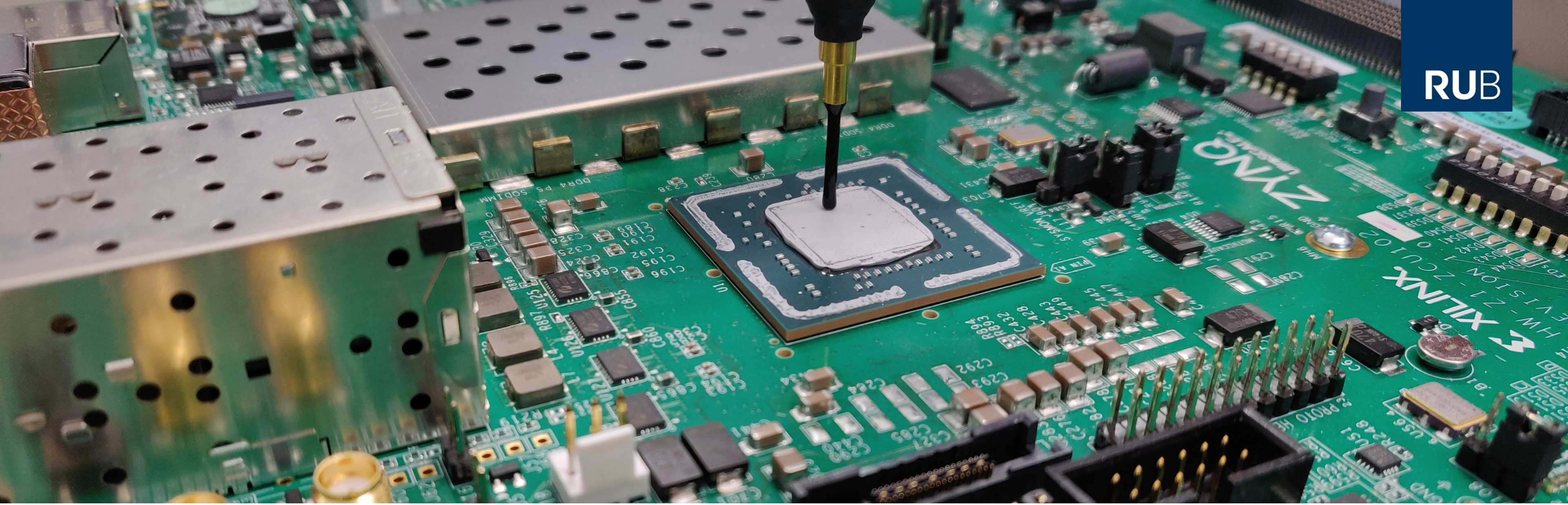


RUHR-UNIVERSITÄT BOCHUM

Computer-Aided Verification of Countermeasures against Physical Attacks

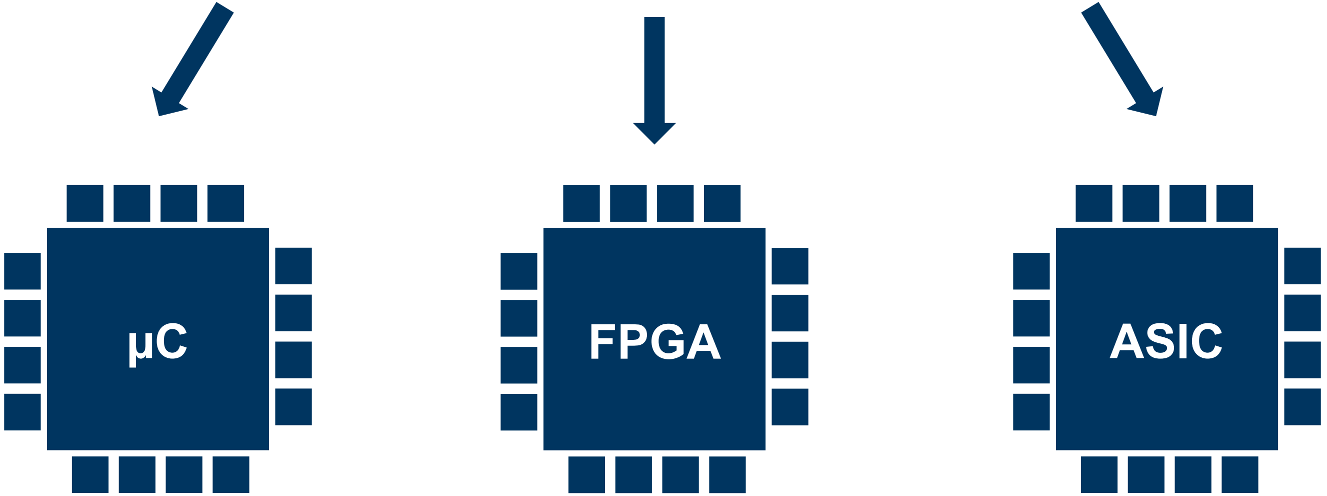
Jan Richter-Brockmann, Jakob Feldtkeller, Pascal Sasdrich, Tim Güneysu

Chair for Security Engineering
Faculty of Computer Science
Ruhr University Bochum

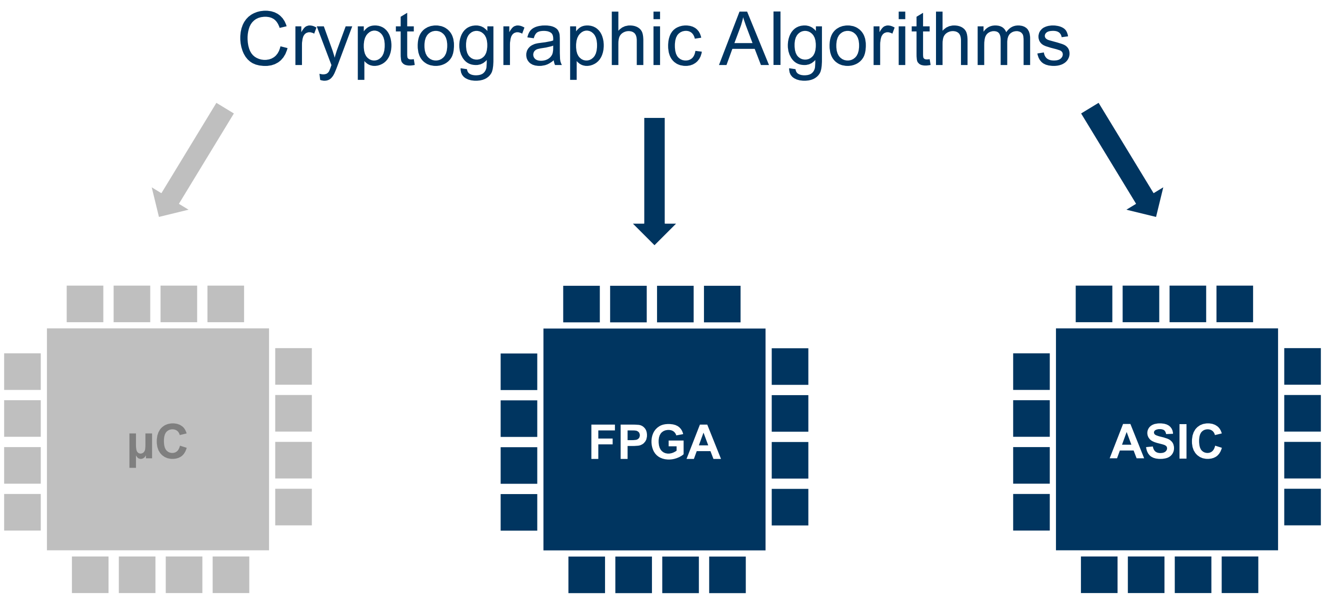
16. Januar 2025

Cryptography on Embedded Devices

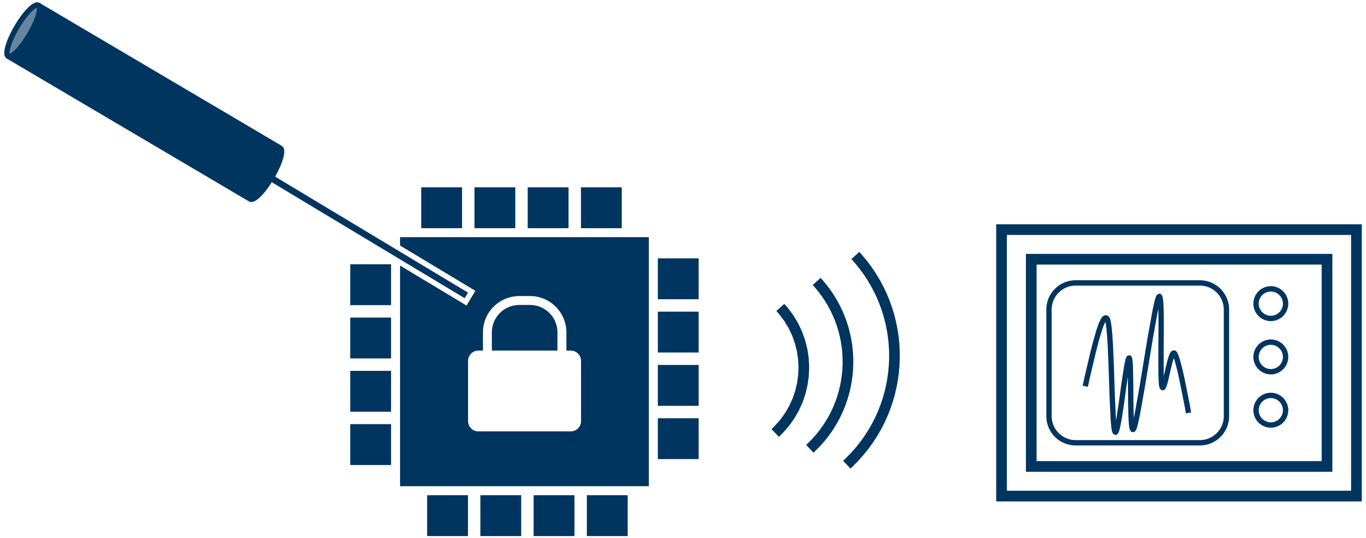
Cryptographic Algorithms



Cryptography on Embedded Devices



Physical Attacks



Fault-Injection Attacks

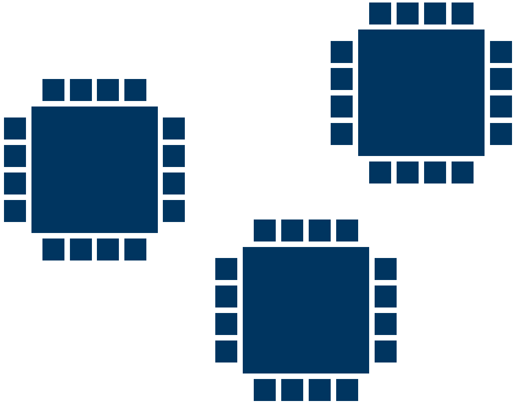
Side-Channel Attacks

Countermeasures

Side-Channel Attacks

Masking

Fault-Injection Attacks

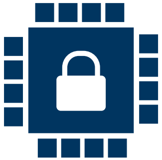


Redundancy

Research Questions

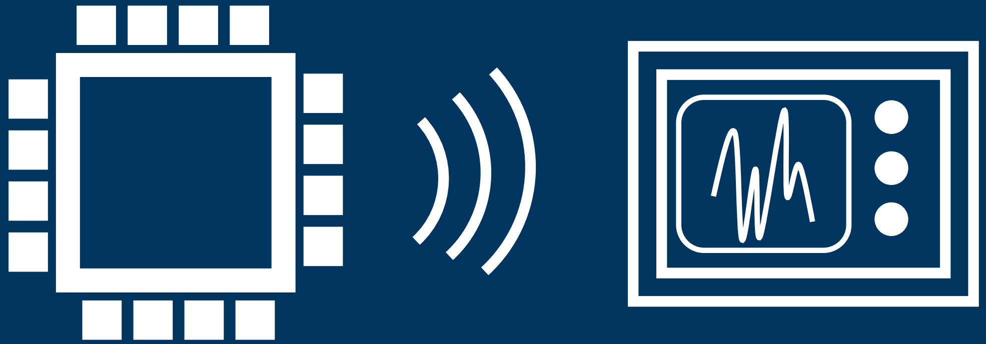
1. Combined Attacks

2. Combined Protection

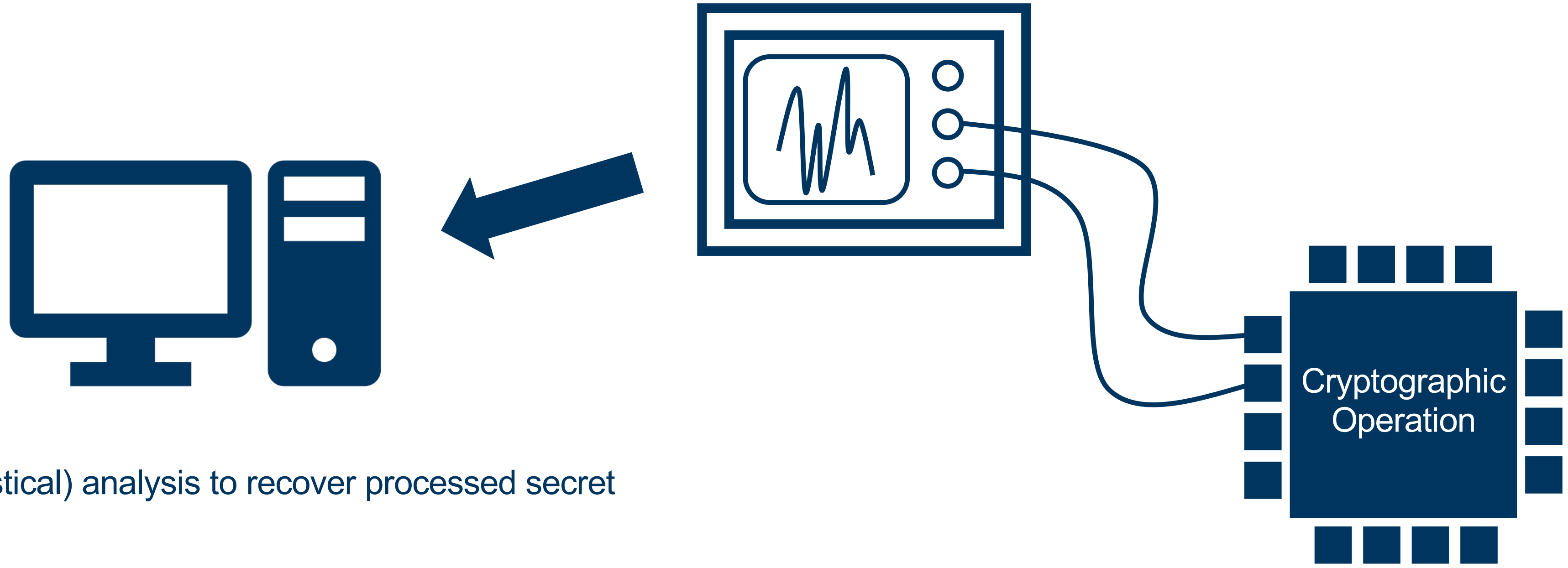


3. Formal Verification

Side-Channel Analysis



Side-Channel Attacks



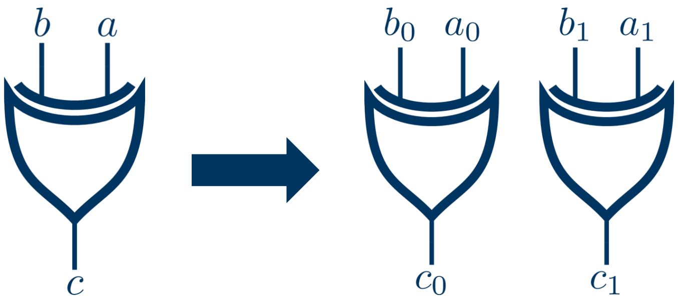
(Statistical) analysis to recover processed secret

Measure power consumption during ongoing operation

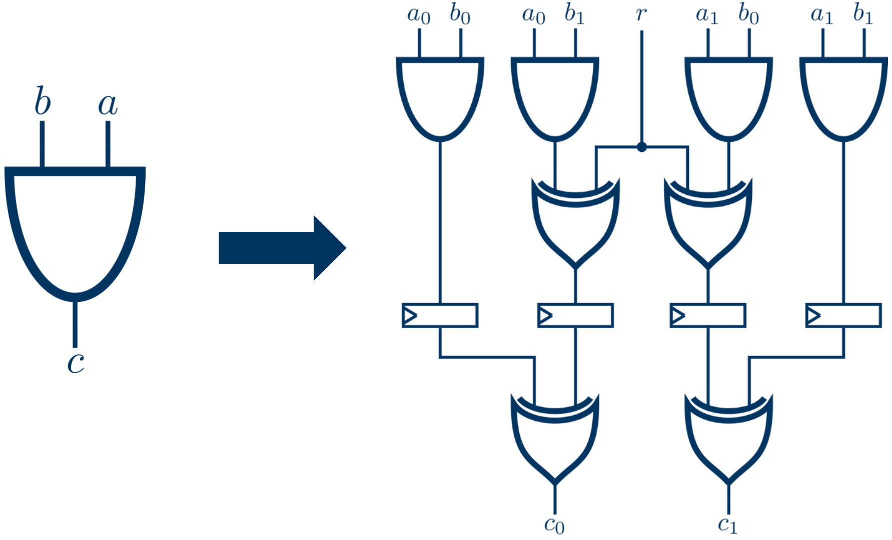
Countermeasures against Side-Channel Attacks

$$x = x_0 \oplus x_1 \oplus \ldots \oplus x_{s-1}$$

Linear Functions

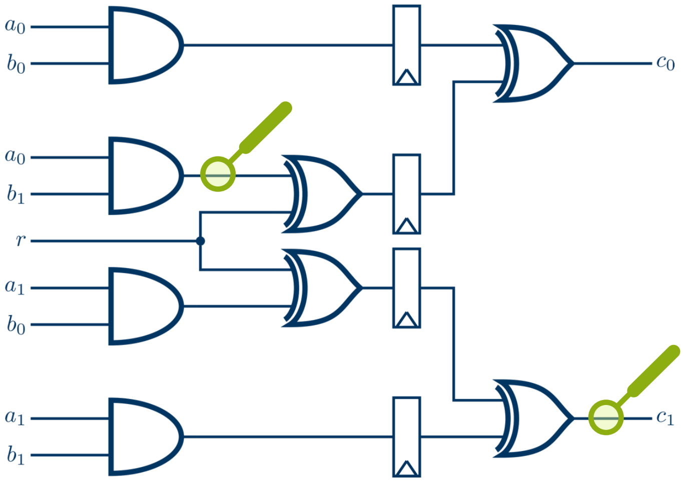


Non-linear Functions



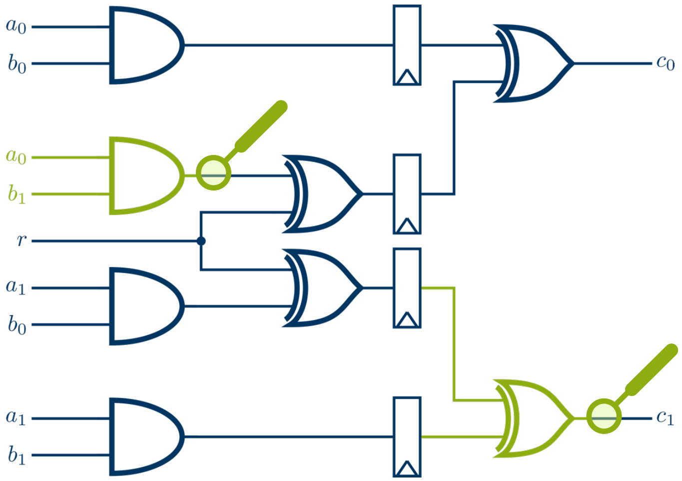
Modeling Side-Channel Attacks

d-probing model [ISW03]



An adversary is given the exact values of up to d wires of a circuit $\mathtt{C}$.

