

Via-Programmable Structured ASIC Fabric Based on MCML Cells: Design Flow and Implementation

Stéphane Badel, İlhan Hatırnaz, Yusuf Leblebici
EPFL-STI-IMM-LSM
Microelectronic Systems Laboratory
Station 11, 1015 Lausanne, Switzerland
e-mail: {ilhan.hatirnaz|stephane.badel|yusuf.leblebici}@epfl.ch

Elizabeth J. Brauer
Department of Electrical Engineering
Northern Arizona University
Flagstaff, AZ 86001-5600 USA
e-mail: liz.brauer@nau.edu

Abstract—This paper presents a regular layout fabric made of via-programmable MCML universal logic cells for structured ASIC applications and the associated design flow. The proposed structured ASIC fabric offers high speed operation, very high noise immunity, as well as low production cost due to the via-programmable properties of the universal logic cell. Implementations of a number of circuits are presented and the area/speed performances are compared with classical CMOS implementation using a commercial standard cell library in 0.18 μm CMOS technology.

I. INTRODUCTION

Structured ASICs are becoming an increasingly popular alternative for rapid, low cost realization of ICs, filling a gap between FPGAs and full-custom ASICs. They can provide a higher level of integration and increased performance compared to FPGAs, while cutting down the non-recurring engineering (NRE) costs and turnaround time compared to custom ASICs [1]. Structured ASICs are composed of a prefabricated array of standard building blocks, and their functionality is programmed via a number of customized layers. In addition, the regularity of the prefabricated structures allows better control of the problems associated with manufacturing variations.

In this paper, we propose an implementation of a cell fabric suitable for structured ASIC applications, where the basic building block is a via-programmable universal logic gate in MOS current-mode logic (MCML). The MCML design style has proven to offer good speed performance and addresses the noise immunity and crosstalk problems thanks to its differential operation [2]. Furthermore, the MCML logic style, in which logic functions are implemented with current-switching trees, allows the implementation of a wide range of logic functions with a small number of configurations. In comparison to earlier implementations of universal logic gates using MCML [3], which have the functionality of a 2-input MUX, we present an expanded universal cell which has the capability of implementing all 3-input Boolean functions as well as a significant subset of 4- and 5-input functions. Also, the power dissipation is about one order of magnitude lower than earlier designs to allow high density integration.

This paper is organized as follows : in Section II, we describe the cell that is used as a building block in our

structured ASIC approach. In Section III, we describe the design-flow to implement an RTL code into a regular tile of via-programmable cells, with fully differential routing. Next, in Section IV, implementation results are presented, and comparisons are drawn with the CMOS standard-cell implementation of the same designs. Some perspectives are provided in Section V, followed by the conclusion.

II. BUILDING BLOCKS AND ARRAY ARCHITECTURE

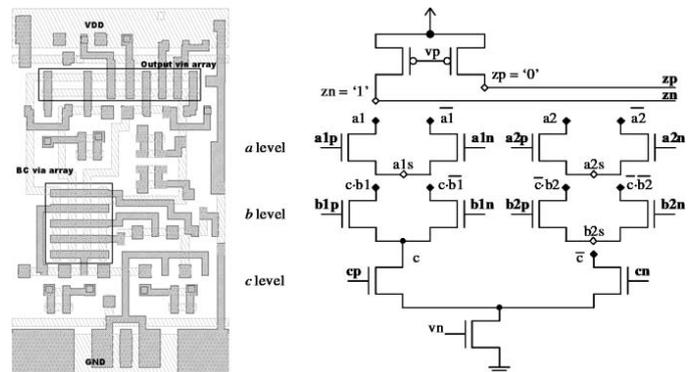


Fig. 1. The layout and the schematic view of the via-programmable cell.

In this work we used the via-programmable MCML universal logic gate (designed with 0.18 μm digital CMOS technology) described in [4], as the fundamental building block (Fig. 1). Here, the functionality of the cell can be easily customized by setting the appropriate via connections while preserving the same layout topology. In order to utilize this cell in a classical logic synthesis tool, a number of functions have been implemented by setting the via matrix accordingly, and each one was characterized for timing and power. The resulting library is composed of 17 functions, with up to 5 inputs and all based on the same basic cell layout (i.e. same area). Considering the fact that all inputs and outputs can be inverted at no additional cost due to differential signaling, this library effectively produces a wide range of functions available to the synthesis tool. Also, the library contains 3 types of flip-flops, including asynchronously resettable and scan flip-flops.

In order to obtain a regular array of identical cells, the placement grid is set to the size of a cell, and empty spaces are

filled with dummy cells. Metal1 and Metal2 layers are used for intra-cell connections, while the first via layer is used for customizing the cell functions. All metal layers above Metal2 can be used for inter-cell routing.

