iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.50528

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

Implementation of Hardware and Energy Efficient Approximate Multiplier Architectures Using 4-2 Compressor for Images

Hema C[1], Shravani G[2], P Sivaphaneendra[3], Sinchana[4], Soundarya L[5]
[1]*Assistant Professor, Department of ECE, EWIT,*
[2, 3, 4, 5]*Student, Department of ECE, EWIT*

Abstract: ***Approximate computing is tentatively applied in some digital signal processing applications which have an inherent tolerance for erroneous computing results. The approximate arithmetic blocks are utilized in them to improve the electrical performance of these circuits. Multiplier is one of the fundamental units in computer arithmetic blocks. Moreover, the 4-2 compressors are widely employed in the parallel multipliers to accelerate the compression process of partial products. In this brief, three novel approximate 4-2 compressors are proposed and utilized in 8-bit multipliers. Meanwhile, an error-correcting module (ECM) is presented to promote the error performance of approximate multiplier with the proposed 4-2 compressors. In this brief, the number of the approximate 4-2 compressor's outputs is innovatively reduced to one, which brings further improvements in the energy-efficiency. This Design is implemented using Verilog HDL and simulated by Modelsim 6.4 c and synthesized by Xilinx tool.***
Keywords: 4-2 Compressors, Error-Correcting Module (ECM), 8-bit Multipliers.

I. INTRODUCTION

In applications like multimedia signal processing and data mining which can tolerate error, exact computing units are not always necessary. They can be replaced with their approximate counterparts. Research on approximate computing for error tolerant applications is on the rise. Adders and multipliers form the key components in these applications. In approximation, full adders are proposed at transistor level and they are utilized in digital signal processing applications. Their proposed full adders are used in accumulation of partial products in multipliers. To reduce hardware complexity of multipliers, truncation is widely employed in fixed-width multiplier designs. Then a constant or variable correction term is added to compensate for the quantization error introduced by the truncated part. Approximation techniques in multipliers focus on accumulation of partial products, which is crucial in terms of power consumption.

Broken array multiplier is implemented, where the least significant bits of inputs are truncated, while forming partial products to reduce hardware complexity. The proposed multiplier saves few adder circuits in partial product accumulation. Three designs of approximate 4-2 compressors are presented and used in partial product reduction tree of four variants of 8 × 8 Dadda multiplier. The major drawback of the proposed compressors is that they give nonzero output for zero valued inputs, which largely affects the outcome. The approximate design proposed in this brief overcomes the existing drawback. This leads to better precision.

Multiplication is a most commonly used operation in many computing systems. A number (multiplicand) is added to itself a number of times as specified by another number (multiplier) to form a result (product). But the implementation of multiplier takes huge hardware resources and the circuit operates at low speed. Multiplication speed determines processor speed. So high speed multipliers are needed in the processors for many applications. For increasing the speed of multiplication different algorithms are used. Multipliers form an important hardware block in the DSP and Embedded applications.

Driven by the development of applications and the advancement of semiconductor process technology, the complexity, scale, and density of integrated circuits are fast increasing. While it comes with a rapid increase on power consumption, which in turn reduces the lifetime and reliability of devices. Fortunately, in many applications such as multimedia, digital signal processing, and machine learning, the accuracy loss in a proper range does not influence the quality of what we appreciate, due to a limited human perception. This leads to an opportunity that the full precision computing blocks are substituted with the approximate counterparts.

For 16-bit approximate multiplier 26% of reduction in power is accomplished compared to exact multiplier. Approximation of 8-bit Wallace tree multiplier due to voltage over-scaling (VOS) is discussed.

Lowering supply voltage creates paths failing to meet delay constraints leading to error. Previous works on logic complexity reduction focus on straightforward application of approximate adders and compressors to the partial products. In this brief, the partial products are altered to introduce terms with different probabilities. Probability statistics of the altered partial products are analysed, which is followed by systematic approximation. Simplified arithmetic units (half-adder, full-adder, and 4-2 compressor) are proposed for approximation. The arithmetic units are not only reduced in complexity, but care is also taken that error value is maintained low. While systematic approximation helps in achieving better accuracy, reduced logic complexity of approximate arithmetic units consumes less power and area. The proposed multipliers outperform the existing multiplier designs in terms of area, power, and error, and hence they are used in image processing application.

