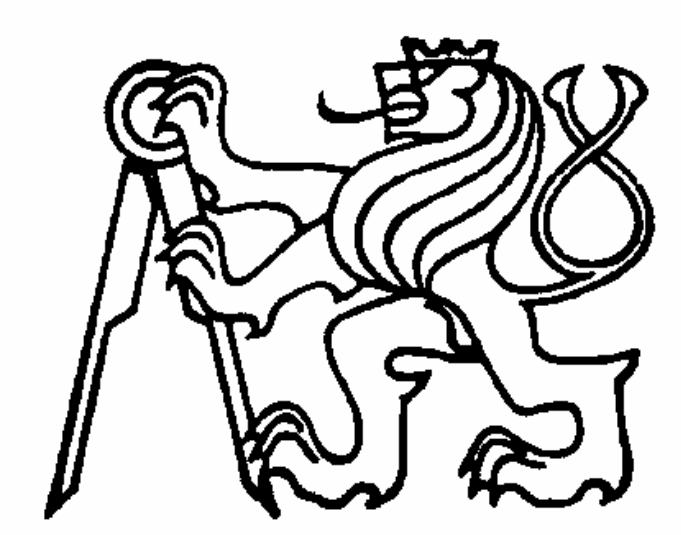
CZECH TECHNICAL UNIVERSITY IN PRAGUE



DOCTORAL THESIS STATEMENT

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Microelectronics

Ing. Miloslav Kubař

NOVEL OPTIMIZATION TOOL FOR ANALOG INTEGRATED CIRCUITS DESIGN

Ph.D. Programme: Electrical Engineering and Information Technology
Branch of study: Electronics

Doctoral thesis statement for obtaining the academic title of "Doctor", abbreviated to "Ph.D."

Prague, June 2013

The doctoral thesis was produced in combine manner

Ph.D. study at the department of Microelectronics of the Faculty of Electrical Engineering of the CTU in Prague

Candidate: Ing. Miloslav Kubař
Department of Microelectronics
Faculty of Electrical Engineering of the CTU in Prague
Technická 2, 166 27 Prague 6.

Supervisor: Ing. Jiří Jakovenko, Ph.D.
Department of Microelectronics
Faculty of Electrical Engineering of the CTU in Prague
Technická 2, 166 27 Prague 6.

Opponents: ……

……

……

The doctoral thesis statement was distributed on: ……

The defence of the doctoral thesis will be held on …… at …… before the Board for the Defence of the Doctoral Thesis in the branch of study Electronics in the meeting room No. …… of the Faculty of Electrical Engineering of the CTU in Prague.

Those interested may get acquainted with the doctoral thesis concerned at the Dean Office of the Faculty of Electrical Engineering of the CTU in Prague, at the Department for Science and Research, Technická 2, Praha 6.

Chairman of the Board for the Defence of the Doctoral Thesis
in the branch of study Electronics
Faculty of Electrical Engineering of the CTU in Prague,
Technická 2, 166 27 Prague 6.

TABLE OF CONTENTS

1. CURRENT SITUATION OF THE STUDIED PROBLEM

Integrated circuits (IC) are used in every industry field at present. Demand of their performance is increasing rapidly. Therefore their complexity and number of devices in one IC is growing quickly.

Since digital part of the chip covers usually 90 % of chip area IC technology development focus mainly on improving digital circuits. Switching time, complexity and power consumption are main criteria that push IC technology to shorter length of transistor to reduce chip area, increase operating frequency and lower supply voltage to decrease power consumption.

The software that automates digital part of the chip is available and used frequently for many decades. Digital synthesizers from hardware description language (HDL) and tools for place and route of the synthesized netlist into the final optimized layout are used in everyday industry work saving lot of digital circuit design time.

Analog part of the IC covers just about 10 % of the chip area. But its design and validation takes about 90% of the time needed to design the whole circuit. It can be even worse because of the trends in IC technology development. Lower supply voltage and shorter transistor L makes the analog design more complicated and challenging.

The aforementioned reasons lead to the growing needs of robust tools which can help to automate parts of the analog design flow. An automated design of the analog circuits can save enormous part of the design time and expenses needed to design the chip. At present much effort is spent to develop an analog synthesis tool or OT that will shorten the design time of the analog circuits. Many scientists and research institutions all around the world are trying to develop and improve such tool [1, 2, 3, 4, 5].

It is not usual that the design teams use similar tools for the analog part of the IC. The development of OT is not in such stage yet to be able to use it in industry design fully. Commercial optimization tool [6] exists. Nevertheless it is used just as a support tool for example to fine tune some circuit parameter of already designed circuit. This tool is not used to create final schematics of analog circuit from the scratch.

Typical analog circuit synthesizer works in three steps [7] shown on Figure 1-1:

- First, circuit architecture is chosen in accordance with the specification.
- Then the devices in the circuit are sized by the optimization.

- Finally the devices in the circuit are automatically placed to the layout and routed together.

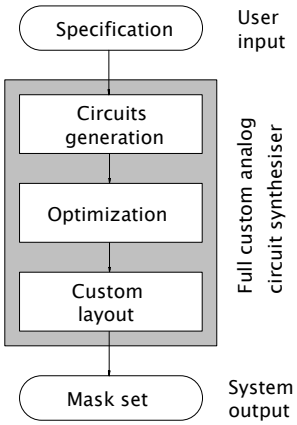


Figure 1-1. Flow diagram of the analog synthesis

Step two – the optimization – is usually the most challenging task. It also consumes most of the analog design time, e.g. proper sizing of the circuit.

Since I work as an analog IC designer I personally feel the need of a tool to – at least – optimize the analog circuits. Thus I chose the analog circuit optimization as the topic of my thesis.

The topic of the analog circuit synthesizer is quite old [7] and has generally been considered a difficult problem [2]. No powerful, reliable and versatile tool for automated analog circuit design has been created yet. Any optimization tool is widely used in industry analog circuit design.

