Implementation of 64-bit BCD Adder using Majority Gates

[1]Prashu Chauhan, [2]Manoj Bandil

[1]M.tech Scholar, [2]Associate Professor
EC, Department ITM
ITM Gwalior, India

***Abstract*: As technology is growing very fast, to cope with technology, advance techniques are used for designing circuits or systems. In today's era low power, high speed and small area are the key parameters of new technique. CMOS technology gives all the mentioned parameters in desired values, still there are some limitations. Quantum-dot cellular automata (QCA) technology provides a promising opportunity to overcome the limits of conventional CMOS technology. This paper explains the design of 64-bit BCD adder using Majority gates. Xilinx 14.5 ISE simulator is used for analysis of this BCD adder.**

***Index Terms*: CMOS, QCA, Majority gates, BCD adder.**

I. INTRODUCTION

Quantum-dot cellular automata (QCA) technology provides a promising opportunity to overcome the limits of conventional CMOS technology. To designing more complicated circuits in future the QCA is getting recognition by designing engineers. In direction of QCA some arithmetic circuits and logical circuit already had been designed such as adder [3]-[4], multipliers[5]–[7],and comparators . Autonomously of the performed logic function, nonelementary digital modules are designed intelligently combining inverters and majority gates (MGs), which are the basic fundamental logic gates inherently available within the QCA technology.

II. BASICS OF QCA(QUANTUM DOT AUTOMATA)

The most fundamental element of the QCA structure is shown in figure 1. It is the building block of the QCA technology. As we can see in the figure it is having four dots at all its corner. The structure has two free electrons. The structure of the QCA in provided in the figure 1.

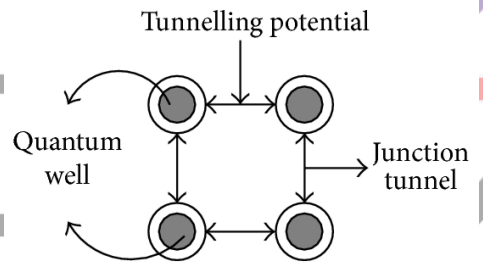


Fig. 1 Structure of a QCA Cell

Because of the coulombic repulsion these two electrons can only be placed in two stable states. The electrons are always resided in diagonally opposite corners of the QCA cell. The diagonally opposite corners are having maximum distance. The stable states also called as polarization. As per the locations of the electrons in the cell two states can be take place. These states are considered as binary states 1 and 0. For the purpose of explanation the figure 2 is shown.

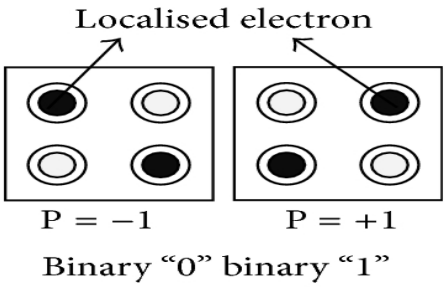


Fig.2 QCA cell with two polarization

III. Proposed BCD Adder

The following relation is taking place for creating majority gate for the implementation of QCA technique in front end tool.

$$M(a,b,c) = a.b + a.c + b.c \quad (3.1)$$

The 1-digit BCD adders previously proposed in Binary arithmetic in Quantum-dot cellular automata (QCA) technology has been extensively investigated in recent years. However, only limited attention has been paid to QCA decimal arithmetic. In this research work, two cost-efficient binary-coded decimal (BCD) adders are presented. One is based on the carry flow adder (CFA) using a conventional correction method. The other uses the carry look ahead (CLA) algorithm which is the first QCA CLA decimal adder proposed to date.

Compared with previous designs, both decimal adders achieve better performance in terms of latency and overall cost. The proposed CFA-based BCD adder has the smallest area with the least number of cells. QCA is attracting a lot of attentions due to its very small sizes and low power consumption. The primary device, a quantum-dot cell, can be used to make gates, wires, and memories as such it is the basic building block of nanotechnology circuits. Because of its innovation, the current literature shows design several circuit structures. This research work proposes a modular design of a BCD adder in QCA and shows analyses based upon those designs A crucial building block for decimal operations is the decimal adder. Thus, it is important to explore its design in future nanotechnologies. This work presents the first decimal adder design in quantum-dot cellular automata (QCA) nanotechnology using majority voting gates. The top-level architecture is in Figure 3.1. In the figure 3.1 the input values are loaded by block "A1" which is responsible for the addition of binary data of 64-bits. After the addition if the result of "A1" is more than the BCD limit the required operations are done by correction logic (CL). "A2" is getting inputs from "CL" and "A1" for its process. The result dS[63:0] of 64 bits is provided by 64 bit BCD adder i.e., "A2" whereas the carry "dcout" is took out by "CL".

