AN INTELLIGENT DESIGN SYSTEM FOR ANALOGUE INTEGRATED CIRCUITS

Georges GIELEN(*), Koen SWINGS, Willy SANSSEN
Katholieke Universiteit Leuven, Dep. Elektrotechniek, ESAT-MICAS
Kardinaal Mercierlaan 94, B-3030 Leuven, Belgium
(*) research assistant with the national fund of scientific research (Belgium)

Abstract
A new design methodology for the design of arbitrary analogue functional blocks (op amps, comparators...) is presented. The method combines symbolic simulation, numerical based techniques and covers the whole design path from analytic modeling over optimal circuit sizing down to layout. This path is repeatedly passed through on different hierarchical levels for higher-level blocks.

The main advantage of the new CAD system is that it is not limited to a fixed set of circuit topologies. It can automatically design any new circuit by the introduction of artificial intelligence techniques: building block recognition, design equation manipulation, self-learning capabilities... These features make it an intelligent and flexible analogue design system.

1. Introduction
Advances in VLSI technology nowadays allow the realization of complex circuits and systems. ASIC's are moving towards the integration of complete systems on one chip, including both digital and analogue parts. However, the few analogue circuits require an increasing share of the total design time. Analogue design automation is therefore a field of growing interest.

The following observations are of primary importance for the development of analogue CAD tools:
• due to the large number of specifications imposed on analogue circuits, and due to the large variations of these specifications, the use of fixed analogue standard cells is uneconomic. To guarantee an optimal solution in terms of area, power and overall performance, the analogue circuits should be redesigned and tuned for every application, especially for ASIC's.
• analogue designers have developed a large number of circuit schematics performing the same function (such as comparators, comparators...). Each of them has advantages and disadvantages and is most appropriate in a certain range of performance specifications. Since analogue circuits are moving towards high-performance applications (high frequencies, low noise, low distortion), more and more new topologies are invented to overcome the shortcomings of the present-day ones.
• the complicated interaction between the device characteristics and the circuit performance is one of the major reasons for the very knowledge-intensive character of the design of high-performance analogue circuits. An efficient design strongly relies on the insight and expertise of the analogue designer: insight into the circuit behaviour but also into performance degrading second-order effects such as noise, distortion, temperature feedback...

The goals of analogue CAD tools are therefore the following:
1) they have to shorten the overall design time, from the behavioral system description down to silicon.
2) they have to avoid errors in the design. The resulting silicon should be guaranteed first-time correctly working (and testable).
3) they must be able to predict the performance of mixed analogue-digital designs, since analogue and digital blocks interact through the common substrate, common supply lines, capacitive couplings...
4) they have to capture and formalize the existing knowledge of analogue designers. Real analogue experts are rare, hard to find and hard to keep. Their knowledge should be transferred to the CAD system.
5) they have to provide insight to the tool user. This allows the transfer of the built-in knowledge to novice designers and shortens their education time.

In order to realize these goals, the analogue CAD tools are to be integrated into one framework, compatible with digital tools. This CAD system should be hierarchical, intelligent and open:
• the system may not be limited to a fixed set of topologies, but should be able to tackle new circuits entered by the user. This implies a built-in methodology to understand and to design an entirely new circuit.
• the system generally addresses two kinds of users: system designers who just want a circuit designed in a very short time, and more experienced circuit designers who want to explore existing schematics or try out new ones. Inexperienced (system) designers can run the system in automatic mode, relying on the built-in knowledge. On the other hand, expert designers should always be able to put in new knowledge and to ask explanation for the system's decisions. Self-learning capabilities should enable the system to learn from designers and to extract the knowledge out of the designers in a discrete way. analogue experts cannot be expected to program the way they design a circuit.
• in order to make system design feasible, the design methodology should also be structured on several levels of hierarchy. For analogue CAD however, these levels are not so strictly defined.

