IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AREA EFFICIENT APPROXIMATE MULTIPLIER USING 4: 2 COMPRESSORWITH IMPROVED ACCURACY

1Pandiri Padma, 2Sanga Ramya

1assistant professor, 2PG Student

1UCE,Osmania university ,

2UCE.Osmania university

Abstract: Fast implementations have formed the basis in a whole rare form of rising performance and fault acceptable circuits premised on approximation technology. High performance comes at the expense of accuracy in many applications. Additionally, such methods minimise system architectural complexity, latency, and power consumption. When compared to existing designs, this study investigates and proposes the design and analysis of an approximation compressor with reduced size, latency, and power and equivalent accuracy. The stated approximation 4: 2 compressor now has relatively small footprint, lower power losses, and reduced latency in comparison with precise 4: 2 compressor. With the optimized compressors, 8 bit and 16 bitDadda multipliers can be designed efficiently. In comparison to current approximation multipliers, these multipliers offer equivalent accuracy.

Keywords: Approximate 4:2 compressors, approximate multipliers, error resilient applications, image processing.

I. INTRODUCTION

Approximate computing is a new trend in digital architecture that allows for significant performance gains in terms of power, speed, and space while avoiding the need for exact computation. For embedded and mobile systems with strict energy and speed limits, this method is becoming increasingly crucial. Several error-resilient applications can benefit from approximate computing. This technology involves image processors, data gathering and identification, and deep learning, to mention a very few. Multipliersare crucial aspects in microchips, digital signal, and embedded devices, that are used for a multitude of activities from filtration to convolutional layers. Due to its complicated logical framework, multipliers, from the other extreme, are among the most energy-intensive digital blocks. As a consequence, approximation multiplier development became a prominent area in recent studies. Partially generated products, partially reduced products, and carry-propagate addition are the building elements of a multiplier. In any of these blocks, approximations can be used. For instance, partial product truncation is a well-established approximation approach in which some partial products are not generated and the truncation error is minimised using appropriate correction functions.

In the design of computing systems, energy efficiency has risen to the forefront. Simultaneously, as computer systems become more integrated and mobile, computational activities have expanded to encompass image processing It is difficult to tell the difference between a decent search result and the top result in data mining. Imprecision tolerance is a feature of these applications.

Measurement error tolerated feature can be triggered by a variety of conditions, which include the: (1) human visual constraints: these are regulated by the power of the human cortex to substitute up missing key info and block out high-frequency sequences; (2) data redundancy for input: this redundancy imply that such a lossy system can indeed be viable and acceptable due to inherent tolerance; and (3) messy input data which are unwanted/ noisy info. This presentation's core objective is to evaluate new and significant improvements and advances in estimation technology which can also be referred as approx. computation (AC).The word encompasses a broad range of research efforts, from programming languages to the transistor level. The hunt for solutions that allow computing systems to exchange energy for the quality of the calculated output is a common thread running across these divergent projects.

Given that traditional Dennard s scaling yields declining returns as technology advances, taking use of the new source of energy-efficiency afforded by approximation computing is becoming increasingly vital.

Because of the huge volumes of data and intricate computations necessary in these applications, a new issue has emerged as big data processing and artificial intelligence become more important. To accelerate the development of these new technologies, energy-efficient and high-performance general-purpose compute engines, as well as application-specific integrated circuits, are in great demand. Exact or high-precision computing, on the other hand, isn't always required. Small mistakes, on the other hand, can compensate for each other or have no substantial impact on the computed results. As a result, approximation computing (AC) has evolved as a novel method to energy-efficient design as well as enhancing the performance of a computing system with little accuracy loss.

