DOI: 10.21917/ijme.2025.0352

CMOS BASED ARCHITECTURE FOR HIGH SPEED BCD ADDITION

Chintam Shravan, Pitla Tharuni, Kummari Gayathri and Alli Vishwa Bhavani
Department of Electronics and Communication Engineering, Rajiv Gandhi University of Knowledge Technologies, India

Abstract

Accurate decimal calculations are a fundamental requirement in accurate-operated domains such as finance and scientific research. Traditional binary arithmetic circuits, although widely used, often introduce rounding inaccuracy that can cascade in adequate errors in sensitive applications. To address this challenge, this research examines solution by designing the decimal arithmetic circuit using the CMOS-based binary-coded decimal (BCD) Adders. Unlike software-level reforms, the proposed approach embedded accuracy in circuit design itself. Decimal Adders are modeling and valid through the rhythm using 90nm CMOS technology, which ensures high loyalty in logic implementation. In addition, both Cadence virtuoso and tanner are organized to evaluate demonstration matrix using wide simulation, power efficiency and operating delays using EDA equipment. The benchmark comparison with existing architecture reveals significant reforms, especially in power consumption. This study confirms the argument that integrating the decimal arithmetic directly into silicon can greatly increase both reliability and computational accuracy in important systems.

Keywords:

Decimal Processing, CMOS Design, BCD Adder's, Principal Arithmetic, Tanner EDA, Low Power Circuit

1. INTRODUCTION

Power Profiling of BCD Adders in Transistor-Level Design and scaled CMOS technologies since the demand for rapid and more energy-efficient integrated circuits increases, adaptation of arithmetic components has become a major design priority [1]. An important operation in the digital system requiring accurate decimal representation is in addition to using binary-coded decimal (BCD).

BCD arithmetic is important in areas such as financial computing, embedded processing and digital control systems, where traditional binary operation can introduce rounding and accurate errors. Unlike binary adders, BCD Adders offer more design complications due to their digit-wise computation and requirement for additional improvement logic [2-3]. This complexity often causes challenges in electricity usage, delays and reducing the field. High-level synthesis techniques, usually used for FPGA or ASIC design, remove transistor-level behavior required for intensive adaptation-especially in the context of modern nanoscale technologies. To remove these obstacles, this study introduces a custom, low level design method for BCD Adders using transistor-level circuit construction.

The proposed users were developed and analyzed within the table's chromosome and Tanner EDA environment. These platforms enabled the manufacture, simulation and power evaluation of BCD Adder Circuit using 90nm CMOS technology in the rhythm for layouts and verification, and tanner tools for power assessment in 45nm, 65nm and 90nm technology nodes several BCD additive configurations were designed using custom logic and pass-transistor techniques from 1-digit to digit implementation [4-6]. The design was scaled to various technology nodes so that the effect could be investigated.

2. LITERATURE REVIEW

Design and analysis of BCD Adder Circuit for High Purification Digital Applications. The binary coded decimal (BCD) is a foundation stone in the arithmetic digital system where the accurate decimal representation is non-prevent especially in financial processing, commercial computing and embedded control environment. Unlike the binary system, which encodes the entire values using Base-2, BCD provides each decimal digit with a separate 4-bit binary equivalent. This method allows spontaneous integration with decimal-based systems and usually prevents errors faced in floating-point binary operation [7-12]. However, the advantage of accuracy comes at the cost of increased arithmetic complexity, especially in addition. To handle multi-digit BCD added, a standard approach includes arranging several 1-digit BCD Adders in a sequel, ripple-carry configuration. In this structure, each stage processes a single decimal digit and passes any overflow as moved to the next stage. If the processing delay for a point is represented by T, an n-digit BCD joint will roughly delay nT, making the delays an important factor in overall circuit performance [13].

Each individual BCD additive performs two primary operations [14-16]. First, it adds a straight binary of two BCD digits. Second, it checks whether intermediate results require improvement. An improvement is either required when more than the decimal value of the resulting amount or when a carry is generated - those sections that violate the valid limits of the BCD output. To clarify, consider adding 4 and 5. In the binary, they are depicted as $(0100)_2$ and $(0101)_2$ respectively. Their binary sum is $(10011)_2$, or decimal 9, which is within the valid BCD range so no improvement is required. In contrast, adding 7 and 6 results in $(0111)_2 + (0110)_2 = (1101)_2$, equal to decimal 13, which is invalid in BCD. To correct this, a binary 6 (i.e., $(0110)_2$) is added, production $(0011)_2$ and a carry. It is divided into a valid digit and a carry, which is passed further. Even in cases where the result is more than 9, the generation of a carry can be invalid. For example, 8 + 9 yield $(1000)_2 + (1001)_2 = (0001)_2$ and a carry are generated. Although the lower 4 bits represent 1, the carry flag indicates that improvement is required. In the decimal $(0110)_2$ is added to yield $(0111)_2$ or decimal 7, ensuring that BCD rules are followed. Each 1-digit BCD adder is usually divided into two logical stages. The first is a binary adder that calculates raw yoga, while the second is responsible for detecting out-of-ranges output and implementing corrective arguments. This modular structure enables the implementation of scalable multi-digit and simplifies the reuse and expansion of the circuit.

