INTEGRATION, the VLSI journal ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Two-stage fuzzy inference system for symbolic simplification of analog circuits

Mohammad Shokouhifar *, Ali Jalali

Department of Electrical Engineering, Shahid Beheshti University G.C., Tehran, Iran

ARTICLE INFO

Article history:
Received 7 June 2016
Received in revised form
19 August 2016
Accepted 22 August 2016

Keywords:
Symbolic analysis
Simplification
Analog integrated circuits
Fuzzy inference systems

ABSTRACT

This paper presents a knowledge-based fuzzy approach to symbolic circuit simplification in an effort to imitate human reasoning and knowledge of circuit designer experts. The fuzzy approach differs from the conventional simplification techniques in that it can efficiently combine different input variables to obtain optimal simplified expressions. Additionally, this method was chosen due to its adjustability and interpretability, as well as its ability to manage very complex symbolic expressions. The proposed algorithm uses fuzzy logic to simplify the symbolic circuit transfer functions in two stages. In the first stage, a fuzzy system is applied to directly eliminate nonessential circuit components, resulting simplified circuit topology which also yields simpler transfer function. In the second stage, another fuzzy system is used to further simplify the symbolic transfer function from the already simplified circuit, such that deeper insight into the circuit behavior can be obtained. Symbolic and numerical results show that the fuzzy approach outperforms the conventional techniques in terms of accuracy, expression complexity, and CPU running time.

1. Introduction

Symbolic circuit analysis is an analytic technique used to generate symbolic expressions for the performance parameters of analog circuits [1–5]. Symbolic analysis has been applied to compute different circuit characteristics in frequency- and time-domain, e.g., sensitivity analysis [6], signal-to-noise ratio (SNR) [7], power-supply rejection ratio (PSRR) [8], common-mode rejection ratio (CMRR) [9], pole/zero analysis [10], performance bound analysis considering process variations [11], noise analysis [12], harmonic distortion analysis [13], and fault modeling [14]. Moreover, symbolic tools can be applied to analyze voltage-mode [15], current-mode [16] and mixed-mode circuits [17].

Unfortunately, fully symbolic analysis suffers from the exponential growth of expression complexity with the circuit size [18]. The problem will be even worse for the circuits containing semiconductor devices (e.g., Diodes, BJTs and MOSFETs) [19]. Therefore, circuit modeling is a challenging problem in symbolic analysis. There are different approaches for the modeling of different kinds of amplifiers. For instance, the behavior of active devices can be modeled via controlled sources or pathological elements (e.g., via nullor equivalents) [20,21]. Moreover, the operation of semiconductor devices can be linearized by small-signal models.

In order to deal with practical analog circuits, either hierarchical or simplification techniques should be applied. Hierarchical decomposition generates symbolic expressions in "sequence-of-expression" form. The hierarchical analysis methods can be categorized into network formulation [22], topological analysis [23] and DDD-based techniques [24]. Generally, the sequence of expressions achieved via hierarchical analysis is difficult to interpret and manipulate. On the other hand, simplification techniques look for the simplified expressions with the minimum number of terms with high accuracy in representing the exact ones. To keep the symbolic expressions useful for interpretation and manipulation, the use of simplification techniques is mandatory. Firstly, symbolic tools like SSPICE [25] ASAP [26], ISAAC [27], SYNAP [28], and SCYMBAL [29] applied the simplification techniques.

In classical simplification algorithms [25–27], polynomials of a symbolic expression are simplified based on the relative magnitudes of terms. In these methods, the correlation between the different polynomials is not taken into account. Moreover, overall generated errors (e.g., pole/zero displacements, magnitude/phase error, etc.) are not considered. Although the conventional classical techniques may perform correctly for some cases, under some circumstances, significant magnitude/phase error or pole/zero displacements may be occurred. In recent years, evolutionary and swarm intelligence algorithms have been used for symbolic simplification [8,30–32]. Although these methods outperform the

* Corresponding author.
E-mail addresses: m_shokouhifar@sbu.ac.ir, shokoohi24@gmail.com (M. Shokouhifar), a_jalali@sbu.ac.ir (A. Jalali).

http://dx.doi.org/10.1016/j.vlsi.2016.08.004
0167-9260/

Please cite this article as: M. Shokouhifar, A. Jalali, Two-stage fuzzy inference system for symbolic simplification of analog circuits, INTEGRATION, the VLSI journal (2016), http://dx.doi.org/10.1016/j.vlsi.2016.08.004

classical criteria in term of accuracy, an iteratively based optimization algorithm is performed on very complex exact symbolic expressions. Therefore, computational complexity and CPU running time of these methods are much more than those of in the classical criteria.

Existing simplification techniques exhibit some drawbacks which limit their usage for practical analog circuits: First, the resultant expressions in classical criteria are expected to be not so compact as it could be, because no innovative techniques is performed. Second, classical criteria cannot guarantee the accuracy of the simplified expressions, as the overall generated errors in transfer function are not under control. Third, in classical criteria, pole/zero displacements are not taken into account, and consequently, significant pole/zero errors may be generated. Fourth, although metaheuristic approaches have relatively good performance, they have very high computational complexity and running time, which limits their usage only for small-size circuits. Fifth, these techniques have a blind random-based optimization procedure. On the other hand, they do not utilize any knowledge of circuit design experts into the simplification algorithm.

