SWITCHED-CAPACITOR SILICON COMPILER

Mehrdad Negahban-Hagh

Daniel Gajski

Department of Information and Computer Science
University of California, Irvine
Irvine, California 92717

Technical Report No.88-26

ABSTRACT

A switched-capacitor (SC) silicon compiler will be described in this report. The input to this system is a set of specifications and a circuit description of an SC block. It has an opamp synthesis tool which is based on the selection and assembly of primitive modules. The opamp circuit topology is formed by searching through a set of styles yielding high performance opamp circuits serving a wide range of SC network applications. The layout strategy and architecture are such that they eliminate any crossover of sensitive signals by insensitive signals; Thus avoiding any special routing algorithm for sensitive routing. The result is a compact, parasitic-insensitive and "standard cell" type layout of an SC block with user-defined height. All blocks are placed and routed together by a general auto-placement and router tool.

1. INTRODUCTION

In the past few years, design automation techniques have been presented to improve productivity, turn around time, and design quality of analog MOS ICs. Reported DA techniques range from development of layout architectures and design methodologies for auto placement and routing (APAR) of analog standard and parametrized cells, to development of knowledge-based systems [1-4,7-11].

SC design techniques have proven to be an effective approach to several signal-processing applications. Several different linear and nonlinear functions such as filtering, and data conversion (ADC and DAC) can be implemented with precision and stability. SC networks have recieved special attention due to their topological structure's similarity. The SC integrator shown in Figure 1 , is a typical SC block. An SC block is composed of an opamp, one or several capacitor arrays that are either directly connected to the inverting input of the opamp or switched to it, and several MOS switches. An SC network is composed of several SC blocks. Thus an SC block can be regarded as a basic component. Translating behavioral specifications of an SC network to working silicon is a complex process and involves several steps. The main three steps are: First, an SC structure is synthesised. This step finds the value and ratio of capacitors and determines the switching scheme of each block. Determining topology and

capacitor ratio and the switching scheme of a multi-order filter is in this category. Secondly, opamp (or comparator), MOS switches, and capacitor arrays are defined and sized. This synthesis step is at the transistor level. Thirdly, layout architecture is chosen and basic layout components are connected according to the chosen architecture.

Several tools have been developed to assist designers of SC networks, while others completely automate different steps of design and layout. Naira et. al. [12] reported a program to help the synthesis and topological formation of SC filters which aids the first step of the design process. One of the earliest reported approaches for the design and layout of SC networks was standard cell with APAR tools [3,9]. Several different switched capacitor silicon compilers have been reported [11]. They perform different tasks in the design process and are based on a standard cell library for a fixed technology. A methodology for technology-independent layout of SC filters has been reported in [10]. It has three arrays of op amps, capacitors, switches and two routing channels. It uses an algorithm that takes multiple iterations to adjust the height of capacitor arrays so that a square block of a multi-order filter is generated. Then the channels are routed so that overcrossing of sensitive signals and insensitive signals is minimized.

Since standard cells have a fixed topology, device sizes and bias current, they can not provide the flexibility and performance for a wide range of SC networks. Another disadvantage of standard cell libraries is their technology dependency. Any time the technology parameters change a new set of cells must be developed.

Our focus has been on the development of design automation tools for design and the silicon implementation of linear and nonlinear SC networks and its fundamental elements, such as opamps, comparators and precision capacitor arrays. The main objective of this research is to enable a system designer with little or no IC design experience to design and implement an SC network. In this paper, we describe a system that accepts a set of specifications and a schematic of an SC block and generates a compact layout of that block in a predefined CMOS technology.

2. OVERVIEW OF THE SYSTEM

Figure 2 shows the general block diagram of the system. The system is composed of three different tools:

(1) **CAST** , a circuit synthesis tool. CAST defines opamp design requirements, and synthesises an opamp. Also it sizes MOS switches.

(2) **ALEX** , a layout tool for analog module generation and placement and sensitive signal routing. The major distinction of ALEX layout strategy

from previously reported work [10] is in its special placement of primitive modules and layout architecture to completely eliminate overcrossing of sensitive signals by insensitive signals.

(3) An **SC**network generator using standard auto routing and placement tools.

This system is technology-independent. The requirements are MOS device model file describing process parameters, and a technology file including process layout rules. A circuit simulation tool such as SPICE is required to evaluate circuits synthensized by CAST.

Input to this system has two parts:

(1) a set of SC block **specifications** which include: clock frequency , maximum signal amplitude, bandwidth of the input signal, maximum expected error due to the op amp nonidealities (finite gain and bandwidth), and a set of power and area **constraints.**

(2) Circuit-description file specifying structure and topological interconnection of each element in an SC block.

Figure 3 shows an equivalent schematic of the SC integrator shown in Figure 1 . Four elements are shown: an opamp, input switch (ISW), output switch (OSW), and capacitor array. Table 1 shows a typical input description of this circuit. This file is a user generated file or is generated by a higher level synthesis tool. It describes the type, nodal connection and size of each element.

The result is a compact, parasitic insensitive layout of a "standard cell" type SC block. This layout will be placed and routed with a general purpose APAR tool, together with other SC blocks.

3. STRUCTURAL SYNTHESIS

The structural sythesis of SC blocks consists of two tasks:

(1) sizing input/output MOS switches, and

(2) designing an opamp based on given SC block specifications.

Our approach follows a human designer strategy in combination with computer aided problem solving techniques which use heuristics and algorithmic procedures.

Typical steps taken by a designer are as follows:

(1) A proper topology is formed to best fit the application and requirements. This is based on judgement, and search through different design styles.

(2) Using simple first-order models, the device sizes and a biasing scheme are found. If the result of this first order analysis-synthesis process is not satisfactory, then either the topology or the specifications should be modified.

(3) Next, the result is verified using a simulation tool such as SPICE.

(4) If the simulation result is not acceptable, the design is modified, possibly resulting in the redefinition of the topology, size, and/or bias current.

(5) Once the requirements are satisfied, the design is optimized further with respect to power, area, and other performance parameters.

Among the five steps mentioned, the first four of them have been implemented through our system, CAST. The system requires both a precision device model to be used in simulation and a set of first order worst case device models to be used in step 2. Figure 4, provides an overview of each step of structural synthesis.

