Operational Amplifiers: An Example for Multi-Objective Optimization on an Analog Evolvable Hardware Platform

Martin Trefzer, Jörg Langeheine, Karlheinz Meier, Johannes Schemmel

Ruprecht-Karls-University of Heidelberg
Kirchhoff Institute for Physics
Im Neuenheimer Feld 227, 69120 Heidelberg, Germany
martin.trefzer@kip.uni-heidelberg.de, +49 (0)6221 54-9838
http://www.kip.uni-heidelberg.de/vision/projects/eh/

Abstract. This work tackles the problem of synthesizing transferable and reusable operational amplifiers on a field programmable transistor array: the Heidelberg FPTA. A multi-objective evolutionary algorithm is developed, in order to be able to include various specifications of an operational amplifier into the process of circuit synthesis. Additionally, the presented algorithm is designed to preserve the diversity within the population troughout evolution and is therefore able to efficiently explore the design space. Furthermore, the evolved circuits are proven to work on the chip as well as in simulation outside the FPTA. Schematics of good solutions are presented and their characteristics are compared to those of basic manually created reference designs.

1 Introduction

Analog circuit development is a discipline of electronic design that demands a lot of knowledge and experience as well as a considerable amount of creativity in solving diverse problems from the designer. The design of task specific operational amplifiers (OP) is an example for an exercise that has to be done by experienced designers and exactly such OPs are essential building blocks of many electronic circuits. Contrary to digital circuit design there is still a lack of supporting tools for automatic synthesis and sizing of transistor circuits.

To date, to the authors knowledge, only a few analytic solutions for analog design automation are available. Examples in which previously known topologies are tested while the sizing of the components is done by an optimization algorithm are given in [1, 2]. In a great number of approaches, the topology is also to be found automatically, therefore, developmental strategies are applied in order to deal with the high complexity of amplifiers [3–6]. An alternative possibility is to choose a multi-objective evolutionary algorithm [7, 8], in order to face the fact that, for the solution of almost every complex problem, numerous variables have to be taken into account for optimization. Operational amplifiers as well as other transistor circuits found to this point by means of hardware evolution in conjunction with multi-objective optimization (MO) are reported in [9–11]. Furthermore, a multi-objective approach provides the designer

with a variety of choices instead of only one more or less good solution. This is a great advantage, especially in cases in which trade-offs have to be made, e.g. between gain and speed of an amplifier.

In this paper a multi-objective evolutionary algorithm, based on previous work [12] and referred to as the *MO-Turtle GA*, is presented and successfully used for the synthesis of differential amplifiers on the Heidelberg FPTA [13]. Other current results, obtained by using this FPTA, can be found in [14]. As proposed in an earlier publication [12], one of the aims is to synthesize circuits that contain only relevant components, thus, are easier to understand according to engineering criteria. The evolved circuits are proven to work on the chip as well as in simulation outside the FPTA. Schematics of good solutions are presented in this work and their characteristics are compared to those of manually created OPs. Two series of experiments are carried out using in one case a pair of PMOS transistors and in the other case a pair of NMOS transistors as input.

2 Evolvable Hardware System

The evolution system consists of three main parts: The FPTA that hosts the configurable CMOS transistor array, a controller with a PCI interface that connects the FPTA to a standard PC and the software that runs the multi-objective evolutionary algorithm and communicates with the FPTA via the controller. Thus, the experimental setup and the candidate configurations for the transistor array are generated on the PC and then transferred to the controller. Subsequently, the controller configures the FPTA and measures the output of the circuits under test. The software on the PC reads back the results and carries out the evolutionary steps. These components provide a real time test environment for the evolved circuits.

The transistor array consists of 16x16 configurable CMOS transistor cells (Fig. 1). Each cell contains a transistor that can be configured by selecting values for its width *W* and length *L* within $W = 1, 2, ..., 15\ \mu\mathrm{m}$ and $L = 0.6, 1, 2, 4, 8\ \mu\mathrm{m}$. The terminals (source, drain and gate) can be connected to one of the cells outside connections (N,S,W,E), vdd or gnd. Additionally, it is possible to directly connect the nodes (N,S,W,E) to each other, which provides routing capabilities. Half of the cells are designed as programmable PMOS and NMOS transistors respectively and are arranged in a checkerboard pattern. Owing to the four nodes available for routing and terminal connections, one cell mostly serves either as transistor cell or routing cell. However, both capabilities are not separated. The array is enclosed by IO cells that can apply voltages to the border cells or measure the output voltages of the evolved circuit. A detailed description of the FPTA is given in [13].

