A Transistor-level Symmetrical Layout Generation for Analog Device

Bo Yang, Qing Dong, Jing Li, Shigetoshi Nakatake

Department of Information and Media Engineering, The University of Kitakyushu
Hibikino 1-1, Wakamatsu, Kitakyushu, Fukuoka 808-0135, Japan

Abstract— This paper introduces a transistor-level symmetrical layout generation algorithm aiming at maximum diffusion-merging to the current paths for analog circuit. We present a SA-based algorithm to symmetrically assign the transistor pair into two rows and meanwhile minimize the total wirelength and diffusion gaps. Two examples are used to demonstrate the effectiveness of our algorithm.

I. Introduction

Since the mixed-signal SoC technology emerged, the CMOS analog circuits, interfacing the digital signal to our "real" world, are now able to be integrated with the digital circuits on a single chip. The analog signal can be quantized by the interface circuit, and further be processed by taking advantage of the high speed and high precision capability of digital processor. However, analog circuits have been proved more sensitive to the process and thermal induced variation than digital circuits. When they are implemented with the digital circuits, their layouts need be paid more attention to, especially as the process node is scaling down to the nanometer realm. In oder to alleviate the variation, the sensitive transistors are usually need be placed in a symmetrical manner to suppress the mismatch. However automatically synthesis of symmetrical layout is still far from maturity.

A lot of papers have been published to deal with this problem in the past decades[1] [2][3]. Most of the algorithms proposed in these papers employ the placement techniques for the digital circuits as basis, and apply the symmetry constraints to a set of prescribed placement individuals during the optimization. These algorithms hardly considered the individuals' orientations, especially when the placement individuals are transistors, hence introducing unnecessary routing detours.

This paper presents a direct transistor-level symmetrical layout generation algorithm, considering the maximum diffusion merging to reduce the routing complexity. The rest of this paper is organized as follows. Section II gives the preliminaries regarding the diffusion graph extraction. Section III discusses the twin-row symmetrical layout generation. Section IV gives two examples to show the effectiveness of our algorithm. Section V concludes the paper.

II. Preliminaries

Traditionally analog placement problems are studied in terms of blocks, which are geometrically rectangles extracted from the bounding boxes of transistors or functional, and are called cells or modules. The algorithms to deal with placement problems are then built up on these abstract blocks. After the cells been placed, some tools, such as KOAN [4], involve a process to merge the diffusion of transistors to reduce the total area future, but this post-process usually cannot achieve maximum diffusion merging.

This paper presents a transistor-level layout generation method, which employs an Euler trail searching algorithm in a diffusion graph extracted from a given analog circuit to maximize the diffusion merging.

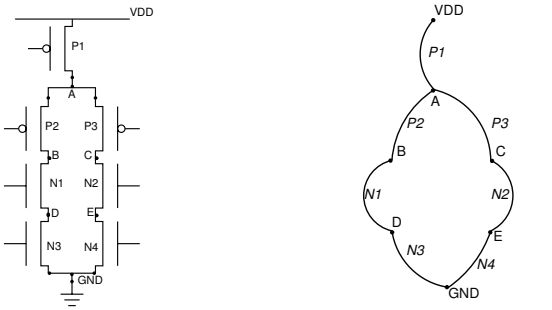
One analog CMOS circuit is usually biased to its quiescent operation point by a large DC signal. In this quiescent point, the bias current are always assumed to be through the channel of a MOS transistor, and the current between the gate and the channel is assumed non-existed. Therefore for the netlists of an analog CMOS circuit, we can prune it by removing the gate nodes and map the rest to a diffusion graph $G = \{E, V\}$, where the edge set $E = \{e_i, ...\}$ is mapped from the transistors $t_i, ...$ in the circuit, and each vertex of V corresponds to a non-empty pruned netlist, i.e., a netlist at least connecting to one transistor's diffusion region. As a matter of fact, since generally the resistor and capacitor can not share diffusion region with transistor, they are all removed as well.

Figure 1 gives an example of diffusion graph mapping. Figure 1(a) shows the pruned netlist, i.e., all the gate-associated netlist segments are removed, and Figure 1(b) is the corresponding diffusion graph.

