This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2018.2878717, IEEE Transactions on Circuits and Systems II: Express Briefs

New Majority Gate Based Parallel BCD Adder Designs for Quantum-dot Cellular Automata

Tingting Zhang, Vikramkumar Pudi and Weiqiang Liu, *Senior Member, IEEE*

***Abstract*—In this paper, we first theoretically re-defined output decimal carry in terms of majority gates and proposed a carry lookahead structure for calculating all the intermediate output carries. We have used this method for designing the multi-digit decimal adders. Theoretically, our best n-digit decimal adder design reduces the delay and area-delay product (ADP) by 50% compared with previous designs. We have implemented our designs using QCADesigner tool. The proposed QCADesigner based 8-digit *PBA-BCD* adder achieves over 38% less delay compared with the best existing designs.**

***Index Terms*—Majority gate, parallel BCD adder, carry lookahead, quantum-dot cellular automata**

I. Introduction

The decimal arithmetic has received wide attention in response to the increasing demand for precision in financial and commercial based applications [1]. Several digital processors and computers were designed including decimal arithmetic hardware units [2], [3].

The current CMOS technology is approaching its scaling limitation. New nanotechnologies including quantum-dot cellular automata (QCA) [4], nanomagnetic Logic (NML) [5], and spin-wave devices (SWD) [6] are studied due to their advantages in terms of low power and high density. These emerging nanotechnologies are based on majority logic, which is different from conventional Boolean logic in CMOS.

As the core of decimal arithmetic, previous works have been conducted into majority-based parallel decimal adders [7]–[12]. The existing majority logic based parallel decimal adders mostly share the same structure, but differ from each other in the usage of binary adders. The 1-digit ripple carry adder (RCA) based BCD adders are proposed in [7], [8]. However, these designs can be further optimized to reduce hardware complexity. Carry flow adder (CFA) based and carry lookahead adder (CLA) based BCD adders are presented in [9], which show good performance. Moreover, [10] exploits novel binary adder to propose the efficient 1-digit BCD adder, reducing comprehensive consumption. In order to fully utilize the majority gates, [11] and [12] rewrite the correction function for less majority gates. Different from the existing designs, we use a new approach to compute carry logic in the multi-digit BCD adder.

In this paper, we propose a new definition for BCD adder output carry computation in terms of majority gates and use it for computing all the carries of the multi-digit BCD adder in parallel. We have introduced *decimal group generate* and *decimal group propagate* signals to calculate carries in the BCD adder. As a result, we have reduced delay in the multi-digit BCD adder. We have used different types of binary adders, such as RCA, CFA and parallel binary adder (PBA) for realizing the proposed multi-digit BCD adder. Theoretically, our PBA based n-digit BCD adder reduces the delay and area-delay product (ADP) by 50% compared with the existing designs.

We have implemented our designs using QCA technology and designed using QCADesigner [4]. The proposed QCADesigner based 8-digit *PBA-BCD* adder achieves at least 38% less delay compared with the best existing designs in [10]–[12].

The rest of the paper is organized as follow. The conventional structure of decimal adders are presented in Section II. The proposed structure of decimal adders, especially the circuit for calculating decimal carry is provided in Section III. Section IV presents the complexity analysis, area complexity and delay complexity included. Experimental results using QCA and comprehensive comparison with previous relevant works are provided in Section V. Finally, Section VI concludes the paper.

II. Background

Fig. 1 shows the block diagram of conventional 1-digit BCD adder. The 1-digit BCD adder consists of 4-bit binary adder (*ADD1*), correction logic (*CL*) and 4-bit binary adder (*ADD2*). The binary adder (*ADD1*) adds the decimal number $dA_{3:0}$, $dB_{3:0}$ and dC_{in} to produce the binary sum $bS_{3:0}$ and output carry bC_{out}. The *CL* circuit produces the $cL_{3:0}$ and decimal output carry signals dC_{out} for converting binary sum $bS_{3:0}$ to decimal sum $dS_{3:0}$. The $cL_{3:0} = (0110)_2$, if $dC_{out} = 1$ otherwise $cL_{3:0} = (0000)_2$. The binary adder (*ADD2*) produces decimal digit $dS_{3:0}$ by adding $bS_{3:0}$ and $cL_{3:0}$.

