Leveraging QDI Robustness to Simplify the Design of IoT Circuits

Marcos L. L. Sartori, Rodrigo N. Wuerdig, Matheus T. Moreira, Sergio Bampi, Ney L. V. Calazans
PUCRS - School of Technology - Ipiranga Ave, 6681 - Porto Alegre - Brazil, 90619-900
PPGC-UFRGS - Informatics Institute - Bento Gonçalves Ave, 9500 - Porto Alegre - Brazil, 91509-900
{marcos.sartori,rodrigo.wuerdig,matheus.moreira}@acad.pucrs.br, bampi@inf.ufrgs.br, ney.calazans@pucrs.br

***Abstract*— Internet of Things devices require innovative power efficient design techniques that ensure correct operation in harsh environments, where using synchronous design can be challenging. The timing sign-off of synchronous circuits requires analysis and optimisation under multiple corners and operating modes. Considering that energy efficient circuits demand dynamic voltage ranges and harsh environments impose significant variations, design sign-off may become prohibitively expensive. An alternative is quasi-delay-insensitive asynchronous design, which presents robustness against timing variations, simplifying timing sign-off. This paper leverages recent developments in asynchronous circuits design automation to achieve higher degrees of energy efficiency using voltage scaling, while ensuring solid robustness to variability.**

I. Introduction and Related Work

With a compound annual growth rate (CAGR) of 39% in the 2018-2023 period, and an expected market size of US$520 billions in 2021 [1] the Internet of Things (IoT) is expected to dominate attention of high technology enterprises. This is to be compared e.g. with the International Data Corporation (IDC) estimate that in the same period the CAGR of the mobile phone market will reach -1.2% [2]. Since much of the expenditures in the IoT market goes into new solutions for inexpensively connecting devices to the Internet, this poses new challenges to the integrated circuit (IC) design research and development community. Some of the characteristics of these challenges are that IoT edge devices: (1) must be used in huge numbers; (2) must often be employed in harsh environments; (3) need not be implemented in latest technology nodes, but require high energy efficiency; (4) uses are very diverse and expected to evolve rapidly. This work focuses on design techniques for IoT edge nodes and associated devices.

Digital circuit design methods based on the synchronous paradigm most often do not achieve the highest level of power efficiency due to the global nature of clock signals. These signals must be distributed using clock trees and have delays from its sources to its sinks finely adjusted and carefully checked for every functional mode and for each combination of values for operating voltage and temperature and process corner. Such requirements impose a heavy burden on design (i.e. design closure is hard) and clock handling can easily spend 50% or more of the IC power budget. An alternative to these classic methods is to employ asynchronous design. However, even asynchronous circuits operating at nominal voltage may not fit green computing and circuit ageing requirements. Voltage scaling is a design/operation technique targeting power reduction and ageing mitigation. The larger the supply range a circuit can operate in, the more adequate it is for use in the IoT. The authors analysed the literature spectrum on wide supply range circuit proposals and selected two state of the art representative works, which will be compared to the approach described herein.

Pons et al. [3] describe the design and test of *icyflex*, a synchronous processor for operation at subthreshold supply voltages, but which can operate in a wide range of supply voltage choices and associated clock frequencies. Icyflex employs the TSMC $180\,\mathrm{nm}$ technology and has an operating voltage range from 0.43 to $1.80\,\mathrm{V}$, the latter being the nominal supply for the technology. This supply range results from use of latches and a careful design of subthreshold-aware modules (logic gates, memories and level shifters).

Hand et al. [4] propose Blade, an approach to design asynchronous error-resilient systems. Blade relies on a scheme similar to the synchronous timing error resilient architecture Razor [5], its successors (Razor-II, Bubble Razor, Razor Lite, etc.) and similar approaches, like Timber [6]. As Razor, Blade employs speculation on the time computations take to execute, and detects when this is too optimistic, producing errors. Error detection and correction are performed on the fly, relying on clever schemes of two or more reconfigurable delay lines, adaptable both at IC test and at run time. Since there is no global clock in Blade, timing errors are always local events, with no need to stop the whole system to allow recovery.

II. The Proposed Approach

Asynchronous design have long suffered from lack of support from methods, libraries etc. The authors have developed such support along the years, relying on established synchronous frameworks such as Cadence's or Synopsys's to design optimised asynchronous circuits. The approach employed herein comprises using ASCEnD cell libraries [7] to design circuits with the SDDS-NCL asynchronous template [8], synthesised and optimised with the Pulsar method [9]. The next Sections describe ASCEnD, SDDS-NCL and Pulsar, after an introduction to some asynchronous circuit concepts.

