

# Implementation of Structured ASIC Fabric Using Via-Programmable Differential MCML Cells

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 very high noise immunity due to the differential operation, 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  $\mu\text{m}$  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 reducing 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 conclusions.

## II. BUILDING BLOCKS AND ARRAY ARCHITECTURE

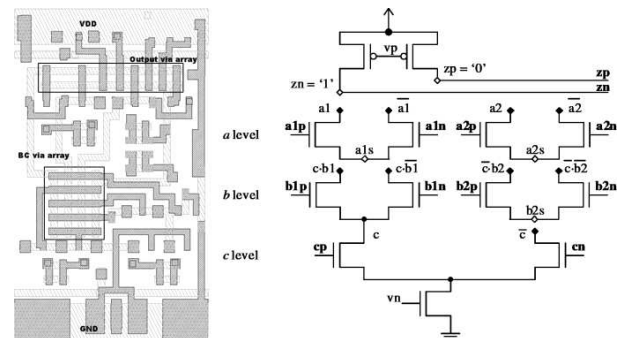


Fig. 1. The layout and the schematic view of the via-programmable cell.

In this work we use the via-programmable MCML universal logic gate (designed with 0.18  $\mu\text{m}$  digital CMOS technology) described in [4], as the fundamental building block. The transistor-level schematic and the corresponding layout of the via-programmable gate is shown in Figure 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

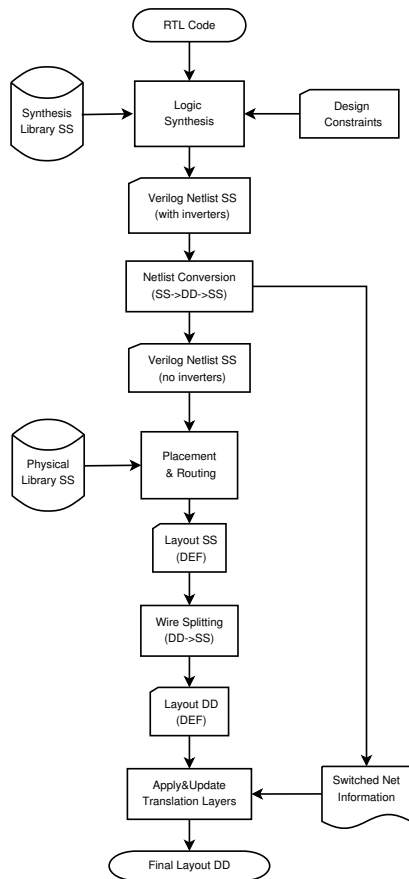


Fig. 2. The diagram of the proposed differential design flow. “SS” stands for single-ended input/output and “DD” stands for differential input/output.

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

We have developed a top-down design flow (Fig. 2) to accommodate the differential signals of the cells in the MCML library. We use standard logic synthesis and place-and-route tools with additional scripts to handle the differential signals. The input is a synthesizable HDL description of the design, together with its constraints. The code is not required to incorporate any knowledge of differentiability, i.e., include differential signals and etc. The final output is a regular layout, which consists of differential cells and matched differential nets.

#### A. Differential Cell Characterization

For a regular CMOS library, the delay is measured between the 50% transition points of the input and the output. The 50% transition point is the threshold level, above or below which, the output starts switching its value. In the case of differential signals, a similar approach is followed, the delay between the functional switching points is taken as the differential delay. Due to the differential signaling scheme in the circuit structures used in this work, the functional switching

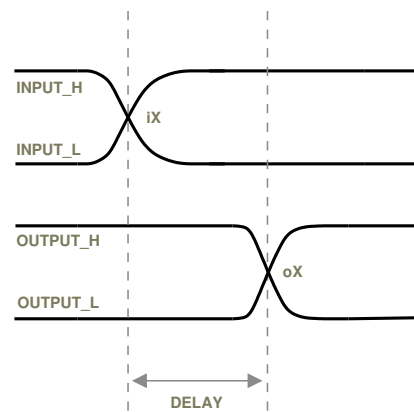


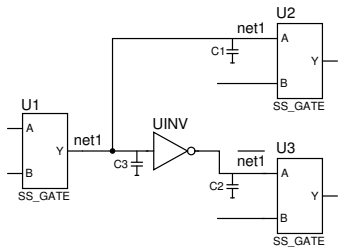
Fig. 3. The calculation of the delay of a differential gate.

points correspond to the zero-voltage crossings of the voltage difference at the input or at the output. This is illustrated in Figure 3; the crossing points  $iX$  and  $oX$  are the points where the difference voltage at the input and at the output is 0. The difference in time between the two points corresponds to the delay for that input-output pair.

