Turkish Journal of Computer and Mathematics Education
DOI: https://doi.org/10.17762/turcomat.v9i3.13898 *Vol.09 No.03 (2018), 1090-1099*

Research Article

Investigation on Efficient FPGA Architectures for Image Coding Algorithm

Aditiya Agnihotri

Asst. Professor, Department of CSE (Computer sc)

GEHU-Dehradun Campus

Abstract: In this research, we provide an effective hardware architecture for various image processing, enhancement, and filtering algorithms that is based on FPGAs. The inherent spatial and temporal parallelism in FPGA architecture makes them a popular choice as implementation platforms for real-time image processing applications. The filters are applied by iteratively cycling over an image's pixels using a windowing operator method. Software becomes less effective and real-time hardware solutions are required as picture sizes and bit depths increase. While the findings shown here are for a picture with a resolution of 585 x 450 pixels, the stated method may be used to photos of any resolution, provided that the FPGA memory can accommodate it. The design was developed using the Nexys3 board and Xilinx Spartan-6 FPGA in mind.

Keywords: FPGAs, spatial, temporal parallelism, Xilinx Spartan-6 FPGA, Nexys3

Introduction

A digital picture is defined as a series of values in a two-dimensional space with finite bounds: f(x,y) = x * M * y * N [], where x and y are the spatial coordinates, M and N are the vertical and horizontal bounds [1]. The intensity is indicated by the amplitude at location (x, y). A digital picture is one for which the x, y, and amplitude coordinates are discrete numbers. Pels are the building blocks of digital photos. An illustration of what is meant by the word "pels" is shown in Figure 1. Row index is denoted by the x-coordinate, and column index by the y-coordinate.

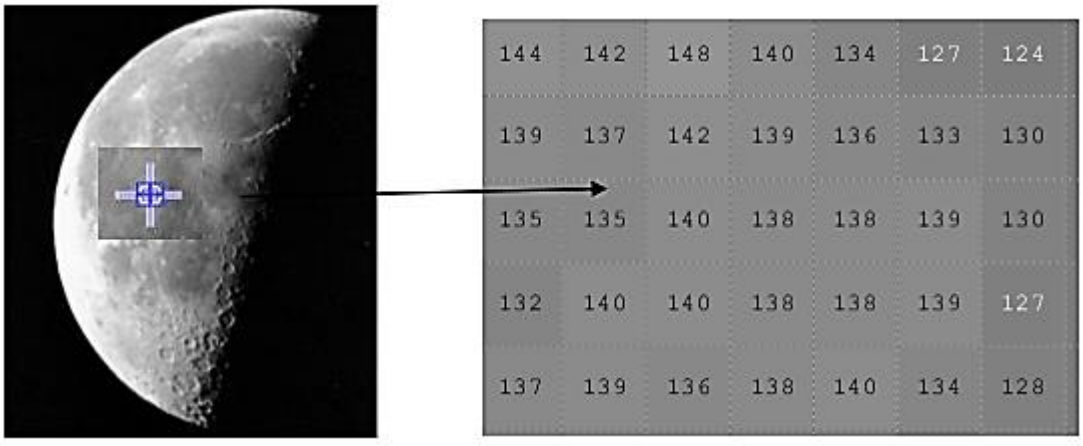

Figure 1. Digital Image Representation

In recent decades, more sophisticated digital photographs have been used[2]. Everyone enjoys taking high-quality photos with their cameras and storing them in their devices for later use. The internet has made it easier than ever to exchange photos with loved ones far and wide. Important documents and letters of authorization may be scanned and transmitted electronically. There was just one illustration for every four pages in the 1911 version of the Britania encyclopaedia. With the rise in popularity of computer-generated imagery, the need for

software that can efficiently handle such pictures on personal computers has only increased since the 1999 version was published. The degree to which a picture may be compressed depends on its specifics. That's why we're always referring back to certain categories of pictures. The following are some potentially useful groupings: As a result, we often make references to various types of symbolism. Some helpful instructions are:

Photos of nature and the family may both be found under the "Natural Images" heading.

Text Images are digital representations of textual and graphical information, such as scanned documents or content generated on a personal computer, such as facsimile pictures.

Scanned images may be anything from a line comic strip to a painting or a computer file including words.

The composite imaging type encompasses the whole spectrum of tri-content formats, including scanned documents.

Image compression

To reduce the amount of space a picture takes up on a computer's hard drive, image compression may be used [3]. It's a concise method of expressing data. The figure 2 encoder and decoder pair compresses and stores or transmits an input picture, and then the decoder restores the original image.

