ISSN 2515-8260 Volume 07, Issue11, 2020

AN EFFICIENT DESIGN OF MULTIPLIER AND ADDER IN QUANTUM-DOT CELLULAR AUTOMATA TECHNOLOGY USING MAJORITY LOGIC

N. BHAVANI SUDHA[1], GAMINI SRIDEVI[2]

[1] PG Scholar, Dept of ECE, Aditya Engineering College, Surampalem, E.G(Dt), AP, India
[2] Professor, Dept of ECE, Aditya Engineering College, Surampalem, E.G(Dt), AP, India.

Abstract: ***Approximate computer arithmetic circuits based on CMOS technology have been extensively studied. Designs of approximate adders, multipliers and dividers for both fixed point and floating-point formats have been proposed. As a new paradigm in the nano scale technologies, approximate computing enables error tolerance in the computational process, it has also emerged as a low power design methodology for arithmetic circuits. Majority logic (ML) is applicable to many emerging technologies and its basic building block (the 3-input majority voter) has been extensively used in digital circuit design. In this project, we propose the design of a one bit approximate full adder based on majority logic. Furthermore, multi-bit approximate full adders are also proposed and studied, the application of these designs to quantum-dot cellular automata (QCA) is also presented as an example. The designs are evaluated using hardware metrics (including delay and area) as well as error metrics. Compared with other circuits found in the technical literature, the optimal designs are found to offer superior performance. Approximate half adder and full adder is designed. These Half adder and Full adder combinations are used to implement the Brent Kung Adder and multiplier. In the present work fast adders like RCA adders using compressor methodology and multiplication operations are performed by utilizing Majority gates. This paper also proposes the Wallace tree multiplier using proposed majority based Full adder .The designed multiplier is effective and efficient in terms of area-delay trade off, delay(speed) and power utilization.Project will be developed using verilog HDL. Xilinx ISE tool is used to perform the Simulation and Synthesis.***

Keywords: —majority logic, approximate adder, approximate multiplier, complement bits, approximate compressor, image processing.

1. INTRODUCTION:

Nanotechnology draws much attention from the public now-a-days. Because the current silicon transistor technology faces challenging problems, such as high power consumption and difficulties in feature size reduction, alternative technologies are sought from researchers.

Quantum-dot cellular automata (QCA) is one of the promising future solutions. Since it was first introduced in 1993, experimental devices for semiconductor, molecular, and magnetic approaches have been developed. Quantum dot cellular automata, which is an array of coupled quantum dots to implement Boolean logic functions. The advantage of QCA is high packing densities due to the small size of the dots, simplified interconnection and low area delay product. In 1982, Richard Feynman suggested an initial approach to quantizing a model of cellular automata. In 1985, David Deutsch presented a formal development of the subject. Later, Gerhard Grössing and Anton Zeilinger introduced the term "quantum cellular automata" to refer to a model they defined in 1988, although their model had very little in common with the concepts developed by Deutsch and so has not been developed significantly as a model of computation.

The first formal model of quantum cellular automata to be researched in depth was that introduced by John Watrous. This model was developed further by Wim van Dam, as well as Christoph Dürr, Huong LêThanh, and Miklos Santha, Jozef Gruska and Pablo Arrighi. However it was later realized that this definition was too loose, in the sense that some instances of it allow superluminal signaling. A second wave of models includes those of Susanne Richter and Reinhard Werner, of Benjamin Schumacher and Reinhard Werner, of Carlos Pérez-Delgado and Donny Cheung, and of Pablo Arrighi, Vincent Nesme and Reinhard Werner. These are all closely related, and do not suffer any such locality issue. In the end one can say that they all agree to picture quantum cellular automata as just some large quantum circuit, infinitely repeating across time and space.

