Multiplexer Based Error Efficient Fixed-Width Adder Tree Design for Signal Processing Applications

Rashmi Seethur\textsuperscript{1}, Deeksha Sudarshan\textsuperscript{2}, Vaibhavi Naik\textsuperscript{3}, Tushar Masur\textsuperscript{4}, and Shreedhar H K\textsuperscript{5}

\textsuperscript{1,2,3,5}Rashmi Seethur, Deeksha Sudarshan, Vaibhavi Naik and Tushar Masur, Electronics and Communication Engineering Department, PES University, Bangalore, India
\textsuperscript{5}Shreedhar H K, Electronics and Communication Engineering Department, GAT, Bangalore, India
Email: rashmiseethur@pes.edu

Abstract—In copious mixed media applications, humans cannot necessarily discern error free or erroneous outputs, owing to the small range of perception abilities. Crucial information can still be obtained from marginally inexact outputs. Leveraging this, many algorithms such as Digital Signal Processing (DSP), Discrete Cosine Transform (DCT), Motion Compensation (MC) use approximate calculations while still maintaining appreciable computation accuracy. When data processing algorithms are taken into consideration, adders play an important role in the arithmetic module by managing the power and area utilization of the system. A fixed-width adder tree design for approximate calculations is proposed, that uses the trade-off between area and accuracy as the base for analysis. Our design uses 12.42%, 18.17% and 5.05% lesser area compared to the full width adder tree, FX-AT-PT and FX-AT-DT respectively. Additionally, when compared to FX-AT-DT, TFX-AT and ITFX-AT, the proposed design has an improved Maximum Error Distance (MED) of (33.33%, 29.03%), (63.64%, 65.63%) and (38.46%, 35.29%) for N = 8, 16 respectively. To cope with the inaccuracy caused by truncation, the proposed design employs mux-based radix-4 addition coupled with bias estimation. Further, to examine the error performance we have incorporated the proposed design and a few other existing designs into the Walsh-Hadamard Transform (WHT), to process images with different metrics and compare the Peak Signal to Noise Ratio (PSNR) of the images. It was observed that the proposed design showed significant improvement in the PSNR score when compared to ITFX-AT and maintains a score similar to that of FX-AT-PT.

