Schmitt Trigger on Output Inverters of NCL Gates for Soft Error Hardening: is it Enough?

Ricardo Guazzelli, Guilherme Heck, Matheus Moreira, Ney Calazans
GAPH – FACIN
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Brazil
{ricardo.guazzelli, guilherme.heck}@acad.pucrs.br, {matheus.moreira, ney.calazans}@pucrs.br

***Abstract*—Interest in asynchronous circuits has increased in the VLSI research community due to the growing limitations faced during the design of synchronous circuits, which often result in over constrained design and operation. Albeit a wide variety of techniques for designing asynchronous circuits are available, quasi-delay-insensitive approaches are often preferable due to their simple timing analysis and closure. Null Convention Logic is a style that supports quasi-delay-insensitive design and enables power-, area- and speed-efficient circuits using a standard-cell methodology. However, the correct functionality of such circuits can be jeopardized by transients caused by single event effects, which can generate single event upsets. This work evaluates how Schmitt triggers on output inverters can help mitigating such problems in Null Convention Logic gates and if this approach is sufficient.**

Keywords—Asynchronous circuits, null convention logic, single event effects, hardening, schmitt trigger

I. INTRODUCTION

The use of a clock signal for control and sequencing of events in a digital circuit is one of the key factors for the development of integrated circuits (ICs) design. Such an assumption allows designers to ignore wire and gates delay given that a set of timing constraints, related to the clock signal, are met. This is the basis of the synchronous paradigm, which has been extensively employed on IC design over the last years. Classically, the possibility of perceiving time as a discrete variable considerably reduced design complexity, guaranteeing economic solutions for IC design companies. However, as technologies evolve to deep submicron nodes, the physical design of fully synchronous circuits start to be very challenging and achieving timing constraints closure under contemporary technologies aggressive PVT variations can be a prohibitively complex task [1][2].

In this scenario, asynchronous logic started to gain the attention of the research community, due to its capability to better cope with technological difficulties of the ultra-deep-submicron era. Accordingly, asynchronous logic provides better robustness towards PVT variations and can also lead to power and performance optimizations, as several studies suggest, like [3]-[5]. Asynchronous logic can be implemented based on different templates, each with its own advantages and drawbacks, as discussed in [1], [2] and [6]. However, according to Martin and Nyström [6], the most used template for asynchronous design is the quasi-delay-insensitive (QDI). Some reasons behind that are the fact that it allows simpler timing closure and analysis and that it can be implemented using standard-cell-based approaches. Furthermore, to design QDI circuits, 1-of-n delay insensitive (DI) codes [1] coupled to a 4-phase handshake protocol [1] are mostly used, due to their high power/area/speed efficiency and reduced complexity. Null Convention Logic (NCL) [7] is one of the possibilities to implement such circuits. A major advantage of NCL is that it enables power-, area- and speed-efficient standard-cell-based QDI design.

A drawback is that NCL gates are, in fact, sequential circuits. Thus, single event effects (SEEs), which are becoming one of the biggest reliability issues for advanced electronics [8] [9], impose a big challenge for NCL design. Single event transients (SETs) can generate single event upsets (SEUs) in many locations within an NCL-based circuit, which potentially make the circuit to stall, given the local data dependent nature of QDI circuits [1]. Recently, Kuang et al. proposed in [10] the usage of Schmitt Triggers in the output inverters of NCL gates in order to harden asynchronous circuits against such effects. Accordingly, the authors evaluated a scenario where the gates had their inputs attacked using a particle strike model. The obtained results suggested that having a Schmitt Trigger on the output inverter allowed obtaining a high tolerance of SEEs. The drawback is that the authors evaluated only attacks on the inputs and not on the other nodes.

In this work we designed a set of NCL gates with embedded Schmitt Triggers and simulate attacks on inputs, outputs and internal nodes of the gates using a particle strike model as proposed in [11]. The obtained results indicate that, differently from what was previously believed, the approach is not sufficient for hardening NCL gates against SEEs. The rest of this work is organized in four sections. Section II presents basic concepts on asynchronous design and NCL. Section III describes the usage of Schmitt Trigger on NCL gates and the employed particle strike model. Section IV presents experimental results and discussion. Finally, Section V draws conclusions and directions for further work.