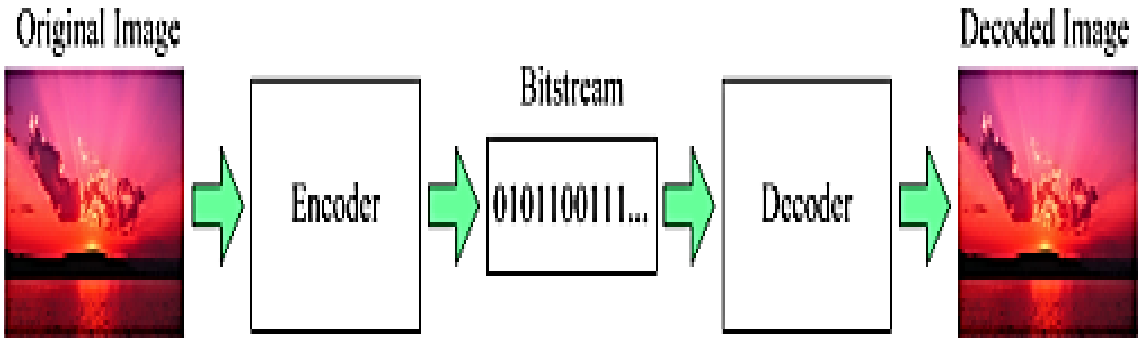

Figure 2 Block Diagram of an Image Compression System

When an image file is processed by an encoder, the result is a bit stream, which is a collection of binary data. After receiving the bit stream [2] from the encoder, the picture is decoded by the decoder. To compress a picture is to reduce the amount of information contained within the bit stream such that the resulting image has fewer bytes than the original.

Need of Image Compression

The massive amount of data meant to be saved and could be transferred is one of the key difficulties faced in image processing applications like video conferencing, uploading and downloading high definition photographs, [3] broadcast television, etc. As compression becomes more widely available, there is a diminishing need to restrict the size of the generated data. Without compression, many applications may be impractical. Compression is a method for creating a smaller representation of a signal in digital form. When data is compressed, the number of bytes needed to uniquely identify it is cut in half. Medical image sizes and their characteristics are shown in Table 1. Table 2 lists several uses for images and videos; some need data compression, while others may get by with only the raw data [4].

Modality	Image dimension	Gray level (bits)	Avg.Number of Images per Exam	Avg.Mbytes per Exam
CT	512 x 512	12	30	16
MRI	256 x 256	12	50	6.5
DRA	1024 x 1024	8	20	20
Ultrasound	512 x 512	6	36	9.5
SPECT	128 x 128	8 or 16	15	0.8 or 1.6
PET	128 x 128	16	50	2
CR	2048 x 2048	12	62	32

Table 1.Radiologic Image Sizes

Multimedia Data	Size/duration	Bits/pixel or bits/sample	Uncompressed Size (in Bytes)	Transmission Bandwidth (in bits)	Transmission time (using a 28.8K modem)
A Page of text	11" X 8.5''	Varying resolution	4-8 KB	32-64 Kb/page	1.1-2.2 sec
Grey scale Image	512 X 512	8 bpp	262 KB	2.1Mb/image	1 min 13 sec
Color Image	512 X 512	24 bpp	786 KB	6.29Mb/image	3 min 39 sec
SHD Image	2048 X 2048	24 bpp	12.58 MB	100Mb/image	58 min 15 sec

Table 2. Data rates for various applications with and without compression

Thus image compression is needed to

- Decrease data quantity needed to represent an image
- Enhance image storage and transmitted capacity.

Quicker methods are needed that can successfully execute the work of compression with less hardware expense for modern uses like multimedia, cellphone exchanges, and the web. The amount of information sent over computer networks is also increasing rapidly. Better images at earlier stages of transmission are therefore an essential part of such codecs. When searching for a picture across a network when the available channel capacity is low, the amount of storage space required and the associated processing cost might be crucial factors. The benefits of compression are not limited to the realm of media. As the transmission, storage, and processing of all digital information plays an increasingly important role in the global economy, a substantial physical infrastructure for data transmission, processing, and storage is required to handle the ever-increasing data quantities. Just as it was more important in old economies to account for transit efficiency, space, and material handling, so too is the accuracy of binary data representation now more economically significant. The term "data compression" refers to both the theoretical foundations and practical implementations of reducing the number of bits required to store or transmit digital information.

