LOW DESIGN IMPLEMENTATION AREA RIPPLE CARRY ADDER IN QUANTUM DOT CELLULAR AUTOMATA

1. GURRALA LAKSHMI APARNA, 2.A.B. CHANDRIKA
1. PG Scholar, Dept. of ECE, , VSM College of Engineering, Ramachandrapuram, East Godavari, A.P
2.Assistant Professor, Dept. of ECE, , VSM College of Engineering, Ramachandrapuram, East Godavari, A.P.

ABSTRACT: Quantum-dot cellular automata (QCA) are an attractive emerging technology suitable for the development of ultra-dense-low-power high-performance digital circuits. Efficient solutions have recently been proposed for several arithmetic circuits, such as adders, multipliers, and adders. Nevertheless, since the design of digital circuits in QCA still poses several challenges, novel implementation strategies and methodologies are highly desirable. This paper proposes a new design approach oriented to the implementation of binary adders in QCA. New formulations of basic logic equations required to perform the comparison function are proposed. The new strategy has been exploited in the design of two different adder architectures and for several operands word lengths. With respect to existing counterparts, the adders proposed here exhibit significantly higher speed and reduced overall area. The proposed scheme, we deal with 32-bit numbers with less number of resources unlike conventional adders, which leads to the realization of low power and area efficient adder. This adder can be widely used in central processing units (CPUs) and microcontrollers.

INTRODUCTION: Quantum dot Cellular Automata (QCA) technology provides a promising opportunity to overcome the approaching limits of conventional CMOS technology. For this reason, in recent years the design of logic circuits based on QCA has received a great deal of attention, and special efforts have been directed towards arithmetic circuits, such as adders, multipliers , and adders. This paper focuses on the design of efficient parallel QCA-based n-bit full adders. The main contribution of this paper is the introduction of a novel design methodology that allows low computational time and very compact layouts to be achieved. In particular, original theorems and corollaries are stated and demonstrated that directly impact on the QCA realizations of some basic Boolean functions used within the adder architectures.

The novel theorems were applied to achieve innovative QCA-based structures of n-bit full adders that were laid out and simulated using the QCA Designer tool for n ranging between 2 and 32. As an example, one of the 32-bit adders designed exploiting the proposed theory is implemented using less than 2800 cells within an overall area of about 2.66 μm2; moreover, it requires only 15 clock cycles to complete the operation.

2. QCA BASED ADDER: There are several QCA designs of adders in the literature. A 1-bit binary adder receives two bits *a* and *b* as inputs and establishes whether they are equal, less than or greater than each other. These possible states are represented through three output signals, here named $A_{\text{eq}} B$, $A_{\text{big}} B$, $B_{\text{big}} A$, that are asserted, respectively, when $a = b$, $a > b$, and $a < b$. Full adders are those that can separately identify all the above cases, whereas non-full adders recognize just one or two of them. As an example, the adder designed in and depicted in Fig. 5(a) can verify only whether $a = b$. Conversely, the circuits shown in Fig. 5.1(b) and (c), proposed , are full adders. The latter also exploits two 1-bit registers *D* to process *n*-bit operands serially from the least significant bit to the most significant one.With the main objective of reducing the number of wire crossings, which is still a big challenge of QCA designs , in the universal logic gate (ULG) $f(y_1, y_2, y_3) = M(M(y_1, y_2, 0), M(y_1, y_3, 1)$,was proposed and then used to implement the adder illustrated in Fig. 1(d). It is worth noting that, two *n*-bit numbers $A_{(n-1:0)} = a_{n-1} \dots a_0 \dots b_0$ can be processed by cascading n instances of the 1-bit adder. Each instance receives as inputs the *i*th bits a_i and b_i (with $i = n - 1, \dots, 0$) of the operands and the signals $A_{\text{big}} B_{(i-1:0)}$ and $B_{\text{big}} A_{(i-1:0)}$ .The former is asserted when the sub word $A_{(i-1:0)} = a_{i-1} \dots a_0$ represents a binary number greater than $B_{(i-1:0)} = b_{i-1} \dots b_0$. In a similar way, ${}^{B}\text{big}\ {}^{A}(i-1:0)$ is set to 1 when ${}^{A}(i-1:0)\ {}^{<B}(i-1:0)$. The outputs $A_{\text{big}} B_{(i:0)}$ and $B_{\text{big}} A_{(i:0)}$ directly feed the next stage.

