Available online at www.sciencedirect.com
SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science 153 (2006) 37–52

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Time Constrained Verification of Analog Circuits using Model-Checking Algorithms

Darius Grabowski[1]

Institute of Microelectronic Systems
University of Hannover, Germany

Daniel Platte[2]

Infineon Technologies AG
Munich, Germany

Lars Hedrich[3]

Institute for Computer Science
University of Frankfurt/Main, Germany

Erich Barke[4]

Institute of Microelectronic Systems
University of Hannover, Germany

Abstract

In this contribution we present algorithms for model checking of analog circuits enabling the specification of time constraints. Furthermore, a methodology for defining time-based specifications is introduced. An already known method for model checking of integrated analog circuits has been extended to take into account time constraints. The method will be presented using three industrial circuits. The results of model checking will be compared to verification by simulation.

Keywords: Model Checking, Analog Circuits, CTL, Time Constraints

[1] Email:darius.grabowski@ims.uni-hannover.de

[2] Email:daniel.platte@infineon.com

[3] Email:hedrich@informatik.uni-frankfurt.de

[4] Email:barke@ims.uni-hannover.de

doi:10.1016/j.entcs.2006.01.026

1 Introduction

Formal verification methods are widely used in design validation of digital circuits. In contrast to the digital domain formal verification of analog circuits is still under research. Some approaches in the area of hybrid systems are well known: Linear or piecewise linear hybrid systems using ordinary differential equations (ODE) can be analyzed in terms of reachability analysis [9]. Recent approaches are also able to deal with nonlinear equation systems [5]. Analog circuits described on transistor level have to be described with a strongly nonlinear differential-algebraic equation system. Methods for these circuit types can be divided into invariant set computations/reachability analyses [6] and model checking approaches [7,8] enabling CTL-like property descriptions.

So far, the latter methods were not suitable for verification of time constraints. As a major part of circuit specifications contains time criteria, an extension of analog model checking towards consideration of time constraints is essential for future use. There exits approaches to define and check real time properties for example for timed automata: TCTL ([2]) and more dedicated to delays of digital circuits: WQCTL ([4]). The extensions presented in this paper are based on algorithms for analog circuits taking time behavior into account during discretization of the state space and checking specifications in an extended computation tree logic (CTL-AT).

The first part of this publication focuses on algorithms for analog model checking and a methodology for the verification of time constraints. The functionality of the new algorithms will be demonstrated in the second part of the paper using three analog circuit blocks.

2 Model Checking

The presented method aims at the analysis of nonlinear dynamic analog circuits. Based on an extended modified nodal analysis [10] a system of nonlinear differential-algebraic equations (DAE system) is set up for the circuit. The state variables of the energy storing elements (voltages at capacitances, currents through inductors) are used as independent variables. These variables and the input variables span an extended state space.

In [7,8] an analog model checking method was presented to compare a specification represented by a CTL expression to the circuit behavior. To make this possible the continuous n-dimensional state space has to be mapped to a finite discrete transition system.

Therefore, the state space is bounded and automatically divided into a finite number of n-dimensional hyperboxes. Each of these hyperboxes represents a homogeneous part of the state space and is treated as a discrete state of the simplified system. Figure 1a) shows the discrete state space of a damped resonant circuit.

The finite set of discrete hyperboxes fully covers the limited state space. To obtain successor relations between the discrete states a constant number of randomly generated points is placed within each hyperbox of the state space. Each of these points represents a combination of values for the input and state variables by its coordinates. By numerically solving the nonlinear DAE-system of the circuit solution vectors can be obtained for those points. A vector can be transformed into a discrete state-transition by using an over-estimating estimation. In Figure 1a) the solution vectors are shown as arrows. Additionally, within the enlarged box the resulting state transitions of the dark-grey hyperbox are visualized by the midpoint-to-midpoint connection.

Different criteria take care of the resulting error during discretization and try to automatically minimize the error by chosing a suitable subdivision of the state space. The specification of a circuit formulated in CTL-AT can then be checked by using digital model checking algorithms.

2.1 Time Constrained Computation Tree Logic

The analysis of timing behavior is based on a computation tree logic (CTL) as already used in digital circuit analysis [2,3,4]. With the aid of CTL it is possible to define specifications for finite state machines. The compliance of these CTL expressions with the system's behavior can be proven automatically by using model checking algorithms. CTL uses special operations that are built of a so called path quantifier (A, E) and a temporal operator (F - finally, G - generally, U - until). The path quantifier E defines that a temporal operation has to be fulfilled by at least one path within the finite state machine. The A quantifier determines that the CTL condition has to be fulfilled on all possible paths.

The temporal operators accept one (F, G) or two (U) sets of states as arguments. All CTL operations result in a set of states that meet the specified condition. Sets of states are marked with capital greek letters.

Using temporal operations the dynamic specification of state machines can be described. E.g. $EF(\Phi)$ results in a set of starting states of paths that finally reach the set Φ. All states of a set Φ that do not have a path leaving the set can be found by using the CTL formula $AG(\Phi)$.

