

Automated Constraint-driven Topology Synthesis for Analog Circuits

Oliver Mitea, Markus Meissner, Lars Hedrich
Electronic Design Methodology, Department of Computer Science,
University of Frankfurt/Main, Germany
Email: {mitea, meissner, hedrich}@em.cs.uni-frankfurt.de

Peter Jores
Robert Bosch GmbH
Reutlingen, Germany
Email: peter.jores@de.bosch.com

Abstract—This contribution will present a fully automated approach for explorative topology synthesis of small analog circuit blocks. Circuits are composed from a library of basic building blocks. Therefore, various algorithms are used to explore the entire design space, even allowing to generate unusual circuits. Correct combination of the basic blocks is accomplished through generic electrical rules, which ensure the fundamental electrical functionality of the generated circuit. Additionally, symmetry constraints are introduced to narrow the design space, which leads to more reasonable circuits. Further a replaceable bias-voltage generator is included into the circuit to replicate real world circumstances. For the first evaluation and selection of best candidate circuits, fast symbolic analysis techniques are used. The final sizing is done through a parallelized industrial based sizing method. Experimental results show the feasibility of this synthesis approach.

I. INTRODUCTION

Nowadays, especially in System-on-Chip designs, relatively small analog parts of the circuit require a high engineering effort. Due to system complexity and needed expert-knowledge a productivity gap arose between analog and digital designs. A higher automation in analog CAD tools is needed to overcome this gap.

During the last decade probably the most important step towards the automation of analog circuit design was the introduction of simulation based sizing tools. From first successful academic demonstrations evolved several commercial tools. Usually these tools rely on SPICE-level simulations, which leads to trustworthy results [1].

The next challenge towards an automated design flow is the generation of circuit topologies. Numerous approaches are known already, but none of them reached a successful commercialization [2]–[4]. All methods have to cope with the huge design space. For the final benchmarking of a single circuit topology a proper sizing is needed, which is still consuming a lot of computation time. So, a full coverage of the theoretical design space is impossible for circuits consisting of more than 5 transistors [5]. For dealing with bigger circuits an amount of designer knowledge has to be introduced into the design flow. There are different possibilities in doing this. The highest amount of designer knowledge is used by

This work was partly developed within the project SyEnA (project label 01 M 30 86) which is funded within the Research Program ICT 2020 by the German Federal Ministry of Education and Research (BMBF).

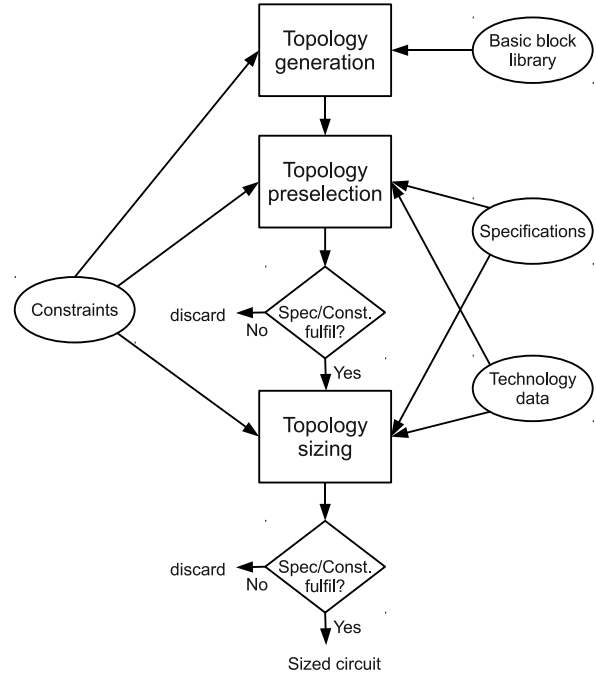


Fig. 1. Synthesis methodology.

topology selection instead of generation. This methods have the drawback in high setup effort, low flexibility and no possibility to generate unknown topologies [6], [7]. Every generation methodology has to do a trade-off between allowed creativity with large design space on one hand, and application of designer knowledge with small design space on the other. Even the different methods using multiobjective evolutionary algorithms [8], which gained popularity during the last years, are using a lot of designer knowledge. Either through using building blocks on a grid structure [9], [10], or even using a full circuit library, which is diversified [11].

