



HAL
open science

Automatic Flat-Level Circuit Generation with Genetic Algorithms

Miguel Campilho-Gomes, Rui Tavares, João Goes

► **To cite this version:**

Miguel Campilho-Gomes, Rui Tavares, João Goes. Automatic Flat-Level Circuit Generation with Genetic Algorithms. 11th Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS), Jul 2020, Costa de Caparica, Portugal. pp.101-108, 10.1007/978-3-030-45124-0_9. hal-03741565

HAL Id: hal-03741565

<https://inria.hal.science/hal-03741565v1>

Submitted on 1 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Automatic Flat-Level Circuit Generation with Genetic Algorithms

Miguel Campilho-Gomes^{1,2}, Rui Tavares^{3,4} and João Goes^{3,4}

¹ NOVA-FCT, Nova School of Science and Technology of Universidade Nova de Lisboa,

² Instituto Superior de Engenharia de Lisboa,

mpeg@isel.ipl.pt

³ CTS/UNINOVA, Centre of Technology and Systems of UNINOVA

⁴ DEEC/NOVA-FCT, Computer and Electrical Engineering Department of Nova School of Science and Technology of Universidade NOVA de Lisboa,

{rmt,jg}@fct.unl.pt

Abstract. This paper describes a novel methodology to generate analog and digital circuits, autonomously, using the transistor (or other elementary device, e.g resistor) as the basic elementary block – flat-level. A genetic algorithm is employed as the generation engine and variable length chromosomes are used to describe the circuit topology that evolves during the search. The circuit devices type and sizing are described by each gene of genetic algorithm. The automatic process starts with the circuit input and output specifications, and proceeds with the circuit topology and sizing evolution to meet those specifications, eventually, ending up with a novel topology. During the evolution, each generated circuit is electrically evaluated by a spice-like circuit simulator, i.e. Ngspice, using full model specifications - like BSIM3 for transistors - in a highly parallelized architecture built over a multi-thread model.

Keywords: Automatic Topology Generation, Genetic Algorithm, Variable Length Chromosome, Digital Circuit, Analog Circuit, Ngspice, Amplifier.

1. Introduction

Digital and analog circuit design for Integrated Circuits (ICs) classically has three distinctive stages: topology choice, component sizing and layout generation. For some years now, digital synthesis is almost fully automated [**Erro! A origem da referência não foi encontrada.**] but analog ICs, or the analog part of modern Mixed-Signal System-on-a-Chip (MS-SoC), still take a lot of human intervention in the topology choice stage in spite of the research effort of the past few decades in the automated synthesis field.

The Electronic Design Automation (EDA) field, a trend in research effort for the past three decades, targeted in technology roadmaps like International Technology Roadmap for Semiconductors (ITRS)[**Erro! A origem da referência não foi encontrada.**], focuses in reduction of designer effort, minimization of time-to-market and minimization of production costs. Included in EDA, the subfield of Analog

Design Automation (ADA) deals specifically with analog circuits, and it also contains the three stages mentioned above: topology synthesis, component sizing and layout generation, though when the circuit is not to be integrated the layout stage is replaced by a by a Printed Circuit Board (PCB) generation stage.

The topology synthesis stage is the least developed stage. Automatic topology synthesis of analog circuits is far from being a straightforward task and, despite having received some research effort in recent decades, it has not yet matured enough to become available in software packages for the integrated circuit industry. Analog circuit design is still essentially based on expert designers and has often been considered an art [Erro! A origem da referência não foi encontrada.]. This design work is generally geared toward the reuse of well-known topologies in new circuit functions or to improve existing circuit topologies and, less often, to create new circuit topologies. But these procedures not always achieve the best performance within the shortening design-to-market cycle time. Furthermore, as the technology is heading to deep sub-micrometer sizes, the present circuit topologies and design methods are often not good enough to completely fulfill the final circuit high-end specifications. It is imperative to develop a systematic and automatic method to generate novel circuit topologies that combine high efficiency with full specification compliance, accommodate submicron technology constraints, reduce the trial and error cycle of the circuit design lifecycle.

1.1 Research Question, Hypothesis and Motivations

Generation of circuit topologies can be seen as a search and optimization problem and, therefore, it can be performed with Genetic Algorithms (GAs). What is unclear is to what extent this can be accomplished by a GA, particularly when handled at flat circuit-level, and in concrete for the case of CMOS amplifiers. If some human designed topologies can be generated by GAs, perhaps GAs may also generate new topologies.

