Exploiting Spatial Redundancy and Approximate Computing for Area Efficient Image Compression

SAURAV ARJUN
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

LUND
UNIVERSITY

Master Thesis

Exploiting Spatial Redundancy and Approximate Computing for Area Efficient Image Compression

Author:
Saurav Arjun, sauravarjun27@gmail.com

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University

January 24, 2022

Exploiting Spatial Redundancy and Approximate Computing for Area Efficient Image Compression

Department of Electrical and Information Technology
Lund University

Saurav Arjun
sauravarjun27@gmail.com

Main Supervisor: **Joachim Rodrigues**
joachim.rodrigues@eit.lth.se

Supervisor: **Ali Shami**
ali.shami@arm.com

Examiner: **Pietro Andreani**
pietro.andreani@eit.lth.se

January 24, 2022

Printed in Sweden
Tryckeriet i E-huset, Lund

Acknowledgement

This exciting journey of the thesis filled with ups and downs has been credibly overcome by the amazing supportive people around me. It would have been impossible for me without the contribution of the Almighty and my spiritual guru AC BhaktiVedanta Srila Prabhupada.

I extend my deepest gratitude to Henrik Ohlsson for believing in me throughout. I wish to show my sincere appreciation to my supervisor at Arm, Ali Shami. His consistent guidance assured that I focus on the right direction during the course of this thesis which nurtured my belief in the completion of the project.

I convey my special indebtedness to my professor and my academic supervisor, Joachim Rodrigues who encouraged me to be professional and competent throughout my thesis work. The contribution and the valuable comments from my examiner at Lund University, Pietro Andreani is truly appreciated.

I would like to thank all my colleagues and friends who directly or indirectly helped shape this thesis project.

Last but not the least, I express my deepest gratitude to my parents and family for their unfailing support and continuous encouragement throughout my years of study.

Abstract

Owing to the intensive computation involved in the Discrete Cosine Transform during image compression, the design of the efficient hardware architectures for fast computation of the transform has become imperative, especially for real-time applications. Although fast computation techniques have been able to minimise the hardware computation complexity to a certain limit, they could further extend the research to figure out the interesting approaches which can be implemented on applications where power, speed and area are crucial factors to determine the performance of the system.

This thesis work is an attempt towards implementing a novel approach to provide image compression with low area and power requirement . Various reduced computational compression algorithms were proposed by exploiting hardware efficient image compression algorithms. Furthermore, to understand and compare their performances, the concepts of spatial redundancy and approximate computing in images are exploited. The work designs a number of hardware efficient image compression algorithms.

In this thesis work, the models try to group the pixel data by taking the image's feature space similarity and spatial coherence characteristics into consideration. These models have been tested successfully on a wide range of images, including black and white images and coloured images. The proposed architectures in this paper bring forth equal or higher image performance with higher compression ratio with less hardware requirement. These architectures are also compared among each other to provide an understanding on design-space exploration.

Popular Science Summary

Despite the advances in semiconductor technologies and the development of energy-efficient design techniques, the overall energy consumption of computer systems is still growing at an alarming rate in order to process an ever-increasing amount of information. It is essential to dramatically improve the energy efficiency for these emerging workloads to keep up with the growth of information.

Approximate computing [1][2] or inexact computing trades off computation quality with the effort expended. As rising performance demands confronting with plateauing resource budgets, approximate computing has become not merely attractive, but even imperative. It is one of the ways which is gaining popularity for applications where accuracy is not very important. By compromising on accuracy, engineers can achieve better power/energy, performance or area efficiency for such applications. Researchers have applied inexact computing techniques at an algorithmic level as well as at a circuit-level to improve the power, performance and area numbers. Many authors have also described imprecise adders for low-power approximate computing applications including image processing [3].

Image compression is a process of reducing the size of the representation of the graphics file in binary format without affecting the quality of the image to an objectionable level. This reduction helps to store more images for the same amount of storage device. It also decreases the transmission time for the images to be sent over the various technologies like internet [4]. The discrete cosine transform (DCT) which is the most widely used technique for image compression was initially defined in [4]. The DCT can be used to convert the signal (spatial information) into numeric data ("frequency" or "spectral" information) so that the image's information exists in a quantitative form that can be manipulated for compression.

Table of Contents

List of Figures

List of Tables

List of Acronyms

TPU	Tensor Processing Unit
PSNR	Peak Signal to Noise Ratio
DCT	Discrete Cosine Transform
ALU	Arithmetic Logic Unit
GPU	Graphics Processing Unit
LUT	Look Up Tables
PNG	Portable Network Graphics
GIF	Graphic Interchange Format
IDCT	Inverse Discrete Cosine Transform
DSP	Digital Signal Processor
MSE	Mean Square Error
SSIM	Structured Similarity Index Method
MS-SSIM	Multi Scale Structured Similarity Index Method
MSD	Mean Square Deviation
IC	Integrated Circuits
TFP	Twiddle Factor Precision
DIP	Digital Image Processing

Chapter 1

Introduction

The performance of various computing systems, from sensors, to smartphones, other mobile devices to servers, supercomputers, to cloud computing data centers, has been increasing dramatically in the past several decades in line with the advances in the IC design according to the famous Moore's Law. However, as Moore's Law is approaching its limit [5], the conventional techniques are unable to further improve the computing performance of systems with limited power budget, i.e., the power consumption restricts the performance of computing systems. It becomes challenging to continue improving the system performance by conventional CMOS technologies.

Due to the error-resilient and fault-tolerant ability of the human brain, visual and auditory systems, certain level of processing errors will not affect the quality of human perception and recognition of the processed data [6][7]. Examples of such instances have been reported in artificial intelligence, machine learning, data mining, multimedia signal processing [7][8][9][10] etc. In these applications, the data includes noisy or redundant information, and therefore it makes little sense to compute the precise result based on erroneous data or perform redundant computation.

It is clear that rising performance demands will soon outpace the growth in resource budget and hence, over-provisioning of resources alone will not solve the conundrum that awaits the computing industry in the near future. A promising solution for this dilemma is approximate computing and storage, which is based on the intuitive observation that while performing exact computation or maintaining peak-level service demand require high amount of resources, allowing selective approximation or occasional violation of the specification can provide disproportionate gains in efficiency.

Digital image compression has been the focus of a large amount of research in the recent years. As a result, image compression methods grow as new algorithms or variations of the already existing ones are introduced. In order to utilize the digital images effectively, specific techniques are needed to reduce the number of bits required for their presentation. It has led to an instant growth in the area of Digital Image Processing. Image compression is not only concentrated on reducing size but also concentrated on doing it without losing quality and information of the image. An image is essentially a 2-D signal processed by the human visual system. The signals representing images are usually in analog form. However, for processing, storage and transmission by computer applications, they are converted from analog to digital form.

1.1 Thesis Structure

Recent implementations have been focused on area and power to a considerable extent. There are a few of them which seem to achieve low area with low power. The goal of the thesis is to develop few algorithms for image compression hardware accelerator with high throughput and high image quality using less hardware resources at an acceptable average memory compression rate. The entire work of the thesis has been organized into six different categories.

- **Introduction:** To provide the reader with brief discussions on the overview of the thesis. The chapter is mainly to discuss about the ideas, previous work, motivation and purpose of the work accompanied.
- **Background:** The aim of the chapter is to explain the theoretical concepts related to the topics of the thesis. Along with this, discussions on different metrics are taken into consideration to analyse the image quality.
- **Case Studies:** Various experiments are performed in this chapter including the characteristics of approximate computing, image segmentation and spatial redundancy. Proposal of different image compression algorithms and their respective image performances in terms of memory, computational complexity, hardware consumption and image quality has been presented.
- **Results:** The chapter is to expand all the developed models with their respective results. The characteristics of implementations is studied thoroughly to help the reader to understand various scenarios. The outcome of the results have been presented in the form of tables.
- **Analysis:** The reader is presented with more detailed interpretation of the outcomes of the proposed algorithms. Graphical representations of various results were studied and compared among each other. Discussions on the advantages and disadvantages have been taken into consideration in this chapter.
- **Conclusions:** Summary of the entire work performed in the thesis has been discussed in this chapter along with the future prospects of the proposed architecture.

In order to achieve such efficient hardware, output results of such algorithms were made in terms of various metrics and several comparisons were summarized to clearly describe the design space exploration.

The neighbouring pixels in an image show a tendency of being highly correlated to each other. Such theories along with coding redundancy between nearby pixels, blocks, images are exploited in-depth. Ideas where approximate computing techniques can be applied on images to develop different model have been discussed. Various advanced hardware efficient Discrete Cosine Transform (DCT) approaches used in modern days have been explored and their behaviour have been studied meticulously. DCT features were studied to investigate similarity between each neighbouring pixels and also for an entire image information. These have helped in designing a hardware-efficient DCT. Analysis on several sets of images have been performed to further develop image compression algorithms.

Chapter 2

Background

Reduced precision computation for approximate computing is a technique that represents variables and data structures in a program with fewer bits (compared to conventional integer and floating-point numbers). This allows utilization of less expensive and more energy-efficient hardware to perform the reduced precision computation using Arithmetic Logic Units (ALUs). The area and power of ALUs roughly scale quadratically with bit width and therefore, using reduced precision hardware enables packing significantly more ALUs within the same area or power envelope. These benefits are especially noticeable in accelerator architectures, such as GPUs and coarse-grained reconfigurable architectures, where a significant fraction of the area is occupied by these ALUs.