The analog design automation approaches published so far for typical analog ICs are [8, 9]:

- Knowledge based [2].

Following three design automation approaches are optimization based. They use an optimization engine instead of a design plan to perform the design task.

- Equation based [3, 10, 11].
- Simulation based [12, 13, 14, 15].
- Learning strategy based [16, 17].

Main drawbacks of knowledge based approach are a lot of work to define a new design plan and time consuming process to encode design knowledge for a given set of specifications. The equation based methods are not accurate enough to design automatically analog ready-to-use circuits. Learning based strategies can produce powerful circuit but their setup time can be longer than a design without any OT because of training samples creation. Simulation based tools produce the most accurate circuits and its setup time is the shortest. Therefore the simulation based approach was chosen despite the fact that the computation time is longest in this approach [7].

Many works about the automated analog circuit design published recently are quite sophisticated and present powerful analog circuit synthesis ideas and improvements. On the other hand these approaches or the principles they present are not really usable in industry OT since they do no meet the requirements mentioned in Chapter 2.

Optimization methods [3, 12, 18, 19] use equation based optimization method that produces generally not very accurate results. The reason is that the equation based optimization uses just simplified description of the circuits by the equations and usually uses simplified transistor models.

Labrar et al. [20] use equation based rough pre-optimization that can be also difficult for more complex circuits that are not easy to describe by equations. That work uses it to describe well know two-stage Miller OTA.

Optimization method used by Pereira-Arroyo et al. [10] generates just a map of results and the user needs to choose the solution manually. This can take a long time in case of solving of complicated designs.

Automation algorithm presented by Jafari et al. [1] uses simulation based method just in the initialization phase that should lead to inaccurate results.

Somani et al. [2] use knowledge based initial setup of their method resulting in long setup time.

Method presented by Bo et al. [4] is not very robust since it would take long time to find penalty coefficient value. This penalty coefficient is used in their tool. To find those coefficients can cause long setup time as well.

The tool presented by Barros et al. [16] is not very useful since long setup time is required for their learning strategy. Its accuracy is questionable as well since its accuracy strongly depends on the number of training samples.

OT presented by Mishra et al. [17] uses knowledge based learning mechanism. It can be inaccurate, it can cause wrong decisions and it can be also problematic for more complex circuits.

Algorithm presented by Fakhfakh et al. [5] just replaces all design variables by one variable (substitution) that disables the option to use (most accurate) simulation based optimization approach.

Optimization approach presented by Thakker et al. [13] would not produce very robust design because of performing PVT (Process Voltage Temperature) corner simulations after the optimization to save the computation time. Moreover Doménech-Asensi et al. [21] presented a tool that does not perform PVT analysis at all.

Quite powerful commercial tool [6] is used widely to partly automate the design work. It is quite hard to design circuits purely automatically. It needs quite long setup time to create test-benches needed for the optimization and manual definition of the design task that can take longer time than optimization using the proposed tool. The created design examples are also technology dependent. Moreover the optimization method is not under control like in out optimization method using the novel optimization watchdog feature.

State-of-the-art of the analog circuit design automation is well described in [8, 9]. Moreover open research points in this field are discussed in [8].

I propose a novel Optimization watchdog feature to deal with one of these open points. It reduces the design space of the optimization task (to reduce intervals of widths and lengths of transistors in the design). This reduction leads to faster convergence of the optimization progress and thus to the reduction of the computation time. It also helps the optimization algorithm to find better circuit if the design space is limited to the area where the optimal circuit is not.

I implemented another novel feature. It sizes current mirror transistors to be more accurate that the one used in [14, 15]. It is based on transistor sizing dependent on the current flowing through the transistor. The transistor sizes are based on interpolation or extrapolation using look-up table with simulated transistor sizes.

2. AIMS OF THE DOCTORAL THESIS

My work is mainly focused on the optimization of the analog circuits. I worked on creation of novel OT for industry everyday design work. The tool shall be able to generate sized circuit schematic ready for the application and to save most of the design time. The OT shall be able to size the devices of various analog circuits in accordance to their circuit specification.

I enhanced my OT by two novel features I had developed. Optimization watchdog causes shorter optimization time of the design task and increases

the possibility to converge to the better results. Advances current mirror sizing algorithm generates current mirrors with good matching and thus with higher accuracy.

The tasks solved in my work are listed below:

- Creation of the OT for analog circuit optimization.
  - Design of the OT core – implementation of the chosen optimization algorithm.
  - Design of the OT interface between the core and the user to reduce OT setup time.
  - Implementation of the PVT analysis to design robust circuit but in a reasonable time.
- Optimization watchdog feature design and its implementation.
  - Its composition to the optimization algorithm.
  - Verification of its benefits on design examples.
- Novel algorithm of the current mirror design .
  - Simulation of mirror transistors for different currents.
  - Creation of the algorithm for automated transistor sizing.
  - Implementation of the novel algorithm to the OT.
- Optimization of several analog circuits using the OT to verify its function.
  - Design of the test-benches needed to extract circuit parameters.
  - Using the OT to optimize the circuits.
- Chip - containing the optimized circuits - creation to verify the complete design flow from the specification to the measurement of the chip.
  - Chip design.
  - Its layout.
  - Fabricated chip measurements.

The novel OT shall be usable in the industry design work thus satisfying following requirements:

- Very short setup time of the OT. This is the time needed to setup the tool according to the specific design task before the automated optimization is started. If the setup time is long (several hours or even days) the development of the circuit using standard design ways can be equivalent (or even shorter) to the automated one.

- Accurate ready-to-use results produced by the novel OT. The optimization of the circuit must take into account PVT corners since PVT analysis is able to find the worst case of each specified circuit parameter. Moreover the simulations of the optimized circuit must be done by using realistic models of the devices used in IC technology (transistors, resistors, capacitors etc.). It makes the results of the automated design more reliable.
- Robustness of the tool. It means that the OT must be able to converge to the solution for a generic circuit and its specification (not only for limited range of the circuits). Usage of the robust optimization algorithm is also needed to ensure convergence to the global extreme and not stuck during the optimization progress.

