Formal Verification of Analog Designs using MetiTarski

William Denman, Behzad Akbarpour, Sofiène Tahar[1]
Mohamed H. Zaki[2]
Lawrence C. Paulson[3]

[1]Concordia University, Montreal, Canada
[2]University of British Columbia, Vancouver, Canada
[3]University of Cambridge, United Kingdom

FMCAD'09
November 17th, 2009

Motivation

Should we *care* about **formal verification** for analog circuits?

Verifiers / Researchers

Yes!

Designers

Not really...

Common motivation

Motivation

- Some interesting statistics [IBS Corporation]
  - Analog Circuitry 2% of the transistor count
  - 20% of the IC **Area**
  - 40% of the design **Effort**

Analog verification continues to be a
serious bottleneck

50% of the errors that require re-design
are from analog circuitry

Motivation

Formal Verification for Analog Circuits?

- Challenges
  - Infinite/Continuous state space
  - Infinite time
  - PVT : Sensitivity to process variation, voltage, temperature
  - Non-linear behaviour
- We propose
  - A time unbounded verification
  - Using MetiTarski : An Automated Theorem Prover

Outline

- Motivation
- Related Work
- Proposed Methodology
- Brief Introduction to MetiTarski
- Illustrative Example
- Conclusion
- Future Plans

Related Work

Equivalence Checking | Model Checking/ Reachability Analysis | Proof Based

- Balivada [1995]
  - Discretization of a circuit's transfer function to the Z-domain
  - Apply digital based equivalence checking techniques
- Hartong, Klausen and Hedrich [2004]
  - From analog circuit transfer functions
  - Verify dynamic behaviour of the specification and implementation state spaces.

Presence of tolerance margins

Related Work

Equivalence Checking | Model Checking/ Reachability Analysis | Proof Based

- Kurshan and McMillan [1991]
  - State space subdivision of transistor behaviour
  - Predict possible transitions between states

- Gupta [2004] , Dang [2006], Frehse [2006], Little [2006], Greenstreet [2007]
  - Reachability relations using projection techniques
  - Over-approximation, but verification still sound

Possible Time Bounded Verification

Related Work

Equivalence Checking

Model Checking/ Reachability Analysis

Proof Based

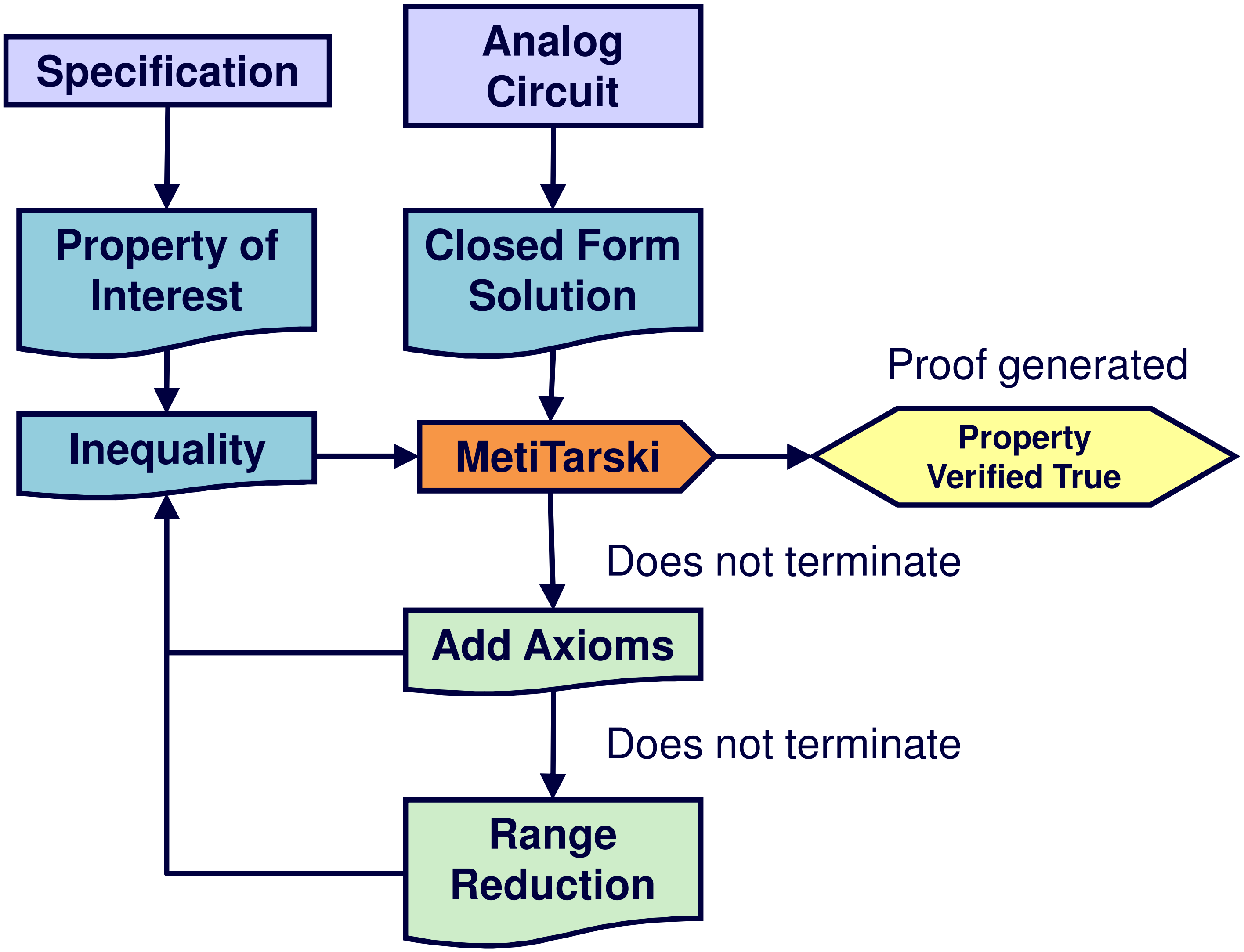
- Ghosh and Vemuri [1999]
  - PVS used to prove functional equivalence between models
  - Specification built in VHDL-AMS
  - Approximated DC models
- Hanna [2000]
  - Predicates defining voltage and current behaviour
  - Theorem Proving used
  - Conservative approximation