A. A Few Asynchronous Circuit Basics

Synchronous circuits rely on a global clock signal to provide a discrete common time reference. Typically, the clock is

978-1-7281-3320-1/20/$31.00 ©2020 IEEE

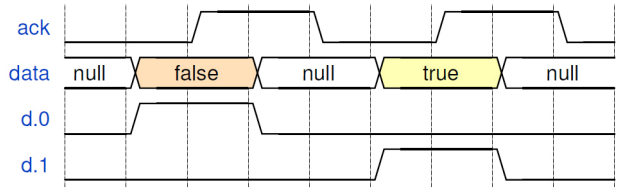
a wave with a period greater than the worst combinational logic delay in any path in the circuit between two consecutive temporal barriers, usually registers. All synchronous circuit registers simultaneously capture data (within a certain time window affected by skews and clock jitters), as determined by clock transitions at the registers. These characteristics guarantee that as soon as registers capture data, all combinational logic will have finished computing. Asynchronous circuits have no such single common time reference. To ensure correct operation, asynchronous logic blocks communicate with each other using *handshake channels* [10]. This approach eliminates the need for distributing a global clock. It also produces circuits that operate based on the average delay of combinational blocks, and not on the worst-case circuit path.

Handshake channel protocols comprise two distinct steps: (i) data *request*, where a transmitter announces data availability; and (ii) data *acknowledgement*, where a receiver acknowledges data reception, allowing the transmission of new data. The use of dedicated request/acknowledge signals separate from the data lines characterises what is known as the *Bundled Data* (BD) design style. BD allows simpler (and close to synchronous) data path implementations, at the expense of more complex timing assumptions. Since combinational logic transforming data must be transparent to the local handshake protocol [10], requests must arrive at the consumer only after all computations on channel data are concluded and results are ready at the consumer inputs, otherwise the latter may capture incorrect data. This poses a design challenge, as the request line may be required to be delayed with respect to data to guarantee correct operation. Delay lines are a must here.

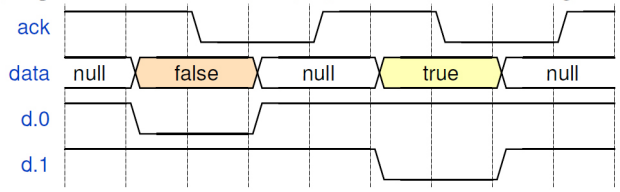
As an alternative, requests can be embedded within the data, using delay-insensitive (DI) data encoding. Circuits using such encoding type follow either Delay Insensitive (DI) or Quasi Delay Insensitive (QDI) design *templates*. The DI template class is the ideal one for maximum robustness, but it was demonstrated to be of little use [11]. QDI templates are considered the least compromise between robustness and practicality. QDI is a class of asynchronous circuit design templates that perform computation on DI encoded information. A QDI circuit design requires less restrictive timing assumptions than BD circuits. This makes QDI circuits less sensitive to process, voltage and temperature (PVT) variations and ageing. Examples of specific QDI design templates are Weak Conditioned Half Buffer (WCHB), Delay Insensitive Minterm Synthesis (DIMS) and Null Convention Logic (NCL) [10]. This work advocates the use of NCL. There are also several handshake protocols available and choosing one of these is part of producing specific QDI design techniques. Most often, QDI circuits rely on DI codes and on *completion detection* circuits to recognise data availability. Figure 1 depicts two specific QDI handshake protocols; other protocols exist.

B. Asynchronous Libraries and ASCEnD

The NCL template relies on the availability of a set of special logic gates, distinct from the ordinary Boolean gates such as ANDs, ORs and inverters. NCL gates mostly implement



(a) 4-phase return-to-zero (RTZ) QDI handshake, active-high *ack*.



(b) 4-phase return-to-one (RTO) QDI handshake, active-low *ack*.