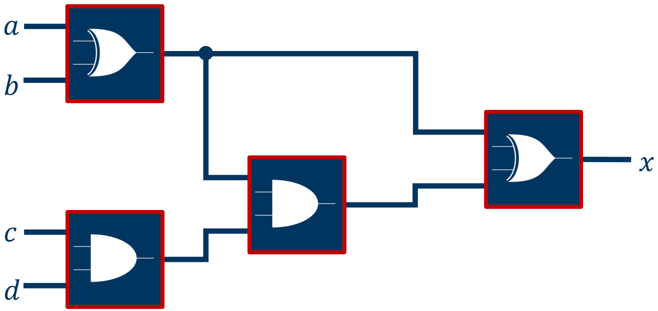
Glitch-extended d-probing model [FGP+18]

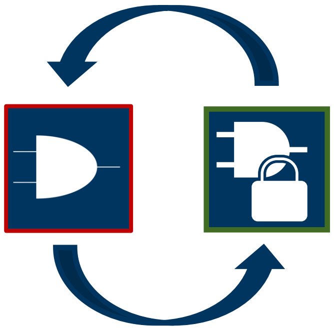


An adversary is given the exact values of all synchronization points influencing up to d wires of a circuit $\mathtt{C}$.

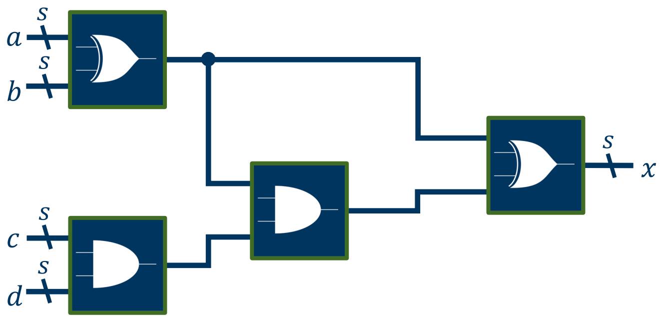
Protection by Secure Gadgets

Insecure Circuit





Protected Circuit



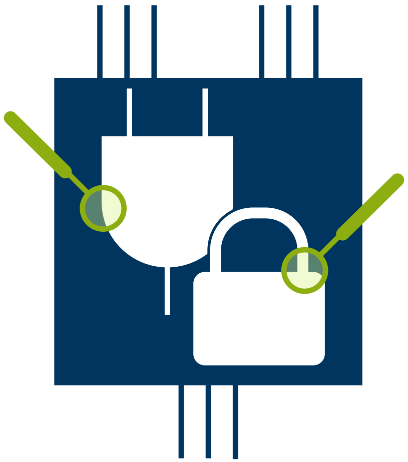
Replace insecure gates by secure gadgets

Share inputs and outputs

Maintain timing (pipelining)

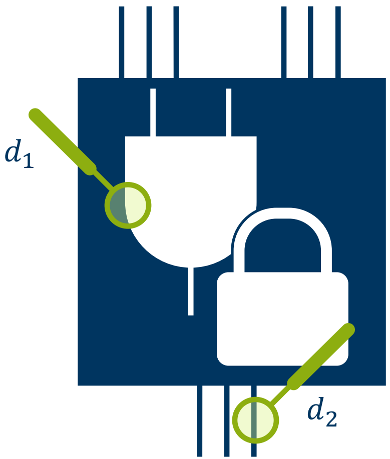
Modeling Side-Channel Attacks – Composability

PNI [BBD+15]
Probe Non-Interference



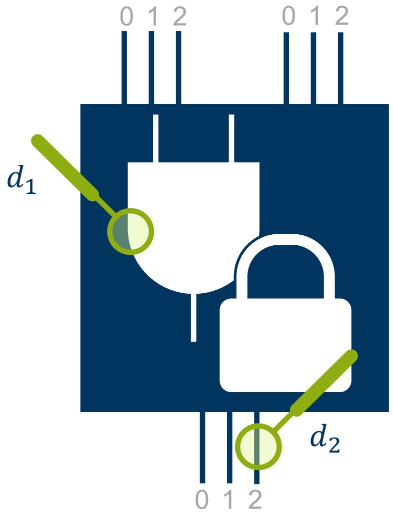
$d' \leq d$

PSNI [BBD+16]
Probe Strong Non-Interference



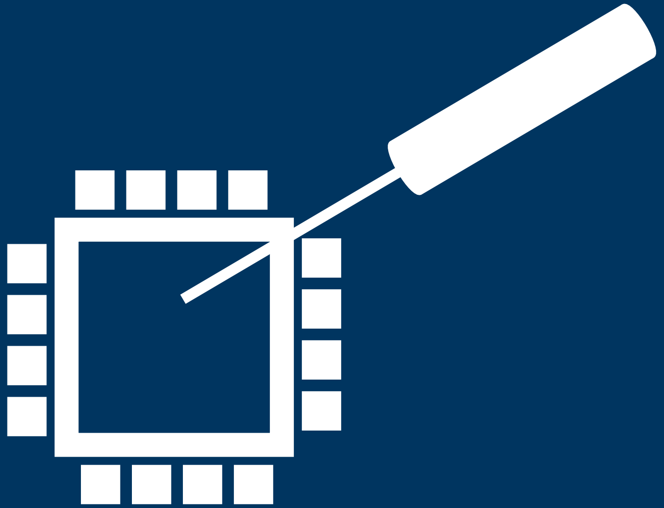
$d_1 + d_2 \leq d$

PINI [CS20]
Probe-Isolating Non-Interference



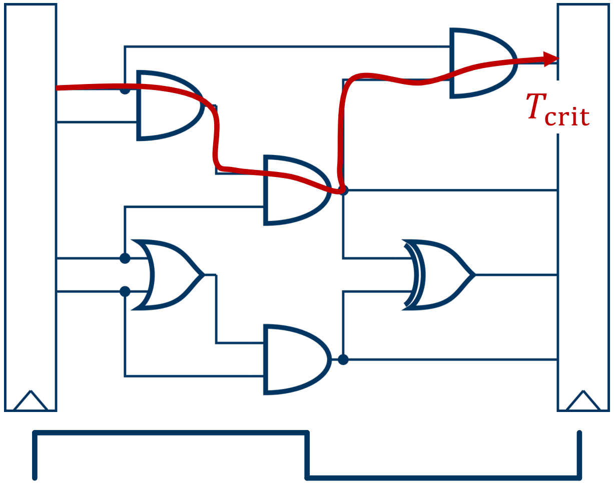
$d_1 + d_2 \leq d$

Fault-Injection Attacks



Fault-Injection Attacks [RBSG21]

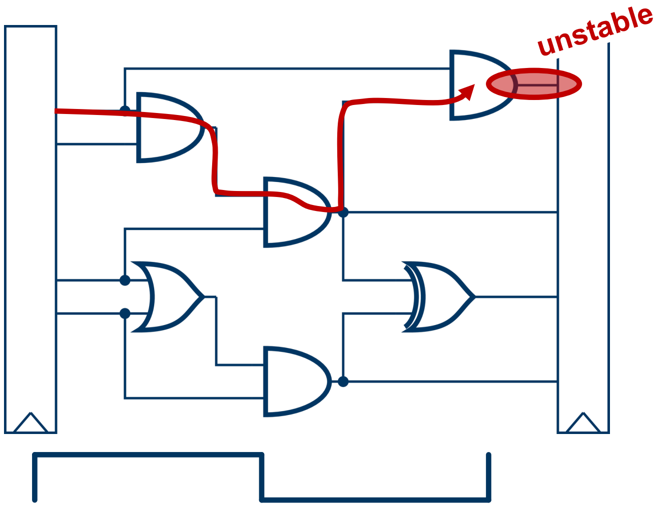
Clock Glitches



$$T_{\text{clk}} \geq T_{\text{crit}} + t_{\text{clkq}} + t_{\text{setup}} - \delta$$

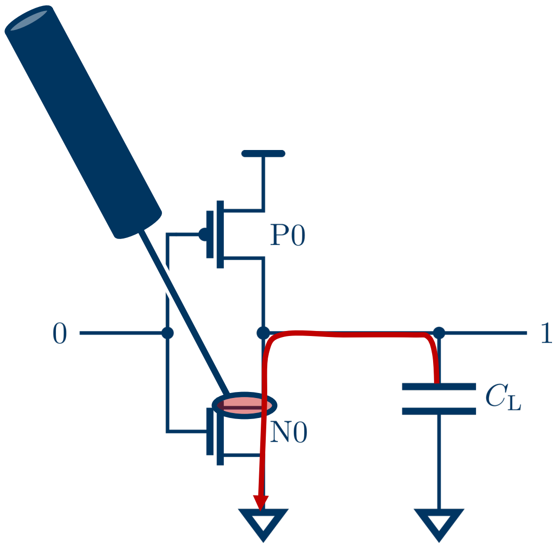
Fault-Injection Attacks [RBSG21]

Clock Glitches



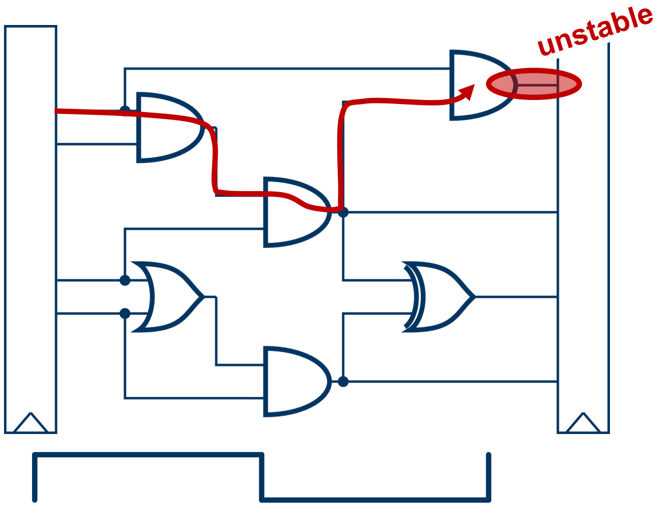
$$T'_{\text{clk}} < T_{\text{crit}} + t_{\text{clkq}} + t_{\text{setup}} - \delta$$

Laser Fault Injection



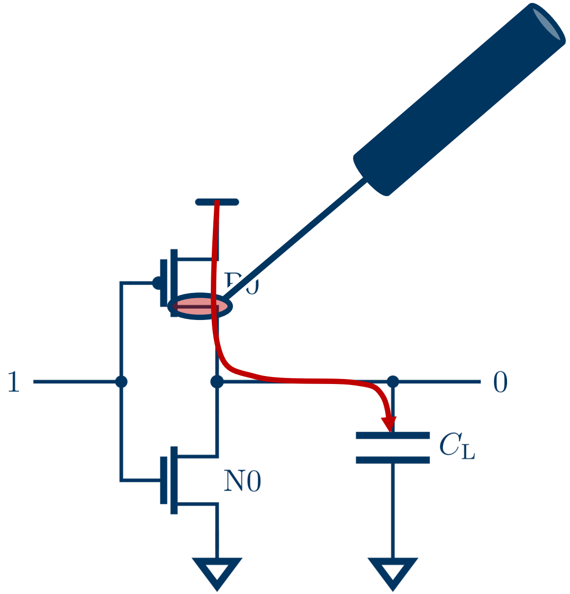
Fault-Injection Attacks [RBSG21]

Clock Glitches



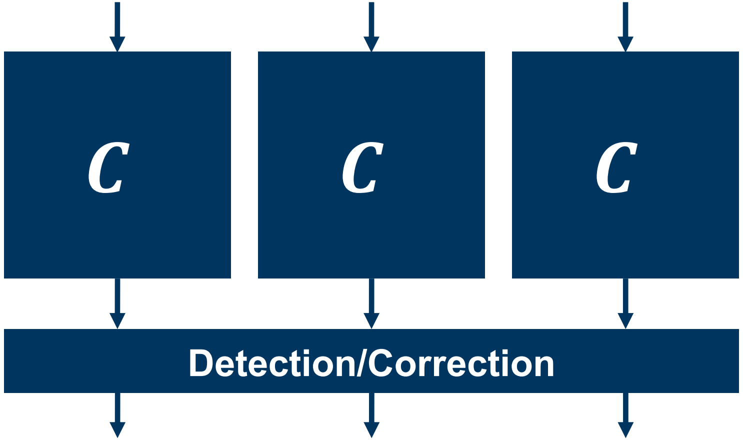
$$T'_{\mathrm{clk}} < T_{\mathrm{crit}} + t_{\mathrm{clkq}} + t_{\mathrm{setup}} - \delta$$

Laser Fault Injection



Countermeasures against Fault-Injection Attacks

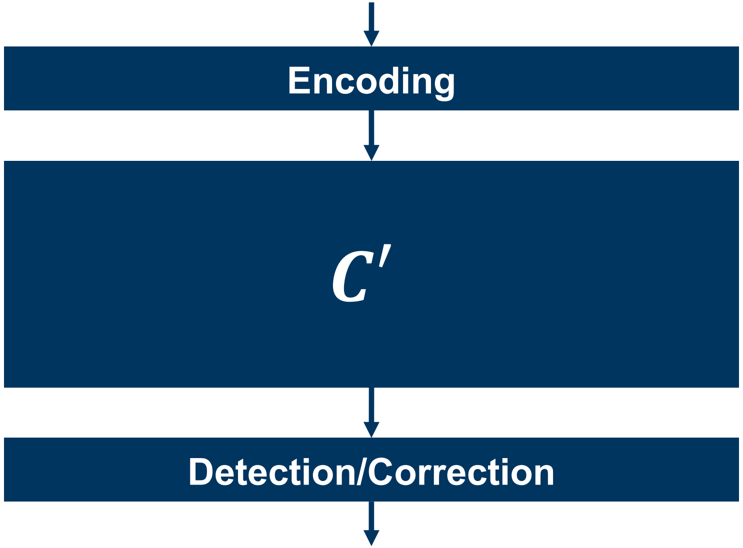
Duplication



Detection: $k+1$ instantiations

Correction: $2k+1$ instantiations

Linear Error Correcting Codes



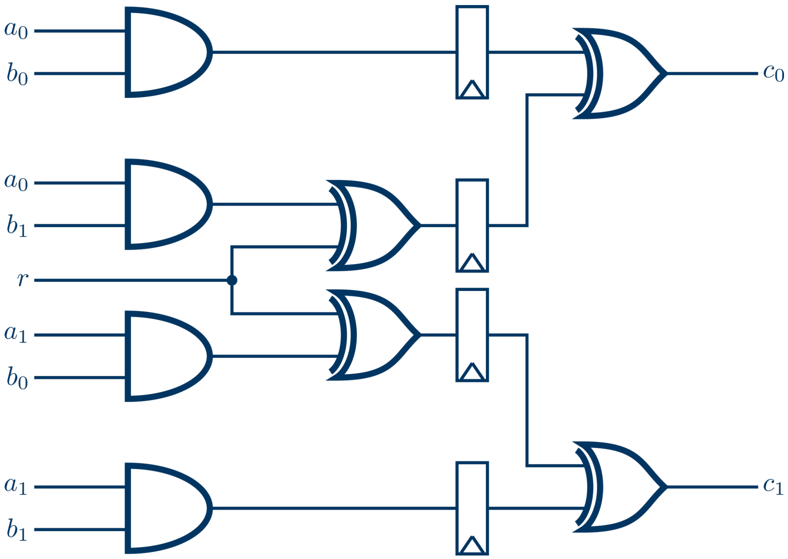
Detection: $d_{\min} - 1$

Correction: $\left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$

Modeling Fault-Injection Attacks [RBSG21]