The theoretical delay required for generating 1-digit BCD adder dC_{out} signal ($d_c(1)$) and $dS_{3:0}$ signal ($d(1)$) are given in (1) and (2).

$$d_c(1) = d_{a1} + d_{cl} \quad (1)$$

$$d(1) = d_{a1} + d_{cl} + d_{a2} \quad (2)$$

where d_{a1}, d_{cl} and d_{a2} represent the delays required for single *ADD1*, *CL* and *ADD2* blocks, respectively.

This work is supported by grants from National Science Foundation China (No. 61871216 and No. 61401197) and Six Talent Peaks Project in Jiangsu Province (XYDXX-009).

T. Zhang and W. Liu are with College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China (emails: {ztt0416, liuweiqiang}@nuaa.edu.cn)

V. Pudi is with the Hardware and Embedded Systems Lab, School of Computer Science and Engineering, Nanyang Technological University, 639798, Singapore (email: pudi@ntu.edu.sg)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2018.2878717, IEEE Transactions on Circuits and Systems II: Express Briefs

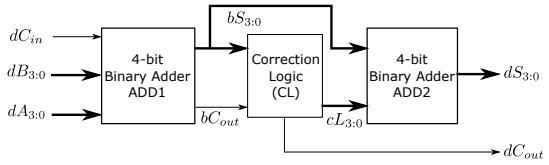

Fig. 1: Block diagram of 1-digit BCD adder.

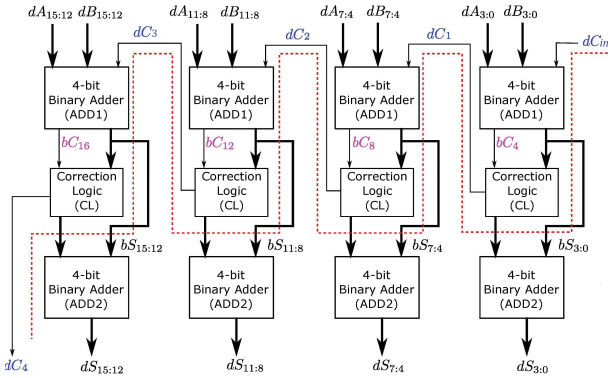

Fig. 2: Block diagram of 4-digit BCD adder.

Similarly, the theoretical delay required for generating n-digit BCD adder output carry and decimal sum signals are given in (3) and (4). From Fig. 2, we can observe that the delay required for the calculation of output carry signal ($d_c(n)$) and decimal sum signal ($d(n)$). The delay path in Fig. 2 is marked using the dotted line.

$$d_c(n) = n(d_{a1} + d_{cl}) \quad (3)$$

$$d(n) = n(d_{a1} + d_{cl}) + d_{a2} \quad (4)$$

The delay $d_c(n)$ and $d(n)$ are in multiples of n. This is due to the computation of output carry in the form of ripple carry style.

The theoretical definition for calculating the ripple carry style output carry of the single digit output carry is given as follow:

$$dC_{out} = bC_{out} + (bS_{3:0} >= 10) \quad (5)$$

The recent proposed BCD design in [10] uses the output carry shown in (5). The multi-digit BCD adder design in [10] achieved low delay due to the parallel nature of 4-bit binary adder (*ADD1*). In this paper, we propose a new definition for the output carry in (5), which is employed into parallel implementation of the multi-digit BCD adders.