Edge detection techniques have been successfully used for different applications. In edge detection, the abrupt changes in the pixel intensity are determined. These change in pixel intensities are determined by different techniques, in which different parameters are tuned to refine the edges of salient objects while suppressing the redundant objects from image. The edges obtained by different edge detector are broadly classified into two types: correct edges and false edges. Correct edge represent salient object and false edges are produced due to detector sensitiveness. The edge detection algorithms have three steps: Filtering, Enhancement and Detection. Filtering is normally used to remove the noise from image.

Enhancement is used to magnify the pixel intensity values in local area of an image and in Detection the strong edges are determined. Recent research in the fields of Artificial Intelligence computer vision and Pattern Recognition reveals that the edge detection is very important in some way or the other. Key point detection is one major part of the process that majorly deals with image edges not only edges but also true edges. Identified key points are then used to describe the feature vectors that are further used in different applications. There is much research going on nowadays on edge detection as it has a key role in almost all upcoming fields.

A. Problem Statement

Approximate designs are introduced in multipliers. For multipliers of larger bit widths, hybrid-radix Booth encoding method is normally utilized, which focuses on the approximation of the partial products generation. Multipliers of smaller bit widths, as their partial products are usually generated by simple AND gates, the approximation are applied in compression trees, which is arranged to accumulate all the generated partial products.

4-2 compressor is the core of such compression trees, which balances the compression efficiency and hardware cost.

B. Purpose and Scope

Purpose

The purpose of our project is to reduce errors to implement it easily in practical time, and also to reduce complexity in Logic Simplification.

Scope

It reduces Area and Power Consumption.

C. Existing System

In Existing Design, three designs of approximate 4-2 compressors are presented and used in partial product reduction tree of four variants of 8 × 8 Array multiplier. The major drawback of the proposed compressors is that they give nonzero output for zero valued inputs. Three different schemes for utilizing the proposed approximate compressors are implemented and analysed for an Array multiplier. Extensive simulation results are provided and an application of the approximate multipliers to image processing is presented.

D. Proposed System

In this project, three approximate 4-2 compressors and an error-correcting module (ECM) are proposed. These three 4-2 compressors are designed from a holistic aspect based on the compensation characteristic of addition. The error performance of an approximate compressor chain is considered instead of a single approximate compressor. We innovatively reduce the number of approximate 4-2 compressor's outputs to one. These multipliers have comparable accuracy when compared with state-of-the-art other multipliers. Performance of the proposed multiplier is evaluated with an image processing application like Sobel edge detection and implemented in MATLAB and Model Sim.

II. METHODOLOGY

A. Proposed System Flow

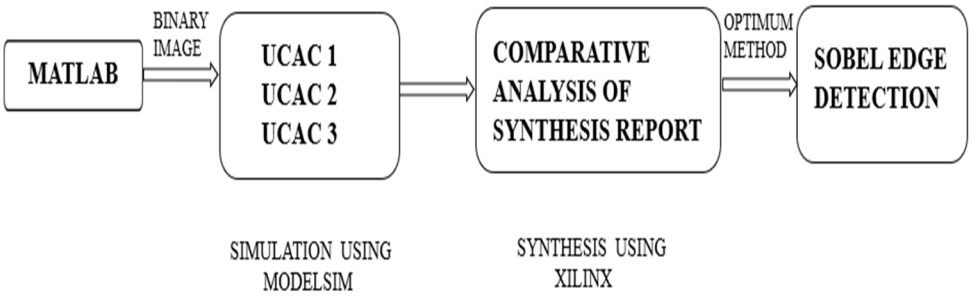


Fig 1: Block Diagram of Proposed System Flow

UCAC: Ultra Low Power Consumption Approximate Compress
Three approximate 4-2 compressors (UCAC1, UCAC2, and UCAC3) are proposed in this section. Then, the ECM is presented to detect an input pattern with a large probability and correct the erroneous compensation in this case. Furthermore, the proposed designs are embedded in 8-bit multipliers based on the partial product tree. And all the analyses are performed with the uniform distribution.