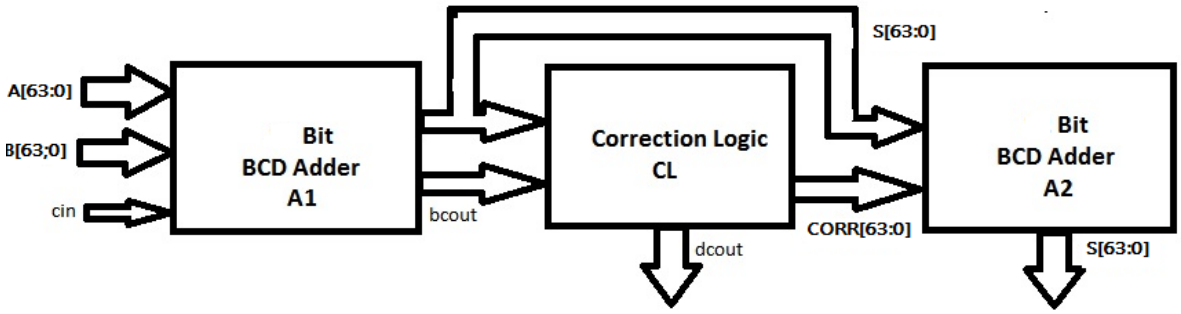


Figure 3 Block diagram of proposed BCD adder

3.1.1 Proposed BCD adder architecture

In the new proposed design the basic and foremost important concept is majority gate. The majority gate is mention by "M" in the below given figure 4. The majority gate is constructed by using logical gates. Combination of AND gate, OR gate and NOT gates are include primarily in it.

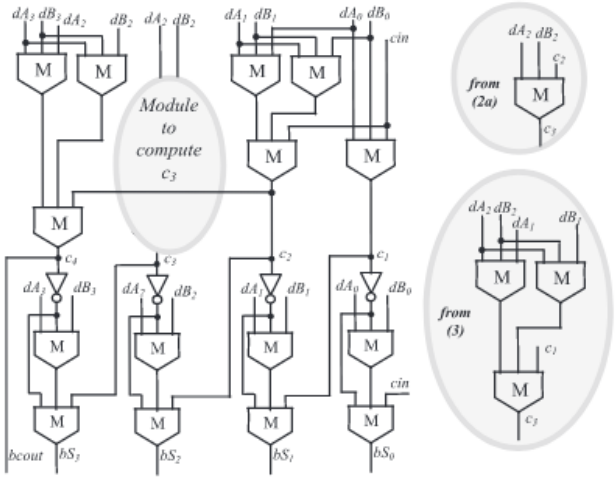


Figure 4 New ADD1 module architecture.

Few more architecture is needed to be considered and which plays an important role in the overall functionality. ADD1 is responsible for the addition of input 64 bits data and the carry which may create in the previous calculation.

