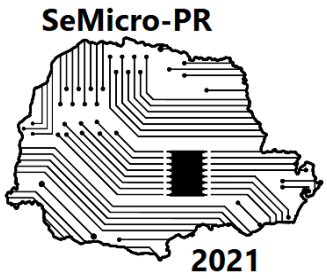

Synthesis of QDI Combinatorial Circuits with Weak-Indication using Synchronous Tools

Duarte L. Oliveira, Gracieth C. Batista
Instituto Tecnológico de Aeronáutica – ITA, SJC, Brazil
E-mail: duarte@ita.br, gracieth@ita.br

Abstract*—Due to the clock signal problems on synchronous circuits design, asynchronous circuits is a design alternative, because it not use clock signal. An essential class of asynchronous circuits, due to simplified timing analysis and are robust to PVT variations (process, supply voltage, temperature) is the class called quasi delay insensitive (QDI). However, QDI circuits have high overhead in area, due to signal coding, which uses delay insensitive code and the excessive use of C elements. This paper proposes a new method based on basic gates and C elements to implement QDI combinatorial circuits (QDI_CC). The QDI_CC circuits synthesized in the new method interact with the environment in weak-indication therefore they are robust in the interaction with the environment. The proposal provides to be promising for a set of eight benchmarks comparing the proposed architecture with two methods from the literature. Our proposal obtained for eight examples an average reduction of 22.4% and 12.6% that respectively are number of C elements and number of transistors, when compared with the known NCL_D method.

I. INTRODUCTION

A promising alternative to digital design are the asynchronous circuits, because they don't work with a clock signal, so eliminate the problems related to clock signal [1]. Asynchronous circuits are designed in different classes [1]. The delay model in which the asynchronous circuit works defines its class [1]. The Unbounded and Gate Wire Delay (UGWD) model is robust, where the delays of gates and the wires are undefined (any delay value), but finite [1]. The delay insensitive (DI) circuits satisfy the UGWD model. Martin [2] shows that this delay model is very restricted, that is, few circuits can be synthesized in this delay model. There are two less restricted variants of this delay model [1]. Firstly, the delay model in which the delay in the gates is undefined, but finite and the delay in the wires is zero (UGD). The speed-independent circuits work on UGD model, but the assumption of zero delay on wire is not real in DSM-MOS technology [3]. Secondly, the UGWD delay model, but with the isochronic fork assumption. This assumption says that, in some wires with fan-out> 1 (fork), the delays should be equal [2]. Quasi-insensitive delay (QDI) circuits work on UGWD model with the isochronic fork assumption. QDI circuits interact with the environment in I/O mode. This mode says that when an output is activated, a specified input can be activated immediately. This class has important features that are interesting, such as: *a)* a potential to present better latency time; *b)* a higher robustness to variations in temperature, supply voltage and process (PVT); *c)* a higher robustness to delay and to Stuck-at faults (easily tested fault classes); *d)* a higher modularity, allowing reusability and design as intellectual property; *e)* a better performance in security systems design; and *f)* the timing analysis is highly simplified; *g)* operates in natural form with sub-threshold supply voltages.

QDI Combinational circuits (QDI_CC) employ m-of-n DI codes to represent data, being the “4-phase” protocol the most common processing [1]. Different techniques and architectures have been proposed for the synthesis of QDI_CC [4-15]. Where [4] [7-15] are techniques that design QDI_CC at the basic gate level and C element [1], while the techniques of [5,6] design at the transistor level. The methods proposed in [8,9,11,14,15], despite being highly efficient in area, use some type of valid data and null data operation detector to satisfy QDI requirements, but the detector degrades latency. The methods [4,7, 12,13] do not use any type of detector, where [12,13] are geared towards multipliers. Sparsø, et al. [4] proposed a very simple method called Delay-Insensitive Minterm Synthesis (DIMS) that starts from canonical functions, where each product term is implemented by C elements and the sum term with OR gates. Figure 1 illustrates the DIMS method implementing a 2x1 MUX.

