Soutenance de thèse

Université Claude Bernard Lyon 1

Modèles formels des circuits intégrés pour la vérification électrique au niveau transistor

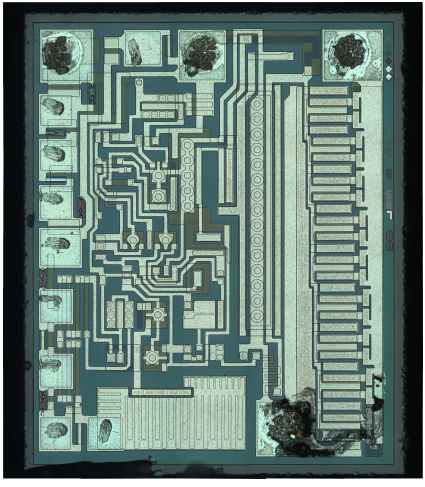
Oussama Oulkaid

21 novembre 2025

Devant le jury composé de

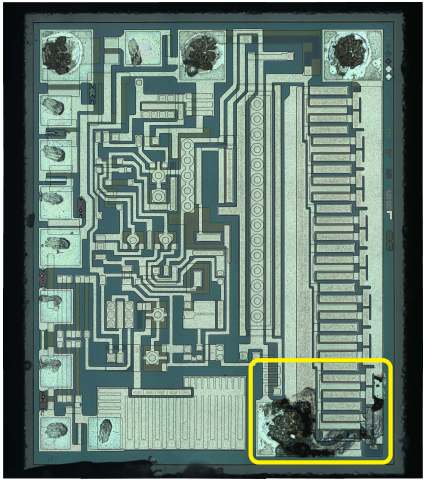
Emmanuelle Encrenaz	Professeure	Sorbonne Université	LIP6	Rapporteure
Katell Morin-Allory	Professeure	Grenoble INP	TIMA	Rapporteure
Arnaud Virazel	Professeur	Université de Montpellier	LIRMM	Examinateur
Lars Hedrich	Professeur	Goethe-Universität Institut für Informatik		Examinateur
Xavier Urbain	Professeur	Université Claude Bernard Lyon 1	LIRIS	Président du jury
Matthieu Moy	Maître de Conférences	Université Claude Bernard Lyon 1	LIP	Directeur de thèse
Pascal Raymond	Chargé de Recherche	CNRS	Verimag	Co-encadrant de thèse
Bruno Ferres	Maître de Conférences	Université Grenoble Alpes	Verimag	Co-encadrant de thèse
Mehdi Khosravian	Ingénieur de Recherche	Aniah		Co-encadrant de thèse

Errors in the Hardware



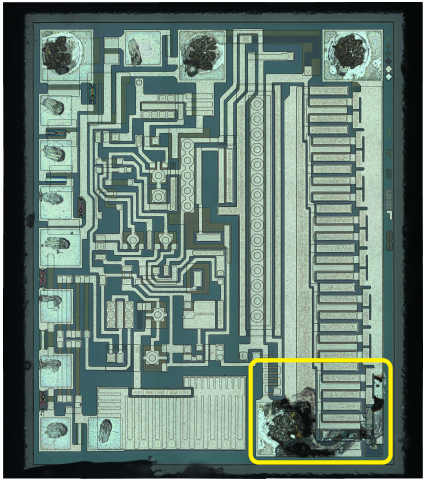
Credit: Andrew Huang (2007)

Electrostatic discharge



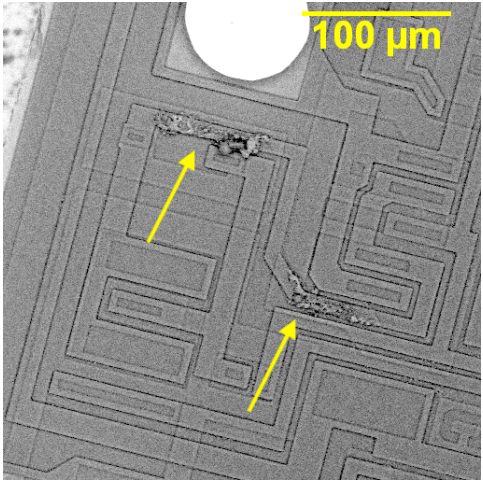
Credit: Andrew Huang (2007)

Electrostatic discharge



Credit: Andrew Huang (2007)

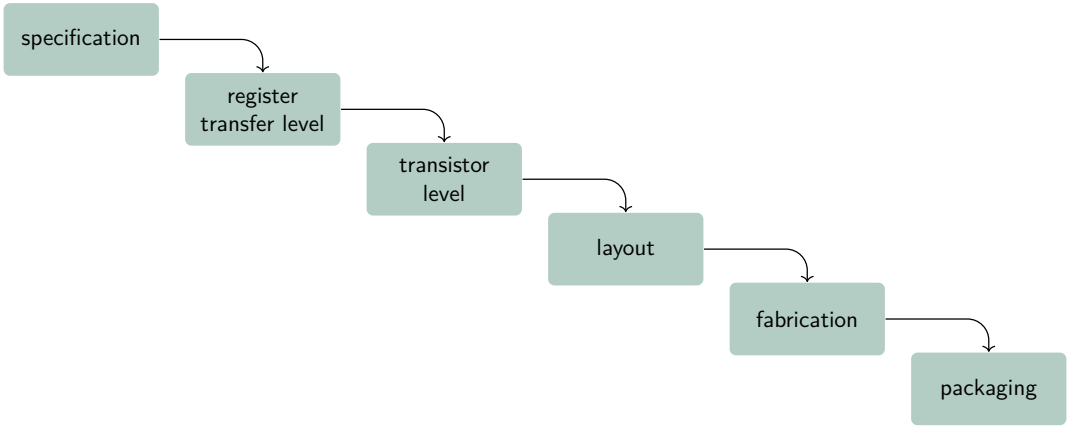
Electrical overstress



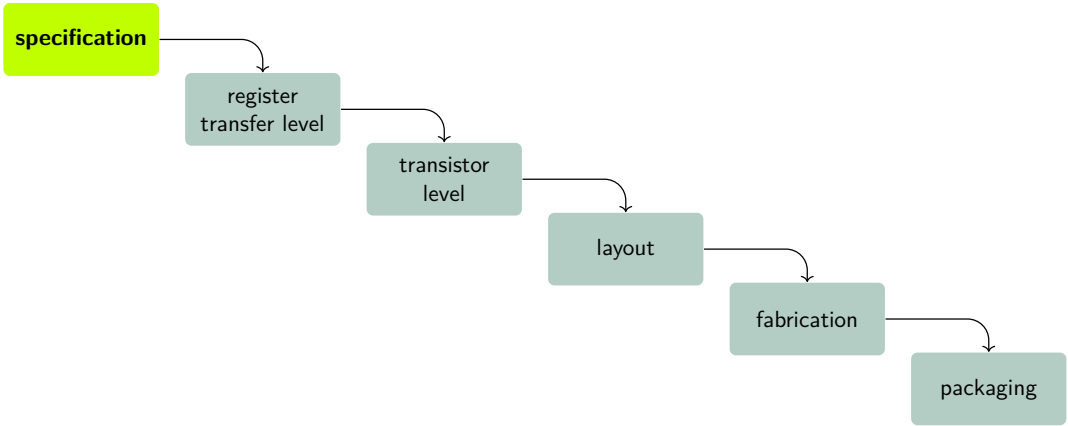
Credit: Ed Hare (2020)

Where do errors come from?

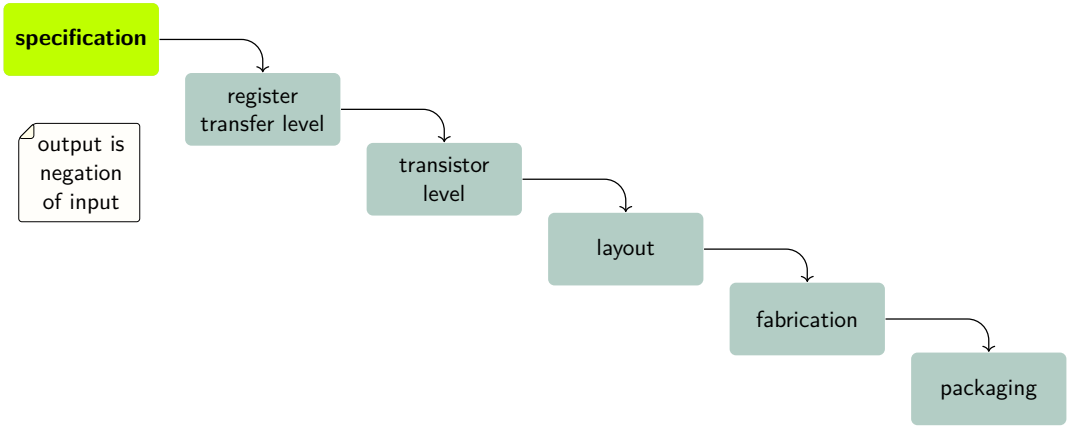
Where do errors come from?



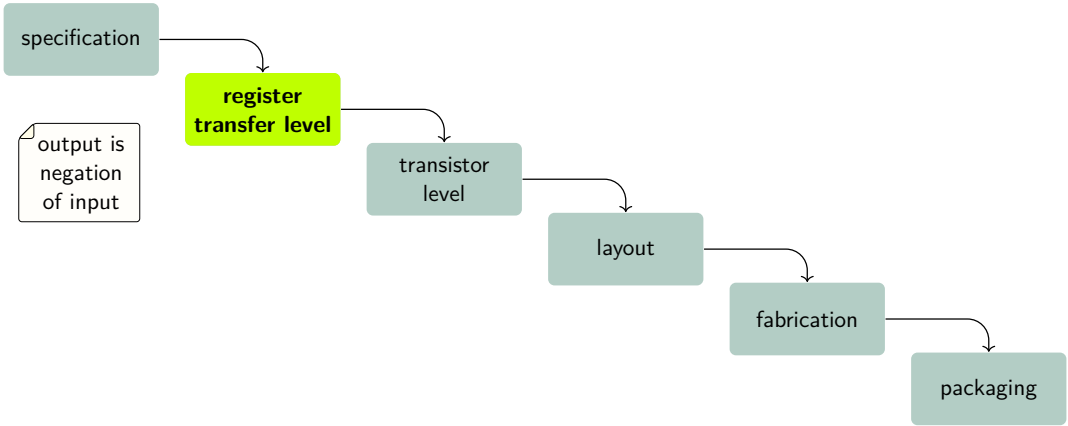
Where do errors come from?



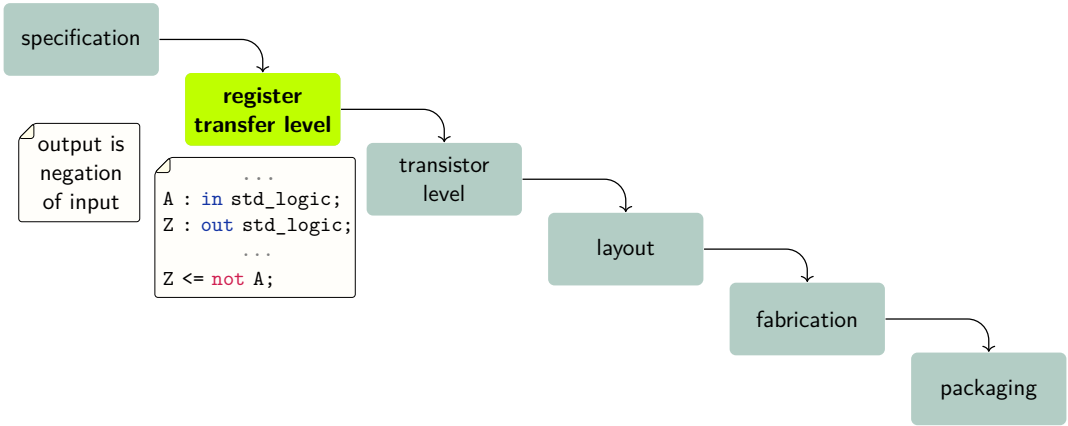
Where do errors come from?



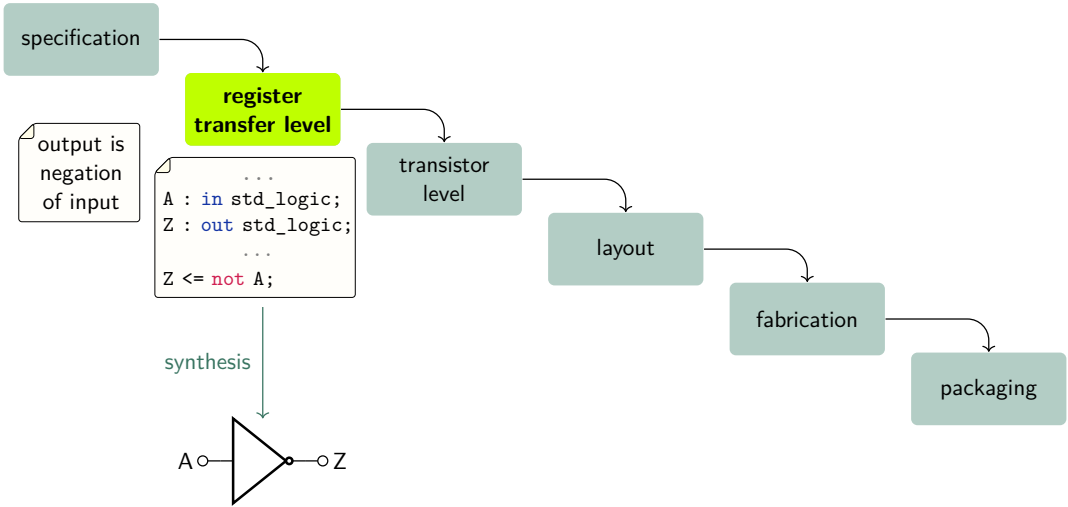
Where do errors come from?



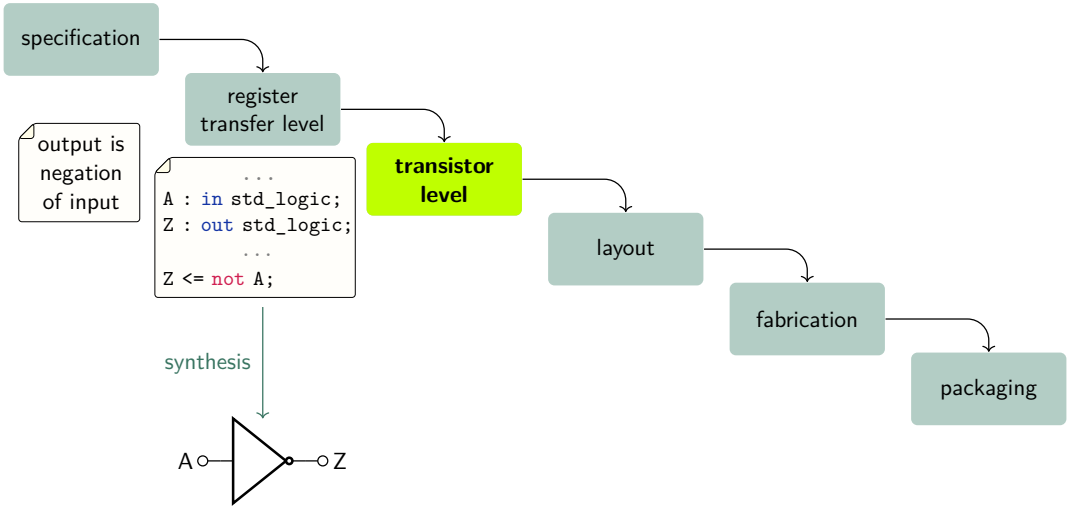
Where do errors come from?



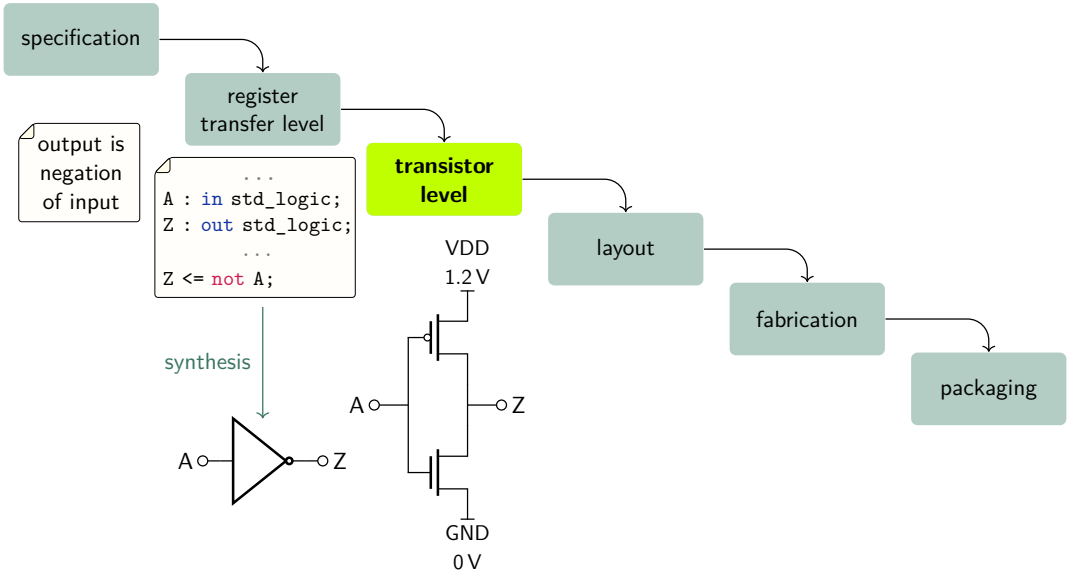
Where do errors come from?



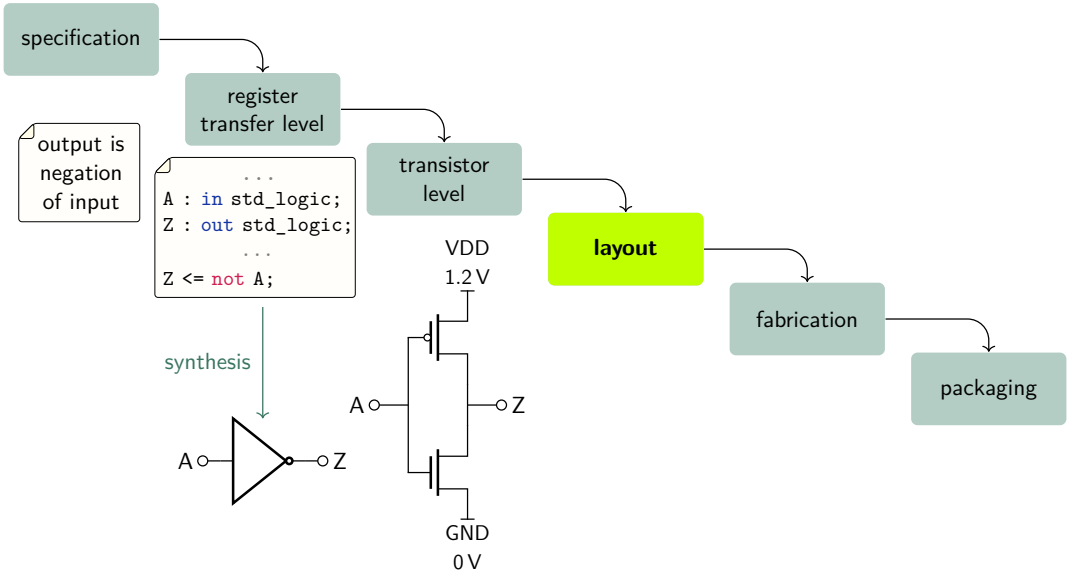
Where do errors come from?



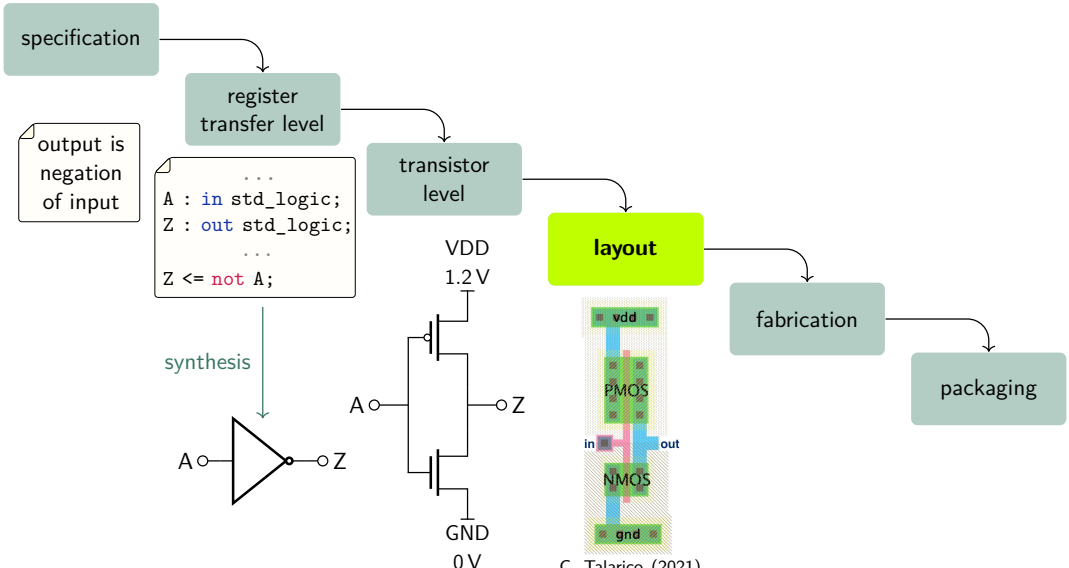
Where do errors come from?



Where do errors come from?

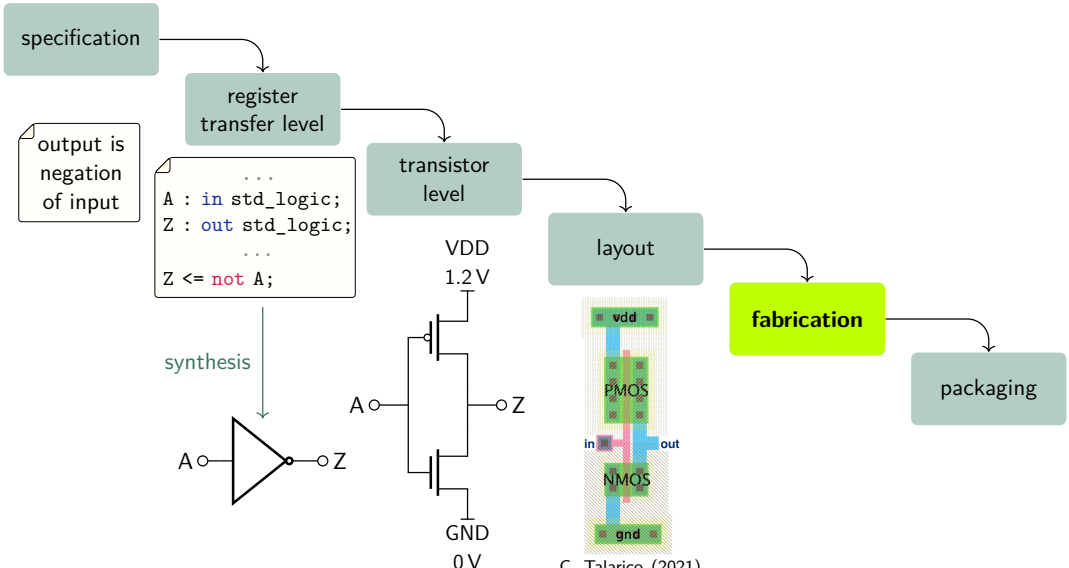


Where do errors come from?



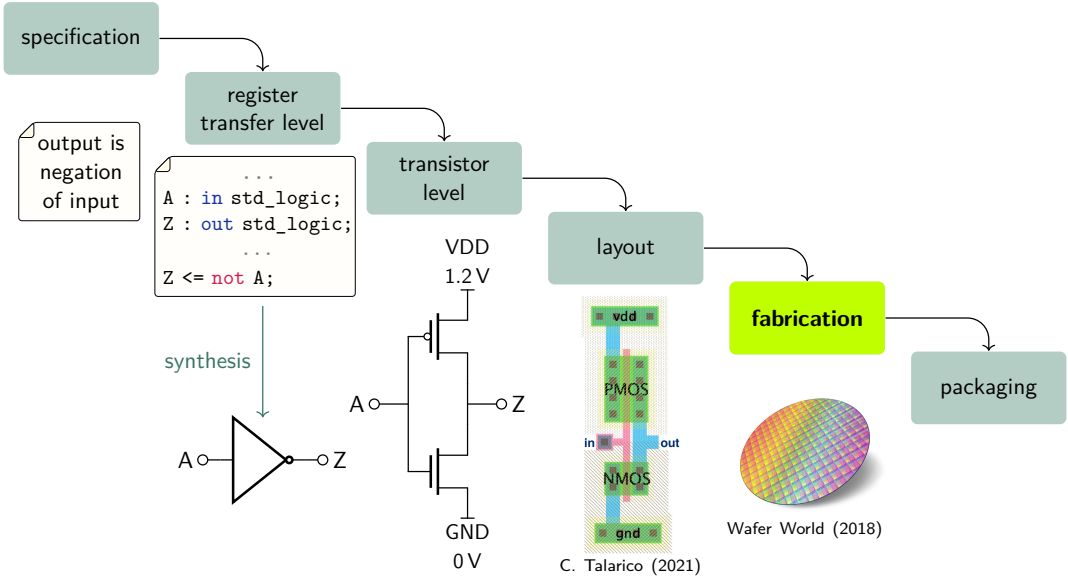
C. Talarico (2021)

Where do errors come from?

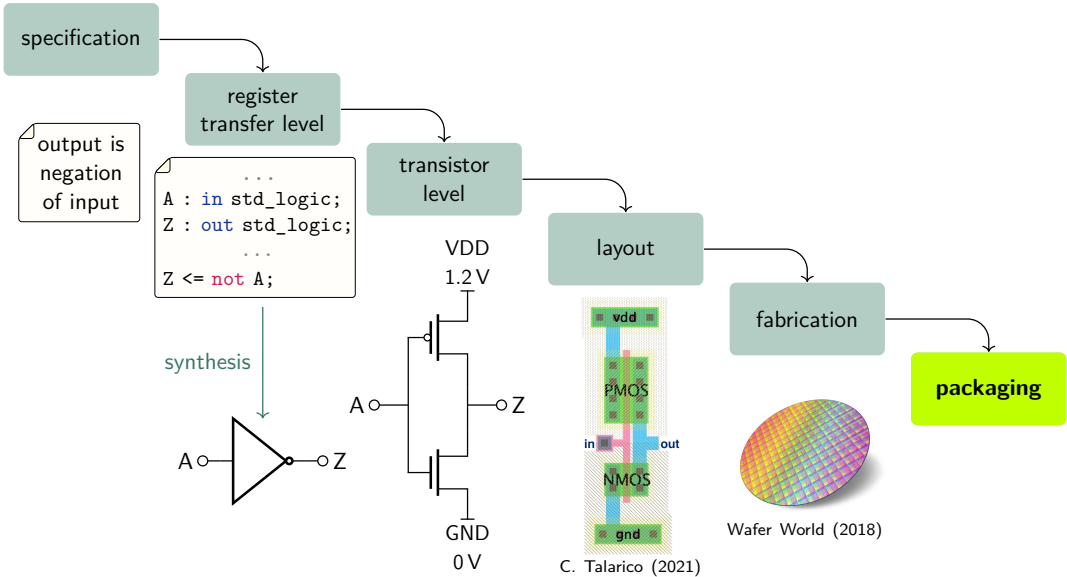


C. Talarico (2021)

Where do errors come from?



Where do errors come from?



Where do errors come from?

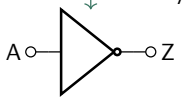
specification

register transfer level

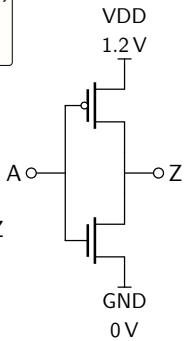
output is negation of input

```
...
A : in std_logic;
Z : out std_logic;
...
Z <= not A;
```

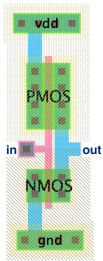
synthesis



transistor level

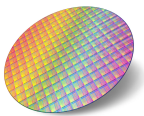


layout



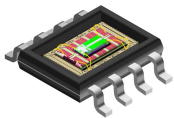
C. Talarico (2021)

fabrication



Wafer World (2018)

packaging



T. Kim (2021)

Where do errors come from?

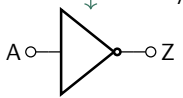
specification

register transfer level

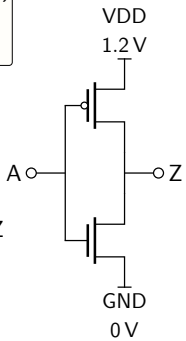
output is negation of input

```
...
A : in std_logic;
Z : out std_logic;
...
Z <= not A;
```

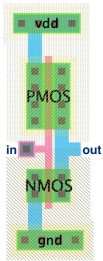
synthesis



transistor level

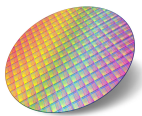


layout



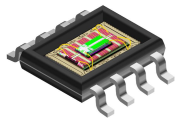
C. Talarico (2021)

fabrication



Wafer World (2018)

packaging



T. Kim (2021)

Where do errors come from?

specification

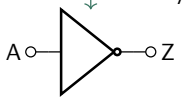
functional verification

register transfer level

output is negation of input

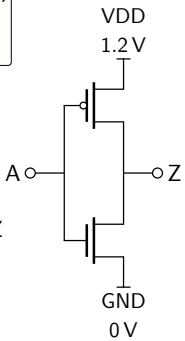
```
...
A : in std_logic;
Z : out std_logic;
...
Z <= not A;
```

synthesis



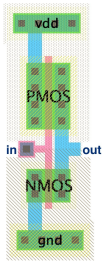
electrical rule checking

transistor level



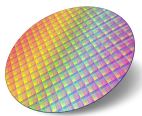
layout versus schematics

layout



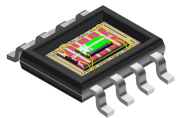
C. Talarico (2021)

fabrication



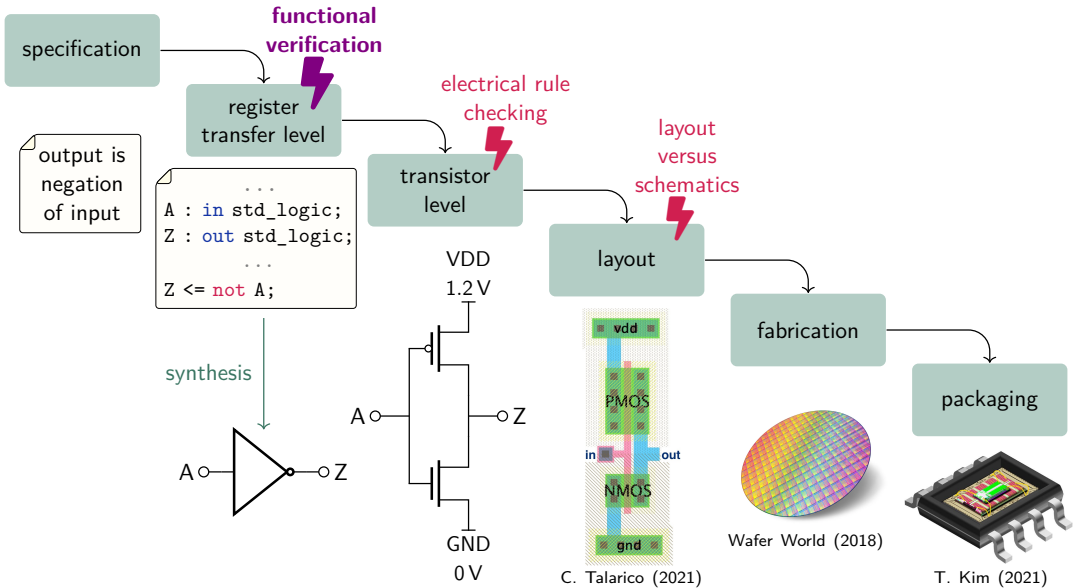
Wafer World (2018)

packaging

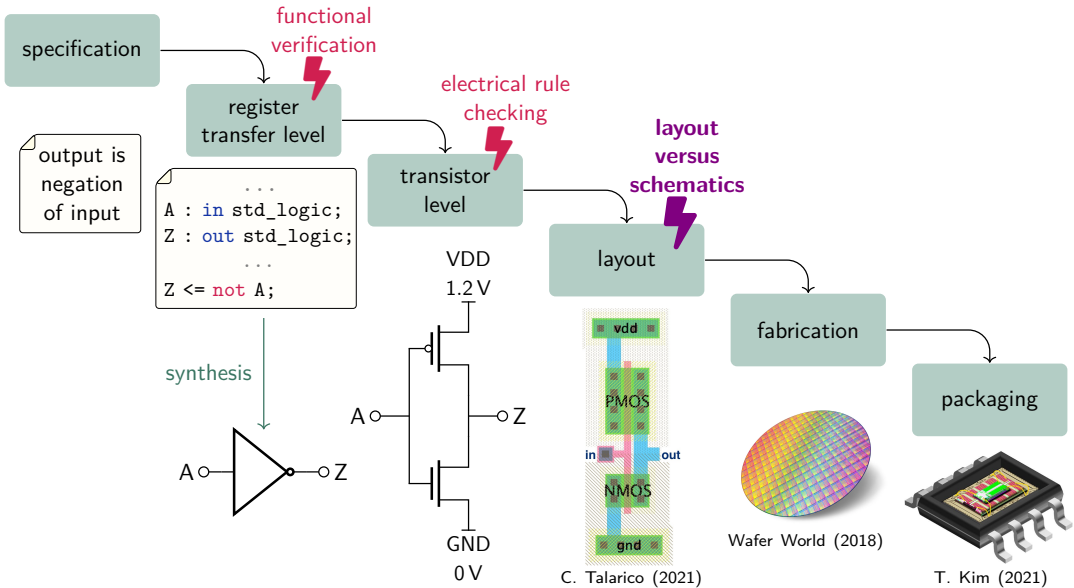


T. Kim (2021)

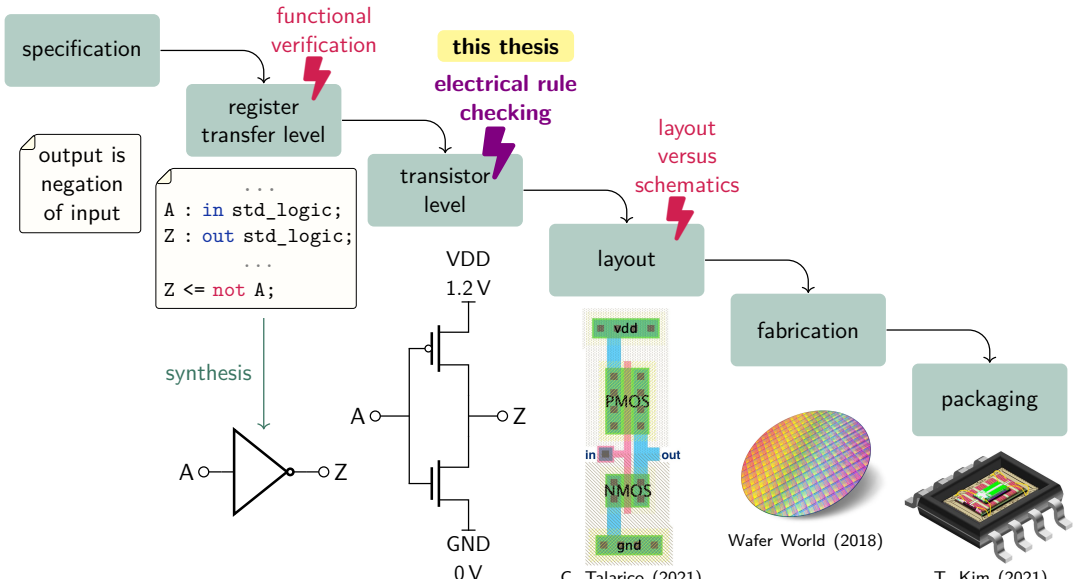
Where do errors come from?



