A Generic Topology Selection Method for Analog Circuits with Embedded Circuit Sizing Demonstrated on the OTA Example

Andreas Gerlach, Juergen Scheible
Robert Bosch Centre for Power Electronics
Oferdingerstr. 50
72768 Reutlingen, Germany
andreas.gerlach@reutlingen-university.de
juergen.scheible@reutlingen-university.de

Thoralf Rosahl, Frank-Thomas Eitrich
Robert Bosch GmbH
Tuebinger Strasse 123
72762 Reutlingen, Germany
thoralf.rosahl@de.bosch.com
frank-thomas.eitrich@de.bosch.com

***Abstract*—We present a new methodology for automatic selection and sizing of analog circuits demonstrated on the OTA circuit class. The methodology consists of two steps: a generic topology selection method supported by a "part-sizing" process and subsequent final sizing. The circuit topologies provided by a reuse library are classified in a topology tree. The appropriate topology is selected by traversing the topology tree starting at the root node. The decision at each node is gained from the result of the part-sizing, which is in fact a node-specific set of simulations. The final sizing is a simulation-based optimization. We significantly reduce the overall simulation effort compared to a classical simulation-based optimization by combining the topology selection with the part-sizing process in the selection loop. The result is an interactive user friendly system, which eases the analog designer's work significantly when compared to typical industrial practice in analog circuit design. The topology selection method and sizing process are implemented as a tool into a typical analog design environment. The design productivity improvement achievable by our method is shown by a comparison to other design automation approaches.**

I. Introduction

Design automation for analog integrated circuits is still immature compared to the highly automated synthesis flow for digital circuits. We observe a severe bottleneck in analog VLSI design; the design of the analog parts of a mixed signal chip project is usually more time consuming than the design of the digital parts [14], [15], even when the analog parts contain far less components. This shows the need for new and better analog design methods. Analog circuit design requires two steps: topology creation and subsequent sizing. Analog synthesis methods are typically (1) methods for topology selection or synthesis methods and (2) automated sizing methods [1]. Many attempts have been made in the past to improve the analog design process. Approaches in the last century were based on pool selection (e.g. IDAC [12]), heuristics (e.g. OASYS [4], OPASYN [5]) or genetic algorithms (e.g. DARWIN [6]), but saw little acceptance by industrial designers. Recent approaches have focused on determining sizing rules which lead to automatically sized and centered, technically meaningful and robust designs (e.g. MARS [2]), detecting matching conditions for sizing using symmetry computation (where there are multiple symmetry axes as well: Sizing Rules [8]), structural synthesis of analog circuits based on hierarchically-organized building blocks (e.g. MOJITO [9]) or the intentional utilization of existing design knowledge (e.g. FEATS [10]).

II. Analog Reuse of Suitable OTA Circuits

Our initial industrial chip investigation showed, that the transconductance amplifier (OTA) belongs to the most widely used circuit classes with a large number of topology variants [3]. We extracted the most commonly used topology variants in the OTA circuit class. These are differential amplifiers (the simplest OTAs) with p-MOS-input transistors and symmetrical OTAs with p-MOS input transistors. We found some simple OTA topologies that can be reused fix-dimensioned within the same technology. Based on interviews with analog designers and findings of our chip investigation, we extracted two dozen OTA-topologies with a high expected reuse rate and integrated them in our reuse library. Currently, we restrict our OTA reuse library to topology sizes of about 30 transistors in order to keep it manageable. Based on the chip investigation, we anticipate that the reuse library will cover 80 % of the requirements commonly arising in practice.

III. Topology Selection Method

Different topologies in one circuit class are ordered in a tree structure into a reuse library [3]. Enhancements in analog circuit topologies – resulting in higher circuit complexity – arise out of the circuit development process. Each circuit enhancement, which typically is created by introducing additional basic building blocks such as current mirrors or the like, results in a new child (next complexity level) in the tree structure. Only topologies with significant enhancements are selected to build the tree. For selecting an appropriate topology the topologies are checked one by one to see if they meet the specified output values. Checking starts with the simplest circuits (parents in the tree structure) and goes on to increasingly more complicated structures (children). Yes-no questions are defined for navigation through the tree structure. These are questions a designer would ask, if he had to choose a topology for solving a problem. These are related to the output parameters, that are characteristic for the enhancement, and that make the relevant difference between the architectures in a circuit class. We have set the questions by comparing the presented circuit topologies. These questions can be answered by simulations or by solving equations [3]. We answer them in our new method with a "part-sizing" process and simulations.

