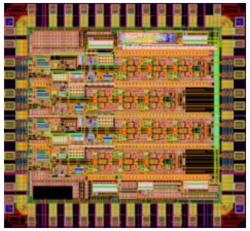
Stealthy Dopant-Level Hardware Trojans

Georg T. Becker[1], Francesco Regazzoni[2], Christof Paar[1,3],
and Wayne P. Burleson[1]
[1]University of Massachusetts Amherst, USA
[2] TU Delft, The Netherlands and ALaRI – University of Lugano, Switzerland
[3] Horst Görtz Instistut for IT Security, Ruhr Universität Bochum, Germany

Agenda

- **Introduction to Hardware Trojans**
- Dopant-Level Hardware Trojans
- Case study 1: RNG design
- Case study 2: Side-channel resistant Sbox
- Conclusion & future work

Hardware Trojans

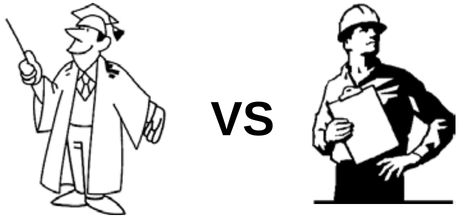
 + 

Malicious change or addition to a IC that adds or remove functionality or reduces reliability

– Can be inserted at many stages:
  - **Design stage:** 3rd party IP-cores, malicious employee, hackers etc.
  - **Manufacturing stage**: Malicious factory (often off-shore → untrusted government)
  - **Assembly and shipping**: Replace IC with a copy

Trojan designs

- No “real” Hardware Trojan found yet
- All examples from academia



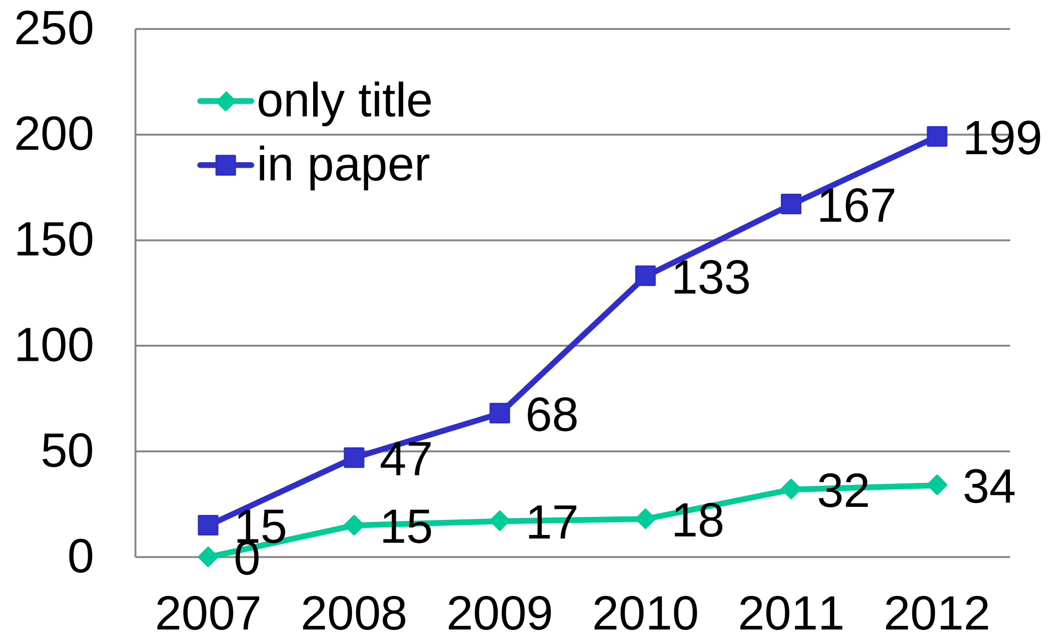
- Most Trojans at the HDL level
- Often FPGAs are used for prototypes

- Yearly NYU-Poly “Embedded Systems Challenge”

Hardware Trojans - What is the trend?

[1] Report of the defense science board task force on high performance microchip supply. Defense Science Board, US DoD, February 2005.

Published papers with „hardware Trojans“ or „malicious Hardware“
(using Google Scholar, Aug 2013)

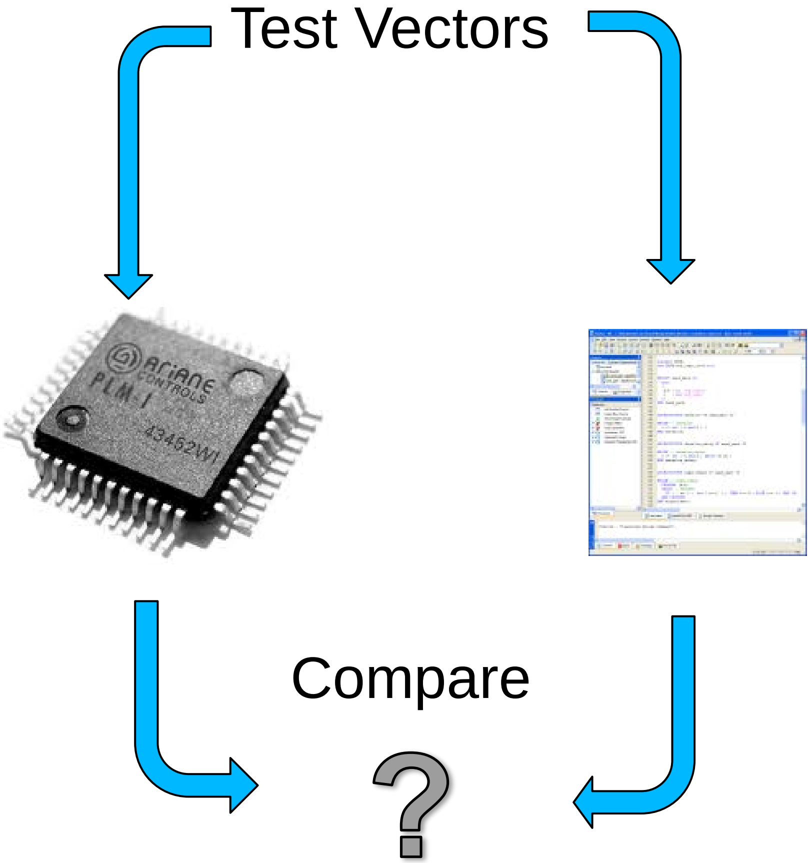


Proposed Hardware Trojan Detection Methods

- Formal verification
- Functional testing
- Optical inspection
- Side-channels
- Trojan detection circuitry

Functional testing

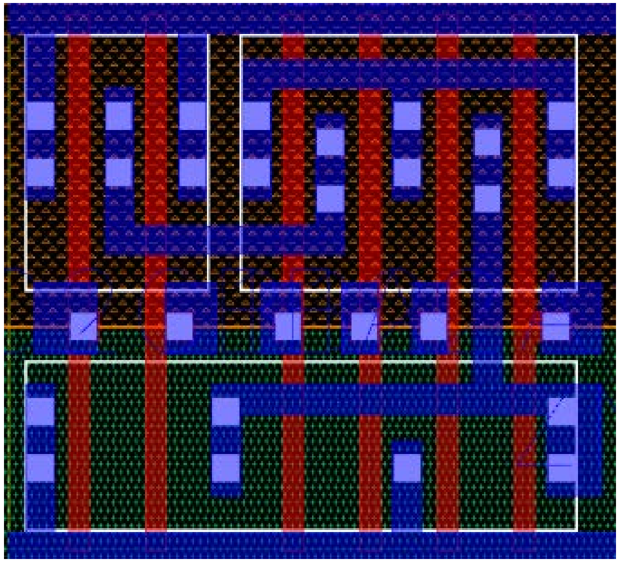
- Standard procedure
- Usually done to detect manufacturing defects
- Sometimes build-in circuitry is used (BIST)



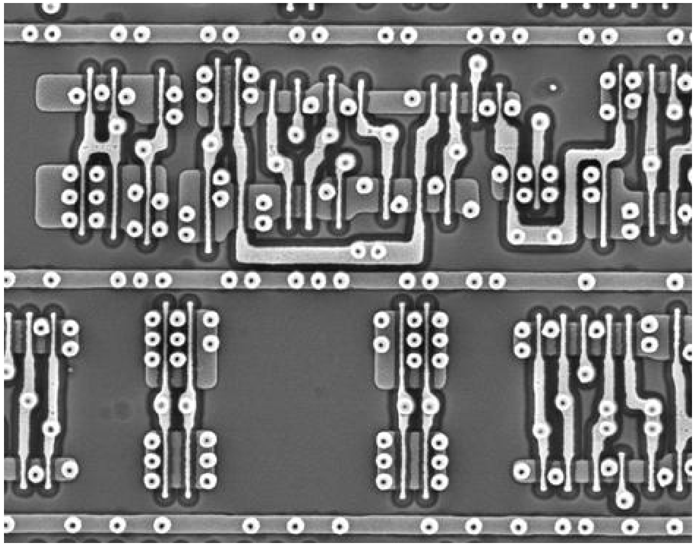
Optical Reverse-Engineering

Compare layout-mask with die-photos (e.g. SEM)

