Formally Verifying Transfer Functions of Linear Analog Circuits

Syeda Hira Taqdees and Osman Hasan
National University of Sciences and Technology (NUST)

Editor's note:
This article presents an approach to extend mathematical formal analysis towards verification of linear analog circuits.

—Jayanta Bhadra, NXP

The Internet of Things (IoT) is characterized as a broad network of several physical entities, such as embedded systems, softwares, electronics, and electrical machinery. It provides an integration platform to these subsystems for exchanging information and interacting with the continuously changing physical surroundings. Due to the convergent nature, the concept of IoT System-on-Chip (SoC) devices has been made realizable and they are widely being used in a variety of applications, ranging from consumer electronic devices, such as tele-operated health-care units and autonomous vehicles, to safety critical domains, such as tele-surgical robotics, space travel and smart disaster response and evacuation. However, these SoC devices contain analog and sensor circuitry and there is a dire need for efficient mixed-signal verification methodologies. Usually, the methodologies used for their functional verification are dynamic and mostly depend on the effectiveness and rigor of testing procedures. However, their modeling must rely on specialized mathematical and theoretical basis for consistent verification.

Traditionally, the analog and sensor circuitry of SoC is analyzed using the state-space models, i.e., capturing the behavior of different components by appropriate differential equations and then solving these differential equations to obtain the required design constraints. However, as the complexity and parallelism of the continuous components in an SoC increases, the state-space models become inefficient and in certain cases impossible to capture. Various transformation techniques, such as the Laplace and an exponential transforms, are used to convert the state-space models to the corresponding transfer function models in order to analyze various design metrics. Specifically, the Laplace transform, which is an integral transform method, is widely used to convert the time varying signals and continuous models to their corresponding *s*-domain representations while analyzing linear analog circuits. This transformation provides a very compact representation of the overall behavior of the given time varying signals and continuous models. The Laplace transform theory allows us to solve state-space models using simple algebraic techniques as the transformation allows us to convert the integration and differentiation functions from the time-domain to multiplication and division functions in the *s*-domain.

The analog components of SoC are usually analyzed using computer-based testing or simulation methods, where the main idea is to deduce

Digital Object Identifier 10.1109/MDAT.2017.2713388
Date of publication: 8 June 2017; date of current version: 13 September 2017.

Copublished by the IEEE CEDA, IEEE CASS, IEEE SSCS, and TTTC

the validity of a property by observing its behavior for some test cases. Whereas, both state-space and transfer function models of continuous components, based on differential equation algebra and the Laplace transform methods, respectively, are analyzed using computer simulations, computer-based numerical techniques, or symbolic methods. However, results obtained via these traditional methods cannot be termed as 100% accurate due to the approximations introduced by using computer arithmetics, such as floating or fixed point numbers, for constructing computer-based models of the continuous physical components. Moreover, the circuits are analyzed for some specific test cases only since exhaustive simulation is not possible due to the continuous nature of inputs and even the simulations for a subset of possible test cases may take several days. For example, numerical methods cannot ascertain an accurate value of the improper integral of the Laplace transform as there is always a limited number of iterations allowed depending on the available memory and computational resources. Due to these limitations, more rigorous and accurate analysis techniques for analyzing continuous components of an SoC are actively being sought out and formal verification, i.e., a computer-based mathematical analysis technique, offers a promising solution.

In the past couple of decades, formal verification methods [1] have been successfully used for the precise analysis of a variety of software, hardware, and physical systems. The main principle behind formal analysis of a system is to construct a computer-based mathematical model of the given system and formally verify, within a computer, that this model meets rigorous specifications of intended behavior. Given the extensive usage of SoC in safety critical applications, there is a dire need of using formal methods for their analysis. However, the frequent involvement of complex-valued physical quantities, ordinary differential equations and the Laplace transformation in their analysis are the main limiting factors in this direction. The automatic state-based formal methods, such as model checking, SMT solvers, and automatic theorem provers cannot be used to model and analyze the true SoC models due to their inability to model continuous systems. This is the main reason why most of the formal verification work about SoC utilizes their abstracted discrete models. In [2], conformance checking techniques have been presented to show the equivalence between the specified and implemented transfer function of analog circuits. In these techniques, the verification ideas are primarily based on the discretization of the s-domain transfer functions to the z-domain using the bilinear transformation, which raises issues, like the error analysis of transfer function coefficients and the state-space explosion when the inherited discretization of the design is encoded for larger models. Model checking [1] has also been used to formally verify continuous components of SoC but the entire model checking-based techniques work with the abstraction of continuous dynamics because of the inability of these methods to model and analyze continuous systems in their true form. Thus, despite the inherent soundness of formal verification methods, such analysis cannot be termed as absolutely accurate.

