Design Synthesis
of
Monolithic CMOS Operational Amplifiers

By

Han Young Koh

B.S. (Seoul National University) 1980
M.S. (Georgia Institute of Technology) 1984

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ENGINEERING
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA at BERKELEY

Approved: Carlo H. Séquin ..... April 17, 1989
Chair ..... Date
Paul R. Gray ..... April 17, 1989
Charles J. Stone ..... April 17, 1989

Design Synthesis of Monolithic CMOS Operational Amplifiers

Han Young Koh

Ph.D.

Department of Electrical Engineering
and Computer Sciences

Abstract

This research effort explores efficient methods for the design synthesis of monolithic CMOS operational amplifiers. A synthesis system has been developed. It, called OPASYN, takes as inputs system level specifications, fabrication-dependent technology parameters, and geometric layout rules. Based on the general domain of the specifications, the program first selects an appropriate circuit topology from a database using heuristic pruning of the decision tree. Optimal values for the set of design parameters of the chosen circuit are then determined so as to meet the design objectives. Analytic models of several widely applicable operational amplifier circuit topologies have been developed to eliminate expensive circuit simulation and sensitivity analysis in the inner loop of the optimization step. Subsequently, design-rule-correct mask geometries are constructed using a macro cell layout style. Primitive circuit elements such as transistors, transistor pairs, and capacitors are produced by parameterized leaf cell generators and assembled according to circuit-dependent slicing trees that guarantee sound arrangements of the individual components. The synthesis process is fast enough for the program to be interactively used at the system design level by system designers who are inexperienced in op amp design.

Carlo H. Séquin
Chairman of Committee

ACKNOWLEDGMENTS

I would like to express my heart-felt thanks to Professor Carlo H. Séquin and Professor Paul R. Gray for their excellent guidance and continuous encouragement. Professor David A. Hodges, my major field advisor, deserves much gratitude for his warm consideration and consistent advice ever since I started my graduate study. I also appreciate Professor A. Richard Newton, who was one of my qualifying examination committee members, for his interest and support on this work. Without their help, this work could not have been successful.

I am also grateful to the Berkeley CAD group as well as many other graduate students for their contributions. Deog-Kyoon Jeong has provided a great deal of constructive feedback at the beginning of this work. Special thanks are due to Jeff Burns who helped me to understand the operation of his spacing program, SPARCS. Rick Spickelmier, David Harrison, and Peter Moore taught me how to use the CAD framework developed by the Berkeley CAD group. I would also like to thank Hyunchul Shin for providing his router, MIGHTY. I appreciate Beomsup Kim, Yuh-Min Lin, Gani Jusuf, Glenn Adams, and Ping-San Tzeng for their various contributions.

Finally, I would like to express my heart-felt thanks to my parents and my wife who have supported me with their love and patience throughout my Ph.D program.

DEDICATION

This dissertation is dedicated to my parents, **Hye Sook Chun** and **Bum Seo Koh**, and my beloved wife, **Hyeri Koh**. They have made a big sacrifice to make it possible for me to pursue my graduate studies and they have continuously encouraged me with their wonderful love. Thus this ph.D. thesis is the result of all our joint efforts.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The pervasive trend in recent years towards the integration of whole systems into a single chip VLSIC's requires that both digital functional units and dedicated analog interface subsystems (such as A/D converters and filters) be implemented onto the same chip. Many digital parts of such chips can nowadays be synthesized rapidly and reliably using CAD tools developed for semicustom design methods such as gate arrays, standard cells, and macro cells. On the other hand, analog interface subsystems still need to be fully handcrafted by a specialist. Therefore, the design time and cost associated with dedicated analog interface components often constitute a bottleneck in semicustom design of mixed analog/digital systems such as voiceband data modems and high speed data transceivers.[1]

For semicustom design of analog circuits, building blocks may be stored in the form of parameterized generators or as entries in macro cell or standard cell libraries.[2,3,4,5] As in the digital domain, the usage of libraries of predefined analog building blocks can shorten the design period significantly. However, in the analog domain it is difficult to configure a rich enough set of library cells for the wide spectrum of possible applications. Performance specifications for analog building blocks are much more diverse and complicated than those for digital blocks. For a digital inverter, delay and area are the major performance specifications. On the other hand, specifications for an operational amplifier (op amp) may include power dissipation, small signal dc gain, bandwidth, phase margin, slew rate, settling time, output voltage swing, 1/f noise, area, and so forth. Some of these specifications, such as the small signal dc gain of a CMOS op amp may vary over a wide range, depending on applications and system architectures being used. Furthermore, analog library entries

become obsolete even more quickly when technology or design rules change. A performance characteristic such as 1/f noise of a CMOS op amp degrades as technology scales down while that for small signal dc gain improves. As a result, generators that operate at the circuit or netlist level are more flexible and can be useful over a much larger domain of applications than fixed library cells. With such generators, design parameters (such as device sizes or bias currents) of a building block can be individually optimized for a particular application and for the particular technology to be used. The physical layout geometry of the block with its optimal design parameter values is then also produced by the generator.

The most often used analog building block is the operational amplifier. It is at the heart of many interface circuits, in particular, A/D and D/A converters, and filters. An efficient design of optimal op amps is thus a corner-stone of a design environment for many applications. As discussed above, op amp specifications for different applications vary so widely that it is impractical to store op amps as library cells for all applications. Using one of a few 'standard' library cells that are poorly matched to a particular application is unacceptable; if the performance of the op amp falls below a certain 'threshold', the quality of the overall system will suffer.

Designing a good op amp is a rather complicated multi-facet task.[6,7] An op amp topology appropriate for the given specifications must be chosen. Then its design parameters such as device sizes and bias currents must be adjusted under multiple design objectives and constraints. The many degrees of freedom in parameter space as well as the need for repeated circuit performance evaluation make this a lengthy and tedious process. This circuit optimization can be carried out by a computer but the process is still expensive and requires a skilled operator. The optimized circuit needs then to be transformed into mask level geometries. The layout process is critical for good performance since op amps like other analog circuits are sensitive to parasitic elements, to process/thermal gradients, and to noise. In CMOS technology, device sizes in op amps vary over a large range of values.

Some devices may have many alternative shapes, and terminal configurations. Thus, most of the classical layout generation methods developed for digital IC's are insufficient for CMOS op amps.[8,9,10,11,12] As a result, work on automatic layout of monolithic switched capacitor filters uses op amps as fixed cells from a library.[13,14,15] For many applications, this approach does not give enough flexibility and performance.

Most of the work published on the automated synthesis of op amps concern the schematic generation and parametric optimization of op amp *circuits*. The most common approaches are based on optimization, an expert systems approach, or a combination of both. *The optimization-based approach* uses various optimization algorithms combined with circuit simulation techniques to produce general purpose parametric optimization tools.[16,17,18] Since no a priori circuit design knowledge is necessary, this approach is applicable to a broad range of analog circuits. High degrees of optimality can be obtained through the use of robust and elaborate optimization algorithms. However, optimization tools based on this approach are not tailored to be used by novice designers. For instance, a system designer who wants to design an op amp for his/her A/D converter, has to provide the system with detailed design knowledge on op amps such as how to calculate slew rate from a particular node voltage. Besides, the designer must determine from manual or other design techniques a good initial guess for each design parameter. Without such a good starting point, an optimization run can converge very slowly or converge to a local minimum whose performance is significantly worse than the circuit's best capabilities.[17] This approach is normally very costly in CPU time because repeated circuit simulation is performed in the inner loop of the optimization step.

In the knowledge-based approach, domain specific knowledge has been used to produce synthesis tools which emulate expert designer's design processes. [19,20,21,22,23] The domain specific design knowledge integrated into these systems permits novice designers to specify simply their design requirements. Fine tuning of the system for a specific application area

can also be achieved easily. But a large complicated search space occurring from design of op amp circuits and a lack of an efficient performance estimation method make it difficult and inefficient to perform design optimization using this approach. In addition, a priori design knowledge has to be integrated into the system whenever a new type of building block is added. A general synthesis system that starts from first principles of circuit design such as Kirchoff's current and voltage laws does not yet exist!

Alternatively, advantages of both approaches can be combined to devise synthesis tools whose optimization is done in an algorithmic way but substitutes expensive circuit simulation with algebraic evaluation of analytic design equations acquired from domain specific design knowledge.[24] This is the approach taken by OPASYN described in this paper.

Automatic layout generation for CMOS op amps has been less widely explored.[25, 23, 26] The approaches rely either on dedicated slicing trees[25, 23] or use a general-purpose layout program based on simulated annealing and constraint-based routing.[26]

This thesis describes a design synthesis tool for monolithic CMOS op amps (OPASYN), that starts from a set of performance specifications (open loop gain, bandwidth, slew rate, etc) and produces design-rule-correct mask geometries in a macro cell layout style.

In Chapter 2 the performance objectives and generic architectures for op amps are introduced. The conventional synthesis procedure for op amps is examined and automatic synthesis systems for op amps are reviewed. The section is intended to provide background material necessary for understanding the OPASYN compilation method.

Chapter 3 presents an overview of an analog silicon compiler for CMOS op amps (OPASYN) developed using the described method. The framework of the program is delineated, and the function of each program module is explained. The OPASYN's interface to

the Berkeley CAD environment is also discussed.

In Chapter 4 a heuristic circuit selection method is explained. The method is used to select a promising op amp circuit topology based on the user specifications from the alternatives provided in the database. This is contrasted with other methods, such as creation of a new architecture in hierarchical manner, and incremental modification of topologies, and the difficulties and impracticalities of these approaches are discussed. The implementation of the method as a rule-based system is presented.

In Chapter 5 the problem of parametric optimization in op amp synthesis is addressed and a new optimization method using 'analytic circuit models' of op amps is discussed in detail. Analytic circuit models are defined and their acquisition process is presented. The shape of the parameter space is examined for the op amps used in OPASYN and the optimization strategy developed for this parameter space is described. Other optimization methods are briefly reviewed and discussed.

Chapter 6 is concerned with layout generation. Layout requirements for analog circuits and the needs for analog cell generators (instead of analog cell libraries) are discussed. A new floorplanning method based on parameterized cell generators and 'circuit dependent slicing trees' (also known as 'decomposition tree') is first described. It is then explained how Stockmeyer's algorithm can be used to determine optimal shapes for the various modules according to the user's shape constraint. Routing and layout spacing strategies are also described.

In Chapter 7 the implementation of OPASYN and experimental results obtained with the program are discussed. A synthesis example, that starts from a set of performance specifications and completes with an op amp layout in mask geometries, is provided to illustrate the design data flow in the system.

In Chapter 8 the problem of incorporating OPASYN as a module generator into larger

synthesis systems for switched capacitor filters and A/D converters is addressed. Subsequently, macro modeling for op amps is discussed.

Conclusions based on this research are presented in Chapter 9.

CHAPTER 2

SYNTHESIS OF CMOS OP AMPS

2.1 Background

An ideal operational amplifier (op amp) is a differential input, single-ended output amplifier with infinite gain, infinite input resistance, zero output resistance, and no internal delay. A conceptual schematic diagram is shown in Fig. 2.1. While actual op amps do not have these ideal characteristics, their performance is usually sufficiently good that in most applications the circuit behavior closely approximates that of an ideal op amp. With the advent of monolithic-circuit technology, the cost of monolithic op amps has been strongly reduced, and as a result op amps are widely used in the design of all types of analog systems. They are key elements of most analog subsystems, particularly in switched capacitor filters, and the performance of many systems is strongly influenced by the performance of the op amps used.

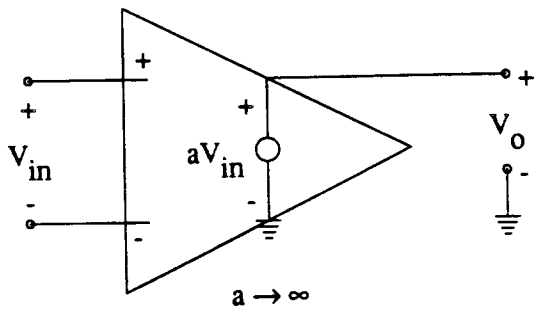


Figure 2.1 Ideal operational amplifier.

Traditionally, monolithic op amps have been fully hand-crafted by expert designers, either as components in high-volume, cost-sensitive interface subsystems, or as general-

purpose SSI building block elements. However, the large design time and cost associated with full custom design is hardly justified if volume of production is low or if time to market requirements are more important than component costs as is the case in application-specific and customer-specific IC design problems. Suitable synthesis tools must be developed allowing a system engineer who is not an expert analog designer to successfully carry out design, layout, and verification for complex op amps matched to a particular application.

Op amps have been implemented in many different technologies: bipolar, NMOS, CMOS, and BiMOS technologies, depending on such variables as current drive required, requirement for low dc offset voltages, requirement for low power dissipation, requirement for operation at high supply voltages, etc. Currently, general-purpose SSI op amps are mostly implemented in bipolar technology because of its capability of providing high current drive, high gain, and low dc offset voltage. When used as components in dedicated analog interface subsystems such as A/D converters and SC-filters, op amps are implemented exclusively in CMOS technology due to high complexity of on-chip logic functions required in such subsystems. When implemented in BiMOS technology, op amps can have advantages available in both bipolar and CMOS technologies: high input impedance, low dc offset voltages, high current drive capability, etc. However, the larger number of processing steps required for BiMOS technology limits its popularity. In this chapter, only work on CMOS op amps is reviewed because of its prevailing usage in semicustom IC design.

In Section 2.2, the important performance requirements and objectives for op amps within monolithic analog subsystems are discussed. In Section 2.3, generic op amp architectures are introduced and their performance ranges are examined. In Section 2.4, traditional ways of designing op amps are discussed. Finally, automatic synthesis systems for op amps are reviewed in Section 2.5.

2.2 Performance Objectives

The performance requirements for op amps to be used within a monolithic analog subsystem are often quite different from those of traditional stand-alone bipolar amplifiers. Perhaps the most important difference concerns output drive capability. For the amplifiers in a monolithic system, the load that each of the amplifiers has to drive is well defined, and is often purely capacitive with values of a few picofarads. In contrast, stand-alone general-purpose op amps must be designed to achieve a certain level of performance independent of loading for capacitive loads up to several hundred picofarads and resistive loads down to 2 kΩ or less.[7] Within a monolithic analog subsystem, only a few of the amplifiers must drive a signal off chip where the capacitive and resistive loads are significant and variable. In this thesis, these amplifiers are termed 'output buffers', and the amplifiers whose outputs do not go off chip are termed 'internal amplifiers', following Gray and Meyer.[7] From this point on, internal operational amplifiers will be simply called 'op amps'.

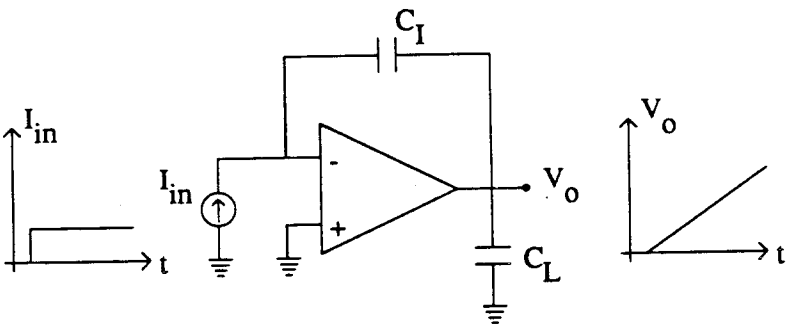


Figure 2.2 Typical application of an internal MOS operational amplifier, a switched capacitor integrator.

A typical application of an op amp, a switched capacitor integrator, is illustrated in Fig. 2.2. The basic function of the op amp is to produce an updated value of the output in response to a switching event at the input in which the sampling capacitor C_I is charged from the source and discharged into the summing node. The output of the integrator must assume

the new updated value within the required accuracy, typically on the order of 0.1 percent of the final value, within one clock period (typically on the order of 1 μsec for voiceband filters). It should also be capable of swinging fully between the upper and lower limits specified by the filter designer: The dynamic range of the filter (defined as the ratio of the largest input signal to smallest input signal discernible above the noise level) is strongly dependent on these limits because for multiple stage filters, the input of the present stage comes from the output of the previous one. On the other hand, the smallest input signal treatable by the integrator is limited by noise coming from the op amp, the switch, etc. Accordingly, the noise of the op amp should be also minimized.

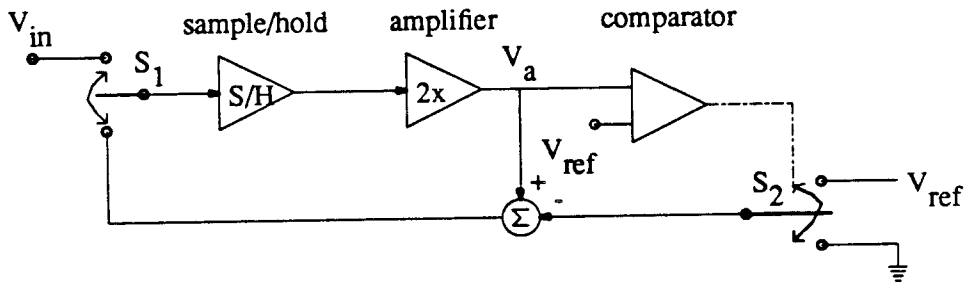


Figure 2.3 Typical application of an internal MOS operational amplifier, algorithmic A/D converter.

Let's look at another application, an algorithmic A/D converter shown in Fig. 2.3.[27,28] The op amp in this case works as a multiply-by-two block as required by the following conversion technique:

At the start of conversion, the input signal is sampled onto the sample/hold amplifier through the switch S_1. This signal is then passed onto the precision gain block (op amp) where it is multiplied by the exact factor of 2.0. To extract the digital information from this analog signal, the loop signal, denoted by V_a in Fig. 2.3, is compared to the reference. If it is larger, then the first bit is set to 1 and the reference is subtracted off from this signal. Otherwise the first bit is set to a 0 and the loop signal is passed on untouched for the second bit conversion.

The A/D conversion process is continued in this manner until the desired number of bits is obtained. The digital information thus comes out in a serial pattern, starting with the most significant bit.

The op amp must provide an 'exact' gain of 2.0 and its output must be settled within one clock period which can be on the order of a few tens of nanoseconds for video and radar signal processing applications. The noise characteristics and the output swing capability of the op amp are also important in terms of the converter's resolution and dynamic range.

When considered for a wide spectrum of applications, the important performance characteristics for op amps concern power dissipation, maximum allowable capacitive load, open-loop voltage gain, unity-gain bandwidth, output voltage swing, equivalent input flicker noise, equivalent thermal noise, power supply rejection ratio (PSRR), common-mode rejection ratio (CMRR), common-mode range (CM range), supply capacitance and die area.[7] However, for a specific application, some of these parameters are less important than others: For instance, when an op amp is used as an integrator in a switched capacitor filter, input offset voltage, common-mode rejection ratio, and common-mode range are less important but they can be important in other applications. A typical set of values for the parameters given above for a conventional amplifier design in 4 μm CMOS technology are given in Table 2.1.

2.3 Generic Architectures for OP AMPs

2.3.1 Two-Stage Architecture

Currently, the most widely used architecture for the implementation of MOS op amps is the two-stage configuration shown in Fig. 2.4(b) which is derived from its bipolar counterpart illustrated in Fig. 2.4(a). This circuit configuration provides good common-mode range, output swing, voltage gain, and CMRR in a simple circuit that can be compensated with a

Table 2.1

Typical performance, conventional two-stage CMOS Internal operational amplifier

(+/-5 V supply, 4 μm SI-gate CMOS)

Ref. P. R. Gray and R. G. Meyer, IEEE J. Solid-State Circuits, 1982.

Dc gain (capacitive load only)	5000
Settling time, 1 V step, C_L = 5 pF	500 nsec
Equiv. input noise, 1 kHz	100 nV/$\sqrt{Hz}$
PSRR, dc	90 dB
PSRR, 1 kHz	60 dB
PSRR, 50 kHz	40 dB
Supply capacitance	1 fF
Power dissipation	0.5 mW
Unity-gain bandwidth	4 MHz
Die area	75 mil^2
Systematic offset	0.1 mV
Random offset std. deviation	2 mV
CMRR	80 dB
CM range	within 1 V of supply

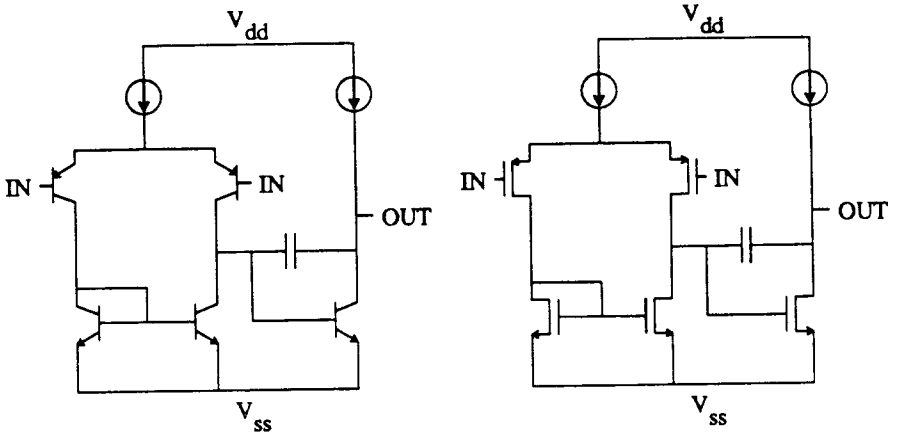
single pole-splitting capacitor.

However, in precision applications involving large values of closed-loop gain, the voltage gain available from the basic circuit shown in Fig. 2.4(b) may be inadequate. One approach to improving the voltage gain, without adding an additional common-source stage with its associated high impedance node and pole is to add a common-gate or 'cascode' transistor to increase the output resistance of the basic amplifier. The basic cascode circuit is shown in Fig. 2.5. It is easily demonstrated that the incremental output resistance of this current source is equal to

$$r_o = r_{o2}(1+g_{m2}r_{o1}) + r_{o1}$$

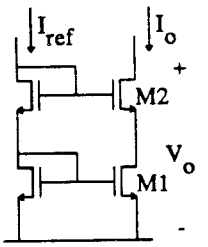
The output resistance is increased by an amount equal to the open circuit gain of the cascode transistor. This circuit may be directly applied to the basic two-stage amplifier in either the first stage, the second stage, or in both stages. The circuit of Fig. 2.6 illustrates the use of cascodes in the first sage. However, a substantial reduction in input stage common-mode range has been incurred which should be alleviated by optimizing the biasing of the cascodes.

The negative PSRR of the basic two-stage architecture can be improved as follows: Conceptually, if the left end of a capacitor could be connected to a virtual ground, then the capacitor voltage would not have to change whenever the negative supply voltage changed in order to have the output remain constant. This is accomplished by inserting a cascode device in this loop with the gate connected to ground, as shown in Fig. 2.7. The displacement current from the capacitor flows into the source of this transistor and out the drain into the compensation point. An additional current source and current sink of equal values (denoted as I's in Fig. 2.7) must be added to bias the common-gate device in the active region and so as not to contribute any systematic offset. The resulting negative PSRR at high frequencies is greatly improved at the cost of a slight increase in complexity, random



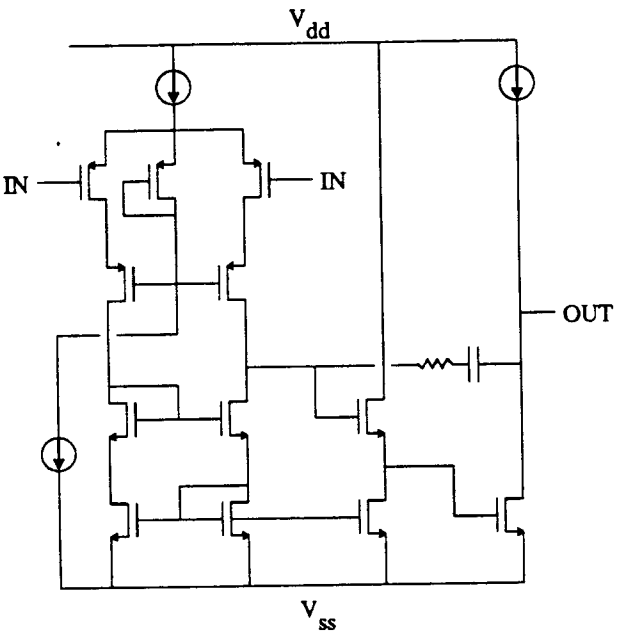
Ref. P. R. Gray and R. G. Meyer, IEEE J. Solid-State Circuits, 1982.

Figure 2.4 Two-stage operational amplifier architecture.
(a) Bipolar implementation. (b) CMOS implementation.



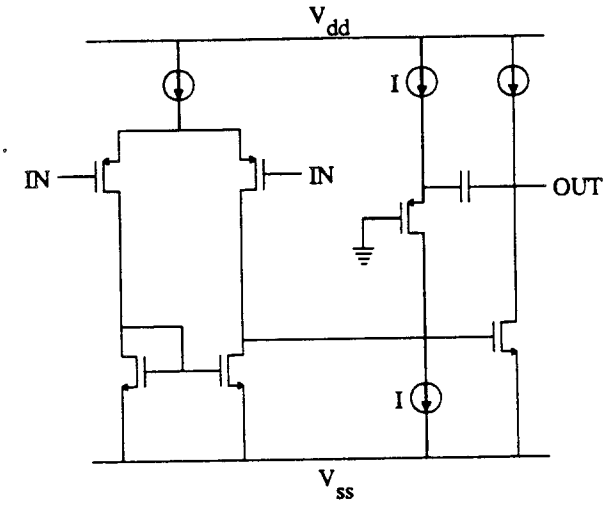
Ref. P. R. Gray and R. G. Meyer, IEEE J. Solid-State Circuits, 1982.

Figure 2.5 Cascode current source.



Ref. P. R. Gray and R. G. Meyer, IEEE J. Solid-State Circuits, 1982.

Figure 2.6 Two-stage architecture with cascoded first stage.



Ref. P. R. Gray and R. G. Meyer, IEEE J. Solid-State Circuits, 1982.

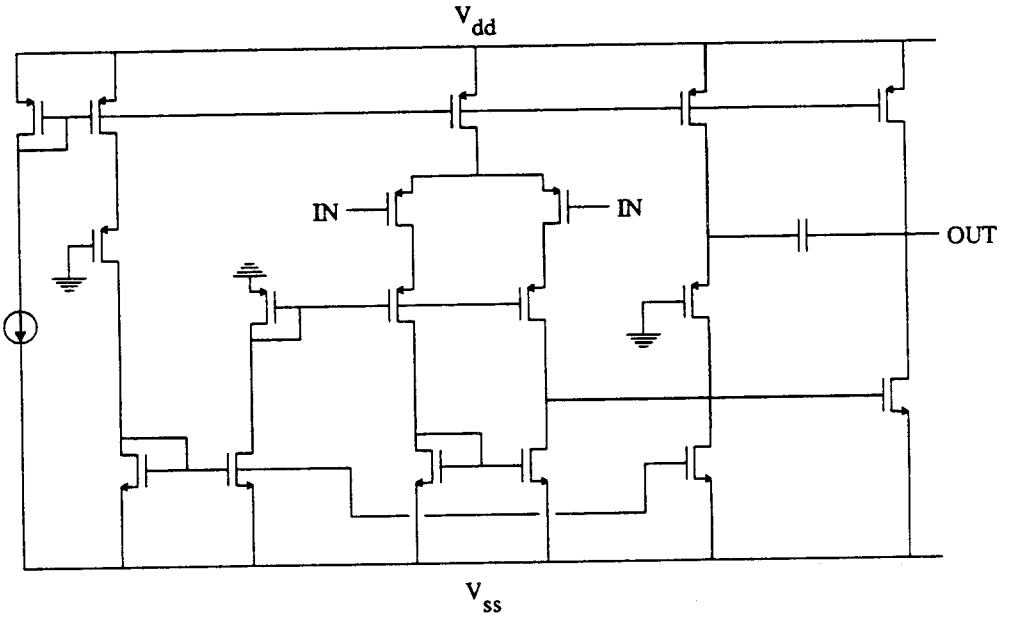
Figure 2.7 Two-stage architecture with cascode feedback compensation.

offset, and noise.[7]

Ahuja also described a modified two-stage architecture shown in Fig. 2.8, where an enhanced compensation technique provides stable operation for a much larger range of capacitive loads as well as much improved negative power supply rejection for very wide frequency range.[29]

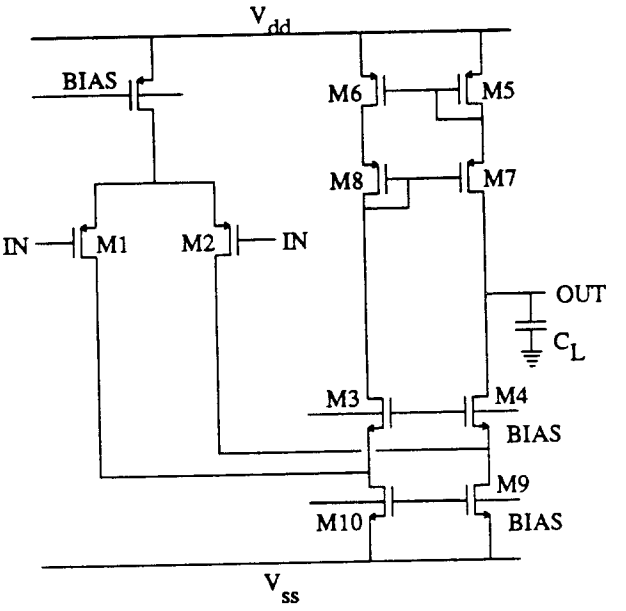
2.3.2 Single-Stage Architecture

The basic two-stage architecture considered thus far is actually a cascade of two common-source stages. An alternative approach is to use a cascade of a common-source stage and a common-gate stage, often called a cascode amplifier. An example of an amplifier utilizing this architecture is shown in Fig. 2.9. The voltage gain of this circuit at dc is approximately the same as that of the basic two-stage circuit due to the increased output impedance. The principal reasons for considering this architecture are twofold. First, the compensation capacitor and load capacitor (C_L) are the same element in this circuit. The first nondominant pole comes from the g_m/C_{gs} time constant of the n-channel cascode devices, and gives a pole frequency approximately at the f_t of these devices. A second nondominant pole results from the differential to single-ended converter. However, the nondominant pole due to the load capacitance present in the two-stage circuit is not present in this circuit. Thus, this circuit is capable of achieving higher stable closed-loop bandwidth with large capacitive load. The principal application of this architecture has been in high-frequency switched capacitor filters.[30,31] Another important advantage of this circuit is that it does not suffer from the degradation of the high-frequency power supply rejection problem inherent in the pole-split compensated two-stage architecture, assuming that the load capacitance or part of it is not tied to a power supply. One of the disadvantages is that the output swing of this circuit is lower than the basic two-stage circuit because of the cascode transistors used at the output. A second disadvantage is that more devices contribute to the



Ref. B. K. Ahuja, IEEE J. Solid-State Circuits, 1983.

Figure 2.8 Two-stage architecture with improved frequency compensation.



Ref. P. R. Gray and R. G. Meyer, IEEE J. Solid-State Circuits, 1982.

Figure 2.9 Single-stage amplifier architecture.

input-referred flicker noise voltage and input offset voltage. Assuming that transistors M5-M8 are biased at the same current as the input devices, the input-referred flicker noise can be shown to be[7]

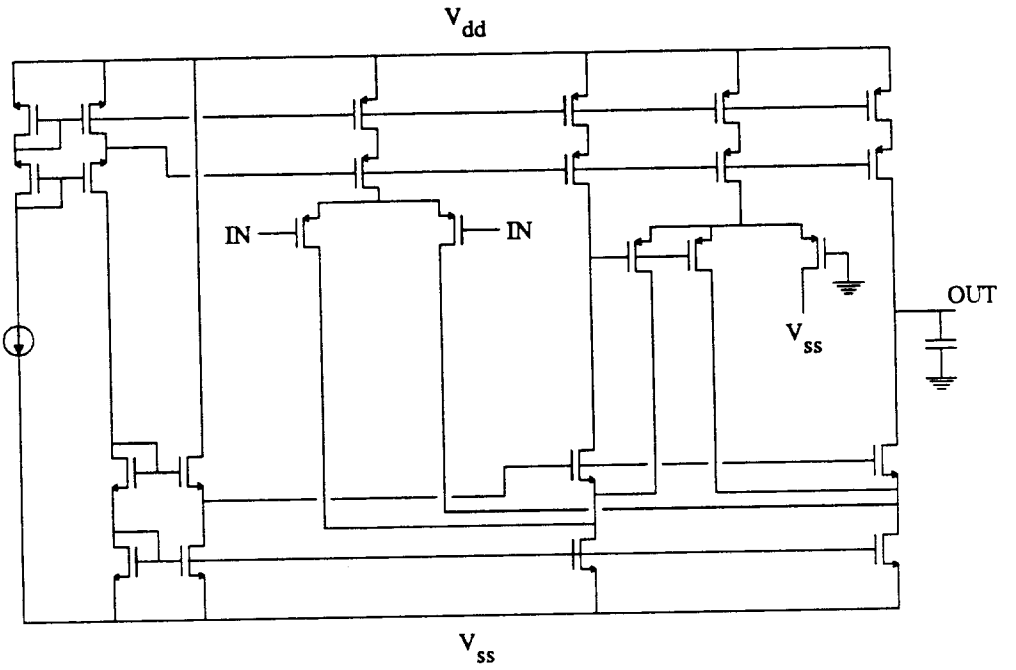
$$v_{eq}^2 = \frac{2K_p}{W_1 L_1}\left[1 + \frac{2K_n\mu_n}{K_p\mu_p}\left(\frac{L_1}{L_9}\right)^2 + \left(\frac{L_1}{L_5}\right)^2\right]\frac{\delta f}{f}.$$

where K_p and K_n are the flicker noise coefficients, and μ_p and μ_n are the mobilities for the p-channel and n-channel devices, respectively. C_{ox} is the oxide unit capacitance, and f is frequency. W's and L's are widths and lengths of the devices.

By enhancing the biasing scheme used in Fig. 2.9, the low and nonsymmetrical output voltage swing of the circuit can be alleviated. The resulting circuit configuration is shown in Fig. 2.10.[32]

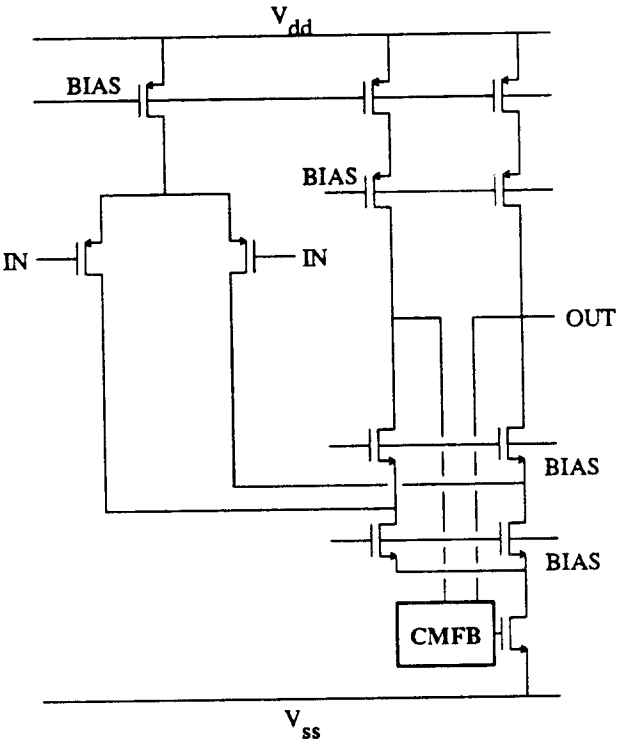
2.3.3 Fully Differential Architecture

Power supply rejection ratio is an important performance parameter for amplifiers to be used in complex mixed analog/digital systems. In addition, one inevitable result of technology scaling is a reduction in power supply voltage with an accompanying reduction in internal signal swings and dynamic range.[33] These two considerations make use of fully differential signal paths throughout the analog portion of the system which is attractive for some systems such as switched capacitor filters. The inherently differential nature of the circuit tends to give very high PSRR since the supply variations appear as a common-mode signal. Also the effective output swing is doubled, while the magnitude of the input-referred op amp noise remains the same, giving a 6 dB improvement in op amp noise-limited dynamic range. An example of a differential op amp architecture is shown in Fig. 2.11. An important problem in such amplifiers is the design of a feedback loop to force the common-mode output voltage to be ground or some other internal reference potential. This feedback path can be implemented with transistors in a continuous-time circuit or can be implemented with



Ref. B. K. Ahuja, IEEE J. Solid-State Circuits, 1984.

Figure 2.10 Single-stage architecure with enhanced output swing.