However, this basic form of CTL is neither suitable for analyzing analog circuits nor for specifying timing behavior. Hence, the logic was extended by analog operators ($>$,$<$) that offer the possibility to define sets of states within the continuous state space of analog systems. To specify dynamic sys-

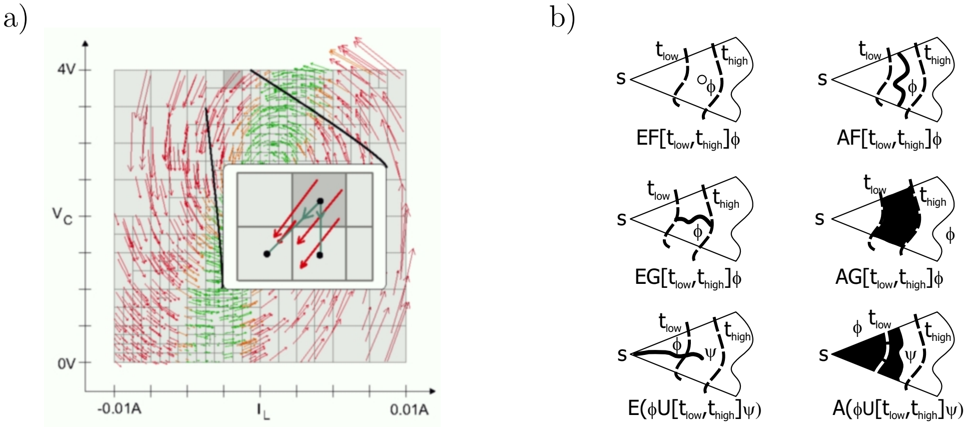

Fig. 1. a) Discrete State Space and b) CTL-AT Reference

tem behavior with a CTL syntax (CTL-AT) it was necessary to introduce time-constrained temporal operators that additionally constrain the scope of the operations. Further extensions concern the direction of state transitions. Therefore, the time inversion has been introduced, described by the $^{-1}$ operator.

Figure 1b) shows CTL-AT operations for analog circuits. S represents a starting state of a path within an automata that fulfills the CTL-AT condition specified below each drawing. The cone-like form depicts the branching of paths starting in S.

The usage of the time-constrained temporal operators F and U causes a reduced result set in comparison to the unconstrained operations. The resulting set of states only contains those states that reach the argumental set of states on paths whose temporal length is within the time interval given with the operation. In contrast, the time constraint of the G operation increases the result set of the operation as the previous meaning of G is attenuated from "generally" to "during the time interval".

2.2 Algorithms for Time Constrained Analyses

The identification of delay times in the discretized state space is necessary when taking into account time constraints. For modelling delay times in finite automata there are two promising approaches – timed automata [1] and delayed state transitions. The presented algorithms use delayed transitions to model the time within the determined transition system as this is algorithmically easier to handle. Furthermore, they give a good approximation of analog circuits' behavior. The calculation of the delay times takes place during the discretization of the state space. By evaluating the time steps of the equation solver during the identification of successing states a ratio of delay time and length of the solution vector is obtained. Simultaneously, the delay time for the resulting transition is corrected proportionally to the midpoint distance of both states. If more than one solution vector leads to the same transition the arithmetic mean of the corrected delay times is associated with the transition.

Determining the result sets of time-constrained CTL operations requires the processing of temporal path lengths within the transition system. The intention is e.g. to find all states that reach a set of states on at least one path within a given time interval (time-constrained EF operation). The end of those paths is determined by the argument of the CTL-operation and may consist of a set of discrete hyperboxes of the state space.

The calculation of CTL-operations can be reduced to two basic operations EU and EG by using CTL theorems [8]. For processing the result sets of the EF and EU operation the former algorithm was extended in a way that it assigns a set of discrete delay times to each state. This set contains all temporal lengths of paths starting in this state and ending within the target set. The total delay time of a path is calculated by summing up all delay times of transitions on its path. After finishing the algorithm the resulting set of states is reduced by states that do not have a delay time within the given time interval (usage of E-quantifier) or by states having one delay time outside of the time interval (usage of A quantifier).

Exemplarily, we will discuss the algorithm for calculating the solution of a time constrained EU operation. Afterwards, a simple example (Figure 2) is shown to clarify the functionality.

The $E(\Phi U[t_{low}, t_{high}]\Psi)$ operation accepts two sets of states as arguments. The resulting set is intended to contain all states that reach Ψ on at least one path within Φ within the given time interval $[t_{low}, t_{high}]$. The algorithm starts with the set Ψ which is copied into an intermediate set of states Ω. Next, the previously introduced sets of delay times are initialised with zero for the states Ψ.

Ω converges towards the resulting set by repeatedly adding previous states that fulfill several conditions: The previous state has to reach Ω by one of its transitions, it has to be part of Φ, it can not be added multiple times to Ω and finally it has to satisfy the upper time constraint t_{high}. Adding a state to the set Ω results in a propagation of its set of delay times to the previous state after increasing all contained times by the transition time. The algorithm terminates as soon as an invariant set Ω is found. The resulting set is obtained by finally reducing Ω by all states that do not contain a delay time greater than t_{low}.