As a natural result of the development of amplifier design conducted to this day, and also boosted by the current submicron technologies used in today's Integrated Circuits (ICs), CMOS integrated amplifiers have high efficiencies and very optimized specifications compared to the not-so-distant past. The main research question adopted in this work is therefore:

- Is it possible to generate new circuit topologies for CMOS amplifiers, at flat circuit-level, in an automatic way (software generated), with optimized specifications and high efficiency at submicron technologies?

And the proposed hypothesis to address this problem and to guide the research effort is:

- It is possible to develop a methodology and a software tool based in genetic algorithms to, automatically, generate novel at circuit-level topologies for CMOS amplifiers with high efficiency at submicron technologies.

The research question and the proposed hypothesis enclose some other questions that will be addressed in the course of this work, such as:

- To what extent does the canonical genetic algorithm is suited to circuit synthesis?

Since Holland's presentation of canonical (or conventional) Genetic Algorithm (GA-c) [Erro! A origem da referência não foi encontrada.] there have been many developed variants for the algorithm, and some may be better suitable for circuit synthesis than others. It is unclear what are the actual limitations of GA-c when applied to analog circuit synthesis, particularly to amplifier synthesis, so it is relevant to try to get some sense of what can actually be accomplished by the GA-c before any of its variants are considered.

- What circuit codification is best suited for circuit synthesis, namely for the synthesis of analog topologies?
It is well known that part of the difficulty of obtaining good results with GAs is to choose a good codification scheme of the problem. It should be evaluated whether the classical GA-c bit strings are suitable for circuit synthesis, such as analog synthesis, or whether it is necessary to experiment with other schemes that may prove more suitable.

1.2 Contribution to Life Improvement

Life improvement is expected to be the main outcome of technology advances across different areas for human well-being. In the area of circuit design and electronic systems, the advances are certainly in reducing energy consumption, improving performance and reducing the area used in integrated circuits. These technological developments have impacts on health, environment, transportation and communication systems worldwide.

More energy-efficient electronic circuits and systems contribute to extend the duration of the battery cycles. Extending battery life is expected to reduce the negative environmental impact of the batteries.

Integrated Pico and Nano-scale electronic circuits and systems are the primary vehicle for building data collectors, either for real-time monitoring or for building bigdata containers. Two possible examples are the freight sector and the healthcare domain. In the freight sector, during packaging, transport and storage, the goods are constantly tracked, and their conditions are supervised throughout the process. In the healthcare domain, technological evolution closely related to life improvement is being driven by monitoring the patient all the time and collecting data closer to the patient. To minimize the hassle to the patient, the circuits' size must be reduced, and efficiency maintained. The bigdata collected from the patient is real-time processed with local hardware processors.

Energy and performance improvements are prime indicators to consider in transducers in-between real-world analog domain and the digital domain of the cyber physical systems. These electronic circuits are the first system block in the analog front end and, again, must be energy efficient, with low area consumption.

To cope with the momentum of the IC industry to embrace all the real-world needs to increase knowledge, wellbeing, quality of life, and facilitate the collaboration among different entities, one must come up with more efficient hardware: electronic circuits and systems.

This paper pretends to contribute with a novel methodology to, automatically, generate electronic circuits, efficiently outcome the state-of-the-art circuit performance specifications and simultaneously deal with downsizing of the technology complexity and challenges.

2. Genetic Algorithms and Circuit Synthesis

Genetic Algorithms (GAs) are search and optimization algorithms based on the mechanics of natural selection and natural genetics. They were developed by J. Holland and colleagues [Erro! A origem da referência não foi encontrada.] in the early seventies and today have broad application in the areas of business, engineering and science. GAs belong to the larger group of Evolutionary Algorithms (EAs), which constitute a class of stochastic search and optimization methods that simulate the process of natural evolution.

Problem representation, which translates to individual encoding in GAs, is a major component of GAs application to a concrete problem, and therefore encoding a circuit in a GA is a topic frequently discussed and object of significant research effort [Erro! A origem da referência não foi encontrada., Erro! A origem da referência não foi encontrada.]. In this work there was a first attempt to use a codification technique for circuits that applies the classic bit strings used by GA-c to represent a circuit. In this representation there is a fixed number of components available to the GA and the evolution takes place by changing the nodes to which each component terminal is connected to. The circuit of Fig. 1 was generated automatically by the GA using this codification technique.