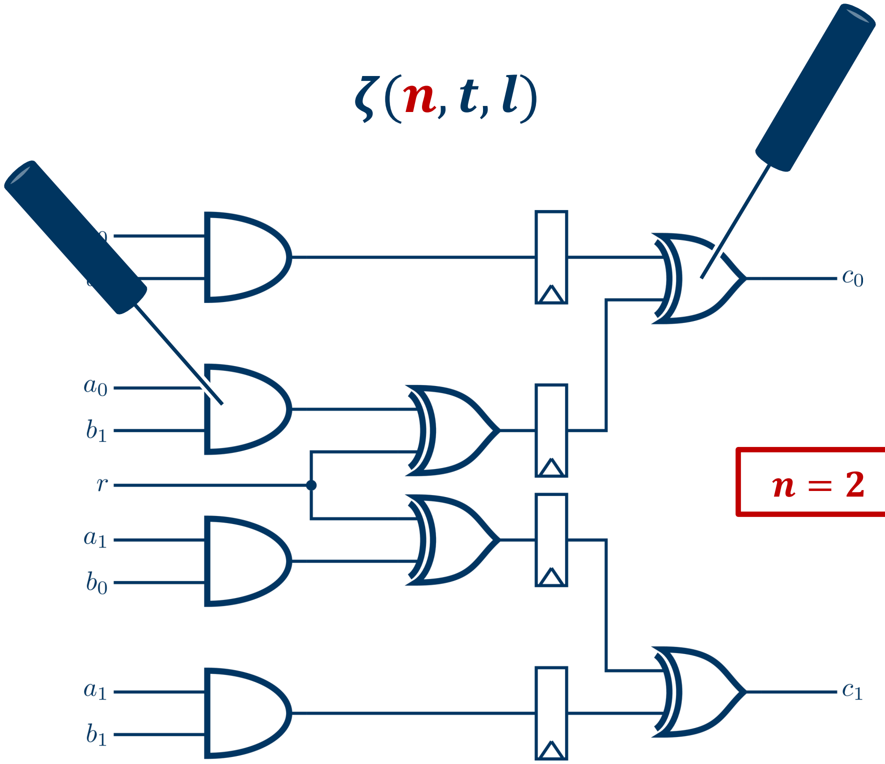
RUB

$$\zeta(n, t, l)$$



Modeling Fault-Injection Attacks [RBSG21]

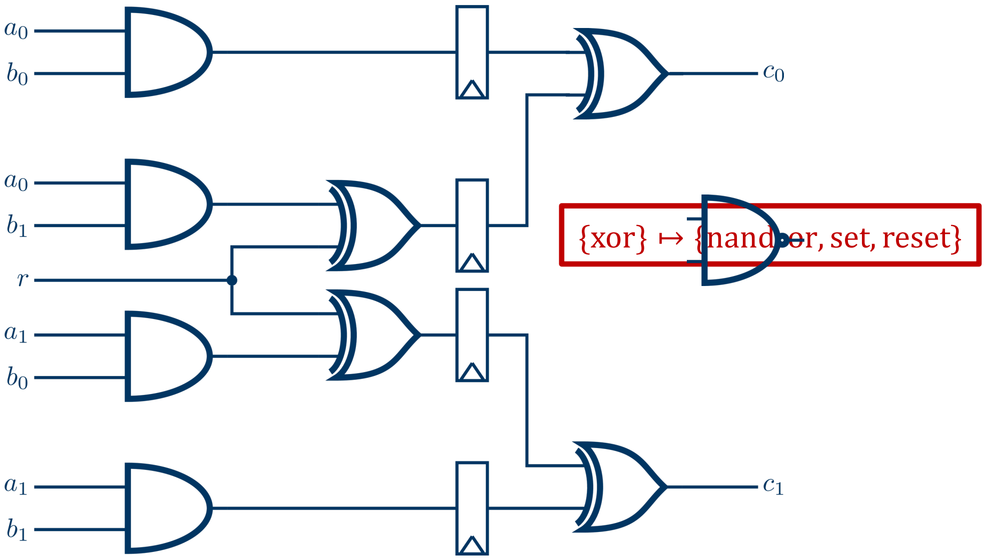
$$\zeta(\boldsymbol{n}, \boldsymbol{t}, \boldsymbol{l})$$



$\boldsymbol{n} = 2$

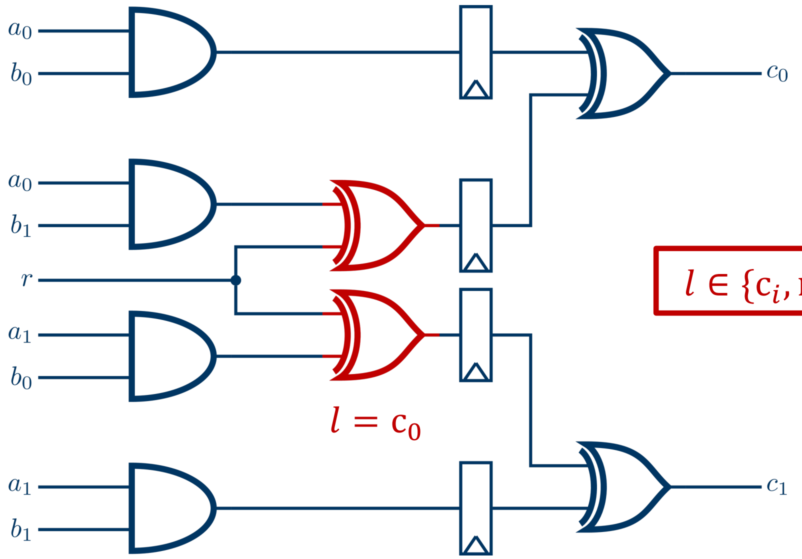
Modeling Fault-Injection Attacks [RBSG21]

$$\zeta(n, t, l)$$



Modeling Fault-Injection Attacks [RBSG21]

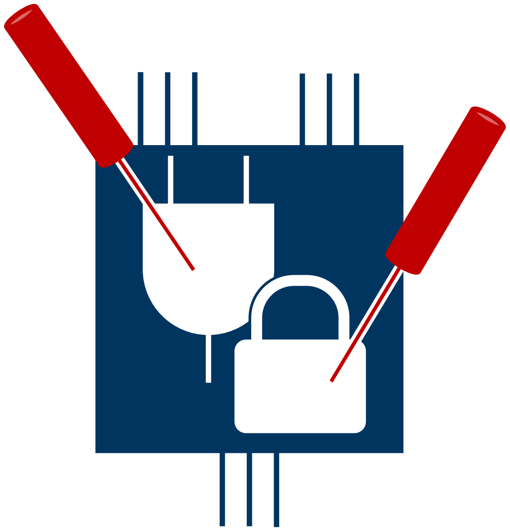
$$\boldsymbol{\zeta(n, t, \textcolor{red}{l})}$$



$l \in \{\mathrm{c}_i, \mathrm{m}, \mathrm{mc}_i\}$

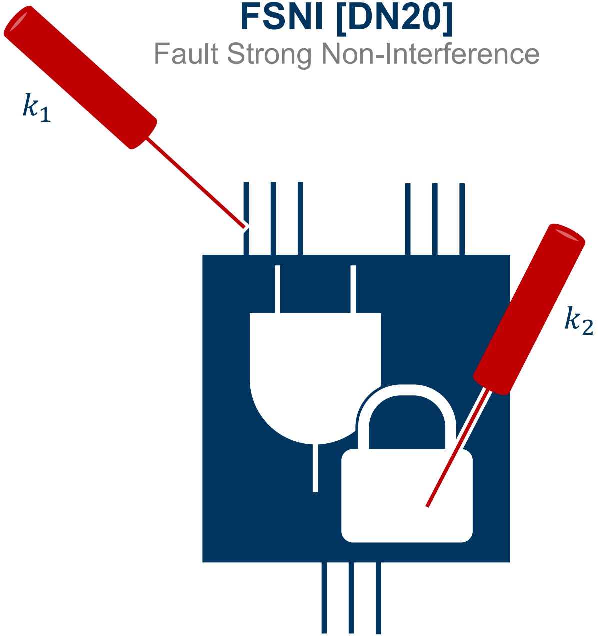
Modeling Fault-Injection Attacks – Composability

FNI [DN20]
Fault Non-Interference



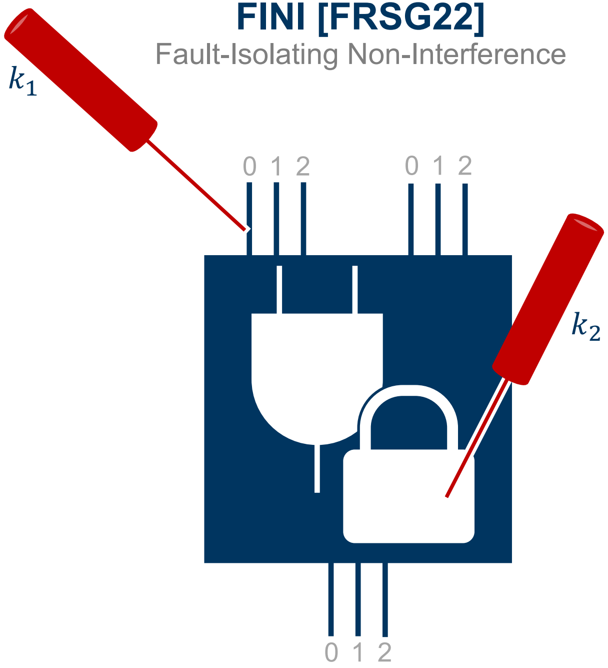
$k' \leq k$

FSNI [DN20]
Fault Strong Non-Interference



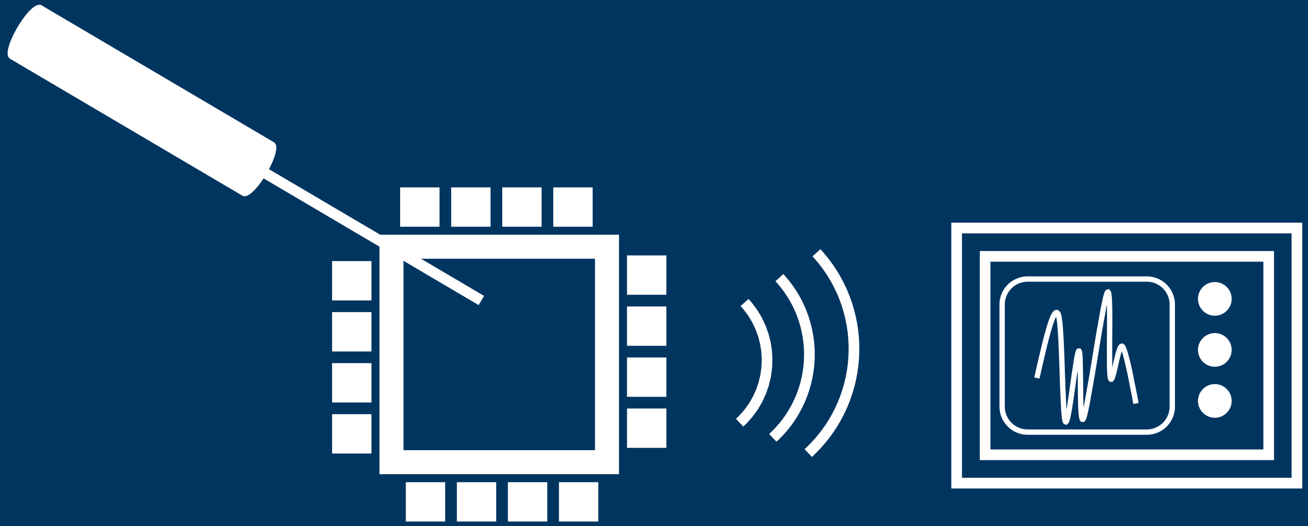
$k_1 + k_2 \leq k$

FINI [FRSG22]
Fault-Isolating Non-Interference



$k_1 + k_2 \leq k$

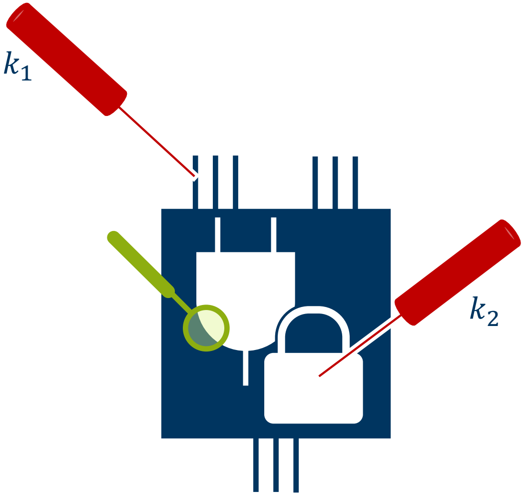
Combined Attacks



Modeling Fault-Injection Attacks – Composability

CNI [DN20]

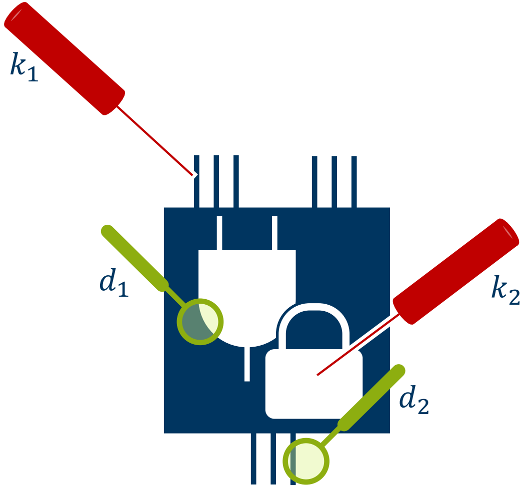
Combined Non-Interference



$$d' + k_1 + k_2 \leq d$$
$$k_1 + k_2 \leq k$$

CSNI [DN20]

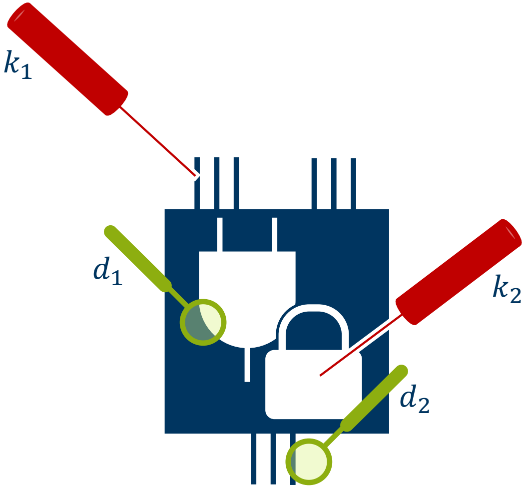
Combined Strong Non-Interference



$$d_1 + d_2 + k_1 + k_2 \leq d$$
$$k_1 + k_2 \leq k$$

ICSNI [DN20]

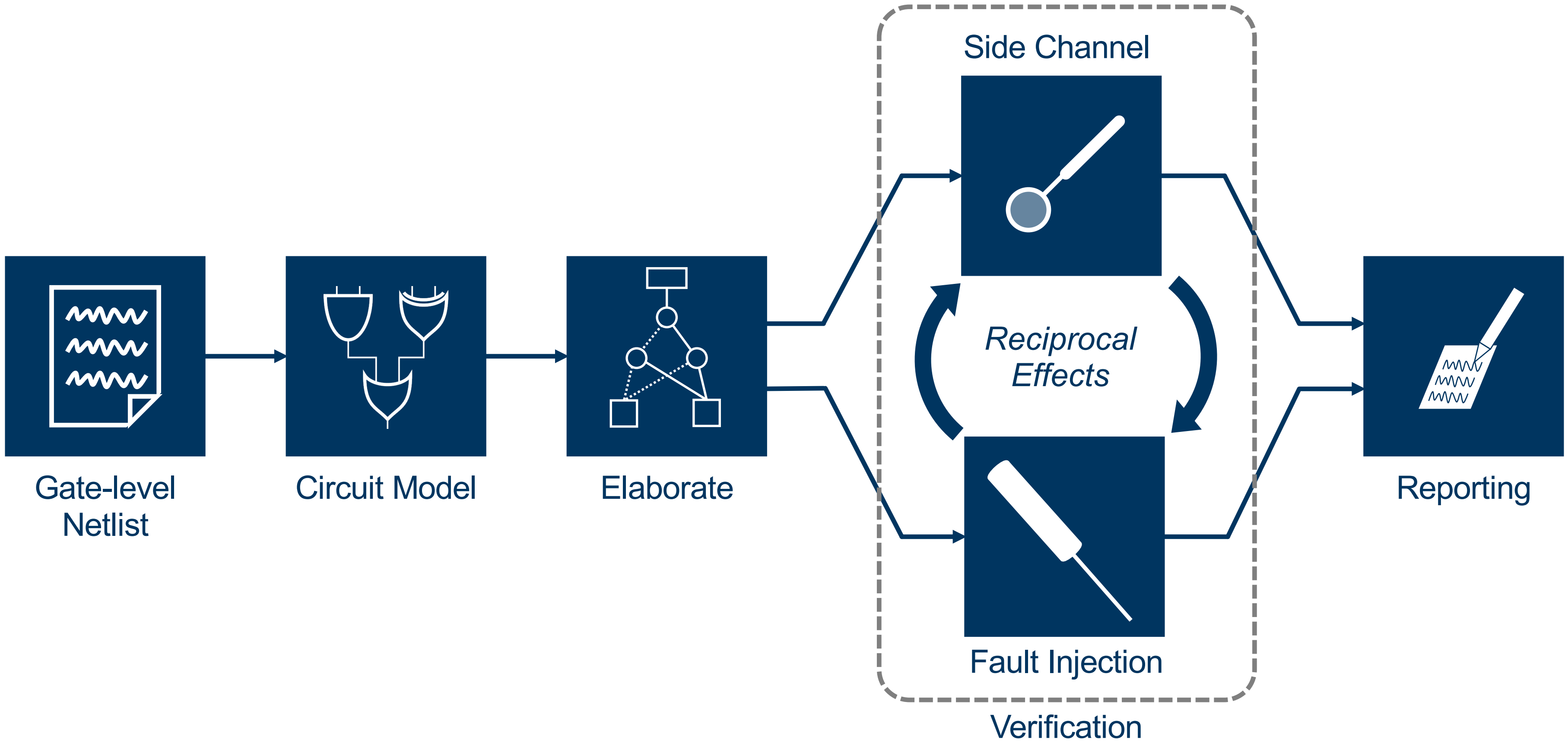
Independent Combined Strong Non-Interference



$$d_1 + d_2 \leq d$$
$$k_1 + k_2 \leq k$$

Verification

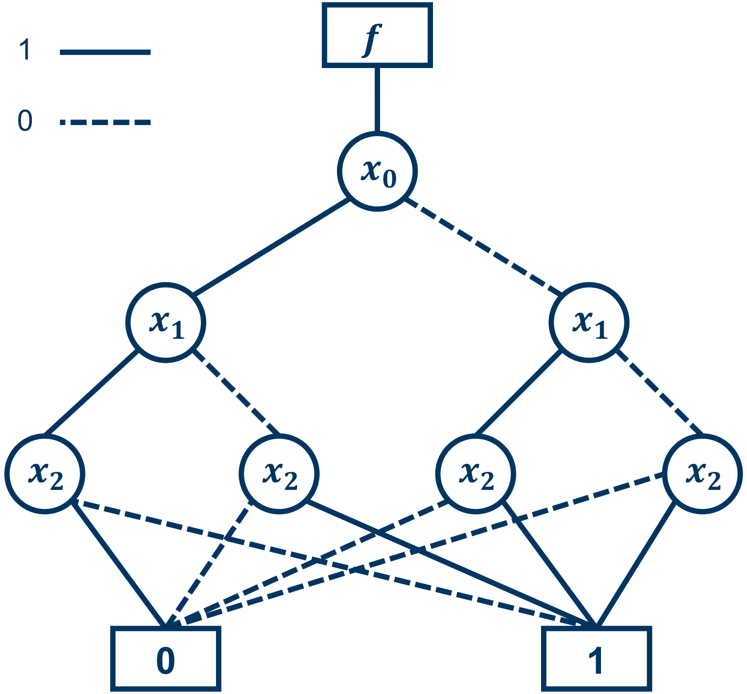
Verification Concept



Binary Decision Diagrams (BDDs)

