# AUTOMATIC SYNTHESIS AND LAYOUT OF SWITCHED-CAPACITOR FILTERS

by

Hormoz Yaghutiel

Memorandum No. UCB/ERL M88/56

16 August 1988

**ELECTRONICS RESEARCH LABORATORY**

College of Engineering
University of California, Berkeley
94720

# Automatic Synthesis and Layout of Switched-Capacitor Filters

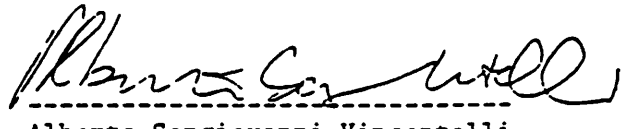
Hormoz Yaghutiel

Ph.D.

Department of Electrical Engineering
and Computer Science

## Abstract

Design cycle time for complex analog and mixed analog/digital Integrated-Circuits (IC's) is expected to be drastically reduced with the help of computer-aided design tools. Unfortunately, very few steps have been taken towards automatic generation of efficient analog circuits. An important class of analog IC's whose manual design is particularly time consuming is switched-capacitor filters. This thesis describes a switched-capacitor filter module generator which can produce optimized layouts for these circuits, thus saving large amounts of manual design time. Previous efforts in the area of reducing design cycle times have often resulted in increased areas and reduced performance in comparison with manual designs. The system described in this thesis exploits various degrees of freedom in the design of switched-capacitor filters to generate efficient circuits. Special emphasis has been placed on the physical design aspect of module generation, due to the heavy dependence of analog performance on fine details of layout. Experimental results for this system indicate that generation of area-efficient switched-capacitor filters with good quality is possible in reasonable CPU times.



Alberto Sangiovanni-Vincentelli
Thesis Committee Chairman

# Acknowledgments

I wish to express my deepest gratitude to my advisor Professor Alberto Sangiovanni-Vincentelli for his continuous support during the course of my Ph.D. studies, and to Professor Paul Gray for his guidance in the field of analog integrated circuit design. I would also like to thank them along with Professor Shoshichi Kobayashi for serving on my dissertation committee. The help of Professors Robert Brodersen and David Hodges is also gratefully acknowledged. I am also indebted to DARPA for financial support received under Grant N00039-87-C-0182.

The help and friendship of many fellow students have contributed to this work and made my years at Berkeley more enjoyable. Homayoon Ansari, Farhad Ayrom, Hamid Bahadori, Saman Behtash, Jeff Burns, Giorgio Casinovi, Andrea Casotto, Ramin Esmail-Beygui, Tammy Huang, Reza Kazerounian, Young Kim, Ramin Khalili, Haideh Khorramabadi, Davar Khoshnevisan, Ken Kundert, Steve Lewis, Kartikeya Mayaram, Linda Milor, Omid Razavi, Dave Riley, Fabio Romeo, Hamid Savoj, Shahram Shahruz, Scott Shen, Hyunchul Shin, Jyuo-Minh Shyu, Sehat Sutarja, Chorng-kuang Wang, Ray Wei, and Nick Weiner are just a few of the names that come to mind.

My most sincere thanks to my best friend Susie, who has endured many of the hardships all along with me. Finally, I like to thank my parents for their support and patience during these long years.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

Recently, there has been an increasing interest in the development of Computer-Aided Design (CAD) tools for analog Integrated Circuits (ICs). This can be attributed partly to the need, as dictated by economic factors, to integrate digital and analog functions on the same chip. Such chips open up applications in telecommunications, data storage, speech processing, robotics, and other analog fields. Design time for complex digital ICs has been dramatically reduced with the help of CAD tools. Without such tools for analog circuits, which typically occupy a small area of the chip, the analog design can become a bottle-neck for the overall design of the chip.

In some applications such as filtering, there is a choice between a digital or an analog solution. For the current 1.25 micron CMOS technologies, a well designed analog circuit typically occupies smaller area and consumes less power than its digital counterpart. The digital circuit, on the other hand, is capable of realizing more accuracy. These trade-offs are usually over-shadowed by the fact that most IC designers are oriented towards digital design, and more effort is required for analog design. Development of CAD tools, and in particular design automation tools, for analog ICs will greatly reduce the design effort and amount of expertise needed.

An important class of analog ICs whose manual design and layout are particularly time consuming are Switched-Capacitor (SC) filters [1]. In this thesis we focus on automatic synthesis, and specifically physical design of these filters.

## 1.2 Analog Versus Digital

In analog systems the signals, which are in the form of voltages, currents, and charges, are continuous functions of time. By contrast, in digital systems each signal is represented by a sequence of finite number of binary digits; therefore, these signals can take on discrete values only. Due to the binary nature of signals, digital circuits are realized using gates with only two states, each state being defined in some range of the continuous signal. This makes digital circuits to a large degree immune to various noise sources inherent in integrated circuits. Then the design effort can be directed mainly towards trade-offs between power consumption, speed, and area. Since in digital circuits, devices act as switches moving information between storage elements, they can often have the minimum size permissible in the process; only those devices on critical paths or driving large loads need to have radically different sizes.

Analog circuits in general require more design freedom in order to be applied effectively. They often exploit the full spectrum of capabilities exhibited by individual devices. In an analog circuit the individual devices often have substantially different sizes and electrical characteristics. These circuits require optimization of various performance measures. As an example, among the performance measures for operational amplifiers are gain, bandwidth, noise, power supply rejection, dynamic range, offset voltage, and so on. The importance of each performance measure depends on circuit application. For this reason, fine tuning plays a crucial role in design of analog circuits.

Because IC wafer fabrication processes do not produce active and

passive devices with the same stability and accuracy as discrete components, analog designers over the years have developed circuits which cancel out the first order variations in key parameters. This means, however that the analog circuits then become sensitive functions of the second order variations of such parameters, for example the matching of input devices in differential pairs, or capacitor matching in SC filters. Sensitivities to second order variations impose constraints on design and specially layout of analog circuits. These constraints are in general not present in digital design.

## 1.3 CAD and Design Automation

The continued shrinking of device dimensions has made it possible to integrate more functions on the same chip. This increased complexity has in turn necessitated development of CAD tools to assist designers and reduce design cycle times and costs. Development of efficient CAD tools has depended on structured design strategies to reduce the complexity of IC design. Some properties motivating these strategies are [2]:

**Modularity** Modularization involves dividing a system into a set of "well formed" modules such that the interactions between these modules are well characterized. For example from a layout point of view, the phrase "well formed" corresponds to a well defined physical interface that indicates the position, name, layer, type, size, and signal type of external interconnections. Modularity helps the designer to clarify and document an approach to a problem, and also allows a designer to be of more utility by checking attributes of a module as it is constructed. The ability to divide a task into a set of well-defined modules also aids in a team design where a number of designers have a portion of a complete chip to design.

**Hierarchy** The use of hierarchy involves dividing a module into submodules and then repeating this operation on the submodules until the com-

plexity of the submodules is at an appropriately comprehensible level of detail. Hierarchy also refers to the abstraction levels through which a design must pass. For example, the various abstraction levels in digital design are behavioral, register-transfer, logic-gate, switch, transistor, and mask-layout.

**Regularity** The use of iteration to form arrays of identical cells is an example of the use of regularity in an IC design. However, extended use may be made of regular structures to simplify the design process. For example if one was constructing a "data-path," the interface between modules (power, ground, clocks, busses) might be common but the internal details of modules may differ according to function.

**Locality** By defining well-characterized interfaces for a module, we are effectively stating that the internals of the module are unimportant to any exterior interface. In this way we are performing a form of information hiding that reduces the apparent complexity of that module.

These properties have been extensively used in the development of strategies and tools for simulation, synthesis, layout, and testing of digital circuits. For example simulation tools have been developed for every level of design hierarchy. This makes it possible to verify the functionality of large digital designs by simulating entire systems at behavioral level, the complex modules at logic-gate level, and so on. As another example, the combination of modularity and regularity has contributed to development of standard-cell and gate-array layout styles. These styles have proven more amenable to design automation than irregular structures.

Modularization is a common practice in manual design of analog circuits. For example, an A/D converter is usually described in terms of its operational amplifiers, comparators, matching capacitors, matching resistors, and switches rather than the individual devices making up the circuit. Hierarchy is also employed in analog circuit design. An A/D converter is

composed of common functional blocks such as operational amplifiers and comparators. These functional blocks are themselves composed of simpler common blocks such as current mirrors and differential pairs. At the lower end of the hierarchy we find individual devices such as transistors. resistors. and so on.

These properties serve as motivation for development of analog synthesis environments similar to the ones existing for digital design. An example of a digital synthesis system is the work done at Berkeley. This system accepts a behavioral or algorithmic description of a digital system and produces a circuit mask layout through four synthesis steps: behavioral, structural, logic [3], and layout [4]. A more detailed structure of this system is presented in figure 1.3. In this system the various modules communicate through the Oct data manager [5].

Since analog problems are very different from digital, their design automation requires new methodologies and development of new tools. In the next section the proposed approaches to automatic synthesis of analog circuits will be discussed.

## 1.4 CAD Systems for Analog IC Design

The field of CAD for analog circuits has progressed at a considerably slower rate than has been the case for digital. Part of the reason has been the heavier dependence of analog performance on fine details of layout and device behavior. Another reason might be the present difficulty to identify a level of abstraction where generic models such as the ones developed for digital synthesis can be derived. While for digital circuits. several steps have been taken towards automatic generation of efficient circuits, for analog functions only few results are available. So far most of the work has been directed mainly towards reducing design cycle time, often at the expense of area and performance.

The system reported by Allen [6] can generate layout for some range

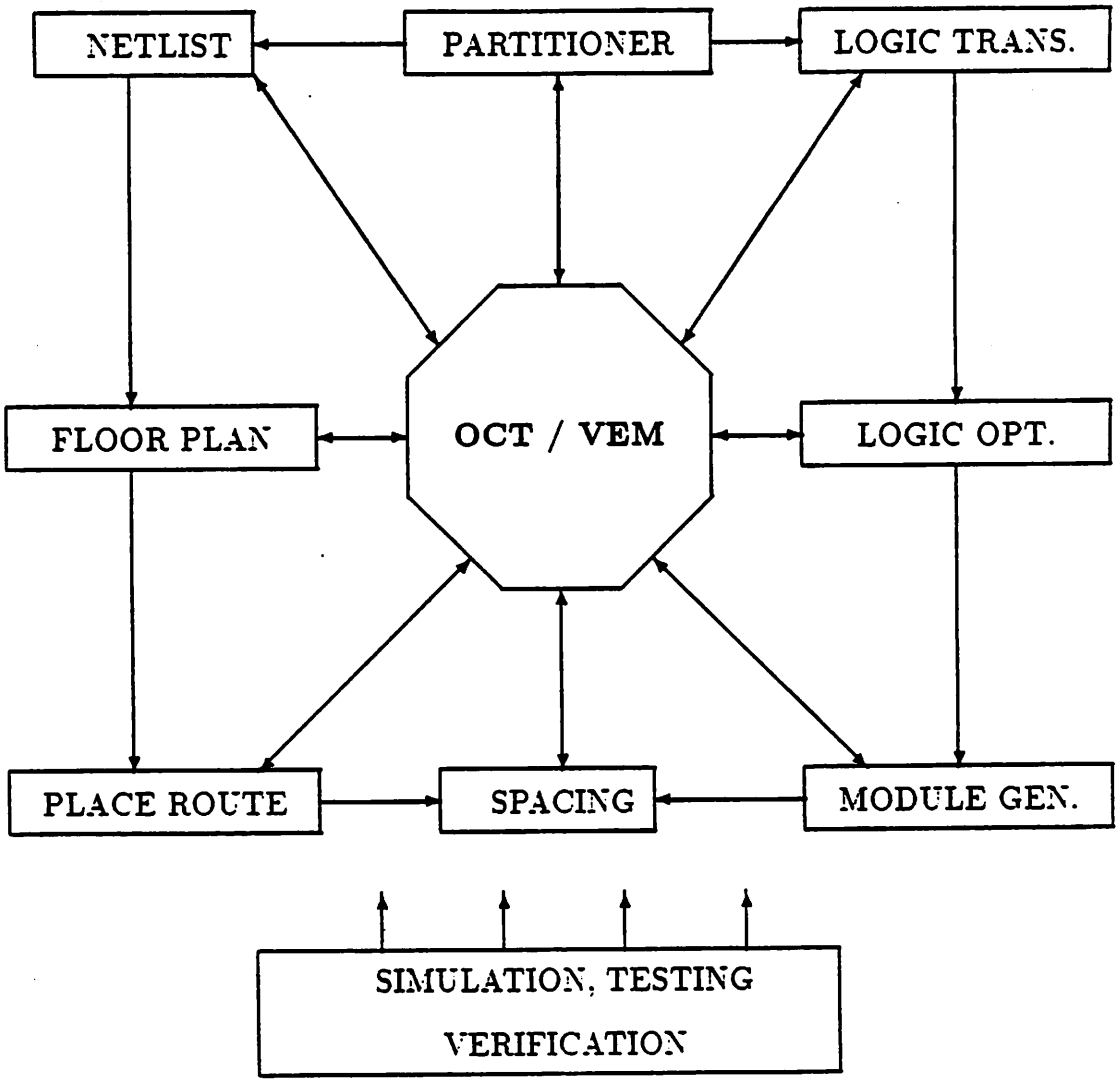



Figure 1.1: The Berkeley Synthesis System.

of linear, nonlinear, continuous and discrete-time ICs. Among its capabilities are SC filters and successive approximation A/D and D/A converters [7]. It can also produce circuit simulation files for the generated circuits. These circuits are realized in a standard cell layout style by interconnections of fixed and programmable pre-designed blocks.

In another system [8], a library of analog standard cells contains the most common circuit elements including analog switches, operational amplifiers, comparators, peak detectors, current controlled oscillators, D/A converters and delay cells. These cells have fixed heights with power supply rails running along the top and bottom. To generate more complex analog functions, these cells are automatically placed and routed in a standard cell layout style. Isolation of digital and large swing analog signals from sensitive analog signals has been achieved by special design of library cells and by a placement strategy to guarantee the containment of sensitive nets in a local routing channel.

In the methodology for layout generation of analog circuits presented in [9], the circuit is partitioned into functional blocks consisting of library cells previously developed. The automatic placement and routing is again performed in a standard cell layout scheme. This system allows the user to provide an incomplete specification of the basic cells required for the design. It then scans the library for the best match between the specified description and the cells in the library.

DeGrauwe [10] reported a system which generates device dimensions and bias currents for a library of analog schematics as a function of technology and desired specifications. The basic library includes 12 transconductance amplifiers, 5 voltage references, 4 operational amplifiers, 3 low-noise BIMOS amplifiers, 1 quartz oscillator and 2 oversampling A/D converters including a digital decimation filter. In this approach the expert knowledge for each type of circuit is hard coded in the framework and also analytical equations are derived for each circuit in the library, which is then used to optimize the performance of the circuit.

In the knowledge-based approach to analog synthesis reported in [11]. circuit schematics including device sizes are produced for common analog functional blocks. The framework is based on the idea that the circuit topologies are selected from among fixed alternatives; they are not constructed transistor-by-transistor for each new design. A topology for a high-level module is specified as an interconnection of subblocks, not as an interconnection of transistors. Based on this approach a prototype op-amp synthesis system has been presented.

The approach we have adopted to automatic synthesis of analog circuits, is to focus on techniques that allow design of circuits with efficiency comparable to the ones manually designed by skilled designers. To realize this goal the problem of automatic synthesis is decomposed into three closely coupled stages. In the first stage the components of the design are chosen to meet the given performance specifications. For example these components for a telecommunication chip could be an A/D converter with specified accuracy and speed. a set of switched-capacitor filters, and a DSP processor. In the second stage the components are synthesized by module generators to meet the goals set in the first stage [12,13]. In parallel with the first two stages, simulation has to be performed at the system and component level [14,15, 16,17] (figure 1.2).

This process can be hierarchical in the sense that the modules such as SC filters can be designed with a similar process. The components of the filter are chosen and designed to meet the specifications and then can be automatically generated, placed, and routed (figure 1.3). The only difference is modules such as SC filters and A/D converters are well studied and much is known about the effects of the component selection and architecture on the system level performance.

Optimization is an important part of module generation [18]. Most often the synthesis step does not take into account the second order effects. In the optimization step the design parameters are fine tuned so that the synthesized circuit with parasitic effects included. meets the specifications.

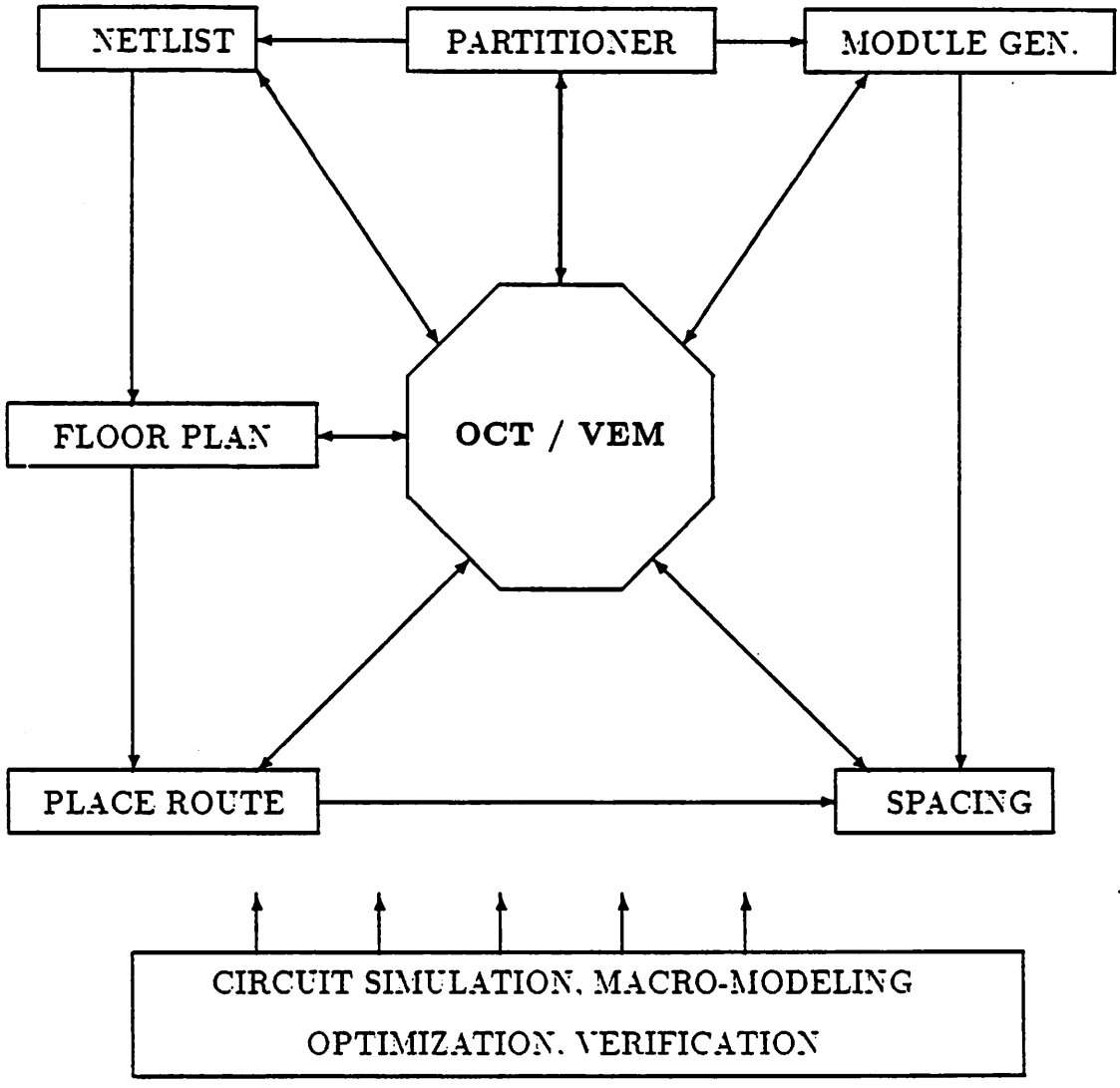



Figure 1.2: An analog synthesis system.

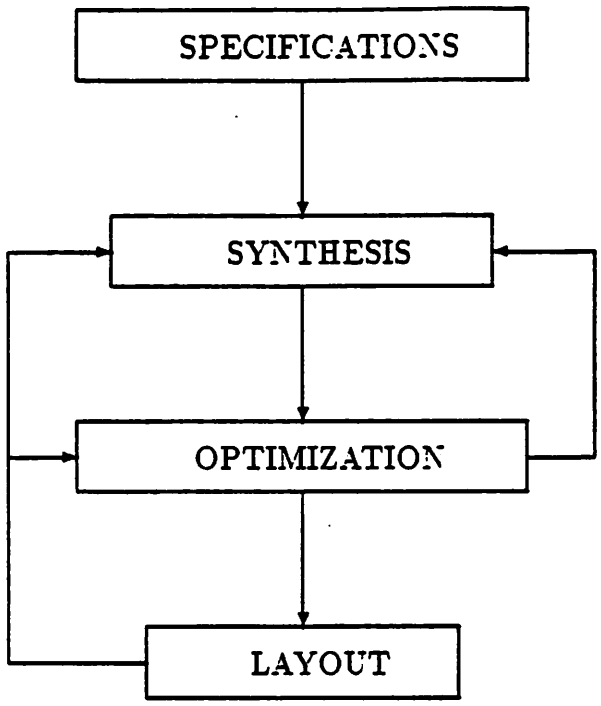



Figure 1.3: The process of module generation.

In this thesis the module generation aspect of a synthesis system is studied.

## 1.5 Contribution of the Thesis

Filtering is an important operation in many electronic systems. In audio, telecommunications and a wide spectrum of other applications filters have been used extensively for pre- and postprocessing of signals. SC filters are analog sampled data ICs implemented in MOS technology. They are capable of precision frequency-selective filtering and can be integrated with digital LSI circuits. Since their advent in the late seventies, these filters have matured to a great degree.

Since SC filters can be integrated with digital circuits, they are in direct competition with digital filters. SC filters are generally smaller and consume less power than similar digital filters in the context of current 1.25

micron technologies. and unlike digital filters, they do not require the large front end A/D converters. As a result of the analog nature of SC filters. their performance is heavily dependent on fine details of device behavior and layout. This combined with their time varying nature has made their manual design and layout particularly time consuming, therefore making them a prime candidate for design automation.

A number of systems have been reported for automatic synthesis of these filters. Some of these systems are capable of producing layouts for the generated filters. However. they all have limited use as a result of their layout strategies. These limitations will be discussed in detail in chapter 3. Our goal in this project was to study the feasibility of a fairly general automated layout system for SC filters with very little sacrifice of area efficiency or performance. This study led to the development of a technology and topology independent system for generation of efficient single-ended SC filters. The main body of the thesis is dedicated to presenting this layout system.

## 1.6 Organization of the Thesis

In chapter 2. the principles of operation of SC filters are reviewed. It is followed by a discussion of current approaches to synthesis of these filters. Then a step-by-step procedure for synthesis of SC ladder filters is presented. The final section of this chapter is a survey of tools for synthesis of these filters. In chapter 3. I introduce our strategy for automated layout of SC filters. I begin this chapter with discussing the issues involved in the layout of SC filters. Next, I present a survey of reported SC layout tools. Then. the merits of the new layout strategy are discussed. It is followed by our choice of input format. Finally the organization of the layout system is reviewed. To exploit this strategy. we have solved a set of layout problems. This is the topic of the next two chapters. In chapter 4, the problems associated with generating area/perimeter constant capacitor layouts to meet the proposed layout strategy are discussed in detail. To solve these problem, several algo-

rithms are developed. In this chapter the performance of each algorithm is analyzed both mathematically and experimentally. In chapter 5, the placement and routing stages of layout generation are presented. First algorithms for linear placement of modules to minimize the routing area and improve the quality of the routing are presented. Then, the features of MIGHTY [19], the channel router employed by the system, are discussed. I finish this chapter by presenting the method we have developed to shield the unwanted crossovers in the routing channels. In chapter 6, experimental results on a set of prototype filters generated using the layout system and fabricated in a 3 micron CMOS technology are discussed. As a consequence of our experiments, we observed a trade-off between the total capacitance and power supply rejection of the filters. Chapter 7 is a discussion of extension of ideas of this thesis to other applications, such as fully-differential architectures and A/D converters.

# Chapter 2

# Synthesis of Switched-Capacitor filters

## 2.1 Introduction

The basis for operation of switched capacitor filters originated from the observation that if a capacitor is periodically switched between two voltage sources, it acts as a resistor between the two sources [20]. In other words, the switches transform the capacitor, a non-dissipative memory element, into a dissipative memoryless one. In Figure 2.1 capacitor $C$ is periodically switched between two voltage sources $V_1$ and $V_2$. Then the average current

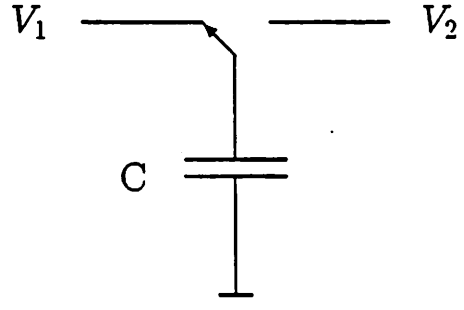



Figure 2.1: A switched capacitor.

flowing from $V_1$ to $V_2$ is $C(V_1 - V_2)f_c$ where $f_c$ is the switching frequency. Therefore the switched capacitor behaves like a resistor having a value $\frac{1}{Cf_c}$. This idea was later incorporated into the development of circuit techniques in MOS technology where the fundamental precision of the analog quantity was defined by the ratio of two on chip MOS capacitors. These circuits are typically comprised of capacitors, operational amplifiers, and clock-controlled switches. With proper process design, the monolithic MOS capacitor shows very stable characteristics with respect to changes in temperature and voltage. While the absolute value of the capacitance exhibits random processing variations of the order of 10-20 percent, with proper layout techniques, the MOS capacitor ratio can be made accurate to within 0.1 percent.

Initial SC filters suffered from sensitivity to non-linear parasitic capacitances associated with various nodes in the circuit. For this reason, the capacitors had to be designed substantially larger than these parasitics in order to maintain the analog precision. This contributes to large filter areas and severely limits the frequency range of operation of the filter by heavily loading the operational amplifiers. This shortcoming was overcome by the advent of parasitic-insensitive circuit techniques [21]. This chapter starts with presenting a parasitic-insensitive SC integrator. followed by discussing the general parasitic-insensitivity condition for two phase SC filters.

There have been a number of design techniques reported for SC filters. The two most common SC filter realizations are cascade of second order sections and ladder-type designs. In section 2.4 the two techniques are discussed.

The sampled-data nature of SC filters calls for their design to be carried out in the discrete frequency ($z$) domain [22]. Nevertheless, a common practice in the design of SC filters is to evaluate the filter's capacitor ratios directly from the time constants associated with the continuous active filter having the desired frequency response. The frequency response of the resulting SC filter closely approximates the desired response if the filter time constants are many times smaller than the sampling frequency. This

assumption is supported by the following argument:

> To avoid aliasing effects, the incoming signals to SC filters should be band-limited to less than half the sampling frequency. This is typically carried out by a continuous-time active pre-filter. In order to ease the pre-filtering requirements, the frequency range of operation of the filter is kept many times smaller than the sampling frequency.

The approximation method of synthesis is illustrated in section 2.5 by presenting an approach to the deign of ladder-type filters. First, a systematic approach to map the continuous LC ladder filter to interconnections of integrators is presented. This section also deals with mapping of finite transmission zero elements into capacitive coupling of integrators. Then, mapping of continuous element values into capacitor ratios is discussed. It is followed by discussion of optimization of dynamic range and capacitor ratio spread via circuit transformations. The factors involved in choosing the switching scheme will also be studied. Finally adjustment of capacitor ratios for design centering is considered.

In some cases the designs resulting from approximate synthesis are not acceptable. One solution is to fine-tune the capacitor ratios to match the desired response. This can be achieved by using mathematical programming techniques [23]. Another approach is to design the filter directly in the z-domain. These techniques have been employed in the development of a number of automatic SC filter synthesis systems. In the final section, these systems will be discussed.

## 2.2 LDI Bottom-Plate Integrator

A switched-capacitor filter can be viewed as an active RC filter, where each resistor has been replaced by a combination of a capacitor and few switches. The switches are implemented by MOS transistors with clock controlled gates. Since the basic building block of an active RC filter is

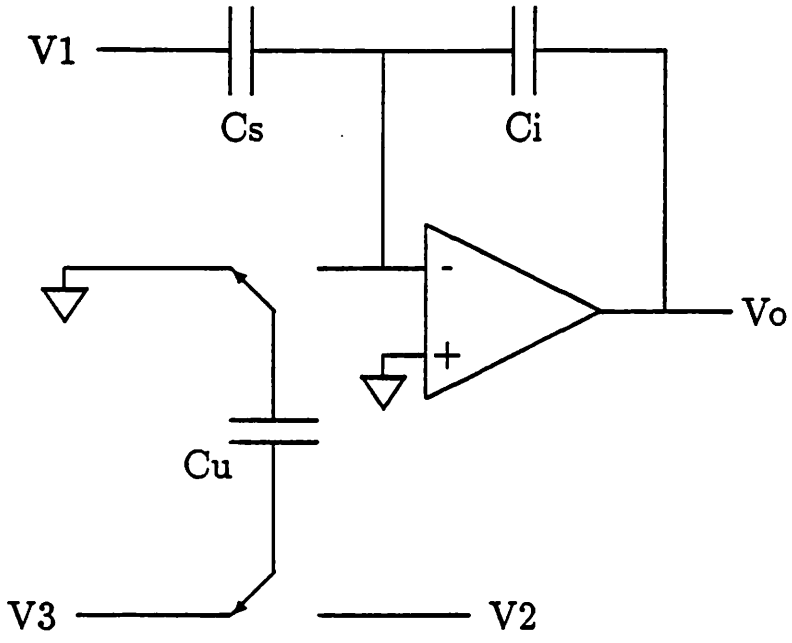



Figure 2.2: A switched-capacitor integrator/summer.

active integrator/summer, we can gain insight into the operation of SC filters by examining an SC implementation of integrator/summer (figure 2.2). Assuming ideal elements, the circuit of figure 2.2 can be described by the following z-transform equation [24]:

$$V_o = -\frac{C_s}{C_i}V_1 + \frac{\frac{C_u}{C_i}}{1-z^{-1}}(z^{-\frac{1}{2}}V_3 - V_2) \qquad (2.1)$$

If the fastest fluctuations of the input voltage are much slower than the clock frequency, namely $sT \ll 1$, where $T$ is one clock period, then

$$1 - z^{-1} = 1 - e^{-sT} \approx sT,$$

$$z^{-\frac{1}{2}} \approx 1 - \frac{sT}{2} \approx 1 .$$

Substituting these approximate expressions in Equation 2.1 we find:

$$V_o \approx -\frac{C_s}{C_i}V_1 + \frac{C_u}{sTC_i}(V_3 - V_2) \qquad (2.2)$$

The above expression shows that this circuit acts as an integrator/summer with a time constant of $\frac{C_u}{TC_i}$ for the range of frequencies $s \ll \frac{1}{T}$. Thus, the operation of the circuit depends on the capacitor ratios and not on their absolute values. This remains true when several integrating blocks are interconnected to form more complicated circuits. As discussed earlier, an accuracy of the order of 0.1% can be achieved for the ratio of two on chip MOS capacitors [25]. The clock frequency can also be controlled very accurately. Thus, the time constants of SC filters can be realized with excellent accuracies.