2.1 Data Analysis

Data Analysis is a process of collecting, transforming, cleaning and modelling data to discover the required information. The results so obtained are communicated, suggesting conclusions and supporting the decision-making. Data visualization is at times used to portray the data for the ease of discovering the useful patterns in the data.

2.2 Image Compression

Image compression addresses the problem of reducing the amount of information required to represent a digital image. It is a process intended to yield a compact representation of an image, thereby reducing the image storage transmission requirements. Every image will have redundant data. The duplication of pixel data in an image provide redundant information. Mainly it is the repeating pixels across the image or pattern, which is repeated more frequently in the image.The image compression occurs by taking benefit of redundant information in the image. Reduction of redundancy helps to minimise storage space of an image. Image compression is achieved when one or more of these redundancies are reduced or eliminated. In image compression, three basic data categories are identified and exploited. Compression has been achieved by involving one or more of the three basic data redundancies.

2.2.1 Spatial Redundancy

There is high spatial coherency between pixels in an image space, which means it is highly probable that between two pixels having identical or similar colours we will find another pixel having the same (or similar) colour. The characteristic of the image is exploited in image compression techniques [11]. Although spatial coherence is not a constraint explicitly built-in, each cluster in the feature space is expected to group pixels on the grounds of their homogeneous properties which will come from a coherent region in the image. The neighbouring pixels of an image are not statistically independent due to the correlation between them. This type of redundancy is called Spatial redundancy or sometimes also called inter-pixel redundancy. This redundancy can be explored in several ways, one of which is by predicting a pixel value based on the values of its neighbouring pixels. If the original image pixels can be reconstructed from the transformed data set, the mapping is said to be reversible [12].

2.2.2 Coding Redundancy

Coding redundancy consists of using variable-length codewords selected to match the statistics of the source, in this case, the image itself or a processed version of its pixel values. This type of coding is always reversible and usually implemented using Lookup Tables (LUTs). Examples of image coding schemes that explore coding redundancy are the Huffman codes and the arithmetic coding technique.

2.2.3 Lossless Compression

This technique compresses the image data by encoding all the information from the original image, allowing the data to be completely reconstructed from the compressed file resulting in it to be identical to the original image. Lossless [13] image compression are image formats such as Portable Network Graphics (PNG) and (Graphics Interchange Format) GIFs. To produce shorter output data from the original data, most lossless compression programs first generates a statistical model for the input data, and secondly use the model to map input data to bit sequences. Some of the most common lossless compression algorithms are Entropy encoding, Huffman coding, Lempel-Ziv compression, Lempel–Ziv–Welch, Zstandard, Prediction by partial matching and Run-length encoding.

2.2.4 Lossy Compression

These types of compression techniques are mainly used in images. The compressed image is similar to the original uncompressed image but unlike lossless compression, some information concerning the image is lost. This introduces vulnerability to the performance of the reconstructed images, demanding the quality of the image for lossy compression to be analysed. The most common examples of lossy compression are Moving Picture Experts Group (MPEG) & Joint Photographic Experts Group (JPEG). The lossy compression technique provides a much higher compression ratio than lossless compression. The major performance schemes during a lossy compression techniques include:

- compression ratio
- signal to noise ratio

- operation time of encoding-decoding.

2.3 Discrete Cosine Transform

Discrete cosine transform (DCT) has been widely used to convert a dynamic signal into frequency components to reduce digital image storage size, expedite data transmission and remove redundant information. DCT is closely related to discrete Fourier transform with the advantage of concentrating the energy of the transformed signal in low-frequency range where human eyes are less sensitive in image processing [14]. The joint ISO committee adopted DCT to JPEG international standard of 8 x 8 block size to reduce the blocking effect in image compression. A basic JPEG image encoding is composed of three procedures: image transform, quantization, and encoding. DCT can map original data into the frequency domain by cosine waveform and conversely IDCT transfers frequency domain data into the spatial domain.

Numerous coding methods based on DCT have been presented for digital image processing; however, the associated memory size, bandwidth, and safety issues are of significant concern to real-time applications. Sun and Yang [15] proposed an image compression method based on a Laplace transparent composite model to achieve high coding efficiency. Jridi et al. [16] presented image compression hardware to reduce computational complexity. Others have proposed to optimize image computation by DSP. Kumbhare and Gokhale [17] developed a low complexity architecture for computing an algebraic integer-based 8-point DCT in digital image processing. Jridi et al. [18] designed a low complexity DCT engine in digital video and image processing. Sub-band decomposition algorithms based on DCT have also been used in transmitting image data of low resolution to the rebuilt image of better quality [19][20][21], but they are required high complexity and thus time-consuming computation. Stassen's matrix multiplication algorithm was proposed to reduce complex matrix multiplication in DCT [22]. Khan et al. [23] increased the coordination between the pixel size and subword size to maximize resource utilization for multimedia applications, but the work required heavy computation.

2.4 Image Quality Metrics

Image quality analysis is the science of analysing and comparing the characteristics of an image concerning the original image of predetermined/preset standards. The analysis can be performed subjectively as well as objectively. In subjective analysis, the measures of image quality are evaluated by human beings. The main disadvantage of this method is that it is highly inconvenient, sluggish and inaccurate, whereas objective methods use computerized algorithms to compute image quality. This is the reason for the development of objective image quality (IQA) which predicts the quality of the image automatically. Therefore, objective analysis plays an important role in determining image quality. There are so many image quality techniques largely used to evaluate and assess the quality of images such as Mean Square Error (MSE), Universal Image Quality Index (UIQI), Peak Signal to Noise Ratio (PSNR), Structured Similarity Index Method (SSIM), Human Vision System (HVS), Feature Similarity Index Method (FSIM), Multi-Scale SSIM (MS-SSIM) etc. In this project work, four different metrics namely MSE, PSNR, SSIM and MS-SSIM methods have been considered to analyse the behaviour of the image. MS-SSIM and SSIM are two most closely

modeled human observations [24]. MSE and PSNR are the most common methods used in the image processing research field.

2.4.1 Mean Square Error

Mean Square Error (MSE) is the most common estimator of image quality measurement metrics. It is compared with the input image against a pristine reference image with no distortion metric and the values closer to zero are the better. The variance of the estimator and its bias are both incorporated with mean squared error. The MSE is the variance of the estimator in the case of an unbiased estimator. It has the same units of measurement as the square of the quantity being calculated like variance. The MSE measures the average of the square of the errors. The error is the difference between the estimator and the estimated outcome.

2.4.2 Peak Signal to Noise Ratio

Peak Signal to Noise Ratio (PSNR) is used to calculate the ratio between the maximum possible signal power and the power of the distorting noise which affects the quality of its representation. This ratio between two images is computed in decibel form. The PSNR is usually calculated as the logarithm term of the decibel scale because the signals have a very wide dynamic range. This dynamic range varies between the largest and the smallest possible values which are changeable by their quality. The PSNR is the most commonly used quality assessment technique to measure the quality of reconstruction of lossy image compression codecs. The signal is considered as the original data and the noise is the error yielded by the compression or distortion. The PSNR is the approximate estimation of human perception of reconstruction quality compared to the compression codecs. In image and video compression quality degradation, the PSNR value varies from 30 to 50 dB for 8-bit data representation and from 60 to 80 dB for 16-bit data. In wireless transmission, the accepted range of quality loss is approximately 20 - 25 dB [25].

2.4.3 Structured Similarity Index Method

Structured Similarity Index Method (SSIM) is a perception-based model. In this method, image degradation is considered as the change of perception in structural information. It also collaborates with some other important perception-based facts such as luminance masking, contrast masking, etc. The term structural information emphasizes the strongly inter-dependant pixels or spatially closed pixels. These strongly inter-dependant pixels refer to some more important information about the visual objects in the image domain. Luminance masking is a term where the distortion part of an image is less visible in the edges of an image. On the other hand, contrast masking is a term where distortions are also less visible in the texture of an image.

2.4.4 Multi Scale Structured Similarity Index Method

The active SSIM algorithm is a single scale method. The method which is more flexible than the other single scale methods is the multi-scale structural similarity measure. This multi-scale method image details with different resolutions can be included. Lowpass filtering and downsampling are the two main operations used in this multi-scale structure similarity method. The original and

the distorted or noisy images are iteratively low-pass filtered and then downsampling will be done on that by a factor of 2. For this multi-scale operation, the original image is taken as scale 1. The highest scale is for example scale M so a total of M-1iterations are taken place. In the SSIM method, three comparisons have been done i.e., contrast comparison, luminance comparison and structure comparison, similar to that multi-scale structure similarity also have three comparisons. Luminance comparison is performed on scale M. Other two comparisons are performed on the intermediate scale and after all these the final quality measurement metrics is the combination of these three comparisons, so one can say that this is a more convenient image quality metric than the other single scale methods.

Chapter 3
Case Studies

Higher correlation between the neighbouring pixels provides higher compression ratio and therefore lower memory bandwidth. Behavioral analysis of the image using data analysis and visualization is performed to understand image pixels characteristics under various circumstances. These analysis were performed by obtaining different orientations of the input image. Mathematical models were developed and interesting results were obtained to exploit further the image correlation within neighbouring pixels.