Manual/Heuristic steps

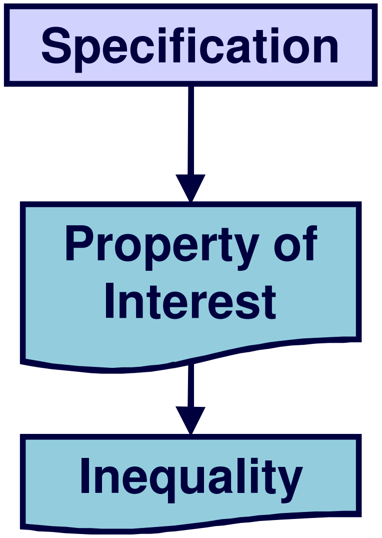
Outline

- Motivation
- Related Work
- Proposed Methodology
- Brief Introduction to MetiTarski
- Illustrative Example
- Conclusion
- Future Plans

Methodology

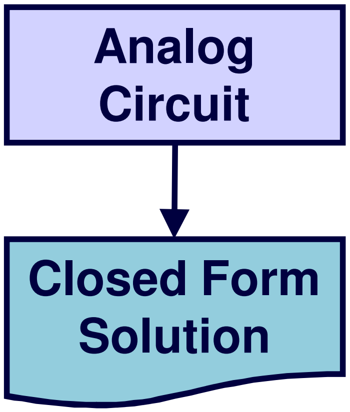


Methodology



- Analog circuit specification
  - Circuit must oscillate
  - Gain for certain frequency range
- Isolate the property
  - Oscillation : Is it present?
  - Gain : 3dB Bandwidth
- Inequality
  - Voltage < Upper threshold
  - Gain > Minimum Required Value

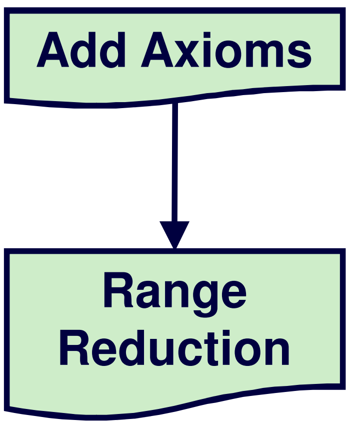
Methodology



- Analog circuit
  - Differential equations
  - Kirchoff law Equations

- Closed Form Solution
  - Bounded number of analytical functions
  - No differential operators
  - Not always easy to obtain

Methodology



- Automated Theorem Proving
  - The axioms are specific mathematical facts
  - Bounding properties
  - Definition of functions
- Range Reduction
  - Functions are not defined over all ranges
  - Large bounds cause proof to never end
  - Apply basic trigonometric identities