Where do errors come from?



Where do errors come from?

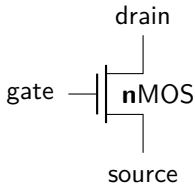


Part 1 of 5

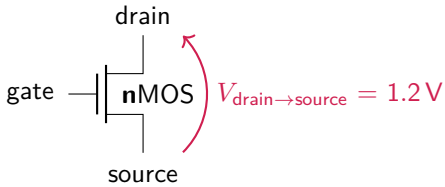
Understanding Transistors and Electrical Errors

Transistor level, the basics

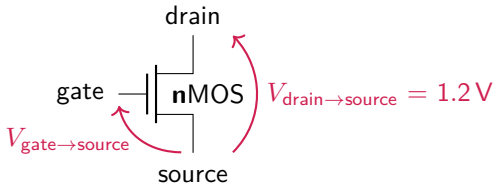
Transistor level, the basics



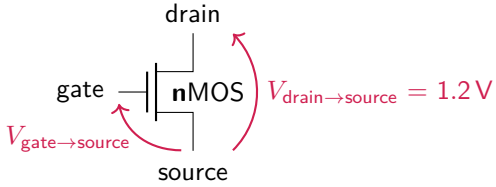
Transistor level, the basics

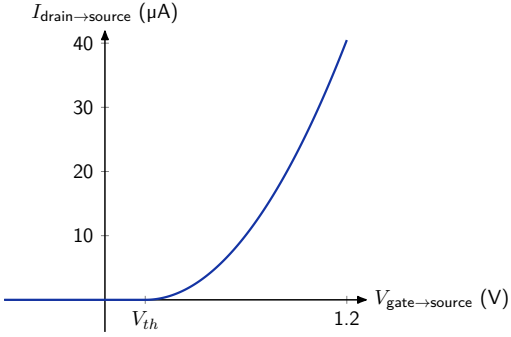


Transistor level, the basics

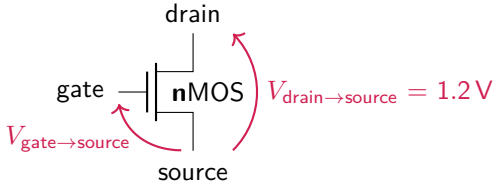


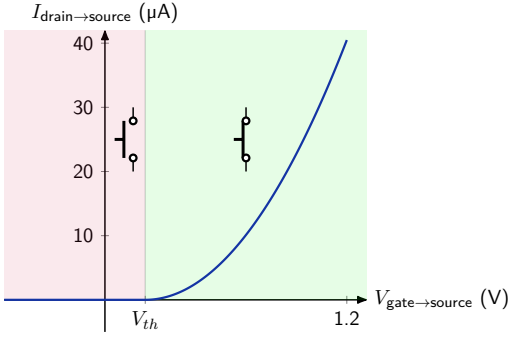
Transistor level, the basics



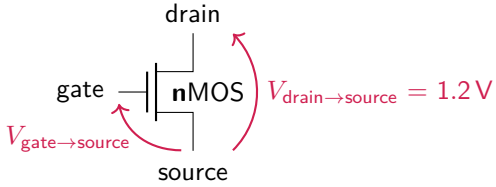


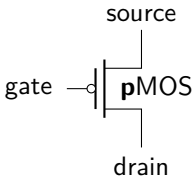
Transistor level, the basics

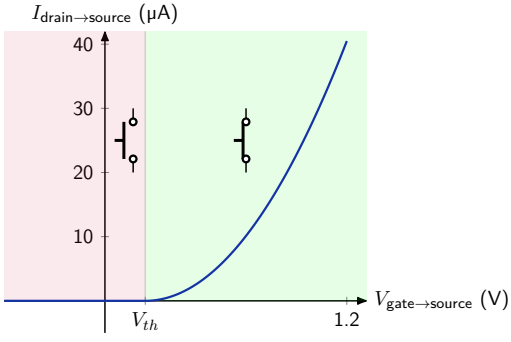




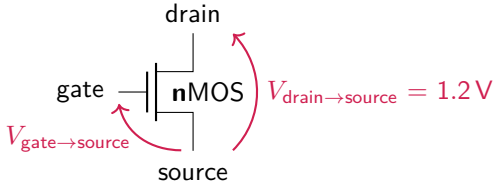
Transistor level, the basics

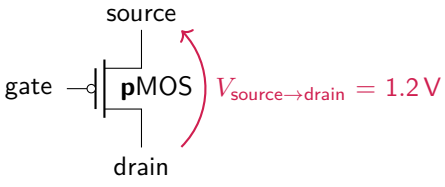


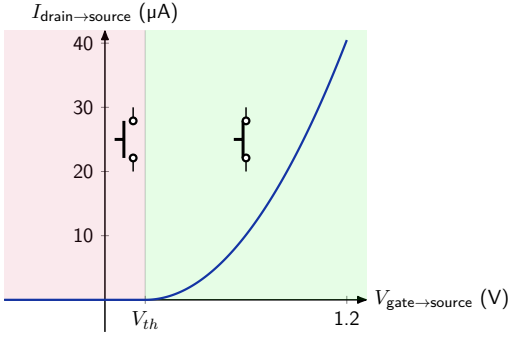




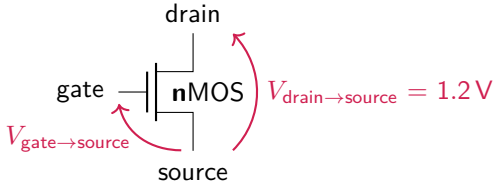
Transistor level, the basics

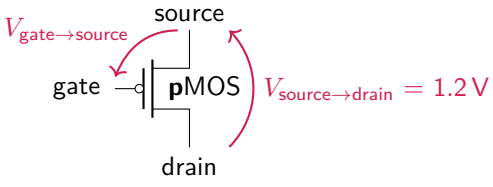


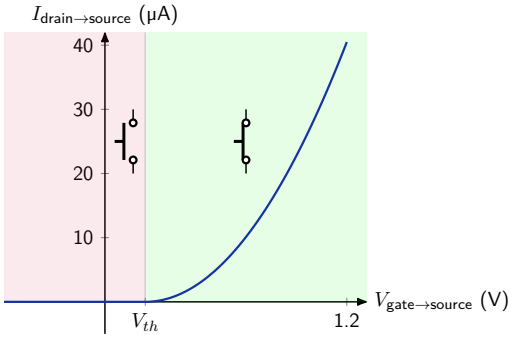




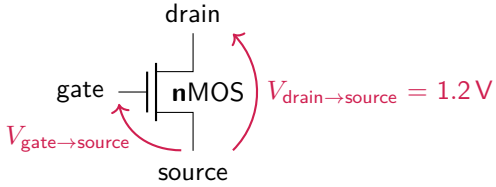
Transistor level, the basics

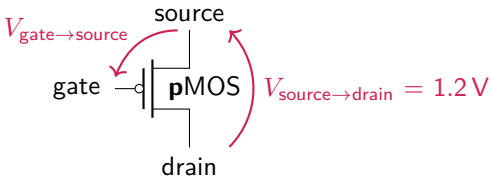


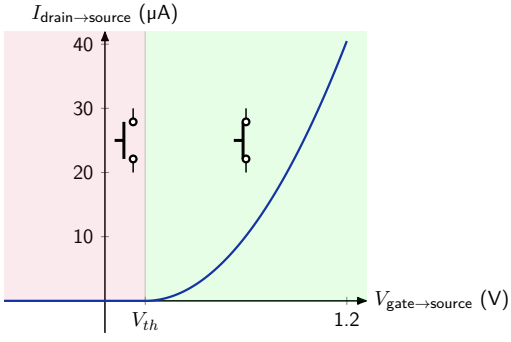


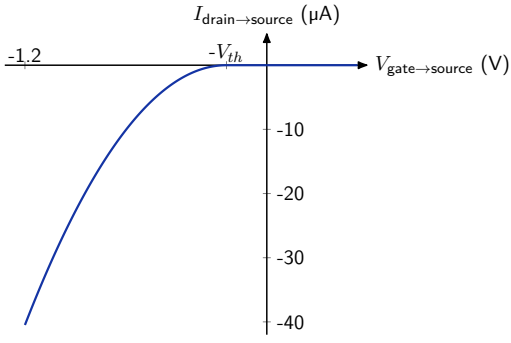


Transistor level, the basics

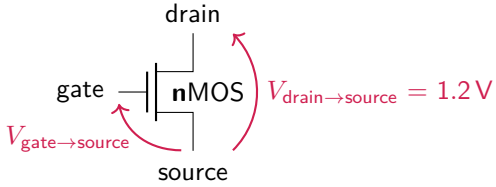


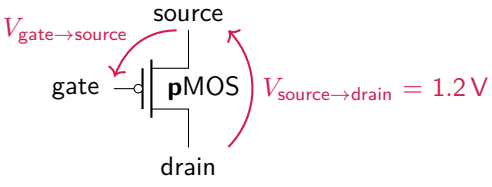


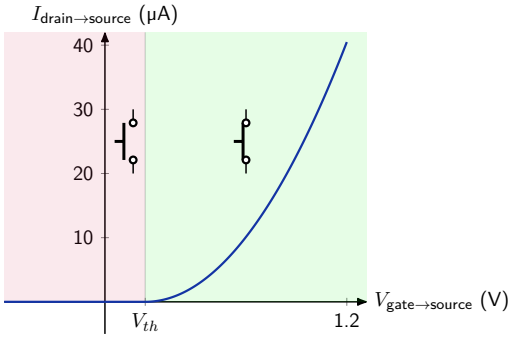


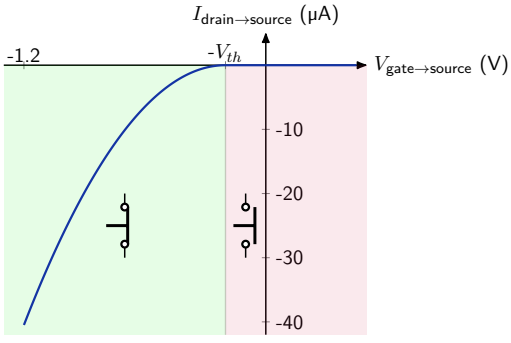


Transistor level, the basics

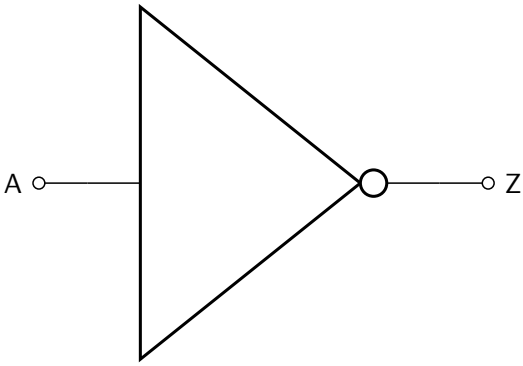




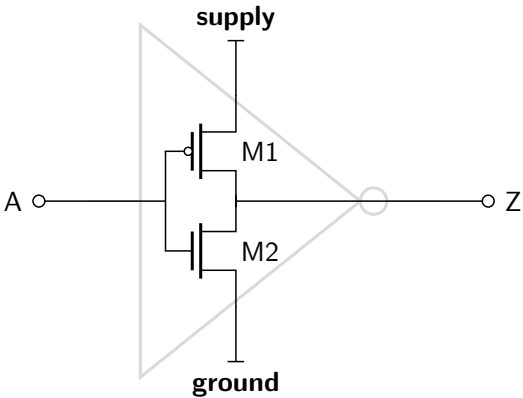




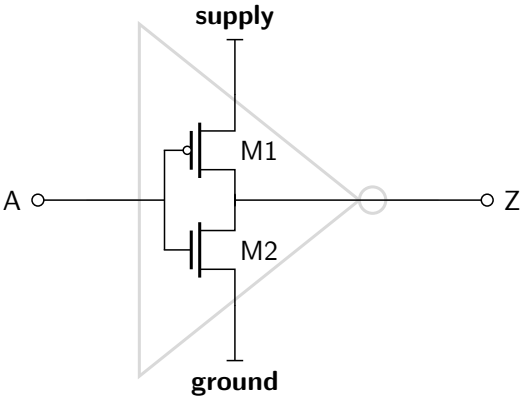
The inverter circuit



The inverter circuit



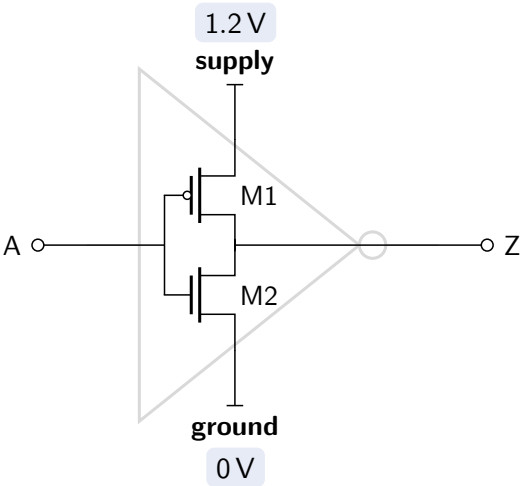
The inverter circuit



inverter.cdl

```
.subckt inverter A Z supply ground

M1 Z A supply supply PMOS ...
M2 Z A ground ground NMOS ...

.ends
```

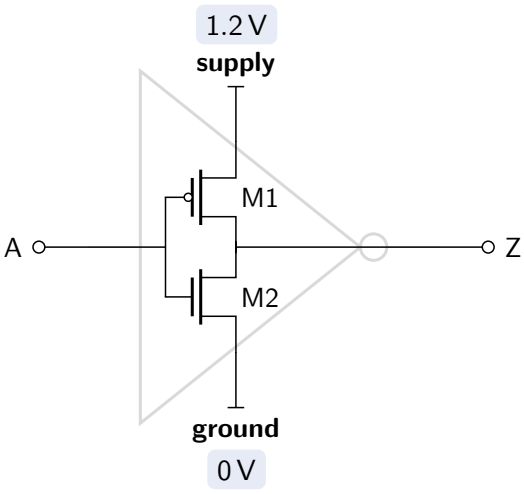
The inverter circuit



inverter.cdl

```
.subckt inverter A Z supply ground

M1 Z A supply supply PMOS ...
M2 Z A ground ground NMOS ...

.ends
```

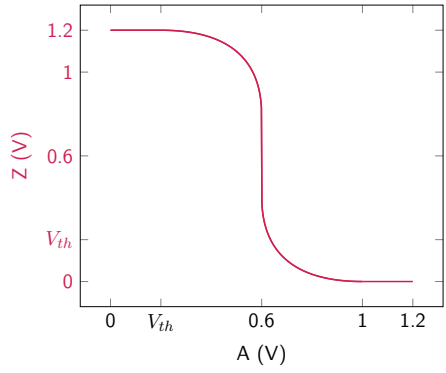
The inverter circuit



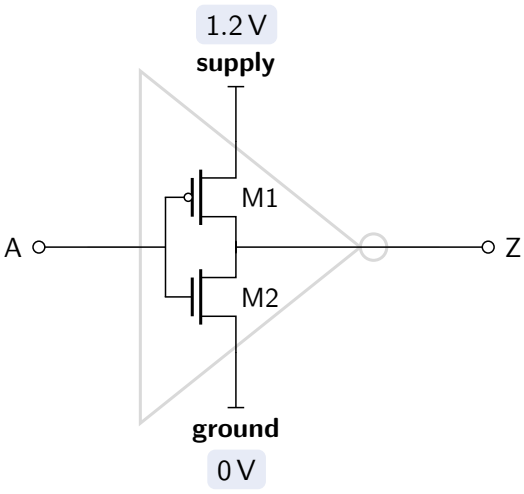
inverter.cdl

```
.subckt inverter A Z supply ground

M1 Z A supply supply PMOS ...
M2 Z A ground ground NMOS ...

.ends
```

Electrical simulation

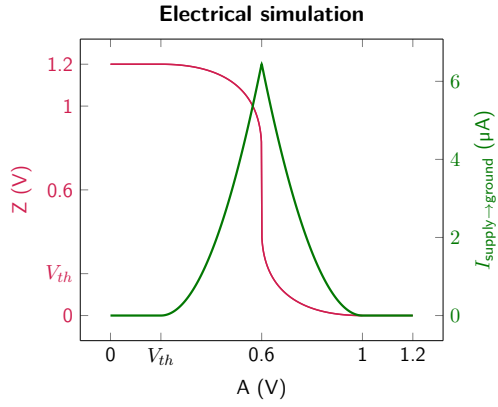


The inverter circuit



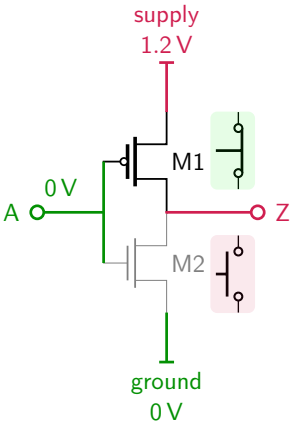
inverter.cdl

```
.subckt inverter A Z supply ground

M1 Z A supply supply PMOS ...
M2 Z A ground ground NMOS ...

.ends
```

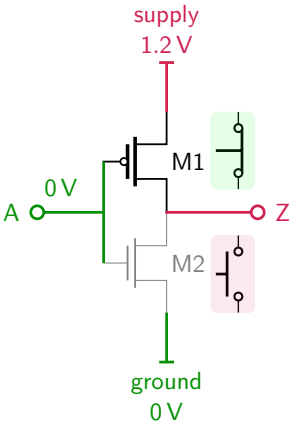


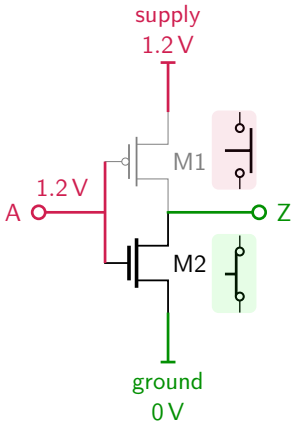
Symbolic reasoning on the inverter

Symbolic reasoning on the inverter

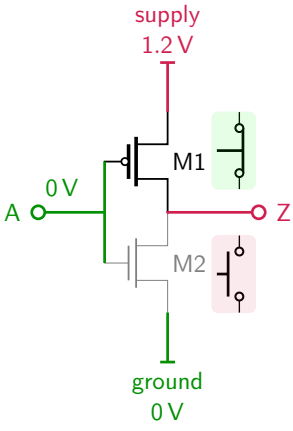


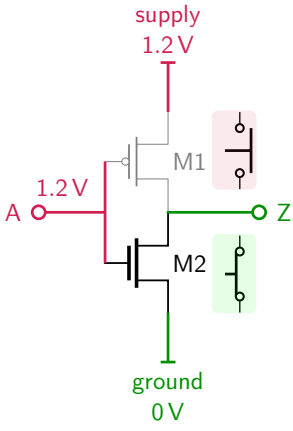
Symbolic reasoning on the inverter

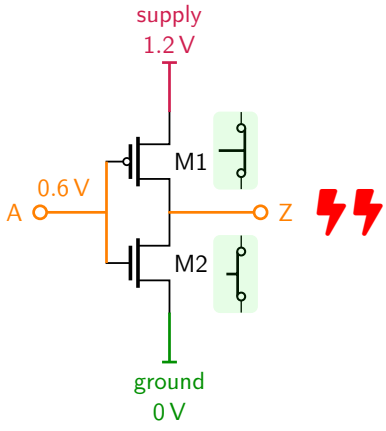




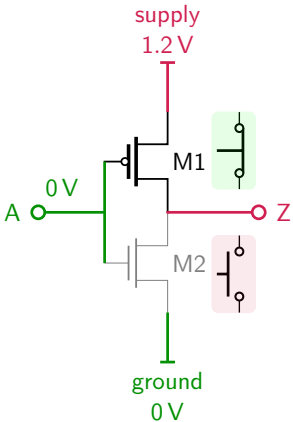
Symbolic reasoning on the inverter

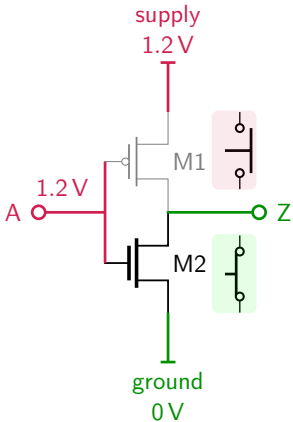


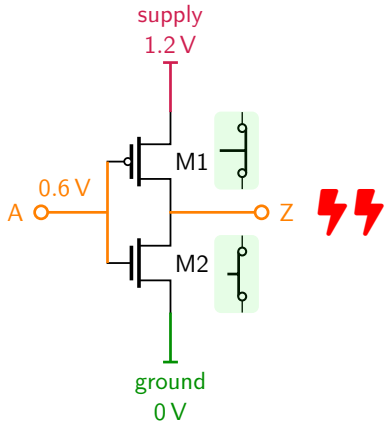




Symbolic reasoning on the inverter

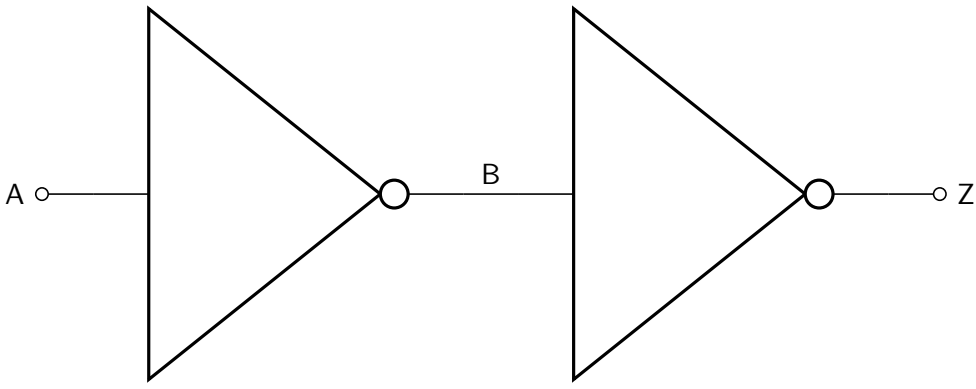




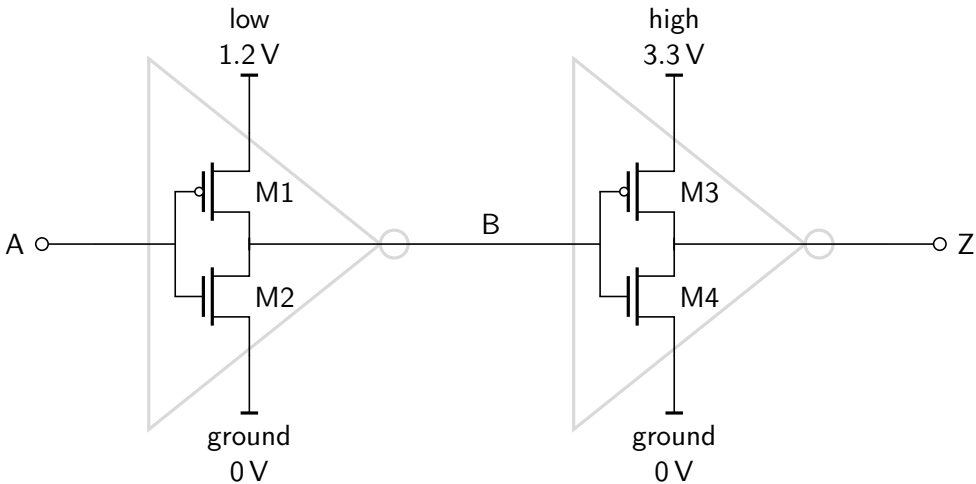


Can this happen on a real-life circuit?

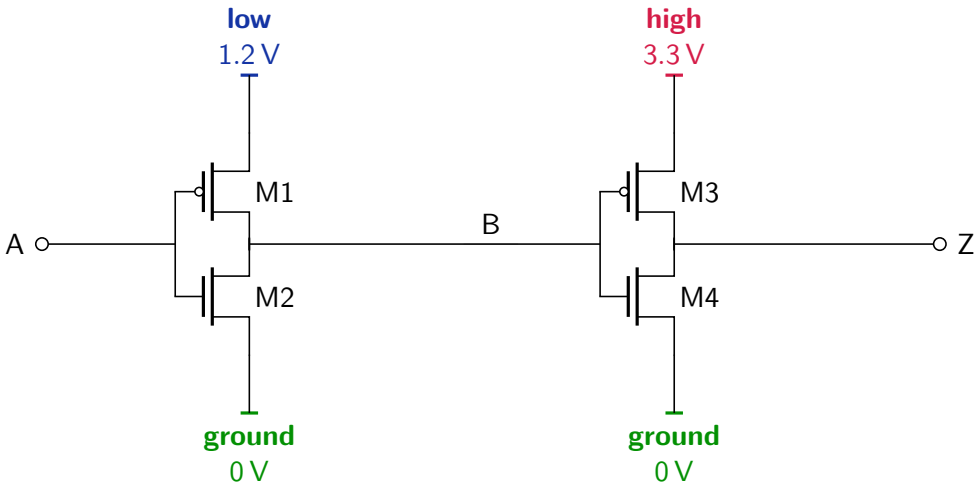
A buffer



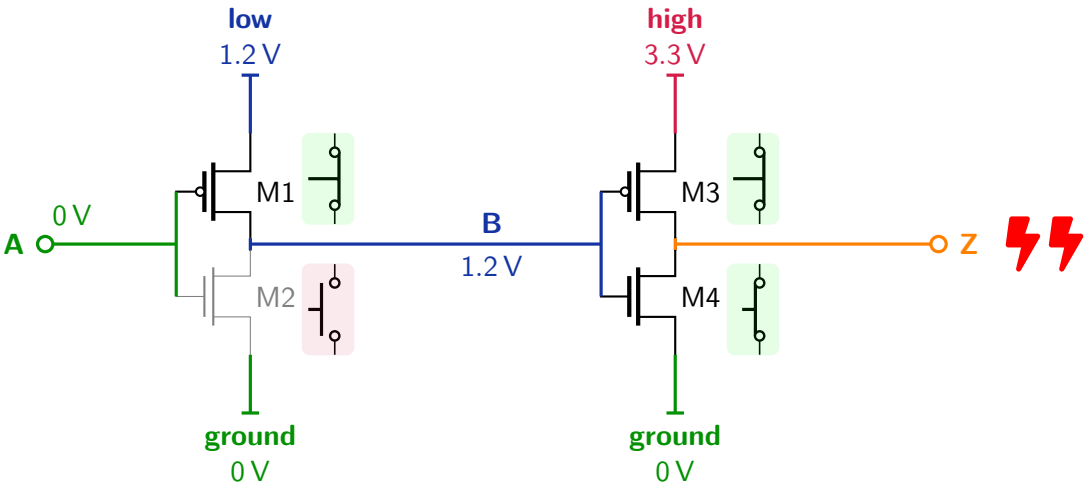
A buffer, a *buggy* one



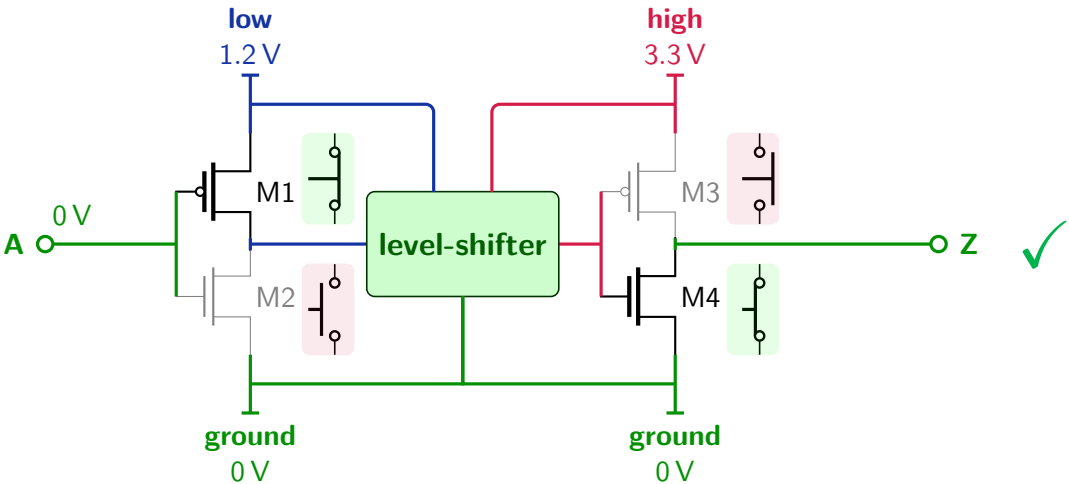
A buffer, a *buggy* one



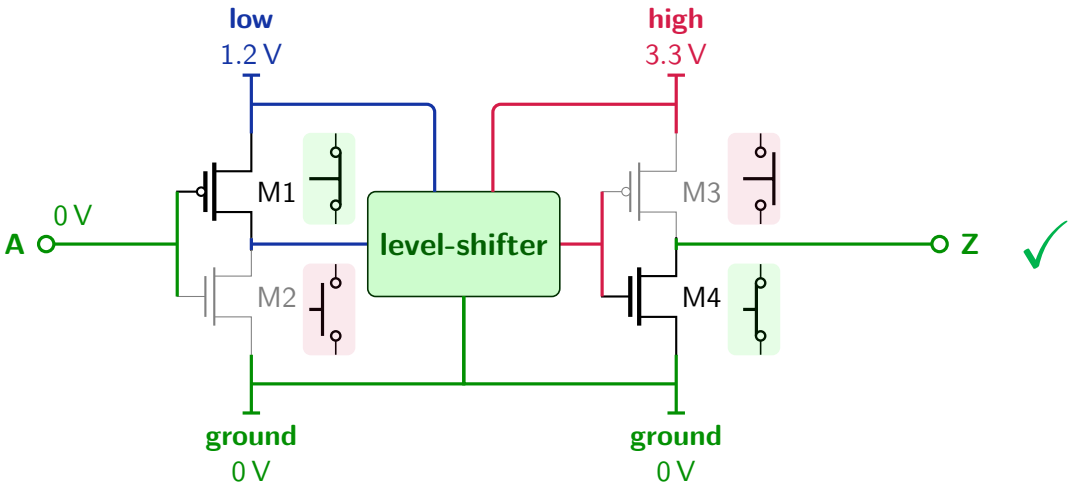
A buffer, a *buggy* one



A buffer, ~~a *buggy* one~~ fixed



A buffer, ~~a *buggy* one~~ fixed



How to detect such violation in the first place?

Part 2 of 5

State of the Art in Electrical Rule Checking

Modeling techniques for electrical rule checking

Modeling techniques for electrical rule checking

ACM TODAES 2025

A Survey on Transistor-Level Electrical Rule Checking of Integrated Circuits

BRUNO FERRES, Univ. Grenoble Alpes, CNRS, Grenoble INP[2], VERIMAG, 38000 Grenoble, France
OUSSAMA OULKAID, Université Claude Bernard Lyon 1, CNRS, ENS de Lyon, Inria, LIP, UMR 5668, 69342, Lyon cedex 07, France; Univ. Grenoble Alpes, CNRS, Grenoble INP[2], VERIMAG, 38000 Grenoble, France; Aniah, Grenoble, France
MATTHIEU MOY, GABRIEL RADANNE, and LUDOVIC HENRIO, CNRS, ENS de Lyon, Inria, Université Claude Bernard Lyon 1, LIP, UMR 5668, 69342, Lyon cedex 07, France
PASCAL RAYMOND, Univ. Grenoble Alpes, CNRS, Grenoble INP[2], VERIMAG, 38000 Grenoble, France
MEHDI KHOSRAVIAN GHADIKOLAEI, Aniah, Grenoble , France

Hardware verification is crucial to ensure the quality of Integrated Circuits, and prevent costly bugs down the manufacturing flow. *Electrical Rule Checking* (ERC) is a verification step used to assert that a circuit complies with some electrical rules, from the absence of short-circuits to dedicated constructor rules. In this survey, we provide a global overview of existing ERC techniques at transistor-level, where voltage values are explicit. We propose a new classification method to compare the existing approaches based on their semantic modeling of circuits. This survey precisely describes transistor-level ERC research challenges and existing solutions. We believe it will help structure this research domain by positioning existing approaches with respect to each other. Obviously, a survey should also facilitate technological transfer and this one should help CAD vendors identify the most relevant approaches to integrate in their tools. Finally, we highlight several promising directions to improve the existing solutions.

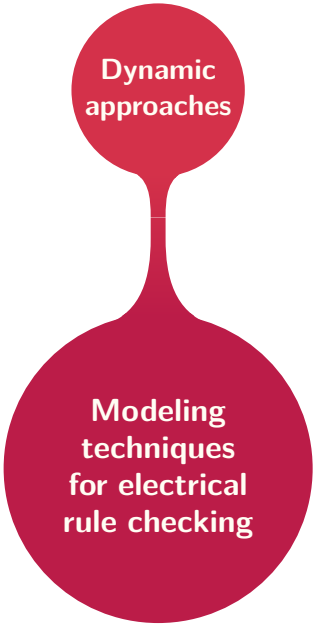
CCS Concepts: • **General and reference** → **Surveys and overviews**; **Verification**; • **Hardware** → **Electronic designs automation**; **Design rule checking**.