For practical verification, all the proposed circuits were applied using the 90nm CMOS technological node, which remain relevant in educational investigation and low-power commercial solutions. The table was employed for planned design, simulation

and layouts, allowing accurate control over transistor-level behavior. Major performance indicators such as propagation delays, average power was measured and documented. These designs were then benchmark against the current architecture obtained from each other along with other architectures. Comparative analysis suggests that custom transistor-level implementation can make significant improvements in power efficiency and speed, especially when the application-specific integrated circuit (ASIC) is seen. Unlike high-level synthesized options, these provide more flexibility for low-level circuit power-delay tradeoffs and design space exploration.

This task provides a meaningful contribution in the field of arithmetic circuit design, indicating that adapted, transistor-level BCD additions are not only possible in terms of energy efficiency and accuracy but also beneficial. The findings are particularly relevant to professionals working on digital arithmetic blocks, VLSI systems development and low-power chip design.

3. PROPOSED BCD ADDER

In order to meet the increasing performance demands of modern digital systems, especially in domains such as financial computing, embedded electronics and commercial data processing-this work presents a high speed, improvement-free BCD (binary-coded decimal) appointment used to use CMOS logic. Traditional BCD Aders usually follow a two-step process: 4-bit binary joint performance when the amount exceeds 9 (1001_2) after a conditional improvement. Although reliable, this process slows down performance due to the additional improvement phase. Conversely, the proposed additive ends this post-processing step completely by integrating a novel two-to-stage approach. This architecture directly calculates the correct BCD amount without the requirement of a conditional fix, ensuring less delay and increased performance. The BCD adds two custom-designed CMOS Netlist using: Netlist1 and Netlist 2, who work in sequence, but adapted to parallel processing and minimal argument depth.

The adder accepts an alternative carry-in bit (Cin) as well as two 4-bit BCD inputs, which are defined as $A=A_3A_2A_1A_0$ and $B=B_3B_2B_1B_0$. For customized processing, high three bits of each operand are grouped ($J=A_3A_2A_1$ and $I=B_3B_2B_1$), while at least important bits A_0 and B_0 are distinguished. It yields:

$$A = (2 \times J) + A_0$$

$$B = (2 \times I) + B_0$$

This decomposition reduces the logic complexity by narrowing the range of intermediate values I and J between 0 and 4, which significantly simplifies additional logic. The joint becomes operation:

$$\{C_{out}, sum\} = A + B + C_{in} = 2 \times (i + j) + (A_0 + B_0 + C_{in})$$

$$K = I + J \text{ and } L = A_0 + B_0 + C_{in},$$

The final expression is simple:

$$\{C_{out}, sum\} = 2 \times K + L$$

Example case study for clarity, consider A = 6 (0110) and B = 7 (0111). We remove J = 3 and I = 3, give K = 6. Therefore, 2K = 12. If $A_0 = 0$, $B_0 = 1$, and $C_{in} = 1$, then L = 2. Result: 2K + L = 12 + 2 = 14 (0001 0100 in BCD) So, the sum is 4 (0100) and carry-out is 1.

Netlist1 K (I + J) is dedicated to calculating the BCD value, provides K_0 through K_3. Each bit is calculated using the truth table analysis and argument, using the derivative, optimized Nand Logic Expression (X_{24} from X_{24}). This ensures rapid calculation with a depth of low-level logic, and benefits from balanced drive strength and layout simplicity such as CMOS Nand Gate benefits. After receiving 2K, the Netlist2 becomes active, which adds a value of 2K from L. This logic network uses another layer of pre-ordered manifestations (Y_0 to Y_{26}) which covers all possible combinations of L and output from Netlist1. For example, with A = 4 (0100) and B = 5 (0101), we get I = J = 2, so K = 4 and 2K = 8. $A_0 = 0$, $B_0 = 1$, $C_{in} = 1$, then L = 2 is given. The total amount is 10 (BCD = 0001 0000), with the last carry of 1.

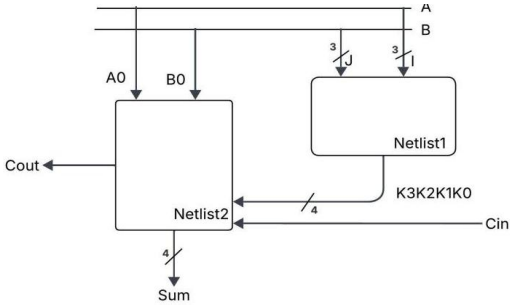

Fig.1. Block diagram of Efficient BCD Adder

By eliminating the traditional improvement phase, this design ensures the results of every possible input pair in a direct and valid BCD output. Each output bit is argued and is designed by static CMOS Gates, ensuring reliable operations under high noise margin, full voltage swing and separate process conditions. To enable multi-step operations, the adder supports cascading in a ripple-carry manner. Each unit can pass its carry-out in the next stage, enabling scalable BCD arithmetic for processors [17]. This modular architecture makes the proposed adder a strong candidate for integration in decimal arithmetic units found in advanced processors-like IBM Z196 [10-11], which handles decimal floating-point calculations. Speed and accuracy are maintained without introducing improvement-based delays.

