

A Low-Power High-Accuracy Approximate Multiplier Using High-Order Approximate Compressors

Banoth Ashwini

*Electronics and Communication Engineering
Institute of Aeronautical Engineering
Hyderabad, Telangana, India
banothashwini9980@gmail.com*

Vyasa Ranjith Kumar

*Electronics and Communication Engineering
Institute of Aeronautical Engineering
Hyderabad, Telangana, India
ranjithkumarvyasa@gmail.com*

Nallawar Amulya

*Electronics and Communication Engineering
Institute of Aeronautical Engineering
Hyderabad, Telangana, India
nallawaramulya@gmail.com*

Dr.P.Munaswamy, Professor and Head

*Electronics and Communication Engineering
Institute of Aeronautical Engineering
Hyderabad, Telangana, India
p.munaswamy@iare.ac.in*

Abstract—To address the need to reduce power consumption, approximate multipliers have emerged as a potential solution for fault-tolerant applications. In this work, we present a new 8x8 approximate multiplier that focuses on minimizing performance while maintaining a high degree of accuracy. The design features two key features: firstly, based on their importance, different weights are handled by the compressors with different levels of precision, allowing for a trade-off between energy efficiency and minimum error. Second, higher order approximation compressors such as 8:2 compressors are used for intermediate weights to simplify the drive chain logic. This is, to our knowledge, the first design to successfully integrate higher-order approximate compressors into an approximate multiplier. Compared to a precision multiplier such as the Dadda tree multiplier, experimental results show that the proposed design offers significant energy savings while maintaining high accuracy.

Key Words:- Approximate computing; Arithmetic circuits; Logic design; Low-power design; Partial Product reduction

I. INTRODUCTION

Multipliers are an essential part of modern digital signal processing and various other applications. [1,2] With technological advances, researchers have focused on designing multipliers that meet specific goals such as high speed, low power consumption, and compact layout, while some designs aim to balance all these aspects for efficient VLSI implementations. A widely used multiplication technique is the add-and-shift algorithm. In parallel multipliers, the number of sub-products is a critical factor affecting performance. [3,5] The Modified Booth algorithm is commonly used to reduce the number of sub products. To further increase the speed, the Wallace Tree algorithm can be used, which minimizes the number of stages of sequential addition. [12] By combining a modified algorithm and the Wallace Tree technique, it

is possible to utilize the strengths of both approaches in a single multiplier design. However, increased parallelism can lead to more drift between sub products and subtotals, which can cause reduced speed, increased silicon area due to structural irregularities, and higher power consumption from complex routing and increased interconnects. Series-parallel multipliers, on the other hand, trade speed for better power efficiency and area utilization. Whether to choose a parallel or serial multiplier depends largely on the requirements of the particular application. This discussion introduces key multiplication algorithms and architectures and compares them across metrics such as speed, area, performance, and combinations thereof. Binary multiplication works similarly to digit-based multiplication, where partial products are generated and then added using adders such as half adders and full adders.

EXISTING SYSTEM

[3,5] The architecture of the proposed 8x8 approximation multiplier features two key features: Significance-driven logical compression: In this approach, different weights are assigned different levels of compression precision based on their significance.[9] Higher-end scales use 4:2 precision compressors to ensure accuracy, while mid-level scales use near-precision compressors.[3] For lower values, less accurate compressors are used. This method helps to reduce power consumption while keeping the resulting error to a minimum. Approximate higher-order compression: Higher-order compressors are used for specific weights, especially in the mid-significance range, to help streamline the drive chain logic. This approach further helps reduce overall power consumption without sacrificing too much accuracy.

