A Cost-Efficient Look-Up Table based Binary Coded Decimal Adder Design

Zarrin Tasnim Sworna*, Mubin Ul Haque*, Hafiz Md. Hasan Babu* and Lafifa Jamal*
*Department of Computer Science and Engineering, University of Dhaka, Dhaka-1000, Bangladesh.
Email:mubin10haque@gmail.com;sworna07@yahoo.com;hafizbabu@cse.univdhaka.edu;lafifa@yahoo.com
Corresponding author's email: hafizbabu@cse.univdhaka.edu, hafizbabu@du.ac.bd

***Abstract*—The Binary Coded Decimal (BCD) being the more accurate and human-readable representation with ease of conversion, is prevailing in the computing and electronic communication. In this paper, a tree-structured parallel BCD addition algorithm is proposed with the reduced time complexity $O(N(\log_2 b) + (N-1))$, where N = number of digits and b = number of bits in a digit. BCD adder is more effective with a LUT (Look-Up Table)-based design, due to FPGA (Field Programmable Gate Array) technology's enumerable benefits and applications. A size-minimal and depth-minimal LUT-based BCD adder circuit construction is the main contribution of this paper. The proposed parallel BCD adder gains a radical achievement compared to the existing best-known LUT-based BCD adders. The proposed BCD Adder provides prominent better performance with 20.0% reduction in area and 41.32% reduction in delay for the post-layout simulation. Since the proposed circuit is improved both in area and delay parameter, it is 53.06% efficient in terms of area-delay product compared to the best known existing BCD adder, which is surely a significant achievement.**

***Keywords*—Adder; Binary Coded Decimal (BCD); Field Programmable Gate Array (FPGA); Look-Up Table (LUT); correction**

I. Introduction

Binary Coded Decimal (BCD) representation is advantageous due to its finite place value representation, rounding, easy scaling by a factor of 10, simple alignment and conversion to character form [1], [2]. It is highly used in embedded applications, digital communication and financial calculations [3], [4]. Hence, faster and efficient BCD addition method is desired. In this paper, a N-digit addition method is proposed which omits the complex manipulation steps, reducing area and delay of the circuit. The application of FPGA in cryptography, NP (Non Polynomial)-Hard optimization problems, pattern matching, bioinformatics, floating point arithmetic, molecular dynamics is increasing radically [5], [6], [7]. Due to re-configurable capabilities, FPGA implementation of BCD addition is of concern. LUT being one of the main components of FPGA, a LUT-based adder circuit is proposed.

Two main contributions are addressed in this paper. Firstly, a new tree-based parallel BCD addition algorithm is presented. Secondly, a compact and high-speed BCD adder circuit with an improvement in time complexity of $O(N(\log_2 b)+(N-1))$ is proposed, where N represents the number of digits and b represents the number of bits in a digit.

The organization of this paper is as follows: In the next section, the existing approaches and their limitations are described. Section III introduces a novel BCD addition method. Then, the BCD adder circuit construction is presented. In Section IV, the simulation results and performance analysis of the proposed circuit are illustrated. Last of all, Section V concludes the paper.

II. Literature Overview

This section presents various types of the latest existing LUT-based BCD adders.

A. Existing LUT-Based BCD Adders

BCD adder uses BCD numbers as input and output [8]. For a 4-bit binary code, there are 16 different binary combinations in total. If the addition of two BCD digits exceeds the largest BCD digit $(9)_{10} = (1001)_{BCD}$, it may produce incorrect BCD output [8]-[10]. For such cases, the result must be corrected by adding $(6)_{10} = (0110)_{BCD}$ to ensure that the result is correct BCD output. Then the resultant decimal carry output generated by the correction process is added to the next higher digit of the BCD addends.

In [9], authors proposed a direct implementation of BCD adder circuit. They had proposed two different architectures for the construction of LUT-based BCD adder circuit. A truth table had been formed for each input/output combination and the corresponding circuit was proposed in first architecture. It consumed eleven 6-input LUTs. Two level of abstraction was performed for the second architecture. First least two significant input bits were fed into the first level of circuits whereas the rest input bits along with the output of the first level were provided to the second level of the circuits. The second approach required seven number of 6-input LUTs with a much delay. The direct implementation suffers a significant LUT-delay product.

The BCD adder proposed in [10] used Virtex-6 platform to implement their circuit architecture which had been proposed earlier in [11]. Gao et al. proposed a BCD adder, where the first bit of the addends are added using a full adder and the most significant three bits are added using 6-input LUTs [11]. A correction is ensured in 6-input LUTs by adding $(3)_{10}$ to the sum if the sum of the most significant three bits is greater than or equals to $(5)_{10}$. Moreover, extra circuits are required, when the sum of the most significant three bits is $(4)_{10}$ and the carry generated from the full adder is one. The circuit being serial in architecture except the LUTs portions, requires much time complexity and delay which hinders faster output generation [11]. Bioul et al. proposed a BCD adder, where additions were performed in a carry chain type fashion and thus, suffered

from a significant amount of delay [12]. They used Virtex-4 and Virtex-5 platform to show that the area overhead (in terms of required number of LUTs) with respect to binary computation is not negligible and it is around five times in Virtex-4 and nearly four times in Virtex-5. The main reason of such difference is due to the more complex definition of the carry propagate and carry generate functions.

In [13], authors proposed a BCD addition method, where six is added as a correction factor, when the sum of $A_i^U + B_i^U$ equals or greater than 8, where A_i^U and B_i^U represent the most significant three bits of the input operands A and B, respectively. If the final output is $(111)_2$, then a replacement of $(111)_2$ with $(100)_2$ is required as a final step for the exact BCD output. Vazquez et al. presented various carry chain BCD addition methods and their implementations on the LUT architecture [14]. As the carry-chain mechanisms being serial in architecture, the proposed methods in [14] require much delay which are surely a huge drawback.

A power and area-efficient BCD adder circuit was proposed by the authors in paper [15]. They actually used the circuit architecture exhibited in [1] and estimate the power consumption of the circuit. The delay has been calculated on a Virtex-5 platform by using 6-input LUT and the value obtained was 6.22 ns. The average power consumption of the circuit described in [15] was 25 mW which achieved a significant improvement over conventional LUT-based BCD adder. However, the method proposed in [15] required a total of 48 logic elements which can be optimized further.