$$\cos(x) = \cos(x + 2\pi)$$

$$\sin(x) = \sin(x + 2\pi)$$

Outline

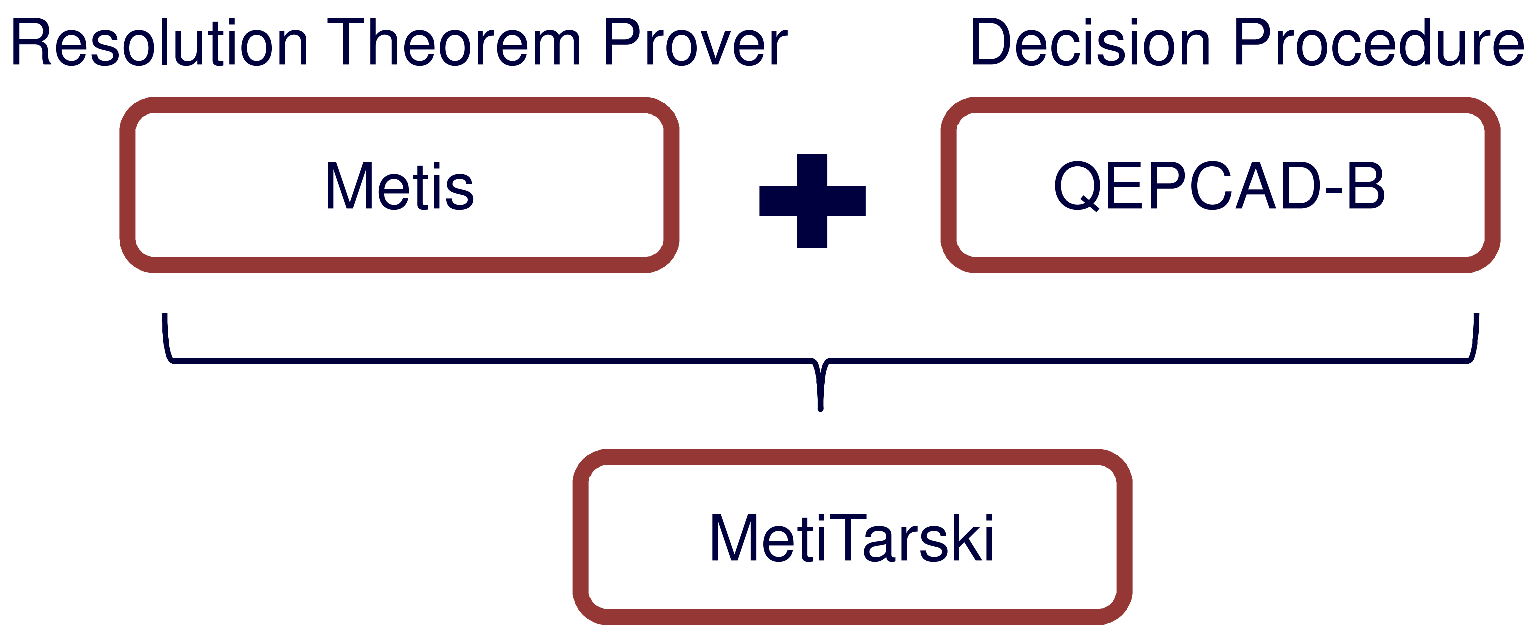
- Motivation
- Related Work
- Proposed Methodology
- Brief Introduction to MetiTarski
- Illustrative Example
- Conclusion
- Future Plans

MetiTarski

- Developed by Akbarpour and Paulson ['07]
  - Automated Theorem Prover
  - Transcendental functions (sine, cosine, ln, exp, etc.)
  - Square Root

- Theory behind the tool
  - Resolution prover combined with a decision procedure
  - Decidability of real closed fields (RCF) by Tarski
  - Function families of upper and lower bounds by Daumas and others

MetiTarski Implementation



MetiTarski

- QEPCAD-B
  - Advanced implementation of cylindrical algebraic decomposition
  - Best available decision procedure for RCF
  - Eliminates quantifiers from a formula