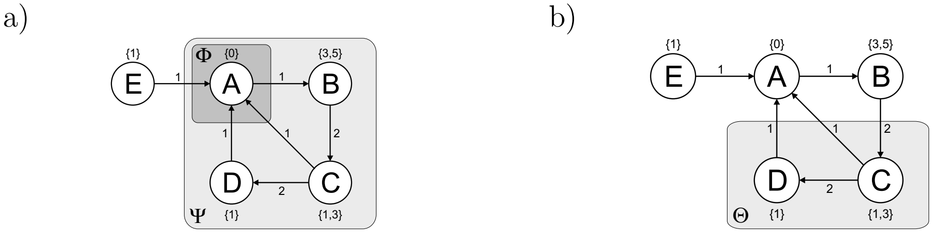

Fig. 2. a) Transition System and b) Resulting Set of Equation (3)

$$\Psi = \{A, B, C, D\} \tag{1}$$

$$\Phi = \{A\} \tag{2}$$

$$\Theta = E(\Psi\ U[1,2]\ \Phi) = \{C, D\} \tag{3}$$

Figure 2a) depicts an exemplary transition system to illustrate the algorithm to determine the result of a EU operation. Beside the transitions, the according delay times are quoted. Each state node contains the set of possible delay times when transitioning to state A. Equations (1) to (3) define two sets of states and a CTL-AT operation. The result set is displayed in Figure 2b). It contains only states that reach Ψ on at least one path within a delay interval of 1 to 2 time units.

3 Methodology for Verification of Time-Based Specifications

In this section a methodology to verify time-based specifications like signal edges and oscillations by using model checking will be presented. The advantage of using model checking instead of circuit simulation is that by using model checking algorithms and CTL-AT formulas the compliance of a system to its specification can be proven.

3.1 Verification of Time Constraints for Signal Edges

Figure 3a) shows the transient plot of a signal edge. The time-constrained F-operation can be applied to prove a rise time using model checking algorithms. For that purpose two areas Φ_{high} and Φ_{low} of the state space have to be defined. Φ_{high} and Φ_{low} contain the states at the upper and lower side of the threshold values. Equations (4) and (5) are used to extract the sets of states by using analog operators ($>$, $<$) and lower as well as upper threshold values. In Figure 3b) the signal edge is plotted in the state space with a voltage U over another state variable X which is not of interest for this evaluation.

$$\Phi_{high} = U > U_{high} \tag{4}$$

$$\Phi_{low} = U < U_{low} \tag{5}$$

A specified rise time T_{LH} can be verified by Equation (6) and the sets of states Φ_{high} and Φ_{low}. After applying the model checking algorithm the resulting set of states Ω_{LH} contains all states which reach the area Φ_{high} starting from Φ_{low} on all paths faster than constrained by T_{LH}. If the result set is empty, the specification is not fulfilled by the circuit.

$$\Omega_{LH} = AF[0, T_{LH}](\Phi_{high}) \wedge \Phi_{low} \tag{6}$$

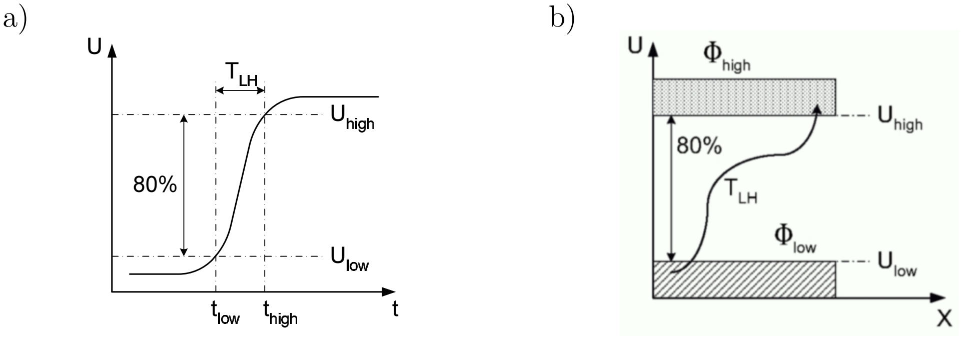

Fig. 3. Positive edge: a) Transient Plot and b) Representation in the State Space

Based on the result for the rise time proof of a specified slew rate can also be given using Equation (7). For falling edges the methodology can also be applied in a slightly modified way by simply interchanging the sets Φ_{high} and Φ_{low}

$$SR = \frac{U_{high} - U_{low}}{T_{LH}} \tag{7}$$

3.2 Oscillation Analysis

The basis for analyses of oscillating behavior is the extraction of the oscillation area Θ. Therefore, the state space is split in two sections Φ_{low} and Φ_{high}. In the next step we determine all states which finally lead to Φ_{low} and Φ_{high} respectively on all paths by applying the AF formula. Θ is part of the superposition of the resulting sets. Applying the AG formula all states leaving the selected area on at least one path are excluded. Finally, we use the inverted EG-formula to exclude all states settling up into the oscillation. The resulting formula is presented in Equation (8).

