Automated Current Mirror Layout (ACML) Tool

Fady Atef^, Marina Hamdy Zaky*, Nada Khaled Ahmed*, Mario Adel Messiha*, Omar Tarek Abdelwahab*, Abanoub Atef Farid*, Omar Mohamed Selim* and Hassan Mostafa*‡

*Department of Electronics and Communications Engineering, Cairo University, Egypt
^Department of Electronics and Communications Engineering, Ain Shams University, Egypt
^Department of Wireless Analog Layout, Si-Vision, Egypt, fady.atef@si-vision.com
‡University of Science and Technology, Nanotechnology and Nanoelectronics Program, Zewail City of Science and Technology, October Gardens, 6th of October, Giza 12578, Egypt

***Abstract* - Most of the area of research of Analog Layout Automation was pursued by computer engineers rather than analog layout engineers, thus, most solutions disregard a lot of restrictions, constraints and conventions followed by engineers. This paper regards a solution that uses the expertise of analog layout engineers by developing an algorithm that follows the same logic they follow and thus creating a more efficient tool than the ones offered by the current stagnant electronic design automation (EDA) market.**

Index Terms – Analog Layout, Matching, Routing, Common Centroid, Interdigitization.

I. Introduction

Digital design has been almost completely automated since several decades. However, Analog design automation is far from being mature and is still performed manually [1]. This is due to the complex structures of Analog IC design. Analog designs are characterized by a much richer and more complex set of design constraints that need to be considered simultaneously and which may span several domains (electrical, electro-thermal, electro-mechanical, technological, geometrical domain). Therefore, this tool moves the design effort from the human side to the computer side and increases the automation involved in the design. Design automation reduces the design time spent by qualified engineers, thus reducing the cost and the time to market.

Typically, any analog CMOS circuit is divided into building blocks. The layout inside each block greatly affects the circuit performance [2]. Therefore, generating such a building block is a critical step in analog layout design. Two of the main building blocks constituting most circuits are Differential Pairs and Current Mirrors; they are the most critical blocks due to their sensitivity to mismatches. Devices show extensive absolute deviations from their intended values of typically 20% or more caused by stochastic variations during the IC fabrication process. However, if two similar devices occupying the same piece of silicon are compared, it can be observed that they have nearly the same electric parameters because they have experienced identical manufacturing conditions. Therefore, when discussing automation solutions, matching techniques is an essential step to preserve the relative accuracy inside the block against the process variation or thermal effect. As minimum feature of technologies decreases, Nanoscale effects like stress (length of diffusion) and lithographic pattern effects can change considerably device characteristics. This tool generates complete layouts of current mirrors fully matched using the common centroid technique which is the most common technique for this block [3]. A vast collection of challenges was faced due to the demanding trade-off between area and good performance.

Previous approaches on analog layout automation include BALLISTIC, ILAC, KOAN/ANAGRAM [4], and others, however they have only been developed in the university setting but have not caught on in industry.

Most of these researchers employed a Simulated Annealing (SA)-based optimization framework with a certain packing representation to encode the solutions and a set of symmetry constraints to symmetrically arrange the prescribed transistors [5]. These algorithms hardly considered the transistors' orientations and hence introduced unnecessary routing detours and routing mismatch. Others depend on translating constraints into a set of equations and then solving them mathematically to get a solution. Also, Constraints graphs [6] have been developed and widely known algorithms have been adopted to solve them.

In this tool, a more deterministic approach is adopted, that takes into consideration geometrical and parasitic constraints but in a rather implicit way than equations or graphs. Algorithms were developed to track the same steps followed by the engineers when employing common centroid matching technique. Consequently, the second step is developing a full layout routed and matched to any given common centroid transistor matrix.

The tool discussed in this paper consists of 2 main parts, aiming at automating the whole flow of physical design of current mirrors. First is the matching pattern generator and second is the placer and router, both backed up by a graphical user interface (GUI) that appears in Synopsis custom compiler and was developed using TCL scripting language.

As a result of direct contact with the industry, results of this tool were confirmed to be most similar to circuits developed by engineers inside Silicon Vision. User input features and multiple solutions aid the designer to choose the best solution according to the needs of the circuit e.g. more sensitive to area waste or mismatches. However, it cuts the time of development dramatically from long hours to numbered seconds. Average runtime was assessed to be about 90 seconds.

