Schematic-Driven Physical Verification: Fully Automated Solution for Analog IC design

Ahmed Arafa* (ahmed_arafa@mentor.com), Hend Wagieh* (hend_wagieh@mentor.com),
Rami Fathy*(rami_fathy@mentor.com), John Ferguson* (john_ferguson@mentor.com),
Doug Morgan** (doug.morgan@onsemi.com), Mohab H. Anis*** (manis@vlsi.uwaterloo.ca),
and Mohamed Dessouky* (mohamed_dessouky@mentor.com)
* Mentor Graphics Corporation
** ON Semiconductor Corp
*** The American University in Cairo

***Abstract*—Designing ICs (integrated circuits) is inherently a complex task involving human expertise as well as aids intended to accelerate the process. A fundamental requirement for design success is a clear strategy that coordinates the entire design flow from specifications to a marketable product. Modem VLSI (very large scale integration) IC designs, especially analog/mixed signal LSIs, must meet various design and electrical constraints such as IR-drop, cross-talk, low power and low voltage design. This complexity poses several physical design challenges for specific analog structures such as device symmetry, net matching and more. Therefore it is essential to have some form of communication between the front-end designers (schematic and transistor level) and the physical layout engineers. The current process involves the front-end designer placing the design constraints in the form of annotations on the schematic netlist. These annotations are then passed to the layout team for manual implementation followed by visual verification only. This paper explores a new methodology providing a fully automated CAD flow that captures the designer's intent from the schematic netlist, links these annotations to the proper devices or nets on the physical layout level, then runs verification checks using the Calibre® tool suite. Several applications can be used with the proposed flow, in this work we will present an application specifically for capturing design constraints related to physical layout recommendations, tested on Opamp circuit.**

***Index Terms*—Physical Verification, Schematic Driven Physical Verification, Analog Design**

I. Introduction

With the move to nanometer geometries, IC design is becoming considerably more complex and time consuming. At the same time, analog content is increasing, reflecting strong growth in wireless and sensing technologies. Market research indicates that although analog circuitry takes up only 20% of the area of today's modern mixed signal devices, it is likely to account for as much as 80% of yield loss. The inevitable shift to smaller geometries and correspondingly lower voltages has been steadily creating design challenges in the analog domain for some years. Unlike digital design, which has to be either on or off, the electrical characteristics of analog (amplification, resistance, saturation to linear, parasitic crossing blocks, etc.) are very complex [1]. Making matters worse, few standard practices have been established among analog designers, especially regarding communication of the design's intent to those who will layout the chip [2]. This is true from company to company, but also from design group to design group within the same organization.

Existing methods such as placement of text notes on schematics, manual annotation of schematic printouts, email, and even verbal instruction have proved to be inadequate. All too frequently, poor communication has resulted in major layout rework that adds significant time to the design cycle. Risk of design problems is also increased because much of this rework is hastily done in the final days prior to tapeout. Worse yet, since some communication has no permanent record, this can lead to completely missed requirements that result in non-parametric performing silicon that requires a costly second run through the fab.

In this paper we present a fully automated solution that captures electrical and physical constraints from the schematic level and propagates these constraints into layout rules to ensure no discrepancies between schematic design and layout. The proposed design methodology guarantees that all design constraints are met, when feasible, thus providing a robust and efficient design environment that reduces design costs and time-to-market.

The rest of this paper is organized as follows: Section II and Sections III depict the proposed schematic driven layout checking methodology, entailed in implementing the engines used in the flow. To verify its usefulness, the proposed methodology is examined on an OPAMP (operational amplifier) design, and the results are highlighted in Section IV. Finally, our concluding remarks and the proposed future work are given in Section V.

II. Flow Overview: Schematic Driven Layout Checking

In this section we will introduce a fully automated flow that is based on Calibre verification tools [3] and some TCL scripts that link these tools together.

The proposed flow, as shown in the Figure 1 assumes that the front-end designer places some annotations on the

