A Multi-Objective Optimisation Methodology Applied to the Synthesis of Low-Power Operational Amplifiers

Ricardo S. Zebulum[1,2] Marco Aurélio Pacheco[2,3] Marley Vellasco[2,3]

[1] CCNR, University of Sussex, Brighton, BN1 9SB UK, e-mail:ricardoz@cogs.susx.ac.uk
[2] ICA - Pontificia Universidade Catolica do Rio de Janeiro - Brasil
[3] Depto de Engenharia de Sistemas e Computação, UERJ -RJ, Brasil

Abstract. This work studies the problem of CMOS operational amplifiers design optimisation. The synthesis of CMOS amplifiers can be translated into a multiple-objective optimisation task, in which a large number of specifications has to be taken into account, i.e., GBW, area, power consumption and others. We apply Genetic Algorithms [7] (GAs) to this problem; GAs are a computational optimisation technique which borrows some principles from biological evolution and have been widely applied to Computer Aided Design (CAD) of electronic circuits. A novel multi-objective optimisation methodology is embedded in our genetic algorithm and we focus mainly on the synthesis of micro-power analog cells.

1 Introduction

We present a novel methodology applied to the problem of analog CMOS cells optimisation. Particularly, we tackle the issue of synthesising low-power operational amplifiers. The acquisition of micropower analog circuits is a major tendency in the electronics industry nowadays and, analog circuitry, though constituting only a small part of the total area of modern chips, is usually the limiting factor for the overall chip performance [8].

We use the Genetic Algorithms optimisation technique [7] in this problem. Genetic Algorithms or GAs have already been employed in many CAD applications, such as digital logic optimisation [6], FPGA mapping [9], floorplanning [1], global routing [3] and others [2]. This search technique can be successfully applied to a class of optimisation problems in which the search space is too large to be sampled by conventional techniques. In the particular context of this application, they will perform a search over the space of analog OpAmps with different *transistor sizes* and *biasing currents*.

Our methodology is tested in problems of micropower OpAmps design optimisation, targeting the OTA, Cascode OTA and class A topologies. The transistors' operating regions are analysed and compared with standard low-power strategies. It is shown that the results are competitive with human designed cells, though our GAs have used nearly no kind of human knowledge in the optimisation process.

This work consists of four additional sections: section 2 describes the problem of CMOS operational amplifiers synthesis; section 3 describes the methodology the authors used to cope with the problem, as well as others authors' approaches; section 4 presents the case studies and section 5 concludes the work.

2 OpAmp Design Optimisation

Whereas in bipolar based design the designer's creativity is in the conception of different topologies, in the case of CMOS design the creativity is used to set the transistors' sizes of a particular topology, and, as a consequence, select also the transistor's operating regions [15].

In the particular case of OpAmp design optimisation, a large set of specifications must be achieved by the design process. The designer must decide which specification will carry the largest weight [12]. It is common that, when a circuit is optimised to fulfill only one or two specifications, the other characteristics may become nonsense [12]. It is important in an optimisation procedure to decide beforehand which specifications should be used in the procedure [12].

Ideally, all the OpAmp specifications should be included, but, usually, only the most important ones are taken into account [12]. The authors decided to include the following specifications in our genetic algorithm system: *gain, GBW, linearity, power consumption, area and phase margin.*

By searching for a particular set of transistor sizes and biasing currents, the GA determines the transistor operating regions. MOS transistors may operate in *strong, weak and moderate inversion* [8]. The strong inversion region, in which V_{GS} - V_{TN} is greater than 0.2V, is the most commonly used in analog design. The weak inversion region, in which V_{GS} - V_{TN} is around 100 mV, is characterised by the low-power consumption of the device, being then employed in low-power applications. Nonetheless, this operating region is also characterised by the following drawbacks: low speed of the device, increase in its area and reduction in the load driving capacity. Another way to characterise the weak inversion region is by the following equation:

$$I_D < 2.n\beta(U_T)^2 \tag{1}$$

, where β is proportional to the transistor dimensions W/L, U_T is the thermal voltage (26 mV at 300k) and n is usually around 1.3. This equation makes explicit the relationship between the transistor's dimensions and its operating regions. The moderate inversion is intermediate between the weak and strong inversion regions.

