

TOWARDS A UNIFIED TOP-DOWN DESIGN FLOW FOR FULLY DIFFERENTIAL LOGIC BLOCKS WITH IMPROVED SPEED AND NOISE IMMUNITY

Ilhan Hatirnaz, Stéphane Badel, Yusuf Leblebici

Microelectronic Systems Laboratory (LSM)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Batiment ELD Station 11, CH-1015 Lausanne, Switzerland
E-mail: {ilhan.hatirnaz, stephane.badel, yusuf.leblebici}@epfl.ch

ABSTRACT

A new top-down design flow (RTL-to-GDSII) is proposed for achieving high-performance and noise-immune designs consisting of differential logic blocks. The differential building blocks are based on the current-mode logic (CML), which offers true differentiability with low-swing signalling, switching-independent constant power dissipation and very high-speed operation. The goal of this flow is to allow effective cancellation of inductive and capacitive noise in high-speed on-chip interconnect lines using a simple generic interconnect architecture.

1. INTRODUCTION

Currently, the implementation of chip design is an iterative process guided by the design automation tools and the conventional linear design flow. In the ideal case, the limiting factor in achieving the design goals would come only from the physical limitations of the process technology used. However, especially starting with the use of deep sub-micron technologies, the design flow and the electronic design automation (EDA) tools started becoming the limiting factors of what final performance can be achieved.

There is a number of reasons behind this fact. One reason is the increasing complexity of the designs, which, can not be handled by currently available tools. Furthermore, traditional VLSI design flows may mask some problems, which only show up during or after the final steps of the flow (for example, after the detailed routing is finished), where, in some cases, a dramatic change at the high-level description of the circuit might be necessary.

In addition to the increasing design complexity, the fabrication costs of ASICs are rising rapidly as the latest technologies with much smaller feature sizes are offered. This leads to a tendency from standard-cell based design to more soft-programmable design solutions, like FPGAs and/or processor-based solutions, where the mask-cost is reduced to minimum. Although these solutions help reducing the total cost and time-to-market, they are not able to offer comparable performance as ASIC-based solutions. Recently, to address this problem, structured-

ASICs (SA) were introduced [1]. Structured ASICs are expected to fill the gap between FPGAs and standard-cell based design approaches. Structured ASICs are based on a predefined and pre-built logic fabric, which is fabricated including the interconnect structure consisting of a number of the bottom metal layers (for example up to Metal3 or Metal4), where the rest of the metal layers are to be laid out later for having the design mapped to the wafer. These wafers are stored as base wafers until ordered by customers.

This project aims to find a generic solution to signal integrity problems and a differential-design flow that employs a fully differential (differential inputs-differential outputs) standard cell family based on current-mode logic (CML). This paper introduces the first step into the search for a complete solution. Proposed is a flow for the implementation of a design using differential gates or structures, where the signal integrity issues are to be addressed early at the design flow, even in the library modelling phase. The target device of this flow can be either a standard-cell library or a structured-ASIC based solution; in this paper only implementation targeting a cell library is discussed. The benefits of such a scheme is the regularity and hence the increased predictability of the final design, and additionally, shifting the limitations from the limits of the tools or flows to the process technology limits. These goals, if achieved, would prevent the under-utilization of the very deep sub-micron (VDSM) technologies, which means that the designers would be able to achieve higher clock speed using the same technology at either an acceptable or no additional cost, resulting in a faster time-to-market.

The remainder of this paper is organized as follows: The proposed differential design flow is introduced in Section 2, including the individual steps and the CML-based cell library. In Section 3 the test chip including the blocks designed using this and regular design flows is presented, followed by the conclusions.

2. DIFFERENTIAL DESIGN FLOW

There are a number of well-known advantages of using differential signalling in high-performance systems in terms of signal integrity and noise immunity. The primary

disadvantage of differential signalling is the increased number of traces per bit of information, which proportionally increases the cost of the associated routing and the total silicon area, which, in fact, constitutes the main reason for making use of differential signalling and differential gates only in some very high performance designs (for example, microprocessors [2]) and only in specific cases, like routing bit lines of RAM-structures. Therefore, there is not as much interest and support from the EDA world for differential designs as there is for conventional single-ended cell libraries. There are no commercial tools that provide differential logic synthesis, moreover, conventional hardware description languages do not support differential design entry.

