DC Operating Point Analysis – A Formal Approach

Mohamed H. Zaki, Ian M. Mitchell, Mark R. Greenstreet

Computer Science Department, University of British Columbia
Vancouver, BC, Canada
{mzaki, mitchell, mrg}@cs.ubc.ca

Abstract. If the inputs to a circuit are held constant, then the state of some circuits settle to a unique equilibrium: the DC operating point. Many circuit analysis techniques seek to determine or use this state and assume that it is unique. However, for some constant input some circuits may settle to one of several different operating points, while still others may never settle. In this paper we describe a procedure that uses symbolic circuit models generated from a netlist level circuit description to rigorously locate and classify all of the equilibria of a circuit model in order to determine the existence, location and number of DC operating points. Implemented with a collection of public tools (HySAT, INTLAB and EigTool) and our own MATLAB circuit modeling system OOmspice, we demonstrate that the technique can deduce the hysteresis of a Schmitt trigger and the lack of DC operating points for a ring oscillator with an odd number of stages.

1 Introduction

A fundamental problem in the study of circuits is DC operating point analysis: what voltages will the nodes of the circuit settle to if the inputs to the circuit remain indefinitely at their quiescent values? Many critical properties of a circuit's behavior are directly connected to analysing its DC operating point(s). The first question is, how many DC operating points does the circuit have? For some circuits (e.g. oscillators), there should be no stable equilibrium points; thus, such circuits should have no DC operating points. Circuits such as amplifiers, on the other hand, typically have a unique DC operating point, and the small signal analysis to characterize such circuits is carried out at this point. Other circuits that are intended to retain digital state information such as Schmitt triggers, flip-flops and sense amplifiers have multiple, distinct DC operating points corresponding to the distinct, discrete states of the circuit.

In this paper we use ideas from formal verification to analyse DC operating points in a semantically and numerically rigorous fashion. Before doing so, however, we first define DC operating points and then examine some of the difficulties faced by traditional operating point analysis.

1.1 Definitions

We can define the continuous state of a circuit with a vector x where the components of x represent the voltages and/or currents in the circuit. Let n denote the dimension of x.

If we assume that the circuit's behavior is described by an ordinary differential equation (ODE), then we write:

$$\frac{d}{dt}x = f(x, in) \tag{1}$$

where f is a function, possibly non-linear, describing the dynamics of the circuit, and in is a vector of *inputs* to the circuit. If the circuit has m inputs, then in is a function from time to input values: $in \in \mathbb{R}^{\geq 0} \to \mathbb{R}^m$. Likewise, $x \in \mathbb{R}^{\geq 0} \to \mathbb{R}^n$ is a function from time to circuit states. The existence and uniqueness of solutions to Equation 1 are guaranteed for any initial state $x(0)$ as long as f and in satisfy some basic smoothness conditions (see [5] for details). In particular, unique solutions are guaranteed to exist for "reasonable" circuit models, including the ones used as examples in this paper.

For a particular constant input $\underline{in} \in \mathbb{R}^m$, $\underline{x} \in \mathbb{R}^n$ is a DC equilibrium point of the circuit with input $\underline{in}$ iff

$$f(x, \underline{in}) = \mathbf{0} \tag{2}$$

Furthermore, we will call $\underline{x}$ a *stable* equilibrium point iff there is some $\varepsilon > 0$ such that for any $\tilde{x} \in \mathbb{R}^{\geq 0} \to \mathbb{R}^n$ satisfying

$$\begin{array}{rl} \frac{d}{dt}\tilde{x} = f(\tilde{x}, \underline{in}), & \tilde{x} \text{ is a solution to Eq. 1} \\ \|\tilde{x}(0) - \underline{x}\| < \varepsilon, & \tilde{x} \text{ starts "near" } \underline{x} \end{array} \tag{3}$$

then $lim_{t \to +\infty} \tilde{x}(t) = x$. In other words, $\underline{x}$ is a stable equilibrium point if all trajectories starting near $\underline{x}$ converge to $\underline{x}$.

The stability of an equilibrium point $\underline{x}$ can be determined by examining the eigenvalues of the Jacobian operator of f. Let $J(f, \underline{x}, \underline{in}) \in \mathbb{R}^{n \times n}$ be the *Jacobian* matrix for f at state $\underline{x}$ for input $\underline{in}$:

$$J_{i,j}(f, \underline{x}, \underline{in}) = \frac{\partial}{\partial x_j} f_i(x, \underline{in})|_{x=\underline{x}} \tag{4}$$

If all eigenvalues have real parts that are less than zero, then $\underline{x}$ is a *stable* equilibrium point. If any eigenvalue has a positive real part, then we will call $\underline{x}$ an *unstable* equilibrium[1]. If any eigenvalue is purely imaginary (real component of zero), then higher order derivatives must be considered to determine the stability of $\underline{x}$; this situation is a degenerate case that we will not consider further here.

Let in_0 be a vector representing quiescent input values for a circuit, and let

$$X_0(f, in_0) = \{\underline{x} \mid \underline{x} \text{ is a stable equilibrium of } f \text{ with input } in_0\} \tag{5}$$

X_0 is the set of DC operating points for the circuit with input in_0. Note that X_0 may be empty; in which case, the circuit has no stable state. Likewise, X_0 may contain multiple, distinct points; in which case, the circuit can settle to any of these stable states.

