RoBA Multiplier: A Rounding-Based Approximate Multiplier for High-Speed yet Energy Efficient Digital Signal Processing

Nekkanti Gowthami[1], Pallapothu. Naveen Trinadh[2]
[12]Assistant professor vkrvnb&agkcolg of engg ,gudivada

Abstract:
In this paper, we have a tendency to propose associate degree approximate number that's high speed however energy economical. Theapproach is to around the operands to the closest exponent of 2. this fashion the machineintensive a part of the multiplication is omitted rising speed and energy consumption at theprice of atiny low error. The projected approach is applicable to each signed and unsignedmultiplications. we have a tendency to propose 3 hardware implementations of the approximate number thatincludes one for the unsigned and 2 for the signed operations. The potency of the projectedmultiplier is evaluated by comparison its performance with those of some approximate andaccurate multipliers victimisation completely different style parameters. additionally, the effectiveness of the projectedapproximate number is studied in 2 image process applications, i.e., image sharpeningand smoothing. The projected design of this paper analysis the logic size, space and powerconsumption victimisation Xilinx 14.2.

Introduction

Some of the previous works in the field ofapproximate multipliers are briefly reviewed. In an approximate multiplier and an approximate adder based on a technique named broken-arraymultiplier (BAM) were proposed.By applying the BAM approximation method to the conventional modified Booth multiplier, an approximatesigned Booth multiplier was presented. The approximatemultiplier provided power consumption savings form 28% to58.6% and area reductions from19.7% to 41.8% for differentword lengths in comparison with a regular Booth multiplier. Kulkarnietal.Suggested an approximate multiplier consisting of a number of 2×2 inaccurate building blocks thatsaved the power by 31.8%–45.4% over an accurate multiplier. An approximate signed 32-bit multiplier for speculationpurposes in pipelined processors was designed. It was20% faster than a full-adder-based tree multiplier while having a probability of error of around 14%.An error-tolerantmultiplier, which computed the approximate result bydividing the multiplication into one accurate and one approximate part, was introduced, in which the accuracies for different bit widths were reported. In the case of a 12-bit multiplier, a power saving of more than 50% was reported. In two approximate 4:2compressors for utilizing in a regular Dadda multiplier were designed and analyzed. The use of approximate multipliers in image processing applications, which leads to reductions in power consumption, delay, and transistor count compared with those of an exact multiplier design, has been discussed in the literature. In ,an accuracy-configurable multiplier architecture (ACMA) was suggested for error-resilient systems. To increase its throughput, the ACMA made use of a technique called carry-in prediction that worked based on a pre-computation logic. When compared with the exact one, the proposed approximate multiplication resulted in nearly 50% reduction in the latency by reducing the critical path. Also, Bhardwaj et al. presented anapproximate Wallace tree multiplier (AWTM).Again; it invoked the carry-in prediction to reducethe criticalpath. In this work, AWTM was used in a real-time benchmarkimage applicationshowing about 40% and 30% reductions inthe power and area, respectively, without any imagequalityloss compared with the case of using an accurate Wallace treemultiplier (WTM) structure.

Disadvantages:

1. Complex design

Proposed System:

Multiplication Algorithm of RoBA Multiplier:

The main idea behind the proposed approximate multiplieris to make use of the ease of operationwhen the numbersare two to the powern(2n). To elaborate on the operationof the approximatemultiplier, first, let us denote the rounded numbers of the input of A and B by Ar and Br, respectively. The multiplication of A by B may be rewritten as

$$A\times B=(A_r-A)\times(B_r-B)+A_r\times B+B_r\times A-A_r\times B_r. \qquad (1)$$

The key observation is that the multiplications of Ar ×Br, Ar×B, and Br×A may be implemented just by the shift operation. The hardware implementation of (Ar −A) × (Br −B),however, is rathercomplex. The weight of this term inthe final result, which depends on differences of theexact numbers from their rounded ones, is typically small. Hence,we propose to omit this partfrom (1), helping simplify the multiplication operation. Hence, to perform the multiplication process, the following expression is used:

$$A\times B\sim=A_r\times B+B_r\times A-A_r\times B_r. \qquad (2)$$

Thus, one can perform the multiplication operation using three shift and two addition/subtraction operations. In this approach, the nearest values for A and Bin the form of 2n should be determined. When the value of A(or B) is equal to the 3 ×2p−2(where p is an arbitrary positive integer larger than one), it has two nearest values in the form of2nwith equal absolute differences that are 2pand 2p−1. While both values lead to the same effect on the accuracy of the proposed multiplier, selecting the larger one (except for the case of p=2) leads to a smaller hardware implementation for determining the nearest rounded value, and hence, it is considered in this paper. It originates from the fact that the numbers in the form of 3×2p−2are considered as do not care in both rounding up and down simplifying the process, and smaller logic expressions may be achieved if they are used in the rounding up.

