Automatic Synthesis of Analog Integrated Circuits Using Genetic Algorithms and Electrical Simulations

Lucas Compassi Severo, Alessandro Girardi
lucascs.eletrica@gmail.com, alessandro.girardi@unipampa.edu.br

Federal University of Pampa – UNIPAMPA
Campus Alegrete

The goal of this paper is to present a tool for automatic synthesis of analog basic integrated blocks using the genetic algorithm heuristic and external electrical simulator. The methodology is based on the minimization of a cost function and a set of constraints in order to size circuit individual transistors. The synthesis methodology was implemented in Matlab and used GAOT (Genetic Algorithm Optimization Toolbox) as heuristic. For transistors simulation we used the Smash simulator and ACM MOSFET model, which includes a reduced set of parameters and is continuous in all regions of operation, providing an efficient search in the design space. As circuit design example, this paper shows the application of this methodology for the design of an active load differential amplifier in AMS 0.35um technology.

1. Introduction

The design automation of analog integrated circuits can be very useful in microelectronics, because it provides an efficient search for the circuit variables, among a set of design constraints, to make it more efficient as possible. Several works have been done in this theme, aiming the development of tools for the automation of time-consuming tasks and complex searches in highly non-linear design spaces [1, 2]. However, as far as we know, there is not a commercial tool capable to perform the synthesis of analog circuits with optimum results in a feasible time. An important improvement in the analog design could be the automation of some design stages, such as transistor sizing and layout generation [3], maintaining the interaction with the human designer. The large number of design variables and the consequent large design space turn this task extremely difficult to perform even for most advanced computational systems. Therefore, it is mandatory the use of artificial intelligence with great computational power to solve these problems.

In this context, we propose an automatic synthesis procedure for basic analog building blocks which can size transistors width (W) and length (L) with efficient time and ordinary computational resources. The synthesis procedure has as main strategy the global search using genetic algorithm and the evaluation of circuit characteristics through the interaction with an electric simulator. The Genetic Algorithm (GA) is a technical idealized in 1975 by scientist John Holland inspired by principles of natural evolution proposed by Charles Darwin. This evolutionary heuristic is very used for automatic design of integrated circuits [4, 5, 6].

This work is organized as follows: section 2 shows the description of the proposed methodology; section 3 presents the application of the methodology in the design of a specific analog block - the differential amplifier - with circuit description and final results; finally, section 4 shows the conclusion.

2. Automatic Synthesis

The methodology is based on the reduction of a cost function for a specific analog block. This cost function is dependent of the circuit electrical characteristic and is implemented as an equation in terms of circuit variables. The electrical characteristics can be power consumption, area, voltage gain, etc, or a combination of these. In this work we used the circuit power dissipation as cost function. The optimization is based on the reduction of a cost function and a set of constraints. This reduction is performed by genetic algorithm with values provided by electrical simulations through an external electrical simulator.

The genetic algorithm is a heuristic for non-linear optimization based on the analogy with biologic evolution theories [7]. It is a non-deterministic algorithm and it works with a variety of solutions (population), simultaneously. The population is a set of possible solutions for the problem. The size of the population is defined in order to maintain an acceptable diversity considering an efficient optimization time. Each possible solution of population is denominate chromosome, where chromosome is a chain of characters (gens) that represent the circuit variables. This representation can be in binary number, float or others. Fig.1 shows an example of binary chromosome. The quality of the solution is defined by an evaluation function (cost function). Fig.2 shows the automated design flow using genetic algorithm. The algorithm receives an initial population, created randomly, some recombination and mutation operators and the MOSFET technology model parameters. The population is evaluated using an external commercial electrical simulator. Based on valuation and roulette method [8] the parent chromosomes are selected for generating new chromosomes. The new chromosomes are
created including recombination and mutation - analogy with biology. In the recombination, the chromosomes parents are parties and the union of parts of parents make the recombination. The mutation is a random error that happens in a chromosome. The probability of mutation is defined and is compared with a random value. If this random value is less than the probability then a gene on chromosome is randomly changed. The next step is the exclusion of parents, evaluation of new chromosomes, using again the electrical simulator and a cost function. Based on these values, new chromosomes are introduced in the population. At the end of each iteration, the stopping condition is tested and, if true, then the optimization is finished. Otherwise, new parents are selected and the process is repeated. The stopping condition can be the number of generations (iterations), minimal variation between variables or cost function, or others.

