DESIGN OF POWER AND SPEED EFFICIENT APPROXIMATE LOW POWER MULTIPLIERS

G.PRASHANTHI
M.Tech (Chalapathi Institute of technology)
grr10764@gmail.com

A.LALITHA
Assistant professor
M.Tech (Chalapathi Institute of technology)
aswanidecs13@gamil.com

Abstract: *A multiplier significantly affects the speed and power dissemination of a number juggling processor. Exact outcomes are not constantly required in numerous calculations, for example, those for characterization and acknowledgment in information preparing This short arrangements with another structure approach for estimated tion of multipliers. The incomplete results of the multiplier are modified to present differing likelihood terms. The proposed estimation is used in two variations of 16-bit multipliers. Amalgamation results uncover that two proposed multipliers accomplish control reserve funds of 60% and 30%, separately, contrasted with a careful multiplier. They have better exactness when contrasted with existing surmised multipliers. Mean relative mistake figures are as low as 5.36% and 0.03% for the proposed surmised multipliers, which are superior to the past works. where one of the proposed models accomplishes the most astounding pinnacle sign to clamor proportion. A Tabu inquiry calculation and planning calculation is then explored as an utilization of the multiplier.*

Key Words *— Approximate computing, error analysis, low error, low power, multipliers.*

Introduction

Math units, for example, adders and multipliers are key segments in a rationale circuit. The speed and power utilization of number-crunching circuits essentially impact the presentation of a processor. Superior number juggling circuits, for example, Carry Look Ahead Adders (CLAs) and Wallace tree multipliers have been broadly used. In any case, customary number-crunching circuits that perform accurate tasks are experiencing challenges in execution improvement. Inexact number-crunching that permits lost precision can diminish the basic way deferral of a circuit. Since most surmised plans influence rearranged rationale, they will in general have a decreased power utilization and region overhead. Along these lines, rough number-crunching is supported as a way to deal with improve the speed, zone and power productivity of a processor because of the blunder flexibility of certain calculations and applications.

In applications like interactive media sign preparing and information mining which can endure blunder, precise processing units are not constantly vital. They can be supplanted with their surmised partners. Research on estimated processing for blunder tolerant applications is on the ascent. Adders and multipliers structure the key segments in these applications., rough full adders are proposed at transistor level and they are used in advanced sign preparing applications. Their proposed full adders are utilized in collection of fractional items in multipliers.

Multiplier dependent on adjusting a passage in the Karnaugh guide is proposed and utilized as a structure square to develop 4 a 8 multipliers. In [9], incorrect counter structure has been proposed for use in power proficient Wallace tree multiplier. Another rough viper is displayed in [10] which is used for halfway item amassing of the multiplier. For 16-bit estimated multiplier in [10], 26% of decrease in power is cultivated contrasted with definite multiplier. Estimation of 8-bit Wallace tree multiplier because of voltage over-scaling (VOS) is talked about in [11]. Bringing down supply voltage makes ways neglecting to meet defer imperatives prompting

mistake. Past takes a shot at rationale unpredictability decrease center around straight-forward utilization of surmised adders and blowers to the fractional items. In this short, the fractional items are adjusted to present terms with various probabilities. Likelihood measurements of the changed incomplete items are examined, which is trailed by efficient estimate.

Rearranged number juggling units (half-snake, full-viper, and 4-2 blower) are proposed for guess. The number juggling units are decreased in multifaceted nature, yet care is additionally taken that mistake worth is looked after low. While foundational estimation helps in accomplishing better exactness, diminished rationale intricacy of surmised number juggling units devours less power and territory. The proposed multipliers beats the current multiplier plans as far as zone, power, and mistake, and accomplishes better pinnacle sign to commotion proportion (PSNR) values in picture preparing application. Mistake separation (ED) can be characterized as the number-crunching separation between a right yield and estimated yield for a given info. In [12], estimated adders are assessed and standardized ED (NED) is proposed as almost invariant metric free of the size of the rough circuit. Additionally, customary mistake investigation, MRE is found for existing and proposed multiplier plans.

