

MULTI-OBJECTIVE OPTIMIZATION OF MCML CIRCUITS USING A GENETIC ALGORITHM.

Roberto Pereira-Arroyo, Pablo Alvarado-Moya, and Wolfgang H. Krautschneider¹

Instituto Tecnológico de Costa Rica,
¹Technische Universität Hamburg-Harburg

rpereira@itcr.ac.cr

ABSTRACT

This paper introduces the Pareto front as a useful analysis tool to explore the design space of MOS Current Mode Logic (MCML) circuits. A genetic algorithm (GA) is employed to automatically detect this front in a process that efficiently finds optimal parameterizations and their corresponding values in an aggregate fitness space.

As an example of the flexibility of this design automation approach, the results for an optimized fundamental inverter logic gate are presented. Measures of the power consumption, propagation delay and output voltage swing are used as fitness functions, since the problem is treated as a multi-objective optimization task.

Index Terms— Genetic algorithms, MOS current mode logic, Multi-objective optimization, Pareto front.

1. INTRODUCTION

THIS document introduces a novel automated optimization strategy that is applied for designing MOS Current Mode Logic (MCML) circuits, taking advantage of the power and versatility of Genetic Algorithms (GAs). Other approaches such as [1] have been published, which tie the optimization problem to the topology of the circuit and to its parameters, making necessary a relatively exhaustive search of the parameter space. Genetic Algorithms, on the other hand, work at a higher abstraction level in which specific information about the circuit being optimized is not required, it only receives a set of fitness values (e.g. real numbers), representing circuit metrics such as delay, power consumption, area, etc. Other metrics or types of digital or analog circuits can also be defined by the user.

The proposed optimizer uses the genetic algorithm named PESA (Pareto Envelope-based Selection Algorithm) [2], and relies on a standard circuit simulator (e.g. Spectre,

Spice and alike) to deal with the complexity of physical MOS transistor parameters and circuit topology. Circuit parameters like supply voltage, width, length of transistors, etc., are generated by the genetic algorithm and passed to the simulator, where the computation of the fitness values takes place. Thus, the designer can change either the optimization algorithm or the simulation models without much effort.

This paper begins with an overview of MCML circuits and the multiple parameters that affect their behavior. Section 3 gives an introduction to the basic concepts regarding the Pareto front, the architecture of our optimizer and how we employ it for circuit parameter search and optimization. Section 4 presents the experimental results obtained. The conclusions will be summarized in Section 5.

2. MCML CIRCUITS OVERVIEW

MCML is a circuit technique that has been used in applications of high-speed, mixed signal environments due to its reduced switching noise, immunity to common-mode noise and, especially, because its power consumption does not increase with the frequency of operation [3], whereas in standard CMOS circuits, power consumption increases linearly with frequency.

The fundamental MCML inverter/buffer is shown in Fig. 1. MCML circuits have three main components: PMOS transistor loads, one or more differential pairs depending on the number of logic inputs and a constant current source, controlled by the voltage V_{bias} . All logic inputs and outputs are fully differential. The circuit operation is based on current steering, i.e. the tail current produced by the transistor M_{bias} is steered into one of the branches depending on the differential inputs. This current develops a resistive voltage drop at the active load of the conducting

branch, while in the non conducting branch the output voltage is pulled to V_{dd} , thus producing complementary outputs.

For a single logic gate, its Delay and Power are given by [4]:

$$D_{MCML} = C(\Delta V / I) \quad (1)$$

$$P_{MCML} = I \times V_{dd} \quad (2)$$

where C is the load capacitance, I is the tail current and ΔV is the output voltage swing. Equation (1) indicates that the propagation delay can be reduced by lowering the voltage swing, decreasing the load capacitance or increasing the tail current. However, from (2) it is seen that increasing the tail current directly impacts the power consumption.