III. Proposed BCD Adder Designs

The block diagram of parallel 1-digit BCD circuit is shown in Fig. 3. The design in [12] used the same block diagram for the implementation of BCD adder but they have used AND-OR gate based output carry as shown in (6).

$$dC_{out} = bC_{out} + (bS_{3:0} >= 10) + (bS_{3:0} == 9)dC_{in} \quad (6)$$

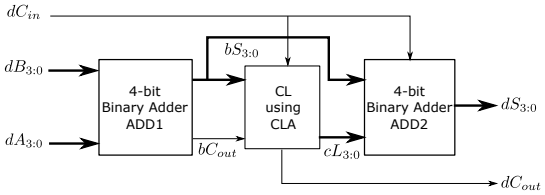

Fig. 3: Proposed block diagram of 1-digit BCD adder.

We are going to define the dC_{out} in terms of majority gates. For this, we rewrite (6) as follows:

$$\begin{aligned} dC_{out} &= bC_{out} + (bS_{3:0} >= 10) + (bS_{3:0} >= 9)dC_{in} \\ &= bC_{out} + (bS_{3:0} >= 10) \\ &\quad + [bC_{out} + (bS_{3:0} >= 9)]dC_{in} \end{aligned} \quad (7)$$

The logic signals $bC_{out} + (bS_{3:0} >= 10)$ and $bC_{out} + (bS_{3:0} >= 9)$ can be rewritten as $[bC_{out} + (bS_{3:0} >= 10)] \cdot [bC_{out} + (bS_{3:0} >= 9)]$ and $[bC_{out} + (bS_{3:0} >= 10)] + [bC_{out} + (bS_{3:0} >= 9)]$, respectively. By substituting these values in dC_{out}, we can rewrite the equation of dC_{out} as follows:

$$\begin{aligned} &dC_{out} \\ &= [bC_{out} + (bS_{3:0} >= 10)] \cdot [bC_{out} + (bS_{3:0} >= 9)] + \\ &[bC_{out} + (bS_{3:0} >= 10) + bC_{out} + (bS_{3:0} >= 9)]dC_{in} \end{aligned} \quad (8)$$

The dC_{out} in (8) is clearly in 3-input majority gate form with inputs $bC_{out} + (bS_{3:0} >= 10)$, $bC_{out} + (bS_{3:0} >= 9)$ and dC_{in}. We can write the dC_{out} using the majority gate as shown in (9).

$$\begin{aligned} dC_{out} = M(&bC_{out} + (bS_{3:0} >= 10), \\ &bC_{out} + (bS_{3:0} >= 9),\ dC_{in}) \end{aligned} \quad (9)$$

The terms $(bS_{3:0} >= 10)$ and $(bS_{3:0} >= 9)$ are binary signals and we are calling these signals as *decimal group generate* and *decimal group propagate* signals. These two signals are represented as $dG_{3:0}$ and $dP_{3:0}$, as shown in (10) and (11), respectively.

$$dG_{3:0} = bC_{out} + (bS_{3:0} >= 10) \quad (10)$$

$$dP_{3:0} = bC_{out} + (bS_{3:0} >= 9) \quad (11)$$

The proposed majority gate form of dC_{out} using $dG_{3:0}$ and $dP_{3:0}$ signals is given as follows:.

$$dC_{out} = M(dG_{3:0}, dP_{3:0}, dC_{in}) \quad (12)$$

The dC_{out} in (12) uses *decimal group generate* and *decimal group propagate* signals for calculation. This is similar to CLA method for the calculation of carry. Because of this, we are calling *CL* stage as *CL-CLA*. The $cL_{3:0}$ signal is calculated using the dC_{out} as shown in (13).

$$cL_{3:0} = \{0, dC_{out}, dC_{out}, 0\} \quad (13)$$

The proposed dC_{out} in (12) requires only 1 majority gate after calculating the $dG_{3:0}$ and $dP_{3:0}$ signals. Fig. 4 shows the majority gate diagram of proposed dC_{out} in (12). We have

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2018.2878717, IEEE Transactions on Circuits and Systems II: Express Briefs

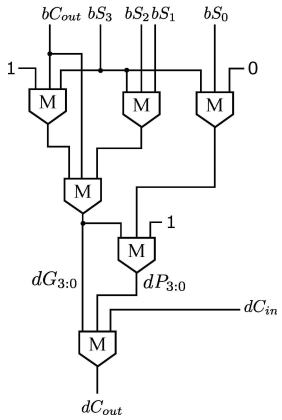

Fig. 4: Proposed majority gate circuit for calculating dC_{out}.