The remainder of this brief is sorted out as pursues. Segment II subtleties the proposed design. Area III gives broad outcome examination of structure and blunder measurements of the proposed and existing estimated multipliers. The proposed multipliers are used in picture process-ing application and results are given in Section IV. Area V finishes up this brief.

Writing Review

During the mid-1920s, a few innovators endeavored gadgets that were expected to control current in strong state diodes and convert them into triodes. Achievement did not come until after WWII, during which the endeavor to improve silicon and germanium gems for use as radar locators prompted enhancements in creation and in the comprehension of quantum mechanical conditions of bearers in semiconductors. At that point researchers who had been redirected to radar improvement came back to strong state gadget advancement. With the creation of transistors at Bell Labs in 1947, the field of hardware moved from vacuum cylinders to strong state gadgets.

With the little transistor at their hands, electrical designers of the 1950s saw the potential outcomes of developing unquestionably further developed circuits. As the intricacy of circuits developed, issues emerged.

One issue was the size of the circuit. A mind boggling circuit, similar to a PC, was reliant on speed. On the off chance that the segments of the PC were excessively huge or the wires interconnecting them excessively long, the electric sign couldn't travel quick enough through the circuit, hence making the PC too moderate to even think about being effective.[1]

Jack Kilby at Texas Instruments found an answer for this issue in 1958. Kilby's thought was to make every one of the segments and the chip out of a similar square (stone monument) of semiconductor material. Kilby displayed his plan to his bosses, and was permitted to fabricate a test rendition of his circuit. In September 1958, he had his initially incorporated circuit prepared. In spite of the fact that the primary incorporated circuit was rough and had a few issues, the thought was notable. By making every one of the parts out of a similar square of material and adding the metal expected to associate them as a layer over it, there was no requirement for discrete segments. No more wires and segments must be

collected physically. The circuits could be made littler, and the assembling procedure could be mechanized. From here, incorporating all parts on a solitary silicon wafer appeared, which prompted improvement in little scale mix (SSI) in the mid 1960s, medium-scale joining (MSI) in the late 1960s, and afterward huge scale combination (LSI) just as VLSI during the 1970s and 1980s, with a huge number of transistors on a solitary chip (later many thousands, at that point millions, and now billions (109)). Usage of multiplier involves three stages: age of incomplete items, fractional items decrease tree, lastly, a vector combine expansion to create last item from the total and convey columns produced from the decrease tree. Second step devours more power. In this concise, guess is connected in decrease tree arrange.

A 8-bit unsigned1 multiplier is utilized for outline to depict the proposed strategy in estimate of multipliers. Consider two 8-bit

Prologue to verilog

The fundamental unit and programming in Verilog is "MODULE"(a content document containing proclamations and assertions). A Verilog module has revelations that portrays the names and sorts of the module data sources and yields just as neighborhood signals, factors, constants and capacities that are utilized inside to the module ,are not noticeable outside. A Verilog module has presentations that depicts the names and sorts of the module sources of info and yields just as neighborhood signals, factors, constants and capacities that are utilized inside to the module ,are not unmistakable outside. A Verilog module has affirmations that depicts the names and kinds of the module sources of info and yields just as neighborhood signals, factors, constants and capacities that are used internally to the module,are not visible outside. The signals in a vector can be ascending or descending order.

LOGIC SYSTEM:

Verilog uses 4 –logic system .a 1 –bit signal can take one of only four possible values.

0 LOGIC 0,OR FALSE

1 LOGICAL 1,OR FALSE

X A UNKNOWN LOGICAL VALUE Z HIGH IMPEDENCE

Structural Design Is the Series of Concurrent Statement .The Most Important Concurrent Statement In the module covered like instance statements, continuous –assignment statement and always block. These gives rise to three distinct styles of circuit design and description.

Statement of these types, and corresponding design styles, can be freely intermixed within a Verilog module declaration.

Each concurrent statement in a Verilog module "executes" simultaneously with other statements in the same module declaration.

In fact, the simulator will propagating changes and updating results until the simulated circuit stabilizes.