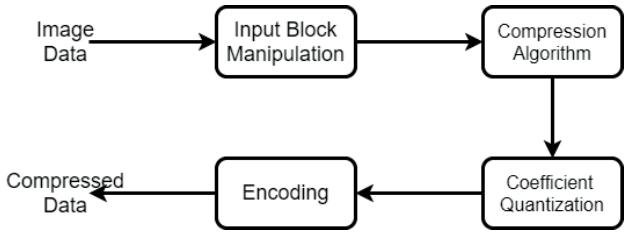

Figure 3.1: Efficient computation image compression technique

Even though a huge amount of data information in an image corresponds to higher quality pictures, it has been seen that there is a distinction between having data information and having knowledge. Although the huge amount of image data is processed, in general there are some image information which could be discarded to attain satisfactory results. This discard helps remove few redundant data and thereby decrease the hardware computation and memory consumption. Figure 3.1 shows the idea of achieving an efficient hardware image compression technique achieved with data manipulation and redundant removal of image information. The block '*Input Block Manipulation*' defines the handling of image data before the sending it to the compression stage to increase the compression efficiency.

3.1 Image Characteristic

This section is to discuss the types of image information which could be used to exploit image compression. These image characteristics are primarily obtained from the statistical models derived

using image data analysis. To explore the segmentation of the image blocks, behavioural-model of different block sizes were tested under a collection of image samples. Thus, reinforcing the usability of the proposed technique and the image response on different scenarios of input image block sizes.

3.1.1 Orientation

It has been observed in the data analysis that when an image is folded, oriented and projected in a specific format, the efficiency of the compression in terms of spatial redundancy and essential pixel information increases dramatically. The oriented pixel block is projected in several ways and the optimum compression is obtained based on the image sub-block performance.

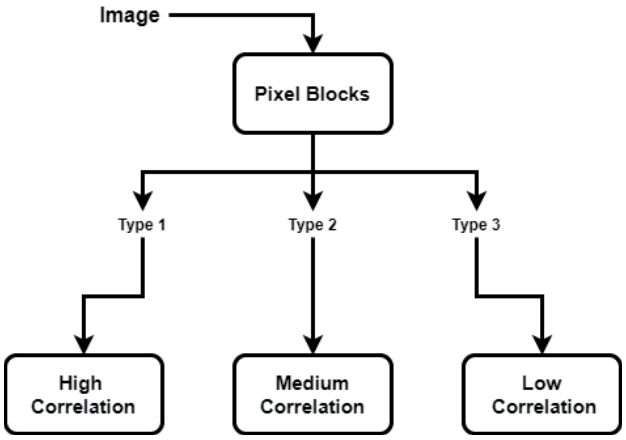

Figure 3.2: Partitioning image pixels blocks as per correlation with neighbouring pixels

These image sub-blocks are classified into three different categories. The first type corresponds to information most essential for the reconstruction of the image. They also provide a high correlation with their neighbouring pixels and hence exhibit a high compression ratio of around 2.5 and above on normal DCT compression. The second category is although necessary for the reconstruction of the image but doesn't contribute much to a higher coherency rate and therefore has a lower compression ratio of around 1.2 to 2.5 on normal DCT compression. Lastly, the third category provides the least compression rate and image information are mostly redundant to provide satisfactory image reconstruction. Figure 3.2 shows the high-level overview of the classification of the image as per input blocks.

3.1.2 Types of Compression

Although the division of an image block into sub-blocks was obtained, these sub-blocks are needed to be particularly handled to achieve optimised image compression. The methods experimented to

process all three types of image sub-blocks are discussed as follows.

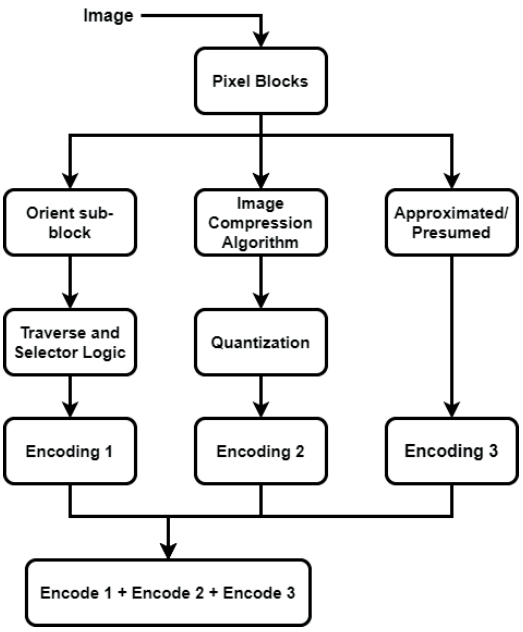

Figure 3.3: Operations on different sub-blocks

Type-1

Type-1 compression is mostly processed on the part where the highest correlation among neighbouring pixels is found. In general, it provides the best results where an average compression rate of around 2.5 and above is achieved, but the compression can be used on any part of the image block. The image sub-block used for Type-1 compression is operated with the number of encoders tracking different traversing logic. The highest compressed output data among the encoders is selected. The final data of the sub-block is a combination of encoded compressed data and a flag depicting the traversing technique used for the particular sub-block. Figure 3.3 shows the block diagram of Type-1 using *Traverse and Selector Logic* and later.

Type-2

It is seen that a part inside the image pixel block could also be processed directly with any DCT image compression technique. It must be noticed that the image sub-block size is smaller than the original image block which reduces the computation and the hardware resources required. This type of compression is named Type-2 compression. Figure 3.3 shows the implementation of Type-2 sub-block where image compression algorithm is applied and then encoded after quantization. In

general, these image sub-blocks are image parts where an average compression rate of a normal compression ratio ranges from 1.2 to 2.5.

Type-3

This type of compression is processed on part of the image sub-block with the highest dissimilarities among neighbouring pixels and does not provide more than a 1.2 compression ratio on a normal DCT compression. Here the pixel block discards most of the data information and reduces the necessary intensive computation. Approximate computing is applied here to introduce similarity among neighbouring pixels and achieve a high compression ratio with minimum hardware computation and resources. Another way to handle such data is by using static approximation which requires zero computation and is processed with pre-fixed values during decoding. Figure 3.3 shows the implementation of Type-3 compression where approximation or presumption of data is applied and encoded in the next stage.

3.2 Lossless Compression

The fundamental approach to achieve lossless compression is to process the entire stimuli of the image pixel block with a Type-1 compression. In other words, the pixel block is operated with several encoders on respective traversing logic and then compared with each compressed output to find the optimum result. The output is a combination of a flag and the compressed data. The flag depicts the traversing logic taken. Further memory layout optimisation is performed to achieve a higher compression rate. During the decode phase, the flag data is fetched at first to determine the traversal logic of the compressed data. The decoded data is placed as per the traversal logic to regain the lossless image.

3.3 Lossy Compression

The proposed techniques on lossy compression with acceptable image quality are discussed here.

3.3.1 Hardware efficient DCT

This technique uses Yang-DCT model [26] as a Type-2 DCT Compression. The architecture operates on 30 multipliers and 106 adders to compute an *8x8* input block size. Figure 3.4 shows the block diagram representation of the architecture. In the Figure, the image block is sub-divided into three categories, namely *ENCODERS* (Type-1), *DCT* (Type-2) and *Static Approximation* (Type-3). The compressed data is obtained from both types of encoders as shown in the figure. The static approximation data is applied here with pre-fixed values and are approximated in the decoder stage.

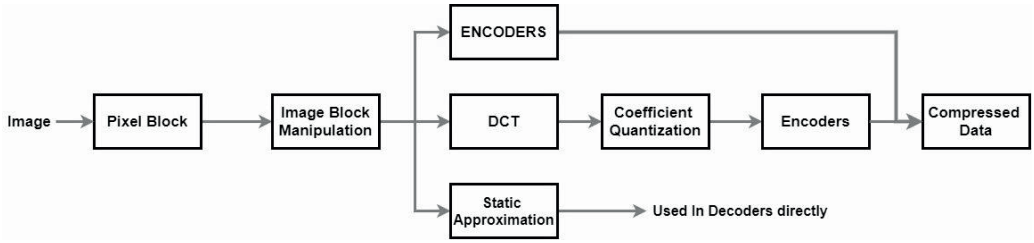

Figure 3.4: Lossy image compression using reduced hardware for DCT computation

3.3.2 Quantized Encoding Compression

The algorithm uses Type-1 and Type-3 compression. The sub-block comprising of the lowest compression ratio is statically approximated (Type-3 compression) with pre-fixed values. The rest of the pixel blocks are Type-1 compressed.