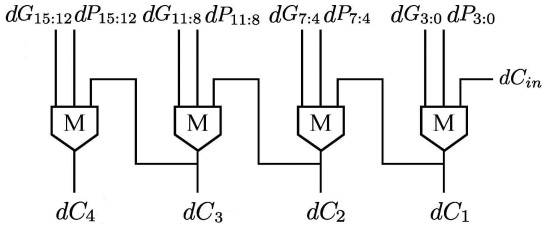

Fig. 5: Proposed majority gate circuit for calculating dC_1, dC_2, dC_3 and dC_4.

used the majority gate results presented in [10] for calculation of $dG_{3:0}$, as shown in (14).

$$\begin{aligned} dG_{3:0} &= bC_{out} + bS_3 \cdot bS_2 + bS_3 \cdot bS_1 \\ &= M(bC_{out}, M(bC_{out}, bS_3, 1), M(bS_3, bS_2, bS_1)) \end{aligned} \quad (14)$$

To save the area, we have calculated $dP_{3:0}$ as follows:

$$\begin{aligned} dP_{3:0} &= bC_{out} + (bS_{3:0} >= 9) \\ &= bC_{out} + (bS_{3:0} >= 10) + (bS_{3:0} == 9) \\ &= dG_{3:0} + bS_3 \cdot bS_0 \end{aligned} \quad (15)$$

We can observe that the *decimal group generate* and *decimal group propagate* signals are independent of decimal input carry, which are produced parallelly in the multi-digit BCD adder. Consequently, all *decimal group generate* and *decimal group propagate* signals of the multi-digit BCD adder share the same delay. Fig. 5 shows the majority gate circuit for calculating the carries dC_1, dC_2, dC_3 and dC_4 using *decimal group generate* and *decimal group propagate* signals. The delay required for calculating the dC_4 in Fig. 5 is only the delay of four majority gates, which can be achieved from the proposed definition of dC_{out} in (12).

The Fig. 6 shows the proposed block diagram of parallel 4-digit BCD adder.

IV. Complexity Analysis

The BCD adder uses the 4-bit binary adder for generation of decimal digits. The performance of BCD adder also depends upon the selection of 4-bit binary adder. In this section, we are going to derive the generalized expression for area and delay complexity (in terms of the majority gate) of n-digit

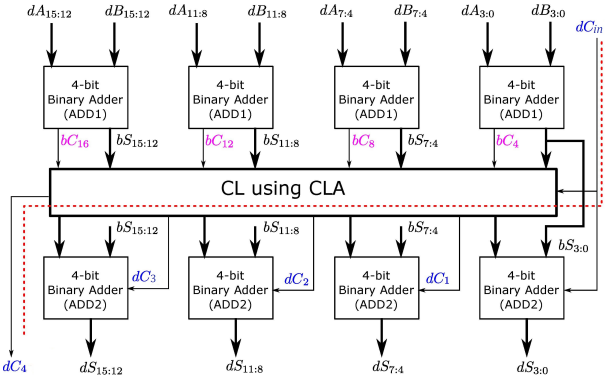

Fig. 6: Proposed block diagram of multi-digit BCD adder.

BCD adder. To verify the vadility, three different types of 4-bit binary adder are employed into proposed designs, which are RCA [7], CFA [9] and parallel binary adder (PBA) [10]. We are calling these designs as *RCA-BCD, CFA-BCD* and *PBA-BCD*, respectively.

