International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 08 | Aug 2020 www.irjet.net p-ISSN: 2395-0072
National Conference on Recent Advancements in Communication, Electronics and Signal Processing-RACES'20
Organised by: Department of ECE, Velammal Engineering College, Chennai-66

MAJORITY LOGIC BASED APPROXIMATE ADDERS AND MULTIPLIERS FOR ERROR TOLERANT APPLICATIONS

V.MAGESH[1], S.HARINI[2], B.R.HARINI PRIYA[3]

[1]Assistant Professor -II, Department of Electronics And Communication, Velammal Engineering College, affiliated to Anna University, Surapet, Chennai, Tamil Nadu, India.
[2]Research Scholar, Department of Electronics And Communication, Velammal Engineering College, affiliated to Anna University, Surapet, Chennai, Tamil Nadu, India.
[3]Research Scholar, Department of Electronics And Communication, Velammal Engineering College, affiliated to Anna University, Surapet, Chennai, Tamil Nadu, India.

Abstract: Approximate computing is one of them and in recent years, has attracted the strongest attention of the scientific community. Approximate computing exploits inherent error resilience of applications and features high-performance energy-efficient software and hardware implementations by trading-off computational quality for computational efforts. This paper proposes a Approximate Adder using Majority logic. This paper also proposes the Wallace tree multiplier using proposed majority based Full adder. The Design is simulated and synthesized using Modelism and Xilinx ISE. The propped Wallace tree multiplier is implemented in Spartan 6 FPGA and results shows that the proposed system consumed less area compare with regular Wallace tree multiplier.

Keywords: Approximate adders, multiplier Wallace tree, Majority logic FPGA.

Introduction

Approximate Adders for approximate multiplication Document The gap between capabilities of CMOS technology scaling and requirements of future application workloads is the so-called normalized error distance) can meet with respect to circuit-based figures of merit of a design (number of transistors, delay and power consumption). Four different schemes for utilizing the proposed approximate compressors are proposed and analyzed for a Dadda multiplier [2]. Extensive simulation results are provided and an application of the approximate multipliers to image processing is presented. The results show that the proposed designs accomplish significant reductions in power dissipation, delay and transistor count compared to an exact design; moreover, two of the proposed multiplier designs provide excellent capabilities for image multiplication with respect to average normalized error distance and peak signal-to-noise ratio (more than 50dB for the considered image examples).The paper is organized as follows: Section 2 reviews related works and error metrics. Designs of ML based approximate full adders increasing rapidly. There are several promising design approaches that jointly can reduce this gap significantly. Approximate computing is an attractive paradigm for digital processing at anemometric scales. Inexact computing is particularly interesting for computer arithmetic designs. The analysis and design of two new approximate 4-2 compressors are presented in Section 3. Section 4 presents the design of Wallace tree approximate multipliers proposed approximate adders. Section 5 Discusses the results obtained using Xilinx and moelsim. Section 6 concludes this paper.

Related Works

As one of the main obstacles to attain high are explained in [2] for utilization in a multiplier. These designs rely on different features of compression, such that performance, power dissipation is increasingly been investigated for imprecision in computation (as measured by the error rate and IC design. Approximate computing is a

promising technique to reduce power consumption and improve performance of circuits and systems by introducing computational errors for 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 fixed point and floating-point formats have been proposed [3-6]. Error metrics such as the mean error distance (MED), the normalized MED (NMED) and the relative MED (RMED) [7] have been proposed to analyze the errors introduced in the operations of approximate arithmetic circuits. As CMOS is approaching its technology limitations, emerging nanotechnologies have been proposed at the end of the so-called Moore's Law, such as Quantum-dot Cellular Automata (QCA) [8-9], nanomagnetic logic (NML) [10], and spin-wave devices (SWD) [11]. All of these technologies rely on majority logic (ML) as digital design framework; this is different from

conventional Boolean logic. The majority gate performs a multi-input logic operation (Fig. 1); the logic expression of the 3-input majority gate (voter, MV) is given by equation 1

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

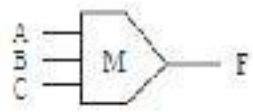


Figure 1: Majority Gate

It is expected that significant improvement in power consumption could be achieved by applying approximate computing also to emerging nanotechnologies. However, approximate designs of CMOS circuits cannot be directly applied due to the underlying different logic; few designs of ML based approximate circuits have been studied [12-15].

