International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 11 Issue 6 –NOVEMBER 2014.

AREA & DELAY EFFICIENT ADDERS FOR ARITHMETIC APPLICATIONS USING QCA

PADIGALA SRINIVASARAO[*1] and K.VENKANNA[#2]

[*]*Student, Dept of ECE, Sree Vahini Institute of Science and Technology, Tiruvuru., A.P, India*

[#]*Asst Professor, Dept of ECE, Sree Vahini Institute of Science and Technology, Tiruvuru., A.P, India dist*

[1]psr.0449@gmail.com

[2]venkanna.kota3@gmail.com

Abstract— As transistors decrease in size more and more of them can be accommodated in a single die, thus increasing chip computational capabilities. However, transistors cannot get much smaller than their current size. The quantum-dot cellular automata (QCA) approach represents one of the possible solutions in overcoming this physical limit, even though the design of logic modules in QCA is not always straightforward. In this brief, we propose a new adder that outperforms all state-of-theart competitors and achieves the best area-delay tradeoff. The above advantages are obtained by using an overall area similar to the cheaper designs known in literature. The 64-bit version of the novel adder spans over 18.72 µm2 of active area and shows a delay of only nine clock cycles, that is just 36 clock phases.

Index Terms—Adders, nanocomputing, quantum-dot cellular automata (QCA).

I. INTRODUCTION

Quantum-dot cellular automata (QCA) is an attractive emerging technology suitable for the development of ultradense low-power high-performance digital circuits [1]. For this reason, in the last few years, the design of efficient logic circuits in QCA has received a great deal of attention. Special efforts are directed to arithmetic circuits [2]–[16], with the main interest focused on the binary addition [11]–[16] that is the basic operation of any digital system. Of course, the architectures commonly employed in traditional CMOS designs are considered a first reference for the new design environment. Ripple-carry (RCA), carry look-ahead (CLA), and conditional sum adders were presented in [11]. The carry-flow adder (CFA) shown in [12] was mainly an improved RCA in which detrimental wires effects were mitigated. Parallel-prefix architectures, including Brent–Kung (BKA), Kogge–Stone, Ladner–Fischer, and Han–Carlson adders, were analyzed and implemented in QCA in [13] and [14]. Recently, more efficient designs were proposed in [15] for the CLA and the BKA, and in [16] for the CLA and the CFA.

In this brief, an innovative technique is presented to implement high-speed low-area adders in QCA. Theoretical formulations demonstrated in [15] for CLA and parallel-prefix adders are here exploited for the realization of a novel 2-bit addition slice. The latter allows the carry to be propagated through two subsequent bit-positions with the delay of just one majority gate (MG). In addition, the clever top level architecture leads to very compact layouts, thus avoiding unnecessary clock phases due to long interconnections. An adder designed as proposed runs in the RCA fashion, but it exhibits a computational delay lower than all state-of the-art competitors and achieves the lowest area-delay product (ADP).

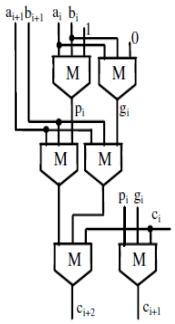

Fig. 1. Novel 2-bit basic module

II. BACKGROUND

A QCA is a nanostructure having as its basic cell a square four quantum dots structure charged with two free electrons able to tunnel through the dots within the cell [1]. Because of Coulombic repulsion, the two electrons will always reside in opposite corners. The locations of the electrons in the cell (also named polarizations *P)* determine two possible stable states that can be associated to the binary states 1 and 0.

Although adjacent cells interact through electrostatic forces and tend to align their polarizations, QCA cells do not have intrinsic data flow directionality. To achieve controllable data directions, the cells within a QCA design are partitioned into

the so-called clock zones that are progressively associated to four clock signals, each phase shifted by 90°. This clock scheme, named the zone clocking scheme, makes the QCA designs intrinsically pipelined, as each clock zone behaves like a D-latch [8].

