The Sizing Rules Method for Analog Integrated Circuit Design

H. Graeb[1], S. Zizala[2], J. Eckmueller[2], K. Antreich[1]

[1] **Institute of Electronic Design Automation, Technical University of Munich**
[2] **Infineon Technologies, Munich**

ABSTRACT

This paper presents the *sizing rules method* for analog CMOS circuit design that consists of: first, the development of a hierarchical library of transistor pair groups as basic building blocks for analog CMOS circuits; second, the derivation of a hierarchical generic list of constraints that must be satisfied to guarantee the function of each block and its reliability with respect to physical effects; and third, the development of an automatic recognition of building blocks in a circuit schematic. The sizing rules method efficiently captures design knowledge on the technology-specific level of transistor pair groups. This reduces the preparatory modeling effort for analog circuit synthesis. Results of industrial applications to circuit sizing, design centering, response surface modeling and analog placement show the significance of the sizing rules method. Sizing rules especially make sure that automatic circuit sizing and design centering lead to technically meaningful and robust results.

1. INTRODUCTION

Analog components are an important part of integrated systems: either in terms of elements and area in mixed-signal systems, or as vital parts in digital systems, for instance power-on reset, pad driving, or clock generation. Despite their importance, design automation for analog circuits still lags behind that of digital circuits. As a consequence, analog components often are a bottleneck in the design flow.

Analog synthesis is complicated because it does not only consist of topology and layout synthesis but also of component sizing. Additionally, it has to incorporate physical effects like process variations, variations of operating conditions, matching constraints, or noise. It becomes even more complicated, as more and more mixed-signal systems and systems-on-chips are designed with customized analog and especially RF components. From the 70ies to the 90ies, analog topology synthesis [1, 2, 3, 4], nominal design optimization [5, 6, 7] and sizing with respect to tolerances (design centering, yield optimzation) [8, 9, 10, 11, 12, 13] were in the focus of research interest. The approaches include equation-based methods like GPCAD [14], or AMGIE [15], where design equations are derived with the help of symbolic analysis [16], and simulation-based methods like ASTRX/OBLX [17, 18] and [12, 19, 20]. Recent publications revealed that sizing tasks have a key potential for providing automation support to the designer, especially when consistently considering process and operating tolerances and mismatch [21, 22, 23, 24, 25].

A major obstacle of automatic sizing in practice are the usually incomplete circuit specifications. Specifying circuit performance bounds e.g. for gain, transit frequency, slew rate, common mode rejection ratio or power supply rejection ratio of an operational amplifier is not sufficient to prevent mathematical optimizers from driving the circuit into technically senseless regions, e.g. transistors out of saturation. This type of malfunction often does not appear in the nominal circuit performance but leads to an increased performance sensitivity with respect to process and operating variations and to noise [23], (see Fig. 12).

In order to cope with that problem, most of the approaches to analog synthesis mention and give examples of e.g. ”design-space constraints” [26], ”dimension constraints” [14], ”manufacturability and operationality constraints” [22], ”component constraints” [24], etc., but do not go into further details. To our knowledge, a *systematics* of such *sizing rules* that enables their *automatic handling*, as given in this paper, has not been presented so far. In this paper, a systematic approach to capture design constraint knowledge for automatic sizing is presented. The presentation aims at providing and supporting analog CAD experts with a hierarchical representation of this knowledge which is especially important for analog design automation. This will be illustrated by applications to circuit sizing, response surface modeling and analog placement.

There are essential differences of the presented method with respect to the state of art including analog design literature [27, 28, 29, 30, 31]. We strictly distinguish between a *constructive* part and a *constraint* part of design knowledge. While a constructive part aims at creating equations to propagate circuit specifications top-down to parameter values, the constraint part characterizes bottom-up conditions that have to be satisfied for a manufacturable design. This constraint part will be gathered in form of sizing rules, which are established for basic building blocks of analog circuits on *transistor-pair level* (Fig. 1). This is an intermediate level between circuit and device level. As each analog circuit is based on transistor-pair building blocks like e.g. current mirrors or differential stages, this allows us to capture a major portion of design constraint knowledge in a *design-independent, technology-specific* manner. Sizing rules can be interpreted as an extension of the concept of design rules to the structural transistor level: While design rules are formulated on the layout level as constraints for line widths and interlayer registration, sizing rules are formulated on the structural level as constraints for transistor geometry parameters (width, length, area) and for electrical transistor quantities (e.g. transistor drain/source voltage). Both types of rules represent an

Level	Example	
System	DAC, ADC, PLL	design-specific
Circuit	OpAmp	
Transistor Pair	**Current Mirror**, **Differential Stage**	**technology-specific**
Device	Transistor	

Figure 1: Levels of Abstraction in Analog Design

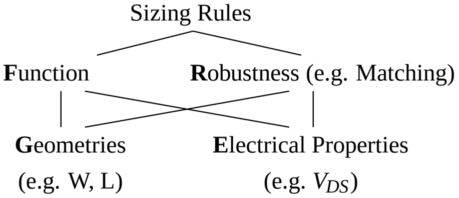


Figure 2: Types of Sizing Rules

optimal compromise between design performance (e.g. area) and process yield. In particular, sizing rules guarantee the dedicated function of a building block and its robustness e.g. towards mismatch or channel length modulation. (Fig. 2).