3 Optimisation Methodology

Due to the size of the search space and to the multi-objective nature of the problem, the authors decided to use genetic algorithms to perform the task.

GAs carry out optimisation through biological evolution simulation [2]. Instead of focusing on just one potential solution, GAs sample a population of potential solutions. A population of individuals is, initially, randomly generated. Each individual is a string that encodes, by means of a particular mapping, a potential solution to the problem. The GA performs then operations of *selection*, *crossover* and *mutation* on the individuals, corresponding to the principals of survival of the fittest, recombination of genetic material and random mutation observed in nature [7]. The selection step is probabilistic, but it favours individuals which have been assigned higher fitness indexes in the fitness evaluation step, performed beforehand. The fitness is a scalar measure of the individual performance. The crossover operator splices the contents of two strings and is the main driving force of the GA [7]; the mutation operator changes, with low probability, a particular string position, and it is regarded as a background operator. The optimisation process is carried out through the generation of successive populations until a stop criteria is met. The basic GA flow is illustrated in Figure 1.

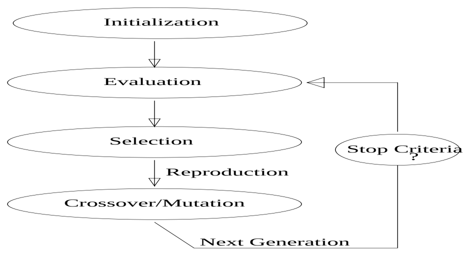

Fig. 1. Basic Genetic Algorithm Flow

The following sections examine the *representation* and the *multi-objective fitness assignment* used in our GA. The other GA parameters have been chosen according to classical GA references, and further details can be found in [7].

3.1 Representation

Each genetic algorithm string (also called individual or genotype) is made up of integer numbers encoding a particular sized OpAmp. The genotype positions are the values of the transistor sizes, biasing current and, if it is the case, compensating capacitor. This representation is illustrated in Figure 2.

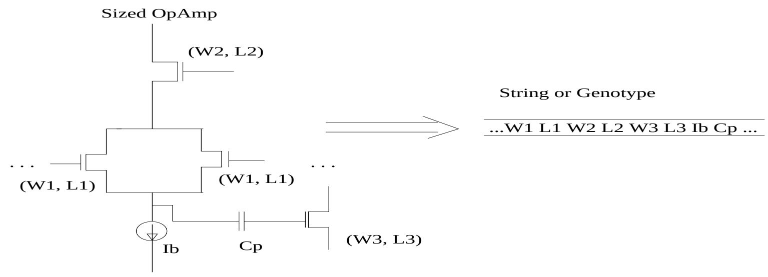

Fig. 2. OpAmp representation

This representation uses a minimum amount of designer knowledge. We have only constrained the differential input transistors to be equally sized, to avoid meaningless OpAmps. Furthermore, the transistors dimensions and biasing current were limited to reasonable values, i.e., $W > W_{min}$ and $L > L_{min}$. More specifically, we allowed W and L to take a hundred discrete values, from W_{min} (L_{min}) to W_{min} + 100 (L_{min} + 100). The biasing current, I_b, and the compensating capacitance, C_p, have also been allowed to take a hundred different values in the following way:

$$0.01\mu A \leq I_b \leq 1\mu A, \quad step \quad 0.01\mu A \quad , for \quad low-power \tag{2}$$

$$1.5\mu A \leq I_b \leq 2.5\mu A, \quad step \quad 0.01\mu A, \quad otherwise \tag{3}$$

$$0.1pF \leq C_p \leq 10pF, \quad step \quad 0.1pF \tag{4}$$

For standard OpAmp cells, with around 10 transistors, this codification results in a search space of the order of $100^{10} = 10^{20}$ possible solutions. This search space is in the range of GA applications [2], since they perform an efficient sampling over large search spaces [7].

3.2 Multi-Objective Evaluation Function

The main challenge of applying genetic algorithms or any other optimisation technique to this problem is the multi-objective nature of the same. The problem of multi-objective optimisation concerns the need to integrate vectorial performance measures with the inherently scalar way in which GAs rewards individual performance [4]. Because genetic algorithms require scalar fitness information on which to work, a scalarization of the objective vectors is always necessary.

