CHAPTER 5

ENHANCED SINGLE-DIGIT BCD ADDERS

5.1 Introduction

The development of high performance, low power digital circuits are achieved by a suitable emerging Nano-device called QCA. Even though many efficient arithmetic circuits were designed using QCA, but still there is a challenging to implement high-speed circuits in an optimized manner. Among these circuits, one of the important structure is Binary-Coded Decimal (BCD)/Decimal Adder. The BCD addition can be used for the applications like financial, industrial, floating-point arithmetic and commercial applications. The development of BCD/Decimal addition hardware is more significant with its intensifying real-time applications.

The existing parallel BCD adders were implemented using conventional BCD structure and the RCA structures require a larger area and longer delay [84, 85]. Johnson–Mobius adder introduced a different coding technique to perform the decimal addition [86]. In Johnson–Mobius method, the parallel adder requires extra hardware, larger area and produces more delay. However, the Serial JMC Adder adder performs the decimal addition with less area, but it requires more clock cycles [86]. The full adder used in parallel BCD adder occupies a larger area and produces more delay [87]. The CFA-based BCD adder uses the optimized correction logic, but it has a longer delay due to the carry propagation in the final part [88]. The cost efficient CLA-based BCD adder uses generate and propagate functions along with multiplexers to perform the decimal addition with higher speed and utilizes more number of gates [88]. Whereas, all the existing Decimal Adders require either more number of cells or consume more delay [84-88], which makes circuit inefficient to do decimal addition faster.

The primary purpose of this research work in this chapter is to present the single-digit BCD adders to perform the BCD/Decimal addition at higher speed with less complexity. In this chapter, two single-digit BCD adder designs are presented. First, a new improved Single Digit BCD Adder (SDBA) is proposed for decimal addition in an optimized manner. The SDBA design is a combination of CFA and CLA-
based BCD adder. The SDBA uses the optimized conventional CFA Type-II binary adder for generating binary sum as the first stage. The correction logic to detect the generated binary sum validity as the second stage and 2:1 multiplexers to produce the final decimal sum bits as the third phase. The proposed SDBA is 42% faster than the CFA and occupies 46% lesser cell count than CLA-based BCD adder. Further, the overall cost (Area x Latency$^2$) of the proposed SDBA design is also reduced by 71% and 25% compared to CFA, CLA-based BCD adders respectively. Hence, the SDBA offers a trade-off between area and speed.

In the second design to perform the Decimal addition with high speed, a new correction logic formulation method is proposed for single-digit BCD adder. The proposed Enhanced Single-Digit BCD Adder (ESDBA) is 26% faster than CFA-based BCD adder with slight overhead in the area. Furthermore, the ESDBA occupies 34% lesser in the area with an increase in the delay by one clock cycle compared to CLA-based BCD adder for single-digit BCD addition, but while the ESDBA is extended for multi-digit operation, a significant improvement in speed is achieved among all the existing designs. The multi-digit operations are also performed using the proposed ESDBA, which is cascaded innovatively.

5.1.1 The Requisite for Decimal Arithmetic

Even though the binary arithmetic is well known in the processors, it has some limitations in its practice. To represent fractional numbers it requires too many bits, e.g., 0.3$_{10}$ = 0.01001…., whereas in BCD, the same number can be represented with finite bits, i.e., 0.3$_{10}$ = 0.0011$_{BCD}$. The binary representation is not appropriate for representing the precise decimal fractions, which impact the inappropriate results, consequently deteriorates the correctness of complete calculations. To overcome the above drawback a standard representation known as BCD numbers is employed [106]. In BCD, each decimal digit from 0 to 9 was represented by four bits, i.e., bits (0000)$_2$ to (1001)$_2$ respectively.