A. Area Complexity

The $I(n)$ represents the total majority gates required for the proposed BCD adder. $I(n)$ is the sum of $I_{a1}(n)$, $I_{a2}(n)$ and $I_{cl}(n)$, where $I_{a1}(n)$, $I_{a2}(n)$ and $I_{cl}(n)$ represent total majority gates required for all *ADD1*, *ADD2* and *CL-CLA* blocks in n-digit BCD adder, respectively. The expressions for $I_{a1}(n)$ and $I_{a2}(n)$ are given as follow:

$$I_{a1}(n) = nI_{a1}(1) \quad (16)$$

$$I_{a2}(n) = nI_{a2}(1) \quad (17)$$

The $I_{a1}(1)$ and $I_{a2}(1)$ values depend upon the selection of binary adder for *ADD1* and *ADD2* blocks. In case of *PBA-BCD* design, both $I_{a1}(1)$ and $I_{a2}(1)$ require 14 majority gates.

The proposed *CL-CLA* block first calculates all the $dG_{i+3:i}$s and $dP_{i+3:i}$s. The calculation of each $dG_{i+3:i}$ and $dP_{i+3:i}$ requires 5 majority gates, as shown in Fig. 4. Overall calculation of all $dG_{i+3:i}$ and $dP_{i+3:i}$ requires $5n$ majority gates for n-digit BCD adder. After calculation of $dG_{i+3:i}$s and $dP_{i+3:i}$s, calculation of dC_is requires n majority gates. The total majority gates required for the calculation $I_{cl}(n)$ are given as follow:

$$I_{cl}(n) = 5n + n = 6n \quad (18)$$

The generalized expression for calculating the area complexity of an n-digit BCD adder (in terms of majority gates) is given as follow:

$$I(n) = n(I_{a1}(1) + I_{a2}(1)) + 6n \quad (19)$$

The area complexity for the n-digit *RCA-BCD, CFA-BCD* and *PBA-BCD* designs using (19) are given in (20)-(22), respectively.

$$I(n) = 30n \quad (20)$$

$$I(n) = 30n \quad (21)$$

$$I(n) = 34n \quad (22)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2018.2878717, IEEE Transactions on Circuits and Systems II: Express Briefs

TABLE I: Theoretical Area, Delay and ADP Comparisons for Different Types of n-digit BCD Adders

Type	Area				Delay				ADP
	$I_{a1}(n)$	$I_{a2}(n)$	$I_{cl}(n)$	$I(n)$	d_{a1}	d_{a2}	$d_{cl}(n)$	$d(n)$	$I(n) \times d(n)$
Prop. RCA-BCD	$12n$	$12n$	$6n$	$30n$	7	7	$3+n$	$17+n$	$30n^2+510n$
Prop. CFA-BCD	$12n$	$12n$	$6n$	$30n$	7	7	$3+n$	$17+n$	$30n^2+510n$
Prop. PBA-BCD	$14n$	$14n$	$6n$	$34n$	5	5	$3+n$	$13+n$	$34n^2+442n$
PBA-BCD [10]	$16n$	$10n$	$3n$	$29n$	5	4	$7n-5$	$7n+4$	$203n^2+116n$
BCD [11]	$12n$	$10n$	$3n$	$25n$	5	2	$7n-5$	$7n+2$	$175n^2+50n$
CFA-BCD [12]	$16n$	$12n$	$6n$	$34n$	5	4	$2+2n$	$2n+11$	$68n^2+374n$

B. Delay Complexity

The total delay required for the n-digit BCD adder is the sum of delay required for *ADD1* (d_{a1}), *ADD2* (d_{a2}) and *CL-CLA* ($d_{cl}(n)$) circuits as shown in Fig. 6. All *ADD1* blocks in n-digit BCD adder can calculate in parallel. The d_{a1}, d_{a2} and $d_{cl}(n)$ represent the delay of 1-digit *ADD1*, *ADD2* and n-digit CL-CLA blocks, respectively. The delay d_{a1} and d_{a2} depend upon the selection of 4-bit binary adder. In case of proposed *PBA-BCD* design, both of the d_{a1} and d_{a2} values are 5 majority gates.

