Cosmin Radu Popa

Current-Mode Analog Nonlinear Function Synthesizer Structures

Springer

Current-Mode Analog Nonlinear Function Synthesizer Structures

Cosmin Radu Popa

Current-Mode Analog Nonlinear Function Synthesizer Structures

Springer

Cosmin Radu Popa
Faculty of Electronics, Telecommunications
and Information Technology
University Politehnica of Bucharest
Bucharest
Romania

ISBN 978-3-319-01034-2 ISBN 978-3-319-01035-9 (eBook)
DOI 10.1007/978-3-319-01035-9
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013941505

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

This book is dedicated to
my Grandmother

Preface

Analog signal processing structures represent an important part of analog VLSI designs, a multitude of specific techniques being developed in order to maximize their performances. Comparing with digital structures that respond to the same requirements, the analog approach of signal processing presents the advantages of a real-time operation and of the possibility of their facile integration, especially for portable devices. The development of specific superior-order approximation functions that constitutes the functional cores of the function synthesizer circuits presented in this book increases the flexibility of designs, making them comparable, from this point of view, with latest digital computational structures. The design of analog signal processing computational structures from the perspective of multifunctionality (the utilization of a single functional core for generating a large number of continuous functions) additionally reduces both power consumption and silicon area per implemented function.

The aim of this book is to propose a large diversity of analog function synthesizer structures, based on accurate approximation functions, that are able to implement a multitude of circuit functions, with many applications in analog signal processing: exponential, Gaussian, or hyperbolic functions. Generalizing the methods for obtaining these particular functions, the book analyzes superior-order approximation functions, which represent the core for developing CMOS analog nonlinear function synthesizers.

The method presented for generating continuous functions is based on the utilization of superior-order approximation functions. Comparing with the classical method of designing analog function synthesizer circuits using the limited Taylor series expansion of the generated function, the utilization of high-accuracy approximation functions permits an important reduction of the overall complexity of designed computational circuits. In order to maximize the performance/complexity ratio, some fundamental requirements must be taken into account when developing the general form of an approximation function. First, technical considerations referring to the CMOS implementation of these approximation functions impose a particular structure of primitive mathematical functions that compose them. Because the simplest and most performance basic CMOS computational building blocks are represented by the squaring and multiplying/dividing circuits, the fractional and the squaring functions represent the most

convenient primitive mathematical functions for developing any superior-order approximation function. Second, the requirements for an increased accuracy of approximations impose an independence of the output variable on technological parameters, from this point of view the current-mode operation of the basic building blocks representing the best possible choice. Additionally, the current-mode operation contributes to a reduction of the minimal supply voltage and to an important improving of the computational structures' frequency response. In most cases, the general form of the approximation function is represented by a linear combination of a finite number of primitive functions (usually smaller or equal with two) and some additional terms (linear or polynomial terms). The particular forms of these superior-order approximation functions are developed considering the specific type of the approximated function (odd or even functions).

A general tradeoff that must be taken into account is referring to relation between the computational structures' accuracy and complexity and their capability of generating a multitude of continuous functions. The increasing of the number of computed functions increases also the complexity of the function synthesizers and reduces the possible accuracy that can be obtained using a fixed approximation function. On the other hand, the grouping of approximated functions in some specific classes and the development of analog function synthesizer circuits dedicated only to one specific class maximize the performances of computational structures and decrease the complexity of their implementation.

The first chapter presents possible realizations of exponential function synthesizer circuits, developing third-order and fourth-order approximation functions in order to design accurate computational structures. The circuits are structured considering the order of the approximation functions they are based on and illustrating the block diagrams and the concrete implementations of these structures. Exponential function synthesizer circuits are evaluated from the point of view of their accuracy in analytical and graphical manner, for each particular computational structure being evaluated its output dynamic range. A new general method for increasing the exponential function synthesizers' output dynamic range is presented, the reducing of the effective variation range of the input variable using proper variable changing contributing to fulfill this desiderate.

