Layout-Oriented Synthesis of High Performance Analog Circuits

Mohamed Dessouky, Marie-Minerve Louërat
Université Paris VI (55/65)
Laboratoire LIP6-ASIM
4 Place Jussieu.
75252 Paris Cedex 05. France
Mohamed.Dessouky@lip6.fr

Jacky Porte
École Nationale Supérieure
des Télécommunications
46 Rue Barrault.
75634 Paris Cedex 13. France
Porte@enst.fr

Abstract

This paper presents a methodology towards synthesis of high performance analog circuits. Layout parasitics are estimated and compensated during circuit sizing. Physical layout constraints are thus taken into consideration early in the design. This approach shortens the overall design time by avoiding laborious sizing-layout iterations. The approach has been implemented using two knowledge-based tools dedicated to analog circuit sizing and layout generation. An example of a high performance OTA is presented at the end to illustrate the effectiveness of the approach.

1. Introduction

During the design of high performance analog circuits, device matching, parasitics, thermal and substrate effects, reliability design rules must all be taken into account. All of these effects can be controlled with a good layout design performed either manually by an expert layout designer or using a dedicated automatic tool. However, the nominal values of performance specifications are subject to degradation due to a large number of parasitics which are generally difficult to estimate accurately before the actual layout is complete. Over-estimation of layout parasitics results in wasted power and area, while under-estimation of parasitics leads to circuits that do not meet the required specifications.

Two main approaches are used for layout generation: A knowledge-based approach which captures the expert design knowledge either in templates [2] or in a procedural form [9], and an optimization-based approach which tries to minimize layout parasitics through numerical optimization [1]. More recently, a performance-driven layout methodology has been introduced [5]. Performance specifications are mapped onto a set of constraints for critical parasitics which are then used to drive the layout tools. In [4], performance constraints are used to drive directly the layout tools. On the circuit sizing side, knowledge-based approaches [3] as well as optimization-based ones [7] are also used.

All of the systems cited above consider the layout as a step which follows the design synthesis process. The layout generation procedure does not interact during the design synthesis. So the circuit synthesis program has no information on the parasitics that the circuit is going to generate during the layout phase.

In this paper a methodology that couples both the circuit sizing and layout phases is presented. The methodology is an extension to that first presented in [8] by considering a more detailed parasitic extraction and analog layout constraints while treating circuit reliability conditions in the same time. Different layout styles can be used, their effect on the overall circuit performance is directly determined and compensated by the sizing procedure if needed. Layout techniques that minimize parasitic capacitances on certain nets and enhance the overall performance can be further exploited *during* the sizing phase in order to optimize certain design aspects. The methodology is implemented using two tools for circuit sizing and layout generation which are also presented here.

This paper is organized as follows: Section 2 introduces the overall methodology. In section 3 the layout generation tool is presented followed by the circuit sizing tool in section 4. An example is given in section 5. Finally, conclusions are summarized in section 6.

2. Layout-oriented design synthesis methodology

The problem of compensating layout parasitics is usually solved as demonstrated by the design flow shown in Fig. 1(a). The design process follows laborious iteration loops during which circuit sizing is followed by generating the layout, extracting the circuit netlist with layout parasitics and evaluating the effect of those parasitics in order to compensate for them by re-sizing the circuit. This resizing modifies the parasitics and the loop is repeated till a satisfying performance is obtained.

Fig. 1(b) shows the proposed layout-oriented methodology. Layout information is passed to the circuit sizing tool early in the design phase. Multiple calls to the layout tool are allowed as the design progresses. This approach guarantees a circuit that satisfies the performance specifications even in the presence of circuit parasitics. The accuracy is largely dependent on the precision of parasitic calculation by the layout tool, as well as its capability to take analog layout constraints into consideration. Physical layout constraints such as the global aspect ratio and circuit reliability design rules can be taken into account *during* circuit sizing.

The first circuit sizing is done assuming one fold per transistor. Only diffusion capacitances are considered. DC bias conditions are also calculated in order to satisfy the given specifications. The layout tool is then called, with the following information:

- Transistor sizes.
- Transistor currents.
- Layout options regarding the implementation of certain devices. For example, a differential pair can be implemented in an interdigitated or a common centroid configuration.
- Layout shape constraint.

