Hierarchical Graph-Based Sizing for Analog Cells Through Reference Transistors

Ramy Iskander, Marie-Minerve Louërat
LIP6 Laboratory
Université Pierre et Marie Curie
4 Place Jussieu, 75252 Paris, France
{ramy.iskander, marie-minerve.louerat}@lip6.fr

Andreas Kaiser
IEMN
ISEN Department
41 Bld. Vauban, 59046 Lille, France
andreas.kaiser@isen.fr

***Abstract*— In this paper, an algorithm for automatic sizing and operating point computation of hierarchical knowledge-based analog cells is presented. The algorithm assumes that an analog cell is described as a hierarchy of devices and modules inside our dedicated framework CAIRO+. Within devices, the concept of the reference transistor is elaborated. The latter is used to construct device dependency graphs for each device. Module dependency graphs are constructed by merging graphs of all children modules and devices. Inside each device, the reference transistor controls the sizing and biasing of the whole device. It propagates electrical parameters to secondary transistors. The used propagation technique ensures that all the device constraints are satisfied by construction. The algorithm was used to size and bias a two-stages single-ended OTA amplifier. It proved to be successful in DC operating point calculation in the context of hierarchical knowledge-based framework.**

I. Introduction

Since the last few decades, research in analog synthesis has been focusing on two major directions: knowledge-based synthesis and simulation-based synthesis. In knowledge-based synthesis, the designer is fully responsible on documenting his own expertise in the form of a reusable knowledge. In OASYS [1], the codification process includes the calculation of DC operating point and the dimensions for each sub-block used in the hierarchy of the analog circuit. The codification task is considered as a tedious work by the designers. On the other hand, in simulation-based synthesis, a simulator is called to calculate the DC operating point at each design point in the design space. DC calculation involves parsing the input netlist, creating the conductance matrix for input circuit and applying the nodal analysis. In addition, some runtime overhead may exist during the simulator execution and control. These factors contribute to the increase of execution time of individual simulations. The impact on execution time will be great, as shown in MAELSTROM [2], that executes thousands of simulations during optimization. It is clear that DC operating point calculation is a mandatory task for both knowledge-based and simulation-based synthesis.

Several researches attempted to solve the problem of DC operating point calculation. Maulik [3] used a relaxed DC formulation that considers the DC solution as a part of the cost function. Gielen *et al* [4] proposed a complete DC solving method that introduces branch currents and voltages in a minimax optimization to solve for the DC operating point. Model equations are then called to solve for the transistor widths. The main disadvantage of these methods is the use of optimization to solve for the DC operating point.

This paper proposes a procedural method to construct analog reusable blocks [5], called *devices*. These blocks can be reused to build more complex higher-level blocks that are called *modules*. To properly construct a device, the concept of *transistor packing* is introduced. It gives guidelines on how to choose to pack transistors into a single device. Also the concept of the *reference transistor* is proposed. It limits the sizing and biasing of a device to the reference transistor. In order to control the flow of electrical information from the reference transistor to the secondary ones, the concept of *device constraints* is developed. Finally, to determine electrical and small signal parameters for transistors in a device, the concept of *sizing and biasing operators* is introduced.

The paper is organized in five main sections. Section one is an introduction. Section two describes some aspects of CAIRO+ framework. Section three discusses the methodology of hierarchical sizing and operating point computation using parameter propagation. Section four discusses the results of sizing and biasing of a transconductance amplifier. Section five concludes the proposed work.

II. Hierarchy in CAIRO+ Framework

The CAIRO+ framework [6], [7] is mainly used to develop parameterized generators. A parameterized generator is an analog reusable block that receives design and technology parameters and provides behavioral, structural and physical views. Devices and modules are considered as parameterized generators. An analog cell can be defined as a hierarchy of modules and devices. Higher-level modules may instantiate lower-level modules and devices. The lowest hierarchical level is a standard transistor model. Each hierarchical level propagates electrical parameters with its predecessor and successor levels only. Fig. (1) illustrates an example of hierarchical instantiation and parameter propagation in CAIRO+.

