



A novel ortho normalized multi-stage discrete fast Stockwell transform based memory-aware high-speed VLSI implementation for image compression

G kiranmaye¹ · Srinivasulu Tadisetty²

Received: 16 May 2018 / Revised: 5 December 2018 / Accepted: 11 December 2018 /

Published online: 10 January 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Image Compression is one of the emerging technique of Digital System for storing, retrieving of digital media applications. The main problem of Image Compression is requiring less space for storage and computation speed. In this paper we address this problem and develop a memory-efficient high speed architecture which is implemented based on orthonormalized multi-stage Fast-DST processing unit to perform lifting operation. The proposed multi-stage transform unit performs the split, predict and the update operations by considering the odd samples which are neglected in other lifting transforms. This results in speeding up the process because of the simultaneous execution of both samples. The RTU and CTU are erected with the aid of delay elements and the lifting coefficient, which further tends to attain the optimized processing speed. To address the problem of high cost of memory, multi stage proposed DST unit are combined to build a parallel multi-stage architecture which can perform multistage parallel execution on input image at competitive hardware cost. Finally, the proposed method attains better results when they are compared with existing in terms of memory complexity, low power, low latency.

Keywords Multistage discrete Stockwell transform (MFDST) · RTU · CTU · DST

1 Introduction

The image compression is an important factor that involves huge data storage, transmission and retrieval such as for multimedia, documents, video conferencing, and medical imaging [23]. It is seen that the objective of image compression technique is to reduce redundancy of the image

✉ G kiranmaye
kirangambala@gmail.com

Srinivasulu Tadisetty
drstadisetty@gmail.com

¹ Jawaharlal Nehru Technological University, Kukatpally, Hyderabad, Telangana, India

² Kakatiya Engineering College, Kakatiya University, Warangal, India

information in order to have the capacity to store or transmit information in an efficient frame [21]. Thereby utilizing the designing technologies of VLSI system we can ready to expel the image redundancy and also compress the image [7]. Due to the fast advance of VLSI design technologies, numerous processors in light of sound and image signal processing have been created recently. The Very-Large Scale Integration (VLSI) is the process of creating incorporated circuits by combining a huge number of transistor-based circuits into a semiconductor chip [14]. The principal of semiconductor chips held one included an ever increasing number of transistors, and as a consequence, more individual functions or systems were coordinated after some time. The principal incorporated circuits held just a couple of devices, maybe upwards of ten diodes, transistors, resistors, and capacitors, making it conceivable to fabricate at least one logic gates on a solitary device. Presently referred to reflective as “small-scale integration” (SSI), enhancements in technique prompted devices with many logic gates, known as large scale integration (LSI), i.e. systems with no less than a thousand logic gates [22, 27].

The technology in the past presents the microprocessors which have a huge number of gates and individual transistors [28]. The advancements in VLSI technology give capable devices to the acknowledgment of complicated image and video processing systems with the knowledge of lifting scheme. Prior the VLSI system utilized lifting based DWT (Discrete Wavelet Transform) for image compression [2]. The execution of DWT in a practical system has issues. To begin with, the complexity of the wavelet change is a few times higher. Second, DWT needs additional memory for storing the intermediate computational results [13].

In addition, DWT needs to process monstrous measures of information at high speeds [30]. The utilization of software usage of DWT image compression gives adaptability to control, however it may not meet planning constraints in certain applications [25]. Hardware execution of DWT has practical obstacles. To start with, that is the high cost of hardware execution of multipliers. Channel bank execution of DWT contains two FIR channels. It has customarily been actualized by convolution or the limited drive reaction (FIR) channel bank structures. Such usage requires both substantial numbers of arithmetic computations and storage, which are not attractive for either high speed or low power image/video processing applications [20, 24].

Nowadays the multiple problems have been achieved by using Memory-Efficient Architecture of 2-D Dual-Mode Discrete Wavelet Transform Using Lifting Scheme for Motion-JPEG2000 [10]. S-transform is performed by isolating the low-frequency information from high-frequency counterparts to expel the disadvantages of DWT. This operation can be acknowledged in various ways, which could be level-by-level, block-based or linear based transformation Compared with other previous architectures, the internal memory size of this architecture is very small. It adopts parallel reduce the internal and increase the operation speed [4]. Shifters and adders are used to replace multipliers in the computation to reduce the hardware cost and low complexity in the VLSI implementation. Due to the characteristics of low memory size and high operation speed, it is suitable for VLSI implementation and can support various real-time image/video applications such as JPEG2000, motion-JPEG2000, MPEG-4 still texture object decoding, and wavelet-based scalable video coding [16, 17].

Explosive development in the digital VLSI field, particularly in deep submicron CMOS Nano electronics, is powered by novel inventions in physics at the device level with technological progressions [18]. In parallel to technological progressions, the performance of the system is quickly enhancing at the algorithms level by enlightened architectural designs, which progressively take into non-typical mathematical methods and optimization techniques that are precisely tailored for digital VLSI devices with superior performance [8].

In specific, the utilization of numerous theoretic and estimation techniques for the development of fast algorithms having lower complexity, lower power consumption and/or higher accuracy is critically challenging one [19]. Several multi-dimensional architectures are explained in the literature for real time implementation of wavelet lifting schemes [1]. Based on these problems mentioned above in this paper we derive an efficient architecture for the hardware implementation of VLSI for image compression.

Section 2 describes the related works. Section 3 describes the proposed architecture. Section 4 describes the simulation results for the proposed architecture. Section 5 describes the conclusion of the work followed by the references.

2 Related works

Karthikeyan et.al [3] explained a new methodology is required to handle such problems, three dimensional(3-D) discrete Wavelet Transform (DWT) is introduced over here to decompose the video into frames substantially with proper signal levels as well as make them as a set of finite rules. This 3-D DWT methodology is mainly used in this approach to reduce the difficult processing delay and improves the throughput efficiency. In this paper, Lifting Approach is used to process the input in more refined manner, which reduces the memory consumption over processing as well as minimize the difficult path-delay over DWT orientations. That is the reason with this approach, the combination of both 3-D DWT and Lifting Scheme are used to attain higher efficiency, lowest delay and highest throughput. Even though, it improves throughput the efficiency is not as much improved.

Abhishek Choubey et.al [9] discussed a novel data-access scheme is proposed to avoid data multiplexing, which is common in the existing parallel architectures. Further, a block formulation is presented for vector computation of multi-level lifting 2-D DWT. A generic design of processing unit for computing one-level 2-D DWT is derived using the proposed block formulation. The proposed generic design resizes by a single parameter, i.e., the input-vector size. A regular and modular parallel architecture is derived using the generic processing unit design. The proposed parallel architecture easily scalable for higher block sizes as well as higher DWT levels without sacrificing its circuit regularity and modularity. Yet, there is a need to improve the throughput.

Goran Savic et.al [12] implemented a more efficient circuit for one-dimensional (1-D) forward discrete wavelet transform (DWT) 5/3 filter. This framework is suitable for image compression systems which use 5/3 filter, e.g., JPEG 2000. We also proposed two-dimensional (2-D) DWT 5/3 architecture which uses implemented 1-D DWT filter design. The proposed 2-D DWT architecture outperforms all previously published architectures in terms of required memory capacity, which is at least 20% lower than memory capacity in any other reported solution. Even though, it improves the memory capacity the throughput has to be yet improved.

Darji et al. [5] presented an algorithm based on a flipping technique dual-scan 1-D and 2-D DWT architectures for a 9/7 lifting filter. The technique was used to implement architecture of both modular and hardware-efficient with a compact and simple control path. In the implemented method, the optimization of the lifting data flow in the serial operation was done using parallel computations and in advance with the independent path pipeline operation to reduce the critical path to one multiplier delay to achieve the efficiency of hardware utilization. In addition, it included a technique to reduce the data path by folding to only six multipliers and

eight adders without affecting the critical path. The architecture mainly concentrated on the hardware efficiencies which was implemented on a field-programmable-gate-array. Memory capacity is said to be increased which exist to be as a drawback.

Huang et al. [11] presented a coordinate rotation digital computer (CORDIC)-based fast radix-2 algorithm for computation of discrete cosine transformation (DCT). Some of the distinguishing advantages deal with the algorithm was Cooley-Tukey fast Fourier transformation (FFT)-like regular data flow, uniform post-scaling factor, and in-place computation and arithmetic-sequence rotation angles. The algorithm which was compared to the existing DCT algorithms had low computational complexity and is highly scalable, modular, regular, and also able to admit implementation of efficient pipelined. The algorithm was provided a suitable implementation for VLSI pipelined. And also, it provided a simple way for reconfigurable implementations or unified architecture for DCTs and inverse DCTs.

In [3] the efficiency rate is said to be decreased. Similarly, in [9] there is reduced throughput and in [12] memory capacity gets improved but the throughput yet to be improved. Then, in [5] the memory capacity is to drawback. Thus on focussing all the above disadvantages in this framework we are going to tackle everything by using Discrete stock Well Transform for memory-aware High speed achieving's.