- Expensive and time consuming for large ICs
- Typically only metal, polysilicon and active area can be detected reliably!
- Destructive technique



VS



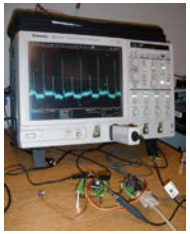
Used to detect Trojans inserted during manufacturing stage

Side-channel comparison

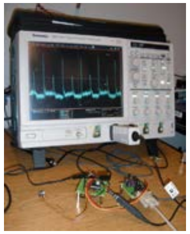
Reference chip
("golden model")

Suspected Chip

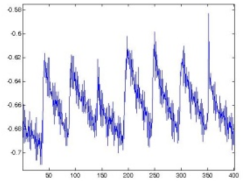
How to get it?



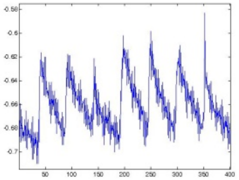
Measure side-channel



Compare



?

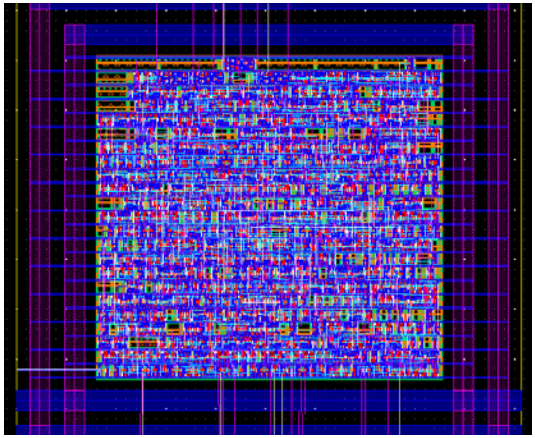


Georg T. Becker

Agenda

- Introduction to Hardware Trojans
- **Dopant-Level Hardware Trojans**
- Case study 1: TRNG design
- Case study 2: Side-channel resistant Sbox
- Conclusion & future work

Dopant-level Hardware Trojans



Main idea: Change the design below the transistor level.

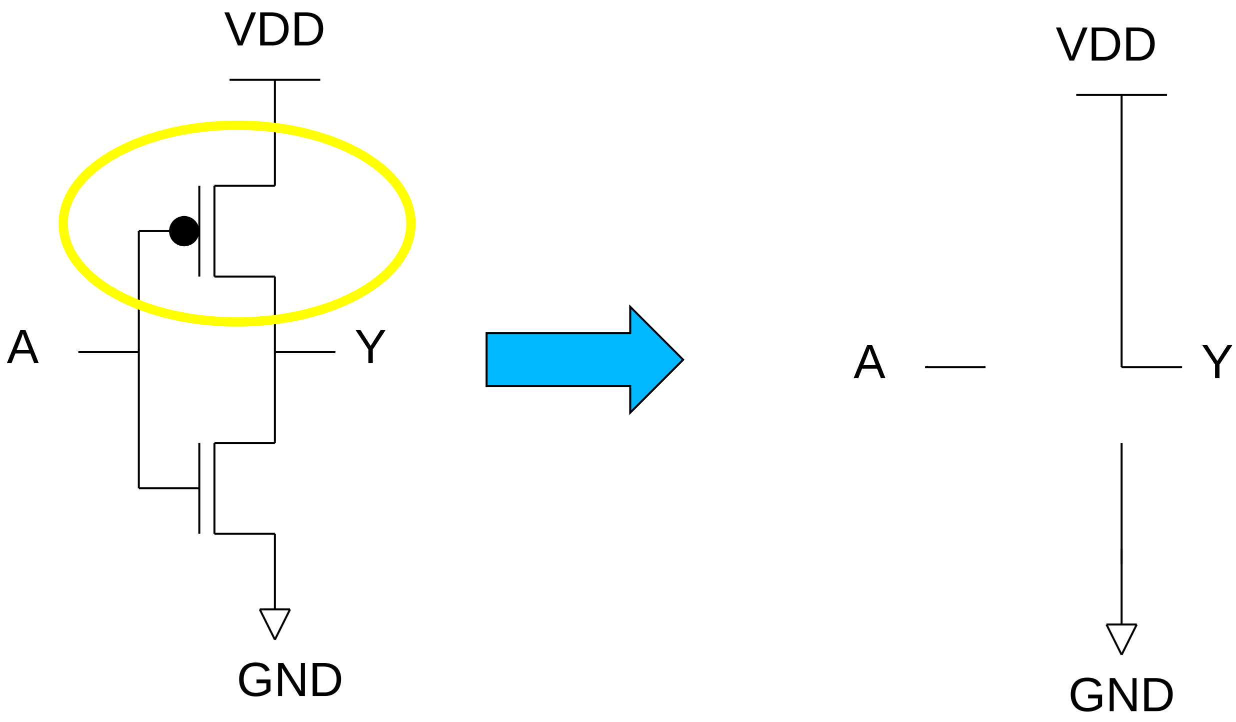
Why Layout?

- Malicious factories one of the major concerns (factories often located in different country)
- Hardly any layout-level Trojans in the literature
- We can make the Trojans extremely stealthy with zero overhead

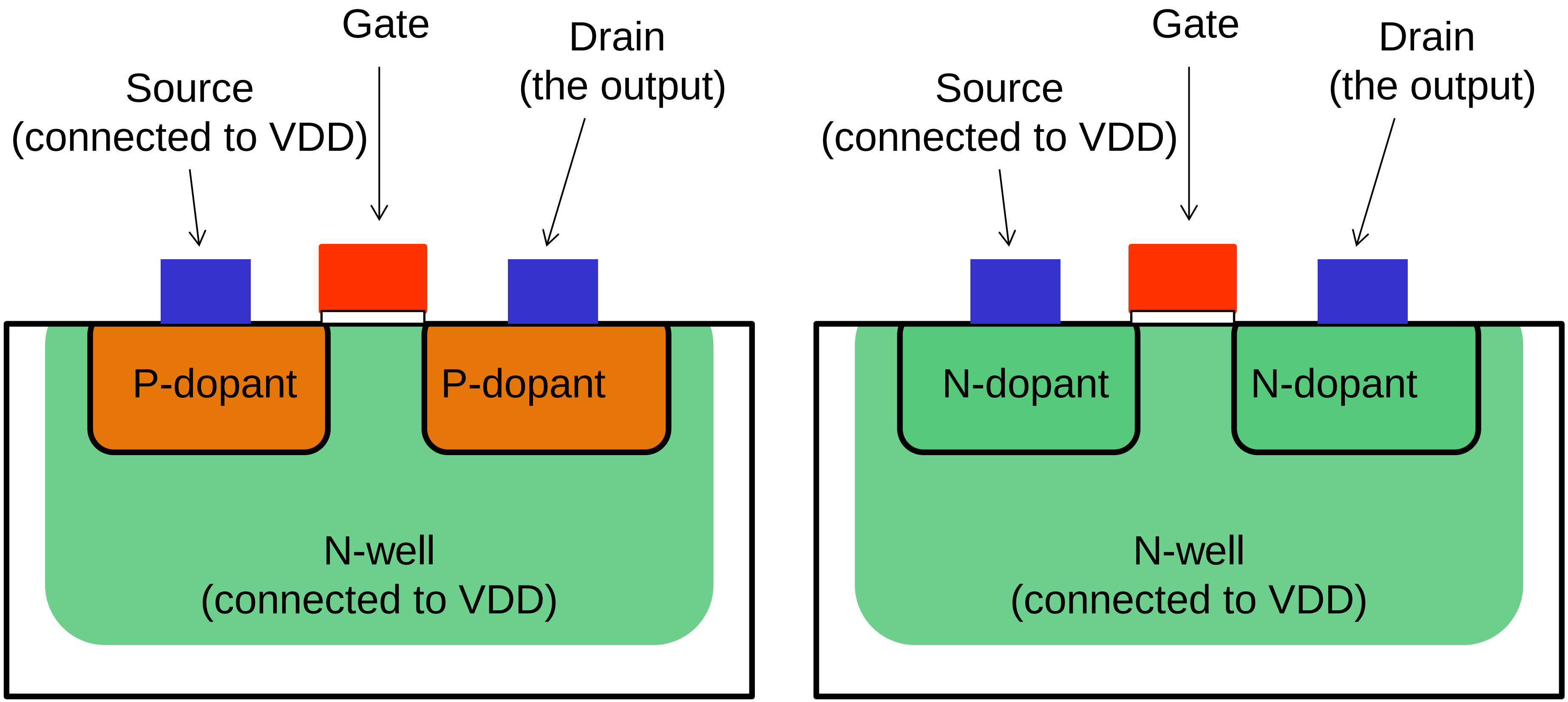
$\Rightarrow$ Defeat optical reverse-engineering?

