Proc. of Microelectronics & Nanotechnology (2014)
Received 2 July 2013; accepted 7 October 2013

FPGA-based Implementation of Daubechies Wavelet Transform with Distributed Arithmetic (DA)

Noor Huda Ja'afar[1,*], Afandi Ahmad[1,2], Abbes Amira[3]

[1]VLSI Architecture and Systems Design Research Laboratory (VASYD)
Microelectronic and Nanotechnology – Shamsuddin Research Centre (MiNT-SRC)
Universiti Tun Hussein Onn Malaysia (UTHM)
P. O. Box 101, 86400 Batu Pahat, Johor, MALAYSIA.

[2]Department of Computer Engineering
Faculty of Electrical and Electronic Engineering
Universiti Tun Hussein Onn Malaysia (UTHM)
P. O. Box 101, 86400 Batu Pahat, Johor, MALAYSIA.

[3]School of Computing, Room F301
University of West of Scotland, Paisley Campus
Paisley, PA1 2BE, Scotland, UNITED KINGDOM.

Abstract: This study presents the design and implementation of efficient architectures for Daubechies 4-tap (Daub4) and 6-tap (Daub6) with distributed arithmetic (DA) design strategy on field programmable gate array (FPGA). Both architectures are targeted to be deployed for adaptive transformation process of three-dimensional (3-D) medical image compression applications. The proposed architectures are synthesised using very-high-speed integrated circuit (VHSIC) of hardware description language (VHDL) using Xilinx integrated software development (Xilinx ISE) 14.2, LabVIEW FPGA and implemented on sbRIO-9632 with Spartan-3 (XC3S2000) device. Various 3-D medical modalities including magnetic resonance imaging (MRI), positron emission tomography (PET) and computed tomography (CT) have been used as an input for the transformation image process. Performance analysis in terms of area, power consumption, maximum frequency, latency and throughput is presented and reveals significant achievements.

Keywords: Daubechies, Distributed Arithmetic (DA), 3-D medical image compression, FPGA

1. Introduction

In the high-tech world, medical imaging is very important to diagnose and analyse illness inside human body without having a surgery. Various types of medical imaging modalities [1], such as computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), ultrasound (US) and X-ray are being used to assist diagnosis and track diseases. The increasing number of patients annually has continuously growth the amount of medical imaging data generated and directly causes a demand for data storage. The fact of this issue has been convinced by AT&T Medical Imaging and Information Management:
"Medical image archives are increasing by 20-40 percent each year. It is projected that by 2012 there will be 1 billion medical images stored in the U.S." [2]

The more widespread use of three-dimensional (3-D) imaging modalities, have generate a massive amount of volumetric data. Therefore, an efficient transmission speed and long-lasting data storage are often very profitable to store and access the images to make diagnosis based on the type of examination. This exchanging data requires both efficient storage and transmission of data through high-bandwidth digital communication lines besides being crucial importance topic [1], [3].

Reconfigurable hardware in the form of field programmable gate arrays (FPGAs) appears as viable system building block in the construction of high-performance systems. The capability to develop a programmable circuit architecture with the flexibility of computational, memory, speed and power requirement, FPGAs seem an ideal candidate to be propose as a hardware technology for prototype a simple, moderate and complex applications. Moreover, other advantages offered by FPGAs are massive parallelism capabilities, multimillion gate counts and special low-power packages [4], [5].

The aim of this paper is to develop an efficient reconfigurable architecture for Daubechies wavelet transform using distributed arithmetic (DA) design strategy. An evaluation of these architectures in terms of area, power consumption, maximum frequency and latency are also carried out. Finally, this research is expected to propose a novel architecture of 3-D discrete

wavelet transform (DWT) using various wavelet filters and different design strategies that can be further applied as an intellectual property (IP) core for compression systems specifically in telemedicine applications.

The rest of the paper is organised as follows. An overview of the related work is given in Section 2. Section 3 explains the mathematical background for Daubechies wavelet transform. Section 4 exposes the proposed 3-D DWT architectures. Experimental results and an analysis of the area, power consumption, maximum frequency, latency as well as throughput are presented in Section 5. Section 6 discusses the resulting outcomes. Finally, concluding remarks and further potential ideas to be explored are given in Section 7.

2. Related Works

The impact and rapid progress in computerised medical image has lead medical imaging into one of the most important sub-fields in scientific imaging. Indirectly, the field of medical image compression has become an enticing topic among researchers to cope with data storage limitation and massive data generated problems. In general, the fundamental unit on computer is represented in a number of bits. For instance, an image containing 640×480 pixels (12-bit gray scale) needs more than 3 Megabits per second of storage. Therefore, to transmit this image via conventional phone lines with speed range of 56 Kilobits per second, it consumes more than 1 second. For this reason, image compression is important to provide an efficient data storage and data transmission [6].