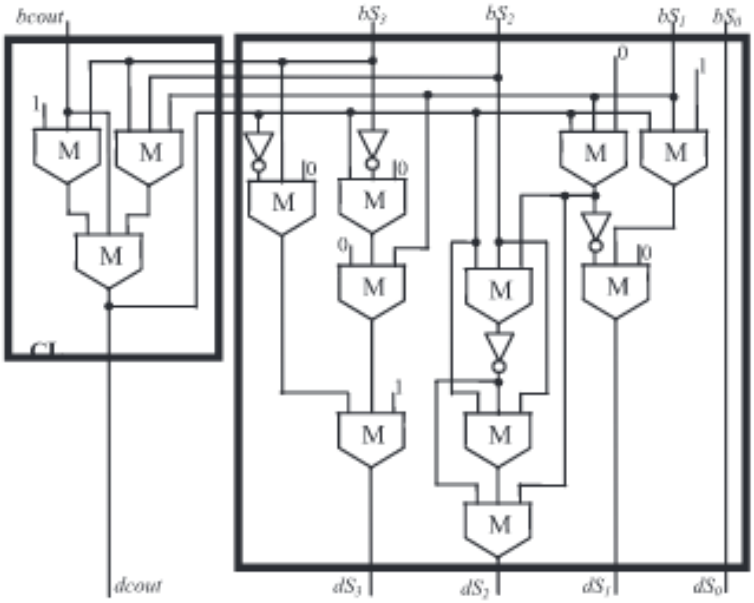


Figure 5 Module CL and ADD2 of the new BCD adder

As mentioned in the above figure that module correction logic (CL) and adder 2 (ADD2) are two more architecture. CL is used here to make the correction if it require when the value is greater than the decimal value 9. In BCD format only decimal which have values up to 9 can be converted and for more than 9 it requires to do some corrections to convert the result in BCD format. The correction needs the few more majority gate which tests that whether the value is more than the maximum limit or not. If the value is more than the limit then it generates the signal and sends it to the next level. The next level is nothing but the adder which is responsible for the addition of binary 6 i.e., 0110. By adding 6 the result again comes under limit of BCD format and provides the correct answer and in BCD format.

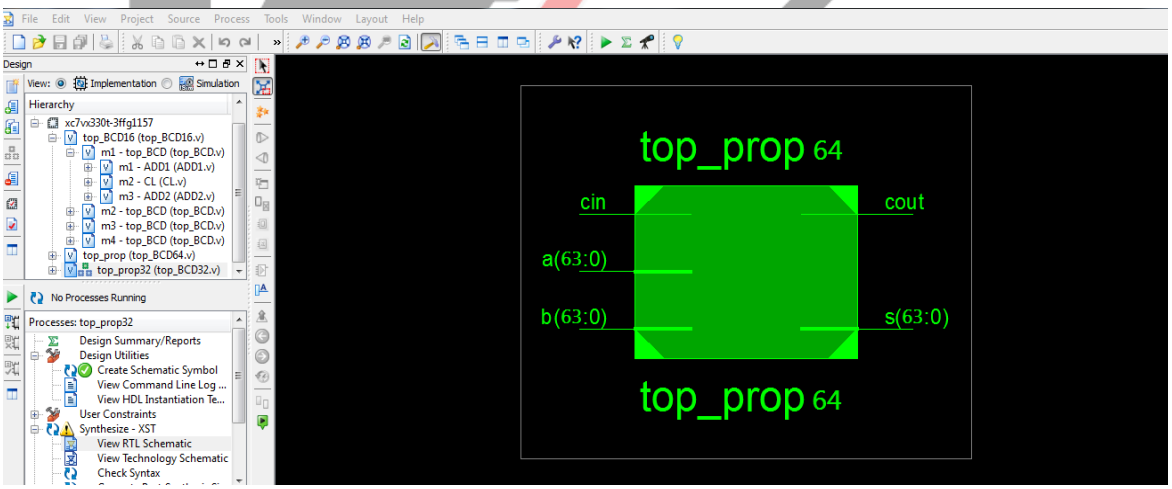
The similar approach can be applied to the adder of more bit size and for other logical or arithmetic operations. As here took the 64 bits for the demonstration purpose the other somewhat more complicated logics can be constructed in it.