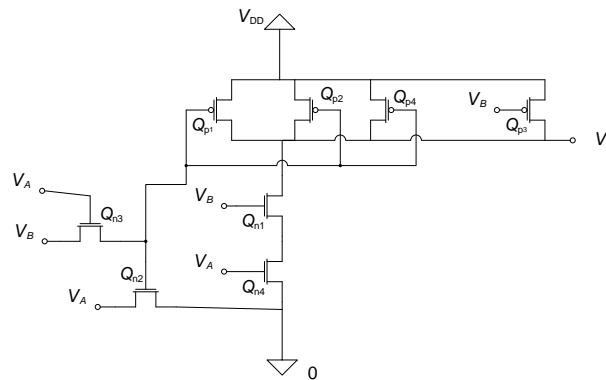


Fig. 1. Circuit generated for a NAND gate.

Although it was possible to accomplish the generation of simple circuits, this codification scheme suffers from lack of scalability to higher complexity topologies. Therefore, it was necessary to try another codification technique, and Variable Length Chromosomes (VLCs) were tested. This concept is a model that has been around for quite some time [Erro! A origem da referência não foi encontrada., Erro! A

origem da referência não foi encontrada.], and has been applied in several scientific fields [**Erro! A origem da referência não foi encontrada.**, **Erro! A origem da referência não foi encontrada.**].

In this work we use VLCs and an encoding scheme in which a chromosome is a circuit descriptor and each gene is a component descriptor. Each chromosome has a variable number of genes, and this number may either increase or decrease during execution of the GA. In this encoding scheme each gene is comprised of three fields, as depicted in Fig 2. The first field identifies the component type and is typically implemented as an integer variable. The second field is a set of parameters needed to describe the characteristics of the component, which are typically implemented as real variables. And the third field is a set of nodes indicating where the component terminals are connected (implemented as positive integer variables).

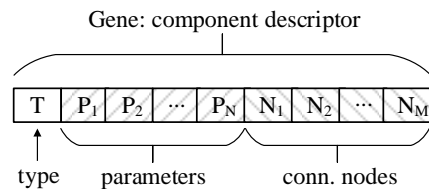


Fig. 2. Circuit component encoding using VLCs.

VLCs poses some questions to the crossover and mutation operators, so these operators are often adapted to work in populations with chromosomes that don't have a constant length. In this work we use a gene boundary crossover operator that implements equal crossover and inside crossover operations [**Erro! A origem da referência não foi encontrada.**] and a mutation operator that can alter any allele of a gene, which means, for instance, that it can change the transistor channel type, its width or length, and the node connected to any of its terminals. This operator also implements *insertion*, *replication* and *deletion* operations [**Erro! A origem da referência não foi encontrada.**]. Another feature of this mutation operator is the ability of refusing a mutation if it worsens the chromosome fitness. The probability p_r of refusing a mutated chromosome is not constant, instead it is adapted during evolution of the population. The probability p_a of acceptance ($p_a = 1 - p_r$) starts high and decreases over time, similarly to a simulated annealing process [**Erro! A origem da referência não foi encontrada.**], allowing broader regions of space to be searched in early generations but later narrowing this search to small regions around the solution(s) already obtained (local exploitation). Instead of decreasing the probability of acceptance over time, in our work we use a heuristic that adapts it according to a function f_Q which measures the quality of the solutions obtained so far (and converts it to a probability). This measure can be obtained in two ways: 1) by the average of all chromosome's fitness smoothed by 2nd order discrete low pass filter; 2) by the fitness of the best chromosome in the population. Both ways were tested and both performed similarly well in the synthesized circuits, making this approach a robustness enhancer of the adaptive process by decreasing its sensitivity to the convergence speed of the

GA (which is known to be highly irregular and dependent of numerous factors, such as initial solution and general GA parametrization).

Using VLCs with the proposed codification scheme, and using a SPICE-based circuit simulator like *Ngspice* [Erro! A origem da referência não foi encontrada.] for fitness evaluation, the GA was able to generate circuits of higher complexity, like the half-adder depicted in Fig. 3 and the 20dB DC amplifier depicted in Fig. 4. A common characteristic of the circuits shown in Fig. 1, Fig. 3 and in Fig. 4 is that there are numerous superfluous (either because they are redundant or useless) components. To minimize the number of these components we propose a technique based in dynamically adapting the weight of a penalty constraint function. This weight must be quite low in early generations, or the GA will hardly converge at all, but it should be high enough in late stages of the GA, particularly after the main objectives of the circuit are fulfilled, so that the GA “cleans” the circuit by removing unnecessary components (but without losing “sight” of those main objectives).

