

Signal-Path Driven Partition and Placement for Analog Circuit

Di Long, Xianlong Hong, Sheqin Dong

EDA Lab, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
longd02@mails.tsinghua.edu.cn; hxl-dcs@mail.tsinghua.edu.cn; dongsq@mail.tsinghua.edu.cn

Abstract—This paper advances a new methodology based on signal-path information to resolve the problem of device-level placement for analog layout. This methodology is mainly based on three observations: thinking of hierarchical design for analog, structural feature of circuit based on signal-path, requirements of matching/symmetry constraint and the reduction of parasitics. The thinking of hierarchical design makes the whole analog circuit divided into core-circuit and bias-circuit. So, the algorithm is designed as two independent steps: core-circuit is placed firstly, and then bias-circuit. The structural feature of circuit based on signal-path and the requirement of matching/symmetry constraint decide the placement pattern of core-circuit. The reduction of parasitics requires the algorithm to select the optimal variants to realize the placement. Experimental results demonstrate that this algorithm can generate the compact layout with high performance and it is universal and effective.¹

Keywords: signal-path, analog placement, layout automation, circuit partition, symmetry constrain, device merging

I Introduction

Nowadays, SOC integrates all of the circuits on one chip, including digital and analog parts. The physical design of digital circuits is automated to a large extent but the layout of analog circuits is still a manual, time-consuming and error-prone, which makes custom analog layout be a bottleneck in the mixed-signal design flow. The fast changes of demands in ASIC market also require analog layout automation tool to accelerate the whole design process. Thus, the time from product demands to market can be greatly shortened. Analog placement is a vital step in the design flow from circuit schematics to layout. Analog placement automation tool must not only provide a good rectangle packing functionality, but also satisfy analog specifications. Mismatch and parasitics induced by the layout are the most important factors to affect the performance of analog circuit.

In the past two decades, researches have done many researches about analog placement, but no successful commercial products have existed. Generally speaking, the existing methodologies for analog placement automation can be classified into the following several categories.

- The constructive placement techniques, which adopts the increase placement thinking that one module is selected at a time and positioned in the best available location, which is calculated by an evaluation function or directed by an expert system. A placement tool based on expert knowledge is developed in [1]. These methods are very fast, and scales

well with the problem size, but the final placement can be poor because there are no effective methods to decide the order of modules and the view of global optimization and the costs of supporting expert system are expensive.

- A placement technique iteratively combining min-cut partitioning and force-directed placement (FDP) has been employed in an interactive environment for full-custom designs [2]. This method can fast obtain a feasible placement solution satisfying geometrical constraints but not ensure that the placement is a good placement.

- Nowadays, the simulated annealing [3] and genetic algorithms [4], which are global and stochastic optimization algorithms, are the most effective engines to solve the analog placement problem. There is no mathematical limit for the solution space of the problem and the cost function. Analog constraints can be easily implemented by adding the extra punishment items in the cost function. ILAC [5], KOAN/ANAGRAM II [6], PUPPY-A [7] and LAYLA [8] all adopt these optimization algorithms. But these methods all adopt absolute coordinate to represent placement and explore an extremely large search space including feasible and unfeasible placement solutions.

- Analog placement method based on topological representations, which still adopts the optimization engines of *SA* or *GA*. Topological representations are used to describe the placement, such as BSG [9], SP [10], CBL [11]. Solutions in search space expressed by topological representations are all feasible, so this solution space is much smaller than that represented by absolute coordinate. In [12], the analog constraint of matching/symmetry is realized by the topological representation, which further reduces the solution space. But, from the analysis of solution space in [9], [10], [11] and [12], we can find these solution spaces are still very big and [12] only considers the symmetry constraint.

This paper advances a novel methodology of signal-path driven partition and placement for analog circuit, which can automatic generate a compact placement with high performance from a circuit schematics based on signal-path information. The methodology is mainly based on three observations: thinking of hierarchical design for analog circuit, structural feature of circuit based on signal-path, requirements of matching/symmetry constraint and the reduction of parasitics. The above heuristic information about circuit make solution space greatly decreased.

This paper is organized as follows. Section II will give some necessary definitions and the algorithm of signal-path driven circuit partition for analog circuit. Section III will analyze foundation why this methodology can direct placement of core-circuit and then present the algorithm of core-circuit placement in details. Section IV will present the

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".

algorithm of bias-circuit placement. In the last section, we will illustrate some layouts generated by the algorithm according to the industrial circuit schematics.

