

A Comparison of Asynchronous QDI Templates Using Static Logic

Ricardo A. Guazzelli, Matheus T. Moreira, Ney L. V. Calazans

GAPH - FACIN - Pontifical Catholic University of Rio Grande do Sul - Porto Alegre, Brazil

{ricardo.guazzelli, matheus.moreira}@acad.pucrs.br, ney.calazans@pucrs.br

Abstract—Asynchronous quasi-delay-insensitive (QDI) circuits are a promising solution for coping with aggressive process variations faced by modern technologies, as they can gracefully accommodate gate and wire delay variations. The literature proposes several QDI design templates with different trade-offs, giving designers a large spectrum of options to use, adapt or even mix. Among these, NULL Convention Logic (NCL), NCL+, Autonomous Signal-Validity Half-Buffer (ASVHB) and Sleep Convention Logic (SCL) are potential alternatives for low and ultra-low power applications. This paper evaluates these four QDI templates through an 8-bit Kogge-Stone full adder case study, showing analysis on cycle time, energy per operation, leakage power, energy-delay product (EDP), leakage-delay product (LDP) and area consumption. It also qualitatively evaluates each template, pointing out specific characteristics that can be suitable for low power applications.

I. INTRODUCTION

Asynchronous circuits design is becoming an increasingly important topic for the VLSI research community. These circuits can tolerate process, voltage and temperature variations more easily than their synchronous counterparts [1], [2]. In fact, by avoiding the use of a global clock signal, they allow more relaxed timing constraints, which enable to cope more efficiently with timing uncertainties arising in modern technologies. Moreover, these circuits rely on the use of local handshake protocols for control and sequencing of events [1]. Therefore, asynchronous logic is only active *when* and *where* required. In other words, parts of the circuit can be quiescent while data flows only through the path that is required to be active, potentially providing power savings and making it easier to cope with contemporary power efficiency and performance problems.

Differently from synchronous circuits, asynchronous circuits are implementable using one out of many different available design templates. However, Martin and Nyström [2] cite that practical circuits most often employ 1-of-n, 4-phase, quasi-delay-insensitive (QDI) templates. This is because these allow easier design and timing closure and are more robust. Because QDI design requires special cells, distinct from those available in standard cell libraries, and because such templates are not naturally supported by conventional EDA tools are the main reasons that keep QDI from wider adoption today. The fact that there are several distinct QDI templates available, each requiring a different set of special cells, further aggravates this problem. In this way, designers that want to experiment with QDI design need to explore the different templates available and, for each of them, design the required cells.

This paper evaluates a set of state of the art QDI circuit templates for designing asynchronous circuits. The target of the investigation is the adequacy of these templates for use in low and ultra-low power designs. To perform such an analysis, a set of cell libraries to support the exploration of the different templates was designed, followed by a set of experiments at the circuit level through a case study Kogge-Stone adder. The obtained results allow to assess the trade-offs between the different

templates and provide a set of guidelines for contemporary IC designers that intend to explore QDI design.

II. STATIC QUASI-DELAY-INSENSITIVE DESIGN

Among the different asynchronous design templates available in the literature, bundled-data and quasi-delay-insensitive (QDI) are the main template families. An advantage of bundled-data design is that it can benefit, to some extent, from the use of conventional design tools and methods, due to its similarity with synchronous design. The drawback, though, is that bundled-data templates require an extremely careful handling of the definition and verification of timing constraints relating data and control signals. An alternative to avoid these issues is to encode control and data together in special channels, which is the main choice defining QDI templates. The special channels make use of some delay-insensitive (DI) code, coupled to some specific handshake protocol. Different choices of encoding scheme and handshake method lead to distinct QDI templates.

One of the most used DI codes is called dual-rail [2]. Dual-rail channels represent each bit of data in two wires, with a special code standing for the absence of data. In this sense the channel represents data availability as well as data values. This requires extra hardware, but softens design timing closure. Dual-rail channels encode information in sets of two wires named $d.t$ and $d.f$ and rely on the use of a 4-phase handshake protocol. The literature presents two main 4-phase alternative handshake protocols: return-to-zero (RTZ) [3] and return-to-one (RTO) [4]. Figure 1(a) presents the basic encoding of a single-bit channel using RTZ and RTO protocols. A request signal can be derived as asserted at the receiver when $d.t$ and $d.f$ have different logical levels. Considering RTZ, to represent a '1' logic level, it is necessary to set $d.t$ high and $d.f$ low. The representation of a '0' logic level follows an opposite convention, i.e. $d.t$ set low and $d.f$ high. Note that, as part of the communication protocol, between each pair of valid data a *spacer* ($d.f=d.t=0$ for RTZ) must always be inserted. Note that both signals set to logic '1' is defined as an invalid state (in RTZ). In RTO, a spacer is represented with all wires set to logic '1' and the invalid state is represented by all wires at '0'. To represent the value '1' with RTO, $d.t$ must be set low and $d.f$ set high. To represent the value '0', the opposite follows: $d.t$ must be set high and $d.f$ set low.

