Hardening QDI Circuits against Transient Faults Using Delay-Insensitive Maxterm Synthesis

Matheus T. Moreira, Ricardo A. Guazzelli, Guilherme Heck and Ney L. V. Calazans
GAPH – FACIN – Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681 Partenon Porto Alegre/RS – Brazil
{matheus.moreira, ricardo.guazzelli, guilherme.heck}@acad.pucrs.br, ney.calazans@pucrs.br

ABSTRACT
The correct functionality of quasi-delay-insensitive asynchronous circuits can be jeopardized by the presence and propagation of transient faults. If these faults are latched, they will corrupt data validity and can make the whole circuit to stall, given the strict event ordering constraints imposed by handshaking protocols. This is particularly concerning for the delay-insensitive minterm synthesis logic style, widely adopted by asynchronous designers to implement combinatory quasi-delay-insensitive logic, because it makes extensive use of C-elements and these components are rather vulnerable to transient effects. This paper demonstrates that this logic style submits C-elements to their most vulnerable states during operation. It accordingly proposes the alternative use of the delay-insensitive maxterm synthesis for hardening QDI circuits against transient faults. The latter is a logic style based on the return-to-one 4-phase protocol. Although this style also relies on extensive usage of C-elements, the states where these components are most vulnerable are avoided. Results display improvements of over 300% in C-elements tolerance to transient faults, in the best case.

Categories and Subject Descriptors
B.8.1 [**Hardware**]: Reliability, Testing, and Fault-Tolerance

General Terms
Design, Reliability.

Keywords
delay-insensitive maxterm synthesis; quasi-delay-insensitive; return-to-one; robustness; transient faults.

1. INTRODUCTION & RELATED WORK
The ever increasing demand for more complex systems and the possibility of integrating billions of transistors in a single chip brought designers to the boundaries of the synchronous paradigm capability. The efficient distribution of a global clock signal in modern, complex designs poses a very complex task. Albeit techniques and tools to help this exist, they can lead to overheads in power and area [1]. Also, as power budgets get tighter, motivated by battery-based appliances demand, and performance gets over-constrained by aggressive technologies' process variations, traditional design techniques become unsustainable [2] [3]. In fact, the International Technology Roadmap for Semiconductors predicts that a shift on integrated circuits design paradigm is required to provide further improvements [4].

In this scenario, asynchronous techniques emerge as a promising solution to cope with technological problems faced by synchronous designers. Such techniques may employ several alternative templates [5]. However, according to Martin and Nyström in [1], the majority of current asynchronous circuits rely on the quasi-delay-insensitive (QDI) delay model, using 4-phase handshaking coupled to 1-of-n delay-insensitive (DI) codes. This is mainly because this template enables simple timing closure and analysis while maintaining insensitivity to wire and gates delay, given that the isochronic fork [6] constraint is respected [7]. For such circuits, different design styles support building combinational logic. Among them, a very popular style is the Delay-Insensitive Minterm Synthesis (DIMS) [5]. One reason behind the wide adoption of DIMS is the fact that it allows the use of semi-custom design approaches, as it requires only C-elements [5] and conventional gates and supports different DI codes that accept the use of Return-to-Zero (RTZ) 4-phase handshake protocols [5].

Although DIMS logic allows achieving delay insensitivity, as any logic circuit it is not insensitive to Single Event Transients (SETs) affecting its signals. These faults are unavoidable in CMOS technologies and can be caused by effects such as glitches produced by crosstalk noise, radiation or charge sharing [8]. In fact, the correct functionality of a DIMS circuit can be easily jeopardized if a glitch propagates to inputs that directly feed C-elements, because these components are quite vulnerable to transient faults [9] [10]. The same problem arises if a glitch is generated in an internal node or in the output of the C-element. Depending on the C-element internal state, glitches large enough can be latched, which produces a Single Event Upset (SEU). This is particularly problematic because DIMS logic relies on the extensive usage of C-elements and SEUs can corrupt their stored data.

This work presents a comprehensive analysis of the analog behavior of C-elements of an in-house library called ASCEnD [11] for all their states, under the presence of glitches in their inputs. We identify situations where this component is more vulnerable and identify the associated critical points. Results show that in DIMS logic blocks, C-elements are often subjected to their most vulnerable states, requiring smaller glitches to produce SEUs. To harden QDI circuits against such faults, we propose the usage of Delay-Insensitive Maxterm Synthesis (DIMxS), which is similar to DIMS but is based in the Return-to-One (RTO) 4-phase handshake protocol [12]. In DIMxS, the states where C-elements are most vulnerable are avoided. Simulation results indicate an increase in tolerance of glitches in the inputs of such components of over 300% in the best case.

