IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 03, March 2020

High Speed Multi-Digit Decimal Adder Using CLA Techniques

V.Sushmitha[1], Mareeswari.L[2], Mareshwari.S[3], Selvarani.N[4], Susee.P[5]

Assistant Professor, Department of Electronics and Communication, Chandy College of Engineering, Tuticorin, India[1]

Student, Department of Electronics and Communication, Chandy College of Engineering, Tuticorin, India[2,3,4,5]

Abstract: In this paper, we first theoretically re-defined output decimal carry in terms of majority gates and proposed a carry look ahead 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 QCA Designer tool. The proposed QCA Designer based 8-digit PBA-BCD adder achieves over 38% less delay compared with the best existing designs.

Keywords: Majority gate, parallel BCD adder, carry look ahead, 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. Several digital processors and computers were designed including decimal arithmetic hardware units. The current CMOS technology is approaching its scaling limitation. New nanotechnologies including Quantum-dot Cellular Automata (QCA), Nano Magnetic Logic (NML), and Spin-Wave Devices (SWD) 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. 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. However, these designs can be further optimized to reduce hardware complexity. Carry flow adder (CFA) based and carry Look Ahead Adder (CLA) based BCD adders are presented, which show good performance. Moreover, exploits novel binary adder to propose the efficient 1-digit BCD adder, reducing comprehensive consumption. In order to fully utilize the majority gates, 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 QCA Designer. The proposed QCA De- signer based 8-digit PBA-BCD adder achieves at least 38% less delay compared with the best existing designs.

II.EXISTING SYSTEM

Among the emerging technologies recently proposed as alternatives to the classic CMOS, Quantum-dot cellular automata (QCA) is one of the most promising solutions to design ultra -low-power and very high speed digital circuits. Efficient QCA-based implementations have been demonstrated for several binary and decimal arithmetic circuits, but significant improvements are still possible if the logic gates inherently available within the QCA technology are smartly exploited. This brief proposes a new approach to design QCA-based BCD adders. Exploiting innovative logic formulations and purpose designed QCA modules, computational speed significantly higher than existing counterparts are achieved without sacrificing either the occupied area or the cells count.A new design approach has been presented and demonstrated to achieve efficient QCA-based implementations of decimal adders. Unconventional logic formulations and purpose-designed logic modules here proposed allows outperforming decimal adders known in literature. In fact, the new 1-digit BCD adder exhibits computational delay and area occupancy and lower than existing competitors. These advantages become even more evident when two n-digit decimal numbers must be summed.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 03, March 2020

It is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a C_{in}, which is the C_{out} of the previous adder. This kind of adder is called a ripple-carry adder (RCA), since each carry bit "ripples" to the next full adder. Note that the first (and only the first) full adder may be replaced by a half adder (under the assumption that $C_{in} = 0$). The layout of a ripple-carry adder is simple, which allows fast design time; however, the ripple-carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder. The gate delay can easily be calculated by inspection of the full adder circuit. Each full adder requires three levels of logic. In a 32-bit ripple-carry adder, there are 32 full adders, so the critical path (worst case) delay is 3 (from input to carry in first adder) + 31 × 2 (for carry propagation in latter adders) = 65 gate delays. A design with alternating carry polarities and optimized AND-OR-Invert gates can be about twice as fast.

III.PROPOSED SYSTEM

The decimal arithmetic has received wide attention in response to the increasing demand for precision in financial and commercial based applications. Several digital processors and computers were designed including decimal arithmetic hardware units.The current CMOS technology is approaching its scaling limitation. New nanotechnologies including quantum-dot cellular automata (QCA), nanomagnetic Logic (NML), and spin-wave devices (SWD) 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. 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. However, these designs can be further optimized to reduce hardware complexity. Carry flow adder (CFA) based and carry look ahead adder (CLA) based BCD adders are presented, which show good performance. Moreover, exploits novel binary adder to propose the efficient 1-digit BCD adder, reducing comprehensive consumption. In order to fully utilize the majority gates, 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.

