

# Hardware Implementation of DWT for Image compression using SPIHT Algorithm

Takkiti Rajashekar Reddy<sup>#1</sup>, Rangu Srikanth<sup>\*2</sup>

<sup>#1</sup>Student M.Tech (VLSI), Assistant Professor<sup>\*2</sup>

Department of ECE

Jayamukhi Institute of Technology & Sciences

Narsampet, Warangal, AP, India

**Abstract**— In this paper, a DWT-based image processing system is developed on Xilinx Spartan3 Field Programmable Gate Array (FPGA) device using embedded development kit (EDK) tools from Xilinx. Two different hardware architectures of two dimensional (2-D) DWT have been implemented as a coprocessor in an embedded system. One is direct implementation of 2-D DWT by cascading two 1-D DWT. Another is 2-D DWT implementation with control and architecture optimization. In addition, the hardware cost of these two architectures is compared for benchmark images.

**Keywords**—Discrete Wavelet Transform (DWT), FPGA, EDK, Micro Blaze, FSL Introduction

## I. INTRODUCTION

Many embedded DSP systems make use of a DSP chip utilizing a single processing core with high-bandwidth memory connections to implement DSP algorithms. In this investigation, we developed an alternative approach based on an embedded FPGA system for image processing. Field Programmable Gate Array (FPGA) is widely used in embedded applications such as automotive, communications, industrial automation, motor control, medical imaging etc. FPGA is chosen due to its reconfigurable ability. Without requiring hardware change-out, the use of FPGA type devices expands the product life by updating data stream files. FPGAs have grown to have the capability to hold an entire system on a single chip; meanwhile, it allows in-platform testing and debugging of the system. Furthermore, it offers the opportunity of utilizing hardware/software co-design to develop a high performance system for different applications by incorporating processors (hardware core processor or software core processor), on-chip busses, memory, and hardware accelerators for specific software functions.

In this paper, a DWT-based image processing system is developed on a Xilinx Spartan3 Field Programmable Gate Array (FPGA) device using an embedded development kit (EDK) from Xilinx. DWT is one of the most popular transform coding techniques for image and video compression. The video processing and image compression

standards such as JPEG, MPEG, and H.26x have adopted DWT as the transform coder [1-3]. Consequently, DWT is chosen as the application algorithm for the embedded system.

This paper is organized as follows: Section II briefly reviews discrete cosine transform. Section III discusses the design flow. Section IV covers different architecture for DWT co-processor and compares their performance. Section V is the conclusion part.

## II. WAVELET TRANSFORMS

The best way to describe discrete wavelet transform is through a series of cascaded filters. We first consider the FIR-based discrete transform. The input image  $X$  is fed into a low-pass filter  $h'$  and a high-pass filter  $g'$  separately. The output of the two filters are then sub sampled, resulting low-pass sub band  $y_L$  and high-pass sub band  $y_H$ .

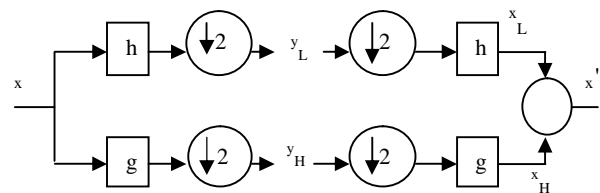


Fig.1 DWT analysis and synthesis system

The original signal can be reconstructed by synthesis filters  $h$  and  $g$  which take the up sampled  $y_L$  and  $y_H$  as inputs [9]. To perform the forward DWT the standard uses a 1-D sub band decomposition of a 1-D set of samples into low-pass samples and high-pass samples. Low pass samples represent a down sampled low-resolution version of the original set. High-pass samples represent a down sampled residual version of the original set, needed for the perfect reconstruction of the original set.

