Power, Area and Delay Efficient Approximate Multiplier Design

[1]Shauvik Panda, [2]Alpana Agarwal

[1]Master of Technology, [2]Associate Professor
[1]Electronics & Communication Engineering Department,
[1]TIET, Patiala, India

Abstract : ***Approximation can improve design complexity, power efficiency and performance metrics in any computation. But the computation system should be error tolerant. In this paper we designed an approximate multiplier with altered partial products based on probability. Accumulation circuits are also designed with approximation and applied based on probability. With modification of existing Carry select adder (CSA) with binary to excess 1 converter (BEC) this approach results into 45.3%, 44.75% and 30.21% power, speed and area savings respectively, compared to exact. Further multi Vt design approach results into 21.1% and 24.8% power and speed improvement. Proposed multiplier's performance is assessed with image processing application with peak signal to noise ratio (PSNR) as performance deciding parameter.***

IndexTerms – Approximation, Low power, Low error, Multiplier.

I. INTRODUCTION

Data mining and multimedia signal processing are some of the examples which are error tolerant. For this type of applications exactness of the output is not necessary. Their computational circuits can be replaced by their approximate counter parts in order to improve performance. This gives the rise to designing of approximate adders and multipliers which are very essential for digital circuits.

In [1], approximate multiplier is designed with a mechanism of removing partial product generation and accumulation for lower order input operands. Depending upon the input operands the non-multiplication part will produce the output. For higher order input operands normal multiplication is performed. Another approach of approximation is truncation of output [2] by one bit in case of multiplier as probability of using that output is very low but approximation will lead to low power and high speed operation. Kind of similar approach is presented in [3], where Wallace tree multiplier is introduced with approximated 4:2 compressor. In [4], full adders are approximated at transistor level change for sum and carry generation. This circuits can be used with any multipliers during their accumulation stage of partial products. In [5], approximation and error correction combining structure is discussed where approximation is done with the preprocessed input data bits interchangeability. Truncation and approximation is applied for multiplier design [6] for lower order input bits for application of error-resilient systems. In [7], multiplier is designed with the approximation of 4:2 compressor with a drawback of having non-zero output even in the case of all inputs as zero. In [8], static segment multiplier is proposed where selected segment of input operands are multiplied instead of whole. Voltage over scaling effect with delay of the circuit is used for approximation in [9], where power is reduced with timing violation. In [10], partial product perforation is done on approximate multiplier design to omit some of the partial products in its generation stage. In [11], error metrics are discussed such as error distance (ED) which is the difference between normal and approximate output. Also mean error distance (MED), mean relative error distance (MRED), normalized error distance (NED) are also defined which can be used to evaluate the error in the approximate circuits. In [12], multi Vt design benefits in terms of leakage power reduction is being discussed.

All these previous works focused on reduction of logical complexity of adder and compressors used in partial product accumulation. This work at first altered the partial products depends on the probability. Then, systematic accumulation is done to maintain the error output as low as possible. Both approximated arithmetic units and probabilistic alteration will lead into low error performance enhanced multiplier design.

Rest of the paper is organized as follows. Section II deals with the proposed architecture details used in the approximate multiplier design. Section III provides result analysis of the approach. Utilization of this design in image processing application is discussed in section IV. Section V tells the conclusion.

II. PRPOSED ARCHITECTURE

Design of approximate multiplier consists of three stages as partial product generation, reduction tree of partial products and a vector combine expansion to deliver the final product from the last rows generated from the reduction tree. Power is mostly consumed due to the second stage whereas delay is mostly associated with third stage. Approximation applied to reduction tree to reduce the power. Faster adder approach and multi Vt based implementation further improve design's performance parameters.

