Automatic Generation of Minimal Mismatch Layouts for Capacitors Based on Simulated Annealing[1]

Di Long, Sheqing Dong, Xianlong Hong

EDA Lab, Institute of Software, Tsinghua University, Beijing 100084, China

Abstract

Device matching is of utmost importance to high performance analog and RF circuit. Traditionally, common centroid layouts are regarded as the best layout method to reduce mismatch of analog devices with strict matching constraints. Based on simple integral model used to evaluate the characteristic parameter of devices, the above conclusion is correct. But based on more accurate segmented integral model, it is wrong. Our algorithm is based on the segmented integral model and adopts simulated annealing algorithm to obtain minimal mismatch layout for capacitors. The algorithm has been implemented in "C++". Many test cases have been run. Experimental results have demonstrated the effectiveness of the algorithm.

Keywords: common centroid layout; mismatch; simulated annealing; analog layout automation; module generation

1. Introduction

Nowadays, since the functionality of most analog circuits is based on relative characteristic parameters of analog devices rather than absolute value of those, mismatch puts a fundamental limit on the achievable circuit performance in a particular technology process. For example, current mirrors and differential pairs carrying accurate bias currents and differential signals must have good match. Performance of the circuits like *A/D, D/A* converters[1] and filters greatly depends on the match of capacitance of different capacitors. However, the generation of minimal mismatch layouts for analog devices is a time-consuming task manually.

Few trails can be found in the literature on automatic generation of minimal mismatch layouts for analog devices. In [2], the algorithm assigns unit-capacitors in a compact array for arbitrary ratios, but mismatch is not considered at all. In [3], common centroid placement, symmetrical routing and parasitic balance are considered through a special optimization algorithm, but it is only restricted to device pairs. In the most recent paper [4], algorithm can generate good matching layout for capacitors with arbitrary capacitor ratio, but the algorithm is greatly based on experiential knowledge and heuristic information. So it cannot guarantee the layout has the minimal mismatch. In fact, the ratio in [4] must satisfy some condition to make the algorithm work and the situation that there are empty locations in matrix is not considered at all.

In this paper, the algorithm adopts the optimization method based on simulated annealing and *Segmented Integral Model* to estimate mismatch. So, theoretically it can obtain the minimal mismatch layout. Actually, the Experimental results demonstrate our algorithm can obtain layout with much less mismatch than that in [4]. This paper is organized as follows. In section 2, Mismatch measure and models are presented. Outline and details of the whole algorithm are described in section 3. At last, some test cases and the conclusions conclude the paper.

2. Mismatch Measurement and Models

Mismatch between two components with strict matching constraints is usually expressed as a deviation of the measured device ratio from the intended device ratio. It is can be formulized in formula (1), where p_1 and p_2 are intended values; p_1' and p_2' are measured values. Formula (1) can been generalized to formula (2), which is used to measure the mismatch between several components.

$$\delta = \left| \frac{p_2'/p_1' - p_2/p_1}{p_2/p_1} \right| \tag{1}$$

$$\delta = \max_{i,j} \left| \frac{p_i'/p_j' - p_i/p_j}{p_i/p_j} \right| \tag{2}$$

For analog device of parallel plate capacitor, Formula (3) is the classical formula to calculate capacitance, where ε, h, W, L stand for the dielectric coefficient, thickness, width, length of dielectric.

$$C = \frac{\varepsilon WL}{h} \tag{3}$$

[1] This work is supported by "National Nature Science Foundation of China: 90307005" and "NSFC: 60473126" and "Hi-Tech Research & Development (863) Program of China 2004AA1Z1050".

Definition 1: ***process bias*** is the difference between the drawing geometrical dimensions and its actual measured geometrical dimensions.