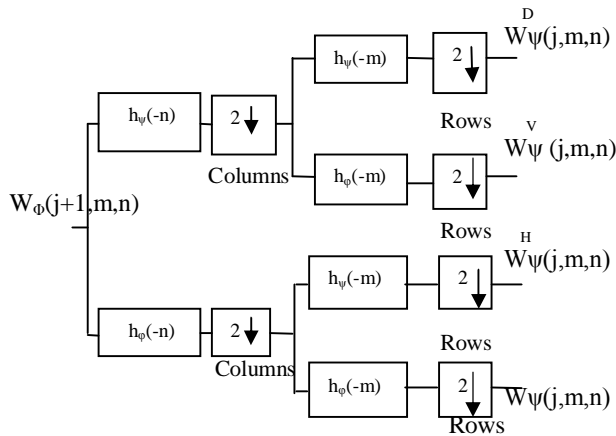


Fig. 2 The 2-D DWT analysis filter bank

#### A. QUANTIZATION

After transformation, all coefficients are quantized. Quantization is the process by which the coefficients are reduced in precision. Each of the transform coefficients  $a_{bs}(u, v)$  of the sub band  $b$  is quantized to the value  $e_b(u, v)$  according to the formula

$$q_b(u, v) = \text{sign}(a_b(u, v)) \left\lceil \frac{a_b(u, v)}{\Delta_b} \right\rceil$$

The quantization step is represented relative to the dynamic range  $R_b$  of sub band  $b$  by the exponent  $\epsilon_b$  and mantissa  $\mu_b$  as:

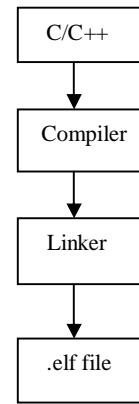
$$\Delta_b = 2^{(R_b - \epsilon_b)} \left( 1 + \frac{\mu_b}{2^{11}} \right)$$

The dynamic range  $R_b$  depends on the number of bits used to represent the original image component and on the choice of the wavelet transform. All quantized transform coefficients are signed values even when the original components are unsigned. These coefficients are expressed in a sign magnitude representation prior to coding. For reversible compression, the quantization step size is required to be 1. This implies that  $\mu_b = 0$  and  $R_b = \epsilon_b$ .

#### III. DESIGN FLOW

To build an embedded system on Xilinx FPGAs, the embedded development kit (EDK) is used to complete the reconfigurable design. Figure 1 shows the design flow.

#### SOFTWARE FLOW



#### HARDWARE FLOW

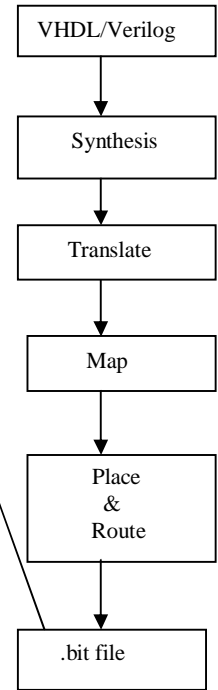


Fig. 3 Design flow

Unlike the design flow in the traditional software design using C/C++ language or hardware design using hardware description languages, the EDK enables the integration of both hardware and software components of an embedded system. For the hardware side, the design entry from VHDL/Verilog is first synthesized into a gate-level netlist, and then translated into the primitives, mapped on the specific device resources such as Look-up tables, flip-flops, and block memories.

The location and interconnections of these device resources are then placed and routed to meet with the timing constraints.

A downloadable .bit file is created for the whole hardware platform. The software side follows the standard embedded software flow to compile the source codes into an executable and linkable file (ELF) format. Meanwhile, a microprocessor software specification (MSS) file and a microprocessor hardware specification (MHS) file are used to define software structure and hardware connection of the system. The EDK uses these files to control the design flow and eventually merge the system into a single downloadable file. The whole design runs on a real-time operating system (RTOS).

#### IV. DWT CO-PROCESSOR

The implementation of hybrid method for image compression for different images is a novel algorithm. The hybrid method for image compression algorithm is as follows:

- In the level shifting step a value of 128 is subtracted from each and every pixel to get the level shifted image as  $g(m, n) = f(m, n) - 128$ .
- Computation of 2D-DWT of the level shifted image.
- Performing Quantization of the DWT matrix based on the energy in the sub band.
- Lossless predictive coding is applied to LL band.

