ISSN 22773061

A Novel VLSI Architecture for SPHIT Encoder

Haritha Motupalle[1]
M.Tech scholar VLSI Design
Madina Engineering College
Kadapa-516003
hari.jetc@gmail.com
Syed Jahangir Badashah[2]
Assco Prof,ECE Dept
Madina Engineering College
Kadapa-516003
Syd_jahangir@yahoo.co.in

Abstract

In this Paper we propose a highly scalable image compression scheme based on the set partitioning in hierarchical trees (SPIHT) algorithm. Our algorithm called highly scalable SPIHT (HS-SPIHT), supports spatial and SNR scalability and provides a bit stream that can be easily adapted (reordered) to given bandwidth and resolution requirements by a simple transcoder (parser). The HS-SPIHT algorithm adds the spatial scalability feature without sacrificing the SNR embeddedness property as found in the original SPIHT bit stream. HS-SPIHT finds applications in progressive Web browsing, flexible image storage and retrieval, and image transmission over heterogeneous networks. Here we have written the core processor Microblaze is designed in VHDL (VHSIC hardware description language), implemented using XILINX ISE 8.1 Design suite the algorithm is written in system C Language and tested in SPARTAN-3 FPGA kit by interfacing a test circuit with the PC using the RS232 cable. The test results are seen to be satisfactory. The area taken and the speed of the algorithm are also evaluated.

Keywords- UART; VHDL; Softcore; Microblaze; SPHIT

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol 10, No 4
editor@cirworld.com
www.cirworld.com, member.cirworld.com

ISSN 22773061

I.INTRODUCTION

One of the major challenges in enabling mobile multimedia data services will be the need to process and wirelessly transmit a very large volume of data. While significant improvements in achievable bandwidth are expected with future wireless access technologies, improvements in battery technology will lag the rapidly growing energy requirements of future wireless data services. One approach to mitigate to this problem is to reduce the volume of multimedia data transmitted over the wireless channel via data compression techniques. This has motivated active research on multimedia data compression techniques such as JPEG [1,2], JPEG 2000 [2] and MPEG [3]. These approaches concentrate on achieving higher compression ratio without sacrificing the quality of the image. However these efforts ignore the energy consumption during compression and RF transmission.

Today hospitals handle their medical image data with computers. The use of computers and a network makes it possible to distribute the image data among the staff efficiently. As the health care is computerized new techniques and applications are developed, among them the MR and CT techniques[4]. MR and CT produce sequences of images (image stacks) each a cross-section of an object. The amount of data produced by these techniques is vast and this might be a problem when sending the data over a network. To overcome this image data can be compressed. For two-dimensional data there exist many compression techniques such as JPEG, GIF and the new wavelet based JPEG2000 standard [7]. All schemes above are used or two-dimensional data (images) and while they are excellent for images they might not be that well suited for compression of three-dimensional data such as image stacks.

II. OBJECTIVE

The purpose of this paper is to look at coding schemes based on wavelets for medical volumetric data. The paper deals with the theoretical issues as well as suggests a practically feasible implementation of a coding scheme. A short comparison between two- and three-dimensional coding is also included. Another goal is to implement highly scalable image compression based on SPIHT. DWT of an image leads to its representation in different scales in the form of Spatial Orientation Trees (SOT). The coefficients in SOT of an image DWT are encoded using embedded zero-tree coding which is proved to be the best image compression technique. After the transform, the lowest frequency coefficients concentrate most of the energy of the transformed image. The high frequency coefficients of different scales and orientations indicate the strong self similarity among themselves. These properties are exploited in SPIHT. After EBCOT[5] in JPEG2000, SPIHT a sophisticated coding technique belongs to next generation of encoders for wavelet transformed images.

The basic SPIHT uses dynamic data structures for exploiting self-similarities mentioned above[6]. These dynamic data structures impose practical limitation on hardware implementation of SPIHT, unlike software implementation where dynamic data structures can be implemented conveniently using linked lists. Hence we have modified the basic SPIHT algorithm to overcome dynamic allocation problem. The FPGA implementation of modified version has resulted in significant optimization in memory requirements and speed. The SPIHT algorithm can be applied to both grey-scale and colored images. SPIHT displays exceptional characteristics over several properties like good image quality, fast coding and decoding, a fully progressive bit stream, application in lossless compression, error protection and ability to code for exact bit rate.