IV .SIMULATION RESULTS

SIMULATION RESULTS OF 64 BIT BCD ADDER USING MAJORITY GATE

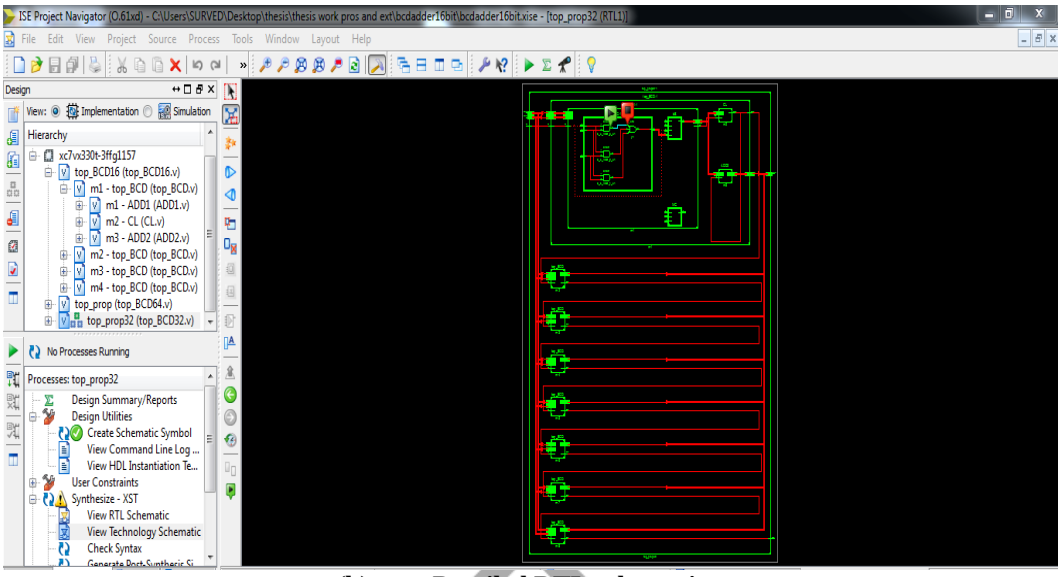
All the BCD adder are coded in the Verilog HDL and implemented on Xilinx tool..

For the application of code in the Xilinx ISE 13.2 the verilog HDL language is used. The output of the code are shown below-



(a) Top level RTL schematic.

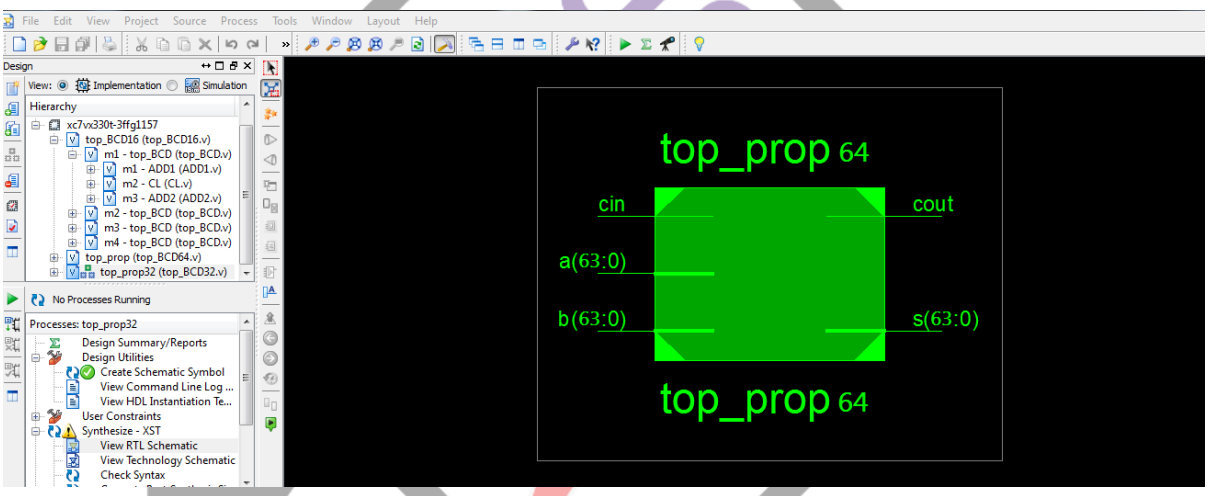
In the above figure the top view of RTL schematic is motioned. The RTL is the abbreviation of Register-Transfer Logic. Register-transfer-level abstraction is used in hardware description languages (HDLs) like Verilog and VHDL to create high-level representations of a circuit, from which lower-level representations and ultimately actual wiring can be derived. Design at the RTL level is typical practice in modern digital design.