The synthesis tool developed in this work was implemented in Matlab® and used the Dolphin Smash® simulator as external electrical simulator. The MOSFET model used was ACM [9], guaranteeing the correct exploration of all transistor operation regions. For the execution of genetic algorithm, we adopted GAOT (Genetic Algorithm Optimization Toolbox), an implementation of GA for Matlab developed by Cristopher R. Houck et al [10].

3. Design Example – Differential Amplifier

As an application for the proposed methodology, we implemented a tool for the automatic synthesis of a CMOS differential amplifier. The differential amplifier is one of the most versatile circuits in analog design. It is compatible with ordinary CMOS integrated-circuit technology and serves as input stage for op amps [11]. Its basic function is to amplify the difference between the input voltages. The circuit for a differential amplifier with active load is basically composed by a load current mirror (M3 and M4), a source-coupled differential pair (M1 and M2) and a reference current mirror (M5 and M6), shown in fig.3. The main electrical parameters of the circuit are low-frequency voltage gain (Av0), gain-bandwidth product (GBW), slew-rate (SR), input common-mode range (ICMR), dissipated power (Pdiss) and area (A), among others.

The low-frequency gain is the relationship between output and input voltages, defined as:

\[ Av_0 = \frac{g_{mi}}{g_{ds2} + g_{ds4}} \quad (1) \]

where \( g_{mi} \) is the gate transconductance of transistor M1 and \( g_{ds2} \) and \( g_{ds4} \) are the output conductance of M2 and M4, respectively.
The slew rate (SR) is the maximum output-voltage rate, either positive or negative, given by:

\[ SR = \frac{I_{\text{ref}}}{C_1} \quad (2) \]

Here, \( I_{\text{ref}} \) is the current source of circuit \( C_1 \) is the total output capacitance. This capacitance is estimated as the sum of the load capacitance \( CL \) and drain capacitance of \( M2 \) and \( M4 \). Input common-mode range (ICMR) is the maximum and minimum input common-mode voltage, defined as:

\[ ICMR^- = v_{DS5\text{(sat)}} + V_{GS1} + V_{SS} \quad (3) \]
\[ ICMR^+ = v_{DD} - V_{GS3} + V_{TN1} \quad (4) \]

In this case, \( v_{DS5\text{(sat)}} \) is the saturation voltage of transistor \( M5 \), \( V_{GS1} \) and \( V_{GS3} \) are gate-source voltages of \( M1 \) and \( M3 \), respectively, \( v_{DD} \) and \( V_{SS} \) are voltage sources of circuit and \( V_{TN1} \) is the threshold voltage of \( M1 \). The \( v_{DD} \) and \( V_{SS} \) source voltages are defined in this example as -1.65V and 1.65V, respectively.

The gain-bandwidth product is given by:

\[ GBW = \frac{g_{m1}}{C_1} \quad (5) \]

The cost function for the circuit in this case is related to the power dissipation, defined as:

\[ f = \frac{(v_{DD} + |V_{SS}|)I_{\text{ref}}}{P_0} + R \quad (6) \]

where, \( R \) is a penalty constraint function, which will be a large value if the constraints are not met, and zero if all constraints are met. \( P_0 \) is the reference power dissipation for normalization purpose.

![Fig.3 - Schematics of a differential amplifier](image)

The optimization procedure was implemented in Matlab, using the GAOT Toolbox as described before. We analyzed three scenarios with different population (10, 100 and 1000 individuals), in order to verify which one is more suitable for this type of problem. Tab.1 shows the values of constraints, optimization time and the achieved final cost function (optimized power dissipation). Analyzing this table it is possible to notice that the best value of individual numbers, between the analyzed, is 1000 individuals, because this population presented a major minimization in the power dissipation. Tab.2 shows the final values of variables and tab.3 shows a comparison between achieved and imposed restrictions. Fig.4 shows the variation of cost function and the slew rate (SR) along the iterations.

