

# AUTOMATIC SYNTHESIS OF OPERATIONAL AMPLIFIERS BASED ON ANALYTIC CIRCUIT MODELS

Han Young Koh, Carlo H. Séquin and Paul R. Gray

*Department of Electrical and Computer Engineering and Computer Sciences  
University of California, Berkeley, CA*

## **Abstract**

An automatic synthesis tool for CMOS op amps (OPASYN) has been developed. The program starts from one of a number of op amp circuits and proceeds to optimize various device sizes and bias currents to meet a given set of design specifications. Because it uses analytic circuit models in its inner optimization loop, it can search efficiently through a large part of the possible solution space. The program has a SPICE interface that automatically performs circuit simulations for the candidate solutions to verify the results of the synthesis and optimization procedure. The simulation results are also used to fine-tune the analytic circuit descriptions in the database. OPASYN has been implemented in Franz Lisp and demonstrated for three different basic circuits with a conventional 3  $\mu\text{m}$  process and a more advanced 1.5  $\mu\text{m}$  process. Experiments have shown that OPASYN quickly produces practical designs which will meet reasonable design objectives.

## **1. Introduction**

In recent years, rapid advances have been made in automating the design of analog and mixed analog/digital circuits [1, 2, 3, 4, 5, 6]. The general approach is to use building blocks which may be stored in the form of parameterized generators or as entries in macrocell or standard cell libraries. While the usage of libraries of predefined building blocks can shorten the design period, it cannot give an optimal design for every application. Furthermore, the library entries become obsolete each time the technology or the design rules are modified. Generators that are operating at the circuit or symbolic level are more flexible and can be useful over a much larger domain.

Among various building blocks used in analog systems, an operational amplifier (op amp) is one of the most widely used circuit components. An efficient design of optimal op amps is thus a corner-stone of a design environment for many applications. Several methods concerning the automated design of op amps have been published [1, 6, 7, 8, 9]. An optimization-based approach is based on algorithmic optimization and circuit analysis techniques. It is applicable to a broad range of analog circuits and produces near optimal solutions. However, this approach is costly in CPU time and has difficulty in properly tuning the system according to the designers' needs. Alternatively, an expert system can be used to store human designer's knowledge. But the large search space resulting from the many degrees of freedom in the design of op amp circuits makes this approach difficult and inefficient.

This paper introduces a practical intermediate approach to automating op amp synthesis. It is based on analytic circuit models of the op amp circuit topologies covered by the synthesis system. It follows the general approach often taken by human designers, but automates the tedious and computationally extensive aspects of the design process. Based on the applications, a suitable circuit topology is

selected, and its device sizes and bias currents are then adjusted in an iterative manner until the circuit optimizes some selected targets. When a reasonable design configuration has been found, it is subject to extensive circuit simulation to verify that indeed all the design specifications are met. This whole process has been automated.

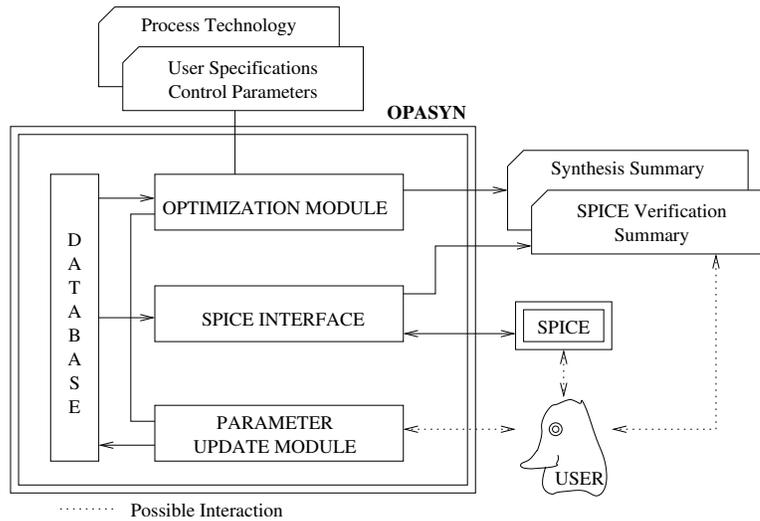


Figure 1. Organization of the OPASYN system.