The purpose of image compression is to reduce the size and blocking artifact of original image without degrading the quality of the image. Because of that, the development of efficient image compression technique becomes the most challenging matter. Moreover, compressing medical image is more challenging compared to non-medical image. This is because for medical images, the compression algorithms are complex and it should always be stored in lossless format eventhough sometimes lossy format is acceptable [7], [8]. In addition, medical images are extremely rich with information contents. The significant high quality information of the images is very important to permanently store for disease diagnoses or treatment.

Most of the existing works carry out an algorithms development and optimisation [4] and [9-11] without having a hardware implementation. In the following, an overview of previous works is described and the first two descriptions [12], [13] will illustrate the contributions on the hardware implementation of 3-D medical image compression, whilst the others are concern on algorithm development and optimisation [4], [9], [14] and [15].

An implementation of 3-D Haar wavelet transforms (HWT) using dynamic partial reconfigurable (DPR) for compression system is presented in [12]. This paper is concern on the transformation algorithm based on DWT with aim to speed up the processing of large medical volumes. For the 3-D HWT computation, the blocks are split into three 1-D HWT cascaded together with transpose module in between. The first 1-D HWT is applied along the rows of the matrix followed by another 1-D HWT along the columns of the transformed matrix. While the third 1-D HWT is applied to the corresponding pixels in each *N* sub-images. To evaluate the compression system, CT and MRI medical image modalities are used. The results briefly shows that the implementation of 3-D HWT with DPR provides better saving of area and power consumption while the value of maximum frequency is much higher than without using DPR. In summary, this paper provides a brief explanation of the 3-D HWT and matrix transpose computation.

Lee *et al.* [13] have presents a 3-D approach towards the implementation of a bio-medical image capture, compression and encryption system. A stacked chip with through silicon via (TSV) technology is selected to implement the 3-DWT architecture. Multiple silicon dies with system functions of complementary metal-oxide semiconductor (CMOS) image sensor layer, memory layer, 3-D DWT, 3-D advanced encryption standard (AES) blocks and radio frequency (RF) transmitter layer are get involved in the implementation process. A comparison in terms of peak signal-to-noise ratio (PSNR) values are carried out, whilst other parameter evaluations are not recorded and analysed in this work. Henceforth, the multiple silicon dies are fabricated to perform specific tasks. This type of implementation has no flexibility to reconfigure the tasks of the silicon chip.

An area-efficient high-throughput 3-D DWT architecture based on Distributed Arithmetic (DA) is presented in [4]. Daubechies 9/7 wavelet coefficients are used for the 3-D DWT processor. The proposed DA architecture used low-pass and high-pass filters with bit-serial implementation. In addition, the bit-serial architecture requires fewer components such as an adder, look up table (LUT), accumulator and 411 kbits memory. The proposed architecture is synthesised for Xilinx Virtex-E FPGAs. Eventhough the result shows the low hardware cost and high speed-area efficiency, the implementation on real FPGAs are not reported.

An evaluation of compressed medical images using wavelet transform has been carried out in [9] by Ghrare *et al*. This work is mainly on medical image compression evaluations with three different medical image modalities which are MRI, CT and X-ray. The compression and decompression processes are based on the DWT method. As other works, an objective test is conducted by calculate the PSNR value. In addition, to evaluate the quality of the reconstructed images, a subjective measurement is also carried out using mean opinion score (MOS) indicator. In a nutshell, both objective and subjective tests are important to measure the quality of the compressed medical images with different compression levels.

Another issue on the medical image compression is presented in [14] A wavelet-based medical image compression using embedded zero tree wavelet (EZW) is proposed. Haar and Daubechies wavelet filters are selected due to its special properties of symmetry and simple calculation respectively. The selected wavelet

filters are used to compress the MRI, X-ray and ultrasound images. Experimentally, the results observed that Daubechies 4 (db4) and Daubechies 6 (db6) give good performances for MRI image and X-ray image respectively while Haar wavelet suitable for ultrasound image. Shortened the contents, this paper gives an overview of wavelet filters properties for compression application and also give a shadow for the wavelet filters choices that deals with medical images.

An approaches for 3-D HWT based on image compression system is described by Montgomery *et al.* in [15]. The proposed design methodology for 3-D HWT as well as hard and soft thresholding algorithms are briefly explained in this paper. As an important part in 3-D image compression, the transforms for 3-D HWT are divided into three 1-D HWT blocks with two matrix transpose modules. Firstly, the 1-D HWT is applied to each row of the data set and give results of an average and detail coefficients. Then, the data set of the average and detail coefficients are processed by another 1-D HWT for each row and lastly the 1-D HWT is applied for all slices of the data set. The final results will give a set of detail coefficients with single average coefficient. In brief, although there is no hardware implementation discussed in this paper, interestingly, it provides a clear description of 3-D HWT algorithms for medical image compression system.

From the previous works that have been discussed, it shows that most of researchers are focusing on the algorithms development and optimisation with objective evaluations. On the other side, a details description about the transformation process and architecture are not briefly described and explained. Thus, an efficient implementation of 3-D transform for medical image compression applications with both objective and subjective evaluations is of significant importance.