Advantages using BDDs

- Proposed for testing VLSI circuits
- Symbolic simulation
- Boolean operations are elementary operations
- Efficiently determine number of satisfying assignments
- Checking statistical dependencies
- Comparing golden and faulty circuit models

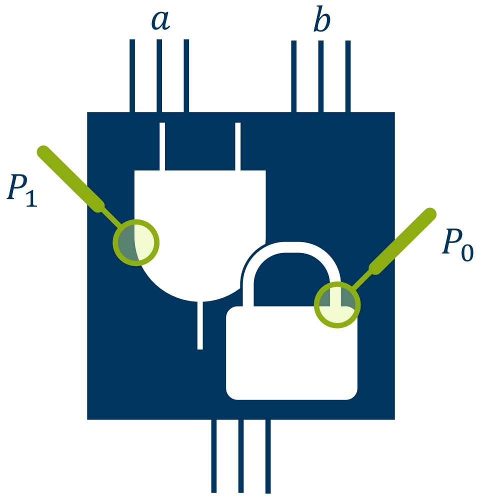


BDD of the function f

Verification Principles for d-Probing [KSM20]

d-Probing Security. A circuit C with secret input set $\boldsymbol{X} \in \mathbb{F}_2^n$ is d-probing secure, if and only if for any observation set $\boldsymbol{Q}$ containing d wires, $\boldsymbol{X}$ is statistically independent of the observation set, i.e., the following condition holds:

$$\Pr[\boldsymbol{Q}|\boldsymbol{X}] = \Pr[\boldsymbol{Q}].$$

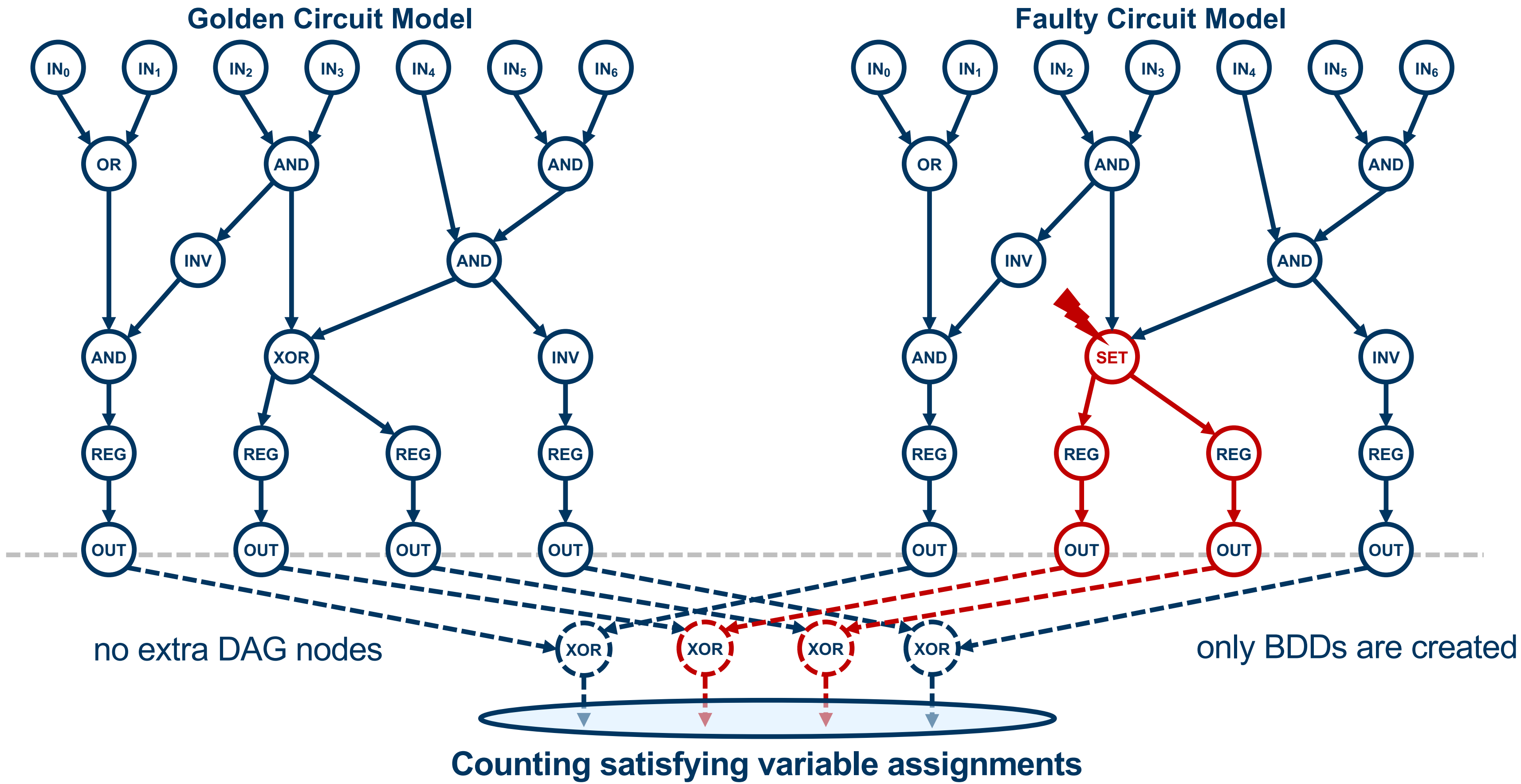


$\boldsymbol{Q} = \{P_0, P_1\}$

$\boldsymbol{X}' \subseteq \boldsymbol{X}$ **i.e.,** $\boldsymbol{X}' \in \{\{a\}, \{b\}, \{a, b\}\}$

$p_{\boldsymbol{Q},\boldsymbol{X}'}(\mathbf{1},\mathbf{1}) \neq p_{\boldsymbol{Q}}(\mathbf{1}) \cdot p_{\boldsymbol{X}'}(\mathbf{1}) \rightarrow$ insecure

Checking Fault Security on BDDs



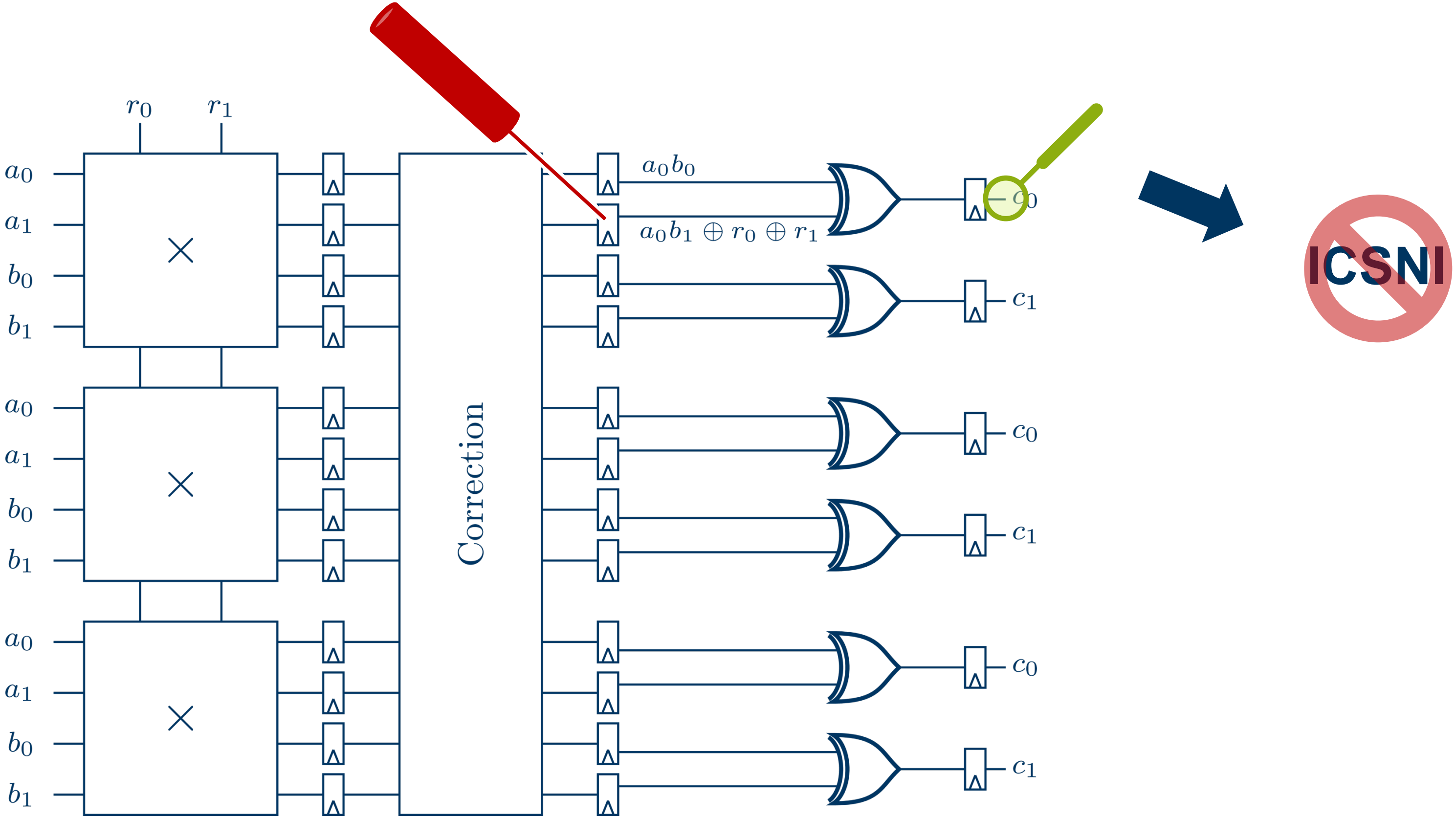
Verification of Countermeasures against Fault Injections [RBRSS+21]

Single round of CRAFT protected by linear error correcting codes

$$t = \tau_{bf} \qquad l = \mathrm{mc}_{\infty}$$

	1-bit Protection	2-bit Protection	3-bit Protection
	925	1 490	1 807
$\binom{n}{k}$	766	329 730	91 737 144
	0.021 s	1.496 s	2 937 s

Verification of Combined Gadgets [RFSG22]



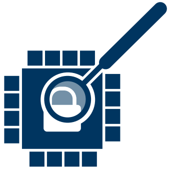
The gadget has been originally proposed in [DN20] and should be $(1{,}1)$-ICSNI.

VERICA – Verification of Combined Attacks

RUB

Features

Limitations

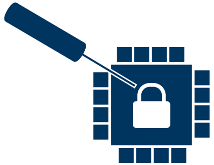


Side-Channel Security

d-probing model
Glitch-extended d-probing

Side-Channel Composability

PNI
PSNI
PINI

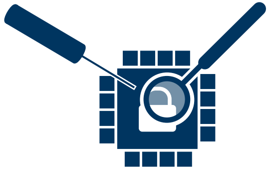


Fault Security

Detection strategy
Correction strategy
SIFA strategy

Fault Composability

FNI
FSNI
FINI



Combined Security

(d, k)-combined security

Combined Composability

CNI
CSNI
ICSNI
CINI
ICINI

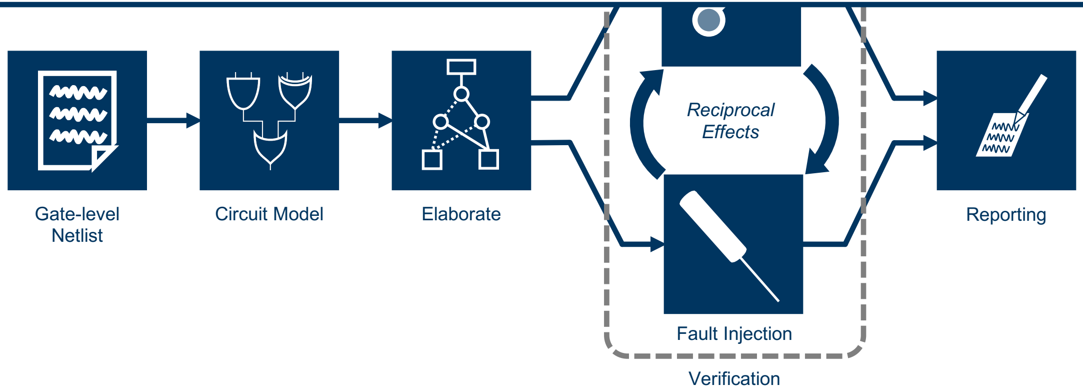
Performing exhaustive analyses - exact verification
for generic unrolled hardware circuits

- Only unrolled circuits
- No transitions nor couplings
- Higher-order verifications

Summary

Code and paper are publicly available

https://github.com/Chair-for-Security-Engineering/VERICA



Modeling of Physical Attacks

Verification of Countermeasures against Physical Attacks

RUB

Thank you!

jan.richter-brockmann@rub.de

SECURITY ENGINEERING

References

RUB

[BBD+15] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, and Pierre-Yves Strub. *Verified Proofs of Higher-Order Masking.* In EUROCRYPT, pages 457–485, 2015.

[BBD+16] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. *Strong Non-Interference and Type-Directed Higher-Order Masking.* In SIGSAC, pages 116–129, 2016.

[CS20] Gaetan Cassiers and Fran¸cois-Xavier Standaert. *Trivially and Efficiently Composing Masked Gadgets With Probe Isolating Non-Interference.* IEEE Trans. Inf. Forensics Secur., 15:2542–2555, 2020.

[DDE+20] Joan Daemen, Christoph Dobraunig, Maria Eichlseder, Hannes Groß, Florian Mendel, and Robert Primas. *Protecting against Statistical Ineffective Fault Attacks.* IACR Trans. Cryptogr. Hardw. Embed. Syst., 2020(3):508–543, 2020.

[DN20] Siemen Dhooghe and Svetla Nikova. *My Gadget Just Cares for Me – How NINA Can Prove Security Against Combined Attacks.* In CT-RSA, volume 12006 of Lecture Notes in Computer Science, pages 35–55. Springer, 2020.

[FGP+18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and Fran¸cois-Xavier Standaert. *Composable Masking Schemes in the Presence of Physical Defaults & the Robust Probing Model.* IACR Trans. Cryptogr. Hardw. Embed. Syst. , 2018(3):89–120, 2018.

[FRSG22] Jakob Feldtkeller, Jan Richter-Brockmann, Pascal Sasdrich, and Tim Güneysu. *CINI MINIS: Domain Isolation for Fault and Combined Security.* CCS, 2022

[HPB21] Vedad Hadzic, Robert Primas, and Roderick Bloem. *Proving SIFA protection of masked redundant circuits.* In Automated Technology for Verification and Analysis, volume 12971 of Lecture Notes in Computer Science, pages 249–265. Springer, 2021.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. *Private Circuits: Securing Hardware against Probing Attacks.* In Dan Boneh, editor, CRYPTO, volume 2729 of Lecture Notes in Computer Science, pages 463–481. Springer, 2003.

[KSM20] David Knichel, Pascal Sasdrich, and Amir Moradi. *SILVER – Statistical Independence and Leakage Verification.* In ASIACRYPT, volume 12491 of Lecture Notes in Computer Science, pages 787–816. Springer, 2020.

[RBRSS+21] Jan Richter-Brockmann, Aein Rezaei Shahmirzadi, Pascal Sasdrich, Amir Moradi, and Tim Güneysu. *FIVER – Robust Verification of Countermeasures against Fault Injections.* IACR Trans. Cryptogr. Hardw. Embed. Syst., 2021(4):447–473, Aug. 2021.