The remainder of this paper is organized as follows: Sec. 2 develops a hierarchical library of basic building blocks on transistor-pair level. A procedure for automatic hierarchical recognition of these building blocks is presented in Sec. 3. In Sec. 4, sizing rules for building blocks according to functional and robustness constraints are presented. Sec. 3 and 4 include experimental results for three operational amplifiers ranging from about 20 to 70 transistors. In Sec. 5, applications of sizing rules to circuit sizing, response surface modeling and analog placement are presented, and Sec. 6 concludes the paper.

2. HIERARCHICAL BUILDING BLOCK LIBRARY

Fig. 3 shows a list of basic building blocks for CMOS technology.[1] It can be seen that building blocks are simultaneously viewed on the behavioral (functional) and the structural level. A hierarchy of building blocks results from the structural property that basic functions are realized based on transistor pairs and groups of transistor pairs. On the lowest hierarchy level 0, transistors are mainly used for two functions, i.e. as voltage-controlled current sources and as voltage-controlled resistors. On level 1, we have identified 7 transistor pair structures (Fig. 3) that uniquely correspond to basic functions. This has been done by first creating all $B_6 = 206$ possible transistor pair structures, where B_k is the Bell number [32]. After discarding all symmetric structures and all technically senseless structures (no connection between transistors, drain-source or gate-source connection of one transistor, crosswise drain-source connection between transistors), the given 7 transistor pairs remain, plus 2 pairs representing parts of a current-mirror bank or a level-shifter bank, plus 11 transistor pairs that do not correspond to a functional relationship. In so far, the list of pairs on level 1 is complete.

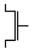
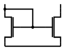
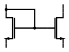
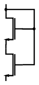
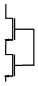
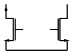
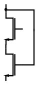
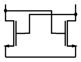
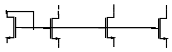
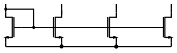
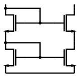
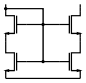
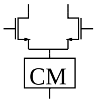
Function	Schematic NMOS (PMOS do.)	Hierarchical Level
Voltage-Controlled Resistor (vcres) Volt.-Contr. Current Source (vccs)		0
Simple Current Mirror (cm)		1
Level Shifter (ls)		
Voltage Reference 1 (vr1)		
Current Mirror Load (cml)		
Differential Pair (dp)		
Voltage Reference 2 (vr2)		
Flip-Flop (ff)		
Level Shifter Bank (LSB)		2
Current Mirror Bank (CMB)		
Cascode Current Mirror (CCM)		
4-Transistor Current Mirror (4TCM)		
Differential Stage (CM=cm/CCM/4TCM)		3

Figure 3: Library of Basic Building Blocks

On level 2, we have identified 4 different ”pairs of transistor pairs” as major building blocks. These are: level shifter bank, current mirror bank, cascode current mirror (CCM), and 4-transistor current mirror (4TCM). A CCM can be modeled as a combination of a simple current mirror (cm) and a level shifter (ls), a 4TCM can be modeled as a combination of a voltage reference 1 (vr1) and a current mirror load (cml). This modeling enables the automatic recognition within a circuit structure and upon this the assignment of sizing rules.

On level 3, a differential stage is modeled as a combination of a differential pair (dp) and a generic current mirror (CM). A CM in turn is implemented as a simple current mirror (cm) or a cascode current mirror (CCM) or a 4-transistor current mirror (4TCM). Hence three different types of differential stages with 2 or 3 transistor pairs can be recognized.

Of course, this library is not complete for levels above 1, and a variety of other building blocks can be included. But it represents a majority of used building blocks and can be considered as a standard building block library (which could be used as a library for schematic entry during topology design). Here we intend to use it to introduce sizing rules to the sizing part of analog synthesis. For that, we need a generic list of constraints for each building block that result from function and robustness requirements (i.e.

[1] The presentation concentrates on CMOS as a mainstream technology. The principles hold e.g. for bipolar as well. Block schematics and sizing rules are given for the NMOS part and hold analogously for the PMOS part.