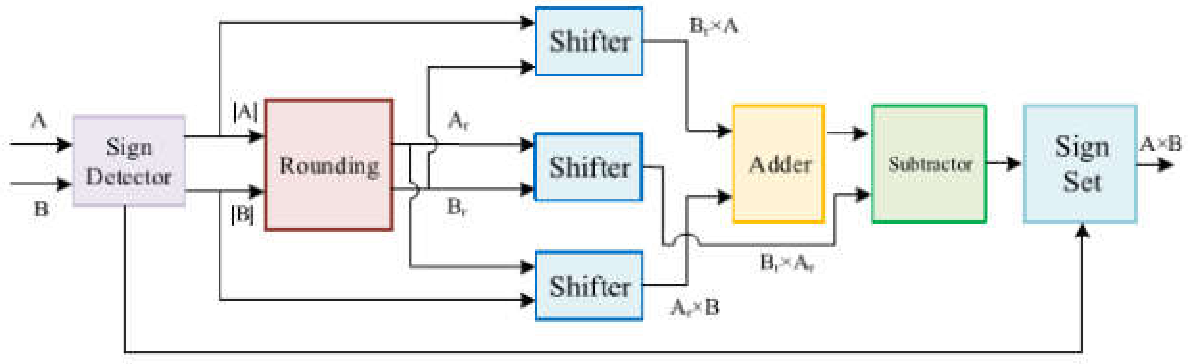


Fig.1:Block diagram for the hardware implementation of the proposed multiplier

Input 1 ($A_r\times B+B_r\times A$)	Input 2 ($A_r\times B_r$)	Output
000…11…xxx	000…10…000	000…01…xxx
000…11…xxx	000…01…000	000…10…xxx
000…10…xxx	000…01…000	000…01…xxx

$$\begin{aligned}
A_r[n-1] &= \overline{A[n-1]}\cdot A[n-2]\cdot A[n-3] \\
&\quad + A[n-1]\cdot\overline{A[n-2]} \\
A_r[n-2] &= (\overline{A[n-2]}\cdot A[n-3]\cdot A[n-4] \\
&\quad + A[n-2]\cdot\overline{A[n-3]})\cdot\overline{A[n-1]} \\
&\vdots \\
A_r[i] &= (\overline{A[i]}\cdot A[i-1]\cdot A[i-2]+A[i]\cdot\overline{A[i-1]})\cdot\prod_{i=i+1}^{n-1}\overline{A[i]} \\
&\vdots \\
A_r[3] &= (\overline{A[3]}\cdot A[2]\cdot A[1]+A[3]\cdot\overline{A[2]})\cdot\prod_{i=4}^{n-1}\overline{A[i]} \\
A_r[2] &= A[2]\cdot\overline{A[1]}\cdot\prod_{i=3}^{n-1}\overline{A[i]} \\
A_r[1] &= A[1]\cdot\prod_{i=2}^{n-1}\overline{A[i]} \\
A_r[0] &= A[0]\cdot\prod_{i=1}^{n-1}\overline{A[i]}. \qquad (3)
\end{aligned}$$

Hardware Implementation of RoBA Multiplier: Based on (2), we provide the block diagram for the hardware implementation of the proposed multiplier in Fig. 1 where the inputs are represented in two's complement format. First, the signs of the inputs are determined, and for

each negative value, the absolute value is generated. Next,the rounding block extracts the nearest value for each absolute value in the form of 2n. It shouldbe noted that the bit width of the output of this block is n(the most significant bit of the absolutevalue of ann-bit number in the two's complement format is zero).To find the nearest value ofinput A, we use the following equation to determine each output bit of the rounding block:
In the proposed equation, Ar [i] is one in two cases. In the first case, A[i] is one and all the bits on its left side are zerowhileA[i −1] is zero. In the second case, when A[i]andall its left-side bitsare zero, A[i −1] and A[i −2] are both one. Having determined the rounding values, using threebarrel shifter blocks, the products Ar×Br, Ar×B, and Br×A are calculated. Hence, the amount ofshifting is determined based on logAr2 −1(orlogBr2 −1)in the case of A(or B) operand.Here, theinput bit width of the shifter blocks is n, while their outputs are 2n.A single 2n-bit Kogge-Stone adder is used to calculate the summation of Ar ×Band Br ×A. Theoutput of this adder and the result of Ar ×Br are the inputs of the subtract or block whose outputis the absolute value of the output of the proposed multiplier. Because Ar and Br are in the formof 2n,the inputs of the subtract or may take one of the three input patterns shown in Table I. Thecorresponding output patterns are also shown in Table I.The forms of the inputs and output inspired us to conceivea simple circuit based on the followingexpression:

$$Out=(PXORZ)AND(\{(P<<1)XOR(PXORZ)\}or\ \{(PANDZ)<<1\}) \quad (4)$$

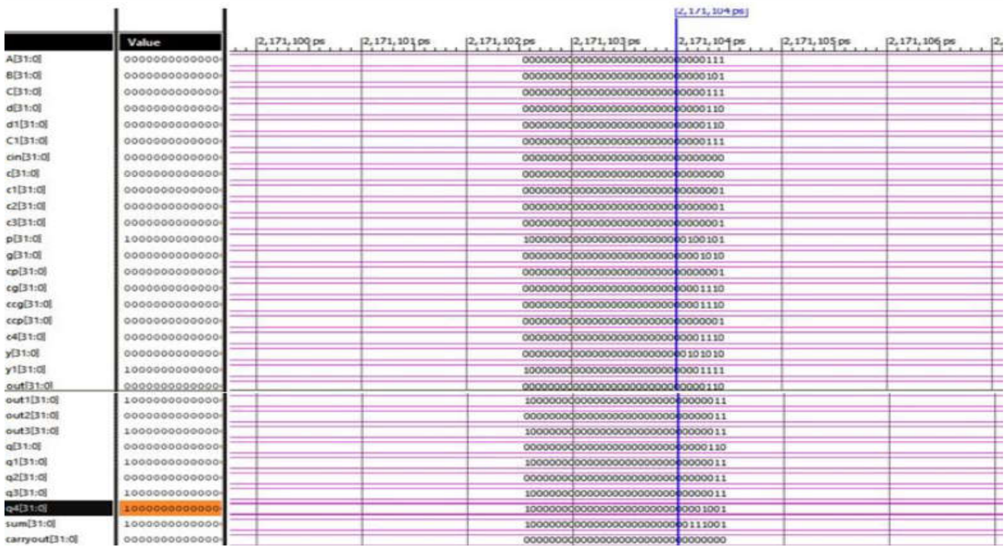
Where P is Ar ×B+Br ×A and Z is Ar ×Br. The corresponding circuit for implementing this expression is smaller and faster than the conventional subtraction circuit.

Advantages:

2. Design complexity is less

Simulations and results

The MROBA multiplier was designed using thehardware description language verilog Xilinx ISE 14.7and other parameters like area, power and delay arebeen calculated in Cadence Encounter at 180 nano.



```
A[31:0]         0000000000000
B[31:0]         0000000000000
C[31:0]         0000000000000
d[31:0]         0000000000000
d1[31:0]        0000000000000
C1[31:0]        0000000000000
cin[31:0]       0000000000000
c[31:0]         0000000000000
c1[31:0]        0000000000000
c2[31:0]        0000000000000
c3[31:0]        0000000000000
p[31:0]         1000000000000
g[31:0]         0000000000000
cp[31:0]        0000000000000
cg[31:0]        0000000000000
ccg[31:0]       0000000000000
ccp[31:0]       0000000000000
c4[31:0]        0000000000000
y[31:0]         0000000000000
y1[31:0]        1000000000000
out[31:0]       0000000000000
out1[31:0]      1000000000000
out2[31:0]      0000000000000
out3[31:0]      1000000000000
q[31:0]         0000000000000
q1[31:0]        1000000000000
q2[31:0]        0000000000000
q3[31:0]        1000000000000
q4[31:0]        1000000000000
sum[31:0]       1000000000000
carryout[31:0]  0000000000000
```

Conclusion

In this paper we propose a Modified rounding basedapproximate multiplier (MROBA). By modifying theconventional multiplier for the accurate results.Compared with the conventional multiplier themodified multiplier gives exact result to the giveninputs and the multiplier can also perform operationswhich are not in the form of 2n (as performed in theconventional method) so, the exact result can beobtained for the numbers irrespective of 2n . TheMROBA is designed using Xilinx ISE 14.7 and resultsare shown above and other parameters like area powerdelay are been calculated using cadence encounter andthe design of MAC unit is also implemented in XilinxISE 14.7

References

[1] Reza Zendegani, Mehdi Kamal, MiladBahadori,Ali Afzali- Kusha and MassoudPedram,” RoBAMultiplier: A Rounding-Based ApproximateMultiplier for High-Speed yet Energy Energy-Efficient Digital Signal Processing” IEEEtransactions on very large scale integration (VLSI)systems syst., pp.1-9, July 2017.

[2]. K. Bhardwaj and P. S. Mane, “ACMA:Accuracy-configurable multiplier architecture forerror-resilient system-on-chip,” in Proc. 8th Int. Workshop Reconfigurable Commun.-Centric Syst.-Chip, 2013, pp.1–6.

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

[4] D. R. Kelly, B. J. Phillips, and S. Al-Sarawi,“Approximate signed binary integer multipliers forarithmetic data value speculation,” in Proc. Conf.Design Archit. Signal Image Process., 2009, pp. 97–10.

[5] A.B. Kahng and S.Kang,”Accuracy-configurableadder for approximate arithmetic designs,” in Proc49th Design AutomConf (DAC), Jun 2012, pp.820-825.

[6] S. Deepak, B J kaliath,”Optimized MAC unitDesign”.Electron devices and solid state circuit IEEEinternational conference on 2012.