We propose to use higher-order-logic theorem proving [1] for formally verifying transfer function models of linear analog components of SoC. Higher-order logic is a system of deduction with a precise semantics and, due to its high expressiveness, can be used to describe any mathematical relationship, including the state-space and transfer function models of continuously varying analog components of SoC devices and their desired transfer function specifications. Their equivalence can then be verified within the sound core of a theorem prover. Due to the high expressibility of higher-order logic, the proposed approach is very flexible in terms of analyzing a variety of SoC devices and transfer functions.

In this paper, as a first step toward the proposed direction, we develop a generic methodology for the verification of transfer functions of linear analog circuits. We mainly extend our existing work on the formalization of the Laplace transform theory [3] by formally verifying the Laplace transforms of important trigonometric functions, such as exponential and sines/cosines functions. These functions are extensively required for verifying many continuous aspects of SoC, such as the voltage analysis of analog components. In addition, they are the fundamental entities in geometric control theory, which is commonly used for modeling the feedback-based control components of SoC as well as for the reasoning of security of overall systems. For the application of our methodology to the linear analog circuit portion of SoC, we also formalize the well-known Kirchhoff's voltage and current laws (commonly known as KVL and KCL) and a few basic components of analog circuits, like resistor, inductor, and capacitor. Based on these results, transfer function models of a wide range of linear analog circuits of an SoC device can be formally verified within the sound core of a higher-order-logic theorem prover and the paper presents a stepwise methodology for this purpose.

Related Work

Denman et al. [4] proposed a functional verification approach for analog circuits using MetiTarski, which is an automated theorem prover for real-valued trigonometric functions. The behavioral model of the analog circuit is transformed into its closed form solution by using the inverse Laplace (*invlaplace*) function of Maple and an inequality relating the closed form solution with the required property is fed to MetiTarski, which in turn determines if the inequality holds and in this case also generates the corresponding formal proof. A similar approach is also proposed in [5] for the verification of analog circuits using MetiTarski in the presence of noise and process variation by introducing stochastic modeling. Tiwari et al. [6] proposed to use piecewise interval device modeling for analog circuits and these models were used to verify DC-analysis properties using SAT solvers. Besides verifying the DC-analysis related properties, formal verification methods have also been used in the context of verifying transient properties using traditional model checking [7].

However, all the above-mentioned techniques do not aim for the transfer function analysis and deal only with the real-valued analog quantities compromising the complex-valued solution of the modeling differential equations. However, the complete solution of the modeling differential equation must also include the imaginary part, which provides the phase information and also helps in analyzing the steady-state behavior of the circuit in functional verification [8]. Symbolic methods, provided by Maple and Mathematica, are based on algorithms that consider the improper integral of the Laplace transform as the continuous analog of the power series, i.e., the integral is discretized to summation and the complex exponentials are sampled. Moreover, the usage of computer algebra algorithms, which are unverified (cf. [4] p. 3), for calculating the closed form solution of the behavioral model also compromises on the accuracy of the analysis.

These formal and semiformal techniques have also been used for verifying some basic constituent components and building blocks of analog circuits, like operational amplifier (op-amp) [4], oscillators [4], op-amp integrator [5], phase locked-loop [9], and a frequency domain equalizer [10]. The proposed technique is generic enough to cater for the verification of all these components and their arbitrary combinations. For illustration purpose, we present the verification of Sallen-Key low-pass filters, which are quite compatible in complexity to the existing formally verified circuits.