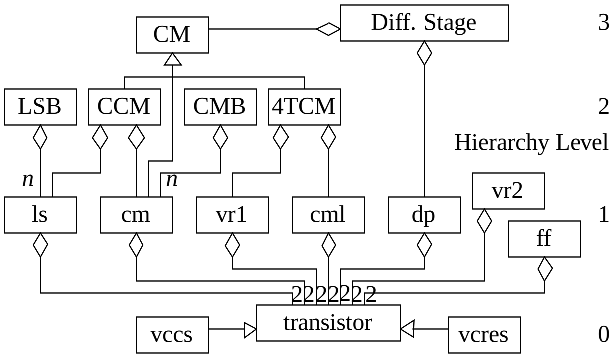


Figure 4: Hierarchy of Building Blocks

a generic list of sizing rules). Sizing rules will be equalities and inequalities of transistor geometries and DC quantities and can be checked during simulation-based analog synthesis without simulation overhead. Additionally, we provide a procedure that searches a given circuit schematic to identify all library elements. For each identified library element, the actual list of sizing rules is instantiated. Even for small circuits, a large number of library elements will result, which in turn leads to a large number of actual sizing rules.

In the following, an automatic recognition procedure of building blocks is presented first, and afterwards, the library of generic sizing rules is introduced.

3. AUTOMATIC HIERARCHICAL RECOGNITION OF BUILDING BLOCKS

Fig. 4 shows the hierarchical library of the building blocks from Fig. 3 in UML notation [33]. For instance it reads: a differential stage ”consists of” a current mirror (CM) and a differential pair (dp); a cascode current mirror (CCM) ”is a” a current mirror (CM), a simple current mirror (cm) ”is a” a current mirror (CM); a level shifter bank (LSB) ”consists of” *n* level shifters (ls).

This library representation describes a closed set of circuit components in a hierarchical manner, where each component consists of either a single transistor, a transistor pair, or transistor pair groups. The task of the recognition procedure now is to find all components in a given circuit schematic that correspond to the given set of components in the library, going through the hierarchy from simple components to more and more complex ones. Formally this is a search of isomorphic subgraphs [34], where the subgraphs correspond to the schematics of the component set defined by Fig. 4.

The pairwise block-building in our component library reduces the complexity of the probem so that we can afford to introduce an exhaustive search, which is illustrated in Fig. 5. In practice, the computational cost of this procedure can be neglected.

In a first step, all pairs of existing components, i.e. components that are connected at least at one pin, are listed. The two nested loops realize the exhaustive search for component pairs that correspond to building blocks in the library (Figs. 4 and 3). For each detected building block, a new component is instantiated. These steps are repeated until no more new components of the library are detected.

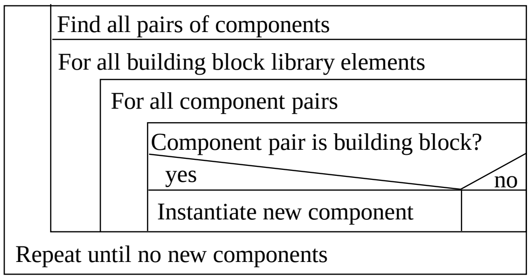

Find all pairs of components
For all building block library elements
For all component pairs
Component pair is building block?
yes
no
Instantiate new component
Repeat until no new components

Figure 5: Building Block Recognition Procedure

The procedure leads to a bottom-up recognition of components according to the hierarchical levels of the library. In the first run of the outer loop, the available components are transistors, and hence all transistor pairs from the hierarchical level 1 (Fig. 3) will be detected and instantiated as new components. In the second run, the available components are transistors and transistor pairs, and hence all pairs of transistor pairs from the hierarchical level 2 will be detected and instantiated as new components. Finally, in the third run, the 3 different types of differential stages (Fig. 4) will be detected.

Please note that the procedure in Fig. 5 holds for any other hierarchy of building blocks than that in Fig. 4. For detection of potential component pairs that correspond to a library element, it is necessary to define all pin connections of the library element, those connected and those not connected.

3.1 Results

In Figs. 6, 7, and 8, the schematics of three different circuits are given, that have been analyzed by automatic building block recognition. Detected components have been shaded.

Fig. 9 summarizes the number of detected components of the different levels of hierarchy. These results are obtained automatically. The automatic recognition algorithm is implemented in JAVA. It is worth mentioning that the implemented program automatically parameterizes the circuit components and formulates and assigns the sizing rules in order to prepare analog synthesis tasks like circuit sizing. In the following section, the generic list of sizing rules for the building block library will be presented.

