Modified Rounding Based Approximate Multiplier (MROBA) and MAC Unit Design for Digital Signal Processing

Pentum Suhasini
MTech VLSI design, Department of ECE
SRM Institute of Science and Technology
Chennai, India
suhasinisuhani2405@gmail.com

Dr. J. Selvakumar
Assistant Professor (Sr. Grade) Department of ECE
SRM Institute of Science and Technology, Chennai
selva2802@gmail.com

Abstract - In this paper we propose a Modified rounding based approximate multiplier (MROBA) which is more accurate than the conventional multiplier (ROBA). The main concept of multiplier depends on rounding of numbers. This multiplier can be applied for both signed and unsigned numbers. Three hardware implementations are proposed in which one implementation for unsigned and two for signed operations. The accuracy of the multiplier is compared with the conventional rounding based approximate multiplier (which are in \(2^n\)) where the modified rounding based approximate multiplier gives an exact output for the given inputs (irrespective of \(2^n\)) and various parameters like area, power delay, error significance, pass rates are been calculated and compared with conventional multiplier where, MROBA gives better results and with the MROBA MAC unit is implemented.

Keywords - Approximate multiplier; accuracy; MAC unit.

I. INTRODUCTION
The crucial part of the arithmetic units are basically built by the multiplier hardware, so multipliers play a prominent role in any design. [1] If we consider a Digital signal processing (DSP) the internal blocks of arithmetic logic designs, where multiplier plays a major role among other operations in the DSP systems [1]. So, in the design of multiplier and accumulate unit (MAC) multipliers play an important role. Next, important design in the MAC unit is the Adder. Adders also share the equal important in this design. By the appropriate function methods different kinds of adders and multipliers designs are been suggested. By the approximate computing the designer can make tradeoffs, accuracy, speed, energy and power consumption. In this paper we proposed the modified form of rounding based approximate multiplier which is low power design, high speed and energy efficient.

II. LITERATURE SURVEY
The multiplier designed was built using the conventional multiplier approach at the algorithm level by considering the rounded input values which are not in the form of \(2^n\) so, we call this multiplier the modified rounding based approximate multiplier. This multiplier can be applied for Signed and Unsigned operations by which three different architectures are implemented. The collection of this paper can be further classified has:

a) Describing the conventional rounding based approximate multiplier (ROBA) and its inaccuracy.

b) Proposed modified rounding based approximate multiplier (MROBA) and its hardware implementation.

c) With the MROBA MAC unit is designed.

The paper further is given has follows: Section II describes the work of other approximate multiplier designs. Section III describes the conventional rounding based approximate multiplier and its inaccuracy. Section IV describes proposed modified rounding based approximate multiplier. Section V describes the hardware implementation. Section VI describes the comparison of MROBA with ROBA multiplier. Section VII describes specifications of MROBA. Section VIII describes the MAC unit design using MROBA. Section IX describes simulations and results. Section X describes the conclusion and Section XI is references.
proposed by two parts one for the appropriate result and the other was for the accurate results. A 32-bit signed appropriate multiplier was designed which were used in many pipelined processors [5] DRUM6 and DSM8 are some of the approximate multiplier compared with the ROBA multiplier which yields in poor results in almost in all the cases. So, the design of MROBA helps in securing better results and with this MROBA MAC unit is been designed.

III. Conventional rounding based multiplier and its inaccuracy (ROBA)

The main concept of conventional rounding based approximate multiplier [1] is selecting the rounded values for both the inputs which are in form of $2^n$ and both the inputs should be in the form of $3x2^{p-1}$ ($p$ is considered as arbitrary positive integer value which is greater than 1) in this case of the conventional approach the final value obtained by the multiplier would be less or more than the exact result obtained. Depending on the Ar (rounded input value of A) and Br (rounded input value of B) respectively and the result obtained is inaccurate. Considering the table 1

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Rounded value Ar</th>
<th>Rounded value Br</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>A=B</td>
<td>Larger value</td>
<td>larger value</td>
<td>Smaller value than the exact value</td>
</tr>
<tr>
<td>A&gt;B</td>
<td>Large value</td>
<td>Larger value</td>
<td>Smaller value than the exact value</td>
</tr>
<tr>
<td>A&lt;B</td>
<td>Larger value</td>
<td>Larger value</td>
<td>Larger value than the exact value</td>
</tr>
</tbody>
</table>

So in this paper we design a modified rounding based multiplier which produces the exact output for the given inputs based on the selecting the appropriate rounding values Ar and Br respectively.