In [12] a method for circuit generation using a basic building block library is used. The blocks are interconnected according to a set of rules, which ensure fundamental electrical functionality of the resulting circuit topology. For the detailed evaluation of the circuits symbolic analysis is used. For more trustworthy results in [13] the approach was extended by a

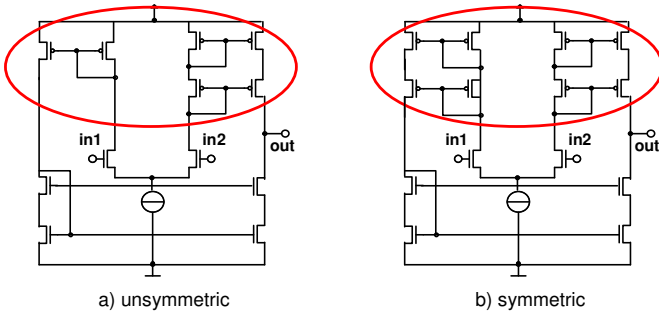


Fig. 2. Generated symmetric OTA.

final sizing with SPICE-accuracy by a commercial tool [14].

For faster synthesis of circuits with higher quality, the previous work is extended by several techniques:

- A symmetry constraint is introduced during topology generation, which leads to more reasonable circuits and narrows the design space.
- The symbolic analysis technique is extended by a constraint checker. By the use of some simple rules, circuits which are impossible to be biased properly are discarded early in the synthesis process.
- A bias voltage generator is included to the generated circuit ahead of starting the sizing.

The rest of this paper is organized as follows. Section II describes the different design steps of the proposed flow and also explains the novel extensions. Section III shows the impact of the extensions on the synthesis method. A full design example allows statistical evaluations and also detailed evaluation of example circuits. Section IV concludes the work.

II. SYNTHESIS METHODOLOGY

Figure 1 shows the overall design flow. A step by step approach is proposed which ensures that the faster algorithms are executed at first, and the most computation time consuming ones at last. In every design step designer knowledge is integrated by using different types of constraints, which leads to a reasonably reduced design space without affecting the creativity of the methodology too much. New topologies are generated during some milliseconds, and constraints ensure the electrically correct connection as well as symmetric circuits. During topology preselection, which takes seconds to one minute per circuit, saturation constraints are checked, and symmetry constraints reduce the number of design parameters. In the final sizing step a set of constraints ensure the correct biasing of the circuit and also the number of design parameters is reduced.

A. Generation of Circuits with Symmetry Constraints

New circuit topologies are composed from a hierarchical library of basic building blocks, which can be found in standard analog literature [15]. Every block has one or more terminals with some specific information, which is necessary for the electrically correct connection of different blocks (input/output terminal, high/low impedance and voltage/current signal). One

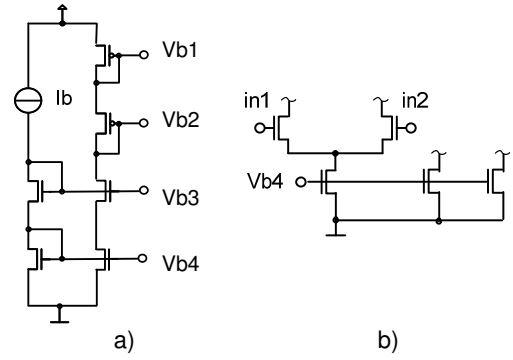


Fig. 3. Bias voltage generation.

block can contain different netlists, for example the current mirror block has three different netlists (simple CM, cascode CM, low-voltage cascode CM). So, different circuit variants can be generated from one block topology. The method is described in more detail in [12].

