Implementation of Majority Logic Based Multi Bit Approximate Adders and Multipliers

S.Ashiq[1], S.M.K.Sukumar Reddy[2], Dr.S.Siddeswara Reddy[3]

[1]PG scholar, Dept of ECE, VITS , Proddatur, Kadapa, AP, India

[2]Assisant Professor,M.Tech(Ph.D), Dept of ECE, VITS, Proddatur, Kadapa, AP, India

[3]Associate Professor & HOD, Dept of ECE, VITS, Proddatur, Kadapa, AP, India

Abstract— Approximate computing, as a new paradigm for nanoscale technologies, addresses error tolerance in the computational process in order to increase performance and minimize power consumption. Many upcoming nanotechnologies can benefit from majority logic (ML), and its fundamental building block (the 3-input majority voter, MV) has been widely employed in digital circuit design. The suggested multipliers use approximate compressors and a reduction circuitry with so-called complement bits; the proposed adders use approximate compressors and a reduction circuitry with so-called complement bits. To examine the relevance of different complement bits depending on the size of the multiplier, an influence factor is established and analyzed; a strategy for complement bit selection is also offered. Hardware measurements (such as delay and gate complexity) and error metrics are used to assess the suggested designs. When compared to other ML-based designs reported in the literature, the proposed designs are found to perform better. To demonstrate the validity of the proposed designs, case studies of error-resilient applications are shown.

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

1. Introduction

Power dissipation is increasingly becoming a challenge for advanced integrated circuit design; although emerging nanoscale technologies have been proposed to replace CMOS at the end of Moore's law, the issue of power consumption remains unabated, because integration density of these nanoelectronics devices continues to increase at a high rate. Approximate computing is a promising technique to reduce power consumption and improve performance of circuits and systems by allowing computational errors in error-tolerant applications, such as multimedia signal processing, machine learning and pattern recognition [1-2]. Approximate computer arithmetic circuits based on CMOS technology have been extensively studied. Designs of approximate adders, multipliers and dividers for both fixedpoint and floating-point formats have been proposed [3-7]. Error metrics such as the mean error distance (MED), the normalized MED (NMED) and the relative MED (RMED) [8] have been proposed to analyze the errors introduced in the operations of approximate arithmetic circuits.

However, the approximate designs of CMOS circuits cannot be immediately applied to many emerging technologies such as QCA [9-10], nanomagnetic logic[11], and spinwave devices [12] due to the very different underlying logic structure of these devices. Emerging devices rely on majority logic (ml) which is a substantially different framework from conventional boolean logic. The majority gate performs a multi-input logic operation and is shown in fig. 1;

$$F = M(A, B, C) = AB + BC + AC \quad (1)$$

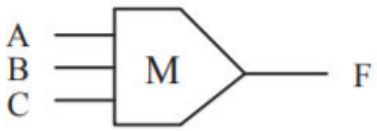


Fig. 1. Majority gate (3-input voter).

Research on the design of ml approximate circuits has only recently been pursued; in [3] the authors proposed a one-bit full adder circuit. In this paper, we propose a few ml-based approximate full adders; one-bit as well as multi-bit approximate adders are considered. Qca technology is used as a case study to show the validity of the proposed designs. Hardware evaluation and error analysis are also provided.

2. Background

2.1. Quantum-dot Cellular Automata

QCA makes use of the polarization state of cells to encode binary information and Coulombic force interactions between cells to achieve circuit functionality; these features make this technology substantially different from CMOS. As shown in Fig. 2a, a QCA cell consists in its simplest form of four quantum dots and two electrons that can tunnel between them. Due to Coulombic repulsion, electrons are forced to occupy the opposite diagonal vertices (dots). This forms two different polarization states (i.e., -1, +1) for each cell, thus representing logic values of 0 and 1. QCA requires a clocking scheme with four different operational phases, i.e. Switch, Hold, Release, and Relax; each adjacent so-called zone is shifted in phase by 90 degrees to control the flow of information. For a majority gate in QCA, which can be seen in Fig. 2b, there is a 0.25 clock delay in each clocking zone. As another basic gate in QCA, an inverter is shown in Fig. 2c. More information on the clocking scheme and QCA technology can be found in [9-10]