II. Signal-Path Driven Circuit Partition for Analog

Generally speaking, the design of analog circuit is hierarchical. Firstly, according to the whole specifications of analog circuit, designers divide it into several sub-functional modules, and then design sub-circuit schematics for each sub-functional module. In each sub-circuit, designers separately design core-circuit and bias-circuit. The layout design is also hierarchical. The methodology of hierarchical design and layout is illustrated in Fig.1.

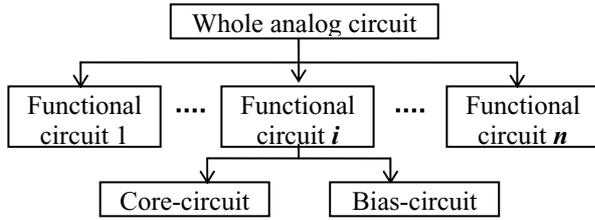


Fig.1 hierarchical design and layout for analog circuit

The hierarchical methodology of manual design and layout for analog circuit indicates that automation algorithm of analog placement should follow the methodology. Firstly each sub-circuit are placed separately and then combined together. For each sub-circuit, firstly core-circuit are placed and then bias-circuit are placed with core-circuit, which are regarded as a black box. Before introducing the methodology of signal-path driven partition and placement for analog circuit, it is necessary to give some definitions.

Definition 1: core-circuit, which is the main circuit in each unit-functional circuit, is responsible for transporting and processing analog signal. There are very strict constraints of matching, symmetry and parasitics minimization on it.

Definition 2: bias-circuit is responsible for providing bias voltages and bias currents for some MOS transistors in *core-circuit*.

Definition 3: power-earth transistor chain is a chain composed of transistors satisfying the following conditions:

- The source/drain of each transistor in the chain must be connected to the power net or the earth net or the source/drain of another transistor in the same chain.
- There is one and only one transistor in the chain, source/drain of which is connected to the power net and there is one and only one transistor in the chain, source/drain of which is connected to the earth net.

In analog circuit, there often exist several *power-earth transistor chains*, each of which is named *pec* and denoted by $\langle Mi_1, Mi_2, \dots, Mi_n \rangle$. There is an example for *pecs* in Fig.2, where different color lines stand for different *pecs*. In Fig.2 there are total six *pecs*, which are $pec_1 = \langle M5, M1, M3 \rangle$, $pec_2 = \langle M5, M2, M4 \rangle$, $pec_3 = \langle M8, M6, M10, M3 \rangle$, $pec_4 = \langle M9, M7, M11, M4 \rangle$, $pec_5 = \langle M12, M14 \rangle$ and $pec_6 = \langle M13, M15 \rangle$. It is also possible that the same transistor belongs to different *pecs*. In Fig. 2, the

transistor *M5* belongs to pec_1 and pec_2 at the same time and the transistor *M3* belongs to pec_1 and pec_3 at the same time.

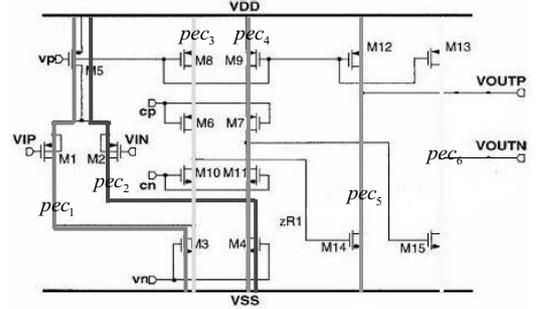


Fig.2 Schematics of full-differential Miller-compensated two-stage amplifier and the diagram of all *pecs*

Definition 4: signal-path is a special type of *pec*, the generation algorithm of which is listed as follows:

• Define the set of signal input nets as I and the set of all *pecs* is Γ_{pec} and transistor $Mj = \{g_j, s_j, d_j\}$ is regarded as a set of its gate net, source net and drain net.

