Application-Independent Hierarchical Synthesis Methodology for Analogue Circuits

R. Wittmann¹, W. Schardein¹, B. Hosticka¹ M. Schanz¹, R. Vahrmann², and S. Kern²

¹ Fraunhofer Institut of Microelectronic Circuits and Systems, 47057 Duisburg, Germany
² TEMIC Telefunken microelectronic GmbH, 74025 Heilbronn, Germany

Abstract

This paper presents a new strategy for synthesis of analogue circuits in expert systems with an expert oriented approach to solving analogue design problems. Analogue circuits are hierarchically synthesised in a top-down manner while focusing on the main features of cells and subcells. This reduces drastically problems known in existing concepts, which involve implementation, expansion, and maintenance of expert knowledge, transparency of decisions, and application dependency.

1. Introduction

In contrast to a wide variety of available synthesis tools in digital circuit design corresponding tools for analogue circuits are rare. Since an analogue circuit design offers many more degrees of freedom and existing solutions for a given design problem vary very widely and are not following fixed rules, a lot of experience is necessary.

Available tools in the field of analogue design automation like OPASYN [1], IDAC [2], MIDAS [3], CADICS [4], OASE [5] are following a rule based approach. They have a database (or databases) with a fixed number of topologies for one application. A selection plan (based on ‘if-then’ rules) is used for comparing (input requirements and topology performances) and selecting a suited topology. This comparison, which is a suitable/unsuitable check, is only practicable for a low number of hierarchical levels (2-3) because of “explosion” of interval ranges for design parameters. So, for an application-independent analogue expert system with an open depth of hierarchy such a solution is not suitable. Further critical aspects are the missing transparency of the decision process for a user of the system since the selection plan is coded and the impossibility of the database expansion for additional topologies. In this case a new selection plan has to be worked out which considers a new topology.

During synthesis different kinds of constraints have to be considered which are expressed using equations, inequalities, and intervals and also mixed combinations of these. Up to now no analogue design tool makes use of interval arithmetic in which different kind of constraints can be handled easily. This introduces a lot of exception handling and often causes unnecessary reduction of solution space already in the beginning of synthesis without a chance of finding a solution at lower levels. Further problems are presented by dependencies between connected cells, macros and primitives. A progress in analogue design automation has occurred with appearance of the synthesis tool ARIADNE [6]. It uses a more global approach to analogue circuit synthesis. In ARIADNE a clear separation between design knowledge and general design procedures has been made. Critical point in this concept is, however, the high computation effort for handling the design constraints.

In this contribution a new concept for analogue design automation will be demonstrated in which no selection plan is needed. This yields the opportunity to create an application-independent analogue expert system. An effective interval algorithm procedure guarantees high efficiency of synthesis and reduces complexity of expert knowledge integration.

2. Characterisation of synthesis

The main feature of the synthesis strategy employed in the presented concept is the strict hierarchical treatment of all circuit structures as cells, macros, or primitives. Starting at the top level with input specifications for the design using a data sheet, the synthesis process runs step by step down the hierarchy adding at each level the information defined by design relationships derived from equations and inequalities. Such relationships are given by Kirchhoff’s laws, component rules and formulas, specifications describing performance, and other constraints. Since the objective of synthesis is to find an optimum solution for the required set of specifications given by the user, the expert system has to offer a
maximum of flexibility at any level and any state and allow various interactions with the designer. This enables "tailoring" within the expert system.

Fig. 1: Library independent circuit synthesis

For this purpose a new strategy (Fig. 1) has been developed in which each design level is split into 2 domains:

1. Rating domain with final selection
   This domain replaces the traditional application dependent decision tree. All cells of a library for a specific application are evaluated and rated according to their suitability for use in the current application with focus on the main features. This is not only a suitable/unsuitable check, but it involves a continuous relative grading with respect to the input specifications. Different rating methods like min-max, matching method and an interface for user-defined methods are available for the designer. Since the rating procedure requires a low computational effort it can handle a high number of cells and design parameters. The final cell selection can be done automatically or manually by the user after evaluation of the rating results so that the expert system can be used by experienced as well as inexperienced users. This means that all process steps are transparent on demand by the user and that all decisions are retraceable and verifiable. So the beginner can learn by using the system for design while the expert can improve the system decision processes if he does not find them to be optimal.