Formalization of Laplace Transform

In this section, we present a brief overview of our formalization of the Laplace transform theory using the HOL-Light theorem prover [3]. Mathematically, the Laplace transform is a complex function defined for a function *f*, which can be either real or complex-valued, as follows:

$$F(s) = \int_0^\infty f(t)\, e^{-st} dt,\ s \in \mathbb{C}. \qquad (1)$$

The Laplace transform function can be formally defined as

Definition 1: Laplace Transform

`⊢ ∀ s f. laplace f s = lim at_posinfinity` $(\lambda b. \int_0^b f(t)e^{-st}dt)$

In the above definition, the function `laplace` accepts a complex number *s* and a complex-valued function *f*. It returns a complex number representing the Laplace transform of *f*. The limit of improper integral of laplace transform is modeled by using the `lim at_posinfinity` function of HOL-Light [11].

We have also verified some of the classical properties of the Laplace transform, provided in Table 1, which play a vital role in the analysis of linear analog circuits.

Proposed Methodology

The proposed methodology for the formal verification of transfer functions of linear analog circuits of an SoC is shown in Figure 1. The inputs required for the proposed verification methodology are (*A*) a structural view of the given analog circuit of SoC representing the connections of its sub-components, (*B*) the modeling differential equation of the given circuit relating its input and output quantities in the time domain, and (*C*) the transfer function representing the required behavior in the *s*-domain. The first step in

Table 1 Formalization of Laplace transform properties.

Property	Mathematical Form	Formalized Form
Limit Existence	$\exists l. \int_0^\infty f(t)e^{-st}dt = l$	$\vdash \forall$ f s.laplace_exists f s $\Rightarrow$ ($\exists$l.lim at_posinfinity ($\lambda b. \int_0^b f(t)e^{-st}dt$) = l)
Linearity	$(\mathcal{L}\ \alpha f(x) + \beta g(x)\ )(s) = \alpha(\mathcal{L}f)(s) + \beta(\mathcal{L}g)(s)$	$\vdash \forall$ f g s a b.laplace_exists f s $\wedge$ laplace_exists g s $\Rightarrow$ laplace (λx.a$\star$f x + b$\star$g x) s = a$\star$laplace f s + b$\star$laplace g s
Frequency Shifting	$(\mathcal{L}\ e^{bt}f(t))(s) = (\mathcal{L}f)(s-b)$	$\vdash \forall$ f s b.laplace_exists f s $\Rightarrow$ laplace (λt.e^{bt} $\star$ f t) s = laplace f (s-b)
Uniqueness	$f = g \Rightarrow (\mathcal{L}f)(s) = (\mathcal{L}g)(s)$	$\vdash \forall$ f g s.($\forall$x. 0<x $\Rightarrow$ f x = g x) $\wedge$ laplace_exists f s $\wedge$ laplace_exists g s $\Rightarrow$ laplace f s = laplace g s
Integration	$(\mathcal{L}\int_0^t f(\tau)d\tau)(s) = \frac{1}{s}(\mathcal{L}f)(s)$	$\vdash \forall$ f s.(0<Re s) $\wedge$ (laplace_exists ($\lambda x. \int_0^x f(t)dt$) s) $\wedge$ (laplace_exists f s) $\wedge$ ($\forall$x.f continuous_on [0,x]) $\Rightarrow$ ($\lambda x. \int_0^x f(t)dt$) s = inv(s) $\star$ laplace f s
General Differentiation	$(\mathcal{L}\frac{d^n f}{dx^n})(s) = s^n(\mathcal{L}f)(s) - \sum_{k=1}^n s^{k-1}\frac{d^{n-k}f(0)}{dx^{n-k}}$	$\vdash \forall$ f s n. (laplace_exists ($\lambda x.\frac{d^n f}{dx^n}$) s) $\wedge$ ($\forall$x. differentiable n f (at x)) $\Rightarrow$ laplace ($\lambda x.\frac{d^n f}{dx^n}$) s = $s^n$$\star$laplace f s$-\sum_{k=1}^n(\lambda y.\ s^{y-1} * \frac{d^{n-y}f(0)}{dx^{n-y}})$