IV. Proposed Modified Rounding based approximate multiplier (MROBA)

A. Algorithm for MROBA multiplier

The main concept of the proposed modified rounding based approximate multiplier is to design the multiplier so that it takes all values which are irrespective of $2^n$. The detailed explanation of the multiplier is described further. Initially let us consider Ar as the rounded input value of input A and Br as the rounded input value of input B. Now the multiplication of the A* B is written has follows:

$$A * B = (Ar * B) + (Br * A) - (Ar * Br) - [1]$$

The basic key point to be considered is the product of $(Ar * B)$ and $(Br * A)$ is complex and the weight of the term would result in small values when compared with the exact numbers so the product of this term can omitted and it also leads to complex hardware design approach. Hence the multiplication can be performed by the following expression as

$$A * B = (Ar * B) + (Br * A) - (Ar * Br) - [1]$$

The product terms of $(Ar * B)$, $(Br * A)$, $(Ar * Br)$ can be implemented by three barrel shifters of N bit and one adder is implemented by the parallel prefix kogge-stone adder of N bit and one Subtractor is also needed. If the values of A or B is equal to N where $(N=1, 2, 3, …..N)$ it has two rounded values $N+1$ and N. depending on the inputs of A and B the rounded values can be chosen this kind of rounding values are applicable for both A greater B (A>B) and A less than B condition (A<B). Exception that for A equal B (A=B) the rounded input of A and B would be the middle number of $N+1$ and N i.e. $(2N+1)/2$ In the conventional rounding based approximate multiplier the numbers (which represents inputs of ROBA) in the form of $3x2^{p-2}$ (where p is arbitrary positive number which is greater than one) are considered and also we have two rounded values in the form of $2^n$ and $2^{p-1}$ both the values lead to the same effect but in this case the larger value is considered as the rounded value for both the inputs because larger values leads in smaller hardware implementation. But the accuracy of this multiplier is poor and the exact result is not obtained in this case so, we consider the modified rounding based approximate multiplier which is applicable to all N $(N=1, 2, 3, …..N)$ numbers and the exact output is obtained for the given input. Fig 1.

V. Hardware implementation of the proposed modified rounding based approximate multiplier

Based on fig 2 we have the block diagram for the modified rounding based approximate multiplier. Initially we have a sign detector block where the inputs A and B are given. As in the case of the unsigned numbers the sign detector block can be applied to positive numbers. But if the signed numbers are considered the absolute values are to be generated and then inputs are to be given. The input of A can have its rounded value has the larger value $(N+1)$ or the smaller value $(N)$. Similarly for input B the rounded value can either be the larger value or the smaller value.
This condition is applicable for both A greater than B (A > B) and for A less than B (A < B) Condition. But for the A equal to B (A=B) the rounded values are considered according to (2N+1)/2. Consider table 2.

NOTE: The rounded values of both the inputs should be compulsory differ from other rounded value eg: for input A its rounded value Ar should be larger value and for the input B the rounded value Br should be smaller value or vice versa for both inputs. Expect for (A=B) the rounded value should be the middle value or (2N+1)/2 which is same for both the inputs.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Rounded value Ar</th>
<th>Rounded value Br</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>A=B</td>
<td>Middle value</td>
<td>Middle value</td>
<td>Exact</td>
</tr>
<tr>
<td>A&gt;B</td>
<td>Larger value</td>
<td>Larger or smaller value</td>
<td>Exact value</td>
</tr>
<tr>
<td>A&lt;B</td>
<td>Larger or smaller value</td>
<td>Larger or smaller value</td>
<td>Exact value</td>
</tr>
</tbody>
</table>

Next block we have a basic adder we can add +1 i.e. (N+1). Now we need three shifters to compute the product terms (Ar * Br), (Br * A), (Ar * Br). The amount of shifting is basically done on no of bits we considered and the shifting operation is performed on the rounded values of Ar and Br respectively Then the output of shifter 1 and shifter 3 are given has the input to the adder basically we considered an kogge-stone adder. And the output of the adder and the output of the shifter2 is given has the input to the Subtractor block and finally we have a sign set block which is needed if we considered the signed numbers. For the unsigned it is not necessary and the final multiplication A*B is obtained the output is exact for the given input when compared with the conventional rounding based approximate multiplier it gives the exact result for the input in all the possible conditions.