High-accuracy superior-order approximation functions are developed in the second chapter for generating the Gaussian function. The particular form of the approximation functions presented in this chapter represents a consequence of some important particularities of the Gaussian function (which is even function that allows facile variable changing in order to increase the output dynamic range of computational circuits that generate it). The chapter analyzes a multitude of possible realizations of Gaussian function synthesizer circuits, based on superior-order particular approximation functions: fourth-order, sixth-order, and eighth-order approximation functions. In order to improve the area of operation of developed Gaussian circuits, convenient variable changing are considered for each analyzed approximation function. Analytical and graphical analyses are performed for determining the performances of these superior-order approximation functions.

The analysis, design, and performances' optimization of hyperbolic functions represent the goal of the third chapter. The nonlinear function synthesizer circuits, developed for generating the hyperbolic sinusoidal, co-sinusoidal, and tangent functions use specific superior-order approximation functions. The form of these approximation functions is strongly influenced by the particularities of hyperbolic functions, being correlated with the requirements of minimizing the computational circuits' complexity.

The fourth chapter analyzes the possibility to realize general analog function synthesizer structures, designed for implementing a multitude of circuit functions. Comparing with classical designs in which for any particular function it is designed a specific circuit, the re-utilization of the same functional core for generating a large number of circuit functions presents important advantages. As the most important part of the power is consumed by the functional core and because the largest complexity and design efforts are concentrated for designing this part of the circuit, the utilization of a single computational structure for generating a multitude of circuit functions removes all redundant parts of the circuits and, in consequence, strongly reduces the power consumption and the required complexity per implemented function. The second-order approximation is not able to approximate with enough accuracy a continuous function, resulting the necessity of increasing the order of approximation for improving the analog function synthesizer' performances. In the context of the equilibrium that must be considered between the accuracy of computational circuits and the complexity of their CMOS implementation, the third-order of the approximation represents, for most of continuous functions, a convenient choice. Similar technological restrictions like in previous chapters must be considered for the implementation in CMOS technology of these third-order approximation functions.

For applications that require a more accurate approximation of the circuit functions, the fourth-order approximation is usually enough accurate for generating the most important circuit functions, from the perspective of their applications in analog signal processing and of fulfilling the conditions imposed by these applications. Comparing with previously presented computational structures based on third-order approximation functions, the same requirements must be taken into account when designing a high-precision fourth-order analog function synthesizer circuit.

Contents

Chapter 1
Wide Output Dynamic Range Exponential Function Synthesizers

1.1 Introduction

Exponential functions present a multitude of applications in VLSI designs, such as telecommunication applications, medical equipments, disk drives, hearing aids, neural networks, neural algorithms, neuro-fuzzy and classification applications, pattern recognition, on-chip unsupervised learning, or wavelet transforms. The requirements related to the frequency response of computational structures that implement the exponential function impose important limitations to their CMOS realizations. In this sense, the exponential characteristic of the subthreshold-operated MOS active devices cannot be used, because the extremely low values of drain currents in this region strongly limit the frequency response of the designed circuits. The single choice is represented by the biasing in saturation of MOS transistors that compose the exponential function synthesizers, exploiting the approximately squaring characteristic of the active devices in this region.

A very important requirement of an exponential function synthesizer is related to the output dynamic range that could be obtained for a high-accuracy generating of the exponential function. Additionally, the complexity of CMOS circuit that implements the function synthesizer circuit must be decreased using specific design techniques. In this context, some particularities of developing the computational architectures can be considered. First, the current-mode operation of the function synthesizer usually removes (in a first-order analysis) the dependence of the circuit operation on technology and on temperature-caused errors. This error reduction allows a decreasing of the order of approximation for the designed exponential function synthesizer circuit, contributing to an important decreasing of the overall complexity of the computational structure. Second, the blocks that implement the approximation functions must be chosen from the circuits having the lowest complexity of their implementations in CMOS technology. Exploiting the squaring characteristic of MOS transistors biased in saturation region, the simplest computational structures are represented by the current-mode squaring and multiplier/divider circuits (MDs).

C. R. Popa, *Current-Mode Analog Nonlinear Function Synthesizer Structures*,
DOI: 10.1007/978-3-319-01035-9_1,

Taking into account the previous general considerations, accurate exponential function synthesizer circuits could be developed, considering a trade-off between the accuracy and the silicon area of the computational structure, for a given CMOS technology. Using specific design techniques, the developed computational structures will present very wide output dynamic ranges comparing with classical designs.