Index Terms - Full width, Fixed width, Adder Tree (AT), approximate AT

1. INTRODUCTION

Artificial intelligence, big data analytics, data mining, machine learning, multimedia processing, cloud computing, Internet of Things, etc. are examples of exhaustive computation technologies that routinely deal with a data flood, making correct computing approaches expensive in terms of time and resources. In these situations, computation that produces answers that are only roughly, inaccurately, or imprecisely right might be more practical and cost-effective. For instance, in image processing applications minor deviation in the quality of the image is not necessarily captured by the human eye [9]. This provides us with some latitude to do erroneous or approximative calculations. This flexibility allows us to develop low-power systems at many design abstraction levels, including logic, architecture, and algorithm [5]. The approximation computing paradigm is exclusive to a few hardware implementations of DSP blocks. Hardware, software, and memory storage all fall under the umbrella of approximate computing. Research on approximate hardware implementations are mostly concentrated on arithmetic and logic circuits. Since addition and multiplication are often carried out in microprocessors and digital signal processors, adders and multipliers have drawn a lot of interest in the field of approximate arithmetic circuits. This has led to an increased usage of low-power and area-efficient methodologies. To support this, signal processing systems are realized on fixed-point VLSI applications. This paper talks about approximation adders, which are created by inserting errors into an exact adder. There are essentially two different types of approximate adders: static approximate adders (SAs) and dynamic approximate adders (DAs) as discussed in [2]. When compared to an accurate adder, a SA's fixed approximation feature ensures certain savings in design metrics and may output either a correct sum or an approximation with the desired precision in a single clock cycle. While a DA may compute an exact or approximate total upon request using single or more clock cycles, approximation is changeable in a DA. DAs also contain an additional Error Detection and Correction logic (EDCL) which in some cases proves to be an overhead and to bypass this we can simply make use of the SA. Typically, a SA is divided into two sections: a precise section where addition is performed accurately, and an imprecise section where addition is performed incorrectly or inexactly. The exact component receives more significant adder input bits, whereas the imprecise section receives less important adder input bits. The accurate portion therefore has more importance than the imprecise portion. Using approximate adders, desired adder-tree structures can be designed for various applications. Adder-trees are routinely used in matrix-vector multiplication and parallel designs of inner product computation. Implementing approximate adder-tree designs in computation intensive applications prove to be profitable as a significant decrease in metrics such as area, power and error performance can be observed when compared to conventional or full-width adder-tree structures. Fixed-width adder-tree (AT) design is often derived from full-width adder-tree design by implementing direct or post truncation. The full-width AT (FL-AT) generates a \((w + p)\) bit output for \(N\) input vector, where \(N\) is the total number of inputs, \(w\) is the number of bits for each input, and \(p = \log_2N\), which determines the number of stages in the adder.
tree. In direct truncation, the least significant bit of the output from each adder is truncated after computation at each stage. Whereas, in post truncation, the least significant bit of the final output from the entire adder tree is truncated. However, these approximations may not hold good for a wide range of applications. In the recent past, several approximate designs have been proposed to achieve efficient arithmetic computation [6]. As we play around with truncation at different levels, the error induced becomes significantly high and computation accuracy may vary. [1] Investigated the Fixed Width AT using bias estimation aspects. Using the bias estimation, the maximum error distance of adder tree is improved to 13 with an area optimized by 40% compared to full width adder tree. [2] Provides a comparative analysis of different gate-level SAAs, suitable for both FPGA and ASIC type implementations, was performed from the perspective of error and design metrics. Different state of art approximate adders is compared, and it is observed that [7] is better in terms of area and [8] is preferred in terms of accuracy of result. [3] This paper discusses transistor level adder optimizations. To perform switch level arithmetic design mirror design concepts are explored. [1] uses type 1 2 3 and 4 of [3] to compare his proposed design with existing state of art. Previous works [1] have reported the Maximum Error Distance of 13 and with area improvement of 29% compared to FX-AT-PT. In our work a multiplexer-based design is proposed, to achieve a maximum error distance of 8 for N = 8, 16 irrespective of the word length (w). The rest of the paper is organized as follows, Section II deals with existing adder tree designs and their comparisons based on different metrics. Section III discusses about the proposed Adder Tree design. Section IV deals with hardware and time complexity performance study for image processing applications. Section V deals with conclusion and future scope.

II. EXISTING ADDER TREE DESIGNS

In order to obtain a distinct outcome when compared to the proposed design, we have chosen five divergent existing adder tree designs and performed comparative analysis against key metrics such as area, power, delay and error performance. For the sake of conformity, all architectures were designed using Cadence NC-simulator tools using the saed90nm typical library with built in constraint.

A. Full Width Adder Tree

The full width adder tree is a conventional tree which provides an accurate summation of N inputs, and each input of size w bits. Multiple full width adders are used in all the stages and the number of stages depends on N. For instance, we have provided 8 inputs, each of size 8 bits. As full width adders are used at each stage the output bit size increments by 1. As depicted by Fig.1(a), for 8 inputs, the tree will have 3 stages, which is determined by log2N. Therefore, as shown by Fig.2 the final output is the total of N and the number of stages, i.e., 8 + 3 = 11 bits, which is same as w+log2N. The accuracy of this full width adder tree is 100%, since all bits are added, along with their carry bits generated in all the stages.