Verilog has several built in gate types, the names of these gates are reserved words, some of these are

And	xor
	bufif0
Nand	xnor
	bufif1
Or	buf
	notif0
Nor	not
	notif1

PORT DECLERATION

Verilog module declaration begins with a keyword "module" and ends with"endmodule". The input and output ports are signals by which the module communicates with each other's.

Syntax:

Input identifier ..... identifier;

Output identifier..... identifier;

Inout identifier ..... identifier;

Input [msb:lsb] identifier ..... identifier;

Output[msb:lsb] identifier..... identifier

Inout [msb:lsb] identifier ..... identifier;

Module definitions contain various components. Keywords module and end module are mandatory. Other components-port list, port declarations, variable and signal declarations, dataflow statements, behavioral blocks, lower-level module instantiations, and tasks or functions-are optional and can be added as needed. ports provide the module with a means to communicate with other modules or its environment. A module can have a port list. Ports in the port list must be declared as input, output, or in out. When instantiating a module, port connection rules are enforced by the Verilog simulator. Ports can be connected by name or by ordered list. Each identifier in the design has a unique hierarchical name. Hierarchical names allow us to address any identifier in the design from any other level of hierarchy in the design. Each port that is named at the beginning of the module ,in the input, output list must have corresponding input and output declaration.

OPERATIONS AND ASSIGNMENTS:

The design description at the behavioral level is done through a sequence of assignments. These are called 'procedural assignments' – in contrast to the continuous assignments at the data flow level. The procedure assignment is characterized by the following:

The assignment is done through the "=" symbol (or the "<=" symbol) as was the case with the continuous assignment earlier

An operation is carried out and the result assigned through the "=" operator to an operand specified on the left side of the "=" sign – for example,

N = ~N;

Here the content of reg N is complemented and assigned to the reg N itself. The assignment is essentially an updating activity.

All the operands are given in Tables 6.1 to 6.9. The format of using them and the rules of precedence remain the same. The operands on the right side can be of the net or variable type. They can be scalars or vectors. It is necessary to maintain consistency of the operands in the operation expression – e.g., N = m / l;

Here m and l have to be same types of quantities – specifically a reg, integer, time, real, realtime, or memory type of data – declared in advance.

The operand to the left of the "=" operator has to be of the variable (e.g., reg) type. It has to be specifically declared accordingly. It can be a scalar, a vector, a part vector, or a concatenated vector. As soon as a specified operation on the right is carried out, the result is assigned to the quantity on the left – for example

- N = m + l;
- N1 = N * N;

A unique approximate number with a dynamic vary choice theme. They style the number to own AN unbiased error distribution, that results in lower machine errors in real applications as a result of errors cancel one another out, instead of accumulate, because the number is employed repeatedly for a computation. Our approximate number style is additionally ascendible, facultative designers to parameterize it betting

PROPOSED ARCHITECTURE

Style are able to do power savings of fifty four – eightieth, whereas introducing finite errors with a normal distribution with near-

zero average and customary deviations of zero.45% – 3.61%. They conjointly report power savings of up to fifty eight once exploitation the projected. Implementation of multiplier comprises three steps: generation of style in applications. They show that our style considerably outperforms partial products, partial products reduction tree, and finally, a vector alternative approximate multipliers recently projected within the merge addition to produce final product from the sum and carry rows literature. generated from the reduction tree. Second step consumes more power. In this brief, approximation is applied in reduction tree stage. A 8-bit unsigned1 multiplier is used for illustration to describe the proposed method in approximation of multipliers. Consider two 8-bit unsigned.

EXISTING SYSTEM

Implementation of number contains 3 steps: generation of partial input operands α = _7 m=0 αm2m and β = _7n=0 βn2n. The partial product, partial product reduction tree, and eventually, a vector merge product am,n= αm • βn in Fig. 1 is the result of AND operation between addition to provide final product from the add and carry rows generated the bits of αm and βn. 1The proposed approximate technique can be from the reduction tree. Second step consumes additional power. during applied to signed multiplication including Booth multipliers as well, this temporary, approximation is applied in reduction tree stage. A 8-bit except it is not applied to sign extension bits.

