Design and FPGA implementation of image compression based fuzzy technique

Pradeep S
M.Tech Student, Dept. of ECE Engg., SJBIT Bangalore, psn026@gmail.com.

Abstract- Fuzzy logic is a way to embed an engineer’s experience into the system. In recent years, many researchers have applied the fuzzy logic to develop new techniques for contrast improvement. Fuzzy logic is a well-known rather simple approach with good visual results, but proposed fuzzy operation algorithm is default nonlinear. Here proposed algorithm is a default nonlinear thus not straightforward applicable on the JPEG bit stream. However, it is possible when the right combination is found. Image compression is one of the major image processing techniques that are widely used in medical, automotive, consumer and military applications. In this paper fuzzy technique has been used in image compression with Discrete Wavelet Transforms (DWT) technique. Discrete wavelet transforms is the most popular transformation technique adopted for image compression. Image compression has become important as storage or transmission of images requires large amount of bandwidth. In order to minimize the complexity of DWT, fuzzy technique has been proposed and implemented on FPGA.

Key Words: Discrete wavelet transforms (DWT), Fuzzy logic, Fuzzy Intensification Operator, Image Compression.

I.INTRODUCTION

The digital multimedia is popular nowadays because of their highly perceptual effects and the advanced development its corresponding technology. However, it often requires a large amount of data to store these multimedia contents due to the complex information they may encounter. Besides, the requirement of resolution is much higher than before, such that the data size of the image is surprisingly large. As images convey more information to a user, it is many of the equipment today have image displays and interfaces. Image storage on these smaller, handled devices is a challenge as they occupy huge storage space also image transmission requires higher bandwidth. In particular, there is an ever increasing use of digital images, especially in the internet. This has led to the development of image compression techniques that enable more economical use of storage space and faster data transfer. Implementation of wavelet transform is complex, time consuming and costly. FPGAs provide a new way to digital signal processing. Implementation of DWT on FPGAs provides higher processing speeds and lower costs than other implementation methods. DA (Distributed Arithmetic) can be used to design low-pass and high-pass filters to perform FDWT and Inverse DWT [1]. The problem associated with DA is that the requirement of memory/LUT capacity increases exponentially with the order of the filter. Even ROM access time will be increased which becomes a bottleneck for system speed [3]. This issue can be overcome by LUT decomposition, reducing the hardware complexity instead of using single large LUTs. It also further increases the computation speed and minimizes the dynamic power consumption of the system [3]. Multilayer Perceptron’s based on Fuzzy Flip-Flops [7]. Fuzzy logic is a useful technique in image contrast enhancement. It is provided by the application of fuzzy sets theory and fuzzy inference systems [2]. Using the inverse function [5] the implementation of arithmetic operations on fuzzy numbers is computationally complex and the implementation of the extension principle [6] is equivalent to solving a nonlinear programming problem. There exist a number of universal data compression algorithms that can compress almost any kind of data. These methods are lossless in the sense that they retain all the information of the compressed data. However, they do not take advantage of the 2-dimensional nature of the image data. Images have certain statistical properties, which can be exploited by encoders especially designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more storage space and bandwidth. This means lossy image compression technique can be used in this area. The Discrete Wavelet Transform (DWT) is being increasingly used for image coding. This is because the DWT can decompose the signals into different sub-bands with both time and frequency information. It also supports features like progressive image transmission, compressed image manipulation, and region of interest coding. In wavelet transforms, the original signal is divided into frequency resolution and time resolution contents. In this work explores the implementation of 2-Dimensional DWT algorithm on FPGA for image compression using fuzzy logic. Several architectures have been proposed such as convolution based, lifting based and B-spline based. These architectures are compared in terms of hardware complexity, critical path, and registers. As for the 2D-DWT the large amount of the memory access and the area required that becomes the most critical issue so it also affects the total area and power. Finally, a flexible and efficient proposed architecture has been selected for designing two-level 2-D-DWT. The proposed architecture has advantages over other architectures such as very less number of multipliers, less area and less memory utilization. The paper is organized as follows. In Section II, the hardware structure of the fuzzy logic and DWT is discussed and the system performance is evaluated and the typical implementation and timing analysis characteristics are presented. Finally, simulation results and conclusions are discussed in Section III and IV respectively.