An 8 bit unsigned DADDA multiplier is taken to apply the proposed techniques. Considering two input operands x and y, the partial products are usually the output of AND operation between the input operands, $z_{m,n} = x_m y_n$ where m and n are the bit positions. The partial product $z_{m,n}$ is being 1 with probability of 1/4. In DADDA architecture the partial products are structured in columnar format based on their binary weights. Each column where the products are more than 3, we replaced the product terms with propagate and generate signals as given in Eq. 1,

$$p_{m,n} = z_{m,n} + z_{n,m}$$

$$g_{m,n} = z_{m,n}.z_{n,m} \quad (1)$$

The probability of being 1 for $p_{m,n}$ is 7/16 whereas for $g_{m,n}$ is 1/16. This difference is resulted into different approach for their reduction approach.

2.1 Approximation of $g_{m,n}$

Due to each element have only 1/16 chance of being 1 for generate signals, during column wise accumulation of these products is done using simply OR gates. Error probability with these approach is shown in Table 1 where pr being 1/16.

Table 1: Probability scenario of generate signal accumulation

No. of elements in a column	Probability of being all elements as 0	Probability of being only one element as 1	Probability of being two elements as 1	Probability of being more than two elements 1	Probability of error
2	$(1-pr)^2$	$2pr(1-pr)$	pr^2	-	pr^2
3	$(1-pr)^3$	$3pr(1-pr)^2$	$3pr^2(1-pr)$	pr^3	$3pr^2(1-pr)+pr^3$
4	$(1-pr)^4$	$4pr(1-pr)^3$	$6pr^2(1-pr)^2$	$4pr^3(1-pr)+pr^4$	$6pr^2(1-pr)^2+4pr^3(1-pr)+pr^4$

Error is only generated when two or more than two elements become 1. So as the generate signal element increases the error probability is increased in a linear fashion.

2.2 Approximation of $z_{m,n}$ and $p_{m,n}$

Due to higher probability of 1/4 and 7/16 for normal partial product and propagate signals approximate half adder, full adder and 4:2 compressor is used for the accumulation purpose. Sum and carry being two outputs of these circuits where carry having higher weight of binary bit, approximation could result into higher error propagation. Approximation of these circuits are designed to have absolute difference of one between actual and approximate output in order to achieve carry approximation only when sum is approximated. Also all these circuit uses XOR gates which tends to have more area and delay which is replaced for approximation purpose.

In half adder circuit, approximation took place in sum calculation. XOR gate used in sum calculation is replaced by OR gate in the approximate half adder whereas carry circuit is kept same as shown in Eq. 2.

$$Sum = I1 + I2$$
$$Carry = I1.I2 \quad (2)$$

This results into 1 error case out of four cases in sum calculation.

In approximate full adder circuit one of the XOR gate is replaced by OR gate in sum calculation. Also to make absolute difference as 1 carry circuit is modified which uses 2 lesser gates than original circuit. The modifications are shown in Eq. 3.

$$Sum = (I1 + I2) \oplus I3$$
$$Carry = (I1 + I2).I3 \quad (3)$$

These approximation costs two error in sum and one error in carry computation out of eight possible cases.

In approximate 4:2 compressor design three output bits is reduced to two. This is because three output bits is only required when all four inputs are 1 which has a probability of 1/16 to happen. Removal of third output will cause significant change in number of gates required. The sum architecture with approximation uses only two XOR gates and one OR gate instead of five XOR gates in conventional architecture. An extra term of AND operation of all inputs are ORed to have sum as 1 when all inputs are one. Carry circuitry is approximated using only OR and AND gate. Sum and carry expressions are given in Eq. 4.

$$Sum = (I1 \oplus I2) + (I3 \oplus I4) + I1.I2.I3.I4$$
$$Carry = I1.I2 + I3.I4 \quad (4)$$

This approximation will results into five error cases for both sum and carry out of 16 cases.

Figure 1 is showing the partial product alteration and reduction process for 8x8 unsigned DADDA multiplier.