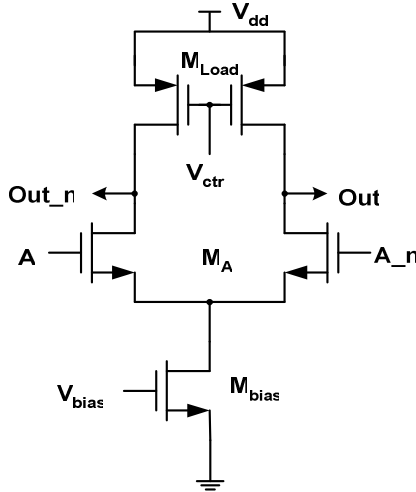


Fig. 1. MCML inverter/buffer. Transistors at the differential pair and at the active loads have identical dimensions. Therefore their naming as “ M_A ” and “ M_{Load} ”, respectively.

If the circuit shown in Fig. 1 is operating in the mid swing point of its voltage transfer curve, the currents in the two branches are equal to $I/2$, both transistors in the differential pair are in saturation and their currents can be expressed as [1]:

$$I/2 = \frac{\mu_0 C_{ox} (W/L)_A}{2} \times \frac{(V_{GSA} - V_t)^2}{1 + (U_d + \frac{V_{GSA} - V_t}{E_c L})} \quad (3)$$

where U_d is the mobility degradation coefficient, E_c the critical electric field for velocity saturation, μ_0 the permeability of vacuum, C_{ox} the oxide capacitance per unit gate area, V_t the threshold voltage and $(W/L)_A$, V_{GSA} are the width, length and gate-source voltage for transistor

M_A , respectively.

MCML design is a complex task as its objective metrics are interdependent, and numerous circuit parameters have an effect on these metrics. In the example shown in Fig. 1, nine circuit parameters can be varied by the designer, namely: (W/L) for M_{bias} , M_A and M_{Load} plus V_{dd} , V_{ctrl} and V_{bias} .

Fig. 2 shows an example of a 2-level MCML circuit. It has been shown in [6] and [13] how optimizing 2-level gates becomes even more complex, as more parameters come into play, making its design by hand calculations a task of little practical use. Therefore it is apparent the need for an optimization strategy that is automated and independent of circuit topology.

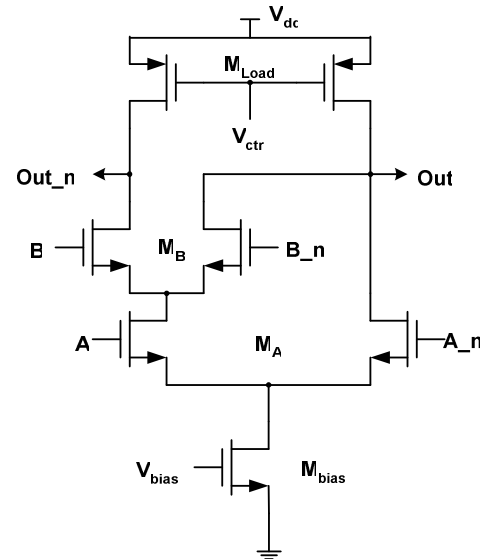


Fig.2. An example of a 2-level MCML gate, implementing both Nand/And logic functions.

3. GENETIC ALGORITHM FOR MULTI-OBJECTIVE CIRCUIT OPTIMIZATION

Genetic algorithms have previously been used in circuit optimization [7]. In that research, however, the optimization carried out was not multi-objective as only a single measure of fitness was considered, obtained by a linear combination of three circuit performance metrics. In our approach, multiple fitness values are independently computed and fed to the genetic algorithm for their evaluation. Multi-objective optimization both using a deterministic method [10] and a genetic algorithm [9], [11], [14] have been proposed for analog circuit design and, although very similar in practice, our solution

stemmed independently from the optimization of digital circuits.

Recent efforts to develop optimization strategies for MCML circuits have already been published [1], [4], [5], [13], although these approaches are based on deterministic algorithms (gradient followers) which are prone to be trapped in local minima.