The usage of hierarchical blocks often results in very unsymmetric circuits which are undesirable. This can lead to circuits with high offset voltages or to low yield after fabrication. Because these circuits work in principle, the problems usually arise in the computation time consuming sizing step of the actual design flow. Therefore, a symmetry constraint is introduced into topology generation step. During circuit variant generation, the algorithm memorizes the used basic block instances and inhibits the usage of other instances. Circuits like figure 2 a) are not generated any more.

B. Circuit Preselection

The simulation based automated circuit sizing is the most computation time consuming step, usually taking 5-30 minutes per circuit instance. Hence faster methods are needed for the evaluation of thousands of generated circuit topologies. In principle the symbolic analysis based methods described in [12] are used as a preselection step. All transistors are assumed to be in saturation and are replaced by a small signal model. Important technology data is automatically extracted from the used CMOS process and includes the transconductance factor $K_{p/n} = \mu_{p/n} C_{ox}$, the parasitic capacitances C_{gs} and C_{gd} , the threshold voltage V_{th} and the Early voltage V_E . Herewith the small signal parameters g_m and r_{ds} can be expressed symbolically in terms of W , L and I_{bias} .

Using a predefined range of device and biasing parameters (W_i , L_i and I_{bias_i}) a differential algebraic equation (DAE) system can be setup, simplified and solved for the output variables resulting in a transfer function

$$H(s) = \frac{Y(s)}{X(s)} = f(W_1, L_1, I_{bias_1}, \dots, W_n, L_n, I_{bias_n}). \quad (1)$$

This transfer function enables a fast estimation of circuit performances using a greedy sizing algorithm for the design parameters W_i , L_i and I_{bias_i} .

	unsymmetric	symmetric	reduction
3 blocks	52 blockchains		
circuit variants	264	264	- 0 %
4 blocks	448 blockchains		
circuit variants	4 424	3 455	- 21 %
5 blocks	3 448 blockchains		
circuit variants	57 328	39 928	- 30 %
6 blocks	27 207 blockchains		
circuit variants	819 800	475 400	- 42 %

TABLE I
SYMMETRIC CIRCUIT VARIANT GENERATION.

In order to improve the reliability of this performance estimation two additional constraint sets are introduced.

1) *Sizing Constraints*: Also the W_i and L_i parameters comply to a set of structural symmetry constraints, which are included into the basic block library, including identical L in current mirrors and identical W and L in differential pairs.

2) *Biasing Constraints*: For ensuring the possibility of biasing the circuit an equation system is derived from the basic saturation constraints $V_{GS} > V_{th}$ and $V_{DS} > V_{GS} - V_{th}$ of each transistor. If no feasible operation point can be derived the circuit is discarded.

C. Sizing and Biasing

All circuits passing the preselection are automatically sized with a commercial tool [14]. In the former approach [13] all biasing voltages, which are needed for cascode structures or current sources, were treated autonomously. For every voltage a voltage source was inserted to the circuit netlist, with the output voltage being a design parameter for the sizing tool, which led to unrealistic behavior for some circuits. Also the optimizer could not find the right voltage level and possibly good circuits were discarded.

To overcome this drawbacks a bias voltage generator shown in figure 3 a) is included to each netlist. The sizing tool can use three design parameters to optimally adjust the generator for the specific circuit. These are the bias current I_b , transistor length L_{bias} and width W_{bias} with pMOS width being $pMOSfactor \cdot W_{bias}$. The right bias voltages are distributed through the basic block library and allow an automated con-

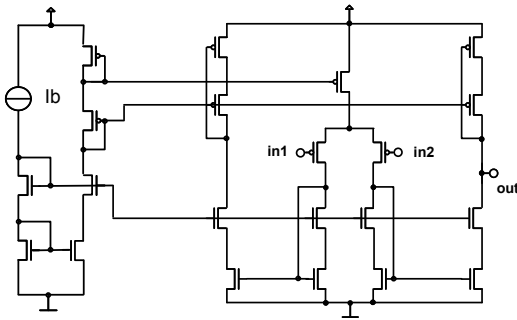


Fig. 4. Generated example circuit OTA 1.

nection between bias voltage generator and the rest of the circuit.

