Approximate Multipliers using Decoder based logic

Literature Survey

Numerous approximate multipliers have been suggested based on procedures to the complexity of the computation. Jiang et al. [1] proposed a broken-array multiplier by reducing the carry save adder which are utilized for the accumulation stage. By removing either the PP rows or the PP columns, this strategy was able to save space. Yang et al. [2] presented a 4-2 compressor with the goal of achieving a low error rate. This technique aims to take advantage of the AND gate output's probabilistic properties, which are exploitedin the GPP stage at the RPP stage. By modifying Yang's compressor design and incorporating a straightforward error correcting circuit into the RPP stage, Ha and Lee were able to increase accuracy. An effective compressor was suggested by Venkatachalam and Ko [3] to get minimal area and optimized power.To lower the error rate, Yi et al. [4] suggested the compressor for approximation. The Tree Multiplier using Wallace Tree in mode4 with prediction of carry-in for creation of the recursive blocks was introduced by Bhardwaj et al. [5] This design is based on the approach of using smaller, less accurate sub-blocks for the Least Significant Bits (LSB) of the computation, while more accurate multiplier modules are used for the Most Significant Bits (MSB). The carry-in prediction is used to improve the accuracy and performance ofthe recursive blocks. To achieve low-power, Hashemi et al. [6] used a rapid bit selection approach to generate approximate inputs for error-resistant applications.

P. Mahendra and S. R. Ramesh in a titled “FPGA Implementation of High Performance Precise Signed and Unsigned Multiplier using Ternary 6-LUT Architecture” [7] Approximation multipliers which takes LUT6 tables , chain of carry and sum see in recent FPGAs. The proposed exact signed and unsigned multipliers which lowers the total number of lookup tables by 63%. They have developed a design of efficient adder for the output of sub multipliers. As structure of LUT6 is accessible in FPGA. A. Kumar, A. Ansari, A. Srivastava, and K. Suneja, "Fast Approximate Matrix Multiplier based on Dadda Reduction and Carry Save Ahead Adder," In this paper authors have proposed [8] the Dadda compressor with Carry-Save Ahead adder which consumes less chip area and power with less delay. As a result, the proposed architecture delay shows a linearity with doubling the operand size. There is a trade off with the speed of matrix multiplier. G. Thakur, H. Sohal, and S. Jain, "Implementation of Approximate Multiplier using Inexact Compressors" have proposed [9] high energy efficient 9_2 higher order compressor for complexity reduction of circuit and increasing multiplication block. The proposed multiplier consumes less power as 43.93% and utilization of area 49.52% less as compared to exact multiplier. L. H. Krishna, J. B. Rao, S. Ayesha, S. Veeramachaneni, and S. Noor Mahammad, "Energy Efficient Approximate Multiplier Design for Image/Video Processing Applications" proposed method [10] which provides average 15% energy improvement. The authors have proposed two variants, first with only approximate compressors and second with approximate compressors and exact compressors. As result the second design proposed saves energy by 10.8% as compared to existingdesigns with trade off 18.75% error rate. K. Naresh, Y. P. Sai, and S. Majumdar, "Design of 8-bit Dadda Multiplier using Gate Level Approximate 4:2 Compressor". The authors have proposed [12] the Dadda multiplier which reduces large and costly compressors. Four cases of 8bit Dadda multiplier with gate level approximation technique proposed wherein each case have low error rate to get fewer affected function with low area and power dissipation. The outcome accuracy rates for four cases are 92%,87.5%,85,5%,78%. The proposed design gets the delay gain and power of 35% and 60% respectively.