3.1 Evaluating circuits using the Pareto Front

The aggregate fitness function F for a circuit A with the parameterization \mathbf{u} , evaluated using a reference data G is defined as

$$F(\mathbf{A}_\mathbf{u}) = \Phi(f_1(\mathbf{A}_\mathbf{u}), f_2(\mathbf{A}_\mathbf{u}), \dots, f_n(\mathbf{A}_\mathbf{u})) \quad (4)$$

with the individual fitness functions $f_i(\mathbf{A}_\mathbf{u})$ defined to increase monotonically with the fitness of some particular aspect of the circuit's behavior. For example in our MCML inverter, fitness values will be directly related to its output voltage swing and inversely related with its power consumption or its propagation delay.

The functions f_i span a multidimensional fitness space, where each point represents the performance of a circuit parameterized with one point \mathbf{u} in a parameter space. The general form of Φ is assumed unknown, but it has to increase monotonically with increasing values of all fitness functions f_i . This condition ensures that a point in the fitness space can be considered fitter than all other points with smaller values in all dimensions. In Fig. 3, for example, the point q_1 is fitter than the point q_4 and all other elements within the gray rectangle. In this context, the point q_1 is said to dominate q_4 . All non-dominated points in a set define the Pareto front of that set. In the example of Fig. 3 this front is defined by the points q_1 , q_2 and q_3 . Choosing a parameterization that is not in the Pareto front is always a bad choice, since there is another point on the front with a better aggregate fitness.

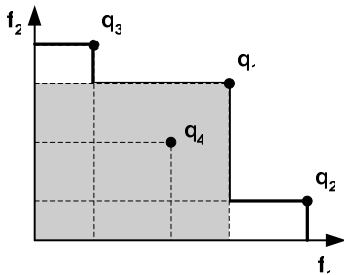


Fig.3. Pareto Front. The point q_1 dominates the region highlighted with a gray rectangle. Dashed lines delimit the dominated regions of the points q_2 , q_3 and q_4 . The thick solid line segments represent the Pareto front for the four

points.

The previous concepts can be expressed mathematically using the following equation:

$$\hat{P} = \left\{ \left\langle \mathbf{u} \in \mathbb{P}_A, \mathbf{f}(\mathbf{A}_\mathbf{u}) \right\rangle \mid \neg \exists \mathbf{v} \in \mathbb{P}_A : \mathbf{f}(\mathbf{A}_\mathbf{v}) \succ \mathbf{f}(\mathbf{A}_\mathbf{u}) \right\} \quad (5)$$

where \hat{P} is the Pareto front, \mathbf{f} is the vector of fitness functions $[f_1, \dots, f_n]^T$ and \mathbb{P}_A is the parameter space of circuit A . The partial ordering relation " \succ " on \mathbf{f} describes the domination property and is defined as:

$$\mathbf{f}(\mathbf{A}_\mathbf{v}) \succ \mathbf{f}(\mathbf{A}_\mathbf{u}) \Leftrightarrow \forall i : f_i(\mathbf{A}_\mathbf{v}) \geq f_i(\mathbf{A}_\mathbf{u}) \wedge \exists i : f_i(\mathbf{A}_\mathbf{v}) > f_i(\mathbf{A}_\mathbf{u}) \quad (6)$$

Any algorithm that finds the Pareto front for a set of fitness points implements (5) and (6). Since the parameter space \mathbb{P}_A usually contains an infinite number of parameterizations, the next problem consists in choosing a representative set of samples from \mathbb{P}_A , such that their Pareto front can be assumed to be a reliable approximation of the exact front extracted for the complete space.

A naive approach would be to regularly sample the values of each parameter, since the number of necessary evaluations would increase exponentially with the number of parameters. For example, a circuit with seven parameters (design variables), each sampled five times, would require $5^7 = 78125$ evaluations.

To avoid this brute-force parameter search, here the multi-objective evolutionary algorithm PESA is employed. This genetic approach suppresses the computation of useless parameterizations and concentrates the analysis on those regions of the parameter space that provide promising results. Even if this algorithm also discretizes the parameter space, the resolution used for each parameter can be as high as necessary, without the menace of an exponential explosion of the search space.