The decimal computation ruins indispensable for various applications like financial, commercial, and internet-based applications such as e-banking and e-commerce [106-108]. The decimal addition is used in many applications of
programming languages to develop the libraries in COBOL, Basic, Java and C# and high-end processor [109-113]. In recent times, decimal arithmetic encompassed in Floating point standard renowned as IEEE 754-2008 [114,115]. In recent times, a rehabilitated attention in providing area-efficient hardware implementation with a higher speed for performing the decimal arithmetic will considerably speed up an extensive range of applications [116-123]. Thus, an investigation into decimal arithmetic has multiplied momentum.

The remaining part of the chapter is organized as follows. In Section 5.2, the design methodology of SDBA block diagram and its layout is described in detail. Section 5.3 provides simulation results and discussions of the proposed SDBA design parameters with the existing designs. The ESDBA with new correction logic methodology, its importance, and the layout of ESDBA is explained in detail in Section 5.4. Section 5.5 explains the simulation results of ESDBA and its performance parameters comparison with existing designs. Section 5.6 concludes the proposed designs and its future work.

5.2 Design Methodology of SDBA

The proposed SDBA is the modified version of the CFA-based and CLA-based BCD adders to enhance the performance of decimal addition. The proposed SDBA uses the optimized conventional CFA Type-II binary adder for generating binary sum as the first stage. The final stage of the conventional structure is replaced with multiplexers, which improves the speed by decreasing the delay anticipated by carry propagation in CFA-based BCD adder, and reduces complexity as compared to the CLA-based design. As the final stage contains multiplexers, it is very much compatible with Field Programmable Gate Array (FPGA) structures and supports parallelism. Two cases are considered to produce the decimal sum bits in the final stage as described below.

**Case I: Binary Sum ≤ 9** - When adding any two BCD numbers (using 8421 code), if the binary sum \(Z_8Z_4Z_2Z_1\) is equal or less than 9 and carry out of MSB = 0, then both the binary and decimal sums are same and no correction logic is needed. The final decimal sum bits are given as
\[ \begin{align*}
S_8S_4S_2S_1 &= Z_8Z_4Z_2Z_1 \\
\end{align*} \] (5.1)

Where,

\[ Z_8Z_4Z_2Z_1 \] – Four-bit Binary sum,

\[ S_8S_4S_2S_1 \] – Decimal Sum.

**Case II: 9 < Binary Sum ≤ 19** - Either the sum is greater than 9 and ≤ 19 or sum ≤ 9 with a carry out of MSB will be 1, then both the binary and decimal sums are not same. The correction logic (C) is needed to convert the binary sum into valid decimal sum [88]. The correction logic is obtained by comparing the Binary sum and BCD sum and simplified using the five-variable K-map. The general correction logic is expressed as

\[ C = K + Z_8Z_4 + Z_8Z_2 \] (5.2)

Equation (5.2) can be simplified and written as

\[ C = K + Z_8(Z_4 + Z_2) \] (5.3)

Using majority gates, Equation (5.3) can be written as

\[ C = M(K, M(Z_8, M(Z_4, Z_2, 1), 1), 1) \] (5.4)

Where, K-Carry out from MSB of 4-bit Binary sum

If \( C = 1 \), then \((0110)_2 = (6)_{10}\) is added to the binary sum using the final addition circuit to represent the correct decimal sum [82]. The above two conditions are used in CFA-based BCD adder design approach to obtain the valid decimal sum [88]. However, by observing the function table of BCD adder [82], a new logic is developed in the proposed design to replace the final part of the conventional RCA structure with 2:1 multiplexers. The new logic is developed from case II as given in the following sentences. For all the cases if the binary sum ≤ 19, both \( S_1 \) and \( Z_1 \) will be same. Hence, no addition circuit or multiplexer is needed.
If $9 < \text{Binary Sum} \leq 19$ consequently $C=1$ then $S_2 = \overline{Z}_2$ else $S_2 = Z_2$. The third-bit sum $S_4 = \text{'}1\text{'}$ for the binary sum $(01110)_2 - (10001)_2$, whereas $S_4 = \text{'}0\text{'}$ for binary sums $(01010-01101,10010\text{ and }10011)_2$, except this binary sum values for all other conditions (binary sum) $S_4 = Z_4$. Similarly, the $S_8 = \text{'}1\text{'}$ for binary sum $(10010\text{ and }10011)_2$ whereas, $S_8 = \text{'}0\text{'}$ for binary sum $(01010-10001)_2$ except the values given, for all the other conditions $S_8 = Z_8$. The evaluation of logic expressions based on the analysis of case II is furnished as follows.

