Development of Low Power Image Compression Techniques

Sunil Kumar Pattanaik

Department of Electronics & Communication Engineering
National Institute of Technology Rourkela

DEVELOPMENT OF LOW POWER IMAGE COMPRESSION TECHNIQUES

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology (Research)
in
Electronics & Communication Engineering

By

Sunil Kumar Pattanaik
Roll No: 60407004

Under the supervision of

Dr. Kamala Kanta Mahapatra
Professor

Department of Electronics & Communication Engineering
National Institute of Technology Rourkela

May, 2006

Department of Electronics & Communication Engineering
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
ORISSA, INDIA – 769 008

CERTIFICATE

This is to certify that the thesis titled "***Development of Low Power Image Compression techniques***", submitted to the National Institute of Technology, Rourkela by **Mr. Sunil Kumar Pattanaik**, Roll No. **60407004** for the award of the degree of Master of Technology (Research) in Electronics & Communication Engineering, is a bona fide record of research work carried out by him under my supervision and guidance.

The candidate has fulfilled all the prescribed requirements.

The thesis, which is based on candidate's own work, has not been submitted elsewhere for a degree/diploma.

In my opinion, the thesis is of standard required for the award of a Master of Technology (Research) degree in Electronics & Communication Engineering.

To the best of my knowledge, he bears a good moral character and decent behavior.

Prof. K. K. Mahapatra
Department of Electronics & Communication Engineering
National Institute of Technology
Rourkela-769 008 (INDIA)
Email: kkm@nitrkl.ac.in

Dedicated

to

my Grand Parents

BIO-DATA OF THE CANDIDATE

Name of the Candidate	:	Sunil Kumar Pattanaik
Father's Name	:	Purna Chandra Pattanaik
Permanent Address	:	Gajapati Marga At/po: Hinjilicut Dist: Ganjam Orissa 761102
Email ID	:	pattanaiksunil@yahoo.co.in

ACADEMIC QUALIFICATION

- Continuing **M. Tech. (Research)** in Electronics & Communication Engineering, National Institute of Technology Rourkela, Orissa (INDIA).
- **M.Sc.** in Electronics, *Berhampur University*, Orissa (INDIA).
- **P.G.D.C.A,** *Pondicherry University,* (INDIA)
- **B. Sc.** Science College Hinjilicut, *Berhampur University*, Orissa (INDIA).

PUBLICATION

- Published/Accepted 05 papers in National and International Conferences;
- Communicated 02 papers to International Journals.

ACKNOWLEDGEMENT

I would like to take this opportunity to extend my deepest gratitude to my teacher and supervisor, Prof. K. K. Mahapatra, for his continuous encouragement and active guidance. I am indebted to him for the valuable time he has spared for me during this work. He is always there to meet and talk about my ideas, to proofread and mark up my research papers and chapters, and to ask me good questions to help me think through my problems.

I am very much thankful to Prof. G. Panda, for his continuous encouragement. Also, I am indebted to him who provided me all official and laboratory facilities.

I am grateful to Dr. S. Meher and Prof. T.K. Dan for his valuable suggestions and comments during this research period.

My sincere thanks go to Prof. G.S. Rath, Dr. S.K. Patra, Prof. S. Pramanik, Dr. B. Majhi, and Prof. B.D. Sahoo whose valuable suggestions helped me a lot in completing this thesis.

I would like to thank Dr. Debi Prasad Das, Scientist, CEERI, Pilani for his kind support during the on period of my thesis work. In addition, let me thank all my friends Saroj, Pankaj, Rana babu, Sushant, Manas, Vamsi, Alekhika, Babita Madam, Sabuj, Debi, Nilamani sir, Jitendra sir and Peter for their great support and encouragement during the research period. Also, I am thankful to all the non-teaching staffs of ECE Department for their kind cooperation.

During the course of this work, I am supported by a project VLSI-SMDP sponsored by DIT, Govt. of India. I am really thankful to them.

Last but not the least, I take this opportunity to express my regards and obligation to my parents and family members for educating me in all expects. I can never forget for their unconditional support and encouragement to pursue my interests.

Sunil Kumar Pattanaik

LIST OF ABBREVIATIONS

AC	Arithmetic Compression
ANN	Artificial Neural Network
CR	Compression Ratio
DCT	Discrete Cosine Transform
DHT	Discrete Hartley Transform
DPCM	differential pulse code modulation
DWT	Discrete Wavelet Transform
EQ	Energy Quantisation
FPGA	Field Programmable Gate Array
JPEG	Join Picture Expert Group
MEQ	Modified Energy Quantisation
MLP	Multi-Layer Perceptron
MPEG	Moving Pictures Experts Group
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
RBEQ	Rule Based Energy Quantisation
RBF	Radial Basis Function
SNR	Signal to Noise Ratio
VLC	Variable Length Code
VLSI	Very Large Scale Integration
VQ	Vector Quantisation

LIST OF FIGURES

ABSTRACT

Digital camera is the main medium for digital photography. The basic operation performed by a simple digital camera is, to convert the light energy to electrical energy, then the energy is converted to digital format and a compression algorithm is used to reduce memory requirement for storing the image. This compression algorithm is frequently called for capturing and storing the images. This leads us to develop an efficient compression algorithm which will give the same result as that of the existing algorithms with low power consumption. As a result the new algorithm implemented camera can be used for capturing more images then the previous one.

1) Discrete Cosine Transform (DCT) based JPEG is an accepted standard for lossy compression of still image. Quantisation is mainly responsible for the amount loss in the image quality in the process of lossy compression. A new Energy Quantisation (EQ) method proposed for speeding up the coding and decoding procedure while preserving image quality. Some of the high frequency components of the transformed sub image are preserved in accordance to the quantisation value. There is no need of a dequantiser at the decoder side. This would enable reduction of hardware and would make the implementation much simpler. The proposed EQ method is modified and two new quatisation techniques Modified Energy Quantisation (MEQ) and Rule Based Energy Quantisation (RBEQ) proposed to further reduce the hardware requirement.

2) DCT and IDCT are used for the coding and decoding of the image. Calculations of both are independent to each other, so there is a need of two different hardware. Where as in case of DHT transform the forward and inverse transforms are same except a scale factor in the inverse transform. As a result the hardware requirement to compute both the forward and inverse DHT is approximately reduced by 50% as those of the DCT and IDCT.

3) All the Energy Quantisation techniques proposed are applied to further reduce the complexity of DHT based JPEG compression technique.

4) A new simple arithmetic compression technique is proposed for lossless compression. The computational complexity is very less compare to previous techniques.

5) All the image compression techniques proposed are synthesised for Virtex XCV1000 device for the testing and verification of low complexity and hence low power. DHT based JPEG with rule based energy quantisation is an ideal solution for lossy image compression technique as the hardware requirement is less and power consumption is very low compare to other techniques.

Contents

CHAPTER-7: Conclusion

Bibliography

Chapter

1

LOW POWER
Image Compression Techniques

Introduction

Data compression is the process of converting data files into smaller files for efficiency of storage and transmission. As one of the enabling technologies of the multimedia revolution, data compression is a key to rapid progress being made in information technology. The digital data have become an important source of information in the present world of communication systems. In this Internet age the power of the digital images to convey information as compared to the text data is obvious to all. This power of the images can be accessible and possible through the digital technology that enhances through its ability to process, transmit and reproduce with unparalleled faithfulness to the original images. It would not be practical to put images, audio, and video alone on websites without compression.

What is data compression? And why do we need it? Many people may have heard of JPEG (Joint Photographic Experts Group) and MPEG (Moving Pictures Experts Group), which are standards for representing images and video. Data compression algorithms are used in those standards to reduce the number of bits required to represent an image or a video sequence. Compression is the process of representing information in a compact form. Data compression treats information in digital form, which is, as binary numbers represented by bytes of data with very large data sets. Fox example, a single small 4" x 4" size color picture, scanned at 300 dots per inch (dpi) with 24 bits/pixel of true color, will produce a file containing more than 4 megabytes of data. At least three floppy disks are required to store such a picture. This picture requires more than one minute for transmission by a typical transmission line (64k bit/second ISDN). That is why large image files remain a major bottleneck in a distributed environment. Although increasing the bandwidth is a possible solution, the relatively high cost makes this less attractive. Therefore, compression is a necessary and essential method for creating image files with manageable and transmittable sizes.

Data and information are not the same thing rather the data are the means by which the information is conveyed. Thus while conveying the same information by different formats of the data, there is every possibility that some data are common between the ways of representing the information and can be removed as redundant. The basic motive behind the data compression is to search for such redundancy and then to remove them faithfully without compromising with the quality of the

reconstructed data. Mathematically the concept of redundancy may be expressed as follows:

If n_1 and n_2 denote the number of information–carrying units in two data sets which represents the same information, the relative data redundancy R_D of the first data set (the one characterized by n_1) can be defined as

$$R_D = 1 - \frac{1}{C_R} \tag{1.1}$$

where C_R, commonly called the compression ratio, is

$$C_R = \frac{n_1}{n_2} \tag{1.2}$$

For the case of $n_1 = n_2$, $C_R = 1$ and $R_D = 0$, indicating that the first representation of the information contains no redundant data. When $n_2 << n_1, C_R \to \infty$ and $R_D \to 1$, signifies that highly redundant data. When $n_2 >> n_1, C_R \to 0$ and $R_D \to -\infty$, indicates that the second data set contains much more data than the original representation. Generally a $C_R = 10$ (or 10:1) defines that the first data set has 10 information carrying units for every 1 unit in the second or compressed data set. Thus corresponding redundancy of 0.9 means 90 percent of the data in the first data set is redundant with respect to the second one.

In order to be useful, a compression algorithm has a corresponding decompression algorithm that reproduces the original file once the compressed file is given. There have been many types of compression algorithms developed. These algorithms fall into two broad types, lossless algorithms and lossy algorithms. A lossless algorithm reproduces the data exactly same as the original one. A lossy algorithm, as its name implies, loses some data. Data loss may be unacceptable in many applications. For example, text compression must be lossless because a very small difference can result in statements with totally different meanings. There are also many situations where loss may be either unnoticeable or acceptable. In image compression, for example, the exact reconstructed value of each sample of the image

is not necessary. Depending on the quality required of the reconstructed image, varying amounts of loss of information can be accepted.

1.1 Image Compression and Reconstruction

Three basic data redundancies can be categorized in the image compression standard.

1. Spatial redundancy due to the correlation between neighboring pixels.

2. Spectral redundancy due to correlation between the color components.

3. Psycho-visual redundancy due to properties of the human visual system.

The spatial and spectral redundancies are present because certain spatial and spectral patterns between the pixels and the color components are common to each other, whereas the psycho-visual redundancy originates from the fact that the human eye is insensitive to certain spatial frequencies. The principle of image compression algorithms are (i) reducing the redundancy in the image data and (or) (ii) producing a reconstructed image from the original image with the introduction of error that is insignificant to the intended applications. The aim here is to obtain an acceptable representation of digital image while preserving the essential information contained in that particular data set.

Figure 1.1: Image Compression System

The problem faced by image compression is very easy to define, as demonstrated in figure 1.1. First the original digital image is usually transformed into another domain, where it is highly de-correlated by using some transform. This de-correlation concentrates the important image information into a more compact form. The compressor then removes the redundancy in the transformed image and stores it into a compressed file or data stream. In the second stage, the quantisation block reduces the accuracy of the transformed output in accordance with some pre-

established fidelity criterion. Also this stage reduces the psycho-visual redundancy of the input image. Quantisation operation is a reversible process and thus may be omitted when there is a need of error free or lossless compression. In the final stage of the data compression model the symbol coder creates a fixed or variable-length code to represent the quantiser output and maps the output in accordance with the code. Generally a variable-length code is used to represent the mapped and quantised data set. It assigns the shortest code words to the most frequently occurring output values and thus reduces coding redundancy. The operation in fact is a reversible one.

The decompression reverses the compression process to produce the recovered image as shown in figure 1.2. The recovered image may have lost some information due to the compression, and may have an error or distortion compared to the original image.

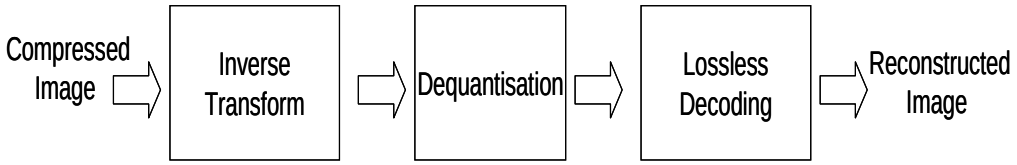


Figure 1.2: Image Decompression System

1.1.1 Transforms

The first stage in an image compressor is the transform. Transformation of the image data is required to convert the image into a domain where it is easier to compress. A transform operates on an image's pixel intensities and converts them into a set of transform coefficients. Natural images (which are the most common images to be compressed) have a lot of spatial correlation between pixel intensities, and these correlations can be exploited by the transform. This is achieved by mapping similar large scale changes in the data onto single transform coefficients. This type of mapping causes the transformed image to become highly decorrelated and standard compression techniques can then be used to further compress the transform coefficients.

The general form of a spatial intensity transform used on the image data is shown by equation (1.3):

$$c(k,l) = \sum_i \sum_j h(f(i,j)) \tag{1.3}$$

where $c(k,l)$ is the transform coefficient, $f(i,j)$ is the image pixel intensity array, $h(f(i,j))$ is the transform function. This equation shows that the transform coefficients are the sum of the effects of the transform on the pixel intensities, over the whole section of the image to be transformed.

The transform is rarely applied to the whole of the image. As the area of the image to which the transform has to be applied increases, the number of calculations also increases proportionally. This suggests that to keep the number of calculations small (and manageable), the area that the transform is applied to should be as small as possible. However, the decorrelation effects on the transform improve, when a larger area of the image is considered, and this in turn improves the compression performance.

In a real system a compromise is established between the compression and the speed of the transform. The effects of decorrelation are not linearly proportional to the area used so it is not possible to theoretically determine the best area to apply the transform to; it has to be done using practical results.

The image is broken into a sub-blocks and the transform is applied to each block separately. Each block then has a set of transform coefficients, which describe it. Although it has been stated that images are highly correlated, this is only true over local areas of the image. There may be little or no correlation between distant sections (100 pixels) of the image. Applying the transform to image blocks exploits the local similarity of the image without losing the benefits of decorrelation in the transform coefficients.

Transforming image blocks also introduces a blocking artifact effect, which can be a major problem. Since the coefficients that describe one block are not related to those describing the surrounding blocks, it is possible for discontinuities to occur along the block edge of compressed images. Blocking artifact is only visible at higher compression rates, in most systems, but can severely reduce the visual quality of a compressor, even if the rate distortion performance is still acceptable. The

blocks that the image is broken into do not have to be a fixed size or shape, but they are generally non-overlapping.

Transforms are generally formed in pairs so that the image can be reconstructed by applying the inverse transform. If the transforms are applied to an image without compression, then there are two possibilities, as shown in figure 1.3:

1. The transform is perfectly reconstructing and there is no error in the recovered image. This type of transform is often quite complex, since no data is lost and it can take a long time to compute.

2. The transform is "lossy" and information is lost on either the forward or inverse transform stage. These transforms can be useful, since they are very easy to calculate and often represent the image using a low number of transform coefficients. They produce a moderate compression without any further processing.

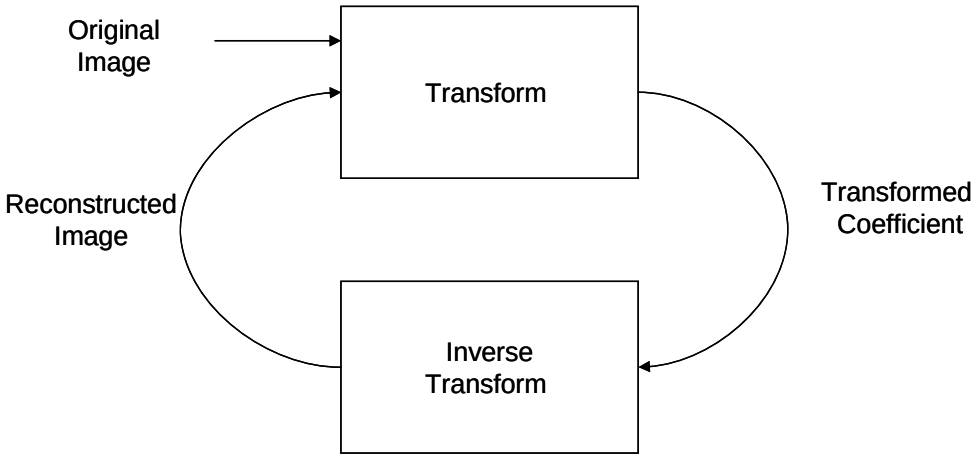


Figure 1.3: Block Diagram of Transform Loop

If the image is passed through the transform loop (figure 1.3) multiple times, then theoretically either lossless or lossy transforms should produce the same recovered error each time. However since lossy transforms often favour a variable block size method, it is not always the case that reapplying a codec to a recovered image (coded with the same codec) will produce the same result i.e. an image is coded many times it may never converge to a fixed result. This problem is cause for great concern in broadcast compression as images can be compressed many times after they are originally captured.

1.1.2 Quantisation

The transform stage spatially decorrelates the spatial property of the image, but does not always produce compression. Quantisation relevant to image coding is discussed in this section and simplified to allow a general rule for the quantisation of transform image coefficients to be developed. Quantisation is the main stage in case of lossy image compression, where most of the image compression is achieved. Before quantisation a transform coefficient may take an infinite range of values, limited only by the accuracy of the medium it is stored in. After quantisation the transform coefficient will be represented by a number of discrete values. This could be represented as:

$$c_q = q \circ c \tag{1.4}$$

where q is the quantisation function, c is the transform coefficients and $c_q = \{c_0, c_1, c_2, ...c_n\}$.

Linear quantisation is the most basic form of quantisation. The transform coefficients are divided by a quantisation step and the result is converted to an integer, by truncation of the decimal point equation (1.5).

$$c_q = Integer(\frac{c_i}{q_i}) \tag{1.5}$$

where q_i is the quantisation step, c_i is the transform coefficient, and c_q is the integer quantised coefficients.

The transform data is limited based on the 8 bit pixel intensities of standard images. This allows a quantisation step to be chosen, which limits the number of quantised states available, hence compressing the coefficient to a desired number of bits. However, it is not possible to control compression in this way, since a real system losslessly codes the quantised coefficients and this operation is not well defined.

The choice of q can vary since some transform coefficients are more important than others and, as a result, a quantisation table $(q(k,l))$ is usually formed, providing different q values for each transform coefficient. How to choose q is quite

difficult since there is no simple relationship between the compression and the value *q* takes.

1.1.3 Lossless Coding

Lossless coding aims at reducing the redundant data, by exploiting its statistics. Theoretically this coding method should compress a data source without introducing any new errors into the data. It is possible to reduce the information required to store data to a theoretical minimum, by exploiting the 'blind' statistics of the data, without considering the order in which it is received. This is usually achieved with a Huffman coding [1 - 2], but arithmetic coding [1 - 2] can also be useful.

In lossless coding it is useful to refer the inputs as data symbols to be compressed and the output from the lossless coder as compressed symbols. The data symbols are usually quantised transform coefficients in image compression, but they can be anything, provided the coder has knowledge of their statistics.

1.1.3.1 Entropy Coding

This is the most effective method of lossless coding and is nearly always present in image compression. Entropy is the average minimum number of bits that a data symbol stream can be compressed into, when each symbol is considered in isolation, based on its statistics. It can be calculated directly using equation (1.6), but it is sometimes not possible to reach the theoretical minimum entropy due to the implementation of the entropy coder.

$$Entropy = \sum PDD(x)\log_2(PDD(x)) \tag{1.6}$$

where *PDD(x)* is the probability density distribution of symbol x.

There are two different approaches to entropy coding, Huffman coding [1 - 2] and arithmetic coding [1 - 2]. Huffman compression is a more common and robust method, however cannot compress data to less than one bit/symbol. Arithmetic coding is less controllable and does not compress well at higher bits/symbol, however it can reduce its entropy below one bit/symbol. For this reason arithmetic

coders are not often used, unless the average entropy of the source is expected to drop below one bit / symbol.

1.1.3.2 Huffman Coding

Huffman compression [1 - 2] is designed to reduce the entropy of a data source close to the theoretical minimum entropy described in equation (1.6). The Huffman coder does this by representing common data symbols with short compressed symbols and rare data symbols with long compressed symbols. The average effect of this method is to reduce the redundancy of each compressed symbol to a minimum.

A Huffman coder determines the compressed symbols by forming a data tree from the original data symbols and their associated probabilities. The tree is formed by applying the following rules until it is complete.

1. Link the two unlinked data symbols with the lowest probability to form a new data symbol with probability equal to the sum of the previous two symbols.

2. Continue to link the active data symbols with the smallest probability until a complete data tree is formed.

Table 1.1 showing four symbols compressed using Hoffman Coding

Symbol	Probability	Binary Symbol	Binary Compressed Symbol
A	0.06	00	111
B	0.04	01	110
C	0.2	10	10
D	0.7	11	0

Consider the example shown in Table 1.1. There are four symbols, which can be described by the binary data symbols shown, and have an average bits/symbol coding rate of 2. If the four symbols are Huffman coded, then we obtain the compressed binary symbols shown in Table 1.1. The average bits/symbol for the compressed binary symbols is 1.3, which is a lossless compression of 1.5:1. The

example shown in Table 1.1 shows the major failing of the Huffman coder. The entropy of the source given is 1.0, but the Huffman coder is only able to reduce the bit/symbol coding rate to 1.3. If the entropy of the data source is often close to or below 1 bit/symbol, then the Huffman coder does not perform optimally, and needs to be improved.

1.1.3.3 Arithmetic Coding

Arithmetic coding [1-2] works by treating a stream of data symbols as a whole and does not replace individual data symbols with compressed versions. The coder is always implemented in binary and to avoid confusion it will be explained by reference to binary numbers.

An arithmetic coder takes an upper and lower limit, and defines a range between these upper and lower limits to be equivalent to a symbol with the probability of 1.0. Symbols are encoded by modifying the range of the arithmetic coder and sending symbols to reconstruct this range information at the decoder. The operation of an arithmetic coder can be demonstrated by using the data previously used in Table 1.1. The data can be represented as probabilities in the arithmetic coder as shown in figure 1.4.(b). It can be seen that the symbol probabilities stack to form a continuous range of probabilities between 0.0 and 1.0. This gives a range of probabilities that represent each symbol.

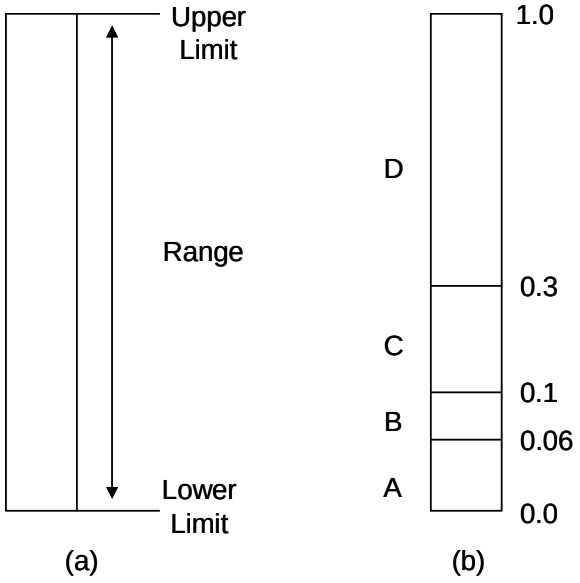


Figure 1.4: Limits of an Arithmetic Coder

The upper limit of the arithmetic coder is initially set to a value which corresponds to a probability of 1.0 but with infinite precision. This can be represented in binary as an infinite number of bits set to 1, since there is no fixed position for the decimal point. The lower limit is set to zero with infinite precision, again using an infinite number of binary bits set to 0. The only problem with this assumption is that computers work with fixed length numbers, commonly 4 to 8 bytes long, so it is impossible to represent the infinite precision. To over come this the limits are set as large as possible and their infinite length is simulated, by the coding algorithm.

The operation of encoding a symbol by the coder requires the range to be reduced in the following way:

$$Range = Upper\ Limit - Lower\ Limit$$

$$New\ Upper\ Limit = Lower\ Limit + Range \times P_{HIGH}(Symbol)$$

$$New\ Lower\ Limit = Lower\ Limit + Range \times P_{LOW(}(Symbol)$$

where $P_{HIGH}(Symbol)$ is the higher probability range of the symbol, and $P_{LOW}(Symbol)$ is the lower probability range of the symbol.

1.1.3.4 Run Length Coding

Run length coding is effective on data sources that have linear runs of the same symbol inside a data stream. The run length coder works by counting the number of occurrences of the same symbol and then forming a new symbol which describes the run length and the run symbol type. This is demonstrated in figure 1.5.

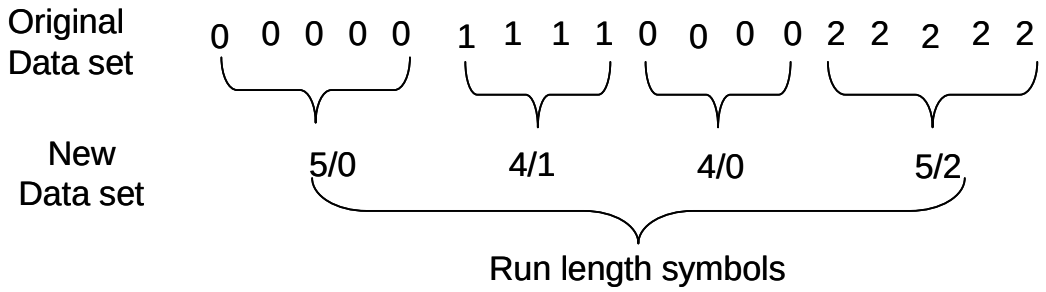


Figure 1.5: Diagram Illustrating the Run Length Coder

The entropy of the new symbol stream, which contains the run length codes is less than the original, providing there are sufficient runs to make the method viable. Run length coding has a similar effect to arithmetic coding without the complication of the arithmetic coder.

1.2 Performance Measures of Image Compression

Normally the performance of a data compression scheme can be measured in terms of three parameters. These are:

i. compression efficiency

ii. complexities

iii. Distortion measurement for lossy compression

Compression efficiency is measured through compression ratio (CR). The CR can be defined as the ratio of the data size (number of bits) of the original data to the size of the corresponding compressed data. The complexities of a digital data compression algorithms are measured by a number of data operations required performing both the encoding and decoding process. The data operations include additions, subtractions, and multiplication, divisions and shift operations. Generally there are two parameters: efficiency and complexities, which are mutually conflicting. One can not be benefited from both sides that are the efficiency of a compression algorithm varies proportionally with the complexity. More complex algorithm yields better compression performance with a greater data reduction and takes a longer time to execute. In the lossy compression algorithms, distortion measurement is used to measure the amount of information lost on the reconstruction of the original signal or image data that has been recovered from the compressed data through encoding and decoding operations. The mean square error (MSE) is one of the distortion measurements in the reconstructed data. The performance measurement parameter; signal to noise ratio (SNR) is also used to measure the performance of the lossy compression algorithms. These measurement criteria are often used for 1D data. For image data (2D) compression, the SNR is replaced by a parameter known as peak signal to noise ratio (PSNR). Another way of distortion measurement is the

percentage of energy retained in the compressed data. The above distortion measurements can be mathematically expressed as:

For One Dimensional data Mean Square Error calculated as

$$MSE = \frac{1}{N}\sum_{n=0}^{N-1}\left|x(n) - x'(n)\right|^2 \tag{1.7}$$

where N is the number of pixels in the image, $x(n)$ is the original data and $x'(n)$ is the compressed data.

PSNR(Peak Signal to Noise Ratio) for two dimensional image data is

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE1}\right) \tag{1.8}$$

Where

$$MSE1 = \frac{1}{MN}\sum_{m}^{M-1}\sum_{n=0}^{N-1}\left|x(m,n) - x'(m,n)\right|^2 \tag{1.9}$$

Percentage of energy retain is given by

$$\left|\frac{(Vector\ norm\ of\ \ data\ after\ compression)^2}{(Vector\ norm\ of\ \ data\ before\ compression)^2}\right| \times 100 \tag{1.10}$$

None of the main methods for measuring image distortion takes into account, how good the recovered image looks to the human visual system. This is an area called psycho-visual image analysis, and it is an area of research which has a large scope. Unfortunately little progress has been made into an automated method for calculating a psycho-visual distortion measure.

1.3 Digital Camera

A digital camera is a popular consumer electronic device that can capture images, or "take pictures," and store them in digital format. A digital still camera does not contain film, but rather then it contains one or more ICs, possessing processors and memories [7]. The key advantage of digital photography systems is their ability to provide noise-free storage, duplication, editing, and transmission of the

digital images. The increased processing, storage, and communications capabilities of desktop PCs, coupled with their decreasing cost, have sparked the growth of digital photography.

From a designer's point of view, a simple digital camera performs two key tasks. The first task is that of processing images and storing them in internal memory. The second task is that of uploading the images serially to an attached PC. The task of processing and storing images is initiated when the user presses the shutter button. At this point, the image is captured and converted to digital form by a charge-coupled device (CCD) or CMOS. Then, the image is processed and stored in internal memory.

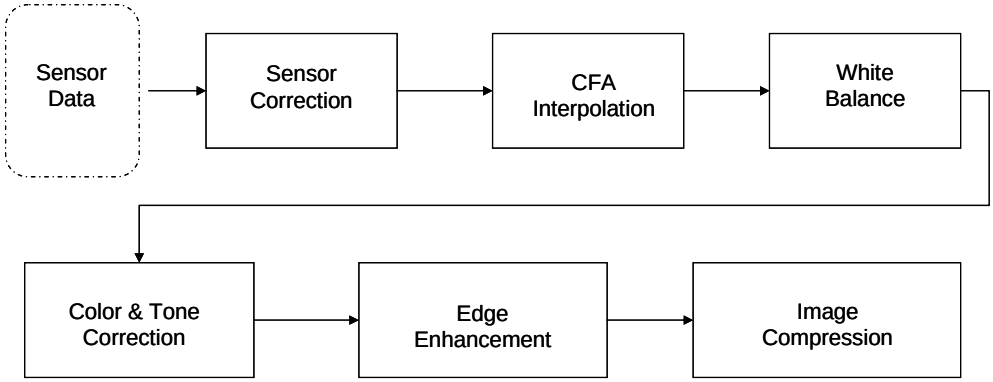


Figure 1.6: Digital Image Processing Steps in a Digital Camera

Figure 1.6, shows the key image processing steps implemented in digital cameras [4, 6, 7], or in host computers that process the raw image files provided by digital cameras. The sensor data may be processed to conceal defective or noisy pixels, and to correct for sensitivity variations caused by the lens or sensor.