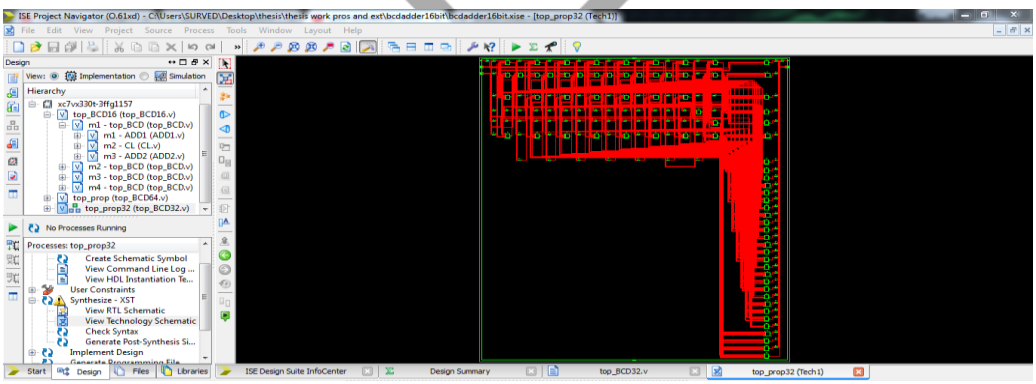
(b) Detailed RTL schematics.

Figure 6 RTL Schematics of 64 bits proposed BCD adder (a) Top level RTL schematics and (b) Detailed RTL schematics

In the detailed schematic all the components are placed so that individual components can be observed. It is basically used to understand the inner structure of the device. Here the use of logic OR gate, AND gate, NOT gate as well as the wires can observed.



(a) Top level Technology schematics



(b) Detailed Technology schematics Figure 7 Technology Schematics of 64 bits proposed BCD adder (a) Top level Technology schematics and (b) Detailed Technology schematics

In the similar manner the technology schematic consist of buffer, multiplexer and LUTs. The element which is used to store the data temporarily is termed as buffer. It is used to move the data from one location to another during different operations. The device that selects one of several input signals and forwards the selected input into a single line is a multiplexer . Look-up tables are tools

for function generation in CLBs. Four independent inputs are provided to each of two function generators (F1-F4 and G1-G4). These function generators can implement any arbitrarily defined Boolean function of four inputs.

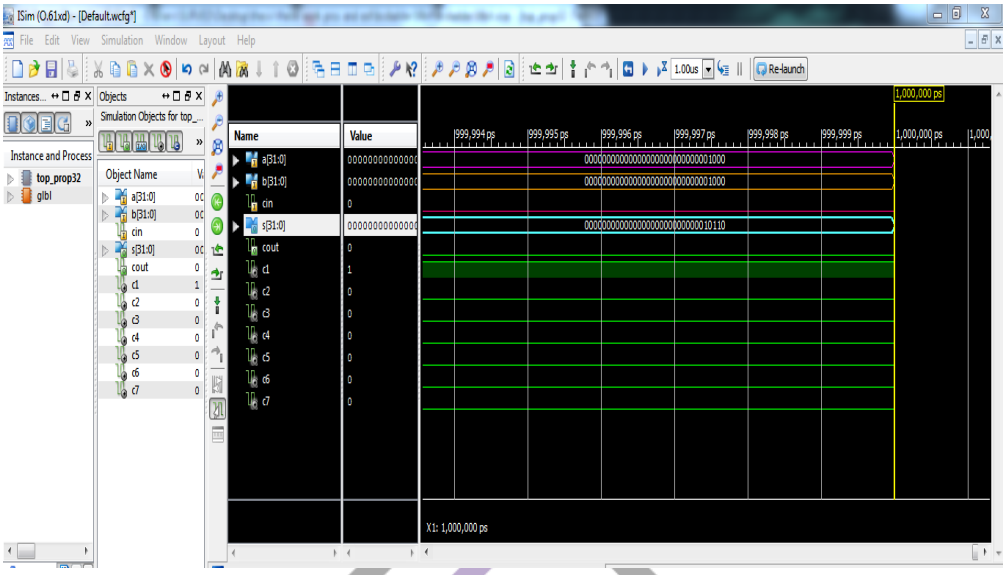


Figure 8 Output Waveform

The output can be observed in above form. It is the only way by which can test the output of the device. Here binary values are presented. Values are in form of HIGH and LOW here the HIGH is used to mention the output as "1" in the binary number system whereas the LOW is used to show the "0" in the binary number system.