DEMERITS OF EXISTING SYSTEM

1. Previous compressors generate non-zero outputs for zero inputs, increasing the mean relative error (MRE), but the proposed design addresses this for greater accuracy. 2. A static segment multiplier (SSM) performs $m \times m$ multiplications using segments from n-bit operands, while a partial product perforation multiplier (PPP) omits certain partial products for efficiency. In addition, an approximate 2×2 multiplier based on modifications of Karnaugh maps is used to construct larger multipliers

II. PROPOSED SYSTEM

The proposed multipliers outperform existing designs in terms of area, power consumption, and error metrics while achieving higher peak signal-to-noise ratio (PSNR) values in image processing applications. The error distance (ED) is defined as the arithmetic difference between the correct and approximate output for a given input. In previous studies, approximate adders have been evaluated using the normalized error distance (NED), a nearly invariant metric independent of the size of the approximate multiplier circuit. In addition, traditional mean relative error (MRE) analysis is calculated for both existing and proposed multiplier designs. The remainder of this brief is structured as follows: Chapter 4 presents a detailed description of the proposed architecture, including a comprehensive analysis of the design as well as error metrics for both the proposed and existing approximate multipliers.

A. MERITS OF PROPOSED SYSTEM OVER EXSISTING SYSTEM

The goal is to develop a low power, high-accuracy ap- proximate multiplier that uses higher-order approximate compressors and achieves a balance between energy efficiency, area reduction, and computational accuracy. The goal is to design a multiplier that can be integrated into power-constrained and high-performance applications while maintaining the output quality within accept- able error limits. This approach focuses on optimizing power and area savings without significantly reducing accuracy, making it suitable for applications where energyefficiency is critical but performance must remain robust.

- High accuracy: Even when targeting low performance, it is important to maintain a high degree of accuracy in multiplication results. The use of higher order compressors is key to achieving this balance between energy efficiency and accuracy.
- Reduced power consumption: The primary goal is to significantly reduce the power consumption required for multiplication operations. This is especially important for portable battery devices where energy efficiency is a top priority.
- Area efficiency: Another critical factor is reducing the silicon area required for the multiplier design. By optimizing the design without sacrificing even more performance functionality can be integrated into a limited chip area, increasing overall system efficiency.

B. APPROXIMATE MULTIPLIER USING HIGH ORDER COMPRESSORS

The critical path of a multiplier is often determined bythe maximum height of the partial product matrix (PPM),which creates the need to efficiently compress the PPM. The n:2 compressor is part of a multiplier that, when properly replicated, reduces the n product to two numbers. In each multiplier segment, the compressor takes n:2n bits from agiven position along with one or more carry bits from lower positions (such as i-1) and outputs two bits at positions i and i+1 along with carry bits to higher positions. Traditionally, multipliers use 4:2 compressors. In a 4:2 precision (accurate) compressor, the four input bits are labeled X0, X1, X2, and X3, while the outputs at positions i and i+1 are labeled Sum and Carry. The carry bit from the bottom position is labeledas Cin and the carry bit as Cout. When designing approximate 4:2 compressors, the Cin and Cout carry bits are omitted to simplify the logic and save resources. In addition, to further reduce the error rate, the sum generation and transfer logic is adapted to the traditional design used in precision 4:2 ratio compressors. This reconfiguration optimizes performance by balancing logic complexity and computational accuracy. A. Transfer Access Here we learn about logical transfer access.In a conventional half adder, the carry bit Ch is defined as: $Ch(X0,X1) = X0.X1$ (1) In a conventional full adder, the carrybit Cf is defined as a Carry approach Here, we learn about the approach of thecarry logic. In the conventional half adder, the carry bit Ch is defined as:

$Ch(X0,X1) = X0.X1$ (1) In the conventional full adder, the carry bit Cf is defined as :

$Cf (X0,X1,X2) + Cf (X3,X4,X5) + Ch(X6,X7) + Cf(X0+X1+X2, X3+X4+X5, X6 +X7).$

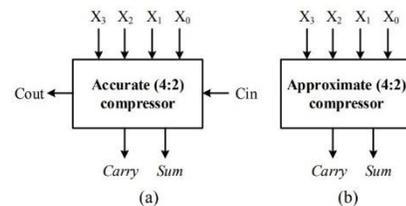


Fig. 1. (a) Accurate 4:2 compressor (b) Approximate 4:2 compressor

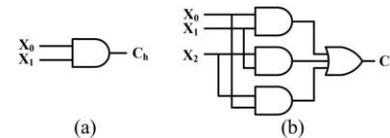


Fig. 2. (a) Modified half adder (b) Modified full adder

a. Approximation of the Sum

In conventional designs, the Sum output is typically generated using a network of XOR gates. However, XOR gates tend to have higher design overhead compared to other logic gates. For example, using the SAED 32nm cell library, a comparison between different gate types such as OR, NOR, XNOR, and XOR shows that XOR gates consume the most