• Define the set of initial input transistors as $\Phi = \{Mj \mid Mj \cap I \neq \emptyset\}$, so the initial set of signal-paths can be defined as $\Psi_0 = \{pec_i \mid pec_i \cap \Phi \neq \emptyset\}$;

• This step is a loop procedure: $\Psi_{n+1} = \Psi_n \cup \Psi_{sd_n} \cup \Psi_{gn}$, where $\Psi_{sd_n} = \{pec_i \mid (pec_i \in \Gamma_{pec} - \Psi_n) \wedge (\exists pec \in \Psi_n) pec_i \cap pec \neq \emptyset\}$ and $\Psi_{gn} = \left\{ pec_i \mid \begin{array}{l} (pec_i \in \Gamma_{pec} - \Psi_n) \wedge (\exists pec \in \Psi_n) \\ (\exists Mj \in pec_i) (\exists Mh \in pec) s_h = g_j \vee d_h = g_j \end{array} \right\}$.

In fact, the current of pec_i in the set Ψ_{sd} is controlled by the source/drain current of pec , which is the direct front-stage of pec_i and the current of pec_i in the set Ψ_g is controlled by the voltages of the source/drain nets of pec , which is also the direct front-stage of pec_i .

• The stop condition of the above loop procedure is $\Psi_{sd_n} = \emptyset \wedge \Psi_{gn} = \emptyset$.

We still take the schematics in Fig. 2 as an example to demonstrate the *signal-paths* generation algorithm.

1. $\Gamma_{pec} = \{pec_1, pec_2, \dots, pec_6\}$ and $I = \{VIP, VIN\}$, so $\Phi = \{M1, M2\}$ and $\Psi_0 = \{pec_1, pec_2\}$;
2. when $n = 0$, $\Psi_{sd_0} = \{pec_3, pec_4\}$ and $\Psi_{g_0} = \emptyset$. So $\Psi_1 = \Psi_0 \cup \Psi_{sd_0} \cup \Psi_{g_0} = \{pec_1, pec_2, pec_3, pec_4\}$;
3. when $n = 1$, $\Psi_{sd_1} = \emptyset$ and $\Psi_{g_1} = \{pec_5, pec_6\}$. So $\Psi_2 = \Psi_1 \cup \Psi_{sd_1} \cup \Psi_{g_1} = \{pec_1, pec_2, pec_3, pec_4, pec_5, pec_6\}$.
4. when $n = 2$, $\Psi_{sd_2} = \emptyset$ and $\Psi_{g_2} = \emptyset$, stop.

Now, we can find $\Gamma_{pec} = \Psi_2$, which explains that the schematics in Fig. 2 only include *core-circuit* because the *bias-circuit* providing bias voltage for nets of *vp*, *cp*, *cn*, *vn* is not drawn in the schematics. Generally speaking, when the generation algorithm stops the set Ψ_n is *core-circuit* and the set $\Gamma_{pec} - \Psi_n$ is *bias-circuit*.

III. Algorithm of Core-Circuit Placement

In the two phases of *core-circuit* placement and *bias-circuit* placement, we must orient all the MOS transistors in the same direction, because MOS transistors that do not lie parallel to one another become vulnerable to stress and tilt induced mobility variations, which can cause several percent variations in their transconductance. In this method, the gate of each MOS transistor is kept vertical.

Before placing the devices on the die, we must know what placement constraints are required on these devices. These constraints can be appointed by customs or distinguished by circuit analysis algorithm. Matching and symmetry are the most important constraints for placement, which can be extracted by the algorithm in [13].

Through the generation of the set Ψ_n , we obtain circuit structure information that the *core-circuit* is composed of *signal-paths* and these *signal-paths* have obvious sequence feature from input to output. So, the following placement pattern can be used to realize the *core-circuit* placement:

- Firstly, realize the inner placement of each *signal-path* in the set Ψ_n . Pay attention to the feature that the MOS transistors in the same *signal-path* all have the sequential drain/source connection relationship and the signal is transported in the form of drain/source current, so minimization of connection capacitor and resistor requires these MOS transistors are placed close to each other. Based on the above consideration, Layout all the MOS transistors in each *signal-path* from left to right according to the connection sequence from the power net to earth net. To obtain a compact layout, the height of these MOS transistor layouts must approximate to each other.