Approximate computing has developed as a new paradigm for circuit and system design that is both high-performance and energy efficient. With so many approximate arithmetic circuits suggested, it's become vital to comprehend a design or approximation approach for a given application in order to maximise performance and energy economy while minimising accuracy loss. The goal of this paper is to offer a complete overview and comparative evaluation of newly developed approximation arithmetic circuits under various design restrictions. Approximation integrators/summators, multipliers, and divisions are synthesized and evaluated in the perspective of efficiency optimizations and size reductions. The fault and circuitry attributes are then globalized all over a set of design classifications.

The circuits with lower error rates or error biases perform better in simple computations, such as the sum of products, whereas more complex accumulative computations that involve multiple matrix multiplications and convolutions are vulnerable to single-sided errors that result in a large error bias in the computed result. Because addition mistakes are more sensitive than multiplication errors in such complicated calculations, multipliers can tolerate a larger approximation than adders. In addition to the gains in performance and power consumption for these applications, the adoption of approximation arithmetic circuits can improve image processing and deep learning quality.

Approximation can be used to reduce the overhead on calculation units of a processor, resulting in improved performance and efficiency. The speed of operation, which is inversely related to the system delay, necessitates massive parallel processes, which consume a lot of hardware and energy. By lowering the accuracy and dependability of the system, energy and space efficient solutions may be realised. Approximation concept has recognized as a highly recommended technique for reconciling a balance between latency, space occupancy, and energy consumption. Faster systems with lower design complexity and power consumption result from approximated arithmetic processes.

II. EXISTING METHOD

In AI and DSPtechniques, multiplication is and will always be doubt an efficiency influencing operation. Such tasks involve considerable high performance parallel operations having minimal error, so it necessitates rising performance multiplier topologies. With the use of estimation in multipliers, quick and efficient analyses can be obtained with even less computational effort, time, and with better power efficiency yet ensuring optimum reliability.But since adder structures've got delay time, partial product addition is the worst delay effecting process in multiplication.

Compressors are now used to decline the propagation delay time. The summing value and carry generated are evaluated by compressors at every step. The resulting carry is then combined with a bit with a greater significant sum value in the subsequent steps.This technique is done repeatedly until the final outcome is achieved.Through use of estimation in multiplication procedure leading to fast computation which can occupy minimum hardware architecture requirement, computational latency, and power consumption while preserving tolerable accuracy. Partial product summation is the worst step where the delay for computation is high in overall multiplication due to the propagation latency of adder networks. Compressors are used to shorten the propagation delay. Compressors at each level compute the sum and carry. In the following stage, the resulting carry is combined with a bit with a higher meaningful amount. This process is repeated until the desired outcome is achieved.

CONVENTIONAL 4:2 COMPRESSOR:

Figure 1 depicts the overall topology of an exact 4: 2 compressor that is having five input with three outputs, and that is formed by two complete adders that are cascaded. The precise 4:2 compressor's inputs are A1, A2, A3, A4, and CIN, while the outputs are COUT, CARRY, and SUM.

$$C_{OUT} = A_3(A_1 \oplus A_2) + A_1(\overline{A_1 \oplus A_2}) \qquad (1)$$

$$CARRY = C_{IN}(A_1 \oplus A_2 \oplus A_3 \oplus A_4) + A_4(\overline{A_1 \oplus A_2 \oplus A_3 \oplus A_4}) \qquad (2)$$

