ELSEVIER

Microelectronics Journal 37 (2006) 792–799

Microelectronics Journal

www.elsevier.com/locate/mejo

Automated synthesis of passive filter circuits including parasitic effects by genetic programming

Shoou-Jinn Chang *, Hao-Sheng Hou, Yan-Kuin Su

Department of Electrical Engineering, Institute of Microelectronics, National Cheng Kung University, No. 1 Ta Hseuh Road, Tainan 70101, Taiwan, ROC

Received 1 July 2005; received in revised form 6 December 2005; accepted 28 December 2005
Available online 9 March 2006

Abstract

In this paper, we propose a genetic programing method to synthesize passive filter circuits including parasitic effects, which are very common in high-frequency application. This approach allows circuit topology and component values to be evolved simultaneously; therefore, novel circuits different from those generated by traditional methods can be explored. Experimental results show the proposed method can effectively generate not only compliant but also efficient solutions of such problems where the traditional approaches fail.

Keywords: Genetic programing; Parasitic effects; Passive filter synthesis

1. Introduction

Traditional passive filter synthesis methods treat circuit components as ideal ones. Unfortunately, this assumption is no longer suitable for high-frequency application because of parasitic effects. For a given specification, a compliant circuit resulting from traditional design methods may fail to satisfy the same specification when the parasitic effects are taken into account. The authors of [1] have proposed a genetic algorithm to synthesize analog circuits including parasitic effects. However, the genetic algorithm approach deals with only the evolution of component values, and the circuit topology should be predetermined by other methods. Some methods have been proposed to deal with the evolution of circuit topology as well as the component values and shown their capability of generating a variety of circuit topologies [2–11]. In our previous work, we proposed a novel tree representation of RLC circuits and a genetic programing (GP) method to synthesize passive filter circuits [12,13]. This approach allows circuit topology as well as component values to be evolved simultaneously. The proposed method is restricted to series–parallel circuit topology. However, this restriction makes it more efficient than those general-topology methods in passive filter synthesis if the target circuits are confined to series–parallel topology, which is commonly used in practice owing to less component value sensitivity and reduced circuit complexity [14]. Furthermore, for a given specification our method can automatically find efficient circuit (in terms of circuit complexity) without predetermining the number of the circuit components required. In this paper, we extend our previous work to take the parasitic effects into consideration. By exploring the possible solution space consisting of various circuit topologies and component values, the GP approach can generate circuits different from those generated by traditional methods. The experimental results show the extended GP method can effectively generate compliant circuits including the parasitic effects, while the traditional methods fail. In addition, the GP-evolved circuits are more efficient than the traditional ones.

The remainder of the paper is organized as follows. Section 2 introduces the tree representation of RLC circuits and a circuit analysis algorithm based on the tree representation. Section 3 introduces the GP method to synthesize passive filter circuits. Section 4 describes how to model the parasitic effects and how to incorporate them into the GP method. Experimental results are given in Section 5, and the conclusion is given in Section 6.

2. Tree representation of RLC circuits

We find that a binary tree structure can be used to represent series–parallel RLC circuits. An RLC circuit and the

* Corresponding author. Tel.: +886 6 2757575x62391; fax: +886 6 2761854.
E-mail address: changsj@mail.ncku.edu.tw (S.-J. Chang).

0026-2692/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mejo.2005.12.012

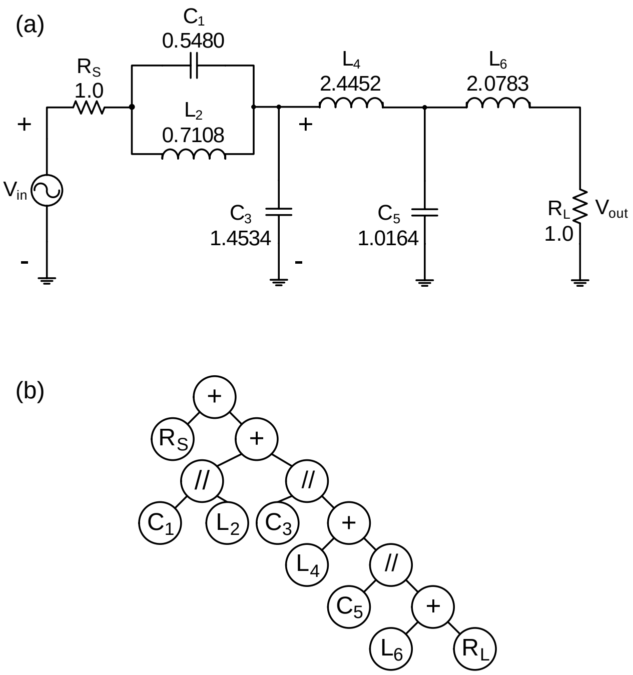

Fig. 1. A GP-evolved low-pass LC filter: (a) schematic and (b) tree representation.