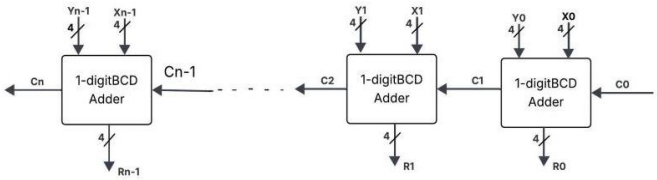

Fig.2. A Block diagram for n-digit BCD Adder

The proposed BCD digit adder implemented through CMOS technology provides a sharp, energy-skilled and improvement-free method for decimal joints. Its dual-tool approach ensures a low delay, while the design modularity allows for flexible integration in both simple and high-performance computing systems. Architecture is particularly beneficial where accurate decimal arithmetic is important, making it highly relevant to future processor design and embedded applications.

NETLIST-1 Outputs:

$$K_0 = \overline{X_0} \cdot \overline{X_1} \cdot \overline{X_2} \cdot \overline{X_3} \cdot \overline{X_4} \cdot \overline{X_5} \cdot \overline{X_6} \cdot \overline{X_7}$$

$$K_1 = \overline{X_0} \cdot \overline{X_8} \cdot \overline{X_9} \cdot \overline{X_{10}} \cdot \overline{X_{11}} \cdot \overline{X_{12}} \cdot \overline{X_{13}} \cdot \overline{X_7}$$

$$K_2 = \overline{X_{14}} \cdot \overline{X_{15}} \cdot \overline{X_{16}} \cdot \overline{X_{17}} \cdot \overline{X_{18}}$$

$$K_3 = \overline{X_{19}} \cdot \overline{X_{20}} \cdot \overline{X_{21}} \cdot \overline{X_{22}} \cdot \overline{X_{23}} \cdot \overline{X_{24}} \cdot \overline{X_7}$$

where,

$X_0 = \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot B_2 \cdot B_1 ; X_1 = A_1 \cdot \overline{B_3} \cdot \overline{B_2} \cdot \overline{B_1}$

$X_2 = \overline{A_3} \cdot \overline{A_1} \cdot \overline{B_2} \cdot \overline{B_1} ;\ X_3 = \overline{A_2} \cdot A_1 \cdot B_2 \cdot \overline{B_1}$

$X_4 = A_2 \cdot A_1 \cdot B_2 \cdot B_1 ;\ X_5 = A_2 \cdot \overline{A_1} \cdot B_3$

$X_6 = A_3 \cdot B_2 \cdot \overline{B_1} ;\ X_7 = A_3 \cdot B_3$

$X_8 = A_2 \cdot \overline{B_3} \cdot \overline{B_2} \cdot \overline{B_1}; X_9 = \overline{A_3} \cdot \overline{A_2} \cdot B_2 \cdot \overline{B_1}$

$X_{10} = A_2 \cdot \overline{A_1} \cdot \overline{B_2} \cdot B_1; X_{11} = \overline{A_2} \cdot A_1 \cdot \overline{B_2} \cdot B_1$

$X_{12} = A_2 \cdot A_1 \cdot B_3; X_{13} = A_3 \cdot B_2 \cdot B_1$

$X_{14} = \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot B_3; X_{15} = A_3 \cdot \overline{B_3} \cdot \overline{B_2} \cdot \overline{B_1}$

$X_{16} = A_2 \cdot \overline{A_1} \cdot B_2 \cdot \overline{B_1}; X_{17} = A_2 \cdot A_1 \cdot \overline{B_2} \cdot B_1$

$X_{18} = \overline{A_2} \cdot A_1 \cdot B_2 \cdot B_1; X_{19} = A_2 \cdot A_1 \cdot B_2$

$X_{20} = A_2 \cdot \overline{A_1} \cdot B_3; X_{21} = A_2 \cdot B_2 \cdot B_1$

$X_{22} = A_3 \cdot B_2 \cdot \overline{B_1}; X_{23} = A_1 \cdot B_3 ;\ X_{24} = A_3 \cdot B_1$