CMFB: common-mode feedback circuit

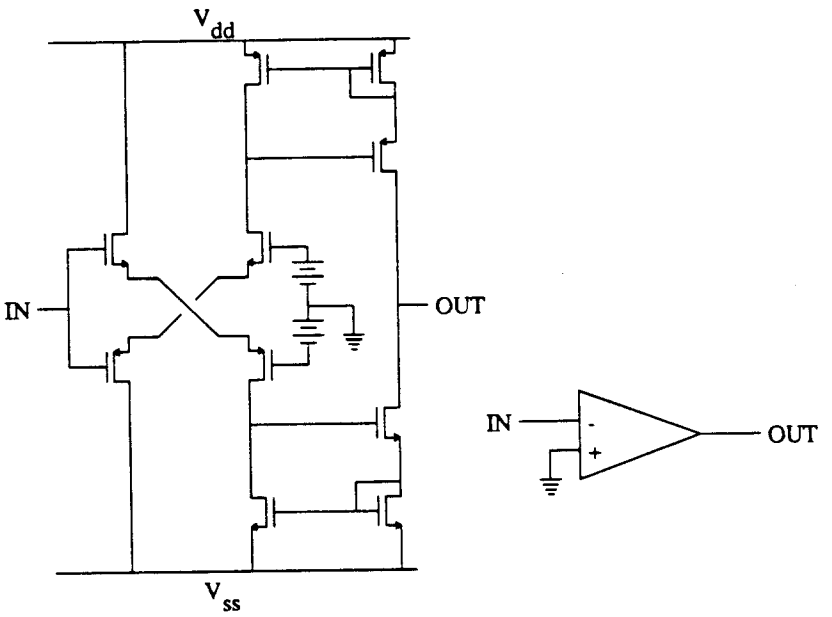
Ref. P. R. Gray and R. G. Meyer, IEEE J. Solid-State Circuits, 1982.

Figure 2.11 Fully differential amplifier architecture.

switched capacitor circuitry. The continuous approach is potentially simpler, but presents a difficult design problem in making the common-mode output voltage independent of the differential-mode signal voltage.[31,34] Switched capacitor circuitry can make use of the linearity of MOS capacitors to achieve this goal.[35] The choice between the two techniques depends on the sensitivity of the particular application to variations in common-mode voltage. Another important advantage of the differential architecture is that the differential single-ended converter with its associated nondominant poles is eliminated. Thus the configuration is particularly well suited to the implementation of high-frequency switched capacitor filters.

2.3.4 Class AB and Dynamic Architectures

In a low-power amplifier where an important objective is the minimization of chip power, a 'class AB' architecture can be effectively used. Here the term class AB is taken to mean a circuit which can source and sink current from a load which is larger than the dc quiescent current flowing in the circuit. The motivation for using class AB architecture is that one of the factors that dictate the value of the quiescent current in an MOS amplifier is the value of current required to charge the load and/or the compensation capacitor in a specified time. However, it is relatively rare that the op amp outputs actually have to change the maximum amount in one clock cycle. Large power savings can be achieved by drawing only that amount of current required to charge the capacitance on that particular cycle. An example of single-stage op amp which operates on this principle is shown in Fig. 2.12. This particular circuit can be used in the inverting mode only. With the input grounded, the quiescent current in the input transistors is determined by the bias voltage shown. Upon the application of a voltage to the input, the current in one side of the input stage increases monotonically with the applied voltage until the power supply is reached, while the other side of the input stage turns off. The amount of current available at the output is much larger



Ref. P. R. Gray and R. G. Meyer, IEEE J. Solid-State Circuits, 1982.

Figure 2.12 Single-stage class AB amplifier architecture.

than the quiescent current, and the circuit does not display slew rate limiting in the usual sense. Another advantage of this circuit is that the small-signal voltage gain in the quiescent mode can be quite high because of the low current level. The fact that the voltage falls off during transients because of the high current levels is of little consequence for the ac response of the circuits.

DeGrauwe *et al.* have described a novel approach achieving the same objective.[36] A conventional class A amplifier architecture is used, but an auxiliary circuit detects the presence of large differential signals at the input. The bias current in the class A circuitry is then increased when such signals are present.

Dynamic architecture has been explored by several authors. Copeland[37] described a configuration in which the quiescent current in the absence of signals is allowed to decay to zero. Such amplifiers are fully dynamic in the sense that no dc paths exist for current to flow from the supply. While very low power dissipation can be obtained, difficult problems of settling time and power supply rejection remain to be solved with these amplifiers. Hostica [38] invented a biasing scheme where the power supply current is independent of signal amplitude, and is made large during the early part of the clock period (used in the switched capacitor filter) for fast slewing and is made small during the later portion for high gain and power savings.

2.4 Conventional Synthesis Procedure

Op amps like other analog circuits traditionally have been fully hand-crafted by expert analog circuit designers. A conventional synthesis procedure for op amps as illustrated in Fig. 2.13 can be divided into three design phases: Circuit topology selection, device sizing, and layout generation. First, a suitable op amp circuit is selected based on the general domain of the performance specifications. Then each device in the chosen circuit is properly

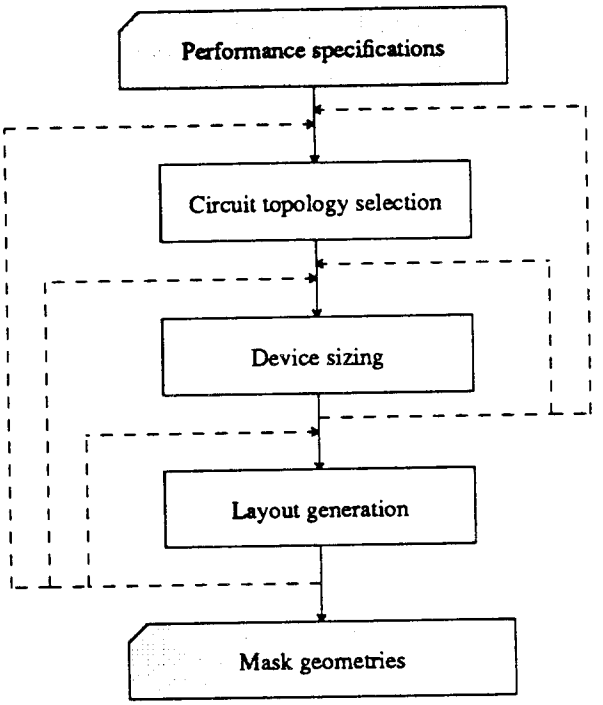


Figure 2.13 Conventional synthesis procedure for op amps - Simplified view.

sized to meet the given specifications. Finally, the sized circuit schematic is laid out in mask geometries. Feedback paths (shown in dotted lines in Fig. 2.13) around some of the design phases represent possible re-design efforts needed when the design results from those phases violate the design requirements implied by the performance specifications. Let's look at each of the design phases in more detail.

The circuit topology selection phase is subdivided into two steps as shown in Fig. 2.14. First, based on the performance specifications given and the process technology to be used, a suitable op amp architecture must be selected: Two-stage, single-stage, fully differential, dynamic, class AB, etc. Upon deciding which architecture is suitable for the particular application, a specific circuit topology is selected for the synthesis. The examples of op amp performance characteristics which are used to make such decisions include power supply voltage, power consumption, and DC transfer curve. The many possible variations of different architectures make this phase of the synthesis rather dependent on design experience. A poor selection decision made in this phase may cause a catastrophe at the end of the design cycle as shown in Fig. 2.14 or at best the designer realizes the poor decision in the device sizing phase.

In the device sizing phase the designer tries to adjust variable parameters of each device (which are called 'design parameters') so as to meet all performance specifications given. Width and length of a MOSFET, capacitance value of a capacitor, and current value of a current source are the design parameters for CMOS op amps. The design task is initiated by setting the design parameter values to achieve the desired DC characteristics (quiescent bias points). Next the designer readjusts the initial parameter values to get the specified ac (small signal) response about the quiescent points. Finally, the characteristics of the transient response are used to make further changes in the design parameter values and operating points in order to meet the performance requirements. The entire procedure is repeated until all the design requirements are satisfied. In case some of the requirements are still violated

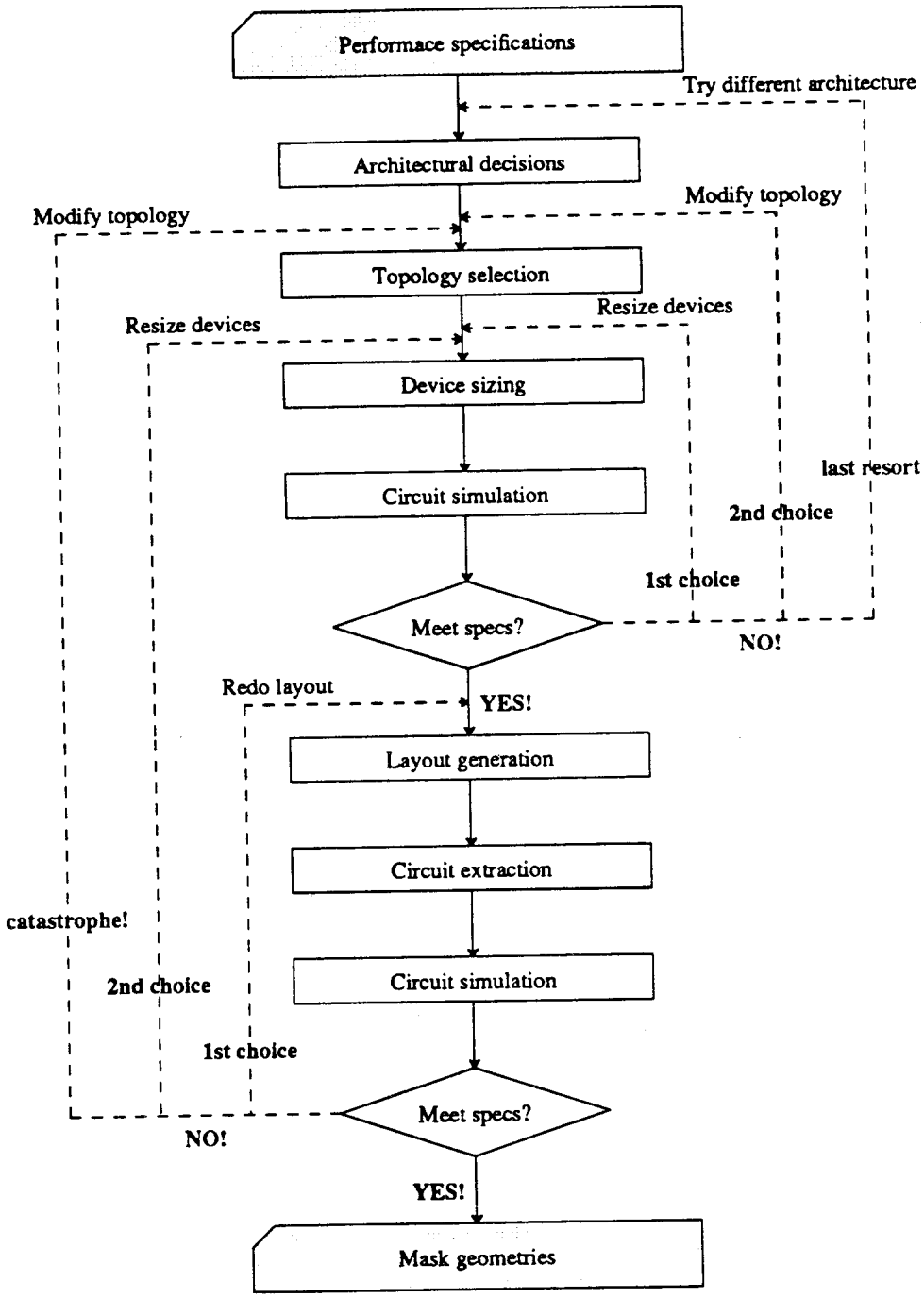


Figure 2.14 Conventional synthesis procedure for op amps - Detailed view.

in spite of the repeated design efforts, the chosen topology is often modified in order to enhance certain performance characteristics as needed. If all these efforts go wrong, a different architecture may have to be selected as a last resort.

In fact, the afore-mentioned design task (device sizing) can be formulated as a parametric optimization problem. Given a fixed circuit topology and performance specifications, some of the variable device parameters can be selected as design parameters and the specifications can be transformed into constraints and performance objectives. Then the rest of the problem is reduced to finding the optimal values of the design parameters that maximize the performance objectives while keeping all constraints satisfied. Chapter 5 is devoted to the parametric optimization aspect of op amp design.

Layout generation starts with a sized circuit schematic of an op amp. The designer first reduces the complexity of the problem by adding a higher level of abstraction to the circuit. For instance, a few circuit elements are grouped together to form a building block based on their connectivity, geometrical matching, functionality, etc. This abstracts the circuit as a composition of several building blocks. The designer then sketches a rough placement of these blocks considering layout rules, terminal locations, routing of critical signals and so forth. After the initial placement of the blocks, the designer routes the nets and at the same time compacts the resulting layout as much as the spacing rules allow. Sometimes the initial placement must be modified for better routing capability, better area utilization, or better performance. When the layout is done, parasitic elements such as parasitic capacitances, and resistances from either devices or wirings are extracted from the mask geometries. These parasitic elements are then added to the initial circuit schematic and the resulting circuit is simulated for verification. If all the specifications are met, the design is completed. Otherwise, the current layout must be modified to reduce the excessive parasitic elements. However, if repeated modification efforts do not produce the correct design, some of the design parameters of the circuit must be readjusted. In the worst case, the designer

may have to modify the current circuit topology to improve the performance as necessary.

From the above discussion, it can be readily understood that the efforts and time required to design even a single op amp may be significant when the performance specifications are difficult to be achieved: The entire design cycle (as shown in Fig. 2.14) may have to be repeated many times. As a result, the time and efforts devoted to the design of analog interface subsystems used in mixed analog/digital systems constitute a bottleneck in application-specific and customer-specific design of IC's where time to market requirements are more important than component cost.

2.5 Review of Automatic Synthesis Systems

The history of automatic synthesis tools for op amps goes back to the days of APLSTAP[39] and DELIGHT.SPICE.[17] These systems consist of a general-purpose optimization program and a circuit simulation program. Accordingly, they have been used in many IC design problems, with bipolar op amp design being just one of them. However, it is not until late 1980's that intensive research efforts have been put into design automation for analog circuitry including CMOS op amps. Knowledge-based design tools such as An_Com,[23] IDAC,[20,40] and OASYS[19] have been announced.†

On the other hand, an optimization-based system such as ECSTACY[18] has been developed as a general-purpose parametric optimization tool. In this section, these synthesis tools are reviewed to discuss their contributions and limitations.

OASYS[19] is a behavior-to-schematic synthesis tool for analog functional blocks. In OASYS, analog circuit topologies are represented as a hierarchy of functional blocks and a planning mechanism is introduced to translate performance specifications between levels in

† There exist a few other programs developed for analog functional block synthesis such as BLADES,[41] PROSAIC,[22] and one by Allen.[21] However, at the time of their publications, they were either not mature enough to handle the entire synthesis of op amps or not applied to op amp design.

this circuit hierarchy. The program synthesizes a sized circuit schematic for CMOS operational amplifiers and, more recently, for CMOS comparators[42] from performance specifications and process parameters. A sized circuit schematic from OASYS is then sent to the full-custom analog cell layout tool, ANAGRAM[26] to complete the schematic-to-layout path.

IDAC[20,40] is the first system that formally announced a behavior-to-schematic synthesis for CMOS op amps in 1984. The system has been further enhanced to design not only op amps but also other analog functional blocks such as voltage references, comparators, and A/D converters. In IDAC, a specialized software routine is associated with each circuit topology covered by the system. Each routine is self-contained: Not only design decisions (in the form of if-then-else rules) customized for that particular topology but also process-technology-specific data have been included in the routine. The program also generates mask geometries of the sized circuit schematic using the analog layout tool, ILAC.

An_Com [23] is a behavior-to-layout synthesis tool developed for analog functional blocks. In An_Com, the silicon compilation process is broken down into several steps, thus resulting in a sequence of successive decompositions. At each decomposition step, the high level specifications are broken into the specifications for the elements or subblocks at a lower level in the hierarchy until the resulting elements are leaf cells that exist in a library or can be produced by a layout generator. This decomposition process is driven by domain-specific knowledge pre-stored in templates. Currently, An_Com can produce a mask-level layouts of CMOS op amps starting from performance specifications.

ECSTASY[18] is an interactive, general-purpose IC design optimization tool applicable to a broad spectrum of circuits. A forms-based user interface developed for analog design allows the designer to formulate design problems easily. Gradient-based algorithms (such as feasible directions methods) as well as random search algorithms are employed in order to avoid numerical difficulties in the optimization. The tool has been used for switched

capacitor filters and bipolar op amps but not for CMOS op amps so far.

To summarize, OASYS and An_COM are based on a hierarchical decomposition of performance specifications down to leaf cells (such as current mirrors or differential pairs). But for the synthesis of such basic functional blocks as op amps, comparators, voltage references, output buffers, etc, it is very difficult to decide how the given specifications should be decomposed because of the close, sometimes intricate dependencies between the leaf blocks. Besides, in order for the method to take advantage of specific topologies and their known usefulness, the system must store such information in the database as templates. This violates the hierarchical approach from the method. In fact, such templates are dominantly used for op amp synthesis in both systems.

The OASYS' layout program, ANAGRAM[26] uses a simulated annealing method for placement and a constraint-based line expansion algorithm for routing. However, simulated annealing is in general too inefficient to handle versatile constraints arising from variable-shaped blocks different terminal configurations, geometrical symmetry among some blocks, etc. The routing algorithm developed mainly concerns itself with noise coupling due to internodal capacitances between the signal wires and with shared parasitic resistances in the DC power supply wiring. Constraints such as upper bound in routing capacitance, matching routing parasitics between signal nets, and specified assignments of certain wires to certain layers are not treated. After all, op amp layouts from ANAGRAM look quite different from those hand-crafted by expert designers. In An_Com, op amp layouts are generated based on a slicing structure; its overall layout strategy (such as a floorplanning method) is not mentioned except that it is based on leaf cell generators.[23]

IDAC[20,40] operates on a flat circuit description, i.e. each topology is a basic unit. However, the consolidation of design 'strategy' and topology-specific 'data' creates much overhead whenever a new circuit topology is added into the system.

Some disadvantages common in all three systems are as follows: Firstly, they require a priori design knowledge to be integrated into the system for each new type of circuitry. Secondly, high degrees of optimization are difficult to be achieved. Since the design strategies consider some but not all of specifications at a time, it is hard to avoid order-dependency in the design process. The op amp layouts from IDAC look promising but no detailed description on IDAC is available.

On the contrary, ECSTASY neither uses a priori design knowledge nor suffers from the order-dependency in its optimization strategies. But the system is capable of only parametric optimization: It can neither select a good circuit topology for an application nor generate a layout. Another drawback is that the user must have detailed understanding of the circuit he/she tries to optimize.

Recognition of the shortcomings of both knowledge-based and optimization-based synthesis approaches has lead to a new approach to the compilation of op amps which is the subject of this thesis and which is described in chapters 4 through 6.

CHAPTER 3

OPASYN: A COMPILER FOR CMOS OP AMPS

3.1 System Overview

OPASYN is a behavior-to-layout synthesis tool for CMOS op amps. It is developed with the following objectives:

a) *Mapping from performance specifications to mask geometries.* The synthesis system must perform the complete synthesis from specifications to detailed layout.

b) *High quality solution.* A solution from the program must meet all the performance specifications and consume minimal power and area.

c) *High computation efficiency.* The program must run fast enough so that it can be used interactively at the system design level; this means turnaround time should be on the order of a few minutes or less.

d) *Minimal user interaction.* The program must possess ample design knowledge about op amps to complete the synthesis task with minimal help from the user. In this way, even a novice designer, i.e., a system designer who needs to employ an op amp in his/her system design can use the program without having to learn op amp design.

The rest of this section presents the framework of the program and gives an overview over the flow of the synthesis.

The program consists of an internal database and three functional modules: a circuit selection module, a parametric circuit optimization module, and a layout generation module.

The framework of OPASYN and its interface to other physical design tools in the Berkeley CAD environment are shown in Fig. 3.1.

The *database* contains the necessary design knowledge for each op amp topology available to the users of the system; it includes[24,25]

- a decision tree for topology selection,
- analytic circuit models for parametric circuit optimization,
- slicing structure descriptions for floorplanning,
- netlist descriptions for routing.

Device parameters and layout design rules for different process technologies are retrieved from a technology library which is a part of the database.

The *circuit selection module* is a rule-based system where an inference engine selects a promising circuit topology based on the user specification. The decision tree used for topology selection is implemented as a set of production rules (if-then rules) in the database to permit flexible and easy upgrading of the system. In addition, a simple query mechanism has been incorporated to explain the decision made by the system or to show the rule(s) used in the decision process. A detailed description is presented in Chapter 4.

The *parametric optimization module* is composed of an optimization routine and an interface to the circuit simulator SPICE. First, the analytic model for the given circuit topology (that is pre-designed by an expert circuit designer and stored into the database) and the data for the specified process technology are loaded from the database. Starting from the given circuit schematic, optimal sizes of the devices in the circuit are determined so as to meet all input specifications. The optimization routine is quite general and works in the same manner for all circuits to be optimized and for all process technologies. The SPICE interface routine generates input decks for SPICE, makes system calls to SPICE, and interprets the results from SPICE to verify the optimization results. Chapter 5 discusses this subject in detail.

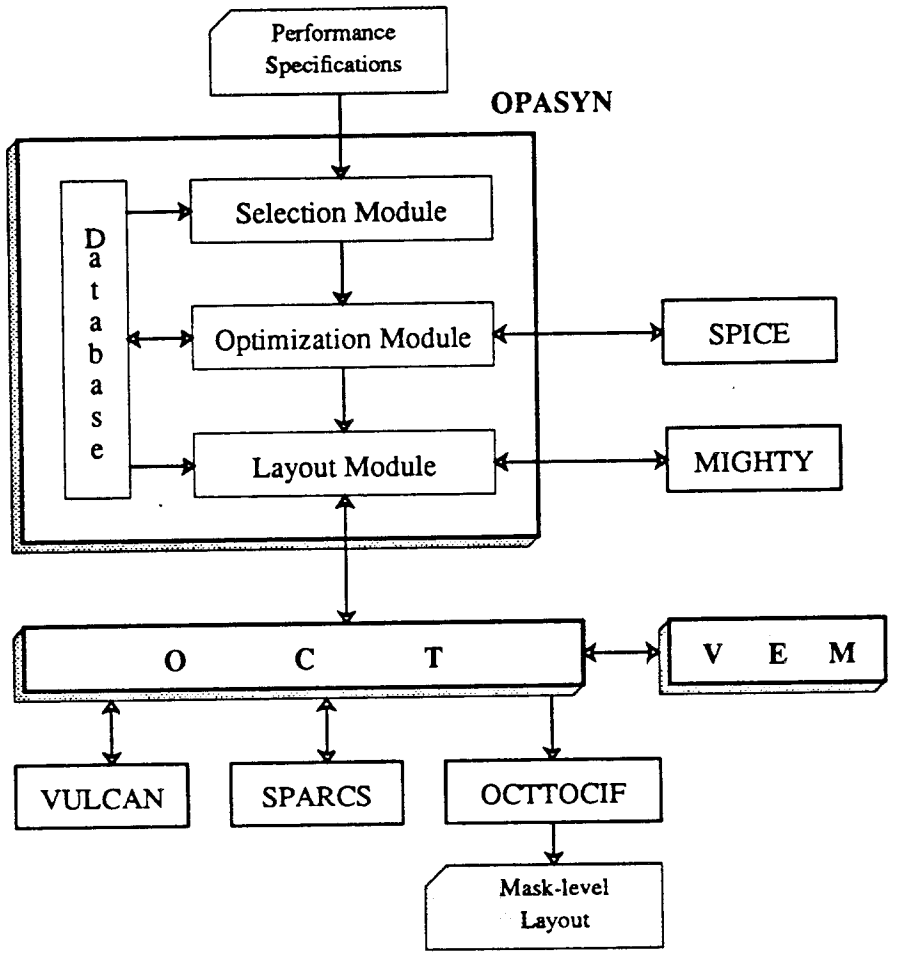


Figure 3.1 OPASYN Framework and its Interface to the Berkeley CAD Environment.

The *layout generation module* includes a floorplanning routine, parameterized leaf cell generators and interface routines to a router and a spacing program. The input to this module is a sized circuit schematic. The module outputs design-rule-correct mask geometries of an op amp layout. The leaf cell generators provide small circuit elements such as MOSFET's, MOSFET pairs, and capacitors at run time. The interface routines take care of constraint generation and intermediate data format conversion.† The layout module also works independent of any specific layout design rules; these rules are parameterized and read from the database. The layout generation is the topic of Chapter 6.

The synthesis flow in OPASYN is shown in Fig. 3.2. Synthesis starts from a set of op amp performance specifications. Based on the general domain of the specifications, the program selects an appropriate option out of a small database of generic, widely applicable op amp circuit topologies, unless the user explicitly specifies the particular circuit topology to be used. For the chosen circuit, optimal values of its set of design parameters will be determined to meet the objectives implied by the given specifications. [25] The SPICE interface program is called by this module to verify the optimization result through extensive circuit simulation.[43,44] Finally, the layout generation module takes the netlist of the sized circuit schematic from the previous synthesis phase together with a specification of a desired aspect ratio or of a vertical/horizontal size constraints, and a reference to a file of geometrical design rules. It produces an unspaced, symbolic layout of the circuit using a macro cell style. The symbolic layout is then properly spaced by a spacing program to produce design-rule-correct (spaced) mask geometries.

The modular architecture and flexible functional modules used in OPASYN make it possible to extend the system to synthesize other analog building blocks such as output buffers, comparators, and bandgap references with minimal alterations to the program.

† In fact, the intermediate data format conversion is necessary for routing only. All the other tools communicate through the data manager OCT (which is explained in the next section).

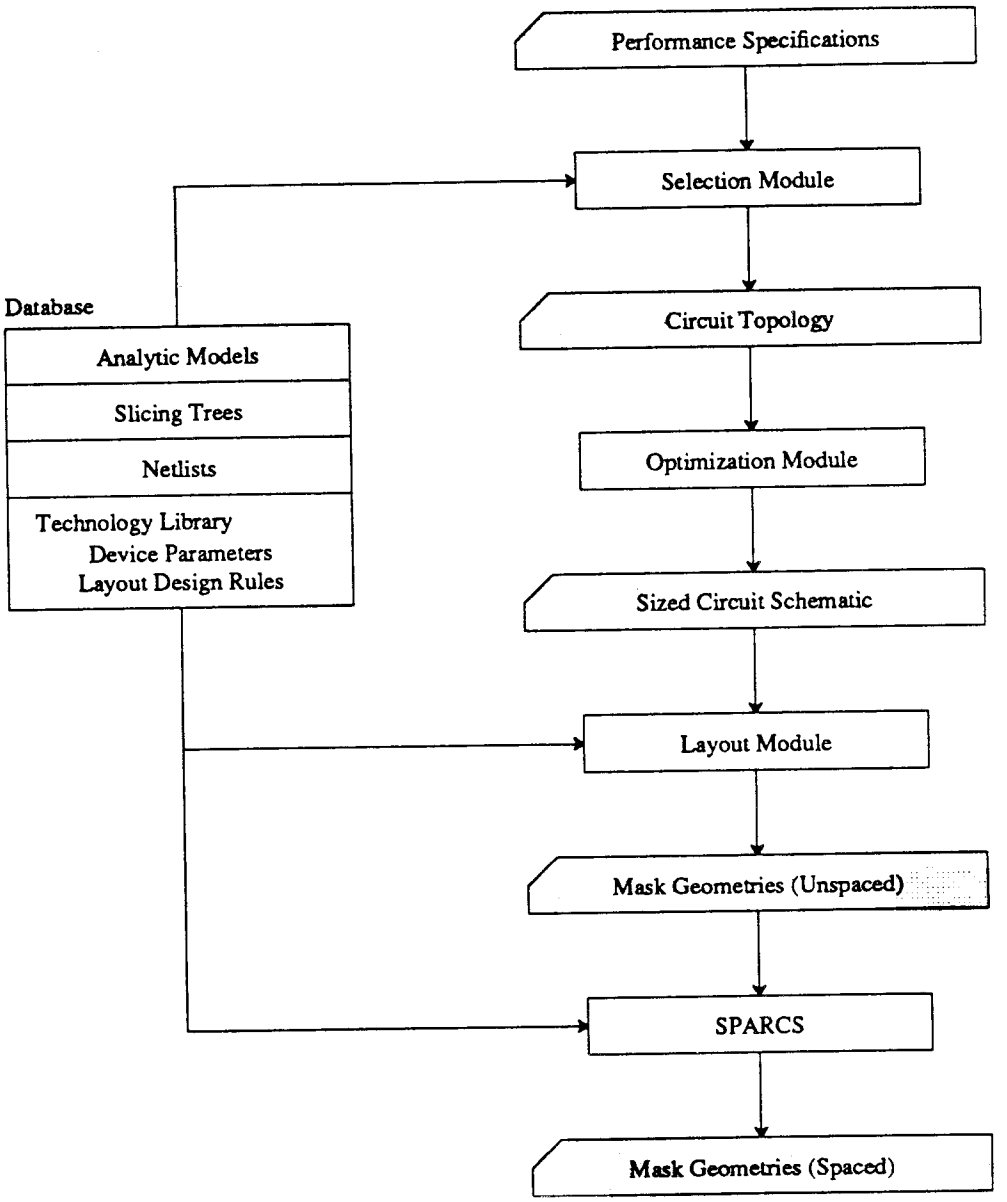


Figure 3.2 Synthesis Data Flow in OPASYN.

Various technologies such as CMOS, bipolar, and BiMOS can be also accommodated. Up to now, CMOS technology has been exclusively employed because of its prevailing usage in analog interface component designs.

3.2 OPASYN and the Berkeley CAD Environment

OPASYN gets much benefit from the Berkeley CAD environment, especially for its layout generation. Fig. 3.1 shows that many physical design tools communicate with OPASYN either directly or indirectly through a data manager. In fact, the internal database of OPASYN provides only the specific design knowledge necessary for the layout synthesis: layout design rules, 'slicing structures' for the circuit topologies (will be explained in Chapter 6), and netlists. Design data (mask geometries) produced at each phase of the layout generation is hierarchically managed by the data manager OCT[45,46] developed at U.C. Berkeley. In the rest of this section, the abstraction hierarchy of OCT and its application to OPASYN are explained, and other design tools used in the synthesis process are briefly described.

OCT is a hierarchical, object-oriented data manager developed for VLSI/CAD applications. The program offers a simple interface for storing information about the various aspects of an evolving chip design. A basic unit in a design is called a 'cell'. A cell is a portion of a chip (or the entire chip) the designer wishes to consider as a unit. A cell has many aspects called 'views', depending on the design stage (i.e. logic synthesis, layout synthesis, circuit simulation, etc) and on the design style to be used. Each view has one 'facet' named 'contents' that contains the actual definition of the view as well as various application dependent 'interface' facets that contain the minimal amount of information about the view necessary for its manipulation or for interfacing to it from the outside. With these abstraction entities, cells can be defined hierarchically. This means that a view of a cell can contain 'instances' of other cells whereas each of these cells has their own views and possibly

instances of some other cells). The different views communicate with one another through suitably abstracted interface facets.

In OPASYN, during various phases of the layout process, different views such as a 'physical view', an 'unspaced view', and a 'spaced view' are used. Leaf cells are represented in a physical view with fully specified geometry for all devices and connections stored in their contents facets. They observe all specified design rules and need no further spacing. Their interface facets contain overall cell shapes, terminal locations, and protection frames. The entire layout of an op amp, on the other hand, is represented in an unspaced view where only relative placement, sizes and shapes of instances of component blocks, and the general routing of the interconnections are recorded in its contents facet. This abstraction of the layout needs to be subject to a spacing program where all the design rules are checked and the placements of the leaf cells are suitably adjusted. The result is a design-rule-correct spaced view of the entire op amp. Fig. 3.3 illustrates the hierarchical data abstraction discussed so far.

The advantage of using such a general data manager is two-fold. By observing a set of pre-defined policies , our layout program can communicate with other existing tools with minimal overhead and with no need for intermediate data conversions. For instance, OPASYN can call the spacing program SPARCS to re-space the layout in symbolic format, or the dedicated graphics editor VEM to visually verify the synthesis result without having to provide explicit interface modules to these programs. The system development time could be significantly reduced by relying on this data manager and on many existing tools.

VEM is an interactive graphics shell/editor for IC designs represented using the OCT data manager.[45] The primary use of VEM by OPASYN is to provide a means of looking at the graphics representation of an OCT view and invoking various CAD tools on these views. VEM also provides standard graphics editing capabilities to the users.

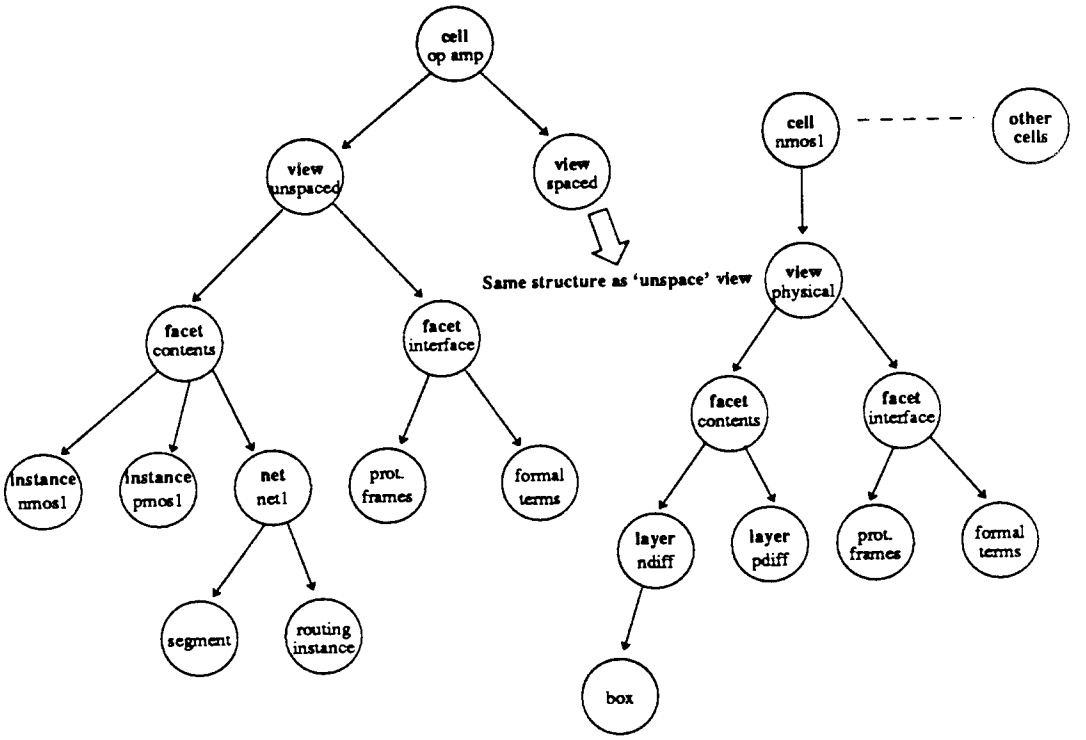


Figure 3.3 Hierarchical data abstraction used in OPASYN.

VULCAN creates/updates geometrical information and required properties (such as technology, cell type, etc.) in an interface facet for an OCT view from its contents facet.[47] Formal terminals of a cell are copied to the interface facet along with their implementation, and all the geometry other than terminal implementation geometry on each layer in the contents facet is merged into a protection frame for that layer in the interface facet. Instances of other cells are replaced by the geometry contained in their interface facets before this merging process.

SPARCS is a graph-based, 1-dimensional compaction program for spacing IC symbolic layouts. Its input is an OCT view of type 'unspaced' and its output is another OCT view called 'spaced' with the elements translated so that the overall area is minimized and the spacing rules are satisfied. The usage of SPARCS in OPASYN is further discussed in Chapter 6.

OCTTOCIF accepts as input an OCT facet and produces a CIF file.[48] The program is used to generate mask geometries of a layout that is suitable for a certain IC foundry (i.e. MOSIS).

MIGHTY is a two-layer symbolic detailed router for any rectagonal routing area.[49] The routing process is also explained in Chapter 6.

CHAPTER 4

HEURISTIC CIRCUIT SELECTION

4.1 Alternatives

The first step in op amp design is to obtain a promising circuit topology based on the given design requirements. As illustrated in Fig. 2.14, expert designers perform this task in two phases. They first make an architectural decision using some of the design requirements such as load capacitance, load resistance, power dissipation, noise requirement, etc. Among many, existing variants of the chosen architecture,† they then select a topology which, based on prior experience, fits the new application. Or they may modify one that almost fits by enhancing specific performance characteristics, again partly based on experience and partly by trial and error. This often requires many design cycles.

For knowledge-based synthesis systems that must perform the same task, the following approaches can be considered:

- Creation of a topology from basic building blocks,
- Modification of basic (seed) topologies,
- Selection from a library of 'proven'†† topologies.

Let's investigate advantages and drawbacks of these approaches to find out which one is most practical. It should be noted that the above approaches can be applied not only to op amps but also to most other analog functional building blocks such as output buffers, voltage references, comparators, etc.

---

† We mean, by an architecture, a collection of similar topologies.