4. SIZING RULES

For each building block according to Fig. 3 a set of sizing rules will be given. These rules result from constraints guaranteeing the dedicated function and its robustness e.g. towards mismatch or channel length modulation. These constraints refer to transistor geometry parameters (width, length, area) and electrical transistor quantities (e.g. transistor drain/source voltage) according to Fig. 2. In correspondance to Fig. 2, a rule will be labeled with *FG* or *FE* if it is a **g**eometric or **e**lectrical constraint concerning **f**unction, and with *RG* or *RE* if it is a **g**eometric or **e**lectrical constraint concerning **r**obustness.

Note that the hierarchical library representation allows a redundance-free storage of generic sizing rules. Since each building block on level *i* consists of building blocks from lower levels of hierarchy, just the additional constraints that arise on level *i* have to be stored. Only after automatic recognition in a real circuit schematic, the

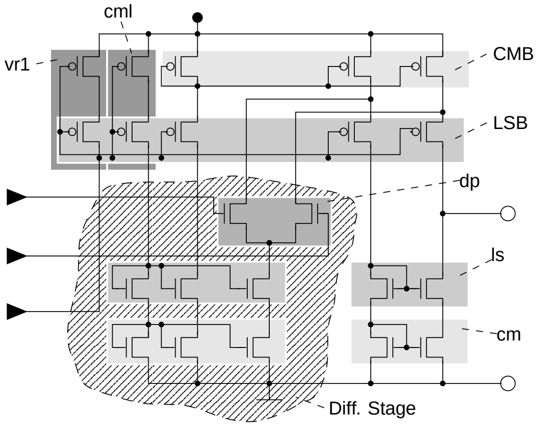


Figure 6: Detected Building Blocks of Folded-Cascode OpAmp

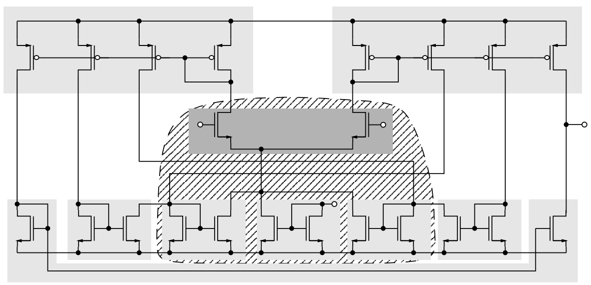


Figure 7: Detected Building Blocks of OpAmp with Dynamic Bias Current

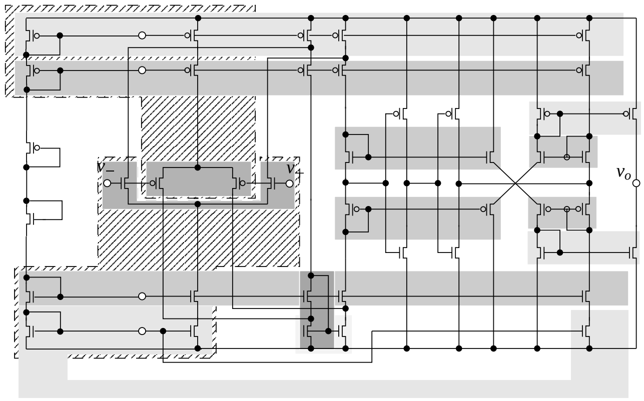


Figure 8: Detected Building Blocks of Output Driver

	level of hierarchy				
Circuit	0	1	2	3	Total
Fig. 6	22	15	8	1	46
Fig. 7	22	13	2	1	38
Fig. 8	38	24	10	2	74

Figure 9: Number of Detected Building Blocks on different levels of hierarchy

sizing rules for each identified building block are instantiated from the generic sizing rules for that block and for all included building blocks from lower hierarchy levels.

4.1 Building Blocks on Hierarchy Level 0

4.1.1 Voltage Controlled Current Source (vccs)

A transistor working as a ”vccs” has to operate in saturation, hence:

$FE:\quad v_{ds} - (v_{gs} - V_{th}) \geq V_{sat_{min}};\ v_{ds} \geq 0;\ v_{gs} - V_{th} \geq 0$

From [35] and the Shichman-Hodges transistor model follows that the drain-source current variation depends on variations of channel width and length, threshold voltage, electron mobility and specific gate oxide capacitance with factors $1/w^2$, $1/l^2$ and $1/(w \cdot l)$. Additionally, $1/f$ noise is also proportional to $1/(w \cdot l)$. Hence for robustness we require:

$RG:\quad w \cdot l \geq A_{min};\ w \geq W_{min};\ l \geq L_{min}$

The constants $V_{sat_{min}}$, A_{min}, W_{min}, L_{min}, as well as the constants introduced in the following, are technology-specific constants, that have to be determined only once for each technology.