NETLIST-2 Outputs:

$$S_0 = \overline{Y_0} \cdot \overline{Y_1} \cdot \overline{Y_2} \cdot \overline{Y_3} ; S_1 = \overline{Y_4} \cdot \overline{Y_5} \cdot \overline{Y_6} \cdot \overline{Y_7} \cdot \overline{Y_8} \cdot \overline{Y_9}$$

$$S_2 = \overline{Y_{10}} \cdot \overline{Y_{11}} \cdot \overline{Y_{12}} \cdot \overline{Y_{13}} \cdot \overline{Y_{14}} \cdot \overline{Y_{15}} \cdot \overline{Y_{16}}$$

$$S_3 = \overline{Y_{17}} \cdot \overline{Y_{18}} \cdot \overline{Y_{19}} \cdot \overline{Y_{20}} \cdot \overline{Y_{21}} \cdot \overline{Y_{22}} ;\ C_{\text{out}} = \overline{Y_{23}} \cdot \overline{Y_{24}} \cdot \overline{Y_{25}} .$$

where,

$Y_0 = C_{\text{in}} \cdot \overline{A_0} \cdot \overline{B_0}; Y_1 = \overline{C_{\text{in}}} \cdot A_0 \cdot \overline{B_0}$

$Y_2 = \overline{C_{\text{in}}} \cdot \overline{A_0} \cdot B_0; Y_3 = C_{\text{in}} \cdot A_0 \cdot B_0$

$Y_4 = \overline{K_2} \cdot \overline{K_0} \cdot C_{\text{in}} \cdot B_0; Y_5 = \overline{K_2} \cdot \overline{K_0} \cdot C_{\text{in}} \cdot A_0$

$Y_6 = \overline{K_2} \cdot \overline{K_0} \cdot A_0 \cdot B_0; Y_7 = K_0 \cdot \overline{A_0} \cdot \overline{B_0}$

$Y_8 = K_0 \cdot \overline{C_{\text{in}}} \cdot \overline{B_0}; Y_9 = K_0 \cdot \overline{C_{\text{in}}} \cdot \overline{A_0}$

$Y_{10} = \overline{K_1} \cdot K_0 \cdot A_0 \cdot B_0; Y_{13} = K_1 \cdot \overline{C_{\text{in}}} \cdot \overline{B_0}$

$Y_{14} = K_1 \cdot \overline{C_{\text{in}}} \cdot \overline{A_0}; Y_{15} = K_1 \cdot \overline{A_0} \cdot \overline{B_0}$

$Y_{16} = K_1 \cdot \overline{K_0}; Y_{17} = K_1 \cdot K_0 \cdot C_{\text{in}} \cdot B_0$

$Y_{18} = K_1 \cdot K_0 \cdot C_{\text{in}} \cdot A_0; Y_{19} = K_1 \cdot K_0 \cdot A_0 \cdot B_0$

$Y_{20} = K_2 \cdot \overline{C_{\text{in}}} \cdot \overline{B_0}; Y_{21} = K_2 \cdot \overline{C_{\text{in}}} \cdot \overline{A_0}$

$Y_{22} = K_2 \cdot \overline{A_0} \cdot \overline{B_0}; Y_{23} = K_2 \cdot C_{\text{in}} \cdot B_0$

$Y_{24} = K_2 \cdot C_{\text{in}} \cdot A_0; Y_{25} = K_2 \cdot A_0 \cdot B_0 ;\ Y_{26} = K_3$

4. RESULTS AND COMPARISONS

To fully evaluate the efficiency and reliability of the proposed BCD adder, the simulation was conducted using two separate Electronic Design Automation (EDA) tools: Tanner EDA and Cadence Virtuoso.

