Return-to-One DIMS Logic on 4-phase m-of-n Asynchronous Circuits

Matheus T. Moreira [‡†], Ricardo A. Guazzelli [‡], Ney L. V. Calazans [‡]

‡Pontifical Catholic University of Rio Grande do Sul
Faculty of Computer Science – Porto Alegre, Brazil
{matheus.moreira, ricardo.guazzelli}@acad.pucrs.br,
ney.calazans@pucrs.br

†University of Santa Cruz do Sul
Department of Informatics – Santa Cruz do Sul, Brazil
matheustrevisan@unisc.br

Abstract— **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. In fact, fully synchronous SoCs may soon become unfeasible to build. Among multiple asynchronous design styles, the quasi delay insensitive (QDI) stands out for its robustness to delay variations. When coupled to DI codes like m-of-n and to four-phase handshake protocols, the QDI style produces the dominant asynchronous template currently in use. This paper evaluates the use of the Return-to-One 4-phase handshake protocol on Delay-Insensitive Minterm Synthesis (DIMS) logic blocks. Results point that this protocol leads to significant reductions on power consumption when compared to classic Return-to-Zero protocols. No extra hardware is required by the evaluated protocol, as the only required modification is that OR gates are replaced by AND gates, adding no extra delay to the resulting circuit.**

Keywords- asynchronous design; low power; m-of-n; return-to-one; QDI; delay-insensitive codes; DIMS.

I. Introduction

Asynchronous design techniques are becoming common in very large scale integration (VLSI) circuits. In fact, according to the ITRS in its 2011 edition [1], a key challenge in modern IC design is the distribution of a single clock signal throughout the chip to control the whole circuit. Therefore, a shift on VLSI design paradigm seems inevitable.

The Quasi Delay Insensitive (QDI) design style is attractive for the design of asynchronous circuits for many reasons, but especially for enabling simple timing closure and analysis [2]. Designing with QDI requires the use of delay insensitive (DI) codes [3]. Albeit a wide variety of such codes exist, just a few are practical in CMOS design [2]. Currently, the class of m-of-n codes is used with predominance. Specific codes used very often are the dual-rail (or 1-of-2 for each data bit) code and the 1-of-4 code for each pair of data bits. It is necessary to select a handshake protocol and a code to define an asynchronous design style, producing an asynchronous template. Such templates may substitute the synchronous design template. The most popular asynchronous handshake protocol is the 4-phase, due to its reduced design complexity and robustness.

This work presents the results of a study that managed to reduce the power consumption of DI Minterm Synthesis (DIMS) logic blocks [4] based on m-of-n DI codes [2], [3] and [4]. Obtaining this reduction depends on a slight change in the handshake protocol only. Moreover, the change adds no design complexity and implies no loss of performance. Results were obtained through simulations for a 65nm CMOS technology.

II. Asynchronous Circuits Concepts

A digital circuit is synchronous if its design implies the use of a single clock signal controlling all circuit events. Otherwise it is called non-synchronous. As a special case, a digital circuit is asynchronous when no clock signal is used to control any sequencing of events. These employ explicit handshaking among their components to synchronize, communicate and operate [4]. The resulting behavior is similar to a synchronous system where registers are clocked only *when* and *where* necessary. Characterizing an asynchronous design style requires: (i) the choice of a delay model, (ii) a handshaking protocol, and (iii) a set of basic devices. Each of these is explored in the rest of this Section.

The most robust and restrictive delay model is the DI model [2] [3] [4], 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. 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 asynchronous class [2].

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 [2] [3] [4]. These consist of all n-bit code words where exactly m bits are at logic '1' and all other (n-m) bits are at logic '0'. In circuits that use m-of-n codes, the request signal to communicate data is computed from the data itself and, therefore, requires extra hardware for detection. Circuits used to compute request signals are usually called validity detectors. 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, which are here called d.t and d.f. When multi-bit representations are necessary, each bit may be represented as two wires d.t and d.f carrying DR code words. This multi-bit representation is itself a DI code.

