A Python-Based Analog Layout Generation Tool For Nanometer CMOS Technologies

Stephanie YOUSSEF, Damien DUPUIS, Ramy ISKANDER and Marie-Minerve LOUERAT
LIP6-SoC Laboratory, University of Pierre and Marie-Curie, Paris VI, France
Email: stephanie.youssef@lip6.fr

Abstract— **This paper presents a python based analog layout generation tool for nanometer CMOS technologies. To demonstrate the ease of use and extension of this tool, the paper presents how to automatically compute and plot stress effect parameters for two layout techniques of a differential pair device (interdigited and symmetric).**

I. INTRODUCTION

Traditionnaly, the design process of an analog circuit consists of laborious iteration loops. Each iteration is composed of several steps: circuit sizing, layout generation, parasistics extraction and performance evaluation. While performances are not satisfactory, the loop is repeated.

Recently, a new layout-oriented methodolody was proposed [1][2]. In this metholody, design iterations contains simultaneous sizing and layout generation since the layout genreation tool is able to provide parasistics estimation to the sizing tool.

Our tool implements this methodology and thus allows to easily add computations functions to the layout description.

II. PROBLEM DEFINITION

Generating layout using new nanometer CMOS technologies implies to take into consideration two important constraints on layout:

a) Problem of mismatch of the analog devices [3]: Due to manufacturing process, it is hard to accurately control high transistors' widths and lenghts. Transistor folding technique is commonly used to reduce widths values and grid resistance. Thus, this technique provides more accurate geometries and better electrical performances.

b) The Shallow trench isolation (STI) [4] [5] [6]: For technologies below 0.25 um, a new method for transistors isolation was introduces. This method, called Shallow Trench Isolation (STI), consists in trenches etched into the wafer and filled with silicon dioxide to isolate the active area of transistor. Although the STI provides some degree of latchup protection, this isolation degrades the electrical parameters of the transistors. To reduce the impact of this mechanical stress, the layout must be designed so that each transistors of a device is affected in the same way.

III. OUR LAYOUT GENERATION TOOL ENVIRONEMENT

The layout generation tool we present is based on Python language. This choice was motivated by the fact that Python is an easy to learn, object-oriented, portable and interpreted language. This allows designer to write concise and simple code to describe complex layouts.

A. *Stack object*

As previously mentioned, folding technique is commonly used in analog circuits. Since this structure is important, we have defined a 'Stack' object in our layout generation tool. Thanks to this object, the designer only has to call `createStack()` method with well specified input parameters to create the layout of a complete stack.

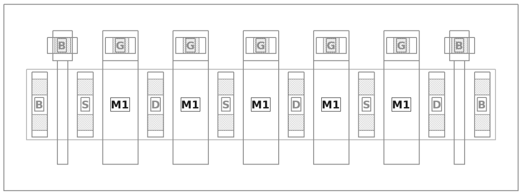


Fig. 1. Layout stack example W=2.0 μ.m., L=0.2 μ.m., NFs=7, Type=NMOS and NDummies=1.

The input parameters of the `createStack()` method are :

- **W:** The sum of all stack's fingers' widths
- **L:** The length of each finger (except dummies)
- **NFs:** The number of stack's fingers (including dummies)
- **NDummies:** The number of dummies at each extremity
- **Type:** The type of the transistor NMOS or PMOS

Figure 1 present an example of a stack layout.

Once a stack object has been created, it can be 'questioned' to get useful layout distances. For each distance presented on figure 2, the stack object provides a method that returns the distance.

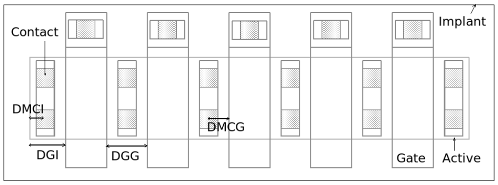


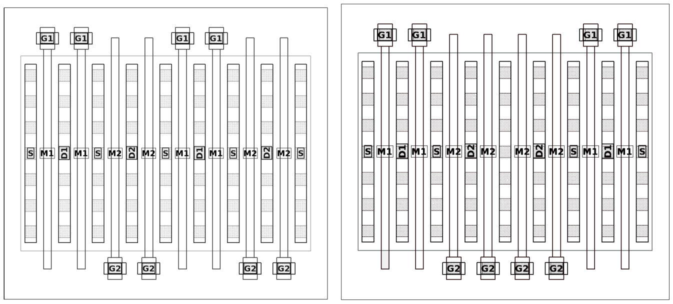
Fig. 2. Useful distances provided by the Stack object.

B. *Extension Functions*

An interesting point is that, thanks to Python, it is very simple to create new calculation functions. For each device we have developed, we have defined a function to compute stress effect parameters and another one to compute area and perimeter of the drain and source zones; which are typical parameters that can be passed to the sizing tool.