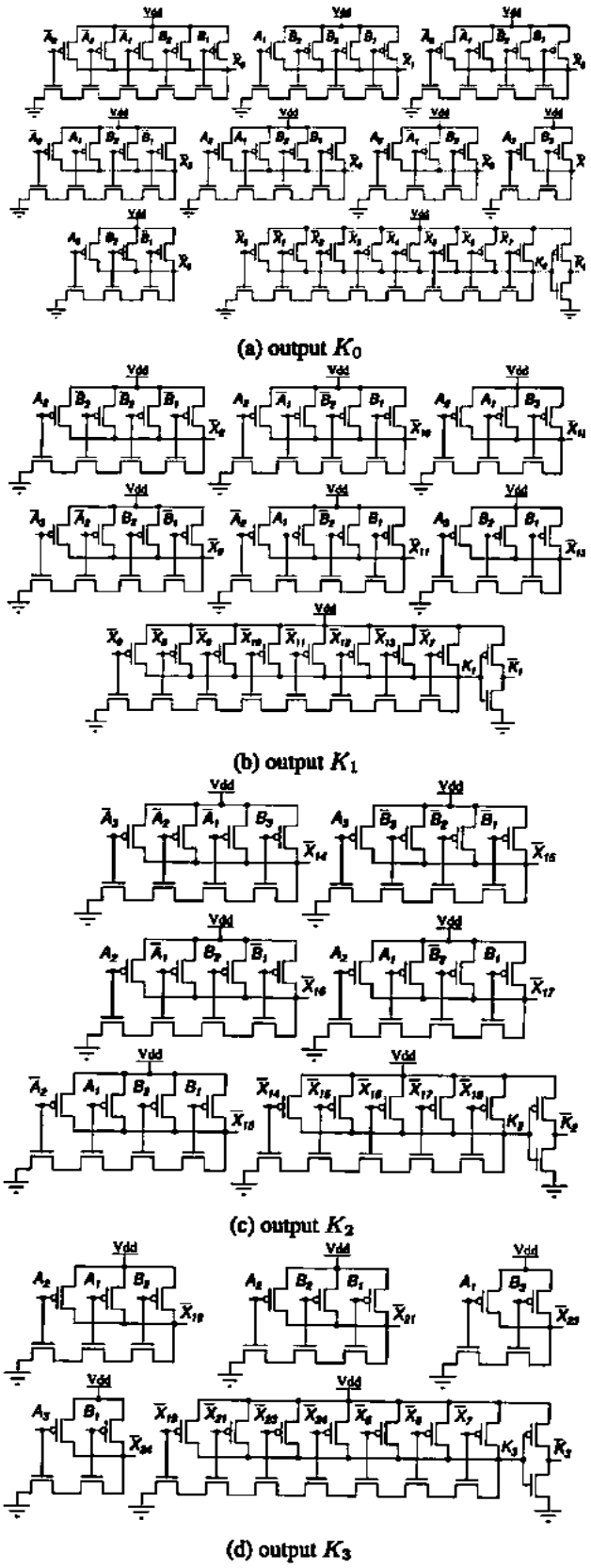

Fig.1. Netlist1 of proposed BCD Adder

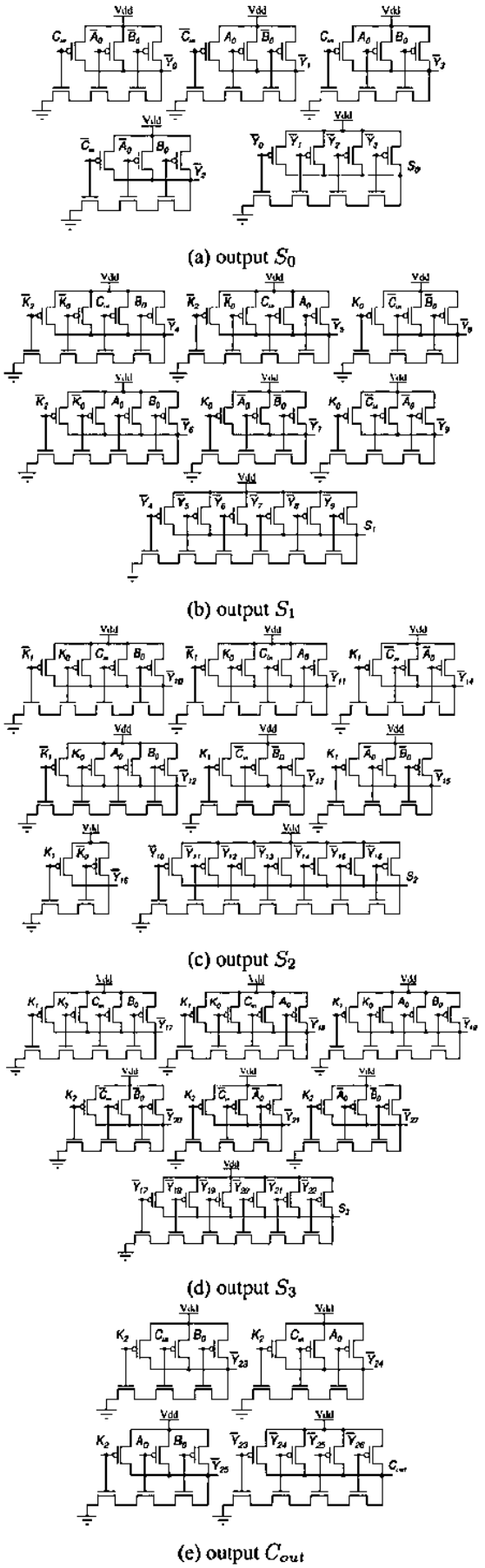

Fig.2. Netlist2 of proposed BCD Adder

These devices were selected to provide both comparative perspective and cross-convenience of results. While Tanner EDA facilitates quick assessment of power consumption in various technology libraries, the table of enabled time behavior and accurate analysis of energy matrix. This dual-tool approach ensured the widespread verification of performance, especially focused on the delay and in power efficiency and dissemination in realistic operating conditions [18-21].

Unlike the approach taken by, who used LTSPICE to test their BCD additive design [15] in many CMOS process nodes (45nm, 65nm, and 180nm), our work only focused on simulation using 90nm CMOS technology in Cadence and 45nm, 65nm and 90nm in Tanner within the table environment. To ensure continuity and enable meaningful comparisons, we adopted the same transistor dimensions reported in their study- especially, PMOS width of 540nm and NMOS width of 270nm for 90nm technology. This alignment in the device size provided a reliable basis to evaluate switching behavior and power efficiency under similar design conditions [22-26].

