Vol 05, Article 11493, November 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

IMPLEMENTATION OF COMPRESSED IMAGE USING DWT ON FPGA

D.BINDU TUSHARA[1], P.A.HARSHA VARDHINI[2], N.RANGANADH[3]
[1,2]Department of ECE, V.I.T.S., Deshmukhi, Hyderabad, INDIA.
[3]Department of ECE, ICFAI, Hyderabad, INDIA.
[1]tushara.dewdrops@gmail.com , [2]pahv19@rediffmail.com, [3]ranganadh.narayanam@gmail.com

ABSTRACT

The Spartan-3 generation FPGAs offers a choice of five platforms, each delivering a unique cost-optimized balance of programmable logic, connectivity, and dedicated hard IP for your low-cost applications. It is logic optimized for applications where logic densities matter more than I/O count and is ideal for logic integration, DSP co-processing and embedded control, requiring significant processing and narrow or few interfaces. This paper presents the method of implementing discrete wavelet transform compression technique using lifting process on FPGA.

Keywords: *FPGA, Image Compression Techniques, DWT, Lifting Process.*

1. INTRODUCTION

FPGAs are programmable semiconductor devices that are based around a matrix of Configurable Logic Blocks (CLBs) connected through programmable interconnects. In contrast Application Specific Integrated Circuits (ASICs), the device is custom built for the particular design. FPGAs can be programmed to the desired application or functionality requirements. Although One Time Programmable (OTP) FPGAs are available, the dominant type are SRAM-based which can be reprogrammed as the design evolves [1]. FPGAs allow designers to change their designs very late in the design cycle even after the end product has been manufactured and deployed in the field. In addition, Xilinx FPGAs allow for field upgrades to be completed remotely, eliminating the costs associated with re-designing or manually updating electronic systems.

The proposed paper presents an appropriate technique for compressing an image and displaying the compressed image with an FPGA. Basic techniques and components are used in this approach to obtain the proposed results. Section 2 describes the image compression methods and Discrete Wavelet Transform (DWT) is described in section 3. Section 4 details the image compression procedure carried out in the present work and the corresponding results. Section 5 concludes the paper.

2. IMAGE COMPRESSION

2.1. Image

An image is a matrix of pixel. Each pixel has the corresponding pixel intensity value. The various operations performed on the image is with respect to these image intensities. Images can be classified as coloured image or gray-scale image or black and white (binary image). The differentiation among these types of images is with respect to the intensity values of the pixel. They differ in the ranges of these pixel intensity values.

(i) The colour image has three intensity values with respect to the three basic colours: red, green and blue. Every pixel can have combination of one or many color values.

(ii) Gray-scale image values are represented from 0-255 where ‘0’ is for representing black and ‘1’ for white. Every pixel has single value.

(iii) Binary image is represented in terms of 1’s and 0’s.

2.2 Image Compression

This work involves compression and its necessity for a DWT using lifting scheme. Compression is a process of reducing the number of data bits necessary for representing information, to properly utilize the available bandwidth and reduce storage Space. Compression techniques carried out can be lossless or lossy. In lossless compression data can be completely recovered after decompression. Recovered data is identical to original. In contrast, in lossy compression data cannot be completely recovered after decompression, some information is lost forever but gives more compression than lossless and discards the insignificant data components. Hence lossless compression is preferred as it in order to have fewer losses while recovering.

3. DISCRETE WAVELET TRANSFORM

One of the lossless compressions that is used in the proposed work is DWT. The DWT has been developed as an efficient DSP tool for signal analysis, image compression, and even video compression. There are many architectures proposed for the implementation of DWT. For the 1-D (one-dimensional) DWT, the architectures can be categorized into the convolution- based, lifting-based, and B-spline-based .The first one is to implement two-channel filter banks directly. The second one is to exploit the relationship of low pass and high pass filters

Vol 05, Article 11493, November 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

for saving multipliers and adders. The third one can reduce the multipliers based on the B-spline factorization. The B-spline-based architectures could provide fewer multipliers while the lifting scheme fails to reduce the complexity.

1-D DWT is used for the operations on the data where data is fetched from the memory modules and using the buffers, they are stored temporarily for the operations. Once the operation is done and completed, the data is stored back in the memory form which the data is fed [2,3]. The 2-D (two-dimensional) module computes the transform for 1D Data. It is responsible for extending the 1D DWT module to images (2D). 2D transform is implemented by two passes of 1D transform - Horizontal Pass (Row wise) and Vertical Pass (Column Wise). The buffers used here is twice the size of picture data.