ADD1, CL and ADD2 blocks, respectively.

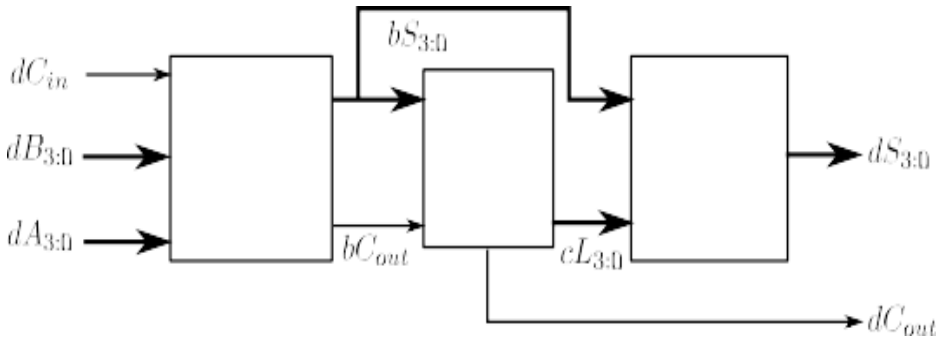

Fig. 1: Block diagram of 1-digit BCD adder.

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.

$d_c(n) = n(d_{a1} + d_{cl})$
$d(n) = n(d_{a1} + d_{cl}) + d_{a2}$

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

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 03, March 2020

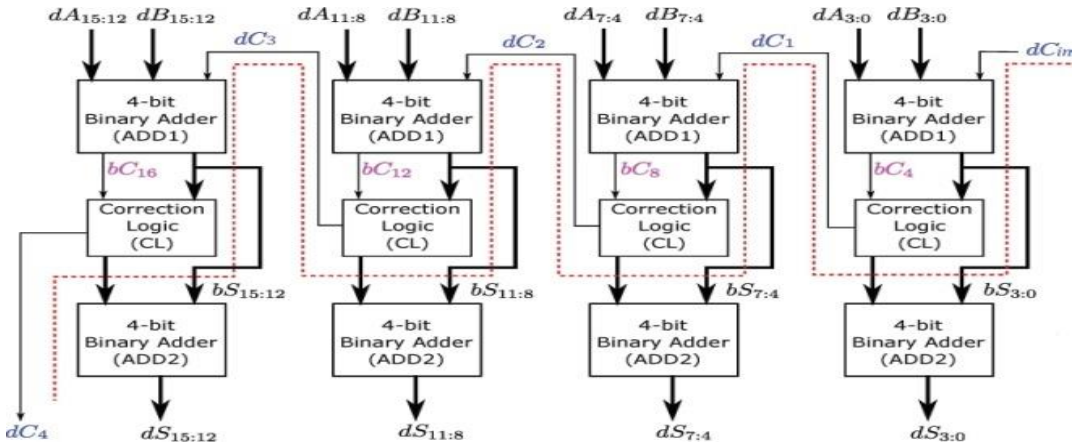

Fig. 2: Block diagram of 4-digit BCD adder.

IV.PROPOSED BCD ADDER DESIGNS

The block diagram of parallel 1-digit BCD circuit is shown in Fig. 1. 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).

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

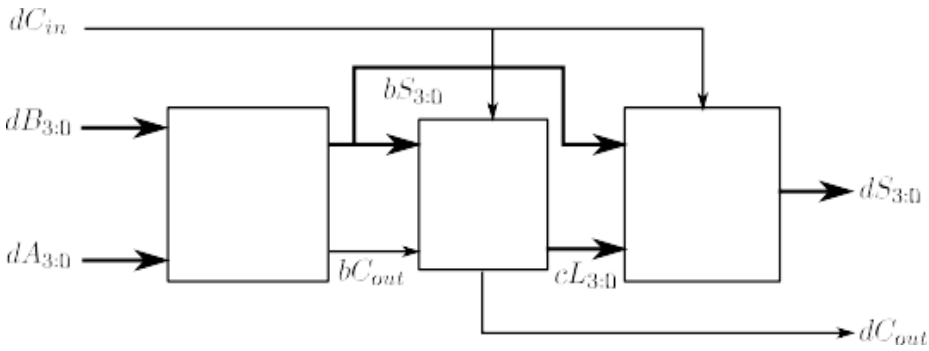

Fig. 3: Proposed block diagram of 1-digit BCD adder.