B. FX-AT-PT (Fixed-Width Post-Truncated Adder-Tree)

In the post truncation method, the adder tree will provide precise addition of the inputs, similar to the full width adder tree. Although in the penultimate stage, the data will be divided into two parts, namely MSP (Most Significant Part) and LSP (Least Significant Part) as shown in Fig.1(b). The size of LSP is decided with the help of log2N. The MSP is given to an exact adder to get the accurate output, whereas the LSP is processed through a “C” block to provide Cm to the last adder block, in order to reduce the error distance. The tree will have ceil of log2N stages. The output of the first stage is w+1 bits, the output of the second stage is w+2 bits so on as observed by Fig.2. The function of the “C” block is as follows:

\[ a = \{a_2, a_1, a_0\} \]

\[ b = \{b_2, b_1, b_0\} \]

\[ c_1 = a_0 \cdot b_0 \]  

\[ c_2 = a_1 \cdot c_1 + b_1 \cdot c_1 + a_1 \cdot b_1 \]

\[ r = a_2 \cdot c_2 + b_2 \cdot c_2 + a_2 \cdot b_2 \]

This method gives a maximum error distance of 7 which is considered as the minimum error encountered in adder tree design as per state-of-the-art. For the sake of comparison, the area of this AT is considered as upper ceiling and its maximum error distance is considered as minimum ceiling for upcoming AT design comparison.

C. FX-AT-DT (Fixed-Width 1-bit Direct-Truncated Adder-Tree)

In the Direct truncation method, the adder tree will perform precise addition of the inputs, but the Least Significant Bit (LSB) of the output at each stage will be truncated/ignored/neglected, without affecting the remaining bits. This results in the size of final output to be same as w, as depicted by Fig.3. Let us consider N inputs each of w bits in size, and let p be the number of stages, required to compute final output of AT. The output from the first stage will be w+1 bits, but as the LSB is truncated, input vector...
size to the subsequent stage will remain as w bits. This computation method is followed until the final stage. At the final stage, we append p zeros to LSB to maintain the size of final answer, which is w+p bits. FX-AT-DT method achieves 20% in area efficiency at the cost of 23% increased MED.

D. TFX-AT (Truncated Fixed-Width Adder-Tree)

In TFX-AT [1], the data is divided into two parts, MSP (Most Significant Part) and LSP (Least Significant Part). The LSP size depends on p = \log_2 N, where N is the number of inputs. In TFX-AT, the LSP is truncated at the input stage and there is a constant bias, carry in “Cin = 1”, provided to the first stage of the tree to compensate for the significant error incurred due to truncation. This can be understood by Fig.5. For N input data, w-log_2 N bit MSP is added precisely. The final number of stages required to compute for N inputs will be ceil of log_2 N. The final output of the tree will again be w bits. Input bit matrix is shown in Fig.4.

For the 8 inputs of 8 bits each, the maximum error distance of the truncated tree adder is 56 which can be reduced to 28 by adding bias input of (4=8/2) in the first stage of the adder tree. The mathematical approach used to compute fixed bias in adder tree is given by Equation 6.

\[ Y = MSP + LSP \]  

Where Y = Final output of adder tree  
MSP = Most Significant Part used for precise addition  
LSP = Least Significant part which is truncated

To compensate the error due to truncation the LSP part is replaced by fixed bias estimation. Each number in LSP matrix is binary occurring with the probability of ½. The output y is expressed as

\[ Y = MSP + 2^p \sigma \]  

Where 2^p \sigma is LSP term and \sigma is estimated bias. If all p=log_2 N are considered for truncation, then w×p LSP matrix is estimated as

\[ E[LSP] = N \sum_{i=1}^{p} (\frac{1}{2})^2^p \]  

w = number of bits in input, p = LSP bits considered for truncation as discussed in [4].

TFX-AT shows a 42.37% improvement in area when compared to FX-AT-PT. But compromises in terms of MED by 68.18% when compared to FX-AT-PT.

E. ITFX-AT (Improved Truncated Fixed-Width Adder-Tree)

In ITFX-AT [1], the data is divided into two parts, MSP (Most Significant Part) and LSP (Least Significant Part). The LSP size depends on the formula log_2 N, where N is the number of inputs. Depending on the number of LSP bits considered for truncation, accuracy of adder tree will be configured. Here in ITFX-AT among three bits of LSP, higher significant 1-bit (\sigma major) is computed precisely whereas lower two bits of LSP (\sigma minor) bits are truncated. Further to improve accuracy, fixed bias estimation is added in to the first stage of adder tree. MSP is added accurately like that of TFX-AT. Higher significant bit of LSP is also added precisely using A* and A blocks in the tree stages. In this design the accuracy of the tree improved by
Rashmi et al.: Error Efficient MT-FX-AT for Signal Processing Applications