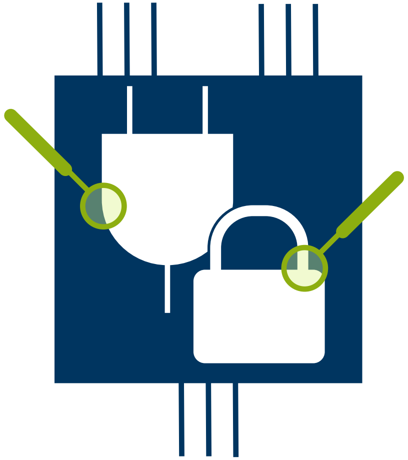
[RBSG21] Jan Richter-Brockmann, Pascal Sasdrich, and Tim Güneysu. *Revisiting Fault Adversary Models - Hardware Faults in Theory and Practice.* Trans. On Computers, 2022

[RFSG22] Jan Richter-Brockmann, Jakob Feldtkeller, Pascal Sasdrich, and Tim G¨uneysu. VERICA - Verifi cation of Combined Attacks: Automated formal verification of security against simultaneous information leakage and tampering. IACR Trans. Cryptogr. Hardw. Embed. Syst. , 2022(4), 2022.

[SMG16] Tobias Schneider, Amir Moradi, and Tim Güneysu. *ParTI - Towards Combined Hardware Countermeasures Against Side-Channel and Fault-Injection Attacks.* In CRYPTO 2016, volume 9815 of Lecture Notes in Computer Science, pages 302–332. Springer, 2016.

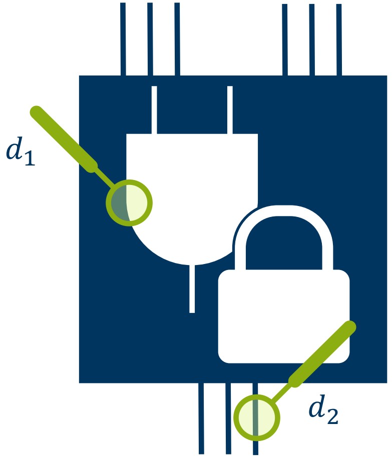
Modeling Side-Channel Attacks – Composability

PNI [BBD+15]
Probe Non-Interference



$d' \leq d$

PSNI [BBD+16]
Probe Strong Non-Interference

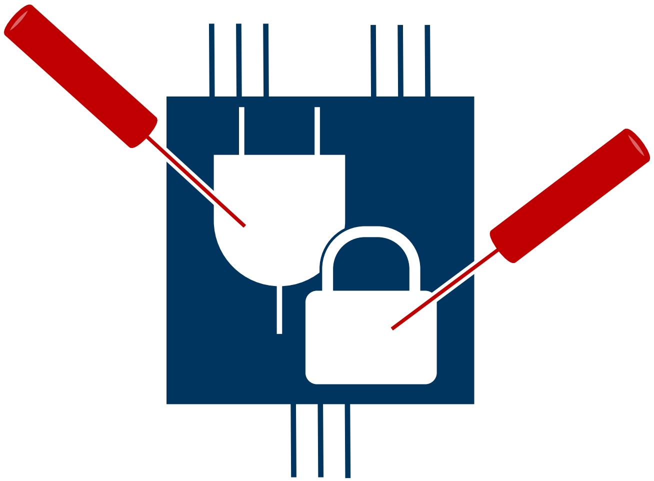


$d_1 + d_2 \leq d$

Modeling Fault-Injection Attacks – Composability

FNI [DN20]

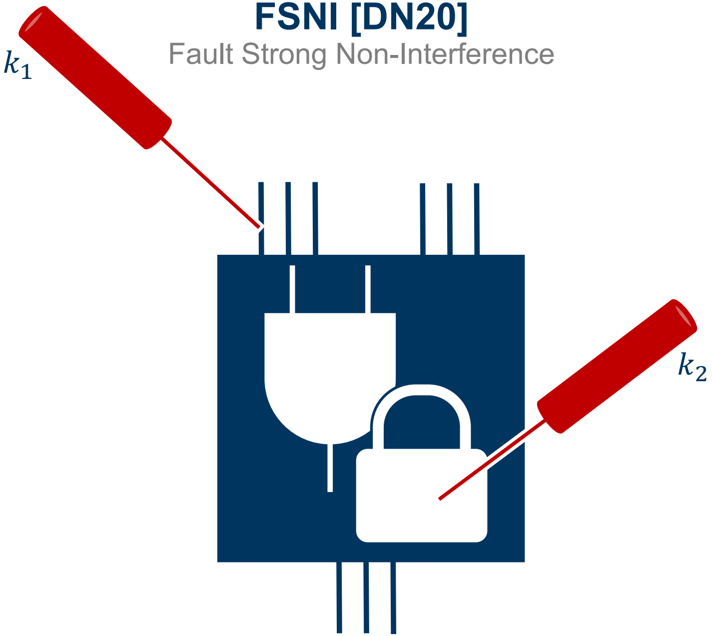
Fault Non-Interference



$k' \leq k$

FSNI [DN20]

Fault Strong Non-Interference

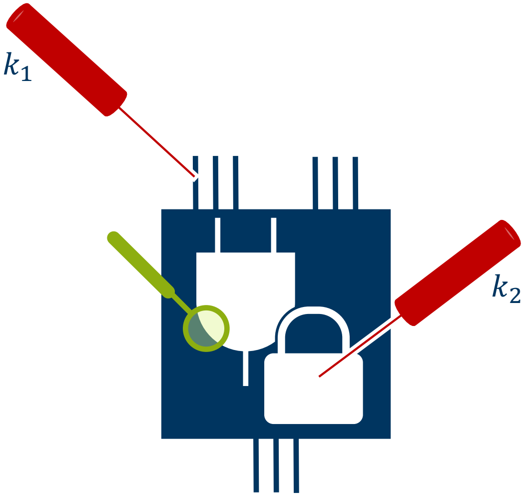


$k_1 + k_2 \leq k$

Modeling Combined Attacks – Composability

CNI [DN20]

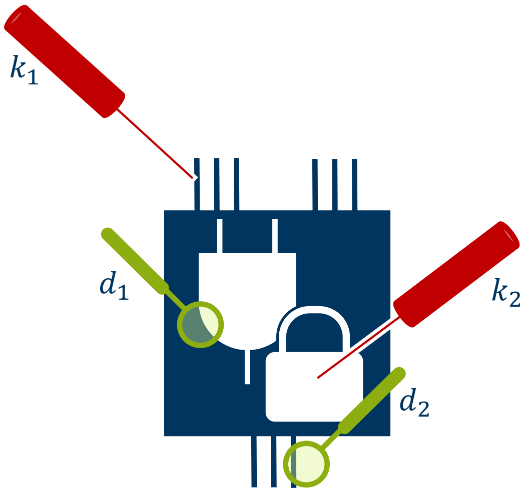
Combined Non-Interference



$$d' + k_1 + k_2 \leq d$$
$$k_1 + k_2 \leq k$$

CSNI [DN20]

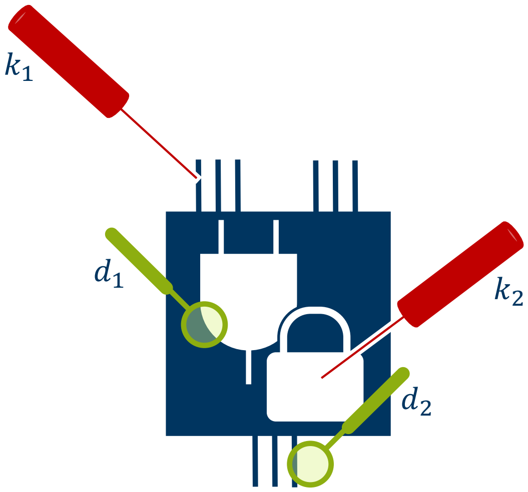
Combined Strong Non-Interference



$$d_1 + d_2 + k_1 + k_2 \leq d$$
$$k_1 + k_2 \leq k$$

ICSNI [DN20]

Independent Combined Strong Non-Interference

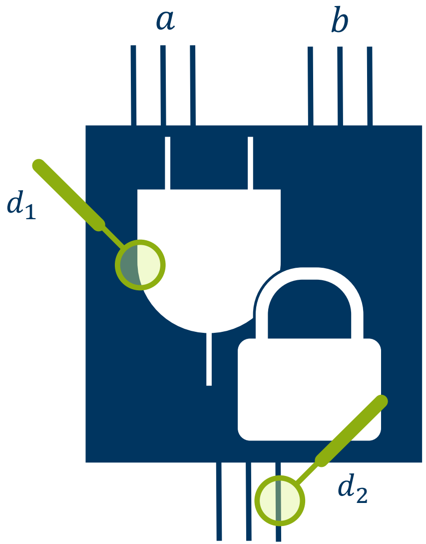


$$d_1 + d_2 \leq d$$
$$k_1 + k_2 \leq k$$

Modeling Side-Channel Attacks – Composability

PINI [CS20]

Probe-Isolating Non-Interference

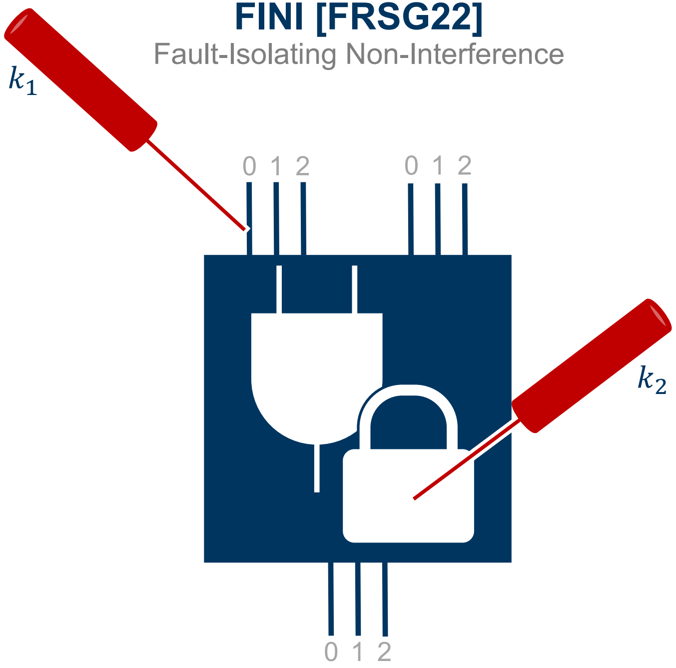


$d_1 + d_2 \leq d$

Modeling Fault-Injection Attacks – Composability

FINI [FRSG22]

Fault-Isolating Non-Interference

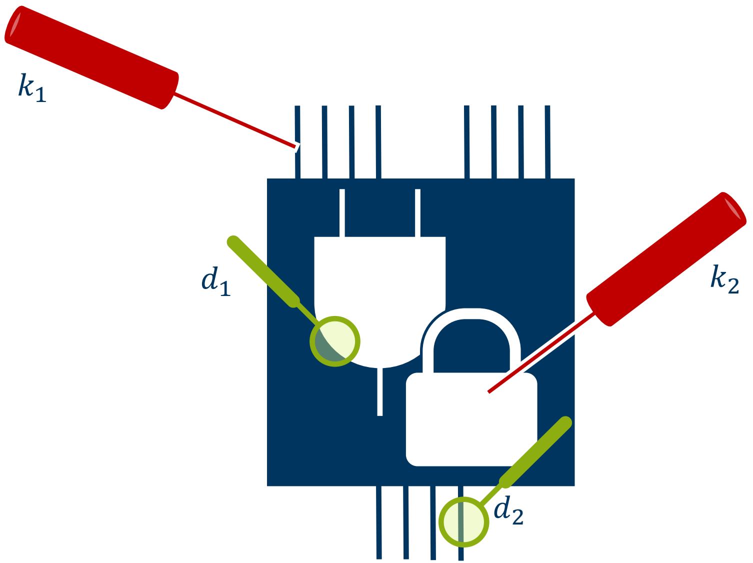


$$k_1 + k_2 \leq k$$

Modeling Combinded Attacks – Composability

CINI [FRSG22]

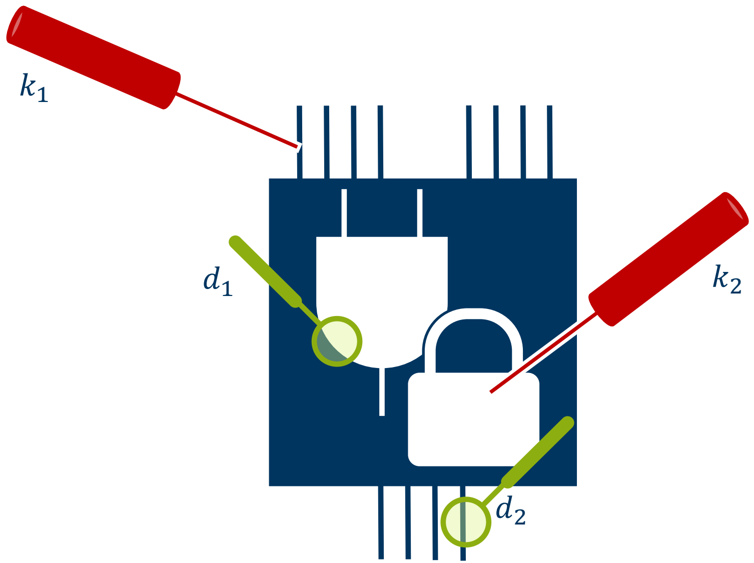
Combined-Isolating Non-Interference



$$d_1 + d_2 + k_1 + k_2 \leq d$$
$$k_1 + k_2 \leq k$$

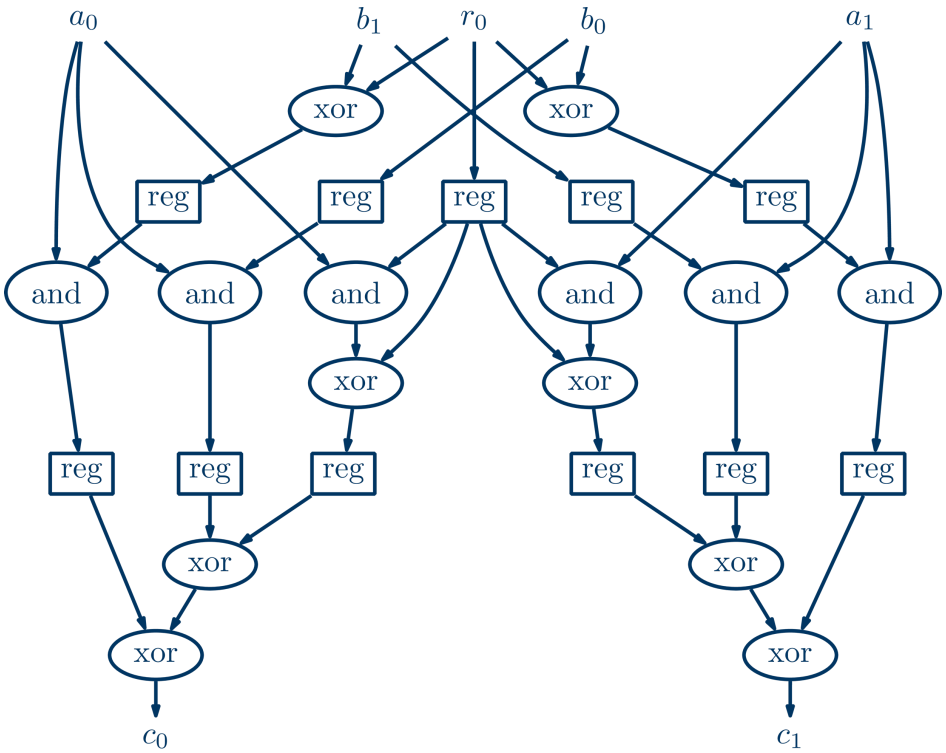
ICINI [FRSG22]

Independent Combined-Isolating Non-Interference

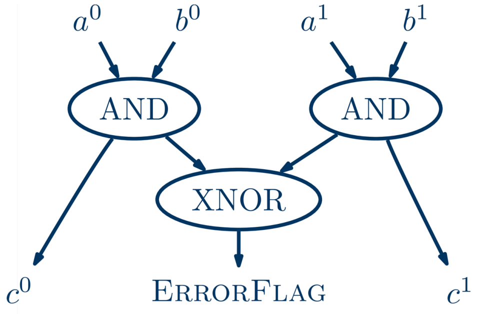


$$d_1 + d_2 + k_1 + k_2 \leq d$$
$$k_1 + k_2 \leq k$$

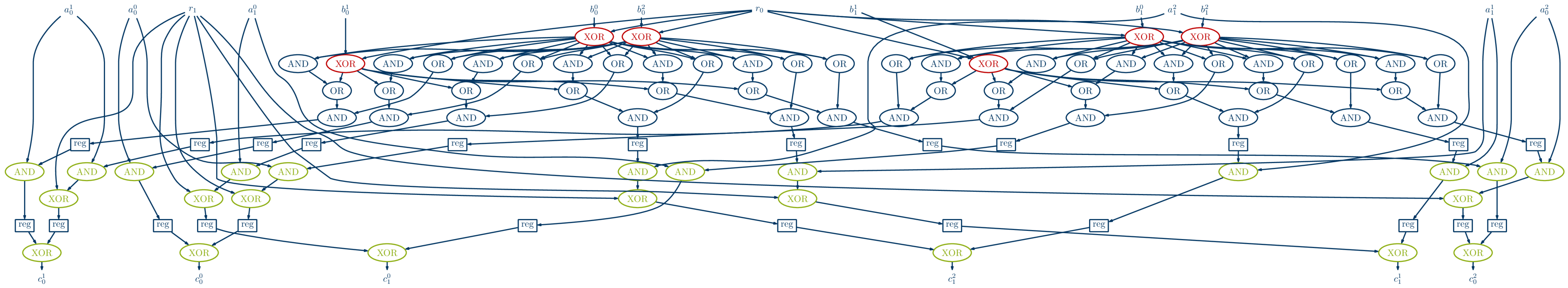
PINI Gadget (HPC2)