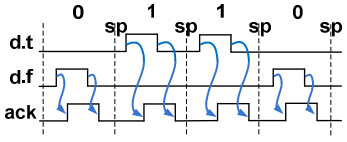
II. ASYNCHRONOUS CIRCUITS

A. Asynchronous Design

A digital circuit is synchronous if its operation is controlled by the use of a single clock signal. Otherwise it is called asynchronous. Asynchronous circuits employ explicit handshaking among their components to synchronize, communicate and operate [1]. Characterizing an asynchronous design style requires the choice of: (i) a delay model, (ii) a code to represent information, (iii) a handshake protocol, and (iv) a set of basic components. The most robust and restrictive delay model is the delay-insensitive (DI) model [1], which operates correctly regardless of gate and wire delay values. Unfortunately, this class is too restrictive. The addition of an assumption on wire delays in some carefully selected forks enables to define the (QDI) class [12]. Here, signal transitions must occur nominally at the same time only at each end point of the mentioned forks. QDI circuits are, currently, the most used class of asynchronous circuits [6].

In QDI circuits, data is encoded through DI codes, to guarantee their robustness to delay variations. The most used DI codes belong to the m-of-n class [6]. These consist of all n-bit code words where exactly m bits are at 1 and all other (n-m) bits are at 0. In circuits that use m-of-n codes, the request signal to communicate data is encoded in the data itself and, therefore, requires extra hardware for detection. A specific case of m-of-n code is the dual-rail (DR) code or 1-of-2. This code uses two wires to represent an information bit, here called d.t and d.f. Regardless the data encoding scheme, handshake protocols can be classified in 2-phase or 4-phase, as discussed in detail in [1]. The first is often based on wire transitions identifying values, while 4-phase handshaking usually assumes that wire logic level combinations define data values and transitions are required for synchronization. Usually, 2-phase protocols enable faster speeds but consume more hardware than 4-phase protocols [6].

When DR codes are associated to a 4-phase handshake protocol, all communications start with all wires at a predefined value (usually 0), called spacer. Next, after each valid value is acknowledged by a receiver, all wires must return to the spacer value. Figure 1 (a) shows an ex-ample of data transmission with this protocol, where typical values are represented in Figure 1 (b). In the displayed waveform, the first propagated value is a 0, encoded by d.t=0 and d.f=1. After the value is acknowledged by a low-to-high transition in the ack signal, a spacer is issued, represented in this case by d.t=0 and d.f=0. Next, the ac-knowledge signal switches to 0, and a new transmission can initiate.



(a)

Value	d.t	d.f
Spacer	0	0
Logic 0	0	1
Logic 1	1	0
Invalid	1	1

(b)

Figure 1 – 4-phase DR data transmission (a) and wire encoding (b).

To synchronize data transmission, asynchronous circuits using m-of-n codes require devices other than ordinary logic gates and flip-flops available in current commercial standard cell libraries. These include e.g. asynchronous registers, event fork, join and merge devices [6]. Although most of these may be built from logic gates this is often inefficient. A logic style that allows building such elements more effectively is the NCL [7].

B. Null Convention Logic

NCL was proposed by Theseus Logic, Inc. [7] and has been employed for implementing QDI asynchronous systems on silicon. It is an alternative to other design styles like delay insensitive minterm synthesis (DIMS) [1] and was applied to cope with power problems [13]-[15] and to design high speed circuits [16][17]. One of its advantages is that it enables power-, area- and speed-efficient QDI design with a standard-cell-based approach, while other asynchronous templates require recourse to full-custom approaches. In fact, there are some degrees of automation for NCL-based design, as explored in [18]-[22].