III. BRIEF OVERVIEW OF SPHIT ALGORITHM

The SPIHT algorithm, developed by Said and Pearlman in 1996 is a fast and efficient image compression algorithm works by testing ordered wavelet coefficients for significance in a decreasing bit plane order, and quantizing only the significant coefficients. The high coding efficiency obtained by this algorithm is due to a group testing of the coefficients of a wavelet tree. The SPIHT (Set Partitioning in Hierarchical Trees) algorithm is a refined version of EZW algorithm. It can perform better at higher compression ratios for a wide variety of images than EZW. The algorithm uses a partitioning of the trees in a manner that tends to keep insignificant coefficients together in larger subsets. The SPIHT algorithm groups the wavelet coefficients and trees into sets based on their significance information. The encoding algorithm consists of two main stages, sorting and refinement. In the sorting stage, the threshold for significance is set as 2n, where n is the bit level, and its initial value is determined by the number of bits required to represent the wavelet coefficient with the maximum absolute value. Significance for trees is obtained by checking all the member detail coefficients

III. DISCRETE WAVELET TRANSFORM

Calculating wavelet coefficients at every possible scale is a fair amount of work, and it generates an awful lot of data. If the scales and positions are chosen based on powers of two, the so-called dyadic scales and positions, then calculating wavelet coefficients are efficient and just as accurate. This is obtained from discrete wavelet transform (DWT).

A.SINGLE-STAGE FILTERING

For many signals, the low-frequency content is the most important part. It is the identity of the signal. In wavelet analysis, the approximations and details are obtained after filtering. The approximations are the high-scale, low frequency components of the signal. The details are the low-scale, high frequency components. The filtering process is represented in Figure1.

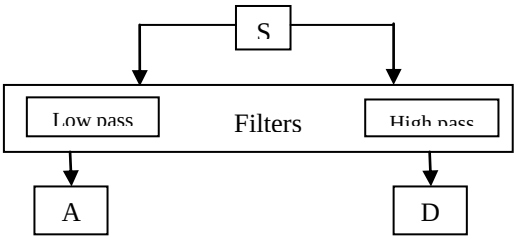

Figure 1: Single stage filtering

The original signal, S, passes through two complementary filters and emerges as two signals. Unfortunately, it may result in doubling of samples and hence to avoid this, down sampling is introduced. The process on the right, which includes down sampling, produces DWT coefficients. The schematic diagram with real signals inserted is as shown in Figure 2.

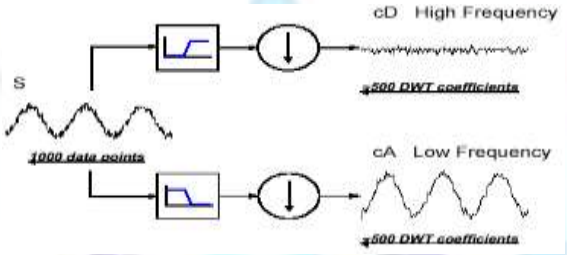

Figure 2: Decomposition and decimation.

B.MULTIPLE-LEVEL DECOMPOSITION

The decomposition process can be iterated, with successive approximations being decomposed in turn, so that one signal is broken down into many lower resolution components. This is called the wavelet decomposition tree and is depicted as in Figure 3.

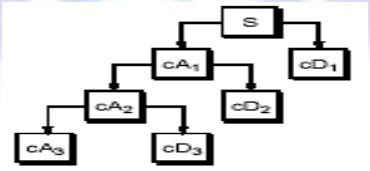

Figure 3: Multilevel decomposition.

C.WAVELET RECONSTRUCTION

The reconstruction of the image is achieved by the inverse discrete wavelet transform (IDWT). The values are first up sampled and then passed to the filters. This is represented as shown in Figure 4.

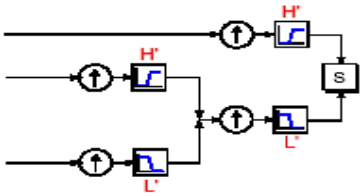

Figure 4: Wavelet Reconstruction

The wavelet analysis involves filtering and down sampling, whereas the wavelet reconstruction process consists of up sampling and filtering. Up sampling is the process of lengthening a signal component by inserting zeros between samples as shown in Figure 5.

