

Implementation of High Speed BCD Adder Designs Based on Majority Gates and Three-Input XOR

Latha Sri Vijjagiri¹, G.Dhanalakshmi², S.Gopalakrishnan³, S. Nafiza⁴

¹ Research Scholar, Siddhartha Institute of Technology and Sciences Koremulla Road,
Narapally, R.R. District, Telangana.

² Professor, Siddhartha Institute of Technology and Sciences Koremulla Road,
Narapally, R.R. District, Telangana.

^{3,4} Associate Professor, Siddhartha Institute of Technology and Sciences Koremulla Road,
Narapally, R.R. District, Telangana.

ABSTRACT

Majority-of-three (MAJ) gates may be realized natively by a variety of new ways. Although numerous MAJ gates can utilize upgraded gates there is directly derived from the physical qualities of new approaches, such as exclusive-OR (XOR) gates, is a very efficient XOR architecture. We covered three-input exclusive-OR multi-digit binary coded decimal (BCD) adder concepts (XOR3), square root carry choose binary adder, and MAJ gates in this short. Financial, commercial, and industrial computers use BCD adders. Using QCA technology, the designs were realized. The suggested logic representations achieve high speed, and the proposed architecture was generated and simulated using Verilog HDL in Xilinx ise.

Index terms: *QCA technology, majority gate, 3input xor gate and BCD architecture.*

I. INTRODUCTION

High-performance digital circuits might be built in the future using QUANTUM-DOT cellular automata (QCA) technology. The sub modules for arithmetic are among the most studied architectures among the logic circuits lately introduced. Non-elementary digital modules are cleverly built by mixing In the QCA technology, only inverters and majority gates are biologically accessible (MGs). Despite logic function. Applications in business, finance, and the Internet need to be more accurate. Decimal arithmetic has gotten a lot of attention lately.

The mistakes resulting from the translation in these circumstances, the distinction between binary and decimal

number representations could no longer be maintained, As a result, certain modern microprocessor cores have hardware decimal arithmetic units, which are compliant with the IEEE 754–2008 standard. The cellular automata (QCA) technology developed by QUANTUM-DOT may one day be used to construct digital circuits with exceptional performance.

In this study, we provide a novel approach to designing QCA-based n-digit (with n 1) BCD adders, which improve upon its forerunners in both computational efficiency and area/cell use. Some of the advantages are shown in Figure 1. The BCD adders' structure is shown in. accomplished by combining a novel logic technique with specially built QCA modules. Optimization

of the latter takes into account the fact When a carry is formed by summing the operands' The most important sum digit is found by starting with the least important digits, moving them to the next n 2-digit locations, and then absorbing them at the end.

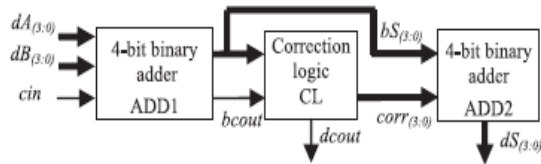


Fig. 1. Structure of the BCD adders.

II. LITERATURE SURVEY

"Cost-efficient decimal adder design in quantum-dot cellular automata," by W. Liu, Swartzlander, and M. O'Neill. Integrated Circuit (IC) Technology is evolving at a rapid pace to improve circuit performance and densify small systems. For the last four decades, conventional CMOS technology has played a critical role in digital processing. However, scaling CMOS devices has posed some difficulties in recent years.

QCA is a unique Nano electronic technology that has been recognized as a prospective Nano electronic technology for overcoming the constraints of the transistor-based CMOS technology. This has led to its identification as a promising Nano electronic technology. An effective architecture for a BCD adder in QCA technology is shown here. This architecture makes use of an area-optimized QCA full adder to provide a new concept of IC design in a way that is both effective and optimal.

The suggested BCD adder circuit optimizes cost function, layout space, and QCA cells, which results in an improvement over past designs of BCD adder circuits. An Architecture for Quantum Dot-Based

Computing Devices by C. S. Lent and P. D. Tougaw. Using quantum dots and their ability to interact, a new kind of cellular automaton has been developed (QCA). We demonstrate how quantum-dot cell arrays may be utilized to conduct important calculations. A novel adiabatic switching paradigm is created that allows for timed control, removes metastability issues, and allows for a pipelined design.