A. ML based One-Bit Exact and Approximate Full Adders

Fig. 3 shows a one-bit accurate full adder based on ML; it consists of 3 majority gates and 2 inverters [14]. The inputs to the one-bit adder are given by A, B, C while S and C are the outputs. The outputs Cand S are expressed as follows:

$$C_{out} = AB + BC + AC = M(A, B, C) \quad (2)$$

$$S = A \oplus B \oplus C = M(\overline{C_{out}}, M(A, B, \bar{C}), C) \quad (3)$$

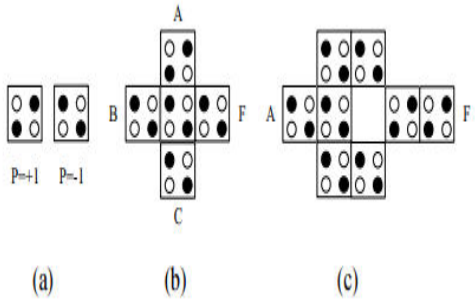


Fig. 2. QCA basic elements: (a) QCA cell, (b) QCA majority gate (voter), and (c) QCA inverter

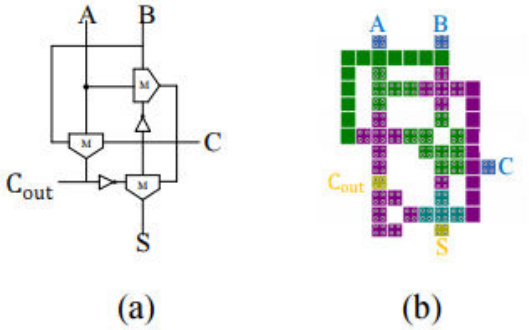


Fig. 3. One-bit accurate full adder: (a) schematic of accurate adder, (b) layout of accurate adder in QCA [14]

Labrado et al have proposed a one-bit approximate full adder (AFA1) in [3], whose schematic and layout are shown in Fig. 4. AFA1produces the output S as the complement of C but introduces 2 errors (among the 8 input combinations) when computing the output S, (as shown in Table I).

Fig. 4. One-bit approximate full adder: (a) schematic of AFA1, (b) layout of AFA1 in QCA [3].

The circled entries in the truth table denote the instances in which the outputs of the approximate full adder differ from the accurate full adder. The equations for the carry out and the sum are as follows:

$$C_{out} = M(A, B, C) \quad (4)$$

$$S = \overline{C_{out}} \quad (5)$$

B. Error Metrics for Approximate Circuits

As approximate computing introduces errors, error metrics are required to evaluate the accuracy of approximate circuits. In this paper, we evaluate approximate designs using the MED and the NMED. The MED is defined as the average of the Error Distance (ED) which is the absolute difference between the approximate and the accurate results across all possible inputs. The NMED is the normalized MED. The definitions of ED, MED and NMED are as follows: where X, Y, n and MAX denote the accurate result, the approximate result, the counts of all possible inputs and the maximum value of the result, respectively.

3. PROPOSED ONE-BIT APPROXIMATE FULL ADDER

In this section, a one-bit approximate full adder is proposed, which is presented and compared with one-bit accurate full adder and an existing one-bit approximate full adder.

A. Proposed One-bit Approximate Full Adder

Inspired by [3], we propose a new one-bit approximate full adder, namely, AFA2. Consider the truth table in Table I, C is nearly the same as C except in two of the 8 input combinations. Therefore, C can be approximately considered as C to save a majority gate compared with the one-bit accurate full adder when computing the carry out of a one-bit full adder)

B. Comparison and Discussion

A comparison in terms of number of majority gates (MV), number of inverters (INV), MED, NMED, delay (D) and area (A) between the accurate adder [14], AFA1 [3] and the proposed approximate full adder AFA2 is reported in Table II. Compared with the accurate adder [14], AFA2 saves 2 majority gates, 1 inverter and 0.25 clock cycles of delay and improves the area of the design by up to 72%. Moreover, AFA2 has a smaller area than AFA1[3] in QCA.

