Synthesis of CMOS Operational Amplifiers Through Genetic Algorithms

Ricardo Salem Zebulum
Centre for Computational Neurosciences and Robotics
University of Sussex
Brighton, BN1 9SB , UK
ricardoz@cogs.susx.ac.uk

Marco A. Pacheco Marley Vellasco
ICA
Pontifícia Universidade Católica
Rio de Janeiro - Brasil
ICA@ele.puc-rio.br

Abstract

This work studies the problem of CMOS operational amplifiers (OpAmps) design optimisation. The synthesis of these 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 introduce and apply the Genetic Algorithm [4] (GA) optimisation technique to the proposed problem. A novel multi-objective optimisation methodology is embedded in our genetic algorithm and we focus on the synthesis of a standard analog operational amplifier. The proposed methodology is very general, in the sense that it can be applied to digital and analog VLSI design with multiple-objectives specifications.

1.Introduction

We present a novel methodology applied to the problem of OpAmp design optimisation, which is a problem of practical interest. Analog circuitry, though constituting only a small part of the total area of modern chips, is usually the limiting factor of their overall performance [5]. Moreover, the acquisition of low-power, high-speed and small-area analog circuits is a major tendency in the electronics industry nowadays. The use of automatic design tools together with efficient multiple-objectives optimisation algorithms is, therefore, of great importance to the development of the field.

We use the Genetic Algorithms optimisation technique[4] in this problem. Genetic Algorithms or GAs have already been employed in many CAD applications[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, the GA will perform *cell sizing*, i.e., search for transistors sizes, biasing current and compensating capacitance values in order to meet a set of specifications. It is shown that this task involves sampling a large search space and, furthermore, it is a highly multi-objective problem: a compromise among gain, dissipation, area and other factors must be achieved.

The authors apply a novel methodology to handle multi-objective optimisation problems using Genetic Algorithms. This methodology is tested in the synthesis of a standard Miller OTA cell and our results are compared with a human made design.

This work consists of four additional sections: section 2 describes the problem of CMOS operational amplifiers synthesis; section 3 describes the methodology used by the authors to cope with the problem; section 4 presents the case study and section 5 concludes the work.

2. OpAmp Design Optimisation

Whereas in bipolar based circuits the designer's creativity is used 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, to select also the transistors' operating regions[6]. 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* [5]. The transition current from the weak to strong inversion region is defined by the following equation:

$$I_D = 2.n\beta(U_T)^2 \quad (1)$$

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 sizes and its operating regions.

In the particular case of OpAmp design optimisation, a large set of specifications must be achieved by the design process[6]. Ideally, all the OpAmp specifications should be included, but, usually, only the most important ones are taken into account. Human design relies on the solution of a system of equations; the main problem with this methodology stems from the fact that there is not an exact solution to the system when many objectives are taken into account. In

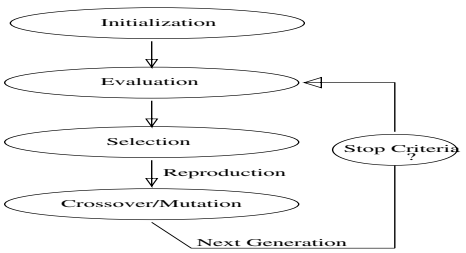

Figure 1. Basic Genetic Algorithm flow

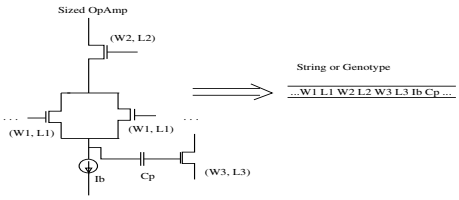

Figure 2. OpAmp representation

this case, OpAmp design is more easily viewed as a search task.

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 respectively to the principals of survival of the fittest, recombination of genetic material and random mutation observed in nature[4]. 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; 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.

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 [4].

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.

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:

$$1.5\mu A \leq I_b \leq 2.5\mu A,\ step\ 0.01\mu A \tag{2}$$

$$0.1pF \leq C_p \leq 10pF,\ step\ 0.1pF \tag{3}$$

For standard OpAmp cells, it will be shown that this encoding results in a search space of the order of 10^{30} possible solutions, which is in the range of GA applications [2].