The proposed methodology was initialized with a random generated population. After more than 2000 iterations, final population satisfies all constraints and provided optimized power dissipation. This is a valuable characteristic of design automation using genetic algorithms, because a good guess for initial values is not mandatory.
Fig. 4 – Evolution of algorithm response along with iterations: a) cost function; b) slew rate variation.

Tab. 1 – Optimization results for different population sizes

<table>
<thead>
<tr>
<th>Individuals</th>
<th>GBW</th>
<th>Slew Rate</th>
<th>Voltage</th>
<th>ICMR-</th>
<th>ICMR+</th>
<th>Power dissipation</th>
<th>Time</th>
<th>Generations</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3.91MHz</td>
<td>5.12V/µs</td>
<td>61.18 dB</td>
<td>-1.02V</td>
<td>1.07V</td>
<td>148.33µW</td>
<td>22min</td>
<td>2524</td>
</tr>
<tr>
<td>100</td>
<td>949kHz</td>
<td>5.00 V/µs</td>
<td>62.29dB</td>
<td>-0.94V</td>
<td>0.81V</td>
<td>148.80µW</td>
<td>19min</td>
<td>2354</td>
</tr>
<tr>
<td>1000</td>
<td>5.95MHz</td>
<td>5.00 V/µs</td>
<td>60dB</td>
<td>-0.70V</td>
<td>1.03V</td>
<td>139.46µW</td>
<td>25min</td>
<td>2026</td>
</tr>
</tbody>
</table>

Tab. 2 – Circuit variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Initial value</th>
<th>Optimized value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W(M1 \land M2) )</td>
<td>random</td>
<td>99.98 µm</td>
</tr>
<tr>
<td>( W(M3 \land M4) )</td>
<td>random</td>
<td>24.17 µm</td>
</tr>
<tr>
<td>( W(M5 \land M6) )</td>
<td>random</td>
<td>67.49 µm</td>
</tr>
<tr>
<td>( L(M1 \land M2) )</td>
<td>random</td>
<td>1.85 µm</td>
</tr>
<tr>
<td>( L(M3 \land M4) )</td>
<td>random</td>
<td>2.56 µm</td>
</tr>
<tr>
<td>( L(M5 \land M6) )</td>
<td>random</td>
<td>23.96 µm</td>
</tr>
<tr>
<td>( I_{ref} )</td>
<td>random</td>
<td>51.22 µA</td>
</tr>
</tbody>
</table>

Tab. 3 – Circuit constraints

<table>
<thead>
<tr>
<th>Restriction</th>
<th>Required</th>
<th>Reached</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain (Av0)</td>
<td>60.00dB</td>
<td>60.00dB</td>
</tr>
<tr>
<td>GBW</td>
<td>1.00MHz</td>
<td>5.95MHz</td>
</tr>
<tr>
<td>SR</td>
<td>5.00V/µs</td>
<td>5.00V/µs</td>
</tr>
<tr>
<td>( I_{CMR^-} )</td>
<td>-0.70V</td>
<td>-0.70V</td>
</tr>
<tr>
<td>( I_{CMR^+} )</td>
<td>0.70V</td>
<td>1.03V</td>
</tr>
<tr>
<td>Gate area</td>
<td>-</td>
<td>3727.79µm²</td>
</tr>
<tr>
<td>Power diss.</td>
<td>minimize</td>
<td>139.46 µW</td>
</tr>
</tbody>
</table>

4. Conclusion

The proposed methodology for the synthesis of basic analog building blocks presented good results in a reasonable computing time. Genetic algorithms are very suitable for analog design automation because the convergence of the final solution is not directly dependent of initial solution, and it is not necessary a great knowledge by the human designer about the circuit characteristics. However, it is very important to determine the size of population (number of individuals) because it is directly related to the quality and the time of optimization. In our work, a population of 1000 individuals provides the best result.

Electrical simulator using the ACM model implemented in this methodology guarantee the search in all regions of operation of MOSFET transistors.

As future work, we intend to compare the solution obtained with GAs with other optimization heuristics. Also, we can explore the use of electrical simulators from different vendors, expand the methodology for other analog basics blocks and create an interface for the human designer.

5. Acknowledgments

The grant provided by FAPERGS research agency for supporting this work is gratefully acknowledged.

6. References