Simple Example: Inverter Trojan

Goal: Modify an Inverter so that it always outputs VDD **without visible changes**.



PMOS Transistor Trojan

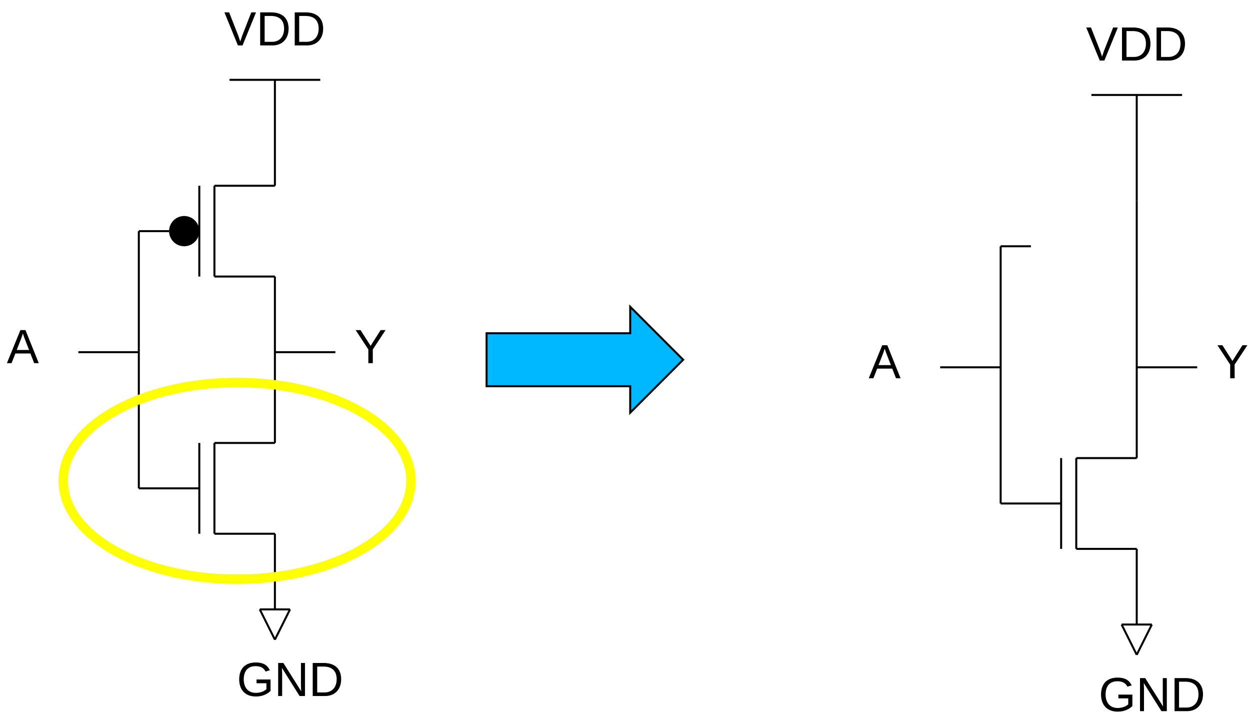


Unmodified PMOS Transistor

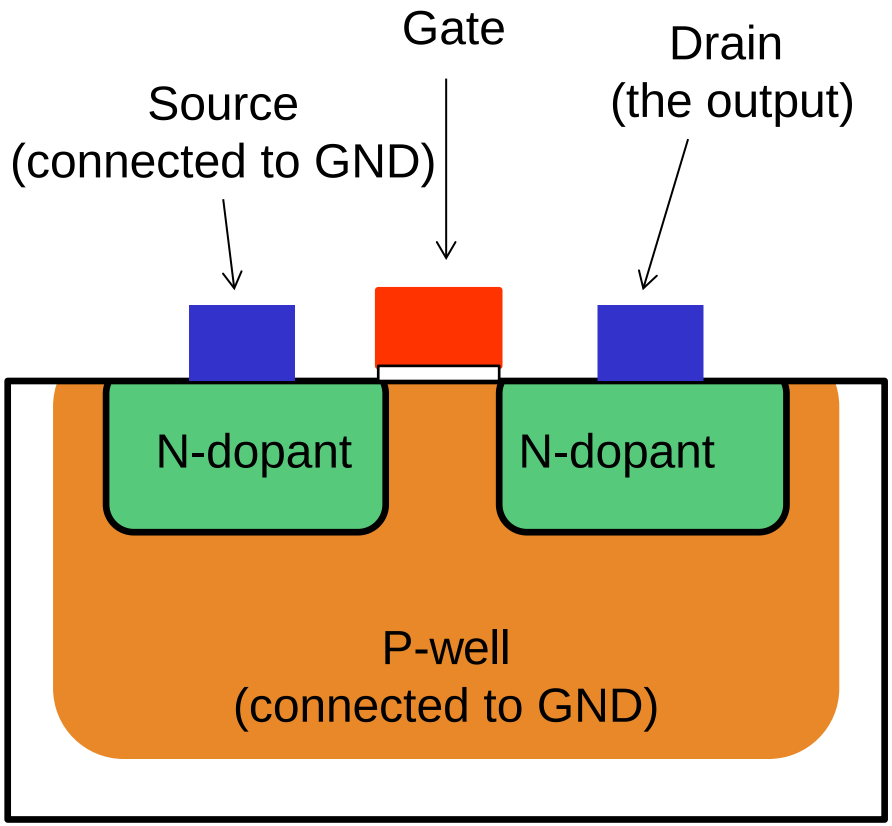
Trojan Transistor with a constant output of VDD

Result after modifying the PMOS:

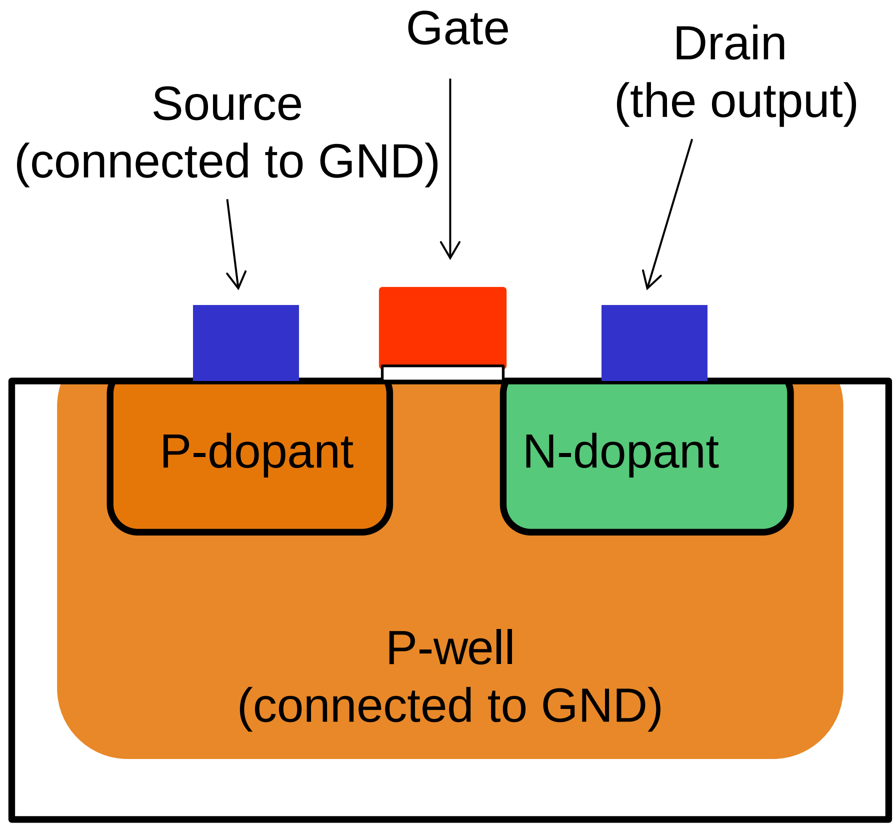
Constant connection to VDD, but the NMOS transistor is still connected.