$$\Theta = EG^{-1}(AG(AF(\Phi_{high})) \wedge AG(AF(\Phi_{low}))) \tag{8}$$

Within Θ two sections Φ_0 and Φ_1 can be chosen splitting the oscillation area into a positive and a negative half-space. In Figure 4 Θ, Φ_0 und Φ_1 are schematically displayed in a two-dimensional state space spanned by the state variables X_1 and X_2.

$$\Omega_{01} = E(\Theta\ U[0, T_{high}]\ \Phi_1) \wedge \Phi_0 \tag{9}$$

$$\Omega_{10} = E(\Theta\ U[0, T_{low}]\ \Phi_0) \wedge \Phi_1 \tag{10}$$

By using model checking combined with the time-constrained EU operation shown in Equations (9) and (10), the delay times T_{high} and T_{low} for the positive and the negative half space can be proven. The procedure is equal to the analysis of a signal edge; the only difference is the additional restriction of the paths to the oscillation area by using the EU operation instead of AF.

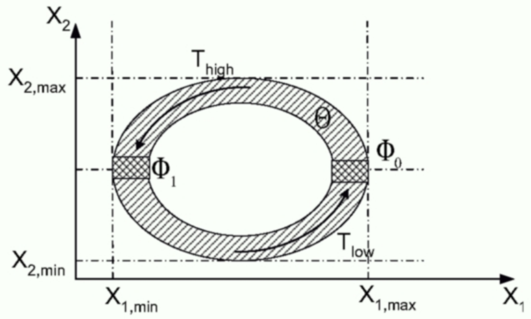

Fig. 4. Oscillation in the State Space

$$f = \frac{1}{T} = \frac{1}{T_{low} + T_{high}} \tag{11}$$

$$DC = \frac{T_{high}}{T_{low} + T_{high}} \tag{12}$$

The compliance of the circuit to a specified oscillation frequency and duty cycle can be derived from Equations (11) and (12).

4 Applications

The proposed CTL-AT approach was implemented as an extension of an existing model checking prototype tool called *amcheck*, developed at the IMS, University of Hannover. The presented results were processed on a SUN server with a sparcv9 processor. We use three circuits containing two respectively three state variables.

At this point we should mention that processing time is a crucial problem, because in principle it rises exponentially with the number of system states. The main part of the runtime concerns the discretization of the state space requiring the computation of transition vectors for each hyperbox. In the last paragraph we give some information about runtime of the algorithm.

4.1 Schmitt Trigger

The first example is an inverting Schmitt trigger shown in Figure 5a). The extended state space of the circuit is spanned by the input voltage v_{in} and the voltage v_{out} across the capacitance C_{load}. The operational amplifier is modelled in a static manner considering a voltage and current limitation at the output node.

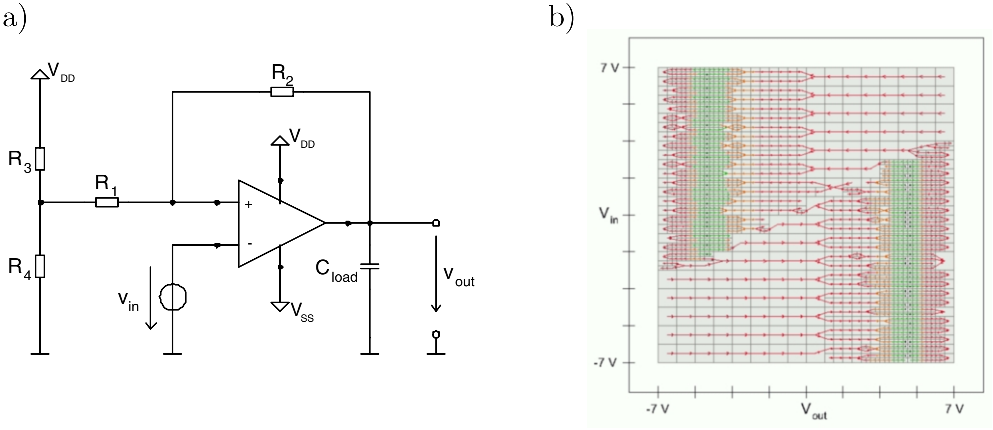

Fig. 5. Schmitt Trigger: a) Schematic and b) Discretized State Space

Initially, we have to define the limits of the state space. Therefore, we evaluate the threshold voltages given by the resistances $R_{1..4}$. Further restrictions result from the supply voltages of the operational amplifier given by $V_{DD} = 5\,V$ und $V_{SS} = -5\,V$. Hence, we set the limits to $\Gamma := [-7\,V \,..\, 7\,V] \times [-7\,V \,..\, 7\,V]$. The discretized state space is shown in Figure 5b). The state space contains two quasi-stable areas represented by small hyperboxes at $v_{out} = \pm 5\,V$.