MATERIALS AND METHODS

Participants

Number of Participants: 2 Number of Female Participants : 2 unsigned1 number is employed for illustration to explain the projected methodology in approximation of multipliers. think about 2 8-bit unsigned input operands α = eighty seven m=0 αm2m and β = _7n=0 βn2n.The partial product am,n= αm ・βn in Fig. one is that the results of AND operation between the bits of αm and βn.The projected Ethnicity or Cultural Background : I am from Tamil Nadu. Tamil Nadu approximate technique is applied to signed multiplication together with is home for historical, heritage, pilgrimage sites, hill stations, waterfalls Booth multipliers likewise, except it's not applied to sign extension bits and beaches. We get to learn about different thought processes, art, culture. Procedure Tabu search can be seen as an extension of hill climbing heuristics.

PROPOSED SYSTEM

Structure of AN inexact mechanical gadget is arranged The primary contrast between the two techniques is that, not normal for slope to more build execution also as decreasing the mistake rate. climbing heuristics, Tabu hunt does not stop in nearby optima but rather it Multiplication could be an essential activity in most sign procedure holds moving to the best arrangement inside the area of the calculations. Multipliers have gigantic space, long idleness and devour current arrangement, while expecting that this non-improving move can broadened control. So low-control number style has a crucial half in low-lead to the ID of a superior neighborhood ideal further in the power VLSI framework style. A framework is for the most part dictated by the procedure. So as to abstain from cycling between an answer and a neighborhood execution of the number because of the number is for the most part the part ideal that has been recently visited by the calculation, Tabu pursuit and a great deal of room overpowering inside the framework. hence upgrading the utilizations data structures called Tabu records. This rundown stores a given speed and space of the number is one among the chief style number of moves that would prompt a past neighborhood ideal. The size issues. Be that as it may, space and speed zone unit commonly clashing and substance of Tabu records rely upon the issue yet they are generally limitations so improvements in speed

prompts bigger zones. not exceptionally long (regularly transient memory is utilized for the rundowns) and Multiplication could be a numerical procedure that exemplify procedure of for the most part they contain the backwards of the last couple of alterations made to change up itself a given number of times. run the present arrangement. In view of the attributes of the issue in (multiplicand) is more itself assortment of times as given by another number hand, some Tabu limitations are characterized and Tabu records are developed (multiplier) to make an outcome (product).play a fundamental job in the present computerized by these imperatives. Tabu limitations are generally expressed to process and differed elective applications. during this segment, the make inversion or at times reiteration of certain moves outlandish. Effect of exploitation the arranged blowers for increase is researched. a fast (accurate) number is normally made out of 3.

blowers inside the CSA tree of a number prompts A rough approximations connected in all n bits in Multiplier1 and just in n −1 least number. 8x8 unsigned DADDA tree number is considered to critical part in Multiplier2. Multiplier1 and Multiplier2 accomplish evaluate the effect of exploitation the arranged blowers in critical decrease in region and power utilization contrasted and inexact multipliers. The arranged number uses inside the first half careful plans. With APP reserve funds being 87% and 58% for Multiplier1and AND entryways to think of all halfway product. inside the second Multiplier2 as for careful multipliers, they additionally outper structure This a large portion of, the estimated blowers arranged inside the past area article has been acknowledged for consideration in a future issue of this diary. Territory unit used inside the CSA. Substance is last as introduced, except for pagination.

