Finding All DC Operating Points Using Interval Arithmetic Based Verification Algorithms

Itrat A. Akhter*, Justin Reiher*, Mark R. Greenstreet*
*The University of British Columbia

Abstract—**This paper applies interval-arithmetic based verification algorithms to circuit verification problems. In particular, we use Krawczyk's operator to find all DC operating points of CMOS circuits. We present what we believe to be the first, completely automatic verification of the Rambus ring-oscillator start-up problem. Comparisons with the dReal and Z3 SMT shows large performance and scalability advantages to the interval verification approach. We provide an open-source implementation that supports state-of-the-art short-channel device models.**

Key words—**AMS verification, formal verification, interval arithmetic, SMT solvers.**

I. Introduction

DC operating point analysis is a crucial first step for simulation, analysis, and verification of analog and mixed-signal circuits. Current designs can make frequent mode transitions where digital control signals alter the operating points of analog circuits to optimize power and performance. If a circuit has spurious, stable operating points, such mode transitions can cause design failures (e.g. see [1]). There are two main approaches to finding DC operating points: homotopy-based approaches and SAT and SMT based methods [2]–[4]. Homotopy methods derive a variation of the circuit for which finding the operating point is straightforward. A homotopy function is used to interpolate between the simple model and the actual circuit. Under appropriate conditions, the DC operating point(s) will move smoothly as the interpolation parameter is varied. Prior work has shown methods for finding all or many of the DC operating points using homotopy methods [5]–[7] for circuits with up to a few dozen DC operating points. In Section III we present results for circuits with hundreds of operating points.

SAT based methods [3], [4] typically divide the range of voltage values for each node in a circuit into a moderate number of intervals and compute relations that must hold on these quantized voltages at a DC operating point. The runtime for such methods grows rapidly as finer quantizations are used. Many SMT solvers extend SAT with decision procedures involving real arithmetic. For example, Z3 [8], [9] uses Collin's cylindrical algebraic decomposition algorithm [10] as a decision procedure for systems of polynomial equalities and inequalities. Other researchers have used interval arithmetic in an SMT context [11], [12]. Zaki *et al.* reported on using such solvers for DC operating point problems [2] and found that they could only solve very simple examples.

We present an approach based on interval arithmetic and interval verification algorithms [13], [14]. Specifically we use the Krawczyk method [13, Sec. 13] that can prove the existence and uniqueness of solution to a system of non-linear equations given interval bounds. Our bisection based search is similar to that of iSAT [11] or dReal [12]. By using the Krawczyk method, our solver can establish the existence of a unique solution to a system of non-linear equations in a hyper-rectangle. The Krawczyk method also provides faster convergence than bisection alone, allowing our approach to scale to problems with hundreds of solutions.

The contributions of our work are as follows:

- We apply interval verification algorithms [13] to the verification of analog and mixed signal circuits. While other researchers have used interval arithmetic, we believe that our work is the first demonstration of these verification algorithms for analog circuit verification.
- We present the first, completely automatic verification of the Rambus Ring Oscillator challenge problem. Prior work includes a proof that relies on manual proofs of monotonicity properties [15], adding assumptions that DC equilibria in certain regions must be unstable [3], or using particle filtering methods that provide a visualization, but not a proof [16].
- To provide further comparison with other solvers, we use our tool to analyze a Schmitt trigger circuit.

Algorithm 1 Solver

procedure SOLVER($f, \mathbf{H}_0$)
    solutions $\leftarrow \emptyset$
    workList $\leftarrow$ new stack()
    workList.push($\mathbf{H}_0$)
    while not workList.empty() **do**
        $\mathbf{H} \leftarrow$ workList.pop()
        [status, $\mathbf{H}_K$] $\leftarrow$ Krawczyk(f, $\mathbf{H}$, solutions)
        if status == 'unknown' **then**
            [$\mathbf{H}_1$, $\mathbf{H}_2$] = bisect($\mathbf{H}_\mathbf{K}$)
            workList.push($\mathbf{H}_1$)
            workList.push($\mathbf{H}_2$)
        end if
    end while
    return solutions
end procedure

This work was supported in part by the Institute for Computing, Information and Cognitive Systems (ICICS) at UBC and grants from Intel and NSERC.