Scientific contributions of my work are:

- Optimization watchdog feature
  - reduces design space during the optimization progress to improve convergence speed and quality of the results.
  - controls the optimization progress to shorten optimization time.
- First OT ever able to optimize analog circuit by full PVT simulation using real technology models.
- OT with very short setup time.
  - No need to create circuit schematic, test-benches and define extraction of the circuit parameters – design examples are re-usable.
  - OT implemented to the widely used design environment.
- Possible optimization of a generic circuit.
- Optimization independent on the design technology.
- Novel algorithm for design of accurate current mirrors.
- Verification of the OT by the measurement of the chip containing circuits optimized by the created OT.

3. WORKING METHODS

Since a lot of companies all around the world use Cadence design environment the OT user interface is inserted to main Cadence CIW. This kind of implementation was used to ensure easy usage of the OT and to ensure its very short setup time. It is implemented by adding a new "Optimize" toolbar by a script written in Skill language. It is possible since

Cadence design environment is created to enable its modification to be more user friendly. The Cadence design environment is created by Skill and it is also possible to modify it by that language. Each modification is caused by modules/script in text file format that is loaded by Cadence during its start.

The new toolbar contains sub-menus for each circuit type that can be optimized by the proposed OT. Specification of the circuit is inserted to simple filling form that appears after choosing of the circuit to be optimized. This form can be filled very quickly that ensures very short setup time of the new OT. The OT core is entered from that form after it is filled.

The OT core is written as the Ocean script. This language is also created by Cadence thus cooperate with Cadence and Skill scripts well. The main reasons to implement OT core by Ocean scripting language are:

- Ocean script can be launched from Cadence CIW by a Skill script.
- Mathematical function of the optimization algorithm can be programmed by the Ocean script.
- Pre-created net-lists of the circuits to be optimized can be loaded by the Ocean script.
- Ocean can run spectre simulation to perform circuit analysis.
- Spectre simulation results can be post-processed by Ocean to extract circuit parameters.
- Corners of the simulations can be changed by the script to perform PVT simulations.
- Parameters (design variables) of the circuit that is optimized can be loaded from parameter file for spectre simulator.

Text output files are created by the script as well. These files contain details about the optimization process to be track-able to be possible to make a decision about the circuit, specification or bounds of the design variables.

The OT core uses the net-list of the test-benches of the optimized circuits to simulate them to extract its parameters that are optimized. Those test-benches and their net-lists are pre-created for the OT in the Cadence design environment for each circuit that can be optimized by the proposed OT.

Optimization watchdog is simple but effective algorithm. It needs just definitions of several variables and several arithmetic operations. It is possible to implement it to the OT core by Ocean language.

Current mirror design algorithm is based on several transistor simulations. Results of those simulations are implemented to the OT by definition of several variables. The arithmetic operation needed by the

algorithm (interpolation and extrapolations of the simulated values) are implemented by the Ocean language.

The circuits optimized by OT are designed in AMIS 0.35 μm CMOS technology and are optimized by PVT simulations.

After the OT was created the circuit optimization was tested. Then layout of the chip containing the optimized circuits was created in Cadence Virtuoso tool and gds data were sent to Europractice to be manufactured. Finally parameters of the circuits that were optimized were measured in the laboratory to verify the OT and complete design flow from the circuit specification to the fabricated chip measurements.

4. RESULTS

Optimization Tool

The OT core interface is implemented to the Cadence GUI by the Skill language to be easily improved or expanded in the future. Cadence was chosen because it is frequently used analog design environment. That kind of implementation makes the setup time of our optimization very short (few tens of seconds). Short setup time is one of the main criterions of designed optimization tool. The flow diagram of the OT is shown on Figure 4-1.

The optimization core together with implementation of the Differential evolution optimization method was designed using Ocean scripting language. Designed scripts enable:

- Implementation of the optimization algorithm.
- Spectre circuit simulations required by the optimization method.
- Post-processing of the simulations output to extract circuit optimized parameters.

The GUI is used only to select the circuit type to be optimized, enter the circuit specification and run the optimization procedure. The SKILL script was created to insert new “optimize” toolbar to the Cadence main window. The circuit which has to be optimized can be selected together with the definition of circuit parameters in that toolbar. The specification of the circuit is send to the optimization core as a text file in the format of Ocean scripting language.

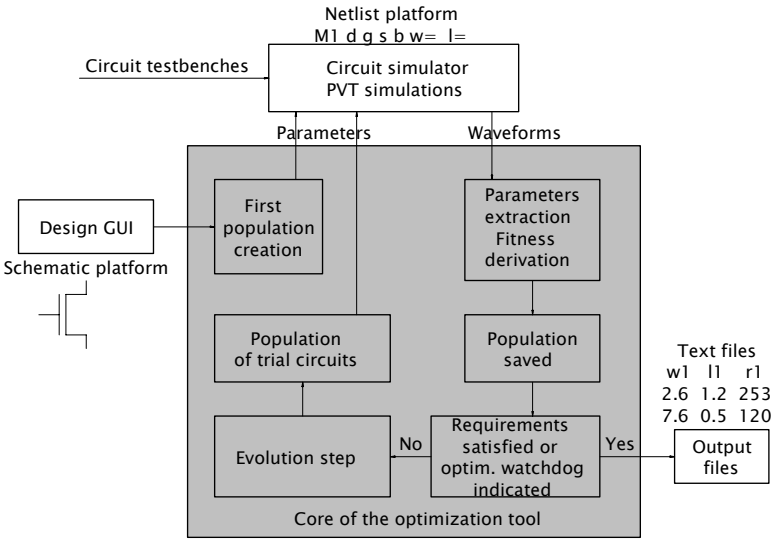


Figure 4-1. Algorithm of the OT.