NMOS Transistor Trojan



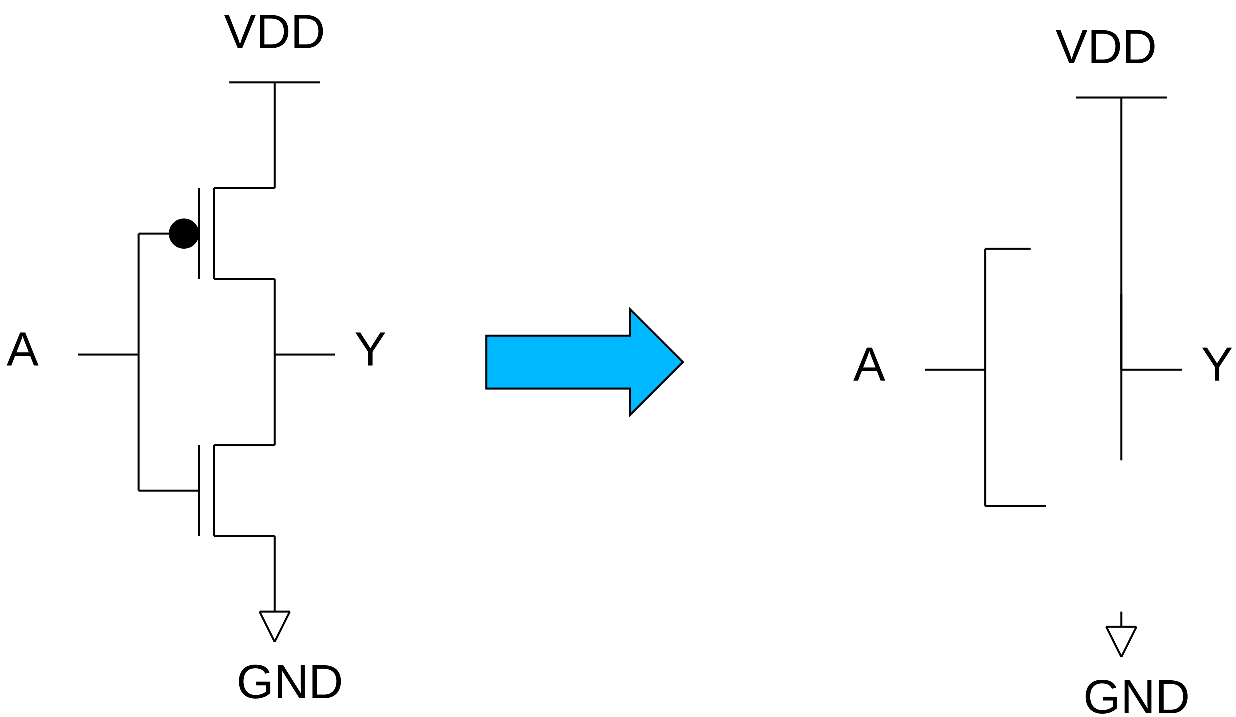
Unmodified PMOS Transistor



Trojan Transistor with a floating output

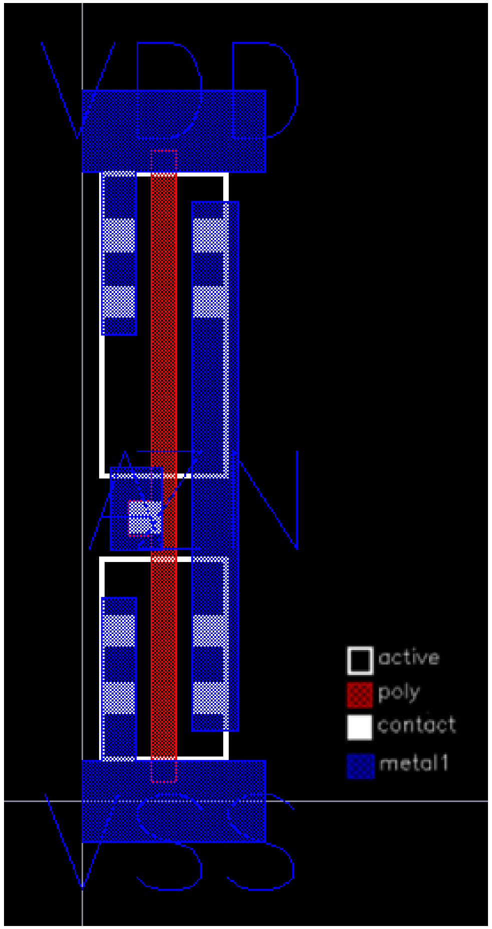
Result: Inverter Trojan

1. The PMOS transistor is replaced with a constant connection to VDD.
2. The source of the NMOS transistor is removed and hence it is floating.

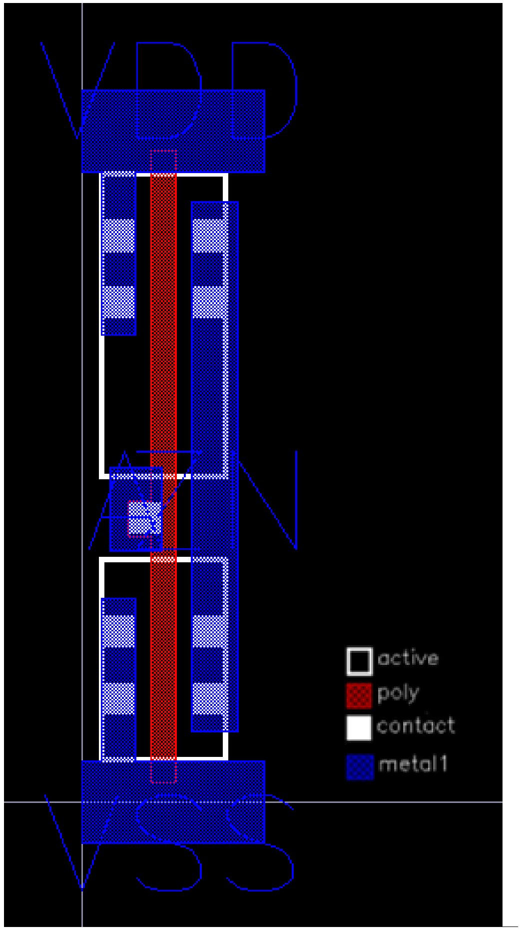


“Always One” Inverter Trojan

Original Inverter



“Always One” Trojan



Unchanged:

- All metal layers
- Polysilicon Layer
- Acitve area
- Wells

⇒ Dopant changes extremely difficult to detect using optical reverse-engineering!