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 most optimisation techniques rewards individual performance[3].

Two types of multi-objective optimisation approaches can be identified[3]: Plain aggregating approaches, consisting of the popular weighted-sum equation and Pareto-Based approaches, which use the Pareto concept of dominance[3].

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 plain aggregating approach in this work, since it produced better results than the Pareto methodology in this particular application. The main problem of the aggregating methodology is 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 updated according to *the average fitness value with respect to each objective and to the user 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.

$$\text{Overall Fitness} = \sum_{i=1}^{n} w_i Fn_i \tag{4}$$

$$Fn_i = \frac{F_i}{\bar{F}_i} \tag{5}$$

$$w_i = \frac{100 \ \ user_i}{\bar{F}_i}, \ \ if \ \bar{F}_i < accep_i \tag{6}$$

$$w_i = \frac{10 \ \ user_i}{\bar{F}_i}, \ \ if \ accep_i < \bar{F}_i < user_i \tag{7}$$

$$w_i = 1, \ \ if \ \bar{F}_i > user_i \tag{8}$$

if objective i is to be maximised, or:

$$w_i = -\frac{100 \ \ \bar{F}_i}{user_i}, \ if \ \bar{F}_i > accep_i \tag{9}$$

$$w_i = -\frac{10 \ \ \bar{F}_i}{user_i}, \ if \ accep_i > \bar{F}_i > user_i \tag{10}$$

$$w_i = -1, \ \ if \ \bar{F}_i < user_i \tag{11}$$

if objective i is to be minimised.

Equation 4 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 5 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, $\bar{F}_i$. 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 6 to 11. 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 $\bar{F}_i$, for a particular objective i. This ratio is multiplied by an amplification factor; this factor is set to 100 when $\bar{F}_i$ is lower than a minimal acceptable value, $Accep_i$, defined by the user; and it is set to 10 if $\bar{F}_i$ is acceptable, but still not complying with the user specification $user_i$. If the objective is to be minimised, its weight 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 ($\bar{F}_i > user_i$ for maximisation and $\bar{F}_i < user_i$ for minimisation), the weight is set to the unit.

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.

4. Case Study

The authors chose the Miller OTA amplifier cell to show the effectiveness of the proposed methodology. The Miller OTA is a two stage amplifier, whose compensation capacitance acts as a Miller capacitance, and presents a low output impedance for most of its frequency range[6]. We apply the multi-objective strategy defined previously to optimise gain, GBW, linearity, power consumption, area [1], phase margin (PM) and slew-rate (SR) [2]. Except for the slew-rate, all the performance statistics have been directly measured. The direct measure of the slew-rate along the GA execution is computationally expensive, because it requires a transient analysis. The slew-rate has then been estimated by the following equation:

$$Slew - Rate = \frac{I_{T5} - I_B}{C_L} \tag{12}$$

This is called external slew-rate, because it is related to the output capacitance C_L[6]. Only the final slew-rate values are actually measured. Linearity is taken into account by minimising the bias voltage in the output when both input voltages are grounded.

The GA manipulates the transistors' sizes, biasing current, I_b, and compensation capacitance, C_p, according to equations 2 and 3 respectively. Table 1 provides a comparison among a human made design (HM) [6] and two evolved cells (Cel1 and Cel2). The search space size tackled by the Genetic Algorithm, S.P., is also given by this table.

The vector of acceptable values, in the form (Minimum Gain, Minimum GBW, Maximum Bias Voltage, Maximum Consumption, Maximum Area, Minimum Phase Margin, Minimum Slew-Rate), has been set to (60dB, 1MHz, 0.2V bias, 400 μW, 5,000 μm^2, 55^o, 1V/μs) and the vector of desired values has been set to (70dB, 2MHz, 0.1V bias, 200 μW, 2,000 μm^2, 70^o, 3V/μs). It can be seen that both Cel1 and Cel2 comply with the acceptable values and, in many respects, comply with the desired values, which have been selected at hard specifications. From Table 1, it can be seen

[1] The Miller capacitance has not been taken into account in the area calculation