Quantum Cellular Automata (QCA) refers to models of quantum computation, which have been devised in analogy to conventional models of cellular automata introduced by von Neumann. It may also refer to quantum dot cellular automata, which is a proposed physical implementation of "classical" cellular automata by exploiting quantum mechanical phenomena. QCA has attracted a lot of attention as a result of its extremely small feature size (at the molecular or even atomic scale) and its ultra-low power consumption, making it one candidate for replacing CMOS technology. A quantum-dot cellular automaton (QCA) is an attractive emerging technology suitable for the development of ultra dense low-power high-performance digital circuits [1]. For this reason, in the last few years, the design of efficient logic circuits in QCA has received a great deal of attention. Special efforts are directed to arithmetic circuits [2]–[16], with the main interest focused on the binary addition [11]–[16] that is the basic operation of any digital system. Of course, the architectures commonly employed in traditional CMOS designs are considered a first reference for the new design environment. Ripple-carry (RCA), carry look-ahead (CLA), and conditional sum adders were presented in [11]. The carry-flow adder (CFA) shown in [12] was mainly an improved RCA in which detrimental wires effects were mitigated. Parallel-prefix architectures, including Brent–Kung (BKA), Kogge–Stone, Ladner–Fischer, and Han–Carlson adders were analyzed and implemented in QCA in [13] and [14]. Recently, more efficient designs were proposed in [15] for the CLA and the BKA, and in [16] for the CLA and the CFA.

2. RELATED WORK

Approximate computing: an emerging paradigm for energy-efficient design by J. Han and M. Orshansky Approximate computing has recently emerged as a promising approach to

energy-efficient design of digital systems. Approximate computing relies on the ability of many systems and applications to tolerate some loss of quality or optimality in the computed result. By relaxing the need for fully precise or completely deterministic operations, approximate computing techniques allow substantially improved energy efficiency. This paper reviews recent progress in the area, including design of approximate arithmetic blocks, pertinent error and quality measures, and algorithm-level techniques for approximate computing.

Design of Majority Logic Based Approximate Arithmetic Circuits by C. Labrado, H. Thapliyal and F. Lombardi The increasing amount of circuit density possible in CMOS technology has the consequence of also increasing the power consumption of circuits using the technology. One possible method of offsetting these increased power demands is to use approximate computing designs in circuits where complete accuracy is not a strict requirement. These circuits use fewer logic gates which reduces power consumption at the cost of accuracy. Another possible method for reducing power consumption is to use an emerging nanotechnology which is already low power in nature. Combining approximate computing with an emerging nanotechnology has the potential to further cut power consumption. Unfortunately, existing approximate computing circuits were designed using standard logic gates found in CMOS technology which in turn can limit their effectiveness when implemented with the majority based logic used by some emerging nanotechnologies.

For that reason, we propose designs of approximate arithmetic units which are specifically designed for use in majority logic based technologies. Impact: Imprecise adders for low-power approximate computing by V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy. Low-power is an imperative requirement for portable multimedia devices employing various signal processing algorithms and architectures. In most multimedia applications, the final output is interpreted by human senses, which are not perfect. This fact obviates the need to produce exactly correct numerical outputs. Previous research in this context exploits error-resiliency primarily through voltage over-scaling, utilizing algorithmic and architectural techniques to mitigate the resulting errors. In this paper, we propose logic complexity reduction as an alternative approach to take advantage of the relaxation of numerical accuracy. We demonstrate this concept by proposing various imprecise or approximate Full Adder (FA) cells with reduced complexity at the transistor level, and utilize them to design approximate multi-bit adders. In addition to the inherent reduction in switched capacitance, our techniques result in significantly shorter critical paths, enabling voltage scaling. We design architectures for video and image compression algorithms using the proposed approximate arithmetic units, and evaluate them to demonstrate the efficacy of our approach. Post-layout simulations indicate power savings of up to 60% and area savings of up to 37% with an insignificant loss in output quality, when compared to existing implementations.

Implementation of Wallace Multiplier

The algorithm of WALLACE multiplier is based on the below matrix form. The partial product matrix is formed in the first stage by AND stages

ISSN 2515-8260 Volume 07, Issue11, 2020

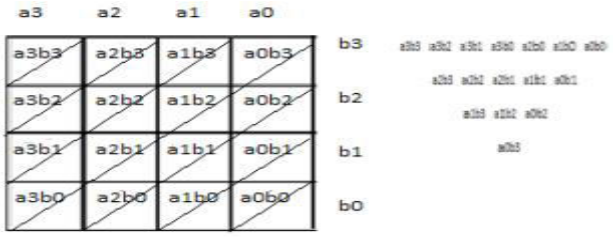


Fig.1-4x4 WALLACE Algorithm

Steps involved in WALLACE TREE multipliers Algorithm

Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding N results. Depending on position of the multiplied bits, the wires carry different weights. Reduce the number of partial products to two layers of full adders. Group the wires in two numbers, and add them with a conventional adder.. Product terms generated by a collection of AND gates.