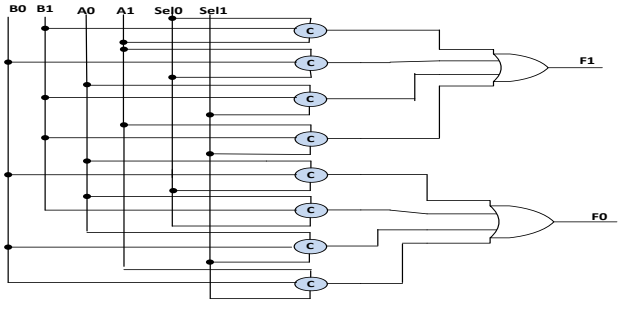

Fig. 1. *DIMS method with strong-indication: QDI MUX-2x1.*

The problem with this DIMS method is the excessive use of C elements, which drastically increases the area and degrades latency. Ligthart, et al. [7] proposed a method called NCL_D (Null Convention Logic – DIMS) that starts from minimized functions where each term (AND,OR) is implemented as DIMS. This technique reduces the number of C elements, leading to reduced area and latency. Figure 2 illustrates the NCL_D method implementing a 2x1 MUX.

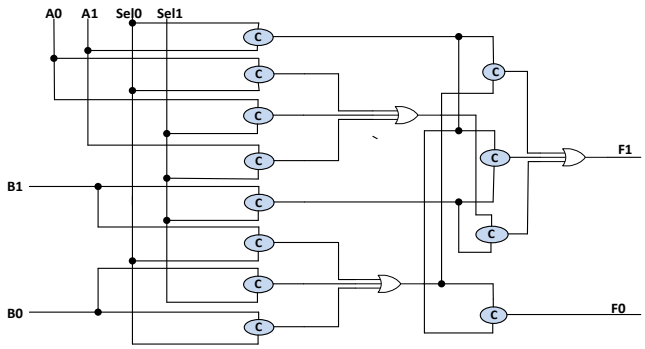

Fig. 2. *NCL_D method with wak-indication: QDI MUX-2x1.*

Despite the reduction of C elements, obtained by the NCL_D method, the use of C elements is still excessive. This paper proposes a novel method for synthesis of QDI_CC which is an extension of the NCL_D method, where the terms (AND,OR) is implemented by optimized DIMS called DIMOS (Delay Insensitive Minterms Optimized Synthesis). This new method allows a further reduction in the use of C elements. Figure 3 illustrates two implementations of the C element, where the implementation of Fig. 3d is not QDI.

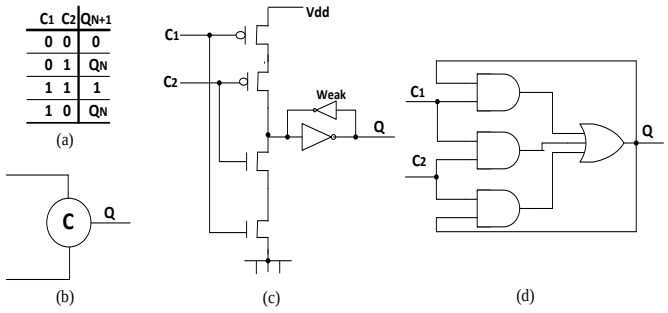

Fig. 3. *C element: a) table of operations; b) symbol; c) semi-static version; d) based on basic gates version.*

II. QUASI DELAY INSENSITIVE FUNCTION: CONCEPTS

QDI Boolean functions are synthesized in DI codes. There are different DI codes and, in this paper, we adopted the dual-rail coding [11]. The QDI combinatorial circuits (QDI_CC) that will be synthesized operate behave the 4-phase handshake protocol [11]. In the dual-rail code, each variable is encoded with two bits. For the variable *A*, we have *A1A0*=00 (null - space), *A1A0*=01 (data 0), *A1A0*=10 (data 1) and *A1A0*=11 (never occurs). The DI codes generate the operation conclusion signal without the need of a delay element and with a relatively simple circuit. The delay insensitive (DI) combinatorial circuits are subject to hazard. Hazardous circuit means there is a potential for glitches to occur, so it may lead to malfunctioning. The hazard manifests in DI circuits through gate orphan and wire orphan, i.e. a circuit is DI if it is free of gate orphan and wire orphan. QDI_CC are free of wire orphans because they satisfy the isochronic fork assumption. So the combinatorial circuit is QDI if it is free of gate orphan [13,14].