CFA - Most digital cameras use this approach. A mosaic *color filter array* (CFA) is fabricated on top of the individual sensor photosites. Many different arrangements of colors are possible [4, 6], but each photosite is sensitive to only one spectral band. An optical anti-aliasing filter is normally used to reduce false color artifacts.

Color sequential - The color image is created with a monochrome sensor using three successive exposures taken with different optical red, green, and blue (RGB) filters, or a tunable LCD, in the optical path. These cameras are used for studio photography of inanimate scenes, but cannot be used for portraits, because subject motion causes colored edge artifacts. The CFA data from the sensor is interpolated to

reconstruct the "missing" color pixel values. Adaptive FIR filters are used for better results.

White balance corrects for the scene illuminant, since daylight has a greater proportion of blue energy than tungsten lamps. The R and B signals are multiplied by R and B gain values intended to provide equal RGB pixel values for neutral (e.g., white or gray) objects. Color correction may use a 3 x 3 matrix to correct the camera spectral sensitivities, and tone correction uses a set of lookup tables. Image sharpening, achieved by simple or adaptive spatial filters, compensates lens blur and provides a subjectively sharper image.

To store images on memory cards, standard JPEG compression [2, 3, 32] is typically used to reduce the file size to less than 2 bits per pixel, in order to increase the number of images that can be stored. Exif format JPEG-compressed files produced by digital cameras, and by film scanners to picture CD discs, include metadata that provides information about the image capture device and the picture taking conditions. This metadata can be used to simplify image retrieval and provide higher quality prints from the digital files. The metadata often includes a "thumbnail" size image at the beginning of the PEG file, to allow groups of images to be rapidly viewed so that appropriate images can be quickly selected for viewing, copying, or printing.

Some of the new compression standard features are:

- Improved compression efficiency
- Multiple resolution representation
- Embedded bit stream (progressive decoding from lossless to lossy)
- Region-of-interest coding

Many of these features are invaluable for digital cameras. For example, improved compression efficiency allows for either more images to be stored on the same memory card or for the same number of images to be stored with a higher image quality. The multi-resolution feature allows the various segments of the bit stream to be decoded to provide multiple resolutions of the image as needed. For example, the

low-resolution image can be used as a thumbnail display on the camera LCD, while the medium resolution image is shared through e-mail, and the higher resolution images are used to create high-quality prints.

1.3.1 Requirements Specification

Specification describes what a particular 'system should do, namely the system's requirements [6]. Specifications include both functional and nonfunctional requirements. Functional requirements describe the system's behavior, meaning the system's outputs as a function of inputs (e.g., "output X should equal input y times 2"). Nonfunctional requirements describe constraints on design metrics (e.g., "the system should use 0.001 watt or less"), battery life should be as long as possible. The initial functional specification of a system may be very general and may come from the company's marketing department.

1.3.2 Nonfunctional Requirements

Given our initial requirements specification, we might want to pay attention to several design metrics in particular: performance, size, power, and energy. Performance is the time required to process an image [6]. Size is the number of elementary logic gates (such as a two input NAND gate) in our IC. Power is a measure of the average electrical energy consumed by the IC while processing an image. Energy is power times time, which directly relates to battery lifetime. Some of these metrics will be constrained metrics-those metrics must have values below (or in some cases above) a certain threshold. Some metrics may be optimization metrics - those metrics should be improved as much as possible, since this optimization improves the product. A metric can be both a constrained and optimization metric.

Regarding size, our design must use an IC that fits in a reasonably sized camera. Suppose that, based on current technology, we determine that our IC has a size constraint of 200,000 gates. In addition to being a constrained metric, size is also an optimization metric, since smaller ICs are generally cheaper. They are cheaper because we can either get higher yield from a current technology or use an older and hence cheaper technology.

1.3.3 Functional Requirements

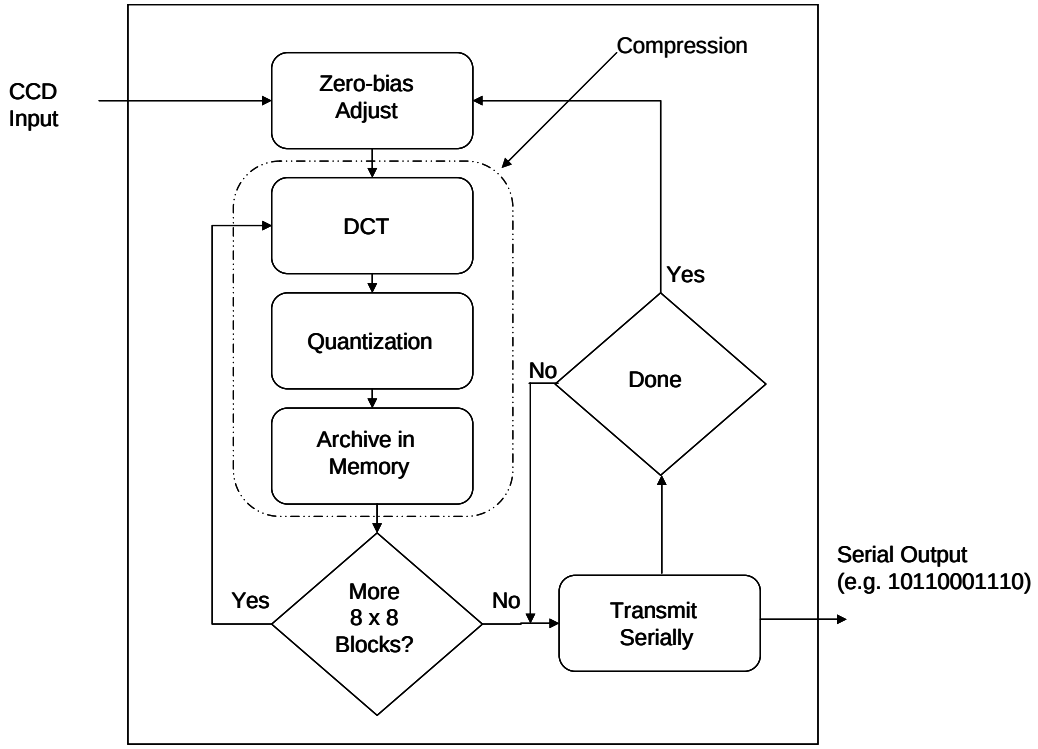


Figure 1.7: Functional block-diagram specification of a digital camera

The high-level functionality of the digital camera can be described by using the flowchart in figure 1.7 [6]. The major functions involved in image capture, namely zero-bias adjust, Image compression (DCT, quantisation and archive in memory). The compressed image data transmitted serially. Note that figure 1.4 does not dictate that each of the blocks be mapped onto a distinct processor. Instead, the description only aids in capturing the functionality of the digital camera by breaking that functionality down into simpler functions. The functions could be implemented on any combination of single-purpose and general-purpose processors.

Finally, power is a constrained metric because the IC must operate below a certain temperature. Note that the digital camera cannot use a fan to cool the IC, so low power operation is crucial. Let's assume we determine the power constraint to be 200 mill watt. Energy will be an optimization metric because we want the battery to last as long as possible. Notice that reducing power or time each reduces energy.

1.4 Motivation

Image compression is an important issue in digital image processing and finds extensive applications in many fields. This is the basic operation performed frequently by any digital photography technique to capture an image. For longer use of the portable photography device it should consume less power so that battery life will be more. To improve the Conventional techniques of image compressions using the DCT have already been reported and sufficient literatures are available on this. The JPEG is a lossy compression scheme, which employs the DCT as a tool and used mainly in digital cameras for compression of images. In the recent past the demand for low power image compression is growing. As a result various research workers are actively engaged to evolve efficient methods of image compression using latest digital signal processing techniques. The objective is to achieve a reasonable compression ratio as well as better quality of reproduction of image with a low power consumption. Keeping these objectives in mind the research work in the present thesis has been undertaken. In sequel the following problems have been investigated.

For solving for low power image compression we have modified the existing JPEG architecture which is the basic technique used for image compression and proposed some new low complexity method of compression technique. The performances of the proposed compression techniques have been evaluated and have been compared with that of the standard technique. Finally the proposed algorithms implemented in hardware and the power consumption is estimated. It is in general, observed that the proposed techniques are efficient than the conventional one. The results of the investigation have been outlined and discussed in subsequent Chapters.

1.5 Literature Survey

Interest in portable devices [6] has enhanced the requirement of developing low-power signal processors and algorithms, as well as the development of low-power general purpose processors. Designers have been able to reduce the energy requirements of particular functions, such as video compression, by several orders of magnitude [15, 63]. Low power techniques [11 - 23] can be discussed at various levels of abstractions: system level, algorithm and architecture level, logic level,

circuit level, and technology level [12 - 14]. One important technique for low power is at algorithmic level by using algorithmic transformations [12 - 14]. This technique exploits the complexity, concurrency, regularity, and locality of an algorithm. Reducing the complexity of an algorithm reduces the number of operations and hence the power consumption. This technique is used for reduction of power consumption by reducing the complexity.

The JPEG [24, 32, 41, 50] is one of the standard digital image compression for multilevel still images including both grayscale and colour images. Among the different models the JPEG baseline is the most widely used. It is based on the Transform coding using the DCT. Due to the lossy transformation, the JPEG output at high CR (at 15-20:1) becomes effected by the blocking artifacts and ridges separated in the image that are found to merge during compression. Normally the DCT has been used in the JPEG compression scheme. However, in the DSP literature many other efficient transforms such as Discrete Hartley Transform (DHT) [56 – 60] can be used in place of DCT, which may enhances the performance of the JPEG.

Vector quantisation, (VQ) mainly used for lossy image compression to reduce the image data [42]. Many VQ techniques applied for image compression has been reported [41 - 45]. The main problem with existing VQs is preparation of the code book [44, 52]. In VQ techniques creating the code book is a complicated process which has to be designed mainly to preserve the low frequency coefficients, and most of the high frequency coefficients are discarded.

Many lossless image compression techniques [66 -73] are exists. The main aim behind these compression techniques to get good quality image reconstruction and low compression. These techniques are mainly used for those images like medical images [66, 68, 69], satellite images etc.

1.6 Chapter-wise Organization of the thesis:

The thesis is organised as follows:

(1) **Chapter-1** outlines the basic principle of image compression and deals with the performance measurement of digital image compression. It gives a brief idea of digital camera architecture. It also reviews the relevant

literature on digital image compression and formulates the problems to be investigated in sequel. The motivation behind choosing various problems is also outlined.

(2) **Chapter-2** presents an overall idea about low power VLSI design, such as what are the sources of power loss in a circuit and how to control these losses. Describes all the low power techniques such as system level, algorithm level, architecture level, logic level and circuit level. Lastly it describes the basic steps to be followed for low power design.

(3) **Chapter-3** introduces DCT and its application in JPEG image compression technique briefly. The work reported in the literature on JPEG based compression is clearly reviewed. Three new quantisation techniques are proposed to reduce the computational complexity. The proposed techniques are compared with the standard JPEG technique through computer simulation on standard images.

(4) In **Chapter-4** the standard DCT is replaced by DHT to reduce the computational complexity, ringing effect and blocking effect. The three quantisation techniques proposed in chapter-1 is used with DHT to further reduce the complexity of compression technique. The performance of these new techniques is compared with the standard conventional JPEG.

(5) In **Chapter-5** a new method of lossless image compression technique is proposed. The computational complexity of this technique is very less compared to other image compression technique. The performance of these new techniques is compared with the standard conventional lossless JPEG compression technique.

(6) **Chapter-6** briefly introduces VLSI design and AccelChip synthesis process. Also lists the FPGA implementation results of all the proposed techniques and compared with the standard JPEG technique.

(7) In **Chapter-7,** conclusion has been made and some further research scopes are suggested.

Chapter

2

LOW POWER
Image Compression Techniques

Low Power VLSI Design Techniques

In the past few years there has been an explosive growth in the demand for portable computing and communication devices, from mobile telephones to sophisticated multimedia gadgets [11]. This interest in these devices has enhanced the requirement of developing low-power signal processors and algorithms, as well as the development of low-power general purpose processors. In the digital signal processing area, the results of this attention to power are quite remarkable. Designers have been able to reduce the energy requirements of particular functions, such as video compression, by several orders of magnitude [14]. This reduction has come as a result of focusing on the power dissipation at all levels of the design process, from algorithm design to the detailed implementation. In the general purpose processor area, however, there has been little work done to understand how to design energy efficient processors.

Performance of processors has been growing at an exponential rate, doubling every 18 to 24 months. However, at the same time the power dissipated by these processors has also been growing considerably. Although the rate of growth of power dissipation is perhaps not quite proportional to performance and size (the number of transistors) it still has led to processors which dissipated more than 50W. For such processors cooling becomes an absolute necessity and at high power dissipation level this is even difficult and expensive. If this trend continues processors will soon dissipate hundreds of watts, which would be unacceptable in most systems. Thus there is great interest in understanding how to continue increasing performance without increasing the power dissipation.

For portable applications the problem is even more severe since battery life depends on the power dissipation. Lithium-ion batteries have an energy density of approximately 100Wh/kg, the highest available today. To operate a 50 W processor for 4 hours requires a 2 kg battery; hence it can hardly be termed as a portable device. In order to compare processor designs that have different performance and power one needs a measure of "goodness". If two processors have the same performance or the same power, then it is trivial to choose which is better—users prefer higher performance for the same power level or the lower power one if they have the same performance. But processor designs rarely have the same performance. Designers have to determine whether to add a particular feature will make a processor more

desirable or not. However micro-architectural designing changes the amount of parallelism of the processor, affects the efficiency of the processor. Since both the performance and energy dissipation of modern processors depend heavily on the design of the memory hierarchy, one must look not only at the processor itself, but also have to look at the design of the memory. Since memories and clocking circuits are critical components of every digital system, much work already has been done to reduce the energy requirements. A different approach to reduce the energy dissipation of clocks and memories is to change the technology by scaling the supply voltage and the threshold voltage of transistors.

2.1 Power Dissipation Sources

In CMOS circuits, the main contributions to the power consumption are from short-circuiting current, leakage current, and switching currents [18], [21]. In the following subsections, we introduce them separately.

2.1.1 Short-Circuit Power

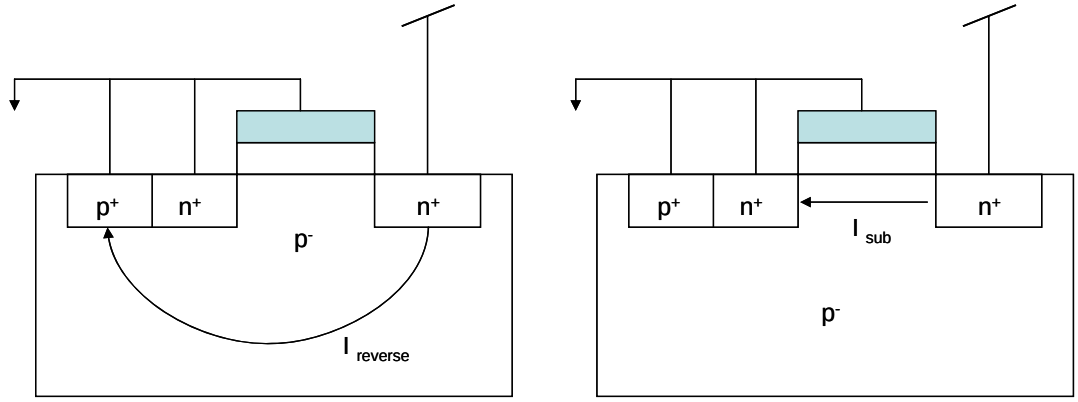
In a static CMOS circuit, there are two complementary networks: p-network (pull-up network) and n-network (pull-down network). The logic functions for the two networks are complementary to each other. Normally when the input and output state are stable, only one network is turned on and conducts the output either to power supply node or to ground node and the other network is turned off and blocks the current from flowing. Short-circuit current exists during the transitions as one network is turned on and the other network is still active. For example, the input signal to an inverter is switching from 0 to V_{dd} . During this transaction, there exists a short time interval where the input voltage is larger than V_{tn} but less than $V_{dd} - \left|V_{tp}\right|$. During this time interval, both PMOS-transistor (p-network) and NMOS-transistor (n-network) are turned on and the short-circuit current flows through both kinds of transistors from power supply line to the ground.

The exact analysis of the short-circuit current in a simple inverter [18] is complex, this is analyzed by SPICE simulation. It is observed that the short-circuit current is proportional to the slope of input signals, the output loads and the

transistor sizes [19]. The short-circuit current consumes typically less than 10% of the total power in a "well-designed" circuit [19].

2.1.2 Leakage Power

Leakage currents are due to two sources: one from the currents that flow through the reverse biased diodes (reverse biased PN-Junction current), the other from the currents that flow through transistors that are non-conducting (subthreshold channel conduction current).



(a) (b)

Figure 2.1: Leakage current types: (a) reverse biased diode current, (b) subthreshold leakage current.

The leakage currents are proportional to the leakage area and exponential of the threshold voltage. The leakage currents are due to manufacturing technology and cannot be modified by the designers except in some logic styles. Subthreshold leakage and reverse-biased junction leakage, both increases dramatically with temperature and are independent of the operating voltage for a given fabrication process.

The leakage current is in the order of pico-Ampere, but it increases as the threshold voltage is reduced. In some cases, like large RAMs, the leakage current is one of the main concerns. The leakage current is currently not a severe problem in most digital designs. However, the power consumed by leakage current can be as large as the power consumed by the switching current for $0.06\mu m$ technology. The usage of multiple threshold voltages can reduce the leakage current in deep-submicron technology. Leakage current is difficult to predict, measure or optimized.

Generally, leakage current serves no useful purposes, but some circuits do exploit it for intended operations, such as power-on reset signal generation. The leakage power problem mainly appears in very low frequency circuits or ones with "sleep modes" where dynamic activities are suppressed.

2.1.3 Switching Power

The switching currents are due to the charging and discharging of node capacitances. The node capacitances mainly include gate, overlapping, and interconnection capacitances. The power consumed by switching current [18] can be expressed as

$$P = \alpha C_L f V_{dd}^2 / 2 \tag{2.1}$$

where α a is the switching activity factor, C_L is the load capacitance, f is the clock frequency, and V_{dd} is the supply voltage.

The above equation (2.1) shows that the switching power depends on a few quantities that are readily observable and measurable in CMOS circuits. It is applicable to almost every digital circuit and hence provides guidelines for the low power design.

The power consumed by switching current is the dominant part of the power consumption. Reducing the switching current is the focus of most low power design techniques. For large capacitance circuits, reduction of the frequency is the best way to reduce the switching power. The use of different coding methods, number representation systems, continuing sequences and data representations can directly alter the switching frequency of the design, which alters the switching power. The best method of reducing switching frequency is to eliminate logic switching that is not necessary for computation.

2.2 Low Power Techniques

Low power techniques can be discussed at various levels of abstractions: system level, algorithm and architecture level, logic level, circuit level, and technology level [18], [21]. Figure 2.2 shows some examples of techniques at the different levels.

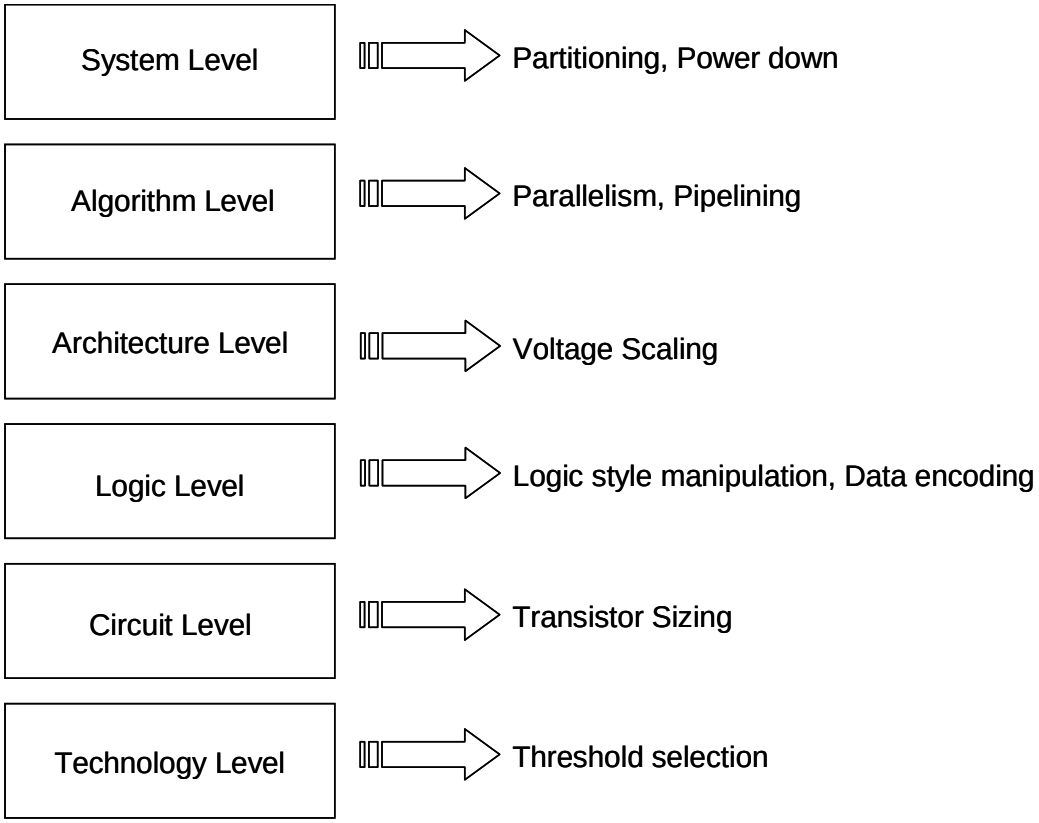


Figure 2.2: Low-power design methodology at different abstraction levels.

In the following sections, an overview for different low power techniques has been described in detail. This is organized on the basis of abstraction level.

2.2.1 System Level

A system typically consists of both hardware and software components, which affect the power consumption. The system design includes the hardware/software partitioning, hardware platform selection (application-specific or general-purpose processors), resource sharing (scheduling) strategy, etc. The system design usually has the largest impact on the power consumption and hence the low power techniques applied at this level have the most potential for power reduction.

At the system level, it is hard to find the best solution for low power in the large design space and there is a shortage of accurate power analysis tools at this level. However, if, for example, the instruction-level power models for a given processor are available, software power optimization can be performed [23]. It is observed that faster code and frequently usage of cache are most likely to reduce the power consumption.

The order of instructions also have an impact on the internal switching within processors and hence on the power consumption.

The power-down and clock gating are two of the most used low power techniques at system level. The non-active hardware units are shut down to save the power. The clock drivers, which often consume 30-40% of the total power consumption, can be gated to reduce switching activities as illustrated in figure 2.3.

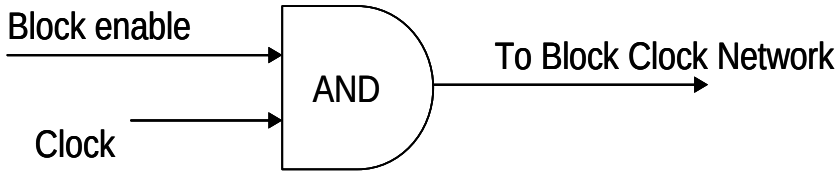


Figure 2.3: Clock gating.

The power-down can be extended to the whole system. This is called sleep mode and widely used in low power processors. The system is designed for the peak performance. However, the computation requirement is time varying. Adapting clocking frequency and/or dynamic voltage scaling to match the performance constraints is another low power technique. The lower requirement for performance at certain time interval can be used to reduce the power supply voltage. This requires either feedback mechanism (load monitoring and voltage control) or predetermined timing to activate the voltage down-scaling.

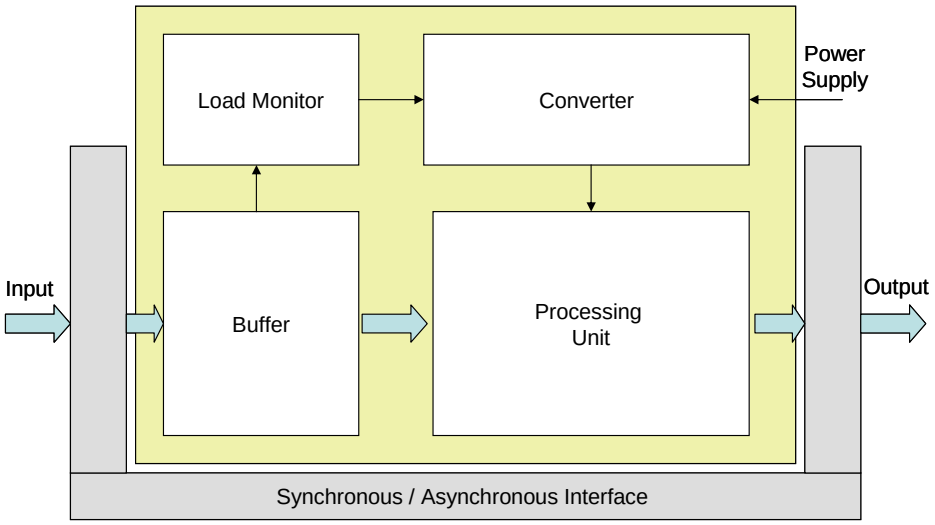


Figure 2.4: Asynchronous design with dynamic voltage scaling.

Asynchronous design of the circuit can also be used as another low power designing technique. The asynchronous designs have many attractive features, like non-global clocking, automatic power-down, no spurious transitions, and low peak

current, etc. It is easy to reduce the power consumption further by combining the asynchronous design technique with other low power techniques, for instance, dynamic voltage scaling technique [20], as shown in the following figure 2.4.

2.2.2 Algorithm Level

The algorithm selection has large impact on the power consumption. The task of algorithm design is to select the most energy-efficient algorithm that just satisfies the constraints. The cost of an algorithm includes the computation part and the communication/storage part. The complexity measurement for an algorithm includes the number of operations and the cost of communication and storage. Reduction of the number of operations, cost per operation, and long distance communications are key issues to algorithm selection.

One important technique for low power of the algorithmic level is algorithmic transformations [21]. This technique exploits the complexity, concurrency, regularity, and locality of an algorithm. Reducing the complexity of an algorithm reduces the number of operations and hence the power consumption. The possibility of increasing concurrency in an algorithm allows the use of other techniques, e.g., voltage scaling, to reduce the power consumption. The regularity and locality of an algorithm affects the controls and communications in the hardware.

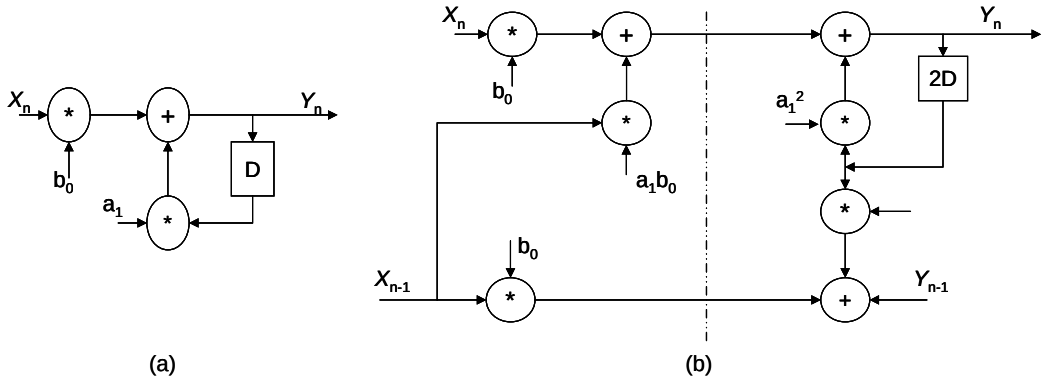


Figure 2.5: (a) Original signal flow graph. (b) Unrolled signal flow graph.

The loop unrolling technique [18], [12 - 13] is a transformation that aims to enhance the speed. This technique can be used for reducing the power consumption. With loop unrolling, the critical path can be reduced and hence voltage scaling can be applied to reduce the power consumption.

In figure 2.5, the unrolling reduces the critical path and gives a voltage reduction of 26% [18 – 23]. This reduces the power consumption with 20% even the capacitance load is increases with 50% [13]. Furthermore, this technique can be combined with other techniques at architectural level, for instance, pipeline and interleaving, to save more power. In some cases, like digital filters, the faster algorithms, combined with voltage-scaling, can be used for energy-efficient applications [18].

2.2.3 Architecture Level

According to the selection of the algorithm, the architecture can be determined for the given algorithm. From equation (2.1) we can say that, an efficient way to reduce the dynamic power consumption is the voltage scaling. When supply voltage is reduced, the power consumption is reduced. However, this increases the gate delay. To compensate the delay, low power techniques like parallelism and pipelining [14] architectures were used.

The use of two parallel datapath is equivalent to interleaving of two computational tasks. A datapath to determine the largest number of C and (A + B) is shown in figure 2.6. It requires an adder and a comparator. The original clock frequency is 40 MHz [14].

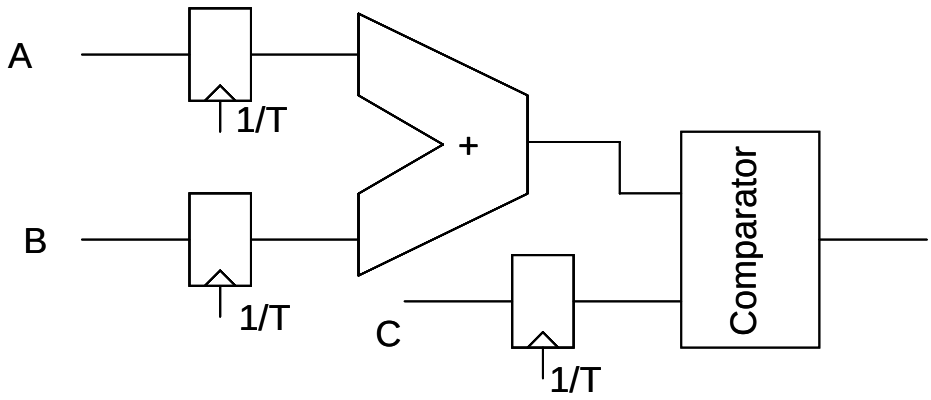


Figure 2.6: Original data path.