The proposed approximate compressors and ECM are designed to simplify and accelerate the compression process, four 8-bit multipliers (N = 8) are designed to evaluate these blocks, accordingly.

1) MUL1: Multiplier with UCAC1 and ECM
2) MUL2: Multiplier with UCAC2 and ECM
3) MUL3: Multiplier with UCAC3 and ECM

B. Design Of Power And Area Efficient Approximate Multiplier

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore, low power multiplier design has an important part in low-power VLSI system design. A system is generally determined by the performance of the multiplier because the multiplier is generally the slowest element and more area consuming in the system. Hence optimizing the speed and area of the multiplier is one of the major design issues. However, area and speed are usually conflicting constraints so that improvements in speed results in larger areas. Multiplication is a mathematical operation that include process of adding an integer to itself a specified number of times. A number (multiplicand) is added itself a number of times as specified by another number (multiplier) to form a result(product). Multipliers play an important role in today's digital signal processing and various other applications.
Multiplier design should offer high speed, low power consumption. Multiplication involves mainly 3 steps:

1) Partial product generation
2) Partial product reduction
3) Final addition

Fast arithmetic computation cells including adders and multipliers are the most frequently and widely used circuits in VLSl systems. Microprocessors and digital signal processors rely on the efficient implementation of generic arithmetic logic units and floating-point units to execute dedicated algorithms such as convolution and filtering.
In most of these applications, multipliers have been the critical and obligatory component dictating the overall circuit performance when constrained by power consumption and computation speed. With trends of VLSl technologies towards deep-submicron regime, the most eminent means of achieving power efficacy is by lowering the power supply voltage.

C. *Array Multiplier*

The Array multiplier is a hardware multiplier design invented by computer scientist Luigi Array in 1965. It is similar to the Wallace multiplier, but it is slightly faster (for all operand sizes) and requires fewer gates (for all but the smallest operand sizes). In fact, Array and Wallace multipliers have the same 3 steps:

1) Multiply (logical AND) each bit of one of the arguments, by each bit of the other, yielding n^2 results. Depending on position of the multiplied bits, the wires carry different weights, for example wire of bit carrying result of $a_2 b_3$ is 32.
2) Reduce the number of partial products to two by layers of full and half adders.
3) Group the wires in two numbers, and add them with a conventional adder.

However, unlike Wallace multipliers that reduce as much as possible on each layer, Array Multiplier do as few reductions as possible. Because of this, Array Multiplier have a less expensive reduction phase, but the numbers may be a few bits longer, thus requiring slightly bigger adders.

To achieve this, the structure of the second step is governed by slightly more complex rules than in the Wallace tree. As in the Wallace tree, a new layer is added if any weight is carried by three or more wires. The reduction rules for the Array tree, however, are as follows:

1) Take any three wires with the same weights and input them into a full adder. The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires.
2) If there are two wires of the same weight left, and the current number of output wires with that weight is equal to 2 (modulo 3), input them into a half adder. Otherwise, pass them through to the next layer. If there is just one wire left, connect it to the next layer.

This step does only as many adds as necessary, so that the number of output weights stays close to a multiple of 3, which is the ideal number of weights when using full adders as 3:2 compressors.

The same procedure follows in Dadda multiplier, the contrast is Dadda's method does the minimum reduction necessary at each level to perform the reduction in the same number of levels as required by a Wallace multiplier.

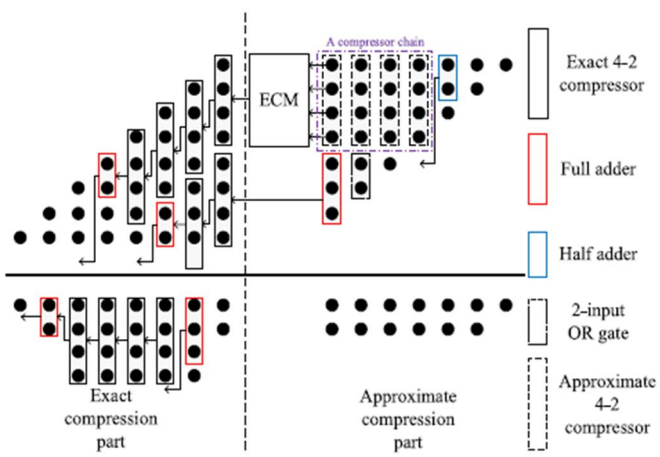


Fig 2: Array multiplier

D. *Proposed multiplier block diagram*

Modules Used By Dadda Method

1) Partial Product Generation Block
2) OR Gate
3) Approximate Half Adder
4) Approximate Full Adder
5) Approximate 4-2 Compressor
6) UCAC Compressor
7) Error-Correcting Module