[12] has proposed a 1-bit approximate full adder (AFA), but no multi-bit designs suitable for practical applications have been outlined. Several ML-based AFAs have been proposed in [13]; 1-bit as well as multi-bit AFAs are considered. For an approximate multiplier (AM), [14] has proposed a 4:2 approximate compressor based on truth table manipulation for designing an approximate multiplier for image processing. [15] has proposed two 4:2 approximate compressors based on the 1-bit AFA of [12]. In this paper, both ML-based AFAs (MLAFAs), and ML based AMs (MLAMs) are designed. For the MLAFA, multibit MLAFAs are designed by combining 1-bit MLAFAs. Moreover, a 2-bit MLAFA with a higher accuracy is designed based on logic reduction. For the MLAM, a 2 _ 2 design with complement bits is proposed. Furthermore, complement bit selection is analyzed as function of the size of a multiplier; a so-called influence factor is introduced to assess the importance of different complement bits. Few ML-based approximate compressors (MLACs) are designed by MLAFAs or K-Map simplification; then they are employed in the reduction circuitry. Error analysis and a hardware evaluation are presented to validate the proposed designs. Case studies with the proposed approximate adders and multipliers for image processing are also provided as part of this assessment.

Majority Logic Based Approximate adders

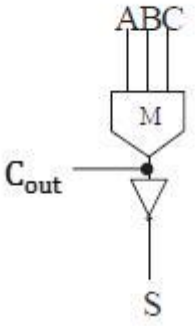
In this paper, both ML-based AFAs (MLAFAs), and MLbased AMs (MLAMs) are designed. For the MLAFA, multibit MLAFAs are designed by combining 1-bit MLAFAs. Moreover, a 2-bit MLAFA with a higher accuracy is designed based on logic reduction. For the MLAM, a 2 _ 2 design with complement bits is proposed. Furthermore, complement bit selection is analyzed as function of the size of a multiplier; a so-called influence factor is introduced to assess the importance of different complement bits. Few ML-based approximate compressors (MLACs) are designed by MLAFAs or K-Map simplification; then they are employed in the reduction circuitry. Error analysis and a hardware evaluation are presented to validate the proposed designs. Case studies with the proposed approximate adders and multipliers for image processing are also provided as part of this assessment.

Proposed ML-based Approximate Designs

ML-based Approximate Full Adder A 1-bit MLAFA (MLAFA1) has been proposed in [12] (Fig. 2). The inputs are given by A, B, C while S and Cout are the outputs. MLAFA1 generates the output S as the complement of Cout; it introduces 2 errors (among the 8 input combinations) when computing the output S (Table 1), but saving two majority gates and one inverter. The circled entries in the truth table denote the instances in which the outputs of MLAFA differ from the exact full adder (EFA). The equations for the carry out and the sum are as follows:

Cout = M (A; B; C)

S = Cout



Wallace Tree multiplier using ML-based Approximate Full adder(MLFA)

The figure 3 shows the Wallace Tree multiplier using ML based Full adder. The regular full adders used in the LSB part of the Multiplier is replaced by proposed approximate full adders. Which significantly reduce the hardware area of Wallace tree multiplier.