†† By a proven topology, we mean that the topology has been proven to have good, stable performance and is widely used in the industry.

4.1.1 Creating New Topologies

Creating a topology required by the performance specifications using building block elements and basic knowledge about circuit design would be the most general solution. If such an approach is feasible, no circuit topologies need be pre-stored in the database, and there would be no overhead of adding new topologies as the application range of the synthesis program is expanded. However, creating a topology from basic principles is very difficult for a couple of reasons. First, the tight and often intricate coupling between functional modules in typical op amp circuits makes hierarchical decomposition of functionality very difficult. As a result, breaking the high-level specifications into the specifications for the lower-level building blocks in the hierarchy is not well understood. For instance, the phase margin of a two-stage op amp must be determined by considering the frequency response of the entire circuit. Propagating the constraints imposed by the phase margin of the circuit to its constituent modules, namely input stage, output stage, and compensation stage is very difficult. As a matter of fact, most synthesis techniques for analog circuits are based on analysis; an appropriate circuit topology is first created and all component values are then adjusted to meet the overall design requirements by analyzing the circuit. Second, a circuit topology should be designed to cancel out first-order variations in design parameters to yield stability and immunity against a rather wide range of spreads in integrated circuit active and passive component values. In other words, whenever a new circuit topology is constructed, the sensitivities of its performance characteristics to its component value changes should be derived and checked. If the sensitivities are considered too high for the design requirements, the topology should be modified.

Bowman et al.[1] made an effort to create op amp topologies from basic design principles but with very limited success; the generated topologies are simple two-stage op amp circuits even without any compensation scheme. Similar efforts have been put into OASYS[2,3] and An_Com.[4] However, no new topologies have been invented by these systems so far.

Practical op amp topologies are still being devised by expert designers. When one takes a closer look, one realizes that the above systems do not really create any topologies but rather *assemble* basic building blocks into complete topologies based on pre-stored templates (in An_Com) or plans (in OASYS) which implicitly contain the proven op amp topologies. Therefore, one might as well store proven topologies explicitly and select one of them (as described in 4.1.3).

4.1.2 Modifying Seed Topologies

An easier approach is to start from some basic (seed) op amp topologies and to modify it to meet certain design goals. For each architecture to be used, a seed topology is defined. For instance, a topology composed of an input stage, a compensation stage, and an output stage is a seed topology for the two-stage op amp architecture. Similarly, one with a differential pair and a gain stage serves as a seed topology for the single-stage op amp architecture. The most appropriate seed topology is then selected according to the architectural decision made from the specifications. The selected seed topology is then modified during the device sizing phase of the synthesis as required by the design requirements. This approach has the advantage that the seed topology can be incrementally modified as it becomes clear which design requirements are hard to match. Thus a detailed topology does not have to be selected in the beginning; only the architectural decision has to be made at the top level. Subsequently, two kind of modifications can be made: weak modification and strong modification.

In *weak modification*, the topology change is incremental and is made within the same architecture. Fig. 4.1 shows an example of the weak modification. In this case, the seed topology (two-stage op amp with pole-zero compensation) is modified to produce a higher small signal gain; the input differential pair and the active load have been replaced with cascoded versions. However, since these changes limit the voltage swing at node A, the

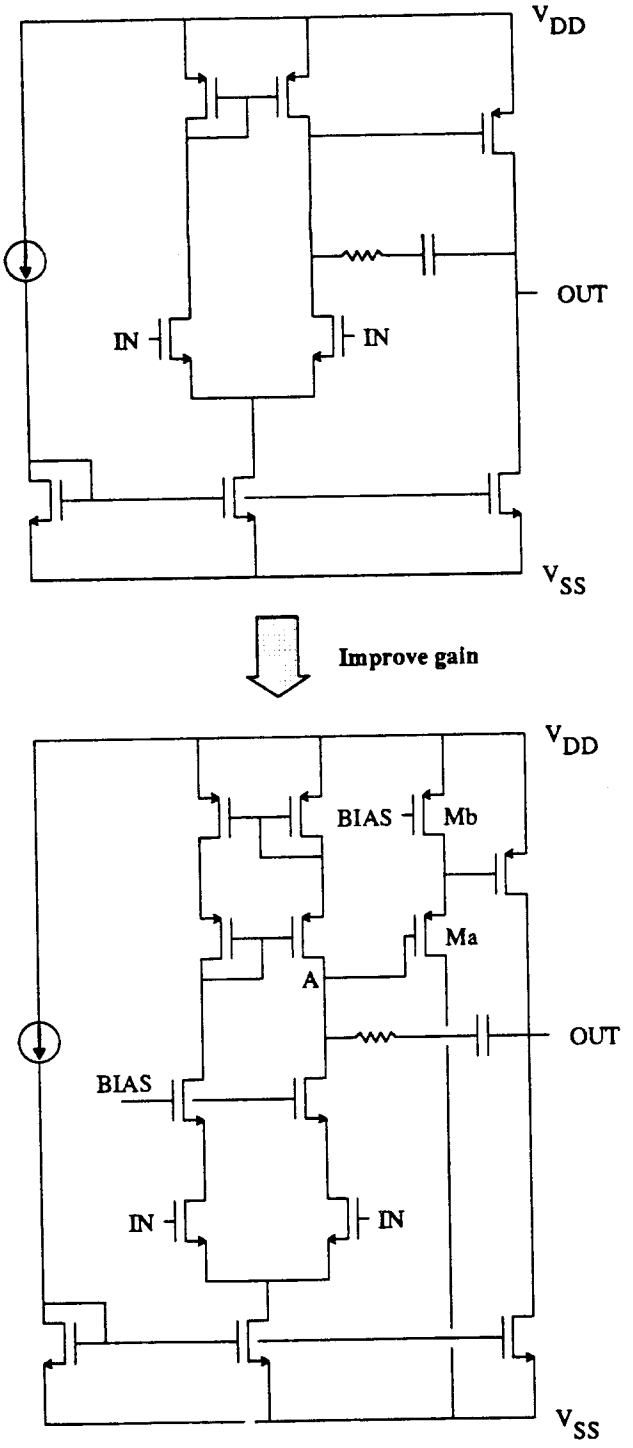


Figure 4.1 Weak modification of topology for improved performance.

level shifter composed of transistors Ma and Mb is added to the circuit. We consider the change as incremental in the sense that the resulting topology is still a variant of the two-stage architecture. But even with these incremental changes, the resulting circuit is a rather new topology in terms of its ac performance (phase margin, unity-gain bandwidth, etc). This is because of the newly introduced doublet (pole-zero pair) from the source follower Ma. Now, the compensation problem has been changed from that of a two-pole and one-zero system to that of a three-pole and two-zero system. It is well known that the latter system is very difficult to compensate and the currently used pole-zero compensation scheme may not work for this topology. This incremental change also requires more power and area consumption as well as different bias voltages. The implication here is that weak modification in most cases is not robust; an incremental change made to improve a certain performance characteristic often causes other characteristics to deteriorate.

In *strong modification*, the topology change is rather global and the resulting topology is often no longer a variant of the same architecture. Fig. 4.2 illustrates an example of strong modification. The seed topology in the figure is modified to improve the slew rate characteristic. An incremental modification in this case would be to increase the bias current I_{SS}; the slew rate is calculated from $\frac{I_{SS}}{C}$ and reducing C would worsen the ac response. But increasing the bias current I_{SS} will increase the power and area consumption. Therefore a strong modification of the topology is made, which can increase the slew rate without increasing the power dissipation. The resulting topology shown in Fig. 4.2 is the outcome of such a strong modification where the output stage of the original topology has been replaced with a class AB output stage.[5] Consequently, the new topology slews better even with reduced power dissipation. However, the change made here is so drastic that not only the transient characteristics but also the ac characteristics and the dc characteristics of the original topology have been changed; the new topology is actually a variant of the class AB

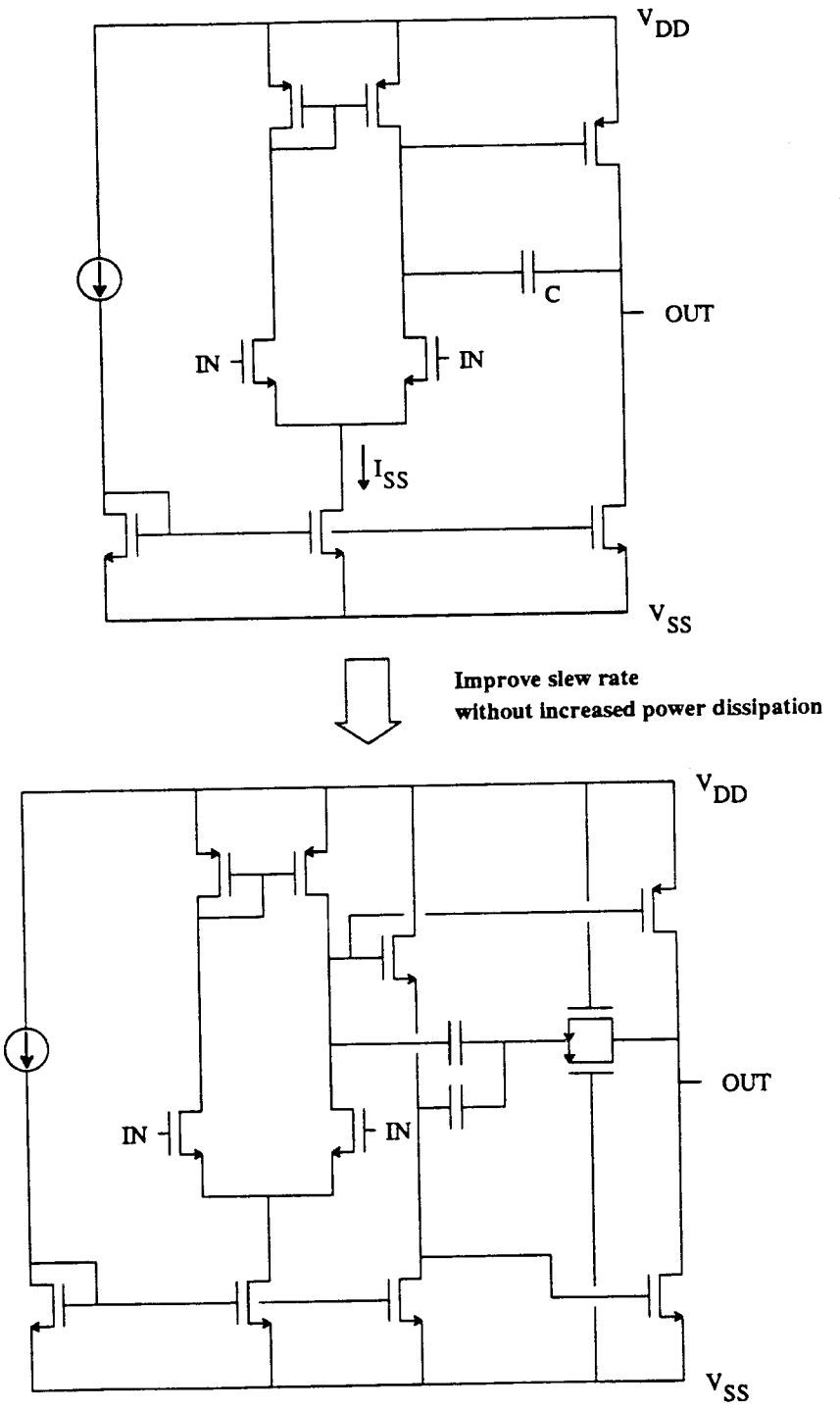


Figure 4.2 Strong modification of topology for improved performance.

architecture.

From the above discussion, it can be seen that an incremental modification of a topology in order to improve a certain performance characteristic often fails. The effect of the modification to all the other performance characteristics must be considered at the same time. When sensitivity issues are considered as well, the situation gets even more difficult; 'intelligent' strong modification is required in most cases. However, such modifications are as difficult as creating new topologies. Therefore, it is more efficient and robust to switch among a set of proven topologies rather than to try to make intelligent strong modifications because they most likely just produce the few proven topologies that have been already developed by human designers.

4.1.3 Selecting From Proven Topologies

Selection from a library of proven topologies is a rather limited solution. Topology selection has to be made at the top decision level and additional design knowledge must put into the database each time a new circuit topology is added. However, the approach is efficient and robust for the following reasons:

- The approach most closely emulates a human designer's approach.
- The selected topology is robust - it is proven by thorough analysis and by its repeated usage.
- All the known useful properties and approximations specific to a certain topology can be fully utilized in this approach.
- The poor judgement made at the top decision level can also be quickly recovered because a typical run time of the automatic synthesis process is in the order of a few minute; whenever the current architecture cannot meet a certain specification, another architecture can be considered, and the synthesis can thus be carried out for a several

potentially suitable topologies.

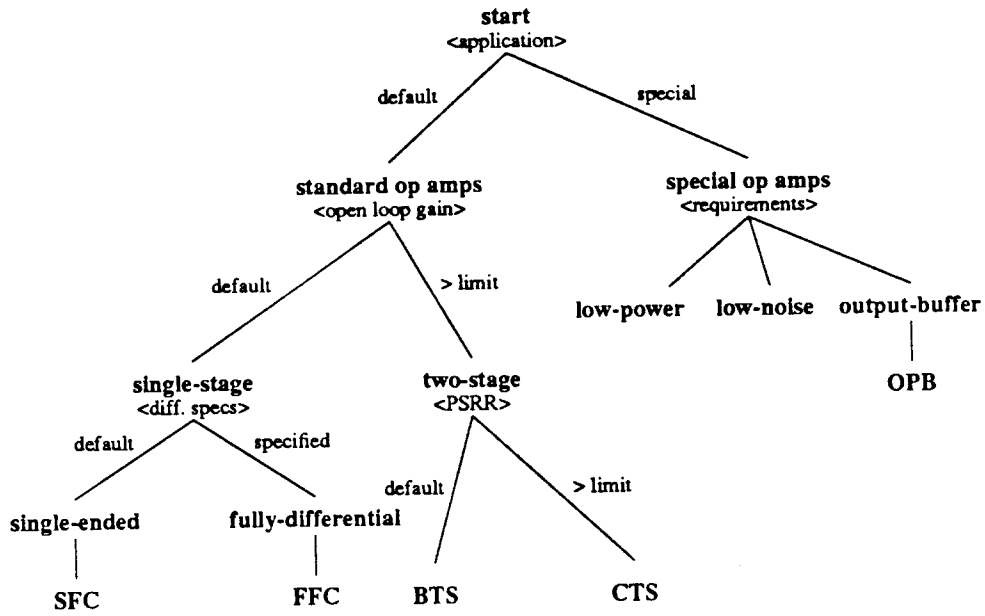
- The effort of adding a new topology to the database does not seem to be a problem because the number of proven op amp topologies that are typically used for analog interface subsystems is quite limited. For instance, a single-stage fully-differential op amp is almost everyone's choice for an A/D converter application. Furthermore, as symbolic analysis techniques are being developed,[6] such effort may become significantly smaller.

The heuristic circuit selection method used in OPASYN has been developed based on the above discussion. The detailed explanation of the method is given in the next section.

4.2 Heuristic Pruning of the Decision Tree

The circuit selection strategy used in OPASYN is based on a decision tree (shown in Fig. 4.3) which has been defined based on some key performance specifications such as general application area, open loop gain, PSRR, fully differential topology requirement, load capacitance, load resistance, power dissipation, or noise requirement. The leaf nodes in this tree correspond to proven op amp topologies commonly used in many applications, and the corresponding design knowledge is stored in the database of OPASYN. Currently five different circuit topologies have been fully incorporated and they are denoted in Fig. 4.3 as SFC, FFC, BTS, CTS and OPB. The meanings of various notations used in Fig. 4.3 are as follows:

- 'default' — represents the default choice.
- '> limit (< limit)' — the branch is to be selected if the corresponding design specification at the node is greater (less) than the limit value retrieved from the technology library.



default - default choice
> limit - select if the attribute value at the node is greater than the preset limit.
< limit - select if the attribute value at the node is less than the preset limit.
specified - select if it is so specified.
special - select if application needs a special op amp.

Figure 4.3 Decision tree for heuristic circuit selection.

- 'special' — the branch is to be selected if any of the special-purpose op amps is required.
- 'specified' — the branch is to be selected if it has been specified explicitly by the user.

Searching for a suitable topology starts at the root of the decision tree; nodes of this tree are checked in turn whether some subtrees can be pruned away (eliminated from further consideration) based on the range of the given specifications. The un-pruned leaf nodes are forwarded to the optimization module. The following example illustrates the basic idea.

Example:

Let's assume that the application demands an 'ordinary' op amp with an open loop gain of 10,000 and a PSRR of 60dB at dc using the MOSIS 3 μm process. The decision process starts at the top and immediately eliminates all the 'special-purpose' op amps since none of these special characteristics has been called for. Among the 'standard' op amps, the subtree of two-stage op amps is selected since the specified open loop gain of 10,000 is higher than the gain limit of single-stage op amps for the MOSIS 3 μm process (which is set to 5,000 in the technology library). Between the two two-stage op amps (BTS and CTS), the BTS op amp has been selected because the relaxed requirement on PSRR does not dictate the use of the CTS op amp. The particular decision path of this example is shown in Fig. 4.4 with bold arrows.

If the specification-based decision criteria are too vague, more than one leaf node may be returned, and the optimization process is carried out for all of these nodes. The user makes the final selection among various alternatives. By default, the system selects 'standard' op amps over 'special-purpose' op amps since they are simpler and smaller. If none of the 'standard' op amps can meet the specifications, then the user asks for a 'special-purpose' op amp with an explicit specification in the input file.

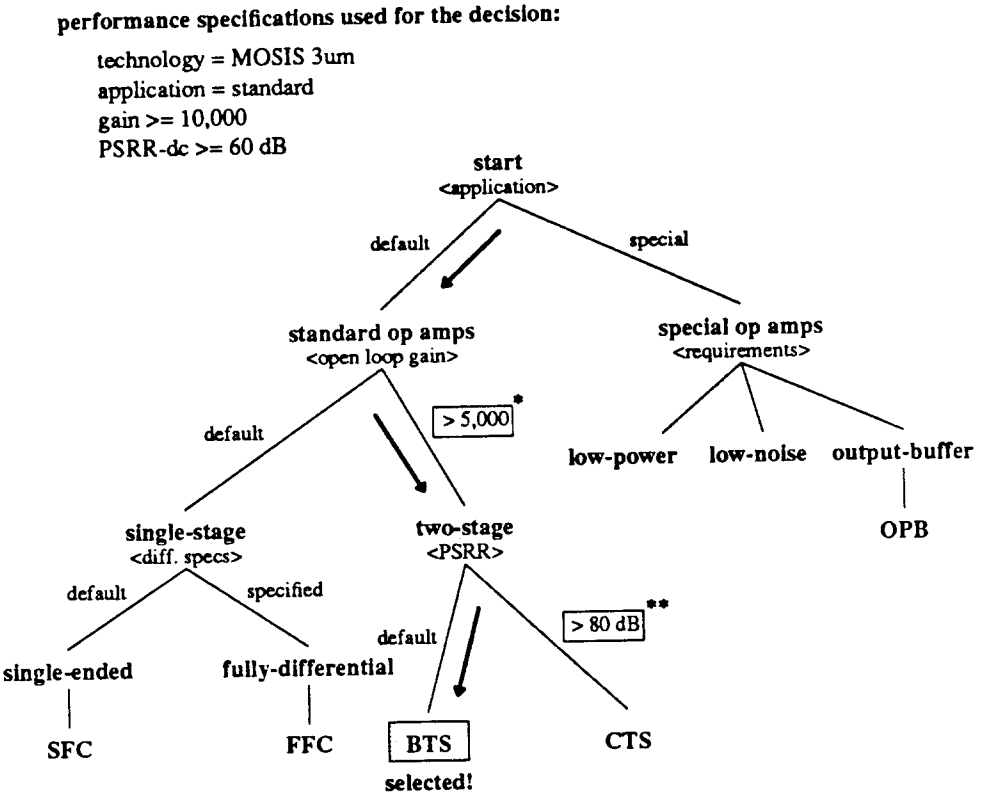


Figure 4.4 A circuit selection example.

Lastly, let's consider the theoretical background for how the decision tree shown in Fig. 4.3 has been derived. As discussed in Chapter 2, the following architectures are the ones frequently used for internal op amps:

- two-stage architecture
- single-stage architecture
- fully-differential architecture
- class AB architecture
- dynamic architecture.

In addition, even though it does not belong to the internal op amp category, an output buffer is frequently used for any application where a resistive load (2 kΩ or less) and/or a big capacitive load (over a few hundred picofarads) must be driven.[5]

The architectures mentioned in the above are not completely distinct from one another. A class AB two-stage op amp can be viewed as a variant of either the two-stage architecture or the class AB architecture. The FFC op amp used in OPASYN is a fully-differential single-stage op amp which can be again a variant of either the fully-differential architecture or the single-stage architecture. However, historically the fully-differential architecture, the class AB or dynamic architecture are introduced to improve certain performance characteristics of the two-stage and single-stage architectures. Therefore it can be assumed that those three architectures are variations of the two-stage and single-stage architectures. Based on this discussion, we can put the latter two architectures at the top of the decision tree.

On the other hand, some op amps are developed for specific purposes; an output buffer and a low-noise op amp are good examples of this. Such op amps can be directly chosen without going through the entire decision hierarchy. For instance, if an application requires to drive a resistive load of 1 kΩ, we can directly proceed to a group of output buffers and select one of them as required by other specifications. This observation has lead to the

notions of 'standard' and 'special-purpose' op amps which supersede the two-stage and single-stage architectures in the decision hierarchy.

Among the fully-differential, the class AB, and the dynamic architectures, priority has been given to the fully-differential architecture because the class AB and dynamic architectures are in fact variations in biasing schemes to achieve fast transient characteristics with less power consumption. Since the fully-differential architecture is primarily used for the single-stage architecture, it is not considered for the two-stage architecture.

The above discussions result in the decision tree illustrated in Fig. 4.3. At the top of the decision hierarchy, general application area divides op amps into two groups — 'standard' op amps and 'special-purpose' op amps. Then the standard op amps are further divided by their architectures — two-stage architecture and single-stage architecture. The single-stage op amps possess either the single-ended architecture or the fully-differential architecture. The PSRR characteristic becomes a branch criterion between the two alternative two-stage op amps — the BTS op amp and the CTS op amp. On the other hand, the 'special-purpose' op amps are divided by applications such as low-power, low-noise, and buffering applications. With this decision hierarchy, new op amps can be easily added into the decision hierarchy and obsolete ones, deleted. The decision heuristics (such as PSRR for two-stage op amps) can be also updated with minimal efforts; either the branching attributes of certain nodes are updated or new nodes are added as needed.

4.3 Implementation

4.3.1 Implementation Requirements

Just as there are many choices for the topology selection strategy, there are many ways to implement the chosen strategy. Since the adopted selection strategy, as explained in the previous section, is based on expert design knowledge developed over a long period of time and is rather heuristic in its nature, the following points must be considered for its implementation:

- Selection heuristics must be easy to update. Evolution of technology and innovation at circuit and architecture levels either make it necessary to add new topologies to the database or modify some of the existing decision heuristics.
- The program must be capable of explaining its decisions as human experts do. This is not only for detecting possible flaws in the decision heuristics but also for enhancing the user's confidence in the program's operation.
- The selection process must be straight-forward and fast — the program is meant to be a frontend for a module generator that can be used 'interactively' at the system design level.

4.3.2 A Rule-based Systems Implementation

To fulfill all the above requirements, a rule-based system's implementation has been adopted. Fig. 4.5 shows a typical structure of rule-based systems.[7,8,9,10] A rule base is different from a database in that a rule base is executable while a database can only be queried and updated. In fact, rules correspond to program codes in conventional software programs in the sense that rules are interpreted/executed by an inference engine whereas program codes by an interpreter or a compiler. An *inference engine* knows how to actively use the knowledge in the base. A *user interface* caters for smooth communication between the user

and the system, also providing the user with an insight into the problem-solving process carried out by the inference engine. Finally, a *rule-base* comprises the knowledge that is specific to the domain of application (op amp design in this case): such things as simple facts about the domain, rules that describe relations or phenomena in the domain, and possibly also methods, heuristics and ideas for solving problems in this domain.

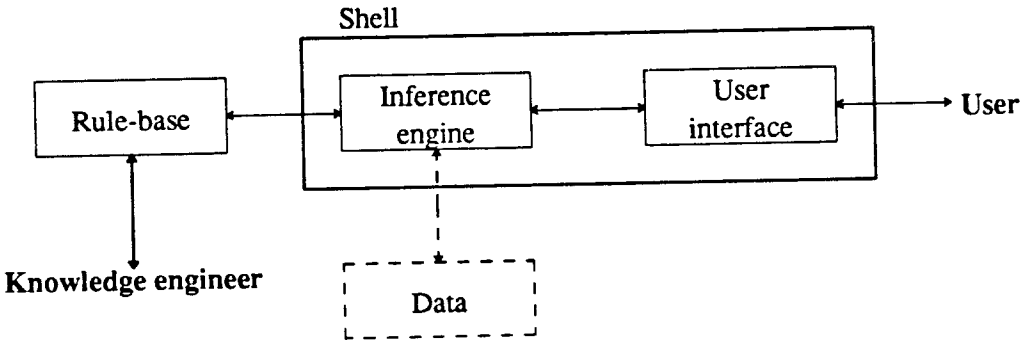


Figure 4.5 Structure of rule-based systems.

In the chosen implementation, the circuit selection heuristics explained in Section 4.2 is thus stored as a set of decision rules in a rule-base. In OPASYN, production rules have been used for representing decision rules. For instance, a decision rule (as expressed in OPASYN),

```
(rule identify-BTS
  (if ((> opamp) is standard)
     ((< opamp) is single-ended)
     ((< opamp) has gain   (restrict (> gain-lower-bound) gain-gt-p))
     ((< opamp) has psrrdc (restrict (> psrr-upper-bound) psrr-le-p)))
  (then ((< opamp) is BTS))
)
```

means that if an op amp is a 'standard'† op amp, is single-ended, has a gain greater than a certain limit (gain-lower-bound), and has a PSRR at dc less than a certain limit (psrr-upper-

bound), then choose the BTS op amp.

Then both modification and addition of the decision rules are quite convenient because the rules are not in the form of functional procedures embedded as a part of the program code segments but rather in the form of (executable) data. Only the rule-base, neither the inference engine nor the user interface, needs to be updated.

The inference mechanism used is a forward chaining method.[8,9] This uses the following strategy: when premise clauses (if-clauses) match the situation, then the conclusion clauses (then-clauses) are asserted. For example, in the rule of identify-BTS, if the performance specifications satisfy the premise, the BTS op amp will be selected as a proper op amp topology. The reason for adopting the forward chaining method over the backward chaining method is illustrated with the following; in the op amp selection problem, we try to select a particular op amp from a set of performance specifications and not vice versa. Therefore, given specifications are compared with premise clauses of the rules in the rule-base. Once a matching rule is fired, it will not be used again in the same search process; however, the fact concluded as the result of that rule's firing will be added to the rule-base as a derived fact. The same procedure is then repeated until we have reached a conclusion clause that identifies an op amp suitable for the given specifications or no more matching rules can be found. On the other hand, if we use the backward chaining method, a certain op amp is selected based on some 'simple heuristics' and the inference engine checks whether all of its premise clauses are satisfied by the given specifications. If not, another op amp is selected and re-tried (this process is called 'backtracking'). However, this process is contrary to what we are trying to do! This method is thus not suitable for implementing our selection strategy.

---

† An application area of an op amp can be either specified as 'standard' by the user or deduced as results of applying identification rules.

The user interface incorporated into the circuit selection module was designed to make the decision-making process clear to the user and to assist the process of debugging new rules introduced into the rule-base. It communicates with the user through such query processes as 'how' and 'why'. How-questions are used to query how the current conclusion has been reached; why-questions reveal why a certain premise has been tested. Such query processes are shown in Fig. 4.6. In the figure, a part of the performance specification file is shown in part (a). Then a circuit selection example session is illustrated in part (b). The program first selects the SFC (single-stage folded cascode) op amp based on the given specifications. The user then queries 'how' the SFC op amp has been selected. The program replies to the question by displaying the corresponding identifying rule. The user again asks 'why' the shape specification 'single-ended' has been used during the decision process, and so forth.

An output from the circuit selection module described so far is a circuit topology suitable for a set of performance requirements specified by the user. Once such a topology is selected, the problem of schematic synthesis reduces to that of parametric optimization, which is the subject of the next chapter.

(a) Performance specifications used in the decision process.

```
(APPLICATION = standard) ;; Standard application area
(SHAPE = single-ended)   ;; Single-ended architecture
(Vdd = 2.5)
(Vss = -2.5)
(CL = 5E-12)
(PWR-DIS < 1E-3)
(GAIN > 3000)
(PSRR-dc > 60)
(TECH = mosis3u)      ;; Technology is MOSIS-3um process
```

(b) An example of an interactive session.

```
% frontend              ;; Start a selector!

Lisp> (load "production")
    input file? << amp.specs     ;; An input specification file.

Lisp> (forward-chain)            ;; Select an appropriate op amp circuit.
(RULE IDENTIFY3 SAYS OPAMP IS SFC) ;; An op amp 'SFC' has been selected.

Lisp> (how '(opamp is sfc))      ;; How 'SFC' has been selected?
(OPAMP IS SFC DEMONSTRATED BY)
 (OPAMP HAS GAIN 3000)
 (OPAMP IS SINGLE-ENDED)
 (OPAMP IS STANDARD)

Lisp> (why '(opamp is single-ended))   ;; Why s-ended has been used?
(OPAMP IS SINGLE-ENDED IS NEEDED TO SHOW)
 (OPAMP IS SFC)

Lisp> (usedp 'identify-BTS)      ;; Was the rule 'identify-BTS' used?
NO, Rule IDENTIFY-BTS has NOT been used:

Lisp> (usedp 'identify-SFC)      ;; Was the rule 'identify-SFC' used?
YES, Rule IDENTIFY-SFC has been used:
 IF:
   (OPAMP HAS GAIN 3000) and
   (OPAMP IS SINGLE-ENDED) and
   (OPAMP IS STANDARD)
 THEN:
   (OPAMP IS SFC)
```

Figure 4.6 A circuit selection example.

CHAPTER 5

PARAMETRIC CIRCUIT OPTIMIZATION

5.1 Sizing a Circuit Schematic

In circuit design problems, once a promising circuit schematic (topology) has been selected, the synthesis problem is reduced to a task of parametric component optimization. A set of values need to be found for all design parameters that together optimize some performance attributes while meeting all explicitly stated specifications. This multiple-objective, constrained optimization task can be formulated into a standard mathematical programming problem as follows:[16]

$$\min\left\{ f(p) \mid g(p) \le 0 \right\}$$

where $p \in R^n$ is the vector of design parameters, $f: R^n \rightarrow R^m$ is the objective function, and $g: R^n \rightarrow R^r$ is the constraint function. It is important to note that in standard mathematical programming problems, the functions f and g are explicit functions of p, while in circuit design problems, the dependence of these functions on the design parameter vector, p, is implicit through various circuit equations. Thus function evaluation may require in the worst case expensive circuit simulation.

For op amps, key design components are widths and lengths of MOSFET's, bias currents, and compensation capacitor values (if any). The performance characteristics of an op amp are then non-linear functions of these design components through circuit responses. For Instance, the open-loop dc gain of a two-stage op amp is determined by such circuit responses as transconductances (g_m) and output conductances (g_o) of certain transistors in the circuits, where g_m and g_o are in turn functions of device sizes and bias currents of the

transistors. Therefore, conventional optimization programs can be used to perform the task of sizing the design parameters in an op amp schematic if they are made to communicate with a circuit simulation program. Such efforts produced DELIGHT.SPICE[17] and ECSTASY[18] which have been used for a broad spectrum of IC design optimization problems including the op amp circuit sizing.

However, these general-purpose optimization programs have had only limited successes when applied to op amp synthesis. First, the extensive circuit simulation required in the inner loop of the optimization step has made the tools expensive in terms of CPU time. Second, their user interfaces are hard to customize for a specific application (i.e., for op amp synthesis); the user must possess detailed design knowledge for that application to provide the many pieces of information needed by the program. Such design knowledge is often critical for the effectiveness of these programs. For instance, widths and lengths of MOSFET's, bias currents, and compensation capacitor values are all legitimate candidates for the design parameters in a particular op amp schematic-sizing task. But when all possible design components are used as design parameters, the resulting design space is too large for any optimization technique. In fact, many of these component values cannot be adjusted independently. Sound design strategies link many of these values into groups that depend on just a few independent design parameters.

From the above discussion, it can be seen that if expert design knowledge is captured and stored in the program's database, it can be used not only to minimize the dimensionality of the optimization problem by choosing the minimal number of design parameters but also to estimate the performance of the circuit without running a circuit simulator in the inner loop of the optimization step. Such design knowledge also makes it possible for the user to successfully perform a synthesis task without detailed knowledge about the particular circuit to be optimized. The expert design knowledge (on op amps) stored in OPASYN has been acquired by applying first-order circuit analysis techniques and topology-specific

approximations on each of the proven op amp topologies handled by the program. The acquired knowledge is then transformed into analytic design equations than can be easily used by optimization algorithms. A set of such analytic design equations, sound a priori design decisions, as well as other properties of a circuit useful for its synthesis are the components of the 'analytic circuit models', discussed in the next section.

5.2 Analytic Circuit Models

5.2.1 Definition

The parametric optimization process in OPASYN relies on analytic models developed specifically for each of the op amp circuit topologies contained in the database.[24] An analytic model of a circuit is a set of analytic expressions representing sound a priori design decisions and macro behavior of the circuit. The analytic model for a given circuit typically includes:

- netlist description of the circuit,
- declaration of the independent design parameters,
- reasonable upper and lower bounds for the design parameter values,
- analytic design equations to express functional dependencies of the performance of the circuit on the design parameters.

In order to reduce the dimensionality of the search domain, one should define a minimal set of independent design parameters and set reasonable upper and lower bounds on their range. Our investigations have shown that using expert design knowledge and first-order circuit analyses, the number of independent design parameters of the circuit can be greatly reduced. In the basic two-stage (BTS) op amp shown in Fig. 5.1 which consists of 13 mosfets, a capacitor, and a current source, we have found that the number of design parameters can be

reduced to only five: $(W/L)_1$, $(W/L)_5$, $(W/L)_6$, C_f, and I_{bias}. Similarly, the fully-differential folded cascode op amp (FFC) shown in Fig. 5.2 with over 60 devices has only 7 independent design parameters.[55] The upper and lower bounds for the design parameter values can be set to further restrict the search space. For instance, the compensation capacitor value in the BTS op amp has been limited to the range from 2 pF to 20 pF since we know that values outside this range are not useful in practical applications.

The analytic design equations were derived by using first-order circuit analysis techniques and topology-specific approximations.[6,7,29,56,57,33,58] For most dc characteristics, these computed approximations are excellent. For highly non-linear specifications such as small signal dc gain, phase margin, and settling time of the circuit, fitting parameters have been introduced to obtain more accurate prediction of specific performance characteristics. Here is an example of such analytic design equations:

$$a_v = cf_{gain} \times \frac{g_{m2}\, g_{m6}}{(g_{o2} + g_{o4})(g_{o6} + g_{o7})}$$

$$g_m = \sqrt{2\, K_p \frac{W}{L} I_d}$$

$$g_o = \lambda\, I_d$$

where a_v, is the small signal dc gain of the BTS op amp shown in Fig. 5.1, g_m is a transconductance, g_o is an output conductance of a transistor, and cf_{gain} is a fitting parameter. All the conductances are dependent on transistor sizes and bias currents. The fitting parameters are being updated as the system acquires more information from repeated synthesis and verification steps.