Problem Statement

Wavelets are widely used because to their superiority in conveying mean-square error of a non-linear approximation in k maximum wavelet coefficients, making them ideal for displaying one-dimensional signals with limited discontinuities. To achieve simplicity and efficiency when processing a picture, wavelets are often utilised independently on the row and column axes. It causes the signal to be incompletely decorrelated, which manifests as energy coefficient clusters at the picture boundaries. Even after increasing residual independence and imperfect capture using sub-band codes, it may be possible to obtain a transform that avoids such disadvantages in image compression by filtering along the picture's contours. Visual signals need low-bit-rate encoding in the web-based and cellular multimedia domains. The goal of traditional methods of image compression is to use available bits to save the non-zero, quantized transform coefficients of the full picture. The quality of the encoded picture degrades as the compression ratio increases because as the compression ratio increases, greater quantization step sizes are used, reducing the bits per pels. Due to insufficient bit representation of each coefficient, traditional approaches often result in exact blockiness and a few other coding artefacts at lower bit rates; also, the cost of the hardware design is quite expensive and its complexity is greater for conventional wavelet transform.

Literature Survey

Raposo Sánchez et al (2016) proposed a novel formulation based on splines. The suggested method allows the spline order to be treated as a real variable, meaning that it may be utilised in any design methods where integer values are not required. As shown by the provided design examples, the generated approximation errors are far less than those achieved using alternative methods based on traditional windows.

Yongfei Zhang (2016) has advocated for the significance of memory and critical route parameters for 2-D DWT. This research solves this problem by creating a fast, memory-efficient framework for multi-level, two-dimensional discrete wavelet transforms. The 2-D 9/7 DWT processing unit then makes use of a dual data scanning approach for its lifting operations, thereby doubling its cycle throughputs. Furthermore, the suggested Row Transform Unit and Column Transform Unit for 2-D DWT architecture make advantage of input sample availability and offer computational resources correspondingly to optimise the processing performance, hence reducing hardware costs. Third, a parallel multi-level architecture has been built using multiple proposed 2-D DWT units to perform up to six 2-D DWT levels at a resolution level all the way to any arbitrary image size at competitive hardware cost. This addresses the issues of high memory cost for immediate computing results from each level and computation time as resolution level increases.

D. Venugopal (2015) has proposed a lossless block-based medical picture compression method. Because of their value in illness detection, medical pictures need a straightforward and effective compression method. In order to compress images without losing quality, this work suggests a quick and less difficult approach based on the Hadamard transform and Huffman encoding. To eliminate the intra-block correlation, the input picture is first deconstructed using the Integer Wavelet Transformation (IWT), and then the LL subband is changed using the lossless Hadamard Transformation (LHT). To get rid of neighbouring block correlation, we apply a technique called direct current prediction (DCP). The threshold for the Nontransformed block (NTB) is used to verify the presence of non-LL sub-bands. The relevance of this procedure lies in the fact that it is a simple DCP that results in a valid NTB after truncation. Encoding is either done immediately or after LHT transformation and truncation, depending on the findings of the NTB. At last, a Huffman encoder that compresses data is used on both coefficients. Simulation results show that the proposed technique outperforms well-established lossless compression algorithms like JPEG 2000 in terms of compression ratio. The technique has been extensively tested with both commonplace and medical photos, and it returns optimal compression ratio values in both cases.

Chih-Hsien Hsia (2013) memory parameters (for storing intermediate signals) and critical routes are crucial for 2-D (or multi-dimensional) transformations, and this paper presents a memory-efficient hardware architecture of 2-D dual-mode lifting-based discrete wavelet transform. In this research, we provide novel algorithms and hardware designs for the lifting-based discrete wavelet transform (LDWT) in 2-D dual-mode (supporting 5/3 lossless and 9/7 loss-coding). In order to facilitate extremely large-scale integration, LDWT proposes a 2-D dual-mode architecture that benefits from low transposition memory (TM), low latency, and frequent signal flow. Both LDWT 2-D 5/3 and LDWT 2-D 9/7 call for a TM of 2N, whereas LDWT 2-D 9/7 calls for a TM of 4N. The suggested hardware design requires a smaller lifting-based TM than earlier architectures, according to the comparison findings. JPEG2000, motion-JPEG2000, MPEG-4 static texture object decoding, and wavelet-based scalable video coding applications are just some of the visual tasks that may now be performed in real time.