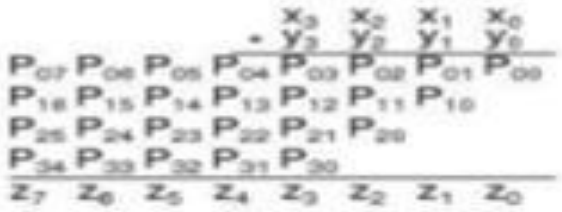
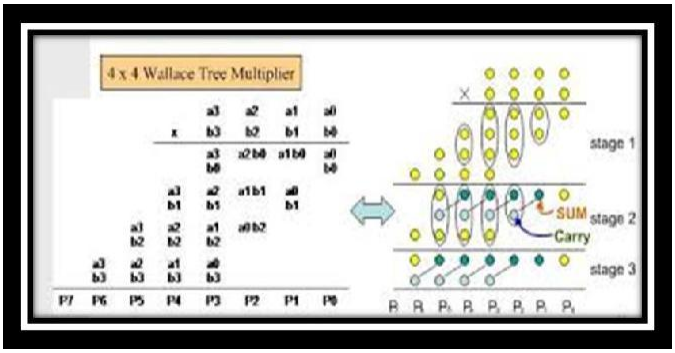


Figure 2: Full adder Gate

Walace tree multiplier

Multiplication is one of the most widely used arithmetic operations. Due to this a wide range of multiplier architectures are reported in the literature providing flexible choices for various applications. Among them the simplest is array multiplier [1] which is also the slowest. Some high performance multipliers are presented in [2–5]. The focus of this paper is Wallace multiplier [6]. Wallace multiplier uses full adders and half adders to reduce the partial product tree to two rows, and then a final adder is used to add these two rows of partial products.



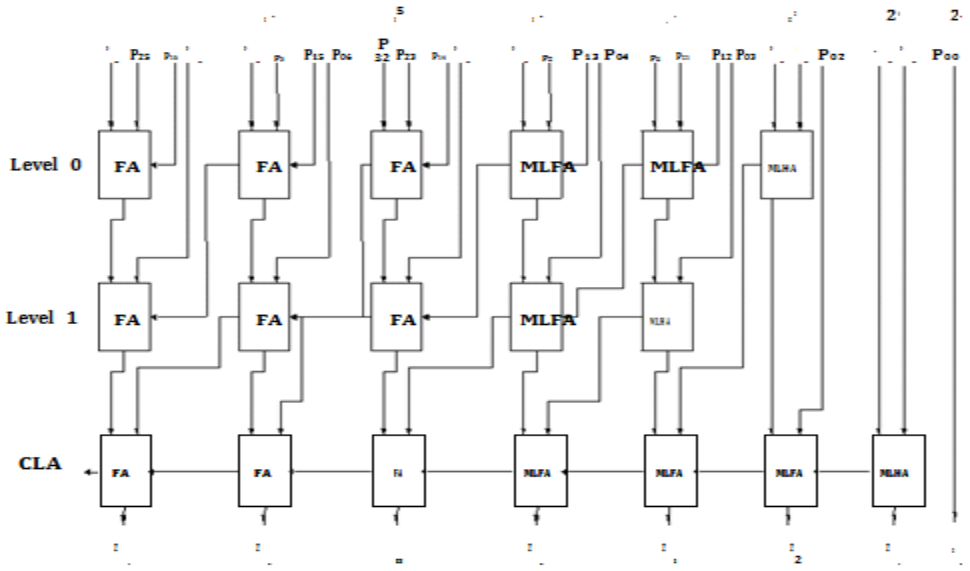


Figure 4: Proposed Wallace tree multiplier

Results

Proposed majority based full adder is designed using verilog programming and this module is used to construct the Wallace tree multiplier using the verilog programming. The functionality of wallace tree multiplier is verified using Modelsim simulator. The simulation result shown in figure 5.