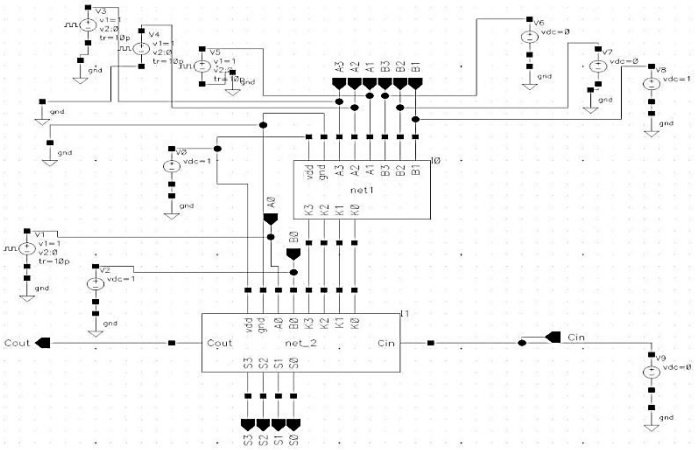

Fig.3. 1-digit BCD Adder in Cadence

The simulation performed in the table produced a major matrix including transient response time and average power use. Using the 90nm technology node, the design showed its suitability for the demand for digital arithmetic operations with the minimum power consumption, showing strong time efficiency [27-28]. This performance was largely attributed to the physical layout, accurate parasitic modeling and accurate control over the careful transistor size, all of which increased the loyalty and effectiveness of the simulation results [27].

Table.1. Power in Tanner for 45nm and 65nm

BCD Adder	Power (uW)	Power (uW)
1-Digit	87	121
2-Digit	236	294
3-Digit	499	712
4-Digit	572.75	763

The Tanner EDA was also employed to evaluate the proposed additives in three separate CMOS technology nodes 45nm, 65nm and 90nm. Unlike the rhythm, these simulations were held using similar dimensions for both PMOS and NMOS transistors along the same width and length. The median in the tanner was mainly

on measuring the average power consumption. While it strengthens the simulation process and provides valuable basic data, it does not offer possible tuning levels through transistor shape adapted as implemented in the atmosphere of the Table.1.

The delays and the average powers [29-30] calculated in the cadence virtuoso for 1-digit, 2-digit, 3-digit, 4-digit BCD adders, which are mentioned in Table.2-Table.5 respectively. The schematic structure of 1-Digit BCD Adder implementation by using cadence virtuoso is shown in Fig.3. Simulation wave forms of 1-digit BCD adder shown in Fig.4.

Table.2. Delay for all possible critical paths in 1-digit BCD Adder

Path	Delay
A_0 to S_3	146.7ps
A_1 to S_3	277.0ps
A_2 to S_3	730.6ps
A_3 to S_3	1.347ns
Max delay to S_3	1.347ns
A_0 to C_{out}	676.0ps
A_1 to C_{out}	700.2ps
A_2 to C_{out}	700.5ps
A_3 to C_{out}	230.6ps
Max delay to C_{out}	700.5ps

Table.3. Delay for all possible critical paths in 2-digit BCD Adder

Path	Delay
A_0 to S_{13}	382.7ps
A_1 to S_{13}	756.1ps
A_2 to S_{13}	745.5ps
A_3 to S_{13}	1.361ns
Max delay to S_{13}	1.361ns
A_0 to C_{out}	123.5ps
A_1 to C_{out}	112.8ps
A_2 to C_{out}	674.4ps
A_3 to C_{out}	250.1ps
Max delay to C_{out}	674.4ps

Table.4. Delay for all possible critical paths in 3-digit BCD Adder

Path	Delay
A_0 to S_{23}	384.8ps
A_1 to S_{23}	758.1ps
A_2 to S_{23}	1.37ns
A_3 to S_{23}	1.36ns
Max delay to S_{23}	1.37 ns
A_0 to C_{out}	697.3ps
A_1 to C_{out}	697.0ps
A_2 to C_{out}	1.31ns
A_3 to C_{out}	1.30ns
Max delay to C_{out}	1.31ns

Table.5. Delay for all possible critical paths in 4-digit BCD Adder

Path	Delay
A_0 to S_{33}	772.537ps
A_1 to S_{33}	772.56ps
A_2 to S_{33}	774.12ps
A_3 to S_{33}	762.41ps
Max delay to S_{33}	774.12ps
A0 to C_{out}	698.5ps
A1 to C_{out}	699.4ps
A2 to C_{out}	1.3ns
A3 to C_{out}	1.302ns
Max delay to C_{out}	1.3ns

Table.6. Power and Delay for BCD Adder for all lengths by using Cadence

BCD Adder	Power (uW)	Delay(ns)
1-Digit	80.38	1.347
2-Digit	140	1.361
3-Digit	235	1.37
4-Digit	344.7	1.302

A comparison of simulation outcomes at the 90nm technology node revealed that the average power consumption recorded in Cadence was noticeably less than that obtained using Tanner. This highlights the greater energy efficiency of the Cadence-implemented design, which can be attributed to its finely tuned transistor sizing and more sophisticated simulation models. Cadence's environment further enabled a detailed assessment of dynamic switching and load characteristics, which played a key role in minimizing overall power usage [31].

