Three-Valued Automated Reasoning on Analog Properties

R. Gentilini
Kaiserslautern University,
Computer Science Dep.
Gottlieb-Daimler-Straße
Kaiserslautern - Germany
gentilin@
informatik.uni-kl.de

K. Schneider
Kaiserslautern University,
Computer Science Dep.
Gottlieb-Daimler-Straße
Kaiserslautern - Germany
Klaus.Schneider@
informatik.uni-kl.de

A. Dreyer
Fraunhofer ITWM
Adaptive Systems
Fraunhofer-Platz 1
Kaiserslautern - Germany
alexander.dreyer@
itwm.fraunhofer.de

ABSTRACT

We deal with the problem of designing suitable languages for the *modeling* and the *automatic verification* of properties over analog circuits. To this purpose, we suitably enrich classical temporal logics with basic formulæ allowing to model arbitrary functions relating analog variables. We show how to automatically check the resulting CTL_f formulæ on analog circuits. In particular, we rely on interval arithmetic methods and we extend to the analog context a number of techniques for the abstraction and the verification of digital systems, based on three-valued temporal logics.

Categories and Subject Descriptors

J.6 [**Computer-Aided Enjeneering**]: Analog Properties Formal Specification, Computer Aided Design (CAD)

General Terms

Verification

Keywords

(Multi-Valued) Temporal Logics & Model Checking, Interval Arithmetic, Analog Circuits

1. MOTIVATION

Analog circuit design is of great importance in nowadays microelectronics [4, 6]. In order to avoid a large number of defective prototypes, it is necessary to check significant properties of an analog circuit before manufacturing it. Traditionally, the above task relies largely on the cooperation of expert knowledge, manual calculation, and numerical circuit simulation [4]. A quite recent and active research field, named as *symbolic analysis* of analog circuits, aims at (1) endowing the verification process of some degree of *automation* (2) coping with *variability* of parameters and input signals [10, 6, 11]. The above objectives are fullfilled by providing computer algebra algorithms to simplify the mathematical model of an analog circuit. Typically, such procedures work by detecting and deleting subformulæ whose influence on the evolution of the system is insignificant. Despite the promising developments, *system analysis* is still far from being a *completely automatic* method for the verification of analog circuits.

On the contrary, in the context of digital circuits, *model checking* allows to verify (1) entirely automatically (2) along all possible executions of the systems, properties that are *formally* stated in a temporal logic. A natural way to apply model checking to the context of analog circuits verification is the extraction of some finite digital abstraction from the corresponding analog model. Such an approach has been adopted in [5], where the authors qualitatively proved its effectiveness by applying it to some concrete (and well-understood) examples. As part of their work, the authors proposed an enrichment of classical temporal logic languages, recognizing the inadequacy of the latter to model analog properties. However, their extension is still not satisfactory: As an example, simple properties comparing the values of analog variables along their evolution can still not be dealt with in [5]. Indeed, even if a constraint of the kind $x < y$ could be expressed in [5], a major problem to its verification would be the impossibility to define grid abstractions of the analog system's (infinite) state-space, such that on each grid box, the property is either true or false.

In this paper, we face the problems of (1) designing a suitable language for modeling analog properties, (2) providing tools to verify properties expressed in such a language over abstractions of analog models (3) formally state the relation of preservation of properties from the abstract model to the (concrete) analog system. The latter problem—fundamental in the spirit of *formal* verification techniques—is indeed not taken in consideration in [5]. It can be proved that their framework allows to preserve only truth of universal quantified properties from the abstract to the concrete models. Our technique, which combines three-valued temporal logic with interval arithmetic techniques, allows us to apply *multivalued* model checking to analog systems (rather than classic 2-valued model checking). We show that in the context of analog circuit verification, the gain of using multivalued model checking is the same achieved for digital (infinite) systems [1]: Namely both truth and falsity of general branching temporal formulæ are preserved.