III. TOP-DOWN DESIGN FLOW

The proposed design-flow accommodates standard logic synthesis tools as well as place-and-route tools, and allows true differential routing that provides the benefits of higher noise immunity and reduced crosstalk. This design flow does not only target the designs using the MCML based structured ASIC library, but also designs based on other differential cell libraries that conform to a certain set of rules.

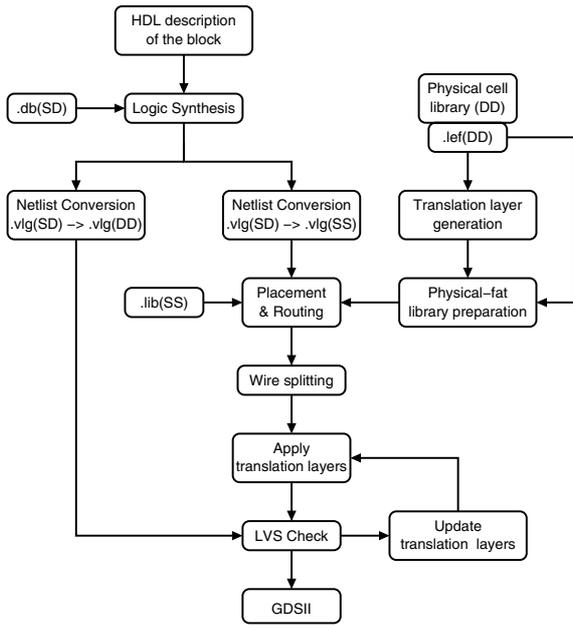


Fig. 2. The top-down synthesis and placement/routing flow.

A. Differential Logic Synthesis

The proposed design flow is given in Fig. 2. The components of this flow include 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 differentiality, it only describes the design in a single-ended manner.

Even though the cells are fully differential (complementary input and outputs), currently available synthesis tools are not able to provide mapping of nets to differential inputs pairs of gates from this differential library. To utilize the complementary nature of the differential cell outputs, a new synthesis library is extracted from the fully differential library. This new library consists of single-ended input and differential output (SD) gates. The logic synthesis tool is able to benefit from the differential outputs of the logic gates offered by the SD cell library, i.e., the tool 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 and then converted to a fully differential Verilog netlist using the netlist conversion scripts.

B. Differential 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 limited previous work available on differential routing; the existing solutions are based on 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 [5], [6]. A similar approach was followed for the routing of the interconnects in the proposed design flow.

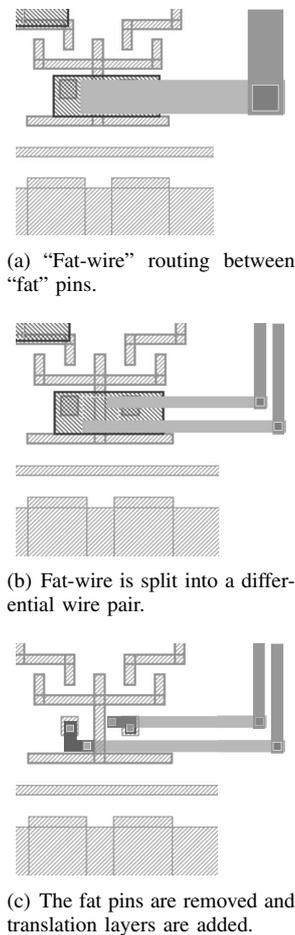


Fig. 3. Individual steps of wire splitting on a routed “fat” segment.

The Verilog netlist and a LEF file (Library Exchange Format) representing the fat-wire technology and the cell library are provided as the starting point of the placement-and-routing step. 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) describing the final circuit of single-ended-input and single-ended-output (SS) gates and wide-wire interconnections. The next step is to run a 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 (Fig. 3). The final step is to verify the interconnection network by either running LVS or using an equivalence checker tool. During placement the tool is allowed to use horizontal symmetry for adjacent cell rows, to reduce wiring length. 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.

IV. IMPLEMENTATION

To evaluate the efficiency of the proposed structured ASIC platform and the associated design methodology, a range of different circuits were synthesized using the MCML based universal logic gate library, followed by differential placement and routing as explained in the previous section. The first example is based on the realization of large-input majority decision units (number of input bits: 16, 32, 64 and 128). Figure 4 shows a close-up view of the regular cell array that is created at the end of the placement step. The *area utilization* rate was found to be approximately 90%, for all designs. The close-up view of a section of the array after the completion of differential routing is shown in Fig. 5, with a few differential wire pairs highlighted for easier recognition. The net-length histogram in Fig. 5 also shows that the vast majority of interconnects have a length of less than 50 μm , and only a very small fraction of the nets exceed 200 μm . This indicates that the interconnect delays shall not dominate the timing, in most cases. Table I provides a comparison of different majority decision unit designs with the MCML universal logic gate, and with a commercial CMOS standard-cell library, in terms of cell count and input-to-output delay. It can be seen that the designs based on MCML universal logic gates consistently produce lower delays, and that the cell count is also comparable to that of the CMOS standard cell design.