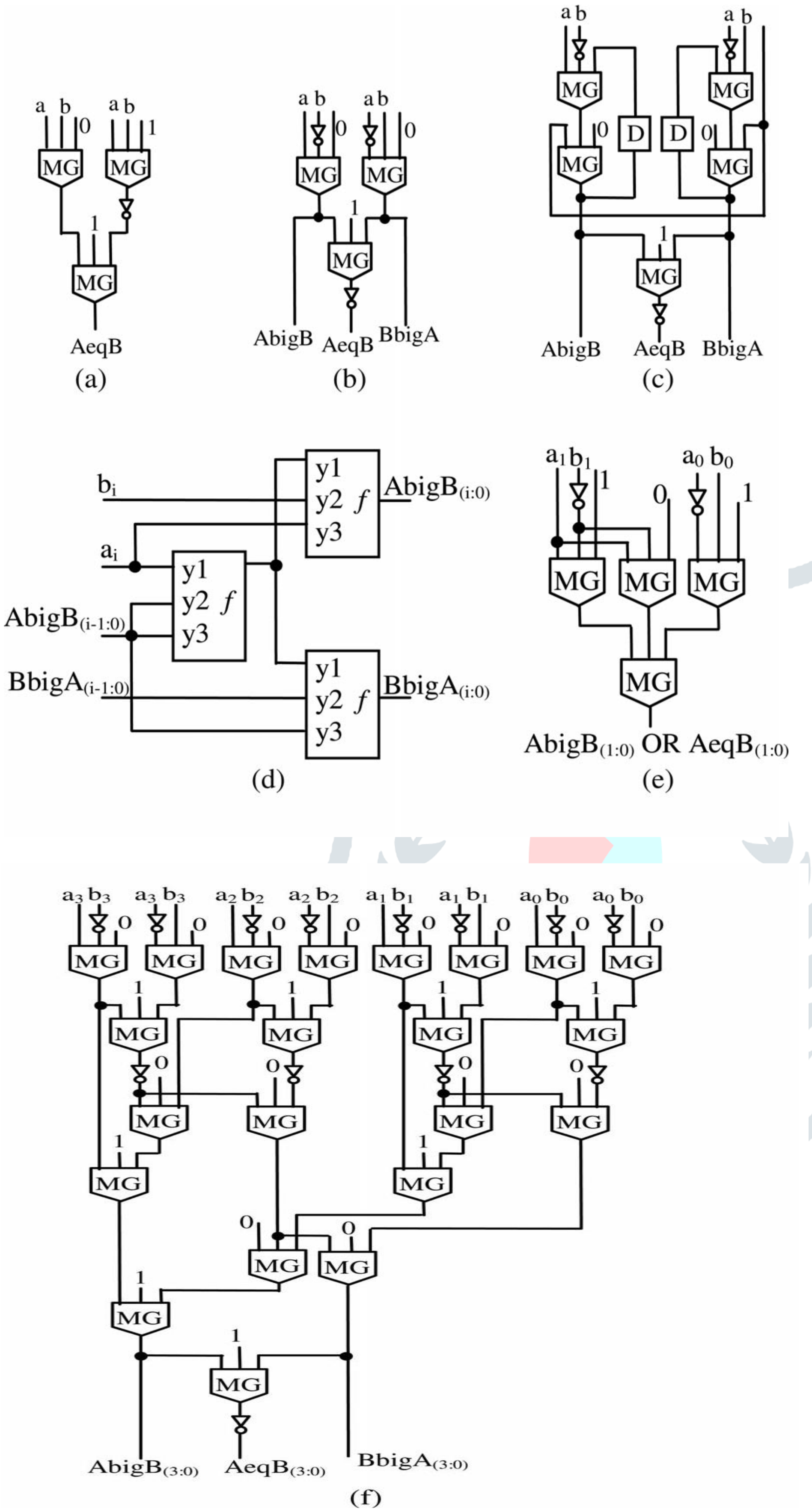
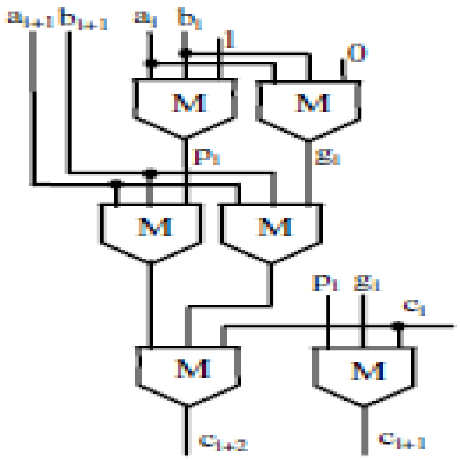


