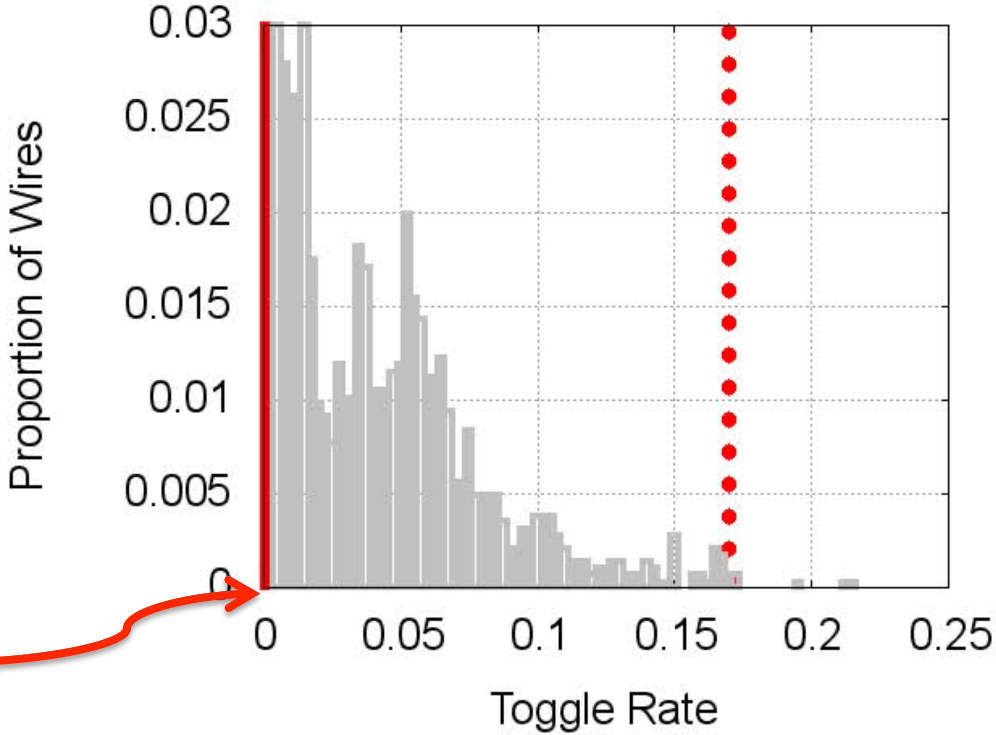


We activate A2 in real hardware using only user mode code



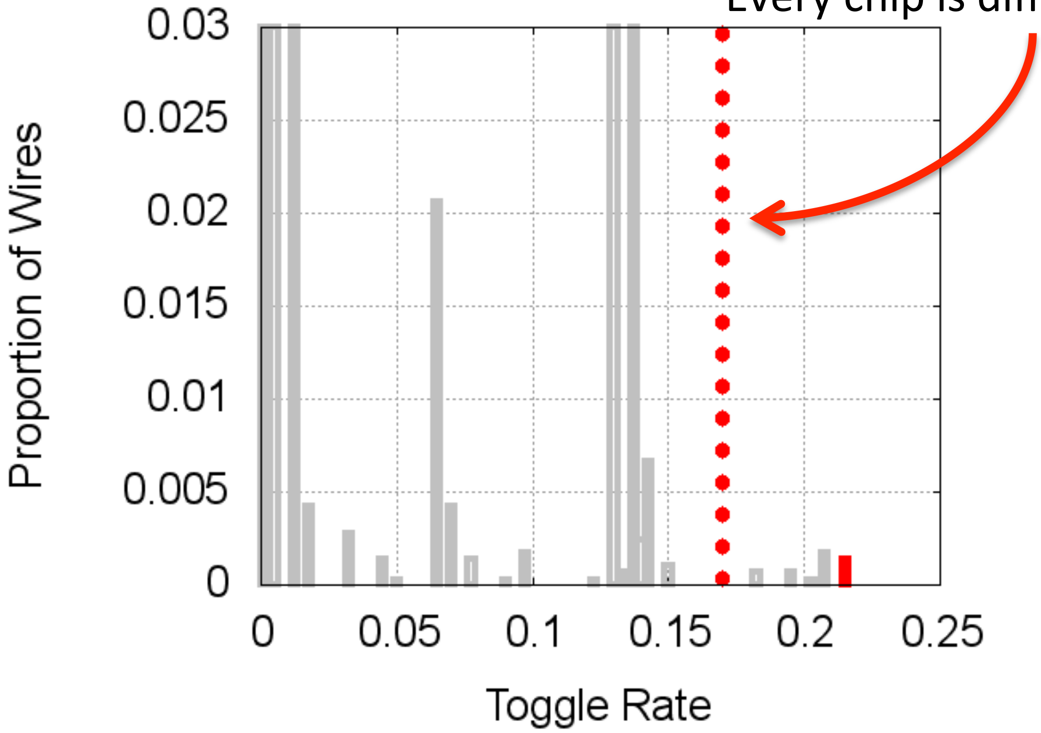