3 Memory-aware high-speed VLSI implementation for image compression using discrete stock well transform

The application of VLSI architectures for image compression is categorised into two i.e. convolution and lifting schemes, where the lifting scheme offers better compression ratio with higher image quality and error immunity than previous JPEG standard and process is carried out with lesser arithmetic resources and also yields better computations with low on-chip memory. In most of the existing works the images were been compressed using DWT. In DWT the high frequency decomposition coefficients are transferred to wired or wireless media for retrieve the compressed image, which in turn cause the coefficients to loss partially. Similarly, it utilizes only line buffers for transposition and storage of temporal data. To overcome above stated problem, there is a need to use restoration algorithms to recover the quality of the image which is acceptable by human eyes. So, in this research paper we are introducing Stockwell Transform instead of wavelet for image compression. Stockwell Transform is used to restore the compressed image without loss in image quality. In Stockwell transform lifting scheme, there is an issue which is given by full time frequency decomposition coefficient and also it which is more expensive for Multi-dimensional. For reduce the expensive rate and also tackle the problem in buffering during multi stage design along intermediate band and also increase in computation time of decision stages a framework is framed out which is stated as the proposed orthonormal Multi-Stage Fast Discrete Stockwell Transform (MFDST). This transform is utilized to form the parallel processor by the property of Orthonormal basis, which in turn transforms a N -point time-frequency representation in the parallel processing, thus achieving the maximum efficiency of representation. This factor is considered to be better than other wavelet transforms like DWT and DCT based processor and in addition to that the number of processor is additionally optimized which extremely decrease the hardware cost. Totally, the proposed architecture attains parallel operation of processor at several level, which also reduce the computational time and memory complexity when compared to other method. The overall proposed architecture is given below Fig. 1.

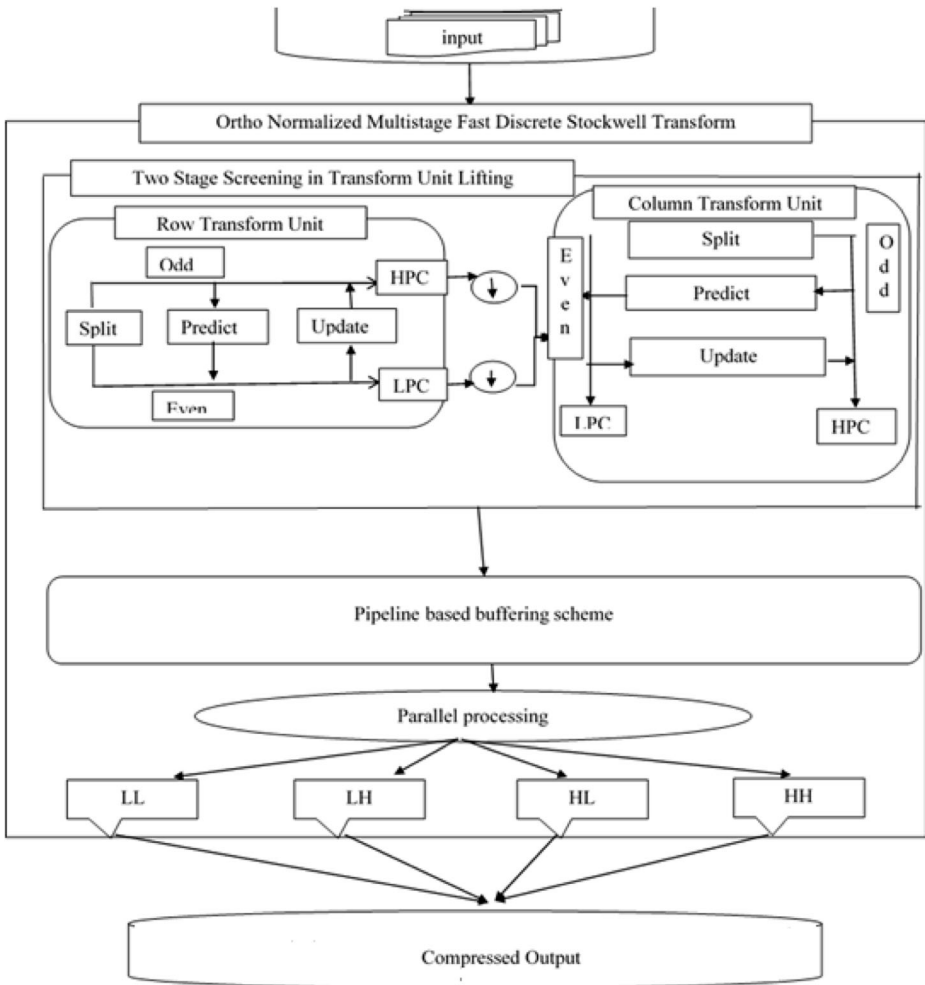


Fig. 1 Flow diagram of proposed MFDST model

In the above proposed architecture, a memory aware high speed VLSI architecture implementation approach based on orthonormal Multi-Stage Fast Discrete Stockwell Transform (MFDST) is explained which is used for reduce the memory cost and increasing the computational speed. Initially, the input image is passed to the transform unit to perform lifting operation. The two stage scanning method is primarily combined in the proposed Transform lifting element that are split, predict and update. The transform unit consist of 2 stage scanning i.e. Row Transform Unit (RTU) and Column Transform Unit (CTU). In the RTU unit, the input is split into odd and even samples and eventually predict and update operations are done respectively. In order to improving the processing speed, we used odd samples in this Transform unit. The RTU and CTU is erected with the aid of delay elements and the lifting coefficients and henceforth it tends to attain an optimize speed, which maintain more multiplier in the lifting element. To eradicate the mentioned problem regarding multiplier in the lifting element an enhanced pipeline based buffering scheme is proposed, which promoted the computation, update of the outcome from intermediate lifting stages. Subsequently, to solve

the complication of buffering in intermediate bands, the proposed MFDST processor are utilized to form the parallel processor by the advantage of orthonormal basis and the number of processors is additionally optimized to reduce the hardware cost. Finally, the proposed architecture attains the parallel operation of processor at several level, which reduce the memory to lowest cost.

3.1 Fast Stockwell transform

The lifting scheme is the one of the approach to compute Stockwell transform which is used for image compression Technique. This method factorizes the wavelets into simple lifting steps and then performs the transform, which causes less computational complexity and memory cost. In the proposed work an VLSI (Very Large Scale Integration) architecture is designed and within this architecture an input is given which is to be compressed. The two stage scanning method Include the Row Transform Unit (RTU) and Column Transform unit (CTU). This RTU and CTU include the lifting coefficient a, b, c and d whose values are found to be constant. Here in the two stage scanning method the input is given which is split to odd and the even samples. In other transformation some DWT models, the odd samples are neglected but in this case, the odd samples are included which results in speeding up the process. The odd samples are referred as the High frequency coefficients and the even samples are referred as the low frequency coefficient. The multi stage Fast Discrete Stockwell Transform is executed essentially in three stages which are called split, predict and update respectively. The Fig. 2 describes the about the operation of RTU and CTU.

Let us consider the input sample image ‘X’. The split phase separates even samples as $x_l[n]$ and an odd sample as $x_h[n]$ from the input samples x , represented in the accompanying condition:

$$\begin{cases} x_l[n] = x[2n] \\ x_h[n] = x[2n + 1] \end{cases} \tag{1}$$

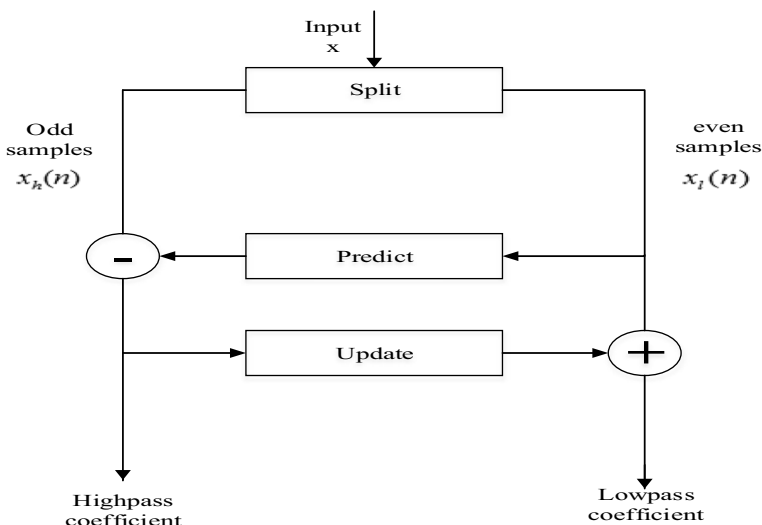


Fig. 2 Transformation unit architecture

The predicted phase generates odd samples as $x_h[n]$ constructed on even samples $x_l[n]$ which prompt the error $f[n]$.

$$f[n] = x_h[n] - P(x_l[n]) \quad (2)$$

Where $P(\cdot)$ represents predict operator in form of interpolating polynomial. The predict is a reversible procedure and $x_h[n]$ can be re-established with $x_l[n]$ and $f[n]$.

The update phase updates the predicted values based on even samples as the follows:

$$g[n] = x_l[n] + U[x_h[n]] \quad (3)$$

Where $U(\cdot)$ represents update operator. The updated values represent the low-frequency coefficient and the predicted values represent the high-frequency coefficient. For the Stockwell-transform received in JPEG2000 standard for lossy compression, the predict and update phases are executed twice as takes after.