The algorithm proposed in this paper is to generate a layout for a given diffusion graph aiming at merging the diffusions of adjacent transistors as many as possible. Figure 2 gives an layout example of maximum diffusion merging for the NMOS transistors in Figure 1.

III. Twin transistor-rows Symmetrical layout generation

For the transistors with the match constraint, they usually need to be placed in a symmetrical manner in the layout to suppress the process- and thermal-induced mismatch. The algorithm presented in this section, using a twin- row layout



(a) Netlist with gate-gate and gate-diffusion connection being pruned

(b) Diffusion graph

Fig. 1. Example of diffusion graph construction

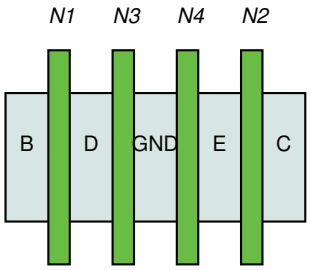


Fig. 2. Layout with diffusion merging

style, by defining a global symmetry axis between the rows, can generate a well-structured symmetrical layout for the transistor pairs with prescribed symmetry constraints, and meanwhile merge the diffusions as many as possible to reduce the routing complexity. We demonstrate the whole procedure by taking an OPAMP as shown in Figure 3 as an example. It should be noted that our algorithm currently can only handle the matching pairs, i.e., the discussion on three or more matched transistors in one constrained group is not covered in this paper.

A. *Symmetrical diffusion-graph construction*

Our algorithm starts with a diffusion graph mapped from a pruned transistor-level netlist by removing all gate-to-gate and gate-to-diffusion connections (and, as explained in the previous section, resistors and capacitors). By searching an Euler trail in the diffusion graph, a diffusion-sharing transistor chain can be extracted if the Euler trail exists in the graph, and two adjacent transistors in the chain will share one diffusion. Therefore it's unnecessary to involve any extra routing tracks for connecting the adjacent transistors in the trail, and the transistors are compactly placed as well (see Figure 1 and 2 for example).

Since the PMOS and NMOS transistors can not be connected directly by the diffusion, so there will be no sharing between the PMOS transistor and NMOS transistor. Therefore the diffusion graph can be partitioned into two sub-graphs: a P-diffusion graph and a N-diffusion graph, and the netlists that connect both P- and N-diffusions will appear as vertices in the both sub-graphs. Figure 3 and 4 show the process by taking an OPAMP circuit as an example. The circuit as shown in Figure 3 is mapped to a diffusion graph (Figure 4(a)) after removing the resistors, capacitors, gate-to-diffusion connections and gate-to-gate connections, and the diffusion graph then is partitioned into p- and N-sub-graph (Figure 4(b)). The netlist $D1$, $D2$ and $VOUT$, due to their connections to both P- and N-diffusions, appear in both sub-graphs.

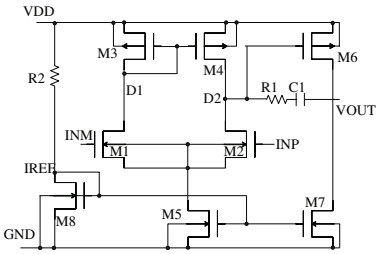
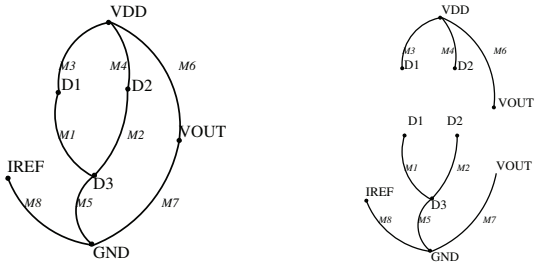


Fig. 3. Circuit of OPAMP

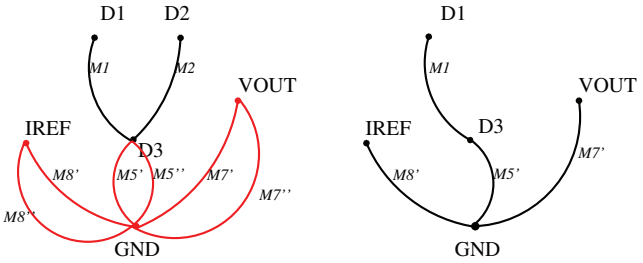


(a)Diffusion graph of OPAMP

(b)P/N-sub-diffusion graph

Fig. 4. Diffusion graph extracted from the OPAMP and its P-/N-sub-diffusion graph