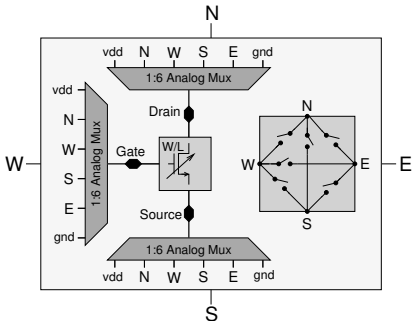

Fig. 1. The block diagram of an FPTA MOS transistor cell.

3 The Multi-Objective Evolutionary Algorithm

Since the evolution of operational amplifiers is a challenging task, a multi-objective strategy, first proposed in [15], is used for the experiments. This allows for a separate evaluation and optimization of different properties of the circuits, which would not be possible with a single objective algorithm. The *MO-Turtle GA* consists of a non-dominated sorting algorithm and a crowding distance measure, which are described in Sec. 3.2 and are based on those from the non-dominated sorting genetic algorithm, presented in [7, 16]. Using an MO approach offers two important advantages: First, numerous results can be harvested from the non-dominated front (NDF) instead of only one, providing trade-off solutions for the different objectives. Second, the population is of great diversity during the whole evolution, for the reason that individuals with a bad over-all performance survive as long as they are superior in at least one objective. Thus, crossover gains importance by combining differently specialized individuals.

3.1 Variation Operators of the MO-Turtle GA

The variation operators of the *Turtle GA*, reported in [12], are employed, namely the *Random Wires* mutation and the *Implanting Block of Cells* crossover. The implementation of both operators is adapted to the FPTA's architecture and described in the following. A complete description is reported in [12].

Random Wires (Mutation). The mutation operator consists of the create mode and the erase mode. The create mode connects random nodes within the FPTA's transistor array and thereby randomly inserts components into the active circuit. Contrary to that, the erase mode randomly disconnects nodes and removes transistors. The mutation operator is carried out recursively until the resulting circuit contains no dangling nodes and no floating transistor terminals. The width and length of all active transistors is mutated due to a configurable probability.

Implanting a Foreign Block of Cells (Crossover). The *implanting* crossover operator is carried out in two stages. The first stage exchanges randomly sized and positioned rectangular blocks of transistor cells between two randomly selected individuals. While the size of both blocks has to be the same for each individual, the positions of the blocks may differ. Since this operation in general breaks the layout of both previously intact circuits, the second stage fixes the occurring floating nodes by executing the random wires mutation operator for each of them. Thus, again, the resulting circuits contain no floating nodes.

3.2 Non-Dominated Sorting and Crowding Distance

Non-Dominated Sorting. All individuals are classified by calculating their level of non-domination, as shown in Fig. 2, due to their objective values p_i. An individual p is said to dominate q, denoted by $p \preceq q$, if and only if p is partially less than q (Eq. 1).

$$\forall i \in (1, \ldots, n),\, p_\mathrm{i} \leq q_\mathrm{i} \quad \wedge \quad \exists i \in (1, \ldots, n) : \, p_\mathrm{i} < q_\mathrm{i} \tag{1}$$

$$\mathrm{NDF} := \{p \in P \mid \nexists p' \in P : \, p' \preceq p\} \tag{2}$$

All p satisfying Eq. 1, 2 provide the first non-dom. front NDF_1. The succeeding NDFs are found by removing the individuals of NDF_k from the population $P' = P \setminus \mathrm{NDF}_\mathrm{k}$ and by recalculating Eq. 1, 2 for the new population P' until $\mathrm{NDF}_{\mathrm{k+1}}$ is empty.

Crowding Distance. The crowding distance c_{dist} is a measure for the density of solutions within the vicinity of a particular individual p within the fitness landscape (Fig. 2). All objective values are considered for calculating the quantity c_{dist} which represents an average distance to the nearest neighbors of p and is assigned to each individual of the population. Therefore c_{dist} is used to steer the evolution towards a uniform distribution of the individuals over the NDF.

3.3 Evolutionary Step

Three populations are used for evolution: A repository population *RP* and a new population *NP* of size N and an intermediate population *IP* of size $2N$. The algorithm is initialized by randomly generating individuals for *IP* and measuring their objective values. Subsequently, the evolutionary loop is started by performing non-dominated sorting and calculating crowding distances for $IP = RP \cup NP$. The next step is to refill *RP* with the best individuals of *IP* by using tournament selection with the first selection method (SM_1), described in the next subsection, on the obtained NDFs. Hereby, NDF_k is allowed to occupy at most $\frac{1}{2^\mathrm{k}}$ of the available space in *RP*. In case the size of NDF_k is less than or equal to the available space, the whole NDF_k is copied to *RP*. Finally, a new population *NP* of size N is created from *IP* by using tournament selection with SM_2 and applying mutation and crossover.