Using CTL-AT we determine some relevant quantities of the Schmitt trigger like fall time, rise time and slew rate. Initially, we define the sets Φ_{low} und Φ_{high} representing starting and ending values for the voltage edges by Equations (13) and (14). The set limits contain a tolerance of ±10 %.

$$\Phi_{low} = v_{out} \ < \ -4.5\,V \tag{13}$$

$$\Phi_{high} = v_{out} \ > \ 4.5\,V \tag{14}$$

The first analysis concerns the rise time of the circuit. Applying the time-constrained AF formula in Equation (15) we get all states which contain paths with the given time interval ending in Φ_{high}. The corresponding analysis of the fall time considers Φ_{low} as the source set of the AF formula (see Equation (16)). Figures 6a) and b) show the sets fulfilling the given criteria.

$$\Omega_{LH} = AF[0,\ 1.45 \cdot 10^{-6}](\Phi_{high}) \tag{15}$$

$$\Omega_{HL} = AF[0,\ 1.65 \cdot 10^{-6}](\Phi_{low}) \tag{16}$$

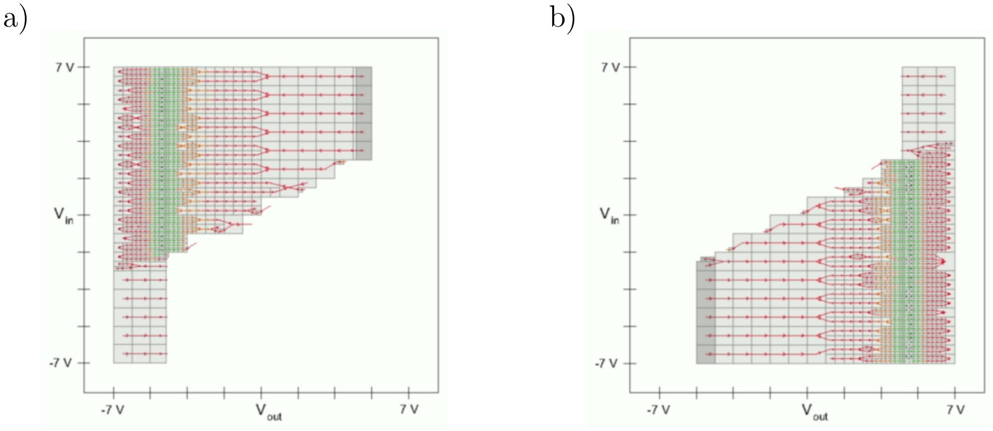

Fig. 6. Analysis of a) Fall Time and b) Rise Time

Finally, we have to check if the resulting sets satisfying Equations (15) and (16) intersect the respective sets containing the start values. Using the $\wedge$-operator in Equations (17– 18) we evaluate all states performing a rising edge with a given rise time of $1.45 \cdot 10^{-6}\,s$ and on the other hand all states performing a falling edge in $1.65 \cdot 10^{-6}\,s$. The time constraints were determined in an iterative manner, however in practise, they are given by the circuit's specification.

$$\Theta_{LH} = \Omega_{LH} \wedge \Phi_{low} \tag{17}$$

$$\Theta_{HL} = \Omega_{HL} \wedge \Phi_{high} \tag{18}$$

Figures 6a) and b) show the resulting sets satisfying Equations (15) and (16). The dark areas represent the sets Θ_{LH} and Θ_{HL} fulfilling Equations (17) and (18).

	Model Checking	Simulation	Difference
T_{LH}	$< 1.45\,\mu s$	$1.4\,\mu s$	$3.5\,\%$
SR_{LH}	$6.21\,\frac{V}{\mu s}$	$6.43\,\frac{V}{\mu s}$	$3.4\,\%$
T_{HL}	$< 1.65\,\mu s$	$1.6\,\mu s$	$3.1\,\%$
SR_{HL}	$5.46\,\frac{V}{\mu s}$	$5.63\,\frac{V}{\mu s}$	$3,0\,\%$

Table 1
Comparison of Simulation vs. Model Checking

Another quantity characterizing the behavior of a Schmitt trigger is the slew rate. It is defined as a ratio between voltage difference and time. The results are shown in Table 1 containing the relations between both verification methods. The rise time measured by simulation is $1.4\,\mu s$ whereas the fall time is about $1.6\,\mu s$. The values confirm the results achieved by model checking.

4.2 Operational Amplifier

The second application presents the slew rate verification of a transistor level circuit. We use an industrial operational amplifier shown in Figure 7. The schematic contains the circuit itself (boxed part) and additionally a testbench circuit controlling input and output behavior. The input voltage is a superposition of a DC source V_{DC} and the AC voltage v_{In} at the positive input of the operational amplifier. In order to measure the slew rate and the overshoot we connect the output to the negative input. The power supply voltage V_{DD} is set to $1.5\,V$. The state space of the circuit is spanned by the input voltage v_{In} as well as by the voltages v_{out} and v_{comp} across the capacitances C_{load} and C_{comp}.

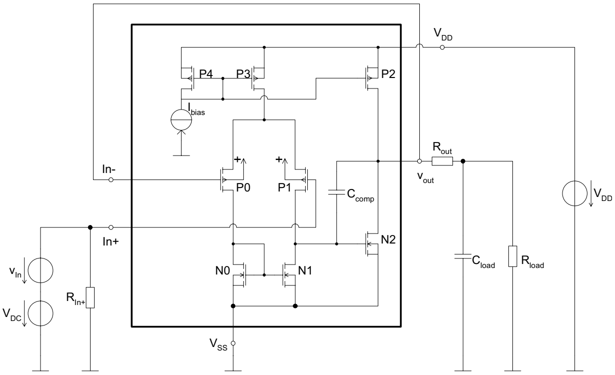

Fig. 7. Operational Amplifier

The operational amplifier consists of two stages. The first one is a differential pair (*P0*, *P1*) with an active load (*N0*, *N1*) controlled by the current source I_{bias}. The second stage is a driver stage built up by *P2* and *N2*. To improve the transient performance of the circuit the so called "Miller" capacitance C_{comp} is added.