Many studies have already been carried out in multi-objective optimisation [4], but, so far, no standard methodology to handle the problem has been achieved. Two types of multi-objective optimisation approaches can be identified [4]:

1. Plain aggregating approaches, consisting of the popular weight-sum approach, as shown in the following equation:

$$Fitness = \sum_{i=1}^{n} w_i f_i \tag{5}$$

   where f_i is the fitness component of each objective, w_i is the respective weight and n is the total number of objectives. The main problem of this approach is the specification of the weights
2. Pareto-Based Approaches - these approaches are based on the Pareto concept of dominance. According to this concept, an individual v_i dominates another individual u_i if and only if:

$$\forall i \in (1, ..., n), v_i \leq u_i \quad \cap \quad \exists i \in (1, ..., n), v_i < u_i \tag{6}$$

which leads to the concept of Pareto-optimal set. In the above equation n is the number of individuals and we assume a minimisation task. The Pareto-based fitness assignment has been first proposed by Goldberg [7], as a means of assigning equal probability of reproduction to all non-dominated points in the population. Fonseca and Flemming [4] used this concept by assigning the individual's rank to the number of individuals in the current population by which it is dominated.

Particularly, CAD problems are intrinsically multi-objective [2] and some tools have been developed to handle applications in the area. The software EXPLORER [1] is a GA based tool that minimises chip layout area, deviation from a target aspect ratio, routing congestion and maximum path delay in VLSI cells; these four objectives are handled through a Pareto based approach.

The authors applied the former approach in this work. The main problem of the aggregating methodology relies on the setting of the weights associated to each objective. In order to overcome this problem, we used adaptive weights along the optimisation process, in the sense that their values will be adapted according to *the average fitness value with respect to each objective and to the users specification for each objective*. The following set of equations summarises our multi-objective evaluation strategy. We will refer to fitness as the score achieved by a circuit regarding to a particular objective, while the overall fitness is the aggregation of all objectives' scores.

$$\text{Overall Fitness} = \sum_{i=1}^{n} w_i Fn_i \tag{7}$$

$$Fn_i = \frac{F_i}{< F_i >} \tag{8}$$

$$w_i = \frac{100 \;\; user_i}{< F_i >} \qquad if < F_i > \; < \;\; user_i \tag{9}$$

$$w_i = 1, \;\; otherwise \tag{10}$$

if objective i is to be maximised, or:

$$w_i = -\frac{100 \;\; < F_i >}{user_i} \qquad if < F_i > \; > \;\; user_i \tag{11}$$

$$w_i = -1, \;\; otherwise \tag{12}$$

if objective i is to be minimised.

Equation 7 shows the overall fitness expression, which aggregates the fitness corresponding to all the objectives. n is the number of objectives, w_i is the weight vector and Fn_i is the normalised fitness vector.

The normalised fitness vector, Fn_i, is given by equation 8 as the ratio between the actual scored fitness with respect to objective i, F_i, and the respective average fitness value over all individuals in the population, $< F_i >$. The normalisation is to account for the fact that the objectives are measured in different units and all of them must have the same influence in the overall fitness.

The weight vector expression is defined from equations 9 to 12. If a particular objective is to be maximised, its weight is defined as the ratio between the desired specification, $user_i$, and the current average fitness value $< F_i >$, for a particular objective i. This ratio is multiplied by an amplification factor of 100. If the objective is to be minimised, it takes a negative value and the previous ratio is inverted. The amplification factor enhances the influence of the weight w_i as long as the objective is not satisfied. When the objective is satisfied ($< F_i > \; > \;\; user_i$ for maximisation and $< F_i > \; < \;\; user_i$ for minimisation), the weight is set to 1.

Summarising this technique, the idea is to assign large weights to objectives for which the average fitness is far from the target value, and low weights to objectives whose average values are around the desired ones. The search will then be driven by unsatisfied design requirements.