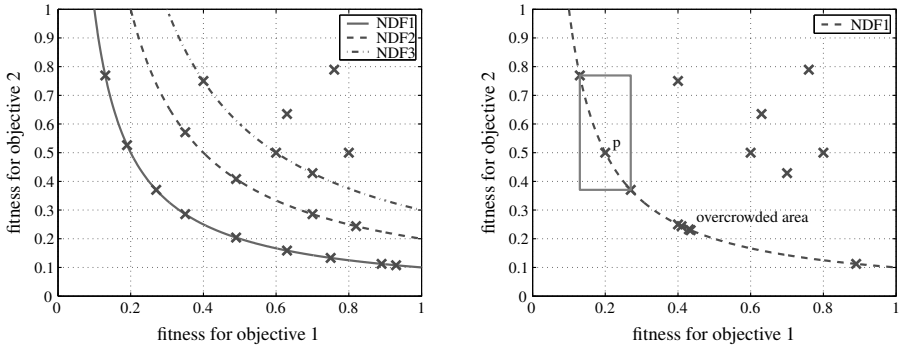

Fig. 2. *Left:* An example set of individuals—which are to be optimized for two objectives—is depicted. The first three NDFs, obtained by evaluating Eq. 1, 2, are drawn in. It is expected that the NDFs propagate towards better fitness values throughout evolution. Additionally, the rank of the NDF is equal to the level of non-domination for each individual of the respective NDF. *Right:* In this example, the individuals are not distributed uniformly over the NDF. Therefore, in order to be able to drive evolution towards such a uniform distribution, a partial order of the individuals within an NDF is defined by the crowding-distance c_{dist}. The value of c_{dist} for an example individual p is derived from the distance to the next neighbors of p.

3.4 Tournament Selection Schemes

Tournament selection with a tournament size of 2 is used as selection scheme. The selection mechanism (SM) for creating the repository is slightly different from that for creating the new population. In the first case (SM_1), the decision which competitor wins is simply based on the comparison between the individuals' level of non-domination and crowding distance c_{dist} (Cond. 3 is true), whereas in the second case (SM_2) it is additionally based on a randomly selected objective and on the main objective (Tab. 1) (more than one of the Cond. 3-5 are true).

These two kinds of tournament selection provide on the one hand high diversity within the repository population by making pure pareto-decisions (SM_1) and, on the other hand, drive evolution to improve single objectives and the main objective (SM_2).

$$p, q \in P : \; p \preceq q \; \vee \; (p = q \; \wedge \; c_{\text{dist}}(p) > c_{\text{dist}}(q)) \tag{3}$$

$$\text{Fitness}(p_{\text{main-objective}}) < \text{Fitness}(q_{\text{main-objective}}) \tag{4}$$

$$\text{Fitness}(p_{\text{random-objective}}) < \text{Fitness}(q_{\text{random-objective}}) \tag{5}$$

4 Experimental Setup

The experiments are run at a generation size of 200 for *IP* and a number of 4000 generations per evolution run. Individuals are mutated with a probability of 0.6 and crossover is carried out with a probability of 0.4 and a maximum block-size of 4×4 transistor cells. An area of 9×9 transistor cells is provided to the evolving circuit. Both, the non-inverting (I_+) and the inverting (I_-) input of the circuit are statically connected to the gate of a transistor of the same flavor, in order to avoid meaningless amplifiers. Two series of experiments, each of 20 evolution runs, are carried out using PMOS input in the

TM	objective	fitness	description
TM_1	pull to rails	min.	$(V_{\text{tar}} - V_{\text{out}})^2$ (main objective)
TM_1	DC offset	min.	sum of DC offsets of the set of curves
TM_1	dev. of DC offset	min.	standard deviation of the DC offsets
TM_2	slew-rate	max. (use recip.)	sum of slew-rates of all steps
TM_2	settling-time	min.	time when V_{out} settles within $\pm 10\%$ of V_{tar}
TM_2	deviation from V_{tar}	min.	$(V_{\text{tar}} - V_{\text{out}})^2$
TM_3	magnitude	max. (use abs.)	damping of the fundamental frequency at unity gain
TM_3	harmonic distortion	min.	sum of ampl. of harmonics if above $-60dB$
TM_4	phase-shift	min.	phase-shift of sin between V_{out} and $V_{\text{I+}}$
TM_4	sin-curve deviation	min.	$(V_{\text{tar}} - V_{\text{out}})^2$
—	resource consumption	min.	sum of used transistors