$$SUM = C_{IN} \oplus A_1 \oplus A_2 \oplus A_3 \oplus A_4 \qquad (3)$$

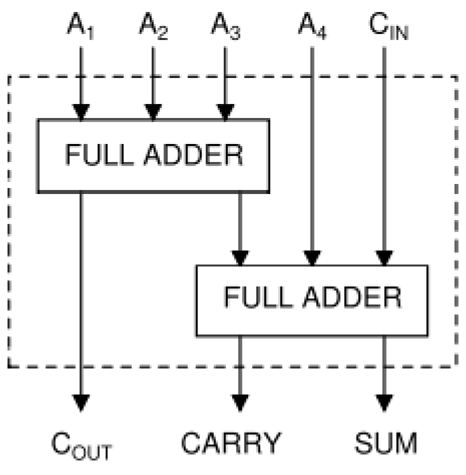


Fig.1: conventional 4 bit compressor design

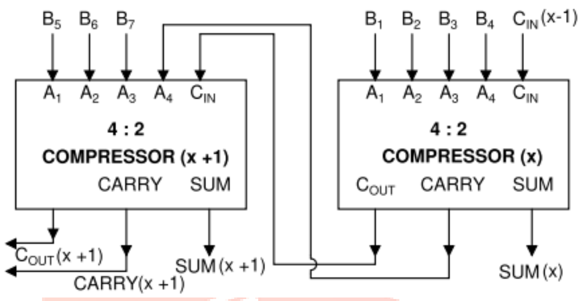


Fig.2: cascading of two compressors

Figure 2 depicts a compressor chain. The input carry from the previous 4: 2 compressor that handled the lower significant bits is represented by CIN.

Two estimated compressors are suggested in this work. Figure 3depicts the planned 4: 2 approximation high-speed area-efficient compressor. v1, v2, v3, and v4 are the compressor inputs; ca and su are the outputs, where su indicates sum of the compressor and ca indicates carry of the compressor design.

To design su, experts used a mux design approach. The return of the ex-or Logic circuit is sent into the MUX s control signal. (v3v4) is picked when the selecting line is high, and (v3 + v4) when it becomes lower. Here is a collection of the logical formulas for realizing sum and carry.

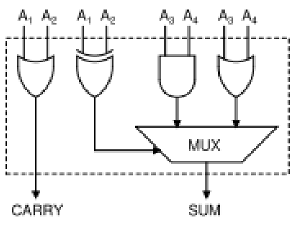


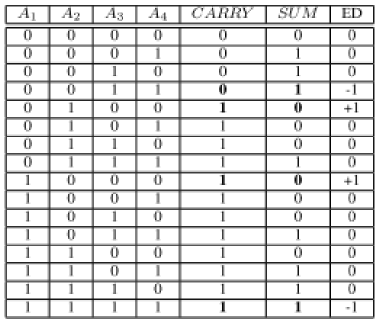
Fig.3: area-efficient 4:2 compressor

sum = (v1 v2)v3v4 + (v1 v2)(v3 + v4) (1)

carry = v1+v2 (2)

Where A1, A2, A3, A4 in this are v1, v2, v3, v4 and sum is su and carry is ca. The error has been added for the input numbers 0011, 0100, 1000, and 1111, To ensure that equal positive and negative deviation is created with ED = 1, The truth table (Table 1) of the planned 4: 2 compressor must be employed (minimum).

Table.1: Truth table of area efficient 4:2 compressor



A_1	A_2	A_3	A_4	$CARRY$	SUM	ED
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	**0**	**1**	-1
0	1	0	0	**1**	**0**	+1
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	1	0
1	0	0	0	**1**	**0**	+1
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	**1**	**1**	-1

This study proposes an alternate architecture for multipliers with more than three stages of cascaded compressors as a method of improving the hardware utilisation of the proposed design. The high performance with area-efficient compressor topology requires a XOR, AND, and 2 OR logical gates along with a MUX. OR and AND gates each utilize six CMOS based transistors. This research suggests and implements a design that affects the number of transistors reduction by leveraging NAND and NOR gates, as seen in Figure 4. Although the SUM and CARRY obtained by the modified design are not identical to those provided by the suggested 4: 2 compressor architecture, the mistake is eliminated by cascading the compressor in multiples of 2.

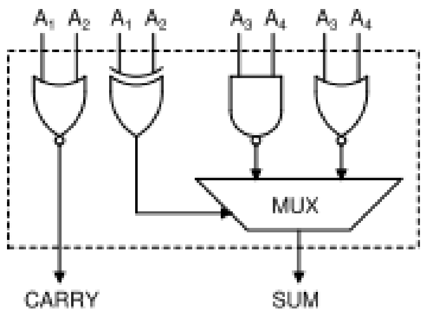


Fig.4 The suggested improved Dual-stage 4: 2 compressors basic building block.