QCA cells are used for both logic structures and interconnections that can exploit either the coplanar cross or the

bridge technique [1], [2], [5], [17], [18]. The fundamental logic gates inherently available within the QCA technology are the inverter and the MG. Given three inputs *a*, *b*, and *c*, the MG performs the logic function reported in (1) provided that all input cells are associated to the same clock signal clk*x* (with *x* ranging from 0 to 3), whereas the remaining cells of the MG are associated to the clock signal clk $_{x+1}$

$$M(abc) = a \cdot b + a \cdot c + b \cdot c.$$

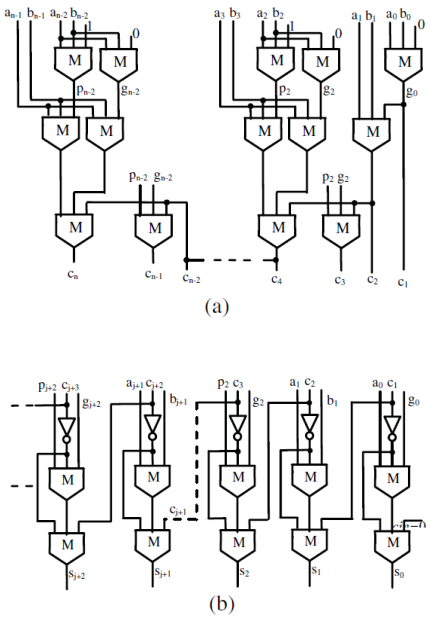

Fig. 2. Novel *n*-bit adder (a) carry chain and (b) sum block.

Several designs of adders in QCA exist in literature. The RCA [11], [13] and the CFA [12] process n-bit operands by cascading n full-adders (FAs). Even though these addition circuits use different topologies of the generic FA, they have a carry-in to carry-out path consisting of one MG, and a carry-in to sum bit path containing two MGs plus one inverter. As a consequence, the worst case computational paths of the n-bit RCA and the n-bit CFA consist of (n+2) MGs and one inverter. A CLA architecture formed by 4-bit slices was also presented in [11]. In particular, the auxiliary propagate and generate signals, namely pi = ai + bi and gi = ai · bi , are computed for each bit of the operands and then they are grouped four by four. Such a designed n-bit CLA has a computational path composed of 7+4×(log4 n) cascaded MGs and one inverter. This can be easily verified by observing that, given the propagate and generate signals (for which only one MG is required), to compute grouped propagate and grouped generate signals; four cascaded MGs are introduced in the computational path. In addition, to compute the carry signals, one level of the CLA logic is required for each factor of four in the operands word-length. This means that, to process nbit addends, log4 n levels of CLA logic are required, each contributing to the computational path with four cascaded MGs. Finally, the computation of sum bits introduces two further cascaded MGs and one inverter.

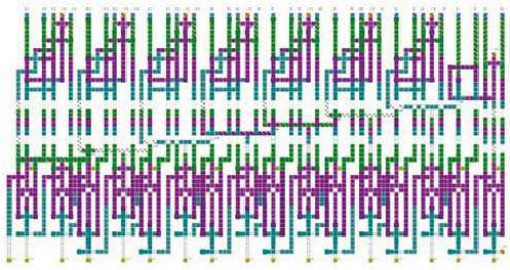

Fig. 3. Novel 16-bit adder

III. NOVEL QCA ADDER

To introduce the novel architecture proposed for implementing ripple adders in QCA, let consider two n-bit addends

A = an−1, . . . , a0 and B = bn−1, . . . , b0 and suppose that for the i th bit position (with i = n − 1, . . . , 0) the auxiliary propagate and generate signals, namely pi = ai + bi and gi = ai · bi , are computed. ci being the carry produced at the generic (i−1)th bit position, the carry signal ci+2, furnished at the (i+1)th bit position, can be computed using the conventional CLA logic reported in (2). The latter can be rewritten as given in (3), by exploiting Theorems 1 and 2 demonstrated in [15]. In this way, the RCA action, needed to propagate the carry ci through the two subsequent bit positions, requires only one MG. Conversely, conventional circuits operating in the RCA fashion, namely the RCA and the CFA, require two cascaded MGs to perform the same operation. In other words, an RCA adder designed as proposed has a worst case path almost halved with respect to the conventional RCA and CFA.

