OASYS: A Framework for Analog Circuit Synthesis

Ramesh Harjani

Mentor Graphics Corporation*
1960 Zanker Road
San Jose, California 95112

Abstract

We describe a hierarchical structure for a knowledge-based analog circuit synthesis tool. Analog circuit topologies are represented as a hierarchy of abstract functional blocks each with associated design knowledge. We also describe mechanisms to select from among alternate design styles, and to translate performance specifications from one level in the hierarchy to the next lower. A prototype implementation, OASYS, synthesizes sized transistor schematics for CMOS operational amplifiers and comparators from a set of performance specifications and process parameters. We describe the role such a synthesis system can play in exploring the space of designable circuits. And finally, we briefly describe other related research in analog synthesis at Carnegie Mellon, including OASYS-VM, which facilitates the addition of new design topologies to the framework, and ANAGRAM, the counterpart to OASYS, which performs the circuit schematic to physical layout phase of analog circuit design.

1. Introduction

As feature sizes shrink even further, it is not surprising to find complete systems on a single chip increasing the proportion of ASIC ICs that have analog circuit designs in them. Unfortunately, analog CAD synthesis tools are primitive in comparison to their digital counterparts. Until recently, the analog section of mixed ASIC designs were designed manually by experts while the digital sections were mostly designed automatically. It is in these analog sections of the design that adversely affect the turn-around time of such mixed analog/digital ASIC designs.

Even though hierarchy and other structured abstractions have played major roles in helping design digital circuits with hundreds of thousands of components, such abstractions have only recently been introduced into analog circuit design [6]. Early analog CAD tools have been influenced by digital design methodologies, such as standard cell libraries and module generators. But these techniques have been severely limited in their range of achievable performance and provide only spot solutions that rapidly become obsolete with technology changes.

In this paper we provide an overview of a more generalizable solution that attacks a specific portion of the analog circuit synthesis problem. During a manual design a designer spends a sizable portion of his time developing an initial design which he then follows by a tuning and layout phase [8]. The aim of this initial design phase is to generate a sized circuit schematic from a set of performance specifications. In the paper we shall concentrate primarily on this initial design phase ending with a short discussion on layout phase as well as some discussion on adding design knowledge to the system in future.

2. Background

To fully appreciate the analog synthesis problem it is essential to see how it both differs from and is similar to the digital synthesis task. Before describing the details of analog design per se, we shall examine synthesis methodologies that have been used effectively in the digital domain. In digital design extremely large designs are designed by breaking up the task into smaller and smaller sub-tasks, divide and conquer. This division of the the design flows down a well accepted hierarchy from behavioral level to transistor level designs. In fact much of the progress in digital synthesis can be attributed to this concept of hierarchy. Hierarchy helps to hide the lower-level details and also help focus attention more tractable sub-tasks. Even though hierarchy does exist in analog it tends to be looser and less well accepted. For example, analog systems like analog to digital (A/D) converters are not designed at the transistor level right from the start. The design is first broken up into functional blocks like operational amplifiers and comparators. These block designs in turn are broken up into current mirrors and differential pair which in turn finally become transistors.

In digital design device level process parameters and layout constraints are isolated from the higher levels of design. One worries in terms of the maximum clock-rate and the total number of devices that can be accommodated on a chip at the highest level of digital design. In analog, nuances of the process can quite markedly affect the choice of the methodology selected even at the highest level of design. For example, a implementation methodology that requires capacitors to be matched to a particular precision may not be feasible if the such components are not available in the process. Or there could be a limit set on the resolution of a delta-sigma A/D converter that is governed by the thermal noise of the individual transistors contained within some lower level sub-block.

Given the differences between the two domains, it is clear that we cannot directly import design methodologies from the digital domain into the analog domain without significant changes. But we can distill some salient ideas that have worked in digital to help guide approaches for analog circuit synthesis.