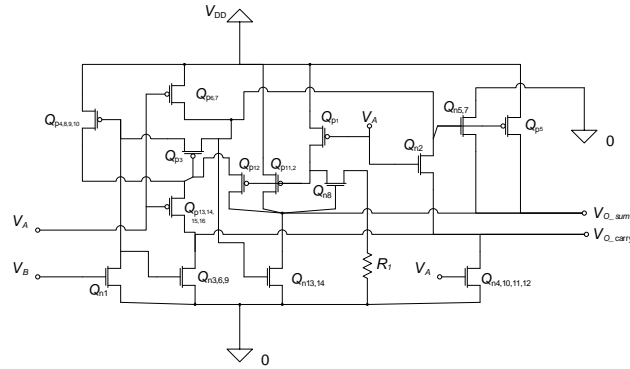


Fig. 3. An half-adder automatically generated by a GA using VLCs.

To adapt that weight we reuse function f_Q described earlier which measures the quality of the solutions obtained so far, and use a heuristic that converts it to the penalty weight.

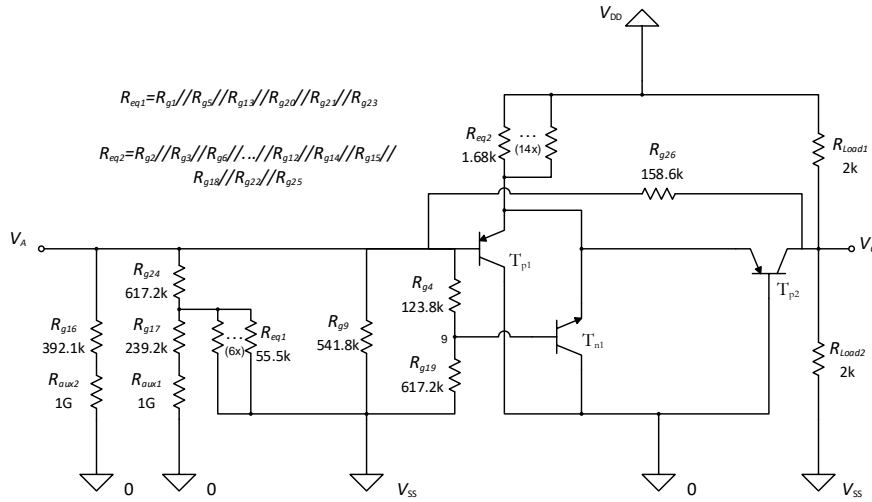


Fig. 4. A 20dB DC amplifier automatically generated by a GA using VLCs.

This technique has proven to be effective and the circuits depicted in Fig. 5 illustrate its outcome when applied to the circuits previously introduced in Fig. 1 and in Fig. 4.

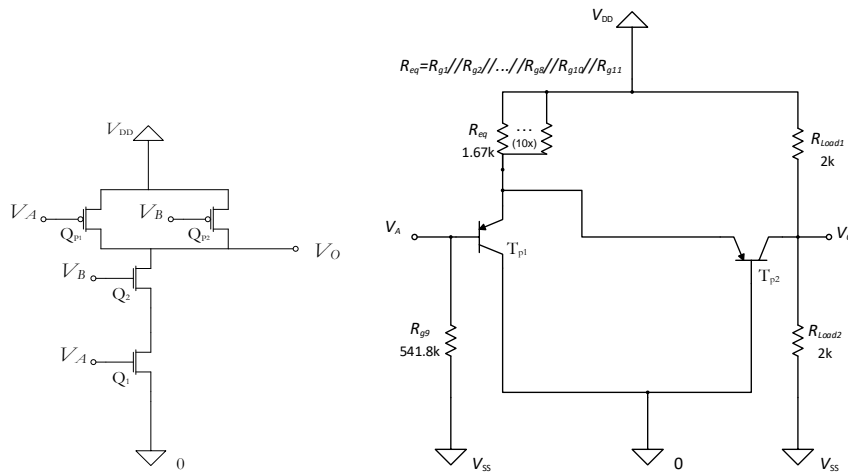


Fig. 5. Circuits generated by an GA using unnecessary component removal heuristics.