978-3-9819263-2-3/DATE19/©2019 EDAA

II. Solving Non-Linear Systems of Equalities

We solve for DC operating points using modified nodal analysis. A circuit is a collection of capacitors and "devices". Devices include MOSFETs and inverter macro-models that are voltage-controlled current sources. Let $I_{\text{dev}}(V)$ be the current flowing in into a node from the devices given node voltages V. From Kirchoff's current law $I_{\text{dev}}(V) + C\dot{V} = 0$. At a DC equilibrium point, $\dot{V} = 0$. Thus, we are looking for solutions to the system of equation $I_{\text{dev}}(V) = 0$. Algorithm 1 gives an overview of our approach. This section first describes the Krawczyk method and then presents the overall algorithm.

1) The Krawczyk Method: The interval-arithmetic notation we use in this paper is based on [13]. We write $\mathbb{IR}$ to denote the set of intervals over the reals, $\mathbb{IR}^n$ to denote the set of n-dimensional hyper-rectangles, and $\text{mid}(\mathbf{X})$ to denote the midpoint of hyper-rectangle $\mathbf{X}$.

Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a differentiable function (e.g. I_{dev}). We write $F : \mathbb{IR}^n \to \mathbb{IR}^m$ to denote the *interval extension* of f where each arithmetic operation and elementary function is replaced with its interval arithmetic equivalent. It is straightforward to show (see [13, Eq. 5.17]) that $x \in \mathbf{X} \Rightarrow f(x) \in F(\mathbf{X})$ For each device in our examples, $I_{dev}(V)$ is monotonic in each of its arguments. In this case, tighter bounds for $I_{dev}(\mathbf{V})$ can be obtained by evaluating each device model at the corners of $\mathbf{V}$ that minimize and maximize the node currents.

The Krawczyk method is an iterative method that computes interval bounds for the solution of a system of non-linear equations. Let $\mathbf{X}^{(k)}$ be the interval bounds after the k^{th} iteration. Then given an interval at iteration k, $\mathbf{X}^{(k)}$:

$$\begin{aligned} K(\mathbf{X}^{(k)}) \;\; = \;\; & \text{mid}(\mathbf{X}^{(k)}) - Cf(\text{mid}(\mathbf{X}^{(k)})) \\ & +(I - C\mathbf{J}_f(\mathbf{X}^{(k)}))(\mathbf{X}^{(k)} - \text{mid}(\mathbf{X}^{(k)})) \end{aligned} \tag{1}$$

K denotes the Krawczyk operator; $C = J_f(\text{mid}(\mathbf{X}^{(k)}))^{-1}$ is a preconditioning matrix; I is the identity matrix; and $\mathbf{J}_f(\mathbf{X}^{(k)}))$ is the interval Jacobian matrix at $\mathbf{X}^{(k)}$. The Krawczyk operator has the following properties (see [14, Thm. 8.2]): if $K(\mathbf{X}^{(k)}) \subset \mathbf{X}^{(k)}$, then $f(x) = 0$ has a unique solution for $x \in \mathbf{X}^{(k)}$; if $(K(\mathbf{X}^{(k)}) \cap \mathbf{X}^{(k)}) = \emptyset$, then $f(x) = 0$ has no solution for $x \in \mathbf{X}^{(k)}$; otherwise, any solution of $f(x) = 0$ in $\mathbf{X}^{(k)}$ is in $K(\mathbf{X}^{(k)}) \cap \mathbf{X}^{(k)}$.