2.1. Previous approaches to analog design

Analog circuit synthesis is characterized by a number of steps from the highest level to the lowest as shown in Figure 1. First, very abstract behavior, usually characterized by a set of input/output waveforms, is translated to a set of functional specifications. For example, we are given a set of sinusoidal curves from which we deduce that an oscillator is required. We call this step abstract synthesis. These functional specs are then translated to generate a sized circuit schematic. We call this step circuit synthesis. Circuit synthesis is
usually then followed by an optimization phase where the devices sizes are altered to generate a more optimal design in terms of yield and performance. And finally, from these more optimal designs we generate IC masks.

Previous approaches have attempted to attack both different steps of the analog design process as well as have applied different approaches at the various problems. In particular they are characterized as being either bottom-up approaches or top-down approaches. In bottom-up approaches, the design is largely determined by layout constraints while for top-down approaches typically strive to translate behavior to structure to physical layout.

![Figure 1 Analog Synthesis Sequence](image)

In bottom-up approaches the constraints on the design arise from restricting or fixing some aspect of a lower level of design. Examples of bottom-up approaches are transistor arrays [13], analog library cells [10], and fixed topology module generators [9]. All three approaches primarily help ease the path to layout, but do not necessarily help a designer generate a circuit schematic from a set of performance specifications. These approaches are also limited by their limited range of input performance specifications and are really only good for low to medium performance designs.

In top-down approaches design constraints arise from the set of higher-level specifications that are used to guide design. These topdown approaches can broadly be classified as optimization-based and as knowledge-based. Optimization-based approaches model the overall synthesis task as a large set of tightly coupled non-linear equations, and rely primarily on numerical techniques (e.g., gradient-based search, simulated annealing, etc) for a solution [11]. Knowledge-based approaches fall into two broad classes. One class, primarily those that have attempted high-level abstract synthesis, use analog design as a testbed for AI techniques [14]. Another class of knowledge-based approaches are less esoteric and attempt to translate a set of functional specifications into sized circuit schematics using detailed analog domain knowledge [1,2,3,4]. Only the optimization-based and the second class of knowledge-based designs have had any reasonable success synthesizing circuits.


In the next section we shall discuss the components of the OASYS framework. We shall then detail some implementations for these components.

3. A Framework for analog circuit synthesis

OASYS uses as input a set of performance specifications and device level process specifications and generates as output sized circuit level schematics. The system contains a framework and detailed analog design knowledge that fits into this framework. The framework itself exists independent of the design knowledge. The design knowledge that is contained in the system is specific to each class and design style of circuit. There are three basic components that form the basis for the framework. We describe these in details and explore some mechanisms to implement these components.

3.1. Components of the Framework

There are three primary components of the framework: hierarchy, design style selection and translation. Each analog design is represented as a hierarchy of functional blocks. And design proceeds down this hierarchy until it bottoms out when it reaches the level of transistors. This hierarchy help decompose the synthesis task into smaller sub-tasks and also makes it possible to reach a number of topologies from a compact hierarchical representation.

OASYS relies on the existence of a number of mature design styles at each level of the hierarchy. At each level of the hierarchy we first select from among the various candidate styles at that level. Having once selected a design style we then perform translation. Each of these design styles is an interconnection of abstract sub-blocks. For example, in figure 2 we see a hierarchical representations for a 1-Stage operational amplifier design style. At this level the design style for the operational amplifier consists of abstracts sub-blocks such as current mirrors and differential pairs. Other example design styles for the operational amplifier would be the Miller-compensated 2-Stage and the folded cascode. The different design styles all provide the same functional behavior but offer different performance tradeoffs and usually has different topological interconnections as well.

![Figure 2 1-Stage OTA Operational Amplifier](image)

Having once selected a design style we now need to translate these higher level specifications into specs at a lower level of the hierarchy. During translation phase we wish to determine how each sub-block needs to behave such that the whole block is able to meet its specifications. For example in the case of the 1-stage operational amplifier in figure 2 we have to translate the gain, the slew rate and the unity gain frequency into the current, the transconductance and output impedances for the the current mirrors and differential pair.

Design style selection and translation are the OASYS counterparts to topology design and sizing in manual design. Selecting among mature design styles avoids the need to dynamically inter-connect primitive devices. Instead we are able to choose from among these mature and hierarchically-defined styles. Performing translation steps at all levels of the hierarchy distributes sizing decisions over the hierarchy. The decisions that now need to be made are a sequence of smaller and more local choices. The principal advantage of viewing design as a sequence of design style selection and translation steps is that we are able to exploit expertise specific to each individual style in the hierarchy.