Mohanty & Meher (2013) proposed a convolution-based generic framework for performing 3-level 2-D Daubechies and bi-orthogonal DWT computations with minimal memory use. A structural buffer is unnecessary for the design. Although it achieves considerable reductions in space, power, and memory size/time, lifting structures have the upper hand since they only need a limited number of arithmetic components. This approach avoids the need for a frame buffer by computing the DWT level at the same time. Parallel data access is used to simplify memory access on each DWT level.

FPGA Implementation of Fast Lifting Wavelet Transform for Lossless Image Compression

Time- and frequency-domain analysis is provided via the wavelet transform. There are two primary methods for creating and using wavelet transformations. Space domain and frequency domain analysis rely on these techniques. The Lifting Scheme refers to the frequency-based approach used by Filter Banks (FB). Discrete-Wavelet Transformation (DWT) has gained widespread use in recent decades as a signal-processing method. It has found widespread use in the analysis of signals and visual data. DWT is widely used nowadays for image compression because it makes it possible to transmit high-quality low-resolution images. Convenient tools for adjusting and focusing on images. FIR filter bank architectures or convolution have typically offered master performance matched to the current JPEG standard [2], which is why they form the basis of the most recent JPEG2000 image compression technology. These methods aren't ideal for high-speed processing since they demand a lot of data storage and a variety of mathematical operations. The wavelet may be lifted or transformed using this unique technique. The main characteristic of the DWT lifting system is that the execution of the HP and LP filters are transformed into multiplications of banded matrices by dividing them into an order of up-low triangular matrices. The DWT based on convolution requires a lot more maths than this structure does [3]. Recent developments in technology have allowed the widespread implementation of DWT on FPGA and DSP devices. In a pipeline, the clk speed is determined by the number of multipliers present at each stage of the process. The lifting structure is a sequence of these multipliers and other structural processing components. According to [4], the main difficulties in designing 1D-DWT hardware designs are related to the processing speed and the number of multipliers, whereas the memory issue in 2D-DWT is dominated by hardware costs and architectural complexity.

General Hardware Implementation of 5/3 Lifting Wavelet

This connection relies on the employment of a single-port picture memory. Off-chip memory is the typical for the memory picture. In any event, the on-chip option should also be considered, in case we need to deal with a large number of FPGAs or need to manage a large number of tiny edges. Given that image memory is sometimes located on-chip, we ensure a subtle link between the usual clk signal for image memory and the remaining frame data. Due to the need for the same number of sheets regardless of memory capacity, multi-port memory squares are not included in our evaluation. Furthermore, although the picture memory is often off-chip, it is noteworthy to note that single off-chip RAM ports use less energy on access than dual-port RAMs. Memory becomes highly relentless in information-scaled algorithms like 2D discrete wavelet transforms. Off-chip picture

memory uses less power, thus multi-port RAMs with better performance may be turned off. This is why single-port RAMs are ideal. Two memory, one for accumulating the input picture and another for accumulating the output, are often employed to get the image ready for display. We have employed a mapping design setup to avoid the second unit; the channel's output is made up of a memory material that has been consumed and is no longer needed. For this layout to work, the width of each memory section must scale with the output variation. The procedure for creating a map is outlined. Theoretically, an 8x8 picture is linked to the dyadic decomposition, and the results of each step are combined in previously searched and partitioned memory regions. In Fig. 1, the shaded areas correspond to the rough picture's pixels at the contribution of each level. Figure.1 depicts the components of the FPGA-based lifting wavelet transform hardware implementation: a memory unit, a discrete wavelet transform (DWT) 5/3 filter, an FSM-based control route, and a chip buffer.