In order to maintain the throughput while reducing the power supply voltage, we use a parallel architecture. The parallel architecture with twice the amount of resources is shown in figure 2.7. The clock frequency can be reduced to half, from 40 MHz to 20 MHz since two tasks are executed concurrently. This allows the supply

voltage to be scaled down from 5 V to 2.9 V [14]. Since the extra routing is required to distribute computations to two parallel units, the capacitance load is increased by a factor of 2.15 [14]. The power is calculated by using equation (2.2).

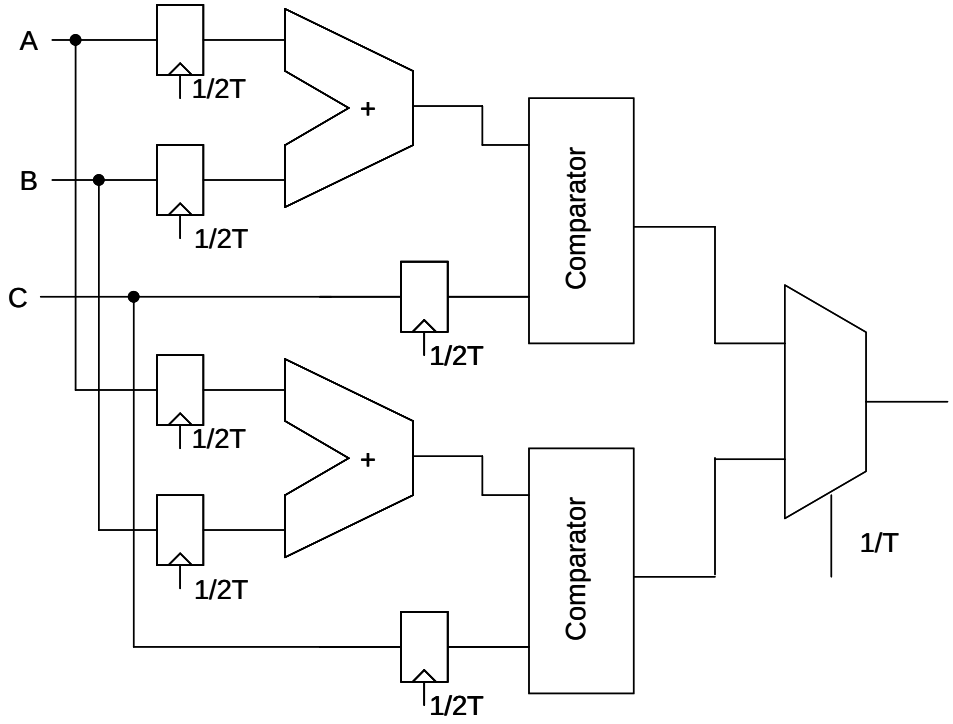


Figure 2.7: Parallel implementation.

$$P_{par} = C_{par} V_{par}^2 f_{par} = (2.15 C_{actual})(0.58 V_{actual})^2 (\frac{f_{actual}}{2}) = 0.36 P_{actual} \quad (2.2)$$

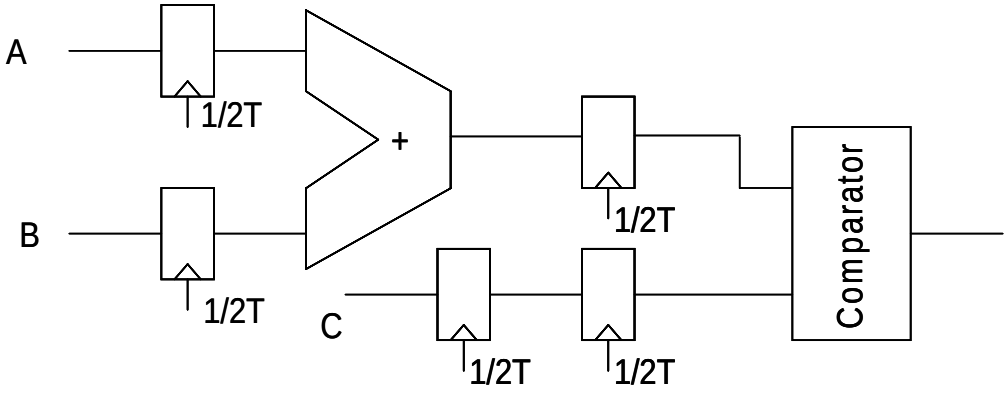


Figure 2.8: Pipelining implementation.

$$P_{pipe} = C_{pipe} V_{pipe}^2 f_{pipe} = (1.15 C_{actual})(0.58 V_{actual})^2 (f_{actual}) = 0.39 P_{actual} \quad (2.3)$$

Pipelining is another method for increasing the throughput. By adding a pipelining buffer / register after the adder in figure 2.8., the throughput can the

increased from $1/(T_{add} + T_{comp})$ to $1/\max(T_{add}, T_{comp})$. If T_{add} is equal T_{comp}, this increases the throughput by a factor of 2. As a result the supply voltage also scaled down to 2.9 V (the gate delay doubles) [14]. The effective capacitance increases to a factor of 1.15 because of the insertions of latches [14]. The power consumption for pipelining [14] is calculated using equation (2.3).

Main advantage of pipelining is the low area overhead in comparison with using parallel data paths. Another benefit is that the amount of glitches can be reduced. However, since the delay increases significantly as the voltage approaches the threshold voltage and the capacitance load for routing and/or pipeline registers increases, there exists an optimal power supply voltage. Reduction of supply voltage lower than the optimal voltage increases the power consumption.

2.2.4 Logic Level

The power consumption depends on the switching activity factor, which in turn depends on the statistical characteristics of data. The low power techniques at the logic level, however, focus mainly on the reduction of switching activity factor by using the signal correlation and the node capacitances. In case of the gated clocking, the clock input to non-active functional block does not change by gating, and, hence, reduces the switching of clock network.

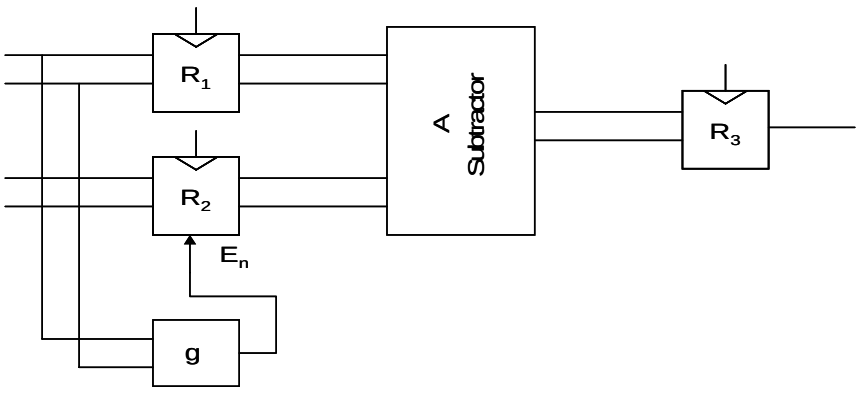


Figure 2.9: A precomputation structure for low power.

Precomputation [18] uses the same concept to reduce the switching activity factor: a selective precomputing of the output of a circuit is done before the outputs are required, and this reduces the switching activity by gating those inputs to the circuit. As shown in figure 2.9, the input data is partitioned into two parts, corresponding to

registers R_1 and R_2. One part, R_1, is computed in precomputation block g, one clock cycle before the computation of A. The result from g decides gating of R_2. The power can then be saved by reducing the switching activity factor in A.

Lets consider a comparator as an example of precomputation for low-power. The comparator takes the MSB of the two numbers to register R_1 and the others to R_2. The comparison of MSB is performed in g. If two MSBs are not equal, the output from g gated the remaining inputs. In this way, only a small portion of inputs to the comparator's main block A (subtracter) is changed. Therefore the switching activity is reduced.

Gate reorganization [18] is another technique used to restructure the circuit. This can be decomposition a complex gate to simple gates, or combines simple gates to a complex gate, duplication of a gate, deleting/addition of wires. The decomposition of a complex gate and duplication of a gate help to separate the critical and non-critical path. Which reduce the size of gates in the non-critical path, as a result reduces the power consumption. In some cases, the decomposition of a complex gate increases the circuit speed and gives more space for power supply voltage scaling. The composition of simple gates can reduce the power consumption. The complex gate can reduce the charge/discharge of high-frequently switching node. The deleting of wires reduces the circuit size as a result, reduces the load capacitance. The addition of wires helps to provide an additional interconnection for better results.

Logic encoding defines the way data bits are represented on the circuits. The encoding is usually optimized for reduction of delay or area. In low power design, the encoding is optimized for reduction of switching activities since various encoding schemes have different switching properties.

In a counter design, counters with binary and Gray code have the same functionality. For N-bit counter with binary code, a full counting cycle requires $2(2^n - 1)$ transitions [18] A full counting cycle for a Gray coded N-bit counter requires only 2^n transitions. For instance, the full counting cycle for a 2-bit binary coded counter is from 00, 01, 10, 11, and back to 00, which requires 6 transitions. The full counting cycle for 2-bit Gray coded counter is from 00, 01, 11, 10, and back to 00, which requires 4 transitions. The binary coded counter has twice transitions as the

Gray coded counter when the n is large. Using binary coded counter therefore requires more power consumption than using Gray coded counter under the same conditions.

Traditionally, the logic coding style is used for enhancement of speed performance. Careful choice of coding style is important to meet the speed requirement and minimize the power consumption. This can be applied to the finite state machine, where states can be coded with different schemes.

A bus is the main on-chip communication channel that has large capacitance. As the on-chip transfer rate, increases, the use of buses contributes with a significant portion of the total power. Bus encoding is a technique to exploit the property of transmitted signal to reduce the power consumption.

2.2.5 Circuit Level

At the circuit level, the power saving techniques is quite limited if compared with the other techniques at higher abstract levels. However, this cannot be ignored. The power savings can be significant as the basic cells are frequently used. A few percents improvement for D flip-flop can significantly reduce the power consumption in deep pipelined memory systems.

In CMOS circuits, the dynamic power consumption is caused by the transitions. Spurious transitions typically consume between 10% and 40% of the switching activity power in the typical combinational logic. In some cases, like array multipliers, the amount of spurious transitions is large. To reduce the spurious transitions, the delays of signals from registers that converge at a gate should be roughly equal. This can be done by insertions of buffers and device sizing [18]. The insertions of buffer increase the total load capacitance but can still reduce the spurious transitions. This technique is called path balancing.

Many logic gates have inputs that are logically equivalent, i.e., the swapping of inputs does not modify the logic function of the gate. Some examples of gates are NAND, NOR, XOR, etc. However, from the power consumption point of view, the order of inputs does effect the power consumption. Lets consider the figure 1.10, the A-input, which is near the output in a two-input NAND gate, consumes less power than the B-input closed to the ground with the same switching activity factor. Pin

ordering is to assign more frequently switching input pins near to the output node, which will consume less power. In this way, the power consumption will be reduced without cost. However, the statistics of switching activity factors for different pins must be known in advanced and this limits the use of pin ordering [18].

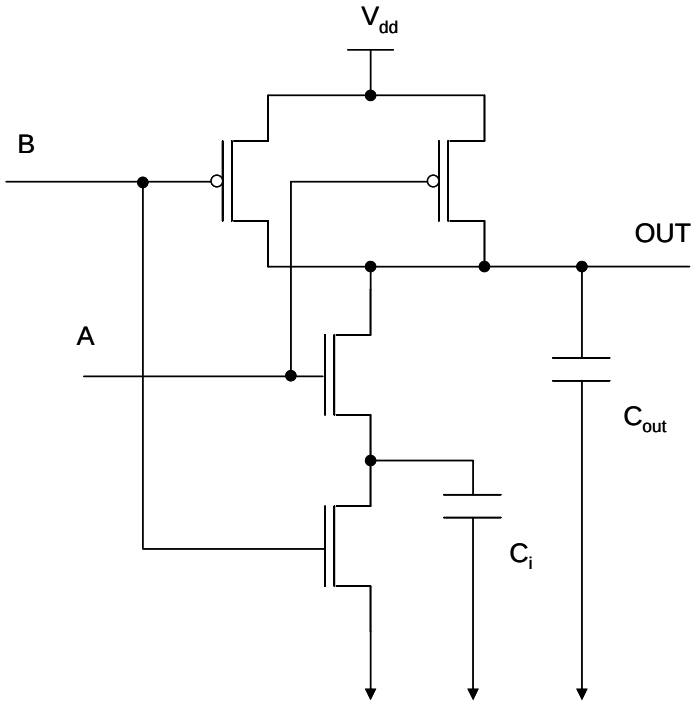


Figure 2.10: A two input NAND gate.

Different logic styles have different electrical characteristics. The selection of logic style affects the speed and power consumption. In most cases, the standard CMOS logic is used for speed and power trade-off. In some cases other logic styles, like complementary pass-transistor logic (CPL) is efficient.

Transistor sizing affects both delay and power consumption. Generally, a gate with smaller size has smaller capacitance and consumes less power. To minimize the transistor sizes and meet the speed requirement is a trade-off. Typically, the transistor sizing uses static timing analysis to find out those gates (whose slack time is larger than 0) to be reduced. The transistor sizing is generally applicable for different technologies.

2.3 Summary

Several approaches to reduce the power consumption have been briefly discussed. Below we summarize some of the most commonly used low power techniques.

- Reduce the number of operations. The selection of algorithm and/or architecture has significant impact on the power consumption.
- Power supply voltage scaling. The voltage scaling is an efficient way to reduce the power consumption. Since the throughput is reduced as the voltage is reduced, this may need to be compensated by using parallel and/or pipelining techniques.
- I/Os between chips can consume large power due to the large capacitive loads. Reducing the number of chips is a promising approach to reduce the power consumption.
- Power management. In many systems, the most power consuming parts are often idle. For example, in a lap-top computer, the portion of display and hard disk could consume more than 50% of the total power consumption. Using power management strategies to shut down these components when they are idle for a long time can achieve good power saving.
- Reducing the effective capacitance. The effective capacitance can be reduced by several approaches, for example, compact layout and efficient logic style.
- Reduce the number of transitions. To minimize the number of transitions, especially the glitches, is important

Chapter

3

LOW POWER
Image Compression Techniques

DCT Based Image Compression

Discrete cosine transform (DCT) [39, 44] is widely used in image processing, especially for compression. Some of the applications of two-dimensional DCT involve still image compression and compression of individual video frames, while multidimensional DCT is mostly used for compression of video streams. DCT is also useful for transferring multidimensional data to frequency domain, where different operations, like spread-spectrum, data compression, data watermarking, can be performed in easier and more efficient manner. A number of papers discussing DCT algorithms is available in the literature that signifies its importance and application.

Hardware implementation of parallel DCT transform is possible, that would give higher throughput than software solutions. Special purpose DCT hardware decreases the computational load from the processor and therefore improves the performance of complete multimedia system. The throughput is directly influencing the quality of experience of multimedia content. Another important factor that influences the quality is the finite register length effect that affects the accuracy of the forward-inverse transformation process.

Hence, the motivation for investigating hardware specific DCT algorithms is clear. As 2-D DCT algorithms are the most typical for image compression, the main focus of this chapter will be on the efficient hardware implementations of 2-D DCT based compression by decreasing the number of computations, increasing the accuracy of reconstruction, and reducing the chip area. This in return reduces the power consumption of the compression technique. As the number of applications that require higher-dimensional DCT algorithms are growing, a special attention will be paid to the algorithms that are easily extensible to higher dimensional cases.

The JPEG standard has been around since the late 1980's and has been an effective first solution to the standardisation of image compression. Although JPEG has some very useful strategies for DCT quantisation and compression, it was only developed for low compressions. The 8×8 DCT block size was chosen for speed (which is less of an issue now, with the advent of faster processors) not for performance. The JPEG standard will be briefly explained in this chapter to provide a basis to understand the new DCT related work.

This chapter contains three new and different approaches for using the DCT. The first uses a 'Energy Quantisation' method for compression. The aim of this energy quantisation and then modified energy quantisation techniques is to reduce the quantisation complexity in terms of hardware implementation. Finally, a 'Rule-based Energy Quantisation' method is presented that automatically sets the threshold value for the quantization of the transformed DCT coefficients. The main aim behind these two proposed schemes is to reduce the hardware requirement for quantisation and dequantisation, and to reduce the use of memory, to reduce the power consumption.

3.1 JPEG Compression

The JPEG (Joint Photographic Experts Group) standard [32,50] has been around for some time and is the only standard for lossy still image compression. There are quite a lot of interesting techniques used in the JPEG standard and it is important to give an overview of how JPEG works. There are several variations of JPEG, but only the 'baseline' method is discussed here.

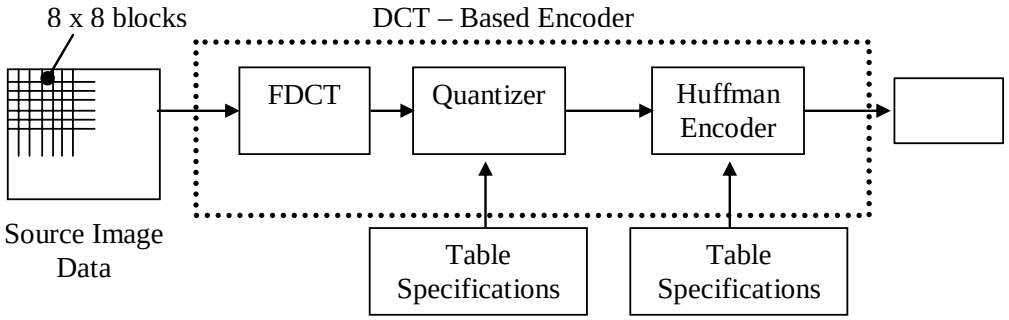


Figure 3.1: JPEG Encoder

As shown in the figure 3.1, the image is first partitioned into non-overlapping 8×8 blocks. A Forward Discrete Cosine Transform (FDCT) is applied to each block to convert the spatial domain gray levels of pixels into coefficients in frequency domain. To improve the precision of the DCT the image is 'zero shifted', before the DCT is applied. This converts a $0 \rightarrow 255$ image intensity range to a $-128 \rightarrow 127$ range, which works more efficiently with the DCT. One of these transformed values is referred to as the DC coefficient and the other 63 as the AC coefficients.

After the computation of DCT coefficients, they are normalized with different scales according to a quantization table provided by the JPEG standard conducted by psychovisual evidence. The quantized coefficients are rearranged in a

zigzag scan order for further compressed by an efficient lossless coding algorithm such as runlength coding, arithmetic coding, Huffman coding. The decoding process is simply the inverse process of encoding as shown in figure 3.2.

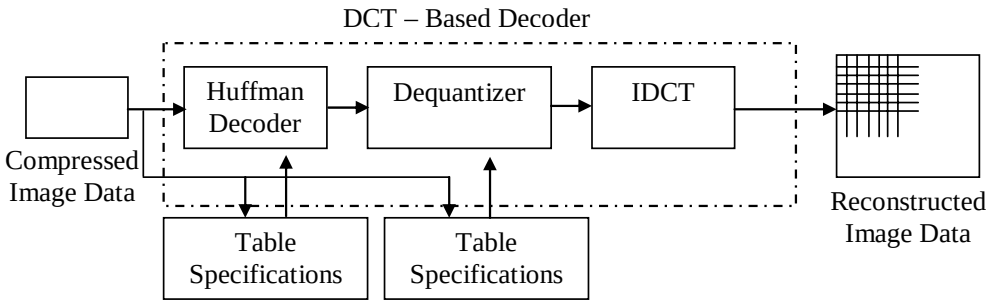


Figure 3.2: JPEG Decoder

3.1.1 DISCRETE COSINE TRANSFORM (DCT)

The DCT is a widely used transformation in transformation for data compression. It is an orthogonal transform, which has a fixed set of (image independent) basis functions, an efficient algorithm for computation, and good energy compaction and correlation reduction properties. Ahmed et al [39] found that the Karhunen Lòeve Transform (KLT) basis function of a first order Markov image closely resemble those of the DCT. They become identical as the correlation between the adjacent pixel approaches to one.

The DCT belongs to the family of discrete trigonometric transform, which has 16 members [44]. The 1D DCT of a $1\times N$ vector $x(n)$ is defined as

$$Y[k] = C[k]\sum_{n=0}^{N-1} x[n]\cos\left[\frac{(2n+1)\,\mathrm{k}\pi}{2N}\right] \tag{3.1}$$

where $k = 0,1,2,..., N-1$ and

$$C[k] = \begin{bmatrix} \sqrt{\frac{1}{\mathrm{N}}} & \text{for k} = 0 \\ \sqrt{\frac{1}{\mathrm{N}}} & \text{for k} = 1,2,..., \mathrm{N}-1 \end{bmatrix}$$

The original signal vector $x(n)$ can be reconstructed back from the DCT coefficients $Y[k]$ using the Inverse DCT (IDCT) operation and can be defined as

$$x[n] = \sum_{k=0}^{N-1} C[k]Y[k]\cos\left[\frac{(2n+1)\,\mathrm{k}\pi}{2N}\right] \tag{3.2}$$

where $n = 0,1,2,..., N-1$

The DCT can be extended to the transformation of 2D signals or images. This can be achieved in two steps: by computing the 1D DCT of each of the individual rows of the two-dimensional image and then computing the 1D DCT of each column of the image. If $x(n_1, n_2)$ represents a 2D image of size $N \times N$, then the 2D DCT of an image is given by:

$$Y[j,k] = C[j]C[k]\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x[m,n]\cos\left(\frac{(2m+1)j\pi}{2N}\right)\cos\left(\frac{(2n+1)k\pi}{2N}\right) \tag{3.3}$$

where $j,k,m,n = 0,1,2,..., N-1$ and

$$C[j] \;\; and \;\; C[k] = \begin{bmatrix} \sqrt{\frac{1}{N}} & \text{for j,k} = 0 \\ \sqrt{\frac{1}{N}} & \text{for j,k} = 1,2,...,N-1 \end{bmatrix}$$

Similarly the 2D IDCT can be defined as

$$x[m,n] = \sum_{j=0}^{N-1}\sum_{k=0}^{N-1} C[j]C[k]Y[j,k]\cos\left(\frac{(2m+1)j\pi}{2N}\right)\cos\left(\frac{(2n+1)k\pi}{2N}\right) \tag{3.4}$$

The DCT is a real valued transform and is closely related to the DFT. In particular, a $N \times N$ DCT of $x(n_1, n_2)$ can be expressed in terms of $2N \times 2N$ DFT of its even-symmetric extension, which leads to a fast computational algorithm. Because of the even-symmetric extension process, no artificial discontinuities are introduced at the block boundaries. Additionally the computation of the DCT requires only real arithmetic. Because of the above properties the DCT is popular and widely used for data compression operation.

The DCT presented in equations (3.1) and (3.4) is orthonormal and perfectly reconstructing provided the coefficients are represented to an infinite precision. This means that when the coefficients are compressed it is possible to obtain a full range of compressions and image qualities. The coefficients of the DCT are always quantised for high compression, but DCT is very resistant to quantisation errors due to the statistics of the coefficients it produces. The coefficients of a DCT are usually linearly quantised by dividing by a predetermined quantisation step.

The DCT is applied to image blocks N x N pixels in size (where N is usually multiple of 2) over the entire image. The size of the blocks used is an important factor since they determine the effectiveness of the transform over the whole image. If the blocks are too small then the images is not effectively decorrelated but if the blocks are too big then local features are no longer exploited.

The tiling of any transform across the image leads to artifacts at the block boundaries. The DCT is associated with blocking artifact since the JPEG standard suffers heavily from this at higher compressions. However the DCT is protected against blocking artifact as effectively as possible, without interconnecting blocks, since the DCT basis functions all have a zero gradient at the edges of their blocks. This means that only the DC level significantly affects the blocking artifact and this can then be targeted.

Ringing is a major problem in DCT operation. When edges occur in an image DCT relies on the high frequency components to make the image shaper. However these high frequency components persist across the whole block and although they are effective at improving the edge quality they tend to 'ring' in the flat areas of the block. This ringing effect increases, when larger blocks are used, but larger blocks are better in compression terms, so a trade off is usually established.

3.1.2 QUANTISATION

In lossy image compression the transformation decompose the image into uncorrelated parts projected on orthogonal basis of the transformation. These basis are represented by eigenvectors which are independent and orthogonal in nature. Taking inverse of the transformed values will result in the retrieval of the actual image data. For compression of the image, the independent characteristic of the transformed coefficients are considered, truncating some of these coefficients will not affect the others. This truncation of the transformed coefficients is actually the lossy process involved in compression and known as quantization [43]. So we can say that DCT is perfectly reconstructing, when all the coefficients are calculated and stored to their full resolution. For high compression, the DCT coefficients are normalized by different scales, according to the quantization matrix as shown in the Table-3.1 provided by JPEG standard, which is designed by conducting some psycho visual

evidence. The JPEG quantizer is a bank of 64 linear (uniform) quantizers, one for each DCT coefficients as shown in figure 3.3. The i'th quantizer is evaluated as

$$Y_i = Round\left(\frac{X_i}{Q_i}\right) \tag{3.5}$$

where Q_i is the i'th quantization step size, X_i is the input where as Y_i is the scaled and quantized version of X_i. The JPEG decoder dequantizes Y_i to obtain a quantized version of X_i using $X' = Q_i * Y_i$.

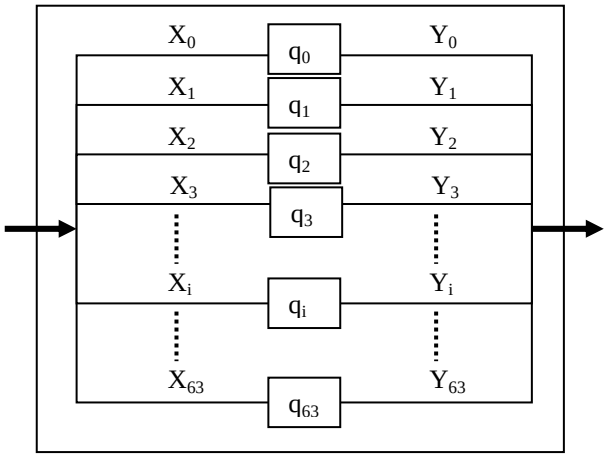


Figure 3.3: JPEG Quantiser

Table 3.1: JPEG Quantisation matrix

16	11	10	16	24	40	51	61	80	60	50	80	120	200	255	255
12	12	14	19	26	58	60	55	55	60	70	95	130	255	255	255
14	13	16	24	40	57	69	56	70	65	80	120	200	255	255	255
14	17	22	29	51	87	80	62	70	85	110	154	255	255	255	255
18	22	37	56	68	109	103	77	90	110	185	255	255	255	255	255
24	35	55	64	81	104	113	92	120	175	255	255	255	255	255	255
49	64	78	87	103	121	120	101	245	255	255	255	255	255	255	255
72	92	95	98	112	100	103	99	255	255	255	255	255	255	255	255
MAT1								MAT2							

Vector quantization, (VQ) mainly used for reducing or compressing the image data [42]. Application VQ on images for compression started from early 1975 by Hilbert mainly for the coding of multispectral Landsat imaginary. Many VQ techniques applied for image compression has been reported [41 - 45]. The main problem with existing VQs is preparation of the code book [43, 45]. In VQ

techniques creating the code book is a complicated process which has to be designed mainly to preserve the low frequency coefficients, and most of the high frequency coefficients are discarded, which results in the loss of edges of the image. The encoder search the whole code book to identify the nearest matching vector template to an input vector, which also takes a lot of time.

3.1.3 CODING

After the DCT coefficients have been quantised, the DC coefficients are DPCM coded and then they are entropy coded along with the AC coefficients. The quantised AC and DC coefficient values are entropy coded in the same way, but because of the long runs in the AC coefficient, an additional run length process is applied to them to reduce their redundancy.

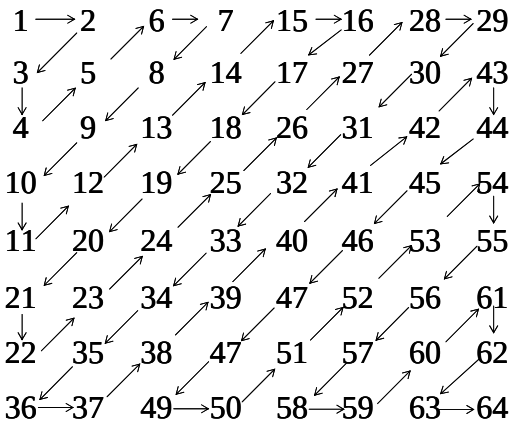


Figure 3.4: Zigzag Scanning

The quantised coefficients are all rearranged in a zigzag order as shown in figure 3.4. The run length in this zigzag order is described by a RUN-SIZE symbol. The RUN is a count of how many zeros occurred before the quantised coefficient and the SIZE symbol is used in the same way as it was for the DC coefficients, but on their AC counter parts. The two symbols are combined to form a RUN-SIZE symbol and this symbol is then entropy coded. Additional bits are also transmitted to specify the exact value of the quantised coefficient. A size of zero in the AC coefficient is used to indicate that the rest of the 8×8 block is zeros (End of Block or EOB).

3.2 SIGNAL ENERGY

The idea of the "size" of a signal is crucial to many applications. It is also nice to know if the signal driving a set of headphones is enough to create a sound. The given example deals with electric signals, which generally vary with different signals with very different tolerances. For this reason, it is convenient to quantify this idea of "size". This leads to the ideas of signal energy and signal power.