III. Proposed Design of LUT-based BCD Adder

In this section, firstly a parallel BCD addition algorithm is proposed. Then, a new LUT-based BCD adder circuit is constructed. Essential figures and lemmas are elucidated to clarify the proposed ideas.

A. Proposed Parallel BCD Addition Method

The carry propagation is the main cause of delay of BCD adder circuit, which gives BCD adder a serial architecture. As the reduction of delay is one of the most important factor for the efficiency of the circuit, carry propagation mechanism needs to be removed for faster BCD addition. In this paper, a highly parallel BCD addition method is proposed with a tree-structured representation with significant reduction of delay. The proposed BCD addition method has mainly two steps which are as follows:

- *Bit-wise addition of the BCD addends produce the corresponding sum and carry in parallel. For the addition of first bit, the carry from the previous digit will be added too and the produced sum will be the direct first bit of the output.*
- *If the most significant carry bit is zero then, except the first sum and last carry bit, add the other sum and carry bits in pair in parallel; and if the sum is greater than or equals to five, add three to the result to obtain the correct BCD output.*
- *If the most significant carry bit is one then, update the final output values according to (1) and (2).*

Suppose, A and B be the two addends of a 1-digit BCD adder, where BCD representations of A and B are $A_4A_3A_2A_1$ and $B_4B_3B_2B_1$, respectively. The output of the adder will be a 5-bit binary number $\{C_{out}S_3S_2S_1S_0\}$, where C_{out} represents the position of tens digit and $\{S_3S_2S_1S_0\}$ symbolizes unit digit of BCD sum. A_0 and B_0 are added along with C_{in} which is the carry from the previous digit addition. If it is the first digit addition, the carry will be considered as zero. The produced sum bit will be the direct first bit of the output. Other pairwise bits (B_1, A_1), (B_2, A_2), (B_3, A_3) will be added simultaneously. The resultant sum and carry bits $(S_3^\alpha, C_2, S_2^\alpha, C_1, S_1^\alpha$ and $C_0)$ are added pairwise providing output $\{C_{out}^\gamma\ S_3^\gamma\ S_2^\gamma\ S_1^\gamma\ \}$ and corrected by addition of three according to the following (1) and (2):

$$C_{out}^\gamma S_3^\gamma S_2^\gamma S_1^\gamma$$

$$= \begin{cases} (C_{out}^\gamma S_3^\gamma S_2^\gamma S_1^\gamma), & \text{if } C_3 = 0\, and C_{out}^\gamma S_3^\gamma S_2^\gamma S_1^\gamma < 5 \\ (C_{out}^\gamma S_3^\gamma S_2^\gamma S_1^\gamma) + 3, & \text{if } C_3 = 0\, and C_{out}^\gamma S_3^\gamma S_2^\gamma S_1^\gamma \geq 5 \\ 1C_0 S_2^\beta S_1^\beta, & \text{otherwise} \end{cases} \tag{1}$$

$$\text{where} \quad S_1^\beta = S_2^\beta = \begin{cases} 0, & \text{if } C_0 = 1 \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

In Table I, the truth table is designed with (A_3, A_2, A_1) and (B_3, B_2, B_3) as input and $(C_{out}\ S_3\ S_2\ S_1)$ as the final BCD output by following required correction. $(S_3^\alpha, C_2, S_2^\alpha, C_1, S_1^\alpha$ and $C_0)$ are added pairwise as intermediate step, producing (F_4, F_3, F_2, F_1) by considering carry C_0 always 1. A numeric $3((011)_2)$ is added to the intermediary output F, if F is greater than or equals to five. A similar table considering C_0 as 0 can be calculated which is shown in Table II. The truth tables verify the functions of each output of the LUTs of the BCD adder. The algorithm of N-digit BCD addition method is presented in Algorithm 1.

Two example of BCD addition method using the proposed algorithm is demonstrated in Fig. 1 and 2, where $C_3^i = 0$ and $C_3^i = 1$, respectively. Each step of the example is mapped to the corresponding algorithm step for more clarification.

The proposed BCD addition method can be represented as a tree-structure as it is parallel which is shown in Fig. 3. There are basically two operational levels of the tree. Starting from the inputs, in level 1, the bit-wise addition is performed and the intermediary resultants are obtained. Then, in level 2, the addition and correction are performed providing the correct BCD output. Hence, the time complexity of the proposed algorithm is logarithmic according to the operational depth of the tree. Lemma 3.1 is given to prove the time complexity of our proposed method. The time complexity of existing and proposed BCD adders are elucidated in Table III.

Lemma 3.1 *The proposed BCD addition algorithm requires at least $O(N(log_2 b) + (N - 1))$ of time complexity, where N is number of BCD digits and b is the number of bits in a digit.*

TABLE I. THE TRUTH TABLE OF 1-DIGIT BCD ADDITION WITH $C_0 = 1$

$B(3:1)$	$A(3:1)$	$S^{\alpha}(3:1)$	$C(3:1)$	C_0	$F(4:1)$	Add 3	C_{out}	S_3	S_2	S_1
000	001	001	000	1	0010	-	0	0	1	0
000	010	010	000	1	0011	-	0	0	0	0
000	011	011	000	1	0100	-	0	0	0	0
000	100	100	000	1	0101	Add 3	1	0	0	0
001	001	000	001	1	0011	-	0	0	0	0
001	010	011	000	1	0100	-	0	0	0	0
001	011	010	001	1	0101	Add 3	1	0	0	0
001	100	101	000	1	0110	Add 3	1	0	0	1
.	.	.	.				.			
.	.	.	.				.			
.	.	.	.				.			
100	001	101	000	1	0110	Add 3	1	0	0	1
100	010	110	000	1	0111	Add 3	1	0	1	0
100	011	111	000	1	1000	Add 3	1	0	1	1
100	100	000	100	1	1001	Add 3	1	1	0	0

'-' Represents "No correction by adding 3 is required."