As shown in Fig. 1 our op amp synthesis system (OPASYN) consists of a topology database, an optimization module, an interface to the circuit simulator, SPICE, and a parameter update module; the latter three are circuit and technology independent. The topology database stores analytic circuit performance models for each of the adopted circuit topologies so that the design parameters can be calculated without the aid of a circuit simulator. The optimization module improves the optimality of the design with an algorithmic search procedure. The SPICE interface automatically simulates the chosen circuit and determines its exact performance parameters, thus verifying the synthesis results. The SPICE simulation summary is also used by the parameter update module to improve the analytic circuit models in the database. A modular system configuration makes it easy to update OPASYN's database as device technology changes or better circuit models become available.

## 2. Circuit Synthesis Based on Analytic Circuit Models

At the heart of OPASYN is an efficient circuit optimization module. The optimization process relies on analytic circuit models which contain analog circuit designers' expert knowledge about each op amp circuit topology and the dependencies between device sizes, bias currents, and performance characteristics of the overall circuit. Thus the inner loop of the optimization process can run much more efficiently using the explicit dependencies in these models rather than performing a circuit simulation after each optimization step. In addition, this analytic modeling of a given op amp circuit topology produces a simple and smooth solution space so that the optimization procedure can use faster - but less robust - search procedures. In OPASYN, a simple steepest-gradient descent method has been used to explore large portions of the solution space within a short period of time.

Each of the analytic circuit models contains a netlist for the corresponding circuit topology, a declaration of independent design parameters for the circuit, and a reasonable range of values for these parameters. On top of these basic properties, the model stores analytic expressions to compute circuit performances. These expressions were derived by using first-order circuit analysis techniques and topology-specific approximations [13]–[17]. For most dc characteristics these computed approximations are excellent. For highly non-linear specifications such as gain, phase margin, and settling time of the circuit, fitting parameters have been introduced to obtain more accurate predictions of specific performance characteristics. One example of such analytic expressions is

$$a_v = cf_{\text{gain}} \times \frac{g_{m2}g_{m6}}{(g_{o2} + g_{o4})(g_{o6} + g_{o7})}$$

where  $a_v$ , is the small signal dc gain of the circuit topology in Fig. 2,  $g_m$  is a transconductance,  $g_o$  is an output conductance of a transistor, and  $cf_{\text{gain}}$  is a fitting parameter. All the conductances are dependent on transistor sizes and bias currents. The fitting parameters are being updated as the system acquires more information from repeated synthesis and verification steps.

Based on these equations and the user-defined design targets, a cost function is computed which represents a relative figure-of-merit for any particular combination of design parameter values. In the present version, the cost function ( $C$ ) is formed as follows:

$$C = \sum_1^n \left( \exp \left( \pm \frac{w_i (\text{spec}_i - p_i)}{\text{spec}_i} \right) - 1 \right)$$

where  $n$  is the number of circuit performance parameters considered in the program,  $p_i$  is the  $i$ -th performance parameter,  $w_i$  is a relative design priority of  $p_i$  and  $\text{spec}_i$  is the corresponding design specification. The plus sign is chosen when meeting the specification requires the value of  $p_i$  to be greater than that of  $\text{spec}_i$ . The cost function produces a smooth search space where the gradient descent method works effectively. The exponential nature of the cost function prevents the penalty for violating any specification from being compensated by overly satisfying other specifications.

The search for an optimal solution starts with a coarse-grid sampling through the entire parameter space fine enough to yield a starting point in each cost function well. From the more promising sampling points found in this survey, a steepest-gradient descent algorithm searches for the optimal solution in this neighborhood. OPASYN will often return several solutions if many different locally optimal solutions are found that come close enough to the stated design targets.

The described optimization algorithm is independent of the specific device technology or circuit topology used. However, when a new circuit topology needs to be introduced into OPASYN, the corresponding analytic circuit models must first be created. This is a non-trivial task that demands the attention of a good analog circuit designer.

In addition to creating the proper expressions that describe the functional dependencies in a new circuit model, reasonable values for the various fitting parameters must be determined; this requires a substantial number of simulation runs. Any discrepancy between the predictions and the results of the circuit simulations can be used to fine-tune these fitting parameters and to improve the accuracy of the predictions. Thus the more the system is used, the more accurate it gets because of the ‘experience’ gained from previous design tasks.