The architecture Fast Stockwell-Transform involves multi stage modules. The image pixel are articulate to the RTU (Row Transform Unit) which transforms the adjacent pixel in the same line within each clock. RTU computes 1D S-Transform along the row of pixel data in the original image and store the result in the Row buffer, each of which is configured as a dual port RAM. In this at least 9 input are expected to produce a pair of outputs, thus as soon as the row buffer is fully filled the CTU can be started. The CTU is designed to take into 9 input data and produce a pair of output in each clock, which are at last scaled to the wavelet coefficient by the scale unit.

The main processing element are the RTU and CTU, which will perform predict and update operation. As the input samples available for RTU and CTU are distinctive, which are dependent on the scanning technique and row buffer scheme, the RTU and CTU ought to be composed separately to accomplish an optimized speed. For instance, image data are usually scanned line by line in application, and several rows of the processed result by Row transform are buffered. With the appropriate high band storage scheme for buffer, more data can be fetched into column transform in a clock period. Furthermore, we have proposed a column transform unit according to the availabilities of input samples and also an optimizing method based on registering pipeline technique, which altogether reduce the cost of hardware for the computation of the column transform without compromising the performance of the speed.

3.1.1 The architecture of row transform unit

Row Transform Unit (RTU) perform the parallel processing and it receives a dual scanning scheme for enhancing the processing speed and tackle the wastage of computing resources with the help of RTU architecture. Here the input x is given where the even and the odd samples are split by means of split operation. The RTU includes four delays by which the pipelining architecture is involved for storage along with parallel processing. The odd and the even samples are given to the multiplexer with the help of the delay. The delay is responsible for the pipelining operation so that there will not be any traffic during the process which beautify the strength of DST over DWT. In each process, two adders and one multiplier are found. The first adder is responsible for providing the sum of two consecutive even samples and the output from the adder is given to the multiplier which multiplies along with the lifting coefficient a . The output from the multiplier is utilized by the next adder which adds it along

with the odd coefficients. By which, totally four delay components work with four lifting coefficients a, b, c, and d. Thus, the RTU includes four multipliers and eight adders. The odd samples and the even samples are performed simultaneously by which parallel processing exist. The architecture also includes multiplexers which are employed to provide better results. After the splitting phase, the predict and the update operations are done in the RTU. The RTU architecture is given in Fig. 3.

$$\begin{cases}
 g_n^{(0)} = x[2n], f_n^{(0)} = x[2n + 1] \\
 f_n^{(1)} = e_n^{(0)} + a(g_n^{(0)} + g_{n+1}^{(0)}) \\
 g_n^{(1)} = g_n^{(0)} + b(f_n^{(1)} + f_{n-1}^{(1)}) \\
 f_n^{(2)} = f_n^{(1)} + c(g_n^{(1)} + g_{n+1}^{(1)}) \\
 g_n^{(2)} = g_n^{(1)} + d(e_n^{(2)} + e_{n-1}^{(2)}) \\
 g_n = g_n^{(2)} / v \\
 f_n = v f_n^{(2)}
 \end{cases} \tag{4}$$

The eq. (4) describes the row transform unit. Where $g_n^{(i)}$ and $f_n^{(i)}$ represents the low (even samples) and high (odd samples) frequency coefficients respectively, which is carried out split phase. Then the predicted phase is executed based on even and odd samples of split phase which find the error $f_n^{(n)}$ by the usage of lifting coefficient. The update phase makes use of output of predict phase to generate the updated odd and even samples. $g_n^{(i)}$ and $f_n^{(i)}$ represents the update phase where $i = 0,1,2$ the intermediate results in the lifting procedure represents the scaling coefficient then a, b, c, d represents lifting coefficients Fig. 4 describes the data flow diagram in a RTU based on the above eq. (4). The output of $g_n^{(2)}$ again split as $x(2n)$ and $x(2n + 1)$ which is fed as input to CTU. The detailed flow of data can be elaborated by the usage of a dataflow diagram depicted in Fig. 4.

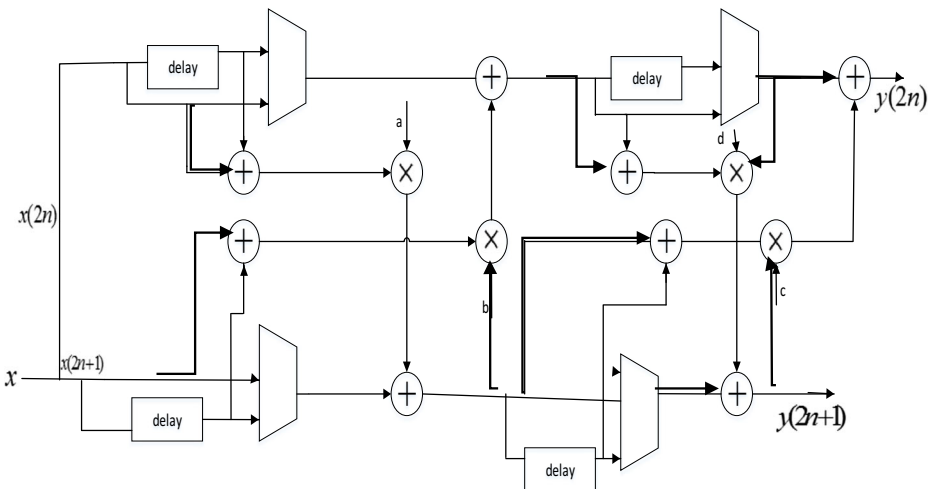


Fig. 3 RTU architecture

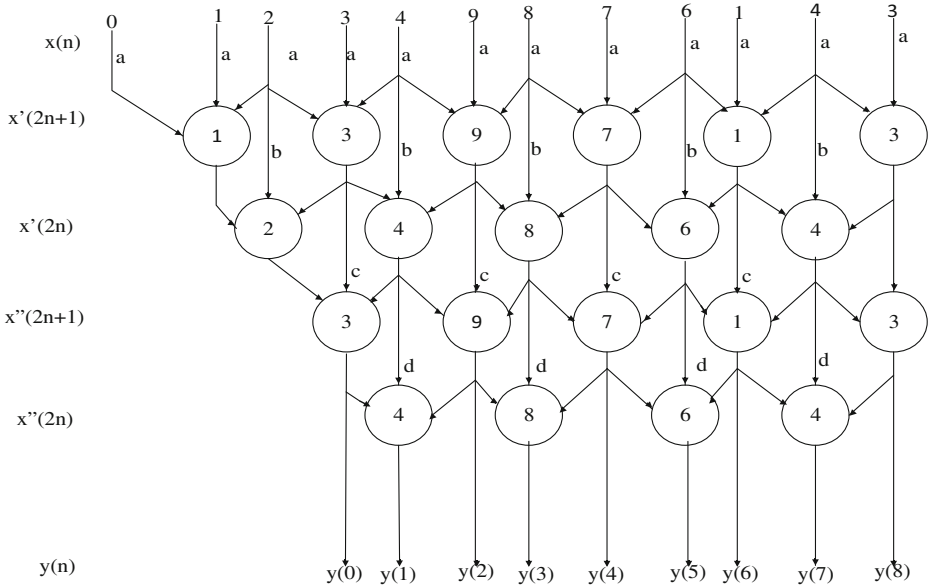


Fig. 4 Dataflow diagram for pipeline system

In the stock well-transform, the input sample sequence is represented by $y[n]$, and as the figure demonstrates by using minimum 9 samples, i.e. The low as well as high-frequency coefficient produced by $y [0]$, $y [23]$, $y [21]$, y . the symbol of the lifting coefficient such as a , b , c , and d was specified previously. The lifting operation is composed of four steps expressed in eq. (4). Each computational node basically consist of a multiplier and two adders perform the summation of two adjacent even samples, then the multiplier performs multiplication of the lifting coefficient and summation value, the adder performs the summation of the product and odd sample. For the other three lifting steps, the computing nodes have the same component with various lifting coefficients.

Through these lines, the four lifting steps together in a combination of data path delay. The Stockwell transform receives one input/output at per cycle, which involves wastage of computing resources and a lower processing speed. This architecture receives a dual data scanning technique where the adjacent samples in the same rows are scanned in $N/2$ speed per cycle, which implies a row of image of size $N \times N$ could be scanned in $N/2$ clock cycles, and the whole image can be filtered in $N^2/2$ clock cycles thus eliminating the disadvantage of Stockwell Transform. This scanning scheme is easy to be realized without additional address generator or control circuit.

3.1.2 The architecture of column transform unit

In the Column Transform Unit (CTU), which perform dual scanning scheme to produce the coefficient element using 9 samples in the row buffer. The CTU scans 9 consecutive samples to produce the low and high coefficient in this architecture and performs four cycles. In each cycle, the lifting coefficient is reduced. The CTU includes 10 multiplexers and it includes independent adder and multiplier. The delay units found are meant for efficient storage of the process by which the results are stored promoting the storage to be accessed whenever

demanded. The first cycle includes all the lifting elements such as the a, b, c, and d. This includes four delays, four multiplexers, eight adders and four multipliers. In the second cycle, the output from the first cycle is taken as the input but exploit only three lifting coefficients such as the b, c, and d. Thus, the second cycle includes three delays, three multiplexers, seven adders and three multipliers. The third cycle takes the second cycle output as the input and promotes further processing with the adders and the multipliers but only with two lifting coefficients such as c and d. Then the third cycle includes two delays, two multiplexers, four adders and three multipliers. In the final cycle, the output from the third cycle is taken as the input where the adder and the multiplier utilize only one lifting coefficients. The four cycle includes one adder single multiplier, two adders and one multiplier. The odd samples and the even samples are performed simultaneously by which parallel processing exist. The architecture also includes multiplexers which are employed to provide better results. After the splitting phase, the predict and the update operations are done in the CTU. The CTU architecture is given in Fig. 5.