The following inputs and outputs are providing to test the functionality of the BCD adder. Two inputs are "a" and "b" and another inputs carry in is presented as "cin". Two outputs are presented "s" and carry out as "cout". All these values are of 32 bits. In the current era of technology different kind of methods are adopted for designing the binary adders. These are the important methods which are currently using in the different system are per the requirement.

Here 64-bits of adder are presents and their performance are displayed in term of time delay and number LUTs to show the area covered.

Table 1 Comparison of different adder in delay

Topology	Delay (in ns)
Kogge-Stone Adder	38.656
Ladner-Fischer Adder	45.282
Brent-Kung Adder	43.961
Proposed BCD Adder	33.101

In table 1 shows that Kogge-Stone Adder, Ladner-Fischer Adder and Brent-Kung Adder all these having delay of 38.656 ns , 45.282 ns and 43.961 ns respectively. But in case of proposed BCD adder takes only 33.101 ns.

Proposed BCD Adder using majority gates gives the best results among the above mentioned Adders. It is almost 25% more efficient than Kogge-Stone Adder and more.

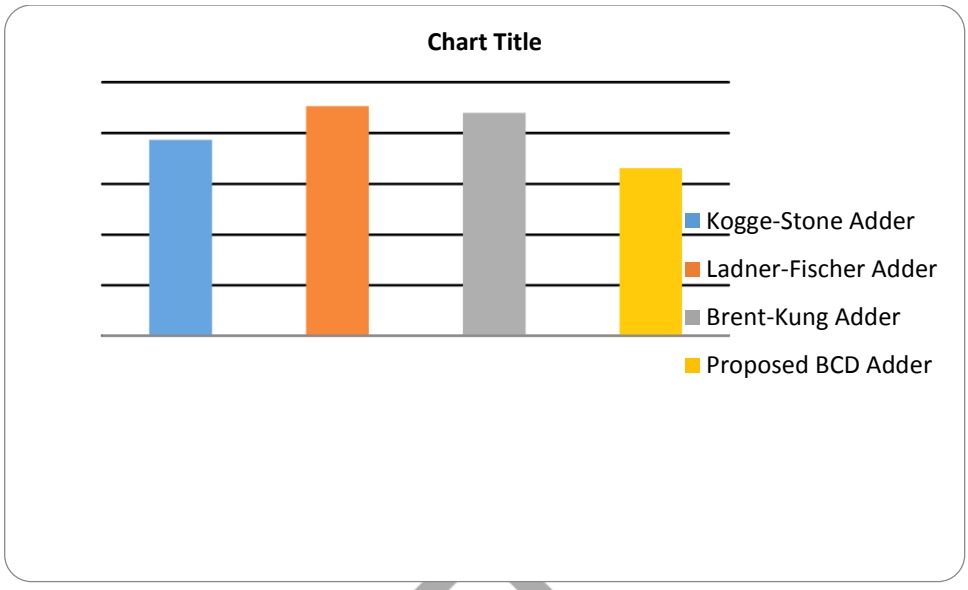


Figure 9 Comparison of different adder for delay in time

As observed in the figure 9 graph that the time taken by different adder to complete the task. The brief of the result is that in Kogge-Stone Adder that the delay is about 38 nanoseconds but in case of Ladner-Fischer Adder delay in 45 nanoseconds whereas in another case of Brent-Kung Adder it requires 43 nanoseconds and in Proposed BCD Adder the delay is of just 33 nanoseconds. The proposed BCD adder using majority gates is efficient in term of delay.

Table 2 Comparison of different adder in LUTs

Topology	No. of LUTs(#area)
Kogge-Stone Adder	408
Ladner-Fischer Adder	174
Brent-Kung Adder	170
Proposed BCD Adder	144

The table 2 shows that Kogge-Stone Adder, Ladner-Fischer Adder and Brent-Kung Adder all these having LUTs of 408, 174 and 170 respectively. But in the proposed BCD adder uses only 144,which means area covered by proposed BCD adder consumes much less area as compared to others. It is nearly 64.7 % less area compared to Kogge-Stone Adder.