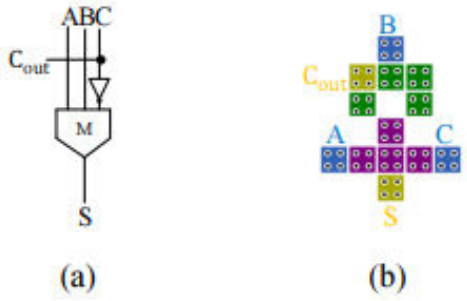


Fig. 5. Proposed one-bit approximate full adder: (a) schematic of AFA2, (b) layout of AFA2 in QCA.

Table1: Truth Table of 1-bit MLAFAs

Inputs			Accurate		AFA1[3]		AFA2	
A	*B*	*C*	C_{out}	*S*	C_{out}	*S*	C_{out}	*S*
0	0	0	0	0	0	(1)	0	0
0	0	1	0	1	0	1	(1)	(0)
0	1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	1	0
1	0	0	0	1	0	1	0	1
1	0	1	1	0	1	0	1	0
1	1	0	1	0	1	0	(0)	(1)
1	1	1	1	1	1	(0)	1	1

Table 2: Comparison of 1-bit MLAFAs

ADDER TYPE	MV	INV	D	A(nm^2)	MED	NMED
Accurate [14]	3	2	0.5	60050	0	0
AFA1[3]	1	1	0.25	18902	0.25	0.083
AFA2	1	1	0.25	16284	0.25	0.083

3.1 Proposed multi-bit approximate adders

In this section, multi-bit approximate full adders are proposed by merging the proposed and the existing one-bit approximate full adders. Both the corresponding hardware designs and error metrics are evaluated.

A. Proposed two-bit approximate adders: The inputs to the two-bit adder are given by a = aa, b = bb, c,while s = ss, and care the outputs. By cascading two one-bit approximate full adders (AFA1 and AFA2), four different combinations are possible for the two-bit approximate full adder; they are shown in fig.6. AFA1 cascaded with AFA1 results in the two-bit AFA11 design. Similarly, AFA2 cascaded with AFA2 results in AFA22 design. AFA12 consists of AFA1 and AFA2, in which AFA1 is used to compute the LSB; the opposite is applicable to AFA21. The layouts of these two-bit approximate QCA adders are shown in fig. 7.

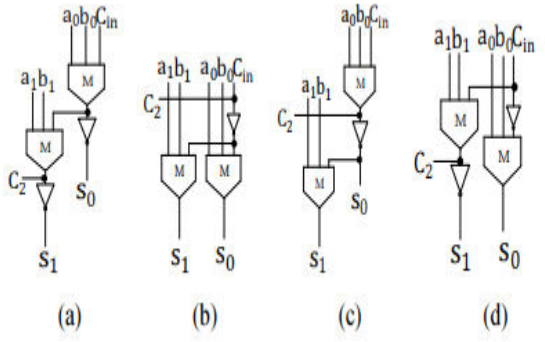


Fig. 6. Schematics of proposed 2-bit approximate full adders: (a) AFA11, (b) AFA22, (c) AFA12 and (d) AFA21.

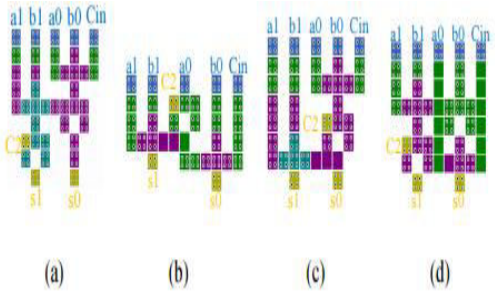


Fig. 7. Layouts of proposed 2-bit approximate full adders in qca:(a) AFA11, (b) AFA22, (c) AFA12 and (d) AFA21.

The proposed two-bit approximate adders introduce errors for 14 of the 32 input combinations; the MED and NMED of the four approximate adders are provided in table iii. The error results show that by cascading two of the same type of one-bit approximate full adder, the med and NMED are larger than cascading two different types of one-bit approximate full adder; however, AFA22 incurs in the smallest area and has less delay than AFA12. Considering the number of gates required in an implementation, AFA12 requires one less inverter than AFA21. In terms of the delay, AFA21 needs 0.25 less clocking zones than AFA12.