The cycles in CTU are expressed where the first cycle is the same as the RTU. The mathematical function for the first equation is given as

$$\begin{aligned}
 g_n^{(0)} &= y[2n], f_n^{(0)} = y[2n + 1] \\
 f_n^{(1)} &= e_n^{(0)} + a(g_n^{(0)} + g_{n+1}^{(0)}) \\
 g_n^{(1)} &= g_n^{(0)} + b(f_n^{(1)} + f_{n-1}^{(1)}) \\
 f_n^{(2)} &= f_n^{(1)} + c(g_n^{(1)} + g_{n+1}^{(1)}) \\
 g_n^{(2)} &= g_n^{(1)} + d(e_n^{(2)} + e_{n-1}^{(2)})g^{(0)}
 \end{aligned}
 \tag{5}$$

In above eq. (5) describes the first cycle includes all the lifting elements such as the a, b, c, and d. This includes four delays, four multiplexers, eight adders and four multipliers which is a least count compared to existing lifting schemes.

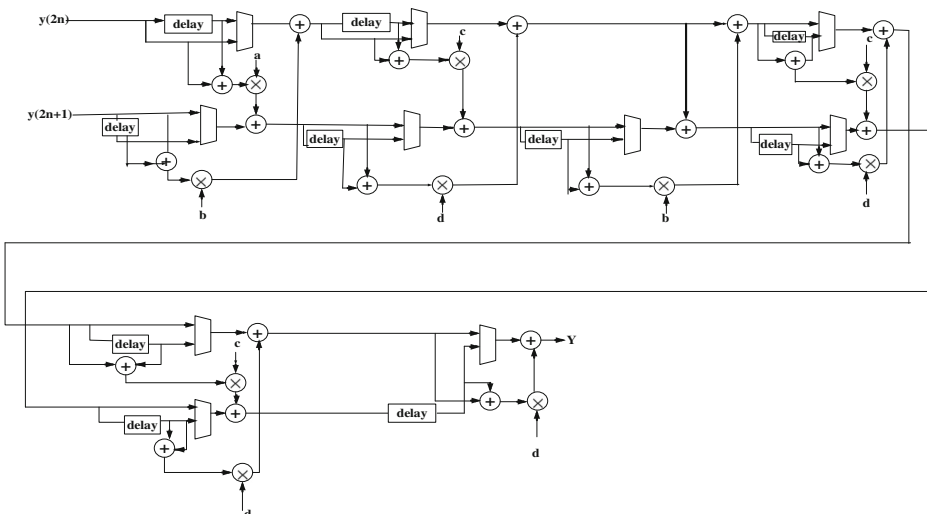


Fig. 5 CTU architecture

The mathematical function for the second equation is given as

$$\begin{aligned}
 g_{(2)}^{(0)} &= y[2n], f_{(2)}^{(0)} = y[2n + 1] \\
 g_{(2)}^{(1)} &= g_{(2)}^{(0)} + b \left(f_{(2)}^{(0)} + f_{(1)}^{(0)} \right) \\
 f_{(2)}^{(2)} &= f_{(2)}^{(1)} + c \left(g_{(2)}^{(1)} + g_{(1)}^{(1)} \right) \\
 g_{(2)}^{(2)} &= g_{(2)}^{(1)} + d \left(x(2n)_{(2)}^{(2)} + x(2n)_{(1)}^{(2)} \right)
 \end{aligned} \tag{6}$$

The above eq. (6) represents the output from the first cycle is taken as the input but exploit only three lifting coefficients such as the b, c, and d. Thus, the second cycle includes three delays, three multiplexers, seven adders and three multipliers. In this phase one lifting coefficient is reduced.

The mathematical function is explained for the third step is given below which includes two lifting coefficients such as the c and d.

$$\begin{aligned}
 g_{(3)}^{(0)} &= y[2n] \\
 f_{(3)}^{(0)} &= y[2n + 1] \\
 f_{(3)}^{(1)} &= f_{(3)}^{(0)} + c \left(g_{(3)}^{(0)} + g_{(2)}^{(0)} \right) \\
 g_{(3)}^{(1)} &= g_{(3)}^{(0)} + d \left(f_{(3)}^{(1)} + f_{(3)}^{(1)} \right)
 \end{aligned} \tag{7}$$

The third cycle takes the second cycle output as the input and stimulates additional processing with the adders and the multipliers but only with two lifting coefficients such as c and d. In the cycle 2 lifting schemes are eliminated.

The mathematical function is explained for the fourth cycle is given below including one lifting coefficient d.

$$\begin{aligned}
 g_{(4)}^{(0)} &= y[2n] \\
 f_{(4)}^{(0)} &= y[2n + 1] \\
 g_{(4)}^{(1)} &= g_{(4)}^{(0)} + d \left(f_{(4)}^{(0)} + f_{(4)}^{(0)} \right)
 \end{aligned} \tag{8}$$

In the final cycle, the output from the third cycle is taken as the input where the adder and the multiplier utilize only one lifting coefficients. The four cycle includes one adder single multiplier, two adders and one multiplier. The odd samples and the even samples are performed simultaneously by which parallel processing.

In spite of CTU and RTU perform precisely the same Lifting Step Element (LSE_S), they differ from each other for the input samples availabilities. The RTU could take in two samples, while smaller and more flexible storage of Row buffer could exhibit higher parallelism. It can be seen from Fig. 2 that 9 samples are needed to produce a low frequency and high frequency coefficient, thus in our proposed design, the row buffer module consist of 9 row buffer. Thus, a column scanning technique, which scan out nine consecutive samples from nine row buffer in a clock, could be realized. Nine coefficient in row buffer could be taken out in each clock and processed by a CTU. Thus a CTU needs more computation resources to process these samples.

The CTU takes nine samples as input, computes through four lifting steps, and produces two transform coefficients finally, when contrasted with RTU, CTU needs four LSEs in the

initial step which prepare nine samples altogether. Three LSEs, two LSEs and one LSE are required in the accompanying 3 stages respectively, which could expend the output of previous lifting steps reliably. This architecture also has the critical path delay indicated in red color, and simple control on synchronization.

In spite of the fact that the column transform accomplishes 2 output /cycle and matches RTU, CTU cost much more computing resources, when column strip from x [14] to x [20] and column strip from x [22] to x [4] are handled, the computing nodes in dash lines are computed twice. The repeated calculation thus calls for more resources.

To overcome this issue, this paper comes with a solution of pipeline buffering scheme, in which the result of initial 3 lifting steps is buffered, fetched and updated in pipeline. If repeated calculation in dash lines are expelled, just the red nodes are required to compute, which calls for only two new samples inputs, x [10] and x [4], considering that x [20] is already in pipeline. Thus, the column transform naturally takes 2 input/output as the RTU, which can be scanned strip-wisely as shown in Fig. 6.

As can be seen from Fig. 7, the red node in predict 1 step requires x [20], x [10] and x [4], and the calculated result is indicated as predict 1. It is necessary to calculate X [20], Predict of node10 and Predict 1 for calculating the red node in Update1. In this manner the value in predict 1 can be fetched and updated with $p1$. For the convenience of description, the predict 1 lifting operation results are denoted as $p1(I,j)$, where I and j are starting row and end row of the column strip respectively. As can be seen from Fig. 6, the initial two rows are processed by RTU, the third row samples produce buffer. So this lead to the emergence of orthonormal execution by the employment of parallel processing.