NCL gates are sometimes called threshold gates, but this is imprecise. In fact, NCL gates couple a threshold function with positive integer weights assigned to inputs to the use of a hysteresis mechanism. This is required to support QDI circuit design using 1-of-n data encoding. Figure 2 shows the basic NCL gate symbol. In such gates, N is the number of gate inputs, M is the gate threshold or a threshold function, and each input has a weight (wi). Wherever no weight is specified, weight 1 is assumed. Weights come after the W specified. The output switches to 0 when all N inputs are 0 and to 1 when the sum of weights for inputs at 1 reaches threshold M or satisfies the threshold function. Otherwise, the previous output value is maintained. These gates are usually called M-of-N gates.

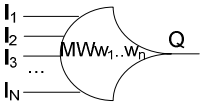


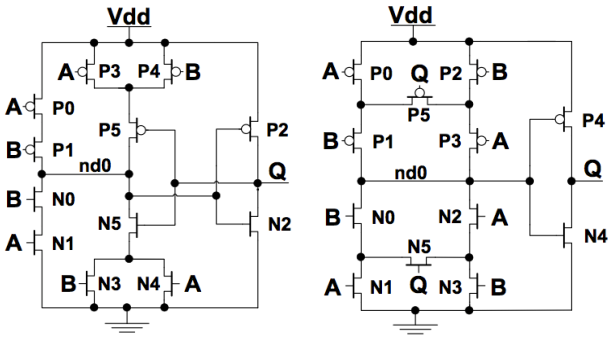
Figure 2 – Basic NCL gate symbol.

There are different manners of implementing NCL gates in the level of standard-cells. Some rely on the usage of differential logic, as discussed in [23] and [24]. However, this requires structural modifications in the design and is not compatible with state-of-the-art methods and tools for NCL design automation. Others rely on the usage of multi-threshold CMOS technologies, as discussed in [25]. However, these also require the availability of specific technologies and are not directly compatible with methods and tools proposed to date. Therefore we do not consider differential and multi-threshold topologies in this work.

In fact, in this work we considered two topologies that are discussed in detail in [26]-[28]: the Sutherland (SU), also known as static, and the van Berkel (VB), also known as symmetric topologies. These topologies are all compatible with NCL design, as discussed in [26]-[28], and are the most referred in literature. In fact, it is common knowledge that these topologies are typically more power efficient, as discussed in [28] and [29]. Also, according to Moreira et al.

[30] and Bastos et al. [31], the topologies are more suited to voltage scaling applications. Note that we discarded the dynamic [27] topology as it is not suited to QDI design. The semi-static topology [27] was also discarded for its high costs in energy and poor efficiency. We will use here an example of a 2-of-2 NCL gate, also known as a 2-input C-Element. In fact, C-Elements are a special case of NCL where the threshold is the same as the number of inputs.

In the SU topology, presented in Figure 3 (a), the reset and set functions correspond to pull-up network composed by *P0* and *P1* (RESET) and pull-down network composed by *N0* and *N1* (SET). These functions are followed by an output inverter. RESET detects when all inputs are 0, corresponding to a series of N PMOS transistors. Note that the output Q is then inverted by output inverter composed by *P2* and *N2*. SET depends on the gate threshold. Also, to ensure delay insensitivity, the gate keeps its output value when neither RESET nor SET functions are true. Since these are not complementary, the static gate requires a feedback inverter, composed by *P5* and *N5*. This inverter is controlled by a pull-up network composed by *P3* and *P4* (HOLD0) and a pull-down network composed by *N3* and *N4* (HOLD1). The former is the complement of RESET and the latter is the complement of SET. This allows turning off the feedback inverter while switching the output, allowing reductions of interference while the gate is switching. Note that this feedback scheme has typically minimum sized transistors, as it is not required for output switching and is used just for maintaining the output stable. In fact, having it minimum sized allow reducing its interference while the gate is switching.



(a) (b)

Figure 3 – Two alternative CMOS transistor topologies for NCL: (a) Sutherland's and (b) van Berkel's.

The VB topology, presented in Figure 3 (b), is similar to the SU. However RESET (*P0-P3*) and SET (*N0-N3*) networks are used for both switching the output and maintaining it. Therefore they are typically duplicated and the feedback inverter, composed by *N5* and *P5*, connects the parallel stacks. Output is given by the inverter composed by *N4* and *P4*. Note that, similarly to SU, *P5* and *N5* are minimum sized to reduce interference on output switching.