Figure 1(b) illustrates the transmission of two data bits in sequence (a '1' bit followed by a '0' bit), using the RTZ and RTO handshake protocols. As an initial state, for RTZ, all data signals are reset in the beginning of the communication cycle, indicating a spacer. Then, the data channel presents a valid data codification (marked as 1 in Figure 1). As a consequence, the *ack* signal is asserted, signaling that the data was received (2). Next, the data channel shows a spacer, indicating the absence of valid data (3). At last, the *ack* signal is reset, ending the communication cycle (4). This same behavior applies to the RTO protocol, only using a distinct data encoding.

Several QDI templates have been proposed. Some of these are: Weak-Conditioned Half-Buffer (WCHB), Delay Insensitive Minterm Synthesis (DIMS) [3], Pre-Charge Half-Buffer

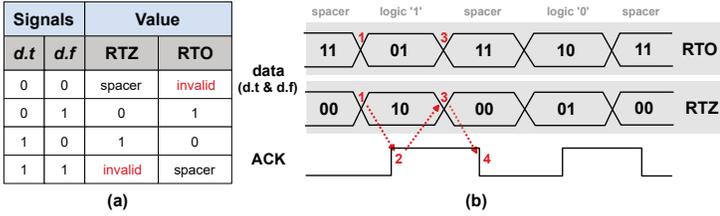


Fig. 1. (a) Codification for a 1 bit dual-rail channel using RTZ/RTO protocol and (b) an example of data transmission through a 2 bits dual rail channel based on both protocols.

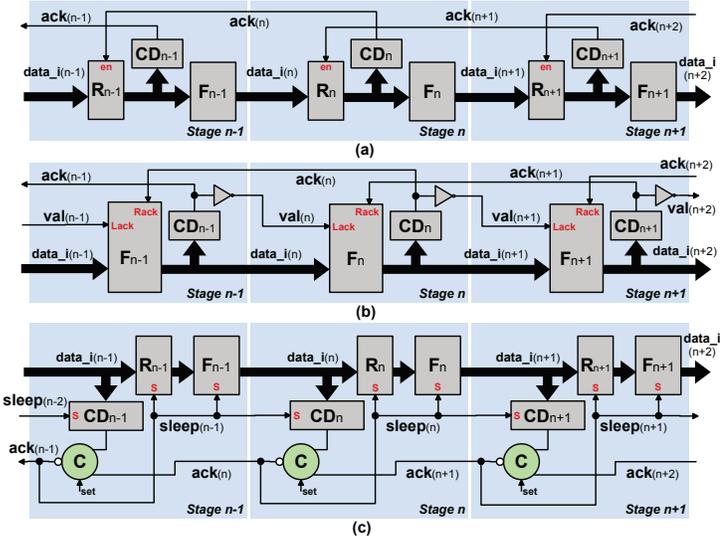


Fig. 2. Example of a 3-stage linear pipeline for (a) NCL/NCL+, (b) ASVHB and (c) SCL templates. NCL and NCL+ have the same structure, but employ different handshake protocols. Their respective block implementations (R_i , F_i and CD_i) are also distinct.

(PCHB) [1], Null Convention Logic (NCL) [5], Sleep Convention Logic (SCL) [6], Autonomous Signal-Validity Half-Buffer (ASVHB) [7] and Sense Amplifier Half-Buffer (SAHB) [8]. Each of these templates requires a set of unique logic gates, the use of which can imply distinct design trade-offs. Such trade-offs may favor the use of one or another template for specific target applications. For instance, PCHB relies on the use of domino logic. Its structure can lead to fast circuits, but signal integrity is complicated in the design of the basic cells, due to the use of dynamic logic, even if output staticizers are always present. For this reason, dynamic logic-based templates are ignored here. Among the mentioned templates, WCHB, DIMS and SAHB all rely on static logic cells, but are not addressed here either, due to the very high area consumption they imply. These also employ long series connections of transistors in the design of their basic gates, which often brings undesirable design problems, specially for low power design. Accordingly, this work restricts attention to the following four static QDI templates: NCL, NCL+, SCL and ASVHB. The NCL+ template [4] is an RTO-based variation of the NCL template.