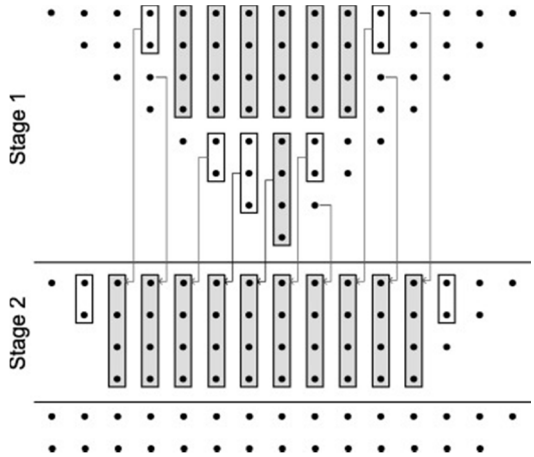


Fig 3: Dadda Multiplier Using Proposed Designs

E. Sobel Edge Detection

Edge detection algorithms are widely used in various research fields like Image Processing, Video Processing and Artificial Intelligence etc. Edges are most important attribute of image information and a lot of edge detection algorithms are defined in literature. Sobel edge detection algorithm is chosen among of them due to its property of less deterioration in high level of noise. FPGA is becoming the most dominant form of programmable logic over past few years and it has advantages of low investment cost and desktop testing with moderate processing speed and thereby offering itself as suitable one for real time application.

III. IMPLEMENTTION

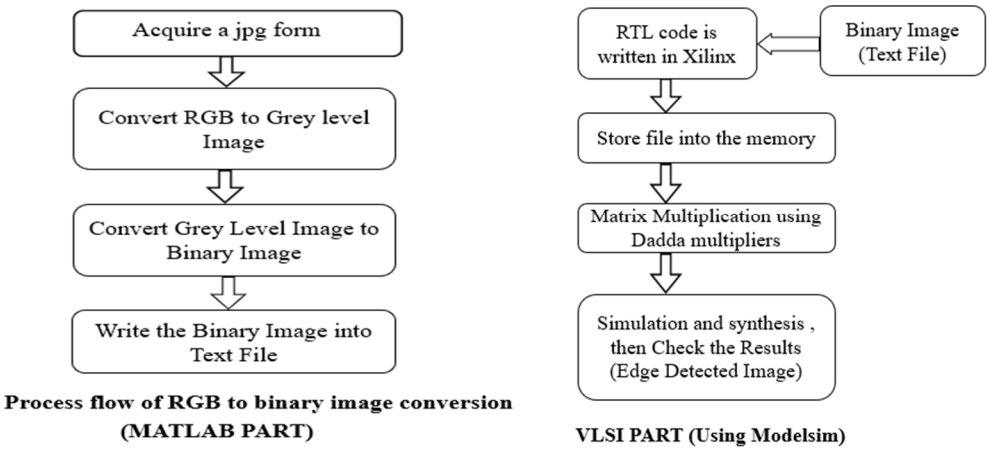


Fig 4: Flowchart of the Proposed System

A. Snapshots of Proposed Model

INPUT IMAGE GREY LEVEL IMAGE

RESIZED IMAGE SOBEL EDGE IMAGE

IV. RESULT

TABLE I
COMPARISON RESULTS

PARAMETER	OVERALL DELAY	LUT	GATES COUNT	POWER
Dadda Normal	45.844 ns	176	1056	144
Booth Normal	48.409 ns	185	1188	141
UCCAC1 DADDA	31.479 ns	141	852	256
UCCAC2 DADDA	30.889 ns	140	852	148
UCCAC3 DADDA	32.628 ns	145	882	129