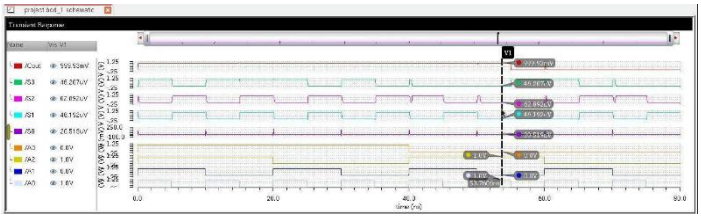

Fig.4. Simulation waveforms of 1-Digit BCD Adder

Table.7. Power comparison in 90nm Technological node

BCD Adder	Power (uW) (Cadence EDA Tool)	Power (uW) (Tanner EDA Tool)
1-Digit	80.38	120
2-Digit	140	290
3-Digit	235	703
4-Digit	344.7	806

5. CONCLUSION

The study introduces high-speed, BCD additives of single-ankles depending on a dual-mining architecture, where correction logic is originally embedded within a computational framework. Logic equations were obtained and applied carefully using CMOS technology on the 90nm scale. It is designed for a single-unknown unit module, which allows straightforward expansion to the multi-digit BCD joint using a ripple-carry configuration, preserving low power characteristics. To evaluate the performance of the design, the adaptor was simulated into the chromosome of the pool for 1-conductive through a 4-conductive input. Additionally, alternative implementation was tested using Tanner EDA in 45nm and 65nm nodes for performance benchmarking. Near all configurations, the proposed design consistently demonstrated better energy efficiency when compared with the existing solutions manufactured in the pool platform. By embedding direct correction functionality in the logic tract, the requirement of separate improvement circuits is eliminated - a significant reduction in low transition activity and a significant decrease in overall power usage.

REFERENCES

[1] M. Vestias and H. Neto, "Improving the Area of Fast Parallel Decimal Multipliers", *Microprocessors and Microsystems*, Vol. 61, pp. 96-107, 2018.

[2] H. Thapliyal and N. Ranganathan, "Design of Efficient Reversible Logic-based Binary and BCD Adder Circuits", *ACM Journal on Emerging Technologies in Computing Systems*, Vol. 9, No. 3, pp. 1-31, 2013.

[3] A. Al Share, "Design of High-Speed BCD Adder using CMOS Technology", *IEEE Access*, Vol. 11, pp. 141628-141639, 2023.

[4] L. Dadda, "Multioperand Parallel Decimal Adder: A Mixed Binary and BCD Approach", *IEEE Transactions on Computers*, Vol. 56, No. 10, pp. 1320-1328, 2007.

[5] H. Calderon, G. Gaydadjiev and S. Vassiliadis, "Reconfigurable Universal Adder", *Proceedings of the International Conference on Application-Specific Systems, Architectures and Processors*, pp. 186-191, 2007.

[6] S. Siddha, "Area Efficient 4-Input Decimal Adder using CSA and CLA", *International Journal of Scientific Engineering and Technology*, Vol. 2, No. 6, pp. 550-553, 2013.

[7] M.F. Cowlishaw, "Decimal Floating-Point: Algorism for Computers", *Proceedings of the International Symposium on Computer Arithmetic*, pp. 104-111, 2003.

[8] Y.D. Ykuntam and S.H. Prasad, "A Modified High Speed and Less Area BCD Adder Architecture using Mirror Adder", *Proceedings of the International Conference on Smart Electronics and Communication*, pp. 624-627, 2021.

[9] M. Cornea, "IEEE 754R Decimal Floating-Point Arithmetic: Reliable and Efficient Implementation for Intel Architecture Platforms", *Intel Technology Journal*, Vol. 11, No. 1, pp. 10-19, 2007.

[10] F.Y. Busaba, C.A. Krygowski, W.H. Li, E.M. Schwarz and S.R. Carlough, "The IBM z900 Decimal Arithmetic Unit", *Proceedings of the International Conference on Signals, Systems and Computers*, pp. 1335-1339, 2001.

[11] C.F. Webb, "IBM z10: The Next-Generation Mainframe Microprocessor", *IEEE Micro*, Vol. 28, No. 2, pp. 19-29, 2008.

[12] M.D. Al-Khaleel, M.J. Gander and A.E. Ruehli, "Optimized Waveform Relaxation Solution of RLCG Transmission Line Type Circuits", *Proceedings of the International Conference on Innovations in Information Technology*, pp. 136-140, 2013.