ISSN 22773061

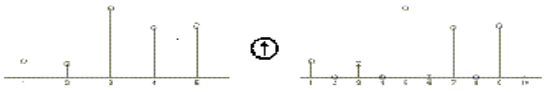

Figure 5: Reconstruction using up sampling.

D. RECONSTRUCTING APPROXIMATIONS AND DETAILS

It is possible to reconstruct the original signal from the coefficients of the approximations and details. The process yields a reconstructed approximation which has the same length as the original signal. The reconstructed details and approximations are true constituents of the original signal. Since details and approximations are produced by down sampling and are only half the length of the original signal they cannot be directly combined to reproduce the signal. It is necessary to reconstruct the approximations and details before combining them. The reconstructed signal is schematically represented as in Figure 6.

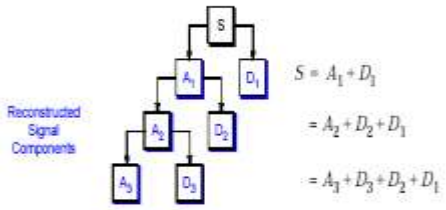

Figure 6: Reconstructed signal components.

E.1-D WAVELET TRANSFORM

The generic form for a one-dimensional (1-D) wavelet transform is shown in Figure 7.Here a signal is passed through a low pass and high pass filter, h and g, respectively, then down sampled by a factor of two, constituting one level of transform.

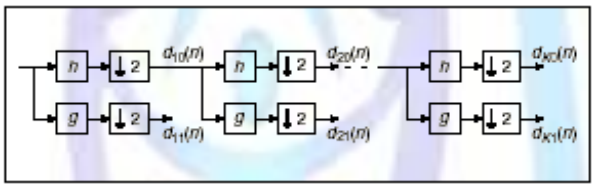

Figure7: 1D Wavelet Decomposition.

Repeating the filtering and decimation process on the lowpass branch outputs make multiple levels or “scales” of the wavelet transform only. The process is typically carried out for a finite number of levels K, and the resulting coefficients are called wavelet coefficients.The one-dimensional forward wavelet transform is defined by a pair of filters s and t that are convolved with the data at either the even or odd locations. The filters s and t used for the forward transform are called analysis filters.

$$l_{i=} \sum_{j=-n_l}^{n_l} S_j X_{2i+j} \quad \text{and} \quad h_{i=} \sum_{j=-n_h}^{n_h} t_j X_{2i+1+j}$$

Although l and h are two separate output streams, together they have the same total number of coefficients as the original data. The output stream l, which is commonly referred to as the low-pass data may then have the identical process applied again repeatedly. The other output stream, h (or high-pass data), generally remains untouched. The inverse process expands the two separate low- and high-pass data streams by inserting zeros between every other sample, convolves the resulting data streams with two new synthesis filters s’ and t’, and adds them together to regenerate the original double size data stream.

$$Y_i = \sum_{j=-n_h}^{n_h} t'_j l'_{i+j} + \sum_{j=-n_h}^{n_h} s'_j h'_{i+j}$$

$$\text{where } l'_{2i} = l_i, l'_{2i+1} = 0, h'_{2i+1} = h_i, h'_{2i} = 0$$

To meet the definition of a wavelet transform, the analysis and synthesis filters s, t, s’ and t’ must be chosen so that the inverse transform perfectly reconstructs the original data. Since the wavelet transform maintains the same number of coefficients as the original data, the transform itself does not provide any compression. However, the structure provided by the transform and the expected values of the coefficients give a form that is much more amenable to compression than the original data. Since the filters s, t, s’ and t’ are chosen to be perfectly invertible, the wavelet transform itself is lossless. Later application of the quantization step will cause some data loss and can be used to control the degree of compression. The forward wavelet-based transform uses a 1-D subband decomposition process; here a 1-D set of samples is converted into the low-pass subband (L1) and high-pass subband (H1). The low-pass subband represents a down sampled low-resolution version of the original image. The high-pass

subband represents residual information of the original image, needed for the perfect reconstruction of the original image from the low-pass subband.

F. 2-D TRANSFORM HEIRARCHY

The 1-D wavelet transform can be extended to a two-dimensional (2-D) wavelet transform using separable wavelet filters. With separable filters the 2-D transform can be computed by applying a 1-D transform to all the rows of the input, and then repeating on all of the columns.