GLSVLSI'14, May 21–23, 2014, Houston, Texas, USA.

http://dx.doi.org/10.1145/2591513.2591531

2. RETURN-TO-ONE HANDSHAKING

QDI templates require the choice of a DI code and a handshake protocol. Classically, the RTZ 4-phase protocol is used in 1-of-n DI codes, where n 0s represent a spacer and valid code words are those that contain a single 1. Fig. 1(a) shows the RTZ 1-of-2 code, which uses 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 is invalid data. Fig. 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.

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

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

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

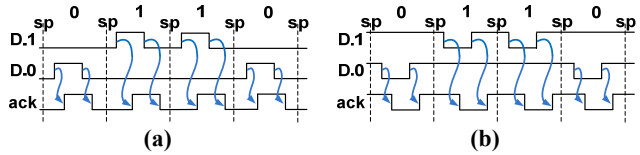


Fig. 2. Example of 4-phase (a) RTZ and (b) RTO 1-of-2 data transmission, where sp stands for spacers.

The RTO 4-phase protocol [12] is similar to RTZ. One difference is that valid data values are reversed compared to RTZ. Fig. 1(b) shows conventions for a 1-of-2 code based on RTO. Spacers are represented by n wires at 1 (all-1s). A '1' bit is denoted by D.1 at 0 and a '0' bit by D.0 at 0. As Fig. 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 channel (D.0, D.1) the receiver may acknowledge it, by lowering the ack signal. Next, all data wires must return to 1 to produce a spacer. When the spacer is detected by the receiver, it raises the ack signal and new data can follow. Thus, RTO-RTZ domain interfaces for 1-of-n codes require only n inverters. As a generalization, an RTO D.x wire logical value can be translated from RTZ by Eq. (1).

$$\forall x, 0 \le x \le n-1 : RTO(D.x) = \neg RTZ(D.x) \quad (1)$$

Here, expressions RTO(D.x) and RTZ(D.x) correspond to wire logic values in the RTO and RTZ domains, respectively. In this way, according to Martin [6], the conversion of data from one domain to another is DI. Throughout this work 1-of-2 codes will be employed, but all presented techniques can be adjusted to any 1-of-n code in a straightforward way.

3. C-ELEMENTS, SETS AND SEUS

3.1 Background

A C-element is a basic gate in QDI design, which is used for the synchronization of events. Fig. 3(a) shows its symbol. Basically, the output Q of a 2-input C-element will only switch to 1 when both inputs are at 1 and to 0 when both inputs are at 0. In any other case, the output will remain with its previous value. Alternative CMOS transistor topologies for C-elements are the Sutherland, Martin and van Berkel, showed in Fig. 3 (b), (c) and (d) respectively [13]. The logic stack (1) is the part of the circuit that is responsible for making the output to switch when both inputs have the same logic value, by charging/discharging the internal node nd0, which feeds the output inverter (2). When inputs are different, the output logic value is kept through the feedback mechanism (3).

Pontes et al. [14] and Bastos et al. [15] demonstrated that C-elements can be rather vulnerable to transient faults. When operating as a buffer (A=B), the worst consequence is the generation of a SET in the output, which will propagate as a glitch. However when operating as a memory (A≠B) this SET can be latched and generate an SEU. For instance, Fig. 4 shows the state transition graph of a 2-input C-element considering SEU and SET generation, as reported in [14]. The graph on Fig. 4 shows that when both inputs are at the same logic value, the output assumes this logic value and enters in a stable state where Q=A=B. During this state, any effect that causes the output to switch will not generate an SEU because the inputs ensure the output value. In fact, the result is the generation of an SET, where the output will incorrectly switch for a moment and then switch back, given that A=B, generating a glitch. But when A≠B, the C-element relies only on its internal memory scheme to keep the output. If the output switches incorrectly, this transition will be latched and generate an SEU. This fault will not be forced back to the correct value, because A≠B. Throughout this paper we refer to C-element memorizing states as vulnerable states.

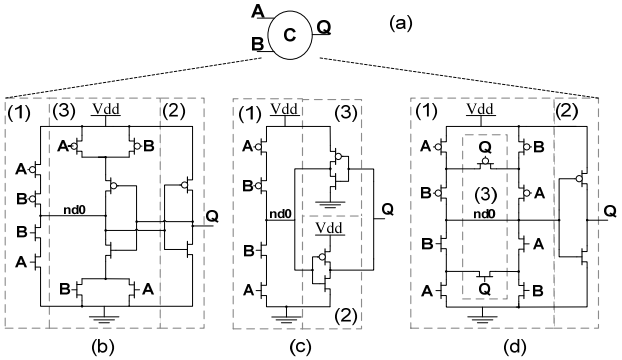


Fig. 3. C-element symbol (a) and Sutherland (b), Martin (c) and van Berkel (d) topologies [13]. For each topology (1) is the logic stack, (2) is the output inverter and (3) is the feedback mechanism.