In Section 2 an intelligent, non-fixed-topology analogue design methodology is presented to automate the design of functional analogue blocks. The major tools in this CAD system are described in the following sections. Some of these tools have already been presented before and are described only briefly for the sake of completeness. In this paper, attention is focused on the overall analogue design methodology and the novel intelligent features. The analytic circuit modeling is presented in Section 3. Section 4 describes circuit sizing with the design equation manipulation and the optimization. Section 5 presents the analogue layout generation. Section 6 concentrates on the integration of the previous tools into an intelligent design system by the introduction of artificial intelligence techniques such as building block recognition and qualitative simulation. Concluding remarks are provided in Section 7.

2. An intelligent analogue design system
For the automatic synthesis of analogue building blocks, several approaches and tools are being published [1-7]. In each of these programs, the analogue circuits are resized for every application, to satisfy the performance constraints, possibly also optimizing an objective function. The programs [2-4] all use some analytic description (design equations) of the circuit to perform the sizing and all exploit expert designer knowledge to
some extent. However, the main drawback of these tools is that they are fixed-topology systems, limited to a fixed set of circuit topologies. None of them is able to automatically generate the appropriate analytic equations for a new topology. And the manual derivation of design equations is a tedious and error-prone job, especially for large circuits.

In IDAC [2] the circuit equations are - up till now - derived after an in-depth analytic study and solved explicitly by experienced designers, to yield a straightforward step-by-step procedure. This is a time consuming job, to be repeated for every circuit in the program's database. OASYS [3] uses design plans to successively select topologies and translate specifications downwards in a hierarchical knowledge-based synthesis framework. However, the construction of a design plan for a new topology requires an explicit representation of knowledge about the circuit behaviour, heuristic design decisions and performance trade-offs, which complicates the inclusion of new topologies. OPASYN [4] optimizes an analytic opamp model with a multi-start steepest gradient descent algorithm. Again, the analytic model for a new topology must first be created by a good analogue designer.

Other expert system approaches [5-7] store human designer's knowledge and exploit this knowledge for circuit design. However, none of them indicates how the knowledge is extracted from the human designers and how long it takes to include new topologies.

Overview of the new design methodology

In this paper, a new and intelligent analogue design methodology is presented [1,8]. It combines symbolic simulation, numerical optimization and knowledge-based techniques to allow an optimal circuit sizing and a fast inclusion of new topologies. The design methodology is schematically shown in Fig. 1. Generally, the sizing of an analogue circuit from performance and technology specifications is formulated as a constrained optimization problem, based on a circuit model. This model contains analytic equations, as well as general and circuit specific knowledge.

At the top, the user enters the design system with specifying the technology process and performance specifications for some type of analogue functional block. An expert system then inspects its knowledge base and proposes an appropriate topology for the given specifications and technology. The user can accept this topology, or he can enter a new one (or some part of it) himself. The result is passed to a building block recognizer. If the topology (or some part of it) is not present in the database, a circuit model is automatically generated by calls to the symbolic simulator ISAAC (Interactive Symbolic Analysis of Analog Circuits) [9-10]. The new model is then stored into the database so that it can be used in future design problems.

Next, the program DONALD (Design equatOn manipulation for Analog circuit Design) [11] transforms this circuit model, which is internally represented as an equation network, into a solution plan for the circuit. The OPTIMAN program (OPTIMization of Analog circuits) [1] then sizes all circuit elements based on this plan and the technology data, to satisfy the performance specifications. The degrees of freedom in the design are used to minimize a user supplied cost function (such as minimal power consumption and minimal noise).

The design is then verified. The use of circuit models and design rules speeds up the optimization and produces good, near optimal results. However, a final performance check is carried out with the SPICE simulator. If some specifications are violated, the optimization routine is recalled with the deviating specifications modified. If the design is accepted, the circuit schematic and the device sizes are passed to the layout program. If no acceptable solution can be found, control is transferred to DONALD, which traces the characteristics that are responsible for the failure. Consequently, a backtrace is carried out to trade off some input specifications or to modify or redesign a new topology.