From the requirements and constraints set on the SC block, the following opamp parameters are generated:

1) DC open loop gain, A_0

2) Unity gain bandwidth, f_u

3) phase margin, ϕM

4) slew rate, SR

5) capacitive load, C_L

6) input, output signal range

MOS switches have finite **ON** resistance. The RC time constant caused by the

switch on resistance puts further burden on speed requirements of the SC block. Knowing the value of C, a worst case value of R can be calculated. Then, through a table lookup which relates the ON resistance value to the size of the MOS devices, the sizes of input and output switches are estimated.

The next step is to design an opamp. The following distinct tasks are performed. First, a proper circuit topology is selected. Second, MOS devices are sized, and the DC biasing scheme of the opamp is defined. A third task is to minimize power and area if possible and/or necessary. The necessity of minimization is determined by power/area constraints defined by the user.

The result is an initial solution (an opamp circuit) which meets all or part of the requirements. This design is based on first order models and equations and simple heuristics.

A schematic and layout are generated using ALEX. At this point a SPICE input file can be created which will include all parasitics of the actual layout. SPICE is asked to perform simulations so that gain and speed requirements of the opamp can be verified. If the SPICE simulation results are not satisfactory, they are fed back to the system where the cause of the problem is detected first.

Then, two approaches may be taken to correct the problem:

(1) modify device sizes or the biasing scheme as long as it does not disturb power and area constraints, otherwise

(2) modify the circuit topology and start from the beginning.

The final result could meet all or part of the specifications. If it is possible to resolve any of the problems by relaxing power and area constraints, the system will make suggestions to do so.

The tasks of style selection, device sizing, DC biasing, and minimization of power and area are totally interrelated. They influence each other during every step of the design process for an opamp. However, for the sake of argument, these steps are discussed as separate issues.

There are major common characteristics between opamps in an array of SC networks. They all drive capacitive loads, and since all SC blocks are generally clocked with the same frequency, all opamps have the same speed requirements. Thus, all opamps for most practical cases can share the same topology and differ only in size and dc biasing scheme. Use of the same opamp topology makes the layout of a big array of SC blocks an easier task.

SC networks serve a wide range of applications, requiring high performance opamps. To fulfill this requirement, the system provides flexibility of **forming** different opamp circuit topologies from primitive opamp elements. By searching through a style selection tree, primitive opamp modules that satisfy given requirements are selected. An opamp circuit topology is formed by assembling these modules.

A typical gain stage is comprised of two elements. An input device provides transconductance (Gm), while a load device, in conjunction with the input device, provides output impedance (R_0). Figure 5 provides an insight into currently available modules for input and load devices. From these primitive modules, a wide range of opamp circuits can be formed for high gain, high speed, and low power applications.

The decision making process is heuristic, while the process of evaluating each primitive style is algorithmic. The search through the style selection tree starts from a simple module (simple in terms of number of elements). The search continues by selecting more complex elements as requirements become more difficult.

Gain and speed requirements of the opamp are the main factors influencing style selection. First, devices sizes and dc bias current are calculated. Then, the result is evaluated against given constraints. If the result is not satisfactory, a new style is selected. For example, suppose a high output impedance circuit is required. A basic load module may not provide the desired performance within the given area and power constraints. Thus a cascode load device which provides a higher output impedance is selected.

Input common mode range and output swing are also influential on the style selection process. A change in power and area constraints as well as gain

and speed requirements can result in different circuit topologies.

First order MOS models and equations are used at this stage. Area and power constraints provide a bracket for fast convergence to a result. The process of sizing devices and designing the dc biasing scheme is best explained through the simple chart shown in Figure 6. Speed, gain, signal range and area/power constraints are inputs to the system, and a topology with known width (W) and length (L) is the output. Intermediate parameters such as G_M, R_0, I (current), Z (W over L ratio) and C_C (required capacitance for frequency stabilization) are used as bridging input attributes to output variables.

This chart serves two purposes. First, it interrelates all different attributes involved in our system. Second, it shows the flow of design steps, starting from speed requirements (f_u, ϕM, SR) which yields variables G_M, I and C_C, to the bottom of the chart where W and L of each device are calculated. Note that the signal range is a function of MOS device threshhold and drain to source voltage which is not shown on this chart.

This step provides an initial guess which must be further evaluated.

As was mentioned, SPICE and a complex MOS device model (including second order effects) are used to evaluate the results. SPICE can provide more accurate information about A_0, f_u, SR, ϕM and signal range of an opamp over process, supply voltage and temperature variation. If the result is not

satisfactory, the cause of the problem is detected by providing values of opamp parameters to the system.

SPICE simulation is the most time consuming portion of the design process. There are two major problems associated with SPICE analysis:

(1) SPICE dc and transient convergences, and

(2) many iterations may be required to obtain an acceptable design. The discrepancy is due to differences in first order models and equations used in design calculations as opposed to the second order complex model used in SPICE simulations.

To avoid the first problem, the input file contains SPICE parameters known to help the convergence problem. Initial nodal voltage guesses provided by our system are also used to avoid the convergence problem. These precautionary actions have proven to help the convergence problem.

To mimimize the number of SPICE iterations, the following two actions have been taken:

(1) The system uses first order models which are a worst case representation of SPICE models under power supply, process and temperature variation. Together with safe design practices secure meeting the most of design specifications. The above approximation may result in gross overdesign which need to be further optimized in some areas.

(2) SPICE simulation is a "root finding" process. Given K, a specification parameter one should find x0, so that F(x0)=K. Note that F(x) is unknown and every SPICE iteration only provides one point on F(x). CAST provides a 1st order approximation of F(x). Proper choice of a "root finding" algorithm speeds up convergence.

Secant and Newton-Raphson methods are two well known root finding algorithms. To take advantage of these models for a fast convergence, F(x) and its derivatives (in the case of Newton-Raphson) must be known. Neither are known. Our approach takes advantage of the fact that CAST has some 1st order knowledge about the behavior of SPICE simulation results (F(x)). After each iteration, CAST modifies its knowledge using Lagrange equations [13] to better approximate F(x). This process of "learning and adaptation" is similar to Brent's method of root finding which is argued to be the best algorithm where only points of a function and not the function itself are available [14].

4. PHYSICAL IMPLEMENTATION

ALEX is the automatic layout tool. It is based on Silicon Compiler Systems' Generator Design Tools (GDT). There are two major distinctions between the layout of analog ICs and their digital counterparts:

(1) precision component matching, and

(2) avoiding crossover of sensitive signals by insensitive (undesired) signals.