2. PRELIMINARIES

Interval arithmetic [8] assumes closed intervals over the reals as its fundamental computational object. We denote by $\mathbb{IR}$

GLSVLSI'07, March 11–13, 2007, Stresa-Lago Maggiore, Italy.
.

the set of all such closed intervals of $\mathbb{R}$. Each elementary function on $\mathbb{R}$ (+,-,*,/,*sin*, *cos*, *log*, e^x, x^a) can be easily redefined in the context of $\mathbb{IR}$, by considering the bounds of the corresponding closed intervals. As an example, the rules for recasting each binary operator $\diamond \in \{+,-,*\}$ on the domain $\mathbb{IR}$ are given by $[\underline{x},\overline{x}] \diamond [\underline{y},\overline{y}] = [\underline{z},\overline{z}]$, for $\diamond \in \{+,-,*\}$, with $\underline{z} = \min\{\underline{x}\diamond\underline{y}, \underline{x}\diamond\overline{y}, \overline{x}\diamond\underline{y}, \overline{x}\diamond\overline{y}\}$, and $\overline{z} = \max\{\underline{x}\diamond\underline{y}, \underline{x}\diamond\overline{y}, \overline{x}\diamond\underline{y}, \overline{x}\diamond\overline{y}\}$.

A *box* of dimension n is a subset of $\mathbb{R}^n$ that can be defined as the cartesian product of n closed intervals. We denote by $\mathbb{IR}^n$ the set of all boxes of dimension n.

Definition 2.1 (Inclusion Function). *The function $F : \mathbb{IR}^n \mapsto \mathbb{IR}^m$ is an inclusion function for the function $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ if and only if $\forall b \in \mathbb{IR}^n (f(b) \subseteq F(b))$*

Let $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ be a function expressed as a finite composition of elementary real valued functions. The *natural inclusion function* of f is the inclusion function $[f] : \mathbb{IR}^n \mapsto \mathbb{IR}^m$, obtained by replacing each elementary function by its interval counterpart. In the rest of the paper we deal with functions obtained by composing elementary functions, and it is sufficient to assume the use of natural inclusion functions as the corresponding inclusion functions.

The second part of this preliminary section introduces some basic concepts and notations for analog circuits. Analog circuits can be formally described via a set of differential algebraic equations (DAE):

$$F(x(t), \dot{x}(t), u(t)) \tag{1}$$

where $x \in \mathbb{R}^m$ denotes the vector of system variables, $\dot{x} \in \mathbb{R}^m$ denotes the corresponding time derivatives, and $u \in \mathbb{R}^{m'}$ is the vector of input signals. However, most real-world circuits can be modelled in the so called *semi-explicit* form:

$$\begin{array}{c} \dot{x}(t) = F(x(t), y(t)) \\ 0 = g(x(t), y(t)) \end{array} \tag{2}$$

In this paper, we refer to analog circuits models as *concrete models*, and we denote them by tuples of the form $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I}\rangle$, where:

- X^I is a set of *independent variables*; X^D is a set of *dependent variables*, $X^D \cap X^I = \emptyset$. $\mathcal{I}$ is an invariant on $X^D \cup X^I$.
- $\mathcal{D}$ is a differential rule describing the evolution of dependent variables, that can be written as:

$$\mathcal{D} = \bigwedge_{1 \le i \le p} (\dot{x_i} = f_i(x_1, \ldots, x_n))$$

  where $p = |X^D|$ and $n = |X^D \cup X^I|$.

Given $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I}\rangle$, where $n = |X^D \cup X^I|$, a *state* in $\mathcal{C}$ is a vector $v \in \mathbb{R}^n$ of real values. A *trajectory* of $\mathcal{C}$ is a function $\rho : \mathbb{R} \mapsto \mathbb{R}^n$, which describes a possible time evolution of dependent/independent variables according to the differential rule $\mathcal{D}$ and to the invariant condition $\mathcal{I}$.

3. CTL_F: A TEMPORAL LOGIC FOR SPECIFICATION OF ANALOG PROPERTIES

In this section, we develop a suitable temporal logic for specification of the temporal evolution of analog and mixed signals. Our CTL_f language can be viewed as an enrichment of the classical CTL language [2, 9], originally designed for the temporal analysis of digital finite systems. In particular, we allow CTL_f 'basic formulæ' to state the membership in boxes of values given by arbitrary elementary functional relations. The evolution of these functional relations is then modeled using classical CTL temporal operators.