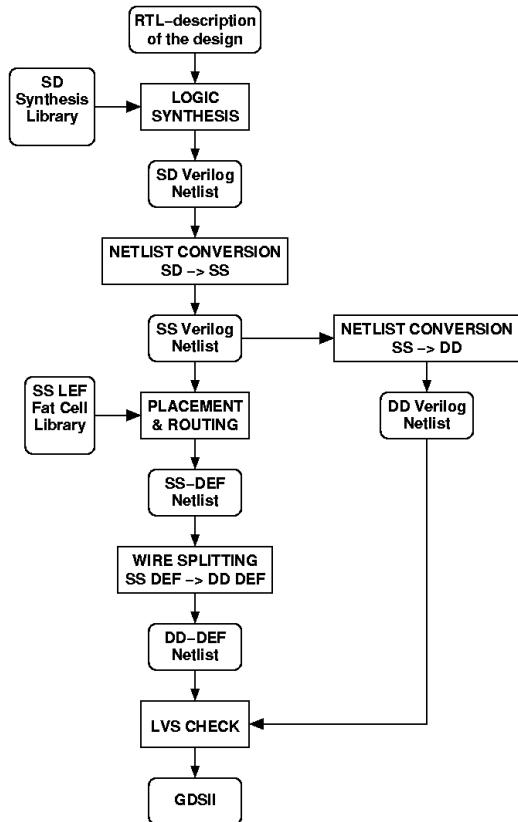


Figure 1 The proposed RTL-to-GDSII differential design flow. It should be noted that ‘S’ stands for single-ended and ‘D’ stands for differential in the descriptions of the different netlists.

2.1 Logic Synthesis

The proposed design flow is given in Figure 1. The main pieces of this flow are commercially available EDA tools and a number of netlist conversion scripts. The main input to the flow is a synthesizable RTL description of the design. The RTL code does not need to include any knowledge of differentiability, it only describes the design in a single-ended manner.

Even a fully characterized differential cell library (differential inputs-differential outputs) is available, current synthesis tools are not able to provide mapping of nets to differential inputs pairs of gates from this differential library. To overcome this issue and also make use of the complementary nature of the differential cell outputs, a new synthesis library is extracted from the fully differential library, where this new library consists of single-ended input/differential output (SD) gates. The logic synthesis tool (Synopsys Design Compiler [3]) is able to benefit from the differential outputs of the logic gates offered by the SD cell library, i.e., the tools uses either both signals (inverted and non-inverted) or one of them without needing to invert the complementary net of the pair.

After the mapping process is finished, the synthesized circuit is written out as a Verilog netlist. This netlist, consisting of SD gates, are then converted first to a single-ended input/single-ended output netlist and then to a fully differential Verilog netlist using the netlist conversion scripts. The observe if the same functionality is kept for all the different netlist of the design, these scripts also provide a run file to be fed to the equivalency checker tool (Synopsys Formality [3]) with the Verilog netlists under comparison.

During logic synthesis, it can happen that only one of the complementary outputs (either Y or Y') drives an input of any other gate, whereas the other signal stays floating, because the inputs are only single-ended. This means that the loading of the complementary nets in one differential pair might not be the same. During the SD-to-DD Verilog netlist conversion all the SD gates are replaced with their DD counterparts. Hence, both signals of any output pair will have the same fan-out. If routed together as a pair, the differential nets will be exposed to the same wire load, and therefore, exhibit similar timing behaviour.

2.2 Placement and Routing

As in the logic synthesis case, tools for routing differential signals as a differential wire pair do not exist. Some of the currently available routers can route signals together at a specific distance from each other, as desired for differential pair routing, but, this feature can be applied only to few user-defined nets.

There is not much previous work available on differential routing; the existing solutions are based on a routing the differential pairs as one wider net, where the width of this “fat” wire is equal to the sum of the individual widths of each net and the spacing between them. This method was introduced in [4] to be used in multi-chip modules (MCM), and it was adapted to be part of a design flow in [5], in order to obtain secure hardware implementations of crypto algorithms against the differential power analysis (DPA) attacks.