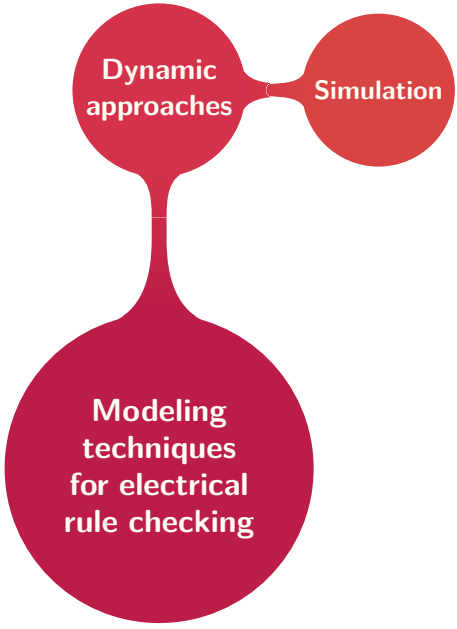
Additional Key Words and Phrases: Electrical Rule Checking, Integrated Circuits, Electro-Static Discharge, Electrical OverStress, Static Verification

ACM Reference Format:
Bruno Ferres, Oussama Oulkaid, Matthieu Moy, Gabriel Radanne, Ludovic Henrio, Pascal Raymond, and Mehdi Khosravian Ghadikolaei. 2025. A Survey on Transistor-Level Electrical Rule Checking of Integrated Circuits. *ACM Trans. Des. Autom. Electron. Syst.* 1, 1, Article 1 (January 2025), 28 pages. https://doi.org/10.1145/3748327

Authors' addresses: Bruno Ferres, Univ. Grenoble Alpes, CNRS, Grenoble INP[2], VERIMAG, 38000 Grenoble, France; bruno.ferres@univ-grenoble-alpes.fr; Oussama Oulkaid, Université Claude Bernard Lyon 1, CNRS, ENS de Lyon, Inria, LIP, UMR 5668, 69342, Lyon cedex 07, France; Univ. Grenoble Alpes, CNRS, Grenoble INP[2], VERIMAG, 38000 Grenoble, France; Aniah, Grenoble, France; oussama.oulkaid@univ-grenoble-alpes.fr; Matthieu Moy, matthieu.moy@univ-lyon1.fr; Gabriel Radanne, gabriel.radanne@inria.fr; Ludovic Henrio, CNRS, ENS de Lyon, Inria, Université Claude Bernard Lyon 1, LIP, UMR 5668, 69342, Lyon cedex 07, France; ludovic.henrio@ens-lyon.fr; Pascal Raymond, Univ. Grenoble Alpes, CNRS, Grenoble INP[2], VERIMAG, 38000 Grenoble, France; pascal.raymond@univ-grenoble-alpes.fr; Mehdi Khosravian Ghadikolaei, Aniah, Grenoble , France, mehdi.khosravian@aniah.fr.

ACM 1084-4309/2025/1-ART1
https://doi.org/10.1145/3748327

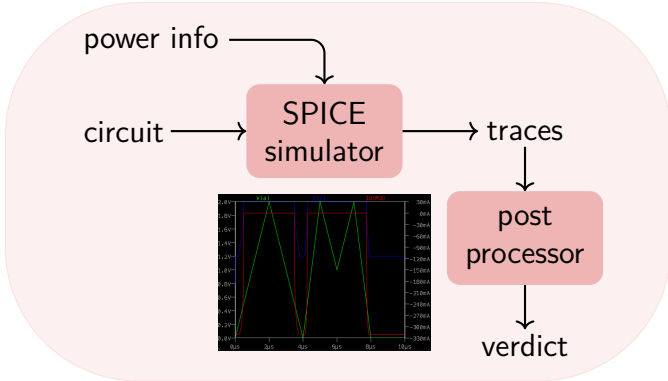

Dynamic approaches
Modeling techniques for electrical rule checking

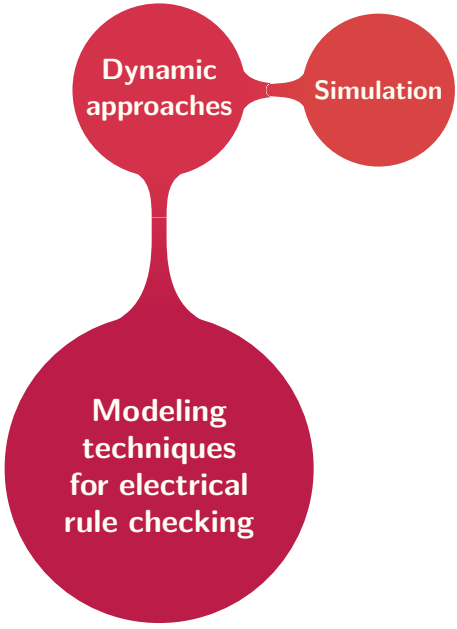

Dynamic approaches
Simulation
Modeling techniques for electrical rule checking

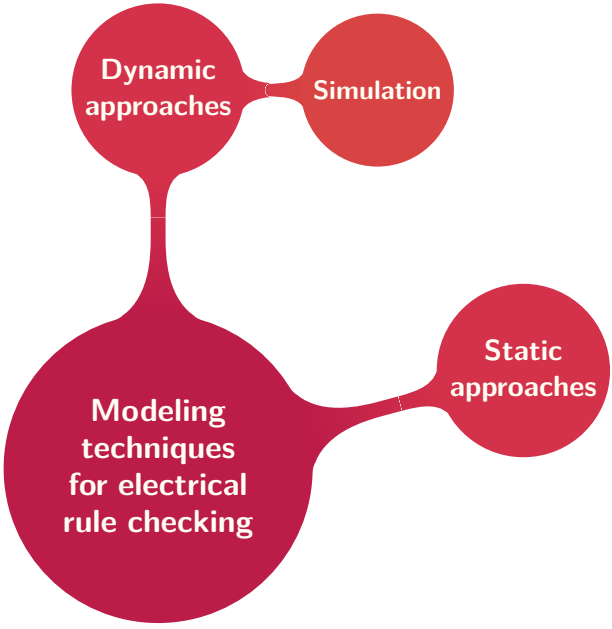
Dynamic approaches

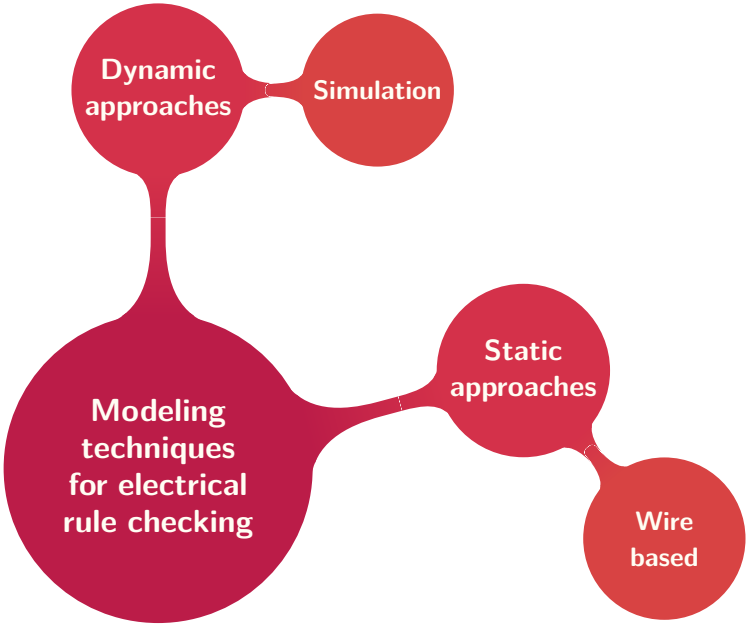
Simulation

Modeling techniques for electrical rule checking




Dynamic approaches
Simulation
Modeling techniques for electrical rule checking


Dynamic approaches
Simulation
Modeling techniques for electrical rule checking
Static approaches


Dynamic approaches
Simulation
Modeling techniques for electrical rule checking
Static approaches
Wire based

Modeling techniques for electrical rule checking

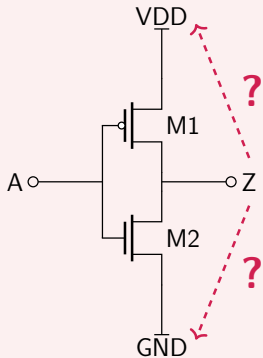
Dynamic approaches

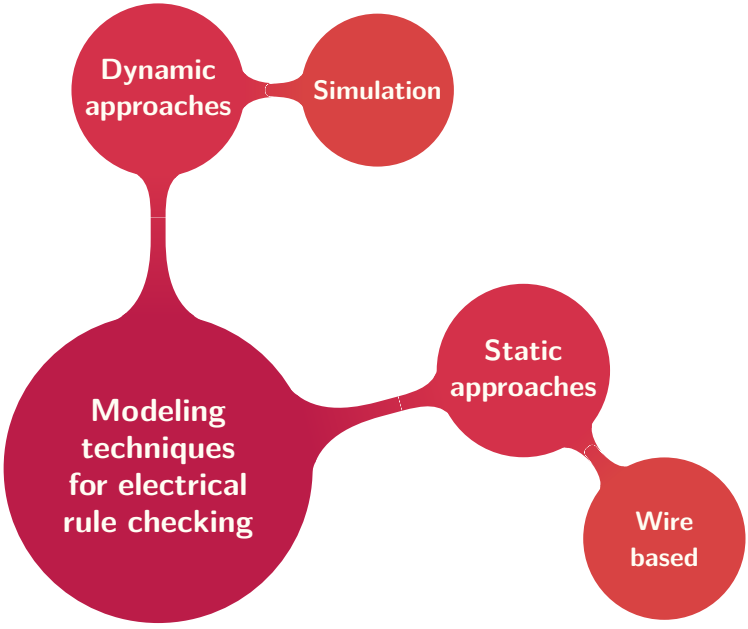
Simulation

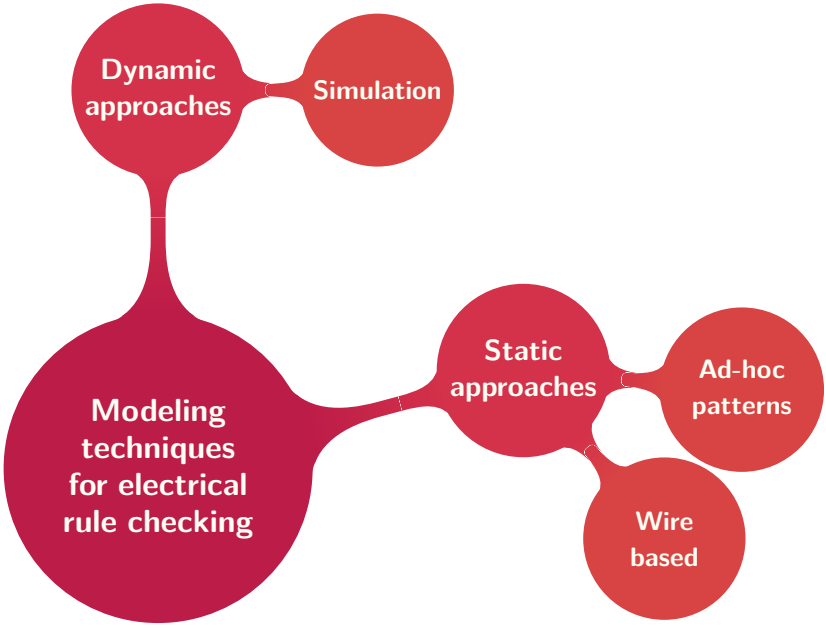
Static approaches

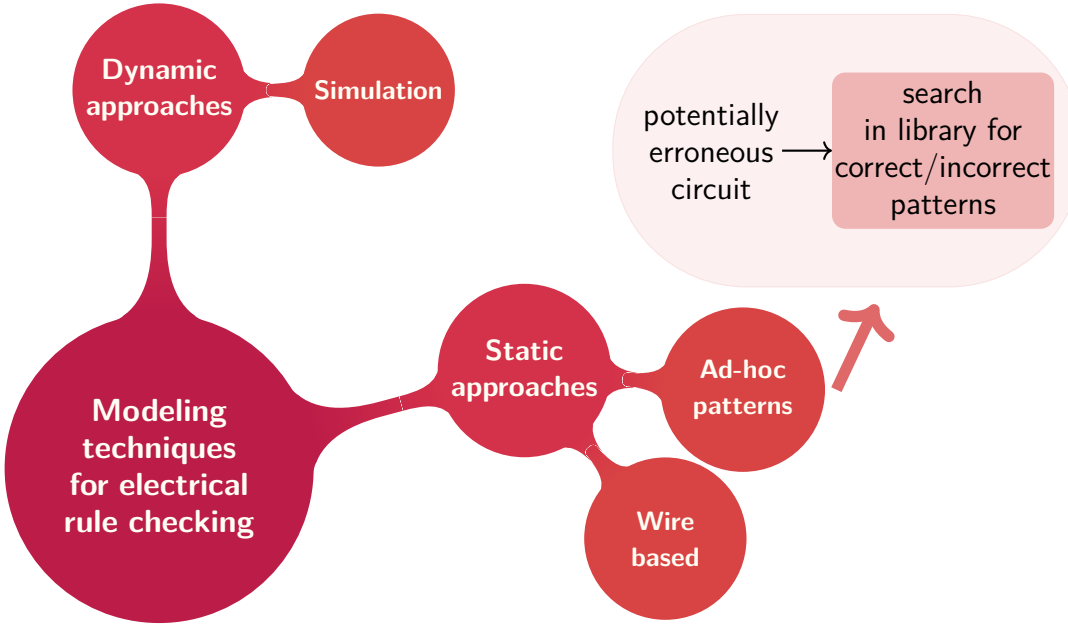
Wire based

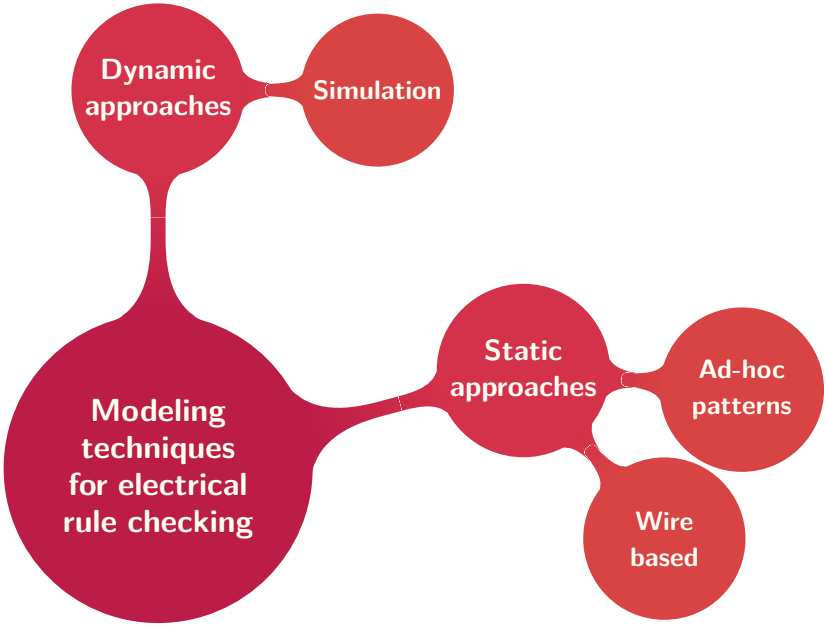
What are the reachable supplies?

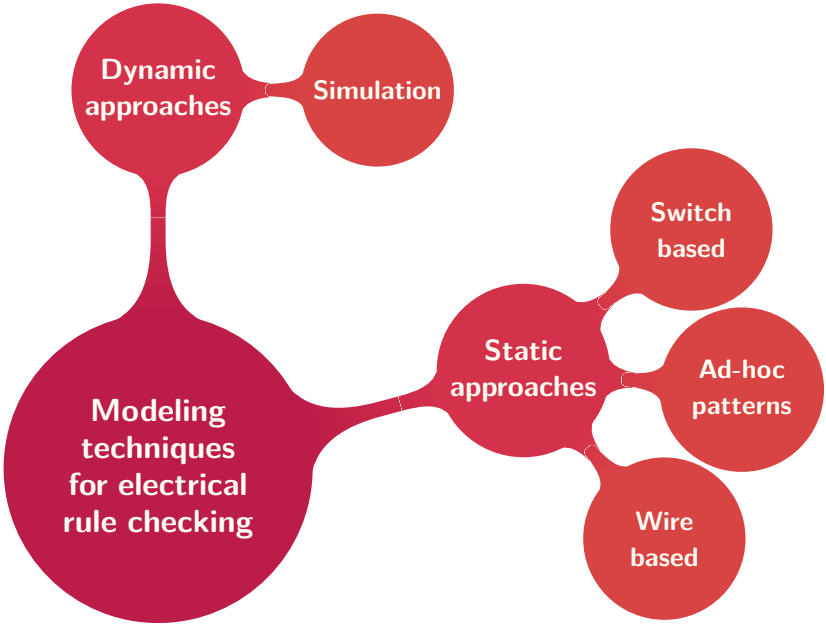


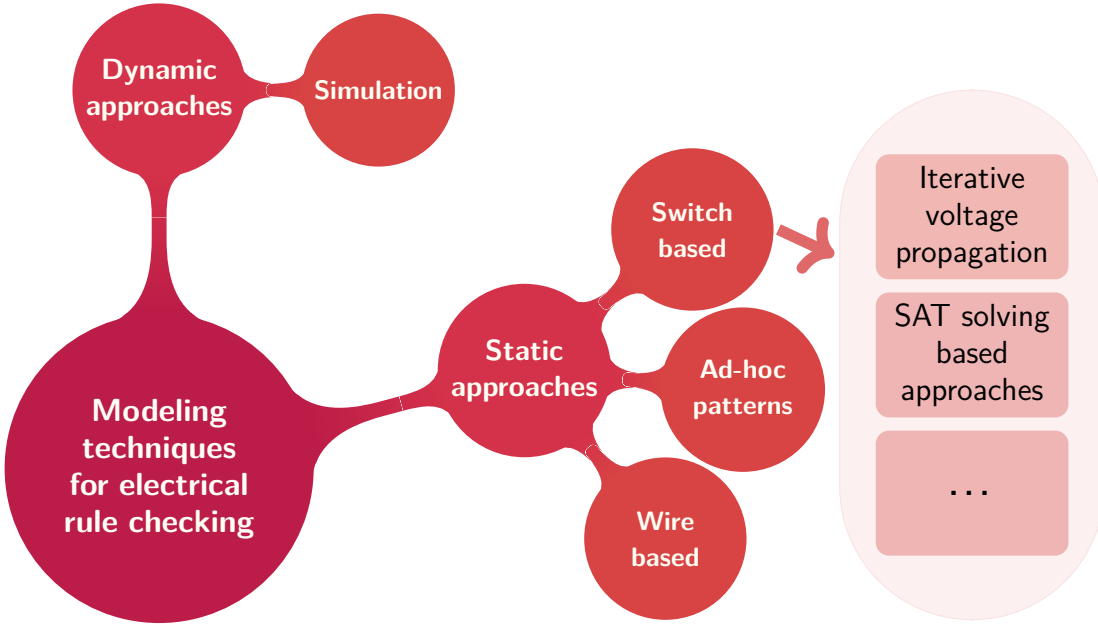

Dynamic approaches
Simulation
Modeling techniques for electrical rule checking
Static approaches
Wire based

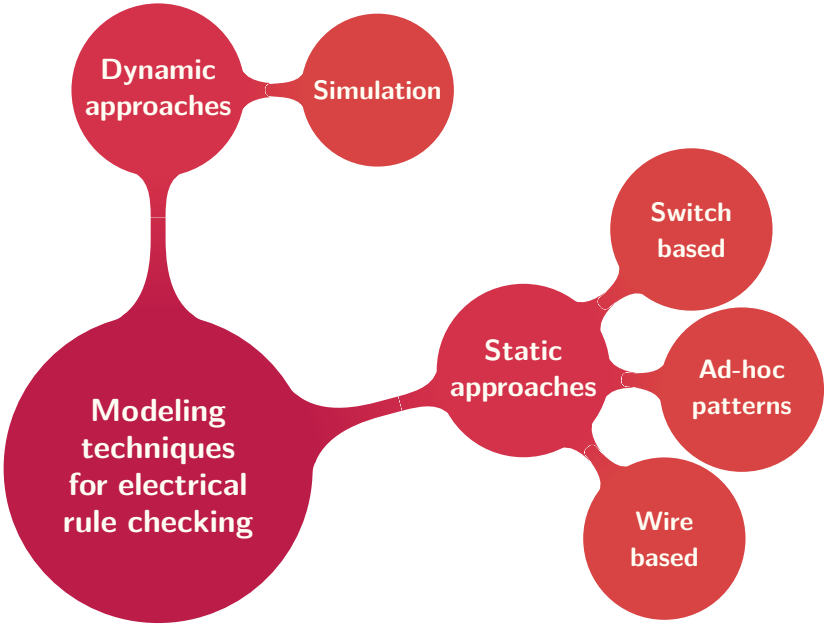

Dynamic approaches
Simulation
Modeling techniques for electrical rule checking
Static approaches
Ad-hoc patterns
Wire based

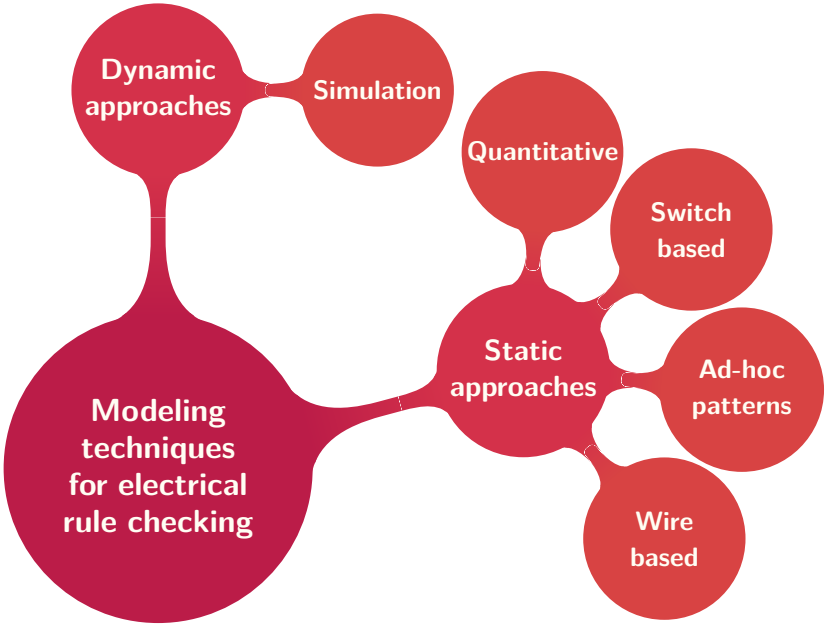

Modeling techniques for electrical rule checking
Dynamic approaches
Simulation
Static approaches
Ad-hoc patterns
Wire based
potentially erroneous circuit
search in library for correct/incorrect patterns

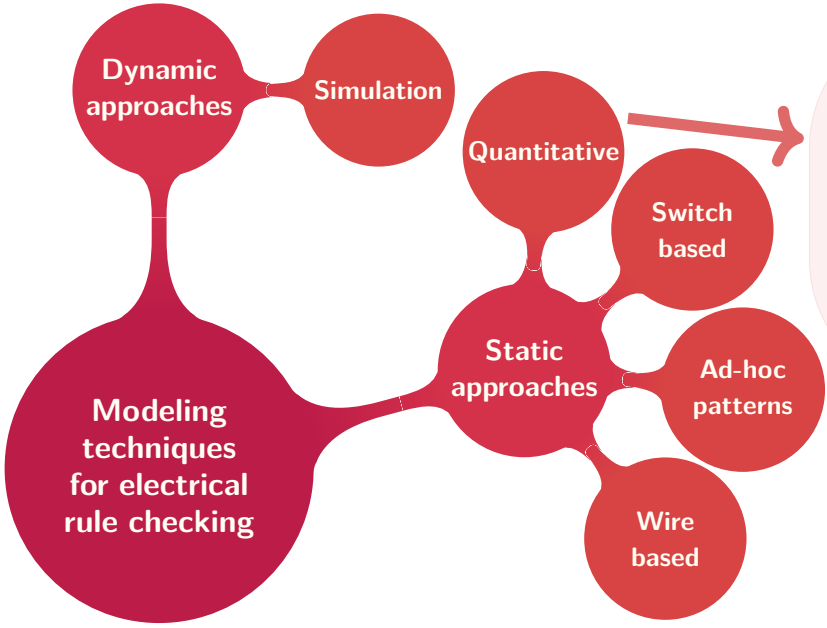

Dynamic approaches
Simulation
Modeling techniques for electrical rule checking
Static approaches
Ad-hoc patterns
Wire based


Dynamic approaches
Simulation
Modeling techniques for electrical rule checking
Static approaches
Switch based
Ad-hoc patterns
Wire based

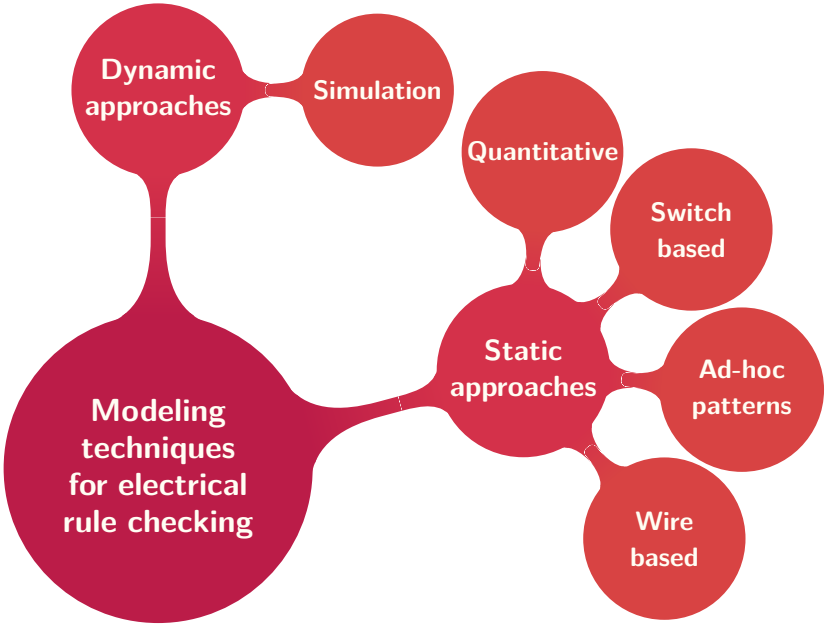

Modeling techniques for electrical rule checking
Dynamic approaches
Simulation
Static approaches
Switch based
Ad-hoc patterns
Wire based
Iterative voltage propagation
SAT solving based approaches
...

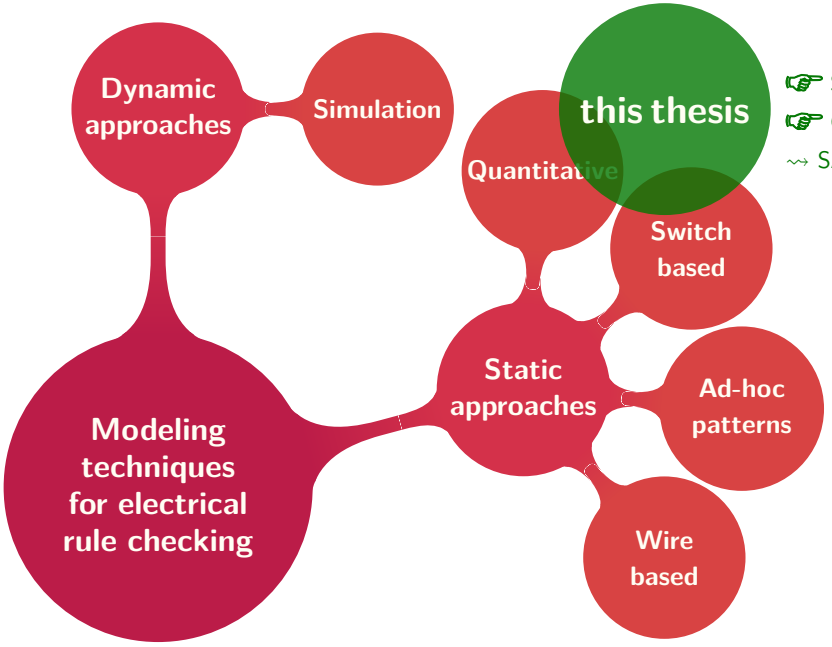

Dynamic approaches
Simulation
Modeling techniques for electrical rule checking
Static approaches
Switch based
Ad-hoc patterns
Wire based


Modeling techniques for electrical rule checking
Dynamic approaches
Simulation
Static approaches
Quantitative
Switch based
Ad-hoc patterns
Wire based



Miller et al. (2013)


Modeling techniques for electrical rule checking
Dynamic approaches
Simulation
Static approaches
Quantitative
Switch based
Ad-hoc patterns
Wire based


Modeling techniques for electrical rule checking
Dynamic approaches
Simulation
Static approaches
Quantitative
this thesis
Switch based
Ad-hoc patterns
Wire based
☞ Switch-based semantics
☞ Quantitative semantics
⇝ SAT/SMT solving

Satisfiability (modulo theories)

SAT

Let $a, b, c \in \{\bot, \top\}$, and
a formula $\phi = (a \vee b) \wedge \neg c$

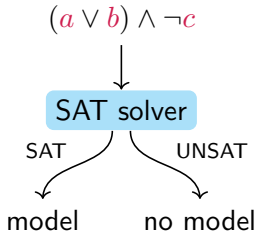
Is there an assignment of a, b, and c,
such that ϕ evaluates to $\top$?

Satisfiability (modulo theories)

SAT

Let $a, b, c \in \{\bot, \top\}$, and
a formula $\phi = (a \vee b) \wedge \neg c$

Is there an assignment of a, b, and c,
such that ϕ evaluates to $\top$?

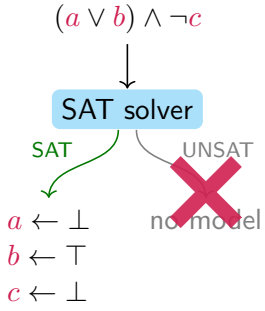


Satisfiability (modulo theories)

SAT

Let $a, b, c \in \{\bot, \top\}$, and
a formula $\phi = (a \vee b) \wedge \neg c$

Is there an assignment of a, b, and c, such that ϕ evaluates to $\top$?

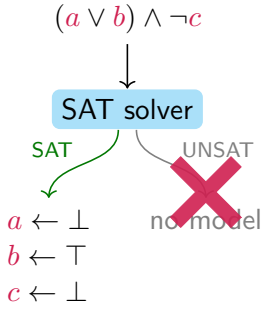


Satisfiability (modulo theories)

SAT

Let $a, b, c \in \{\bot, \top\}$, and
a formula $\phi = (a \vee b) \wedge \neg c$

Is there an assignment of a, b, and c, such that ϕ evaluates to $\top$?



SMT

Let $a \in \{\bot, \top\}$, and $x, y \in \mathbb{Q}$, and
a formula $\phi = (x + y \leq \frac{1}{2}) \wedge (x < y \vee a)$

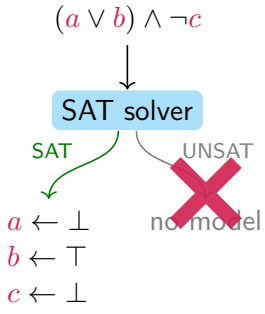
Is there an assignment of a, x, and y, such that ϕ evaluates to $\top$?

Satisfiability (modulo theories)

SAT

Let $a, b, c \in \{\bot, \top\}$, and
a formula $\phi = (a \vee b) \wedge \neg c$

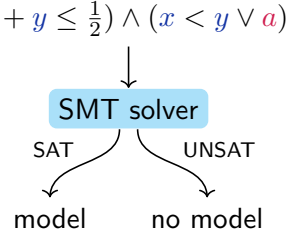
Is there an assignment of a, b, and c, such that ϕ evaluates to $\top$?



SMT

Let $a \in \{\bot, \top\}$, and $x, y \in \mathbb{Q}$, and
a formula $\phi = (x + y \leq \frac{1}{2}) \wedge (x < y \vee a)$

Is there an assignment of a, x, and y, such that ϕ evaluates to $\top$?

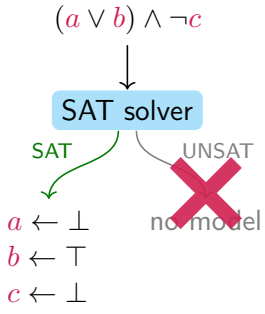


Satisfiability (modulo theories)

SAT

Let $a, b, c \in \{\bot, \top\}$, and
a formula $\phi = (a \vee b) \wedge \neg c$

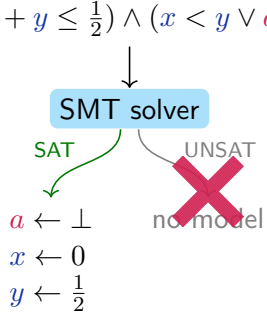
Is there an assignment of a, b, and c, such that ϕ evaluates to $\top$?



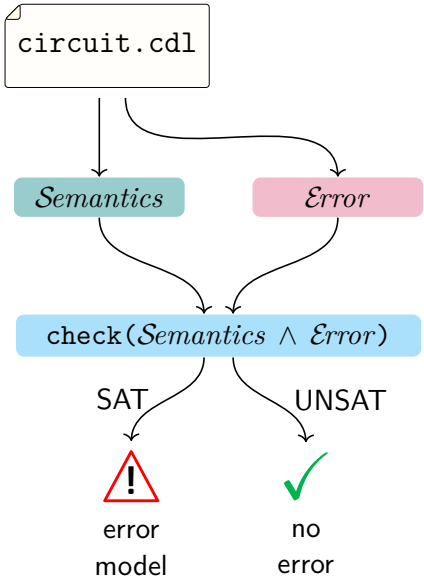
SMT

Let $a \in \{\bot, \top\}$, and $x, y \in \mathbb{Q}$, and
a formula $\phi = (x + y \leq \frac{1}{2}) \wedge (x < y \vee a)$

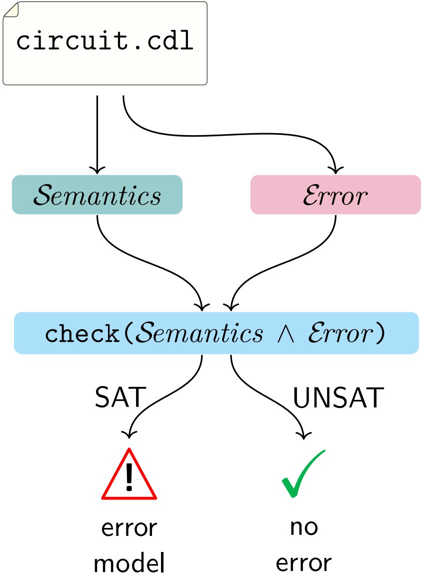
Is there an assignment of a, x, and y, such that ϕ evaluates to $\top$?