III. SCHMITT TRIGGER AND PARTICLE STRIKE MODELING

In this section, we describe the effect of using Schmitt Triggerand our approach to model particle strikes in order to validate the Schmitt Trigger's improvement. After presenting the Schmitt Trigger, we describe the adopted simulation environment, its advantages and limitations. At last, we show how particle strikes are modeled and generated during our simulations.

Schmitt Trigger has found many applications in numerous circuits, both analog and digital. However, we will focus on robustness improvements that this mechanism allows. In digital circuits, Schmitt Triggers can increase the circuit's robustness due to their ability to reject noise using hysteresis properties. In this way, the circuit can reduce its sensitive to single-event effects that could make the circuit to fail. Figure 4 shows three CMOS Schmitt Trigger inverters arrangements. In the scheme proposed in Figure 4 (b), the high-to-low glitches on the input *In* will be filtered. For instance, assume that *In*is at 1, *Q* is at 0. Now assume that the voltage in *In*is slightly decreased. Instead of this glitch propagating to output *Q*, transistor *P2* will filter it, as it ensures that, in such conditions, the source of *P1* will be connected to *gnd*. In a similar manner, the scheme proposed in Figure 4 (c) enables filtering low-to-high glitches on *In*. The scheme proposed Figure 4 (a) implements both mechanisms and allows filtering both high-to-low and low-to-high glitches.Note thatfor all these arrangements, the feedback transistors *P2*/*N2* are the source of the robustness improvement provided by the usage of Schmitt Triggers.

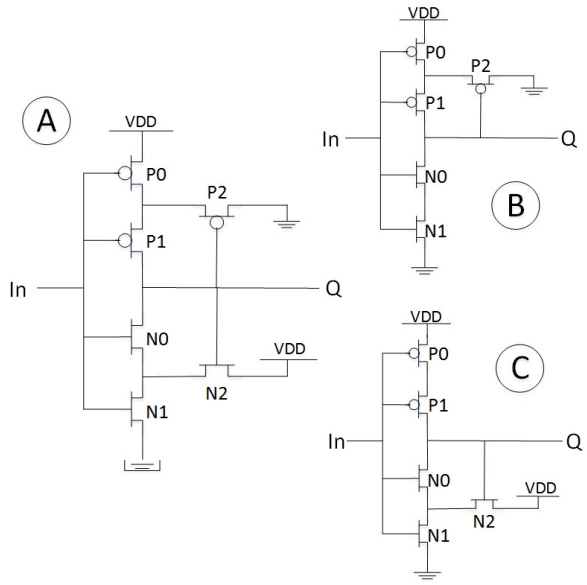


Figure 4 - Schmitt Trigger CMOS implementations

For our simulation scenarios, we used transistor level description of NCL gates. In this way, we could perform our experiments using SPICE, which would reduce computational complexity for particle strike modeling, taking device-level simulations as a reference. The approach is adequate for our case, despite the fact that circuit-level simulation could bring accuracy limitations for such models [31]. Several circuit-level

approaches approximate SEEs by square or trapezoidal pulses with equivalent magnitude-duration, but again these approaches are inadequate due its inaccuracy, as discussed in [11]. In fact, the work presented by Garg and Khatriin[11] presents a more sophisticated technique that improves the accuracy of circuit-level simulation compared to previous approaches. The technique describes the particle strike as a transient current source connected to the attacked node. Figure 5 presents example of three simulation structures used in our experiments. In these cases, a *NCL 2-of-2*gate has current sources (*I0* or *I1*) connected at its inputs, as showed in Figure 5 (a) and (b), or at its output (*I2*), as showed in Figure 5 (c). The rest of the structure will be explained further. For now, we will focus in the current sources. The current source has a double exponential behavior described by the following equation:

$$I(t) = \frac{Q}{(\tau_\alpha - \tau_\beta)}(e^{-t/\tau_\alpha} - e^{-t/\tau_\beta})$$