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

Features	H.M.	Cel1	Cel2
S.P.	—	10^{24}	10^{24}
Gain	71.2dB	72.5dB	68.6dB
GBW	2MHz	1.9MHz	1 MHz
Bias	-0.65mV	-1.64mV	-0.56mV
Offset	-54.3mV	-0.14V	-42.1mV
SR	2.2 V/μs	2.3 V/μs	0.8 V/μs
I_b	2.5μA	1.9μA	1.9μA
C_p	1 pF	2.05 pF	2.8 pF
V_{cc}	2.5V	2.5V	2.5V
Dissip.	527.8μW	330μW	172.1μW
Area	1,929.4μm^2	2,300 μm^2	5,380 μm^2
PM	65^o	52^o	60^o
Tech.	3μ n-well	3 μ n-well	3μ n-well
R_L	100k	100k	100k
C_L	10pF	10pF	10pF

Table 1. Comparison among hand-made and evolved Miller OTA OpAmps

that Cel1 arrives at values akin to the ones observed in the hand-made design, presenting, though, a lower power consumption value. Cel 2 presents a power consumption much smaller than the other cells at the expense of a larger area and lower slew-rate. It is up to the user to choose which evolved cell suits for the particular application.

Figure 3 displays two of the three cells shown in the previous table. From this figure, it can be seen that transistor pairs (T1, T2), (T3, T4) and (T7, T8) have been constrained to have equal sizes, as in hand made design. It has been verified that the evolved designs followed the hand made one in the following respects:

1. High value of $(W/L)_6$, since transistor T6 provides gain [6];
2. High value of $(W/L)_5$, which improves the output swing, hence, the slew rate;
3. $(W/L)_1 > 1$, since transistor T1 also provides gain.

The GA arrived at these strategies without any kind of previous knowledge being supplied to the system.

The evolution of the Miller capacitance has been important to achieve stability requirements. This capacitance determines the amplifier dominant pole, BW. A guideline for stability requirements is to keep the non-dominant pole around 3 x GBW.

We used the level 2 transistor model in the SMASH simulator [7] in this case study, with parasitic capacitances being taken into account (the complete set of used transistors parameters may be found in [6]). The proposed methodology has also been applied to low-power OpAmp design and results can be found in [8].

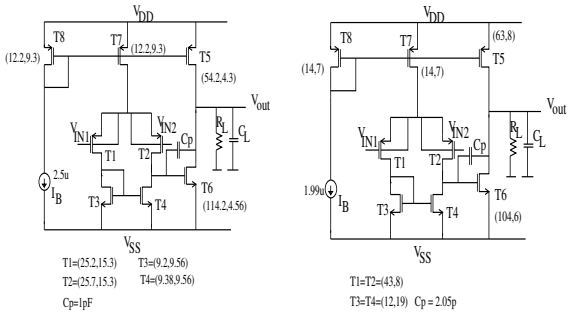

Figure 3. Miller OTA : (A) - Hand Made (B) - Cel1

5. Conclusions

We presented a tool that performs automatic synthesis of operational amplifiers using minimal human knowledge. It has been verified that the automatic tool produced results competitive with human designed cells, following standard design strategies.

Our methodology has the potential to be applied in a broad scope of VLSI design problems, ranging from low-power analog design [8] to digital design, for which less objectives, comparing to analog design, have to be taken into account.

6. Acknowledges

The authors wish to thank Mr. Fabio Salazar for the collaboration and CAPES 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, School in Evolutionary Algorithms, Denmark, Jan., 1998.

[3] - Fonseca, C. M, Fleming P. J. , "An Overview of Evolutionary Algorithms in Multiobjective Optimization", Evolutionary Computation, Vol. 3, No.1, pp.1-16, 1995.

[4] -Goldberg, D. E., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, 1989.

[5] - Johns, D. A., Martin, K., "Analog Integrated Circuit Design", John Wiley Sons, Inc., 1997.

[6] - Laker, K. R., Sansen, W., "Design of Analog Integrated Circuits and Systems", Mc. Graw-Hill Inc., 1994

[7] - "SMASH User and Reference Manual", Dolphin Integration, France, 1993.

[8] - Zebulum, R.S., Pacheco, M.A., Vellasco, M., "A Multi-Objective Optimisation Methodology Applied to the Synthesis of Low-Power Operational Amplifiers", to appear in the Proc. of the Int. Conf. on Microelectronics and Packaging, Brazil, Aug, 1998.