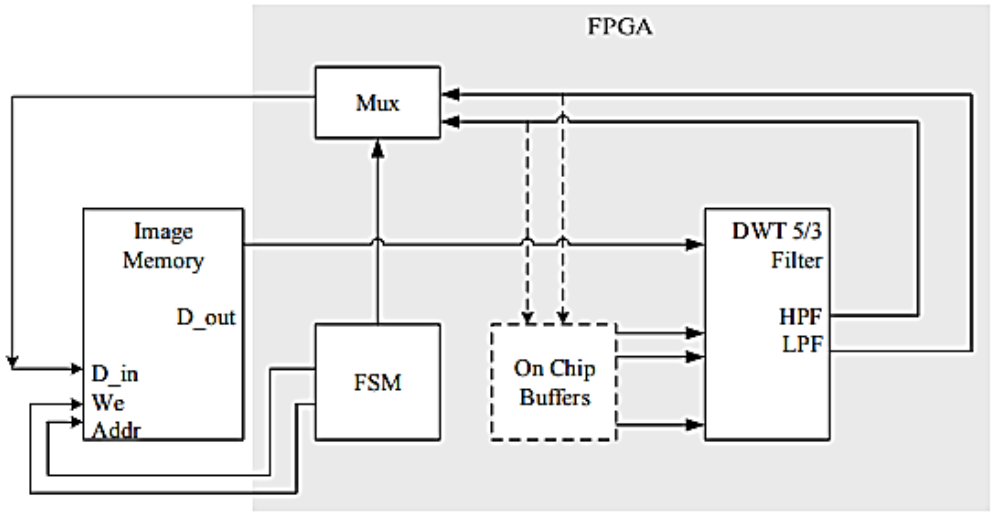

Figure.1. Implementation of Lifting Wavelet in FPGA

This reduces the DWT filter's memory use and critical path latency. Accumulating the coefficients of the intermediate level of filtering in chip buffers enhances memory access and computational performance in comparison to other scanning methods.

Figure 4.3 depicts the proposed architecture of the 2D-DWT 5/3 lift system. - DWT generates four subbands, i.e. LL, LH, HL, and HH, which are linked to the 70 temporary processor for conversion. The internal dividing, predicting, and updating architecture is shown in Figure 2 (a), Figure 3 (b), and Figure 4 (c). Both vertical and horizontal filtering may be accomplished using the same filter. The primary benefit of row-column architecture is its reliance on single-port read-only memory. When compared to other filter approaches, it requires fewer lifting stages, is simpler to construct, and minimises hardware cost and critical path time delay. Because of its symmetry and orthonormality.