The inputs to the placement-and-routing (P&R) step are the Verilog netlist consisting of SS-gates and a LEF file (Library Exchange Format by Cadence [6]) representing the “fat-wire” technology and the cell library, in which,

each gate has single-ended IO pins. These pins are defined as “virtual pins” located on a higher level of metal. The regular P&R flow is followed until a DRC clean and logically verified layout is obtained. The output of this step is a DEF file (Design Exchange Format by Cadence [6]) describing the final circuit of SS-gates and wide-wire interconnections. The next step is to run the script, which replaces each SS-cell with its counterpart from the fully differential DD library, and splits the “fat wires” into the two nets of regular wire width dictated by the original technology. Then, to complete the connections between the differential IO pins and the differential nets, the fully differential DD Verilog netlist is read and a corresponding connection mask is applied to between every pin pair and the so-called virtual pins. The final step is to verify the interconnection network by either running LVS or using an equivalency checker tool.

The proposed method involves a more detailed work with less constraints on the cell design compared to other methods in previous works, serving the goal of achieving a noise-immune design solution. During placement the tool is allowed to use any symmetry for the cells, which might save a lot of area. The method allows the designer to run clock tree synthesis, in fact, it does not prevent the tool to apply any ECO changes that might be necessary. Moreover, it can be applied to existing differential cell libraries with little additional work.

2.3 The Differential Cell Library

A fully differential cell library has been designed and characterized to be used in logic synthesis and P&R. The cells are based on current mode logic (CML), where, the operation is based on the principle of re-directing (or switching) the current of a constant current source through a fully differential network of input transistors, and utilizing the reduced-swing voltage drop on a pair of complementary load devices as the output. CML circuits have been introduced as very high speed design alternatives that offer robust operation, reduced power supply/common mode noise, and improved immunity against process variations [7]. The switching delays can be significantly reduced due to limited output voltage swing, while fully differential inputs and outputs contribute to improved noise immunity and robustness. In addition, the power dissipation of the MCML gate remains virtually independent of the switching frequency, which means that the power dissipation at higher operating frequencies is actually lower than that of an equivalent CMOS gate under the same output load conditions.

Figure 2 shows a generic 3-input CML gate. The transistor M1, driven by a fixed voltage (“nbias”), provides the current for the circuit, and the transistors M2 and M3, driven by another fixed voltage (“pbias”) act as resistors. The generic function of this structure is $Y = AB + A'C$ ($Y' = [A + B'][A + C']$), it corresponds to a multiplexer-operation. All the 2-input logic functions, including XOR operation and some of 3-input logic functions can be realized using only one generic CML gate, by assigning the input pairs to appropriate logic levels.

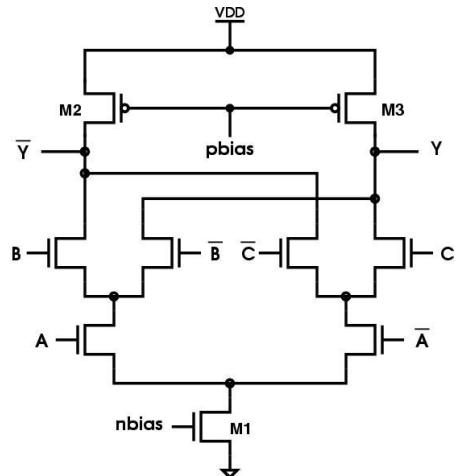


Figure 2 Transistor-level view of the generic CML gate.

The CML cell library consists of a limited number of basic logic gates, including buffers, latches and resettable flip-flops. Each cell is designed with six different drive strengths. Typical CML two-input gate delays are found to be between 90ps, where a full-adder delay is approximately 50ps.

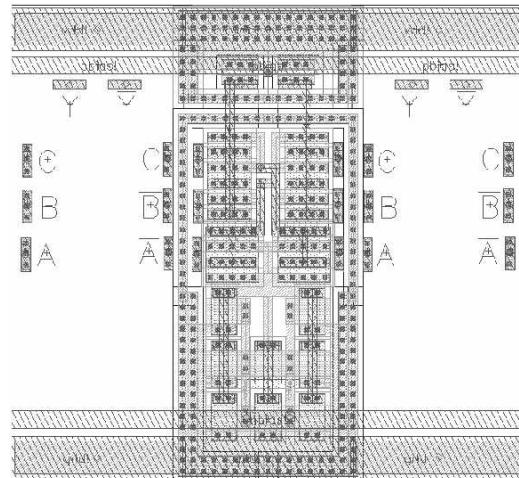


Figure 3 The layout-view of the generic CML gate, including two neighboring cells of the same type, which do not have all the physical layers visible.

The layout view of the generic CML gate is given in Figure 3, with two neighboring instances of the same cell together. The total size of the given layout is 22.0 um x 24.5 um, where one cell has a height of 22.0 um and a width of 8.5 um. The cells use up only the lowest two metal layers, hence, the rest of the upper levels are free for routing purposes. .