To illustrate the covered templates, Figure 2 shows the structure of a 3-stage linear pipeline implementations using the NCL/NCL+ template (a), the ASVHB template (b) and the SCL template (c). Each pipeline stage contains at least a register block R_i , a completion detector CD_i and a logic block F_i , except the ASVHB template, which combines R_i and F_i in a single block. The SCL template, however, requires an extra settable C-element C , to implement its handshake logic. Thick and narrow lines represent data channels and handshake signals, respectively. The next sections detail each template implementation.

A. The NCL and NCL+ Templates

The NCL template was proposed in the 1990's by Theseus Logic as a alternative to optimize the otherwise large and expensive QDI implementations at the time. Its logic optimizations brought significant performance and power improvements when compared to conventional, DIMS-based designs. As Figure 2(a) suggests, the NCL pipeline has a straightforward implementation, where the *ack* signal is the only handshake signal required to implement synchronization between stages. Despite the fact that most NCL gates employ a hysteresis mechanism in its logic – Figure 3(a) shows an implementation of the true rail (Q_t) of a dual-rail NAND logic – it is possible to adopt the static logic implementation to guarantee output integrity. On the other hand, static logic implementations lead to large PMOS transistor stacking for complex gates, which can degrade circuit performance and complicate transistor sizing. In the same context, NCL+ includes similar trade-offs as NCL because it employs a dual implementation. According to [9], NCL+ provides lower leakage power and better energy efficiency, at the cost of an increase in forward propagation delay, when compared to NCL.

B. The ASVHB Template

The authors of [7] claim that the ASVHB template is suitable for ultra-low power applications. As highlighted, this template optimizes the circuit in two aspects: throughput and implementation. It improves circuit throughput by using single-cell pipeline stages, which are hard to achieve when using e.g. WCHB/DIMS. The template avoids large logic blocks in pipeline stages, increasing throughput at the cost of extra handshake logic. Regarding implementation, ASVHB integrates handshake signals *ack* into its logic and utilizes input validity data signals *val* to pre-charge internal nodes, simplifying the implementation. These signals can be visualized in Figure 2(b), which shows the a more sophisticated handshake structure when compared to the NCL pipeline. Despite these modifications, ASVHB gates are still similar to NCL gates – see Figure 3(a) and (b). The gates also employ a hysteresis mechanism with a staticized logic implementation. Again, this template improves the output value integrity but employs big stacks of PMOS transistors for complex logic function gates.

C. The SCL Template

The SCL template proposes an enhancement to the NCL template, by integrating fine-grained sleep logic into its structure. The sleep logic activates and deactivates pipeline stages according to the presence or not of data in its input data channel. When no data is present, the stage is in *sleep mode*. The sleep logic resets the pipeline stage, inserting spacers through its data channels. When the input provides valid data, the sleep logic “wakes up” the pipeline stage, putting it into *active mode* and allowing the new information to propagate through the data channels. As Figure 3(c) suggests, the use of the sleep logic not only provides significant area reduction, but also removes the need for hysteresis mechanisms in all gates – different from what occurs with NCL and ASVHB. These advantages point to the SCL template as a low area overhead, high performance approach. On the other hand, the SCL template brings timing assumptions that need to be evaluated to determine the template stability and feasibility for ultra-low power operation.

III. EXPERIMENTS

To assess the trade-offs associated with the selected QDI templates, we evaluate performance and power characteristics

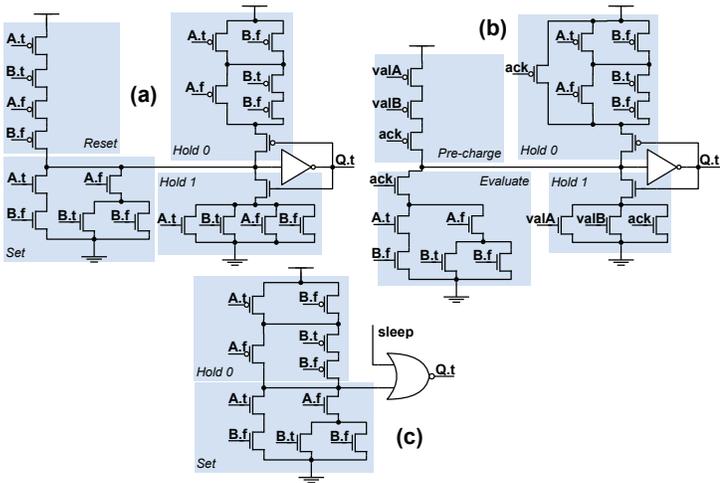


Fig. 3. Implementation of the true rail (Q_t) for a dual-rail NAND gate: (a) NCL, (b) ASVHB and (c) SCL. The corresponding false rail gates (Q_f) have a distinct structure with similar transistor count.