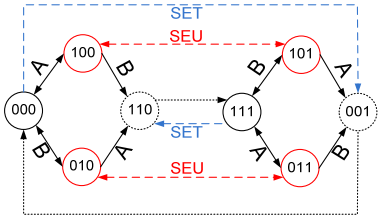


Fig. 4. State transition graph (ABQ) of a C-element considering SET and SEU effects [14]. Full-line nodes are static states. Dotted-line nodes are transition (unstable) states. Dashed lines are transitions that generate SETs or SEUs in the output. Dotted lines are transitions from unstable to stable states.

Some previous works explored the susceptibility of C-elements to transient faults. Mohammadi et al. [16] explore the effects of charge sharing on C-elements internal nodes. They demonstrate that glitches caused by such effects are a significant source of potential failure and propose modifications in C-element topology to alleviate the problem. A similar work [17] proposes rearranging transistors to alleviate the generated glitches and avoid SEU generation. Vaidyanathan et al. [18] evaluate the sensitivity of C-elements to radiation effects and propose a new transistor topolo-

gy to increase robustness. None of these works presents a solution to cope with glitches that are propagated to the output, which will end up in the input of another cell, potentially a C-element in DIMS. Bastos et al. [15] identify all the vulnerable logic states of C-elements, including possible propagation of transient faults in the inputs and demonstrate that the correct choice of transistor topology can mitigate, but not eliminate the problem. These authors propose a technique for further improvements, which consists in increasing internal nodes capacitance for filtering transient faults, but this incurs in additional delay/power/area penalties.

The revised works propose modifications in the C-element at the circuit level for hardening QDI circuits against transient faults, which degrade performance and increase area. Also, most do not explore techniques for coping with possible transient faults that can propagate to the inputs of C-elements. Using the components of the ASCEnD library [11], we note that fault generation is facilitated when the output of a C-element is at logic 0. In fact, considering radiation effects in the output of a C-element [14], state 111 (refer to Fig. 4) requires a charge 39% bigger than state 000 to generate a SET and states 011 and 101 require 36% and 3% bigger charges than states 010 and 100, respectively, for generating SEUs. This work presents simulation results for all scenarios where 2-input C-elements are at vulnerable states, injecting 0 to 1 and 1 to 0 glitches of varying sizes to the inputs in order to quantify how vulnerable the components are in each state, defining states to avoid. In this way, problems generated by transient faults can be mitigated without incurring delay/power/area penalties.

3.2 Experiments

In a first experiment, we simulate the three C-element topologies described in Fig. 3 for vulnerable states (100, 010, 101 and 011 from Fig. 4). For each state, we simulate the injection of a trapezoidal glitch in the input that can cause the output to switch. In other words, for states 100 and 010, 0 to 1 glitches were injected in B and A, respectively. For states 101 and 011, 1 to 0 glitches were injected in A and B, respectively. Such glitches had a varying height from 5 mV to 1 V in steps of 5 mV and a varying width, from 1 ps to 200 ps, in steps of 1 ps. Note that for 1 to 0 glitches the height was given as a negative variation. According to the work presented in [19], such setup provides a realistic analysis. Performing such investigation required a total of 200*200*4=160,000 simulation scenarios for each C-element, totalizing 480,000 simulations. Experiments use the Spectre simulator, from Cadence, employing post-layout extraction C-elements from ASCEnD [11], designed using a 65nm bulk CMOS technology from STMicroelectronics. Extractions occur for typical fabrication process and typical operating conditions (1 V and 25 C). Also, for all scenarios, C-elements have an output load equivalent to four inverters of the same driving strength input capacitances (FO4). Gates were also extracted for worst and best fabrication process and simulated using worst (125C and 0.9V) and best (-40C and 1.1V) operating conditions, for a total of 4,320,000 simulation scenarios. However, given that results vary only quantitatively and not qualitatively, and given the big volume of data, we only present results for the typical fabrication process under typical operating conditions. During simulation, we measure the variation in the voltage of the output for each scenario. The obtained results are summarized by Fig. 5. Results for Martin are showed in Fig. 5(a) and (b), for Sutherland in Fig. 5(c) and (d) and for van Berkel in Fig. 5(e) and (f). The first line of charts presents the results collected for 1 to 0 glitches (Glitch Down). Accordingly, in these charts, the worst case between 011 and 101 was collected. Similarly, the second line of charts present worst case for 0 to 1 glitches (Glitch Up), collected as the worst case between 010 and 100 states. Note that the points where the charts of the first line reach the bottom indicate scenarios (combination of a height and width for a glitch) that generate a SEU, because the output switched its logic value. The same applies to the points in the charts of the second line that reach the top. We refer to points in the frontier of those cases that generate SEUs as critical points, since they represent the minimum width for each glitch height (or vice-versa) that produces SEUs.