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

$$dC_{ot} = bC_{out} + (bS_{3:0} >= 10) + (bS_{3:0} >= 9)dC_{in}$$
$$= bC_{out} + (bS_{3:0} >= 10) + [bC_{out} + (bS_{3:0} >= 9)]dC_{ir} \quad (7)$$

The logic signals $bC_{out} + (bS_{3:0} >= 10)$ and $bC_{out} + (bS_{3:0} >= 9)$ can be rewritten as $[bC_{out} + (bS_{3:0} >= 10)]$ $[bC_{out} + (bS_{3:0} >= 9)]$ 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: .

$DC_{out} = [bC_{out} + (bS_{3:0} >= 10)] \cdot [bC_{out} + (bS_{3:0} >= 9)] + [bC_{out} + (bS_{3:0} >= 10) + bC_{out} + (bS_{3:0} >= 9)]dC_{in}$ (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).

$dC_{out} = M(bC_{out} + (bS_{3:0} >= 10), bC_{out} + (bS_{3:0} >= 9), dC_{in})$ (9)

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 03, March 2020

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.

$$dG_3 = bC_{out} + (bS_{3:0} >= 10) \quad (10)$$

$$dP_3: = bC_{out} + (bS_{3:0} >= 9) \quad (11)$$

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

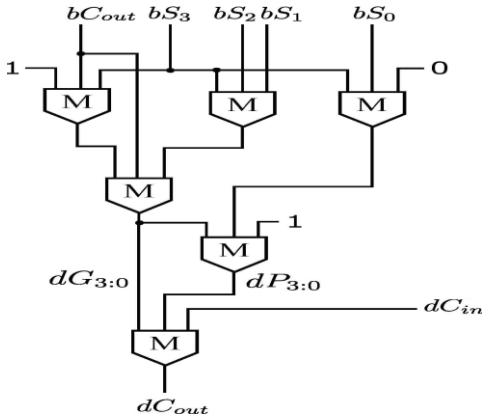

Fig. 4: Proposed majority gate circuit for calculating dC_{out}

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

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).

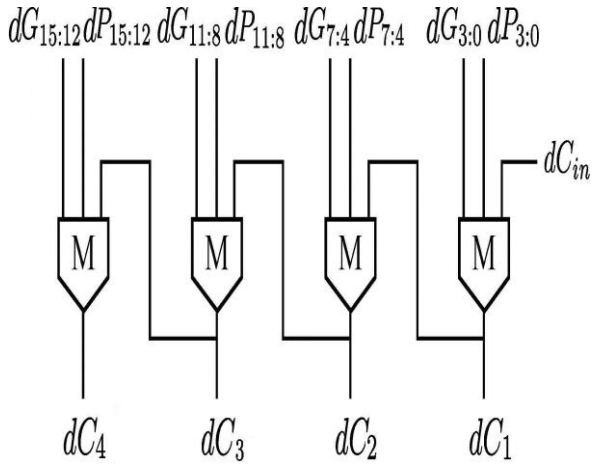

Fig. 5: Proposed majority gate circuit for calculating dC_1, dC_2, dC_3 and dC_4.

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

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).