Definition 3.1 (CTL_f Syntax). *Let $X = \{x_1, \ldots, x_n\}$ be a finite set of real valued variables. The set of CTL_f formulæ is defined by the following grammar:*

$$\begin{array}{lll} \phi ::= & f(x_{i_1}, \ldots, x_{i_m}) \triangleright I & \textit{(basic formulæ testing membership in box } I\textit{)} \\ & \neg\phi | \phi \vee \phi & \textit{(boolean combinators)} \\ & \mathsf{E}\phi\mathsf{U}\phi | \mathsf{A}\phi\mathsf{U}\phi & \textit{(temporal combinators \& path quantifiers)} \end{array}$$

where $1 \le i_1 \le \cdots \le i_m \le n$, $f : \mathbb{R}^m \mapsto \mathbb{R}^p$ is an arbitrary composition of elementary functions, and I is a box in $\mathbb{IR}^p$.

Note that we do not include in our CTL_f logic the usual neXt temporal operator [3]. In fact, since we want to use CTL_f to model *analog* properties, the notion of *next* point in time is meaningless in our context. Other temporal operators (as for example, the operators stating that some property holds Globally or Finally on some path) can be encoded in our grammar (endowed of the only Until temporal operator) by standard definitions [2, 9].

Definition 3.2 (CTL_f Concrete Semantics). *Let $v = (v_1, \ldots, v_n)$ be a state in the concrete system $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I}\rangle$ and let ϕ be a CTL_f formula. Then, the two-shaped value $[v \models \phi] \in \{\mathsf{tt}, \mathsf{ff}\}$ is inductively defined as follows:*

- $[v \models f(x_{i_1}, \ldots, x_{i_m}) \triangleright I] = \mathsf{tt}$ *iff* $f(v_{i_1}, \ldots, v_{i_m}) \in I$.
- $[v \models \phi_1 \vee \phi_2] = \mathsf{tt}$ *iff* $[v \models \phi_1] = \mathsf{tt} \vee [v \models \phi_2] = \mathsf{tt}$.
- $[v \models \neg\phi_1] = \mathsf{tt}$ *iff* $[v \models \phi_1] = \mathsf{ff}$.
- $[v \models \mathsf{E}\phi_1\mathsf{U}\phi_2] = \mathsf{tt}$ *iff there exists a trajectory ρ departing from v and a point of time t such that $[\rho(t) \models \phi_2] = \mathsf{tt}$ and for all $0 \le t' \le t$, $[\rho(t') \models \phi_1] = \mathsf{tt}$.*
- $[v \models \mathsf{A}\phi_1\mathsf{U}\phi_2] = \mathsf{tt}$ *iff for each trajectory ρ departing from v, there exists a point of time t such that $[\rho(t) \models \phi_2] = \mathsf{tt}$ and for all $0 \le t' \le t$, $[\rho(t') \models \phi_1] = \mathsf{ff}$.*

It is well known [7] that simple problems such as *reachability* are generally computationally intractable for analog systems, or even not decidable for hybrid (i.e. mixed analog/digital) systems endowed with rather trivial continuous dynamics. In our context, the systems motivating the analysis are characterized by complex analog behaviours. Moreover, it is likely that the analog properties that we want to check for validity can not be expressed in any decidable theory of the reals.

Hence, to reconcile the need of reasoning on general analog behaviors with the spirit of *automatic* formal analysis, we develop here a notion of *three-valued abstract* semantics for our logic CTL_f. Specifically, we first define the concept of *box modal abstraction* structure (cf. Definition 3.3). On this ground, we use the power of interval arithmetic to evaluate CTL_f formulæ to one of the *three* values $\{\mathsf{tt}, \mathsf{ff}, \bot\}$, on a given box modal abstraction.