of a case study circuit implemented in NCL, NCL+ ASVHB and SCL. The chosen circuit is an 8-bit dual-rail Kogge-Stone adder (KSFA). Figure 4 illustrates a single-rail implementation of this adder with its three component blocks. Besides the adder itself, the case study also includes an 8-bit input register, a completion detector and extra handshake hardware, which were implemented and placed according to each template specification. For instance, the SCL-based adder uses SCL registers to implement the input registers, whereas the NCL-based one uses a conventional WCHB implementation. These additions allow a better estimation regarding cycle time, power and energy as the case study includes not only the combinational part but also the sequential and synchronization parts. All four implementations were described at the transistor level using the SPICE language and addressing the 65nm Bulk CMOS technology from STMicroelectronics. The adopted transistor sizing for these experiments follows the strategy described in [7], which employ minimum width sizing to reduce leakage and area.

Figure 5 shows an overview of the simulation environment structure. This simulation environment incorporates a mixed-signal (VHDL-AMS) tesbench, which instantiates the SPICE description of each case study and where verification blocks are described in SystemC. DAC/ADC stand for Digital to Analog Conversion and Analog to Digital Conversion.

The verification block generates the input stimuli, checks the outputs correctness and measures the cycle time characteristics (forward and backward latencies). Moreover, the environment also captures leakage power, dynamic power and energy consumption using measurements from the SPICE setup. As some templates have different I/O signals and/or data encodings, the simulation environment implements dedicated interfaces and stimuli sets for each template.

During simulation, the environment evaluates all implementations with nominal supply voltage (1V), in a typical process corner (TT) and at room temperature (25°C). The obtained simulation results are collected in Figure 6, which indicates performance and power characteristics of each QDI template. Note that all results are normalized to the results of the NCL-based case study, which is thus used as reference in the discussion. In Figure 6(a), the cycle time captures not only the data propagation latency through the case study circuit itself but also the extra handshake latency, giving a better perspective on the templates relative performance. Area estimation (Figure 6(b)) uses the circuit transistor count for each template. Energy per operation (EPO) and leakage power consumption charts provide

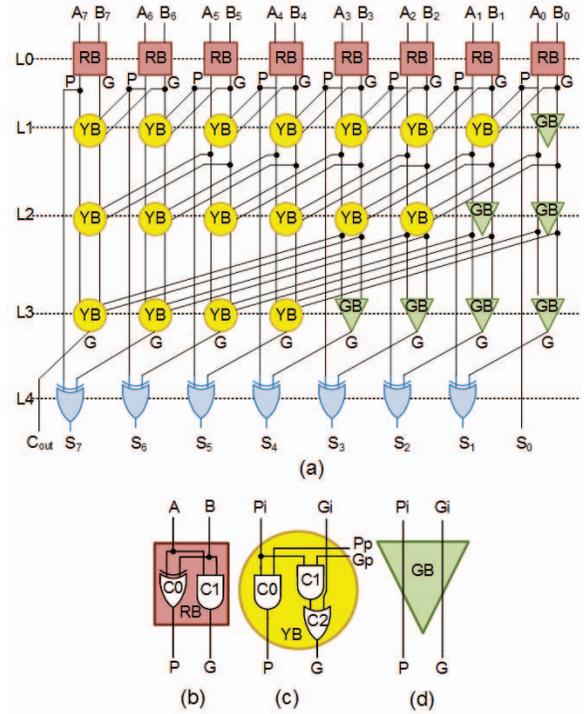


Fig. 4. Single-rail implementation of the 8-bit Kogge-Stone adder (KSFA) and its component blocks: (a) block diagram; (b) red box (RB); (c) yellow box (YB); (d) green box (GB).

