Abstract

This paper presents a method towards automatic structural synthesis of analog multiplier based on a hierarchical topology “super-topology”, which is abstracted from the most standard four-quadrant multipliers. The essential components in the super-topology are four identical cells, which consist of several MOS-transistors and determine features and performances of multipliers. We build all possible cells within 3 transistors. Experimental results present three new multiplier structures with simulation results to show the creativity of our method.

1. Introduction

A multiplier is an important analog basic circuit and can be used as a computational building block, as a down conversion mixer or as a modulator in mobile RF communication systems [1] [2]. A four-quadrant multiplier is favorite and popular because of the good linearity and the low process variation sensitivity.

EDA tools can be used to help the designer with circuit design. However EDA tools for analog circuits are still not mature compared to those for digital circuits [3]. By now a lot of papers present different structural synthesis methods using knowledge-based approaches [4], full exploration techniques [5] or genetic algorithm [6]. Most of them are applied in generating circuits with linear functions e.g. op amps and LNAs.

In this paper a new design method to automatically synthesize multiplier structures is presented. It uses the special symmetric hierarchical structure of four-quadrant multiplier and abstracts this into a super-topology with four identical cells. In this method a systematic exploration of the whole design space built via the super-topology will be done.

Figure 1. Super-topology with cells

2. Structural Synthesis

2.1. Super-topology

Considering the common properties of the symmetric structures in standard four-quadrant multipliers, an abstract symmetric hierarchical topology named super-topology can be figured out. It is shown in Fig. 1(a).

Before describing and studying the super-topology, we need to define its basic component named cell (Fig. 1(b)). The features of the cell are briefly stated here:

- A cell has four ports: \(x\) – port, \(y\) – port, high – port and low – port. \(x\) and \(y\) port are input ports for signal \(\pm x\) and \(\pm y\). The high-port is treated as output, and the low-port connects to ground.
- A cell is built with several transistors. A cell can be treated as “black box” in the super-topology in the hierarchical view, which is denoted by a double-lined box in Fig. 1(a) and Fig. 1(b).
A cell with its transistor structure can realize a nonlinear function and yield an output $z = f(x, y)$.

The output currents of cells are connected in pairs and flow through the resistors to build a differential voltage output in the super-topology. According to the connection of four cells we can get the output function of the super-topology:

$$U_{out} = R[(z(x, y) + z(-x, -y)) - (z(-x, y) + z(x, -y))] \quad (1)$$

where $z$ is the cell function.

A typical multiplier output can be represented generally in the form of $U_{out} = Kxy + e$, with $K \in \mathbb{R}$ and $e$ for linearity error. From (1), we can derive that the error $e$ is dependent on the cell function $z$, hence the performances and the quality (linearity) of a four-quadrant multiplier are determined also by $z$.

Generally, the well-known functions, such as simple multiple or square law mentioned in [1], are supported in super-topology. Furthermore, many other nonlinear functions (e.g. exp, sqrt) can be used for synthesis, which can also build a form of $Kxy$ with help of approximation. In the following consideration we treat the cell function $z$ more generally as the Taylor series:

$$z = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} c_{m,n} x^m y^n \quad (2)$$

where $c_{m,n}$ are coefficients. Expressing $x^m y^n$ from (2) in odd and even terms, we have

$$z = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} (c_{1,m,n} x^{2m} y^{2n} + c_{2,m,n} x^{2m} y^{2n+1} + \cdots)$$

$$+ \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} (c_{3,m,n} x^{2m+1} y^{2n} + c_{4,m,n} x^{2m+1} y^{2n+1} + \cdots) \quad (3)$$

For term $x^{2m} y^{2n}$ in (3) we can calculate one part of the output function of (1),

$$U_{1,\text{out}} = R \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} (c_{1,m,n} (x^{2m} y^{2n} + (-x)^{2m} (-y)^{2n} - ((-x)^{2m} y^{2n} + x^{2m} (-y)^{2n})) = 0 \quad (4)$$

Similarly, we have $U_{2,\text{out}} = U_{3,\text{out}} = 0$ for terms $x^{2m+1} y^{2n}$ and $x^{2m} y^{2n+1}$. And for term $x^{2m+1} y^{2n+1}$ we have,

$$U_{4,\text{out}} = R \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} (c_{4,m,n} (x^{2m+1} y^{2n+1} + (-x)^{2m+1} (-y)^{2n+1} - ((-x)^{2m+1} y^{2n+1} + x^{2m+1} (-y)^{2n+1}))) = 4R \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} c_{4,m,n} x^{2m+1} y^{2n+1} \quad (5)$$

Thus, the total output of super-topology is equal to

$$U_{\text{out}} = \sum_{i=1}^{4} U_{i,\text{out}} = 4R \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} c_{4,m,n} x^{2m+1} y^{2n+1} = 4R(c_{4,0} xy + c_{4,1} x^3 y + c_{4,2} x^2 y^3 + \cdots) \quad (6)$$

Of particular interest to the cancellation property is the odd order of $x$ and $y$ products in (6). As the input power increases, the higher-order nonlinearities in (6) can generally not be neglected. Despite the above shortcoming, the super-topology can successfully discard the terms with constant offsets and even order of $x$ or $y$ to apply a good suppression of nonlinearity. And moreover, a circuit can be a multiplier, if the cell function $z$ contains the term $xy$ in Taylor series.