Fig1: QCA based adder presented in:(a),(b),(c),(d), (e),(f)

3. NOVEL QCA ADDERS

The first proposed adder exploits a cascade-based (CB) architecture. To explain better how the overall computation is performed.It shows a possible implementation of a 32-bit adder based on the proposed theory.As it is well known, the number of cascaded MGs within the worst computational path of a QCA design directly affects the delay achieved. In fact, each MG introduces one clock phase in the overall delay. Each instance of T4 introduces one more MG, whereas C2 is responsible for one MG and one inverter. As a consequence, the critical computational path of the novel n-bit CB full adder consists of n/3+ 3 MGs and 2 inverters.This paper allows carry to be propagated through 2 main bit positions with majority gate delay (MG). In addition, clever top-level architecture is needed for very compact layouts, thus avoiding not used clock phases due to long interconnections



. **Fig 2:Novel 2-bit basic module**

This adder was established as an RCA mode, but showed a lower computing delay than all sophisticated and achieved the less product delay area .

4.PROPOSED ARCHITECTURE

The first proposed architecture presented in based on parallel approach and has two output bits (A>*B*), S(i.e. *A*<*B*). The circuit for the 4-bit adder is displayed in Fig. 2 and is slightly a modified version of the traditional adder (which works on bit-weight comparison of two numbers from LSB to MSB) to understand the logic for the proposed architecture, let us consider an example for the comparison of *A*=(10110)2 and *B* =(1100)2. In the first stage, we identify and extract the 1s of first number which have a 0 in the corresponding position of the second number and are allowed to remain. The basic idea behind this is that only such 1s of a number make it greater than the other number. All other bit positions which have a 1 in the corresponding position of the other number, are made 0.This is done for both the numbers in parallel, that is, *A* with respect to *B* (i.e. $A_i\overline{B_i}$) and *B* with respect to A (i.e. Bi, Ai`) , thereby forming two numbers *A'* and *B'* as shown

A = 1 0 1 1 *B* = 1 1 0 0
B = 1 1 0 0 *A* = 1 0 1 1
B'= 0 0 1 1 *A'*= 0 1 0 0

In the second stage, only the most significant 1s of *A'* and *B'* are extracted by giving it higher priority. Other 1s are made 0. This stage incorporates logic similar to the priority logic of a priority encoder. This way two new numbers, *A''* and *B''* are formed as shown below. Due to the priority logic incorporated, the number of 1s in *A''* and *B''* is either one or zero.

B'= 0 0 1 1 *A'* = 0 1 0 0
B''= 0 0 1 1 *A''*=0 1 0 0

In the final stage, from *A''* and *B''* two new signals are extracted. These are H (i.e. *A*>*B*) and S (i.e. *A*<*B*), both are of single bit, obtained by extracting the most significant bit (1) from *A''* and *B''*. If the 1 of *A''* is in a more significant position than that of *B''* or if *B''* has all 0s but *A''* has a 1, then this 1 is used to form output bit H. Similarly, if the 1 of *B''* is in a more significant position than that of *A''* or if *A''* has all 0s but *B''* has a 1, then this 1 is used to form output bit S as follows