corresponding tree representation are shown in Fig. 1(a) and (b), respectively. Adopting the terminology commonly used in GP, we divide the nodes of the tree into two sets, the terminal set and the function set [15]. The non-terminal nodes belong to the function set including series and parallel nodes, which are denoted by + and //, respectively. The terminal nodes belong to the terminal set consisting of three types of passive electrical components: R (resistor), L (inductor), and C (capacitor). Note that each terminal node can be characterized by the component type and the component value.

We also find that this representation is a suitable data structure for circuit analysis. Assume the goal of circuit analysis is to find the voltage gain (frequency response) of RLC circuits, such as the circuit shown in Fig. 1. The circuit analysis algorithm based on the tree representation is described as follows.

(1) For a given frequency, the impedance of each terminal node is calculated by the given frequency, the component type, and the component value.
(2) Each terminal node begins to return the impedance upward. After receiving the impedances from its children, each non-terminal node performs the corresponding function (series or parallel) on the received impedances to obtain its own impedance, which is then returned upward further.
(3) The source voltage is divided by the impedance of the root node to obtain the current flowing into the tree, and this current flows down through the tree according to the following rules. If the node is series, each current flowing out is equal to the current flowing in. If the node is parallel, the current flowing in is divided inversely proportional to children's impedances and then flows out to the children nodes. If the node is terminal node, the process stops.
(4) After the foregoing steps, the voltage of each node can be obtained by multiplying the current by the impedance. We can subsequently obtain the voltage gain by dividing the voltage of the specified output node by the source voltage.

This circuit analysis algorithm is used to evaluate the fitness of each evolved circuit in the GP process. Note that the above steps 2 and 3 can be implemented by recursive tree traversal algorithms, which are shown as Algorithms 1 and 2, respectively, in Appendix A.

3. Genetic programing

The main idea of GP is based on the evolutionary process observed in nature. To begin with, an initial population composed of a number of solutions (called individuals) is randomly generated. Each individual, in circuit synthesis problems, is a circuit consisting of circuit topology, component type, and component values. New potential solutions are created by genetic operations, such as crossover and mutation. Each individual is evaluated and assigned a scalar value (called fitness) reflecting the performance of the individual. The selection operation is then applied to the population with a bias toward the individuals having higher fitness. The population undergoes the above-mentioned operations iteratively until a stop criterion is met.

The proposed GP used to synthesize resistively terminated passive filters is described below, and the template of the evolved circuits is shown in Fig. 2. Note that the GP is based on the tree representation mentioned above.

Initialization. An initial population with 100 individuals is generated. Each individual is randomly chosen among the four building blocks as shown in Fig. 3, and the L or C node is randomly assigned a component value.

Selection. Tournament selection strategy is adopted here to select two parental trees. According to this strategy, two individuals among the current population are randomly chosen, and the one with higher fitness win the right to mate. The process is repeated to select the other parental tree.

Crossover. For each selected parental tree, a cutting point is randomly chosen with uniform probability, and the subtrees below the cutting points are exchanged to create two

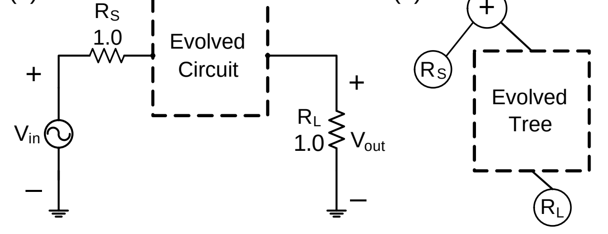

Fig. 2. Template of equally terminated passive filter circuits: (a) schematic and (b) tree representation. The evolved circuit is located between the source and load resistors.