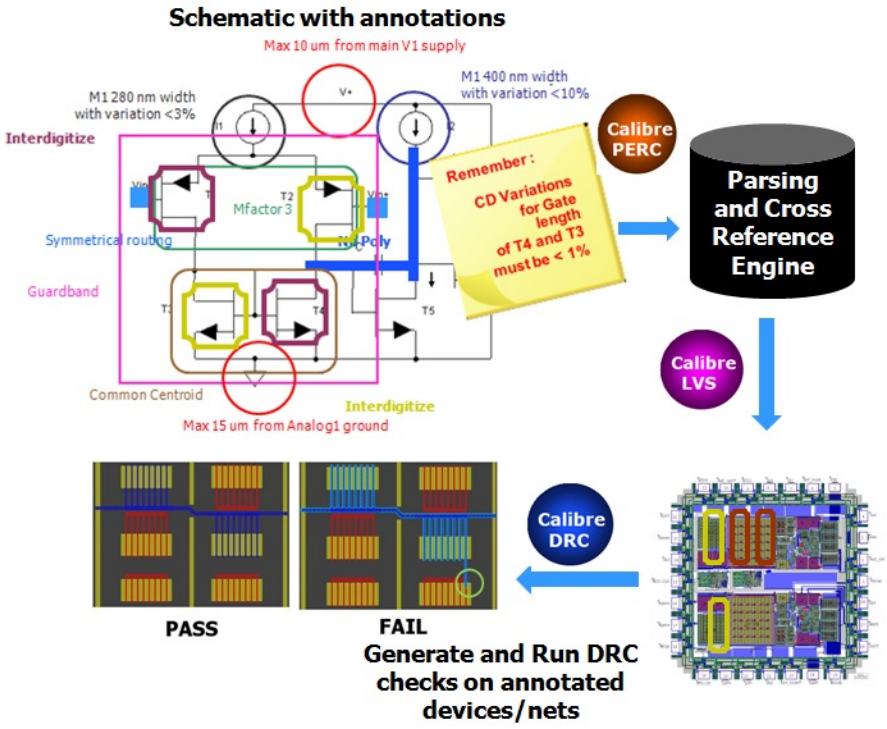

Fig. 1. Flow Overview: Schematic Driven Layout Checking

schematic netlist in a certain format. These annotations informs the physical layout engineers that certain devices and nets should have special treatments such as device symmetry [4] and orientation [5], device width segment matching, net balancing [6], devices placed in common centroid arrangement, devices not isolated by guard rings, net is not capable of meeting the device performance requirements, and more.

Once the layout engineers finish their work, the designer uses the proposed CAD flow to check that his intent has been correctly passed and properly implemented on the physical layout design.

The process involves gathering the annotation and device/net information from the schematic netlist and the layout. Then this information is processed to generate text, marker layers or other geometry identification on the layout to mark the annotated devices/nets. The type of annotation together with the text and marker layer numbers are used to generate specific rules based on the assigned annotations. These rules are then checked using Calibre tools [3] and the results are reported. The flow can be divided into the following steps:

A. Parsing the schematic annotations
B. Linking schematic to layout database
C. Placing the annotations on the layout
D. Generating DRC based on the schematic annotations
E. Running the generated DRC
F. Integrating the flow

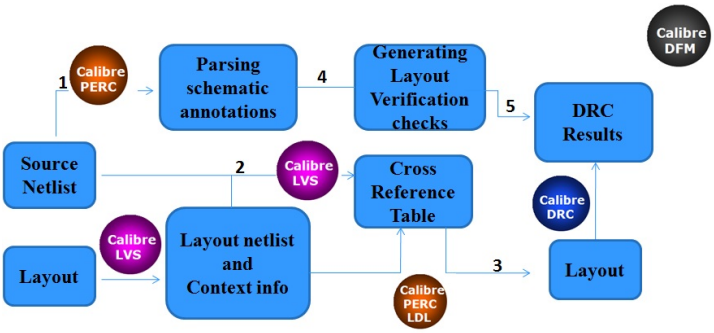

Fig. 2. Schematic Driven Layout Checking Flow

III. Flow Structure

A. Parsing the schematic annotations

The first step parses the annotations placed in the schematic netlist (such as physical layout recommendations). This step also correlates each device/net with its corresponding design constraint check; this information may be reported in a separate output file containing each device or net and its associated annotation.

B. Linking schematic to layout database

The second step links the parsed device or net to its corresponding mate in the layout. The main link between the schematic netlist and the layout is the Layout versus Schematic (LVS) deck that is used to perform the LVS comparison.

Once the layout is LVS-clean, an LVS database is generated. This database contains complete information that specifies the device and net properties on both the schematic and the layout databases. In addition, information about the connectivity and the layer properties of each layer on the layout is available in this database.

In the Calibre LVS output database, there are two main files needed:

1) Instance cross-reference (IXF) file: A text file that contains matched instances between the schematic and the layout, representing for each instance its name, properties, and are the lower left coordinates on both the schematic and layout databases.
2) Net cross-reference (NXF) file: A text file that contains matched nets between schematic and layout, representing for each net its name, properties, and are the lower left coordinates on both the schematic and layout databases.