Table 3: Comparison of 2-bit MLAFAs

ADDER TYPE	MV	INV	D	A(nm^2)	MED	NMED
AFA11	2	2	0. 5	44073	0.75	0.107
AFA22	2	1	0.25	35383	0.75	0.107
AFA12	2	1	0.5	39267	0.625	0.089
AFA21	2	2	0.25	41069	0.625	0.089

Fig. 8 shows the comparison results by considering both the area-delay product and the NMED; AFA21 is the best design as it is the closest to the origin. Generally, AFA12 and AFA21 (with mixed types of one-bit approximate full adders) show better performance compared with those with only a single type of approximate full adders.

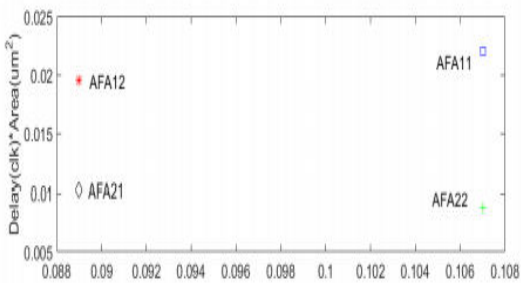


Fig. 8. Evaluation of proposed 2-bit approximate full adders (NMED vs delayarea product).

B. Proposed four-bit approximate adders

Similar to the two-bit approximate full adder, we can design a four-bit approximate full adder by cascading two two-bit approximate full adders. AFA12 and AFA21 are selected from these two two-bit approximate full adders as these designs show better overall performance than the other two schemes. These four combinations are shown in fig. 9, while their implementations in QCA are shown

in fig. 10. Table iv shows that the proposed designs require fewer gates than an accurate full adder, but at the cost of a reduced accuracy. An improvement of up to 50% in delay and up to 67% in area is achieved. Although AFA1221 has advantages in terms of the reduced number of gates and delay, its med/NMED is the largest. AFA2121 and AFA2112 have the same med/NMED, but AFA2121 has less delay. Compared with AFA2112, AFA1212 requires one less inverter with a reduction in med.

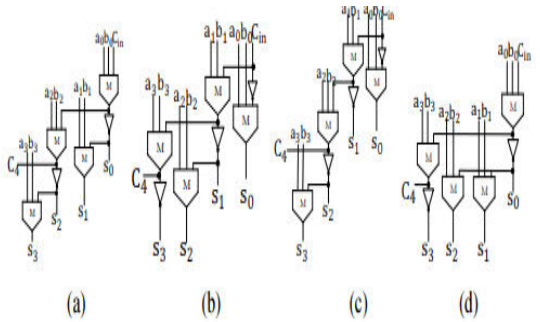


Fig. 9. Schematics of proposed 4-bit approximate full adders: (a) AFA1212, (b) AFA2121, (c) AFA2112 and (d) AFA1221.

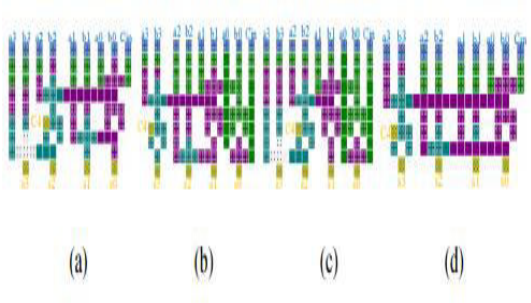


Fig. 10. Layouts of proposed 4-bit approximate full adders in qca: (a) AFA1212, (b) AFA2121, (c) AFA2112 and (d) AFA1221.

Fig. 11 shows that AFA2121 is the best design. The NMED for AFA1221 is rather large and the values for NMED of AFA1212, AFA2121, AFA2112 are very close, but AFA2112 and AFA1212 require more delay. For four-bit designs, the schemes in which two of the same type of the proposed two-bit approximate full adders are cascaded have better performance than cascading different types of approximate full adders.