3. Mathematical Background

Each of wavelet transform has their own algorithms includes the scaling and wavelet functions. The only different between them is the way that the scaling signals and wavelets are defined [4]. It is importat to understand the flow diagram of the wavelet transform computations for apply them in the proposed architectures. Therefore, the mathematical background for Daubechies wavelet transform which specific for Daubechies 4-tap (Daub4) and 6-tap (Daub6) are presented in the following subsections.

In addition, the basic algorithm computation approached by DA is to replace the multipliers which occupy large areas by LUTs. Moreover, DA design technique relies on the fact that the filter coefficients are known. Therefore, the pre-computed sums of two vectors, $c[n]x[n]$ can be stored on the FPGA LUTs, indirectly offers a reduction for the hardware resources. Consider the following dot product y of A_k and X_k shown in equation (1), where A_k is the constant coefficient filter and X_k represents the input sample vector:

$$y = \sum_{k=0}^{M-1} A_k X_k \tag{1}$$

The input sample vector X_k can be represented in N-bit 2's complement notation. Thus, the equation (1) is rearranging to equation (2).

$$y = \sum_{k=0}^{M-1} A_k \left[-b_{k(N-1)} 2^{N-1} + \sum_{n=0}^{N-2} b_{kn} 2^N \right] \tag{2}$$

$$y = -\sum_{k=0}^{M-1} A_k b_{k(N-1)} 2^{N-1} + \sum_{n=0}^{N-2} \left[\sum_{k=0}^{M-1} A_k b_{kn} \right] 2^N \tag{3}$$

where $b_{kn(N-1)}$ is the sign bit of the input sample and b_{kn} is the nth bit of input sample. Expanding equation (2) gives equation (3), where the value of b_{ki} is either 1 or 0. Thus, the multiplication process of $A_k b_{kn}$ can have the value either A_k or 0. Furthermore, all the possible values of the dot product $A_k b_{kn}$ can be pre-computed and stored in the LUTs. On the other side, the input samples are fed to the cascade shift register to addressing the LUT contents.

3.1 Daubechies 4-tap (Daub4) algorithm

The Daub4 wavelet is the simplest wavelet among the Daubechies wavelet families. Generally, Daub4 have four scaling signals and wavelets coefficients as given in equation (4) and (5) respectively.

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \tag{4}$$

$$g_0 = h_3, g_1 = -h_2, g_2 = h_1, g_3 = -h_0 \tag{5}$$

The 1-level Daub4 scaling signals and wavelets can be defined as follows:

$$\begin{pmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 & & \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 & & \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & & \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \end{pmatrix} \tag{6}$$

The scaling and wavelet functions are calculated by taking the inner product of the coefficients and data input values. In the last iteration, data input of s[N] and s[N+1] are not exist (they are beyond the end of the array) and cause the edge problem. To handle this edge problem, the data set is treated as it is periodic, where the last two values of the input vector is shifted to the beginning of the input data sequence.

Theoretically, for N-tap filter a 2^N word LUT is required. Therefore, the memory space is expended directly with the increasing number of the N-bits input data. Lets the pre-computed values are represented in 8-bits, thus these known Daub4 filter coefficients requires $8 \times 2^4 = 128$ bit of memory. In fact, the Daub4 wavelet filter is includes in a long filters category, in which a large memory size is required to store the pre-computed values. These long filter cases will indirectly affects the hardware performances since more N-words of the possible pre-computed values are calculated. To avoid the

using of large memory size, the LUT are partitions into two small LUTs as illustrated in Table 1 and 2.

Table 1 The first LUT for Daub4 wavelet filter.

Address	Data	Address	Data
0000	0	1000	h_4
0001	h_1	1001	$h_1 + h_4$
0010	h_2	1010	h_2+h_4
0011	$h_1+ h_2$	1011	$h_1+h_2+h_4$
0100	h_3	1100	$h_3+ h_4$
0101	$h_1+ h_3$	1101	$h_1+h_3+h_4$
0110	h_2+h_3	1110	$h_2+h_3+h_4$
0111	$h_1+ h_2+h_3$	1111	$h_1+h_2+h_3+h_4$

Table 2 The second LUT for Daub4 wavelet filter.

Address	Data	Address	Data
0000	0	1000	g_4
0001	g_1	1001	$g_1+ g_4$
0010	g_2	1010	g_2+g_4
0011	$g_1+ g_2$	1011	$g_1+g_2+g_4$
0100	g_3	1100	$g_3+ g_4$
0101	$g_1+ g_3$	1101	$g_1+g_3+g_4$
0110	g_2+g_3	1110	$g_2+g_3+g_4$
0111	$g_1+ g_2+g_3$	1111	$g_1+g_2+g_3+g_4$