In order to illustrate the effectiveness of our methodology, the graphs of Figure 3 illustrate the results for a particular run for the optimisation of a low-power tranconductance amplifier (OTA). The graphs show the values of each particular objective along 80 generations; the objectives are gain, GBW, bias voltage, power consumption, area and phase margin. Gain, GBW and phase margin have to be maximised and the other ones have to be minimised. Minimising the bias voltage is the means whereby we improved linearity. It can be verified that our methodology acts in such a way to try to satisfy all the objectives. For this particular example, we set the target vector as: minimum 100dB gain, minimum 300 kHz unit gain frequency, maximum 0.3 V bias, maximum Power consumption of 10 μW, maximum area of 10,000 μm^2 and minimum phase margin of 50^o. Except for the gain, which achieved a value close to the specification, all the objectives have been satisfied. We also observe that graph F shows the minimisation of the variable $PM* = |90 - PM|$, corresponding to trying

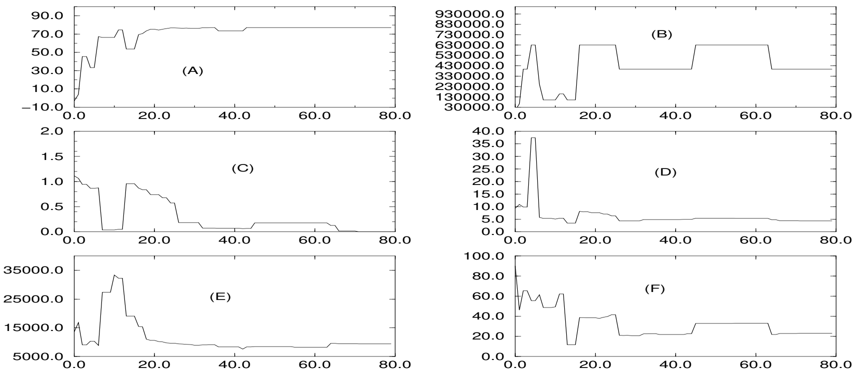

Fig. 3. Six objective evolution for a transconductance amplifier: axis X is the number of generations; Axis Y is Gain in dB, band in Hz, bias voltage in Volts, dissipation in μ W, area in μm^2 and $PM*$ in degrees for graphs (A), (B), (C), (D), (E) and (F) respectively.

to place the phase margin close to 90^o. Values of PM above 45^o are necessary to keep the amplifier stable.

Finally, Figure 4 provides a broader view for the case of two-objectives evolution. The authors selected as objectives gain and $PM*$, and plotted the population distribution for generations 1, 5 and 20. The population size is of 30 individuals (some points may represent the group of many individuals). It can be seen that, in the beginning of the process (graph (A)), there are no individuals with values of $PM*$ smaller than 10^o or a gain higher than 20 dB. As evolution proceeds, the points start to move to the bottom and to the right within the graph. In generation 20 (graph (C)), circuits with 80 dB gain and $PM*$ values of less then 10^o (corresponding to a phase margin of 80^o) can be found. It can be verified that, only a few generations is sufficient when the number of objectives is small. It can also be verified that GAs provide the designer with a population of choices, instead of only one solution [1].

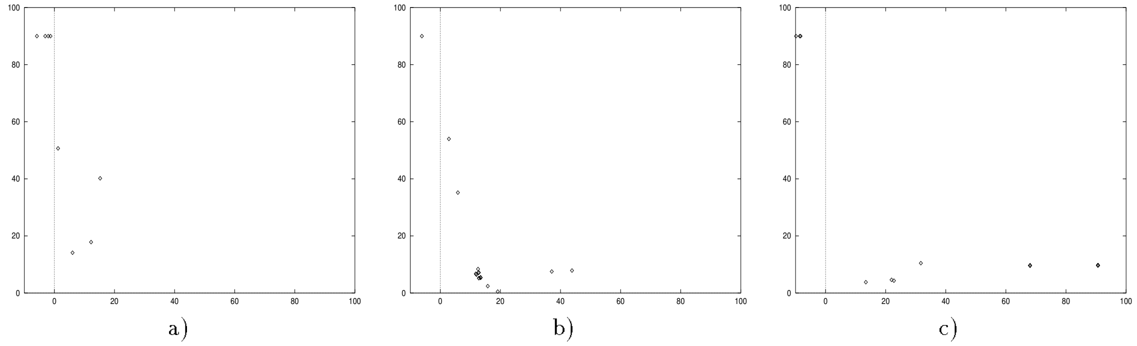

Fig. 4. Population distribution for a two objective evolutionary process; axis X accounts for the gain in dB and axis Y accounts for $PM*$ in degrees; Graphs (A), (B), (C) correspond to generations 1, 5, and 20 respectively.