III. DESIGN OF HIGH SPEED VLSI ARCHITECTURE FOR BCD ADDER USING QCA MAJORITY GATES

Since CMOS technology is reaching its physical limit, new computer paradigms are being applied to increase processing efficiency. New techniques emerge as a consequence of either increasing the performance of current devices or developing whole new devices using novel materials. Some of the things being looked at include quantum-dot cellular automata (QCA), nanomagnet logic, spin-wave devices, and two-dimensional polarity-controllable transistors.

These developing approaches, unlike traditional CMOS Boolean logic, may intrinsically implement majority-of-three (MAJ) basic gates. $M(x, y, z)$ Only if two of the three arguments are true does this method return true. Disjunctive and conjunctive normal forms may both be used to express it:

$$\begin{aligned} M(x, y, z) &= xy + xz + yz \\ &= (x + y)(x + z)(y + z). \end{aligned} \quad (1)$$

Because $xy = M(x, y, 0)$ and $x + y = M(x, y, 1)$. In addition to MAJ gates, emerging methods might be utilized to demonstrate compact XOR and XNOR gates. It's important to remember that

XOR/XNOR gates in CMOS need complicated designs or a lot of gates. These gates, on the other hand, may be built with fewer components using new approaches. According to recent research, QCA designs with upgraded gates (XOR/XNOR) outperform designs with just native MAJ gates.

As a result, it's critical to take advantage of MAJ and XOR/XNOR gate-based arithmetic circuit designs. Decimal arithmetic makes use of a binary coded decimal (BCD) adder. Numerous studies have been conducted on majority logic-based BCD adders. Computational architectures and logical representations based on Boolean function adders are common. Carry-propagation structures for BCD adders with many digits are where RCAs shine, but they're simple and inefficient. Carry-flow and carry-lookahead computing architectures are presented as potential solutions to this problem.

The physical layout's design complexity is proportional to the logic representations of BCD adders. Rewriting Boolean functions, for example, to fully use the MAJ and reduce the use of constant inputs, may result in considerable area/delay gains. The literature, on the other hand, has paid little attention to BCD adder designs with increased gates. According to this research,

The main improvement is a hybrid logic architecture for a three-input, n-digit (n-1) BCD adder that uses XOR3 and MAJ basic gates. Because $X(x, y, 0) = xy$ and $X(x, y, 1) = x \bar{y}$, by making any one input constant, the three-input XOR3 function $X(x, y, z) = x \bar{y}z$ may be converted to a two-

input XOR/XNOR. $X(x, y, z) = (x \bar{y})z = (x \bar{y}) \bar{z}$ As a result, the suggested designs may be advantageous to technologies that favour

MAJ and XOR3 (XOR/XNOR) primitives. Using the ADDER BCD ADDER Figure 1(a) depicts the RCA-BCD adder, which consists of two 4-bit binary adders (ADD1 and ADD2) and a correction logic (CL) circuit. The ADD1 adds $dA_{3:0}$, $dB_{3:0}$, and dC_{in} to make $bS_{3:0}$, the binary sum, and bC_{out} , the output carry. If $bS_{3:0} = 10$,

When the BCD total is converted to a positive integer using the CL circuit, the result is shown as a CL. It makes the $cL_{3:0}$ and dC_{out} carry signals for the decimal output. If $dC_{out} = 1$, the $cL_{3:0}$ is $(0110)_2$, else it is $(0000)_2$. To get the decimal total $dS_{3:0}$, the ADD2 adds $bS_{3:0}$ and $cL_{3:0}$. A carry-look ahead structure is presented in for computing the output carries, as illustrated in Fig. 1(b), and it was used in multi-digit BCD adder designs with parallel implementation. In addition, a new structure was added to the CL circuit, with dC_{in} functioning as an extra input. As a consequence, the dC_{out} latency for multi-digit PBA-BCD adders has been greatly decreased.