Matching of components is a major task whether it is transistors inside an opamp or capacitor arrays of an analog-digital converter. The matching requires the same geometric orientation, close proximity and symmetry with respect to each other. Mismatches generally result in unwanted errors that reduce the performance of an analog circuit. We have addressed this problem by using primitive module generators for any two or more elements that need to be matched. Module generators have been used extensively before and have been shown to be an adequate replacement for human layout designer skills with respect to precision layout [3]. Some typical modules are shown in figure 7. Each module in CAST has a corresponding module in ALEX. CAST generates a layout file which is a selection of the required layout modules and their interconnections.

All sensitive signals are routed inside the block. Insensitive signals are brought out to the routing channel. Figure 3 can provide a better insight to understand our layout methodology. It shows that there are four typical components at the SC block level.

(1) The op amp which is laid out and compacted as a separate cell. Generally each op amp has six terminals: two inputs, one output, two power supply

connections, and one dc biasing input. But in the schematic, only one input and one output are shown. All other signals are considered to be global. The noninverting input of an op amp used in a SC block must be forced to a constant voltage source for proper operation of the SC network. CAST assumes the same bias network for all op amps, so that the bias input is also global. Inverting input node "inm"of the op amp is a high impedance node. This node and any other node switching to it are considered to be sensitive and are laid out inside the block.

(2) ISW, are MOS input switches. ISWs are connected to "inm" and switch capacitors to "inm". They are single NMOS switches, since node "inm" is midway between the two supplies.

(3) OSW, are MOS output switches. They are connected to output nodes. OSWs are CMOS switches (transmission gates) since they switch to output signals that could move to near the supplies.

(4) Capacitor arrays. Precision double poly capacitor arrays can be generated containing both unit and fractional unit capacitors with the same area to perimeter ratio. A fixed area to perimeter ratio is a key factor in matching of unit square capacitors to unit fractional rectangular [6].

Table 1 shows an input-description file for the circuit shown in Figure 1. Nodes "inm", and "n2" are considered to be sensitive. These two nodes and

node "n1" which connects the OSWs to capacitors are routed inside the block. All other nodes are brought out to the routing channel.

Figure 8 shows a typical placement of an SC block's elements. As can be seen there exists a sensitive routing channel inside the block, where node "inm" and all other sensitive signals get routed. This layout architecture guarantees complete elimination of overcrossing sensitive signals by insensitive signals and avoids extra steps or complex routing algorithms required to minimize overcrossing. A similar approach is taken in the layout of op amps, where the primitive modules are placed in a special manner to avoid any overcrossing of sensitive signals.

A compaction step takes place without disturbing the sensitive routing and the modules placement. It primarly uses capacitor arrays to fill empty gaps and compact the layout. After the placement of the modules and sensitive signal routing, power lines are placed at the top and the bottom of the block. The end result is a compact "standard cell" type layout of a SC block. Finally all blocks are placed and routed with a general purpose APAR tool. A typical placement of SC blocks is shown in Figure 9 . This method places blocks together so that they can share one of the power lines, or the routing channel. Placing the capacitor arrays of the two blocks together could result in further compaction.

As mentioned above, all opamps used in a SC network drive capacitive loads and generally have the same speed requirements. Therefore, they can use the same topology, but with different sizes or bias currents. CAST tries to select one opamp technology for an array of SC blocks and modifies the device sizes for each block. In the ALEX layout architecture, the height of all op amps stay the same while the opamp widths vary for different device sizes. The height of the op amp defines the height of the SC block. Each block grows only in width, maintaining a fixed height. The height of the SC block and op amp is a user-defined parameter. If the height is not defined, ALEX selects the most optimum height with respect to the area.

5. EXPERIMENT

A fourth-order bandpass filter from a ladder prototype has been designed and implemented in a 3um CMOS double-poly, single-metal process. The specification of this SC network is listed in Table 2. Note that the capacitive loading may differ for each individual op amp. CAST optimized and designed four different opamps. The performance comparision are shown in Table 3. The area penalty was more than 23 percent. The design and layout process using this compiler took less than a day, compared to a typical two weeks turn-around time. The layout is shown in Figure 10.

6. CONCLUSION

We have presented a technology independent Switched-Capacitor Silicon-Compiler. It performs all required structural and physical tasks of implementing an SC network. High performance opamps for a wide range of SC network applications can be dsigned using the methodology described. The layout architecture eliminates the need for any special auto-routing tool for sensitive lines and results in high quality layout of SC networks. Our example shows a 23 percent area penalty from a hand-crafted version and can be further reduced by improving the compaction algorithm. This system will be part of a general SC compiler for design and implementation of filters and other linear, and nonlinear sampled data circuits.

7. REFERENCES:

[1] P.E. Allen "A Design Methodology for CMOS Analog Integratde Circuits", Proc. IEEE ICCAD, Nov. 1983.

[2] P.E. Allen, et al, "AIDE2: AN Automated Analog IC Design System", Proc. IEEE CICC, May 1985.

[3] C.D. Kimble, et al, "Autorouted Analog VLSI" Proc. IEEE CICC, May 1985.

[4] W.J. Helms, et al, "Switched Capacitor Filter Compiler", Proc. IEEE CICC, May 1986.

[5] P.R. Gray and R.G. Meyer, " MOS Operational Amplifier Design-A Tutorial Overview" IEEE Journal of Solid-State Circuits, No. 6, Dec. 1982.

[6] R. Gregorian, G.C. Temes, Analog MOS Integrated Circuits for Signal Processing, John Wiley and Sons, 1986.

[7] R. Harjani, R.A. Rutenbar and L.R. Carley, "A Prototype Framework for Knowledge-Based Analog Circuit Synthesis", Proc 24th ACM/IEEE DAC., June 1987.

[8] H.Y. Koh, C.H. Sequin and P.R. Gray, "Automatic Synthesis of

Operational Amplifiers Based on Analytic Circuit Modules", Proc. IEEE ICCAD, Nov. 1987.

[9] D. Maeding, M. Negahban, and B. Klein, "Combining Analog and Digital Using Standard Cells", Proc. IEEE CICC , May 1985.

[10] H. Yaghutiel, A. Sangiovanni-Vincentelli and P.R. Gray, "A Methodology for Automated Layout of Switched Capacitor Filters" Proc. of ICCAD, Nov. 1986.

[11] J. Assael, et al, "A Switched-Capacitor Filter Silicon Compiler", IEEE Journal of Solid-State Circuits, No.1, Feb. 1988.