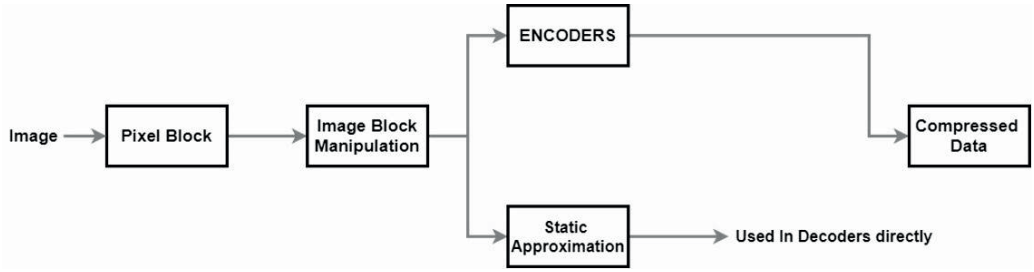

Figure 3.5: Lossy image compression using reduced hardware for encoding computation

Figure 3.5 shows the block level diagram of Static Approximate Compression. The Pixel Block is divided into two parts and handled separately with *'ENCODERS'* and *'Static Approximation'*.

3.3.3 Approximate Computing Compression

The algorithm is similar to the previous compression technique with the Type-3 compression implemented under an approximate computing technique. Approximate computing is used to achieve higher accuracy by introducing similarity among neighbouring pixels and thereby creating higher correlation. This increases the compression ratio by a large margin. The approach uses minimal hardware resources to compress the sub-block. The rest part of the image block is compressed with Type-1 compression.

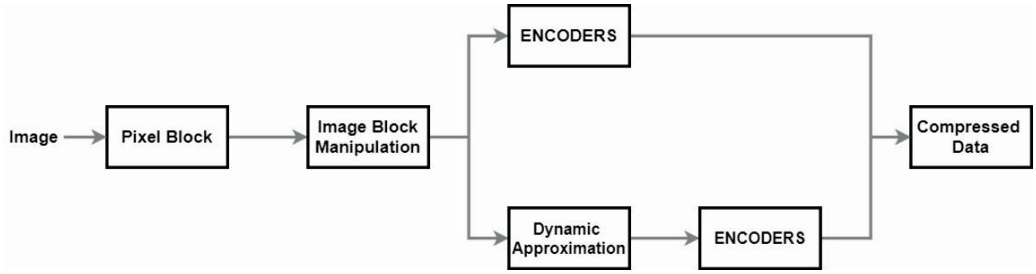

Figure 3.6: Encoder compression technique with approximate computing

Figure 3.6 shows the block diagram of Approximate Computing Compression. Pixel Block is split into two parts, where one part is sent to the ENCODERS (Type-1) while the other part is approximately computed and then encoded to increase the compression.

Figure 3.7: Decoder compression technique with approximate computing

Figure 3.7 shows the decoding logic. Compressed data are split and fetched to two different decoders. The compressed data from the Type-3 sub-block are decoded and approximated back to get a similar image replica of the sub-block. Both sub-blocks are merged to obtain back the image block.

Chapter 4
Results

The results of the proposed models are discussed in this chapter. Many different aspects such as image quality metrics, memory consumption, hardware requirement, compression ratio, etc are considered to draw the curve of comparison between each of the architectures.

Although the architectures were verified on different sets of dissimilar and similar images during the case study, the result analysis are based on common set of images which are thoroughly referred in Digital Image Processing research papers. A set of 15 different test-images are used as an input to the model. It has reassured an unbiased comparison from the previous image compression research papers.

4.1 Block Size Experiment

In this experiment, the *Hardware Efficient DCT* compression model is tested with different sizes of pixel blocks. It is primarily performed to find the most efficient image block size for the compression model. Different weights of quantization tables were used during the experiment. The DCT input is a 2^n by 2^n array of integers, where n is an integer. Thus the input stimuli of the model are tested with pixel block sizes of $4x4$, $8x8$, $16x16$, etc,. The experiment is performed solely to understand the characteristics of the *Hardware Efficient DCT* model with respect to the compression ratio on different input block sizes. From Table 4.1, it is seen the best image information are stored using the image block size of $8x8$. Block size of $8x8$ is able to maintain higher compression ratio with satisfactory image quality in comparison to block size of $4x4$ and $16x16$. A detailed information of the comparison is performed(see Appendices **A.1, A.2 and A.3**).

Table 4.1: Average Block Size Image Characteristics

Block Size	PSNR	MSE	MS-SSIM	SSIM	Memory(bits)	Comp Ratio
4x4	36.4	13.8	0.993	0.944	123	1.04
8x8	34.5	23.3	0.994	0.9548	351	1.64
16x16	30.6	78.4	0.934	0.814	1659	1.28

4.2 Computation Logic Utilization

In this section, the *Hardware Efficient DCT* image compression model is compared with the Reference DCT model in terms of their computational hardware utilisation. *Yang DCT* [26] model is used as the Type 2 compression to develop the *Hardware Efficient DCT* model. The hardware resources required to computation element in terms of adders and multipliers are split into NAND and AND gates and calculated for both reference Yang DCT and *Hardware Efficient DCT* (integrated with *Yang DCT*). The numbers of gates required to compute the entire input block of size $8x8$ in 1 clock cycle are compared against each other (see Appendices **A.5 and A.4**).

The results are obtained for a wide range of fixed point twiddle factor precision (TFP). It is observed that for 4-fixed point TFP model, the images start to show visual artefacts. Thus the logic gate calculations are considered from 5 bits to maximum of 8 bits twiddle factor precision.

4.3 Regular DCT Image Compression

In order to understand the pros and cons of all the other proposed models, these are compared to a reference model. This section is to discuss the behaviour of a regular DCT image compression model, acting as a reference to the other models. The primary motive of any compression technique is to keep lower data bandwidth. Due to this DCT compressions are mostly constrained with a finite twiddle factor precision. Here the twiddle factor precision is varied from a range of 5 to 8. Each of the different twiddle factor precision models are analysed separately. The image characteristics are studied under different quantization tables. Quantization tables are varied to provide a wide ranges of compression rate.

A threshold value of 0.990 MS-SSIM is kept as a benchmark of satisfactory image quality. The SSIM threshold value is kept as 0.960. As per industry standard, PSNR threshold value is kept as 28. It has been observed from the experiments that image artefacts starts to occur under these threshold value. It must be noted that same benchmarks value for the image metrics are kept for all other proposed models. An average of the hardware configurations considering 5 twiddle factor precision with various quantization tables is shown in Table 4.2. Here all the image results are able to maintain the threshold value with a compression ratio of 2.23. The image characteristics of these configuration are discussed thoroughly(Appendices **A.6, A.7, A.8, A.9**).

Table 4.2: Regular DCT

Image	PSNR	MSE	MS-SSIM	SSIM	Memory(bits)	Comp Ratio
Set 1	29.8	72.5	0.993	0.960	230	2.23

4.4 Hardware Efficient DCT

The image quality behaviour of Hardware Efficient DCT model is discussed in this section. The quantization tables are varied to provide a wide ranges of compression rate. An average of all the configuration for 5 twiddle factor precision along with various quantization tables is shown in Table 4.3. It is seen that the model is able to maintain similar MS-SSIM characteristics with a compression ratio 1.8. PSNR of 33 is achieved which is well above the benchmark value. It is seen that there is a small decrease in compression ratio in comparison to Regular DCT Compression. The model outperformed PSNR image quality result and maintained similar MS-SSIM result as compared to the Regular DCT compression. Various other configuration with combination of different twiddle factor and quantization table are shown (see Appendices **A.10, A.11, A.12, A.13**).

Table 4.3: Hardware Efficient DCT

Image	PSNR	MSE	MS-SSIM	SSIM	Memory(bits)	Comp Ratio
Set 1	33	31.5	0.993	0.95	293	1.8

4.5 Lossless Compression

The image characteristics of Lossless Compression is discussed in this section. Two different image sets are used here to further understand the reliability of the model. '*Set 1*' comprises of the images used in '*Regular DCT Image Compression*' and '*Hardware Efficient DCT*' models. '*Set 2*' comprises of 30 high resolution images. It is seen that a lossless image output is obtained with an average compression rate of 1.33. The results are discussed and compared graphically in the next chapter.

Table 4.4: Lossless Encoding Compression

Images	PSNR	MSE	MS-SSIM	SSIM	Memory(bits)	Comp Ratio
Set 1	∞	0.00	1.0	1.0	389	1.32
Set 2	∞	0.00	1.0	1.0	382	1.34

4.6 Quantized Encoding Compression

This section is to discuss the image characteristics of statically approximated image compression technique. Based on the degree of compression, three different hardware configuration for the same model are proposed. Similar to the '*Lossless Compression* two different sets of images are used in this model. An average of all three models are shown in Table 4.5. It is seen that all of the image metrics are well above their respective threshold value and provide an outstanding image quality with a compression ratio of around 2. The three different compression levels of *Quantized Encoding*

Compression models are shown(see Appendices **A.15, A.16 and A.17**). The results are discussed and compared graphically in the next chapter.

Table 4.5: Average Quantized Encoding Compression

Images	PSNR	MSE	MS-SSIM	SSIM	Memory(bits)	Comp Ratio
Set 1	43.9	7	0.998	0.985	270	1.9
Set 2	43.3	6.9	0.996	0.983	257	2

4.7 Approximated Computing Compression

The manipulated image pixel information with low spatial redundancy is approximately computed. Simple hardware using lower numbers of adders were used to approximate and compress image data. Similar to Quantized Encoding Compression three different models based on the compression levels are developed. An average of all of these compression models are shown in Table 4.6. It is seen that Approximated Computing Compression model performs better in comparison to Regular DCT model. Both MS-SSIM and SSIM is well above the threshold value with compression ratio of around 2. It shows better PSNR performance in comparison to Quantized Encoding Compression. Detailed information of the image characteristics for these models are shown(Appendices **A.18, A.19 and A.20**). The results are discussed and compared graphically in the next chapter.

