Design of High Speed and Low Power Approximate Multiplier for Image and Digital Signal Processing Applications

Ranjith Kumar T, Saravana Kumar R, Dharani S and Asuvanti M A
Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, India
ranjithkumar.vd18@bitsathy.ac.in

Abstract. Truncation based approximate multiplier is a scalable multiplier in which truncation is performed based on the leading one bit positions in each and every input operands, hence the number of partial products are reduced. In this multiplier design the multiplication process is done by addition, shifting and multiplication processes which has a very good improvement in space occupation and power consumption when compared to the exact multiplier. To enhance the multipliers speed each and every input operand in the multiplication side are truncated using the method given below. Based upon the leading 1’s the input operands, the inputs are truncated show that the accuracy does not depend on input operand’s width. Hence this multiplier is scalable. Very large improvement in multiplier’s design parameters like power and area consumption are obtained which is independent of the width of the input operand. To find the working efficiency of this multiplier we have compared the design parameters with some recently proposed multipliers and the exact multiplier. They obtained result states that the proposed multiplier enhances power consumption, delay and area up to 90%, 80% and 95% when compared to be exact multiplier. Also our design outsmarts other multipliers in parameters like power consumption, speed and area. Proposed approximate multiplier has zero mean value Gaussian distribution. We have figured it out it by MATLAB applications and a good PSNR and MSE is obtained. The output shows that the degradation in quality is almost negligible. Also we recommend Truncation based approximate multiplier where power consumption has to be adjusted in multiplication operation.

1 Introduction
Power consumption is very much important parameter in the design of digital circuit systems. To reduce power consumption and to increase the speed one of the methods called approximate computing is used. It's result may not be correct so it is used in error Resilient applications. Here are some of the examples of those applications which includes audio processing, data mining, image processing and machine learning. In case of signal processing applications, major portion in power consumption is because of the arithmetic operations. This multiplication operations are repeatedly used and has high latency and power consumption. Hence this approximate multipliers for very good suitable for reducing time delay and faster processing.

There are three stages in binary multiplication process. The first stage is where the partial products are produced based upon the input operands. The partial product terms are stored in the next stage until only two rows exist which are then added by using fast adders in the final third stage. The approximation technique is introduced in the first stage in order to reduce the number of partial products. It may also be applied during the second step in order to reduce the latency for energy consumption. These two parameters in multiplication operation is affected by the adders architecture which are used in final step of multiplication.
Important parts in this paper are summarised as follows

1) A scalable approximate multiplier which finds leading 1’s in the bit terms to perform truncation in order to improve multiplication operation's accuracy
2) Truncation(t) and rounding(h) parameters to decide power consumption, delay and accuracy
3) Circuit level design of the Approximate multiplier
4) Image processing application in MATLAB software

2 Related Work
We discuss some research efforts to develop estimated multipliers in this section.

A Booth multiplier is studied. Thus, approximate encoders can be used to generate partial products to reduce the complexity [6]. In [7] the partial products were generated and accumulated about by an approximate radix-4 booth multiplier. In addition, in the [8] radix-8 booth multiplier, which produced the least significant triple-multiplicand bits using approximate adjusters, was proposed earlier.

Then an exact Wallace tree multiplier is analysed as shown in Figure 1. The delay for a N X N Wallace tree multiplier is reduced to log2N [9]. Hence it is faster than an exact multiplier but making the multiplier much complex thereby taking a huge computational time reducing its efficiency. Here during reduction process, each row were grouped into three set numbers. (3,2) counters are reduced to (2,2) sets [10]. The RTL schematic diagram is obtained as in Figure 2.
3 Proposed Approximate Multiplier

3.1 TAM

The multiplying result of A and B is calculated as

\[ A \times B = 2^{kA +kB} \times DA \times DB \]  

(1)

k is the position of the leading 1 bit. \(DA\) and \(DB\) widths are equal to that of A and B for the exact time and power of \(DA \times DB\). The remaining part shows the fractional portion of D in \(R = D - 1\)  

(2)

The range 0,0–1.0 is divided into \(T\) equal halves where \(T\) is given by

\[ T = 2^h \]  

(3)

in which \(h\) indicates the positive integer(an arbitrary value) which is there in our design parameters. The duration of every section is naturally equivalent to 1/T.