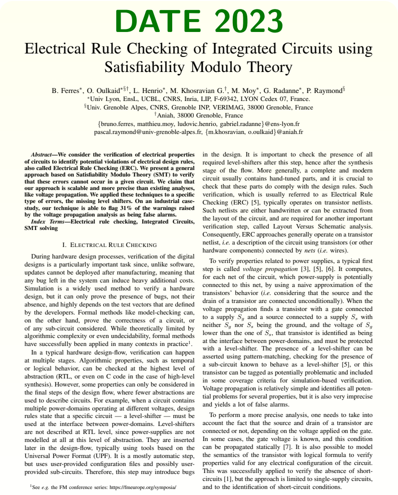


My circuit verification framework



My circuit verification framework





DATE 2023

Electrical Rule Checking of Integrated Circuits using Satisfiability Modulo Theory

B. Ferres*, O. Oulkaid*§†, L. Henrio*, M. Khosravian G.†, M. Moy*, G. Radanne*, P. Raymond§
*Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France.
§Univ. Grenoble Alpes, CNRS, Grenoble INP, VERIMAG, 38000 Grenoble, France
†Aniah, 38000 Grenoble, France
{bruno.ferres, matthieu.moy, ludovic.henrio, gabriel.radanne}@ens-lyon.fr
pascal.raymond@univ-grenoble-alpes.fr, {m.khosravian, o.oulkaid}@aniah.fr

***Abstract*—We consider the verification of electrical properties of circuits to identify potential violations of electrical design rules, also called Electrical Rule Checking (ERC). We present a general approach based on Satisfiability Modulo Theory (SMT) to verify that these errors cannot occur in a given circuit. We claim that our approach is scalable and more precise than existing analyses, like voltage propagation. We applied these techniques to a specific type of errors, the missing level shifters. On an industrial case-study, our technique is able to flag 31% of the warnings raised by the voltage propagation analysis as being false alarms.**

***Index Terms*—Electrical rule checking, Integrated Circuits, SMT solving**

I. Electrical Rule Checking

During hardware design processes, verification of the digital designs is a particularly important task since, unlike software, updates cannot be deployed after manufacturing, meaning that any bug left in the system can induce heavy additional costs. Simulation is a widely used method to verify a hardware design, but it can only prove the presence of bugs, not their absence, and highly depends on the test vectors that are defined by the developers. Formal methods like model-checking can, on the other hand, prove the correctness of a circuit, or of any sub-circuit considered. While theoretically limited by algorithmic complexity or even undecidability, formal methods have successfully been applied in many contexts in practice[1].

In a typical hardware design-flow, verification can happen at multiple stages. Algorithmic properties, such as temporal or logical behavior, can be checked at the highest level of abstraction (RTL, or even on C code in the case of high-level synthesis). However, some properties can only be considered in the final steps of the design flow, where fewer abstractions are used to describe circuits. For example, when a circuit contains multiple power-domains operating at different voltages, design rules state that a specific circuit — a level-shifter — must be used at the interface between power-domains. Level-shifters are not described at RTL level, since power-supplies are not modelled at all at this level of abstraction. They are inserted later in the design-flow, typically using tools based on the Universal Power Format (UPF). It is a mostly automatic step, but uses user-provided configuration files and possibly user-provided sub-circuits. Therefore, this step may introduce bugs in the design. It is important to check the presence of all required level-shifters after this step, hence after the synthesis stage of the flow. More generally, a complete and modern circuit usually contains hand-tuned parts, and it is crucial to check that these parts do comply with the design rules. Such verification, which is usually referred to as Electrical Rule Checking (ERC) [5], typically operates on transistor netlists. Such netlists are either handwritten or can be extracted from the layout of the circuit, and are required for another important verification step, called Layout Versus Schematic analysis. Consequently, ERC approaches generally operate on a transistor netlist, *i.e.* a description of the circuit using transistors (or other hardware components) connected by *nets* (*i.e.* wires).

To verify properties related to power supplies, a typical first step is called *voltage propagation* [3], [5], [6]. It computes, for each net of the circuit, which power-supply is potentially connected to this net, by using a naive approximation of the transistors' behavior (*i.e.* considering that the source and the drain of a transistor are connected unconditionally). When the voltage propagation finds a transistor with a gate connected to a supply S_g and a source connected to a supply S_s with neither S_g nor S_s being the ground, and the voltage of S_g lower than the one of S_s, that transistor is identified as being at the interface between power-domains, and must be protected with a level-shifter. The presence of a level-shifter can be asserted using pattern-matching, checking for the presence of a sub-circuit known to behave as a level-shifter [5], or this transistor can be tagged as potentially problematic and included in some coverage criteria for simulation-based verification. Voltage propagation is relatively simple and identifies all potential problems for several properties, but it is also very imprecise and yields a lot of false alarms.

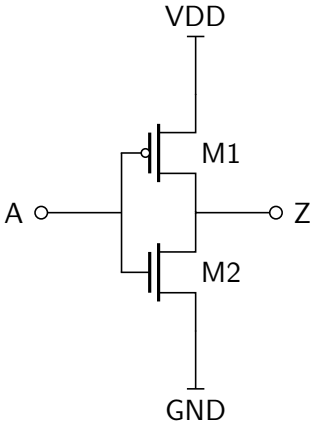
To perform a more precise analysis, one needs to take into account the fact that the source and drain of a transistor are connected or not, depending on the voltage applied on the gate. In some cases, the gate voltage is known, and this condition can be propagated statically [7]. It is also possible to model the semantics of the transistor with logical formula to verify properties valid for any electrical configuration of the circuit. This was successfully applied to verify the absence of short-circuits [1], but the approach is limited to single-supply circuits, and to the identification of short-circuit conditions.

[1]See *e.g.* the FM conference series: https://fmeurope.org/symposia/

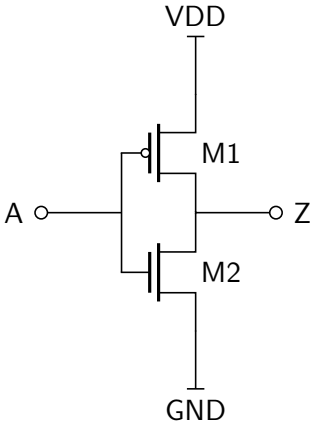
Part 3 of 5

Switch-based Circuit Semantics

Switch-based circuit semantics



Switch-based circuit semantics

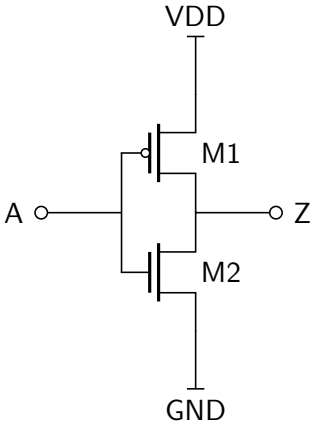


Variables of the SMT formula

$\mathcal{V}$: **Nets** $\longrightarrow$ **Voltages**

$\mathcal{O}n$: **Transistors** $\longrightarrow \{\bot, \top\}$

Switch-based circuit semantics

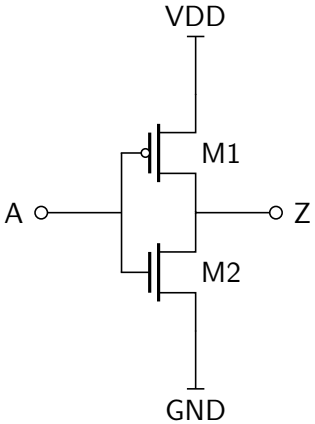


Variables of the SMT formula

$\mathcal{V}$: **Nets** $\longrightarrow$ **Voltages**

$\mathcal{O}n$: **Transistors** $\longrightarrow \{\bot, \top\}$

Switch-based circuit semantics

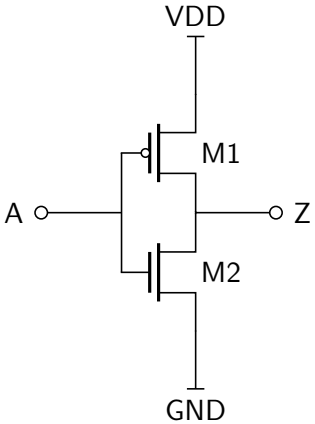


Variables of the SMT formula

$\mathcal{V}$: **Nets** $\longrightarrow$ **Voltages**

$\mathcal{O}n$: **Transistors** $\longrightarrow \{\bot, \top\}$

Switch-based circuit semantics

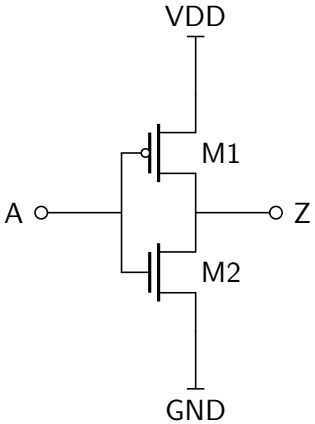


Variables of the SMT formula

$\mathcal{V}$: **Nets** $\longrightarrow$ **Voltages**

$\mathcal{O}n$: **Transistors** $\longrightarrow \{\bot, \top\}$

Switch-based circuit semantics



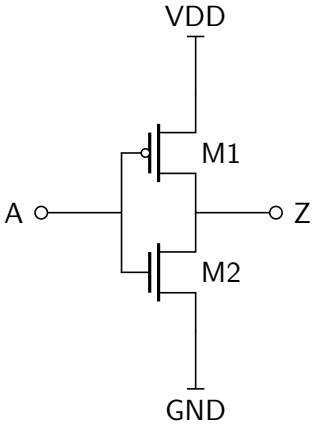
Variables of the SMT formula

$\mathcal{V}$: **Nets** $\longrightarrow$ **Voltages**

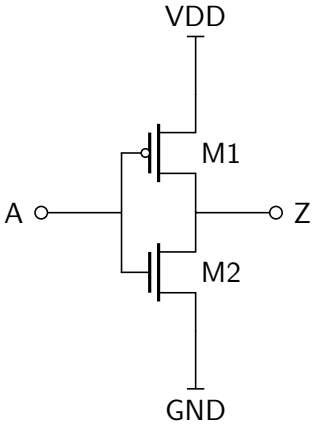
$\mathcal{O}n$: **Transistors** $\longrightarrow \{\bot, \top\}$

Goal ☞ Define constraints on variables $\mathcal{V}$ and $\mathcal{O}n$

Switch-based circuit semantics



Switch-based circuit semantics

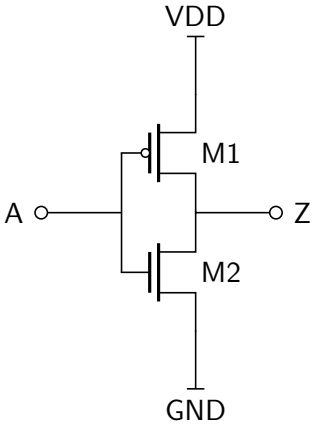


Transistor states

for **p**MOS devices:

$$\mathcal{O}n(\text{device}) \stackrel{\text{def}}{=} \begin{pmatrix} & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{source}) - V_{th} \\ \vee & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{drain}) - V_{th} \end{pmatrix}$$

Switch-based circuit semantics

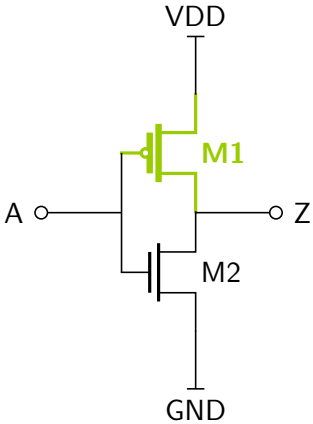


Transistor states

for **p**MOS devices:

$$\mathcal{O}n(\text{device}) \stackrel{\text{def}}{=\!=} \begin{pmatrix} & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{source}) - V_{th} \\ \vee & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{drain}) - V_{th} \end{pmatrix} \left(\mathcal{R}^{\texttt{pMOS}}_{\mathcal{O}n}\right)$$

Switch-based circuit semantics

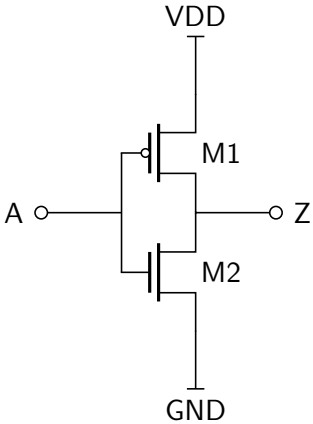


Transistor states

for **p**MOS devices:

$$\mathcal{O}n(\mathsf{M1}) \stackrel{\text{def}}{=} \begin{pmatrix} \mathcal{V}(\mathsf{A}) < \mathcal{V}(\mathsf{VDD}) - V_{th} \\ \vee \quad \mathcal{V}(\mathsf{A}) < \mathcal{V}(\mathsf{Z}) - V_{th} \end{pmatrix} \left(\mathcal{R}^{\mathtt{pMOS}}_{\mathcal{O}n}\right)$$

Switch-based circuit semantics

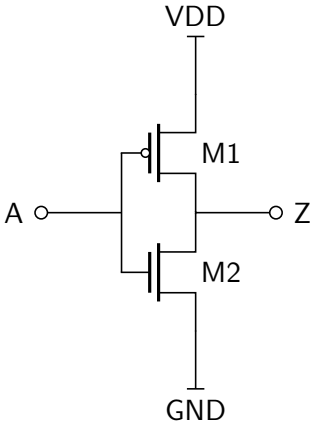


Transistor states

for **p**MOS devices:

$$\mathcal{O}n(\text{device}) \stackrel{\text{def}}{=\!=} \begin{pmatrix} & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{source}) - V_{th} \\ \vee & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{drain}) - V_{th} \end{pmatrix} \left(\mathcal{R}^{\texttt{pMOS}}_{\mathcal{O}n}\right)$$

Switch-based circuit semantics



Transistor states

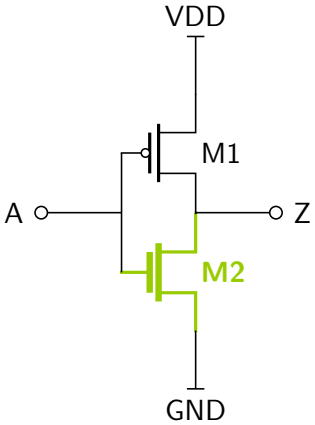
for **p**MOS devices:

$$\mathcal{O}n(\text{device}) \stackrel{\text{def}}{=\!=} \begin{pmatrix} & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{source}) - V_{th} \\ \vee & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{drain}) - V_{th} \end{pmatrix} \left(\mathcal{R}^{\texttt{pMOS}}_{\mathcal{O}n}\right)$$

for **n**MOS devices:

$$\mathcal{O}n(\text{device}) \stackrel{\text{def}}{=\!=} \begin{pmatrix} & \mathcal{V}(\text{gate}) > \mathcal{V}(\text{source}) + V_{th} \\ \vee & \mathcal{V}(\text{gate}) > \mathcal{V}(\text{drain}) + V_{th} \end{pmatrix} \left(\mathcal{R}^{\texttt{nMOS}}_{\mathcal{O}n}\right)$$

Switch-based circuit semantics



Transistor states

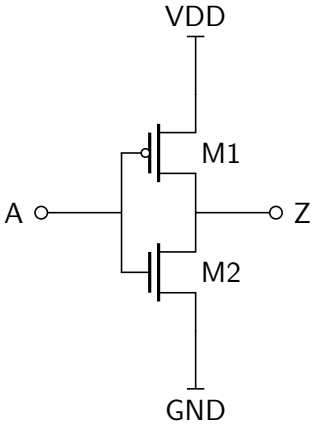
for **p**MOS devices:

$$\mathcal{O}n(\text{device}) \stackrel{\text{def}}{=\!=} \begin{pmatrix} & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{source}) - V_{th} \\ \vee & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{drain}) - V_{th} \end{pmatrix} \left(\mathcal{R}^{\texttt{pMOS}}_{\mathcal{O}n}\right)$$

for **n**MOS devices:

$$\mathcal{O}n(\text{M2}) \stackrel{\text{def}}{=\!=} \begin{pmatrix} & \mathcal{V}(\text{A}) > \mathcal{V}(\text{GND}) + V_{th} \\ \vee & \mathcal{V}(\text{A}) > \mathcal{V}(\text{Z}) + V_{th} \end{pmatrix} \left(\mathcal{R}^{\texttt{nMOS}}_{\mathcal{O}n}\right)$$

Switch-based circuit semantics



Transistor states

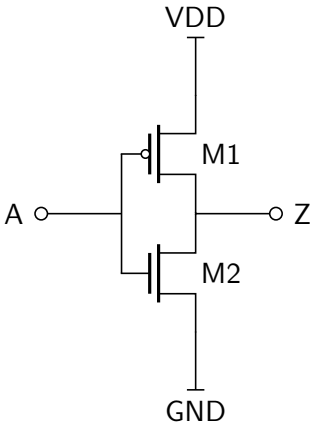
for **p**MOS devices:

$$\mathcal{O}n(\text{device}) \stackrel{\text{def}}{=\!=} \begin{pmatrix} & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{source}) - V_{th} \\ \vee & \mathcal{V}(\text{gate}) < \mathcal{V}(\text{drain}) - V_{th} \end{pmatrix} \left(\mathcal{R}^{\texttt{pMOS}}_{\mathcal{O}n}\right)$$

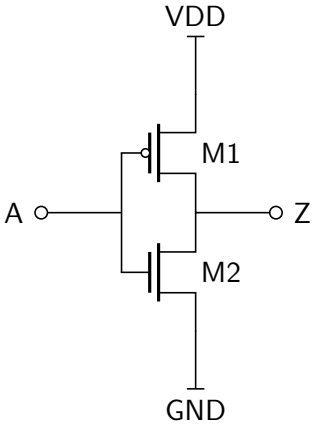
for **n**MOS devices:

$$\mathcal{O}n(\text{device}) \stackrel{\text{def}}{=\!=} \begin{pmatrix} & \mathcal{V}(\text{gate}) > \mathcal{V}(\text{source}) + V_{th} \\ \vee & \mathcal{V}(\text{gate}) > \mathcal{V}(\text{drain}) + V_{th} \end{pmatrix} \left(\mathcal{R}^{\texttt{nMOS}}_{\mathcal{O}n}\right)$$

Switch-based circuit semantics



Switch-based circuit semantics

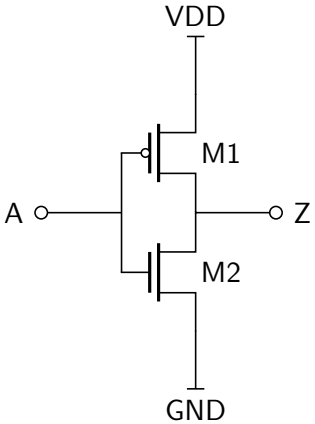


Net neighbors

Net a reaches net b through some device M:

$\textsc{Neighbors}(a) = \{a \xrightarrow{\mathsf{M}} b, \dots\}$

Switch-based circuit semantics



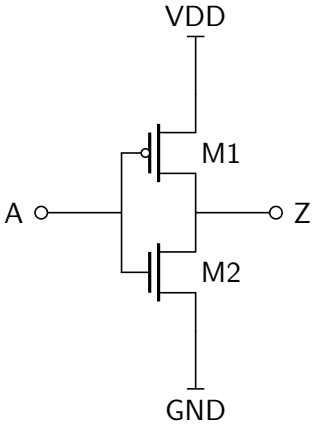
Net neighbors

Net a reaches net b through some device M:

$\textsc{Neighbors}(a) = \{a \xrightarrow{\mathsf{M}} b, \dots\}$

$\textsc{Neighbors}(\mathsf{A}) = \emptyset$

Switch-based circuit semantics



Net neighbors

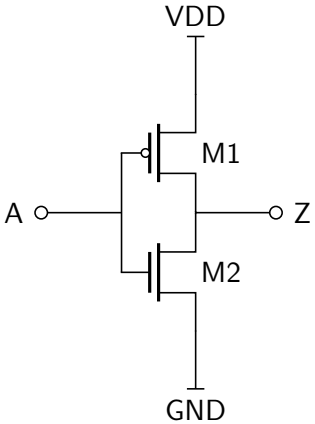
Net a reaches net b through some device M:

$\textsc{Neighbors}(a) = \{a \xrightarrow{\text{M}} b, \dots\}$

$\textsc{Neighbors}(\text{A}) = \emptyset$

$\textsc{Neighbors}(\text{Z}) = \{\text{Z} \xrightarrow{\text{M1}} \text{VDD}, \text{Z} \xrightarrow{\text{M2}} \text{GND}\}$

Switch-based circuit semantics



Net neighbors

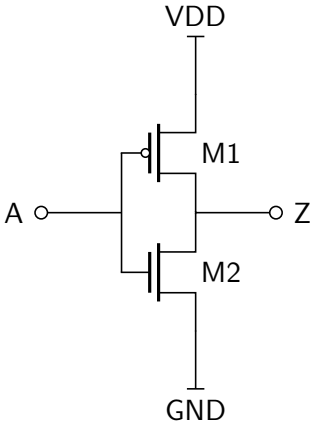
Net a reaches net b through some device M:

$\textsc{Neighbors}(a) = \{a \xrightarrow{\text{M}} b, \dots\}$

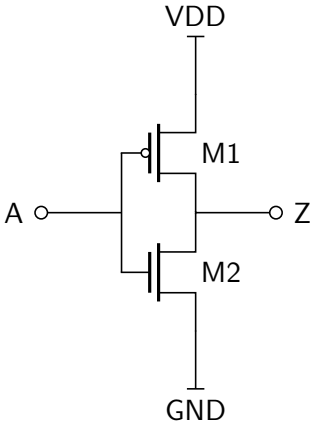
$\textsc{Neighbors}(\text{A}) = \emptyset$

$\textsc{Neighbors}(\text{Z}) = \{\text{Z} \xrightarrow{\text{M1}} \text{VDD}, \text{Z} \xrightarrow{\text{M2}} \text{GND}\}$

Switch-based circuit semantics



Switch-based circuit semantics



Local voltage constraints

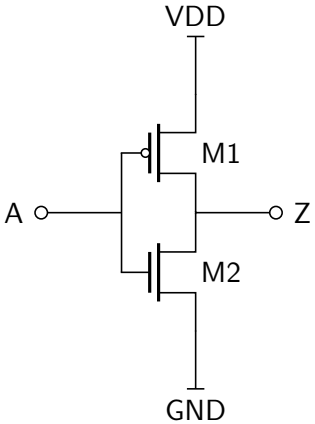
for every net *self*:

current enters *self*

$\Leftrightarrow$

current leaves *self*

Switch-based circuit semantics



Local voltage constraints

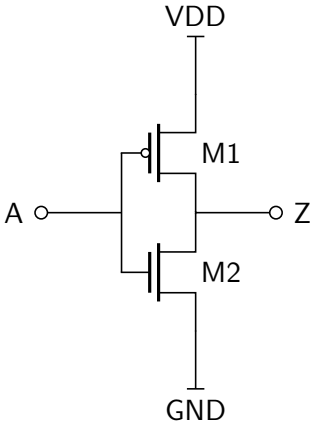
for every net *self*:

$$\exists\ self \xrightarrow{\mathsf{M}} n \in \textsc{Neighbors}(self),$$
current enters *self* **via** n

$$\Leftrightarrow$$

$$\exists\ self \xrightarrow{\mathsf{M}'} n' \in \textsc{Neighbors}(self),$$
current leaves *self* **via** n'

Switch-based circuit semantics



Local voltage constraints

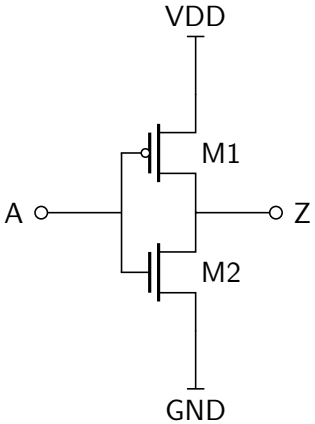
for every net *self*:

$$\exists\ self \xrightarrow{\mathsf{M}} n \in \textsc{Neighbors}(self),\ \mathcal{O}n(\mathsf{M}) \wedge (\mathcal{V}(self) < \mathcal{V}(n))$$

$\Leftrightarrow$

$$\exists\ self \xrightarrow{\mathsf{M}'} n' \in \textsc{Neighbors}(self),\ \textbf{current leaves } self \textbf{ via } n'$$

Switch-based circuit semantics



Local voltage constraints

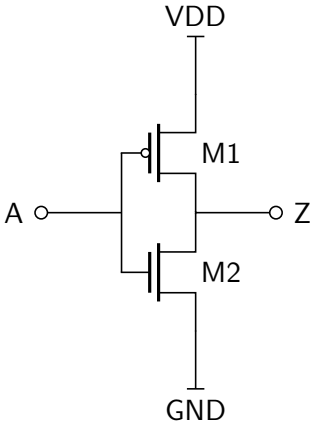
for every net *self*:

$$\exists\ self \xrightarrow{\mathsf{M}} n \in \textsc{Neighbors}(self),\ \mathcal{O}n(\mathsf{M}) \wedge (\mathcal{V}(self) < \mathcal{V}(n))$$

$$\Leftrightarrow$$

$$\exists\ self \xrightarrow{\mathsf{M}'} n' \in \textsc{Neighbors}(self),\ \mathcal{O}n(\mathsf{M}') \wedge (\mathcal{V}(n') < \mathcal{V}(self))$$

Switch-based circuit semantics



Local voltage constraints

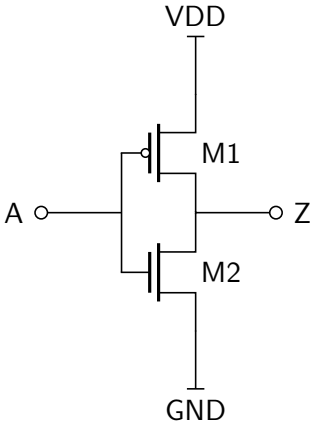
for every net $self$:

$$\bigvee_{self \xrightarrow{\mathsf{M}} n \in \textsc{Neighbors}(self)} \mathcal{O}n(\mathsf{M}) \wedge (\mathcal{V}(self) < \mathcal{V}(n))$$

$$\Leftrightarrow$$

$$\exists\ self \xrightarrow{\mathsf{M}'} n' \in \textsc{Neighbors}(self),\ \mathcal{O}n(\mathsf{M}') \wedge (\mathcal{V}(n') < \mathcal{V}(self))$$

Switch-based circuit semantics



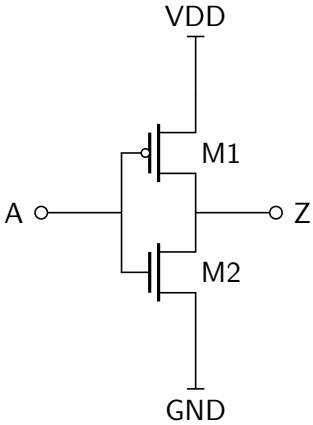
Local voltage constraints

for every net $self$:

$$\bigvee_{self \xrightarrow{\mathsf{M}} n \in \textsc{Neighbors}(self)} \mathcal{O}n(\mathsf{M}) \wedge (\mathcal{V}(self) < \mathcal{V}(n))$$

$$\Leftrightarrow \bigvee_{self \xrightarrow{\mathsf{M}'} n' \in \textsc{Neighbors}(self)} \mathcal{O}n(\mathsf{M}') \wedge (\mathcal{V}(n') < \mathcal{V}(self))$$

Switch-based circuit semantics



Local voltage constraints

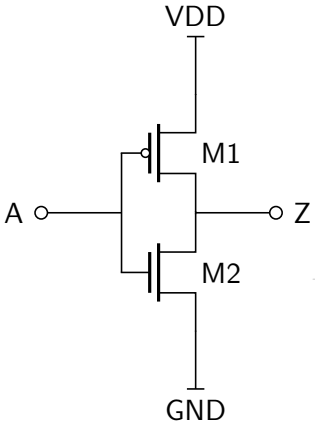
for every net $self$:

$$\bigvee_{self \xrightarrow{\mathsf{M}} n \in \textsc{Neighbors}(self)} \mathcal{O}n(\mathsf{M}) \wedge (\mathcal{V}(self) < \mathcal{V}(n))$$

$$\Leftrightarrow \bigvee_{self \xrightarrow{\mathsf{M}'} n' \in \textsc{Neighbors}(self)} \mathcal{O}n(\mathsf{M}') \wedge (\mathcal{V}(n') < \mathcal{V}(self))$$

$$\left(\mathcal{R}_{\substack{local \\ voltage}}\right)$$

Switch-based circuit semantics



for net Z,

$\mathcal{R}_{\substack{local \\ voltage}}$ is satisfied with

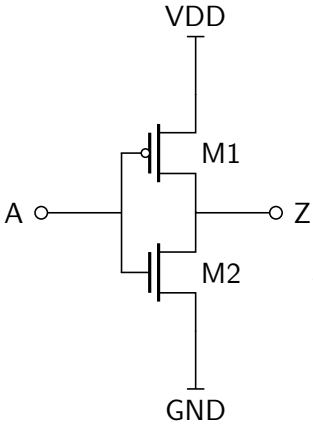
$$\mathcal{O}n(\mathsf{M1}) \wedge \neg\mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{VDD})$$

$$\neg\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{GND})$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{GND}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{VDD})$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{VDD}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{GND})$$

Switch-based circuit semantics



for net Z,

$\mathcal{R}_{\substack{local \\ voltage}}$ is satisfied with

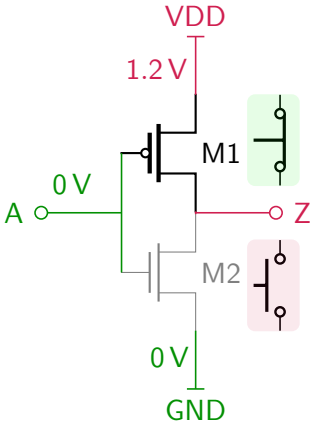
$$\mathcal{O}n(\text{M1}) \wedge \neg\mathcal{O}n(\text{M2}) \wedge \mathcal{V}(\text{Z}) = \mathcal{V}(\text{VDD})$$

$$\neg\mathcal{O}n(\text{M1}) \wedge \mathcal{O}n(\text{M2}) \wedge \mathcal{V}(\text{Z}) = \mathcal{V}(\text{GND})$$

$$\mathcal{O}n(\text{M1}) \wedge \mathcal{O}n(\text{M2}) \wedge \mathcal{V}(\text{GND}) < \mathcal{V}(\text{Z}) < \mathcal{V}(\text{VDD})$$

$$\mathcal{O}n(\text{M1}) \wedge \mathcal{O}n(\text{M2}) \wedge \mathcal{V}(\text{VDD}) < \mathcal{V}(\text{Z}) < \mathcal{V}(\text{GND})$$

Switch-based circuit semantics



for net Z,

$\mathcal{R}_{\substack{local \\ voltage}}$ is satisfied with

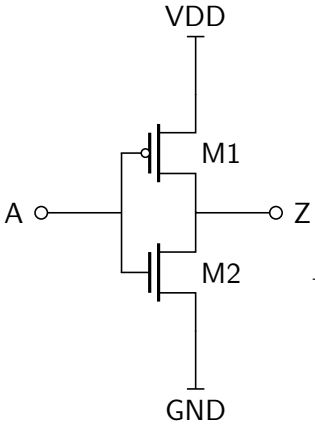
$$\mathcal{O}n(\mathsf{M1}) \wedge \neg\mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{VDD}) \qquad \top$$

$$\neg\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{GND})$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{GND}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{VDD})$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{VDD}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{GND})$$

Switch-based circuit semantics



for net Z,
$\mathcal{R}_{\substack{local \\ voltage}}$ is satisfied with

$$\mathcal{O}n(\mathsf{M1}) \wedge \neg\mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{VDD})$$

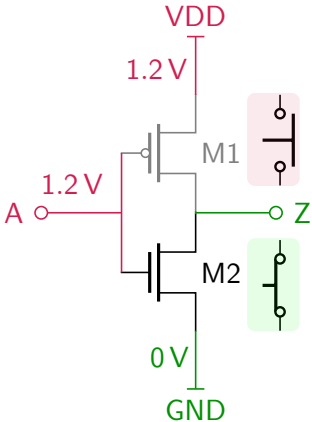
$$\neg\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{GND})$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{GND}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{VDD})$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{VDD}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{GND})$$

$\top$

Switch-based circuit semantics



for net Z,

$\mathcal{R}_{\substack{local \\ voltage}}$ is satisfied with

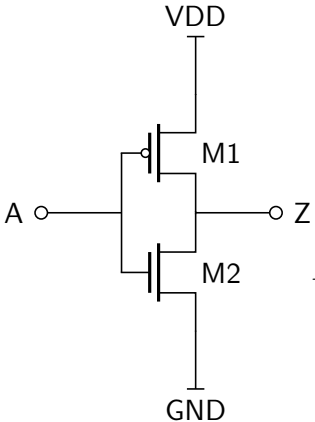
$$\mathcal{O}n(\mathsf{M1}) \wedge \neg\mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{VDD}) \quad \top$$

$$\neg\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{GND}) \quad \top$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{GND}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{VDD})$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{VDD}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{GND})$$

Switch-based circuit semantics



for net Z,

$\mathcal{R}_{\substack{local \\ voltage}}$ is satisfied with

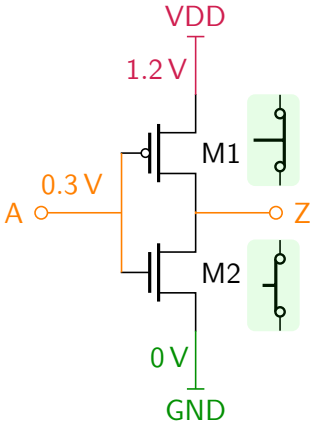
$$\mathcal{O}n(\mathsf{M1}) \wedge \neg\mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{VDD}) \quad \top$$

$$\neg\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{GND}) \quad \top$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{GND}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{VDD})$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{VDD}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{GND})$$

Switch-based circuit semantics



for net Z,

$\mathcal{R}_{\substack{local \\ voltage}}$ is satisfied with

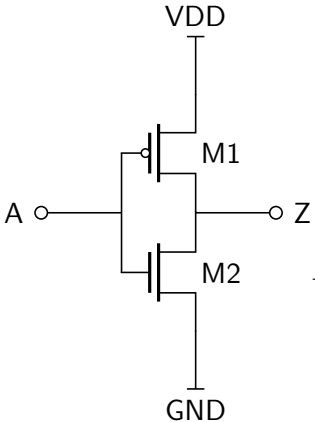
$$\mathcal{O}n(\mathsf{M1}) \wedge \neg\mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{VDD}) \quad \top$$

$$\neg\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{GND}) \quad \top$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{GND}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{VDD}) \quad \top$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{VDD}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{GND})$$