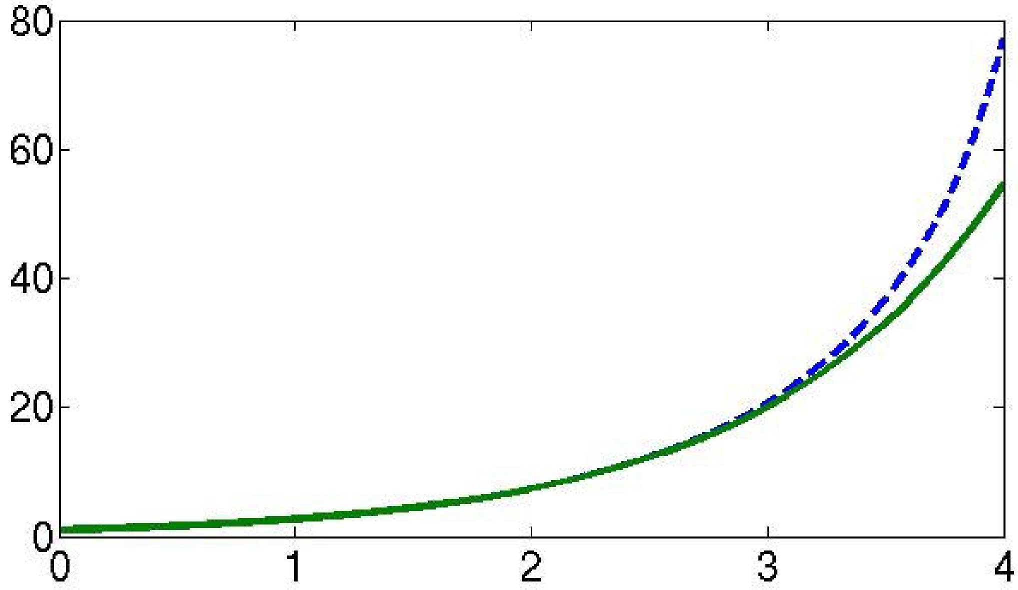
$$\exists x. ax^2 + bx + c = 0$$

reduces to

$$(a \neq 0 \wedge b^2 - 4ac \geq 0) \vee (a = 0 \wedge b \neq 0) \vee (a = b = c = 0)$$

Example Axiom

- Assuming $0 \le x \le 4$
- We are given a function containing exp(x)
  - Upper bound axiom is $\dfrac{-(x^3+12x^2+60x+120)}{x^3-12x^2+60x-120}$

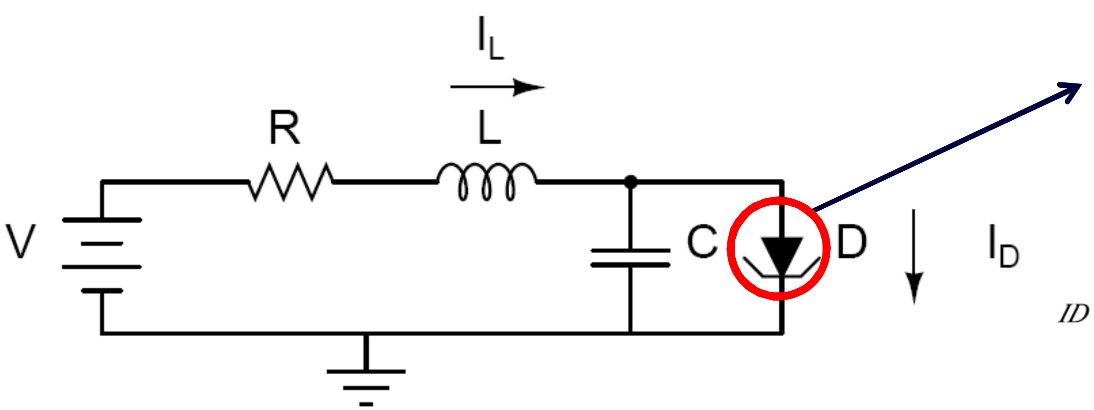


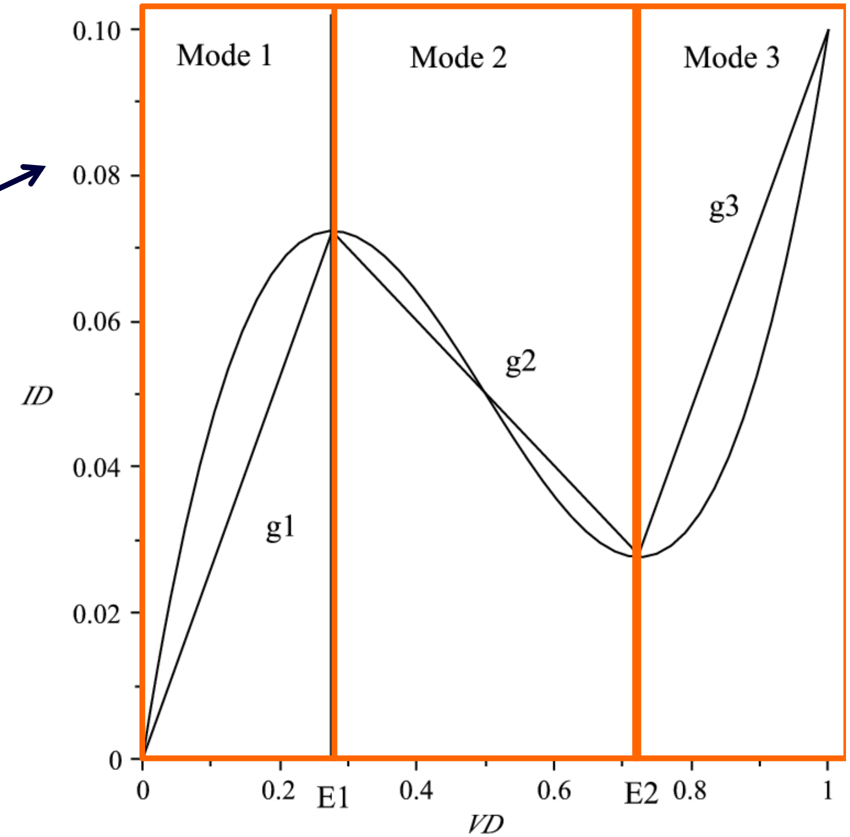
  - Will usually need more than one axiom