TABLE II. THE TRUTH TABLE OF 1-DIGIT BCD ADDITION WITH $C_0 = 0$

$B(3:1)$	$A(3:1)$	$S^{\alpha}(3:1)$	$C(3:1)$	C_0	$F(4:1)$	Add 3	C_{out}	S_3	S_2	S_1
000	001	001	000	0	0001	-	0	0	0	1
000	010	010	000	0	0010	-	0	0	1	0
000	011	011	000	0	0011	-	0	0	1	1
000	100	100	000	0	0100	-	0	1	0	0
001	001	000	001	0	0010	-	0	0	1	0
001	010	011	000	0	0011	-	0	0	1	1
001	011	010	001	0	0100	-	0	1	0	0
001	100	101	000	0	0101	Add 3	1	0	0	0
.	.	.	.				.			
.	.	.	.				.			
.	.	.	.				.			
100	001	101	000	0	0101	Add 3	1	0	0	0
100	010	110	000	0	0110	Add 3	1	0	0	1
100	011	111	000	0	0111	Add 3	1	0	1	0
100	100	000	100	0	1000	Add 3	1	0	1	1

'-' Represents "No correction by adding 3 is required."

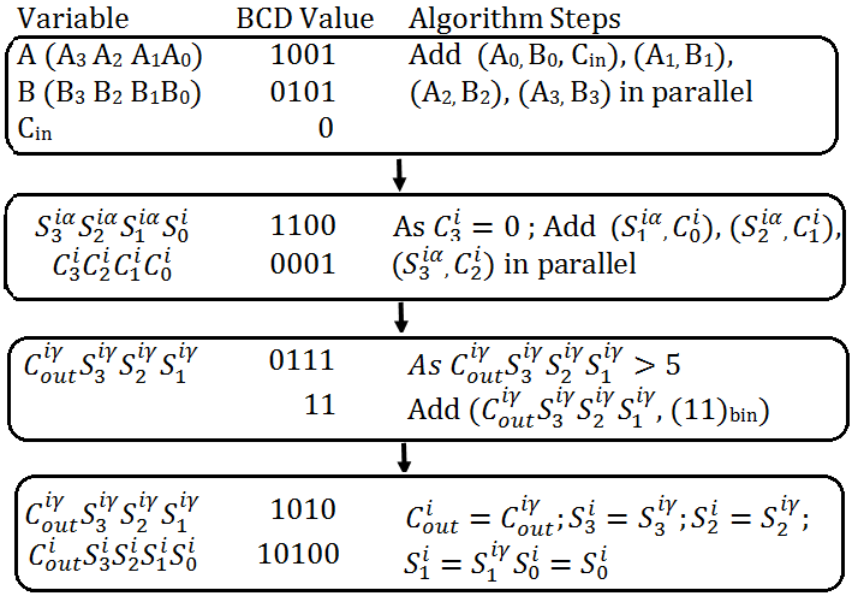

Fig. 1. Example Demonstration of the Proposed BCD Addition Algorithm for $C_3^i = 0$.

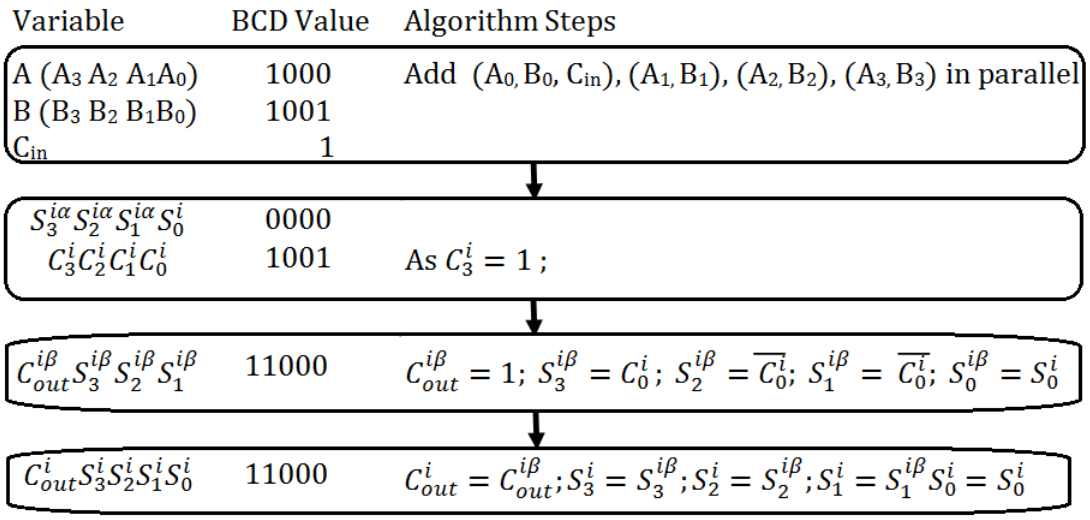

Fig. 2. Example Demonstration of the Proposed BCD Addition Algorithm for $C_3^i = 1$.