The optimization core uses pre-created netlist of the optimized circuit test-benches to run Spectre simulation. The design variables (as transistors widths and lengths) are sent to circuit simulator by text file in the Spectre simulator format. Output of the optimization tool is text files containing the details of all the circuits in all created populations.

The main optimization tool core script is shown on Figure 4-2. It calls several second-order scripts that are not included in this statement because covers together more than 1000 lines.

First of all the initialization of the optimization tool is made (scripts *declaration.ocn* and *skill_out.ocn*). Design variables bounds, parameters of the differential evolution (population size *n*, scaling factor *SF*, crossover probability *CR*) are set. Moreover design specification is loaded from *skill_out.ocn* script. That script is created by the Skill script of the optimization tool interface. The last step of the initialization phase is the opening of the output files (design variables values and circuit parameters values).

Then first population is created (*first_params.ocn* script). PVT Spectre simulations of the circuits in the first population are run. The worst case optimized circuit parameters are extracted from the simulations results (*first_sim.ocn* script). The details about first population are stored in the

output files (*output_data.ocn* script). The fitness of the circuits in the first population is computed and checked if the specified circuit was found. The optimization process continues until the solution is found.

```
load("/2s_ota/skill_out.ocn")
load("/2s_ota/declaration.ocn")
of=outfile("/2s_ota/results/outputfile.scs"))
xff=outfile("/2s_ota/results/xfactor_file.scs"))

for(i 1 n
   load("/2s_ota/first_params.ocn"))
   load("/2s_ota/first_sim.ocn")
   load("/2s_ota/output_data.ocn")
   done = if(((F[i]=0.0)||(done==1)) 1 0)
)

z = z + 1

while((done<1)
   for(i 1 n
      load("/2s_ota/random.ocn")
      load("/2s_ota/param_evo.ocn"))
      load("/2s_ota/evo_sim.ocn"))

      delta = F[i] - Ftmp
      stuck = if(((delta>=wdd)||(stuck==0)) 0 1)

      load("/2s_ota/evo_new_population.ocn"))
      done = if(((F[i]=0.0)||(done==1)) 1 0)
      load("/2s_ota/output_data.ocn"))
   )
   z = z + 1
   wdc = if((stuck==1) (wdc+1) 0)
   stuck = 1
   done = if((wdc>=wdp) 1 done)
)

load("/2s_ota/final_output.ocn"))
close(of)
close(xff)
```

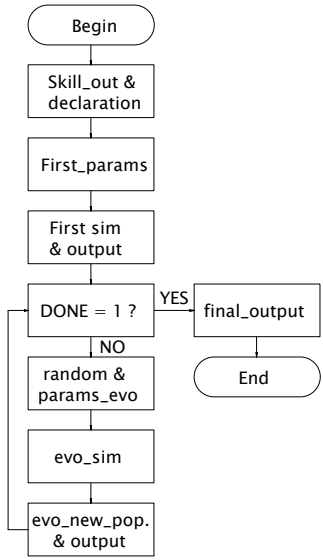


Figure 4-2. OT core top script.

Only the worst case corner simulations are run for each circuit parameter to speed up the optimization. It was needed to run the optimization for all corners to determine the worst case for each circuit parameter. It was done during the design examples creating (only one population of one individual was enough to specify the worst case).

The procedure of the differential evolution (mutation, crossover and selection) is done in the second *"for"* loop. Three random numbers are computed (*random.ocn* script), design variables of the trial circuits are generated (*param_evo.ocn* script) and trial circuits are simulated. Their fitness is computed and new population is created (*evo_new_population.ocn* script). Again the details of the new population are stored to the output files. The *"while"* loop is run again until the final condition is satisfied.

Optimization Watchdog

First final condition of the optimization is the occurrence of circuit with fitness function equal to zero – goal of the optimization. Another final requirement is the occurrence of predefined number of populations (parameter WD_P) in a row without significant progress of the optimization. It is defined by specific difference between fitness of the trial and target circuit – parameter WD_D. This indicates that the optimization is not able to get much better circuit in reasonable time. This is the baseline of the novel feature – optimization watchdog – that helps to optimize better circuit in shorter time. The watchdog is implemented as follows:

$$\begin{aligned} WD_C &= 0 \; if \; \exists i = 1,...,n \;\; F_i(t+1) \le F_i(t) - WD_D \\ WD_C &= WD_C + 1 \quad otherwise \end{aligned} \tag{1}$$

where WD_C, WD_D and WD_P are special watchdog variables, n is the number of individuals in one population. The optimization ends without satisfying the specification if WD_C is equal to WD_P.

WD_D and WD_P parameters are set to their default values by the OT. On the other hand the setting can be changed by the tool user. WD_C variable is evaluated during the optimization process by the tool as shown on Figure 4-1.

The first reason for optimization watchdog implementation is the natural feature of the differential evolution. The optimization process can diverge if the specification of the circuit is set beyond its limits. It can also diverge if the bounds of the design variables are set too high or too low. If several populations without significant individual improvement occur the optimization is stopped consecutively.

The main reason for the optimization watchdog implementation is the convergence time improving by a reducing of the search space. The output file generated by *final_output.ocn* script contains information about design variables bound settings and indication how the bounds should be improved. The ranges of some design variables are narrowed down to reduce the search space to improve convergence speed of the optimization.

The simplified Ocean script that recommends how to modify the design variables bounds is shown on Figure 4-3. First, individuals in the final population are sorted from the best one to the worst one. Then m variable is counted. It denotes to the number of the best circuits taken into account in the bounds modification. We chose m to be 50 % of n (rounded down). It is a trade-off between optimization optimality in terms of convergence to the

local minimum danger and acceleration. Last step of the script creates new bounds of the design variables. Figure 4-3 shows an example with only one design variable w1.

```
m = n - 1
Ftmp=F[1]
for(i 1 m
        for(j 1 m
                while((F[j+1]<F[j])

                Ftmp=F[j]
                F[j]=F[j+1]
                F[j+1]=Ftmp

                w1tmp=w1[j]
                w1[j]=w1[j+1]
                w1[j+1]=w1tmp
)))

if((mod(n 2)>0) m=((n-1)/2) m=(n/2))

w1min_new = w1[1]
w1max_new = w1[1]

for(i 2 m
if((w1[i]<w1min_new) w1min_new=w1[i])
if((w1[i]>w1max_new) w1max_new=w1[i])
)
```

Figure 4-3. Ocean script – design variables bounds modification.