The most common use of SC filters has been in the voice band applications, where the required filter time constants are on the order of 10 Krad/sec. Integrated active RC filters requiring such time constants, take up prohibitively large silicon areas, typically hundreds of times larger than what would be required for SC implementation of these filters.

For sufficiently large op-amp gains, this SC integrator is insensitive to the harmful effects of stray capacitances between various nodes of the circuit and ground. For this reason, this integrator known as LDI bottom plate integrator, has been extensively used in the design of practical SC filters. In the following section the general condition of stray-insensitivity for SC filters will be presented.

## 2.3 Stray-Insensitive Switched-Capacitor Filters

In SC filters, parasitic capacitances exist at every node of the circuit. Each parasitic capacitance affects some performance parameter of the filter. Consider again the SC integrator/summer discussed above, where the parasitic capacitances associated with each node are represented by a single lumped capacitor from that node to the ground (figure 2.3). The parasitics $C_{p1}, C_{p2}, C_{p3}, C_{p4}$, and $C_{p8}$ add to the capacitive loading of the various oper-

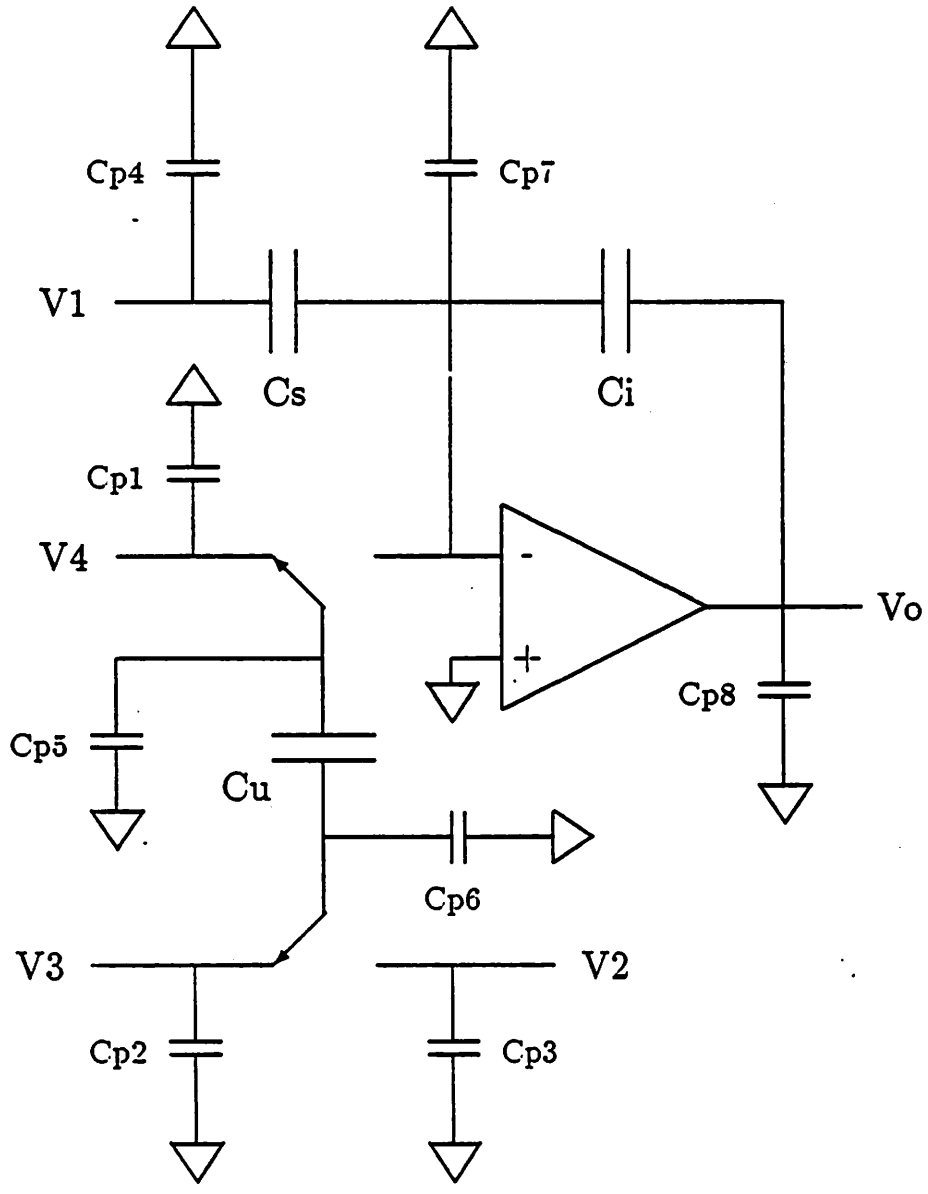



Figure 2.3: An integrator/summer with nodal parasitic capacitances.

ational amplifiers. They should be taken into account when designing the op-amps. The parasitics $C_{p5}$ and $C_{p6}$ must be included in calculating the RC time constants when designing the switch sizes. Parasitic capacitance $C_{p7}$ is generally comprised of a component to the negative supply and a component to the positive supply. Therefore, this parasitic is critical in determining the power supply rejection (PSR) of the integrator and hence, that of the entire SC filter.

If the op-amps and switches are designed correctly, the parasitics mentioned so far, except for $C_{p5}$, do not affect the fundamental precision of the filter. However, if $V_4$ is used as an inverting input to the integrator, $C_{p5}$ is charged in parallel with $C_s$, thus changing the gain constant associated with that input. So, it becomes apparent that $V_4$ should be connected to analog ground if the integrator is to be parasitic-insensitive. In that case, $C_{p5}$ is switched between analog ground and the virtual ground of the amplifier, and has no effect on the frequency response of the filter. Thus, this integrator becomes to a first order parasitic-insensitive, if $V_3$ and $V_2$ are used as the non-inverting and inverting inputs of the integrator, respectively.

In general the stray insensitivity condition for a two phase SC network can be stated as following [26,27]:

> The frequency response of a two phase SC filter is to a first order *parasitic-insensitive* if every capacitor terminal is EITHER connected to an input source, op-amp input, op-amp output, or the ground, OR switched between an op-amp input and the ground, an op-amp output and the ground, two op-amp outputs, an op-amp output and an input source, or an input source and the ground.

In the remaining of this chapter, only parasitic-insensitive SC structures will be discussed.

## 2.4 Design Methods

While a number of design methods such as SC N-path [28] and multiple-loop feedback [29,30] have been reported for SC filters, the majority of practical designs and all of the automatic synthesis tools are based on one of the following two approaches:

- Cascade first and second-order sections (biquads) to obtain the desired filtering function [31,32]. This is the most straight-forward method to apply.

- Design the SC filter based on the doubly-terminated LC ladder network [33,34].

The popularity of cascade design method has been mainly due to the following:

- Any transfer function of the form:

$$H_c(s) = \frac{N_c(s)}{D_c(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0},$$

  or

$$H_d(z) = \frac{N_d(z)}{D_d(z)} = \frac{c_m z^m + c_{m-1} z^{m-1} + \cdots + c_1 z + c_0}{z^n + d_{n-1} z^{n-1} + \cdots + d_1 z + d_0}$$

  can be realized as cascade of first and second order sections.

- A large number of easy-to-design-and-tune SC biquad structures have been reported [35,31,32,27]. The majority of these structures are based on the active RC realization of biquads, which have been thoroughly studied in the literature.

- Different biquad sections can be clocked at different rates. Thus, we can make very efficient use of the silicon area by using multiple clock rates [36].

The main drawback of the cascade design approach is that the resulting designs are inherently more sensitive to component variations than are the SC ladder topologies. A well designed SC ladder filter has minimum sensitivity to element variations in the pass-band. In fact, their analog prototypes have zero sensitivity when the power transfer is matched between the source and load. For this reason SC ladder structures are preferred over cascade designs for high order and high Q filters.

The most widely used strategy to derive parasitic-insensitive SC filters from the LC ladder prototypes is based on the simulation of the voltage-current relations of the analog filter. These filters can be designed on the basis of simple approximations which are valid for clock frequencies much larger than the fastest fluctuations of the input voltage. In case the deviation of the frequency response from ideal resulting from an approximate design is not acceptable, an exact but more complicated design strategy may be used [37,38,34,39,40]. An alternative approach is to fine-tune the capacitor ratios of the approximate design to meet the desired frequency response. This will be illustrated in the following section.

A systematic procedure for the design of cascaded SC filters has been proposed in [32]. In the following section a step-by-step procedure for the design of parasitic-insensitive SC ladder filters based on the simulation of the voltage-current relations of the analog filter will be presented.

## 2.5 A Synthesis Procedure for SC Ladder Filters

There is a large body of knowledge and experience with passive ladder networks. Thus, it is natural to design SC filters to emulate the behavior of these networks. A method which retains the low-sensitivity properties of ladder networks, is to simulate the voltage-current relations of the passive ladder networks via signal flow graphs. This simulation will result in

SC networks comprising damped and lossless integrator and/or second order resonator sections.

The first step in the synthesis of SC ladder filters (figure 2.4) is to realize the passive ladder network to meet the given frequency response specifications. In this approach, the filter synthesis program Filsyn [41] has been employed for this purpose. Figure 2.5 shows the general form of passive ladder networks transformable to SC ladder filters through the steps outlined in this section. Before mapping the passive ladder into interconnections of integrators, the L and C loops which might exist in the passive filter should be broken. Inductor loops can result in unstable D.C. conditions [24], while breaking the capacitor loops will result in fewer number of op-amps in the SC implementation of the filter [33]. These loops can be broken through the Thevenin equivalent circuit transformations shown in figure 2.6. The transformed network can be realized as a leapfrog interconnection of integrators (figure 2.7). The time constant associated with each integrator is proportional to a corresponding capacitor or inductor value of the transformed passive network. Namely:

$$\tau_i = RC_i \quad \text{or} \quad \tau_i = \frac{L_i}{R},$$

where $R$ is a free parameter which is later used to optimize the capacitor ratio spread of the resulting SC filter. The resistive terminations are simulated as feedbacks closing around the integrator whose one input is the input of the filter and the integrator whose output is the output of the filter. The gain constant associated with a resistive termination $R_t$ has a value of $\frac{R_t}{R}$ if the integrator corresponds to an inductor or $\frac{R}{R_t}$ if it corresponds to a capacitor. The voltage-controlled voltage sources are simulated as feed-forward and feedback paths between the integrator outputs. A step whose importance becomes clear shortly, is to negate the polarity of the coupling paths which feed into the outputs of the integrators. This can be easily accomplished by reversing the polarity of every signal coming into and going out of one of the two integrators incident to a coupling path (figure 2.8). This operation does

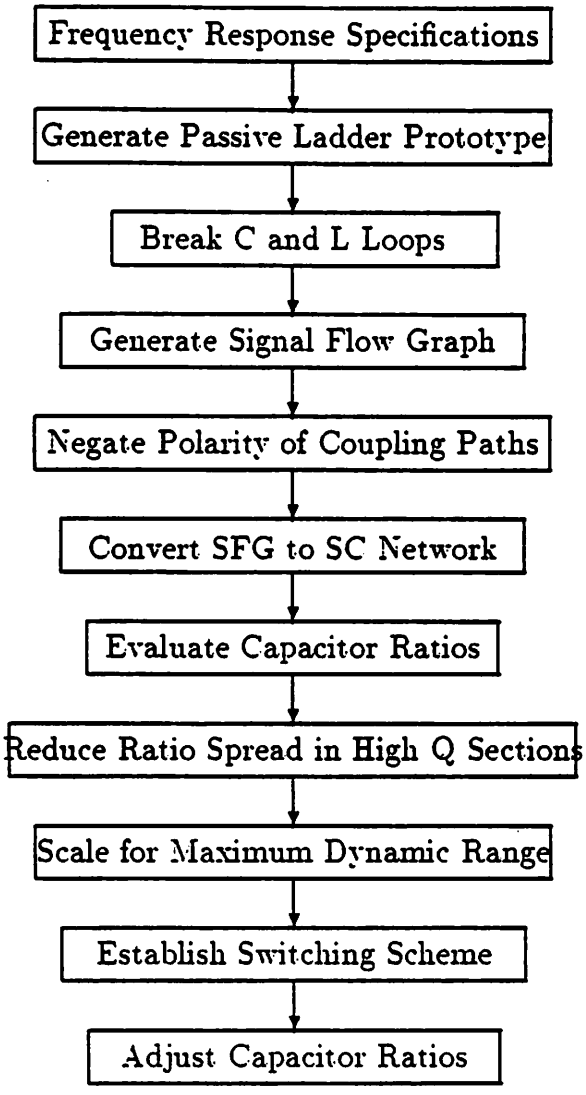



Figure 2.4: The synthesis steps for SC ladder filters.

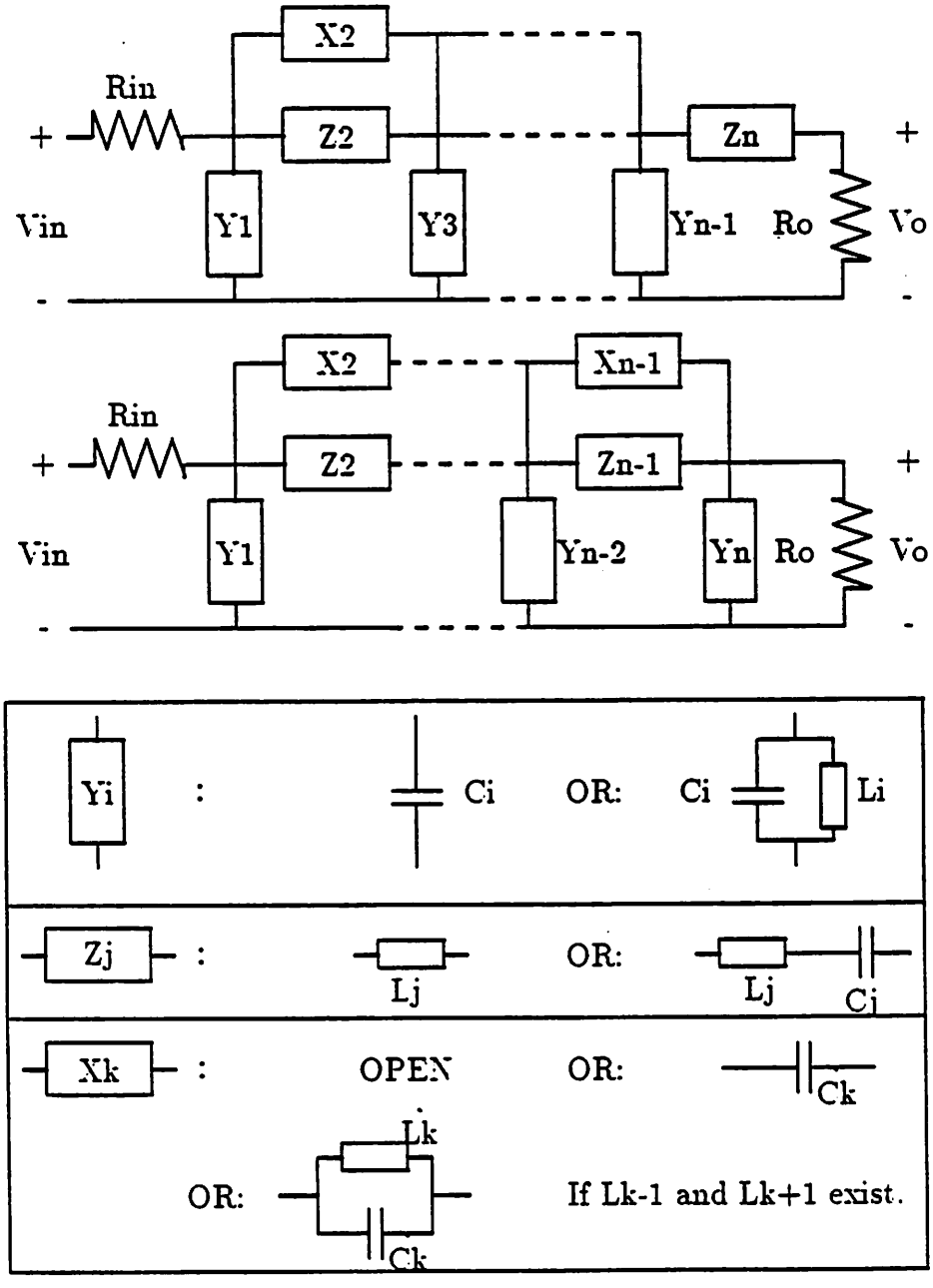
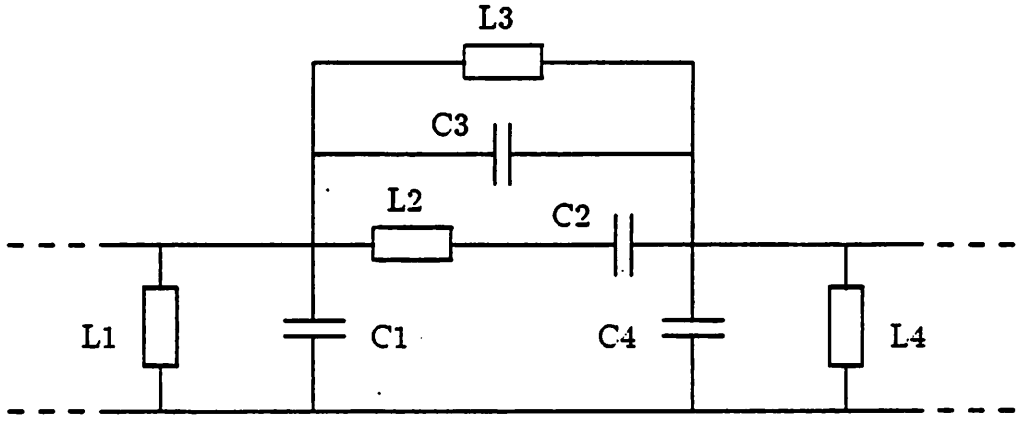
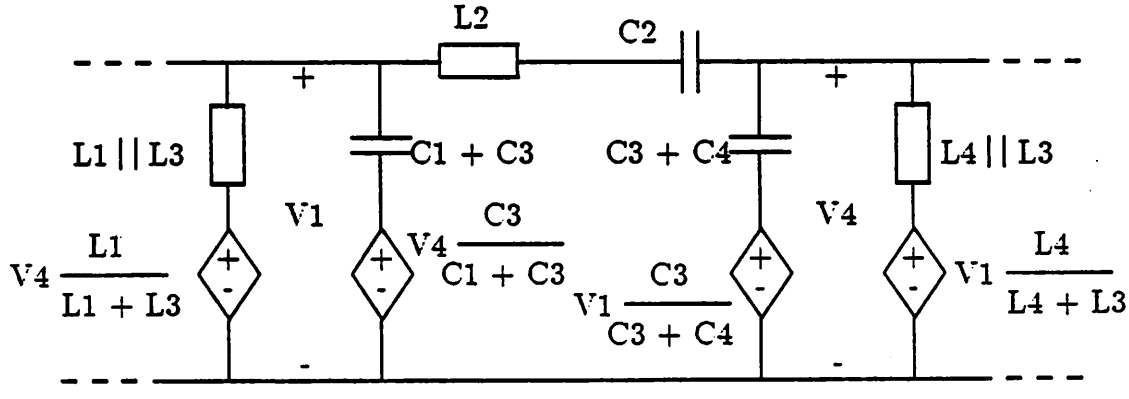



Figure 2.5: The general form of passive ladder networks.

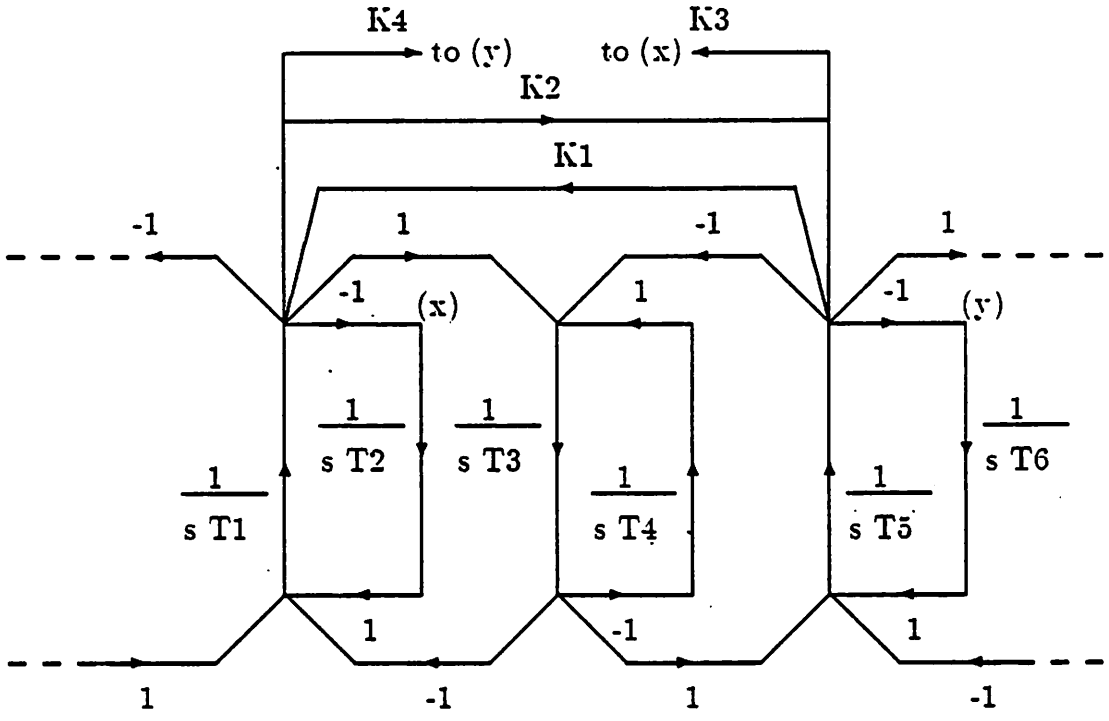



(a)




(b)

Figure 2.6: (a) A section of a passive ladder network containing capacitor and inductor loops. (b) Its Thevenin equivalent circuit.




$$T1 = R\,(C1 + C3) \qquad T2 = (L1 \,||\, L3) / R$$

$$T3 = L2 / R \qquad T4 = R\,C2$$

$$T5 = R\,(C3 + C4) \qquad T6 = (L4 \,||\, L3) / R$$

$$K1 = C3 / (C1 + C3) \qquad K2 = C3 / (C3 + C4)$$

$$K3 = L1 / (L1 + L3) \qquad K4 = L4 / (L4 + L3)$$

Figure 2.7: A signal flow graph representation of the passive ladder segment of figure 2.6.

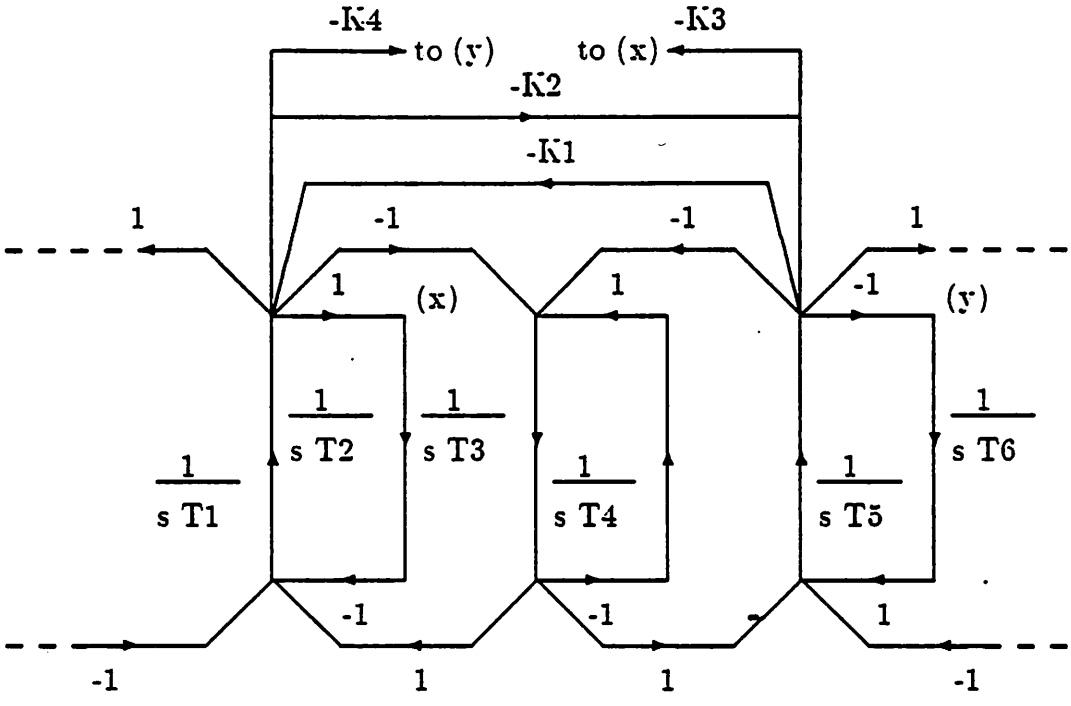



Figure 2.8: Changing the polarity of coupling paths.

not change the transfer function of the filter since all the loop gains remain the same.

At this point, the resulting Signal Flow Graph (SFG) can be used to generate an SC ladder filter. This is done by implementing each integrator as an LDI SC integrator/summer discussed previously. All the coupling paths which feed into the outputs of the integrators in the SFG have negative values. Thus, they can be realized via the summing capacitors. Those coupling paths feeding into the input of the integrators can be realized via the sampling capacitors. For a clocking period $T$, ratios of integrating to sampling capacitors become:

$$\frac{c_{I_i}}{c_{s_i}} = \alpha \frac{\tau_i}{T},$$

where $\alpha = 1$ if the above ratio doesn't correspond to a resistive feedback path. Otherwise, $\alpha = \frac{R_f}{R}$ or $\alpha = \frac{R}{R_f}$ if the integrator emulates an inductor or a capacitor, respectively. The ratio of a summing to an integrating ca-

pacitor is simply the absolute value of the gain constant associated with the corresponding path. There is a degree of freedom in choosing the switching phases. This will be discussed later.

If the filter contains high Q resonator sections, the capacitor ratio spread can be reduced by converting the switched-capacitor inputs of an integrator into continuous couplings at the input of the other integrator of the same resonator [24] (figure 2.9). This transformation is expected to improve the noise performance of the filter by reducing the number of switches in the circuit. One drawback is the settling behavior of the op-amps within the filter becomes more complex as a result of additional continuous coupling paths.

The generated SC filter can now be scaled for maximum dynamic range. The idea behind scaling comes from the following observation:

> If all the branches connected to the output of an op-amp in an SC filter are multiplied by some factor $k_i$, the output voltage of that op-amp is reduced by the same constant factor without affecting any other charges or voltages in the filter.

It has been shown that for maximum dynamic range, all op-amp outputs should be scaled such that each ( at its own maximum frequency ) saturates for the same input voltage level [42]. It should be noted that the logarithmic sensitivity of the filter to element variations remains unchanged by the above scaling operation. Scaling for maximum dynamic range can be performed in the following manner:

1. Set the amplitude of the input voltage sweep to the largest value for which the output op-amp does not saturate.

2. For each op-amp $i$. excluding the output op-amp, calculate $V_{i,s}$, the maximum value of the output voltage over all frequencies. This value usually occurs near a passband edge.

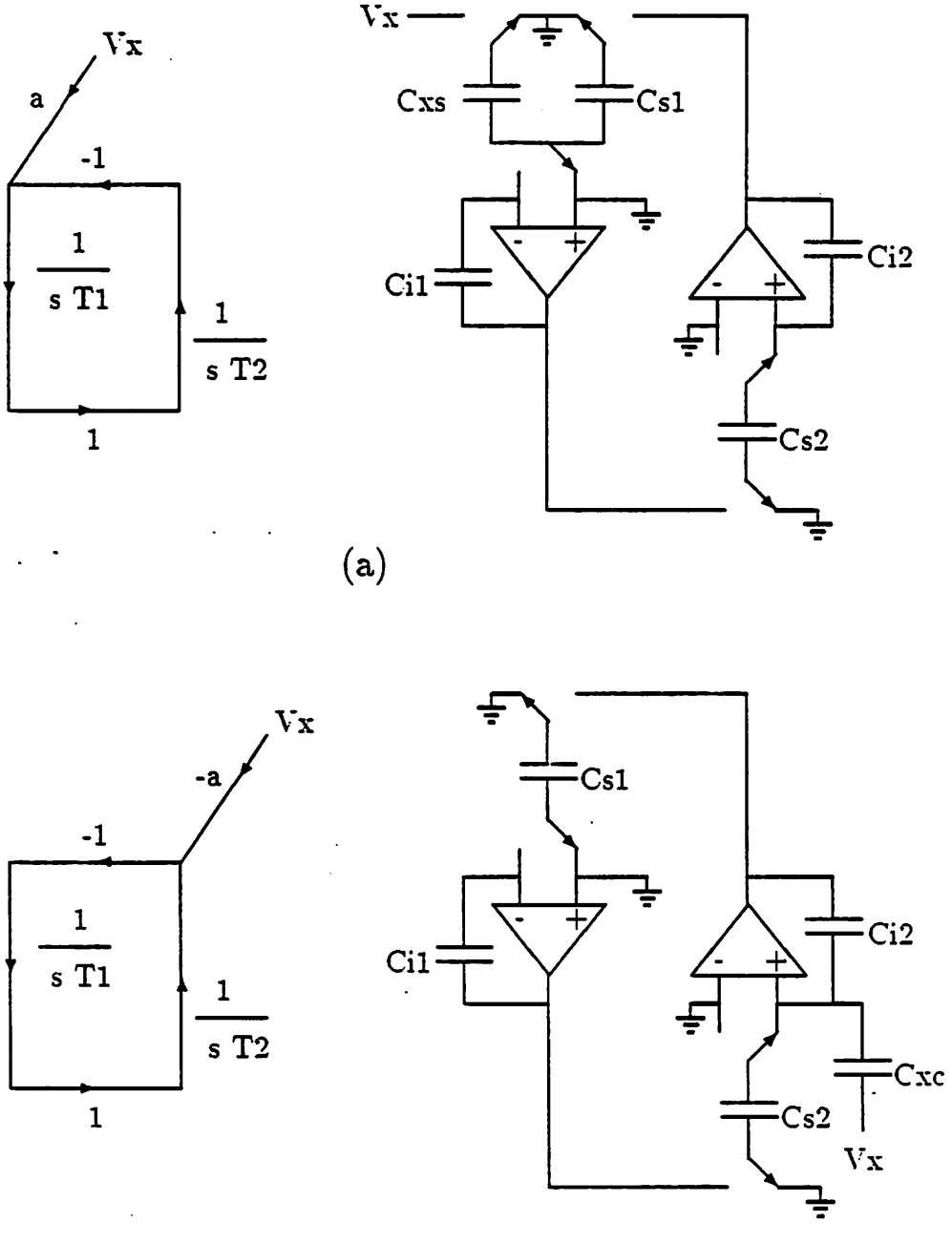



Figure 2.9: Circuit transformation for reducing capacitor ratio spread in high Q resonators. (a) Before transformation. (b) After transformation.