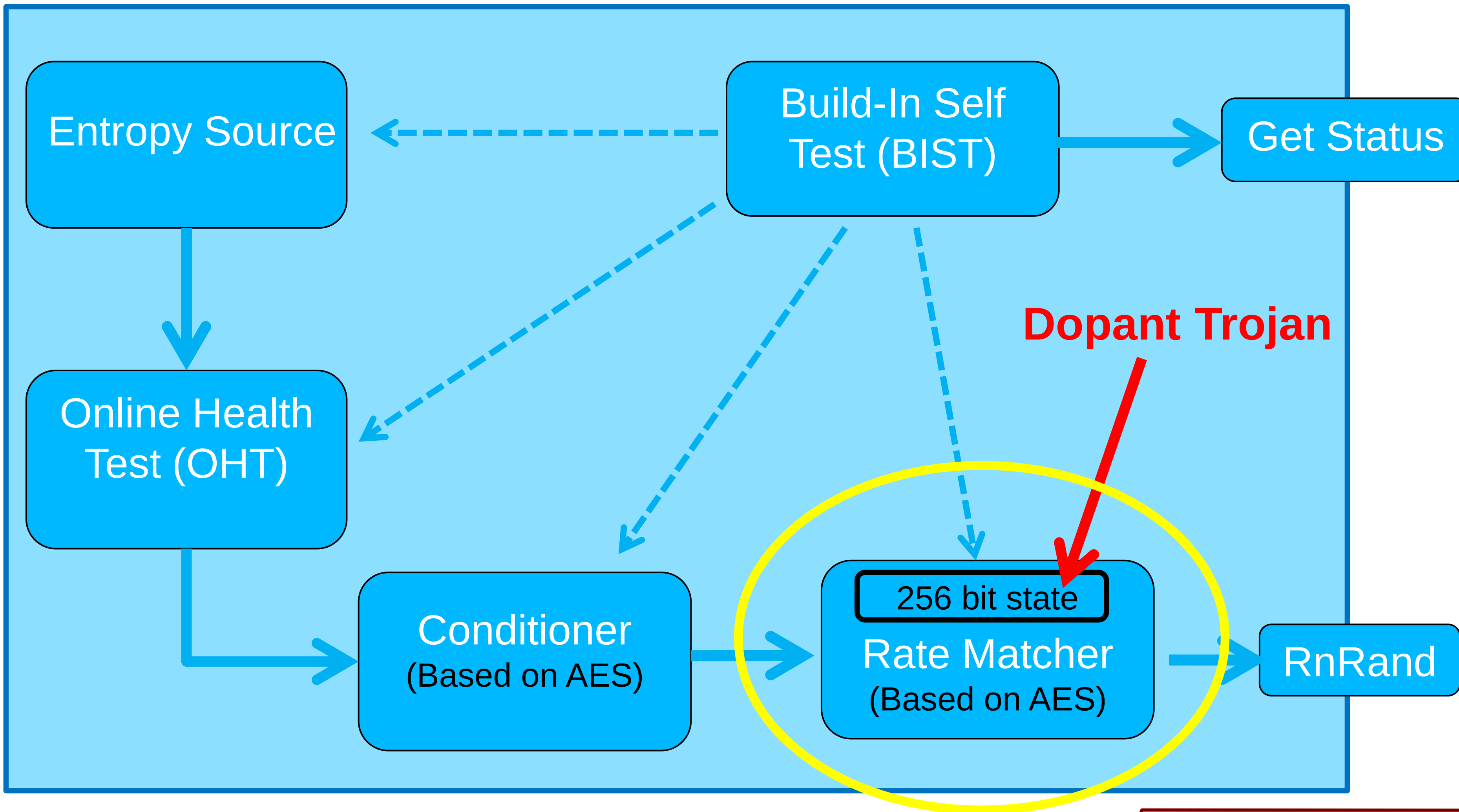
Remaining question:

Can we build a **meaningful** Trojan using dopant modifications that passes **functional testing**?

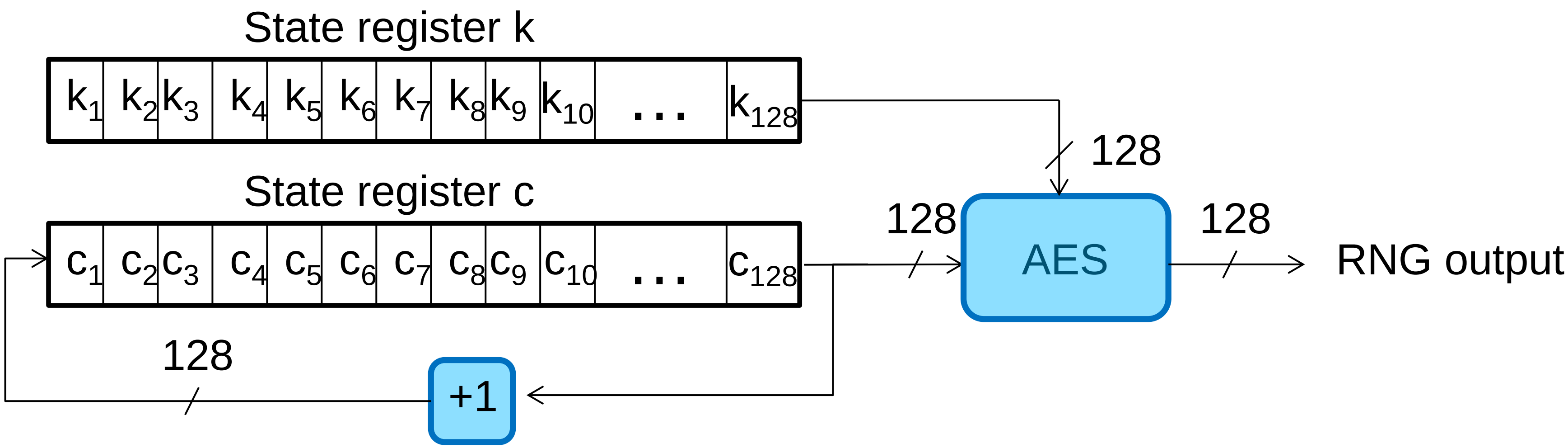
Agenda

- Introduction to Hardware Trojans
- Dopant-Level Hardware Trojans
- **Case study 1: RNG design**
- Case study 2: Side-channel resistant Sbox
- Conclusion & future work

Intel's Ivy Bridge RNG design



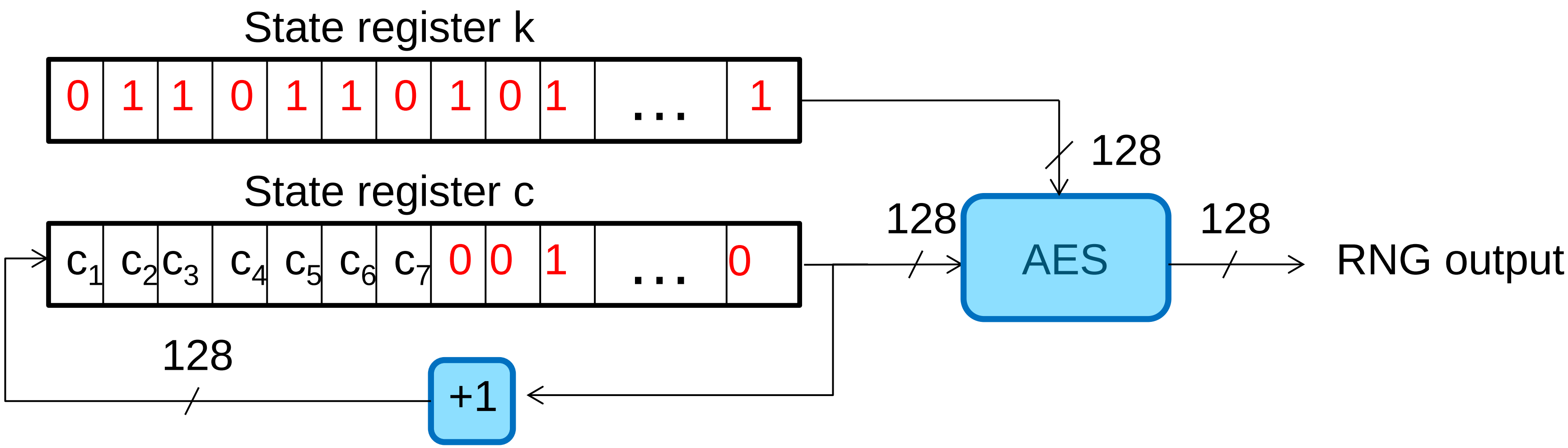
Simplified view of the Rate Matcher



- Rater Matcher uses AES in counter mode
- Stage registers k and c contain truly random numbers
- Stage registers k and c are updated after iteration

Georg T. Becker

Trojan Rate Matcher



- Modify registers of k so that they output a constant
- Modify 128-n registers of c in the same way