Table 4.6: Average Approximate Computing Compression

Images	PSNR	MSE	MS-SSIM	SSIM	Memory(bits)	Comp Ratio
Set 1	46.8	3	0.996	0.982	280	1.9
Set 2	46	3	0.996	0.982	267	1.98

4.8 Approximate Quantized Compression

Approximate Quantized Compression is a combination of both Quantized Encoding Compression and Approximate Computing Compression. Two different hardware configuration based on compression levels are discussed here. An average of all of these compression models are shown in Table 4.7. It is seen that the average compression ratio is higher with a little deterioration in image performance when compared to Approximated Computing Compression and Quantized Encoding Compression. Detailed information on the image characteristics are discussed(Appendices **A.21 and A.22**. The results are discussed and compared graphically in the next chapter.

Table 4.7: Average Approximate Quantized Compression

Images	PSNR	MSE	MS-SSIM	SSIM	Memory(bits)	Comp Ratio
Set 1	43	5.4	0.995	0.975	248	2.1
Set 2	42.4	5.5	0.994	0.974	234	2.2

4.9 Power Simulation and Area Synthesis

The power and area consumption of the synthesized hardware accelerator is calculated. The simulations are performed only for the hardware-accelerated Lossless Compression and Quantized Encoding Compression-2 on a clock frequency of 1000MHz. The simulated values of the Lossless Compression encoder and decoder with different power groups are shown in Table 4.8 and 4.10. The area and the power are measured in $micrometer^2(\mu m^2)$ and $nanoWatt(nW)$ respectively. The power and area distribution of Lossless Compression model are shown (Appendices **4.9 and 4.11**). It is seen that the sequential and the logical types consume a larger part of the total power. In the Table, the Internal Power is the Static Power consumption of the accelerator. The Leakage Power represents both dynamic and static IR drop.

Table 4.8: Power & Area consumption on Lossless Compression Encoder

Type	Instances	Area(μm^2)	Leakage Power(nW)	Internal Power(nW)
Sequential	6553	3.3 $\times 10^3$	2.2 $\times 10^5$	195.7 $\times 10^5$
Inverter	3152	0.2 $\times 10^3$	0.2 $\times 10^5$	2.6 $\times 10^5$
Buffer	355	0.3 $\times 10^2$	0.3 $\times 10^4$	0.9 $\times 10^5$
Clock Network	120	0.3 $\times 10^1$	0.2 $\times 10^4$	1.4 $\times 10^5$
Logic	22537	2.0 $\times 10^3$	1.0 $\times 10^5$	31.1 $\times 10^5$
Total	32717	5.530 $\times 10^3$	3.44 $\times 10^5$	231.7 $\times 10^5$

Table 4.9: Power & Area distribution on Lossless Compression Encoder

Type	Area(%)	Leakage Power(%)	Internal Power(%)
Sequential	59.2	64.4	84.5
Inverter	3.3	4.9	1.1
Buffer	0.5	0.8	0.4
Clock Network	0.5	0.5	0.6
Logic	36.5	29.3	13.4
Total	100.0	100.0	100.0

Table 4.10: Power & Area consumption for Lossless Compression Decoder

Type	Instances	Area(μm^2)	Leakage Power(nW)	Internal Power(nW)
Sequential	3216	1.4 $\times 10^3$	0.9 $\times 10^5$	79.7 $\times 10^5$
Inverter	2388	0.2 $\times 10^3$	0.3 $\times 10^5$	3.9 $\times 10^5$
Buffer	824	0.1 $\times 10^3$	0.1 $\times 10^5$	2.1 $\times 10^5$
Clock Network	21	0.4 $\times 10^1$	0.3 $\times 10^3$	0.3 $\times 10^5$
Logic	17500	1.8 $\times 10^3$	1.2 $\times 10^5$	31.4 $\times 10^5$
Total	23949	3.56 $\times 10^3$	2.44 $\times 10^5$	117.27 $\times 10^5$

Table 4.11: Power & Area distribution on Lossless Compression Decoder

Type	Area(%)	Leakage Power(%)	Internal Power(%)
Sequential	40.2	37.5	68.0
Inverter	6.4	11.3	3.3
Buffer	2.0	3.2	1.8
Clock Network	0.1	0.1	0.2
Logic	51.3	47.9	26.7
Total	100.0	100.0	100.0

The power and area consumption (see Appendices **A.23, A.25**) and the percentage distribution (Appendices **A.24 and A.26**) for the hardware accelerators of Quantized Encoding Compression-2 encoder and decoder are shown. The tables show similar nature to the Lossless Compression model with sequential and logic type consuming most part of the total power.

Chapter 5
Analysis

In this chapter, the design-space exploration of all the architectures is discussed and compared among each other. This provides a profound understanding of the architecture with regards to its usability. Characteristics such as memory consumption, compression ratios, image quality metrics, and effect on the weights of quantization tables have been thoroughly analysed here. Comparisons of different image metrics of all the proposed and reference architectures are studied as well.

5.1 Image Input Block Behaviour

This experiment is to find the most efficient block size to perform Hardware Efficient DCT. The DCT image compression behaviour is studied on various image input block sizes for the proposed Hardware Efficient DCT architecture. Three different block sizes are experimented with here a similar set of images. The results for 3 different block sizes namely, $4x4$, $8x8$, $16x16$ were obtained in the previous chapter. A graphical comparison is made based on their respective MS-SSIM metric to further analyse the model in terms of different compression ratios.

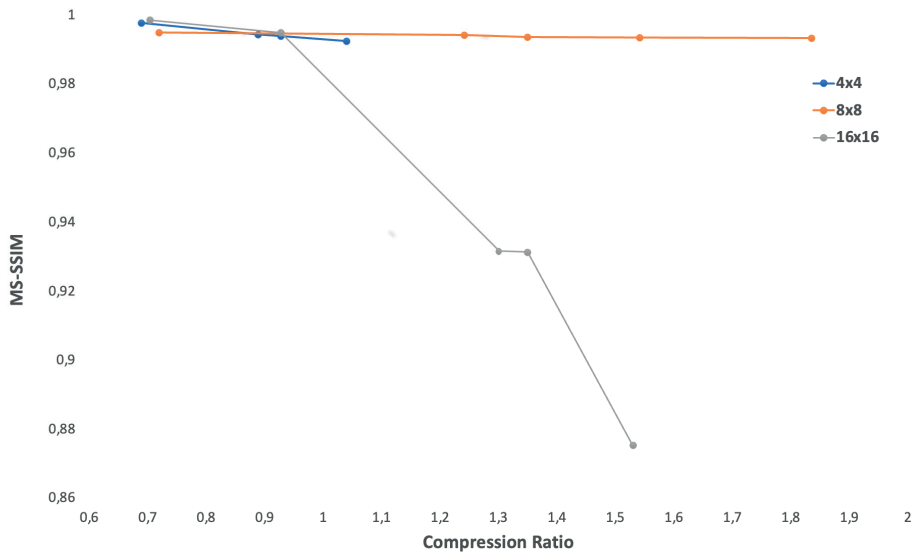

Figure 5.1: Hardware Efficient DCT performance on different image block sizes

Figure 5.1 shows that the best image quality results as per MS-SSIM versus compression ratio are achieved on $8x8$ block size. MS-SSIM values for both input block sizes of $4x4$ and $16x16$ start to decrease even before the compression ratio reaches 1. $8x8$ block size thus is considered during the further study.

5.2 Peak Signal to Noise Ratio Performance

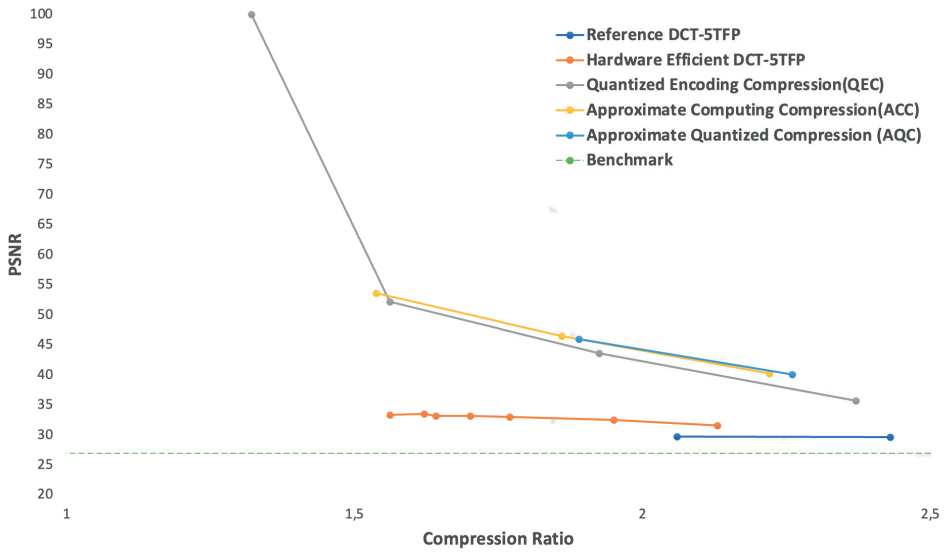

Figure 5.2: PSNR vs Compression Ratio

The state of the art models along with the Reference DCT model are analysed and compared in terms of their respective PSNR performance. Figure 5.2 shows that Approximate Quantized Compression (AQC), Approximate Computing Compression (ACC) and Quantized Encoding Compression (QEC) outperform Reference DCT and Hardware Efficient DCT models of 5 Twiddle Factor Precision(TFP). The Reference DCT maintains a PSNR of around 28 on the compression ratios ranging from 2 to 4.4. It is seen for images with a PSNR value below 27 mostly shows visual artefacts. A minimum PSNR value of 27 is kept as a benchmark to determine a satisfactory image output.