Furthermore single current sources are connected to current mirror banks (figure 3 b)) with identical transistor length L_{bias} and variable width $W_{1...n}$ allowing individual bias current sizing.

Ahead of running the sizing program a runtime environment has to be set up, which includes the circuit netlist with the bias generator, the testbench and the instructions for the characterization of the circuit. For a fast and proper sizing a set of dc constraints is needed which is described in [16]. Some constraints are approximately resolved in advance by the biasing constraints method in section II-B2. These circuit specific constraints are automatically extracted by an structure recognition tool and inserted into the runtime environment.

The first sizing step is a built in feasibility optimization, checking if the constraints are satisfiable and setting up a correct operating point. If the constraints are not fulfilled, the circuit topology is discarded. Afterwards successful circuits are nominal optimized to meet the given specifications using the WiCkeD tool [14].

III. EXPERIMENTAL RESULTS

The synthesis tool has been written in C++. The topology preselection is done in Maple. The sizing tool uses for simulations SPECTRE [17]. The technology was a 0.18μ CMOS process with 3 V supply voltage and a 5pF load capacitance. The technology is also easily replaceable. All calculations were done on a dual-core Linux machine with 2.4 GHz. For the full synthesis run the maximum number of basic blocks per circuit is limited to 4. The specification from table II were used for sizing. During topology preselection a subset was used, containing gain, PSRR and output voltage range. The runtime was about 50 hours. More blocks are not applicable at the moment due to the high number of circuits, as can be seen in the next section.

A. Design Space Reduction through Symmetry Constraints

Through the usage of symmetry constraints during topology generation the design space is dramatically reduced. Table I compares a full generation run with different maximum allowed block chain length and the corresponding number of circuit variants which can be created from the hierarchical

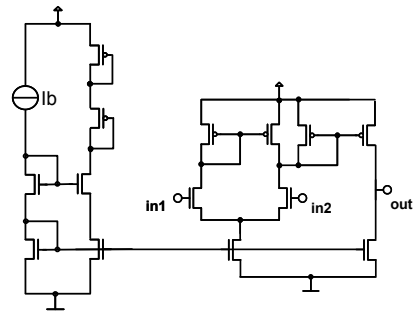


Fig. 5. Generated example circuit OTA 2.

Active area (μm^2) <	5000
DC voltage gain (dB) >	50
Output voltage range (V) >	2
Offset (mV) <	10
Transit frequency (MHz) >	4
Power dissipation (μW) <	200
Phase margin (deg) >	45
Slew rate (V/ μs) >	1
PSRR (dB) >	50
CMRR (dB) >	50
Overshoot (%) <	30
Settling time (μs) <	1

TABLE II
USED SPECIFICATIONS.

block chain library. With higher number of blocks the percentaged reduction increases. When only 3 blocks are allowed no circuit is affected by symmetry, but with 6 blocks the reduction rate is over 40 percent. For bigger circuits an even higher reduction can be expected.

B. Improved Circuits Preselection

One major drawback of the previous work was a relatively large error during the topology preselection step. That means that too many circuits had to be sized with the comparatively slow circuit sizer, and there were still a lot of false negatives, which are lost for the evaluation. Due to the numerous modifications on the design flow the results are not actually comparable to the old ones. Remarkable is the high rate of successfully sizable circuits which is over one third. Also only few circuits are discarded the after the feasibility optimization of the subsequent sizing step.

C. Synthesis Example: Operational Amplifier

From the 3455 generated topologies of the synthesis run with 4 blocks 2723 are discarded through the preselection step. The calculations took only about 4 hours, which is only 8 percent of the total computing time. 97 of the remaining 723 circuits are discarded after the feasibility optimization. That means over 86 percent of the preselected circuits have a feasible operating point. Finally, 239 of the nominal optimized circuits meet all the given specifications.