Fig. 1: A sample of QDI handshake communication protocols.

threshold functions with hysteresis, and many of these are constructed in CMOS as a network of transistors with feedback. Although NCL gates can be built from ordinary gates, this is sub-optimal in terms of area and performance. More importantly, feedback lines drawn outside gates can have a strong impact on the robustness of the QDI template, due to the uncertainty of the timing characteristics for these lines, when generated by automated routing tools. The authors proposed a method called ASCEnD to implement asynchronous standard cell libraries in [7], and have since then produced several NCL gate libraries for commercial and predictive technologies. This work employs the ASCEnD-TSMC180 library with target on the TSMC 180 nm bulk CMOS technology, containing dozens of NCL gates and fully compatible with the ARM SAGE-X standard cell library for the mentioned technology. Compatibility with a commercial library enables relying on the latter for buffers, inverters and I/O pad circuits, reducing the asynchronous cell library development effort.

C. The SDDS-NCL Template

NCL is a well-established template to produce asynchronous QDI circuits [12]. However, logic design with gates presenting hysteretic behaviour is not supported by commercial synthesis tools like Cadence *Genus* or Synopsys *Design Compiler*. A typical hysteretic threshold NCL gate is described by a 3-valued function that outputs “0” (the NULL value) when all its inputs are “0”, outputs “1” when a weighted sum of “1”s at its inputs reaches the gate threshold and holds the previous output value otherwise (i.e. a hold or “H” value). Accordingly, NCL synthesis tools tend to be very specific, as is the case of the open source Uncle environment [13]. Unfortunately, Uncle and other efforts do not compete favourably with powerful commercial tools. Extending NCL, authors have proposed and developed the SDDS-NCL asynchronous QDI template [8] that allows circumventing this ignorance of commercial tools about hysteretic behaviour. One important insight to enable the use of commercial tools for asynchronous design was extending the NCL gate set to accept also the RTO protocol, besides the RTZ communication protocol of conventional NCL gates.

This typically doubles the size of libraries, but allows applying the unate function-based synthesis and optimisation methods of commercial tools seamlessly to QDI design.

D. The Pulsar Design Method

Recently, the authors proposed the Pulsar design flow for asynchronous QDI design [9]. On top of the SDDS-NCL QDI template and using ASCEnD libraries, the authors provide an infrastructure of guiding scripts that call Cadence tools like *genus*, coupled to the concept of *virtual functions* [14] associated to each NCL gate (RTZ or RTO). Virtual functions enable *fooling* the tools to believe they are synthesising/optimising conventional synchronous circuits. A virtual function is a Boolean function that outputs "1" when its corresponding NCL gate should output "1" and outputs "0" otherwise. Since the condition to have NULL at the gate output is pre-determined for all NCL gates and corresponds to a single line in the function truth table, the virtual function fully describes an NCL gate. After synthesising a circuit with virtual functions, results can be transformed back to the NCL gates of template SDDS-NCL. Synthesis errors do arise in this process, but these were proved to be trivial to detect and solve using low computational complexity error correcting scripts, run after each synthesis [15]. Note that the Pulsar design flow is an iterative process, able to look for implementations that reach a user-defined cycle time constraint.

III. Experiments and Results

To validate the proposed approach, the authors synthesised a pipelined 5-stage asynchronous 16-bit multiply-accumulate (MAC) unit. The synthesis environment relied on the Pulsar flow [9], and targeted cells from the ASCEnD library [7] developed for the TSMC $180\,\mathrm{nm}$ technology. The choice for this design was due to its reasonable complexity, which poses a non-trivial problem for the Pulsar flow while still being a simple design to trace. The circuit was signed off after timing analysis using the worst corner of the library (SS transistors, 1.62V and -125C). The target cycle time constraint was $20\,\mathrm{ns}$, which provides a good timing and area trade-off in the chosen technology while not over- or under-constraining the circuit.

After synthesis, authors exported the netlist of the circuit to Spice and performed analogue simulation to collect precise electrical characteristics. To do so, they relied on an analogue-mixed-signal (AMS) environment, which enabled the use of a digital testbench to generate stimuli and verify functionality. Simulation was carried using the Cadence Framework and the BSIM4 MOSFET transistor models provided by the foundry. The digital testbench simulates an ideal zero-delay environment, which provides new random data as soon as the circuit consumes previous ones and absorbs data coming from the circuit as soon as they become available, thus sustaining the circuit at its maximum throughput capacity. This guarantees that the circuit cycle time is only affected by its internal delays.