$S_1 = Z_1$ if $C = 0$ or $1$ \hspace{1cm} (5.5)

$S_2 = \overline{Z}_2$ if $C = 1$ else $S_2 = Z_2$ \hspace{1cm} (5.6)

$S_4 = \overline{K}Z_8Z_4Z_2 + K\overline{Z}_8\overline{Z}_4\overline{Z}_2$ \hspace{1cm} (5.7)

The expression (5.7) can be rewritten to optimize the circuit as follows

$S_4 = \overline{K}Z_8Z_4Z_2 + K\overline{Z}_8.(\overline{Z}_4 + Z_2)$ \hspace{0.5cm} If $C = 1$ else $S_4 = Z_4$ \hspace{1cm} (5.8)

$S_8 = KZ_2$ if $C = 1$ else $S_8 = Z_8$ \hspace{1cm} (5.9)

Equations (5.6, 5.8, and 5.9) regarding majority gates to implement the multiplexer logic is given as follows.

$S_2 = M(M(\overline{C},Z_2,0),M(C,\overline{Z}_2,0),1)$ \hspace{1cm} (5.10)

$S_4 = M(M(\overline{C},Z_4,0),M(C,M(M(\overline{K},(M(Z_8,M(Z_4,Z_2,0),0),0)),..M(K,M(\overline{Z}_8,(M(Z_4,Z_2,1),0),0),1)),1) \hspace{1cm} (5.11)$

$S_8 = M(M(\overline{C},Z_8,0),M(C,M(K,Z_2,0),0),1)$ \hspace{1cm} (5.12)

The final decimal sum bits in case II are given as

\[ \text{Decimal sum} = C S_8 S_4 S_2 S_1. \]
It is observed from the equations (5.5, 5.6, 5.8, and 5.9) the 2:1 multiplexers are well suitable using ‘C’ as selection input instead of RCA in the final part. The proposed SDBA design requires some additional circuitry in addition to the correction logic to generate the inputs to the multiplexers in the final part. The SDBA uses new logic circuitry in the final part of a decimal adder, which provides high speed, smaller cell count, and area. The schematic diagram of the proposed SDBA design is shown in Figure 5.1.

![Schematic diagram of the proposed SDBA design](image)

**Fig. 5.1** Schematic diagram of the proposed SDBA design

The proposed SDBA design layout is shown in Figure 5.2. The upper part of the layout design shown in Figure 5.2 is constructed by the optimized CFA Type-II binary
adder, and in addition to the correction logic equation (5.4), the final part is built using the equations (5.10, 5.11, and 5.12). In the existing CFA-based BCD adder layout, the upper part of the design requires one additional clock for the inverter to generate the binary sum and 7 clock zones in the lower part of the design. Four-bit binary adder, requires $2^{1/4}$ clock cycles (i.e 9 clock zones) [88]. These extra clock zones make the BCD addition could not improve the speed of the device. Therefore, in order to improve the speed of the design, a new logic in addition to the correction logic and 2:1 multiplexers are introduced. Moreover, in the upper part of the proposed design, the CFA Type-II 4-bit binary adder is optimized by eliminating the additional clock for the inverter [22]. Hence, $1^{1/2}$ clock cycles (i.e. 6 clock zones) are enough in the upper 4-bit binary adder instead of $2^{1/4}$ clock cycles. Furthermore, the correction logic needs 2 clock zones and the three 2:1 multiplexers in the final part of the proposed design require only three clock zones.