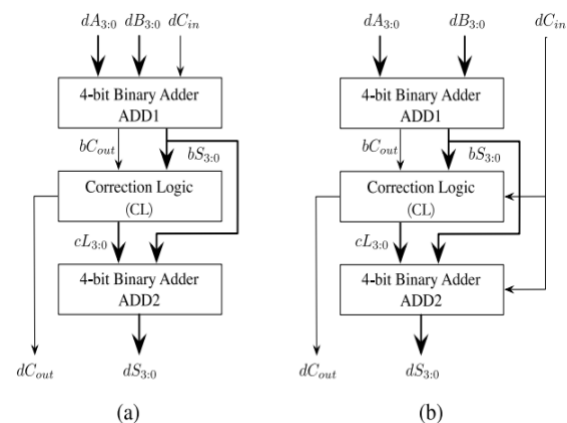


Fig. 2. (a) Standard design and

(b) Recommended design of a BCD adder.

In this project, the XOR-Majority Graph (XMG) logic model, which was created in, is explained first. One-digit and multi-digit BCD adders based on RCA and PBA architectures are shown in turn. XMGs are a form of XML document. The primitives of the XMG are the MAJ, the XOR with two inputs, and the inverter. In XMGs, instead of two-input XORs, we employ XOR3s in our designs. To distinguish the XMGs,

The three MAJ and XOR inputs might be recast as a new extended XMG labelled XMG. It is possible to reduce the total number of XOR nodes in XMG from three to two by changing the z in $(x-y-z)$ to $(x-y)$. Because inverters are expensive in developing approaches, the introduction of XOR3 opens up new possibilities for inversion optimization. Because two-input XOR expressions, as demonstrated in (2), can only realise one inverter, two-input XOR expressions can only realise one inverter.

$$\begin{aligned}\bar{x} \oplus \bar{y} &= x \oplus y \\ \overline{x \oplus y} &= x \oplus \bar{y} = \bar{x} \oplus y \\ &= 0 \oplus \bar{x} \oplus y = 1 \oplus x \oplus y\end{aligned}\quad (2)$$

By rewriting the equation as an XOR3, the inversion is reduced to constant input, resulting in a low-cost implementation.

Converter from RCA to BCD For the 1-bit complete adder with a , b , and c_{in} binary inputs, adder S and c_{out} are defined as follows.

$$\begin{aligned}s &= a \oplus b \oplus c_{in} = \mathbb{X}(a, b, c_{in}), \\ c_{out} &= \mathbb{M}(a, b, c_{in})\end{aligned}\quad (3)$$

To describe a 1-bit complete adder, we just need two operations: one MAJ and

one XOR3. For logic representation. An example of an ADD1 circuit that uses four MAJ and four XOR3 operations may be seen in the (4).

$$\begin{aligned}G_0 &= \mathbb{X}(dA_0, dB_0, dC_{in}) \quad H_0 = \mathbb{M}(dA_0, dB_0, dC_{in}) \\ G_1 &= \mathbb{X}(dA_1, dB_1, H_0) \quad H_1 = \mathbb{M}(dA_1, dB_1, H_0) \\ G_2 &= \mathbb{X}(dA_2, dB_2, H_1) \quad H_2 = \mathbb{M}(dA_2, dB_2, H_1) \\ G_3 &= \mathbb{X}(dA_3, dB_3, H_2) \quad bC_{out} = \mathbb{M}(dA_3, dB_3, H_2)\end{aligned}\quad (4)$$

The suggested decimal carry out dC_{out} is written as

$$dC_{out} = \mathbb{M}(bC_{out}, G_3 + bC_{out}, \mathbb{M}(G_1, G_2, G_3)).\quad (5)$$