[12] D. Nairn and A.S. Sedra, "Auto-SC, an Automated Switched-Capacitor Ladder Filter Design Program", IEEE Circuits and Devices Magazine, March 1988.

[13] J. Stoer, and R. Bulirsch, Introduction to Numerical Analysis, New York, Springer-Verlag, 1980.

[14] R.P. Brent, Algorithms for Minimization without Derivatives, Englewood Cliffs, N.J. Prentice-Hall, 1973.

TABLE 1

Typical SC Block Description Input File

```
type    name    node1   node2           size (optional for switches)
-----   -----   -----   -----           ----------------------------
opa     opa1    inm     output;
osw     osw1    input   n1      p1      w=10            l=3;
isw     isw1    inm     n2      p1      w=10            l=3;
cap     c0      n1      n2              value=1.0       usize=25.0;
cap     cf      inm     output          value=23.78     usize=25.0;
```

TABLE 2

SC Block Specifications

parameter	value	comment
fc	100.0	clock frequency, in KHz.
cl	10.0	capacitive loading, in PF.
et	0.1	total error, in percent.
v0	5.0	peak amplitude of signal, in Volt.
f0	5.0	input signal bandwidth, in KHz.
imax	1.0	maximum current consumption, in mAmp.
zmax	200.0	maximum W/L of MOS devices.

TABLE 3

OP AMP Specifications

parameter	required value,	SC silicon compiler	units
DC Gain	66	71	dB
Unity Gain Bandwidth	2	3	Meg Hz
Phase Margin	70	70	Degree
Slew Rate	3	4	Volt/USec
Current Consumption	1000	110	UAmp
Total Area	1230	1590	sq. mils

FIGURES:

Figure 1. An SC Integrater

Figure 2. System Overview

Figure 3. Functional Schematic, Equivalent to Figure 1

Figure 4. CAST Flow-Chart

Figure 5. Decision-Decomposition (Style Selection) Tree

Figure 6. Design Flow-Chart

Figure 7. Typical Layout Module Generated by ALEX

Figure 8. Floorplan, Placement and Routing of an SC Block

Figure 9. SC Block Placement and Floorplanning

Figure 10. Layout of a 4th Order Bandpass Filter

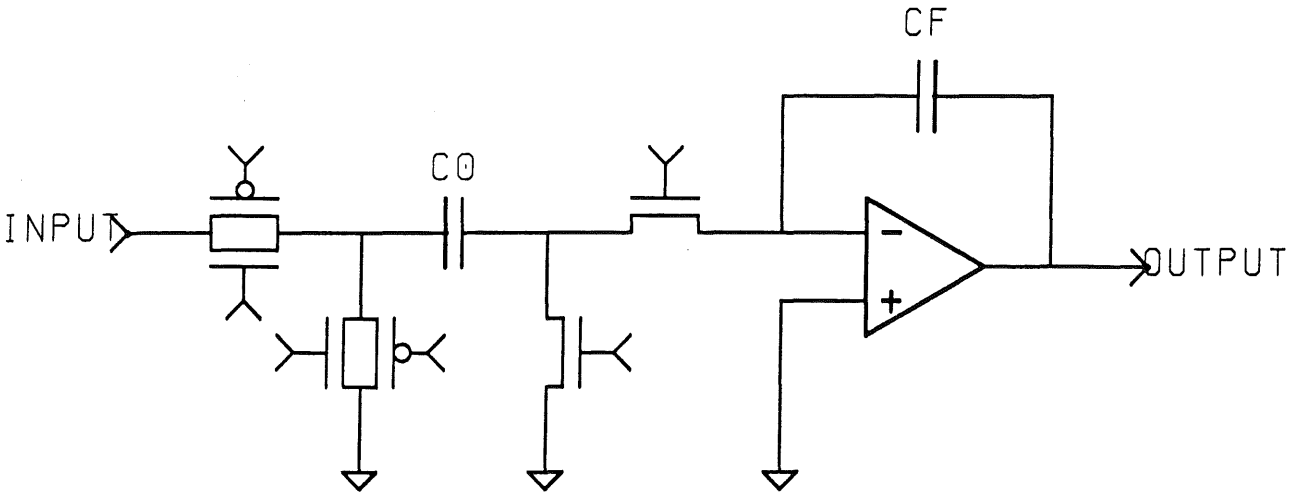


FIGURE 1

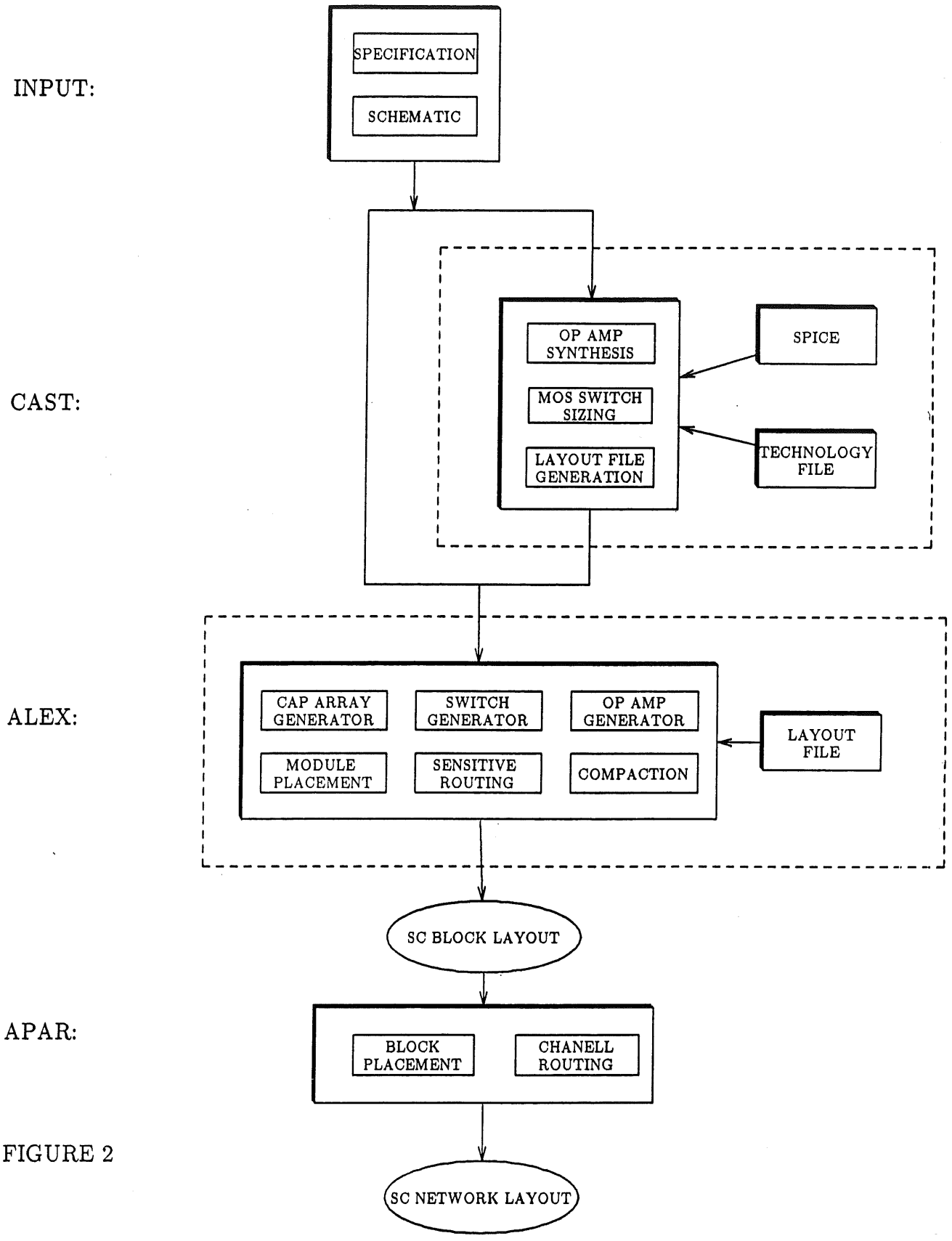

INPUT:
SPECIFICATION
SCHEMATIC
CAST:
OP AMP SYNTHESIS
MOS SWITCH SIZING
LAYOUT FILE GENERATION
SPICE
TECHNOLOGY FILE
ALEX:
CAP ARRAY GENERATOR
SWITCH GENERATOR
OP AMP GENERATOR
MODULE PLACEMENT
SENSITIVE ROUTING
COMPACTION
LAYOUT FILE
SC BLOCK LAYOUT
APAR:
BLOCK PLACEMENT
CHANELL ROUTING
SC NETWORK LAYOUT