- Realize the whole placement of the *core-circuit* composed of *signal-paths*. There are following three different placement patters according to different situations of symmetry constraints: (here, assume $\Psi_n = \{pec_1, \dots, pec_j\}$, and according to generation algorithm of *signal-paths*, the sets of $\Psi_0, \Psi_{sd_0}, \Psi_{g_0}, \dots, \Psi_{sd_{n-1}}, \Psi_{g_{n-1}}$ are arranged according to generation sequence)

1. If the *core-circuit* is full-symmetrical structure, the integer j must be even number. We assume pec_k is symmetrical with pec_{j-k+1} . So, the set Ψ_n is divided into two sets: $\Psi S_n = \{pec_1, \dots, pec_{j/2}\}$ and $\Psi M_n = \{pec_{1+j/2}, \dots, pec_j\}$. Place all the *signal-paths* in set ΨS_n from bottom to top according to

the sequence of elements in the sets $\Psi_0 - \Psi M_n, \Psi_{sd_0} - \Psi M_n, \Psi_{g_0} - \Psi M_n, \dots, \Psi_{sd_{n-1}} - \Psi M_n, \Psi_{g_{n-1}} - \Psi M_n$. At last, mirror the placement of ΨS_n with the horizontal symmetry axis.

Fig. 6 is an example for this case.

2. If the *core-circuit* is not full-symmetrical structure, the placement of the symmetrical part is realized according to method in item 1 and the placement of the non-symmetrical part is realized according to the placement method for ΨS_n in item 1. Finally, place the non-symmetrical part above the symmetrical part. Fig. 7 is an example for this case.

3. If part but not all of transistors in pec_i have symmetry constraints with others in pec_j , the pec_i and pec_j are grouped together and the transistors without symmetrical counterparts are regarded as self-symmetry. Fig. 7 is an example for this case.

- If $Mi_n \in pec_h$ and $Mi_m \in pec_k$ need to be placed according to common centroid style [15], pec_h and pec_k , must be placed in the most front of the generation sequence. pec_j , in which there are the output nets, must be placed in the most back of the generation sequence.

To make the outline of *core-circuit* layout approximate to a rectangle, the width of all *signal-paths* must approximate to each other.

The thinking of the above placement pattern is mainly based on the following two observations. Firstly, the metals used to connect the inner nets in *signal-path* are to transport the currents, so these metals are required very short. What's more, it is possible that the adjacent transistors can be merged together. Secondly, the metals used to connect the nets among *signal-paths* are to transport the voltage, so these metals are not required very short.

After the pattern of *core-circuit* placement is determined, we will introduce the algorithm to realize the equal height of MOS transistors in the same *signal-path* and the equal width of all the *signal-paths*. Before introducing the algorithm, we must firstly introduce the fundamental element to realize the algorithm. That is the variants of MOS transistors.

The transistor is often divided into several sections called fingers. Different partition methods will generate different variants with different aspect ratio, finger number and parasitic capacitor. Any $Variant_m$ and $Variant_n$ of the same MOS transistor must hold the equation $W_m \times F_m = W_n \times F_n$, where W and F stand for channel width and the finger number respectively. Different layouts for the same transistor are illustrated in Fig.3.

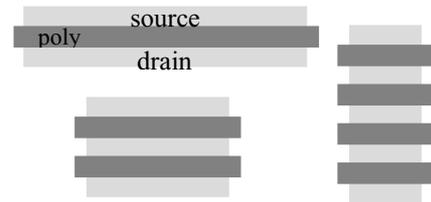


Fig. 3 layouts of varieties of the same MOS transistor

In COMS circuit, the parasitic capacitor of MOS transistor is mainly composed of source/drain bulk capacitance, which can be calculated by formula (1), where C_{jSBt} refers to the total source/drain bulk capacitance, A and P to the source/drain area and perimeter, C_j and C_{jso} to the bottom and sidewall junction capacitances in absence of any junction voltage and ϕ_j to the built-in junction potential. m_j and m_{jso} depend on the doping profile of the junction.

$$C_{jSBt} = \frac{AC_j}{\left(1 - \frac{V_{BS}}{\phi_j}\right)^{m_j}} + \frac{PC_{jso}}{\left(1 - \frac{V_{BS}}{\phi_j}\right)^{m_{jso}}} \quad (1)$$

For the given MOS transistor, channel width W and length L are decided and the only variable is finger number F . For simple we assume $\alpha = \left(1 - \frac{V_{BS}}{\phi_j}\right)^{m_j}$ and $\beta = \left(1 - \frac{V_{BS}}{\phi_j}\right)^{m_{jso}}$, so C_{jSBt} can be expressed as the function of F in formula (2), which can be simplified as formula (3). It is obvious that the formula (3) exits the minimum. The meaning of δ_{gap} is illustrated in Fig. 5.

$$C_{jSBt}(F) = \alpha \frac{W}{F} \times [FL + (F+1)\delta_{gap}] + 2\beta \times \left[\frac{W}{F} + FL + (F+1)\delta_{gap} \right] \quad (2)$$

$$C_{jSBt}(F) = (\alpha WL + \alpha W\delta_{gap} + 2\beta\delta_{gap}) + 2\beta(L + W\delta_{gap})F + W(\alpha\delta_{gap} + 2\beta)\frac{1}{F} \quad (3)$$

For a given group of parameter values, we will obtain a graph of function $C_{jSBt}(F)$ in Fig.4, where the horizontal coordinate represents the finger number and the vertical coordinate represents the source/drain bulk capacitance.