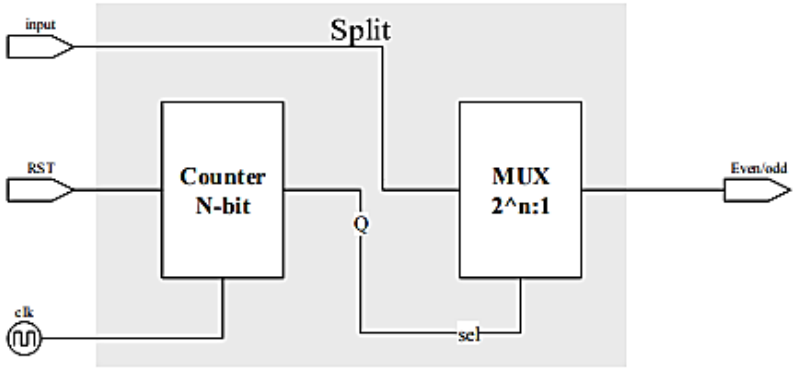

Figure 2(a) Internal Architecture for Split Unit

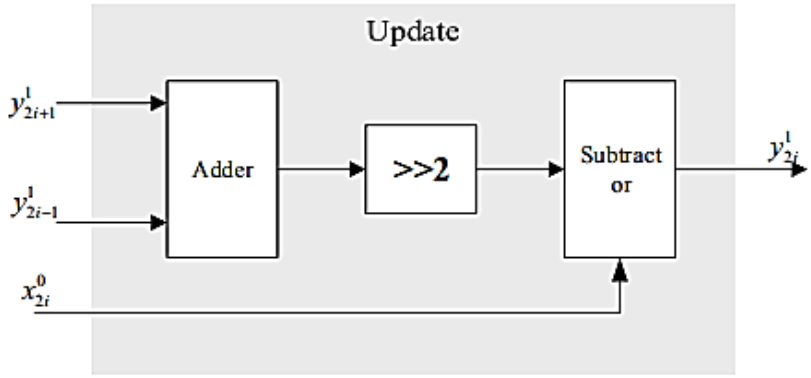

Figure 3 (b) Internal architecture of Update Unit

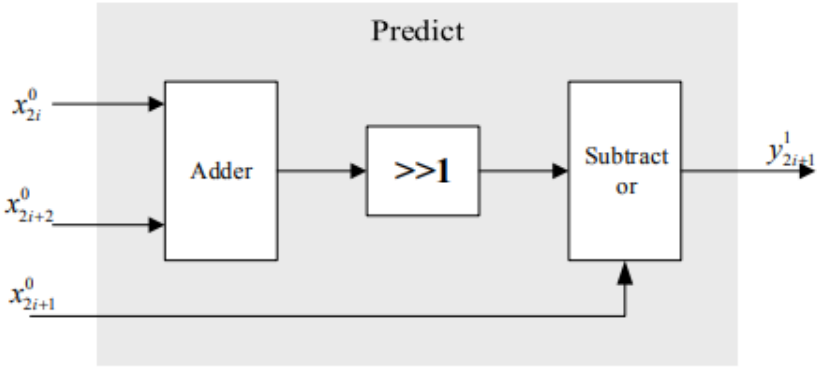

Figure 4. (c). Internal Architecture of Predict Unit

Implementation Results

Tables 1 and 2 provide the hardware specifications and utilisation overview for COIF 9/7. A proposed 1D-processor calls for 8 adders and 4 SALs, with a 20-register latency. Tm When compared to the suggested NxN design, it has an essential route delay for handling the image control path of two N2/2 cycles. The recommended solution is quite similar to the original computation, but with just one multiplication. Therefore, the 1-D processing element cannot be folded, but the 1-D processing component proposed for the scheme can be readily folded, resulting in a reduction of the information route to only two adders and a multiplier without affecting the

important path. Equally important for route delay, the suggested better lifting architecture of needs 2 N clock cycles.

Row Filter	Adder	38
	Subtractor	4
	Register	60
Column Filter	Adder	38
	Subtractor	4
	Register	17
Scaling	Adder	9
	Subtractor	0
	Register	6
Total	Adder	85
	Subtractor	8
	Register	112

Table 1: Hardware Requirements of 2D DWT

The suggested 1D-DWT structure has a substantially bigger route latency and superior outcomes compared to the EFA [46]. As can be seen in Figure 6, the suggested 2D-DWT architecture consists of 31 adders, 8 subtractors, and 4 scaling operations. To complete 2D-DWT processing, a total of 16 components are required, two of which are employed in each processing element. Six SAL and eight extra modules may have the DSFA installed without causing any problems along the lifeline.