3.3 Related Works

We compare our representation and evaluation function with other works in this section. Kruiskamp et. al. [10] have developed a GA based tool called Darwin, which optimises operational amplifier designs by selecting its constituting blocks from standard amplifier stages. Transistor sizes have also been optimised within some constraints. The optimisation objectives were gain, unit gain frequency, area, power consumption and other OpAmp statistics. When a specification of an OpAmp did not meet the user requirements, this resulted in a negative contribution to the fitness of the OpAmp, proportional to the gap between specification and requirement. When all requirements were satisfied, the fitness function was defined by an optimisation function based on power dissipation. Instead of representing the individuals as strings whose positions consist of the transistor sizes, they have mapped each OpAmp into a binary string encoding the solution of a set of linear equations. The disadvantages of this representation are that all transistors are constrained to operate in the saturation region and designer previous knowledge is being used to solve the set of equations. The first disadvantage is more serious in the case of low-power design.

Rogenmoser and fellows [13], from the Swiss Federal Institute of Technology, have applied Genetic Algorithms to optimise transistor sizes of CMOS digital circuits. Their work focused on digital cells, such as NANDs and Flip-Flops, and the fitness evaluation function has taken into account delay and power consumption requirements. For some cells, the aggregating approach to multi-objective optimisation has been used, whilst for other cells, the product of delay and power consumption has been employed as a fitness statistics. Through comparison with standard optimisers and the Monte Carlo scheme, they have found out that genetic algorithms were more stable for larger circuits.

4 Case Studies

Our experiments have focused on three different OpAmp topologies: class A operational amplifier, Transconductance Amplifier (OTA) with simple output and OTA with Cascode output. The SMASH simulator [17] has been used to evaluate the OpAmps' performance; the level 5 CMOS transistor model, which simulates more accurately the weak inversion region, has been used in the applications.

The fitness evaluation function has been designed to optimise gain, GBW, linearity, power consumption, area and phase margin, in the way shown in the previous section. Table 1 shows the statistics of the experiments, describing the transistor technology used, magnitude of the search space, gain, GBW, bias value, biasing current, DC voltage source value, slew-rate [1], offset, power consumption, area and phase margin of each amplifier. The load has been set to 2 pF for all tests.

The target objective vector, in the form (Gain(dB), GBW(kHz), Bias (Volts), Power Consumption (μW), Area (μm^2), PM (degrees)), has been set to: (120, 200, 0.5, 8 , 6,000, 60) to the class A opamp, (70, 200, 0.5, 5, 5,000, 60) to the OTA and (90, 200, 0.5, 15, 8,000, 60) to the Cascode OTA. These specifications are on-line with typical human designed low-power OpAmps [14] [11].

Features	Class A OpAmp	Simple OTA	Cascode OTA
Search Space	10^{36}	10^{34}	10^{58}
Gain	141dB	79dB	100dB
GBW	900 kHz	90 kHz	210 kHz
Bias Voltage	0.25V	0.45V	0.25V
Offset	-1.85mV	0.67μV	65.9nV
Slew-Rate	340mV/us	46.7 mV/us	200mV/us
Biasing Current	0.5μA	0.26μA	0.75μA
DC Voltage Source	1.5V	1.5V	1.5V
Power Consumption	10.54μW	2.45μW	33.8μW
Area	6,218μm^2	10,085 μm^2	16,920 μm^2
Phase Margin	55^o	45^o	60^o
Technology	Marin 3μ	AMS 1.2μ	AMS 1.2μ

Table 1. Results on the synthesis of three operational amplifiers

For all cases, a total of 4 x 10^4 sized OpAmps have been sampled by the GA, i.e., only a small fraction of the search spaces shown in Table 1. It can be seen in this table that, even though not all the objectives have been fully satisfied, the GA could arrive at acceptable values for the specifications.