Two variations of multipliers square measure anticipated. inside the first correlation with existing rough plans. They are likewise found to case (Multiplier1), guess is connected through and through sections of have better exactness when contrasted with existing rough multiplier halfway result of n-bit number, while in Multiplier2, surmised plans. The proposed multiplier plans canbe utilized in applications circuits square measure used in n − one least crucial sections. All with insignificant misfortune in yield quality while sparing huge influence and rough multipliers square measure intended for n = sixteen. The territory.

multipliers square measure upheld in Verilog and combined Approximate number for advanced picture process , DSP applications are abuse Synopsys style Compiler and a TSMC sixty five nm considered. Writing papers square measure considered for different methodologies galvanic cell library at the standard strategy corner, with temperature of rough number. Execution of number contains 3 stages twenty five °C and give voltage1 V anticipated multipliers is age of fractional item, halfway item decrease tree, and vector contrasted and existing surmised multipliers. In careful mechanical union expansion to supply last item from the aggregate and convey lines gadget style a couple of is utilized to style mechanical gadget fundamentally created from the decrease tree. Second step devours extra based multipliers ACM1, any place all segments square measure control. to downsize power and improve rough refinement, an approximated and ACM2, any place exclusively fifteen least imperative sections totally one of a kind mechanical gadget based generally inexact number square measure approximated.SSM kind = twelve and n = sixteen is anticipated. Estimated mechanical gadget is anticipated to any expansion expected for usage. palatopharyngoplasty style referenced for

execution also as lessening the mistake rate. Surmised 0.5 j = a couple of, k = a couple of is proposed and authorized underneath Dadda viper and full snake planned and recreation yields square measure tree structure. The halfway item framework of16-bit underneath planned referenced. Noval inexact mechanical gadget related to number (UDM) contains estimated 2 × a couple of halfway item incomplete item and decrease stages are style to a limited extent a couple of. Gathered related to real convey spare adders.

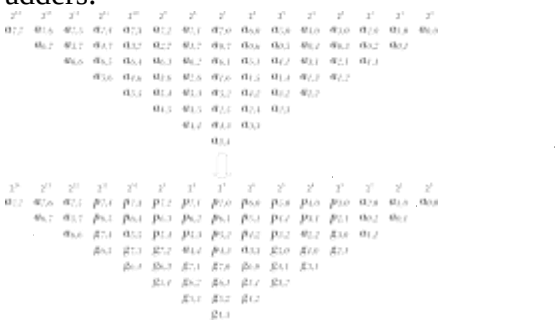


Fig. 1. Transformation of generated partial products into altered partial products.

TABLE I
PROBABILITY STATISTICS OF *Generate* SIGNALS

m	Probability of the *generate* elements being				P_{crr}
	all zero	one 1	two 1's	three 1's and more	
2	0.8789	0.1172	0.0039	-	0.00390
3	0.8240	0.1648	0.0110	0.00024	0.01124
4	0.7725	0.2060	0.0206	0.00093	0.02153

From statistical point of view, the partial product $a_{m,n}$has a probability of 1/4 of being 1. In the columns containing more than three partial products, the partial products $a_{m,n}$ and $a_{n,m}$ are combined to form *propogate* and *generate* signals as given in (1).The resulting *propogate* and *generate* signals form altere dpartial products $p_{m,n}$and $g_{m,n}$. From column 3 with weight 2^3 to column 11 with weight 2^{11}, the partial products $a_{m,n}$and $a_{n,m}$are replaced by altered partial products $p_{m,n}$and $g_{m,n}$. The original and transformed partial product matrices are shown in Fig. 1

$$p_{m,n} = a_{m,n} + a_{n,m}$$
$$g_{m,n} = a_{m,n} \cdot a_{n,m}. \quad (1)$$

The probability of the altered partial product $g_{m,n}$being one is 1/16, which is significantly lower than 1/4 of $a_{m,n}$. The probability of altered partial product $p_{m,n}$being one is 1/16 + 3/16 + 3/16 = 7/16, which is higher than $g_{m,n}$. These factors are considered, while applying approximation to the altered partial product matrix.

A. *Approximation of Altered Partial Products*$g_{m,n}$

The accumulation of *generate* signals is done columnwise. Aseach element has a probability of 1/16 of being one, two elements being 1 in the same column even decreases. For example, in a column with 4*generate* signals, probability of all numbers being 0 is $(1\ pr)^4$, only one element being one is $4\ pr(1\ pr)^3$, the probability of two elements being one in the column is $6\ pr^2\ (1\ pr)^2$, three ones is $4pr^3(1pr)$andprobabilityofallelementsbein g1ispr^4,where*pr*is 1/16. The probability statistics for a number of *generate* elements *m* in each column are given in TableI.