Since the RTU produce two samples and CTU consumes two samples per clock period, they have no issue in sharing the row buffer, nine coefficients in row buffer could be taken out in each clock and processed by a CTU. Thus a CTU needs more computation resources to process these samples and also the high frequency decomposition coefficients are transferred to wired or wireless media for retrieve the compressed image, which lose the image quality partially. In Stockwell transform lifting, there is an issue which is given by full time-frequency decomposition coefficient and furthermore, which is costlier for Multi-dimensional. With a specific end goal to handle the complexity in buffering along the intermediate band and furthermore increment in computation time of multiple stages a system is available, which is expressed as the orthonormal Multi-Stage, which tackle these above and it gives the low time frequency coefficient and also overcome these problem our proposed orthonormal Multi-Stage Fast Discrete Stockwell Transform (MFDST) using pipeline buffering scheme for parallel

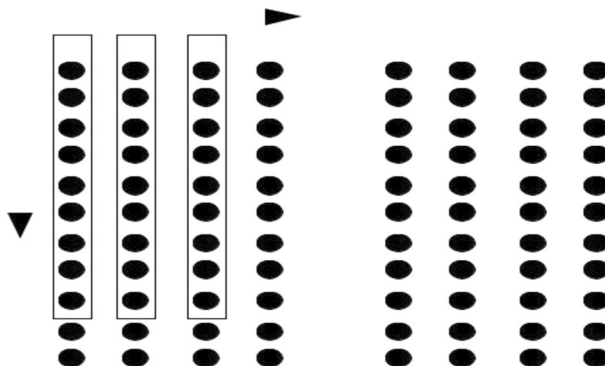


Fig. 6 Column scanning

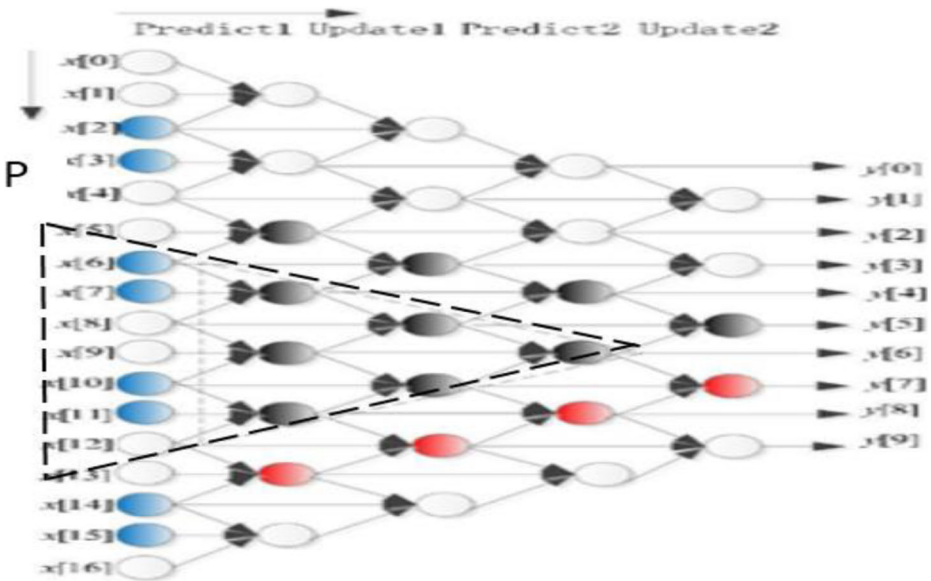


Fig. 7 Data flow diagram of pipeline system

processing and speed efficient way. In addition to the parallelism between the RTU and CTU, the present level and the accompanying levels additionally have the potential parallel limit.

3.2 Multi-stage design for parallel processing

This unit is used for storage. In addition to the parallelism between the RTU and CTU, the present level and the accompanying levels additionally have the potential parallel limit. As long as the previous level produces a specific measure of low frequency transform coefficient, the next level could be activated instantly. Motivated by this observation, Image pixels are

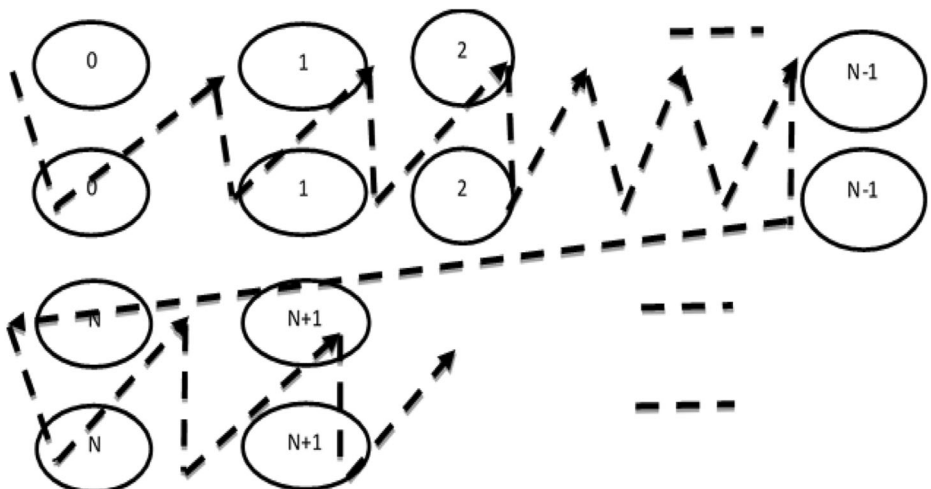


Fig. 8 Optimized column transformation

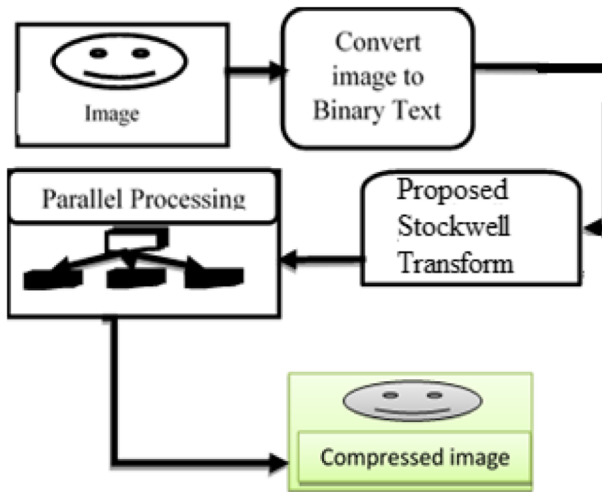


Fig. 9 Block diagram of proposed scheme

scanned out of RAM at two samples per clock, which are processed by Fast-S-Transform of the first Level. As long as the LL sub band coefficients produced are transformed to the next level, Fast S-Transform of the level 2 could be started, and so on.

Our proposed multi-level design spares the need to store intermediate LL sub band coefficient, which could be considerable amount if the original image is extensive. As for speed, multi-level architecture could transform the image up to six levels almost within the time of scanning the image. Indeed, every one of the six levels are finished in nearly the same time, while the time for the conventional serial architecture equals to the summation of processing time of each level. This paper further reduces the quantity of processor to 2, one of which is utilized to execute level 1-transform, and the other to execute level-2 to level j -transform. Three processor 1 contains as LL1 row buffer, buffering one row of low frequency coefficient from level 1–1 transform. The processor 2 contains 5 row buffer, which enable RTU and CTU processing parallel as described already. Likewise, four low frequency row buffer are utilized to synchronize the multi-level transforms. When the processor 1 produces LL low frequency coefficient, they are buffered and processed by processor 2. When processor 2 produce and buffer a row of low frequency from level 2, level is then started, and so on. As each level DST has its own



Fig. 10 Input image

Table 1 Multiplexer, multipliers and adder utilization

Sl.no	Element	No.of Element
1	multiplexers	5
2	16-bit multiplexers	4
3	32-bit multiplier	4
4	16-bit multiplier	3
5	32-bit adder	4
6	16-bit adder	3
7	32-bit Latches	2

frequency components and the even samples or the low-frequency components work simultaneously or in parallel. The memory efficiency is done by using the delay components in the stockwell transform unit. With the aid of the delay components the memory is stored time to time without lose of image quality and hence the memory can be efficiently utilized.

4 Performance evaluation and comparision

Figure 9 describes the proposed scheme by using the Stockwell transform. Initially, an image is given for converting the binary value. After converting binary text format, the image binary data is to be stored in the text file. This is given as input to xilinx and the process of proposed work happens.

Processor: Intel ® Pentium ® CPU G2030 @ 3.00 GHz 3.00 GHz.
RAM: 4 GB.

4.1 Simulation results

The input for proposed VLSI architecture and the compressed output based on this VLSI architecture is given in Fig. 10.

Figure 11 explains the final output after the transform unit. In the transform unit since parallel processing is done with the odd and even samples, the speed is increased. Finally, a VLSI architecture with efficient memory and speed is achieved.

The RTL (Register Transform Logic) schematic overall architecture design is given in Fig. 10 that a synchronous circuit that comprises of two kinds of elements such as the registers and the combinational logic. Further, this describes the pre-optimized design and includes the details such as the adders, multipliers, counters, AND gates and OR gates and thus describing the multipliers and the multiplexers in the RTU and CTU. The schematic architecture diagram is shown in (Fig. 12).

Table 2 Device utilization summary

Device Utilization Memory			
Logic Utilization	Used	Available	Utilization
Number of slice LUTs	16	21,000	0%
Number of slices fully used LUTs FFT pairs	0	16	0%
Number of bonded IOBs	36	210	17%

Table 3 X power analyzer report for 256×256

Supply Summary		Total Current (A)	Dynamic Current(A)	Quiescent Current(A)
Source	Voltage			
Vccint	1.207	0.576	0.045	0.499
Vccaux	2.487	0.098	0.000	0.096
Vcc025	2.498	0.005	0.000	0.002
Supply Power (W)		Total	Dynamic	Quiescent
		0.893	0.050	0.843

Table 1 describes the different number of multiplexers, multipliers and the adders that are utilized in the entire processing of the proposed Stockwell transform both in RTU and CTU.

As the Table 1 describes their total number of multiplexers utilized is 9, total number of multipliers used is 7. with different specifications such as the 32 – bit, 16 – bit This also specifies the total number of adders i.e.7 with different specifications such as the 32 – bit, 16 – bit and the latches or the storage unit utilized is also specified.