A problem related to finding the DC operating points of a circuit is finding the circuit's *DC transfer relation*. In this case, for each input in a specified space, we find the DC operating points for that input. For example, we could sweep the input of an inverter

[1] If there are a mixture of eigenvalues with negative and positive real components, then the circuit can show what are commonly called metastable behaviors, but we will treat such behaviors as unstable for the purposes of this paper.

circuit from ground to power to observe how the output goes from power to ground. As we illustrate in Section 3, this DC transfer relation can provide a simple visualization of the circuit's operation. It is also commonly used for characterizing properties of a circuit such as its noise margin.

1.2 Limitations of Traditional DC Operating Point Analysis

DC operating point analysis is the starting point for circuit simulation tools such as SPICE. When performing transient analysis, SPICE first identifies the DC operating point of the circuit and uses this point as the initial state for the simulation. When performing small-signal analysis, SPICE uses the DC operating point as the point about which to perform the sensitivity analysis. However, this analysis can be problematic, and the problems are both numerical and semantic. The numerical difficulties arise because finding a solution to Equation 2 involves finding the root of a non-linear, vector valued function. The root-search problem becomes intractable for circuits with a moderate number of nodes for anything but relatively trivial dynamics. Thus, various heuristic methods are employed, and they cannot be guaranteed to find a solution.

The semantic problems arise because Equation 2 may have no stable solutions (e.g. for an oscillator), or it may have many solutions (e.g. a Schmitt trigger). Either situation is incompatible with the assumption of most simulation tools that the simulation starts from a uniquely defined initial state. Thus, when attempting to simulate such circuits, simulators such as SPICE report an error that a DC operating point could not be identified. To cope with these "errors" designers typically add initial condition statements (to force particular node voltages or branch currents to particular values), or they avoid the problem by, for example, starting the circuit in a state with no power applied (so that all voltages and currents are uniquely zero), ramping the power supply up to the operating value, and letting the numerical vagaries of the simulation algorithm bring the circuit to some state. Unfortunately, these methods "solve" the problem of finding a DC operating point by imposing additional constraints that do not reflect the behavior of the real circuit. Thus, these techniques can mask true design errors, resulting in a circuit that works in simulation but fails on the test bench.

1.3 Our Approach

From a formal verification perspective, a circuit with multiple DC operating points is simply an example of non-determinism: when power is applied, the circuit will converge to one of these points, but we don't know which one. This non-determinism need not be a problem if our analysis examines the circuit's behavior from an initial state *set* that contains all of these points. Likewise, a circuit may have no DC operating point. This situation means that it never settles to a stable state, which may be the desired behavior; for example, in an oscillator circuit. In this case, we can verify the designer's intention that the circuit has no stable equilibrium points.

In the remainder of this paper, we present a technique for identifying the DC equilibria of a circuit and characterizing the stability of these equilibrium points. We do not claim to solve the intractable problem of finding all roots of a multidimensional, nonlinear function. Instead we use interval arithmetic combined with SAT solving techniques to identify regions that *definitely* contain (stable) equilibrium points, and regions that *may* contain such points. The union of these regions could provide the set of points in the continuous state space to use for further verification. For example, these regions appear to be small enough that each such region could be treated as a point by traditional circuit simulation tools. By computing a collection of regions rather than looking for a unique DC operating point, we provide a simple and semantically sound alternative to traditional operating point analysis.

To demonstrate our techniques, we examine a Schmitt trigger and a ring oscillator. For the Schmitt trigger we are able to definitely determine DC operating points for all but a small subset of input voltages, and to determine the existence of two such operating points for midrange inputs, thus confirming the hysteresis (memory) of the circuit. For n-stage ring oscillators, we are able to determine the existence of two DC operating points for n even and no such points for n odd, as expected.

These results are generated through a software toolchain. We have written some software called OOmspice that takes netlist descriptions and circuit parameters and generates appropriate circuit models for the various subsequent tools, which were written by others and are publicly available: HySAT, INTLAB, and EigTool.

The rest of the paper is organized as follows: Section 2 describes in detail the method and tools used to locate and classify DC equilibrium points of a circuit. Section 3 demonstrates the approach on the Schmitt trigger and ring oscillator examples. Section 4 provides a brief overview of related work.

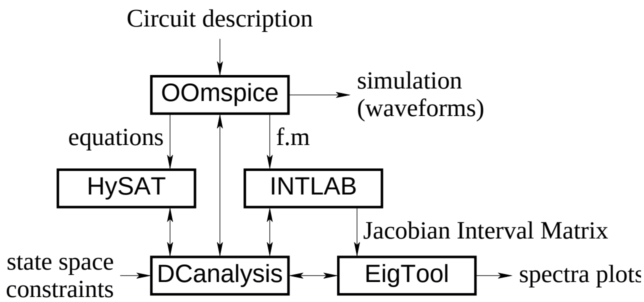


Fig. 1. Analysing DC Equilibria.