Gradient in thickness of dielectric is the most general and important type of *process bias*, so we assume the mismatch is dominated by it and regard ε, *h, W* as constants. Because the thickness of dielectric is different at different location in die, according to the formula (3), it makes the actual measured capacitance different from the intended capacitance. So we must establish model to estimate the actual measured capacitance. There are two major models, which are *Simple Integral Model*[5] and *Segmented Integral Model*[6]. In fig.1, we assume C_1 and C_3 are two fingers of capacitor C_a; C_2 and C_4 are two fingers of capacitor C_b. All the fingers have a linear dielectric gradient α in the direction of angle θ. We assume the thickness of dielectric satisfy the functional relationship in formula (4). For the same layout in fig.1, these two models obtain different mismatch.

$$H(x,y) = h_0 + (\alpha x \cos\theta + \alpha y \sin\theta) \tag{4}$$

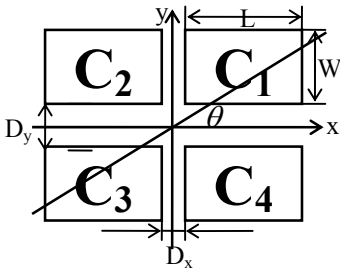

Fig. 1 Illustration of mismatch measure

- ***Simple Integral Model***

$$h_n = \frac{\iint_{DeviceArea_n} H(x,y)dxdy}{DeviceArea_n}; C_n = \frac{\varepsilon W_n L_n}{h_n} \tag{5}$$

Formula (5) is the *Simple Integral Model* used to calculate the capacitance of C_n. Using formula (5) to fig.1, we obtain the thickness of capacitor C_a is h_0 and that of capacitor C_b is also h_0. So the capacitance mismatch between C_a and C_b are zero.

- ***Segmented Integral Model***

$$h_{nm} = \frac{\iint_{FingerArea_{nm}} H(x,y)dxdy}{FingerArea_{nm}}; C_n = \sum_{m=1}^{k} \frac{\varepsilon W_{nm} L_{nm}}{h_{nm}} \tag{6}$$

Formula (6) is the *Segmented Integral Model* used to calculate the capacitance. Using formula (6) to fig.1, we obtain the capacitance of C_a, C_b in formula (11). The capacitance of C_a is not the same with that of C_b. This more accurate model is adopted in our algorithm.

$$h_1 = h_0 + \alpha\left[(D_x + L)\cos\theta + (D_y + W)\sin\theta\right]/2 \tag{7}$$

$$h_2 = h_0 + \alpha\left[-(D_x + L)\cos\theta + (D_y + W)\sin\theta\right]/2 \tag{8}$$

$$h_3 = h_0 + \alpha\left[-(D_x + L)\cos\theta - (D_y + W)\sin\theta\right]/2 \tag{9}$$

$$h_4 = h_0 + \alpha\left[(D_x + L)\cos\theta - (D_y + W)\sin\theta\right]/2 \tag{10}$$

$$C_a = \varepsilon WL(1/h_1 + 1/h_3); C_b = \varepsilon WL(1/h_2 + 1/h_4) \tag{11}$$

3. Algorithm

In this section, the proposed algorithm for generating minimal mismatch layouts for capacitors is presented. The algorithm searches the optimal solution in a countable and big solution space, so the universal optimization algorithm of simulated annealing is adopted. Inputs of the algorithm are listed as follows:

- Capacitance vector $\langle C_1, C_2, ..., C_n \rangle$ recording each capacitance value of capacitor;
- Column number C_{num} and row number R_{num} of final layout matrix;
- Design rules.

Outputs of the algorithm are minimal mismatch layout for capacitors. The whole algorithm is presented in Fig. 2. Each sub-step will be described below in details.