While moving through the tree structure some topologies are excluded without the necessity of being checked. This saves computing power, and consequently time. We expect designers to embrace this method as it closely resembles the way they choose topologies to solve their problems. Fig. 1 shows a simplified flowchart of our method.

IV. Automatic Sizing Process

The sizing process in our method is based on scripts, which make use of a universal test bench customized for this purpose with script-based equations and Cadence's "global optimization"

978-3-9815370-8-6/17/$31.00

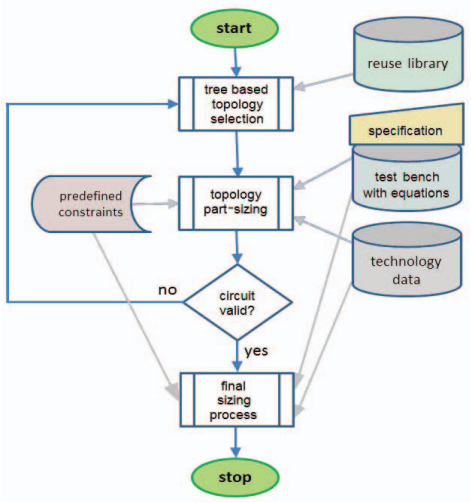

Fig. 1. Circuit topology selection method

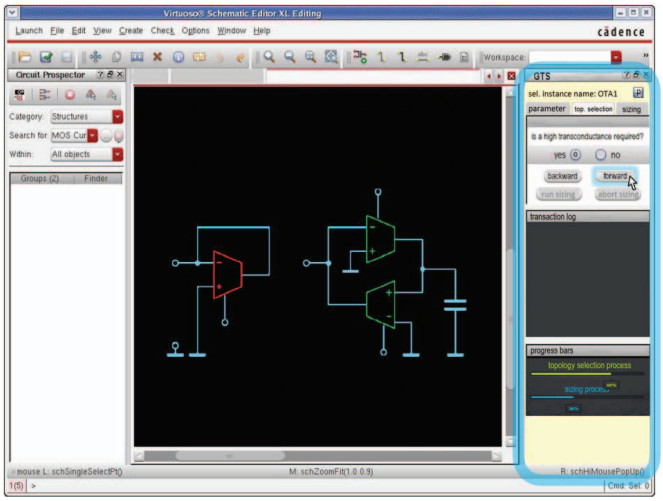

Fig. 2. Our tool implemented as a Cadence Assistant

technology. The simulations deliver technology-specific results, using a technology-independent test bench with technology-independent equations. Simulation-based results are more accurate than simplified algebraic equations prepared for this purpose. With these scripts we check if there are transistor parameters which, when placed into our topologies, yield the desired results. Test cases for universal checking of all possible OTA topologies are defined.

Eleven performance parameters can be regarded: transconductance, input voltage range, offset voltage (systematic and statistical), noise, max. power consumption, max. output current, voltage gain and output resistance, frequency response, common mode rejection ratio (CMRR) and power supply rejection ratio (PSRR). Technology-specific values of transistor parameters are considered through the pdk data in the sizing process in simulations. This causes the choice of the circuit topology and the sizing process to be specific for the currently used technology.

Having observed the day-to-day sizing work done by designers, we came to the conclusion that it is almost impossible to avoid iterations during the complete sizing process. If an output performance parameter is optimized, the others are usually affected as well. This happens because of the correlations between the performance parameters of the circuit and the transistor parameters. Thus transistor parameters which have already been set, have to be readjusted more than once. We use an iterative algorithm to fit the performance parameters to the specified values. The sizing process is a series of checking and optimization operations in a loop as shown in Fig. 3.

V. Implementation in an Industrial Design Flow

The tool based on our method is implemented in the Cadence DFII 6.1.6 Design Environment. It is programmed in the Skill programming language, as a Cadence Assistant (see Fig. 2). It consists of three sections implemented in the GUI as index tabs. The first index tab is a section with a table where a designer puts in the design specification by setting the desired values of the performance parameters. The second index tab is the section which navigates him through the topology tree to find the right architecture for his task. With the third index tab he can start the entire sizing process for the chosen architecture.