46.15% compared to TFX-AT at the cost of 24.7% area overhead. The design of TFX-AT fixed width adder tree with N=8 and w=8 is depicted in Fig. 6. The details of A and A* blocks are shown in Fig. 7. Here A* stage is used in first stage of the tree design to incorporate 1-bit accurate addition with bias added for error correction. A block is added in subsequent stages to perform 1-bit exact addition to improve error performance parameter of the AT. Implementation of the fixed bias can be understood by Table 1. Figure 8 shows the input bit matrix.

Fig.7 A* block for 1-bit accurate arithmetic with bias estimation and A block for the subsequent stages

Table 1: Fixed bias implementation for A* and A block

<table>
<thead>
<tr>
<th>Fixed Bias</th>
<th>A</th>
<th>B</th>
<th>S</th>
<th>CO</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

For ITFX-AT to estimate the bias of last two bits (truncated part) of LSP, the estimation equation is given by

\[ E[LSP] = N \sum_{i=1}^{W} \frac{1}{2} 2^{p-2-i} \] \hspace{5cm} (12)

\[ W = \text{number of bits of each input} \]

\[ p-1 = \text{LSP bits considered for truncation} \]

expressed as

\[ E[LSP] = \frac{N}{4} 2^p (1 - 2^{-p+1}) \approx \frac{N}{8} 2^p \] \hspace{5cm} (13)

III. PROPOSED MULTIPLEXER BASED FIXED WIDTH ADDER TREE DESIGN (MT-FX-AT)

In MT-FX-AT design, the data is divided into two parts, MSP (Most Significant Part) and LSP (Least Significant Part). The MSP is computed using the traditional full-width adders without truncation. The LSP is computed by using AXA* and AXA block in the first and second stage of the tree as depicted in Fig. 9, where AXA stands for approximate addition. The number of bits analyzed by LSP block in the tree is given by \( \log_2 N \), where N is the number of inputs. In proposed design the LSP is further divided into \( \sigma \) major and \( \sigma \) minor as shown in Fig. 10. \( \sigma \) major is the 2 higher order bits of the LSP and remaining least significant bit of LSP comprise the \( \sigma \) minor which needs to be truncated in the proposed design. The \( \sigma \) major bits are used to generate the fixed bias in the LSP stage of the tree by computing radix-4 addition. For N inputs each of w bits, the MSP comprise of w-\( \log_2 N \) bits and LSP comprise of \( \log_2 N \) bits. The \( \sigma \) major is passed through “AXA*” block in first stage and “AXA” block in the subsequent stage. AXA* block performs precise radix-4 addition with 1-bit fixed bias. Whereas AXA block does only accurate radix 4 addition with minimum area requirements. AXA* block comprises of two 2:1 multiplexer, XOR, OR and XNOR gates and AXA block comprises of two 2:1 multiplexer, two XOR and AND gates as depicted in Fig. 11. The least significant bit of the input is truncated. The design analysis of AXA and AXA* block is detailed in Table 2.

For MT-FX-AT to estimate the bias of Least significant bit (truncated part) of LSP, the estimation equation is given by

\[ E[LSP] = N \sum_{i=1}^{W} \frac{1}{2} 2^{p-2-i} \] \hspace{5cm} (10)

\[ W = \text{number of bits of each input} \]

\[ p-1 = \text{LSP bits considered for truncation} \]

expressed as

\[ E[LSP] = N \frac{2^p (1 - 2^{-p+1})}{4} \approx \frac{N}{8} 2^p \] \hspace{5cm} (11)
AT). Conventional Full Width adder tree and state-of-the-art
analyzed for, the proposed fixed-width AT design (MT-FX-
Different error performance metrics are compared and
A. Analyzing and Comparing Different Error Measurements
 ease.

Consequently, the approximate hardware complexity and error performance metrics of
the tree design. It is crucial to examine

The most classic error measurements for an
approximate arithmetic design includes Maximum Error
Distance (MED), Average Error Distance (AED) and Average
Accuracy (AAC), [10].