Analytic models can be developed not only for op amps but also for any type of analog functional blocks such as comparators, bandgap references, and output buffers. However, to derive an analytic model for these circuits often demands the attention of 'good' analog circuit designers. From our experience, the one time design effort to add a fairly complicated

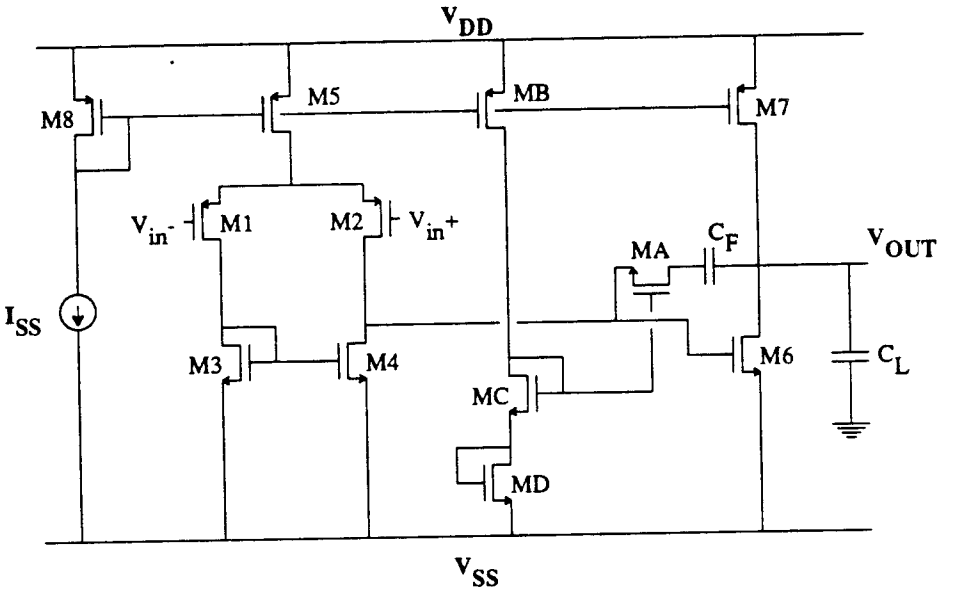
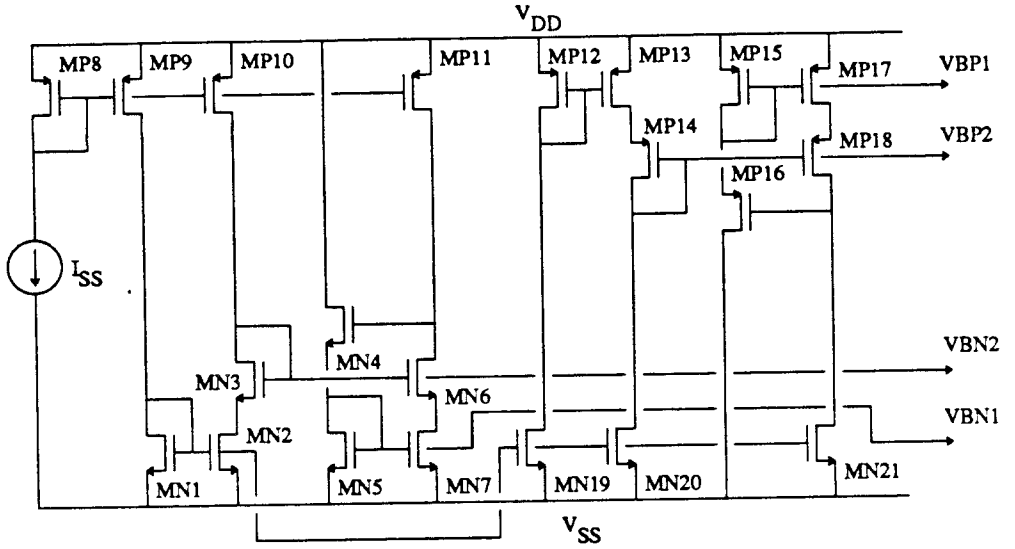
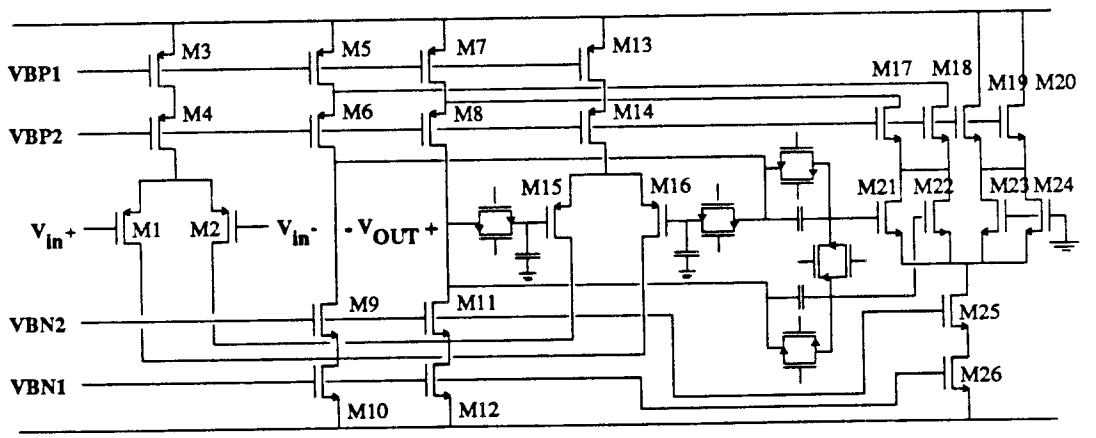


Figure 5.1 Basic two-stage (BTS) op amp.



(a) Bias circuit



(b) Amplifier

Figure 5.2 Fully-differential folded cascode (FFC) op amp.

op amp topology such as the FFC op amp (shown in Fig. 5.2) is about a couple of weeks, while it requires only a few days for simpler circuits. Recently, a symbolic analysis tool has been developed to generate analytic design equations for analog circuits.[50] As symbolic circuit analysis techniques are being advanced, this effort may become significantly smaller.

5.2.2 Model Acquisition

Given a circuit topology, the first step towards the acquisition of an analytic model of the circuit is to make 'sound' a priori design decisions; this means to choose a minimal number of design parameters and to set up suitable upper and lower bounds for them. These decisions, however, must be based on design experience. Many articles and books as well as expert designers have been used as sources of reference and insight. Analytic design equations have then been constructed from detailed circuit analyses based on first-order circuit analysis techniques and topology-specific approximations.

In summary, the model acquisition steps that have been taken in OPASYN are as follows:

step 1: Select independent design parameters by finding relevant relations between design components based on the first-order circuit analysis. Journal articles, books, and interviews with expert designers can help further reducing the number of independent design parameters.

step 2: Upon selecting a set of independent design parameters, set the proper upper and lower bounds for those parameters interviewing expert designers and referring to the previous design examples.

step 3: Express performance characteristics of the circuit in terms of the design parameters chosen in step 1. Transform the derived expressions into analytic design equations that can be easily used by optimization algorithms — continuity, differentiability, linearity, etc of these equations

are important for choosing a specific optimization algorithm.

5.2.3 An Example — A Basic Two-Stage OP Amp

The detailed model acquisition process for the BTS op amp shown in Fig. 5.1 is presented in this sub-section. As summarized in the previous sub-section, the acquisition process divides into three phases: design parameter selection, parameter range setting, and derivation of analytic design equations.

5.2.3.1 Design Parameter Selection

The choice of a set of independent design parameters should take into account:

- device matching
- biasing
- systematic offset voltage
- slew rate
- compensation (pole-zero cancellation)
- input-referred 1/f noise.

The input devices M1 and M2 should be equal in every respect; thus:

$$\left(\frac{W}{L}\right)_1 = \left(\frac{W}{L}\right)_2 \quad \text{where } L_1 = L_2 \tag{5.2.3.1a}$$

To minimize the systematic offset voltage, choose:

$$\left(\frac{W}{L}\right)_3 = \left(\frac{W}{L}\right)_5 \quad \text{where } L_3 = L_4 \tag{5.2.3.1b}$$

$$\frac{1}{2}\frac{\left(\frac{W}{L}\right)_5}{\left(\frac{W}{L}\right)_7} = \frac{\left(\frac{W}{L}\right)_4}{\left(\frac{W}{L}\right)_6} \quad \text{where } L_4 = L_6 \text{ and } L_5 = L_7. \tag{5.2.3.1c}$$

To have the same slew rate for both the positive and negative output voltage swings, set:

$$(\frac{W}{L})_7 = \frac{C_L + C_F}{C_F} (\frac{W}{L})_5 \qquad (5.2.3.1.d)$$

To meet the pole-zero canceling compensation requirement, make:

$$(\frac{W}{L})_c = 4 \qquad (5.2.3.1.e)$$

$$(\frac{W}{L})_b = (\frac{W}{L})_d = \frac{(\frac{W}{L})_6}{8} \qquad (5.2.3.1.f)$$

$$(\frac{W}{L})_a = \sqrt{8 (\frac{W}{L})_6 (\frac{W}{L})_c \frac{C_F}{C_L + C_F}} \qquad (5.2.3.1.g)$$

where $L_c = L_d = L_{min}$ (L_{min} is a minimum channel length) and $L_b = L_5$.

From equations (5.3.2.1a) to (5.3.2.1g), the following set of independent design parameters emerges:

W_1, W_5, W_6, C_F, I_5, L_5, L_6, and L_1.

Among these parameters, the channel length of the input transistors (L_1) are typically set by the designer. Accordingly, L_1 is made to be specified in the input file. The channel lengths of the NMOS transistors are conventionally set to 2 L_{min} and those of the PMOS transistors to 1.5 L_{min} because of the different noise characteristics of NMOS and PMOS transistors. As a result, only five independent parameters are left:

W_1, W_5, W_6, C_F, and I_5.

Without applying some expert design knowledge, the number of design parameters would be twenty five: twenty two of them come from the 11 MOSFET's, one from I_5, and one from C_F. Thus by using good design practices, a large reduction in the dimensions of the design space has been achieved. Since the (W/L) ratios instead of W's are frequently used in most of the design equations, we use (W/L) ratios as design parameters. From this point on, a (W/L) ratio is noted as 'A' (Aspect Ratio) and the bias current of the MOSFET M5 is named

'I_{SS}'. With these conventions, the independent design parameters of the BTS op amp in Fig. 5.1 are

$$A_1,\ A_5,\ A_6,\ C_F,\ I_{SS} \quad \text{where } A_i \equiv \left(\frac{W}{L}\right)_i. \tag{5.2.1}$$

5.2.3.2 Parameter Range Setting

The independent designer parameters selected in the previous sub-section must be given 'reasonable' upper and lower bounds to further limit the design space. However, this task also strongly relies on design experience because the design parameters can vary over a wide range of values depending on the application. Reasonable upper and lower bounds for each design parameter have been obtained from interviews with expert designers and these are the limits established for the BTS op amp:

$$50 \le A_1 \le 200 \tag{5.2.3.2a}$$

$$10 \le A_5 \le 200 \tag{5.2.3.2b}$$

$$50 \le A_6 \le 500 \tag{5.2.3.2c}$$

$$10\ \mu A \le I_{SS} \le 50\ \mu A \tag{5.2.3.2d}$$

$$2\ pF \le C_F \le 20\ pF \tag{5.2.3.2e}$$

The (W/L) ratio (noted as A_1) of the transistor M1 is set to vary from 50 to 200 because the input-referred 1/f noise is inversely proportional to the channel area (W * L) of M1 (thus the lower bound is set to 50) and the open loop gain is proportional to A_i (thus the upper bound is set to 200). In case of M5 which provides a bias current and limits the common-mode input voltage swing, the lower bound is set to 10 and the upper bound to 200. For the output pull-up transistor M6, A_6 changes from 50 to 500 to provide high current driving capability whenever necessary. The lower bound for the bias current I_{SS} is set to 10 μA because the BTS op amp is not intended for a low-power application. The 50 μA range is determined from the observation that the typical bias current range using 3 μm processes is around 20 μA. Finally, the range of the compensation capacitor C_F is set to from 2 to 20 pF because

the BTS op amp is normally used as an internal operational amplifier that drives small capacitive loads as in switched capacitor filters. The parameter ranges set in the above equations are wide enough for most standard applications but nevertheless they result in a substantial reduction in the search space and make the optimization process quite efficient.

5.2.3.3 Derivation of Analytic Design Equations

The analytic design equations characterizing the dc performance of the circuit can be derived from Fig. 5.1 simply applying the first-order circuit analysis techniques under the assumption that all the transistors in the circuit (except Ma) must be kept in the saturation region. It should be mentioned here that we have used the term 'first-order analysis' because we have used the first-order circuit models for both a transistor and other passive circuit elements where the second-order effects are ignored.

Power Dissipation

From Fig. 5.1 the total power dissipation is expressed as

$$(V_{DD} + |V_{SS}|) (I_{SS} + I_7 + I_b)$$

where V_{dd} and V_{SS} are positive and negative power supply voltages and I_7 and I_b are drain currents of M7 and Mb (both are positive). In the following equations I_i denotes the drain current of the transistor Mi.

Common-mode Input Voltage Limits

The common-mode input voltage swing is limited by the transistors M5 (high swing) and M3 (low swing).

$$V_{ic,h} = V_{DD} - |V_{DSAT,5}| - |V_{TP,1}| - V_{DSAT,1}$$
$$V_{ic,l} = V_{SS} + V_{TP,1} + V_{DSAT,3} + V_{TN,3}$$
$$V_{DSAT,i} = \sqrt{\frac{2 I_i}{k_{p,i} A_i}}$$

where $V_{TP,i}$ and $V_{TN,i}$ are the threshold voltages of the p-channel and n-channel transistors, $V_{DSAT,i}$ is the drain to source voltage at the saturation edge, k_p is the transconductance parameter of the transistor, and A_i is the W/L ratio of the transistor. In the above equations and all the other equations that follow, the sub-script 'i' denotes a particular transistor Mi.

Output Voltage Limits

The output voltage swings are limited by the transistors M6 and M7.

$$V_{o,h} = V_{DD} - |V_{DSAT,7}|$$
$$V_{o,l} = V_{SS} + V_{DSAT,6}$$

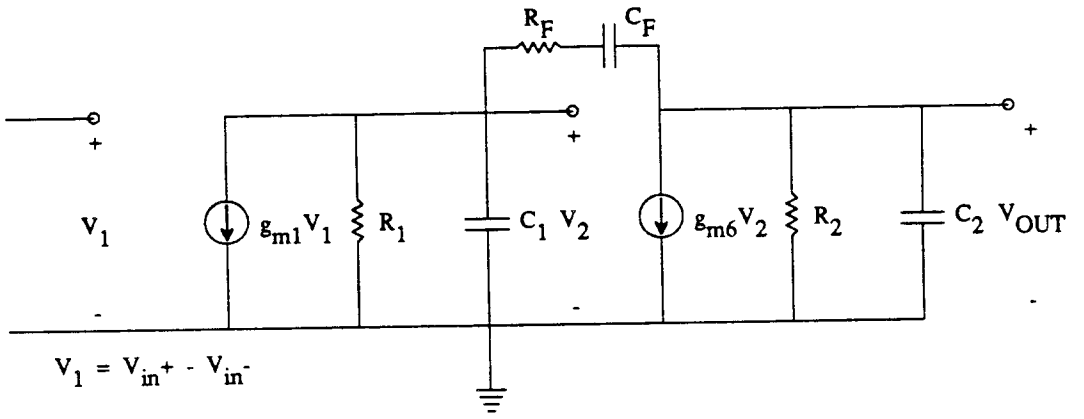


Figure 5.3 Small-signal equivalent circuit for the BTS op amp.

The ac performance characteristics of the circuit can be more easily derived from Fig. 5.3 rather than from Fig. 5.1. The circuit in Fig. 5.3 has been obtained by applying topology-specific approximations to the original circuit to reduce the complexity of the circuit and thus that of the first-order circuit analyses. As expressed in Fig. 5.3, the BTS op amp consists an input stage, an output stage and a compensation stage each of which is a relatively simple circuit using two or three basic elements such as resistors, capacitors, and

voltage dependent current sources. Any of the well-known circuit analysis techniques can be effectively applied to Fig. 5.3.

Open-loop Voltage Gain

At dc, all the capacitances in Fig. 5.3 are open-circuited. Therefore, the gain of the amplifier is the product of the gains of the input and the output stages.

$$a_v = \frac{g_{m1}}{g_{o2} + g_{o4}} \frac{g_{m6}}{g_{o6} + g_{o7}}$$

$$g_{m,i} = \sqrt{2\, k_p\, A_i\, I_i}$$

$$g_{o,i} = I_i\, \lambda_o\, (\frac{L_o}{L_i})$$

where g_{mi} is the transconductance and g_{oi} is the output conductance of the transistor Mi, λ_o is the channel length modulation factor, L_o is the default channel length specified by the technology and L_i is the channel length of the transistor Mi.

Unity Gain Bandwidth

The unity gain bandwidth is derived from Fig. 5.3 assuming pole-zero compensation.

$$\omega_u = \frac{g_{m,2}}{C_F}$$

Phase Margin

The phase margin is derived from Fig. 5.3 assuming pole-zero compensation.

$$p_m = \frac{\pi}{2} - \tan^{-1}(\frac{\omega_u}{p_3})$$

$$p_3 = \frac{1}{R_F\, C_1}$$

$$C_1 = C_{gs,6} + C_{gd,6}$$

$$C_{gs,5} = \frac{2}{3} C_{ox} W_6 L_6$$

$$C_{gd,6} = C_{gdo,n}W_6 + C_{gdo,p}W_2$$

where $C_{gs,i}$ is the gate to source oxide capacitance, $C_{gdo,n}$ and $C_{gdo,p}$ are the gate to drain overlap capacitances of the n-channel and p-channel transistors, and C_{ox} is the capacitance of the gate oxide per unit area.

PSRR (Power Supply Rejection Ratio)

The power supply rejection ratio is defined as

$$\frac{\text{open loop gain of input signal}}{\text{open loop gain of power supply noise}}.$$

The PSRR at dc can be derived from the open loop gain and the fact that the pull-up and pull-down transistors M7 and M6 at dc can be considered resistors connected in series. For ac signals, the PSRR of the negative power supply V_{SS} degrades as the same rate as the open loop gain decreases with frequency. The above discussions yield the following equations.

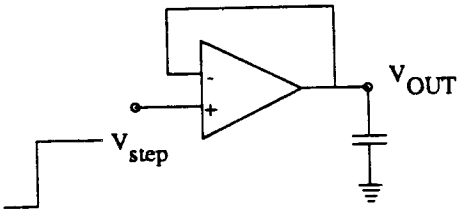
$$PSRR_{dc} = 20\log_{10}(a_v) - 20\log_{10}\left(\frac{g_{ds,6}}{g_{ds,6} + g_{ds,7}}\right)$$

$$PSRR_{ac} = PSRR_{dc} - 20\log_{10}\left[\frac{2\pi f g_{m,6}C_F}{(g_{o,2} + g_{o,4})(g_{o,7} + g_{o,7})}\right]$$

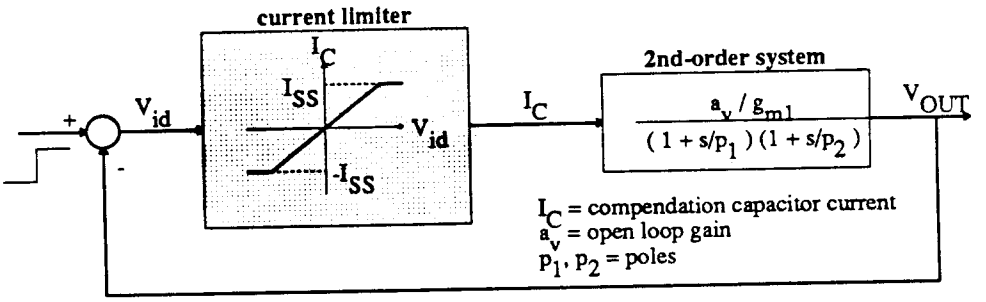
Slew Rate

The transient response of the circuit is derived when the op amp is in the unity gain feedback configuration as shown in Fig. 5.4(a). The s-domain diagram in Fig. 5.4(b) emphasizes the fact that the maximum current available in the input stage upon a large-signal input is I_{SS}. Therefore, the loop cannot be closed until the output voltage reaches I_{SS}/g_{m1}. During that period, the op amp output increases linearly and is said to 'slew'. From an open-loop analysis of Fig. 5.4, the slew rate is given as

$$SR = \frac{I_{SS}}{C_F}.$$



(a) Circuit configuration for transient analyses.



(b) Systematic diagram for (a).

Figure 5.4 Transient analyses for the BTS op amp.

Settling Time

After the loop is closed, the output follows the input and it settles at the input voltage. The time needed for the output to reach the input voltage is called 'settling time'. Chuang[56] published the analytic expressions for calculating a settling time of a system with two poles and one zero. However, experiments have shown that these expressions do not predict the right settling time in most cases. This is because the calculated pole and zero locations are not exact; they are derived based on first-order approximations. The other reason is that a doublet coming from an incomplete compensation of a pole and a zero degrades the settling behavior.[57] Accordingly, human designers estimate the settling time of an op amp as some multiple of the time constant given by the load capacitor and the transconductance of the pull-down transistor ($\frac{C_L}{g_{m,6}}$). A typical factor is 8 for settling within 0.1% of the input voltage. Since the total settling time is the sum of the slewing time and the settling time after the feedback loop is closed, the total settling time is expressed as

$$TS = \frac{V_{step} - I_{SS}/g_{m,1}}{SR} + a\left(\frac{C_L}{g_{m,6}}\right) + b\,.$$

where V_{step} is a step voltage applied at the input and a and b are fitting parameters that are determined from repeated comparisons between the results from the above expression and those from SPICE simulation.

Input-Referred 1/f Noise

The input-referred noise of the BTS op amp is given by[7]

$$V_{1/f}^2 = \frac{2\,K_p}{W_1\,L_1\,C_{ox}}\left[1 + \frac{K_n\,\mu_n\,L_1^2}{K_p\,\mu_p\,L_3^2}\right]\left[\frac{\delta f}{f}\right].$$

where K_n and K_p are the flicker noise coefficients for the n-channel and p-channel transistors, respectively, f is frequency, and μ_n and u_p are the mobilities of the electron and hole.

Total Gate Area

The total gate area of the circuit is

$$A_{gate} = \sum W_i L_i + \text{Area}(C_F).$$

This is not the actual area required for laying out the circuit. However, it provides a relative measure of the layout area since the routing area for a particular circuit topology remains rather constant as sizes of the devices change.

5.3 Parameter Space

5.3.1 Formation of Parameter Space

Based on the analytic design equations and the user-defined design targets, a total cost (C) representing a relative figure-of-merit for any particular combination of design parameter values is computed as follows:

$$C = \sum_{1}^{n} f_p\left[\frac{w_i\left(spec_i - p_i \right)}{spec_i} \right] \qquad (5.3.1.1)$$

where n is the number of circuit performance parameters considered in the program, p_i is the i-th performance parameter, w_i is a relative design priority of p_i, and $spec_i$ is the corresponding design specifications. The $f_p(\ )$ is an objective function that varies with the type of user constraints on the design specifications; we distinguish equality constraints (centering), inequality constraints (one-sided limits), and minimization (maximization) type constraints.

In the following, we consider each type of the user constraints to explain how the cost function defined in equation (5.3.1.1) has been chosen to handle it effectively.

Equality type constraints (centering) permit a user to express a desirable center value for some parameter such as a filter cutoff frequency. They typically originate from system-level

constraints rather than from design constraints of analog functional blocks. For many commonly used analog functional blocks such as op amps, voltage references, comparators, and output buffers, all the specifications are normally *inequality* (one-side limits) or *minimization/maximization* type constraints. Table 5.1 summarizes the specifications and their associated constraint types for the above functional blocks. However, as illustrated in Table 5.2, the passband cutoff frequency of a low-pass filter specification is an equality constraint; we do want to place the cutoff frequency at the specified location neither above nor below it. Therefore, the equality constraint is adopted in OPASYN for its future extension. The constraint is implemented using a quadratic objective function — $f_p(x) = x^2$. Therefore, as the circuit performance deviates from the specified value in either direction it penalizes the corresponding configuration. The quadratic objective function produces smooth search spaces where simple numerical optimization algorithms such as a steepest descent method works effectively.[59, 60]

Inequality type constraints (one-sided limits) are very common in most functional blocks as shown in Table 5.1. The inequality constraints express one-sided limits and are implemented using an exponential objective function — $f_p(x) = e^x$. In addition to producing smooth search spaces, the exponential function prevents the penalty for violating any specification from being compensated by overly satisfying other specifications.

Minimization (maximization) constraints arise in the following design situations:

(a) Minimize the power (or chip area) while satisfying all the other specifications — this type of constraint can be interpreted as design objective.

(b) Maximize the bandwidth and output swing while all the other performance characteristics assume 'practical' values — this type of constraint creates slacks for the other characteristics and often arises in output buffer designs.

(c) Maximize (minimize) the slew rate regardless of all other performance characteris-

functional block	performance specification	typical constraint type
op amp	open-loop gain	inequality/maximize
	settling time	inequality/minimize
	1/f noise	inequality/minimize
	PSRR	inequality/maximize
	power dissipation	inequality/minimize
	unity gain bandwidth	inequality/maximize
	die area	inequality/minimize
	phase margin	inequality/maximize
	slew rate	inequality/maximize
	output range	inequality/maximize
	common-mode input range	inequality/maximize
comparator	input drive	inequality/minimize
	response time	inequality/minimize
	power dissipation	inequality/minimize
voltage reference	temperature coefficient	inequality/minimize
	supply coefficient	inequality/minimize
	output voltage coefficient	inequality/minimize
	process coefficient	inequality/minimize
output buffer	bandwidth	inequality/maximize
	output range	inequality/maximize
	gain peaking	inequality/minimize

Table 5.1 Performance constraint types for typical analog functional blocks.

functional subsystem	performance specification	typical constraint type
low-pass filter	cutoff frequency	equality
	stopband corner frequency	inequality
	dc gain	inequality/maximize
	passband ripple	inequality/minimize
	stopband attenuation	inequality/maximize
A/D converter	bit resolution	inequality/maximize
	power dissipation	inequality/minimize
	monotonicity	inequality/maximize
	integral nonlinearity	inequality/minimize
	input noise	inequality/minimize
	conversion rate	inequality/maximize
	dc offset	inequality/minimize
D/A converter	bit resolution	inequality/maximize
	monotonicity	inequality/maximize
	integral nonlinearity	inequality/minimize
	gain drift	inequality/minimize
	conversion rate	inequality/maximize
	dc offset	inequality/minimize

Table 5.2 Performance constraint types for typical analog subsystems.

tics — this type of constraint is most straight forward but it never happens in practical design problems.

There are a couple of alternative approaches to handling the above situations: a *design objective* approach and a *priority* approach.

In the *design objective* approach,[17] we can express the desired performance characteristic with a maximization (minimization) type constraint as a design objective. Then all the other characteristics are treated as constraints and they are satisfied at the earlier stages of the optimization. For the characteristics with slacks as in (b), default values from the program can be used. The problem with this approach is that by meeting all the constraints the resulting design configuration is sometimes no longer a practical one; it may use a huge amount of chip area or power. The program in this case does not provide any practical 'estimate' about the capabilities of the given circuit. A useful program should be able to state that within certain limits on chip area and power consumption, the given circuit can only perform with certain bounds. The user can then decide whether some design specifications should be loosened or whether another circuit topology should be selected.

In the *priority* approach,[24] all the design specifications are considered simultaneously right from the beginning. If any performance characteristics need to be preferentially maximized (minimized), they can be given higher priorities over the rest of the desired characteristics. Again, un-specified characteristics take the default values. Using this approach, the program tries to meet all the specifications with different priorities and presents the best possible solution to the user. Three situations are possible:

(a) All the specifications are met, and the user is satisfied with the current values of the design objectives.

(b) All the specifications are met, but the user wants to further minimize (maximize) some design objectives.

(c) All the specifications are not met, and the user may or may not want to further minimize (maximize) the design objectives.

In case (a), the optimization is done. In case (b), the user may give higher priority values to the design objectives and run the optimization again or the program could automatically increase the priority value until some of the design specifications are violated. In case (c), the designer needs to relax some design requirements or switch to another circuit.

In OPASYN, the priority approach has been selected based on the above discussion and implemented as shown in equation (5.3.1.1).

5.3.2 Characteristics (Smoothness) of Parameter Space

The chosen cost function which is a sum of the quadratic and exponential functions yields smooth search spaces. Fig. 5.5 and Fig. 5.6 show some search space profiles for the BTS op amp and the FFC op amp respectively. The total design cost is plotted as one of the design parameters changes over a wide range of values while all the other design parameters are fixed at nominal design values. The profiles show that the search space is indeed very smooth. From those figures, it can be observed that the smoothness of the profiles are quite similar for both the relatively simple BTS op amp with 13 devices and the FFC op amp which is a quite complicated analog functional block with over 60 devices!! The smoothness of the solution space is a rather general phenomenon, as long as continuously differentiable cost functions are used.

5.4 Optimization Strategy

The search for an optimal solution starts with a coarse grid sampling through the bounded domain of all the independent design parameters. To avoid poor locally optimal solutions, a user specifiable number of grid points (typically about ten points) are randomly selected as starting points. A steepest descent algorithm then searches for the optimal

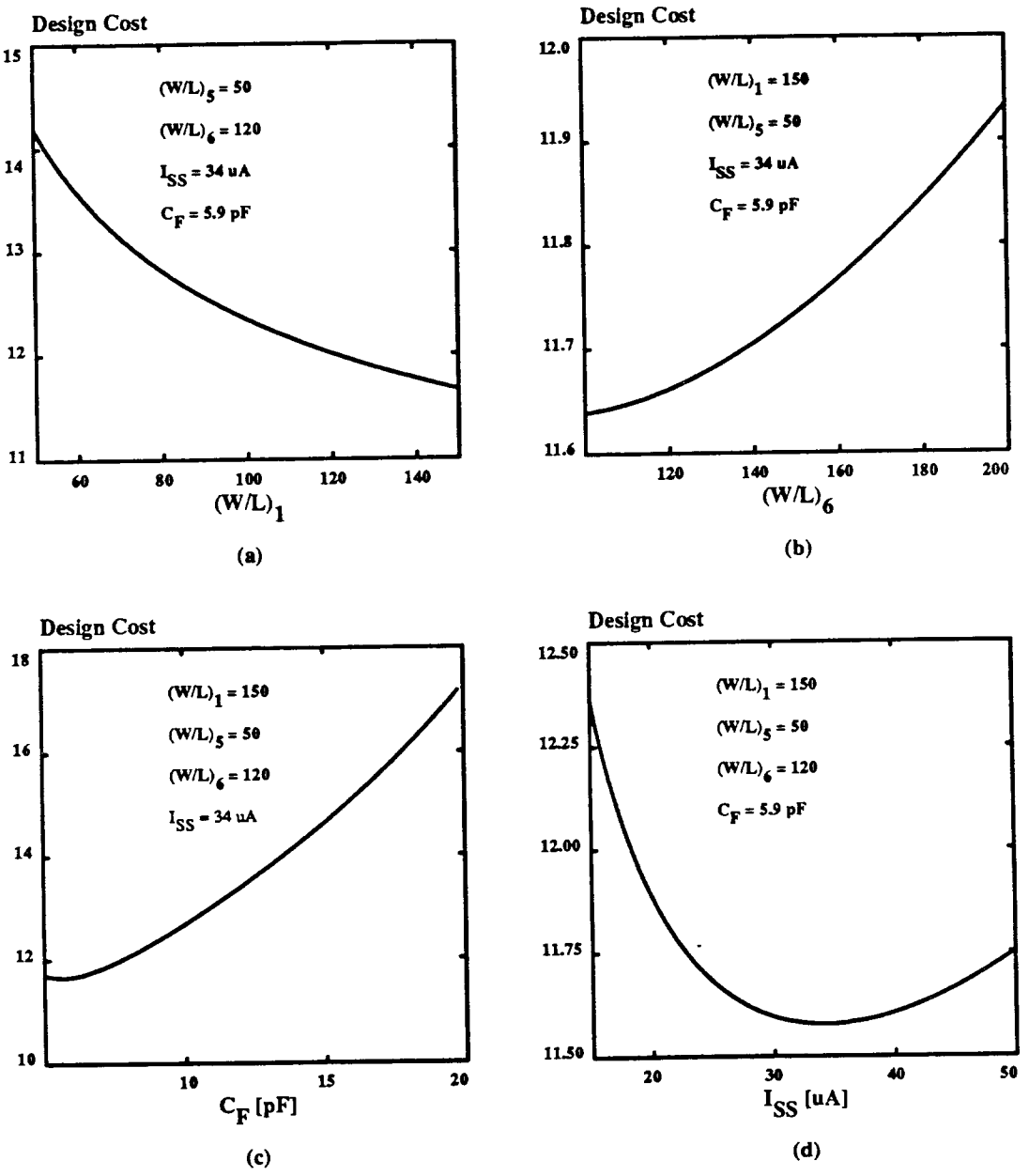


Figure 5.5 Search space profile for the BTS op amp.

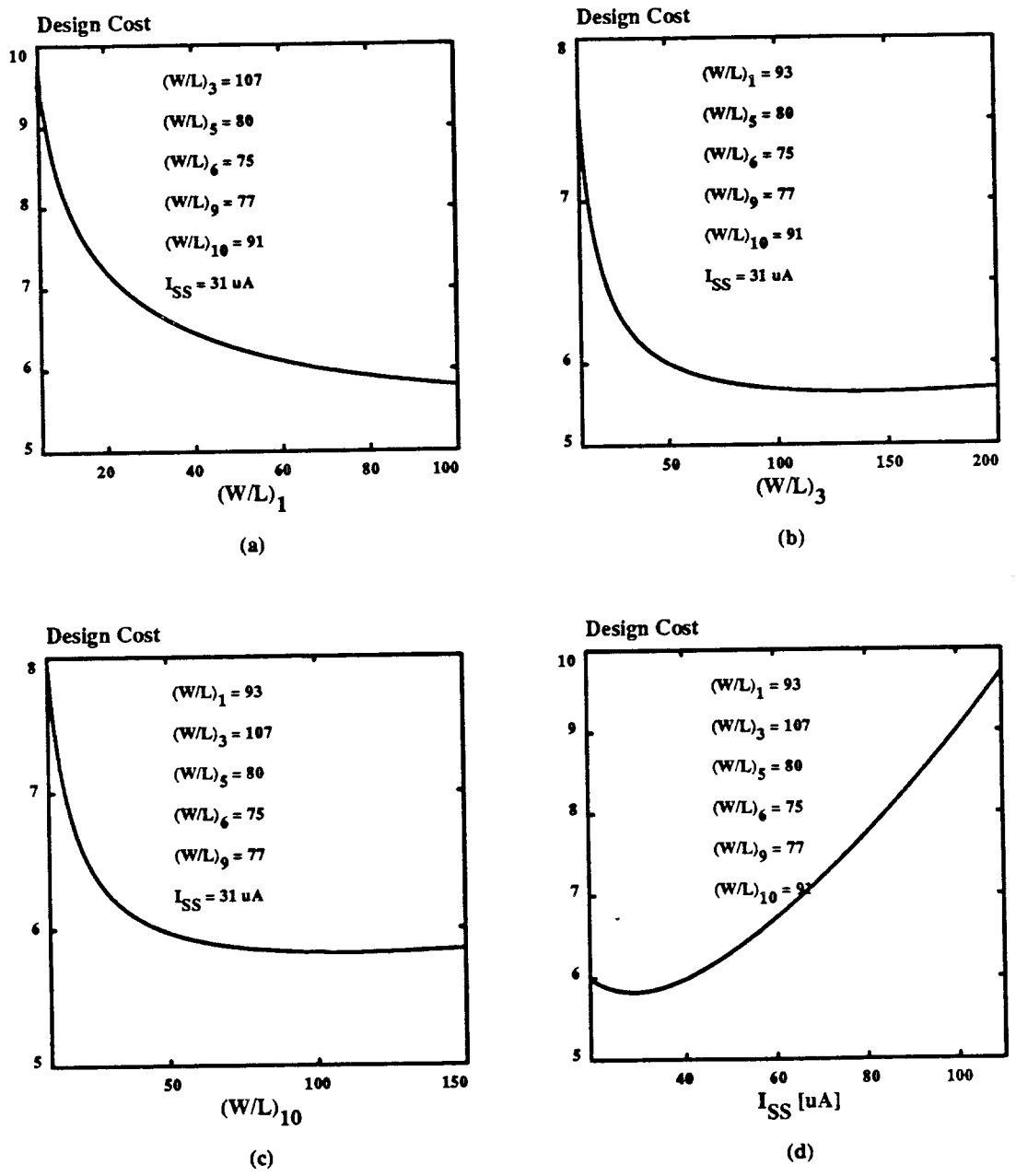


Figure 5.6 Search space profile for the FFC op amp.

solution in the neighborhood of each of these points. OPASYN returns all the different solutions (if any) that meet specifications to the user who finally decides which one to select. The optimization algorithm is summarized as follows:

Optimization Algorithm

Input:

- An analytic circuit model form the database
- Number of trials chosen by the user

Output:

- Promising design configurations

Algorithm:

- For specified number of trials do {
  Random selection of a starting point.
  Steepest descent from the starting point to local minimum.
  Save the design configuration found.
  }
- Return design configurations found.

The steepest descent method used is an enhanced version of a conventional method (see Fig. 5.7). A golden section search method[60] and a history heuristic have been added. The golden section search method eliminates the possibility of being trapped in a saddle point and increases search speed. The history heuristic used is to detect whether the search point is on a steep hill or inside a narrow valley and to appropriately adjust the initial leaping distance in the golden section search method. The steepest descent algorithm is summarized in Fig. 5.7.

In the enhanced steepest descent algorithm such factors as INIT-LMABDA and GOLD-SECTION, and the switch conditions for leap distance such as STEEP-HISTORY $\geq$ 10 and VALLEY-HISTORY $\geq$ 5 have been determined rather empirically from experiments. The initial leaping distance INIT-LAMBDA has been set to 5 units of each parameter considering the fact that the leaping unit of an (W/L) ratio of a transistor is set to 0.5, that of a

Steepest Descent Algorithm

Input:

$f(\mathbf{x})$ with $\mathbf{x}_{min} \le \mathbf{x} \le \mathbf{x}_{max}$.
A starting point $\mathbf{x}_s$.