III. Hierarchical Sizing

In this section, a general formalization of *hierarchical sizing* is proposed. In operating point driven sizing [8], one specifies

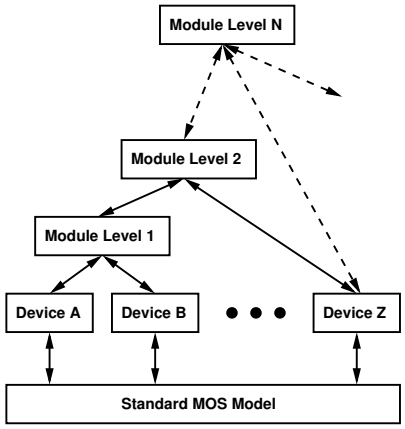

Fig. 1. Hierarchical levels and parameter propagation in CAIRO+

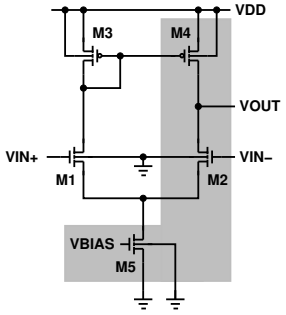

Fig. 2. Transconductance amplifier

the operating point and determines transistor widths out of it. Once the dimensions are found, all small signal parameters are determined for each transistor. The designer specifies the operating point in terms of electrical parameters such as $Temp$, I_{DS}, V_{EG}, V_{GS}, V_D, V_B, V_S, and V_G. In this section, the OTA amplifier in Fig. (2) will be sized and biased using this traditional approach. The sizing scenario will be further analyzed. Consequently, a more general approach for hierarchical operating point driven sizing will be formalized.

A. Traditional Approach

Let us suppose that V_{OUT}, V_{IN+} and V_{IN-} are known. The designer may specify the overdrive voltage $V_{EG} = V_{GS} - V_{TH}$ for all transistors. If L_4, I_{DS4}, V_{DS4}, μ_p, C_{ox} and λ_p are known, then the width of M_4 can be calculated from eq. (1)

$$I_{DS}(PMOS) = \frac{\mu_p}{2} C_{ox} \frac{W}{L} (V_{EG})^2 (1 + \lambda_p V_{DS}) \quad (1)$$

Next, the designer chooses to impose V_{DS2}. Knowing the same parameters as for M_4, the designer uses eq. (2) to determine W_2.

$$I_{DS}(NMOS) = \frac{\mu_n}{2} C_{ox} \frac{W}{L} (V_{EG})^2 (1 + \lambda_n V_{DS}) \quad (2)$$

Once M_4 and M_2 are sized, their widths are copied to M_3 and M_1 respectively. Finally, the designer biases M_5 using $V_{BIAS} = V_{EG5} + V_{TH5}$ and uses eq. (2) to determine W_5. Once all the geometrical and electrical parameters are fixed, the small signal parameters are directly determined.

B. The Concept of Transistor Packing

A *device* is defined as one or more transistors packed together as one atomic and reusable building block. Fig. (2) shows that the OTA consists of a set of primitive analog devices: the current mirror (M_3, M_4), the differential pair (M_1, M_2) and the MOS transistor M_5. In order to elaborate device representation, some conditions have been identified for packing transistors. At least, one of these conditions should be respected during device construction. To construct a device in CAIRO+, the following conditions have been proposed:

1) Any set of transistors that form one distinct electrical function should be packed together
2) Any set of transistors that share a subset of electrical parameters, such as L, V_{GS}, ... , may be packed together
3) Any set of transistors that should be matched in the physical level should be packed together

As shown in Fig. (2), transistors M_3 and M_4 should be packed together as:

1) They form one complete function of a current mirror
2) They share the same W, L and V_{GS}
3) They should be physically matched

The same criteria holds for the differential pair (M_1, M_2).

C. The Concept of Reference Transistor

To simplify analog design, the designer selects to size a minimum set of transistors. In order to define each of these transistors, the concept of *the reference transistor* has been introduced. A device contains only one reference transistor, which is sized first. The geometrical and electrical parameters of the reference transistor define, in a unique way and through simple relations, the ones of the secondary transistors. It can be said that the parameters of the reference transistor are *propagated* to the secondary ones. As a result, the designer may choose:

1) M_4 as a reference transistor for (M_3, M_4)
2) M_2 as a reference transistor for (M_1, M_2)
3) M_5 as a reference transistor for itself

These reference transistors are marked as grayed in Fig. (2).

D. Device Constraints

In the traditional approach described in subsection (A), the designer copied the widths of M_4 and M_2 to M_3 and M_1 respectively. This reflects that electrical parameters may propagate from the reference transistor to secondary ones. A simple approach for propagation is to define linear equality constraints having the form:

$$\Big[P_{elec,i}\Big]_{N\times 1} = \Big[K_i\Big]_{N\times M} \cdot \Big[P_{elec,ref}\Big]_{M\times 1} \quad (3)$$

where $P_{elec,i}$ is a matrix of the electrical parameters of all secondary transistors, K_i is a sparse matrix of constants and $P_{elec,ref}$ is a matrix of the electrical parameters of the reference transistor.