MED is calculated by taking the maximum of the absolute value
of the difference between the output of the full-width AT and
the output of the fixed-width AT. It is given by:

\[ \text{med} = \max(\text{abs}(y_0 - y_i)) \]  \hspace{1cm} (14)

where \( y_0 \) is the output of the full-width AT and \( y_i \) is the output of
the fixed-width AT.

AED is calculated by taking the average of the absolute value
of the difference between the output of the full-width AT and
the output of the fixed-width AT. It is given by:

\[ \text{aed} = \frac{\text{avg}(\text{abs}(y_0 - y_i))}{y_0} \]  \hspace{1cm} (15)

where \( y_0 \) is the output of the full-width AT and \( y_i \) is the output
of the fixed-width AT.

AAC is calculated by first taking the difference between the
output of the Full Width AT \( y_0 \) and the absolute ED, where ED
is the difference between the approximate and accurate sum
and then this is divided by \( y_0 \). Average of this taken for all the
10,000 samples. Finally, the percentage value is calculated by
multiplying the average results by 100. AAC is given by:

\[ \text{aac} = \left( 1 - \left( \frac{\text{abs}(\text{ed})}{y_0} \right) \right) \times 100 \]  \hspace{1cm} (16)

Predominantly the error metrics are mostly dependent on the
number of the inputs (i.e., value of \( N \)) rather than the size of
the inputs. Table 4 shows the error performance metrics of all state-
of-the-art fixed width and proposed fixed width AT for \( N = 8,16 \)
and \( w = 8,12,16 \). For \( N = 8 \), MT-FX-AT has the lowest AED
and FX-AT-PT has the lowest MED when compared to other
designs. The proposed design also aims to maintain a uniform
MED with change in \( w \). Taking AAC into consideration MT-
FX-AT has the highest percentage and therefore this proves to
have an acceptable tradeoff whilst maintaining good accuracy.

B. Analyzing and Comparing Synthesized Results
All the six designs were simulated and synthesized on
Cadence using the SAED 90nm typical cell library. The metrics
being compared are area and power and are discussed in Table
5. Proposed MX-FX-AT has a 18.17% and 5.05% decrease in
area when compared to FX-AT-PT and FX-AT-DT respectively,
while maintaining the MED close to that of FX-
AT-PT. When compared to TFX-AT and ITFX-AT, area of
the proposed MX-FX-AT has a 41.99% and 16.24% increase but
MED has a 63.64% and 38.46% improvement, respectively.

IV. RESULTS AND DISCUSSIONS

Full-width AT, which involves more logic complexity than
fixed-width AT, precisely calculates the result. However, fixed
width AT generates some inaccurate output by improving
performance metrics of the tree design. It is crucial to examine
the hardware complexity and error performance metrics of
designed fixed width adder tree, by doing so the approximate
adder-tree structure for a desired application can be picked with
ease.

A. Analyzing and Comparing Different Error Measurements

Different error performance metrics are compared and
analyzed for, the proposed fixed-width AT design (MT-FX-

Table 2 Truth table analysis of AXA block of proposed Adder tree design

<table>
<thead>
<tr>
<th>A2</th>
<th>B2</th>
<th>A1</th>
<th>B1</th>
<th>S1</th>
<th>C1</th>
<th>X</th>
<th>S2</th>
<th>Cout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig.11 AXA* block for radix-4 addition with 1-bit fixed bias and AXA block for only radix-4 addition

Table.2 Truth table analysis of AXA block of proposed Adder tree design

<table>
<thead>
<tr>
<th>A2</th>
<th>B2</th>
<th>A1</th>
<th>B1</th>
<th>S1</th>
<th>C1</th>
<th>X</th>
<th>S2</th>
<th>Cout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

MED has a 63.64% and 38.46% improvement, respectively.