Let us consider the design sequence that might occur during the synthesis of a op amp. We are initially provided with a set of performance specifications for an op amp, like the gain, the slew rate, the bandwidth and maybe the input referred noise voltage. Using these performance specifications we need to select a candidate op design style. Let us assume that we select the 1-stage topology shown in figure 2. We now need to translate these op amp performance specifications into performance specifications for the sub-blocks contained within the op amp design style. Translation at this level of the hierarchy generates performance specifications for the three current mirrors and the differential pair contained in the 1-stage design style. The process is repeated at a lower level of the hierarchy.
as well. Specifications that were generated for the current mirror are used to select a candidate current mirror design style. Examples of current mirror design styles are shown in figure 3. Having once selected a current mirror design style we need to translate the performance specifications for the current mirror into specifications for sub-blocks within the current mirror which at this level are transistors.

![Current Mirror Design Styles](image)

Figure 3 Current Mirror Design Styles

Unfortunately, design does not always flow top-down. It is possible that a sub-block may not be able to meet its input specifications. We attempt to recover from such failure as locally as possible and only if a more local fix fails do we attempt a more global fix. For example, in the case of the op amp example just described, let us assume the a simple current mirror was selected and during the translation process we discovered that the device sizes were absurdly large. We first attempt to alter the translation step for the simple current mirror. If this fails we exhaustively try all the alternate current mirror design styles. If all the current mirror design styles fail to meet specification then only do we attempt a more global fix by returning to the op amp so that we may relax some of the requirements on the current mirror. This failure recovery process continues until we have exhausted all possibilities, attempting a local fix as much as possible and only requiring a more global fix if a more local fix does not solve the problem. By its very nature such design is iterative and the final designs that are produced are the result of negotiating constraints among sub-blocks at the same level of hierarchy and between levels of the hierarchy. In general, more aggressive designs require larger numbers of iterations up and down the hierarchy.

OASYS differs from other closely related research [2,3,4,7] in a number of ways. Primarily, the notion of an underlying framework that exists independent of the design knowledge is quite distinct here. A framework provides a uniform and structured view of the synthesis process. Design can now flow as a sequence of selection and translation steps through all levels of the hierarchy. The framework also helps to compartmentalize design knowledge and makes it reusable in that it need only be acquired once for each design style and can be reused a number of times. OASYS also differs in that it relies on hierarchy as the central philosophy of the design process. Most previous efforts have relied on a library of fixed, flat transistor level schematics.

### 3.2 Implementation mechanisms

A number of questions about the framework remain to be answered. In particular, how do we implement the various components? The structure of the hierarchy is the sum total of how each design style is structured. We regard each design style as a fixed template of abstract sub-blocks. It is important to note that these sub-blocks are not transistor but abstract sub-blocks. They become transistors only at the very lowest level of the hierarchy.

Selection is implemented as sequence of structural discrimination followed by either rank ordering or generate and test. Quite often it is relatively simple to infer that a design style may not be able to meet its input specifications. The design style could be mismatched structurally or we could use a simple algebraic test to decide whether it can meet its input specifications. Since these tests are relatively simple it makes sense to perform structural discrimination before attempting any other selection mechanism. The other selection mechanism that is simple to implement, but could potentially take a long time, is to exhaustively attempt all the design styles out and then finally decide on the best candidate. We find that this mechanism works well at the lower levels of the hierarchy, where the design space is still small, but we feel that it may become a problem at higher levels of the hierarchy. In rank ordering we order the design style and try the one that is most likely to succeed first. Some simple techniques for rank ordering and are mentioned in [8].