The optimization watchdog first generates WD_P low (from 1 to 2 in dependence on the circuit complexity – 1 for circuit with about 5 design variables and 2 for circuit with 15 design variables) and WD_D high (from 0.05 to 0.1 – lower value for higher number of design variables) for short optimization which quickly scans the circuit, its specification and setting of the design variables bounds.

Then the design variables bounds are improved in accordance to the output of the first short optimization and the circuit requirements (using the script shown on Figure 4-3). WD_P is set higher (from 3 to 5) and WD_D lower (from 0.01 to 0.03) for precise optimization that is able to generate powerful circuits.

Second task performed by the watchdog is to identify the design variables bounds set inappropriately. It recommends what bound should be extended to achieve enhanced circuit performance.

The specification can be found to be beyond the limit of the circuit after first short optimization. At that point the specification can be changed for example by some trade-off between consumption and slew-rate of the circuit.

Current Mirror Sizing

Accurate approach to size the current mirror transistors was used for all design examples. It is based on using the same width and length of all current mirror transistors. Multiple transistors in parallel are used to increase or decrease bias current in the specific branch of the circuit. It is much better transistor matching approach than to size widths and lengths of the current mirror transistor independently as in [14, 15].

Current I_B (μA)	**NMOS**			**PMOS**		
	W (μm)	L (μm)	W/L/I_B (μA^{-1})	W (μm)	L (μm)	W/L/I_B (μA^{-1})
0.1	1.0	16.5	0.61	0.5	1.6	3.13
0.2	1.0	8.5	0.59	1.0	1.6	3.13
0.5	1.0	3.5	0.57	1.4	1.0	2.80
1.0	1.0	2.0	0.50	2.7	1.0	2.70
1.5	1.0	1.4	0.48	3.9	1.0	2.60
2.0	0.9	1.0	0.45	5.0	1.0	2.50
3.0	1.3	1.0	0.43	7.3	1.0	2.43
5.0	2.0	1.0	0.40	12.0	1.0	2.40
10.0	4.0	1.0	0.40	10.2	0.5	2.04
20.0	8.0	1.0	0.40	20.0	0.5	2.00

Table 4-1. W/L of the current mirror transistors.

I used a look-up table for setting correct dimensions of the current mirror transistors. I simulated width and length of the transistors to have correct operation point (to be in the strong inversion). The simulation details are

listed in Table 4-1. for various bias currents and for PMOS and NMOS transistors.

Thus the width and length of the current mirror transistors are not optimized (they are not design variables). They are set to be in correct mode in accordance with the bias current that is optimized. It is done by "rule of thumb" used in analog circuit design:

$$V_{GS} \geq V_{TH} + 100\,mV \qquad (2)$$

Where V_{GS} is gate-source voltage of the current mirror transistor and V_{TH} is its threshold voltage. This rule must be fulfilled for all PVT corners.

The dimensions of the transistors are finally sized in accordance with the bias current and values in Table 4-1. Specific minimum size of the transistors is also taken into account. It was set to 2 μm^2.

Widths and lengths of the PMOS and NMOS current mirror transistors are shown on Figure 4-4. for various bias currents. Values for current higher than 20 µA and lower than 0.1 µA are extrapolated. The dimensions generation is not very sophisticated but simple and well working.

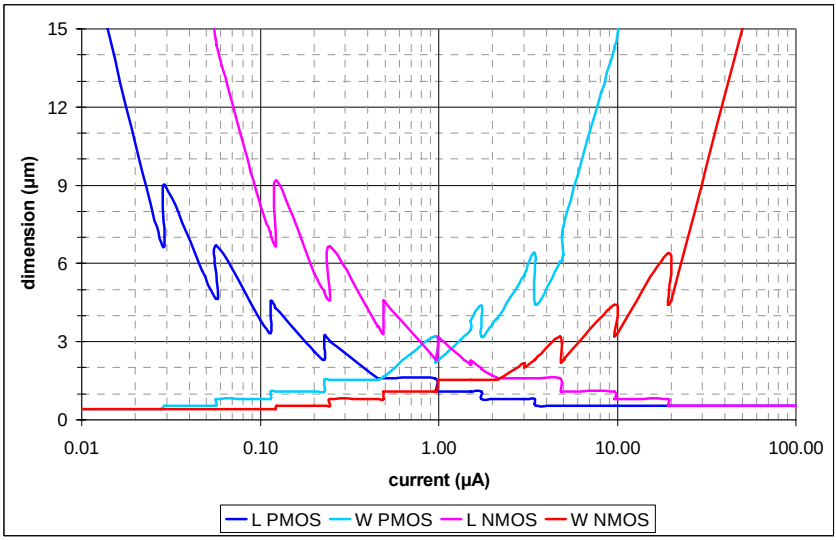


Figure 4-4. Width and length of the current mirror transistors.

Design Examples

The presented optimization tool was tested on optimization of two-stage Miller OTA. This circuit was also used to compare the performance with other works since it is most frequently used circuit in the works published recently. This comparison is not described in this statement but showed that my OT is competitive to other optimization approaches presented so far. Other design examples - folded cascode OTA and voltage regulator – were also implemented to the OT. All these three circuits optimized by my OT were fabricated and measured. Measurements are not described in this statement but were successful.