where Q is the collected charge by the node during the particle strike, τ_α is the collection time-constant of the junction, and τ_β is the ion-track establishment time-constant. $\tau_\alpha$and τ_β are constants that depend on process-related factors. In order to implement the described technique in SPICE language, first the current source was modeled using MATLAB and then, converted to SPICE language in order to implement the function obtained with MATLAB. Also, recalling Figure 5, it is important to clarify that a pair of inverters was inserted in the inputs of the gates to allow placing a current source (*I0* or *I1*) without interference of fixed input sources, which were employed for feeding the input inverters. Furthermore, four parallel inverters of similar driving strength to the gate being simulated were added at the output to respect the FO4 output load, typical in practical circuits. This allows a realistic scenario for evaluation.

IV. Experiments and Discussion

In our experiments, a 65nm bulk CMOS technology from STMicroeletronics was used. The two classic topologies, VB and SU (presented in Section II.B and discussed in detail in [28]), were evaluated with six different driving strengths (X2, X4, X7, X9, X13 and X18). The driving strength defines the capability of the cell to charge/discharge a given amount of load in a given period of time. The choice for the topologies was due to the fact that they provide the best energy, power, delay and area tradeoffs and have been used in practical circuits design. From each evaluated cell, three new cells were created, which have its output inverter replaced bythe Schmitt trigger inverters previously presented in Figure 4. This generated a total of 2*6*4=48 case studies.

Using the model described in the previous section, attacks were generated at inputs, output and internal nodes. For each simulation scenario, the collected charge Q was varied from 0.1fC to 30fC, with step of 0.1fC. According to the vendor, these values are realistic for the target technology. All simulations scenarios employed typical fabrication process and operating temperature. During simulation we measured the minimum collected charge that caused the output of the gate to flip incorrectly, i.e. the minimum collected charge that generated a SEU. Table 1 presents the information collected after simulations. Note that results for driving strengths bigger than X4 are omitted. In fact, the obtained results for all scenarios that used these strengths indicated that the minimum charge for generating SEUs for all topologies is over the boundary of the performed experiments (30 fC). Moreover, the Schmitt Trigger inverters shown in Figure 4(b) and (c) presented results very similar to those obtained for the scheme shown in Figure 4 (c), varying only quantitatively and not qualitatively. Therefore, all cells using these schemes were discarded, just leaving the first Schmitt Trigger inverter in our analysis.

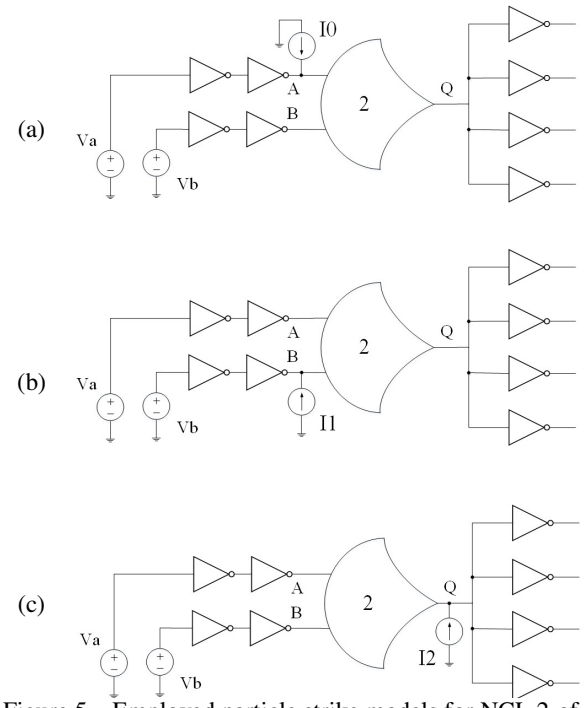


Figure 5 – Employed particle strike models for NCL 2-of-2 gates.