The layout tool is then executed in a *parasitic calculation* mode. In this mode layout area optimization, based on the given shape constraint, results in a given number of folds for each transistor as well as full determination of the width and position of all routing wires which allows precise calculation of their capacitances. No layout is physically generated.

The following information is then sent back to the sizing program:

- Transistor layout style. This includes the number of folds of each transistor and their widths, the number of source/drain diffusions which are external, internal to the transistor or shared with other transistors.
- Parasitic routing capacitance including coupling capacitance between wires.
- Exact well sizes so that floating well capacitance can be calculated.

This allows the sizing tool to compensate for layout parasitics by simply modifying transistor sizes. This process is repeated till the calculated parasitics remain unchanged. At the end the layout tool is called in a *generation* mode where it physically creates the corresponding layout.

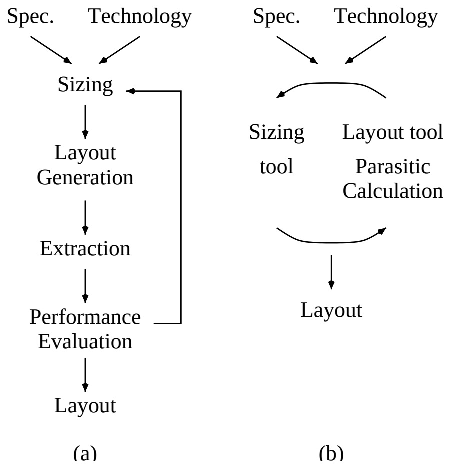

Figure 1. Design flow: (a) traditional and (b) proposed

3. Layout generation

In order to be used in the proposed design methodology, the layout generation tool must satisfy the following conditions:

- It must be fast as it is normally called several times during circuit sizing.
- It must support an accurate method for parasitic estimation.
- It must support conventional analog layout constraints.
- In order to explore various design space points, it must support different layout options for each device.

It is clear from the first condition that optimization-based layout generation approaches [1, 4, 5] can't be used due to their high computational cost. On the other hand, the knowledge-based approach seems to be attractive for its short layout generation time. The procedural approach has been thus chosen for reasons of flexibility and generality. This is achieved through a dedicated layout language (CAIRO) that allows to easily describe relatively both module placement and routing.

Analog layout effects must be carefully treated. Layout constraints taken into account in the language are presented hereafter.

Parasitic constraints. All transistors are built using a single motif generator which allows total control over terminals and wires. This gives an additional degree of freedom to control overlapping parasitic capacitance inside transistors according to the target application [10].

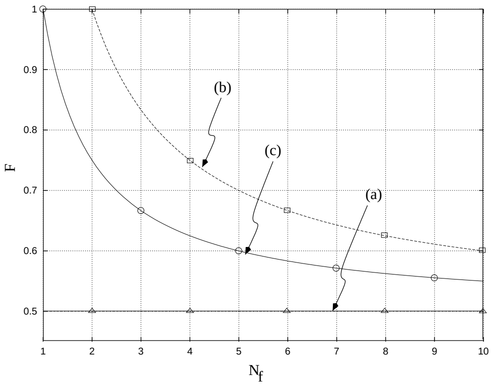

Figure 2. Capacitance reduction factor F

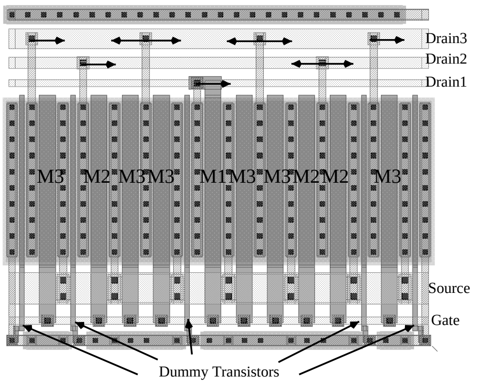

Figure 3. Current mirror, M1:M2:M3=1:3:6

Transistor folding reduces the diffusion-bulk parasitic capacitance (drain-bulk and source-bulk capacitances). This is due to the sharing of these diffusion areas between folds. The total effective diffusion width W_{eff} is usually a fraction F of the transistor width W ($W_{eff} = F.W$), where F is the capacitance *reduction factor*. In case of a non-folded transistor $F = 1$. While for a folded one, F depends on the number of folds N_f and the position of the diffusion (for alternate source/drain diffusions) as follows:

$$F = \begin{cases} \frac{1}{2} & \text{if } N_f \text{ even \& internal diffusion (a),} \\ \frac{N_f+2}{2N_f} & \text{if } N_f \text{ even \& external diffusion (b),} \\ \frac{N_f+1}{2N_f} & \text{if } N_f \text{ odd} \qquad\qquad\qquad\qquad \text{(c).} \end{cases} \qquad (1)$$