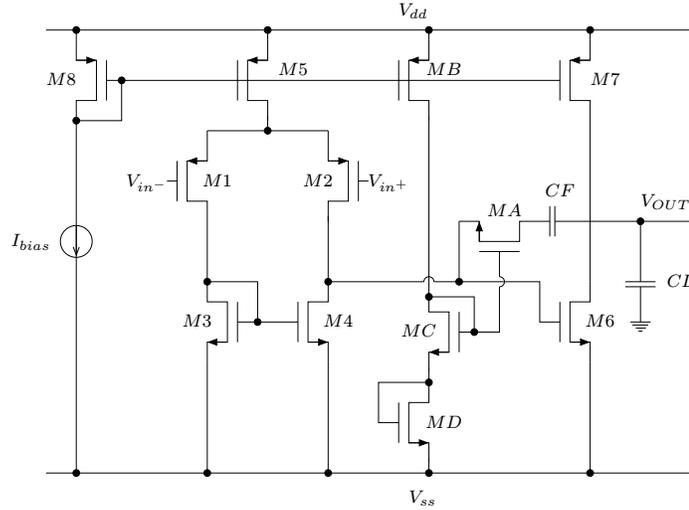


Figure 2. Basic Two Stage OP Amp.

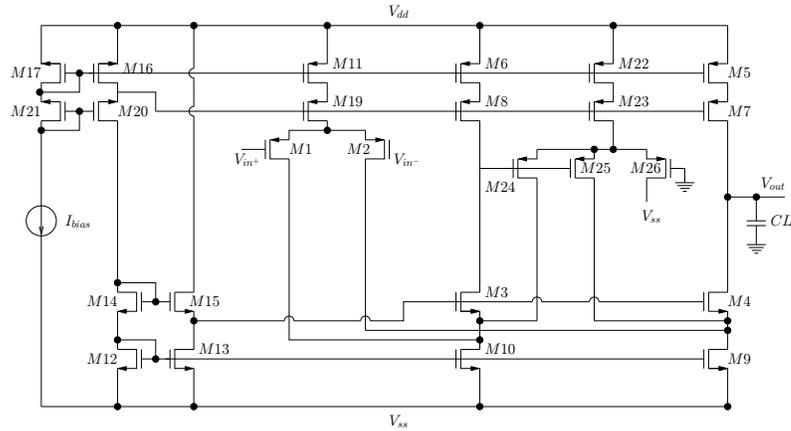


Figure 3. Single Stage Folded Cascode OP Amp.

### 3. Automatic Verification of Synthesis Results

After the device parameters have been determined in the optimization phase, the resulting circuit(s) need to be carefully checked against the given design specifications. This is done with a series of SPICE simulation runs to accurately determine the various performance characteristics. At this stage it is also possible to study the influence of processing variations and device tolerances.

OPASYN's SPICE interface module generates the needed SPICE input files, makes the necessary system calls to execute the various simulations, interprets the SPICE output files to determine the various performance characteristics, and compares the latter against the given design targets. OPASYN then supplies a summary of these simulation results to the user. It also utilizes these results to improve the fitting parameters in the analytic models in the database.

One of the difficulties with SPICE simulation of analog MOS circuits is to achieve dc convergence. Designers often spend a considerable amount of time try-



CPU time for the synthesis phase varies with the difficulty of the user specifications and the degree of optimality to be achieved. The range of CPU times observed for the synthesis phase of the interpreted version of OPASYN is from 70 to 280 seconds on a VAX 8800. The SPICE verification phase typically requires about 200 seconds.

## 5. Conclusion

An efficient CMOS op amp synthesis tool (OPASYN) has been developed. It uses analytic circuit models to estimate circuit performance during the search for the optimal solution. Our experiments have shown that this approach greatly reduces the required CPU time while still producing near-optimal results.

OPASYN has been applied successfully to three of the most commonly used op amp circuit topologies using two different process technologies. The synthesis process has been reliable and produced good results.

OPASYN can be easily used by engineers inexperienced in op amp design. It does not normally require any user intervention in the design phase. Users simply specify their design requirements and optimization priorities and select the most desirable result out of several options produced by OPASYN.

## Acknowledgments

The authors would like to thank Dr. Richard D. Baertsch in General Electric Company for evaluating the program and providing valuable feedback on many practical aspects. We also gratefully acknowledge the contributions of many graduate students at U.C. Berkeley.

This work has been supported by DARPA VLSI-N00039-87-C-0182 and SRC-82-11-008.