In order to find \(Y_{APX}\), only \(h\) most important \(Y\) bits must be considered. If, for instance [11], \(S_{4(h-2)}\) is zero in two of the most significant bits of \(Y\), this means \(0 < Y < 1/4\). So, \(1/8 = (0.001)_{2}\) is chosen as \(Y_{APX}\). If \(2/4 < Y < 3/4\), \(Y_{APX}\) is thus roughly approximate as \(5/8 = (0.101)_{2}\), then \(Y_{APX}\) is the highest bits of \(Y\).

\[ A \times B = 2^{kA +kB} \times (1 + Y_A + Y_B + Y_A \times Y_B) \]  

(4)

Its approximate is written as

\[ A \times B \approx 2^{kA +kB} \times (1 + Y_A + Y_B + (Y_A)_{APX} \times (Y_B)_{APX}) \]  

(5)

"1" in the term 1+(Y_A)t+(Y_B)t+(YA)APX(YB)APX is added as shown in Figure 3. The number of partial product terms in a 16-bit multiplier is 256, which must be totalled in order to produce the final result while only 31 of the partial products are maintained in the proposed process.

Finally, for non-signed operands, the proposed multiplication approach is feasible.

**Figure 3.** Example of Truncated 16 bit approximate multiplier (3, 7).

A = 11761 and B = 2482.

Approximate result [(A× B)_{APX}] = 28 901 376.

Exact result [(A× B)_{Exact}] = 29 190 802.

Absolute error = 289 426 which is about 0.99% of the exact output (the error is less than 1% in this case).

The precision of the approach proposed is dependent on the validity of \(t\) and \(h\) and the RTL schematic is shown in Figure 4.
4 Implementation

The approximate multiplier block consists of two inputs A and B which are fed to the absolute approximate unit [12]. This block inverts the input value if it is high. So many LUT’s and gates are formed up for the circuit design. Then these two outputs are given as input to leading 1’s detector and truncation unit. Leading 1’s detector output is given to truncation and shifter unit through arithmetic unit and then the multiplied approximate result is obtained. The RTL schematic is given in Figure 5.

Figure 4. RTL schematic of truncated approximate multiplier

Figure 5. RTL schematic of truncated mux based approximate multiplier
5 Results and Accuracy Analysis

5.1 Design Parameters Extraction

5.1.1 8 Bit Multipliers

The use of this approximate multiplier can increase by up to 32%, 77%, 84%, and 90% in terms of area, power, speed and delay.

5.1.2 16 Bit Multipliers

The area is 72% smaller than that of Booth multiplier. With an efficiency value of 90%, this structure improves the multiplier speed by approximately 18% while consuming 86% less energy, while it occupies 70% less of the area than the exact Wallace multiplier.

5.1.3 32 Bit Multipliers

The use of the truncated muxed approximate multiplier structure results in an improvement in the latency, power, space, energy, EDP, and PDA respectively of approximately 94%, 85%, 95% and 96%.

The Figure 6, Figure 7 and Figure 8 shows the outputs of proposed multiplier and other multipliers.

The multiplier’s parameters like area, power, speed are compared with some other existing approximate multipliers in this section as shown in table 1.

Figure 6. Output result for Wallace multiplier
Figure 7. Output result for Truncated approximate multiplier

Figure 8. Output result for truncated mux based approximate multiplier

Table 1. Comparison of area and power consumption for the proposed multiplier with other multipliers

| MULTIPLIER                  | AREA  | POWER(mW) |
|-----------------------------|-------|-----------|
| Wallace                     | 20361 | 184       |
| Truncated Approx. Multiplier | 3656  | 83        |
| Modified Truncated Approx. Multiplier | 2735  | 76        |
6 Applications

In this section, the results are analysed for the proposed algorithm with the help of MATLAB. The scalability of the multiplier is examined by modifying the length of the bits and measuring the relative mistake of the multiplier. The efficiency of the proposed truncated muxed approximate multiplier is compared with other existing exact multipliers. With 8-bit multipliers, 65,536 numbers have been used to extract accuracy [13].

6.1 Image Processing Application

AC is a solution to enhance computation efficiency in image processing applications, as mentioned previously. We have used this in different applications of image processing, such as sharpening and JPEG compression, to evaluate the efficiency of the proposed multipliers. There have been a number of benchmarks in this section, including Baboon and Lena. The PSNR ratio and the Structural Similarity (SSIM) of approximate results were extracted using MATLAB simulations in the sharpening application compared with precise output images.