4.1.2 Voltage Controlled Resistor (vcres)

A transistor as a ”vcres” operates in the triode region, hence:

$FE:\quad v_{gs} - V_{th} - v_{ds} \geq V_{trio_{min}};\ v_{ds} \geq 0;\ v_{gs} - V_{th} \geq 0$

4.2 Building Blocks on Hierarchy Level 1

4.2.1 Simple Current Mirror (cm)

The function of a ”cm” is a certain factor K between the two drain-source currents. To keep the influence of the drain-source voltage low, both transistors are ”vccs”. To avoid systematic mismatch, the drain-source voltage difference needs to be small and the transistor lengths should be equal, hence:

$FG: l_1 = l_2;\ K = w_2/w_1 \qquad FE: |v_{ds2} - v_{ds1}| \leq \Delta v_{ds_{max}}$

To avoid mismatch due to local process variations, the effective gate-source voltage has to be large enough, hence:

$RE:\quad v_{gs1,2} - V_{th1,2} \geq V_{min1,2}$

4.2.2 Level Shifter (ls)

A common function of a ”ls” is to equalize the source potentials of its transistors. This is obtained by utilizing the quadratic dependence of the drain-source current from the gate-source voltage (i.e. both transistors are ”vccs”) and the following functional constraints:

$FG:\ l_1 = l_2 \qquad FE:\ |i_{ds2}/i_{ds1} - w_2/w_1| \leq K_{iW_{max}}$

Additionally, *RE* of a ”cm” holds.

4.2.3 Voltage Reference 1 (vr1), Current Mirror Load (cml)

”vr1” and ”cml” are building blocks for a 4-Transistor Current Mirror. The lower transistors are ”vcres”, the upper transistors are ”vccs”.

4.2.4 Difference Pair (dp)

A ”dp” transforms a gate-source voltage difference into a drain-source current difference. To reduce the effect of drain-source voltage differences, both transistors work as ”vccs”. *FG* and *FE* of

Circuit	#Equalities	#Inequalities	Total
Fig. 6	31	154	185
Fig. 7	30	158	188
Fig. 8	57	274	331

Figure 10: Number of Sizing Rules for Circuit Examples

a ”cm” hold analogously for the ”dp”, with $K = 1$ for symmetry reasons. To reduce the effect of local variations in threshold voltages on the drain-source current difference, large amplification should be avioded. This means that the effective gate-source voltage should not exceed a certain value:

RE : $$v_{gs1,2} - V_{th1,2} \leq V_{max1,2}$$

4.2.5 Voltage Reference 2 (vr2)

A ”vr2” can be used instead of the lower left transistor of a level shifter or a cascode current mirror (CCM). The lower transistor is a ”vccs” and inherits the sizing rules of the replaced transistor of the ”ls” or ”CCM”. The gate connection of the upper transistor is circuit-dependent.

4.2.6 Flip Flop (ff)

For this simple memory with two transistors, symmetrical on and off behavior requires:

FG : $$l_1 = l_2 \ ; \ \ w_1 = w_2$$

4.3 Building Blocks on Hierarchy Level 2

4.3.1 Level Shifter Bank (LSB) and Current Mirror Bank (CMB)

”CMB” and ”LSB” are obtained by detecting and merging m current mirrors (level shifters) with identical reference transistors. The resulting m-CMB (m-LSB) inherits the sizing rules of the individual current mirrors (level shifters).

4.3.2 Cascode Current Mirror (CCM)

A ”CCM” is a ”cm” together with a ”ls” to reduce the influence of the drain-source voltage of the driven transistor on the current ratio (i.e. to increase the output impedance of the ”cm”). For symmetry reason, we have:

FG : $$w_{1,\text{”}ls\text{”}} = w_{1,\text{”}cm\text{”}} \ ; \ \ w_{2,\text{”}ls\text{”}} = w_{2,\text{”}cm\text{”}}$$

4.3.3 4-Transistor Current Mirror (4TCM)

A ”4TCM” consists of a ”vr1” and a ”cml”.

It has the same advantage over a ”cm” as the ”CCM” and the additional advantage of a lower drain-source voltage drop, which is important for small supply voltages.

The 4 sizing rules of a ”cm” have to be satisfied by the two upper transistors. Additionally, for the transistor widths the symmetry requirements of the ”CCM” hold, hence:

FG : $$w_{1,\text{”}vr1\text{”}} = w_{2,\text{”}vr1\text{”}} \ ; \ \ w_{1,\text{”}cml\text{”}} = w_{2,\text{”}cml\text{”}}$$

4.4 Results

Although the list of 30 generic sizing rules is fairly small, expansion according to the building block detection leads to a quite large number of sizing rules for each circuit. Fig. 10 shows that in total several hundred sizing rules are established for each of the three circuit examples. It can easily be seen that automatic construction of these rules on circuit level, as presented here, is inevitable even for small circuits.