978-1-7281-4058-2/19/$31.00 ©2019 IEEE

II. Matching Pattern Generator

A. Common Centroid Matching

In this approach, we try to go through the same flow as the logic followed by the engineers, instead of dealing with constraints equations. Using the expertise of industrial engineers, Quarter Cell approach has proven to be the best technique in terms of matching of large circuits.

Quarter Cell approach [Fig.1] means building only one quarter of the matching pattern of the whole current mirror (shaded area) then mirroring it to the other side to build a half (2), and to implement the cross-quad technique [Fig.2], we flip this half horizontally then vertically and attach it to the other half, building the whole pattern..

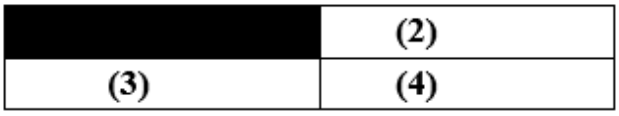

Figure 1: Quarter Cell: (3) is the flipped version of (2) & (4) is the flipped version of the shaded part

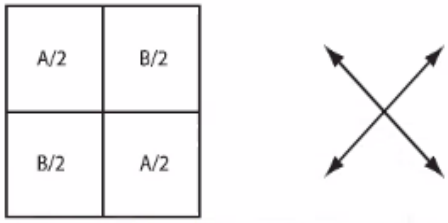

Figure 2: Cross Quad technique [1]

This Generator is divided into two main subsections, one before creating the pattern and the other is creating the pattern itself. In the first subsection, Devices are classified into 2 sets; according to the product of the number of fingers and multipliers (total number of units) for each device; either the set of fours or set of twos. Where the set of fours contains the devices, which can be put in the Quarter (replicated four times), while the set of twos contains the devices which can be put in the Half (repeated two times or their remainder after the division by 4 is 2).

Example:

Device A: 4 (4 units in the set of 4), Device B: 6 (4 units in the set of 4 and 2 units in the set of 2), Device C: 2 (2 units in the set of 2)

Set of fours: [A, B]; Set of twos: [B, C]

Then, we proceed to calculate the total (the sum of product of each device) and create a list of all possible shapes of the current mirror which will be referred to as the list of pairs. These pairs represent the number of rows and columns of the expected matching pattern. This list is ordered according to the aspect ratio and one pair will be chosen according to the floor planner's choice. In addition to the number of rows and columns the exact length and width of the pattern is provided to the floor planner taking into consideration the added space for routability (routing channel estimation)[7].In the second subsection, the devices are arranged into a pattern with two main considerations: first, the diode connected device (reference) is placed in the center, second, number of devices in the same row is minimized for better routability.

B. Simulation Results

1- Simulation Setup:

The input for this tool is a list of devices to be matched - and their number (n) - each paired with the device's product of number of fingers and multipliers which will be referred to as number of units for simplicity. Also, the user must define if the sources of the devices are shared or not and if width division is allowed or not. Source sharing implies that fingers should be placed 2 by 2. It has benefits which are reducing needed silicon area and reduces parasitic capacitance [8]. However, it cannot be done if the sources of devices are not electrically connected (cascode devices) and it creates unequal stresses between devices at the edges and devices in the center of the pattern. Therefore, the designer is given the choice between source sharing or disabling this feature.

Width division means dividing each transistors width by two which implies doubling the number of units. This procedure is done if the number of units of any device is odd, as this represents a challenge for this tool as this device cannot be added in neither the quarter nor the half. This challenge can be solved by two different means; consequently, the user can define which of the solutions will be presented. The first solution is width division; this one will not be practical if the width of any device is close to the technology minimum width; that's why technology minimum width must be an input to this tool. The second solution is to add a dummy finger, by increasing the number of fingers by one, it will be even, so the device can be added in the half or the quarter. However, this solution implies wasting silicon area.

Hereby, after selecting the current mirror of interest inside Synopsis custom compiler, the following dialog box appears; requiring the user to choose the 2 user defined inputs. Other parameters are used by the tool to calculate the exact length and width of the mirror.