By approximating the 4: 2 compressor, the output count can be reduced to two. To approximate the outcome, COUT is removed. When the input combination is 1111, an error occurs. The CARRY and SUM are both set to 11 when the input bits are 1111, resulting in a one-bit error.

III. PROPOSED METHOD

High-speed multipliers are becoming increasingly used in a variety of computing applications, including computer graphics, scientific calculations, and image processing, and so on. The multipliers speed determines how fast the processors run, and designers are currently concentrating on achieving high speed while utilising minimal electricity. The multiplication architectural style including of three main layers such as: partial product outputs production or calculation, partial product minimization/reduction process, and final adding the obtained two rows of results with general adder like RCA, PPA etc. The PPR is responsible for a significant portion of the multiplier time, energy, and area overhead. Compressors are frequently built to produce partial products in order to reduce delay and improve performance.

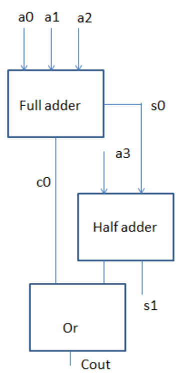


Fig: 5 new design for approximate compressor.

To analyse compressor, we are now building an 8 bitdadda and 16 bit dadda multiplier.Below is an illustration of an 8-bit dadda multiplier. The suggested multiplier offers a faster speed and better accuracy than the present multiplier.

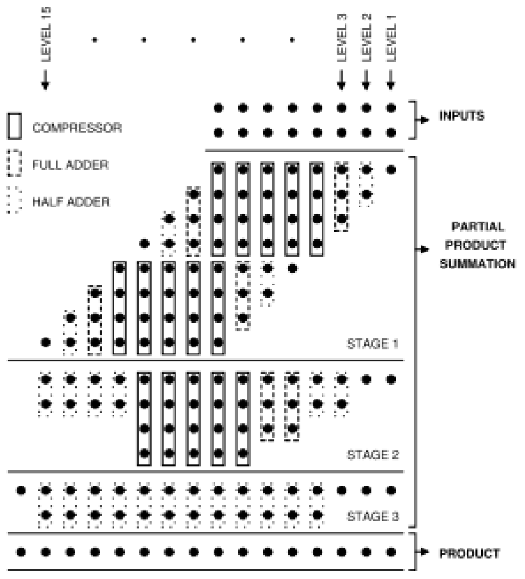


Fig.6: approx. multiplier for 8 x 8 bit

The 16-bit dadda multiplier is created by combining proposed compressors with an identical compressor that already exists. Due to proposed design, exactness is somewhat increased compared to the current multiplier, and the multiplier s performance is enhanced compared to the existing multiplier because of the suggested compressor architecture. Below is a diagram of the multiplier design utilising the suggested 4:2 compressor.