There will be analyzed in this chapter a multitude of possible realizations of exponential function synthesizer circuits, based on third-order and fourth-order approximation functions. The presented computational circuits will be structures following the order of the approximation function they are based on, illustrating the block diagrams and the concrete implementations of these structures. The accuracies of exponential function synthesizer circuits are evaluated both in analytical and graphical manner, determining, for each particular computational structure, the output dynamic range.

1.2 General Considerations About Exponential Function Synthesizers

Third-order and fourth-order general approximation functions that are presented in this chapter will be particularized for implementing the important class of exponential functions. Additionally, in order to improve the output dynamic ranges of developed exponential function synthesizer circuits, specific variable changing will be presented. The increasing of the output dynamic ranges of the computational structures using this method exploits the possibility of reduction in the equivalent variation range for the input variable. This is equivalent to the biasing of exponential function synthesizer circuits in the neighborhood of their static operating point, this fact having as consequence the increasing of their overall accuracy and, thus, of their output dynamic range.

1.2.1 General Variable Changing

The particular variable changing that can be used for increasing the output dynamic range of exponential function synthesizer circuits is correlated with the practical considerations imposed by their CMOS implementation. Taking into account these restrictions, the general variable changing can be expressed as follows:

$$x \to \frac{x}{\alpha}, \tag{1.1}$$

α parameter being imposed by a trade-off between the output dynamic range of the exponential computational circuit and the complexity of its CMOS

implementation. The following mathematical identity will represent the functional basis for increasing the output dynamic range of the function synthesizer:

$$\exp(x) = \left[\exp\left(\frac{x}{\alpha}\right)\right]^{\alpha}. \tag{1.2}$$

The value of α parameter is also given by a trade-off between the circuit complexity and its output dynamic range. Additionally, the technological context imposes other restrictions for the value of this parameter. Most concrete, its value will be correlated with the possibility of a facile implementation in CMOS technology of x^{α} function. Because the most convenient realization of a CMOS computational structure using MOS active devices biased in saturation region is represented by the squaring function and, with a reasonable additionally complexity increasing, by the x^4 functions, the practical usual variable changing can be considered to be $\alpha = 2$ and $\alpha = 4$. So, the variable changing will be $x \rightarrow x/2$ and $x \rightarrow x/4$, the functional relation (1.2) becoming:

$$\exp(x) = \left[\exp\left(\frac{x}{2}\right)\right]^{2}, \tag{1.3}$$

or

$$\exp(x) = \left[\exp\left(\frac{x}{4}\right)\right]^{4}. \tag{1.4}$$

On theoretical level, it is possible to choose others variable changing (for example, $x \rightarrow x/6$), but they will require a relatively large complexity of function synthesizers they will be based on. In conclusion, it is preferable to maintain the previous variable changing and to increase the accuracy of the approximation functions that implement the $\exp(x/2)$ or $\exp(x/4)$ functions by increasing their form or order of approximation.

1.2.2 Choosing the Order of Approximation for the Fundamental Approximation Function

Design of high-accuracy exponential function synthesizer circuits with wide output dynamic range is based on two important things. The first one is represented by the previously analyzed variable changing. The second one is based on an accurate approximation of the fundamental function that generates particular forms of the exponential function ($\exp(x/2)$ or $\exp(x/4)$ functions, respectively). In order to implement these mathematical functions, high-accuracy approximation functions must be developed. The problems that appear are related to the order of approximation that is convenient to be considered for these approximation functions in order to fulfill all the accuracy requirements, but with consuming minimal hardware resources for implementing the exponential function synthesizers.

The most important goals of designing exponential function synthesizers are related to theirs accuracies and with their output dynamic ranges. The output dynamic range is correlated with the maximal range of the x input variable and with the maximal accuracy that is accepted for the designed synthesizer circuit. For example, a maximal range for x variable between 0 and 6 is equivalent to a maximal output dynamic range of the exponential function synthesizer of $20\,\lg[\exp(8)]\text{dB} \cong 52$ dB. An increasing of the x range to 0–8 domain increases the output dynamic range to approximately 70 dB, while an additional increasing of the x variable range to 0–10 domain allows to obtain a maximal dynamic range of the synthesizer circuit of approximately 87 dB. In conclusion, in correlation with the maximal output dynamic range that is imposed to the exponential function synthesizer circuit and, also, with the minimal accuracy that is accepted for this circuit, the minimal order for the approximation function can be determined. For the exponential function approximation, second-order, third-order, and fourth-order approximations for the fundamental exponential function are usually enough for most of the practical applications. These reasonable values of the order of approximations are consequences of the variable changing the developed exponential function synthesizers are based on.