The first test checks the slew rate of the operational amplifier. It should be at least $3.6\,\frac{V}{\mu s}$. We use the assumption, that to perform such a slew rate, a rising step in the output voltage has to be fulfilled in a predefined time. In this case we use an input step from $0.3\,V$ to $0.5\,V$. Hence, the maximum time spent for this step has to be $5.5 \cdot 10^{-8}\,s$. The formula is constructed for a 3-dimensional extended state space.

$$\Phi_{U_{low}} = EG^{-1}|_{v_{In}=const.}(v_{In} > 0.29\,V \wedge v_{In} < 0.31\,V) \tag{19}$$

$$\Phi_{U_{high}} = EG^{-1}|_{v_{In}=const.}(v_{In} > 0.49\,V \wedge v_{In} < 0.51\,V) \tag{20}$$

$$\Psi_{SR} \models EF[0,\,1 \cdot 10^{-20}](AF[0,\,5.5 \cdot 10^{-8}]|_{v_{In}=const.}(\Phi_{U_{high}} \wedge \Phi_{U_{low}})) \tag{21}$$

$v_{In} = const.$ means that the transitions due to input change are disabled. In contrast, the expression $EF[0, 1 \cdot 10^{-20}]$ is used to get an input jump to the appropriate value without any change of other state variables. In the very short time of $1 \cdot 10^{-20}\,s$ the change of state variables is very small. Expressions (19–21) are generalized expressions of the manual steps in Equations (13–17). A run with this formula results in a non empty set of three boxes showing that the slew rate specification is fulfilled.

As an additional test the overshoot specification is given:

$$\Gamma_{In} = (v_{In} > -0.2\,V \wedge v_{In} < 0.2\,V) \tag{22}$$

$$\Gamma_{Out} = (v_{Out} > 0.55\,V \wedge v_{Out} < 0.95\,V) \tag{23}$$

$$\Theta_{DC} = EG^{-1}|_{v_{In}=const.}(\Gamma_{In}) \tag{24}$$

$$\Omega_{OS} \models E(\Gamma_{In} U^{-1} \Theta_{DC}) \wedge \Gamma_{Out} = \varnothing \tag{25}$$

The regions are defined for the input and output voltage range for an offset voltage $V_{DC} = 0.75\,V$. The resulting set of the overshoot expression is not empty. Hence, the specification does not hold. The reason for the overshoot is the resistor R_{out} which in combination with the output capacitance leads to a delayed feedback and finally results in an overshoot.

4.3 Voltage Controlled Oscillator

The third example is a voltage controlled oscillator (VCO) shown in Figure 8. The corresponding state space is spanned by the voltages at the capacitances C_1 and C_2. The input is modelled by an ideal voltage controlled current which is mirrored by *TN1, TP1, TN2, TP2, TN5* charging the capacitance C_2.

Assuming the output voltage v_{C_1} to be at V_{DD}, C_2 is charged up linearly by the input current through the switch *TN3, TP3* controlled by the inverter *(TN4, TP4)*. As v_{C_2} exceeds the positive threshold voltage of the Schmitt trigger, determined by the resistors R_1 and R_2, v_{C_1} changes to V_{SS}. Consequently, C_2 is discharged until the initial status is reached leading to an oscillation.

Based on CTL-AT we check two properties of the VCO. The first property is the oscillating frequency for some distinct input voltages. Using the results, we check the frequency linearity as a function of the input voltage. Considering the supply voltages and the values of the resistances R_1 and R_2, the state space is defined by $\Gamma := [-3.1\,V \,..\, 3.1\,V] \times [-0.08\,V \,..\, 0.08\,V]$. Figure 9a) shows the discretized state space of the VCO. It contains some horizontal paths representing fast transitions between the maximum and minimum of the output voltage. The oscillation time is dominated by the slow (vertical) transitions at $v_{C_1} = \pm 2.5\,V$.