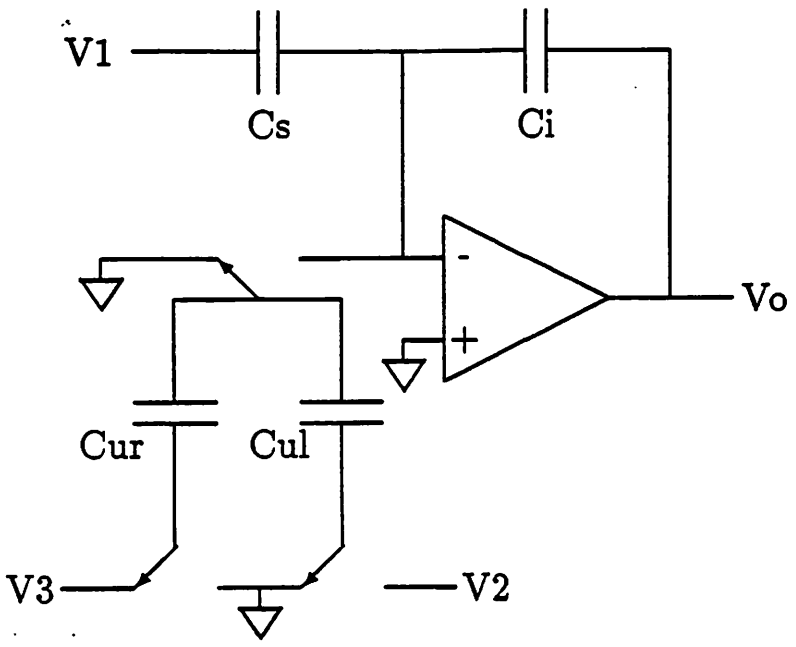



Figure 2.10: The SC integrator of figure 2.2 after scaling for maximum dynamic range.

3. For each op-amp $i$ as above, multiply all the capacitors connected or switched to its output by $k_i = \frac{V_{i,s}}{V_{i,m}}$, where $V_{i,m}$ is the saturation voltage of op-amp $i$.

As a result of scaling operation. some integrators might require additional switched capacitors to accommodate different gains for different input polarities (figure 2.10).

At the next step, the value of the impedance scale $R$ is evaluated to minimize the maximum capacitor ratio. Let us assume the set of capacitor ratios related to parameter R are given by:

$$\{ \{\frac{a_i}{R}\} , \{b_j R\} \} \quad i = 1, \cdots n \quad , j = 1, \cdots, m,$$

where $a_i$'s and $b_j$'s are constant parameters. Then, $R_{opt}$ the optimum value of $R$ is given by:

$$R_{opt} = \sqrt{\frac{\max_{j \epsilon m}(b_j)}{\max_{i \epsilon n}(a_i)}}.$$

As mentioned before, there is some degree of freedom in choosing the switching scheme. In figure 2.11a, both switched-capacitors $C_{ur}$ and $C_{ul}$ implement inverting inputs to the integrator. In figure 2.11b, the clock phases of the top and bottom switches of $C_{ur}$ have been swapped, so that a single switch can be used at the input of the integrator for both switched capacitors. This reduces the channel charge injection into the input of the integrator. This step does not affect the transfer function of the filter as long as there is no resistive feedback path around the integrator. For damped integrators, the clock phasing determines whether the feedback loop is delay free or it takes a full clock cycle to go around a loop (figure 2.12). Choi [24] has shown that using full delay termination in one end and zero delay at the other end of the SC filter (complex conjugate terminations) can result in better approximation of the desired frequency response. Then, according to the arguments above, the switching scheme should be established for minimum number of switches connected to the input of each amplifier, and if possible, for complex conjugate terminations.

As a result of the approximate nature of the design, the generated SC filter might not meet the frequency response specifications. In order to meet the specifications, the capacitor ratios can be adjusted with the help of mathematical programming techniques [23]. Figure 2.5 shows the effect of optimization on the pass-band response of a ninth order low-pass SC ladder filter designed according to the steps above. The clock frequency is 64 KHz. The specifications require 0.25 dB passband ripple with cut-off frequency at 6 KHz. The bandstop requirement are: -26 dB at 7.5 KHz, -45 dB at 9 KHz, and -80dB at 18 KHz. The approximate design easily met the bandstop specifications. However, the original passband ripple was 0.44 dB which is unacceptable.

The process above has been incorporated into a program for automatic synthesis of SC ladder filters. Other such efforts are described in the next section.

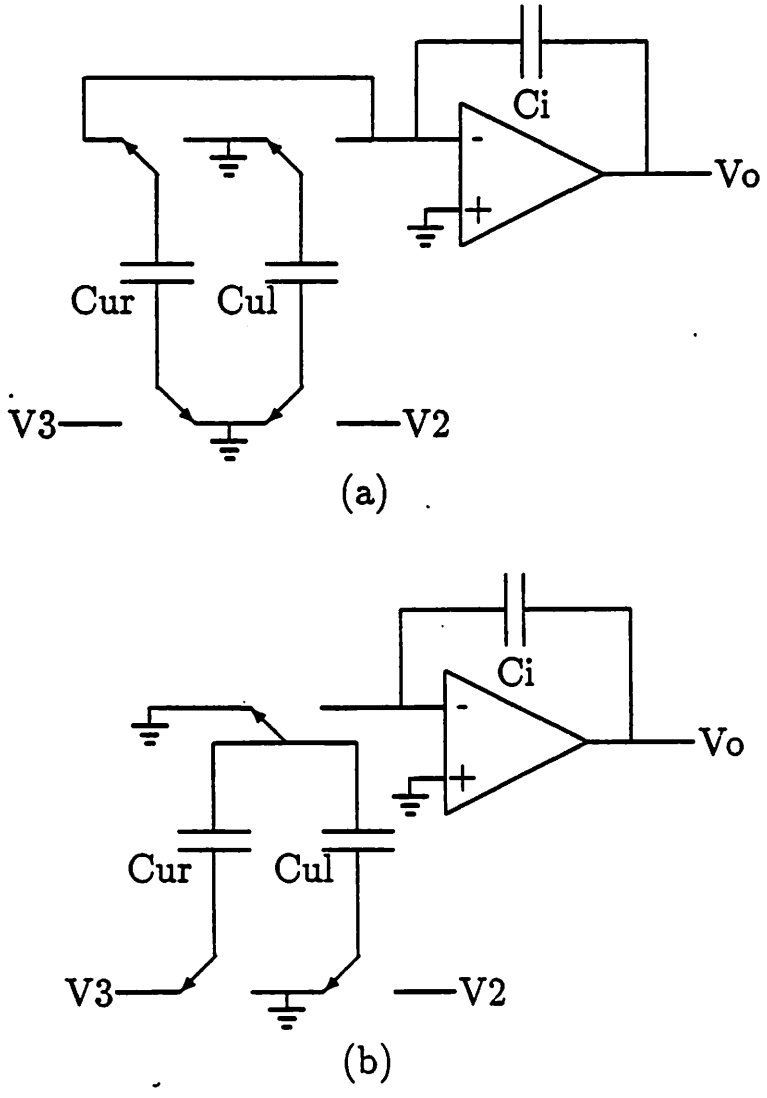



Figure 2.11: Reducing the number of input switches. The circuit of Figure (b) has one less switch than the circuit of Figure (b).

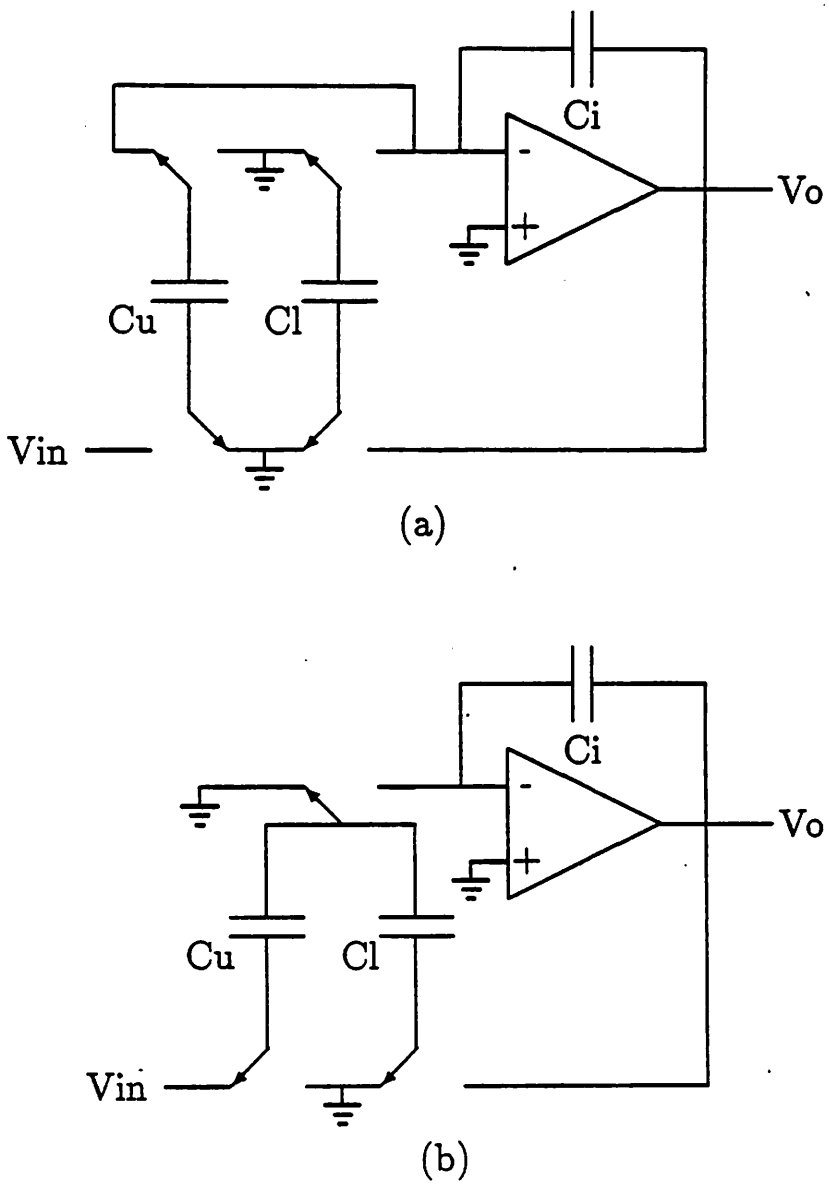



Figure 2.12: (a) Full delay termination. (b) Zero delay termination.

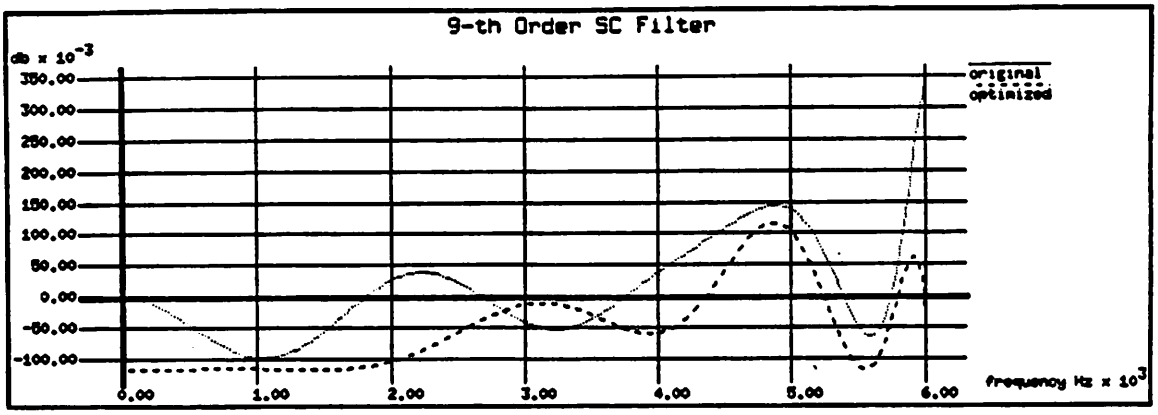



Figure 2.13: Pass-band response of the optimized SC ladder filter.

## 2.6 Review of Automatic Synthesis Systems

The system reported by Eaton [43] is capable of synthesizing parasitic insensitive low-pass SC ladder filters for classical designs (e.g. Butterworth, Chebychev, elliptic, etc.). The generated filters are scaled for maximum dynamic range. In this system there is a choice of either an approximate LDI or an exact [39] design methodology.

The system reported by Assael [44], is capable of synthesizing low-sensitivity low-pass and bandpass SC ladder filters for standard polynomial approximations and also for elliptic filters. The capacitor values are evaluated for maximum dynamic range. An approximate LDI or an exact bilinear designs are possible with this system.

Sanchez-Sinecio [45] reported a program for synthesis of biquad SC filters. The program accepts either a frequency specification, an s-domain transfer function, or a z-domain transfer function for the filter. It permits trade-offs between sensitivity, dynamic range, and total capacitance of the filter.

Another program for synthesis of biquad SC filters has been re-

ported in [46]. The program accepts an amplitude frequency specification for the filter. Then, it evaluates the poles and zeros of the classical type transfer function to meet the desired frequency response. The dynamic range of the filter is optimized by pairing the poles and zeros of the transfer function and ordering of the resulting biquad stages. Finally, the SC filter topology and capacitor ratios are evaluated for each biquad section.

While there is some optimization performed in the reported synthesis systems, there are many more degrees of freedom yet to be explored. An example for this is the optimal choice of the integrating capacitor values. These values affect, among many, the area, noise, and power supply rejection of the filter. An optimal decision requires accurate measures of the performance criteria. This is possible only after a layout is generated for the filter and the parasitics are extracted from it. Then, an optimization step can be used to produce more efficient circuits.

# Chapter 3

# Layout Strategies for Switched-Capacitor Filters

## 3.1 Introduction

The performance of analog ICs depends heavily on fine details of layout. This has been part of the reason why manual layout has been the most time consuming aspect of the design cycle for most analog ICs and in particular SC filters. The development of a number of automatic layout systems for these filters is thus motivated. As it will be discussed later in this chapter, an important shortcoming of earlier systems is that they do not make as efficient use of the area as does manual layout. Area directly translates into cost, and also larger area prevents integration of more functions on the same chip. In order to approach the area efficiency of manual layouts, a strategy with three basic characteristics has been adopted. The heart of this strategy is the use of a layout floor-plan patterned after a style widely used in manual layout of these filters. The second aspect of this strategy is to take advantage of the flexibility inherent in the design of ratio-accurate capacitors to obtain more area efficient layouts. The final aspect is to route all the signals including the sensitive signals automatically to obtain more area-efficiency and extend the capabilities of the system to a larger class of

circuits.

This chapter starts with a review of technological considerations for the layout of SC filters. An important decision is the choice between a single-ended or a fully differential architecture. These architectures are introduced and compared in the last part of this section. Previous work in the area of automatic layout of SC filters will be discussed in Section 3.3. The final section describes an automatic layout system based on the approach discussed above. This section starts with introducing the floor-planning strategy. Then the choice of input format is discussed. Finally, the organization of the layout system is presented.

## 3.2 Considerations for the Layout of Switched-Capacitor Filters

### 3.2.1 Capacitor Ratio Accuracy

To a first order the value of a monolithic MOS capacitor with a rectangular shape of dimensions $W$ and $L$ is given by:

$$C = \frac{\epsilon_0 \epsilon_{ox} W L}{t_{ox}}. \tag{3.1}$$

If the variables are statistically independent, then the normalized standard deviation in the capacitor value is

$$\frac{\sigma_C}{C} = \left[ \left(\frac{\sigma_{\epsilon_{ox}}}{\epsilon_{ox}}\right)^2 + \left(\frac{\sigma_{t_{ox}}}{t_{ox}}\right)^2 + \left(\frac{\sigma_W}{W}\right)^2 + \left(\frac{\sigma_L}{L}\right)^2 \right]^{\frac{1}{2}}. \tag{3.2}$$

The errors associated with the first two terms of the equation above are referred to as *oxide* effects, and the errors associated with the last two terms are called the *edge* effects. For large capacitors the oxide effects are dominant while for small capacitors, the edge effects are dominant. To avoid systematic errors resulting from process variations and mask moverments, MOS capacitors are often realized as parallel connections of several smaller identical units commonly referred to as "unit-capacitors" [25]. Then, the the ratio

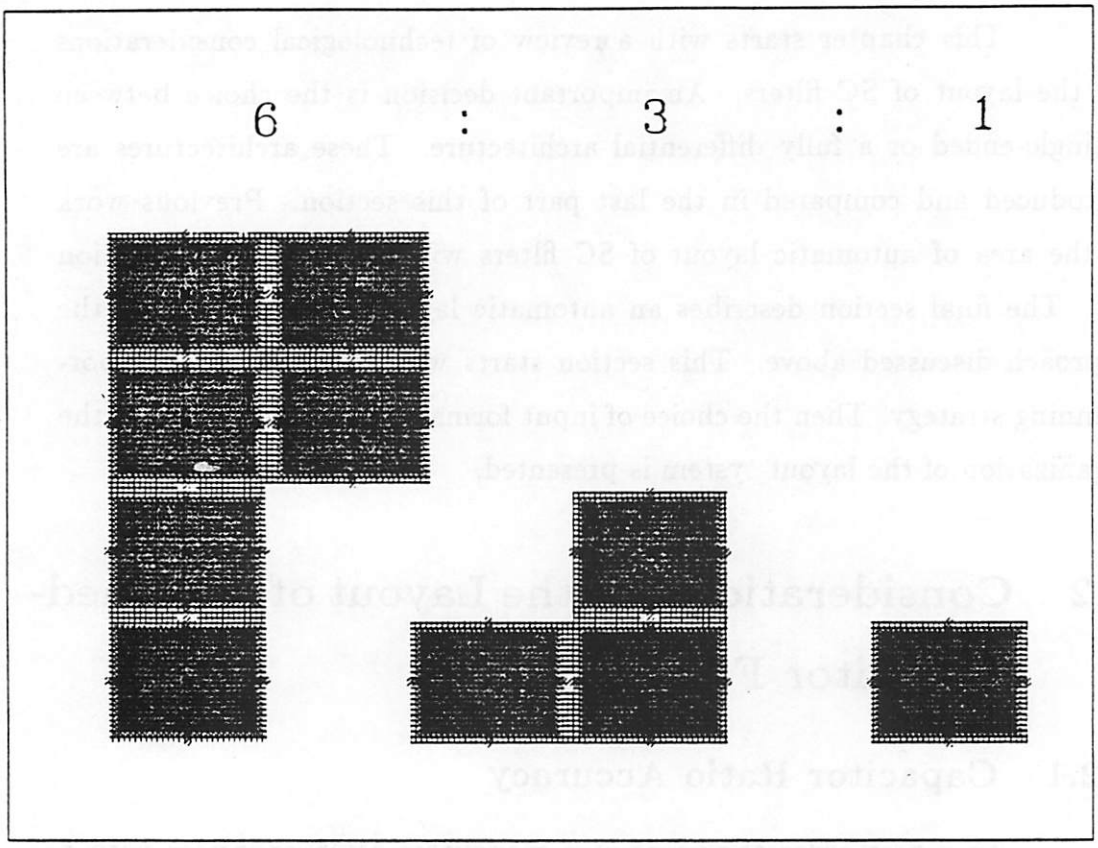



Figure 3.1: Accurate capacitor ratioing.

of the area of the capacitor to the total perimeter of the plate defining the capacitor is constant for each capacitor in the set of ratioed capacitors. This eliminates the effects of systematic errors on the ratio of nearby MOS capacitors having identical unit-capacitors. Then the only remaining sources of error are the random process variations. With this strategy, capacitor ratio-accuracies of the order of 0.1 percent are achievable, even though the absolute values of monolithic capacitors typically vary by ±20 percent.

Let us derive a relationship between ratio accuracy and the shape of a unit capacitor. With a procedure similar to the one given in [47] it can be shown that for a $W \times L$ $\mu m^2$ MOS capacitor $C$, with all the systematic errors reduced to zero, the normalized standard deviation from nominal capacitor

value can be given by:

$$\frac{\sigma_C}{C} = \sqrt{K_e \frac{L + W}{L^2 W^2} + K_o \frac{(L + W)^2}{L^2 W^2}} .$$

where process-dependent, geometry-independent parameters $K_e$ and $K_o$ are edge-effect and oxide-effect factors respectively. Now, consider an array of two capacitors $C_m$ and $C_n$, made of parallel connections of $m$ and $n$ identical unit capacitors $C$, respectively. Let us denote $r = \frac{C_m}{C_n} \geq 1$ to be the ratio of the two capacitors. Since we have assumed that systematic errors have been reduced to zero, $C_m$ and $C_n$ are statistically independent. Then, for small deviations from nominal values we can write:

$$\sigma_r = \frac{m}{n} \sqrt{\left(\frac{\sigma_{C_m}}{C_m}\right)^2 + \left(\frac{\sigma_{C_n}}{C_n}\right)^2}$$

Recognizing that $C_m = mC$, $\sigma^2_{C_m} = m\sigma^2_C$, and similarly for $C_n$, we obtain:

$$\sigma_r = \frac{m}{n} \frac{\sigma_C}{C} \sqrt{\frac{1}{m} + \frac{1}{n}} .$$

For a given $C$, the minimum $\sigma_r$ is achieved when $L = W = \sqrt{\frac{t_{ox} C}{\epsilon_0 \epsilon_{ox}}}$ (square geometry). If typical values of $K_o$ and $K_e$ are known for a given process, then the size of square unit capacitor to result in a $\sigma_r$ for a given ratio $\frac{m}{n}$ can be evaluated. In some cases, the unit capacitor size required to achieve the noise and PSR requirements of the filter is larger than the value determined above. Then, there would be a degree of freedom in choosing the shape of the unit-capacitor to guarantee the same capacitor ratio accuracy. The minimum side dimension of such rectangular unit-capacitors can be evaluated by substituting $W = \frac{t_{ox} C}{\epsilon_0 \epsilon_{ox} L}$ in the equations above and solving for the value of $L$ which results in a given $\sigma_r$. The layout system to be described, takes advantage of flexible unit-capacitor shapes to obtain more area-efficient layouts.

### 3.2.2 Techniques for Improving Power Supply Rejection

As we have seen in the previous chapter, the parasitic capacitance $C_{p7}$ (figure 2.3) is of critical importance in determining the PSR of an SC filter. To a first order, the ratio of this capacitor to the integrating capacitor determines the low frequency PSR. To minimize the contribution of the capacitors to $C_{p7}$, only the top-plates of the capacitors should be connected to the summing nodes of the op-amps. This could also reduce leakage current effects. Another procedure is to place a grounded well under the capacitors to shield the capacitor plates from substrate noise. In order to minimize the contribution of the switches to $C_{p7}$, a number of measures can be taken:

- If possible, the interconnect lines which are connected to the summing nodes of the amplifiers should be shielded.
- In CMOS technologies, one should use single-channel switches for the top-plates, and full transmission gates for the bottom plates.
- For each integrator, use only one set of top-plate switches. These switches should be as small as possible.

In some applications such as high Q bandpass filters, the PSR of the filter might be poor even with the above provisions. For them, a fully-differential architecture is desirable. This will be discussed in the next section.

### 3.2.3 Fully-Differential Versus Single-Ended Architectures

Every SC filter can be implemented in either a single-ended or a fully differential architecture [48,24]. In the latter, the effective signal swing is doubled relative to the former, by the use of a differential output integrator (figure 3.2). In fully-differential architectures, the signals injected due to power supply variations and clock charge injections are greatly reduced due

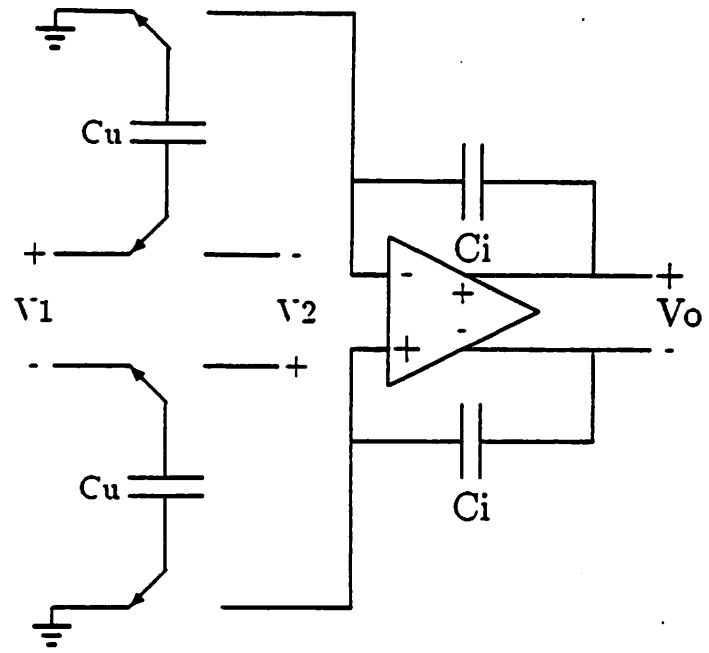



Figure 3.2: A fully-differential SC integrator.

to balanced signal paths. The disadvantages of fully-differential technique are filter areas are larger than single-ended versions. and differential to single ended conversion is necessary in most applications.

In this section. the requirements for the layout of both architecture are examined. I start by listing the requirements common to both architectures:

- In order for two capacitors to track. their unit-capacitors should be of the same size and shape.
- All the capacitors belonging to the same integrator have to be accurately ratioed with the integrating capacitor of that integrator.
- It is desirable to place the tracking capacitors close to each other to reduce the effects of uniform thickness gradient on matching of capacitors.
- The lower plate of each capacitor must be connected to a voltage source

or switched between voltage sources, so that the nonlinear parasitic capacitance between the lower plate and the substrate does not effect the filter response. Therefore, only the top-plate of each capacitor can be connected or switched to the summing inputs of an op-amp.

Requirements particular to a single-ended architecture are:

- The non-inverting op-amp inputs should be kept at a constant voltage, and cannot be connected to a signal voltage: otherwise, the filter response will be sensitive to the parasitic capacitances associated with the inverting input.

- The parasitics associated with the wiring to summing nodes should be reduced, or if possible, eliminated. These parasitics distort the frequency response, and degrade the power supply rejection and noise of the filter.

Requirements particular to fully-differential architecture are:

- The parasitic capacitances due to switches, bus lines, and substrate connected to the inverting input of each op-amp should match those connected to the non-inverting input of the same op-amp.

- Each pair of differential capacitors should have identical layouts, and should be placed close to each other.

- In a truly differential architecture, the layout should be insensitive to mask movements; namely, a pair of differential elements, should have identical layouts in lieu of mask movements.

We conclude that the constraints on signal routing are very different for the two architectures. It will be shown later that it is possible to obtain area-efficient layouts with good performance for single-ended filters by using standard routing tools; however, this is not the case for fully-differential

architectures. For these architectures. new routing tools are needed to maximize the symmetry of the differential signals while minimizing the routing area. To take advantage of existing routing tools. we placed our initial effort on generating layouts for single-ended architectures. the subject of the remaining of this thesis.

## 3.3 Systems for Automatic Layout of Switched-Capacitor Filters

A number of automatic layout systems for switched-capacitor filters have been described previously.

Allen [6] reported the first system for automatic layout of SC filter. This system uses a standard-cell layout scheme for placing and routing fixed height bi-quad sections to generate the filter layout (figure 3.3). Each parameterized bi-quad section is made up of two op-amp sections. cascaded with a number of capacitor tile sections. In this approach. the summing node connections (sensitive signals) are hidden inside the bi-quad sections to ease the task of placement and routing. An important shortcoming of this and other similar parameterized bi-quad/integrator layout approaches is that they do not make as efficient use of the silicon area as does manual layout. This is partly due to the semi-rigid structure of the bi-quad sections which leaves only few parameters adjustable for area optimization. and also due to the large area used up by signal bussing. Another characteristics is the use of a large number of leaf cells which must be regenerated for each new technology.

In the analog standard cell system reported by Kimble [8] the signals are routed in alternating sensitive and insensitive channels (Figure 3.4). In this system the channel router routes certain classes of nets before others to achieve higher performance.

In the system reported by Pletersek [49], the signals are routed in

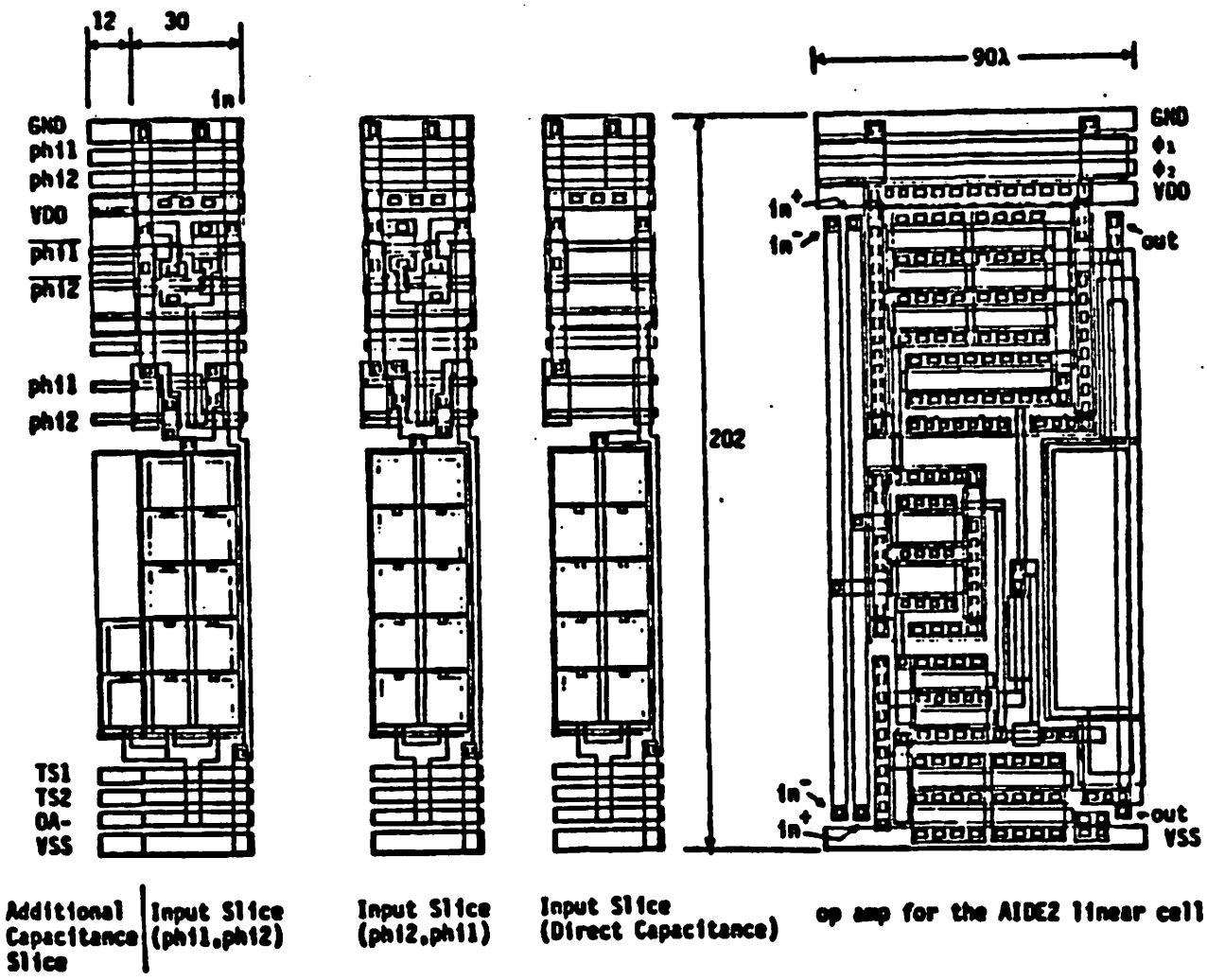



Figure 3.3: Parameterized bi-quadratic structure reported in [6]

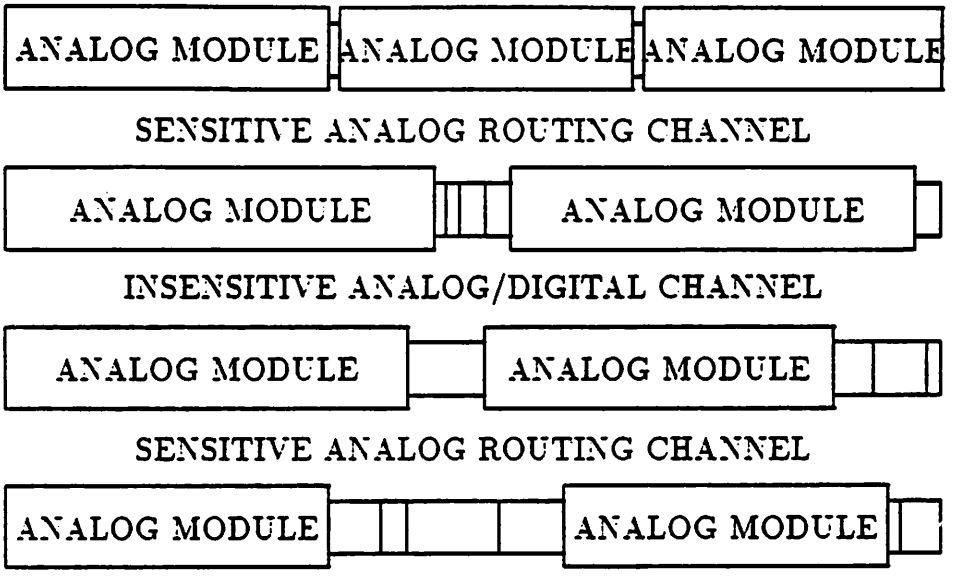



Figure 3.4: Standard-cell layout scheme reported in [8]

separate analog and digital channels to reduce the cross-talk between analog and digital signals (figure 3.5). The sensitive signals are hidden inside pre-designed bi-quad sections. These sections are blocks of fixed height with variable widths depending on the total number of unit-capacitors associated with that block. This is another example of parameterized bi-quad approach.