Table 4: Comparison of 4-bit MLAFAs

ADDER TYPE	MV	INV	D	A(nm^2)	MED	NMED
CFA4[14]	12	8	1.5	405000	0	0
RCA4[15]	12	4	1.75	254200	0	0
AFA1212	4	2	0.75	76120	2.83	0.091
AFA2121	4	3	0.5	82619	2.87	0.092
AFA2112	4	3	0.75	83936	2.87	0.092
AFA1221	4	2	0.5	82085	5.45	0.175

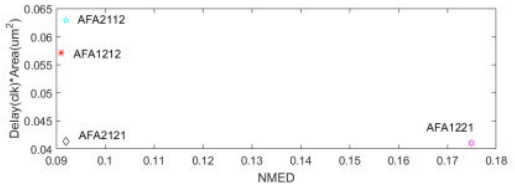
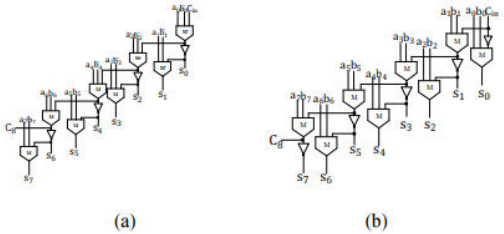
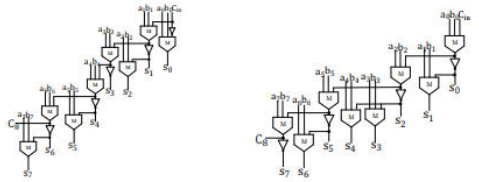


Fig. 9. Evaluation of proposed 4-bit approximate full adders (NMED vs delay area product).

C. Proposed eight-bit approximate adders

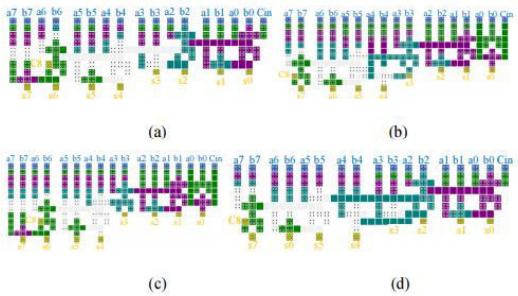
consider an eight-bit adder with inputs, we have designed eight-bit approximate adders by cascading two four-bit approximate adders by using AFA1212 and AFA2121, as they show better overall performance than the other two designs. The proposed eight-bit approximate adders are shown in fig. 12; their implementations in qca are shown in fig. 13. The comparison results are detailed in table v. The proposed designs significantly reduce the number of gates and delay but at the cost of a decrease in accuracy. In terms of gates, AFA1212-1212 and AFA1212-2121 require one less inverter than the other adders; AFA2121-2121 and AFA1212-2121 incur less delay than the other adders.





(c) (d)

Fig. 12. Schematics of proposed 8-bit approximate full adders: (a) AFA1212-1212, (b) AFA2121-2121, (c) AFA2121-1212 and (d) AFA1212-2121.



(a) (b)

(c) (d)

Fig. 13. Layouts of proposed 8-bit approximate full adders in QCA: (a) AFA1212-1212, (b) AFA2121-2121, (c) AFA2121-1212 and (d) AFA1212- 2121.

Table 5: Comparison of 8-bit MLAFAs

ADDER TYPE	MV	INV	D	A(nm^2)	MED	NMED
CFA8[14]	24	16	2.5	948700	0	0
RCA8[15]	24	8	2.75	745200	0	0
AFA1212-1212	8	4	1.25	184640	46.20	0.090
AFA2121-2121	8	5	1	206425	47.02	0.092
AFA2121-1212	8	5	1.25	218257	46.40	0.091
AFA1212-2121	8	4	1	191126	46.82	0.092

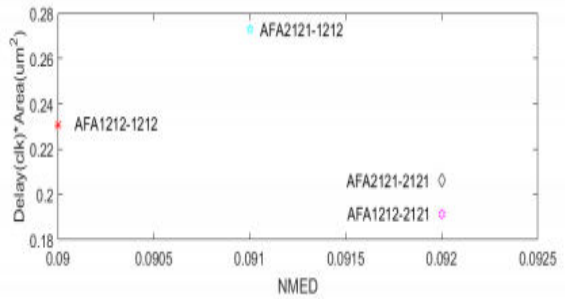


Fig. 14. Evaluation of proposed 8-bit approximate full adders (NMED vs delay area product).