Algorithm 1: Proposed Algorithm for an N-digit Parallel BCD Addition

Input: Two N-digit BCD numbers
$A = A^N ... A^3 A^2 A^1$ and $B = B^N ... B^3 B^2 B^1$
where $A^i = A_3^i A_2^i A_1^i A_0^i$ and
$B^i = B_3^i B_2^i B_1^i B_0^i$ with $i = 1, 2, 3, ..., N$;
Output: Sum, $S = S^N ... S^3 S^2 S^1$ where
$S^i = S_3^i S_2^i S_1^i S_0^i$ with $i = 1, 2, 3, ..., N$ and
Carry, $C = C_{out}^N$;
$i \leftarrow 1$;
repeat
$S_0^i \leftarrow A_0^i \oplus B_0^i \oplus C_{in}^i$ and $C_0^i \leftarrow A_0^i.B_0^i.C_{in}^i$;
$S_1^{i\alpha} \leftarrow A_1^i \oplus B_1^i$ and $C_1^i \leftarrow A_1^i.B_1^i$;
$S_2^{i\alpha} \leftarrow A_2^i \oplus B_2^i$ and $C_2^i \leftarrow A_2^i.B_2^i$;
$S_3^{i\alpha} \leftarrow A_3^i \oplus B_3^i$ and $C_3^i \leftarrow A_3^i.B_3^i$ in parallel;
if $(C_3^i = 1)$ **then**
$S_3^{i\beta} \leftarrow C_0$; $C_{out}^{i\beta} \leftarrow 1$;
if $C_0^i = 0$ **then**
$S_1^{i\beta} \leftarrow 1$; $S_2^{i\beta} \leftarrow 1$;
else
$S_1^{i\beta} \leftarrow 0; S_2^{i\beta} \leftarrow 0$
end
else
$C_{out}^{i\gamma}\ S_3^{i\gamma}\ S_2^{i\gamma}\ S_1^{i\gamma} \leftarrow (S_3^{i\alpha} S_2^{i\alpha} S_1^{i\alpha}) + (C_2^i C_1^i C_0^i)$;
if $C_{out}^{i\gamma}\ S_3^{i\gamma}\ S_2^{i\gamma}\ S_1^{i\gamma} \geq 5$ **then**
$C_{out}^{i\gamma} S_3^{i\gamma} S_2^{i\gamma} S_1^{i\gamma} \leftarrow C_{out}^{i\gamma} S_3^{i\gamma} S_2^{i\gamma} S_1^{i\gamma} + 3$;
end
end
if $(C_3^i = 1)$ **then**
$S_1^i \leftarrow S_1^{i\beta}; S_2^i \leftarrow S_2^{i\beta}; S_3^i \leftarrow S_3^{i\beta}; C_{out}^i \leftarrow C_{out}^{i\beta}$;
else
$S_1^i \leftarrow S_1^{i\gamma}; S_2^i \leftarrow S_2^{i\gamma}; S_3^i \leftarrow S_3^{i\gamma}; C_{out}^i \leftarrow C_{out}^{i\gamma}$;
end
until $i = N$;

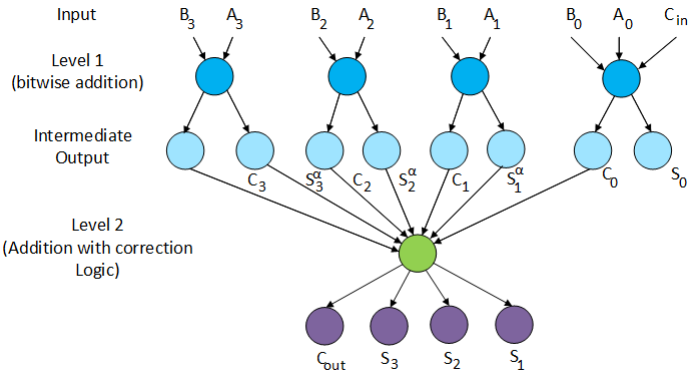

Fig. 3. Tree Structure Representation of the Proposed BCD Addition Method.

TABLE III. COMPARISON OF THE TIME COMPLEXITIES OF THE PROPOSED AND EXISTING BCD ADDITION METHODS

Method	Time Complexity
Existing [11]	$O(N(b+2)+(N-1))$
Existing [12]	$O(N(b+1)+(N-1))$
Existing [13]	$O(N(b+3)+(N-1))$
Existing [14]	$O(N(b+3)+(N-1))$
Proposed	$O(N(\log_2 b)+(N-1))$

'b':"number of bits in a digit" and 'N': "number of digits".

B. Proposed Parallel BCD Adder Circuit Using LUT

A LUT-based parallel BCD adder circuit is designed by using the proposed BCD addition algorithm and LUT architecture. An algorithm for the construction of the proposed BCD adder circuit is presented in Algorithm 2. According to the algorithm, the circuit is depicted in Fig. 4. For the addition of the least significant bit with carry from the previous digit addition, a full adder is used. Three half-adders are used for individual bit-wise addition operation of the most significant three bits. Depending on the value of C_3, (1) and (2) are followed in the proposed circuit architecture by using the transistors and LUTs, where four number of 6-input LUTs are used to add the output from the half-adders and full adder $\{S_3^\alpha, ..., S_1^\alpha, C_0\}$ with the correction by adding 3, if the sum is greater than or equals to five. Depending on the value of C_3,

a switching circuit is used to follow Equation 3. The proposed circuit gains huge delay reduction due to its parallel working mechanism compared to existing BCD adder circuits.

By using the proposed 1-digit BCD adder circuit, an N-digit BCD adder circuit can be constructed easily, where the C_{out} of one digit adder circuit is sent to the next digit of the BCD adder circuit as a C_{in}. Therefore, the generalized N-digit BCD adder computes sequentially by using the previous carry, the block diagram of which is shown in Fig. 5.

$$C_{out}S_3S_2S_1 = \begin{cases} C^{\beta}_{out}S^{\beta}_3S^{\beta}_2S^{\beta}_1, & \text{if } C_3 = 1 \\ C^{\gamma}_{out}S^{\gamma}_3S^{\gamma}_2S^{\gamma}_1, & \text{otherwise} \end{cases} \quad (3)$$

Algorithm 2: Proposed Algorithm for the Construction of an 1-Digit BCD Adder Circuit

Input: Two 1-digit BCD numbers $A = \{A_3A_2A_1A_0\}$ and $B = \{B_3B_2B_1B_0\}$;
Output: Sum $S = \{S_3S_2S_1S_0\}$ and Carry = C_{out};
Apply a full adder circuit where Input:= $\{A_0, B_0, C_{in}\}$ and Output:= $\{C_0, S_0\}$;
$i \leftarrow 1$;
repeat
  Apply a half adder circuit where Input:= $\{A_i, B_i\}$ and Output:= $\{C_i, S_i\}$;
until $(i = 3)$;
if $(C_3 = 1)$ **then**
  $S^{\beta}_3 \leftarrow C_0$; $C^{\beta}_{out} \leftarrow 1$;
  if $C_0 = 0$ **then**
    $S^{\beta}_1 \leftarrow 1$; $S^{\beta}_2 \leftarrow 1$;
  else
    $S^{\beta}_1 \leftarrow 0$; $S^{\beta}_2 \leftarrow 0$;
  end
else
  Apply four 6-input LUTs where each LUT's Input:= $\{S^{\alpha}_3, S^{\alpha}_2, S^{\alpha}_1, C_3, C_2, C_1\}$ and combined Output:= $\{C^{\gamma}_{out}\ S^{\gamma}_3\ S^{\gamma}_2\ S^{\gamma}_1\}$;
end
$j \leftarrow 1$;
repeat
  Apply a switching circuit where Input:= $\{S^{\gamma}_j, S^{\beta}_j\}$ and Output:= $\{S_j\}$;
until $(j = 3)$;
Apply fourth switching circuit where Input:= $\{C^{\gamma}_{out}, C^{\beta}_{out}\}$ and Output:= $\{C_{out}\}$;