Expansion and simulation led to the identification $(\mathbb{M}(a, b, c), c + d, d) = \mathbb{M}(a + b, c + d, d)$ is valid, As a result, we may rewrite (5) as

$$dC_{out} = \mathbb{M}(bC_{out}, G_3 + bC_{out}, G_1 + G_2).\quad (6)$$

Because the ADD2 section adds $bS3:0$ and $cL3:0$ to get the decimal total $dS3:0$, we may represent it in bits as $G3G2G1G0 + 0dC_{out}dC_{out}0 = dS3dS2dS1dS0$, which necessitates another 4-bit adder. The following are the expressions:

$$\begin{aligned}dS_0 &= G_0 \\ dS_1 &= \mathbb{X}(G_1, dC_{out}, 0) \\ dS_2 &= \mathbb{X}(G_2, \mathbb{M}(dC_{out}, \bar{G}_1, 0), 0) \\ dS_3 &= \mathbb{X}(G_3, \mathbb{M}(dC_{out}, \mathbb{M}(G_1, G_2, 1), 0), 0)\end{aligned}\quad (7)$$

Note that in $dS3$, $\mathbb{M}(G1, G2, 1) = G1 + G2$ appears in (6) but not in (5), indicating that we may reuse this signal to save area.

PBA-BCD Adder

Despite the fact that the proposed BCD adder enlarges the target area, it does not significantly improve latency. Carry signals must be generated for a high-performance multidigit BCD adder design. dC_{out} of the RCA-BCD adder has a lengthy critical path, and dC_{in} from the preceding digit's dC_{out} determines the delay for the subsequent digit. In the CL section, a MAJ-based circuit is utilised to calculate dC_{out} by

taking into account the input dC_{in} . The PBA-BCD adder is supported by this design type. Furthermore, the $dC[3:0]$ signal was derived by subtracting dC_{out} from dC_{out} .

$$dC_{[3:0]} = [0, dC_{out}, dC_{out}, 0]. \quad (8)$$

The BCD adder structure in Fig. 1 is made using the same CL (b). XMG representations were used to redesign the last two 4-bit binary adders, ADD1 and ADD2. The ADD1 logic formulae are provided in (9), with dC_{in} eliminated.

$$\begin{aligned} G_0 &= \mathbb{X}(dA_0, dB_0, 0) \quad H_0 = \mathbb{M}(dA_0, dB_0, 0) \\ G_1 &= \mathbb{X}(dA_1, dB_1, H_0) \quad H_1 = \mathbb{M}(dA_1, dB_1, H_0) \\ G_2 &= \mathbb{X}(dA_2, dB_2, H_1) \quad H_2 = \mathbb{M}(dA_2, dB_2, H_1) \\ G_3 &= \mathbb{X}(dA_3, dB_3, H_2) \quad bC_{out} = \mathbb{M}(dA_3, dB_3, H_2) \end{aligned} \quad (9)$$

The mechanics of producing the dC_{out} signal for the CL circuit may be found in.

The phrases are rewritten as follows:

$$K_1 = \mathbb{M}(bC_{out}, \mathbb{M}(1, bC_{out}, G_3), \mathbb{M}(G_1, G_2, G_3))$$

$$K_2 = \mathbb{M}(1, K_1, \mathbb{M}(0, G_0, G_3))$$

$$dC_{out} = \mathbb{M}(dC_{in}, K_1, K_2) \quad (10)$$

The dC_{in} signal is incorporated in the logic formulas for the ADD2 section as:

$$\begin{aligned} dS_0 &= \mathbb{X}(G_0, dC_{in}, 0) \\ dS_1 &= \mathbb{X}(G_1, dC_{out}, \mathbb{M}(0, G_0, dC_{in})) \\ dS_2 &= \mathbb{X}(G_2, dC_{out}, \mathbb{M}(G_1, dC_{out}, \mathbb{M}(0, G_0, dC_{in}))) \\ dS_3 &= \mathbb{X}(G_3, \mathbb{M}(G_2, dC_{out}, \mathbb{M}(G_1, dC_{out}, \\ &\quad \mathbb{M}(0, G_0, dC_{in}))), 0). \end{aligned} \quad (11)$$

PROPOSED BINARY ADDER

After receiving the carry-in (C_{in}) bit, the ripple carry adder creates the carry out bit by rippling the incoming carry produced in the previous step. Since a result, the pace of RCA is reduced, as each stage's C_{out} is reliant on the previous stage's C_{out} or the current stage's C_{in} . The linear dependence issue of C_{out} on C_{in} is solved by assuming that C_{in} has just two potential values: '0' and '1'. These possible C_{in} values are used to

produce the partial Sum (S) and C_{out} in simultaneously.