power, require the largest area, and have the highest delay. Given these drawbacks, replacing XOR gates with alternative logic gates could lead to significant design improvements. By using gates with lower power consumption, smaller area requirements, and reduced delay—such as OR or NOR gates—the overall design can achieve better efficiency in terms of power, area, and timing. This approach allows optimization of the sum output logic in approximate multipliers while minimizing the associated overhead. To construct the approximation logic for the Sum output in the high-order approximation compressor, we design a tree of logic gates. In the first stage, an XNOR gate is used instead of an XOR gate because the XNOR gate produces the inverse of the XOR output. A NOR gate is used in the second level and an OR gate is used in the third level to handle the error caused by this change. This approach greatly reduces the design overhead because all the XOR gates they typically consume more power and area, are replaced by these alternative gates.

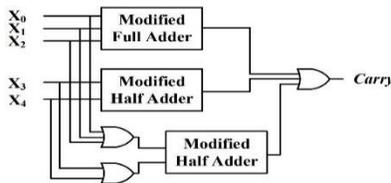


Fig. 3. The logic of Carry output of our approximate 5:2 compressor

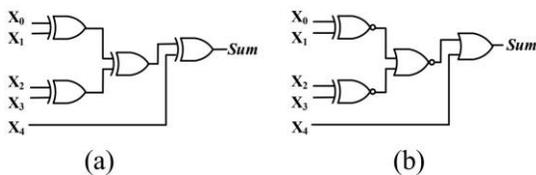


Fig. 4. Sum of 5:2 compressor. (a) Accurate (b) Our approximate

As an example, let's consider the total power of the compressor 5:2. Figure 4(a) shows the sum output logic of a 5:2 precision compressor using conventional XOR gates. In contrast, Fig. 4(b) illustrates the logic for the sum output in an approximate 5:2 compressor, where the XOR gates are replaced by XNOR, NOR, and OR gates, resulting in a more efficient design in terms of power, area, and delay while maintaining acceptable accuracy.

C. PROPOSED APPROXIMATE MULTIPLIER DESIGN

A typical multiplier design consists of three main parts: (1) partial product generation using AND gates, (2) partial product matrix (PPM) reduction using a carry-save addition tree, and (3) computation of the final result using a carry adder. The complexity of the multiplier design is primarily driven by the PPM reduction circuit, which is the focus of many multiplier design optimization efforts. In the proposed 8x8 approximate multiplier design, the PPM reduction process is divided into two stages. The structure of this reduction is based on the significance of the weights, which are divided into three

groups: weights of higher significance, weights of medium significance and weights of lower significance. Designers can adjust the number of weights in each category to balance power consumption and calculation accuracy. Power Reduction Strategy To reduce power consumption while maintaining accuracy, PPM reduction uses a logical significance-driven compression technique: Weights with higher significance use exact 4:2 ratio compressors to ensure accuracy. Weights of moderate importance use approximate higher order compressors. Lower significance weights use a simple OR tree approximation. First stage: Approximation for weights of lower and medium importance

- Lower significant weights: An OR tree approximation is used to save energy. If the number of product terms (n) is 2 or less, no operation is required. If $n > 2$, the OR tree reduces $n - 1$ inputs, leaving at most two product terms per mass after this stage.
- Importance Weights: An n:2 approximate compressor is used, where the number of product terms is n. There are two configurations: Exact Sum and Approximate Conversion Approximate Sum and Approximate Conversion After this stage, each intermediate importance weight has at most two product terms.

