Schematic Driven Layout Using the MOSIS Design Kit for Advanced Analog and Mixed-Signal Design

Mircea Stan, David Garrett, Abhimanyu Kolla, Zaid Salman[1]
mircea@virginia.edu, http://watt.seas.Virginia.EDU/~mrs8n/
University of Virginia - Department of Electrical Engineering, Charlottesville, VA 29903

1. currently with Maxim Integrated Products, Inc.

Abstract

There is renewed interest in the field of analog and mixed-signal circuits, especially among designers of high-speed and wireless communication systems. Due to the typical custom design approach, a large amount of time is spent laying out large (matched) devices which greatly increases the probability of human error. Schematic-driven layout (SDL) has proven to be a powerful tool for assisting custom digital design in Universities and there is a need for similar SDL capabilities for analog and mixed-signal design.

We are developing analog SDL extensions to the MOSIS Design Kit (MDK) with device generators for inter-digitated transistors, resistors, capacitors and inductors, and for matched pairs of inter-digitated transistors, resistors and capacitors in common centroid configurations.

1. Introduction

A "system-on-a-chip" tries to incorporate digital logic, processors and memory together with analog and mixed-signal blocks on the same IC. This concept is made necessary by the fast advances in IC manufacturing which enable millions of transistors to be economically packed on a single die and make the use of discrete parts (still common in analog design) very undesirable. Furthermore, the push for low-power and miniaturization for mobile communication also require the integration of analog functions together with digital blocks.

Digital and analog VLSI design methodologies are very different though. With the exception of some high-volume or high-performance circuits (e.g. microprocessors, memories, etc.), most of the current digital ICs are semicustom designs using standard cells or gate arrays. The same is not true about analog circuits which typically still need to be custom designed in order to achieve desired characteristics. Although there have been efforts to automate analog VLSI design with CAD tools [1,2,4], their acceptance by analog designers is minimal. One reason for the mistrust in analog CAD tools [5] is that their results are typically much worse in terms of performance than the ones manually designed. Another reason is that due to a relatively low complexity (several tens or hundreds of transistors typically) and high volume, manual design of "standard" analog ICs (e.g. opamps) is still economically sound. This situation is changed though when both digital and analog blocks need to be integrated on the same IC, and the two need to use compatible design methodologies. Figure 1 shows the schematic of a CMOS opamp like the ones we want to automate with various transistor sizes and transistor configurations (differential amplifier, current mirror, etc.) which require precise matching and reduced parasitics.

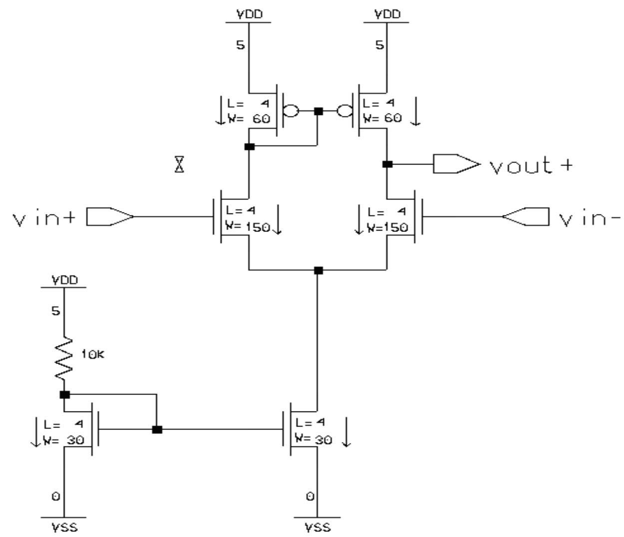

Fig. 1 CMOS operational amplifier

1.1 Custom vs. automatic analog layout

The design of digital blocks in ASICs can be fully automated with excellent results by using a standard-cell or gate array approach. It is very unlikely that such methods will ever work for general analog circuits since transistor sizes, placement and routing tend to be more varied and critical in this case. It is hard to imagine the equivalent of an "analog" standard cell library that would allow arbitrary circuits to be build from such building blocks (there are attempts to build analog "field programmable devices", with mixed results though). On the other hand it is perfectly reasonable to have typical analog cells predesigned as "macroblocks" and only design new circuits when necessary, but this is not the focus of our work.