Table 1. An overview of all objectives. The aim is to minimize the fitness; thus, in the cases where the objective value is to be maximized, the reciprocal or absolute value is used as fitness. *Pull to rails* is chosen as main objective, for the reason that it judges a fundamental behavior of an amplifier and the fitness-value improves smoothly.

first case and NMOS input in the second case. Free resources of the transistor array are used to attach a randomly (by mutation) variable capacitive load to the circuits output and to implement two test benches for the circuit under test: One for open loop testing and another one with full feedback to the inverting input. Thus, a gain of 1 is assumed for the latter. Since the feedback is realized using only the configuration capabilities of the transistor array—where no constant resistors, capacities or current sources are available—it is not feasible to measure properties like gain or common-mode rejection ratio (CMRR) directly on the chip. Nevertheless it is possible to measure and evaluate important properties of an amplifier, namely open-loop behavior, slew-rate, settling-time, DC offset, harmonic distortion and phase-shift, directly on the FPTA.

4.1 Test Modes for the Measurements on the FPTA

Three kinds of test-modes (TM_i) have been developed to perform these measurements delivering a total of 11 objective values listed in Tab. 1.

TM_1: Open-Loop Behavior, Offset. The task is to pull V_{out} to $V_{tar} = 5V$ if $V_{I+} > V_{I-}$ and to $V_{tar} = 0V$ if $V_{I+} < V_{I-}$ and to keep the offset voltage V_{os} low or at least constant. A set of nine curves at $V_{I+} = 1.5, 1.75, \ldots, 3.5V$, each consisting of 100 randomly applied sample voltages for $V_{I-} = 0 \ldots 5V$, is used as test pattern. TM_1 delivers fitness values for three objectives, namely *pull to rails*, *DC offset* and *deviation of DC offset*.

TM_2: Slew-Rate, Settling-Time. The challenge for the output is to follow two voltage-steps from $V_{I+} = 1.5V$ to $2.5V$ and from $V_{I+} = 2.5V$ to $3.5V$ in $t_{step} = 0.25\,\mu$s. Fitness values for the *slew-rate* and the *settling-time* are calculated from the period of time between the step and the point of time when V_{out} has settled at the new target voltage $V_{tar} \equiv V_{I+}$. An additional objective is given by the *deviation of V_{tar} from V_{out}*.

TM_3 & TM_4: Magnitude, Phase-Shift, Harmonic Distortion. A further demand on an OP is to distort and damp the input signal as less as possible and to keep the phase-shift constant below $180\,^\circ$ in order to cause the amplifier to remain stable. These properties are measured in TM_3 by applying three different sinusoidal signals with $f = 5, 50$ and $500\,\mathrm{kHz}$ to the input and comparing them to the circuits output $V_{tar} \equiv V_{I+}$. A discrete fourier transform is used to calculate the power spectrum of the output signal for each frequency. Subsequently, fitness values for *magnitude* and *total harmonic distortion (THD)* are calculated from the power spectrum. Additionally, the output of a sinusoidal input signal of $f = 20\,\mathrm{kHz}$ is used in TM_4 to obtain values for the *phase-shift* and the *deviation of V_{I+} from V_{out}*.

4.2 Simulation Setup

The simulations are carried out with the SPICE3 simulator described in [17]. BSim3 transistor models are used for simulation. SPICE netlists are extracted from the circuits that have been evolved on the transistor array by using the *MO-Turtle GA*. The input voltage patterns correspond to those used for the on-chip measurements. A load-capacity of $10\,\mathrm{pF}$ is attached to the circuits' output in simulation. Fitness values, calculated from the simulation results, are obtained by using the same fitness functions as throughout evolution.

5 Results

All evolution runs ended up in similar regions of fitness, although the overall performance of the circuits is slightly better for those with NMOS input than for those with PMOS input, as can be seen from Tab. 2. For all evolved circuits the simulation results