In our methodology described in [9], a sizing procedure is extracted for devices and modules. The sizing procedure

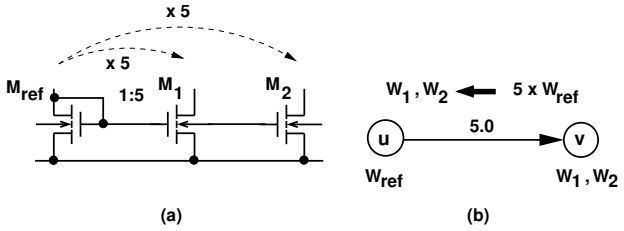

Fig. 3. (a) Device constraint on widths, (b) Representation

TABLE I
DEFINITION OF SOME SIZING & BIASING OPERATORS

Operator	Definition
$OPVS(V_{EG}, V_B)$	$(V_S, V_{TH}, W) \Leftarrow Temp, I_{DS}, L, V_{EG}, V_D, V_G, V_B$
$OPVS(V_{EG})$	$(V_S, V_B, V_{TH}, W) \Leftarrow Temp, I_{DS}, L, V_{EG}, V_D, V_G$
$OPVG(V_{EG})$	$(V_B, V_G, V_{TH}, W) \Leftarrow Temp, I_{DS}, L, V_{EG}, V_D, V_S$
$OPVGD(V_{EG})$	$(V_B, V_G, V_D, V_{TH}, W) \Leftarrow Temp, I_{DS}, L, V_{EG}, V_S$
$OPW(V_{GS})$	$(V_B, V_{TH}, W) \Leftarrow Temp, I_{DS}, L, V_D, V_G, V_S$

for either a device or a lower-level module, is represented in the form of a dependency graph. In the device dependency graph, constraints appear as graph nodes connected by directed and weighted arcs. Graph nodes represent electrical parameters such as $Temp$, W, L, I_{DS}, V_{EG}, V_{GS}, V_D, V_B, V_S, or V_G. Directed arcs represent a weighted dependence of one node v on the other node u. Formally, this is written as $v \leftarrow u$. As part of the graph, device constraints are guaranteed to be satisfied by construction. Fig. (3) shows a constraint representation for the width of secondary transistors. This can be described by the matrix equation

$$\begin{bmatrix} \mathrm{W}_1 \\ \mathrm{W}_2 \end{bmatrix}_{2\times 1} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}_{2\times 1} \cdot [W_{ref}]_{1\times 1} \qquad (4)$$

Different types of constraints can be defined into a device e.g. *functional* and *robustness* constraints as in *sizing rules method* [10].

E. Sizing & Biasing Operators

In the traditional approach, the designer used the quadratic model eq. (1) to calculate the width of a MOS transistor. This quadratic equation represents a simplified model for the MOS transistor. Since the focus was on the integration of standard accurate transistor models, 32 sizing and biasing operators have been designed and implemented in CAIRO+ framework. Each sizing and biasing operator has the form

$$OP{<}class{>}(RV_i, ...) : (LV_j, ...) \Leftarrow (RV_n, ...) \qquad (5)$$

where $<class>$ represents the main electrical parameter to be calculated, RV_i is a subset of the known electrical parameters that dictates which operator version to apply, RV_n is the set of all known electrical parameters required by the operator to execute, and LV_j is the set of unknown electrical parameters that are calculated by this operator. It is said that a parameter is known, if it is either fixed a priori by the designer or previously calculated by CAIRO+ during sizing. As an example, Table I shows the definition of the operator