In the case of the class A OpAmp, not only the transistors' sizes and biasing current have been synthesised, but also the compensating capacitance. For the case of the simple OTA, it can be seen that the GA has arrived at an amplifier with minimum power consumption, at the expense of smaller GBW and higher area statistics.

[1] Both positive and negative slew-rate have been calculated; the table shows the worst value between them

Figures 5.a, 5.b and 5.c depict the best solutions found for the class A OpAmp, simple OTA and Cascode OTA respectively, showing each transistor dimensions (width and length respectively). It can be seen, from these figures, that the differential input transistors have been constrained to have the same dimensions.

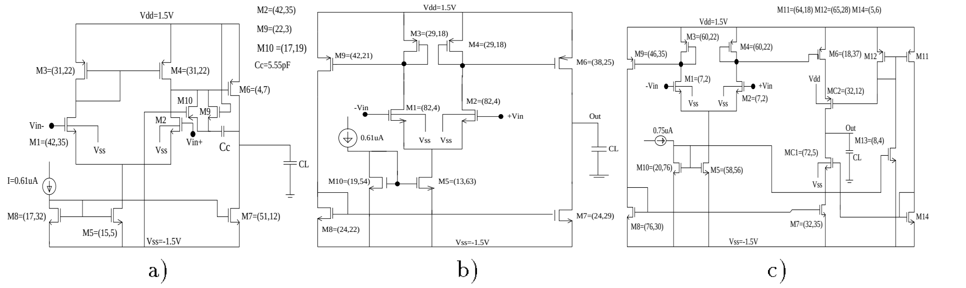

Fig. 5. Operational amplifiers synthesised by the genetic algorithm

To find the transistors' operating region, equation 1 has been applied to each transistor in the three circuits. For the Class A operational amplifier, the GA placed transistors M1 and M2 in weak inversion; the use of transistors M1, M2 and M6 in weak inversion is a common procedure in human-made design, since it improves the amplifier gain [14] for low-power applications.

For the evolved OTA amplifier in Figure 5.b, it has been verified that the GA placed the differential input transistors pair, M1 and M2, in weak inversion, and also transistors M6, M7, M8 and M9, which explains the low dissipation value. A common strategy for micropower design is to place transistors M1 and M2 in weak inversion, as:

$$GBW = \frac{(g_{M1} + g_{M2}).B}{2.C_L} \tag{13}$$

where B is given by the ratio between the currents of M7 and M5 and C_L is the amplifier load. In order to maximise GBW, the designer needs to maximise the transconductance of transistors M1 and M2. This can be achieved by placing these transistors in weak inversion, where the transconductance is proportional to the electric current [14].

In the case of the Cascode OTA amplifier of Figure 5.c, the GA placed MC1 in weak inversion and MC2 in moderate inversion. The use of these transistors in weak inversion is recommended, because the output impedance of the amplifier is proportional to the transconductance of these two transistors [14]. However, the differential input transistors are also usually placed in weak inversion, which was not observed in the synthesised cell.

In an additional experiment, we increased the Cascode OTA load to the value of 10 pF, so as to compare with a human designed cell presented in [11]. Both cells presented a 97 dB gain; our cell has been superior in terms of GBW (250 kHz against 135 kHz), slew-rate (150 mV/μs against 100 mV/μs) and biasing current (0.63 μ A against 2.5 μ A), though presenting a larger area (21,981 μm^2 against 13,852 μm^2).

Therefore, it can be verified that our automatic tool synthesised cells that followed partially micropower strategies; and the results are competitive with human made designs.

5 Conclusions

We presented a tool that performs automatic synthesis of operational amplifiers using minimal human knowledge. Particularly, we focused on low-power analog cells, which is an important industrial trend.

In order to tackle this multi-objective optimisation problem, we applied genetic algorithms, a tool based on some aspects of biological evolution, which is suitable for this particular class of optimisation problems. The authors devised a novel multi-objective evaluation methodology to cope with the problem of translating a vectorial performance measure into a scalar number.