The first LUT stores all the possible combination of the low-pass coefficients, whilst another LUT contains the pre-computed values of the high-pass coefficients. Moreover, the outputs from both LUTs are combined using a tree structure of two input adders. Then, to accesses the LUTs contents, 4-bits input vector is used, where an appropriate elements of the input vector is added using a serial addition. Consider the input vectors have a length of N = 4, then the vector multiplication operations are described as follows:

$$f = (f_3, f_2, f_1, f_0)$$

$$\tilde{f} = (f_1, f_0, f_3, f_2, f_1, f_0)$$

$$\tilde{f} \cdot v = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & 0 & 0 \\ 0 & 0 & h_1 & h_2 & h_3 & h_4 \\ g_1 & g_2 & g_3 & g_4 & 0 & 0 \\ 0 & 0 & g_1 & g_2 & g_3 & g_4 \end{bmatrix} . \begin{bmatrix} f_1 \\ f_0 \\ f_3 \\ f_2 \\ f_1 \\ f_0 \end{bmatrix}$$

$$\tilde{f} \cdot v = \begin{bmatrix} h_1 f_1 & h_2 f_0 & h_3 f_3 & h_4 f_2 \\ h_1 f_3 & h_2 f_2 & h_3 f_1 & h_4 f_0 \\ g_1 f_1 & g_2 f_0 & g_3 f_3 & g_4 f_2 \\ g_1 f_3 & g_2 f_2 & g_3 f_1 & g_4 f_0 \end{bmatrix} \quad (7)$$

Consider the row one and two from equation (7), these computed values represents the low-pass coefficients results. The combination of the appropriate elements, [f_1 + f_3] in the first column are the first input bit of the LUT address, whilst [f_0 + f_2] elements in the second column are used as the second bit of the LUT address and so on. For the high-pass coefficients results that represented in the row three and four are used the same process to gets the LUT address.

3.2 Daubechies 6-tap (Daub6) algorithm

The Daub6 wavelet is the most localised members among Daubechies wavelet families and it has six scaling signals and wavelets coefficients as given in equation (8) and (9) respectively, where $z_1 = \sqrt{10}$ and $z_2 = \sqrt{5+2\sqrt{10}}$.

$$h_0 = \frac{1+z_1+z_2}{16\sqrt{2}}, h_1 = \frac{5+z_1+3z_2}{16\sqrt{2}}, h_2 = \frac{10-2z_1+2z_2}{16\sqrt{2}}$$
$$h_3 = \frac{10-2z_1-2z_2}{16\sqrt{2}}, h_4 = \frac{5+z_1-3z_2}{16\sqrt{2}}, h_5 = \frac{1+z_1-z_2}{16\sqrt{2}} \quad (8)$$

$$g_0 = h_5, g_1 = -h_4, g_2 = h_3, g_3 = -h_2, g_4 = h_1, g_5 = -h_0 \quad (9)$$

The 1-level Daub6 scaling signals and wavelets are defined in the same way as Daub4 wavelet and the last iteration to compute the scaling signals is described in equation (10). Since the scaling signals have length six, it would send h_2, h_3, h_4 and h_5 beyond the end of the array.

$$A^1_{N/2} = (h_2, h_3, h_4, h_5, 0, 0, ..., 0, h_0, h_1) \quad (10)$$

In addition, the Daub6 wavelet filter is also categorised in the long filters case. Thus, the pre-computed values are partitioned into smaller LUTs. The same method as Daub4 architecture is applied in designing the Daub6 architecture. Two small LUTs are built to store the pre-computed values for the Daub6 scaling signals and wavelets coefficients. Furthermore, 6-bits of input vectors, [b_{06}, b_{16}, b_{26}, b_{36}, b_{46}, b_{56}] are used to addresses the LUTs values.

4. Proposed Systems Architecture and Implementation

4.1 System Overview Applications

Fig. 1 illustrates the proposed system applications for 3-D HWT, Daub4 and Daub6. Focusing on the transform block, the proposed architectures for 3-D Daub4 and Daub6 are divided into three 1-D Daub4/Daub6 with transpose module in between. Instead of creating three different block operations (rows, columns, sub-images) it's better to use just one block operation that performs the same process. Indirectly, reduce the circuits design for FPGA implementation. Each 1-D Daub4/Daub6 block will performs the algorithms computations, in which the 3-D image with $N \times N \times N$ point is fed into the first 1-D Daub4/Daub6 block. Then, the second 1-D Daub4/Daub6 block take the transposed coefficients values from the transpose module, T_1 and computes the Daub4 and Daub6 algorithms. Finally, the values of the transposed coefficients produced by the second transpose module, T_2 are fed into the last 1-D Daub4/Daub6 block and the 3-D transformed coefficients are generated.

In addition, the ROM is used for store the pre-computed values. The proposed architectures of 3-D Daub4 and Daub6 use a different number of bits for accessing the ROM values. For the 3-D Daub4 architecture, it uses 4-bits input ROM, whilst 6-bits input

ROM are required to get the pre-computed values stored in the 3-D Daub6 architecture. Moreover, the transpose modules, T_1 and T_2 make use of on chip RAM as a purpose of data storage for the intermediate results and transpose the row vectors. The RAMs are doubled buffered, in which ping-pong buffer architecture is applied. The ping-pong buffer architecture has two separate memories, where both of them are arranged to appear as a single dual-port buffer. Furthermore, the dual ports synchronous RAMs can perform read and write operations simultaneously. During the second 1-D Daub4/Daub6 block reading the transposed coefficients values from the first transpose module, T_1, the first 1-D Daub4/Daub6 block is ready to get a new data.