AT-PT. When compared to TFX-AT and ITFX-AT, area of the
respectively, while maintaining the MED close to that of FX-
area when compared to FX-AT-PT and FX-AT-DT of-the-art fixed width and proposed fixed width AT for \( N = 8,16 \)
inputs. Table 4 shows the error performance metrics of all state-
number of the inputs (i.e., value of \( N \)) rather than the size of
inputs. Table 4 shows the error performance metrics of all state-
of-the-art fixed width and proposed fixed width AT for \( N = 8,16 \)
and \( w = 8,12,16 \). For \( N = 8 \), MT-FX-AT has the lowest AED
and FX-AT-PT has the lowest MED when compared to other
designs. The proposed design also aims to maintain a uniform
MED with change in \( w \). Taking AAC into consideration MT-
FX-AT has the highest percentage and therefore this proves to
have an acceptable tradeoff whilst maintaining good accuracy.

B. Analyzing and Comparing Synthesized Results
All the six designs were simulated and synthesized on
Cadence using the SAED 90nm typical cell library. The metrics
being compared are area and power and are discussed in Table
5. Proposed MX-FX-AT has a 18.17% and 5.05% decrease in
area when compared to FX-AT-PT and FX-AT-DT respectively,
while maintaining the MED close to that of FX-
AT-PT. When compared to TFX-AT and ITFX-AT, area of
the proposed MX-FX-AT has a 41.99% and 16.24% increase but
MED has a 63.64% and 38.46% improvement, respectively.
C. Studying Signal Processing Applications Using Approximate Arithmetic Structures

Additions and multiplications make up the majority of multimedia DSP algorithms. Multiplications are equivalent to shifts and additions. As a result, adders may be seen as the fundamental building blocks for various algorithms. The majority of DSP algorithms used in multimedia systems have intrinsic fault tolerance, which is an interesting trait. So, minor mistakes in intermediate outputs might not significantly affect the ultimate output quality. We concentrate on image processing and describe the outcomes of this algorithm by employing our approximate adder tree structures. To prevent a significant decline in output quality, we solely apply approximation in the LSPs.

We have considered an 8-point Walsh-Hadamard transform (WHT) for processing the images on MATLAB. The original full width adders in WHT algorithm are replaced with existing and proposed fixed width adder tree.

We have examined three digital images as shown in Fig.12, with a spatial resolution of 512X512. PSNR was evaluated to determine the quality of the reconstructed images. The PSNR ratio is used to measure how robust the signal is in comparison to noise or distortion in an image. Less visual distortion or a better reconstructed image is indicated by a high PSNR score and it is measured in terms of decibels. Image processing was carried out, where an original image was translated into a matrix of size $1 \times 262144$, which was then processed by forward and inverse Walsh-Hadamard transform (WHT). WHT is calculated using the formula:

$$y_n = \frac{1}{N} \sum_{i=0}^{N-1} x_i \times WAL(n, i)$$ (17)

$y$ is the output vector returned by WHT, $N$ is the number of inputs to the adder tree and $n$ and $i$ represents the rows and columns of the unitary Walsh matrix. $x$ is a matrix that contains the pixel values of the input image.

The proposed design achieves an improved PSNR score when compared to the ITFX-AT and also maintains a very similar score as that of FX-AT-PT. The scores are detailed in Table 3.

![Original Image](image1.png)

<table>
<thead>
<tr>
<th>Original Image</th>
<th>FX-AT-PT</th>
<th>PSNR = 73.4</th>
<th>ITFX-AT</th>
<th>PSNR = 65.4</th>
<th>MT-FX-AT</th>
<th>PSNR = 70.25</th>
</tr>
</thead>
</table>

![Original Image](image2.png)

<table>
<thead>
<tr>
<th>Original Image</th>
<th>FX-AT-PT</th>
<th>PSNR = 90.23</th>
<th>ITFX-AT</th>
<th>PSNR = 81.35</th>
<th>MT-FX-AT</th>
<th>PSNR = 86.24</th>
</tr>
</thead>
</table>

![Original Image](image3.png)

<table>
<thead>
<tr>
<th>Original Image</th>
<th>FX-AT-PT</th>
<th>PSNR = 83.04</th>
<th>ITFX-AT</th>
<th>PSNR = 79.0</th>
<th>MT-FX-AT</th>
<th>PSNR = 81.7</th>
</tr>
</thead>
</table>