In order to symmetrically assign the transistors, each sub-diffusion graph is further modified as follows. For the transistor without symmetry constraint, the corresponding edge $e_{i,j}$ in the diffusion graph is folded to two parallel edges $e'_{i,j}$ and $e''_{i,j}$, i.e., the transistor without symmetry constraint are decompose into two sub-transistors, represented by the two parallel edges in the modified graph. The two sub-transistors are assumed to be a symmetrical pair then, and will be placed symmetrically as well. The modified diffusion graph G' is called a symmetrical graph. For example, the N-diffusion graph in Figure 4(b) is modified to a symmetrical graph as shown in Figure 5(a). The edge $M5$, $M7$ and $M8$ are replaced by parallel edges ($M5'$, $M5''$), ($M7'$, $M7''$) and ($M8'$, $M8''$), representing 3 new symmetrical transistor-pairs in the circuit. The algorithm proposed in this paper then arranges the transistor-pairs into two rows to obtain a symmetrical layout, and meanwhile pursue the maximum diffusing merging.



(a)Symmetrical graph (b)A sub-graph example for one transistor row

Fig. 5. Symmetrical graph and one of its sub-graph corresponding to one transistor row

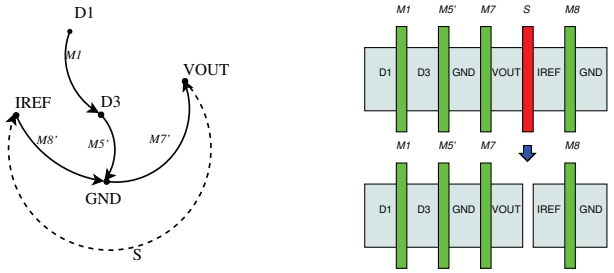
B. *Twin-row symmetrical transistor assignment*

If two transistors in each symmetrical pair are able to be assigned into two transistor rows while keeping their gates alignment, we can obtain a layout with the transistor pairs symmetrical about the mid-line of two rows. However the arbitrary ordering the transistor pairs in the row cannot promise a layout with maximum diffusion-merging, and the total wirelength is not considered either.

To handle these problems we use a simulated annealing (SA) based algorithm to search for a symmetrical layout with maximum diffusion-merging. Our algorithm begins with a random ordering to the transistor pairs. For example, one possible ordering to the transistor pairs in Figure 5(a) can be $\{(M1,M2)$, $(M5',M5'')$, $(M7',M7'')$, $(M8',M8'')\}$. The order of the pairs determines the transistors' arrangement order in both transistor rows. For each pair , the two transistors will be randomly distributed into two groups, G_1 and G_2, and each group corresponds to one transistor row in the layout. This is equivalent to an edge-partitioning to the symmetrical diffusion graph. For example, as shown in Figure 5, a possible partitioning is group G_1 with edge set $E_{G1} = \{M1,M5',M7',M8'\}$ and group G_2 with $E_{G2} = \{M2,M5'',M7'',M8''\}$. We can observe that each group is a sub-graph of the diffusion graph $G = (V,E)$, where the vertex set of sub-graph $V_{G1} = V_{G2} = V$, and the edge set $E_{G1} \vee E_{G2} = E$, $E_{G1} \wedge E_{G2} = \phi$.

Our algorithm firstly accepts a random ordering of the transistor pairs as an initial solution $P^* = \{G_1^*, G_2^*\}$, and perturbs the order the pairs in P^* and swaps the two transistors of randomly picked pairs into different group. In order to achieve a maximum diffusion-merging, our algorithm tries to find an Euler trail for each group in P^*.