A. Boolean function: QDI condition

A QDI circuit has a gate orphan if a sequence of signal transitions across a path of one or more gates is not recognized by a transition signal on any primary output [13,14]. The indicability property indicates how robust the circuit in relation to timing analysis and ensures that the QDI_CC circuit is free of wire orphan. There are three types of indicability: *a)* strong-indication, the output signal transitions will only occur, after all the input signals are NULL or Valid; *b)* weak-indication, some transitions of output signals can occur, before all input signals are NULL or Valid, but in the last transition of the output signal, all input signals are NULL or Valid; *c)* early indication, if all transitions of the output signals are NULL or Valid, it does not mean that all input signals are NULL or Valid.

III. METHOD FOR SYNTHESIS OF QDI_CC

The method DIMOS that synthesizes our QDI circuits, interact with the environment in weak-indication mode and is composed of three steps that uses synchronous synthesis tools:

For each output variable Vs of the Boolean net, do:

a) Perform the logic minimization using the ESPRESSO tool [16] in the sum-of-products style.

b) Perform the technological mapping with the library (INV,AND2,OR2-3) using the SIS tool[16].

c) Each product term and each sum term (when there is any non-canonical product term) of Vs to implement as DIMOS.

A. Synthesis of DIMOS

Figure 4 shows the DIMS coverage and the circuits of the product term *AB* and the sum term *A+B*. For product *AB*, shown in Fig. 4a,b; the canonical (minterms) coverage is *F=AB* $\rightarrow$ as dual-rail is *F1=A1B1* and the canonical (maxterms) coverage is *F=A'B' + AB' + A'B* $\rightarrow$ as dual-rail is *F0=A0B0+ A1B0 + A0B1*. For the sum *A+B*, shown in Fig.4c,d; the canonical (minterms) coverage is *F=AB + AB' + A'B* $\rightarrow$ as dual-rail is *F1=A1B1+A1B0+A0B1* and the canonical (maxterms) coverage is *F=A'B'* $\rightarrow$ as dual-rail is *F0=A0B0*.

Figure 5 shows the DIMOS coverage and the circuits of the product term *AB* and the sum term A+*B*. For product *AB*, shown in Fig. 5a,b; the dual-rail disjoint

(minterms) minimized coverage is *F1=A1B1* and the dual-rail disjoint (maxterms) minimized coverage is *F0=B0 + A0B1* and is introduced the OR gate of the absent signal *A*, leaving *F0=B0(A0+ A1) + A0B1*. For the sum *A+B*, shown in Fig.5c,d; the dual-rail disjoint (minterms) minimized coverage is F1=A1+A0B1 and is introduced the OR gate with the absent signal *B*, leaving *F1=A1(B0+B1) + A0B1* and the dual-rail minimized coverage disjoint (maxterms) is *F0=A0B0*.