We implemented the tool into an industrial analog design flow. The tool supports the analog designer's typical activities. The flow

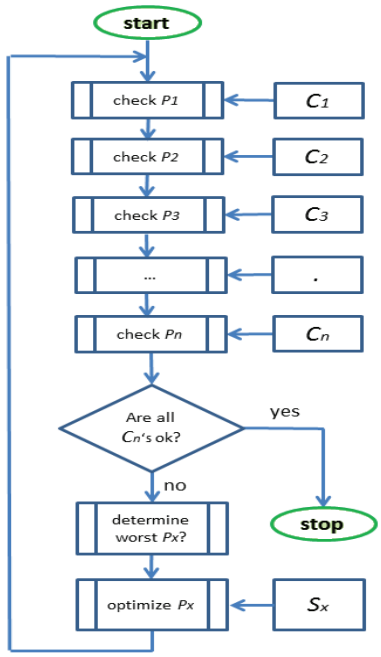

Fig. 3. Automatic sizing process: P_n is a performance parameter. C_n is a circuit checking procedure. S_x is a sizing step (sizing of the chosen circuit for one performance parameter). Performance Parameters with logarithmic characteristic are converted into linear values for comparison (in the step "determine worst P_x"). The performance parameter with the highest relative deviation is sized in next step in the loop.

is as follows: At first the designer places generic OTA-symbols for ideal OTAs into the design. These are ideal voltage-controlled current sources. They are marked in red. Based on these ideal sources the designer is able to simulate his design on the system level. When the simulation at this hierarchical level produces satisfactory results, the designer starts replacing the generic OTAs by real circuits using our tool. To do this the designer has to mark a generic red OTA and start the program. Then the designer has to enter the OTA requirements in the table in the index tab, if he wants to use the automatic selection

option. He then starts the topology selection in the second index tab. He can cede the selection of an appropriate topology to the tool by using our generic topology selection method. When using this option a progress bar shows graphically the progress of the selection process. Instead of this, the designer can also choose the topology manually from a list, if he is sure to know the right architecture for the application. In both cases the corresponding circuit is automatically copied from the reuse library to the project library and the red OTA symbol, indicating an ideal OTA, changes automatically to green. Green colored OTA symbols denote real OTA circuits.

After topology selection and automatic sizing the designer can simulate again at the system level. The topology selection with subsequent automatic sizing is complete when the simulation of the system is successful.

VI. Comparison with Other Methods

Our tree selection method is faster than pool-selection methods, as it cuts computation time. Selecting a topology for an application from a pool of circuits requires checking every architecture and differentiating between those that pass the specification and those that fail. With our tree selection method [3] we have to check only a small subset of topologies contained in the reuse library. In our method all topologies of a circuit class are ordered in a tree structure. For selecting an appropriate topology the topologies are checked one by one to see if they meet the specified output values. Checking starts with the simplest circuits (parents in the tree structure) and goes on to increasingly more complex structures (children). Increasing complexity arises from specific circuit enhancements to meet increased requirements, typically by inserting additional building blocks, e.g. current mirrors or alike. Thus, the tree traversal can be finished, when the first topology is reached which meets the specification. Our reuse library of OTAs consists of 24 topologies. With our tree selection method the maximum number of topologies, which have to be checked within this pool, is 4.

Topology synthesis tools build completely new topologies based on algorithms or by combining building blocks. In contrast to this approaches in our method the architectures have already been used in many designs and therefore are widely tested and robust.

Comparing our sizing method with Cadence's "global optimization", it can be seen that our method is faster. For an illustration we compare the calculation effort of both approaches. The calculation effort of an algorithm is given by the runtime complexity $O(n)$, where the input n denotes the problem size. The problem of sizing a circuit is characterized by an n-dimensional solution space S, where n denotes the number of variables to be determined (typically widths and lengths of the transistors). When the solution is created by performing a number of simulations, the calculation effort can be written as

$$O(n) = i \cdot p^n \tag{1}$$

where p denotes the number of considered values for each variable, i.e. the distance between two neighboring variable values can be regarded as the "resolution" for the search in S. For simplification we assume in this formula, that p has the same value for all variables. The factor i denotes the number of simulations, which has to be performed for one point in S, i.e. one set of variable values.