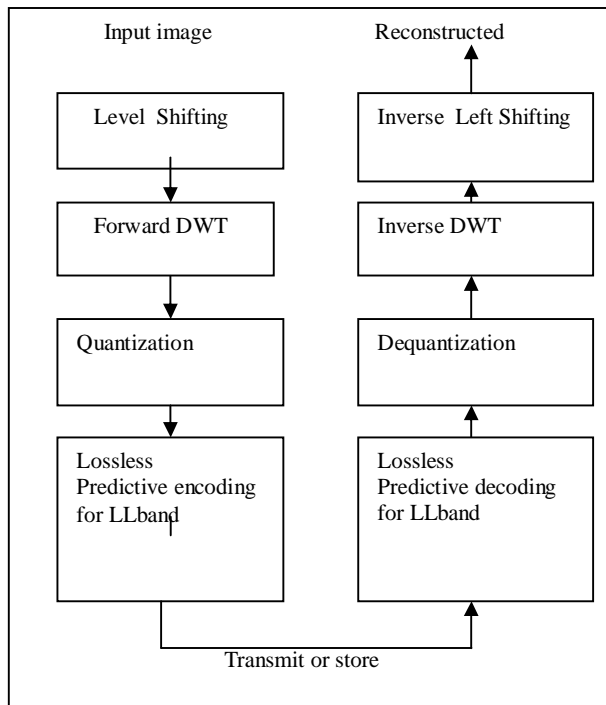


Fig. 4 The block diagram of encoder and decoder.

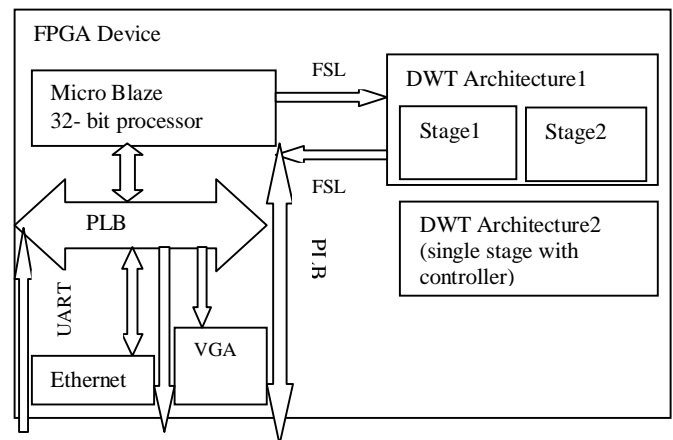
#### A. LOSSLESS PREDICTIVE ENCODING

Predictive coding is an image compression technique which uses a compact model of an image to predict pixel values of an image based on the values of neighboring pixels. A model of an image is a function  $model(x; y)$ , which computes (predicts) the pixel value at coordinate  $(x; y)$  of an image, given the values of some neighbors of pixel  $(x; y)$ , where neighbors are pixels whose values are known. Typically, when processing an image in raster scan order (left to right, top to bottom), neighbors are selected from the pixels above and to the left of the current pixel. For example, a common set of neighbors used for predictive coding is the set  $\{(x-1, y-1), (x, y-1), (x+1, y-1), (x-1, y)\}$ . Linear predictive coding is a simple, special case of predictive coding in which the model simply takes the difference of the neighboring values. There are two expected sources of compression in

predictive coding based image compression (assuming that the predictive model is accurate enough). First, the coded image for each pixel should have a smaller magnitude than the corresponding pixel in the original image (therefore requiring fewer bits to transmit the coded image). Second, the coded image should have less entropy than the original message, since the model should remove many of the “principal components” of the image. To complete the compression, the quantized image is compressed using an entropy coding algorithm such as Huffman coding or arithmetic coding or proposed algorithm. If we transmit this compressed coded image then a receiver can reconstruct the original image by applying an analogous decoding procedure [10].