These two files are used in the proposed flow to cross-reference the schematic and the layout. For this step to run correctly, the design must be LVS-clean, to ensure the generated cross-referencing files correctly map to the equivalent devices.

C. Placing the annotations on the layout

The third step uses the coordinates obtained from the cross-referencing files to place a text layer on each appropriate device. In this step, a software script is used to organize and apply the data obtained from the first two steps to generate the required text layers to mark the annotated devices. Because a net may be composed of many layers, one pair of coordinates is not enough to mark a complete net, this is because the net may be composed of many metal layers connected to each other, so, for the net checks, a script is used to access the LVS database which already contains all the information required about the metals involved in this net including their coordinates and connectivity, this information is used to extract the complete net coordinates and completely cover the concerned net with a marker layer.

D. Generating DRC based on the schematic annotations

Each annotation has an equivalent specific procedure coded in an internal database which corresponds to the physical check ”DRC rule” that corresponds to this annotation. This specific procedure is called whenever this annotation is detected. This procedure accepts the information generated in the previous steps ”text gds layer number, marker gds layer number, text coordinates and layer coordinates”. Afterwards the procedure generates the required DRC rule.

E. Running the generated DRC

Finally, the generated rule files are executed using the Calibre tool suite [3]. The violations are viewed and highlighted on the layout and the designer is now able to identify the violations found in the design.

F. Integrating the flow

When integrating the above five steps in one unit and allowing them to communicate with each other, we get a fully automated solution that the designer uses to verify the layout based on the annotations placed at the beginning in the schematic netlist. The number of generated rules and the associated devices and nets are identified by the detected annotations.

The pseudo code description of the engine is shown in Figure 3.

1. **Parsing The Schematic Annotations:**
   - The schematic netlist is parsed searching for annotation properties
   - Analyze each annotation to produce a new schematic netlist where the annotated devices/nets attached to the annotations are re-named. For example device **M1 &M2** which have constraints that ΔIdsat < 10%, they are re-named to be **M1_Idsat_10 and M2_Idsat_10** . The same is applied for the nets
   - New adjusted schematic netlist is passed to Calibre PERC which reads the device/net names and collect the information about the device/net annotations in a separate file. Run Calibre PERC
   - The first output file has the device annotations
   - The second output file has the net annotations
2. **Linking the schematic database to the layout database:**
   - Run LVS between the schematic netlist vs. the layout (with two options **–ixf** and **–nxf**) to link the device/net annotations in the schematic to their corresponding ones in the layout
     calibre –lvs –ixf –nxf rule_file
     where ixf stands for “instance cross referencing” while nxf stands for “net cross referencing”
   - For correct mapping the layout should be LVS clean
   - Two output files: “***.ixf**” & “***.nxf**”, contains devices/nets ID and coordination. Also contains correlation of the devices/nets between the schematic netlist & the layout.
3. **Marking the annotated devices and nets on the layout database:**
   - Mark the annotated device in the layout by placing the associated annotations as a text on the lower left most of each finger of the device.
   - Run **calibre –query database_name** to get all the coordinates of the complete net. Run a script to place a marker with a predefined gds number
   - If there are general checks that must be applied on predefined analog topological structures, the steps 1 & 2 are bypassed, and a library of these sets of structures is built (using Calibre PERC) to mark these devices on the layout.

Fig. 3. Pseudo Code for the Intent Driven Design Engine

IV. Experiments, Results and Discussions

This flow was implemented and tested on a simple opamp testcase (Figures 4 and 5) provided by ON Semiconductor Corporation (ON Semi). ON Semi designers identified several design issues related to specific devices or specific nets that might significantly affect performance. Annotations were placed in the schematic netlist, and consequently specific DRC rules were generated and checked. Pass/Fail testcases were implemented.

- *Sample annotations defined for device checks*

1) **Device Symmetry Check.** This rule flags an error if the two devices (Mo1 & Mo2) are asymmetrical in the layout. Asymmetry includes the two devices having