A''= 0 0 1 0 *B''*= 0 1 0 0
B''= 0 1 0 0 *A''*= 0 0 1 0
H = 0 S = 1

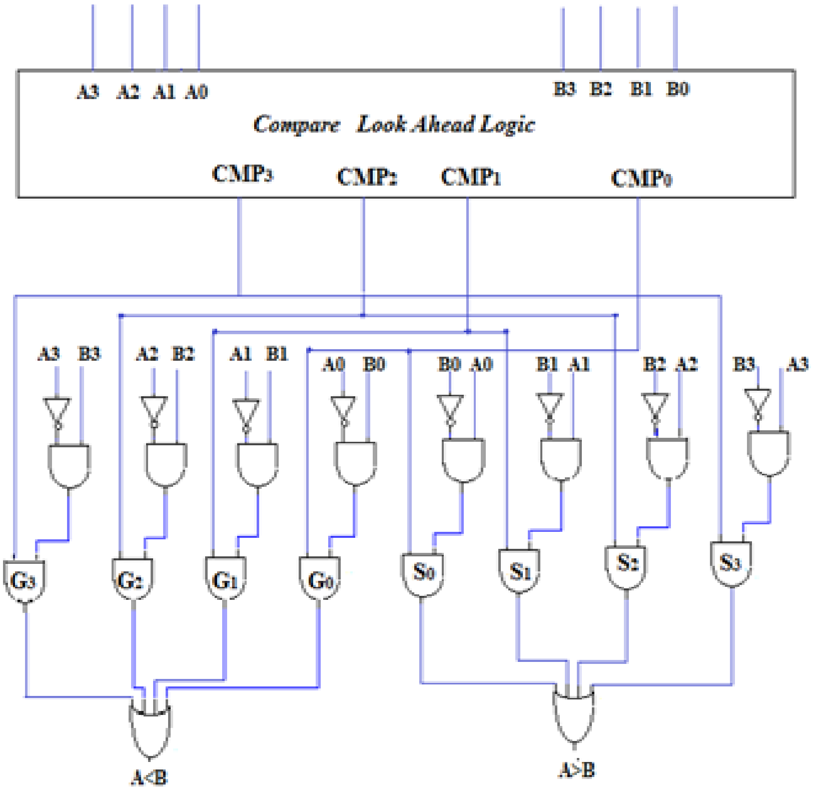


Fig 3: Proposed Architecture.

The schematic for 32-bit level implementation of the traditional and proposed adders is shown in Figure 2.3. The blocks of the first stage compute the comparison result for every 4 bits of the input numbers. The blocks in the second stage take the result of four sets of 4-bit numbers and compute the result for the two 16-bit numbers which are obtained when the four sets of 4-bit numbers are concatenated. This logic is repeated in the third stage where the 2-bit block takes the results of two sets of 16-bit numbers and computes the result for the two 32-bit numbers.

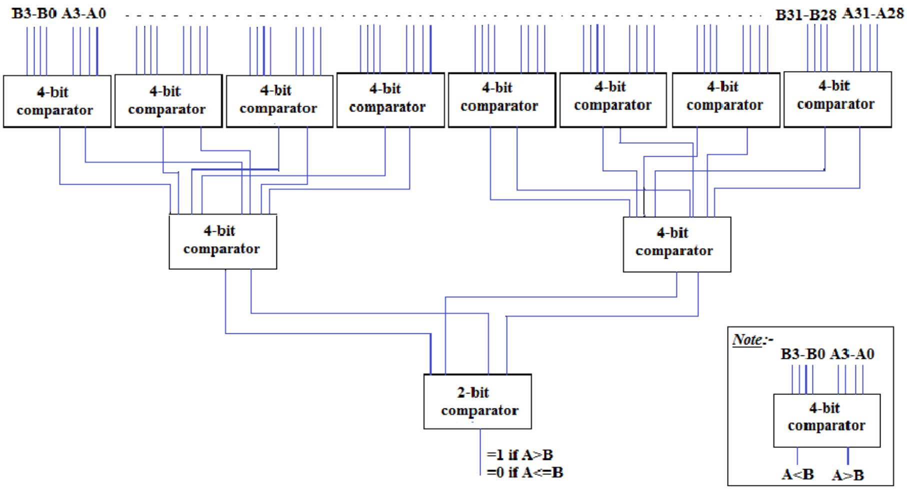
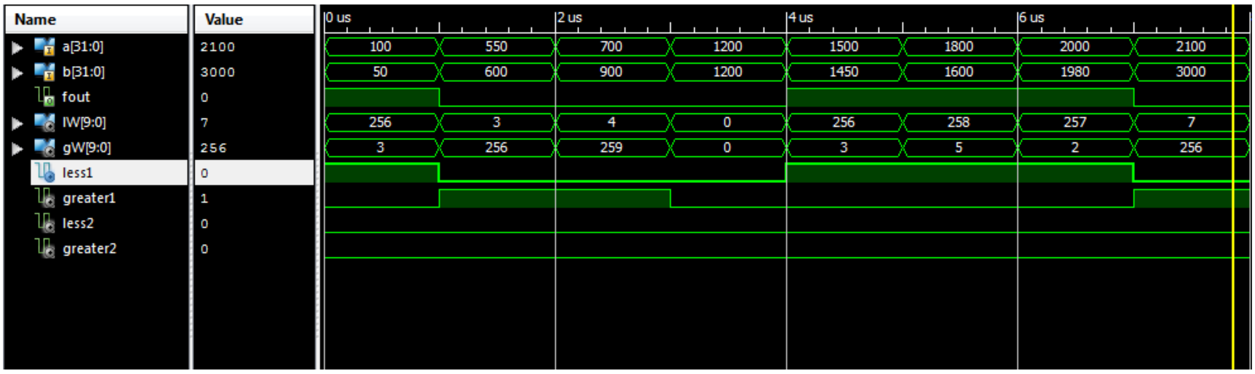