In the successful set known standard circuits like the operational transconductance amplifier (figure 4) with low voltage cascode current mirrors can be found as well as more unusual circuits like the one in figure 5. Note that the bias circuit from figure 3 a) is also included in the circuit.

IV. CONCLUSION

A synthesis methodology has been introduced that is able to compose analog circuits using a library of well-known basic building blocks. Three different improvements have been integrated into the proposed design flow. In every design step different type of constraints ensure a fast and reasonable reduction of the design space. That leads to acceptable overall runtime. The use of a commercial sizing tool, based on SPICE

Performances	OTA 1	OTA 2
Active area (μm^2)	4 912	1 697
Voltage gain (dB)	79.4	55.7
Output voltage range (V)	2.9	2.7
Offset (mV)	5	-7
Transit frequency (MHz)	4.6	4.2
Power dissipation (μW)	625	589
Phase margin (deg)	45	46
Slew rate (V/ μs)	3.2	1.7
PSRR (dB)	69.5	63.9
CMRR (dB)	73.9	56.3
Overshoot (%)	26	17
Settling time (μs)	0.37	0.32

TABLE III
REACHED PERFORMANCES OF THE EXAMPLE CIRCUITS.

accurate simulations, leads to trustworthy results, which show the feasibility of the synthesis approach.

REFERENCES

- [1] R.A. Rutenbar. Design automation for analog: the next generation of tool challenges. In *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, page 460. ACM, 2006.
- [2] C. Toumazou, CA Makris, and CM Berrah. ISAID-a methodology for automated analog IC design. In *Circuits and Systems, 1990., IEEE International Symposium on*, pages 531–535. IEEE.
- [3] HY Koh, C.H. Sequin, and P.R. Gray. OPASYN: A compiler for CMOS operational amplifiers. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 9(2):113–125, 1990.
- [4] BAA Antao and AJ Brodersen. ARCHGEN: automated synthesis of analog systems. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 3(2):231–244, 1995.
- [5] R. Harjani, R.A. Rutenbar, and L.R. Carley. OASYS: A framework for analog circuit synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 8(12):1247–1266, 1989.
- [6] P.C. Maulik, L.R. Carley, and R.A. Rutenbar. Integer programming based topology selection of cell-level analog circuits. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 14(4):401–412, 2002.
- [7] W. Kruiskamp and D. Leenaerts. DARWIN: CMOS opamp synthesis by means of a genetic algorithm. In *Design Automation, 1995. DAC'95. 32nd Conference on*, pages 433–438. IEEE, 2006.
- [8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [9] A. Das and R. Vemuri. Topology synthesis of analog circuits based on adaptively generated building blocks. In *Proceedings of the 45th annual Design Automation Conference*, pages 44–49. ACM, 2008.
- [10] TR Dastidar, PP Chakrabarti, and P. Ray. A synthesis system for analog circuits based on evolutionary search and topological reuse. *IEEE Transactions on evolutionary computation*, 9(2):211–224, 2005.
- [11] P. Palmers, T. McConnaghy, M. Steyaert, and G. Gielen. Massively multi-topology sizing of analog integrated circuits. In *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE'09.*, pages 706–711. IEEE, 2009.
- [12] X. Wang and L. Hedrich. An approach to topology synthesis of analog circuits using hierarchical blocks and symbolic analysis. In *Proceedings of the 2006 Asia and South Pacific Design Automation Conference*, page 705. IEEE Press, 2006.
- [13] X. Wang and L. Hedrich. Hierarchical exploration and selection of transistor-topologies for analog circuit design. In *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pages 4–1470. IEEE, 2006.
- [14] MunEDA GmbH. www.muneda.com.
- [15] B. Razavi. Design of analog CMOS integrated circuits. 2000.
- [16] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich. The sizing rules method for analog integrated circuit design. 2001.
- [17] Cadence Design Framework. www.cadence.com.