4.2 DA Implementations

Fig. 2 depicts the proposed architecture for 1-D Daub4 with DA implementation. The proposed architecture use two smaller LUTs for avoid the increasing memory size. Each of the LUT has 4-bits input address and all the possible combination of the filter coefficients is stored in the LUTs. In addition, the sign bit signal, T_s is used for the LUTs selection. If T_s is high, the first LUT (Unit 1) is selected, otherwise the second LUT (Unit 2) is chosen. On the other side, the purpose of the scaling accumulator is to computes the final output coefficients based on the left-shifted version, similar as Daub4 architecture. The adder tree structure is the actual process that operated in the scaling accumulator, where the previous result values are added together with the current result values to get the final dot product values, Y in four cycles.

The proposed architecture for Daub6 makes use of two small LUTs as illustrated in Fig. 3. Each LUT requires 2^6 = 64 words LUTs with 6-bits input address. Moreover, the pre-computed values of the scaling signals, h_1−h_6 are stored in the LUT (Unit 1), whilst the pre-computed values of the wavelet coefficients, $g_1 - g_6$ are stored in the LUT (Unit 2). The similar process for computes the 3-D wavelet transformation of Daub4 is applied to the Daub6 architecture. The only difference is the uses of N-bits input LUT address and the size of the memory size. Furthermore, the output values from the LUT (Unit 1) are available when the sign bit signal, T_s is high, whilst when the T_s signal is low, the pre-computed values from the LUT (Unit 2) are generated. For the final result computations, three stages of addition processes between the previous and recent result values of the LUTs are performed in six cycles.

5. Results and Analysis

Xilinx ISE 14.2 design flow has been used as a design flow reference and the proposed two architectures have been implemented on the on sbRIO-9632 with Spartan-3 (XC3S2000). To evaluate the performance of the proposed architectures in terms of objective evaluation, four parameters have been selected including the area (slices), maximum frequency (MHz), power consumption (mW), latency (ns) and throughput (Mbps). Whilst, five respondents are selected to evaluates the reconstructed images based on the MOS indicator for the subjective evaluation.

5.1 Objective Evaluations

Table 3 summarise the implementation results for the proposed architectures. In terms of area, Daub4 architecture requires less area (8.0%) with 126 mW power consumption at 51.71 MHz speed. In comparison with Daub4, Daub6 architecture requires 1.6% of more area and consumes 11 mW more power with 50.49 MHz. Moreover, to visualise the chip layout, a chip floor plan for the proposed architectures are given in Fig. 4. The cryan colour scheme in the red line shows the use of CLBs for implement the proposed Daub4 and Daub6 architectures. From the chip layout visualisation, the Daub4 architecture use less area compared with Daub6 architecture. These results verified that more resources are needed to implement the Daub6 algorithms.

Table 3 Implementation results.

Parameters	Proposed DA architectures	
	Daub4	Daub6
Area (slices)	1,648	1,972
Maximum frequency (MHz)	51.71	50.49
Power consumption (mW)	126	137
Latency (ns)	67	120
Throughput (Mbps)	245	294

On the other side, Daub4 architecture necessitates 67 ns to transmit the packet of data for produce an output. In contrast, Daub6 gets a higher latency value which is 120 ns respectively. It shows that Daub4 architecture can perform the data transmission process in short time compared to Daub6. However, Daub6 architecture get higher throughput values of 294 Mbps. Whilst, Daub4 gets 16.7% less throughput values compared with Daub6 architecture.

The comparison of the PSNR values in dB among the proposed architectures have been analysed as summarised in Table 4. In summary, Daub4 architecture provides better result for the transformed image with CT modalities, whilst Daub6 architecture provides better result for MRI and PET transformed images.

Table 4 PSNR values in dB for different wavelet filters.

Medical modalities	Wavelet filters	
	Daub4	Daub6
CT	39.69	38.04
PET	38.01	39.58
MRI	37.81	39.40

5.2 Subjective Evaluations

Five observers of medical doctors carried out the subjective test. The viewers focus on the difference between the reconstructed images with the original image, in which any information loss cannot be accepted. The representative of the subjective test is MOS as

described in Table 5, where it includes absolute and relative scores. In this study, only the absolute score is used in order to seek the consistency between the subjective and objective measurements. A score of 5 is perfect reconstruction (Excellent), score of 4 represents a little noise, which can be ignored (Good), score of 3 illustrates noise which can be seen evidently, but can be accepted (Fair), score of 2 shows a lot of blocking artifacts, which can't be accepted (Bad) and finally score of 1 describes too much blocking artifacts, thus can't be tolerated (Very bad).

Table 5 MOS indicator for subjective evaluations.

Absolute score		Relative score	
5	Excellent	5	Excellent
4	Good	4	Good
3	Fair	3	Fair
2	Bad	2	Bad
1	Very bad	1	Very bad

Table 6 describes the average of absolute score of five observers for the CT, PET and MRI images. Based on the results, it can be summarised that Daub4 wavelet filter give good quality for the CT reconstructed image, whilst the Daub6 wavelet filter provides excellent results for both PET and MRI images.