H. Zhang, H. Xiao, H. Qu, and S. -B. Ko in a titled “FPGA-Based Approximate Multiplier for Efficient Neural Computation” [11] It is based on the Karatsuba algorithm with radix-8 Booth’s multiplier used for each small multiplication. Two variants of the approximate designs have been proposed. Implementation of results displays significant area and energy optimization in neural network computation with less compromise with accuracy, The approximate design methods will be applied and evaluated and other machine learning algorithms will be considered. M. K. Sukla, K. Sethi and A. K. Panda in paper titled “Low Power and Area Efficient 8-bit Approximate Multiplier (and 8-bit Wallace tree multiplier) with Reduced Partial Product” [12] have proposed area and power efficient architecture for error tolerant multiplier. A 45 nm process node and Cadence GENUS synthesis tool in legacy mode is adopted for logic implementation. Improvement in power and area consumption is achieved up to 29.29% as compared to Wallace tree multiplier in static mode and dynamic power up to 33.35% and area begin saved up to 25.7 % due to reduction in product method with trade off with probability error (PE) of 50% by the approximation for the lower stage of partial product. F. -Y. Gu, I. -C. Lin and J. -W. Lin, "A Low-Power and High-Accuracy Approximate Multiplier with Reconfiguration Truncation, "In this paper the authors have proposed [13] the an approximate 4_2 compressors with high accuracy which can also dynamically truncate partial products to achieve variable accuracy requirements. For reduction of error distance, a simple error compensation circuit is also begin proposed. As a result, there is reduction in delay and average power consumption by 27% and 40.33% to 72 % respectively as compared with Wallace tree multiplier. The optimal formulas to choose the algorithm for partitioning Trunk signals is not yet found by authors. T. Kong and S. Li, "Design andAnalysis of Approximate 4–2 Compressors for High-Accuracy Multipliers," The authors have proposed [14] five high accuracies approximate 4_2 compressors with better delay, area, power, and better performance. Where four of these compressors (Pro1-Pro4) are designed by using decomposition. All compressors are begin implemented in TSMC 28nm CMOS process. And recombination methods for critical path delay, area,and power consumption optimization. The fifth (Pro5) is designed by optimizing the sorting technique with best ADP (Area Delay Product)-accuracy trade off. As a result, there is reduction of delay by 18% and 43%- 52% ADP reduction as compared to exact multiplier.

A. Gupta and K. Suneja in a paper titled “Hardware Design of Approximate Matrix Multiplier based on FPGA in Verilog” [15], the approximate matrix multiplier structure balances performance and precision. It demonstrates how computing might be approximated to bring about a major improvement in the matrix procedure by adding a tiny amount of inaccuracy to multiplication. Because the degree of approximation can be adjusted as needed, techniques for error prediction that can be helpful in approximation computing are used in a variety of fields, including data encryption and signal transmission. H. Waris, C. Wang, W. Liuand F. Lombardi in a paper titled “AxBMs: Approximate Radix-8 Booth Multipliers for High-Performance FPGA-Based Accelerators” [16] Based on approximate booth encoders approximate results in a smaller hardware footprint for booth encoded signals and a partial product generator architecture which is more effectively mapped a 6 input LUT. The LUT proposed multipliers increase the resolution and have improvedlatency over the previous FPGA-targeted architecture and energy savings of the edges are detected. The benefit of complementing errors in AxBM2 has been merged with a reduction to produce savings of up to

60%. Bao Fang, Huaguo Liang, Dawen Xu, Maoxiang Yi, Yongxia Sheng, Cuiyun Jiang, Zhengfeng Huang, Yingchun Lu,“Approximate multipliers based on a novel unbiased approximate 4-2 compressor". Theauthors have proposed [17] unbiased approximate 4_2 compressor generating positive and negative signed errors in balance, which results in optimization of errors. Compared to two proposed designs it has been observed that Unbiased Approximate Multiplier (UBAM)-M1 is more accurate, Moreover UBAM-M2 reduces delay and power consumption by 22% and 28% respectively. It also reduces the area and power delay product by almost 39% and 46% respectively. The UBAM-M2 also observes the reduction in chip area by 26%. The authors have also evaluated the design in image filtering wherein it has been observed thatUBAM-M1 is superior in terms of accuracy. As a result, UBAM-M2 has 58.12 % PDP saving with trade off 1.24% SSIM degradation. K. Naresh, S. Majumdar, Y. P. Sai and P. R. Sai, "Efficient Design of Artificial Neural Networks (ANN) using Approximate Compressors and Multipliers," In this design the authors a proposed [18] design using the approximate multiplier for implementation of Artificial neural network. The interfacing has been done between approximate arithmetic circuits,3:2 and 4:2 compressors are designed with unique error positions. Result simulation is begin carried out by synopsys design compiler tool at 90nmTechnology node. The proposed design gains a delay constraint of 3 to 30%. Also, ANN has power, areaand delay constraints of 18.7%,20.31% and 10% respectively.