2 Finding DC Operating Points

Figure 1 diagrams our approach to finding and classifying DC operating points of a circuit. The designer writes the circuit description as a MATLAB script file using an API provided by our tool, OOmspice. This description is at the netlist level and resembles a traditional SPICE input file. From this description, OOmspice produces two representations for the dynamics of the circuit (f from Equation 1). The first description is a system of symbolic equations that can be manipulated by theorem provers and other verification tools (e.g. HySAT). The second description is a MATLAB function that can be used for simulation or as a description of the dynamics for the INTLAB tools. As described in Section 2.2, we use HySAT to identify regions that *might* contain DC equilibrium points. Then, INTLAB is used to confirm or refute each proposed equilibrium. As described in 2.3, in certain boundary conditions, INTLAB may fail to either confirm or refute the existence of an equilibrium—this situation indicates a boundary case that calls for non-determinism in higher-level abstractions. Finally, we use EigTool to classify each equilibrium point as stable, unstable, or indeterminate. The DCanalysis box in the figure represents the MATLAB functions that we have written to implement these tests by coordinating the interactions of the tools mentioned above. The remainder of this section describes each step of this analysis procedure in greater detail.

2.1 OOmspice

We are currently developing a software tool that we call OOmspice: **O**bject **O**riented **M**ATLAB **Spice**. OOmspice provides a circuit description capability similar to SPICE; by embedding this description language in MATLAB, designers have the expressiveness of a flexible scripting language with loops, conditionals, functions, recursion, etc. Our emphasis in OOmspice is to provide a common framework to produce models for formal verification tools based on circuit descriptions that are accessible to practicing designers. The heart of this translation from netlist to models is an intermediate representation of the circuit dynamics based on symbolic equations. OOmspice first creates a set of equations that corresponds directly to the netlist, and then uses symbolic rewrite techniques to produce an equivalent but simpler model that can be used by other tools. This model can be output as a set of symbolic equations for use by symbolic tools such as HySAT. Alternatively, a MATLAB function can be produced. Using the standard MATLAB integrators, OOmspice can perform circuit simulations with this function; for example, the plots of waveforms in Section 3. This MATLAB function for the dynamics is also used by methods in INTLAB as described in Section 2.3.

2.2 HySAT

HySAT [2] is an *unsatisfiability* solver for boolean combinations of arithmetic constraints. Given such a formula, HySAT will either refute the formula or it will report a region where the formula *might* hold. Therefore, we give HySAT the formula that is satisfied by DC equilibria states. Initially, we give HySAT a formula covering the entire state space. (e.g. each voltage can be anywhere in $[0, V_{dd}]$). If HySAT refutes the formula, we conclude that the circuit has no DC equilibria, and therefore no DC operating points. Otherwise, our analysis performs the following two operations on the candidate region Q (a hyperrectangle) that HySAT has identified:

1. Forward Q to INTLAB for further analysis as described in Section 2.3.
2. Augment the formula that we ask HySAT to analyse to exclude points in Q.

In this manner, we obtain candidate regions that may contain DC equilibria until HySAT establishes that the remaining space does not contain any equilibrium points.

2.3 INTLAB

For each candidate region Q identified by HySAT, we use INTLAB [6] to further refine the analysis. For each region considered, INTLAB can produce one of the following results:

Refutation: In particular, INTLAB can show that there are one or more components of $\frac{dx}{dt}$ that do not change sign in the hyperrectangle proposed by HySAT, and therefore that the hyperrectangle cannot contain an equilibrium.

Narrowing: INTLAB can produce a hyperrectangle H that is contained in Q and is guaranteed to contain an equilibrium point. Recall that the region proposed by HySAT is only a candidate—it may be devoid of any actual equilibrium points. In our experience, the region H produced by INTLAB is usually much smaller than the region originally proposed by HySAT. In fact, candidate HySAT regions comparable in size with INTLAB solutions can be obtained through a refined tuning of HySAT options; however, achieving these results in HySAT often dramatically increased its execution time, while INTLAB could perform this narrowing very quickly.

Corner Cases: It is important to note that while HySAT is guaranteed to *refute* the existence of DC equilibrium points if none exist, INTLAB is only guaranteed to *verify* the presence of such points if they do. Thus, there is a possibility that INTLAB may neither refute nor confirm the presence of a DC equilibrium in a region proposed by HySAT. One possible cause of this situation is that Q contains a multiple root or closely spaced roots of the function f. This situation indicates a boundary case that calls for non-determinism in higher-level abstractions. We give such an example in Section 3.1. Alternatively in the case of closely spaced roots, INTLAB may generate an H containing one root but give no indication that other roots exist in the region, thus causing these roots to be missed. We currently do not detect this situation, and it remains a topic for future work.

For each hyperrectangle H that INTLAB produces that contains an equilibrium, INTLAB further produces a Jacobian interval matrix J_H for that hyperrectangle. For any point $x \in H$ each element of the Jacobian matrix of f at x is contained in the interval that is the corresponding element of J_H. In this sense, J_H *contains* the Jacobian matrices for all points in H. INTLAB provides a procedure for the Jacobian calculation of non-linear functions based on automatic differentiation. Automatic differentiation is an approach to obtain the derivative of the function through a sequence of symbolic derivation and evaluation (Interval evaluation in INTLAB) of the basic function constituents, which are combined in accordance with the chain rule [20]. As described in Section 2.4, we use J_H and routines in EigTool to classify each equilibrium point identified at this step.

2.4 Hurwitz methods and EigTool

To classify an equilibrium as a DC operating point or not, the stability of that equilibrium must be determined. As described earlier, the stability of an equilibrium depends on the real component of the eigenvalues of the Jacobian matrix of the dynamics of the circuit at that equilibrium. For a single equilibrium point such classification involves merely computing the eigenvalues of a matrix. Unfortunately, we do not know the precise location of the equilibrium, but rather a box H which contains it. Therefore, we examine the eigenvalues of all Jacobian matrices for all states in H. This set of Jacobian matrices is represented as an interval matrix $J_H = [A_C - \Delta, A_C + \Delta]$.