1.2 Device generation, placement and routing

Some researchers [2] tried automating the entire analog design process with relative success by using device generators followed by automatic placement (e.g. simulated annealing) and routing (e.g. Lee's maze algorithm).

We believe that the best approach for (partly) automating analog design is to provide the designer with powerful device generators but still do the placement and routing manually. In this way the designer is relieved of the labor intensive and error prone task of laying out devices manually, but still has control over the critical placement and routing.

1.3 Common-centroid configurations

Digital circuits typically use minimum-size devices for high speed, low area and low power. This is not true for analog circuits which sometimes need transistors with large "width" and "length" in order to increase gain and reduce 1/f noise (see figure 1). Such large devices are more likely to be affected by process variations across the IC, this situation being critical for matched components.

One method to reduce the effect of process variations is to use interdigitated layouts. The idea is to partition the large devices into smaller pieces and then physically alternate the small pieces such that their relative distance is reduced.

Example: Two large transistors that need to be matched are made out of four smaller pieces, **`AAAA`** and **`BBBB`**, with each letter representing one unit. If the two are placed adjacent but grouped:

`AAAABBBB`

then the distance between their "centers" will be 4 units, hence a large possible variation, but if the two are interlaced:

`ABABABAB`

the relative distance between the "centers" is now only 1 unit, for a four times reduction in process variation. The variation can be completely compensated for linear gradients by using so-called "common-centroid" configurations in which the pieces are arranged such that the two "centers" (or centroids) coincide. Interestingly, there are several way in which this can be done, for our example:

`ABBAABBA` or **`ABABBABA`**

There are many considerations about the merits of various common-centroid arrangements [4,5], for example the second arrangement from above is considered better than the first because it has only one situation where two same letters are adjacent. For a "two-dimensional" arrangement with several rows and columns a possible configuration (the one we use) is:

`AABBAABB`
`BBAABBAA`

Besides process variation, matching can be affected by random effects which are hard to control by the designer, and by possible systematic differences, for example in the connecting wires. Such differences in wire parasitics can be made even worse by interdigitated layouts which increase the amount of local routing. There are ways to route interdigitated layouts with identical wiring parasitics [1], unfortunately common-centroid configurations inherently require a slightly unbalanced routing and identical routing patterns require slightly unbalanced device layouts.

1.4 Analog vs. digital SDL

Schematic-driven layout (SDL) is a design method which uses schematic capture for design entry, followed by layout in which components are instantiated from the schematic in a "correct by construction" or "connectivity driven" fashion.

The schematic can be either at the logic level, with predesigned cell layouts from a library being typically instantiated, or at the transistor level, when procedural device generators are used to generate the layouts in a parameterized way for various devices. After placement of the instantiated cell or generated device, routing is performed either manually or with an autorouter. Using SDL, the designer can achieve results that are very close to full custom manual layout but in a fraction of the time and with reduced effort since the device generators take care of the error-prone phases.

As mentioned before, analog SDL makes more sense at the transistor level with parameterized device generators and this is the topic of our approach. Compared with a "digital" transistor-level SDL [6], the differences are that more complex device generators are needed that can match layouts and wires, as well as generators for other active (diodes, bipolar transistors) or passive (resistors, capacitors, inductors) devices.

2. Active Devices

Our purpose is to enable analog and mixed-signal design in a standard CMOS process like the ORBIT 2um or the AMI 1.2um processes offered through MOSIS. Because of this, our main focus is on MOS transistors, while bipolar transistors and diodes which are available in some CMOS or BiCMOS processes remain for now the subject of future work. Also the subject of future work is the automatic generation of guard rings (which can still be added manually) and the independently biased wells found in some analog circuits.