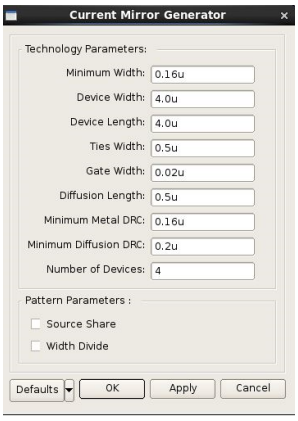

Figure 3: GUI dialog box for user defined inputs

Example: A: 5, B: 4, C: 11, D: 10

Solution 1(Width Divide=true): A: 10, B: 8, C: 22, D: 20

Solution2(Width Divide=false): A: 6, B: 4, C: 12, D: 10

After this step, processing is done on these inputs to produce all possible shapes of the mirror (list of pairs). All these pairs are displayed for the designer for him to choose the convenient shape for his floorplan. Each pair is accompanied by the exact length, width, routing channel needed in addition to the number of rows and columns.

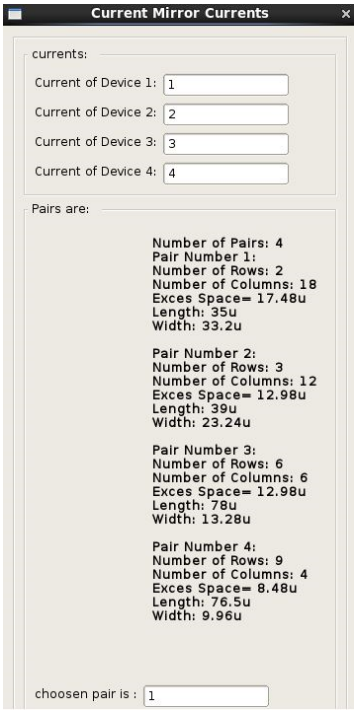

Figure 4: Dialog box to choose the pair

Also, the designer is required to enter the current flowing in each device, which is an important parameter used by the router to compute routes' widths.

After choosing the convenient pair, the pattern is generated and passed to the placer and router to produce full layout.

2- Results:

For n=15, Device list: A:7, B:4, C:7, D: 8, E:8, F:6, G:2, H:4, I:10, J:3, K:9, L: 6, M:10, N:2, O:5

Solution1: Modified Device list: A:28, B:16, C:28, D: 32, E:32, F:24, G:8, H:16, I:40, J:12, K:36, L: 24, M:40, N:8, O:20 and dummies referred as D

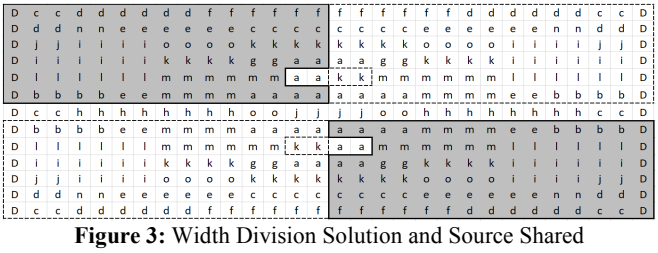

Figure 3: Width Division Solution and Source Shared

Solution2: Modified Device list: A:8, B:4, C:8, D:8, E:8, F:6, G:2, H:4, I:10, J:4, K:10, L:6, M:10, N:2, O:6, Number of added dummies: 21 (referred to as D)

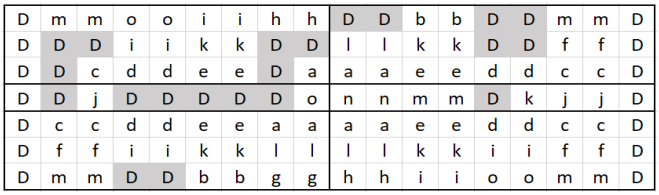

Figure 4: Adding Dummies Solution and Source Shared

III. PLACER AND ROUTER

A. Automated Routing of Current Mirror

In this approach, we build our work on human imitation of routing by using TCL scripting language, also, all routing follows Manhattan method [9].