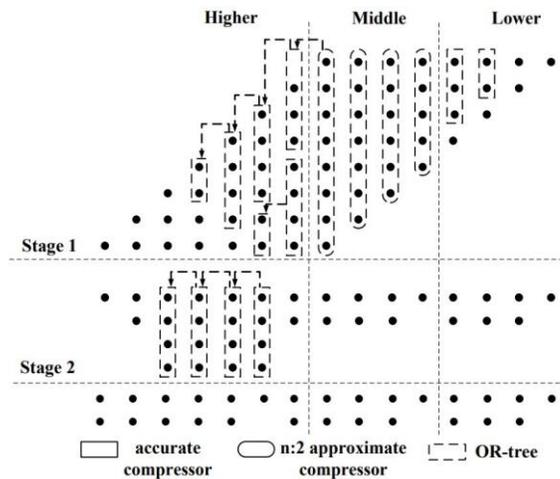


Fig. 5. The PPM reduction in the proposed approximate multiplier

- Higher significant weights: 4:2 precision compressors are used to maintain high accuracy. If less than 4 terms of the product are available, the remaining inputs to the compressor are set to zero. In the case of the highest significance of the rightmost weight, one compressor's Cin carry bit is obtained from the medium significance Carry output, while the other compressor has Cin set to zero.

Phase Two: Reduction for Higher Materialist The second phase focuses exclusively on higher materialist weight.

At this stage, precision 4:2 compressors are used to further reduce the PPM height. The rightmost compressor C_{in} carry bit is set to zero. At the end of the second stage, each higher significance has at most two terms, allowing the carry-spread adder to calculate the final result. This two-stage reduction approach helps to optimize the balance between power efficiency, accuracy and area savings in the multiplier design.

IMPLEMENTATION

Adder: An adder is a digital circuit that performs the addition of numbers. “Sum” and carry - 1. Half adders are digital circuits used for arithmetic operations. br_i Sum = A xor B Carry = A and B
Full Adder: 1. A full adder is a logic circuit that performs the “sum of three units” (A and c).

CARRY SKIP ADDER

The Carry-Skip Adder, also known as the carry bypass adder, enhances the latency of Ripple Carry Adders (RCA) by allowing faster propagation of carry bits. It uses block carry logic with an n -bit ripple chain, an n -input AND gate, and a multiplexer to bypass carry propagation. The propagation signals ($p_i = A_i \oplus B_i$) are combined in the AND gate, with its output controlling the multiplexer. This structure skips over groups of bits when conditions are optimal, reducing delay. For a 4-bit example, the carry C_3 from the last full adder is bypassed directly to the output using skip logic. The design significantly reduces critical path delays in the best case compared to RCAs

CARRY LOOK AHEAD ADDER

The Carry Look-Ahead Adder (CLA) accelerates addition by precomputing carry bits using generation ($G_i = A_i B_i$) and propagation ($P_i = A_i \oplus B_i$) signals. This eliminates sequential carry dependency, as seen in RCAs. CLAs compute each carry in parallel, enabling faster addition. Designs like the Kogge-Stone and Brent-Kung Adders implement this concept for optimized performance. CLA is typically used in modular 4-bit designs to construct larger adders. With minimal latency and efficient use of logic gates, it is suitable for high-speed and low-latency VLSI applications. Its LUT-based architecture ensures compact and efficient design.

In this design, the transfer logic of the fixed bit group adder is simple to two-level logic, which is only a variant of the wave transfer design. This method uses logic gates to represent addition and low-order addition. Elements to see if any major decisions have been made. Let’s talk about it in detail. If we define two variables as follows The first step is to generate G_i and declare P_i , then $P_i = A_i + B_i$ $G_i = A_i B_i$ Steps The next step is as follows: $C_{i+1} = G_i + P_i$ To realize C_i The last step is after processing and output, which can be expressed as G_i is the generation of transfer, whether A_i or B_i is 1, the transfer produces goods. P_i is realized with respect to the expansion transmitted from C_i to C_{i+1} . $G_1 + P_1$ $C_1 = G_1 + P_1$ $G_0 + P_0$ $C_0 = G_0 + P_0$ $C_2 = G_2 + P_2$ $C_3 = G_3 + P_3$ $C_4 = G_4 + P_4$