Fig.12 Three different sets of digital images and their PSNR scores (in dB) obtained using various FX-AT

V. Conclusion

The innate error tolerance of signal processing applications was exploited in this study to create a fixed width adder tree to trade off performance and error metrics. The impact on output quality was relatively negligible when the inaccuracies caused by these approximations were represented at a high level in WHT. The proposed design incorporates multiplexers to compute the approximations in $\sigma_{major}$. A bias estimation approach is made use of to compensate and maintain minimal error, induced due to truncation of data.

Proposed design has a 18.17% and 5.05% decrease in area when compared to FX-AT-PT and FX-AT-DT respectively, while maintaining the MED close to that of FX-AT-PT.

When compared to TFX-AT and ITFX-AT, area of the proposed MX-FX-AT has a 41.99% and 16.24% increase but MED has a 63.64% and 38.46% improvement, respectively. Moreover, the designed adder tree maintains constant MED irrespective of word length for N=8,16 which is more
demanding for high word length data application. Additionally, WHT was implemented for three different digital images using the proposed design and the PSNR score of proposed MT-FX-AT was relatively higher when compared to the PSNR score of ITFX-AT.

Our approach of using multiplexers aids in sustaining nominal error and this trait proves to be essential for different variants of signal processing applications.

### Table 4 Error analysis

<table>
<thead>
<tr>
<th>DESIGNS</th>
<th>N</th>
<th>WORD LENGTH w=8</th>
<th>AED</th>
<th>MED</th>
<th>AAC (%)</th>
<th>WORD LENGTH w=12</th>
<th>AED</th>
<th>MED</th>
<th>AAC (%)</th>
<th>WORD LENGTH w=16</th>
<th>AED</th>
<th>MED</th>
<th>AAC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL WIDTH [1]</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>FX-AT-PT [1]</td>
<td>8</td>
<td>3.50</td>
<td>7</td>
<td>0</td>
<td>99.66</td>
<td>3.50</td>
<td>7</td>
<td>0</td>
<td>99.98</td>
<td>3.50</td>
<td>7</td>
<td>0</td>
<td>99.98</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>7.30</td>
<td>15</td>
<td>0</td>
<td>99.64</td>
<td>7.50</td>
<td>15</td>
<td>0</td>
<td>99.97</td>
<td>7.50</td>
<td>15</td>
<td>0</td>
<td>99.99</td>
</tr>
<tr>
<td>FX-AT-DT [1]</td>
<td>8</td>
<td>6.00</td>
<td>12</td>
<td>0</td>
<td>99.42</td>
<td>6.00</td>
<td>12</td>
<td>0</td>
<td>99.96</td>
<td>6.10</td>
<td>12</td>
<td>0</td>
<td>99.99</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>16.00</td>
<td>31</td>
<td>0</td>
<td>99.21</td>
<td>16.00</td>
<td>31</td>
<td>0</td>
<td>99.95</td>
<td>16.10</td>
<td>31</td>
<td>0</td>
<td>99.99</td>
</tr>
<tr>
<td>TFX-AT [1]</td>
<td>8</td>
<td>6.00</td>
<td>22</td>
<td>0</td>
<td>99.41</td>
<td>6.20</td>
<td>26</td>
<td>0</td>
<td>99.96</td>
<td>6.10</td>
<td>26</td>
<td>0</td>
<td>99.99</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>16.30</td>
<td>64</td>
<td>0</td>
<td>99.20</td>
<td>16.10</td>
<td>76</td>
<td>0</td>
<td>99.95</td>
<td>16.10</td>
<td>71</td>
<td>0</td>
<td>99.99</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>8.70</td>
<td>34</td>
<td>0</td>
<td>99.57</td>
<td>8.60</td>
<td>40</td>
<td>0</td>
<td>99.97</td>
<td>8.70</td>
<td>40</td>
<td>0</td>
<td>99.99</td>
</tr>
<tr>
<td>PROPOSED MT-FX-AT</td>
<td>8</td>
<td>2.98*</td>
<td>8</td>
<td>0</td>
<td>99.89</td>
<td>2.99*</td>
<td>8</td>
<td>0</td>
<td>99.97</td>
<td>2.99*</td>
<td>8</td>
<td>0</td>
<td>99.97</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>7.90</td>
<td>22</td>
<td>0</td>
<td>99.83</td>
<td>7.92</td>
<td>22</td>
<td>0</td>
<td>99.91</td>
<td>7.92</td>
<td>22</td>
<td>0</td>
<td>99.97</td>
</tr>
</tbody>
</table>