DEFINITION 3.3 (BOX MODAL ABSTRACTION (BMA)). *Let $X = \{x_1, \ldots, x_n\}$ be a set of n real valued variables. A* Box Modal Abstraction *on X is a tuple $\langle \mathcal{B}, \overset{must}{\rightarrow}, \overset{may}{\rightarrow} \rangle$, where $\mathcal{B}$ is a finite collection of boxes over $\mathbb{R}^n$; $\overset{must}{\rightarrow}$ and $\overset{may}{\rightarrow}$ are binary relations on $\mathcal{B}$. We assume that $\overset{must}{\rightarrow} \subseteq \overset{may}{\rightarrow} \subseteq \mathcal{B} \times \mathcal{B}$ and that $\overset{may}{\rightarrow}$ is total (i.e. each box has at least one may transition departing from it).*

DEFINITION 3.4 (MUST & MAY PATHS). *Let $\mathcal{A} = \langle \mathcal{B}, \overset{must}{\rightarrow}, \overset{may}{\rightarrow} \rangle$ be a* BMA *on $X = \{x_1, \ldots, x_n\}$. A may-path (must-path) in $\mathcal{A}$ is an infinite sequence of boxes $\{b_i\}_{i \in \mathbb{N}}$ such that for all $i \geq 0$ it holds that $b_i \overset{may}{\rightarrow} b_{i+1}$ ($b_i \overset{must}{\rightarrow} b_{i+1}$). We use the notation $\{b_i\}^{may}_{i \in \mathbb{N}}$ ($\{b_i\}^{must}_{i \in \mathbb{N}}$) to denote may-paths (must-paths). A may-subpath (must-subpath) is a finite prefix of a may-path (must-path).*

DEFINITION 3.5 (CTL_f ABSTRACT SEMANTICS). *Let $\mathcal{A} = \langle \mathcal{B}, \overset{must}{\rightarrow}, \overset{may}{\rightarrow} \rangle$ be a* BMA *on $X = \{x_1, \ldots, x_n\}$. Given $b \in \mathcal{B}$ and a $\phi \in \mathsf{CTL}_f$, we define $[b \models_3 \phi]$ to be one of the three values in $\{\mathsf{tt}, \mathsf{ff}, \bot\}$, according to the following rules. If $\phi = f(x_{i_1}, \ldots, x_{i_m}) \rhd I$, then $[b \models_3 \phi]$ is defined as:*

$$\begin{cases} \mathsf{tt}, & \text{if } [f](b) \subseteq I; \\ \mathsf{ff}, & \text{if } [f](b) \cap I = \emptyset; \\ \bot, & \text{otherwise.} \end{cases}$$

If $\phi = \neg\phi_1$, then $[b \models_3 \phi]$ is defined as:

$$\begin{cases} \mathsf{tt} & \text{if } [b \models_3 \phi_1] = \mathsf{ff}; \\ \mathsf{ff} & \text{if } [b \models_3 \phi_1] = \mathsf{tt}; \\ \bot & \text{otherwise.} \end{cases}$$

If $\phi = \phi_1 \vee \phi_2$, then $[b \models_3 \phi]$ is defined as:

$$\begin{cases} \mathsf{tt} & \text{if } [b \models_3 \phi_1] = \mathsf{tt} \vee [b \models_3 \phi_2] = \mathsf{tt}; \\ \mathsf{ff} & \text{if } [b \models_3 \phi_1] = \mathsf{ff} \wedge [b \models_3 \phi_2] = \mathsf{ff}; \\ \bot & \text{otherwise.} \end{cases}$$

If $\phi = \mathsf{E}\phi_1 \mathsf{U} \phi_2$, then $[b \models_3 \phi]$ is defined as:

$$\begin{cases} \mathsf{tt} & \text{if there exists a must-subpath } \{b_i\}^{must}_{i=0..k} \text{ departing from } b \text{ such that:} \\ & ([b_k \models_3 \phi_2] = \mathsf{tt}) \wedge \forall 0 \leq i \leq k([b_i \models_3 \phi_1] = \mathsf{tt}); \\ \mathsf{ff} & \text{if for all may-path } \{b_i\}^{may}_{i \in \mathbb{N}} \text{ departing from } b \text{ and for all } i \geq 0, \text{ it holds that:} \\ & ([b_i \models_3 \phi_2] \neq \mathsf{ff}) \rightarrow \exists j < i([b_j \models_3 \phi_1] = \mathsf{ff}) \\ \bot & \text{otherwise.} \end{cases}$$