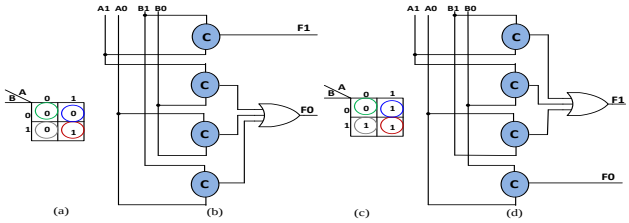

Fig. 4. *Project DIMS: a,b) AND;c,d) OR.*

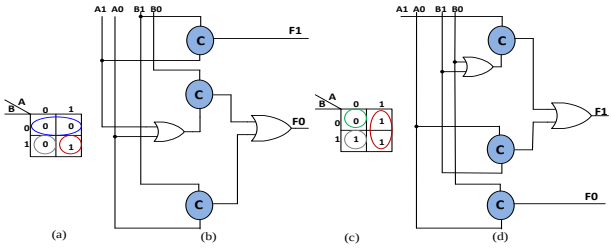

Fig. 5.*Project DIMOS:a,b) AND; c,d) OR.*

IV. EXAMPLES DIMOS

A. Example: QDI F(a,b,c)=Σ(3,6,7)

The minimized F function is: $F_{MIN}=ab+bc$. This function implemented as DIMS needs 16 C elements and 2 OR4 gates. The implementation in the NCL_D method needs 12 C elements and 3 OR3 gates. The two product terms and the sum term are implemented as DIMS. Each DIMS term needs 4 C elements and an OR3 gate. The three terms of F_{MIN} are implemented as DIMOS and Fig. 6 shows the circuit. For term *ab*, as dual-rail, disjoint minimization is *F11=a1b1* and *F10=a1b0+a0 →F10=a1b0+a0(b0+b1)*; for the term *bc*, as dual-rail, the disjoint minimization is *F21=b1c1* and *F20=b1c0+b0 →F20=b1c0+b0(c0+c1)*; for the term *F1-10+F2-10→F1-2*, as dual-rail and disjoint minimization, we have: *F1=F10F21 + F11 → F1=F110F21+F11(F20+F21)* and *F0=F10F20*.

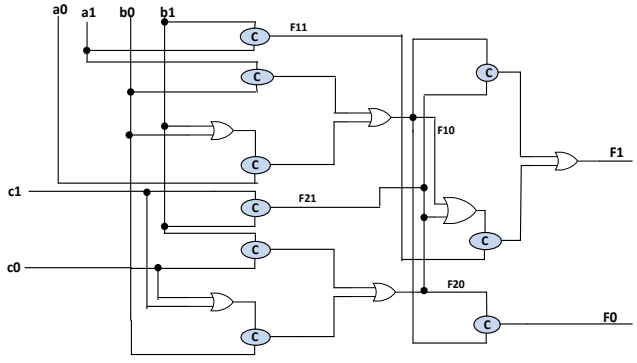

Fig. 6. *Architecture with strong-indication: QDI F.*

B. Example: QDI MUX 2x1

Figure 7 shown the project of QDI MUX 2x1, where the minimized function is: *Out=aSel'+bSel*; for the term *aSel'* as dual-rail, the disjoint minimization is *F11=a1Sel0* and *F10=a0Sel0+Sel1→ F10=a0Sel0 + Sel1 (a0+a1)*; for the term *bSel* as dual-rail, disjoint minimization is:*F21=b1Sel1* and *F20=b1Sel0+b0→ F20=b1Sel0+b0(Sel1+Sel0);* for the term *F1+F2* as dual-rail, the disjoint minimization is:*Out1=F20F11 + F21→Out1=F20F11+F21(F11+F10)*and *Out0=F10F20.*

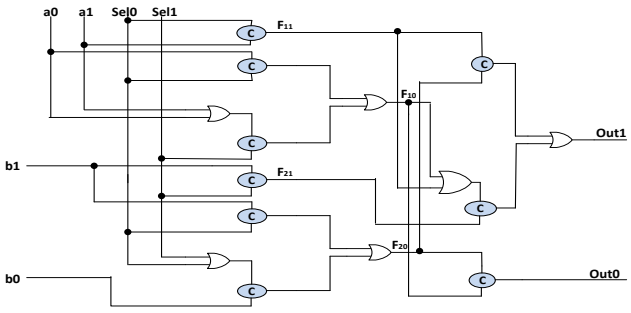

Fig. 7. *DIMOS with weak-indication: QDI MUX 2x1.*

C. Example: QDI Half Adder

Figure 8 shown the project of QDI half adder, where the dual-rail disjoint minimized functions are: *S1=A1B0 + A0B1*; *S0=A0B0 + A1B1*; *Cout1=A1B1*; *Cout0=A0 + A1B0 → Cout0=A0(B0+B1) + A1B0*.