Using a multiplexer, The actual carry input is used to figure out the final total and carry. Despite the fact that this feature aids in the reduction of computation time, the usage of a redundant adder circuit reduces the area efficiency. Carry Select Adder is made up of two rca blocks. In this research, to reduce the optimal latency, We recommended that you use a square root carry save adder.

Square Root Select Adder is a useful tool for calculating the square root of a number. Square Root Carry Select Adder (SRCSA) allows you to modify the block size. The whole analysis is omitted here for simplicity's sake, but a 16-bit adder may be built using block sizes of 2-2-3-4-5 rather than the uniform block size of four (as done before). When the Full-Adder delay equals the MUX delay, this break-up is ideal. The recommended SRCSA adder for 16 bits is depicted in Fig. 3, where the inputs are A and B, the carry-in is C_{in} , and the outputs are sum (S) and carry-out (C) respectively (C_{out}).

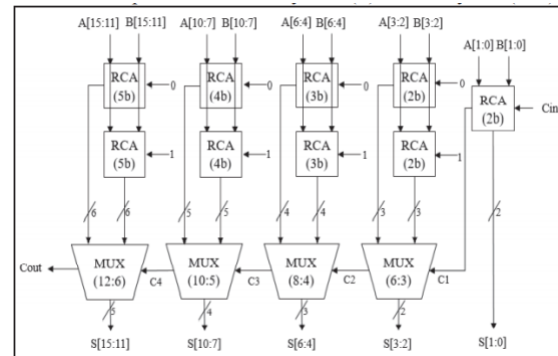


Fig. 3. 16-bit SRCSA

IV. RESULTS

RTL SCHEMATIC: The register transfer level (RTL) schematic denotes the

architecture's blueprint and is used to compare perfect architecture that we must create from the intended architecture. The hdl language is used to transform the architecture's description or summary into the functioning summary using a coding language like verilog or vhdl. The internal connection blocks are even specified in the RTL schematic for easier analysis. Below is a schematic representation of the design's RTL implementation.

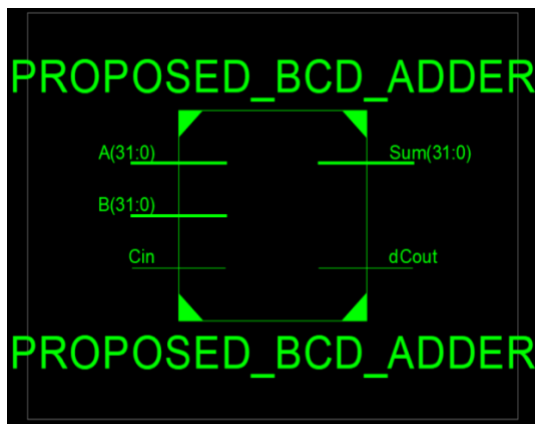


Fig4: Proposed BCD Adder RTL Schematic

TECHNOLOGY SCHEMATIC: With the LUT area parameter being used to estimate architecture design in VLSI, this diagram shows the technology's architecture in LUT format. The FPGA's LUTs, which are square units, represent the code's memory allocation.

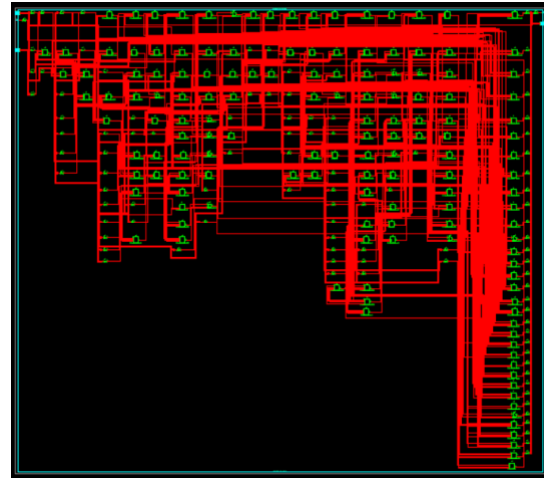


Fig5: View Technology Schematic of proposed BCD adder

SIMULATION: Unlike the schematic, which only verifies the connections and blocks of a circuit, a simulation verifies the circuit's workings. Waveforms are the only form of output that can be viewed in the simulation window because it is launched by shifting from implementation to simulation. Since it can support multiple radix number systems, this is a useful feature.