The delay $d_{cl}(n)$ of n-digit BCD adder is the sum of delay required for calculation of $dG_{i+3:i}$, $dP_{i+3:i}$ and all dC_{out}s, as shown in Fig. 6. The delay required for $dG_{i+3:i}$ and $dP_{i+3:i}$ terms is 3 majority gates, as shown in Fig. 4. An n-digit BCD adder requires delay of n majority gates for calculation of all dC_{out}s after calculation of $dG_{i+3:i}$s and $dP_{i+3:i}$s, as shown in Fig. 4. The delay term $d_{cl}(n)$ is given as follow:

$$d_{cl}(n) = 3 + n \tag{23}$$

The generalized expression for calculating the delay complexity of an n-digit BCD adder (in terms of majority gates) is given as follow:

$$d(n) = d_{a1} + d_{a2} + 3 + n \tag{24}$$

The delay complexity for an n-digit *RCA-BCD, CFA-BCD* and *PBA-BCD* designs using (24) are given in (25)-(27), respectively.

$$d(n) = 17 + n \tag{25}$$
$$d(n) = 17 + n \tag{26}$$
$$d(n) = 13 + n \tag{27}$$

V. Experimental Results and Comparisons

Table I presents the theoretical comparisons of area, delay and ADP of BCD adder circuits in terms of majority gates for proposed designs and designs in [10]–[12]. From Table I, we can observe that the proposed designs require less delay compared with the best existing designs. Theoretically, our n-digit *PBA-BCD* adder design requires less than 80% in delay and ADP compared with the design in [10], about 50% in delay and ADP compared with the design in [12].

Multi-layer design method is adopted in order to be consistent with designs in [10]–[12], for the sake of fair and valid comparisons and evaluations. Fig. 7 and Fig. 8 show the layout diagrams of proposed 1-digit and 4-digit BCD adder using QCADesigner tool (version 2.0.3). Besides, in order to verify that our proposed BCD adder can be efficiently implemented

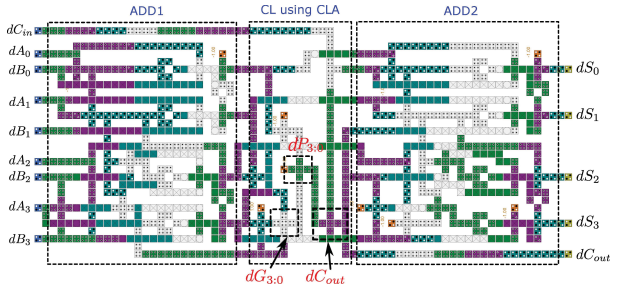

Fig. 7: Proposed layout of 1-digit PBA based BCD adder.

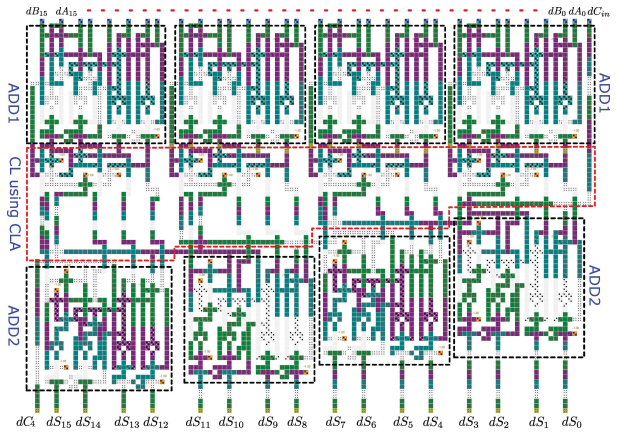

Fig. 8: Proposed layout of 4-digit PBA based BCD adder.