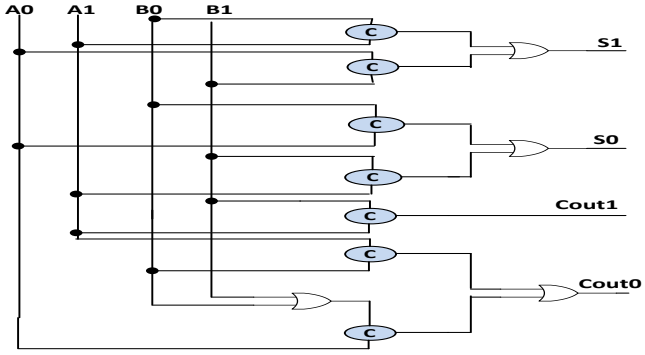

Fig. 8.*Architecture with strong-indication: QDI Half Adder.*

D. Example: QDI 1-bit equality comparator

Figure 9 shown the project of QDI 1-bit equality comparator, where the dual-rail disjoint minimized function is: *Eq1=A1B1 + A0B0*; *Eq0=A1B0 + A0B1*. In this example there was no application of the optimization technique, because the canonical already minimal.

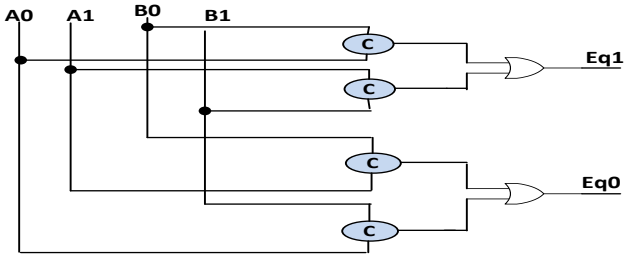

Fig. 9. *Architecture with strong-indication: QDI 1-bit equality comparator.*

V. EXPERIMETAL RESULTS

In order to demonstrate our proposal viability, eight benchmarks {AND4, AND8, MUX2x1, MUX4x1, Half-Adder, Full-Adder, Equality comparator of 1-bit, 4x2, Encoder of [8]} were synthesized, in the methods DIMS [4], NCL_D [7] and in our DIMOS proposal.

Table I shows the results of these eight benchmarks for the four methods, involving number of gates, number of C elements, and estimated number of transistors. The estimation of the number of transistors occurred following the procedure of using a library of gates {AND2, OR2, NAND, NOR, C element}, where six transistors for AND2, OR2 gates, and NAND/NOR is 2*Fan-in; twelve transistors for C element implementation, which is required in the static style.

Comparing our proposal with the NCL_D method, which is also a weak-indication, we had an average reduction of 12.6% and 22.4% respectively number of transistors and number of C elements. Making a comparison with the DIMS method that generates circuits with strong-indication, we had an average reduction of 40.9% and 51.1% respectively number of components (gates + C elements) and number of transistors.

TABLE I RESULT OF EIGHT BENCHMARKS

Example	Method	Number of gates	Number of C elements	Number of transistors
AND4 dual-rail	NCL_D of [7]	3	12	168
	DIMS of [4]	4	48	620
	Proposal DIMOS	6	9	144

(a)

Example	Method	Number of gates	Number of C elements	Number of transistors
AND8 dual-rail	NCL_D of [7]	7	28	392
	DIMS of [4]	9	100	1296
	Proposal DIMOS	14	21	336

(b)

Example	Method	Number of gates	Number of C elements	Number of transistors
MUX 2x1 dual-rail	NCL_D of [7]	3	12	168
	DIMS of [4]	2	16	212
	Proposal DIMOS	6	9	144

(c)

Example	Method	Number of gates	Number of C elements	Number of transistors
MUX 4x1 dual-rail	NCL_D of [7]	11	44	614
	DIMS of [4]	6	48	636
	Proposal DIMOS	22	33	528

(d)

Example	Method	Number of gates	Number of C elements	Number of transistors
Half Adder dual-rail	NCL_D of [7]	3	8	116
	DIMS of [4]	3	8	116
	Proposal DIMOS	4	7	108

(e)

Example	Method	Number of gates	Number of C elements	Number of transistors
Full Adder dual-rail	NCL_D of [7]	7	20	288
	DIMS of [4]	4	16	232
	Proposal DIMOS	10	17	264

(f)

Example	Method	Number of gates	Number of C elements	Number of transistors
Equality Comparator of 1-bit dual-rail	NCL_D of [7]	2	4	60
	DIMS of [4]	2	4	60
	Proposal DIMOS	2	4	60