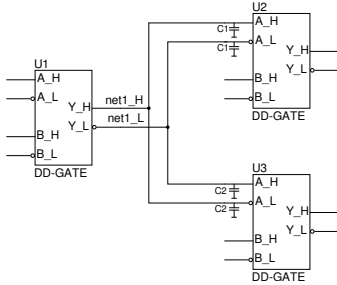
To the authors’s knowledge, there is no commercially available tool at this time that can perform cell library characterization of differential cells. Therefore, in our work we automated this step using scripts which take the extracted cell netlists and description of the cells functions, generate test vectors, run SPICE level simulations and write the results into the synthesis library file (.lib) [5]. In this file the delay and power numbers are stored in look-up tables, addressable by the load capacitance and the input rise/fall times. Setup and hold times are also extracted for the sequential cells in the library with respect to clock and data transition time. Input capacitances are also measured for each differential input port in the cells. The input capacitance of the equivalent single-ended gate is defined as the average of the capacitances of both complementary input, as it can be shown that the delay of an MCML gate is essentially linear with respect to that value. The cell library together with the its characterization data is compiled into an industry-standard .db file, labelled “*Synthesis Library SS*” in Figure 2. The SS library consists single-ended gates, which are extracted from the fully differential library (DD) by keeping only the non-inverted terminals of the gates. The logic synthesis tool takes the RTL code and the design constraints (both described in a single-ended manner) and outputs a Verilog netlist consisting of single-ended gates.

#### B. Differential Logic Synthesis

Due to the requirements of the synthesis tool, the synthesis library is required to include at least one cell with an inversion function. Because the mapping occurs between single-ended HDL and single-ended cells, having inverters in the gate-level netlist is inevitable. Since the output of this flow is a true differential netlist, the inversion can be obtained just by switching nets in a differential pair (Fig.4(b)). We introduced special inverters into the synthesis library, which have their



(a) Circuit with SS gates after synthesis before netlist conversion ( $C2 = C3$ ).



(b) Circuit with DD gates after SS-to-DD conversion.

Fig. 4. Removal of the special inverters.

input capacitance and input slew rate equal to their output capacitance and output slew rate, respectively ( $C3 = C2$  in Fig. 4(a)). These inverters exhibit zero area and delay, so that, when removed during the netlist conversion script (*SS- $\rightarrow$ DD-SS*), they don't affect the performance of the circuit estimated by the synthesis tool. The synthesis tool is also directed to not choosing an inverter for driving a load larger than its input capacitance, therefore, for obvious reasons, it will always choose the inverter with an input capacitance closest to the load capacitance. When the inverter is removed at a later stage, all delay, rise and fall time and load capacitance for this net remain nearly unchanged, so that all calculations made by the timing analyzer remain valid. The same script also replaces the single-ended cells with their differential counterparts and completes the missing connection (i.e., the complementary nets). The output of the netlist conversion is verified using a logical-equivalence checker tool.

As the *inverter-free* single-ended netlist is fed to the place-and-route tool, the differential netlist is used as reference after the detailed-routing is finished, as explained in Section III-C.

### C. Differential Placement and Routing

The goal of the differential P&R is to generate a routing scheme among the differential cells, where the signals of the same pair are routed at a fixed spacing for the entire design. There are several reasons for doing that. First, a large amount of crosstalk noise can be eliminated by using differential signals, provided both wires are subject to the same noise. This can be guaranteed by keeping the two wires as close as possible to each other. Next, a differential routing allows to better match the parasitics on each wire of the pair. This allows to reduce the power supply noise, as it is minimized when the

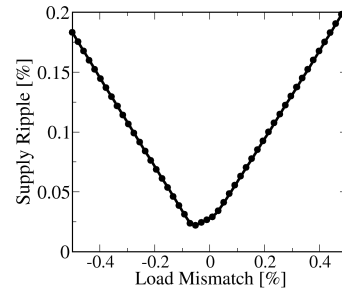


Fig. 5. Simulated current ripple magnitude on the power supply as a function of load mismatch between two wires of a differential pair.

load is perfectly matched on both wires (Fig. 5). Third, to a lower extent this allows to reduce the gate delay, as it can be shown that, for an MCML gate, the delay is minimized when loads are matched.

Only a few commercial routing tools can match the routing of differential nets. These routers are intended for top-level routing, where the design blocks are connected together. But they do not perform satisfactorily when all the differential nets of cell-based design (as in our case) are to be matched each other, mainly due to the higher density.

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 [6], [7].

We also used the "fat" wire approach to route matched differential nets. The gate-level Verilog netlist and the *physical library SS* representing the fat-wire technology and the single-ended cells, are provided as the starting point of the place-and-route step. The conventional 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 layout of single-ended (SS) gates and fat-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. 6). The final step is to verify the interconnection network by either running LVS or using an equivalence checker tool.