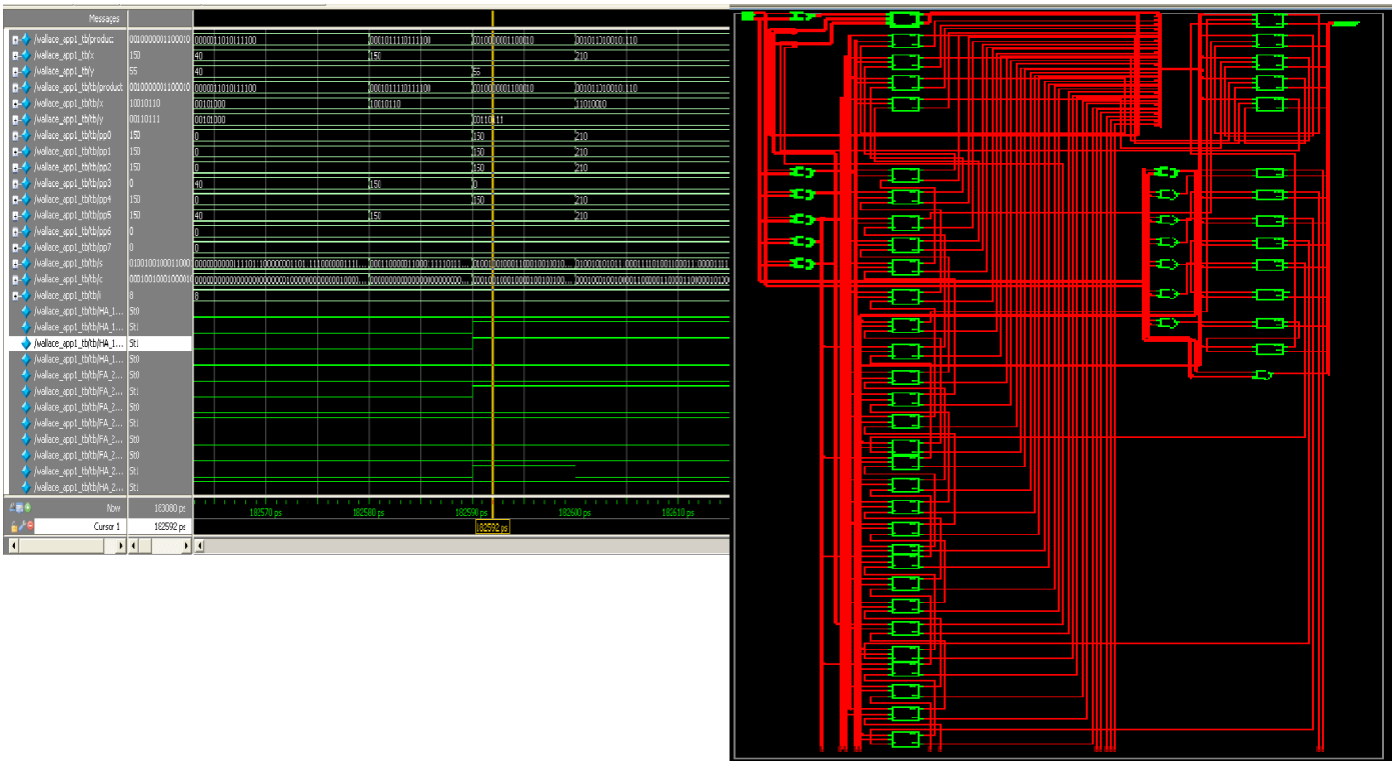


Figure 5: Wallace tree multiplier

Figure 6 Simulation result of Proposed Wallace tree multiplier

The proposed Wallace tree multiplier is synthesized using Xillinx ISE tool for the FPGA Spartan 6. The figure 5 shows the RTL view of proposed Wallace tree multiplier.