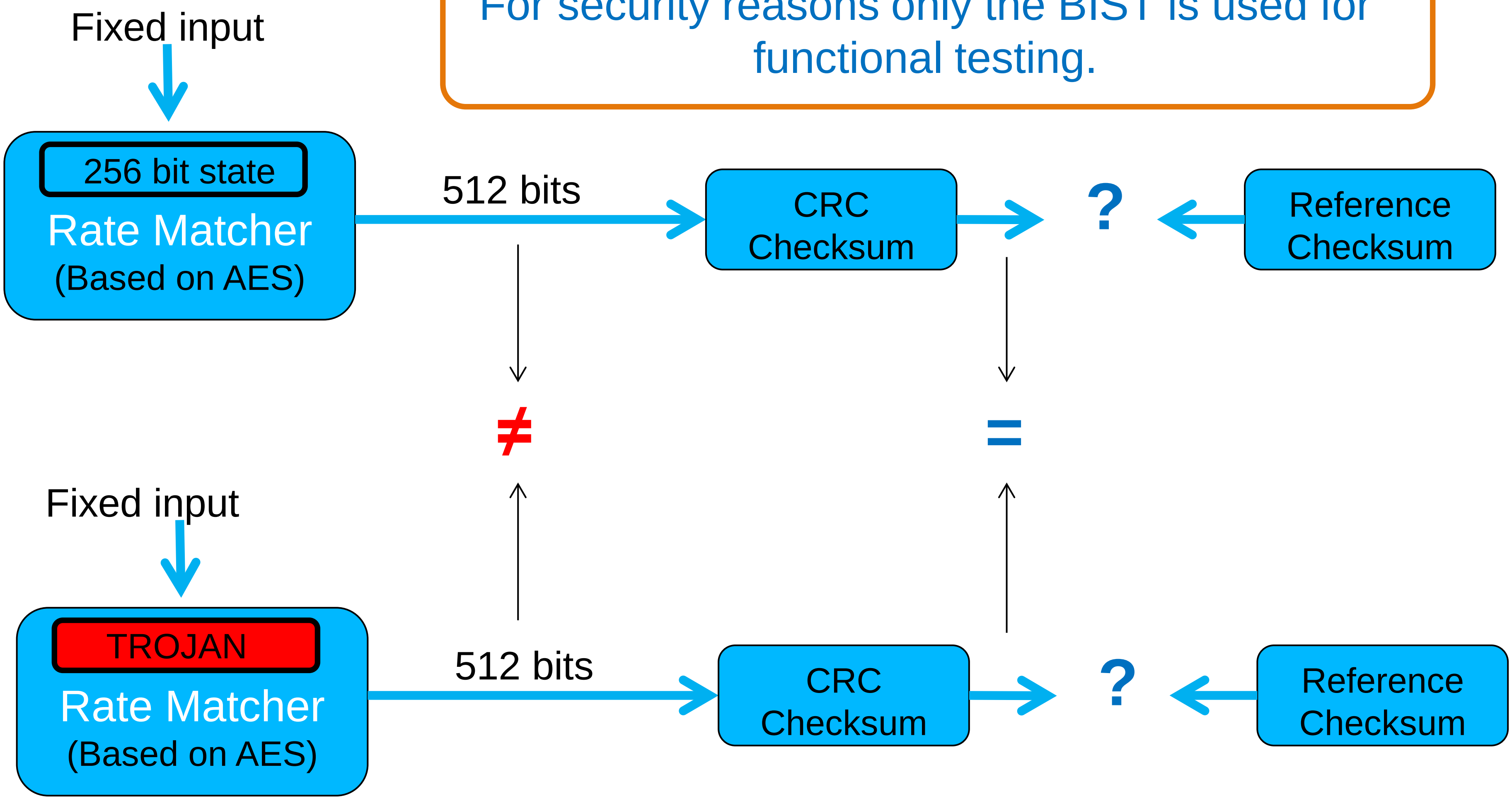
⇒ The output or the RNG depends only on *n* random bits!

⇒ For n=32 the RNG still passes NIST random number test suit

Secret keys generated using this Trojan RNG insecure

Built-In Self Test

For security reasons only the BIST is used for functional testing.



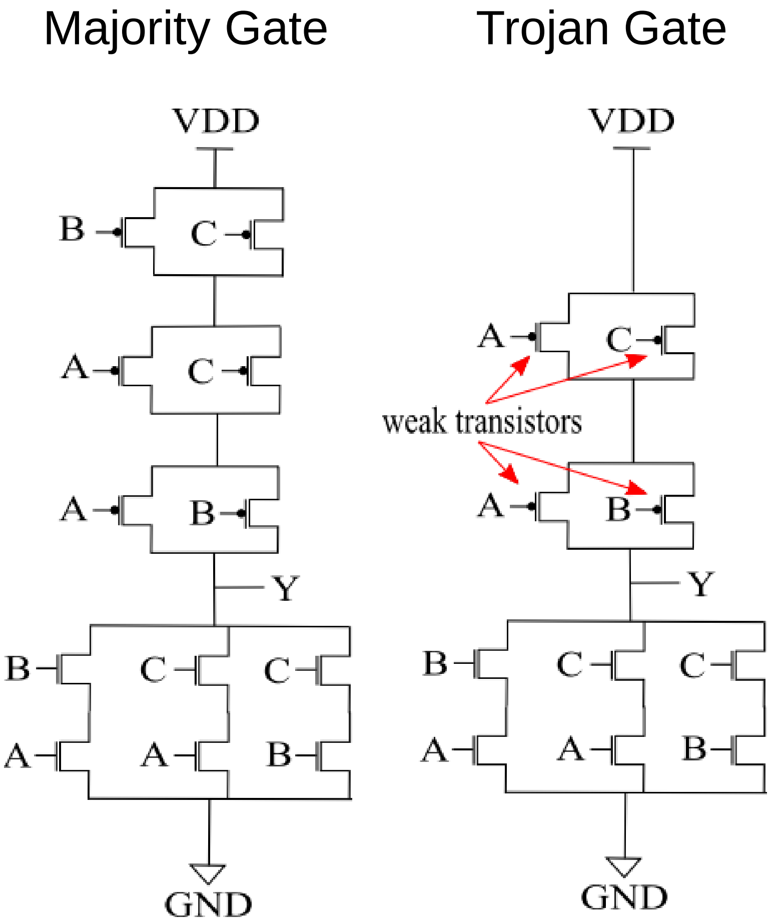
Agenda

- Introduction to Hardware Trojans
- Dopant-Level Hardware Trojans
- Case study 1: TRNG design
- **Case study 2: Side-channel resistant Sbox**
- Conclusion & future work

Target: AES Sbox in side-channel resistant logic style (iMDPL)

- Change the power consumption of only two majority gates of the target design
- No modification to the logic functionality of the entire design!

⇒ Trojan design passes function testing
⇒ Created hidden side-channel that reveals secret key
⇒ Trojan design still resistant against many common side-channel attacks (due to clever placing of the Trojan)



Agenda

- Introduction to Hardware Trojans
- Dopant-Level Hardware Trojans
- Case study 1: TRNG design
- Case study 2: Side-channel resistant Sbox
- **Conclusion & future work**

Conclusion

- Meaningful Hardware Trojans that can pass functional testing can be build by only modifying the dopant.
- Optical-Inspection does not guarantee a Trojan free design!
- Dopant Trojans are flexible, not only logic behavior can be changed but performance such as power consumption or timing as well
- Finding a suitable location the most important part of inserting a Trojan
- Reverse-engineering the design and getting knowledge of the test procedure probably the limiting factor in practice.
- Build-In Self Tests good for detecting defects but not for detecting Trojans

Thank you very much!

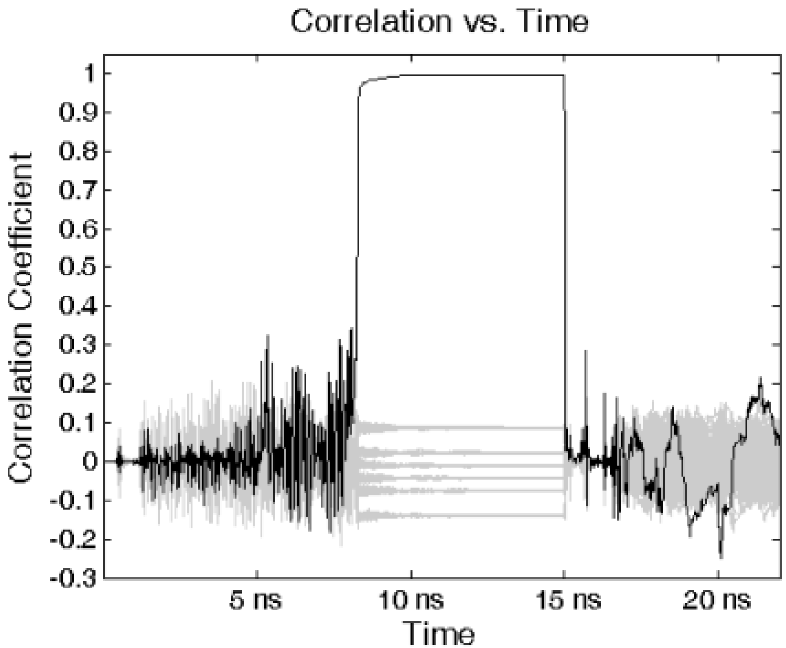
Stealthy Dopant-Level Hardware Trojans

Georg T. Becker[1], Francesco Regazzoni[2], Christof Paar[1,3],
and Wayne P. Burleson[1]
[1]University of Massachusetts Amherst, USA
[2] TU Delft, The Netherlands and ALaRI – University of Lugano, Switzerland
[3] Horst Görtz Instistut for IT Security, Ruhr Universität Bochum, Germany