In the gate-array style strategy reported by Sigg [50], the capacitor top-plate as well as the interconnection masks are automatically generated for an otherwise pre-designed array of integrators (figure 3.6). While the fabrication turn-around time for generated filters is short, the gate-array nature of the system results in very large areas as compared to manual layouts.

In summary the above systems have one or more of the following limitations:

- The area of a generated filter is much larger than that of a compact custom layout of the same filter in the same technology.

- The system is designed around one technology or, if it is technology

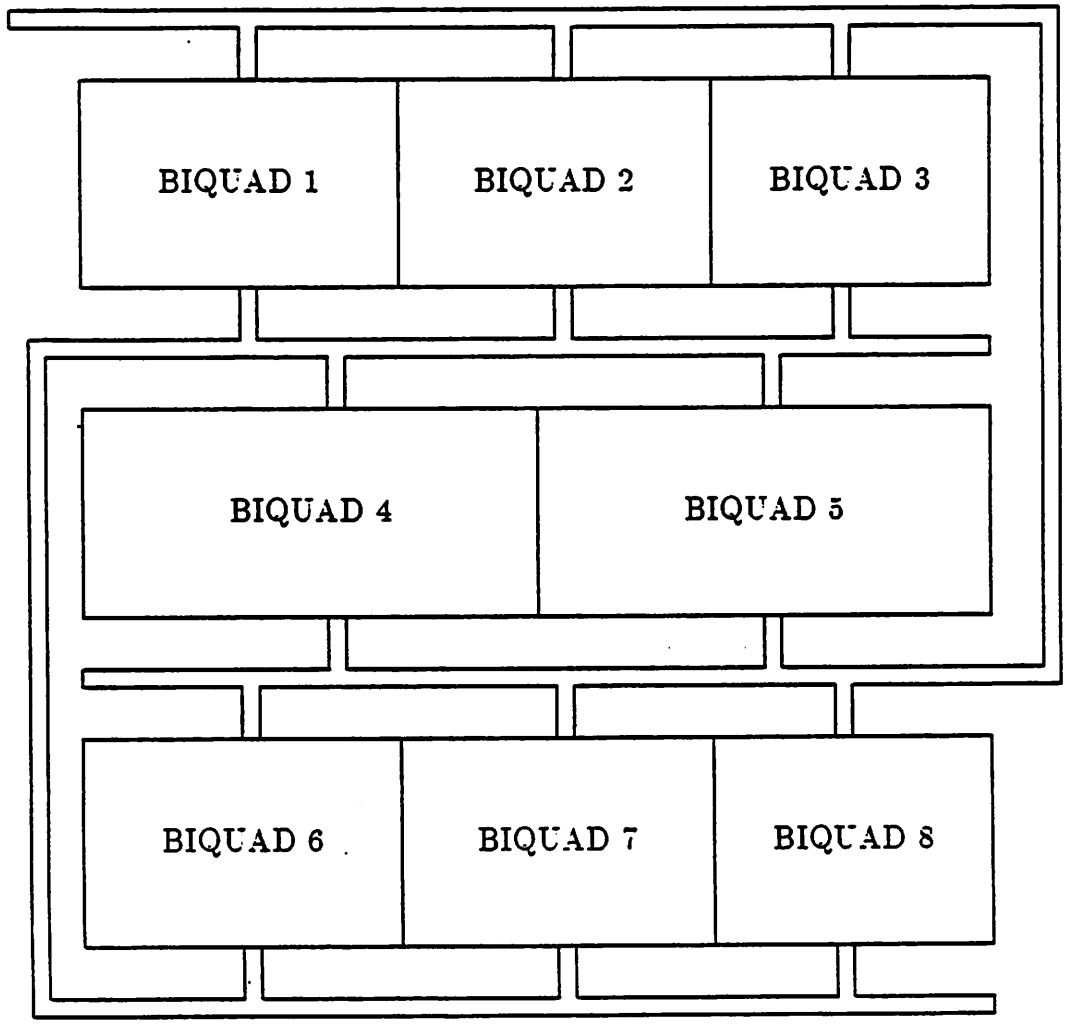



Figure 3.5: Standard-cell layout scheme reported in [49]

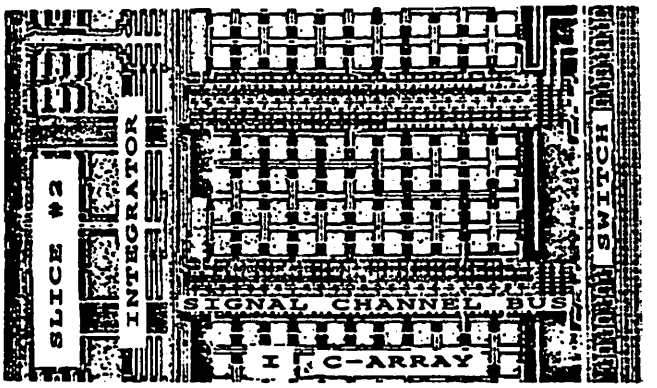



Figure 3.6: Gate-array style layout scheme reported in [50]

independent adopting the system to a new technology requires a large effort.

- The system works only with a library of elements specifically designed for it.

In the next section, our layout strategy for a technology-and-topology-independent system capable of producing area efficient layouts will be presented.

## 3.4 A New Approach to Automatic Layout of Switched-Capacitor Filters

### 3.4.1 Floor-planning strategy

The basic floor-plan consists of an array of capacitors located in between an array of operational amplifiers and an array of switches (Figure 3.7). This arrangement of operational amplifiers and switches is guided by

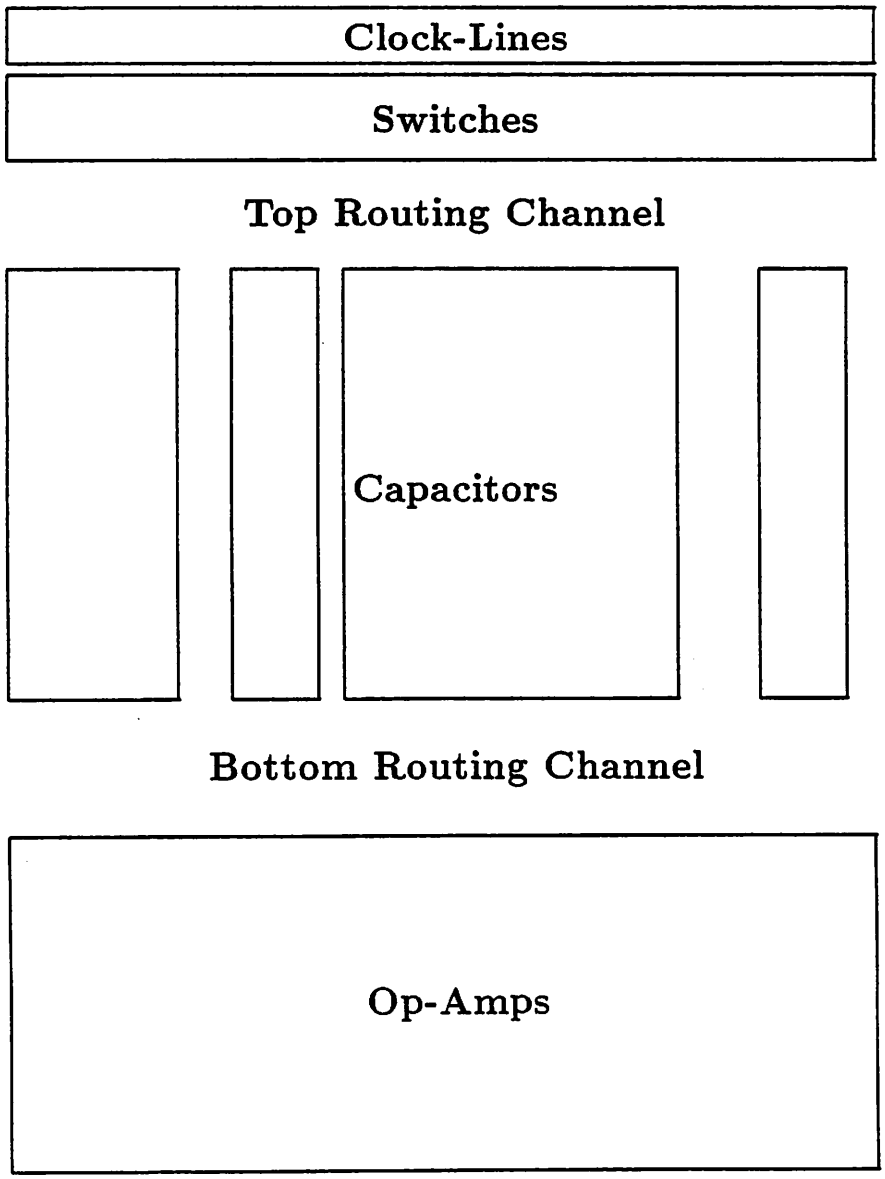



Figure 3.7: The Floor-Planning Strategy.

the bus sharing of the modules in their respective arrays. Each set of capacitors belonging to the same integrator makes up one capacitor block. The capacitors in the same block have identical unit capacitors; however, unit capacitors of different capacitor blocks need not be identical. The size of the unit-capacitor for each block is that of the smallest capacitor in that block. While the unit-capacitor area is constant, its aspect ratio can be manipulated to obtain more compact layouts.

The signals are routed in two channels, one between the capacitor blocks and operational amplifiers (op-amp channel), and another between the switches and capacitor blocks (switch channel). The spacing between capacitor blocks are used as route-through channels for connections between the op-amp and switch channels. Both terminals of each capacitor directly connected to the input of an op-amp are incident to the op-amp channel. Similarly, both terminals of each capacitor switched to the the input of an op-amp are incident to the switch channel. The clock lines run above the switches away from switch channel.

The op-amps can be chosen from any standard cell library, as long as all the op-amps used in the same layout have the same height, share the same busses, and have input and output terminals on the same side. These conditions are not restrictive since a large number of industrial op-amps used in SC applications are designed in this manner. The switches also have the same height, but possibly variable widths. Each switch should have on one side terminals connecting to the clock lines, and on the other side terminals connecting to op-amps or capacitors. Generation of compact layouts, entirely from the individual transistor dimensions and spacing rules, for these switches is possible because of their simple structures. An example of a scalable double-throw CMOS switch is shown in figure 3.8.

This floor-planning strategy serves many purposes:

- By using a channel router, and capacitor and switch generators, the entire layout of a filter, except for the op-amps, can be generated from

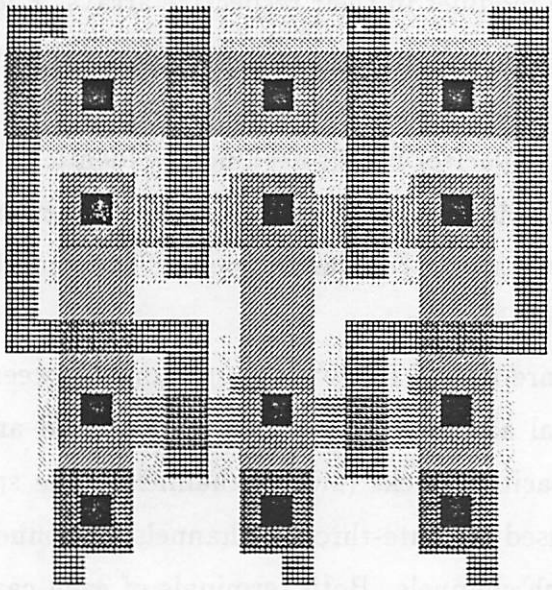


Figure 3.8: An example of a scalable switch.

connectivity information, spacing rules, and capacitor and switch sizes.

- Any circuit comprising switches, capacitors and op-amps, with any number of clock phasings can be implemented in this floor-planning style.
- The routing area, and the quality of routing can be optimized by ordering modules in their respective arrays.
- The degrees of freedom in the layout of capacitor blocks can be utilized to obtain more compact layouts.
- By proper design of the capacitor blocks, the capacitor top-plates which belong to the same net can be shared inside the capacitor block. This reduces the number of sensitive signals in each channel.
- The clock lines are separated from analog signal lines to reduce the coupling of digital and analog signals.

- Since the capacitors and most of the routing areas are separated from the switches and op-amps. a grounded well can be placed under the capacitors and large area of the routing channels to isolate the capacitor plates and most of the signal lines from substrate noise.

To utilize effectively this floor-planning strategy, we had to solve many problems, including the generation of capacitor blocks to meet the floor-planning strategy, and ordering the modules to reduce area and improve performance. These are the topics of Chapters 4 and 5.

### 3.4.2 The Input Format

The system requires the following information for layout compilation:

- A SPICE-like input file for describing the circuit connectivity, capacitor values, switch sizes and library pointers.

- A file for specifying various technology dependent information such as spacing rules.

- The bounding box and terminal locations for each library element specified in the circuit file.

The format of the circuit file, which resembles that of SPICE [51], was chosen for simplicity, ease of use, and generality. Each element in the circuit is specified by an element card that contains the element name, the circuit nodes to which the element is connected, and parameter value(s) or library pointer for the circuit element. Nodes are name or number fields, and the datum (ground) node must be numbered zero (0). The first letter of the element specifies the element type. There are five types of elements allowed in the circuit file: capacitors, op-amps, switches, input elements, and output elements. The input and output elements specify respectively the input and

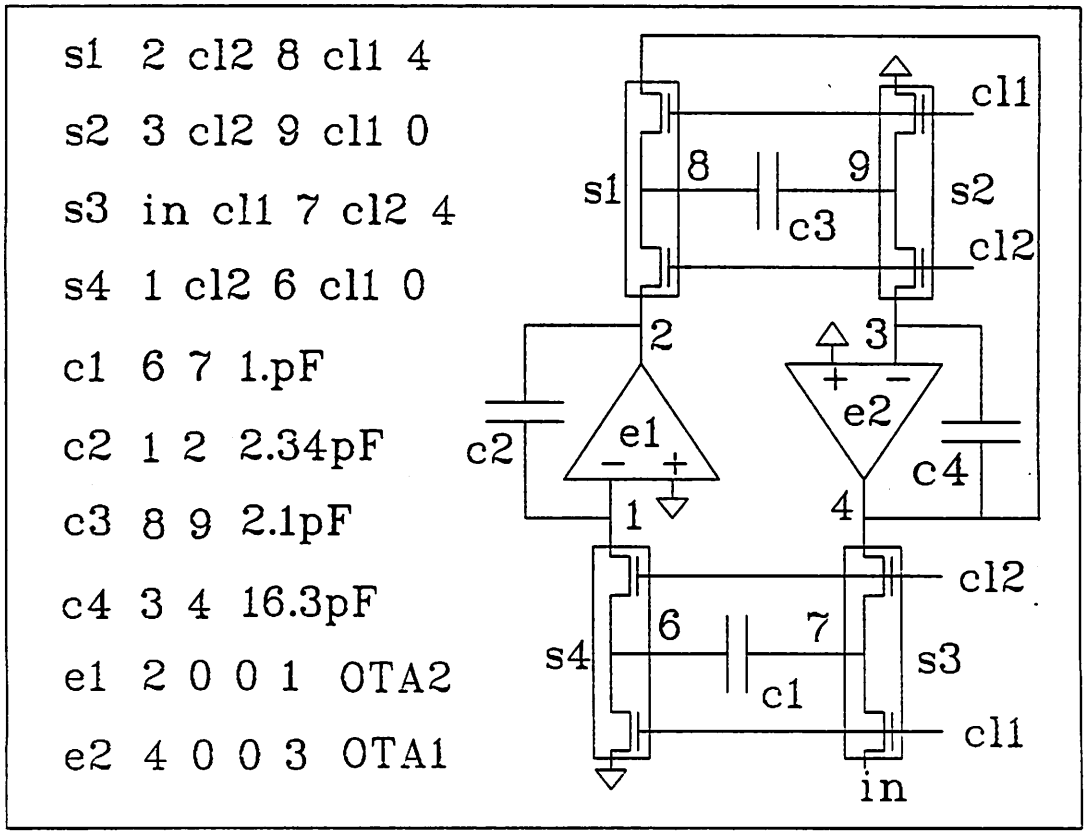



Figure 3.9: An example of a circuit file.

output nets of the filter. An example of a circuit file for a second order filter is shown in figure 3.9.

Since the system is designed to be technology independent, it requires new information for every new technology. A set of generic layers and design rule parameters have been defined to allow for various possibilities in generating layouts for the capacitors, switches, and generating the wiring. The mapping of technology parameters to this set is possible through the technology file.

### 3.4.3 Flow of the System

The system goes through several optimization stages, as shown in figure 3.10, before producing the final mask layout. The program starts with processing the input information. In the circuit file the capacitor nodes are not yet committed to be the top-plate or the bottom-plate. This is because such information is automatically extracted from circuit connectivity. The capacitor top-plate is designated to be the node directly connected or switched to an input of an op-amp. The bottom-plate is the node directly connected or switched to an output of an op-amp or a voltage source. All the capacitors with top-plates associated with the same op-amp input are assigned to the same integrating block. For each capacitor in an integrating block the ratio of its value to that of the smallest capacitor in the block is evaluated. Then, the size of the unit-capacitor for each integrating block is that of the smallest capacitor in the block. If no library names are specified for the switches, the layout for each unspecified switch is generated from technology information and the switch sizes. N-channel double throw switches are generated for those connecting the top-plates of the switched capacitors to the summing nodes of the op-amps or to the ground. Complementary double throw switches are generated for all other ones.

The capacitor arrays are generated in three stages. In the first stage the aspect ratio of the unit capacitors and the number of unit capacitors

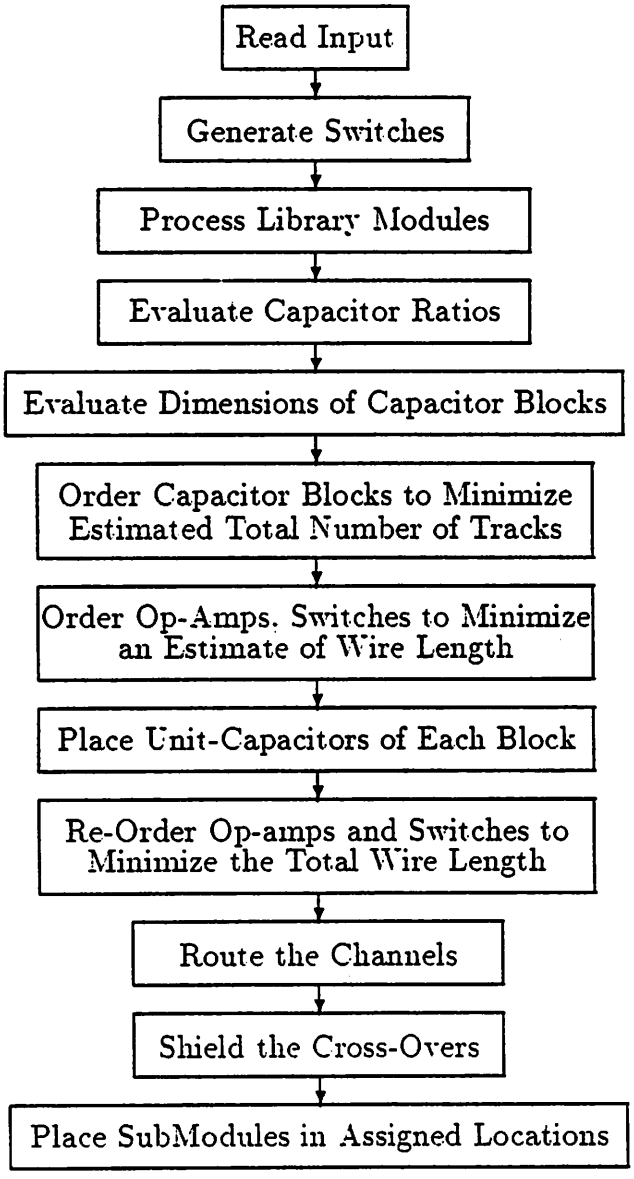



Figure 3.10: The organization of the layout system.

in the $x$ and $y$ directions for each capacitor block are evaluated in such a way to minimize an estimate of the total area of the filter. In the second stage. a symbolic representation for each capacitor block showing the relative positions of unit capacitors belonging to that block is produced. In the third stage a layout for each capacitor block is generated. There are several ordering steps to minimize the maximum density of each channel, as well as to minimize an estimate of the overall wiring length. An optional post-routing step is available to shield the cross-overs between summing nodes of the operational amplifiers and the large signal nets. The generated layout is stored in Oct [5]. In the next two chapters the algorithmic details of this system will be presented.

# Chapter 4

# Generation of Capacitor Arrays

## 4.1 Introduction

Design of compact high-accuracy array of capacitors is an error prone and laborious process even for moderate size SC filters. To meet the requirements of the floor planning, we developed algorithms to generate compact arrays of fixed area/perimeter ratio capacitors.

Generation of capacitor arrays is performed in three stages. In the first stage, the dimensions of each capacitor block and the shape of its unit capacitors are evaluated to minimize the overall area of the filter. In the second stage, for each capacitor block, a symbolic matrix showing the relative positions of unit capacitors of each capacitor in the block is generated. In the third stage, a spacer program converts the symbolic matrix generated for each block into a layout for the given design rules and evaluated unit capacitor dimensions. This method is general enough to be used in almost any other floor-planning style for SC filters or other applications requiring highly ratio-accurate capacitor blocks with flexible aspect ratios.

In Section 4.2, the algorithms for evaluating the optimum capacitor block dimensions are presented and then analyzed. Section 4.3 deals with the algorithms developed for symbolic matrix generation. In Section 4.4, the steps involved in converting the symbolic information into mask layout are

studied. A solution to generating area/perimeter constant geometries for fractional capacitors is presented in the final section.

## 4.2 Evaluation of Block Dimensions

### 4.2.1 Problem Formulation

A capacitor block $i$ can be perceived as an $n_{y_i} \times n_{x_i}$ matrix of interconnected unit-capacitors. For each capacitor block $i$, $n_{x_i}$ and $n_{y_i}$ should be large enough for the capacitor block to accommodate all of the unit-capacitors of all its capacitors. Since in our layout strategy, each capacitor is required to have at least one unit-capacitor either on top or at the bottom of its capacitor block, the minimum number of columns required for capacitor block $i$ is $\max(N_{tc_i}, N_{bc_i})$. where $N_{tc_i}$ and $N_{bc_i}$ are the number of capacitors with connections on top and the number of capacitors with connections at the bottom of block $i$, respectively. As long as the above restrictions are satisfied, one is free to choose the number of rows and columns of each capacitor block. Another degree of freedom is the ability to choose the aspect ratio of the unit-capacitor for each capacitor block. As it has been discussed in Section 3.2, the aspect ratio of a fixed-area rectangular unit-capacitor affects only the capacitor ratio-accuracy. The aspect ratio can therefore take on a range of values, as long as the given capacitor ratio accuracy is maintained. Our goal is to minimize the area of the filter defined by its bounding box (Figure 4.1) by taking advantage of the degrees of freedom described above. To formulate the problem, the following definitions are required:

$N_b$ = number of capacitor blocks.

$w_{uc_i}$ = unit capacitor width for block $i$.

$l_{uc_i}$ = unit capacitor length for block $i$.

$n_{x_i}$ = number of unit capacitors in the $x$ direction for block $i$.

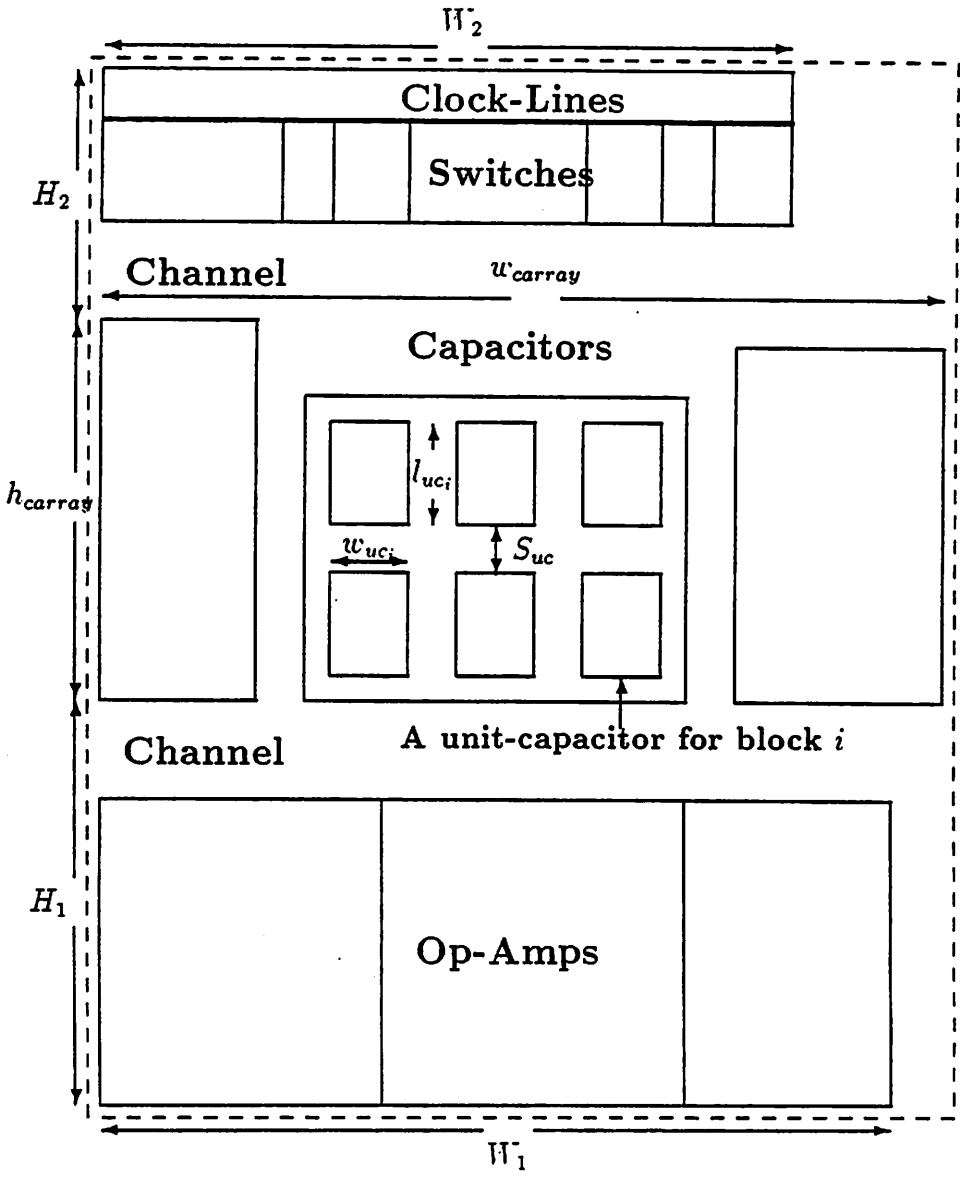



Figure 4.1: A geometric example for Problem 4.1. The bounding box of the filter is shown in dashed lines. A capacitor block has been expanded to show more detail.

$n_{y_i}$ = number of unit capacitors in the $y$ direction for block $i$,

$N_{tc_i}$ = number of capacitors with connections on top of block $i$,

$N_{bc_i}$ = number of capacitors with connections at the bottom of block $i$.

$N_{uc_i}$ = total number of unit capacitors for block $i$,

$A_{u_i}$ = area of unit capacitor for block $i$,

$S_{uc}$ = spacing between two adjacent unit capacitors,

$W_{min}$ = minimum allowable unit capacitor width,

$L_{min}$ = minimum allowable unit capacitor length,

$W$ = the larger of the widths of op-amp array and switch array,

$H$ = sum of the height of the switch array, the height of the op-amp array and estimate of the heights of the two channels,

$S_{caps}$ = sum of the separations of capacitor blocks required for route through channels.

Given these definitions, the height of the capacitor array, which is the height of the tallest capacitor block, can be written as:

$$h_{carray} = \max_i(n_{y_i}\ (l_{uc_i} + S_{uc})). \tag{4.1}$$

and its width can be written as:

$$w_{carray} = S_{caps} + \sum_{i=1}^{N_b} n_{x_i}(w_{uc_i} + S_{uc}). \tag{4.2}$$

The relation between $n_{x_i}$ and $n_{y_i}$, and also between $w_{uc_i}$ and $l_{uc_i}$ are given by:

$$n_{y_i} = \left\lceil \frac{N_{uc_i}}{n_{x_i}} \right\rceil \qquad i = 1, \ldots . N_b, \tag{4.3}$$

$$l_{uc_i} = \frac{A_{u_i}}{w_{uc_i}} \qquad i = 1 \ldots ., N_b. \tag{4.4}$$

Now we can formulate the following constrained optimization problem:

**Problem 4.1 (Find-Dimensions)** Minimize:

$$(H + h_{carray}) \max(W, w_{carray}) \tag{4.5}$$

subject to:

$$W_{min} \leq w_{uc_i} \leq \frac{A_{u_i}}{L_{min}} \qquad i = 1, \ldots, N_b. \tag{4.6}$$

$$\max(N_{tc_i}, N_{bc_i}) \leq n_{x_i} \leq N_{uc_i} \qquad i = 1, \ldots, N_b. \tag{4.7}$$

In the equations above, the technology-dependent parameters $L_{min}$ and $W_{min}$ set the bounds on permissible unit capacitor aspect ratios. The space of the above problem is $2N_b$ dimensional. A vector in the solution space is:

$$\vec{x} = (n_{x_1}, \ldots, n_{x_{N_b}}, w_{uc_1}, \ldots, w_{uc_{N_b}})^T.$$

Since the $n_{x_i}$'s and $n_{y_i}$'s can take on integer values only, the above problem cannot be solved via optimization algorithms which assume the continuity of the function to be optimized. Although integer programming techniques can be used, due to the intrinsic complexity of these algorithms, it would take too much time to find an optimum solution. Thus, I will present a heuristic algorithm that is not guaranteed to find an optimum solution, but finds an excellent ones in a fairly large set of test problems.