(g)

Example	Method	Number of gates	Number of C elements	Number of transistors
Encoder Fig. 9 [8] dual-rail	NCL_D of [7]	6	24	336
	DIMS of [4]	10	48	656
	Proposal DIMOS	12	18	288

(h)

VI. CONCLUSION

Designs of combinatorial digital circuits in DSM-MOS technologies must have characteristics that the QDI style is the most promising. There are different literature methods for the synthesis of QDI_CC, but they all use C elements. In this paper we proposed a novel method for synthesis of QDI_CC which is an extension of the NCL_D method, where the terms (AND,OR) is implemented by optimized DIMS called DIMOS. This new method allows a further reduction in the use of C elements. We show for eight examples, the performance of the proposed architecture, when compared to two methods of literature → 46.4% average reduction of C elements. Further works to develop an automated method for synthesis of large QDI_CC in our DIMOS method.

REFERENCES

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

[2] J. Martin, "The Limitations to Delay Insensitive in Asynchronous Circuits," 6th MIT Conference on Advanced Research in VLSI Processes, pp.263-277, 1990.

[3] B. H. Calhoun, et al., "Digital Circuit Design Challenges and Opportunities in the Era of Nanoscale CMOS," *Proc. of the IEEE*, Vol. 96, No. 2, pp. 343-365, February 2008.

[4] J. Sparsø, J. Staunstrup, "Delay Insensitive Multi-Ring Structures", *Integration, the VLSI Journal*, v15(13), 1993.

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

[6] C. Chuang, Y. Lai, and J. R. Jiang, "Synthesis of PCHB-WCHB Hybrid Quasi-Delay Insensitive Circuits," 51st ACM/EDAC/IEEE Design Automation Conference (DAC), pp.1-6, 2014.

[7] M. Ligthart, K. Fant, R. Smith, A. Taubin, A. Kondratyev, "Asynchronous design using commercial HDL synthesis tools", IEEE 6th Int. Sym. on Advanced Research in Asynchronous Circuits and Systems, pp. 114–125, 2000.

[8] A. Kondratyev, K. Lwin, "Design of Asynchronous Circuits Using Synchronous CAD Tools", *IEEE Design and Test of Computers*, vol. 19, no. 4, pp. 107-117, 2002.

[9] P. Balasubramanian, and D.A. Edwards, "Efficient Realization of Strongly Indicating Function Blocks," IEEE Computer Society Annual Symposium on VLSI, ISVLSI '08, pp.429-432, 2008.

[10] Fu-Chiung Cheng and Chi Chen, "Can QDI Combinational Circuits be Implemented without C-elements?," IEEE 19th International Symposium on Asynchronous Circuits and Systems, pp134-141, 2013.

[11] P. Balasubramanian, et al. "Asynchronous early output section-carry based carry lookahead adder with alias carry logic," IEEE 30th International Conference on Microelectronics (MIEL), pp.1-7, 2017.

[12] D. L. Oliveira, et al. "Synthesis of QDI Combinational Circuits using Null Convention Logic Based on Basic Gates," *Advances in Science, Technology and Engineering Systems Journal* Vol. 3, No. 4, 308-317, 2018.

[13] P. Balasubramanian, D. L. Maskell, N. E. Mastorakis, "Indicating Asynchronous Multipliers," 2nd European Conference on Electrical Engineering and Computer Science (EECS), pp.1-7, 2018.

[14] P. Balasubramanian, et al., "Early Output Quasi-Delay-Insensitive Array Multipliers," *Electronics, MDPI*, 8,444, pp.1-14, 2019.

[15] D. L. Oliveira, et al. "A High Performance Implementation of Quasi Delay Insensitive Booleans Functions," IEEE XXVI International Conference on Electronics, Electrical Engineering and Computing (INTERCON), pp.1-4, 2019.

[16] E. Sentovich et al. "SIS: A system for sequential circuit synthesis, "Technical Report, UCB/ERI, M92/41, ERL, Dept. of EECS, UC Berkeley, 1992.