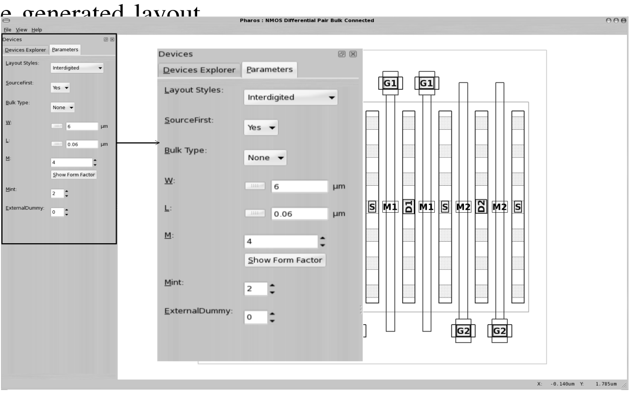
IV. DIFFERENTIAL PAIR EXAMPLE

In the following example, we consider a differential pair device using CMOS 65nm technology. This device is composed of two transistors that have to be matched. Among the several ways to match and organize the two transistors in one stack, we consider the interdigited and the symmetrical techniques [7]. The interdigited technique (fig. 3(a)) simply alternates n fingers of each transistor from left to right, while the symmetrical technique (fig 3(b)) alternates fingers starting from the middle of the stack. The figure 4 presents the graphical user interface of our tool, on which we can see all the input parameters of the device. Each modification on a parameter instantly modifies the generated layout.



(a) Interdigitated technique (b) Symmetrical technique

Fig. 3. Two different layout techniques for a Differential Pair device



Layout Styles: Interdigited

SourceFirst: Yes

Bulk Type: None

W: 6 µm

L: 0.06 µm

M: 4

Mint: 2

ExternalDummy: 0

X: -0.140um Y: 1.785um

Fig. 4. The graphical interface of our layout generation tool.

Our tool offers the possibility to study the evolution (versus M) of parameters (computed by Python functions). Figures 5 and 6 present the evolution of α stress parameter of each transistors for the two considered techniques.

Equation 1 defines α stress parameter. It depends on SA and SB which are distances defined in BSIM4 model. These distances are computed for each transistor of the device and take into account the 'hole fingers' in the alternation caused by the other transistor. Thus they are effective values.

$$\alpha = \frac{1}{\left(\frac{1}{2 \cdot SA}\right) + \left(\frac{1}{2 \cdot SB}\right)} \tag{1}$$

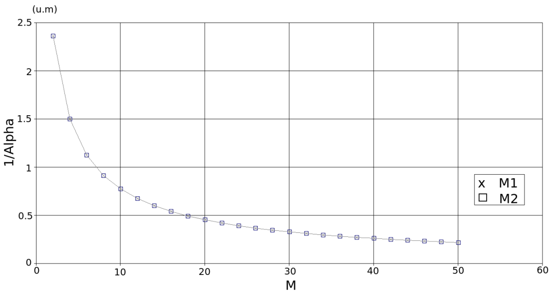


Fig. 5. Stress Effect form factor of the Interdigitated technique.

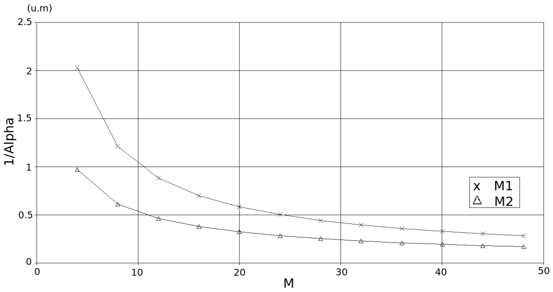


Fig. 6. Stress Effect form factor of the Symetrical technique.

We can observe on figure 5 that, for the interdigited technique, the stress effect affects the two transistors the same way. While, for the symmetrical technique (fig. 6), M1 transistor, placed at the extremities of the stack, suffers a more significant stress effect than transistor M2. This is due to the fact that M1 is closer to the STI.

So, the interdigitated technique in the nanometer technology is more prefereble unlike the older technologies that perefer the symetrical techniques.

V. CONCLUSION

This work is still in progress and will be used as a basis to design analog configurable circuits (before extraction masks or manufacturing). In this paper we have presented a Python-based layout generation tool that allows simple and concise description for complex devices. In addition of creating the layout our tool can be easily extended to compute any layout information such as stress effect parameters.

REFERENCES

[1] Mohamed Dessouky and Marie-Minerve Louerat. A layout approach for electrical and physical design integration of high-performance analog circuits. *IEEE Computer Society*, page pp.291, 2000.

[2] M. Lourat A. Greiner. M. Dessouky, A. Kaiser. Analog design for reuse - case study: very low-voltage sigma-delta modulator. *Proceedings of the conference on Design, automation and test in Europe*, pages pp. 353–360, March 2001.

[3] Behzad Razavi. *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2000.

[4] Wang-Ling Goh and Kiat-Seng Yeo. A comprehensive geometrical and biasing analysis for latchup in 0.18-m cosi2 sti cmos structure. *Journal of Solid-State Electronics*, Vol. 48:pp. 2109–2114, December 2004.

[5] Mark Lambert Cayanes Lee Eng Han, Valerio B. Perez and Mary Grace Salaber. "CMOS Transistor Layout KungFu". 2005.

[6] S. Lazuardi W. Peng K.C. Leong L. Chan W.L. Goh, K.S. Yeo and Alex See. Latchup characterization of 0.18-micron sti cobalt silicided test structures. *Microelectronics Journal*, Vol. 32:pp. 725–731, September 2001.

[7] R. Jacob Baker. *CMOS: circuit design, layout, and simulation*. John Wiley and Sons.