A2 is hidden from post-fab testing



.0002 for division-heavy benchmark

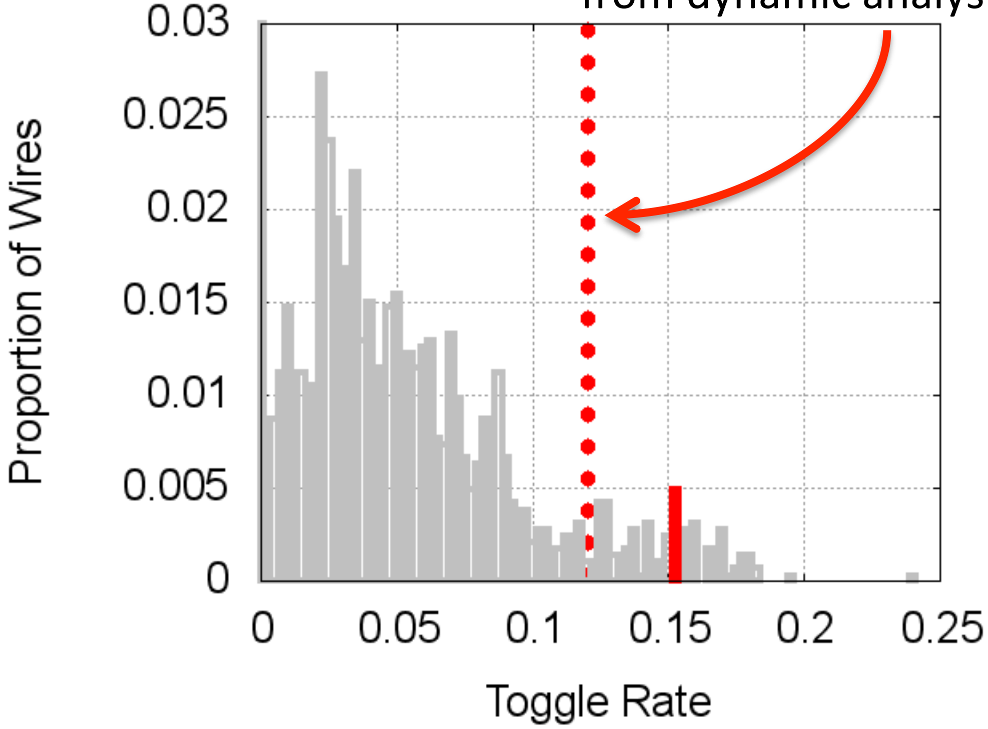
Attackers can reliably model their attacks