VI. Comparison of MROBA multiplier with ROBA multiplier

Compared with the ROBA multiplier the MROBA gives the exact output for the given inputs in all the possible cases.

<table>
<thead>
<tr>
<th>conditions</th>
<th>ROBA multiplier</th>
<th>MROBA multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>A=B</td>
<td>Small value</td>
<td>Exact result</td>
</tr>
<tr>
<td>A&gt;B</td>
<td>Smaller value</td>
<td>Exact result</td>
</tr>
<tr>
<td>A&lt;B</td>
<td>Larger value</td>
<td>Exact result</td>
</tr>
</tbody>
</table>

Examples1:

ROBA multiplier: (unsigned)
When: A=12 (3 * 2^2 ), B=6 (3 * 2^1 )
Ar=2^2 =16    Br=2^1 =8
A * B= (Ar* B) + (Br *A) - (Ar * Br)
Here Ar=16, Br=8
(16*6) + (8*12) – (16* 8)
72 ≠ 64

In the conventional approach the inputs should be in the 3 x 2p-1 form and we have two rounded values in the form of 2p and 2p-1 both the values lead to the same effect but here the larger value is considered as the rounded value for both the inputs because larger values leads in smaller hardware implementation.

Examples 2:

ROBA multiplier: (signed)
When: A=-12 (3* 2^2 ), B=-12 (3*2^2 )
Ar=2^2 =16    Br=2^2 =16
A * B= (Ar* B) + (Br *A) - (Ar * Br)
Here Ar=16, Br=16
(16*-12) + (16*-12) – (16*16)
144 ≠ 256
Fig 2: Block diagram of the modified rounding based approximate multiplier

Here -256 is the value since we have a sign set block at the end only positive value is obtained.

Examples 3:
MROBA multiplier: (unsigned)
When: \( A=7, \quad B=5 \)
\[ A \times B = (A \times B) + (B \times A) - (A \times B) \]
Here \( A_r=7, \quad B_r=6 \)
\[ (7 \times 5) + (6 \times 7) - (7 \times 6) = 35 \]

Examples 4:
MROBA multiplier: (signed)
Considering: \( A=-35, \quad B=47 \)
\[ A_r=36, \quad B_r=47 \]
\[ A \times B = (A \times B) + (B \times A) - (A \times B) \]
Here \( A_r=36, \quad B_r=47 \)
\[ (36 \times 47) + (47 \times -35) - (36 \times 47) = 1645 \]

Here -1645 is the value since we have a sign set block at the end only the positive value is obtained results compared with ROBA. So the main concept here is accuracy.

So, from the above examples of MROBA multiplier and ROBA multiplier we can say that the MROBA is more accurate than ROBA and the main advantage is that we can perform multiplication for all the numbers irrespective of \( 2^n \) since in the ROBA multiplier only the inputs and rounded values are strictly \( 2^n \) but in this MROBA we can perform multiplication for all \( N \) inputs (\( N=1, 2, 3, \ldots \)).

VII. Specifications of MROBA

In this section we calculate some of the important parameters of the MROBA multiplier and compared them with the ROBA multiplier. Most of the parameters are considered by considering \( (A=B) \) condition

1. Maximum error :
\[
\text{Max}(\text{error}(A,B)) = \frac{((2N+1/2)-N) \times (2N+1/2)-N}{N \times N}
\]
\[= 1/4N^2 \]

2. Approximate signed error:
\[
\text{Error}(A, B) = \frac{(A_r-A) \times (Br-B)}{A \times B} + \frac{A+B+1}{N \times N}
\]
\[= \frac{(N+1-N) \times (N-N)}{N \times N} + \frac{A \times B}{N \times N}
\]
\[= \frac{2N+1}{N \times N} = 1.6\%
\]

So, from the above table 4 when we compare the specifications of the MROBA multiplier with ROBA Multiplier so, Modified rounding based multiplier is more accurate in all the cases when compared with the ROBA multipliers specifications.