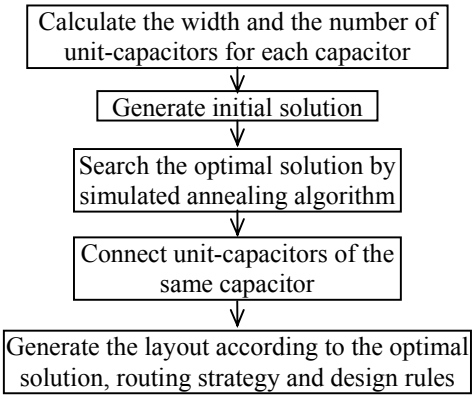

Fig. 2 flow chart of the whole algorithm

3.1. Calculate the width and the number of unit-capacitors for each capacitor

We define vector $\langle N_1, N_2, ..., N_n \rangle$ as *unit-capacitor number vector*, where N_i is the number of unit-capacitors of C_i, and C_0 as capacitance of unit-capacitor, which is assumed to be square. Theoretically, all of the C_i can be positive real number, but in practical layout, the capacitance value has an up limit of precision (L_c). So all of the C_i must be multiple of L_c and we assume all of the C_i and C_0 are positive integers. If some N_i is not an integer, we can connect $[N_i]$ intact unit-capacitors and a N_i-$[N_i]$ unit-capacitor to construct C_i and $(N_i\text{-}[N_i]) \times C_0$ must be greater than L_c. Formula (12) defines the feasible set of C_0. If the $C_{num} \times R_{num}$ is greater than the sum of all the $\lfloor N_i \rfloor$, we must place another more dummy unit-capacitors at empty locations. Each element in Formula (15) defines a matrix with minimal dummies. Each element in Formula (16) defines a matrix with minimal non-intact unit-capacitors.

$$F_c=\left\{C\middle|\sum_{i=1}^{n}\lfloor C_i/C\rfloor\le C_{num}\times R_{num},\lceil C_i/C\rceil\ne C_i/C\Rightarrow(C_i/C-\lceil C_i/C\rceil)\times C\ge L_c\right\} \quad (12)$$

$$f(C)=C_{num}\times R_{num}-\sum_{i=1}^{n}\lfloor C_i/C\rfloor \quad (13)$$

$$g(C)=\sum_{i=1}^{n}\varphi(C_i/C),\varphi(x)=\begin{Bmatrix}0,\lceil x\rceil=x\\1,\lceil x\rceil\ne x\end{Bmatrix} \quad (14)$$

$$\Gamma_c=\left\{C\middle|f(C)=\inf\left(f(F_c)\right)\right\} \quad (15)$$

$$\Psi_c=\left\{C\middle|g(C)=\inf\left(g(\Gamma_c)\right)\right\} \quad (16)$$

Because greater value of C_0 corresponds to layout with greater mismatch, C_0 equals $\inf(\Psi_c)$. Then the width of unit-capacitor (W_{uc}) can be calculated by formula (3); $\langle N_1,N_2,...,N_n\rangle$ equals $\langle C_1,C_2,...,C_n\rangle/C_0$.

3.2. Generate initial solution

After obtaining the vector $\langle N_1,N_2,...,N_n\rangle$, we can naturally generate a matrix as initial solution, where we place the unit-capacitors from left to right and then from top to bottom according to the sequence of capacitor's number. In Fig. 3, the character "u" stands for a non-intact unit-capacitor and the character "o" stands for an intact unit-capacitor. Of cause, we can adopt the heuristic method in [4] to generate the initial solution, but it does not benefit to obtain better solution by the process of simulated annealing.

0o	0u	1o	1o	1o
1o	1o	1u	2o	2o
2o	2o	2o	2o	2o
3o	3o	3o	3o	3o
3o	3o	4o	4o	4o
4o	4o	4o	4o	4o

Fig.3 An example of the initial matrix

3.3. Search the optimal solution by simulated annealing algorithm

The most important things to design simulated annealing algorithm are the definition of cost function and the strategy of new solution generation.

- Cost function

Assuming the center coordinate of ***jth*** finger (unit-capacitor) of ***ith*** capacitor is (x_{ij},y_{ij}), the formula (17) can be used to calculate the capacitance of C_i.

$$C_i(\theta)=\varepsilon W_{uc}^2\left[\frac{N_i-\lceil N_i\rceil}{h_0+\alpha(x_{ij}\cos\theta+y_{ij}\sin\theta)}+\sum_{j=1}^{\lceil N_i\rceil}\frac{1}{h_0+\alpha(x_{ij}\cos\theta+y_{ij}\sin\theta)}\right] \quad (17)$$

Definition 2: *mismatch cost function* is defined in formula (18).

$$f_{mismatch}(\theta)=\max_{i,j}\left|\frac{C_i(\theta)/C_j(\theta)-C_i/C_j}{C_i/C_j}\right| \quad (18)$$

It is very difficult to accurately calculate the maximum of function (18) by the mathematical analysis method. Considering the function (18) is a continuous function, we can use numerical calculation method to estimate the maximum. Dividing the interval of $[0,\pi]$ into m sub-intervals with the same length, we obtain a point set $\Phi=\{\pi i/m\,|\,i=0,1,...,m\}$ and calculate values of the function (18) at these points. At last we select the maximum in the set of $F_{mismatch}=f_{mismatch}(\Phi)$ as the cost of this matrix. Experiments demonstrate effectivity of the method.