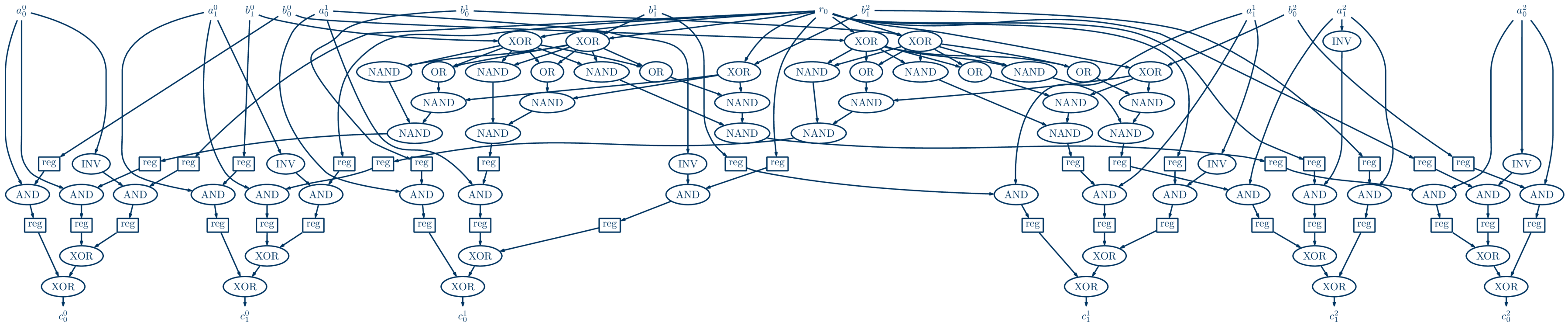
FINI Gadget



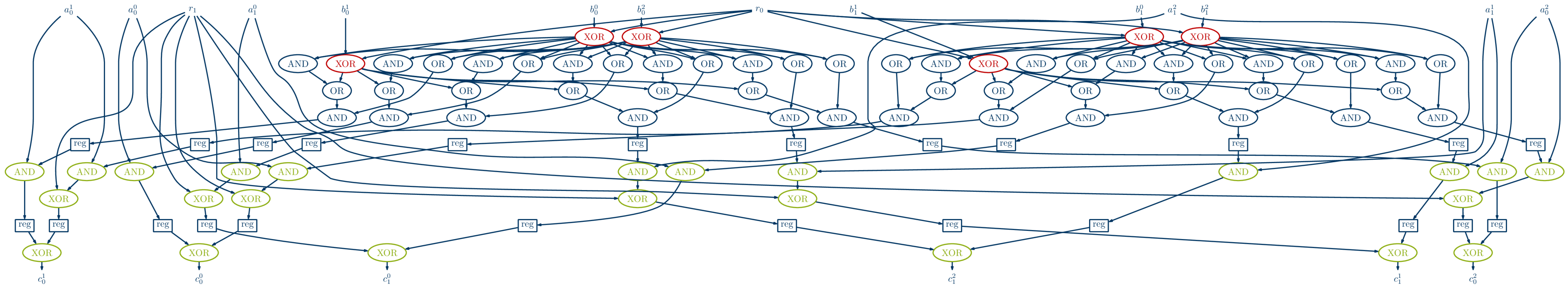
CINI Gadget – $\mathbf{HPC}_1^{\mathbf{c}}$



CINI Gadget – $\mathbf{HPC}_2^{\mathbf{c}}$



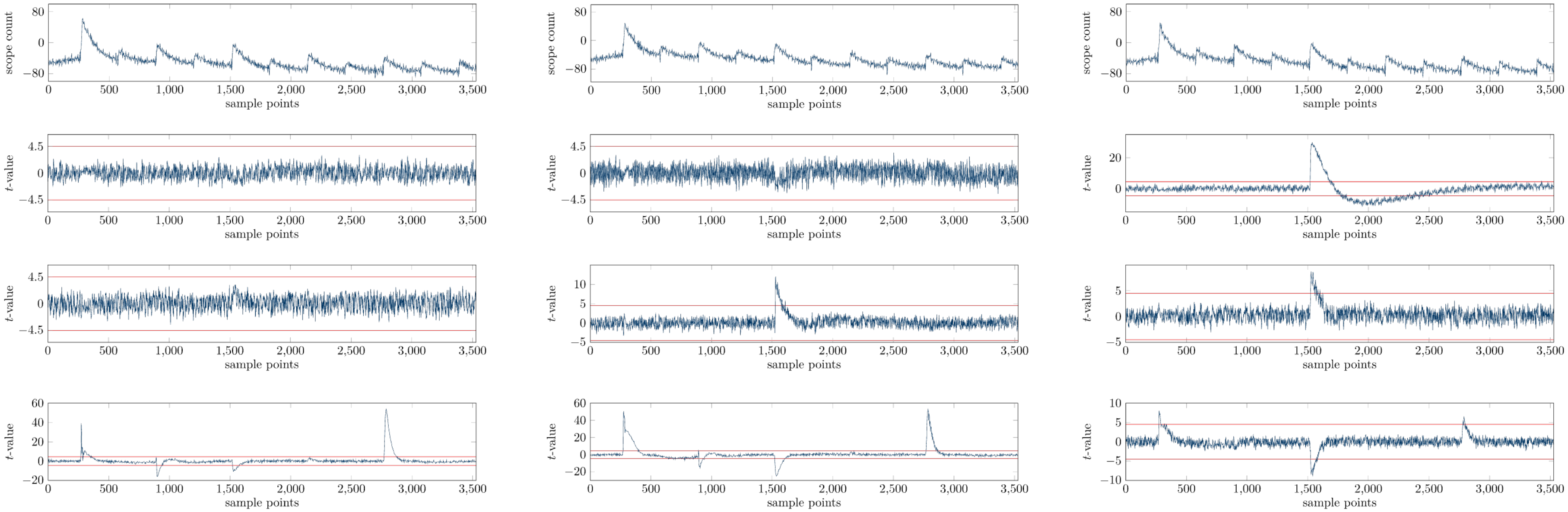
ICINI Gadget - $\mathrm{HPC}_1^{\mathrm{I}}$



Practical Validation

RUB

PRESENT S-box protected by $(2,2)$-CINI Gadgets



fault free – 100 million traces

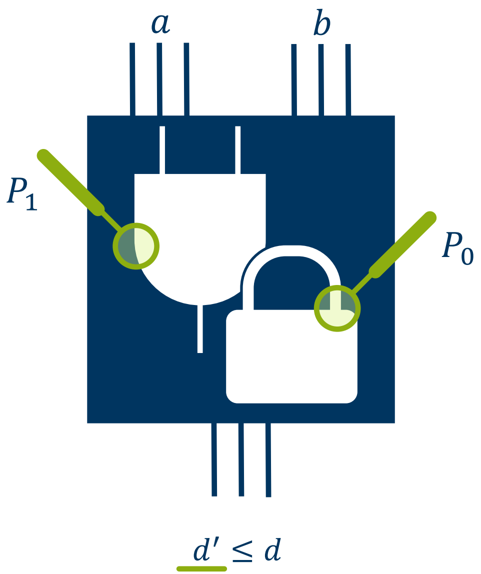
one fault – 100 million traces

two faults – 1 million traces

Verification Principles in SILVER [KSM20]

d-Non-Interference. A circuit C with secret input set $\boldsymbol{X} \in \mathbb{F}_2^n$ provides d-Non-Interference if and only if for any observation set of $d' \leq d$ wires $\boldsymbol{Q}$ there exists a set $\boldsymbol{S}$ of input shares with $|\boldsymbol{S}|_{\forall i} \leq d'$ such that

$$\Pr[\boldsymbol{Q}|\boldsymbol{S}] = \Pr[\boldsymbol{Q}|Sh(\boldsymbol{X})].$$



$\boldsymbol{Q} = \{P_0, P_1\}$

$\boldsymbol{S} = \{a_0, a_2, b_1, b_2\}$

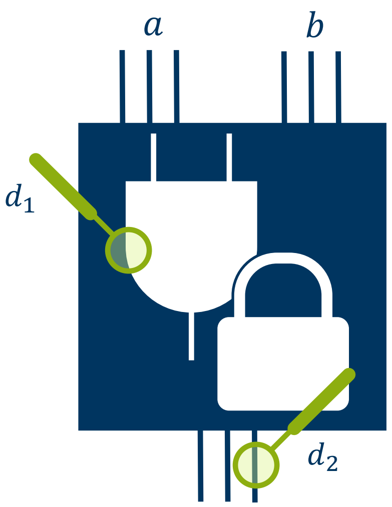
$Sh(\boldsymbol{X}) = Sh(a, b)$

Verification Principles in SILVER [KSM20]

RUB

$\boldsymbol{d}$-Strong Non-Interference. A circuit C with secret input set $\boldsymbol{X} \in \mathbb{F}_2^n$ provides d-Strong Non-Interference if and only if for any observation set of $d' = d_1 + d_2 \leq d$ wires $\boldsymbol{Q}$ of which d_1 are internal wires and d_2 are output wires, there exists a simulation set $\boldsymbol{S}$ of input shares with $|\boldsymbol{S}|_{\forall i} \leq d_1$ such that

$$\Pr[\boldsymbol{Q}|\boldsymbol{S}] = \Pr[\boldsymbol{Q}|Sh(\boldsymbol{X})].$$



$$d_1 + d_2 \leq d$$

$$\boldsymbol{Q} = \{P_0, P_1\}$$

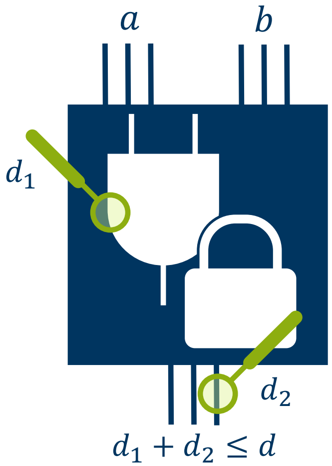
$$\boldsymbol{S} = \{a_0, b_1\}$$

$$Sh(\boldsymbol{X}) = Sh(a, b)$$

Verification Principles in SILVER [KSM20]

d-Probe-Isolating Non-Interference. Let $\boldsymbol{P}$ be the set of internal probes with $|\boldsymbol{P}| = d_1$. Let further $\boldsymbol{I_O}$ be the index set assigned to the probed output wires $\boldsymbol{O}$ with $|\boldsymbol{I_O}| = d_2$.
A circuit C with secret input set $\boldsymbol{X} \in \mathbb{F}_2^n$ provides d-Probe-Isolating Non-Interference if and only if for every $\boldsymbol{P}$ and $\boldsymbol{O}$ with $d_1 + d_2 \leq d$ there exists a set $\boldsymbol{I_I}$ of circuit indices with $|\boldsymbol{I_I}| \leq d_1$ such that $\boldsymbol{Q} = \boldsymbol{P} \cup \boldsymbol{O}$ can be perfectly simulated by $\boldsymbol{S} = Sh(\boldsymbol{X})^{\boldsymbol{I_I} \cup \boldsymbol{I_O}}$, i.e., it holds that

$$\Pr[\boldsymbol{Q}|\boldsymbol{S}] = \Pr[\boldsymbol{Q}|Sh(\boldsymbol{X})].$$



$$\boldsymbol{Q} = \{P_0, P_1\}$$

$$\boldsymbol{S} = \{a_0, a_2, b_0, b_2\}$$

$$Sh(\boldsymbol{X}) = Sh(a, b)$$

Complexity Reduction in FIVER

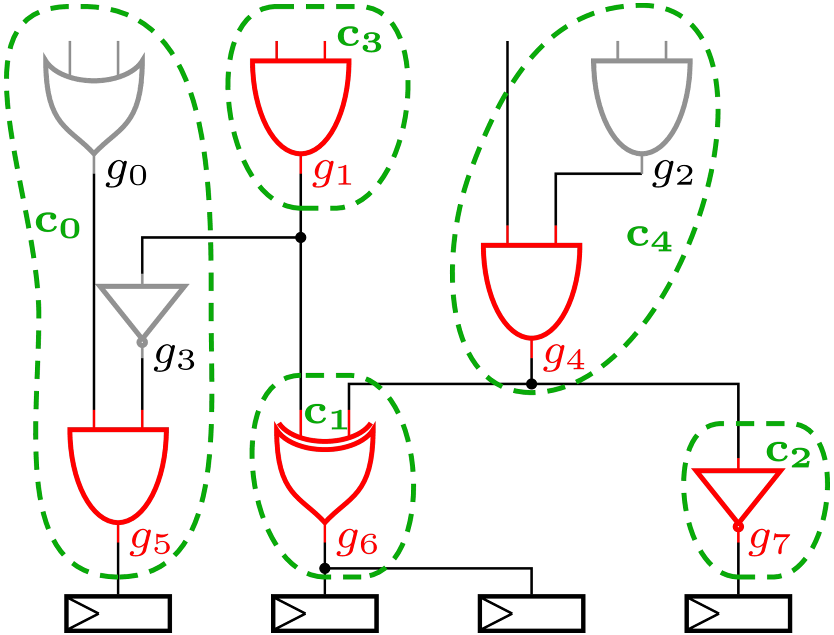
RUB

Algorithm 10 Complexity Reduction.

Require: Golden circuit model $\mathbf{D}$, set of valid fault location (nodes) Λ
Ensure: Set of reduced fault locations Λ_{red}

$\Sigma \leftarrow \emptyset$, $\Lambda_{\text{red}} \leftarrow \emptyset$
for $\forall d \in \mathbf{D}$ **do**
  if type $(d) =$ reg **or** type $(d) =$ out **then**
    $\Sigma \leftarrow \Sigma \cup d$
    if $d \in \Lambda$ **then**
      $\Lambda_{\text{red}} \leftarrow \Lambda_{\text{red}} \cup d$
    end if
  end if
end for
for $\sigma \in \Sigma$ **do**
  $\Lambda_{\text{red}} \leftarrow \Lambda_{\text{red}} \cup \texttt{node\_in}(\sigma)$
  $\Phi \leftarrow \sigma$

  while $\Phi \neq \emptyset$ **do**
    $\alpha \leftarrow \Phi[0]$, $\texttt{delete}(\Phi[0])$
    for $\forall n \in \texttt{node\_in}(\sigma)$ **do**
      if type $(n) \neq$ reg **and** type $(n) \neq$ in **then**
        $\Phi \leftarrow \Phi \cup n$
      end if
      if $\texttt{out\_degree}(\alpha) > 1$ **and** $\alpha \in \Lambda$ **and** $\alpha \notin \Lambda_{\text{red}}$ **then**
        $\Lambda_{\text{red}} \leftarrow \Lambda_{\text{red}} \cup \alpha$
      end if
    end for
  end while
end for



Case Studies from FIVER

Redundancy (Capability*) [bits]	Verification Parameter			Design Properties			Analysis Results		
	$\zeta(n,t,l)$	Variate	Complexity Reduction	Comb. Gates	Seq. Gates	Logic Stages	Combinations	Time [s]	Security
CRAFT – 1 round (detection)									
1 (1)	$\zeta(1, \tau_{bf}, cs)$	univariate	no	845	80	2	766	0.021	✓
1 (1)	$\zeta(2, \tau_{bf}, cs)$	univariate	no	845	80	2	151 561	0.769	✗
3 (2)	$\zeta(2, \tau_{bf}, cs)$	univariate	no	1 410	112	2	329 730	1.496	✓
3 (2)	$\zeta(3, \tau_{bf}, cs)$	univariate	no	1 410	112	2	64 320 469	441	✗
4 (3)	$\zeta(3, \tau_{bf}, cs)$	univariate	no	1 679	128	2	91 737 144	2 937	✓
			yes	1 679	128	2	4 665 200	360	✓
CRAFT – 2 rounds (detection)									
1 (1)	$\zeta(1, \tau_{bf}, cs)$	univariate	no	1 571	160	3	1 491	0.378	✓
1 (1)	$\zeta(2, \tau_{bf}, cs)$	univariate	no	1 571	160	3	417 882	62	✗
3 (2)	$\zeta(2, \tau_{bf}, cs)$	univariate	no	2 526	224	3	868 500	157	✓
3 (2)	$\zeta(3, \tau_{bf}, cs)$	univariate	no	2 526	224	3	250 984 950	∞	–
			yes	2 526	224	3	7 364 279	408	✗
CRAFT – 2 rounds – multivariate (detection)									
1 (1)	$\zeta(1, \tau_{bf}, cs)$	bivariate	no	1 720	160	3	682 832	140	✓
1 (1)	$\zeta(1, \tau_{bf}, cs)$	trivariate	yes	1 720	160	3	99 542 528	26 955	✓
3 (2)	$\zeta(2, \tau_{sr}, s)$	bivariate	no	2 915	224	3	38 651 200	81 897	✓
CRAFT – 1 round (correction)									
3 (1)	$\zeta(1, \tau_{bf}, cs)$	univariate	no	2 868	112	2	2 788	0.081	✓
3 (1)	$\zeta(2, \tau_{bf}, cs)$	univariate	no	2 868	112	2	3 201 690	22	✗
7 (2)	$\zeta(2, \tau_{bf}, cs)$	univariate	no	17 460	176	2	129 651 034	3 543	✓
			yes	17 460	176	2	10 923 888	130	✓
LED-64 – 1 round (detection)									
1 (1)	$\zeta(1, \tau_{bf}, cs)$	univariate	no	1 541	0	1	1 301	0.064	✓
1 (1)	$\zeta(2, \tau_{bf}, cs)$	univariate	no	1 541	0	1	846 951	9.558	✗
3 (2)	$\zeta(2, \tau_{bf}, cs)$	univariate	no	2 435	0	1	1 730 730	27	✓
3 (2)	$\zeta(3, \tau_{bf}, cs)$	univariate	no	2 435	0	1	1 072 477 550	12 722	✗
4 (3)	$\zeta(3, \tau_{bf}, cs)$	univariate	no	2 916	0	1	1 654 087 449	17 348	✓
			yes	2 916	0	1	3 983 413	94	✓
AES-128 – 1 round (detection)									
1 (1)	$\zeta(1, \tau_{bf}, cs)$	univariate	no	24 864	0	1	24 432	22	✓
4 (2)	$\zeta(2, \tau_{bf}, cs)$	univariate	no	34 159	0	1	298 473 528	∞	–
			yes	34 159	0	1	56 632 584	471 281	✓