If $\phi = \mathsf{A}\phi_1 \mathsf{U} \phi_2$, then $[b \models_3 \phi]$ is defined as:

$$\begin{cases} \mathsf{tt} & \text{if for all may-path } \{b_i\}^{may}_{i \in \mathbb{N}} \text{ departing from } b: \text{ there exists an index } i \text{ such that:} \\ & ([b_i \models_3 \phi_2] = \mathsf{tt}) \wedge \forall 0 \leq j \leq i([b_i \models_3 \phi_1] = \mathsf{tt}); \\ \mathsf{ff} & \text{if there exists a must-subpath } \{b_i\}^{must}_{i=1\ldots k} \text{ departing from } b \text{ such that:} \\ & ([b_k \models_3 \phi_1] = \mathsf{ff}) \text{ and } \forall j < i([b_j \models_3 \phi_2] = \mathsf{ff}) \\ \bot & \text{otherwise.} \end{cases}$$

4. RELATING ABSTRACT & CONCRETE SEMANTICS: PRESERVATION RESULTS

An immediate motivation to the choice of defining a *three*-valued abstract semantics for our CTL_f logic is the naturalness of having discrete abstract states where it is not possible to determine a definitive true or false value for basic CTL_f formulæ. In this section, we show that the above choice has a further fundamental advantage. In fact, it allows to apply 3-valued model checking techniques—rather than classical (2-valued) model checking—for the automated reasoning on analog circuits. In the context of digital systems, three-valued abstractions and model checking [1] have the advantage of preserving *both truth and falsity* of totally branching temporal logic from so called partial Kripke structures (playing the rule of abstract systems) to complete Kripke structures (playing the role of concrete digital systems). On the contrary, classical model checking allows only the preservation of formulæ containing exclusively universal path quantifiers and evaluating to the truth over the abstraction. We prove that the same gain is obtained in the context of analog systems abstraction/verification, when using three-valued logics. In order to formalize the above ideas, we introduce the notion of *preserving* box modal abstractions for a given analog model $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$. The conditions ensuring a box modal abstraction to be preserving for $\mathcal{C}$ naturally extends the notion of *completeness preorder* in [1], to the case in which the concrete system evolves according to continuous trajectories (rather than following discrete paths). Let $\prec$ to be the partial relation on $\{\mathsf{tt}, \mathsf{ff}, \bot\}$ such that $\bot \prec \mathsf{tt}, \bot \prec \mathsf{ff} \wedge \forall a \in \{\mathsf{tt}, \mathsf{ff}, \bot\}(a \prec a)$.

DEFINITION 4.1 (PRESERVING BMA). *Consider an analog model $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$ and let $n = |X^D \cup X^I|$. A box modal abstraction $\mathcal{A} = \langle \mathcal{B}, \overset{must}{\rightarrow}, \overset{may}{\rightarrow} \rangle$ on $X = X^D \cup X^I$ is said* preserving *for $\mathcal{C}$ iff there exists a relation $\lessdot \subseteq \mathcal{B} \times \mathbb{R}^n$ such that, whenever $b \lessdot v$, the following rules hold:*

1. *For each basic CTL_f formula ϕ, $[b \models_3 \gamma] \prec [v \models \gamma]$.*
2. *If $\{b_i\}^{must}_{1 \leq i \leq k}$ is a must-subpath departing from b in $\mathcal{A}$, then $\mathcal{C}$ admits a trajectory ρ departing from v such that there exists an increasing sequence of real values $\{t_i\}_{0 \leq i \leq k}$ such that:*
$$\forall t \in \mathbb{R}(t_{i-1} \leq t \leq t_i \rightarrow b_i \lessdot \rho(t))$$
3. *If $\mathcal{C}$ admits a trajectory $\rho(t)$ departing from v, then $\mathcal{A}$ admits a may-path $\{b_i\}^{may}_{i \in \mathbb{N}}$ such that there exists an increasing sequence of real values $\{t_i\}_{i \in \mathbb{N}}$ such that:*
$$\forall t \in \mathbb{R}(t_i \leq t \leq t_{i+1} \rightarrow b_i \lessdot \rho(t))$$