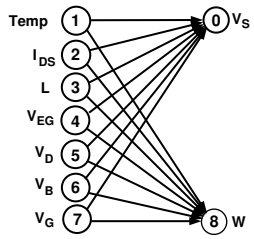

Fig. 4. Dependency graph in case of $OPVS$ for the differential pair

$OPVS$. The $OPVS$ operator class is *source voltage*. The operator has two versions. The first version $OPVS(V_{EG}, V_B)$ is called whenever V_{EG} is known and the reference transistor is not bulk-source connected i.e. V_B should be fixed by the designer. It determines V_S, V_{TH} and W, simultaneously, as a function of all the parameters listed on the right side of the dependence arrow. The second version $OPVS(V_{EG})$ is called whenever V_{EG} is known and the reference transistor is bulk-source connected. It also determines V_S, V_B, V_{TH} and W, simultaneously, as a function of all the parameters listed on the right side of the arrow. Based on the structure of the device and on the set of unknown parameters, the framework selects the most suitable operator to be applied to the reference transistor of each device. This selected operator is used to generate the dependency graph of the device. This graph represents the sizing procedure for the device. As an example Fig. (4) illustrates the generated dependency graph in case of selecting the operator $OPVS$ for the differential pair (M_1, M_2). This graph shows the dependency relation of both unknown parameters V_S and W on the known parameters: $Temp$, I_{DS}, L, V_{EG}, V_D, V_B and V_G. Obviously, each device may include a subset or all of the defined operators depending on its configuration i.e. diode-connected or not.

F. CREATE *and* SIZE *Procedures*

In CAIRO+, each device has its own *CREATE* and *SIZE* procedures. Inside the *CREATE* procedure, the device declares which transistor is the reference transistor. The device also includes the operators that are needed, particularly, by its reference transistor. Inside the *SIZE* procedure, device constraints are specified and the CAIRO+ framework is called. During execution, these procedures are used to generate the device dependency graph based on the reference transistor. The graphs for all devices and lower-level modules are then merged to form the dependency graph of the root module. The root module dependency graph represents the sizing procedure for the whole analog cell. The module dependency graph is further divided into computational levels. These levels are traversed sequentially. In each level, the corresponding sizing and biasing operator is called to calculate the unknown parameter at each node, if any. In the last level, widths are calculated for all transistors in all devices.

IV. RESULTS

The two-stages single-ended OTA amplifier is shown in Fig. (5). Each transistor packing is shown in a dashed box.

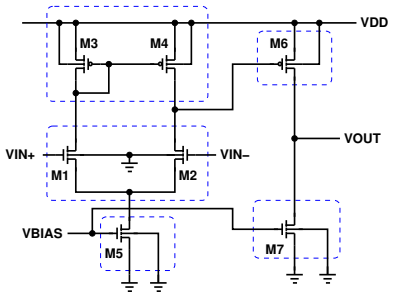

Fig. 5. Two-stages single-ended OTA amplifier

TABLE II
PARAMETERS OF THE DIFFERENTIAL PAIR

Transistor Parameters	M_1 and M_2 Algorithm	M_1 Simulation	M_2 Simulation
$I_{DS}(\mu A)$	49.998	50.00764	50.00764
$V_{GS}(V)$	0.461358	0.4613924	0.4613924
$V_{TH}(V)$	0.34135	0.34134	0.34134
$g_m(m\Omega^{-1})$	0.6707	0.6709299	0.6709299
$g_{ds}(\mu\Omega^{-1})$	4.34349	4.34349	4.345548
$C_{ds}(fF)$	0.229801	0.229797	0.229797
$C_{gs}(pF)$	0.1562	0.1562078	0.1562078

Transistors M_1, M_3, M_5, M_6 and M_7 were selected as the reference transistors. It will be assumed that the amplifier is designed to reduce the systematic offset voltage i.e $V_{DS4} = V_{GS6}$. After merging the devices graphs, the resulting module dependency graph is shown in Fig. (6). Each column consists of all nodes sharing the same computational level. A more detailed look into the dependency graph reveals that:

1) The rightmost computational level shows the computation of transistor widths in all devices
2) The current mirror nodes $(CM/L, 3)$, $(CM/V_{EG}, 5)$, ... propagates parameters to M_3 through $(M_3, M_4/L, 2)$, $(M_3, M_4/V_{EG}, 4)$, ... Since M_3 and M_4 share the same electrical parameters, they share the same nodes
3) $(V_S, 23)$ appears in the module dependency graph for the differential pair. Since the bulk and source are unconnected, operator $OPVS(V_{EG}, V_B)$ has been selected. It is a function of $(V_D, 9)$, $(Temp, 12)$, $(L, 14)$, $(V_{EG}, 16)$, $(I_{DS}, 18)$, $(V_B, 21)$ and $(V_G, 22)$.

Finally, the graph is traversed and executed. Table II shows all parameters for the differential pair. These are calculated using our approach and then compared to the results of an analog simulator. From the table, it is clear that our methodology can precisely size and bias analog cells.