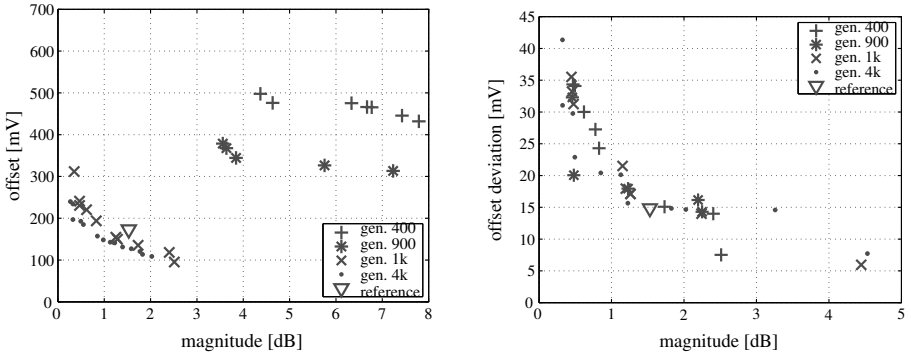

Fig. 3. An example run (NMOS input) with good performance is chosen and the depicted NDFs are recalculated by considering only the two objectives shown in the respective plot for illustration. The position of a manually made OP (reference), described in Sect. 5.2, is marked by a triangle. *Left:* The NDF for *offset* over *magnitude* converges towards better fitness over time. *Right:* In contrast to this, the NDF for *dev. of offset* over *magnitude* is spread over wide ranges of fitness.

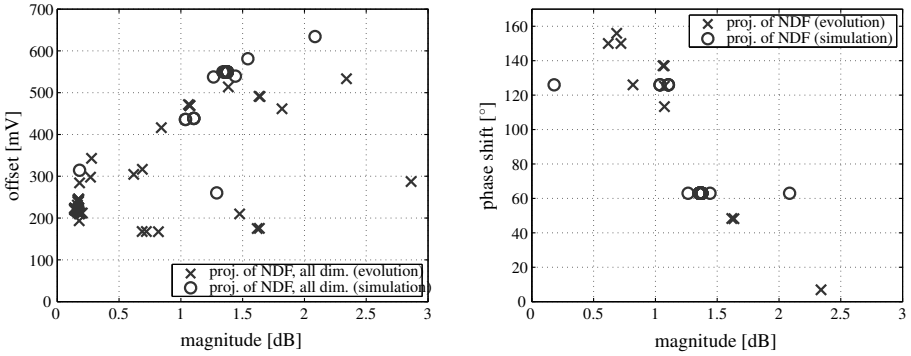

Fig. 4. The NDFs of a run (NMOS input) with good performance, obtained from the measuring on the FPTA and from the simulation, are depicted above. The graphs show a multi-dimensional projection of the NDF into the plane spanned by the respective objectives. *Left: Offset* over *magnitude*. *Right: phase-shift* over *magnitude*.

are worse than those obtained from the chip and about half of them did not work at all outside the FPTA. Nevertheless, each evolutionary run features a significant amount of individuals performing at least similar in simulation and on the transistor array. Example NDFs for the resulting circuits are depicted in Fig. 3 and 4.

5.1 Performance of the Multi-Objective Approach

An example of how the NDF develops throughout evolution is depicted in Fig. 3. For some objectives (e.g. *magnitude*, *offset*) the NDF converges towards better fitness values over time, as can be nicely seen from Fig. 3 (left). Other objectives (e.g. *magnitude*, *offset-deviation*) show a different behavior where the front as a whole does not further converge, but is spread over wide ranges of fitness. An example for the latter is shown in Fig. 3 (right). Additionally, the position of a manually created design, described in Sec. 5, within the objective space is marked by a triangle.

Projections of the whole NDF into the plane spanned by the respective objectives—taking all objectives into account for computation—are graphed in Fig. 4. This illustrates nicely the complexity of the NDF troughout the optimization process. After all, the main benefit of using an MO approach for the evolution of operational amplifiers on the Heidelberg FPTA is the possibility to efficiently explore the search space taking care of both, the diversity of the population and the various demands on the target circuit.

5.2 Solutions for the Operational Amplifier

The FPTA is configured with manually created circuits, one with PMOS and one with NMOS input respectively, in order to be able to assess the quality of the synthesized circuits compared to human-made solutions. Each of the references consists of a differential input stage and a simple inverter-output stage. The fitness values are measured for both reference designs, using exactly the same setup as throughout evolution, and are compared to those of the evolved circuits. As can be seen from Tab. 2, almost each run contains at least one individual that outperforms the corresponding reference OP in up to 3 objectives and about 5 runs feature similar performance in up to 5 objectives. In all cases the manually made OPs obtained better fitness values for *distortion* (noise) and *resource consumption* than the evolved circuits. The reason for this is the placement and routing of the evolved solutions which often contain longer wires and therefore produce more noise.

Opposite to the competition with the reference circuits on the FPTA, the evolved circuits come off worse if typical characteristics of OPs are compared in simulation. As can be seen from Tab. 3 especially those properties that cannot be measured directly on the transistor array during evolution—thus, cannot be evaluated by a fitness function (e.g. open-loop gain)—return rather poor results. Contrary to that, the characteristics that are represented by an objective perform similar, e.g. *offset*, *slew-rate* and *settling-time*. Since the output voltage swing and the 0dB bandwidth are correlated to a good open-loop gain, those values are also not as good as those of the manually made OPs.