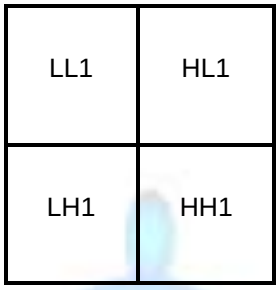

Figure 8: Subband Labeling Scheme for a one level, 2-D Wavelet Transform.

The original image of a one-level (K=1), 2-D wavelet transform, with corresponding notation is shown in Fig.

IV. BACKGROUND

The backbone of the architecture is a single-issue, 3-stage pipeline with 32 general-purpose registers (does not have any address registers like the Motorola 68000 Processor), an Arithmetic Logic Unit (ALU), a shift unit, and two levels of interrupt. This basic design can then be configured with more advanced features to tailor to the exact needs of the target embedded application such as: barrel shifter, divider, multiplier, single preCIsIon floating-point unit (FPU), instruction and data caches, exception handling, debug logic, Fast Simplex Link (FSL) interfaces and others[13].

This flexibility allows the user to balance the required performance of the target application against the logic area cost of the soft processor MicroBlaze also supports reset, interrupt, user exception, and break hardware exceptions. For interrupts, MicroBlaze supports only one external interrupt source (connecting to the Interrupt input port) [13]. If multiple interrupts are needed, an interrupt controller must be used to handle multiple interrupt requests to MicroBlaze shown in figure l0.An interrupt controller is available for use with the Xilinx Embedded Development Kit (EDK) software tools. The processor will only react to interrupts if the Interrupt Enable (IE) bit in the Machine Status Register (MSR) is set to 1. On an interrupt the instruction in the execution stage will complete, while the instruction in the decode stage is replaced by a branch to the interrupt vector (address Ox 10). The interrupt return address (the PC associated with the instruction in the decode stage at the time of the interrupt) is automatically loaded into general-purpose register. In addition, the processor also disables future interrupts by clearing the IE bit in the MSR. The IE bit is automatically set again when executing the RTlD instruction. Writing software to control the MicroBlaze processor must be done in C/C++ language. Using C/C++ is the preferred method by most people and is the format that the Xilinx Embedded Development Kit (EDK) software tools accept. The EDK tools have built in C/C++ compilers to generate the necessary machine code for the MicroBlaze processor.

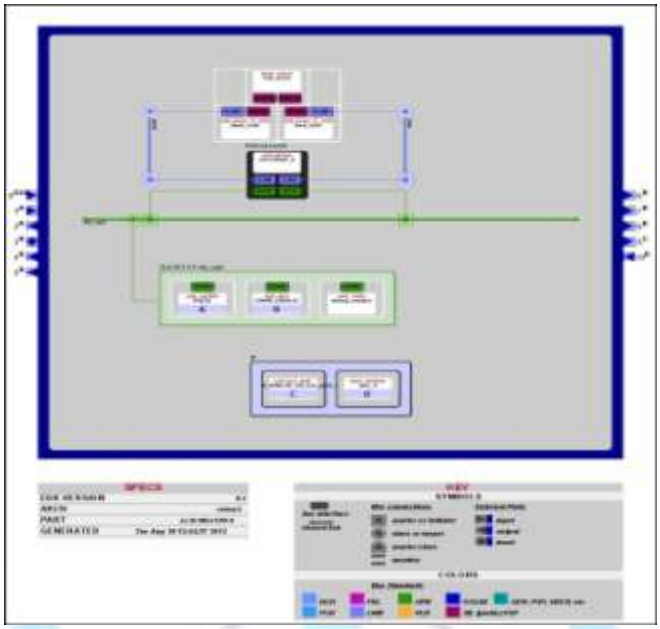

Figure10: Microblaze Architecture Block Diagram.