The algorithm of placement for *core-circuit* utilizes the feature that variants of the same MOS transistor have different height and width to realize the requirements of equal height of MOS transistors in *signal-path* and equal width of *signal-paths* in *core-circuit*.

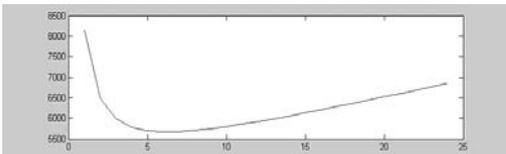


Fig.4 graph of function $C_{jSBt}(n)$

Though the pattern of *core-circuit* placement is determined there are still many different placements because the same MOS transistor has many variants. The *core-circuit* placement algorithm is to realize the following objectives:

- minimizing the differences of the height of all MOS transistor layouts in the same *signal-path*;
- minimizing the differences of the width of all *signal-paths* in the *core-circuit*;
- minimizing the total capacitance parasitics of all the MOS transistors;
- maximizing the area utility.

So, in fact the *core-circuit* placement problem is an optimization problem. We use the simulated annealing

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

For the facility of defining the cost function, we give the following rules: according to placement pattern, we number the MOS transistors in the set pec_i belonging to the set Ψ_n from left to right as $M_{i,1}, M_{i,2}, \dots, M_{i,h(i)}$ and number the *signal-paths* from top to bottom as $pec_1, pec_2, \dots, pec_l$.

The cost function is defined in formula (4), where H_{\max_diff} denotes maximum in all values of the maximum height differences between any two MOS transistors in the same *signal-path*. W_{\max_diff} denotes maximum width difference between any two *signal-paths* in *core-circuit*. P_{cap} denotes the sum of source/drain parasitical capacitance of all MOS transistors. U_{area} denotes the utility of layout area. Customs can tune the coefficients of $\alpha, \beta, \gamma, \delta$ to obtain different *core-circuit* layout as expected.

$$CP_{core} = \alpha H_{\max_diff} + \beta W_{\max_diff} + \gamma P_{cap} + \delta(1 - U_{area}) \quad (4)$$

$$H_{\max_diff} = \max_{pec_i \in \Psi_n} \left[\max_{M_{k,i}, M_{k,j} \in pec_k} \left| height(M_{k,i}) - height(M_{k,j}) \right| \right] \quad (5)$$

$$W_{\max_diff} = \max_{pec_i, pec_j \in \Psi_n} \left| width(pec_i) - width(pec_j) \right| \quad (6)$$

$$P_{cap} = \sum_{i=1}^n \sum_{j=1}^{h(i)} C_{jSBt}(F_{i,j}) \quad (7)$$

$$U_{area} = \frac{\sum_{i=1}^l \sum_{j=1}^{h(i)} height(M_{i,j}) \times width(M_{i,j})}{\max_{pec_i \in \Psi_n} [width(pec_i)] \times \left[\sum_{i=1}^l \max_{M_{i,j} \in pec_i} [height(M_{i,j})] + \sum_{i=1}^{l-1} RSS(pec_i, pec_{i+1}) \right]} \quad (8)$$

$$height(M_{i,j}) = W_{i,j} / F_{i,j} + 2\delta_{head} \quad (9)$$

$$width(pec_i) = \left[\sum_{j=1}^{h(i)} F_{i,j} L_{i,j} + (F_{i,j} + 1)\delta_{gap} \right] + \sum_{j=1}^{h(i)-1} RSM(M_{i,j}, M_{i,j+1}) \quad (10)$$

In the above formulas, δ_{gap} and δ_{head} are the constant parameters decided by design rule. Function $RSM(M_{i,j}, M_{i,j+1})$ denotes the spacing between two MOS transistor layouts and function $RSS(pec_i, pec_{i+1})$ denotes the spacing between two *signal-paths*. These two functions can be assigned two different constants or decided by some interconnection area estimation model. For clarity, the diagram of these parameters is illustrated in Fig. 5. If the transistor $M_{i,j}$ and the transistor $M_{i,j+1}$ can be merged in the horizontal direction, the value of the function $RSM(M_{i,j}, M_{i,j+1})$ is zero.