The layout generator AUTOLAC (AUTOMATIC Layout of Analog Circuits) then lays out the building block, taking into account requirements such as minimum area, capacitive coupling, matching... Parasitic elements are extracted and a final verification is performed. If the design is not acceptable after layout, the circuit is resized or eventually a new topology is required. Finally, a complete macromodel is generated to simulate the building block at the system level.

Advantages and limitations of the methodology

Summarizing, the new design methodology is quite general and topology independent. It allows the automatic inclusion of new topologies and allows to optimize and explore a circuit in all dimensions of the analogue design space. The strength of the methodology lies in the automatic generation and manipulation of the analytic design equations. This emulates the manual design cycle of analogue designers: they design a circuit based on (manually derived) first-order formulas and then fine-tune the result with a numerical simulator (such as SPICE). Our methodology is similar, but the design equations are derived automatically in a fast and correct way, and can be much more accurate. Moreover, a designer cannot master a multitude of performance characteristics and their trade-offs at the same time. For higher-level functional blocks (such as A/D and D/A converters), the diagram of Fig. 1 is repeatedly passed through on different hierarchical (abstraction) levels.

The proposed methodology also has some limitations. First of all, it is only applicable to designs where analytic design equations can be generated. The present symbolic simulators cannot - yet - generate time-domain or large-signal characteristics. Hence, these expressions still have to be generated by a designer or taken from textbooks. Research is going on in this field. Secondly, the analytic equations are always approximations of the real circuit behaviour. The resulting design has to be verified carefully, and possibly has to be redesigned. However, since the design equations can be of virtually any accuracy [10], the number of design iterations is reduced to one or two. By using approximated formulas, the optimization is definitely faster than if a full simulation would be performed at every iteration [1].

The symbolic simulator, the optimization routine, the layout program and more specifically the integration of these tools into an intelligent design
system by means of topology selection, building block recognition and design equation manipulation, are now described in the following sections.

3. Circuit modeling with the symbolic simulator ISAAC

The symbolic simulator ISAAC [9-10] analyzes lumped, linear or linearized (small-signal) circuits in the frequency domain and returns analytic expressions with the complex frequency and (all or part of) the circuit elements represented by symbols. In this way, ISAAC generates analytic expressions for the ac characteristics of both time-continuous and switched-capacitor circuits. For semiconductor circuits, the exact expressions are usually lengthy and complicated. Therefore, a heuristic approximation algorithm has been built in. It simplifies the expressions based on the relative magnitudes of the elements and returns the dominant terms in the result [10].

ISAAC is used as a stand-alone program for instruction and designer assistance to gain insight into a circuit's behaviour. The analytic expressions are valid whatever the element values are and also indicate the dominant design variables. This is especially useful for second-order effects, such as distortion and PSRR, which are hard to understand from numerical simulation data.

However, ISAAC is mainly applied as a module in the general analogue CAD system. In our design strategy, all circuits are characterized by an analytic model. As opposed to IDAC [2] and OPASTYN [4], these circuit models are automatically generated by successive calls to the symbolic simulator. ISAAC returns symbolic formulas for transfer functions, CMRR, PSRR, impedances, noise... In this way, it generates analytic ac models for building blocks such as amplifiers, filters... This allows the fast inclusion of new circuit topologies into the CAD system.

Example

Consider for example the CMOS two-stage Miller-compensated op amp of Fig. 2. The expression for the gain-bandwidth (with a 10% error) is given by:

$$\frac{GM1 \cdot GM3}{2\pi \left[ CC \left( GM3 + CL + G03 + G04 \right) \right]} + \left( CL + CDH3 + CDH4 \right) (G01 + G02)$$

Formula (1) is more precise than the textbook formula $GM1/2\pi CC$ and the circuit behaviour can be accurately described by such analytic expressions [1].