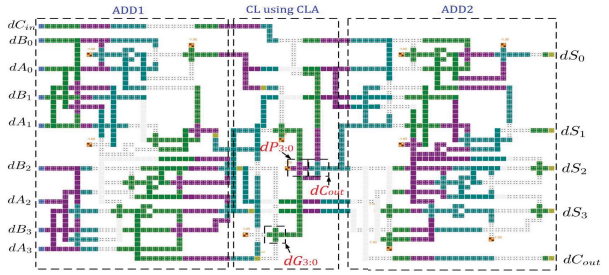

Fig. 9: Proposed layout of 1-digit PBA-BCD adder based on the 2DDWAVE clocking scheme (Area: 1.56μm^2, Delay: 4.25 clocks).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2018.2878717, IEEE Transactions on Circuits and Systems II: Express Briefs

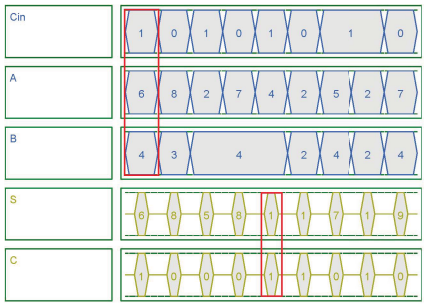

Fig. 10: Simulation result of proposed 1-digit PBA-BCD adder based on the 2DDWAVE clocking scheme.

TABLE II: Comparisons of proposed and other designs obtained from QCADesigner

Approach	Area (μm^2)	Delay (Clocks)	ADP ($\mu m^2 \times$ Clocks)	$Cost_{QCA}$ (10^5)
Prop. RCA-BCD1	1.01	4	4.04	3.47
Prop. RCA-BCD4	5.68	7	39.76	168.71
Prop. RCA-BCD8	16.09	11	176.99	1664.03
Prop. CFA-BCD1	1.52	$3\frac{3}{4}$	5.70	3.05
Prop. CFA-BCD4	8.14	$6\frac{3}{4}$	54.94	156.88
Prop. CFA-BCD8	21.65	$10\frac{3}{4}$	232.73	1589.25
Prop. PBA-BCD1	0.88	$3\frac{3}{4}$	3.30	4.53
Prop. PBA-BCD4	3.82	$5\frac{1}{4}$	20.05	135.15
Prop. PBA-BCD8	11.71	$7\frac{1}{4}$	84.89	1120.30
PBA-BCD1 [10]	0.89	3	3.12	3.08
PBA-BCD4 [10]	3.96	$8\frac{1}{4}$	32.67	288.93
PBA-BCD8 [10]	12.32	$15\frac{1}{4}$	187.75	4091.51
BCD1 [11]	0.76	3	2.28	1.43
BCD4 [11]	3.68	$8\frac{1}{4}$	30.36	172.32
BCD8 [11]	11.44	$15\frac{1}{4}$	174.46	2351.67
CFA-BCD1 [12]	1.37	$3\frac{1}{2}$	4.80	3.71
CFA-BCD4 [12]	6.68	$6\frac{3}{4}$	45.09	293.81
CFA-BCD8 [12]	14.95	$11\frac{3}{4}$	175.66	3084.47

with different clocking schemes, PBA based BCD adder is designed by applying the 2DDWAVE mechanism [13] (Fig. 9). The simulation result is shown in Fig. 10.

The following parameters have been used for coherence vector simulation engine: Number of Samples: 128000; Convergence Tolerance: 0.00001; Radius of Effect: 55 nm. The rest of parameters are set as the default values.

Table II presents comparisons of area, delay, ADP and QCA cost function $Cost_{QCA}$ (defined in [14], where k is 2, l is 2, p is 2) of proposed designs with designs in [10]–[12]. The power consumption of the designs can be further analyzed by the method proposed in [15]. The proposed *PBA-BCD* adders require only 0.88 μm^2 and 11.71 μm^2 area for 1-digit and 8-digit BCD adders, respectively; the delay required are 3.75 and 7.25 clocks, respectively, as shown in Table II. The proposed 8-digit *PBA-BCD* adder achieves at least 38% less delay, 51% less ADP and 52% less $Cost_{QCA}$ compared with designs in [10]–[12]. For the 1-digit case, the 2DDWAVE clocking scheme introduces slightly more clocking delay and also occupies slightly larger area. However, the results are still consistent with that using the conventional clocking scheme. Obviously, due to the proposed new formulations as presented in the previous sections, when the scale of the design increases, our proposed approach shows excellent performance in terms of delay. Consequently, overall designs have achieved substantial savings.