In applications with a compression rate of around 2.2 is required, Reference DCT of 5-TFP can be replaced with Hardware Efficient DCT of 5-TFP, as it provides higher PSNR image quality with a high compression ratio as compared to a Reference DCT. A lossless compression with a compression ratio of around 1.4 is seen as well. Here in Figure, the lossless compression is represented with a PSNR value of 100.

5.3 Structured Similarity Index Method Performance

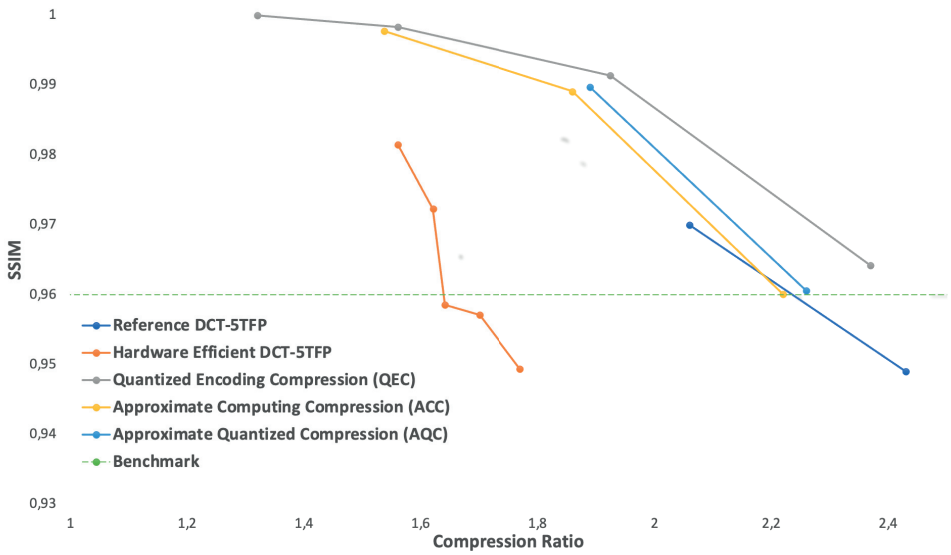

Figure 5.3: SSIM vs Compression Ratio

Similar to PSNR performance, all the discussed models in the previous chapter are analysed and compared here in terms of their respective Structured Similarity Index Method Performance (SSIM) performance. It is seen in Figure 5.3, Approximate Encoding Compression (AEC) provides the highest SSIM performance in comparison to all other models. A minimum SSIM value of 0.96 is kept to determine a satisfactory image output. Below this value, most of the images start to show visual artefacts. Approximate Quantized Compression (AQC) and Approximate Computing Compression (ACC) show higher image quality than Reference DCT with 5-TFP. Although Reference DCT provides similar MS-SSIM results as compared to Hardware Efficient DCT, in Hardware Efficient DCT of 5-TFP there is around 14 per cent reduction in compression ratio in comparison to Reference DCT of 5-TFP.

5.4 Multi Scale Structured Similarity Index Performance

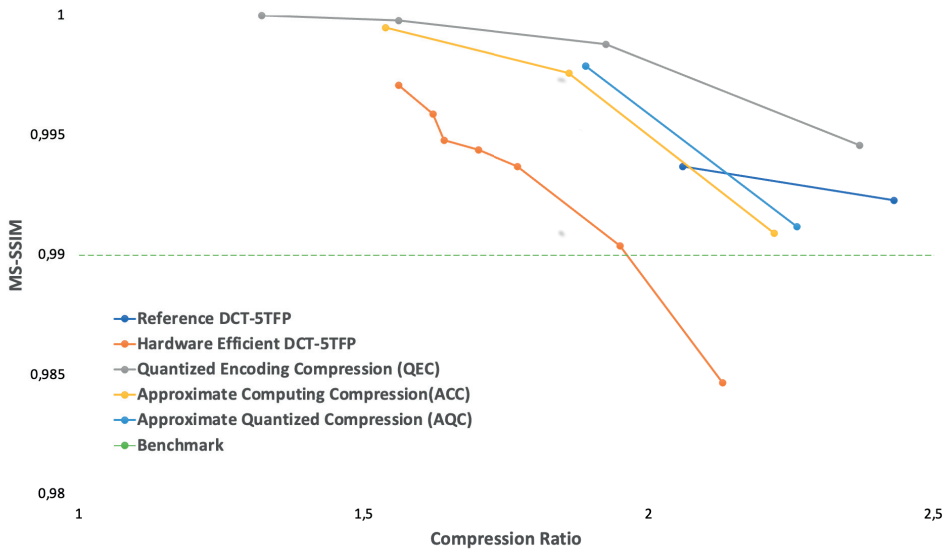

Figure 5.4: MS-SSIM vs Compression Ratio

In this section, all the discussed image compression models are analysed and compared in terms of their respective Multi-Scale Structured Similarity Index Performance (MS-SSIM) performance. In Figure 5.4, it is seen that Quantized Encoding Compression shows the best MS-SSIM performance among all other investigated models. A minimum MS-SSIM value of 0.99 is kept to determine a satisfactory image output. Below this most of the images start to show visual artefacts. Approximated Quantized Compression and Approximate Encoding Compression have better image quality in comparison to Reference DCT of 5-TFP with a compression rate of 2 and below. Similar to SSIM image performance, with comparable image quality, in Hardware Efficient DCT of 5-TFP there is around 14 per cent reduction in compression ratio in comparison to Reference DCT of 5-TFP.

5.5 Model Comparison

5.5.1 Metric Comparison

An overall comparison of all the implemented models is discussed in this section. The idea is to compare each architecture with an equal or similar compression ratio. For a better understanding of the models, the results of the model with a compression ratio close to 2 are considered here. Table 5.1 can be used as a reference for design space exploration and could be therefore used to prune out the undesired performance requirement.

Table 5.1: Overall Comparison of all Proposed and Reference Model

Model	CR	MSSSIM	SSIM	PSNR	MSE
Regular DCT - 5TFP	2.1	0.994	0.970	29.8	71.9
Regular DCT - 6TFP	1.9	0.992	0.946	29.6	73.2
Regular DCT - 7TFP	2.1	0.996	0.972	38.2	10.0
Regular DCT - 8TFP	2.0	0.996	0.964	37.6	11.7
Hardware Efficient DCT - 5TFP.	2.0	0.990	0.901	31.6	32.0
Hardware Efficient DCT - 6TFP.	1.8	0.992	0.944	32.8	34.0
Hardware Efficient DCT - 7TFP.	1.7	0.995	0.963	35.3	19.2
Hardware Efficient DCT - 8TFP.	1.6	0.997	0.982	34.9	20.8
Lossless	1.3	1	1	∞	0
Quantized Encoding Compression - 1	1.6	0.9998	0.998	52.2	0.5
Quantized Encoding Compression - 2	1.9	0.999	0.991	43.7	3.3
Quantized Encoding Compression - 3	2.4	0.995	0.946	35.7	17.5
Approximate Computing Compression - 1	1.5	0.9995	0.998	53.7	0.3
Approximate Computing Compression - 2	1.9	0.998	0.989	46.5	1.7
Approximate Computing Compression - 3	2.2	0.991	0.960	40.3	7.0
Approximate Quantized Compression - 1	2.3	0.991	0.961	8.7	8.7
Approximate Quantized Compression - 2	1.9	0.998	0.990	46.0	2.1

5.5.2 Area Comparison

Hardware area utilisation in $micrometer^2(\mu m^2)$ for the proposed implementation are discussed in Table 5.2.

Table 5.2: Total Area Requirement

Model	Total Area(μm^2)
Lossless Compression	13.4 $\times 10^3$
Quantized Encoding Compression - 1	12.8 $\times 10^3$
Quantized Encoding Compression - 2	12.2 $\times 10^3$
Quantized Encoding Compression - 3	11.6 $\times 10^3$
Approximate Computing Compression - 1	13.2 $\times 10^3$
Approximate Computing Compression - 2	12.6 $\times 10^3$
Approximate Quantized Compression - 1	13.1 $\times 10^3$
Approximate Quantized Compression - 2	13.4 $\times 10^3$

5.5.3 Hardware Reduction

To understand the area utilisation of the Hardware Efficient DCT model in comparison to the Reference DCT, the percentage reduction between the two models on similar Twiddle Factor Precision is shown in Table 5.3. The area utilisation is calculated based on the computational resources required by both the hardware accelerator in terms of NAND and AND logic gates.