FIGURE 2

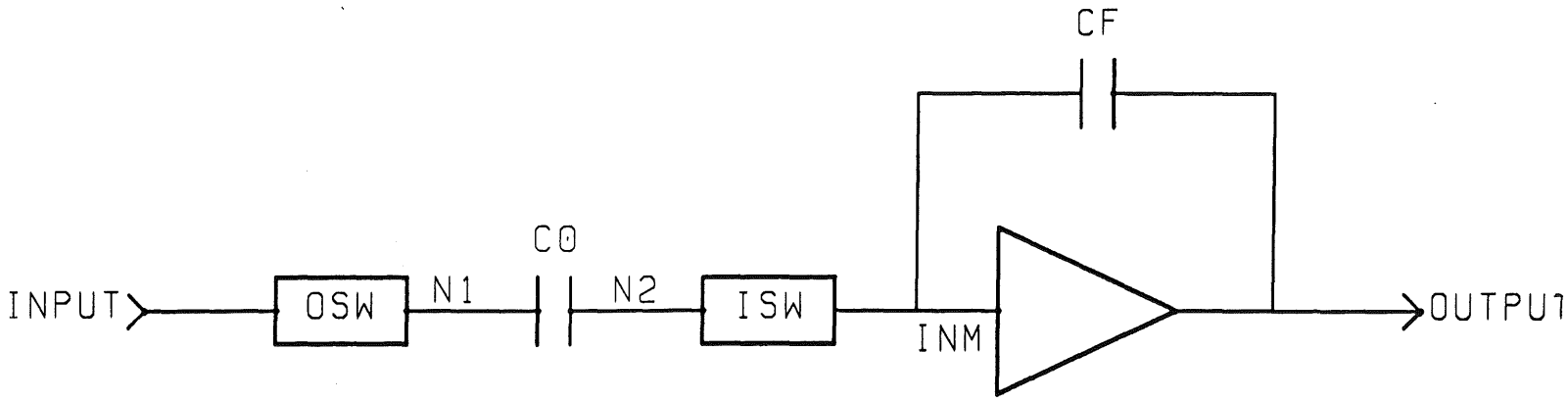


FIGURE 3

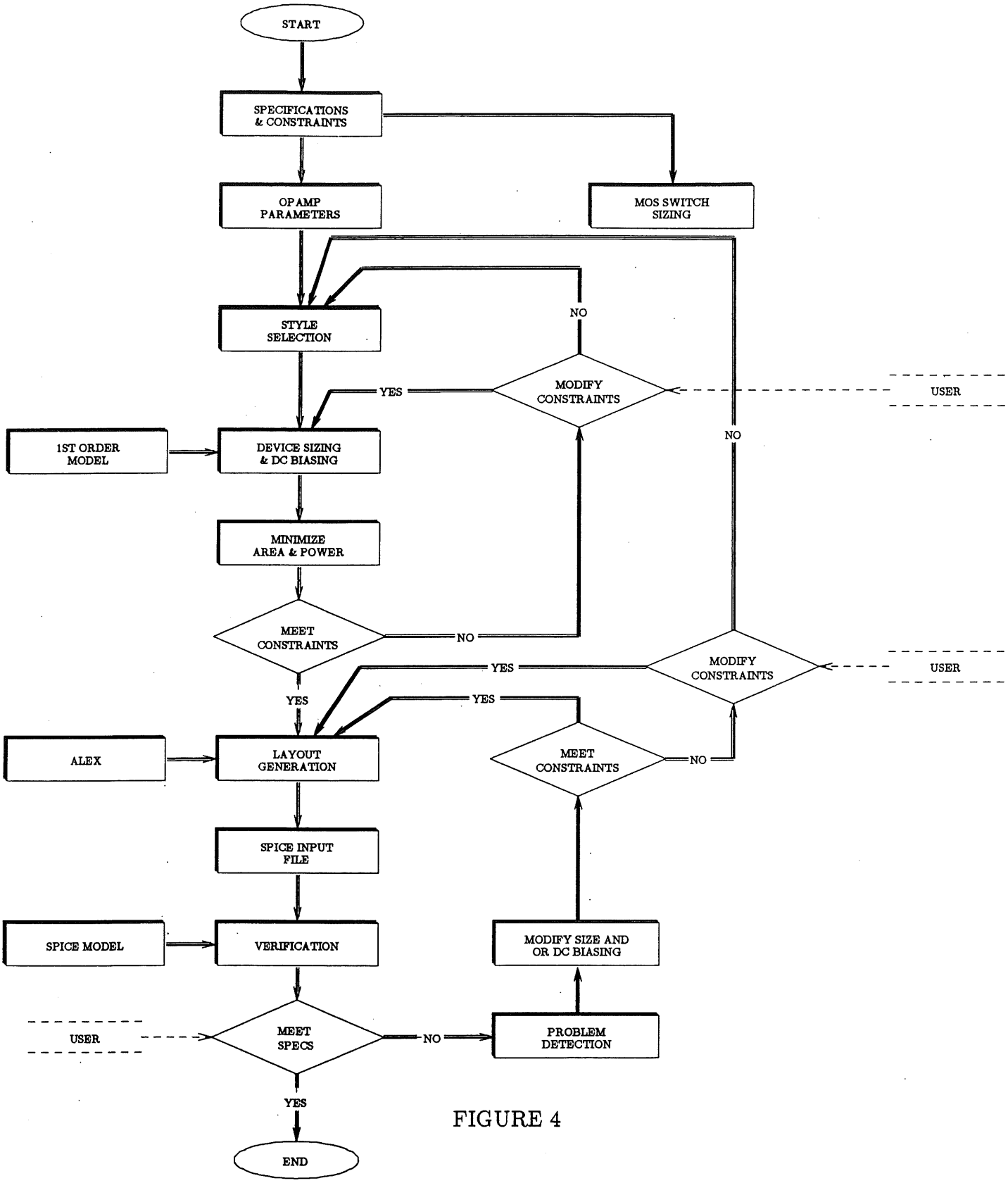

START
SPECIFICATIONS & CONSTRAINTS
OPAMP PARAMETERS
MOS SWITCH SIZING
STYLE SELECTION
NO
MODIFY CONSTRAINTS
USER
YES
1ST ORDER MODEL
DEVICE SIZING & DC BIASING
NO
MINIMIZE AREA & POWER
MEET CONSTRAINTS
NO
MODIFY CONSTRAINTS
USER
YES
YES
YES
ALEX
LAYOUT GENERATION
MEET CONSTRAINTS
NO
SPICE INPUT FILE
SPICE MODEL
VERIFICATION
MODIFY SIZE AND OR DC BIASING
USER
MEET SPECS
NO
PROBLEM DETECTION
YES
END

FIGURE 4

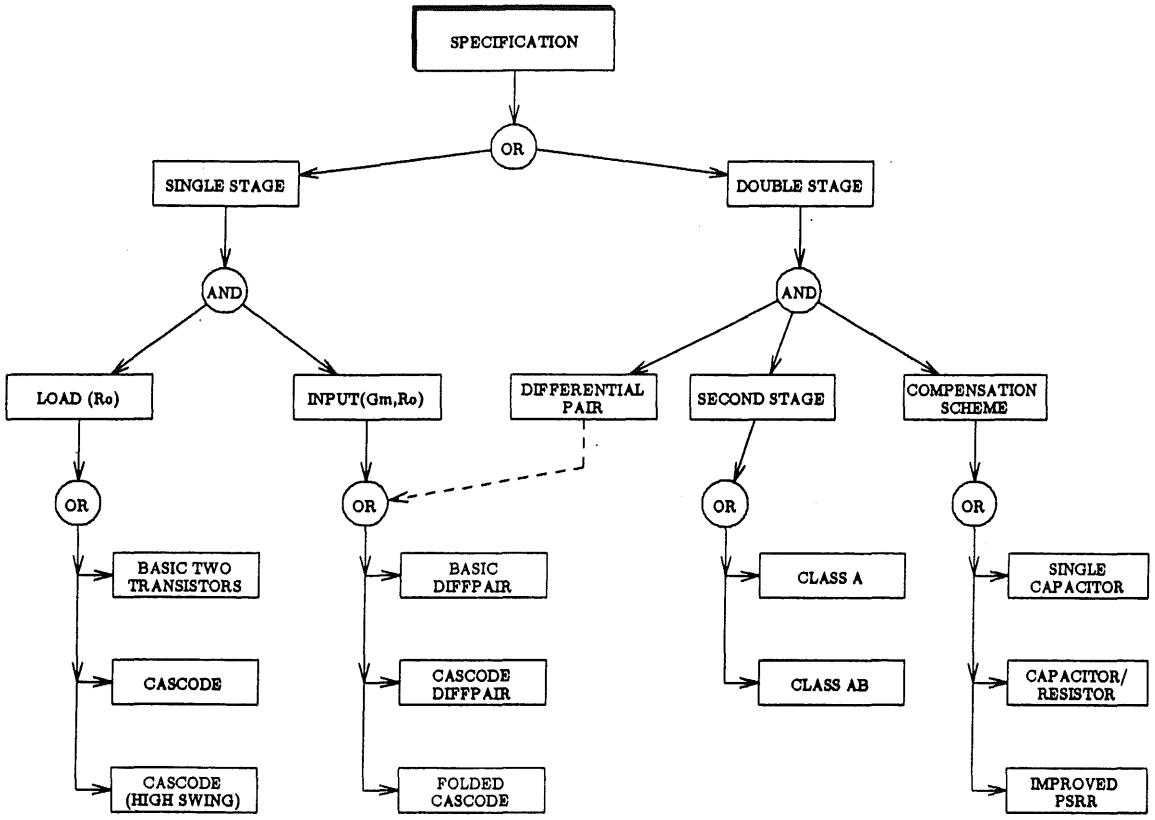

SPECIFICATION
OR
SINGLE STAGE
DOUBLE STAGE
AND
AND
LOAD (Ro)
INPUT(Gm,Ro)
DIFFERENTIAL PAIR
SECOND STAGE
COMPENSATION SCHEME
OR
OR
OR
OR
BASIC TWO TRANSISTORS
CASCODE
CASCODE (HIGH SWING)
BASIC DIFFPAIR
CASCODE DIFFPAIR
FOLDED CASCODE
CLASS A
CLASS AB
SINGLE CAPACITOR
CAPACITOR/ RESISTOR
IMPROVED PSRR