Using Equation (8) presented in Chapter 3.2 we compute all states of the oscillation area. Therefore, we split the given state space into two parts using

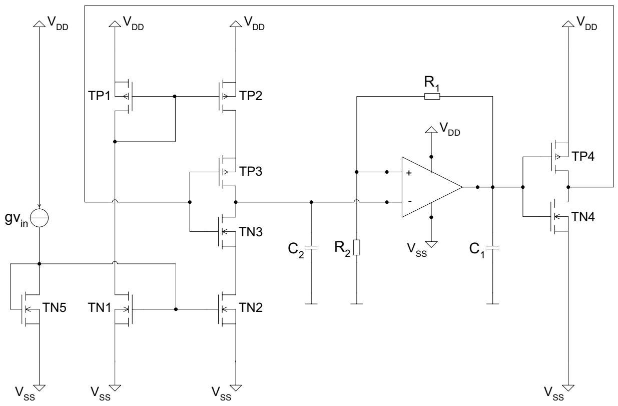

Fig. 8. Voltage Controlled Oscillator

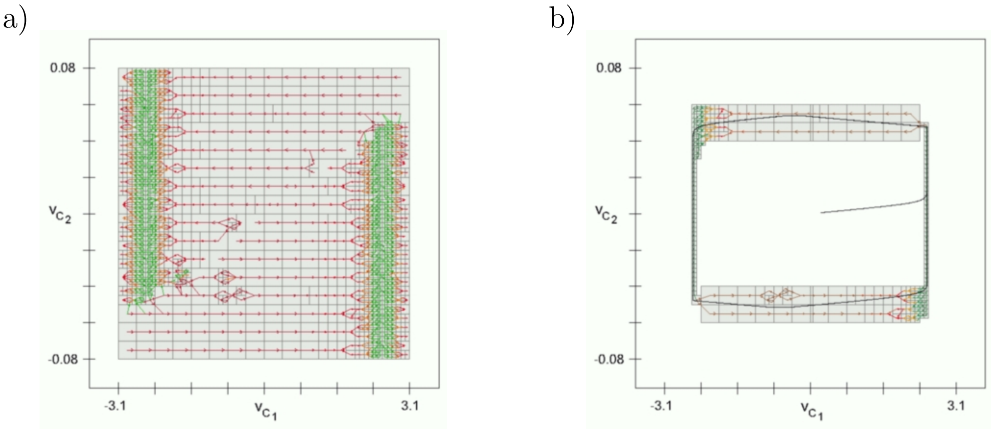

Fig. 9. VCO: a) Discretized State Space and b) Oscillation Area

knowledge about the circuit oscillating around $v_{C_1} = 0V$:

$$\Phi_{low} = v_{C_1} < 0\,V \tag{26}$$

$$\Phi_{high} = v_{C_1} > 0\,V \tag{27}$$

Figure 9b) shows the set of states representing the oscillation area at $v_{in} = 1.7\,V$. Furthermore, it contains the trajectory (black curve) calculated by a transient simulation. As it can be seen the oscillation area intersects the trajectory.

As mentioned in Chapter 3.2, the methodology to determine the oscillation time requires a partitioning of the oscillation area. Hence, we define two sets with a small range. We choose a section with fast transitions to minimize the discretization error. The two sets containing start areas for oscillation paths are defined by Equations (28) and (29).

$$\Phi_0 = \Theta \wedge (v_{C_1} > 0) \wedge (v_{C_1} < 0.05) \wedge (v_{C_2} > 0) \tag{28}$$
$$\Phi_1 = \Theta \wedge (v_{C_1} > 0) \wedge (v_{C_1} < 0.05) \wedge (v_{C_2} < 0) \tag{29}$$

In contrast to the approach in Chapter 4.1, where we used the EF operation to obtain path lengths, we now make use of the time constrained EU operation in an iterative manner to get the appropriate time delays. Therefore, we need an upper time bound to guarantee the resulting set of Equation (30) overlapping Ω_{10} and vice versa.

$$\Omega_{01} = E(\Theta\; U[0, 0.64]\; \Phi_1) \tag{30}$$
$$\Omega_{10} = E(\Theta\; U[0, 0.69]\; \Phi_0) \tag{31}$$

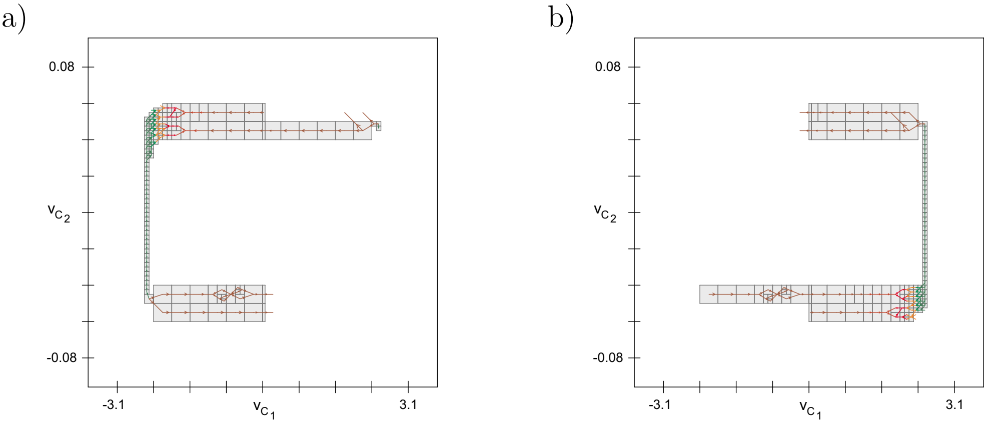

Fig. 10. VCO: a) Left Oscillation Path and b) Right Oscillation Path

Figures 10a) and b) show the results of Equations (30) and (31). In this example using $v_{in} = 1.7\,V$ we get two different time delays given by $0.64\,s$ resp. $0.69\,s$. We evaluate an oscillation time of about $1.33\,s$. It is important to note that model checking gives an overestimation of time delay. However, the inner path of the oscillation area permits faster oscillations than the outer one. Recall that this effect depends on the accuracy of the discretization.