the proposed methodology is to translate the structural representation of the given circuit to its corresponding higher-order-logic function using the component definitions available in the formalized analog library. This provides us with our implementation model as shown in Figure 1. The next step in the proposed methodology is to formalize the given modeling differential equation and the transfer function in higher-order logic to get the formal differential equation based specification and the formal transfer function based specification, respectively. These translations can be done based on the available multivariable calculus formalizations in HOL-Light. The next step is to formally verify the implication between the implementation model and the formal differential equation based specification of the given circuit, i.e., $A \rightarrow B$. This verification can be done in a very straightforward way based on the circuit simplifier functions of formalized analog library and some simple arithmetic reasoning. The next step in the proposed methodology is to verify that the differential equation specification of the given circuit implies the given transfer function specification, i.e., $B \rightarrow C$, using the formalized Laplace transform theory and arithmetic reasoning. The two implications verified in the last two steps also imply that the given structural view of the circuit implies the given transfer function-based specification, which concludes the formal verification of the desired result within the sound core of the theorem prover. Once the transfer function verification is done, circuit behavior at a specific input voltage can also be verified by using the formalized Laplace transforms of commonly used analog functions available in our analog library.

The distinguishing features of this methodology include the higher confidence in the verification results due to the usage of pure complex and real number data-types for modeling the given circuit and the usage of theorem proving for the verification. It is important to note that, just like any other verification approach, the proposed methodology requires the circuit and its desired behavior to be known a priori and it just allows us to formally verify that they correspond to one another.

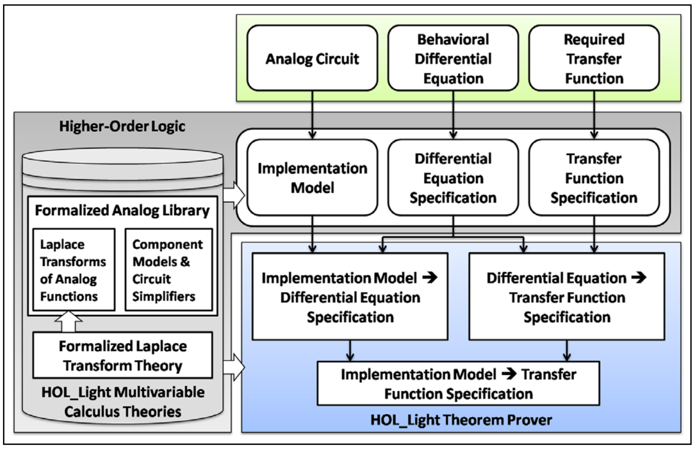
Analog Circuit
Behavioral Differential Equation
Required Transfer Function
Higher-Order Logic
Formalized Analog Library
Laplace Transforms of Analog Functions
Component Models & Circuit Simplifiers
Formalized Laplace Transform Theory
HOL_Light Multivariable Calculus Theories
Implementation Model
Differential Equation Specification
Transfer Function Specification
Implementation Model ➔ Differential Equation Specification
Differential Equation ➔ Transfer Function Specification
Implementation Model ➔ Transfer Function Specification
HOL_Light Theorem Prover

Figure 1. Proposed methodology for the formal verification of linear analog circuits.

Formalization of Analog Library

In this section, we explain our formalization of the various analog components, circuit simplification rules and analog functions.

Analog Components and Circuit Simplification Rules

We begin by formalizing the voltage and current expressions for a resistor, capacitor, and inductor, which are the most commonly used analog circuit components, as the following higher-order-logic functions:

Definition 2: Resistor, Inductor, and Capacitor

```
⊢ ∀ R i. res_vol R i = ( λt.i t * R )
⊢ ∀ R v. res_cur R v = ( λt.v t / R )
⊢ ∀ L i. ind_vol L i =
  ( λt.L * (vector_derivative i (at t)) )
⊢ ∀ L v Io. ind_cur L v Io =
  ( λt.Io + ( 1 / L)) *
    integral (interval [0, t]) v )
⊢ ∀ C i Vo. cap_vol C i Vo =
  ( λt.Vo + ( 1 / C ) *
    integral (interval [0 , t]) i )
⊢ ∀ C v. cap_cur =
  ( λt.C * (vector_derivative v (at t)) )
```

where `(λx. f(x))` represents a lambda abstraction function that accepts a variable *x* and returns *f*(*x*). The functions *i* and *v* represents the time-dependent current and voltage, respectively. While the variables *R*, *L,* and *C* represent the resistance, inductance, and the capacitance of their respective components, respectively. The function `vector_derivative` is used for formalizing the differentiation of complex voltage and current. Whereas, the function `integral` is used for integration over the vector space. The variables *Io* and *Vo* are used in the definitions of the inductance and capacitance to model the initial current in the inductor and the initial voltage across the capacitor, respectively.