Regardless of the data encoding scheme, handshake protocols can be classified in 2-phase or 4-phase handshaking, as discussed in detail in [4]. The first is often

978-1-4673-1260-8/12/$31.00 ©2012 IEEE

based on wire transitions identifying data values, while 4-phase handshaking usually assumes that wires 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. This occurs because the latter requires that each data transmission be followed by a return to a fixed state, where all wires have the same value (the so-called spacer), increasing the time to propagate data values but significantly reducing control complexity. Indeed, 4-phase protocols are currently the most used.

When DR codes are associated to the 4-phase handshake protocol, all communications start with all wires at a predefined value (usually logic ‘0’), called spacer. Next, after each valid value is acknowledged by a receiver, all wires must return to the spacer. Figure 1 (a) shows an example of data transmission with this protocol, where its typical values are represented according to Figure 1 (b). In the displayed waveform, the first propagated value is a logic ‘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 acknowledge signal switches to logic ‘0’, the spacer, and a new transmission can initiate. The sending of any DR code word when employing a 4-phase protocol requires a spacer. In this way, the transmission of each bit of data occurs between two spacers, as showed in Figure 1 (a).

To synchronize data transmissions, 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 [4]. Although most of these may be built from logic gates this is often inefficient. A fundamental device that enables to build such elements more effectively is the C-element. Its importance comes from the fact that C-elements operate as event synchronizers. Figure 1 depicts the truth table in (c) and shows the state diagram for a basic 2 input C-element in (d). Its output may only switch when all inputs are at the same logic value. When inputs A and B are equal, the output Q assumes this same value. However, when the inputs are different, the output keeps the previous logic value at the output. This enables ordering the occurrence of events.

III. The Return-to-one Protocol

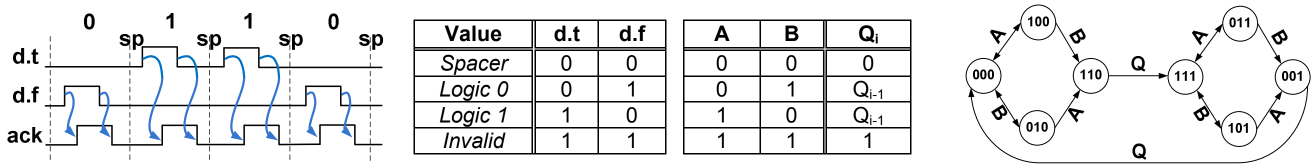
A recent work, presented in [5], demonstrated that using a RTO protocol, in contrast to the classic RTZ [4], leads to substantial savings on the static power consumed by C-Elements in an asynchronous QDI circuit using 4-phase protocol and *m-of-n* codes. Other works that mention the use of this protocol do not employ the technique for performance sake. In these works, it is used to achieve robust cryptographic hardware.

In [6], the authors propose a dual-spacer protocol, where each data value is between two different spacers: the classic all-0s spacer and an all-1s spacer. According to [6], the single spacer scheme proved to fall short in balancing the switching activity between rails, which leads to a cryptographic core vulnerable to side-channel analyses, like power and electromagnetic attacks. Results showed high robustness to these attacks can be obtained by using the dual-spacer scheme. The drawback is the increased overhead in area and power consumption. In [7], a similar work was conducted. Authors present results measured in a prototype AES cryptographic core designed with standard EDA tools, which employ the same dual-spacer technique. Results show that the technique provides high robustness to attacks and a high cost in circuit area. Similarly, in [8] authors use the all-1s encoding to represent an alarm state to obtain a balanced implementation.

To implement Boolean functions without losing the DI property, different templates can be employed for asynchronous circuits in general and specifically for 4-phase DR circuits. One of the most used, due to its simplicity, is the DIMS [4] and [9]. In this approach, all minterms of the input variables are generated by C-elements and are then combined through ORs to perform a given function, similar to two-level logic implementations used e.g. in PLAs. Figure 2 presents the implementation of some 2-input combinational logic gates in 4-phase DR RTZ DIMS: an OR, an XOR and an AND.

In these gates, all minterms are generated by C-elements, and the outputs are computed through typical OR gates. The absence of data is represented by a spacer in all input and output bits. Table I shows the behavior of the 4-phase DR RTZ DIMS logic blocks presented in Figure 2. Accordingly, a spacer in “A” and “B” generates spacers in the outputs of all blocks. These outputs will only switch when there is valid data in both “A” and “B” bits. In RTZ circuits, typical OR gates are required to detect logical 1s in the internal nodes.