Firstly, the function $height(\)$ and $width(\)$ in formulas (5), (6), (8) are substituted with formula (9) and (10) respectively. Secondly, the function $C_{jSBt}(\)$ in formula (7)

is substituted with formula (3). Lastly, substitute all the variable items in formula (4) with formulas (5), (6), (7), (8). Thus, we can find that the formula (4) is the function of finger numbers of all MOS transistors. So, the generation of new solution is to change finger number of some MOS transistor. Firstly, the algorithm randomly selects a MOS transistor assumed to be $M_{i,j}$, and then randomly selects an integer from the range $[1, \lceil W_{i,j}/L_{i,j} \rceil]$ as finger number of $M_{i,j}$.

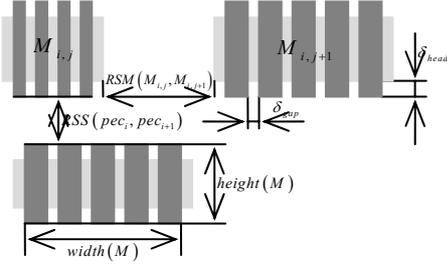


Fig. 5 diagram of parameters in formula (5)

Fig.6 is the *core-circuit* placement of schematics in Fig.2, which is generated by the above *core-circuit* placement pattern and optimization algorithm.

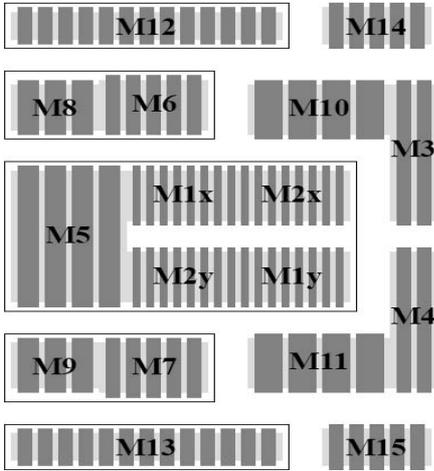


Fig. 6 core-circuit placement of schematics in Fig.2

IV. Placement of Bias-Circuit

After completing the placement of *core-circuit*, we will realize the placement of whole circuit including *core-circuit* and *bias-circuit*, which is called the second phase placement. In this phase of placement, the *core-circuit* layout is regarded as a black box. There is no strict feature of circuit structure in *bias-circuit*, so the placement of *bias-circuit* has no determined pattern and the stochastic placement algorithm is adopted to realize the placement of *bias-circuit*. Of cause, in the phase of *bias-circuit* placement, there are the following objectives to be optimized:

- minimizing the total parasitic capacitance of all the MOS transistors belonging to the *bias-circuit*;
- minimizing the total length of routing metals;
- maximizing the area utility.

$$CP_{bias} = \omega P_{cap} + \theta L_{routing} + \lambda(1 - U_{area}) \quad (11)$$

It is possible there is matching/symmetry constraint forced on some MOS transistors in *bias-circuit*, so the topological representation based on SP [10] is adopted to describe the placement of *bias-circuit*, which is easy to realize symmetry constraint during the process of stochastic placement [12]. The MOS transistors in *bias-circuit* are all oriented in the same direction with those in *core-circuit*. The selection of variants and device merging is also considered in this phase of placement. The selection of variants is realized when new solution is generated and device merging is settled by the algorithm in [14], which is also based on the topological representation of SP. The only difference is only horizontal merging is considered in *bias-circuit* placement because the MOS transistors are oriented in one direction.

V. Experiment and Conclusion

In Fig. 7 (a), $\Gamma_{pec} = \{pec_1, pec_2, \dots, pec_6\}$ and $I = \{inn, inp\}$, through the *signal-paths* generation algorithm, we can obtain $\Psi_n = \{pec_3, pec_4, pec_2, pec_5, pec_6\}$. The elements in the set Ψ_n are arranged by the generation sequence. Elements in the set $\Gamma_{pec} - \Psi_n = \{pec_1\}$ are bias-circuit. The algorithm in [12] obtains pec_3 and pec_4 are full-symmetry pair; pec_2 and pec_5 are partial-symmetry pair; pec_6 is non-symmetry *signal-path*.