### 4.2.2 Algorithmic Solution

I start by presenting an efficient algorithm for finding the solution of the problem above for the case where the variables accept a continuous range of values. Then the only change in the problem formulation is the relation between $n_{x_i}$ and $n_{y_i}$ as shown below:

$$n_{y_i} = \frac{N_{uc_i}}{n_{x_i}} \qquad i = 1, \ldots, N_b. \tag{4.8}$$

Here, I present a gradient-free algorithm based on some properties of this continuous optimization problem. These properties will be presented in the form of theorems and lemmas. Lemma 4.1 states that if the upper bound on the value of $w_{carray}$ is less than $W$, then the problem has a trivial solution. Moreover, if this upper bound is greater than $W$, then according to Theorem 4.1, the optimal width of the capacitor array will also be greater than $W$. These properties will enable us to solve the problem by systematically solving a system of $N_b$ disjoint two dimensional optimization problems. A similar method will then be introduced which finds very good solutions for the original form of Problem 4.1.

Now, I present the properties of the continuous optimization problem. In the following, assume $\vec{x}^0 \doteq (n^0_{x_1}, \ldots, n^0_{x_{N_b}}, w^0_{uc_1}, \ldots, w^0_{uc_{N_b}})^T$ solves Problem 4.1.

**Lemma 4.1** *If*

$$\sum_{i=1}^{N_b} N_{uc_i}\left(\frac{A_{u_i}}{L_{min}} + S_{uc}\right) \leq W - S_{caps} \tag{4.9}$$

*the, for* $i = 1, \ldots, N_b$,

$$n^0_{x_i} = N_{uc_i}, \quad w^0_{uc_i} = \frac{A_{u_i}}{L_{min}}. \tag{4.10}$$

**Proof:** As a result of the condition of Equation 4.9 , the objective becomes the minimization of $h_{carray}$, which is at its minimum when $n_{x_i}$ and $w_{uc_i}$ assume their maximum values. □

The following lemma is required in the proof of Theorem 4.1:

**Lemma 4.2** *The height of the capacitor array is bounded below by:*

$$h_{carray} \geq L_{min} + S_{uc}, \tag{4.11}$$

*with the equality satisfied if and only if, for* $i = 1, \ldots, N_b$,

$$n_{x_i} = N_{uc_i}, \quad w_{uc_i} = \frac{A_{u_i}}{L_{min}}. \tag{4.12}$$

The proof is clear from the definition of $h_{carray}$.

**Theorem 4.1** *If*

$$\sum_{i=1}^{N_b} N_{uc_i}(\frac{A_{u_i}}{L_{min}} + S_{uc}) \geq W^{\cdot} - S_{caps}. \tag{4.13}$$

*then*

$$\sum_{i=1}^{N_b} n^0_{x_i}(w^0_{uc_i} + S_{uc}) \geq W^{\cdot} - S_{caps}. \tag{4.14}$$

**Proof:** By contradiction. Assume the theorem is false. Let $I = \{1, \ldots, N_b\}$ and $I_1 \subset I$ such that $\frac{N_{uc_i}}{n^0_{x_i}}(\frac{A_{u_i}}{w^0_{uc_i}} + S_{uc}) = h^0_{carray}$, $\forall i \in I_1$. Since $f(\vec{x}) = \sum_{i=1}^{N_b} n_{x_i}(w_{uc_i} + S_{uc})$ is a continuous function, there exists an open neighborhood $A$ of $\vec{x}^0$ such that $\forall \vec{x} \in A$ we have $\mid f(\vec{x}) - f(\vec{x}^0) \mid < \delta$ where $\delta = W^{\cdot} - S_{caps} - f(\vec{x}^0) > 0$. But $\mid f(\vec{x}) - f(\vec{x}^0) \mid < \delta$ implies $f(\vec{x}) < f(\vec{x}^0) + \delta = W^{\cdot} - S_{caps}$. Let us choose a point $\vec{x}^1 = (n^1_{x_1}, \ldots, n^1_{x_{N_b}}, w^1_{uc_1}, \ldots, w^1_{uc_{N_b}}) \in A$ such that $n^1_{x_i} = n^0_{x_i} + \epsilon_{1_i}$ and $w^1_{uc_i} = w^0_{uc_i} + \epsilon_{2_i}$ , where $\epsilon_{1_i} = \min(N_{uc_i} - n^0_{x_i}, \epsilon_0) \geq 0$ and $\epsilon_{2_i} = \min(\frac{A_{u_i}}{L_{min}} - w^0_{uc_i}, \epsilon_0) \geq 0$ and $\epsilon_0 > 0$ is the radius of a closed ball around $x^0$ enclosed in $A$. If for some $i \in I_1$ we have $\epsilon_{1_i} = \epsilon_{2_i} = 0$ or in other words $n^0_{x_i} = N_{uc_i}$ and $w^0_{uc_i} = \frac{A_{u_i}}{L_{min}}$, then according to Lemma 4.2 they should hold true $\forall i \in I$. Therefore, we have

$$\sum_{i=1}^{N_b} n^0_{x_i}(w^0_{uc_i} + S_{uc}) = \sum_{i=1}^{N_b} N_{uc_i}(\frac{A_{u_i}}{L_{min}} + S_{uc}) \geq W^{\cdot} - S_{caps}. \tag{4.15}$$

So we need to concentrate only on the case where for each $i \in I_1$ at least one of the inequalities $\epsilon_{1_i} > 0$ or $\epsilon_{2_i} > 0$ is satisfied. But then $h^1_{carray} = \max_i(\frac{N_{uc_i}}{n^1_{x_i}}(\frac{A_{u_i}}{w^1_{uc_i}} + S_{uc})) < h^0_{carray}$ and $\vec{x}^1$ satisfies all the constraints of Problem 4.1. while resulting in a smaller value of the objective function $(H + h_{carray})W^{\cdot}$. This contradicts the assumption that $\vec{x}^0$ solves the continuous version of Problem 4.1. □

As a result of Lemma 4.1, the problem has a trivial solution if the condition of Equation 4.9 is satisfied. Otherwise, we need to solve the problem only for the case when $w_{carray} \geq W^{\cdot}$, since, according to Theorem 4.1 the optimum $w_{carray}$ is greater than $W^{\cdot}$. Problem 4.2 is a re-formulation of the original problem.

**Problem 4.2** *Given the condition of equation 4.13 is satisfied,* Minimize:

$$(h + H)(S_{caps} + \sum_{i=1}^{N_b} n_{x_i}(w_{uc_i} + S_{uc})) \tag{4.16}$$

subject to:

$$W_{min} \leq w_{uc_i} \leq \frac{A_{u_i}}{L_{min}} \qquad i = 1\ldots, N_b, \tag{4.17}$$

$$\max(N_{tc_i}, N_{bc_i}) \leq n_{x_i} \leq N_{uc_i} \qquad i = 1, \ldots, N_b, \tag{4.18}$$

$$\frac{N_{uc_i}}{n_{x_i}}(\frac{A_{u_i}}{w_{uc_i}} + S_{uc}) - h \leq 0 \qquad i = 1\ldots, N_b. \tag{4.19}$$

In the new formulation, the non-differentiable max function (Equation 4.1) has been disposed of by introducing the new variable $h$ and the set of inequalities as in Equation 4.19. Now the dimension of the problem is $2N_b+1$ and a vector of the solution space is $\vec{x} = (h, n_{x_1}, \ldots, n_{x_{N_b}}, w_{uc_1}, \ldots, w_{uc_{N_b}})^T$. An examination of Problem 4.2 reveals that, by fixing $h$, the problem reduces to solving $N_b$ disjoint problems as formulated in SubProblem 4.1.

**SubProblem 4.1** *Given $h$, for $i = 1, \ldots, N_b$ minimize:*

$$n_{x_i}(w_{uc_i} + S_{uc}) \tag{4.20}$$

*subject to the constraints of Problem 4.2.*

This sets up the basis for Algorithm 4.1. This algorithm is a golden section technique for one-dimensional optimization [52], where the unknown variable is $h$ and the function to be minimized is implicitly evaluated by solving Problem 4.1.

Now we are at a point to handle the case where $n_{x_i}$'s take on integer values only. For the discrete case, we decompose the problem into $N_b$ smaller subproblems, as has been done with the continuous version of the problem.

**Algorithm 4.1 (Find-Continuous-Dimensions)**

** The objective function is defined as the total area **

$f_{obj}(\vec{x}) = (h + H)(S_{caps} + \sum_{i=1}^{N_b} n_{x_i}(w_{uc_i} + S_{uc}))$;

$\mathcal{F} = 1. - .61804$; ** The Fibonacci number **

** Lower bound on the height of the capacitor array **

$h_a = h_{min} = L_{min} + S_{uc}$;

** Upper bound on the height of the capacitor array **

$h_b = h_{max} = \max_i(\frac{N_{uc_i}}{\max(N_{tc_i}, N_{bc_i})}(\frac{A_{u_i}}{W_{min}} + S_{uc}))$;

** Initialize the search interval **

$\delta_h = h_{max} - h_{min}$;

** Successively reduce the interval in which the minimum lies **

** $\epsilon$ is a small positive number **

while ($\delta_h > \epsilon$) {

** divide the interval $(h_a, h_b)$ into $(h_a, h_b')$ and $(h_a', h_b)$ **

$h_a' = h_a + \delta_h\mathcal{F}$; $h_b' = h_b - \delta_h\mathcal{F}$;

*solve subproblem 4.1 for $h = h_b$ to find $\vec{x}_b$;*

*solve subproblem 4.1 for $h = h_a'$ to find $\vec{x}_a'$;*

*solve subproblem 4.1 for $h = h_b'$ to find $\vec{x}_b'$;*

** Check in which of the new intervals the minimum lies **

if ($f_{obj}(\vec{x}_b') \leq \min(f_{obj}(\vec{x}_a'), f_{obj}(\vec{x}_b))$ *or* $f_{obj}(\vec{x}_b') > f_{obj}(\vec{x}_b))$ $h_a = h_a'$;

else $h_b = h_b'$;

** Update the search interval **

$\delta_h = h_b - h_a$;

}

** Return the optimal solution vector **

$\vec{x}^0 = \vec{x}_a$;

even though Theorem 4.1 does not hold true any more. This heuristic is justified by the resulting filter areas which are close to their lower bounds (Table 4.2.4).

In the Algorithm 4.2, $r_{search}$ is the radius about $h^0_{carray}$ for exhaustive search, while $\delta_{res}$ confines the number of points to be searched to a finite number. In this algorithm, $\delta_{res}$ is set to the pattern generator's resolution.

Algorithm 4.3 solves SubProblem 4.1 for the discrete case. In this algorithm, for each value of $n_{x_i}$ in its domain, the value of $w_{uc_i}$ which minimizes the objective while satisfying the constraints is evaluated. Each $n_{x_i}$ is incremented in such a way to guarantee a decrease in the value of $\lceil \frac{N_{uc_i}}{n_{x_i}} \rceil$. Otherwise, the value of $w_{uc_i}$ remains unchanged for an increased value of $n_{x_i}$. This method of incrementing $n_{x_i}$ reduces the time complexity of the *while* loop from $N_{uc_i}$ to $\lfloor \sqrt{N_{uc_i}} \rfloor$ as shown in the following section.

### 4.2.3 Computational Complexity

In this section, we derive an upper bound on the time complexity of Algorithm 4.2. In the worst case the maximum number of iterations in the while loop of Algorithm 4.2 is $2\frac{r_{search}}{\delta_{res}}$. For each block $i$, the while loop of Algorithm 4.3 is executed a maximum of $2\sqrt{N_{uc_i}}$ times. This is supported by the following lemma:

**Lemma 4.3** *Let $N \epsilon Z^+$, $I = \{1, \ldots, N\}$, and $m = \lceil \frac{N}{i} \rceil$ for some $i \epsilon I$. Then the following assertions hold:*
*(a) if $n = \lceil \frac{N}{m} \rceil$ then $m = \lceil \frac{N}{n} \rceil$,*
*(b) the number of distinct values of $\lceil \frac{N}{n} \rceil$ for all values of $n \epsilon I$ is less than or equal to $2\lfloor \sqrt{N} \rfloor$.*

**Proof:**

(a) From the definition of ceiling function we can write:

$$m = \lceil \frac{N}{i} \rceil \Leftrightarrow (m-1)i < N \leq mi. \tag{4.21}$$

and:

$$n = \lceil \frac{N}{m} \rceil \Leftrightarrow (n-1)m < N \leq mn. \tag{4.22}$$

By combining the right-hand-side of the top inequality and the left-hand-side of the bottom inequality we get:

$$(n-1)m < N \leq mi \Rightarrow n-1 < i \Rightarrow n \leq i.$$

But then $(m-1)i < N \Rightarrow (m-1)n < N$. Combining this result with the right-hand-side of Equation 4.22 we obtain the desired conclusion:

$$(m-1)n < N \leq mn \Leftrightarrow m = \lceil \frac{N}{n} \rceil.$$

(b) Let us denote $S_N = \{k \mid k = \lceil \frac{N}{n} \rceil \text{ for some } n \in I\}$. From the symmetry property shown in part (a), there is a one to one correspondence between the elements of $S_N$ which are greater than $\lfloor \sqrt{N} \rfloor$ and those elements which are less than this value. So the maximum cardinality of this set is $2\lfloor \sqrt{N} \rfloor$. □

Then the time complexity of Algorithm 4.3 is proportional to $\sum_i^{N_b} \sqrt{N_{uc_i}}$. and that of Algorithm 4.2 is proportional to $\frac{r_{search}}{\delta_{res}} \sum_i^{N_b} \sqrt{N_{uc_i}}$.

### 4.2.4 Experimental Results

These algorithms have been tested on a variety of practical examples. Table 4.2.4 summarizes the parameters defining each test case. In Table 4.2.4, $h^{o,d}$ and $h^{o,c}$ are the height of the capacitor array as evaluated by the discrete and continuous algorithms respectively. Similarly, $A^{o,d}$ and $A^{o,c}$ are the total area of the filter evaluated by the discrete and continuous algorithms respectively. The cpu time is for a thousand evaluations of the while loop of Algorithm 4.2. The cpu time for execution of Find-Continuous-Dimensions has been omitted. since it is negligible compared to the cpu time taken by Algorithm 4.2. By allowing $n_{x_i}$'s to take on continuous range of values, we expand the dimensions of the optimization problem. Therefore

**Algorithm 4.2 (Find-Discrete-Dimensions)**

** $f_{obj}(\vec{x})$, $h_{min}$, and $h_{max}$ are defined as in Algorithm 4.1 **

** $h^0$ is the optimum value of $h_{carray}$ for the continuous problem **

** Find the minimum value of $f_{obj}(.)$ within a radius $r_{search}$ of $h^0$ **

$f_{opt} = \infty$;

$h = \max(h_{min}, h^0 - r_{search})$;

while $(h \leq \min(h_{max}, h^0 + r_{search}))$ {

*solve subproblem 4.1 to find $\vec{x}$;*

if $(f_{obj}(\vec{x}) < f_{opt})$ { $\vec{x}^0 = \vec{x}$; $f_{opt} = f_{obj}(\vec{x})$; }

$h = h + \delta_{res}$;

}

| $i$ | $N_{uc_i}$ | $A_{u_i}$ | $N_{c_i}$ | $N_{uc_i}$ | $A_{u_i}$ | $N_{c_i}$ | $N_{uc_i}$ | $A_{u_i}$ | $N_{c_i}$ |
|---|---|---|---|---|---|---|---|---|---|
| | filter lp. $N_b = 5$ | | | filter bp1. $N_b = 10$ | | | filter bp2. $N_b = 6$ | | |
| 1 | 25 | 2500 | 3 | 10 | 5852.25 | 1 | 23 | 2916 | 3 |
| 2 | 24 | 2500 | 2 | 80 | 714.5 | 4 | 22 | 676 | 2 |
| 3 | 29 | 2500 | 3 | 10 | 5852.25 | 1 | 24 | 600.25 | 2 |
| 4 | 17 | 2500 | 2 | 78 | 714.5 | 3 | 146 | 441 | 3 |
| 5 | 28 | 2500 | 2 | 10 | 5852.25 | 1 | 92 | 441 | 2 |
| 6 | | | | 125 | 441 | 3 | 82 | 441 | 1 |
| 7 | | | | 10 | 5852.25 | 1 | | | |
| 8 | | | | 78 | 714.5 | 3 | | | |
| 9 | | | | 10 | 5852.25 | 1 | | | |
| 10 | | | | 78 | 714.5 | 3 | | | |

Table 4.1: Parameter summary for three test cases used for optimal evaluation of capacitor block dimensions.

**Algorithm 4.3**

** Solves SubProbelm 4.1 for the discrete case **

for ( $i = 1 \ldots . N_b$ ) {

$n_{x_i} = \max( \left\lfloor \frac{A_{u_i}(L_{min}+S_{uc})}{h} \right\rfloor . \max(N_{tc_i}, N_{bc_i}))$;

$n^0_{x_i} = w^0_{uc_i} = \infty$;

** The solution is trivial if h is larger than the upper-bound on the **

** height of this capacitor block **

if ($h \geq \left\lceil \frac{N_{uc_i}}{\max(N_{tc_i}, N_{bc_i})} \right\rceil (\frac{A_{u_i}}{W_{min}} + S_{uc})$) {

$n^0_{x_i} = \max(N_{tc_i}, N_{bc_i})$; $w^0_{uc_i} = W_{min}$;

}

else while ($n_{x_i} \leq N_{uc_i}$) {

** For each feasible $n_{x_i}$, find the $w_{uc_i}$ which minimizes **

** the width of the capacitor block **

$w_{uc_i} = \max( \frac{\frac{A_{u_i}}{h}}{\lceil \frac{N_{uc_i}}{n_{x_i}} \rceil} - S_{uc}, W_{min} )$;

if ($(n_{x_i}(w_{uc_i} + S_{uc}) < n^0_{x_i}(w^0_{uc_i} + S_{uc}))$ { $n^0_{x_i} = n_{x_i}$; $w^0_{uc_i} = w_{uc_i}$; }

** Increment $n_{x_i}$ to guarantee a decrease in $n_{y_i}$ **

$n_{x_i} = \lceil \frac{N_{uc_i}}{\lceil \frac{N_{uc_i}}{n_{x_i}} \rceil - 1} \rceil$;

}

}

| ex. | filter | $W - T_s$ $(\mu m)$ | $H$ $(\mu m)$ | $L_{min}$ $(\mu m)$ | $h^{o.d}$ $(\mu m)$ | $\frac{h^{o.d}}{h^{o.c}}$ | $A^{o.d}$ $(mm^2)$ | $\frac{A^{o.d}}{A^{o.c}}$ | cpu sec.[a] |
|---|---|---|---|---|---|---|---|---|---|
| 1 | lp | 774 | 740.5 | 50 | 510 | 0.995 | 1.30 | 1.13 | 0.72 |
| 2 | lp | 774 | 740.5 | 40 | 537 | 1.075 | 1.17 | 1.03 | 1.64 |
| 3 | lp | 774 | 740.5 | 20 | 528 | 1.057 | 1.16 | 1.02 | 3.01 |
| 4 | lp | 1160 | 740.5 | 50 | 339 | 1.014 | 1.34 | 1.07 | 0.97 |
| 5 | lp | 1160 | 740.5 | 20 | 349 | 1.044 | 1.27 | 1.015 | 3.31 |
| 6 | lp | 400 | 740.5 | 50 | 567 | 1.035 | 1.024 | 1.07 | 0.71 |
| 7 | lp | 400 | 740.5 | 20 | 1161 | 1.038 | 0.694 | 1.03 | 1.49 |
| 8 | lp | 774 | 2000 | 50 | 510 | 0.995 | 2.61 | 1.13 | 0.75 |
| 9 | lp | 774 | 2000 | 20 | 528 | 1.057 | 2.32 | 1.01 | 3.17 |
| 10 | lp | 774 | 150 | 50 | 510 | 1.081 | 0.686 | 1.15 | .76 |
| 11 | lp | 774 | 150 | 20 | 510 | 1.02 | 0.618 | 1.04 | 3.02 |
| 12 | bp2 | 928 | 751.5 | 21 | 445 | 1.1 | 1.32 | 1.04 | 3.43 |
| 13 | bp2 | 928 | 751.5 | 10 | 417 | 1.035 | 1.3 | 1.02 | 6.52 |
| 14 | bp1 | 1544 | 729.5 | 21 | 517 | 1.018 | 2.29 | 1.01 | 1.3 |
| 15 | bp1 | 1544 | 729.5 | 10 | 517 | 1.018 | 2.29 | 1.01 | 2.26 |

[a]On a VAX 8650 computer

Table 4.2: Summary of test results for optimal evaluation of capacitor block dimensions.

$A^{o,c}$ is a lower bound for the minimum total area of the filter. The results of our experiments indicate that every $A^{o,d}$ is within few percent of its lower bounds. Also the values of $h^{o,d}$ and $h^{o,c}$ are very close in every case. This suggests that the radius for exhaustive search can be set to some small percentage of $h^{o,c}$.

## 4.3 Generation of Symbolic Matrices

### 4.3.1 Problem Formulation

After the first stage of capacitor array generation, the number of unit capacitors in the $x$ and $y$ directions for each block are determined. However, there is no information regarding the assignment of a particular unit capacitor to a given capacitor of a block. In the second stage, such information is produced in the form of symbolic matrices representing these blocks. The matrix representing block $i$ has $n_{y_i}$ rows and $n_{x_i}$ columns, where $n_{x_i}$ and $n_{y_i}$ were determined in the previous stage. Entries of these matrices are represented by name fields corresponding to unit capacitor sites. All the unit capacitors of a given capacitor are represented by the same name, which are distinct for any two capacitors in the same block. An entry not occupied by any of the capacitors is represented by a '0'. Two entries of a matrix are neighbors if they have the same row (column) number and their column (row) numbers differ by one. Thus a symbolic capacitor could be defined as a set of entries of one of these matrices such that at least one neighbor of each element of the capacitor is another element of the same capacitor. A formal definition of a symbolic capacitor is presented below:

**Definition 4.1** *A symbolic capacitor $C$ of size $\mid C \mid$ in a matrix $A_{pq}$, is a set of $\mid C \mid$ entries of $A_{pq}$, each entry defined as a pair of row and column indices, such that for $\mid C \mid > 1 \quad \forall (i,j) \in C \quad \exists (k,l) \in C$ such that $i-k=0$ and $\mid j-l \mid = 1 \quad$ or $j-l=0$ and $\mid i-k \mid = 1$.*

An example of a symbolic capacitor $C$ with $| C | = 5$ in a $3 \times 5$ matrix is

$$C = \{(1.2).(1.3).(2.2).(2.3).(2.4)\}.$$

The entries of this symbolic capacitor are represented by the name field $c$ in the matrix shown in Figure 4.2.

$$\begin{array}{ccccc} 0 & c & c & 0 & 0 \\ 0 & c & c & c & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}$$

Figure 4.2: An example of a symbolic capacitor.

The problem of generating a symbolic matrix can then be formulated as in Problem 4.3.

**Problem 4.3** *Given a matrix $A_{pq}$ and two ordered lists of capacitors $L_{top} = (C_1, \ldots, C_m)$ and $L_{bottom} = (C_{m+1}, \ldots, C_{m+n})$ such that*

$$\sum_{i=1}^{m+n} | C_i | \leq p\, q \quad \text{and} \quad q \geq \max(| L_{top} |, | L_{bottom} |),$$

*find an assignment of elements of $C_1$ through $C_{m+n}$ in the given matrix such that each capacitor satisfies the definition given above, each capacitor of $L_{top}$ has at least one element in the top row of the matrix, each capacitor of $L_{bottom}$ has at least one element at the bottom row of the matrix, and there exist a sub-order among the unit-capacitors in the top row corresponding to the order of capacitors in $L_{top}$ and similarly for bottom row and $L_{bottom}$.*

While this problem might not have a feasible solution for the initial dimensions of the matrix, we can see that by a sufficient increase in the number of columns, we can always find a solution.

### 4.3.2 Algorithmic Solution

Before discussing the solution for the general case of the problem above, I present an algorithm which solves the problem for the special case

where $L_{top}$ is empty. Algorithm 4.4 accepts as its input the dimensions of the matrix in number of unit capacitors and a list of capacitors with terminals at the bottom of the array. It grows the capacitors one at a time in order to fill up any holes produced as a result of growing previous capacitors. but at the same time not to block the growth of the future unit-capacitors.

In this algorithm the capacitors are placed in the order they are given in the bottom list. If the algorithm fails to find a feasible solution, the number of columns of the matrix is increased by one and the process is repeated. Figure 4.3.2 illustrates how this algorithm places the capacitors. In this figure, the unoccupied neighbors of a capacitor $i$ are represented by $0_i$. For all the tested algorithms, this algorithm has found a feasible solution for the initial matrix dimensions, if a solution exists. For other cases where a feasible solution does not exist, it has found the minimum increase in number of columns to obtain a solution.

**Definition 4.2** *A matrix $A'_{pq}$ is the* column-mirror *of matrix $A_{pq}$, if every $(i,j)$ entry of $A'_{pq}$ is $(i, q-j+1)$ entry of $A_{pq}$. Similarly, a matrix $A'_{pq}$ is the* row-mirror *of matrix $A_{pq}$, if every $(i,j)$ entry of $A'_{pq}$ is $(p-i+1, j)$ entry of $A_{pq}$.*

Now, we can present Algorithm 4.5, which solves the general case of Problem 4.3. In this algorithm, the top-capacitors are placed in a lower submatrix of $A$ in the reverse order that they appear in the top list. The number of rows of this submatrix is chosen to accommodate all the unit-capacitors belonging to the capacitors of the top list. Then, through two mirroring operations, the placed capacitors are transported to the top-left corner of the matrix $A$ in the order they appear in the top list. Algorithm 4.4 tends to allow all of the unused unit-capacitor sites to cluster at the top-right end of the submatrix. Therefore, the above-mentioned mirroring operations also relocate the unused capacitor sites to the top-left corner of the matrix. This increases the probability that these unused sites will be filled while growing the bottom capacitors in the following step of the algorithm. Figure 4.3.2

**Algorithm 4.4 (Place-Bottom-Caps($A. p. q. L_{caps}$))**

*$A$ is an empty or partially filled matrix with $q$ initial number of columns:*
*$p$ is the height of $A_{pq}$. the lower $p$ $q$ submatrix of $A$ used for placement:*
$L_{caps} = \{C_1 \ldots\ldots C_{n_{caps}}\}$:
for ( $i = 1 \ldots\ldots n_{caps}$ ) {
    *assign the leftmost unoccupied entry of the bottom row of $A$ to $C_i$:*
    while ( *all elements of $C_i$ are not placed* ) {
        *update all the unoccupied neigbors of $C_i$ in $A_{pq}$:*
        if ( *the list of unoccupied neighbors of $C_i$ is empty* ) {
            *add a column of zeros to the right end of $A$:*
            $q = q + 1$:
            *continue:* }
        *find $c_{min}$. the column number of the leftmost*
                *unoccupied neigbors of $C_i$:*
        if ( *the lowest entry of column $c_{min}$ is occupied* or $i == n_{caps}$)
            *among the leftmost neighbors of $C_i$*
                *assign the lowest one to $C_i$:*
        else *among the highest neigbors of $C_i$*
                *assign the leftmost one to $C_i$:*
    }
}

$$
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0_1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0_1 & 0 & 0 & 0 & 0
\end{array}
\rightarrow
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0_2 & 0 & 0 & 0 & 0 \\
1 & 2 & 0_2 & 0 & 0 & 0
\end{array}
\rightarrow
$$

$$
\begin{array}{cccccc}
0 & 0_2 & 0 & 0 & 0 & 0 \\
0_2 & 2 & 0_2 & 0 & 0 & 0 \\
1 & 2 & 0_2 & 0 & 0 & 0
\end{array}
\rightarrow
\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0_3 & 0 & 0 & 0 \\
1 & 2 & 3 & 0_3 & 0 & 0
\end{array}
\rightarrow
$$

$$
\begin{array}{cccccc}
0 & 0 & 0_3 & 0 & 0 & 0 \\
0 & 2 & 3 & 0_3 & 0 & 0 \\
1 & 2 & 3 & 0_3 & 0 & 0
\end{array}
\rightarrow
\begin{array}{cccccc}
0 & 0_3 & 3 & 0_3 & 0 & 0 \\
0 & 2 & 3 & 0_3 & 0 & 0 \\
1 & 2 & 3 & 0_3 & 0 & 0
\end{array}
\rightarrow
$$

$$
\begin{array}{cccccc}
0_3 & 3 & 3 & 0_3 & 0 & 0 \\
0 & 2 & 3 & 0_3 & 0 & 0 \\
1 & 2 & 3 & 0_3 & 0 & 0
\end{array}
\rightarrow
\begin{array}{cccccc}
3 & 3 & 3 & 0_3 & 0 & 0 \\
0_3 & 2 & 3 & 0_3 & 0 & 0 \\
1 & 2 & 3 & 0_3 & 0 & 0
\end{array}
\rightarrow
$$

$$
\begin{array}{cccccc}
3 & 3 & 3 & 0_3 & 0 & 0 \\
3 & 2 & 3 & 0_3 & 0 & 0 \\
1 & 2 & 3 & 0_3 & 0 & 0
\end{array}
\rightarrow
\begin{array}{cccccc}
3 & 3 & 3 & 3 & 0_3 & 0 \\
3 & 2 & 3 & 0_3 & 0 & 0 \\
1 & 2 & 3 & 0_3 & 0 & 0
\end{array}
\rightarrow
$$

$$
\begin{array}{cccccc}
3 & 3 & 3 & 3 & 0 & 0 \\
3 & 2 & 3 & 0_4 & 0 & 0 \\
1 & 2 & 3 & 4 & 0_4 & 0
\end{array}
\rightarrow
\begin{array}{cccccc}
3 & 3 & 3 & 3 & 0 & 0 \\
3 & 2 & 3 & 0 & 0_5 & 0 \\
1 & 2 & 3 & 4 & 5 & 0_5
\end{array}
\rightarrow
$$

$$
\begin{array}{cccccc}
3 & 3 & 3 & 3 & 0_5 & 0 \\
3 & 2 & 3 & 0_5 & 5 & 0_5 \\
1 & 2 & 3 & 4 & 5 & 0_5
\end{array}
\rightarrow
\begin{array}{cccccc}
3 & 3 & 3 & 3 & 5 & 0_5 \\
3 & 2 & 3 & 5 & 5 & 0_5 \\
1 & 2 & 3 & 4 & 5 & 0_5
\end{array}
\rightarrow
$$

$$
\begin{array}{cccccc}
3 & 3 & 3 & 3 & 5 & 0 \\
3 & 2 & 3 & 5 & 5 & 0 \\
1 & 2 & 3 & 4 & 5 & 5
\end{array}
$$

Figure 4.3: An example for algorithm 4.4. where $p = 3$. $q = 6$, $L_{caps} = \{C_1, C_2, C_3, C_4, C_5\}$. $|C_1| = 1$. $|C_2| = 2$. $|C_3| = 7$. $|C_4| = 1$, and $|C_5| = 5$.

| ex. | capacitors | | unit-cells | | height | width | | cpu |
|---|---|---|---|---|---|---|---|---|
| | top | bottom | top | bottom | | generated | lower bound | sec.[a] |
| 1 | 9 | 3 | 71 | 12 | 10 | 9 | 9 | 0.03 |
| 2 | 3 | 5 | 16 | 61 | 16 | 5 | 5 | 0.02 |
| 3 | 3 | 5 | 16 | 61 | 16 | 5 | 5 | 0.02 |
| 4 | 20 | 11 | 88 | 78 | 4 | 42 | 42 | 0.02 |
| 5 | 0 | 12 | 0 | 2048 | 108 | 19 | 19 | 0.96 |
| 6 | 0 | 12 | 0 | 2048 | 44 | 47 | 47 | 0.79 |
| 7 | 0 | 3 | 0 | 6 | 2 | 4 | 3 | 0.02 |

[a]On a VAX 8650 computer.