We implemented the Krawczyk operator using rounded interval arithmetic where a unit of least precision is subtracted from the lower bound of an interval and added to the upper bound for rigorous interval bounds in our computation [17]. Our implementation of the Krawczyk method first checks if it can refute the region by testing $0 \notin F(\mathbf{X})$. This check yields a significant performance improvement because a single interval evaluation of f is, generally, much faster than evaluating the Krawczyk operator. If $0 \in F(\mathbf{X})$, then the Krawczyk operator is applied and returns a status of 'unique', 'unknown', or 'infeasible'. When a 'unique' solution is found, it is added to the set of solutions. Conversely, if $\mathbf{X}^{(k)}$ has been shown to be 'infeasible', no further search of that hyper-rectangle is needed. Finally, when the status is 'unknown', the Krawczyk operator reduces the volume of the search region, often by a fairly large factor. One could be tempted to apply the Krawczyk operator repeatedly to avoid the potential exponential case-splitting of bisection. However, our experiments show that bisection and $0 \notin F(\mathbf{X})$ test are faster than evaluating the Krawczyk operator; so we immediately return to bisection after each Krawczyk test. We incorporate Caprini and Madsen's epsilon inflation method [18] as described by Rump [13, Sec. 10.5] to ensure convergence when solutions are near the boundary of a hyper-rectangle. Our experiments showed the fastest convergence with a relatively small inflation of 0.1% of the hyper-rectangle.

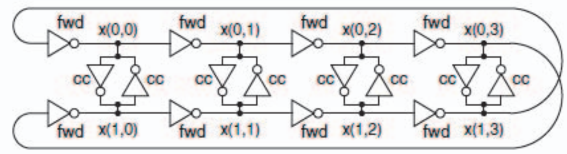

Fig. 1: Rambus Ring Oscillator

2) Overall Algorithm: Algorithm 1 depicts our main approach. The function SOLVER takes an argument f, and returns a list of all solutions for $f(x) = 0$ with $x \in \mathbf{H}_0$. When Krawczyk returns 'unknown', bisection divides the variable with the largest interval in $\mathbf{H}_K$ into two segments. SOLVER returns a list of hyper-rectangles, where each hyper-rectangle contains a unique solution.

Our current implementation is written in Python using the `numpy` package. The interval function and Jacobian computations were simple for the long-channel MOSFET model and the inverter macro model – we implemented the interval versions of these functions manually. The short-channel MOSFET model is more complicated, and we used the FADBAD++ [19] C++ package for automatic differentiation with the interval arithmetic package from MC++ [20]. We used the boost library to create Python bindings for these C++ functions. Our implementation is available as an open-source project at https://github.com/iaakhter/nonLinearSolver.

III. Results

We tried our solver on two sets of examples: finding all DC equilibrium points for the Rambus ring oscillator [1] and a Schmitt trigger circuit. We compare the results of the solver described in this paper with dReal [12] and Z3 [8] SMT solvers. All times are reported for executions on a MacBook Air with a 1.6GHz Intel i5 processor and 4 GB memory.

A. Rambus Ring Oscillator

Jones *et al.* [1] presented the Rambus Ring oscillator shown in Figure 1 as a challenge problem. It consists of an even number of stages connected in a Möbius topology. The Rambus researchers reported that examples of this design had failed intermittently in their test lab. When the forward inverters (labeled as fwd) are much stronger than the cross-coupled inverters (labeled as cc), the n stage oscillator functions as a ring for $2n$ inverters and has two stable equilibrium points. If the cross-coupled inverters are much larger that the forward ones, then the circuit has 3^n equilibrium points, of which 2^n are stable – we discovered that the number of DC equilibria in this case is 3^n by seeing the patterns in the solutions generated by our tool. Designers typically want to make the cross-coupled inverters as small as possible. However, if the cross-coupled inverters are only slightly too small, then failures will only occur for rare initial conditions. This rare but fatal failure motivates using formal approaches to verification.