Fig 6:bit tree structure adder

In the 32-bit level implementation of both the proposed adders, a modified 2-bit adder module has been utilized. Since the numbers input to the 2-bit adder module are the outputs of 4-bit adders, certain pairs of numbers can never be the input combinations: (10,10), (10,11), (11,10), (11,01), (01,11), (01,01). This is because the (A>B) and (A<B) output bits of the 4- bit adder module can never be 1 at the same time. As a result, the Boolean expression for the (A>B) output of 2-bit adder module becomes:

$$(A>B) = A_{01}+\overline{A}_{01}A_{00}\overline{B}_{01}\overline{B}_{00}$$

5.Result:



6.Comparison table:

	No of 4 input LUT'S Used	Delay(ns)	Power(mw)
EXISTING	61	16.091ns	0.4978mw
PROPOSED	19	14.419ns	0.1551mw

7.CONCLUSION

The technology studied in this project, QCA, reveals to be a strong competitor, along with SET, to in a near future, complement CMOS technology in digital integrated circuits. It must be remembered that analog CMOS technology will be needed, at least, to bound the real analog world to QCA quantum-dots. This QCA technology seams particularly suitable for high throughput and deeply pipelined architectures, given the inherent pipelined operation of a single QCA wire, acting as a chain of latches. Applications such as audio and video stream processing might benefit much with QCA architectures. On the other hand, heavily conditional processing would be penalized, given the high cost of a stall in an extremely deep pipeline.

Regarding the design flow of QCA circuits, from system specifications to physical fabrication, many points have to be tuned to reach a viable alternative for CMOS in the digital domain. Logic synthesis is a key operation that may drastically reduce area and delay the circuit.Once the professional tools currently available are deeply oriented to the most basic logic gates feasible in CMOS (usually NAND gates), this will be an area of great interest. The greatest challenge is, perhaps, to adapt current tools and design flows from current CMOS processes in order to accommodate the special features of QCA, such as gate level synchronization and in wire memory .There may be need to adapt many existing CMOS circuits to QCA, and this may result in the exclusive use of And, Or and Inverter gates in QCA circuits

REFERENCES

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

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

[3] S. Perri and P. Corsonello, "New methodology for the design of efficientbinary addition in QCA," *IEEE Trans. Nanotechnol.*, vol. 11, no. 6,pp. 1192–1200, Nov. 2012.

[4] V. Pudi and K. Sridharan, "New decomposition theorems on majoritylogic for low-delay adder designs in quantum dot cellular automata,"*IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 10, pp. 678–682,Oct. 2012.

[5] K. Walus and G. A. Jullien, "Design tools for an emerging SoCtechnology: Quantum-dot cellular automata," *Proc. IEEE*, vol. 94, no. 6,pp. 1225–1244, Jun. 2006.

[6] S. Bhanja, M. Ottavi, S. Pontarelli, and F. Lombardi, "QCA circuits forrobust coplanar crossing," *J. Electron. Testing, Theory Appl.*, vol. 23,no. 2, pp. 193–210, Jun. 2007

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

[8] V. Pudi and K. Sridharan, "New decomposition theorems on majority logic for low-delay adder design in quantum dot cellular automata," IEEE Trans. Circuits Syst. II, Exp. Brief , vol. 59, no. 10, pp. 678–682, Oct. 2012.

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

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

[11] 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.

[12] S. Perri, P. Corsonello, and G. Cocorullo, "Area-delay efficient binary adders in QCA," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 5, pp. 1174–1179, May 2014.

[13] 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.