Three OpAmp topologies have been analysed. It has been verified that the automatic tool produced results competitive with human designed cells and, in some cases, followed standard micropower strategies used in transistor biasing. Cells with power consumption below 10 μW and with acceptable values for gain, GBW and phase margin have been synthesised. The slew-rate has not been optimised, though the obtained values were similar to the ones of micropower cells.

The continuation of this work involves: tackling larger analog cells, including noise as one of the objectives and use this tool for communications applications, in which a good balance between speed and power consumption has to be achieved.

6 Acknowledges

The authors wish to thank Mr. Fabio Salazar for the collaboration and CAPES, Brazilian Federal Agency, for the financial support.

References

1. Esbensen, H., Kuh, E., "EXPLORER: An Interactive Floorplanner for Design Space Exploration", Proc. of the European Design Automation Conference, pp.356-361, 1996
2. Esbensen, H., Dreschler, R., "Evolutionary Algorithms in Computer Aided Design of Integrated Circuits", Notes, Winter School in Evolutionary Algorithms, Denmark Technical University, January, 1998.
3. Esbensen, H., "A macro-cell global router based on two genetic algorithms", in European Design Automation Conference, pp. 428-433, 1994
4. Fonseca, C. M, Fleming P. J. , "An Overview of Evolutionary Algorithms in Multiobjective Optimization", Evolutionary Computation, Vol. 3, No.1, pp.1-16, 1995.
5. Fourman, M.P., "Compaction of Symbolic Layout Using Genetic Algorithms", In J.J. Grefenstette (ed.), Genetic Algorithms and Their Applications: Proceedings of the First International Conference on Genetic Algorithms, (pp. 141-153), Hillsdale, NJ: Lawrence Erlbaum, 1985
6. Gockel, N., Dreschler, R., Becker, B., "Learning Heuristics by Evolutionary Algorithms with Variable Size Representation", in http://www.ai.mit.edu/people/unamay/icga-submissions.html, ICGA'97 Workshop on Evolutionary Computation with Variable Size Representation. July, 20, 1997.
7. Goldberg, D. E., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, 1989.
8. Johns, D. A., Martin, K., "Analog Integrated Circuit Design", John Wiley Sons, Inc., 1997.
9. Kommu, V., Pomenraz, I., "GAFAP: Genetic Algorithm for FPGA technology mapping", in European Design Automation Conference, pp. 300-305, 1993
10. Kruikamp, W., Leenaerts, D., "Darwin: CMOS opamp Synthesis by means of a Genetic Algorithm", 32nd Design Automation Conference, pp. 433-438, CA, USA, 1995.
11. Krummenacher, F., "High Voltage Gain CMOS OTA For Micropower SC Filters", Electronic Letters, 19th February, Vol. 17, No. 4, 1981.
12. Laker, K. R., Sansen, W., "Design of Analog Integrated Circuits and Systems", Mc. Graw-Hill Inc., 1994
13. Rogenmoser, R. , Kaeslin, H. , Blickle, T. , "Stochastic Methods for Transistor Size Optimization of CMOS VLSI Circuits", in Parallel Problem Solving from Nature IV, pp. 849-858, Springer-Verlag, LNCS 1141, 1996.
14. Salazar, F. A., "Projeto de Celulas CMOS Analogicas de Baixo Consumo a Partir de Transistores Operando em Inversao Fraca", Msc Thesis, Electrical Engineering Department, Pontificia Universidade Catolica, Rio de Janeiro, Brasil, 1996.
15. Sansen, W., "Design of Analog CMOS Building Blocks", Fourth Brazilian Workshop on Microelectronics, pp. 149-155.
16. Sansen, W., "Notes on short term course Low-noise Analog CMOS BiCMOS Design", promoted by the Imperial College, February 25-27 , London, 1998
17. "SMASH User and Reference Manual", Dolphin Integration, France, 1993.
18. Vittoz, E. , Fellrath , J., "CMOS Analog Integrated Circuits Based on Weak Inversion Operation ", IEEE Journal of Solid-state Circuits, vol-SC12, No. 3, June, 1977.