World Applied Sciences Journal 4 (5): 655-660, 2008
ISSN 1818-4952

Quantum-dot Cellular Automata Serial Adder Design Exploiting Null Convention Logic

Hamid Reza Mohaqeq and Abbas Vafaei

Department of Computer Engineering, University of Isfahan, Isfahan, Iran

Abstract: Although QCA (Quantum-dot Cellular Automata) has been introduced as a new kind of technology for over a decade, it still continued to be so and its merits and flaws were yet under study for future practical use. One of the problems of this technology has been the dependency of its circuit timing to its layout. An asynchronous design methodology for QCA has been offered to solve this problem. The proposed methodology uses NCL (Null Convention Logic) to approach this issue. Since asynchronous registers played an important role in NCL methodology, to ease the problem this work was aimed to design asynchronous registers and employed them to construct a delay insensitive serial adder. By using the results obtained so far an approximate formula evaluate the required cell number and hence space is proposed in this paper which can be used for estimation proposes in the future QCA system design.

Key words: Timing . layout . register . GALS (Globally Asynchronous Locally Synchronous)

INTRODUCTION

QCA is one of the innovating technologies for Nano scale computation. With the present difficulties in reduction of sizes in transistor circuits, QCA has been proposed as a kind of replacement for such instances [1, 2].

QCA structures are constructed as an array of quantum cells within which, every cell has an electrostatic interaction with its neighboring cells. QCA applies a new form of computation, where polarization rather than the traditional current, contains the digital information. In this trend, instead of interconnecting wires, the cells transmit information throughout the circuit [3-5].

QCA circuits require a clock, not only to synchronize and control information flow, but also to provide the power to run the circuit since there is no external source for powering cells [6, 7]. With the use of the four-phase clocking scheme in controlling cells, QCA processes and forwards information within cells in an arranged timing scheme. This clocking method makes the design of QCA different from CMOS circuits and causes new challenges in the designing of circuits. The most important challenge is the dependency of a QCA circuit's overall timing to its layout. This challenge is referred to, as the "Layout = Timing" problem [8, 9]. To solve this problem, a novel self-timed QCA circuit design methodology has been proposed in [10, 11]. This method applies a delay-insensitive data encoding scheme, called NCL in QCA circuits to omit "Layout = Timing" problem [12]. The QCA layout of a delay-insensitive full adder has been offered and the layout efficiency of the asynchronous multi-bit adders using this structure compared with the synchronous ones has been studied in [13].

Asynchronous registers play an important role in a delay-insensitive NCL circuit in the communication management among the circuit components [14]. In this paper, after designing an asynchronous register in QCA, structure of a delay-insensitive bit serial adder will be offered to look into the role of registers in a delay-insensitive QCA circuit.

This paper is organized as follows: section 2 describes the preliminaries of QCA technology: Section 3 describes the proposed method for designing asynchronous QCA circuits: Section 4 explains the NCL logic and its main structures; In section 5 by constructing NCL gates in QCA, the structure of a register is presented. The layout of a delay-insensitive serial adder in considering the interaction of circuit components is also proposed.

MATERIALS AND METHODS

QCA cell: The fundamental unit in QCA circuits is a quantum cell. As demonstrated in Fig. 1, a quantum cell is constructed with four quantum dots that are positioned at the corners of a square. Each cell is charged with two extra electrons which can tunnel between the dots inside the cell. Due to columbic repulsion the two excess electrons are forced to occupy the opposite diagonal vertexes. This forms two different polarization states (i.e.,-1, 1) for each cell. Using these polarizations to represent logic value 0 and 1, binary information is encoded in QCA, as shown in Fig. 1.

Corresponding Author: Dr. Hamid Reza Mohaqeq, Department of Computer Engineering, University of Isfahan, Isfahan, Iran