![Fig 2. CMOS two-stage Miller compensated opamp](image)

Algorithmically, the analysis is based on the CMNA (Compacted Modified Nodal Analysis) formulation of the network equations. These equations are then solved by a dedicated and efficient determinant expansion algorithm, exploiting the sparse nature of the CMNA matrices. Typical CPU times on a 1-Mips machine are 8 and 26 seconds for the voltage gain of the Miller opamp and of a fifth-order switched-capacitor ladder filter, respectively [10].

4. Analogue design optimization

In our methodology (Fig. 1), the sizing of the selected circuit topology out of its analytic model is based on a generalized formulation of analogue design as a constrained optimization problem. Once the analytic circuit model has been generated (see previous section), the equation manipulator DONALD converts this model into a solution plan for the optimization by extracting the independent design variables and constructing a computational path out of the model. This information is then passed to the optimization program OPTIMAN, which sizes all elements to satisfy the performance constraints, thereby optimizing a user defined design objective.

The analytic circuit model contains a description of the circuit topology, design equations relating the circuit performance to the design variables, additional analogue expert knowledge about the circuit and other design constraints. As design variables, all node voltages, currents and element values in the circuit are taken, as well as all performance specifications. Most of these variables, however, are interrelated because of general constraints (such as Kirchhoff's laws), circuit specific constraints (such as design equations or matching information) and designer constraints (such as offset reduction rules). Hence, in order to allow an efficient optimization, a minimal set of independent variables has to be determined. This task is performed by the equation manipulator DONALD. The number of independent variables is equal to the degrees of freedom in the design and is nonzero in practical cases since there are more variables than equations. The independent variables can be considered as the regulators to tune the circuit. They form the fundamental design parameters, on which all characteristics of the circuit depend. If they are assigned values to, all other variables and hence the total circuit behaviour can be calculated.

Now some algorithmic details about DONALD are presented. Internally, the circuit model is represented as a bipartite graph in which the nodes either represent equations or variables. An example of a bipartite graph representing the following two equations is shown in Fig. 3:

![Fig 3. Illustration of bipartite graph and computational path construction.](image)
not unique, but some sets are computationally more advantageous than others.

Example

These ideas are now illustrated with a simple example. Fig. 3 shows the bi-partite graph of the equations (2) and (3). There are 5 variables \([GBW, GM1, CC, I1, VGT1]\) and 2 equations \([GBW_{eq}, GM_{eq}]\). Hence, there are 3 degrees of freedom and any set of 3 out of the 5 variables may be selected as the set of independent variables. If for example \(I1, GBW\) and \(CC\) are chosen as independent variables, the following computational path is constructed to compute \(GM1\) and \(VGT1\):

\[
GM1 = \text{out of } [GBW, CC] \text{ by means of } [GBW_{eq}] \quad (4)
\]

\[
VGT1 = \text{out of } [I1, GM1] \text{ by means of } [GM_{eq}] \quad (5)
\]

A set of independent design variables for the CMOS Miller opamp of Fig. 2 could be \([I1, V3, VGT2, VGT3, VGT4, CC]\).

Next, the set of independent variables and the computational path are used by the optimization program OPTIMAN to effectively site all circuit elements. During this optimization, the independent variables are varied. The performance specifications are treated either as objectives to be minimized (or maximized) or as (equality or inequality) constraints. For opamps designs, the goal function could be a weighted combination of characteristics such as power, chip area, noise...

The optimization algorithm is simulated annealing [12]. This is a general and robust method, based on random move generation and statistical move acceptance. It does not rely on special properties of the goal function and its derivatives. Since it also allows up-hill moves, a solution close to the global optimum will be found at the expense of a larger number of function evaluations. The simulated annealing implementation used is SAMURAI [13], an efficient kernel with fully adaptive temperature scaling and move range limitation and a novel inner loop criterion. These features keep the CPU-time consumption still acceptable for large circuits [1].