By examining potential eigenvalues of J_H, we seek to categorize the stability of any equilibrium in H into one of three classes: definitely stable, definitely unstable, and unknown. The former category has been well-studied in the control literature (see, for example [19]); in simple form, we wish to demonstrate that all eigenvalues of all matrices within J_H have negative real components. Although no necessary and sufficient conditions for stability of non-symmetric interval matrices are available, there is a simple sufficient test. We say that an interval matrix $[A_C - \Delta, A_C + \Delta]$ is stable if (the "Hurwitz test"):

$$\lambda_{\max}(A'_C) + \rho(\Delta') < 0, \tag{6}$$

where $A'_C = 1/2(A_C + A_C^T)$ and $\Delta' = 1/2(\Delta + \Delta^T)$ are the symmetric components of the center and offset matrices respectively, $\lambda_{\max}(A)$ is the maximum eigenvalue of the symmetric matrix A and $\rho(A)$ is the spectral radius of matrix A [21]. If this test is passed then H contains a DC operating point.

To declare that H does not contain a DC operating point, we would like to demonstrate that *all* matrices in J_H have at least one eigenvalue with positive real component and hence that the DC equilibrium is unstable. Unfortunately, this property does not seem to have been studied in the interval matrix literature. While we intend to investigate whether existing methods can be modified to test for this property, for now we have turned to a less discerning but still rigorous method of demonstrating instability using the concept of matrix pseudospectra.

The spectrum $\Lambda(A)$ of a (non-interval, non-symmetric) matrix A is its set of eigenvalues:

$$\Lambda(A) = \{z \in \mathbb{C} \mid \det(zI - A) = 0\}.$$

There are several equivalent definitions of the ε-pseudospectrum (for some $\varepsilon > 0$) [1], but for our purposes the most useful is

$$\Lambda_\varepsilon = \{z \in \mathbb{C} \mid z \in \Lambda(A + E) \quad \text{for some } E \text{ with } \|E\| \le \varepsilon\}.$$

In other words, the ε-pseudospectrum contains all points which are eigenvalues of matrices $A + E$ for some matrix E with ε bounded norm. Although any norm can be used in the definition, we will focus on the 2-norm pseudospectrum because tools are available to compute it; in particular, we will use EigTool [24].

To demonstrate that all matrices within interval matrix $J_H = [A_C - \Delta, A_C + \Delta]$ have an eigenvalue with a positive real component, we examine the ε-pseudospectrum of A_C with $\varepsilon = \|\Delta\|_2$. If that set contains at least one component lying entirely in the right halfplane, then every matrix in the interval matrix must have at least one unstable eigenvalue and we can classify any equilibrium as unstable. Note that like the Hurwitz test, the pseudospectrum is also a sufficient test, because the set of matrices $A_C + E$ for $\|E\|_2 \le \|\Delta\|_2$ is a superset of the interval matrix $[A_C - \Delta, A_C + \Delta]$. In fact, the pseudospectrum can also be used to diagnose stability: If all components of the ε-pseudospectrum lie in the left halfplane then the interval matrix is definitely stable.

The Hurwitz test (Equation 6) is equivalent to examining the real axis component of the pseudospectrum.

In this paper we use EigTool to graphically plot the desired pseudospectrum (or a pseudospectrum for larger ε) and perform the test visually. If we are unable in the future to find a direct test for instability of interval matrices, the algorithms underlying EigTool could be adapted to produce a more efficient and automated pseudospectrum-based test by just looking for intersections of the ε-pseudospectrum with the complex axis.

3 Experimental Results

We demonstrate the proposed methodology on two basic circuits: an (inverting) Schmitt trigger, and ring oscillators built from chains of inverters. For the Schmitt trigger, we examine the effect of the input signal on the number and properties of DC equilibria. For the ring oscillator, we examine the effect of the number of inverters on the number and properties of the DC equilibria. We use a simple, first-order transistor model:

$$i_{ds} = \frac{kW}{2L}(V_{gsx}^2 - V_{gdx}^2)$$

where:

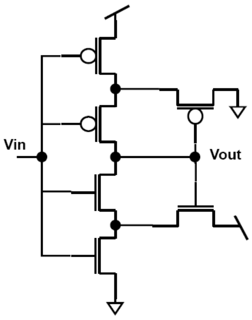
$k = 270 \times 10^{-6} \frac{A}{V^2}$,	for nfet	$k = -90 \times 10^{-6} \frac{A}{V^2}$,	for pfet
$W =$ gate width,		$L =$ gate length	
$V_s =$ source voltage		$V_g =$ gate voltage	
$V_d =$ drain voltage			
$V_{th} = 0.4V$,	threshold for nfet	$V_{th} = -0.4V$,	threshold for pfet
$V_{gse} = V_g - V_s - V_t$		$V_{gde} = V_g - V_d - V_t$	
$V_{gsx} = \max(Vgse, 0)$,	for nfet	$V_{gsx} = \min(Vgse, 0)$,	for pfet
$V_{gdx} = \max(Vgde, 0)$,	for nfet	$V_{gdx} = \min(Vgde, 0)$,	for pfet

The values for k and V_{th} are typical for a 180nm CMOS process. The power supply voltage is $1.8V$. For the circuits described here, all n-channel transistors have a shape factor, W/L, of $\frac{8}{3}$, and all p-channel transistors have a shape factor of $\frac{16}{3}$. Each node has a capacitance estimated as a weighted sum of the widths of the gates and drains/sources connected to the node.