Output:

A local optimal point $\mathbf{x}_{opt}$.

Algorithm:

```
Set λ to INIT-LAMBDA (= 5.0).
Set η to GOLD-SECTION (= 0.8).
Set STEEP-HISTORY and VALLEY-HISTORY to 0.
x_o = x_s
While ( |∇f| > END-FACTOR or stop≠true ) do {
    if ( STEEP-HISTORY ≥ 10 )      λ *= 3
    if ( VALEY-HISTORY ≥ 5 )       λ *= 0.3
    Get ∇f(x_o).
    x_1 = x_o − λ∇f(x_o)
    x_2 = x_o − ηλ∇f(x_o)
    if ( f(x_1) ≤ f(x_2) ) {
        STEEP-HISTORY += 1
        VALLEY-HISTORY = 0
    } end if
    else Do until ( f(x_1) ≤ f(x_2) ) {
        x_1 = x_2
        η = η^2
        if ( η <= ETA-LIMIT ) stop = true
        VALLEY-HISTORY += 1
        STEEP-HISTORY = 0
        x_2 = x_o − ηλ∇f(x_o)
    } end else
    x_o = x_1
} end while
Return x_o = x_opt.
```

Figure 5.7 An enhanced steepest descent algorithm.

bias current to 0.1 μA, and that of a compensation capacitor to 0.05 pF. Therefore, the maximum leaping distance is still too limited to jump from one valley to another valley. The END-FACTOR specifies the magnitude of the gradient at the present point. It can be specified by the user to improve the optimization quality. Another stopping criterion is that when the leaping factor (η) gets below the certain limit ETA-LIMIT — which means that the current search point is very close to a minimum point of a valley whose one side is a cliff, and therefore the search must be terminated.

The described optimization strategy is quite efficient (in all test cases CPU time was less than 10 seconds on a VAX 8800 with a reasonable end factor) and yet independent of any specific device technology or circuit topology to be used; different analytic models or technology files need to be provided for accommodating different types of circuits or process technologies.

5.5 Further Discussions

5.5.1 Fine Tuning the Analytic Models

Preparing an analytic model for a new circuit requires not only to create the proper expressions that describe the functional dependencies in the circuit but also to determine reasonable values for the various fitting parameters used in the expressions. This requires a substantial number of simulation runs and eventually circuit fabrications. However, it is quite difficult and impractical to fine-tune the fitting parameter values using discrepancies between the predictions and the results of actual fabrications. It is too slow and too expensive when technology is updated or when a new circuit is added to the database. The circuit performance predicted by OPASYN is compared with SPICE simulation results. To make the predictions as good as possible, the fitting parameter values are adjusted using zeroth- or first-order curve fitting techniques. Thus the predictions from the program can eventually

reach SPICE accuracy — even without using SPICE in the inner loop of the optimization step.

5.5.2 Other Optimization Methods

The steepest descent algorithm used is simple and efficient but it requires a differentiable search space with a continuous first derivative.[59] Furthermore, it provides only limited constraint handling capability, i.e., a set of specifications that has to be met in a limited subrange of some design parameters cannot be accommodated.

To make the search procedure more robust and to handle more complicated design requirements, a sophisticated method such as "The Phase I-II-III Method of Feasible Directions"[61] or "Successive Linear Approximation Algorithm"[62] can be used. However, the simplicity and the approximate nature of analytic models limit the effectiveness of such brute-force optimization methods — the predictions of the first-order analytic models used may deviate from the actual circuit response by as much as 20%. It appears that most design requirements for basic analog functional blocks (op amps, comparators, voltage reference, output buffers, etc.) can be accommodated by the steepest descent method.

CHAPTER 6

LAYOUT GENERATION

An efficient layout method for CMOS op amps is described in this chapter. It constructs design-rule-correct mask geometries of an op amp in macro cell layout style starting from the netlist of the circuit. Variable-shaped small circuit elements such as transistors, transistor pairs, and capacitors are produced by parameterized leaf cell generators and assembled in a bottom-up manner according to the structure of a circuit-dependent slicing tree that guarantees sound arrangements of the individual components generated. The advantage of flexible element shapes is fully exploited by using shape constraint relations[63] and Stockmeyer's algorithm.[64]

The layout divides into the three well-known phases: floorplanning, routing, and layout spacing. It makes use of many of the layout heuristics used by experienced designers in all three phases. Each of these phases will be explained in the following sections.

6.1 Layout Requirements for Analog Circuits

The performance of an analog circuit, unlike that of a digital circuit, can be degraded by even small (a few percents) deviations of voltages and currents from their nominal design values. In digital designs, circuits are typically designed to maintain a substantial amount of noise margin that prevents small deviations from affecting the performance of the circuit in any significant way. In the analog world, however, such immunity must be achieved by creating circuit topologies that are as insensitive as possible to small deviations from the nominal values of the design. This imposes some strict requirements on the circuit layout process. For instance, the input transistor pair of an op amp must be laid out in such a way

that external noise voltages affect both input leads of the pair in an equal way so that the net effect is reduced by the common-mode rejection ratio (CMRR) of this stage, whose value is typically around 80 dB (10,000). To achieve such good cancellation, both transistors must have exactly the same geometries and electrical properties. Therefore, producing a good layout for an analog circuit is more difficult than to lay out a digital circuit with the same number of components.

In the rest of this section, the layout requirements for analog circuits are itemized and each of them is discussed in detail. The requirements described in this section are then used as guidelines in such layout phases as cell generation, floorplanning, routing, and compaction. The guidelines given in the following, however, are only a small part of what is needed to produce well-functioning circuits; they concern those aspects of a good layout that expert designers can express in the form of explicit rules and expressions. An even larger fraction of their expert knowledge is more illusive and not easily expressed in concrete statements. In any case, the layout process for analog circuits demands a fair amount of experience and can be rather time-consuming.

Parasitic Side-Effects

Parasitic capacitance and resistance from devices and wirings often produce undesirable side effects. In analog functional blocks such as op amps, comparators, output buffer, and voltage references, the main sources of parasitic capacitance are large devices; the signal nets forming the internal connections of the block are reasonably short. Big devices such as output devices when driving large capacitance load can produce as much as a few tens of picofarad parasitic capacitance due to their bulk junction capacitances.

Parasitic resistance mostly comes from poly-silicon routing wires. For instance, if two source terminals of a differential pair are wired in poly-silicon material, the differential gain of the pair will be degraded because of the source regeneration effect[6] caused by the parasi-

tic resistance of the connecting wire. To minimize the parasitic side-effects, the following guidelines are typically observed:

- The source connection of a differential pair is routed in metal.
- 'Finger structures' (Fig. 6.1a) are used for large devices. Where possible, devices with common signal nodes share a single physical terminal region (see Fig. 6.1b).
- A large number of source and drain contacts are used to reduce parasitic resistance of these terminals.

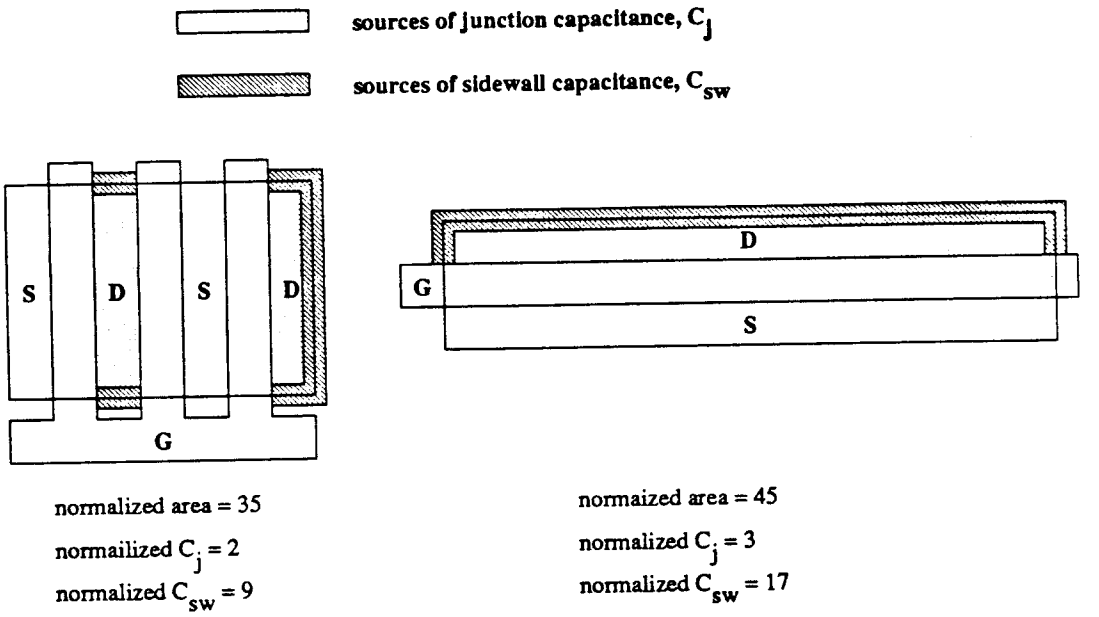
Random Process Variations

Widths and lengths of transistors, capacitance values, and device parameters (such as k_p, gate oxide thickness, etc) show some random variations caused by imperfect processing steps. For instance, an actual value of a capacitor after fabrication, can deviate as much as 20% from its nominal value in most of current MOS technologies. Therefore, certain groups of devices and routing wires are designed in the same shape in order to match their dependencies and sensitivities on random process variations. To minimize the effect of process variations on the circuit performance, the following guidelines are normally observed:

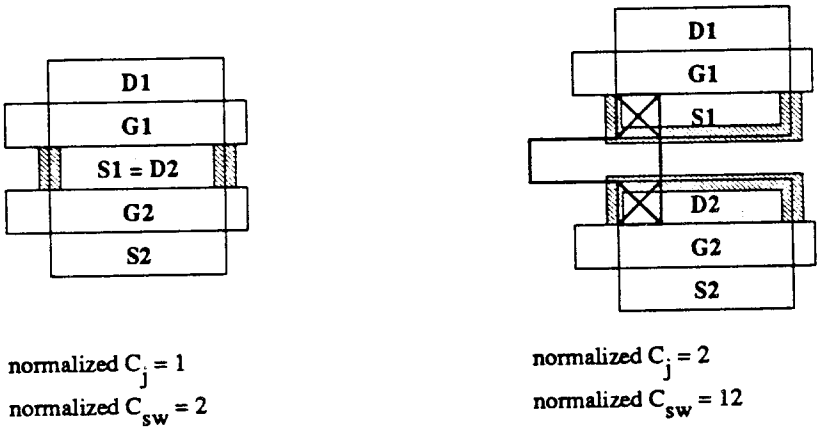
- The devices used in current mirrors are laid out with the same channel lengths.
- Differential pairs are designed with the same geometrical shape.
- The input leads of a differential pair are routed with the same geometrical pattern.

Process and Thermal Gradients

The process gradient may affect the performances of analog functional blocks if these circuits occupy large areas on the chip. In a BiCMOS or bipolar process, a bipolar output stage may produce thermal gradient over its neighbors. However, in a CMOS process the thermal gradient can be safely ignored. To minimize the effect of process and thermal gradients on the circuit performance, the following guidelines are typically observed:



(a) Finger structure vs. single-gate structure.



(b) Terminal sharing vs. explicit connection.

Figure 6.1 Finger structure and terminal sharing.

- The devices used in current mirrors are placed as close as possible.
- Differential pairs are placed next to each other.
- In BiCMOS or bipolar processes, the symmetry axis of a device pair should be oriented toward the heat source.

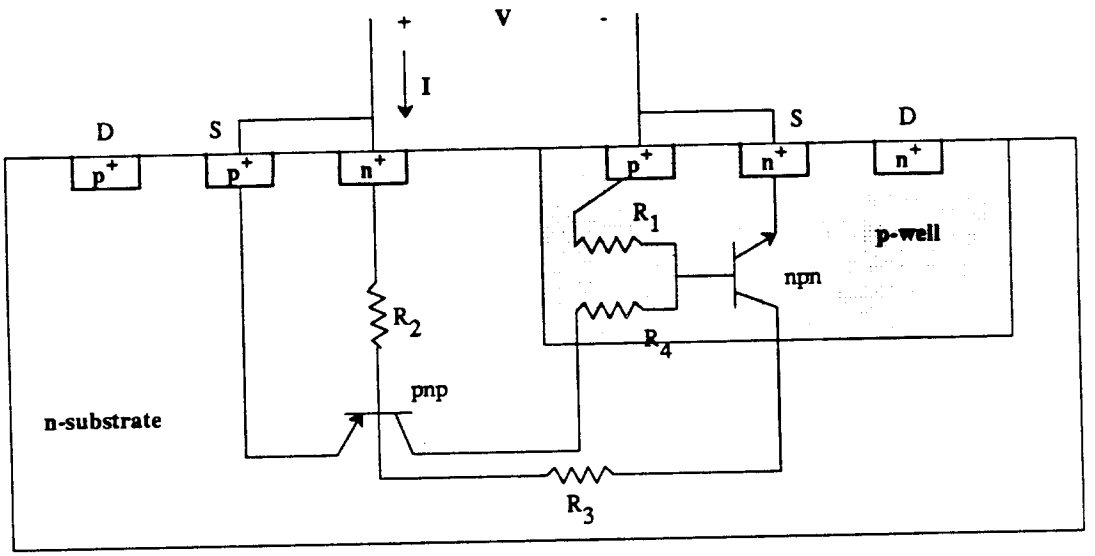
Noise Coupling

Undesirable noise couplings into critical signal paths should be avoided. For basic functional blocks such as op amps, comparators, etc, coupling between inputs and outputs (noise feedback), and coupling between analog signals and digital signals (clock signals in a comparator) are the major performance degradation factors. Following guideline are typically followed:

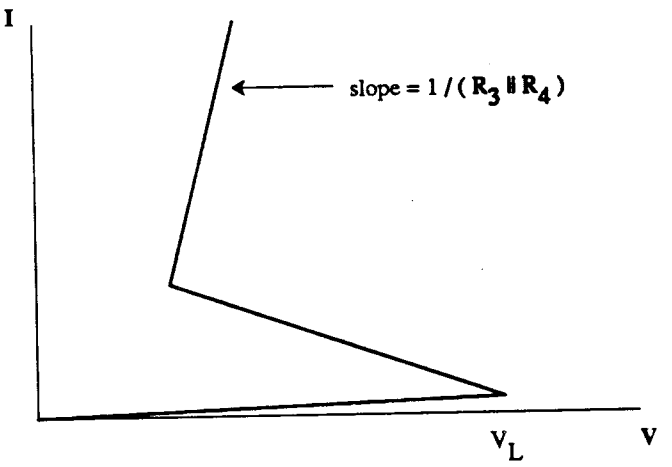
- Input signal nets are routed away from an output signal net. Otherwise, large changes in the output signal may be cross-coupled into the input and causes malfunctioning of the circuit.
- Clock signals in a comparator are routed in a separate channel and thus isolated from analog signals.
- Coupling on positive and negative inputs of op amps must be balanced to get cancellation by CMRR.

Latch-up

All CMOS transistor circuits have undesired and potentially trouble-some parasitic bipolar transistors which will conduct if one or more pn junctions become forward-biased.[65] The potential consequences of bipolar transistor action are much more serious in CMOS circuits. Fig. 6.2a makes it clear that a pnp transistor is possible with the n-type body as its base, while an npn transistor is possible with an n^+ source or drain electrode as its emitter, the pwell as its base, and the n body as a collector. With only rare exceptions, the undesired



(a) Origin of parasitic elements.



(b) IV characteristic of pnp-npn structure.

Figure 6.2 Latch-up in CMOS circuits.

parasitic circuit will be present in any CMOS circuit. The two-terminal current-voltage characteristics of this parasitic circuit are depicted in Fig. 6.2b. Above some critical voltage V_L (related to punch-through; 10 to 20 V for most modern CMOS circuits) both bipolar transistors begin to conduct and the current rises sharply from leakage levels (under 1 μA) to a value limited by resistors R_3 and R_4, often many miliamperes. To prevent the latch-up problem the following guideline must be followed:

- Enough (typically for every 100 μm or so for a 3 μm process) substrate and well contacts should be provided to reduce the resistors R_1 and R_2 in Fig. 6.2a.
- Wells can be protected from the latch-up by enclosing their peripheries with a guard ring.

6.2 Cell Generation for Analog Circuits

6.2.1 Cell Generators Versus Cell Libraries

In the semi-custom design of digital circuits, standard cell or macro cell libraries are frequently employed to provide instant access to highly optimized (both in area and performance) functional blocks. This design approach is possible because a finite set of library cells can cover most functional requirements that occur in digital circuitry. The same approach has been tried to speed up the semi-custom design of analog circuits. Pletersek[4] and Stone[3] undertook an effort to build analog standard cells and building blocks to be stored in the library. However, such efforts have achieved very limited successes.

To understand the limitations of this approach for analog circuit design, let's compare an inverter with an op amp, both of which are one of the most frequently used building blocks in digital and in analog circuitry, respectively. The main functional requirement for an inverter is to 'invert' its binary input within limited time delay, using a minimal amount

of cell area and power. These three values thus become its major specification parameters that need to be traded-off against one another. As a result, a few inverters with different driving capabilities and consequently different area and power requirements are pre-designed and stored in the library to be used for a wide variety of applications. On the other hand, an op amp whose main functional requirement is to amplify an input signal with a 'specific' gain (again with limited time delay, area, and power consumption), has many other specifications to meet: input range, output range, unity-gain frequency, input offset voltage, slew rate, 1/f and thermal noise, etc.† These, many functional requirements make it difficult for a few pre-designed op amps to cover a wide spectrum of applications.

Another challenging problem lies in maintaining the cell library. As technology scales down every two years or so, the performances of digital circuits in general improve, since delay time as well as power and area consumption decrease as the geometries are scaled down. Therefore, digital leaf cells stored in a library often do not have to be re-designed but simply need to be re-characterized with respect to their performances. However, this is not true for analog circuits. In a transconductance amplifier, as technology scales down by a factor of 'b', the thermal noise voltage increases by $b^{1/2}$, while supply voltages are scaled down by a factor of b. The result is that the dynamic range of the amplifier degrades by a factor of $b^{1.5}$. [33] As an another example, an input-referred 1/f noise power of an op amp is known to be inversely proportional to the product of the channel width and length of an input transistor. Therefore, as the geometry scales down by a factor of b, this noise power increases by a factor of b^2 which significantly degrades the dynamic range of an op amp at audio-frequency ranges. Such performance degradation is not tolerable for many high-performance applications and the amplifier design must be suitably modified. This further requires that the library cells must be laid out again to accommodate the necessary modifications.

† The way these factors affect the performance of an op amp has been a subject of a study called 'macro-modeling' and will be covered in chapter 8.

An op amp is one of most frequently used functional blocks in analog circuitry. For instance, 5 to 20 op amps are required in switched capacitor filters. An op amp is also used in algorithmic A/D converters to implement a finite gain block, or used in D/A converters to transform the current to the voltage. In all these applications, the performance of the op amps is critical to the overall performance of the system: the degradation in noise characteristic causes the narrower dynamic range of the filter, or quality of an A/D converter will be badly affected by an insufficient amplifier gain. As a result, it is very common to fine-tune an existing op amp design for a specific application, which also mandates that a new layout be generated.

To summarize, the usage of a cell library for semi-custom design of analog circuits is rather limited. A layout generator that works at the circuit (netlist) level is required.

6.2.2 Leaf Cell Generators in OPASYN

Layout generation in OPASYN starts from the netlist level. The goal is to produce design-rule-correct mask geometries of an op amp given a netlist and a set of device sizes. The layout generation strategy, as will be explained in subsequent sections, depends on efficient leaf cell generators for the basic circuit elements used in CMOS op amps: transistors, capacitors, power busses, transistor-pairs, transistor-cascodes, etc.

The leaf cell generators operate in two modes: *analysis mode* and *synthesis mode*. In *analysis mode*, they take electrically relevant device parameters of a basic circuit element as input and quickly report the range of geometrical dimensions possible for the element without going through the mask level synthesis process. In *synthesis mode*, the generators takes the electrical parameters and the desired aspect ratio of a cell to produce the closest possible detailed mask level layout. The generators in this mode take many options in addition to the electrical specifications such as terminal location, internal connection, as well as the desired aspect ratio of the element. Fig. 6.3 shows the versatility of such a cell

generator. In Fig. 6.3a, a transistor-pair is shown with their terminals: the source terminals are at the bottom, the gate terminals are at the right and left sides of the top, the body terminal is at the middle of the top, and the drain terminals are just below the gate terminals. In this example, it is requested (by the floorplanner) that all the terminals except the gate terminal of the right transistor (which is on the poly silicon layer) be located on the metal layer. The calling program have also specified that the body terminal be implemented only on one side of the element, and that three gate fingers be used. In Fig. 6.3b, the generator is instructed to connect the source terminals of the two transistors in order to use them as a differential input pair. In Fig. 6.3c and Fig. 6.3d, the same transistor-pairs have been generated with different aspect ratios. The possibility to vary the shapes of various elements helps the floorplanner optimize the area usage and meet the user's shape constraints. These features will be explained in the next section.

Furthermore, the following features should be noted in all of these layouts:

- A finger structure is utilized to reduce the parasitic capacitance as well as the cell area.
- Many (in fact, as many as possible) contacts are used for source, drain and well areas to minimize the parasitic resistance and to prevent the latch-up.
- The source connection in Fig. 6.3b is made in metal to minimize the source regeneration effect when the element is used as a differential input pair.
- Poly-silicon usage is minimized to reduce the parasitic capacitances associated with the gate terminals.

These features are in accordance with the rules for analog circuit layout generally used by expert op amp designers.

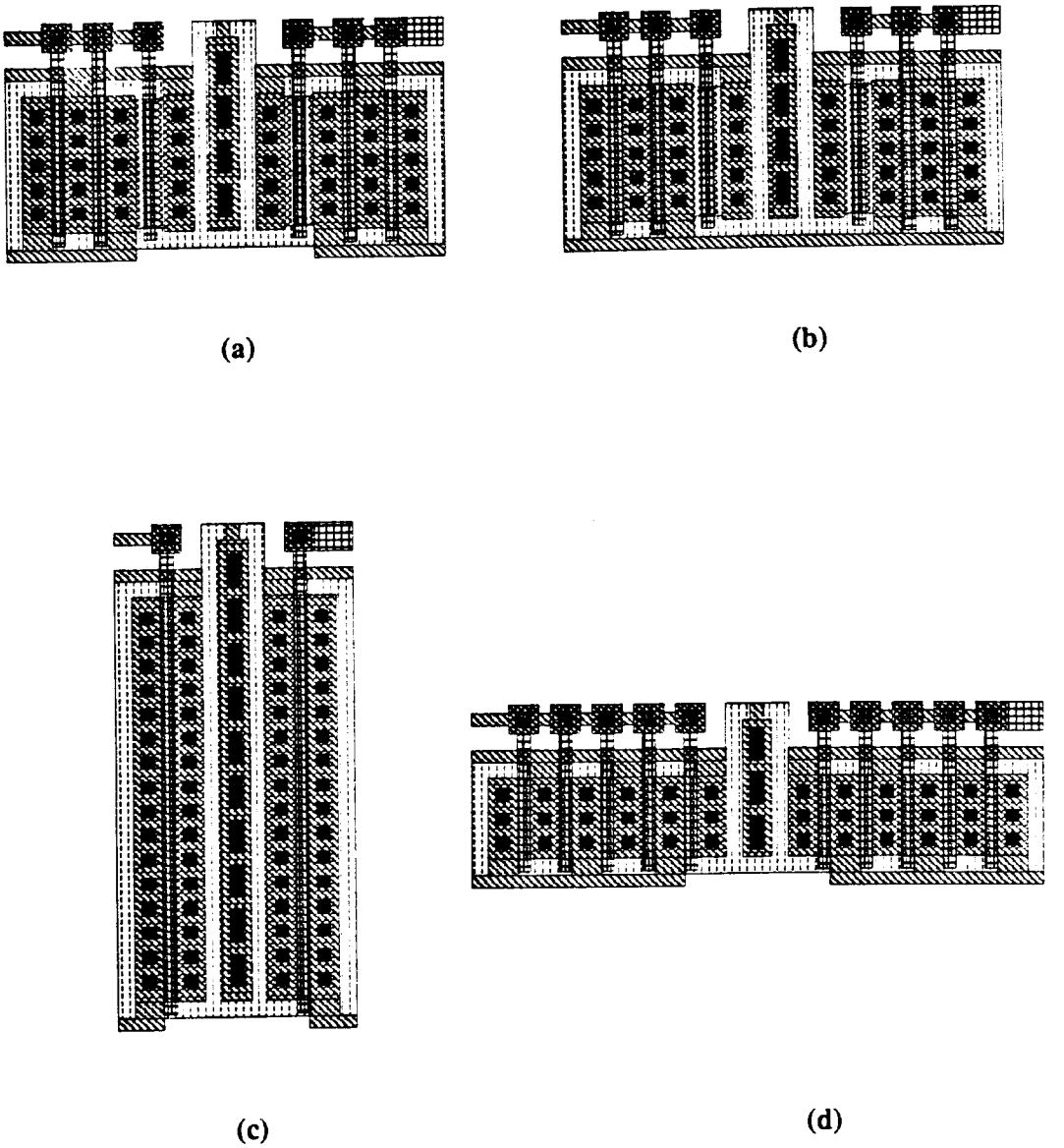


Figure 6.3 An example of a parameterized leaf cell — a transistor pair.

6.3 Floorplanning

Floorplanning is the first phase of the layout process. The layout floor is divided into as many partitions as the number of building blocks in the chip to be laid out, with each partition assigned to one specific block.

One of the difficulties in floorplanning for CMOS op amps arises from the fact that device sizes in the same circuit can vary over a wide range. Depending on the given performance specifications, it is not uncommon to see two orders of magnitude variations in device sizes — the W/L ratio of an output transistor can be as large as several hundreds when a large drive capability is required, while that of a biasing transistor may be less than unity. These large devices can be laid out in many different shapes and can even be divided into sub-devices if necessary. Furthermore, their terminals may be configured in many different ways. With so many degrees of freedom, the conventional layout optimization methods become rather inefficient. What makes things even worse is that, as mentioned in the previous sections, the performance of op amps is strongly affected by parasitic side effects and noise coupling incurred from an inadequate layout process. As a result, in most applications such as switched capacitor filters, A/D, and D/A converters, op amp layouts have been hand-crafted by experts.

In this section, the floorplanning strategy which is based on the parameterized leaf cell generators discussed in 6.2.2 and on circuit-dependent slicing trees is presented.

6.3.1 Use of Slicing Trees

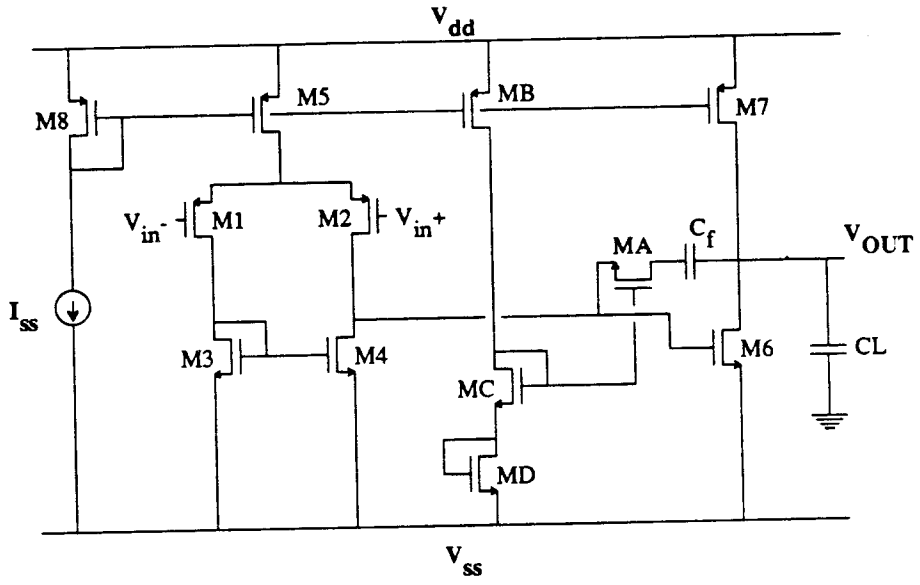
There exist many floorplanning algorithms developed for digital circuits such as min-cut, simulated annealing, clustering, constructive placement and so forth.[9,8,11] However, most of these algorithms will work efficiently only if all the blocks have fixed shapes and fixed pin locations. As mentioned earlier, some of the devices in op amps can take many different shapes. In addition, certain pairs of devices should be put as close as possible and

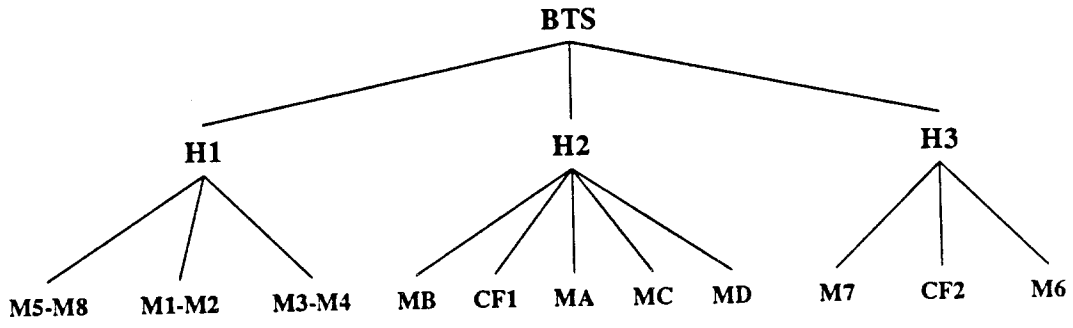
have the same orientation and the same dimensions to prevent random process variations/gradients from affecting the devices in different ways. Based on our investigations, we make the following observations:

- Some of the known algorithms such as min-cut and clustering are not convenient to handle constraints such as variable shapes and geometrical matching of blocks.
- Simulated annealing is able to handle a wide variety of constraints but it becomes too expensive when there are too many degrees of freedom resulting from variable shapes of the blocks and different terminal configurations.
- Top-down constructive placement seems to be the right way to arrive at a good layout topology, but placement methods based on simple net-length minimization cannot deal with variable shapes and geometrical matching conditions for the blocks. A set of general rules that can be applied to a broad range of analog circuits is hard to find. However, it has been observed that expert designers can 'somehow' figure out the relative locations of the blocks in the circuit without too much trouble using various constraints.

Since we want to make use of such expert knowledge, a specific slicing tree has been designed for each of the generic op amp topologies in the database. It specifies a sound topological arrangement for the building blocks of the circuit, reflecting the kind of traditionally accumulated design experience mentioned above. In addition, as explained in the next subsection, there exists an efficient algorithm[64] for optimally determining the shape of each leaf cell in the slicing structure given the shape constraint of the entire chip.

An example of such a slicing tree associated with the basic two-stage op amp is shown in Fig. 6.4. The entire layout floor is first partitioned into three horizontal slices (in the sense that they are produced by the horizontal cuts), H1, H2, and H3, based on both connectivity and functionality. In this example, the horizontal slice H1 contains the circuit elements





Note: C_f is divided into CF1 and CF2.

Figure 6.4 BTS OP amp and corresponding slicing tree.

which constitute an input stage. The horizontal slices H2 and H3 correspond to a compensation stage and an output stage, respectively. The same partition also comes from the fact that the circuit elements in the horizontal slice H1, for instance, are more closely connected to each other than to the elements in other horizontal slices. Then each horizontal slice is in turn divided into several vertical slices (again in the sense that they are produced by the vertical cuts) where their relative ordering primarily depends on their connectivity.

6.3.2 Floorplanning Algorithm

6.3.2.1 An Overview

Floorplanning starts by examining the possible shapes of each leaf cell in the circuit by running the corresponding cell generator in analysis mode (the two operation modes of the leaf cell generators are explained in 6.2.2). Given a set of appropriate specifications, i.e. width and length for a transistor and capacitance value for a capacitor, a cell generator then reports a list of possible size options for the cell. Based on this information, each leaf node is given a shape constraint relation devised by Otten.[63] Then the shape constraint relations of the ancestors of the leaf nodes are computed using Stockmeyer's algorithm[64] to combine the shape constraint relations of the leaf nodes in bottom-up fashion. In a subsequent, top-down traversal, the user-specified aspect ratio or size constraints are converted into actual x- and y-dimensions for each sub-tree, and finally for each leaf cell. It must be noted that Stockmeyer's algorithm (presented in the next sub-section) used in the bottom-up shape annotation and top-down constraint propagation process, selects the optimal shapes of the leaf cells that together best fit the given floor shape with the minimal area consumption. Next, the cell generators are called once again, this time in synthesis mode, to produce the mask layout for each cell with additional implementation options (see section 6.2.2). These layouts are assembled in a bottom-up manner according to the structure of the slicing tree.

6.3.2.2 Stockmeyer's Algorithm

Given a slicing tree that specifies relative positions of leaf cells in the circuit, the problem to be solved is the determination of the optimal shape/orientation of the individual leaf cells. An elegant algorithm due to Stockmeyer[64] which solves this problem in polynomial time is the subject of this sub-section. Stockmeyer proved that for the shape/orientation problem, the worst case complexity is $O(n^2)$, where n is the number of modules to be considered. Therefore, in case of basic analog function blocks including op amps where the number of modules (leaf cells) is under about twenty, the algorithm gives an optimal result without any performance degradation at all.

The Shape Constraint Relation

Given a rectangular module with a width equal to x and a height equal to y, the shape constraint relation **R** is defined as follows:[63]

$$\mathbf{R} = \left\{ (x,y) \mid y \geq h, x \geq w \right\}. \tag{6.3.2.2}$$

In Fig. 6.5, the shaded area represents the set of all pairs satisfying the equation 6.3.2.2 for the module shown in the same figure. If the module can be rotated by 90 degrees, then the relation is characterized by

$$\mathbf{R} = \left\{ (x,y) \mid y \geq h, x \geq w \text{ or } y \geq w, x \geq h \right\},$$

which is represented by the shaded area of Fig. 6.6, obtained by taking the union of the sets of points defined by the first component of the relation and the set of points defined by the second component. If the module can be rotated by 90 degrees or have n different widths and heights $h_1 \cdots ,h_n$ and $w_1 \cdots ,w_n$, then the relation is

$$\mathbf{R} = \left\{ (x,y) \mid y \geq h_1, x \geq w_1 \text{ or } \cdots \; y \geq h_n, x \geq w_n \text{ or } y \geq w_1, x \geq h_1 \text{ or } \cdots \; y \geq w_n, x \geq h_n \right\},$$

obtained by taking the union of the sets of points corresponding to each of the components

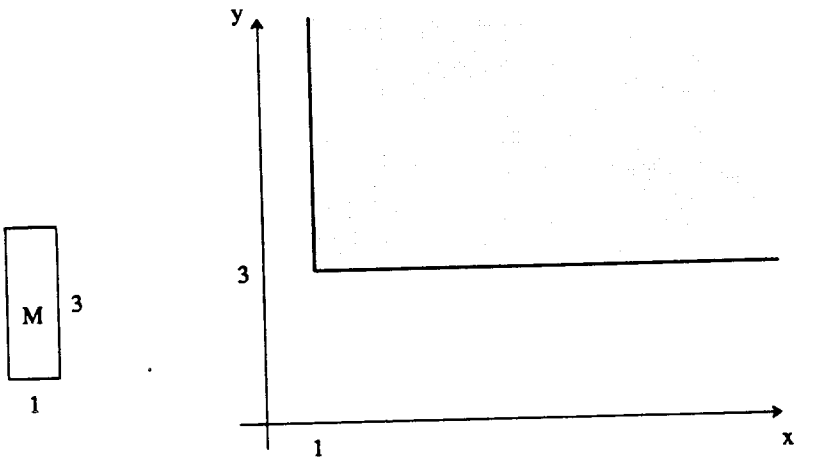


Figure 6.5 The shape constraint relation for module M.