7 Conclusions

The main research topic of this work is to better understand how GAs can be used in circuit synthesis. As part of this study, we intended to understand the main constraints of GA-c for this task and to investigate possible improvements that make these algorithms more suitable for circuit synthesis.

With GA-c, and using a simple codification scheme for circuit representation based in GA-c bit strings, the algorithm was able to generate several simple digital and analog circuits, of which an example of a logic NAND gate was introduced. But scalability for more complex circuits has proved to be an impractical task, at least with the available computing power. The research effort was then focused on using VLCs and using a new circuit coding technique. This shift in the research effort implied several changes and improvements to the original algorithm, namely in the crossing and mutation operations. Another improvement is a "circuit cleaning" technique that eliminates redundant components in the circuit. In this technique we dynamically adapt some parameters of the GA in order to steer the algorithm towards the sub-objective of minimizing the number of components. One more improvement presented is a heuristic that implements an adaptive probability of acceptance of mutated chromosomes according to a measure of the quality of the solutions obtained. The combination of these changes and enhancements has enabled the generation of more complex digital and analog circuits without compromising the scalability of the entire synthesis procedure accomplished by the GA.

Some experimental results were presented to illustrate the results obtained with the methodology and techniques described here. Other experimental results already obtained sustains the scalability of these procedures. Current and future work aims to understand the scope and limitations of these techniques in the synthesis of circuits, namely in the synthesis of analog amplifiers.

Acknowledgments. This work was partially supported by the Fundação para a Ciência e Tecnologia, UIDB/00066/2020 (CTS – Center of Technology and Systems).

References

1. C. Faragó, A. Lodin, and R. Groza, "An operational transconductance amplifier sizing methodology with genetic algorithm-based optimization." *Acta Technica Napocensis. Electronica-Telecomunicatii*, vol. 55, no. 1, 2014.
2. ITRS, "International Technology Roadmap for Semiconductors," ITRS Web-Site: <http://www.itrs2.net>.
3. C. Ferent and A. Dobioli, "Measuring the uniqueness and variety of analog circuit design features," *INTEGRATION, the VLSI journal*, vol. 44, no. 1, pp. 39–50, 2011.
4. J. H. Holland, *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press, 1975, second Edition, 1992.

5. J. D. Lohn, S. P. Colombano, G. L. Haith, D. Stassinopoulos, and P. Norvig, "A parallel genetic algorithm for automated electronic circuit design," in Proc. Computational Aerosciences Workshop, 2000.
6. Y. A. Sapargaliyev and T. G. Kalganova, "Challenging the evolutionary strategy for synthesis of analogue computational circuits," *Journal of Software Engineering and Applications*, vol. 3, no. 11, pp. 1032–1039, 2010.
7. R. S. Zebulum, M. A. Pacheco, and M. Vellasco, "Comparison of different evolutionary methodologies applied to electronic filter design," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on. IEEE, 1998*, pp. 434–439.
8. J. D. Lohn and S. P. Colombano, "Automated analog circuit synthesis using a linear representation," in *International Conference on Evolvable Systems. Springer, 1998*, pp. 125–133.
9. S. N. Pawar and R. S. Bichkar, "Genetic algorithm with variable length chromosomes for network intrusion detection," *International Journal of Automation and Computing*, vol. 12, no. 3, pp. 337–342, 2015.
10. J. Ni, K. Wang, H. Huang, L. Wu, and C. Luo, "Robot path planning based on an improved genetic algorithm with variable length chromosome," in *Proc. Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) 2016 12th Int. Conf. Natural Computation, Aug. 2016*, pp. 145–149.
11. D. S. Deif and Y. Gadallah, "Wireless sensor network deployment using a variable-length genetic algorithm," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC), Apr. 2014*, pp. 2450–2455.
12. B. Hutt and K. Warwick, "Synapsing Variable-Length Crossover: Meaningful Crossover for Variable-Length Genomes," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 118–131, Feb. 2007.
13. M. Barros, J. Guilherme, and N. Horta, "Analog circuits optimization based on evolutionary computation techniques," *INTEGRATION, the VLSI journal*, vol. 43, no. 1, pp. 136–155, 2010.
14. SourceForge, "Mixed Mode-Mixed Level Circuit Simulator based on Berkeley's SPICE 3F5," <http://ngspice.sourceforge.net/>.