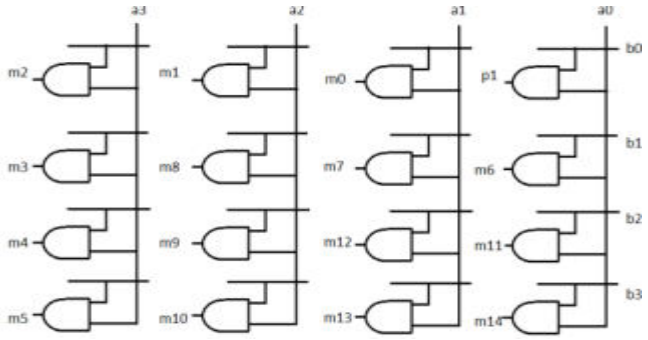
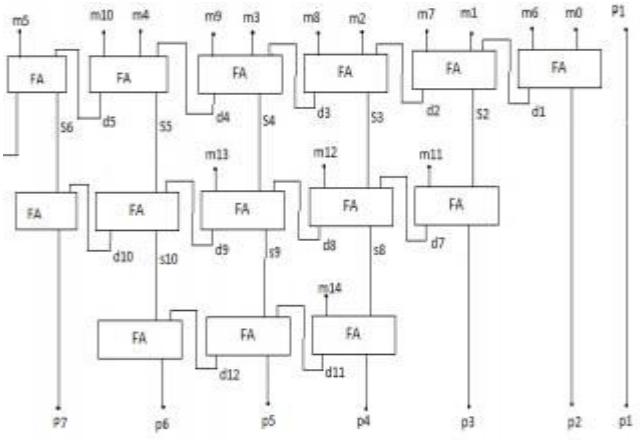


Fig 2 Product terms generated by a collection of AND gates.

Wallace Tree Multiplier Using Ripple Carry Adder

Ripple Carry Adder is the method used to add more number of additions to be performed with the carry in sand carry outs that is to be chained. Thus multiple adders are used in ripple carry adder. It is possible to create a logical circuit using several full adders to add multiple-bit numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is a ripple carry adder, since each carry bit "ripples" to the next full adder. The proposed architecture of WALLACE multiplier algorithm using RCA is shown in Figs.9 to 10 Take any 3 values with the same weights and gives them as input into a full adder. The result will be an output wire of the same weight. Partial product obtained after multiplication is taken at the first stage. The data's are taken with 3 wires and added using adders and the carry of each stage is added with next two data‟s in the same stage. Partial products reduced to two layers of full adders with same procedure. At the final stage, same method of ripple carry adder method is performed and thus product terms p1 to p8 is obtained.

ISSN 2515-8260 Volume 07, Issue11, 2020



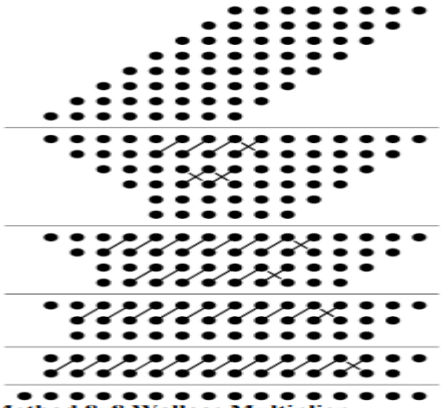


Fig 3 .4x4 Wallace Multiplier Implementation. **Fig 4 Method 8x8 Wallace Multiplier**

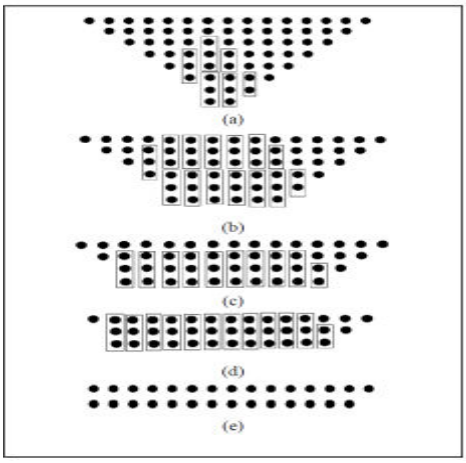


Fig 5 Column Compression scheme for 8x8 wallace multiplier.

