

# Quasi-Delay-Insensitive Return-to-One Design

Matheus T. Moreira, Ney L. V. Calazans (Advisor)

GAPH – FACIN – Pontifical Catholic University of Rio Grande do Sul  
 B. 32/726 Av. Ipiranga 6684 – Porto Alegre – Rio Grande do Sul – Brazil  
 matheus.moreira@acad.pucrs.br, ney.calazans@pucrs.br

## I. INTRODUCTION AND RELATED WORK

Asynchronous design techniques are gaining attention in the scientific community for their ability to cope with current technologies' problems that the synchronous paradigm may fail to cope with. The quasi-delay-insensitive (QDI) design style [1] [2] is attractive to asynchronous circuits, especially because it allows wire and gate delays to be ignored given that isochronic fork [1] delay assumptions are respected. This reduces design complexity and eases timing closure and analysis. Defining a QDI template requires choosing a handshake protocol and a delay-insensitive (DI) code for data [2]. Here, absence of data (a *spacer*) can be signaled by setting all wires of a data channel to 0, defining the *return-to-zero* (RTZ) protocol. The RTZ protocol is well accepted in the research community, but tradeoffs of alternative manners for representing a spacer received little attention. An obvious alternative is the *return-to-one* (RTO) protocol, where a spacer is encoded by all data wires in a channel at 1 [3].

Albeit other works employ different code word representations for spacers, the ones that mention the use of all-1s spacers do not employ these to define an RTO-based protocol. In fact, all of them employ dual spacer techniques for coping with robustness issues for cryptographic cores, as discussed in [4]-[7]. Also, in such works, spacers are temporally distributed in the circuit, which leads to significant area and power overheads. This manuscript presents findings of the Ph.D. candidate displaying that the RTO protocol can lead to better power/area/speed tradeoffs for some QDI logic styles, as presented in [3], [8] and [9]. An analysis of the protocol is presented herein discussing how it enables better design space exploration for QDI design.

## II. THE RETURN-TO-ONE PROTOCOL

According to Martin and Nyström [10], asynchronous designers often employ the QDI design style using a 1-of-n DI code coupled to a 4-phase handshake protocol. One of the reasons to adopt these templates is the fact that they allow simple timing closure and analysis, while maintaining robustness inherent to QDI. Also, 4-phase protocols eases data completion detection and logic blocks are easier to design than in 2-phase, which reduces design and hardware overheads. In 1-of-n codes, data is represented using  $n$  wires. Data validity is identified when exactly one of the  $n$  wires is at a given logic value and data absence can be marked by any of the  $2^n - n$  other code words. The data absence code word is called *spacer*, as it always separates two successive 1-of-n codes in a data channel. Classically, the RTZ protocol is used, where  $n$  zeroes represent a spacer and valid code words are those with a single 1. Figure 1(a) shows the RTZ 1-of-2 code, which use two wires, called **D.1** and **D.0**, to carry a single bit of information. A '0' bit is denoted by **D.0** at 1, and a '1' bit by **D.1** at 1. In 1-of-n RTZ conventions, any code word with more than a wire at 1 represents no valid data. The RTO protocol is similar to RTZ, the only difference is that data wire values are reversed as showed in Figure 1(b).

Figure 2(a) shows data transmission in a system using the RTZ protocol. Communication starts with all wires at 0 (all-0s). Next, the sender puts data in the channel (**D.0**, **D.1**) which is acknowledged by the receiver with the **ack** signal. After the sender receives **ack**, it produces a spacer to end communication. The receiver then lowers the

**ack** signal, after which another communication can take place. As Figure 2(b) shows, differently from RTZ, RTO data transmission starts after the all-1s value is in the data channel. As soon as the sender puts valid data in the channel (**D.0**, **D.1**) the receiver may acknowledge it, by lowering the **ack** signal. Next, all data wires must return to 1 to denote a spacer, ending the transmission. When the spacer is detected by the receiver, it raises the **ack** signal and new data can follow.

In fact, the idea behind the RTO protocol is simple, and albeit a 1-of-2 example is used here, any m-of-n code can support both protocols. Also, an RTO-RTZ domain interface for a same m-of-n code requires only  $n$  inverters. As a generalization for m-of-n codes, an RTO  $D.x$  wire logical value can be translated from RTZ by Eq. (1).

$$\{x \in \mathbb{N} \mid 0 \leq x \leq m - 1\}, RTO(D.x) = \neg RTZ(D.x) \quad (1)$$

Here, expressions  $RTO(D.x)$  and  $RTZ(D.x)$  correspond to the wire logical values in the RTO and RTZ domains, respectively. In this way, according to Martin [1], the conversion of data from one domain to another is DI.