Table 4.3: Test results for Place-Unit-Caps Algorithm.

illustrates the steps of this algorithm. Table 4.3 summarizes some test results for this algorithm. In Example 7, although the lower bound on the number of columns is 3, the minimum number of columns to accommodate three capacitors $| C_1 | = 1$, $| C_2 | = 2$, and $| C_3 | = 3$, in a matrix of height 2 is four.

### 4.3.3 Computational Complexity

In this section the time complexity of Place-Bottom-Caps algorithm is analyzed. In our implementation of this algorithm, the matrix entries corresponding to the unoccupied neighbors of each capacitor $C_i$ are stored in a two dimensional binary search tree. In this tree, the first field is the column number, and the second field is the row number of the stored matrix entry. For each matrix entry assigned to a capacitor $C_i$, a maximum of three other entries have to be added to the two dimensional binary search tree. To find the location of the next matrix entry to be assigned to a capacitor $C_i$, only one search has to be performed in the binary tree. Then, the total number of operations on the search tree is less than four for the addition of each unit element. The average number of comparisons for each search in this two dimensional binary search tree is proportional to $\log(p) + \log(q)$.

**Algorithm 4.5 (Place-Unit-Caps($A, n_y, n_x, L_{top}, L_{bottom}$))**

$k = n_x$; $l = \left\lceil \frac{\sum_{C_i \in L_{top}} |C_i|}{k} \right\rceil$;

*initialize* $A$ = *empty matrix with* $n_y$ *rows and* $n_x$ *columns;*

$L_{cpas}$ = *reverse order of the list of capacitors of* $L_{top}$*;*

*Place-Bottom-Caps*$(A, l, k, L_{caps})$;

$A$ = column-mirror *of* $A$;

$A$ = row-mirror *of* $A$;

*Place-Bottom-Caps*$(A, n_y, n_x, L_{bottom})$;

step 1:

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| t3 | t3 | t1 | 0 |
| t3 | t2 | t1 | t1 |

step 2:

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | t1 | t3 | t3 |
| t1 | t1 | t2 | t3 |

step 3:

| | | | |
|---|---|---|---|
| t1 | t1 | t3 | t3 |
| 0 | t1 | t2 | t3 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

step 4:

| | | | |
|---|---|---|---|
| t1 | t1 | t3 | t3 |
| c2 | t1 | t2 | t3 |
| c2 | c3 | c3 | c3 |
| c2 | c2 | c3 | 0 |
| c1 | c2 | c3 | c4 |

Figure 4.4: An example for Algorithm 4.5.

Then the time complexity of Place-Bottom-Caps algorithm is proportional to $N(\log(p) + \log(q))$, where

$$N = \sum_{C_i \epsilon L_{caps}} | C_i |$$

is the total number of unit elements. Since $q \approx \frac{N}{p}$, then $\log(p) + \log(q) \approx \log(N)$ and the time complexity of the algorithm becomes $O(N \log(N))$.

## 4.4 Generation of Mask Layouts from Symbolic Matrices

This is the final stage of the generation of capacitor blocks. At this stage, the following actions are taken for each capacitor block:

- A layout for the unit-capacitor is generated.
- A matrix of equally spaced unit-capacitors is produced. The number of rows and columns of this matrix are those of the symbolic matrix.
- Bottom-plates of the unit-capacitors belonging to the same capacitor are joined.
- Top-plates of the unit-capacitors belonging to any of the continuous capacitors are connected.
- Top-plates of the unit-capacitors belonging to capacitors switched through the same switch to a summing node of an op-amp are connected.
- Constant area/perimeter layouts for fractional parts of the capacitors in the block are generated.
- For every capacitor with a fractional part, two adjacent unit-capacitors belonging to that capacitor are replaced by its fractional part.

The bottom plates are joined by filling the spacing between two adjacent unit-capacitors with the bottom-plate material. There are 2 methods available for connecting the top-plates. The first method uses minimum width extensions of the top-plate to make the connection between adjacent unit-capacitor top-plates, whereas the second method uses another layer of interconnect to make these connections. Some technologies do not support the second method. To obtain a better matching of unit-capacitors. the parasitic capacitance between the top-plate interconnect and bottom-plate has been accounted for in every unit-capacitor. This has been accomplished by adding a parasitic capacitance to those sides of a unit-capacitor where the top-plate is not connected to an adjacent one. Figure 4.5 shows the layout of generated capacitor block. The top-plate sharing of capacitors reduces the number of pins for sensitive signals in the routing channels. It should be noted that the switched capacitors are ordered in such a way that the capacitors with top-plates connected to the same switch are always adjacent. Then. there is one top-plate pin in the channel per set of such switched capacitors.

## 4.5 Area/Perimeter Constant Fractional Capacitors

The layout of ratio-accurate capacitors is problematic in that the area/perimeter ratio of the fractional part must be the same as that of the unit-capacitor. A solution to this problem is presented below. In the following. $w_{uc}$ and $l_{uc}$ are dimensions of the unit capacitor. and $w_{fp}$ and $l_{fp}$ are dimensions of the fractional part to be evaluated. Assume the area of the fractional part is $r$ times the unit-capacitor area. If we denote $a$ to be the width to length ratio of the unit-capacitor, we can write:

$$(\frac{A}{P})_{unit-capacitor} = (\frac{A}{P})_{fractional-part} \Rightarrow$$

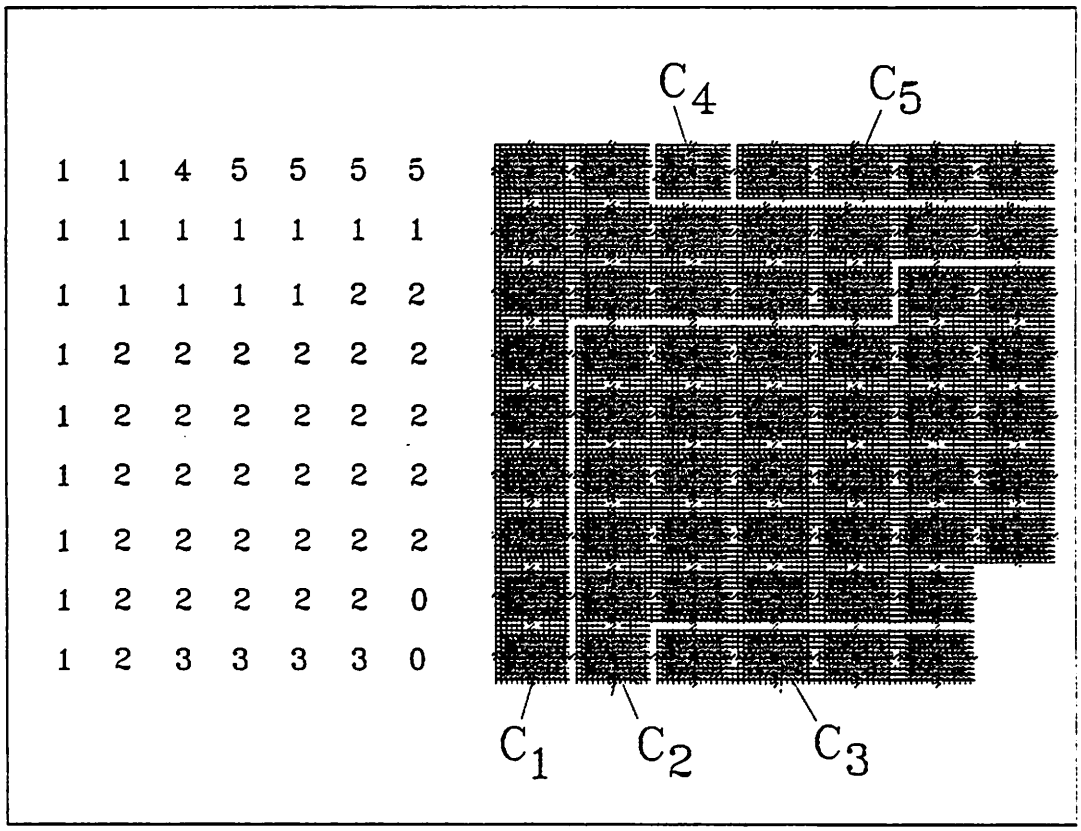



Figure 4.5: A symbolic matrix and the correponding capacitor block.

$$\frac{1}{w_{uc}+l_{uc}} = \frac{r}{w_{fp}+l_{fp}} \Rightarrow \frac{1}{(1+a)l_{uc}} = \frac{r}{\frac{ral_{uc}^2}{l_{fp}}+l_{fp}}$$

Now solving for $l_{fp}$ we find:

$$l_{fp} = l_{uc}\frac{r(1+a)}{2}\left[1 \pm \sqrt{1-\frac{4a}{r(1+a)^2}}\right], \tag{4.23}$$

$$w_{fp} = \frac{rw_{uc}l_{uc}}{l_{fp}} = w_{uc}\frac{r(1+a)}{2a}\left[1 \mp \sqrt{1-\frac{4a}{r(1+a)^2}}\right]. \tag{4.24}$$

In the above equations $l_{pf}$ and $w_{pf}$ have real solutions if

$$r \geq \frac{4a}{(1+a)^2}.$$

For the special case where $a = 1$ (square unit-capacitor), the above becomes $r \geq 1$. To save area in our method of capacitor block generation, only two unit-capacitor sites are allocated for the fractional part of each capacitor. To find the maximum $r$ for which the fractional part does not require more than two sites, we write:

$$l_{pf} \leq 2l_{uc} \Rightarrow r \leq r_{l.max} = \frac{4}{a+2},$$

$$w_{pf} \leq 2w_{uc} \Rightarrow r \leq r_{w.max} = \frac{4a}{2a+1},$$

where $r_{l.max}$ ( $r_{w.max}$ ) is the maximum attainable ratio if the sites used by the fracional part are in the same column (row). An examination of the equations above shows that if $a < 1$ then $r_{l.max} > r_{w.max}$, and therefore, if possible, the sites to be used by the fractional part should be chosen in the same column. With a similar argument the sites should be chosen, if possible, in the same row for $a > 1$. The value of $r_{max} = \max(r_{l.max}, r_{w.max})$ is 1.33 for $a = 1$ and 1.6 for $a = 2$. We are interested in the range of $r$ between 1 and 2. A new method of creating constant A/P fractional capacitors for $r > r_{max}$ has been devised. The main idea is to increase the perimeter of the layout for a given area by using rectilinear shapes instead of simple rectangular ones. Consider a rectangle of dimensions $l_{fp}$ and $w_{fp}$ with a rectangular area $\delta A$ removed

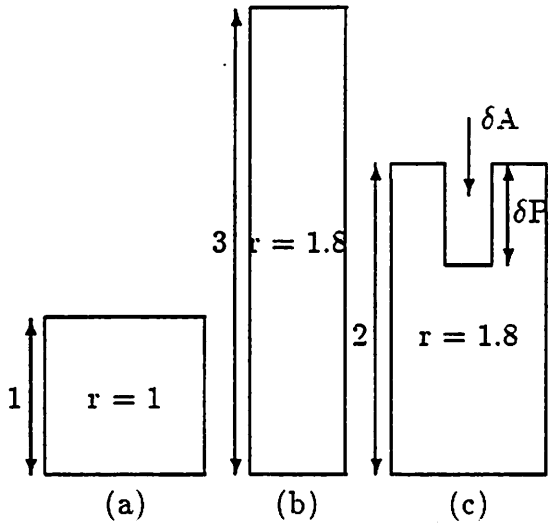



Figure 4.6: Geometry of A/P constant fractional parts. (a) A unit capacitor, (b) rectangular fractional capacitor, (c) rectilinear fractional capacitor.

from one side (figure 4.6). This results in $2\delta P$ increase in the perimeter of the original rectangle. In order for this geometry to have the same A/P ratio as the unit-capacitor we should have:

$$\frac{1}{(1+a)l_{uc}} = \frac{r}{\frac{ral_{uc}^2+\delta A}{l_{fp}} + l_{fp} + \delta P}.$$

Solving for $l_{fp}$ we find:

$$l_{fp} = \frac{1}{2}\left(l_{uc}r(1+a) - \delta P \pm \sqrt{(l_{uc}r(1+a) - \delta P)^2 - 4(ral_{uc}^2 + \delta A)}\right) \quad (4.25)$$

$l_{fp}$ has real solution if:

$$\delta P \leq l_{uc}r(1+a) - 2\sqrt{ral_{uc}^2 + \delta A} \quad (4.26)$$

Without loss of generality, assume the sites are chosen in the same column. Then:

$$l_{fp} \leq 2l_{uc} \Rightarrow r \leq \frac{4 + \frac{2\delta P}{l_{uc}} + \frac{\delta A}{l_{uc}^2}}{a+2} \quad (4.27)$$

The sum of $\delta A$ and the area of the fractional part should be less than twice the area of a unit-capacitor. Then:

$$\delta A \leq a(2-r)l_{uc}^2 \Rightarrow r \leq 2 - \frac{\delta A}{al_{uc}^2} \quad (4.28)$$

From the inequalities 4.27 and 4.28, a ratio of 2 can be achieved if $\delta P = al_{uc}$ and $\delta A$ approaches zero. However, because of the spacing between two adjacent unit-capacitor sites, $l_{fp}$ can be slightly greater than $2l_{uc}$, and hence a ratio of 2 can be achieved for some finite value of $\delta A$. This is consistent with the fact that $r = 2$ for A/P constant geometries can be achieved by parallel interconnection of two unit capacitors.

# Chapter 5

# Placement and Routing Issues

## 5.1 Introduction

Wiring parasitics have many degrading effects on the performance of SC filters. The most significant effects are listed as following:

- The cross-overs between switched or direct wiring to a summing node of an op-amp and wiring to a large signal node result in capacitive coupling of the two signals. This parasitic coupling might affect the accuracy of the filter response.

- In scaled technologies, the capacitive coupling between wires running in parallel become significant if the minimum spacing rules are used in the layout.

- Switched or direct wiring to a summing node of an op-amp can pick up the substrate noise, if the wire runs directly on top of the substrate. This degrades the PSR of the filter.

- The parasitic capacitances stemming from wiring the output nodes, add to the op-amp load. This limits the maximum clock rate.

To deal with the above issues, we have taken several measures:

- The modules are ordered in each array to minimize the overall wire length as well as minimize the sum of maximum density of each channel.
- The cross-overs between the connections to the summing nodes and connections to the large signal nodes. are detected and then shielded.
- A grounded well is placed under the capacitors and a large area of the routing channels to shield the capacitor plates and connections to the summing nodes from substrate noise.
- The capacitor top-plates which belong to the same electrical net. are shared inside the capacitor block. This reduces the number of pins in each channel.

In this chapter we discuss block ordering algorithms, introduce the channel router, and finally present the shielding methodology.

## 5.2 Wiring Optimization

The objective of the block ordering steps is three-fold:

- minimize the wiring length of the sensitive signals.
- minimize the number of cross-overs between the sensitive signals and the remaining ones,
- minimize the area of the filter occupied by the routing.

Accurate measurements of the above criteria are possible only after the routing is completed. This combined with the large number of cost function evaluations required by optimization algorithms calls for a quick estimate of the cost function. We propose the following four steps to achieve short wiring lengths for sensitive signals, small number of cross-overs, and small routing area.

1. The capacitor blocks are ordered to minimize estimated number of tracks needed for each channel.

2. The op-amps and switches are ordered in their respective arrays to minimize an estimate of the wire length in the top and bottom channels. The wire lengths are evaluated as the distance from an op-amp or switch terminal to the closest edge of a capacitor block.

3. The capacitors are ordered to reduce the number of unavoidable cross-overs inside the channels.

4. The op-amps and switches are re-ordered; however, this time wire lengths are evaluated as the distance from the op-amp or switch terminal to the capacitor terminal.

The small number of sensitive pins in each channel combined with wiring length optimization steps, tend to result in short wiring lengths for sensitive signals even though there is no explicit mention of these signals in the above steps.

The capacitor-block ordering step is based on an algorithm developed for printed circuit board ordering [53]. The graph representation of the problem was first reported in [54]. In the following we refer to the graph representation above as state-space graph.

### 5.2.1 Capacitor Block Ordering

While the placement of the switches (op-amps) influences the routing quality of the switch (op-amp) channel, the relative order of the capacitor blocks impacts that of both channels. For this reason the capacitor blocks are ordered first. This step, as shown in figure 3.10, precedes the generation of symbolic matrices, the stage where pin locations are determined for capacitor blocks. Even though at this point we don't know the pin locations, we can order the capacitor blocks to improve a measure of

routing quality described below. The capacitor blocks can be perceived as being interconnected through their connections to the op-amps and switches. Therefore, there would be one net associated with each op-amp or switch. Now the measure to improve can be defined as the total number of interconnections, as described above, crossing boundaries of adjacent blocks. Let $B = \{b_{left}, b_1, b_2, b_3, \ldots, b_r, b_{right}\}$ be the set of capacitor blocks including the dummy blocks $b_{left}$ and $b_{right}$. These dummy blocks serve as left and right position holders, and do not participate in the ordering. A connection between a capacitor block and $b_{left}$ ( $b_{right}$ ) represents the association of that capacitor block with a signal coming from left (right) into the SC layout. Let $S = \{s_1, s_2, \ldots, s_q\}$ be the set of nets common to at least two of the blocks. All the blocks having connections to at least one pin of the same switch (op-amp) are interconnected through some net in $S$. There is one net associated with signals coming from the left side of the layout and directly connecting to the capacitor blocks. Similarly there is a net for the signals coming from the right side of the layout.

Consider an arbitrary ordering of the blocks, for example

$$(b_{left}, b_{k_1}, b_{k_2}, \ldots, b_{k_r}, b_{right}).$$

Lets $S(A)$ be the set of signals associated with the blocks $A \subseteq B$, and $B_{k_i}$ be the set of blocks to the left of $b_{k_i}$, including $b_{k_i}$ itself. Then, the number of wires between $b_{k_i}$ and $b_{k_{i+1}}$ becomes:

$$d(B_{k_i}) = | S(B_{k_i}) \bigcup S(B - B_{k_i}) | \tag{5.1}$$

Therefore the criterion function to be minimized is:

$$\sum_{i=1}^{r-1} d(B_{k_i}). \tag{5.2}$$

Exhaustive search for an optimal solution requires testing $r$ permutations of the blocks. Cederbaum introduced the concept of state space graph for efficient representation of possible permutations of objects in a set [54].

The state space graph for a set $B$ is a directed acyclic graph $G = (V, E)$ with each node $v_i$ corresponding to a distinct subset of $B$ shown as $B_k(v_i)$. Suppose $B$ has $r$ elements. Then there are $2^r$ subsets of $B$ including the empty set and $B$ itself. The graph has one source node $v_s$ corresponding to the empty set, and one sink node $v_f$ corresponding to the set $B$ itself. This graph has $r + 1$ levels with the source node at level 0 and the sink node at level $r$. Each node at level $i$, $0 < i < r$, corresponds to a subset of $B$ with $i$ elements. There is an edge directed from a node $v_i$ representing subset $B_k(v_i)$ at level $i$, $0 \leq i < r$, to another node $v_j$ representing subset $B_k(v_j)$ at level $i+1$, if $B_k(v_i) \subset B_k(v_j)$. The node $v_i$ ($v_j$) is called the parent (child) of $v_j$ ($v_i$). Then, the total number of edges in the graph is $r2^{r-1}$. A property of this graph is that there exists a one-to-one correspondence between the ordering of the elements of $B$ and the directed source to sink paths on the graph. For each source to sink path, the order of the elements of $B$ is defined by the order they are added to the subsets along the path. Figure 5.1 is an illustration of one such graph. Sangiovanni [53] has reported an algorithm for solving the block ordering problem based on the above graph representation. In this algorithm the cost associated with each node $v_i$ of the state space graph is $d(B_k(v_i))$; then, the shortest path from the source to the sink gives the optimal ordering of the blocks. In the search strategy the graph is constructed incrementally by expanding the children of a selected node. The selection is according to an evaluation function $f(v)$. This function is the sum of two components. One is $c(v)$, the cost of tentative shortest path from $v_s$ to $v$, and the other one is $h(v)$, a estimate of the cost of shortest path from $v$ to $v_f$.

**Algorithm 5.1**

**Step 1** *Put the source node $v_s$ on a list called OPEN. Set $c(v_s) = 0$.*

**Step 2** *If OPEN is empty, exit with failure, otherwise continue.*

**Step 3** *Remove from OPEN the node whose $f$ value is smallest and put it*

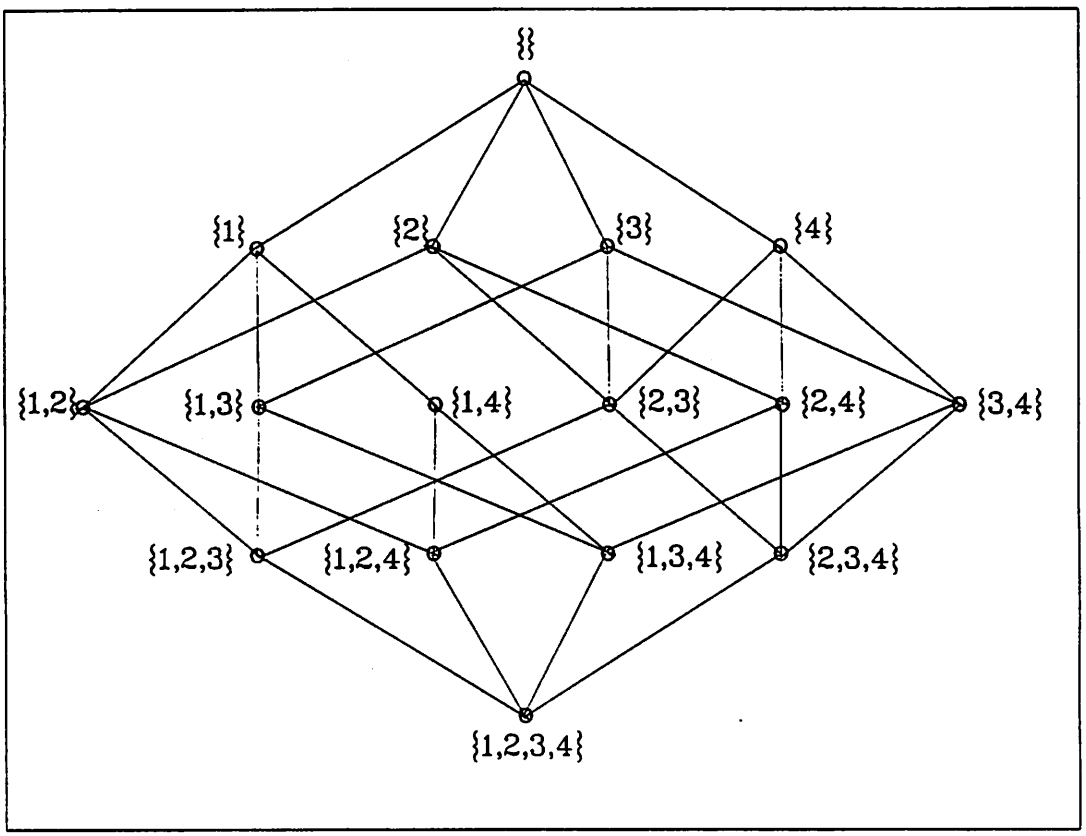



Figure 5.1: An example of a state space graph.

*on a list called CLOSED. If there are more than one node with minimal f value. chose the one closest to the source node.*

**Step 4** *If v is the goal node. exit with the solution path obtained by tracing back through the pointers; otherwise continue.*

**Step 5** *Generate all the children of v. For each child $v_i$, compute*

$$c_i = c(v) + d(B_k(v_i)).$$

**Step 6** *If a child $v_i$ is not already in OPEN , set*

$$c(v_i) = c_i$$

*and*

$$f(v_i) = c(v_i) + h(v_i),$$

*where*

$$h(v_i) = \sum_{b_j \in B - B_k(v_j)} \mid I(b_j) \mid - \mid \{s_k \in S \mid s_k \bigcap B_k(v_i) = \emptyset\} \mid -$$

$$max_{b_i \in B - B_k(v_i)} \mid I(b_i) \bigcap S(B_k(v_i)) \mid.$$

$$I(b_j) = \{s_k \in S \mid b_j \bigcap s_k \neq \emptyset\}.$$

*Put $v_i$ on OPEN and direct a pointer from it back to v.*

**Step 7** *If a child $v_i$ is already on OPEN and $c(v_i) > c_i$, then update it by setting*

$$c(v_i) = c_i$$

*and*

$$f(v_i) = c_i + h(v_i).$$

*Redirect the pointer from $v_i$ to v.*

**Step 8** *Go to Step 2.*

The above algorithm is guaranteed to find the optimal solution.

### 5.2.2 Op-Amp and Switch Ordering

The op-amps and switches are placed in two steps. At each step the location and orientation of each op-amp and switch within its corresponding array is evaluated so that an estimate of the total net length is minimized. The length of either array should be less than the width of the layout module, which has been evaluated for minimum area.

The first ordering step immediately follows capacitor block ordering. The purpose of this step is to find an initial placement of the switches and op-amps to guide the ordering of capacitors within each capacitor block. At this step, the wire length is estimated as the $x$ component of the distance from a terminal location to the closest edge of the capacitor block it connects to. If the terminal is located within the boundaries of the capacitor block, the wire length estimate will be zero.

The second ordering step is performed after the symbolic generation of the capacitor blocks. Now, the wire length estimate is the $x$ component of the distance from a switch or op-amp terminal to the capacitor terminal it is connecting to.

A simulated annealing approach [55] has been used to find the placement of the switches and the op-amps at each step. The general form of the simulated annealing algorithm is presented in algorithm 5.2. The algorithm halts when the temperature goes below a very small threshold value, or the cost does not change over a large number of outer loop iterations. The inner-loop criterion is simply a constant number of iterations at each temperature. The placement is started from an initial feasible state, where there is no overlapping of the modules and the length of the array is within the limits of the layout.

There are three types of moves in this implementation of the simulated annealing algorithm:

1. Swap two modules of the same size,

2. mirror a module about its center $x$ axis,

**Algorithm 5.2 (Simulated-Annealing)**

```
T = T0: * Initial temperature *
X = X0: * Initial state *
oldcost = FindCost( X0 ): * Find the initial cost *
while ( stopping criterion is not satisfied ) {
    while ( inner-loop criterion is not satisfied ) {
        Generate ( move ): * Generate a new move *
        * Apply the generated move to the current state *
        Xnew = NewState( move, X );
        newcost = FindCost( Xnew ): * The cost of the new state *
        if ( Accept (newcost, oldcost, T) ) {
            * Update the current state and cost *
            X = Xnew; oldcost = newcost;
        }
    }
    T = update( T ); * T = αT , 0 < α < 1 *
}

Accept ( newcost, oldcost, T ) * Returns yes if the new cost is accepted *
δC = newcost - oldcost;
if ( δC ≤ 0 ) { return ( yes ); }
else {
    * The cost is less likely to be increased at lower temperatures *
    y = e^(-δC/T);
    r = random ( 0, 1); * A random number between 0 and 1 *
    if ( r < y) { return ( yes ); }
    else { return ( no ); }
}
```

| no. of modules | max. array length ($\mu m$) | total width of modules ($\mu m$) | no. of sim. anneal. trials | avg. cpu sec.[a] | wire length ($\mu m$) | |
|---|---|---|---|---|---|---|
| | | | | | mean | std. dev. |
| 12 | 732. | 612. | 50 | 4.63 | 1155. | 4.7 |
| 14 | 916. | 711. | 50 | 5.92 | 647. | 23. |
| 19 | 945. | 945. | 40 | 5.77 | 2042.5 | 4.5 |
| 28 | 1515. | 1366. | 25 | 11.06 | 3041.5 | 109. |
| 38 | 1815. | 1815. | 25 | 11.84 | 6220. | 126. |

[a]On a VAX 8650 computer

Table 5.1: Test results for Algorithm 5.2.

3. displace a module from its current location without creating overlaps or increasing the length of the array beyond its limits.

From experiments on a number of examples, we found that the best choice of parameters for this version of simulated-annealing algorithm is to set $\alpha$ to 0.9, and the number of inner-loop iterations to ten times the number of modules in the array. Table 5.1 summarizes the test results on a number of examples.

### 5.2.3 Capacitor Ordering

At this step, which follows loose ordering of switches and op-amps, the top and bottom capacitors of each integrating block are ordered to minimize tangling of nets. The capacitors should be ordered in such a way that all the capacitors with top-plates belonging to the same net are clustered in the same group. This condition results in maximum top-plate sharing within a capacitor block. The integrating capacitor should always be either the first or the last capacitor in the list of ordered bottom capacitors. This guarantees that the op-amp output node (bottom plate of the integrating capacitor) is available to the switch channel from either right or left side of the capacitor block. Each capacitor terminal is connected to only one op-amp or switch. This brings about a natural mechanism, as explained below, for ordering the

capacitors. The top capacitors are first sorted according to the $x$ locations of the switch terminals connected to their top-plates. Then, the capacitors with the same top-plate connection are sorted according to the $x$ locations of the switch terminals connected to their bottom-plates.

The bottom capacitors belonging to the same integrating block have the same top-plate connection. Therefore, the bottom capacitors are sorted only according to their bottom-plate connections to the op-amps. Then the integrating capacitor is brought to the left (right) end of the sorted list if there are fewer number of capacitors to its left (right) than to its right (left) in the sorted list.

## 5.3 Channel Routing

The various placement steps discussed so far, usually result in very "clean" channels in the sense that the nets are very localized and the number of cyclic constraints is very low. This has made it possible to employ a modified version of a standard switch-box router for routing these analog channels. The following factors have contributed to the choice of MIGHTY [19] as the channel router within our layout system:

- For a large number of difficult examples, MIGHTY has completed the routing in fewer or as many number of tracks as any reported channel router has done.
- The sensitive nets can be routed before other nets. This can result in shorter lengths with fewer number of vias for sensitive interconnections.
- The use of low resistivity interconnect layer is maximized.
- MIGHTY can handle obstacles and terminals within the channel. The usefulness of this property becomes apparent in the next section.

The use of MIGHTY as a channel router requires prior knowledge of the number of tracks required to complete the routing. Initially, the number of

tracks is chosen as the maximum density of the channel. If MIHGTY fails to complete the routing, the number of tracks is incremented by one. This is repeated until the routing is completed.