According to Table 1, when both topologies were attacked at inputs or at internal nodes, the critical charge to generate a SEU was increased when using Schmitt Triggers. This confirms previous results reported in [10]. However when attacked at the output, all topologies presented a substantial decrease in the critical charge for generating SEUs. In other words, the cells using Schmitt Trigger inverters in their output have a higher sensitive to attacks at the output node. This is in contrast to what was previously believed and requires a reevaluation of the usage of Schmitt Triggers for hardening NCL gates against SEEs.

Table 1 – Input, internal and output critical charges results.

Critical Charges		VB			SU		
		In (fC)	Internal (fC)	Out (fC)	In (fC)	Internal (fC)	Out (fC)
No ST	**X2**	20.5	7.1	**21.4**	21.9	8.5	**19.9**
	X4	-	7.5	**-**	-	8.9	**-**
With ST	**X2**	22.6	10.4	**15**	23.2	13.8	**13.8**
	X4	-	11	**28.4**	-	13.8	**26.2**

Another important result is the fact that the only driving strengths that proved to be problematic were the X2 and the X4, which are the smallest ones. For driving strengths bigger than that, there was no difference between using Schmitt Triggers or not in terms of robustness, no charge was enough for causing a SEU. In this way, the usage of the mechanism at the cell level may not justify, as it is the authors' own experience that Schmitt Triggers infer area, energy, leakage and delay overheads.

V. CONCLUSIONS

This paper demonstrated that having Schmitt Triggers on the outputs of NCL gates is not sufficient for hardening such gates against SEEs. In fact, for some scenarios, it makes them more vulnerable. Therefore, a reevaluation on this technique is required. Currently, we are working on a new scheme that will manage to improve the robustness for attacks either on the inputs, internal nodes or outputs.

REFERENCES

[1] P. Beerel, R. Ozdag M. Ferretti, "A Designer's Guide to Asynchronous VLSI". Cambridge University Press, 2010, 337 p.

[2] A. Yakovlev, P. Vivet, M. Renaudin, "Advances in asynchronous logic: From principles to GALS & NOC, recent industry applications, and commercial CAD tools," In: Design, Automation & Test in Europe Conference & Exhibition, 2013, pp. 1715-1724.

[3] T. Chelcea and S. M. Nowick, "Robust interfaces for mixed-timing systems," In: IEEE Transactions on VLSI Systems, 2004, vol.12(8), pp. 857-873.

[4] Y. Thonnart, P. Vivet, F. Clemerdy, "A Fully Asynchronous Low-Power Framework for GALS NoC Integration," In: Design, Automation & Test in Europe Conference & Exhibition, 2010, pp. 33-38.

[5] S. Herbert, D. Marculescu, "Analysis of Dynamic Voltage/Frequency Scaling in Chip-Multiprocessors," In: International Symposium on Low Power Electronics and Design, 2007, pp. 38-43.

[6] A. J. Martin and M. Nyström, "Asynchronous Techniques for System-on-Chip Design", Proceedings of the IEEE, 2006, vol. 94(6), pp. 1089-1020.

[7] K. M. Fant, S. A. Brandt. "NULL convention logic: a complete and consistent logic for asynchronous digital circuit synthesis". In International Conference on Application Specific Systems, Architectures and Processors, 1996, pp. 261-273.

[8] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," In: IEEE Transactions on Materials and Reliability, vol. 5, no. 3, pp. 305–316, Sep. 2005.

[9] M. Gadlage and J. Ahlbin, "Alpha-particle and focused-ion-beam-induced single-event transient measurements in a bulk 65-nm CMOS technology," In: IEEE Transactions on Nuclear Science, vol. 58, no. 3, pp. 1093–1097, 2011.

[10] W. Kuang, C. M. Ibarra, and P. Zhao, "Soft Error Hardening for Asynchronous Circuits," In: IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, pp. 273–281, Sep. 2007.

[11] R. Garg, and S. P. Khatri,"A Novel, Highly SEU Tolerant Digital Circuit Design Approach," In: IEEE International Conference on Computer Design, 2008, pp 14-20.

[12] R. Manohar and A. Martin, "Quasi-delay-insensitive circuits are Turing-complete," In: International Symposium on Advanced Research in Asynchronous Circuits and Systems, pp. 1–14, 1995.