As shown in Fig. 2, this reduction factor F decreases significantly for the first few folds for cases (b) and (c). It is clear that this parasitic capacitance can be minimized on a given net by controlling the folds of the transistor connected to this net to be even, and connecting the internal diffusion to this net (case (a)). This parasitic control is used by the language to enhance the frequency characteristics of the layout.

Matching constraints. Special layout styles of transistors must be used in order to minimize device mismatch. Based on the motif generator, complex transistor device generators are built. This includes interleaved and common centroid configurations. As the mismatch between transistors is also dependent on their relative channel orientation, a special algorithm that controls transistor placement in stacks which takes into consideration current direction is developed based on the insertion of dummy transistors [6]. As an example, Fig. 3 shows a current mirror with three transistors having width ratios of $M1 : M2 : M3 = 1 : 3 : 6$. The arrows show the current orientation in each transistor. The *current mismatch* [6] is minimized by properly choosing the channel orientation of transistors. In addition, transistor stacking is done such that all transistors are centered around the mid-point of the stack.

Reliability constraints. Reliability design rules are important for the long-term functionality of the circuit. DC current information is used to adjust wire widths inside each module as well as routing wires in order to respect the maximum current density allowed by the technology used. This prevents electromigration from taking place which may lead to open circuits in wires subjected to high current densities [10]. The number of contacts are also increased for wide wires in order to minimize their resistance according to the reliability design rules. This is clearly shown in the current mirror shown in Fig. 3 where wire widths and contact numbers have been adjusted for each transistor assuming high current densities.

Shape constraints. The layout is usually driven by a shape constraint (for example a given height or aspect ratio). Given this constraint, the language tries to produce the corresponding most compact layout. Area optimization is done using a simple and fast algorithm based on *shape functions* and *slicing structures* [2]. The language constructs allow to build up the appropriate slicing structure for the circuit.

Parasitic extraction. In the *parasitic estimation* mode, after the determination of the shape of each module in the area optimization step, each module calculates the values of parasitic components in a predefined parasitic model. Routing parasitics are then calculated. All parasitic estimations are done using simple geometrical methods which combine reasonable accuracy with low computational cost.

Technology independence. Technology independence is a key feature of any layout tool. A symbolic layout approach is used such that all procedures are technology independent.

Specification	Case (1)	Case (2)	Case (3)	Case (4)
DC gain (dB)	70.1(70.1)	55.0(56.59)	66.1(66.1)	64.7(64.7)
GBW (MHz)	64.9(58.1)	66.5(71.2)	65.0(62.6)	65.8(66.1)
Phase margin (degrees)	65.3(56.3)	65.4(72.4)	65.4(64.4)	65.15(65.4)
Slew rate ($V/\mu s$)	94.0(86.5)	103.0(98.1)	93.3(93.3)	93.0(94.4)
CMRR (dB)	100.7(100.7)	76.9(79.6)	93.9(93.9)	91.6(91.6)
Offset voltage (mV)	0.0(0.0)	0.0(-0.1)	0.0(0.0)	0.0(0.0)
Output Resistance ($Mohm$)	2.4(2.4)	0.38(0.47)	1.5(1.47)	1.23(1.23)
Input noise voltage (μV)	83.9(96.1)	101.6(85.6)	83.3(87.8)	82.7(85.8)
Thermal noise density ($nV/\sqrt{Hz}$)	7.2	6.98	7.15	7.13
Flicker noise density @1Hz ($\mu V/\sqrt{Hz}$)	1.95(3.64)	1.4(8.1)	2.59(4.85)	2.82(5.28)
Power dissipation (mW)	2.0(2.0)	2.4(2.2)	2.1(2.1)	2.1(2.1)

Input specifications: $VDD = 3.3$V, $GBW = 65$MHz, phase margin = 65degrees, $C_{load} = 3$pF,
Input CM range = $[-0.55, 1.84]$V, Output range = $[0.51, 2.31]$V.
Case 1: Sizing with no layout capacitances (Neither diffusion nor routing).
Case 2: Sizing with diffusion capacitance assuming single transistor folds and no routing capacitance.
Case 3: Sizing with calculation of exact diffusion capacitance and neglecting routing capacitances.
Case 4: Sizing considering all layout parasitics.
Values between brackets are obtained from layout generation, extraction and simulation.