- The simplest wallace tree is the full adder cell, more generally an n input wallace tree is an n input operation with log(n) base 2 outputs.
- The value of the output word is equal to the number of 1's in the input word.
- The number of adder levels increases logarithmically as the partial product rows increases.
- Make group of three's and apply Carry Save Adder(CSA) reduction in parallel.
- Each CSA layer produces two rows.
- These rows then, with other rows from other partial product groups, form a new reduce matrix.
- Iteratively, apply Wallace Reduction on the new generated matrix.
- This process continues until two rows are left.
- The final rows are added together for the final product.

3. IMPLEMENTATION OF MULTIIER

In electronics, an adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic unit(s), but also in other parts of the processor, where they are used to calculate addresses, table indices, and similar operations. Although adders can be constructed for many

numerical representations, such as binary-coded, decimal or excess-3, the most common adders operate on binary numbers. In cases where two's complement or ones' complement is being used to represent negative numbers, it is trivial to modify an adder into an adder subtract or. Other signed number representations require a more complex adder. Adders are fundamental circuits for most digital systems and several adder designs in QCA have been proposed, and a performance comparison was improved. Better adder performance depends on minimizing the carry propagation delay and reducing the area.

An adder designed as proposed runs in the RCA fashion, but it exhibits a computational delay lower than all state-of the-art competitors and achieves the lowest area-delay product (ADP). A QCA is a nanostructure having as its basic cell a square four quantum dots structure charged with two free electrons able to tunnel through the dots within the cell [1]. Because of Coulombic repulsion, the two electrons will always reside in opposite corners. The locations of the electrons in the cell (also named polarizations *P)* determine two possible stable states that can be associated to the binary states 1 and 0.

Although adjacent cells interact through electrostatic forces and tend to align their polarizations, QCA cells do not have intrinsic data flow directionality. To achieve controllable data directions, the cells within a QCA design are partitioned into the so-called clock zones that are progressively associated to four clock signals, each phase shifted by 90°. This clock scheme, named the zone clocking scheme, makes the QCA designs intrinsically pipelined, as each clock zone behaves like a D-latch [8]. QCA cells are used for both logic structures and interconnections that can exploit either the coplanar cross or the bridge technique [1], [2], [5], The fundamental logic gates inherently available within the QCA technology are the inverter and the MG. Given three inputs *a*, *b,* and *c*, the MG performs the logic function reported in (1) provided that all input cells are associated to the same clock signal clk *x* (with *x* ranging from 0 to 3), whereas the remaining cells of the MG are associated to the clock signal clk *x*+1 operands by cascading *n* full-adders (FAs). Even though these addition circuits use different topologies of the generic FA, they have a carry-in to carry-out path consisting of one MG, and a carry-in to sum bit path containing two MGs plus one inverter. As a consequence, the worst case computational paths of the *n*-bit RCA and the *n*-bit CFA consist of (*n*+2) MGs and one inverter.

$$M(abc) = a \cdot b + a \cdot c + b \cdot c. \qquad (1)$$

A CLA architecture formed by 4-bit slices was also presented in [11]. In particular, the auxiliary propagate and generate signals, namely $pi = ai + bi$ and $gi = ai \cdot bi$, are computed for each bit of the operands and then they are grouped four by four. Such a designed *n*-bit CLA has a computational path composed of $7+4\times(\log 4\ n)$ cascaded MGs and one inverter. This can be easily verified by observing that, given the propagate and generate signals (for which only one MG is required), to compute grouped propagate and grouped generate signals; four cascaded MGs are introduced in the computational path. In addition, to compute the carry signals, one level of the CLA logic is required for each factor of four in the operands word-length. This means that, to process *n*bit addends, log4 *n* levels of CLA logic are required, each contributing to the computational path with four cascaded MGs. Finally,

ISSN 2515-8260 Volume 07, Issue11, 2020

the computation of sum bits introduces two further cascaded MGs and one inverter.The Approximate multiplier using adder's is presented.