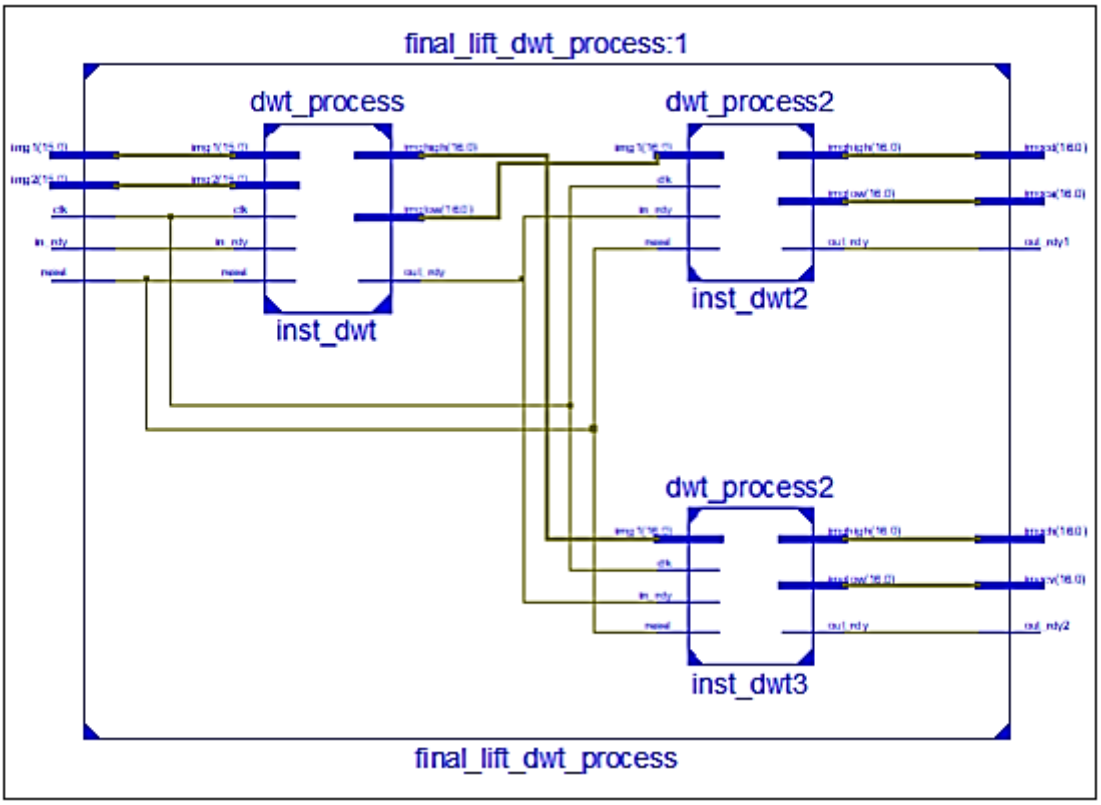

Figure 6. RTL View of COIF9/7 DWT One Level Decomposition

When compared to RAM [82, 83] dual-scan architectures [84], the reciprocal has a shorter critical path time. If the DSFA (a critical back-path delay) is equal to the DSFA, as stated in [22], then the buffer size is 11N. Despite the drawbacks of a single line scan (resulting in a slower throughput rate and more copulation cycles), the parallel lifting concept described in [27] is intended to decrease the time required to flip a coin. While the DSFA only needs 5 transposition registers, the parallel lifting framework's design necessitates a lengthy 1.5N buffer size. High-speed architectures are superior than the suggested device, but they have a lengthy Ta path due of the need to constantly check and update the same hardware. To cut down on device cycles, [16] proposes an architectural fix that boosts parallelism. The use of a 4N time buffer and a crucial path delay of 2Tm and 4Ta in a two-dimensional discrete wavelet transform (DWT) lifting architecture has recently been proposed as a memory-efficient solution. The proposed DSFA makes use of the same amount of storage space as is recommended in, but it has shorter computation times, reduced latency, and a shorter critical path. [17] suggests a DSFA (N+3) structure with a delay of (3/2)N+3 cycles for multiplying and accumulating. The suggested 2D DWT DSFA for a COIF9/7 filter is realised in the Xilinx Spartan 6 xc6slx45t-3-fgg484 FPGA board. The proposed FGPA design employs a total of 17694 input search tables, and statistics on its implementation reveals a critically important 5.6 ns route latency.

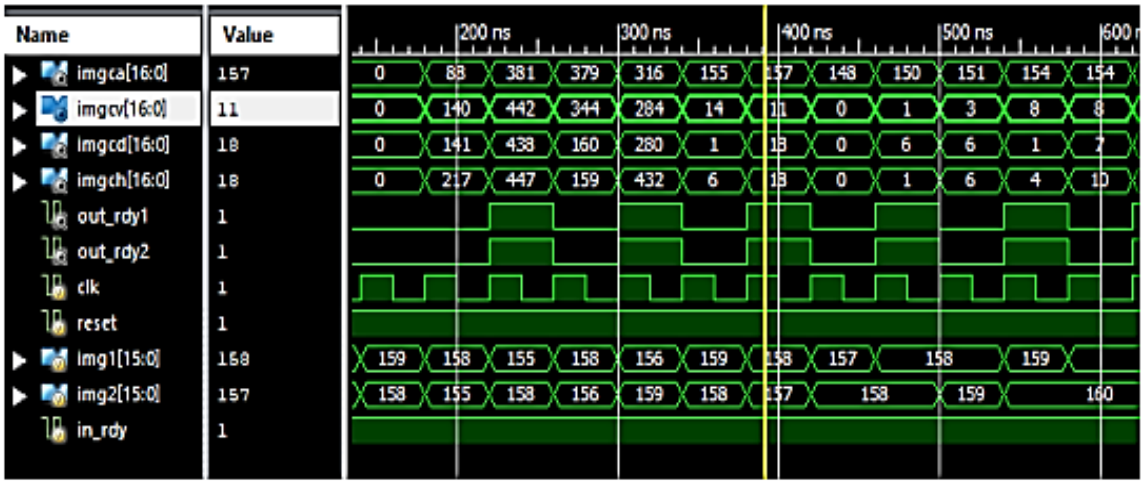

Figure7..Subband Decomposed Output Image Data

Conclusion

We looked at the novel 1D and 2D-DWT flip-based structures proposed for a CDF 9/7 lifting filter in this Proposed study. In addition to suggesting an off-chip 2D DWT multiplier-less design, the proposed DSFA is on par with the best of the current hardware in terms of complexity, path elimination, memory size, and flow architecture. The layout is optimised for fast processing of huge picture datasets. We provide a three-input DWT design. In order to rearrange the vertical calculated value in the transpose unit, the CDF9/7 filter only needs a 4N buffer and five transpose registers instead of a NN inlay picture with a single-level 2D-DWT decomposition. The evaluation results show that the proposed 2D-DWT architecture has lower hardware cost and internal memory needs than existing aware systems with the same throughput rate. Hardware adders and shifters may be used in lieu of multipliers to reduce the lag time. As a result, the new design is more competent in hardware and achieves a critical latency of around a second, making it superior than preexisting parallel systems. It has been determined that the DSFA is a symmetrical high-speed design with minimal hardware requirements. The RAM cap has also been lowered. Spartan 6 with a 148MHz clock frequency is used to implement the hardware design.