In both cases the phase-margin of the evolved solution is higher than those of the reference OPs. This is interesting insofar, that it is on the one hand a very good result, since the aim of the corresponding objective is to minimize the phase-shift. On the

		in no. of objectives										
		1	2	3	4	5	6	7	8	9	10	11
no. of NMOS runs	better than ref.	20	20	18	5	1	0	0	0	0	0	0
	max. 10% worse than ref.	20	20	18	5	5	2	1	1	0	0	0
no. of PMOS runs	better than ref.	20	20	18	4	2	1	0	0	0	0	0
	max. 10% worse than ref.	20	20	19	6	5	3	1	1	1	0	0

Table 2. The no. of runs that contain at least one individual that achieved a better (or not more then 10% worse) fitness value than the manually made circuits for a given no. of objectives. In all cases the manually made OPs obtained better fitness values for *distortion* and *resource consumption* than the evolved circuits. The reason for this is the placement and routing of the evolved solutions which often contain longer wires and therefore produce more noise.

other hand, forcing the phase-shift towards zero could possibly thwart the evolution of output gain-stages. If this is the case, it would be better to allow for a certain phase-margin in the objective function. Hence, this could be the reason why in both examples depicted in Fig. 5—which represent evolved circuits with good performance—the algorithm was able to synthesize clearly recognizable differential input stages as well as biasing circuitry, but failed in appending a simple inverter, which would provide significantly better performance. Finally, some important characteristics of the evolved circuits are shown in Fig. 6.

6 Conclusion and Outlook

The main achievement of the presented method is that reusable and substrate-independent circuits are evolved successfully and human-understandable schematics of good solutions can be drawn. Hence, it is possible to analyze the resulting circuits and to

parameter	NMOS (evo)	NMOS (ref)	PMOS (evo)	PMOS (ref)
open-loop gain	$33\,\mathrm{dB}$	$57\,\mathrm{dB}$	$29\,\mathrm{dB}$	$65\,\mathrm{dB}$
0dB bandwidth	$13\,\mathrm{MHz}$	$77\,\mathrm{MHz}$	$6\,\mathrm{MHz}$	$33\,\mathrm{MHz}$
offset	$-80\,\mathrm{mV}$	$28\,\mathrm{mV}$	$230\,\mathrm{mV}$	$20\,\mathrm{mV}$
slew-rate (+)	$40\,\frac{\mathrm{V}}{\mu\mathrm{s}}$	$100\,\frac{\mathrm{V}}{\mu\mathrm{s}}$	$15\,\frac{\mathrm{V}}{\mu\mathrm{s}}$	$25\,\frac{\mathrm{V}}{\mu\mathrm{s}}$
slew-rate (-)	$15\,\frac{\mathrm{V}}{\mu\mathrm{s}}$	$30\,\frac{\mathrm{V}}{\mu\mathrm{s}}$	$35\,\frac{\mathrm{V}}{\mu\mathrm{s}}$	$45\,\frac{\mathrm{V}}{\mu\mathrm{s}}$
settling-time	$0.4\,\mu\mathrm{s}$	$0.2\,\mu\mathrm{s}$	$0.3\,\mu\mathrm{s}$	$0.2\,\mu\mathrm{s}$
phase-margin	$91\,^\circ$	$50\,^\circ$	$92\,^\circ$	$50\,^\circ$
common mode rejection	$30\,\mathrm{dB}$	$> 40\,\mathrm{dB}$	$20\,\mathrm{dB}$	$> 40\,\mathrm{dB}$
out voltage swing	$2.2\,\mathrm{V}$	$4.8\,\mathrm{V}$	$2.8\,\mathrm{V}$	$4.8\,\mathrm{V}$
input common mode range	$2.5\,\mathrm{V}$	$4.2\,\mathrm{V}$	$3.5\,\mathrm{V}$	$4.3\,\mathrm{V}$

Table 3. Comparison between characteristics of evolved circuits with a good performance and the reference circuits (NMOS and PMOS input). The values are obtained from SPICE simulations.

investigate how the algorithm is solving problems on the hardware substrate. As an example, it has been shown that the presented algorithm is able to synthesize operational amplifiers on the Heidelberg FPTA. The fact that the evolution of OPs is a difficult task suggests that the *MO-Turtle GA* can be applied to a variety of problems.