It is notably that P^* is not a guarantee of valid Euler trails existing. An extra procedure is therefore invoked to validate it. This procedure walks through the solution and tries to construct Euler trail edge by edge for each group. Whenever it fails to join in the partially-completed path to continue the construction of the Euler trail, a "super edge" S, which connects the rear vertex in the partially-completed path and any vertex of the failed edge, is added to continue the path construction. An example is shown in Figure 6(a). The group $G_1 = \{M1,M5',M7',M8'\}$ in the solution P is randomly generated corresponding to the sub-graph in Figure 5(b). The construction of an Euler trail from G_1 fails between edges $M7'$ and $M8'$ due to the disjoint of vertex $VOUT$ to vertices of $M8'$. Therefore an super edge S connecting $VOUT$ and $IREF$ is added to validate the Euler trail. The valid Euler trail of G_1 then will be $\{D1, M1, D3, M5', GND, M7', VOUT, S, IREF, M8', GND\}$, and the super edges can be mapped to dummy gates or diffusion gaps in the final layout, as shown in Figure 6(b). The algorithm will achieve the maximum diffusion-merging by minimizing the total super edges in the solution.



(a)Example with super edge insertion (b)Super edge mapping

Fig. 6. Super edge insertion and mapping

The wirelength of a netlist in the solution is modeled as follows. Since all the transistors in both rows are aligned, the diffusions and the poly-gates are numbered in a uniform manner in horizontal direction, and the diffusions and the poly-gates are called slot hence. The length of a given netlist is measured by its span in horizontal direction in terms of slot. If a netlist connects both P- and N-transistors, during the separate optimization of P-/N-region, its length will be calculated from slot 0 to the max slot that the netlist span in that region. Figure 7 gives an example of wirelength calculation. The p/n interface is assumed to be slot 0 in both regions.

Therefore the cost function F_{cost} of our optimization algorithm is defined as follows,

$$F_{cost}(P) = \alpha \times W(P) + \beta \times T(P), \quad (1)$$

where $P = \{G_1, G_2\}$ is a solution, and $W(P)$ is the total wirelength of the solution in terms of slot. $T(P)$ is the total number of super edges in P after the Euler trail validation. α and β are tunable factors for balancing the total wirelength and diffusion merging. We can see that when $T(P)$ is minimized, so are the number of super edges. Thereby the layout with maximum diffusion merging is obtained, and its area is minimized as well.

IV. Generation Examples

Two circuits are used to generate the symmetrical layouts with the proposed algorithm. The first is a simple differential

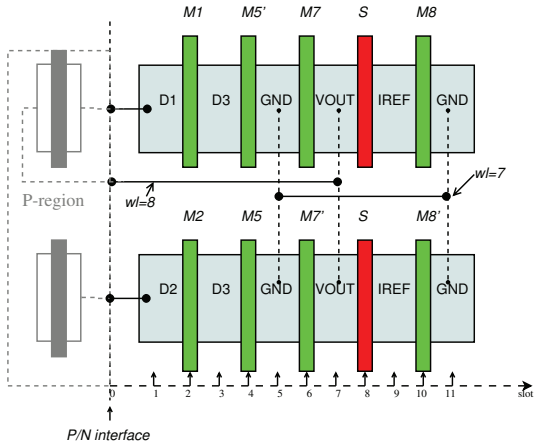


Fig. 7. Wirelength calculation

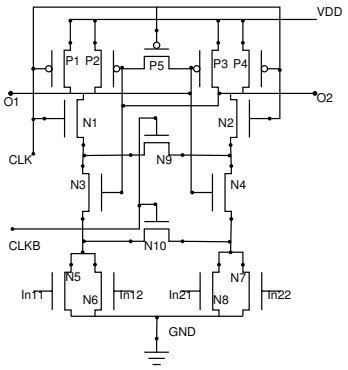


Fig. 9. Circuit of a pre-amplifier of a comparator

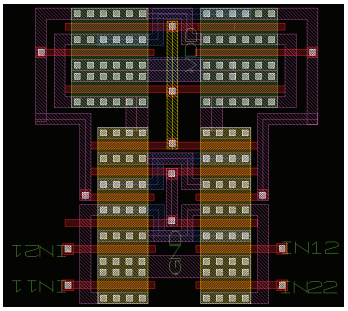


Fig. 10. Final layout of the pre-amplifier

opamp, which has been used to demonstrate how our algorithm works step by step in the previous section. The resultant symmetrical layout is given in Figure 8. The symmetry pairs (i.e., the transistors with symmetry constraint) given by the designer are $\{M1, M2\}$ and $\{M3, M4\}$. We can observe that the diffusion gaps exist in the layout due to the super edges in the optimized solution. However because of the good arrangement of transistors, only small extra routing effort is necessary except for the connections between P-/N-regions or the regions separated by diffusion gaps.