Wire Name	Spacer	Bit '0'	Bit '1'
D.1	0	0	1
D.0	0	1	0

(a)

Wire Name	Spacer	Bit '0'	Bit '1'
D.1	1	1	0
D.0	1	0	1

(b)

Figure 1 – 4-phase 1-of-2 data encoding for (a) RTZ and (b) RTO protocols.

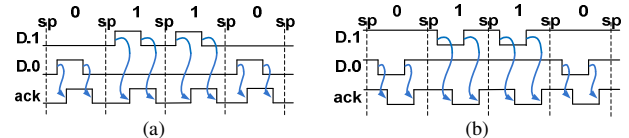


Figure 2 – Example of 4-phase (a) RTZ and (b) RTO 1-of-2 data transmission, where **sp** stands for the spacer.

## III. RETURN-TO-ONE QUASI-DELAY-INSENSITIVE DESIGN

Our work with the RTO protocol began by analyzing its effects in the electrical characteristics of C-Elements of a QDI circuit, as presented in [3]. C-elements are basic components in asynchronous circuits used for synchronization [2]. The output of a C-element will only switch to 1 when all inputs are at 1. Similarly, it will only switch to 0 if all inputs are at 0. For any other input combinations, the output keeps its previous value. Accordingly, in RTZ-based designs, C-Elements output will most often be at logic '0', while for RTO-based designs, they will most often be at logic '1'. This is justified by the fact that for all data transitions, all wires must go back to *spacer*.

During the development of an in-house 65nm library for asynchronous circuits, called ASCeND [11], we observed that static power for logic '0' in the output of C-Elements was at least 70% larger than the one measured when the output was fixed at logic '1'. In view of this observation, it is expected that asynchronous circuits in idle state, i.e. with all C-elements with a spacer in its outputs, may have the static power consumption of all C-elements reduced significantly when employing the RTO protocol rather than RTZ. This is especially true as it is the authors own experience that C-Elements require a big portion of the total area of an asynchronous circuit, up to 60% [12]. Next, we explored the effects of employing the RTO protocol for two design styles that support QDI design: delay-insensitive minterm synthesis (DIMS) and Null-Convention-Logic (NCL), both discussed in [2].

DIMS provides a way to implement Boolean functions without losing the delay insensitivity property and supports a standard-cell-based approach. In fact, it is one of the most used, due to its simplicity. In this approach, all minterms of the input variables are generated by C-elements and are then combined to perform a given function, similar to two-level logic implementations used e.g. in PLAs. For instance, Figure 3 (a) and (b) show the implementation of RTZ DIMS OR and XOR gates, respectively, and Figure 3 (c) and (d) their RTO implementations, as showed in [8]. In RTO circuits, typical AND gates detect logic '0's in internal nodes, instead of the OR gates present in the RTZ protocol for detecting '1's. Therefore, in addition to the reduced static power in C-Elements reported in [3], we expected RTO DIMS logic blocks to present lower dynamic power consumption as well, in comparison to RTZ DIMS blocks. This is due to the fact that the stack of PMOS transistors, present in typical OR gates, is avoided, because, in AND gates, series combinations of transistors appear in the NMOS region, where transistors have better electron mobility and are typically faster, smaller and present less parasitics.

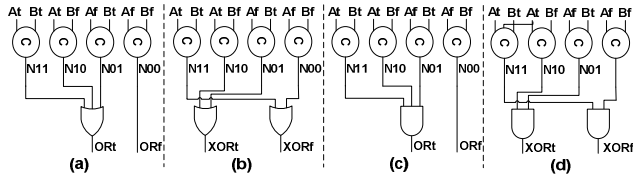


Figure 3 – RTZ (a) and (b) and RTO (c) and (d) OR and XOR DIMS gates.

In fact, what we observed through electrical simulation of the DIMS gates was an improvement in the currents required to drive valid data and spacers for RTO-based circuits, when compared to RTZ, and a tradeoff between the power for storing valid data and spacers [8]. Also, all simulated cases presented worse static power consumption when storing a valid data value for the RTO protocol. This is not very significant, because, usually, DIMS logic blocks keep valid values for short periods of time. Nonetheless, gains in the current required to drive valid values and spacers are very expressive, because they will reflect in savings in the computation of each bit of a complex logic block, and the savings in terms of static power when keeping a spacer are quite relevant for DIMS-based asynchronous circuits as well. This is due to the fact that these circuits are only active when and where required. Furthermore, when the circuit is active, some blocks are computing and some blocks are quiescent. This means that even when the circuit is operating, a portion of its DIMS logic blocks will have spacers on their outputs, reducing the static power consumption at system level. As coping with the challenged imposed by new technologies in terms of power consumption is increasingly difficult, employing the RTO protocol can prove to be very helpful in future applications for DIMS circuits.