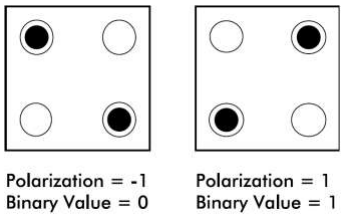

Fig. 1: Encoding binary information in QCA

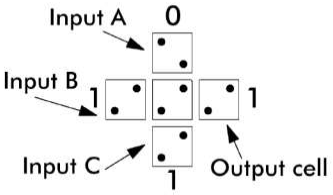

Fig. 2: Majority gate

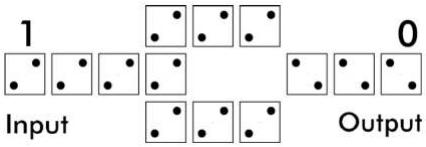

Fig. 3: QCA inverter gate

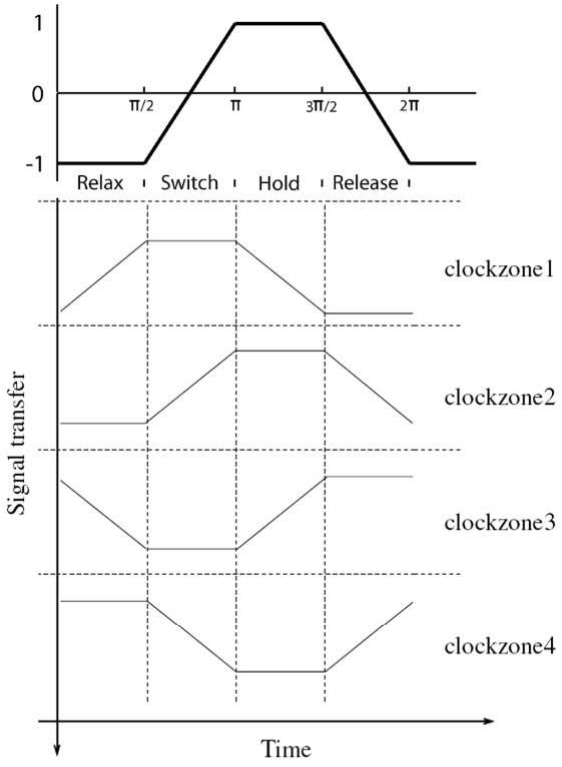

Fig. 4: Four-phase clock signal and its four different zones

The electrostatic repulsion between electrons causes the synchronization of neighboring cells [15, 16]. Therefore, polarization of one cell is determined by the effect of the polarization of its neighboring cells, so an array of QCA cells will be able to propagate binary information through the circuit like a wire [17].

One of the two basic logic gates in QCA is the majority gate. A majority gate with the logic function of M (A, B, C) = AB+AC+BC is composed of five cells as shown in Fig. 2.

By setting one of the inputs of this gate permanently to 0 or 1 and and OR functions will be formed in QCA [18, 19].

The other basic gate is the inverter gate that is shown in Fig. 3.

QCA clocking: As mentioned before, to synchronize and control information flow and to provide the power to run the circuit, clocking is needed in QCA. QCA applies a clocking scheme with four different phases which are: Switch, Hold, Release and Relax [20]. As shown in Fig. 4, four identical clock signals, each shifted in phase by 90 degrees are applied to four adjacent groups of cells to control the flow of information. Each group is considered as a clocking zone. All cells within the same zone are in the same phase and switch simultaneously.

In the Relax phase where clock signal is high, the potential barriers between dots are low and the electrons spread out in the cell. In this phase, cells have no polarization. During the Switch phase, the potential barriers are raised and the electrons are localized while the clock signal is lowered, thus the polarization of a cell is shaped, based on the influence of its neighbors. The potential barriers are held high in the Hold phase and the cell maintains its polarization. This polarization is used as an input for the next zone. Raising the clock signal during the Release phase, barriers are lowered again, therefore the electrons gain mobility and the cell loses its polarization. By applying this clocking scheme and dividing the circuit to different clocking zones, information is transferred and processed in a pipeline fashion in QCA circuits [21, 22].

QCA design free from Layout/Timing dependency: In order to omit the "Layout = Timing" problem from QCA circuits, an asynchronous QCA circuit design methodology referred to as the GALS design, is proposed in [10, 11]. This methodology integrates a delay-insensitive data encoding scheme called "Null Convention Logic (NCL)" [12] to the global network of QCA gates. As a result, the circuit is made delay-insensitive, while the four-phase clocking scheme is preserved for individual gates. Since four-phase

clocking scheme is a requirement in QCA circuits, the designated layout using NCL gates remain synchronous at gate-level, whereas the "Layout = Timing" problem is eliminated from the circuit.