The general optimization loop within OPTIMAN is then as follows: the program statistically selects new values for the independent variables in a move range around the present value. It calculates all other (dependent) variables by means of the computational path, checks if the corresponding design satisfies all boundary conditions and constraints, calculates the goal function and statistically accepts or rejects the new state. This loop is executed until convergence occurs at lower temperatures, while the move range is gradually decreased with decreasing temperature. At the initialization, the independent variables are gridded over their initial range. This initial range is limited by means of the computational path and given boundary conditions. Consider again the example of Fig. 3. The requirements \(GBW > 1 MHz, CC > 1 \mu F\) and \(VGT2 = 0.2 V\) impose the following lower limit for the current \(I1\):

\[
I1 \geq GBW \times CC \times VGT1 = 0.63 \mu A \quad (6)
\]

Example

A CMOS folded-cascode OTA has been designed in 3 \(\mu m\) n-well towards minimum power consumption. The design specifications are compared to OPTIMAN data, SPICE simulations and measurement results in Table 1. Notice the close correspondence between all numbers. The CPU time required was less than 1 minute on a VAX750.

5. Analogous layout generation with AUTOLAC

After the sizing, the circuit is laid out by the AUTOLAC program. AUTOLAC generates the layout in four steps: it first divides the circuit in structural entities, which are then optimally placed and interconnected, after which the layout is compacted. In this way, a near optimal layout can be generated for every application, taking typical analogue constraints into account. AUTOLAC runs in front of the symbolic layout program CAMELEON [14].

<table>
<thead>
<tr>
<th>Specification</th>
<th>spec</th>
<th>OPTIMAN</th>
<th>SPICE</th>
<th>measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBW [MHz]</td>
<td>&gt; 200</td>
<td>223</td>
<td>234</td>
<td>220</td>
</tr>
<tr>
<td>power [mW]</td>
<td>&lt; 1</td>
<td>0.140</td>
<td>0.142</td>
<td>0.15</td>
</tr>
<tr>
<td>gain [dB]</td>
<td>&gt; 60</td>
<td>83</td>
<td>97</td>
<td>86</td>
</tr>
<tr>
<td>phase margin (°)</td>
<td>&gt; 60</td>
<td>89</td>
<td>92</td>
<td>83</td>
</tr>
<tr>
<td>slew rate (V/ns)</td>
<td>&gt; 0.13</td>
<td>0.38</td>
<td>0.38</td>
<td>0.22</td>
</tr>
<tr>
<td>noise (V/√Hz)</td>
<td>&lt; 20</td>
<td>16.1</td>
<td>17.6</td>
<td>17.0</td>
</tr>
<tr>
<td>output range (V)</td>
<td>&gt; 1</td>
<td>1.0</td>
<td>1.0</td>
<td>1.12</td>
</tr>
<tr>
<td>input range (V)</td>
<td>&gt; 1</td>
<td>1.0</td>
<td>1.09</td>
<td>1.14</td>
</tr>
<tr>
<td>offset (mV)</td>
<td>&lt; 0.5</td>
<td>0.14</td>
<td>0.1</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Table 1. Comparison of specifications, OPTIMAN data, SPICE simulations and measurement results for a minimum power design of the CMOS folded-cascode OTA.

The structural entities are functional groups of elements: resistors, capacitors, single transistors, two or more matching transistors... Matching elements are taken together in order not to over constrain the placement routine. All entities can be laid out in several ways. This information is combined into the bounding curve of the entity.

The placement itself is based on a slicing structure, which characterizes the relative placement of all (flexible) entities in the global layout. This slicing structure is then reorganized iteratively by the optimization routine, in order to find the structure which minimizes the goal function. This goal function consists of an area and an interconnection term. The optimization algorithm is simulated annealing [12].

The routing is performed by a conventional maze-router, which is currently being extended to incorporate typical analogue layout features, such as cross-talk minimization... The final compaction is performed by CAMELEON [14].