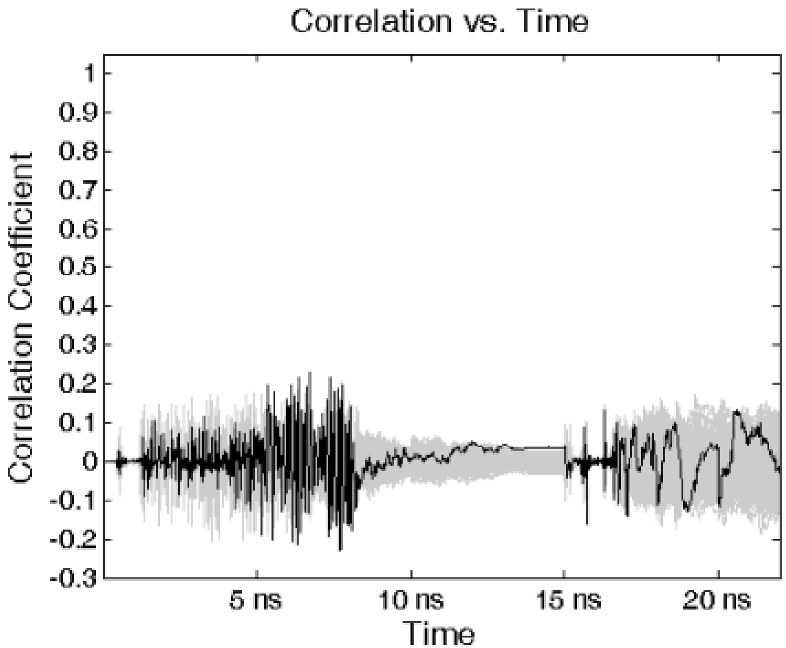
I am graduating this year …
… looking for jobs!

Backup slides

Exploiting the Trojan

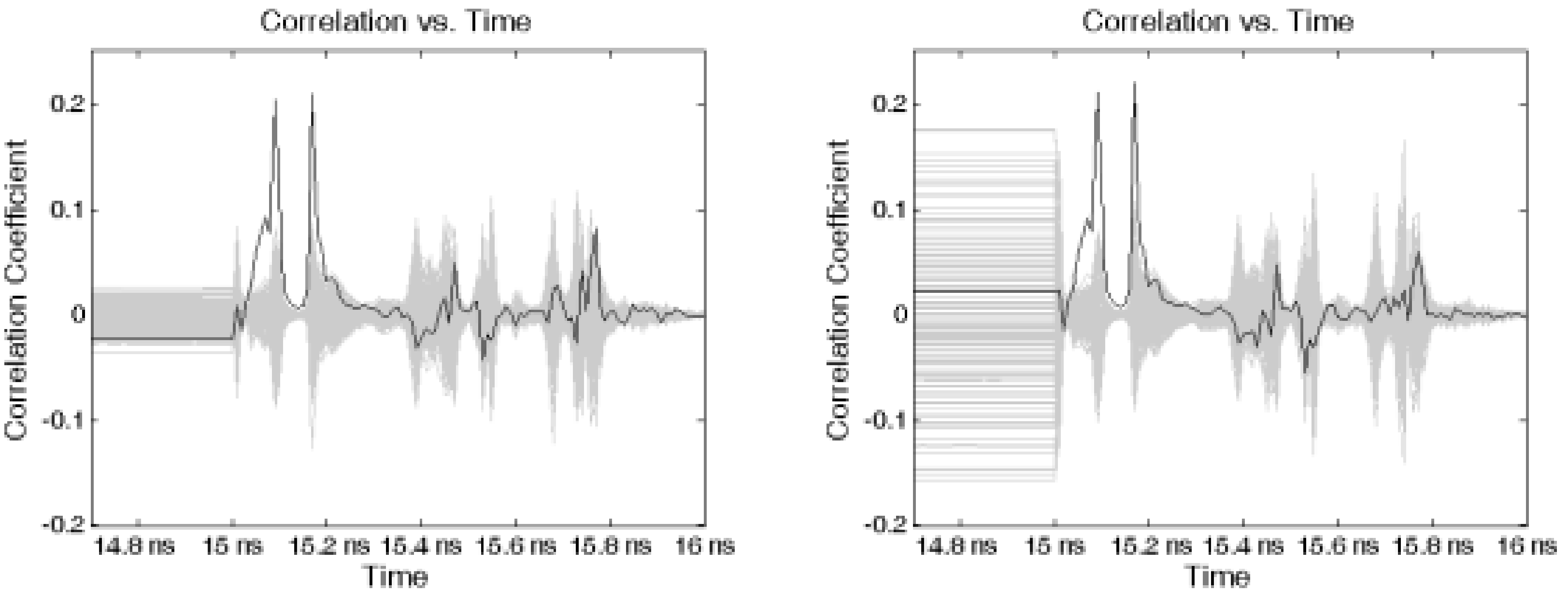


(a) Trojan design



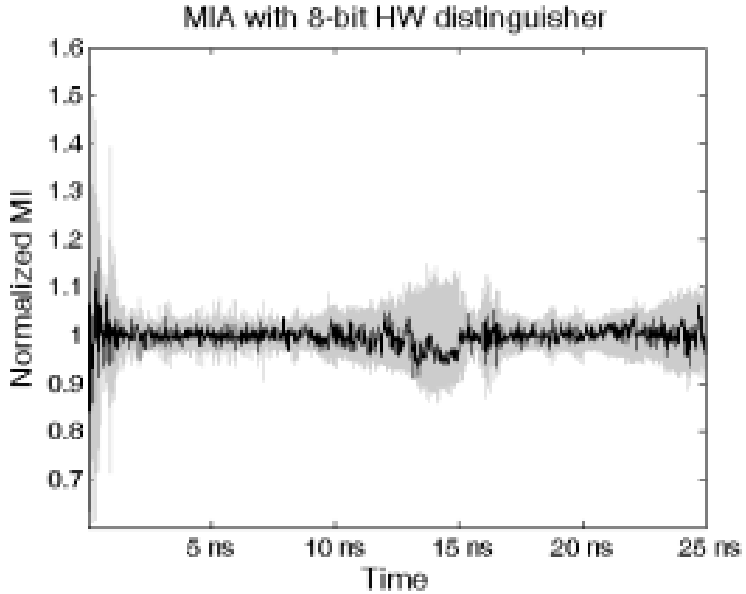
(b) Trojan-free design

8-bit CPA on output of SBox

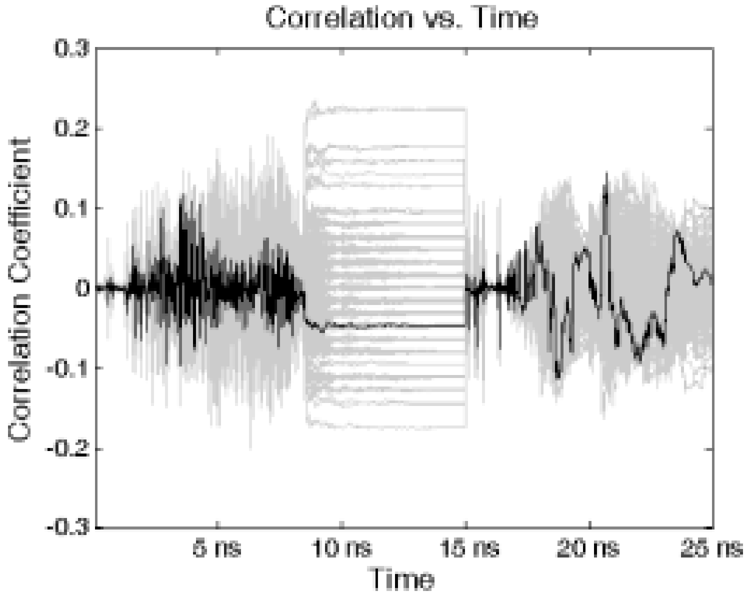


(a) Trojan-free design (b) Trojan design

Other attacks



(a) MIA attack



(b) 1-Bit CPA

Trojan iMDPL Gate:

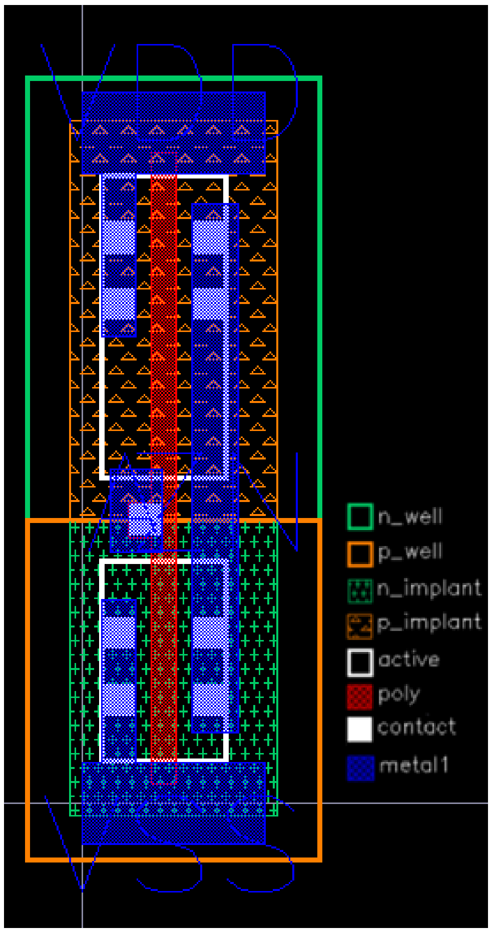
Power consumption of an iMDPL-AND gate

A	B	M	Unmodified iMDPL-AND	Trojan iMDPL-AND
0	0	0	65.61 fJ	63.36 fJ
0	0	1	61.26 fJ	59.31 fJ
0	1	0	66.89 fJ	63.79 fJ
0	1	1	65.34 fJ	62.50 fJ
1	0	0	68.48 fJ	121.47 fJ
1	0	1	66.70 fJ	119.92 fJ
1	1	0	63.19 fJ	61.57 fJ
1	1	1	64.43 fJ	62.63 fJ

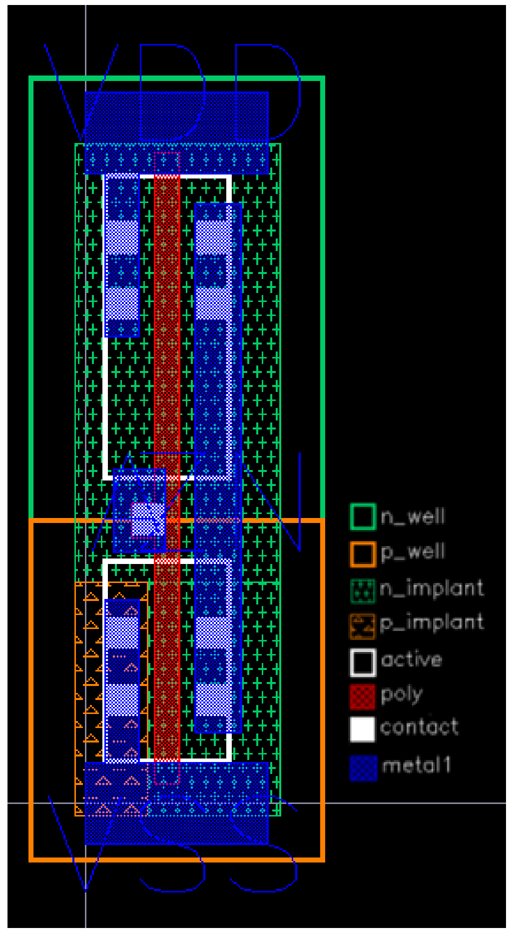
Logic behavior is unchanged!

“Always One” Inverter Trojan

Original Inverter



“Always One” Trojan



<u>The PMOS Transistor</u>

Replaced the P-type dopant with N-type dopant

⇒The contacts are now connected to the N-Well know

⇒Drain and Source are both connected to VDD

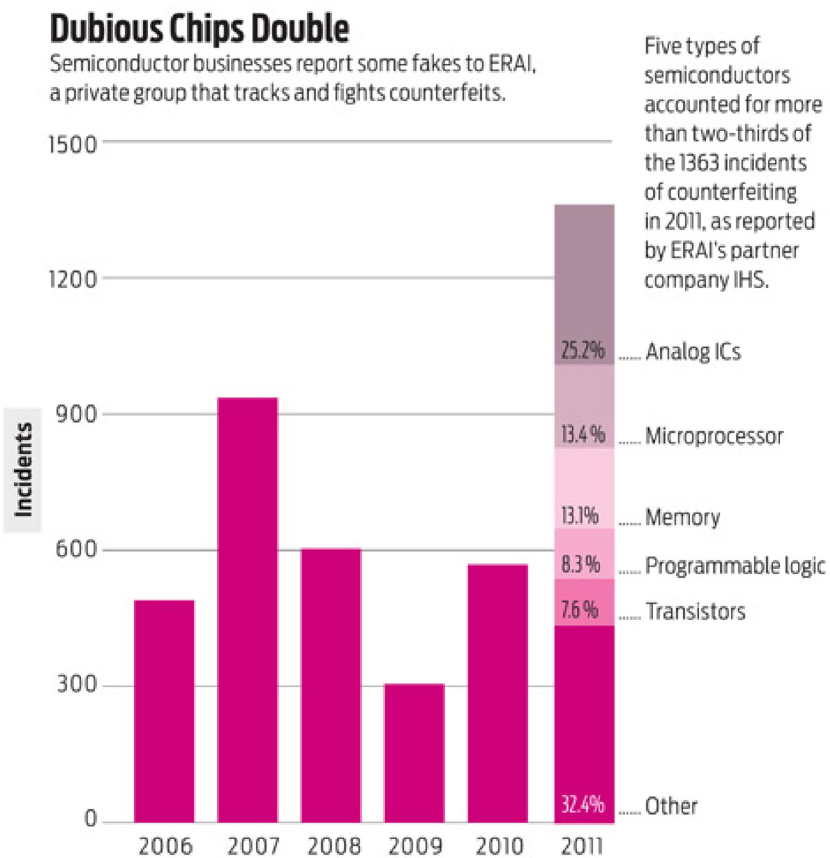
<u>The NMOS Transistor</u>

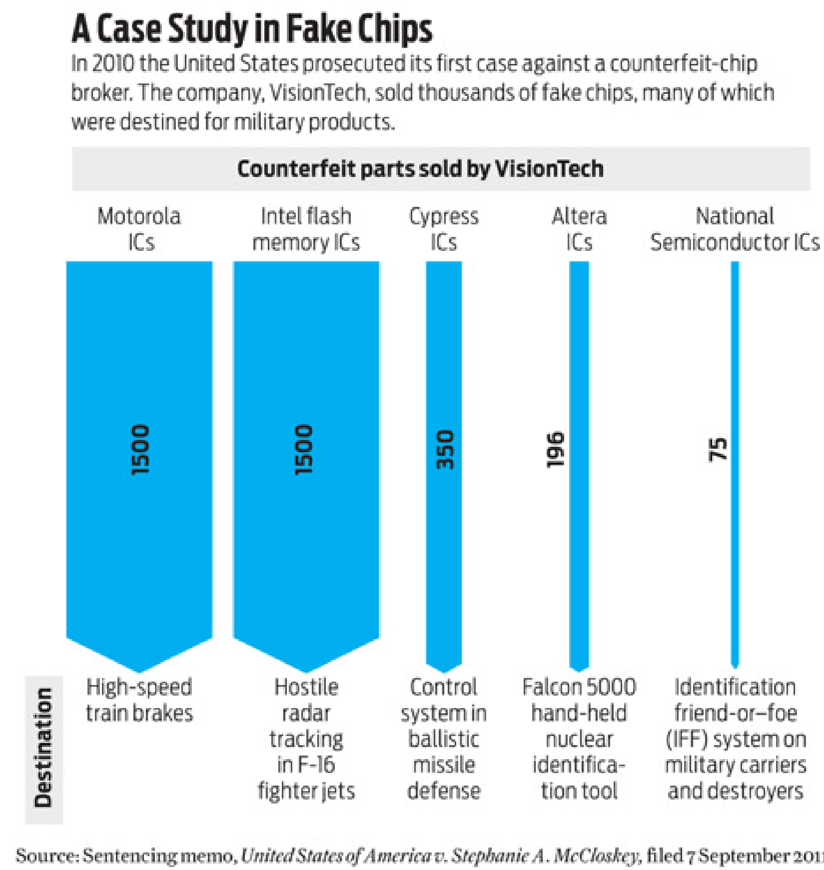
Replaced the N-type dopant of the source contact with P-type dopant

⇒The source contact is now connected to the P-well

⇒The NMOS transistor is “cut off” from GND

Counterfeit ICs





http://spectrum.ieee.org/computing/hardware/counterfeit-chips-on-the-rise