## 5.4 Shielding of Crossovers

In some technologies, it is possible to break the capacitive coupling of signals resulting from one interconnect layer crossing over another, by covering the cross-over with a grounded third layer. This layer should be physically located between the two interconnect layers used in the routing (figure 5.2). In our layout scheme, the cross-overs will happen only in the routing channels. It should be noted that shielding adds to the capacitive loading of the op-amps. Therefore, it should be used only when necessary; then, only cross-overs between a sensitive-signal layer and a large-signal layer are shielded. The mechanism for grounding the shielding plates now follows. As a result of our layout strategy, there is a ground bus adjacent to each of the routing channels. The area of the routing channel, except for the location of vias, can be used to connect the shielding plates to the available ground bus. Reducing the area of the channel used to ground the shielding plates, reduces the effect of shielding on loading of op-amps. This served as motivation to develop a method for efficient grounding of shield plates. The following steps describe our method of shielding the cross-overs in a channel:

**Algorithm 5.3 (Shield-Cross-Overs)**

**Step 1** *Find the grid locations on the ground bus which can be used for grounding the shielding layer. Identify each of these locations with a different net number.*

**Step 2** *Detect all the cross-overs between sensitive and large-signal connections.*

**Step 3** *To each of the above cross-overs assign the net number of the ground-bus grid location which is closest to that cross-over.*

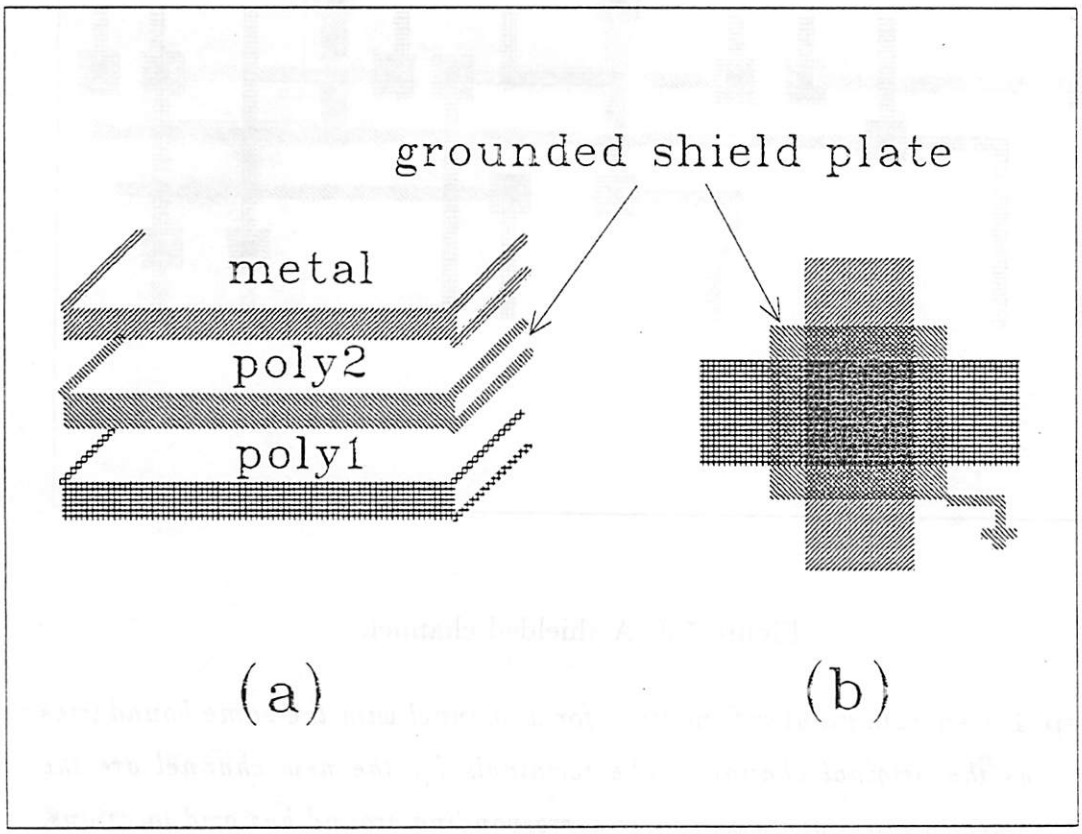



Figure 5.2: (a) Vertical positioning of the shield layer with respect to the routing layers. (b) A shielded cross-over.

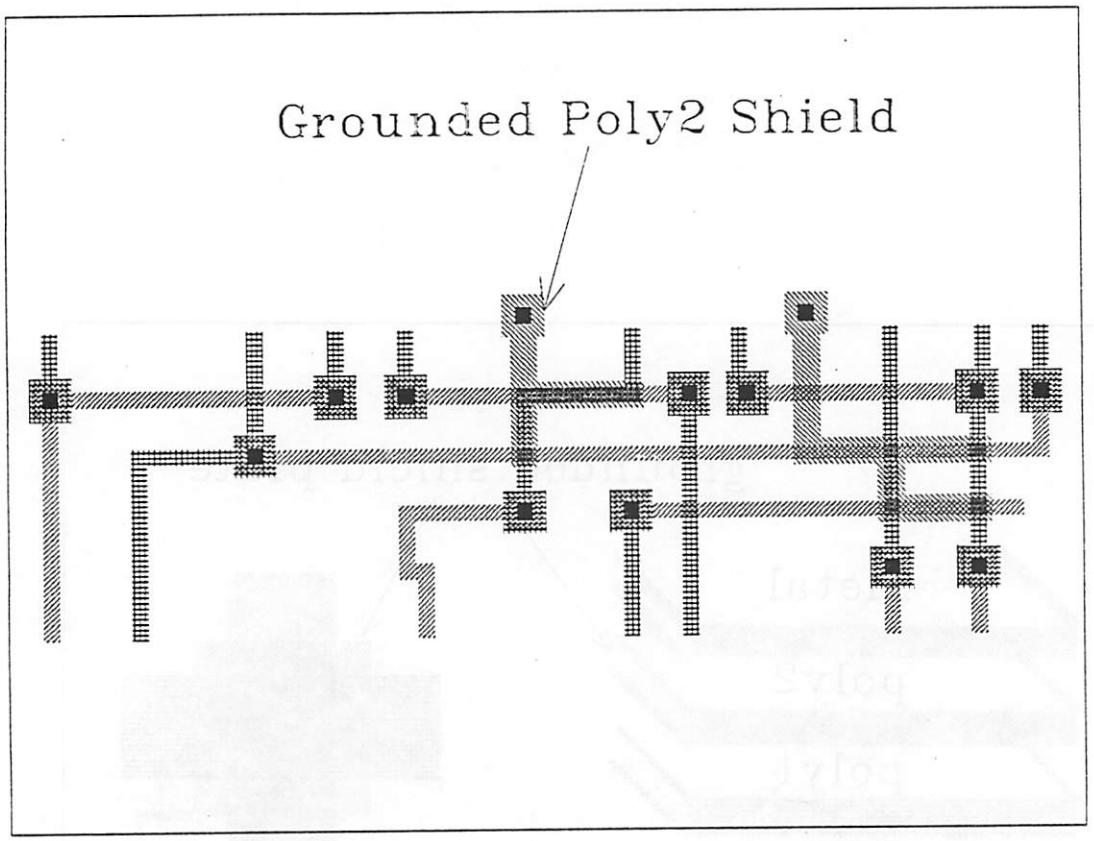



Figure 5.3: A shielded channel.

**Step 4** *Generate input information for a channel with the same boundaries as the original channel. The terminals for the new channel are the detected cross-overs and their corresponding ground-bus grid locations.*

**Step 5** *Add all the vias of the original channel as obstacles to the new channel.*

**Step 6** *Use MIGHTY to generate paths from cross-overs to the ground bus.*

Figure 5.3 illustrates a channel which has been subjected to the above steps.

# Chapter 6

# Experimental results

In this chapter the measurements on a set of prototype filters generated using the layout generator [56] will be discussed. The goals of these experiments were two folds. The first goal was to demonstrate that the program is capable of generating area-efficient filters with good performance. The vehicles for this were a tenth order chebychev bandpass filter and an elliptic PCM lowpass filter. The circuit schematics of the lowpass and bandpass filters are shown in figures 6.1 and 6.2 respectively. The second goal was to demonstrate the program's flexibility in generating area-efficient layouts for a wide spectrum of sampling capacitor values. For this we used the PCM lowpass filter with three different sampling capacitor values of 0.25 $pF$. 1 $pF$ which corresponds to the standard filter. and 4 $pF$. All the filters were fabricated in a 3 micron double poly CMOS technology [57]. The op-amp used by the filters has a folded cascode structure. This op-amp has an area of 150 $mils^2$ and comes from a commercially available library [58].

Figure 6.3 shows a photomicrograph of the bandpass filter. This filter. which has a ladder structure. was designed for a center frequency of 5 $kHz$ and a bandwidth of 100 $Hz$ when it is clocked at 256 $kHz$. The continuous LC ladder filter was generated with the help of Filsyn [41] and the resulting filter was then transformed into an equivalent SC filter. The computer generation time for the filter layout was 17.58 seconds on a VAX

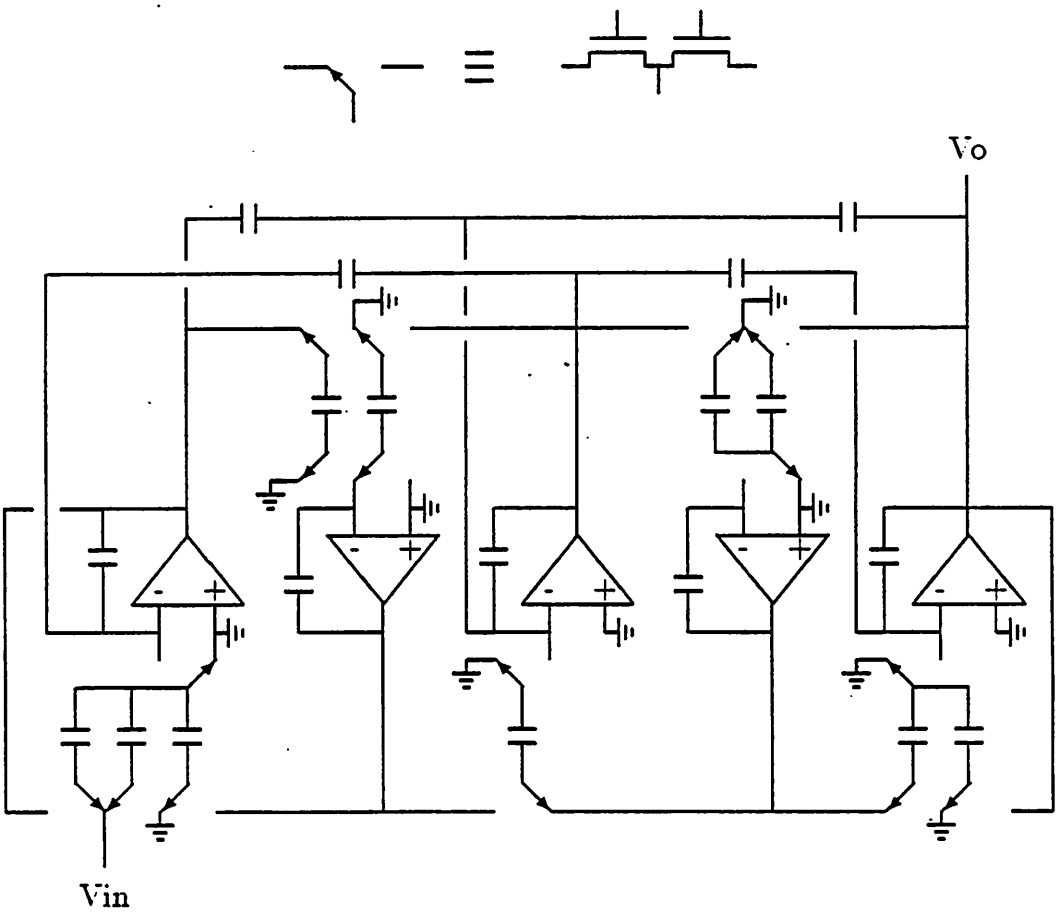



Figure 6.1: Circuit schematic of the PCM low-pass filter.

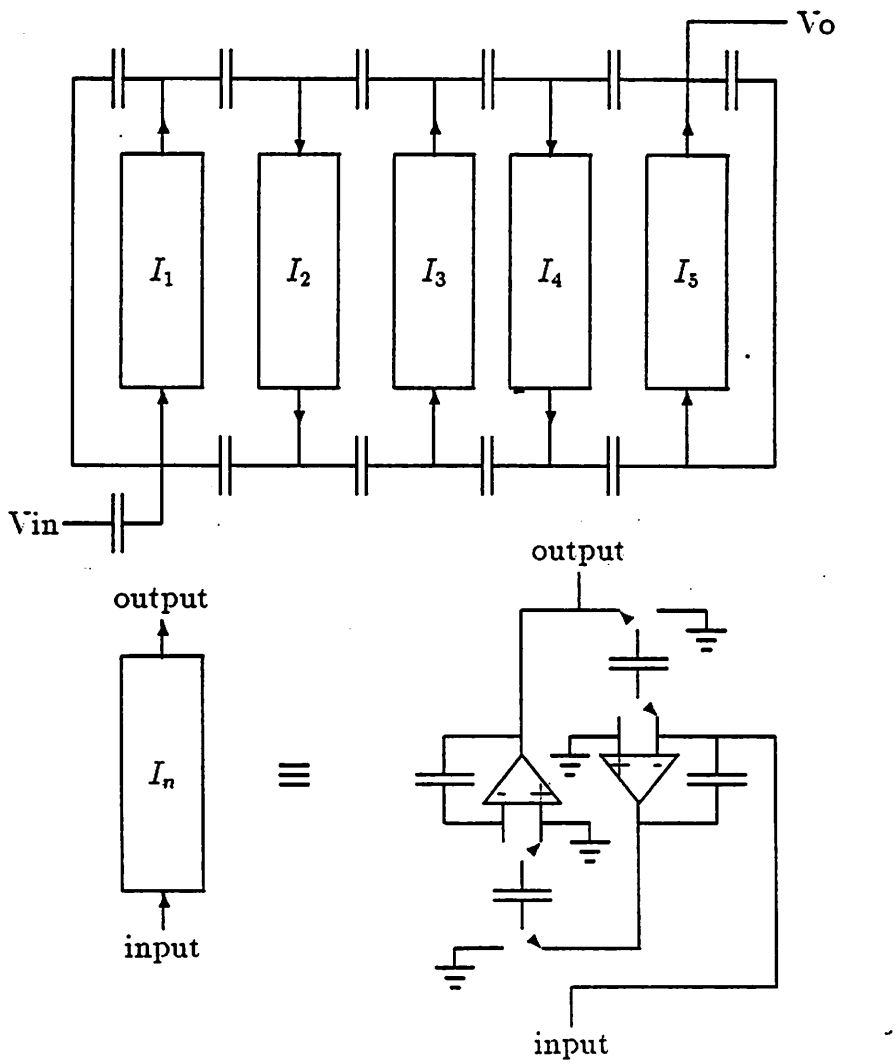



Figure 6.2: Circuit schematic of the 10th order band-pass filter.

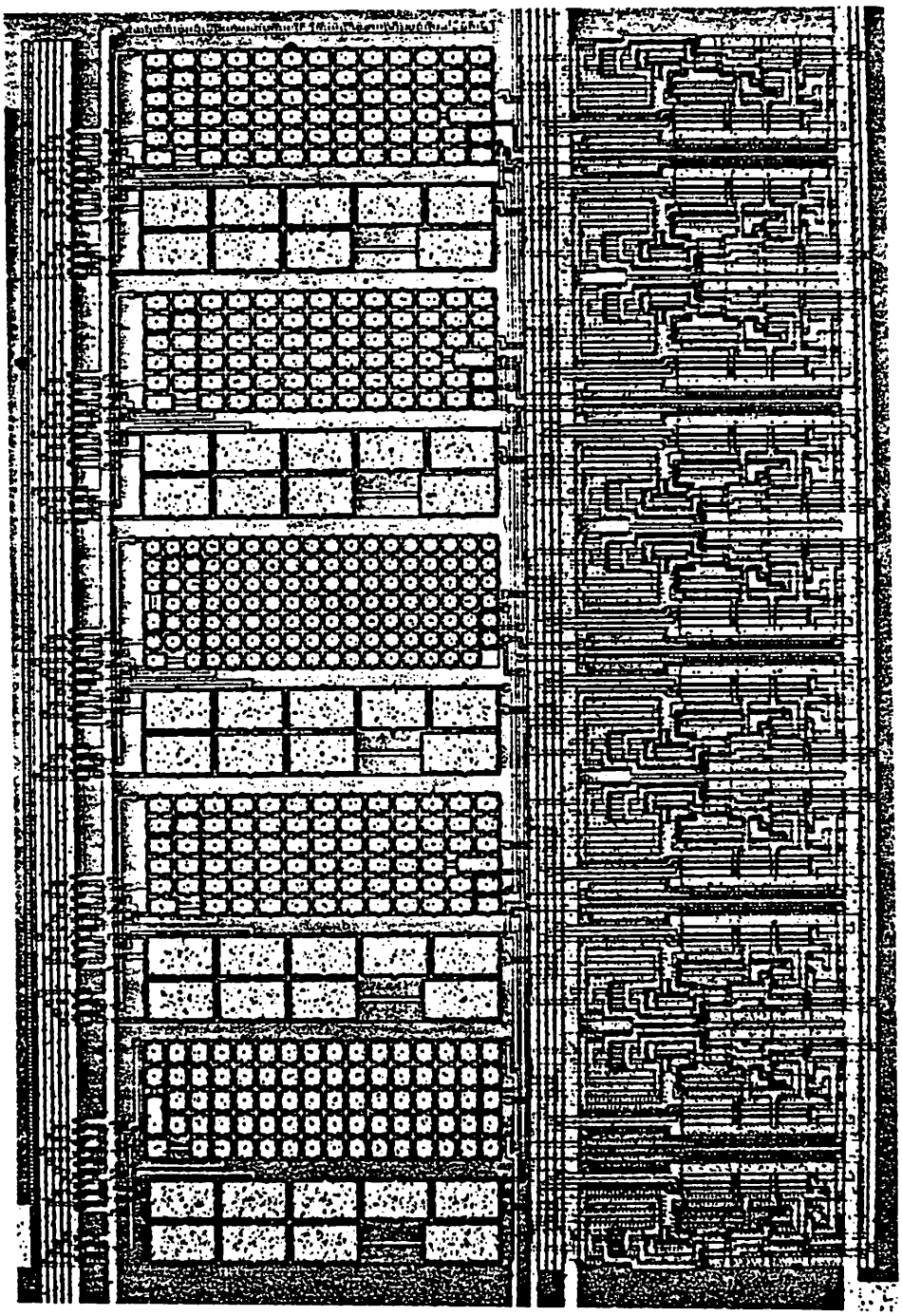


Figure 6.3: Photomicrograph of the 10th order band-pass filter.

8800 computer, and the area of the filter was 3600 $mils^2$ for a total capacitance of 220 $pF$. This is an interesting example because it illustrates the effect of area optimization on the layout when the unit-capacitors of different integrators have drastically different values.

The basic measure of performance for SC filters is how well the measured and ideal response match. The measured response can be different from ideal because of process sensitive capacitor geometries and layout parasitics. Figure 6.4 compares the measured and simulated frequency response of the bandpass filter. The switched-capacitor simulation program Switcap [59] has been employed for all the filter simulations. The simulated response of the bandpass filter does not include the extracted layout parasitics. however it accounts for the finite gain of the op-amps. The maximum deviation of the measured frequency from the simulated was less than 0.1 $dB$ in the pass-band and less than 1 $dB$ in the stop-band. This confirms good capacitor matching as well as small amount of layout parasitics.

The choice of the PCM filter was motivated by the fact that it is a standard filter in telephony and makes a good benchmark because there are many existing hand layouts for the same filter. Figure 6.5 shows a photomicrograph of the standard filter with 1 $pF$ sampling capacitor value. The total area of the filter was 1750 $mils^2$ which can be compared to an area of more than 2000 $mils^2$ for a commercial filter layout manually designed in comparable technology. Again we observe a very close match between the measured and ideal frequency response of this filter (figure 6.6). For this filter the simulation results are for the ideal case and therefore do not include the effects of parasitics and the finite behavior of the op-amps, such as their finite gain and bandwidth.

An important advantage of synthesis programs is that they allow designers to explore various dimensions of the design space very quickly. For example for the PCM lowpass filter a key design parameter is the choice of sampling capacitor value which in turn affects area, power supply rejection and noise. All the integrators of the standard PCM filter have 1 $pF$ sampling

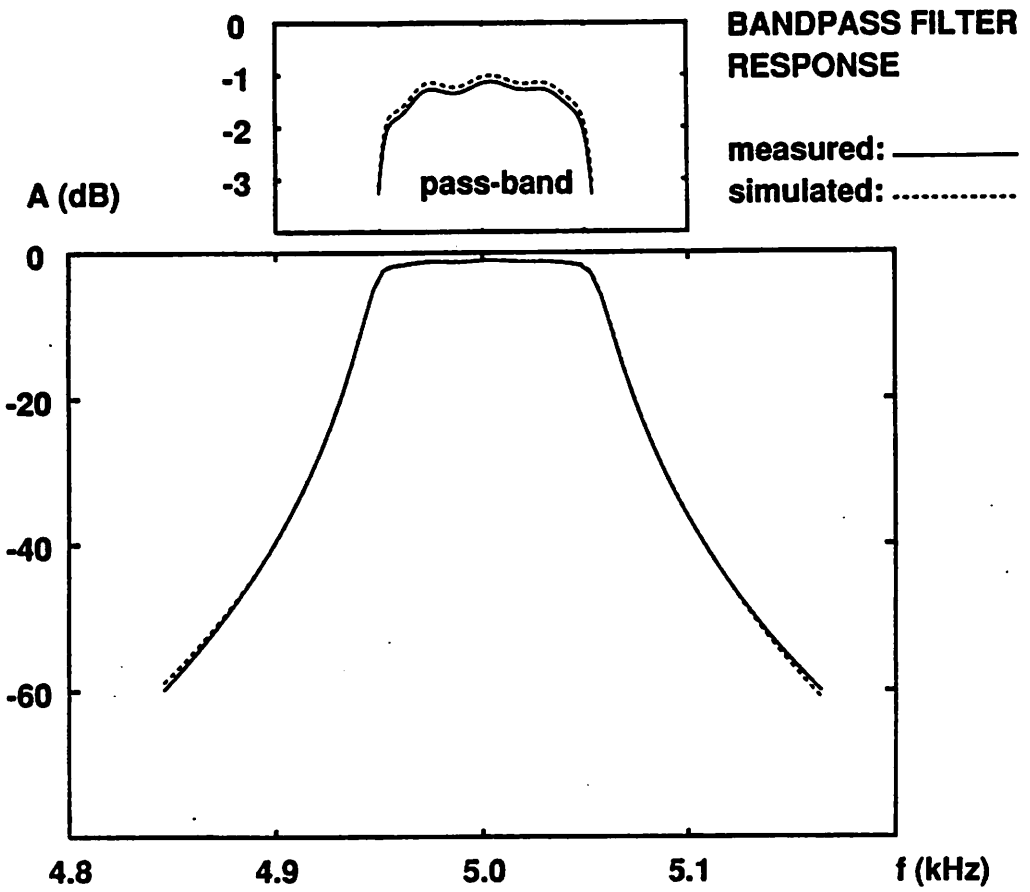



Figure 6.4: Frequency response of the 10th order band-pass filter.

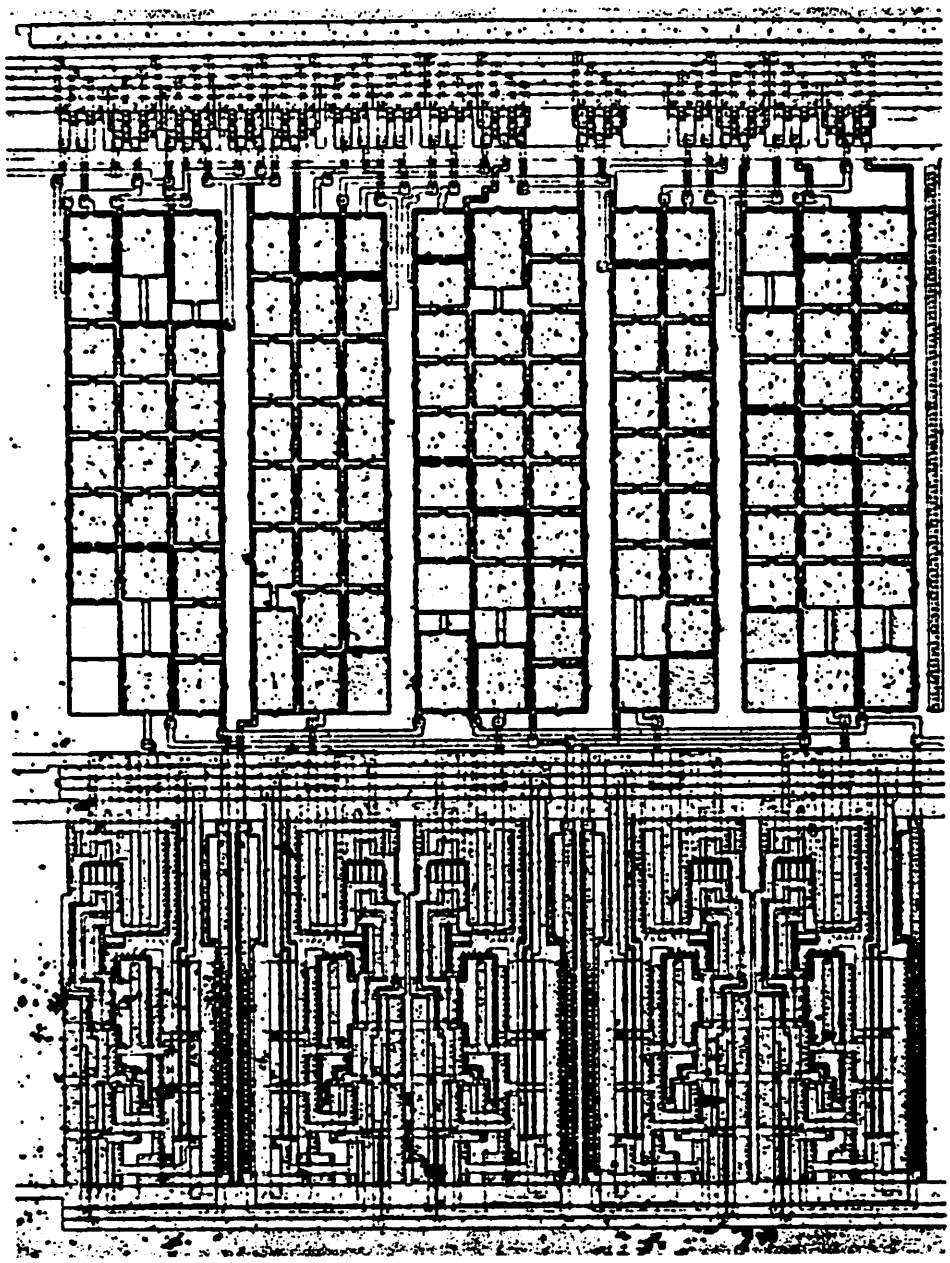


Figure 6.5: Photomicrograph of the standard PCM filter.

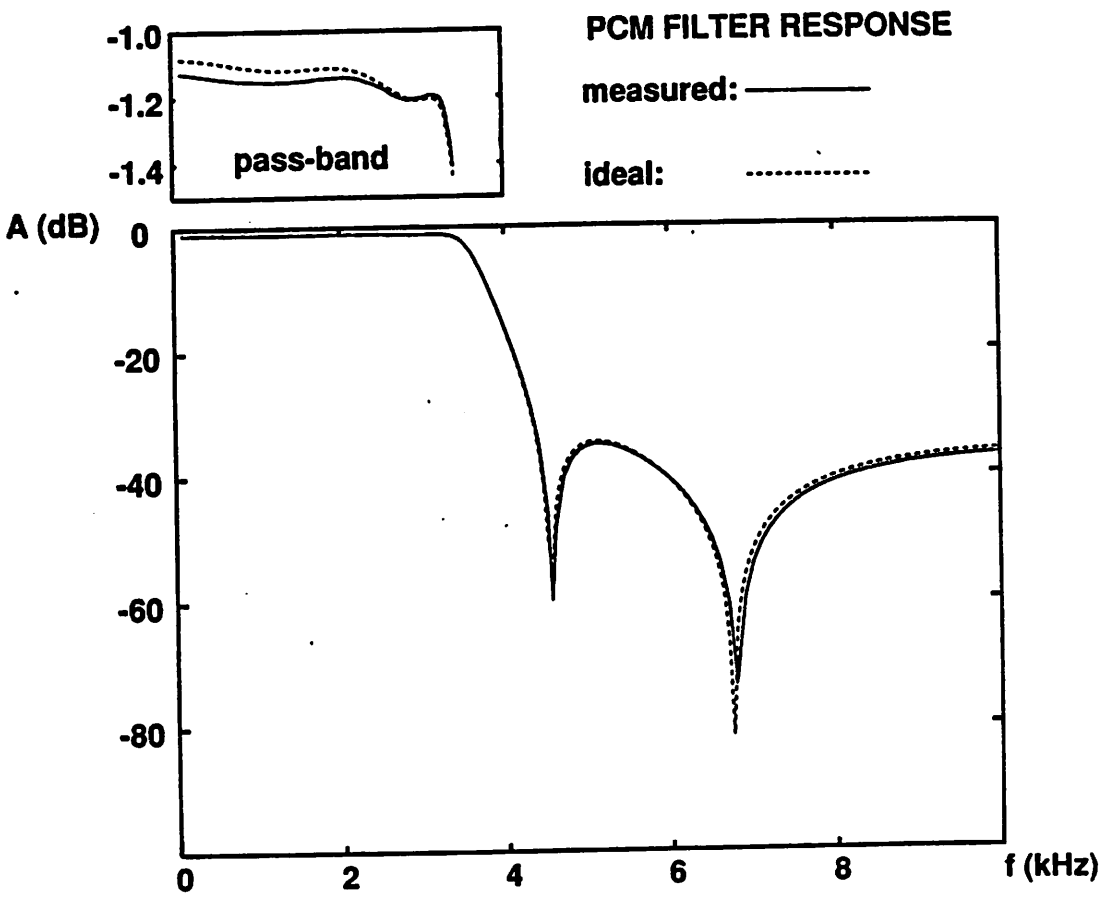



Figure 6.6: Frequency response of the PCM filter.

capacitor size. To study the effects of the sampling capacitor value on performance. two other filters with circuit topology and capacitor ratios identical to the standard filter were generated. When the sampling capacitor value was reduced by a factor of four, the layout shown in figure 6.7 was generated. For four times the standard sampling capacitor value, the filter of figure 6.8 was generated. For this filter, we can observe that both the height and width of the layout are increased over the standard filter. For sampling capacitor values of 0.25, 1, and 4 pF, the required areas were 1250, 1750, and 3900 $mils^2$ respectively. Figure 6.9 shows a size comparison of these filters.

The most interesting aspect of these filters is their noise performance. To a first order, the noise of an SC filter ($v_{sc}$) comes from two sources: the noise due to opamps ($v_{oa}$) which is approximately independent of the capacitor values, and the noise due to switches ($v_{sw}$) which is inversely proportional to integrating capacitor values. Since the noise power is additive we can write:

$$v_{sc}^2 = v_{oa}^2 + v_{sw}^2$$

Now consider two filters with identical topology and capacitor ratios. If the sum of capacitor values of filter 1 is $C_1$ and the sum of capacitor values of filter 2 is $C_2 = \alpha C_1$ such that every capacitor of filter 1 is $\alpha$ multiple of corresponding capacitor of filter 2, then:

$$C_1 v_{sw_1}^2 = C_2 v_{sw_2}^2 \Rightarrow C_1(v_{sc_1}^2 - v_{oa}^2) = C_2(v_{sc_2}^2 - v_{oa}^2).$$

From the above equation we can separate filter noise components by measuring the overall noise of the two filters. The inband noise of the filters were measured using the set-up of figure 6.10. Figure 6.11 shows total measured inband noise as a function of total capacitance for the PCM lowpass filter. This curve was obtained as a best fit for the three PCM filter noise measurements. From these measurements the noise contribution of the op-amps we used for the given process was 20 $\mu V$.