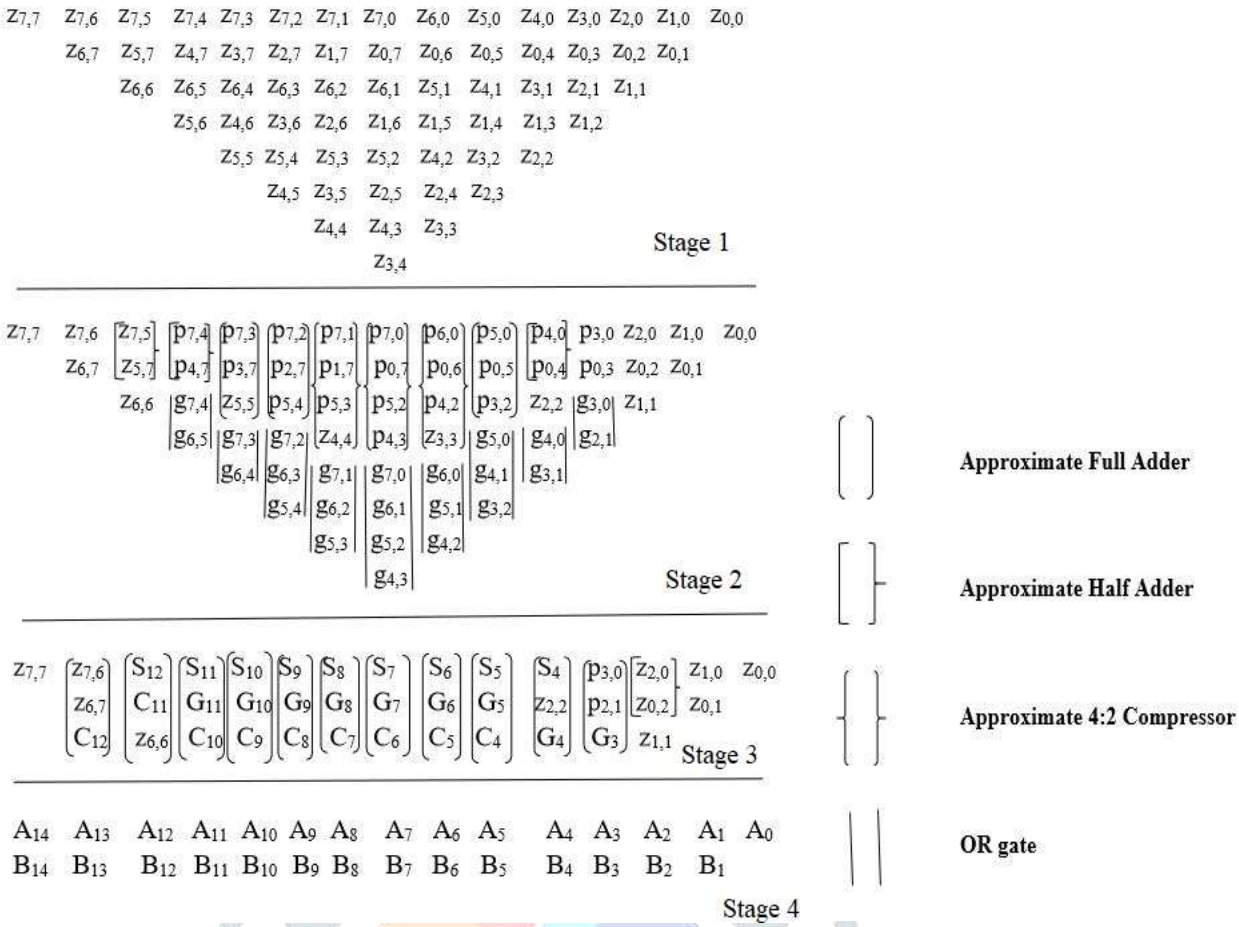

Fig. 1 Partial Product Processing

Stage 1 is usual partial product generation from the input operands using AND operation. Stage 2 deals with propagate and generate signal production from normal partial product using Eq. 1. Stage 2 and stage 3 will show the use of approximate circuits for partial product reduction where G_i, S_i and C_i are the outputs of OR gate and other approximate circuits. Stage 4 is having final two operands as A_i and B_i, which will fed to vector merge adder to produce final result.