Fig6: Simulated Waveforms of proposed BCD Adder

PARAMETERS: Consider that in VLSI, the factors considered are area, delay, and power; using these metrics It's possible to make comparisons between several designs. XILINX 14.7 is used to acquire the parameter, while verilog is used as the HDL language.

Parameter	Existed BCD Adder	Proposed BCD Adder
Delay (ns)	70.091	41.530

Table1: parameter comparison

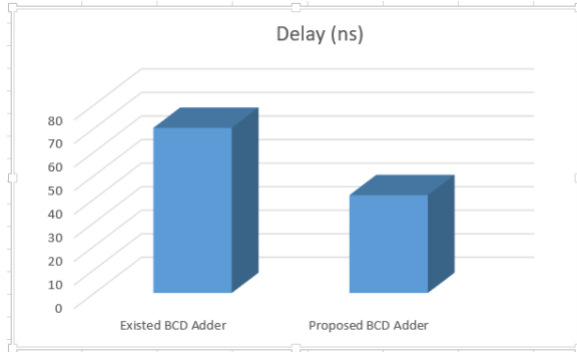


Fig 7: delay comparison bar graph

V. CONCLUSION

In upcoming computer nanotechnologies, majority logic is commonly employed. The implementation of arithmetic functions using majority logic has received a lot of attention. In this short, One and multi-digit BCD adder expressions are created using better logic representations of three-input XOR and majority operations in QCA Designer and we have gotten good delay results. When compared to the greatest designs currently available, our proposed BCD adder has less delay product improvement for the 32-digit scenario.

REFERENCES

[1] W. Liu, L. Lu, E. E. Swartzlander, and M. O'SNeill, "Cost-efficient decimal adder design in quantum-dot cellular automata," in Proc. IEEE Int. Symp. Circuits Syst., Seoul, South Korea, 2012, pp. 1347–1350.

[2] M. F. Cowlshaw, "Decimal floating-point: Algorithm for computers," in Proc. IEEE Symp. Comput. Arithmetic, Santiago de Compostela, Spain, 2003, pp. 104–111.

[3] IEEE Standard for Floating-Point Arithmetic, IEEE Standard 754-2008, pp. 1–70, 2008.

[4] C. S. Lent and P. D. Tougaw, "A device architecture for computing with quantum dots," Proc. IEEE, vol. 85, no. 4, pp. 541–557, Apr. 1997.

[5] G. Csaba, A. Imre, G. H. Bernstein, W. Porod, and V. Metlushko, "Nano computing by field-coupled nanomagnets," IEEE Trans. Nanotechnology., vol. 1, no. 4, pp. 209–213, Dec. 2002.

[6] A. Khitun and K. L. Wang, "Nano scale computational architectures with spin wave bus," Super lattices Microstruct. vol. 38, no. 3, pp. 184–200, 2005.

[7] G. V. Resta et al., "Doping-free complementary logic gates enabled by two-dimensional polarity-controllable transistors," ACS Nano, vol. 12, no. 7, pp. 7039–7047, 2018.

[8] D. E. Knuth, The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1. Boston, MA, USA: Pearson Educ., 2011.

[9] S. Ahmed and S. F. Naz, "Notice of violation of IEEE publication principles: Design of cost efficient modular digital QCA circuits using optimized XOR gate," IEEE Trans. Circuits Syst. II, Exp. Briefs, early access, Oct. 12, 2020, doi: 10.1109/TCSII.2020.3030180.

[10] L. Wang and G. Xie, "A novel XOR/XNOR structure for modular design of QCA circuits," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 67, no. 12, pp. 3327–3331, Dec. 2020.

vol. 37, no. 12, pp. 3031–3041, Dec. 2018.