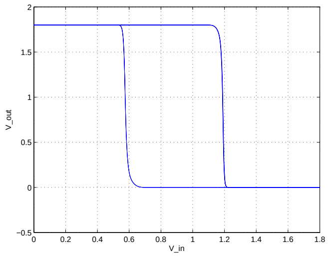
The public tools used in the analysis have a number of parameters which affect their execution; for example, HySAT has parameters which determine the terminal level of refinement, and EigTool has parameters which determine which pseudospectra to plot. At present, these parameters are chosen manually for each example, because no single fixed parameter value is appropriate for all examples. Automatic determination of suitable parameters is a topic for future study.

3.1 Schmitt Trigger

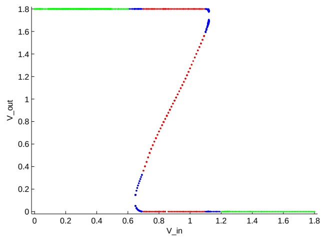
Figure 2(a) shows an inverting Schmitt trigger circuit whose output behavior exhibits hysteresis; for the range of input voltages where this hysteresis occurs, the output voltage depends on the past output voltage. Starting with input low and output high, a rising input will eventually cause the output to go low; if the input is now lowered the output will remain low until the input crosses some much lower voltage threshold; for a plot of the input/output characteristics see figure 2(b). The hysteretic switching behavior is



(a) Schematic.



(b) Simulation.

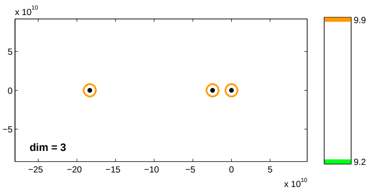


(c) Equilibrium Regions.

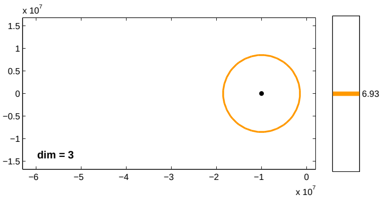
Fig. 2. Schmitt trigger.

useful in level-crossing detectors because it reduces the chances of output chattering. Schmitt triggers are also used in the implementation of relaxation oscillators or multi-vibrator circuits.

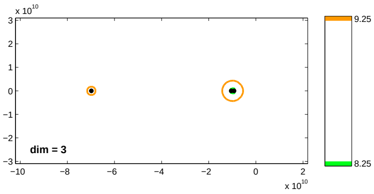
The DC transfer function of the Schmitt trigger can be determined by examining the location and properties of its DC equilibria for fixed inputs. For each input voltage, we perform the analysis described in the previous section: locate potential DC equilibria and then examine their stability. We sampled the DC characteristics of the Schmitt trigger for V_{in} from 0V to 1.8V on a grid with step size 0.01V, and additional refinement of the grid was performed near the bifurcation regions to narrow the interval in which our results were inconclusive. The range of V_{out} in which the DC equilibria lie for each sampled V_{in} are shown in Figure 2(c). The circuit exhibits five qualitatively distinct regions of behavior based on the number of equilibria. There is a single equilibrium region with high output for $V_{in} \in [0, 0.642]$V, three distinct equilibrium regions for $V_{in} \in [0.649, 1.121]$V, and a single equilibrium region with low output for $V_{in} \in [1.127, 1.8]$V. In the narrow intervals $V_{in} \in [0.642, 0.649]$V and $V_{in} \in [1.121, 1.127]$V HySAT returns a potential equilibrium region which is not validated by INTLAB to contain an equilibrium, and so our tests are inconclusive. A higher-level abstraction of the Schmitt trigger should describe a non-deterministic behaviour in these inconclusive regions. For example, when $V_{in} \in [0.642, 0.649]$V, the output will remain high if it was previously high. On the other hand, if the output had been low, then it *might* remain low, or it *might* transition to a high-value. Using non-determinism in this fashion captures the essential properties of the Schmitt trigger circuit without over-constraining its behavior.



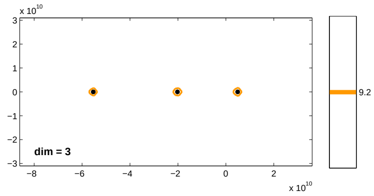
(a) The (stable) DC equilibrium region at $V_{in} = 1.8$V (where $\log_{10}(\|\Delta\|_2) = 6.92$).



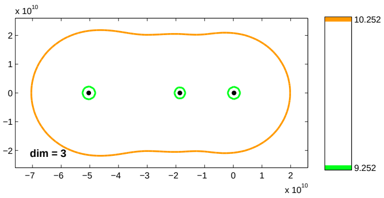
(b) Close Investigation around the right-most eigenvalue for the (stable) DC equilibrium region at $V_{in} = 1.8$V.



(c) The (stable) DC equilibrium region at $V_{in} = 0.6$V (where $\log_{10}(\|\Delta\|_2) = 6.82$).