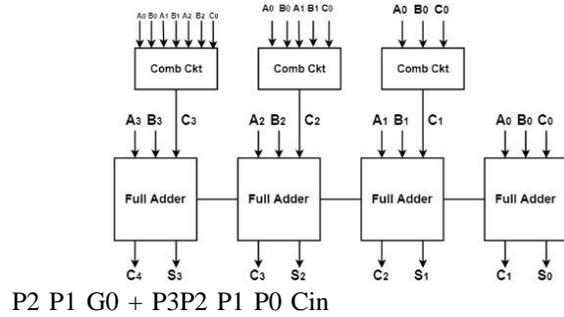


Fig. 6. 4-bit Carry look Ahead adder

From the Bollinger equation above, we see that C_4 does not have to wait for C_3 and C_2 to turn on, but in fact C_4 turns on at the same time as C_3 and C_2 . Since the Boolean expression for each output is a multiplication, it can be implemented with a single-stage AND gate followed by an OR gate. CLA adders are often used as 4-bit modules to build larger adders.

III. RESULTS

A. Existed design result

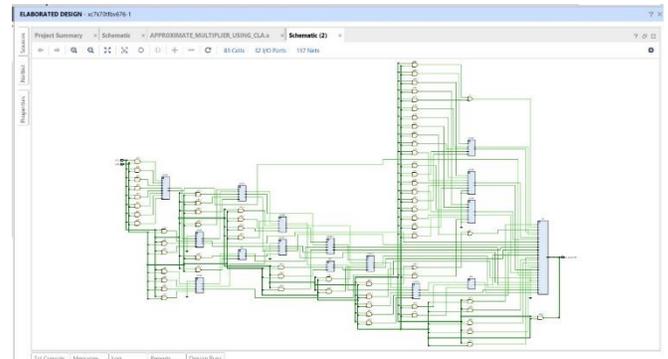


Fig : RTL Schematic of approximate multiplier using carry skip adder

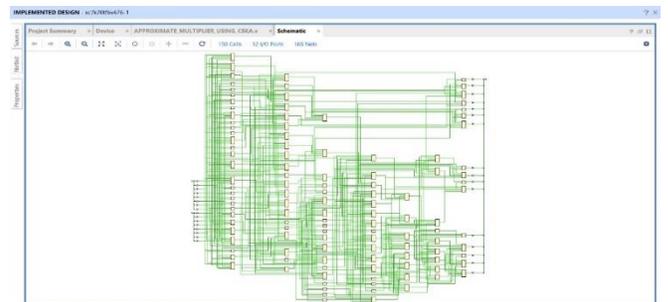


Fig : View Technology Schematic of approximate multiplier using carry skip add

Technical Schematic: The technical schematic represents the architecture in LUT format where the LUT is considered as the area used in VLSI to estimate the architecture design. The LUT is considered as a square unit and the code memory allocation is represented by the LUT in the FPGA.

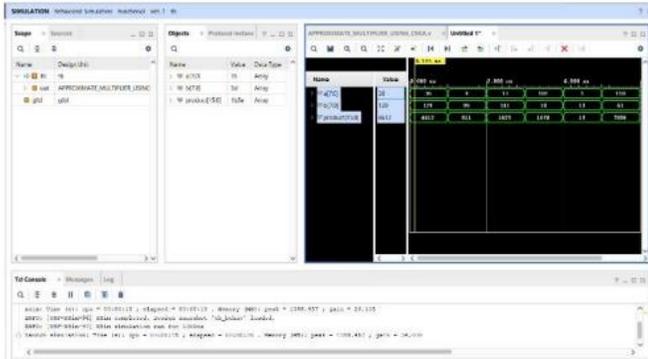


Fig : Simulated Waveforms of approximate multiplier using carry skip adder

B. Simulation

The simulation is the final process of verifying the work and the schematic is the correct wiring and block. When you switch from use to simulation on the main screen of the device, the simulation window will start and the simulation window will limit the output in the output format. Here it is easy to combine different methods. They are also called designer watches because they resemble the designer.