The MAC underwent multiple simulations using typical transistor models on a range of supply voltages below the nominal one. For each voltage level, power and mean cycle time values were measured. The asynchronous cycle time is the time between two consecutive results and the throughput is the inverse of the mean cycle time. Figure 2 shows the results, highlighting power as a function of the supply voltage, which decreases faster than throughput. As expected, voltage scaling yields increased power efficiency, which is clarified by Figure 3. This gain in power efficiency comes at the expense of a drastic drop in timing performance, as the supply voltage is scaled from the nominal supply ($1.8\,\mathrm{V}$) down to $500\,\mathrm{mV}$. There, it reaches an operating throughput one-hundred-fold smaller when compared to that achieved at nominal voltage. The low voltage point is at or very close to the Minimum Energy Point (MEP) of the circuit (see e.g. [16] for a discussion of the MEP). It is important to highlight that all operating modes at distinct voltages were achieved by signing off only at the worst corner of the employed library.

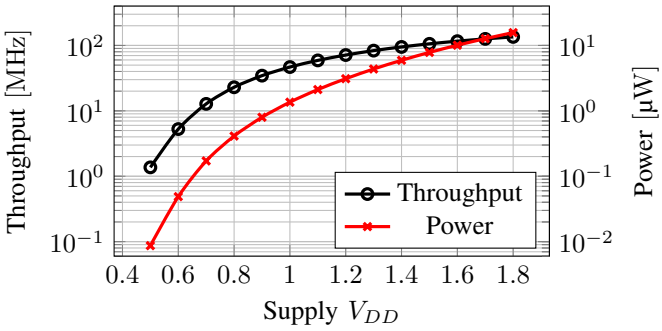


Fig. 2: MAC performance and power under voltage scaling.

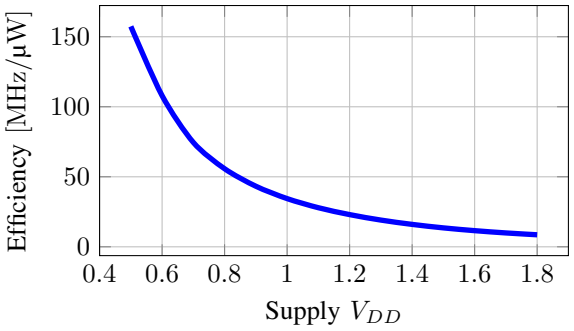


Fig. 3: MAC power efficiency analysis under voltage scaling.

To analyse how global process variations impact circuit operation at different supply voltages, the circuit went through Monte Carlo (MC) analysis using statistical parameter variations provided by the foundry. For each voltage level, the circuit was evaluated under 1000 different variation scenarios for process effects. For each simulation, the digital testbench computes the individual cycle times and evaluates whether the circuit yields correct computational results (i.e. if it operates correctly). Experiments indicate that the MAC yielded correct results under a wide range of supply voltages and under multiple process variation conditions. The cycle time distribution subject to process variation is depicted in Figure 4.

This result shows that as supply voltage reduces, circuit cycle time becomes more sensitive to process variation, which is evidenced by a significant increase in the standard deviation of the timing distribution obtained by the MC samples. This sensitivity can be quantified by the variability coefficient (VC), depicted in Table I. It corresponds to the ratio of standard deviation to the mean cycle time, calculated from the MC simulation results. In the technology under analysis, the VC becomes high at $500\,\mathrm{mV}$ supply, near the threshold value of FETs, as discussed next.

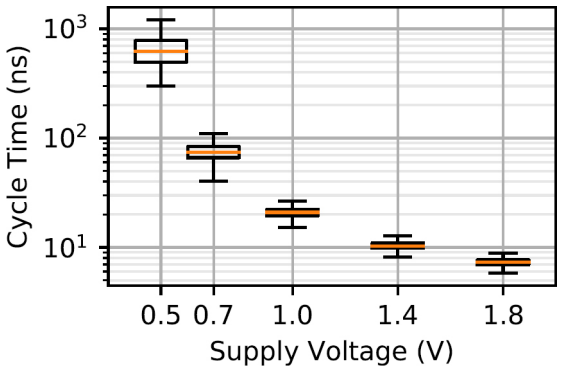


Fig. 4: MAC cycle times under process and supply variations.

0.5 V	0.7 V	1.0 V	1.4 V	1.8 V
0.396	0.1878	0.1079	0.08356	0.07341

TABLE I: Sample cycle time variability coefficients.