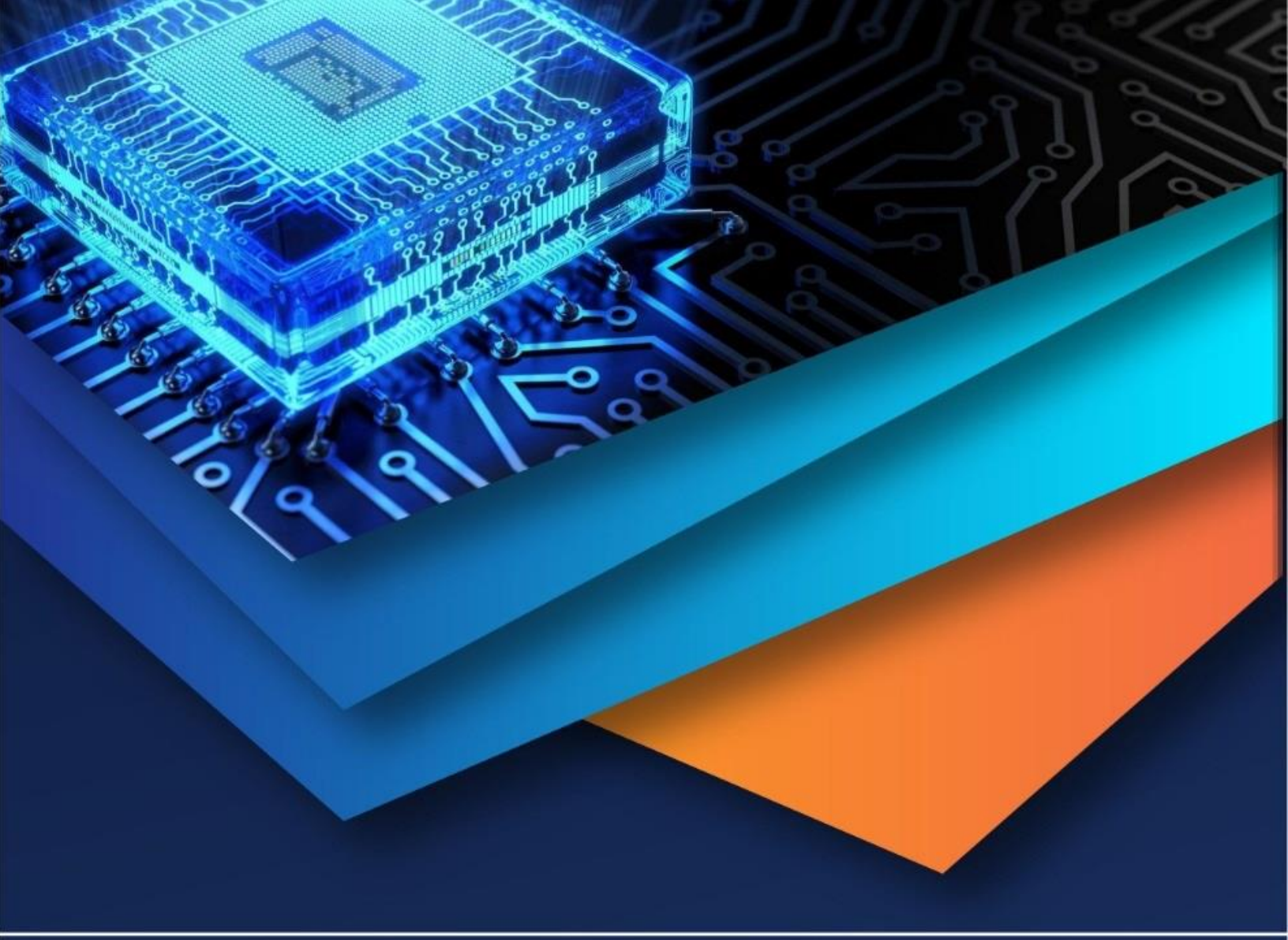
V. CONCLUSION

All approximate multipliers are designed for n = 8. The multipliers are implemented in Verilog and synthesized using this method deals with the analysis and design of two new approximate 4-2 compressors for utilization in a multiplier. we have designed a Efficient Array Multiplier using our proposed Multiplier. The proposed approximate compressors are implemented and analysed for an Array multiplier. This Proposed Multiplier is used in Sobel operator design which is having a better performance in terms of Area, Power and Delay. The proposed method is implemented using Verilog HDL and simulated by Modelsim and synthesized by Xilinx tools. This proposed method is utilized in Sobel operator design. Sobel operator executed in Modelsim (to display waveform) and in MATLAB software, which displays edge detected output image.