IV. Simulation Results and Performance Analysis

As the BCD adder circuits being compared contain different types of logic gates and logic modules, we preserve the basic modules which can usually be found in a typical standard cell library. The area and delay of the proposed BCD adder circuits are derived and expressed in terms of the area and critical path delay of the available basic logic modules for different operator sizes. These theoretical estimates are then calibrated by the basic logic modules from *CMOS 45 nm open cell library* [16]. Table IV shows the area and critical path delay of basic logic gates. In this table, we have taken the core logic gates such as inverter, 2-input AND, OR and EX-OR gates. Table V calculates the area and critical path delay of some logic modules such as full adder, half adder and multiplexer by using the Table IV. It is required to mention that, the area has been calculated in terms of number of transistors.

TABLE IV. Area and Critical Path Delay of Basic Logic Gates

Basic Logic Gates	Area (in transistors)	Critical Path Delay (ns)
Inverter (INV)	1	1
2-input AND	6	4.68
2-input OR	6	4.5
2-input EX-OR	8	4.72

TABLE V. Area and Critical Path Delay of Basic Logic Modules

Elements	Area (in transistors)	Critical Path Delay (ns)
2-to-1 Multiplexer (MUX)	20	10.18
Half Adder (HA)	14	4.72
Full Adder (FA)	34	13.9

The area complexity of the proposed BCD adder is derived from its basic logic modules. The proposed BCD adder requires three half adders, one full adder, four 6-input LUTs, six inverters and twenty six transistors. Thus, the total area of the proposed BCD adder ($A_{proposed}$) can be determined as follows:

$$A_{proposed} = (3 \times A_{HA}) + (1 \times A_{FA}) + (4 \times A_{6-LUT}) + (6 \times A_{INV}) + (26 \times A_{transistor}) \quad (4)$$

Table VI shows the comparison among the proposed and existing BCD adders in terms of area. It is evident from Table VI that the proposed design requires 108 transistors and four 6-input LUTs whereas the best known existing methods [10], [11] require 132 transistors and four 6-input LUTs. Thus the proposed BCD adder gains an improvement of 18.18% in terms of area for pre-layout simulation result. Similarly, the critical path delay of the proposed BCD adder contains one full adder, one 6-input LUT, two inverters and two transistors. Therefore, the critical path delay of the proposed BCD adder ($D_{proposed}$) can be calculated as follows:

$$D_{proposed} = 1 \times D_{FA} + 1 \times D_{6-LUT} + 2 \times D_{INV} + 2 \times D_{transistor} \quad (5)$$

Table VII shows the comparison among the proposed and existing BCD adders in terms of critical path delay. It is shown from Table VII that the proposed BCD adder requires 41.8 ns of delay whereas the best known existing methods [10] [11] require 69.56 ns of delay. Therefore the proposed BCD adder achieves an improvement of 39.9% in terms of critical path delay in pre-layout simulation result.

A. FPGA Implementation and Post-Layout Simulation Results

The proposed BCD adder was coded in VHDL and implemented in a Virtex-6 XC6VLX75T Xilinx FPGA with a -3 speed grade using by ISE 13.1. The results are compared with the earlier approaches proposed in [11]-[14] by using

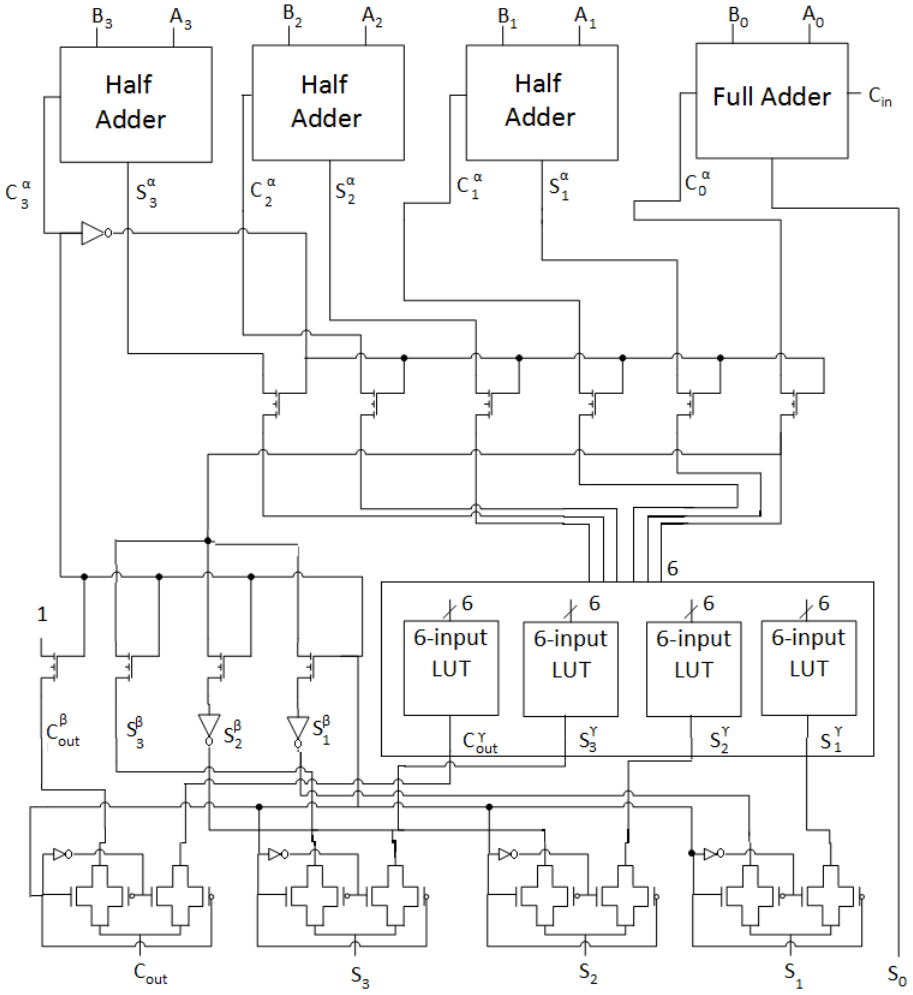

Fig. 4. Proposed 1-Digit BCD Adder Circuit.