In case of "global optimization" we can write

$$O_{go}(n_{go}) = i_{go} \cdot p^{n_{go}} \tag{2}$$

Because in this method the entire solution space S is investigated, n_{go} equals the number of all variables contained in the circuit. Thus, for a circuit with t transistors n_{go} can be up to $2t$, if all widths and lengths have to be varied independently. The "global optimization" approach uses intelligent techniques to navigate in S, thus reducing the number of performed simulations. Therefore i_{go} can be assumed to be significantly less than 1.

In our method we are reducing the solution space. The maximum effort for our method can be described approximately by

$$O_{gts}(n_{gts}) = i_{gts} \cdot i_{go} \cdot p^{n_{gts}} \tag{3}$$

In this equation i_{gts} is the number of iterations needed by the program to size the circuit. The number of simulation points p should have the same value, as in case of the "global optimization" for results with comparable quality. In our method n_{gts} is the highest number of variables needed to set one performance parameter. Our method is faster, because the value of n_{gts} is for more than one performance parameter always smaller than the value of n_{go}.

We made many attempts for sizing different OTAs with our tool. One of the OTAs was a symmetrical transonductance amplifier with p-mos inputs showed in Fig. 4. Specified values and actual values for one sizing process for this example are shown in Fig. 5. For this example we get a n_{go} = 6 when we are using "global optimization". With our method a n_{gts} = 3 is sufficient (only a reduced number of design variables has to be considered), but we are using i_{gts} = 3 iterations. We were working on a Linux system. Our processor is an Intel Xeon CPU E5-2640 with 2.5 GHz and 64 GB RAM. We have measured an optimization time of 11 min and 27 sec.

A comparison of our method with other methods and tools is shown in Fig. 6.

VII. Conclusion

We have presented a tool consisting of two procedures: selection of a circuit topology and a subsequent automatic sizing. Both procedures emulate an expert's best practice of designing analog circuits. Furthermore, both procedures significantly save design effort and computation time. In the tree-based topology selection only a small subset of topologies of a given pool has to be investigated. The automatic sizing method cuts computation time by dividing the simulation-based optimization into sequential optimizations steps. In every step only one performance parameter is optimized, and thus only a reduced number of design variables has to be considered. The architectures in our reuse library have been used in many designs and are "silicon proven". First tests with this tool were so satisfying that we expect an industrial use in 2017.

VIII. Future Research

We are actually working to prove the general convergence of our method.

References

[1] M. Barros, J. Guilherme, and N. Horta. *Analog Circuits and Systems Optimization Based on Evolutionary Computation Techniques*. Springer, 2010.

[2] M. Eick and H. E. Graeb. Mars: Matching-driven analog sizing. *Computer-Aided Design of Integrated Circuits and Systems*, 31, 2012.

[3] A. Gerlach, J. Scheible, T. Rosahl, and F.-T. Eitrich. A generic topology selection method for analog circuits demonstrated on the ota example. *IEEE-Prime 2015 Glasgow University*, pages 77 – 80, July 2015.

[4] R. Harjani. Oasys: a framework for analog circuit synthesis. *ASIC Seminar and Exhibit*, (2):P13 – 1/1–4, September 1989.

[5] H. Y. Koh, C. Sequin, and P. Gray. Opasyn: a compiler for cmos operational amplifiers. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions*, 9(2):113 – 125, February 1990.

[6] W. Kruiskamp and D. Leenaerts. Darwin: Cmos opamp synthesis by means of a genetic algorithm. *Design Automation, 1995*.