Table 6 Average scores.

Medical modalities	Proposed DA architectures	
	Daub4	Daub6
CT	5	4
PET	4	5
MRI	4	5

6. Discussions

In summary, the results indicate that the proposed Daub4 architecture provides better saving area by 16.4% compared to Daub6 architecture. In terms of maximum frequency, Daub4 architecture yields 2.6% better maximum frequency than Daub6 architecture. Whilst, in terms of power, Daub4 architecture consumes less power by 8.0% than Daub6 architecture. Moreover, Daub4 architecture is capable to transmit large amount of data packet in higher speed compared with the Daub6 architectures. On the side, in terms of the image reconstruction quality, the proposed architecture with Daub4 wavelet filter gives 4.2% higher PSNR values for CT image compared to Daub6 wavelet filter. Furthermore, the uses of Daub6 wavelet filter in the proposed architecture yields better reconstruction quality for MRI and PET images by 3.9% and 4.0% respectively compared to Daub4 wavelet filter. In brief, the proposed Daub4 and Daub6 architectures provide significant results in terms of PSNR values for the medical modalities images.

Comparative study for both proposed architectures shows an imperative conclusion concerning the higher vanishing moment of Daub4 and Daub6 wavelet filters. Analysis for the performance achieved in terms of area utilised, maximum frequency, power consumption, latency and throughput reveals that complex designs can be implemented on FPGA and hence carry out a better performance achievement.

7. Conclusion

Two architectures for 3-D Daub4 and Daub6 have been proposed in this paper based on transpose computation for transform block of medical image compression. Comparative study for both architectures have reveals that Daub4 wavelet filter provides better achievements in terms of area than Daub6 wavelet filter, whilst in terms of power consumption, Daub4 wavelet filter consumes less power and directly yields better maximum frequency.

On-going research is focusing on the design and FPGA implementation of 3-D Daub4 and Daub6 using other arithmetic technique such as systolic design. Other wavelet filters such as Symlet, Coiflet and Biorthogonal as well as various transform size and real 3-D medical imaging modalities will be further explored to demonstrate the efficiency of the proposed architecture in medical imaging compression systems.

3.1 Acknowledgment

The authors would like to thank the Universiti Tun Hussein Onn Malaysia (UTHM) and Ministry of Higher Education Malaysia for funding this research work through Fundamental Research Grant Scheme (FRGS).

References

[1] Ahmad A., Amira A., Guarisco M., Rabah H. and Berviller Y. Efficient implementation of a 3-D medical imaging compression system using CAVLC. *In: Proceedings of 2010 IEEE 17th International Conference on Image Processing*, (2010), pp. 3773–3776.

[2] Imaging M. and Management I. Medical imaging in the cloud. Retrieved October 7, (2012), from http://www.corp.att.com/healthcare/docs/medical-imaging-cloud.pdf.

[3] Ahmad A. and Amira A. Efficient reconfigurable architectures for 3D medical image compression. *In: Field-Programmable Technology, 2009. FPT 2009.International Conference on.* (2009), pp. 472-474.

[4] Jiang M. and Crookes D. FPGA implementation of 3D discrete wavelet transform for real-time medical imaging. *In: In Circuit Theory and Design (ECCTD 2007), Proc. 18th European Conf. on*, (2007), pp. 519–522.

[5] Jiang M. and Crookes D. Area-efficient high-speed 3D DWT processor architecture. *In: IEEE Electronics Letter*, Volume 43, (2007), pp. 502–503.

[6] Rafael C.G. and Richard E.W. *Digital Signal Processing Third Edition*, chapter 1. *Pearson Prentice Hall*, (2008), pp. 3–7.

[7] Gokturk S.B., Tomasi C., Girod B. and Beaulieu C. Medical image compression based on region of interest with application to colon CT images. *In: Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, Volume 3, (2001), pp. 2453-2456.

[8] Kassim A.A., Pingkun Y., Wei S.L. and Sengupta K. Motion compensated lossy-to-lossless compression of 4-D medical images using integer wavelet transforms. *In: Information Technology in Biomedicine, IEEE Transaction on*, Volume 9, (2005), pp. 132-138.

[9] Ghrare S., Ali M., Ismail M. and Jumari K. Diagnostic quality of compressed medical images: objective and subjective evaluation. *In: Modeling Simulation, 2008. AICMS 08. Second Asia International Conference on,*, (2008), pp. 923–927.

[10] Sanchez V., Abugharbieh R. and Nasiopoulos P. Symmetry-based scalable lossless compression of 3D medical image data. *In: IEEE Transactions on Medical Imaging*, Volume 28, (2009), pp. 1062-1072.

[11] Sriraam N. and Shyamsunder R. 3-D medical image compression using 3-D wavelet decoders. *Journal of Digital Signal Processing*, 21, (2011), pp. 100–109.

[12] Krill B., Ahmad A., Amira A. and Rabah H. Efficient architectures for 3D HWT using dynamic partial reconfiguration. *Journal of Systems Architecture: the EUROMICRO Journals*, 56(8), (2010), pp. 305-316.