Fig. 5.2 Layout of the proposed SDBA design
5.3 Results and Discussions

The proposed SDBA layout design produces the count, area, the number of clock zones, clock cycles, and the overall cost is calculated from the obtained area and latency (clock cycles). From Figure 5.2, it is also noted that the number of clock zones required is 11. Out of 11 clock zones, the upper 4-bit CFA requires 6 clock zones and the lower part of the multiplexer requires 3 clock zones. Since all the 2:1 multiplexers works in parallel with C as a common select signal. While performing the upper 4-bit CFA operation, the correction logic is arranged, and it requires only 2 clock zones besides 6 clock zones. Hence, in the proposed SDBA design, clock zones are reduced and speed increases significantly. Not only that, even the existing CLA-based adder performs the decimal addition with 10 clock zones, the area occupied by the use of existing CLA technique is more. Moreover, the existing CLA-based BCD adder requires more number of gates and involves with the increased complexity it needs 1838 cells. However, it is reduced to 987 cells in the proposed SDBA design by implementing CFA Type-II 4-bit binary adder. Further, it is also stated that with the use of multiplexers in the final part of the BCD addition in the SDBA, even it is similar to CLA speed is observed competitive in the proposed SDBA design. As the multiplexer circuit requires the same area as that of RCA, while performing the required functionality in a parallel manner the area of the SDBA is competing to the CFA-based BCD adder with higher speed.

The proposed SDBA cell size does not change, but the SDBA method improves the performance of the BCD adder design in terms of speed (CFA-based BCD adder) and reduction in the area (CLA-based BCD adder) compared to the existing designs. The simulation results are shown in Figure 5.3. The simulation results clearly indicate that the correct output appears after 2.75 clock cycles, but before 2.75 clock cycles, it gives the garbage outputs. All the layouts in QCA are simulated with Bistable approximation or Coherence vector simulation method. In this work, the simulation results are obtained with the help of the Coherence vector simulation method. Default settings (i.e. Euler Method and Randomize Simulation Order option) have been chosen for the simulation of the proposed SDBA.
The comparative results shown here are achieved with the cell size $= 18 \text{ nm} \times 18 \text{ nm}$, which is used in the recent designs called CFA-based and CLA-based BCD adder for a fair comparison. The comparative analysis of the proposed SDBA design with existing designs based on parameters such as cell count, area, delay, the number of clock zones and overall cost are clearly explained as follows. Figure 5.4 shows the comparison of the proposed SDBA with respect to the number of cells (cell count) requirement. It is shown in Figure 5.4 that the cell count of SDBA method is 987 for one full BCD addition and observed that the SDBA method needs comparatively lower cell count compared to the other existing methods.

**Fig. 5.3** Simulation results of the proposed SDBA design
The comparison of the proposed SDBA with other existing BCD adder methods with respect to the cell area consumption is shown in Figure 5.5.

**Fig. 5.4** Comparison of SDBA performance in terms of number of cells requirement with existing methods

**Fig. 5.5** Comparison of SDBA performance in terms of cell area with existing methods
It is observed from Figure 5.5 that the cell area requirement of SDBA method is $1.24 \mu m^2$ for one full BCD addition, and the SDBA method occupies a reduced area in terms of the number of cells used when compared with the other existing methods. The comparison of the proposed SDBA with the other known BCD adder methods respectively to the latency (clock cycles) is shown in Figure 5.6.

Figure 5.6 Shows the latency (clock cycles) of SDBA method is 2.75 for one full BCD addition, and the SDBA method takes comparatively less latency (clock cycles) compared to the other existing methods. However, the latency required is slightly higher than CLA-based BCD adder (2.5), it is because of an increase in a multiplexer in the intermediate stage. The proposed SDBA method uses CFA Type-II binary adder and four multiplexers to obtain the required function. Whereas, the CLA-based BCD adder uses generate and propagate structure along with three multiplexers in its architecture.