2. Computing of solution space using interval analysis
   After a cell has been selected constraint and interval propagation occurs. An additional check for the existence of a solution space considering all design parameters and cell dependencies is carried out. If no common solution is found backtracking is performed and the next cell in the priority list is taken. This list is based on the grading scale mentioned above. The chosen Tschernikow algorithm [7] can handle all kinds of linear equations and inequalities for input and gives the complete solution space as result. Worst-case intervals for each design parameter are extracted to enable the rating at the next lower design level. Because this strategy makes a clear separation between design knowledge and general design procedures, the stored expert knowledge is easy to handle and then is exchangeable and expandable. Technological requirements are introduced using spreadsheet.

3. Rating methods

   The min-max method which has been derived from fuzzy systems can be used to evaluate topologies and cells of same type. The min-max method is based on attaching quality functions to design parameters. The quality function represents a numerical measure for the difficulty how to fit a specific parameter of a cell block. The idea behind this rating procedure is to take in account all other parameters not considered so far by satisfying only minimum conditions of the input specifications. Another procedure that can be used is the matching method in which three characteristics of intervals describing one design parameter are considered for performance rating. The weighting of those characteristics can be adjusted manually by the user in order to have the chance to define his own consideration and interpretation of optimal performance. The characteristics of comparing an input interval coming from the highest design level, the user entry, and a characteristic interval describing the performance of a selected cell for one design parameter, are:
   - Input interval coverage of selected circuit
   - Comparison of input and characteristic interval width
   - Input interval centering

   Since these methods are used only for rating the performance of one single design parameter, a global rating for a cell which is an average rating of the parameter ratings has to be performed. Therefore, each design parameter has been attached a weighting factor (adjustable by the user) in order to emphasise desired features of the selected cell.

4. Tschernikow algorithm

   The Tschernikow algorithm represents an easy to use procedure for linear interval arithmetic [8]. It provides
processing of input intervals and yields new and more restricted intervals as output. This makes the method applicable for evaluation of hierarchical design constraints. The analysis can be performed at any synthesis level since the complete solution space resulting from a higher design level will serve along with the design constraints at the present level of hierarchy as input data to calculate an improved solution. The solution automatically adheres to all design constraints at higher levels. Starting with the top level (data sheet) and running top-down to the lowest level (schematics built up using primitives), this procedure can be used to obtain the maximum solution area possible at each level. With an additional target function or a design parameter which has to be maximised or minimised, the procedure is even able to calculate an optimal solution point. This qualifies the Tschernikow algorithm also for use in procedures like circuit optimisation and device sizing.

5. General plan of synthesis

In the following sections a detailed description of circuit synthesis is given. Fig. 2 illustrates the hierarchical structure of synthesis.

![Hierarchical design flow](image)

5.1 Datasheet entry

The interface between the expert system and the IC designer is formed by the datasheet. By entering the input specifications, which are either external parameters like amplification, power consumption, slew rate, etc. or internal cell parameters like bias current, open collector for output stage, etc., the designer defines his circuit requirements. Three different types of input specifications are supported, namely operating intervals, adjusting intervals, and "switches". If an input parameter is attached to an operating interval, the specified circuit has to satisfy all other specifications for every value of this interval (e.g. 8V<Vdd<10V). Interval adjustment attached to parameters defines the allowed range of values for these parameters. Additionally to intervals, parameters may be "switches". These "switches" are used e.g. for defining the used technology, setting special conditions like open collector output stage, or external load to ground or power supply line.

The designer is allowed to describe the data sheet in a more or less complete manner and may define weighting factors for an arbitrary number of input parameters. This information is processed by the expert system to generate a rating scale for evaluating competing requirements and specifications.

5.2 Check of plausibility

After completing the datasheet entry the assigned input parameters have to pass a plausibility check. Here the specified parameters are compared with the allowed parameter ranges. These intervals give the inherent parameter ranges of all possible topologies. Further, the plausibility check searches for contradictions, which can occur by entering dependent parameters.

5.3 Topology selection

The hierarchical synthesis starts with the selection of topology, which represents a coarse block structure of a circuit. Therefore, the topologies of the database have to be rated in order to find the most suited topology, which can fulfill the input specifications for the given input parameters. In order to find a global rating for a topology, first the performance of each input parameter has to be evaluated. These ratings are multiplied by weighting factors and then averaged to obtain a global topology rating. This characterises the performance of the topology that has to meet the input specifications.

Different approaches for performance rating are available. They can be selected automatically or manually by the IC designer. The result of topology synthesis is a sorted list of topologies according to rating results. Topologies, which cannot meet the input specifications are marked and their critical parameters are named. The advantage of this approach is that the best suited topology of the database
can be found, even if there is no topology in the database, which can meet the specifications. The topology synthesis ends up with the proposal for a suitable topology. The designer may accept this proposal or choose another topology after having considered the rating results.