3.2 ML based approximate multipliers

The designs of ml based approximate multipliers are studied in this section based on 2 × 2 MLAMs. The so-called complement bit is introduced through a selection scheme to compensate errors. Consider fig. 7 and the proposed design flow of n × n MLAMs. The multiplicand an−1an−2an−3an−4 · · · a3a2a1a0 and the multiplier bn−1bn−2bn−3bn−4 · · · b3b2b1b0 are first divided into n/2 modules (each of 2 bits as a unit); then, these modules are substituted into the expression to calculate the partial product, while at the same time, selectively adding the compensation bits as per the size of the multiplier. Next, for efficient compression, a partial product reduction (PPR) circuitry which uses exact or approximate compression is employed. This depends on the distribution of the generated partial products (PPS) and the compensation bits, such that the pp of two rows (or a carry in the lowest order) can be obtained. Finally, the final product can be calculated by the final exact adder.

A. 2×2-MLAM

By mapping the 2 × 2 am design into ML, out1 requires three majority gates, which is two more than out0 and out2

$$out_0 = M(A_0, B_0, 0)$$

$$out_1 = M(M(A_1, B_0, 0), M(A_0, B_1, 0), 1)$$

$$out_2 = M(A_1, B_1, 0)$$

Therefore, this should be further improved; furthermore, with an increase of design scale, errors will increase substantially, so unacceptable in most cases. Taking these issues into account, one is employed as the out1 of the 2 × 2 MLAM, the other is used as a compensation bit. In this paper, we just take one case into consideration. The other case follows the same rules.

$$out_1 = M(A_0, B_1, 0)$$

$$\triangle = M(A_1, B_0, 0)$$

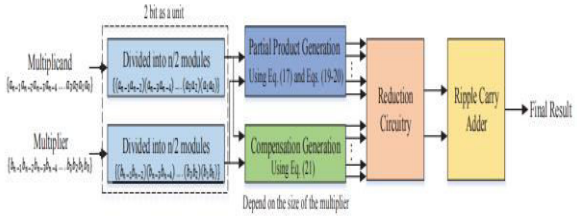


Fig. 15. Proposed design flow of n × n MLAMs.

By considering the 2 × 2 MLAM as a module, larger multipliers can be constructed by dividing the operands into several units, where 4 represents a complement bit. Fig. 15and Figure 16shows the operations of the 4 × 4 and 8 × 8 MLAMs with all complement bits that need to be further reduced.

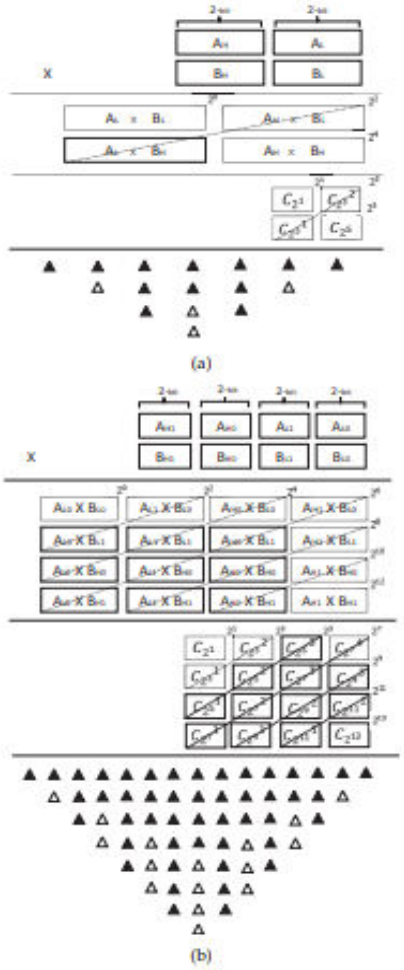


Fig. 16. PPG and complement bit generation of MLAMs: (a) 4×4 MLAM, and (b) 8 × 8 MLAM.

4. Simulation results

The proposed QCA approximate full adders are designed and simulated by using the XILIN ISE tool for the one-bit case. XILIN ISE is a QCA layout and simulation tool developed at the University of Calgary. The design and simulation is as follows. First, we generate the layout of the proposed one-bit approximate full adder. Then, we design multi-bit adders using the one-bit layout.