Null Convention Logic (NCL): NCL is a delay-insensitive data encoding scheme that is used for designing asynchronous circuits. NCL circuits use Data/Null states to show Data or Control values. This separation between Data and control representations supplies a self-timing for the circuit and makes it free from the global timing [12, 23].

The basic building blocks of QCA circuits are threshold gates. These gates use one of the two states, Data or Null, as their inputs and outputs [14, 24]. When the output of a threshold gate is in a Null state, it will not change its state until the specified number of inputs is set in Data state. Once the gate output changes to Data state, it remains in this state until all inputs return to the Null state. This behavior of threshold gates is known as hysteresis property. This property prevents the threshold gates from switching during the intermediate states.

Normally, NCL uses a dual rail signal which consists of two wires, to show information. In this presentation scheme asserting one of the wires is considered as logic 0 and asserting the other one, shows the logic 1. Both line not asserted corresponds to the Null state but two asserted rails simultaneously, are defined as an illegal state.

Threshold gate with hysteresis: A threshold gate with hysteresis property, is denoted with THmn symbol within which, $1 = m < n$ [24]. As shown in Fig. 5, this gate has n input wires. To assert the output of this gate, at least m of n input wires must be asserted. Threshold gates use hysteresis as a mean to control the output transitions and complete transition of inputs back to a Null state, before asserting the output affected by the next state of input Data.

The asynchronous register: One of the important issues in a sequential NCL circuit is the management of the communication and interaction among its combinational components. Asynchronous registers are needed to check completeness of inputs as well as readiness to accept new input Data sets and store the complete set of Data values or the all Null values between components of a circuit. These registers act like the intermediate registers in pipeline structures. As shown in Fig. 6 an asynchronous register consists of an array of threshold gates with hysteresis and a single gate that watches for complete Data sets and all Null states. Each TH22 gate receives one Data wire and one

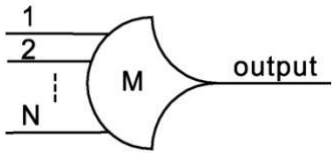

Fig. 5: NCL threshold gate

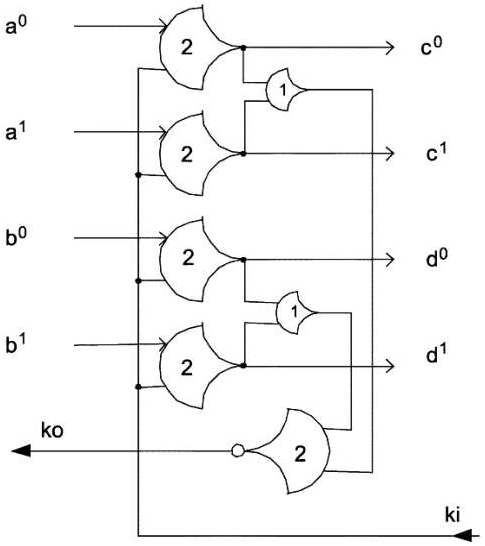

Fig. 6: Two-bit asynchronous register

Control wire (k_i). The single threshold gate (detection gate) with an inverted output is used to detect the completion of the register outputs and create a control signal (k_o). When detection gate senses a complete set of Data values, it asserts Data and when it senses all Null values; it transitions to Null state.

Control signals are used to manage the interaction among registers in a circuit. The control input for each gate comes from the detection gate of the next register. When a calculated Data set through a combinational circuit is received and stored by next register, it sends back a Null acknowledgment through the control line and declares that the previous register can send the Null state. When the detection gate senses all Null values it transitions its control line to Data, to indicate that it has received and stored the Null state and the previous register can pass a new Data set.

A delay-insensitive QCA serial adder design: Asynchronous registers are used to manage the interaction among the components of a circuit as well as the asynchronous flow of information through the circuit. An asynchronous register passes Data and Null among combinational components alternatively. It also indicates the completion of inputs to previous registers after detection. Applying a group of asynchronous registers in NCL circuits makes them delay-insensitive.