5.4 Cell and macro selection

The topology consists of cells and macros which are connected to form the topology. Interactions between cells and macros are described using linear constraints. Those constraints form the internal topology description. Current and voltage equations given by Kirchhoff’s laws are used to characterise cell connection. Other equations express the relationships between cell parameters and topology parameters (translations). Each cell in the topology is a black box representation for a circuit class. In the database a circuit library exists for each class. Goal of cell selection is to find suitable circuits from the libraries, which can replace the black box representation in the topology, while considering the synthesis results of higher levels. If all cells at a certain level are specified, the next level of hierarchy is entered. At this new level again all cell selection for the subcells is carried out before entering the next level. Terminating the process of cell selection the final topology consists only out of primitives and macros, which are basic building blocks.

Again, before selecting a circuit from the library, every circuit performance has to be rated. This rating does not include the structure of topology. The principle for rating is the same as in the topology selection. In order to reduce computation time the input parameters are partitioned into level dependent and independent ones. The level independent parameters are used directly for elimination of cells from the valid cell table, which cannot fulfill the attached restrictions. For the remaining cells a method is employed to obtain a suitable cell combination for the given topology. For the first set of circuits the best rated circuit of each class is taken. External parameter descriptions of these cells and the internal parameter descriptions of the topology are evaluated using the Tschernikow algorithm in order to find a common solution space for all constraints. If this circuit is not realisable, the reason is analysed and considered in the next circuit set. If the Tschernikow algorithm detects a solution space, the next lower level of design can be entered to synthesise the subcells. At this lower level the corresponding selection steps are repeated. The result of the level given before by Tschernikow algorithm can be directly used for subcell selection. Cell synthesis is terminated when all cells and subcells have been specified using macros and primitives. If there is no solution at a certain level, the corresponding synthesis step has to be repeated by backtracking to the next higher level.

5.5 Optimisation and sizing

After defining all cells, macros, and primitives in the hierarchy optimisation and sizing is carried out. Under utilisation of the remaining ranges of circuit parameters yielded by Tschernikow algorithm the sizing parameters are fitted in order to fulfill the input specifications. Additionally to these linear constraints other constraints are included which describe interaction of sizing parameters or give a strategy for component sizing. These constraints can be linear, non-linear, or non-analytical. To compute the nonlinearities a gradient method like "steepest descend" can be used.

The last step is the simulation of the sized circuit using an analogue simulator. Here, the fine tuning of parameter values is carried out and also the non-analytical constraints can be included.

5.6 Backtracking

During synthesis the situation may occur, that no cells can be found that meet the input specifications or that sizing procedure fails. In this case the method of backtracking is used. The rating results of each level passed are stored. So, if no solution is found at a level, the synthesis goes back to the next higher level in which the lower level is represented by a single block. The circuit in this block has been already found and considered as best suited in combination with other blocks while regarding only constraints at present and higher levels. Now cell synthesis can return to that point of level synthesis at which this solution was found. The not realisable configuration is marked and synthesis continues until the next solution is found. If there is no suitable solution at this level at all then the present level has to be changed one level higher in the hierarchy in the same way as described before.

5.7 Database

A structured data format for storage of expert knowledge into the system has been developed. Each cell, macro and primitive gets its own spreadsheet for storing all relevant design knowledge. Every spreadsheet is subdivided into three areas: external and internal cell description and translations of design parameters to lower design levels. The external cell description is used for rating purposes. An identical parameter set allows a comparison of all cells in one circuit class. In the internal cell description a detailed view of cell topology is given. Here subcells are
"visible" and interconnection and relationships are described using equations and inequalities. This task is managed by Tschernikow algorithm.

First the designer enters the input specification into the datasource for a selected circuit family (Fig. 4). After successfully passing the plausibility check the topology selection is carried out. Assume that there are 3 topologies in the database with certain interval specifications. The parameter rating method used represents the degree of matching between input and characterisation intervals (Fig. 5).

First the designer enters the input specification into the datasheet for a selected circuit family (Fig. 4). After successfully passing the plausibility check the topology selection is carried out. Assume that there are 3 topologies in the database with certain interval specifications. The parameter rating method used represents the degree of matching between input and characterisation intervals (Fig. 5).

This data format allows a high degree of transparency. E.g. if the designer wants to verify an expert system design proposal and wishes to know in which constraints a specific design parameter is involved, he only has to observe the one row belonging to this parameter.