The code is divided into a number of functions, firstly a function to put the poly and gate contacts over each gate, secondly a function to put the NIMP to fulfill the DRC of implant enclosure over poly, a function to put the vertical metals (metal 2) which connect all the sources together and also extends the space of the drain metal. This extra space is needed for horizontal internal routing. The rest of functions include a function to make metal 3 horizontal routes that connect the drain of the same devices in each row. The width of each route depends on the current flowing in it. Therefore, not all the horizontal routes have the same width. Also, the number of horizontal routes over each row of devices depends on the maximum number of devices in any row. After placing horizontal routes, a function is executed to put metal 4 vertical routes that connect the drains of each device between different rows. The number of these routes depends on the total number of transistors and their widths depend also on the current capability needed from each route (transistor). Additional functions include a function to put bulk ties (bulk contacts) between different rows and around the whole block to act like a guard ring that isolates the block from the noise of surrounding blocks. Lastly, electrical connections are made by a function that puts a via on every 2 nets have the same name; each net was already named by the name of its device in order to put vias at the intersection.

B. Simulation setup

The input to our algorithm is a text file that results from the matching pattern generator discussed in the previous sections. This file contains data that helps the TCL script build the mirror, like: the pattern, whether the devices are NMOSs or PMOSs and current flowing in each device.

C. Time complexity

According to feedback from the layout design team in silicon vision company; this routing algorithm reduces the time complexity from hours to seconds, as when the current mirror increases in size, the manual routing time increases exponentially but, in our algorithm, there is no remarkable change in time complexity when the mirror gets bigger

The shown example is made manually by the company engineers in 3 hours of working but with our algorithm made in 32 second so we speed up the process in a remarkable way

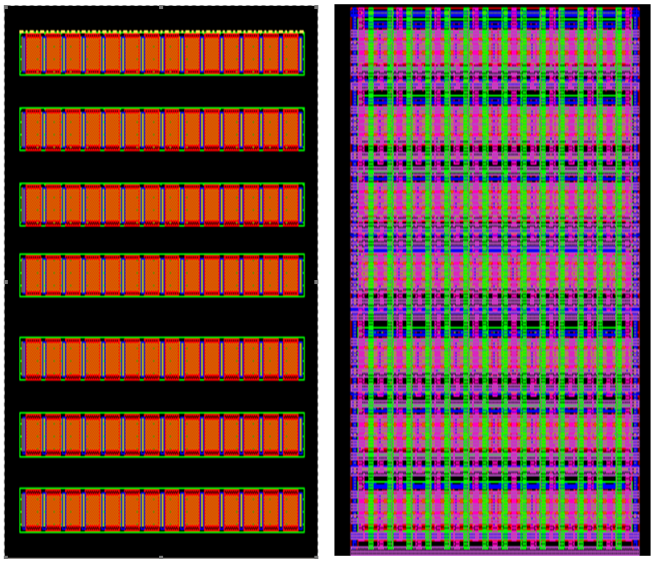

Figure 7: Current mirror before and after the Routing

D. Physical Verification

At this stage we run the DRC and LVS checks on the fully automated routed current mirror and it passes successfully as shown in the next figure.

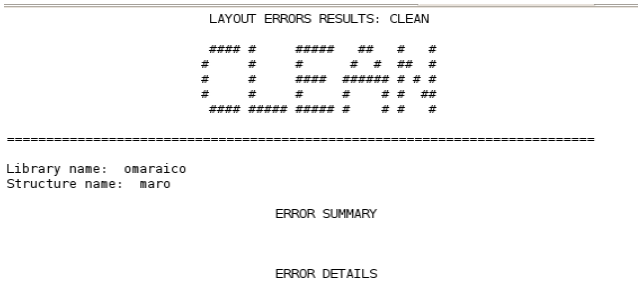

Figure 8: Physical Verification results

V. CONCLUSION

Both parts have various features that make ACML tool stand out; first concerning the matching pattern generator, it uses 2 user defined inputs that produce 4 different patterns for each given mirror. This lets the user choose between variations of the pattern according to which is more important to project requirements whether it is area or parasitic or matching. It also provides the user with a list of all shapes (aspect ratios) of the given mirror, in addition to estimation of the routing channel needed for each shape and then the user can choose which is more convenient to his floorplan. Also, concerning the routing phase, ACML tool produces layouts in which routing is fully matched, current consumption in each device is considered to avoid electron migration, in addition, certain practices were implemented to avoid adding gate parasitic, other practices to properly tie transistors bulks to the supply (VDD or ground for PMOS or NMOS) to avoid latch up problem. All layouts produced passed physical verification checks; either DRC or LVS.