![Comparison of SDBA performance in terms of latency (clock cycles) with other existing methods](image-url)
The comparison of the proposed SDBA with other known BCD adder methods in QCA with respect to the clock zones is shown in Figure 5.7. It is clearly observed from the Figure 5.7 that the SDBA method takes comparatively fewer clock zones compared to the other existing methods. In the recent efficient BCD adder designs [88], the CFA-based BCD adder occupies the area of 1.36 μm² and with the latency of 4.75 clock cycles whereas CLA-based BCD adder occupies the area of 1.86 μm² and with the latency of 2.5 clock cycles. From the above discussions, it is clearly evident that the proposed SDBA design performs the BCD addition is quicker than the earlier designs with the latency of 2.75 clock cycles only which is closer to CLA-based BCD adder, and the area is 1.24 μm².

![Comparison of SDBA performance in terms of clock zones with existing methods](image)

**Fig. 5.7** Comparison of SDBA performance in terms of clock zones with existing methods

The measure of circuits complexity and overall cost function proposed by Thomson related to the area and delay is \( A \times T^2 \) where \( A \) is an area of the circuit or chip, and \( T \) is the delay (latency) or computation time [124]. According to the above relation proposed by Thomson the overall cost function of the CFA-based BCD adder is 31 and
CLA-based BCD adder is 12 whereas the proposed SDBA is only 9. Hence, the overall cost of the proposed SDBA design is also relatively reduced by 71% and 25% when compared to CFA and CLA-based BCD adder respectively.

The proposed BCD adder designs cost function is calculated according to the VLSI chip or circuit instead of QCA cost function since all the BCD adder designs are using the multi-layer crossovers. Hence, the generalized cost function is appropriate to give the best comparison. The comparison of the proposed SDBA with other existing BCD adders with respect to the overall cost (Area x Latency²) is shown in Figure 5.8. The overall cost of SDBA method is 9 for one full BCD addition, which costs comparatively less compared to the other existing methods.

![Comparison of SDBA performance in terms of overall cost (Area x Latency²) with existing BCD adders](image)

**Fig. 5.8** Comparison of SDBA performance in terms of overall cost (Area x Latency²) with existing BCD adders
The proposed SDBA is the best-optimized design, and it allows the trade-off between CFA-based and CLA-based BCD adder. The area of CFA-based BCD and the proposed SDBA design are competitive by utilizing 987 cells with higher speed. The speed of both the CLA-based BCD design and proposed SDBA design are approximately equal, and the complexity of the proposed SDBA is reduced by improving the regularity of the structure.

Though, the proposed SDBA is the best-optimized design for single-digit decimal addition. However, in a conventional ripple carry BCD adder design, the overhead of the delay is caused due to the carry propagation in the upper 4-bit adder, and this has more impact on multi-digit BCD addition. Hence, this is motivated to design a high-speed single-digit BCD adder, which is primarily suitable for multi-digit BCD adders. The following section of this chapter, explains in detail regarding the proposed BCD adder with new correction logic to improve the speed of the decimal addition.

5.4 Design Methodology of ESDBA

The conventional single-digit BCD adder consists of three parts, namely, upper 4-bit binary adder to add two decimal numbers using 8421 code. Correction logic to detect the invalid BCD sum and if any invalid BCD sum ((> 9 and ≤ 19) or (sum ≤ 9 and carry out from MSB)) is detected then to get valid BCD output lower binary adder is used. The overall delay of the conventional BCD adder structure is predominantly imposed by ripple propagation of input carry.