## References

- [1] Harjani, R., A. Rutenbar, and L. R. Carley (1987). "A Prototype Framework for Knowledge-Based Analog Circuit Synthesis," *24th Design Automation Conf.*, pp. 42–49.
- [2] Allen, P. E. (1986). "A Tutorial - Computer Aided Design of Analog Integrated Circuits," *Proc. of IEEE CICC*, pp. 608–616.
- [3] Pletersek, T., J. Trontelj, and L. Trontelj (1985). "Analog LSI Design with CMOS Standard Cells," *Proc. of IEEE CICC*, pp. 479–483.
- [4] DeGrauwe, M. G., et al. (1987). "An Analog Expert Design System," *IEEE ISSCC Digest of Technical Papers*, pp. 212–214.
- [5] Kelson, G. (1985). "Design Automation Techniques for Analog VLSI," *VLSI Design*, pp. 78–82.
- [6] El-Turky, F. M., and R. A. Nordin (1986). "BLADES: An Expert System for Analog Circuit Design," *Proc. of IEEE ISCAS*, pp. 552–555.
- [7] DeGrauwe, M. G. (1984). "A Synthesis Program for Operational Amplifiers," *IEEE ISSCC Digest of Technical Papers*, pp. 18–19.
- [8] Allen, P. E., and H. Nevarez-Lozano (1983). "Automated Design of MOS OP Amps," *Proc. of IEEE ISCAS*, pp. 1286–1289.
- [9] Bowman, R. J., and D. J. Lane (1986). "A Knowledge-Based System for Analog Integrated Circuit Design," *Proc. of IEEE ICCAD*, pp. 210–212.
- [10] Nye, B., D. Riley, and A. Sangiovanni-Vincentelli (1984). "DELIGHT.SPICE User's Guide," Dept. EECS, University of California, Berkeley.
- [11] Vladimirescu, A., K. Zhang, A. R. Newton, D. O. Pederson, and A. Sangiovanni-Vincentelli (1981). "SPICE Version 2G - User's Guide," Dept. EECS, University of California, Berkeley.

- [12] Horowitz, E., and Sahni (1978). *Fundamentals of Computer Algorithms*, Computer Sciences Press.
- [13] Gray, P. R., and Meyer, R. G. (1984). *Analysis and Design of Analog Integrated Circuits*, 2nd Edition. New York: Wiley.
- [14] Gray, P. R. (1982). "MOS Operational Amplifier Design- A Tutorial Overview," *IEEE J. Solid-State Circuits*, vol. SC-17, No. 6.
- [15] Ahuja, B. K. (1983). "An Improved Frequency Compensation Technique for CMOS Operational Amplifiers," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 629-633.
- [16] Chuang, C. T. (1982). "Analysis of the Settling Behavior of an Operational Amplifier," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 74-80.
- [17] Kamath, B. Y., R. G. Meyer, and P. R. Gray (1974). "Relationship between Frequency Response and Settling Time of Operational Amplifiers," *IEEE J. Solid-State Circuits*, vol. SC-9, pp. 347-352.

abbreviations used:

Vdd	=	positive power supply voltage
Vss	=	negative power supply voltage
CL	=	load capacitor
C <sub>c</sub>	=	compensation capacitor
wu	=	unity gain bandwidth
PSRR <sup>-</sup> @dc	=	V <sub>ss</sub> rejection ratio at dc
PSRR <sup>-</sup> @kilo	=	V <sub>ss</sub> rejection ratio at 1 kHz
TS	=	settling time (1 V step, 0.1 % interval)
V <sub>o,max</sub>	=	maximum output voltage
V <sub>o,min</sub>	=	minimum output voltage
V <sub>ic,max</sub>	=	maximum common mode input voltage
V <sub>ic,min</sub>	=	minimum common mode input voltage
1/f noise	=	input equivalent 1/f noise at 1 kHz