Switch-based circuit semantics



for net Z,

$\mathcal{R}_{\substack{local \\ voltage}}$ is satisfied with

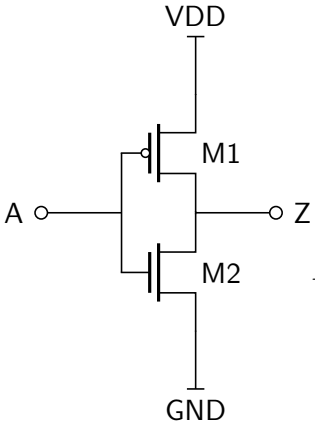
$$\mathcal{O}n(\mathsf{M1}) \wedge \neg\mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{VDD}) \quad \top$$

$$\neg\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{GND}) \quad \top$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{GND}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{VDD}) \quad \top$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{VDD}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{GND})$$

Switch-based circuit semantics



for net Z,

$\mathcal{R}_{\substack{local \\ voltage}}$ is satisfied with

$$\mathcal{O}n(\mathsf{M1}) \wedge \neg\mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{VDD}) \quad \top$$

$$\neg\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{Z}) = \mathcal{V}(\mathsf{GND}) \quad \top$$

$$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{GND}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{VDD}) \quad \top$$

~~$\mathcal{O}n(\mathsf{M1}) \wedge \mathcal{O}n(\mathsf{M2}) \wedge \mathcal{V}(\mathsf{VDD}) < \mathcal{V}(\mathsf{Z}) < \mathcal{V}(\mathsf{GND})$~~ $\bot$

Switch-based circuit semantics

$$
\begin{aligned}
\mathcal{S}^t \quad &\stackrel{\text{def}}{=\!=} \quad \mathcal{R}^{\texttt{pMOS}}_{\mathcal{O}n} \\
&\wedge \quad \mathcal{R}^{\texttt{nMOS}}_{\mathcal{O}n} \\
&\wedge \quad \mathcal{R}_{\substack{local \\ voltage}} \\
&\wedge \quad \mathcal{R}_{supplies} \\
&\wedge \quad \mathcal{R}_{inputs} \\
&\wedge \quad \ldots
\end{aligned}
$$

Switch-based circuit semantics

$$
\begin{aligned}
\mathcal{S}^t \quad \stackrel{\text{def}}{=\!=} \quad & \boxed{\begin{array}{l} \mathcal{R}_{\mathcal{O}n}^{\texttt{pMOS}} \\ \mathcal{R}_{\mathcal{O}n}^{\texttt{nMOS}} \end{array}} \text{ devices} \\
\wedge \quad & \mathcal{R}_{\substack{local \\ voltage}} \\
\wedge \quad & \mathcal{R}_{supplies} \\
\wedge \quad & \mathcal{R}_{inputs} \\
\wedge \quad & \ldots
\end{aligned}
$$

Switch-based circuit semantics

$$
\begin{aligned}
\mathcal{S}^t \quad \stackrel{\text{def}}{=\!=} \quad & \boxed{\begin{array}{l} \mathcal{R}^{\texttt{pMOS}}_{\mathcal{O}n} \\ \mathcal{R}^{\texttt{nMOS}}_{\mathcal{O}n} \end{array}} \text{ devices} \\
\wedge \quad & \boxed{\mathcal{R}_{\substack{local \\ voltage}}} \text{ nets} \\
\wedge \quad & \mathcal{R}_{supplies} \\
\wedge \quad & \mathcal{R}_{inputs} \\
\wedge \quad & \ldots
\end{aligned}
$$

Switch-based circuit semantics

$$
\begin{aligned}
\mathcal{S}^t \quad \stackrel{\text{def}}{=\!=} \quad & \boxed{\begin{array}{l} \mathcal{R}_{\mathcal{O}n}^{\texttt{pMOS}} \\ \mathcal{R}_{\mathcal{O}n}^{\texttt{nMOS}} \end{array}} \text{ devices} \\
\wedge \quad & \boxed{\mathcal{R}_{\substack{local \\ voltage}}} \text{ nets} \\
\wedge \quad & \boxed{\begin{array}{l} \mathcal{R}_{supplies} \\ \mathcal{R}_{inputs} \end{array}} \text{ environment} \\
\wedge \quad & \ldots
\end{aligned}
$$

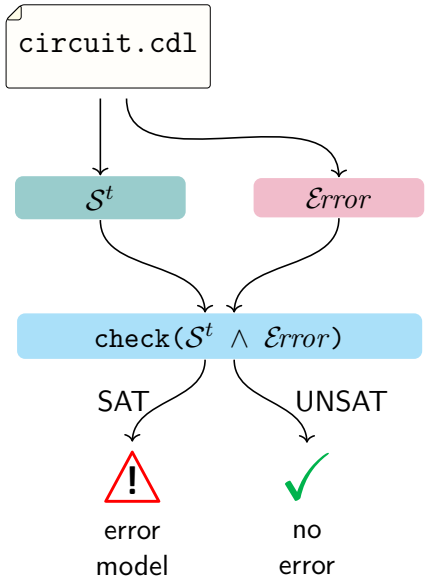
Switch-based circuit semantics

$$\mathcal{S}^t \overset{\text{def}}{=} \mathcal{R}^{\texttt{pMOS}}_{\mathcal{O}n} \wedge \mathcal{R}^{\texttt{nMOS}}_{\mathcal{O}n} \wedge \mathcal{R}_{\substack{local \\ voltage}} \wedge \mathcal{R}_{supplies} \wedge \mathcal{R}_{inputs} \wedge \ldots$$

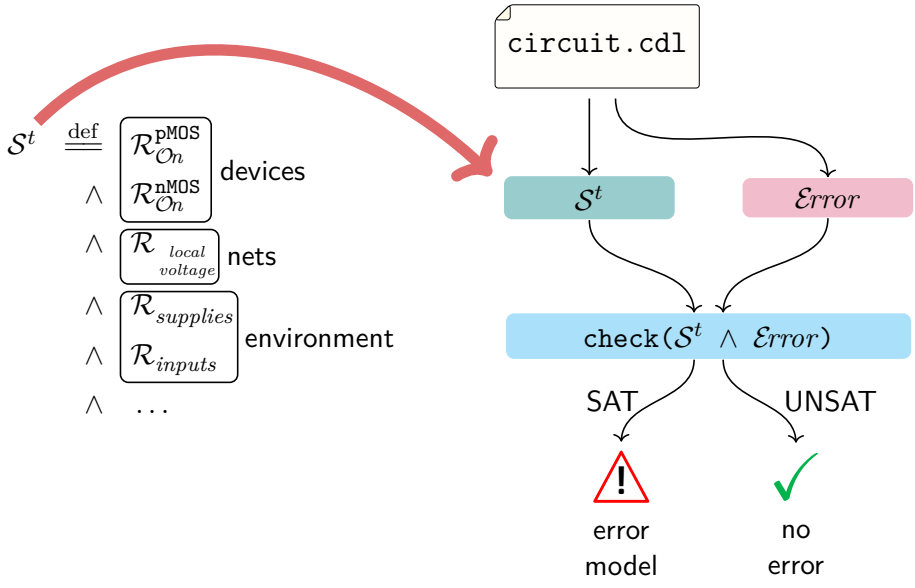
devices

nets

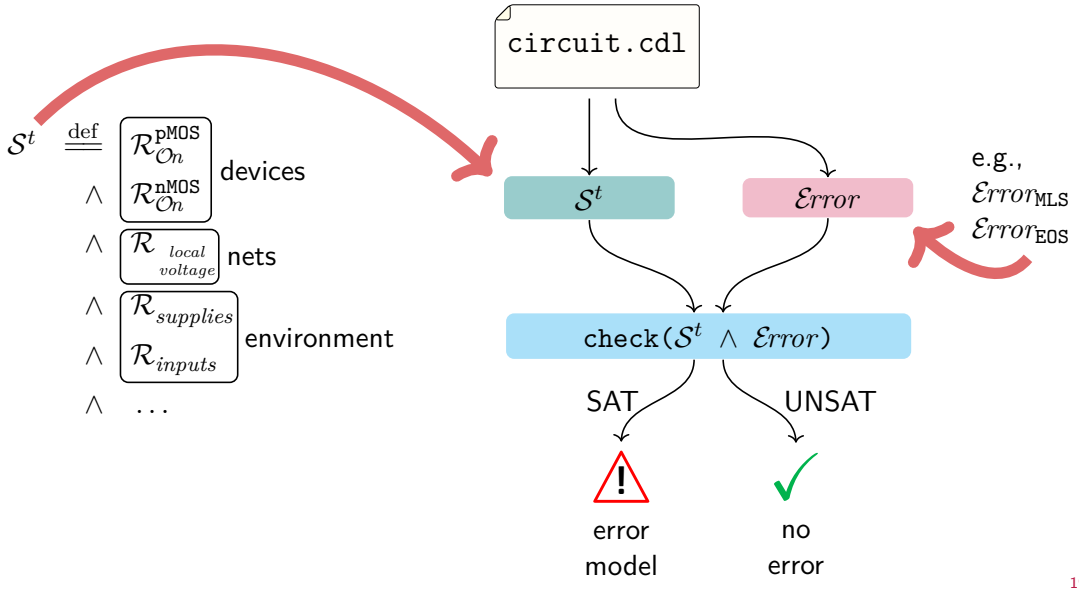
environment



Switch-based circuit semantics

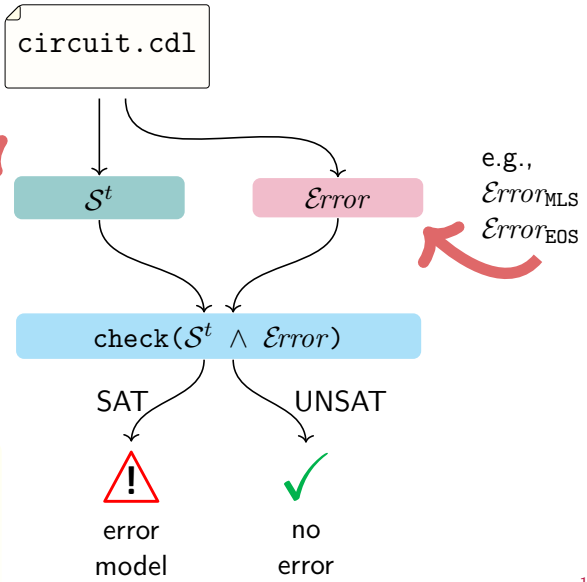


Switch-based circuit semantics



Switch-based circuit semantics

$\mathcal{S}^t \overset{\text{def}}{=} \mathcal{R}_{\mathcal{O}n}^{\text{pMOS}} \wedge \mathcal{R}_{\mathcal{O}n}^{\text{nMOS}}$ devices $\wedge\ \mathcal{R}_{\substack{local\\voltage}}$ nets $\wedge\ \mathcal{R}_{supplies} \wedge \mathcal{R}_{inputs}$ environment $\wedge\ \ldots$



DATE 2024

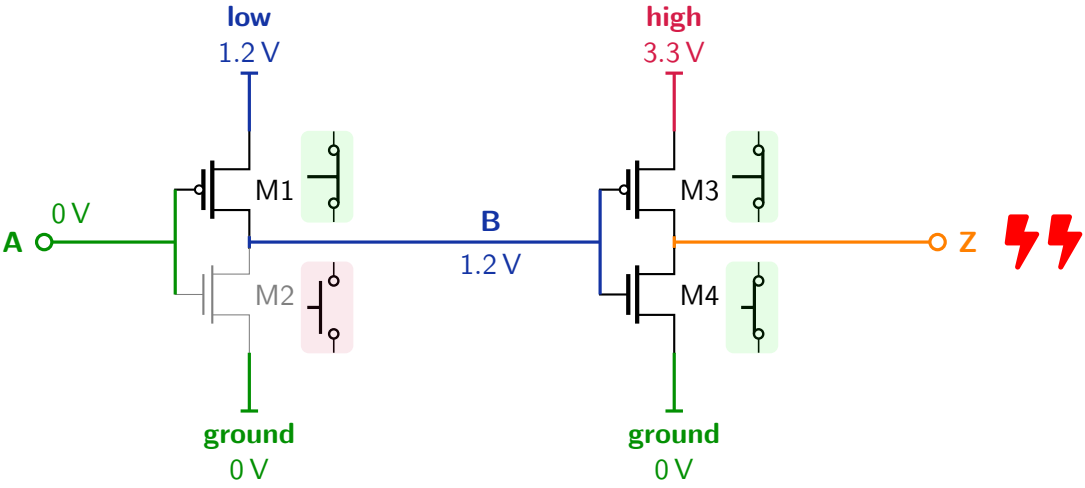
A Transistor Level Relational Semantics for Electrical Rule Checking by SMT Solving

Oussama Oulkaid[§¶], Bruno Ferres[§], Matthieu Moy[§], Pascal Raymond[†], Mehdi Khosravian[¶], Ludovic Henrio[§], Gabriel Radanne[§]

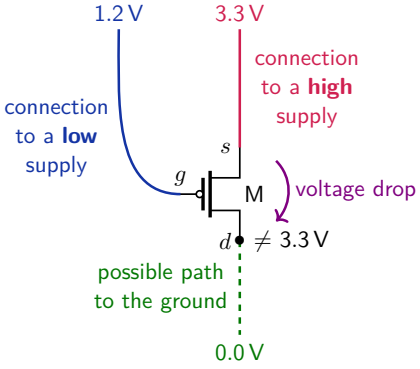
[§]Univ. Lyon, EnsL, UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France
[†]Univ. Grenoble Alpes, CNRS, Grenoble INP*, VERIMAG, 38000 Grenoble, France
[¶]Aniah, 38000 Grenoble, France

Case study: missing level-shifter (MLS)

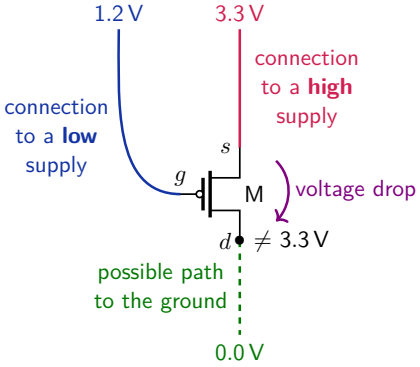
Case study: missing level-shifter (MLS)



Case study: missing level-shifter (MLS)

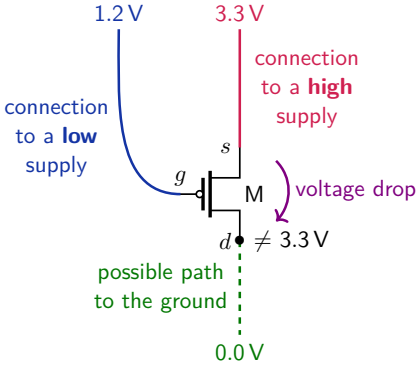


Case study: missing level-shifter (MLS)



$$\mathcal{E}rror_{\texttt{MLS}}(\mathsf{M}) \stackrel{\text{def}}{=} \mathcal{O}n(\mathsf{M}) \wedge \mathcal{V}(g) < \mathcal{V}(s) \wedge \mathcal{V}(g) \neq 0\,\mathsf{V} \wedge \mathcal{V}(s) \neq \mathcal{V}(d)$$

Case study: missing level-shifter (MLS)

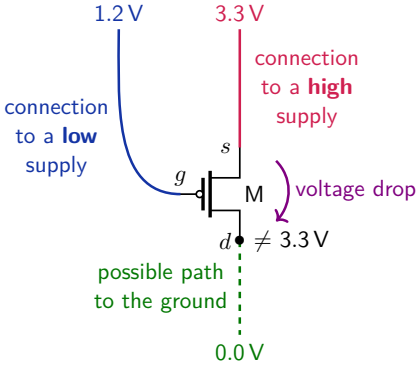


$$\mathcal{E}rror_{\texttt{MLS}}(\mathsf{M}) \stackrel{\text{def}}{=} \mathcal{O}n(\mathsf{M}) \wedge \mathcal{V}(g) < \mathcal{V}(s) \wedge \mathcal{V}(g) \neq 0\,\mathsf{V} \wedge \mathcal{V}(s) \neq \mathcal{V}(d)$$

ADC

Analog-to-digital converter
197 unique subcircuits
20 distinct power supplies
22 598 suspect devices to analyze

Case study: missing level-shifter (MLS)



$$\mathcal{E}rror_{\texttt{MLS}}(\mathsf{M}) \stackrel{\text{def}}{=} \mathcal{O}n(\mathsf{M}) \wedge \mathcal{V}(g) < \mathcal{V}(s) \wedge \mathcal{V}(g) \neq 0\,\mathsf{V} \wedge \mathcal{V}(s) \neq \mathcal{V}(d)$$

ADC

Analog-to-digital converter
197 unique subcircuits
20 distinct power supplies
22 598 suspect devices to analyze

Task

analyze a set of suspect transistors against MLS

Case study: missing level-shifter (MLS)

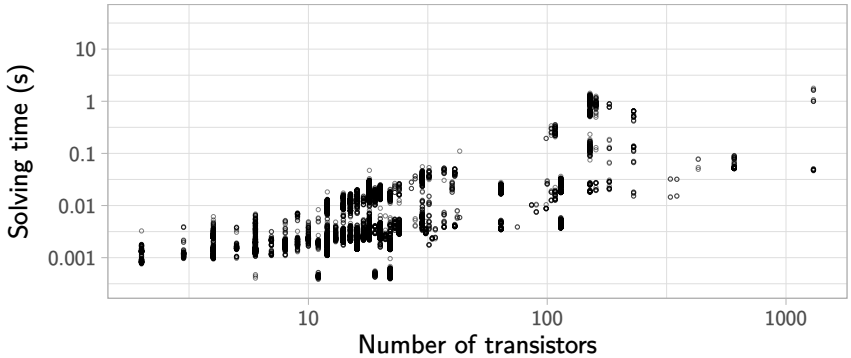
⚠ 10 277 suspects classified as erroneous

✓ 12 321 suspects classified as not erroneous

Case study: missing level-shifter (MLS)

⚠ 10 277 suspects classified as erroneous

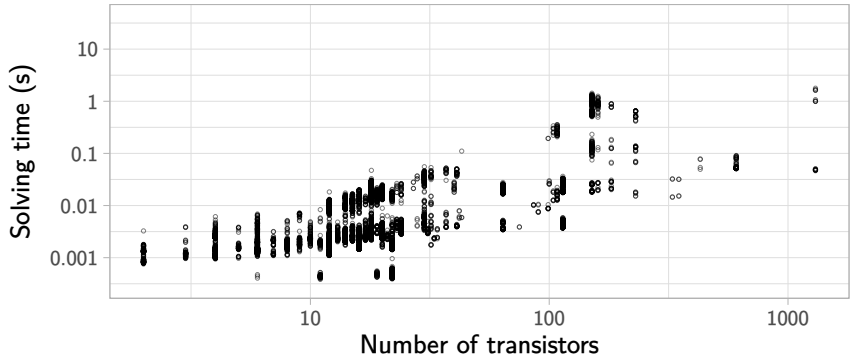
✓ 12 321 suspects classified as not erroneous



Case study: missing level-shifter (MLS)

⚠ 10 277 suspects classified as erroneous

✓ 12 321 suspects classified as not erroneous



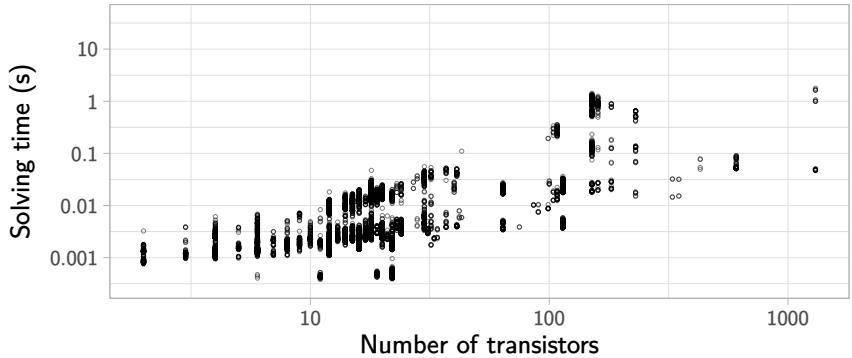
Very efficient for analyzing small-to-medium circuit cells

Case study: missing level-shifter (MLS)

⚠ 10 277 suspects classified as erroneous

✓ 12 321 suspects classified as not erroneous

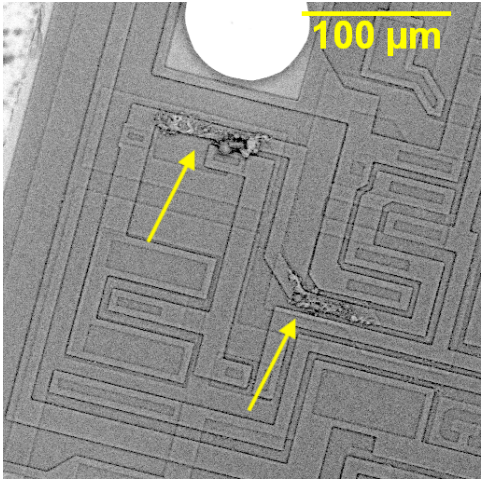
Part of Aniah
ONECHECK



Very efficient for analyzing small-to-medium circuit cells

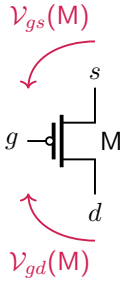
Case study: electrical overstress (EOS)

Case study: electrical overstress (EOS)



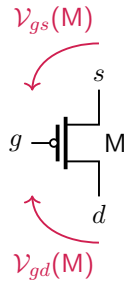
Credit: Ed Hare (2020)

Case study: electrical overstress (EOS)



Is there a too low/high voltage applied on M?

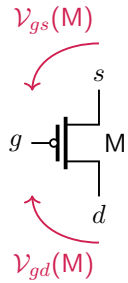
Case study: electrical overstress (EOS)



Is there a too low/high voltage applied on M?

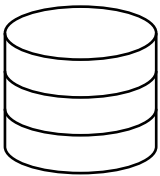
$$\mathcal{E}rror_{\texttt{EOS}}(\mathsf{M}) \stackrel{\text{def}}{=\!=} \mathcal{V}_{gs}(\mathsf{M}) \notin \textsc{allowedInterval}(\mathsf{M}) \\ \vee \quad \mathcal{V}_{gd}(\mathsf{M}) \notin \textsc{allowedInterval}(\mathsf{M})$$

Case study: electrical overstress (EOS)



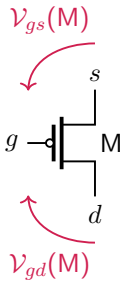
Is there a too low/high voltage applied on M?

$$\mathcal{E}rror_{\texttt{EOS}}(\mathsf{M}) \stackrel{\text{def}}{=\!=} \mathcal{V}_{gs}(\mathsf{M}) \notin \textsc{allowedInterval}(\mathsf{M}) \vee \mathcal{V}_{gd}(\mathsf{M}) \notin \textsc{allowedInterval}(\mathsf{M})$$



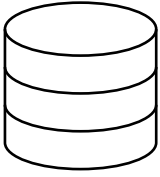
Radio frequency circuitry
549 unique subcircuits
11 distinct power supplies
3144 transistors analyzed

Case study: electrical overstress (EOS)



Is there a too low/high voltage applied on M?

$$\mathcal{E}rror_{\texttt{EOS}}(\mathsf{M}) \stackrel{\text{def}}{=\!=} \mathcal{V}_{gs}(\mathsf{M}) \notin \textsc{allowedInterval}(\mathsf{M}) \vee \mathcal{V}_{gd}(\mathsf{M}) \notin \textsc{allowedInterval}(\mathsf{M})$$



RF

Radio frequency circuitry
549 unique subcircuits
11 distinct power supplies
3144 transistors analyzed

Task

enumerate erroneous devices in each subcircuit

Case study: electrical overstress (EOS)

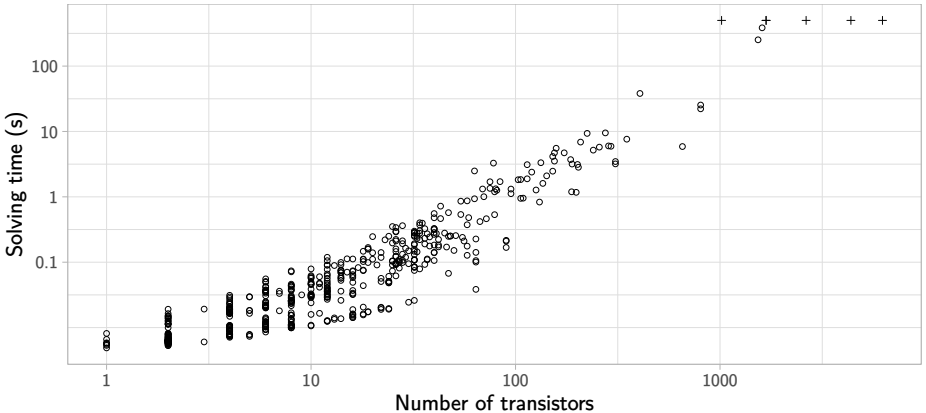
⚠ 2956 errors found

✓ 188 devices classified as not erroneous

Case study: electrical overstress (EOS)

⚠ 2956 errors found

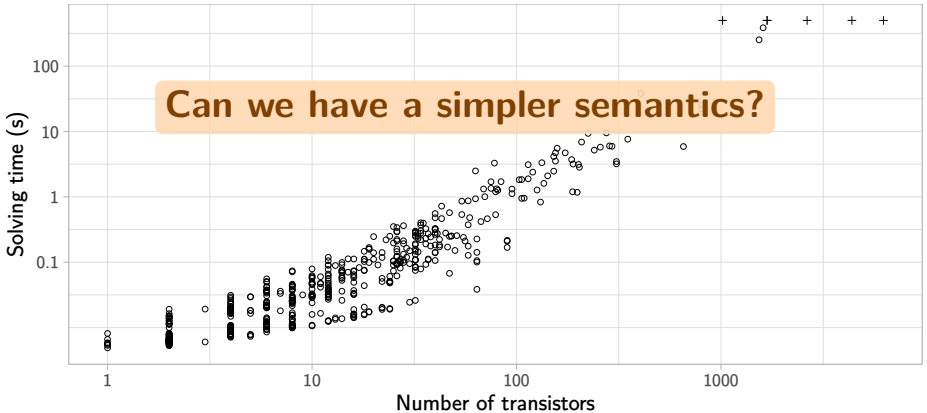
✓ 188 devices classified as not erroneous



Case study: electrical overstress (EOS)

⚠ 2956 errors found

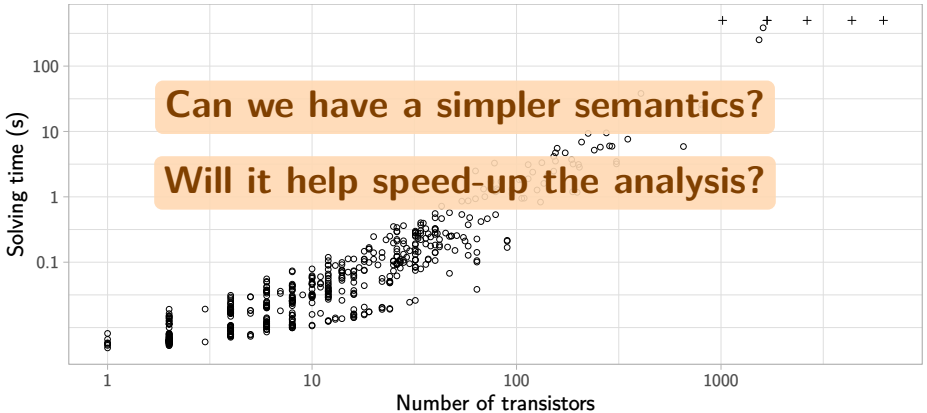
✓ 188 devices classified as not erroneous



Case study: electrical overstress (EOS)

⚠ 2956 errors found

✓ 188 devices classified as not erroneous



A simpler switch-based semantics variant?

A simpler switch-based semantics variant?

for each **n**MOS device M:

$$\mathcal{O}n(\mathsf{M}) \stackrel{\text{def}}{=\!=} \mathcal{V}_g(\mathsf{M}) > \mathcal{V}_s(\mathsf{M}) + V_{th}$$
$$\vee\ \mathcal{V}_g(\mathsf{M}) > \mathcal{V}_d(\mathsf{M}) + V_{th}$$

Semantics $\mathcal{S}^t$

A simpler switch-based semantics variant?

for each **n**MOS device M:

$$\mathcal{O}n(\mathsf{M}) \stackrel{\text{def}}{=\!=} \mathcal{V}_g(\mathsf{M}) > \mathcal{V}_s(\mathsf{M}) + V_{th} \vee \mathcal{V}_g(\mathsf{M}) > \mathcal{V}_d(\mathsf{M}) + V_{th} \qquad \xrightarrow{V_{th} = 0} \qquad \mathcal{O}n(\mathsf{M}) \stackrel{\text{def}}{=\!=} \mathcal{V}_g(\mathsf{M}) > \mathcal{V}_s(\mathsf{M}) \vee \mathcal{V}_g(\mathsf{M}) > \mathcal{V}_d(\mathsf{M})$$

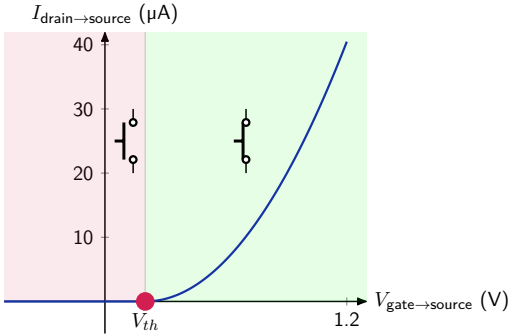
Semantics $\mathcal{S}^t$ Semantics $\mathcal{S}^s$

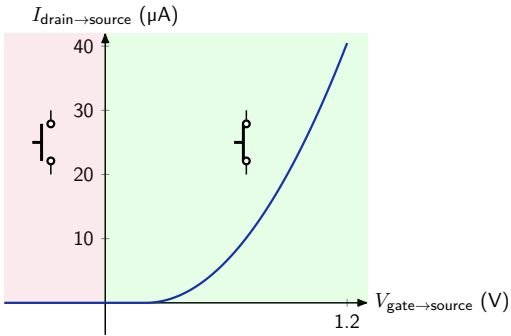
A simpler switch-based semantics variant?

for each **n**MOS device M:

$$\mathcal{O}n(\mathsf{M}) \stackrel{\text{def}}{=\!=} \mathcal{V}_g(\mathsf{M}) > \mathcal{V}_s(\mathsf{M}) + V_{th} \vee \mathcal{V}_g(\mathsf{M}) > \mathcal{V}_d(\mathsf{M}) + V_{th} \quad \xrightarrow{V_{th} = 0} \quad \mathcal{O}n(\mathsf{M}) \stackrel{\text{def}}{=\!=} \mathcal{V}_g(\mathsf{M}) > \mathcal{V}_s(\mathsf{M}) \vee \mathcal{V}_g(\mathsf{M}) > \mathcal{V}_d(\mathsf{M})$$

Semantics $\mathcal{S}^t$ Semantics $\mathcal{S}^s$





$\mathcal{S}^t$ versus $\mathcal{S}^s$ — empirical evaluation

missing level-shifter

$\mathcal{S}^s$	$\mathcal{S}^t$	count
⚠	⚠	9623
⚠	✓	70
✓	⚠	654
✓	✓	12251

$\mathcal{S}^t$ versus $\mathcal{S}^s$ — empirical evaluation

missing level-shifter

$\mathcal{S}^s$	$\mathcal{S}^t$	count
⚠	⚠	9623
⚠	✓	70
✓	⚠	654
✓	✓	12251

false alarms raised by $\mathcal{S}^s$

$\mathcal{S}^t$ versus $\mathcal{S}^s$ — empirical evaluation

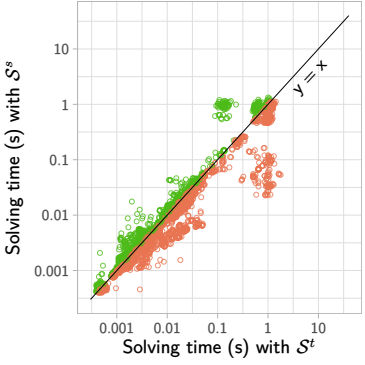
missing level-shifter

$\mathcal{S}^s$	$\mathcal{S}^t$	count	
⚠	⚠	9623	
⚠	✓	70	false alarms raised by $\mathcal{S}^s$
✓	⚠	654	possibly errors missed by $\mathcal{S}^s$
✓	✓	12251	

$\mathcal{S}^t$ versus $\mathcal{S}^s$ — empirical evaluation

missing level-shifter

$\mathcal{S}^s$	$\mathcal{S}^t$	count	
⚠	⚠	9623	
⚠	✓	70	false alarms raised by $\mathcal{S}^s$
✓	⚠	654	possibly errors missed by $\mathcal{S}^s$
✓	✓	12251	



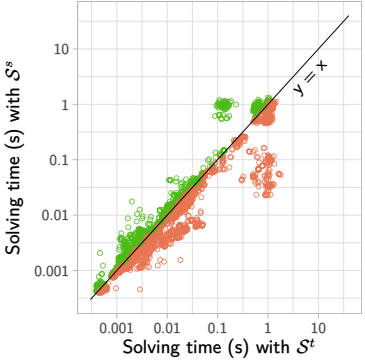
$\mathcal{S}^t$ versus $\mathcal{S}^s$ — empirical evaluation

missing level-shifter

$\mathcal{S}^s$	$\mathcal{S}^t$	count	
⚠	⚠	9623	
⚠	✓	70	false alarms raised by $\mathcal{S}^s$
✓	⚠	654	possibly errors missed by $\mathcal{S}^s$
✓	✓	12251	

electrical overstress

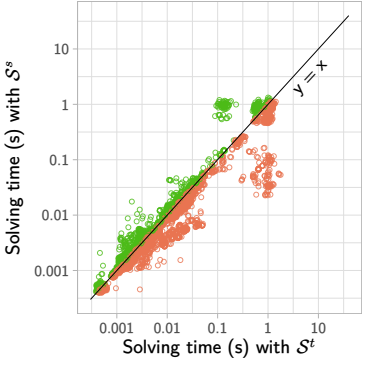
$\mathcal{S}^s$	$\mathcal{S}^t$	count
⚠	⚠	2956
⚠	✓	0
✓	⚠	0
✓	✓	188



$\mathcal{S}^t$ versus $\mathcal{S}^s$ — empirical evaluation

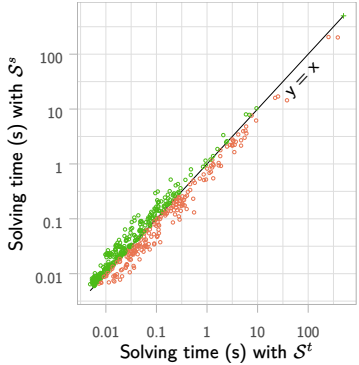
missing level-shifter

$\mathcal{S}^s$	$\mathcal{S}^t$	count	
⚠	⚠	9623	
⚠	✓	70	false alarms raised by $\mathcal{S}^s$
✓	⚠	654	possibly errors missed by $\mathcal{S}^s$
✓	✓	12251	



electrical overstress

$\mathcal{S}^s$	$\mathcal{S}^t$	count
⚠	⚠	2956
⚠	✓	0
✓	⚠	0
✓	✓	188



$\mathcal{S}^t$ versus $\mathcal{S}^s$ — empirical evaluation

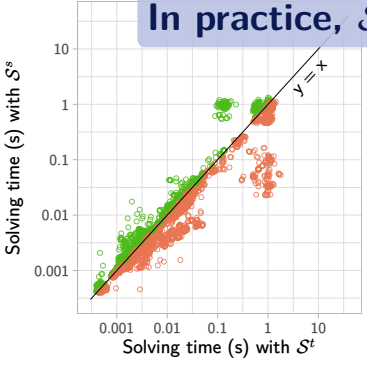
missing level-shifter

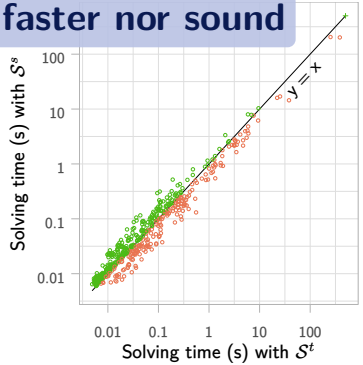
$\mathcal{S}^s$	$\mathcal{S}^t$	count	
⚠	⚠	9623	
⚠	✓	70	false alarms raised by $\mathcal{S}^s$
✓	⚠	654	possibly errors missed by $\mathcal{S}^s$
✓	✓	12251	

electrical overstress

$\mathcal{S}^s$	$\mathcal{S}^t$	count
⚠	⚠	2956
⚠	✓	0
✓	⚠	0
✓	✓	188

In practice, $\mathcal{S}^s$ is neither faster nor sound





$\mathcal{S}^t$ versus $\mathcal{S}^s$ — empirical evaluation

missing level-shifter

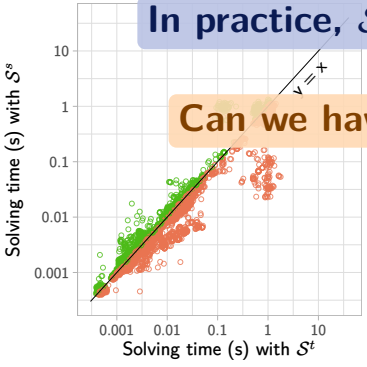
$\mathcal{S}^s$	$\mathcal{S}^t$	count	
⚠	⚠	9623	
⚠	✓	70	false alarms raised by $\mathcal{S}^s$
✓	⚠	654	possibly errors missed by $\mathcal{S}^s$
✓	✓	12251	

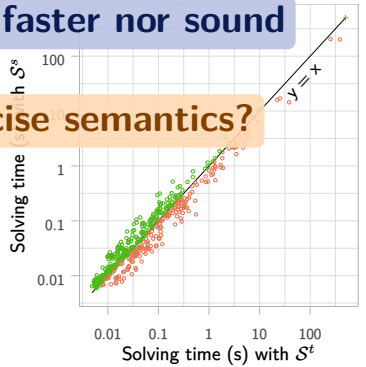
electrical overstress

$\mathcal{S}^s$	$\mathcal{S}^t$	count
⚠	⚠	2956
⚠	✓	0
✓	⚠	0
✓	✓	188

In practice, $\mathcal{S}^s$ is neither faster nor sound

Can we have more precise semantics?