Outline

- Motivation
- Related Work
- Proposed Methodology
- Brief Introduction to MetiTarski
- Illustrative Example
- Conclusion
- Future Plans

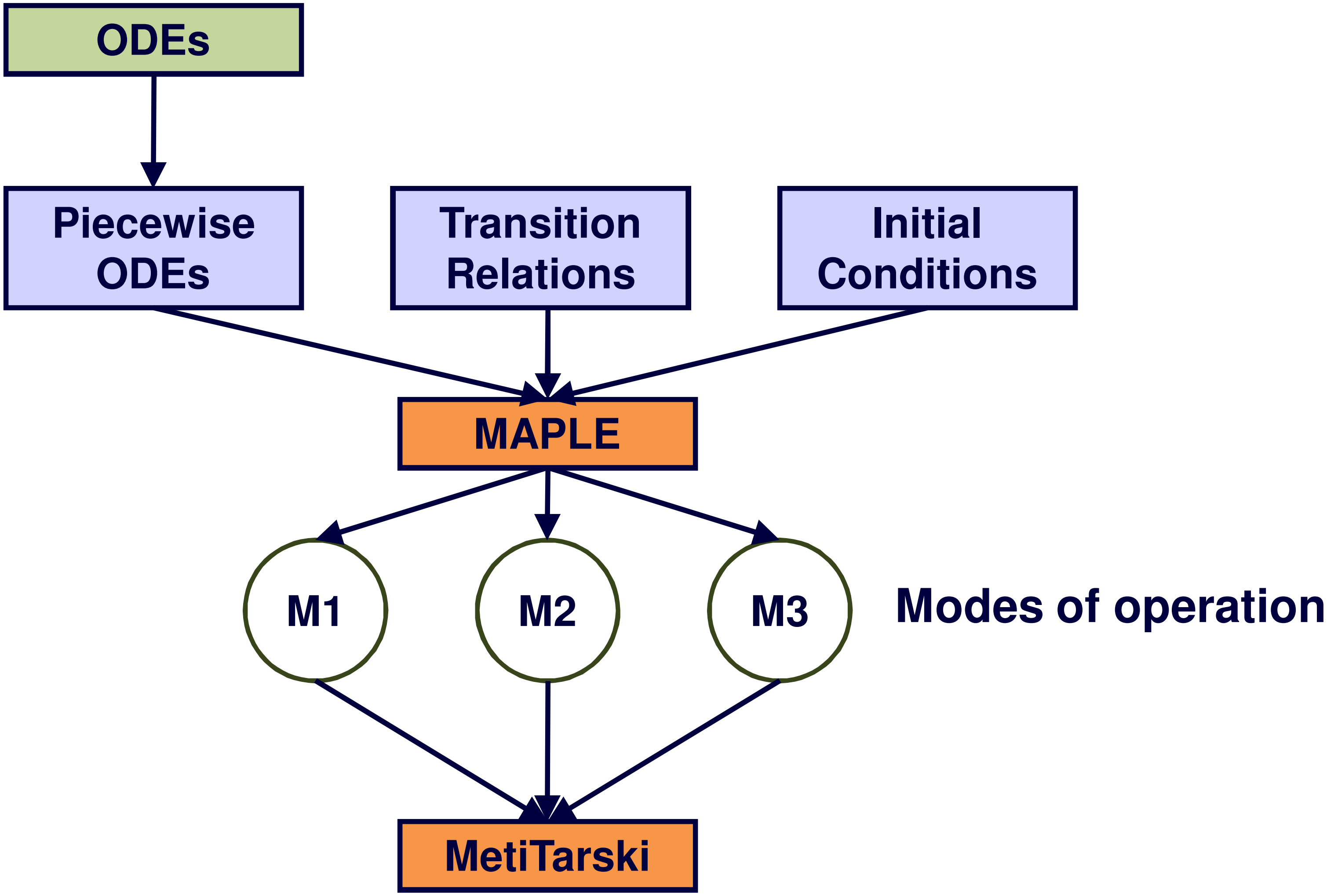
- PWL: Simplest class of nonlinear circuits
- Behaviour can be reasonably approximated





$$I_D(V_C) = \begin{cases} 0.2616V_C & 0 \le V_C \le 0.276 \\ -0.0992V_C + 0.0997 & 0.276 < V_C \le 0.723 \\ 0.2599V_C - 0.1599 & 0.723 \le V_C < 1.0 \end{cases}$$