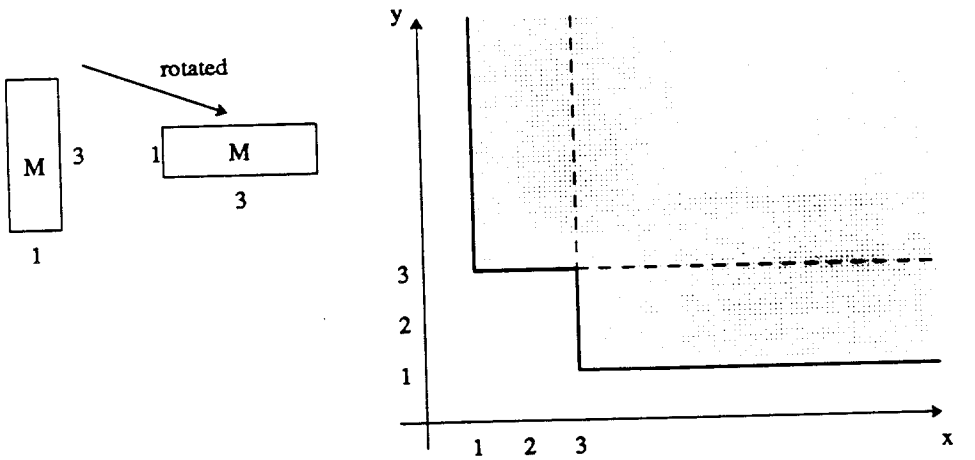


Figure 6.6 The shape constraint relation for M with rotation allowed.

of the relation. If the module is specified to have a fixed area A and a continuously varying aspect ratio like a capacitor, then the relation is given by

$$\mathbf{R} = \left\{ (x,y) \mid yx \geq A,\ y \geq h,\ x \geq w \right\},$$

where h and w are bounds on the size of the modules. Such a relation is shown in Fig. 6.7. Using the shape constraint relations described so far, if one of the two dimensions, say x, is fixed to some value X, then the other dimension giving the minimum size enclosing rectangle is given by the intersection of the constant line x = X with the shape constraint relation boundary as shown in Fig. 6.8. If, on the other hand, the desired aspect ratio is specified, all the available minimum size enclosing rectangles are investigated to select one that most closely observes the specified aspect ratio.

The Composition of Shape Constraint Relations

A slicing structure represented by a slicing tree describes how rectangles containing the modules of the design are positioned with respect to each other. If, according to the information encoded by the slicing tree, two rectangles are positioned on top of another, then the width of the enclosing rectangle is constrained to inherit the maximum of the widths of the two rectangles, while the height is given by the sum of the heights of the two rectangles. By the same token, if two rectangles are specified to be one to the right of the other, then the enclosing rectangle inherits the larger height of the two rectangles.

Thus, given the shape constraint relations of two rectangles, a combined shape relation can be obtained for the enclosing rectangle. If the two rectangles are placed one on top of the other, then the shape constraint relation $R_{1,2}$ for the enclosing rectangle is given by

$$\mathbf{R}_{node} = \left\{ (y,x) \mid (y_1,x) \in R_1,\ (y_2,x) \in R_2,\ \cdots\ (y_n,x) \in R_n,\ y = y_1 + y_2 + \cdots + y_n \right\},$$

where R_1 is the shape constraint relation for the first rectangle and R_2 is that for the second rectangle. This set can be obtained by summing graphically the shape constraint relations as

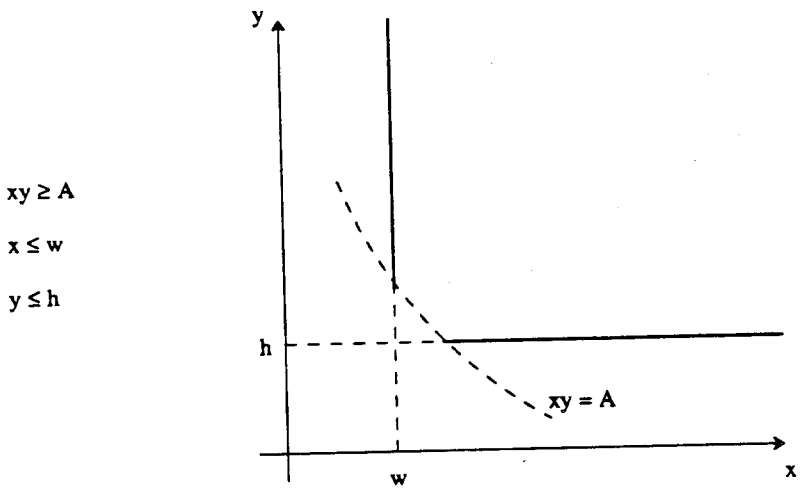


Figure 6.7 The shape constraint relation for a continuously variable module.

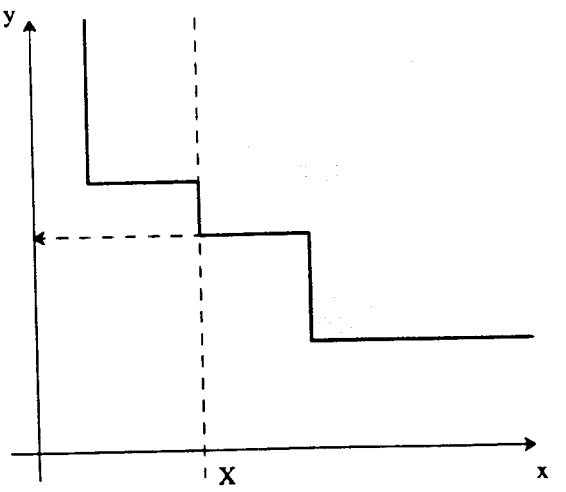


Figure 6.8 X-constraint determines the optimal value for y.

shown in Fig. 6.9. If the two rectangles are placed side by side, the shape constraint relation $R_{1,2}$ for the enclosing rectangle is given by

$$R_{node} = \left\{ (y,x) \mid (x_1,y) \in R_1, (x_2,y) \in R_2, \cdots (x_n,y) \in R_n, x = x_1 + x_2 + \cdots + x_n \right\}.$$

Fig. 6.10 illustrates the summing process. To make the composition of the shape constraint relations simpler to carry out, continuous boundaries such as the one shown in Fig. 6.7 are usually approximated by piece-wise linear functions. In this case, the resulting shape constraint relation can be easily computed since the boundary is also a piece-wise linear function whose break points are computed by combining those of the shape constraint relation boundaries of the enclosed rectangles.

The Module Shape Selection

To determine the minimum area layout satisfying a user constraint on aspect ratio, width, or height, this constraint is imposed on the shape constraint relation of the entire block. Then a boundary point of the relation that satisfies the constraint (if one exists) is selected. This information is then passed to the children of the root node. If the children correspond to a vertical cut, then the x-dimension is inherited from the root node. Then the y-dimension of each child is determined by identifying the y coordinate on its shape constraint relation that corresponds to the x value inherited from the parent node. This process is repeated until the x- and y- dimensions of all the rectangles of the slicing tree are determined. This procedure can be proven to generate a minimum-area chip that has the relative positions of its components specified by a fixed slicing structure. However, the optimization of the aspect ratio is carried out after the relative positions of the modules have been determined so that the result may not be an overall optimal layout.

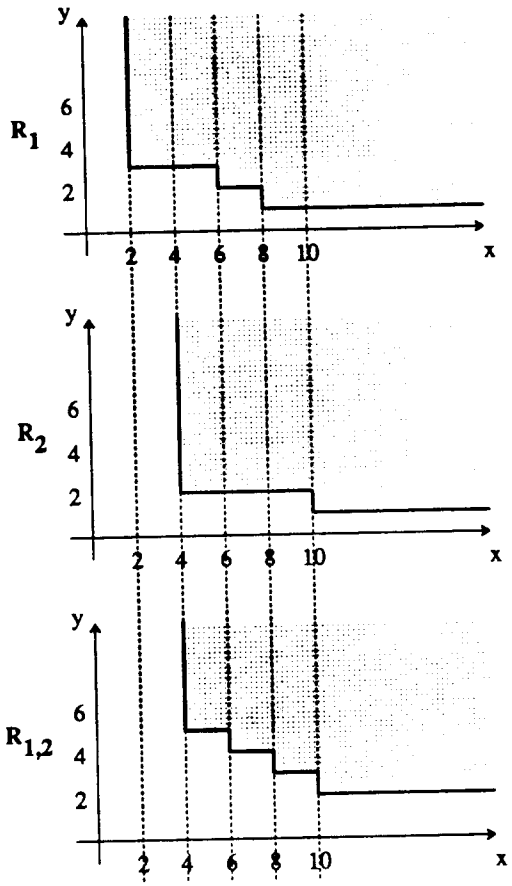


Figure 6.9 Summing the shape relations for a vertical assembly.

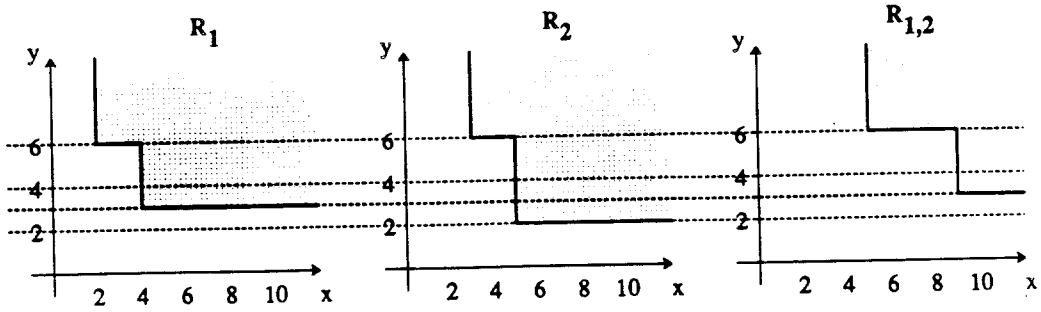


Figure 6.10 Summing the shape relations for a horizontal assembly.

6.3.2.3 Algorithm — OPASYN_FLOORPLAN

The floorplanning algorithm used in OPASYN is based on the discussion in section 6.3.2.1 and on Stockmeyer's algorithm. The algorithm *OPASYN_FLOORPLAN* is summarized in Fig. 6.11.

The example in Fig. 6.12 illustrates how the floorplanning algorithm works. By running an appropriate cell generator in analysis mode for all leaf cells in the circuit, a list of possible size options of each leaf cell is first obtained (Fig. 12a). The leaf cell A, for instance has two alternative shapes (1x3) and (2x2): the numbers representing the width and height, and the number of alternative shapes of the cell have been unrealistically kept small to ease the representation. In Fig. 6.12b, each leaf node of the slicing tree has been annotated with the corresponding shape constraint relation. Then using the summing formula for the horizontal and vertical cuts as shown in Fig. 6.9 and Fig. 6.10, the shape constraint relation is derived for every node in the slicing tree. In Fig. 6.12c, the input aspect ratio 1:1 is converted into the actual x- and y- dimensions of the entire floor — between two alternatives (5 by 4 or 4 by 5), the 5 by 4 floor shape has been arbitrarily selected. Then this constraint is propagated to its descendant nodes until the x- and y-dimensions of all the leaf cells are determined. Fig. 6.12d shows the bottom-up cell assembly process.

6.4 Routing

6.4.1 Routing Requirements

For analog circuits, routing has additional requirements that go beyond those of digital circuits:

1. Routing capacitance and resistance should be minimized in signal paths, since routing parasitics will affect the ac and transient characteristics of the performance.

Floorplanning Algorithm: *OPASYN_FLOORPLAN*

Input:
- Sized circuit schematic
- Slicing tree (T_s) description
- Size constraints
- Layout design rules

Output:
- X-y dimensions of the layout floor
- Optimal shape selection for each leaf cell
- An unrouted symbolic layout

Algorithm:

Step 1. Leaf cell shape inquiry

- **For all** leaf cells in T_s **do** {
  1. Call the corresponding generator in analysis mode with appropriate specifications (W, L, C).
  2. Store the list of possible size options returned from the generator.

  }

Step 2. Bottom-up shape annotation

- **For all** nodes in T_s **do** {

  if (node == leaf node)

  Generate a shape constraint relation (**R**) using a possible shape list.

  else if (node == vertical-cut node with n-children)

$$\mathbf{R}_{node} = \left\{ (y,x) \mid (y_1,x)\in R_1, (y_2,x)\in R_2, \cdots (y_n,x)\in R_n, y = y_1 + y_2 + \cdots + y_n \right\}.$$

  else if (node == horizontal-cut node with n-children)

$$\mathbf{R}_{node} = \left\{ (y,x) \mid (x_1,y)\in R_1, (x_2,y)\in R_2, \cdots (x_n,y)\in R_n, x = x_1 + x_2 + \cdots + x_n \right\}.$$

  }

(contines on the next page)

Figure 6.11 Floorplanning algorithm — OPASYN_FLOORPLAN.

Floorplanning Algorithm: *OPASYN_FLOORPLAN (continues)*

Step 3. Top-down constraint propagation

- Using R_{root} and the the input aspect ratio (otherwise the width or height constraint), determine x- and y-dimensions of the entire floor.
- **Do until** all x- and y-dimensions of the leaf cells are known {
  if (node has vertical-cut parent node) {
    1. Inherit x-dimension from the parent.
    2. Identify the y-dimension on R_{node} that corresponds to the x-dimension inherited from the parent node.

  } **else if** (node has horizontal-cut parent node) {
    1. Inherit y-dimension from the parent.
    2. Identify the x-dimension on R_{node} that corresponds to the y-dimension inherited from the parent node.

  }

Step 4. Cell synthesis

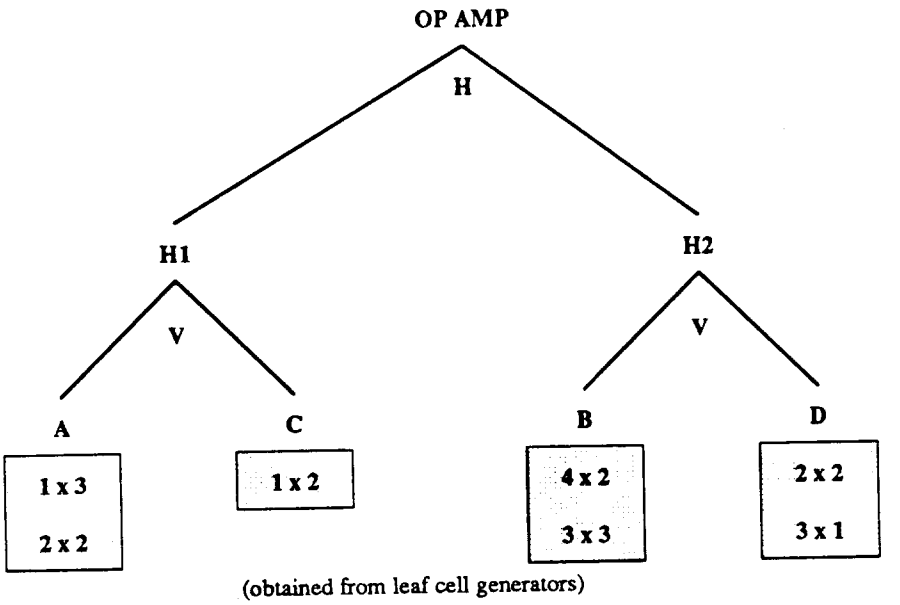
- Using the shape of a leaf cell determined at step 3, implement the cell by running the corresponding cell generator in synthesis mode.

Step 5. Bottom-up cell assembly

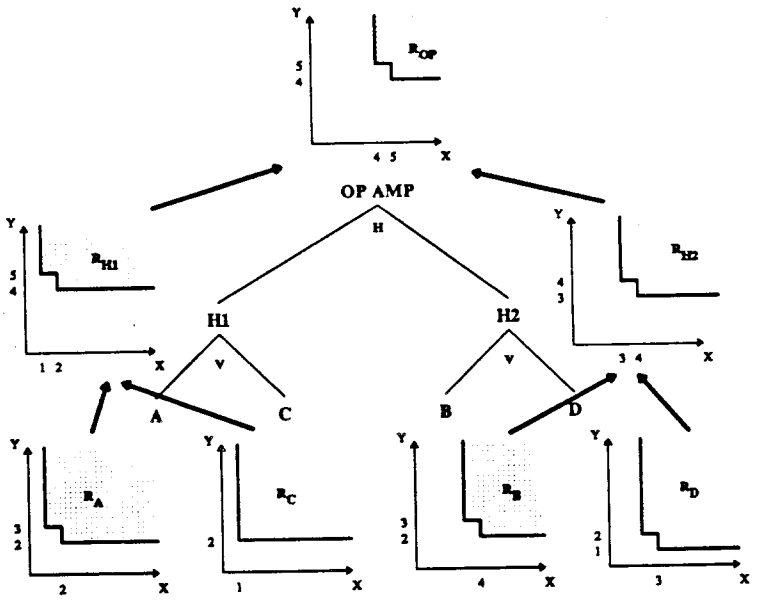
- Assemble the generated leaf cells according to T_s.

(end)

Figure 6.11 Floorplanning algorithm — OPASYN_FLOORPLAN.

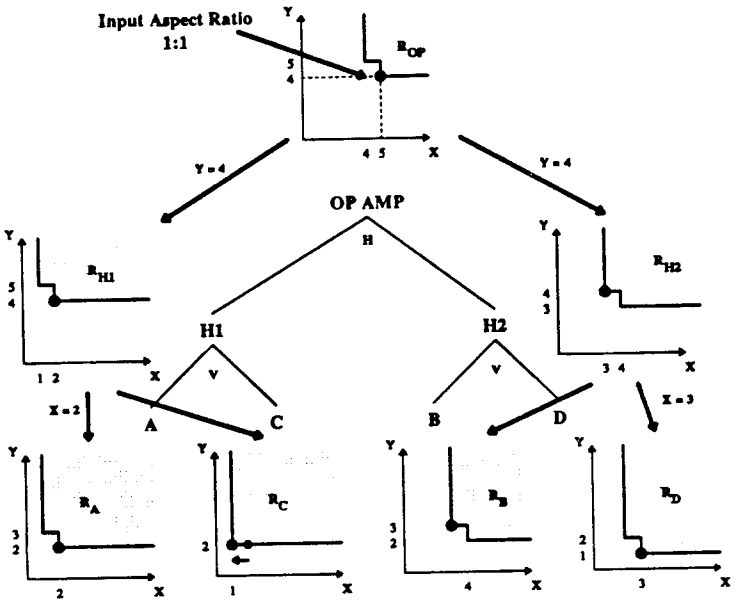


(a) Leaf cell shape inquiry

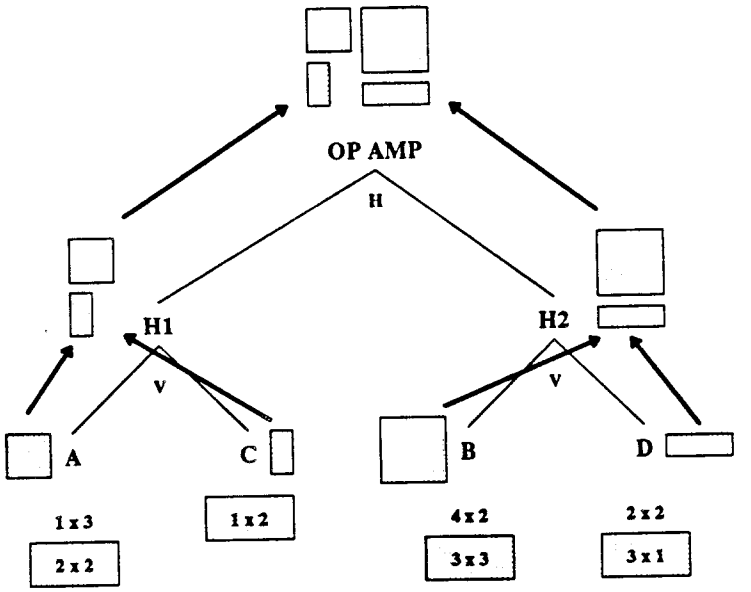


(b) Bottom-up shape annotation

Figure 6.12 Illustration of floorplanning algorithm.



(c) Cell selection by top-down constraint propagation



(d) Bottom-Up Cell Assembly

Figure 6.12 Illustration of floorplanning algorithm.

2. Resistance of some nets are more critical to the circuit performance than others, so they should be routed all in metal rather than in poly silicon. For instance, if source terminals of a differential pair are connected in poly silicon, the open-loop dc gain of the pair gets degraded due to the source regeneration effect.
3. A certain set of nets should be matched in terms of routing parasitics and/or their geometries. Such electrical and geometrical symmetry can provide a differential circuitry with immunity to random noise.
4. Crosstalk between signal nets should be avoided. Therefore, noise-sensitive nets should be buffered from noisy nets such as clock lines and output nets.

Currently, a router that can take care of all the above requirements is not available. Garrod et al.[26] have announced their line-expansion style router which models undesirable crosstalk interactions directly. However, too many other problems remain to be solved. Therefore, to use a conventional router without modification, it is necessary to analyze the above requirements specifically for op amps. By easing some of the less important requirements, it is possible to generate a set of routing constraints suitable for conventional routers: channel, switch box, or area routers.

Let's start with requirement 1. In op amps as well as in most of basic analog functional blocks, the major parasitic element affecting the ac and transient performance is parasitic capacitance rather than parasitic resistance because of the short wiring length (one exception is mentioned in requirement 2). Even though parasitic capacitances originate from both devices and routing wires, parasitic capacitances of devices often dominate those from wires because devices in op amps are rarely with minimum geometries. From the experiments, it is observed that the parasitic capacitance from a big device can be two orders of magnitude larger than that from a wire. Thus, maximizing the use of metal layer in routing will suffice to meet requirement 1 — there are many routers available that can maximize

the usage of a certain layer.

Next, regarding requirement 2, there is only one differential element in a single-ended op amp — the input transistor pair. However, this pair is constructed from a parameterized leaf cell generator that always connects its sources in metal. Therefore, requirement 2 has been already satisfied. For fully-differential op amps, it is typical to lay out one half of the circuit and to copy it with respect to a given symmetry axis in order to obtain the entire circuit. In this case, the differential input pair is centered on this symmetry axis and is removed from the list of blocks to be copied. Thus, requirement 2 can also be solved at the cell generator level.

In a single-ended op amp, geometrical/electrical symmetry is required only for the wires that connect the gates of the input differential pair to the two input terminals of the op amp. Such a symmetry requirement can be handled by using an area router in the following way; one of the two nets is routed without any obstacle, and the routed net is then copied with respect to the given symmetry axis. Afterwards, the routing problem becomes an area routing with obstacles composed of the pre-placed input nets and the building blocks of the circuit. In a fully-differential op amp, every wiring must be symmetrical with respect to the given symmetry axis. By the same principle used to meet requirement 2, only one half of the circuit is routed using additional pseudo-terminals on the symmetry axis. The result is then copied with respect to the symmetry axis.

Major degradation of the performance of an op amp is caused by the cross talk between the input and output nets. This is because the output signal varying with large voltage swings, if coupled into the input signal, even the smallest coupling can be amplified by the high gain of an op amp and appears at the output. For other nets, the cross talk can be safely ignored. Since the input pair is assigned far from the output pull-up and pull-down transistors when the slicing tree is designed, and the input and output nets are pre-routed with the goal of keeping them apart, requirement 4 does not create any additional constraint for the rest of

the routing problem — the input and output nets are just obstacles in the later routing stages.

6.4.2 Routing Strategy

As it becomes clear from the above discussions, routing is done in several steps. All the nets of each circuit are assigned to a few classes with different routing priorities. Then each class of nets is routed in priority order; e.g., the input connections are placed in the first priority class and are wired early, to make it easy to obtain a symmetrical routing with reasonably matched parasitics. Within each class, the nets are routed one at a time with the area router MIGHTY.[49] When a particular net runs into a bottleneck produced by another net, the latter may undergo some local rip-up and rerouting. Once all the nets belonging to a certain class have been routed, they constitute fixed obstacles for the nets in the lower priority classes. As a result, the whole circuit area is considered as a routing area littered with many obstacles, of which some are building blocks of the circuit and the others are pre-placed nets. This approach works well since op amps have normally less than 100 nets and cover only a limited area. Another alternative is to use a channel router. In this case, the channels, their associated netlists, and all floating pins are defined as part of the slicing tree structure for each circuit topology. However, geometrical symmetry between a pair of nets cannot be easily accommodated; it should be taken care of by a compaction program at the later stage. Priority-based routing is also difficult to implement because most channel routers can only differentiate critical nets from the rest. Thus, more than two values of priority cannot be treated.

6.5 Layout Spacing

6.5.1 Layout Spacing Requirements

The final step in the layout process is to re-space a symbolic layout to produce a compact, design-rule-correct mask level layout of an op amp. During the re-spacing process, the topology of the layout is conserved but its geometrical symmetries will in general be destroyed and terminals at the edges of the chip can move towards the center. To prevent such undesired side effects from happening, appropriate constraints must be generated for the spacing program. For analog functional blocks, the following type of constraints are useful:

Active Constraint — forces the relative spacing between one pair of blocks to be the same as the relative spacing between another. The active constraint is useful to conserve geometrical symmetries in the layout such as common-centroid shaped input transistor pairs, symmetrical input wires, etc.

Protection Frame — can be used to prevent some part of the layout from being affected by the re-spacing process.

Fixed Position — is used to specify the terminal positions. By fixing the x- or y-coordinate of a terminal, it is possible to re-space the layout while all the terminals still remain at the cell boundaries.

Therefore, either a one- or two-dimensional spacing program which is able to efficiently handle the above constraints can be used to re-space the symbolic layout of an analog functional block. In the case of an op amp, all of the above constraints are used. The active constraints are used to preserve the symmetrical routing patterns. The protection frames are used to protect the contents of the leaf cells whose geometries already observe the layout design rules. Lastly, terminals are fixed at the cell boundary by specifying either their x- or y-position.

6.5.2 Spacing Strategy

A set of layout design rules together with geometrical constraints such as symmetry and location of I/O pins are sent to a spacing program such as the one-dimensional, graph-based compactor SPARCS.[66] Since a protection frame has already been generated for each leaf cell, this program will not change the layout of the leaf cells themselves, but will simply adjust their separations and compact the wiring. The leaf cells produced from the cell generators are already in a compact, design-rule-correct physical format. In this way, the complexity of the spacing problem is substantially reduced. However, the price paid for the reduced complexity is the loss of control on fine details; i.e., it is not easy to permit terminal sharing between leaf cells or to move power nails onto the terminals of these cells. The average area penalty observed in our experiments is around 5% of the total area. The output from the compactor undergoes a few post-processing steps to straighten out unnecessary jogs and to remove redundant vias.

CHAPTER 7

EXPERIMENTAL RESULTS

7.1 Implementation

The OPASYN modules shown in Fig. 3.1 have been implemented in Common Lisp and in C. The circuit selection module is written in Common Lisp because of its convenience for implementing a rule-based system. The details of the rule-based implementation of the circuit selection module has been described in Chapter 4. The parametric optimization module is written in C because it is iteratively used in an inner loop of the optimization step, and efficiency of execution is most important. The layout generation program is written in Common Lisp since this permitted fast prototyping. However, the leaf cell generation programs (explained in Chapter 6) are coded in C.

The synthesis program is running under the Ultrix operating system on a VAX 8800. None of those modules have any process technology dependent codes; all technology information is stored in the database. Thus, the user simply specifies which fabrication process is to be used for the synthesis. Two different technologies (MOSIS 3 μm and GE 1.5 μm processes) have been used in experiments so far. To-date four commonly-used op amp topologies and an output buffer have been incorporated into the database; they are

a basic two-stage op amp (BTS),

a single-stage fully-differential op amp (FFC),

a single-stage folded cascode op amp (SFC),

a two-stage op amp with cascoded input stage (CTS),

a complementary Class B output buffer (OPB).

Circuit schematics of these op amps and the output buffer are provided in the Appendix. Fig. 4.3 and Fig. 4.4 (in Chapter 4) demonstrate how one of those topologies may be selected based on the specifications provided by the user of OPASYN.

7.2 A Synthesis Example

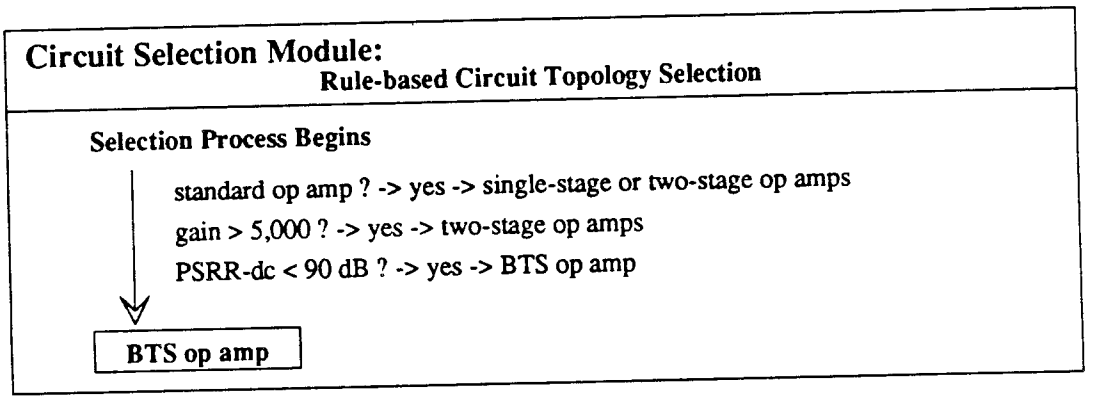
Fig. 7.1 presents a synthesis example from its performance specifications to the complete design-rule-correct mask level layout of an op amp. Fig. 7.1a shows an excerpt of the performance specifications from the user. Bias voltages to be used are +/- 2.5 V, the process technology is the MOSIS 3 μm process, and the specified load capacitance is 5 pF. Then the major performance specifications are given for power dissipation, frequency response, large-signal transient response and noise characteristics. In addition to the specifications shown here, there are a few others such as common-mode input voltage range, output voltage range, and CMRR. If any of the specifications is not specified by the user, a default value provided by the program is used in the synthesis. Fig. 7.1a also shows the optimization priorities; in this case, the top priority is given to minimizing the total gate area.

Based on these specifications, the BTS op amp (shown in the Appendix) has been selected by the circuit selection module. The program's reasoning process is displayed in Fig. 7.1b. First, the program decides whether any of the special-purpose op amps (such as an output buffer, a low-noise op amp, a low-power op amp etc.) is required to meet the given specifications. In this case, none of those special features are needed, and the program selects one of the standard op amps, the BTS op amp, from the database depending on open-loop dc gain and PSRR at dc.

Once a specific circuit topology is selected for the synthesis, optimal design parameter values are determined during the parametric optimization phase. Fig. 7.1c summarizes the parametric optimization results. The expected circuit performance according to OPASYN

User:	Performance Specifications		
Standard OP Amp; V_{dd} = 2.5 V, V_{ss} = -2.5 V; Load Cap. = 5 pF; MOSIS 3 µm Process			
power diss. (mW)	< 1	settling time (nsec)	< 1000
output swing (V)	> 1.5 / < -1.5	total gate area (mil^2)	MINIMIZE
open loop gain	> 5,000	1/f noise ($V/\sqrt{Hz}$)	< 1E-7
unity gain bw. (MHz)	> 2	PSRR-dc (dB)	> 70
slew rate (V/µsec)	> 1	PSRR-1kHz (dB)	> 40

(a)



(b)

Figure 7.1 A Synthesis Example - From Specifications to Silicon.

(a) A summary of performance specifications.
(b) Circuit topology selection process.

Parametric Optimization Module: Optimal Device Sizes	
Current Sources	
Device Name	Size
I_{ss}	16 μA
Capacitors	
Device Name	Size
C_f	8.9 pF
MOSFETs	
Device Name	Width (μm) / Length (μm)
M1 = M2	470 / 3
M3 = M4	84 / 6
M5	250 / 4.5
M6	890 / 6
M7	1300 / 4.5
MA	90 / 3
MB	170 / 4.5
MC	12 / 3
MD	56 / 3

(c)

BTS OP Amp - Performance Summary			
V_{dd} = 2.5 V, V_{ss} = -2.5 V; Load Cap. = 5 pF; MOSIS 3 μm Process			
Performace	**Specifications**	**OPASYN**	**SPICE**
power diss. (mW)	< 1	0.6	0.6
output swing (V)	> 1.5 / < -1.5	2.3 / -2.2	2.3 / -2.2
open loop gain	> 5,000	5,200	4,900
unity gain bw. (MHz)	> 2	3.1	3.0
slew rate (V/μsec)	> 1	1.4	1.4
setting time (nsec)	< 1000	980	1230
total gate area (mil^2)	MINIMIZE	42	N/A
1/f noise ($V/\sqrt{Hz}$)	< 1E-7	1.1E-7	NOT AVAILABLE
PSRR-dc (dB)	> 70	78	78
PSRR-1kHz (dB)	> 40	63	70

(d)

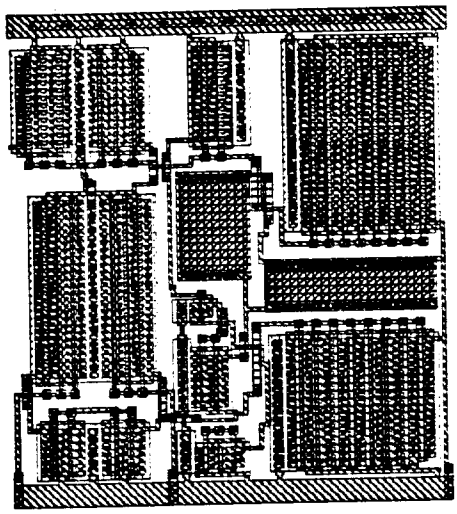
Figure 7.1 A Synthesis Example - From Specifications to Silicon.

(c) Parametric optimization results - optimal devce sizes.

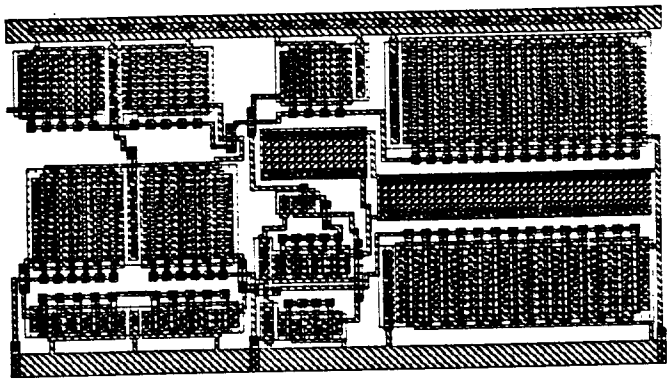
(d) A summary of expected circuit performace.

Layout Generation Module: Mask Level Layouts of Size Circuit Schematics		
BTS OP Amp; MOSIS 3 μm Process		
Geometric Data	**Layout (Fig. 7.1f)**	**Layout (Fig. 7.1g)**
input aspect ratio	1.0	2.0
achieved aspect ratio	0.9	1.8
total area (mil^2)	255	278
routing area (mil^2)	53 (20 %)	68 (24 %)

(e)



(f)



(g)

Figure 7.1 A Synthesis Example - From Specifications to Silicon.

(e) Geometric data for layouts shown in Fig. 7.1f and Fig. 7.1g.
(f) A layout with an input aspect ratio of 1.
(g) A layout with an input aspect ratio of 2.

and that according to SPICE simulation runs are compared in Fig 7.1d.

Finally, using the device sizes determined in the optimization phase and the netlist from the database, the layout generation program constructs a physical layout of the circuit schematic in macro cell layout style. Mask level layouts with an input aspect ratio of 1 and 2 are shown in Fig. 7.1f and Fig. 7.1g with some of their geometric data in Fig. 7.1e.

7.3 Performance Evaluation

7.3.1 Run Times

The entire synthesis, from specifications to layout takes under 5 minutes on a VAX 8800. The exact CPU time varies with the difficulty of the user specifications, the complexity of the circuit, and the degree of optimality to be achieved. More than 80% of the CPU time is currently spent in the layout generation, module which is written in Lisp; the parametric optimization takes less than 10 seconds for all test examples including the OPB output buffer and the FFC op amp which have more than 60 transistors. Reimplementation of the layout module in C together with further code optimization could lead to much shorter CPU time. This would make it quite practical to include OPASYN as a module generator in an interactive system planning tool for mixed analog-digital circuits.

7.3.2 Quality of Results

The quality of the achieved designs is quite high. The OPASYN's topology selection decisions agree with those made by expert human designers. The circuit performance predictions made by OPASYN are in rather in good agreement with SPICE simulation results. For dc characteristics, excellent agreement is generally achieved — observed deviations are less than 5 %. For ac and transient characteristics such as phase margin, gain, and settling time, the deviations are less than 20 %.[24] Such accuracy has been consistent for a broad

range of circuits, from the relatively simple BTS op amp to the complicated FFC op amp. The accuracy is also maintained for across the two different process technologies currently available in the synthesis.

Table 7.1 through Table 7.4 show the experimental results for the BTS op amp, the SFC op amp, the CTS op amp and the FFC op amp with two different process technologies: MOSIS 3 μm process and more advanced GE 1.5 μm process. The experiments are done with a wide range of specifications as well as with different optimization priorities. Power dissipation requirement varies from 1 mW (Table 7.1) to 30 mW (Table 7.3), settling time requirement from 100 nsec (Table 7.3) to 1000 μsec (Table 7.4), and open-loop gain from 60 dB (Table 7.4) to 100 dB (Table 7.1). The optimization program successfully deals with all these cases. If the user's design objectives are too demanding the program will provide the user with the best solution found. Such an example is shown in Table 7.3. In this experiment, the designer specified a very fast settling time of 100 nsec. The optimization procedure tired to comply as much as possible and ended up with 160 nsec. Optimization priorities have been varied also; in Table 1, total gate area, in Table 2, settling time, and in Table 3, settling time and total gate area have been used as top-priority optimization goals.