All circuits were designed using AMIS 0.35 μm CMOS technology device models. The models are referenced in the core script. The technology can be easily changed if required by changing of the references in the script. All optimization were run for temperature range -10 °C to 50 °C, supply (input) voltage range 1.8 V to 2.0 V and bias current variations of 30%. The optimization were run on 2.5 GHz two core Intel processor with 4 GB RAM. Most of the optimization time was consumed by Spectre circuit simulations. 7.0.1.091 Spectre simulator version was used.

Population size *n* was chosen to 15 individuals for all optimized circuits. It was proven to be large enough for the optimization convergence and small enough for the optimization speed. Scaling factor *SF* was set to 0.8 and crossover constant *CR* to 0.7. Those values were found as a good compromise between the optimization convergence speed and the possibility to obtain powerful circuits.

Implemented two stage OTA schematic is shown on Figure 4-5. with design variables highlighted in red. Width and length of the transistors *M1*, *M2* and *M3* (parameters *W1* and *L1*) are not strictly optimized but derived from the bias current by the novel approach. The current flowing through branches of transistors *M2* and *M3* are optimized by design variables *M1* and *M2*. These variables denotes to the number of transistors in parallel. The number of design variables is 11. Load of the OTA was chosen to be purely capacitive (10 pF).

I optimized following circuit parameters of this design example:

- Zero frequency voltage gain – A_0.
- Unity gain bandwidth – GBW.
- Phase margin – PM.
- Slew rate – SR
- Current consumption – I_{DD}.

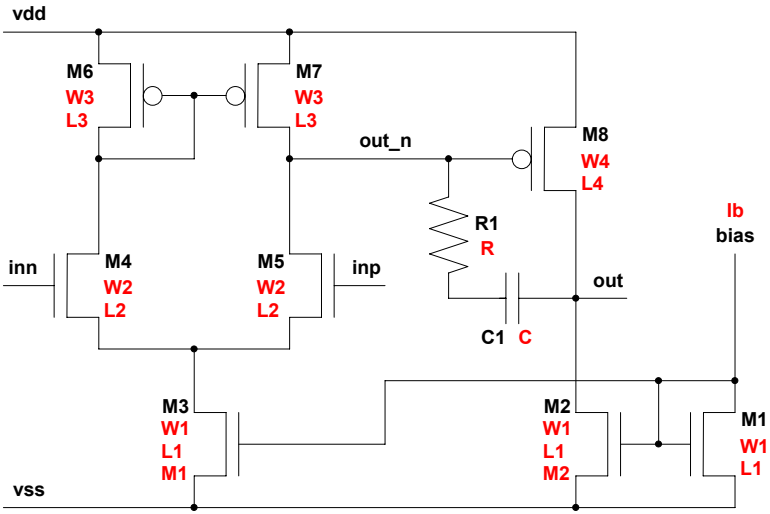


Figure 4-5. Two-stage Miller OTA.

Verification of the optimization acceleration by using of the optimization watchdog is listed here. Demanding specification of the circuit was set and the optimization watchdog parameters were set 1 for WD_P and 0.1 for WD_D for the first rough optimization. The bounds of the design variables were updated in accordance to the result of the first optimization. The parameters WD_P and WD_D were set to 3 and 0.05 respectively for the second optimization with the reduced search space. Parameter WD_P was set to 5 and parameter WD_D to 0.01 for the simulation without using of the watchdog.

Table 4-2. contains information about the circuit specification, optimization results and optimization results without using of the watchdog. The optimization time of the first watchdog optimization was 120 minutes (solution found in 10^{th} population). The second watchdog optimization lasted 276 minutes (solution found in 23^{rd} population). Thus the complete optimization time was 396 minutes with using of the watchdog. The optimization time was 960 minutes (solution found in 80^{th} population) without using of the watchdog. Thus the optimization time was reduced more than two times. Design variables with their bounds before/after the reduction of the search space and optimization results values are in Table 4-3.

Large differences between two optimized circuits are apparent. The reason of these differences can be seen in existing of more comparable

solutions. It is possible since the circuit specification is not a global extreme of the fitness function.

Parameter	A_0	PM	GBW	SR	I_{DD}
Specification.	90 dB	60 °	2.0 MHz	2.0 V/µs	20 µA
Result - WD	90 dB	64 °	2.8 MHz	2.3 V/µs	16 µA
Result - ~~WD~~	94 dB	60 °	2.4 MHz	2.2 V/µs	13 µA

Table 4-2. Circuit parameters - two-stage Miller OTA.

Variable	Lower bound	Upper bound	Result - WD	Result - ~~WD~~
W1 (µm)	0.4	50	2.6	1.6
L1 (µm)	0.55	50	1.6	2.7
W2 (µm)	0.5/10	50	37.2	50.0
L2 (µm)	0.5	20	1.7	5.5
W3 (µm)	0.5/8	50	41.0	18.1
L3 (µm)	0.5/13	50	49.4	50.0
W4 (µm)	0.5/9	50	38.4	46.5
L4 (µm)	0.5	10/7	0.7	0.5
R (k Ω)	0.1	10	0.1	0.1
C (pF)	0.1	10/5	0.58	0.38
M1 (-)	1	10	1	2
M2 (-)	1	10	2	6
Ib (µA)	0.1	30/9	4.0	1.2

Table 4-3. Design variables – two-stage Miller OTA.

5. CONCLUSION

I have created a novel OT for industry analog IC automated design. The tool is implemented to the Cadence design environment to achieve very short setup time of the tool. It also enables its easy use, possible improvement and extension. The tool uses most accurate simulation based optimization approach and robust version of differential evolution algorithm to be able to

optimize generic circuit architecture. My OT optimizes circuits using full PVT analysis to ensure robust resulting circuits that are usually ready to use. Simple but robust differential evolution method was chosen as a best candidate for optimization algorithm. It causes the tool to be robust and thus to be able to converge to the solution for every design example.