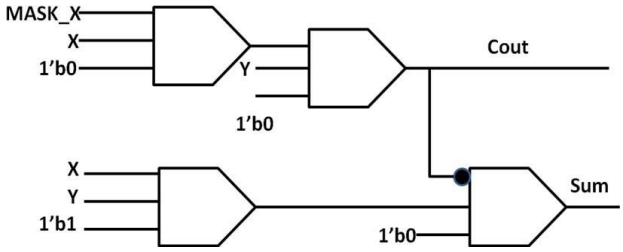


Figure 6. Proposed 1-bit Approximate Half adder

Here, we are using Approximate Half adders and Full adders for approximate and accurate multiplications. The outputs of the approximate half adder are

Sum = ~(MASK_X & X & Y) & (X | Y) (3)

Cout = MASK_X & X & Y (4)

From above equations, the error distance between accurate mode and approximate mode can be analysed as shown in table 1.

Table 1. Truth Table of the Proposed Adder

Accurate mode (MASK_X=1)				**Approximate mode (MASK_X=0)**		
X	Y	S	Cout	S	Cout	Error
0	0	0	0	0	0	0
0	1	1	0	1	0	0
1	0	1	0	1	0	0
1	1	0	1	1	0	1

So, here as the error distance is very less, thus it can act as both approximate and accurate very effectively.

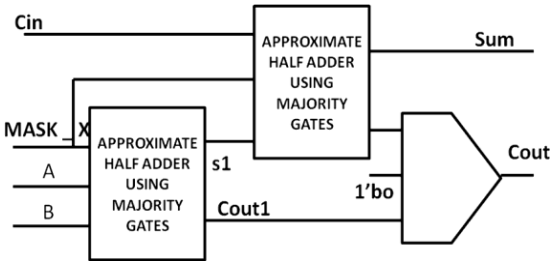
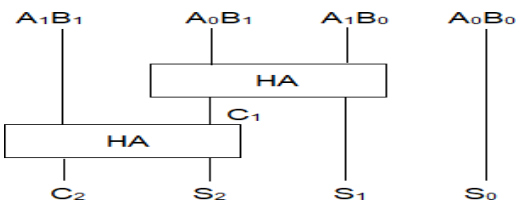


Figure 7. Proposed 1-bit Approximate Full adder



ISSN 2515-8260 Volume 07, Issue11, 2020

Figure 8. Proposed 2-bit approximate multiplication

By using two approximate half adders, the full adder is designed in Figure 7. These Half adder and Full adder combinations are used to implement the Brent Kung Adder. Figure 8 performs the 2-bit multiplication operation between A, B. Initially, four partial products namely A0B0, A1B0, A0B1 and A1B1 are generated by using bitwise AND operation between A, B. The AND gate can be implemented by making any one of the inputs in Majority gate to zero. For generating the final product bits two Half Adders are used as shown in Figure 7. And the product bits are P0=A0B0, C1S1=A1B0+A0B1 (P1=S1), C2S2=C1+A1B1 (P2=S2 and P3=C2). To perform the half adder operation Fig. 6 is used. By making anyone of Full adder input to zero, the circuit will act as a Half Adder. The multiplier consists of a control signal to select its operational mode by controlling the operation of half adders and full adders in Brent Kung Adder.