2.3 Carry select adder (CSA) using binary to excess 1 converter (BEC) [13]

Carry select adder has a higher speed of operation because of its parallel operation uses two ripple carry adder (RCA). Each RCA uses different carry values as there is only two possibilities i.e. 1 and 0. But it has higher area and power overhead. BEC is nothing but adding 1 to binary bit. So the requirement of RCA with carry input as 1 in CSA architecture is replaced by BEC. BEC structure used in CSA should have 1 bit higher structure than the RCA used. A 4-bit BEC expressions is shown in Eq. 5.

$$B0 = \sim I0$$
$$B1 = I0 \oplus I1$$
$$B2 = I2 \oplus (I1.I0)$$
$$B3 = I3 \oplus (I2.I1.I0) \quad (5)$$

Figure 2 is showing the used 16-bit CSA-BEC architecture for 16-bit addition required for implementing the 8x8 multiplier.

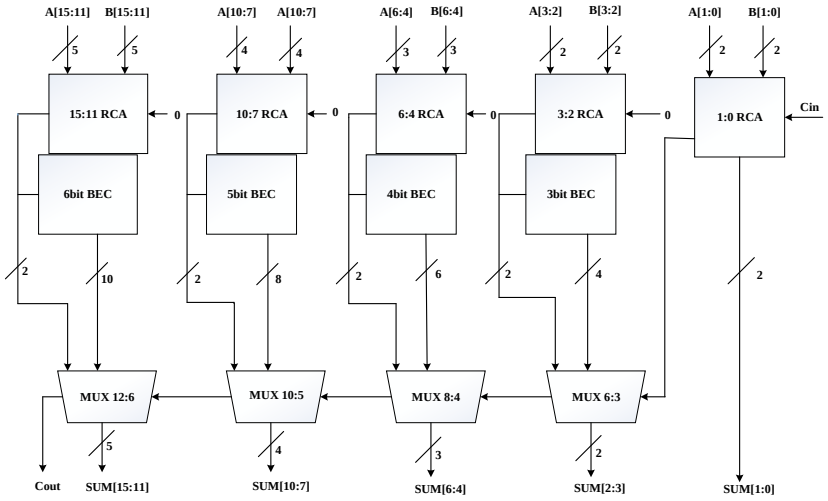

Fig. 2 16-bit CSA-BEC Architecture

2.4 Multi Vt design

High Vt cells having advantage of low leakage power but with higher delay with vice versa relation in case of low Vt cells. This multiplier is designed with multi Vt based approach with modules having higher delay designed with low Vt cells where higher leakage consumed modules with high Vt cells. This approach will led into low power and high speed operation.

III. RESULTS AND DISCUSSION

8 bit multiplier is implemented using Verilog HDL and synthesized using Synopsys design compiler and SAED 32 nm standard cell library at typical process corner with supply voltage of 1.05 and temperature 25°c. For comparison purpose four multiplier structures are taken which are exact multiplier, approximate multiplier with RCA, CSA, and modified CSA with BEC. The simulated waveform from Verilog compiler will show the functionality of the multiplier as shown in Fig. 3. Both exact and approximate outputs are shown for the comparison of the outputs to figure out the deviations due to approximation.