I developed optimization watchdog feature to enhance the circuit optimization. This novel feature was implemented to the OT. Its main purpose is to reduce the search space and thus to accelerate the optimization progress leading to shorter automated design time. This acceleration allows the OT to find better circuits. These circuits would not be found without this feature. The reason is that their optimization would take so long time that the optimization would be stopped before they are found. The optimization watchdog can shorten the optimization more than two times.

Another novel feature for accurate current mirror automated design was designed. This feature was implemented to the OT. It is based on extracting the transistors dimension in accordance to the current flowing through the transistors. Their matching is accurate in comparison with random transistor sizing presented in [14, 15]. This novel approach was successfully used in optimization of three design examples.

The proposed OT was compared with the works presented recently [11, 14, 15] with good results. I did the comparison using two-stage Miller OTA design example since it is most frequently used circuit in automated design approaches. The comparison was a difficult task since not all details needed to compare the results fully were presented in these works.

I proposed and used simulation acceleration feature to reduce number of corners to be simulated in PVT analysis. It is based on finding worst case corner of each circuit parameter and simulating only in this corner. The drawback is that the worst case corners were extracted only from several random circuit full PVT simulations. It requires further development for improvement.

Three design examples were implemented to the presented OT: Miller two-stage OTA, folded cascade OTA and voltage regulator. The design examples were successfully optimized by my tool using AMIS 0.35 μm CMOS technology. The tool is able to optimize a generic circuit in any technology.

My optimized circuits were layouted and fabricated as a chip in Europractice in the AMIS 0.35 μm CMOS technology. The circuits on this chip were measured to verify their performance and thus to verify the OT. Some measurements were complicated since the values were low thus were affected by noise and measurement equipment accuracy. Another difficulty was different conditions of simulation and measurement. Measurement

conditions were unknown in some cases and some measurements were disturbed by other circuits on the chip. Nevertheless the measurements were successful. Their results were close or better than the optimized values.

List of literature used in the thesis statement

[1] A. Jafari, S. Sadri, M. Zekri, *Design Optimization of Analog Integrated Circuits by Using Artificial Neural Networks*, International Conference of Soft Computing and Pattern Recognition, 2010, p. 385-388.

[2] A. Somani, P.P. Chakrabarti, A. Patra, An Evolutionary Algorithm-Based Approach to Automated Design of Analog and RF Circuits Using Adaptive Normalized Cost Functions, IEEE Transactions on Evolutionary Computation, 2007, vol. 11, p. 336-353.

[3] A. Bennour, A. Sallem, M. Kotti, E. Gaddour, M. Fakhfakh, M. Loulou, *Application of the PSO technique to the Optimization of CMOS Operational Transconductance Amplifiers*, 5th International Conference on Design and Technology of Integrated Systems in Nanoscale Era, 2009, p. 1-5.

[4] L. Bo, W. Yan, Y. Zhiping, L. Leibo, L. Miao, W. Zheng, L. Jing, V.F. Francisco, *Analog circuit optimization system based on hybrid evolutionary algorithms*, Integration, the VLSI Journal, 2009, vol. 2, p. 137-148.

[5] M. Fakhfakh, A novel Alienor-based heuristic for the optimal design of analog circuits, Microelectronics Journal, 2009, vol. 40, p. 141-148.

[6] Cadence Design Systems, Inc., Virtuoso NeoCircuit, http://www.cadence.com/ rl/Resources/datasheets/VirNeoCircuit_ds.pdf (Accessed February 2012).

[7] R. Spence, C. Toumazou, P. Cheung, C. Makris, C. Berrah, M. Singha, Xiao Xiangming, J. Stone, *Approaches to Analogue IC synthesis*, IEE Colloquium on VLSI Analogue Design, 1989, p. 1-6.

[8] E. Tlelo-Cuautle, I. Guerra-Gomez, M.A. Duarte-Villasenor, L.G. de la Fraga, G. Flores-Becerra, G. Reyes-Salgado, C.A. Reyes-Garcia, G. Rodriguez-Gomez, *Applications of Evolutionary Algorithms in the Design Automation of Analog Integrated Circuits*, Journal of Applied Sciences, 2010, vol. 10, p. 1859-1872.

[9] M.F.M. Barros, J.M.C. Guilherme, N.C.G. Horta, *State-of-the-art on Analog Design Automation*, Studies in Computational Intelligence, 2010, vol. 294, p. 19-47.

[10] R. Pereira-Arroyo, F. Nicaragua-Guzmán, A. Chacón-Rodríguez, *Design of an Operational Transconductance Amplifier applying multiobjective optimization*, Argentine School of Micro-Nanoelectronics Technology and Applications, 2010, p. 12-17.

[11] S.L. Sabat, K.S. Kumar, S.K. Udgata, *Differential Evolution and swarm intelligence techniques for analog circuit synthesis*, World Congress on Nature & Biologically Inspired Computing, 2009, p. 469-474.

[12] J. Maršík, O. Šubrt, P. Martinek, *Developing Automated Design Procedure for Operational Amplifier blocks*, International Conference on Signals and Electronic Systems, 2008, p. 269-272.
[13] R.A. Thakker, M.S. Baghini, M.B. Patil, *Low-Power Low-Voltage Analog Circuit Design Using Hierarchical Particle Swarm Optimization*, 22nd International Conference on VLSI Design, 2009, p. 427-432.
[14] Y. Jianhai, M. Zhigang, *A design method in CMOS operational amplifier optimization based on adaptive genetic algorithm*, WSEAS Transactions on Circuit and Systems, 2009, vol. 8, no. 7, p. 548-558.
[15] P. Prem Kumar, K. Duraiswamy, *An optimized device sizing of analog circuits using particle swarm optimization*, Journal of Computer Science, 2012, vol. 8, no. 6, p. 930-935.
[16] M. Barros, J. Guilherme, N. Horta, *Analog circuits optimization based on evolutionary computation techniques*, Integration, the VLSI Journal, 2009, vol. 43, p. 136-155.
[17] B.K: Mishra, S.S India, *Novel CAD Design Methodology for Two Stage Opamp with Noise-Power Balance*, International Conference on Signal Acquisition and Processing, 2010, p. 287-290.
[18] B. Dimov, V. Boos, T. Reich, C. Lang, E. Hennig, R. Sommer, *A novel technique for CAD-optimization of analog circuits with bipolar transistors*, Advances in Radio Science, 2009, vol. 7, p. 219-223.
[19] B. Sahu, A.K. Dutta, Automatic Synthesis of CMOS Operational Amplifiers: A Fuzzy Optimization Approach, 15th International Conference on VLSI Design, 2002.
[20] L. Labrak, T. Tixier, Y. Fellah, N. Abouchi, *A hybrid approach for analog design optimisation*, 50th Midwest Symposium on Circuits and Systems, 2007, p. 718-721.
[21] G. Doménech-Asensi, J.A. López-Alcantud, R.R. Merino, *Simulation-based low-level optimization tool for analog integrated circuits*, The International Society for Optical Engineering, 2005.