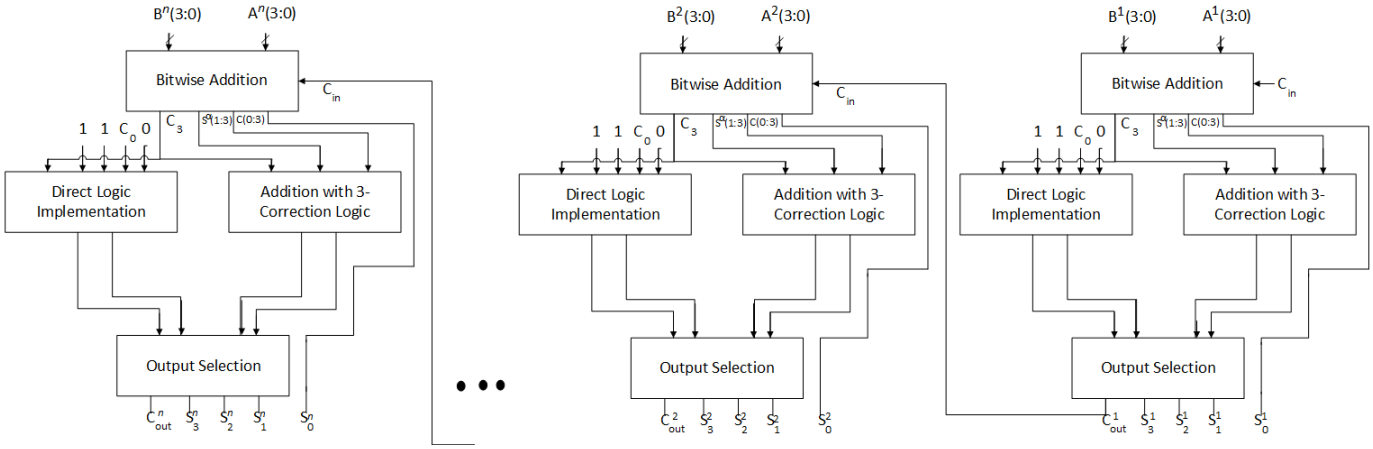

Fig. 5. Block Diagram of the Proposed N-Digit BCD Adder Circuit.

TABLE VI. COMPARISON OF AREA AMONG THE EXISTING AND THE PROPOSED N-DIGIT BCD ADDERS FOR PRE-LAYOUT SIMULATION

Method	Area Expression	Area*	LUT Count
Gao et al. [10], [11]	$N \times (1 \times A_{FA} + 3 \times A_{MUX} + 3 \times A_{Ex-OR} + 2 \times A_{INV} + 2 \times A_{AND} + 4 \times A_{6-LUT})$	$132N$	$4N$
Bioul et al. [12]	$N \times (8 \times A_{6-LUT} + 6 \times A_{MUX})$	$120N$	$8N$
Vazquez et al. [13]	$N \times (5 \times A_{6-LUT} + 4 \times A_{MUX} + 4 \times A_{Ex-OR} + 2 \times A_{INV} + 2 \times A_{AND})$	$134N$	$5N$
Vazquez et al. [14]	$N \times (8 \times A_{6-LUT} + 7 \times A_{MUX} + 8 \times A_{Ex-OR})$	$204N$	$8N$
Proposed	$N \times (3 \times A_{HA} + 1 \times A_{FA} + 4 \times A_{6-LUT} + 6 \times A_{INV} + 26 \times A_{transistor})$	$108N$	$4N$

'*' Represents "Area has been calculated in terms of transistors."

TABLE VII. COMPARISON OF DELAY AMONG THE EXISTING AND THE PROPOSED N-DIGIT BCD ADDERS FOR PRE-LAYOUT SIMULATION

Method	Delay Expression	Critical Path Delay (ns)
Gao et al. [10], [11]	$N \times (1 \times D_{FA} + 2 \times D_{MUX} + 1 \times D_{Ex-OR} + 1 \times D_{INV} + 1 \times D_{AND} + 1 \times D_{6-LUT})$	$69.56N$
Bioul et al. [12]	$N \times (4 \times D_{6-LUT} + 4 \times D_{MUX})$	$140.72N$
Vazquez et al. [13]	$N \times (1 \times D_{6-LUT} + 4 \times D_{MUX} + 2 \times D_{Ex-OR} + 1 \times D_{INV} + 1 \times D_{AND})$	$80.74N$
Vazquez et al. [14]	$N \times (4 \times D_{6-LUT} + 4 \times D_{MUX} + 6 \times D_{Ex-OR})$	$168.64N$
Proposed	$N \times (1 \times D_{FA} + 1 \times D_{6-LUT} + 2 \times D_{INV} + 2 \times D_{transistor})$	$41.8N$

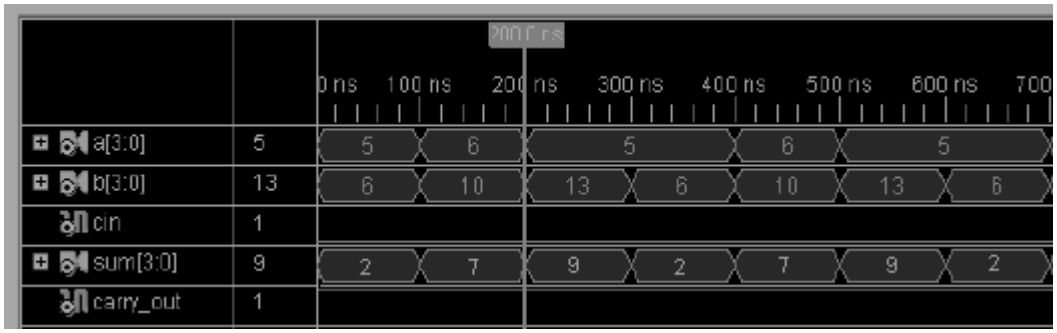

Fig. 6. Simulation Result of BCD Adder with Intermediate Carry C1= 1.