Since we often think of signal as a function of varying amplitude through time, it seems to reason that a good measurement of the strength of a signal would be the area under the curve. However, this area may have a negative part. This negative part does not have less strength than a positive signal of the same size. This suggests squaring the signal or taking its absolute value, then finding the area under that curve. It turns out that what we call the energy of a signal is the area under the squared signal. The original signal and the energy of the signal is shown in figure 3.5, and equation 3.6 shows the formula for the calculation of energy of a continuous signal.

$$E_t = \int_{-\infty}^{\infty} (|\ f(t)\ |^2) dt \tag{3.6}$$

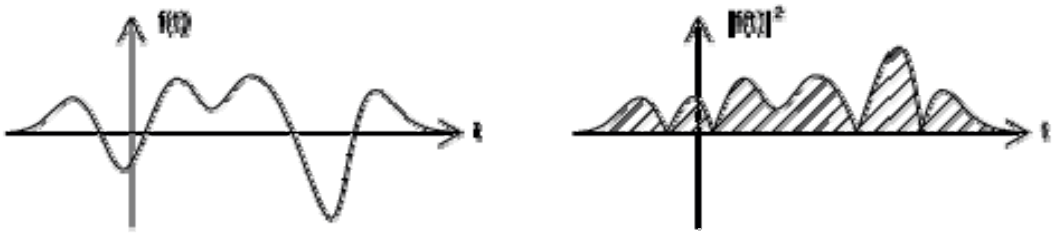


Figure 3.5: Original Signal $f(t)$ and Energy signal $\left|f(t)\right|^2$

3.3 ENERGY QUANTISATION (EQ)

3.3.1 Technique-1: Energy Quantisation (EQ)

The technique of energy calculation discussed in the previous section has been applied to calculate the energy of the transformed coefficients of the image block, where each transformed pixel values are considered as the amplitude of the image

signal. Taking the square of each transformed coefficients and taking the sum gives rise to the energy content in that block. It means that each signal has some contribution to the total energy. Than a threshold value considered for elimination of the transformed coefficients i.e. if the energy of the transformed coefficient is less than the threshold value than make that zero, otherwise keep the coefficient as it is. The threshold value is considered according to the user requirement, i.e. how much energy of the image user want to save. For higher compression and low quality, less transformed coefficients has to be stored i.e. maximum amount of the energy has to be discarded. For low compression and high quality, maximum amount energy has to be saved.

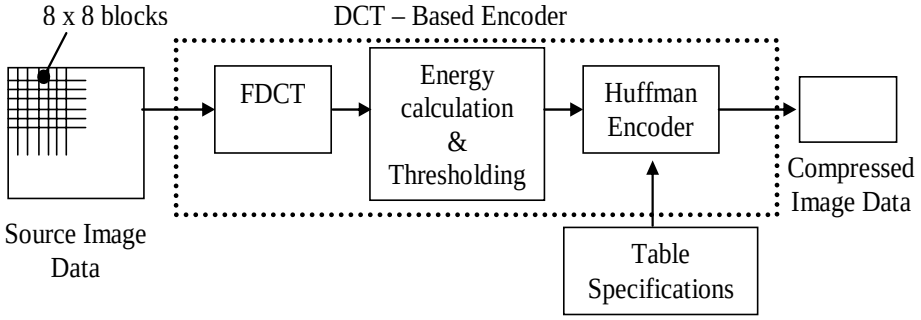


Figure 3.6: Energy Quantisation based Image Compression Encoder

Energy Quantisation based image compression (EQ-IC) technique encoder and is shown in figure 3.6. Figure 3.7, illustrates the proposed energy quantization technique briefly by taking a sample 8 × 8 block. Figure 3.7 (a), is an 8 × 8 block of 8-bit samples taken arbitrarily from a real image. The sample to sample variation is very high, which indicates the existence of high spatial frequency. The actual samples level sifted by subtracting 128 from each sample, as DCT is designed to work on sample values ranging form -127 to 128. The samples are than passed through the FDCT block, for converting the samples to frequency domain. Figure 3.7 (b), shows the resulting DCT coefficients, where it's very clear that most of the coefficients are very small. Figure 3.7 (c), shows the quantized DCT coefficients, normalized by the quantization table (Table -1). Figure 3.7 (d) shows the DCT coefficients quantized by using the proposed algorithm. First the normalized energy of the transformed coefficients is calculated using the following equation,

$$E_n = \frac{1}{MN}\sum_{m=0}^{M}\sum_{n=0}^{N} x(m,n)^2. \tag{3.7}$$

Where M and N are the width and length of the sample block and $x(m,n)$ is the transformed samples. Than according to the threshold value, i.e. a measure to know the contribution of the transformed sample to the normalized energy is considered.

154	123	123	123	123	123	123	136
192	180	136	154	154	154	136	110
254	198	154	154	180	154	123	123
239	180	136	180	180	166	123	123
180	154	136	167	166	149	136	136
128	136	123	136	154	180	198	154
123	105	110	149	136	136	180	166
110	136	123	123	123	136	154	136

(a) Source Image Sample

162.3	40.6	20.0	72.3	30.3	12.5	-19.7	-11.5
30.5	108.4	10.5	32.3	27.7	-15.5	18.4	-2.0
-94.2	-60.1	12.3	-43.4	-31.3	6.1	-3.3	7.2
-38.6	-83.6	-5.4	-22.2	-13.5	15.5	-1.3	3.5
-31.3	17.9	-5.5	-12.4	14.3	-5.9	11.5	-6.0
-0.9	-11.8	12.8	0.2	28.1	12.6	8.4	2.9
4.6	-2.4	12.2	6.6	-18.7	-12.8	7.0	12.1
-9.9	11.2	7.8	-16.3	21.5	0.1	5.9	10.7

(b) Forward DCT coefficients

10	4	2	5	1	0	0	0
3	9	1	2	1	0	0	0
-7	-5	1	-2	-1	0	0	0
-3	-5	0	-1	0	0	0	0
-2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(c) JPEG Quantized Coefficients

10	3	1	5	2	0	-1	0
2	7	0	2	2	0	0	0
-6	-4	0	-3	-2	0	0	0
-2	-5	0	-1	0	0	0	0
-2	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0

(d) Energy Quantized Coefficients
30% of Normalized Discarded

149	134	119	116	121	126	127	128
204	168	140	144	155	150	135	125
253	195	155	166	183	165	131	111
245	185	148	166	184	160	124	107
188	149	132	155	172	159	141	136
132	123	125	143	160	166	168	171
109	119	126	128	139	158	168	166
111	127	127	114	118	141	147	135

(e) Reconstructed image of JPEG Dequantized Coefficients

149	127	113	132	139	122	123	128
195	177	148	146	153	148	134	109
253	194	152	173	184	150	122	110
241	183	142	166	183	154	128	115
188	154	128	149	171	163	152	139
139	135	125	138	158	168	173	162
123	118	116	136	148	148	160	168
101	128	126	113	118	142	156	137

(f) Reconstructed image of energy Dequantized Coefficients

Figure 3.7: Energy Quantisation Decoding and Encoding example

For this example the threshold value is taken as 30% of the normalized energy. For higher compression this threshold value has to be increased. The DCT coefficient can be saved as it is, but the transformed coefficients/samples have been normalized to decrease the amplitudes. At the decoder the compressed data is dequantized and passed through the IDCT operation to reconstruct the image. Figure 3.7 (e) shows the reconstructed samples, after dequantizing the compressed samples using the quantization table.

The decoding technique is shown in figure 3.8. The reconstructed pixel using the energy dequantisation is shown in Figure 3.7 (f). From the reconstructed image

pixel values it is clear that some of the high frequency components are preserved. This indicates that the edge property of the image is preserved.

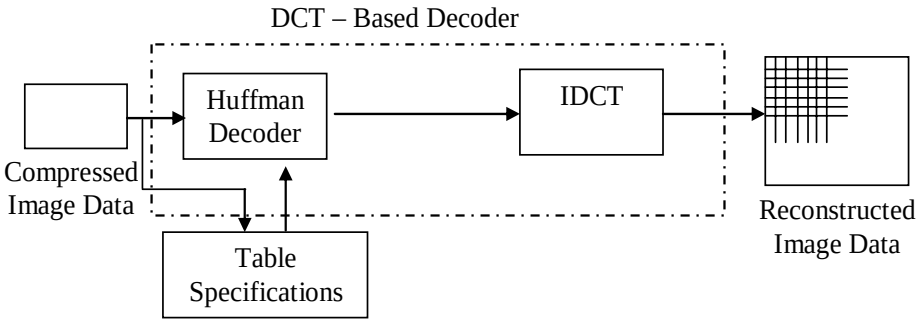


Figure 3.8: Energy Quantisation based Image Compression Decoder

3.3.2 Technique-II : Modified Energy Quantisation (MEQ)

In case of the second method i.e. Modified energy quantisation the energy is calculated by taking the absolute value of the transformed coefficients as shown in figure 3.9. By performing the absolute operation all the transformed coefficients are converted into some positive amplitude. The area under this positive coefficients will give the energy of the signal. It means that each signal amplitude represents the energy of that signal. A threshold value has to be considered for elimination of the transformed coefficients i.e. if the energy of the transformed coefficient is less than the threshold value than make that zero, otherwise keep the coefficient as it is. Mean of the signal is used to set the threshold value for truncation of the transformed coefficients as shown in equation (3.8).

$$A_m = \frac{1}{MN} \sum_{m=0}^{M} \sum_{n=0}^{N} \left| x(m,n) \right| \tag{3.8}$$

where M and N are the width and length of the sample block and $x(m,n)$ is the transformed samples. A_m is the default threshold value. To sift the level of threshold value A_m is multiplied by some constant i.e. for higher compression and low quality of reconstruction A_m should be multiplied by some integer greater then 1, for low compression and high quality A_m should be divided by some integer.

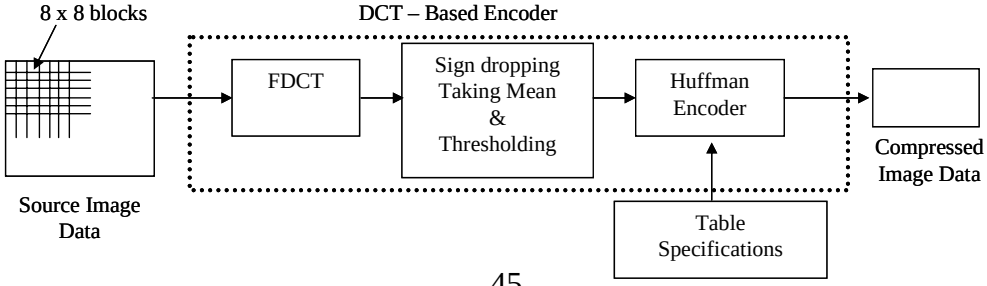


Figure 3.9: Modified Energy Quantisation based Image Compression Decoder

The threshold value is considered according to the user requirement, i.e. how much energy of the image user want to save. For higher compression and low quality, less transformed coefficients has to be stored i.e. maximum amount of the energy has to be discarded. For low compression and high quality, maximum amount energy has to be saved. The decoding procedure is same as the previous technique as shown in figure 3.8.

3.3.3 Rule Based Energy Quantisation (RBEQ)

Instead of representing knowledge in a relatively declarative, static way, rule-based system represent knowledge in terms of a bunch of rules that tell what should be done or what could be concluded in different situations. A rule-based system consists of a bunch of IF-THEN rules, and some interpreter controlling the application of the rules, given the facts.

Rules in the system represent possible actions to take when specified conditions hold on items in the working memory - they are sometimes called condition-action rules. The conditions are usually patterns that must match items in the working memory, while the actions usually involve adding or deleting items from the working memory. The rule based designing broadly divided into two types: forward chaining design, and backward chaining design. In a forward chaining designer have to start with some initial facts, and keep using the rules to draw new conclusions (or take certain actions) given those facts. In a backward chaining designer have to start with some hypothesis (or goal) to prove, and keep looking for rules that would allow to conclude that hypothesis, perhaps setting new subgoals to prove. Forward chaining design are primarily data-driven, while backward chaining design are goal-driven.

Forward chaining design process is used here for compression of the image. Normalized energy E_n is used as the primary data to set the rules for the quantization of the blocks. Calculation of E_n is already explained by the equation 3.7 in section 3.4.1. As the blocks which have edges contributes in large energy and the blocks

which has no edges contributes to less energy. So this property of the image blocks taken in to account and some rules are formed using this energy values. Here two rules are proposed, one for high compression low quality of reconstruction and other one for low compression high quality of reconstruction, compared to the fixed energy quantization method.

Rule-1 (*RBEQ-1*):

If $E_n \geq 1000$, quantisation value is 10% of E_n

Else if $E_n \geq 700$, quantisation value is 30% of E_n

Else if $E_n \geq 500$, quantisation value is 50% of E_n

Else if $E_n \geq 300$, quantisation value is 60% of E_n

Else quantisation value is 70% of E_n

Rule-2 (*RBEQ-2*):

If $E_n \geq 1000$, quantisation value is 10% of E_n

Else if $E_n \geq 500$, quantisation value is 50% of E_n

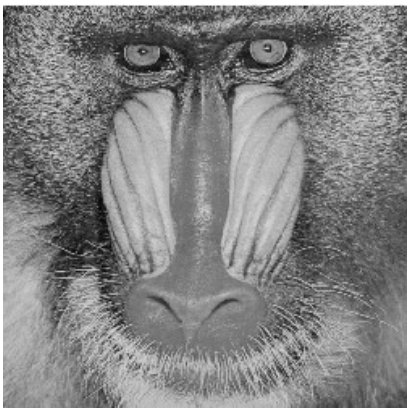
Else quantisation value is 90% of E_n

3.4 SIMULATION RESULTS

Five different standard test images shown in figure 3.10 are used in the simulation of the image compression task using the DCT based JPEG, Energy Quantisation technique (EQ), Modified Energy Quantisation (MEQ) and Rule Based Energy Quantisation (RBEQ-1 & RBEQ-2) respectively. At first the image blocks are passed through the FDCT blocks for conversion into frequency domain. Then the transformed coefficients are allowed to compress using the quantisation operation. After the quantisation the quantized coefficients are coded using some lossless coding algorithm. This completes the compression of the original data and the coded data represent the compressed version of the original sample images.

The performance comparison has been carried out using all the performance indices given in equations (1.2-1.4). These are; PSNR peak signal to noise ratio, energy retained and bits per pixel obtained in the process of compression and reconstruction of signals or images. The same parameters are computed for all the images using the DCT based compression and proposed techniques. Table 3.2 lists the PSNR in dB obtained for JPEG technique and proposed techniques, Table 3.3 lists the bpp (bits per pixel) required for representing one pixel in case of the compressed image. The reconstructed image and the error images for different techniques are shown in figures 3.11 - 3.14.

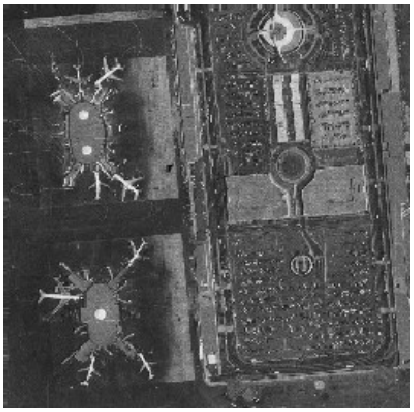
(a) Lena Image



(b) Bobbon Image

(c) Peppers Image

(d) GoldHill Image



(e) Airport Image

Figure 3.10: Standard (512x512) test images used for testing the proposed algorithms

Table 3.2: Comparative results in PSNR (dB) of different Quantisation techniques for five different images

		LENA	BABBON	PEPPERS	GOLDHILL	AIRPORT
JPEG	MAT-1	36.80	28.22	34.74	33.58	28.68
	MAT-2	30.62	23.42	30.09	28.65	24.58
EQ	100%	31.23	25.92	30.53	29.88	25.63
	80%	31.71	26.58	30.97	30.33	26.11
	60%	32.39	27.45	31.55	30.88	26.79
	40%	33.37	28.72	32.34	31.70	27.81
	20%	35.06	30.87	33.63	33.06	29.71
	10%	36.65	32.92	34.75	34.37	31.63
	5%	37.99	34.39	35.61	35.43	33.43
MEQ	5	28.80	20.86	28.04	27.33	22.33
	4	29.90	21.62	29.15	28.19	22.99
	3	31.47	22.88	30.44	29.32	23.93
	2	33.68	24.13	32.17	31.10	25.64
	100%	37.35	29.82	34.80	34.30	29.79
	80%	38.36	31.37	35.45	35.23	31.35
	60%	39.31	33.21	36.10	36.00	33.25
	40%	39.84	34.80	36.49	36.26	34.71
	20%	39.88	35.09	36.54	36.27	34.99
RBEQ	RBEQ1	36.23	37.94	34.37	34.82	31.29
	RBEQ2	36.07	2.001	34.24	34.24	31.11

Table 3.3: Comparative results in bpp (bits per pixcel) of different Quantisation techniques for five different images:

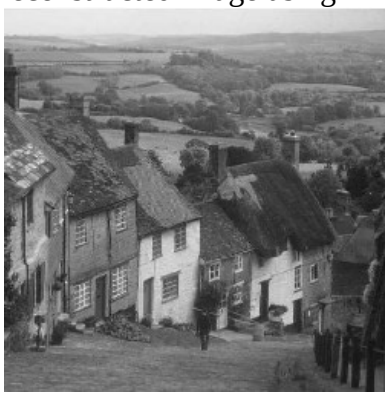
		LENA	BABBON	PEPPERS	GOLDHILL	AIRPORT
JPEG	MAT-1	0.691	1.538	0.700	0.876	1.234
	MAT-2	0.193	0.381	0.197	0.205	0.208
EQ	100%	0.403	1.228	0.395	0.497	0.721
	80%	0.428	1.358	0.424	0.542	0.808
	60%	0.461	1.516	0.461	0.597	0.939
	40%	0.531	1.724	0.520	0.680	1.133
	20%	0.602	2.004	0.632	0.821	1.453
	10%	0.691	2.232	0.750	0.959	1.700
	5%	0.761	2.419	0.874	1.090	1.916
MEQ	5	0.290	0.264	0.267	0.265	0.219
	4	0.340	0.371	0.305	0.324	0.274
	3	0.421	0.593	0.362	0.416	0.391
	2	0.549	1.072	0.475	0.591	0.715
	100%	0.752	1.946	0.804	0.949	1.533
	80%	0.816	2.149	0.898	1.064	1.788
	60%	0.873	2.339	0.992	1.194	2.080
	40%	0.898	2.508	1.063	1.260	2.269
	20%	0.900	2.565	1.076	1.263	2.317
RBEQ	RBEQ1	0.621	2.001	0.627	0.832	1.602
	RBEQ2	0.611	2.062	0.608	0.805	1.576

(a) Reconstructed image using MAT-1

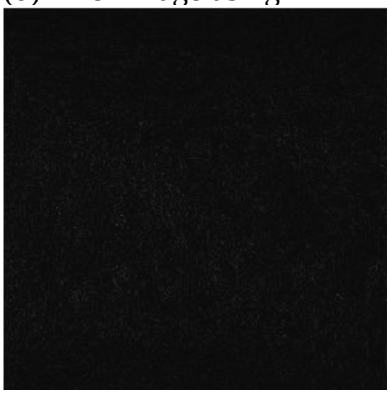
(b) Error image using MAT-1

(c) Reconstructed image using MAT-2

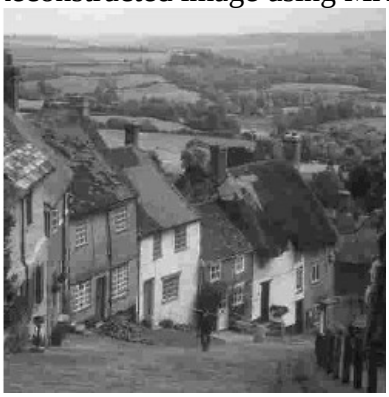
(d) Error image using MAT-2



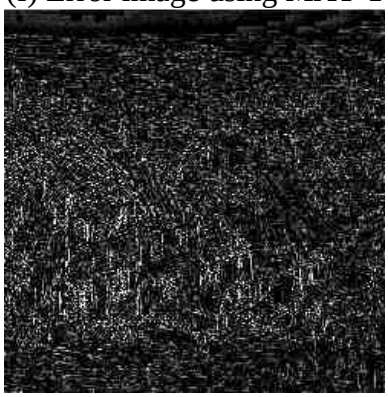
(e) Reconstructed image using MAT-1



(f) Error image using MAT-1

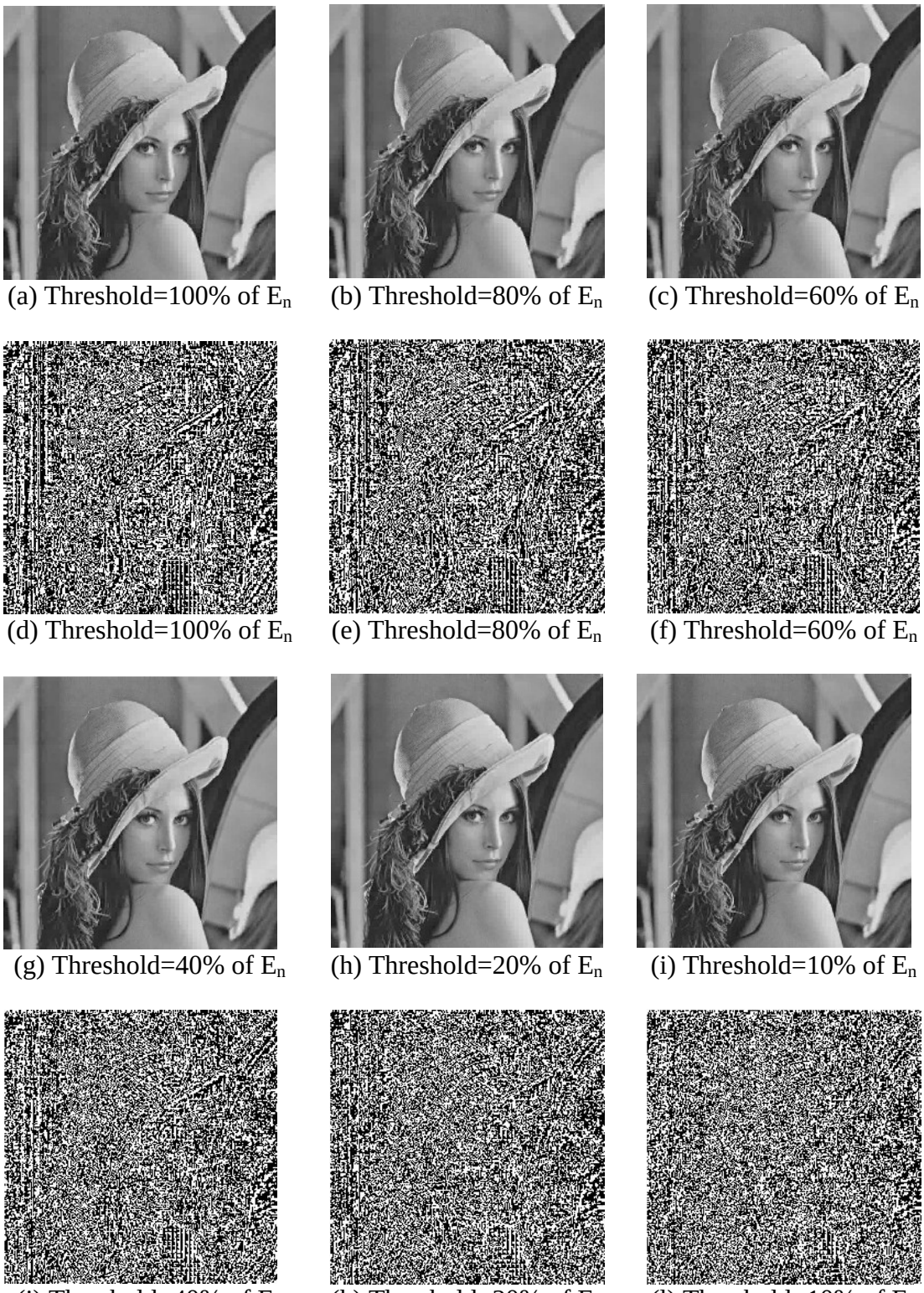


(g) Reconstructed image using MAT-2



(h) Error image using MAT-2

Figure 3.11: Reconstructed image and error images of JPEG quantisation technique



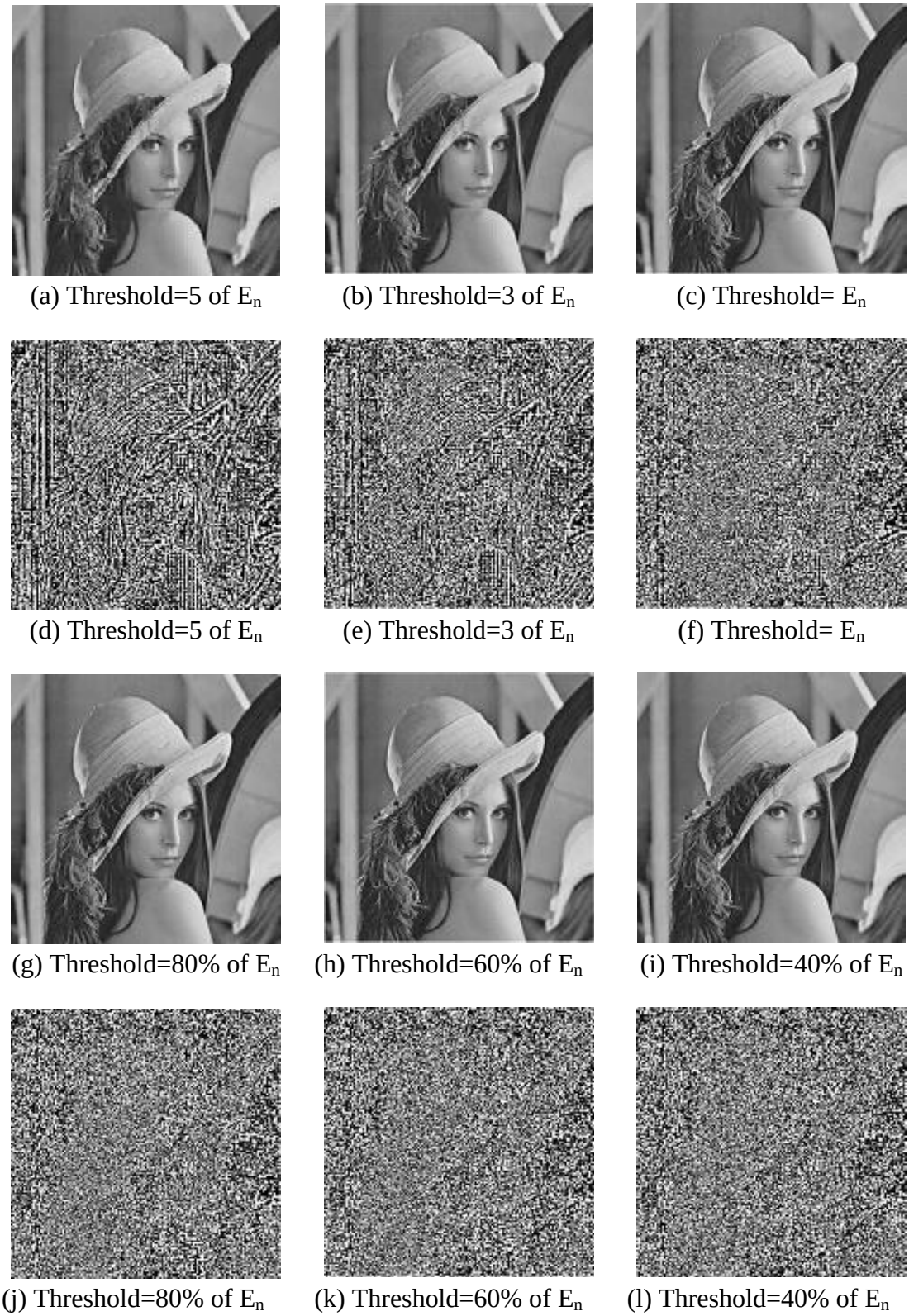
(a) Threshold=100% of E_n (b) Threshold=80% of E_n (c) Threshold=60% of E_n

(d) Threshold=100% of E_n (e) Threshold=80% of E_n (f) Threshold=60% of E_n

(g) Threshold=40% of E_n (h) Threshold=20% of E_n (i) Threshold=10% of E_n

(j) Threshold=40% of E_n (k) Threshold=20% of E_n (l) Threshold=10% of E_n

Figure 3.12: Reconstructed image and error images of EQ technique



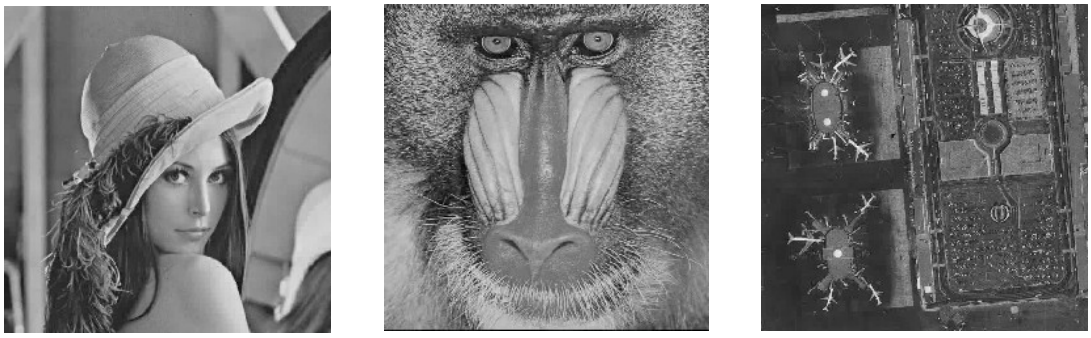
(a) Threshold=5 of E_n (b) Threshold=3 of E_n (c) Threshold= E_n

(d) Threshold=5 of E_n (e) Threshold=3 of E_n (f) Threshold= E_n

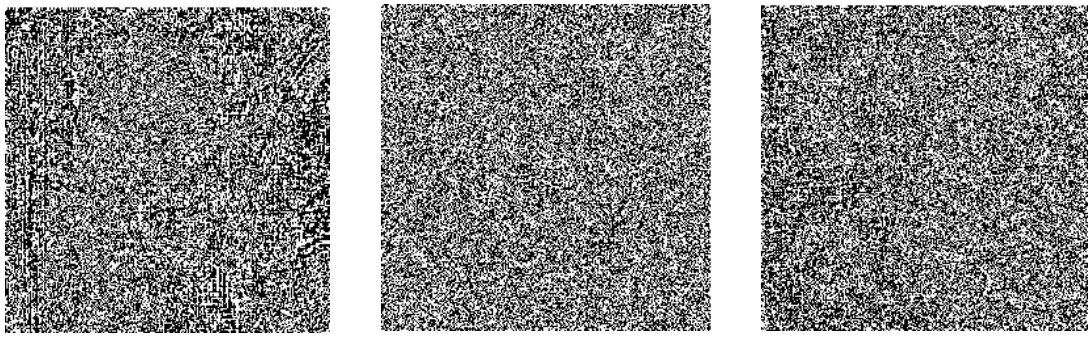
(g) Threshold=80% of E_n (h) Threshold=60% of E_n (i) Threshold=40% of E_n

(j) Threshold=80% of E_n (k) Threshold=60% of E_n (l) Threshold=40% of E_n

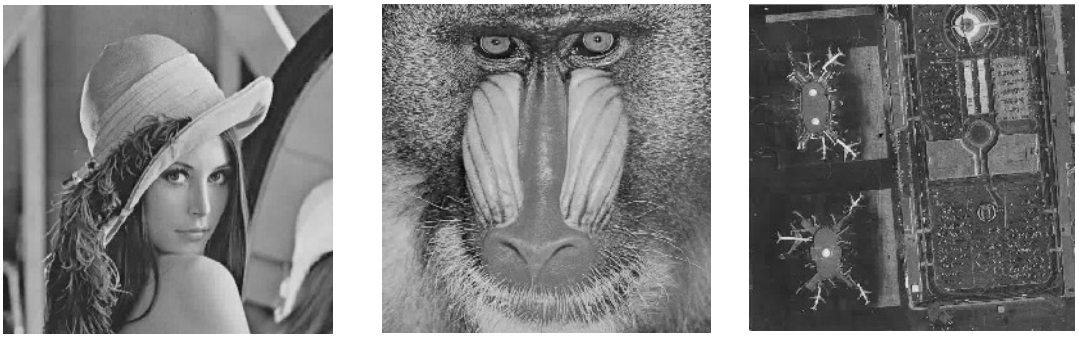
Figure 3.13: Reconstructed image and error images of MEQ technique



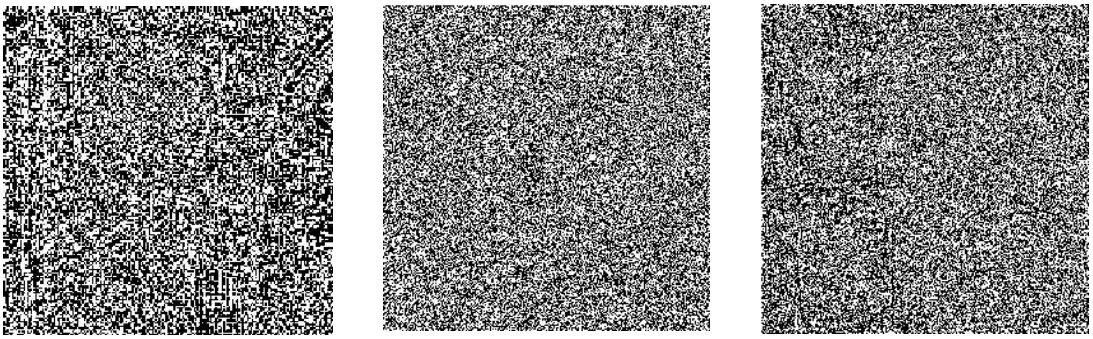
(a) Using RBEQ-1 (b) Using RBEQ-1 (c) Using RBEQ-1



(d) Using RBEQ-1 (e) Using RBEQ-1 (f) Using RBEQ-1



(g) Using RBEQ-2 (h) Using RBEQ-2 (i) Using RBEQ-2



(j) Using RBEQ-2 (k) Using RBEQ-2 (l) Using RBEQ-2

Figure 3.14: Reconstructed image and error images of RBEQ technique

3.5 Summary and Discussion

In this Chapter we have employed the DCT based JPEG image compression technique. Three new quantisation techniques are proposed that can replace the existing vector quantisation technique. Simulation results indicate that the proposed techniques achieve nearly same PSNR value as that of the existing JPEG technique. The third rule based energy quantisation (RBEQ) technique improves the performance of image compression by making it intelligent to select the threshold value for each block separately according to the amount of energy content of that block.