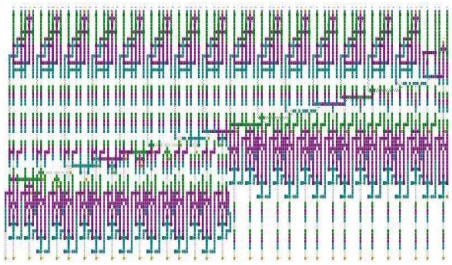

Fig. 4. Novel 32-bit adder.

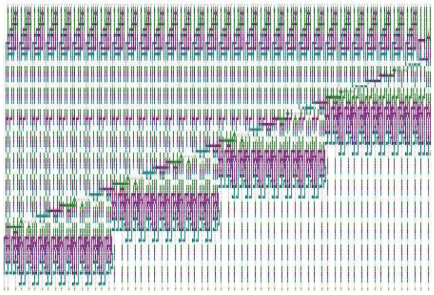

Fig. 5. Novel 64-bit adder.

Equation (3) is exploited in the design of the novel 2-bit module shown in Fig. 1 that also shows the computation of

the carry *ci*+1 = *M(pi gici)*. The proposed *n*-bit adder is then implemented by cascading *n*/2 2-bit modules as shown in Fig. 2(a). Having assumed that the carry-in of the adder is *cin* = 0, the signal *p*0 is not required and the 2-bit module used at the least significant bit position is simplified. The sum bits are finally computed as shown in Fig. 2(b). It must be noted that the time critical addition is performed when a carry is generated at the least significant bit position (i.e., g0 = 1) and then it is propagated through the subsequent bit positions to the most significant one. In this case, the first 2-bit module computes *c*2, contributing to the worst case computational path with two cascaded MGs. The subsequent 2-bit modules contribute with only one MG each, thus introducing a total number of cascaded MGs equal to *(n* − 2*)*/2. Considering that further two MGs and one inverter are required to compute the sum bits, the worst case path of the novel adder consists of *(n*/2*)* + 3 MGs and one inverter.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Temperature	1 K
Relaxation time	1×10^{-15} s
Time step	1×10^{-16} s
Total simulation time	7×10^{-11} s
Radius of effect	80 nm
Relative permittivity	12.9
Layer separation	11.5 nm

IV. RESULTS

The proposed addition architecture is implemented for several operands word lengths using the QCA Designer tool [16] adopting the same rules and simulation settings used in [11]–[16]. The QCA cells are 18-nm wide and 18-nm high; the cells are placed on a grid with a cell center-to-center distance of 20 nm; there is at least one cell spacing between adjacent wires; the quantum-dot diameter is 5 nm; the multilayer wire crossing structure is exploited; a maximum of 16 cascaded cells and a minimum of two cascaded cells per clock zone are assumed. The coherence vector engine is used for simulations with the options shown in Table I.

Layouts for the 16-, 32- and 64-bit versions of the novel adder are shown in Figs. 3–5, respectively. Simulation results for the 64-bit adder is shown in Fig. 6. There, the carry out bit is included in the output sum bus. Because of the limited QCA Designer graphical capability, input and output busses are split into two separate more significant and less significant busses. The polarization values of few single output signals (i.e., *sum*64, *sum*32, *sum*31, and *sum*). Simulations performed on 32- and 64-bit adders have shown that the first valid result is outputted after five and nine latency clock cycles, respectively. As an example, the 20 clock phases (or five cycles delay) of the 32-bit adder are as follows: one clock phase is needed for inputs acquisition; the carry *c*2 related to the least significant bit positions is then computed within the two subsequent clock phases; 15 phases are required for the carry propagation through the remaining bit positions; finally, two more phases are needed for the sum computation. Critical path consistencies and post layout characteristics, such as cell count, overall size, delay, number of clock phases, and ADP, are shown in Table II for all the compared adders. The number of cascaded MGs within the worst case computational path directly impacts on the achieved.