(d) The (unstable) DC equilibrium region at $V_{in} = 1$V (in the hysteresis region) that corresponds to an intermediate V_{out} (where $\log_{10}(\|\Delta\|_2) = 7.75$).



(e) The (indeterminant) DC equilibrium region at $V_{in} = 1.124$V (near a bifurcation point) that corresponds to a high V_{out} (where $\log_{10}(\|\Delta\|_2) = 10.25$).

Fig. 3. Pseudospectra of the interval Jacobians of various DC equilibrium points of the Schmitt trigger, generated with EigTool.

Next, we examine the stability properties of the equilibria using EigTool. Figure 3 shows the pseudospectra for selected regions containing DC equilibrium points. In each plot are one or more points representing the eigenvalue(s) of the center matrix A_C for the interval Jacobian for the equilibrium region. Surrounding each eigenvalue are one or two contours showing the extent of the ε-pseudospectrum for some $\varepsilon = 10^{\rho}$, where ρ is specified in the bar beside the plot. The stability of an eigenvalue of the interval Jacobian is conclusively determined if the contour corresponding to the 10^{ρ}-pseudospectrum does not cross the imaginary axis for $\rho = \log_{10}(\|\Delta\|_2)$.

For example, figure 3(a) shows the pseudospectrum of the interval Jacobian for the single DC equilibrium region located for $V_{in} = 1.8$V. In this case $\log_{10}(\|\Delta\|_2) \approx 6.9$ and the largest contour corresponds to $\rho = 9.9$. The circular contours containing the two leftmost eigenvalues are clearly in the left halfplane and hence represent stable behaviors; however, the contour containing the rightmost eigenvalue appears to straddle the imaginary axis. In order to better characterize this eigenvalue, figure 3(b) zooms in to show the pseudospectrum for $\rho = 6.97$ (so $\varepsilon > \|\Delta\|_2$ still holds) and it becomes clear that this eigenvalue is also in the left halfplane. Consequently any DC equilibrium in the specified region of the state space is stable as expected. This fact can also be confirmed by applying the Hurwitz test to the interval Jacobian.

We can draw a similar conclusion about the single equilibrium region that exists for $V_{in} = 0.6$V from figure 3(c). Note that in this case the leftmost pseudospectrum contour actually contains two eigenvalues, which merely implies that either eigenvalue could be anywhere within this contour.

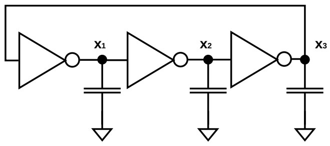
For $V_{in} = 1$V HySAT identifies and INTLAB validates three regions containing equilibria. The pseudospectra of two of these (corresponding to the highest and lowest V_{out}) look similar to those examined above and hence the corresponding equilibria can be classified as stable. The pseudospectrum for the final region is shown in figure 3(d). Since the rightmost eigenvalue is clearly in the right halfplane, we can classify this equilibrium as unstable, and the circuit behaves as expected.

As was mentioned above, there are two regions of the input voltage for which the equilibrium location procedure was inconclusive: HySAT identified (large) regions in which an equilibrium could not be ruled out, but INTLAB was unable to validate those equilibria. An obvious approach would be to treat those regions as suspect and examine their interval Jacobians. Figure 3(e) shows the pseudospectrum for the interval Jacobian of the potential equilibrium region identified by HySAT for $V_{in} = 1.124$. Because the equilibrium region is large, the width of the interval matrix is also large ($\log_{10}(\|\Delta\|_2) \approx 10.25$) and hence so is the relevant pseudospectrum. It is not surprising but perhaps comforting that the stability analysis procedure is just as inconclusive as the equilibrium location procedure in this case: all of the eigenvalues could lie on either side of the imaginary axis.

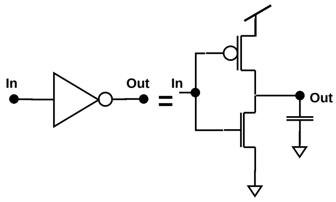
3.2 Ring Oscillators

Ring oscillators are often used to characterize fabrication processes and are adapted to form voltage controlled oscillators in phase locked loops. As illustrated in Figure 4(a), a ring oscillator is built from inverters connected in series, with the final inverter's output feeding back to the first inverter's input. It should be intuitively clear that for an even number of stages (inverters) the ring will become stuck (with half the outputs high and half low) while for an odd number of stages the voltages will oscillate.

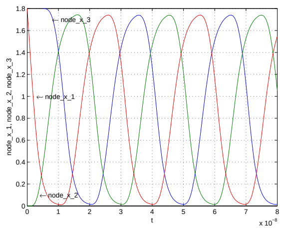
We apply the DC equilibrium location procedure for ring oscillators with different numbers of inverters. Odd rings were found to have a single region containing a DC equilibrium point while even rings were found to have three such regions. The stability of the equilibrium in each of these regions can then be analyzed. Figure 5(a) shows



(a) Schematic.

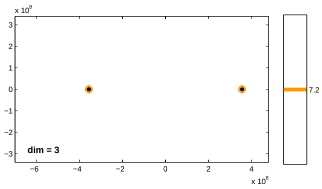


(b) Inverter Cell.

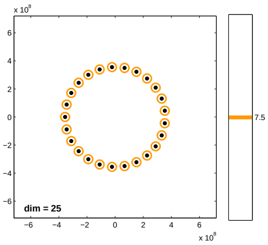


(c) Simulation.