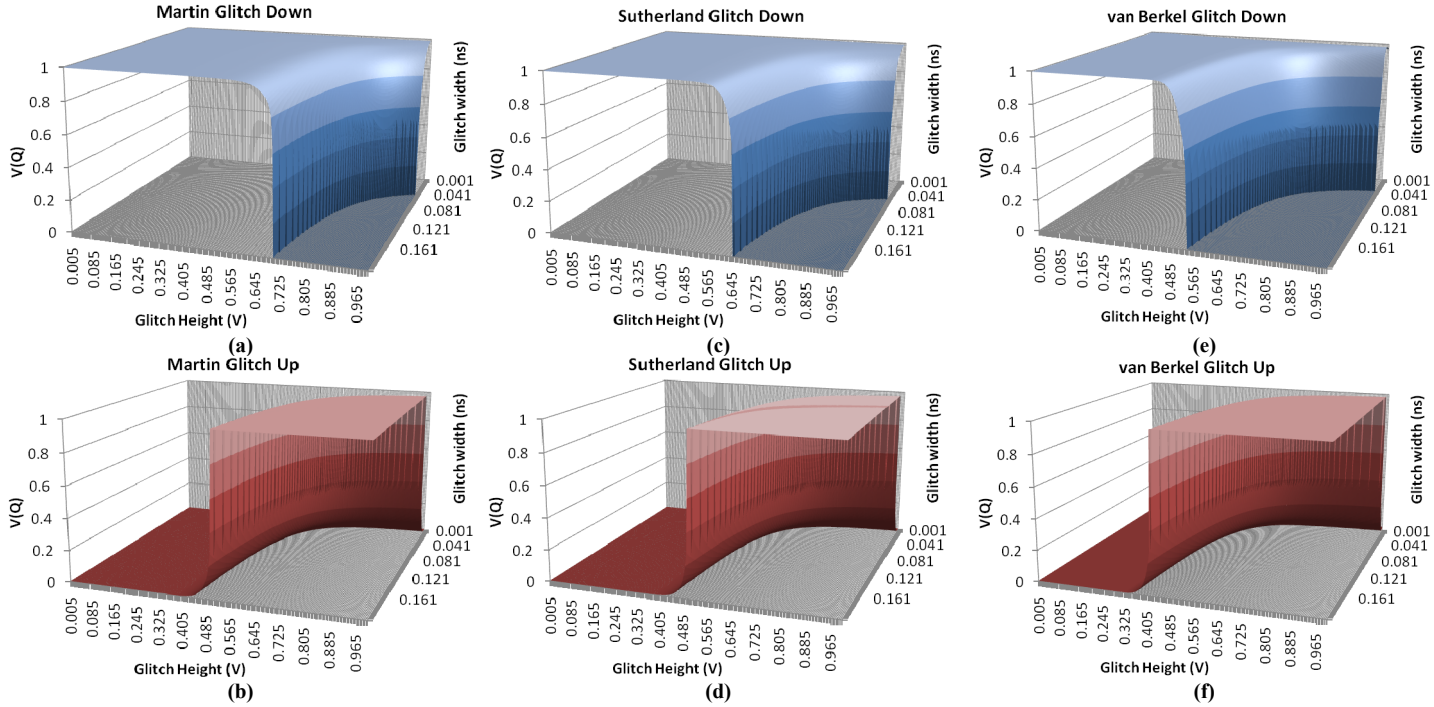


Fig. 5. Output voltage variation in the Martin (a) and (b), Sutherland (c) and (d) and van Berkel (e) and (f) topologies. Glitch Up stands for 010 and 100 scenarios and Glitch Down stands for 011 and 101 scenarios, where the worst case result was collected.

Note that critical points for the Martin topology are much bigger than those of Sutherland and van Berkel, displaying the increased robustness of this topology and confirming results pointed in [15]. However, one interesting characteristic of C-elements that no work available in the literature reports (as far as the authors could verify), is the big discrepancy between critical points for 0 to 1 and 1 to 0 glitches. Analyzing the critical points of the charts of second line of Fig. 5, it is clear that they are much smaller than those of the charts of the first line. What is more alarming is that for the Martin C-element, for instance, 0 to 1 glitches with heights of roughly 0.5 V can already generate SEUs, while for 1 to 0 glitches the minimum height is of over 0.7 V. Also, for the same topology, the minimum width for 0 to 1 glitches to produce an SEU is 22 ps, while for 1 to 0 glitches it is 42 ps. A similar analysis applies to the other two topologies. In this way, it is clear that vulnerable states where the output of the C-element is fixed at 0 are much more sensitive to glitches in the inputs than those where the output is at 1.