Table 1. Sizing, layout and simulation results

4. Circuit sizing

The circuit sizing tool (COMDIAC) uses a knowledge-based approach. Circuit topologies are selected from among fixed alternative (*design style selections*) [3] each with associated detailed design knowledge. A hierarchical approach is used such that fixed routines have been developed for frequently used building blocks with different styles. This simplifies the addition of new topologies.

Sizing is a direct and fast procedure. The dc operating point of *all* transistors is fixed at the beginning of the sizing process. As V_{TH} changes with transistor lengths during sizing, the effective gate-source voltage $V_{GS} - V_{TH}$ is held constant rather than V_{GS}. The procedure starts by an heuristic estimation of transistor dc currents needed for every transistor to realize the given specifications. It then calculates the corresponding transistor sizes by simple monotonic numerical iterations. Performance is then evaluated using predefined equations, and the process is repeated till satisfactory results are obtained.

For example, for opamps the tool starts by estimating transistor currents for the given gain-bandwidth product (GBW). It then iterates on transistor lengths till the required phase margin is achieved. If the resulting GBW is not satisfactory, a new current estimation is calculated and the whole process is repeated. Other specifications such as the opamp gain, noise performance, slew-rate, input common mode and output voltage range are all *results* of the synthesis. They can be controlled by fixing certain transistor lengths or biasing point at the beginning of the synthesis.

Advanced transistor models like *BSIM3V3* and *MM9* as well as traditional SPICE MOS levels 1, 2 and 3 incorporated in the tool are used during circuit sizing and performance evaluation. This has largely improved the calculation accuracy.

Fixing the operating point of each transistor taking into account considerations like matching and temperature dependance increases the reliability of the produced circuits. The fact that the sizing process is very fast and highly accurate allows interactive exploration of wide variety of design space points. A verification interface has also been developed which controls a verification-by-simulation process. It also permits to undergo statistical analysis to check the reliability of the synthesized circuit. A technology evaluation interface allows to easily characterize different technologies and helps to choose the most suitable technology.

5. Example

As an example, the folded cascode OTA shown in Fig. 4 has been synthesized using different layout parasitic considerations in a 0.6-μm technology. The OTA is sized for a VDD of 3.3V, a GBW of 65MHz, a phase margin of 65degrees and a load capacitance of 3pF. For comparison, the input common mode voltage range as well as the output voltage range are kept the same for all cases.

Table 1 shows the obtained results, it also shows the results of simulations of the extracted netlist with all parasitics (diffusion, routing and coupling capacitances) between brackets. Final extraction has been done using the commercial Cadence design system. In case (1) no layout capacitances (neither diffusion nor routing) have been taken in consideration, only gate capacitances and transistor folding are considered. It can be seen that all dc characteristics match the extracted layout simulation results, while for the GBW and phase margin we can notice a considerable difference. In case (2) diffusion capacitance has been taken into consideration but assuming only one fold per transistor and neglecting routing capacitance, i.e. no layout information is used during synthesis. Results show that the GBW and

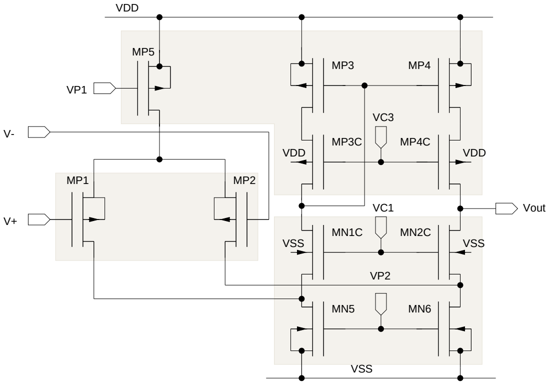

Figure 4. Folded cascode OTA

phase margin exceed the required specifications. In fact, as the diffusion capacitance is over-estimated, thus the obtained transistor sizes are smaller. This implies that other specifications like the input noise, the dc gain and the output resistance could not be optimized. Note also the resulting offset voltage after folding due to the slight modification of transistor widths needed by layout grid. Case (3) shows sizing results with layout information concerning *exact* diffusion capacitance, no routing capacitance is considered. We notice only a slight difference in the *GBW* and phase margin between synthesized and extracted netlist simulation. However, both specifications could not be satisfied. Case (4) shows results with all parasitic capacitance information being considered during the synthesis phase. All results match the extracted netlist simulations.