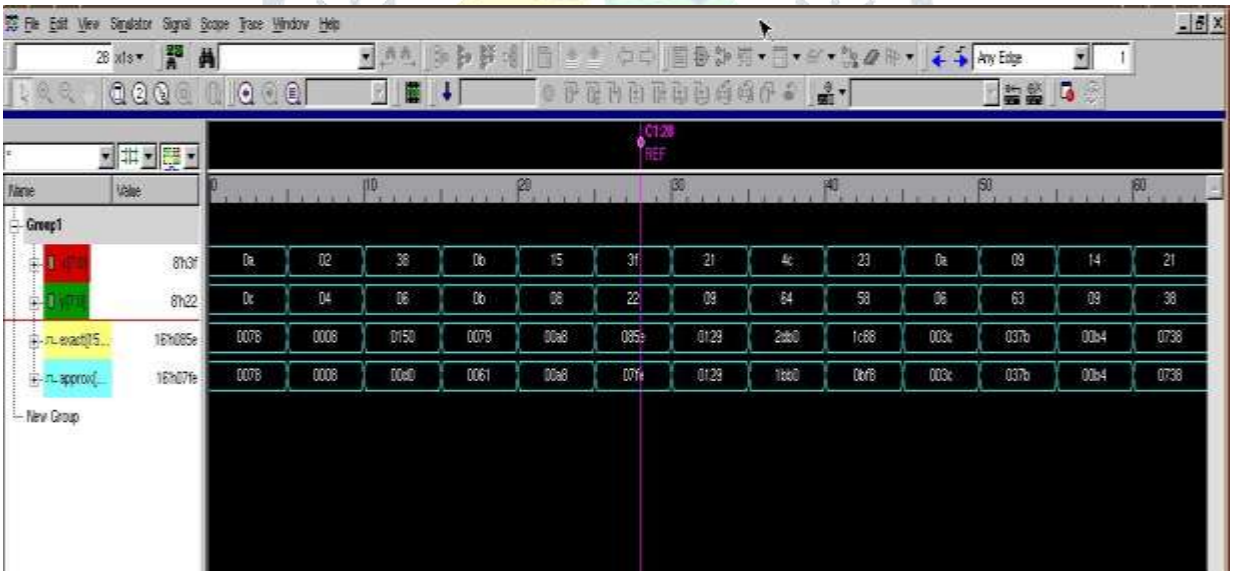

Fig. 3 Simulated Waveform of Exact and Approximate Multiplier

From dc reports we get the power, area, delay, power delay product (PDP) and area power product (APP) information as shown in Table 2.

Table 2 Synthesis Result

Multiplier Type	Area(μm^2)	Delay(ns)	Power(μW)	PDP(μJ)	APP($\mu m^2 . \mu W$)10^3
Exact	780.75	2.19	167.09	365.93	130.455
Approximate with	540.24	2.08	81.28	169.06	43.91

RCA					
Approximate with CSA	703	1.37	114.55	156.93	80.528
Approximate with CSA-BEC	545.66	1.21	91.40	110.59	49.873

Approximate multiplier with RCA, CSA and CSA-BEC shown 51.35%, 31.44% and 45.3% power reduction compared to exact conventional structure. Also these three circuits shown 5%, 37.44%, 44.3% speed improvement whereas 30.8%, 9.96%, 30.11% area savings respectively. High area and power consumed in CSA architecture is reduced by modified CSA with BEC.

The multiplier design is divided into two modules, one contains partial product alteration and accumulation and other contains final stage adder. From timing and power analysis we see that the module contains final stage adder having high delay contribution and the other module with high leakage contribution. So we designed the final stage adder with low Vt cells whereas other module with high Vt cells. The improved result with this approach is shown in Table 3.

Table 3 Multi Vt Based Synthesis Result

Multiplier Type	**Delay(ns)**	**Power(μW)**	**PDP(μJ)**
Exact	1.88	135.66	255
Approximate with RCA	1.67	61.89	103.34
Approximate with CSA	1.04	95.38	99.19
Approximate with CSA-BEC	0.91	72.13	65.64

Multi Vt based implement will further improve the delay of approximate multiplier with RCA, CSA, CSA-BEC by 19.71%, 24.08% and 24.8% whereas power reductions are 63.4%, 39.22% and 34.77% respectively.

Error analysis is also done using randomly generated input vectors with the evaluation error metrics such as MED, NED and MRED which is shown in Table 4.

Table 4 Error Metrics

Parameter	**Value**
Input Vector	10^8
MED	3615.29
NED	5.56×10^{-2}
MRED	2.14×10^{-1}

IV.APPLICATION OF IMAGE PROCESSING

This approximate multiplier is used in the Gaussian filter for image processing to reduce the Gaussian noise. Filtering is performed by convolution of original image with the defined Gaussian mask. Convolution is done using this approximate multiplier with 2D 3x3 Gaussian mask and 8x8 size of input image [14]. The algorithm is coded and implemented with the help of MATLAB and Verilog. PSNR is used as figure of merit to assess the quality of approximate multiplier. PSNR is based on mean-square error found between original image and filtered image. Processed image is shown in fig. 4.