Performance	Target	Initial	After Synthesis	
	Value	Value	WITH	WITHOUT
			Sizing Rules	
A_0 [dB]	75	99	74.7	75.5
f_T [MHz]	54	7.9	54.1	53.9
PHM [^{o}]	70	34.0	69.9	69.9
SR_p [V/μs]	42	4.4	41.9	42.6
Pow_{DC} [mW]	1.7	0.36	1.71	1.71
Cost in #Perf./Gradient Calc.			45	126
#Violated Sizing Rules		6	0	7
Operating Range Violation		yes	no	yes

Figure 11: Comparison of Automatic Sizing Results with/without sizing rules

Please note that the small number of generic sizing rules is a result of the presented hierarchical building block library on transistor pair level. *If the generic rules would be established on circuit level, the preparatory effort for analog synthesis would be significantly higher.*

The inequality part of sizing rules has to be satisfied during the design process, e.g. sizing and design centering. The equality part of sizing rules leads to a reduction of the complexity of the design process, because it reduces the number of free design parameters. Both together enable a reliable design process, as will be shown in the following applications.

5. APPLICATIONS

5.1 Circuit Sizing

With the example of the folded-cascode OpAmp in Fig. 6, it will be illustrated that sizing rules are essential for automatic circuit sizing. We have investigated many other circuits with the same result. The used optimization method is a state-of-art simulation-based, deterministic trust-region algorithm [36]. Circuit sizing here is meant to tune circuit parameters (transistor geometries) in order to achieve certain performance targets.

Figure 11 shows five performances of the OpAmp and the corresponding initial values, target values, and the optimized values after automatic sizing with and without sizing rules. Concerning the optimized values, using sizing rules made no difference.

But looking more closely at the results, there are essential differences in both efficiency and effectiveness. First, it took three times more performance/gradient calculations (which approximately equals to the total computational cost) when sizing rules were not used.

Second, although not using sizing rules in this case leads to the same optimized performance values than using sizing rules, it leads to a design where 7 non-trivial sizing rules are violated. As a consequence, checking the design for the operating range reveals that the circuit does not satisfy the specification for the whole operating range. This is illustrated in Fig. 12. It can be seen that sizing rules ensure that automatic sizing leads to a robust design with respect to operating tolerances.

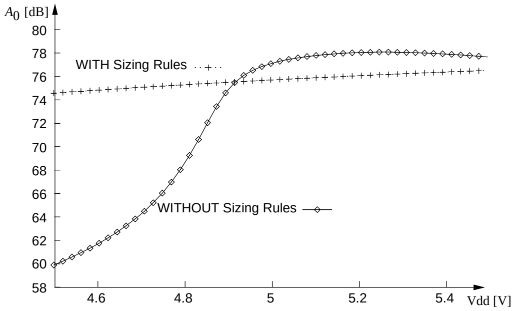


Figure 12: Performance A_0 over operating range

Performance	Specification	Initial Value	After Design Centering WITH	 WITHOUT
			Sizing Rules	
A_0 [dB]	> 40	50.7	57.7	59.4
f_T [MHz]	> 40	37.7	44.2	45.8
$CMRR$ [dB]	> 80	78.1	92.8	77.7
SR_p [V/μs]	> 35	35.2	36.6	38.6
Pow_{DC} [mW]	< 3.5	2.9	2.9	3.4
Circuit Yield		0%	100%	0%

Figure 13: Comparison of Automatic Design Centering Results with/without sizing rules

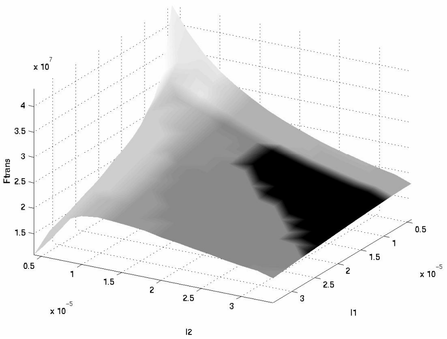


Figure 14: Transit frequency over two transistor lengths. The brighter the color gets, the more sizing rules are violated.

	Analytical modeling WITH	 WITHOUT
	Sizing Rules	
max. error [dB]	1.4	5.4
mean abs. error [dB]	0.2	0.7

Figure 15: Comparison of Response Surface Modeling Result for OpAmp Gain with/without sizing rules

5.2 Design Centering

A similar result is obtained for design centering. The task of design centering is to tune circuit parameters in order to maximize the parametric yield (i.e. percentage of circuits satisfying specified performance bounds) with respect to manufacturing tolerances. Fig. 13 shows performances and specified boundaries for the operational amplifier in Fig. 6, and design centering results with and without using sizing rules. The underlying design centering method is a state-of-art algorithm based on piece-wise linear approximations of the acceptance region [37] Fig. 13 shows that the algorithm completely failed when not using sizing rules. The reason is that *whenever sizing rules are satisified, the circuit performance behavior is not far from linear.* Therefore, sizing rules provide a trust-region for the linearized problem within each iteration step and effectively direct the search.