FIGURE 5

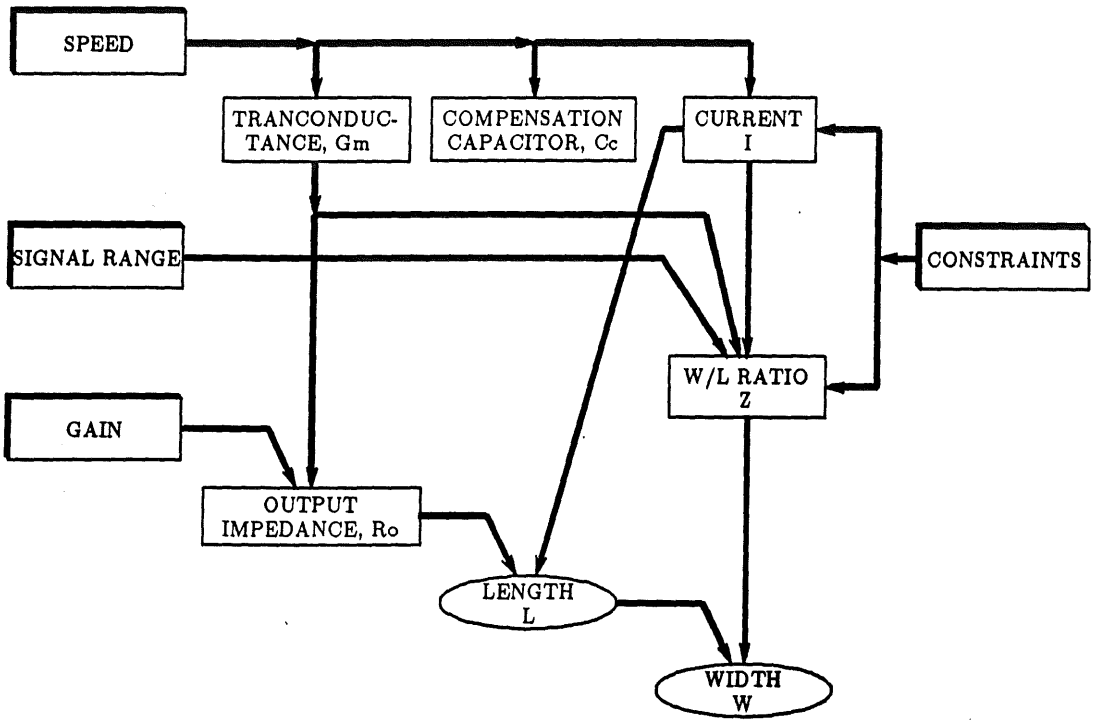
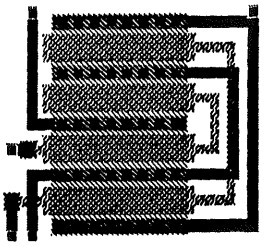
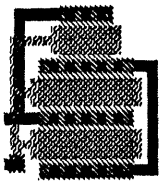

SPEED
TRANCONDUC-
TANCE, Gm
COMPENSATION
CAPACITOR, Cc
CURRENT
I
SIGNAL RANGE
CONSTRAINTS
W/L RATIO
Z
GAIN
OUTPUT
IMPEDANCE, Ro
LENGTH
L
WIDTH
W