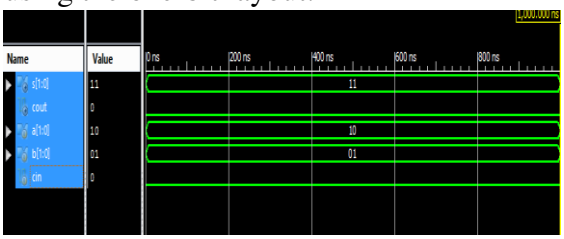


Fig 4.1 AFA 11

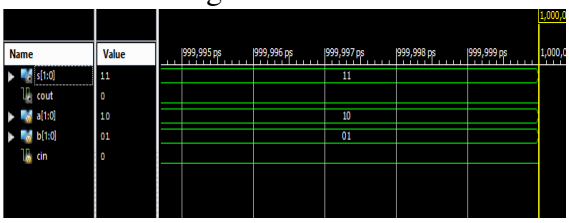


Fig 4.2AFA 22

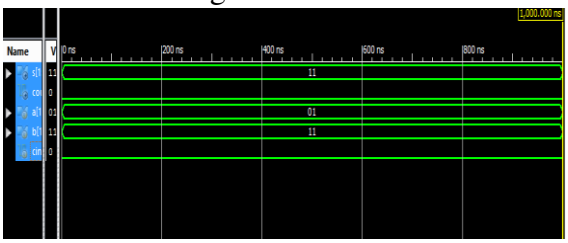


Fig 4.3 AFA 21

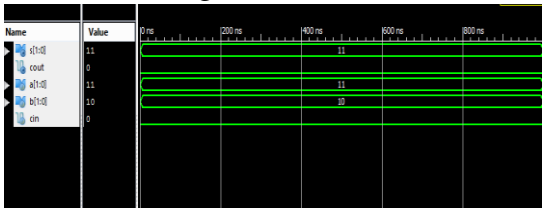


Fig 4.4 AFA 12

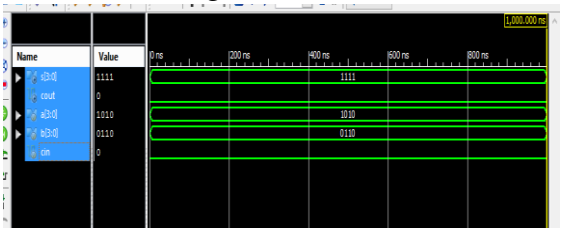


Fig 4.5 AFA 1212

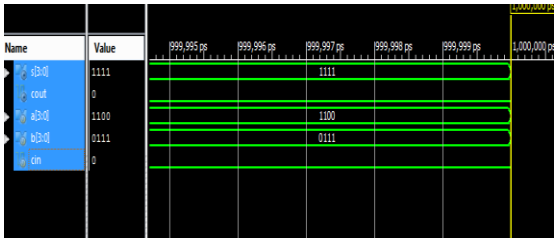


Fig 4.6AFA 1221

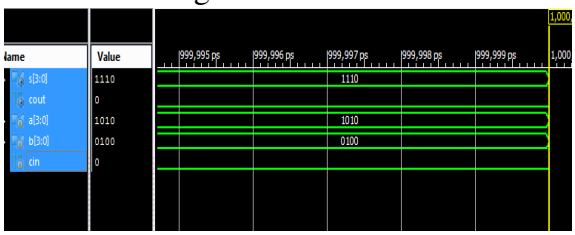


Fig 4.7AFA 2112

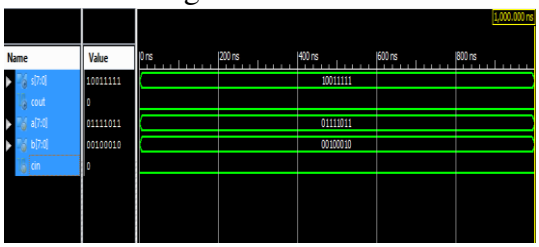


Fig 4.8AFA12122121

5. Conclusion

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.

References

[1] S. L. Lu, ”Speeding up processing with approximation circuits,” Computer, vol. 37, no. 3, pp. 67-73, 2004.

[2] J. Han and M. Orshansky, ”Approximate computing: an emerging paradigm for energy-efficient design,” in Proc. European Test Symposium, pp. 1-6, 2013

[3] S. Hashemi, R. Bahar, and S. Reda, ”DRUM: A Dynamic Range Unbiased Multiplier for Approximate Applications,” in Proc. IEEE/ACM International Conference on Computer Design, pp. 418 - 425, 2015.