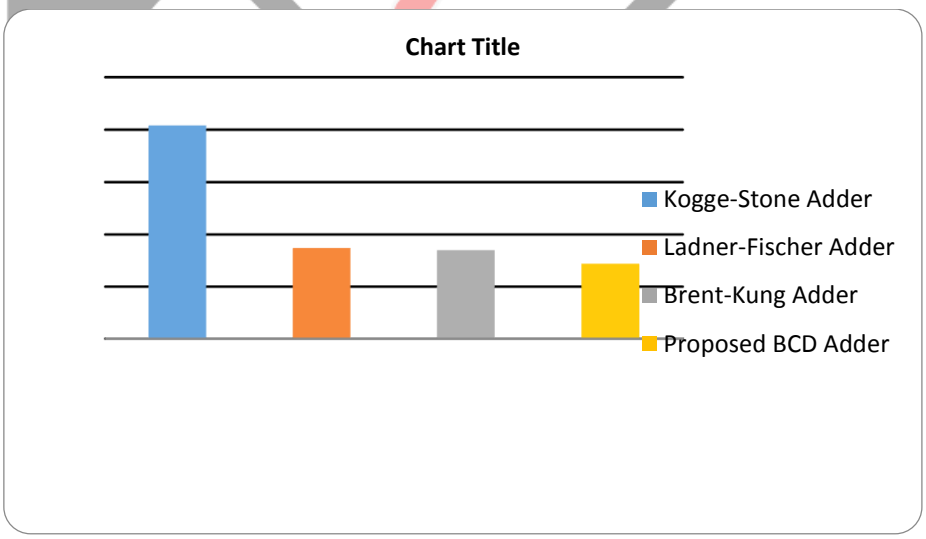


Figure 10 Comparison of different adder for the area in term of LUTs

CONCLUSION

The result of proposed work and different adders are mentioned in table 3. The Kogge-Stone Adder covers the area of 408 LUTs to perform its function now when take Ladner-Fischer Adder it can be construed in 174 LUTs which is quit less as compare to last adder. Now when talk about the Brent-Kung Adder it works by using 170 LUTs but in case of proposed BCD adder it utilized just 144 LUTs. So as result it is concluded that the proposed BDC adder is most efficient in terms of area.

From the above table it clear that proposed BCD adder using majority gates is more efficient than other adders. Here two parameters are compared that are delay and area in terms of LUTs. If delays are considered than proposed BCD adder is 14% faster than Kogge-Stone Adder, 27% faster than Ladner-Fischer Adder,24% faster than Brent-Kung adder.

If area is considered then no. of LUTs are counted. On comparing above mentioned adder it is clear that Kogge-Stone adder consumes 65% more area than proposed BCD adder, Ladner-Fischer adder uses 17% more area than proposed BCD adder and Brent-Kung adder more area ,that is 15% more area than proposed BCD adder using majority gates.

REFERENCES:

[1] Surved Pratap Singh and Shweta Agarwal, "Review Paper on Binary Coded Decimal Adder", International Journal for Science and Advance Research In Technology (IJSART) , Vol. 4 ,Issue 2 – Feb 2018.

[2] Surved Pratap Singh and Shweta Agarwal, "Design of Efficient 32bits BCD Adders in Quantum-Dot Cellular Automata", International Journal for Research in Applied Science & Engineering Technology (IJRASET) Vol. 6 ,Issue II, Feb 2018.

[3] P.D. Tougaw, W. Porod and C.S. Lent, "Quantum cellular automata: The physics of computing with quantum dot molecules", Department of Electrical Engineering, University of Notre Dame, Notre Dame, Nov 1994.

[4] V. Pudi and K. Sridharan, "New decomposition theorems on majority logic for low delay adder design in quantum dot cellular automata", IEEE Transactions. Circuits System. II, Express. Brief, Vol. 59, Issue: 10, pp. 678–682, Oct. 2012.

[5] H. Cho and E. E. Swartzlander, Jr., "Adder design and analyses for quantum-dot cellular automata", IEEE Transactions. Nanotechnology, Vol. 6, Issue: 3, pp. 374–383, May 2007.

[6] A. A. Bayrakci and A. Akkas, "Reduced delay BCD adder", Application-specific Systems, Architectures and Processors, IEEE International conference, pp. 266–271, Jul. 2007

[7] S. Sravani and P. Balakrishna, "Design of Efficient 32-Bit Parallel Prefix Ladner Fischer Adder", International Journal of Advanced Trends in Engineering, Science and Technology (IJATEST), Vol. 2, Issue: 2, Mar. 2017.