REFERENCES

[1] J. Liang, J. Han, and F. Lombardi, “New metrics for the reliability of approximate and Probabilistic Adders,” IEEE Trans. Computers, vol. 63, no. 9, pp. 1760–1771, Sep. 2013.
[2] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, “IMPACT: Imprecise adders for low-power approximate computing,” in Proc. Int. Symp. Low Power Electron. Design, Aug. 2011, pp. 409–414.
[3] S. Cheemalavagu, P. Korkmaz, K. V. Palem, B. E. S. Akgul, and L. N. Chakrapani, “A probabilistic CMOS switch and its realization by exploiting noise,” presented at the IFIP Int. Conf. Very Large Scale Integ., Perth, Australia, Oct. 2005.
[4] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, “Bioinspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” IEEE Trans. Circuits Syst. I: Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.
[5] M. J. Schulte and E. E. Swartz lander Jr., “Truncated multiplication with correction constant,” in Proc. Workshop VLSI Signal Process. VI, 1993, pp. 388–396.
[6] E. J. King and E. E. Swartz lander Jr., “Data dependent truncated scheme for parallel multiplication,” in Proc. 31st Asilomar Conf. Signals, Circuits Syst., 1998, pp. 1178–1182.

[7] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy for power in a multiplier architecture," J. Low Power Electron., vol. 7, no. 4, pp. 490–501, 2011.
[8] 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 Syst., vol. 51, no. 10, pp. 1985–1997, Oct. 2004.
[9] D. Radhakrishnan and A. P. Preethy, "Low-Power CMOS pass logic 4-2 compressor for high-speed multiplication," in Proc. IEEE 43rd Midwest Symp. Circuits Syst., 2000, vol. 3, pp. 1296–1298.
[10] Z. Wang, G. A. Jullien, and W. C. Miller, "A new design technique for column compression multipliers," IEEE Trans. Compute., vol. 44, no. 8, pp. 962–970, Aug. 1995.
[11] J. Gu and C. H. Chang, "Ultra-low-voltage, low-power 4-2 compressor for high-speed multiplications," in Proc. 36th IEEE Int. Symp. Circuits Syst., Bangkok, Thailand, May 2003, pp. v-321–v-324.
[12] M. Margala and N. G. Durdle, "Low-power low-voltage 4-2 compressors for VLSI Applications," in Proc. IEEE Alessandro Volta Memorial Workshop Low-Power Design, 1999, pp. 84–90.
[13] B. Parhami, Computer Arithmetic; Algorithms and Hardware Designs, 2nd ed. London, U.K.: Oxford Univ. Press, 2010.
[14] K. Prasad and K. K. Parhi, "Low-power 4-2 and 5-2 compressors," in Proc. 35th Asilomar Conf. Signals, Syst. Comput., 2001, vol. 1, pp. 129–133.
[15] M. D. Ercegovac and T. Lang, Digital Arithmetic. Amsterdam, The Netherlands: Elsevier, 2003.
[16] D. Baran, M. Aktan, and V. G. Oklobdzija, "Energy efficient implementation of parallel CMOS multipliers with improved compressors," in Proc. ACM/IEEE 16th Int. Symp. Low Power Electron. Design, 2010, pp. 147–152.
[17] D. Kelly, B. Phillips, and S. Al-Sarawi, "Approximate signed binary integer multipliers for arithmetic data value speculation," in Proc. Conf. Design Architect. Signal Image Process., 2009, pp. 97–104.
[18] J. Ma, K. Man, T. Krilavicius, S. Guan, and T. Jeong, "Implementation of high-performance multipliers based on approximate compressor design," presented at the Int. Conf. Electrical and Control Technologies, Kaunas, Lithuania, 2011.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)