The resulting circuits are extracted into netlists and simulated outside the substrate on which they were evolved. About 50% of the outcome is performing equally well on the chip and in simulation and can therefore be transferred to other technologies. The presented multi-objective approach allows for considering various objectives during evolution. Thus, it is possible to efficiently explore the design space and converge to regions of fitness comparable to those which are obtained by basic human reference designs measured on the chip. Unfortunately, the algorithm failed in synthesizing additional gain-stages. The reason for this is probably the lack of a suitable gain test bench due to the fact that even well approved human designs do not achieve significantly better fitness. In this case it is more likely that the abilities of the FPTA limit the search for good solutions than the algorithm itself. This indeed, will only be solved by a second generation FPTA.

Future work will be done to understand to what extent the architecture of the transistor array influences the performance of the algorithm and what can be done to improve it. Furthermore, the *MO-Turtle GA* will be enhanced to allow the creation and deletion of structures like differential pairs or inverters in one step. Hereby, all transistors of those structures could be marked for a simultaneous W/L mutation.

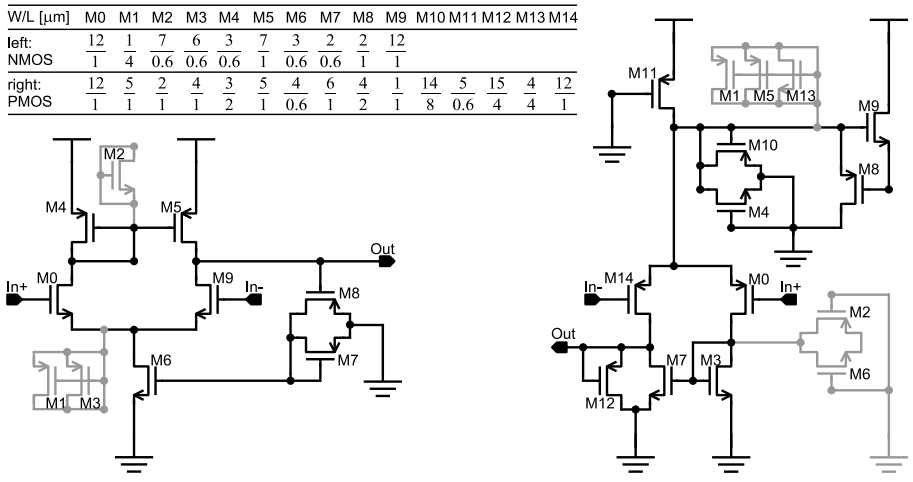

Fig. 5. Schematics of the evolved circuits; shorted transistors are grayed. *Left*: NMOS input transistors. *Right*: PMOS input transistors. In both cases the *MO-Turtle GA* achieved to synthesize differential input stages and some kind of biasing circuitry. The evolved solutions thus far lack of an output gain-stage.

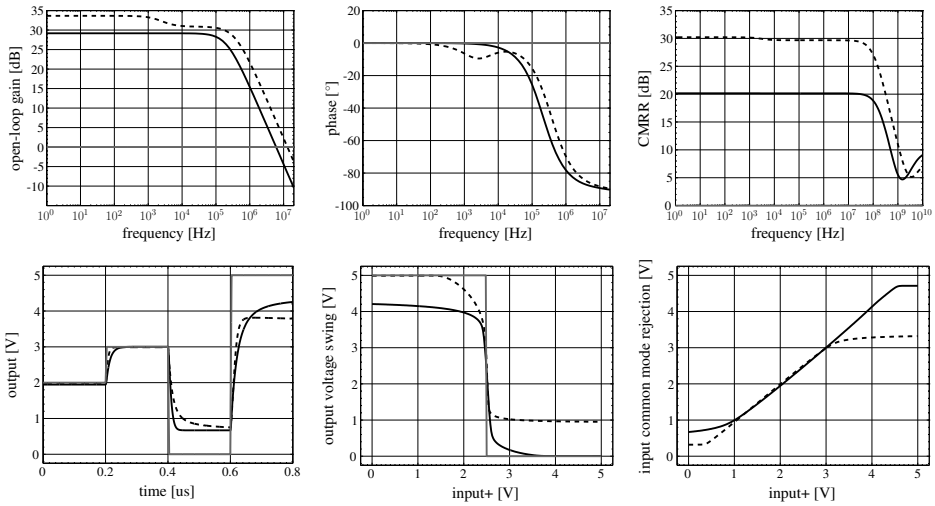

Fig. 6. The graphs above show characteristics of evolved operational amplifiers obtained from spice simulation. For illustration, evolved circuits with PMOS (—) and NMOS (- - -) input respectively and featuring good performance are chosen.