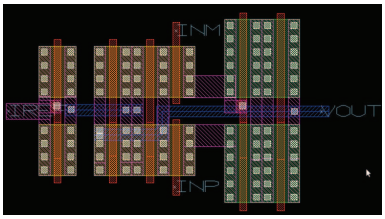


Fig. 8. Final layout of OPAMP

It also should be noted that because of the gate alignment of transistor pair, routing for connecting the gates of transistor-pair is almost unnecessary more than simply joining the two aligned poly-gates.

Another example is a pre-amplifying stage of a dynamic comparator[8], which has more transistors than the first example, as shown in Figure 9. The corresponding layout can be seen in Figure 10. In this result there is no diffusion gaps, i.e., no super edges are introduced in the optimized solution.

V. Conclusions and Future Works

This paper proposes a transistor-level symmetrical layout generation algorithm aiming at maximum diffusion-merging to the current paths for analog circuit. We present a SA-based algorithm to symmetrically assign the transistor pair into two rows and meanwhile minimize the total wirelength and diffusion gaps. Two examples are used to demonstrate the effectiveness of our algorithm.

The current version of our algorithm can only handle the symmetry constraints on transistor pairs, and only a twin-row style layout can be generated. Therefore one priority future work is to extend the algorithm to support more complicated case such as the symmetry constraints on transistor tuples and multi-row layout generation.

Since the two transistor rows are symmetrical about their mid-line, and the transistors in each pair are well-aligned in both rows, we can easily generate the symmetrical routing with the algorithm ROAD proposed in [5] because the non-cross-row net-segments can be routed in one row, and then the results can be mirrored into the other row. and the cross-row net-segments can be routed using "connector" proposed by [6] to match parasitics. This will be our another future work.

Acknowledgement

This research is supported by Semiconductor Technology Academic Research Center (STARC) in the topic of university research collaboration program: Variation-tolerant Analog LSI Design Methodology Based on Fine Grain Transistor Array.

REFERENCES

[1] R. A. Rutenbar and J. M. Cohn, "Layout tools for analog ics and mixed-signal socs: a survey," in *Proceedings of the 2000 international symposium on Physical design*, ser. ISPD '00. New York, NY, USA: ACM, 2000, pp. 76–83. [Online]. Available: http://doi.acm.org/10.1145/332357.332378 .

[2] F. Balasa and K. Lampaert, "Symmetry within the sequence-pair representation in the context ofplacement for analog design," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 19, no. 7, pp. 721–731, 2000.

[3] Y. Pang, F. Balasa, K. Lampaert, and C.-K. Cheng, "Block placement with symmetry constraints based on the o-tree non-slicing representation," in *DAC*, 2000, pp. 464–467.

[4] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, and L. R. Carley, *Analog Device-Level Layout Automation*. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

[5] E. Malavasi and A. Sangiovanni-Vincentelli, "Area routing for analog layout," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 12, no. 8, pp. 1186 –1197, aug 1993.

[6] E. Malavasi, E. Charbon, E. Felt, and A. Sangiovanni-Vincentelli, "Automation of ic layout with analog constraints," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 15, no. 8, pp. 923 –942, aug 1996.

[7] Y.-C. Tam, E. F. Young, and C. Chu, "Analog placement with symmetry and other placement constraints," in *Proc. ICCAD 2006*, 2006, pp. 349–354.

[8] K. Sushihara, and A. Matsuzawa, "A 7b 450MSample/s 50mW CMOS ADC in 0.3mm," in *Digest of. Technical Papers, ISSCC 2002*, 2002, pp.170-171.4,