Chapter

4

LOW POWER
Image Compression Techniques

DHT Based Image Compression

JPEG [32, 50] has been created out of the collaboration between the consultative Committee on International Telephone and Telegraph (CCITT) and the International Standard Organization (ISO) to establish an International standard for image compression. It is based on the transform coding using DCT. The JPEG standard describes a family of image compression techniques for continuous tone (gray-scale) still images. Because of the amount of data involved and the psychovisual redundancy present in the images, JPEG employs a lossy compression scheme based on the transform coding. The JPEG defines four operation modes: sequential DCT based mode, sequential lossless mode, progressive DCT based mode, and hierarchical mode. The simplest sequential algorithm among the DCT based JPEG referred to as JPEG baseline, is widely used compression algorithm in the JPEG family.

4.1 JPEG Baseline Algorithm

Baseline algorithm is also called the sequential algorithm. The JPEG baseline data compression scheme can be summarized in the following three steps. The block diagram of the conventional JPEG framework has been shown in figure 4.1.

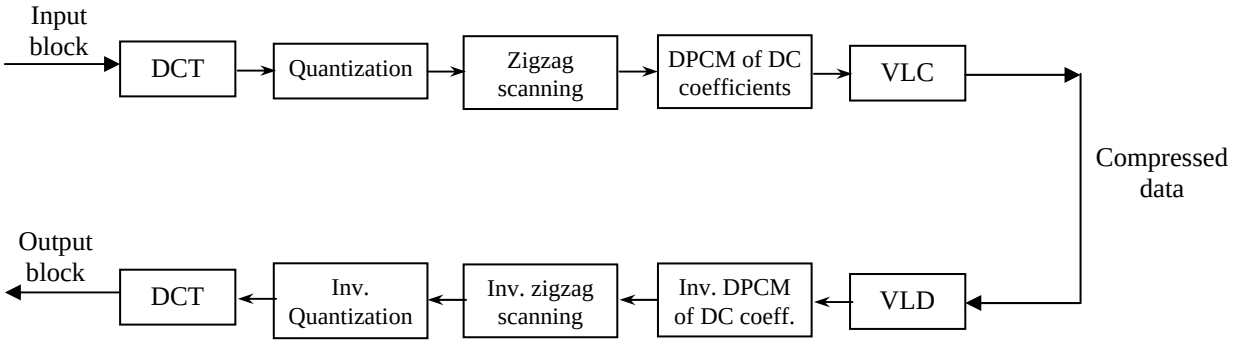


Figure 4.1: Schematic block diagram of the conventional JPEG scheme of data compression

4.1.1 DCT computation

Initially the whole image is subdivided into 8×8 size of pixel blocks. Each pixel is then level shifted by subtracting 2^{n-1}, where 2^n is the maximum number of

given levels. That is for 8-bit images we subtract 128 from each pixel in an attempt to remove the DC level of each block. The 2D DCT of each block is then computed.

4.1.2 Quantisation of the DCT coefficients

The DCT coefficients thus obtained are then thresholded using a Quantisation Matrix (QM) of size (8×8) [4.2, 4.3] and reordered using zigzag scanning to form a 1D sequence of 64 quantised coefficients. The first one is the DC coefficient and the rest 63 coefficients are the AC coefficients. The effect of dividing by such a QM would be to favour the low frequencies. The QM can be scaled to provide a variety of compression levels. The entries of the QM are usually determined according to the psychovisual considerations. A typical QM might be an integer multiple of a matrix such as in [9]. The multiplier might be any positive integer. The larger the multiplier, the better is the compression and accordingly a degradation in the reconstructed image is noticed. Also to reduce the psychovisual redundancy in image, JPEG incorporates the characteristic of the human visual system into the compression process through the specification of QM. It is known that the frequency response of the human visual system drops off with increasing spatial frequency. Further this drop off is faster in the two chrominance channels. As a result, the JPEG allows specification of two QM, one for the luminance and another for the two chrominance channels to allocate more bits for the representation of coefficients, which are visually more significant [54] Thus the choice of the QM and multiplier greatly influences the performance of the method.

4.1.3 Coding of the quantised coefficients

The nonzero AC coefficients are Huffman coded using the Variable length code (VLC) that defines the value of the coefficients and the number of preceding zeros. The DC coefficient of each block is coded using the differential pulse code modulation (DPCM) relative to the DCT coefficients of the previous block.

4.1.4 Reconstruction of the original image

The reconstruction process of this scheme can be obtained by reversing all the forward processes, like Variable Length Decoding, Inverse DPCM, Inverse quantization and the inverse DCT algorithm.

4.2 Motivation

The JPEG conventionally employs the DCT as the transform block in its framework for image compression. The performance of the JPEG is quite satisfactory. In the DSP literature many other efficient transforms such as Good Winograd Transform (GWT) [34, 46] and DHT are being used. Similarly the DHT is a real valued transform whose forward and inverse transforms are same except for an inclusion of a scale factor in the inverse transform. Besides, the DHT can compute both convolution and the DFT efficiently [46]. The memory requirement to compute both the forward and inverse DHT is about 50% as those of the DCT. In the present work we have attempted the use of the DHT in JPEG to find out whether the new version of the JPEG can provide equivalent performance as that of the conventional one. The choice of QM and the multiplier greatly influence the performance of the JPEG scheme of image compression and reconstruction. While replacing the DHT in the JPEG framework, care should be taken in choosing proper QM. The same QM shown in Table-3.1 is used for testing the results. But these QM were unable to give proper results as they are designed to preserve data which follows the zigzag scan order as shown in figure 3.4. DHT transformed coefficients do not follows the zigzag scanning; instead they follow a special scanning order as shown in figure 4.2. So the designing of the QM matrix is quite difficult in this case. To eliminate these difficulties the quantisation techniques proposed in chapter one used and tested for different images to examine whether the newer version has any advantage over the conventional compression scheme or not.

1	2	4	6	8	10	12	14
3	16	17	19	21	23	25	27
5	18	29	30	32	34	36	38
7	20	31	40	41	43	45	47
9	22	33	42	49	50	52	54
11	24	35	44	51	56	57	59
13	26	37	46	53	58	61	62
15	28	39	48	55	60	63	64

Figure 4.2: Scanning order of DHT coefficient

4.3 Discrete Hartley Transform (DHT)

4.3.1 1D DHT

In 1942, R. V. L. Hartley [61] proposed a real integral transform for the analysis of transmission problem. Based on that integral transform, Bracewell [59] proposed a real valued discrete transform called the DHT. The DHT is a real valued alternative to the DFT as the even and odd parts of the DHT of a real valued sequence are same as the real and negative imaginary parts of the corresponding DFT components. The DHT of an N-point real valued sequence $x(n)$ is defined as [60]

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)\ \mathrm{cas}\left(2\pi \mathrm{kn/N}\right) \tag{4.1}$$

and the inverse transform is

$$x(n) = \sum_{k=0}^{N-1} X(k)\ \mathrm{cas}\left(2\pi \mathrm{kn/N}\right) \tag{4.2}$$

where,

$$\mathrm{cas}(2\pi kn/N) = \cos(2\pi kn/N) + \sin(2\pi kn/N)$$

for $k, n = 0,1,2,..., N-1$

Equations (4.1) and (4.2) may respectively be expressed in the matrix product form as:

$$X = (1/N)C_N x, \text{ and } \quad x = C_N X \tag{4.3}$$

where x is the $N \times 1$ input vector and X is the $N \times 1$ DHT vector. C_N is the $N \times N$ Hartley matrix whose elements are given by

$$C_N(k,n) = \cos(2\pi kn/N) + \sin(2\pi kn/N) \tag{4.4}$$

for $k, n = 0,1,2,..., N-1$ from (4.4) it can be seen that

$$C_N(k,n) = C_N(n,k) \tag{4.5}$$

and

$$\sum_{n=0}^{N-1} C_N(n,k) C_N(n,k') = N\delta_{kk'} \tag{4.6}$$

where Kronecker delta $\delta_{kk'}$ is given by

$$\delta_{kk'} = \begin{cases} 1 & \text{for k = k}' \\ 0 & \text{for k} \neq \text{k}' \end{cases}$$

from (4.5) and (4.6) it can be found that the Hartley matrix is a nonsingular Harmitian matrix of eigen value $\pm\sqrt{N}$. An important feature of this transform that makes it more advantageous compared to the DFT and DCT is that the inverse transform defines by (4.2) is identical to the forward transform given by (4.1), except a scale factor of $\frac{1}{N}$. Therefore, only one routine may be coded and stored for the forward as well as inverse transform.

4.3.2 2D DHT

Just like 1D, the 2D DHT computation can be developed and has a potential application in the field of image processing [57]. Development of efficient schemes for its fast computation is, therefore, a subject of interest. Many researches have been carried out to solve this problem. Bracewell et al [60] have proposed an efficient algorithm to compute the multidimensional DHT by adding certain number of intermediate arrays where each of the arrays is computed using 1-dimensional fast DHT algorithm. Another scheme of computation of the DHT has been proposed [58], where the computation is based on the prime-factor decomposition. According to this scheme, multidimensional DHT can be computed using a 1D fast DFT algorithm and an 1D fast DHT algorithm. This scheme has been reported to be less compulsive as well as structurally less complex over the earlier scheme. In the JPEG scheme we have incorporated the DHT that has been computed using the prime factor. A brief overview of this algorithm is follows.

The 2D DHT of an array $[x(m,n)]$ of size $M \times N$ may be defined as

$$X(k,l) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x(m,n)\left[\cos 2\pi\left(\frac{km}{M}+\frac{\ln}{N}\right)\right] + \left[\sin 2\pi\left(\frac{km}{M}+\frac{\ln}{N}\right)\right] \tag{4.7}$$

By splitting the arguments of sine and cosine function of (4.7), we get

$$X(k,l) = \sum_{n=0}^{N-1} u(k,n)\left[\cos\frac{2\pi\ln}{N} + \sin\frac{2\pi\ln}{N}\right] + \sum_{n=0}^{N-1} v(k,n)\left[\cos\frac{2\pi\ln}{N} - \sin\frac{2\pi\ln}{N}\right] \tag{4.8}$$

where

$$u(k,n)=\sum_{m=0}^{M-1}x(m,n)\cos\frac{2\pi km}{M} \tag{4.9}$$

$$v(k,n)=\sum_{m=0}^{M-1}x(m,n)\cos\frac{2\pi km}{M} \tag{4.10}$$

It may be noted that $u(k,n)$ and $v(k,n)$, for $k=0,1,...,M-1$, represent the real parts and the negative imaginary parts of M-point DFT of nth column of $[x(m,n)]$, respectively.

Substituting $n=N-n$ on the second sum of (4.8), we can get

$$X(k,l)=\sum_{n=0}^{N-1}w(k,n)\left[\cos\frac{2\pi\ln}{N}+\sin\frac{2\pi\ln}{N}\right] \tag{4.11}$$

where,

$$w(k,n)=u(k,n)+v(k,(N-n)_N) \tag{4.12}$$

The symbol $(\bullet)_N$ denotes the modulo N operation

Equations (4.9) - (4.12) indicate that a 2D DHT of an array of size $M\times N$ may be computed in the following sequences:

i. The M-point DFT of each column of $[x(m,n)]$ be computed.

ii. The real parts of the DFT of the nth column be added with the –ve of corresponding imaginary parts of the DFT of the (N-n)th column, for n=1,2,…,N-1. The real part of the DFT of zeroth column, however, be added with the -ve imaginary part of the DFT of the same column. The results thus obtained, be stored in the corresponding position of $[x(m,n)]$ under its new variable name $[w(m,n)]$.

iii. Finally, the N-point DHT of each row of $[w(m,n)]$ be computed to obtained the desired 2D DHT.

4.4 Energy Quantisation (EQ)

4.4.1 Technique-1: Energy Quantisation (EQ)

This same technique of energy calculation discussed in the previous chapter has been applied to calculate the energy of the transformed coefficients of the image block, where each transformed pixel values are considered as the amplitude of the image signal. Taking the square of each transformed coefficients and taking the sum

gives rise to the energy content in that block. It means that each signal has some contribution to the total energy. Than a threshold value considered for elimination of the transformed coefficients i.e. if the energy of the transformed coefficient is less than the threshold value than make that zero, otherwise keep the coefficient as it is. The threshold value is considered according to the user requirement, i.e. how much energy of the image user want to save. For higher compression and low quality, less transformed coefficients has to be stored i.e. maximum amount of the energy has to be discarded. For low compression and high quality, maximum amount energy has to be saved.

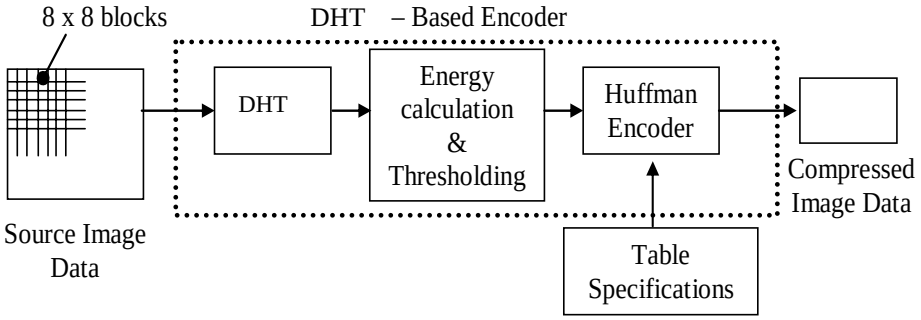


Figure 4.3: Energy Quantisation based Image Compression Encoder

First the normalized energy of the transformed coefficients is calculated using the following equation,

$$E_n = \frac{1}{MN}\sum_{m=0}^{M}\sum_{n=0}^{N} x(m,n)^2. \tag{4.13}$$

Where M and N are the width and length of the sample block and $x(m,n)$ is the transformed samples. Than according to the threshold value, i.e. a measure to know the contribution of the transformed sample to the normalized energy is considered.

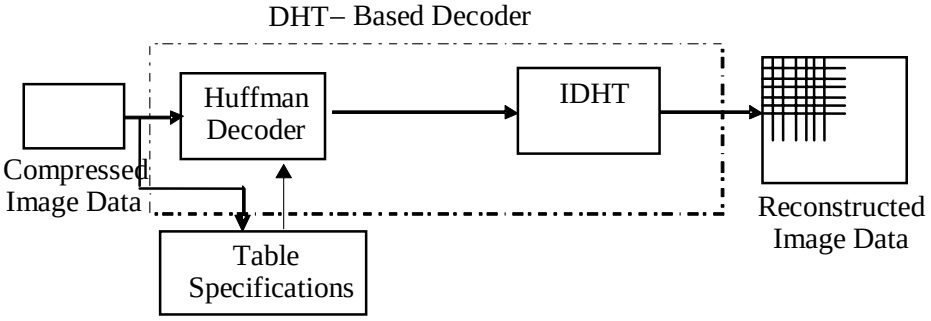


Figure 4.4: Energy Quantisation based Image Compression Decoder

The decoding technique is shown in figure 4.4. From the reconstructed image pixel values it is clear that some of the high frequency components are preserved. This indicates that the edge property of the image is preserved.

4.4.2 Technique-II : Modified Energy Quantisation (MEQ)

In case of the second method i.e. Modified energy quantisation the energy is calculated by taking the absolute value of the transformed coefficients as shown in figure 4.5. By performing the absolute operation all the transformed coefficients are converted into some positive amplitude. The area under these positive coefficients will give the energy of the signal and the amplitude represents the energy of that signal.

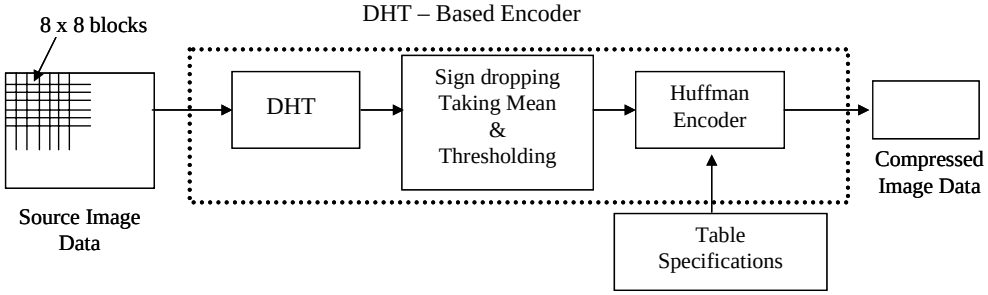


Figure 4.5: Modified Energy Quantisation based Image Compression Decoder

A threshold value has to be considered for elimination of the transformed coefficients i.e. if the energy of the transformed coefficient is less than the threshold value than make that zero, otherwise keep the coefficient as it is. Mean of the signal is used to set the threshold value for truncation of the transformed coefficients as shown in equation (4.15).

$$A_m = \frac{1}{MN} \sum_{m=0}^{M} \sum_{n=0}^{N} \left| x(m,n) \right| \tag{4.15}$$

where M and N are the width and length of the sample block and $x(m,n)$ is the transformed samples. A_m is the default threshold value. To shift the level of threshold value A_m is multiplied by some constant i.e. for higher compression and low quality of reconstruction A_m should be multiplied by some integer greater then 1, for low compression and high quality A_m should be divided by some integer.

The threshold value is considered according to the user requirement, i.e. how much energy of the image user want to save. For higher compression and low

quality, less transformed coefficients has to be stored i.e. maximum amount of the energy has to be discarded. For low compression and high quality, maximum amount energy has to be saved. The decoding procedure is same as the previous technique as shown in figure 4.4.

4.4.3 RULE BASED ENERGY QUANTISATION (RBEQ)

Instead of representing knowledge in a relatively declarative, static way, rule-based system represent knowledge in terms of a bunch of rules that tell what should be done or what could be concluded in different situations. A rule-based system consists of a set of IF-THEN rules, and some interpreter controlling the application of the rules, given the facts.

Rules in the system represent possible actions to take when specified conditions hold on items in the working memory - they are sometimes called condition-action rules. The conditions are usually patterns that must match items in the working memory, while the actions usually involve adding or deleting items from the working memory. The rule based designing broadly divided into two types: forward chaining design, and backward chaining design. In a forward chaining designer have to start with some initial facts, and keep using the rules to draw new conclusions (or take certain actions) given those facts. In a backward chaining designer have to start with some hypothesis (or goal) to prove, and keep looking for rules that would allow concluding that hypothesis, perhaps setting new subgoals to prove. Forward chaining design is primarily data-driven, while backward chaining design are goal-driven.

The energy quantisation rules proposed in chapter 3 is used to threshold the transformed coefficients. Form the simulation result it is very much clear that the performance of the proposed DHT based JPEG technique out performs the existing technique of quantisation. Both the rules are unchanged for testing the proposed technique. These rules can be modified for both better compression and better quality. The two rules are specified

Rule-1 (*RBEQ-1*):

If $E_n \geq 1000$, quantisation value is 10% of E_n

Else if $E_n \geq 700$, quantisation value is 30% of E_n

Else if $E_n \geq 500$, quantisation value is 50% of E_n

Else if $E_n \geq 300$, quantisation value is 60% of E_n

Else quantisation value is 70% of E_n

Rule-2 (*RBEQ-2*):

If $E_n \geq 1000$, quantisation value is 10% of E_n

Else if $E_n \geq 500$, quantisation value is 50% of E_n

Else quantisation value is 90% of E_n

4.5 Simulation Results

In this section we provide simulation results both for the conventional and newer version of the JPEG. Because of the obvious advantage of DHT over DCT, we have replaced the DCT block in the encoder side with the DHT and IDCT block in the decoder side with the DHT having a scale factor of $\left(\frac{1}{N}\right)$.

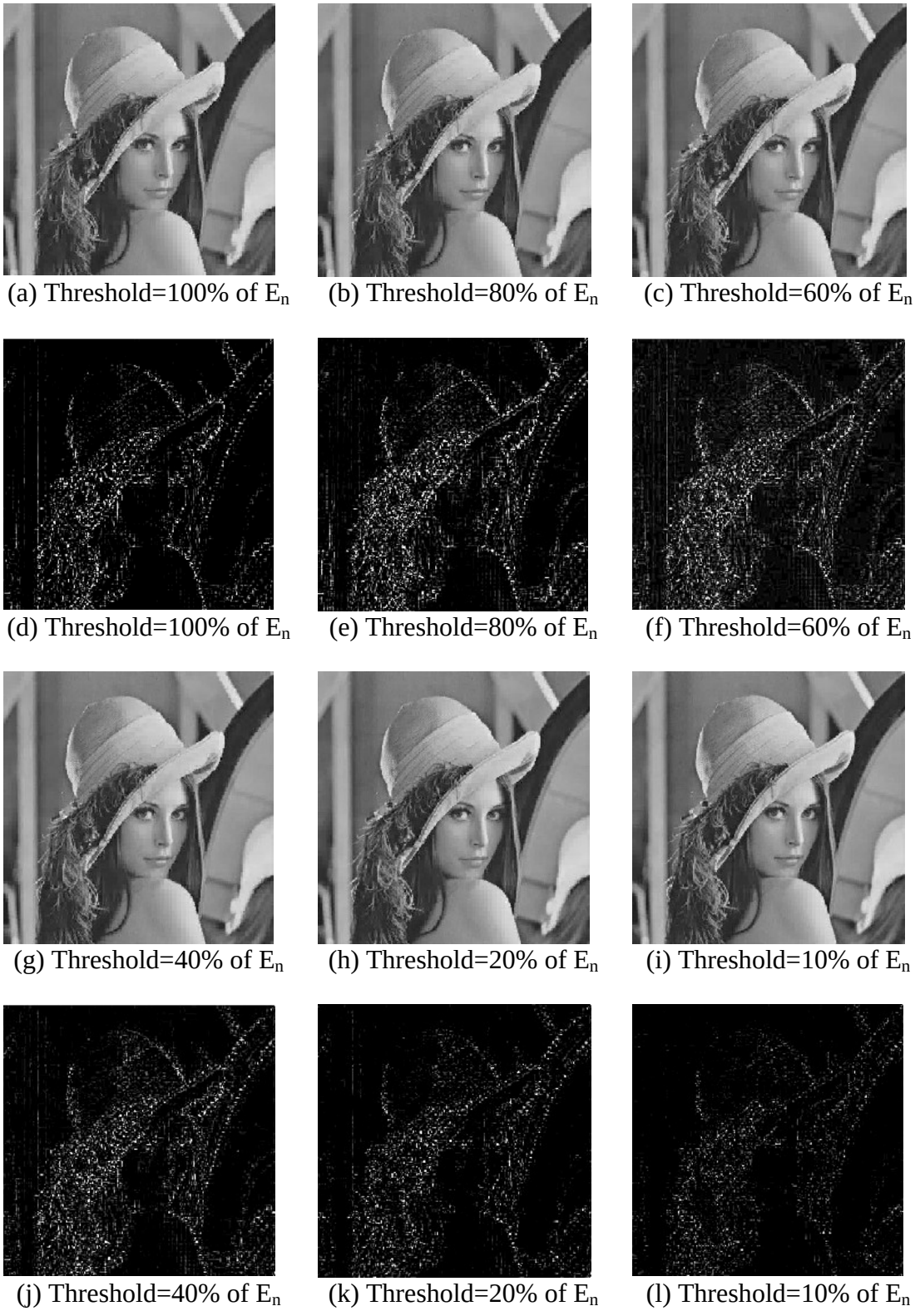
The performance comparison has been carried out using all the performance indices given by (1.2-1.4). These are; PSNR peak signal to noise ratio, energy retained and bits per pixel obtained in the process of compression and reconstruction of signals or images. The same parameters are computed for all the images using the DCT based compression and proposed techniques. Table 4.1 lists the PSNR in dB obtained for JPEG technique and proposed techniques, Table 4.2 lists the bpp (bits per pixel) required for representing one pixel in case of the compressed image. The reconstructed image and the error images for different techniques are shown in figures 4.6 - 4.8.