FIGURE 6

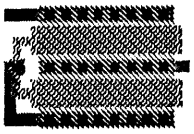
differential pair



sliced transistor with variable height



current source



cascode device

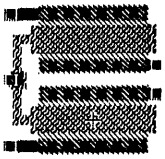


FIGURE 7

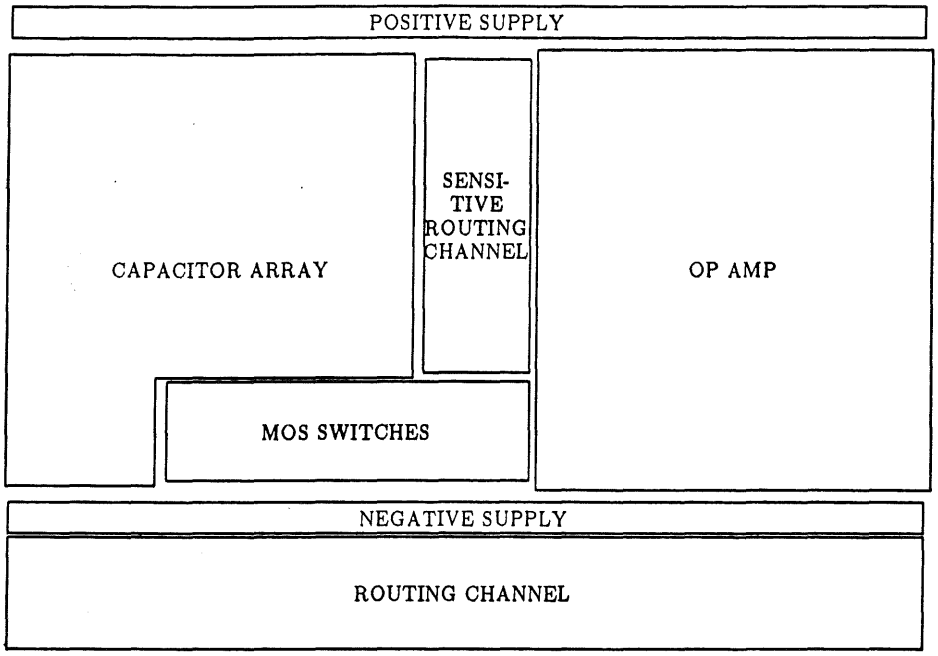

POSITIVE SUPPLY
CAPACITOR ARRAY
SENSI-
TIVE
ROUTING
CHANNEL
OP AMP
MOS SWITCHES
NEGATIVE SUPPLY
ROUTING CHANNEL

FIGURE 8

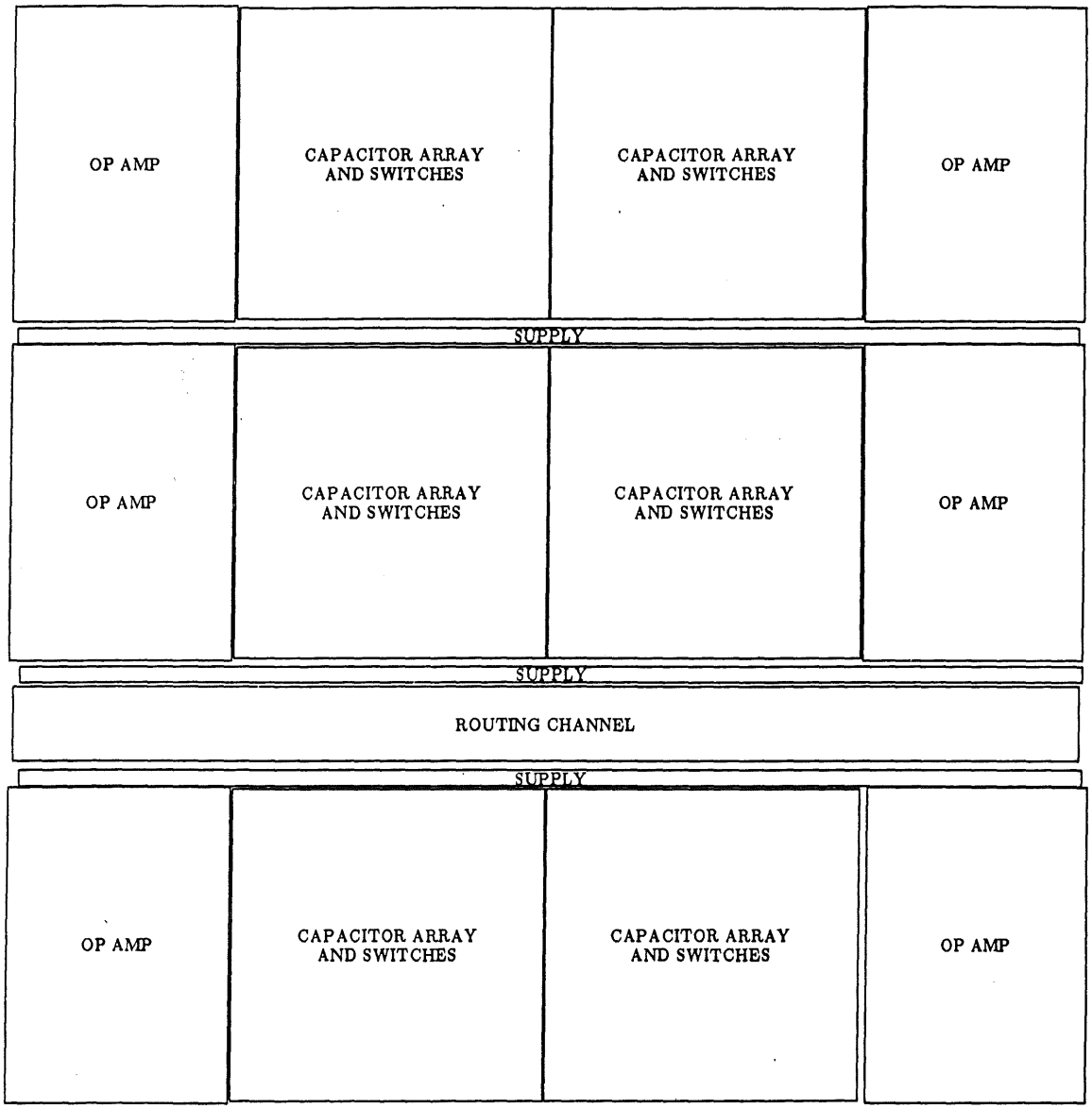

OP AMP
CAPACITOR ARRAY AND SWITCHES
CAPACITOR ARRAY AND SWITCHES
OP AMP
SUPPLY
OP AMP
CAPACITOR ARRAY AND SWITCHES
CAPACITOR ARRAY AND SWITCHES
OP AMP
SUPPLY
ROUTING CHANNEL
SUPPLY
OP AMP
CAPACITOR ARRAY AND SWITCHES
CAPACITOR ARRAY AND SWITCHES
OP AMP

FIGURE 9

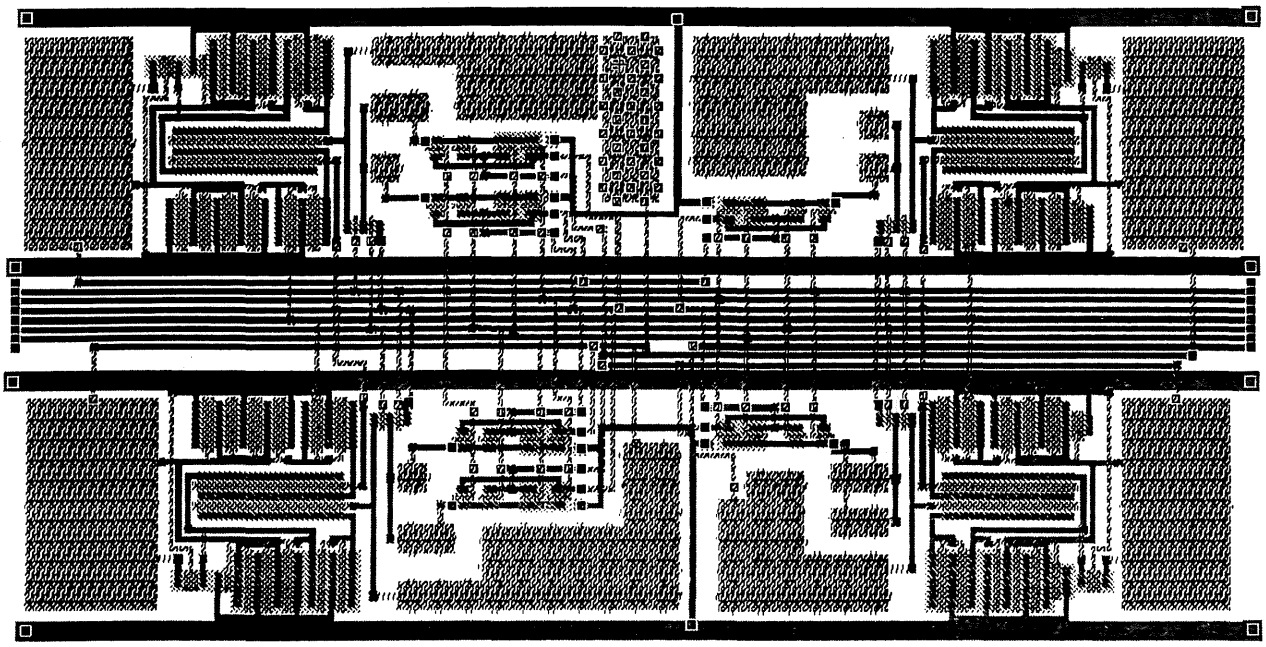


FIGURE 10