Another perspective of the results allows us to quantify this sensitivity more easily. Fig. 6 presents the isolated critical points for the Martin (a), Sutherland (b) and van Berkel (c) topologies, obtained from those of Fig. 5. Accordingly, the minimum heights for generating SEUs in the outputs of these C-elements for 0 to 1 glitches are roughly 0.5 V, 0.5 V and 0.4 V, respectively. For 1 to 0 glitches these values change to 0.7 V, 0.65 V and 0.55 V, respectively. Also, for the same heights, 1 to 0 glitches require much bigger widths than 0 to 1 glitches to produce SEUs. The right Y axis of the charts of Fig. 6 quantifies how bigger these widths need to be. For a worst case, these values are of 91%, 127% and 135% for Martin, Sutherland and van Berkel, respectively. However, the worst case is the one where glitches have bigger heights. If we evaluate the effects of glitches with small heights, the obtained improvements are much more significant, reaching up to 311%, 237% and 215% for Martin, Sutherland and van Berkel, respectively. This means that for the Martin topology, which is preferable for robust applications, 1 to 0 glitches in the inputs require almost two times the width of 0 to 1 glitches to generate SEUs in worst case and over four times in the best case.

Optimizations in DIMS logic may also employ inverted C-elements, as reported by Sokolov in [20]. However, as far as we could verify, no work available in current literature evaluates the effects of transient faults on these components.

In a second experiment, we generated the same 4,320,000 simulation scenarios, however this time using inverted C-elements from the ASCEnD-ST65 library. The critical points were isolated and compared to those obtained for non-inverted C-elements. Accordingly, in most cases we observed improvements in the tolerance of 0 to 1 glitches on the inputs. However for 1 to 0 glitches in general results are worse. Fig. 7 depicts the obtained results for Martin, Sutherland and van Berkel C-element topologies. Note that Glitch Up still stands for 0 to 1 glitches in the inputs while Glitch Down stands for 1 to 0 glitches, albeit the output is inverted. As the chart shows, inverted C-elements tolerate 36%, 45% and 120% wider 0 to 1 glitches for Martin, Sutherland and van Berkel, in the worst case. These values reach 135%, 51% and 166%, respectively in the best case. These results indicate that using inverted C-elements can mitigate transient faults effects for 0 to 1 glitches. However, for 1 to 0 glitches this is not the case. As Fig. 7 shows, for Martin and Sutherland topologies improvements are typically negative, indicating that the components are more sensitive for 1 to 0 glitches. For the van Berkel, results oscillate around 0, which indicate that neither inverted nor non-inverted implementations can be said to tolerate wider 1 to 0 glitches.

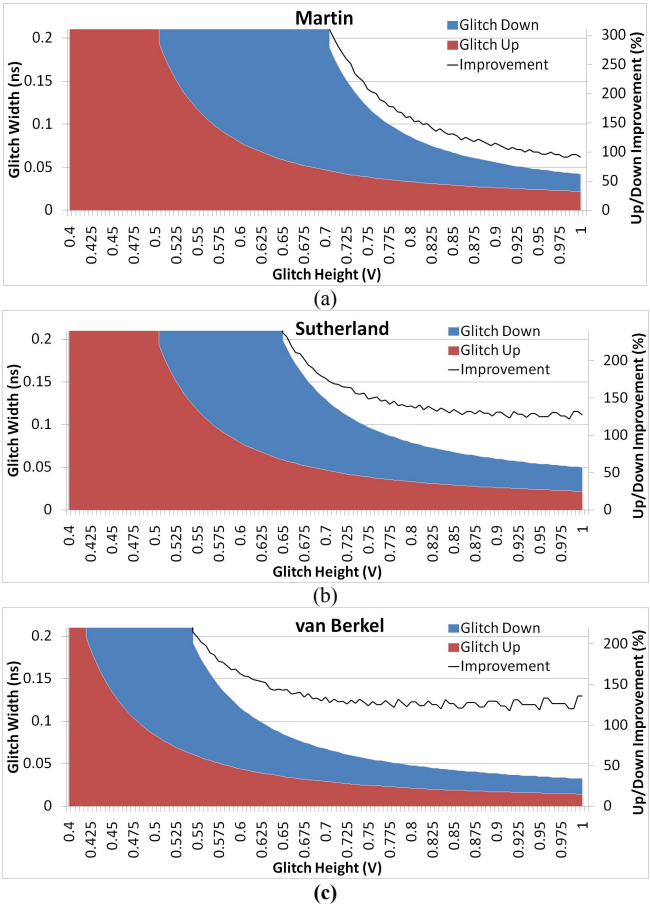


Fig. 6. Isolated critical points for Martin (a), Sutherland (b) and van Berkel (c) topologies.