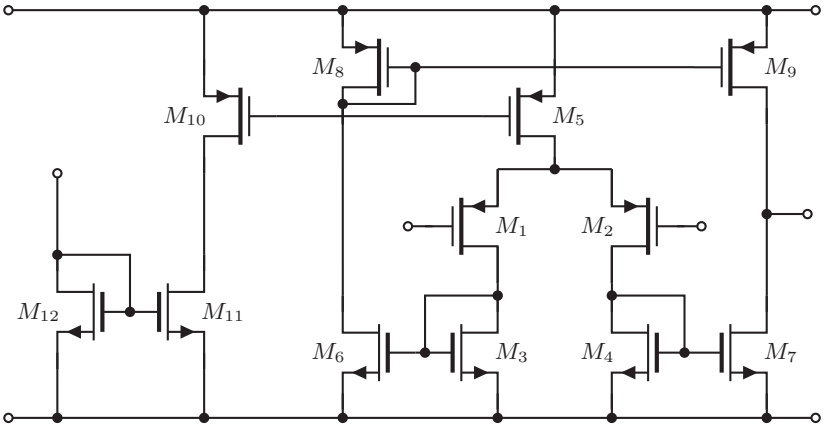

Fig. 4. Symmetrical transconductance amplifier with p-mos inputs; one of the circuits in our reuse library, which have been sized with our tool.

	set value	is value	deviation	description
g_m	200 mS	204 mS	2 %	gain
u_{off}	5 mV	4,89 mV	2,2 %	offset voltage
$I_{out,max}$	10 mA	9,95 mA	0,5 %	max. output current
$CMMR$	–	47,78 dB	–	common mode rejection ratio
$PSRR$	–	49,84 dB	–	power supply rejection ratio

Fig. 5. Example results of a sizing process for three chosen performance parameters for a symmetrical p-input OTA ($PSRR$ and $CMRR$ values are writen for information). The tolerance was set to 5% for linear performance parameters. The program needed 3 iterations for the sizing. Execution time was 11 min and 27 sec.

method	platform	topology selection	layout constraints	setup time	exec. time	technology migration
GTS	Cadence DFII	yes	are considered	2 min	11 min 27 sec	easy
designer (hands on work)	Cadence DFII	yes	should be considered	half day (creating test benches)	1 h	⊘
Cadence "global optimization"	Cadence DFII	no	can be included	half day (creating test benches, defining parameters, holding on constraints)	2 days	automatically
AIDA [7]	Cadence DFII Mentor Graphics Synopsys	no	option	1 h	several hours	⊘
WiCkeD [13]	Cadence DFII Mentor Graphics Synopsys	no	are considered	10 min (without time for creating test bench)	10 min – 30 min	3 days
MOJITO [7], [11]	⊘	yes	no	⊘	<7 days	⊘

Fig. 6. Comparison of our method with hands on work of a designer and other chosen tools with sizing option for the OTA example.

[7] N. Lourenco, R. Martins, and N. Horta. *Automatic Analog IC Sizing and Optimization Constrained with PVT Corners and Layout Effects*. Springer, 2017.

[8] T. Massier and H. Graeb. The sizing rules method for cmos and bipolar analog integrated circuit synthesis. *Computer-Aided Design of Integrated Circuits and Systems*, 27, 2008.

[9] T. McConaghy, P. Palmers, M. Steyaert, and G. Gielen. Trustworthy genetic programming-based synthesis of analog circuit topologies using hierarchical domain-specific building blocks. *Computer-Aided Design of Integrated Circuits and Systems*, 15(4), 2011.

[10] M. Meissner and L. Hedrich. Feats: Framework for explorative analog topology synthesis. *Computer-Aided Design of Integrated Circuits and Systems*, 34(2):213 – 226, February 2015.

[11] R. Riolo, T. Soule, and B. Worzel. *Genetic Programming Theory and Practice V*. Springer, 2008.

[12] R. A. Rutenbar, G. G. E. Gielen, and B. A. Antao. Idac: An interactive design tool for analog cmos circuits. *Wiley-IEEE Press*, pages 45 – 55, 2002.

[13] R. A. Rutenbar, G. G. E. Gielen, and B. A. Antao. Wicked: Analog circuit synthesis incorporating mismatch. *Wiley-IEEE Press*, pages 217 – 220, 2002.

[14] J. Scheible and J. Lienig. Automation of analog ic layout – challenges and solutions. *Proc. of Int. Symp. on Physical Design (ISPD), Monterey, CA, USA*, pages 33 – 40, March 2015.

[15] C. Toumazou, G. Moschytz, and B. Gilbert. *Trade-Offs in Analog Circuit Design: The Designer's Companion*. Kluwer Academic Publishers, 2002.