VI. Conclusions

In this paper, we have developed a general methodology to obtain low-delay for multi-digit BCD adders in QCA. The methodology has been applied to the RCA, CFA, PBA based BCD adders to obtain the low-delay. Theoretically, our n-digit *PBA-BCD* adder design requires 50% less delay and ADP compared with the design in [10]–[12]. We have validated our designs using the QCADesigner tool. From QCADesigner layout results, 8-digit *PBA-BCD* adder requires at least 38% less delay compared with the existing best designs.

References

[1] L. K. Wang, M. J. Schulte, J. D. Thompson, and N. Jairam, "Hardware designs for decimal floating-point addition and related operations," *IEEE Transactions on Computers*, vol. 58, no. 3, pp. 322-335, 2009.

[2] S. Shankland, "IBMs POWER6 gets help with math, multimedia," in *ZDNet News*, 2006.

[3] X. Cui, W. Dong, W. Liu, E. E. Swartzlander Jr, and F. Lombardi, "High performance parallel decimal multipliers using hybrid BCD codes," *IEEE Transactions on Computers*, vol. 66, no. 12, pp. 1994-2004, 2017.

[4] K. Walus and G. A. Jullien, "Design tools for an emerging SoC technology: quantum-dot cellular automata," in *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1225-1244, 2006.

[5] M. Vacca, M. Graziano, J. Wang, F. Cairo, and G. Causapruno, *NanoMagnet Logic: An Architectural Level Overview.* Springer Berlin Heidelberg, 2014.

[6] A. Khitun and K. L. Wang, "Nano scale computational architectures with spin wave bus," *Superlattices & Microstructures*, vol. 38, no. 3, pp. 184-200, 2005.

[7] M. Taghizadeh, M. Askari, and K. Fardad, "BCD computing structures in quantum-dot cellular automata," in *Proc. International Conference on Computer and Communication Engineering*, 2008, pp. 1042-1045.

[8] F. Kharbash and G. M. Chaudhry, "The design of quantum-dot cellular automata decimal adder," in *Proc. IEEE International Conference on Multitopic*, 2008, pp. 71-75.

[9] W. Liu, L. Lu, M. O'Neil, and E. E. Swartzlander Jr, "Cost-efficient decimal adder design in quantum-dot cellular automata," in *Proc. IEEE International Symposium on Circuits and Systems*, 2011, pp. 1347-1350.

[10] G. Cocorullo, P. Corsonello, F. Frustaci, and S. Perri, "Design of efficient BCD adders in quantum-dot cellular automata," *IEEE Transactions on Circuits & Systems II Express Briefs*, vol. 64, no. 5, pp. 575-579, 2017.

[11] D. Abedi and G. Jaberipur, "Decimal full adders specially designed for quantum-dot cellular automata," *IEEE Transactions on Circuits & Systems II Express Briefs*, vol. 65, no. 1, pp. 106-110, 2017.

[12] D. Ajitha, K. Ramanaiah, and V. Sumalatha, "An enhanced high-speed multi-digit BCD adder using quantum-dot cellular automata," *Journal of Semiconductors*, vol. 38, no. 2, pp. 38-46, 2017.

[13] A. Roohi, R. Zand, S. Angizi, and R. F. Demara, "A parity-preserving reversible QCA gate with self-checking cascadable resiliency," *IEEE Transactions on Emerging Topics in Computing*, 2018, DOI: 10.1109/TETC.2016.2593634.

[14] W. Liu, L. Lu, M. O'Neill, and E. E. Swartzlander, "A first step toward cost functions for quantum-dot cellular automata designs," *IEEE Transactions on Nanotechnology*, vol. 13, no. 3, pp. 476-487, 2014.

[15] F. S. Torres, R. Wille, P. Niemann, and R. Drechsler, "An energy-aware model for the logic synthesis of quantum-dot cellular automata," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018, DOI: 10.1109/TCAD.2018.2789782.