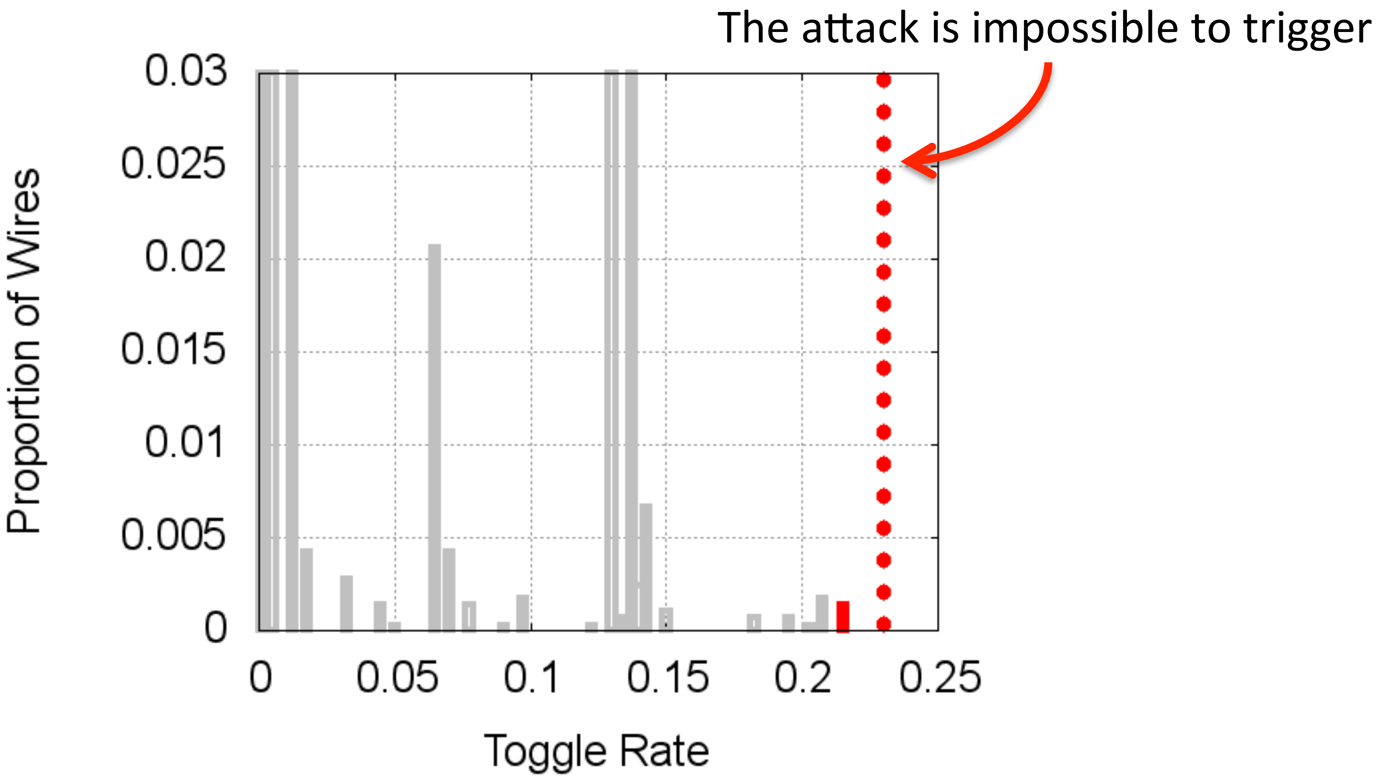
Where is this in real hardware?
Every chip is different!

Attackers can reliably model their attacks

The attack is not well hidden from dynamic analysis (testing)

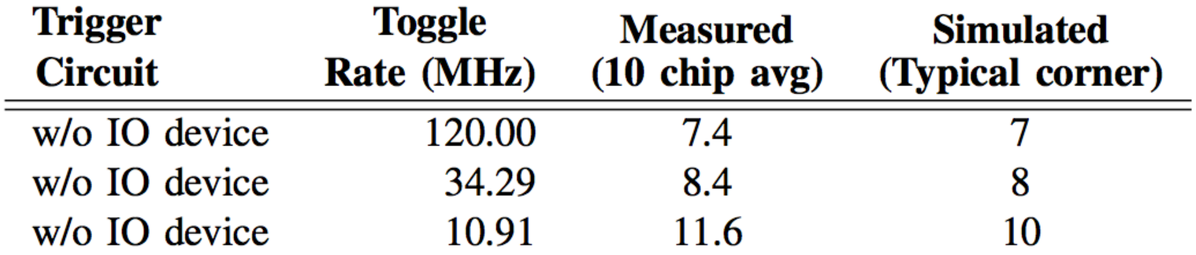


Attackers can reliably model their attacks



The attack is impossible to trigger

Attackers can reliably model their attacks



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

More experiments in the paper

- Comparison of different standard cell sizes and out attack
- Distribution of trigger times
- Distribution of retention times
- Effect of voltage on cycles to trigger
- Effect of temperature on cycles to trigger
- Effect of temperature on retention time
- Power of benchmarks and attack programs

Cross-domain attacks are stealthy and controllable

- A2 spans the analog and digital domains
- A2 is controllable
- A2 is stealthy
  - complex and unlikely trigger sequence
  - a single cell
- Currently, only detectable post-fabrication

We need to try something different:

detection

plus

protection

Research artifacts: github.com/impedimentToProgress/A2

Me: ImpedimentToProgress.com

Fabricator	Popular offshore corp.
Interface	GDSII
Turnaround time	3 months
Added time to project	1 year
Area	1.5mm x 1.5mm
Core	330um x 550um
Memory	1145um x 765um
Process	65nm
Number of chips	100
Cost	\$5k to \$10k per 1mm^2
Other costs	packaging