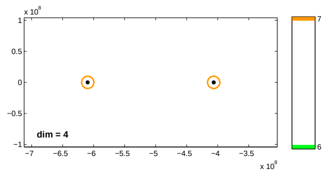
Fig. 4. 3-Stage Ring Oscillator



(a) The (unstable) DC equilibrium point of a 3-stage oscillator (where $\log_{10}(\|\Delta\|_2) = 5.376$).



(b) The (unstable) DC equilibrium point of a 25-stage oscillator (where $\log_{10}(\|\Delta\|_2) = 6.22$).



(c) The (stable) DC equilibrium point of a 4-stage oscillator (where $\log_{10}(\|\Delta\|_2) = 6.44$).

Fig. 5. Pseudospectra of the interval Jacobians of various DC equilibrium points of ring oscillators with various numbers of inverters, generated with EigTool.

the pseudospectrum for the interval Jacobian of the single DC equilibrium region of a 3-stage oscillator, and it is clear that this equilibrium is unstable because there is an eigenvalue in the right halfplane. Similar results were found for other odd-stage oscillators; for example, figure 5(b) shows the pseudospectrum for a 25-stage oscillator. Therefore we conclude that odd-stage oscillators (with no output load) will not get stuck at a DC equilibrium point.

The same is not true for even-stage oscillators. Figure 5(c) shows the pseudospectrum for one of the DC equilibrium regions for a 4-stage oscillator. Both eigenvalues are in the left halfplane, so this equilibrium is stable. Of the other two DC equilibrium regions for this oscillator, one was similarly stable and one was unstable. Intuitively this result makes sense—one of the stable regions corresponds to the first inverter's output being high and the other stable region to this output being low. Similar stability results were confirmed by the Hurwitz test for other even-stage oscillators. From these results we conclude that even-stage oscillators are likely to become stuck in stable equlibria.

4 Related Work

In recent years, the growing interest in developing formal methods for the verification of analog and mixed signal (AMS) designs has led to a wide variety of ideas [26]. Overapproximation of the reachable states using geometric shapes initially proposed in [9] was further developed and implemented in several tools [4, 15, 16, 25] among others. Those reachability methods were used to verify several interesting properties like bounds of currents and voltages across the circuits, checking stability, and switching conditions. Fixed point inclusion was the primary mathematical tool to demonstrate the oscillatory behavior that characterizes certain circuits [3, 15]. Unlike exhaustive reachability analysis, we can deduce properties like stability or instability (a precondition for oscillation) only by inspecting the corresponding DC behavior. One advantage of our method is that we avoid the time bounded requirement associated with reachability analysis. Recently, SAT and SAT-modulo-theories approaches have been applied for bounded model checking (BMC) [12, 13] of AMS designs, where for each step an SMT procedure is called to check for property satisfaction. Unfortunately, such approach requires the circuit to be described in an oversimplified manner. Also, inherited from BMC is its partial verification capability as a complete state space coverage cannot be guaranteed.

From another verification perspective, extending analog simulation with assertions has recently gained a lot of attention. Initiated with the work done in [17], assertion based verification was applied to solve a variety of problems [11, 7, 8] that required monitoring design constraints like voltage bounds and timing requirements. While it was applied to industrial size designs, assertion methods inherit many problems common to simulation-based validation, such as coverage limitations and the lack of confidence towards the approximate results provided by simulators.

The problem of finding the DC operating point is critical to the analysis of circuits, and remains an active area of research (e.g. [18]). From a verification perspective, [14] employed a search-based approach to verify the start-up conditions for a differential ring-oscillator. Recent work has explored finding the DC behavior using SAT based methods [23]. While our approach also uses SAT techniques, our methods go beyond [23] by providing characterization of the DC equilibria and the ability to work directly with non-linear models rather than piecewise linear approximations.

Literature touching the different aspects of the formal verification of analog circuits is quite wide and spans through many different research domains. We highlighted only the most relevant work while in depth investigations can be found in references therein. Also the survey [26], can be helpful for the comparison of different approaches.

5 Conclusion

In this paper we have presented a preliminary procedure for locating the equilibria of an analog circuit model and classifying their stability in order to find the DC operating point(s) of that circuit. This procedure involves integration of a diverse set of tools and techniques from formal verification. The procedure was able to successfully locate and classify the equilibria for all but two tiny intervals (of width 0.007V and 0.006V out of a total range of 1.8V) of the input for a Schmitt trigger, and for ring oscillators with moderate numbers of stages (up to 25 so far). The analysis procedure has so far proved surprisingly scalable; for example, the verification of the 25 stage oscillator model required less than five minutes, and we have not yet encountered circuits where the verification takes significantly longer or returns indeterminant results for significant parts of the state space. The current bottleneck is generation of the symbolic model in OOmspice (about 20 minutes for the 25 stage oscillator), and we are working on our code to increase the efficiency of this step.

There are a number of limitations that we plan to address in future work. First, we are currently using a very simple transistor model because of its convenient symbolic form, but we would like to include the option of more complex, empirically derived models. Second, we need to treat the possibility of multiple roots more rigorously. Third, we would like to automate the instability test that is currently performed visually on the output of EigTool. Fourth, we need to provide an automatic translation between OOmspice netlist and Spice netlist that will be helpful when comparing the presented approach with the Spice based heuristic methods for DC analysis. Finally, we plan to tackle circuits with more complex behaviors; for example, we are currently working on the the Rambus oscillator and an arbiter.