References

1. Lars Asplund. Vunit. URL https://vunit.github.io/.

2. CCSDS 121.0-B-2. Recommendation for space data system standards, lossless data compression. Standard, CCSDS Secretariat, Space Operations Mission Directorate, NASA Headquarters, Washington, DC 20546-0001, USA, May 2012.
3. L. Chen, L. Yan, H. Sang, and T. Zhang. High-throughput architecture for both lossless and near-lossless compression modes of loco-i algorithm. IEEE Transactions on Circuits and Systems for Video Technology, pages 1–1, 2018. ISSN 1051-8215. doi: 10.1109/TCSVT.2018.2881040.
4. H. Daryanavard, O. Abbasi, and R. Talebi. Fpga implementation of jpegls compression algorithm for real time applications. In 2011 19th Iranian Conference on Electrical Engineering, pages 1–4, May 2011.
5. Mohamed A. Abd El ghany ; Aly E. Salama ; Ahmed H. Khalil. Design and implementation of fpga-based systolic array for lz data compression. Proc. of IEEE International Symposium on Circuits and Systems (ISCAS'07), pages 3691–3695, 2007.
6. P. G. Howard and J. S. Vitter. Fast and efficient lossless image compression. In [Proceedings] DCC '93: Data Compression Conference, pages 351–360, March 1993. doi: 10.1109/DCC.1993.253114.
7. David A. Huffman. A method for the construction of minimum-redundancy codes. Proc. IRE, 40(9):1098–1101, 1952.
8. L. Kau and S. Lin. High performance architecture for the encoder of jpeg-ls on sopc platform. In SiPS 2013 Proceedings, pages 141–146, Oct 2013. doi: 10.1109/SiPS.2013.6674495.
9. Erik G. Larsson. Signals, Information and Communication. LiU-Press, Linköping, 2016.
10. D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard. IEEE Transac tions on Circuits and Systems for Video Technology, 13(7):620–636, July 2003. ISSN 1051-8215. doi: 10.1109/TCSVT.2003.815173.
11. D. Salomon; G. Motta. Handbook of Data Compression. Springer, London, 5 edition, 2010. ISBN 978-1-84882-902-2. With Contributions by David Bryant.
12. M. Papadonikolakis, V. Pantazis, and A. P. Kakarountas. Efficient highperformance asic implementation of jpeg-ls encoder. In 2007 Design, Automation Test in Europe Conference Exhibition, pages 1–6, April 2007. doi: 10.1109/DATE.2007.364584.
13. Teledyne DALSA Inc. Patrick Sicard. Lossless data compression and decompression apparatus, system, and method, 2016.
14. B. Rusyn, O. Lutsyk, Y. Lysak, A. Lukenyuk, and L. Pohreliuk. Lossless image compression in the remote sensing applications. In 2016 IEEE First International Conference on Data Stream Mining Processing (DSMP), pages 195–198, Aug 2016. doi: 10.1109/DSMP.2016.7583539.
15. K. Sayood. Introduction to Data Compression. Morgan Kaufmann, 4 edition, 2012. ISBN 978-0-12-415796-5.
16. Operating Instructions Ranger3 3D Vision. SICK AG.
17. Roman Starosolski. Simple fast and adaptive lossless image compression algorithm. Software: Practice and Experience, 37(1):65–91, 2007. doi: 10. 1002/spe.746. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.746.
18. H. Daryanavard ; O. Abbasi ; R. Talebi. Fpga implementation of jpeg-ls compression algorithm for real time applications. Iranian Conference on Electrical Engineering, 19:1–4, 2011.
19. T. Tsai, Y. Lee, and Y. Lee. Design and analysis of high-throughput lossless image compression engine using vlsi-oriented felics algorithm. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 18(1):39–52, Jan 2010. ISSN 1063-8210. doi: 10.1109/TVLSI.2008.2007230.
20. M. J. Weinberger, G. Seroussi, and G. Sapiro. The loco-i lossless image compression algorithm: principles and standardization into jpeg-ls. IEEE Transactions on Image Processing, 9(8):1309–1324, Aug 2000. ISSN 1057-7149. doi: 10.1109/83.855427.
21. X. Wu and N. Memon. Context-based, adaptive, lossless image coding. IEEE Transactions on Communications, 45(4):437–444, April 1997. ISSN 0090- 6778. doi: 10.1109/26.585919.
22. Zynq-7000 SoC Data Sheet: Overview. XILINX, 7 2018. v1.11.1.