Due to the advancement in the fabrication technology and the increase in the density of logic blocks on FPGA, the use of FPGA is not limited anymore to debugging and prototyping digital electronic circuits. Due to the enormous parallelism achievable on FPGA and the increasing density of logic blocks, it is being used now as a replacement to ASIC solutions in a few applications where the time to market is critical and also entire embedded processor systems are implemented on these devices with soft core processors embedded in the system. With the advancement of Field Programmable Gate Arrays (FPGAs), like addition of substantial amounts of memory, a new trend has emerged in the design community to implement the microprocessors on the FPGAs. These kind of processors implemented on a reconfigurable fabric are called soft-processors or soft cores as the design of the microprocessor is available in the form of software bit stream which can be downloaded on FPGA by the user. The users have the choice of selecting the resources on the processor and the memory hierarchy. Soft cores are designed to meet minimum performance specifications over a range of technology implementations, even though core performance varies across technologies. Soft cores are technology independent and require only simulation and timing verification after synthesized to a target technology. This reduces the design cycle development time by a major factor as compared to the development cycle for a hard core processor and has the advantage of customizing the soft core design for a specific application. Currently there are a number of soft cores available in the markets that are developed by giants in the field of reconfigurable devices like Xilinx. Xilinx has their own architecture named MicroBlaze in this arena and they have also ported the popular PowerPC architecture for use in embedded systems. These soft cores are available in the form of synthesized HDL modules or gate level net lists. System designers can embed these cores into their designs and optionally add peripherals to the core.

V. EXPERIMENTAL SETUP

The Xilinx Platform Studio (XPS) is the development environment or GUI used for designing the hardware portion of your embedded processor system. Embedded Development Kit Xilinx Embedded Development Kit (EDK) is an integrated software tool suite for developing embedded systems with Xilinx MicroBlaze and PowerPC CPUs. EDK includes a variety of tools and applications to assist the designer to develop an embedded system right from the hardware creation to final implementation of the system on an FPGA. System design consists of the creation of the hardware and software components of the embedded processor system and the creation of a verification component is optional. A typical embedded system design project involves: hardware platform creation, hardware platform verification (simulation), software platform creation, software application creation, and software verification. Base System Builder is the wizard that is used to automatically generate a hardware platform according to the user specifications that is defined by the MHS (Microprocessor Hardware Specification) file. The MHS file defines the system architecture, peripherals and embedded processors]. The Platform Generation tool creates the hardware platform using the MHS file as input. The software platform is defamed by MSS (Microprocessor Software Specification) file which defines driver and library customization parameters for peripherals, processor customization parameters, standard 110 devices, interrupt handler routines, and other software related routines. The MSS file is an input to the Library Generator tool for customization of drivers, libraries and interrupts handlers.

ISSN 22773061

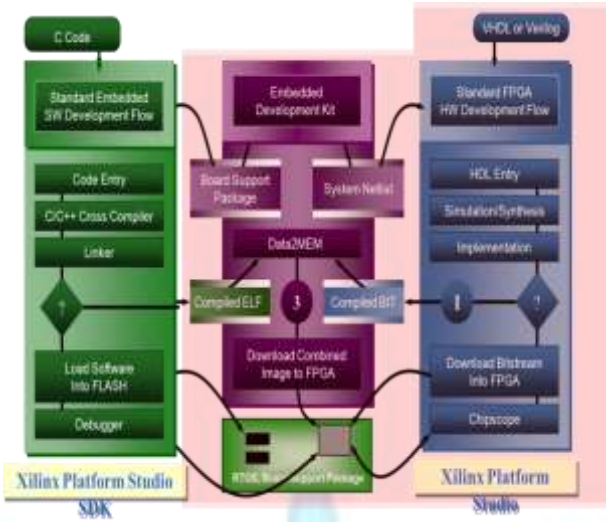

Figure11: Embedded Development Kit Design Flow.

The creation of the verification platform is optional and is based on the hardware platform. The MHS file is taken as an input by the Simgen tool to create simulation files for a specific simulator. Three types of simulation models can be generated by the Simgen tool: behavioral, structural and timing models. Some other useful tools available in EDK are Platform Studio which provides the GUI for creating the MHS and MSS files. Create / Import IP Wizard which allows the creation of the designer's own peripheral and import them into EDK projects. Platform Generator customizes and generates the processor system in the form of hardware netlists. Library Generator tool configures libraries, device drivers, file systems and interrupt handlers for embedded processor system. Bit stream initializer tool initializes the instruction memory of processors on the FPGA. GNU Compiler tools are used for compiling and linking application executables for each processor in the system . There are two options available for debugging the application created using EDK namely: Xilinx Microprocessor Debug (XMD) for debugging the application software using a Microprocessor Debug Module (MDM) in the embedded processor system, and Software Debugger that invokes the software debugger corresponding to the compiler being used for the processor. C. Software Development Kit Xilinx Platform Studio Software Development Kit (SDK) is an integrated development environment, complimentary to XPS, that is used for C/C++ embedded software application creation and verification. SDK is built on the Eclipse open source framework. Soft Development Kit (SDK) is a suite of tools that enables you to design a software application for selected Soft IP Cores in the Xilinx Embedded Development Kit (EDK).The software application can be written in a "C or C++" then the complete embedded processor system for user application will be completed, else debug & download the bit file into FPGA. Then FPGA behaves like processor implemented on it in a Xilinx Field Programmable Gate Array (FPGA) device.