Unpublished results of a pipelined multiplier case study based on DIMS logic provide further discussions on the use of RTO and RTZ for QDI designs. The obtained results demonstrated that using RTO or a mix of the two protocols leads to better choices, depending of the application. Results confirmed that RTO-based blocks present lower idle power and that RTO DIMS logic blocks and validity detectors are more power efficient, in terms of idle and dynamic operation, but displayed that better power compromises can be obtained for asynchronous buffers (required for pipeline designs) by using RTZ. Another set of unpublished results indicate that using RTO can harden a QDI design against transient effects. In fact, electrical simulation indicates that RTO-based logic blocks can be 4 times more robust than RTZ-based ones in best case and almost 2 times in worst case.

Finally, the effects of employing RTO were evaluated in another well-known design style in asynchronous research community: NCL. This design style is quite attractive for QDI design because it allows standard-cell-based design that typically leads to better power/area/speed figures than other styles, such as DIMS. Accordingly, in

[9] we proposed a modification of NCL that we called NCL+. The modification is the assumption of the RTO protocol, which mandates the switching function of an NCL+ gate to be the reverse of its NCL counterpart. In this way, the big series of PMOS transistors present in NCL are moved to the NMOS region. In [9], we also compared the design styles through electrical simulation of two 32 bits adders, one based in NCL and one in NCL+. Both adders were described in SPICE with post-layout extracted views. The obtained results suggested that NCL+ provides lower dynamic and static power and NCL smaller forward propagation delay and the area required by both is equivalent. Also, unpublished results pointed to significant power/speed/area improvements by mixing NCL and NCL+

#### IV. CONCLUSIONS

This work presented a set of experiments conducted to evaluate the usage of the RTO handshake protocol, instead of the classic RTZ. All results obtained so far pointed that RTO can be advantageous in different applications and for different QDI design styles. However, more importantly, the reported experiments suggest that better design space exploration can be obtained by adding RTO to the many possible ways of implementing a QDI circuit.

#### REFERENCES

- [1] A. J. Martin. The limitations to delay-insensitivity in asynchronous circuits. In ARVLSI, 1990, pp. 263-278.
- [2] P. A. Beerel, R. O. Ozdag and M. Ferretti. A Designer's Guide to Asynchronous VLSI. Cambridge University Press, 2010, 337 p.
- [3] M. T. Moreira, R. A. Guazzelli and N. L. V. Calazans. Return-to-One Protocol for Reducing Static Power in C-elements of QDI Circuits Employing m-of-n Codes. In SBCCI'12, 2012, 6p.
- [4] D. Sokolov. Automated synthesis of asynchronous circuits using direct mapping for control and data paths. PhD Thesis, SEECE, University of Newcastle upon Tyne, NCL-EECE-MSD-TR-2006-111, 2006.
- [5] W. Cilio, M. Linder, C. Porter, J. Di S. Smith and D. Thompson. Side-channel attack mitigation using dual-spacer dual-rail delay-insensitive logic (D3L). In SoutheastCon, 2010, pp. 471-474.
- [6] J. Murphy and A. Yakovlev. An alternating spacer AES crypto-processor. In ESSCIRC '06, 2006, pp. 126-129.
- [7] S. Moore, R. Anderson, R. Mullins, G. Taylor and J. J. A. Fournier. Balanced self-checking asynchronous logic for smart card applications. In: Microprocessors and Microsystems 27, 2003, pp. 421-430.
- [8] M. T. Moreira, R. Guazzelli and N. L. V. Calazans. Return-to-One DIMS Logic on 4-phase m-of-n Asynchronous Circuits. In ICECS'12, 2012, pp. 669-672.
- [9] M. T. Moreira, C. H. Menezes, R. C. Porto and N. L. V. Calazans. NCL+: Return-to-One Null Convention Logic. In MWSCAS'13, 2013.
- [10] A. J. Martin and M. Nyström. Asynchronous Techniques for System-on-Chip Design. Proc. of the IEEE, June 2006, 94(6), pp. 1089-1020.
- [11] M. T. Moreira, B. S. Oliveira, J. J. H. Pontes, F. G. Moraes and N. L. V. Calazans. Adapting a C-element Design Flow for Low Power. In: ICECS '11, 2011, pp. 45-48.
- [12] M. T. Moreira, B. Oliveira, F. Moraes and N. L. V. Calazans. Impact of C-elements in asynchronous circuits. In ISQED'12, 2012, pp. 438-444.