References

[1] Jiang, Honglan & Han, Jie & Lombardi, Fabrizio. (2015). A Comparative Review and Evaluation of Approximate Adders. 343-348. 10.1145/2742060.2743760.

[2] Z. Yang, J. Han, and F. Lombardi, “Approximate compressors for error resilient multiplier design,” in Proc. IEEE Int. Symp. DFTS, 2015,pp. 183–186.

[3] S. Venkatachalam and S. B. Ko, “Design of power and area efficient approximate multipliers,” IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 25, no. 5, pp. 1782-1786, 2017.

[4] X. Yi, H. Pei, Z. Zhang, H. Zhou, and Y. He, “Design of an energy-efficient approximate compressor for error-resilient multiplications,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), 2019, pp. 1–5.

[5]K. Bhardwaj, P. S. Mane, and J. Henkel, “Power-and area-efficient approximate wallace tree multiplier for error-resilient systems,” in Proc. 15th Int. Symp. Qual. Electron. Design, 2014, pp. 263–269.

[6] S. Hashemi, R. I. Bahar, and S. Reda, “DRUM: A dynamic range unbiased multiplier for approximate applications,” in Proc. IEEE/ACM Int.Conf. Comput. Aided Design (ICCAD), 2015, pp. 418–425.

[7]P. Mahendra and S. R. Ramesh in a titled “FPGA Implementation of High Performance Precise Signedand Unsigned Multiplier using Ternary 6-LUT Architecture”

[8] Kumar, Arun & Ansari, Asif & Srivastava, Ayush & Suneja, Kriti. (2021). Fast Approximate Matrix Multiplier based on Dadda Reduction and Carry Save Ahead Adder. 1058-1061. 10.1109/ICACCS51430.2021.9441960.

[9] G. Thakur, H.Sohal, and S. Jain, "Implementation of Approximate Multiplier using Inexact Compressors"

[10] L. H. Krishna, J. B. Rao, S. Ayesha, S. Veeramachaneni, and S. Noor Mahammad, "Energy Efficient Approximate Multiplier Design for Image/Video Processing Applications"

[11] H. Zhang, H. Xiao, H. Qu, and S. -B. Ko in a titled “FPGA-Based Approximate Multiplier for Efficient Neural Computation”

[12] K. Naresh, Y. P. Sai, and S. Majumdar, "Design of 8-bit Dadda Multiplier using Gate Level Approximate 4:2 Compressor”.

[13] F. -Y. Gu, I. -C. Lin and J. -W. Lin, "A Low-Power and High-Accuracy Approximate Multiplier With Reconfiguration Truncation,"

[14] T. Kong and S. Li, "Design and Analysis of Approximate 4–2 Compressors for High-Accuracy Multipliers,"

[15] A. Gupta and K. Suneja in a paper titled "Hardware Design of Approximate Matrix Multiplier based on FPGA in Verilog"

[16] H. Waris, C. Wang, W. Liu, and F. Lombardi in a paper titled "AxBMs: Approximate Radix-8 Booth Multipliers for High-Performance FPGA-Based Accelerators"

[17] Bao Fang, Huaguo Liang, Dawen Xu, Maoxiang Yi, Yongxia Sheng, Cuiyun Jiang, Zheng feng Huang, Ying chun Lu, "Approximate multipliers based on a novel unbiased approximate 4-2 compressor".

[18] K. Naresh, S. Majumdar, Y. P. Sai and P. R. Sai, "Efficient Design of Artificial Neural Networks using approximate using Approximate Compressors and Multipliers,"