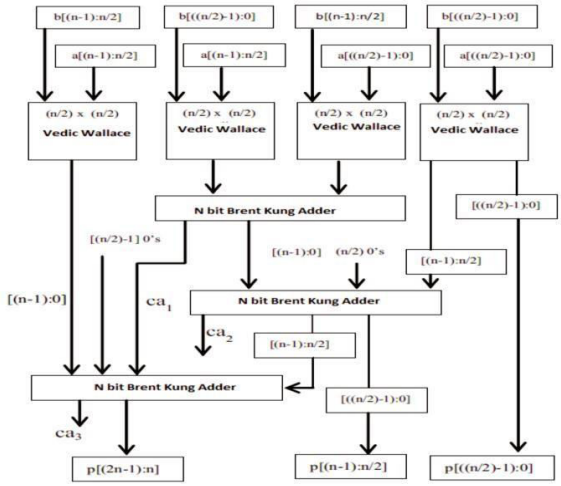


Figure 9. Architecture diagram of N bit Approximate multiplier

To develop a N-bit multiplier, we need 4 Nby2 multipliers. For example, for implementing 4-bit multiplier it requires four 2-bit multipliers. Here, three N-bit Brent Kung approximate adders are needed. First adder adds the second and third multiplier outputs, and the addition output will be forwarded to the second stage adder. First Nby2 multiplier fed half of the LSB output bits as the final outputs, and the next half MSB output bits are fed as input to the second stage adder. To avoid mismatches of the array Nby2 zeros are added to second and third stage adders. After completing all the additions, product P will be generated.

Simulation Method

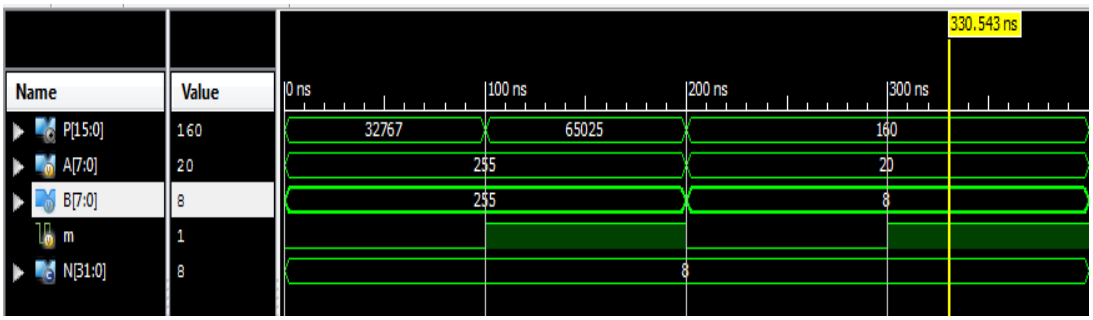


Figure 10. Multiplication of two numbers

The functionality of the circuit was tested using the Xilinx ISE software. The signals a and b

of size 8-bit are given as Inputs. Here **m** is the mode selection bit, when m=0 it will act as approximate multiplier for some testcases and when m(mask)=1 it will act as an accurate multiplier for all testcases. The two input signals a, b are of values {255,255}, when m=0 then the approximate multiplier output is 12887 and when m=1 the accurate multiplier output is 7550 can be observed in output variable P. The simulated result is shown in Fig.10.

4. RESULTS AND DISCUSSION

The study has been carried out by considering Look Up Table(LUT), Time Delay and Power Consumption. The Look Up Tables used in the proposed system are 54 as compared to the previous work of 108. It indicates very less area is used for the proposed design. The consumed path delay in the proposed system is 14.015ns as compared to the previous work of 25.701ns.

Comparison

Parameter	Proposed	Extension
LUTS	**108**	**54**
Time	**25.701ns**	**14.015ns**
Area used	**28%**	**14%**

The performance of the proposed system is compared with that of the existing method. We can observe that the designed multiplier is effective and efficient in terms of area-delay tradeoff, delay (speed) and power utilization when compared to the previous one.

5. CONCLUSIONS

This paper has presented a design, analysis and evaluation of majority logic based approximate adders and approximate multipliers. ML based 1-bit, 2-bit and multi-bit AFAs have been proposed; these deigns have a reduced circuit complexity and reduced delay compared to the exact counterpart while only incurring in a modest loss in accuracy. By combining multiple approximate techniques (such as the proposed MLACs and approximate PPR circuitry) with the so-called complement bits, ML based multi-bit AMs have been proposed: an influence factor has been defined to measure the importance of different complement bits; selection of the complement bits has also been pursued by an in-depth analysis depending on the size of multipliers; multiple MLACs has been proposed based on MLAFAs or K-Map simplification, and has been employed in the approximate PPR circuitry design for 8 _ 8 MLAMs. The following conclusions can be drawn.