Given a preserving box modal abstraction $\mathcal{A}$ of $\mathcal{C}$, the preserving concretization *for $\mathcal{A}$ and $\mathcal{C}$ is the maximum relation satisfying the three above conditions.*

THEOREM 4.2. *Let $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I} \rangle$ to be an analog model and let $n = |X^D \cup X^I|$. If $\mathcal{A} = \langle \mathcal{B}, \overset{must}{\rightarrow}, \overset{may}{\rightarrow} \rangle$ is a preserving box modal abstraction for $\mathcal{C}$, then:*

$$\forall b \in \mathcal{B}, \forall v \in \mathbb{R}^n, \forall \phi \in \mathsf{CTL_f}((b \lessdot v) \rightarrow ([b \models_3 \phi] \prec [v \models \phi]))$$

where $\lessdot \subseteq \mathcal{B} \times \mathbb{R}^n$ denotes the preserving concretization relation for $\mathcal{A}$ and $\mathcal{C}$.

BuildBMA$(\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I}\rangle, b_0, k)$

Input: *a concrete analog model* $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I}\rangle$ *and a natural number* k

Output: *A* CTL_f *preserving box modal abstraction* $\mathcal{A} = \langle \mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\rangle$ *for* $\mathcal{C}$

% *Step 1. Definition of the collection of boxes* $\mathcal{B}$ *in the box modal abstraction for* $\mathcal{C}$
(1) Let $\mathcal{B}_1$ be the set of boxes obtained by performing an uniform k-decomposition of b_0
(2) Let $\mathcal{B}_2$ be the set of (unbounded) boxes obtained by performing an arbitrary box decomposition in $\mathbb{R}^{|X^D \cup X^I|} \setminus b_0$
(3) $\mathcal{B} \leftarrow \mathcal{B}_1 \cup \mathcal{B}_2$
% *Step 2. Definition of the may transitions in the box modal abstraction of* $\mathcal{C}$
(4) $\overset{may}{\rightarrow} := \{\}$
(5) **for each** (pair p of neighbouring boxes in $\mathcal{B}$ having F as a common face) **do**
(6) Let i such that $(p = \langle b_1, b_2\rangle \wedge F$ is the i lower face of $b_1 \wedge F$ is the i upper face of $b_2)$
(7) **if** $(i \leq |X^D| \wedge [\dot{x}_i](F) \cap [0, +\infty] \neq \emptyset)$ **then** $\overset{may}{\rightarrow} := \overset{may}{\rightarrow} \cup \{(b_1, b_2)\}$
(8) **if** $(i \leq |X^D| \wedge [\dot{x}_i](F) \cap [-\infty, 0] \neq \emptyset)$ **then** $\overset{may}{\rightarrow} := \overset{may}{\rightarrow} \cup \{(b_2, b_1)\}$
(9) **for each** $(b \in \mathcal{B})$ **do**
(10) **if** $(\forall 1 \leq i \leq |X^D| (0 \in [\dot{x}_i](b)))$ **then** $\overset{may}{\rightarrow} := \overset{may}{\rightarrow} \cup \{(b, b)\}$
% *Step 3. Definition of the must transitions in the box modal abstraction of* $\mathcal{C}$
(11) $\overset{must}{\rightarrow} := \{\}$
(11) **for each** $(b \in \mathcal{B})$ **do**
(12) **if** $(b \overset{may}{\rightarrow} b'$ is the only may transition sourcing from $b)$ **then** $\overset{must}{\rightarrow} := \overset{must}{\rightarrow} \cup \{(b, b')\}$

Figure 1: Computing a CTL_f Preserving Box Modal Abstraction.