To implement similar DIMS blocks, assuming the RTO protocol, the typical OR gates must be replaced by AND gates and the logical values are represented as the complement of logical values of RTZ. For instance, let *d.t* and *d.f* be the two wires used to represent a 4-phase DR value. When this value is at logical 0, for the RTZ protocol, *d.t* is at logical 0 and *d.f* at logical 1. However, for the RTO protocol, *d.t* is at logical 1 and *d.f* at logical 0.



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

A	B	Q_i
0	0	0
0	1	Q_{i-1}
1	0	Q_{i-1}
1	1	1

(a) (b) (c) (d)

Figure 1. (a) 4-phase DR data transmission; (b) typical 4-phase DR wire encoding; (c) behavioral description of a basic C-Element truth table and its state diagram (d), where the order of values inside states is ABQi.

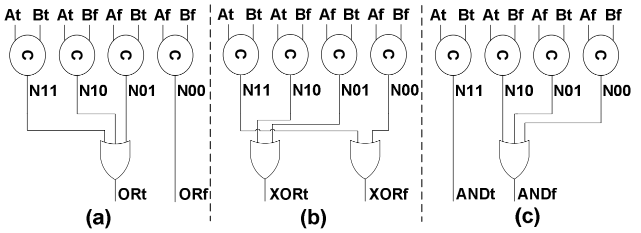


Figure 2. An OR (a), an XOR (b) and an AND (c) 4-phase DR RTZ DIMS gates.

Figure 3 shows the modification of the gates of Figure 2 for the RTO protocol. Again, all minterms are generated by C-elements. Outputs are, however, computed by AND gates. That is due to the fact that in RTO circuits the spacer is represented by all wires at logical 1 and, in conversely to the RTZ protocol, 0s are the logic levels that identify valid data, rather than 1s.

TABLE I. TRUTH TABLE FOR THE 4-PHASE DR RTZ DIMS LOGIC BLOCKS DESCRIBED IN FIGURE 2.

At	Af	Bt	Bf	ORt	ORf	XORt	XORf	ANDt	ANDf
0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	1
0	1	1	0	1	0	1	0	0	1
1	0	0	1	1	0	1	0	0	1
1	0	1	0	1	0	0	1	1	0

The modified behavior of the gates appears in Table II. In RTO circuits, typical AND gates detect logical 0s in internal nodes, instead of the OR gates present in the RTZ protocol. In addition to the reduced static power in C-Elements, reported in [5], it is expected that, RTO DIMS logic blocks present lower dynamic power consumption as well, in comparison to RTZ DIMS logic 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 are smaller and present less parasitics.

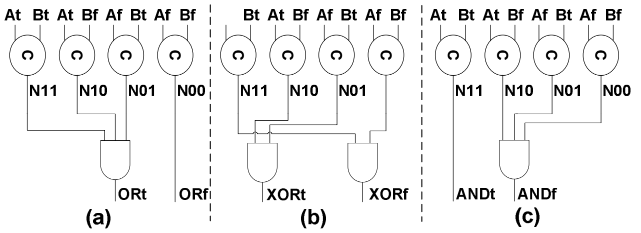


Figure 3. An OR (a), an XOR (b) and an AND (c) 4-phase DR RTO DIMS gates.

TABLE II. TRUTH TABLE FOR THE 4-PHASE DR RTO DIMS LOGIC BLOCKS DESCRIBED IN FIGURE 3.

At	Af	Bt	Bf	ORt	ORf	XORt	XORf	ANDt	ANDf
1	1	1	1	1	1	1	1	1	1
1	0	1	0	1	0	1	0	1	0
1	0	0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	1	1	0
0	1	0	1	0	1	1	0	0	1

For simplicity sake and without loss of generality with regard to any other *m-of-n* code, this paper assumes the use of DR codes, those where each bit value uses a 1-of-2 code.