II. PROPOSED METHOD

![Proposed Block Diagram](image-url)
A. DISCRETE WAVELET TRANSFORM

The Wavelet Series is just a sampled version of continuous wavelet transform and its computation may consume significant amount of time and resources, depending on the resolution required. The Discrete Wavelet Transform (DWT), which is based on sub-band coding, is found to yield a fast computation of Wavelet Transform. It is easy to implement and reduces the computation time and resources required.

In the case of DWT, a time-scale representation of the digital signal is obtained using digital filtering techniques. The signal to be analyzed is passed through filters with different cutoff frequencies at different scales.

B. DWT ARCHITECTURE

DWT processor transforms the spatial domain pixels into frequency domain information that are represented in multiple sub-bands, representing different time scale and frequency points. Human visual system is very much sensitive to low frequency and hence, the decomposed data available in the lower sub-band region and is selected and transmitted, information in the higher sub-bands regions are rejected depending upon required information content. In order to extract the low frequency and high frequency sub-bands

![DWT Architecture](image)

Figure 2: DWT Architecture

C. INVERSE DWT

Just as a forward transform is used to separate the image data into various classes of importance a reverse transform is used to resemble the various classes of data into a reconstructed image. A pair of high pass and low pass filters is used here also. Then filter pair is called the synthesis filter pair. The filtering procedure is just the opposite. We start from the topmost level, apply the filters column wise first and then row wise and proceed to the next level, till we reach the first level. In this section the theoretical background and algorithm development is discussed. The first recorded mention of what is now called a “wavelet” seems to be in 1909, in a thesis by Alfred Haar. An image is represented as a two dimensional (2D) array of coefficients, each coefficient representing the brightness level in that point. When looking from a higher perspective, it is not possible to differentiate between coefficients as more important ones, and lesser important ones. But thinking more intuitively, it is possible. Most natural images have smooth color variations, with the fine details being represented as sharp edges in between the smooth variations.

Technically, the smooth variations in color can be termed as low frequency components and the sharp variations as high frequency components. The low frequency DWT architecture shown in figure below is used. As shown in the figure, input image consisting rows and columns are transformed using high pass and low pass filters. A low pass filter and a high pass filter are chosen, such that they exactly halve the frequency range between themselves. The filter pass is called the analysis filter pair. First the low pass filter is applied for each row of data, thereby getting the low frequency components of the row. But since the low pass filter is a half band filter, the output data contains frequencies only in the first half of the original frequency range. So they can be sub sampled by two, so that the output data now contains only half the original number of samples. Now the high pass filter is applied for the same row of data, and similarly the high pass components are separated and placed by the side of the low pass components. This procedure is done for all rows. Next, the filtering is done for each column of the intermediate data. The resulting two dimensional array of coefficients contains four bands of data, each labeled as LL (low-low), HL (high-low), LH (Low-High) and HH (High-High). The LL band can be decomposed once again in the same manner, thereby producing even more sub bands. This can be done up to any level, thereby resulting in a pyramidal decomposition as shown. The LL band at the highest level can be classified as most important and the other detail bands can be classified as of lesser importance, with the degree of importance decreasing from the top of the pyramid to the bands at the bottom. components (smooth variations) constitute the base of an image, and the high frequency components (the edges which give the detail) add upon them to refine the image, thereby giving a detailed image. Hence the averages/smooth variations are demanding more importance than the details. In wavelet analysis, a signal can be separated into approximations or averages and detail or coefficients. Averages are the high scale, low frequency components of the signal. The details are the low scale, high frequency components. If we perform forward transform on a real digital signal, we wind up with twice as much data as we started with. That’s why after filtering down sampling has to be done. The inverse process is how those components can be assembled back into the original signal without loss of information. This process is called reconstruction or synthesis. The mathematical manipulation that affects synthesis is called the inverse discrete wavelet transform. The original signal is reconstructed from the wavelet coefficients. Where wavelet analysis involves filtering and down sampling, the wavelet reconstruction process consists of up sampling and filtering. DWT outputs are computed every clock cycle. In the computation process fuzzy arithmetic operators and fuzzy neuro D-flip-flop has been used in order to increase the computation speed which is very essential in compression.

![Fuzzy D-flip-flop](image)

Figure 3: Fuzzy D-flip-flop
D. IMAGE PROCESSING THROUGH INTENSIFICATION OPERATOR IN THE COMPRESSED DOMAIN.

This method is to link DWT and proposed algorithm (Fuzzy Intensification).