To further evaluate the efficiency of the approach, different versions of a Radix 4 complex FFT processor were implemented, with varying bit-lengths (from 16 bits to 256 bits) (Fig. 6). The FFT processor design is based on a public-domain VHDL source code from OpenCores [7]. The complete design flow was applied as described in earlier sections, from synthesis to differential placement and routing. The results indicate that the structured ASIC implementation remains competitive with respect to the CMOS standard cell implementation, both in terms of cell count and in terms of input-to-output delay (Table II). It is interesting to note that the cell count of the MCML solution drops well below that of the CMOS standard-cell solution, especially for larger bit-lengths. Finally, the significant advantage of the MCML-based design with respect to *power supply noise* generation is demonstrated

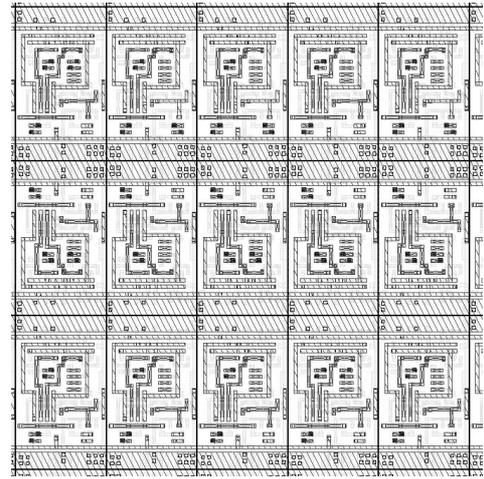


Fig. 4. The regular array layout using the MCML universal gate. This shows only a section of the prefabricated matrix array.

in Fig. 7. Here, the amount and the variation of power supply current of the 16-input majority decision unit is simulated for a large number of consecutive input vectors. It can be seen that the MCML-based structured ASIC implementation draws a nearly-constant amount of power supply current with variations of less than 5%, while the CMOS version of the same circuit produces significant current spikes that are responsible for power supply noise and substrate noise. With the source current of each cell set at 50 μA , the overall power dissipation of the circuit is comparable to that of the standard CMOS cells, especially at higher operation frequencies.

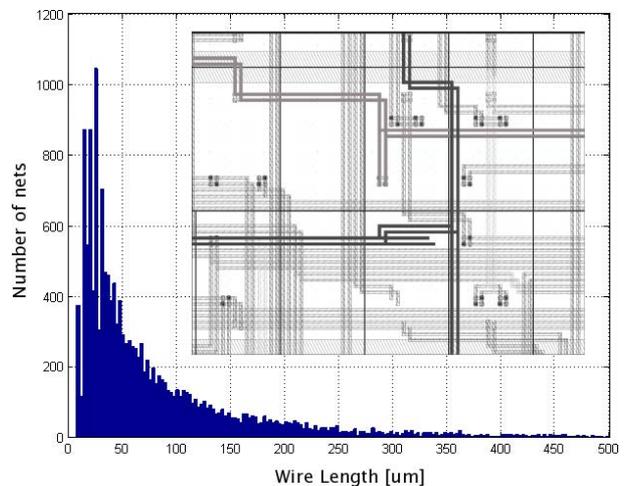


Fig. 5. Wire-length distribution for the 128-bit majority block realization. Inset: Detail of differential routing.

V. PERSPECTIVES

The universal gate proposed in this paper, together with the design flow, is a candidate for high performance structured ASIC applications. However, there are a number of improvements that can further enhance the capabilities. The array floorplan can be designed to contain not only cells but

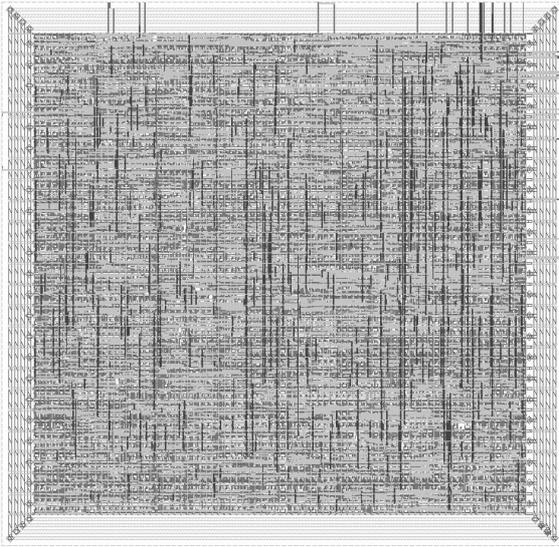


Fig. 6. The layout view of the FFT8 design using the differential design flow ($800\mu\text{m} \times 800\mu\text{m}$). Area utilization is above 85%.