ACKNOWLEDGMENT

This work was funded by Si-Vision Co., a leading IC solution Company following Made in Egypt initiative to boost scientific innovations in Egypt, also his work was partially funded by ONE Lab at Zewail City of Science and Technology and at Cairo University, NTRA, ITIDA, and ASRT.

REFERENCES

[1] Graeb, Helmut E., “Analog layout synthesis: a survey of topological approaches”, Springer, 2010.

[2] Ahmed Adel, Ahmed Mohamed, Dina Reda, Fady Atef, Marwa Mostafa, and Mohamed Omar "Equalizer Implementation for 10 Gbps Serial Data Link in 90 nm CMOS Technology", Microelectronics, 2007.

[3] V. Meyer zu Bexten ; C. Moraga ; R. Klinke ; W. Brockherde ; K.-G. Hess, “ALSYN: flexible rule-based layout synthesis for analog IC's ” IEEE Journal of Solid-State Circuits, vol. 28, Mar 1993.

[4] J. M. Cohn, D. J. Garrod, R. A. Rutenbar and L. R. Carley, "KOAN/ANAGRAM II: new tools for device-level analog placement and routing," in IEEE Journal of Solid-State Circuits, vol. 26, no. 3, pp. 330-342, March 1991.

[5] E. Malavasi, E. Charbon, E. Felt and A. Sangiovanni-Vincentelli, "Automation of IC layout with analog constraints," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 8, pp. 923-942, Aug. 1996.

[6] X. Jingnan, J. Vital, N. Horta, A SKILLTM—based library for retargetable embedded analog cores, in Procedings on Design, Automation and Test in Europe (DATE), pp. 768–769, Mar 2001.

[7] N. Jangkrajarng, S. Bhattacharya, and R. Hartono, “IPRAIL – intellectual property reuse-based analog IC layout automation,” Integration, VLSI J., Nov 2003.

[8] N. Lourengo, M. Vianello, J. Guilherme, and N. Horta, LAYGEN – Automatic layout generation of analog ICs from hierarchical template descriptions, in Research in Microelectronics and Electronics, 2006.

[9] L. Zhang and U. Kleine. A novel analog layout synthesis tool. In Proceedings of the International Symposium on Circuits and Systems, volume 5, pages V-101–V-104, 2004.

[10] K. Lampaert, and G. Gielen, A performance-driven placement tool for analog integrated circuits, IEEE J. Solid-State Circ., 1995.

[11] Balasa, Florin, Sarat C. Maruvada, and Karthik Krishnamoorthy. "On the exploration of the solution space in analog placement with symmetry constraints." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 23, no. 2 (2004): 177-191

[12] Sherif Ahmed Mohamed, Mohamed Dessouky, Fady Atef, Soha hamed, “Analog Layout Placement Based on Unit Elements and Routing Channel Estimation”, SMACD 2019.

[13] Saif, S., Dessouky, M., El-Kharashi, M.W., Abbas, H. and Nassar, S., 2016, March. Pareto front analog layout placement using Satisfiability Modulo Theories. In 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 1411-1416).

[14] K. El-Kenawy, I. Mohammed and M. Dessouky, "Analog layout placement exploiting sub-block shape functions”, SMACD 2016.

[15] Y. Ghallab, H. Mostafa, and Y. Ismail, “A New Current Mode Implementation of a Balanced-Output-Signal Generator”, Analog Integrated Circuits and Signal Processing, SPRINGER, 2014.

[16] A. Elbayoumi, H. Mostafa, and A. M. Soliman, “A Novel MIM-CapacitorBased 1-GS/s 14-bit Variation-Tolerant Fully-Differential Voltage-to- Time Converter (VTC) Circuit”, Journal of Circuits, Systems and Computers (JCSC), vol. 26, 2017.

[17] H. Mostafa, and Y. Ismail, “A Design-Oriented Timing Jitter/Skew Model in Voltage-to-Time Converter (VTC) Circuits”, Analog Integrated Circuits and Signal Processing, SPRINGER, vol. 82, issue 1, pp. 309-321, 2015.