A continuous $f(x)$ function can be expanded in Taylor series using the following polynomial function:

$$f(x) = m + \mathrm{n}x + \mathrm{p}x^2 + \mathrm{q}x^3 + \mathrm{r}x^4 + \mathrm{s}x^5 + \mathrm{t}x^6 + \ldots, \tag{1.5}$$

m, n, p, q, r, s, and t being constant parameters of the expansion, imposed by the particular function $f(x)$. The exponential function can be expand in Taylor series as follows:

$$\exp(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \ldots. \tag{1.6}$$

An nth order approximation of the exponential function is equivalent to the identity of the first $n + 1$ terms of Taylor series expansions of exponential function and approximation function.

1.2.3 CMOS Implementations of Exponential Function Synthesizers' Fundamental Blocks

Considering the requirements of facile implementation in CMOS technology and of improved accuracy, current-mode squaring and MDs represent the most convenient realizations of fundamental blocks for developing performance exponential function synthesizers, with minimal hardware resources consumption.

1.2.3.1 CMOS Implementation of the Current-Mode Squaring Circuit

In order to reduce the complexity of the exponential function synthesizers and to increase their overall accuracy, the current-mode squaring circuit (SQ) presented in Fig. 1.1 will be used:

The functional relation that characterizes the operation of the SQ from Fig. 1.1 is

$$I_G = \frac{I_F}{4} - \frac{I_E}{2} + \frac{I_E^2}{4I_F}. \tag{1.7}$$

The symbolical representation of the SQ from Fig. 1.1 is presented in Fig. 1.2.

1.2.3.2 CMOS Implementation of the Current-Mode Multiplier/Divider Circuit

A possible realization of the current-mode MD, required by the implementation of exponential synthesizers, is presented in Fig. 1.3 [1], while its symbolical representation is shown in Fig. 1.4.

The I and I' currents from Fig. 1.3 can be expressed as follows:

$$I = \frac{I_C}{4} - \frac{I_A - I_B}{2} + \frac{(I_A - I_B)^2}{4I_C} \tag{1.8}$$

and

$$I' = \frac{I_C}{4} - \frac{I_A + I_B}{2} + \frac{(I_A + I_B)^2}{4I_C} \tag{1.9}$$

resulting

$$I_D = I' - I = -I_B + \frac{I_A I_B}{I_C} \tag{1.10}$$

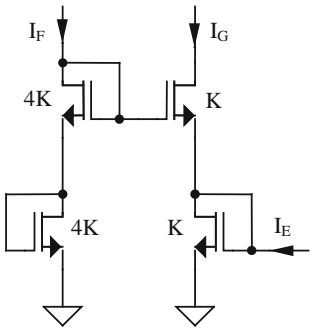

Fig. 1.1 SQ implementation

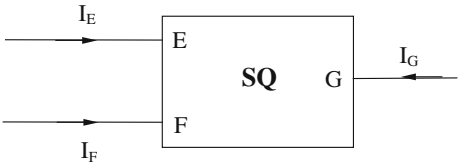

Fig. 1.2 Symbolical representation of the SQ

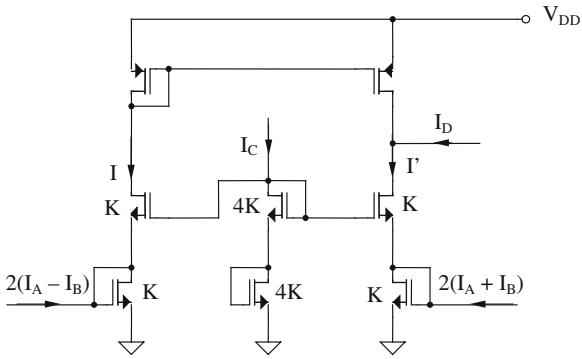

Fig. 1.3 MD implementation [1]