Closed Form Solution



Closed Form Solution

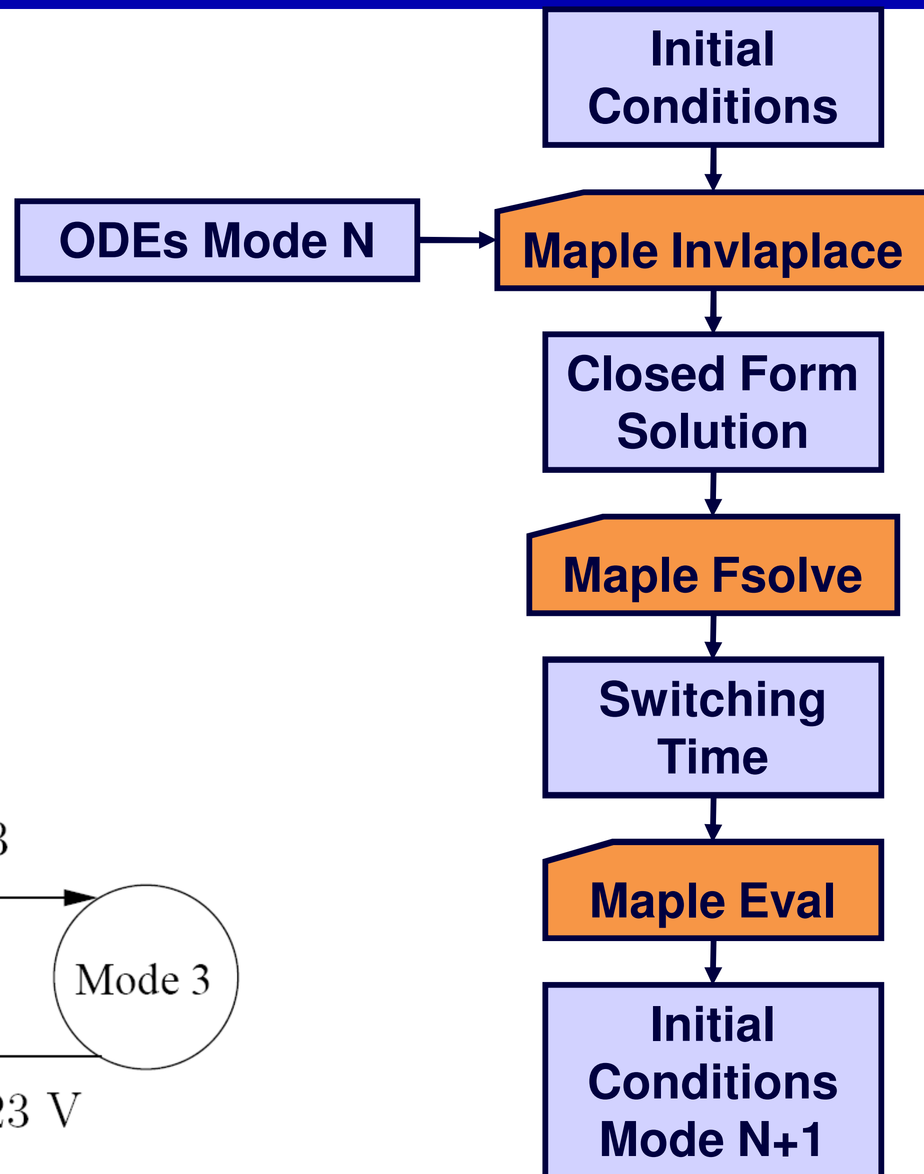
Piecewise ODEs	Transition Relations	Initial Conditions

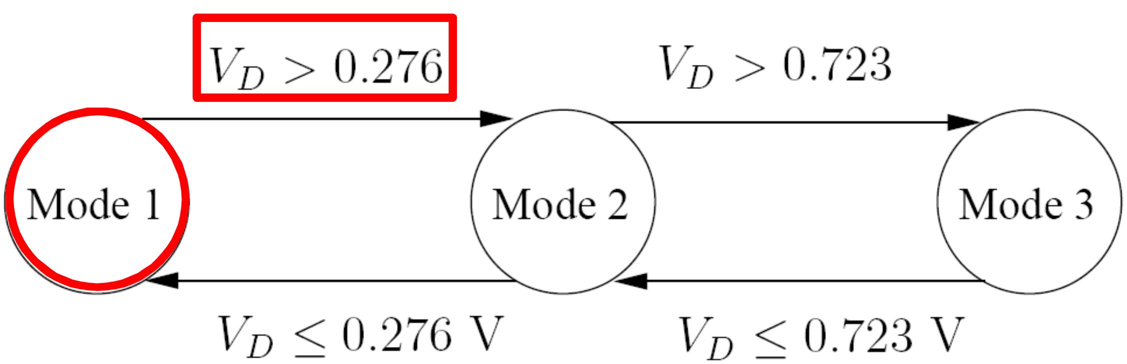
- Using a computer algebra system
- Piecewise ODEs
  - Separate behaviour of the component into modes
- Transition relations
  - Determined by the piecewise model
- Initial Conditions
  - Dependant on the system specification