Table 5.3: Total NAND-AND Gate Reduction

Twiddle Factor Precision	Total Reduction (%)
8 bits	38.6
7 bits	38.5
6 bits	38.4
5 bits	38.3

Chapter 6
Conclusions

In this master's thesis project, different attempts to abate the intensive hardware computation required on an image compression technique were discussed. State-of-the-art hardware efficient designs were developed to eradicate the expenses of heavy computations required to compress an image in a normal image compression technique. Image data analysis was performed to understand various image characteristics on all proposed and reference compression models. Verification and validation of image behaviours were carried out under different sets of images.

All state-of-the-art algorithms were verified and compared among each other in terms of their hardware computational logic, image quality, synthesized area. The relation between image quality to the memory bandwidth was graphically analysed to compare the benefits of each of the proposed models. All the discussions on different models were thoroughly studied and compared to provide the design solutions that could be used to best meet the desired design requirements.

6.1 Future Work

In the Hardware Efficient DCT accelerator, the weights of the quantization table were given randomly. Optimized quantization tables befitting the proposed and reference architecture were not focused. Quantization tables were rather developed to encourage similar behaviour in both cases. An optimized quantization table could thus be studied to further improve the compression rate.

Comparing the results of Approximate Quantized Compression and Quantized Encoding Compression models, it was seen that the dynamic approximation of Type 3 image data increased the image quality in terms of PSNR as compared to Quantized Encoding Compression although the quality in terms of MS-SSIM and SSIM image metrics deteriorates. Studies related to the behaviour of such image metrics could be performed to enhance the performance of image quality using approximate computing.

Appendix A

Data Tables

Table A.1: Image Characteristics for Input Block Size $4x4$

Quantization Table	$Q0_{4x4}$	$Q1_{4x4}$	$Q2_{4x4}$	$Q3_{4x4}$
PSNR	42.6	39.9	38.1	36.4
MS-SSIM	0.998	0.995	0.994	0.993
SSIM	0.984	0.964	0.958	0.944
Avg. Mem	185.8	144.6	138.1	123.2
MSE	4.5	8.4	10.1	13.8
Compression Ratio	0.69	0.89	0.93	1.04

Table A.2: Image Characteristics for Input Block Size $8x8$

Quantization Table	$Q0_{8x8}$	$Q1_{8x8}$	$Q2_{8x8}$	$Q3_{8x8}$	$Q3_{8x8}$
PSNR	35.9	34.5	34.5	34.6	34.5
MS-SSIM	0.995	0.994	0.994	0.994	0.994
SSIM	0.962	0.949	0.948	0.948	0.946
Avg. Mem	711.8	411.8	379.0	332.8	279.4
MSE	16.60	23.3	23.2	23.2	23.4
Compression Ratio	0.72	1.24	1.35	1.54	1.84

Table A.3: Image Characteristics for Input Block Size $16x16$

Quantization Table	$Q0_{16x16}$	$Q1_{16x16}$	$Q2_{16x16}$	$Q3_{16x16}$	$Q4_{16x16}$
PSNR	42.7	37.7	28.3	28.3	28.0
MS-SSIM	0.999	0.995	0.932	0.93	0.88
SSIM	0.9909	0.9503	0.7958	0.79	0.72
Avg. Mem	2912.1	2208.3	1574.9	1516.4	1335.3
MSE	3.498	9.96	95.84	96.1	111.6
Compression Ratio	0.703	0.927	1.3	1.35	1.53

Table A.4: Number of AND Gates for Computational Logic

TFP	Reference	Proposed
8 bits	18048	10364
7 bits	15792	9072
6 bits	13536	7776
5 bits	11280	6480

Table A.5: Number of NAND Gates for Computational Logic

TFP	Reference	Proposed
8 bits	212256	131040
7 bits	189936	117360
6 bits	167616	103680
5 bits	145296	90090

Table A.6: Regular DCT with 5-TFP

Quantization Table	Q_7	Q_8	Q_9	Q_{10}	Q_{11}	Q_{12}	Q_{13}
PSNR	29.8	29.7	29.6	29.6	29.6	29.5	29.4
MSE	71.9	72.9	73.5	73.7	73.7	74.9	75.2
MS-SSIM	0.9937	0.9923	0.9861	0.9859	0.9742	0.9594	0.9454
SSIM	0.970	0.949	0.914	0.912	0.8844	0.8386	0.8132
Memory(bits)	248	211	162	160.7	139	122.66	117.2
Comp Ratio	2.06	2.43	3.16	3.18	3.68	4.17	4.36

Table A.7: Regular DCT with 6-TFP

Quantization Table	Q_7	Q_8	Q_9	Q_{10}	Q_{11}	Q_{12}	Q_{13}
PSNR	29.8	29.8	29.7	29.6	29.6	29.6	29.5
MSE	71.45	72.3	73.1	73.2	73.0	73.7	73.8
MS-SSIM	0.9959	0.9951	0.992	0.9918	0.9873	0.9803	0.9741
SSIM	0.9876	0.975	0.949	0.945	0.933	0.898	0.884
Memory(bits)	427	367.5	280	275	238.4	192.5	180
Comp Ratio	1.2	1.393	1.83	1.88	2.147	2.66	2.84

Table A.8: Regular DCT with 7-TFP

Quantization Table	Q_7	Q_8	Q_9	Q_{10}	Q_{11}	Q_{12}	Q_{13}
PSNR	41.6	40.9	39.3	38.6	38.2	36.1	35.4
MSE	4.7	5.4	7.9	9.1	10.0	16.5	19.2
MS-SSIM	0.9994	0.999	0.9978	0.9975	0.9958	0.9922	0.989
SSIM	0.996	0.9917	0.9774	0.972	0.967	0.9413	0.932
Memory(bits)	424.86	374	292.378	285	249	199.37	187.3
Comp Ratio	1.20	1.37	1.75	1.80	2.05	2.568	2.73

Table A.9: Regular DCT with 8-TFP

Quantization Table	Q_7	Q_8	Q_9	Q_{10}	Q_{11}	Q_{12}	Q_{13}
PSNR	41.8	41.7	40.9	40.4	40.3	38.4	37.6
MSE	4.4	4.6	5.3	5.9	6.2	9.4	11.7
MS-SSIM	0.999	0.999	0.999	0.999	0.998	0.996	0.996
SSIM	0.998	0.997	0.991	0.988	0.987	0.970	0.964
Memory(bits)	483	450	376	366.7	332	267	251
Comp Ratio	1.06	1.137	1.36	1.40	1.54	1.92	2.04

Table A.10: Hardware Efficient DCT with 5-TFP

Quantization Table	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
PSNR	33.4	33.5	33.3	33.2	33.1	32.5	31.6
MSE	29.6	28.5	30.8	30.9	32.2	36.2	32.0
MS-SSIM	0.9971	0.9959	0.9948	0.9944	0.9937	0.9904	0.9847
SSIM	0.982	0.972	0.959	0.957	0.949	0.928	0.901
Memory(bits)	328	317	313.136	302.7	288.71	263	240.5
Comp Ratio	1.56	1.62	1.64	1.70	1.77	1.95	2.13

Table A.11: Hardware Efficient DCT with 6-TFP

Quantization Table	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
PSNR	33.4	33.5	33.4	33.4	33.3	33.1	32.8
MSE	29.6	29.2	29.8	29.8	30.1	31.6	34.0
MS-SSIM	0.9966	0.9962	0.9955	0.9955	0.995	0.9938	0.9917
SSIM	0.979	0.977	0.970	0.970	0.966	0.956	0.944
Memory(bits)	328.3	327.3	326.4	322.9	314.6	292	277
Comp Ratio	1.56	1.56	1.57	1.59	1.63	1.75	1.85

Table A.12: Hardware Efficient DCT with 7-TFP

Quantization Table	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
PSNR	33.7	34.1	34.7	34.8	34.9	35.2	35.3
MSE	27.859	25.41	22.03	21.47	20.9	19.74	19.21
MSSSIM	0.9968	0.9961	0.9959	0.9958	0.9958	0.9956	0.9953
SSIM	0.979	0.972	0.966	0.965	0.964	0.963	0.963
Memory(bits)	328.4	328.2	328.2	327.9	327.0	318.9	307
Comp Ratio	1.56	1.56	1.56	1.56	1.56	1.61	1.68

Table A.13: Hardware Efficient DCT with 8-TFP

Quantization Table	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
PSNR	33.4	33.6	34.0	34.1	34.2	34.6	34.
MSE	29.48	28.45	25.548	25.438	24.7	22.333	20.80
MS-SSIM	0.997	0.9967	0.9961	0.9961	0.996	0.9957	0.9955
SSIM	0.9815	0.979	0.9717	0.9715	0.9697	0.9658	0.9643
Memory(bits)	328.39	328.389	328.382	328.33	328.24	327.6	325.34
Comp Ratio	1.56	1.56	1.56	1.56	1.56	1.56	1.56

Table A.14: Lossless Encoding Compression

Images	PSNR	MSE	MS-SSIM	SSIM	Memory	Comp Ratio
Set 1	∞	0.00	1.0	1.0	389	1.32
Set 2	∞	0.00	1.0	1.0	382	1.34