In the proposed ESDBA design, the input carry is given to the lower 4-bit adder instead of giving to the upper 4-bit adder to anticipate the ripple carry propagation overhead. All the bits in the upper part of BCD digits are added concurrently without waiting for input carry propagation in the upper part of the design, which is shown in Figure 5.9. The design approach of proposed ESDBA is described in brief as follows. For single-digit BCD adder, the area-delay efficient binary adder [125] is modified for 4-bit binary addition, novel correction logic and four bit-RCA structure with CFA Type-II [29] is used.
The Area-efficient 4-bit binary adder is designed using the following expressions in terms of majority gate.

\[ C_1 = M(A_0, B_0, 0) \]  \hspace{2cm} (5.13)

\[ C_2 = M(A_1, B_1, C_1) \]  \hspace{2cm} (5.14)

\[ C_3 = M(M(A_2, B_2, 0), M(A_2, B_2, 1), C_2) \]  \hspace{2cm} (5.15)

\[ C_4 = M(M(A_3, B_3, M(A_2, B_2, 0)), M(A_3, B_3, M(A_3, B_3, 1)), C_3) \]  \hspace{2cm} (5.16)

\[ S_0 = M(M(A_0, B_0, 0), M(A_0, B_0, 1), 0) \]  \hspace{2cm} (5.17)

\[ S_1 = M(M(A_1, B_1, C_2), C_1, C_2) \]  \hspace{2cm} (5.18)

\[ S_2 = M(M(M(A_2, B_2, 1), M(A_2, B_2, 0), C_3), C_2, C_3) \]  \hspace{2cm} (5.19)

\[ S_3 = M(M(A_3, B_3, C_4), C_3, C_4) \]  \hspace{2cm} (5.20)

**Fig. 5.9** Block diagram of proposed ESDBA design
The new correction logic is developed, to improve the speed of single-digit BCD adder, by considering the following conditions and the corresponding logic is described with following three cases.

Proposed correction logic conditions are:

1. If the Binary sum > 9
2. If else the Sum = 9 and input carry (C_in) = 1
3. Else the binary sum is < 9 and a carry out (C_4) from MSB = 1, then

\[ X_0 = (S_3S_2S_1S_0) > 9 + ((S_3S_2S_1S_0) = 9)*C_{in} + C_4 \]  

(5.21)

Where Binary sum = S_3S_2S_1S_0

The logic expression obtained by simplifying the condition given in Equation (5.21) is

\[ X_0 = S_3S_2 + S_3S_1 + S_3S_0C_{in} + C_4 \]  

(5.22)

The expression (5.22) can be modified to optimize the circuit area is as follows

\[ X_0 = S_3(S_2 + S_1) + S_3S_0C_{in} + C_4 \]  

(5.23)

The new logic expression for correction logic based on Equation (5.23) is formulated in terms of majority gate; \( X_0 \) is written as follows.

\[ X_0 = M[M(M(S_3M(S_2S_11),0),C_41),M(M(S_3,S_00),C_{in}0),1] \]  

(5.24)

If \( X_0 = 1 \), then six (0110) is added to the lower 4-bit adder to convert the invalid BCD sum to valid BCD sum else no correction is required.

Where final BCD/ Decimal sum = \( X_0D_3D_2D_1D_0 \)

By using equations (5.13-5.20), the area efficient binary adder is constructed. Equation (5.24) is used to construct the correction logic. Using CFA Type-II binary adder in the final part, the optimized single-digit BCD adder is designed and the corresponding circuit is shown in Figure 5.10.
The resultant of the proposed single-digit BCD adder design layout is shown in Figure 5.11. In the proposed design, to compute the binary sum in the upper part generate and propagate circuitry is used instead of RCA. The output of the binary sum does not depend on the propagation of the input carry. So, it enhances the speed of computation.
operation to perform the decimal addition. Figure 5.11 utilizes 1205 cells with an area of 1.37 μm². From the layout, it is also noticed that the ESDBA takes $3\frac{1}{2}$ clock cycles, i.e., 14 clock zones to produce the final result of BCD addition. Out of 14 clock zones, 6 clock zones are required for generating the binary sum, 4 clock zones for correction logic and by suitable arrangement of correction logic majority gates and input carry, the lower CFA Type-II binary adder needs 4 clock zones.

**Fig. 5.11** Layout of proposed ESDBA design with new correction logic
5.5 Results and Discussions

The configuration settings considered in the QCADesigner tool are same as that of the existing methods for fair comparison of all parameters. Simulation results are obtained using Bistable approximation method. Figure 5.12 shows the simulation results of proposed ESDBA design. In Figure 5.12, the correct output of BCD addition is obtained after 3.5 clock cycles, and it is clearly indicated.