List of candidate's works relating to the doctoral thesis

Publications in Impacted Journals:

[1] M. Kubař, J. Jakovenko, *A Powerful Optimization Tool for Analog Integrated Circuits Design*, Radioengineering, impact factor 0.739 (in press).

Publications in Reviewed Journals:

Patents:

Publications Excerpted by WOS:

Other Publications:

[2] M. Kubař, J. Jakovenko, *A novel Tool for Analog Integrated Circuits Optimization*, Proc. of EDS IMPAS CS 2013 (in press).

[3] M. Kubař, Optimization of Analogue Integrated Circuits Using Ocean and Differential Evolution, Proc. of POSTER 2010, ISBN 978-80-01-04544-2, 2010, p. 1-4.

[4] M. Kubař, J. Jakovenko, *Novel Analog Synthesis Tool Implemented to the Cadence Design Environment*, Proc. of SM2ACD 2010, ISBN 978-1-4244-6815-7, 2010.

[5] M. Kubař, J. Jakovenko, *Novel Methods of the Analogue Integrated Circuit Design Teaching*, Proc. of EDS IMAPS CS 2010, ISBN 978-80-214-4138-5, 2010, p. 190-195.

Response and reviews

Review of the article *A Powerful Optimization Tool for Analog Integrated Circuits Design:*

Integration of the proposed method in Cadence design environment is very helpful for enabling of regular users (not only scientists working in the field of analog circuit design automation research) to benefit from using of the optimization tools. The paper proposes novel optimization tool for analog circuit optimization. Differential evolution algorithm is enhanced using proposed watchdog feature which considerably accelerates tasks. Three optimizations were demonstrated in a detailed way with a comprehensive description of the results.

SUMMARY

The main goal of my Ph.D. thesis is creating of the optimization tool (OT) for sizing of the analog IC circuits. The OT shall be usable in practise industry design work to save most of the design time of basic generic circuits. Thus the proposed OT has very short setup time, it uses robust optimization algorithm and produce accurate ready-to-use results.

Moreover I implemented two novel feature to the OT – optimization watchdog – was implemented to shorten optimization time, to improve optimization convergence and thus to create better results. Another novel feature for current mirror transistor sizing was implemented. This feature ensures good transistor matching.

The OT is implemented to GUI (Graphical User Interface) of the Cadence design environment to be easily used. It is just needed to fill a form with the specification for the desired circuit and wait for the results. The OT performs full PVT (Process Voltage Temperature) simulation. Therefore the result of the optimization by the designed OT is circuit that usually needs no additional schematic change and is ready for layout.

The OT is implemented to the Cadence CIW (Command Interpreter Window) by the Skill language. The core of the OT is created using Ocean scripting language. A robust version of a differential evolution is used as the optimization method. I used accurate simulation based optimization approach for this tool.

Three types of the analog circuits were optimized by the OT. The layout of those circuits was designed and those circuits were fabricated in AMIS 0.35 μm technology by Europractice. The chip measurements finalized the verification of the OT as well as the complete design flow of the analog circuit from the specification to the fabricated chip measurements.

RÉSUMÉ

Hlavní cíl mé disertační práce je tvorba optimalizačního nástroje (ON) pro optimalizaci analogových integrovaných obvodů. ON má být použitelný při praktickém firemním návrhu obvodů, aby ušetřil většinu času potřebného při návrhu libovolných základních obvodů. Proto navržený ON potřebuje velmi krátký čas pro svoje nastavení, užívá robustní optimalizační algoritmus a vytváří přesné obvody.

Navíc jsem do ON přidal dvě zcela nové funkce – optimalizační hlídač – pro zkrácení doby optimalizace a zlepšení konvergence optimalizace, tudíž pro možnost tvořit lepší obvody. Dále jsem implementoval novou funkci pro návrh proudových zrcadel. Tato funkce umožňuje návrh proudových zrcadel s malým proudovým rozptylem.

Kvůli snadnému používaní je ON integrován do grafického uživatelské rozhraní návrhového prostředí Cadence. Stačí pouze vyplnit formulář se specifikací obvodu a počkat na výsledky. ON simuluje obvod ve všech rozích (technologických, napájecích a teplotních), takže výstupem optimalizace je obvod, který obvykle nepotřebuje žádnou změnu a je připravený k layoutu.

ON je integrováno do návrhového prostředí Cadence pomocí jazyka Skill. Jádro ON je vytvořeno pomocí skriptovacího jazyka Ocean. Robustní verze diferenční evoluce je použita jako optimalizační metoda. Optimalizace je založena na obvodových simulacích, což vede k přesným výsledkům.

Pomocí ON byly optimalizovány tři typy analogových obvodů, byl vytvořen jejich layout a dále čip byl vyrobený v AMIS 0,35 μm technologii ve firmě Europractice. Měřením těchto čipů byla ověřena funkce ON a také uzavřen proces návrhu od specifikace až po měření vyrobeného čipu.