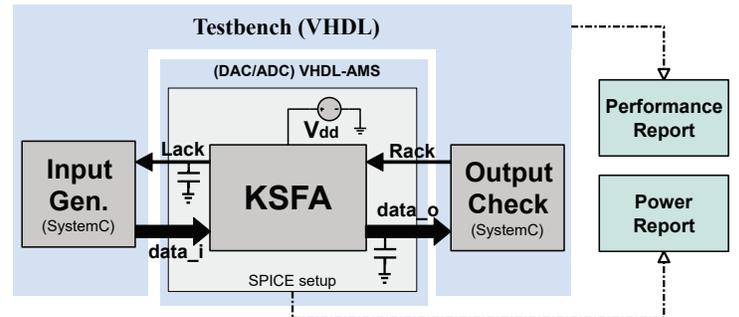


Fig. 5. KSFA simulation environment structure. The AMS-VHDL environment implements the ADC/DAC interfaces, allowing the communication between the case study circuit and the verification logic.

a power analysis regarding all covered templates. While EPO data (Figure 6(c)) pinpoints the energy consumption when the circuit is active, leakage power data (Figure 6(d)) shows the standby mode circuit consumption. The energy-delay product (EDP) (Figure 6(e)) provides a vision of the trade-off between performance and energy consumption. Similarly, the leakage-delay product (LDP) (Figure 6(f)) focuses on the trade-off between performance and leakage power consumption.

The obtained cycle time results suggest that almost all templates present slightly better results than NCL, except the SCL template. More specifically, NCL+ and ASVHB have 4% and 8% lower cycle time than NCL, respectively. Meanwhile, the SCL overhead reaches around 76%. This is explained by the fact that the SCL cycle time is substantially penalized by the completion detector performance, usually regarded as the bottleneck of QDI templates.

Regarding power consumption characteristics, NCL+ and SCL showed 16% and 48% less leakage power than NCL, respectively. The NCL+ results are related with the usage of the RTO protocol – as covered in [9] – and the SCL results derive from the significantly smaller transistor count. However, the same does

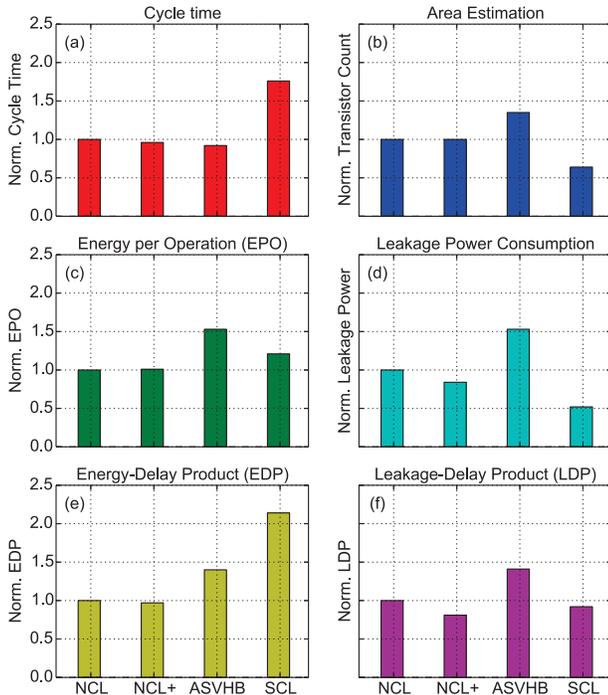


Fig. 6. Simulation results of the 8-bit Kogge-Stone adder. All results of cycle time, leakage power, energy per operation, area estimation, energy-delay product (EDP) and leakage-delay product (LDP) are normalized to the NCL results.

not apply to ASVHB. As ASVHB needs the implementation of handshake logic between each logic gate, extra hardware is required for proper synchronization. This aspect translates into an area overhead of 35%, which also explains the leakage power increase of 53%.

Concerning EPO figures, NCL also provides interesting results. Both NCL and NCL+ templates have similar energy consumption. ASVHB on the other hand increases EPO by 53%, while SCL provides 21% more EPO than NCL. Also, the SCL-based case study registers the highest EDP among all templates. This is justifiable by the fact that the SCL template has significant cycle time and EPO overheads, which are the main components of EDP. The same analysis is applicable to ASVHB. The NCL+ EDP remained similar to that observed for the NCL template, following the trend of results collected for cycle time and EPO figures.

Analyzing LDP, NCL+ and SCL achieve 8% 9% better results than NCL. This is expected as both templates provide substantial leakage power reduction. On the other hand, the higher area consumption of ASVHB implies higher LDP results, which translate to an overhead of 40% when compared to NCL.