3.1 Lifting process

The lifting scheme is an alternative method of computing the wavelet coefficients. Advantages of the lifting scheme:

- Requires less computation and less memory.
- Easily produces integer-to-integer wavelet transforms for lossless compression.
- Backward transform is easy to find.
- Linear and adaptive wavelet transform is feasible, and the resulting transform is invertible and reversible.

3.2 Implementation using FPGA

Image compression technique is implemented and programmed it on the SRAM of the FPGA. The dumping is done by converting into bit stream and viewing the output. For dumping the bit stream converted, Xilinx Platform Studio is used which is the main part in performing the objective. In this, selection of corresponding devices on the board is done and then to perform dumping for viewing the compressed image.

4. PROPOSED METHOD AND SIMULATION RESULTS

Various image compression techniques exist and implementation on FPGA has also got several methodologies involved. The method that is proposed basically consists of Xilinx platform studio running on PC instead of Xilinx software for dumping on FPGA. Usually coding is done in VHDL language using Xilinx but in this method, C is used to write appropriate program as programming is done with respect to the SRAM on the board. The image is taken as input from the system in the form of header file consisting of the image pixel matrix.

4.1. Image considered and the method for converting into gray-scale

The image considered is a gray-scale as it becomes easy for performing the implementations on this type. As each image can be represented in terms of a single pixel value, hence it becomes easy for one to perform any transformation. Any type of image can be converted into a gray-scale image and generate the header file consisting of matrix of pixel intensity values.

There are two possibilities for considering the gray-scale image:

(i) Considering the gray-scale image itself from the PC.

(ii) Converting a colour or black and white into gray-scale.

This conversion is done using MATLAB version where resizing of the image is done and then conversion is performed. After performing the conversion, the header file consisting of pixel values is generated.

4.4 Image compression coding

Every image can be segregated into even and odd values. This segregation is performed by arranging the rows and columns of the image matrix and are arranged in the order of even and odd row and following column. Then they are divided into low and high values as shown in Fig 1.

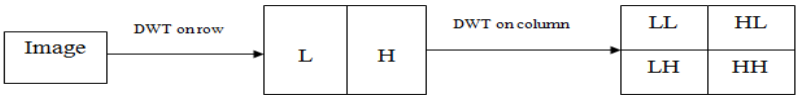

Fig 1. Image Compression with DWT

Hence the above process is carried on for all the levels of compressions that are done. The formulas considered for L,H values are as depicted in Fig 2.

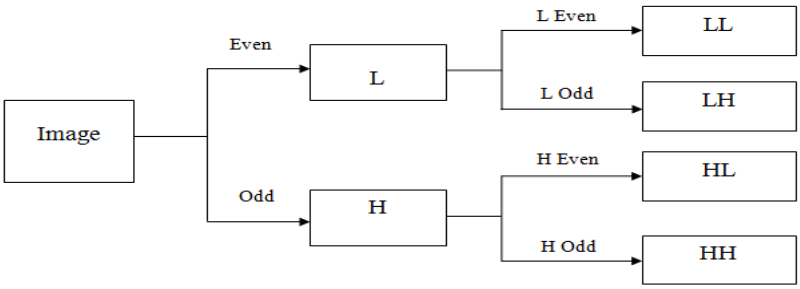

Fig 2. Levels of Compression

Vol 05, Article 11493, November 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

L= even – H/2
H= even – odd
LL = Leven – LH/2
LH= Leven – Lodd
HL = Heven – HH/2
HH = Heven – Hodd
This is carried on for the compression performed. Based on the above formulas the coding is done.

4.5 Visual basic

A Graphical User Interface (GUI) is created using MATLAB with the window showing the buttons for the display of image, pixel values read. It also has the fields indicating for display of input image and output compressed image.

Steps followed for the implementation

The image is taken in the form of a gray-scale image and compression is taken on the pixel intensity values. This compression code is written in C and taken as source file. An appropriate header file is generated showing the pixel intensity values. The source and header files are taken and corresponding bitstream is generated which will be downloaded on to SRAM. This memory is set to the starting address from where the dumping starts. The source PC has Xilinx platform studio running and output PC has visual basic. After successful dumping of the files without any errors, using command prompt output is viewed. This compression is considered as shown in Fig 3.