The number of evaluations required is then proportional to the number of bits used to represent a parameterization.

3.2 CAD tool architecture

The aim of our tool is to generate the Pareto front of a given cell, i.e. the set of all non-dominated parameterizations, since they represent the best individuals we are looking for. This information can be used later on to find the desired operating point of a circuit (choosing a specific individual), depending on the amount of resources available for the designer. A block diagram of the tool is shown in Fig. 4.

The functionality of the tool is divided into two well-

defined and independent processes: a circuit representation phase and an optimization phase.

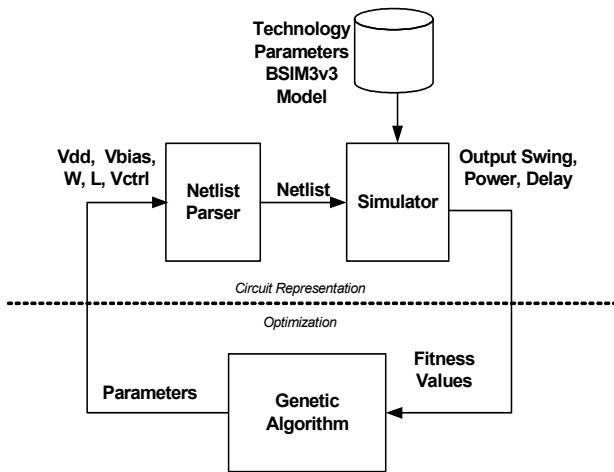


Fig.4. Genetic Circuit Optimizer Architecture. From the optimization point of view, both fitness values and parameters are just numbers, whereas for the circuit representation these numbers mean physical and geometrical variables.

In the current version of our tool, the circuit representation is captured with a Spice-like netlist and fed to the Spectre™ circuit simulator, where the computation of the specified performance metrics takes place. On the other hand, the core of the optimization process is based on the LTI-LIB [8], an open source library originally intended for image processing research [12], which provides the implementation of the PESA algorithm and the generation of the Pareto front. In Fig.4, the thick dotted line illustrates this separation of tasks. Furthermore, the optimization and representation processes are communicated through TCP/IP sockets, which allows for each process to reside on different machines or operating systems.

4. RESULTS AND DISCUSSION

Figure 5. shows a Pareto front, in three dimensions, of a MCML inverter/buffer as the one presented in Fig.1. This particular front contains 2500 individuals (parameterizations) and was generated by the PESA genetic algorithm. This figure shows the design constraints that have to be made for fast circuits: reducing delay will increase power and reduce the voltage swing.

Figure 6 shows the projection onto the “Voltage Swing, Power” plane of the three dimensional front. These fronts are a graphical aid to the designer to quickly see the

tradeoffs between design metrics or fitness functions, in

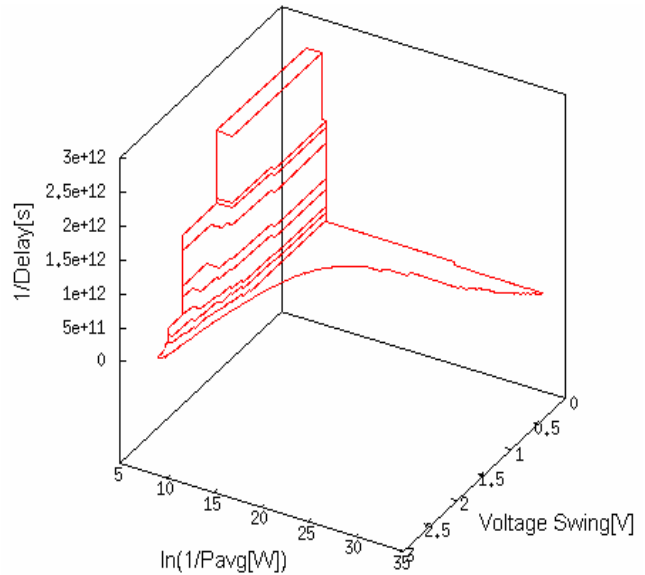


Fig. 5. Pareto front of a MCML inverter/buffer. Three design metrics were defined for this example: voltage swing, average power consumption and propagation delay, although others can also be defined.