Apart from the adders, multipliers, multiplexers and the latches, other device utilizations are also specified as given in Table 2 that provides a detailed description about the different devices that are available and utilized along with the utilization rate. The hardware specification for the proposed method is given as.

Hardware specification:

Family: Spartan3.

Device: XC35200.

Package: FT256.

4.2 Power consumption strategy

The dynamic power consumption of the implemented architecture is calculated using a Xilinx Xpower analyzer for selected image size and different operating frequencies. The Xpower

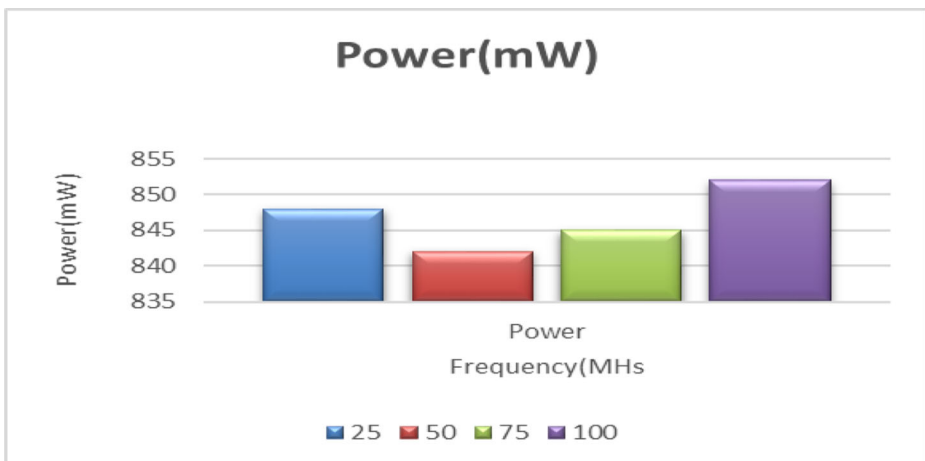
**Fig. 13** Power vs frequency of 256×256

Table 4 X power analyzer report for 8×8

Supply Summary		Total Current (A)	Dynamic Current(A)	Quiescent Current(A)
Source	Voltage			
Vccint	1.206	0.507	0.014	0.498
Vccaux	2.556	0.098	0.000	0.097
Vcc025	2.500	0.002	0.002	0.004
Supply Power (W)		Total	Dynamic	Quiescent
		0.858	0.018	0.841

analyzer report shown in Table 3 and Fig. 13 for 256×256 and 8×8 image pixel size. The dynamic power consumed by the method is 0.050 W and Quiescent power is 0.843 W.

Hence, the total power consumed in coherent MFDST image fusion method is 0.843 W. Hence, the total power consumed in Orthonormal zed MFDST image fusion method is 0.893 W at 100 MHz frequency. This Xpower analyzer Report is taken by the device FPGA Virtex-7 and by using 256×256 and 8×8 image pixel level. The power can be varied at the different operating frequency.

The dynamic power consumption at 25 MHz operating frequency is 858Mw for 256×256 image size and at the similar operating frequency, the dynamic power for 8×8 Image size is 850Mw. The comparison of power for 256×256 image size with various operating frequency is tabulated below. The description of X Power analyser 8×8 image size is given in (Table 4).

Frequency (MHz)	25	50	75	100
Power	848	862	876	883

The above charts show that the power will be increased when increasing the operating frequency. The power can be compared for 8×8 image size with various operating frequency, which is tabulated in Table 5. The power variation according to the operating frequency is clearly described in the chart Fig. 14.

4.3 Comparison results

In this research, an efficient compression technique based on multi stage Fast-discrete Stock well transform (DST) is proposed for image compression and developed the VLSI architecture. The algorithm has been implemented using Xilinx platform. A test images are taken to justify the effectiveness of the algorithm. Figure 15 shows a test image and resulting compressed images using JPEG, GIF and the proposed compression methods.

The experimental results with the proposed compression method have been arranged in the Table 6 for different threshold values. From this table, we find that a threshold value of $\delta = 30$ is a good choice on the basis of trade-off for different compression ratios. Table 7 shows the comparison between JPEG, GIF and the proposed compression method. Experimental results demonstrate that the proposed compression technique gives better.

Table 5 Power variation according to frequency for 8×8 image size

Frequency (MHz)	25	50	75	100
Power	848	842	845	852

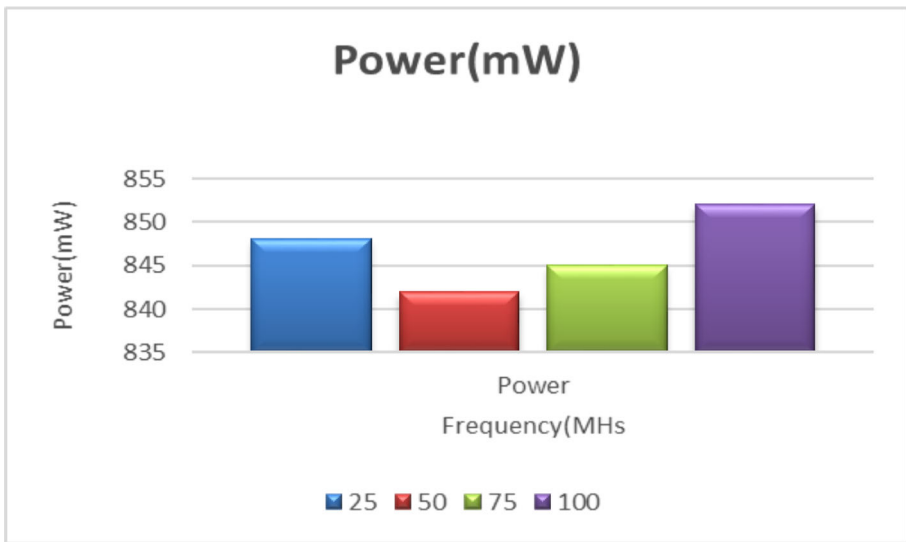


Fig. 14 Power vs frequency of 8 × 8

4.3.1 Compression ratio

Compression ratio is described as the ratio of the uncompressed image to the compressed image as in (10). The compression ratio is described in order to evaluate the performance of the proposed image compression with the aid of MFDST.



a) Original image Resolution: (140x190)



(b) Compressed image using GIF



(c) Compressed image using JPEG



(d) Compressed image using Proposed DS

Fig. 15 A real image and corresponding compressed images with GIF, JPEG and proposed DST methods

Table 6 Proposed method with and without threshold value

Threshold values	Size of the original image (KB)	Size of the compressed image with DST	Compression Ratio
Without Threshold	58	2.47	18.47
$\delta = 10$	58	2.61	20.75
$\delta = 20$	58	2.01	21.38
$\delta = 30$	58	1.48	23.47
$\delta = 40$	58	2.41	22.69
$\delta = 50$	58	2.21	23.81

$$Compression \text{ ratio} = \frac{Uncompressed \text{ image}}{compressed \text{ image}} \tag{10}$$

The above Fig. 16 described comparison of proposed method and existing method in terms of different image compression technique such as GIF, JPEG and DWT. This figure value indicates that the image is compressed much better than the other compression techniques the original image file size is 58 KB.the size of the GIF file of compressed image is 6.40 and the compression ratio is 7.34 then the JPEG compression file size of the image is 3.38 and the compression ratio is 13.90. When compared to above two techniques JPEG attain a higher performance, then DWT compression ratio 24.22 and compressed file size is 1.98 and our proposed method compression of image file size is 1.48 and compression ratio is 23.47. Overall our proposed method attains the greater performance when compared to existing image compression technique.

4.3.2 Time

Minimum input arrival time before clock: 1.662 ns.

Maximum output required time after clock: 112.317 ns.

Maximum combinational path delay: 112.762 ns.

Overall time to compress the image is: 0.001382.

4.3.3 Mean square error (MSE)

MSE is computed by averaging the squared intensity of the input (original) image and the output image pixels is described in below eq. (11),

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e(m, n)^2 \tag{11}$$

Table 7 Comparison ratio using proposed method and existing method

Compression Techniques	File sizes	Compression Ratio
Original Image	58	–
GIF	6.40	7.34
JPEG	3.38	13.90
DWT	1.94	24.22
Proposed	1.48	23.47

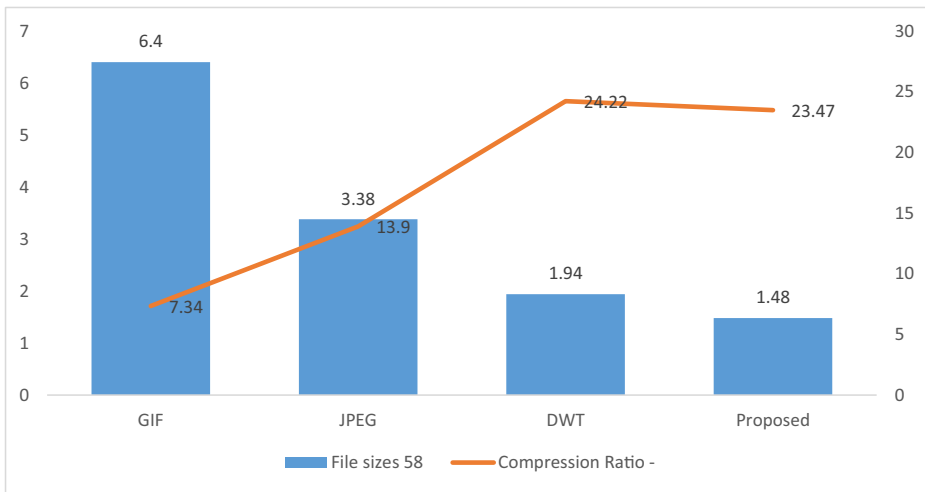


Fig. 16 Comparison ratio of proposed method and existing method

Where,

$e(m, n)$ error difference between the original and the distorted images.