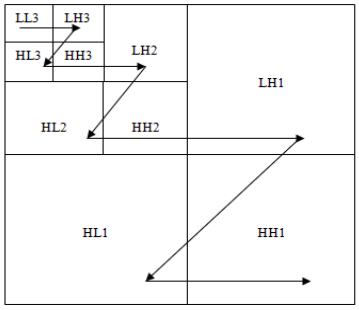

Fig 3. Image Compression procedure

4.7. Simulation Results

The simulation results are shown with respect to the input window, showing the input applied, output window showing the compressed output and then the decompressed output. Each window has two fields for the display of image and reading of the pixel values. Input window is given in Fig 4. Fig 5 depicts the corresponding compressed output and the decompressed output after performing reverse DWT is shown in Fig 6.

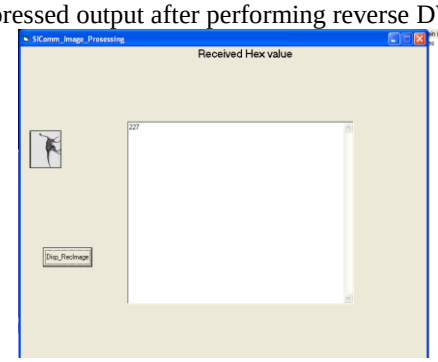

Fig 4. Input image in GUI

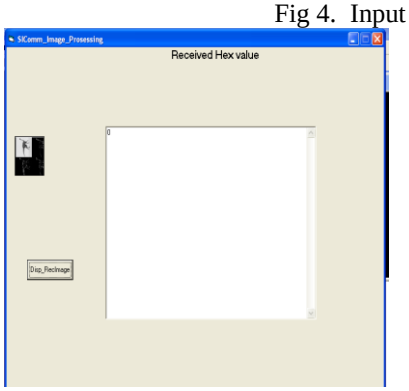

Fig 5. Compressed image (DWT output)

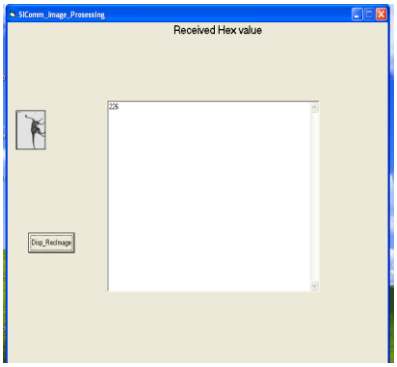

Fig 6. Reverse DWT

Vol 05, Article 11493, November 2014
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

CONCLUSION

This paper proposes the implementation of image compression on FPGA using DWT. Several lossless compression techniques are implemented on an image in order to recover the original image without any loss. DWT lifting process is one of the method among several proposed methods which has several advantages compared to the other techniques. Efficient implementation is carried out in this proposed paper by selecting an efficient method for compression, software used for dumping onto FPGA. This work is continued by extending the design to view the output on another PC using interfacing devices like Zigbee.

REFERENCES

[1] "SPIHT Image Compression on FPGAs" *Circuits and Systems for Video Technology, IEEE Transactions*, Volume 15, Issue 9 , pp 1138 – 1147, Sept 2005.
[2] "Memory reduction methodology for distributed-algorithm –based DWT/IDWT Exploiting data summary" *IEEE transactions on circuits and systems-II*, volume 56, NO. 4, April 2009.
[3] " An efficient VLSI architecture and FPGA implementation of high speed and low power 2-D DWT for (7,9) wavelet transform" *IJCSNS International journal of computer science and security*, volume 9 No. 3, March 2009.
[4]M. Vishwanath, R. M. Owens, and M. 1. Irwin, "VLSI Architectures for the Discrete Wavelet Transform," IEEE Trans. Circuits And Systems II, vol. 42, no. 5, pp. 305-316, May. 1995.
[5] T. Chen and R. W. Hong, "Application of partition-based median type filters for suppressing noise in images," IEEE Transactions on Image Processing, Vol. 10, 2001, pp. 829-836.
[6] D. Ziou & S. Tabbone, "Edge Detection Techniques An Overview", 1998.
[7] Maar, D., Hildreth E., "Theory of edge detection", Proceedings Royal Soc. London, vol. 207, 187-217, 1980.