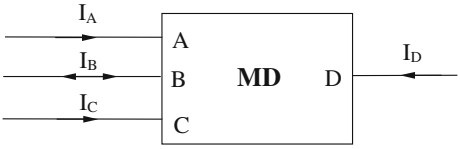

Fig. 1.4 Symbolical representation of the MD

1.3 Wide Output Dynamic Range Exponential Function Synthesizers Based on Second-Order Approximation Functions

1.3.1 First Class of Wide Output Dynamic Range Exponential Function Synthesizers with Second-Order Approximation

1.3.1.1 Approximation Function Without Variable Changing

Considering the previous design restrictions referring to the minimization of the exponential function synthesizer complexity and to the increasing of its overall accuracy, the first general form of the second-order approximation function can be expressed as follows:

$$g_{1a}(x) = \frac{b}{1 + \text{ax}} + c \tag{1.11}$$

a, b, and c being constant coefficients. Their particular values can be determined imposing the condition that the second-order Taylor series of $g_{1a}(x)$ function to be identical with the third-order series of $f(x) = \exp(x)$ approximated function. There are three coefficients of the general expression of the approximation function because it is necessary to have an identity between the first three terms of $f(x) = \exp(x)$ and $g_{1a}(x)$ functions (required by the second-order approximation). This condition imposes the following particular form of $g_{1a}(x)$ function:

$$g_{1a}(x) = \frac{n^2}{p}\frac{1}{1 - \frac{p}{n}x} + \left(m - \frac{n^2}{p}\right) \tag{1.12}$$

Replacing the m, n, and p coefficients from the Taylor series expansion associated with the exponential function, it results

$$g_{1a}(x) = \frac{4}{2 - x} - 1. \tag{1.13}$$

For small values of x variable, the approximation error of $f(x) = \exp(x)$ function using $g_{1a}(x)$ function can be approximated with the third-order error:

$$\varepsilon_{f(x)}^{g_{1a}(x)} \cong \frac{x^3}{12\exp(x)}. \tag{1.14}$$

If the x input variable is increasing, more terms having an order greater or equal with three must be considered. A comparison between $g_{1a}(x)$ approximation function and $f(x) = \exp(x)$ function is presented in Table 1.1. The graphical representations of $g_{1a}(x)$ and $f(x) = \exp(x)$ functions are shown in Fig. 1.5, while the graphical representation of the approximation error, $\varepsilon(x)$, defined as the difference between $g_{1a}(x)$ and $f(x) = \exp(x)$ functions, is presented in Fig. 1.6.

The block diagram of the exponential function synthesizer circuit based on $g_{1a}(x)$ approximation function is presented in Fig. 1.7.

Table 1.1 Comparison between $g_{1a}(x)$ approximation function and $f(x) = \exp(x)$ function

x	$f(x)$ (dB)	$g_{1a}(x)$ (dB)	ε (dB)
−1.4	−12.17	−15.07	−2.90
−1.2	−10.43	−12.04	−1.61
−0.8	−6.95	−7.36	−0.41
−0.4	−3.48	−3.52	−0.04
0.0	0.00	0.00	0.00
0.4	3.48	3.52	0.04
0.8	6.95	7.36	0.41
1.2	10.43	12.04	1.61
1.4	12.17	15.07	2.90