We employed three models for the inverters. With the first, each inverter has a transfer function given by a $\tanh$ function. In particular, $I_{out} = g * \tanh(a * V_{in}) - V_{out}$ where $a < 0$ is the inverter gain (we use $a = -5$), and g models the output conductance. We also modeled an inverter as being composed of two CMOS transistors with the long-channel MOSFET model from [2] and the MVS short-channel MOSFET model from [21]. When bisecting on a voltage interval, many transistors will have no changes to the voltage intervals for their terminals. By memoizing the results of transistor model evaluations for both the long-channel and short-channel models, we obtain a significant improvement in performance over a direct implementation.

Table I shows the results for the Rambus oscillator example. The entries for g_{cc} are the ratio of the size of the cross-coupled inverters to that of the forward inverters. When $g_{cc} = 0.5$, an n-stage oscillator should have 3 equilibrium points. When $g_{cc} = 4.0$, then there should be 3^n equilibrium points.

For dReal, we attempted to find all solutions by finding them one-at-a-time and then adding constraints to exclude a small region around the solution. This is unsound because the exclusion region could accidentally enclose another solution, but dReal does not provide a way to enumerate all solutions. dReal requires a δ value that determines the amount of perturbation to the original system. With a δ value of 10^{-14}, for the $\tanh$ model, an exclusion rectangle of diameter $4 * 10^{-14}$ was sufficient (note that $\tanh$ values are in $(-1, +1)$). For the MOSFET models, we had to increase the size of the exclusion rectangle to a diameter of $2 * 10^{-11}$ for dReal to find all solutions. As seen in Table I, dReal is faster than our method for the three simplest examples. For larger problems, our method using interval-verification algorithms has much better performance and scalability.

We approximated the $\tanh$ function using a three segment, fifth-degree, piece-wise polynomial. Z3 found the solution at the origin for all six instances of the oscillator but failed to find any of the other solutions. The long-channel MOSFET model is piecewise polynomial, and therefore can be expressed in Z3's theory of non-linear arithmetic. However, Z3 timed out before finding any DC equilibria for any of the circuits with the long-channel model. We did not include the short-channel MOSFET model in Z3 trials because the model makes extensive use of transcendental functions.

B. Schmitt Trigger

An inverting Schmitt trigger [22] circuit exhibits hysteresis. There is a range of input voltages for which the output can settle high, or low, depending on its history. For such voltages, there is also a third equilibrium where the output settles at an intermediate value.

With the long-channel MOSFET model, the Jacobian matrix is singular for combinations of node voltages where either both n-channel devices or both p-channel devices are in cut-off. The Krawczyk method returns 'unknown' when confronted with such a singularity. In order to avoid this issue, we added a leakage term in the form of $1e - 8 * (V_d - V_s)$ to our current function where V_d and V_s are the drain and source voltages respectively. Using the long-channel model, dReal found the unique DC equilibrium for the $V_{in} = 0$ and $V_{in} = power$ cases in 0.19 and 0.17 seconds respectively. The $V_{in} = 0.9$ case took 4.14 seconds to find all three solutions. Our algorithm found the solutions for $V_{in} = 0$, 0.9, and $power$ in 0.27, 0.69, and 0.21 seconds respectively. For the short channel MOSFET model, our approach was able to find solutions for $V_{in} = 0$, 0.5, and $power$ in 0.51, 1.87 and 0.47 seconds respectively whereas dReal timed out.