Three calls of the layout tool were needed before parasitic convergence. The sizing time for each case including layout calls does not exceed two minutes.

Fig. 5 shows the generated layout for case (4). As can be seen from the layout, all transistor folds are chosen such that *drains* are internal diffusions to minimize drain capacitance and enhance the frequency behavior. The input differential pair is in a common centroid style with dummy transistors at the end in order to improve transistor matching.

6. Conclusions and future work

An approach to closely couple circuit and layout synthesis has been presented. The approach is implemented using two tools which address both circuit and layout synthesis.

Procedural layout is shown to be the best suitable layout method for such methodologies due to its fast layout generation time. Several layout constraints can be easily taken into consideration during circuit sizing.

Circuit synthesis relies on rapid calculations using built-in design plans. The use of hierarchy simplifies the addition of new topologies in the tool. Accuracy with respect to simulation is greatly improved by using the same transistor

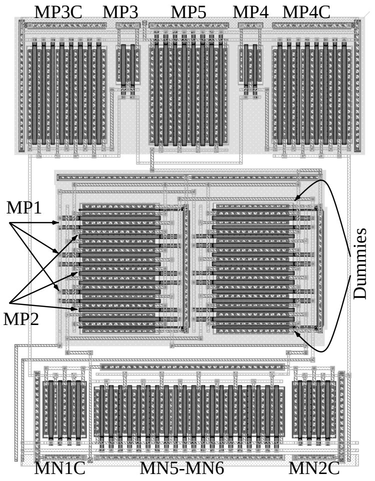

Figure 5. Layout of the circuit shown in Fig. 4

models implemented in the latter.

Future work includes synthesis of larger systems as switched capacitor filters and A/D converters using the same methodology.

References

[1] J. M. Cohn, R. A. Rutenbar, and L. R. Carley. KOAN/ANAGRAM II: New Tools for Device-Level Analog Placement and Routing. *IEEE J. of Solid-State Circuits*, 26(3):330–342, Mar. 1991.

[2] J. D. Conway and G. G. Schrooten. An Automatic Layout Generator for Analog Circuits. In *Proc. European Design Automation Conf.*, pages 513–519, 1992.

[3] M. G. R. Degrauwe and *et. al.* IDAC: An Interactive Design Tool for Analog CMOS Circuits. *IEEE J. of Solid-State Circuits*, 22(6):1106–1115, Dec. 1987.

[4] K. Lampaert, G. Gielen, and W. M. Sansen. A Performance-Driven Placement Tool for Analog Integrated Circuits. *IEEE J. of Solid-State Circuits*, 30(7):773–780, July 1995.

[5] E. Malavasi, E. Charbon, E. Felt, and A. Sangiovanni-Vincentelli. Automation of ICL Layout with Analog Constraints. *IEEE Trans. Computer-Aided Design*, 15(8):923–942, Aug. 1996.

[6] E. Malavasi and D. Pandini. Optimum CMOS Stack Generation with Analog Constraints. *IEEE Trans. Computer-Aided Design*, 14(1):107–122, Jan. 1995.

[7] E. S. Ochotta, R. A. Rutenbar, and L. R. Carley. Synthesis of high-performace analog circuits in ASTRX/OBLX. *IEEE Trans. Computer-Aided Design*, 15(3):273–294, Mar. 1996.

[8] H. Onodera, H. Kanbara, and K. Tamaru. Operational-Amplifier Compilation with Performance Optimization. *IEEE J. of Solid-State Circuits*, 25(2):466–473, Apr. 1990.

[9] B. R. Owen, R. Duncan, S. Jantzi, C. Ouslis, S. Rezania, and K. Martin. BALLISTIC: An Analog Layout Language. In *Proc. IEEE Custom Integrated Circuits Conf.*, 1995.

[10] M. Wolf and U. Kleine. Reliability Driven Module Generation for Analog Layouts. In *Proc. Int. Symposium on Circuits and Systems*, pages 412–415, June 1999.