* It can be observed that the proposed MT-FX-AT has a significantly lesser error rate when compared to TFX-AT and ITFX-AT and hence the proposed design shows a trade-off in area and power. As the aim is to achieve high accuracy.

### Table 5 Area power analysis

<table>
<thead>
<tr>
<th>DESIGNS</th>
<th>N</th>
<th>AREA</th>
<th>POWER</th>
<th>AREA</th>
<th>POWER</th>
<th>AREA</th>
<th>POWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL WIDTH [1]</td>
<td>8</td>
<td>1610</td>
<td>257</td>
<td>2539</td>
<td>283</td>
<td>3476</td>
<td>295</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3261</td>
<td>232</td>
<td>5135</td>
<td>376</td>
<td>7200</td>
<td>537</td>
</tr>
<tr>
<td>FX-AT-PT [1]</td>
<td>8</td>
<td>1723</td>
<td>275</td>
<td>2717</td>
<td>295</td>
<td>4008</td>
<td>367</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3489</td>
<td>363</td>
<td>5394</td>
<td>391</td>
<td>7902</td>
<td>568</td>
</tr>
<tr>
<td>FX-AT-DT [1]</td>
<td>8</td>
<td>1485</td>
<td>225</td>
<td>2342</td>
<td>241</td>
<td>3455</td>
<td>246</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3028</td>
<td>372</td>
<td>4937</td>
<td>735</td>
<td>6734</td>
<td>1256</td>
</tr>
<tr>
<td>TFX-AT [1]</td>
<td>8</td>
<td>993</td>
<td>146</td>
<td>1287</td>
<td>193</td>
<td>1725</td>
<td>207</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>2438</td>
<td>171</td>
<td>3858</td>
<td>278</td>
<td>5255</td>
<td>396</td>
</tr>
<tr>
<td>ITFX-AT [1]</td>
<td>8</td>
<td>1213</td>
<td>183</td>
<td>2035</td>
<td>283</td>
<td>2807</td>
<td>310</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>2540</td>
<td>251</td>
<td>4306</td>
<td>313</td>
<td>5903</td>
<td>455</td>
</tr>
<tr>
<td>PROPOSED MT-FX-AT</td>
<td>8</td>
<td>1410*</td>
<td>225</td>
<td>2253</td>
<td>248</td>
<td>3172</td>
<td>255</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>2855</td>
<td>207</td>
<td>4525</td>
<td>338</td>
<td>6265</td>
<td>497</td>
</tr>
</tbody>
</table>

* It can be observed that the proposed MT-FX-AT has a significantly lesser error rate when compared to TFX-AT and ITFX-AT and hence the proposed design shows a trade-off in area and power. As the aim is to achieve high accuracy.

### Table 3 Comparison of PSNR scores (dB)

<table>
<thead>
<tr>
<th>DESIGNS</th>
<th>Image 1: Lady</th>
<th>Image 2: House</th>
<th>Image 3: Pepper</th>
</tr>
</thead>
<tbody>
<tr>
<td>FX-AT-PT</td>
<td>83.4</td>
<td>73.4</td>
<td>90.23</td>
</tr>
<tr>
<td>ITFX-AT</td>
<td>79.0</td>
<td>65.4</td>
<td>81.35</td>
</tr>
<tr>
<td>PROPOSED MT-FX-AT</td>
<td>81.7</td>
<td>70.25</td>
<td>86.24</td>
</tr>
</tbody>
</table>
REFERENCES