The KVL and KCL state that the directed sum of all the voltage drops around any closed network (loop) of an electrical circuit and the directed sum of all the branch currents leaving an electrical node is zero, respectively. Mathematically:

$$\sum_{k=1}^{n} V_k = 0, \sum_{k=1}^{n} I_k = 0 \qquad (2)$$

where V_k and I_k represent the voltage drops across the *k*th component in a loop and the current leaving the *k*th branch in a node, respectively. Their formalization is as follows:

Definition 3: Kirchhoff's Voltage and Current Law

```
⊢ ∀ V t. kvl V t =
  vsum (0..(LENGTH V-1)) (λn. EL n V t) = 0
⊢ ∀ V t. kcl I t =
  vsum (0..(LENGTH I-1)) (λn. EL n I t) = 0
```

The function `kvl` accepts a list *V*, which represents the behavior of time-dependent voltages in the given circuit and a time variable *t*. It returns the predicate that guarantees that the sum of all the voltages in the loop is zero. Similarly, the function `kcl` accepts a list *I*, which represents the behavior of time-dependent currents and a time variable *t* and returns the predicate that guarantees that the sum of all the currents leaving the node is zero. Based on the proposed methodology, given in Figure 1, the above-mentioned definitions can be used to develop formal models of a wide range of linear analog circuit implementations.

Analog Functions

In order to facilitate the formal analysis of linear analog circuits using the formalized Laplace transform theory, as depicted in Figure 1, we illustrate the process of verifying the Laplace transforms of some commonly used analog signals, such as sine, cosine, exponential decay, and exponential growth. These functions are presented in their mathematical form in Table 2, while their formalized form can be found in [12]. As mentioned earlier, these formalizations are very useful in verifying the transfer functions of linear analog circuits. Moreover, these functions can be used to capture the dimensional model of an SoC in order to reason about the dimensional consistency and invariability using theorem proving.

Application: Sallen-Key Low-Pass Filters

In order to illustrate the practical effectiveness and utilization of the proposed methodology for verifying real-world analog circuits, we have verified the transfer functions of first- and second-order Sallen-Key low-pass filters in this section. Sallen-Key is one of the most widely used filter topologies and Sallen-Key low-pass filters are extensively being used in numerous applications, such as analog-to-digital converters, radio transmitters, audio crossover, and telephone lines. They are also used as the basic building blocks of other higher-order low-pass filters. The main motivation behind choosing Sallen-Key filters as an application for our work is the enormous usage of filters in IoT devices, particularly while identifying devices and accessing their information.

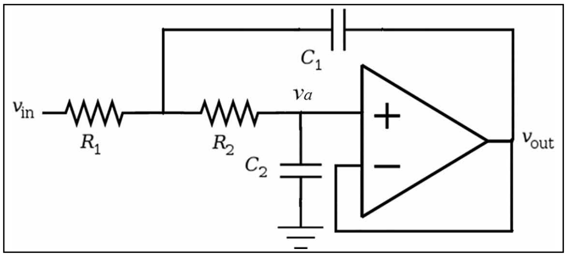

Figure 2. Second-order Sallen-Key low-pass filter.

We will explain the verification of second-order Sallen-Key low-pass filter, depicted in Figure 2, in detail. Its modeling differential equation and transfer function are as follows:

$$R_1C_1R_2C_2\frac{d^2v_{out}(t)}{dt^2}+C_2(R_1+R_2)\frac{dv_{out}(t)}{dt}+v_{out}(t)=v_{in}(t) \quad (3)$$

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{R_1C_1R_2C_2s^2+C_2\,(R_1+R_2)\,s+1}. \quad (4)$$

By using our formal analog library definitions, the implementation model for the second-order low-pass filter is obtained as follows:

Definition 4: Implementation of Second-Order LP Filter

```
⊢ ∀ R1 C1 R2 C2 Vin Vout Va Vb.
  LP_imp R1 C1 R2 C2 Vin Vout Va Vb =
  (∀t. 0 < t ⇒ kcl[res_cur R1 (Vin - Va);
    res_cur R2 -(Va - Vb);
    cap_cur C1 -(Va - Vout)] t) ∧
  (∀t. 0 < t ⇒ kcl [res_cur R2 (Va - Vb);
    cap_cur C2 -Vb] t) ∧
  (∀t. 0 < t ⇒ Vb t = Vout t)
```

where V_a represents the voltage at the node joining R_1, R_2, and C_1 and V_b represents the voltage at noninverting input of the op-amp in Figure 2. The first conjunct represents the node joining R_1, R_2, and C_1 using the formalized KCL, whereas the second conjunct represents the node at the noninverting input of the op-amp. In the given circuit, the op-amp is being used in the negative-feedback configuration. The third conjunct in the above definition represents this negative-feedback configured op-amp.