As mentioned in Chapter 5, analytic models can be developed for any functional module. Table 7.5a through Table 7.5c show three examples of a complementary Class B output buffer synthesis. In those examples, a single 5 V bias voltage is used while load capacitance varies from 10 pF to 100 pF and load resistance from 300 ohms to 10 kohms. Performance specifications are given for bandwidth, output voltage swing, and gain peaking. From the tables, it can be seen that dc performance prediction is excellent i.e., less than 1% deviation from SPICE results. The maximum prediction error for the bandwidth is 26 % and there is no error for gain peaking characteristic.

In general, the generated layouts observe most of the heuristic layout rules used by expert designers and show good area usage; they are about 20 to 30% larger than the sum of

parmameter	spec	GE1.5u		MOSIS3u	
		OPASYN	SPICE	OPASYN	SPICE
Vdd (V)	2.5	2.5	2.5	2.5	2.5
Vss (V)	-2.5	-2.5	-2.5	2.5	2.5
CL (pF)	10	10	10	10	10
gain	> 10,000	23,610	24,250	12,100	11,130
power diss. (mW)	< 1	0.67	0.66	0.69	0.69
phase margin (deg)	> 60	57.1	56.1	61.3	58.9
W_u (MHz)	> 4	4.7	4.7	4.6	4.7
gain margin (dB)	—	—	15.1	—	15.8
CMRR (dB)	—	—	92	—	89
slew rate (V/µsec)	> 2.5	3.9	3.4	3.6	3.0
PSRR⁻@dc (dB)	> 70	94.8	94.8	85.2	90.1
PSRR⁻@1kilo (dB)	> 40	58.0	74.3	50.3	70.7
TS (nsec)	> 500	453	550	690	750
systematic offset (mV)	—	—	0.008	—	0.013
$V_{o,max}$ (V)	> 1.5	2.38	2.40	2.18	2.20
$V_{o,min}$ (V)	< -1.5	-2.33	-2.39	-2.27	-2.31
$V_{ic,max}$ (V)	> 1	1.45	1.40	1.15	1.11
$V_{ic,min}$ (V)	< -1	-2.33	-2.50	-2.27	-2.50
1/f noise ($V/\sqrt{Hz}$)	< 1E-6	2.2E-7	—	1.17E-7	—
total gate area (mil^2)**	< 40	35.3	—	45.9	—

Abbreviations used:
Vdd = positive power supply voltage
Vss = negative power supply voltage
CL = load capacitor
C_c = compensation capacitor
W_u = unity gain bandwidth
PSRR⁻@dc = Vss rejection ratio at dc
PSRR⁻@1kilo = Vss rejection ratio at 1 kHz
TS = settling time (1 V step, 0.1 % interval)
$V_{o,max}$ = maximum output voltage
$V_{o,min}$ = minimum output voltage
$V_{ic,max}$ = maximum common-mode input voltage
$V_{ic,min}$ = minimum common-mode input voltage
1/f noise = input equivalent 1/f noise at 1 kHz

Table 7.1 **Schematic synthesis and verification results for the BTS op amp with optimization priority given to total gate area. GE 1.5µm and MOSIS 3µm processes are are used.**

parmameter	spec	GE1.5u		MOSIS3u	
		OPASYN	SPICE	OPASYN	SPICE
Vdd (V)	2.5	2.5	2.5	2.5	2.5
Vss (V)	-2.5	-2.5	-2.5	2.5	2.5
CL (pF)	2	2	2	2	2
gain	> 1,500	2,810	3,110	1,421	1,496
power diss. (mW)	< 30	2.34	2.31	2.85	2.72
phase margin (deg)	> 60	36.3	29.9	38.8	33.0
W_u (MHz)	> 4	45.5	31.7	42.3	30.0
gain margin (dB)	—	—	14.4	—	15.0
CMRR (dB)	—	—	134	—	132
slew rate (V/μsec)	> 8	25	24	22	19
PSRR⁻@dc (dB)	> 40	125	115	122	112
PSRR⁻@1kilo (dB)	> 10	125	115	122	112
TS (nsec)**	< 100	98	110	164	160
systematic offset (mV)	—	—	0.009	—	0.01
$V_{o,max}$ (V)	> 1.5	2.06	2.20	1.86	2.10
$V_{o,min}$ (V)	< -1.5	-1.93	-2.20	-1.88	-2.10
$V_{ic,max}$ (V)	> 1	1.18	1.70	0.78	1.50
$V_{ic,min}$ (V)	< -1	-2.50	-2.50	-2.50	-2.50
1/f noise ($V/\sqrt{Hz}$)	< 1E-6	3.0E-7	—	1.5E-7	—
total gate area (mil^2)	< 70	45.5	—	75.0	—

Table 7.2 Schematic synthesis and verification results for the SFC op amp with optimization priority given to settling time. GE 1.5μm and MOSIS 3μm processes are are used.

parmameter	spec	GE1.5u		MOSIS3u	
		OPASYN	SPICE	OPASYN	SPICE
Vdd (V)	2.5	2.5	2.5	2.5	2.5
Vss (V)	-2.5	-2.5	-2.5	2.5	2.5
CL (pF)	5	5	5	5	5
gain	> 10,000	15,140	13,890	9,870	9,430
power diss. (mW)	< 1	0.85	0.86	0.89	0.88
phase margin (deg)	> 60	65.3	62.1	68.1	69.6
W_u (MHz)	> 4	7.2	7.2	6.9	7.0
gain margin (dB)	—	—	9.0	—	9.7
CMRR (dB)	—	—	95	—	94
slew rate (V/μsec)	> 2.5	4.7	4.6	4.1	4.1
PSRR⁻@dc (dB)	> 70	90	90	88	89
PSRR⁻@1kilo (dB)	> 40	90	90	88	89
TS (nsec)**	< 500	387	420	580	670
systematic offset (mV)	—	—	0.46	—	0.53
$V_{o,max}$ (V)	> 1.5	2.37	2.39	2.35	2.36
$V_{o,min}$ (V)	< -1.5	-2.32	-2.39	-2.28	-2.33
$V_{ic,max}$ (V)	> 1	1.45	1.50	1.41	1.48
$V_{ic,min}$ (V)	< -1	-0.88	-1.00	-0.84	-0.97
1/f noise ($V/\sqrt{Hz}$)	< 1E-6	1.7E-7	—	1.1E-7	—
total gate area (mil^2)**	< 40	38.7	—	49.3	—

Table 7.3 Schematic synthesis and verification results for the CTS op amp with optimization priority given to settling time and total gate area. GE 1.5μm and MOSIS 3μm processes are are used.

parameter	spec	OPASYN	SPICE
Vdd (V)	2.5	2.5	2.5
Vss (V)	-2.5	-2.5	-2.5
CL (pF)	2	2	2
gain (dB)	> 60	69	67.4
power diss. (mW)	< 5	2.24	2.28
phase margin (deg)	> 60	63.6	87.0
W_u (MHz)	> 15	14.4	13.0
slew rate (V/μsec)	> 5	9.5	11.4
TS (μsec)	< 1000	175	340
$V_{o,max}$ (V)	> 1.5	1.98	1.87
$V_{o,min}$ (V)	< -1.5	-1.98	-1.50
$V_{ic,max}$ (V)	> 0.5	0.77	1.40
$V_{ic,min}$ (V)	< -0.5	-2.18	-2.50
total gate area (mil^2)	< 100	58.8	—

Table 7.4 **Schematic synthesis and verification results for the FFC op amp with no optimization priority. MOSIS 3 μm process is used.**

parameter	spec	OPASYN	SPICE
Vdd (V)	5.0	5.0	5.0
Vss (V)	-5.0	-5.0	-5.0
CL (pF)	10	10	10
RL (kΩ)	10	10	10
W_u (MHz)	> 2.5	2.5	3.2
$V_{o,max}$ (V)	> 4.9	4.89	4.85
$V_{o,min}$ (V)	< -4.9	-4.78	-4.66
gain peaking (%)	< 0	0	0

(a)

parameter	spec	OPASYN	SPICE
Vdd (V)	5.0	5.0	5.0
Vss (V)	-5.0	-5.0	-5.0
CL (pF)	100	100	100
RL (kΩ)	1	1	1
W_u (MHz)	> 2.5	2.5	2.8
$V_{o,max}$ (V)	> 4.5	4.71	4.67
$V_{o,min}$ (V)	< -4.5	-4.43	-4.31
gain peaking (%)	< 0	0	0

(b)

parameter	spec	OPASYN	SPICE
Vdd (V)	5.0	5.0	5.0
Vss (V)	-5.0	-5.0	-5.0
CL (pF)	100	100	100
RL (Ω)	300	300	300
W_u (MHz)	> 2.5	2.4	2.7
$V_{o,max}$ (V)	> 4.0	4.38	4.25
$V_{o,min}$ (V)	< -4.0	-3.80	-3.66
gain peaking (%)	< 0	0	0

(c)

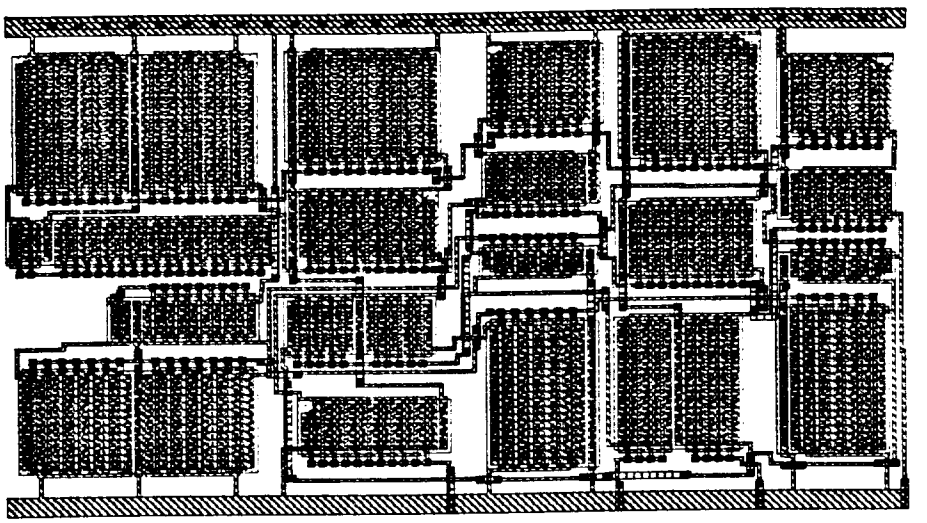
Table 7.5 **Schematic synthesis and verification results for the OPB output buffer with no optimization priority — (a) light, (b) medium, and (c) heavy output load.**

the area of the individual devices.[25] Further improvements seem possible by working at the interface to the spacing program. The other key feature of the layout program is that it can rapidly generate a layout to some desired aspect ratio. The deviations observed ranged from -20% to +20% compared to the specified target. Layout examples are provided in Fig. 7.1f, Fig. 7.1g and Fig. 7.2 together with their vital data.

Another important consideration during the layout phase is minimizing capacitive and resistive layout parasitics. Such parasitics originate both from devices and from routing wires; e.g., the junction perimeter capacitance of a transistor varies depending on the transistor shape. Since it is very difficult to handle device parasitics during the layout phase, analytic circuit models include procedural definitions for estimating the parasitic capacitances of large devices.† Thus the effects of parasitic capacitances due to large devices are considered during the optimization phase, not in the layout phase; the layout phase focuses on the minimization of routing parasitics only. This process is very similar to what human designers do. Designers estimate device dimensions such as the diffusion areas and perimeters of the drain and source junctions of a transistor (before they lay it out). Such information is then handed over to a circuit simulator to consider the parasitic effects. During the manual layout phase, they mainly try to minimize routing parasitics.

Fig. 7.3 shows an example. In Fig. 7.3a, the BTS op amp is shown with its node identification numbers. The parasitic capacitances (including wire capacitances) associated with some of the circuit nodes extracted by Magic's circuit extractor[67] are summarized in Fig. 7.3b. Lastly, the SPICE simulation results on the original circuit configuration (produced from the parametric optimization module) and on the extracted circuit are tabulated in Fig. 7.3c. The comparison shows that the performances of the two circuits are indeed in

† Presently, routing parasitics and parasitic resistances from devices are not considered during the optimization phase because of small layout area occupied by most of op amps. An exception is the source connection resistance of a differential pair which is taken care of by the corresponding leaf cell generator.



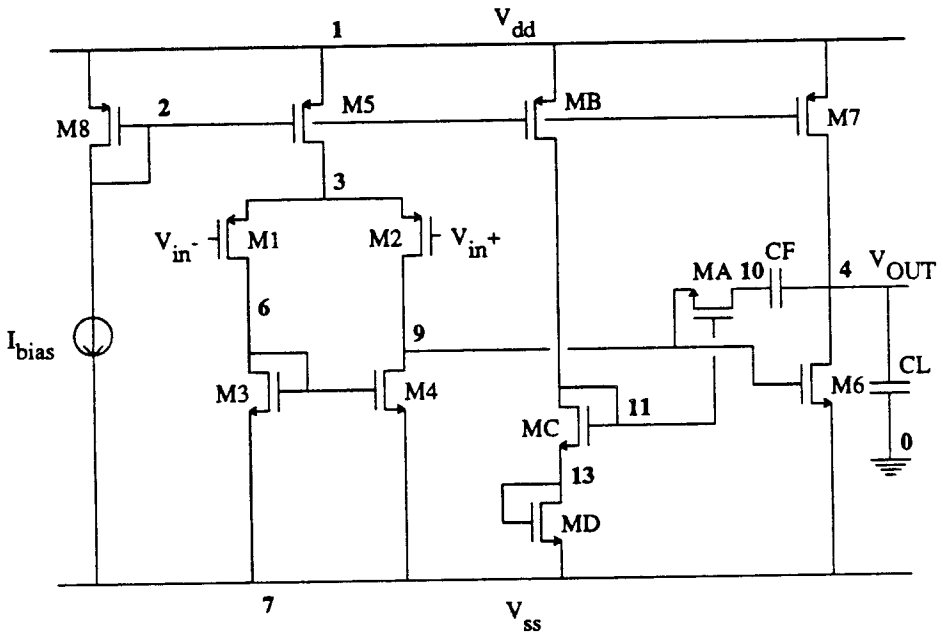
(a)

Layout Statistics	
Geometric Data	**Layout(Fig. 7.2a)**
input aspect ratio	1.5
achieved aspect ratio	1.74
total area (mil^2)	774
routing area(mil^2)	257(33%)

(b)

Figure 7.2 A layout example for the SFC op amp.

(a) A layout with an input aspect ratio of 1.5.
(b) Geometric data for the layout shown in Fig. 7.2a.



(a)

Figure 7.3 A parasitic extraction example.

(a) Basic two-stage op amp (BTS) with node identification numbers.

Extracted Parasitic Capacitances (Including wiring capacitances)	
node number	**capacitance (pF)**
2	0.538
3	1.993
4	4.227
6	1.111
9	1.362
10	0.889
11	0.492
13	0.215

(b)

Performance Comparison		
Performance	**Original Circuit**	**Extracted Circuit**
phase margin (deg)	65	68
unity gain bw. (MHz)	3.1	2.8
PSRR-1kHz (dB)	70.2	69.3
slew rate (V/μsec)	1.43	1.24
settling time (μsec)	1.23	1.40

(c)

Figure 7.3 A parasitic extraction example.

(b) Parasitic capacitance extracted from the layout in Fig. 7.1f.
(c) Performance comparison.

good agreement.

CHAPTER 8

OPASYN AS A COMPONENT IN A LARGER SYNTHESIS SYSTEM

8.1 Concept

As mentioned in the previous chapters, one of the objectives of building a functional module generation program like OPASYN is to include it into sub-system level synthesis systems such as compilers for switched capacitor filters or A/D coverters. The OPASYN program is able to produce a design-rule-correct macro cell layout of an op amp from a set of given performance specifications. By operating at the circuit level rather than relying on fixed library cells, the program is capable of providing an optimal op amp design for the specific requirements in a broad range of applications. As a result, the design time and cost associated with re-designing or fine-tuning existing functional modules are almost eliminated. With the availability of such module generators, sub-system level synthesis tools can make the best use of digital domain semicustom design methodologies including standard cell and macro cell approaches.

The advantage of having functional module generators for sub-system level synthesis goes beyond fast turn-around time. The interaction between a sub-system building tool and its module generators can improve the quality of designs as well; module blocks need not be over-designed to accommodate future, unknown applications. Moreover, their geometrical shapes are flexible, leading to better layouts. Therefore, it is possible for sub-system design tools to explore a much broader design space and thus to find a more optimal solution to a given design problem.

However, a set of efficient module generators alone are not enough to pursue design optimization of typical analog sub-systems. This is mainly because the proper set of optimal performance specifications for each of module components cannot be determined a priori at the beginning of the synthesis — an inherent disadvantage of a 'top-down' design method. In other words, an optimal 'threshold value' for a certain performance characteristic of a functional module (i.e. the slew rate of an op amp) is not known at the top-level of the design hierarchy. Consequently, analog designers have traditionally imposed excessive design requirements on the leaf-level modules to insure that their performances are well above the 'threshold values'; this causes the modules to be bigger and more complex than necessary. What should be done to derive an optimal design for a given sub-system, is to iterate the entire design cycle while adjusting the performance specifications for each of its leaf modules to find out their optimal values. The problem then reduces to that of a parametric optimization of the optimal performance specifications of all circuit components under the following two conditions:

- The circuit topology and the components to be used in the design are fixed,
- A given set of system-level specifications must be satisfied.

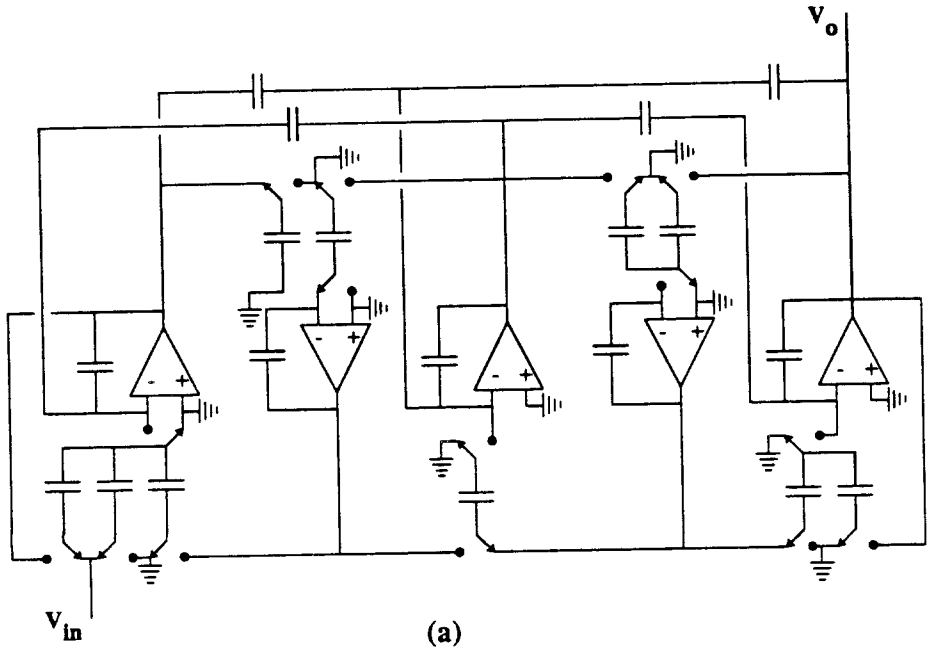
But such iterative design optimization is out of the question when the simulation of a tenth-order switched capacitor filter takes several hours of CPU time using conventional device models. Therefore, what we need in addition to efficient module generators is a fast, yet precise evaluation of the expected system performance.

While most of this thesis has been devoted to the efficient generation of functional modules, especially op amps, the rest of this chapter concentrates on the latter aspect of design automation of analog sub-systems. In section 8.2, the integration of OPASYN into a filter synthesis program is presented as a hypothetical example. Subsequently, an efficient way of evaluating system performance — 'macromodeling' is described.

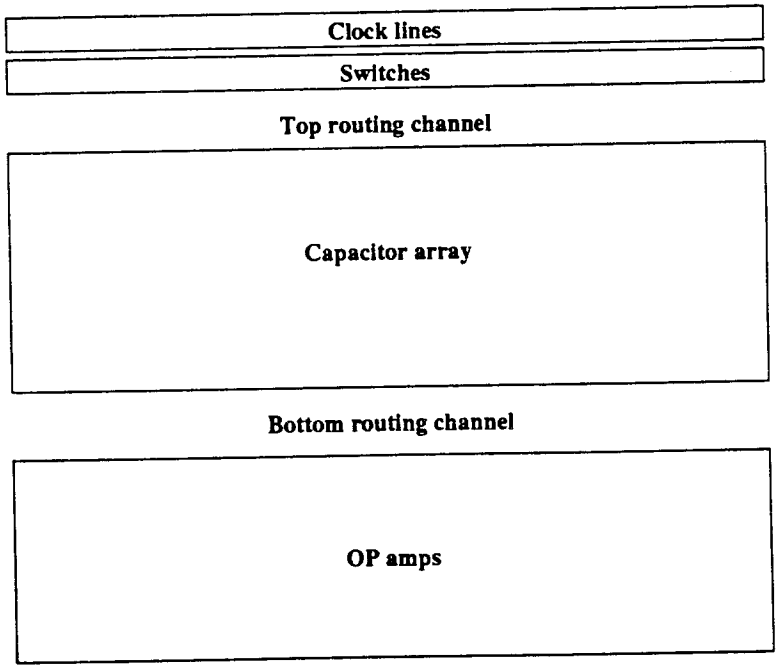
8.2 An Example — A SC-Filter Compiler

To further illustrate the concept of analog sub-system synthesis discussed in the previous section, let's consider the example of the synthesis of a fifth-order PCM low-pass filter. Presently, OPASYN can handle all the components necessary for switched capacitor filter synthesis. The circuit schematic of the filter and the associated floorplan are shown in Fig. 8.1a and 8.1b respectively. The filter consists of five switched capacitor integrator blocks and a number of switches. Each integrator block is composed of an op amp and an integration capacitor. Switches are implemented using transistor gates. The function of the filter is to suppress the high frequency signal components (above the cutoff frequency) present in the input signal. The floorplan shown in Fig. 8.1b is a common way of laying out switched capacitor filters. Note that the capacitor array forms a buffer zone that successfully separates the clock lines and transistor switches from the op amps. Such separation and buffering are necessary because the clock lines and transistor switches produce pulse signals (varying between V_{CC} and ground) which, when coupled into the input terminals of op amps, would cause substantial performance degradation.

Fig. 8.2 shows a hypothetical CAD framework for a switched capacitor filter compiler and how OPASYN can fit into such a framework. The entire system works as follows: The inputs to the system are the filter specifications such as dc gain, cutoff frequency, pass-band ripple, and stop-band attenuation. Based on these specifications, the filter schematic synthesis program first generates initial performance specifications for its component modules: op amps, capacitors, and switches. Next, OPASYN produces circuit configurations for all component modules and reports their performances and geometries to the schematic synthesis program which somehow must determine whether the overall performance is suitable for the given filter specifications or not. Often, such a decision is made by running a circuit simulator on the entire filter circuit. If the schematic synthesis program is convinced that an acceptable result has been obtained, the layout synthesis program assembles the filter layout.



(a)



(b)

Figure 8.1 Circuit schematic of a PCM low-pass filter and its floorplan.

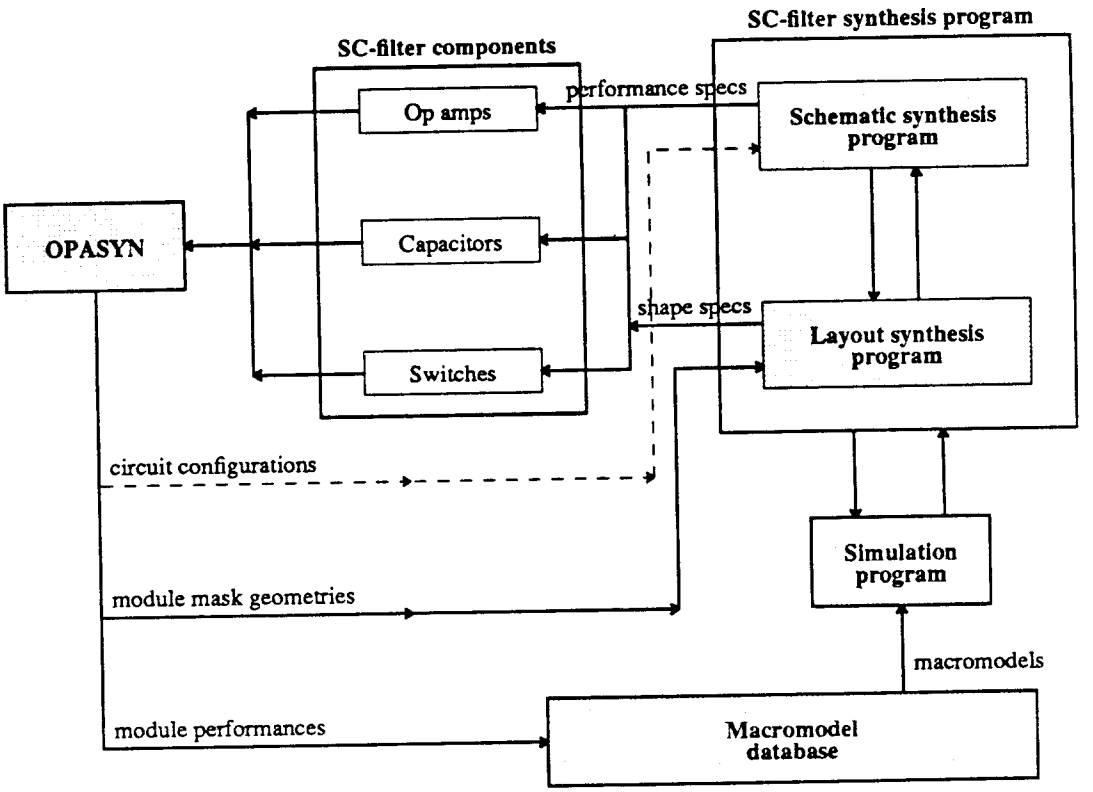


Figure 8.2 OPASYN as a module generator for a switched capacitor filter compiler.

If not, the schematic synthesis program modifies the initial module specifications and sends them to OPASYN again; the whole process may be repeated many times.

The main design objective during layout assembly is to find an optimal shape for each component module so that area utilization is maximized while the shape of the entire cell conforms to the given shape constraint as close as possible. Once the layout synthesis program has decided the most suitable shape for all component modules, OPASYN produces mask geometries for the components that best fits the desired shapes. It should be noted that with variable shapes of the components, the floorplanning problem is more complicated than that of fitting the capacitor array with minimum height into the width of the filter[13] which is given by the number of op amp library cells used.

Realization of the hypothetical CAD framework shown in Fig. 8.2 appears quite feasible except for one aspect — how to precisely evaluate the filter performance without spending too much CPU time. A conventional approach for such a performance evaluation is shown with a dotted line in Fig. 8.2. The module generation program returns circuit configurations for all component modules which are used by the filter synthesis program to run a circuit simulation program on the entire circuit; this may require many hours of CPU time even for a single simulation run depending on the size of the filter. Using this approach, it is impossible to perform iterative design optimization. Since the performance requirements for the op amps are relatively modest for operation in voice-band frequencies, significant area penalty may be incurred from the lack of such an optimization process.

A practical solution to this problem is to develop a 'macromodel' for each circuit component, especially for the op amps used in this example. These macro models should predict the real behavior of the component precisely enough for a system-level simulation, but contain considerably fewer active components (such as transistors) than the real circuit. Using macromodels for all circuit components rather than their actual circuit configurations reduces the simulation time considerably.

Macromodels of op amps were actively studied in the 1970's when the simulation cost was expensive and the number of devices a simulator could accommodate was quite limited.[68,69] Nowadays a typical simulation program can take care of rather a large number of devices and the macromodeling issue does not seem to draw much interest any more. However, today's simulation programs are fast and capacious enough only for one-time design efforts and for sub-system level designs. Iterative design optimization of such systems or mixed analog/digital system level designs would still be impossible. Development of practical macromodels for commonly used functional components are necessary. A detailed study on macromodeling issues, however, goes beyond the scope of this thesis — it is more closely related with device modeling for circuit simulation than design synthesis. In the next section, only a brief review on op amp macromodeling is presented.

8.3 Macromodeling of OP Amps

8.3.1 Circuit Theoretical Backgrounds

Modeling of a device (not a circuitry) has been under investigation for a long period of time. Two approaches are currently available for constructing a device model that agrees qualitatively with the real behavior: *the physical approach and the black-box approach.*[69]

In the physical approach, one tries to translate the physical structure and operation mechanisms of a given device into a circuit model. In this case the elements of the model usually bear a one-to-one relationship with the device's internal structure. This approach is quite sound and should yield realistic results provided that the physical operating mechanisms of the device are well understood. Unfortunately, the physics of many devices are not well understood and therefore the physical approach is not always applicable.

In the black-box approach, a valid dc global model is derived first, and then it is augmented with parasitic inductors and capacitors at one or more strategic locations. These locations are

usually selected so that suitable state equations for the network can be formulated.

Both approaches can be utilized to obtain a macromodel of a circuit. Since the operations of most circuits, especially, op amps are well understood, the physical approach has been used dominantly in the macromodeling of op amps. In this sub-section, a generic macromodel of op amps by L. O. Chua [69] is introduced. Much of the discussion made in the rest of this sub-section is based on Chapter 2 of his book — *Computer-Aided Analysis of Electronic Circuits*. A detailed example in the next sub-section then shows how the generic model can be augmented as needed.

To develop a macromodel of nonideal (real) op amps requires the following characteristics of an op amp to be considered by the model:

1. Finite input resistance $R_i \neq \infty$.
2. Finite output resistance $R_o \neq 0$.
3. Frequency-dependent open-loop voltage gain $G(\omega)$. A typical Bode plot for $G(\omega)$ is shown in Fig. 8.3a. The first corner frequency ω_1 is called the dominant pole frequency; and the frequency ω_o, where the curve $G(\omega)$ crosses the 0 dB axis, is called the unity-gain bandwidth.
4. Frequency-dependent phase shift $\theta(\omega)$ is shown in Fig. 8.3b. The phase shift at ω_o if augmented by 180 degree, gives the phase margin. The phase margin is noted as PM in the figure (note that it can be negative). The gain margin of the circuit, GM (in dB) is determined by

   $$GM = -20\log_{10}G(\omega_f)$$

   where ω_f is a frequency where $\theta(\omega) = -180$ degrees.
5. Output voltage-limiting behavior: $V_{o,min} < V_o < V_{o,max}$.
6. Finite slew-rate limitation SR, where SR is defined to be the maximum time rate of

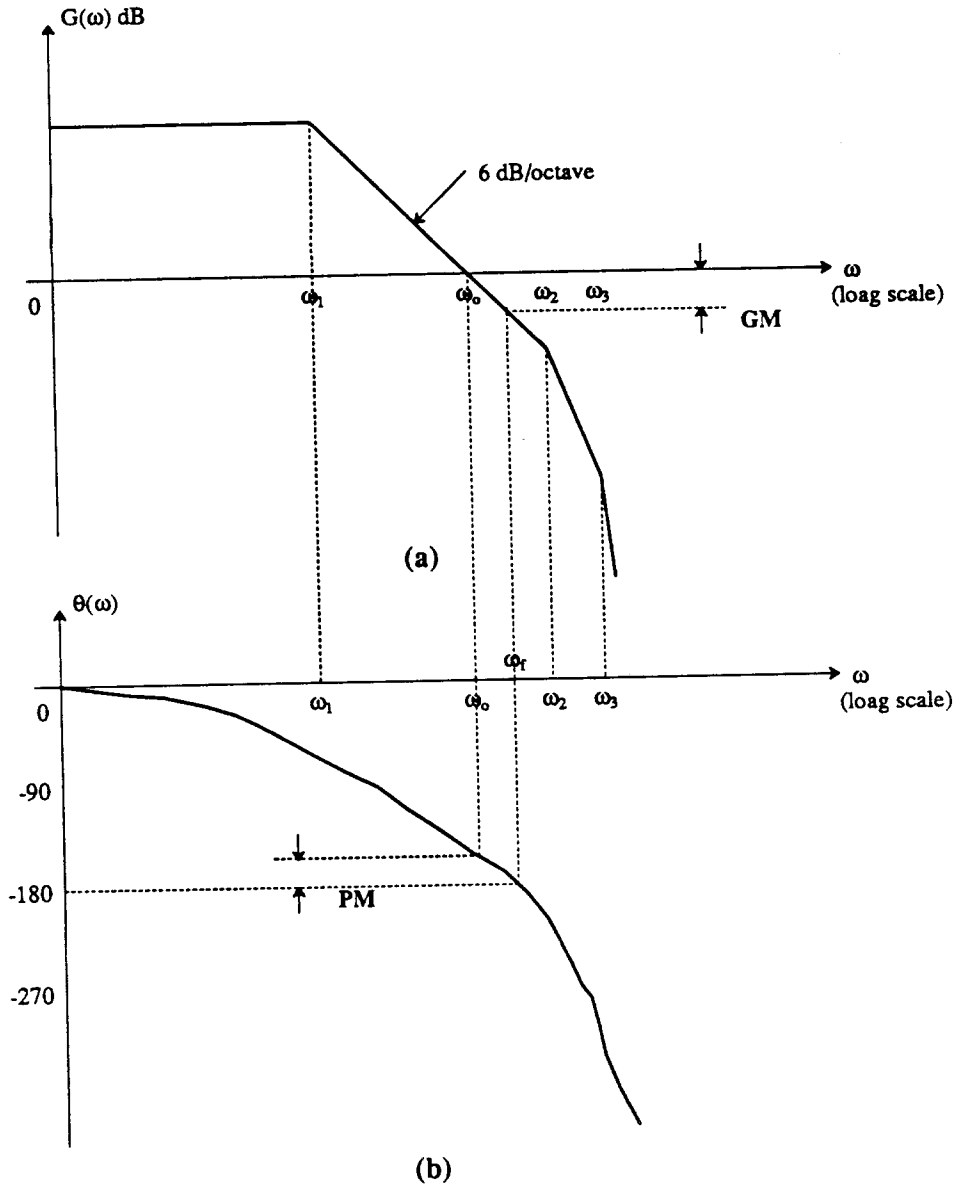


Figure 8.3 Bode plot of voltage gain and phase shift for an op amp.

change of the output voltage that can be attained by the op amp under the least favorable external circuitry. The slew rate is an important op amp parameter in the design of high speed circuits, such as A/D ad D/A converters, because it imposes an absolute limit on the attainable speed.

7. Common-mode input range and CMRR.

The first four types are typical of those found in linear circuits and it is relatively straightforward to simulate them. The fifth characteristic is also typical of that found in voltage limiters and could be simulated with a nonlinear resistor. To simulate the finite slew rate, the nonlinear circuit building block shown in Fig. 8.4 can be used, where the nonlinear voltage-controlled current source is characterized by the saturation curve $I_c = f(V)$.

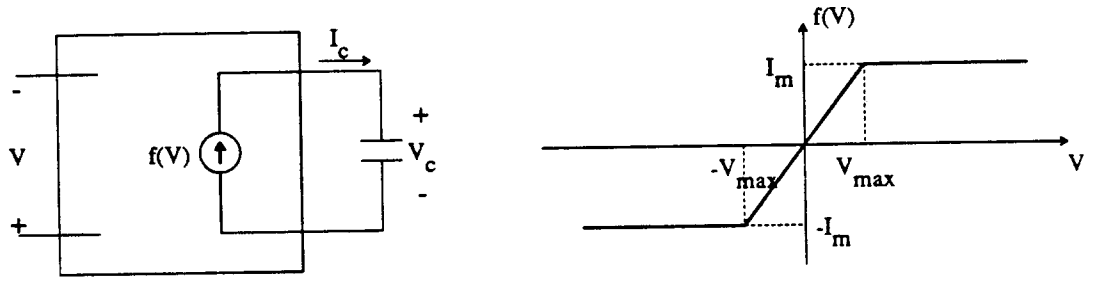


Figure 8.4 Simple nonlinear circuit for simulating slew rate limitation.

Since the output current of the controlled source cannot exceed I_m, we have

$$\left| \frac{dV_c(t)}{dt} \right| = \frac{1}{C} |I_c| \leq I_m \equiv S_r$$

It follows from the above equation that the slew rate of the circuit shown in Fig. 8.4 is equal to S_r. The last characteristic is rather difficult to be modeled using basic circuit components and a detailed example is shown in the next section.

Making use of the preceding observations, a generic macromodel for nonideal op amps

can be formulated as in Fig. 8.5a.[69] The nonlinear function $f(V_i)$ characterizing the nonlinear controlled source is shown in Fig. 8.5b, and the V-I curve characterizing the nonlinear resistor is shown in Fig. 8.5c. The generic model consist of five stages:

1. Input stage N_1.
2. Slew-rate limiting and gain stage N_2.
3. Dominant pole stage N_3.
4. Unity-gain, higher-pole-frequency stage N_4.
5. Output stage N_5.