V. CONCLUSION

A new adder designed in QCA was presented. It achieved speed performances higher than all the existing QCA adders, with an area requirement comparable with the cheap RCA and CFA demonstrated in [13] and [16]. The novel adder operated in the RCA fashion, but it could propagate a carry signal through a number of cascaded MGs significantly lower than conventional RCA adders. In addition, because of the adopted basic logic and layout strategy, the number of clock cycles required for completing the elaboration was limited. A 64-bit binary adder designed as described in this brief exhibited a delay of only nine clock cycles, occupied an active area of 18.72 μm2, and achieved an ADP of only 168.48.

REFERENCES

[1] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernestein, "Quantum cellular automata," *Nanotechnology*, vol. 4, no. 1, pp. 49–57, 1993.

[2] M. T. Niemer and P. M. Kogge, "Problems in designing with QCAs: Layout = Timing," *Int. J. Circuit Theory Appl.*, vol. 29, no. 1, pp. 49–62, 2001.

[3] J. Huang and F. Lombardi, *Design and Test of Digital Circuits by Quantum-Dot Cellular Automata*. Norwood, MA, USA: Artech House, 2007.

[4] W. Liu, L. Lu, M. O'Neill, and E. E. Swartzlander, Jr., "Design rules for quantum-dot cellular automata," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 2361–2364.

[5] K. Kim, K. Wu, and R. Karri, "Toward designing robust QCA architectures in the presence of sneak noise paths," in *Proc. IEEE Design, Autom. Test Eur. Conf. Exhibit.*, Mar. 2005, pp. 1214–1219.

[6] K. Kong, Y. Shang, and R. Lu, "An optimized majority logic synthesis methology for quantum-dot cellular automata," *IEEE Trans. Nanotechnol.*, vol. 9, no. 2, pp. 170–183, Mar. 2010.

[7] K. Walus, G. A. Jullien, and V. S. Dimitrov, "Computer arithmetic structures for quantum cellular automata," in *Proc. Asilomar Conf. Sygnals, Syst. Comput.*, Nov. 2003, pp. 1435–1439.

[8] J. D. Wood and D. Tougaw, "Matrix multiplication using quantumdot cellular automata to implement conventional microelectronics," *IEEE Trans. Nanotechnol.*, vol. 10, no. 5, pp. 1036–1042, Sep. 2011.

[9] K. Navi, M. H. Moaiyeri, R. F. Mirzaee, O. Hashemipour, and B. M. Nezhad, "Two new low-power full adders based on majority-not gates," *Microelectron. J.*, vol. 40, pp. 126–130, Jan. 2009.

[10] L. Lu, W. Liu, M. O'Neill, and E. E. Swartzlander, Jr., "QCA systolic array design," *IEEE Trans. Comput.*, vol. 62, no. 3, pp. 548–560, Mar. 2013.

[11] H. Cho and E. E. Swartzlander, "Adder design and analyses for quantum-dot cellular automata," *IEEE Trans. Nanotechnol.*, vol. 6, no. 3, pp. 374–383, May 2007.

[12] H. Cho and E. E. Swartzlander, "Adder and multiplier design in quantum-dot cellular automata," *IEEE Trans. Comput.*, vol. 58, no. 6, pp. 721–727, Jun. 2009.

[13] V. Pudi and K. Sridharan, "Low complexity design of ripple carry and Brent–Kung adders in QCA," *IEEE Trans. Nanotechnol.*, vol. 11, no. 1, pp. 105–119, Jan. 2012.

[14] V. Pudi and K. Sridharan, "Efficient design of a hybrid adder in quantumdot cellular automata," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 9, pp. 1535–1548, Sep. 2011.