V. CONCLUSION

In this paper, a methodology for hierarchical sizing and operating point computation is presented. Our method automatically creates suitable hierarchical sizing procedure for an arbitrary analog cell. The sizing procedure is described as a dependency graph holding hierarchical information about parameter propagation. The methodology proved to be successful in sizing a two-stages single-ended OTA amplifier. In the context of hierarchical knowledge-based frameworks, our method reaches the same order of precision of an analog simulator.

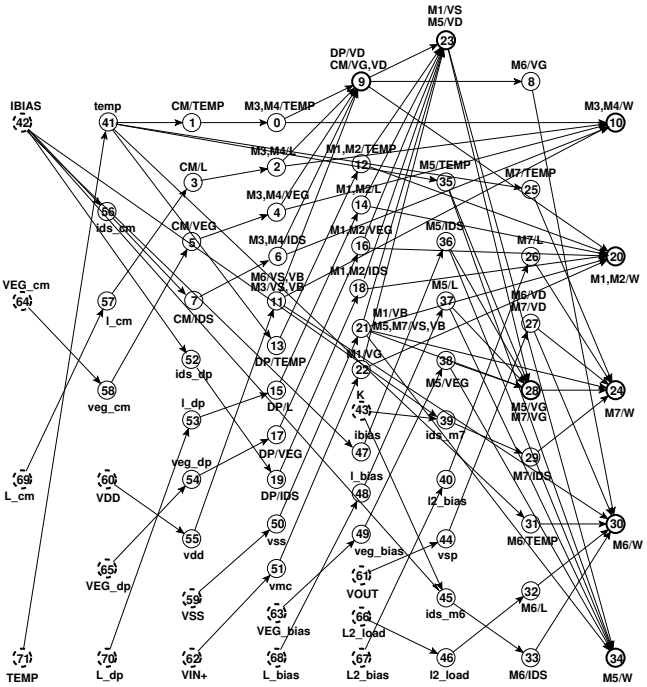

Fig. 6. Module dependency graph for the two-stages single-ended OTA amplifier: *bold dashed* nodes are the known parameters, *bold* nodes are the unknown parameters and *fine* nodes illustrates parameter propagation

REFERENCES

[1] R. Harjani, R. A. Rutenbar, and L. R. Carley, "OASYS: A framework for analog circuit synthesis," *IEEE Trans. Computer-Aided Design*, vol. 8, no. 12, pp. 1247–1266, Dec. 1989.

[2] M. Krasnicki, R. Phelps, R. A. Rutenbar, and L. R. Carley, "MAELSTROM: Efficient simulation-based synthesis for custom analog cells," in *Proceedings of Design Automation Conference*, June 1999, pp. 945–950.

[3] P. C. Maulik, L. R. Carley, and D. J. Allstot, "Sizing of cell-level analog circuits using constrained optimization techniques," *IEEE J. Solid-State Circuits*, vol. 28, no. 3, pp. 233–241, Mar. 1993.

[4] G. V. der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. Gielen, W. Sansen, P. Veselinovic, and D. Leenaerts, "AMGIE a synthesis environment for CMOS analog integrated circuits," *IEEE Trans. Circuits Syst.*, 2001.

[5] R. Castro-Lopez, F. V. Fernandez, and A. R. Vazquez, "A reuse-based framework for the design of analog and mixed-signal ICs," in *Proceedings of SPIE'05*, vol. 5837, no. 3, May 2005, pp. 25–36.

[6] V. Bourguet, L. de Lamarre, and M. M. Louërat, "Analog IC design with a library of parameterized device generators," in *XIX Conference on Design of Circuits and Integrated Systems DCIS'04*, Nov. 2004.

[7] R. Iskander, L. de Lamarre, A. Kaiser, and M. M. Louërat, "Design space exploration for analog IPs using CAIRO+," in *IEEE International Conference on Electrical Electronic and Computer Engineering*, Sept. 2004, pp. 473–476.

[8] F. Leyn, G. Gielen, and W. Sansen, "An efficient DC root solving algorithm with guaranteed convergence for analog integrated CMOS circuits," in *Proceedings ACM/IEEE International Conference on Computer-Aided Design*, Nov. 1998, pp. 304–307.

[9] R. Iskander, M. M. Louërat, and A. Kaiser, "Automatic biasing point extraction and design plan generation for analog IPs," in *The 48th IEEE MidWest Symposium on Circuits And Systems*, Aug. 2005.

[10] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich, "The sizing rules method for analog integrated circuit design," in *IEEE International Conference on Computer-Aided Design*, 2001.