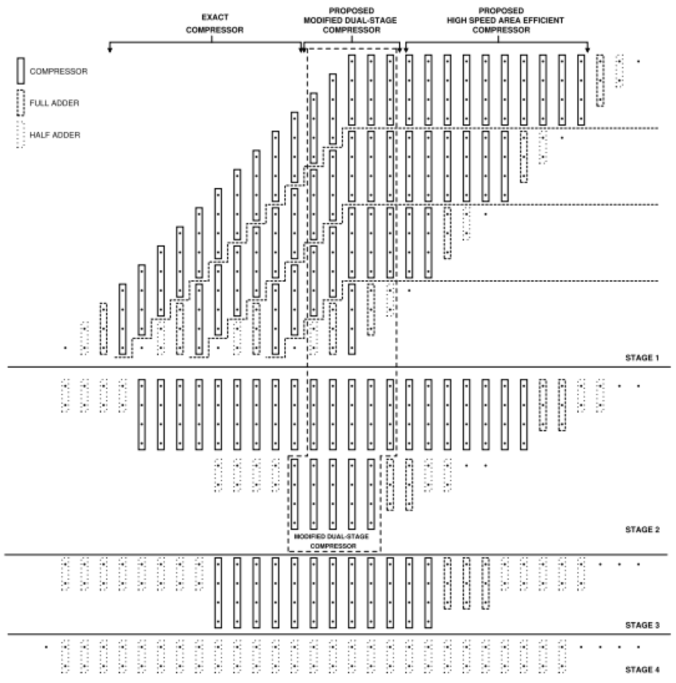

EXACT COMPRESSOR
PROPOSED MODIFIED DUAL-STAGE COMPRESSOR
PROPOSED HIGH SPEED AREA EFFICIENT COMPRESSOR
COMPRESSOR
FULL ADDER
HALF ADDER
STAGE 1
MODIFIED DUAL-STAGE COMPRESSOR
STAGE 2
STAGE 3
STAGE 4

Fig: 7 proposed 4: 2 compressors used in 16 × 16 multiplier.

IV. EXPERIMENTAL RESULTS

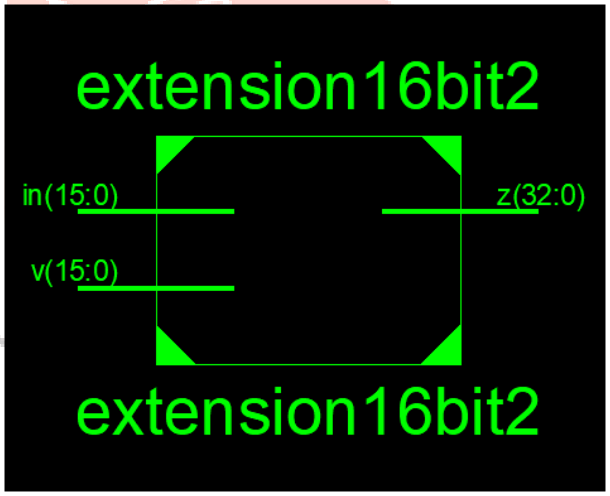


Fig.8: RTL Schematic of FFT

Figure 8 shows RTL block level schematic of the implemented design. RTL refers to register transfer level. This is the schematic produced by the tool based on the code designed by the user as per the top module of the circuit.

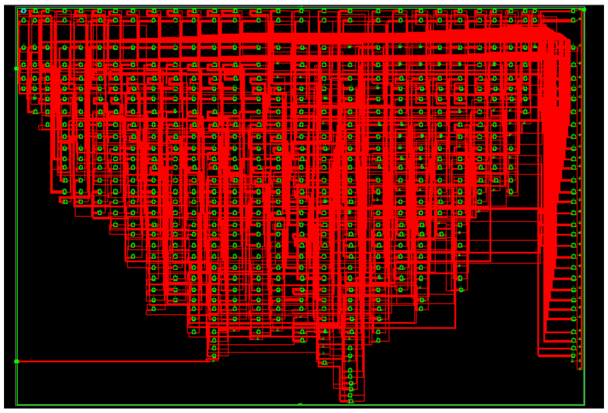


Fig.9 Technology schematic of 16 bit approx. multiplier

Figure 9 shows Technology schematic of the implemented design. This is the schematic produced by the tool based on the optimization of code designed by the user as per the top module of the circuit which converts the entire code into LUT blocks and forms a schematic which is technology schematic and this varies based on the type of FPGA device selected during the project creation in the synthesis tool.