GA’s terminology. For example, it is seen in Fig. 6 that the maximum voltage swing attainable by the circuit is approximately 2.7 Volts, at the expense of a higher power consumption.

A key question is: what values should the variables of a given cell have in order to satisfy some performance criteria? In MCML circuits design there is no obvious answer as, due to the amount of variables, many degrees of freedom exist, as it has already been explained in Section II. Fortunately, Electronic Design Automation (EDA) tools can help in searching for an answer.

Table I presents a selected set of five parameterizations for our MCML inverter, organized by decreasing values of the supply voltage. In other words, for this example we were interested in finding individuals for decreasing values of the supply voltage, regardless of power or delay. Since the supply voltage was not defined as a fitness measure, this information cannot be visualized from the graphical fronts, but instead it is obtained from the table of the whole Pareto set, which is generated by our tool. Table II shows the fitness values for the corresponding individuals. It is clear from the numbers presented in this table that these are non-dominated points, i.e. each individual is better than the others in at least one performance metric (fitness) and therefore, represent optimized parameter sets.

Using the information extracted from the Pareto front,

for example, individual numbered as 3 in the tables, is a circuit parameterization that operates at a supply voltage of 1.2V and has a power consumption of 3.95 μ W. It is also seen from the tables that individual number 4 works at a low voltage and very close to the sub-threshold region ($V_{bias}=0.52V$). Consequently, a design constrained to operate at low voltages would take advantage of parameterization number four. Again, the Pareto front does not show a single winner, is the designer who must pick up from the front the individual that suits best his/her needs.

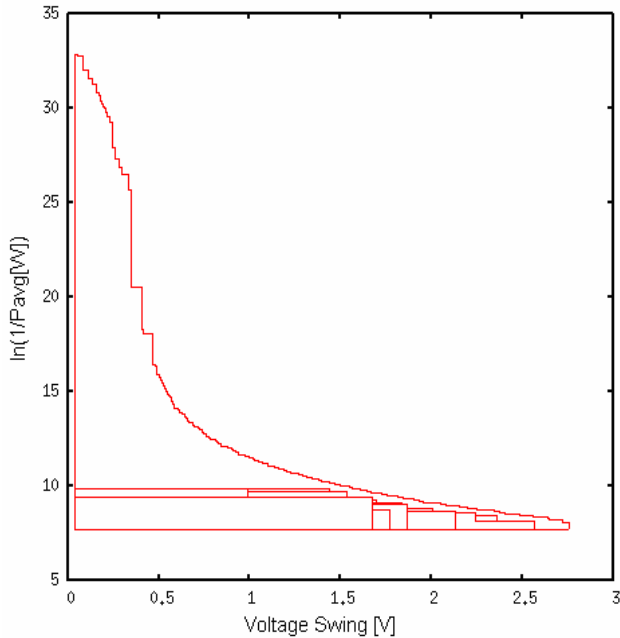


Fig.6. Projection of the Pareto front over the “Voltage Swing, Power Consumption” plane.

TABLE I
INDIVIDUALS CHOSEN FROM THE PARETO FRONT

Individual	1	2	3	4	5
V_{dd} [V]	3.3	2.5	1.2	0.64	0.3
V_{bias} [V]	0.3	1.0	0.7	0.52	1.3
V_{cntrl} [V]	0.6	0.8	0.8	0.72	0.23
W_{bias} [μ m]	6.0	4.12	4.12	5.83	2.92
L_{bias} [μ m]	1.37	0.43	2.1	1.77	0.50
W_a [μ m]	5.32	4.46	2.24	4.12	5.83
L_a [μ m]	1.30	0.51	0.83	1.93	1.06
W_{load} [μ m]	4.63	1.73	1.9	1.55	0.7
L_{load} [μ m]	1.22	1.61	1.85	0.75	0.35