5.3 Response Surface Modeling

Response surface modeling (RSM) serves to replace performance evaluation by computationally expensive circuit simulation models with cheaper performance evaluation by analytical functions. In practical applications, RSM has to select basis functions for the analytical model, select test points where the ”true” circuit is evaluated, and compute the coefficients of the analytical model. Another problem is the definition region of the analytical model.

Sizing rules contribute to RSM in several ways.

First, they provide an accurate and technically relevant definition region of an analytical model: it does not make sense to model a circuit outside the region within which function and robustness are guaranteed and within which design moves take place. Fig. 14 shows a typical performance behavior over design parameters. In the black region all sizing rules are satisfied. That is the definition region where a certain accuracy of an analytical model is guaranteed.

The second effect that can be seen in Fig. 14 is that the region where test points have to be simulated is much smaller than the original region defined by simple box constraints. Considering the exponential problem complexity with respect to the number of model parameters, this results in a significant saving in simulation cost for model construction.

Third, Fig. 14 shows that the performance behavior is near to linear in the region where sizing rules are satisfied. This results in an increased accuracy of the analytical models. Fig. 15 illustrates the model quality of the gain of the folded-cascode OpAmp. The modeling method was a state-of-art algorithm with combined linear and radial basis functions and adaptive volume slicing for test point generation [38]. We can see that the model quality increases by a factor of 3 when using sizing rules. Similar results are obtained for other performances and circuits.

5.4 Analog Placement

The library component recognition part of the sizing rules method has been applied in order to automatically provide placement constraints for analog cells. This enables analog layout designers to create the necessary placement specifications for analog cells starting just from the circuit schematic.

6. CONCLUSIONS

In this paper a method for capturing design knowledge on transistor-pair level has been developed. It consists of:

- a generic hierarchical library of 14 important CMOS building blocks on transistor-pair level (Sec. 2),
- a generic list of 30 sizing rules for geometric or electrical constraints concerning function or robustness for these building blocks (Sec. 4), and
- an automatic recognition of building blocks in a circuit schematic with subsequent parameterization of simulation and synthesis input files (Sec. 3).

The results show that the method captures important design knowledge very efficiently. The threshold values in the generic sizing rules have to be given just once for a technology. The method then expands the generic lists of building blocks and sizing rules for each considered circuit individually and automatically as a preparation for the design. The large number of sizing rules of a circuit (several hundred) clearly shows that these cannot be established by hand for each circuit. The significance of the presented method follows from applications to circuit sizing, design centering, response surface modeling and analog placement. Sizing rules especially make sure that automatic circuit sizing and design centering lead to technically meaningful and robust results. Moreover, sizing rules improve response surface modeling and contribute to the placement of analog circuits.

7. REFERENCES

[1] M. Degrauwe et al, "IDAC: An interactive design tool for analog CMOS circuits," *IEEE J. SC*, 1987.

[2] R. Harjani, R. Rutenbar, L. Carley, "OASYS: A framework for analog circuit synthesis," *IEEE Trans. CAD*, 1989.

[3] H. Koh, C. Sequin, P. Gray, "OPASYN: A compiler for CMOS operational amplifiers," *IEEE Trans. CAD*, 1990.

[4] J. Harvey, M. Elmasry, B. Leung, "STAIC: An interactive framework for synthesizing CMOS and BiCMOS analog circuits," *IEEE Trans. CAD*, 1992.

[5] R. Brayton, G. Hachtel, A. Sangiovanni-Vincentelli, "A survey of optimization techniques for integrated-circuit design," *Proc. IEEE*, 1981.

[6] M. Lightner, T. Trick, R. Zug, "Circuit optimization and design," *Circuit Analysis, Simulation and Design, 2 (A. Ruehli). Advances in CAD for VLSI 3*, 1987.

[7] W. Nye, D. Riley, A. Sangiovanni-Vincentelli, A. Tits, "DELIGHT.SPICE: An optimization-based system for the design of integrated circuits," *IEEE Trans. CAD*, 1988.

[8] S. Director, G. Hachtel, "The simplicial approximation approach to design centering," *IEEE Trans. CAS*, 1977.

[9] K. Antreich, R. Koblitz, "Design centering by yield prediction," *IEEE Trans. CAS*, 1982.

[10] R. Spence, R. Soin, *Tolerance Design of Electronic Circuits*, Addison-Wesley, England, 1988.