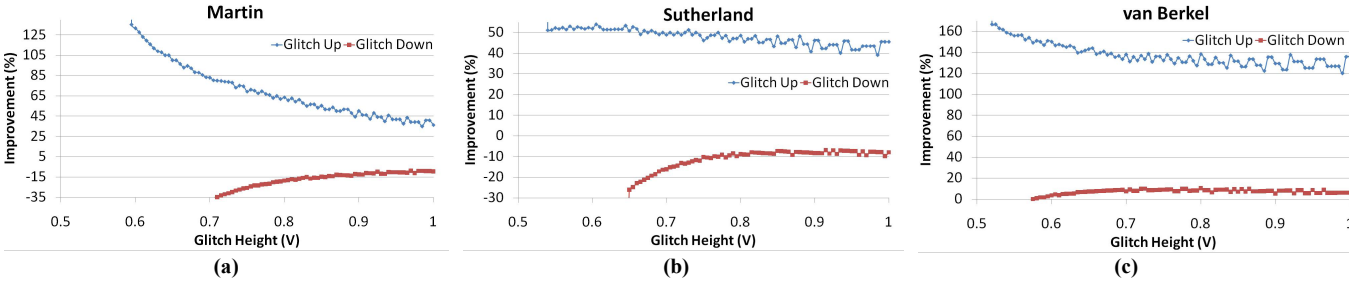


Fig. 7. Optimization provided by inverted C-elements compared to non-inverted ones for the Martin (a), Sutherland (b) and van Berkel (c) topologies.

The obtained results point that for non-inverted C-elements vulnerable states where the output was 1 tolerate bigger and wider glitches. One of the reasons behind this is the fact that in such states, glitches on the inputs of PMOS transistors are those that can generate faults, while in vulnerable states where the output is at 0, fault generation depends of glitches on the inputs of NMOS transistors. The latter are known to work much faster than the former.

4. HARDENING QDI CIRCUITS AGAINST TRANSIENT FAULTS

We demonstrated that submitting C-elements to vulnerable states with 1 in the output rather than those with 0 in the output allows mitigating problems caused by glitches in the inputs during such states, which enables hardening QDI circuits against transient faults. An analysis of the behavior of DIMS logic blocks reveals that the C-elements in this logic are often submitted to vulnerable states with 0s in the output. Meanwhile, in DIMxS blocks subject to vulnerable states, C-elements components have 1s in the outputs. Hence, we propose the usage of DIMxS for mitigating problems caused by transient faults in QDI circuits.

The DIMS logic style is a popular way of implementing combinational logic in QDI circuits. DIMS relies on the generation of the minterms for a given set of variables using C-elements, which are then combined through ORs to perform a given function, similar to two-level logic implementations used e.g. in PLAs. This style is useful to implement circuits with any 1-of-n DI code and employing 4-phase handshaking. Fig. 8(a) presents an example of a 2-input DIMS AND logic block, assuming an 1-of-2, RTZ, 4-phase handshaking template. Given that in this template a spacer precedes each valid data transmission, the reset (start) state for this circuit is all inputs at 0, which sets all internal nodes M00, M01, M10 and M11 to 0 as well. As soon as there are valid data in inputs A and B (represented by their 1-of-2 wires A.0, A.1, B.0 and B.1), one of the internal nodes will be set to 1, switching the output signal directly or triggering the OR gate, which will switch the output. For instance, if both inputs are 1 (A.1=1 and B.1=1), M11 will be set to 1, writing logic 1 in Q.1. The remaining internal nodes will still be at logic 0, keeping Q.0 in 0. This produces a logic 1 in the output. On the other hand, if at least one input is 0, as soon as both inputs have their values available in the C-elements inputs, either M00, M01 or M10 will be set to 1, writing 1 in Q.0, through the OR gate. Q.1 will still hold a 0, due to the previous spacer.

The construction of DIMxS circuits is similar to DIMS. Fig. 8(b) for instance, shows the schematic of a 2-input DIMxS AND logic block, assuming an 1-of-2, RTO, 4-phase handshaking. However, in this case, every data transmission is preceded by an all-1s spacer and C-elements generate the maxterms for the set of inputs, which are combined through an AND that implements the function output. In this way, the value of each signal of a DIMxS block is exactly the inverse of those of a DIMS block during operation, respecting Equation (1). Also, previous works demonstrate that DIMxS is more power efficient than DIMS, as reported in [21] and [22].