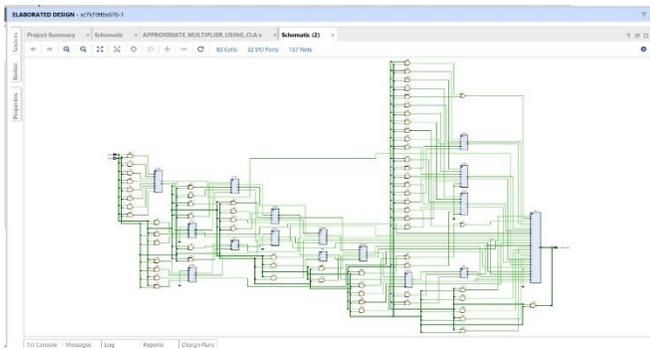


Fig : View Technology Schematic of approximate multiplier using carry look ahead adder

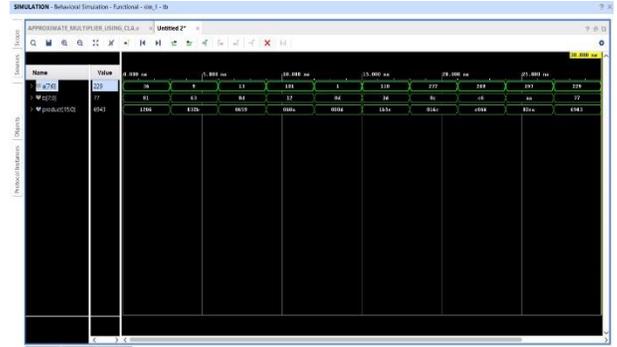


Fig : Simulated Waveforms of approximate multiplier using carry look ahead adder

Technical Schematic: This diagram represents the architecture in LUT format where LUT is considered as the area used to estimate the design in VLSI.

C. Simulation

Simulation is a process which, with regard to its functioning, is called final verification, where the scheme is the verification of connections and blocks. Simulation Simulation on the instrument home screen and in the simulation window limits the output in the form of a waveform output. Here it has the flexibility to provide different radix number systems.

D. EXISTED DESIGN PARAMETERS

Name	Constraint	Status	WNS	TNS	WHS	THS	WSS	TPWS	Total Power	Failed Routes	Methodology	RCA Score	Quit Suggestions	LUT	FF	BRAM	URAM	DSP
WNS_1	WNS_1	Complete	NA	NA	NA	NA	NA	NA	12.178	0				91	0	0	0	0
WNS_2	WNS_2	Complete	NA	NA	NA	NA	NA	NA	11.494	0				83	0	0	0	0

Summary

Power analysis from implemented circuit. Activity detected from constraints files, simulation file or sector's analysis.

On-Chip Power

- Dynamic: 12.080 W (99%)
- Static: 0.88 W (7%)
- Logic: 0.62 W (5%)
- IO: 10.539 W (88%)
- Device Static: 0.110 W (1%)

Total On-Chip Power: 12.178 W

Design Power Budget: Not Specified

Power: 99%

Power Budget Margin: N/A

Junction Temperature: 47.8°C

Thermal Margin: 37.1°C (193.5 W)

Ambient Temperature: 25.0°C

Effective EIR: 1.9°C/W

Power supplied to off-chip devices: 0 W

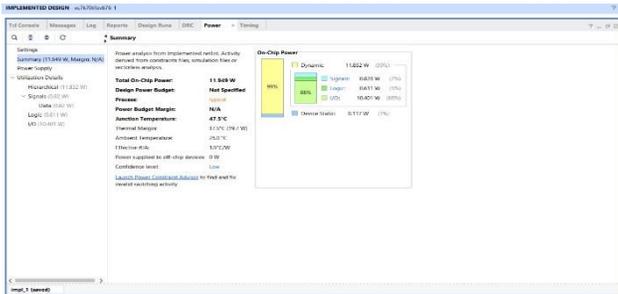
Confidence level: Low

[Launch Power Significance Advisor](#) to find and fix enabled parking activity.