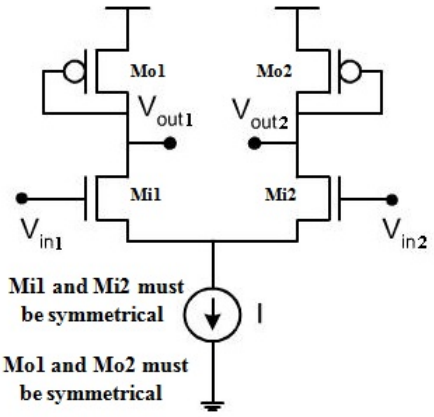

Fig. 4. OPAMP schematic design

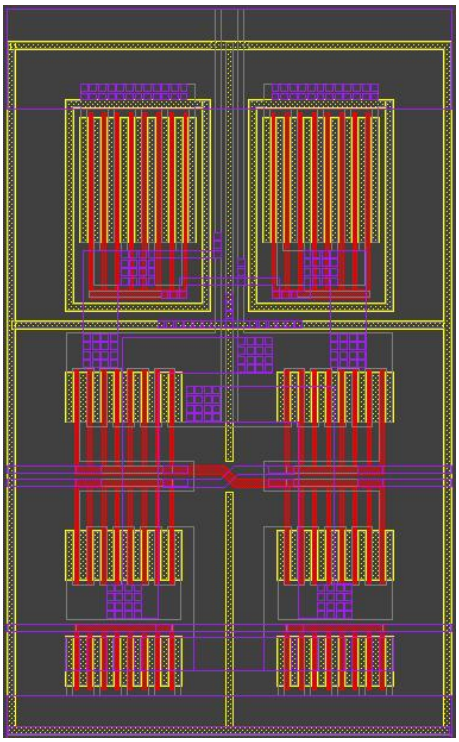

Fig. 5. OPAMP layout design

different shapes or being unaligned. Figure 6 shows a Pass/Fail testcase, where the fail case has a slight shift in Y direction.

2) **Device Width Check.** This rule flags an error if the two involved devices (Mo1 & Mo2) do not have equal widths in the layout.
3) **Device Orientation Check.** This rule flags an error if the two involved devices (Mo1 & Mo2) do not have the same orientation in the layout. Figure 7 shows a Pass/Fail testcase, where one device has different orientation from its matched pair.
4) **Common Centroid Check.** This rule flags an error if the two involved devices (Mi1 & Mi2) are not placed

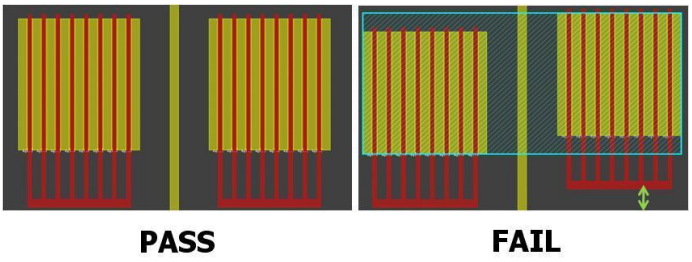

Fig. 6. Device Symmetry Check

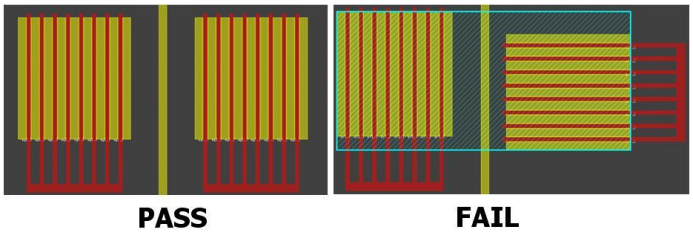

Fig. 7. Device Orientation Check

in a common centroid placement in the layout. Figure 8 shows a Pass/Fail testcase.

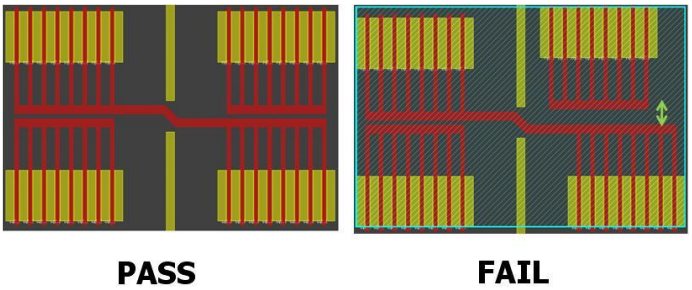

Fig. 8. Common Centroid Check

- *Sample annotations defined for net checks*

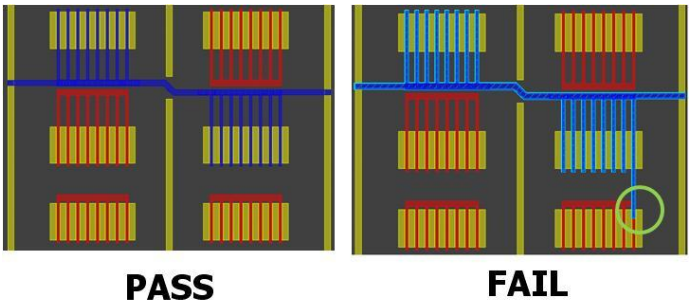

Fig. 9. Do Not Route over a Device Gate Check

1) **Do Not Route Over a Device Gate Check.** This rule flags an error if a net is routed over the gate of a device not connected to that net. As the net passing over a gate induces stress effects on that gate. Figure 9 shows a

Pass/Fail testcase for nets (Vin1 & Vin2), where the fail case has a net crossing over a device.