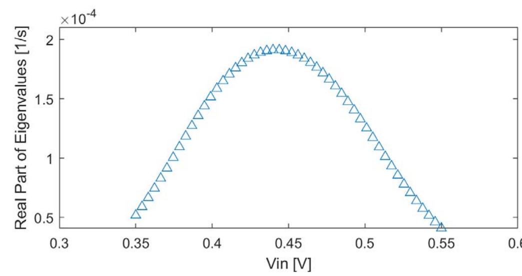

Fig. 2: Eigenvalues of the Schmitt Trigger's Unstable Equilibrium Point

To illustrate a further application of our DC operating point analysis, we analyze the effectiveness of the Schmitt trigger for resolving metastability. The Jacobian of f characterizes the equilibrium point: if all eigenvalues have negative real parts, the equilibrium point is stable; otherwise, the largest positive eigenvalue gives the rate of escape from the unstable equilibrium point. Figure 2 shows the positive eigenvalue of the Jacobian of f for the unstable equilibria. The effectiveness of the Schmitt trigger approaches zero as it gets closer to the ends of the hysteresis interval. Even at its peak, this eigenvalue is less than 2×10^{-4} for this example. For comparison, a cross-coupled inverter pair (like those in the Rambus oscillator) has a positive eigenvalue of 3×10^{-3}. The cross-coupled inverter pair is more than $15X$ as effective at resolving metastability. These experiments used the short-channel MOSFET model for transistor currents and a "normalized" capacitance of 1.

IV. Conclusions

We presented a new approach for finding all DC equilibrium points of circuits using interval arithmetic based verification algorithms. While interval arithmetic has been used by solvers

TABLE I: Rambus Ring Oscillator – comparison of our algorithm and dReal

Model	# Stages	g_{cc}	# solutions	#bisect – our method	Time – our method (s)	Time – dReal (s)
tanh	2	0.5	3	473	0.921	0.075
tanh	2	4.0	9	305	0.593	0.342
tanh	4	0.5	3	17189	52.763	0.480
tanh	4	4.0	81	12925	44.663	incomplete
tanh	6	0.5	3	570503	2458.652	time-out
tanh	6	4.0	729	486119	3496.199	prior time-out
LCMOSFET	2	0.5	3	545	5.722	4935.832
LCMOSFET	2	4.0	9	163	1.948	incomplete
LCMOSFET	4	0.5	3	24064	324.251	time-out
LCMOSFET	4	4.0	81	5022	73.316	prior time-out
LCMOSFET	6	0.5	3	986510	18400.544	prior time-out
LCMOSFET	6	4.0	729	139149	2951.956	prior time-out
SCMOSFET	2	0.5	3	1623	16.166	time-out
SCMOSFET	2	4.0	9	450	5.439	time-out
SCMOSFET	4	0.5	3	42128	556.849	prior time-out
SCMOSFET	4	4.0	81	17910	285.650	prior time-out
SCMOSFET	6	0.5	3	1056304	24353.846	prior time-out
SCMOSFET	6	4.0	729	513694	12567.220	prior time-out

Note: "time-out" means that the solver did not complete in 10 hours. 'incomplete" means that the solver found some, but not all solutions before reaching the 10 hour time out. "prior time-out" means that because solver timed out on a smaller version of the problem, we did not try the larger one.

such as iSAT [11] and dReal [12], the use of interval verification algorithms (e.g. the Krawczyk method) for analog and mixed signal verification is new. Our method finds *all* DC equilibrium points. We have demonstrated our approach using a simplified, tanh-based macro-model, a simple long-channel MOSFET model, and a state-of-the-art short-channel MOSFET model. Our tool achieves the first, fully automatic solution to the Rambus Ring Oscillator challenge problem [1] and shows much better scalability than the state-of-the-art solvers dReal [12] and Z3 [8].

REFERENCES

[1] K. D. Jones, J. Kim, and V. Konrad, "Some "real world" problems in the analog and mixed-signal domains," in *Proc. Workshop on Designing Correct Circuits*, Apr. 2008.

[2] M. H. Zaki, I. Mitchell, and M. R. Greenstreet, "Towards a formal analysis of DC equilibria of analog designs," Formal Verification for Analog Circuits (FAC'09), Grenoble, France, Jun. 2009.

[3] S. Tiwari, A. Gupta *et al.*, "First steps towards SAT-based formal analog verification," in *Proceedings of the 2010 International Conference on Computer Aided Design*, Nov. 2010.