Furthermore, the Multiplier and Accumulator (MAC) module was installed to implement a sharper unit and the MAC energy consumption was measured for various non-signed 8-bit multipliers.

The output in the below Fig 9 shows that the difference between the MATLAB output and the VLSI output is almost negligible thus increasing the efficiency of the proposed multiplier structure. A very good signal to noise ratio and mean square error is obtained is compared in Table 2.

![Image multiplication process using MATLAB](image_url)

**Figure 9.** Application in Image multiplication process using MATLAB.

| MULTIPLIER | PSNR   | MSE      |
|------------|--------|----------|
| Wallace    | 48.1309| 1.0000   |
| Tuncated Approx. Multiplier | 9.2893 | 7.6587e+03 |
| Modified Truncated Approx. Multiplier | 9.2102 | 7.7994e+03 |

**Table 2.** Comparison table of PSNR and MSE for the multipliers
7 Conclusion
Thus in this paper the input operands were truncated to 2 different lengths: t and h. So in order to maintain the image quality even though compressing the image this multiplier can be used. In terms of speed, area and energy the proposed multiplier could be employed. When comparing the Wallace multiplier with the proposed multiplier, the 32-bit variant of the proposed has an improvement in power consumption of about 92%. The multiplication latency and power usage are increased and decreased by 4% –41% and 89% –97% respectively, in comparison with that of the other multipliers. The area, delay, and the power consumption of the proposed multiplier is increased as the multiplier width increases in comparison to that of the other multiplier. So in image processing and in JPEG, MPEG compression applications this multiplier with high accuracy is highly advantageous and is always recommended.

References
[1] Naraynamoorthy S, Moghaddam H A, Liu Z, Park T, and Kim N S 2015 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.
[2] Vahdat S, Kamal M, Afzali-Kusha A and Pedram M 2017 LETAM: A low energy truncation-based approximate multiplier Comput. Elect. Eng. vol. 63 pp. 1–17.
[3] Pedram M 2017 RoBa multiplier: A rounding-based approximate multiplier for high-speed yet energy-efficient digital signal processing, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25 no. 2 pp. 393–401.
[4] Leon V, Zervakis G, Soudris D and Pekmestzi K 2018 Approximate hybrid high radix encoding for energy-efficient inexact multipliers, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26 no. 3 pp. 421–430.
[5] Ko H J and Hsiao S F 2011 Design and application of faithfully rounded and truncated multipliers with combined deletion, reduction, truncation, and rounding, IEEE Trans. Circuits Syst. II, Exp. Briefs vol. 58 no. 5, pp. 304–308.
[6] Liu W, Qian L, Wang C, Jiang H, Han J and Lombardi F 2017 Design of approximate radix-4 booth multipliers for error-tolerant computing, IEEE Trans. Comput., vol. 66 no. 8 pp. 1435–1441.
[7] Venkatachalam S, Lee H J and Ko S B 2018 Power efficient approximate booth multiplier in Proc.IEEE Int. Symp. Circuits Syst. (ISCAS) Florence Italy pp. 1-4.
[8] Jiang H, Han J, Qiao F and Lombardi F 2016 Approximate radix-8 booth multipliers for low power and high-performance operation, IEEE Trans. Comput., vol. 65 no. 8 pp. 2638–2644.
[9] Combet M, Van Zonneveld H and Verbeek L 1965 Computation of the base two logarithm of binary numbers IEEE Trans. Electron. Comput. vol. EC-14 no. 6 pp. 863–867.
[10] Mitchell J N 1962 Computer multiplication and division using binary logarithms IRE Trans. Electron. Comput. vol. EC-11 no. 4 pp. 512–517.
[11] The USC-SIPI Image Database. Accessed: Jun. 2017 [Online]. Available http://sipi.usc.edu/database
[12] Wang Z, Bovik A C, Sheikh H R and Simoncelli E P 2004 Image quality assessment: From error visibility to structural similarity IEEE Trans. Image Process. vol. 13 no. 4 pp. 600–612.
[13] Myler H R and Weeks A R 2009 The Pocket Handbook of Image Processing Algorithms in C. Englewood Cliffs NJ USA: Prentice-Hall.