In the fig 5 we observe that the area and PDA of MROBA is more when compared with ROBA. Hence the main concept here is to obtain accurate results which are obtained by MROBA and in almost all cases MROBA achieve better results.
### Table 4: Specifications of MROBA multiplier compared with ROBA multiplier

<table>
<thead>
<tr>
<th>Multiplier</th>
<th>Maximum Error</th>
<th>Error rate</th>
<th>Error significance</th>
<th>Acc inf</th>
<th>Relative error</th>
<th>Pass rates [32-bit]</th>
<th>A.cc amp</th>
</tr>
</thead>
<tbody>
<tr>
<td>MROBA UNSIGNED</td>
<td>&lt; 0.25%</td>
<td>-0.2%</td>
<td>-0.2%</td>
<td>45.3%</td>
<td>~0.2%</td>
<td>1.05%</td>
<td>0.997%</td>
</tr>
<tr>
<td>MROBA SIGNED</td>
<td>1.5%</td>
<td>-0.2%</td>
<td>-0.2%</td>
<td>42.5%</td>
<td>~0.5%</td>
<td>2.3%</td>
<td>0.5%</td>
</tr>
<tr>
<td>MROBA AS-SIGNED</td>
<td>1.6%</td>
<td>-0.2%</td>
<td>-0.2%</td>
<td>42.3%</td>
<td>0.09%</td>
<td>5.3%</td>
<td>3.0%</td>
</tr>
<tr>
<td>ROBA UNSIGNED [1]</td>
<td>0.1%</td>
<td>~1%</td>
<td>~1%</td>
<td>60.0%</td>
<td>18.1%</td>
<td>1.5e-6</td>
<td>1.77%</td>
</tr>
<tr>
<td>ROBA SIGNED [1]</td>
<td>2.66%</td>
<td>~1%</td>
<td>~1%</td>
<td>58.0%</td>
<td>18.1%</td>
<td>2.98e-6</td>
<td>2.58%</td>
</tr>
<tr>
<td>ROBA AS-SIGNED [1]</td>
<td>2.35%</td>
<td>~1%</td>
<td>~1%</td>
<td>58.0%</td>
<td>17.9%</td>
<td>&gt;7.45e-7</td>
<td>2.58%</td>
</tr>
</tbody>
</table>

**Fig 3:** MROBA Power (m W), delay (ns), Energy delay product (EDP in ns X p J) and Energy (p J) of 32-bit MROBA multiplier.

**Fig 4:** ROBA Power (m W), delay (ns), Energy delay product (EDP in ns X p J) and Energy (p J) of 32-bit ROBA multiplier [1].

**Fig 5:** AREA ($\mu M^2$) and PDA (p J X $\mu M^2$) of MROBA and ROBA multiplier[1].

Basically MAC (Multiply and Accumulate) unit is formed by Adders and Multipliers and the result is accumulated.[7] We find many advantages of MAC in Digital signal processing and many logical units design and implementation. The basic MAC unit using MROBA is shown in fig 6. Here we are designing a MAC unit of 32*32 bit using MROBA multiplier. The detailed explanation of multiplier can be seen in the above description. The basic MAC has three sub systems namely Multiplier, Adder and Accumulator. The main purpose of multiplier is to find the partial products. [7]Adder helps in adding the partial products generated by the multiplier and finally we have the accumulator to store those values. Here in this implementation kogge-stone adder is been used because when compared to other adder like carry save adder, carry select adder etc kogge stone adder has less delay, minimum fan-out and it performs fast logical addition when compared with the other designs.
IX Simulations and results

The MROBA multiplier was designed using the hardware description language verilog Xilinx ISE 14.7 and other parameters like area, power and delay are been calculated in Cadence Encounter at 180 nano

X Conclusion

In this paper we propose a Modified rounding based approximate multiplier (MROBA). By modifying the conventional multiplier for the accurate results. Compared with the conventional multiplier the modified multiplier gives exact result to the given inputs and the multiplier can also perform operations which are not in the form of $2^n$ (as performed in the conventional method) so, the exact result can be obtained for the numbers irrespective of $2^n$. The MROBA is designed using Xilinx ISE 14.7 and results are shown above and other parameters like area power delay are been calculated using cadence encounter and the design of MAC unit is also implemented in Xilinx ISE 14.7

XI References