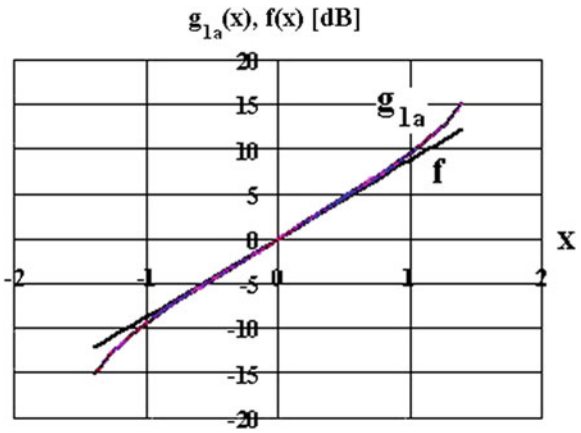

Fig. 1.5 Graphical representation of $g_{1a}(x)$ and $f(x) = \exp(x)$ functions

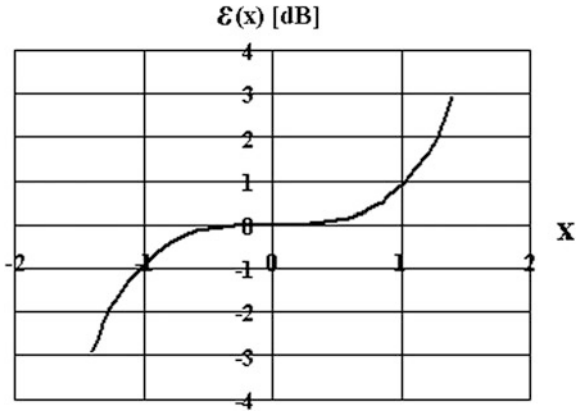

Fig. 1.6 Graphical representation of the approximation error, $\varepsilon(x)$

The expression of I_G current is

$$I_G = \frac{1}{4}(2I_O - I_{\mathrm{IN}}) - 2I_O + \frac{16I_O^2}{4(2I_O - I_{\mathrm{IN}})}, \tag{1.15}$$

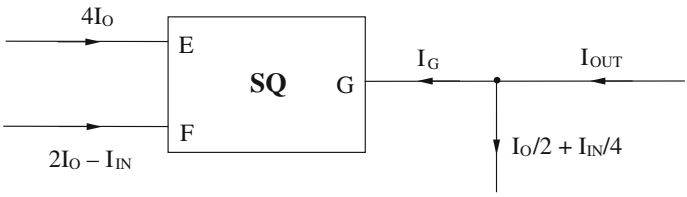

Fig. 1.7 Block diagram of the exponential function synthesizer circuit based on $g_{1a}(x)$ approximation function

resulting the following expression of I_{OUT} current:

$$I_{\mathrm{OUT}} = I_G + \frac{1}{4} I_{\mathrm{IN}} + \frac{1}{2} I_O = \frac{16 I_O^2}{4(2I_O - I_{\mathrm{IN}})} - I_O, \tag{1.16}$$

equivalent to

$$I_{\mathrm{OUT}} = I_O \left[\frac{4}{2 - \left(\frac{I_{\mathrm{IN}}}{I_O} \right)} - 1 \right] \tag{1.17}$$

So, I_{OUT} current approximates the exponential function using $g_{1a}(x)$ approximation function:

$$I_{\mathrm{OUT}} = I_O g_{1a}\left(\frac{I_{\mathrm{IN}}}{I_O} \right) \cong I_O \exp\left(\frac{I_{\mathrm{IN}}}{I_O} \right). \tag{1.18}$$

The CMOS implementation of the exponential function synthesizer circuit based on $g_{1a}(x)$ approximation function is presented in Fig. 1.8 [1].

1.3.1.2 Approximation Function with First Variable Changing

In order to increase the maximal output dynamic range of the exponential function synthesizer based on $g_{1a}(x)$ approximation function, one of the previous presented variable changing can be used. The first variable changing that can be implemented in CMOS technology with a reasonable increasing of the circuit complexity is $x \to x/2$, resulting the following improved approximation function:

$$g_{1b}(x) = \left(\frac{8}{4 - x} - 1 \right)^2. \tag{1.19}$$

A comparison between $g_{1b}(x)$ approximation function and $f(x) = \exp(x)$ function is presented in Table 1.2. The graphical representations of $g_{1b}(x)$ and

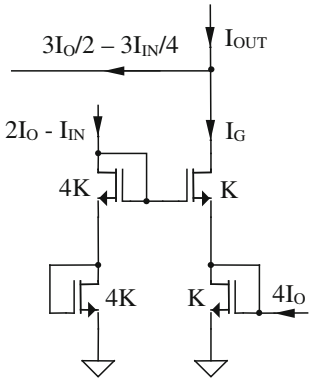

Fig. 1.8 CMOS implementation of the exponential function synthesizer circuit based on $g_{1a}(x)$ approximation function [1]