There are two possible approaches in device generation for analog design:

- the first one only generates "basic" devices and then expects the placement to find opportunities for region (e.g. diffusion) sharing and matching [2]. This approach only works if placement is fully automated, and even in that case it is impossible to obtain interdigitated matched layouts as we want.
- the second has generators for devices and for commonly used groups of devices [4]. This approach gives excellent results for the types of generated device groups but increases the number and complexity of the required generators. It works equally well with automatic and manual placement and for this reason it is the approach we take.

2.1 Interdigitated MOS transistors

There are many ways to lay out large transistors and transistor pairs, even for common centroid configurations. Our goal is to use as few configurations as possible but still be able to support most of the possible cases. The building block for all configurations is the interdigitated (folded) transistor. Such a folded transistor was also recently introduced in the digital SDL MDK package [6], the difference being that we also provide the local routing that connects the individual pieces. Figure 2 shows such a transistor layout in the **AAAAAAAA** row configuration.

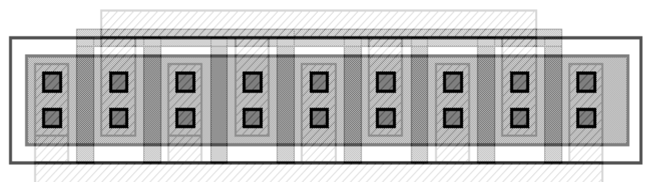

Fig. 2 Interdigitated transistor

2.2 Differential pair

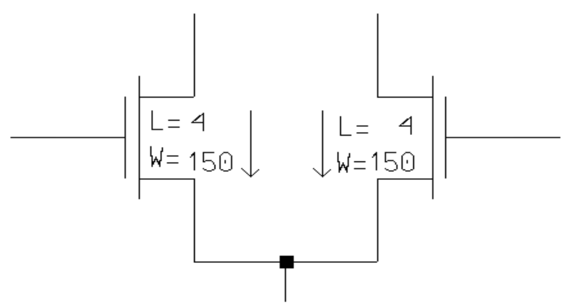

Fig. 3 Differential pair

The differential pair is one of the most commonly used analog building block and consists of two matched transistors with a common source (see figure 3). One possible single-row layout for an interdigitated differential pair in the **AABBAABB** configuration is shown in figure 4. Such a configuration is not common centroid (very close though) but has identical routing for both transistors which matches their parasitics. More importantly, a single diffusion area can be shared by all subtransistors for reduced area. Since there are many choices in laying out interdigitated devices in rows we chose to only implement configuration since it can also be used as the building block for common-centroid multi-row differential pairs.

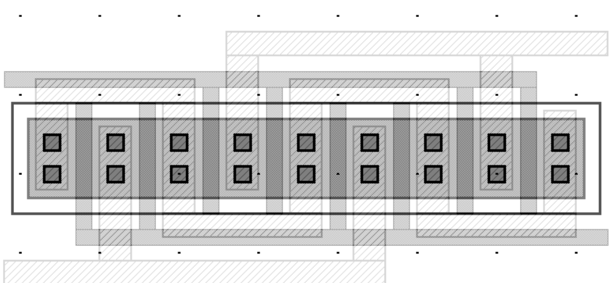

Fig. 4 Interdigitated differential pair

2.3 Matched pair

There are many cases when matched transistor pairs are required that don’t have a common source or drain as in figure 5 (e.g. cascode devices). Interdigitated layouts can still help in that case for matching and wiring but the diffusion area can no longer be shared as can be seen in figure 7 for a **AABBAABB** configuration. This non-common centroid interdigitated row is also used for multi-row common-centroid configurations.