![Simulation Results](image)

Fig. 5.12 Simulation results of proposed ESDBA design
The performance of the proposed ESDBA is evaluated in terms of the cell count, total area, delay (clock cycles), clock zones and overall cost with the existing designs and it is clearly explained as shown below. Figure 5.13 shows the comparison of the proposed ESDBA about the cell count requirement. It appears from the Figure 5.13 that the cell count (number of cells) of ESDBA method is 1205 for one full BCD addition. It is also observed from Figure 5.13 that the ESDBA method requires comparatively lower cell count compared to the other existing methods, but slightly higher than the CFA-based BCD adder due to the generate and propagate functions to perform the parallel binary addition using without input carry in the upper part of the BCD addition.

![Comparison of ESDBA performance in terms of number of cells with existing methods](image)

**Fig. 5.13** Comparison of ESDBA performance in terms of number of cells with existing methods

The comparison of the proposed ESDBA with other existing BCD adder methods regarding the cell area is shown in Figure 5.14. It is observed from the Figure 5.14 that the proposed ESDBA design occupies 1.37 \( \mu \text{m}^2 \) and almost same area as the CFA-based BCD adder with a reduction of 1.25 clock cycle (five-clock zones) significant improvement in speed is achieved.
Fig. 5.14 Comparison of ESDBA performance in terms of cell area with existing methods

The comparison of the proposed ESDBA with other known BCD adder methods about the latency (clock cycles) is shown in Figure 5.15

Fig. 5.15 Comparison of ESDBA performance in terms of delay with existing methods
From Figure 5.15 it is observed that the delay (clock cycles) of ESDBA method achieved is 3.5 for one full BCD addition, and the ESDBA method takes comparatively less latency (clock cycles) compared to the other existing methods. The comparison of the proposed ESDBA with other known BCD adder methods in QCA with the clock zones is shown in Figure 5.16.

![Comparison of ESDBA performance regarding clock zones with existing methods](image)

**Fig. 5.16** Comparison of ESDBA performance regarding clock zones with existing methods

It is clearly observed from the Fig. 5.16 that the ESDBA method takes comparatively lesser clock zones compared to the other existing methods. The overall cost of the ESDBA evaluated by using CMOS cost function [124] is compared with the other existing methods is shown in Figure 5.17. Figure 5.17 evidently shows that the overall cost of the ESDBA is less compared to all the existing methods due to optimized area and delay.
The design performance can be compared with QCA cost function [96], which is explained in Equation (3.6) in Chapter 3, Section 3.3. As the number of majority gates is associated with compactness and crossovers is associated with difficulty in fabrication a double weighting is applied to $M$, i.e., $k = 2$, and also applied to $C$ (i.e., $l = 2$). The delay metric ($T$) weighting also be given with $p = 2$. If high speed is the primary concern, a higher weighting can be given to the delay metric i.e., $p = 4$. To evaluate the cost function of a 1-digit BCD adder here both metrics are considered with $l = 2, k = 2, p = 2$ and $l = 2, k = 2, p = 4$. The number of majority gates is $M$, number of inverters is $I$, number of coplanar wire crossovers $C_{cp}$, the number of multilayer crossovers $C_{ml}$ (where $C_{ml} = 3 C_{cp}$) and the number of clock cycles $T$.