Table A.15: Quantized Encoding Compression - 1

Images	PSNR	MSE	MS-SSIM	SSIM	Memory	Comp Ratio
Set 1	52.199	0.489	0.9998	0.9983	328	1.56
Set 2	51.27	0.493	0.9996	0.9975	319.67	1.60

Table A.16: Quantized Encoding Compression - 2

Images	PSNR	MSE	MS-SSIM	SSIM	Memory	Comp Ratio
Set 1	43.66	3.27	0.9988	0.9914	266	1.925
Set 2	42.87	3.41	0.9981	0.9888	257	1.99

Table A.17: Quantized Encoding Compression - 3

Images	PSNR	MSE	MS-SSIM	SSIM	Memory	Comp Ratio
Set 1	35.7	17.49	0.9946	0.9642	216	2.37
Set 2	35.86	17.03	0.9914	0.9614	196	2.61

Table A.18: Approximate Computing Compression - 1

Images	PSNR	MSE	MS-SSIM	SSIM	Memory	Comp Ratio
Set 1	53.66	0.344	0.9995	0.9977	333	1.537
Set 2	52.7	0.35	0.9994	0.9976	325	1.58

Table A.19: Approximate Computing Compression - 2

Images	PSNR	MSE	MS-SSIM	SSIM	Memory	Comp Ratio
Set 1	46.47	1.71	0.9976	0.9891	276	1.86
Set 2	45.67	1.75	0.9973	0.9885	267	1.92

Table A.20: Approximate Computing Compression - 3

Images	PSNR	MSE	MS-SSIM	SSIM	Memory	Comp Ratio
Set 1	40.3	7.047	0.9909	0.9601	231	2.22
Set 2	39.822	6.94	0.9903	0.9600	211	2.43

Table A.21: Approximate Quantized Compression - 1

Images	PSNR	MSE	MS-SSIM	SSIM	Memory	Comp Ratio
Set 1	40.11	8.72	0.9912	0.9606	226	2.26
Set 2	39.57	8.83	0.9904	0.9598	206	2.48

Table A.22: Approximate Quantized Compression - 2

Images	PSNR	MSE	MS-SSIM	SSIM	Memory	Comp Ratio
Set 1	46.01	2.08	0.9979	0.9897	271	1.89
Set 2	45.21	2.13	0.9976	0.9884	262	1.95

Table A.23: Power & Area consumption on Approximate Computing Compression - 2 Encoder

Type	Instances	Area(μm^2)	Leakage Power(nW)	Internal Power(nW)
Sequential	5803	2.93 $\times 10^3$	1.99 $\times 10^5$	174.08 $\times 10^5$
Inverter	2831	0.17 $\times 10^3$	0.16 $\times 10^5$	2.42 $\times 10^5$
Buffer	236	0.02 $\times 10^3$	0.02 $\times 10^5$	0.75 $\times 10^5$
Clock Network	96	0.02 $\times 10^2$	0.02 $\times 10^5$	1.15 $\times 10^5$
Logic	19854	1.83 $\times 10^3$	0.93 $\times 10^5$	28.87 $\times 10^5$
Total	28820	4.97 $\times 10^3$	3.12 $\times 10^5$	207.27 $\times 10^5$

Table A.24: Power & Area distribution on Approximate Computing Compression - 2 Encoder

Type	Area(%)	Leakage Power(%)	Internal Power(%)
Sequential	58.9	64.0	84.0
Inverter	3.4	5.1	1.2
Buffer	0.4	0.7	0.4
Clock Network	0.4	0.5	0.6
Logic	36.9	29.8	13.9
Total	100.0	100.0	100.0

Table A.25: Power & Area consumption on Approximate Computing Compression - 2 Decoder

Type	Instances	Area(μm^2)	Leakage Power(nW)	Internal Power(nW)
Sequential	3218	1.44 $\times 10^3$	0.93 $\times 10^5$	83.07 $\times 10^5$
Inverter	2809	0.25 $\times 10^3$	0.30 $\times 10^5$	4.33 $\times 10^5$
Buffer	772	0.06 $\times 10^3$	0.07 $\times 10^5$	1.93 $\times 10^5$
Clock Network	22	0.05 $\times 10^2$	0.03 $\times 10^4$	0.27 $\times 10^5$
Logic	17059	1.80 $\times 10^3$	1.15 $\times 10^5$	31.83 $\times 10^5$
Total	23880	3.56 $\times 10^3$	2.45 $\times 10^5$	121.42 $\times 10^5$

Table A.26: Power & Area distribution on Approximate Computing Compression - 2 Decoder

Type	Area(%)	Leakage Power(%)	Internal Power(%)
Sequential	40.6	37.8	68.4
Inverter	7.1	12.3	3.6
Buffer	1.8	2.9	1.6
Clock Network	0.1	0.1	0.2
Logic	50.4	46.8	26.2
Total	100.0	100.0	100.0

References

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," *8th IEEE European Test Symposium (ETS), Avignon, 2013, pp. 1-6. doi: 10.1109/ETS.2013.6569370.*, 2013.

[2] T. M. Q. Xu and N. S. Kim, "Approximate computing: A survey," *IEEE Design & Test, vol. 33, no. 1, pp. 8-22, Feb. 2016. doi: 10.1109/MDAT.2015.2505723*, 2016.

[3] S. P. P. A. R. Vaibhav Gupta, Debabrata Mohapatra and K. Roy, "Impact: imprecise adders for low-power approximate computing," *In Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design (ISLPED '11). IEEE Press, Piscataway, NJ, USA, 409-414.*, 2011.

[4] K. R. N. Ahmed, T. Natarjan, "Discrete cosine transform," *EEE T Comput 23(2), 90–93*, 1974.

[5] H. J, "Nvidia's ceo declares moore's law dead," 2017.

[6] K. R. S. T. C. Vinay K Chippa, Debabrata Mohapatra and A. Raghunathan., *Scalable effort hardware design on Very Large Scale Integration (VLSI) Systems.* IEEE, 2014-2016.

[7] S. E. A. N. A. S. R. S. D. Modha, R. Ananthanarayanan, "Cognitive computing. commun. acm 54(8), 62–71 (2011)," 2011.

[8] K. R. A. R. V. Chippa, S. Chakradhar, "Analysis and characterization of inherent application resilience for approximate computing," 2013.

[9] S. B. S. Hua, G. Qu, "An energy reduction technique for multimedia application with tolerance to deadline misses," 2003.

[10] S. B. S. Hua, G. Qu, "Probabilistic design of multimedia embedded systems," 2007.

[11] M. Levoy, *Volume rendering by adaptive refinement.* The Visual Computer, 1990.

[12] "An introduction to image compression, pp1-29,"

[13] M. Yang and N. Bourbakis, *An Overview of Lossless Digital.* IEEE Transactions on Image Processing, 2005.

[14] H. H. et al., *Using code perforation to improve performance, reduce energy consumption, and respond to failures.* MIT-CSAIL-TR2009-042, 2009.

[15] C. Sun and E.-H. Yang, *An efficient DCT-based image compression system based on Laplacian transparent composite model*. IEEE Transactions on Image Processing, 2015.

[16] A. Alfalou and P. K. Meher, *Optimized architecture using a novel subexpression elimination on Loeffler algorithm for DCT-based image compression*. IEEE Transactions on Image Processing, 2012.

[17] P. Kumbhare and U. M. Gokhale, *Design and implementation of 2D-DCT by using Arai algorithm for image compression*. Journal of The International Association of Advanced Technology and Science, 2015.

[18] Y. O. M. Jridi and A. Alfalou, *Low complexity DCT engine for image and video compression*. Real-Time Image and Video Processing, vol. 8656, pp. 1–9, 2013.

[19] R. M. M. Marimuthu and P. Swaminathan, *Sub-band based DCT for image compression*. Research Journal of Applied Sciences, Engineering and Technology, 2012.

[20] H.-C. H. Y.-S. S. L.-T. Ko, J.-E. Chen and T.-Y. Sung, *A unified algorithm for subband-based discrete cosine transform*. Research Journal of Applied Sciences, Engineering and Technology, 2010.

[21] Y.-S. S. T.-Y. Sung and H.-C. Hsin, *An efficient VLSI linear array for DCT/IDCT using subband decomposition algorithm*. Mathematical Problems in Engineering, 2010.

[22] M. Manoria and P. Dixit, *An efficient DCT compression technique using Strassen's matrix multiplication algorithm*. International Journal of Computer Applications, 2012.

[23] E. C. S. Khan and D. Menard, *High performance discrete cosine transform operator using multimedia oriented subword parallelism*. Advances in Computer Engineering, 2015.

[24] *Study of subjective and Objective Quality Assessment of Video*. IEEE Transactions on Image Processing, June 2010.

[25] R. L. Deshpande, R.G. and Sharma, *Video Quality Assessment through PSNR Estimation for Different Compression Standards*. Indonesian Journal of Electrical Engineering and Computer Science, 11,., 2018.

[26] S. E. Tsai and S. M. Yang, *A Fast DCT Algorithm for Watermarking in Digital Signal Processor*. Mathematical Problems in Engineering, 2017.

Printed by Tryckeriet i E-huset, Lund 2022

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2022-856
http://www.eit.lth.se