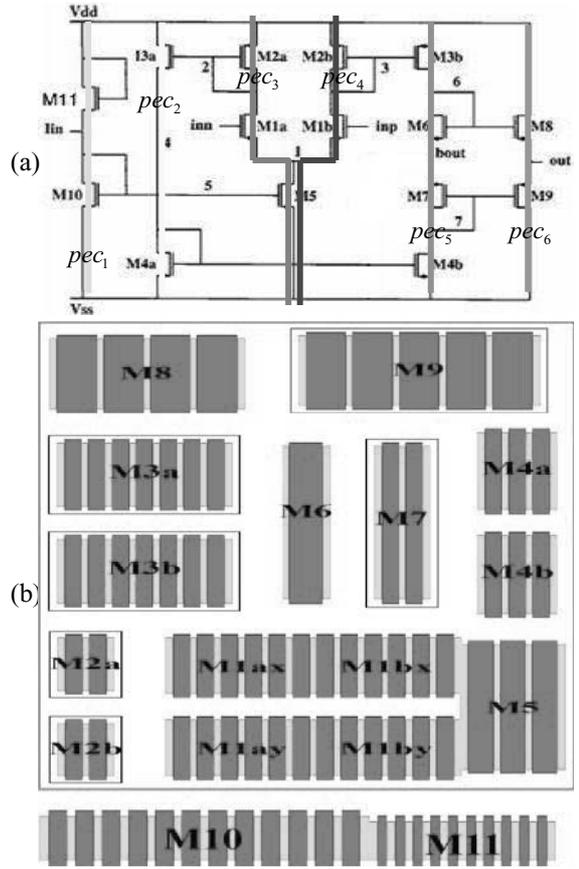


Fig.7 test case-1 (a) Schematics of a high-speed CMOS comparator (b) Placement of the schematics

In Fig. 8, for clarity core-circuit and bias-circuit are drawn separately. When (a) and (b) are regarded as an integer, we obtain $\Gamma_{pec} = \{pec_1, pec_2, \dots, pec_{12}\}$ and $I = \{I_{in+}, I_{in-}\}$. Through the *signal-paths* generation algorithm, we obtain $\Psi_n = \{pec_4, pec_3, pec_6, pec_7, pec_1, pec_2, pec_3, pec_8, pec_9, pec_{10}\}$. The elements in the set Ψ_n are arranged by the generation sequence. After adjusting the elements sequence, we obtain sequence $pec_3, pec_6, pec_4, pec_7, pec_2, pec_3, pec_8, pec_9, pec_{10}, pec_1$. The algorithm in [12] can obtain the full-symmetry group of pairs $(pec_3, pec_6), (pec_4, pec_7), (pec_1, pec_{10}), (pec_8, pec_2), (pec_3, pec_9)$. Elements in set $\Gamma_{pec} - \Psi_n = \{pec_{11}, pec_{12}\}$ are bias-circuit.