(1) By combining the proposed 1-bit MLAFA and the existing 1-bit MLAFA, multi-bit MLAFAs can be designed with a relatively large error. The proposed MLAFA33 designed by truth table reduction can provide the solution for improving accuracy. For 8-bit adders, MLAFA1212-1233 (with a reduction of 58% in majority gates as well as 50% in delay) and MLAFA1212-3333 (with a reduction of 50% in majority gates as well as 50% in delay) are superior to other designs.

(2) Based on the proposed 2_2 MLAM, we can selectively design multi-bit multipliers by adding the complement bits. From the presented theoretical analysis, for a 4 _ 4 multiplier, when p changes from 3 to 4, the NMED increases sharply; for an 8 _ 8 multiplier, when p

is smaller than 10, the NMED increases slowly but when p is larger than 10, the NMED increases rapidly. So p=3 and p=10 are the best choices for the 4 _ 4 multiplier and the 8 _ 8 multiplier, respectively.

6. REFERENCES

1. S.-L. Lu, “Speeding up processing with approximation circuits,” Computer, vol. 37, no. 3, pp. 67–73, Mar. 2004.
2. J. Han and M. Orshansky, “Approximate computing: an emerging paradigm for energy-efficient design,” in Proc. ETS, pp. 1-6, May2013
3. C. Labrado, H. Thapliyal and F. Lombardi “Design of Majority Logic Based ApproximateArithmetic Circuits,” inProc. IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2122-2125, May 2017.
4. V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, “Impact: Imprecise adders for low-power approximate computing,”in Proc. Int.Symp. Low Power Electronics and Design (ISLPED) , pp. 409–414, Aug. 2011.
5. Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, “Approximate xor/xnor-based adders for inexact computing,” in Proc. 13th IEEE Conf. Nanotechnology (IEEE-NANO), pp. 690–693, Aug.2013.
6. V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy. “Low-power digital signal processing using approximate adders,”IEEE Trans. Comput.-Aided Des. Integ. Circuits Syst, vol. 32,pp. 124–137, 2013.
7. S. Rehman, W. El-Harouni, M. Shafique, A. Kumar, and J. Henkel. “Architectural-Space Exploration of Approximate Multipliers,”. in Proc. Int. Conf. Comput.-Aided Des. (ICCD), pp. 1-6,Nov.2016.
8. N. Maheshwari, Z. Yang, J. Han, and F. Lombardi. “A design approach for compressor based approximate multipliers,” in Proc. 28th Int. Conf. VLSI Des., pp. 209-214,Jan. 2015.
9. J. Liang, J. Han, and F. Lombardi, “New metrics for the reliability of approximate and probabilistic adders,” IEEE Transactions on Computers, vol. 62, no. 9, pp. 1760–1771, 2013.
10. K. Walus and G. Jullien,"Design tools for an emerging SoC technology: quantum-dot cellular automata," in Proc. IEEE, vol. 94, no. 6,pp. 1225-1244,2006.
11. C. Lent and P Tougaw, "A device architecture for computing with quantum dots," in Proc. IEEE, vol. 85,no. 4,pp. 541-557,1997.
12. M. Vacca, M. Graziano, J. Wang, F. Cairo, G. Causapruno, G. Urgese,A. Biroli, and M. Zamboni, “Nano magnet logic: An architectural level over view,” Lecture Notes in Computer Science, pp. 223–256, 2014.
13. A. Khitun and K. L. Wang, “Nano scale computational architectures with spin wave bus,” Superlattices and Microstructures, vol. 38, no. 3,pp. 184– 200, 2005.
14. H. Cho and E. E. Swartzlander, “Adder and multiplier designs in quantum dot cellular automata,” IEEE Transactions on Computers, vol. 58, no. 6, pp. 721–727,2009.
15. V. Pudi and K. Sridharan, “Low Complexity Design of Ripple Carry and Brent–Kung Adders in QCA,”IEEE Transactions on Nanotechnology, vol. 11, no. 1, pp. 105-119, 2012.

ISSN 2515-8260 Volume 07, Issue11, 2020

16 G. Sridevi and O K Kennedy, “Performance analysis of a low power and high speed carry select adder”, International conference on current trends in computer, electronics and Communication, September 2017.