7 Acknowledgment

This work is supported by the Ministerium für Wissenschaft, Forschung und Kunst, Baden-Württemberg, Stuttgart, Germany.

References

1. M. Hershenson, S.B., Lee, T.H.: Optimal design of a cmos op-amp via geometric programming. In: IEEE Transactions on Computer-Aided Design. (2001) 1–21
2. Arpad Buermen, Janez Puhan, T.T.: Robust design and optimization of operating amplifiers. (2003) 745–750
3. Vieira, P.F., Botelho, L.B., Mesquita, A.: Evolutionary synthesis of analog circuits using only mos transistors. In Zebulum, Ricardo S., Gwaltney, David, Hornby, Gregory, Keymeulen, Didier Lohn, Jason, and Stoica, Adrian, ed.: Proceedings of the 2004 NASA/DoD Conference on Evolvable Hardware, Los Alamitos, IEEE Computer Society Press (2004) 38–45
4. Sripramong, T., Toumazou, C.: The invention of cmos amplifiers using genetic programming and current-flow analysis. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **21** (2002) 1237–1252
5. Koza, J.R., Bennett III, F.H., Andre, D., Keane, M.A.: Design of a high-gain operational amplifier and other circuits by means of genetic programming. In In Angeline, Peter J., Reynolds, Robert G., McDonnell, John R., and Eberhart, Russ, ed.: Evolutionary Programming VI. 6th International Conference, EP97, Proceedings. Volume 1213 of Lecture Notes in Computer Science., Indianapolis, Indiana, USA, Springer-Verlag, Berlin (1997) 125–136

6. Koza, J.R., Bennett III, F.H., Andre, D., Keane, M.A.: Evolution using genetic programming of a low-distortion 96 decibel operational amplifier. In: Proceedings of the 1997 ACM Symposium on Applied Computing, San Jose, California, USA, New York: Association for Computing Machinery (1997) 207–216
7. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: Nsga-ii. In: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature. (2000) 849–858
8. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York (2002)
9. Hernandez Aguirre, A., Zebulum, R.S., Coello Coello, C.A.: Evolutionary multiobjective design targeting a field programmable transistor array. In Zebulum, Ricardo S., Gwaltney, David, Hornby, Gregory, Keymeulen, Didier Lohn, Jason, and Stoica, Adrian, ed.: Proceedings of the 2004 NASA/DoD Conference on Evolvable Hardware, Los Alamitos, IEEE Computer Society Press (2004) 199–205
10. Zebulum, R., Pacheco, M., Vellasco, M.: A novel multi-objective optimization methodology applied to the synthesis of cmos operational amplifiers. Journal of Solid-State Devices and Circuits (2000) 10–15
11. Zebulum, R.S., Pacheco, M.A., Vellasco, M.: A multi-objective optimisation methodology applied to the synthesis of low-power operational amplifiers. In Cheuri, I.J., dos Reis Filho, C.A., eds.: Proceedings of the XIII International Conference in Microelectronics and Packaging. Volume 1., Curitiba, Brazil (1998) 264–271
12. Trefzer, M., Langeheine, J., Schemmel, J., Meier, K.: New genetic operators to facilitate understanding of evolved transistor circuits. In Zebulum, R.S., Gwaltney, D., Hornby, G., Keymeulen, D., Lohn, J., Stoica, A., eds.: Proceedings of the 2004 NASA/DoD Conference on Evolvable Hardware, Los Alamitos, IEEE Computer Society Press (2004) 217–224
13. Langeheine, J., Becker, J., Fölling, S., Meier, K., Schemmel, J.: A CMOS FPTA chip for intrinsic hardware evolution of analog electronic circuits. In: Proc. of the Third NASA/DOD Workshop on Evolvable Hardware, Long Beach, CA, USA, IEEE Computer Society Press (2001) 172–175
14. Garvie, M.: Reliable Electronics through Artificial Evolution. PhD thesis, University of Sussex (2004)
15. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989)
16. Deb, K., Goel, T.: Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence. In Zitzler, E., Deb, K., Thiele, L., Coello, C.A.C., Corne, D., eds.: First International Conference on Evolutionary Multi-Criterion Optimization, Springer-Verlag. Lecture Notes in Computer Science No. 1993 (2001) 67–81
17. Quarles, T., Newton, A., Pederson, D., Sangiovanni-Vincentelli, A.: SPICE3 Version 3f3 User s Manual. Department of Electrical Engineering and Computer Sciences, University of California Berkeley, Ca., 94720. (1993)