Figure 6.12 compares the power supply rejection (PSR) of the three low-pass filters with clock regulation. It is apparent from these measurements

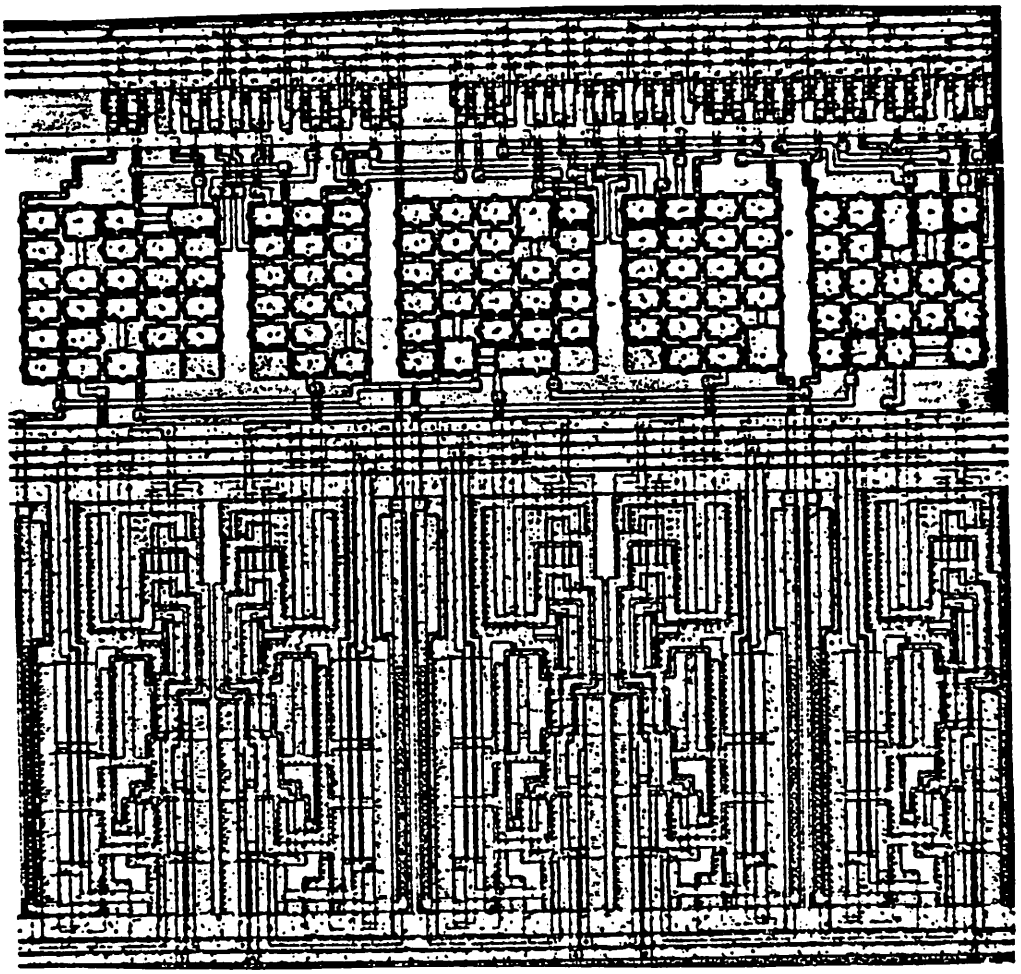


Figure 6.7: Photomicrograph of the PCM filter with 0.25 pF sampling capacitor.

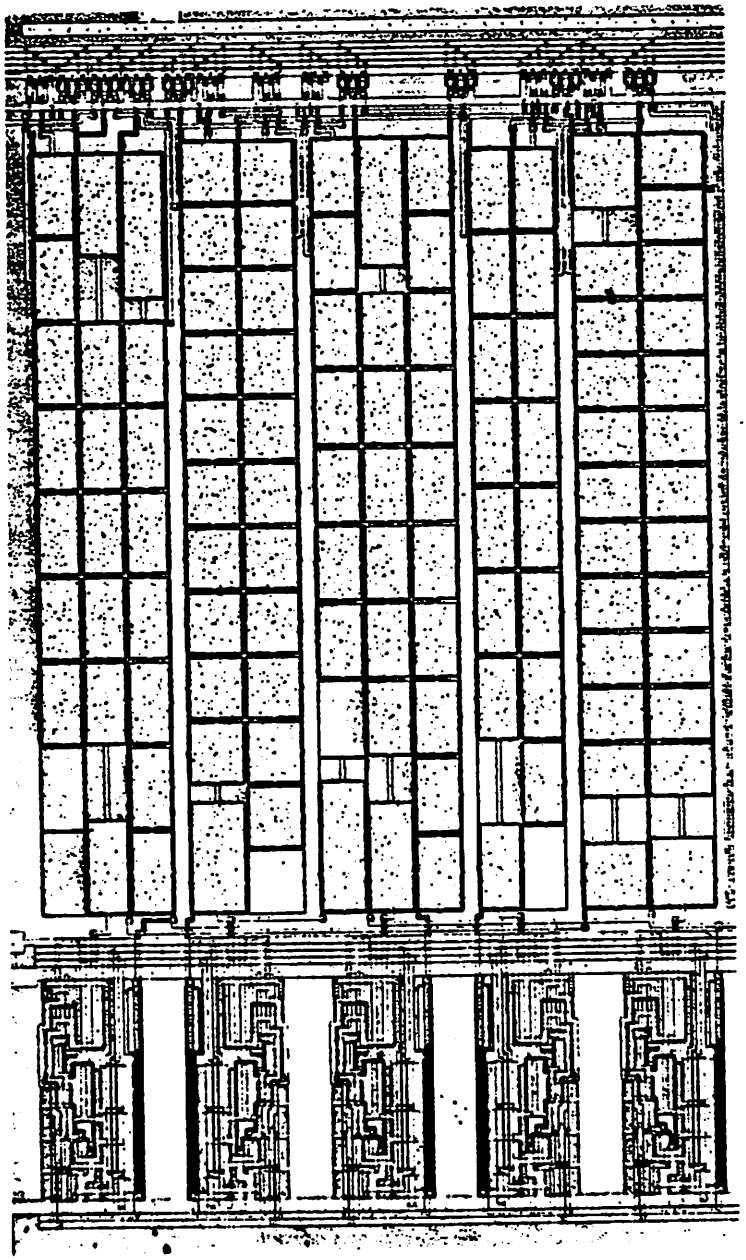


Figure 6.8: Photomicrograph of the PCM filter with 4 pF sampling capacitor.

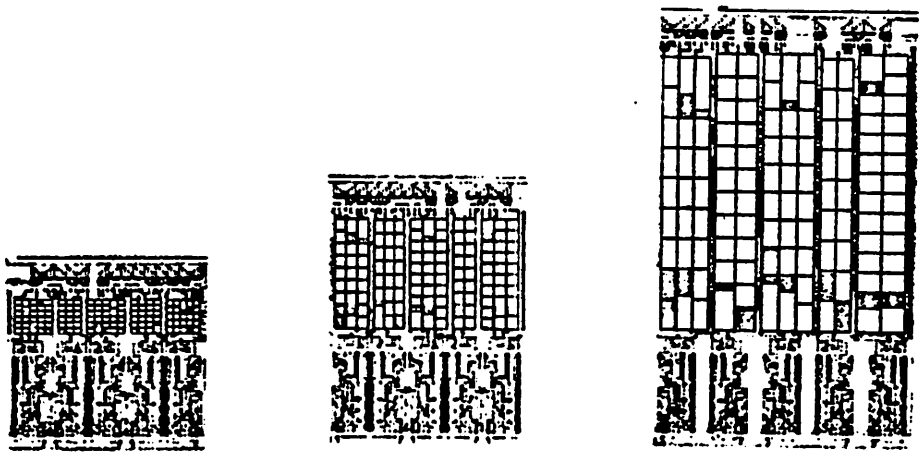



Figure 6.9: Photomicrographs of the PCM low-pass filters with total capacitances of (a) 31.5 pF, (b) 126 pF, (c) 504 pF.

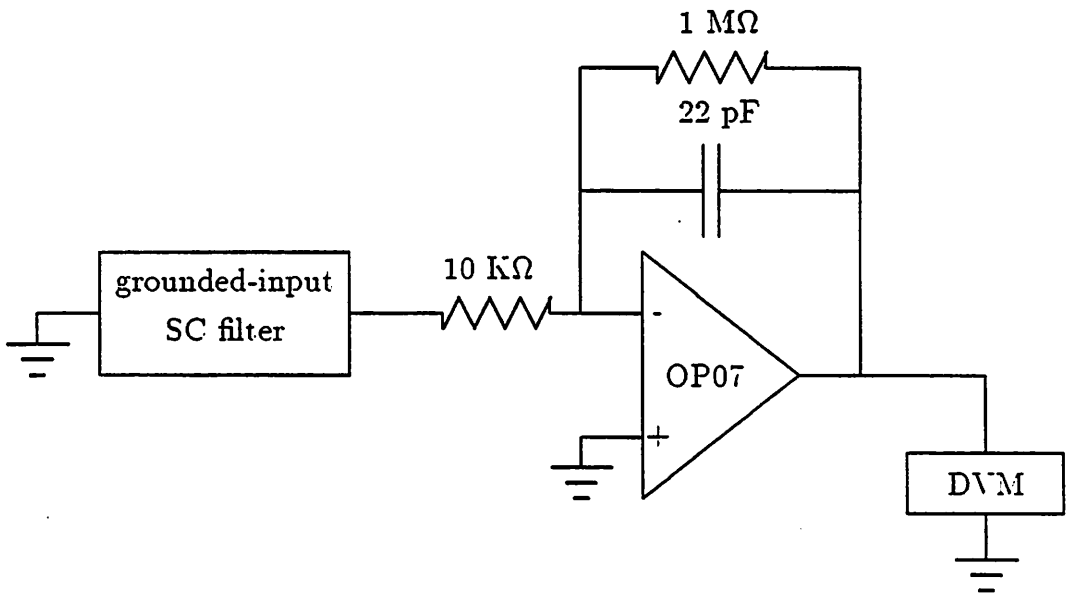



Figure 6.10: The set-up for in-band noise measurements.

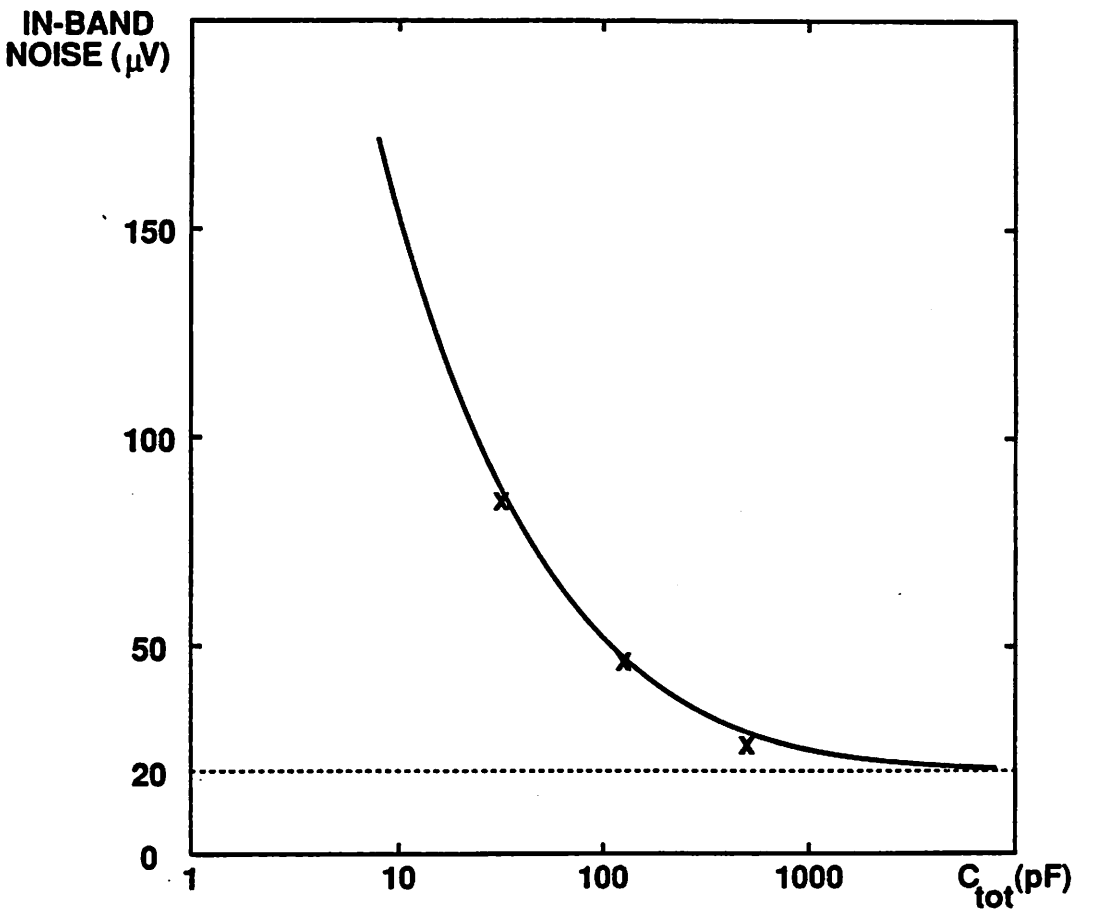



Figure 6.11: In-band measured noise as a function of total capacitance for the PCM low-pass filter.

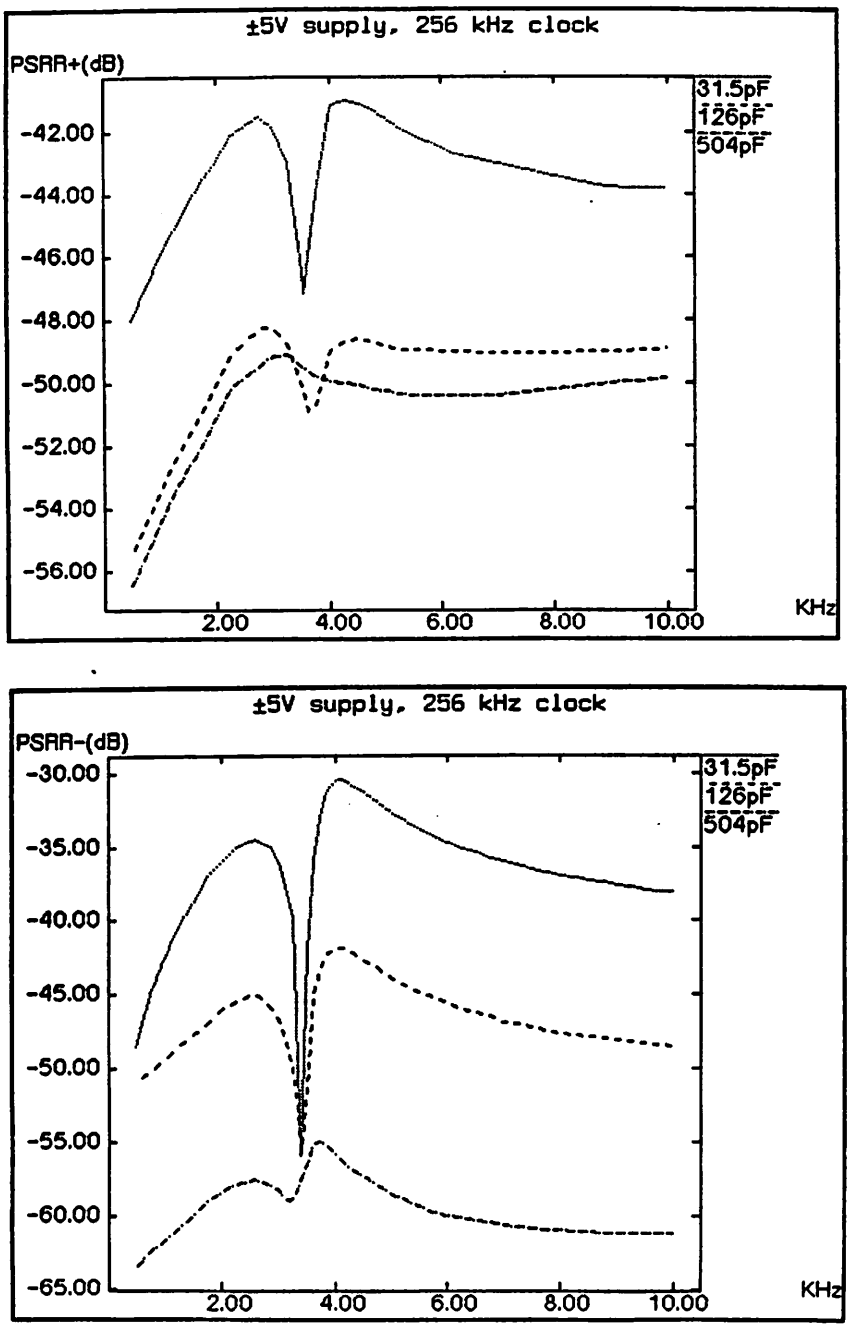



Figure 6.12: Power supply rejection of the PCM low-pass filters. (a) For positive supply. (b) For negative supply.

that there is a trade-off between the PSR and area of the filters. This might be somewhat explained by the observation that by increasing the total capacitance of the filter. the ratios of parasitic capacitances responsible for power supply degradation, to integrating capacitors are reduced. The limit to this improvement is expected to be the contribution of PSRs of the op-amps to PSR of the filter.

Table 6.1 shows a comparison of the layout-dependent performances of the three PCM filters. All three filters have excellent PSR which is partial

| total capacitance ($pF$) | generation time[a] (cpu *sec.*) | area per pole ($mils^2$) | total noise ($\mu V$) | PSRR+ @ 1 $kHz$ ($dB$) | PSRR- @ 1 $kHz$ ($dB$) |
|---|---|---|---|---|---|
| 31.5 | 10.18 | 250 | 85.7 | 46 | 42.5 |
| 126 | 8.99 | 350 | 47.8 | 53.6 | 49.4 |
| 504 | 15.79 | 780 | 27.8 | 54.5 | 61.7 |

[a]On a VAX 8650 computer

Table 6.1: Summary of the performances of the PCM low-pass filters.

confirmation of good filter layouts. The short computer generation time for these filters encourages various similar experiments.

# Chapter 7

# Conclusions and Future Directions

A layout system for SC filters has been presented. The performance of the system has been analyzed both theoretically and experimentally. The success of the project can perhaps be attributed to the use of expert advise on layout issues responsible for filter performance degradation, use of rigorous algorithmic approaches in exploiting the available degrees of freedom in the layout, and restricting the problem to a computationally manageable size. This and other similar work have shown the feasibility of automatic synthesis tools for analog applications.

The most natural extensions of this work seem to be automatic layout generation for fully-differential and programmable SC filters. A difficult problem in automatic layout of fully-differential filters is efficient routing of balanced signals. There is already effort in this direction [60]. With such capability, the layout system can be used to produce area-efficient layouts for fully-differential filters, as well.

A large number of SC filters are designed in such a way that their frequency response parameters can be programmed by the application of digital control signals [61,62]. Such filters generally require a large number of switches and capacitors per op-amp. Thus, the layouts generated by this

system for programmable filters are generally long with large amount of wasted area between the op-amps. Therefore, new layout floor-planning is required for area efficient generation of programmable filters.

Some classes of Analog-to-Digital (A/D) and Digital-to-Analog (D/A) converters require precise matching of several capacitors. Thus, the method presented in this thesis for generation of compact ratio-accurate capacitor arrays can serve as a first step towards automatic layout generation for these classes of A/D and D/A converters.

A challenging problem in SC filter module generation is to closely couple the synthesis and layout generation steps. This can result in more efficient circuits. Among the difficulties are developing a fast but accurate method of simulating the filter performance criteria. The simulation results should include layout parasitics and non-ideal behavior of filter elements. An understanding of how the layout variables affect the performance of the filter can help develop efficient methods for coupling the filter synthesis and the layout generation steps.

# Bibliography


[1] Robert W. Brodersen. Paul R. Gray, and David A. Hodges. MOS switched-capacitor filters. *Proceedings of the IEEE*, 67(1):61–75. January 1979.

[2] I. Buchanan. *Modelling and Verification in Structured Integrated Circuit Design.* PhD thesis, University of Edinburgh, Scotland, 1980.

[3] Robert Brayton, Richard Rudell, and Alberto Sangiovanni-Vincentelli. MIS: a multiple-level logic optimization system. *IEEE Transactions on Caomputer-Aided Design*, CAD-6(6), November 1987.

[4] J. Burns. A. Casotto, M. Igusa. F. Marron, F. Romeo. A. Sangiovanni-Vincentelli, H. Shin. G. Srinath, and H. Yaghutiel. Mosaico: an integrated macro-cell layout system. In *VLSI 87*, August 1987.

[5] P. P. Moore. *OCT Database Programmer's Manual.* Internal Memorandum. University of California. Berkeley, January 1986.

[6] Phillip E. Allen. Edmond R. Macaluso. Stephen F. Bily. and Ashok Nedungadi. AIDE2: an automated analog IC design system. In *IEEE Custom Integrated Circuits Conference*, pages 498–501, 1985.

[7] P. E. Allen and P. R. Barton. A silicon compiler for successive approximation A/D and D/A converters. In *IEEE Custom Integrated Circuits Conference*, 1986.

[8] C. D. Kimble, A. E. Dunlop, G. F. Gross, V. L. Hein, M. Y. Luang, K. J. Stern, and E. J. Swanson. Autorouted analog VLSI. In *IEEE Custom Integrated Circuits Conference*, pages 72–78, 1985.

[9] J. Trontelj, L. Trontelj, T. Pletersek, G. Shenton, M. Robinson, K. Floyd, and C. Jungo. Expert system for automated mixed analog digital layout compilation. In *IEEE Custom Integrated Circuits Conference*, pages 165–167, 1987.

[10] Marc DeGrauwe, Olivier Nys, Eric Vittoz, Evert Dijkstra, Jef Rijmenants, Serge Bitz, Stefan Cserveny, and Jose Sanchez. An analog expert design system. In *IEEE International Solid-State Circuits Conference*, pages 212–213, 1987.

[11] Ramesh Harjani, Rob A. Rutenbar, and L. Richard Carley. A prototype framework for knowledge-based analog circuit synthesis. In *24th ACM/IEEE Design Automation Conference*, pages 42–49, 1987.

[12] H. Yaghutiel, A. Sangiovanni-Vincentelli, and P. R. Gray. A methodology for automated layout of switched-capacitor filters. In *IEEE International Conference on Computer-Aided Design*, pages 444–447, 1986.

[13] Han Young Koh, Carlo H. Sequin, and Paul R. Gray. Automatic synthesis of operational amplifiers based on analytic circuit models. In *IEEE International Conference on Computer-Aided Design*, pages 502–505, 1987.

[14] Jacob K. White. *The Multirate Integration Properties of Waveform Relaxation, with Applications to Circuit Simulation and Parallel Computation.* PhD thesis, University of California, Berkeley, November 1985.

[15] Resve A. Saleh. *Nonlinear Relaxation Algorithms for Circuit Simulation.* PhD thesis, University of California, Berkeley, April 1987.

[16] Kartikeya Mayaram and Donald O. Pederson. CODECS: an object-oriented mixed-level circuit and device simulator. In *IEEE International Symposium on Circuits and Systems*, pages 604–607, May 1987.

[17] Kenneth S. Kundert and Alberto Sangiovanni-Vincentelli. Simulation of nonlinear circuits in the frequency domain. *IEEE Transactions on Computer-Aided Design*. CAD-5(4):521–535, October 1986.

[18] William Nye, David C. Riley, Alberto Sangiovanni-Vincentelli, and Andre L. Tits. DELIGHT.SPICE: an optimization-based system for the design of integrated circuits. *IEEE Transactions on Computer-Aided Design*, 7(4):501–519, April 1988.

[19] H. Shin and A. Sangiovanni-Vincentelli. MIGHTY: a rip-up and reroute detailed router. In *IEEE International Conference on Computer-Aided Design*, 1986.

[20] David L. Fried. Analog sampled-data filters. *IEEE Journal of Solid-State Circuits*, SC-7:302–304, August 1972.

[21] Ken Martin. Improved circuits for the realization of switched-capacitor filters. *IEEE Transactions on Circuits and Systems*, CAS-27(4):237–244, April 1980.

[22] L. Zadeh. Frequency analysis of variable networks. *Proceedings of IRE*. 38:291–299, March 1950.

[23] Jyuo-Min Shyu and Alberto Sangiovanni-Vincentelli. ECSTASY: a new enviroment for IC design optimization. In *IEEE International Conference on Computer-Aided Design*, 1988.

[24] Tat Cheung Choi. *High Frequency CMOS Switched-Capacitor Filters*. PhD thesis, University of California, Berkeley, May 1983.

[25] James L. McCreary. Matching properties, and voltage and temperature dependence of MOS capacitors. *IEEE Journal of Solid-State Circuits*, SC-16(6):608–616, December 1981.

[26] Martin Hasler. Stray capacitance insensitive switched capacitor filters. In *IEEE International Symposium on Circuits and Systems*, pages 42–45, 1981.

[27] J. M. Bermudez and B. B. Bhattacharyya. A systematic procedure for generation and design of parasitic insensitive SC biquads. *IEEE Transactions on Circuits and Systems*, CAS-32(8):767–783, August 1985.

[28] M. S. Lee and C. Chang. Exact synthesis of n-path switcehd-capacitor filters. In *IEEE International Conference on Circuits and Systems*, pages 166–169, April 1981.

[29] Nowjand Attaie and Ezz I. El-Masry. Multiple-loop feedback switched-capacitor structures. *IEEE Transactions on Circuits and Systems*, CAS-30(12):865–872, December 1983.

[30] Ezz I. El-Masry and Hock-Loo Lee. Low-sensitivity realization of switched-capacitor filters. *IEEE Transactions on Circuits and Systems*, CAS-34(5):510–522, May 1987.

[31] Roubik Gregorian. Switched-capacitor filter design using cascaded sections. *IEEE Transactions on Circuits and Systems*. CAS-27(6):234–240. June 1980.

[32] Urs W. Brugger, Daniel C. vonGrunigen, and George S. Moschytz. A comprehensive procedure for the design of cascaded switched-capacitor filters. *IEEE Transactions on Circuits and Systems*, CAS-28(8):803–810, August 1981.

[33] Gordon M. Jacobs, David J. Allstot, Robert W. Brodersen, and Paul R. Gray. Design techniques for MOS switched capacitor ladder filters.

*IEEE Transactions on Circuits and Systems*, CAS-25:1014–1021, December 1978.

[34] Man Shek Lee, Garbor C. Temes, Chieh Chang, and Mir Bahram Ghaderi. Bilinear switched-capacitor ladder filters. *IEEE Transactions on Circuits and Systems*, CAS-28(8):811–821, August 1981.

[35] P. E. Fleischer and K. R. Laker. A family of active switched capacitor biquad building blocks. *Bell System Technical Journal*, (58):2235–2269, December 1979.

[36] M. S. Ghausi and K. R. Laker. *Modern Filter Design, Active RC and Switched Capacitor*. Prentice-Hall, 1981.

[37] Richard D. Davis and Timothy N. Trick. Optimum design of low-pass swithced-capacitor ladder filters. *IEEE Transactions on Circuits and Systems*, CAS-27(6):522–527, June 1980.

[38] Ken Martin and Adel Sedra. Exact design of switched-capacitor band-pass filters using coupled-biquad structures. *IEEE Transactions on Circuits and Systems*, CAS-27(6):469–474, June 1980.

[39] R. B. Datar and A. S. Sedra. Exact design of strays-insensitive switched-capacitor ladder filters. *IEEE Transactions on Circuits and Systems*, CAS-30(12):888–897, December 1983.

[40] E. Hokenek and George S. Moschytz. Design of parasitic-insensitive bilinear-transformed admittance-scaled (BITAS) SC ladder filters. *IEEE Transactions of Circuits and Systems*. CAS-30(12):873–887, December 1983.

[41] G. Szentirmai. FILSYN - a general purpose filter synthesis program. *Proceedings of the IEEE*, 65:1443–1458, October 1977.

[42] Roubik Gregorian. Kenneth W. Martin, and Garbor C. Temes. Switched-capacitor circuit design. *Proceedings of the IEEE*, 71(8):941–966. August 1983.

[43] G. V. Eaton, D. G. Nairn, W. M. Snelgrove, and A. S. Sedra. SICOMP: a silicon compiler for switched-capacitor filters. In *IEEE International Symposium on Circuits and Systems*, pages 321–324, 1987.

[44] J. Assel, P. Senn, and M. Tawfik. A switched capacitor filter silicon compiler. In *Symposium on VLSI Circuits*, June 1987.

[45] Edgar Sanchez-Sinecio and Jaime Ramirez-Angulo. AROMA: an area optimized CAD program for cascade SC filter design. *IEEE Transactions on Computer-Aided Design*, CAD-4(3):296–303, July 1985.

[46] L. Trontelj, J. Trontelj, T. Slivnik Jr., R. Sosic, D. Strle, and D. Lucas. Analog silicon compiler for switched capacitor circuits. In *IEEE International Conference on Computer-Aided Design*, pages 506–509, 1987.

[47] Jyn-Bang Shyu, Garbor C. Temes, and Kung Yao. Random errors in MOS capacitors. *IEEE Journal of Solid-State Circuits*, SC-17(6):1070–1075. December 1982.

[48] Kuo-Chiang Hsieh. *Noise Limitations in Switched-Capacitor Filters*. PhD thesis, University of California. Berkeley. May 1982.

[49] T. Pletersek, J. Trontelj, L. Trontelj, I. Jones, and G. Shenton. High performance designs with CMOS analog standard cells. *Journal of Solid-State Circuits*, SC-12(2):215–222, April 1986.

[50] Rainer P. Sigg, August Kaelin, Arnold Muralt, William C. Black Jr., and George S. Moschytz. Fully automated filter synthesizer and mask generator for a CMOS gate-array-type filter chip. In *IEEE International Conference on Computer-Aided Design*, pages 510–513, 1987.

[51] L. W. Nagel. *SPICE2, a computer program to simulate semiconductor circuits.* Internal Memorandum ERL-M520, University of California. Berkeley, May 1975.

[52] E. Polak. *Computational Methods in Optimization.* Academic Press, 1971.

[53] Alberto L. Sangiovanni-Vincentelli and Mauro Santomauro. Optimal printed circuit board ordering via Hart-Nilson-Raphael algorithm. *Circuit Theory and Applications*, 7:247–256. 1979.

[54] I. Cederbaum. Optimal backboard ordering through the shortest path algorithm. *IEEE Transactions on Circuits and Systems*, CAS-21(5), September 1974.

[55] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.

[56] H. Yaghutiel, S. Shen, P. R. Gray, and A. Sangiovanni-Vincentelli. Automatic layout of switched-capacitor filters for custom applications. In *IEEE International Solid-State Circuits Conference*, pages 170–171, 1988.

[57] *Mosis 3 Micron CMOS Design Rules.* August 1985. Revision 2.

[58] C. Laber, R. Rahim, S. Dreyer, G. Uehara, P. Kwok, and P. R. Gray. A high-performance 3u CMOS standard cell library. *IEEE Journal of Solid-State Circuits*, April 1987.

[59] S. C. Fang, Y. P. Tsividis, and O. Wing. SWITCAP - a switched capacitor network analysis program. In *1981 Europian Conference on Circuit Theory and Design*, page 512, 1981.

[60] Nick Weiner. Private communications.

[61] David J. Allstot, Robert W. Brodersen, and Paul R. Gray. An electrically-programmable switched capacitor filter. *IEEE Journal of Solid-State Circuits*, SC-14:1034–1041, December 1979.

[62] Howard M. Sandler and Adel S. Sedra. Programmable switched-capacitor low-pass ladder filters. *IEEE Journal of Solid-State Circuits*, SC-21(6):1109–1119, December 1986.