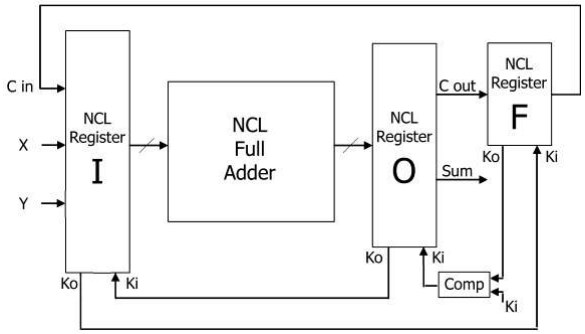

Fig. 7: Structure of a delay-insensitive serial adder

It also allows Data to be pipelined and fed back to the circuit. NCL employs a group of registers that are organized as a ring to feed back Data in a circuit [25]. This registration technique controls the alternate feedback of Data/Null states through the circuit.

A bit serial adder is constructed by using a one-bit full adder beside a feedback line for transmitting the carry output of each stage to the input of the next. For designing a delay-insensitive serial adder, a delay-insensitive one-bit full adder, in addition to a group of asynchronous registers for the feedback of the carry output is required. The structure of proposed delay-insensitive adder is shown in Fig. 7. Three asynchronous registers are used in this structure to feed back carry output and flow Data or Null states through the circuit. The first two registers ('I' and 'O' registers) control and detect completion of full adder inputs and outputs, respectively. The third register ('F' register) is used as an intermediate register, to store and feedback carry output to 'I' register.

Passage of Data or Null through a register occurs with request for Data or Null from the register of the next stage, after detection of complete Data or Null state in that register. In this scheme the full adder will receive Null or Data alternatively with passage of Null or Data and carry through 'I' register, after detection of Null or sum and carry completion in 'O' register. The Null or next result of adder passes through the 'O' register, after the detection of completed saved carry in intermediate register and the reception of a request for the next sum result from the register of next stage. The feedback register ('F') passes Null or new carry output after passing of the current carry through the 'I' register. To have a correct result for adder in this scheme, 'I', 'O' and 'F' registers must be initialized with Null, Null and Zero respectively.

Design of a delay-insensitive serial adder includes designing a delay-insensitive full adder and the needed registers. In this paper, design of full adder is based on the proposed full adder layout in [13] with changes in its timing. As shown in Fig. 6, in order to design the required registers, TH12 and TH22 gates are needed. The proposed QCA layouts for these two gates are shown in Fig. 8.

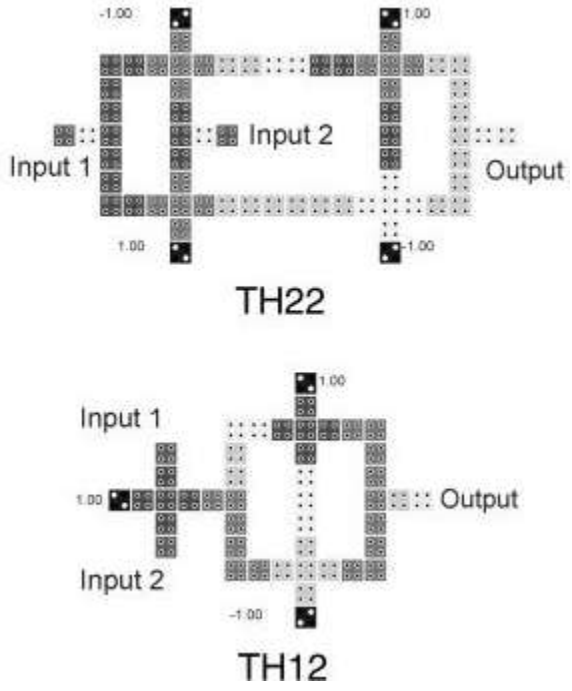

Fig. 8: Layouts of TH12 and TH22 gates

A two-bit register composed of four TH22 gates to receive inputs and the k_i control signal, two TH12 gates to detect presence of Data and a single TH22 gate with an inverter to detect completion of inputs and producing the k_o control signal.

The layout of a two-bit asynchronous QCA register is shown in Fig. 9.

The QCA layout of this adder is shaped by applying layouts of the single bit full adder and three registers. This layout is shown in Fig. 10, in which a 2:1 multiplexer is used for carry zero initialization.

CONCLUSION

A formula for calculating the number of cells and hence the required space for full adders has been proposed in [13]. By using the results achieved in this work a more precise formula for determining the same parameters for a delay-insensitive adder can be deduced.

In [13] the required cells for asynchronous registers are not taken into account. Each asynchronous register for every data line needs two TH22 and one TH12 gates. In proposed design each of the TH22 and TH12 gates need 66 and 44 cells respectively.