[4] M. Miller and F. Brewer, "Formal verification of analog circuit parameters across variation utilizing SAT," in *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2013, pp. 1442–1447.

[5] R. Geoghegan, J. Lagarias, and R. Melville, "Threading homotopies and dc operating points of nonlinear circuits," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 159–178, 1998. [Online]. Available: https://doi.org/10.1137/S1052623496251586

[6] J. Lee and H.-D. Chiang, "Constructive homotopy methods for finding all or multiple DC operating points of nonlinear circuits and systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 1, pp. 35–50, Jan 2001.

[7] L. B. Goldgeisser and M. M. Green, "A method for automatically finding multiple operating points in nonlinear circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 4, pp. 776–784, April 2005.

[8] L. de Moura and N. Bjørner, "Z3: An efficient SMT solver," in *Tools and Algorithms for the Construction and Analysis of Systems*, C. R. Ramakrishnan and J. Rehof, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 337–340.

[9] D. Jovanović and L. de Moura, "Solving non-linear arithmetic," in *Automated Reasoning*, B. Gramlich, D. Miller, and U. Sattler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 339–354.

[10] G. E. Collins, "Quantifier elimination for real closed fields by cylindrical algebraic decomposition," in *Proceedings 2^{nd} GI Conference on Automata Theory and Formal Languages*. Berlin: Springer, 1975, pp. 134–183, LNCS 33. [Online]. Available: http://dx.doi.org/10.1007/3-540-07407-4_17

[11] M. Fränzle, C. Herde, T. Teige, S. Ratschan, and T. Schubert, "Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure," *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 1, pp. 209–236, 2007.

[12] S. Gao, S. Kong, and E. M. Clarke, "dReal: An SMT solver for nonlinear theories over the reals," in *Automated Deduction – CADE-24*, M. P. Bonacina, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 208–214.

[13] S. M. Rump, "Verification methods: Rigorous results using floating-point arithmetic," in *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, ser. ISSAC '10. New York, NY, USA: ACM, 2010, pp. 3–4. [Online]. Available: http://doi.acm.org/10.1145/1837934.1837937

[14] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009.

[15] M. R. Greenstreet and S. Yang, "Verifying start-up conditions for a ring oscillator," in *Proceedings of the 18^{th} Great Lakes Symposium on VLSI (GLSVLSI'08)*, May 2008, pp. 201–206. [Online]. Available: http://dx.doi.org/10.1145/1366110.1366160

[16] S. Steinhorst, M. Peter, and L. Hedrich, "State space exploration of analog circuits by visualized multi-parallel particle simulation," in *International Conference on Signal Processing Systems (ICSPS'09)*, May 2009, pp. 858–862.

[17] R. Moore, "A test for existence of solutions to nonlinear systems," *SIAM Journal on Numerical Analysis*, vol. 14, no. 4, pp. 611–615, 1977. [Online]. Available: https://doi.org/10.1137/0714040

[18] O. Caprani and K. Madsen, "Iterative methods for interval inclusion of fixed points," *BIT Numerical Mathematics*, vol. 18, no. 1, pp. 42–51, Mar 1978. [Online]. Available: https://doi.org/10.1007/BF01947742

[19] C. Bendtsen and O. Stauning, "FADBAD, a flexible C++ package for automatic differentiation - using the forward and backward methods," Department of Mathematical Modeling, Technical University of Denmark, Tech. Rep. IMM-REP-1996-17, 09 1996.

[20] "MC++," https://omega-icl.github.io/mcpp/index.html, accessed: 2018-09-310.

[21] M. S. Lundstrom and D. A. Antoniadis, "Compact models and the physics of nanoscale FETs," *IEEE Transactions on Electron Devices*, vol. 61, no. 2, pp. 225–233, Feb 2014.

[22] L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits*. Addison-Wesley, 1985.