Steps are explained as below:
1. First read the input image, here images of resolution (100x100) is used for Image Processing.
2. Fuzzification is described by

\[ B(l) = a.l + b \] for any \( l \) in the image

The value of \( a \) is calculated by

\[ a = \frac{0.5}{\text{max}(l_{\text{min}}) - l_{\text{max}}} \]

The values of the second parameter \( b \) are obtained from \( B(l_{\text{th}}) B(l_{\text{th}})=0.5 \Rightarrow a.l_{\text{th}}+b=0.5 \Rightarrow b=0.5-a.l_{\text{th}} \)

\( I_m=0, I_M=255, I_{\text{th}}=128 \) this value is considered by seeing below figure 4.

3. Defuzzification is given by \( B'(l) \)

\[ B'(l) = \text{INT}(B(l)) = \begin{cases} 2.(a.l + b)^2, & \text{if } 0 \leq (a.l + b) < 0.5 \\ \end{cases} \]

Figure 4: Fuzzy intensification operator

E. IP BLOCK

In electronic design a semiconductor intellectual property core, IP core, or IP block is a reusable unit of logic, cell, or chip layout design that is the intellectual property of one party. IP cores may be licensed to another party or can be owned and used by a single party alone. The term is derived from the licensing of the patent and source code copyright intellectual property rights that subsist in the design. IP cores can be used as building blocks within ASIC chip designs or FPGA logic designs.

In this project IP core has been used for both the memory read and memory write operations. Hence reduces the coding complexity involved in the whole process and optimized design can be achieved.

A large size of image has been resized into 100 by 100 pixel image. A pixel values are stored into the memory. The stored values of pixels from the memory read out serially and perform fuzzification, and then based on DWT compression and decompression, De-fuzzification has been performed. The recovered Pixel values have been written back to RAM. Once the whole process gets over, the generated bit file is dumped into a FPGA. Through VGA port both the original and recovered image can be seen in the monitor.

### III. SIMULATION RESULTS

Simulation result for the proposed design is shown in figure 5. Input vectors that were obtained from MATLAB test inputs were used for validating the HDL results. Input vectors are stored in an ROM and are read into the modified Fuzzy-DWT architecture. The decomposed outputs are stored back and are also displayed using simulation waveforms.

![Simulation Results](image)

Table.1. Comparison exiting with proposed method

<table>
<thead>
<tr>
<th></th>
<th>Signed (basic arch)</th>
<th>Signed (split LUT)</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertex II Pro FPGA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of 4 input LUTs</td>
<td>312</td>
<td>358</td>
<td>339</td>
</tr>
<tr>
<td>No. of Flip-flops</td>
<td>107</td>
<td>134</td>
<td>113</td>
</tr>
<tr>
<td>Delay</td>
<td>11.359ns</td>
<td>3.469ns</td>
<td>16.259ns</td>
</tr>
<tr>
<td>Total power (W)</td>
<td>0.02717W</td>
<td>0.04857W</td>
<td>0.031W</td>
</tr>
<tr>
<td>Total Quiescent power (W)</td>
<td>0.03542W</td>
<td>0.0374W</td>
<td>0.028W</td>
</tr>
<tr>
<td>Total Dynamic power (W)</td>
<td>0.00535W</td>
<td>0.04875W</td>
<td>0.005W</td>
</tr>
</tbody>
</table>

Figure 5: shows the compressed and decompressed output in terms of pixel values

![Simulation Results Table](image)

Figure 6: Original and Reconstructed Image
IV. CONCLUSION

The developed algorithm is found very efficient for gray image compression. During the analysis it is found that, developed algorithm provides higher compression ratio as compare to normal discrete wavelet transform. In addition to this fuzzyfied discrete wavelet transform based developed technique is also able to keep error between input image and reconstructed image in allowable range, though it is generating slightly higher error but at the same time the compression ratio is much higher than available NDWT technique. The advantage of the developed algorithm is, Fuzzy fication performed before using DWT because normal DWT cannot able to handle imperfections presented in the input images.

An image compression algorithm was simulated using Verilog to comprehend the process of image compression. The relative area and speed efficiencies of Fuzzy turn out to be good on hardware implementation on FPGA. An implementation process included fuzzy arithmetic operators which make the computation quite faster and fuzzy helps to enhance the image better way in the presence of noise. The reconstructed image has got better edges in this design compared to the previous work. In the implementation results it is clear that Fuzzy logic based architectures have an area, speed and simplicity advantage over any other method based on implementations.

REFERENCES