TABLE II
TRUTH TABLE OF APPROXIMATE HALF ADDER

Inputs		Exact Outputs		Approximate Outputs		Absolute Difference
$x1$	$x2$	*Carry*	*Sum*	*Carry*	*Sum*	
0	0	0	0	0 ✔	0 ✔	0
0	1	0	1	0✔	1✔	0
1	0	0	1	0✔	1✔	0
1	1	1	0	1✔	1✗	1

TABLE III
TRUTH TABLE OF APPROXIMATE FULL ADDER

Inputs			Exact Outputs		Approximate Outputs		Absolute Difference
$x1$	$x2$	$x3$	*Carry*	*Sum*	*Carry*	*Sum*	
0	0	0	0	0	0 ✔	0 ✔	0
0	0	1	0	1	0✔	1✔	0
0	1	0	0	1	0✔	1✔	0
0	1	1	1	0	1✔	0✔	0
1	0	0	0	1	0✔	1✔	0
1	0	1	1	0	1✔	0✔	0
1	1	0	1	0	0 ✗	1 ✗	1
1	1	1	1	1	1 ✔	0 ✗	1

B. *Approximation of Other Partial Products*

The accumulation of other partial products with probability 1/4 for$a_{m,n}$and 7/16 for

*pm,n*uses approximate circuits. Approximate half-adder, full-adder, and 4-2 compressor are proposed for their accumulation. *Carry* and *Sum* are two outputs of these approximate circuits. Since *Carry* has higher weight of binary bit, error in *Carry* bit will contribute more by producing error difference of two in the output. Approximation is handled in such a way that the absolute difference between actual output andapproximate output is always
maintained as one. Hence *Carry* outputs are approximated only for the cases, where *Sum* is approximated.

In adders and compressors, XOR gates tend to contribute to high area and delay. For approximating half-adder, XOR gate of *Sum* is replaced with OR gate as given in (2). This results in one error in the *Sum* computation as seen in the truth table of approximate half-adder inTable II. A tick mark denotes that approximate output matches with correct output and cross mark denotes mismatch

$$Sum = x1 + x2$$
$$Carry = x1 \cdot x2. \quad (2)$$

In the approximation of full-adder, one of the two XOR gates is replaced with OR gate in *Sum* calculation. This results in error in last two cases out of eight cases. *Carry* is modified as in (3) introducing one error. This provides more simplification, while maintaining the difference between original and approximate value as one. The truth table of approximate full-adder is given in Table III

$$W = (x1 + x2)$$
$$Sum = W \oplus x3$$

Multiplier	Mean Relative Error	Normalized Error Distance
	7.63×10^{-2}	1.78×10^{-2}
	2.44×10^{-4}	7.10×10^{-6}
	16.6	4.96×10^{-2}
	2.30×10^{-3}	6.36×10^{-6}
	6.34×10^{-4}	1.07×10^{-4}
	8.98×10^{-4}	4.58×10^{-5}
	3.32×10^{-2}	1.39×10^{-2}

RANKING OF APPROXIMATE MULTIPLIERS IN TERMS OF DESIGN AND ERROR METRICS

Approximate Multiplier Type	APP Gain	PDP Gain	NED	MRE
Multiplier1	1	2	6	6
Multiplier2	3	3	2	1
ACM1 [5]	2	1	7	7
ACM2 [5]	4	4	1	4
SSM [6]	5	5	4	2
PPP [7]	7	7	3	3
UDM [8]	6	6	5	5