[13] R. D. Jorgenson et al. "Ultralow-Power Operation in Subthreshold Regimes Applying Clockless Logic" In: Proceedings of the IEEE, 98(2), 2010, pp. 299-314.

[14] Z. Liang et al."Bit-Wise MTNCL: An ultra-low power bit-wise pipelined asynchronous circuit design methodology," In: IEEE International Midwest Symp, on Circ. and Systems, 2010, pp. 217-220.

[15] G. Xuguang et al. "Performance Analysis of Low Power Null Convention Logic Units with Power Cutoff," In Asia-Pacific Conference on Wearable Computing Syst., 2010, pp. 55-58.

[16] W. Jun and C. Minsu, "Latency & area measurement and optimization of asynchronous nanowire crossbar system," In Instrumentation and Measurement Technology Conference, 2010, pp. 1596-1601.

[17] Y. Yang et al."A high-speed asynchronous array multiplier based on multi-threshold semi-static NULL convention logic pipeline," In IEEE International Conf. on ASIC, 2011, pp. 633-636.

[18] M. Ligthart et al. "Asynchronous design using commercial HDL synthesis tools," In: International Symposium on Advanced Research in Asynchronous Circuits and Systems, 2000, pp. 114-125.

[19] J. Cheoljoo and S. Nowick. "Technology mapping and cell merger for asynchronous threshold networks,". In: IEEE Trans. on Computer-Aided Design of Integrated Circ. and Systems, 2008, Vol. 27(4), pp. 659-672.

[20] F. A. Parsan et al. "Gate mapping automation for asynchronous null conventionlogic circuits,". In: IEEE Transactions on VLSI Systems, 2013, early access.

[21] R. Reese et al. "Uncle - an rtl approach to asynchronous design," In: InternationalSymposium on Advanced Research in Asynchronous Circuits and Systems, 2012, pp. 65-72.

[22] M. T. Moreira, C. H. Menezes, R. C. Porto and N. Calazans "Design of NCL Gates with the ASCEnD Flow" In: Latin American Symposium on Circuits and Systems, 2013, 6p.

[23] S. Yancey and S. Smith. "A differential design for C-elements and NCL gates," In: IEEE International Midwest Symposium on Circuits and Systems, 2010, pp. 632-635.

[24] H. Lee and Y. Kim. "Low Power Null Convention Logic Circuit Design Based on DCVSL," In: IEEE International Midwest Symposium on Circuits and Systems, 2013, pp. 29-32.

[25] A.D. Bailey et al. "Ultra-low power delay-insensitive circuit design,". In: IEEE International Midwest Symp. on Circ. and Sys., 2008, pp.503-506.

[26] F. Parsan and S. Smith, "CMOS implementation of static threshold gates with hysteresis: A new approach," In IEEE International Conference on VLSI and System-on-Chip, 2012, pp. 41-45.

[27] G. E. Sobelman and K. Fant. "CMOS circuit design of threshold gates with hysteresis," In: IEEE International Symposium on Circuits and Systems, 1998, pp. 61-64.

[28] M. T. Moreira et al. "Impact of C-Elements in Asynchronous Circuits," In: IEEE International Symp on Quality Electronic Design, 2012, pp. 85-90.

[29] M. Shams et al. "Modeling and Comparing CMOS Implementations of the C-Element," In: IEEE Trans. on VLSI, 1998, Vol.6(4), pp. 563-567.

[30] M. T. Moreira and N. L. V. Calazans. "Voltage Scaling on C-Elements: A Speed, Power and Energy Efficiency Analysis," In: IEEE International Conference on Computer Design, 2013, pp. 329-334.

[31] R. P. Bastos et al. "Evaluating transient-fault effects on traditional C-Elements implementations," In: International On-Line Testing Symp., 2010, pp. 34-40.

[32] D. Menteanu and J.-L. Autran, "Modeling and Simulation of Single Effects in Digital Devices and ICs" In: IEEE Transactions on Nuclear Science, Vol. 55, No. 4, 2008, pp. 1854-1878.