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 03, March 2020

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

$dG_{3:0} = bC_{out} + bS_3 \cdot bS_2 + bS_3 \cdot bS_1$

$= M(bC_{out}, M(bC_{out}, bS_3, 1), M(bS_3, bS_2, bS_1))$ (14)

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

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

$= bC_{out} + (bS_{3:0} >= 10) + (bS_{3:0} == 9)$

$= dG_{3:0} + bS_3 \cdot bS_0$ (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}

V.RESULTS

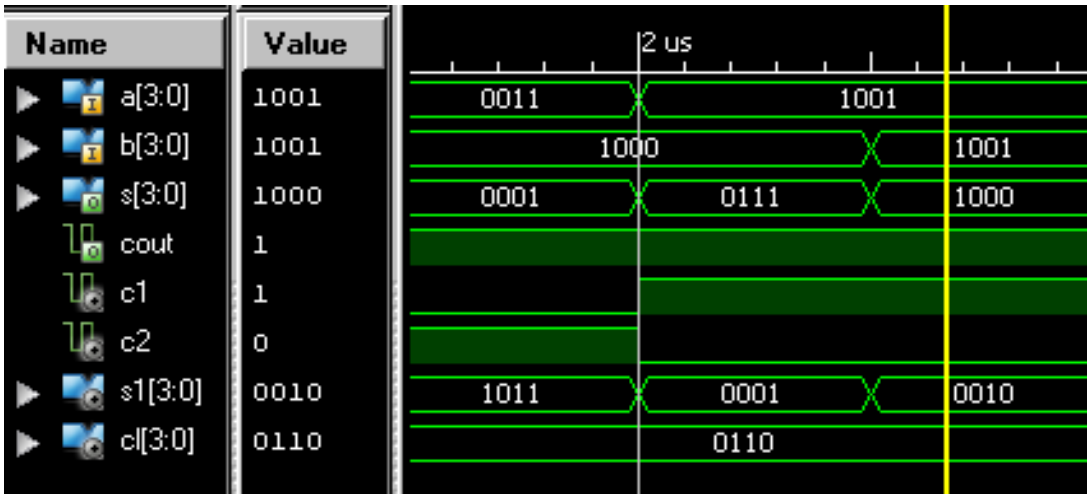

Fig.6.Proposed 4bit BCD Adder Simulation output

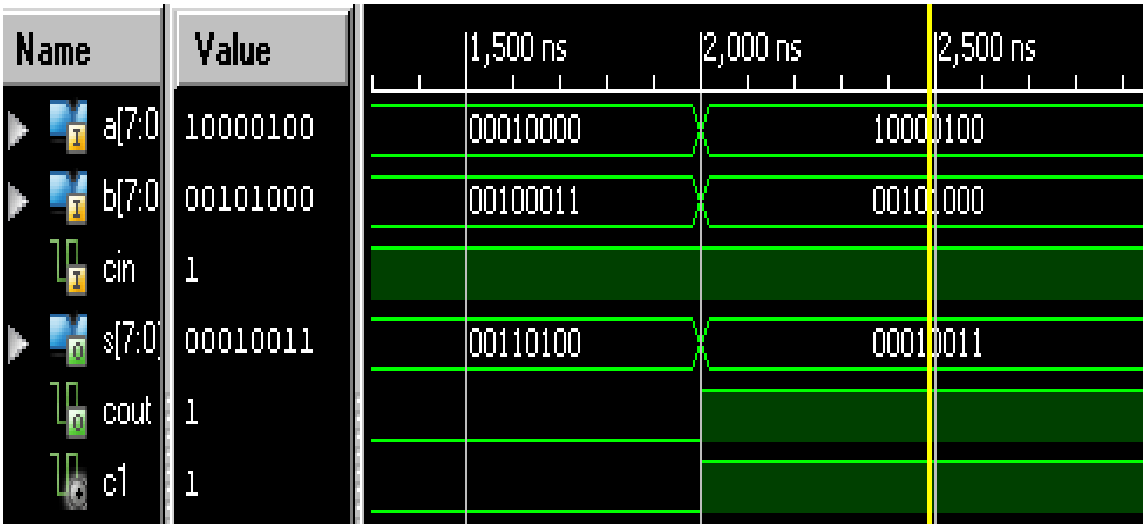

Fig.7.Proposed 8bit BCD Adder Simulation output

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 03, March 2020

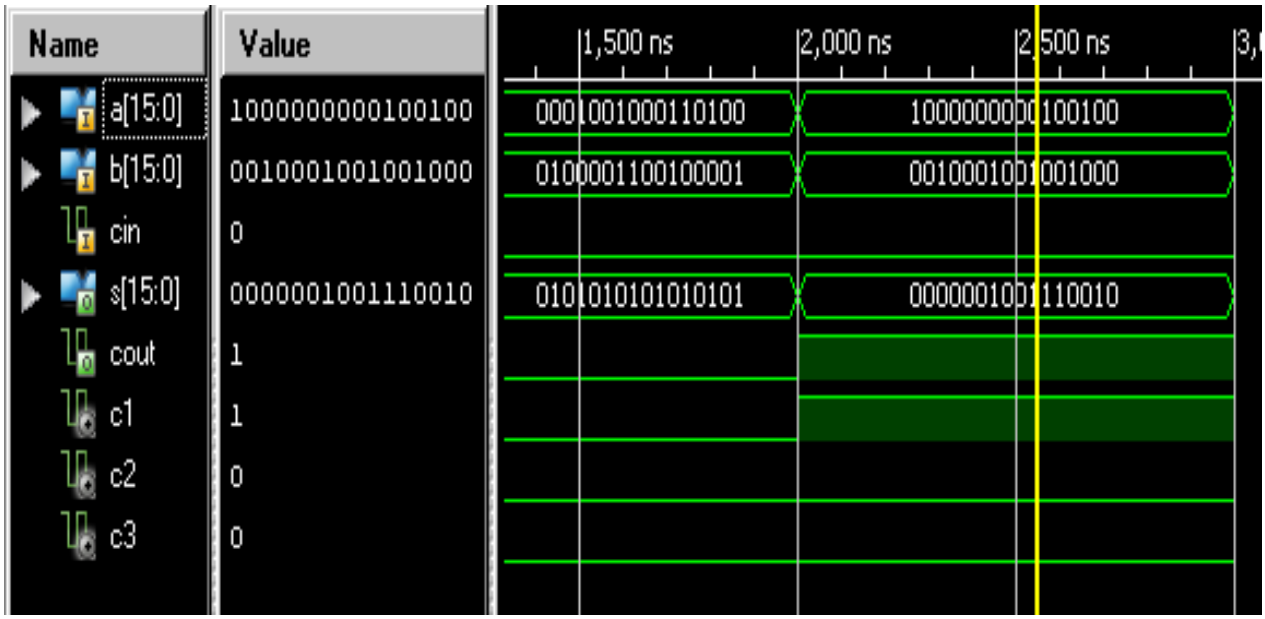

Fig.8.Proposed 16bit BCD Adder Simulation output

TABLE I: 4BIT BCD ADDER

	Existing	**Proposed**
Number of slices	6	6
Number of 4 input LUTs	*11*	*11*
Delay (ns)	**8.195**	**7.863**
Power (W)	0.168	0.168
Energy (J)	1.376	1.320

TABLE II: 8BIT BCD ADDER

	Existing	**Proposed**
Number of slices	11	13
Number of 4 input LUTs	20	22
Delay (ns)	**10.690**	**10.241**
Power (W)	0.169	0.169
Energy (nJ)	**1.806**	**1.73**

TABLE III: 16BIT BCD ADDER

	Existing	**Proposed**
Number of slices	22	25
Number of 4 input LUTs	40	44
Delay (ns)	**16.347**	**14.997**
Power (W)	0.172	0.172
Power delay product (energy) nJ	**2.811**	**2.579**

DOI 10.17148/IJIREEICE.2020.8319

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

**International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering**

Vol. 8, Issue 03, March 2020

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

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 QCA Designer tool. From QCA Designer 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 technol- ogy: 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, NanoMag- net Logic: An Architectural Level Overview. Springer Berlin Heidelberg, 2014.
[6]. 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 resilien- cy," 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.