[11] S. Director, W. Maly, A. Strojwas, *VLSI Design for Manufacturing: Yield Enhancement*, Kluwer Academic Publishers, USA, 1990.

[12] P. Feldmann, S. Director, "Integrated circuit quality optimization using surface integrals," *IEEE Trans. CAD*, 1993.

[13] J. Zhang, M. Styblinski, *Yield and Variability Optimization of Integrated Circuits*, Kluwer Academic Publishers, 1995.

[14] M. del Mar Hershenson, S. Boyd, T. Lee, "Optimal design of a CMOS Op-Amp via geometric programming," *IEEE Trans. CAD*, 2001.

[15] G. Gielen, G. Debyser, K. Lampaert, F. Leyn, K. Swings, G. Van Der Plas, W. Sansen, D. Leenaerts, P. Veselinovic, W. Van Bokhoven, "An analogue module generator for mixed analogue/digital ASIC design," *Int. J. Circuit Theory and Applications*, 1995.

[16] G. Gielen, H. Walscharts, W. Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing," *IEEE J. SC*, 1990.

[17] E. Ochotta, R. Rutenbar, L. Carley, "Synthesis of high-performance analog circuits in ASTRX/OBLX," *IEEE Trans. CAD*, 1996.

[18] E. Ochotta, T. Mukherjee, R. Rutenbar, L. Carley, *Practical Synthesis of High-Performance Analog Circuits*, Kluwer Academic Publishers, 1998.

[19] K. Antreich, H. Graeb, C. Wieser, "Circuit analysis and optimization driven by worst-case distances," *IEEE Trans. CAD*, 1994.

[20] A. Dharchoudhury, S. Kang, "Worst-case analysis and optimization of VLSI circuit performances," *IEEE Trans. CAD*, 1995.

[21] M. del Mar Hershenson, S. Boyd, T. Lee, "GPCAD: A tool for CMOS op-amp synthesis," in *IEEE/ACM ICCAD*, 1998.

[22] G. Debyser, G. Gielen, "Efficient analog circuit synthesis with simultaneous yield and robustness optimization," in *IEEE/ACM ICCAD*, 1998.

[23] T. Mukherjee, L. Carley, R. Rutenbar, "Efficient handling of operating range and manufacturing line variations in analog cell synthesis," *IEEE Trans. CAD*, 2000.

[24] R. Phelps, M. Krasnicki, R. Rutenbar, L. Carley, J. Hellums, "Anaconda: Simulation-based synthesis of analog circuits via stochastic pattern search," *IEEE Trans. CAD*, 2000.

[25] K. Antreich, J. Eckmueller, H. Graeb, M. Pronath, F. Schenkel, R. Schwencker, S. Zizala, "WiCkeD: Analog circuit synthesis incorporating mismatch," in *IEEE CICC*, 2000.

[26] P. Mandal, V. Visvanathan, "CMOS 0p-Amp sizing using a geometric programming formulation," *IEEE Trans. CAD*, 2001.

[27] D. Johns, K. Martin, *Analog Integrated Circuit Design*, John Wiley & Sons, 1997.

[28] K. Laker, W. Sansen, *Design of Analog Integrated Circuits and Systems*, McGraw-Hill, New York, 1994.

[29] M. Ismail, T. Fiez, *Analog VLSI Signal and Information Processing*, Electrical and Computer Engineering. McGraw–Hill, 1994.

[30] P. Gray, R. Meyer, *Analysis and Design of Analog Integrated Circuits*, John Wiley & Sons, 3 edition, 1993.

[31] P. Allen, D. Holberg, *CMOS Analog Circuit Design*, Oxford University Press, 1987.

[32] H. Mattson, *Discrete Mathematics with Applications*, John Wiley & Sons, Inc., 1993.

[33] Object Management Group, *OMG Unified Modeling Language Specification*, Object Management Group, Inc., 1999.

[34] G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw–Hill, Inc., New York, 1994.

[35] K. Lakshmikumar, R. Hadaway, M. Copeland, "Characterization and modeling of mismatch in MOS transistors for precision analog design," *IEEE J. SC*, 1986.

[36] R. Schwencker, J. Eckmueller, H. Graeb, K. Antreich, "Automating the sizing of analog CMOS-circuits by consideration of structural constraints," in *DATE*, 1999.

[37] F. Schenkel, M. Pronath, S. Zizala, R. Schwencker, H. Graeb, K. Antreich, "Mismatch analysis and direct yield optimization by spec-wise linearization and feasibility-guided search," in *ACM/IEEE DAC*, 2001.

[38] S. Zizala, J. Eckmueller, H. Graeb, "Fast calculation of analog circuits' feasibility regions by low level functional measures," in *IEEE ICECS*, 1998.