[14] S. Perri, P. Corsonello, and G. Cocorullo, “Design of efficient binary comparators in quantum-dot cellular automata,” IEEE Trans. Nanotechnol., vol. 13, no. 2, pp. 192–202, Feb. 2014.

[15] S. Perri and P. Corsonello, “New methodology for the design of effic ient binary addition in QCA,” IEEE Trans. Nanotechnol., vol. 11, no. 6, pp. 1192–1200, Jun. 2012.

[16] C. A. Krygowski et al., “Functional verification of the IBM system z10 processor chipset,” IBM J. Res. Dev., vol. 53, no. 1, pp. 1–11, Jan. 2009.

[17] B. Sinharoy et al., “IBM POWER8 processor core micro architecture,” IBM J. Res. Dev., vol. 59, no. 1, pp. 1–21, Jan./Feb. 2015.

[18] L. Eisen et al., “IBM POWER6 accelerators: VMX and DFU,” IBM J. Res. Dev., vol. 51, no. 6, pp. 1–21, Nov. 2007. [13] Sparc64TM X/X+ Specification Ver. 29.0 Fujitsu, Kawasaki, Japan, Jan. 2015. [Online]. Available: http://www.fujitsu.com

[19] “SilMinds DFP adder units,” SilMinds, Cairo, Egypt. [Online]. Available: http://www.silminds.com/ip-products/dfp-units/56

[20] IEEE Standard for Floating-Point Arithmetic, IEEE Working Group of t
Microprocesser Standards Subcommittee, IEEE Std. 754-1985, 2008.

[21] M. Gladshtein, “Quantum-dot cellular automata serial decimal adder,” IEEE Trans. Nanotechnol., vol. 10, no. 6, pp. 1377–1382, Jun. 2011.

[22] E. M. Schwarz, J. S. Kapernick, and M. FCowlishaw, “Decimal floatingpoint support on the IBM system z10 processor,” IBM J. Res. Dev., vol. 53, no. 1, pp. 1–10, Jan. 2010.

[23] R. D. Kenney and M. J. Schulte, “High-speed multioperand decimal adders,” IEEE Trans. Comput., vol. 54, no. 8, pp. 953–963, Aug. 2005.

[24] C. Tsen, S. Gonzalez-Navarro, M. J. Schulte, and K. Compton, “Hardware designs for binary integer decimal-based rounding,” IEEE Trans. Comput., vol. 60, no. 5, pp. 614–627, May 2011.

[25] L. Han and S. B. Ko, “High-speed parallel decimal multiplication with redundant internal encodings,” IEEE Trans. Comput., vol. 62, no. 5, pp. 956–968, May 2013.

[26] A. A. Bayrakci and A. Akkas, “Reduced delay BCD adder,” in Proc. IEEE Int. Conf. Appl.-Specific, Syst., Archit. Process., Montreal, QC, Canada, Jul. 2007, pp. 266–271.

[27] C. Sundaresan, C. V. S. Chaitanya, P. R. Venkateswaran, S. Bhat, and J. M. Kumar, “Modified reduced delay BCD adder,” in Proc. IEEE Int. Conf. Biomed. Eng. Informat., Shanghai, China, Oct. 2011, pp. 2148–2151.

[28] W. Liu, L. Lu, M. O’Neill, and E. E. Swartzlander, “Cost-efficient decimal adder design in quantum-dot cellular automata,” in Proc. IEEE Int. Symp. Circuits Syst., Seoul, South Korea, May 2012, pp. 1347–1350.

[29] M. Taghizadeh, M. Askari, and K. Fardad, “BCD computing structures in quantum-dot cellular automata,” in Proc. IEEE Int. Conf. Comput. Commun. Eng., Kuala Lampur, Malaysia, May 2008, pp. 1042–1045.

[30] F. Kharbash and G. M. Chaudhry, “The design of quantum-dot cellular automata decimal adder,” in Proc. IEEE Int. Multitopic Conf., Karachi, Pakistan, Dec. 2008, pp. 71–75.

[31] 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.

[32] K. Walus and G. A. Jullien, “Design tools for an emerging SoC technology: Quantum-dot cellular automata,” Proc. IEEE, vol. 94, no. 6, pp. 1225–1244, Jun. 2006.

[33] V. Vankamamidi, M. Ottavi, and F. Lombardi, “Two-dimensional schemes for clocking/timing of QCA circuits,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 27, no. 1, pp. 34–44, Jan. 2.