VI. PICTORIAL RESULTS:

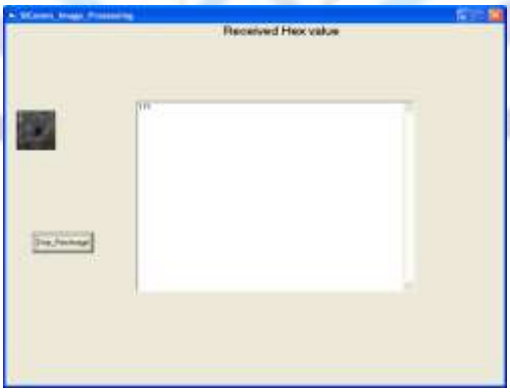

Figure12: Image before compression.

ISSN 22773061

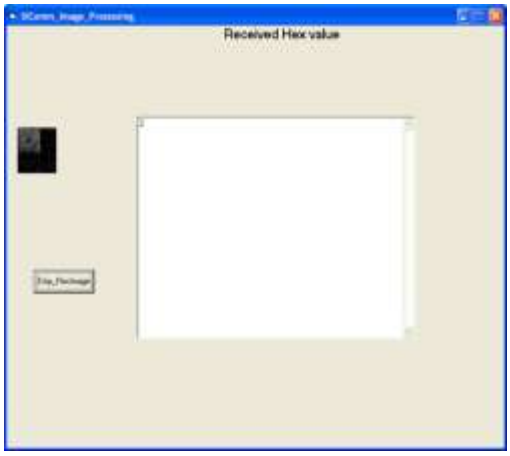

Figure13: Images after compression.

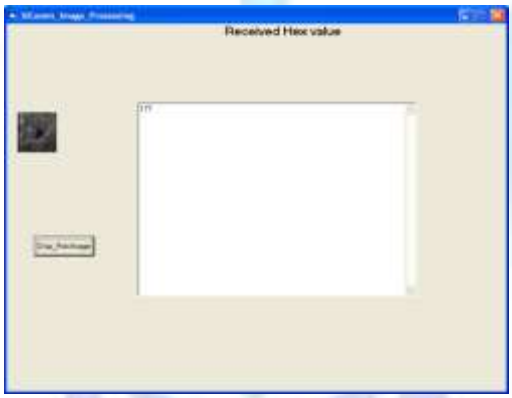

Figure14: Image after reconstruction

VII. TABULATION RESULTS

The Algorithm is implemented in Microblaze Processor and the results are furnished in the tabulation below.

Table-1

Resource Type	Used	Available	Percent
Slices	712	1920	37
Slice Flip Flops	898	3840	23
4 input LUTs	1377	3840	35
IOs	2296	NA	NA
bonded IOBs	0	97	0
MULT18X18s	3	12	25

VIII. CONCLUSION

In this paper we have developed a technique for line based wavelet transforms. We pointed out that this transform can be assigned to the encoder or the decoder and that it can hold compressed data. We provided an analysis for the case where both encoder and decoder are symmetric in terms of memory needs and complexity. We described highly scalable SPHIT coding algorithm that can work with very low memory in combination with the line-based transform, and showed that its performance can be competitive with state of the art image coders, at a fraction of their memory utilization. To the best of our knowledge, our work is the first to propose a detailed implementation of a low memory wavelet image coder. It offers a significant advantage by making a wavelet coder attractive both in terms of speed and memory needs.

Further improvements of our system especially in terms of speed can be achieved by introducing a lattice factorization of the wavelet kernel or by using the lifting steps. This will reduce the computational complexity and complement the memory reductions mentioned in this work.