The next step is to formalize the differential equation and the required transfer function of the given second-order low-pass filter as follows:

Definition 5: Differential Equation of Second-Order LP Filter

```
⊢ ∀ R1 C1 R2 C2 Vout Vin y. LP_behav R1 C1 R2
C2 Vout Vin y =
 diff_eq_lhs [λt.1; λt.C2 * (R1 + R2);
   λt.R1 * C1 * R2 * C2] Vout 3 y = Vin y
```

Definition 6: Transfer Function of Second-Order LP Filter

```
⊢ ∀ R1 C1 R2 C2 Vin Vout s. tf_spec R1 C1 R2
C2 Vin Vout s =
  laplace Vout s / laplace Vin s =
   1 / (R1 * C1 * R2 * C2) * (s pow 2) +
    C2 * (R1 + R2) * s + 1
```

where the function diff_eq_lhs is used to formalize the left-hand side of a differential equation. Then, the following theorem representing the implication between the implementation and the formal differential equation specification can be verified:

Theorem 1: Implementation Implies Differential Equation

```
⊢ ∀ R1 C1 R2 C2 Vin Va Vb Vout.
 0 < R1 ∧ 0 < C1 ∧ 0 < R2 ∧ 0 < C2 ∧
 ∀t. differentiable 2 Vout (at t)) ∧
 ∀t. differentiable 2 Vin (at t) ∧
 ∀t. differentiable 2 Va (at t) ∧
 LP_imp R1 C1 R2 C2 Vin Vout Va Vb ⇒
 ∀t.(0 < t) ⇒ LP_behav R1 C1 R2 C2 Vin Vout t
```

In the above theorem, the first four assumptions ensure that the resistor's and the capacitor's values in the given circuit must be greater than zero, which is the necessary condition for the circuits to exhibit the behavior of (3). In the next three assumptions, the differentiable function is used to ensure the differentiability of the input–output and the nodal voltage of the circuit which is also a necessary condition. The proof of Theorem 1 is based on the function definitions along with some multivariable arithmetic reasoning and is thus very straightforward.

Next, we verify the implication between differential equation and transfer function specification as follows:

Theorem 2: Differential Equation Implies Transfer Function

```
⊢ ∀ R1 C1 R2 C2 Vin Vout t s.
 0 < R1 ∧ 0 < C1 ∧ 0 < R2 ∧ 0 < C2 ∧
 ∀t. differentiable 2 Vout (at t) ∧
 ∀t. differentiable 2 Vin (at t) ∧
 ∀t. differentiable 2 Va (at t) ∧
 laplace_exists_higher_deriv 2 Vout s ∧
 laplace_exists_higher_deriv 2 Vin s ∧
 laplace Vin s ≠ 0 ∧ laplace Vin s ≠0)∧
 (∀t.(0 < t ⇒
  LP_behav R1 C1 R2 C2 Vin Vout t) ∧
    (t = 0 ⇒ Vin = 0 ∧ Vout = 0)) ⇒
  tf_spec R1 C1 R2 C2 Vin Vout s
```

The above theorem is proven by using the functions and theorems of the formalized Laplace transform and multivariable calculus theories. This concludes the formal verification of the transfer function of the second-order Sallen-Key low-pass filter. In a similar

Table 2 Formalized Laplace transform of commonly used analog signals.