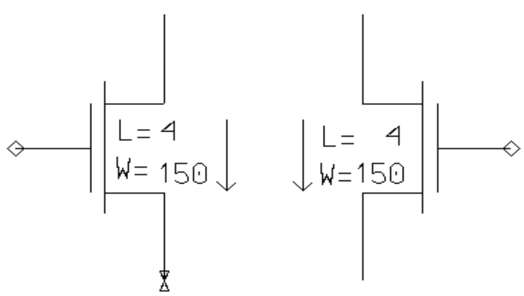

Fig. 5 Matched transistor pair

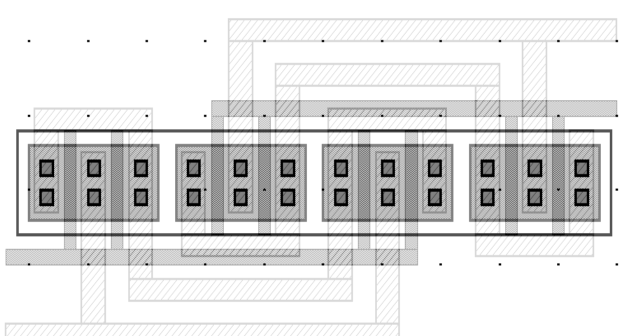

Fig. 6 Interdigitated matched pair

2.4 Multi-row configurations

Multi-row configurations are needed when single-row devices have a skewed aspect ratio. We also chose to support true common-centroids only for multi-row devices. If the number of rows is odd the routing can be identical for both devices such that their parasitics also match, as for the differential pair in figure 7. If the number of rows is even the most efficient routing will no longer be identical for the two transistors (the differences are very small) as can be seen in figure 8. The designer has a choice in either using an even number of rows for common-centroid devices, or an odd number of rows for matched wiring parasitics.