(a) (b) (c)

Fig. 4 Processed Image- (a) Original Image (b) Processed Image with Exact Multiplier (c) Processed Image Using Approximate Multiplier

PSNR value corresponding to approximate multiplier processing is 53.97 dB whereas using exact multiplier is 56.11dB. As we can see only 3.8% reduction in PSNR values.

V. CONCLUSION

In this paper, the proposed approximate multiplier uses propagate and generate signals which are modified from the partial products. Simple OR gates are used for generate partial product where approximate half adder, full adder and 4:2 compressor are used for other partial products accumulation. Three architecture of adder is evaluated where CSA-BEC structure stands best. The use of CSA-BEC, our design results into 70%

and 67.8% savings in terms of PDP and APP in comparison with exact design. Further multi Vt design approach will results into 39.7% reduction of PDP. Error metrics with low values will conform to better precision result of our design. This design can be implemented for different applications where significant performance improvement can be achieved with minimum loss of output quality.

REFERENCES

[1] Khaing Yin Kyaw, Wanh Ling Goh and Kiat Seng Yeo, 2010. Low power high speed multiplier for error tolerant systems. IEEE Intl. Conf. Electron Devices and Solid-State Circuits (EDSSC): 1-4.
[2] P. Kulkarni, P. Gupta, and M. D. Ercegovac, 2011. Trading Accuracy for Power with an Underdesigned Multiplier Architecture. IEEE 24th Intl. Conf. on VLSI Design:346-351.
[3] Chia-Hao Lin and Ing-Chao Lin, 2013. High accuracy approximate multiplier with error correction. 31st Proc. IEEE Int. Conf. Computer Design (ICCD): 33– 38.
[4] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, 2013. Low-Power Digital Signal Processing Using Approximate Adders. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32(1): 124-137. C. Liu, J. Han, and F. Lombardi, **[5]** 2014. A Low-Power, High-Performance Approximate Multiplier with Configurable Partial Error Recovery. Design, Automation and Test in Europe Conference and Exhibition (DATE): 1-4.
[6] Z. Yang, J. Han, and F. Lombard, 2015. Approximate Compressors for Error-Resilient Multiplier Design. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS): 183-186.
[7] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, 2015. Design and Analysis of Approximate Compressors for Multiplication. IEEE Transactions on Computers, 64(4): 984-994.
[8] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim , 2015. Energy-Efficient Approximate Multiplication for Digital Signal Processing and Classification Applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 23(6): 1180-1184.
[9] David May and Walter Stechele, 2016. Voltage Over-Scaling in Sequential Circuits for Approximate Computing. 11th Design and Technology of Integrated Systems in Nanoscale Era (DTIS): 1-6.
[10] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, 2016. Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 24(10): 3105-3117.
[11] J. Liang, J. Han, and F. Lombard, 2013. New Metrics for the Reliability of Approximate and Probabilistic Adders. IEEE Transactions on Computers, 62(9): 1760-1771.
[12] M. Srivastav, S.S.S.P. Rao and H. Bhatnagar, 2005. Power reduction technique using multi-Vt libraries. Fifth International Workshop on System-on-Chip for Real-Time Applications (IWSOC): 363-367.
[13] Deepak Kumar Patel, Raksha Chouksey and Minal Saxena, 2015. An Efficient VLSI Architecture for Carry Select Adder Without Multiplexer. International Journal of Computer Applications, 127(9): 37-40.
[14] Leila kabbai, Anissa Sghaier, Ali Douik and Mohsen Machhout, 2016. FPGA implementation of filtered image using 2D Gaussian filter. International Journal of Advanced Computer Science and Applications (IJACSA), 7(7): 514-520.