M, N describes the pixel size of the original and the distorted image.

4.3.4 Peak signal-to-noise ratio

Peak Signal-to-Noise Ratio (PSNR) is an exact measure of image quality based on the pixel difference between two images as in (11)

$$PSNR = 10 \log \frac{S^2}{MSE} \tag{12}$$

Where $S = 255$ for an 8-bit image. The PSNR and the MSE are thus mathematically related to each other inversely.

The Comparison graph and table are given in Table 8, Figs. 17, 18 and the comparison is given with the traditional methods such as the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Stockwell transform and finally with the proposed.

In the above Fig. 17 described the comparison of Mean Square Error between the proposed method and Existing method. The Existing Work Mean Square value of DCT is 7.89 then MSE value of DWT is 3.227 Which shows that DCT is higher MSE value than DWT and the MSE value of Stockwell is 3.23, i.e. the Stockwell value of MSE is lower than DWT when

Table 8 Comparison table of MSE and PSNR

Method	MSE	PSNR
DCT [3, 28]	7.89	25.728
DWT [5, 6]	3.227	38.93
Stockwell [26]	3.23	37.5
PROPOSED	2.2727	39.7461

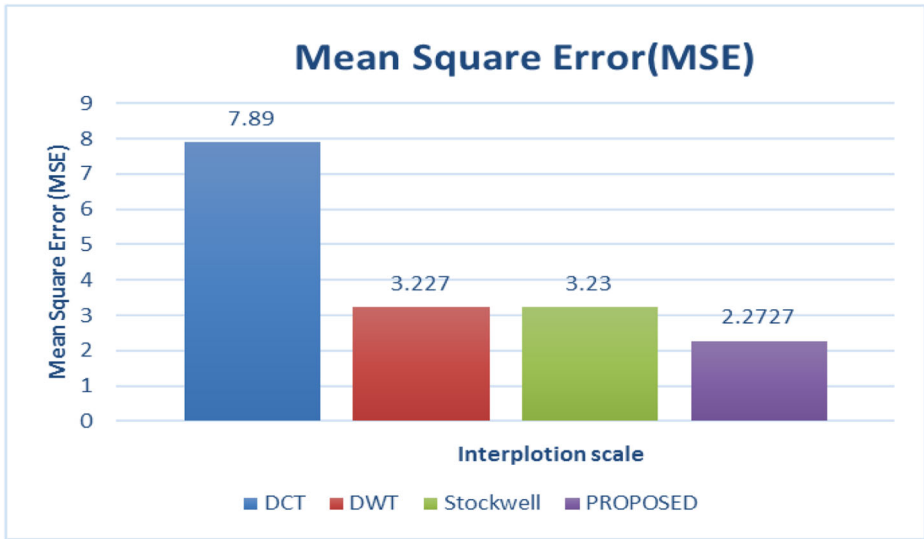


Fig. 17 MSE comparison

compared to our proposed work and existing work our proposed work attains a better performance of Mean Square Value that is 2.2727. Finally, our proposed work attains the lower Mean Square Value.

The above Fig. 18 described the Comparison of Peak signal-to –Noise Ratio (PSNR) existing work and proposed methodology, The Existing Work PSNR value of DCT is 25.728 then PSNR value of DWT is 38.3 Which shows that DWT is higher PSNR value than DCT and the PSNR value of Stockwell is 37.5, i.e. the Stockwell value of PSNR is lower than

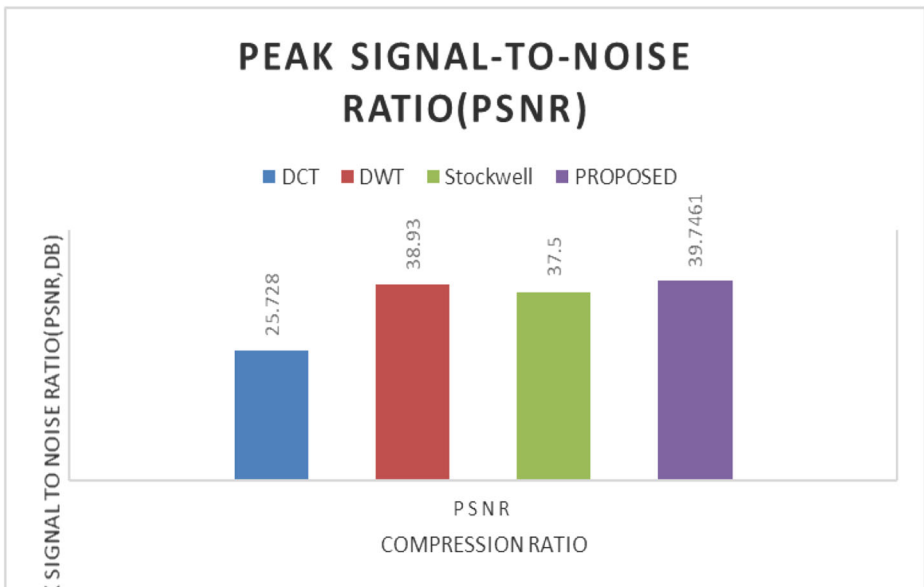


Fig. 18 PSNR comparison

Table 9 Hardware resource comparison

Architecture	Multiplier	Adder	Buffer
Pipeline-DWT [29]	6	8	0
2D-DWT [15]	18	32	4
Parallel-2D-DWT [33]	10	16	4
DCT[34]	10	16	3
Proposed Method	7	7	4

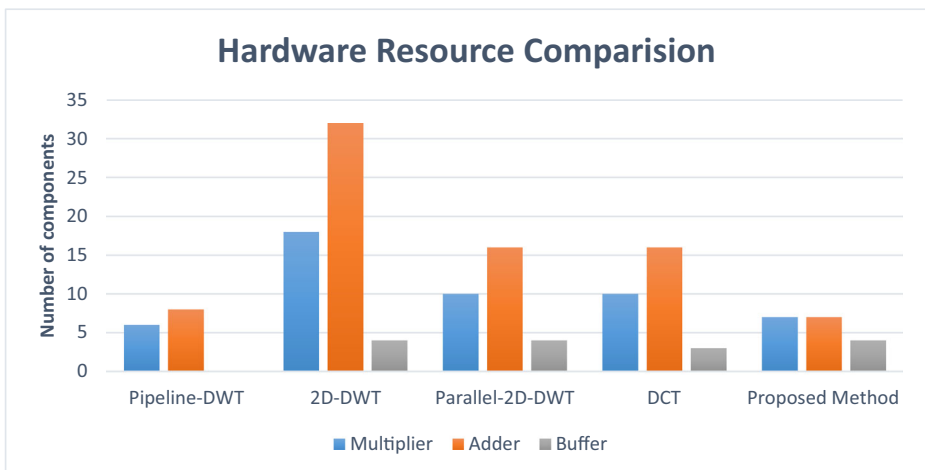
DWT and Higher than DCT. When compared to our proposed work and existing work our proposed work attains a Higher performance of Peak signal-to –Noise Ratio (PSNR) Value that is 39. 7491. Finally, our proposed work attains the higher PSNR value.

In Table 9, the similarity between multistage Fast-DST architecture and existing architecture regarding hardware resources are tables.

From the above Fig. 19, it demonstrates that the multiplier applied in the MULTISTAGE FAST-DST is 7. Adder is 7 and buffer is 4, the hardware resource is decreased when compared with other method. Hence it came to know that the hardware complexity of the proposed method reduces the hardware cost.