2) **Net Balancing Check.** The rule flags an error if the two involved nets are not balanced in the layout. Balancing requires that the two nets have equal shapes and be symmetrical around the X & Y axis. In Figure 10, one of the nets has a space of 0.5um while the other has a space of 0.495um, creating a balance violation.

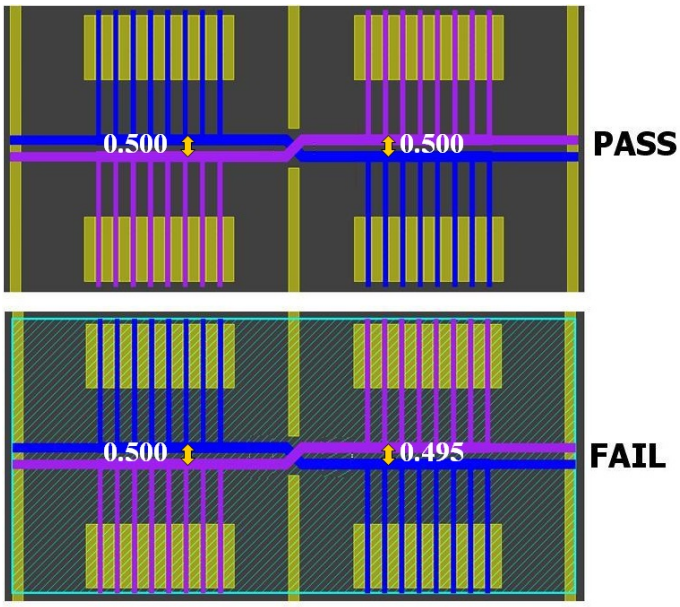

Fig. 10. Net Balancing Check

V. Conclusions and Future Work

A new methodology that links the front-end design stage to the back-end verification stage has been introduced. The ability to link the layout side to the schematic side is very advantageous, as it allows applying further procedures and rule checks on the physical design based on the information provided on the schematic. This flow facilitates passing the necessary information through a standard communication channel from the schematic designer to the layout engineer, in which the information that is placed in the schematic netlist can be passed over to the corresponding instance, net and device in the layout. This enables the layout engineer to apply further checks, procedures and applications that make use of this back-annotated information to verify the reliability of his design. A detailed flow is explained with its different stages. This flow introduces a new concept we call layout driven-schematic. It detects the annotated instances, devices and nets from the schematic netlist and consequently mark them in the layout with the specific annotations given to each. By means of cross-referencing and a linking methodology, further layout design rule checks can be applied on the marked geometries according to the annotations given to each, and corresponding to certain design rule that needs to be verified.

Future work would be automation and integration of this flow with different layout verification tools to be more user-friendly and facilitate the flow invocation. Also, the described flow needs slight modifications to enable reading the annotations of the schematic netlist if the format is different than the one we used. Other work may involve covering different aspects of checks; not only design rule checks but checks concerned with current density calculations and point to point resistance extraction as well.

References

[1] L. L. Lewyn, "Physical design and reliability issues in nanoscale analog cmos technologies," *NORCHIP*, pp. 1–10, 2009.

[2] S. Bhattacharya, N. Jangkrajamg, R. Hartono, and C.-J. R. Shi, "Hierarchical extraction and verification of symmetry constraints for analog layout automation," *Asia and South Pacific Design Automation Conference*, 2004.

[3] *Calibre Verification User Manual*. Mentor Graphics Corp.

[4] F. Balasa, S. C. Maruvada, and K. Krishnamoorthy, "On the exploration of the solution space in analog placement with symmetry constraints," *IEEE Transcations on Computer-Aided Design of Integrated Circuits and Systems*, pp. 177 – 191, 2004.

[5] B. Wang, H. Nguyen, J. Mavoon, A. Horch, Y. Ma, T. Humes, and R. Paulsen, "Effect of layout orientation on the performance and reliability of high voltage n-ldmus in standard submicron logic sti cmos process," *IEEE International on Reliability Physics Symposium*, pp. 654 – 655, 2005.

[6] Y.-M. Yang and I. H.-R. Jiang, "Analog placement and global routing considering wiring symmetry," *International Symposium on Quality Electronic Design (ISQED)*, pp. 618 –623, 2010.