Analog Function	Formalized Form
Exponential Growth with Real Argument	$(\mathcal{L}e^{\alpha t})(s) = \frac{1}{s - \alpha}$, $Re(s) > \alpha$ and $\alpha \in \mathbb{R}$
Exponential Decay with Real Argument	$(\mathcal{L}e^{-\alpha t})(s) = \frac{1}{s + \alpha}$, $Re(s) > -\alpha$ and $\alpha \in \mathbb{R}$
Exponential Growth with Complex Argument	$(\mathcal{L}e^{\alpha t})(s) = \frac{1}{s - \alpha}$, $Re(s) > Re(\alpha)$ and $\alpha \in \mathbb{C}$
Exponential Decay with Complex Argument	$(\mathcal{L}e^{-\alpha t})(s) = \frac{1}{s + \alpha}$, $Re(s) > -Re(\alpha)$ and $\alpha \in \mathbb{C}$
Sine	$(\mathcal{L}sin(\omega t))(s) = \frac{\omega}{s^2 + \omega^2}$, $Re(s) > 0$
Cosine	$(\mathcal{L}cos(\omega t))(s) = \frac{s}{s^2 + \omega^2}$, $Re(s) > 0$
Exponentially Decaying Sine	$(\mathcal{L}e^{-\alpha t}sin(\omega t))(s) = \frac{\omega}{(s+\alpha)^2 + \omega^2}$, $Re(s) > -\alpha$ and $\alpha \in \mathbb{R}$
Exponentially Decaying Cosine	$(\mathcal{L}e^{-\alpha t}cos(\omega t))(s) = \frac{s+\alpha}{(s+\alpha)^2 + \omega^2}$, $Re(s) > -\alpha$ and $\alpha \in \mathbb{R}$

way, we have also verified the transfer function of the first-order low-pass filter and the corresponding proof details can be found in [12].

The usefulness of our proposed formalization is that it greatly facilitates verifying the transfer function of analog circuits using higher-order-logic theorem proving, as the analog circuit designers do not need to go into the subtle details of the Laplace transform mathematics. The foundational Laplace transform and analog circuit library formalization had to be done in an interactive way, due to the undecidable nature of higher-order logic, and took around 5000 lines of HOL-Light code and approximately 800 man-hours. The main challenge in this formalization is the enormous amount of user intervention required due to the undecidable nature of the higher-order logic. Moreover, we had to develop the formal reasoning for the correctness of many proof goals ourselves as detailed proof scripts of these properties are usually not available in mathematical texts. Utilizing this work, the proof script corresponding to the Sallen-Key low-pass filters verification consists of approximately 650 lines of HOL-Light code [12] and the proof process took just a couple of hours by a proficient HOL-Light user, who was quite familiar with the working of the above-mentioned formalization of the Laplace transform. This kind of straightforward verification clearly indicates the effectiveness of our core formalizations of the Laplace transforms and analog components. It is important to note that all of the assumptions have to be explicitly mentioned along with the theorems in order to prove them in HOL-Light. For instance, the positive values of the circuit components and differentiability of the voltages are often ignored in the analog circuit design literature but have been explicitly indicated in our analysis. Similarly, the poles of the given circuit can also be explicitly observed from the formally verified theorem. Moreover, we have been able to capture the continuous models of analog circuits completely in our framework thus eliminating the basic limitation of discretized models in the existing formal verification techniques for analog circuits.

This paper advocates the usage of higher-order-logic theorem proving for verifying the transfer functions of linear analog circuits for SoC devices. Due to the high expressiveness of the underlying logic, we can formally model the structure of the given analog circuit and the differential equation depicting its behavior in its true form. The formalized Laplace transform method can then be used in a theorem prover to deduce the transfer function of the given circuit from this equation. The inherent soundness of theorem proving guarantees correctness of analysis and ensures the availability of all preconditions of the analysis as assumptions of the formally verified theorems. To the best of our knowledge, these features are not shared by any other existing computerized analog circuit verification technique and thus, the proposed approach can be very useful for the analysis of linear analog circuits used in safety-critical domains. Based on this paper, we are able to conduct the transfer function verification of first- and second-order Sallen-Key low-pass filters in a very straightforward way.