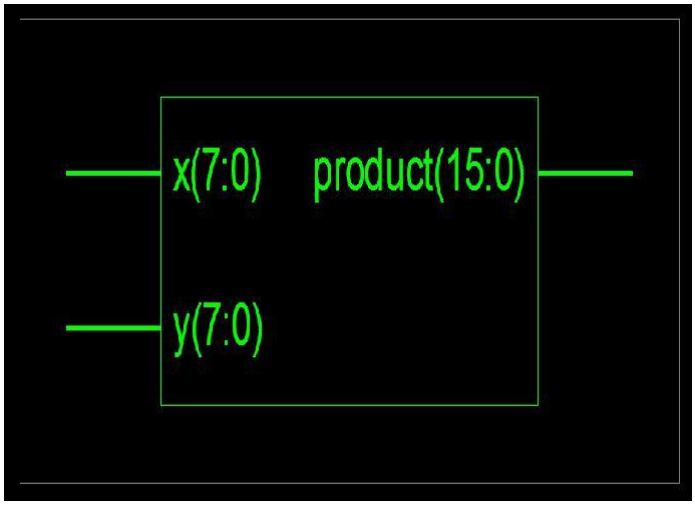


Figure 7: RTL View of Proposed Wallace tree multiplier

The table shows the device utilization Summary of both Existing and proposed Wallace tree multiplier

Table 1: Comparison of existing and proposed system

Parameters	Number of Slices	Number of 4 input LUTs	Number of IOs	Number of bonded IOBs	Maximum combination path delay
Existing wallace tree	74 out of 4656 1%	129 out of 9312 1%	32	32 out of 190 16%	26.948ns
Proposed system	65 out of 4656 1%	115 out of 9312 1%	32	32 out of 190 16%	26.324ns

Discussion

This Results in the design, analysis and evaluation of majority logic based approximate adders and approximate multipliers.

ML based 1-bit, 2-bit and multibit 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 KMap simplification, and has been employed in the approximate PPR circuitry design for 8 _ 8 MLAMs.

Conclusion

Approximate multipliers are widely being advocated for energy-efficient computing in applications that exhibit an inherent tolerance to inaccuracy. However, the inclusion of accuracy as a key design parameter, besides the performance, area and power, makes the identification of the most suitable approximate multiplier quite challenging. This paper proposes a Approximate Adder using Majority logic. This paper also proposes the Wallace tree multiplier using proposed majority based Full adder. The Design is simulated and synthesized using Modalism and Xilinx ISE. The propped Wallace tree multiplier is implemented in Spartan 6 FPGA and results shows that the proposed system consumed 14% less area compare with regular Wallace tree multiplier.

The propagation delay is decreased from 26.948ns to 26.324ns.

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: anemerging paradigm for energy-efficient design," inProc. European Test Symposium, pp. 1-6, 2013

[3] K. Walus and G. Jullien, "Design tools for an emergingSoC tech- nology: quantum-dot cellular automata," inProc. IEEE, vol. 94, no. 6, pp. 1225-1244, 2006.

[4] A. Khitun and K. L. Wang,"Nano scale computational architec- tures with spin wave bus," Superlattices andMicrostructures, vol. 38, no. 3, pp. 184-200, 2005.

[5] M. H. Moaiyeri, F. Sabetzadeh, and S. Angizi, "Anefficient majority-based compressor for approximate computing in the nano era," MicrosystemTechnologies, vol. 4, pp. 1-13, 2017.

[6] S. L. Eddins and M. T. Orchard, "Using MATLAB and C in an image processing lab course," Proc. of 1stInternational Conference on Image Processing, vol. 1, pp. 515-519, 1994.

[7] 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.

[8] M. H. Moaiyeri, F. Sabetzadeh, and S. Angizi, "Anefficient majority-based compressor for approximate computing in the nano era," Microsystem Technologies, vol. 4, pp. 1-13, 2017

[9] C. Chang, J. Gu, and M. Zhang, "Ultra low-voltagelow-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.

[10] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han and F. Lombardi,"DesignofapproximateRadix4Boothmultipliersforerror-tolerant computing," IEEE Transactions on Computers, vol. 66, no. 8, pp. 105- 119, 2015.