Analogue simulations show that the circuit behaves incorrectly only for supply voltages below $500\,\mathrm{mV}$. Note that the cell library used here was designed for guaranteed operation at $1.8\,\mathrm{V}$ $(+/-10\%)$ by the ASCEnD characterisation process. One possible cause that can be advanced for the circuit malfunction at the mentioned voltages is the asymmetry between the PMOS and NMOS threshold voltage changes. To verify this, the threshold voltages of both transistor types were analysed over a range of gate and drain voltages and their values are depicted in Figure 5. The graph displays a PMOS-to-NMOS threshold asymmetry that can lead to malfunction under subthreshold voltages, as transistor width sizes are not properly selected for near- and subthreshold operation.

IV. Conclusions, Ongoing and Future Work

This work explored the robustness of the QDI design in general and of the SDDS-NCL template in particular to voltage scaling and process variations. Results show that QDI circuits are able to dynamically withstand the increased delay variability at quite low voltage regimes. The ASCEnD library enables aggressive dynamic voltage scaling of designs constructed to operate at nominal voltage without requiring any associated frequency scaling schemes, which streamlines the design of low power and ultra low power circuits. Results demonstrate an eighteen-fold increase in power efficiency in a circuit operating stably at 27% of its nominal voltage.

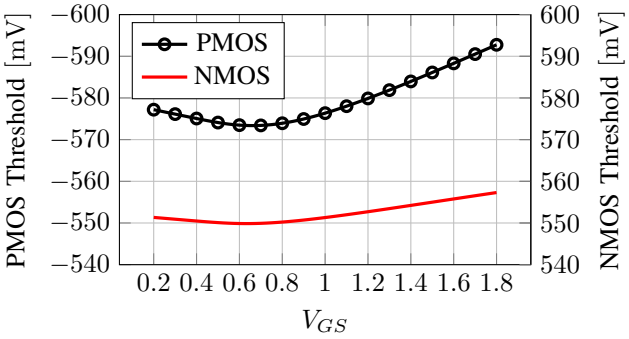


Fig. 5: TSMC 180 nm transistor threshold variation with V_{GS}.

Experiments also revealed limitations when using ASCEnD digital libraries for sub or near-threshold operation regimes. But that only occurred far away from the guaranteed supply, a merit credited to the use of QDI design templates. Circuit malfunction due to threshold voltage asymmetry on subthreshold levels highlights the importance of eventually developing specific standard-cell design for weak-inversion operation if the target is ultra low power operation. Blesken et al. [17] proposed an multi-objective optimisation process for transistor sizing directed to subthreshold operation that could be adopted to improve the design of ASCEnD standard cell libraries. Other problems might arise on subthreshold operation due to reduced noise margins and threshold asymmetry, requiring further study. Analysis and improvement of asynchronous gates for subthreshold operation on other process nodes, such as $130\,\mathrm{nm}$, $65\,\mathrm{nm}$ and below are ongoing work.

It is useful to compare the results achieved here with those in the selected works cited in Section I, latch-based synchronous subthreshold design [3] and Blade [4]. Pons et al. describe a design process leading to operation in a supply range very similar to that reached in the experiments above in a same technology node, but they rely in a specific cell library, designed to target subthreshold operation. The approach here uses less restrictive, conventionally designed libraries. Also, the synchronous approach requires a tight match between supply voltages and clock frequencies, demanding dynamic voltage and frequency scaling subsystems. QDI circuits need no clock, automatically adapting to voltage scaling. Reduced computation rates slow down local handshake controllers.

Blade is a BD approach where local handshake controllers have two reconfigurable delay lines, with their delays sum being constant. Blade reaches best performance with timing-error rates near 30% [4], increasing throughput without too much error-correction overhead. Voltage scaling requires adjusting delay elements to keep this rate, or even redefining a new optimal error rate. This is again more effort than QDI circuits require. Of course, even QDI circuits may require adjustments to advance into deep voltage scaling, either by the insertion of delay elements in carefully selected circuits nodes [18] and/or through the use of specifically designed subthreshold cells [19]. This is a required future work.

ACKNOWLEDGEMENTS

This research was partially funded by Brazilian government funding organisms CNPq (grants no. 200147/2014-5 and no. 312917/2018-0) and FAPERGS (grant DocFix 18/0558-4). Authors also acknowledge the partial support of HP Brazil.

REFERENCES

[1] I. Lee, "The Internet of Things for enterprises: An ecosystem, architecture, and IoT service business model," *Internet of Things*, vol. 7, no. 3, pp. 1–12, Sept. 2019.