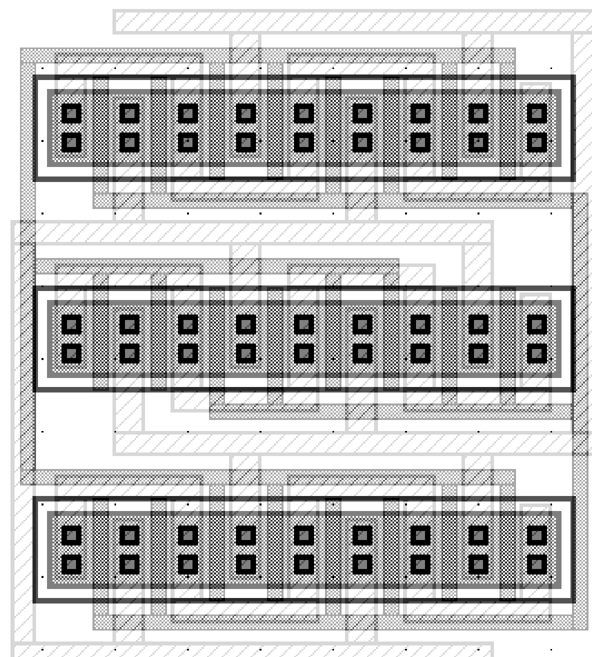

Fig. 7 Multi-row devices with identical routing

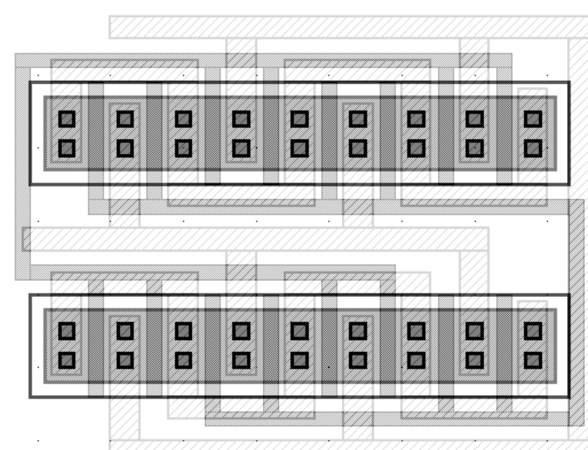

Fig. 8 Multi-row common-centroid

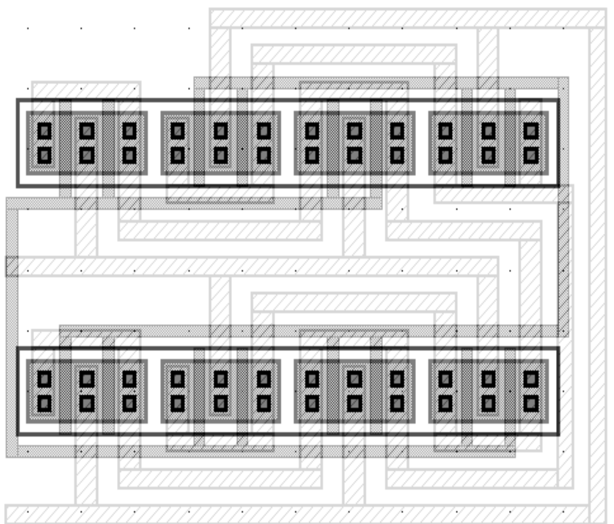

Fig. 9 Multi-row matched pair

In a similar fashion, a pair of matched transistors without a common source can be generated in an even multi-row configuration as in figure 9 for common-centroid, or in an odd multi-row configuration for matched wiring parasitics.

3. Passive devices

Passive devices typically have poor characteristics in a standard CMOS process. Especially resistors and inductors are very difficult, resistors because of large variances in sheet resistance, inductors because of coupling to ground which drastically reduces their quality factor Q. Capacitors have much better characteristics, especially for dual-poly processes. For these reasons it is recommended that resistors and inductors be only used where absolutely necessary and with a good understanding of all the limitations.

There are nonstandard processes that improve resistor tolerance by using special extra layers (e.g. NiCr), and inductor Q by using special dielectrics or postprocessing to allow for suspended wires. Devising generators for such devices may become the subject of future work.

3.1 Resistors

Sheet resistance for many layers is small (ohms/square) and has a very large deviation (more than +100%) from the average value. For this reason the absolute value of resistors in a standard CMOS process can only be interpreted as an approximation. The purpose should be to use matched resistor pairs for their relative values. As in the case of transistors, we generate individual resistors as one- or two-dimensional arrays, as well as resistor pairs in common-centroid configurations. Figure 10 shows such a one-dimensional (folded) resistor array.

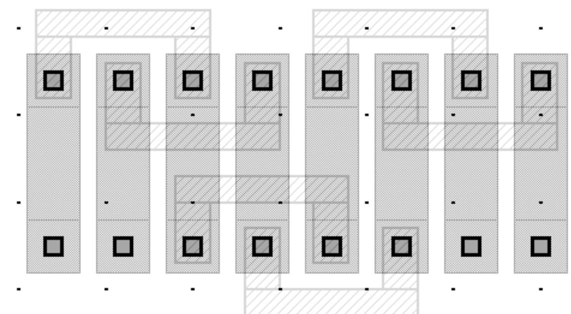

Fig. 10 Folded resistor

3.2 Capacitors

Capacitors have better characteristics in most CMOS processes. In single-poly processes, capacitors can be obtained between poly and diffusion, but a much better result can be obtained in double-poly processes by using the two layers of poly as the two electrodes for the capacitor. We generate individual capacitors as one- or two-dimensional arrays, as well as capacitor pairs in common-centroid configurations. Figure 11 shows such a two-dimensional capacitor array.

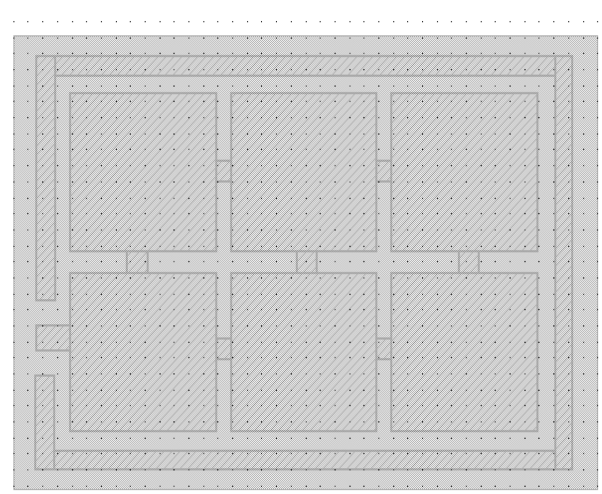

Fig. 11 Two-dimensional capacitor array

3.3 Inductors

Inductors have extremely poor characteristics in a CMOS process but there are situations where they need to be used. For this reason we generate inductor layouts, but more work is needed in characterizing such devices. Figure 12 shows a spiral inductor layout.