- Generation of new solutions

The algorithm adopts three methods to generate new solutions:

1. exchange two unit-capacitors

If the unit-capacitors to be exchanged are the same, the algorithm will continue to select another two different unit-capacitors. If two unit-capacitors are the fingers of the same capacitor, but one is intact and the other is non-intact, they are regarded as different ones.

2. exchange two rows of unit-capacitors
3. exchange two columns of unit-capacitors

Because the second and the third exchanges have great disturbance to original solution, the algorithm makes the probability to adopt these two exchanges much less than the first one.

3.4. Connect unit-capacitors of the same capacitor

That only the layouts of unit-capacitors have the minimum mismatch is not enough. To obtain better matches the algorithm must realize symmetry routing.

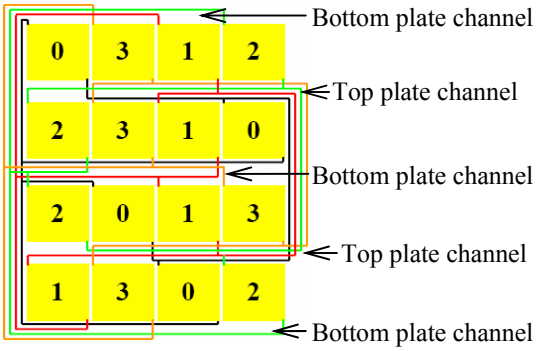

Fig. 4 Symmetry routing

Routing channels are chosen either horizontal or vertical. For parallel plate capacitor, there are two plates. In a matrix with m rows, there are $m+1$ horizontal routing channels and two vertical routing channels. From bottom to top, the horizontal routing channels are Ch_1, Ch_2,..., Ch_{m+1}. Channels with odd serial number are used to connect bottom plates, channels with even serial number for top plates. Vertical channel on the right of matrix is used to connect the top plates in different rows and the left is used to connect the bottom plates in different rows. In the Fig. 4, we only use lines with different colors to stand for the connections of different capacitors. In

actual layout, we use metal-6 as top plate and metal-5 as bottom plate. In horizontal bottom plate channels, metal-5 is used as vertical routing and metal-6 is used as horizontal routing. In horizontal top plate channels, metal-6 is used as vertical routing and metal-5 is used as horizontal routing. Metal-5 is routed in the left vertical channel and metal-6 is routed in the right one.

4. Experiment results

The whole algorithm has been implemented in "C++" on the platform of Solaris 5.2 on Sun-V880. In the following test cases, L_c is $10^{-3}\,pf$, h_0 is $40\times10^{-9}m$, ε is $3.33\times10^{-11}\,f/m$. In test case-1, Capacitance vector is <4,4,4,4>pf; the numbers of columns and rows are both 4 and the runtime is 156 seconds. In test case-2, Capacitance vector is <1.2,5.8,7,7,8>pf; the numbers of columns and rows are 5 and 6 and the runtime is 406 seconds. In test case-3, Capacitances vector is <1,1.4,2,9.2,17>pf; the numbers of columns and rows are 4 and 8 and the runtime is 967 seconds. The experimental results demonstrate our algorithm can obtain layout with much less mismatch than that in [4].