TABLE II
FITNESS VALUES FOR EACH INDIVIDUAL

Individ.	$\ln(1/Power[W])$	Volt.Swing[V]	1/Delay[s]
1	17.66	0.43	55300
2	8.96	2.0	625272
3	12.44	0.8	2160270
4	19.04	0.37	45546
5	29.13	0.11	524

5. CONCLUSION

This work presented an automated design strategy for MCML circuits, intended to help designers find optimized parameterizations. In this paper, the optimization problem has been treated as multi-objective and as such, the Pareto front has been used as an instrument to find these optimized points. Initially, our EDA tool has been applied to MCML circuits, however, due to its flexibility and independence of circuit topology, it has also been tested with other types of digital circuits and to characterize an Operational Amplifier.

We expect to combine both simulation results and analytic expressions as a future refinement of this optimization methodology.

ACKNOWLEDGMENT

The authors thank Renato Rimolo-Donadio for his support in setting up the simulation environment.

This work was supported in part by the Costa Rican Ministry of Science and Technology (MICYT), the German Academic Exchange Service (DAAD) and Intel Corporation.

6. REFERENCES

- [1] Ayman H. Ismail, M. Sharifkhani and M.I. Elmasry. “On the Design of Low Power MCML Based Ring Oscillators” Electrical and Computer Engineering, 2004. *Canadian Conference on*, 2-5 May 2004 Page(s):2383 - 2386 Vol.4
- [2] Corne, D.W., Knowles, J.D. The Pareto-envelope based selection algorithm for multiobjective optimization. *Proceedings of the International Conference on Parallel Problem Solving from Nature (PPSN VI)*. (2000) 839-848
- [3] M. Mizuno, *et al.* “A GHz MOS, Adaptive Pipeline Technique Using MOS Current-Mode Logic” *IEEE Journal of Solid-State Circuits*, Vol 31. No. 6, June 1996, p. 784-791.
- [4] J. M. Musicer, J. Rabaey, “MOS Current Mode Logic for Low Power, Low Noise, CORDIC Computation in Mixed-Signal Environments,” *Proceedings of the 2000 International Symposium on Low Power Electronics and Design, 2000. ISLPED '00*. July 2000, p.102-107

- [5] Jason Musicer, "An Analysis of MOS Current Mode Logic for Low Power and High Performance Digital Logic," M.Sc. Thesis, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley.
- [6] Hassan, H.; Anis, M.; Elmasry, M. MOS current mode logic: design, optimization, and variability. Proceedings of IEEE International SOC Conference, 2004. Sept. 2004, Pages: 247 – 250
- [7] MacEachern, L.A. Constrained Circuit Optimization Via Library Table Genetic Algorithms. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems. 30 May-2 June 1999 Pages: 310 - 313 vol.6
- [8] LTI Image Processing Library: Developer's Guide. Version 29.10.2003. <http://www.techinfo.rwth-aachen.de>
<http://ltilib.sourceforge.net>
- [9] B. D. Smedt and G. G. E. Gielen. WATSON: Design space boundary exploration and model generation for analog and RF IC design. *IEEE Trans. CAD* 2003.
- [10] D. Mueller, *et al.* Deterministic approaches to analog performance space exploration (PSE). In *ACM/IEEE DAC 2005*.
- [11] G. Gielen, *et al.* Performance space modeling for hierarchical synthesis of analog integrated circuits. In *ACM/IEEE DAC 2005*.
- [12] J.P. Alvarado-Moya. Segmentation of color images for interactive 3D object retrieval. PhD thesis, Rheinisch-Westfälischen Technischen Hochschule Aachen, Germany. July, 2004.
- [13] MOS current mode circuits: analysis, design, and variability. Hassan, H.; Anis, M.; Elmasry, M. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on . Volume 13, Issue 8, Date: Aug. 2005, Pages: 885 – 898
- [14] B. D. Smedt and G. Gielen. HOLMES: Capturing the Yield-Optimized Design Space Boundaries of Analog and RF Integrated Circuits. In Proceedings of *IEEE DATE 2003*.