IV. EXPERIMENTS AND RESULTS

The RTO and RTZ 4-phase DR DIMS gates presented in Figure 2 and Figure 3 were implemented and simulated for three different versions of static C-Elements (Martin, Sutherland and van Berkel) [10] in the 65nm STMicroelectronics CMOS technology. In this way, it is possible not only to verify the systematic reduction on power consumption achieved through the RTO protocol on C-Elements, but also the effect of using AND gates instead of OR gates to implement their functionality.

The simulated scenario employed C-Elements with a same driving capability. These were designed according to the flow proposed in [11]. The gates AND/OR, employed to generate the RTO and RTZ blocks, respectively, also had the same driving capability, to perform a fair comparison. The output load in all pins was fixed at 10fF and the input slope in all pins at 33ps. In this way, the delay of all circuits is normalized to a fixed value, to allow comparing the power consumption obtained with each DIMS block. All results presented here were obtained through SPICE simulation after physical layout extraction and are based in general purpose standard threshold transistors, for a typical fabrication process operating at 25C and 1V.

Static and dynamic power consumption characterization took place after defining a same output propagation delay for all logic blocks. The current necessary to compute valid data and spacers (directly proportional to dynamic power) and the static power required to store valid data and spacers were measured for each combination of logic function, protocol and static C-Element implementation. Table III summarizes the obtained results.

As the table shows, for the Martin C-Element, the RTO protocol proved to be very advantageous, in terms of the current required to compute values and spacers. In comparison to the RTZ protocol, savings of up to 17% and 19% were obtained for computing valid data and spacers, respectively. Savings are more modest when it comes to store a spacer, up to 7%. However, RTO proved to be more expensive in terms of storing valid data, a cost of roughly 10%. The van Berkel C-Element also proved to benefit from the RTO protocol in terms of the current required to drive output values, up to 9%. For propagating spacers, savings of up to 15% were obtained for the RTO protocol. However, gains are much more expressive when it comes to storing a spacer, up to 28%. Storing valid data also proved to be more expensive in this C-Element when employing the RTO protocol, roughly 3%. The Sutherland C-Element presented the biggest economy in terms of current required to propagate values and spacers in the RTO protocol, at least 44% and up to 48% less than the one required by RTZ. Static power consumption was also beneficial for storing spacers, roughly 13%, and worse to store valid data, roughly 4%, when comparing RTO to RTZ.

Albeit all 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. However, 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. Additionally, the savings in terms of static power when keeping a spacer are quite relevant for DIMS-based asynchronous circuits. This is due to the fact that these circuits are only active when and where required. 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 [12], employing the technique proposed in this work can prove to be very helpful in future embedded system applications.

TABLE III. SIMULATION RESULTS FOR THE RTO AND RTZ 4-PHASE DR DIMS OR, XOR AND AND. CURR. DATA AND CURR. SPACER STAND FOR CURRENT DRAWN TO COMPUTE VALID DATA AND SPACER, RESPECTIVELY. ST. DATA AND ST. SPACER STAND FOR THE STATIC POWER REQUIRED TO STORE VALID DATA AND SPACER, RESPECTIVELY

OR		**Curr. Data**	**Curr. Spacer**	**St. Data**	**St. Spacer**
Martin	*RTZ*	12.371 fA	12.410 fA	0.178 µW	0.170 µW
	RTO	10.335 fA	10.142 fA	0.197 µW	0.158 µW
	Savings	16%	18%	-10%	7%
Sutherland	*RTZ*	12.012 fA	12.015 fA	0.148 µW	0.147 µW
	RTO	6.429fA	6.300fA	0.152 µW	0.126 µW
	Savings	46%	48%	-3%	14%
van Berkel	*RTZ*	10.736 fA	10.910 fA	0.217 µW	0.182 µW
	RTO	9.767fA	9.313 fA	0.221 µW	0.131 µW
	Savings	9%	15%	-2%	28%

XOR		**Curr. Data**	**Curr. Spacer**	**St. Data**	**St. Spacer**
Martin	*RTZ*	12.254 fA	12.296 fA	0.207 µW	0.198 µW
	RTO	10.159 fA	9.934 fA	0.231 µW	0.186 µW
	Savings	17%	19%	-10%	6%
Sutherland	*RTZ*	12.251 fA	12.257 fA	0.177 µW	0.176 µW
	RTO	6.904 fA	6.744 fA	0.186 µW	0.154 µW
	Savings	44%	45%	-5%	12%
van Berkel	*RTZ*	10.959 fA	11.136 fA	0.246 µW	0.211 µW
	RTO	10.222 fA	9.736 fA	0.255 µW	0.158 µW
	Savings	7%	13%	-4%	25%