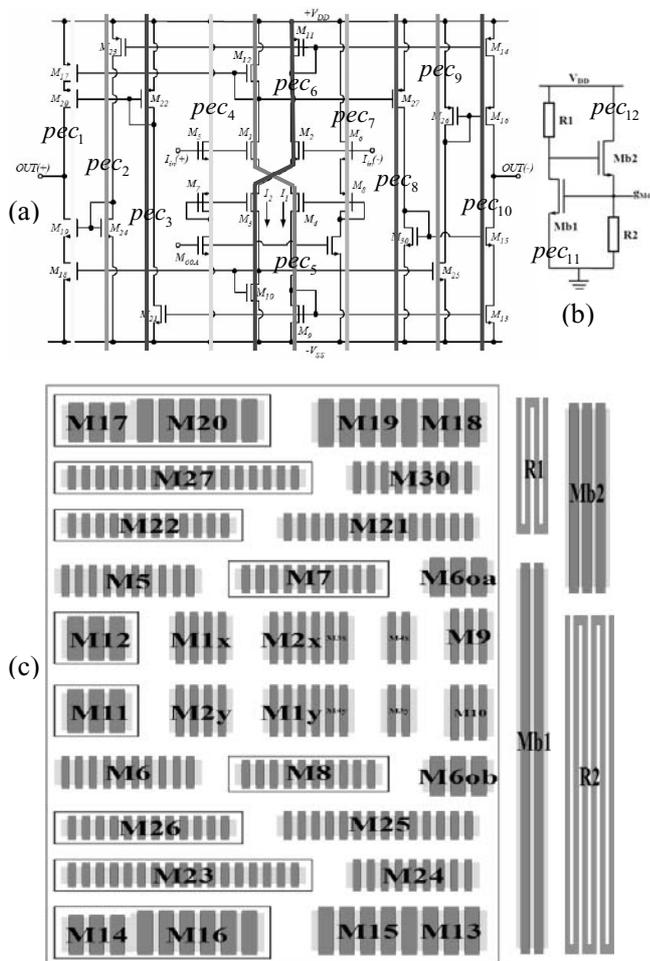


Fig. 8 test case-2 (a) core-circuit of class AB open-loop opamp (b) bias-circuit providing bias-voltage for *M60a* and *M60b* (c) the whole layout of the schematics

In this paper, a new methodology of signal-path driven partition and placement for analog circuit is proposed, which sufficient utilizes the thinking of hierarchical design, structural feature of analog circuit based on signal-path and variants of MOS transistors. Experimental results demonstrate this algorithm can generate compact layout with high performance and it is universal and effective.

References

[1] M. Kayal, S. Piguet, M. Declercq, and B. Hochet, "SALIM: A layout generation tool for analog ICs," in Proc. IEEE Custom Integrated Circuits Conf., 1988, pp. 7.5.1–7.5.4.

[2] E. Malavasi, J. L. Ganley, and E. Charbon, "Quick placement with geometric constraints," in Proc. IEEE Custom Integrated Circuits Conf., 1997, pp. 561–564.

[3] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, no. 4598, pp. 671–680, May 1983.

[4] J. Cohoon and W. Paris, "Genetic placement," IEEE Trans. Computer-Aided Design, vol. CAD-6, pp. 956–964, Nov. 1987.

[5] J. Rijmenants, J. B. Litsios, T. R. Schwarz, and M. Degrauwe, "ILAC: An automated layout tool for analog CMOS circuits," IEEE J. Solid-State Circuits, vol. SC-24, no. 2, pp. 417–425, Apr. 1989.

[6] J. Cohn, D. Garrod, R. Rutenbar, and L. Carley, "KOAN/ANAGRAMII: New tools for device-level analog layout," IEEE J. Solid-State Circuits, vol. 26, pp. 330–342, Mar. 1991.

[7] E. Malavasi, E. Charbon, E. Felt, and A. Sangiovanni-Vincentelli, "Automation of IC layout with analog constraints," IEEE Trans. Computer-Aided Design, vol. 15, pp. 923–942, Aug. 1996.

[8] K. Lampaert, G. Gielen, and W. Sansen, "A performance-driven placement tool for analog integrated circuits," IEEE J. Solid-State Circuits, vol. 30, pp. 773–780, July 1995.

[9] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, "Module packing based on the BSG-structure and IC layout applications," IEEE Trans. Computer-Aided Design, vol. 17, pp. 519–530, June 1998.

[10] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair," IEEE Trans. Computer-Aided Design, vol. 15, pp. 1518–1524, Dec. 1996.

[11] Xianlong Hong, Sheqin Dong, et al, "Corner Block List Representation and Its Application to Floorplan Optimization", IEEE Transactions on Circuits and Systems, II, 51(5):228-233.

[12] Florin Balasa, Koen Lampaert, "Module Placement for Analog Layout Using the Sequence-Pair Representation", in Proc. DAC, pp. 274-279, 1999.

[13] Su Yi, Sheqin Dong, Qingsheng Hao, Xiangqing He, Xianlong Hong, "Automated Analog Circuits Symmetrical Layout Constraint Extraction by Partition", in Proc. ASICON, pp. 166-169, Oct. 2003.

[14] Liu Rui, Dong Sheqin, Hong Xianlong, Long Di, Gu Jun, "Algorithms for analog VLSI 2D stack generation and block merging", Proc. ISCAS, vol. IV, pp. 716-719, 2003.

[15] Di Long, Xianlong Hong, Sheqin Dong, "Optimal Two-Dimension Common Centroid Layout Automation for MOS Transistor Unit-Circuit", Proc. ISCAS, pp 2999-3002, 2005.