This paper proposes the use of fuzzy logic [33], which recently is gaining wider interest in other optimization areas, to simplify the symbolic circuit transfer functions. One of the main advantages in using fuzzy logic in such an application is that it can effectively handle uncertainties inherent in the nature of circuit analysis. Fuzzy logic can model the decision making and knowledge of circuit design experts to provide accurate results from uncertain and vague information. Moreover, the different input variables can be effectively combined to obtain optimal simplified expressions. The main contributions in this paper can be summarized as follows:

- A fuzzy expert system is proposed as an efficient symbolic circuit simplification algorithm.
- To the best of our knowledge, this is the first study utilizing fuzzy logic for the symbolic circuit analysis.
- We use Mamdani fuzzy system in two stages. In the first stage, a fuzzy system is applied to eliminate nonessential circuit components, resulting simplified circuit topology. In the second stage, another fuzzy system is used to generate the final simplified transfer function by selecting really significant terms.
- The fuzzy algorithm outperforms the conventional methods (classical and metaheuristic) in minimizing the number of terms and CPU running time, while achieving high accuracy is guaranteed.

The rest of the paper is organized as follows: In Section 2, classical and metaheuristic simplification criteria are discussed. The proposed fuzzy simplification methodology is presented in Section 3. In Section 4, the fuzzy algorithm is simulated over three analog circuits and compared with the existing techniques. Finally, conclusion remarks can be seen in Section 5.

2. Literature review

By considering the step of the symbolic analysis process at which the simplification procedure is performed, three types of techniques can be distinguished [18]: simplification-before-generation (SBG), simplification-during-generation (SDG), and simplification-after-generation (SAG). The SAG approaches are the most common methods used in symbolic tools. Even if SBG and/or SDG methods are performed, it is necessary to utilize also SAG techniques to ensure achieving the most compact symbolic expressions. In these methods, simplification is applied directly on the symbolic solution, once the circuit has been analyzed.

2.1. Classical simplification techniques

Let us consider the voltage transfer function of the circuit in expanded form as

$$H(s,\mathbf{x}) = \frac{\sum_{a=0}^{A}\left(s^a f_a(\mathbf{x})\right)}{\sum_{b=0}^{B}\left(s^b g_b(\mathbf{x})\right)} = \frac{f_0(\mathbf{x}) + sf_1(\mathbf{x}) + s^2 f_2(\mathbf{x}) + \ldots + s^A f_A(\mathbf{x})}{g_0(\mathbf{x}) + sg_1(\mathbf{x}) + s^2 g_2(\mathbf{x}) + \ldots + s^B g_B(\mathbf{x})}, \quad (1)$$

where each polynomial $f_a(\mathbf{x})$ or $g_b(\mathbf{x})$ is a sum-of-product (SOP) of the symbolic circuit parameters $\mathbf{x}$ as

$$h_k(\mathbf{x}) = \sum_{t=1}^{l_k} h_{kt}(\mathbf{x}) = h_{k1}(\mathbf{x}) + h_{k2}(\mathbf{x}) + \ldots + h_{kl_k}(\mathbf{x}), \quad (2)$$

where t ($t = 1,2, \ldots, l_k$) is term index, k is the polynomial index, and $h_k(\mathbf{x})$ is the k-th polynomial of the transfer function, which has l_k symbolic terms. The term $h_{kt}(\mathbf{x})$ is a product of circuit components $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}^T$, where N is the total number of circuit components represented by symbols.

Since there are $A+1$ polynomials in the numerator and $B+1$ polynomials in the denominator of Eq. (1), there are totally $(A+B+2)$ polynomials within the overall transfer function. Therefore, the total number of symbolic terms can be calculated as

$$L = \sum_{k=1}^{A+B+2} l_k = l_1 + l_2 + \ldots + l_{(A+B+2)}. \quad (3)$$

In SAG methods, simplification is done by heuristically pruning insignificant terms in the each polynomial of the transfer function so that an approximate polynomial, $h_k^{new}(\mathbf{x})$, is found for each polynomial $h_k(\mathbf{x})$, in such a way that the approximate polynomial fits the original one within a user-specified maximum error parameter ε_M. Selecting a proper nominal point for the simplification procedure is of major importance. Finding the sizes and biases of analog integrated circuits has been discussed in [34]. Generally in symbolic analysis, the exact value of symbolic parameters is not known beforehand. However, simplification algorithms use a typical nominal point $\mathbf{x}_0$ to calculate the values of the different $h_{kt}(\mathbf{x})$ at the nominal point, and compare them to that of $h_k(\mathbf{x}_0)$ [4]. The nominal point is typically specified by the user in the circuit netlist.

There are four conventional classical SAG criteria which were implemented in most of symbolic tools. In Criterion 1 [25], any term $h_{kt}(\mathbf{x})$ is eliminated from the polynomial $h_k(\mathbf{x})$, if it fulfils the following condition

$$\left|h_{kt}(\mathbf{x}_0)\right| < \varepsilon \times max\left(\left|h_{k1}(\mathbf{x}_0)\right|, \left|h_{k2}(\mathbf{x}_0)\right|, \ldots, \left|h_{kl_k}(\mathbf{x}_0)\right|\right). \quad (4)$$

The main drawback is the lack of control on the accumulated error for each polynomial. The accumulated magnitudes of the eliminated terms may represent either a small or a large part of the total magnitude of the polynomial. The other three criteria were introduced to overcome this drawback. These criteria require the previous sorting of terms in each polynomial according to their magnitude at the nominal point $\mathbf{x}_0$. In Criterion 2 [26], p smallest magnitude terms can be eliminated from the polynomial $h_k(\mathbf{x})$, where p is the largest integer for which the following condition is satisfied

$$\frac{\left|\sum_{t=1}^{p} h_{kt}(\mathbf{x}_0)\right|}{\left|\sum_{t=1}^{l_k} h_{kt}(\mathbf{x}_0)\right|} < \varepsilon. \quad (5)$$

Mutually canceling terms do not contribute to Eq. (5), however, such terms may become significant when the simplified expression is evaluated at points other than the nominal. One solution is to modify the elimination condition of Eq. (5) as