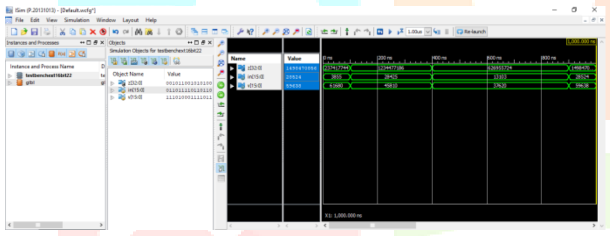

ISim (P.20131013) - [Default.wcfg*]
File Edit View Simulation Window Layout Help
Instances and Processes
Objects
Simulation Objects for testbenchext16bit22
Instance and Process Name
testbenchext16bit22
glbl
Object Name
Value
Name
z[32:0]
in[15:0]
v[15:0]
Re-launch

Fig.10: Simulation results for 16 bit approx. multiplier

Figure 10 shows thesimulation results of waveformsfor implemented design using test bench. In this, we observe that when the based on the input signals and, for every set of inputvalues, theoutput values are generated which are the approximate results that makes use of the proposed compressor for the reduction of partial products in the multiplication process.

	AREA (LUT's)	DELAY (ns)	POWER(W)
proposed4:2 COMP	2	6.23	0.034
4:2 HIGH SPEED	2	6.23	0.034
4:2 DUALSTAGE COMP	2	6.23	0.034
8BIT DUAL STAGE COMP	98	11.98	0.034
8 BIT MULTIPLIER USING HIGH SPEED COMP	95	12.12	0.034
proposed 8 BIT MULTIPLIER USING DUAL STAGE COMP	111	10.62	0.034
16BIT MULTIPLIER	524	44.8	0.034
16BIT MULTIPLIER proposed	558	42.3	0.034

Table2: comparison between Existing and Proposed methods.

Table2 describes the comparison of performance parameters such as Area and Delay between multipliers implemented with high speed and dual stage compressors (Existing) and multipliers using novel compressor (Proposed). Based on these findings, we claim that the suggested method's duration is minimized by deploying an innovative compressor approach. Integrating preexisting compressors with creative compressors which are later proposed, a tradeoff between area overhead and computational delay is built between existing and later implemented 8 bit and 16 bit multiplication algorithms.

V. CONCLUSION

Here in this article, a new compressor based multiplier is designed and implemented. In comparison to traditional compressors, the proposed design has a faster speed and a higher degree of accuracy. To validate the comparator design, the 8 x 8, 16 x 16, Dadda multiplier was utilised. In terms of performance and accuracy, output waveforms indicates that it outperforms existing approx. multiplier designs. In the future by making considerations and by establishing a tradeoff we can attain better outcomes.

REFERENCES

[1] Mohapatra, et.al through a transaction in IEEE published, Voltage scalable high-speed robust hybrid arithmetic units using adaptive clocking, in 2010.

[2] Baran, et.al, in IEEE ISCAS, Multiplier structures for low power applications in deep-CMOS, from Brazil, May 2011 published.

[3] S. Mittal, published a report at ACM in March 2016 on, A survey of techniques for approximate computing.

[4] Liu, et.al, analyzed a research in 2017, A review classification and comparative evaluation of approximate arithmetic circuits.

[5] Lombardi, with his team published a transaction work in IEEE on New metrics for the reliability of approximate and probabilistic adders, in Sep. 2013.

[6] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, Energy-efficient approximate multiplication for digital signal processing and classification applications, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

[7] G. Zervakis, S. Xydis, K. Tsoumanis, D. Soudris, and K. Pekmestzi, Hybrid approximate multiplier architectures for improved poweraccuracy trade-offs, in Proc. IEEE/ACM Int. Symp. Low Power Electron. Des. (ISLPED), Rome, Italy, Jul. 2015, pp. 79–84.

[8] B. Shao and P. Li, Array-based approximate arithmetic computing: A general model and applications to multiplier and squarer design, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 62, no. 4, pp. 1081–1090, Apr. 2015.

[9] C.-H. 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 Syst. I, Reg. Papers, vol. 51, no. 10, pp. 1985–1997, Oct. 2004.