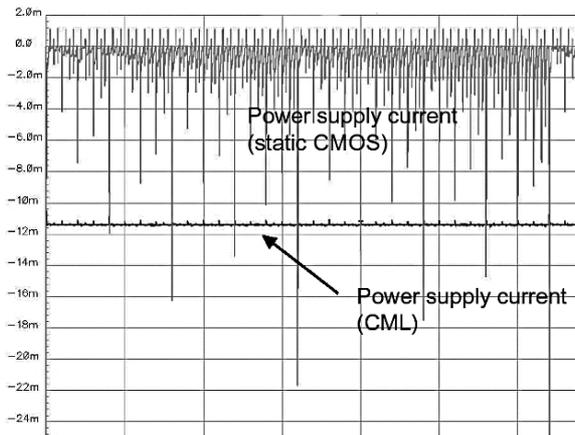


Fig. 7. Simulation results showing the current drawn from the power supply for MCML and for CMOS. The generated power supply noise in MCML is about two orders of magnitude lower.

also a number of buffers that can be inserted as repeaters to drive long interconnects. Although most metal layers were left fully customizable in this work, a fixed, via-programmable routing pattern could be designed on a number, if not all, of the metal layers. This would further reduce the masks costs, and also result in more regularity and thus more predictability. In this case, the wires can be characterized extensively since their environment would be known in advance. Also, cell characterization can take into account the surrounding wires.

VI. CONCLUSION

In this paper, a design flow was proposed which allows the implementation of regular fabrics using a via-programmable MCML universal logic module as building block. The proposed design flow solves the issues related to the differential nature of the cells, and allows true differential routing to

Bit length	MCML		CMOS	
	cell count	delay [ns]	cell count	delay [ns]
16	351	3.17	746	4.42
32	2077	8.63	2385	10.35
64	7448	16.83	5471	20.98
128	17305	33.91	13030	41.01

TABLE I

COMPARISON OF CELL COUNT AND DELAY FOR THE REALIZATION OF MAJORITY DECISION FUNCTIONS WITH MCML STRUCTURED ASIC ARRAY AND WITH CMOS STANDARD CELLS.

Bit length	MCML		CMOS	
	cell count	delay[ns]	cell count	delay[ns]
16	4674	1.20	3511	1.12
32	8704	1.40	7869	1.19
64	15958	1.70	16289	1.22
128	29944	1.94	34771	1.36
256	57201	2.37	71819	1.52

TABLE II

COMPARISON OF CELL COUNT AND DELAY FOR THE REALIZATION OF FFT DESIGNS WITH MCML STRUCTURED ASIC ARRAY AND WITH CMOS STANDARD CELLS.

exploit the full benefit of noise immunity and speed provided by the MCML cell.

The universal gate can be utilized as an atomic block in structured ASIC applications using the proposed design flow, where only the top metal and via layers are customized by the designers. In particular, it is well suited to a mixed-signal structured ASIC environment, and to applications which require high noise immunity.

The experimental results show a very high cell utilization rate, considerably smaller number of cells per design compared to standard cell CMOS realizations, and delay times that are comparable (or better) with respect to CMOS implementation. Considering the clear advantages of noise generation/noise immunity and significantly lower mask costs, this design platform can be utilized as a feasible option for high-performance ASICs.

REFERENCES

- [1] G. Xu, R. Tian, Z. Pan, and M. Wong, "CMP-aware shuttle mask floorplanning," in *ASPAC*, 2005.
- [2] M. Yamashina and H. Yamada, "An MOS current mode logic (mcml) circuit for low-power sub-GHz processors," *IEICE Trans. Electronics*, vol. E75-C, pp. 1181–1187, 1992.
- [3] S. Khabiri and M. Shams, "Implementation of MCML universal logic gate for 10 GHz-range in $0.13\mu\text{m}$ cmos technology," in *Proceedings of ISCAS*, 2004.
- [4] E. Brauer, İ. Hatirnaz, S. Badel, and Y. Leblebici, "Via Programmable Expanded Universal Logic Gate in MCML for Structured ASIC Applications: Circuit Design," in *ISCAS 2006*, May 2006.
- [5] K. Tiri and I. Verbauwhede, "A VLSI design flow for secure side-channel attack resistant ICs," in *Proceedings of DATE*, vol. 3, 2005, pp. 58–63.
- [6] J. Loy, A. Garg, M. Krishnamoorthy, and J. McDonald, "Differential routing of MCMs-CIF: The ideal bifurcation medium," in *Proceedings of ICCD94*, October 1994, pp. 599–603.
- [7] "http://www.opencores.org."