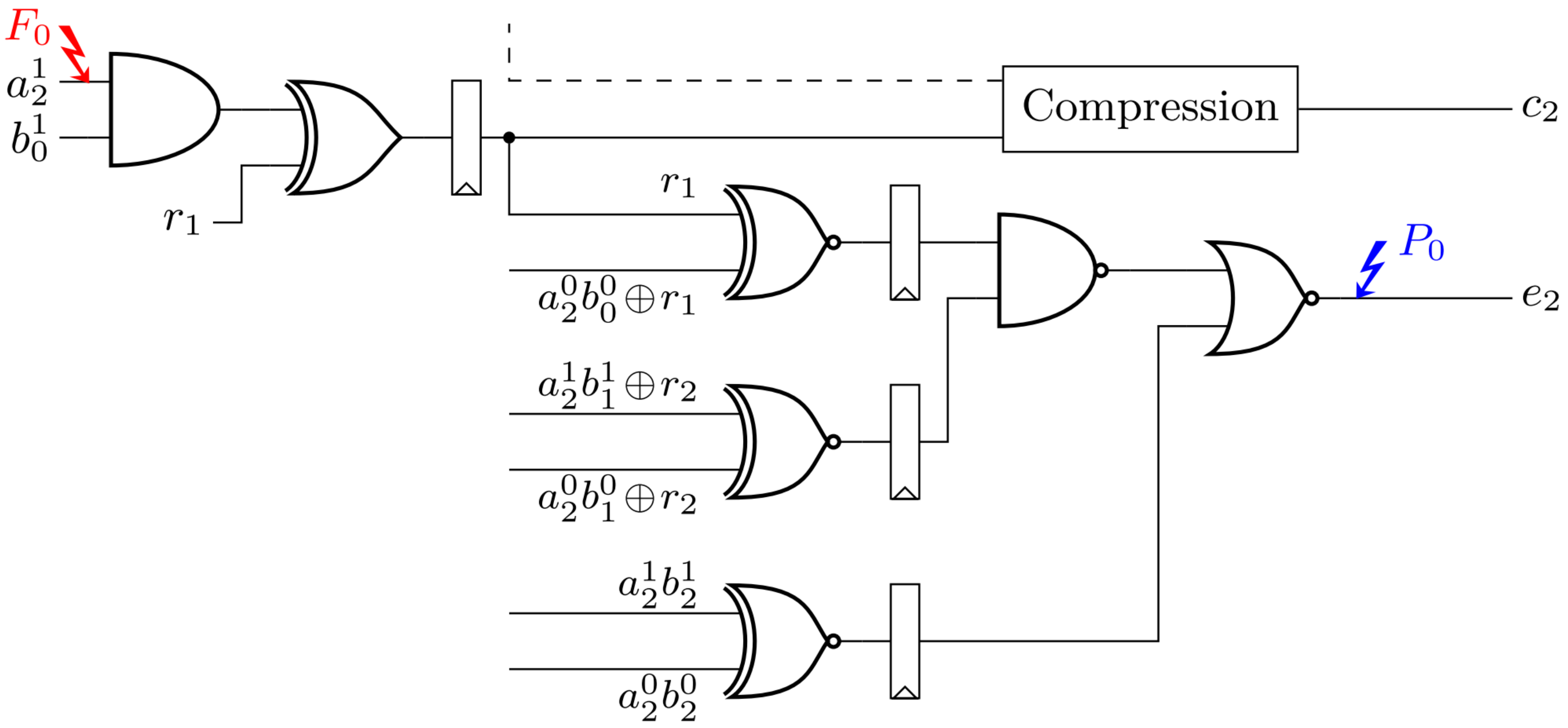
* The capability determines the maximum number of faults that can be detected or corrected by the corresponding countermeasure.

Case Studies from VERICA – Combined Gadgets

Gadget	Design					SCA			FIA			Combined		
	d	k	rand.	comb.	memory	PNI	PSNI	Time	FNI	FSNI	Time		(d, k)	Time
NINA	1	1	0	4	0	$1^{\checkmark}$	–	0.460 s	$1^{\checkmark}$	–	0.429 s	CNI	$(1,1)^{\checkmark}$	0.430 s
NINA	1	2	0	6	0	$1^{\checkmark}$	–	0.455 s	$2^{\checkmark}$	–	0.445 s	CNI	$(1,2)^{\checkmark}$	0.492 s
NINA	2	1	0	6	0	$2^{\checkmark}$	–	0.471 s	$1^{\checkmark}$	–	0.451 s	CNI	$(2,1)^{\checkmark}$	0.436 s
NINA	2	2	0	9	0	$2^{\checkmark}$	–	0.442 s	$2^{\checkmark}$	–	0.444 s	CNI	$(2,2)^{\checkmark}$	0.442 s
SNINA	1	1	1	22	16	–	$1^{\checkmark}$	0.476 s	–	$1^{\checkmark}$	0.449 s	CSNI	$(1,1)^{\checkmark}$	0.473 s
SNINA	1	2	1	38	26	–	$1^{\checkmark}$	0.451 s	–	$2^{\checkmark}$	0.500 s	CSNI	$(1,2)^{\checkmark}$	0.519 s
SNINA	2	1	3	57	33	–	$2^{\checkmark}$	0.566 s	–	$1^{\checkmark}$	0.456 s	CSNI	$(2,1)^{\times}/(1,1)^{\checkmark}$	0.592 s
SNINA	2	2	3	96	54	–	$2^{\checkmark}$	0.821 s	–	$2^{\checkmark}$	0.673 s	CSNI	$(2,2)^{\times}/(1,1)^{\checkmark}$	1.062 s
SININA	1	1	2	90	30	–	$1^{\checkmark}$	0.450 s	–	$1^{\checkmark}$	0.461 s	ICSNI	$(1,1)^{\times}/(0,0)^{\checkmark}$	0.456 s
SININA	1	2	3	360	50	–	$1^{\checkmark}$	0.555 s	–	$2^{\checkmark}$	1.395 s	ICSNI	$(1,2)^{\times}/(0,0)^{\checkmark}$	17.985 s
SININA	2	1	6	207	63	–	$2^{\checkmark}$	1.334 s	–	$1^{\checkmark}$	0.511 s	ICSNI	$(2,1)^{\times}/(0,0)^{\checkmark}$	73.574 s
SININA*	2	2	9	825	105	–	$2^{\checkmark}$	76.030 s	–	$2^{\checkmark}$	5.300 s	ICSNI	$(2,2)^{\times}/(0,0)^{\checkmark}$	>2.7 h

* Due to the high verification complexity, we interrupted the combined analysis after testing $(2,1)$-SININA where VERICA already reported a failure.

Case Studies from VERICA – SNINA Flaw



Case Studies from VERICA – ParTI

RUB

Implementation	Design		$\zeta(0, \tau_{sr}, \mathrm{mc}_\infty)$		$\zeta(1, \tau_{sr}, \mathrm{mc}_\infty)$	
	comb.	memory	Det./Corr.	Prob.	Det./Corr.	Prob.
ParTI S-box (Detection)	678	78	–	1✓ [0.866 s]	1✓ [1.010 s]	0✗ [1.950 s]
ParTI S-box (Correction)	2063	72	–	1✓ [4.103 s]	1✓ [3.677 s]	0✗ [336.239 s]

Case Studies from VERICA – SIFA

Implementation	Design comb.	Design mem.	$\zeta(0, \tau_{sr}, \mathrm{mc}_\infty)$ SIFA	$\zeta(0, \tau_{sr}, \mathrm{mc}_\infty)$ Prob.	$\zeta(1, \tau_{sr}, \mathrm{mc}_\infty)$ SIFA	$\zeta(1, \tau_{sr}, \mathrm{mc}_\infty)$ Prob.	$\zeta(2, \tau_{sr}, \mathrm{mc}_\infty)$ SIFA	$\zeta(2, \tau_{sr}, \mathrm{mc}_\infty)$ Prob.
p_{TS}	8	6	–	1✓[0.47 s]	1✓[0.45 s]	1✓[0.45 s]	1✗[0.46 s]	1✓[0.44 s]
$p_{\chi S}$	10	6	–	1✓[0.45 s]	1✓[0.44 s]	1✓[0.45 s]	1✗[0.46 s]	1✓[0.45 s]
χ_3	30	30	–	1✓[0.43 s]	1✓[0.46 s]	0✗[0.46 s]	1✗[0.46 s]	0✗[0.49 s]
χ_5	52	42	–	1✓[0.44 s]	1✓[0.48 s]	0✗[0.44 s]	1✗[0.48 s]	0✗[0.54 s]
AES S-box, g_{104} [HPB21]	631	0	–	0✗[13.80 s]	0✗[194.89 s]	0✗[191.93 s]	[∞]	[∞]
AES S-box, full [HPB21]	634	0	–	0✗[13.90 s]	1✓[194.58 s]	0✗[194.70 s]	[∞]	[∞]

Case Studies from CINI – Gadgets

Gadget	Design						Verification		
	d	k	rand.	comb.	reg.	area [GE]	Def.	(d,k)	Time
Detect	–	1	0	3	0	4.7	FINI	$(0,1)^{\checkmark}$	0.387 s
	–	2	0	6	0	9		$(0,2)^{\checkmark}$	0.397 s
	–	3	0	13	0	15		$(0,3)^{\checkmark}$	0.429 s
	–	4	0	18	0	19.7		$(0,4)^{\checkmark}$	1.280 s
Correct	–	1	0	15	0	17	FINI	$(0,1)^{\checkmark}$	0.383 s
	–	2	0	75	0	98.3		$(0,2)^{\checkmark}$	0.445 s
	–	3	0	147	0	194.3		$(0,3)^{\checkmark}$	16.501 s
	–	4	0	297	0	390		$(0,4)^{\checkmark}$	6.24 h
$\mathsf{HPC}_1^{\mathsf{C}}$	1	1	2	78	24	238	CINI	$(1,1)^{\checkmark}$	0.409 s
	2	1	6	189	54	567		$(2,1)^{\checkmark}$	0.485 s
	3	1	12	356	96	1 032		$(3,1)^{\checkmark}$	39.544 s
	1	2	2	340	40	685		$(1,2)^{\checkmark}$	1.490 s
	2	2	6	795	90	1 595		$(2,2)^{\checkmark}$	6.321 s
	3	2	12	1420	160	2 860		$(3,2)^{\checkmark}$	4.662 min
	1	3	2	590	56	1 087		$(1,3)^{\checkmark}$	16.817 min
	2	3	6	1362	126	2 502		$(2,3)^{\checkmark}$	3.897 h
	3	3	12	2456	224	4 509		*	∞
$\mathsf{HPC}_2^{\mathsf{C}}$	1	1	1	66	36	294	CINI	$(1,1)^{\checkmark}$	0.389 s
	2	1	3	189	90	768		$(2,1)^{\checkmark}$	0.775 s
	1	2	1	210	60	640		$(1,2)^{\checkmark}$	0.804 s
	2	2	3	615	150	1 730		$(2,2)^{\checkmark}$	5.643 s
	3	1	6	372	168	1 460		$(3,1)^{\times}/(2,1)^{\checkmark}$	18.386 h
$\mathsf{HPC}_1^{\mathsf{I}}$	1	1	2	78	24	240	ICINI	$(1,1)^{\checkmark}$	0.397 s
	2	1	6	189	54	573		$(2,1)^{\checkmark}$	4.329 s
	3	1	12	356	96	1 044		*	∞
	1	2	4	360	40	728		$(1,2)^{\checkmark}$	7.153 s
	2	2	12	855	90	1 725		*	∞
	3	2	24	1540	160	3 120		*	∞
	1	3	6	646	56	1 203		$(1,3)^{\checkmark}$	4.743 h
	2	3	18	1530	126	2 852		*	∞
	3	3	36	2792	224	5 209		*	∞

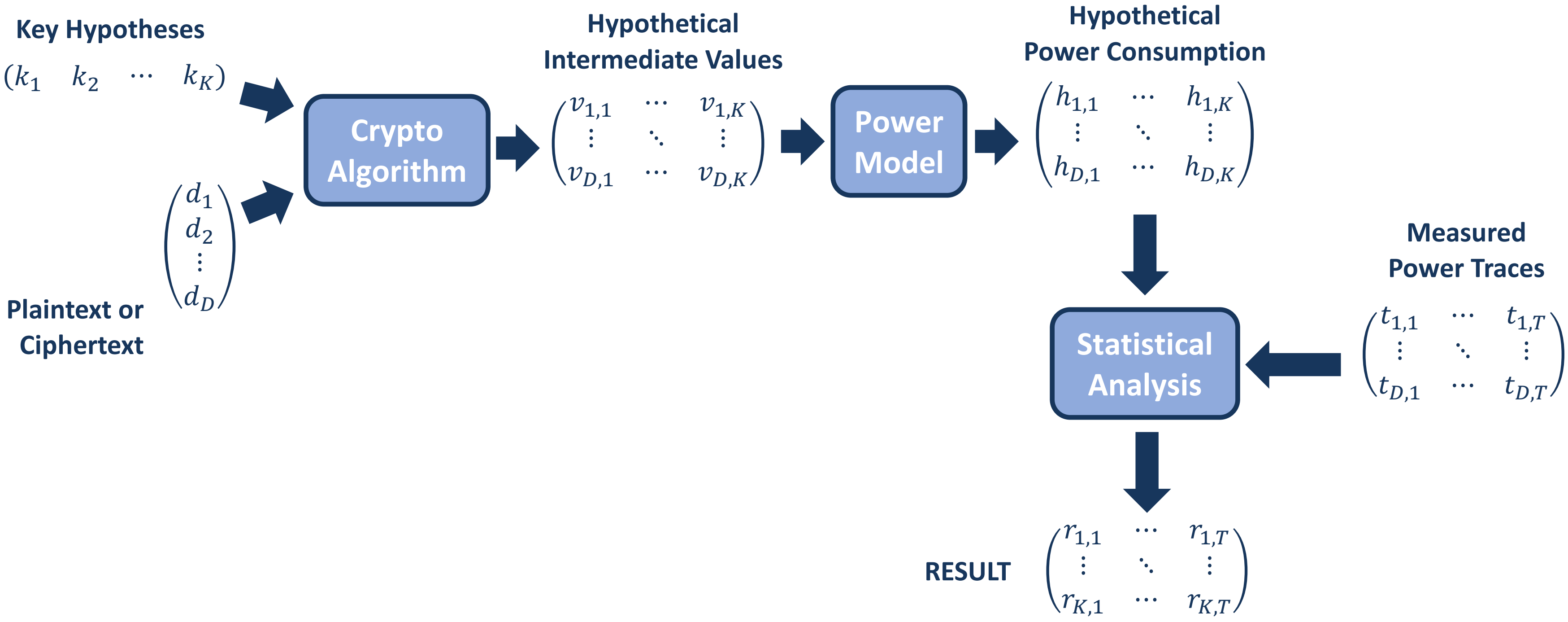
* Due to the extensive amount of combinations, these gadgets could not be verified with VERICA.

Welsh‘s t-test

$$\boxed{t = \frac{\mu_0 - \mu_1}{\sqrt{\frac{\sigma_0^2}{n_0} + \frac{\sigma_1^2}{n_1}}}}$$

Welsh‘s t-test is used to validate the null hypothesis.

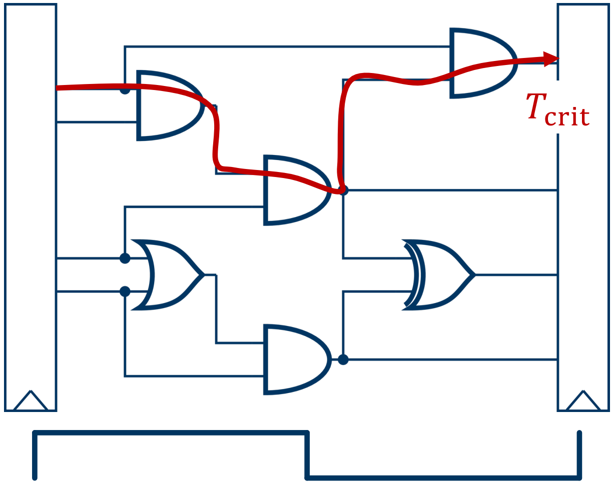
Differential Power Analysis (DPA)



[MOP08] Power analysis attacks: Revealing the secrets of smart cards.