6. Implementation

The Falcon Framework environment of Mentor Graphics has been chosen for implementation of the synthesis strategy proposed in this contribution. With Falcon Framework the powerful spreadsheet tool DSS (Decision Support System) is available. All general design procedures (user interfaces, algorithms and rating methods) have been programmed using this tool. Also, for storage of the expert knowledge and the technology data the spreadsheet format has been chosen because it is easy to handle and a wide set of mathematical functions are available. For schematic entry Design Architect has been employed. Data exchange between Design Architect and DSS is possible using the common communication channel IPC.

7. Example

In this section a small simplified example for handling hierarchical design constraints is presented.

Now the internal constraints are available for synthesis.
In this example the constraints are:

\[ V_{cc} = V_1 + V_2 \quad V_{in} \leq V_{out} \quad V_{in} \geq 0.8 \quad V_{out} = V_2 \]

The restrictions for the design parameters depending on user requirements and topology dependency are evaluated using Tschernikow algorithm. The solution of this design problem has been found as:

\[
\begin{align*}
\begin{bmatrix} V_{in} \\ V_{out} \\ V_1 \end{bmatrix} &= \begin{bmatrix} 10 \\ 10 \\ 0 \end{bmatrix} p_1 + \begin{bmatrix} 1 \\ 0 \\ 9 \end{bmatrix} p_2 + \begin{bmatrix} 0.8 \\ 10 \\ 0 \end{bmatrix} p_3 + \begin{bmatrix} 0.8 \\ 1 \\ 9 \end{bmatrix} p_4 : \sum p = 1
\end{align*}
\]

This equation describes the whole solution space. The computed intervals for the parameters, which can be read directly from the equation, can be used additionally for the next cell selection. The new intervals are:

\[ 0.8 \leq V_{in} \leq 10 \quad 1 \leq V_{out} \leq 10 \quad 0 \leq V_1 \leq 9 \]

Now cell selection for the two cells C1 and C2 is performed. The example continues at the point when cell selection has been completed and C1 & C2 are specified (Fig 4b). At this level additional design constraints have to be solved:

\[ \text{cell1: } V_1 = V_{21} + V_{22} \]

\[ \text{cell2: } V_{out} \geq V_{23} \quad V_{in} \geq 2 \cdot V_{23} \quad V_{23} \geq 0.8 \]

Together with the result of first computing these equalities are evaluated using Tschernikow. The solution space of this level results in obtaining the following expression:

\[
\begin{bmatrix} V_{in} \\ V_{out} \\ V_{1} \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ 0 \end{bmatrix} q_1 + \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} q_2 + \begin{bmatrix} 16 \\ 10 \\ 8.4 \end{bmatrix} q_3 + \begin{bmatrix} 16 \\ 16 \\ 8.4 \end{bmatrix} q_4 + \begin{bmatrix} 16 \\ 16 \\ 8.2 \end{bmatrix} q_5 + \begin{bmatrix} 16 \\ 16 \\ 8.4 \end{bmatrix} q_6 \quad \text{with: } \sum q_i = 1
\]

The new resulting worst-case interval ranges are:

\[ 1.6 \leq V_{in} \leq 10 \quad 0 \leq V_{1} \leq 8.4 \quad 0 \leq V_{23} \leq 8.4 \]

\[ 1.6 \leq V_{out} \leq 10 \quad 0 \leq V_{23} \leq 8.4 \quad 0.8 \leq V_{23} \leq 5 \]

Again these interval ranges can be used additionally for next lower level of design. If the bottom level has been reached, we know that optimisation and sizing strategies fit the design parameters in the allowed ranges. Now, the interactions between the design parameters, which are also included in the computed solution space, have to be considered. So, if one parameter has been fitted, the reduced interval ranges of the other design parameters have to be recomputed before fitting the next parameter.

8. Summary

A general concept for the synthesis of analogue cells has been presented. It is based on a strictly hierarchical approach, a step by step qualification of topologies, cells, macros and primitives, and a transparent knowledge base for optimal handling, learning, and maintenance. The use of a special set of rating procedures avoids the use of programming procedures and non-transparent application-dependent decision trees. Rating results which are based on the main features of all cells of same class stored in the database, assist the analogue designer in finding a preliminary decision when choosing a suited topology or cell. In a further step a powerful interval analysis algorithm checks the solution space for all design parameters and enables propagation of parameter interval restrictions and dependencies of interconnected cells down the hierarchy. In addition, use of backtracking eliminates unnecessary computation effort when no common solution space has been found using Tschernikow algorithm.

9. Acknowledgements

This work has been supported by JESSI AC12 and TEMIC Telefunken microelectronic, Heilbronn, Germany.

10. References