There are different ways to include processors inside Xilinx FPGA for System-on-a-Chip (SoC): PowerPC hard processor core, or Xilinx MicroBlaze soft processor core, or user-defined soft processor core in VHDL/Verilog. In this work, The 32-bit MicroBlaze processor is chosen because of the flexibility. The user can tailor the processor with or without advance features, based on the budget of hardware. The advance features include memory management unit, floating processing unit, hardware multiplier, hardware divider, instruction and data cache links etc. The architecture overview of the system is shown in Figure 2.

It can be seen that there are two different buses (i.e., processor local bus (PLB) and fast simplex link (FSL) bus) used in the system [5-6]. PLB follows IBM coreconnect bus architecture, which supports high-bandwidth master and slave devices, provides up to 128-bit data bus, up to 64-bit address bus and centralized bus arbitration. It is a type of shared bus. Besides the access overhead, PLB potentially has the risk of hardware/software incoherent due to bus arbitration. On the other hand, FSL supports point-to-point unidirectional communication. A pair of FSL buses (from processor to peripheral and from peripheral to processor) can form a dedicated high speed bus without arbitration mechanism. Xilinx provides C and assembly language support for easy access. Therefore, most of peripherals are connected to the processor through PLB; the DWT coprocessor is connected through FSL instead.



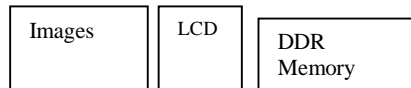


Fig. 5 DWT System Overview

The current system offers several methods for distributing the data. These methods are a UART, and VGA, and Ethernet controllers. The UART is used for providing an interface to a host computer, allowing user interaction with the system and facilitating data transfer. The VGA core produces a stand-alone real-time display. The Ethernet connection allows a convenient way to export the data for use and analysis on other systems. In our work, to validate the DWT coprocessor, an image data stream is formed using VISUAL BASIC, then transmitted from the host computer to FPGA board through UART port.

## V.EXPERIMENTAL RESULTS

Experiments are performed on gray level images to verify the proposed method. These images are represented by 8 bits/pixel and size is 128 x 128. Image used for experiments are shown in below figure.

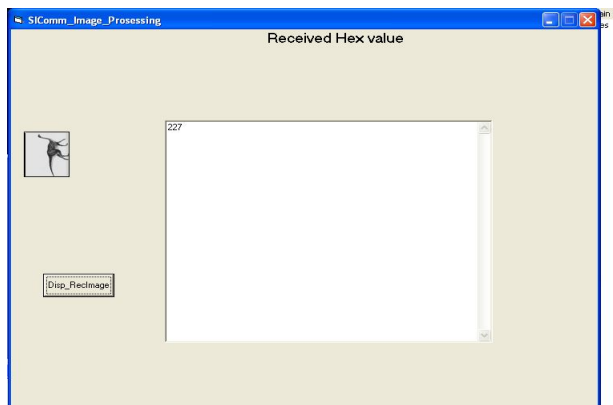


Fig. 6 Input images

The measurands used for proposed method are as follows:

The entropy (E) is defined as

Where  $s$  is the set of processed coefficients and  $p(e)$  is the probability of processed coefficients. By using entropy, number of bits required for compressed image is calculated.

An often used global objective quality measure is the mean square error (MSE) defined as

Where,  $n_{xm}$  is the number of total pixels.  $f(i,j)$  and  $f(i,j)'$  are the pixel values in the original and reconstructed image. The

peak to peak signal to noise ratio (PSNR in dB) [11-13] is calculated as

Usable gray level values range from 0 to 255. And the Compressed Image is showed in below Figure.