parameter	specification	synthesis	SPICE
Vdd	2.5 V	2.5 V	2.5 V
Vss	-2.5 V	-2.5 V	-2.5 V
CL	10 pF	10 pF	10 pF
gain	10,000	23,610	24,250
power dissipation	1 mW	0.67 mW	0.66 mW
phase margin	60 deg	57.1 deg	56.1 deg
wu	4 MHz	4.7 MHz	4.7 MHz
gain margin	none	none	15.1 dB
CMRR	80 dB	none	92 dB
slew rate	2.5 V /μsec	3.9 V/μsec	3.4 V/μsec
PSRR <sup>-</sup> @dc	70 dB	94.8 dB	94.8 dB
PSRR <sup>-</sup> @1kilo	40 dB	58.0 dB	74.3 dB
TS	500 nsec	453 nsec	550 nsec
systematic offset	none	none	0.008 mV
V <sub>o,max</sub>	1.5 V	2.38 V	2.40 V
V <sub>o,min</sub>	-1.5 V	-2.33 V	-2.39 V
V <sub>ic,max</sub>	1 V	1.45 V	1.40 V
V <sub>ic,min</sub>	1 V	-2.33 V	-2.50 V
1/f noise	1E-6 V/√Hz	2.2E-7	none
total gate area**	40 mil <sup>2</sup>	35.3 mil <sup>2</sup>	none
C <sub>c</sub>	none	4.8 pF	4.8pF

Table 1. Synthesis and verification results for the basic two stage op amp (shown in Fig. 2) with optimization priority given to total gate area. MOSIS 3 μm process was used. CPU time for the synthesis was 175 seconds on a VAX 8800.

parameter	specification	synthesis	SPICE
Vdd	2.5 V	2.5 V	2.5 V
Vss	-2.5 V	-2.5 V	-2.5 V
CL	2 pF	2 pF	2 pF
gain	1,500	1,421	1,496
power dissipation	30 mW	2.85 mW	2.72 mW
phase margin	60 deg	38.8 deg	33.0 deg
wu	4 MHz	42.3 MHz	30.0 MHz
gain margin	none	none	15 dB
CMRR	none dB	none	132 dB
slew rate	8 V / $\mu$ sec	22 V/ $\mu$ sec	19 V/ $\mu$ sec
PSRR <sup>-</sup> @dc	40 dB	122 dB	112 dB
PSRR <sup>-</sup> @1kilo	10 dB	122 dB	112 dB
TS**	100 nsec	164 nsec	160 nsec
systematic offset	0.1 mV	none	0.01 mV
V <sub>o,max</sub>	1.5 V	1.86 V	2.10 V
V <sub>o,min</sub>	-1.5 V	-1.88 V	-2.10 V
V <sub>ic,max</sub>	1 V	0.78 V	1.50 V
V <sub>ic,min</sub>	1 V	-2.50 V	-2.50 V
1/f noise	1E-6 V/ $\sqrt$ Hz	1.5E-7	none
total gate area	70 mil <sup>2</sup>	75 mil <sup>2</sup>	none
C <sub>c</sub>	none	none	none

Table 2. Synthesis and verification results for the single stage folded cascode op amp (shown in Fig. 3) with optimization priority given to settling time. MOSIS 3  $\mu$ m process was used. CPU time for the synthesis was 278 seconds on a VAX 8800.

parameter	specification	synthesis	SPICE
Vdd	2.5 V	2.5 V	2.5 V
Vss	-2.5 V	-2.5 V	-2.5 V
CL	5 pF	5 pF	5 pF
gain	10,000	15,140	13,890
power dissipation	1 mW	0.85 mW	0.86 mW
phase margin	60 deg	65.3 deg	62.1 deg
wu	4 MHz	7.2 MHz	7.2 MHz
gain margin	none	none	9 dB
CMRR	none dB	none	95 dB
slew rate	2.5 V/ $\mu$ sec	4.7 V/ $\mu$ sec	4.6 V/ $\mu$ sec
PSRR <sup>-</sup> @dc	70 dB	90 dB	90 dB
PSRR <sup>-</sup> @1kilo	40 dB	90 dB	90 dB
TS**	500 nsec	387 nsec	420 nsec
systematic offset	none	none	0.46 mV
V <sub>o,max</sub>	1.5 V	2.37 V	2.39 V
V <sub>o,min</sub>	-1.5 V	-2.32 V	-2.39 V
V <sub>ic,max</sub>	1 V	1.45 V	1.50 V
V <sub>ic,min</sub>	1 V	-0.88 V	-1.00 V
1/f noise	1E-6 V/ $\sqrt$ Hz	1.7E-7	none
total gate area**	40 mil <sup>2</sup>	38.7 mil <sup>2</sup>	none
C <sub>c</sub>	none	3.4 pF	3.4 pF

Table 3. Synthesis and verification results for the two stage op amp with cascoded first stage (shown in Fig. 4) with optimization priority given to settling time and total gate area. GE 1.5  $\mu$ m process was used. CPU time for the synthesis was 82 seconds on a VAX 8800.