Fig. 9(a) shows the transition diagram for the 2-input DIMS and DIMxS AND blocks. Communications always start with two spacers. Next, valid data is inserted in both inputs, producing an output. Then, inputs must return to spacers so that new values can follow. The state of each C-element for the DIMS AND, for each state of Fig. 9(a) appears in Fig. 9(b). The only state where no C-element is at a vulnerable state is with two input spacers. For all other 8 states there are always two C-elements vulnerable. This shows that the problem cannot be ignored, as most of the time components are vulnerable. Given that in RTZ spacers are represented by the all-0s codeword, as Fig. 9(b) shows, vulnerable states of C-elements in DIMS blocks are always with a 0 in the output. Fig. 9(c) presents the same analysis for a similar DIMxS logic block. In this case, there is also a single state where C-elements are not vulnerable. However, in all remaining vulnerable states, components have 1s in the output, because spacers in RTO are represented by the all-1s codeword. From Fig. 8(a) it is clear that any fault generated by C-elements C0, C1, C2 or C3 corrupt the values of M00, M01, M10 or M11, causing the circuit to operate incorrectly. The same is valid for the equivalent DIMxS circuit in Fig. 8(b). Therefore, we believe that employing DIMxS rather than DIMS logic blocks mitigates problems caused by transient effects, as these effects are more expressive when the output of C-elements is at logic 0. To do so, RTO must be adopted in the design rather than RTZ. This can be done globally or locally in the circuit.

This approach does not require modifications in the C-elements and does not increase area or power [21]. The only modification is in the protocol assumption, which requires the usage of DIMxS rather than DIMS. In other words, OR gates are replaced by AND gates. Also, according to Equation (1), the translation of RTZ signals to RTO requires only inverters. However, because C-elements are usually employed for constructing QDI sequential and combinational components, inverted C-elements could be used for components in the borders between RTZ and RTO. In fact, results also indicate that using inverted C-elements is beneficial for mitigating problems caused by transient effects. These components could also be employed for DIMS logic optimizations such as those proposed in [20]. The equivalent state of each C-element of such optimized blocks is the same as the one in Fig. 9(c), given the inverted output. However, albeit the obtained results show that improvements could be obtained, these are much more modest than the ones obtained by using DIMxS. Therefore, we strongly advise the usage of DIMxS for hardening QDI circuits against transient faults.

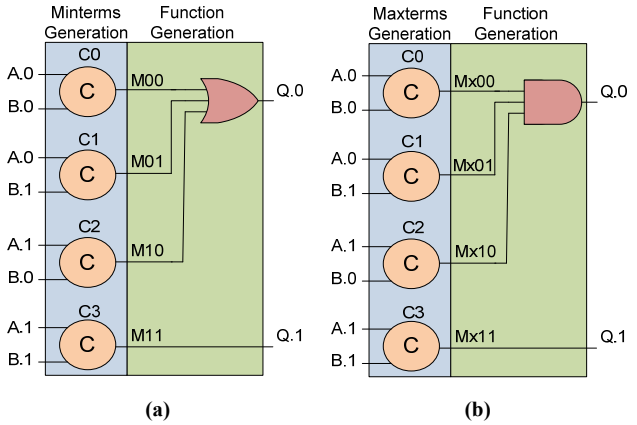
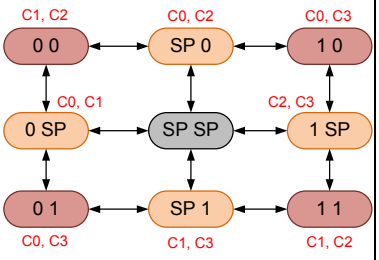


Fig. 8. Schematics for 2-input, 1-of-2 AND in (a) DIMS and (b)DIMxS



(a)

1-of-2 logic values		Wires' logic values				C-elements' states			
A	B	A.1	A.0	B.1	B.0	C0	C1	C2	C3
SP	SP	0	0	0	0	000	000	000	000
SP	0	0	0	0	1	010	000	010	000
SP	1	0	0	1	0	000	010	000	010
0	SP	0	1	0	0	100	100	000	000
0	0	0	1	0	1	111	100	010	000
0	1	0	1	1	0	100	111	000	010
1	SP	1	0	0	0	000	000	100	100
1	0	1	0	0	1	010	000	111	100
1	1	1	0	1	0	000	010	100	111

(b)

1-of-2 logic values		Wires' logic values				C-elements' states			
A	B	A.1	A.0	B.1	B.0	C0	C1	C2	C3
SP	SP	1	1	1	1	111	111	111	111
SP	0	1	1	1	0	101	111	101	111
SP	1	1	1	0	1	111	101	111	101
0	SP	1	0	1	1	011	011	111	111
0	0	1	0	1	0	000	011	101	111
0	1	1	0	0	1	011	000	111	101
1	SP	0	1	1	1	111	111	011	011
1	0	0	1	1	0	101	111	000	011
1	1	0	1	0	1	111	101	011	000

(c)

Fig. 9. Comparing DIMS and DIMxS. (a) Transition diagram for 2-input DIMS and DIMxS AND blocks; (b) States of the C-elements for a 2-input DIMS AND; (c) States of the C-elements for a 2-input DIMxS AND. Inputs are assumed to be encoded using the 1-of-2 code. SP stands for spacer.