Parameter	Approximate multiplier using CSKA	Approximate multiplier using carry look ahead adder
No .of LUTs	91	83
Power(Watt)	12.178	11.494

Table 1: Parameter comparison

E. PROPOSED DESIGN PARAMETERS



Parameter	Approximate multiplier using CSKA	Approximate multiplier using carry look ahead adder
Power(Watt)	12.178	11.494

Table2: Parameter comparison

PARAMETERS:- Consider the parameters processed in VLSI such as area, delay and power, based on these parameters one architecture can be judged against another. here, area and power consumption are considered, parameters are obtained using XILINX 12.3 and HDL is Verilog language.

IV. CONCLUSION

This project presents a low-power, high-accuracy approximation of the standard 8 x 8 equation using a forward scalar adder. To achieve accuracy, we use a real (ie real) 4:2 compressor in the weight factor. To reduce energy consumption, we use high-order approximation compressors in the middle of the weighing. The experimental results show that the proposed predictive model design can save the energy consumption area and increase the speed compared to the existing predictive model. To the best of our knowledge, this design is the first study to achieve high-resolution predictive compressors in a predictive design to achieve a smaller footprint and lower delay. Future scope: This project proposes a low-power, high-precision 8 x 8 multiplier design. To achieve precision, we use a true (ie, true) 4:2 compressor in mass factor. Low power consumption, small dimensions and high accuracy. Therefore, it finds application in applications with a small area and low consumption. Estimates are computational methods that they return false results more often than their accuracy is

guaranteed and can be used in applications that predict results well enough for their purposes. For security reasons, these copies will be used in future image processing, filtering and encryption applications. An example of this is search engines that may not provide clear answers to some search queries, so multiple answers are possible. Similarly, sometimes a lost frame may not be found in a photo application due to limitations of human perception. The estimate is based on the observation that in most cases, although performing the calculation requires a significant amount of resources, enabling the estimate can yield reliable results in terms of power and strength while achieving accurate results.

V. REFERENCES

- [1] Z. Yang, J. Han, and F. Lombardi, "Approximate Compressor for Error-Resilient Multiplier Design", Proc. of IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, 2015.
- [2] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and Analysis of Approximate Compressors for Multiplication", IEEE Trans. on Computers, vol. 64, no. 4, pp. 984-994, 2015.
- [3] C. Liu, J. Han, and F. Lombardi, "A Low-Power, High-Performance Approximate Multiplier with Configurable Partial Error Recovery", Proc. of IEEE Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014.
- [4] G. Zervakis, et al., "Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation", IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol.24, no.10, pp. 3105- 3117, 2016.
- [5] T. Yang, T. Ukezono, and T. Sato "A Low-Power High-Speed Accuracy-Controllable Approximate Multiplier Design", Proc. Of IEEE Asia and South Pacific Design Automation Conference (ASPDAC), 2018.
- [6] A. Cilaro, et al., "High-Speed Speculative Multipliers Based on Speculative Carry-Save Tree", IEEE Trans. on Circuits and Systems - I, vol. 61, no. 12, pp. 3426-3435, 2014.
- [7] J. Liang, et al., "New Metrics for The Reliability of Approximate and Probabilistic Adders", IEEE Trans. on Computers, vol. 62, no. 9, pp. 1760-1771, 2013.
- [8] P. Kulkarni, P. Gupta, and M. D. Ercegovic, "Trading accuracy for power in a multiplier architecture," J. Low Power Electron., vol. 7, no. 4, pp. 490-501, 2011.
- [9] C.-H. Lin and C. Lin, "High accuracy approximate multiplier with error correction," in Proc. IEEE 31st Int. Conf. Compute. Design, Sep. 2013, pp. 33-38.
- [10] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in Proc. Conf. Exhibit. (DATE), 2014, pp. 1-4.
- [11] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in Proc. IEEE/ACM Int. Conf. Compute.-Aided Design (ICCAD), Oct. 2011, pp. 667-673.

J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Compute., vol. 63, no. 9, pp. 1760-1771, Sep. 2013.