The *input stage* consists of a single resistor R_i whose resistance is assigned to the prescribed op amp differential-mode input resistance. For CMOS op amps, this resistance is absent because of their high gate-to-substrate impedance. However, as shown in next sub-section, this simple model must be augmented to account for the finite common-mode input range and CMRR. The *second stage* is simply transplanted from Fig. 8.4 to consider the finite slew rate. If R_1 is chosen to be sufficiently high to provide negligible loading effects on N_2, then the slew rate is determined by I_m/C_1. The *third stage* simulates the dominant pole frequency ω_1 in Fig. 8.3 assuming that the Bode plot $G(\omega)$ falls off at 6 dB/octave which is typical for most op amps. The *fourth stage* is a grounded unity-gain linear two-port designed to simulate the higher-pole-frequency breakpoints of the prescribed Bode plot for $G(\omega)$ and $\theta(\omega)$. Many RC circuits can be chosen for this purpose and the associated circuit parameters can be determined by standard approximation and optimization techniques (refer to an example in the next sub-section). The *last stage* consists of a linear and a nonlinear resistor. The value of R_o is equal to the prescribed op amp output resistance. The V-I curve for the nonlinear resistor is shown in Fig. 8.5c. Hence, the last stage is used for simulating both the output resistance and the output voltage limiting behavior.

In the next sub-section, the generic model shown in Fig. 8.5 is augmented to produce a complete macromodel for a two-stage op amp.

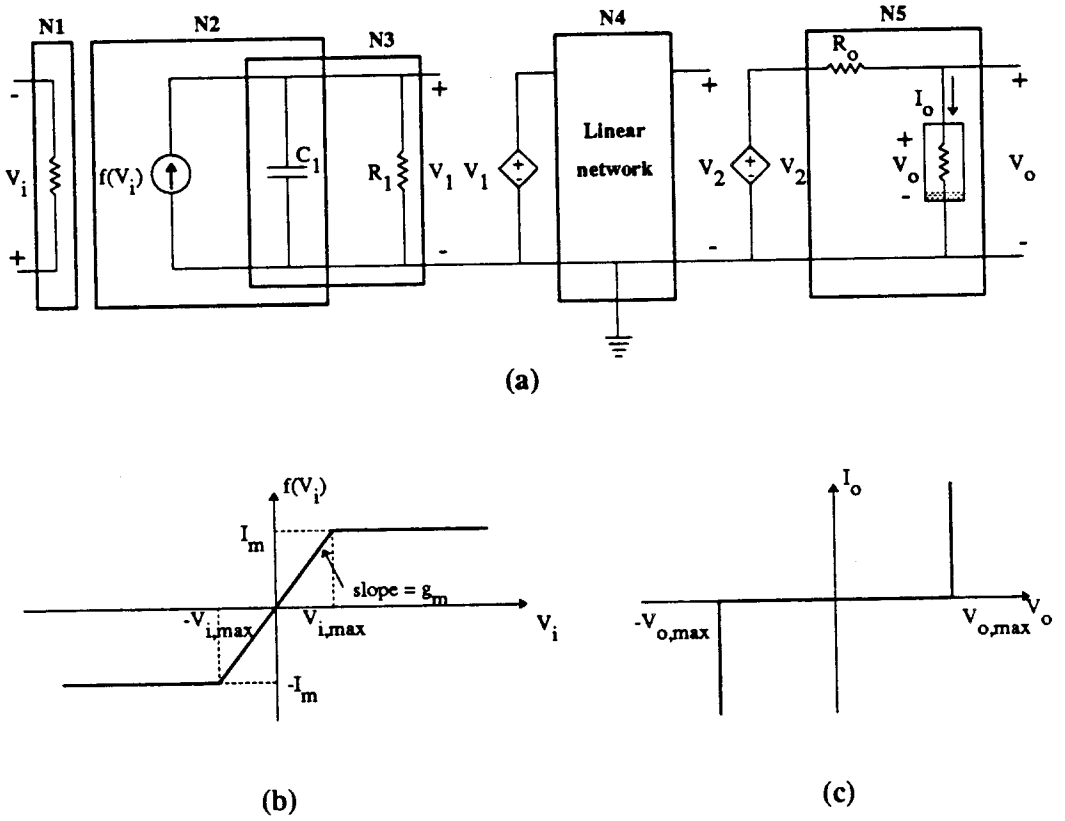


Figure 8.5 A generic macromodel for an op amp.

8.3.2 An Example — A Two-Stage OP Amp

A macromodel for a two-stage op amp developed by Boyle[68] is shown in Fig. 8.6. The configuration, with suitable choice of modeling parameters and elements, accurately models a broad class of two-stage op amps. It is shown that for a given op amp, the model provides an essentially pin-to-pin correspondence with the op amp, and accurately represents the circuit behavior for nonlinear dc, ac, and large-signal transient responses. The model shown in Fig. 8.6 is subdivided into three stages: an input stage, a compensation stage, and an output stage. The input stage consists of ideal transistors Q1 and Q2 and the associated sources and passive elements. This stage produces the necessary linear and nonlinear differential-mode and common-mode input characteristics. The capacitor CE is used to introduce a second-order effect for the slew rate, and the capacitor C1 is designed to introduce a second-order effect for the phase response.

The common-mode and differential-mode gains of the op amp are defined by the linear interstage and output stage elements consisting of G_{cm}, G_a. R2, G_b and RO2. The dominant pole frequency of the op amp is produced with the internal feedback capacitor C_2. A feedback connection in the macromodel is used for C2 in order to provide necessary ac output resistance change with frequency. Notice that the complete isolation between the input and the interior stages simplifies modeling of frequency response and slew rate characteristic.

The output stage provides the proper dc and ac output resistance of the op amp. The elements D1, D2, RC, and G_c produce the desired maximum short-circuit current. The elements D3, V_C and D4, V_E are voltage-clamp circuits to produce the desired maximum output voltage excursion.

It was reported[68] that using this model, a reduction of approximately 6 in branch and node count had been achieved while providing a very close approximation to the actual performance of an op amp. Expressions that relate the performance of the op amp and the

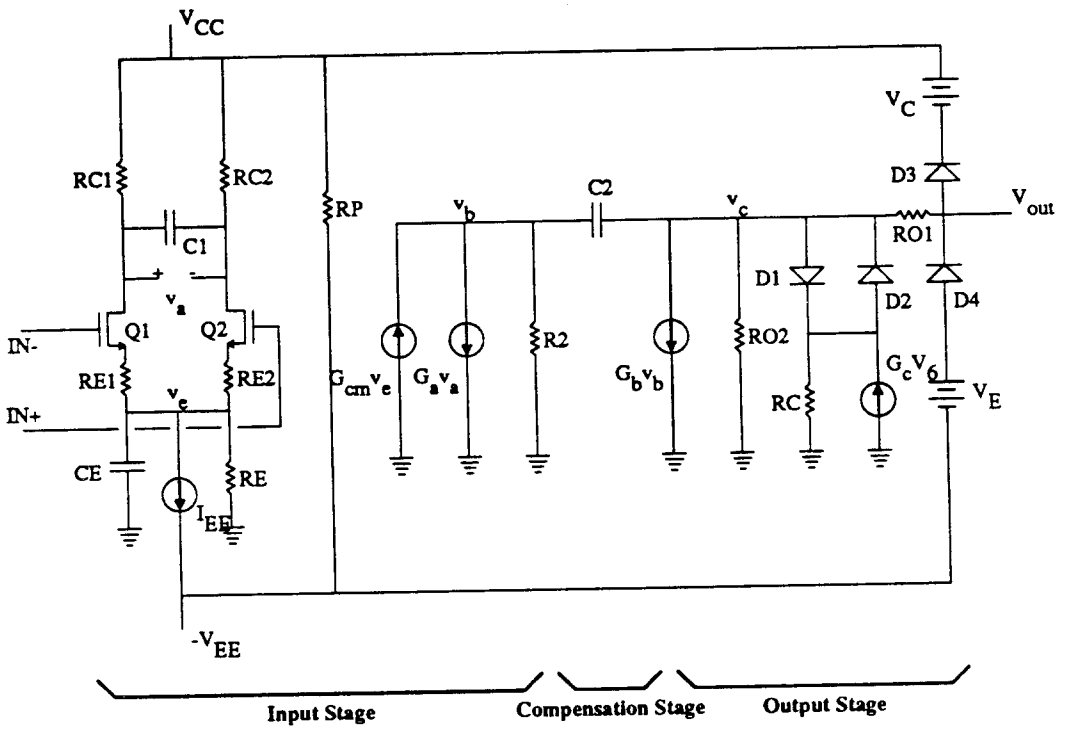


Figure 8.6 Circuit diagram of the op amp macromodel.

macromodel parameters can be found in the article by Boyle.[68]

Macromodels like the one discussed above may become important again when large systems with many op amps are modeled and optimized iteratively with an approach that requires repeated simulation of the entire circuit.

CHAPTER 9

CONCLUSIONS

Efficient methods for silicon compilation of CMOS op amps have been devised and a compiler (OPASYN) has been constructed. The synthesis starts from a set of system level performance specifications. The program selects the most promising op amp circuit in its internal database based on design requirements implied by the specifications. This circuit is parametrically optimized to determine optimal values for its design parameters. Design-rule-correct mask geometries are then constructed using a macro cell layout style. The integration of the whole layout subsystem could be accomplished in a relatively short period of time, since it has been built on top of the Berkeley CAD environment, re-using existing tools as much as possible.

Experimental results show that OPASYN produces practical circuit configurations and layouts. The entire synthesis process is fast enough to be interactively used at the system design level. The program can be easily used by system engineers who are inexperienced in op amp design. They simply specify their design requirements and optimization priorities at the system level; a few minutes later, after minimal interaction, the system offers one or more design-rule-correct mask layouts that meet functional specifications except for the most stringent designs.

The compilation task is divided into three phases: circuit topology selection, parametric optimization, and layout generation.

Circuit topology selection is based on the heuristic pruning of a decision tree that is derived by expert designers and stored in the database. The rule-based systems implementation of the selection process not only makes it convenient to update or correct decision heuristics

but also helps the user understand program's behavior and decisions.

Parametric optimization relies on analytic models specially developed for the op amp circuit topologies in the database. An analytic model contains analytic design equations that express dependency of circuit performance on design parameters. Using such analytic design equations, expensive circuit simulation in the inner loop of the optimization phase is replaced by evaluation of simple functions and their gradients. As a result, substantial reduction in CPU time has been achieved while reasonable accuracy of performance prediction is maintained.

Layout generation is based on slicing trees and a set of parameterized leaf cell generators. For each of the generic op amp topologies in the database, a slicing tree has been designed that specifies a sound topological arrangement for the building blocks of the circuit based on traditionally accumulated design experience. Parameterized leaf cell generators produce mask geometries of circuit primitives such as transistors, capacitors, and power busses in many different ways as required by the floorplanning module. Design expertise incorporated both in slicing trees and in leaf cell generators makes it possible to produce compact and reliable layouts in macro cell style. Besides, layout tools originally developed for digital circuits (a data manage, a router, a compactor, etc) have been effectively used in the layout synthesis.

Lastly, neither the optimization method nor the layout method used in OPASYN make any assumption on what kind of circuits can be represented using analytic models or using slicing trees. Thus the demonstrated approach can be applied for the synthesis of other analog building blocks such as output buffers, comparators, bandgap references even though OPASYN has so far only been used as an op amp compiler. In fact, an output buffer is already fully incorporated into the OPASYN database and this circuit is quite different from an internal op amp. Work is in progress to integrate OPASYN as a module compiler into higher-level subsystem synthesis programs such as compilers for A/D converters and for

active filters.

REFERENCES

1. P. R. Gray, "Analog ICs In the Submicron Era: Trends and Perspectives," *IEEE IEDM Digest of Technical Papers*, pp. 5-9, 1987.

2. P. E. Allen , "A Tutorial- Computer Aided Design of Analog Integrated Circuits," *Proc. of IEEE CICC* , pp. 608-616, 1986.

3. D. C. Stone *et al.*, "Analog CMOS Building Blocks for Custom and Semicustom Applications," *IEEE J. Solid-State Circuits*, vol. SC-19, pp. 55-61, February 1984.

4. T. Pletersek, J. Trontelj , and L. Trontelj, "Analog LSI Design with CMOS Standard Cells," *Proc. of IEEE CICC*, pp. 479-483, 1985.

5. G. Kelson, "Design Automation Techniques for Analog VLSI," *VLSI Design*, pp. 78-82, January 1985.

6. P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, Wiley, New York, 1984.

7. P. R. Gray and R. G. Meyer, "MOS Operational Amplifier Design- A Tutorial Overview," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 969-982, December 1982.

8. S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *SCIENCE*, vol. 220, May 1983.

9. M. A. Breuer, "Min-cut placement," *J. Design and Fault-Tolerant Computing*, vol. 1, no. 4, pp. 343-362, October 1977.

10. C. M. Fodiccoa and R. M. Mattheyses, "A Linear-time Heuristic for Improving Network Partitions," *Proc. of 19th ACM/IEEE Design Automation Conference* , pp. 175-181, 1982.

11. D. M. Schuler and E. G. Ulrich, "Clustering and Linear Placement," *Proc. 9th Design Automation Workshop*, pp. 50-56, 1972..

12. P. R. Suaris and G. Kedem, "Standard Cell Placement by Quadrasection," CS-1986-34, Department of Computer Science, Duke University.

13. H. Yaguthiel, A. Sangiovanni-Vincentelli, and P. R. Gray, "A Methodology for Automated Layout of Switched-Capacitor Filters," *Proc. of IEEE ICCAD*, pp. 444-447, 1986.

14. D. Lucas, "Analog Silicon Compiler For Switched Capacitor Filters," *Proc. of IEEE ICCAD*, pp. 506-509, 1987.

15. R. P. Sigg *et al.*, "An SC Filter Compiler: Fully Automated Filter Synthesizer and Mask Generator For A CMOS Gate-Array-Type Filter Chip," *Proc. of IEEE ICCAD*, pp. 510-513, 1987.

16. R. K. Brayton, G. D. Hachtel, and A. Sangiovanni-Vincentelli, "A Survey of Optimization Techniques for Integrated-Circuit Design," *Proceedings of IEEE*, vol. 69, no. 10, pp. 1334-1362, October 1981.

17. B. Nye, D. Riley , and A. Sangiovanni-Vincentelli, DELIGHT.SPICE User's Guide , Dept. EECS, University of California, Berkeley, May 1984.

18. J. Shyu and A. Sangiovanni, "ECSTASY: A New Environment for IC Design Dnvironment," *Proc. of IEEE ICCAD*, pp. 484-487, 1988.

19. R. Harjani, R. A. Rutenbar, and L. R. Carley, "A Prototype Framework for Knowledge-Based Analog Circuit Synthesis," *Proc. of 24th ACM/IEEE Design Automation Conference*, pp. 42-49, 1987.

20. M. G. DeGrauwe, "A Synthesis Program for Operational Amplifiers," *IEEE ISSCC Digest of Technical Papers*, pp. 18-19, 1984.

21. P. E. Allen and H. Nevarez-Lozano, "Automated Design of MOS OP Amps," *Proc. of IEEE ISCAS*, pp. 1286-1289, 1983.

22. R. J. Bowman and D. J. Lane, "A Knowledge-Based System for Analog Integrated Circuit Design," *Proc. of IEEE ICCAD*, pp. 210-212, 1986.

23. E. Berkcan, M. d'Abreu, and W. Laughton, "Analog Compilation Based on Successive Decompositions," *Proc. of 25th ACM/IEEE Design Automation Conference*, pp. 369-375, 1988.

24. H. Y. Koh, C. H. Séquin, and P. R. Gray, "Automatic Synthesis of Operational Amplifiers Based on Analytic Circuit Models," *Proc. of IEEE ICCAD*, pp. 502-505, 1987.

25. H. Y. Koh, C. H. Séquin, and P. R. Gray, "Automatic Layout Generation for CMOS Operational Amplifiers," *Proc. of IEEE ICCAD*, pp. 548-551, 1988.

26. D. J. Garrod, R. A. Rutenbar, and L. R. Carley, "Automatic Layout of Custom Analog Cells in ANAGRAM," *Proc. of IEEE ICCAD*, pp. 544-547, 1988.

27. Ping-Wai Li, "Ratio-Independent Algorithmic Analog to Digital Conversion Techniques," Ph.D. Thesis , Dept. EECS, University of California, Berkeley, August, 1984.

28. Cheng-Chung Shin, "Precision Analog to Digital and Digital to Analog Conversion Using Reference Recurculation Algorithmic Architecture," Ph.D. Thesis , Dept. EECS, University of California, Berkeley, July 1985.

29. B. K. Ahuja, "An Improved Frequency Compensation Technique for CMOS Operational Amplifiers ," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 629-633, December 1983.

30. P. R. Gray, R. W. Broderson, D. A. Hodges, T. C. Choi, R. Kaneshiro, and K. C. Hsieh, "Some Practical Aspects of Switched Capacitor Filter Design," *IEEE ISCS Digest of Technical Papers*, 1981.

31. T. Choi, R. Kaneshiro, R. W. Broderson, and P. R. Gray, "High Frequency CMOS Switched Capacitor Filters for Communications Applications," *IEEE ISCS Digest of Technical Papers*, pp. 797-802, 1983.

32. B. K. Ahuja, "Programmable CMOS Dual Channel Interface Processor for Telecommunications Applications," *IEEE J. Solid-State Circuits*, vol. SC-19, pp. 892-899, December 1984.

33. M. R. Haskard and I. C. May, *Analog VLSI Design - nMOS and CMOS*, Prentice Hall, New York, 1988.

34. K. C. Hsieh, P. R. Gray, D. Senderowicz, and D. Messerschmitt, "A Low-Noise Differential Chopper Stabilized Switched Capacitor Filtering Techinque," *IEEE J. Solid-State Circuits*, vol. SC-16, pp. 708-715, December 1981.

35. D. Senderowicz, S. F. Dreyer, J. M. Huggins, C. F. Rahim, and C. A. Laber, "Differential NMOS Analog Building Blocks for PCM Telephony," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 1014-1023, December 1982.

36. M. C. DeGrauwe, J. Rijmenants, E. A. Vittoz, and H. J. De Man, "Adaptive Biasing CMOS Amplifiers," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 522-528, June 1982.

37. M. A. Copeland and J. M. Rabaey, "Dynamic μ Amplifiers for MOS Technology," *Electronic Lett.*, vol. 15, pp. 301-302, May 1979.

38. B. J. Hostica, "Dynamic CMOS Amplifiers," *IEEE J. Solid-State Circuits*, vol. SC-15, pp. 887-892, October 1982.

39. G. D. Hachtel and P. Zug, APLSTAP - Circuit Design and Optimization System - User's Guide, Technical Report, IBM Yorktown Research Facility, 1981.

40. M. G. DeGrauwe *et al.*, "An Analog Expert Design System," *IEEE ISSCC Digest of Technical Papers*, pp. 212-214, 1987.

41. F. M. El-Turky and R. A. Nordin, "BLADES: An Expert System For Analog Circuit Design," *Proc. of IEEE ISCAS*, pp. 552-555, 1986.

42. R. Harjani, R. A. Rutenbar, and L. R. Carley, "Analog Circuit Synthesis for Performance in OASYS," *Proc. of IEEE ICCAD*, pp. 492-495, 1988.

43. A. Vladimirescu, K. Zhang, A. R. Newton, D. O. Pederson, and A. Sangiovanni-Vincentelli, *SPICE Version 2G - User's Guide*, Dept. EECS University of California, Berkeley, 1981.

44. L. W. Nagel and D. O. Pederson, "Simulation Program with Integrated Circuit Emphasis," *Proc. Sixteenth Midwest Symposium on Circuit Theory*, Waterloo, Canada, April 12, 1973.

45. D. Harrison, P. Moore, R. L. Spickelmier, and A. R. Newton, "Data Management and Graphics Editing in the Berkeley Design Environment," *Proc. of IEEE ICCAD*, 1986.

46. P. Moore, "OCT Database Programmer's Manual," Internal Memorandum, University of California, Berkeley, 1986.

47. T. Laidig, "UNIX Programmer's Manual," Internal Memorandum, University of California, Berkeley, 1988.

48. P. Moore, "UNIX Programmer's Manual," Internal Memorandum, University of California, Berkeley, 1988.

49. H. Shin and A. Sangiovanni-Vincentelli, "MIGHTY: A 'Rip-Up and Reroute' Detailed Router," *Proc. of IEEE ICCD*, pp. 10-13, 1986.

50. S. J. Seda, M. G. DeGrauwe, and W. Fichtner, "A Symbolic Analysis Tool for Analog Circuit Design Automation," *Proc. of IEEE ICCAD*, pp. 488-491, 1988.

51. I. Bratko, *PROLOG - Programming For Artificial Intelligence*, Addison-Wesley, Reading, 1986.

52. E. Charniak, *Artificial Intelligence*, Addison-Wesley, Reading, 1986.

53. P. H. Winston, *Artificial Intelligence*, Addison-Wesley, Reading, 1984.

54. P. H. Winston and B. K. P. Horn, *LISP*, Addison-Wesley, Reading, 1984.

55. G. Jusuf, "Synthesis of Fully Differential Folded Cascode With Autozeroing Circuit," EE219 Class Project Report, University of California, Berkeley, December 1988.

56. C. T. Chuang, "Analysis of the Settling Behavior of an Operational Amplifier," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 74-80, February 1982.

57. B. Y. Kamath, R. G. Meyer, and P. R. Gray, "Relationship between Frequency Response and Settling Time of Operational Amplifiers," *IEEE J. Solid-State Circuits*, vol. SC-9, pp. 347-352, December 1974.

58. A. B. Grebene, *Bipolar and MOS Analog Integrated Circuit Design*, Wiley, New York, 1984.

59. D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, 1984.

60. W. H. Press et al., *Numerical Recipes in C - The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1988.

61. W. T. Nye and A. L. Tits, "An Application-Oriented, Optimization-Based Methodology for Interactive Design of Engineering Systems," *International Journal of Control*, vol. 43, no. 6, pp. 1693-1721, 1986.

62. K. Madsen, H. Schjaer-Jacobsen, and J. Voldby, "Automated Minimax Design of Networks," *IEEE Trans. on Circuits and Systems*, vol. CAS-22, pp. 791-795, October 1975.

63. R. Otten, "Laout Compilation," in *Design Systems for VLSI Circiits - Logic Synthesis and Silicon Compilation*, ed. G. De Micheli, A. Sangiovanni-Vincentelli, and P.

Antognetti, Martinus Nijhoff Publishers, Dordrecht, 1987.

64. L. Stockmeyer, "Optimal Orientations of Cells in Slicing Floor-plan Designs," *Information and Control*, vol. 59, pp. 91-101, 1983.

65. D. A. Hodges and H. G. Jackson, in *Analysis and Design of Digital Integrated Circuits*, McGraw-Hill, New York, 1983.

66. J. L. Burns and A. R. Newton, "SPARCS: A New Constraint-Based IC Symbolic Layout Spacer," *Proc. of IEEE CICC*, pp. 534-539, 1986.

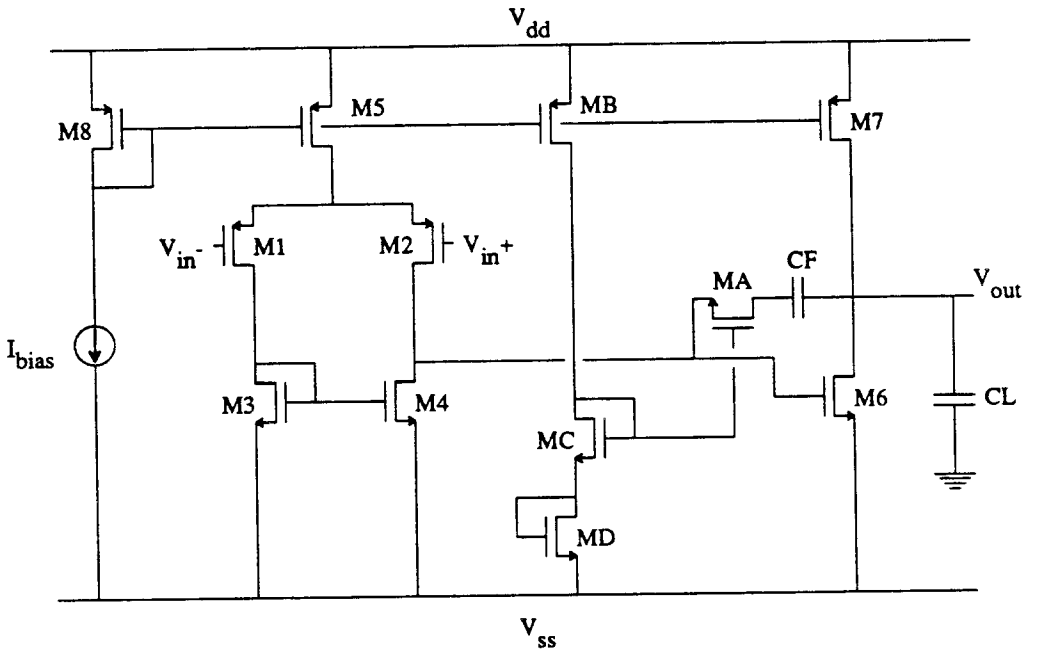
67. G. Hamachi, R. Mayo, J. Ousterhout, W. Scott, and G. Taylor, "UNIX Programmer's Manual," Internal Memorandum, University of California, Berkeley, February 1986.

68. G. R. Boyle, B. M. Cohn, D. O. Pederson, and J. E. Solomon, "Macromodeling of Integrated Circuit Opereational Amplifiers," *IEEE J. Solid-State Circuits*, vol. SC-9, pp. 353-364, December 1974.

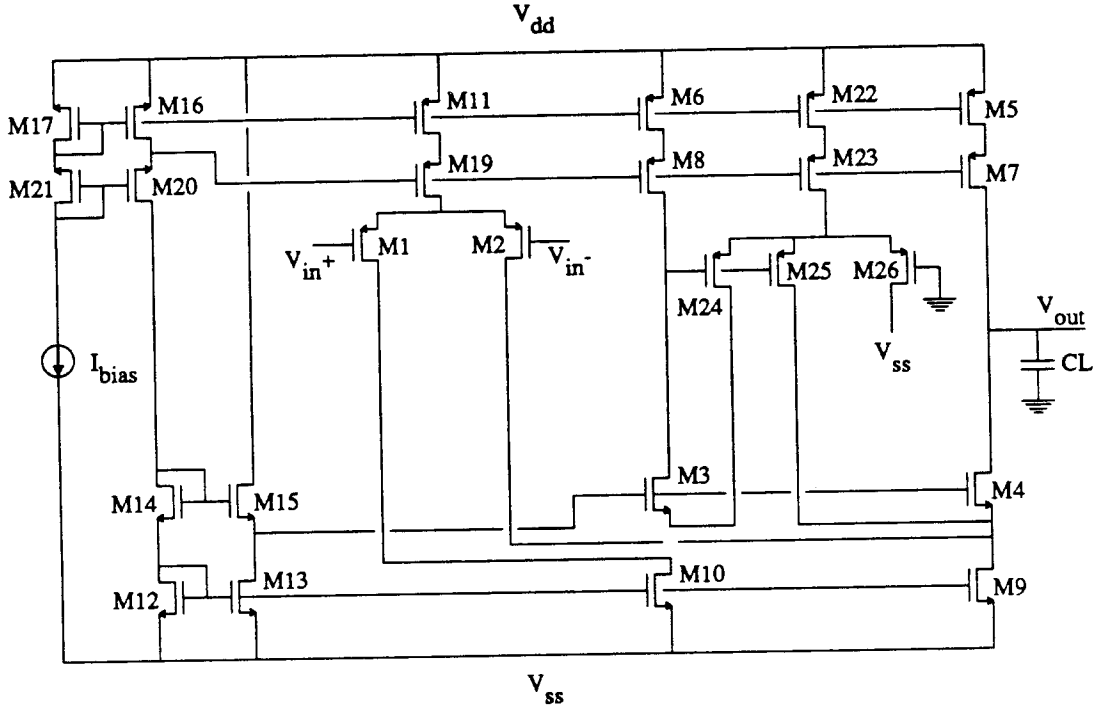
69. L. O. Chua and P-M. Lin, in *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*, Prentice-Hall, Englewood Cliffs, 1975.

APPENDIX

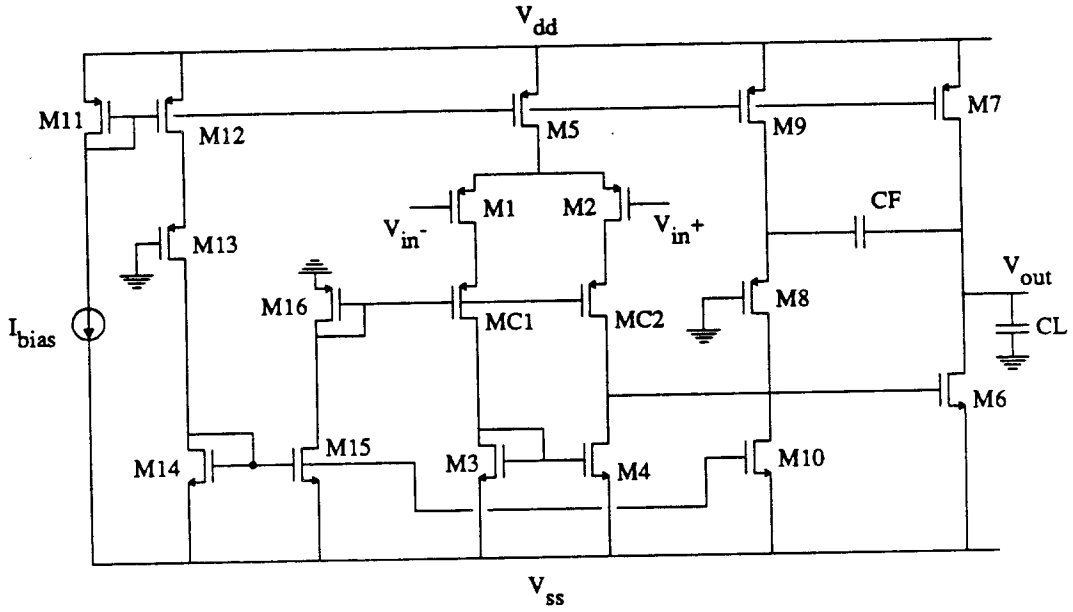
CIRCUIT SCHEMATICS



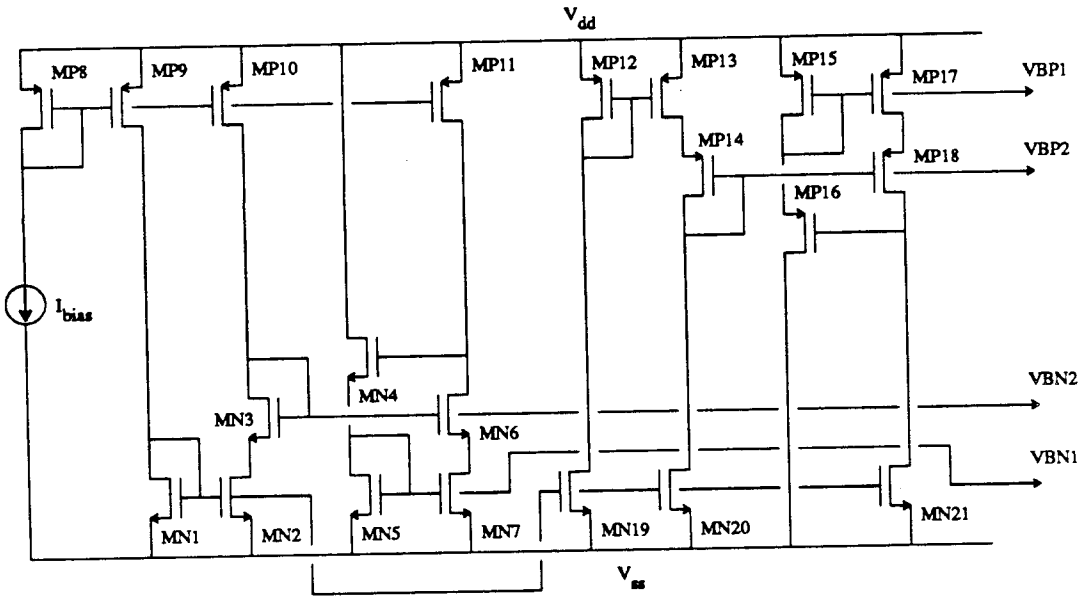
A.1. Basic two-stage (BTS) OP amp.



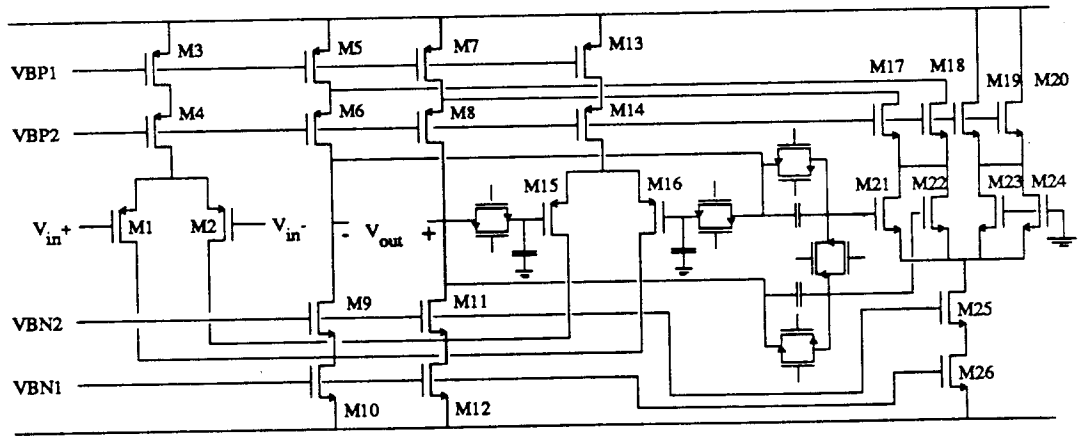
A.2. Single-stage folded cascode (SFC) OP amp.



A.3. Cascoded two-stage (CTS) OP amp.

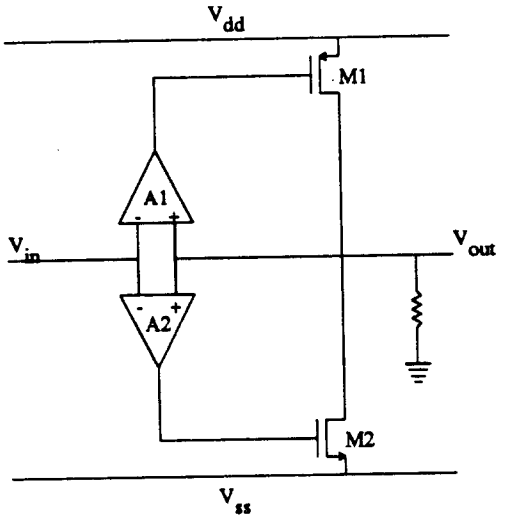


(a) Bias circuit

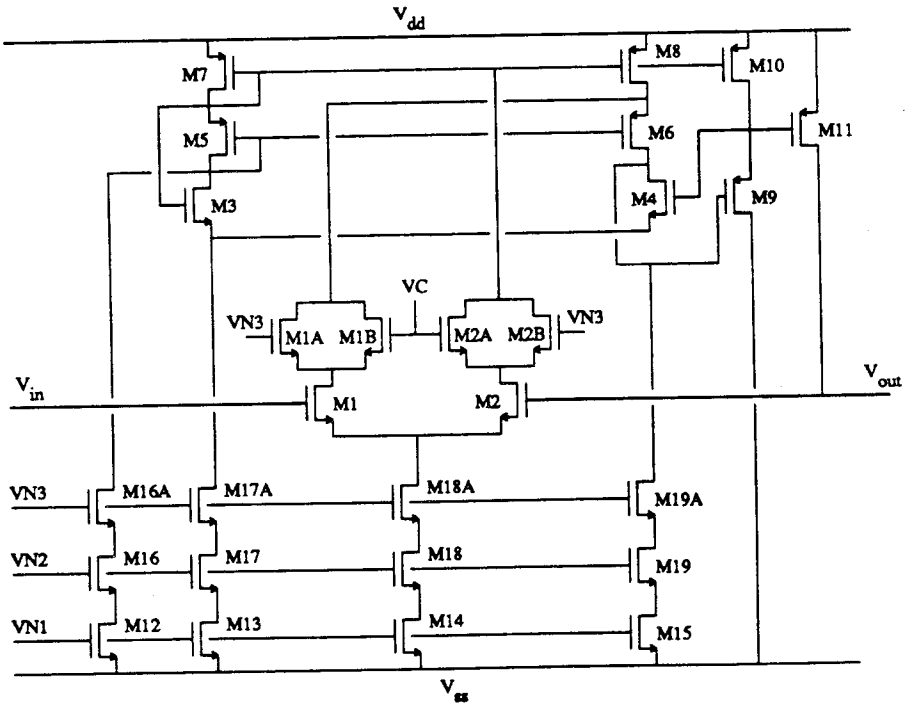


(b) Amplifier

A.4. Fully differential folded cascode (FFC) OP amp.



(a) Conceptual schematic



(b) Circuit schematic of one composite device

A.5. Class AB complimentary output buffer.