[13] Sang-Jin L., Kyung-Chang P., Yeon-Ho K., Yun-Ki H., Younggap Y., Kyoung-Rok C., Tae W.C. and Kamran E. 3D data compression and encryption for bio-medical health care monitoring and management system. *In: Biomedical Circuits and Systems Conference, 2009.BioCAS 2009, IEEE*, (2009), pp. 161-164.

[14] Low Y. and Besar R. Wavelet-based medical image compression using EZW. *In: Telecommunication Technology, 2003. NCTT 2003 Proceedings. 4th National Conference on*, (2003), pp. 203-206.

[15] Montgomery D., Murtagh F. and Amira A. A wavelet based 3D image compression system. *In: Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, volume 1, (2003), pp. 65-68.

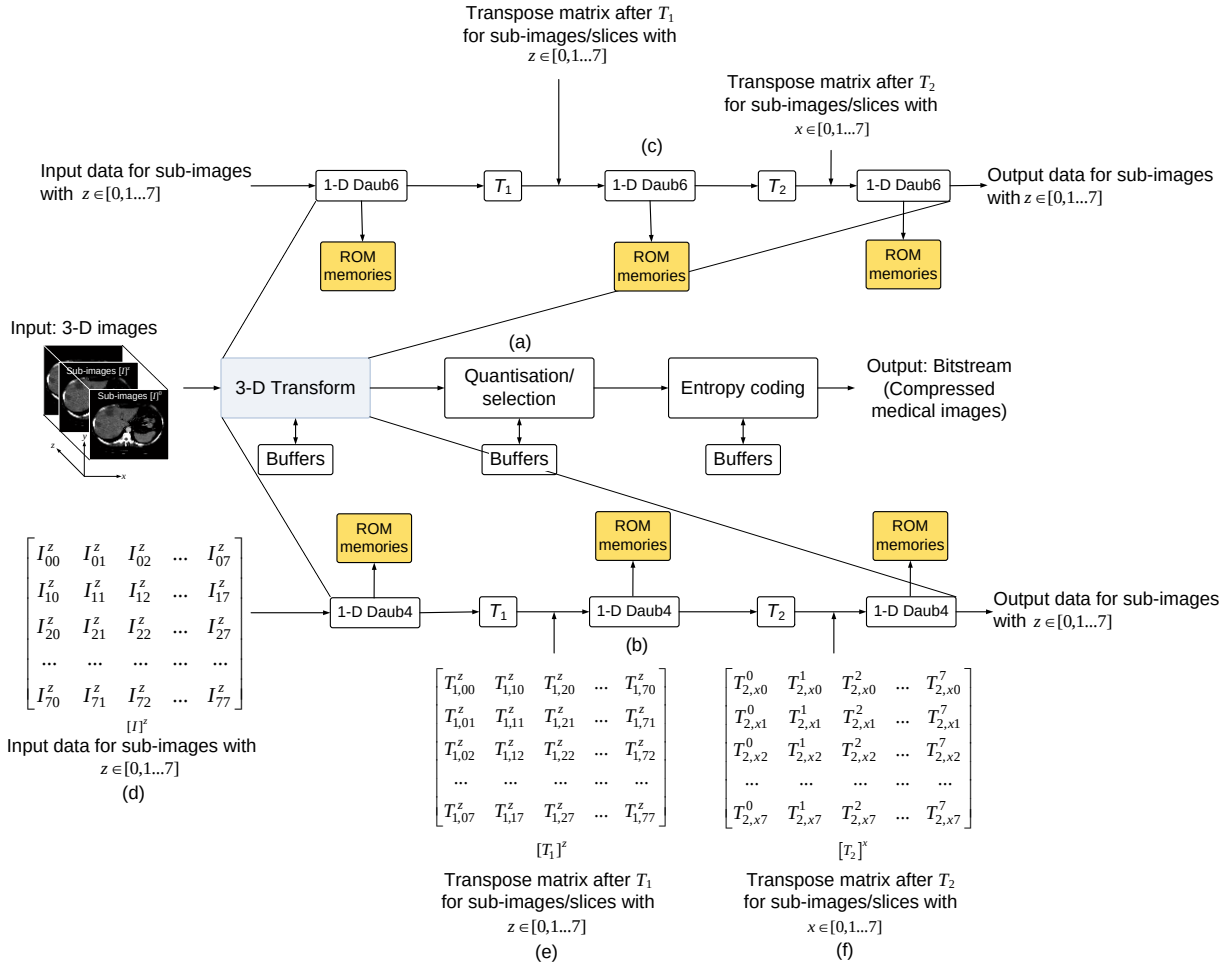

Fig. 1 Proposed system architectures.
(a) Compression system overview (b) Architecture for 3-D Daub4 with transpose-based computation
(c) Architecture for 3-D Daub4 with transpose-based computation (d) Input data for sub-images for $[I]_z$ (e) Transpose matrix after T_1 (f) Transpose matrix after T_2.