OUR FORMALIZATION CAN also be built upon to formalize the inverse Laplace transform function and its associated properties, which can be very useful in analyzing the behavior of analog circuits in the time-domain. Moreover, circuits whose transfer functions have been verified by our proposed technique can be added as formalized components in the formalized Analog Library and then can be used to facilitate the verification of more complex circuits used in the domains of signal processing, wireless communication, controls, and optics. Moreover, in order to reduce the manual verification effort, dedicated simplifiers can be developed for simplifying the proof goals involved in the verification of transfer functions for analog circuits. Our methodology can also be integrated with the traditional verification tools, such as Simulink and SCADE, by classifying the analog components of the given SoC into critical and noncritical components. Noncritical components can be verified by simulation techniques whereas the transfer function of critical parts can be modeled and verified using theorem proving. After the correct reasoning of the critical parts, the complete system can be integrated in the simulation framework to verify the complete system with greater confidence. ■

Acknowledgments

This work was supported by the National Research Program for Universities grant (number 1543) of Higher Education Commission (HEC), Pakistan.

References

[1] O. Hasan and S. Tahar, "Formal verification methods," in *Encyclopedia of Information Science and Technology, IGI Global*, 2015, pp. 7162–7170.

[2] H. Aridhi, M. H. Zaki, and S. Tahar, "Toward improving simulation of analog circuits using model order reduction," in *Proc. Design Automation Test Eur.* 2012, pp. 1337–1342.

[3] H. Taqdees and O. Hasan, "Formalization of Laplace transform using the multivariable calculus theory of HOL-light," in *Logic for Programming Artificial Intelligence and Reasoning (LPAR-19)*, LNCS 8312, 2013, pp. 744–758.

[4] W. Denman, B. Akbarpour, S. Tahar, M. Zaki, and L. C. Paulson, "Formal verification of analog designs using MetiTarski," in *Formal Methods in Computer Aided Design*, 2009, pp. 93–100.

[5] R. Narayanan, B. Akbarpour, M. H. Zaki, S. Tahar, and L. C. Paulson, "Formal verification of analog circuits in the presence of noise and process variation," in *Proc. Design Automation Test Eur.* 2010, pp. 1309–1312.

[6] S. K. Tiwary, A. Gupta, J. R. Phillips, C. Pinello, and R. Zlatanovici, "First steps toward SAT-based formal analog verication," in *Proc. Int. Conf. Comp.-Aided Design,* 2009, pp. 1–8.

[7] S. Steinhorst and L. Hedrich, "Model checking of analog systems using an analog specification language," in *Proc. Design Automation Test Eur.* 2008. pp. 324–329.

[8] A. P. Godse and U. A. Bakshi, *Analog Integrated Circuits—Design and Applications*. Technical Publications, 2009.

[9] M. Althoff, A. Rajhans, B. H. Krogh, S. Yaldiz, X. Li, and L. T. Pileggi, "Formal verification of phase-locked loops using reachability analysis and continuization," in *Proc. Int. Conf. Comp.-Aided Design*, 2011, pp. 659–666.

[10] A. Souari, A. Gawanmeh, S. Tahar, and M. L. Ammari, "Design and verification of a frequency domain equalizer," *Microelectronics J.*, vol. 45, pp. 167–178, 2014.

[11] J. Harrison, "The HOL light theory of Euclidean space," *Automated Reasoning*, vol. 50, no. 2, pp. 173–190, 2013.

[12] S. H. Taqdees and O. Hasan, "Formally verifying transfer functions of analog circuits using theorem proving," National University of Sciences and Technology, 2017. [Online]. Available: http://save.seecs.nust.edu.pk/projects/fvtf/fvtf.html

Syeda Hira Taqdees is a PhD scholar in the School of Computer Science and Engineering, University of New South Wales, Australia. She is working with TrustWorthy Systems Group, Data61, CSIRO. Her research interests include formal methods and operating systems. Taqdees has a master's degree in electrical engineering from National University of Sciences & Technology, Pakistan, with specialization in digital system design and digital signal processing. Contact her at hira.taqdees@seecs.nust.edu.pk.

Osman Hasan is an Assistant Professor at the School of Electrical Engineering and Computer Science, National University of Science and Technology (NUST), Islamabad, Pakistan. Hasan has a PhD from Concordia University, Montreal, QC, Canada. He is a Senior Member of IEEE, and a member of the ACM, Association for Automated Reasoning (AAR), and the Pakistan Engineering Council. Contact him at osman.hasan@seecs.nust.edu.pk.

■ Direct questions and comments about this article to Osman Hasan, School of Electrical Engineering and Computer Science, NUST, H12, Islamabad, Pakistan; e-mail: osman.hasan@seecs.nust.edu.pk.