### 2.2. Cell generator

To explore the whole design space of cells with 1-3 transistors a cell-generator will be used to synthesize all possible potential proper structures. In this generator, some knowledge-based rules are included in two categories: general rules and restrictive rules. Furthermore a feasibility check is used for cell selection. In the following, the design space of all possible transistor structures is called structure space. Besides, the design space about the sizing, the operating points, the biasing and input ranges is called sizing space.

#### 2.2.1 General rules

We use a simple model of MOS transistor with three terminals and omit the connection of bulk terminal to ground for NMOS or to $V_{DD}$ for PMOS. We have defined a set of general rules, which are constructive and define the initial structure space.

- All combinations of 1-3 transistors are listed here: N, P, NN, NP, PP, NNN, NNP, NPP and PPP with N for NMOS and P for PMOS.
- MOS terminals can be connected to external ports of a cell.
- Two or more transistor terminals can build an internal port.
- Transistor terminals can be connected to a bias voltage, $V_{DD}$ or ground.

#### 2.2.2 Restrictive rules

The general rules create numerous circuit structures. However, many of them are useless because of senseless connection of transistor terminals. The rules defined in this category restrict the structure space to possible usable circuits.

- The external ports of a cell (x-port, y-port, high-port) should be connected to at least one of transistor terminal (G, S or D) separately.
- Drain and source of the same transistor cannot be connected together. Gate and source of the same transistor can not either.
- Only gate terminals can be connected to a bias voltage.
- Only source of PMOS and drain of NMOS can be connected to $V_{DD}$. 

• High-port should not be connected only with gates of transistors.
• Low-port (ground) can not be connected to gate of NMOS. The same is true for gate of PMOS and $V_{DD}$.
• Drain of NMOS and source of PMOS can not be connected to ground.
• It is not allowed, that one transistor does not have any connection to other transistors or only has a connection to other transistors at the reference ports (low-port, $V_{DD}$, $V_{bias}$) and inputs.

2.2.3 Feasibility check

Here we present a primitive feasibility check including an input range check for cells. The transistors in a cell can work in different regions. The conditions for each biasing variety can build a feasible region, which is the white space depicted in Fig. 2. If there is no feasible region for all varieties in the sizing space of a cell, it means that this cell cannot be biased to work correctly. Another benefit of feasibility check is that the input range can be obtained easily. Input $x$ and $y$ build a square in the feasible region and the maximal expanded square presents the input range, if $x$ and $y$ have the same input ranges. The area of the square can be used to check the quality of multipliers, and moreover, the center point of the square can be used as a good choice of primitive biasing.

2.2.4 Linearity/gain check

The measurements of linearity and gain through DC-simulation with the initial bias conditions obtained from feasibility check can be used to roughly determine the circuit type and its performances. If a circuit has a small linearity error and a reasonable gain, we can draw the conclusion that this circuit operates like a multiplier. Thus a further step of topology selection is done.

3. Experimental Result

In this section, two new analog multipliers are shown in order to illustrate and verify the proposed approach. All transistors use a 180nm CMOS technology and have the same size.

3.1. Results with one transistor

The multiplier with one NMOS as cell is well known. The first example considered here is a new multiplier with a PMOS as cell. We got 216 possibilities from general rules, and only two of them fulfill the restrictive rules and feasibility check. Both of them are almost identical only with interchanged inputs ($x$- and $y$-ports). We have chosen one of them (Fig. 3(a)) and simulated with two possible biasing variations (in linear or saturation region).

According to the DC transfer characteristics done in the linearity/gain check this multiplier has a better linearity and a larger gain in the linear region than in the saturation region, namely 0.58% error and 0.53V$^{-1}$ gain with the input range of $\pm 0.25V$ for $x, y$. Its DC result is shown in Fig. 3(b).

3.2. Results with two transistors

We have synthesized multipliers with a cell, which contains one NMOS and one PMOS (NP). The cell generator created 117649 circuits with general rules and then reduced the number to 364 using restrictive rules. Afterwards the feasibility check was done for further selection and we got a total of 232 circuits. Finally the linearity/gain check was applied. The whole algorithm ran in less than 2h. There are total 28 circuits with 38 biasing variations, which have a absolute gain better than 0.5V$^{-1}$ and a linearity error smaller than 2%.

We show here one circuit with lowest error rate. Its topology is shown in Fig. 4(a). The NMOS and PMOS work in the linear region. The DC response curve of this circuit is shown in Fig. 4(b).
3.3. Results with three transistors

Another result is a multiplier with cells, and each cell contains two NMOS and one PMOS (NNP). The cell generator created more than 88M circuits with general rules and then reduced the number to 845 using restrictive rules. Afterwards the feasibility check was done for further selection and we got a total of 390 circuits. Finally the linearity/gain check was applied. The synthesis process cost totally less than 7h. The most interesting thing is that the circuit with smallest error from all circuits shown in Fig. 5 has the same gain and error rate as the circuit shown in Fig. 4(a).

3.4. Comparison

Three new circuits compared with two common multipliers in terms of performance are summarized in Table 1. The structures of two standard multipliers are illustrated in Fig. 6, which have manual optimized biasing conditions. According to the Table 1 the new multipliers have better linearity than the standard circuits.

4. Conclusions

We have presented a new scheme for structural synthesis of analog 4-quadrant multipliers using a new defined super-topology in hierarchical view. It enables the designer to search the structure space and generates a rich variety of circuits. We have demonstrated three new multipliers with rarely used PMOS. They have better performances than the common known circuits. For future improvement, an automatic sizing will help to optimize the performances of generated multipliers.

References