Closed Form Solution

- Closed form solution for each mode

- Procedure followed until each mode visited





Closed Form Solution

- Starting with the ODEs of the system

$$\dot{V_C} = \frac{1}{C}(-I_D(V_C) + I_L)$$
$$\dot{I_L} = \frac{1}{L}(-V_C - R \times I_L + V_{in})$$

- $I_D(V_C)$ is the current through the tunnel diode
- Inverse Laplace transform taken to get closed form solutions in each mode

$$V_C(t) = 0.116e^{-2.58\times10^8 t} + 0.278 - 0.262e^{-4.19\times10^6 t}$$
$$I_L(t) = 0.448 \times 10^{-3} e^{-2.58\times10^8 t} + 0.0727 - 0.0677e^{-4.19\times10^6 t}$$

Closed Form Solution

- Using the produced solution
  - Fsolve used to compute time when switches modes
  - Mode 1 -> Mode 2 : $V_D > 0.276$
- Initial conditions determined
  - Take solution from Fsolve
  - Use Eval to evaluate function values
- Continue until each mode visited

Verified Properties

- Choose the property of interest
  - Reason about oscillation
  - Reason about bounded behaviour
- Turn into an inequality
  - Non-oscillation : I_L will never pass an upper bound
  - Bounded Behaviour : I_L and V_C will remain bounded
- Input into MetiTarski

MetiTarski Input

- Transform inequality into the MetiTarski syntax
- Remember: each mode must be checked

For All

Time in a specific mode

Mode Switch Time

```
fof(
 Tunnel,conjecture, ! [X] :
 (
   (0 <= X & X <= 2.39*10^(-9)) =>
   -0.0059 - 0.000016*exp(-2.55*10^8*X) + 0.031*exp(-5.49*10^7*X)
   < 0.03
  )
).
```

Closed form solution

Property inequality

Results

- Property 1
  - Non-Oscillation
- In each mode upper threshold not passed
  - I_L : Current through the inductor

Mode	Variable	Bound	CPU Time (sec.)
1	I_L	U	0.1
2	I_L	U	4.0
3	I_L	U	0.3

Results

Property 2 – Bounded Behaviour

Mode	Variable	Bound	CPU Time (sec.)
1	V_C	U	0.2
1	V_C	L	0.4
2	V_C	U	$\boxed{2.7}$
2	V_C	L	0.6
3	V_C	U	0.3
3	V_C	L	0.5
1	I_L	U	0.5
1	I_L	L	0.3
2	I_L	U	0.6
2	I_L	L	$\boxed{3.9}$
3	I_L	U	0.3
3	I_L	L	0.6

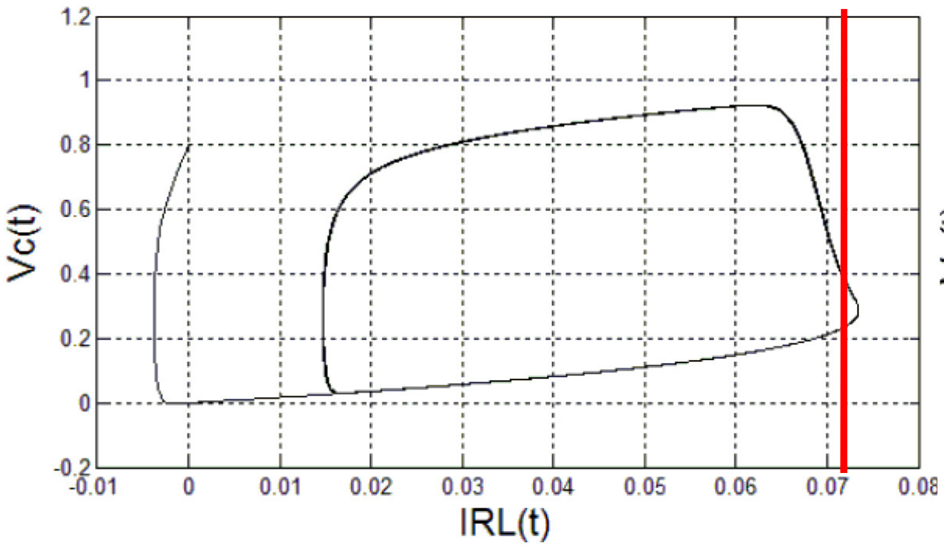
- In each mode the current and voltage are bounded
- Necessary to add axioms in 2 cases.