[13] M. Al-Khaleel and S.L. Wu, ''A Mathematical Analysis of Discrete Waveform Relaxation Algorithms for Transmission Line Type Circuits", *Proceedings of the International Conference on Computer Science and Data Engineering*, pp. 1-5, 2021.

[14] M. Riaz-ul-haque, M. Shintani and M. Inoue, ''Decimal Multiplication using Combination of Software and Hardware", *Proceedings of the International Conference on Circuits and Systems*, pp. 239-242, 2018.

[15] O. Al-Khaleel, M. Al-Khaleel, Z. Al-Qudah, C.A. Papachristou, K. Mhaidat and F.G. Wolff, "Fast Binary/Decimal Adder/Subtractor with a Novel Correction-Free BCD Addition", *Proceedings of the International Conference on Electronics, Circuits and Systems*, pp. 455-459, 2011.

[16] O. Al-Khaleel, Z. Al-Qudah, M. Al-Khaleel, R. Bani-Hani, C. Papachristou and F. Wolff, "Efficient Hardware Implementations of Binary-to-BCD Conversion Schemes for Decimal Multiplication", *Journal of Circuits, Systems and Computers*, Vol. 24, No. 2, 2015.

[17] B. Shirazi, D.Y.Y. Yun and C.N. Zhang, "RBCD: Redundant Binary Coded Decimal Adder", *IEE Proceedings E (Computers and Digital Techniques)*, Vol. 136, No. 2, pp. 1-9, 1989.

[18] A.A. Bayrakci and A. Akkas, "Reduced Delay BCD Adder", *Proceedings of the International Conference on Application-Specific Systems, Architectures and Processors*, pp. 266-271, 2007.

[19] S.A. Kumar and M. Kavitha, "Design of Efficient BCD Adders Correction Logic in QCA Technology", *International Journal of Scientific Research and Engineering Trends*, Vol. 8, No. 2, pp. 869-874, 2022.

[20] I.T. Abougindia, I. Cevik, S.U. Ay and F.N. Zghoul, "A Fast Two-Step Coarse-Fine Calibration (CFC) Technique for Precision Comparator Design", *Proceedings of the International Conference on Electronics, Circuits and Systems*, pp. 153-156, 2013.

[21] V. Sukumar, M. Alahmad, K. Buck, H. Hess, H. Li, D. Cox, F.N. Zghoul, J. Jackson, S. Terry, B. Blalock, M.M. Mojarradi, W.C. West and J.F. Whitacre, "Switch Array System for Thin Film Lithium Microbatteries", *Journal of Power Sources*, Vol. 136, No. 2, pp. 401-407, 2004.

[22] F.N. Zghoul, H. Alteehi and A. Abuelrub, "A Mayfly-based Approach for CMOS Inverter Design with Symmetrical Switching", *Algorithms*, Vol. 16, No. 5, pp. 1-21, 2023.

[23] F.N. Zghoul, S.U. Ay and A. Ababneh, "Gain and Offset Analysis of Comparator using the Bisection Theorem and a Balanced Method", *International Journal on Electronics*, Vol. 103, No. 12, pp. 1965-1983, 2016.

[24] M. Srinivas and K.V.D. Sagar, "Analysis on Power Gating Circuits based Low Power VLSI Circuits (BCD Adder)",

Journal of Physics: Conference Series, Vol. 2089, No. 1, pp. 1-14, 2021.

[25] I. Gassoumi, L. Touil and A. Mtibaa, "Design of Efficient Binary-Coded Decimal Adder in QCA Technology with a Regular Clocking Scheme", *Computers and Electrical Engineering*, Vol. 101, pp. 1-7, 2022.

[26] P. Durgaprasadarao and K.V.D. Sagar "Implementation of Electronic Devices of CMOS Full Adder in Low Power VLSI Circuits", *Journal of Physics: Conference Series*, Vol. 2089, No. 1, pp. 1-8, 2021.

[27] M.S. Schmookler and A. Weinberger, "High Speed Decimal Addition", *IEEE Transactions on Computers*, Vol. 20, No. 8, pp. 862-866, 1971.

[28] A. Vazquez and E. Antelo, "Conditional Speculative Decimal Addition", *Proceedings of the International Conference on Real Numbers and Computers (RNC)*, pp. 47-57, 2016.

[29] R. Santhiya and T. ThamaraiManalan, "Power Gating based Low Power 32-Bit BCD Adder using DVT", *International Journal for Scientific Research and Development*, Vol. 3, No. 2, pp. 802-805, 2015.

[30] D. Saha, S. Basak, S. Mukherjee and C.K. Sarkar, "A Low-Voltage, Low Power 4-Bit BCD Adder, Designed using the Clock Gated Power gating and the DVT Scheme", *Proceedings of the International Conference on Signal Processing, Computing and Control*, pp. 1-6, 2013.

[31] D. Siddhamshittiwar, "An Efficient Power Optimized 32-Bit BCD Adder using Multi-Channel Technique", *International Journal of New Practices in Management and Engineering*, Vol. 6, No. 2, pp. 7-12, 2017.