We now focus on the linearity of input voltage and oscillation frequency. In this example we estimate the oscillation frequencies applying three different input voltages given by $v_{in} = [1.7\,V, 2.2\,V, 2.7\,V]$. In order to measure the linearity we define the linearity error by using the differencial quotient of frequency and input voltage: $\Delta f / \Delta v_{in}$.

In Table 2 we present the model checking results in comparison to transient simulation. The relative error of about $5\,\%$ concerns frequency differences and amounts. The relative linearity error is measured as $10.0\,\%$ (transient simulation) and $12.5\,\%$ (model checking).

Finally, we give some information about the runtime of the presented examples. We take into account the number of circuit equations. The actual

	Model Checking		Simulation		
v_{in}	period	frequency	period	frequency	rel. error
$1.7\,V$	$1.33\,s$	$0.75\,Hz$	$1.27\,s$	$0.79\,Hz$	$5.06\,\%$
$2.2\,V$	$1.01\,s$	$0.99\,Hz$	$1.01\,s$	$0.99\,Hz$	$0.00\,\%$
$2.7\,V$	$0.83\,s$	$1.20\,Hz$	$0.85\,s$	$1.17\,Hz$	$2.56\,\%$

Table 2
Comparison of Simulation vs. Model Checking

intersection depth of each box depends on the maximum predefined intersection depth and the nonlinearity of the equation system. Table 3 shows the runtimes. All circuits' state spaces are discretized using five testpoints for each hyperbox. Generally, the runtime of the algorithm is dominated by the discretization of the state space, whereas the evaluation of the CTL operations has a negligible duration. Comparing the Schmitt trigger to the VCO shows that the calculation of the transitions strongly depends on the number of equations due to the numerical solver. As mentioned before the runtime rises exponentially with the number of state variables which can be seen in the operational amplifier example. This circuit is described by 51 equations which is similar to the VCO. The state space contains three state variables producing 58962 hyperboxes which is about 60 times more than the VCO circuit with two state variables.

Circuit	Schmitt Trigger	Operational Amplifier	VCO
Equations	11	51	55
Max. Intersection Depth	12	16	14
Hyperboxes	1229	58962	1267
Discretization Time	20 s	2873 s	414 s

Table 3
Model Checking: Runtime

5 Conclusion

In recent years, model checking was successfully applied to verify digital circuits making a nearly fully automated verification possible. By the discretization of the continuous state space of analog systems, model checking algorithms can also be applied to analog circuits.

Generally, the specification of analog systems is based on time constraints. Therefore, the discretization algorithm has been extended considering delay times which are later on modelled as delayed state transitions. To apply time constrained computation tree logic (CTL-AT) new algorithms have been developed.

Furthermore, a methodology to perform time analyses as used in verification of signal edges and oscillations has been introduced. Three circuits were presented as examples to show the practical results of the method. Comparing the results of the model checking algorithm to simulations shows that model checking is a good method to automatically verify analog circuits.

References

[1] Alur, R., *Timed automata*, CAV '99: International Conference on Computer-Aided Verification, LNCS (1999), pp. 8–22.

[2] Alur, R., C. Courcoubetis and D. Dill, *Model-Checking for Real-Time systems*, LICS '90: Proceedings of the Annual IEEE Symposium on Logic in Computer Science **5** (1990), pp. 414–425.

[3] Alur, R., C. Courcoubetis and D. Dill, *Model-checking in dense real-time*, Information and Computation (1993), pp. 2–34.

[4] Chatterjee, K., P. Dasgupta and P. Chakrabarti, *A Branching Time Temporal Framework for Quantitative Reasoning*, Journal of Automated Reasoning, **30** (2003), pp. 205–232.

[5] Dang, T., A. Douze and O. Maler, *Verification of analog and mixed-signal circuits using hybrid system techniques*, FMCAD 04: Formal Methods in Computer-Aided Design (2004).

[6] Dellnitz, M., G. Froyland and O. Junge, *The algorithms behind gaio - set oriented numerical methods for dynamical systems*, Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems (eds. B. Fiedler), Springer (2001), pp. 145–174.

[7] Hartong, W., L. Hedrich and E. Barke, *Model checking algorithms for analog verification*, DAC '02: Design Automation Conference (2002), pp. 542–547.

[8] Hartong, W., L. Hedrich and E. Barke, *On discrete modeling and model checking for nonlinear analog systems*, CAV '02: International Conference on Computer-Aided Verification, LNCS **2404** (2002), pp. 401–413.

[9] Henzinger, T. and P.-H. Ho, *Algorithmic analysis of nonlinear hybrid systems*, CAV '95: International Conference on Computer-Aided Verification, LNCS **939** (1995), pp. 225–238.

[10] Ho, C., A. Ruehli and P. Brennan, *The modified nodal approach to network analysis*, IEEE Transactions on Circuits and Systems **22** (1975), pp. 504–509.