AND		**Curr. Data**	**Curr. Spacer**	**St. Data**	**St. Spacer**
Martin	*RTZ*	12.372 fA	12.410 fA	0.177 µW	0.170 µW
	RTO	10.336 fA	10.142 fA	0.197 µW	0.158 µW
	Savings	16%	18%	-10%	7%
Sutherland	*RTZ*	12.011 fA	12.014 fA	0.148 µW	0.147 µW
	RTO	6.426 fA	6.297 fA	0.152 µW	0.126 µW
	Savings	46%	48%	-3%	14%
van Berkel	*RTZ*	10.739 fA	10.913 fA	0.217 µW	0.182 µW
	RTO	9.774 fA	9.319 fA	0.221 µW	0.131 µW
	Savings	9%	15%	-2%	28%

V. CONCLUSIONS

This work evaluated the use of a method to reduce the power consumption of 4-phase *m-of-n* DIMS blocks of asynchronous circuits. The results display power savings in terms of storing spacers and computing data. The only cost is a slight increase in the static power consumption when storing valid data. However, this represents a small cost in a practical asynchronous circuit, especially when compared with the gains obtained for the static power consumption when storing a spacer.

All presented results were obtained through electrical simulations in a 65nm CMOS technology with previously designed components validated after physical layout extraction. C-elements are basic blocks in asynchronous circuit design and represent up to 60% of the total area required by standard cells in a complex module. In addition, in more advanced nodes like 45nm and 28nm, power savings in asynchronous modules are expected to be even larger. Future work includes different benchmark circuits that employ *m-of-n* RTO DIMS blocks, in order to identify where they are more critical and how to achieve greater total power reductions using the RTO protocol. Another future work is associating RTO with other *m-of-n* codes and evaluating the resulting implementations.

ACKNOWLEDGEMENTS

This work is partially supported by the CAPES-PROSUP and FAPERGS (under grants 11/0455-5 and 11/1445-0). Ney Calazans acknowledges CNPq support under grant 310864/2011-9. Authors acknowledge support granted by the INCT-SEC, process no. 573963/2008-8. Matheus Moreira acknowledges UNISC support.

REFERENCES

[1] ITRS, “Design Section”, 2011, available at http://www.itrs.net.

[2] W. J. Bainbridge, W. B. Toms, D. A. Edwards, and S. B. Furber. “Delay-insensitive, point-to-point interconnect using m-of-n codes”. In: ASYNC´03, 2003, pp. 132- 140.

[3] T. Verhoeff. Delay-insensitive codes- an overview. Distributed Computing, vol. 3(1), 1988, pp. 1-8.

[4] J. Sparsø and S. B. Furber. “Principles of asynchronous circuit design – a systems perspective”. Kluwer Academic Publishers, Boston, 2001, 360 p.

[5] 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: SBCCI’12, 2012 6p.

[6] 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´10, 2010, pp. 471-474.

[7] J. Murphy and A. Yakovlev. “An alternating spacer AES crypto-processor”. In: ISSCC´06, 2006, pp. 126-129.

[8] S. Moore, R. Anderson, R. Mullins, G. Taylor, J. J. A. Fournier. “Balanced self-checking asynchronous logic for smart card applications”. Microproc. and Microsys., vol. 27, 2003, pp. 421-430.

[9] A. J. Martin. “Formal program transformations for VLSI circuit synthesis”. In: Formal Development of Programs and Proofs, E. W. Dijkstra, Editor, Addison-Wesley, 1989, pp. 59-80.

[10] M. T. Moreira, B. S. Oliveira, F. G. Moraes, and N. L. V. Calazans. “Impact of C-elements in asynchronous circuits”. In: ISQED´12, 2012, pp. 438-444.

[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] N. Ekekwe. “Power dissipation and interconnect noise challenges in nanometer CMOS technologies”. IEEE Potentials, vol. 29(3), 2010, pp. 26-31.