Table 4.1: Comparative results in PSNR (dB) of different Quantisation techniques for five different images

		LENA	BABBON	PEPPER	GOLDHILL	AIRPORT
EQ	100%	29.83	25.55	29.13	29.22	25.32
	80%	30.35	26.09	29.60	29.68	25.80
	60%	31.03	27.05	30.18	30.24	26.53
	40%	31.92	28.40	31.02	31.11	27.48
	20%	33.59	30.01	32.55	32.69	29.56
	10%	34.98	32.98	33.85	34.00	30.80
MEQ	5	29.90	20.80	26.23	26.78	22.21
	4	27.88	21.56	27.27	27.60	22.79
	3	29.25	22.72	28.54	28.78	23.69
	2	31.33	29.69	30.52	30.69	25.37
	100%	34.98	29.82	33.64	34.06	29.64
	90%	35.41	30.13	34.11	34.71	30.32
	70%	36.66	31.34	34.96	35.76	32.04
	50%	37.68	34.69	35.95	36.06	34.38
	30%	37.91	35.09	36.21	36.07	35.04
RBEQ	RBEQ1	30.04	25.81	29.35	29.45	25.56
	RBEQ2	30.67	26.48	29.86	29.95	26.71

Table 4.2 Comparative results in bpp (bits per pixcel) of different Quantisation techniques for five different images:

		LENA	BABBON	PEPPER	GOLDHILL	AIRPORT
EQ	100%	0.501	0.502	0.502	0.635	0.749
	80%	0.546	0.547	0.547	0.701	0.856
	60%	0.607	0.609	0.609	0.784	1.030
	40%	0.688	0.710	0.710	0.911	1.251
	20%	0.844	0.901	0.901	1.107	1.624
	10%	0.958	1.077	1.077	1.293	1.826
MEQ	5	0.317	0.295	0.284	0.317	0.220
	4	0.398	0.435	0.371	0.413	0.278
	3	0.517	0.687	0.508	0.583	0.417
	2	0.709	1.255	0.774	0.984	0.790
	100%	1.057	2.321	1.152	1.408	1.013
	90%	1.086	2.385	1.197	1.464	2.047
	70%	1.114	2.536	1.266	1.526	2.225
	50%	1.175	2.728	1.319	1.537	2.386
	30%	1.181	2.741	1.328	1.539	2.407
RBEQ	RBEQ1	0.521	1.427	0.522	0.667	0.806
	RBEQ2	0.576	1.588	0.576	0.740	0.950



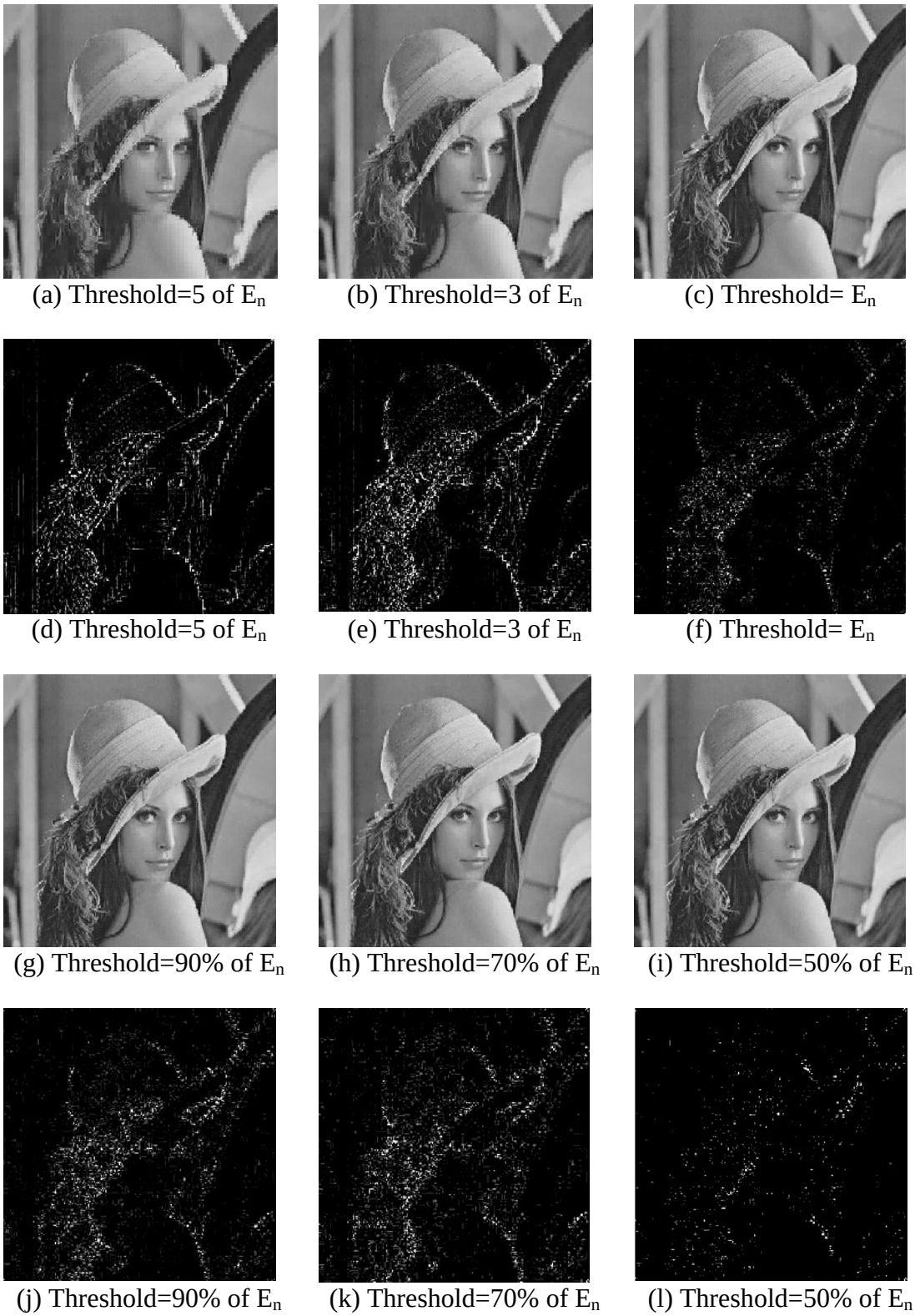
(a) Threshold=100% of E_n (b) Threshold=80% of E_n (c) Threshold=60% of E_n

(d) Threshold=100% of E_n (e) Threshold=80% of E_n (f) Threshold=60% of E_n

(g) Threshold=40% of E_n (h) Threshold=20% of E_n (i) Threshold=10% of E_n

(j) Threshold=40% of E_n (k) Threshold=20% of E_n (l) Threshold=10% of E_n

Figure 4.6: Reconstructed image and error images of EQ technique



(a) Threshold=5 of E_n (b) Threshold=3 of E_n (c) Threshold= E_n

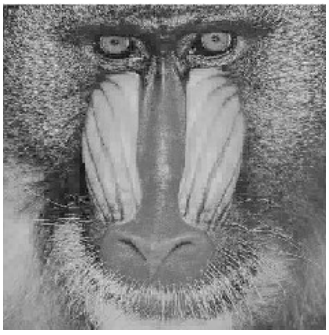
(d) Threshold=5 of E_n (e) Threshold=3 of E_n (f) Threshold= E_n

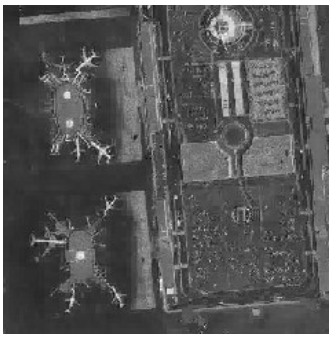
(g) Threshold=90% of E_n (h) Threshold=70% of E_n (i) Threshold=50% of E_n

(j) Threshold=90% of E_n (k) Threshold=70% of E_n (l) Threshold=50% of E_n

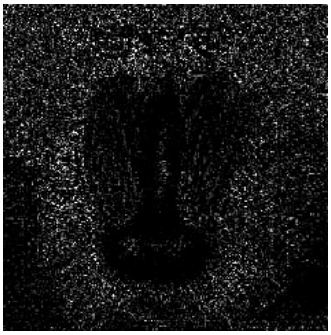
Figure 4.7: Reconstructed image and error images of MEQ technique

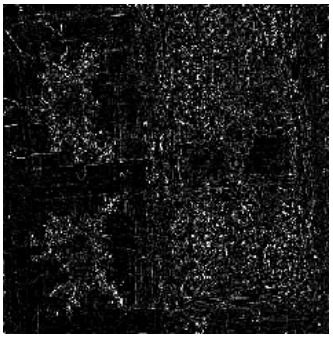

(a) Using RBEQ-1


(b) Using RBEQ-1

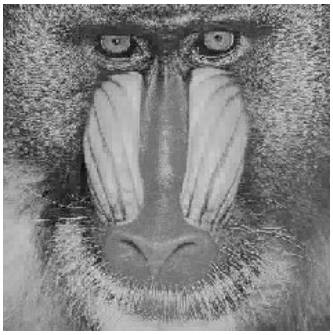

(c) Using RBEQ-1


(d) Using RBEQ-1


(e) Using RBEQ-1

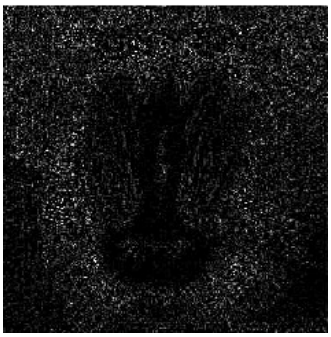

(f) Using RBEQ-1


(g) Using RBEQ-2


(h) Using RBEQ-2


(i) Using RBEQ-2


(j) Using RBEQ-2


(k) Using RBEQ-2

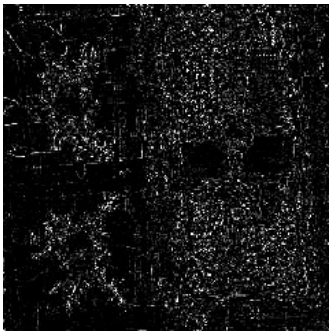

(l) Using RBEQ-2

Figure 4.8: Reconstructed image and error images of RBEQ technique

4.6 SUMMARY

Exhaustive computer simulation results indicate that the DHT based JPEG is a useful alternative to that of a DCT based JPEG because of less storage requirement and improved compression and reconstruction performance. Qualitatively it is found to offer superior performance compared to the conventional JPEG. It is evident from the plots computed from the saving energy and PSNR in dB in the reconstructed images. Besides, there is saving in memory space in employing the DHT rather than the DCT and IDCT.

Chapter

5

LOW POWER
Image Compression Technique

Arithmetic Image Compression

With the increasing use of multimedia technologies, image compression requires higher performance. To address needs and requirements of multimedia and internet applications, many efficient image compression techniques, with considerably different features, have been developed. An image compression technique exploits a common characteristic of most images that is the neighboring picture elements or pixels are highly correlated [9]. It means a typical still image contains a large amount of spatial redundancy in plain areas where adjacent pixels have almost the same values. In addition a still image contains subjective redundancy, which is determined by the properties of Human Visual System (HVS). HVS presents some tolerance to distortion depending upon the image and viewing conditions. Consequently, pixels must not always be reproduced exactly same as the original one but still HVS will not detect the difference between original image and reproduced image [65].

All compression techniques can be basically classified into two main categories: called as lossy compression and lossless compression. Lossy schemes offer compression by a factor of 20 or more, but do not allow exact recovery of original images. In lossless schemes on the other hand compression ratio is limited to a factor of 2, but it allows exact reconstruction of images from the compressed imaged. This method is considered to be useful when finer details of the images are required to be retained for certain reasons as in case of medical and space images, remote sensing images and fine arts etc. Exact lossless recovery is, however, not an essential requirement in many situations because different applications may tolerate different limits of deviation from the original value. A third category of compression technique called as near lossless compression [62 - 79] is, therefore, evolving in the recent years. Near lossless compression method guarantees that difference between a pixel value of the reconstructed image and the value of corresponding pixel in the original image cannot exceed a specified upper limit [66, 69].

Transform domain approach using the discrete cosine transform (DCT) or the discrete wavelet transform (DWT) along with an entropy coding is usually deployed for lossy compression. For lossless and near lossless compression using differential pulse code modulation (DPCM) based on different predictors and error-modeling schemes are popularly used due to their simplicity as well as its efficiency although

wavelet-based filter bank coding and some hybrid schemes are also suggested [36], But, all these methods for near lossless coding cannot be directly extended to lossy region to have a unified approach for compression that can be suitable for variable compression requirement. Recently, a unified coding algorithm is suggested for lossy, lossless as well as the near lossless compression using lossless-DCT (L-DCT) [77]. Since L-DCT maps an integer vector to a vector mat is also expressed by integer it does not introduce reconstruction error when no quantization is performed. L-DCT can therefore be used for lossless compression, and the compression performance of the L-DCT is found to be comparable to that of conventional DCT for lossy as well as near lossless compression.

Based on some simple arithmetic calculation, an efficient Lossless Image Compression technique is proposed. This technique is designed for high quality still image compression, especially PSNR value above 34. This algorithm is most applicable for those images where lossy compression is avoided such as medical and scientific images. The encoding and decoding procedure is very fast.

5.1 Arithmetic Compression (AC)

For the gray level image, the pixel value is usually represented in integer format. The LOG-EXP image compression [74 - 75] is based on the logarithmic number system (LNS) properties, and brings the pixel values to 8 as a result the neighboring difference between the pixels also reduced. Using the proposed different transforms we concentrate the statistical property of the image to a particular range and discard some of the transformed values depending upon the requirement.

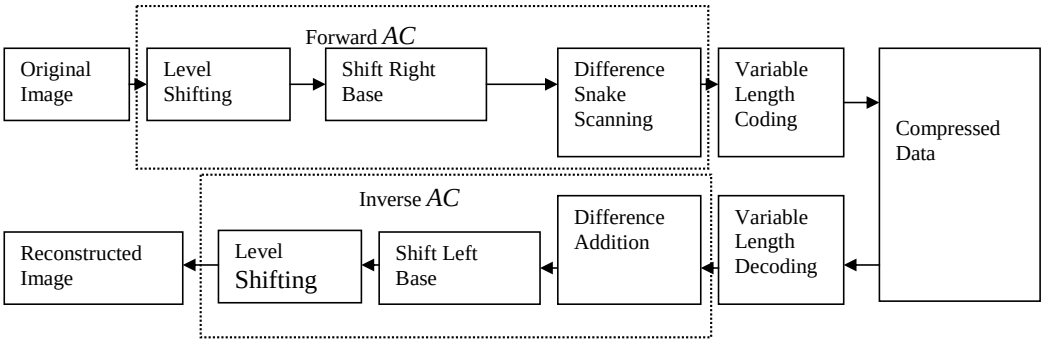


Figure 5.1: Schematic Diagram of Arithmetic Compression (AC) System

The compression and decompression flow of the new arithmetic compression algorithm is shown in figure 5.1. Where for reducing the distribution of the pixel value to locate in a small and continuous range, the pixels are divided by a base number and only the division value is stored and the reminder is truncated. As a result it removes the redundancy of the data, at the same time the large neighboring distance between two pixels reduced to a small value. The neighbor pixels usually represent the same object so the values of the pixels are similar. The neighboring differences will be very small and there may be many similar differences. When the pixels are processed in sequential scan at the time of changes form one line to next line occurs, there will be one large neighboring difference. To avoid such kind of large neighboring difference, the snake scan is used as shown in figure 5.2.

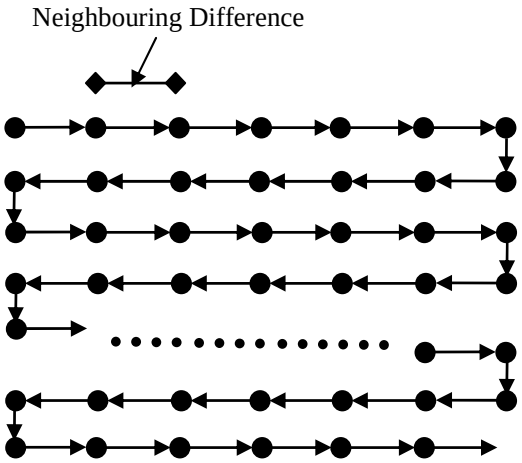


Figure 5.2: Snake Scanning

5.1.1 Forward Arithmetic Compression Algorithm

Step 1: The input image is level shifted by 2P-1 that is, 2P-1 is subtracted from each pixel value, where P is the number of bits used to represent each pixel. Thus 128 is subtracted form each pixel if the image is 8-bit.

Step 2: Divide the level shifted pixel values by the base, where base is any value in between 4 to 16 and take the integer part only, similar as modulo operation. For high quality the base should be small and for high compression the base should be high.

Step 3: Perform snake scan and calculate the difference of the adjacent pixel values as shown in figure 5.2.

Step 4: Perform Variable Run length coding for lossless coding of the decorrelated coefficients.

5.1.2 Inverse Arithmetic Compression Algorithm

Step 1: Perform Run length decoding to get back the pixel values form the lossless coded data.

Step 2: Perform snake scan and calculate the original pixel values by adding the differences.

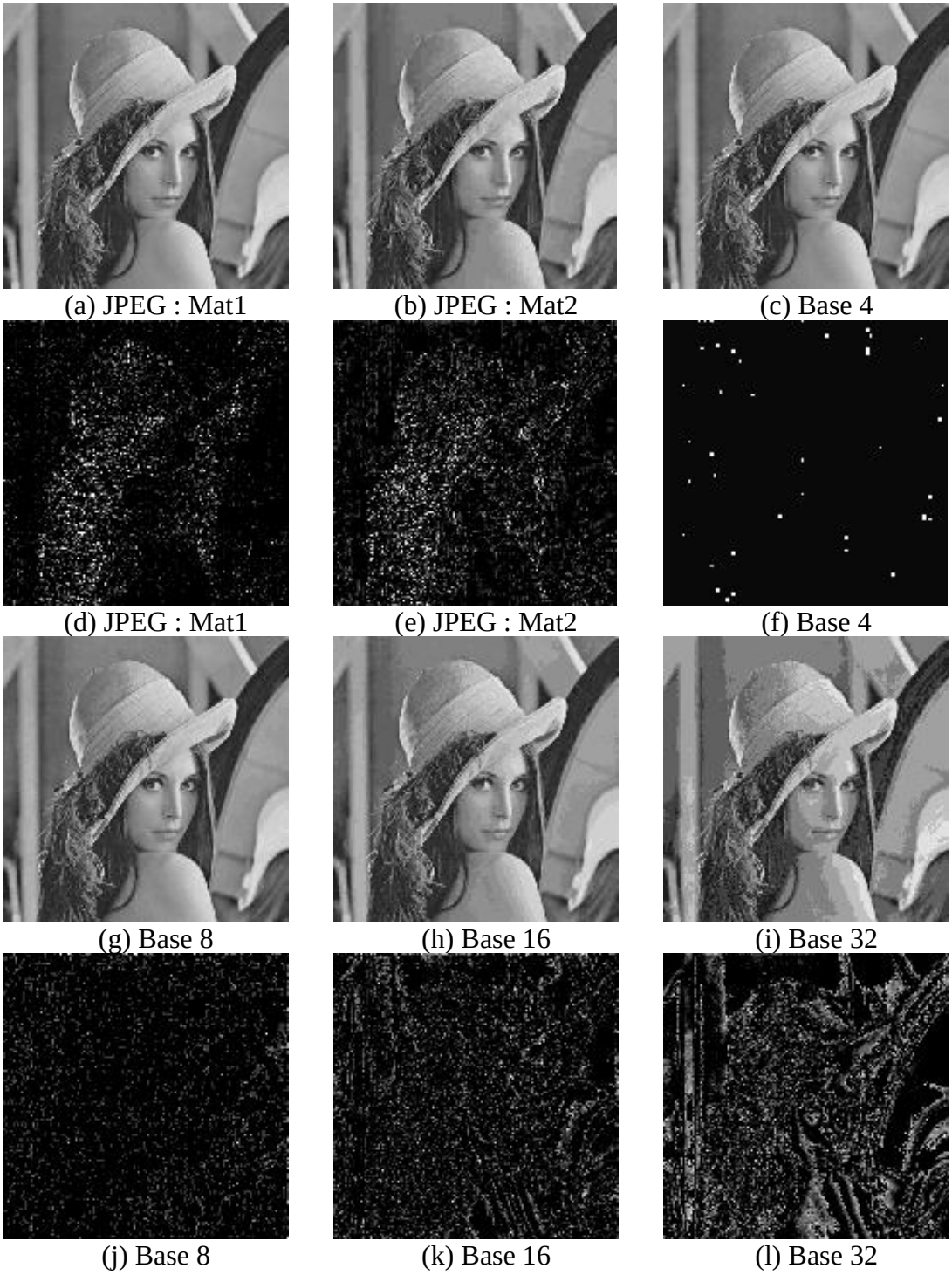
Step 3: Multiply the pixel values by the base to get back the original pixel values. In this case the maximum loss will be equal to base -1.

Step 4: Level shifted by 2P-1 that is, 2P-1 is added to each pixel value, where P is the number of bits used to represent each pixel. Thus 128 is added to each pixel if the image is 8-bit.

5.2 Simulation Results

The programs are implemented using Matlab 7.1. Performance of the proposed scheme is evaluated on a set of test images namely, Lena, Baboon, Pepper, Gold Hill and Airport images seen in figure 3.10 of second chapter. In Table 5.1 and Table 5.2, the performance of the proposed technique is compared with the JPEG compression. Two different quantization matrixes Mat1 & Mat2 as shown in Table 3.1 of second chapter has been considered for JPEG compression.

Table 5.1 lists the PSNR in dB obtained for JPEG technique and proposed techniques for modulo values 4, 8, 16, 32 i.e. 2^n where n= 1, 2, 3… etc. Table 5.2 lists the bpp (bits per pixel) required for representing one pixel in case of the compressed image. The reconstructed image and the error images for different techniques are shown in figures 5.3 – 5.7.



(a) JPEG : Mat1 (b) JPEG : Mat2 (c) Base 4

(d) JPEG : Mat1 (e) JPEG : Mat2 (f) Base 4

(g) Base 8 (h) Base 16 (i) Base 32

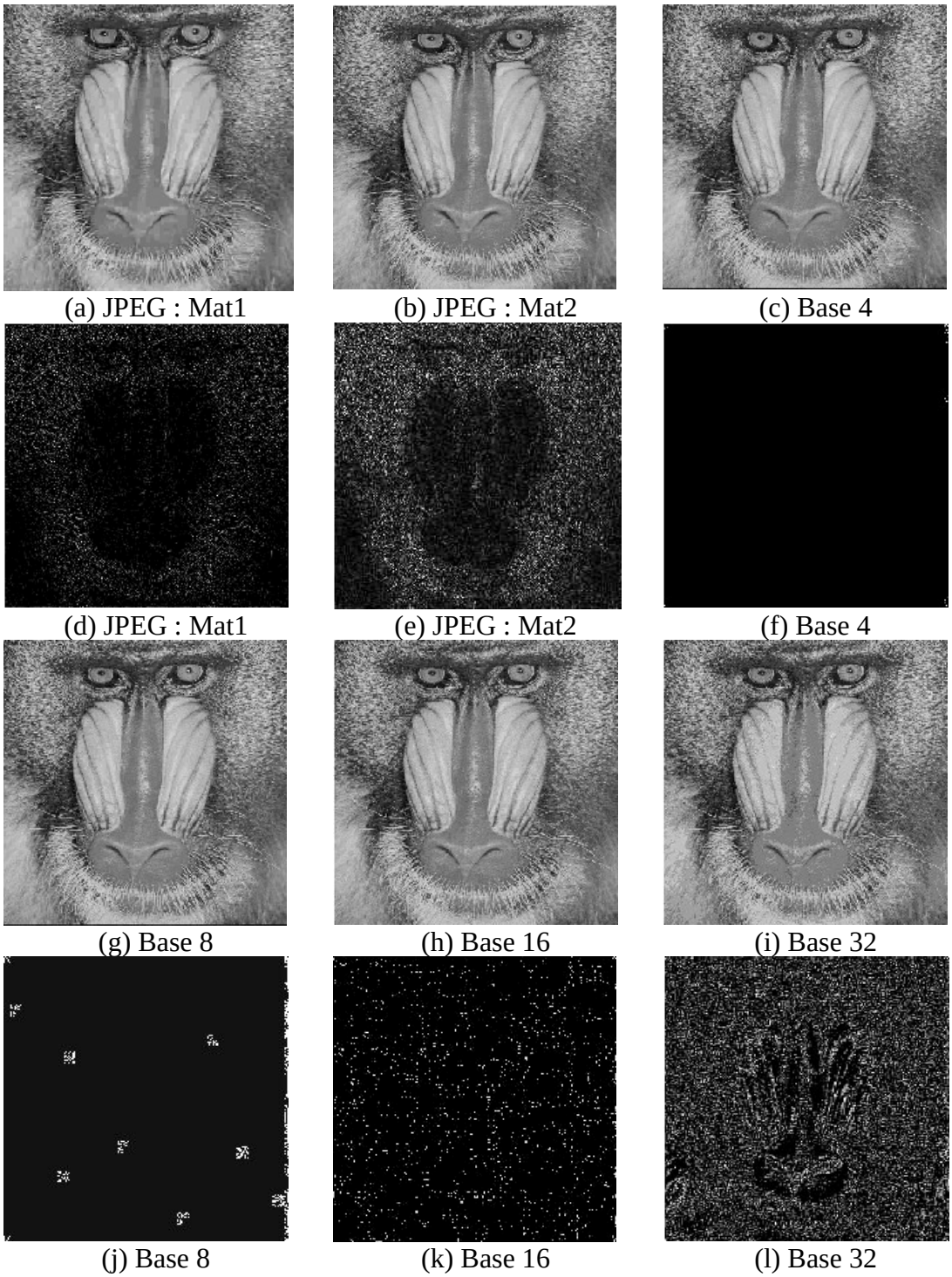
(j) Base 8 (k) Base 16 (l) Base 32

Figure 5.3: (a, b) Decoded Lena images using JPEG quantization matrix Mat1 & Mat2

(d, e) Error images of JPEG quantization matrix Mat1 & Mat2

(c, g, h, i)Decoded images using proposed Tech for modulo 4, 8, 16, 32

(f, j, k, l) Error images using proposed Tech for modulo 4, 8, 16, 32



(a) JPEG : Mat1 (b) JPEG : Mat2 (c) Base 4

(d) JPEG : Mat1 (e) JPEG : Mat2 (f) Base 4

(g) Base 8 (h) Base 16 (i) Base 32

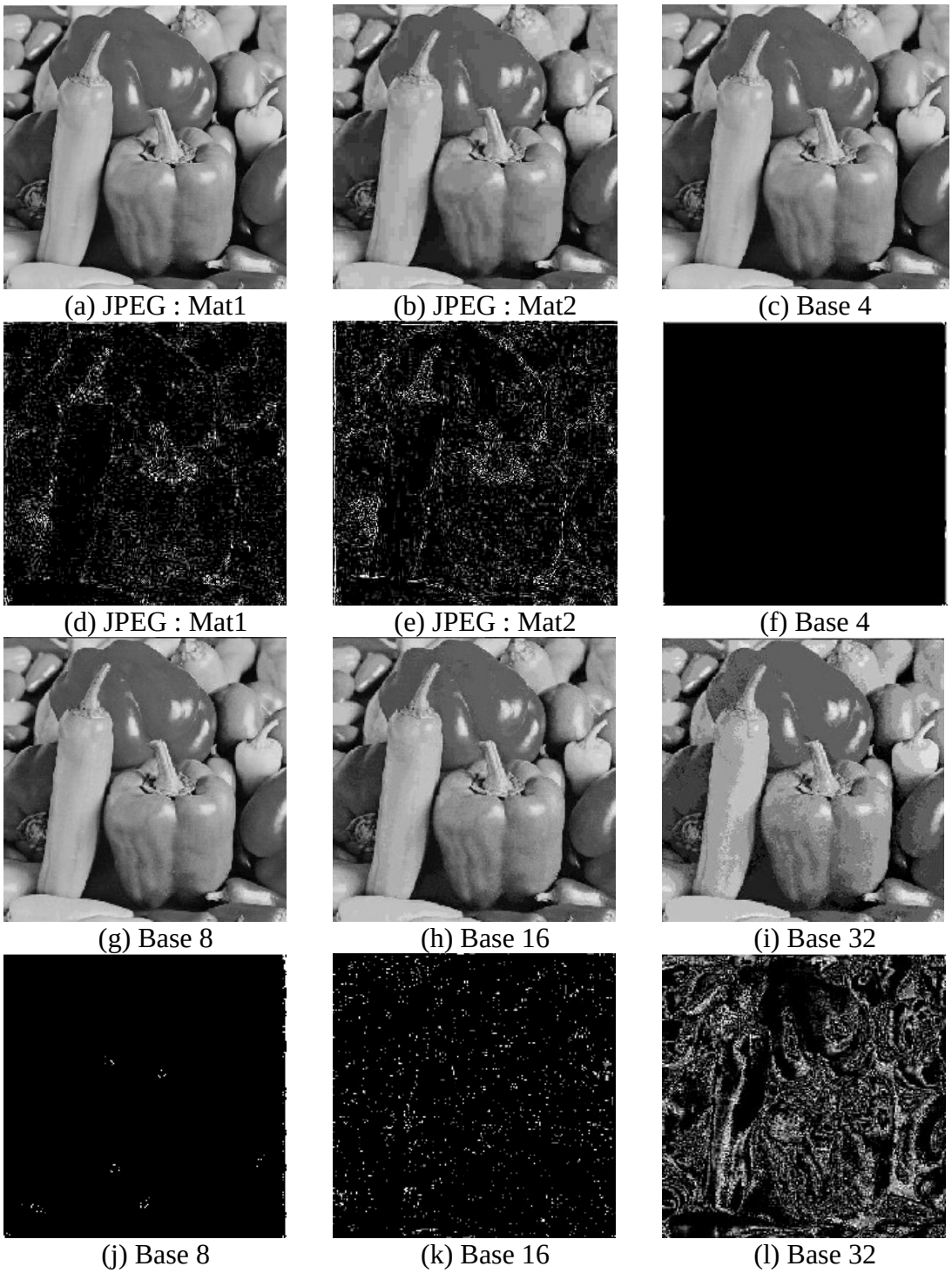
(j) Base 8 (k) Base 16 (l) Base 32

Figure 5.4: (a, b) Decoded Babbon image using JPEG quantization matrix Mat1 & Mat2

(d, e) Error images of JPEG quantization matrix Mat1 & Mat2

(c, g, h, i)Decoded images using proposed Tech for modulo 4, 8, 16, 32

(f, j, k, l) Error images using proposed Tech for modulo 4, 8, 16, 32



(a) JPEG : Mat1 (b) JPEG : Mat2 (c) Base 4

(d) JPEG : Mat1 (e) JPEG : Mat2 (f) Base 4

(g) Base 8 (h) Base 16 (i) Base 32

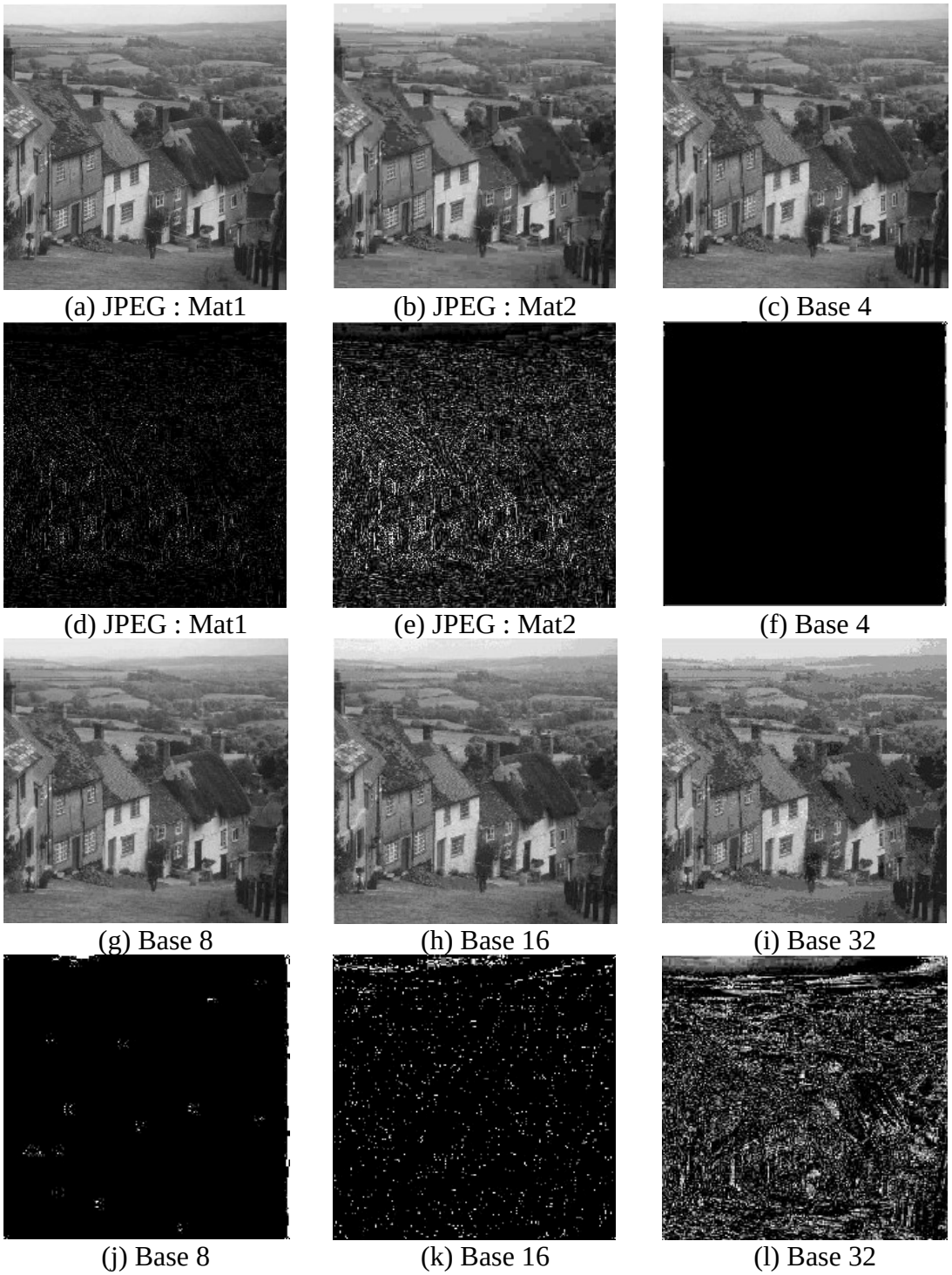
(j) Base 8 (k) Base 16 (l) Base 32

Figure 5.5: (a, b) Decoded Peppers image using JPEG quantization matrix Mat1 & Mat2