## 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 7 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

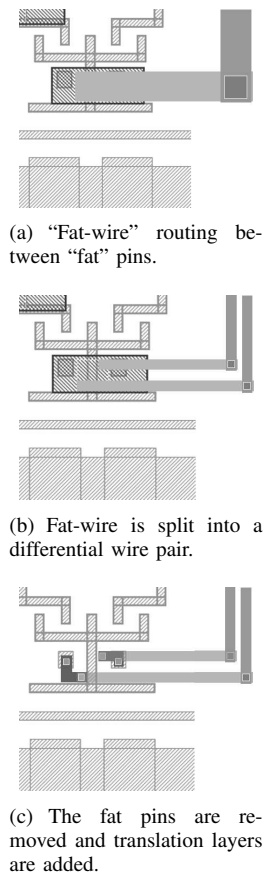


Fig. 6. Individual steps of wire splitting on a routed "fat" segment.

designs. The close-up view of a section of the array after the completion of differential routing is shown in Fig. 8, with a few differential wire pairs highlighted for easier recognition. The net-length histogram in Fig. 8 also shows that the vast majority of interconnects have a length of less than  $50 \mu\text{m}$ , and only a very small fraction of the nets exceed  $200 \mu\text{m}$ . This indicates that the interconnect delays do 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, the second example is a Radix 4 complex FFT processor with varying bit-lengths (from 16 bits to 256 bits). The FFT processor design is based on a public-domain VHDL source code from OpenCores [8]. 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

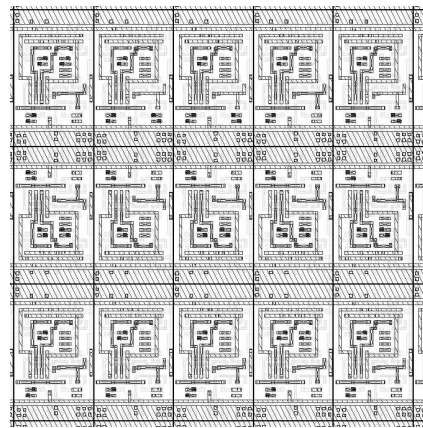


Fig. 7. The regular array layout using the MCML universal gate. This shows only a section of the prefabricated matrix array.

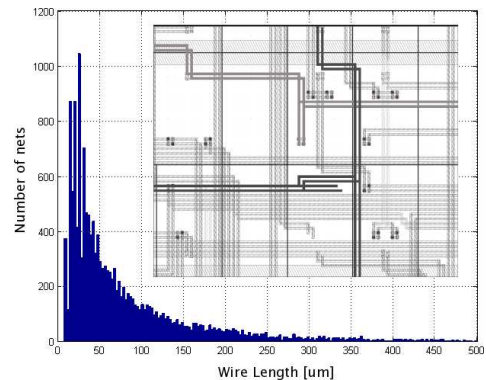


Fig. 8. Wire-length distribution for the 128-bit majority block realization. Inset: Detail of differential routing.

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 in Fig. 9. 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  $50 \mu\text{A}$ , the overall power dissipation of the circuit is comparable to that of the standard CMOS cells, especially at higher operation frequencies.

## V. PERSPECTIVES

The universal logic gate, together with the design flow, proposed in this paper 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 also a number of buffers that can be inserted as repeaters to drive long interconnects. Although most metal layers were left

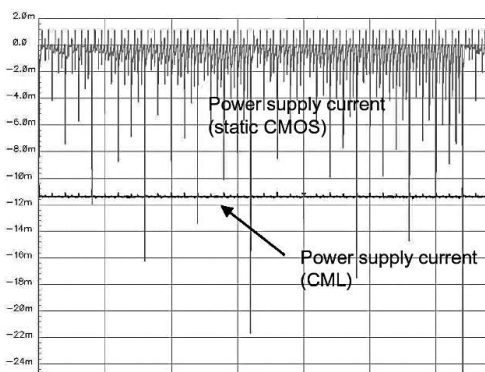


Fig. 9. Simulation results showing the current drawn from the power supply for MCML and for CMOS. The generated power supply noise in MCML is about two orders of magnitude lower.

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 OF MAJORITY FUNCTIONS WITH MCML STRUCTURED ASIC ARRAY AND WITH CMOS CELLS.

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 design flow solves the issues related to the differential nature of the cells, and allows true differential routing to 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

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 OF FFT DESIGNS WITH MCML STRUCTURED ASIC ARRAY AND WITH CMOS CELLS.

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 $\mu$ 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] S. O. Documentation, *Library Compiler and Data Preparation*, 2005th ed.
- [6] 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.
- [7] 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.
- [8] "http://www.opencores.org."