The design of first QCA decimal adder requires $M = 27, I = 16, C_{ml} = 20$ and its delay $T = 8$ clock cycles [85]. The serial JMC decimal adder requires $M = 45, I = 18, C_{ml} = 14$ and its delay $T = 10$ clock cycles [86]. The details of parallel JMC adder was not available hence, it is not included for cost function comparison. QCA based efficient BCD adder structure consists of $M = 24, I = 14, C_{ml} = 21$ and with a delay of $T = 9$ clock cycles [87]. The CFA based BCD adder comprises of $M = 24, I = 14, C_{ml} = 18$, delay $T = 3.5$ clock cycles and CLA based BCD adder requires $M = 35, I = 13, C_{ml} = \ldots$
56, delay $T = 2.5$ clock cycles [88]. The proposed SDBA design entails $M = 31$, $I = 15$, $C_{ml} = 21$, delay $T = 2.75$ clock cycles and ESDBA design comprises of $M = 34$, $I = 12$, $C_{ml} = 28$, delay $T = 2.5$ clock cycles. The QCA cost function of different 1-digit BCD adders is comparison is shown in Figure 5.18.

The SDBA performance achieves significant improvement over the existing BCD adders due to design structure with respect to complexity. However, the ESDBA is complex than the SDBA design due to the more number of crossovers. Even though, the complexity is of ESDBA is more for single-digit it is competing with recent designs concerning the speed and it is clearly shown in Figure 5.19. Figure 5.19 displays the comparison of QCA cost function for various 1-digit BCD adders with primary concern of speed by giving the high weighting to the delay metric. The QCA cost function of SDBA is reduced by 57% compared to the existing designs with respect to complexity. The SDBA adder design is the best optimized design for single-digit BCD addition. The QCA cost function of SDBA is reduced by 75% compared with the other existing designs in the context of speed. Whereas, the ESDBA cost function is slightly than the SDBA and equal to the CLA based BCD design. However, the ESDBA performance predominates all the existing designs in multi-digit BCD addition and explained in detail in the next Chapter.

![Fig. 5.18 Comparison of BCD Adders QCA cost function in the context of Area](image-url)
The proposed ESDBA method is the best-optimized design, and it allows the trade-off between area and delay compared to CFA-based BCD adder design. The area of CFA-based BCD design and the proposed ESDBA design are competitive by utilizing 1205 cells with higher speed. Even though, the proposed ESDBA design has one clock cycle more than that of CLA-based BCD adder and ¾ clock cycle higher than the SDBA design while the number of digits is increased, it attains an increase in speed compared to all the other existing methods. The increase in the speed is possible because of the novel correction logic, also the complexity of the proposed design is reduced by improving the regularity of the structure, which is more useful for high-speed multi-digit BCD addition, and it is clearly described in the next Chapter.

5.6 Summary

In this chapter, we have presented two single-digit BCD adder designs to do the Decimal addition in a faster and optimized manner. Initially, an enhanced structure of SDBA design is proposed with the principle of Quantum-dot Cellular Automata. The proposed SDBA architecture makes the use of CFA Type-II binary adder and multiplexer as a hybrid structure. The proposed method achieves 46% of the cell count reduction compared to the existing CLA-based BCD adder with the same speed, and 42% reduction in the clock zones, nominal reduction in area compared to the CFA-
based BCD adder method. From all the obtained results, it is observed that the proposed SDBA design can process high-speed BCD addition with minimum area. The proposed SDBA design methodology can be applied to optimization of multi-digit BCD adder design for high-speed applications.

The second ESDBA design is implemented with area efficient binary adder using generate and propagate functions and introduces a new correction logic to anticipate the ripple carry propagation in the upper part of the existing BCD adder. The ESDBA offers the best trade-off among area-delay of CFA and CLA-based BCD adders. Even though the delay and overall cost of the ESDBA are more compared to the CLA-based design, and SDBA design it is well suitable to perform the multi-digit BCD addition with higher speed. The ESDBA design is mainly designed to overcome the speed limitation by the ripple carry binary adder to propagate the input carry from Least Significant Decimal Digit (LSDD) to the Most Significant Decimal Digit (MSDD) in the upper part in conventional multi-digit decimal addition. Furthermore, apart from the individual advantages of the two proposed single-digit BCD adder designs compared to existing BCD adders, both the SDBA and ESDBA designs are efficient while they are extended to construct the multi-digit BCD adders which are clearly explained in detail in the next chapter.