The area estimation using transistor count highlights the logic reduction achieved by the SCL template. In fact, the SCL-based case study uses 36% and 47% less transistors than its NCL/NCL+ and ASVHB counterparts, respectively.

IV. DISCUSSION AND CONCLUSIONS

The results presented in Section III enable a qualitative evaluation of each template. The following discussion consider the main advantages of each template and highlights candidates for low power and ultra-low power applications.

NCL and NCL+ templates have similar trade-offs, only differing in leakage power where NCL+ stands out as a better choice.

In fact, the authors in [10] pinpoint NCL as a good candidate for ultra-low power applications. Thus, NCL+ can achieve further improvements, as this type of application is usually dependent on limited power source (e.g. a battery) and leakage power consumption is critical.

The ASVHB template presents the best performance results among all templates. However, its implementation has significant area and power overheads, which may compromise its use in low power applications. This indicates that ASVHB can be adopted by performance-based applications that demand the robustness provided by QDI design but need to avoid the power consumption of high performance QDI templates such as PCHB. In addition, the fine-grained pipeline structure proposed by ASVHB is a potential alternative to other templates to optimize their performance and increase energy efficiency, for instance.

Results also indicate that SCL is a potential alternative to significantly reduce area consumption of QDI circuits. This is of great relevance because QDI design is classically known for its excessive area overhead. As a side effect, area reduction also diminishes leakage power consumption. Similarly to NCL+, the leakage reduction indicates that SCL is a good candidate for low power circuits that operate in standby mode for long periods of time. Moreover, SCL can be made compatible with aggressive techniques such as sub-threshold operation. The robustness of QDI design can mitigate the significant delay variations caused by sub-threshold operation and the SCL relatively low area consumption can enable QDI design as a potential alternative for ultra-low power applications. However, it is important to highlight that the SCL cycle time still needs to be investigated. The authors are currently working on optimizations of this template, which will likely lead to a novel QDI template.

REFERENCES

- [1] P. A. Beerel, R. O. Ozdag, and M. Ferretti, *A Designer's Guide to Asynchronous VLSI*. Cambridge University Press, 2010.
- [2] A. J. Martin and M. Nyström, "Asynchronous techniques for system-on-chip design," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1089–1120, Jun. 2006.
- [3] J. Sparsø and S. Furber, *Principles of Asynchronous Circuit Design – A Systems Perspective*. Springer, 2001.
- [4] M. Moreira, R. Guazzelli, and N. Calazans, "Return-to-one Protocol for reducing Static Power in C-elements of QDI Circuits employing m-of-n Codes," in *2012 25th Symposium on Integrated Circuits and Systems Design (SBCCI)*, Aug. 2012, pp. 1–6.
- [5] K. Fant and S. Brandt, "NULL Convention Logic™: a complete and consistent logic for asynchronous digital circuit synthesis," in *International Conference on Application Specific Systems, Architectures and Processors (ASAP)*, Aug. 1996, pp. 261–273.
- [6] F. A. Parsan, S. C. Smith, and W. K. Al-Assadi, "Design for Testability of Sleep Convention Logic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 11, Article in Early Access, 2015.
- [7] W.-G. Ho, K.-S. Chong, B.-H. Gwee, and J. Chang, "Low Power Sub-threshold Asynchronous Quasi-Delay-Insensitive 32-bit Arithmetic and Logic Unit based on Autonomous Signal-validity Half-buffer," *IET Circuits, Devices & Systems*, vol. 9, no. 4, pp. 309–318, 2015.
- [8] K. S. Chong, W. G. Ho, T. Lin, B. H. Gwee, and J. S. Chang, "Sense Amplifier Half-Buffer (SAHB): A Low-Power High-Performance Asynchronous Logic QDI Cell Template," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2016, Early access at <http://ieeexplore.ieee.org>.
- [9] M. Moreira, A. Neutzling, M. Martins, A. Reis, R. Ribas, and N. Calazans, "Semi-custom NCL Design with Commercial EDA Frameworks: Is it Possible?" in *20th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, May 2014, pp. 53–60.
- [10] R. Jorgenson, L. Sorensen, D. Leet, M. Hagedorn, D. Lamb, T. Friddell, and W. Snapp, "Ultralow-Power Operation in Subthreshold Regimes Applying Clockless Logic," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 299–314, Feb. 2010.