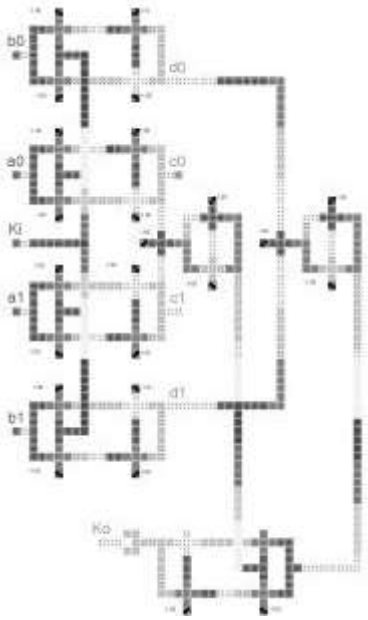

Fig. 9: The QCA two-bit asynchronous register layout

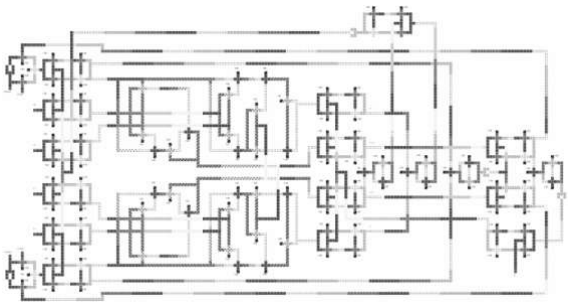

Fig. 10: QCA delay-insensitive serial adder layout

A recognizing gate is also needed whose number of inputs and threshold level depends on the number of adder output. Wires are also needed for connecting deferent part of the circuit together and n-bit adder needs 2n wires in order to be connected to its own register.

If the number of necessary cells for obtaining the outputs of an n-bit adder is denoted by Tn and the average number of cells needed for each connection wire is taken as w then the total number of cells required to build up an n-bit delay-insensitive full adder can be expressed as:

$$1158\times n+2\times c\times(n-1)+2\times 66\times n+1\times 44\times n+$$
$$2\times w\times n+Tn=1334\times n+2\times c\times(n-1)+2\times w\times n+Tn$$

In this formula the value 1158 corresponds to the number of cells needed to fabricate the full adder presented in [13] and c is the average number of cells needed to connect each adder stage to its higher level neighbor. By not considering values w, c and Tn one can obtain the following results which are comparable with those obtained in [13]

Although obtained number is more than what was predicted in [13] but for large values of n of n-bit delay-insensitive adders over proposed circuit has distinctive advantage in terms of required number of cells over ordinary adder designs.

n	8	16	30	32	64	128
Cell Count	10672	21344	40020	42688	85376	172032

RESULTS AND DISCUSSION

Asynchronous registers manage the asynchronous flow of Data and Null states among the Null Convention Logic circuits. With these registers, delay-insensitive systems can be built. Applying at least one register in input or output of each component is needed in delay-insensitive systems.

In previous works on asynchronous QCA circuits, the role of registers has not been considered and a complete layout for studying the communication among the components of such circuits has not yet been presented. Designing a QCA asynchronous register and the structure of a delay-insensitive QCA serial adder (in which the flow of information among the circuit components is controlled by registers) is presented for the first time in this paper.

The Layouts of synchronous and asynchronous QCA circuits were compared in [13], but as registers were not considered in this comparison, the results are not quit exact. However by considering the cell count of registers, the layout space of synchronous and asynchronous QCA circuits can be compared more accurately.

REFERENCES

1. ITRS, 2003. International Technology Roadmap for Semiconductors, http://public.itrs.net/Files/2003ITRS/Home2003.htm
2. Lent, C.S., P.D. Tougaw, W. Porod and G.H. Bernstein, 1993. Quantum cellular automata. Nanotechnology, 4: 49-57.
3. Townsend, W.J. and J.A. Abraham, 2004. Complex Gate Implementations for Quantum Dot Cellular Automata. 4th IEEE Conference on Nanotechnology, pp: 625-627.
4. Niemier, M.T. and P.M. Kogge, 2001. Exploring and exploiting wire level pipelining in emerging technologies. International Symposium of Computer Architecture, pp: 166-177.