The main contribution of CML gates with respect to predictability of the proposed flow is based on the fact that there is virtually no change in current drawn by the gate,

even after switching of inputs. This eases overcoming some signal integrity problems like IR-drop and electromigration (EM), just by making them easier to calculate or to be known at the very first steps of the design flow.

3. HARDWARE IMPLEMENTATION

To test and to evaluate the proposed design flow, a test chip is produced which consists of three different realizations of the same RC4 block [9] as listed below:

1. RC4_ART_SER: Implemented using a commercially available single-ended CMOS standard-cell library.
2. RC4_CML_FDF: Implement using the full-differential cells, placed-and-routed according to the proposed flow.
3. RC4_CML_SDF: Implemented using the CML based fully differential library, same cell placement as in RC4_CML_FDF, without full-differential routing.

The layout of the test chip is shown in Figure 4. One key observation is the difference in size between the single-ended implementation ‘RC4_ART_SER’ (located at the top-left corner) and the differential implementations. RC4_ART_SER occupies an area of 400um x 400um, whereas the differential circuits have the dimensions of approximately 1mm x 1mm. This difference is caused mainly by the area difference of the cells from both libraries and of course the need for routing two nets instead of one. In return, it is expected that the fully differential version of the circuit exhibits significantly improved signal integrity characteristics, and hence, higher operating speed. Complete experimental characterization of the circuits will be done following the fabrication of the test chip.

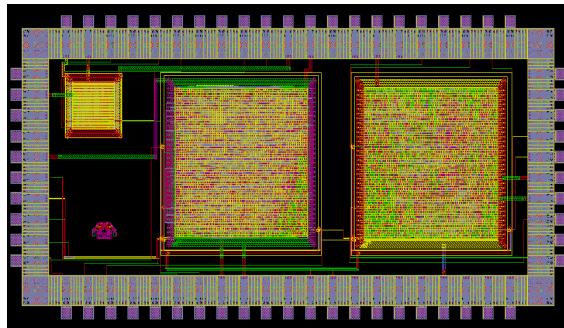


Figure 4 The top-level layout of the test chip designed with 0.18um CMOS technology.



Figure 5 A closer view to the layout of the RC4 block that is routed fully differential wiring. The cell layouts are omitted for better visibility of the differential routing.

4. CONCLUSION

In this paper a design flow for implementing fully differential designs is proposed. It is shown that following this flow leads to successful final layouts which are DRC and LVS clean. Next step in this work is to show that the signal integrity issues encountered with the deep sub-micron technologies can be decreased to an acceptable level.

5. REFERENCES

- [1] B.Zahiri, *Structured ASICs: Opportunities and Challenges*, Proceedings of ICCD'03, 2003.
- [2] P. E. Gronowski, W.J. Bowhill, R.P. Preston, M.K. Gowen, R.L. Allmon, *High Performance Microprocessor Design*, IEEE JSSC, vol. 33, Issue 5, May, 1998, pp. 676-686.
- [3] Synopsys, Inc., <http://www.synopsys.com>
- [4] Loy, J.; Garg, A.; Krishnamoorthy, M.; McDonald, J., *Differential Routing of MCMS-CIF: the ideal bifurcation medium*, Electrical Performance of Electronic Packaging, 2004. IEEE 13th Topical Meeting on 25-27 Oct. 2004 Page(s):135 – 138.
- [5] Tiri, K.; Verbauwhede, I., *A VLSI design flow for secure side-channel attack resistant ICs*, DATE 2005 Proceedings Page(s):58 - 63 Vol. 3.
- [6] Cadence Design Systems, <http://www.cadence.com>
- [7] P. Heydari, *Design and Analysis of Low-Voltage Current-Mode Logic Buffers*, 4th International Symposium on Quality Electronic Design, March 2003.
- [8] I. Hatirnaz, Y.Leblebici, *Twisted-Differential On-Chip Interconnect Architecture for Inductive/Capacitive Crosstalk Noise Cancellation*, Proceedings of Int. Symposium on System-on-Chip (SOC), Finland, 2003.
- [9] P. Hartmann, S. Witt, *Vergleichende Analyse und VHDL-basierte Implementation von Zufallszahlengeneratoren auf Chipkarten*, Studienarbeit, University of Hamburg, 2001.