$\mathcal{S}^t$ versus $\mathcal{S}^s$ — empirical evaluation

missing level-shifter

$\mathcal{S}^s$	$\mathcal{S}^t$	count	
⚠	⚠	9623	
⚠	✓	70	false alarms raised by $\mathcal{S}^s$
✓	⚠	654	possibly errors missed by $\mathcal{S}^s$
✓	✓	12251	

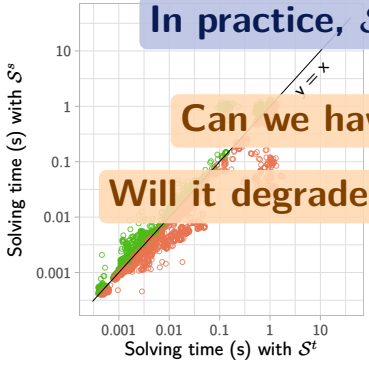
electrical overstress

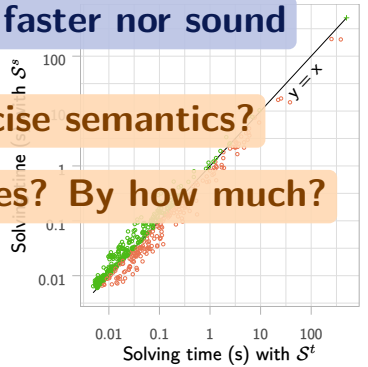
$\mathcal{S}^s$	$\mathcal{S}^t$	count
⚠	⚠	2956
⚠	✓	0
✓	⚠	0
✓	✓	188

In practice, $\mathcal{S}^s$ is neither faster nor sound

Can we have more precise semantics?

Will it degrade performances? By how much?





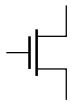
Part 4 of 5

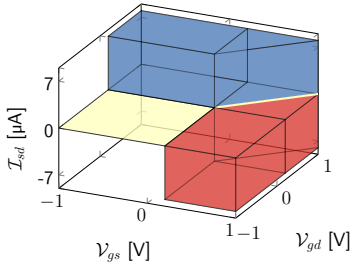
Quantitative Circuit Semantics

The problem with $\mathcal{S}^t$ and $\mathcal{S}^s$

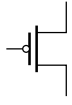
The problem with $\mathcal{S}^t$ and $\mathcal{S}^s$

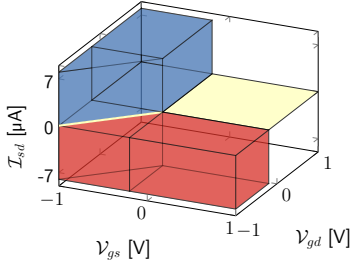
nMOS





pMOS

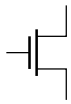


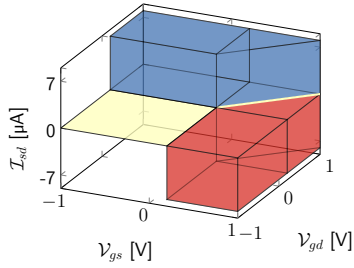


$\mathcal{R}_{\substack{local \\ voltage}}$ allows an interval of voltage drop values

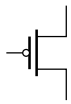
The problem with $\mathcal{S}^t$ and $\mathcal{S}^s$

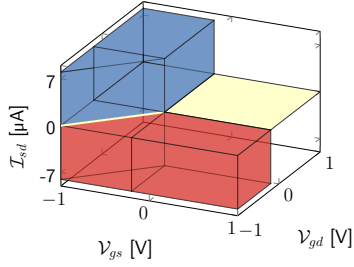
nMOS



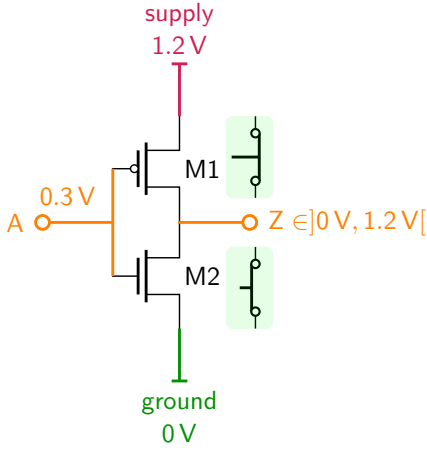


pMOS



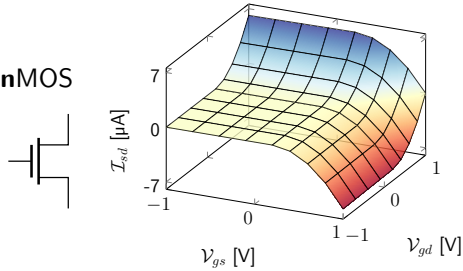


$\mathcal{R}_{\substack{local\\voltage}}$ allows an interval of voltage drop values

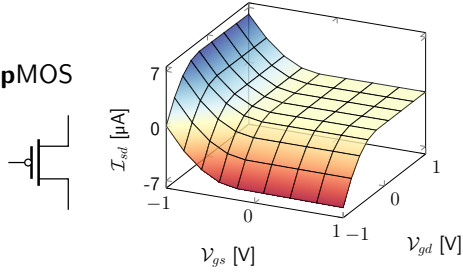


Transistor's regions of operation: I-V characteristics

nMOS

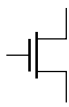


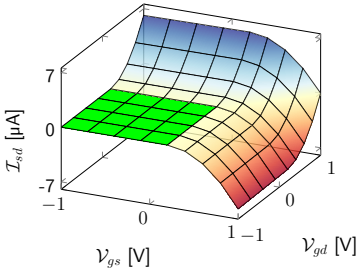
pMOS



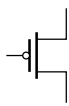
Transistor's regions of operation: I-V characteristics

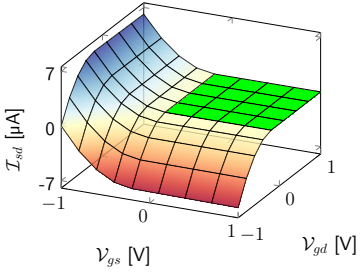
nMOS





pMOS



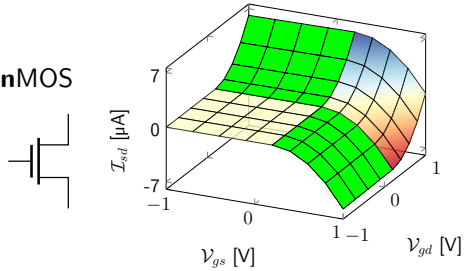


Cut-off

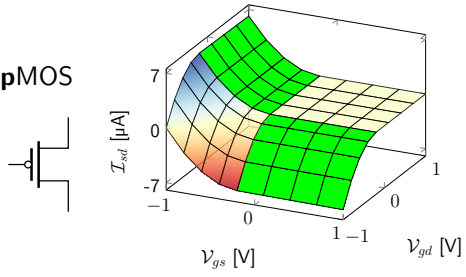
$$\mathcal{I}_{sd}(\mathsf{M}) = 0$$

Transistor's regions of operation: I-V characteristics

nMOS



pMOS

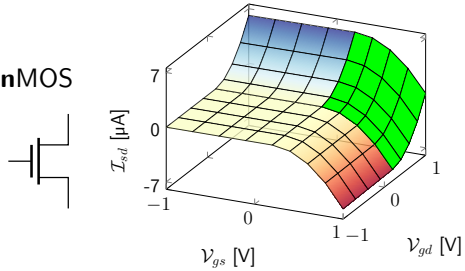


Saturation

$$\mathcal{I}_{sd}(\mathsf{M}) = K_p \frac{W}{L}(1 + \lambda \mathcal{V}_{ds}(\mathsf{M})) \times (\mathcal{V}_{ds}(\mathsf{M}) - V_{th})^2$$

Transistor's regions of operation: I-V characteristics

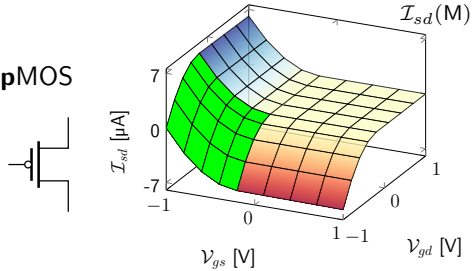
nMOS



Linear

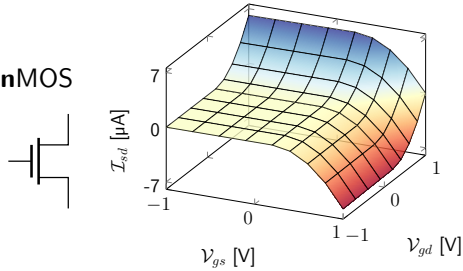
$$\mathcal{I}_{sd}(\mathsf{M}) = \tfrac{1}{2}\tfrac{W}{L}K_p(1+\lambda\mathcal{V}_{ds}(\mathsf{M}))\times\mathcal{V}_{ds}(\mathsf{M})\times(\mathcal{V}_{gs}(\mathsf{M})-V_{th}-\tfrac{1}{2}\mathcal{V}_{ds}(\mathsf{M}))$$

pMOS

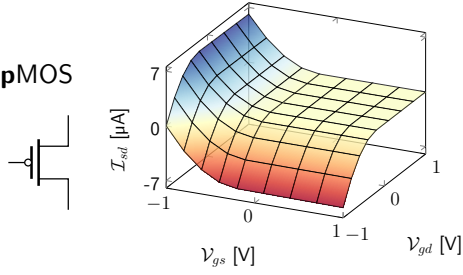


Transistor's regions of operation: I-V characteristics

nMOS

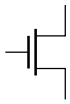


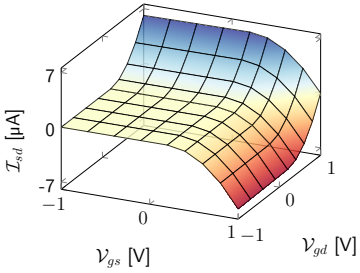
pMOS



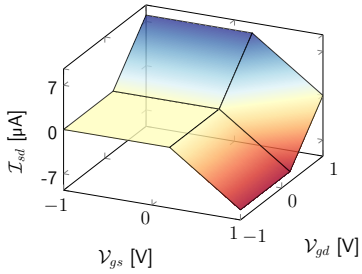
Transistor's regions of operation: I-V characteristics

nMOS

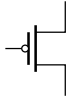


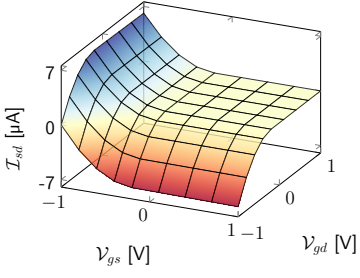


approximation

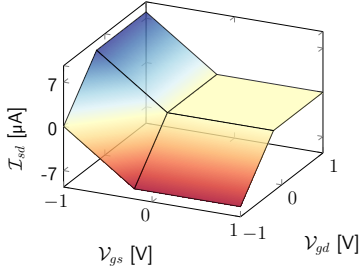


pMOS



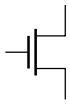


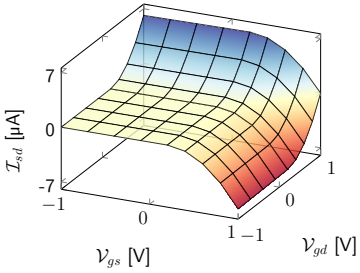
approximation



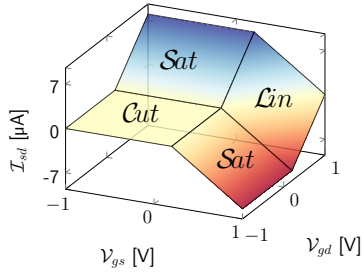
Transistor's regions of operation: I-V characteristics

nMOS

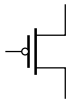


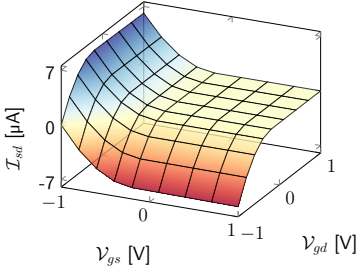


approximation

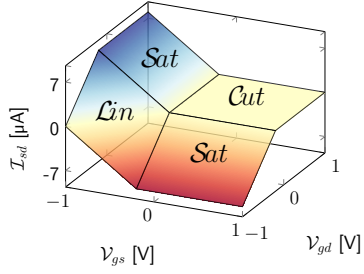


pMOS



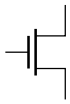


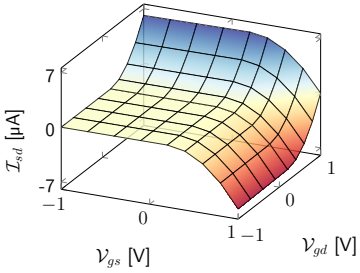
approximation



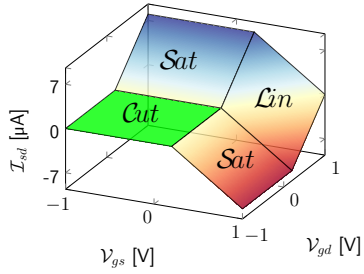
Transistor's regions of operation: I-V characteristics

nMOS

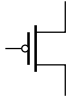


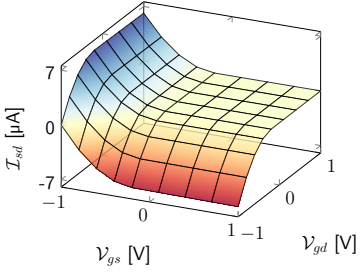


approximation

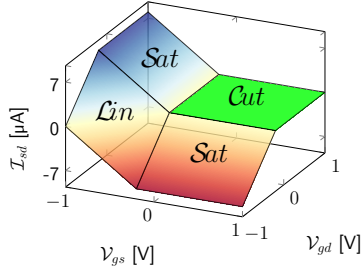


pMOS



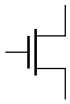


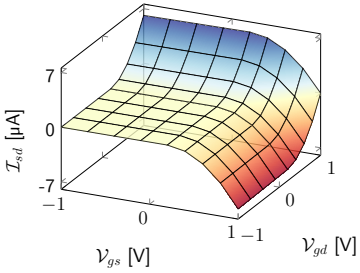
approximation



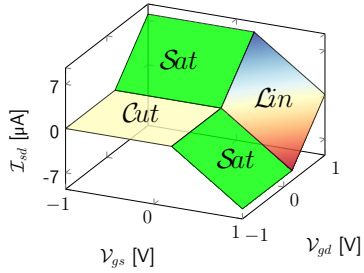
Transistor's regions of operation: I-V characteristics

nMOS

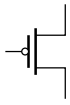


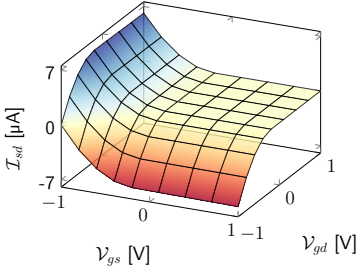


approximation

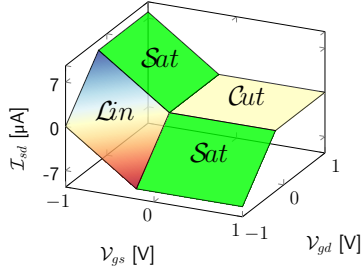


pMOS



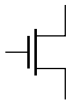


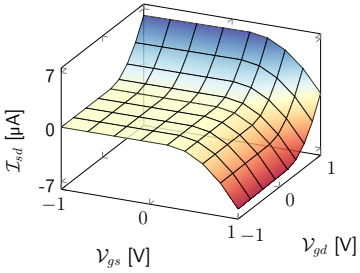
approximation



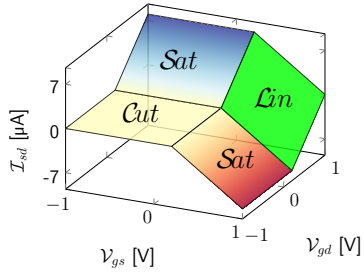
Transistor's regions of operation: I-V characteristics

nMOS

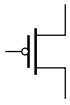


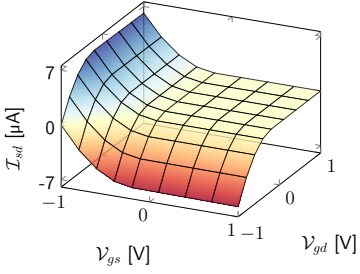


approximation

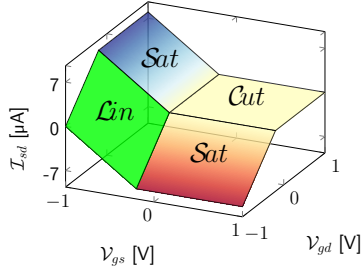


pMOS



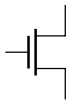


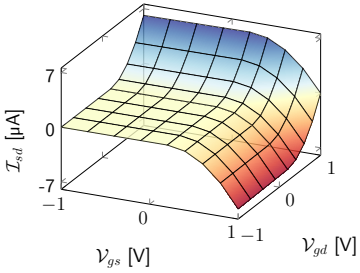
approximation



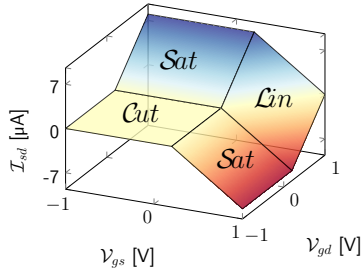
Transistor's regions of operation: I-V characteristics

nMOS

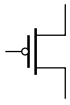


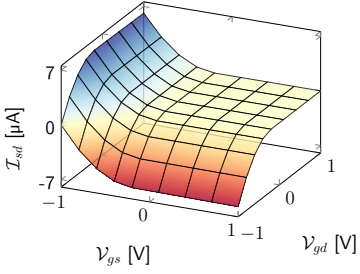


approximation

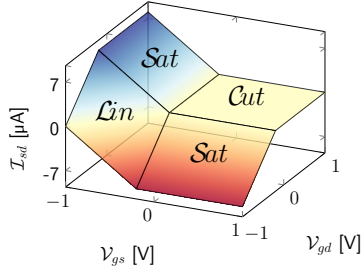


pMOS





approximation



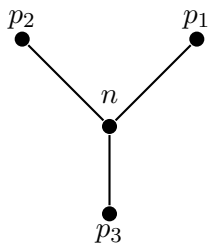
Quantitative semantics rules

Quantitative semantics rules

$$\bigwedge_{\mathsf{M}\in\textbf{Transistors}} \begin{pmatrix} & \mathcal{C}ut_{\mathsf{M}}(\mathcal{V},\mathcal{I}) \\ \vee & \mathcal{S}at_{\mathsf{M}}(\mathcal{V},\mathcal{I}) \\ \vee & \mathcal{L}in_{\mathsf{M}}(\mathcal{V},\mathcal{I}) \end{pmatrix} \qquad (\mathcal{R}_{regions})$$

Quantitative semantics rules

$$\bigwedge_{\mathsf{M}\in\mathbf{Transistors}} \begin{pmatrix} & \mathcal{C}ut_{\mathsf{M}}(\mathcal{V},\mathcal{I}) \\ \vee & \mathcal{S}at_{\mathsf{M}}(\mathcal{V},\mathcal{I}) \\ \vee & \mathcal{L}in_{\mathsf{M}}(\mathcal{V},\mathcal{I}) \end{pmatrix} \qquad (\mathcal{R}_{regions})$$

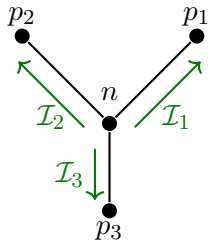


$$\bigwedge_{n\in\mathbf{Nets}} \left(\sum_{\substack{n \xrightarrow{\mathsf{M}} p \\ \in \textsc{Neighbors}(n)}} \mathcal{I}(n \xrightarrow{\mathsf{M}} p) = 0 \right) \qquad (\mathcal{R}_{Kirchhoff})$$

Quantitative semantics rules

$$\bigwedge_{\mathsf{M}\in\textbf{Transistors}} \begin{pmatrix} & \mathcal{C}ut_{\mathsf{M}}(\mathcal{V},\mathcal{I}) \\ \vee & \mathcal{S}at_{\mathsf{M}}(\mathcal{V},\mathcal{I}) \\ \vee & \mathcal{L}in_{\mathsf{M}}(\mathcal{V},\mathcal{I}) \end{pmatrix} \qquad (\mathcal{R}_{regions})$$

$$\bigwedge_{n\in\textbf{Nets}} \left(\sum_{\substack{n \xrightarrow{\mathsf{M}} p \\ \in \textsc{Neighbors}(n)}} \mathcal{I}(n \xrightarrow{\mathsf{M}} p) = 0 \right) \qquad (\mathcal{R}_{Kirchhoff})$$



$$\mathcal{I}_1 + \mathcal{I}_2 + \mathcal{I}_3 = 0$$

Quantitative circuit semantics

Quantitative circuit semantics

$$
\begin{aligned}
\mathcal{S}^q \quad &\stackrel{\text{def}}{=\!=} \quad \mathcal{R}_{regions} \\
&\wedge \quad \mathcal{R}_{Kirchhoff} \\
&\wedge \quad \mathcal{R}_{supplies} \\
&\wedge \quad \mathcal{R}_{inputs} \\
&\wedge \quad \ldots
\end{aligned}
$$

Quantitative circuit semantics

$$\mathcal{S}^q \stackrel{\text{def}}{=\!=} \boxed{\mathcal{R}_{regions}}\ \text{devices} \wedge \mathcal{R}_{Kirchhoff} \wedge \mathcal{R}_{supplies} \wedge \mathcal{R}_{inputs} \wedge \ldots$$

Quantitative circuit semantics

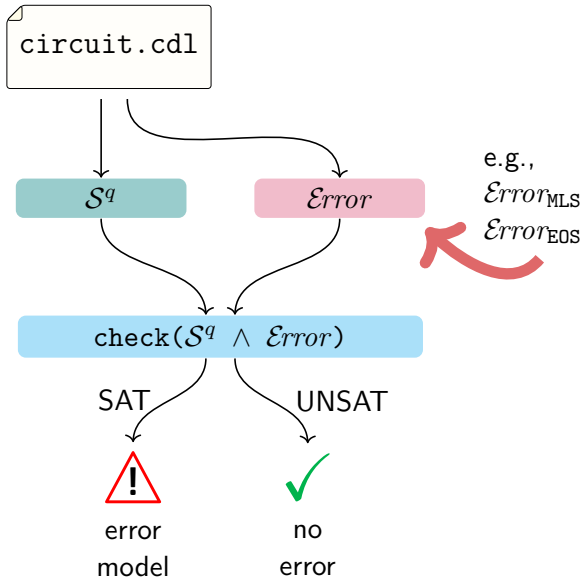
$$
\begin{aligned}
\mathcal{S}^q \quad &\stackrel{\text{def}}{=\!=} \quad \boxed{\mathcal{R}_{regions}} \text{ devices} \\
&\wedge \quad \boxed{\mathcal{R}_{Kirchhoff}} \text{ nets} \\
&\wedge \quad \mathcal{R}_{supplies} \\
&\wedge \quad \mathcal{R}_{inputs} \\
&\wedge \quad \ldots
\end{aligned}
$$

Quantitative circuit semantics

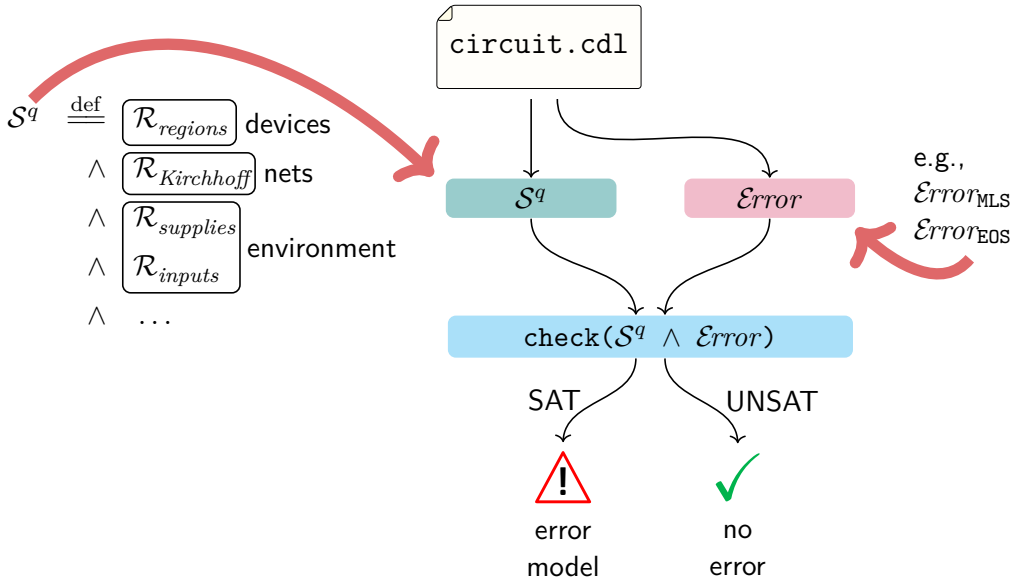
$$
\begin{aligned}
\mathcal{S}^q \quad &\stackrel{\text{def}}{=\!=} \quad \boxed{\mathcal{R}_{regions}} && \text{devices} \\
&\wedge \quad \boxed{\mathcal{R}_{Kirchhoff}} && \text{nets} \\
&\wedge \quad \boxed{\begin{array}{l}\mathcal{R}_{supplies} \\ \mathcal{R}_{inputs}\end{array}} && \text{environment} \\
&\wedge \quad \ldots
\end{aligned}
$$

Quantitative circuit semantics

$$\mathcal{S}^q \stackrel{\text{def}}{=} \boxed{\mathcal{R}_{regions}} \text{ devices} \wedge \boxed{\mathcal{R}_{Kirchhoff}} \text{ nets} \wedge \boxed{\mathcal{R}_{supplies} \\ \mathcal{R}_{inputs}} \text{ environment} \wedge \ldots$$

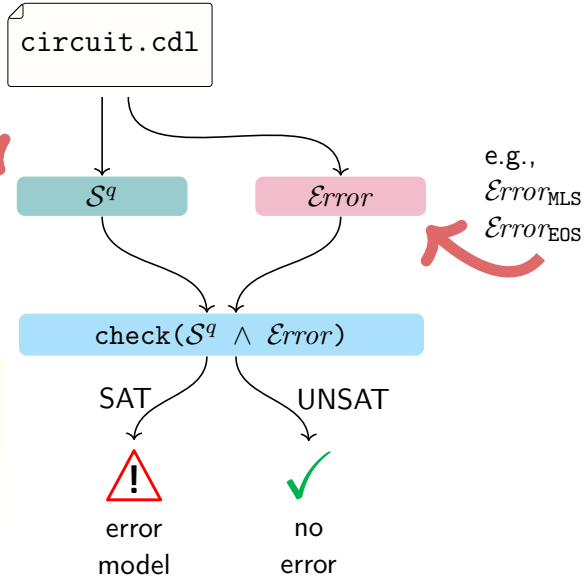


Quantitative circuit semantics



Quantitative circuit semantics

$$\mathcal{S}^q \stackrel{\text{def}}{=} \boxed{\mathcal{R}_{regions}} \text{ devices} \wedge \boxed{\mathcal{R}_{Kirchhoff}} \text{ nets} \wedge \boxed{\mathcal{R}_{supplies} \; \mathcal{R}_{inputs}} \text{ environment} \wedge \ldots$$



IEEE TCAD 2025

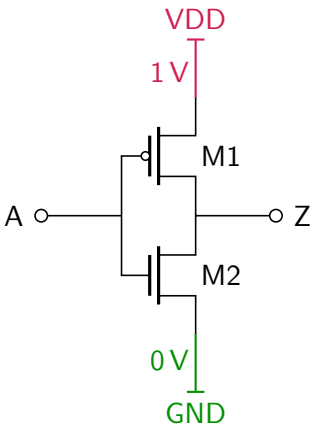
Modeling Techniques for the Formal Verification of Integrated Circuits at Transistor-Level: Performance Versus Precision Tradeoffs

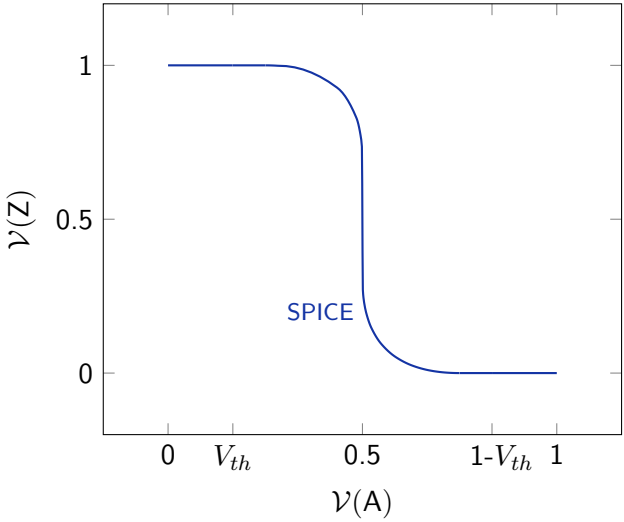
Oussama Oulkaid[1,¶], Bruno Ferres[2], Matthieu Moy[3], Pascal Raymond[3], Mehdi Khosravian[¶]
[1]Univ. Lyon, Ens[d]., UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP*, VERIMAG, 38000 Grenoble, France
[¶]Aniah, 38000 Grenoble, France

Abstract—The behavior of any electronic system can be traced back to how its constituting components physically interact with each other. Such low-level interactions explain how specific states of a given circuit are physically possible. Some circuit states can be erroneous, e.g., applying a voltage stress greater than what some device can tolerate. It is of particular importance to know whether such errors can happen on a given circuit, so that required corrections can be made. Identifying errors requires executions can cover a representative subset. Formal methods, on the other hand, can explore an abstraction of the state-space more exhaustively. To ensure soundness, one therefore needs to consider conservative assumptions about the circuit behavior. However, the more coarse the modeling used in a verification tool is, the more the rate of false alarms the tool reports is likely to be high. False alarms can only be avoided by being

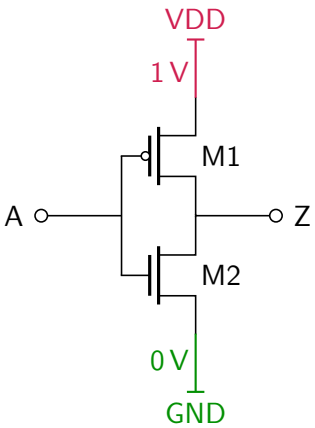
Semantics comparison: case of the inverter

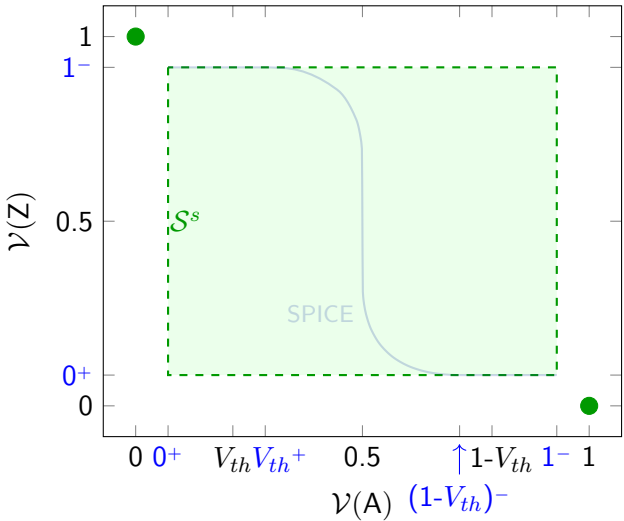
Semantics comparison: case of the inverter