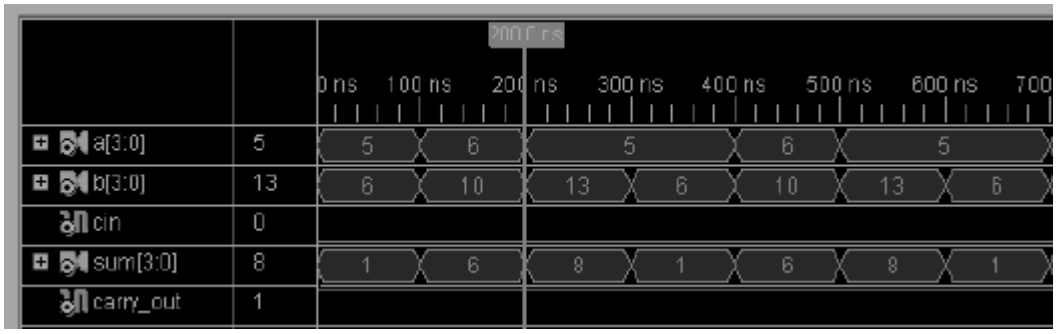

Fig. 7. Simulation Result of BCD Adder with Intermediate Carry C1= 0.

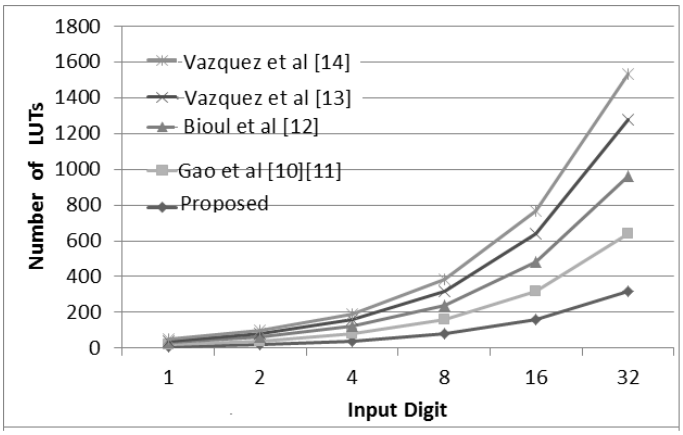

Fig. 8. Graphical analysis of area of existing and proposed BCD adder circuits for post-layout simulation.

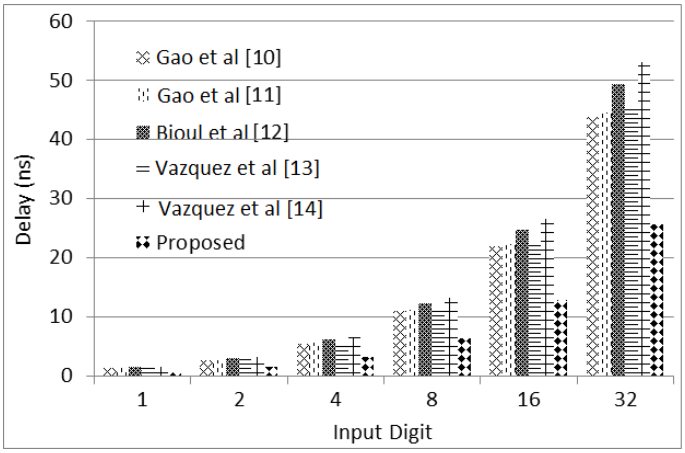

Fig. 9. Graphical analysis of delay of existing and proposed BCD adder circuits for post-layout simulation.

the same experimental setup for fair comparison. The delays were extracted from Postplacement-and-Routing Static Timing Report and the LUTs usage was obtained from Place-and-Routing Report. Besides, the simulations of the proposed BCD adder are demonstrated in Fig. 6 and 7 with carry 1 and 0, respectively.

The proposed BCD adder is high-speed due to its less time complexity with optimum critical path delay and cost-efficient due to its area and area-delay product efficiency. Comparison of area, delay and area-delay product among existing [11]-[14] and the proposed BCD adder circuits for various number of input digits are shown in graphical representation in Fig. 8, 9 and 10, respectively with improvement of 20%, 41.32% and 53.06% in terms of area, delay and area-delay product, respectively compared to the existing best method [10], [11]. It is to be noted that, the results shown in Fig. 8, Fig. 9 and 10 for earlier approaches [11]-[14] have been re-implemented by using Virtex-6 platform.

V. Conclusion

Reconfigurable computing has now become a better alternative to Application-Specific Integrated Circuits (ASICs) and fixed microprocessors. Besides, BCD (Binary Coded Decimal)

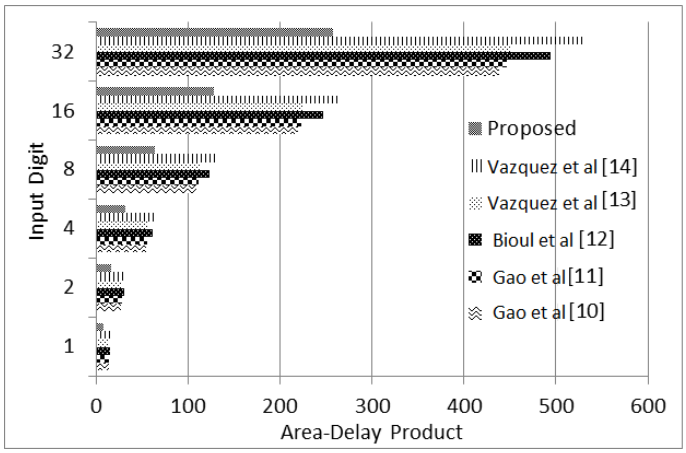

Fig. 10. Graphical analysis of area-delay product of existing and proposed BCD adder circuits.