[2] A. Scarsella and W. Stofega. (2019) Worldwide Mobile Phone Forecast, 2019–2023. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=US44916519.

[3] M. Pons, T.-C. Le, C. Arm, D. Séverac, J.-L. Nagel, M. Morgan, and S. Emery, "Sub-threshold Latch-based icyflex2 32-bit Processor with Wide Supply range Operation," in *European Solid State Device Research Conference (ESSDERC)*, 2016, pp. 33–36.

[4] D. Hand, M. T. Moreira, H. H., D. Chen, F. Butzke, M. Gibiluka, M. Breuer, N. L. V. Calazans, and P. A. Beerel, "Blade - A Timing Violation Resilient Asynchronous Template," in *IEEE International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, May 2015, pp. 21–28.

[5] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: a Low-power Pipeline based on Circuit-level Timing Speculation," in *Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec. 2003, pp. 7–18.

[6] M. Choudhury, V. Chandra, K. Mohanram, and R. Aitken, "Timber: Time borrowing and error relaying for online timing error resilience," in *European Conference on Design Automation (DATE)*, Mar. 2010, pp. 1554–1559.

[7] M. T. Moreira, B. S. Oliveira, J. J. H. Pontes, and N. L. V. Calazans, "A 65nm Standard Cell Set and Flow Dedicated to Automated Asynchronous Circuits Design," in *IEEE International System on Chip Conference (SoCC)*, 2011, pp. 99–104.

[8] M. T. Moreira, G. Trojan, F. G. Moraes, and N. L. V. Calazans, "Spatially Distributed Dual-Spacer Null Convention Logic Design," *Journal of Low Power Electronics*, vol. 10, no. 3, pp. 313–320, Sept. 2014.

[9] M. L. L. Sartori, R. N. Wuerdig, M. T. Moreira, and N. L. V. Calazans, "Pulsar: Constraining QDI Circuits Cycle Time Using Traditional EDA Tools," in *IEEE International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, 2019, pp. 114–123.

[10] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design – A Systems Perspective*. Springer, 2001.

[11] A. J. Martin, "The Limitations to Delay-Insensitivity in Asynchronous Circuits," in *6th MIT Conference in Advanced Research in VLSI (AUSCRYPT)*, Mar. 1990, pp. 263–278.

[12] K. M. Fant, *Logically Determined Design: Clockless System Design with NULL Convention Logic*. Wiley, 2005. [Online]. Available: https://books.google.com.br/books?id=igVTAAAAMAAJ

[13] R. B. Reese, S. C. Smith, and M. A. Thornton, "Uncle - An RTL Approach to Asynchronous Design," in *IEEE International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, 2012, pp. 65–72.

[14] M. T. Moreira, M. R. A. I. Silva, Augusto Neutzling; Martins, R. P. Ribas, and N. L. V. Calazans, "Semi-custom NCL Design with Commercial EDA Frameworks: Is it Possible?" in *IEEE International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, 2014, pp. 53–60.

[15] M. T. Moreira, "Asynchronous Circuits: Innovations in Components, Cell Libraries and Design Templates," Ph.D. dissertation, Pontifícia Universidade Católica do Rio Grande do Sul, FACIN-PPGCC, 2016.

[16] W. Lim, I. Lee, D. Sylvester, and D. Blaauw, "Batteryless Sub-nW Cortex-M0+ Processor with Dynamic Leakage-Suppression Logic," in *IEEE International Solid-State Circuits Conference (ISSCC)*, Apr. 2015, pp. 146–148.

[17] M. Blesken, S. Lütkemeier, and U. Rückert, "Multiobjective Optimization for Transistor Sizing Sub-threshold CMOS Logic Standard Cells," in *IEEE International Symposium on Circuits and Systems*, May 2010, pp. 1480–1483.

[18] R. A. Guazzelli, W. Lau Neto, M. T. Moreira, and N. L. V. Calazans, "Sleep Convention Logic Isochronic Fork: an Analysis," in *Symposium on Integrated Circuits and Systems Design (SBCCI)*, Sept. 2017, pp. 103–109.

[19] H. K. O. Berge, A. Hasanbegović, and S. Aunet, "Muller C-elements based on Minority-3 Functions for Ultra Low Voltage Supplies," in *IEEE Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, Apr. 2011, pp. 195–200.