5 Conclusion

In this work, an Orthonormalized Multistage Fast Discrete Stockwell Transform has been implemented in VLSI architecture. Here the input image is compressed with the compression ratio of 23.47 and time to compress the image is: 0.001382, this architecture diminishes the memory as well as the time computation. That is achieved through parallel processing of odd and even samples this paves the way to reduce the loss of data and also the computational time. Here delay elements were utilized this directs to reduce the collision occurred during the compression process and also the memory referring process also reduced by means of this architecture. Henceforth the input is compressed through the proposed VLSI architecture and

**Fig. 19** Hardware resources comparison

attains better results compared with the previous methods. The comparisons of Mean Square Error with existing works, our proposed work improves the better results MSE Value is 2.2727 and also value of PSNR is 39.7461 improve the higher performance when compared to previous works. Here also the speed is improved with the help of the parallel processing of the odd and the even samples. Finally, to compare the hardware resources proposed method and existing work, our proposed work reduce hardware resources like multiplier, adder and buffer. Which reduce the number of adder is 7, multiplier is 7 and buffer is 4, Hence it came to know that the hardware complexity of the proposed method reduces the hardware cost. The proposed method performs better when compared to the other existing method. Thus, the VLSI architecture is implemented with high speed and low memory.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Aziz SM, Pham DM (2012) Efficient parallel architecture for multi-level forward discrete wavelet transform processors. *Comput Electric Eng* 38(5):1325–1335
2. Cheng C, Keshab Parhi K (2008) High-speed VLSI implementation of the 2-D discrete wavelet transform. *IEEE Trans Signal Process* 56(1):393–403
3. Darji A et al (2014) Dual-scan parallel flipping architecture for a lifting-based 2-D discrete wavelet transform. *IEEE Trans Circ Syst II: Express Briefs* 61(6):433–437
4. Das A, Hazra A, Banerjee S (2012) An efficient architecture for 3-D discrete wavelet transform. *IEEE Trans Circ Syst Video Technol* 20(2):286–296
5. Haghghi BB, Taherinia AH, Mohajezadeh AH (2018) TRLG: Fragile blind quad watermarking for image tamper detection and recovery by providing compact digests with quality optimized using LWT and GA. *arXiv preprint arXiv:1803.02623*
6. Hashim AT, Ali SA Color image compression using DPCM with DCT, DWT and Quadtree Coding Scheme
7. Hsia C-H, Guo J-M, Chiang J-S (2009) Improved low-complexity algorithm for 2-D integer lifting-based discrete wavelet transforms using symmetric mask-based scheme. *IEEE Trans Circ Syst Video Technol* 19(8):1202–1208
8. Hsia C-H, Chiang J-S, Guo J-M (2013) Memory-efficient hardware architecture of 2-D dual-mode lifting-based discrete wavelet transform. *IEEE Trans Circ Syst Video Technol* 23(4):671–683
9. Huang H, Xiao L (2013) CORDIC based fast radix-2 DCT algorithm. *IEEE Signal Process Lett* 20(5):483–486
10. Huang C-T, Tseng P-C, Chen L-G (2015) Generic RAM-based architectures for two-dimensional discrete wavelet transform with the line-based method. *IEEE Trans Circ Syst Video Technol* 15(7):910–920
11. Imgraben S, Dittmann S Leaf litter dynamics and litter consumption in two temperate South Australian mangrove forests. *J Sea Res* 59(1–2):83–93
12. Kahu S, Rahate R (2013) Image compression using singular value decomposition. *Int J Adv Res Technol* 2(8):244–248
13. Krasikov A (2010) Socioeconomic determinants of infant mortality rate disparities. Clemson University
14. Lai Y-K, Chen L-F, Shih Y-C (2009) A high-performance and memory-efficient VLSI architecture with parallel scanning method for 2-D lifting-based discrete wavelet transform. *IEEE Trans Consum Electron* 55(2):400–407
15. Lai Y-K, Chen L-F, Shih Y-C A high-performance and memory-efficient VLSI architecture with parallel scanning method for 2-D lifting-based discrete wavelet transform. *IEEE Trans Consumer Electron* 55(2):400–407
16. Lee S-J et al (2010) 3-D system-on-system (SoS) biomedical-imaging architecture for health-care applications. *IEEE Trans Biomed Circ Syst* 4(6):426–436
17. Liang D et al (2013) Stacked phased array coils for increasing the signal-to-noise ratio in magnetic resonance imaging. *IEEE Trans Biomed Circ Syst* 7(1):24–30
18. Mohanty Basant K, Pramod Meher K (2011) A memory-efficient architecture for 3-D DWT using an overlapped grouping of frames. *IEEE Trans Signal Process* 59(11):5605–5616
19. Mohanty Basant K, Mahajan A, Pramod Meher K (2012) Area-and power-efficient architecture for high-throughput implementation of lifting 2-D DWT. *IEEE Trans Circ Syst II: Express Briefs* 59(7):434–438

20. Mohanty BK, Meher PK (2013) Memory-efficient high-speed convolution-based generic structure for multilevel 2-D DWT. *IEEE Trans Circ Syst Video Technol* 23(2):353–363
21. Nian YJ et al. (2012) Near-lossless compression of hyperspectral images based on distributed source coding. *Science China Inf Sci*: 1–10
22. Oweiss KG et al (2007) A scalable wavelet transform VLSI architecture for real-time signal processing in high-density intra-cortical implants. *IEEE Trans Circ Syst I: Reg Papers* 54(6):1266–1278
23. Seo Y-H, Kim D-W (2010) A new VLSI architecture of parallel multiplier–accumulator based on Radix-2 modified booth algorithm. *IEEE Trans Very Large Scale Integ (VLSI) Syst* 18(2):201–208
24. Tian X et al (2011) Efficient multi-input/multi-output VLSI architecture for two-dimensional lifting-based discrete wavelet transform. *IEEE Trans Comput* 60(8):1207–1211
25. Wu B-F, Lin C-F (2005) A high-performance and memory-efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec. *IEEE Trans Circ Syst Video Technol* 15(12):1615–1628
26. Wu BF, Lin CF A high-performance and memory-efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec. *IEEE Trans Circ Syst Video Technol* 15(12):1615–1628
27. Xiong C-Y, Tian J-W, Liu J (2006) A note on flipping structure: an efficient VLSI architecture for lifting-based discrete wavelet transform. *IEEE Trans Signal Process* 54(5):1910–1916
28. Xiong C, Tian J, Liu J (2007) Efficient architectures for two-dimensional discrete wavelet transform using lifting scheme. *IEEE Trans Image Process* 16(3):607–614
29. Xiong C, Tian J, Liu J Efficient architectures for two-dimensional discrete wavelet transform using lifting scheme. *IEEE Trans Image Process* 16(3):607–614
30. Zhang W et al (2012) An efficient VLSI architecture for lifting-based discrete wavelet transform. *IEEE Trans Circ Syst II: Express Briefs* 59(3):158–162



Gambala Kiranmaye received the B.E degree in Electronics and Communications Engineering from Osmania University, Hyderabad. M.Tech in Very Large Scale Integrated System Design from the Jawaharlal Nehru Technological University Hyderabad, Hyderabad, India. Pursuing Ph.D. in Very Large Scale Integrated Systems & Architectures from the Jawaharlal Nehru Technological University Hyderabad, Hyderabad, India. Presently, she is working as a Associate Professor in the Department of Electronics and Communications Engineering at Guru Nanak Institutions Technical Campus, Hyderabad. She had 4 years of industrial experience as VLSI Design Engineer from ACCEL TECH, Chennai Her current research interests include VLSI Systems Design and Architectures. She publisher 4 papers in various international Journals and 4 papers in international Conferences



Dr. Srinivasulu Tadisetty is a PhD (Embedded Real Time Systems) from Kyushu University, JAPAN and JSPS Gold Medallist, M. Tech (E&I), B.Tech (ECE), DCPA and several diplomas from IISc Bangalore, IIT Kharagpur, CEDTI and NRDC. His research areas are Embedded Systems, Image Processing, Wireless Communication and Digital Signal Processing. He is having around two and half decades of experience in teaching, research and industry. Prior to joining, he undertook prestigious duties as Scientist in Charge of R & D at NIRM, (An autonomous research institute, Government of India) and as a General Manager (R&D) at ICSAL India Limited, Hyderabad. He developed several innovative products /systems for mining, oil & gas and power sectors application for improving productivity and safety. He received Japanese Gold Medal for his outstanding research contributions in the field of Embedded Real Time Systems. He received patent on MBSC system and successfully commercialized. He completed 21S&T and industry sponsored projects as project leader. He was successfully completed several collaborative research projects with IISc Bangalore and Kyushu University, Japan. He was received numerous international fellowships and visited several countries like Japan, Singapore, Hong Kong, Malaysia, Thailand, South Korea, Nepal, USA and soon for attending scientific meetings, chairing technical sessions, delivering invited talks and discussions on collaborative research. He was a visiting researcher to Kyushu University, Japan. He received numerous honours, awards and distinctions. He received merit promotion

too. Received several best paper awards and chaired international conferences in and outside of India. He was a joint secretary to instrument society of India. He is a member of many professional and scientific bodies like ISTE, IEEE, ISOI, IRMTT, and MGMI. To his credit published around 90 publications in international journals and conferences as first author. He delivered expert lectures on 'RFID' during IETE regional conference, Andhra University. He is visiting faculty to Engineers Staff College. He is Editorial committee member of JSPS JAPAN scientific news bulletin 'CONCORD' and Technical reviewer of several international journals like International Journal of Measurements of Elsevier publications, UK, and Journal of Instrumentation and Editor to Journal of Innovations in Electronics and Communication. He is an executive committee member of Indo-Japan Alumina Association. He was guided two PG students dissertation of BITS and several academic projects. He was received Gold Card from National Standard Laboratories of USA for his best research paper. He was successfully completed consultancy projects sponsored by APSEB, HZL, SCCL, BGML, OIL, IOCL, GAIL, NIRM and Gas Malaysia. He is guiding five PhD students under JNTU and other universities. He is a chief project coordinator to AICTE sponsored projects. He was worked as a coordinator to TEQUIP and DSIR programs. He has received international JSPS bridge fellowship. He was ratified as professor & Principal by JNTU Hyderabad. He is a member of Board of Study, JNTU Hyderabad. Now, he is working as a Professor & principal.