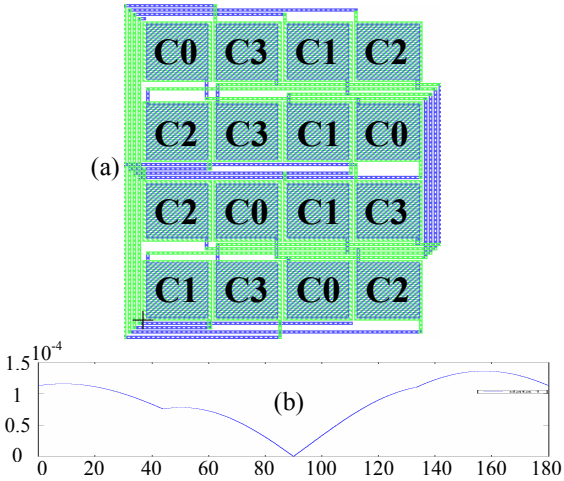

Fig. 5 Test case-1 (a) The minimal mismatch layout (b) Plot of mismatch with gradient angle

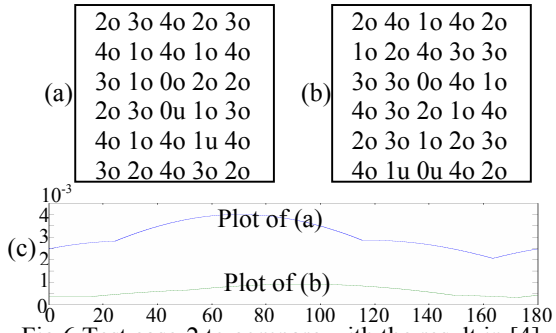

Fig.6 Test case-2 to compare with the result in [4]
(a) Layout matrix in [4]
(b) Layout matrix obtained by our algorithm
(c) Plot of mismatch with gradient angle in (a), (b)

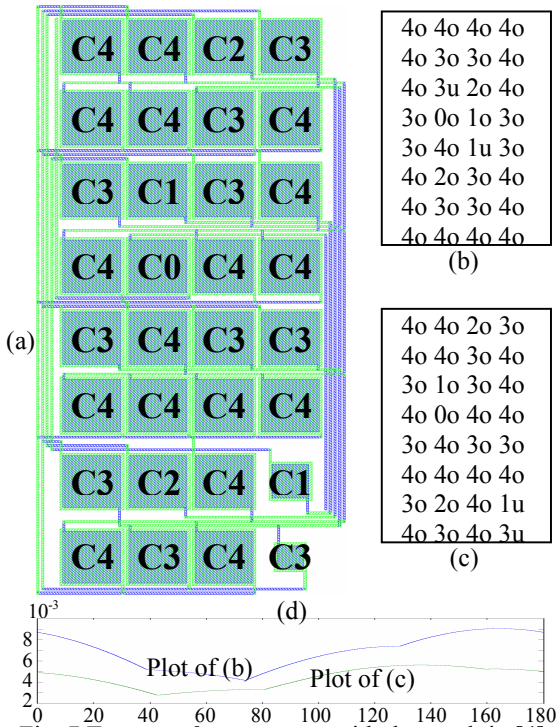

Fig. 7 Test case-3 to compare with the result in [4]
(a) The minimal mismatch layout
(b) Layout matrix in [4]
(c) Layout matrix obtained by our algorithm
(d) Plot of mismatch with gradient angle in (b), (c)

5. References

[1] J. L. McCreary and P. R. Gray. "All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques", Part I. IEEE J. of Solid-State Circuits, 10(6):371-379, Dec. 1975.

[2] H. Yaghutiel, A. Sangiovanni-Vincentelli and P. R. Gray. "A Methodology for Automated Layout of Switched-Capacitor Filters". In Proc. ICCAD, pages 444-447, 1986.

[3] J. D. Bruce, H. W. Li, M. J. Dallabetta and R. J. Baker. "Analog Layout using ALAS", IEEE J. of Solid-State Circuits, 31(2): 271-274, Feb. 1996.

[4] DiaaEldin Sayed and Mohamed Dessouky, "Automatic Generation of Common-Centroid Capacitor Arrays with Arbitrary Capacitor Ratio", in Proc. IEEE DATE, 2002.

[5] M. Pelgrom, A. Duimnaijer and A. Welbers. "Matching Properties of MOS transistors". IEEE J. of Solid-State Circuits,24(5):1433-1440, Oct. 1989.

[6] M.-F. Lan, A. Tammineedi and R. Geiger. "Current Mirror Layout Strategies for Enhancing Matching Performance". Analog Integrated Circuits and Signal Processing, Kluwer Academic Publishers, 28:9—26, Jul. 2001.