(d, e) Error images of JPEG quantization matrix Mat1 & Mat2

(c, g, h, i)Decoded images using proposed Tech for modulo 4, 8, 16, 32

(f, j, k, l) Error images using proposed Tech for modulo 4, 8, 16, 32



(a) JPEG : Mat1 (b) JPEG : Mat2 (c) Base 4

(d) JPEG : Mat1 (e) JPEG : Mat2 (f) Base 4

(g) Base 8 (h) Base 16 (i) Base 32

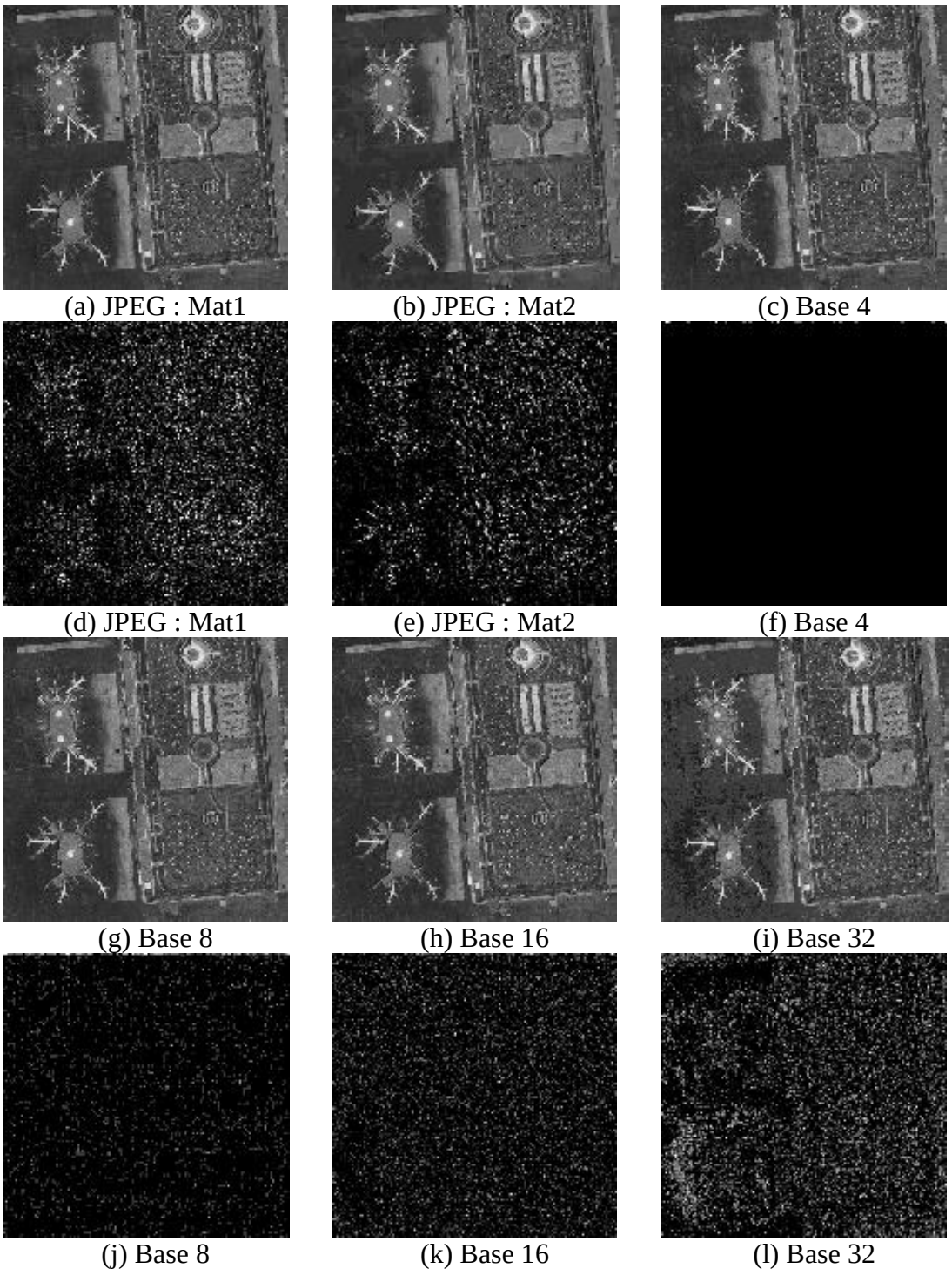
(j) Base 8 (k) Base 16 (l) Base 32

Figure 5.6: (a, b) Decoded Gold Hill image using JPEG quantization matrix Mat1 & Mat2

(d, e) Error images of JPEG quantization matrix Mat1 & Mat2

(c, g, h, i)Decoded images using proposed Tech for modulo 4, 8, 16, 32

(f, j, k, l) Error images using proposed Tech for modulo 4, 8, 16, 32



(a) JPEG : Mat1 (b) JPEG : Mat2 (c) Base 4

(d) JPEG : Mat1 (e) JPEG : Mat2 (f) Base 4

(g) Base 8 (h) Base 16 (i) Base 32

(j) Base 8 (k) Base 16 (l) Base 32

Figure 5.7: (a, b) Decoded Airport image using JPEG quantization matrix Mat1 & Mat2

(d, e) Error images of JPEG quantization matrix Mat1 & Mat2

(c, g, h, i)Decoded images using proposed Tech for modulo 4, 8, 16, 32

(f, j, k, l) Error images using proposed Tech for modulo 4, 8, 16, 32

Table 5.1: Comparative results in PSNR (dB) for JPEG compression and Aritmatic

compression techniques for five different images

		LENA	BABBON	PEPPERS	GOLDHILL	AIRPORT
JPEG	MAT-1	36.80	28.22	34.74	33.58	28.68
	MAT-2	30.62	23.42	30.09	28.65	24.58
Arithmetic Compression	4	46.37	46.36	46.37	46.36	46.24
	8	40.74	40.70	40.71	40.74	40.66
	16	34.93	34.78	34.87	34.75	34.90
	32	28.66	28.80	28.87	28.87	28.79
Arithmetic Compression + Snake scan	4	46.37	46.36	46.37	46.36	46.24
	8	40.74	40.70	40.71	40.74	40.66
	16	34.93	34.78	34.87	34.75	34.90
	32	28.67	28.80	28.86	28.87	28.34

Table 5.2: Comparative results in bpp (bits per pixcel) for JPEG compression and Aritmatic compression techniques for five different images:

		LENA	BABBON	PEPPERS	GOLDHILL	AIRPORT
JPEG	MAT-1	36.80	28.22	34.74	33.58	28.68
	MAT-2	30.62	23.42	30.09	28.65	24.58
Arithmetic Compression	4	6.458	6.285	6.771	6.552	7.004
	8	5.556	5.393	5.856	5.643	6.077
	16	4.707	4.550	4.988	4.779	5.189
	32	3.887	3.768	4.180	3.956	4.344
Arithmetic Compression + Snake scan	4	3.698	4.953	3.978	4.060	4.818
	8	3.097	4.158	3.272	3.358	4.034
	16	2.648	3.451	2.722	2.817	3.335
	32	2.380	2.870	2.393	2.439	2.861

5.3 Summary and Discussion

A new method of lossless image compression technique is proposed to speedup the encoding and decoding procedure to fulfill the present-day requirement of multimedia technology. This is very much useful for those images where the information content is very large i.e. redundant data is very less. From the simulation results it's clear that the edges of the images are lost in case of the JPEG compression, where as in case of the proposed technique the edges are preserved. The coding and decoding procedure is very faster then the existing algorithms as only few addition/ subtraction and sift operation are performed to obtain the compression. The hardware requirement for the implementation of the proposed technique will be less then that of the existing technique.

Chapter

6

LOW POWER
Image Compression Technique

FPGA Implementation Results

Electronics design automation must cope with technological trend in silicon integration. Nowadays, the possibility of integrating millions of transistors onto a single silicon chip is demanding for new CAD tools, to bridge the gap between technology and design. As integration technology enables the development of deep submicron CMOS circuits for digital signal processing (DSP) in an ever and ever increasing variety of applications, designers have to face new problems which require new design methodologies and tools.

The increasing gap between technology capability and designer productivity is demanding for new design methodologies and innovative CAD tools, as in previous stages of the “design crisis”. Following the introduction of electronics design automation tools for physical design (partitioning, placement and routing) and, more recently, for digital circuit synthesis from behavioral description (hardware description languages), now a remarkable effort is being spent, aiming at the automatic synthesis starting from functional specifications rather than from behavioral representation.

The image compression techniques discussed in the previous chapters are implemented in hardware for the testing and verification of low complexity and hence low power. In this chapter a brief overview of VLSI design is given. In the second section a overview of AccelChip software is given, which is used for implementation of all the algorithms. And finally the implementation results are tabulated and compared to draw definite conclusions.

6.1 VLSI Overview and Reconfigurable Computing

The semiconductor industry has evolved the first integrated circuits (ICs) that matured rapidly since then. Early small-scale integration (SSI) ICs contain a few (1 to 10) logic gates NAND gates, NOR gates and few tens of transistors. The era of medium-scale integration (MSI) increased the range of integrated logic available to some counters and some larger functions. The era of large-scale integration (LSI) packed even larger logic functions, such as the first microprocessors, into a single chip. Then the evolution of very large scale integration (VLSI) has developed when millions of transistors can be integrated into a single chip. By using VLSI, the design of 64-bit microprocessors, complete with cache memory and floating point arithmetic

units has become possible. Based on all the new technologies which have been grown from several years, the digital integrated circuit (IC) is one of the most phenomenal growths in terms of circuit complexity, switching speed and the power dissipation. The design of digital system from specification to final product involves a lot of effort. Among the possible design methodologies of full-custom, mask programmable and field programmable logic devices, field programmable gate array (FPGAs) [80] with different architecture and programming capacity are recently used overwhelmingly in different Application Specific Integrated Circuit (ASIC) development. Various types of sophisticated Computed Aided Design (CAD) tools are now available which really made the whole process feasible economically and timely.

6.1.1 Advantages of Using ASIC

The major advantages of using an ASIC are as follows;

(a) **Miniaturization**

The usage of custom ICs will reduce the size of the end product. An ASIC may replace the functions of a number of PCBs in the system, resulting in size reduction.

(b) **Lesser inventory**

The reduced number of components per system reduces the inventory. This in turn reduces the overall cost.

(c) **Reduced maintenance cost**

Lesser components lead to fewer failures and lesser system down time. Maintenance will be easy. All it may need would be the replacement of a single PCB.

(d) **Lower power consumption**

Lesser number of components in a system reduces the power consumption. Most of the ASICs are based on the low power CMOS technology.

(e) **Performance**

More number of functions can be integrated to the ASIC, without increasing the size, cost or power consumption of the product

6.1.2 Major Risks of Using ASIC

(a) **Higher Cost**

ASIC will be always expensive than the standard components. We have to invest more time and money for the design and development phase. The selection of the ASIC technology is very important.

(b) **Time to market**

The product lead time will be more for an ASIC based system. The market research team should define the requirement of the end product well in advance. Last minute changes in the specification will result in delayed market entry and revenue loss. The right product should be introduced in to the market at the right time.

(c) **First time success**

The ASIC design should be properly simulated and thoroughly tested to insure the first time success. Any failure will affect the time to market and resulting huge loss in revenue.

6.2 VLSI Design Methodologies

The basic steps involved in designing an ASIC can well be understood from the figure 6.1 [80]. This is an iterative process of development and testing.

ASIC can broadly be classified into following categories.

(A) **Full Custom ASIC**

Possibly all logic cells and mask layers are customized in a full custom ASIC. So, full custom ICs are more costly. Manufacturing time (not including design time) requires 8 to 10 weeks. Designers go for full custom if there is no suitable existing cell library which is fast enough or the logic cells are small enough or consume too much power.

This ASIC technology is chosen, based on many reasons like

1) Power consumption

2) Functional requirement

3) Speed of the ASIC

4) Non availability of suitable library

5) Product life, etc.

Microprocessor and Microcontroller are examples of full custom IC. Full custom technology is widely used in the mixed analog digital ASICs.

(B) **Semi-Custom ASIC**

In the semi custom ASIC, the designer will be using the pre-characterized and sometimes prefabricated logic cells. This approach reduces the design time and increases the turn around time of the ASIC.

Again the semi-custom ASIC can be subcategorized into,

1) Standard Cell based

2) Gate Array base

(C) **Programmable ASIC**

The programmable ASIC is the latest invention of the IC family. A programmable ASIC can be reprogrammed according to a change in the specification. Just it needs to reconfigure itself by the new configuration file, unlike the case of other ASIC where one has to go through all the steps starting from specification to fabrication. It reduces the development time and cost. The programmable ASIC can be sub grouped into two types according to their architecture and function.

1) Programmable Logic Device (PLD)

2) Field Programmable Gate Array (FPGA)

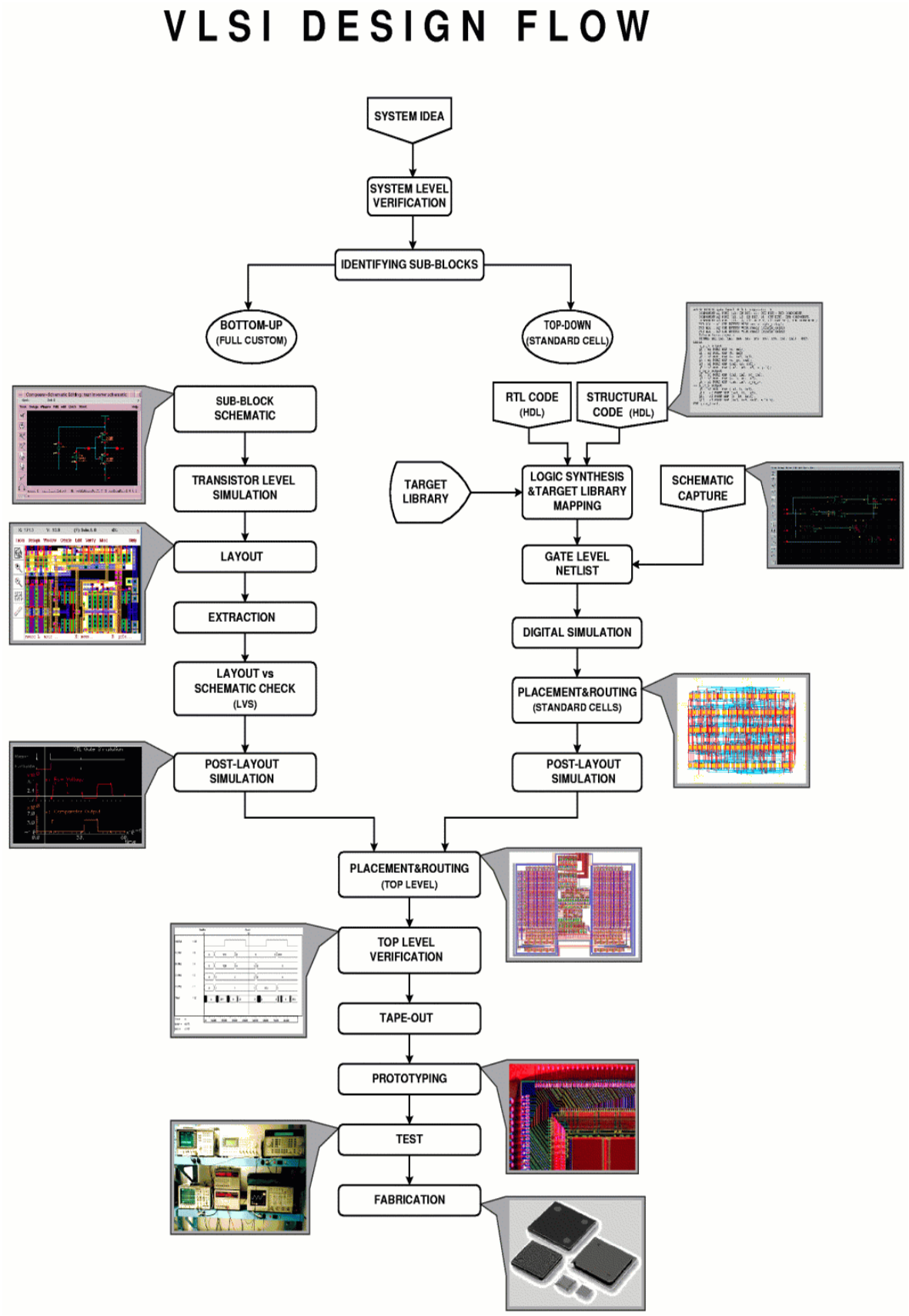


Figure 6.1: Steps involved in VLSI designing

6.3 FPGA Devices

FPGA is a high capacity programmable logic device [82]. An FPGA consists of an array of programmable basic logic cells surrounded by programmable interconnect. It can be configured by end user (field programmable) to have specific circuitry with in it. Any combinational or sequential circuit can be designed using FPGA. The programmable logic array was introduced in the late 1970s and was followed by the introduction of the first FPGA, XC2000 series, by Xilinx in 1985. The advantage of FPGAs is that they combine the performance that can be achieved by ASICs with the flexibility of programmable microprocessors. With these merits, FPGAs destroyed the balance of gate array market and have taken a significant proportion of the standard cell market. Conceptually a programmable FPGA has three key elements as illustrated in figure 6.2.

Programmable logic cells: It provides the functional elements for constructing the user's logic.

Programmable Input/Output (I/O) blocks: It provide the interface between the package pins and the logic cell.

Programmable interconnects: It provides routing paths to connect the inputs and outputs of the logic cell and I/O blocks.

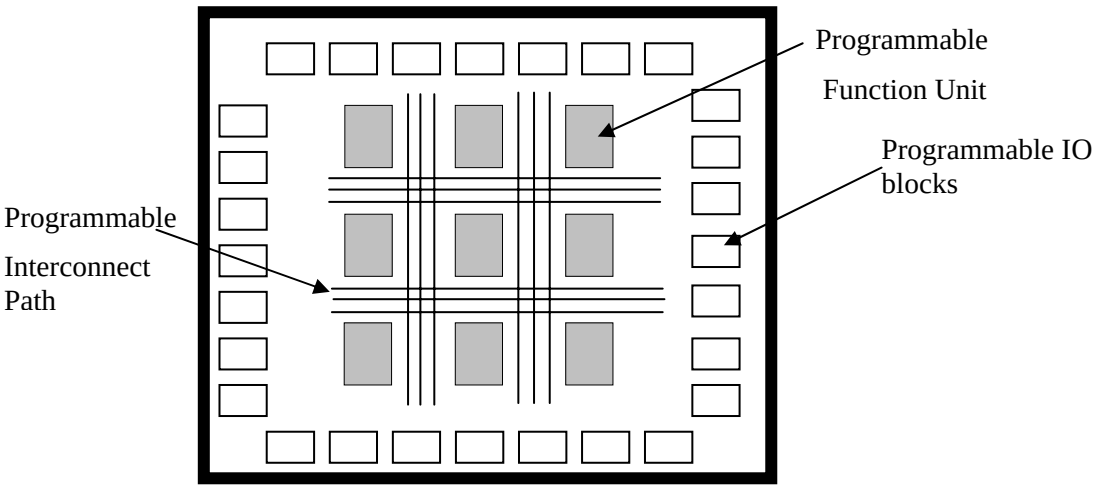


Figure 6.2: Internal Structure of an FPGA

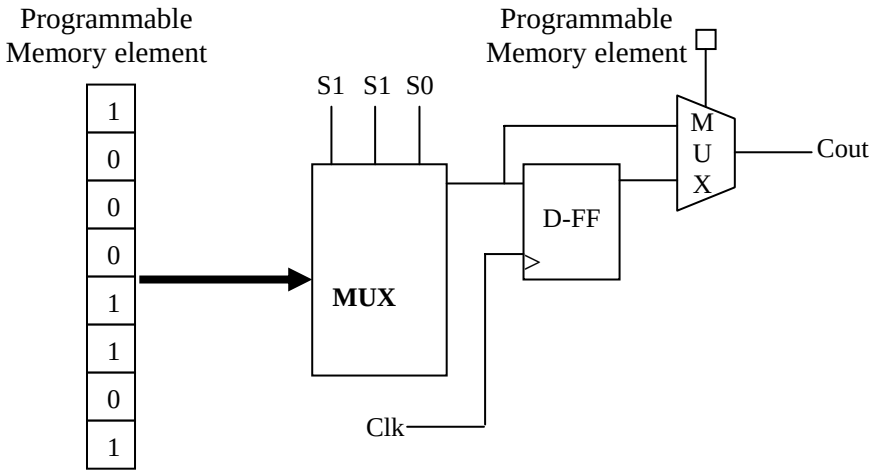


Figure 6.3: Building block in basic cell design

For any programmable device the control information is stored in the memory which can be programmed as shown in figure 6.3. According to the select input from the programmable memory the accurate data are selected by the MUX and thus the switches are connected accordingly.

6.3.1 Selection of FPGA device

The performance and cost of the final FPGA based product, depends on the target FPGA. Therefore before making a prototype, it is very much essential to choose the target FPGA device. Large variations of FPGA devices are available, by different vendors. In this present research work the target FPGA ***Virtex, XCV1000*** is selected and is a product from the vendor Xilinx [83]. The resources offered by this family of FPGA are summarized in Table 6.1 and Table 6.2. The Virtex FPGA family delivers high-performance, high-capacity programmable logic solutions. Dramatic increases in silicon efficiency result from optimizing the new architecture for place-and-route efficiency and exploiting an aggressive 5-layer-metal 0.22 µm CMOS process. These advances make Virtex FPGAs powerful and flexible alternatives to mask-programmed gate arrays.

Table – 6.1 : FPGA Resources

Device	System Gates	CLB Array	Logic Cells	Available I/O	Block RAM
XCV1000	1,124,022	64 X 96	27,648	512	131,072

Table – 6.2 : FPGA Resources

Device	Slices	Flip Flops	4 Input LUTs	Available I/O	Block RAM
XCV1000	12,288	24,576	24,576	512	131,072

Virtex devices feature a flexible, regular architecture as shown in figure 6.4, comprises an array of configurable logic blocks (CLBs) surrounded by programmable input/output blocks (IOBs), all interconnected by a rich hierarchy of fast, versatile routing resources. The abundance of routing resources permits the Virtex family to accommodate even the largest and most complex designs.

Virtex FPGAs are SRAM-based, and are customized by loading configuration data into internal memory cells. In some modes, the FPGA reads its own configuration data from an external PROM (master serial mode). Otherwise, the configuration data is written into the FPGA (Select- MAP™, slave serial, and JTAG modes). The standard Xilinx Foundation™ and Alliance Series™ Development systems deliver complete design support for Virtex, covering every aspect from behavioral and schematic entry, through simulation, automatic design translation and implementation, to the creation, downloading, and read-back of a configuration bit stream. Virtex devices provide better performance than previous generations of FPGA. Designs can achieve synchronous system clock rates up to 200 MHz including I/O. Virtex inputs and outputs comply fully with PCI specifications, and interfaces can be implemented that operate at 33 MHz or 66 MHz.

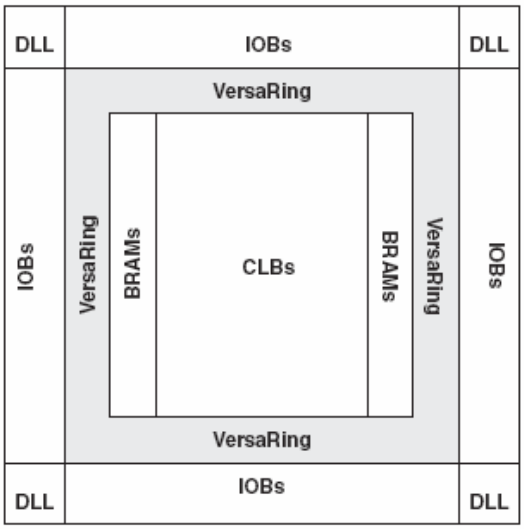


Figure 6.4: Virtex Architecture

6.4 Introduction to AccelChip

AccelChip DSP Synthesis, is a DSP (Digital Signal Processing) synthesis tool that allows to transform a MATLAB floating-point design into a synthesized hardware module that can be implemented in silicon (FPGA or ASIC). AccelChip DSP Synthesis features an easy-to-use Graphical User Interface, called AccelView, that controls an integrated environment with other design tools such as MATLAB, HDL simulators, logic synthesizers, and some vendor tools.

AccelChip® DSP Synthesis provides the following capability:

• Reads and analyzes a MATLAB floating-point design

• Automatically creates an equivalent MATLAB fixed-point design

• Invokes a MATLAB simulation to verify the fixed-point design

• Provides you with the power to quickly explore design tradeoffs of algorithms that are optimized for target FPGA and ASIC architectures

• Creates a synthesizable RTL HDL model and a Testbench to ensure bit-true, cycle-accurate design verification

• Provides scripts that invoke and control downstream tools such as HDL simulators, RTL logic synthesizers, and vendor implementation tools.

6.4.1 AccelChip DSP Synthesis Flow

A MATLAB floating-point model is synthesized into a hardware module using the following basic steps in the AccelChip DSP Synthesis Flow shown in figure 6.5.

1. Examine the Coding Style of the Floating-Point Model. We should first verify that the MATLAB design conforms to minimum AccelChip style guidelines that are explained in the manual provided.

2. Create an AccelChip Project. We invoke AccelChip DSP Synthesis, then click Project and specify the name of a new Project file. The Project file is placed in the Project Directory where all future AccelChip-generated files are saved.

3. Verify the Floating-Point Model. We should have verification constructs in the MATLAB script file to apply stimulus and plot results. This output plot is the "golden" reference for comparing future results. If we have verified the floating-point model outside of AccelChip DSP Synthesis, we may skip this step.

4. Analyze the Floating-Point Model. This step creates an in-memory model of the design. In a later pass through this flow, one can add design directives that guide AccelChip toward finding the best hardware architecture for the design.

5. Generate a Fixed-Point Model. This step generates a fixed-point model of the design, and then places the design files in a newly generated project sub-folder named MATLAB.

6. Verify the Fixed-Point Model. When we select Verify Fixed Point, AccelChip automatically runs a MATLAB fixed-point simulation. Then after the simulation visually compare the Fixed-Point Plot with the Floating-Point Plot to verify a match.

7. Generate an RTL Model. This step generates an RTL Model from the in- memory design data base. The RTL model can be generated in VHDL or Verilog format.

8. Verify the RTL Model with the HDL Simulator. When Verify RTL option is selected, HDL simulator will test the generated RTL code by using the generated Testbench. Finally PASSED or FAILED will be indicated in a simulation report.

9. Synthesize the RTL Design into a Gate-Level Netlist. This step invokes a pre-specified RTL synthesis tool for the design. The generated gate-level netlist is ready for place and route using the Vendor's implementation tools.

10. Implement the Gate-Level Netlist. When Implement is selected it invoke the vendor's implementation tools to place and route the design. The generated files of interest are a gate-level HDL simulation file and a configuration file containing the bitstream for configuring the FPGA hardware.

11. Verify the Gate Level Design. This step uses the AccelChip Testbench to run a bit-true simulation check on the gate-level HDL simulation model. A PASSED indication means that the implemented design is bit-true with the original fixed-point MATLAB design.

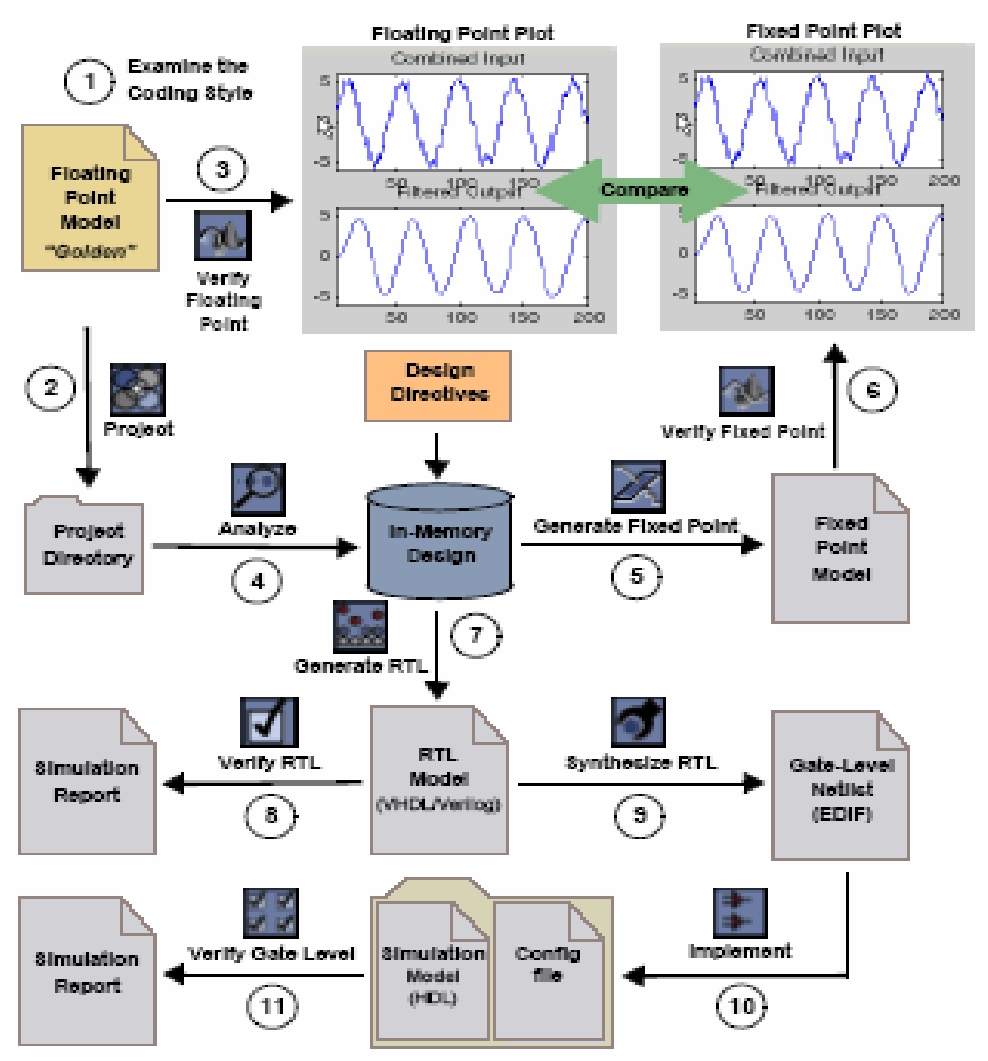


Figure 5.5: AccelChip DSP Synthesis Flow

6.5 Simulation Results

All the compression techniques discussed in the previous chapters are synthesized by using the AccelChip DSP Synthesis tools. FPGA ***Virtex, XCV1000*** device is used for synthesize all the algorithms to get an approximate hardware requirements for their calculation. The comparative study is performed in terms of number of slices, number of Flip Flops, number of LUTs used, Table 6.3 presents lists for calculation of DCT, IDCT, JPEG quantisation, energy quantisation (EQ), modified energy quantisation (MEQ), rule based energy quantisaton (RBEQ), DHT and IDHT, Arithmetic compression technique respectively. Table 6.4 presents a comparative study for different image compression techniques in terms of size and

power consumption in mW calculated using the Xilinx websites power calculation tool [81]. For calculation of the power consumption we have assumed that all the devices are operated at 10MHz frequency.