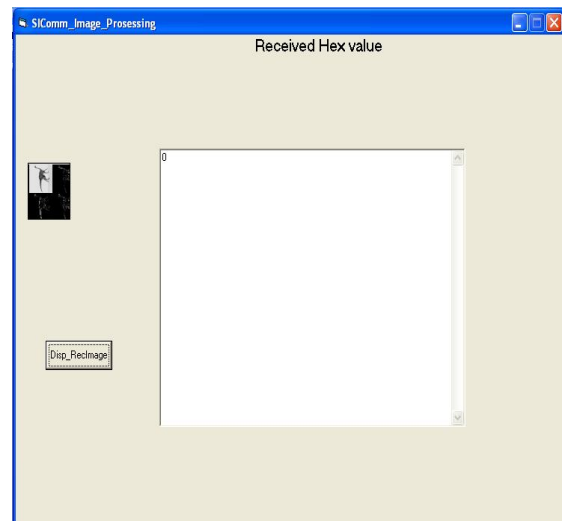


Fig. 7 Compressed image

And the synthesis report is below

Selected Device : 3s500efg320-4			
Number of Slices:	2649	out of 4656	56%
Number of Slice Flip Flops:	3343	out of 9312	35%
Number of 4 input LUTs:	3794	out of 9312	40%
Number used as logic:	3118		
Number used as Shift registers:	356		
Number used as RAMs:	320		
Number of IOs:	83		
Number of bonded IOBs:	40	out of 232	17%
IOB Flip Flops:	55		
Number of BRAMs:	7	out of 20	35%
Number of MULT18X18SIOs:	3	out of 20	15%
Number of GCLKs:	7	out of 24	29%
Number of DCMs:	2	out of 4	50%

Timing Summary:

Speed Grade: -4

Minimum period: 12.384ns (Maximum Frequency: 80.749MHz)  
Minimum input arrival time before clock: 41.553ns  
Maximum output required time after clock: 13.840ns  
Maximum combinational path delay: 3.344ns

Fig. 8 Synthesis report

## VI. CONCLUSIONS

In this paper, a DWT-based reconfigurable system is designed using the EDK tool. Hardware architectures of two dimensional (2-D) DWT have been implemented as a coprocessor in an embedded system. One is direct implementation of 2-D DWT by cascading two 1-D DWTs. Another is 2-D DWT implementation with control and architecture optimization. In addition, the hardware cost of these two architectures is compared for benchmark images. This type of work using EDK can be extended to other applications of embedded system. These two architectures applications compared for benchmark images. This type of work using EDK can be extended to other applications of embedded systems.

## REFERENCES

- [1]. Z. Fan and R. D. Queiroz. Maximum likelihood estimation of JPEG quantization table in the identification of Bitmap Compression. IEEE, 948-951, 2000.
- [2]. Charilaos Christopoulos, Athanassios Skodras, Touradj Ebrahimi, "The JPEG2000 still Image coding system: An overview". IEEE, 1103-1127, November 2000.
- [3]. G. K. Wallace, "The JPEG Still Picture Compression Standard", IEEE Trans. Consumer Electronics, Vol. 38, No 1, Feb. 1992
- [4]. W. B. Pennebaker and J. L. Mitchell, "JPEG: Still Image Data Compression Standard", Van Nostrand Reinhold, 1993.
- [5]. V. Bhaskaran and K. Konstantinides, "Image and Video Compression Standards:
- [6]. ISO/IEC JTC1/SC29/WG1 N505, "Call for contributions for JPEG 2000 (JTC 1.29.14, 15444):Image Coding System," March 1997
- [7]. ISO/IEC JTC1/SC29/WG1 N390R, "New work item: JPEG 2000 image coding system," March 1997.
- [8]. M. Boliek, C. Christopoulos and E. Majani (editors), "JPEG2000 Part I Final Draft International Standard," (ISO/IEC FDIS15444-1), ISO/IEC JTC1/SC29/
- [9]. T. Acharya and A. K. Ray, Image Processing: Principles and Applications. Hoboken, NJ: John Wiley & Sons, 2005
- [10]. Alex Fukunaga and Andre Stechert, Evolving Nonlinear Predictive Models for Lossless Image Compression with Genetic Programming,
- [11]. D.Salomon. Data compression. Springer, 2<sup>nd</sup> edition, 2000.
- [12]. Dr.K.Veera swamy, Dr.B.Chandra Mohan, Y.V.Bhaskar Reddy and Dr.S.Srinivasa Kumar, Image Compression and Watermarking scheme using Scalar Quantization. IJNGN, 2000.
- [13]. B.Shrestha, Dr.Charles,G.O'Hara and Dr. Nicolas H. Younan, JPEG2000:Image Quality metrics.ASPRS,2005.