5. Henderson, S.C, E.W. Johnson, J.R. Janulis and P.D. Tougaw, 2004. Incorporating standard CMOS design process methodologies into the QCA logic design process. IEEE Transactions on Nanotechnology, 3 (1): 2-9.
6. Kummamuru, R., J. Timler, G. Toth, C. Lent, R. Ramasubramaniam, A. Orlov, G. Bernstein and G. Snider, 2002. Power Gain in a Quantum-Dot Cellular Automata Latch. Applied Physics Letters, 81: 1332-1334.
7. Timler, J. and C. Lent, 2002. Power Gain and Dissipation in Quantum-Dot Cellular Automata. Journal of Applied Physics, 91: 823-831.
8. Niemier, M. and P. Kogge, 2001. Problems in Designing with QCAs: Layout Equals Timing. International. Journal of Circuit Theory and Applications, 29: 49-62.
9. Ntonelli, D.A., D.Z. Chen, T.J. Dysart, X.S. Hu, A.B. Kahng, P.M. Kogge, R.C. Murphy and M.T. Niemier, 2004. Quantum-Dot Cellular Automata (QCA) Circuit Partitioning: Problem Modeling and Solutions. The 41st Design Automation Conference, pp: 363-368.
10. Choi, M and, N Park, 2005. Locally Synchronous, Globally Asynchronous Design for Quantum-Dot Cellular Automata (LSGA QCA). IEEE International Conference on Nanotechnology, pp: 121-124.
11. Choi, M., Z. Patitz, B. Jin, F. Tao, N. Park and M. Choi, 2007. Designing layout-timing independent quantum-dot cellular automata (QCA) circuits by global asynchrony. Journal of Systems Architecture, 53 (9): 551-567.
12. Yuan, J.S. and W. Kuang, 2004. Teaching Asynchronous Design in Digital Integrated Circuits. IEEE Transactions on Education, 47 (3): 397-404.
13. Choi, M., M. Choi, Z. Patitz and N. Park, 2006. Efficient and Robust Delay-Insensitive QCA (Quantum-Dot Cellular Automata) Design. 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, pp: 80-88.
14. Fant, K.M. and S.A. Brandt, 2002. Null Convention Logic™, Theseus Research Inc.
15. Toth, G. and C. Lent, 2001. Quantum computing with quantum-dot cellular automata. Physical Review, 63 (5): 052315(9).
16. Orlov, A., I. Amlani, G. Toth, G. Bernstein and G. Snider, 1999. Experimental demonstration of a binary wire for quantum-dot cellular automata. Applied Physics Letters, 74: 2875-2877.
17. Lent, C.S. and P.D. Tougaw, 1993. Lines-of interacting quantum-dot cells: A binary wire. Journal of Applied Physics, 74 (10): 6227-4233.
18. Tougaw, P.D. and C.S. Lent, 1994. Logical devices implemented using quantum cellular automata. Journal of Applied Physics, 75 (3): 1818-1825.
19. Snider, G., A. Orlov, I. Amlani, G. Bernstein, C. Lent, J. Merz and W. Porod, 1999. Quantum-dot cellular automata: Line and majority logic gate. Japanese. Journal of Applied. Physics, 38: 7227-7229.
20. Hennessy, K. and C.S. Lent, 2001. Clocking of molecular quantum-dot cellular automata. J. Vac. Sci. Technology, 19 (5): 1752-1755.
21. Toth, G. and C.S. Lent, 1999. Quasiadiabatic switching for metal-island quantum-dot cellular automata. J. Appl. Phys., 85 (5): 2977-2984.
22. Lent, C.S. and P. Douglas Tougaw, 1997. A device architecture for computing with quantum dots. Proceedings of the IEEE, 85 (4): 541-557.
23. Fant, K.M. and S.A. Brandt, 1996. Null convention logic: A complete and consistent logic for asynchronous digital circuit synthesis. International Conference on Application Specific Systems Architectures and Processors, pp: 261-273.
24. Smith, S.C, 2001. Gate and throughput optimizations for null convention self-timed digital circuits. Ph.D. Dissertation, School of Electrical Engineering and Computer Science, University of Central Florida.
25. Sparso, J., J. Staunstrup and M. Dantzer-Sorensen, 1992. Design of delay insensitive circuits using multi-ring structures. In Proceedings of the European Design Automation Conference, pp: 15-20.25.