Table 6.3: Synthesize reports of different techniques

	Number of Slices Used	Number of Flip Flops used	Number of 4 Input LUTs used
DCT Calculation	4836	6561	8538
IDCT Calculation	6295	8346	11188
JPEG Quantisation	15338	22827	27529
JPEG Dequantisation	14146	11250	25811
Energy Quantisation (EQ)	17927	25816	33099
Modified Energy Quantisation (MEQ)	15602	23163	27951
Rule Based Energy Quantisation (RBEQ)	11867	14741	9990
DHT and IDHT	5939	4420	10844
Arithmetic difference Encoding	950	149	1381
Arithmetic difference Decoding	22	37	35
Arithmetic Encoding with snake scan	1897	236	2766
Arithmetic Encoding with snake scan	52	64	94

Table 6.4: Comparison between different image compression technique in terms of size and power consumption

Compression Techniques	Number of Slices Used	Number of Flip Flops used	Number of 4 Input LUTs used	Power consumed in mW
DCT + IDCT + JPEG Quantisation	40615	48984	73066	2011
DCT + IDCT + EQ	29058	40723	52825	1637
DCT + IDCT + MEQ	26733	38070	47677	1501
DCT + IDCT + RBEQ	22998	29648	29716	1056
DHT + IDHT + JPEG Quantisation	35423	38497	64184	1811
DHT + IDHT + EQ	23866	30236	43943	1301
DHT + IDHT + MEQ	21541	27583	38795	1165
DHT + IDHT + RBEQ	17806	19161	20834	720
Arithmetic difference coding and decoding	972	186	1416	11
Arithmetic Snake scan coding and decoding	1949	300	2860	22

6.6 Summary and Discussion

It is clear form the implementation that the hard ware requirement gradually decreases form vector quantisation to rule based energy quantisation. DHT based JPEG with rule based energy quantisation is an ideal solution for lossy image compression technique as the hardware requirement is less and power consumption is very low compare to other techniques. If we want to go for good quality, low compression and high speed then the arithmetic compression with snake scanning is the best choice, which also consumes very less power compare to the rest compression techniques. The schematic generated after implementation of the last technique is shown in Appendix-1.

Chapter

7

LOW POWER
Image Compression Techniques

Conclusion

The work in this thesis, primarily focuses on image compression with less computation and low power. Novel schemes for quantisation of transformed coefficients have also been devised. The work reported in this thesis is summarized in this chapter. Section 7.1 lists the pros and cons of the work. Section 7.2 provides some scope for further development.

7.1 Achievements and Limitations of the work

A brief study on low power VLSI design is discussed in Chapter II. The main sources of power losses in the transistors are explained and the necessary action has to be taken to reduce this losses were specified. All the low power techniques such as system level, algorithm level, architecture level, logic level and circuit level are explained elaborately also explains how these techniques are used for power reduction. Lastly it describes the basic steps to be followed for low power design.

It is observed from the investigations made in Chapter III that the proposed three quantisation techniques: energy quantisation, modified energy quantisation and rule based energy quantisation are better candidate for signal and image compression compared to the standard scalar and vector quantisation techniques. Further the rule based energy quantisation technique makes the quantisation process intelligent by setting different threshold value for different image blocks and yields superior compression performance as compared to other techniques. The computational complexity of the proposed techniques is also less compare to the standard JPEG technique.

In Chapter IV we have discussed an alternative scheme of JPEG by substituting the DHT in place of the conventional DCT. Such a substitution has resulted improved compression and reconstruction performance. It is shown that qualitatively the new JPEG offers superior performance. Further there is a saving in hardware if DHT is employed in place of the DCT. We have also used the quantisation techniques proposed in chapter-III with this DHT based JPEG compression technique. Which further reduces the computational complexity and also facilitates easier implementation.

As digital storage is becoming so cheap and so wide spread and the available transmission channel bandwidth is increasing due to the deployment of cable, fiber optics and ADSL modems, why is there a need to provide more powerful compression scheme? The answer is, with no doubt mobile video transmission

channels and Internet streaming, which mainly requires high quality and high speed. Chapter V we provide a novel lossless image compression technique. The main aim behind this technique is to reduce the complexity and secondly to make the compression technique fast with a reasonable image compression. With this new technique the compression and decompression technique becomes very faster and also hardware requirement is very less compare to other techniques re out signal compression the transmission time as well as the storage requirement will be large but inclusion of compression and reconstruction scheme reduces the storage requirement and data transmission time.

VLSI designing procedure is discussed briefly in chapter-VI. All the advantages and disadvantage of using FPGA over ASIC design is discussed. How to select the FPGA device for different requirements were discussed. AccelChip DSP Synthesis, is a DSP (Digital Signal Processing) synthesis tool that allows to transform a MATLAB floating-point design into a synthesized hardware module that can be implemented in silicon (FPGA or ASIC), all the step involved for the implementation of different algorithms were discussed. Finally some conclusions were made on the basis of the implementation results. It is clear form the implementation that the hard ware requirement gradually decreases form vector quantisation to rule based energy quantisation. DHT based JPEG with rule based energy quantisation is an ideal solution for lossy image compression technique as the hardware requirement is less and power consumption is very low compare to other techniques. If we want to go for good quality, low compression and high speed then the arithmetic compression with snake scanning is the best choice, which also consumes very less power compare to the rest compression techniques.

In general, it is concluded that the investigation made in the present thesis pertains to the development of some alternative compression-reconstruction schemes, which offer superior performance compared to the conventional works. The proposed quantisation techniques for image compression proposed in the thesis has aimed to enhance the CR required for some multimedia applications while reducing the computational complexity for coding and decoding. Simulation study has been carried out to support this idea. Further all the compression techniques proposed were implanted to get the size required in terms of number of slices, LUTs, Flip-Flops used which is used to calculate the power consumption by different techniques. Which

exploits the use of efficient image compression and reconstruction schemes with low power consumption.

7.2 Further Development

To conclude this thesis, following are some points that may lead to some better and interesting results.

The different image compression techniques developed in the thesis can suitably be applied for 3D video signals. Some hybrid image compression scheme or algorithms can be developed by using some soft computing techniques like multilayer Artificial Neural Network (ANN), Radial Basis Function (RBF), Multi-Layer Perceptron (MLP) with the proposed techniques. This investigation may lead to intelligent and adaptive efficient compression scheme. The proposed quantisation techniques can be used in other image compression techniques such as Discrete Wavelet Transform (DWT) and Slantlet transform to threshold the transformed coefficients.

Bibliography

[1] David Salomon, Data Compression, The Complete Reference, 2nd Edition Springer-Verlag 1998.

[2] Digital Compression and coding of Continuous-tone still images, part 1, requirements and Guidelines. ISO/IEC JTC1 Draft International Standard 10918-1, Nov. 1991.

[3] Digital Compression and coding of Continuous-tone still images, part 2, compliance Testing. ISO/IEC JTC1 Committee Draft 10918-1, Dec. 1991.

[4] E. Fossum, "CMOS Image Sensors: Electronic Camera on a Chip", IEEE Int. Electron. Devices Meeting Tech. Digest, Dec. 1995, pp. 1-9.

[5] Eyadat, M.; Muhi, I.; “Compression standards roles in image processing: case study”, International Conference on Information Technology: Coding and Computing, 2005. ITCC 2005. Volume 2, 4-6 April 2005 Page(s):135 - 140 Vol. 2.

[6] Frank Vahid, Tony Givargis, “Embedded System Design: A Unified Hardware/Software Introduction”, John Wiley & Sons, 2002.

[7] K. A. Parulski, "Color Filters and Processing Alternatives for One-chip Cameras," IEEE Trans. Electron Devices, Vol. ED-32, NO. 8, pp. 1381-1389, Aug. 1985.

[8] N. Kihara, et al., "The Electronic Still Camera: A New Concept in Photography," IEEE Trans. Cons. Electron., Vol. CE-28, NO. 3, pp. 325-335, Aug. 1982.

[9] R. C. Gonzalez and R. E. Woods, “Digital Image Processing”, Reading. MA: Addison Wesley, 2004.

[10] Ranjan Bose, Information theory coding and Cryptography, Tata McGraw-Hill 2003.

[11] S. Narayanaswamy, S. Seshan, E. Amir, et al., “A low-power, lightweight unit to provide ubiquitous information access application and network support for InfoPad”, IEEE Personal Communications, pp. 4–17, Apr. 1996.

[12] A. Chadrakasan and R. W. Brodersen, Low Power Digital CMOS Design, Kluwer, 1995.

[13] A. Chadrakasan, M. P. Potkonjak, R. Mehra, J. Rabey, and R. W. Brodersen, “Optimizing power using transformations,” IEEE Trans. on Computer-Aided Design, Vol. 14, No. 1, pp. 12–31, Jan., 1995.

[14] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, “Low-power CMOS digital design”, IEEE Journal of Solid-State Circuits, vol. 27, no. 4, pp. 473–483, Apr. 1992.

[15] Bright, M.S.; Arslan, T.; "Synthesis of low-power DSP systems using a genetic algorithm", IEEE Transactions on Evolutionary Computation, Volume 5, Issue 1, Feb 2001 Page(s):27 – 40.

[16] Chunyan Wang; "Low-power VLSI design and implementation, from another point of view", Proceedings of 2005 IEEE International Workshop on VLSI Design and Video Technology, 2005. 28-30 May 2005 Page(s): xix – xix.

[17] Elassal, M.; Baker, A.; Bayoumi, M.; "A low power VLSI design paradigm for iterative decoders", IEEE Workshop on Signal Processing Systems Design and Implementation, 2005. 2-4 Nov. 2005 Page(s):272 – 277.

[18] G. K Yeap, "Practical Low Power Digital VLSI Design", Kluwer Academic Publishers, Norwell, Mass., 1998.

[19] H. J. M. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," IEEE Journal of Solid-State Circuit, Vol. 19, pp. 468–473, Aug., 1984.

[20] L. Nielsen, C. Nielsen, J. Spars, and K. van Berkel, "Lowpower operation using self-timed circuits and adaptive scaling of the supply voltage," IEEE Trans. on VLSI Systems, Vol. 2, No. 4, pp. 391–397, Dec., 1994.

[21] Rabaey JM and Pedram M, "Low Power Design Methodologies", Kluwer Publishers, 1996.

[22] Sakurai, T.; "Low-power and high-speed VLSI design with low supply voltage through cooperation between levels", International Symposium on Quality Electronic Design, 2002. Proceedings. 18-21 March 2002 Page(s):445 – 450.

[23] V. Tiwari, S. Malik, and P. Ashar, "Compilation techniques for low energy: an overview," In Proc. of 1994 IEEE Symp. On Low Power Electronics, San Diego, California, USA, Oct., 1994, pp. 38–39.

[24] Agostini, L.V.; Silva, I.S.; Bampi, S.; "Pipelined entropy coders for JPEG compression", Proceedings. 15th Symposium on Integrated Circuits and Systems Design, 2002. 9-14 Sept. 2002 Page(s):203 – 208.

[25] Agostini, L.V.; Silva, I.S.; Bampi, S.; "Pipelined fast 2D DCT architecture for JPEG image compression", 14th Symposium onIntegrated Circuits and Systems Design, 2001, 10-15 Sept. 2001 Page(s):226 – 231

[26] Arcangelo, B.; Massimo, M.; "JPEG compression factor control: a new algorithm", International Conference on Consumer Electronics, 2001. ICCE. 19-21 June 2001 Page(s):206 – 207

[27] Battiato, S.; Mancuso, M.; Bosco, A.; Guarnera, M.; "Psychovisual and statistical optimization of quantization tables for DCT compression engines", Proceedings. 11th International Conference on Image Analysis and Processing, 2001. 26-28 Sept. 2001 Page(s):602 – 606.

[28] Bruna, A.; Capra, A.; Battiato, S.; La Rosa, S.; “Advanced DCT rate control by single step analysis”, International Conference on Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. 8-12 Jan. 2005 Page(s):453 – 454.

[29] Campbell, D.; Maeder, A.; Tapia-Vergara, F., “Mammogram JPEG quantisation matrix optimisation for PACS”, The Seventh Australian and New Zealand 2001 Intelligent Information Systems Conference, 18-21 Nov. 2001 Page(s):1 – 6.

[30] Chun-Hsien Chou; Kuo-Cheng Liu, “Color Image Compression using Adaptive Color Quantization”, Image Processing, 2004. ICIP '04. 2004 International Conference on Vol. 4, 24-27, pp. 2331 – 2334, Oct. 2004

[31] D.L. Tull, Robert J. Safranek, “Variable dimension quantization in the transform domain”, ICIP 1995, pp. 302-305, 1995

[32] G. K. Wallace, “The JPEG Still Picture Compression Standard”, IEEE Trans. On Consumer Electronics, vol.38, No.1, pp. xviii – xxxiv, Feb 1992.

[33] Golner, M.A.; Mikhael, W.B.; Krishnan, V.; Ramaswamy, A, “Region Based Variable Quantization for JPEG Image Compression”, Circuits and Systems, 2000. Proceedings of the 43rd IEEE Midwest Symposium on Vol. 2, pp 604 – 607, Aug. 2000.

[34] H. F. Silverman, “An introduction to programming the Winograd Fourier transform algorithm (WFTA)”, IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-25, pp. 152-165, Apr. 1977.

[35] John G. Proakis and D. G. Manolakis, “Digital Signal Processing”, Prentice Hall, 3rd Edition, May’ 1994.

[36] L. Karray, P. Duhatnel, and O. Rioul, " Image coding with an if norm and confidence interval criteria", IEEE Trans. Imaging Processing, Vol.7, no. 5, pp.621-631, Sep 1994.

[37] M. Goldberg, P. R. Boucher, and S. Shlien, “image Compression Using Adaptive Vector Quantization”, IEEE Trans. Commun., Vol. COM-34, No. 2, Feb. 1986.

[38] Melnikov, G.; Katsaggelos, A.K.; “A jointly optimal fractal/DCT compression scheme”, IEEE Transactions on Multimedia, Volume 4, Issue 4, Dec. 2002 Page(s):413 – 422.

[39] N. Ahmed, T. Natrajan, and K. R. Rao, “Discrete Cosine Transform”, IEEE Transactions on Computers, vol. 23, pp. 90-93, July 1989.

[40] Neelamani, R.; deQueiroz, R.; Fan, Z.; Dash, S.; Baraniuk, R.G.; “JPEG Compression History Estimation for Color Images”, International Conference on Image Processing, 2003. ICIP 2003. Volume 3, 14-17 Sept. 2003 Page(s):III - 245-8 vol.2

[41] R. L. de Queiroz, P. Fleckenstein, “Very Fast JPEG Compression Using Hierarchical Vector Quantization”, IEEE Signal Processing Letters, Vol. 7, No. 5, pp. 97-99, May 2000

[42] R. M. Gray, “Vector Quantization”, IEEE ASSP Mag., PP 4-29, Apr. 1984.

[43] R. M. Gray, D. L. Neuhoff, “Quantization”, IEEE Trans. Inform. Theory, Vol. 44, No. 6, 1998.

[44] S. Martucea, “Symmetric convolution and the discrete sine and cosine transform”, IEEE Transaction on Signal Processing, vol. 42, pp. 1038-1051, May’ 1994.

[45] S. P. Lioyd, “Least-Squares Quantization in PCM”, IEEE Trans. Inform. Theory, Vol. IT-25, pp. 373-380, July 1979.

[46] S. Winograd, “On computing the discrete Fourier transform”, Math. Comput., vol. 32, pp. 175-199, Jan. 1978.

[47] Samadani, R.; Sundararajan, A.; Said, A.; “Deringing and deblocking DCT compression artifacts with efficient shifted transforms”, 2004 International Conference on Image Processing, 2004. ICIP '04. Volume 3, 24-27 Oct. 2004 Page(s):1799 - 1802 Vol. 3.

[48] Shneier and M. Abdel-Mottaleb., “Exploiting the JPEG compress scheme for image retrieval.” IEEE Trans. Pattern Anal. Machine Intel, vol. 18, pp. 849-853, Aug. 1996.

[49] Soo-Chang Pei; Jian-Jiun Ding; “Generalized eigenvectors and fractionalization of offset DFTs and DCTs”, IEEE Transactions on Signal Processing, Volume 52, Issue 7, July 2004 Page(s):2032 – 2046.

[50] W. B. Pennebaker and J. L. Mitchell., JPEG Still Image Data Compression Standard, Van Nostrand Reinhold, New York, 1993.

[51] Wu, Y.-G.; “GA-based DCT quantisation table design procedure for medical images”, IEE Proceedings-Vision, Image and Signal Processing, Volume 151, Issue 5, 30 Oct. 2004 Page(s):353 – 359.

[52] Y. Linde, A. Buzo, and R. M. Gray, “An algorithm for vector quantizer design”, IEEE Trans. Commun., Vol. COM-28, pp. 84-95, Jan. 1980.

[53] Ying Luo; Ward, R.K.; “Removing the blocking artifacts of block-based DCT compressed images”, IEEE Transactions on Image Processing, Volume 12, Issue 7, July 2003 Page(s):838 – 842.

[54] A. Murat Tekalp, “Digital Video Processing”, Prentice Hall Publication, NJ, 1995.

[55] Jiebo Luo; Qing Yu; Miller, M.E.; “A triage method of determining the extent of JPEG compression artifacts”, Proceedings. 2002 International Conference on Image Processing. 2002. Volume 1, 22-25 Sept. 2002 Page(s):I-473 - I-476 vol.1.

[56] Meher, P.K.; Srikanthan, T.; Gupta, J.; Agarwal, H.K.; "Near lossless image compression using lossless Hartley like transform", Fourth Pacific Rim Conference on Multimedia, Information, Communications and Signal Processing, 2003 and the Volume 1, 15-18 Dec. 2003 Page(s):213 - 217 Vol.1.

[57] O, Buneman, et al: "Multidimensional Hartley transform", Proc. Of IEEE, vol. 75, no. 2, pp. 267, Feb. 1987

[58] P. K. Meher, J. K. Satapathy, and G. Panda, "Fast computation of multidimensional discrete Hartley Transform", Electronics letter, vol. 28, no. 12, pp. 1077-1078, June 1992.

[59] R. N. Bracewell, et al: "Discrete Hartley Transform", J. Opt. Soc. Amer., vol. 73, no. 12, pp. 1832-1835, Dec. 1983.

[60] R. N. Bracewell, O. Buneman, et al: "Fast two-dimensional Hartley transform", Proc. IEEE, vol. 74, no. 9, pp. 1282-1283, Sept. 1986

[61] R. V. L. Hartley, et al: "A more symmetrical Fourier analysis applied to transmission problem"", Proc. IRE, vol. 30, pp. 144-150, Mar. 1942.

[62] Avcibas, I.; Memon, N.; Sankur, B.; Sayood, K.; "A successively refinable lossless image-coding algorithm", Communications, IEEE Transactions on Volume 53, Issue 3, March 2005 Page(s):445 – 452

[63] Boulgouris, N.V.; Tzovaras, D.; Strintzis, M.G.; "Lossless image compression based on optimal prediction, adaptive lifting, and conditional arithmetic coding", IEEE Transactions on Image Processing, Volume 10, Issue 1, Jan. 2001 Page(s):1 – 14.

[64] Chin-Chen Chang; Chih-Ping Hsieh; Ju-Yuan Hsiao; "A new approach to lossless image compression", International Conference on Communication Technology Proceedings, 2003. ICCT 2003. Volume 2, 9-11 April 2003 Page(s):1734 - 1738 vol.2.

[65] Jayant, N-Johnston, J. Safranek, "Signal Compression Based in Models of Human Perception", Proc. of IEEE 81(1993), 1385-1422.

[66] K. Chen and T.V. Ramabadran, "Near-lossless compression of medical images through entropy-coded DPCM", IEEE Trans. Med. Imaging, Vol.13, no. 3, pp.538-548, Sep 1994.

[67] Krivoulets, A.; "A method for progressive near-lossless image compression", 2003 International Conference on Image Processing, 2003. ICIP 2003. Proceedings. Volume 2, 14-17 Sept. 2003 Page(s):II - 185-8 vol.3.

[68] Krivoulets, A.; "Progressive near-lossless coding of medical images", Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis, 2003. ISPA 2003. Volume 1, 18-20 Sept. 2003 Page(s):202 - 207 Vol.1.

[69] M. Das, D.L. Neuhoff and C.L. Lin, "Near-lossless compression of medical images", Proc. International Conference on Acoustics, Speech, and Signal Processing, 1995. ICASSP-95, Vol.4, pp. 2347 -2350, 1995.

[70] Mielikainen, J.; Toivanen, P.; "Clustered DPCM for the lossless compression of hyperspectral images", IEEE Transactions on Geoscience and Remote Sensing, Volume 41, Issue 12, Part 2, Dec. 2003 Page(s):2943 – 2946.

[71] Rane, S.D.; Sapiro, G.; "Evaluation of JPEG-LS, the new lossless and controlled-lossy still image compression standard, for compression of high-resolution elevation data", IEEE Transactions on Geoscience and Remote Sensing, Volume 39, Issue 10, Oct. 2001 Page(s):2298 – 2306.

[72] Ratakonda, K.; Ahuja, N.; "Lossless image compression with multiscale segmentation", IEEE Transactions on Image Processing, Volume 11, Issue 11, Nov. 2002 Page(s):1228 – 1237.

[73] Redondo, R.; Cristobal, G.; "Lossless chain coder for gray edge images", Proceedings. 2003 International Conference on Image Processing, 2003. ICIP 2003. Volume 2, 14-17 Sept. 2003 Page(s): II - 201-4 vol.3.

[74] S. C. Huang, L. G. Chen, "LOG-EXP Compression System Design and Implementation", IEEE International Sym. On Consumer Electronics, 1998.

[75] S. C. Huang, L. G. Chen, "LOG-EXP Still Image Compression Chip Design", IEEE Trans. On Consumer Electronics, vol.45, No.3, pp. 812 – 818, Aug 1999.

[76] S. C. Huang, L. G. Chen, H. C. Chan,"A Novel Image Compression Algorithm by Using LOG-EXP Transform", IEEE Sym. On Circuits and System, 1999.

[77] S. Chokchaitam and M. Iwahashi, "Lossless, near-lossless and lossy adaptive coding based on the lossless DCT," Proc. IEEE International Symposium on Circuits and Systems, 2002. ISCAS 2002. Voi.l, pp. I-781 -1-784, 2002.

[78] Van der Vleuten, R.J.; "Low-complexity lossless and fine-granularity scalable near-lossless compression of color images", Proceedings. DCC 2002Data Compression Conference, 2002. 2-4 April 2002 Page(s):477.

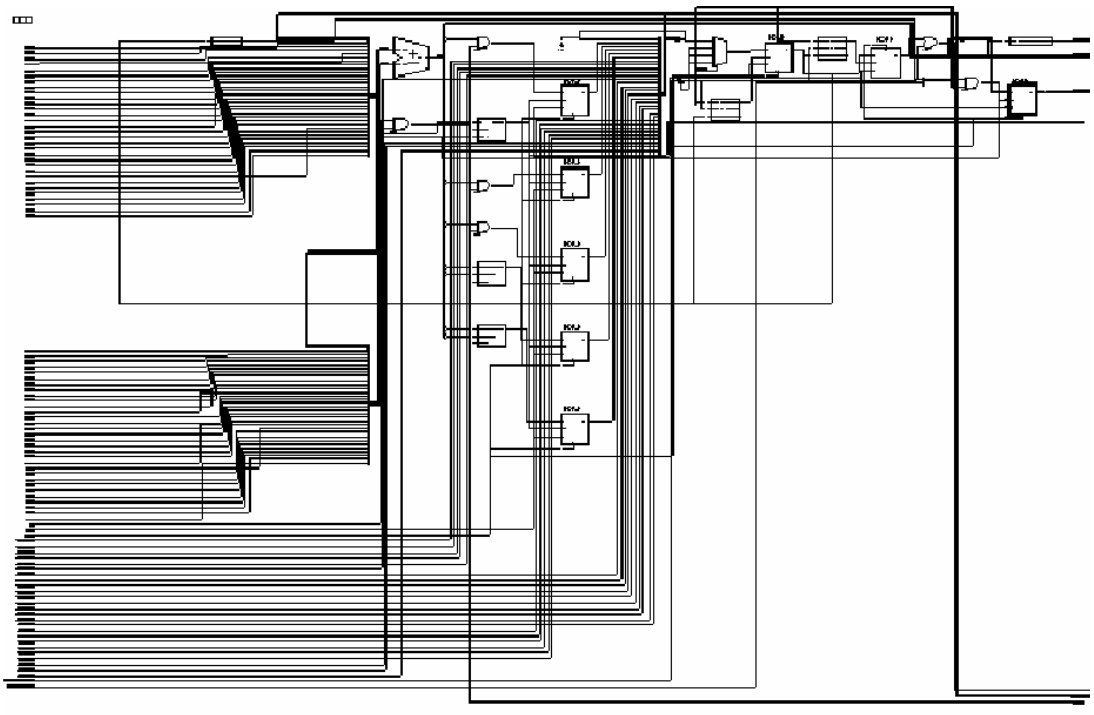
[79] Xin Chen; Ju-fu Feng; Kwong, S.; "Lossy and lossless compression for color-quantized images", Proceedings. 2001 International Conference on Image Processing, 2001. Volume 1, 7-10 Oct. 2001 Page(s):870 - 873 vol.1.

[80] M.J.S. Smith, Application Specific Integrated Circuits, Pearson Education, 2003.

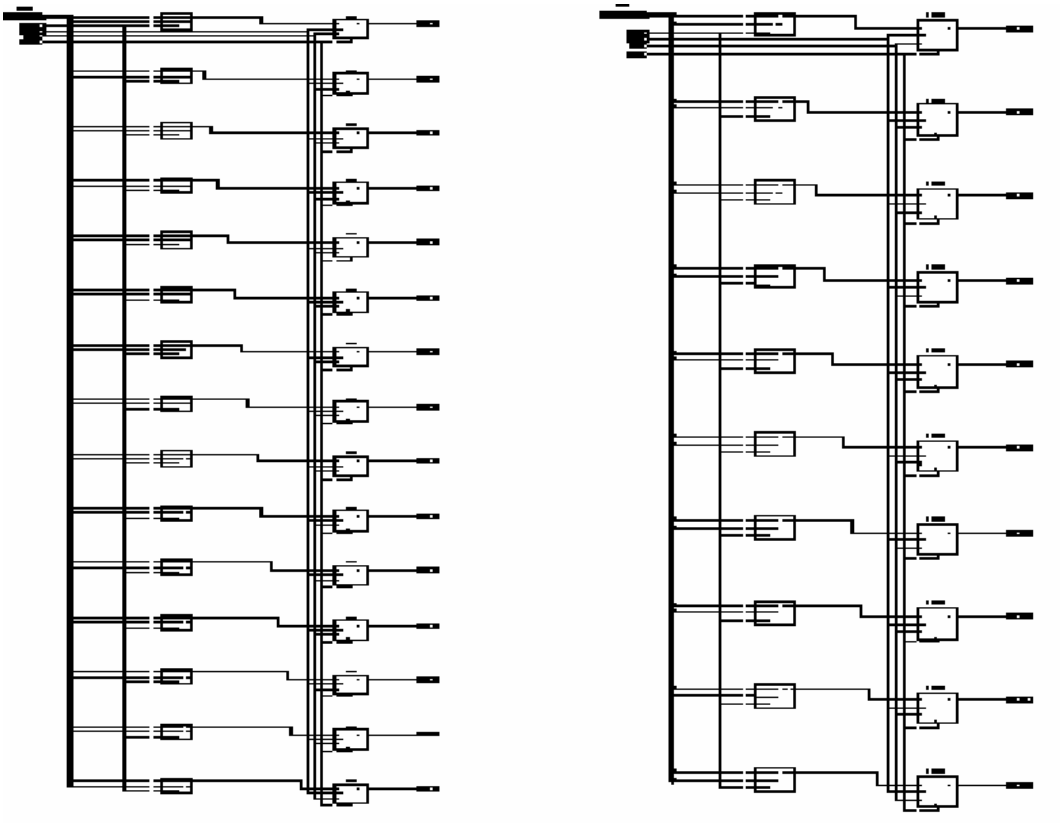
[81] Power calculation site: http://www.xilinx.com/cgi-bin/powerweb.pl

[82] Stephen Trimberger, Field Programmable Gate Array Technology, Kluwer Academic Publishers, 1994.

[83] The Xilinx Home page: www.xilinx.com



Block - 1



Block -2 and Block -3

(Implementation Block Diagram of Arithmetic Compression Technique)