In OASYS translation is implemented as a very simple planning mechanism. Each design style has an attached design plan, that is used to refine the block level specifications into sub-block specifications. This planning process involves the decomposing of the main design goal into a number of sub-goals, the ordering of these sub-goals, proposing individual actions for each sub-goal and changing the plan if a goal is not met. We find that for each design style, good designers have a good rough-plan-of-attack to complete the design. Since these design plans exist as domain knowledge it is quite wasteful to attempt to reinvent them on the fly. We also feel that these design plans are difficult to rediscover on the fly because they have tightly coupled goals with a fine grained numerical flavor.

### 4. Synthesis results: Performance evaluation

To test out the various ideas of the framework mentioned we developed a prototype analog synthesis system (OASYS) capable of synthesizing CMOS op-amps and comparators. A system like OASYS is capable of synthesizing a circuit to meet a set of input specifications. Or it is also capable of performing design space exploration. In design space exploration, we perform many individual designs to explore possible trade-offs among performance or process specifications. In this section we first present conventional synthesis results and then describe some exploratory applications of OASYS.

#### 4.1 Conventional synthesis results

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Input Spec</th>
<th>SPICE Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Overdrive</td>
<td>mV</td>
<td>≤ 2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>Response Time</td>
<td>ns</td>
<td>≤ 12.5</td>
<td>12.05</td>
</tr>
<tr>
<td>Estimated Power</td>
<td>mW</td>
<td>≤ 5.0</td>
<td>4.38</td>
</tr>
<tr>
<td>Load Capacitance</td>
<td>pF</td>
<td>10.0</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 Comparator Specifications and SPICE results

![Comparator Schematic](image)

Figure 4 Comparator Schematic

OASYS is capable of designing 1-stage OTA and 2-stage Miller-compensated CMOS operational amplifiers and CMOS comparators. Designs produced by OASYS have been tested extensively by simulation and by actual measurements from fabricated ICs [6,7,8]. Here we describe a CMOS comparator design generated by OASYS. The total design time to generate this relatively aggressive performance specification is about 3 seconds on a Vaxstation2110GFX. The specification and spice results are shown in Table 1 and we see that we meet all input performance specifications.
4.2. Design space exploration

As we noted from the execution time of the previous OASYS result, single designs are generated rapidly. Hence it becomes possible to explore design alternatives by synthesizing a large number of circuits and then examining the characteristics of the whole bunch of circuits.

In the first example we wish to explore the effects of varying the response time specification on the area and power for a comparator. In this experiment we fix all process and performance specification except the response time. The results of varying the response time is shown in figure 5. As expected, smaller response times require both increased area and power. In this fashion OASYS is capable of providing a quantitative view of the tradeoffs involved during design. Such design space explorations can easily be performed in more than one dimension. For example, we could vary both the response time and the load capacitance and examine the effects on area and noise.

There is another closely related design space exploration called threshold hunting. In threshold hunting we wish to determine the surface that separates feasible and infeasible design achievable by OASYS. For example in figure 5 we fix all input specifications for an op amp except the gain and unity gain frequency. Then for each value of unity gain frequency we find the maximum gain that OASYS is able to meet. The line that separates the feasible and infeasible region is somewhere between the success and failure plots in figure 5.

![Figure 5 One Dimensional Exploration](image)

It is important that analog designers be able to easily add design knowledge to a synthesis system. OASYS-VM is currently being developed to ease this process of adding design knowledge. OASYS-VM uses a high-level block structured language description (OADL) to facilitate the inclusion of detailed circuit design knowledge in the form of design plans and design styles into the OASYS framework.

6. Conclusions

We have presented three related design tools OASYS, ANAGRAM and OASYS-VM all of which help automate the translation of a set of performance specifications to mask geometry. Experience with the OASYS system is very encouraging and suggests that a hierarchy of selection/translation steps is a workable approach to analog circuit synthesis. Even though OASYS requires that the design proceed hierarchically results suggest that it is still feasible to attack relatively high-performance design problems using this methodology. We have also defined the role of design space exploration provided by such a synthesis tool.

7. Acknowledgments

The author would like to thank Rob A. Rutenbar, L. Richard Carley, David J. Garrod and Emiliano S. Ochotta for their continued support and guidance. This work was supported in part by a grant from the Gould Foundation, the Semiconductor Research Corporation and by NSF under grants ENG-8451496 and MIP-8647369.

8. References