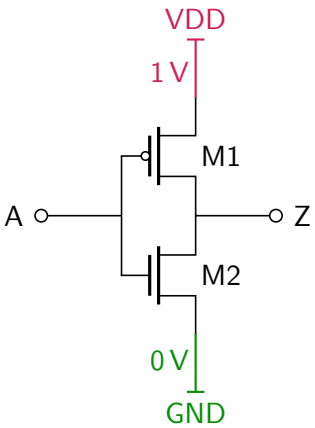


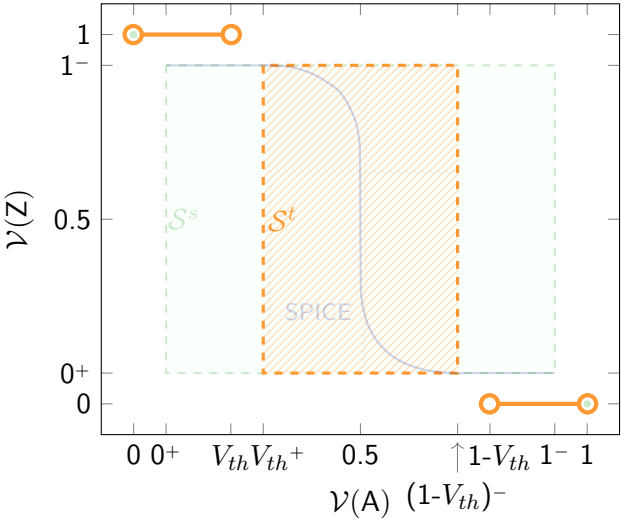
Semantics comparison: case of the inverter



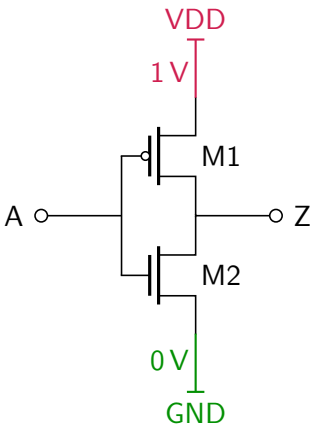


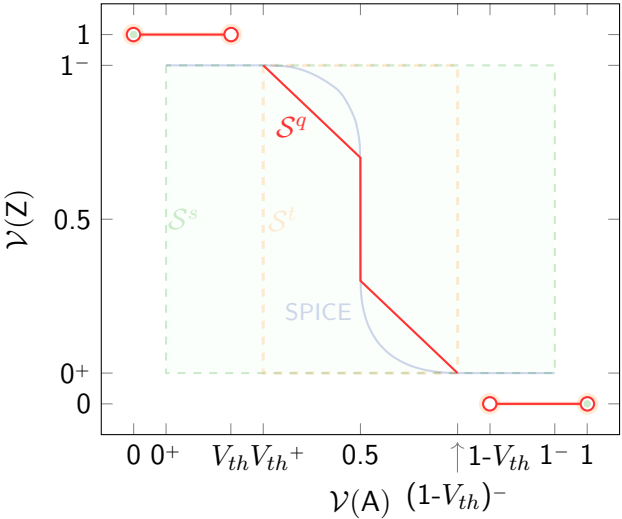
Semantics comparison: case of the inverter



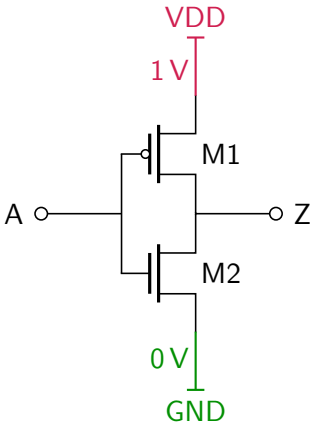


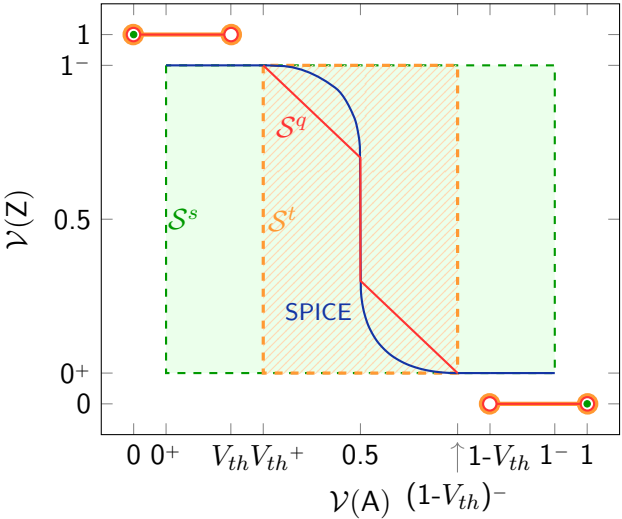
Semantics comparison: case of the inverter





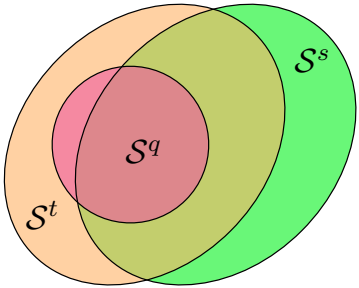
Semantics comparison: case of the inverter



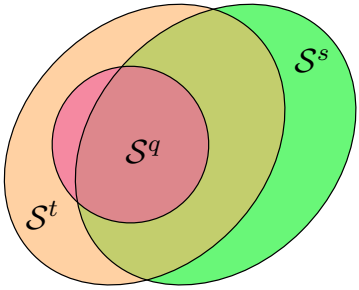


Formal comparisons of semantics

Formal comparisons of semantics

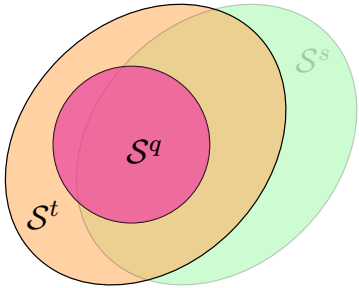


Formal comparisons of semantics



$$\forall \mathcal{V},\ \forall \mathcal{I},\ (\mathcal{V},\mathcal{I}) \models \mathcal{S}^q \ \Rightarrow\ \mathcal{V} \models \mathcal{S}^t$$
$$\exists \mathcal{V},\ \forall \mathcal{I},\ (\mathcal{V},\mathcal{I}) \models (\mathcal{S}^t \wedge \neg \mathcal{S}^q)$$
$$\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^s \wedge \neg \mathcal{S}^t)$$
$$\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^t \wedge \neg \mathcal{S}^s)$$
$$\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^s \wedge \neg \mathcal{S}^q)$$
$$\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^q \wedge \neg \mathcal{S}^s)$$

Formal comparisons of semantics



- $\forall \mathcal{V},\ \forall \mathcal{I},\ (\mathcal{V},\mathcal{I}) \models \mathcal{S}^q \ \Rightarrow\ \mathcal{V} \models \mathcal{S}^t$

  $\exists \mathcal{V},\ \forall \mathcal{I},\ (\mathcal{V},\mathcal{I}) \models (\mathcal{S}^t \wedge \neg \mathcal{S}^q)$

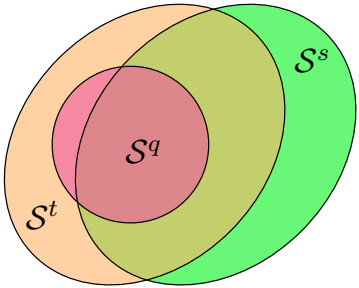
  $\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^s \wedge \neg \mathcal{S}^t)$

  $\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^t \wedge \neg \mathcal{S}^s)$

  $\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^s \wedge \neg \mathcal{S}^q)$

  $\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^q \wedge \neg \mathcal{S}^s)$

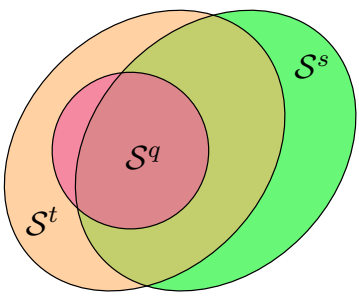
Formal comparisons of semantics



$\forall \mathcal{V},\ \forall \mathcal{I},\ (\mathcal{V},\mathcal{I}) \models \mathcal{S}^q \ \Rightarrow\ \mathcal{V} \models \mathcal{S}^t$

- $\exists \mathcal{V},\ \forall \mathcal{I},\ (\mathcal{V},\mathcal{I}) \models (\mathcal{S}^t \wedge \neg \mathcal{S}^q)$
- $\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^s \wedge \neg \mathcal{S}^t)$
- $\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^t \wedge \neg \mathcal{S}^s)$
- $\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^s \wedge \neg \mathcal{S}^q)$
- $\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^q \wedge \neg \mathcal{S}^s)$

Formal comparisons of semantics



$$\forall \mathcal{V},\ \forall \mathcal{I},\ (\mathcal{V},\mathcal{I}) \models \mathcal{S}^q \ \Rightarrow\ \mathcal{V} \models \mathcal{S}^t$$

$$\exists \mathcal{V},\ \forall \mathcal{I},\ (\mathcal{V},\mathcal{I}) \models (\mathcal{S}^t \wedge \neg \mathcal{S}^q)$$

$$\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^s \wedge \neg \mathcal{S}^t)$$

$$\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^t \wedge \neg \mathcal{S}^s)$$

$$\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^s \wedge \neg \mathcal{S}^q)$$

$$\exists \mathcal{V},\ \mathcal{V} \models (\mathcal{S}^q \wedge \neg \mathcal{S}^s)$$

IEEE TCAD 2025

Modeling Techniques for the Formal Verification of Integrated Circuits at Transistor-Level: Performance Versus Precision Tradeoffs

Oussama Oulkaid[†¶], Bruno Ferres[†], Matthieu Moy[§], Pascal Raymond[†], Mehdi Khosravian[¶]
[†]Univ. Lyon, Ens., UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France
[§]Univ. Grenoble Alpes, CNRS, Grenoble INP*, VERIMAG, 38000 Grenoble, France
[¶]Aniah, 38000 Grenoble, France

***Abstract*—The behavior of any electronic system can be traced back to how its constituting components physically interact with each other. Such low-level interactions explain how specific states of a given circuit are physically possible. Some circuit states can be erroneous, e.g., applying a voltage stress greater than what some device can tolerate. It is of particular importance to know whether such errors can happen on a given circuit, so that required corrections can be made. Identifying errors requires some circuit modeling technique, and a way to explore the state space of the circuit model (which may be very large if at all finite). In this work, we show the limitations of classical verification techniques, and propose a new approach based on formal methods to overcome them. We propose new circuit semantics for transistor-level descriptions from (1) recalling and improving existing semantics, and (2) introducing novel alternate ones. We then demonstrate their usage in our verification framework—which makes use of a satisfiability modulo theories (SMT) solver—to verify specific electric properties of circuits. Specifically, we address the problem of the search for circuit transistors that are subject to electrical overstress (EOS). We draw interesting conclusions by comparing the presented circuit semantics, both formally and via experimental benchmarks.**

***Index Terms*—Electrical overstress (EOS), Electrical rule checking (ERC), Formal verification, Integrated circuits, Satisfiability modulo theories (SMT) solving.**

I. Introduction

VERIFICATION is an essential phase in the design flow of integrated circuits. It represents 50–60% of the entire project time [1]. Albeit existing verification techniques are numerous, new Application Specific Integrated Circuit (ASIC) designs are still subject to logical and functional flaws, among which 60% are caused by design errors [2]. The ideal verification flow is both *sound* (i.e., never misses an actual error) and scalable. However, in practice, it is very hard to achieve both properties, which leads to many design errors only being discovered after tape-out. Unsoundness in a verification tool is due, in practice, to either (1) wrong hypotheses about the circuit behavior, which lead to some circuit states not being captured in the abstract modeling of the circuit [3], or (2) not considering the full circuit's state-space (which can be very large, or even infinite if we consider continuous intervals of possible voltages). In practice, the concrete state-space of a non-trivial circuit cannot be covered enumeratively, but concrete executions can cover a representative subset. Formal methods, on the other hand, can explore an abstraction of the state-space more exhaustively. To ensure soundness, one therefore needs to consider conservative assumptions about the circuit behavior. However, the more coarse the modeling used in a verification tool is, the more the rate of false alarms the tool reports is likely to be high. False alarms can only be avoided by being more precise. Yet, precise techniques (e.g., simulation) fail to scale to large designs. It is hence difficult to find a good performance/precision trade-off.

In this work, we focus on formally defining *rules* to generate abstract circuit modelings, with the intent of being sound. We present and comparatively study different circuit semantics—aimed, each, at reasoning about electrical properties of circuits for a given level of modeling granularity. We recall the switch-based modeling initially introduced in [4], and we propose enhancements to their semantics. We also propose new quantitative semantics to address limitations that were identified with switch-based semantics in some of the considered use-cases. Finally, we study the usability of each of the presented approaches, and conduct experiments on real-life circuits, namely, for the detection of electrical overstress (EOS) errors.

The paper is organized as follows: Section II presents some background on circuit behavior and the notation used throughout the paper. Section III shows the positioning of this work with respect to related work. Section IV then presents existing modeling techniques and introduces new ones. Section V compares all semantics formally and informally. A demonstration of the use of our semantics-based analyses in tackling the problem of the verification of electrical overstress is presented in Section VI. The demonstration is backed by an experimental evaluation of the approach, conducted on an industrial circuits database. Finally, the paper is concluded with a summary and discussion of future work (Section VII).

II. Circuits: Notation and behavior

A. Background and notation

A circuit description (or *netlist*—i.e., the set of devices the circuit is made of, and how they are interconnected) is structural information which alone does not say much about how the circuit behaves. It is only in the presence of some environment configuration (e.g., power supplies and

[†]Institute of Engineering Univ. Grenoble Alpes
Accepted for publication at IEEE TCAD in September, 2025.

$\mathcal{S}^q$ versus $\mathcal{S}^t$: missing level-shifter

$\mathcal{S}^t$	$\mathcal{S}^q$	count
⚠	⚠	7889
⚠	✓	2388
✓	⚠	0
✓	✓	12321

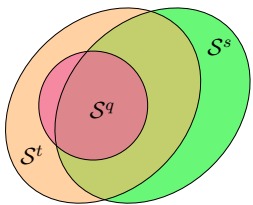
$\mathcal{S}^q$ versus $\mathcal{S}^t$: missing level-shifter

$\mathcal{S}^t$	$\mathcal{S}^q$	count
⚠	⚠	7889
⚠	✓	2388
✓	⚠	0
✓	✓	12321

errors refuted by $\mathcal{S}^q$

$\mathcal{S}^q$ versus $\mathcal{S}^t$: missing level-shifter

$\mathcal{S}^t$	$\mathcal{S}^q$	count	
⚠	⚠	7889	
⚠	✓	2388	errors refuted by $\mathcal{S}^q$
✓	⚠	0	expected, since $\mathcal{S}^q \subseteq \mathcal{S}^t$
✓	✓	12321	

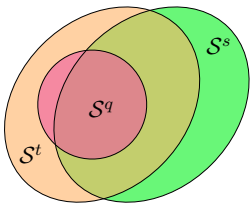


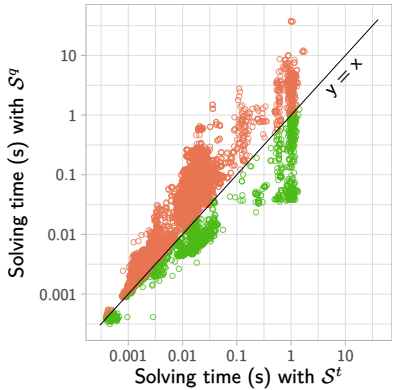
$\mathcal{S}^q$ versus $\mathcal{S}^t$: missing level-shifter

$\mathcal{S}^t$	$\mathcal{S}^q$	count
⚠	⚠	7889
⚠	✓	2388
✓	⚠	0
✓	✓	12321

errors refuted by $\mathcal{S}^q$

expected, since $\mathcal{S}^q \subseteq \mathcal{S}^t$



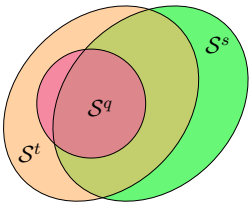


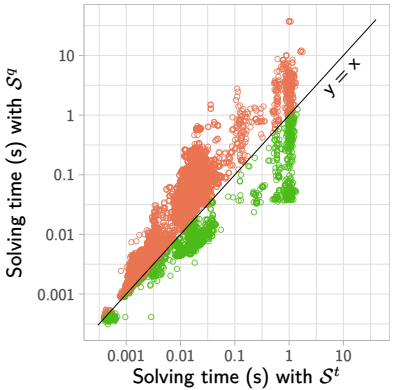
$\mathcal{S}^q$ versus $\mathcal{S}^t$: missing level-shifter

$\mathcal{S}^t$	$\mathcal{S}^q$	count
⚠	⚠	7889
⚠	✓	2388
✓	⚠	0
✓	✓	12321

errors refuted by $\mathcal{S}^q$

expected, since $\mathcal{S}^q \subseteq \mathcal{S}^t$





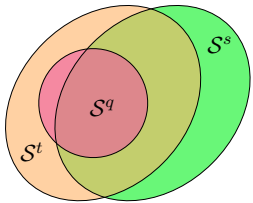
$\mathcal{S}^q$ globally slower

$\mathcal{S}^q$ versus $\mathcal{S}^t$: electrical overstress

$\mathcal{S}^t$	$\mathcal{S}^q$	count
⚠	⚠	2804
⚠	✓	152
✓	⚠	0
✓	✓	188

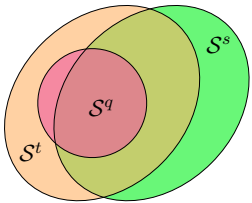
$\mathcal{S}^q$ versus $\mathcal{S}^t$: electrical overstress

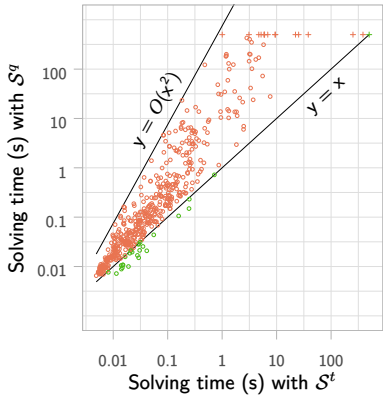
$\mathcal{S}^t$	$\mathcal{S}^q$	count	
⚠	⚠	2804	
⚠	✓	152	errors refuted by $\mathcal{S}^q$
✓	⚠	0	expected, since $\mathcal{S}^q \subseteq \mathcal{S}^t$
✓	✓	188	



$\mathcal{S}^q$ versus $\mathcal{S}^t$: electrical overstress

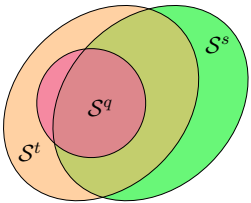
$\mathcal{S}^t$	$\mathcal{S}^q$	count	
⚠	⚠	2804	
⚠	✓	152	errors refuted by $\mathcal{S}^q$
✓	⚠	0	expected, since $\mathcal{S}^q \subseteq \mathcal{S}^t$
✓	✓	188	

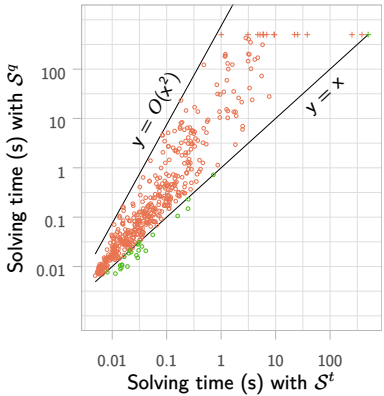




$\mathcal{S}^q$ versus $\mathcal{S}^t$: electrical overstress

$\mathcal{S}^t$	$\mathcal{S}^q$	count	
⚠	⚠	2804	
⚠	✓	152	errors refuted by $\mathcal{S}^q$
✓	⚠	0	expected, since $\mathcal{S}^q \subseteq \mathcal{S}^t$
✓	✓	188	





$\mathcal{S}^q$ promising to reduce false positives rate at the cost of performance

Semantics $\mathcal{S}^s$, $\mathcal{S}^t$ and $\mathcal{S}^q$ are used to identify errors

Semantics $\mathcal{S}^s$, $\mathcal{S}^t$ and $\mathcal{S}^q$ are used to identify errors

Designers fix the circuit ...

That is not the end of the story

That is not the end of the story

Correct circuits are still faced with aging

Part 5 of 5

Circuit Reliability Analysis

Every circuit will eventually die

Every circuit will eventually die

Every circuit will eventually die

Every circuit will eventually die

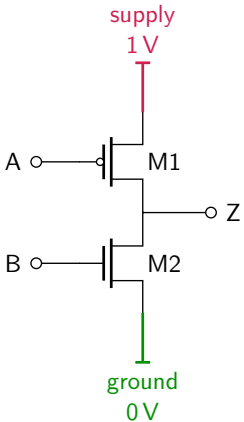
Every circuit will eventually die

Every circuit will eventually die

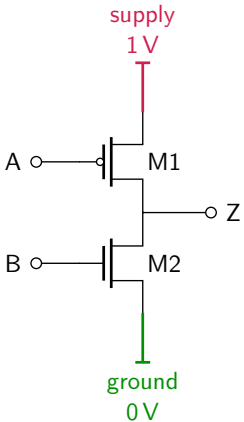
Every circuit will eventually die

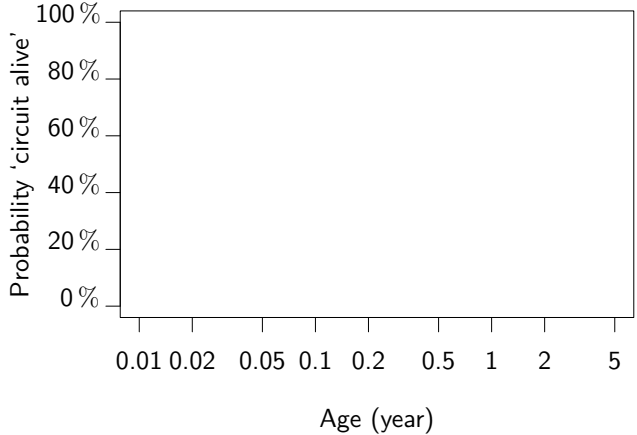
Time-dependent dielectric breakdown (TDDB)

Time-dependent dielectric breakdown (TDDB)

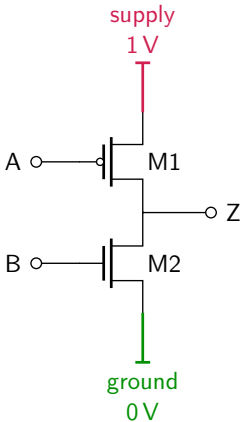


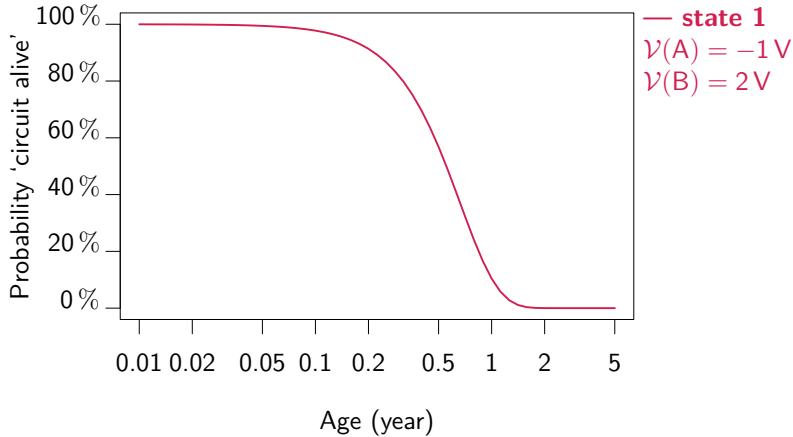
Time-dependent dielectric breakdown (TDDB)



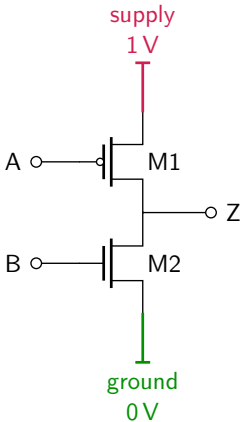


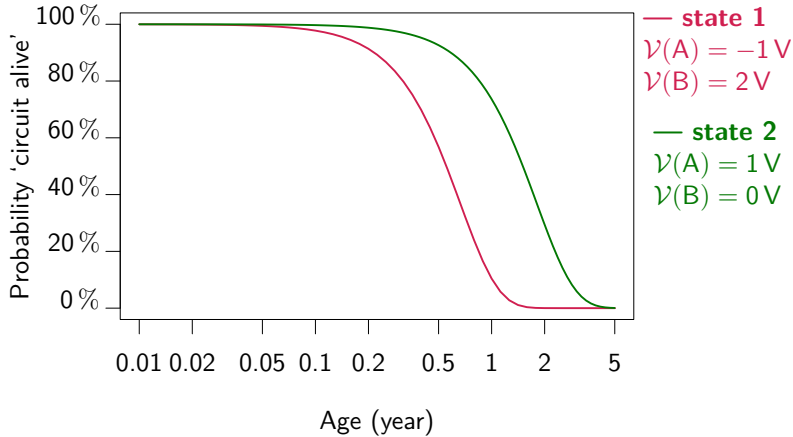
Time-dependent dielectric breakdown (TDDB)



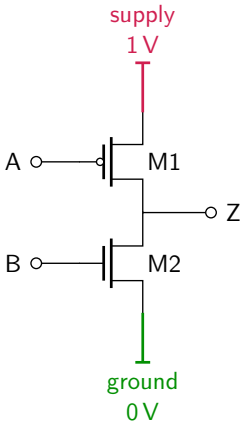


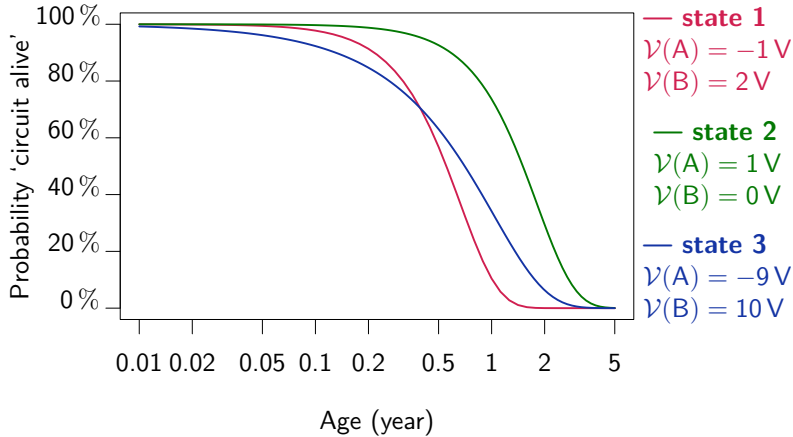
Time-dependent dielectric breakdown (TDDB)



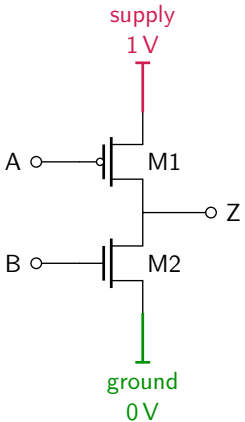


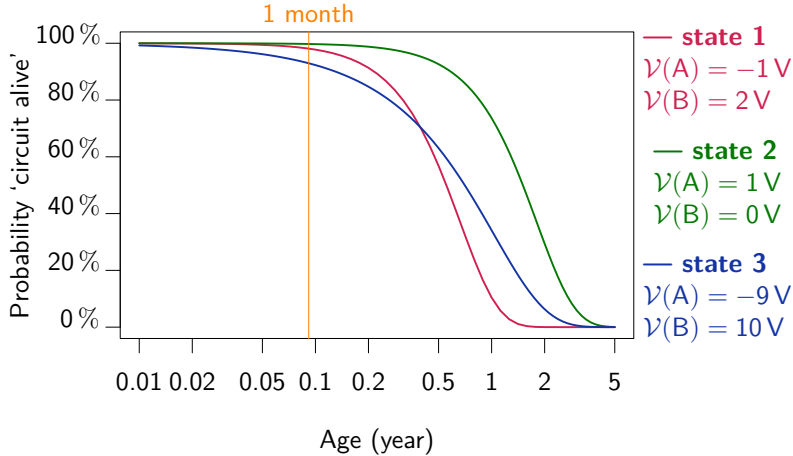
Time-dependent dielectric breakdown (TDDB)



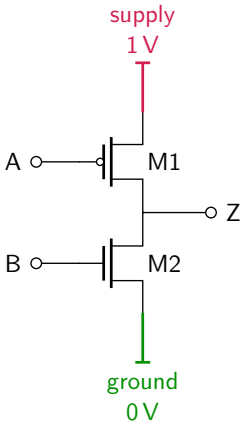


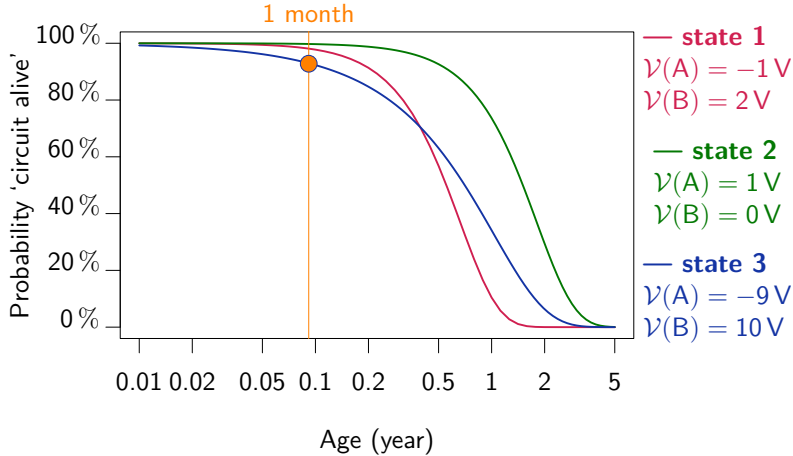
Time-dependent dielectric breakdown (TDDB)



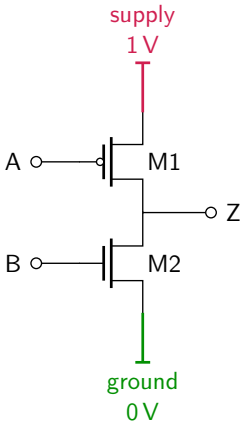


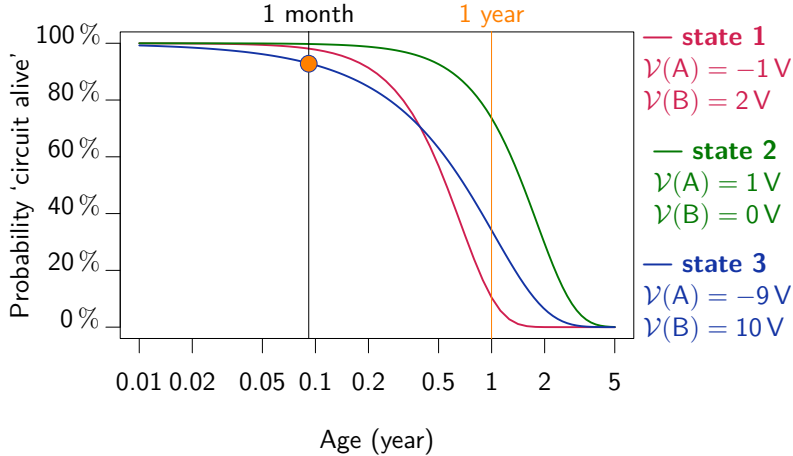
Time-dependent dielectric breakdown (TDDB)



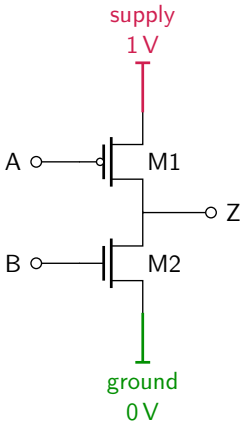


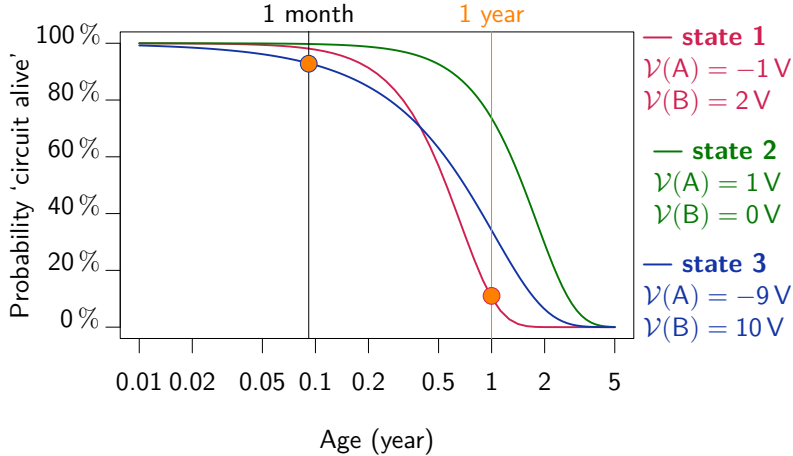
Time-dependent dielectric breakdown (TDDB)



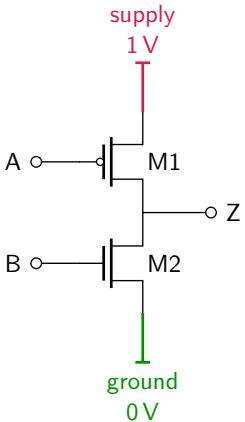


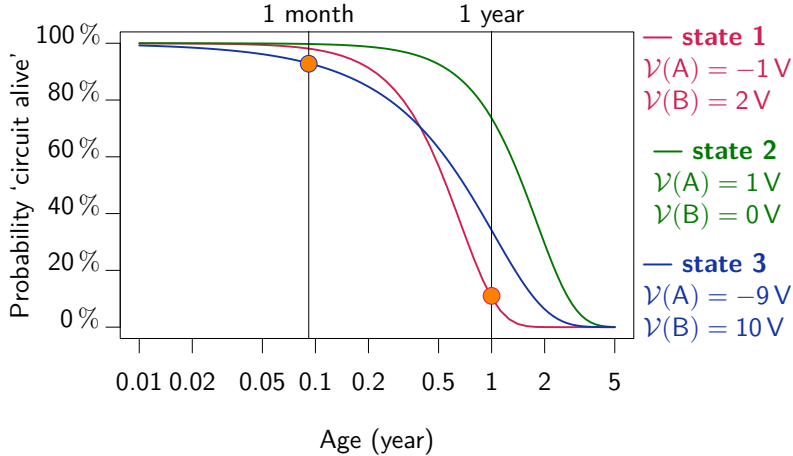
Time-dependent dielectric breakdown (TDDB)





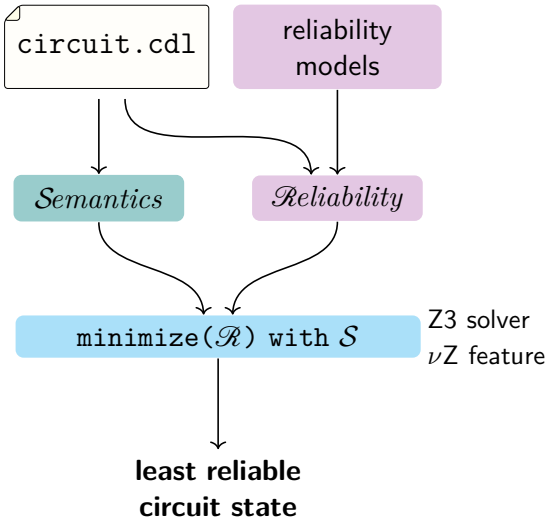
Time-dependent dielectric breakdown (TDDB)



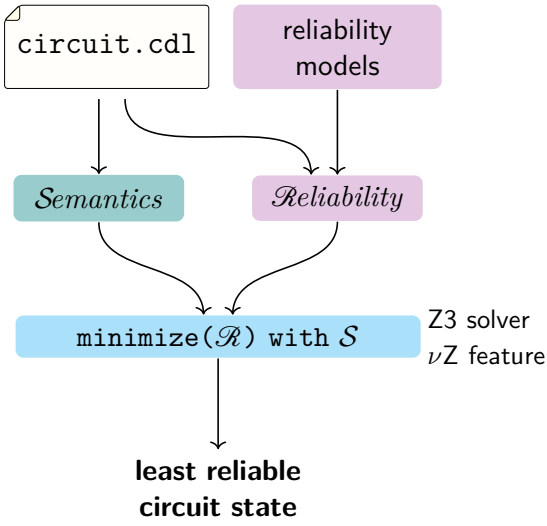


What is the least reliable circuit state?