5. CTL_F MODEL CHECKING

Let $\mathcal{C}$ be an analog model. The aim of this section is of showing how to use interval arithmetic to define a preserving box modal abstraction for $\mathcal{C}$. Given a $\phi \in \mathsf{CTL}_f$, it is then possible to compute its abstract semantics over $\mathcal{A}$. Finally, (underapproximations of) sets of states in $\mathcal{C}$ against which ϕ evaluates to true/false can be obtained via the concretization relation for $\mathcal{C}$ and $\mathcal{A}$. In detail, Figure 1 depicts a procedure to extract a preserving box modal abstraction from an analog model[1] $\mathcal{C} = \langle X^D, X^I, \mathcal{D}, \mathcal{I}\rangle$. The procedure BuildBMA is composed by three macro steps, one for each component in the box modal abstraction $\mathcal{A} = \{\mathcal{B}, \overset{may}{\rightarrow}, \overset{must}{\rightarrow}\}$, which is computed as output. In particular, the first macrostep uses an input parameter $k \in \mathbb{N}$, to perform an uniform subdivision in each component of the box of interest b_0 (also given as input). The remaining space in $\mathbb{R}^{|X|} \setminus b_0$ is covered by an arbitrary decomposition into (possibly unbounded) boxes. The second macro step uses simple interval arithmetic techniques to define the set of may-transitions in the abstract model. Finally, in the third macro step, some of the previously defined may transitions are considered to be colored also as must transition.

Theorem 5.1. *Given* $k \in \mathbb{N}$ *and an analog system model* $\mathcal{C}$*, the procedure* BuildBMA$(\mathcal{C}, k, b_0)$ *computes a* CTL_f *preserving box modal abstraction for* $\mathcal{C}$*.*

Assume that a box modal abstraction $\mathcal{A}$ has been generated from a concrete system $\mathcal{C}$ using the algorithm BuildBMA, and let ϕ be a CTL_f formula. The problem of checking, for each box $b \in \mathcal{B}$, whether $b \models_3 \phi$ can be easily solved on the base of the three-valued classic CTL model checking for digital systems [1]. More precisely:

- According to Definition 3.5, interval arithmetic can be used to label each box $b \in \mathcal{B}$ with the set of pairs:
$$\{\langle \gamma_i, [b \models_3 \gamma_i]\rangle \mid \gamma_i \text{ is a subformula of } \phi\}$$
- Each composed CTL_f formula, can be dealt with using the corresponding subprocedure for three-valued classic CTL model checking on digital systems.

[1]Here, we assume to deal with a input model such that input variables maintain a constant (arbitrary) value overall the execution of the system. However, a symmetrical extreme parameter model in which input variables are admitted to assume any value while the system evolves, could be also easily considered for the abstraction generation. Note that the same extreme input models were adopted in [5].

6. REFERENCES

[1] G. Bruns *et al*, "Model Checking Partial State Spaces with Three Valued Temporal Logics," *LNCS Proc. of the 13th Conference on Computer Aided Verification*, 2001.
[2] E. Clarke *et al*,, "Model Checking," *MIT Press*, 1999.
[3] E. Clarke *et al*,, "Design and synthesis of synchronization skeletons using brancing time temporal logic," *LNCS Proc. of Workshop on Logic of Programs*, 1999.
[4] P. Gray *et al*, "Analysis and Design of Analog Integrated Circuits," *John Wiley & Sons Inc.*, 1993.
[5] W. Hartong *et al*, "Model checking algorithms for analog verification," *Proc. of the 39th Conference on Design Automation (DAC'02)*, 2002.
[6] E. Hennig, "Symbolic Approximation and Modeling Techniques for Analysis and Design of Analog Circuits," *Shaker Verlag*, 2000.
[7] T. Henzinger *et al*,, "What's decidable about hybrid automata?," *ACM Proc. of the 27th Annual Symposium on Theory of Computing*, 1995.
[8] L. Jaulin *et al*, "Applied Interval Analysis," *Springer Verlag*, 2001.
[9] K. Schneider, "Verification of Reactive Systems," *Springer Verlag*, 2004.
[10] R. Sommer *et al*, "A Generic Circuit Modelling Strategy Combining Symbolic and Numerical Analisys," *Proc. of the 5th Int. Workshop on on Symbolic Methods and Applications to Circuit Design (SMCAD'98)*, 1998.
[11] R. Sommer *et al*, "Analog Insydes 2 - New Features and Applications in Circuit Design," *Proc. 6th Int. Workshop on Symbolic Methods and Applications in Circuit Design (SMACD 2000)*, 2000.