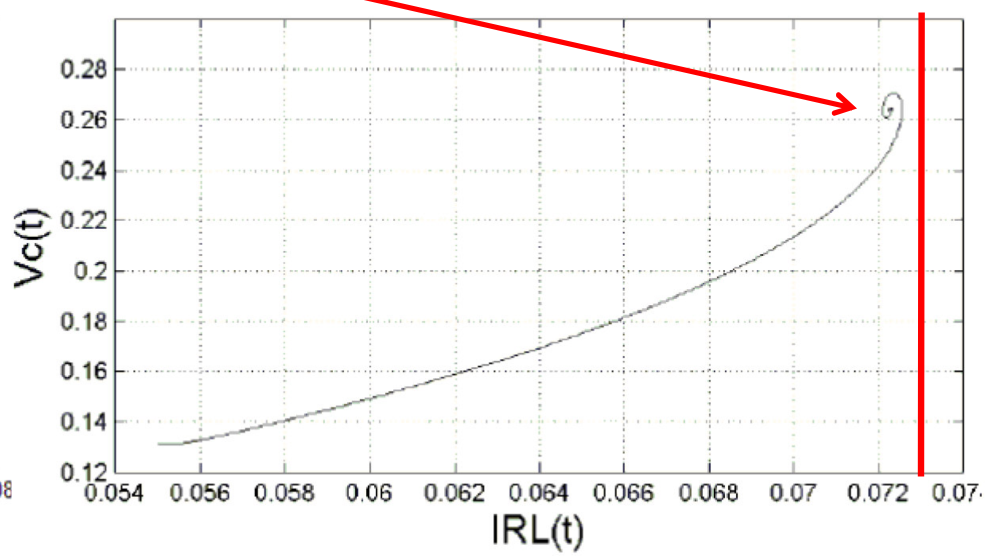
Verified Results

- Recall the property

Non Oscillation

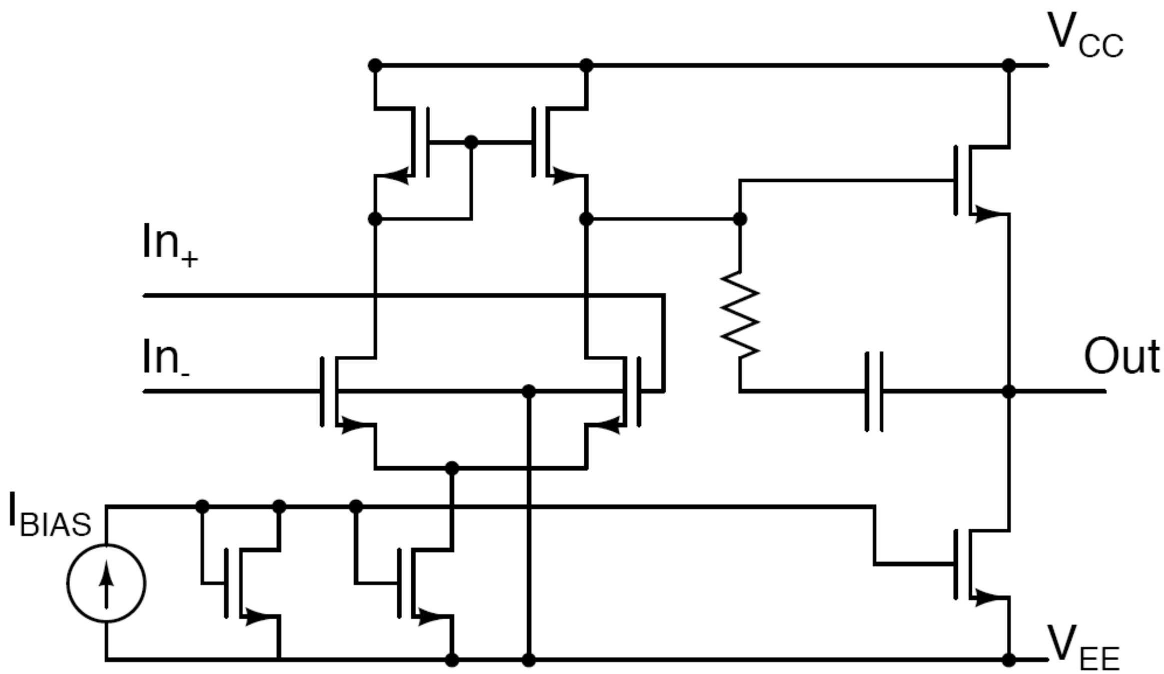
I_L will never pass an upper bound





Results

- Applied methodology to a basic OP-AMP



- Required additional method to obtain a closed form solution.

Outline

- Motivation
- Related Work
- Proposed Methodology
- Brief Introduction to MetiTarski
- Illustrative Example
- Conclusion
- Future Plans

Conclusion

- Developed a methodology for the automated verification of analog designs
  - Algebra system steps are *semi-automated*, but mechanical in nature
  - MetiTarski **completely automated**
  - Most proofs complete quickly
- Applied to several analog circuits
  - Interesting and complex behaviour
  - Two different methods for closed form solutions

Future Plans

- Computing Closed Form Solutions
  - Investigate methods for solving nonlinear ODEs
- Scale to Larger Problems
  - Efficient methods for calculating piecewise linear functions
  - Apply methodology to more precise models

Thank You!

More details at: hvg.ece.concordia.ca