Fault-Injection Attacks [RBSG21]

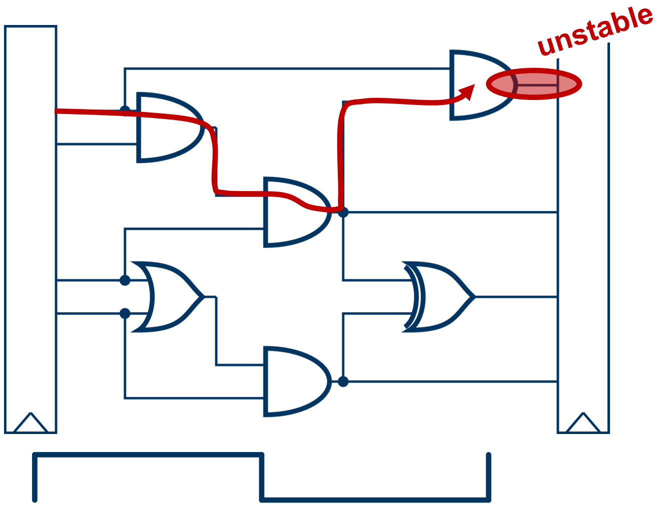
Clock Glitches



$$T_{\text{clk}} \geq T_{\text{crit}} + t_{\text{clkq}} + t_{\text{setup}} - \delta$$

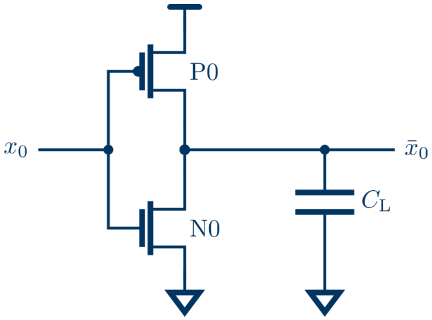
Fault-Injection Attacks [RBSG21]

Clock Glitches



$$T'_{\mathrm{clk}} < T_{\mathrm{crit}} + t_{\mathrm{clkq}} + t_{\mathrm{setup}} - \delta$$

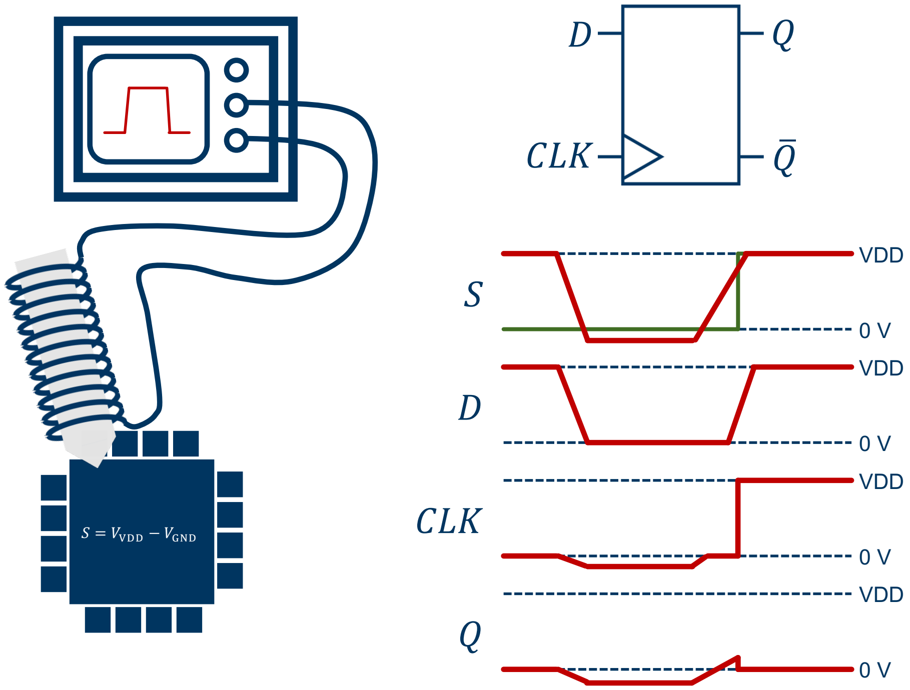
Underpowering and Voltage Glitches



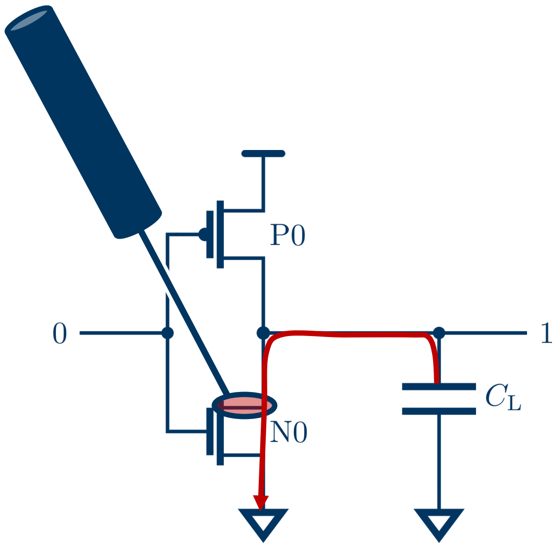
$$t_{pLH} = \frac{C_L \cdot \left[\frac{2|V_{TH2}|}{V_{DD} - |V_{TH2}|} + \ln\left(3 - \frac{4|V_{TH2}|}{V_{DD}}\right)\right]}{\mu_p C_{OX} \left(\frac{W}{L}\right)_2 (V_{DD} - |V_{TH2}|)}$$

Fault-Injection Attacks [RBSG21]

Electromagnetic Pulses

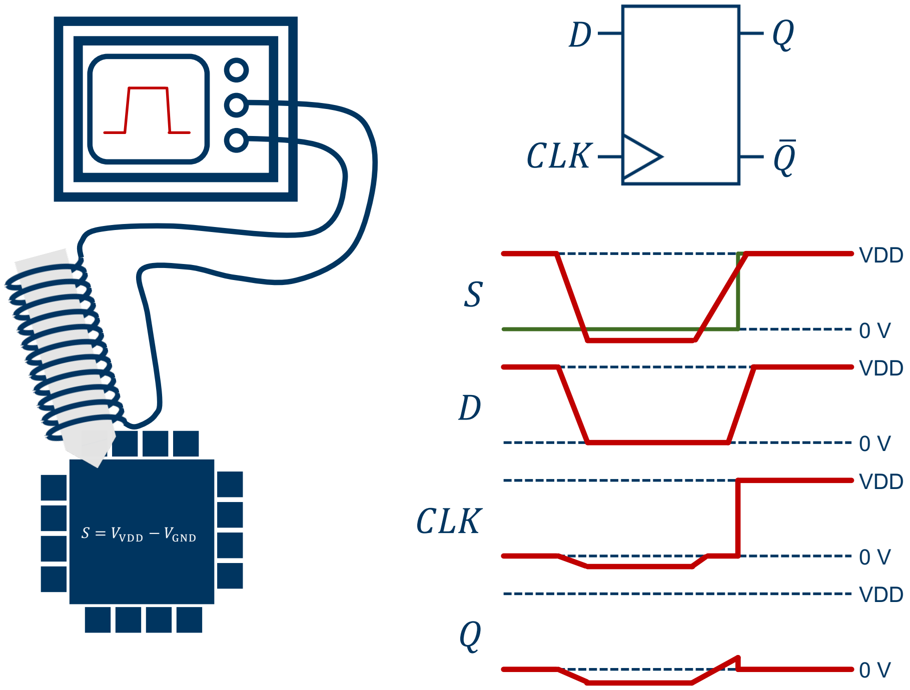


Laser Fault Injection

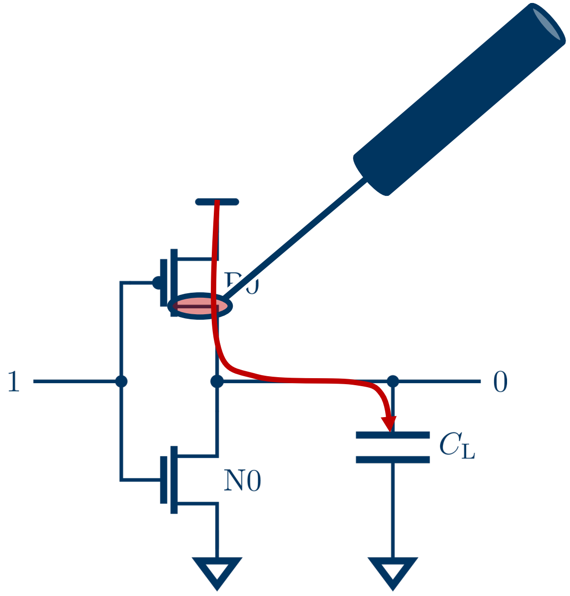


Fault-Injection Attacks [RBSG21]

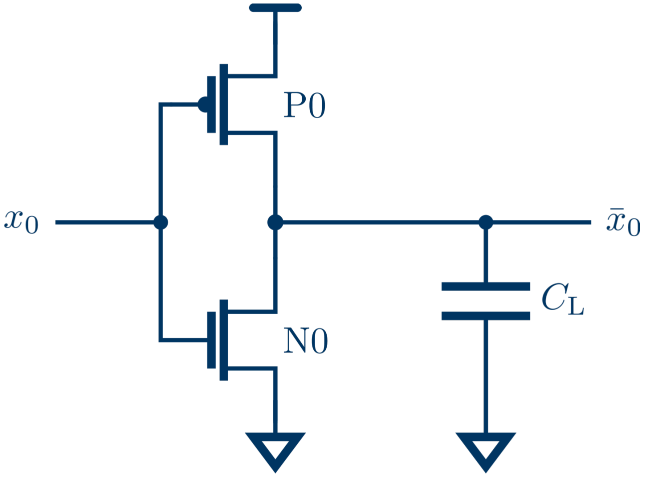
Electromagnetic Pulses



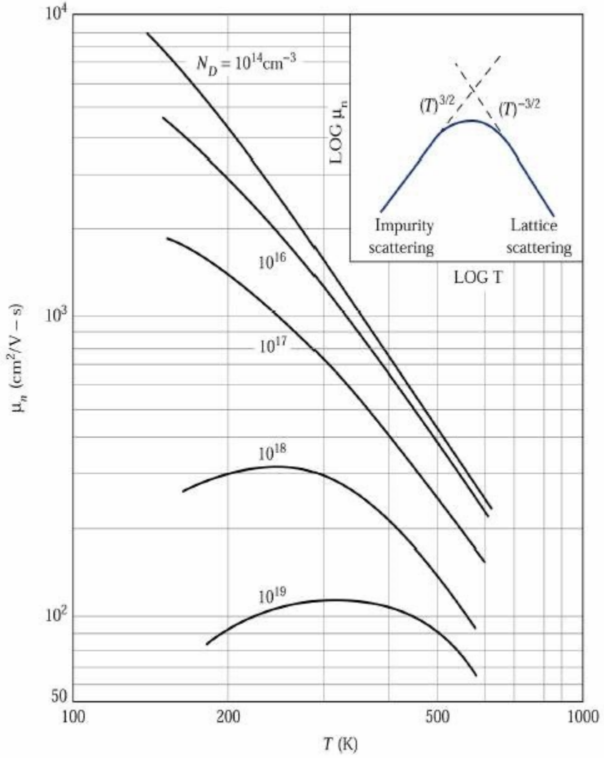
Laser Fault Injection



Fault-Injection Mechanisms – Temperature

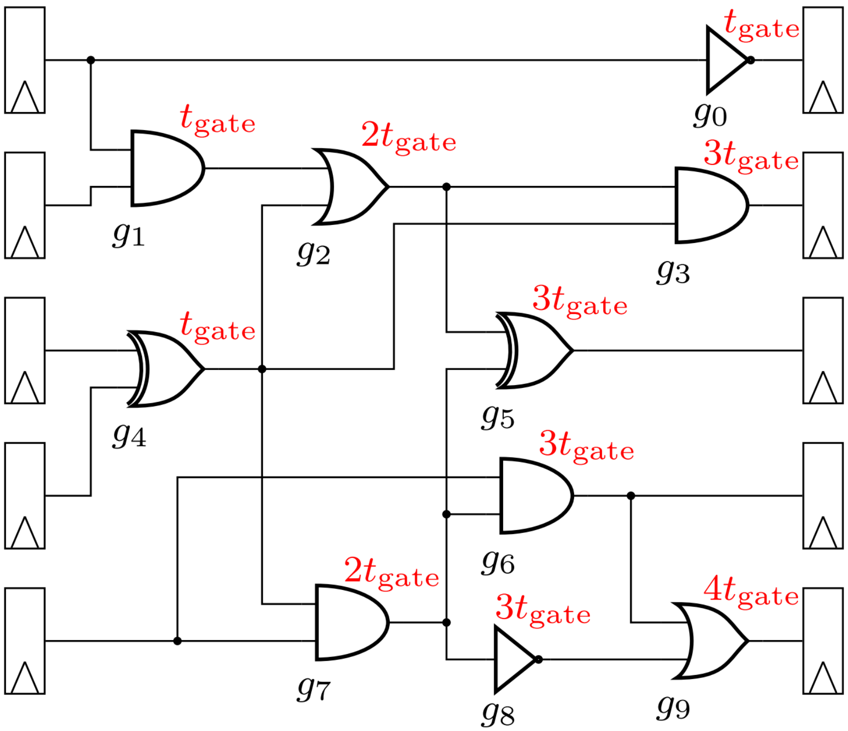


$$t_{PLH} = \frac{C_L \cdot \left[\frac{2|V_{TH2}|}{V_{DD} - |V_{TH2}|} + \ln\left(3 - \frac{4|V_{TH2}|}{V_{DD}}\right)\right]}{\mu_p C_{OX} \left(\frac{W}{L}\right)_2 (V_{DD} - |V_{TH2}|)}$$



[R08] Fundamentals of microelectronics.

Details about the Location Parameter



$$\mathcal{P} = \{t_o, t_1, \dots, t_{T-1}\}$$

where $t_0 > t_1 > \cdots > t_{T-1}$ and $T \leq |\mathcal{G}_{\text{regin}}|$

$$\mathcal{G}_{\text{cluster},i} = \{g \in \mathcal{G}_{\text{regin}} \mid t(g) \geq t_i, t_i \in \mathcal{P}\}$$

$$\mathcal{G}_{\text{cluster},0} = \{g_9\}$$

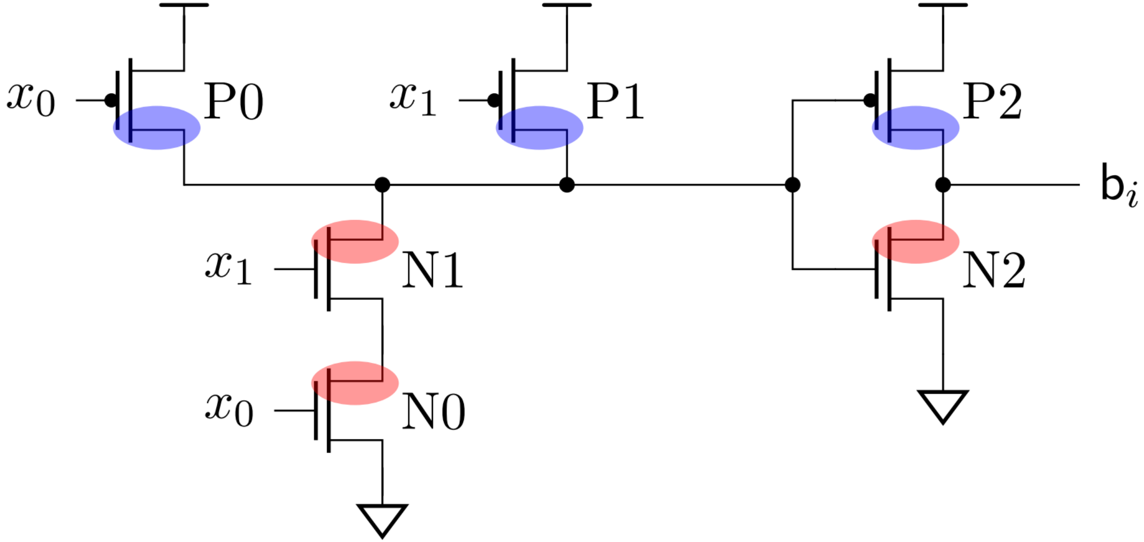
$$\mathcal{G}_{\text{cluster},1} = \{g_3, g_5, g_6\} \cup \{g_9\}$$

$$\mathcal{G}_{\text{cluster},2} = \{g_0\} \cup \{g_3, g_5, g_6\} \cup \{g_9\}$$

$$c_\infty = \{g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9\}$$

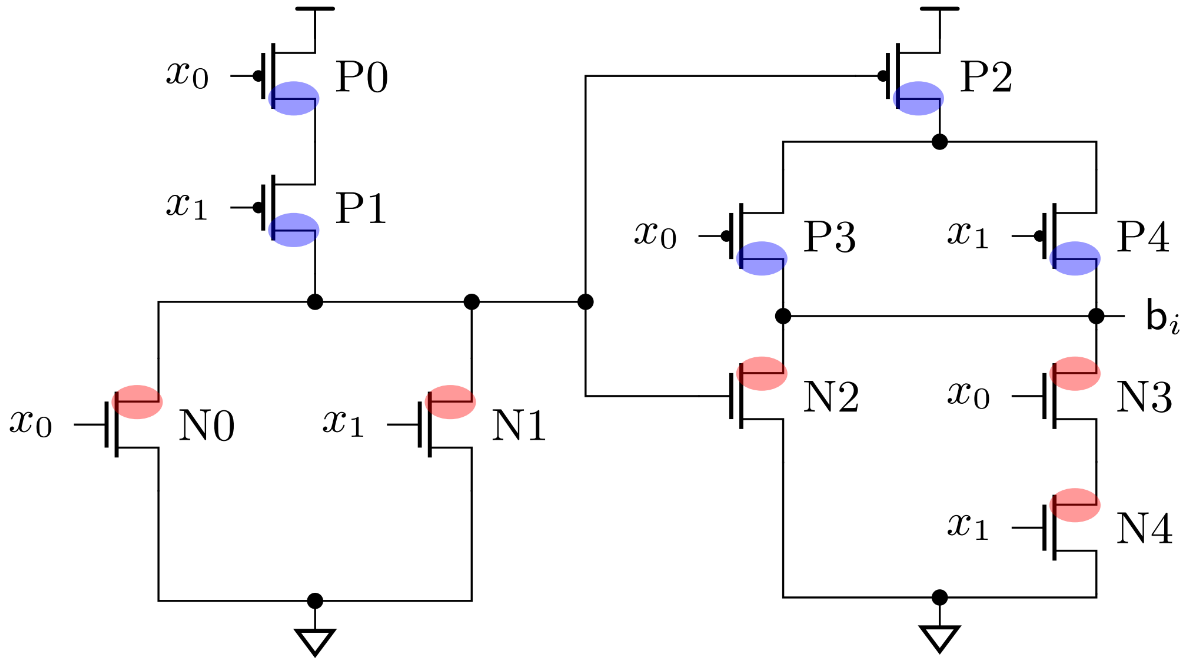
Fault Injections in CMOS Gates – AND

RUB



$$\{and\} \mapsto \{or, set, reset\}$$

Fault Injections in CMOS Gates – XOR



$$\{and\} \mapsto \{nand, or, set, reset\}$$