References

1. M. Embree and N. Trefethen. Pseudospectrua gateway, http://web.comlab.ox.ac.uk/pseudospectra/index.html
2. M. Franzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert. Efficient Solving of Large Non-linear Arithmetic Constraint Systems with Complex Boolean Structure, in Journal on Satisfiability, Boolean Modeling and Computation, Vol.1, pp. 209-236, 2007.
3. S. Gupta, B.H. Krogh, and R.A. Rutenbar. Towards Formal Verification Of Analog Designs, in Proc. IEEE/ACM International Conference on Computer Aided Design, pp. 210-217, 2004.
4. W. Hartong, K. Klausen, and L. Hedrich. Formal Verification For Nonlinear Analog Systems: Approaches To Model And Equivalence Checking, in Advanced Formal Verification. Kluwer, pp. 205-245, 2004.

5. M.W. Hirsch and S. Smale. Differential Equations, Dynamical Systems, and Linear Algebra, Academic Press, 1974.
6. INTLAB- INTerval LABoratory, in Developments in Reliable Computing (T. Csendes, ed.), Kluwer, pp. 77-104, 1999.
7. A. Jesser, S. Laemmermann, A. Pacholik, R. Weiss, J. Ruf, W. Fengler, L. Hedrich, T. Kropf, and W. Rosenstiel. Advanced Assertion Based Design for Mixed-Signal Verification. in IEICE Trans. Fundamentals: Special Section on VLSI Design and CAD Algorithms, Vol.E91-A(12), pp. 3548-3555, 2008
8. K. Jones, V. Konrad, and D. Nickovic. Analog Property Checkers: A DDR2 Case Study. In Proc. Formal Verification of Analog Circuits (FAC), Princeton, 2008
9. R. P. Kurshan and K. McMillan. Analysis Of Digital Circuits Through Symbolic Reduction, in IEEE Transactions on Computer-Aided Design, Vol.10, pp. 1350-1371, 1991.
10. Matlab User Guide: http://www.mathworks.com/
11. R. Mukhopadhyay, S.K. Panda, P. Dasgupta, and J. Gough. Instrumenting AMS Assertion Verification on Commercial Platforms. in ACM Trans. Design Automation of Electronics Systems, Vol.14(2), pp.1-47, 2009.
12. D. Walter, S. Little, C. Myers, N. Seegmiller, and T. Yoneda. Verification of Analog/Mixed-Signal Circuits Using Symbolic Methods, in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.27(12), pp.2223-2235, 2008.
13. J. Schaffnherr, M. Freibothe, B. Straube, J. Bormann. Semi-formal verification of the steady state behavior of mixed-signal circuits by SAT-based property checking. in Theoretical Computer Science, Elsevier, Vol.404(3): 293-307, 2008.
14. M.R. Greenstreet, S. Yang. Verifying start-up conditions for a ring oscillator. in ACM Great Lakes Symposium on VLSI, pp. 201-206, 2008.
15. G. Frehse, B.H. Krogh, and R.A. Rutenbar. Verifying Analog Oscillator Circuits Using Forward/Backward Abstraction Refinement, in IEEE/ACM Design, Automation and Test in Europe, pp. 257-262, 2006.
16. T. Dang, A. Donze, and O. Maler. Verification of Analog and Mixed-signal Circuits using Hybrid System Techniques, in Formal Methods in Computer-Aided Design, LNCS 3312, pp.14-17, Springer, 2004.
17. O. Maler, and D. Nickovic. Monitoring Temporal Properties of Continuous Signals. in Formal Modelling and Analysis of Timed Systems, LNCS 3253, pp.152-166, Springer, 2004.
18. W. Ma, L. Trajkovic, and K. Mayaram. HomSSPICE: A Homotopy-Based Circuit Simulator For Periodic Steady-State Analysis Of Oscillators, in Proc. IEEE Int. Symp. Circuits and Systems, pp. I-645-I-648, 2002.
19. M. Mansour. Robust stability of interval matrices, in IEEE Proc. of Conference on Decision and Control, pp. 46-51, 1989.
20. L. Rall. Automatic Differentiation: Techniques and Applications, LNCS 120, Springer, 1981.
21. J. Rohn. A Handbook of Results on Interval Linear Problems. http://uivtx.cs.cas.cz/ rohn/handbook/
22. J.G. Tront. PSpice for Basic Circuit Analysis. McGraw-Hill, 2005.
23. S.K. Tiwary, A. Gupta, J.R. Phillips, C. Pinello, and R. Zlatanovici. fSpice: A Boolean Satisfiability Based Approach For Formally Verifying Analog Circuits, in Proc. Workshop on Formal Verification of Analog Circuits (FAC), Princeton, 2008.
24. T.G. Wright. EigTool. http://www.comlab.ox.ac.uk/pseudospectra/eigtool/, 2002.
25. C. Yan, M.R. Greenstreet. Verifying an Arbiter Circuit, in IEEE Proc. Formal Methods in Computer-Aided Design, pp.1-9, 2008.
26. M. Zaki, S. Tahar, and G. Bois, Formal Verification Of Analog And Mixed Signal Designs: A Survey, Microelectronics Journal, vol. 39(12), pp. 1-10, 2008.