IX. REFERENCES

1. N.SKODRAS, T.Evrahmi “JPEG2000 image coding system theory and application”.
2. T.Acharya and Ping-Sing Tsai, JPEG2000 Standard for Image Compression Concepts, algorithms and VLSI Architectures. John Wiley & Sons press,2005.
3. E.farzad,C.Matthieu,and W.stefan,“JPEG versus JPEG2000:an objective comparison of image encoding quality,” SPIE Proceedings ,vol 5558,pp.300-308,2004.
4. J.P. Agrawal, & DR.RITU Vijay, “WAVELET COMPRESSION OF CT MEDICAL IMAGES”, Volume 1 Issue3 pp 045 -051 June 2012.
5. David Taubman, “High Performance Scalable Image Compression with EBCOT” IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 9, NO. 7, JULY 2000.
6. Jun-Ren Ding and Jar-Ferr Yang, “A SIMPLIFIED SPIHT ALGORITHM”, Journal of the Chinese Institute of Engineers, Vol. 31, No. 4, pp. 715-719 (2008).
7. Sadashivappa , Mahesh Jayakar , K.V.S Anand Babu , Dr. Srinivas K “Color Image Compression using SPIHT Algorithm”, International Journal of Computer Applications (0975 – 8887) Volume 16– No.7, February 2011
8. Xilinx Inc., PicoBlaze 8-bit Embedded Microcontroller UserGuide. http://www.xilinx.comlsupport/documentation/ userJluides/ugl29.
9. Digilent Inc., Digilent Nexys2 Board Reference Manual
10. Xilinx. Inc., Platform Specification Format Reference Manual, Embedded Development Kit EDK 9.2i.
11. Marc Antonini, Michel Barlaud, “Image Coding Using Wavelet Transform,” IEEE Transactions on Image Processing, 1992.
12. David Taubman, Michael Marcellin,“JPEG2000: Standard for Interactive Imaging,” IEEE Proc., Vol 90, No. 8, 2002.
13. Xilinx Inc. MicroBlaze Reference Manual, version 10.1.
14. Xilinx Inc. Xilinx ISE and Xilinx EDK tools.
15. Spartan-3 Starter Kit Board User Guide, Xilinx, Inc.
16. Embedded System Tools Reference Manual, Xilinx, Inc
17. Spartan-3 FPGA Family: Complete Data Sheet
18. Platform Studio User Guide, Xilinx, Inc.
19. Xilinx. Inc., Platform Specification Format Reference Manual, Embedded Development Kit EDK 9.2i
20. Xilinx, Embedded System Example, XAPP433, version 2.2, 2006.
21. Forchheimer R. (1999), Image coding and data compression, Linköping:Department of electrical engineering at Linköpings University.
22. Chui C. K. (1992), An introduction to wavelets, Boston, Academic Press, ISBN 0121745848
23. Sayood K. (2000), Introduction to data compression, Morgan Kaufmann Publrs,US, 2 revised edition, ISBN 1558605584.
24. Daubechies I., Barlaud M., Antonini M., Mathieu P. (1992), “Image coding using wavelet transform” in IEEE Transactions on image processing, Vol. 1, No. 2, pp.205-220.
25. Brislawn C. M.. (1996), “Classification of nonexapnsive symmetric extension transforms for multi rate filter banks” in Applied and computational harmonic analysis, Vol. 3, pp. 337-357.
26. Azpiroz-Leehan J., Lerallut J.-F. (2000), “Selection of biorthogonal filters for image compression of MR images using wavelet packets” in Medcial engineering & physics, Vol. 22, pp. 225-243.

X. BIOGRAPHIES

Author:1 M.Haritha, received B.Tech degree in Electronics & Communication Engineering from Madanapalle Institute of Technology and Science, Madanapalle,in 2009 & Worked as Assistant Professor in MITS during 2009 to 2011.She is now M.tech scholar,in VLSI Design,Madina Engineering College,Kadapa,AP.

E-mail: hari.jetc@gmail.com

Author:1 Syed Jahangir Badashah received B.E.degree in Electronics & Communication Engineering from Gulbarga University, in 2002,M.E.in Applied Electronics from Sathyabama University in 2005.He is currently doing research in image processing from Sathyabama University. He is having an experience of 12 years, in the field of teaching, presently working as Assco Professor in the department of ECE, Madina Engg College, Kadapa. He is a life time member of IETE & ISTE.

E-mail: syd_jahangir@yahoo.co.in