6. Integrating ISAAC, DONALD, OPTIMAN and AUTOLAC into an intelligent design system

Our present research activities are focused onto integrating the previously described programs (ISAAC, DONALD, OPTIMAN and AUTOLAC) into an intelligent design system by the introduction of artificial intelligence techniques. To this purpose, research is carried out in the following domains:

1. A topology selection program is being constructed. It is based on a library of frequently used circuit structures which are hierarchically organised on several levels. On each level, possible choices for decomposing a circuit into a lower level are rated and presented to the user by an expert system. The final decision is left to the user. The expert system is therefore to be considered as an intelligent advising design assistant. In the case of an opamp for example, first a decision is taken whether to use a one-stage or two-stage buffered/non-buffered opamp topology. Next, possible generic circuit topologies for a stage or a buffer are presented consisting of building blocks such as differential pairs, current mirrors, etc. This process continues until finally a topology at transistor level is assembled.

2. Since the user can always refuse a topology proposed by the expert system and provide his own, new circuit structures not yet included in the library, the CAD system should have capabilities to extract information about the functioning of the new circuit and to generate a new analytic model. To this purpose, building block recognition and qualitative simulation are essential tools. They are used to extract as much information as possible from the user provided circuit.

A building block recognizer [15] tries to extract the composing building blocks out of the circuit schematic by comparing the schematic with circuit
structures from the library. This provides more hierarchical information about the functioning of a circuit. The principle is illustrated in Fig. 4 for the CMOS Miller op amp of Fig. 2. In this schematic, two current sources, a differential pair, a current mirror, a one-transistor inverter and a compensation network are recognized. For those circuits, the appropriate circuit models can be retrieved from the library. Only the circuit structures which cannot be recognized, need to be analysed by ISAAC.

![Fig. 4. Building block recognition for the CMOS Miller opamp of Fig. 2.](image)

A qualitative simulator [16] provides the user with qualitative information about the functioning of a circuit, such as which combination of on/off states of transistors are feasible, which part of the circuit belongs to the biasing and which part to the signal path, how will an input signal be propagated through the circuit... Its main purpose is to provide more insight into the behaviour of the circuit.

3. When no satisfying result can be found after the optimization, some redesign strategy has to be considered. The user can choose between trading off design specifications or changing or reselecting a circuit topology. He can use the computational path constructed by the equation manipulation routine as a trace from which to identify the possible culprits, responsible for the unsuccessful optimization results. For example subcircuits that do not provide enough gain.

4. A successful analysis of an unknown user supplied circuit can be stored in the library and used in other design sessions later on. This ability to learn from previous design experiences prevents duplication of work and strongly extends the capabilities of the system.

7. Conclusions

An intelligent design methodology is presented for the design of functional analogue building blocks (opamps, comparators...). The method combines symbolic simulation, numerical optimization and knowledge-based techniques to allow an optimal circuit sizing and a fast inclusion of new topologies. The circuit optimization is based on analytic models, which characterize the circuit performance and which are generated by the symbolic simulator ISAAC. ISAAC generates symbolic expressions for all ac characteristics of both time-continuous and time-discrete circuits. It returns both exact and simplified analytic expressions.

The design equation manipulator DONALD converts the analytic model into a solution plan by selecting a set of independent design variables and by constructing a computational path for the calculation of all variables. This information is then passed to the optimization routine OPTIMAN, which sizes all circuit elements in order to satisfy all performance constraints, thereby optimizing a user defined design objective. The optimization algorithm is simulated annealing.

The design is then passed to the layout generator AUTOLAC. First, all functional groups of elements in the circuit schematic are recognized. These groups are then optimally placed, by a routine based on slicing structures and bounding curves. Finally, the functional groups are interconnected and the layout is compacted.

Research is still going on in the domains of building block recognition, qualitative simulation, topology selection, redesign strategies and learning capabilities in order to integrate the previous tools into a fully intelligent analogue design system.

Acknowledgements

The authors are grateful to Herman Walscharts and Carl Verdoneck for their contributions to the analogue design system.

References