5. CONCLUSIONS

In this work authors demonstrated that C-elements of DIMS circuits are often subjected to states that are more prone to generate SEUs. In fact, from the 9 possible states of a 2 input DIMS logic block, 8 states submit 50% of the C-elements in the block to such states. We propose DIMxS for implementing combinational logic for QDI circuits, which alleviates the problem by keeping C-elements more time in robust states. Results indicate that in these states, components tolerate glitches 311% wider than in equivalent DIMS states, in the best case, and 91% in the worst case. Also, no modifications are required in the C-elements' internal topology. The only modification is in the asynchronous template assumption. As future work, we will perform an evaluation of the behavior of DIMS and DIMxS blocks for near- and sub-threshold voltages.

6. ACKNOWLEDGMENTS

This work was partially supported by the CAPES-PROSUP (under grant 11/0455-5) and FAPERGS (under grant 11/1445-0). Authors acknowledge the support of CNPq under grants 310864/2011-9 (N.Calazans) and 142079/2013-8 (M.Moreira).

7. REFERENCES

[1] A. Martin and M. Nyström, “Asynchronous techniques for system-on-chip design,” Proceedings of the IEEE, 94(6), 2006, pp.1089-1120.

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

[3] K. Chang and J. Chang, “Synchronous-Logic and Asynchronous-Logic 8051 Microcontroller Cores for Realizing the Internet of Things: A Comparative Study on Dynamic Voltage Scaling and Variation Effects,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 3(1), March 2013, pp. 23–34.

[4] International Technology Roadmap for Semiconductors, 2011 Edition - Design Chapter, available at http://www.itrs.net, 2013, 52p.

[5] P. Beerel, R. Ozdag, and M. Ferretti, “A Designer’s Guide to Asynchronous VLSI,” Cambridge University Press, Cambridge, 2010, 337 p.

[6] A. J. Martin, “The Limitations to Delay-Insensitivity in Asynchronous Circuits,” in MIT Conference on Advanced Research in VLSI, 1990, pp. 263-278.

[7] 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, 2003, pp. 132-140.

[8] J. M. Rabaey, A Chandrakasan, and B. Nikolic, “Digital Integrated Circuits a Design Perspective,” Pearson Education, 2003, 761p.

[9] K. van Berkel, “Beware the isochronic fork,” Integration, the VLSI Journal, 13(2), June 1992, pp. 103–128.

[10] F. Ouchet, K. Morin-Allory, and L. Fesquet, “Delay Insensitivity Does Not Mean Slope Insensitivity!,” in ASYNC, 2010, pp. 176–184.

[11] M. T. Moreira, B. Oliveira, J. J. H. Pontes and N. L. V. Calazans. “A 65nm Standard Cell Set and Flow Dedicated to Automated Asynchronous Circuits Design,” in SoCC, 2011, pp. 99-104.

[12] 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, 2012, 6p.

[13] M. T. Moreira, B. Oliveira, F. G. Moraes, and N. L. V. Calazans, “Impact of C-elements in Asynchronous Circuits,” in ISQED, 2012, pp. 438-444.

[14] J. J. H. Pontes, N. L. V. Calazans, and P. Vivet, “Adding Temporal Redundancy to Delay Insensitive Codes to Mitigate Single Event Effects,” in ASYNC, 2012, pp. 142–149.

[15] R. Bastos, G. Sicard, F. Kastensmidt, M. Renaudin, and R. Reis, “Evaluating transient-fault effects on traditional C-element’s implementations,” in IOLTS, 2010, pp. 35–40.

[16] S. Mohammadi, S. Furber, and J. Garside, “Designing robust asynchronous circuit components,” IEE Proceedings - Circuits, Devices and Systems, 150(3), 2003, pp. 161–166.

[17] M. T. Moreira, B. Oliveira, F. G. Moraes, and N. L. V. Calazans, “Charge Sharing Aware NCL Gates Design,” in DFT, 6p. 2013.

[18] B. Vaidyanathan, and Y. Xie, “Soft error analysis and optimizations of C-elements in asynchronous circuits,” in SELSE, 2006, pp. 2–5.

[19] M. Omaña, D. Rossi and C. Metra, “Model for Transient Fault Susceptibility for Combinational Circuits,” Journal of Electronic Testing, 2004, 20(5), pp. 501-509.

[20] D. Sokolov, “Automated synthesis of asynchronous circuits using direct mapping for control and data paths,” PhD Thesis, University of Newcastle, 2006. 203p.

[21] M. T. Moreira, R. A. Guazzelli, and N. L. V. Calazans, “Return-to-One DIMS Logic on 4-phase m-of-n Asynchronous Circuits,” in ICECS, pp. 669-672, 2012.

[22] M. T. Moreira, J. Pontes, and N. L. V. Calazans, “Tradeoffs Between RTO and RTZ in WCHB QDI Asynchronous Design,” in ISQED, 2014, pp 692-699.