What is the least reliable circuit state?



What is the least reliable circuit state?



Submitted
IEEE TCAD 2026

Time-Dependent Dielectric Breakdown Worst-Steady-State Analysis of Integrated Circuits using Optimization Modulo Theories

Oussama Oulkaid[§¶], Matthieu Moy[§], Bruno Ferres[§], Pascal Raymond[§], Mehdi Khosravian[¶]
[§]Université Claude Bernard Lyon 1, CNRS, ENS de Lyon, Inria, LIP, UMR 5668, 69342, Lyon cedex 07, France
[§]Univ. Grenoble Alpes, CNRS, Grenoble INP*, VERIMAG, 38000 Grenoble, France
[¶]Aniah, 38000 Grenoble, France

***Abstract*—Integrated circuits are subject to aging, which eventually leads to the loss of their functionality. Their lifetime depends on both their conditions of operation, and devices' technological parameters. In this paper, we propose a novel method for the analysis of the lifetime of circuits, with respect to the Time-Dependent Dielectric Breakdown (TDDB) failure mechanism. Using recently published formal encodings of circuits into logic formulas, we leverage an Optimization Modulo Theories (OMT) solver to yield the circuit state which leads to the fastest aging. This makes it possible to identify the 'worst state' of a circuit, i.e., the state that would lead to the fastest aging if the circuit's state did not evolve over time. This is valuable information for engineers who want to increase the lifetime of their circuits and protect them against catastrophic events like TDDB. Such goal is formalized as an optimization problem (in terms of the variables of the formula encoding the circuit), which can be solved with the help of optimization features of modern solvers, like Z3's νZ OMT engine. We demonstrate the usability of our approach on a database of industrial circuits, showing how it can be used to efficiently analyze small-to-mid size circuits (in the order of a hundred transistors).**

***Index Terms*—Formal methods, Integrated circuits, Optimization Modulo Theories (OMT), Reliability, Time-Dependent Dielectric Breakdown (TDDB), Transistor-level analysis**

I. Introduction

When it comes to integrated circuits design, aging is a type of degradation which can negatively impact both lifetime and performances of a circuit [1], [2]. The exact impact of aging depends on the nature of the failure that occurs in a circuit. To cope with those failures, designers must hence consider various failure mechanisms, to build circuits that are less prone to aging. In this work, we address the Time-Dependent Dielectric Breakdown (TDDB) failure mechanism — also called time-dependent oxide breakdown. The physics behind TDDB may be explained by the creation of defects in the transistor's oxide film, as a result of long-time application of a high voltage. The accumulation of defects leads to a loss of dielectric properties, which causes permanent structural damage in the silicon oxide film [3].

Modeling the precise physics behind TDDB failures is a well studied problem, and can be summarized with models that compute the evolution of the probability of failure for each device as a function of circuit age. This function also depends on physical parameters that can directly be obtained from the foundry or may be derived from existing standard models like JEDEC [4] that describes the probability of failure as a Weibull distribution [5]. Moreover, some ways to use the circuit make it age faster, hence the need to take into account the behavior of the circuit before breakdown.

In the literature, the problem of the prediction of a circuit's lifetime is mainly addressed considering execution traces (typically in simulation) and a physical model, to evaluate the aging of the circuit. These approaches typically reuse and extend Simulation Program with Integrated Circuit Emphasis (SPICE) [6], [7]. The main limitation of SPICE is that it cannot cover continuous intervals of input vectors.

In this work, we reuse a physical failure model (JEDEC), but use a formal, symbolic approach to reason about the behavior of the circuit. We compute the worst case of a circuit with respect to TDDB, i.e., the steady-state of the circuit that makes it age the fastest. Since TDDB depends on circuit state and not on switching activity, we restrict ourselves to the analysis of aging in a given steady-state. We use previously introduced approaches [8], [9] based on formal semantics to symbolically model the behavior of the circuit.

The main contribution of this paper is to formalize an objective function representing the circuit reliability, expressed in terms of variables of the circuit formula. We then use an Optimization Modulo Theories (OMT) [10] solver — i.e., Satisfiability Modulo Theories (SMT) [11] with a support for optimization capabilities — to find the worst case circuit state showcasing a safe lower bound value of reliability, or to find the lifetime at which a specific reliability is acheived, based on the use-case.

II. Overview of the approach

Given a circuit description (i.e., transistor level netlist) and reliability models of its devices, our approach first encodes the circuit into a logic formula $\mathcal{S}$ that is built from circuit semantics presented in previous works. We use two kinds of circuit semantics: (1) switch-based semantics introduced in [8], and (2) quantitative semantics introduced in [9]. We recall these semantics, briefly, in Sections IV-A and IV-B. The

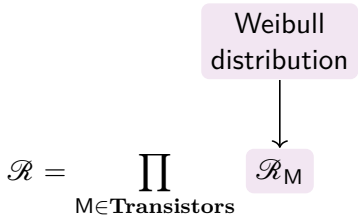
*Institute of Engineering Univ. Grenoble Alpes

Defining circuit reliability $\mathscr{R}$, to `minimize`

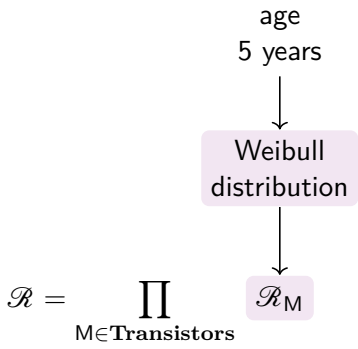
Defining circuit reliability $\mathscr{R}$, to `minimize`

$$\mathscr{R} = \prod_{\mathsf{M} \in \mathbf{Transistors}} \mathscr{R}_{\mathsf{M}}$$

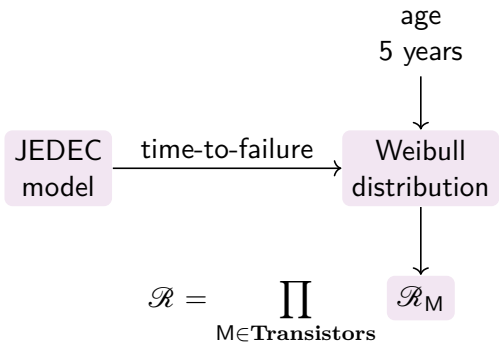
Defining circuit reliability $\mathscr{R}$, to `minimize`



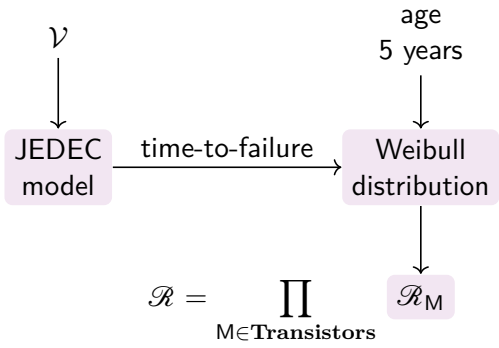
Defining circuit reliability $\mathscr{R}$, to `minimize`



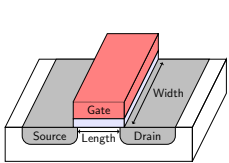
Defining circuit reliability $\mathscr{R}$, to `minimize`



Defining circuit reliability $\mathscr{R}$, to `minimize`



Defining circuit reliability $\mathscr{R}$, to `minimize`



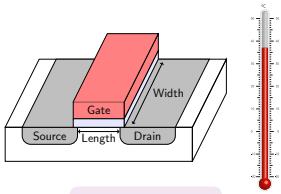
$\mathcal{V}$

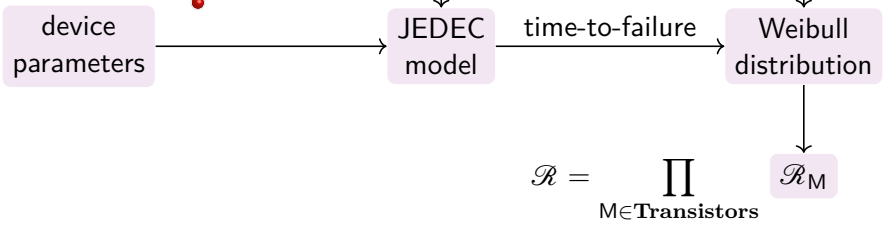
age
5 years

device parameters → JEDEC model → time-to-failure → Weibull distribution → $\mathscr{R}_{\mathsf{M}}$

$$\mathscr{R} = \prod_{\mathsf{M} \in \mathbf{Transistors}} \mathscr{R}_{\mathsf{M}}$$

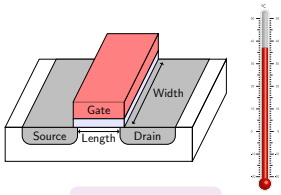
Defining circuit reliability $\mathscr{R}$, to `minimize`





$\mathscr{R}$ is hard to minimize (non-linear arithmetics)

Defining circuit reliability $\mathscr{R}$, to `minimize`



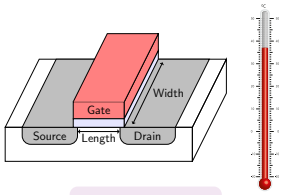
device parameters → JEDEC model ($\mathcal{V}$) → time-to-failure → Weibull distribution (age: 5 years) → $\mathscr{R}_{\mathsf{M}}$

$$\mathscr{R} = \prod_{\mathsf{M} \in \mathbf{Transistors}} \mathscr{R}_{\mathsf{M}}$$

$\mathscr{R}$ is hard to minimize (non-linear arithmetics)

$$\texttt{minimize} \left(\prod_{\mathsf{M}} \mathscr{R}_{\mathsf{M}} \right)$$

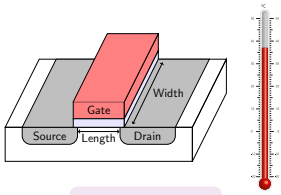
Defining circuit reliability $\mathscr{R}$, to `minimize`



$\mathscr{R}$ is hard to minimize (non-linear arithmetics)

$$\texttt{minimize}\left(\prod_{\mathsf{M}} \mathscr{R}_{\mathsf{M}}\right) \quad \rightsquigarrow \quad \texttt{minimize}\left(\sum_{\mathsf{M}} \ln(\mathscr{R}_{\mathsf{M}})\right)$$

Defining circuit reliability $\mathscr{R}$, to `minimize`



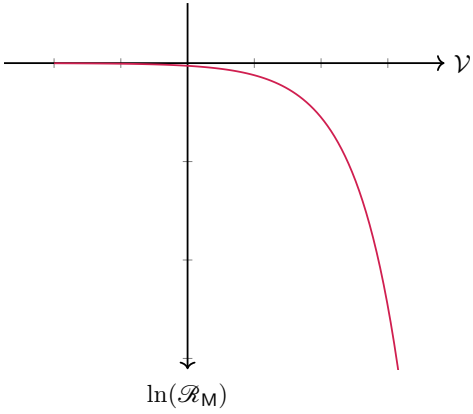
device parameters → JEDEC model ($\mathcal{V}$) → time-to-failure → Weibull distribution (age: 5 years) → $\mathscr{R}_{\mathsf{M}}$

$$\mathscr{R} = \prod_{\mathsf{M} \in \mathbf{Transistors}} \mathscr{R}_{\mathsf{M}}$$

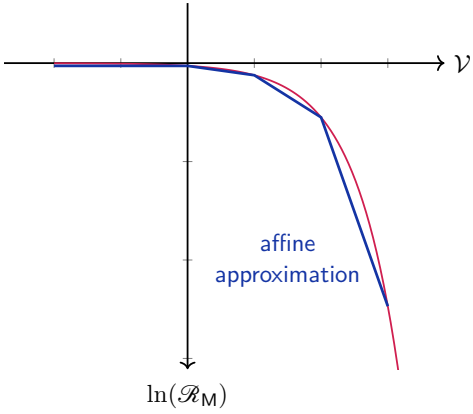
$\mathscr{R}$ is hard to minimize (non-linear arithmetics)

$$\texttt{minimize}\left(\prod_{\mathsf{M}} \mathscr{R}_{\mathsf{M}}\right) \quad \rightsquigarrow \quad \texttt{minimize}\left(\sum_{\mathsf{M}} \underbrace{\ln(\mathscr{R}_{\mathsf{M}})}_{\text{still non-linear}}\right)$$

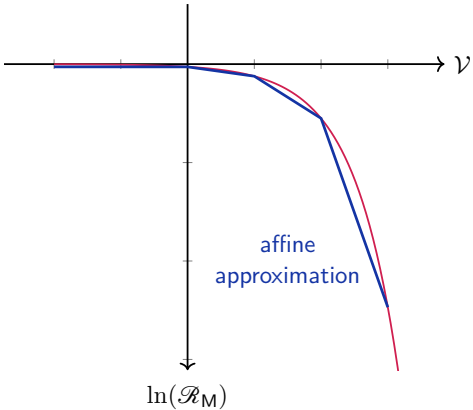
Defining circuit reliability $\mathscr{R}$, to `minimize`

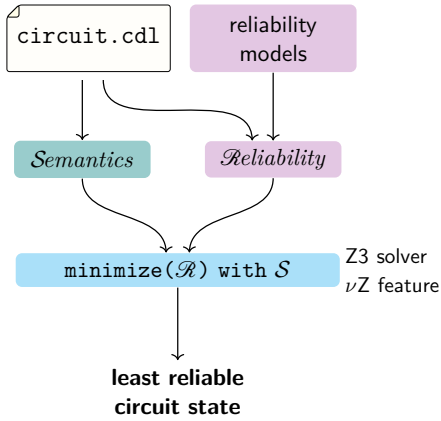


Defining circuit reliability $\mathscr{R}$, to `minimize`

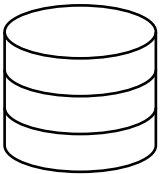


Defining circuit reliability $\mathscr{R}$, to `minimize`

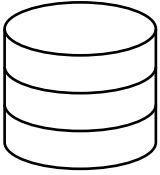




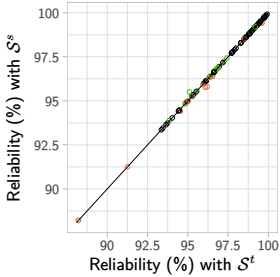
Minimizing circuit reliability: empirical evaluation



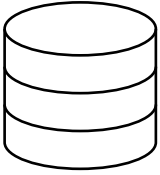
Minimizing circuit reliability: empirical evaluation

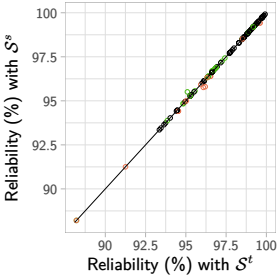


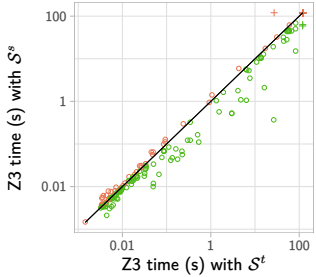
ADC



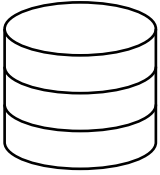
Minimizing circuit reliability: empirical evaluation

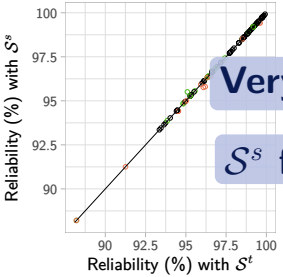






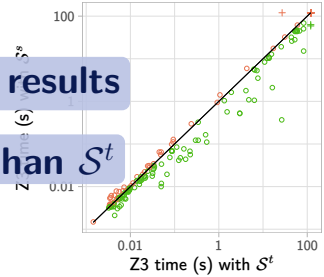
Minimizing circuit reliability: empirical evaluation



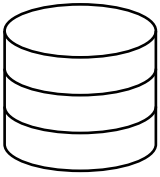


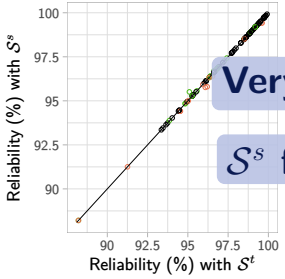
Very close results

$\mathcal{S}^s$ faster than $\mathcal{S}^t$



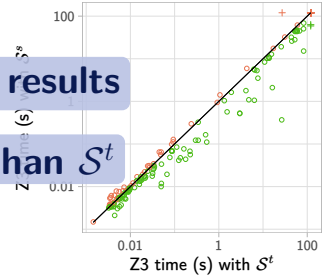
Minimizing circuit reliability: empirical evaluation

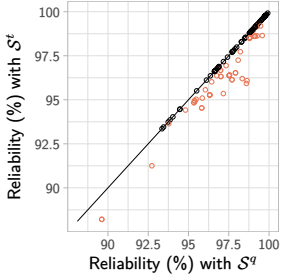


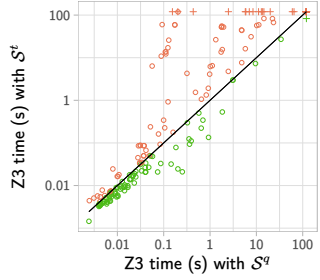


Very close results

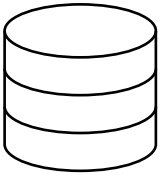
$\mathcal{S}^s$ **faster than** $\mathcal{S}^t$

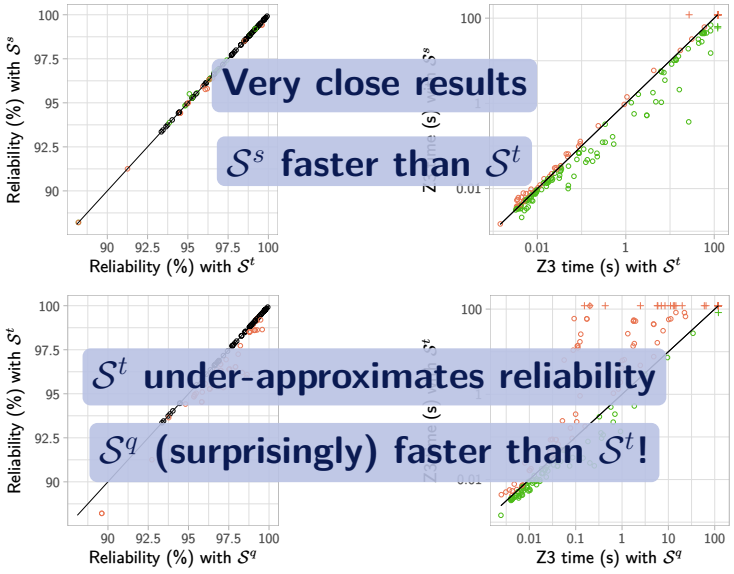




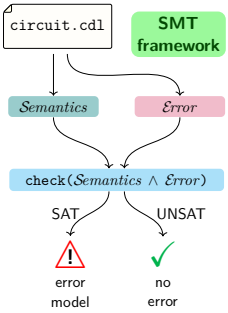


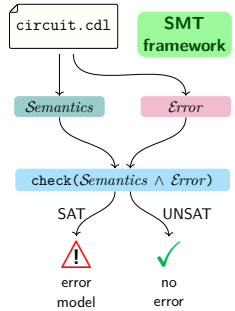
Minimizing circuit reliability: empirical evaluation

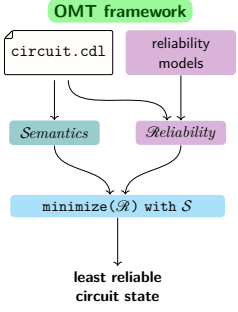


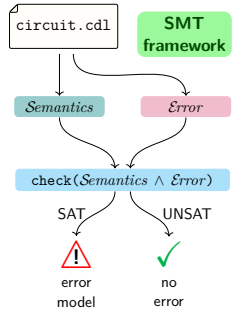


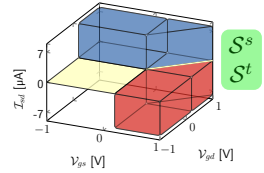
Summary

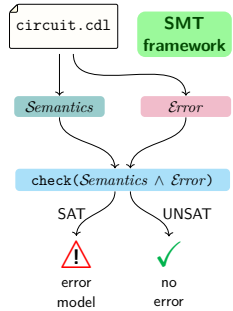

circuit.cdl
SMT framework
$\mathcal{S}emantics$
$\mathcal{E}rror$
check($\mathcal{S}emantics \wedge \mathcal{E}rror$)
SAT
UNSAT
error model
no error

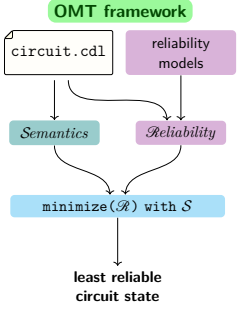

circuit.cdl
SMT framework
$\mathcal{S}emantics$
$\mathcal{E}rror$
check($\mathcal{S}emantics \wedge \mathcal{E}rror$)
SAT
UNSAT
error model
no error

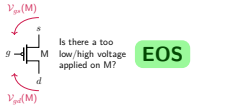

OMT framework
circuit.cdl
reliability models
$\mathcal{S}emantics$
$\mathcal{R}eliability$
minimize($\mathcal{R}$) with $\mathcal{S}$
least reliable circuit state

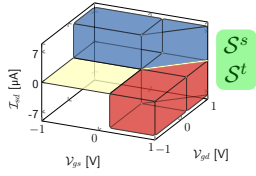

circuit.cdl
SMT framework
Semantics
Error
check(Semantics ∧ Error)
SAT
UNSAT
error model
no error
OMT framework
circuit.cdl
reliability models
Semantics
Reliability
minimize(R) with S
least reliable circuit state

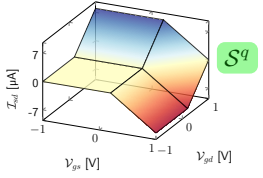

$\mathcal{S}^s$
$\mathcal{S}^t$
$\mathcal{S}^q$
$\mathcal{I}_{sd}$ [μA]
$\mathcal{V}_{gs}$ [V]
$\mathcal{V}_{gd}$ [V]

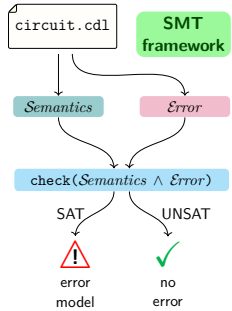

circuit.cdl
SMT framework
Semantics
Error
check(Semantics ∧ Error)
SAT
UNSAT
error model
no error

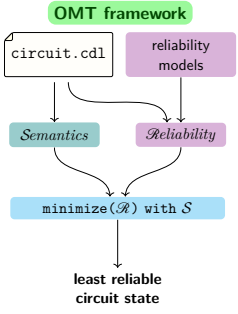

OMT framework
circuit.cdl
reliability models
Semantics
Reliability
minimize(R) with S
least reliable circuit state

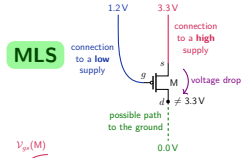

$\mathcal{V}_{gs}$(M)
s
g
M
d
$\mathcal{V}_{gd}$(M)
Is there a too low/high voltage applied on M?
EOS

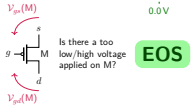

$\mathcal{S}^s$
$\mathcal{S}^t$
$\mathcal{I}_{sd}$ [μA]
7
0
-7
−1
0
1
$\mathcal{V}_{gs}$ [V]
$\mathcal{V}_{gd}$ [V]

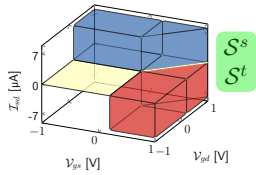

$\mathcal{S}^q$
$\mathcal{I}_{sd}$ [μA]
7
0
-7
−1
0
1
$\mathcal{V}_{gs}$ [V]
$\mathcal{V}_{gd}$ [V]

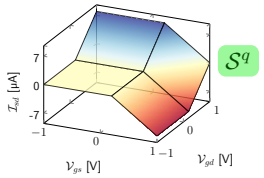

circuit.cdl
SMT framework
Semantics
Error
check(Semantics ∧ Error)
SAT
UNSAT
error model
no error

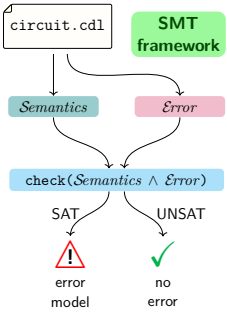

OMT framework
circuit.cdl
reliability models
Semantics
Reliability
minimize(R) with S
least reliable circuit state

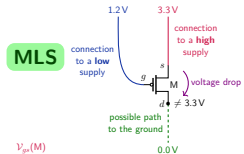

MLS
1.2 V
3.3 V
connection to a low supply
connection to a high supply
voltage drop
≠ 3.3 V
possible path to the ground
0.0 V

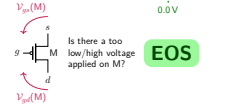

Is there a too low/high voltage applied on M?
EOS

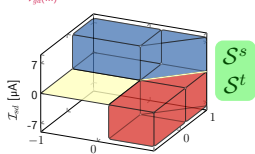

$\mathcal{S}^s$
$\mathcal{S}^t$
$\mathcal{I}_{sd}$ [μA]
$\mathcal{V}_{gs}$ [V]
$\mathcal{V}_{gd}$ [V]

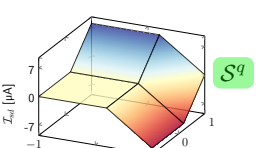

$\mathcal{S}^q$
$\mathcal{I}_{sd}$ [μA]
$\mathcal{V}_{gs}$ [V]
$\mathcal{V}_{gd}$ [V]

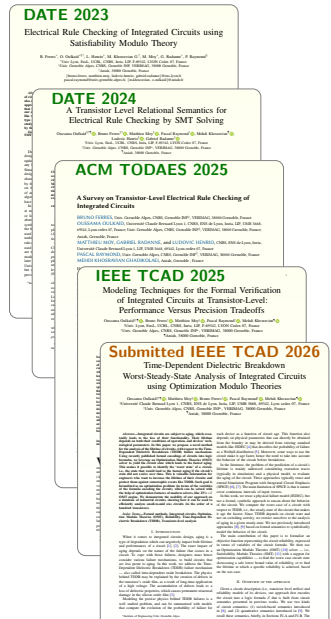

circuit.cdl
SMT framework
Semantics
Error
check(Semantics ∧ Error)
SAT
UNSAT
error model
no error
OMT framework
circuit.cdl
reliability models
Semantics
Reliability
minimize(ℛ) with 𝒮
least reliable circuit state

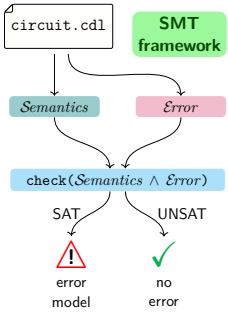

MLS
1.2 V
3.3 V
connection to a low supply
connection to a high supply
s
g
M
d
voltage drop
≠ 3.3 V
possible path to the ground
0.0 V

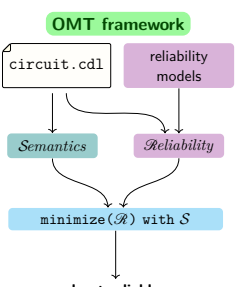

$V_{gs}(M)$
s
g
M
d
$V_{gd}(M)$
Is there a too low/high voltage applied on M?
EOS

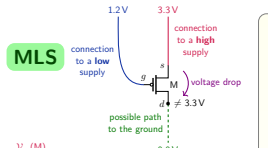

$\mathcal{S}^s$
$\mathcal{S}^t$
$\mathcal{I}_{sd}$ [μA]
$\mathcal{V}_{gs}$ [V]
$\mathcal{V}_{gd}$ [V]

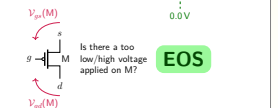

$\mathcal{S}^q$
$\mathcal{I}_{sd}$ [μA]
$\mathcal{V}_{gs}$ [V]
$\mathcal{V}_{gd}$ [V]

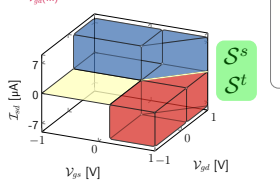

DATE 2023
Electrical Rule Checking of Integrated Circuits using Satisfiability Modulo Theory
DATE 2024
A Transistor Level Relational Semantics for Electrical Rule Checking by SMT Solving
ACM TODAES 2025
A Survey on Transistor-Level Electrical Rule Checking of Integrated Circuits
IEEE TCAD 2025
Modeling Techniques for the Formal Verification of Integrated Circuits at Transistor-Level: Performance Versus Precision Tradeoffs
Submitted IEEE TCAD 2026
Time-Dependent Dielectric Breakdown Worst-Steady-State Analysis of Integrated Circuits using Optimization Modulo Theories

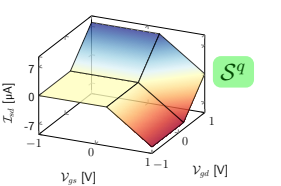

circuit.cdl
SMT framework
Semantics
Error
check(Semantics ∧ Error)
SAT
UNSAT
error model
no error

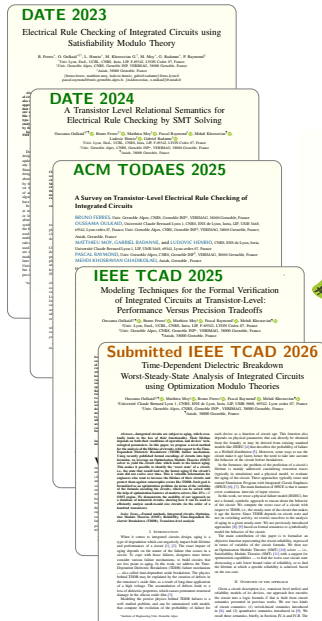

OMT framework
circuit.cdl
reliability models
Semantics
Reliability
minimize(R) with S
least reliable circuit state

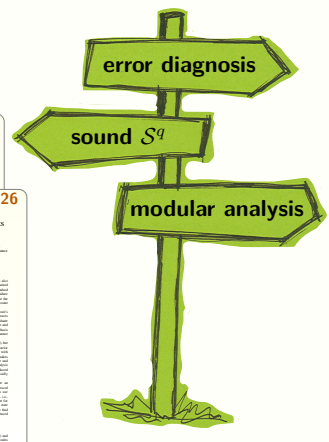

MLS
connection to a low supply
1.2 V
3.3 V
connection to a high supply
s
g
M
voltage drop
d
≠ 3.3 V
possible path to the ground
0.0 V

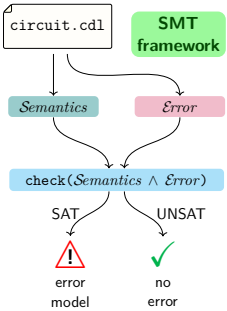

Vgs(M)
s
g
M
d
Vgd(M)
Is there a too low/high voltage applied on M?
EOS

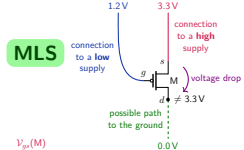

Sˢ
Sᵗ
Iₛd [μA]
Vgs [V]
Vgd [V]

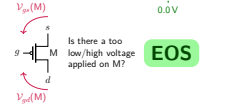

Sᵠ
Iₛd [μA]
Vgs [V]
Vgd [V]

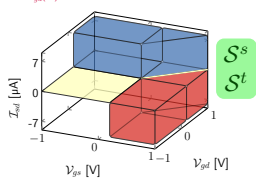

DATE 2023
Electrical Rule Checking of Integrated Circuits using Satisfiability Modulo Theory
DATE 2024
A Transistor Level Relational Semantics for Electrical Rule Checking by SMT Solving
ACM TODAES 2025
A Survey on Transistor-Level Electrical Rule Checking of Integrated Circuits
IEEE TCAD 2025
Modeling Techniques for the Formal Verification of Integrated Circuits at Transistor-Level: Performance Versus Precision Tradeoffs
Submitted IEEE TCAD 2026
Time-Dependent Dielectric Breakdown Worst-Steady-State Analysis of Integrated Circuits using Optimization Modulo Theories

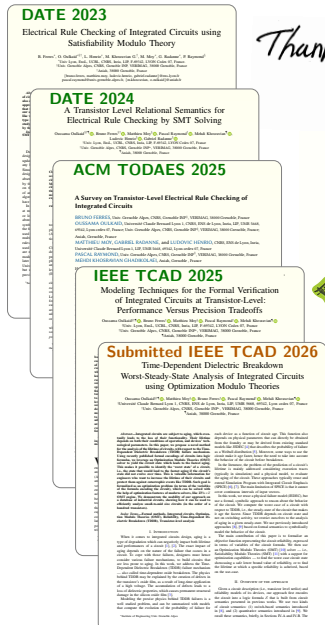

error diagnosis
sound Sᵠ
modular analysis











Thank you for listening !
Questions ?