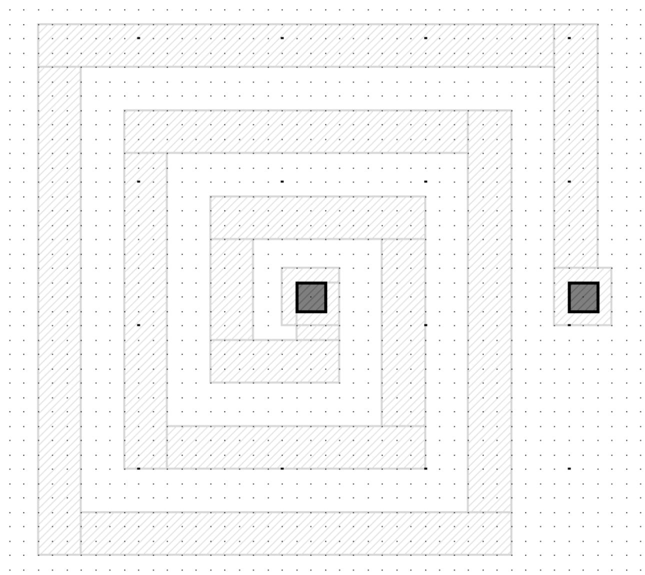

Fig. 12 Spiral inductor layout

4 Other configurations

There are many other active or passive configurations that would benefit from having dedicated device generators:

- matched transistor pair with a common gate,
- a diode-connected MOS transistor,
- transistors in cascode configuration with the source connected to the drain,
- current mirror,
- multi-transistor matching (e.g. several current sources that use the same reference current),
- transistor matching for non-equal transistors (e.g. 1T-2T transistor pair for current source),
- Bipolar devices (transistor, diode, etc.).
- groups of more than two matched passive devices (resistors, capacitors),
- groups of matched but not equal passive devices (e.g. R-2R resistor pair).

Some of the above configurations can be trivially obtained from the existing device generators (e.g. current mirror from differential pair, MOS diode from MOS transistor), others don't seem extremelly critical (e.g. transistors in cascode configuration) while others may become the subject of future work (e.g. multi-device matching, matching for non-equal active or passive devices, bipolar devices).

Conclusion

We have presented the principles behind a schematic driven layout package for analog and mixed-signal design with the MOSIS design kit available from Mentor Graphics Higher Education Program (HEP). This is a complement of the digital SDL already available with MDK [6] and was inspired by several existing tools for analog layout [1,2,4].

Acknowledgments

We thank Mentor Graphics for supporting this effort with a summer grant for 1997. We also thank the members of the univ_sig Special Interest Group for voting for this project and hope they will make good use of it. The project was also partially supported by NSF Career MIP-9703440.

References

[1] J. D. Bruce et al, "Analog Layout Using ALAS!", *IEEE Journal of Solid State Circuits*, vol. 31 no. 7, February 1996.

[2] J. Cohn et al, *Analog Device-Level Layout Automation*, Kluwer Academic Publishers, 1994.

[3] M. Ismail and T. Fiez, *Analog VLSI: Signal and Information Processing*, McGraw-Hill Inc., 1994.

[4] B.R. Owen et al, "Ballistic: An Analog Layout Language", *IEEE Custom Integrated Circuits Conference*, pp. 41-44, May 1995. http://www.eecg.toronto.edu/~gdt

[5] R. Pease, "Comments on Analog Layout Using ALAS!", *IEEE Journal of Solid State Circuits*, vol. 31 no. 9, 1996.

[6] D. Zar, "MOSIS Design Kit Information", http://germanium.ee.wustl.edu/dzar/mentor.html "Using the MDK for schematic driven layout", http://germanium.ee.wustl.edu/dzar/tutorials/sdl/sdl_toc.html