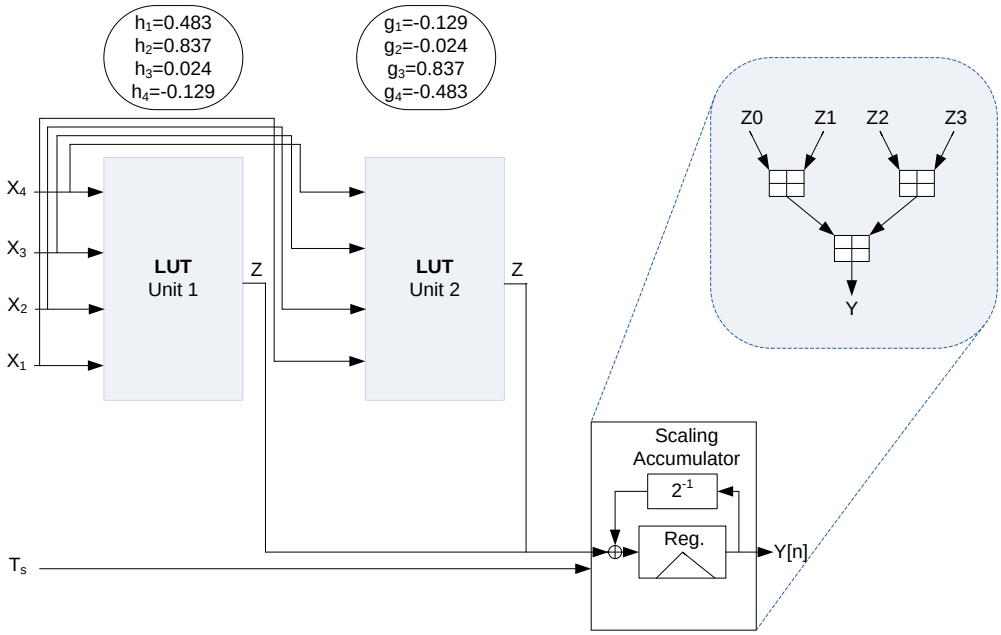

Fig. 2 Proposed system architecture for 1-D Daub4.

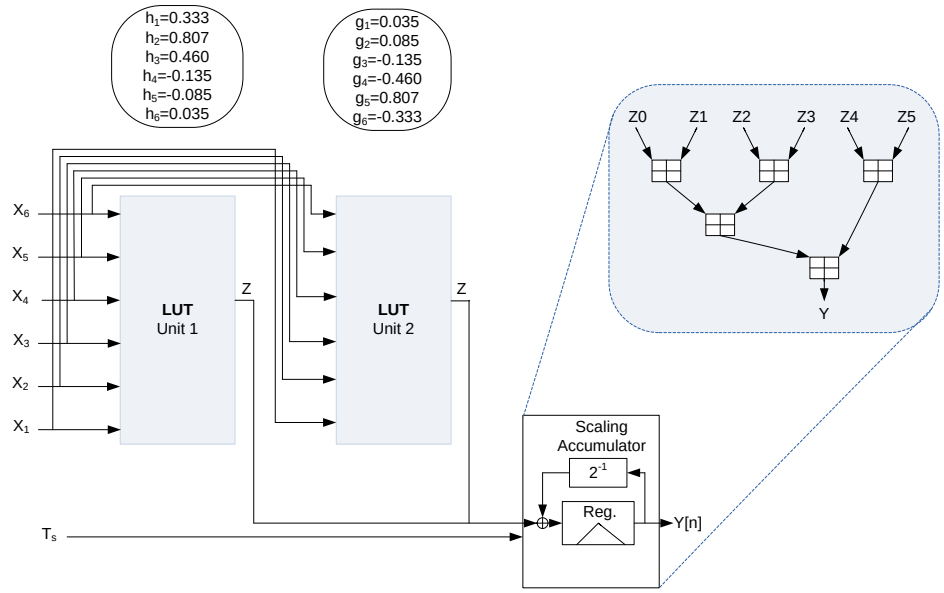

Fig. 3 Proposed system architecture for 1-D Daub6.

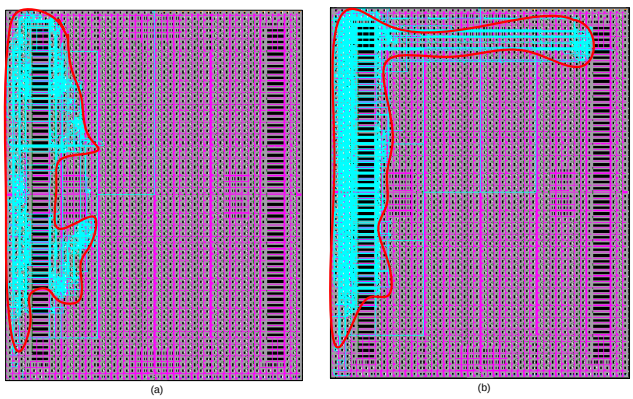

Fig. 4 Chip layouts visualisation. (a) Daub4 (b) Daub6