addition being the basic arithmetical operation, it is the main focus. The proposed BCD adder is highly parallel, which mitigates the significant carry propagation delay of addition operation. The proposed BCD adder circuit is not only faster but also area-efficient compared to the existing best known circuit. The pre-layout simulation provides 18.18% and 39.9% efficiency in terms of area and critical path delay reduction, respectively compared to the existing best known BCD adder circuit. The proposed BCD adder circuit is simulated using Xilinx Virtex-6. The correctness and efficiency of the circuit is proved in the proposed section and simulation section using corresponding tables, figures and lemma. It is shown by the comparative analysis that the proposed BCD adder is 20% and 41.3% improved in terms of area and delay, respectively compared to the existing best known adder circuit along with 53.06% improvement in area-delay product. As it is more convenient to convert from decimal to BCD than binary, the proposed efficient FPGA-based BCD addition will subsequently influence the advancement in computation and manipulation of decimal digits. Besides, FPGA implementation will be beneficial to be applied in bit-wise manipulation, private key encryption and decryption acceleration, heavily pipe-lined and parallel computation of NP-hard problems, automatic target generation and many more applications [4], [5].

Acknowledgment

Zarrin Tasnim Sworna and Mubin Ul Haque has been granted fellowship from the Ministry of Information and Technology, People's Republic of Bangladesh under the program of higher studies and research with the reference no. 56.00.0000.028.33.058.15-629.

References

[1] Al-Khaleel, Osama, Mohammad Al-Khaleel, Zakaria Al-QudahJ, Christos A. Papachristou, Khaldoon Mhaidat, and Francis G. Wolff. *"Fast binary/decimal adder/subtractor with a novel correction-free BCD addition."* In Electronics, Circuits and Systems (ICECS), 2011 18th IEEE International Conference on, pp. 455-459. IEEE, 2011.

[2] Sundaresan, C., C. V. S. Chaitanya, P. R. Venkateswaran, Somashekara Bhat, and J. Mohan Kumar. *"High speed BCD adder."* In Proceedings of the 2011 2nd International Congress on Computer Applications and Computational Science, pp. 113-118. Springer, Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-28308-6_15.

[3] Z. T. Sworna, M. U. Haque and H. M. H. Babu. *"A LUT-based matrix multiplication using neural networks."* 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, Canada, 2016, pp. 1982-1985. DOI: 10.1109/ISCAS.2016.7538964.

[4] Z. T. Sworna, M. U. Haque, N. Tara, H. M. H. Babu and A. K. Biswas. *"Low-power and area efficient binary coded decimal adder design using a look up table-based field programmable gate array."* IET (The Institution of Engineering and Technology) Circuits, Devices & Systems, 2015, volume: 10, issue: 3, pp. 1-10. DOI: 10.1049/iet-cds.2015.0213.

[5] Pocek, Kenneth, Russell Tessier, and Andr DeHon. *"Birth and adolescence of reconfigurable computing: A survey of the first 20 years of field-programmable custom computing machines."* In Field-Programmable Custom Computing Machines (FCCM), 2013 IEEE 21st Annual International Symposium on, pp. 1-17. IEEE, Seattle, WA, USA, 2013. DOI: 10.1109/FPGA.2013.6882273.

[6] Han, Liu, and Seok-Bum Ko. *"High-speed parallel decimal multiplication with redundant internal encodings."* IEEE Transactions on Computers 62, no. 5 (2013): 956-968. DOI: 10.1109/TC.2012.35.

[7] Ning, Yonghai, Zongqiang Guo, Sen Shen, and Bo Peng. *"Design of data acquisition and storage system based on the FPGA."* Procedia Engineering 29 (2012): 2927-2931, Elsevier. DOI: 10.1016/j.proeng.2012.01.416

[8] G. Sutter, E. Todorovich, G. Bioul, M. Vazquez, and J.-P. Deschamps. 2009. *"FPGA Implementations of BCD Multipliers"*. In International Conference on Reconfigurable Computing and FPGAs, Quintana Roo, Mexico, 2009. ReConFig 09, pp: 3641. DOI: 10.1109/ReConFig.2009.28.

[9] O.D. Al-Khaleel, N.H. Tulic, and K.M. Mhaidat. *"FPGA implementation of binary coded decimal digit adders and multipliers."* In 8th International Symposium on Mechatronics and its Applications (ISMA), harjah, U.A.E 2012. DOI: 10.1109/ISMA.2012.6215199.

[10] Gao, Shuli, Dhamin Al-Khalili, J. M. Langlois, and Noureddine Chabini. *"Efficient Realization of BCD Multipliers Using FPGAs."* International Journal of Reconfigurable Computing 2017 (2017). DOI: 10.1155/2017/2410408.

[11] ShuliGao, D. Al-Khalili, and N. Chabini. 2012. *"An improved BCD adder using 6-LUT FPGAs."* In 10th International Conference on New Circuits and Systems (NEWCAS), 2012 IEEE , pp: 1316. DOI: 10.1109/NEWCAS.2012.6328944.

[12] G. Bioul, M. Vazquez, J. P. Deschamps, and G. Sutter. 2010. *"High-speed FPGA 10s Complement Adders-subtractors."* Int. J. Reconfig. Comput. 2010, Article 4 (Jan. 2010), DOI: 10.1155/2010/219764.

[13] Alvaro Vazquez and Florent De Dinechin. 2010. *"Multi-operand Decimal Adder Trees for FPGAs."* Research Report RR-7420. 20 pages. DOI: hal.inria.fr/inria-00526327.

[14] M. Vazquez, G. Sutter, G. Bioul, and J.P. Deschamps. 2009. *"Decimal Adders/Subtractors in FPGA: Efficient 6-input LUT Implementations."* In Reconfigurable Computing and FPGAs, 2009. ReConFig 09. International Conference on. 4247. DOI: 10.1109/ReConFig.2009.29.

[15] Mishra, Shambhavi, and Gaurav Verma. *Low power and area efficient implementation of BCD Adder on FPGA."* In Signal Processing and Communication (ICSC), 2013 International Conference on, pp. 461-465. IEEE, Noida, India, 2013. DOI: 10.1109/ICSPCom.2013.6719834.

[16] "CMOS 45 nm Open Cell Library". Available at http://www.si2.org/openeda.si2.org/projects/nangatelib Last access date: 14 March, 2017.