[4] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, ”Low-power digital signal processing using approximate adders,” IEEE Trans Computer-Aided Design of Integrated Circuits and Systems, vol. 32, pp. 124-137, 2013.

[5] S. Rehman, W. El-Harouni, M. Shafique, A. Kumar, and J. Henkel, ”Architectural-space exploration of approximate multipliers,” in Proc. Int. Conf. Computer-Aided Design, pp. 1-6, 2016.

[6] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, ”Design and analysis of approximate compressors for multiplication,” IEEE Transactions on Computers, vol. 64, no. 4, pp. 984 - 994, 2014.

[7] J. Liang, J. Han, and F. Lombardi, ”New metrics for the reliability of approximate and probabilistic adders,” IEEE Trans Computers, vol. 62, no. 9, pp. 1760-1771, 2013.

[8] 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.

[9] C. Lent and P. Tougaw, ”A device architecture for computing with quantum dots,” in Proc. IEEE, vol. 85, no. 4, pp. 541-557, 1997.

[10] M. Vacca, M. Graziano, J. Wang, F. Cairo, G. Causapruno, G. Urgese, A. Biroli, and M. Zamboni, ”Nanomagnet logic: an architectural level overview,” Lecture Notes in Computer Science, pp. 223-256, 2014.
[11] 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.
[12] C. Labrado, H. Thapliyal and F. Lombardi, ”Design of majority logic based approximate arithmetic circuits,” in Proc. IEEE Int. Symp. Circuits and Systems, pp. 2122-2125, 2017.
[13] T. Zhang, W. Liu, E. McLarnon, M. O’Neill and F. Lombardi, ”Design of majority logic (ML) based approximate full adders,” in Proc. IEEE Int. Symp. Circuits and Systems, 2018.
[14] M. H. Moaiyeri, F. Sabetzadeh, and S. Angizi, ”An efficient majority-based compressor for approximate computing in the nano era,” Microsystem Technologies, vol. 4, pp. 1-13, 2017.
[15] S. Angizi, H. Jiang, R. F. DeMara, J. Han and D. Fan, ”Majoritybased spin-CMOS primitives for approximate computing,” IEEE Trans Nanotechnology, vol. 17, no. 4, pp. 795-806, 2018.
[16] C. Chang, J. Gu, and M. Zhang, ”Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits,” IEEE Trans Circuits & Systems I, vol. 51, no. 10, pp. 1985-1997, 2004.
[17] V. Pudi, K. Sridharan, and F. Lombardi, ”Majority logic formulations for parallel adder designs at reduced delay and circuit complexity,” IEEE Trans Computers, vol. 66, no. 10, pp. 1824-1830, 2017.
[18] H. Cho and E. E. Swartzlander, ”Adder and multiplier designs in quantum dot cellular automata,” IEEE Trans Computers, vol. 58, no. 6, pp. 721-727, 2009.
[19] V. Pudi and K. Sridharan, ”Low complexity design of ripple carry and Brent-Kung adders in QCA,” IEEE Trans Nanotechnology, vol. 11, no. 1, pp. 105-119, 2012.
[20] S. Mittal, ”A survey of techniques for approximate computing,” ACM Computing Surveys, vol. 48, no. 4, pp. 62, 2016.
[21] S. L. Eddins and M. T. Orchard, ”Using MATLAB and C in an image processing lab course,” Proc. of 1st International Conference on Image Processing, vol. 1, pp. 515-519, 1994.
[22] P. Kulkarni, P. Gupta, and M. Ercegovac, ”Trading accuracy for power with an underdesigned multiplier architecture,” Proc. VLSI Design, pp. 346-351, 2011.
[23] S. W. Kim and E. E. Swartzlander, ”Multipliers with coplanar crossings for quantum-dot cellular automata,” Proc. 10th IEEE Conference on Nanotechnology, pp. 953-957, 2010.
[24] S. W. Kim and E. E. Swartzlander, ”Parallel multipliers for quantum-dot cellular automata,” Proc. Nanotechnology Materials and Devices Conference, pp. 68-72, 2009.
[25] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han and F. Lombardi, ”Design of approximate Radix-4 Booth multipliers for error-tolerant computing,” IEEE Transactions on Computers, vol. 66, no. 8, pp. 105- 119, 2015