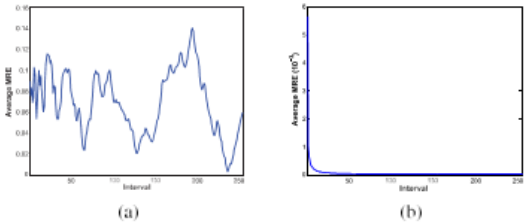
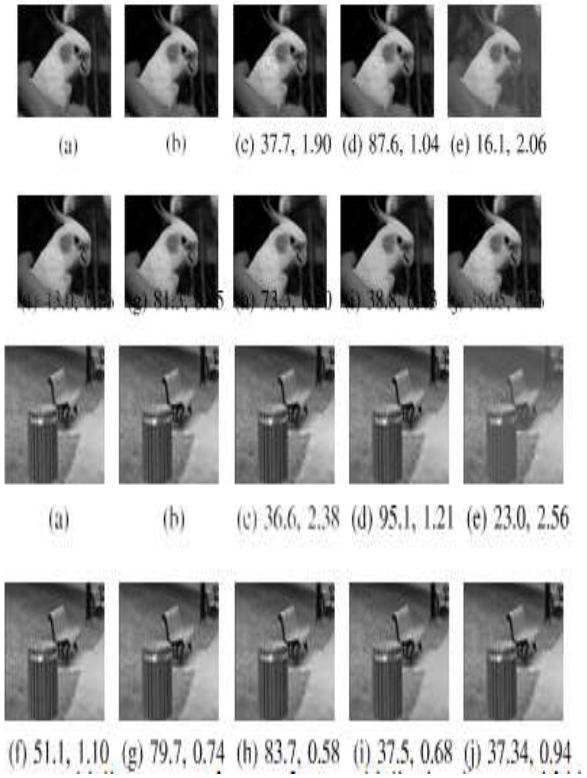
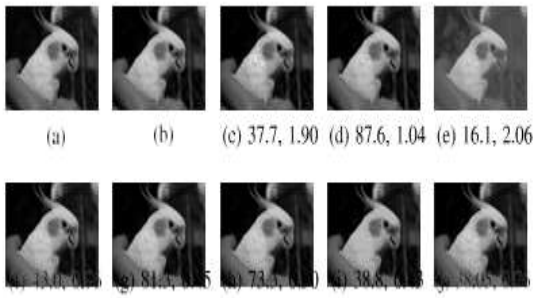


Fig. 2.MRE distribution of (a) Multiplier1 and (b) Multiplier2.





Conclusion:

In this brief, to propose proficient inexact multipliers, fractional results of the multiplier are altered utilizing create and engender signals. Guess is connected utilizing basic OR entryway for modified create fractional items. Estimated half-snake, full-viper, and4-2 blower are proposed to decrease staying halfway items .Two variations of surmised multipliers are proposed. Two variations of inexact multipliers are proposed, where approx-Imation is connected in all n bits in Multiplier1 and just in n 1 least critical part in Multiplier2. Multiplier1 and Multiplier2 accomplish noteworthy decrease in zone and power utilization contrasted and accurate plans. With APP reserve funds being 87% and 58% for Multiplier1 and Multiplier2 with respect to exact multipliers, they likewise beat.

REFERENCES

[1] *V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power dig- ital signal processing using approximate adders," IEEE Trans. Comput.- Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan.2013.*

[2] *E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncation scheme for parallel multipliers," in Proc. 31st Asilomar Conf. Signals, Circuits Syst., Nov. 1998, pp.1178–1182.*

[3] *K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi, "Design of low-error fixed-width modified booth multiplier," IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., vol. 12, no. 5, pp. 522–531, May2004.*

[4] *H.R.Mahdiani,A.Ahmadi,S.M.Fakhraie,andC .Lucas,"Bio-inspired 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] *S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal process- ing and classification applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun.2015.*

[6] *G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, "Design-efficient approximate multiplication circuits through partial product perforation," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 10, pp. 3105–3117, Oct.2016.*

[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.-H. Lin and C. Lin, "High accuracy approximate multiplier with error correction," in Proc. IEEE 31st Int. Conf. Comput. Design, Sep. 2013, pp.33–38.*

[9] *C. Liu, J. Han, and F. Lombardi, "Alow-power, high-performance approximate multiplier with configurable partial error recovery," inProc. Conf. Exhibit. (DATE), 2014, pp.1–4.*

[10] *R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Oct. 2011, pp.667–673.*

[11] *J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol. 63, no.9,pp.1760–1771,Sep.2013.*